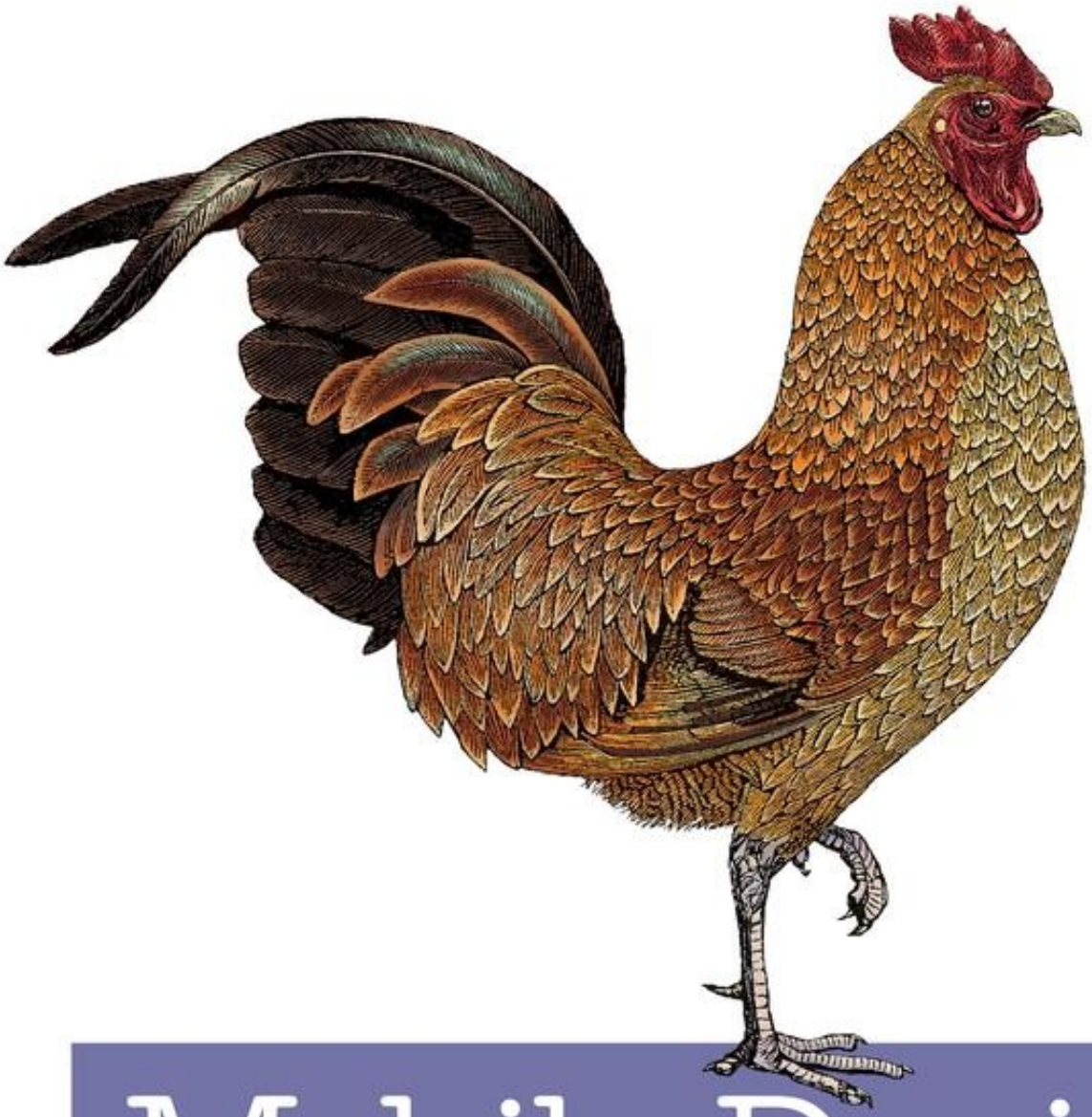
O'REILLY®

2nd Edition



# Mobile Design Pattern Gallery

UI PATTERNS FOR SMARTPHONE APPS

Theresa Neil

*Foreword by Jenifer Tidwell*

# Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps

Theresa Neil


Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

# Special Upgrade Offer

If you purchased this ebook directly from oreilly.com, you have the following benefits:

- DRM-free ebooks — use your ebooks across devices without restrictions or limitations
- Multiple formats — use on your laptop, tablet, or phone
- Lifetime access, with free updates
- Dropbox syncing — your files, anywhere

If you purchased this ebook from another retailer, you can upgrade your ebook to take advantage of all these benefits for just $4.99. Click here to access your ebook upgrade.

*Please note that upgrade offers are not available from sample content.*

# Foreword

To name something is to begin to understand it.

My five-year-old son, like many children, enjoys looking at clouds. Recently, he clued into the fact that different kinds of clouds have different names. And so, being of good geek stock, he proceeded to memorize them — cirrus, cumulus, stratus, cirrostratus, cumulonimbus, altostratus, lenticular — all of the ones I knew, and then some. I'd certainly never heard of "cumulus congestus" before.

Now, when he looks at the sky, he can tell me which clouds are which. More than that, he notices more than he did before, and with greater nuance. He has learned to visually discriminate among cloud types based on texture, color, height, movement, and who knows what else. (They're not always easy to tell apart, of course, but that doesn't bother him.) He can predict, with some accuracy, which ones might drop rain on us and which won't.

And in his limited preschooler fashion, he uses his cloud knowledge to analyze the big picture. "Cirrostratus clouds might mean a warm front," he points out. Or, "Cumulus congestus might turn into cumulonimbus! Then we could get a storm."

Above all, he enjoys knowing these names. Little kids seem to get a kick out of naming the things they love, whether they're clouds, dinosaurs, bugs, cars, dolls, or movie characters. Certainly their imaginations aren't limited by that left-brain knowledge, despite our grown-up romantic biases — my son still sees palaces and ducks and cauliflowers in the clouds, even as he names them "cumulus."

So it is with us grown-ups. That brings us to the topic at hand: by recognizing and naming patterns in interfaces, we "see" those interfaces better. We notice more details, because our brains are more attuned to what we should look for. We can start to predict the workings of the software we use, because we know how certain interface patterns should behave. Then we can tell other people what we see via an expressive new vocabulary.

And how do we learn these patterns?

When my son learned about clouds, the best tool he had was pictures. Lots of pictures. After looking at some of these "catalogs" in books and on websites, he learned to see rather subtle differences between cloud types, some of which are hard to describe verbally.

Likewise, the best way to learn interface patterns is to see visual examples. Now, I'm a writer, so I love words. When not restrained by courtesy, I would happily go on endlessly about what patterns are, how to choose them, and the differences between them! But it's clear to me that anyone who simply wants to design interfaces — that is, anyone who needs to know patterns as one component of their craft knowledge — won't really need all those words. For a given pattern, they need just enough explanation to "get it," and then they need to see a range of well-chosen real-life examples to solidify and internalize that knowledge.

In this book, Theresa Neil has pulled together a spectacular collection of pictures of patterns. I can't imagine the work that went into this, having tried it myself; it's no small feat to review this many mobile apps, see what works best in them, and gather up all these carefully cataloged screenshots.

For mobile interface designers, this book is a treasure. Read it straight through if you'd like, but more than that, use its examples to improve your own designs:

- Use your own judgment about what works well in these examples, and figure out what may work best in the context of whatever you're designing.
- Use it as a sourcebook for design inspiration. I found myself admiring these screenshots for design aspects that had nothing to do with the patterns themselves, such as icon design and color usage.
- Use it to expand your knowledge of how existing apps work, without laboriously downloading and using them all (and on several devices, don't forget).

You might even go out and find your own pattern examples in the mobile apps you use daily. In fact, I'd bet that once you learn these pattern names, you won't be able to avoid doing so. Having had my son point out "cumulus congestus" in the wild a few times, I know it well, and, gosh — I don't know how I ever lived without that knowledge.

Enjoy!

— Jenifer Tidwell

# Preface

Sometimes it’s good to stop and reflect on the many factors that affect usable design. But more often, there’s no time for that — you’ve just got to roll up your sleeves and get to work. This book is for those times.

From one perspective, the mobile world has changed a lot since this book first came out in 2011. Three of the six mobile operating systems I included in 2011 — WebOS, Symbian, and BlackBerry — are no longer contenders in the mobile space.

From another perspective, not that much has changed: out of over 70 patterns from the first edition, most are still with us, with only a handful of new ones added. Those latest patterns, though, exhibit more “mobile-centric” thinking. Designers are finally looking beyond desktop and web metaphors to craft solutions that are organic to mobile interfaces. I expect this to continue, and to accelerate.

Another change: in 2011, I was also optimistic about OS-neutral designs, meaning that perhaps we could as designers and developers create a single interface that would work well on multiple OSs. In fact, the opposite has occurred; distinct design conventions for iOS, Android, and Windows Phone have become more formalized, particularly with regard to navigation.

It’s now more important than ever to understand those OS guidelines, and even more crucial that you are truly familiar with the actual devices your users rely on 24/7/365. I strongly advise that you spend a minimum of six weeks using devices for each OS you are designing for. That way, when you do roll up your sleeves to get to work, your own experience — along with the patterns in this book — will give you the confidence you need to design beautifully usable apps.

## Intended Audience for This Book

*Mobile Design Pattern Gallery* is for product managers, designers, and developers who are creating mobile applications. As companies are defining and refining their mobile strategies, it can be a challenge to find examples of design best practices, especially for multiple operating systems. Whether you have been tasked with designing a simple iPhone application or designing for every popular operating system on the market, these patterns will provide solutions to common design challenges.

## Safari® Books Online

Safari Books Online (*http://my.safaribooksonline.com*) is an on-demand digital library that delivers expert content (*http://www.safaribooksonline.com/content*) in both book and video form from the world's leading authors in technology and business. Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of product mixes (*http://www.safaribooksonline.com/subscriptions*) and pricing programs for organizations (*http://www.safaribooksonline.com/organizations-teams*), government agencies (*http://www.safaribooksonline.com/government*), and individuals (*http://www.safaribooksonline.com/individuals*). Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and dozens more (*http://www.safaribooksonline.com/publishers*). For more information about Safari Books Online, please visit us online (*http://www.safaribooksonline.com/*).

## How to Contact Us

Please address comments and questions concerning this book to the publisher:

O’Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

*http://oreil.ly/mobile-design-2e*

To comment or ask technical questions about this book, send email to:

*bookquestions@oreilly.com*

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

# Acknowledgments

I am in debt to Rich Malley for helping me write this book; I hope this is just the first of many. Many thanks to Cathlin McCullough for pulling together the chapter on social patterns, and Suze Kemper for helping me meet the deadline. And a huge thanks to Ivan Bachev for once again prepping all 1,000 images!

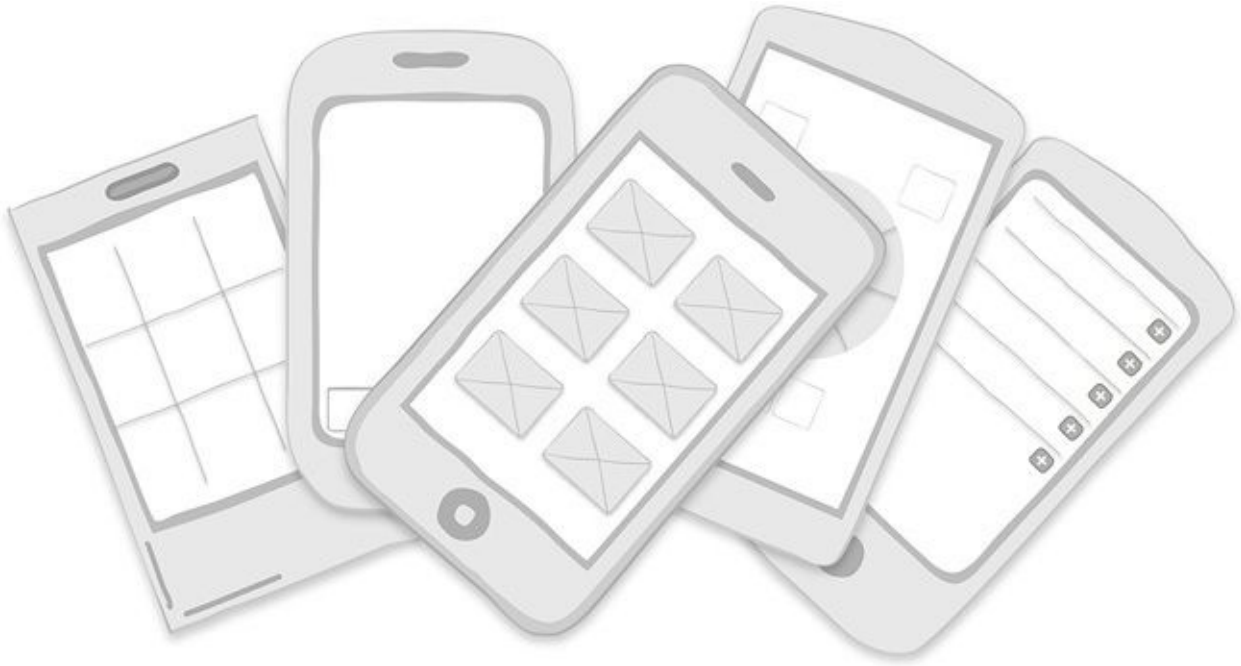
Thank you to Alissa Briggs, Greg Nudelman, and Eli Holder for sharing their stories, and Aaron Jansinski for the pattern illustrations.

I'd also like to acknowledge all the creative and dedicated teams out there designing and developing mobile applications. I feel privileged to use many of the great apps showcased in this book and appreciate all the hard work that went into them.

This is my third book for O'Reilly Media, and it has been a pleasure to work with Mary Treseler and her team again.

Finally, a very warm thank you to the newest member of my family, Marlena Elizabeth Ann, for providing me with the motivation to finish this book before she arrived (with a whole 12 hours to spare)!

# Chapter 1. Navigation



*Primary Navigation Patterns, Persistent*

Springboard, List Menu, Dashboard, Gallery, Tab Menu, Skeuomorphic

*Primary Navigation Patterns, Transient*

Side Drawer, Toggle Menu, Pie Menu

*Secondary Navigation Patterns*

Page Swiping, Scrolling Tabs, Expand/Collapse Panel

I like to read reviews in mobile marketplaces to better understand how people are using apps. The marketplace rating system offers incredibly valuable feedback of a kind that doesn't exist for web and desktop applications. It provides a rich source of information about customer preferences and expectations.

In general, most 4-and 5-star reviews aren't very specific. They often don't go beyond "What a great app; it looks good and works well." But the 1-and 2-star reviews are much more telling; they tend to offer a truer picture of problems users are having with applications. The most common complaints seem to revolve around:

- Crashing

- Lack of key features (e.g., syncing, filtering, account linking)
- Confusing interface design
- Poor navigation (e.g., can’t go back, can’t find things)

The first two issues can’t be fixed with design patterns — they’ll both require user and device testing — but the third and fourth complaints certainly can. Following the common design patterns for navigation will ensure that people can find and use the valuable features in your application.

Good navigation, like good design, is invisible. Applications with good navigation just feel simple and make it easy to accomplish any task, from browsing through pictures to applying for a car loan.

## Primary Navigation Patterns, Persistent

The first set of patterns we'll look at are used for primary navigation, like navigating from one primary category to another, as with the top-level menus of a desktop application. Since the first edition of this book, primary navigation has evolved into two distinct types: persistent and transient.

Persistent navigation encompasses simple menu structures like the List Menu and Tab Menu. As soon as you open an app with persistent navigation, it is immediately clear what the primary navigation options are.

Transient navigation, however, must be explicitly revealed with a tap or gesture. These patterns arise from the constraints of smartphone screen sizes, which have pushed mobile designers to think "outside of the box," literally.

To me, this classic 9-dot puzzle perfectly illustrates the change in thinking around mobile navigation patterns. Give it a try: your challenge is to connect all of the dots using four straight lines or fewer, without taking your pencil off the paper or retracing any of the lines.

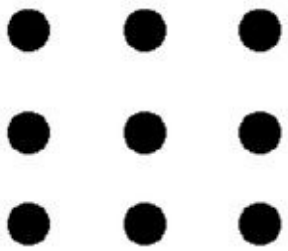


*Figure 1-1. 9-dot puzzle*

How'd you do? You probably figured out that the only way to solve this puzzle is to break free of the artificial boundaries. In the mobile world, it's called thinking "off-canvas."

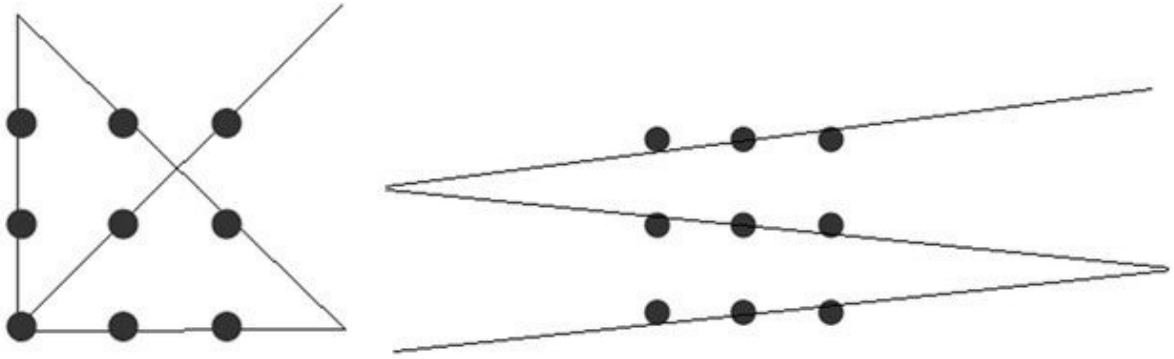


*Figure 1-2. Two of the possible solutions to the 9-dot puzzle*

This off-canvas thinking inspired the Side Drawer, which is currently one of the most popular primary navigation patterns in iOS and Android apps.

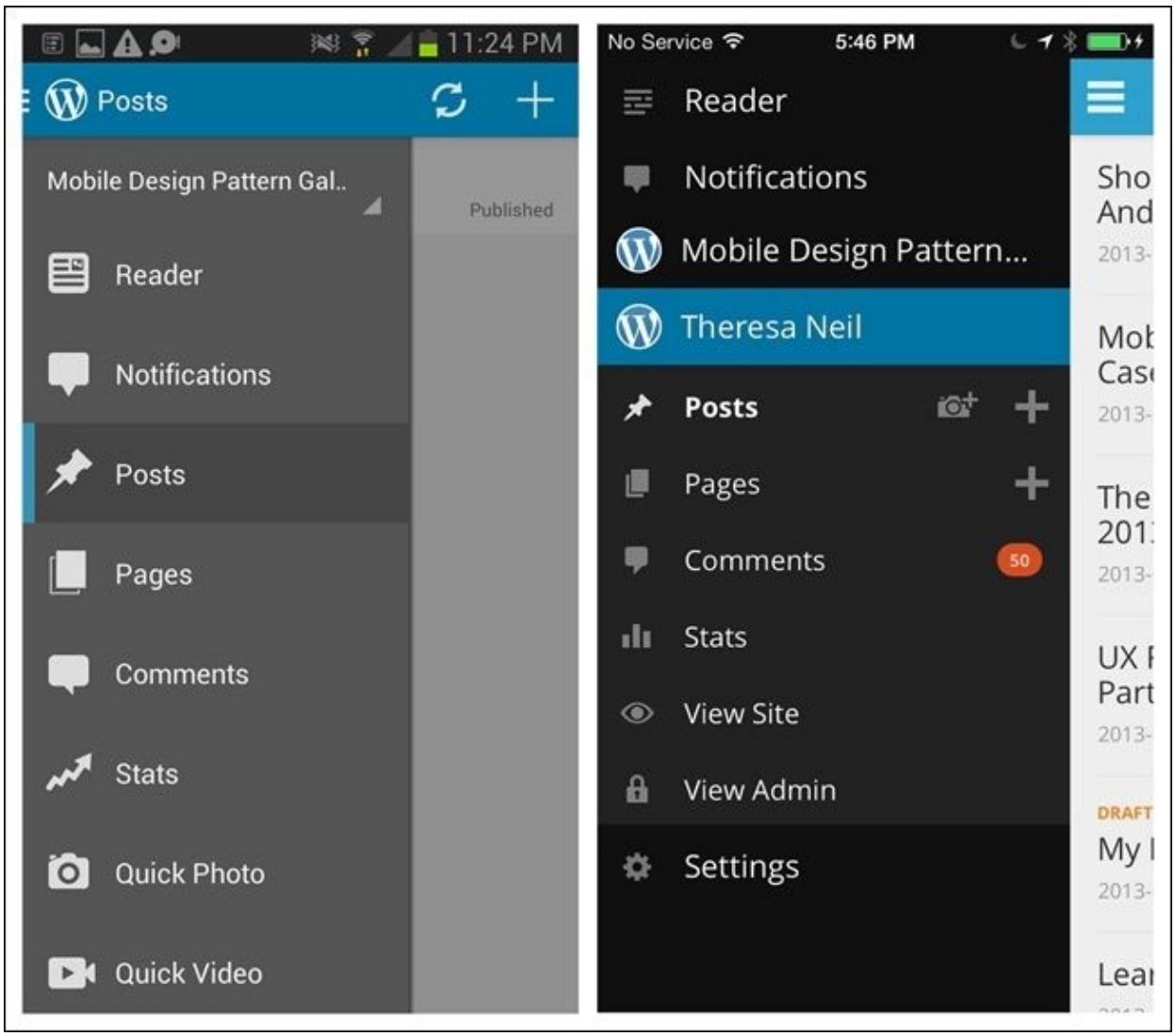


*Figure 1-3. WordPress for Android and iOS: Side Drawer represents "off-canvas" thinking*

Windows Phone 8 and Ubuntu Touch, a new open source mobile OS, are both highly influenced by this move to break artificial boundaries as well.


*Figure 1-4. Windows Phone Panorama control*

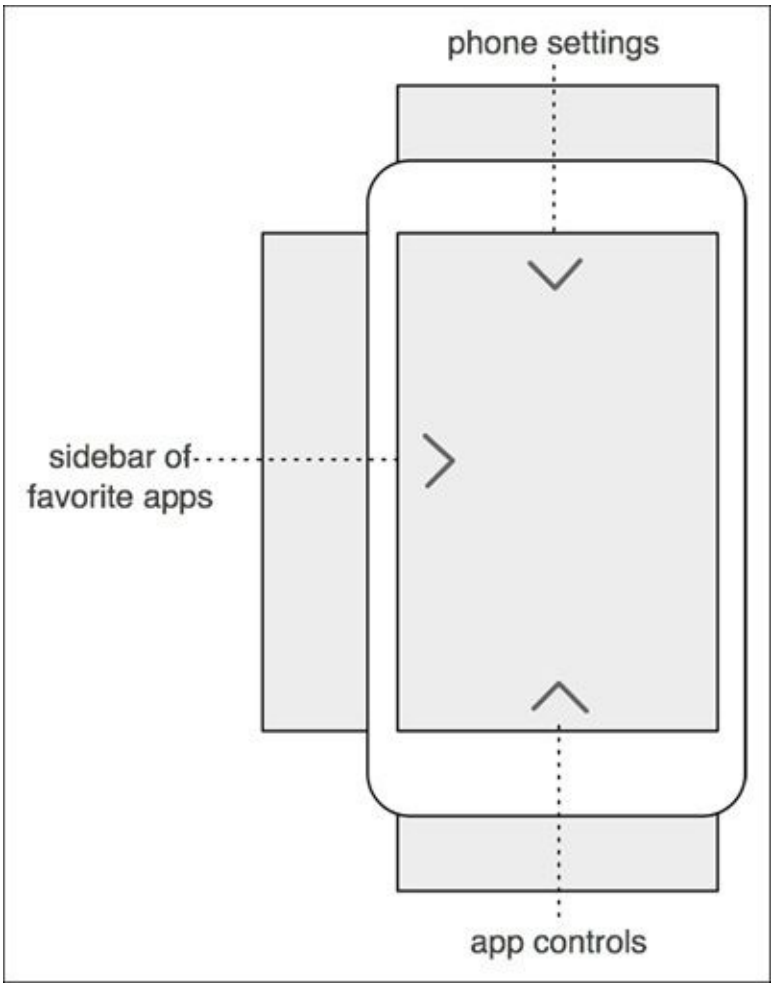



*Figure 1-5. In Ubuntu, you can swipe the screen edges to reveal settings and menus, leaving the screen entirely free for the application's content*

Designers have also made a significant shift in design thinking, layering content instead of relegating the UI to a single plane. Twitter's early iPad design was a fantastic example of how 3D layers and gestures can create a uniquely mobile experience: the left panel is the menu bar, the middle panel is the listing of contents, and the right panel displays those contents. Tapping an item in the middle section collapses the left menu bar and shows a preview of the contents within the right panel. When tapped, the right panel expands to cover about 70% of the screen.


infinvision
Timeline
Mentions
Messages
Lists
Profile
Search
TheOpenLabel
@infinvision: Thank you for the great tweet!!
infinukas
Follow us on @infinvision
YoFolyo
Welcome to @infinukas! Check out their profile here: folyo.me/designers/202
frank_martin_86
Love the illustration on infinvision.com @infinukas
bonjour404
Page 404 du jour : InfinVision - http://tumblr.com/x9f3rr4ji7 #oops @Infinukas
EvelinaSrna
@Infinukas o yeah me too! Less is more :)
malemans
Si les interesa estudiar algunos casos sobre SEO les recomiendo infinvision.com @Infinukas
malemans
siguiendo a @Infinukas gracias por los contenidos!! :)
dpmachado
paper.li Diego Machado is out! http://bit.ly/i36BrN ▸ Top stories today via @bamboobrasil @infinukas @difluir @vskaren @designconcursos
TheOpenLabel
@infinvision: Th
tweet!!
TheOpenLabel
@infinvision: Thank yo
infinvision
Discovery of the day: M
people's opinions abou
simply scanning its labe
bonjour404
Page 404 du jour : InfinVision - http://tumblr.com/x9f3rr4ji7 #oops @Infinukas
xoKee_Yan_Uh
RT "@inDependent_ox: #FREE @SHI_millz & @SwagChampNoodle . . they cant hold you love birds forever . . . #oops ! BOL ! ."
ohshannon3
Lol totally just sent a cute text saying I love you baby to my coworker Marta... #oops you're not Cam. Awk
brittbirrrd
why do you act so snooty? you're not even pretty enough to do that. or pretty at all. #oops #sorrynotsorry
emeriearellano
It felt like the right thing to say at that moment.. but, it really wasn't. #oops
megankayglenn
Just skinny dipped with a minor... @MadisonMcDonnel #oops #mexicanprisonsarefunright
jessiedarlingx
Save Search
pierotucci
History always fascinates me, Love and War in Tuscany (part 1) trustandtravel.com/blog/iris-orig... via @twitter_username
Katharina's Italy
All
Art & Culture
Food & Wine
Things to do
Fun with kids
Life at the villa
Shopping
Interior design
Love and War in Tuscany (part 1)
La Foce and Iris Origo - Love and War in Tuscany

*Figure 1-6. Early version of Twitter for iPad: layers and gestures took advantage of the mobile platform*

When deciding between persistent and transient navigation, ask yourself a few questions:

- Is your application “flat”? Are the menu categories equivalent in hierarchy, and are there *just a few primary categories* (i.e., three to five) in the app?
- Do your users need the menu to be *always visible* for quick access?
- Do the menu categories have *status indicators*, like the number of unread emails, for instance?

If you answered “yes” to one or more of these questions, it’s probably best to stick with persistent navigation. Now let’s take a look at those patterns.

## Springboard

The Springboard pattern, also called a Launchpad, was the most popular navigation pattern in 2011. This design is a landing screen with options that act as launch points into the application.

One of the reasons for its popularity was that it worked equally well across platforms. At the time, many of us were still thinking in terms of OS-neutral designs that allowed for consistency and reuse. It was also popular because up to nine options (in a 3×3 grid) could be displayed, compared to the limits of three to five tabs imposed by iOS and Android tab bars. And by adding a paging indicator (those little dots at the bottom), designers could provide even more menu options.

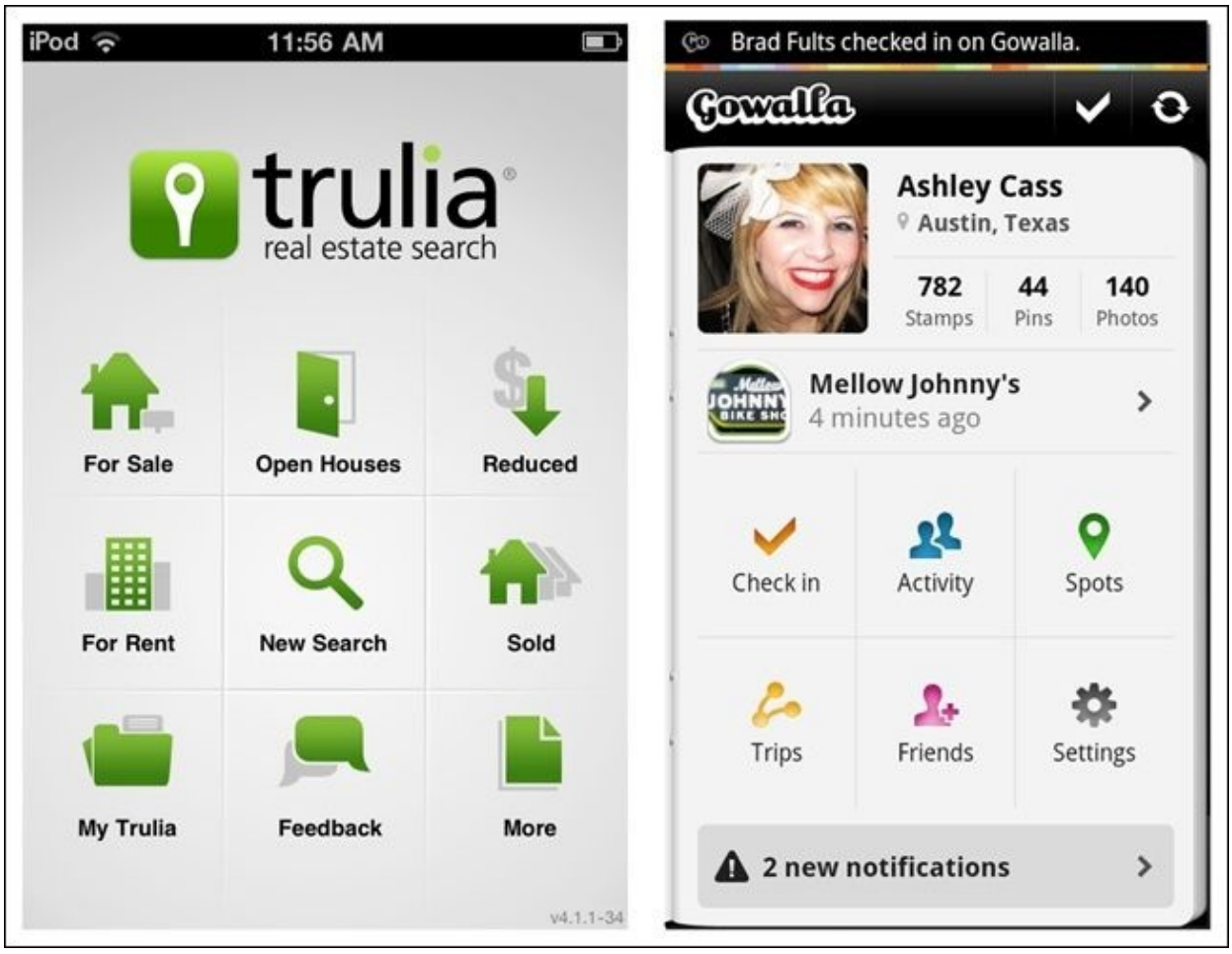



*Figure 1-7. Trulia for iOS and Gowalla for Android*


*Figure 1-8. Facebook for iOS and LinkedIn for Android: Springboard designs from 2011*

The main drawback of the Springboard pattern is that it flattens all options to the same level of importance. Enter the Side Drawer pattern, first designed by Aza Raskin for Firefox Mobile (*http://www.azarask.in/blog/post/firefox-mobile-concept-video/*), and adopted by Path in 2011. This pattern accommodates more options than a tab bar, and those options can be logically grouped to communicate importance and/or hierarchy. We'll discuss it more later in this chapter.

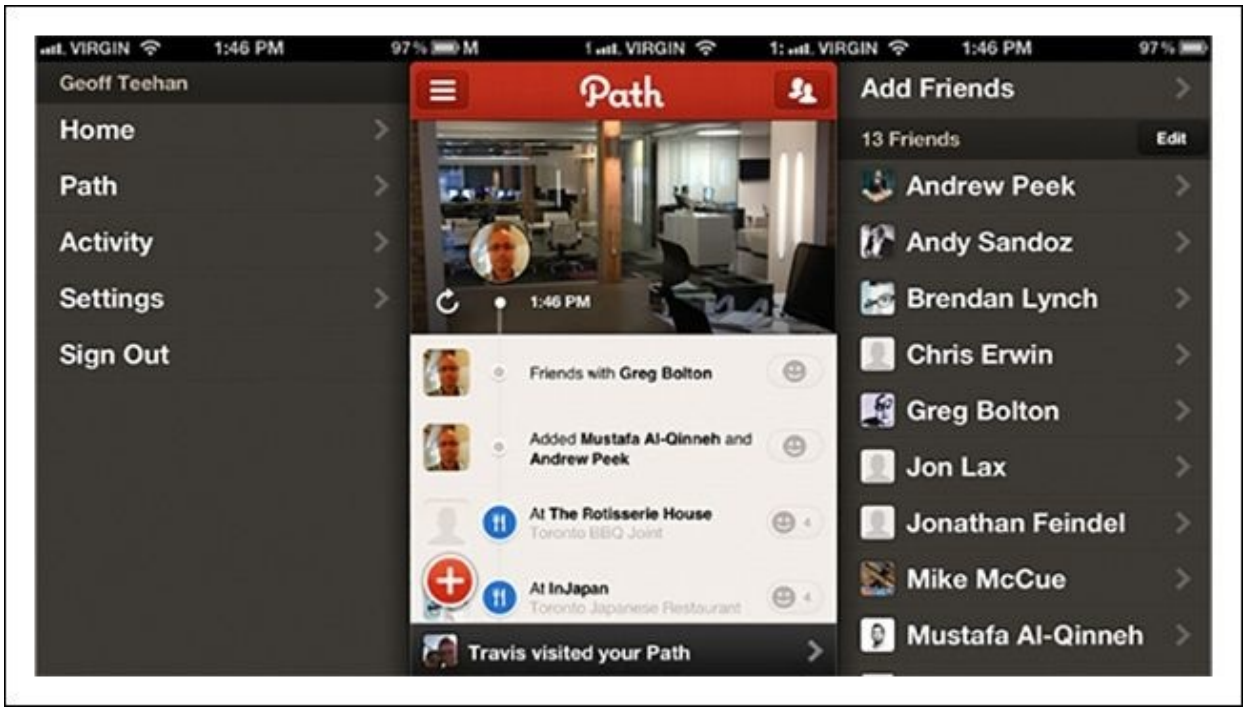



*Figure 1-9. Path for iOS (November 2011): enter the Side Drawer*

However, the Springboard pattern is not dead. Android, iOS, and Windows Phone all still use this navigation pattern at the OS level.


*Figure 1-10. iOS 7, Android KitKat, and Windows Phone 8 all use the Springboard pattern at the OS level*

And there are still apps with traditional implementations of the Springboard pattern around. LearnVest, BBC Radio, and Vimeo use basic 4-, 6-, and 9-grid layouts, respectively.


*Figure 1-11. LearnVest for iOS, BBC Radio for Windows Phone, and Vimeo for Android: traditional Springboard alive and well within apps*

Orbitz and EasyJet vary the icon treatment and grid layout to introduce visual hierarchy to the menu.


*Figure 1-12. Orbitz for iOS and EasyJet for Android: graphic treatment and layout imply a hierarchy*

Windows Phone has pushed the Springboard pattern the farthest with *tiles*. Tiles can be live or static, and come in three different sizes. Live tiles convey dynamic information like number of calls missed, details of your next appointment, or the avatar of your last caller. Read the Windows Design Guide for more about tiles at *http://bit.ly/1hsl2R5*.

Windows Phone tiles can be used for primary navigation or paired with the Panorama control as a secondary navigation pattern. Examples from three apps show their versatility. CalendarPro uses live tiles for primary navigation and NBC News for subnavigation, while Evernote uses static tiles for subnavigation.


*Figure 1-13. CalendarPro, NBC News, and Evernote for Windows Phone: versatile tile implementations*

Evernote Hello for Android and iOS uses a tile-inspired design for the Springboard. Users can customize the UI, first by adding people, then by adding meetings.


*Figure 1-14. Evernote Hello for Android: tile-inspired customizable Springboard*

## Cards

Cards may seem familiar to those of us who had a Palm in 2010–2011. Card navigation is based on a card deck metaphor, including common card deck manipulations such as stacking, shuffling, discarding, and flipping.

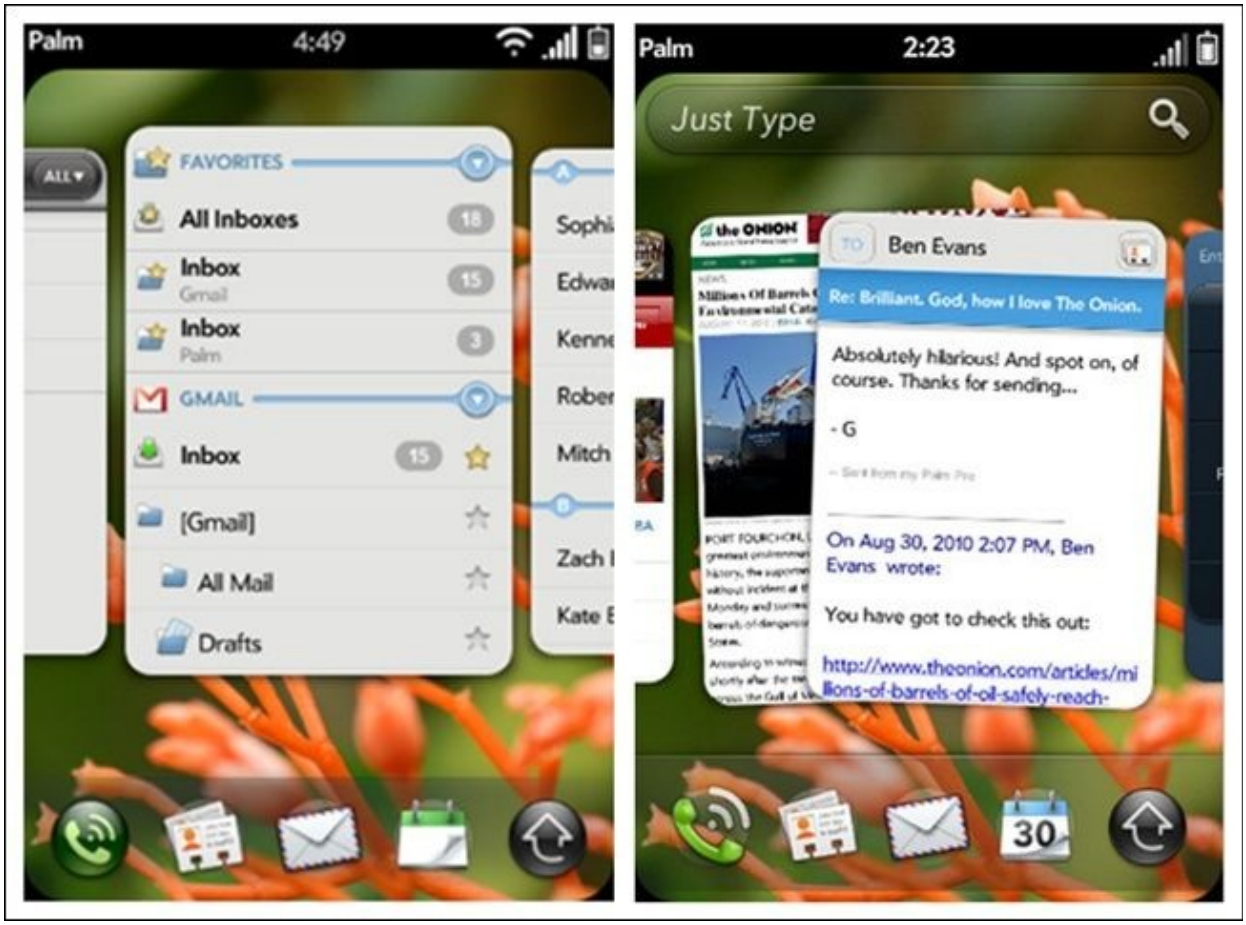



*Figure 1-15. Palm webOS circa 2010–2011: apps fanned out like cards in a deck*

This pattern has become popular again with the release of Google Now, which stacks information-rich cards vertically to display a long list of launch points into the app, or quick actions in context.

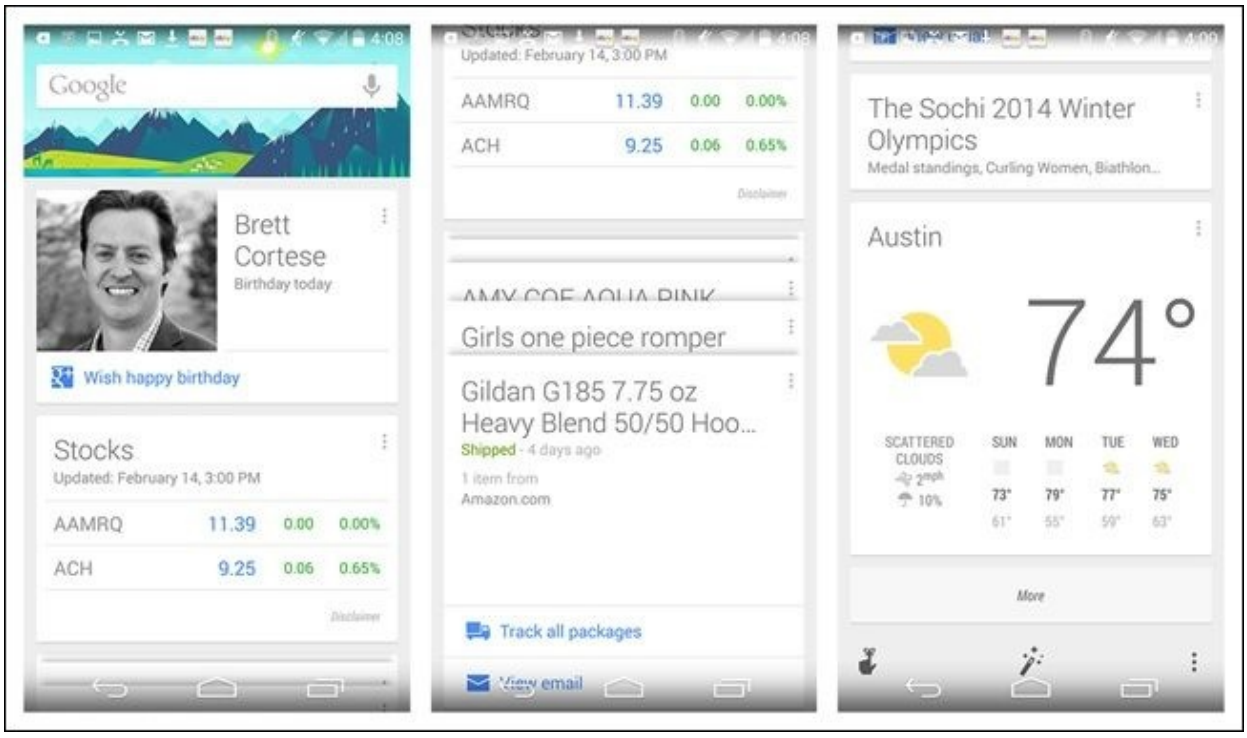



*Figure 1-16. Google Now for iOS and Android: Cards for primary navigation*

In a similar vein, Jelly and Potluck use Cards as the primary means to navigate and interact with content. With Jelly, when you swipe the card down to remove it from the screen — indicating that you can't help answer the posted question — a new card replaces it. With Potluck, swiping left on the top card in the stack will skip the story; swiping right will move it to a new pile, the "keep" pile.

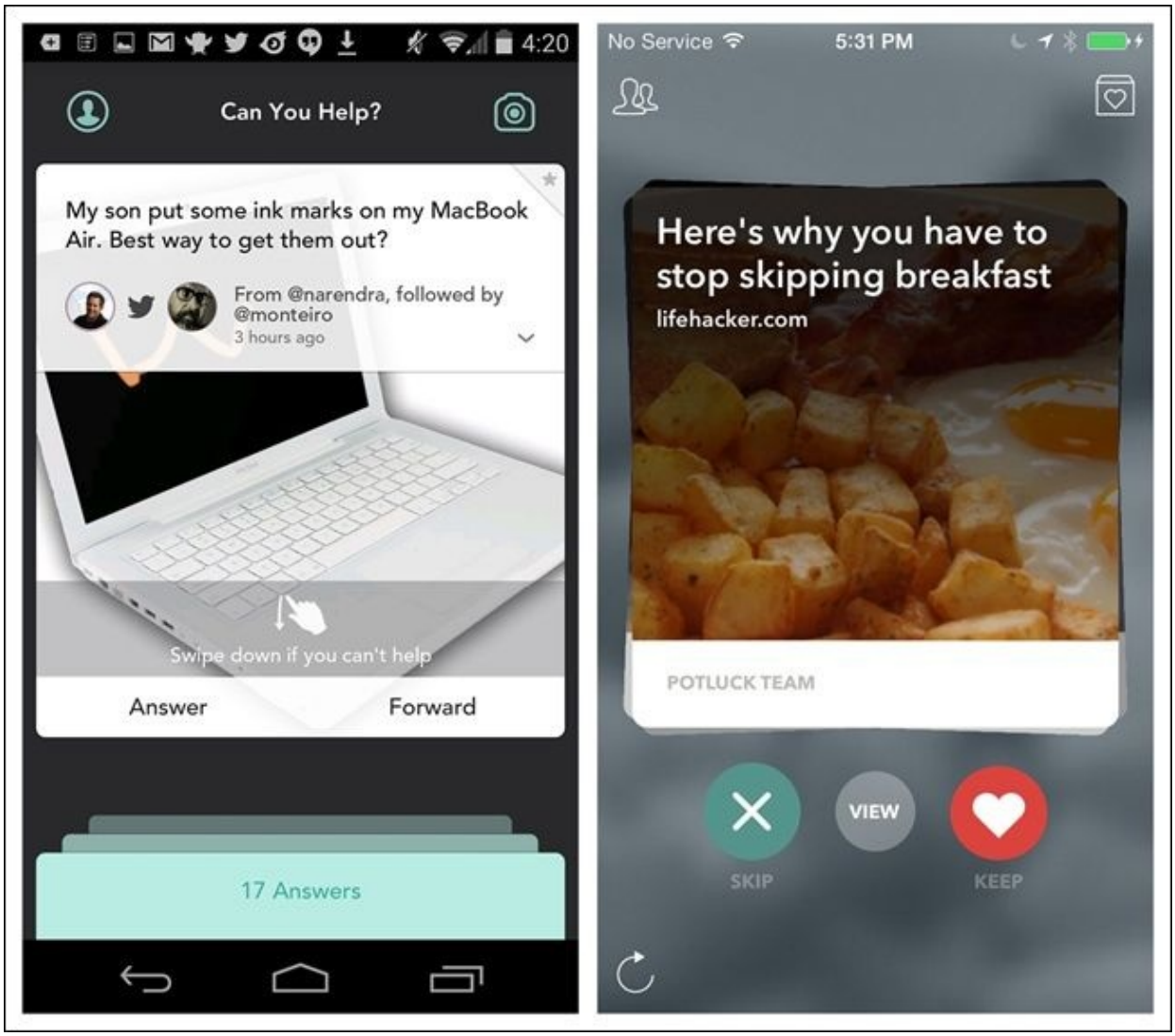



*Figure 1-17. Jelly (http://www.youtube.com/watch?v=bCDB_TrAhSY) and Potluck (http://www.youtube.com/watch?v=pcfNFuvvdrA) for iOS*

Facebook and Pinterest use the visual style of Cards but are missing the gesture-based interactions of the aforementioned examples. This makes them more like stylized list elements than true Cards.

For an example of the Card pattern gone wrong, see the discussion of Alaska Airlines in the section Novel Notions in Chapter 11.

**NOTE**

Cards provide an elegant way to display content for browsing. A true Card pattern will offer interactions like stacking, swiping, or flipping.

## List Menu

The List Menu pattern is similar to the Springboard in that each list item is a launch point into the application, and switching modules requires navigating back to the list. Apple (*http://bit.ly/1dZDU8J*) calls this *hierarchal navigation*:

> In a hierarchical app, users navigate by making one choice per screen until they reach their destination. To navigate to another destination, users must retrace some of their steps — or start over from the beginning — and make different choices. Settings and Mail are good examples of apps that use a hierarchical structure.

The Kayak, Day One, and AroundMe apps illustrate various implementations of the List Menu.

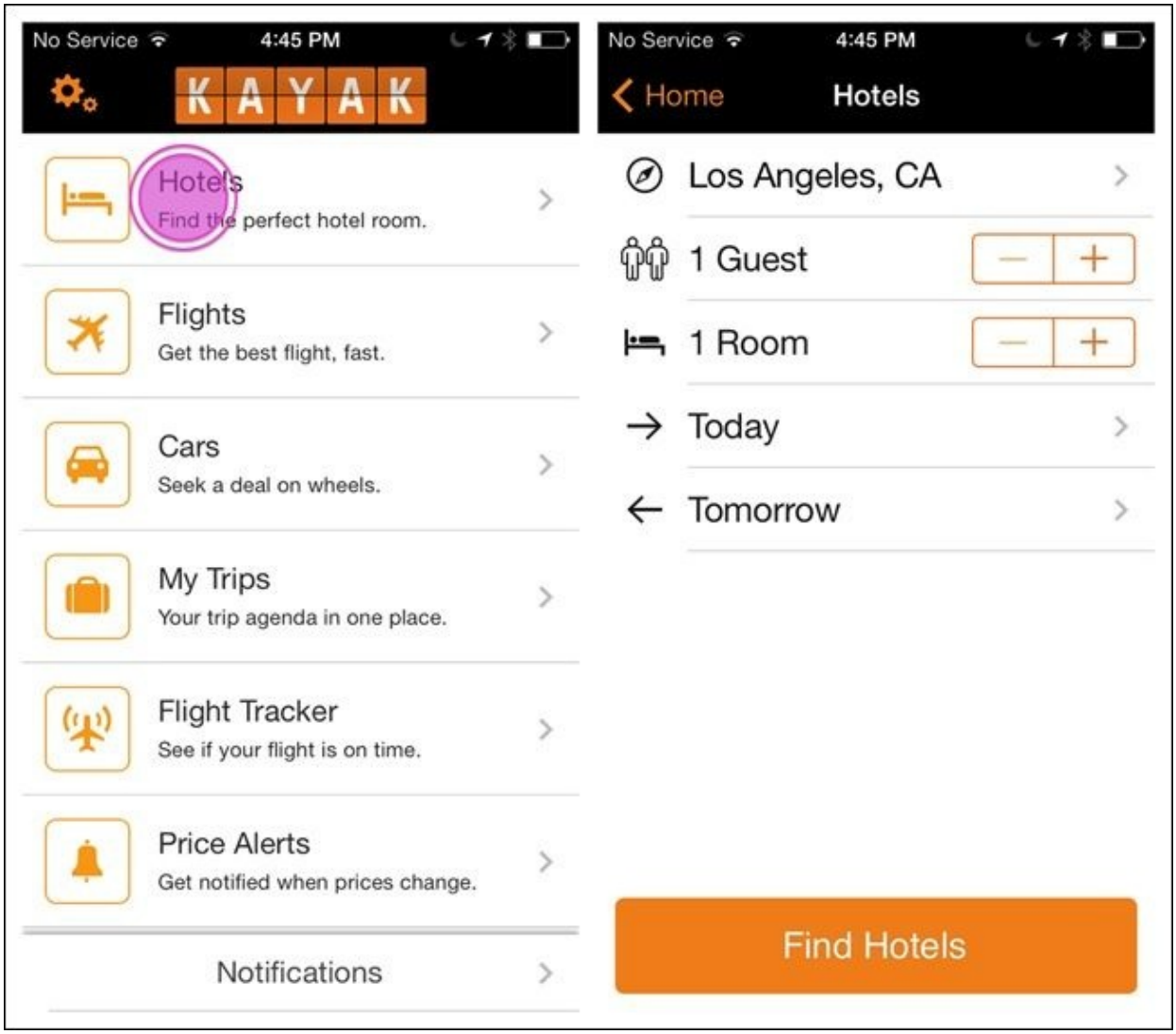



*Figure 1-18. Kayak for iOS: tap Home (screen at right) to return to the List Menu*

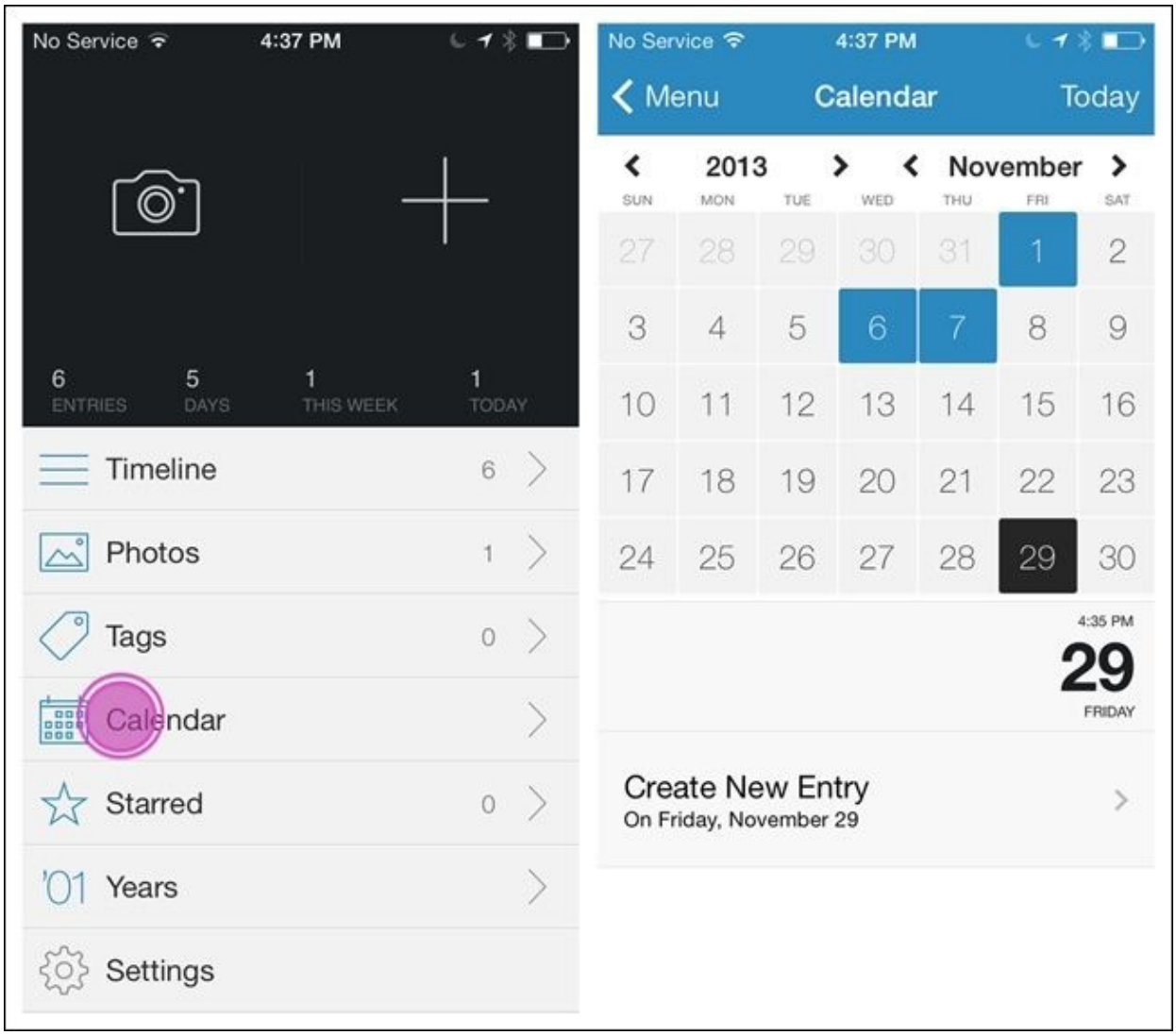



*Figure 1-19. Day One for iOS: List Menu as primary navigation*

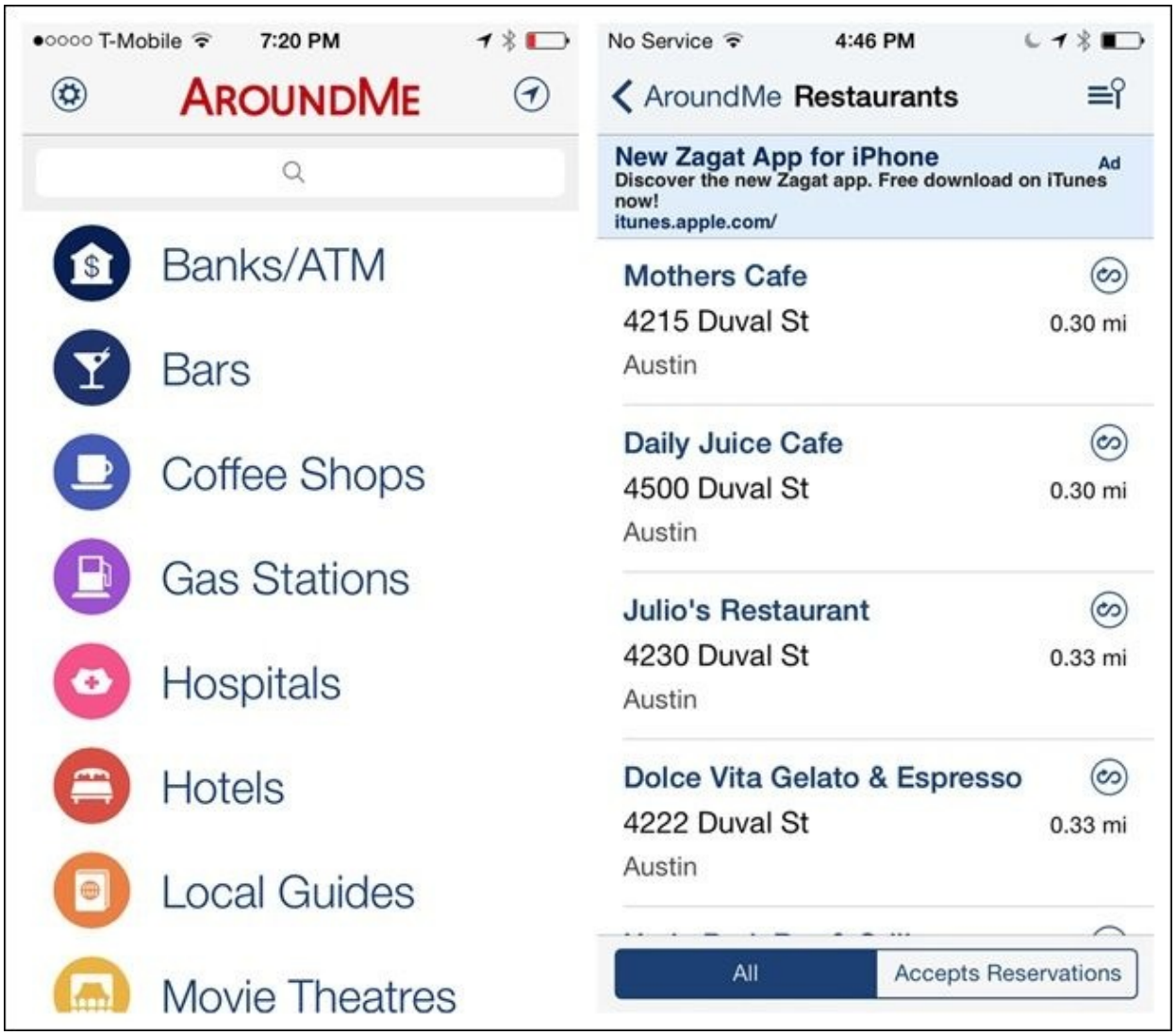



*Figure 1-20. AroundMe for iOS: "Home" or "Menu" might've been a better choice for the Back button label*

The List Menu navigation pattern is similar in Android, but the Back button is called the Up button, conveying the pattern's hierarchical structure, as described in the Android documentation:

> The Up button is used to navigate within an app based on the hierarchical relationships between screens. For instance, if screen A displays a list of items, and selecting an item leads to screen B (which presents that item in more detail), then screen B should offer an Up button that returns to screen A. If a screen is the topmost one in an app (that is, the app's home), it should not present an Up button.

An example of this is shown in the eBay app; the Up button is the app icon preceded by a left chevron. Note that most users expect the chevron *plus* the logo or icon to be tappable.

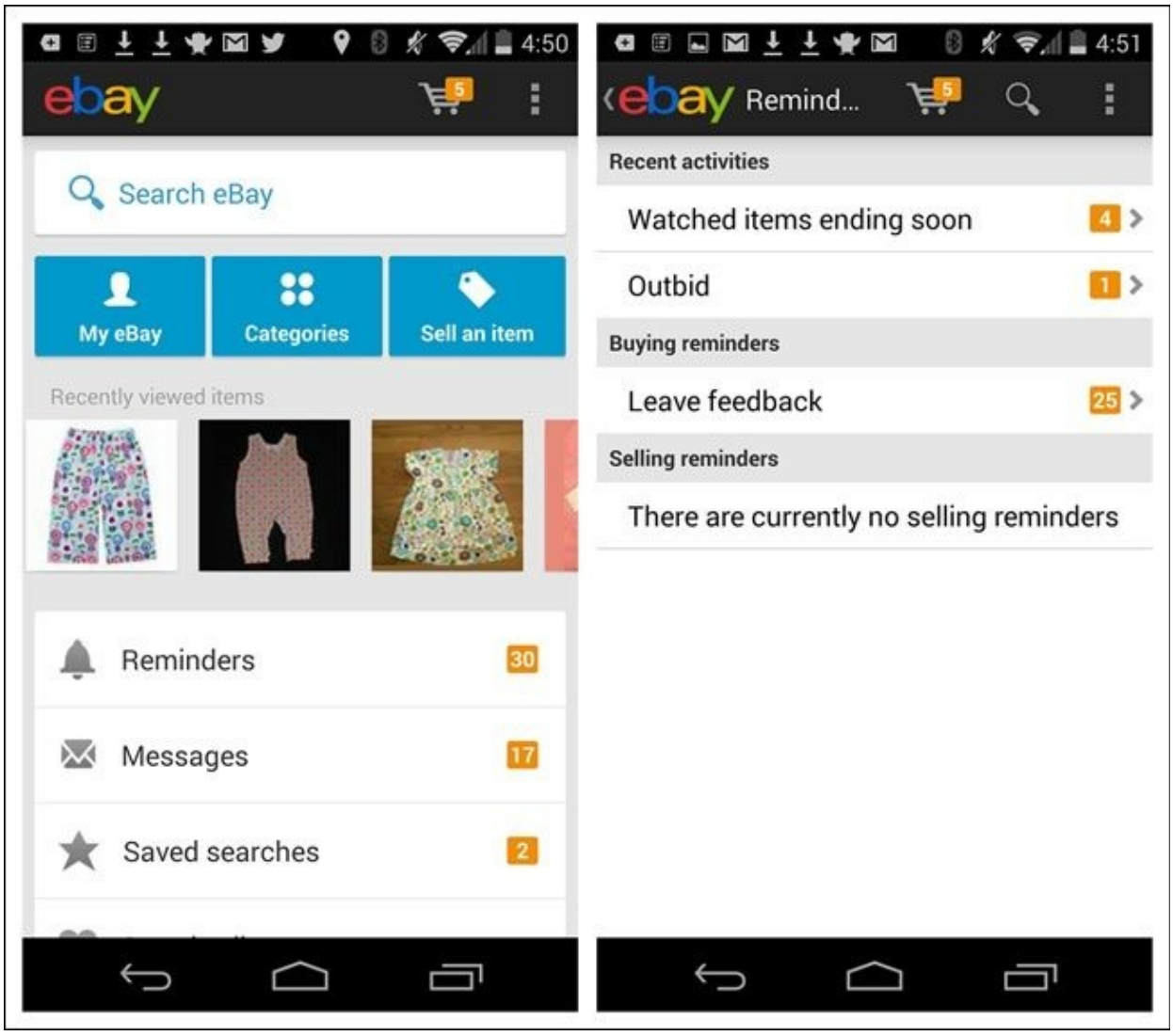



*Figure 1-21. eBay for Android: the chevron is the "Up" button*

**NOTE**

Consider List Menus for navigating within a hierarchy. They also work well for menus with long item names, and where items need descriptions as well as titles. Follow OS conventions for implementing this navigation pattern.

## Dashboard

The Dashboard pattern is similar to the Springboard and List Menu patterns.

With a quick glance, a good Dashboard gives the user a snapshot of the most relevant information she needs to know, without making her navigate into another screen.

When you design the drill-down screens, the rules for providing navigation back to the Dashboard are the same as with the List Menu and Springboard. See Chapter 6 for more on the Dashboard design pattern.

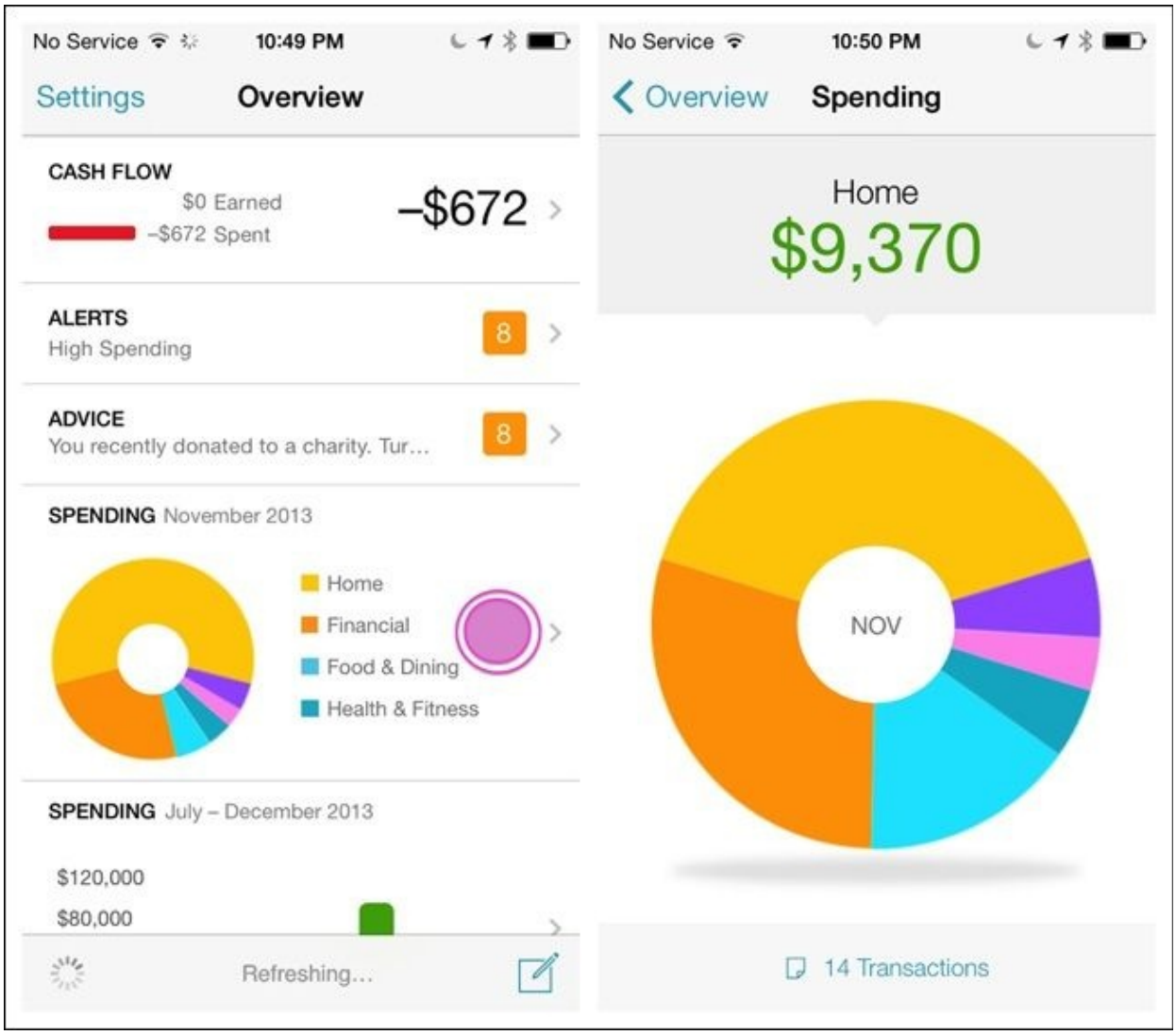



*Figure 1-22. Mint for iOS: Dashboard makes data the launch points*

> **NOTE**
>
> Use a Dashboard when it makes sense to use key metrics or data as launch points into the app. But don't overload the Dashboard; conduct research to determine which key metrics or data to include.

## Gallery

The Gallery pattern displays live content — like news stories, recipes, or photos — arranged in a grid (as with Recipeas and Square Wallet), a carousel (as with LinkedIn Pulse and BBC News), or a slideshow.

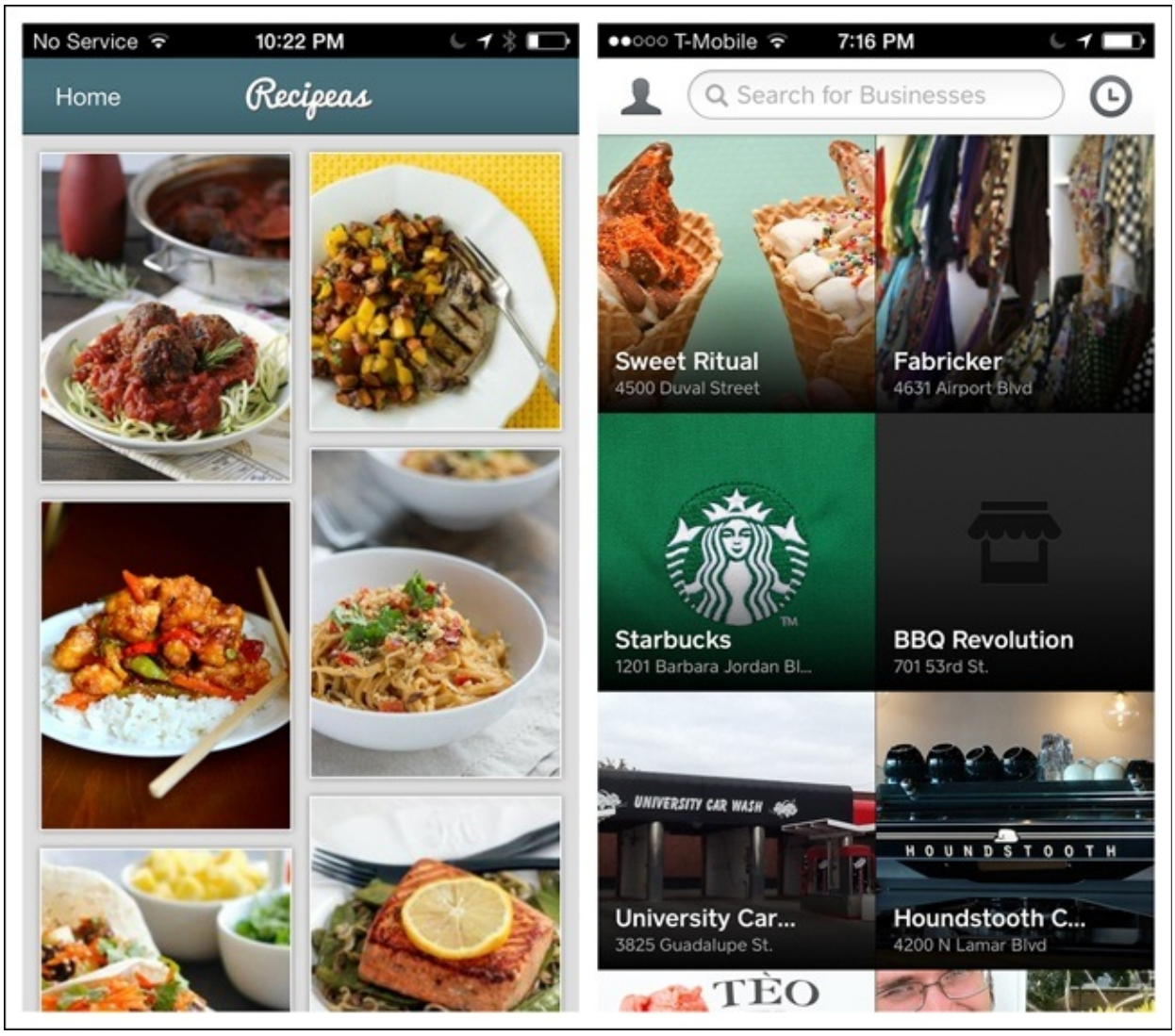



*Figure 1-23. Recipeas and Square Wallet for iOS: galleries present individual, nonhierarchical items*

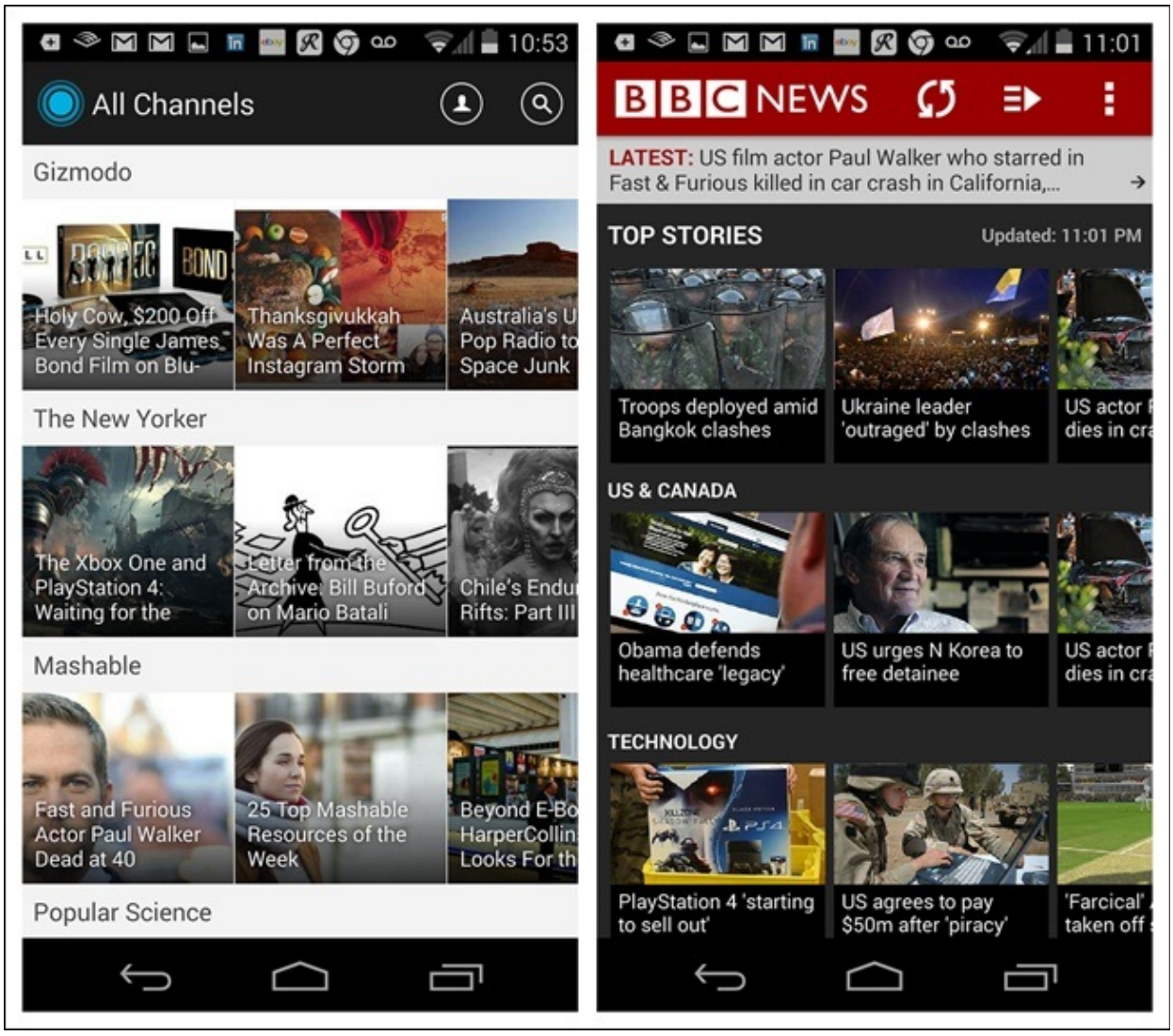



*Figure 1-24. LinkedIn Pulse and BBC News for Android: subtitles are easier to read than overlays*

Notice the BBC News example is easier to scan than the LinkedIn Pulse example, because the titles are below the photo instead of overlaid on them.

> **NOTE**
>
> The Gallery pattern works best for showing frequently updated, highly visual content where no hierarchy is implied.

## Tab Menu

Android, iOS, and Windows Phone each have their own specific nomenclature and design guidelines for Tab Menus. I'm going to go over them here, because it is important that you understand them, even if you choose to deviate from them

in your design iterations.

### iOS

Since the first release of iOS, Apple has recommended (*http://bit.ly/1dZF9Vt*) the Tab Bar for navigating *flat apps*:

> In an app with a flat information structure, users can navigate directly from one primary category to another because all primary categories are accessible from the main screen. Music and App Store are good examples of apps that use a flat structure.

Interestingly, Facebook recently has returned to the Tab Bar after two years of using Side Drawer navigation. Read more about its user testing process and results at *http://tcrn.ch/1dZFlUF*.

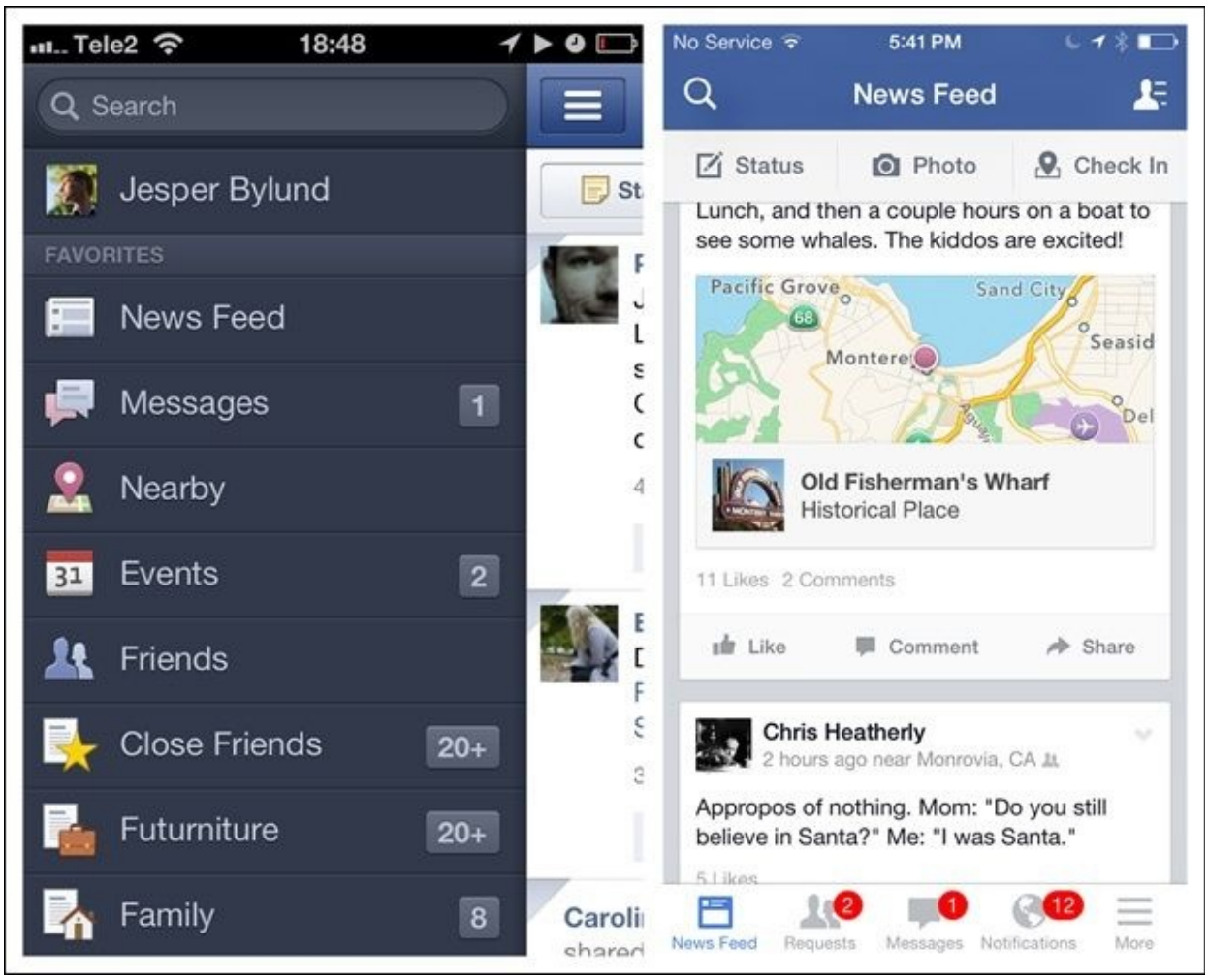



*Figure 1-25. Facebook for iOS, old and new: Tab Bar (right) beat out the Side Drawer (left) and other navigation patterns in 10-million-user test batches*

The iOS Tab Bar is restricted to five menu items. If the application has more than five primary categories, a More option can be provided as the fifth tab on

the right.

It is important to understand the difference between the Tab Bar and Toolbar in iOS. The Tab Bar is for navigating the main categories of the application; the Toolbar presents the tools, or possible actions, for a specific screen.

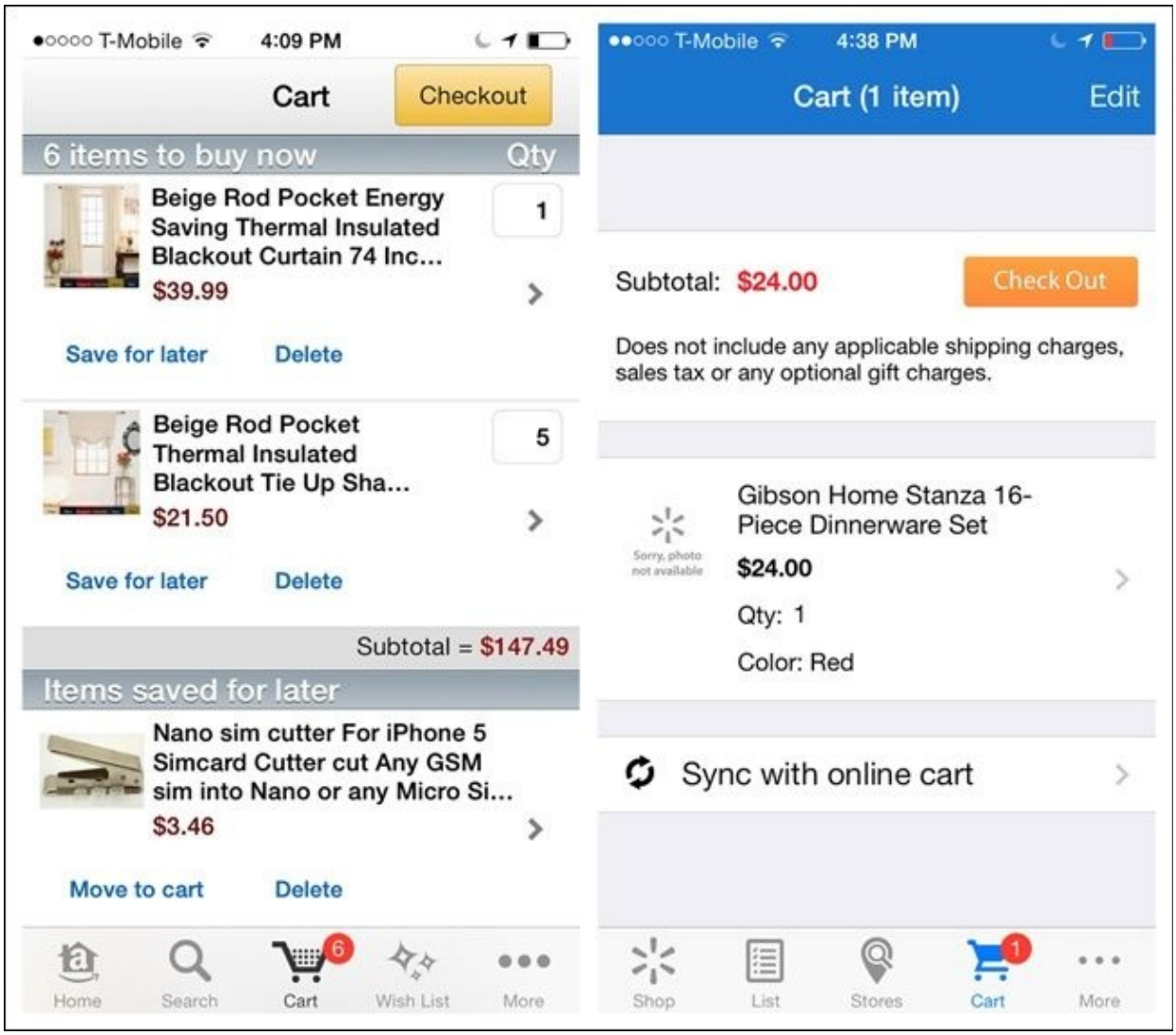



*Figure 1-26. Amazon and Walmart for iOS: Tab Bar treatments*


*Figure 1-27. Tab Bar (left) has menu items; Toolbar (right) has tools for actions*

Some applications, like Instagram and RunKeeper, rely so heavily on the user taking a single action (like taking a picture or starting a run) that they place *calls to action* (a single action more prominent than the rest) in their Tab Bars.

If you design for this variation, make sure the selected tab is conspicuous. It's hard to tell where you are in Everlapse and Tumblr, for instance, because the selected tabs are overshadowed by the visual emphasis on the action buttons.

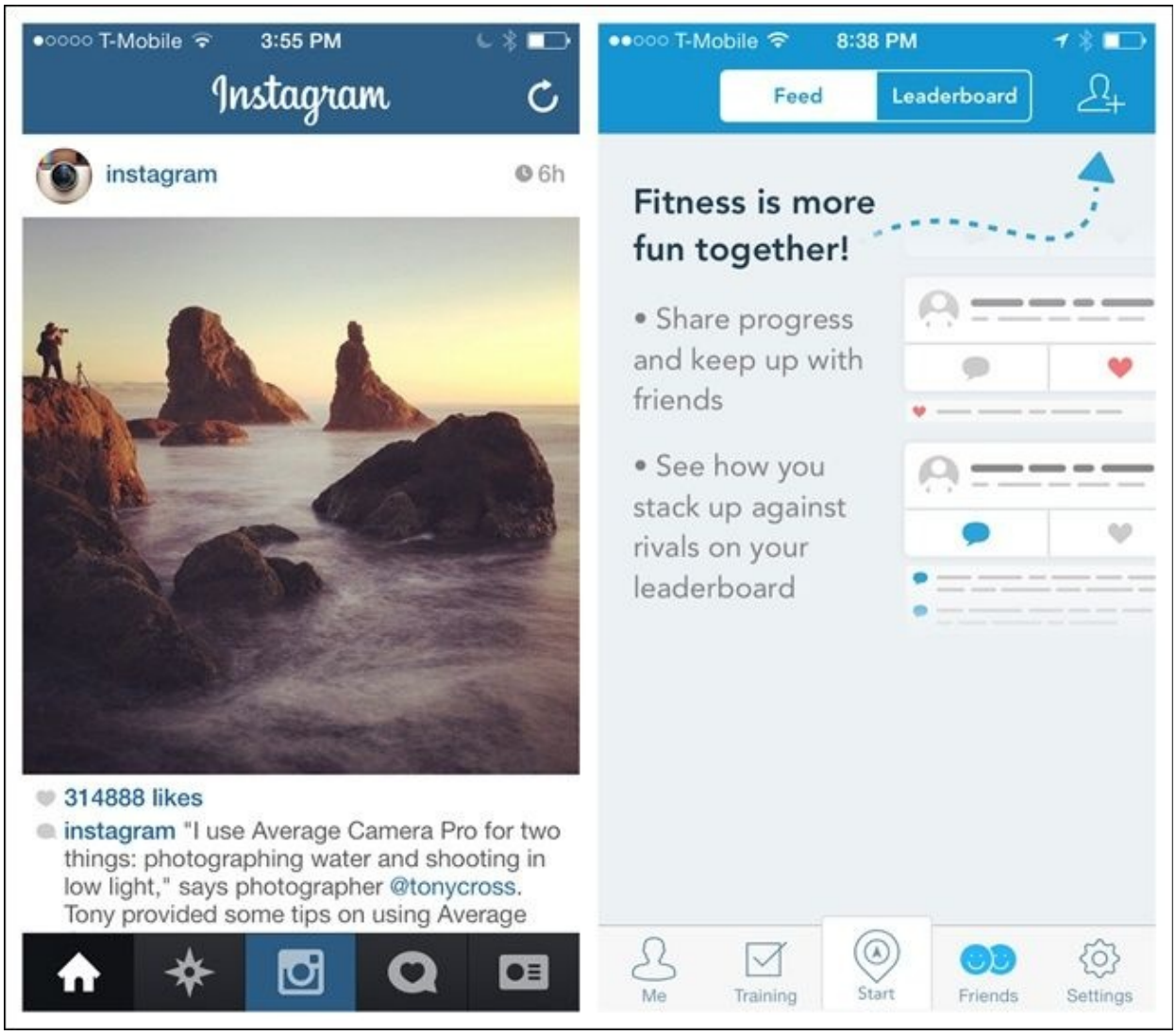



*Figure 1-28. Instagram and RunKeeper for iOS: calls to action in the Tab Bar*

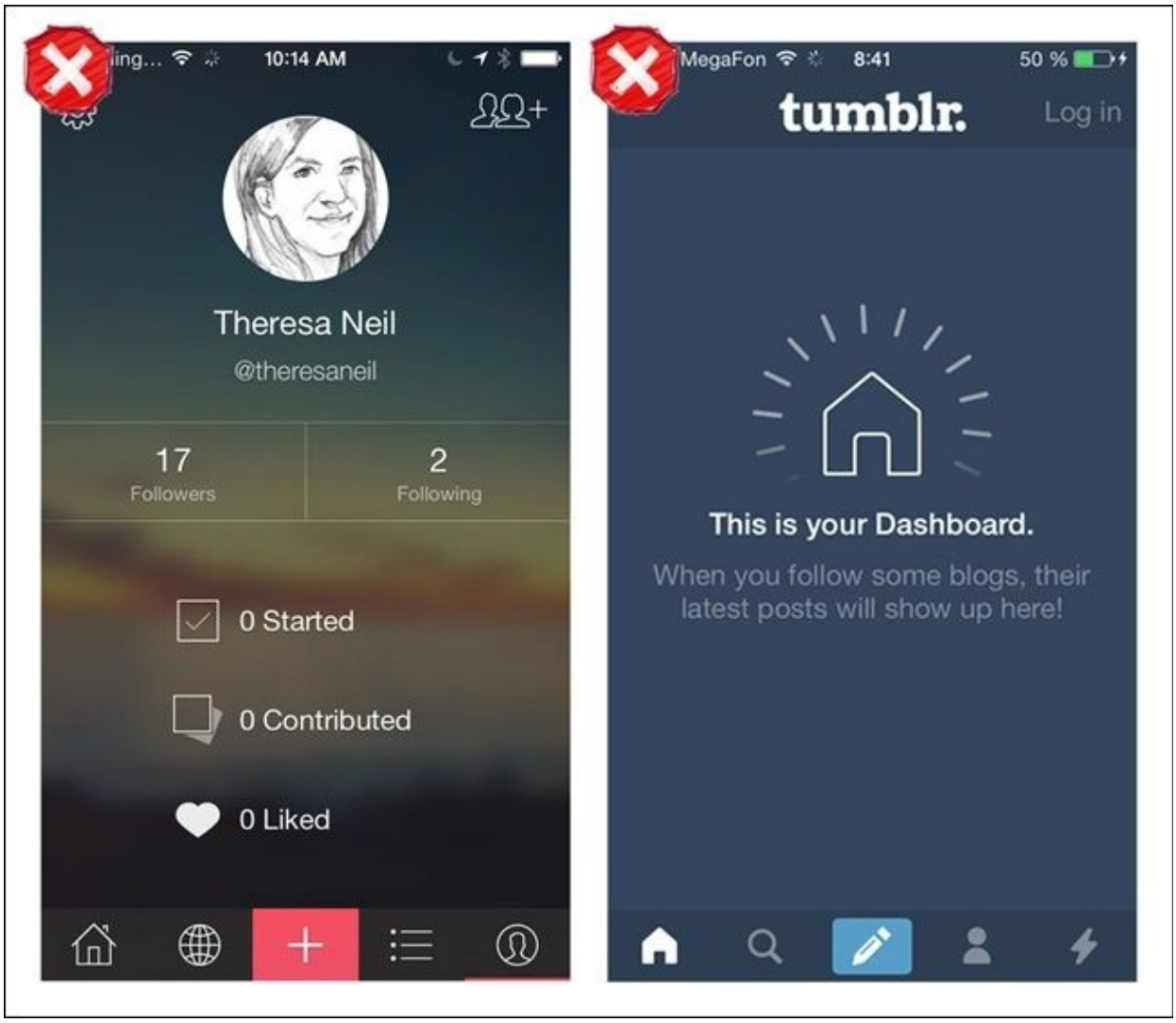



*Figure 1-29. Everlapse and Tumblr for iOS: the prominence of action buttons overshadows menu location*

## Android

Android offers three different Tab Menu patterns for top-level, or primary, navigation: Fixed Tabs, Spinners, and Navigation Drawers. Here are the Android guidelines (*http://developer.android.com/design/patterns/app-structure.html*) for Fixed Tabs:

> *Fixed tabs* display top-level views concurrently and make it easy to explore and switch between them. They are always visible on the screen, and can't be moved out of the way like scrollable tabs. *Fixed tabs* should always allow the user to navigate between the views by swiping left or right on the content area.
> Use tabs if:
>
> - You expect your app's users to switch views frequently.
> - You have a limited number of up to three top-level views.

- You want the user to be highly aware of the alternate views.

Path makes Fixed Tabs work by using icon-based tabs, while Quora pushes the limit, squeezing in four text-based tabs. More often than not, designers incorrectly use the Scrolling Tab control for primary navigation when they should be using a Spinner or Navigation Drawer instead.

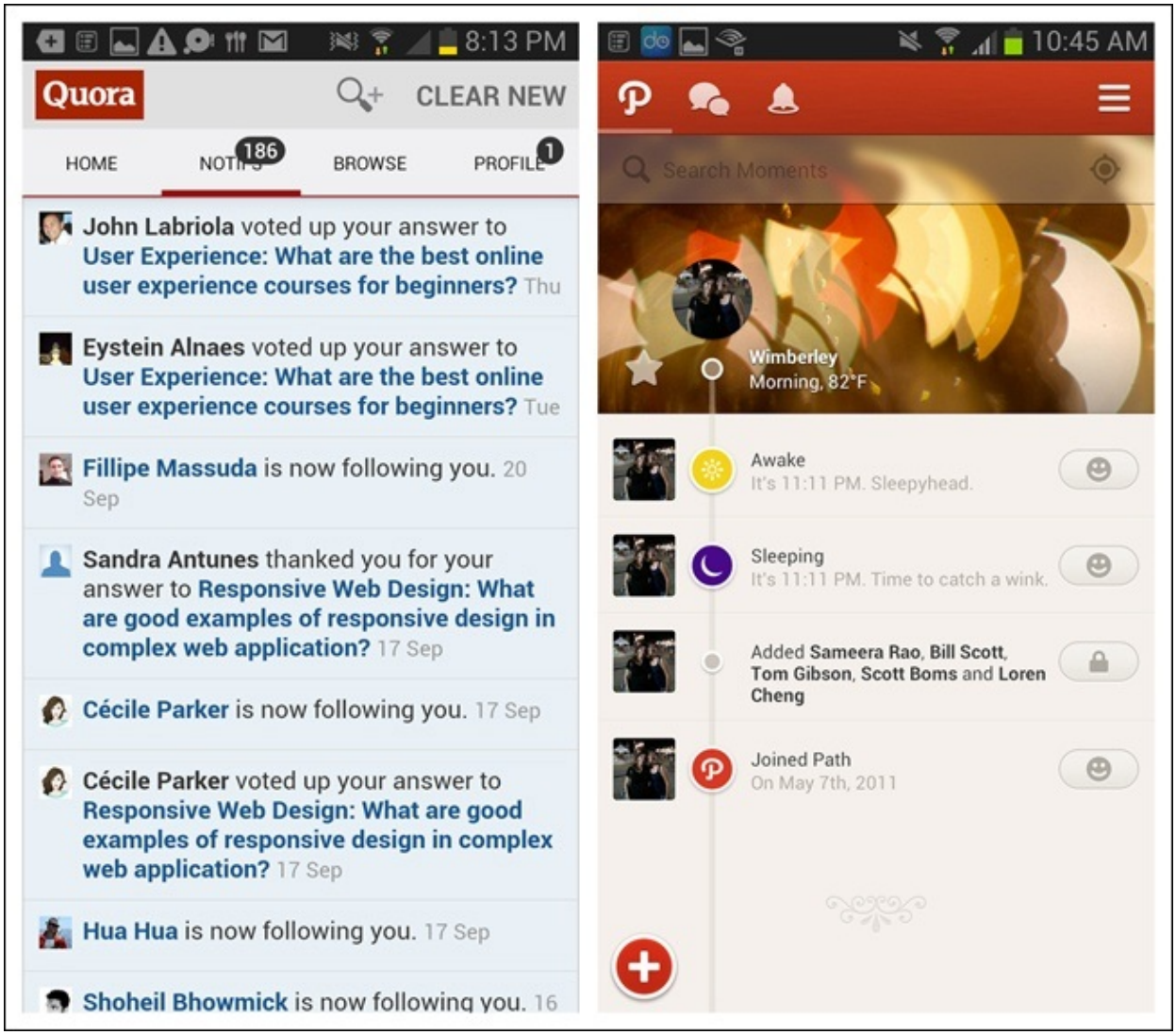



*Figure 1-30. Quora for Android pushes the Fixed Tabs limit by squeezing in four items; Path for Android uses icons for Fixed Tabs*

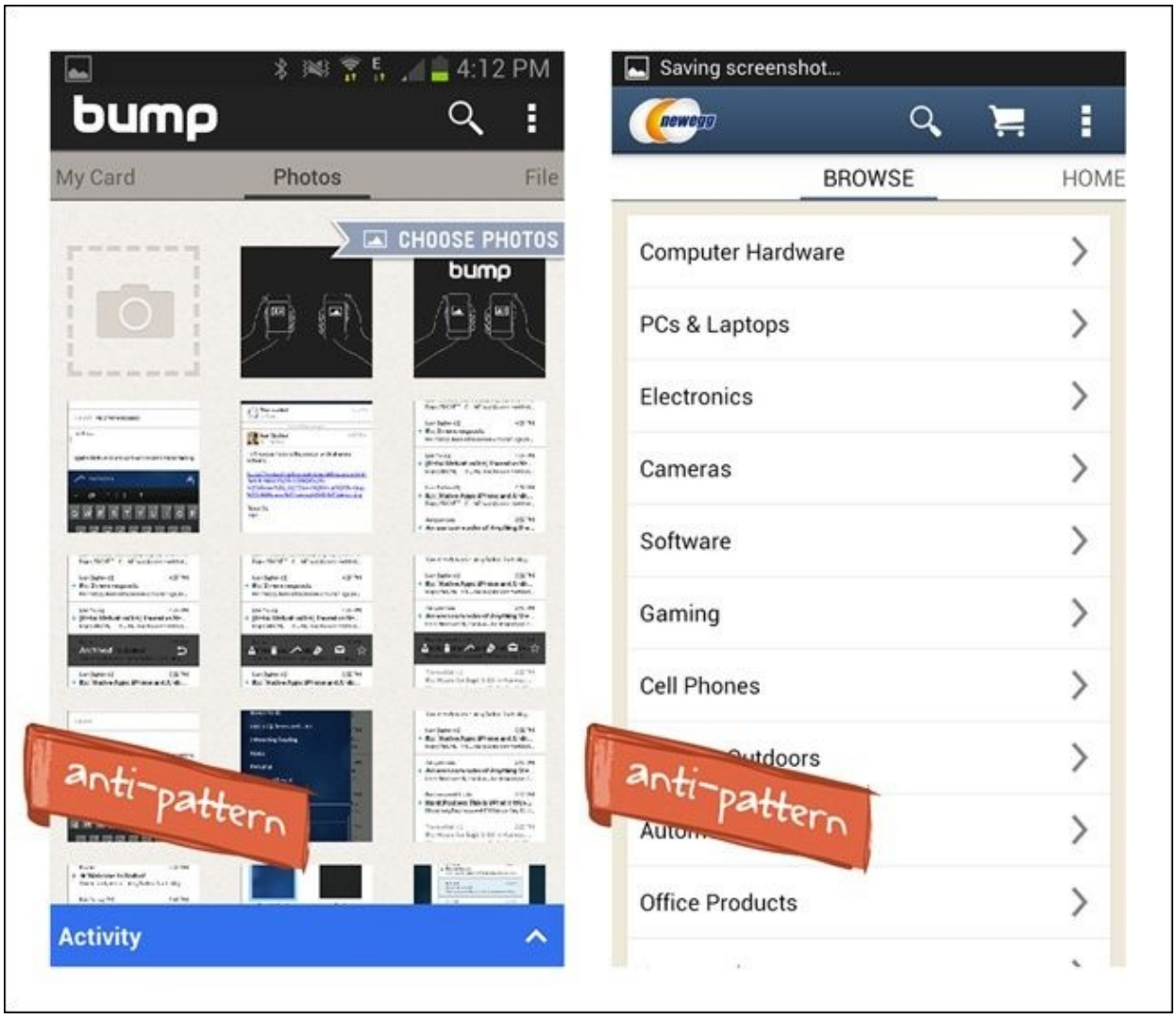



*Figure 1-31. Bump and Newegg for Android: incorrect use of Scrolling Tabs for primary navigation*

## Windows Phone

In Windows Phone, the Tab Menu is called App Tabs, and tabs that extend offscreen are accessible via panning, through the Pivot control. The Windows Phone design guide (*http://bit.ly/1iJWnpE*) suggests:

> You can use the Pivot control (*http://bit.ly/1iJWy4v*) to implement the App Tab UI style. This control allows the user to navigate right and left through each page (called a *pivot page*).


*Figure 1-32. Netflix for Windows Phone: Pivot control implementation of App Tabs*

## Emerging Patterns

There is a new trend in both desktop and mobile web design to hide or collapse the website header when the user is scrolling or swiping down through content.

This Retracting Tab design is showing up in native apps too. Pinterest retracts the Toolbar when the user swipes down to browse content. The Toolbar reappears when the user swipes up. Luvocracy and Polar for iOS are other apps that use this effect.

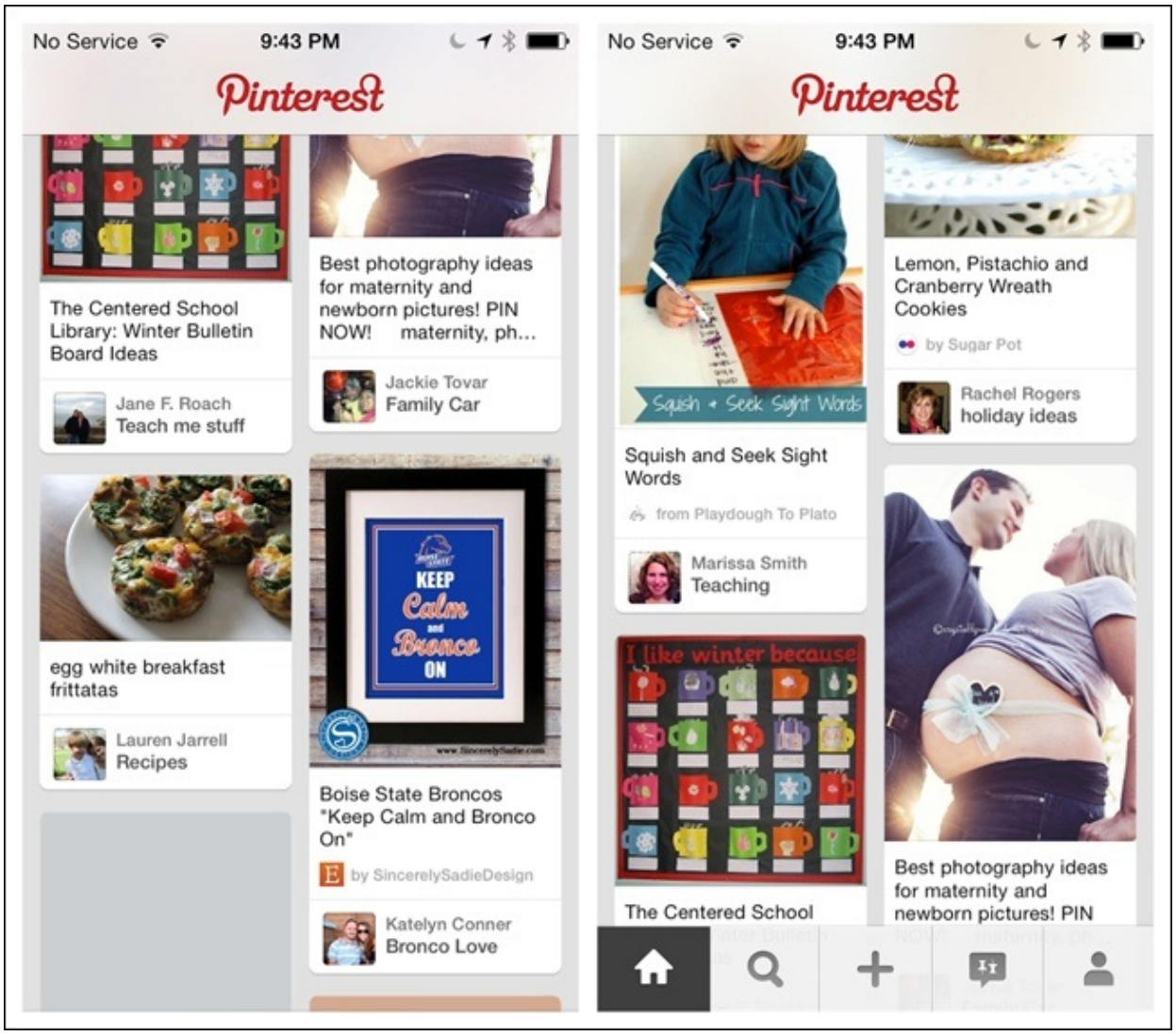



*Figure 1-33. Pinterest for iOS: scrolling down hides the Toolbar; scrolling up reveals it (http://www.youtube.com/watch?v=joaaJgvTN28)*

Configurable Tabs are another variation of the standard Tab Menu. The design of Frequency mimics the way tabs are implemented in all the major desktop web browsers. Adding a channel adds a new tab. If there are too many tabs to fit on the screen, the header scrolls. To reorganize the tab order, the user can simply press and hold a tab, then drag it to the desired location.

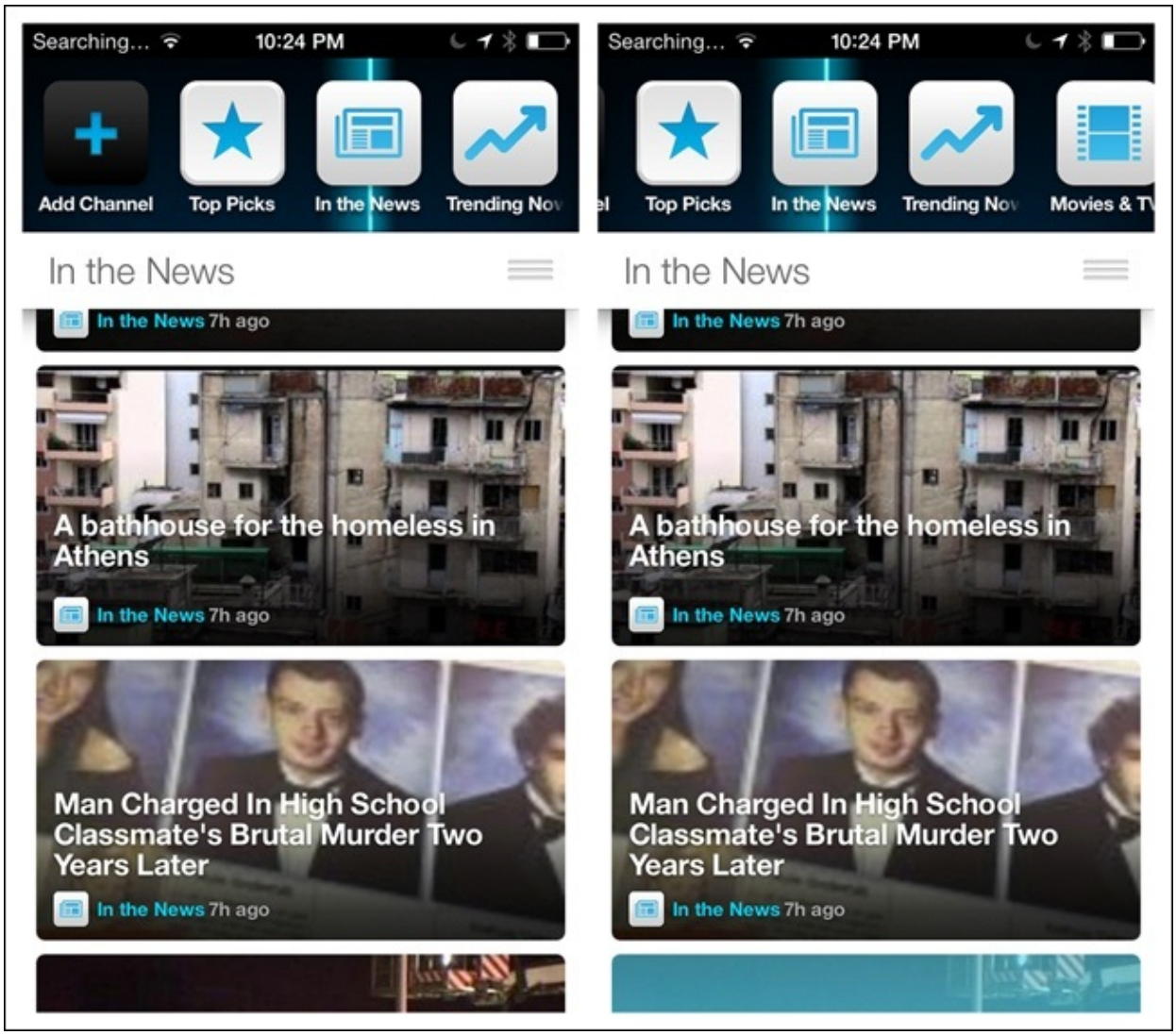



*Figure 1-34. Frequency for iOS: Configurable Tabs, an emerging pattern, work like web browser tabs*

Sidebars are gaining popularity in web applications as well as native tablet apps. Twitter offers clearly labeled side tabs for navigating the main views of its iPad application. Yammer has almost twice as many tabs, but no labels, making navigation more of a challenge.

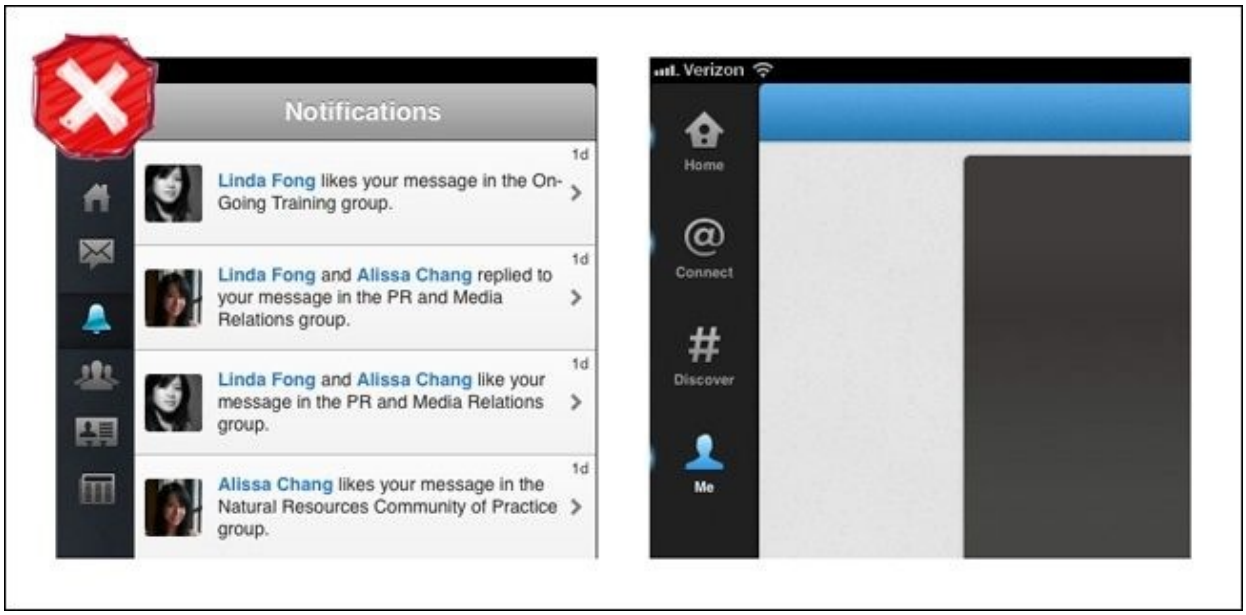



*Figure 1-35. Twitter for iPad has tabs clearly labeled; Yammer for iPad does not*

It is unlikely that Sidebars will ever be widely adopted as a persistent navigation pattern on smartphones, for two reasons:

- Most people hold their smartphones in portrait mode, and the sidebar takes up a fair amount of horizontal real estate.
- Since the space is so limited, labels will get dropped, which reduces the usability of the app.

Dwell is a cautionary example. It looks nice, but you have to tap every single icon to see what the primary categories are — a classic example of mystery meat navigation (*http://www.webpagesthatsuck.com/mysterymeatnavigation.html*).

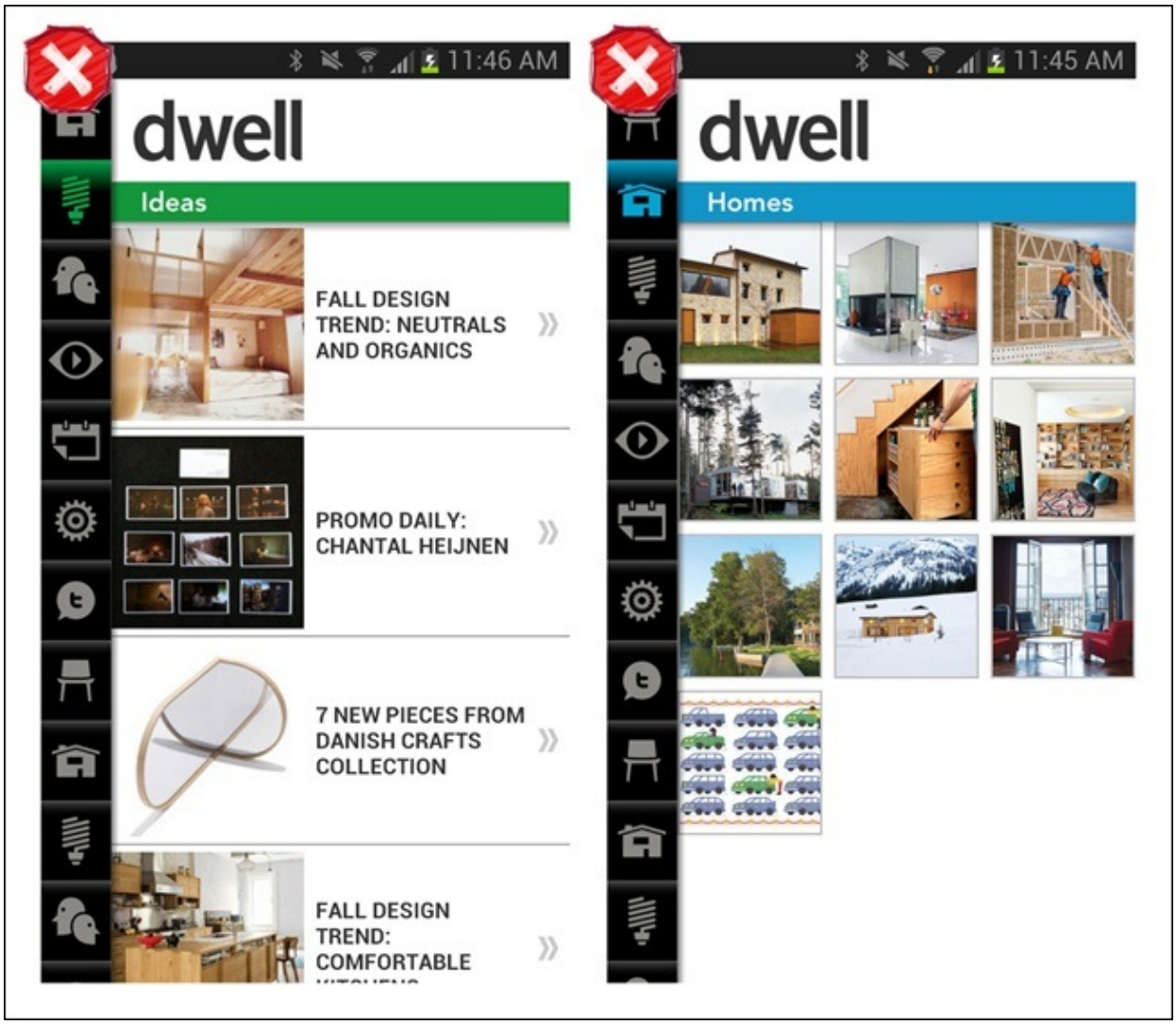



*Figure 1-36. Dwell for Android: the Sidebar as mystery meat navigation*

**NOTE**

Learn and follow the different OS design guidelines for Tab Menus. Clearly indicate where the user is by visually differentiating the selected tab from the others.

## Skeuomorphic

The Skeuomorphic pattern is characterized by an interface designed to match its real-world counterpart. Even though the trend in *visual design* is now toward a flatter design aesthetic, some apps can still aid usability by emulating objects and tools from the real world.

Skeuomorphism is the dominant navigation pattern in game design, as in *Sniper Ghost Warrior* 2, but it can also have its uses in nongame apps, like the music-

mixing app Cross DJ, the photo app Hipstamatic, and the travel app FlightBoard.

*Figure 1-37. Sniper Ghost Warrior 2 for iOS and Android: skeuomorphism is common in game navigation*


*Figure 1-38. Cross DJ for iOS: skeuomorphic design is also valid for some nongame apps*


*Figure 1-39. Hipstamatic for iOS: emulation of real-world objects*

Other examples include the iOS 7 redesign of Awesome Note and Apple's Newsstand. Both designs are immediately recognizable and feel intuitive because we recognize the folder and bookshelf metaphors.

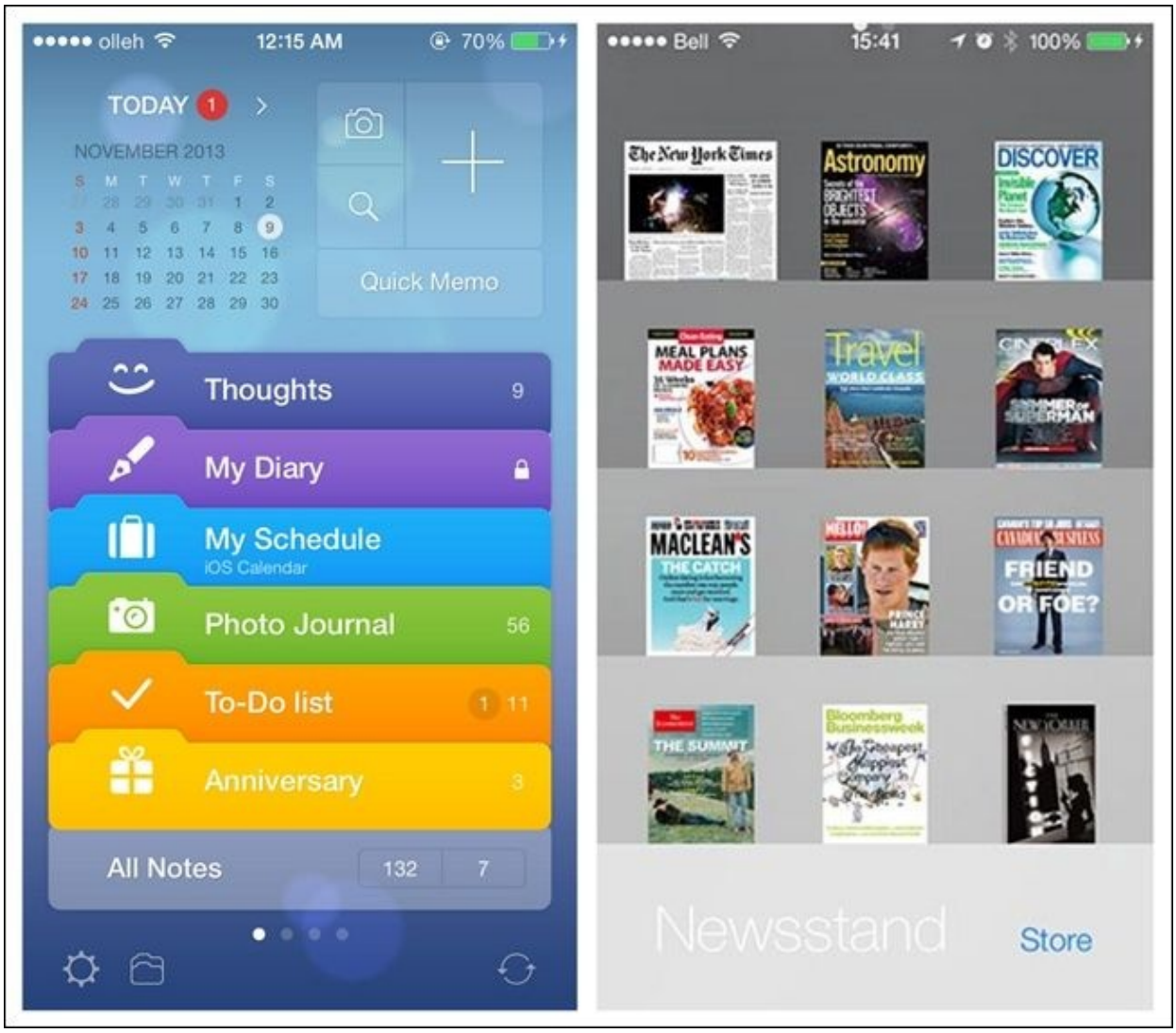



*Figure 1-40. Awesome Note and Newsstand for iOS: skeuomorphism can help make navigation intuitive*

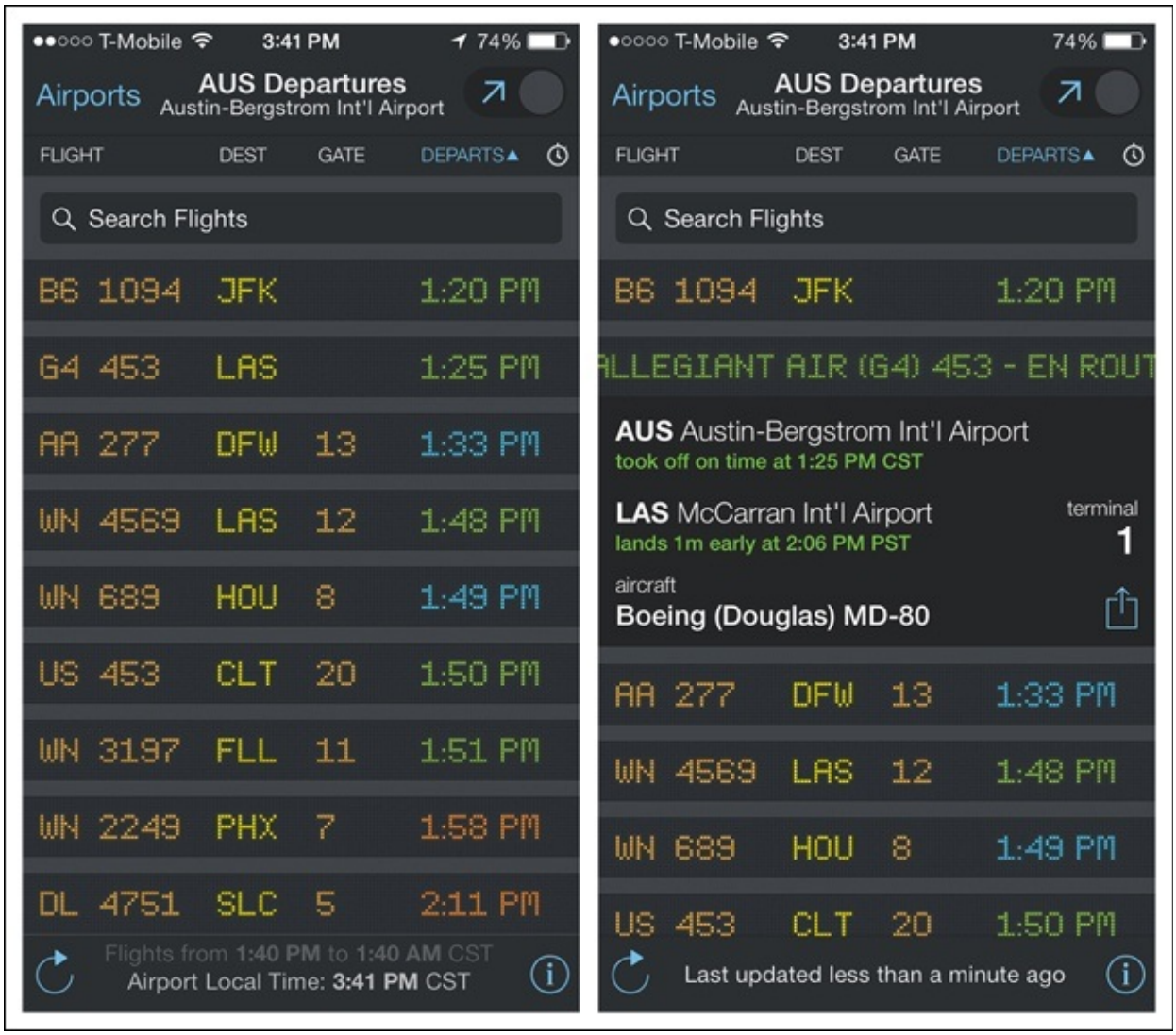



*Figure 1-41. FlightBoard for iOS: designed to match the flight information display systems in airports*

There are, of course, limits to any metaphor being extended to the digital realm. For example, there was confusion around the early versions of iBook: some users thought that the digital bookshelf size implied a limit to the number of ebooks it could hold, which it didn't.

**NOTE**

Exercise restraint in choosing the metaphor to model the navigation on — a poor implementation can result in the Novel Notion anti-pattern discussed in Chapter 11.

# Primary Navigation Patterns, Transient

The term *transient* means staying a short time, which is exactly how the following navigation menus work. They are hidden until we reveal them; then we make a selection and they disappear again. The three patterns we'll look at here are Side Drawers, Toggle Menus, and Pie Menus.

## Side Drawer

There are two styles of Side Drawers. The first is an *overlay*, meaning a swipe or tap gesture will reveal a drawer that partially covers or overlaps the original screen content, as in RetailMeNot. The second style is an *inlay*, in which a swipe, pan, or tap will open a drawer that pushes the original screen content partially off-canvas, as in Path.

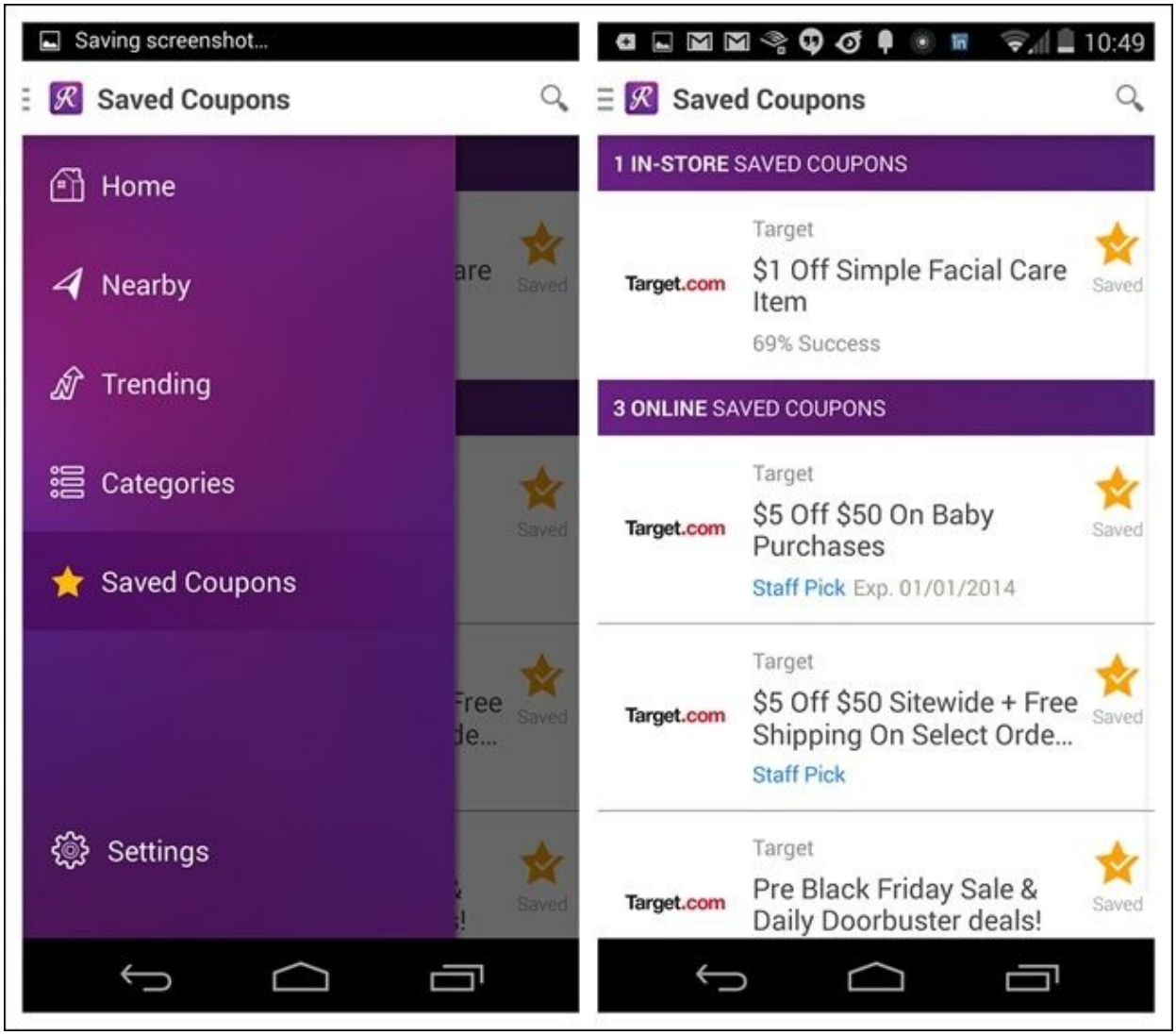


*Figure 1-42. RetailMeNot for Android: tap navicon or swipe edge to reveal the overlay Side Drawer, which partially covers main screen content*

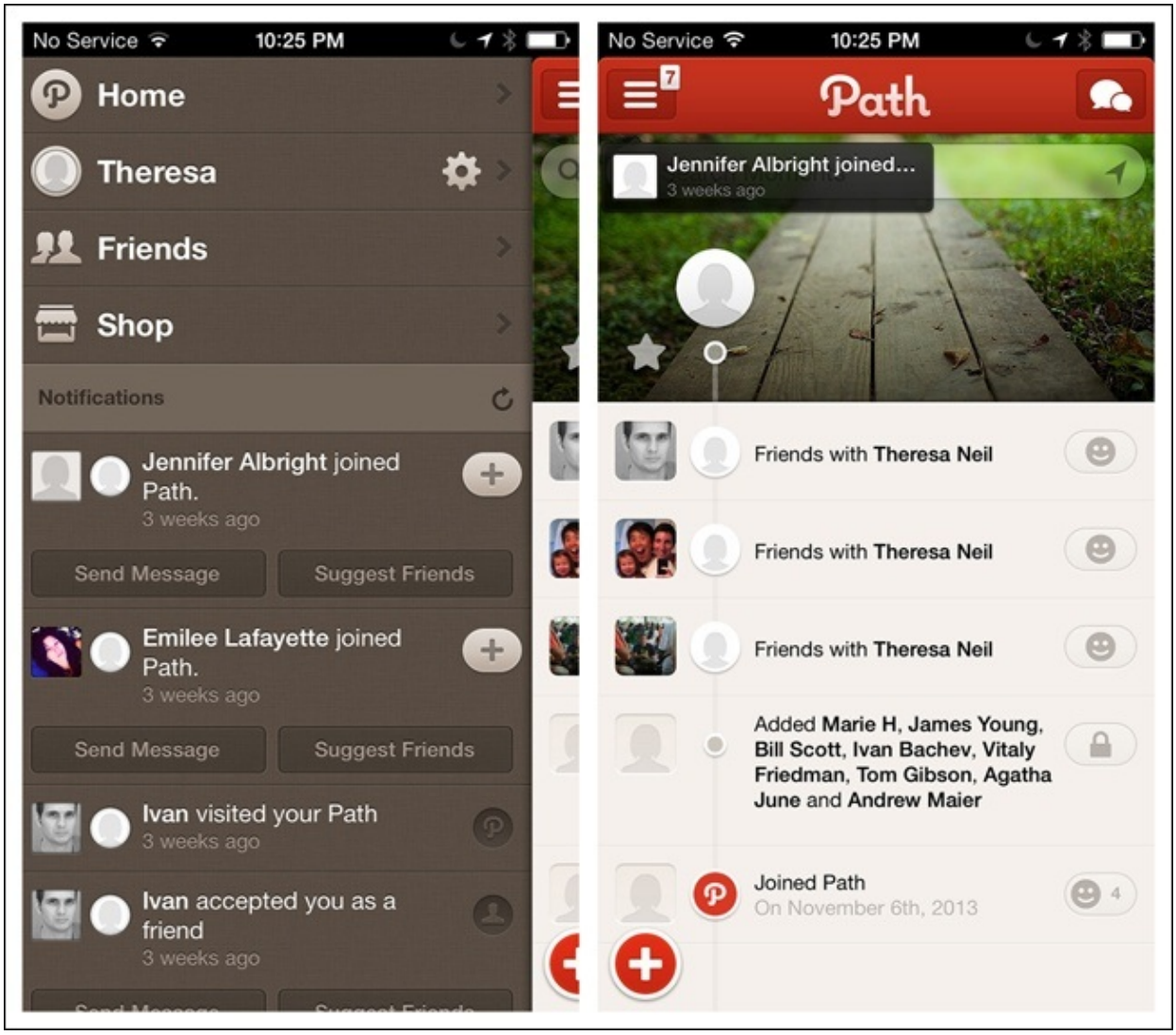



*Figure 1-43. Path for iOS: tap navicon or pan to reveal the Side Drawer as an inlay, pushing main screen to the side*

What is the best way to let users know that there is a Side Drawer?

The Android design guidelines (*http://bit.ly/1iKuQV5*) recommend having the drawer open on first use so the user can see the menu and learn how to close the drawer:

> Upon first launch of your app, introduce the user to the navigation drawer by automatically opening it. This ensures that users know about the navigation drawer and prompts them to learn about the structure of your app by exploring its content. Continue showing the drawer upon subsequent launches until the user actively expands the navigation drawer manually. Once you know that the user understands how to open the drawer, launch the app with the navigation drawer closed.

Sounds good, right? However, this suggestion has not performed well in the user testing I've conducted for clients. Instead, I recommend a design like Allthecooks, where the drawer "bumps" open just the first time the app is opened.

The most popular orientation for the Side Drawer is on the left, but it can be on the right instead, as with IfThisThenThat, or there can be drawers on the right and the left, as with the Facebook Beta for Windows Phone.

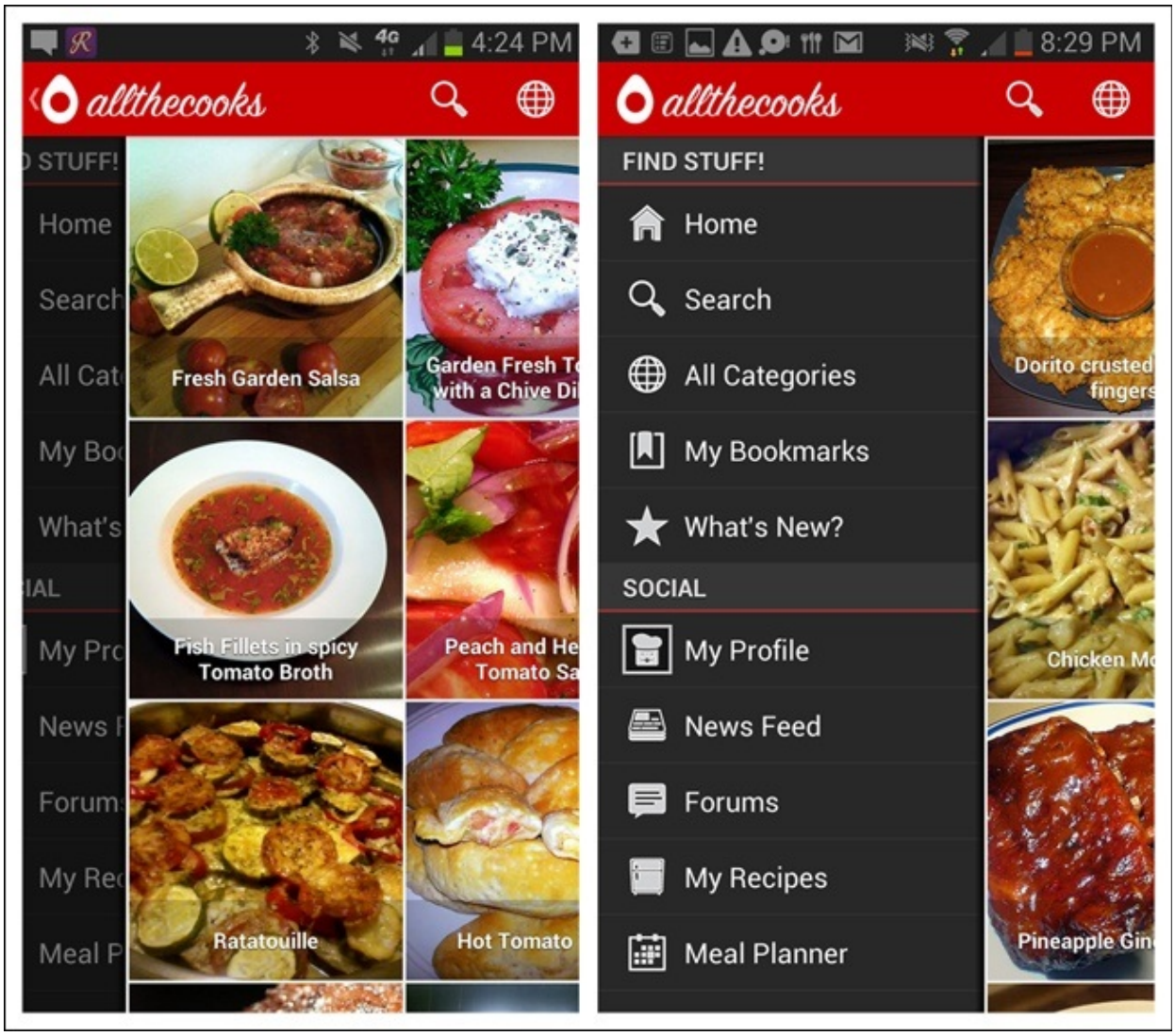



*Figure 1-44. Allthecooks for Android: Side Drawer bumps open when app is launched to alert users it's there (http://youtu.be/iLlCs-gyNaE)*

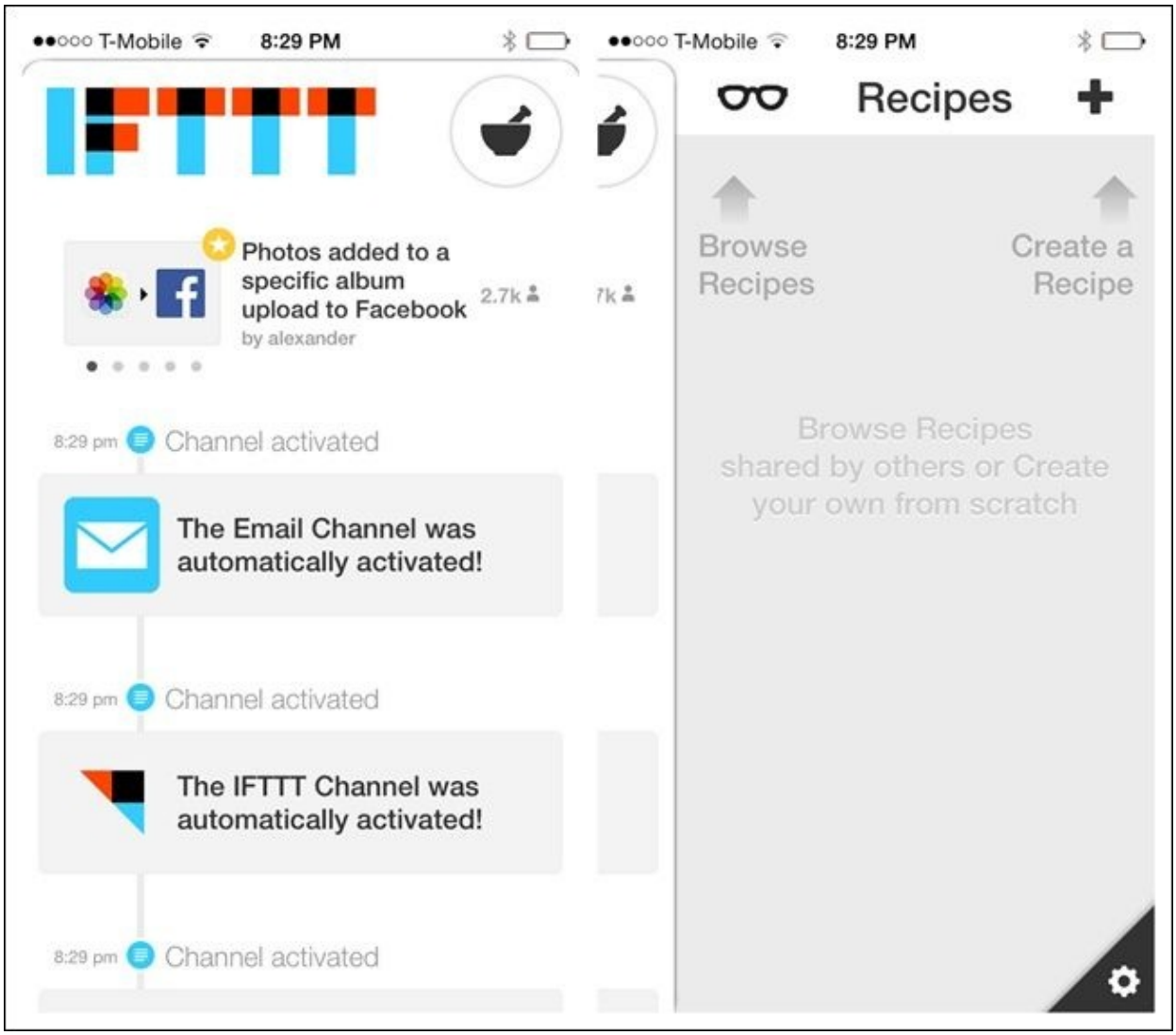



*Figure 1-45. IfThisThenThat for iOS: Side Drawer for configuration and navigation is at right*


*Figure 1-46. Facebook beta for Windows Phone: two Side Drawers — left for the main menu, right for quick links*

But don’t position the drawer on the bottom, as in the om finder and Frost apps. This positioning conflicts with the swipe-up gesture that reveals the Control Center in iOS 7.

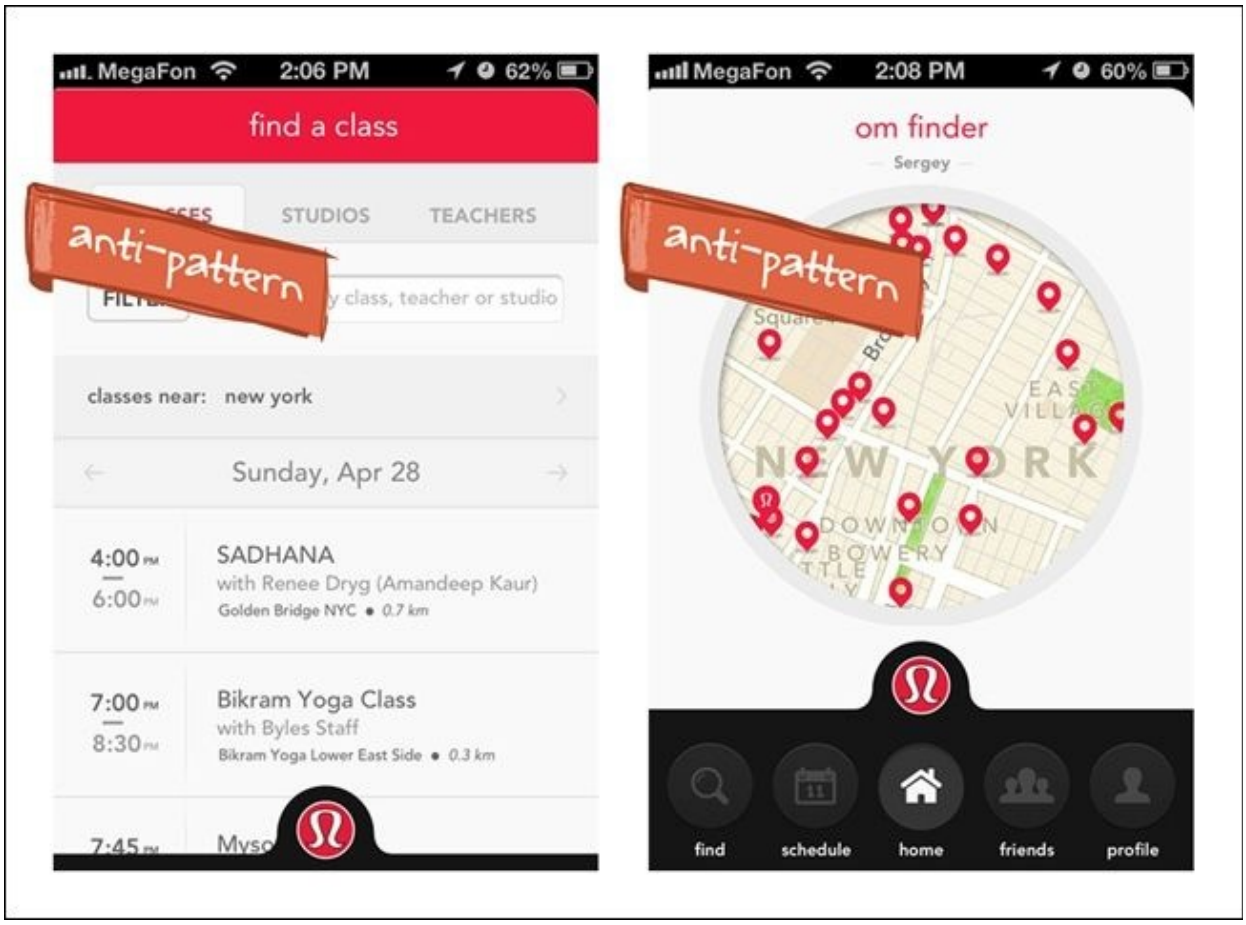



*Figure 1-47. om finder for iOS: drawer at the bottom of the screen conflicts with the iOS 7 Control Center*

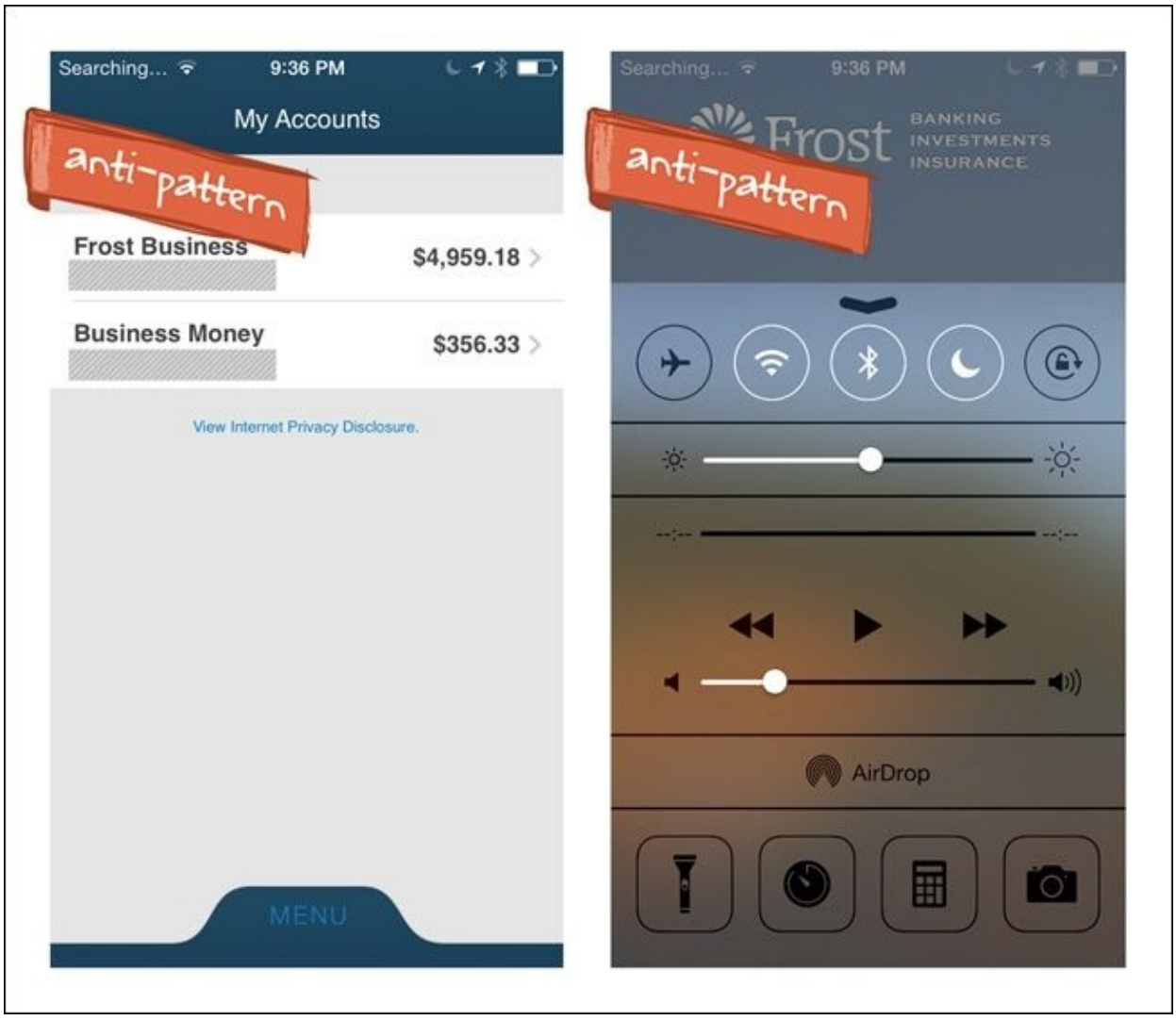



*Figure 1-48. Frost for iOS: almost every time I try to open the menu, the iOS Control Center opens instead*

Side Drawer content need not be limited to only navigation options. Zillow's Mortgage Marketplace drawer has a real-time chart of mortgage rates, and social apps like LinkedIn frequently include profile information.

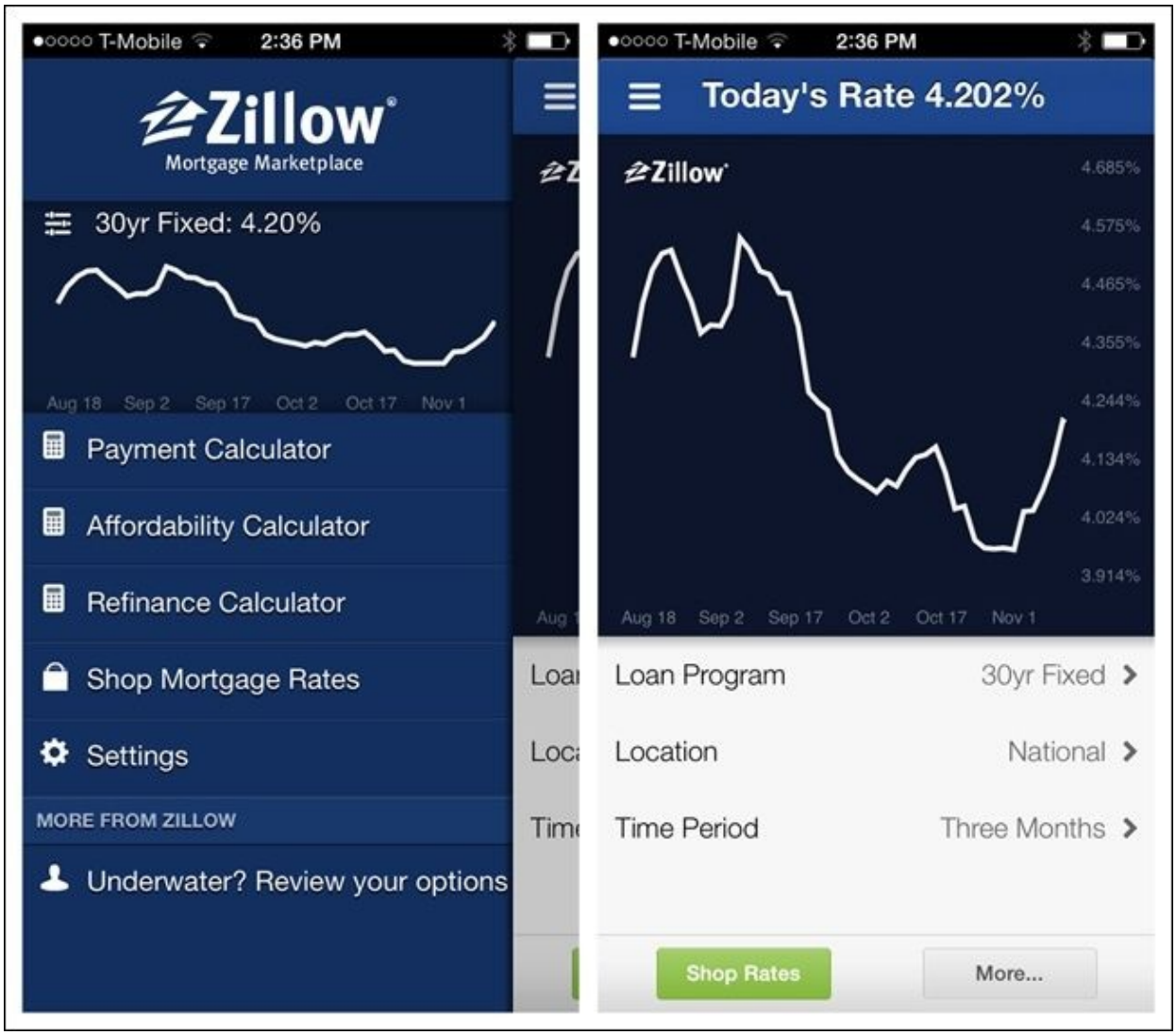



*Figure 1-49. Zillow Mortgage Marketplace for iOS: Side Drawer has content and menu items*

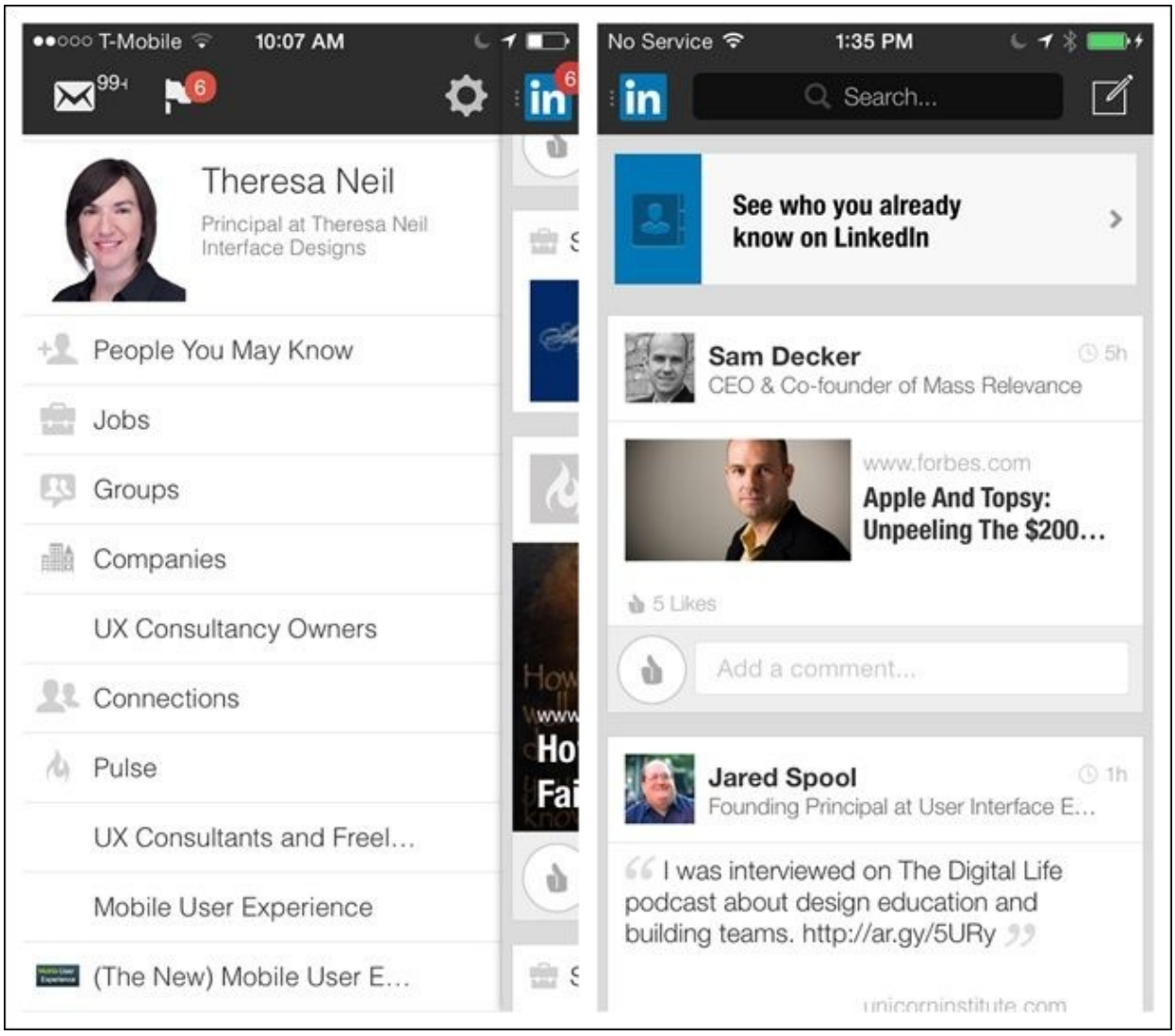



*Figure 1-50. LinkedIn for iOS: Side Drawer has profile info (oops, I have mail!)*

The Side Drawer can be more than one level deep. In Fancy, for instance, you can tap-tap-tap down the path until you reach the lowest-level category. As you drill down through the categories, the content on the right updates. In Wish, the Side Drawer path for Categories is only two levels deep; categories are selected Springboard-style.

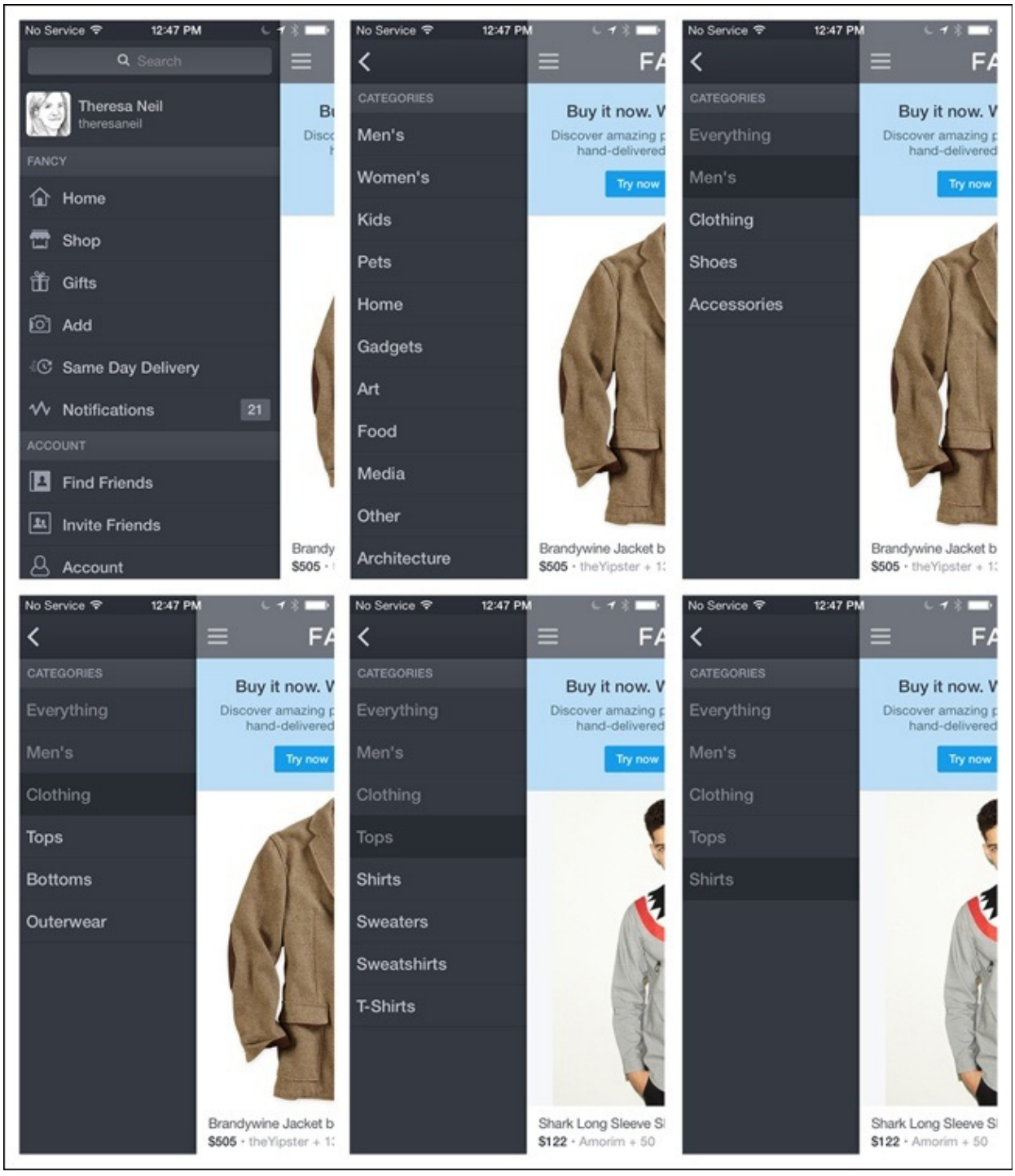



*Figure 1-51. Fancy for iOS: a multilevel Side Drawer*

The Side Drawer can also let users switch high-level context. With the Side Drawer in Gmail for iOS, tapping the arrow by my name opens a panel that slides down over the menu options. There I can switch between email accounts, or add a new one.

**NOTE**

The Side Drawer can be versatile, but be careful not to overload it with too many features. It should show the primary navigation options first and foremost.

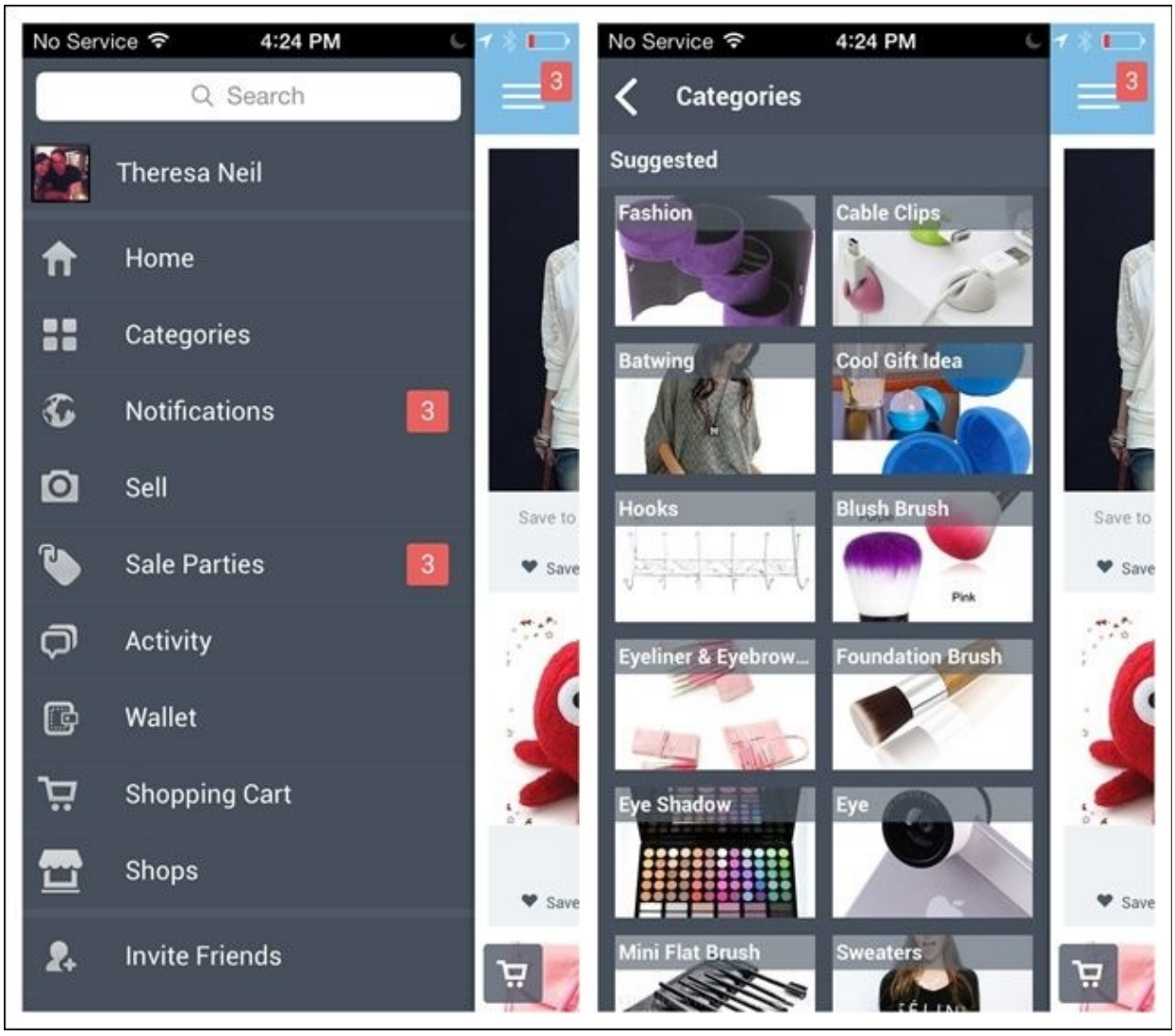



*Figure 1-52. Wish for iOS: Side Drawer path to Categories is only two levels deep, then switches to a Springboard*

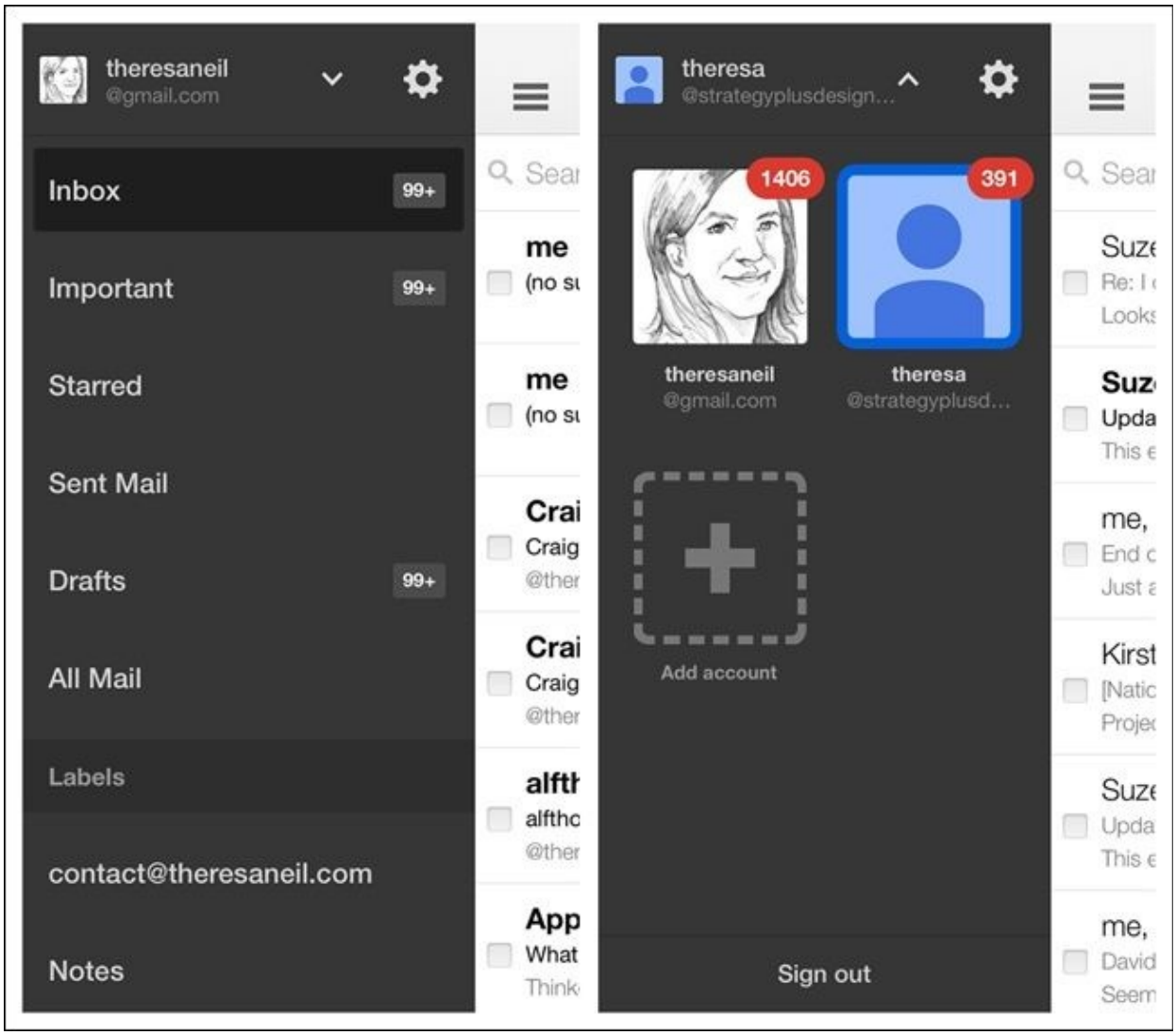



*Figure 1-53. Gmail for iOS: slide-down panel within a Side Drawer lets me switch or add accounts*

## Emerging Pattern

In response to iOS 7 guidelines, the designers of Luvocracy have been experimenting with a variation of the inlay Side Drawer (*http://uxmag.com/articles/adapting-ui-to-ios-7-the-side-menu*).

Tapping the navicon (or the “hamburger”) reveals the Side Drawer, but instead of the drawer inlay simply pushing the parent screen to the right, it also uses a 3D effect to push it back.

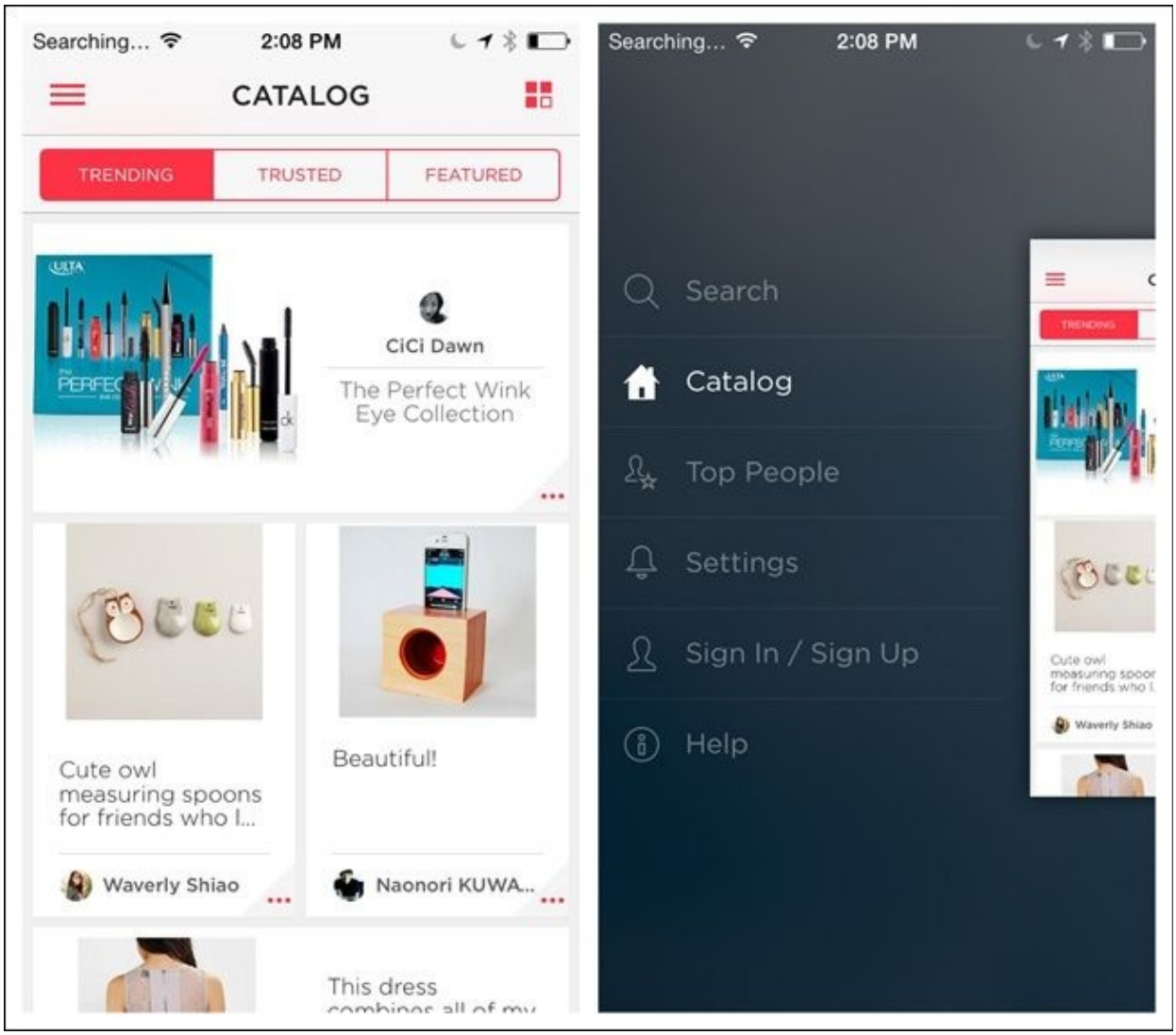



*Figure 1-54. Luvocracy for iOS: tap the navicon, or "hamburger," and Side Drawer pushes parent screen aside and back*

Luvocracy is simple to navigate and the transitions are smooth. However, a similar-style menu in Airbnb for iOS constitutes an anti-pattern. The new design creates three high-level menu categories: Travel, Host, and Log In. The Travel title is positioned across the top (with the actual travel menu options disconnected down below), while Host and Log In are next to each other on the bottom edge. Tapping on Host or swiping vertically switches Host to the top and Travel to the bottom (Log In stays static). Swiping again switches them back. This design is impractical and inefficient for users. I don't always agree with everything in the iOS Design Guide (*http://bit.ly/1iK1juP*), but substitute the word *swipe* for *scroll* in the following and it's on point in this case:

> Don't make users scroll to see all their choices. This causes a disconcerting experience for users,

because they must spend extra time to distinguish the choices. Also, it can be very difficult for users to scroll without inadvertently tapping an option.

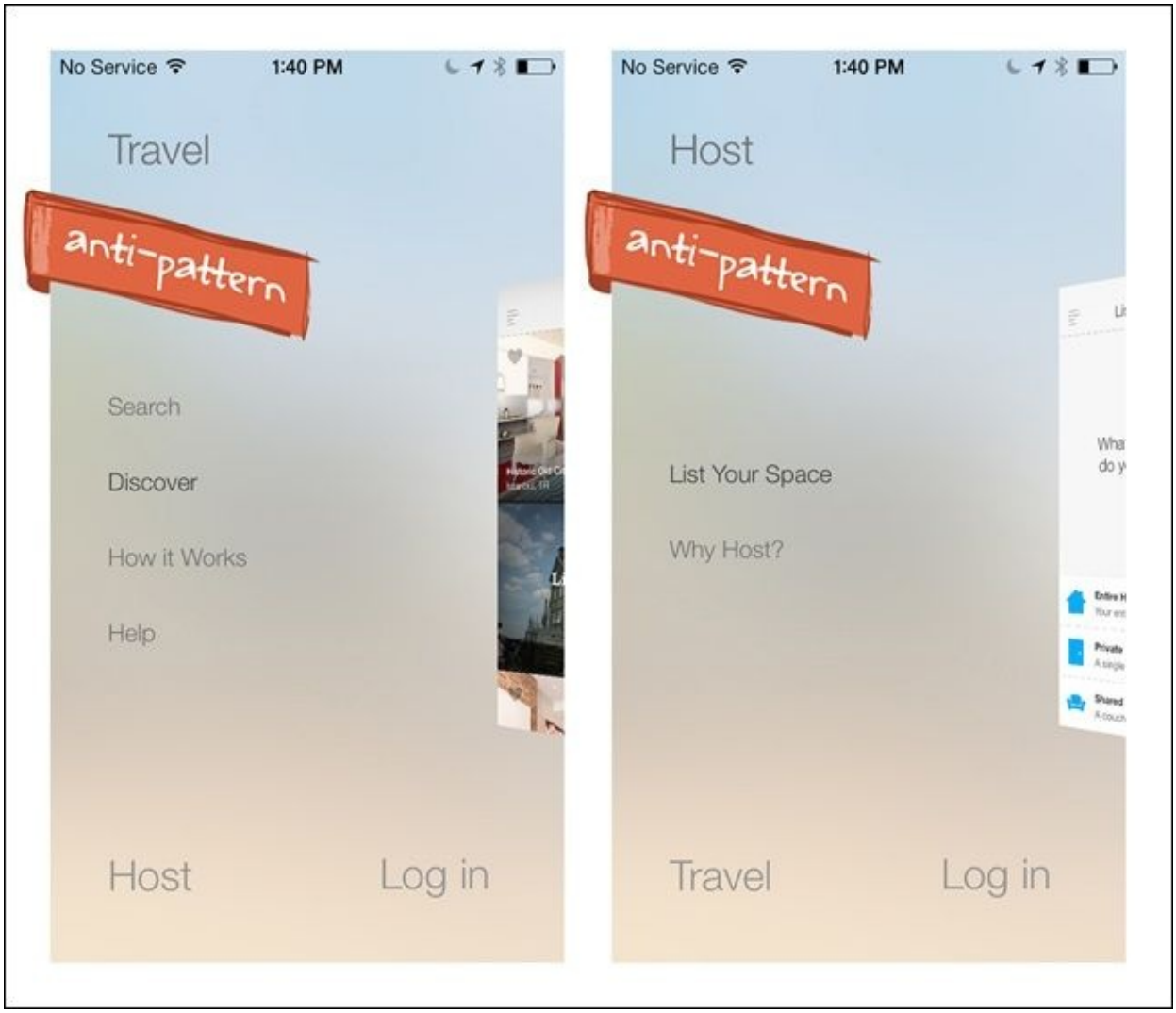



*Figure 1-55. Airbnb for iOS 7: confusing implementation of the emerging Side Drawer style (http://www.youtube.com/watch?v=Rl1ZwXINhIc)*

The Navigation Drawer in the Android version of Airbnb, by comparison, is crystal-clear — no guesswork required.

A better implementation for iOS is American Airlines, where the Side Drawer reveals a well-designed grouped menu. I wish the parent screen weren't translucent, though, since it is still a touch target.

**NOTE**

Before you choose Side Drawer navigation, map out the information architecture for the app and validate it with users. Then, if a Side Drawer seems to make sense, you should prototype and test a couple of variations to see which one works best.

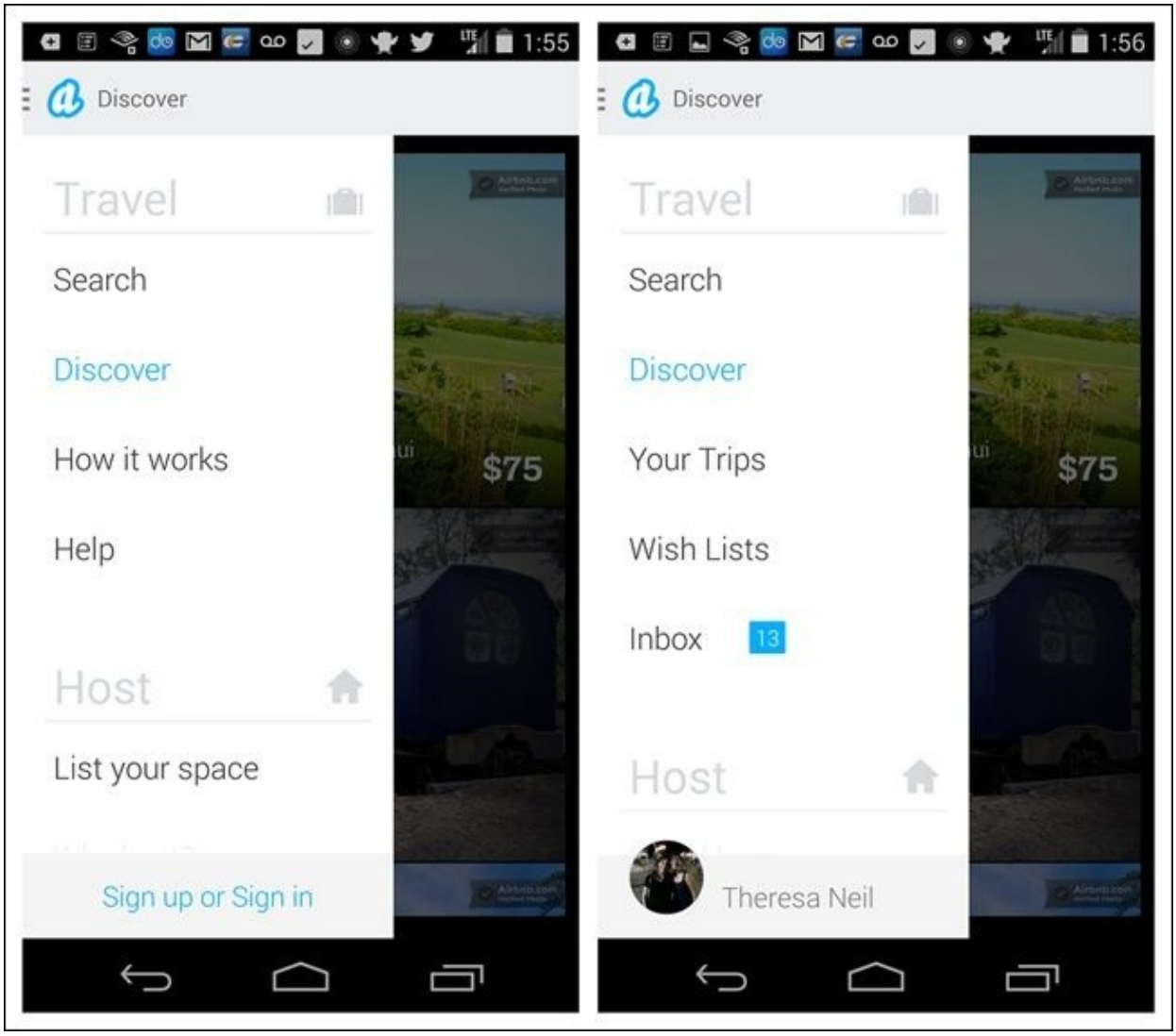



*Figure 1-56. Airbnb for Android: ah, crystal-clear Side Drawer navigation*

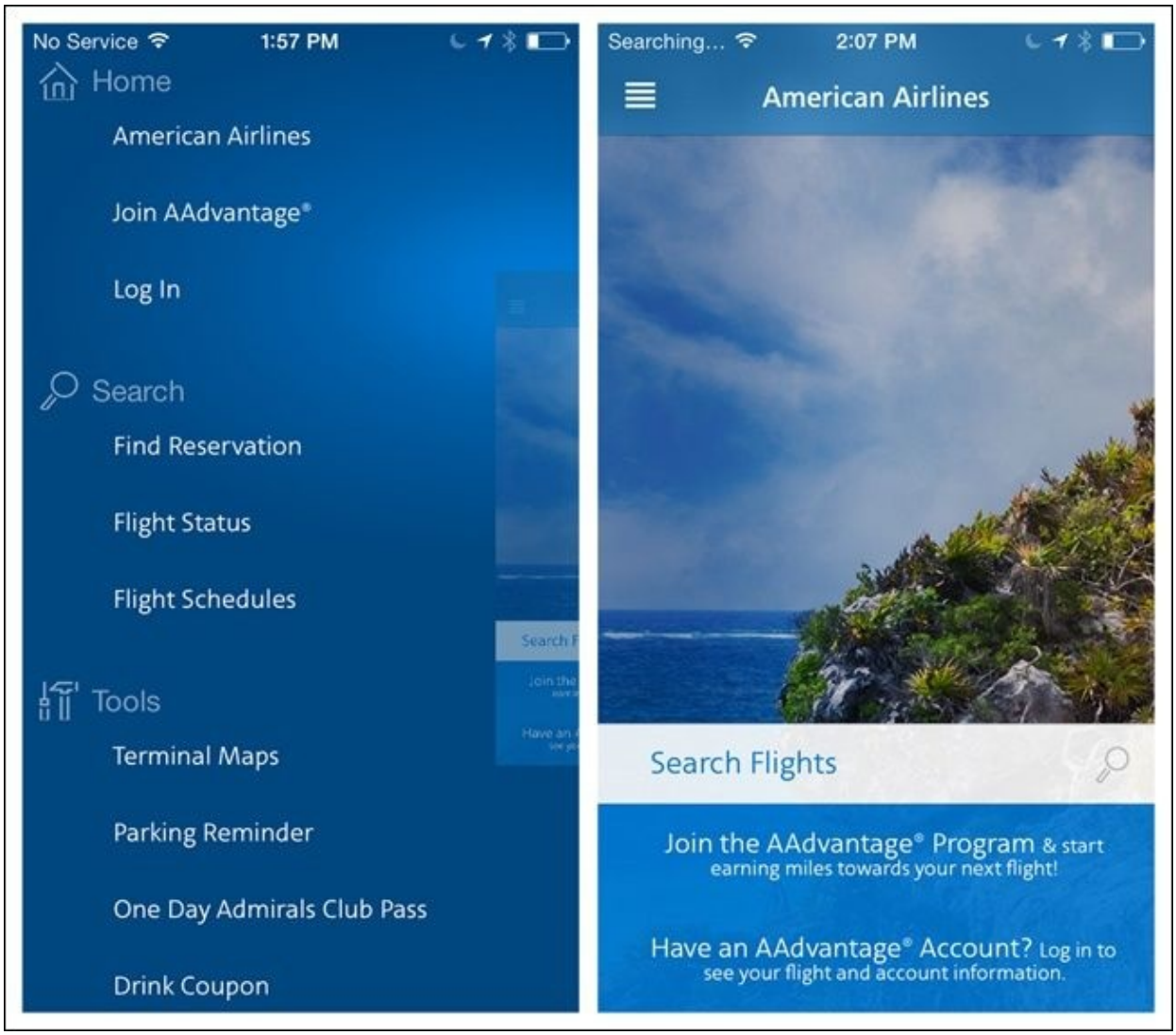



*Figure 1-57. American Airlines for iOS: the grouped menu for Side Drawer is nice, but the parent screen should be solid, not translucent*

## Toggle Menu

In this book's first edition, I labeled this pattern the Mega Menu, after its web equivalent. Since then, mobile web and responsive web design have pushed this design further, and it is now more commonly known as the Toggle Menu.

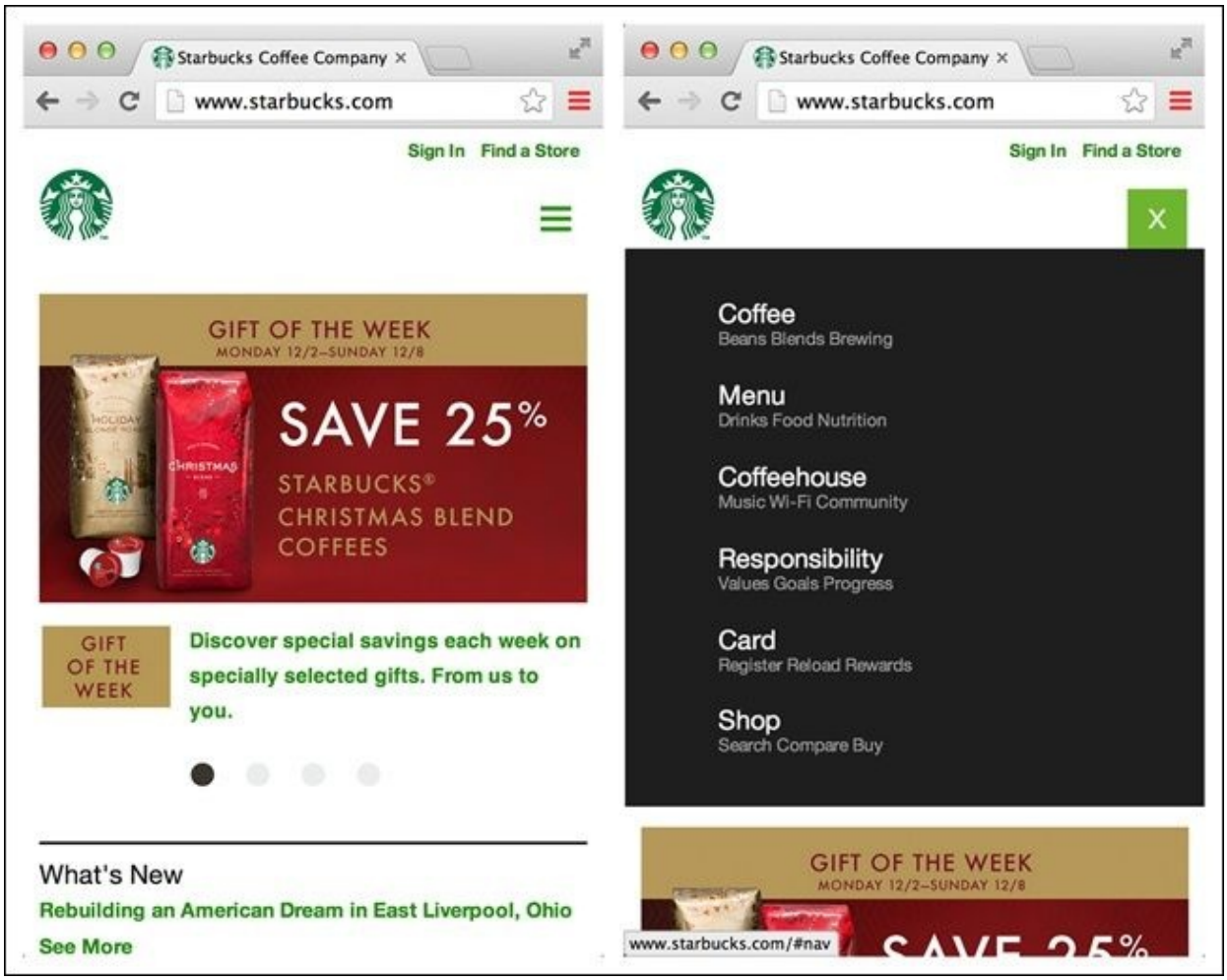



*Figure 1-58. Starbucks.com: responsive web design with a Toggle Menu*

Like the Side Drawer, the Toggle Menu can be an inlay that pushes the content down below the menu, as with Pocket or Qwiki, or an overlay that appears as a layer above the content, as with Walmart and Home Depot. The overlay design is the more common option in native mobile apps. In Ultravisual, the overlay Toggle Menu comes up from the bottom.

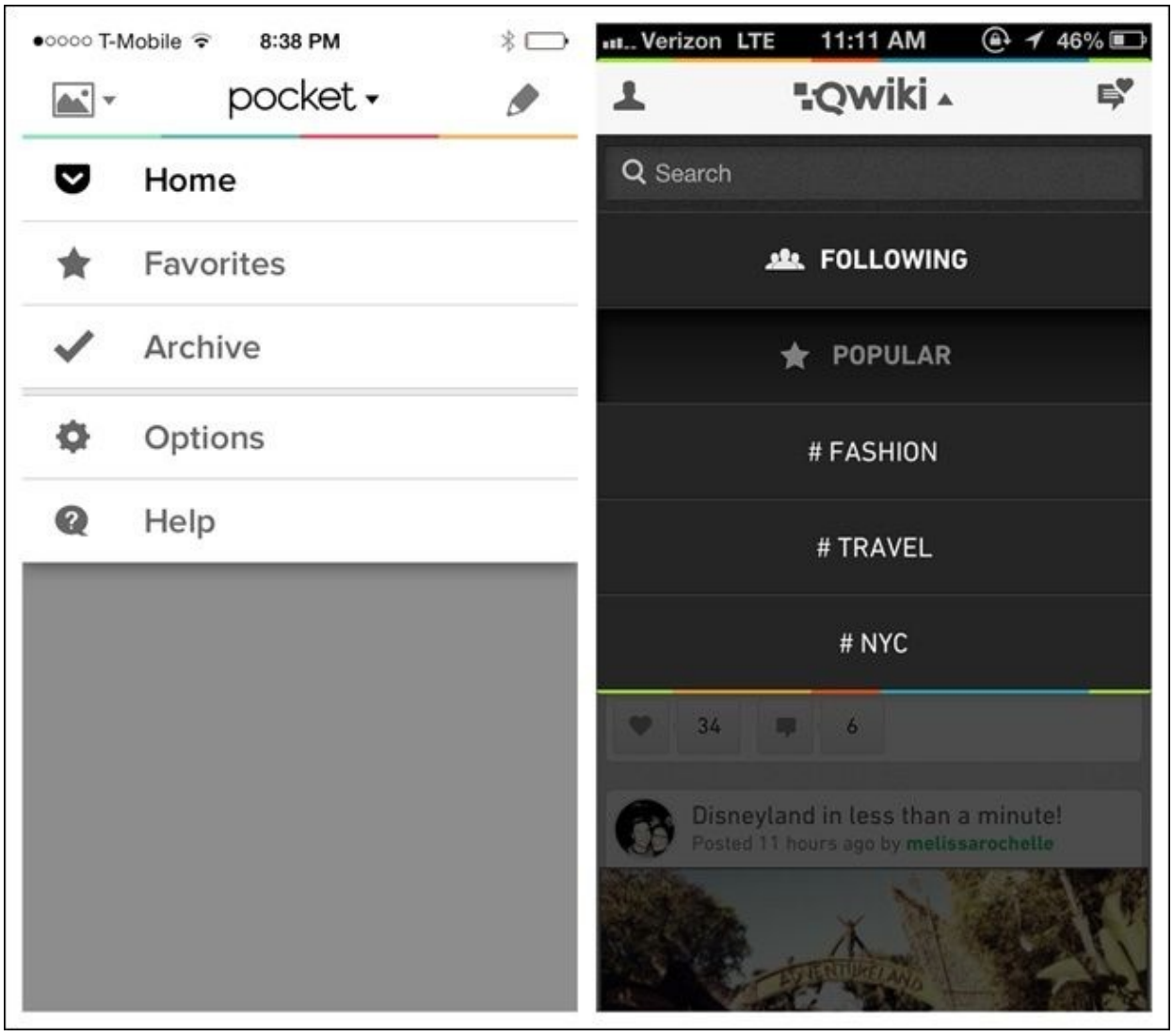



*Figure 1-59. Pocket and Qwiki for iOS: overlay Toggle Menus*

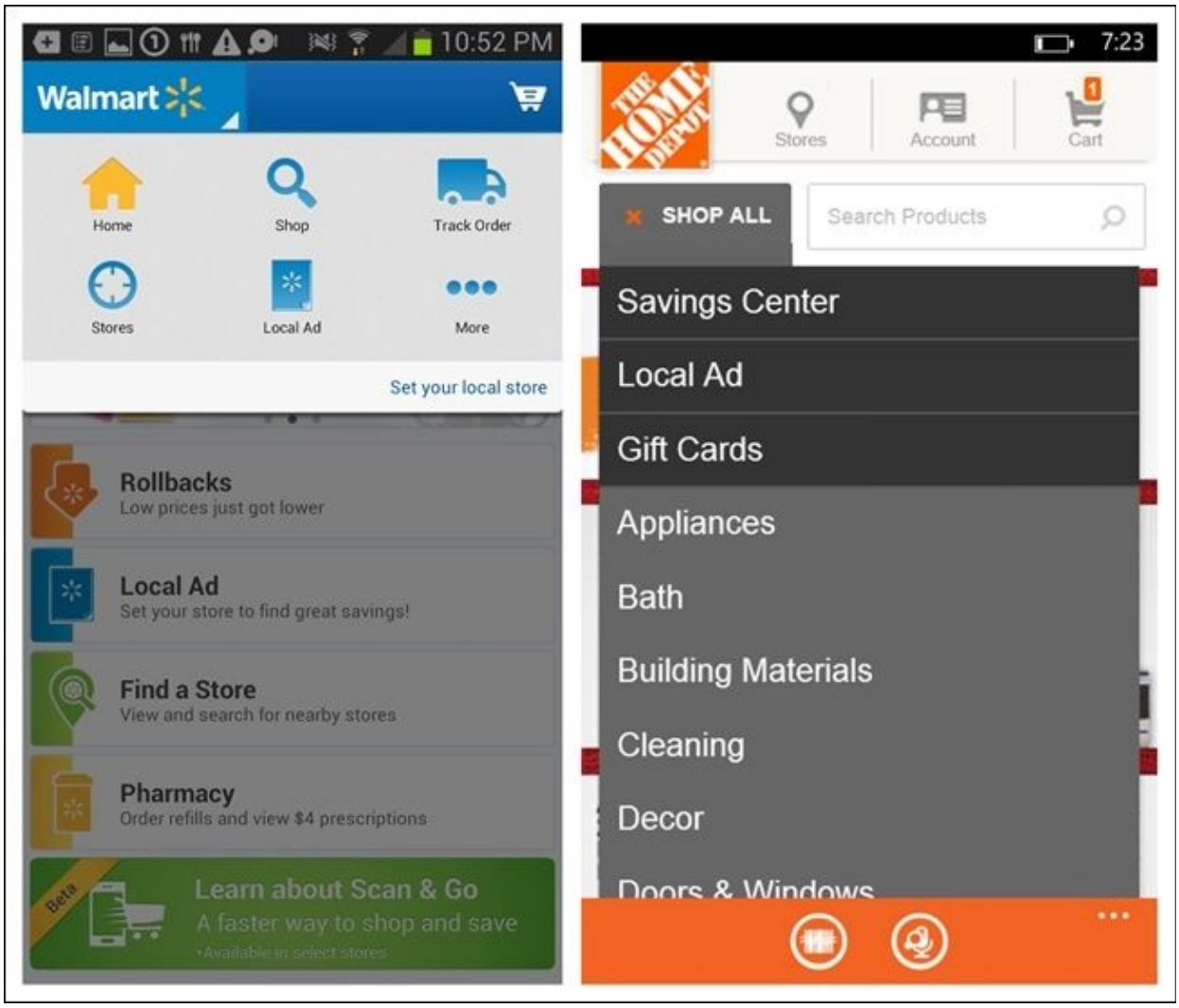



*Figure 1-60. Walmart for Android and Home Depot for Windows Phone: overlay Toggle Menus*

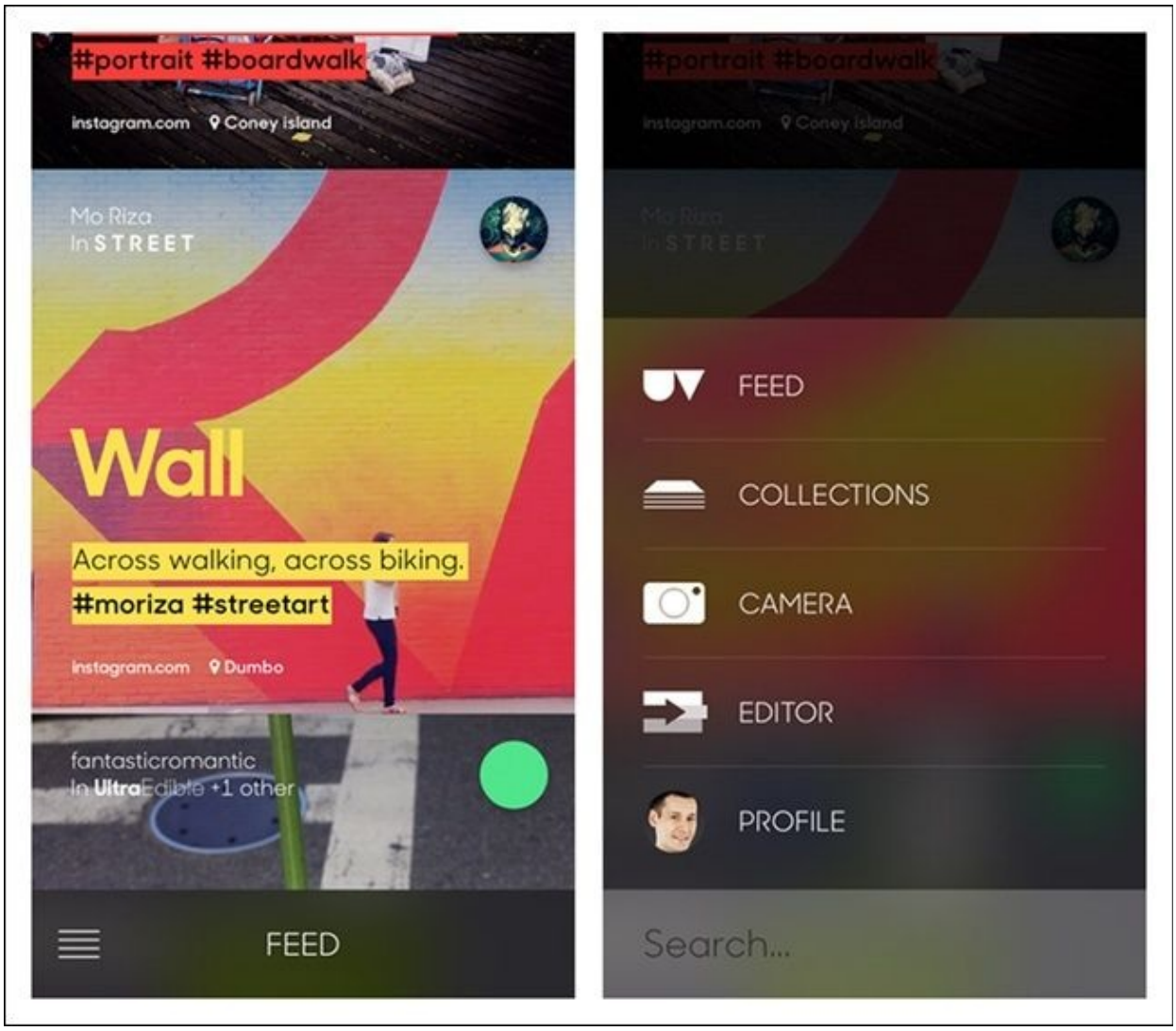



*Figure 1-61. Ultravisual for iOS: overlay Toggle Menu comes up from the bottom*

A key convention of the Toggle Menu is that whatever gesture reveals the menu — tapping an icon, swiping, or panning, for example — should also hide it.

The menu shouldn’t cover the whole screen, but instead let the background peek through. Tapping anywhere in the background should also hide the menu.

Android provides a specific control, the Spinner, for this type of primary navigation. But keep in mind the Spinner should be reserved for navigating between views in a category, as opposed to jumping between completely different categories.

For example, both the NPR and the NYTimes apps serve up news, and the Spinner offers different ways to slice and dice the massive amount of news content they offer. But if, say, NPR needed to offer other options in this menu,

like Music or Weather, Android guidelines would dictate using a Tab Bar or Navigation Drawer instead.

For Android, use the Spinner control where the Toggle Menu is intended to show the views within a category. For iOS and Windows, bear in mind that the Toggle Menu is a custom control, which could take more time to implement, test, and maintain.

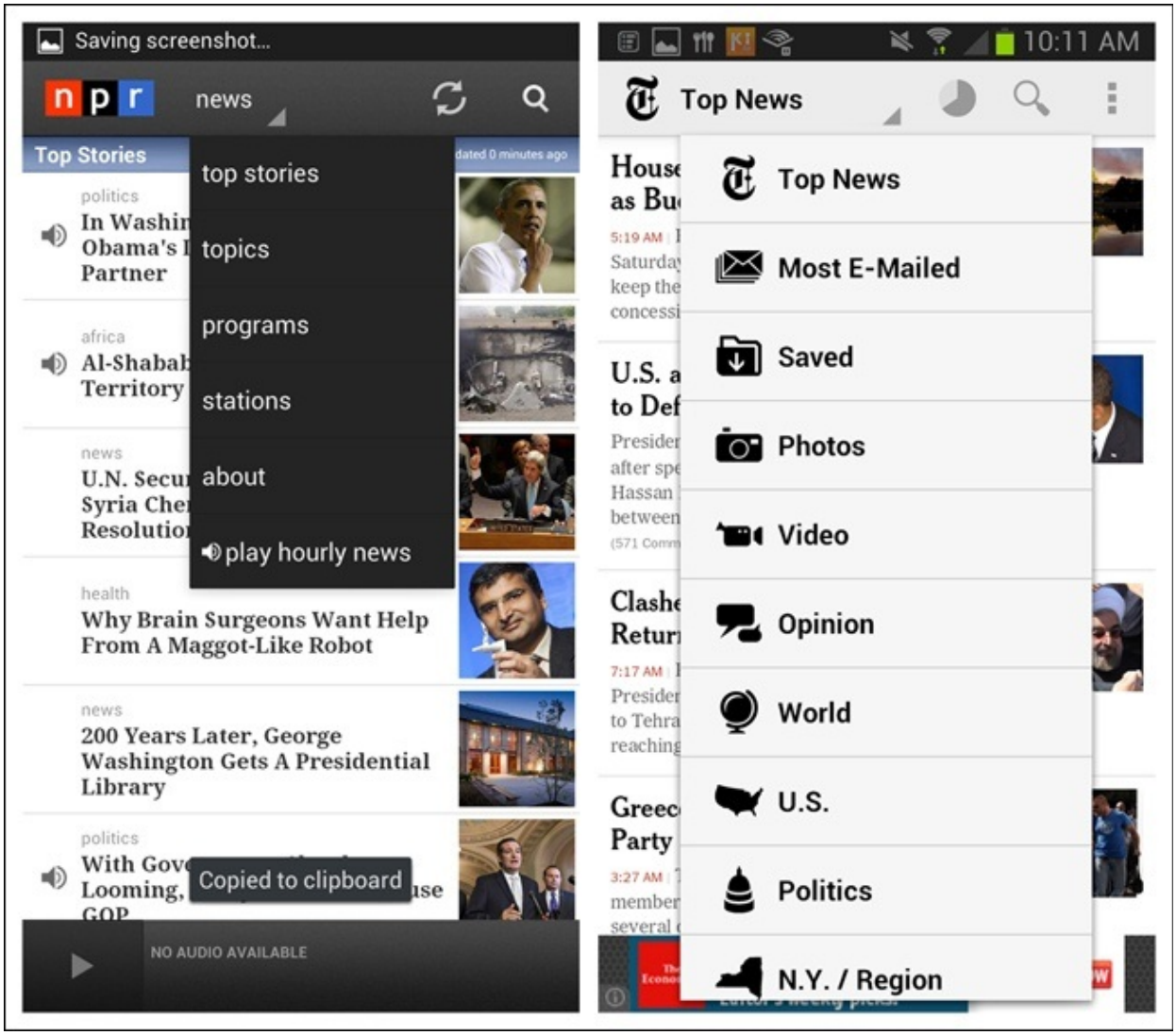



*Figure 1-62. NPR and NYTimes for Android: the Spinner Toggle Menu is for views within a category*

## Pie Menu

Pie Menus — also known as wheels, circular menus, or radial menus — have been around since the ’90s in desktop software, and more recently in web applications. They are also very popular in game design. So I was excited to play with PIE, a Pie Menu interface in the open source Paranoid Android OS, and

was surprised at how well it works.

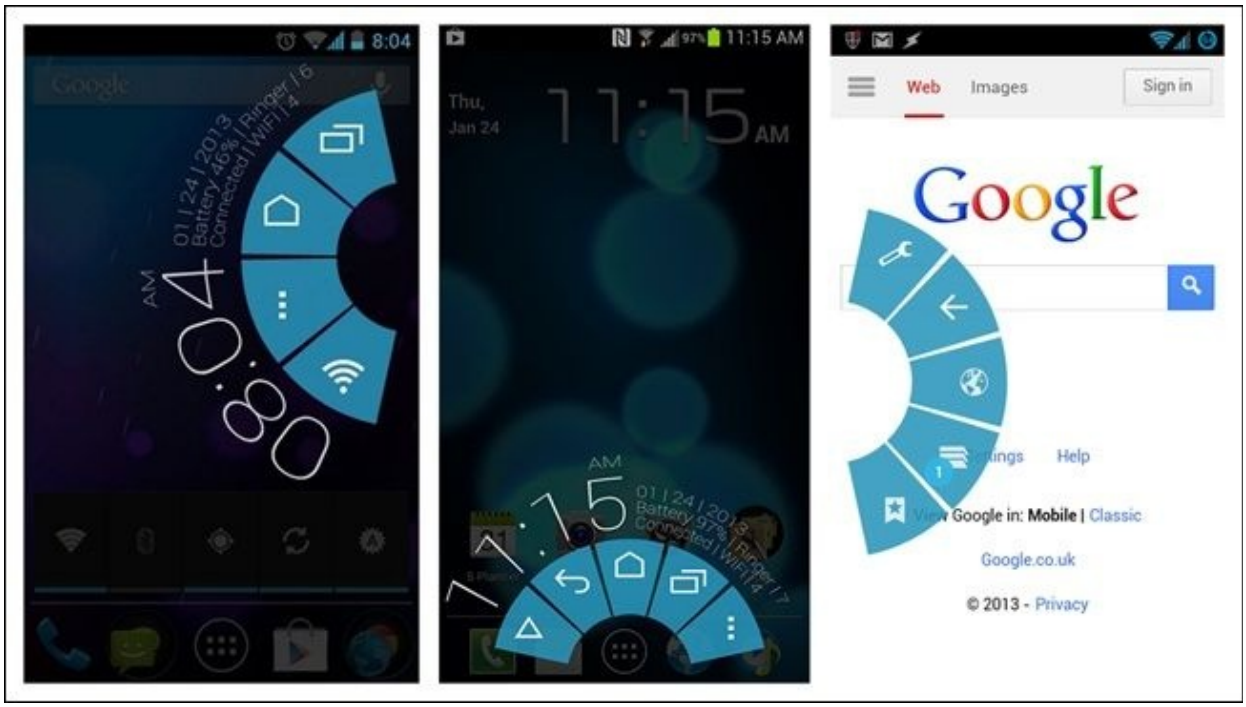



*Figure 1-63. Examples of PIE menus from a phone running the open source Paranoid Android OS (http://bit.ly/1iK2jPr)*

If the Pie Menu option ever becomes a standard part of the stock Android OS, though, I think that will disqualify it for use as primary navigation within apps — it would create too many conflicts between the OS and the apps.

Looking at the other operating systems, I discovered only a handful of apps experimenting with this pattern for primary navigation. The examples I did find were discouraging. PortalWebBrowser, for instance, has a multitier wheel that requires psychic powers and surgeon-like precision to navigate.

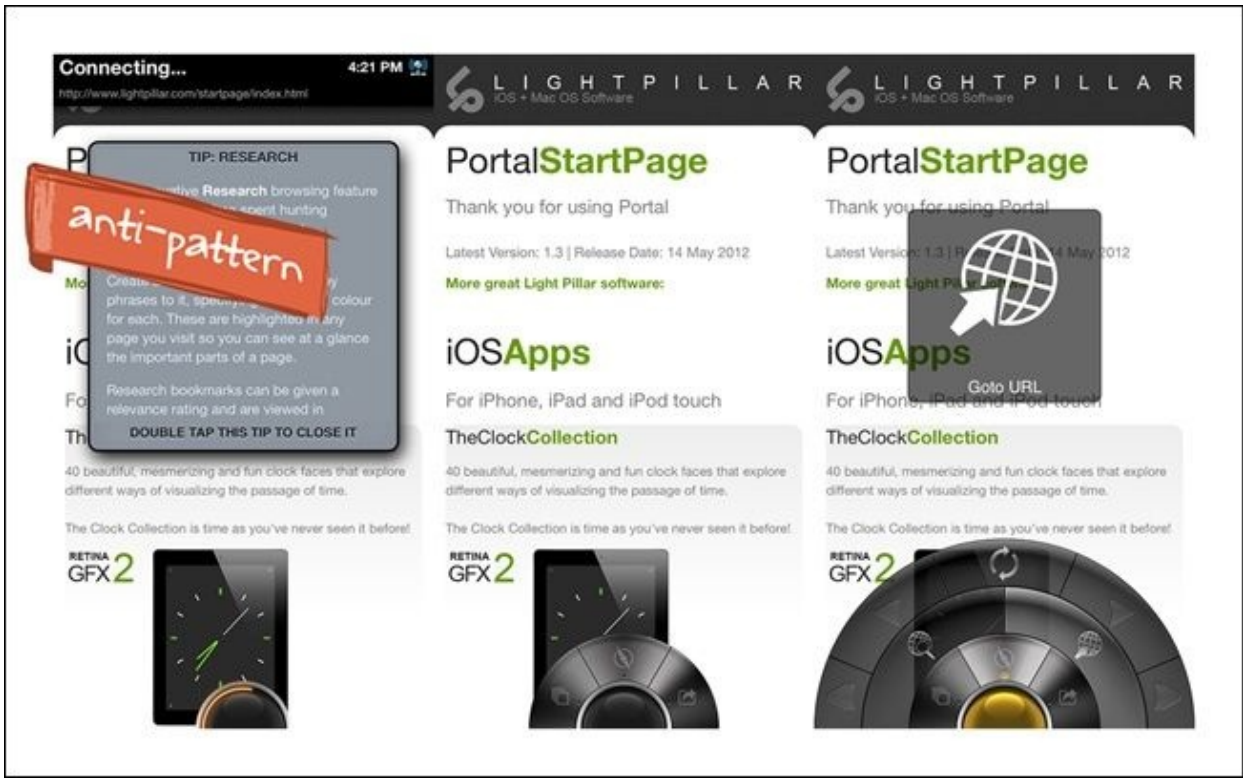



*Figure 1-64. PortalWebBrowser for iOS: multitiered Pie Menu requires a surgeon's touch to use*

The primary design differences between Paranoid Android's PIE and the Pie Menu in PortalWebBrowser are the latter's multiple tiers and tiny touch targets. Having to tap, hold, and then slide across tiers to variably sized wedges is not a simple or natural gesture. There are some good examples of Pie Menus for selecting actions, though; see Chapter 5.

**NOTE**

This is probably the weakest pattern for primary navigation and should be avoided for any menu with multiple tiers. If you have an application with flat information architecture, consider the Tab Menu — familiar to all users — instead.

## Secondary Navigation Patterns

This chapter didn't feel complete with only primary navigation patterns, so I broadened it to include secondary navigation. By *secondary* navigation, I mean moving about within a selected module. For example, Starbucks uses the Tab Menu for primary navigation and a Springboard for secondary navigation on the Home screen. Similarly, Brit & Co. uses a Tab Menu for primary navigation (a Windows Pivot control) and a Springboard for secondary navigation in the Browse module.

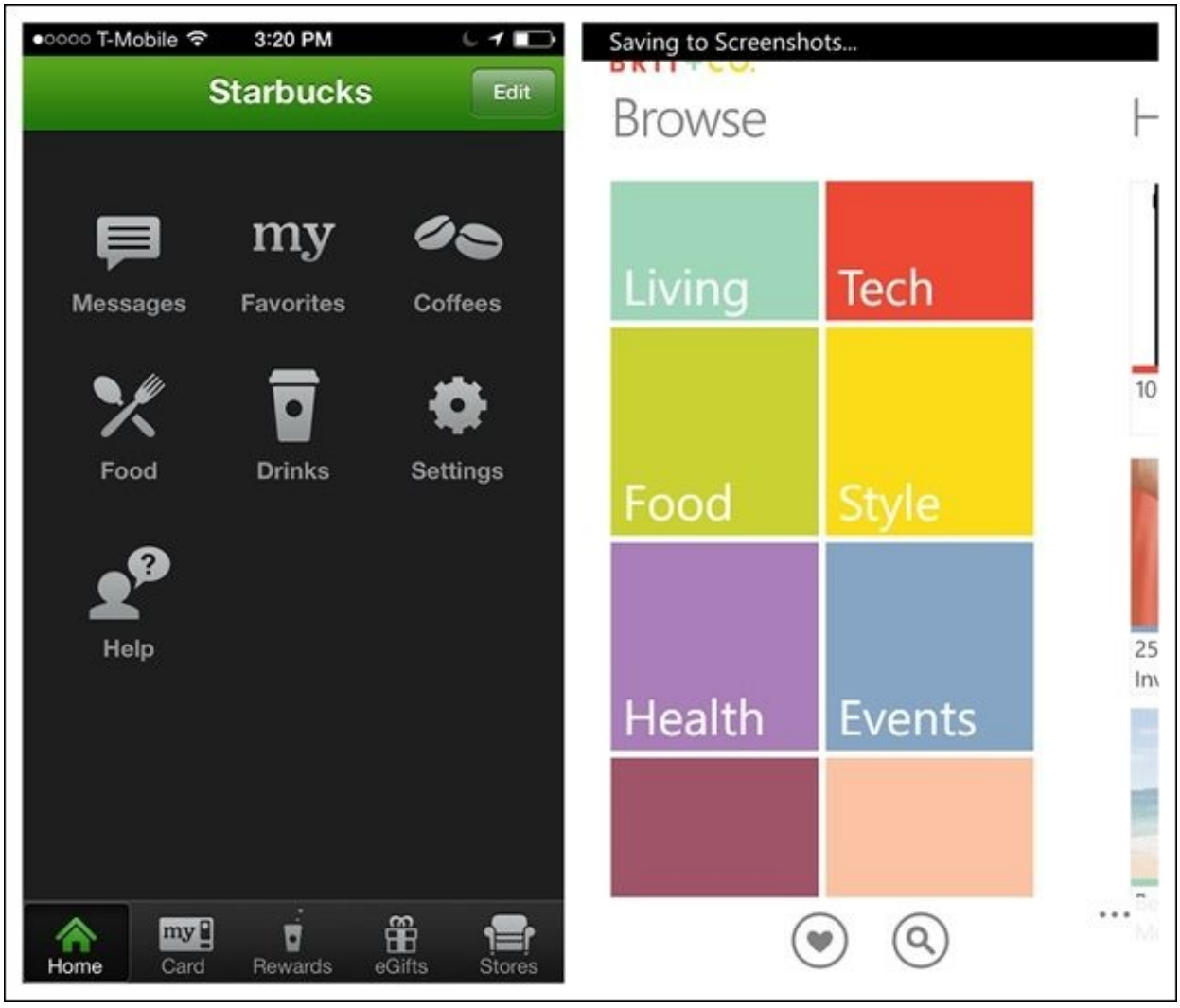



*Figure 1-65. Starbucks for iOS and Brit & Co. for Windows Phone: Springboards as secondary navigation*

All of the primary navigation patterns can also serve as secondary navigation patterns. It is common to see Tabs with Tabs, Tabs with Lists, Tabs with a

Dashboard, a Springboard with a Gallery, and so on.

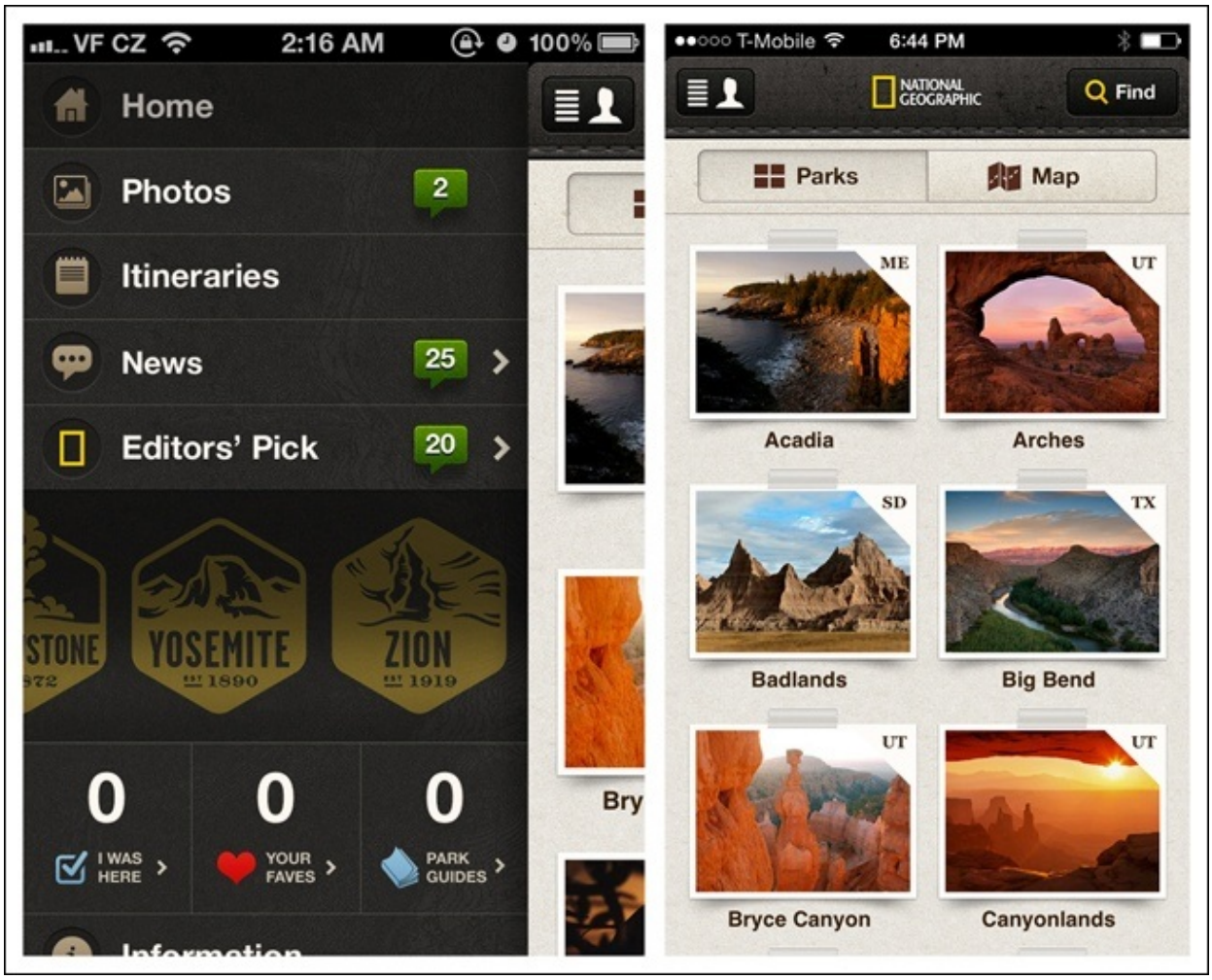



*Figure 1-66. National Parks by National Geographic for iOS: Side Drawer for primary navigation, Gallery for secondary*

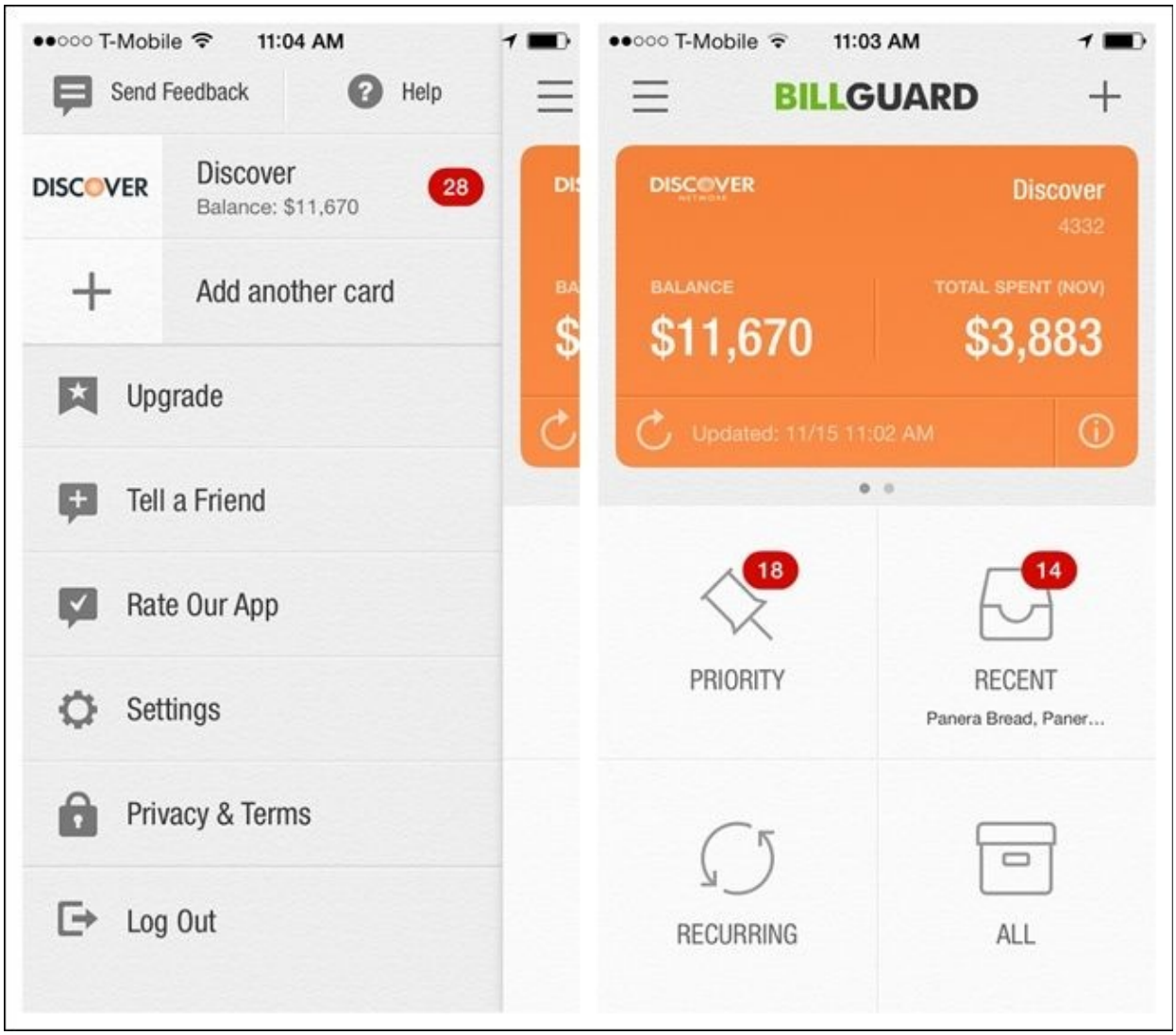



*Figure 1-67. BillGuard for iOS: Side Drawer for primary navigation, Springboard for secondary*

Additional patterns that work well for secondary navigation include Page Swiping, Scrolling Tabs, and the Expand/Collapse Panel.

## Page Swiping

This pattern can be used to navigate quickly through content using the swipe gesture. The most common way to communicate this navigation pattern is via *page indicators* (the iOS term for the horizontal line of little dots). The card metaphor also works for paging, as in Ness and Foodspotting. In these examples, partially visible background cards or pages cue the user to swipe.


*Figure 1-68. Audible for iOS: Tabs for primary navigation, Page Swiping for secondary; note page indicators*

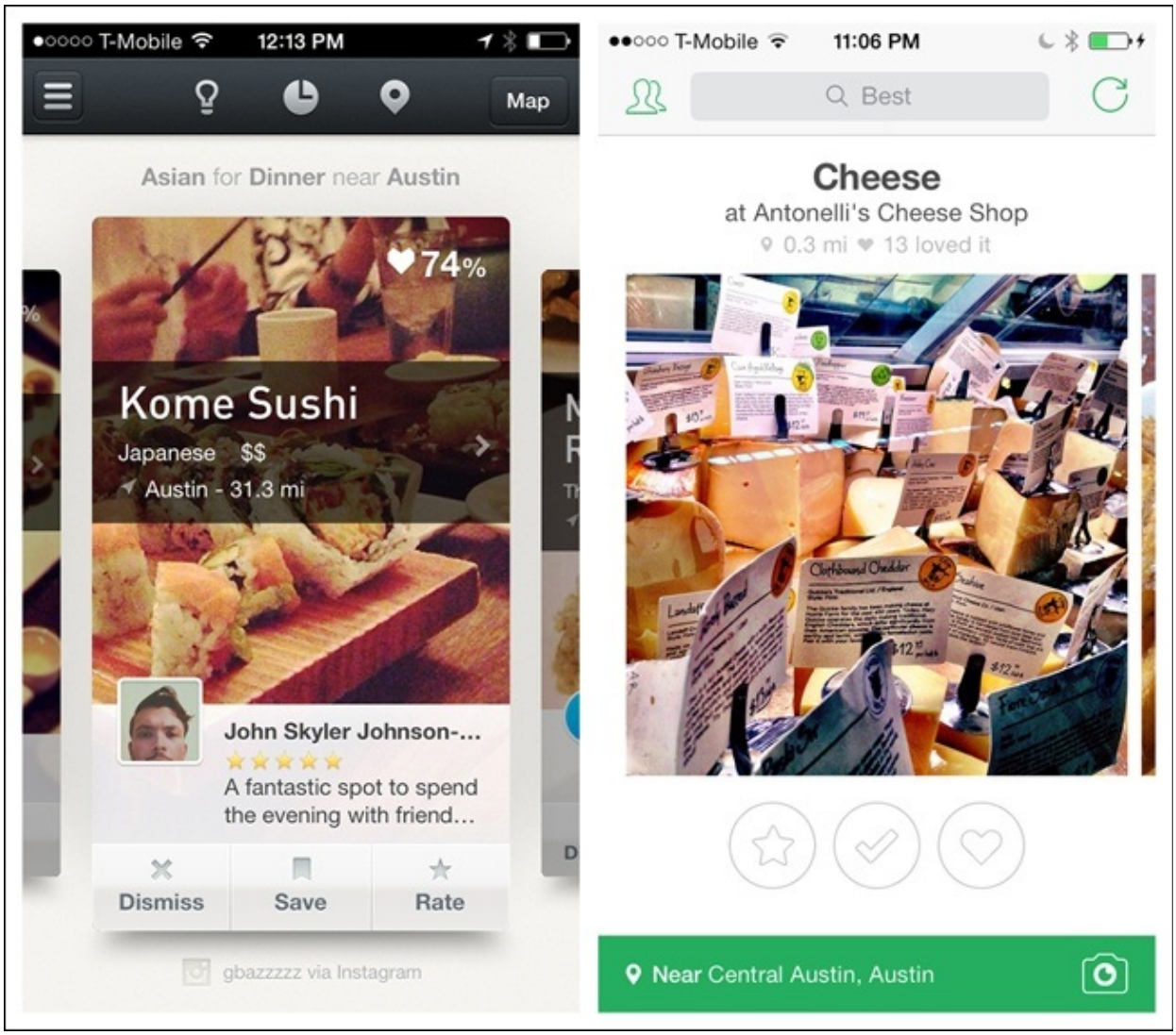



*Figure 1-69. Ness and Foodspotting for iOS: partially visible content cues user to swipe*

Many Android apps offer a similar paging experience. In Google Maps, you can swipe through the list of search results. A tip is shown on first use to communicate this Page Swiping option.

Ark Mail, like Gmail, offers Page Swiping for quick navigation through email messages. A thin footer shows the total number of messages, current message number, and Newer and Older labels in the corners, giving some indication that swiping horizontally will reveal another email.

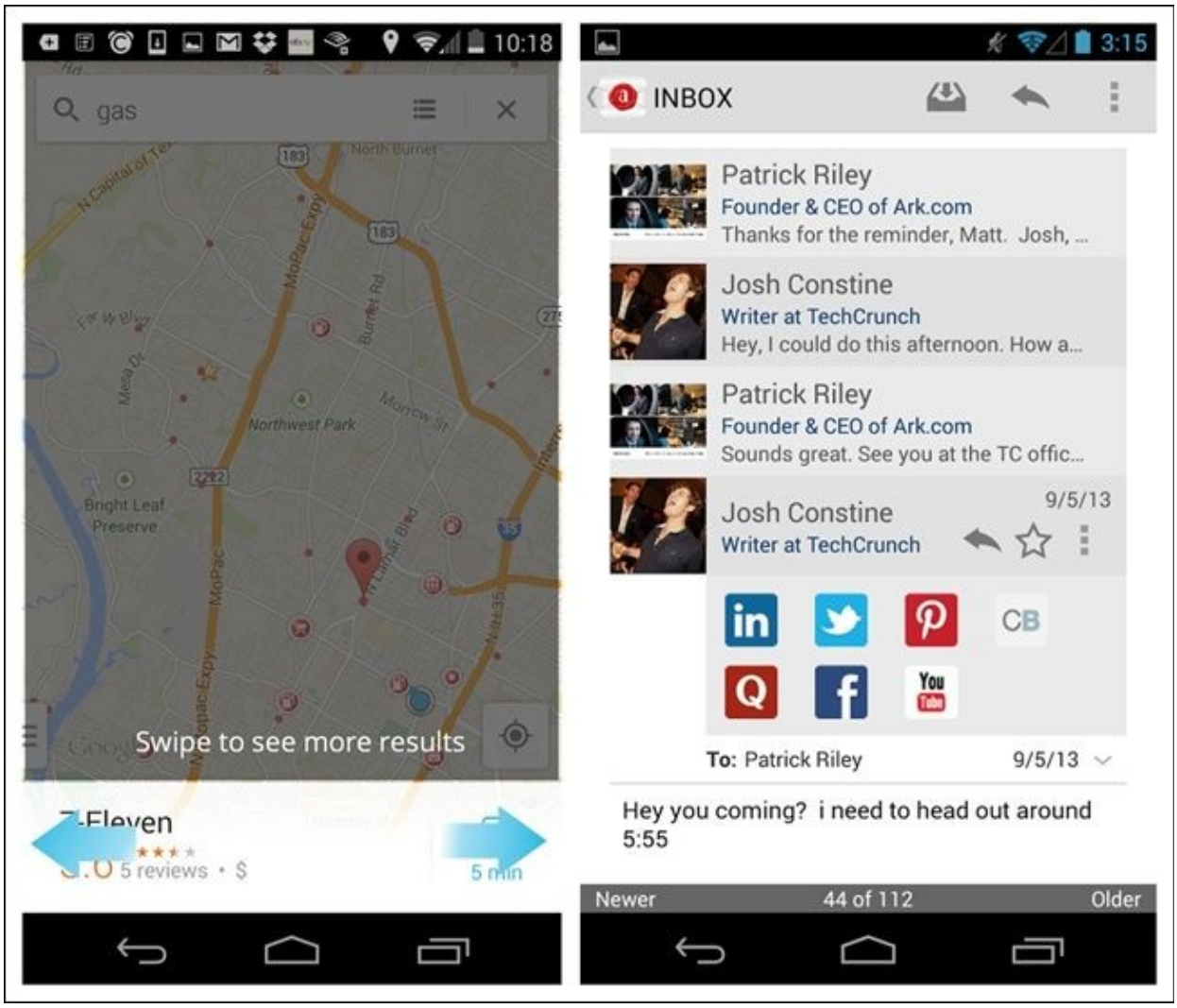



*Figure 1-70. Google Maps and Ark Mail for Android*

News360 is an anti-pattern implementation of Page Swiping, because of its lack of affordance. Adding the paging indicator dots across the top, as in HuffPost, would have made it immediately obvious that a horizontal swipe gesture is required to see the next story, versus a vertical scroll.

**NOTE**

Use Page Swiping to take advantage of mobile's gestural controls, instead of relying on desktop holdovers like Next buttons or Tabs. But provide visual affordance that swiping is available.

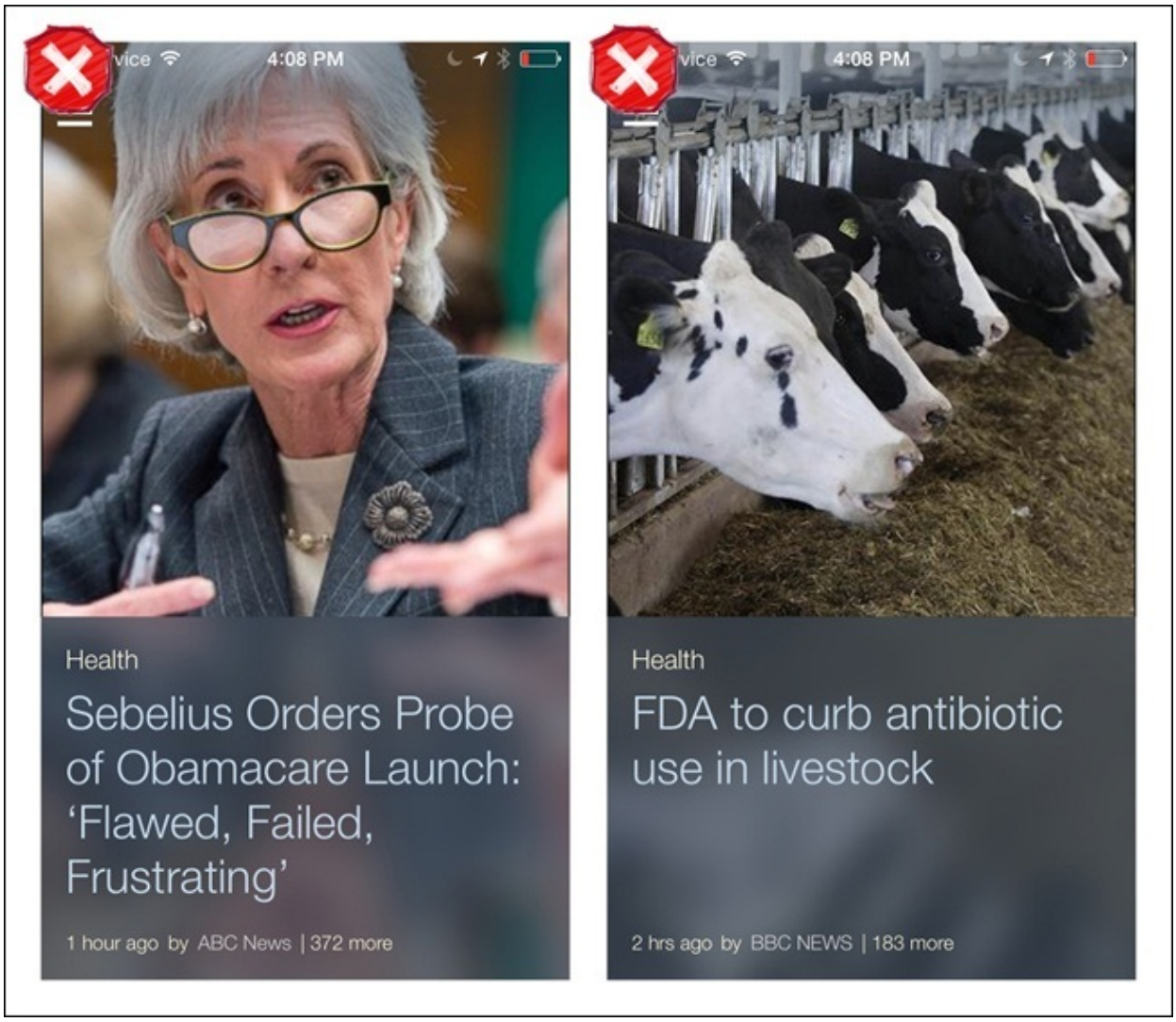



*Figure 1-71. News360 for iOS: no affordance that horizontal swiping shows the next article*

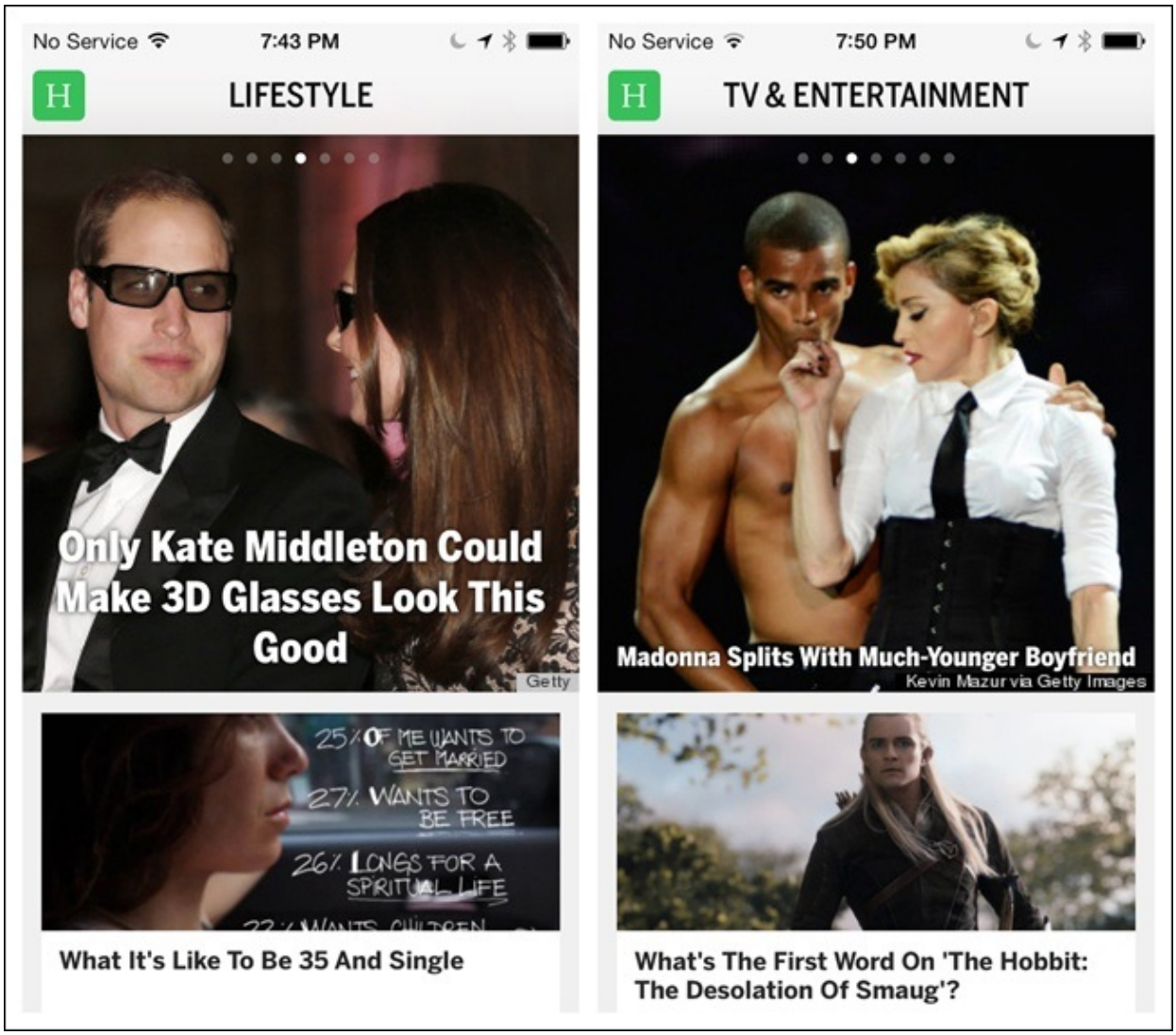



*Figure 1-72. HuffPost for iOS: subtle paging indicators across the top provide affordance to swipe*

## Scrolling Tabs

The Android design guidelines (*http://bit.ly/1hsr9VF*) refer to these secondary navigation controls as Scrolling Tabs, so I stuck with that term. This pattern is useful for displaying multiple categories or views within a specific module. Scrolling Tabs are usually thinner than standard Tab Bars since they are not necessarily touch targets. More typically, they are an affordance to swipe horizontally.

In Google Play, Scrolling Tabs offer a way to filter the results after selecting an item (such as “Movies & TV”) in the Navigation Drawer. Examples from Songza and TuneIn show other ways to integrate the pattern.

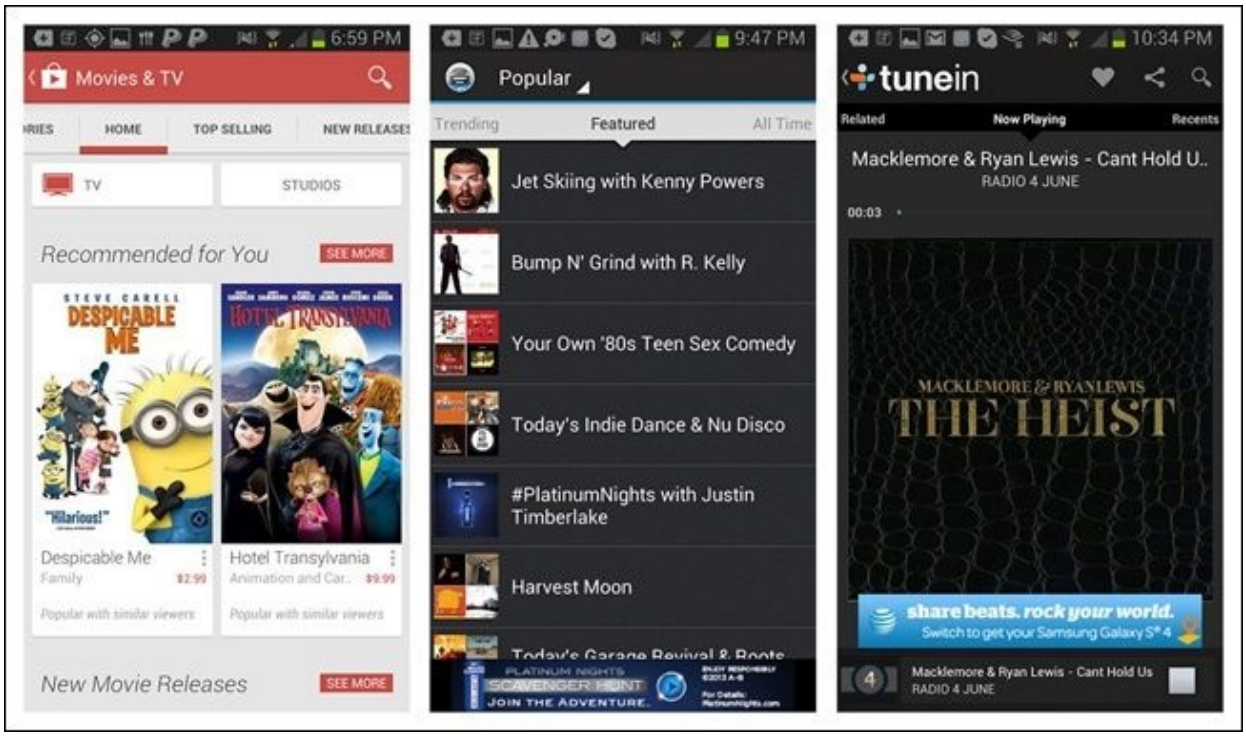



*Figure 1-73. Google Play, Songza, and TuneIn for Android: Scrolling Tabs for secondary navigation*

If you incorporate this pattern, make sure your design clearly indicates the selected tab.

## Accordion

An Accordion lets the user see more information while staying on the same screen. This pattern can be more efficient than navigating to a new screen, and then having to navigate back up. Note that examples from Elevatr, Flava, and the Android Play Store all use familiar icons to indicate a panel's expanded or collapsed state.

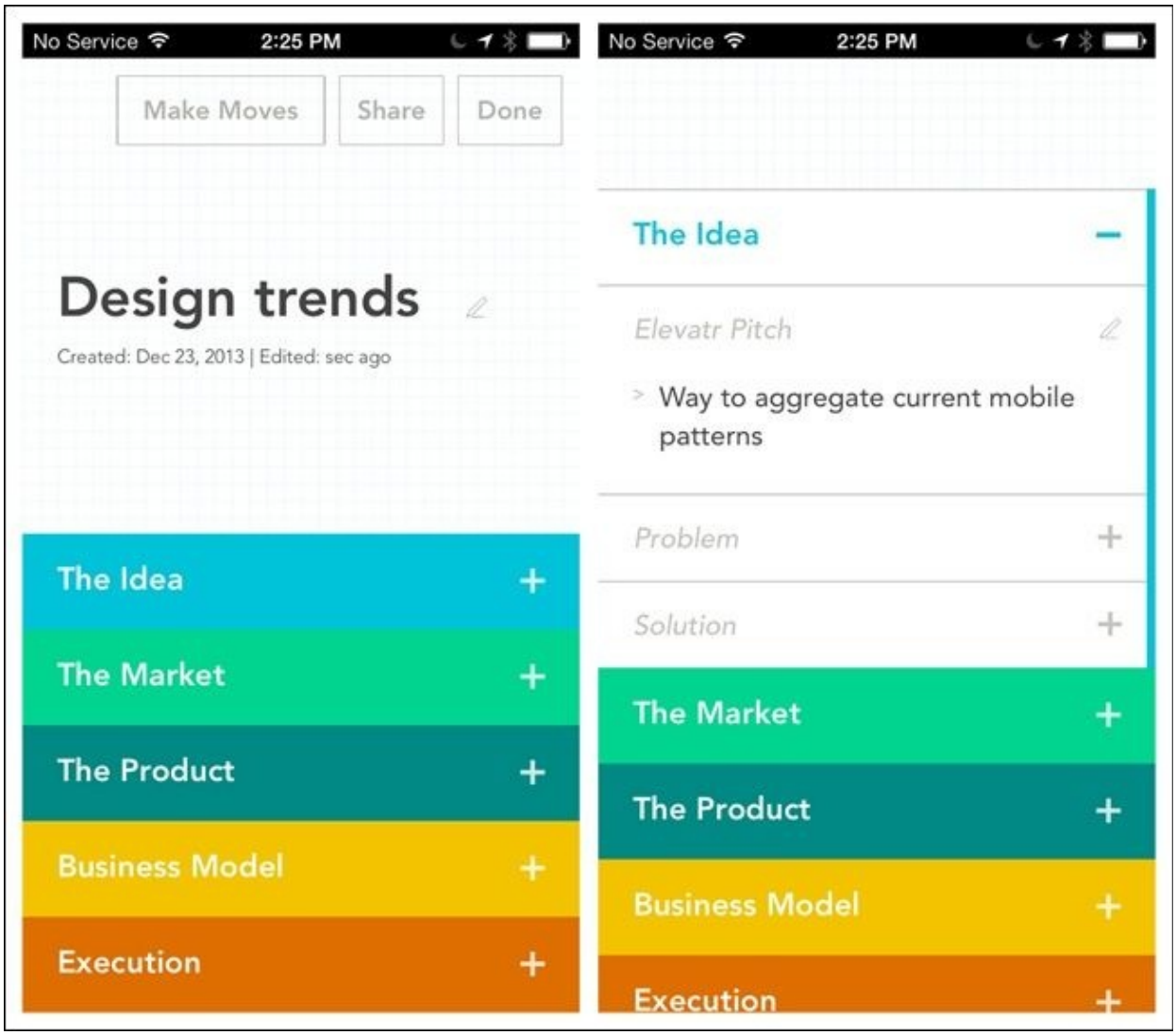



*Figure 1-74. Elevatr for iOS: "+" turns to "–" and heading colors reverse when panel expands*

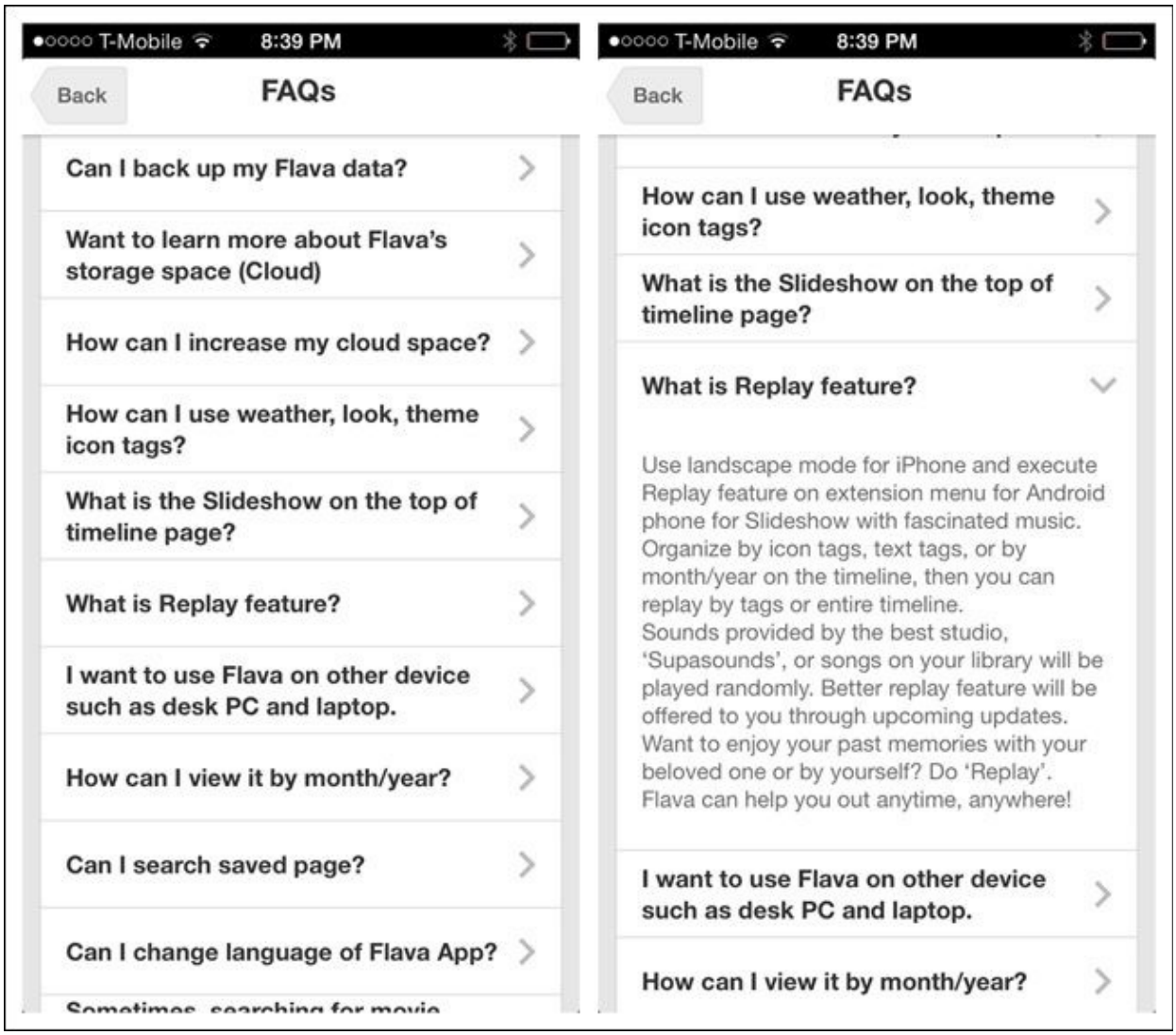



*Figure 1-75. Flava for iOS: down arrow indicates expanded panel*

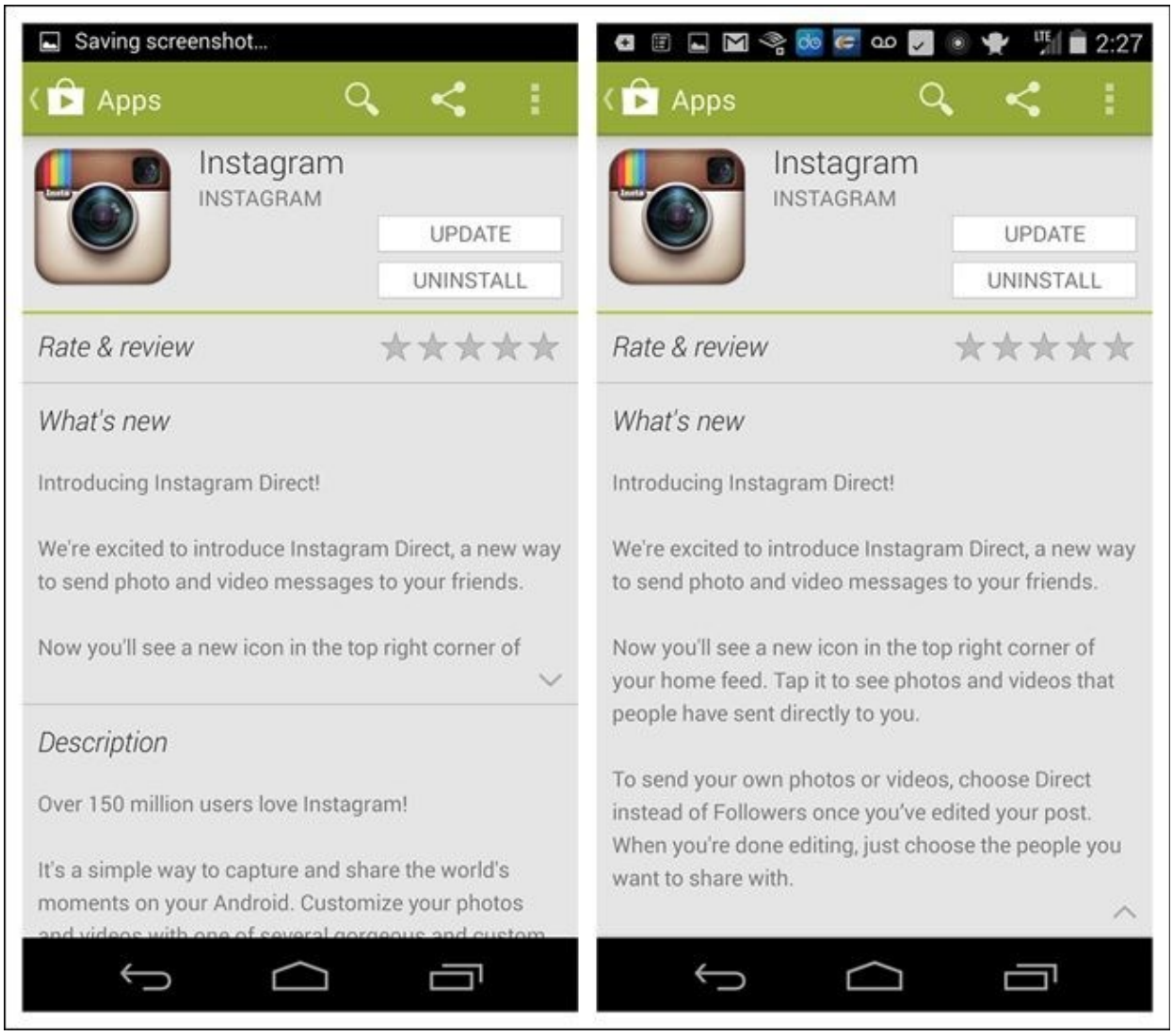



*Figure 1-76. Play Store for Android: expand/collapse in the What's New panel*

**NOTE**

Use a familiar icon for communicating the Accordion's open or closed state.

# Chapter 2. Forms


*Patterns*

> Sign In, Registration, Registration with Personalization, Checkout, Calculator, Search Form, Multi-Step, Long Form

Most web applications rely extensively on forms for data entry and configuration. And although we have compelling research and design strategies for effective form design, there are still horrible forms all over the Web. We do our best to muddle through them to set up online accounts, buy merchandise, submit applications, answer surveys, and the like.

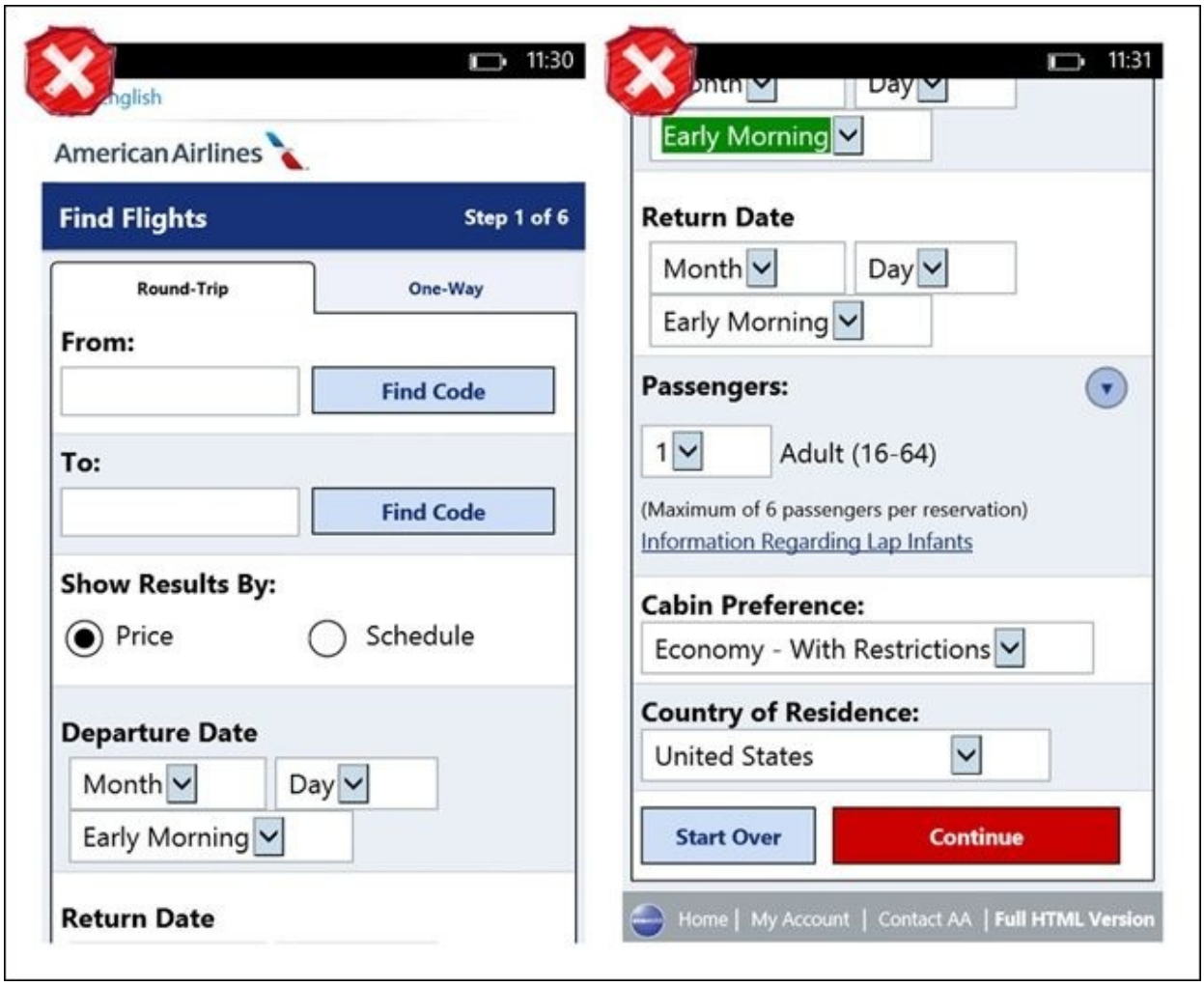



*Figure 2-1. American Airlines flight booking on Windows Phone: poor form design*

Sometimes we succeed, but often we don't. Form abandonment (users who fail to complete and submit forms before giving up) is an enormous and costly problem. But compared to mobile, the Web is pretty forgiving, because mobile forms — rendered on devices that have small screen sizes and restricted user input — give you, the designer, virtually no leeway for bad design.

So before you design any mobile forms, I highly recommend brushing up on form basics with these resources:

- *Web Form Design: Filling in the Blanks* (*http://amzn.to/1g364lv*) by Luke Wroblewski (Rosenfeld Media, 2011)
- *Forms on Mobile Devices: Modern Solutions* (*http://bit.ly/1g369pc*) by Luke Wroblewski
- *Better Mobile Form Design* (*http://bit.ly/1g36zvV*) by Luke Wroblewski
- *Mobile Form Design Strategies* (*http://bit.ly/1g36jwU*) by Chui Chui Tan

- *Mobile Inline Form Validation* (*http://bit.ly/1g36pVe*) by Steven Hoober
- *Removing Stumbling Blocks in Mobile Forms* (*http://bit.ly/1g36uZb*) by Robert Brauber

As a complement to these resources, the form design patterns that follow will help you build apps with forms that work for your users. They include:

- Sign In
- Registration
- Registration with Personalization
- Checkout
- Calculator
- Search Form
- Multi-Step Form
- Long Form

## Sign In

Sign In forms should require a minimal number of inputs: username, password, command button, password help, and an option to register. Some applications do this in a single screen, like Remember the Milk and Groupon.

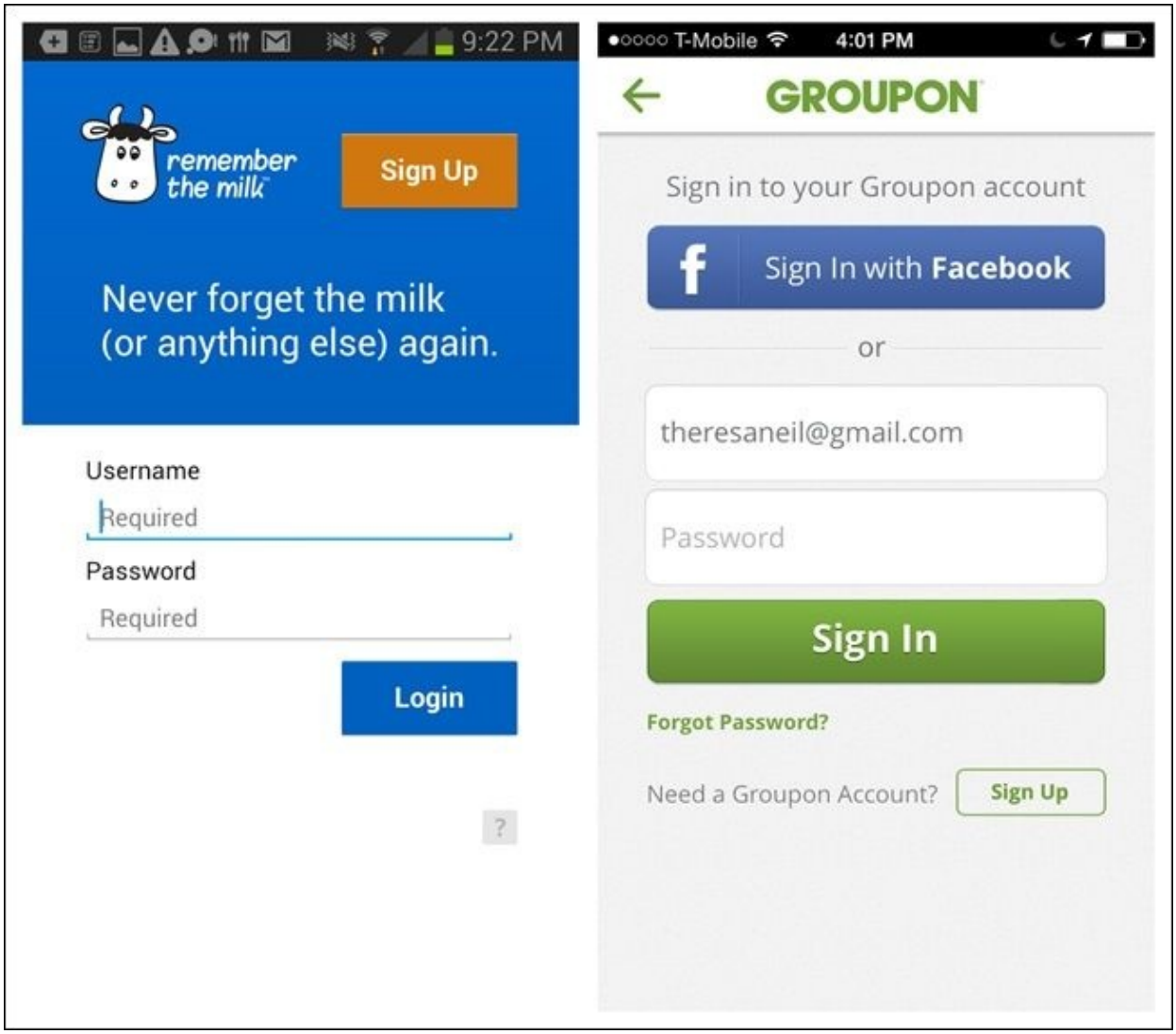



*Figure 2-2. Remember the Milk for Android and Groupon for iOS: minimal Sign In fields*

Other apps, like Instagram, present the Sign In and Register options up front, then take the user to the appropriate form. You should autofocus on the first form field, saving the user an extra tap to open the keyboard.

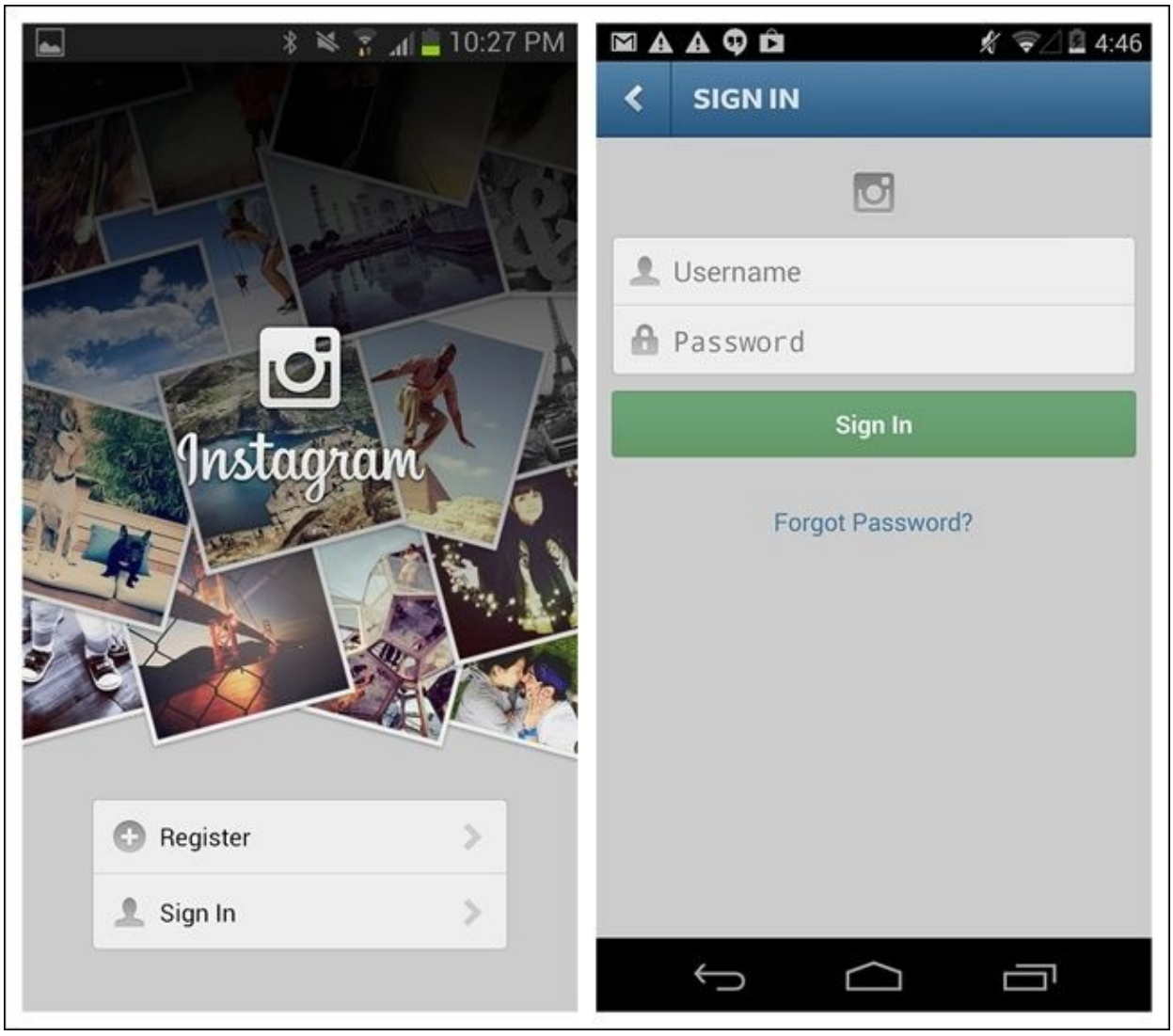



*Figure 2-3. Instagram for Android: Register and Sign In up front*

The Sign In form itself should use a new technique that has substantial support from usability testing and live usage: the unmasked password field. As in LinkedIn, show the password unmasked and/or offer the option to mask it. The open and close icon is common, or just the words Hide/Show (see the Polar example in the section Registration later in this chapter).

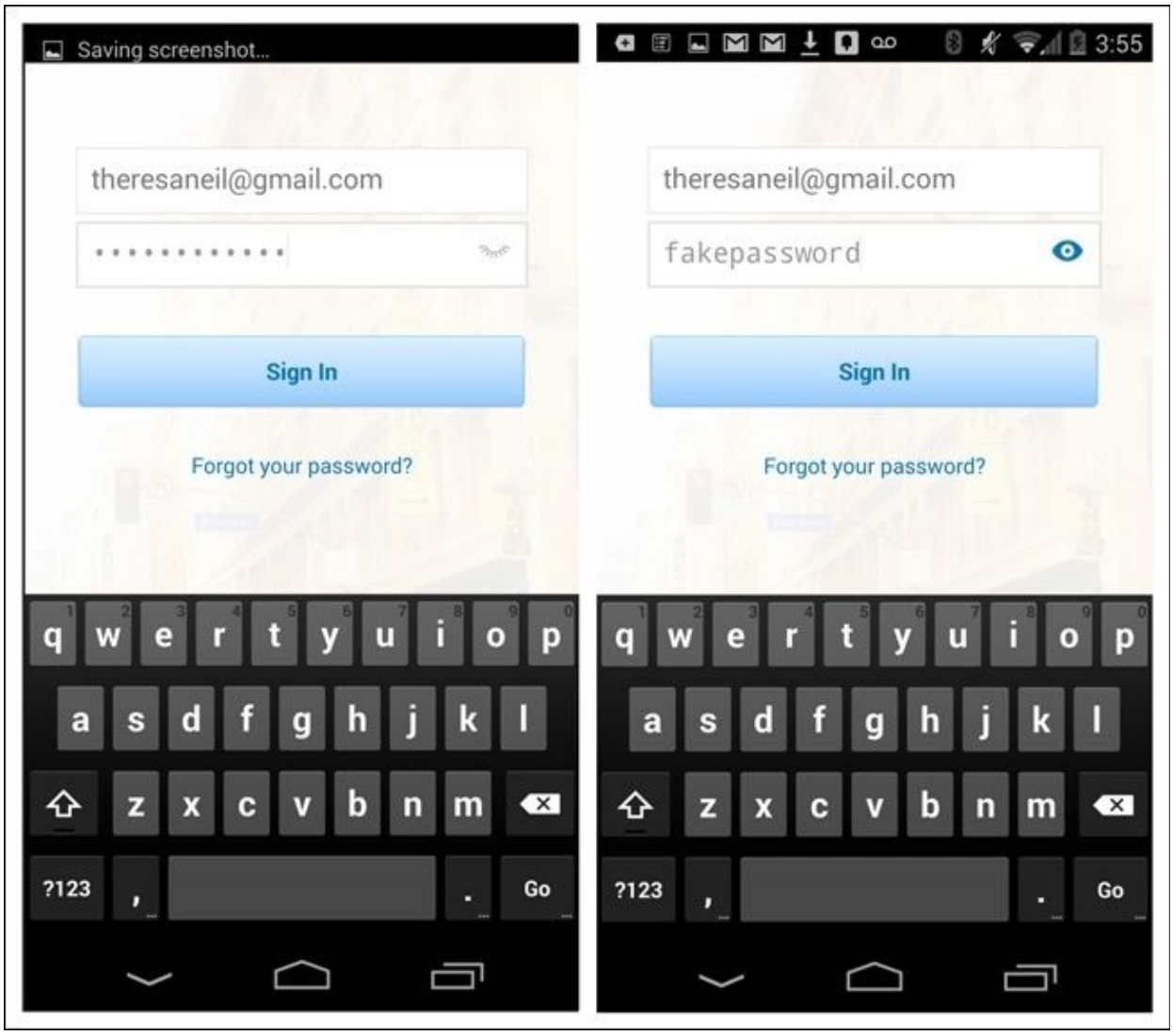



*Figure 2-4. LinkedIn for Android-Unmask password option*

Other apps, like Wunderlist and Box, place the Sign Up and Log In options in context within the promotional tour. By keeping these options present on each screen in the tour, the apps make it easy for prospective users to take action once they have been persuaded to use the app, and equally easy for returning users to simply log in.


*Figure 2-5. Wunderlist for iOS: Sign Up and Log In are persistent throughout tour*

Tabs are another viable option for presenting these two options while keeping navigation to a minimum, as shown with Foodspotting and SigFig.

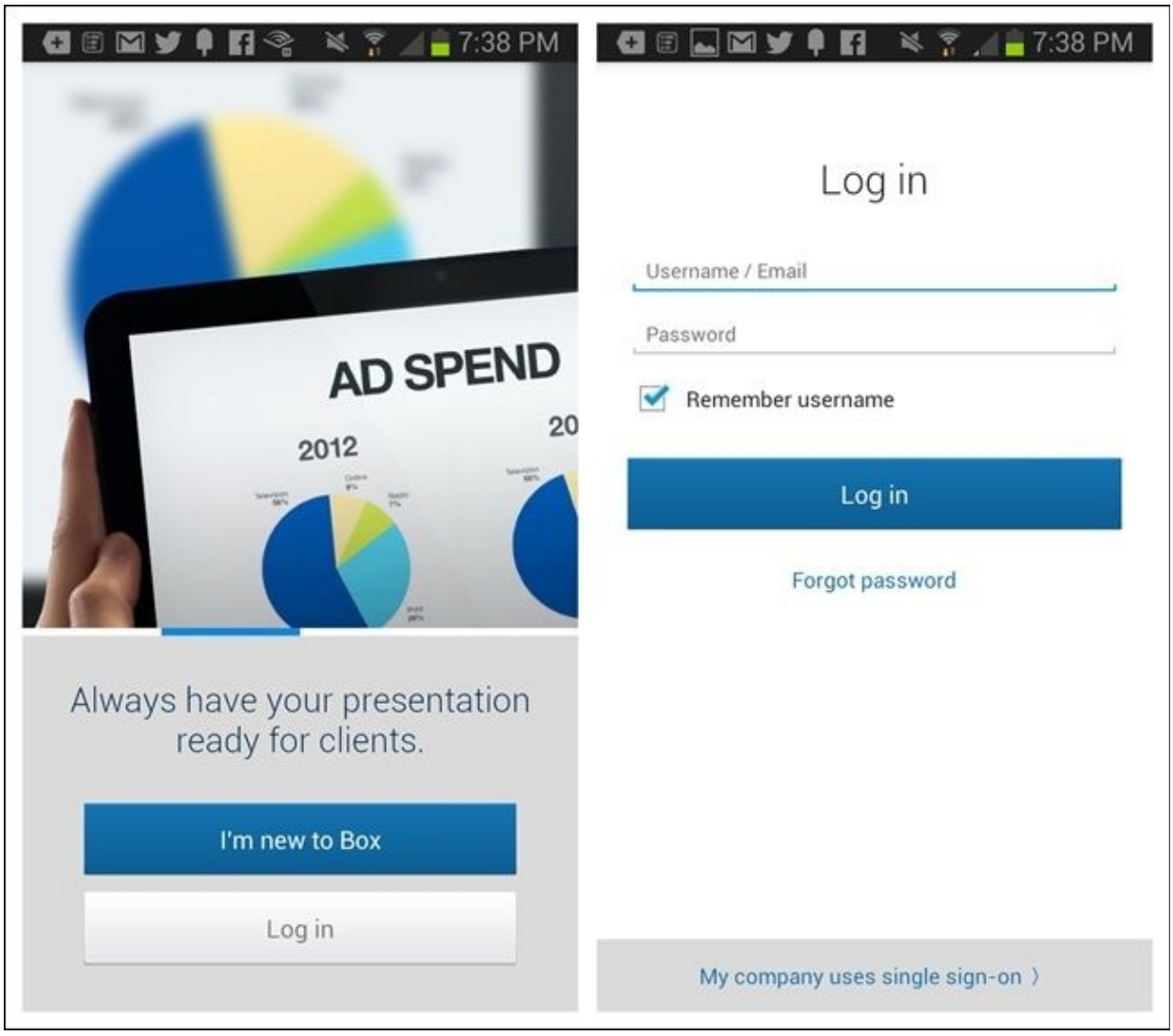



*Figure 2-6. Box for Android makes it easy for users to sign up once sold by the tour*

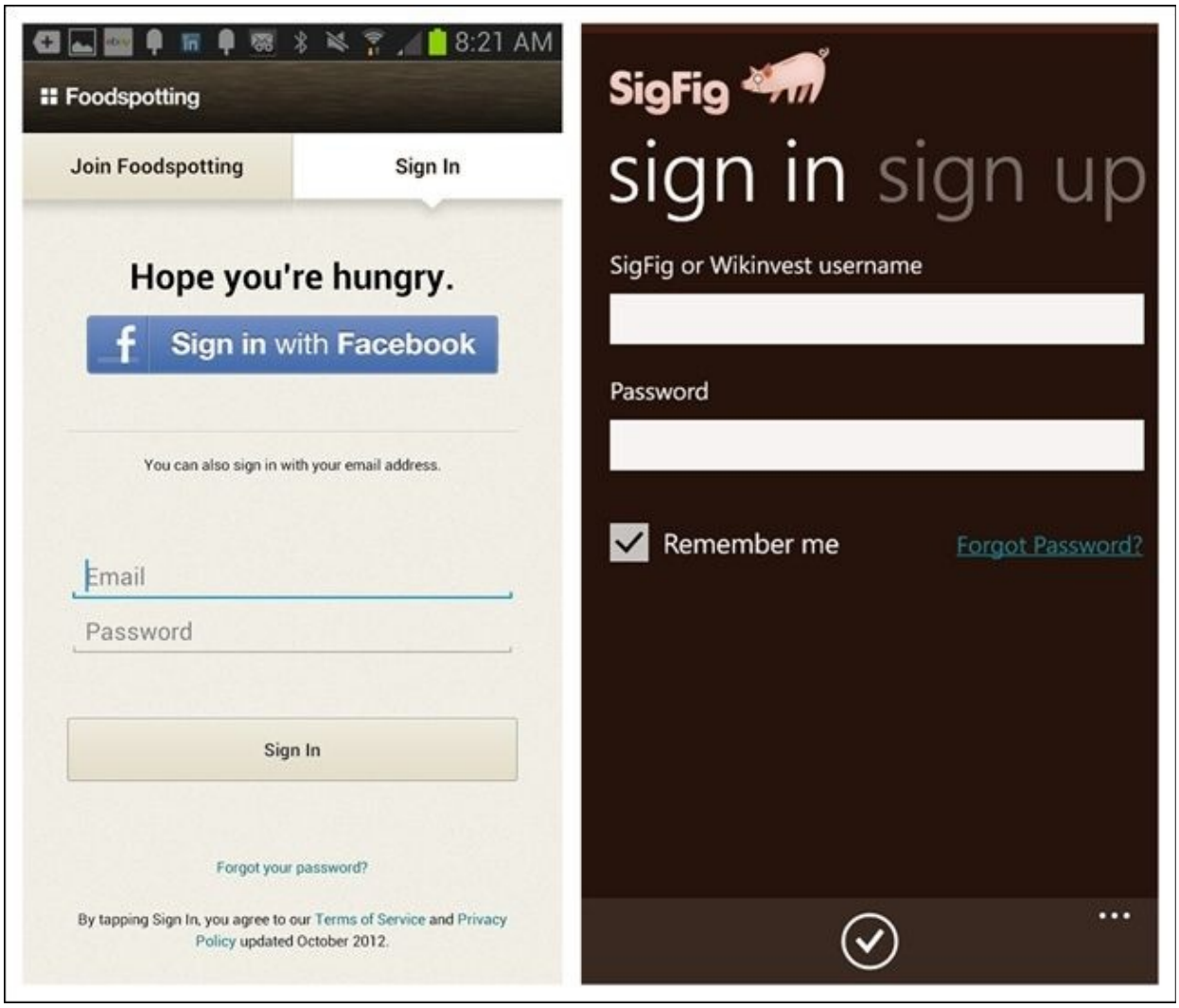



*Figure 2-7. Foodspotting for Android and SigFig for Windows Phone: Sign In and Sign Up on tabs*

Any of these patterns can also include a Social Sign In path, not to be confused with Social Sign Up, which we'll cover in the Registration pattern. Social Sign In will take the user through the social network of his choice to gain access to the application.

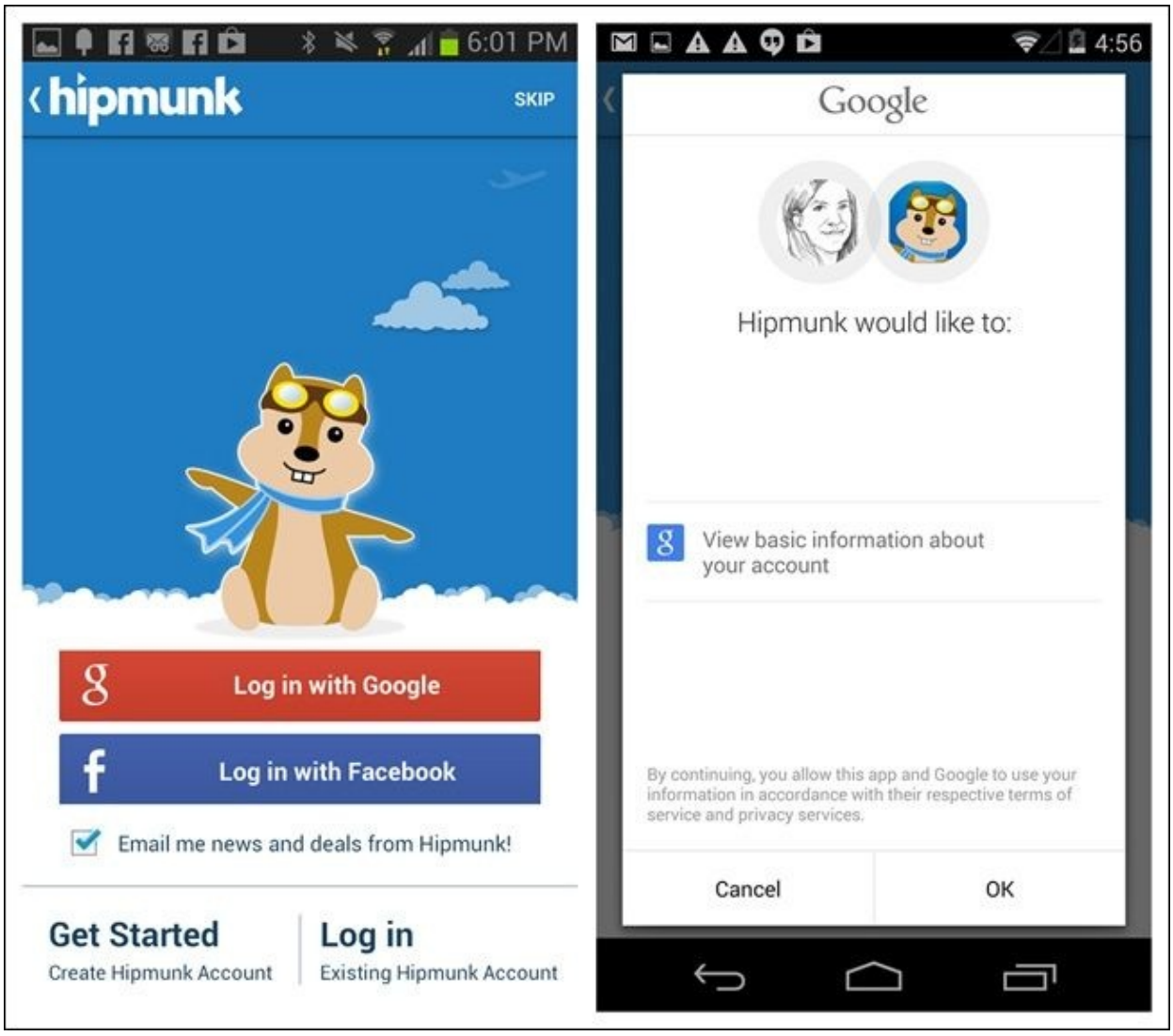



*Figure 2-8. Hipmunk for Android offers three Sign In options: create an account, use Facebook, or use Google*

Another Sign In option gaining popularity with the finance industry is to forgo the username field and just require a password. The user's identity can be authenticated when the application is installed. Users are then only prompted for a password to access sensitive data. Requiring a mobile PIN as a password serves the same purpose, as shown in the examples from Frost Bank and Personal Capital.

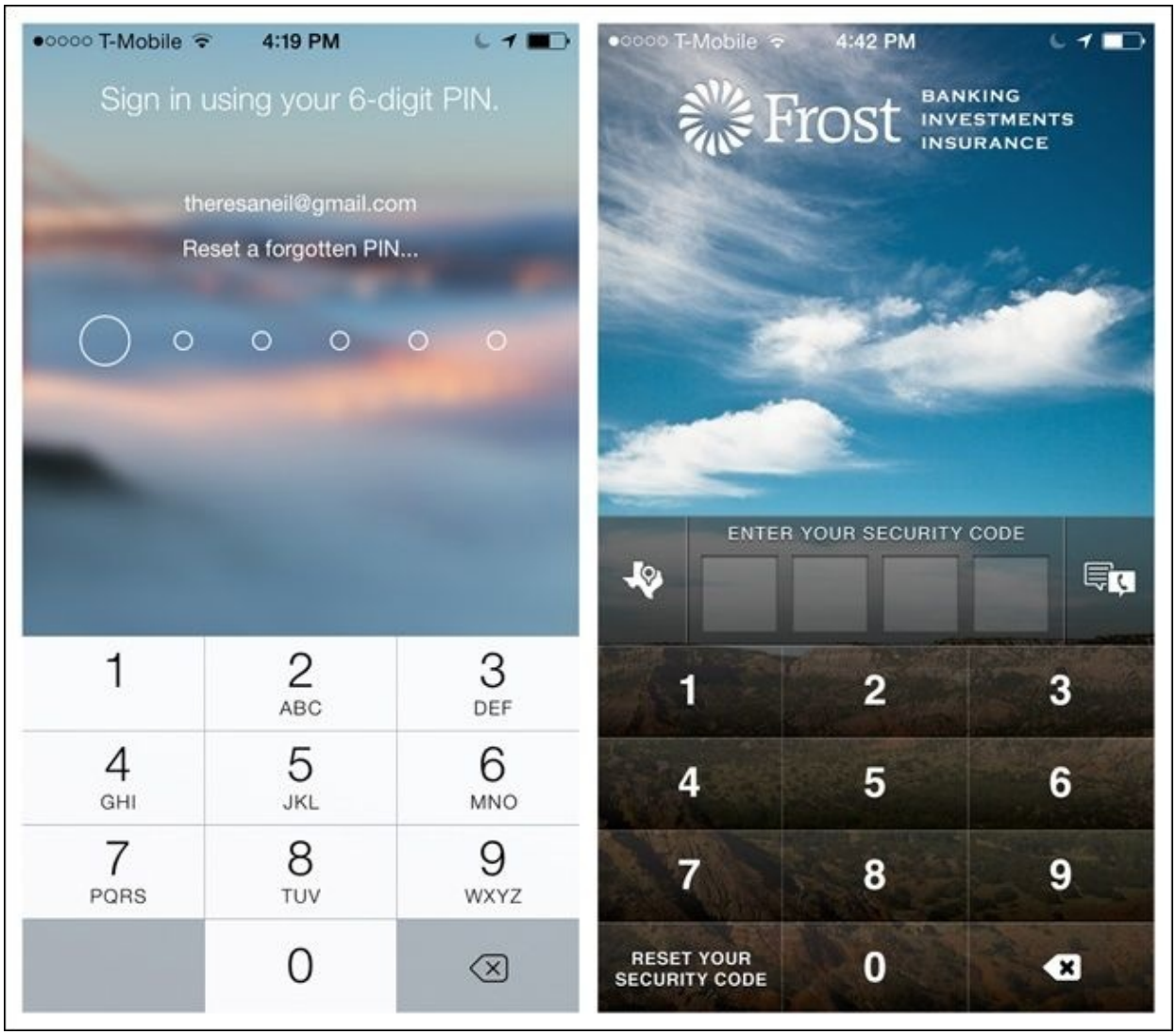



*Figure 2-9. Personal Capital and Frost for iOS require just a password or PIN for Sign In*

In most of the examples we've seen so far, the Sign In form is presented when the app is launched. But there are many cases where it's preferable to allow the user to start using the application without having to sign in or register.

You can happily explore Groupon and Etsy unauthenticated, until it is time to buy an item or save a favorite (authentication could also be required to share, add an item to your wish list, add a comment, etc.). Why should you let folks shop without signing in or registering? Jared Spool explains it in his article "The $300 Million Button" (*http://www.uie.com/articles/three_hund_million_button*). And if you need further convincing, take a look at the "Chapter Extra" by Greg Nudelman in Chapter 11.

**NOTE**

Sign In may not be a necessary first step. Consider where it makes sense to authenticate the user. And don't get creative with the Sign In screen; use standard designs to make it easy for users to sign in.

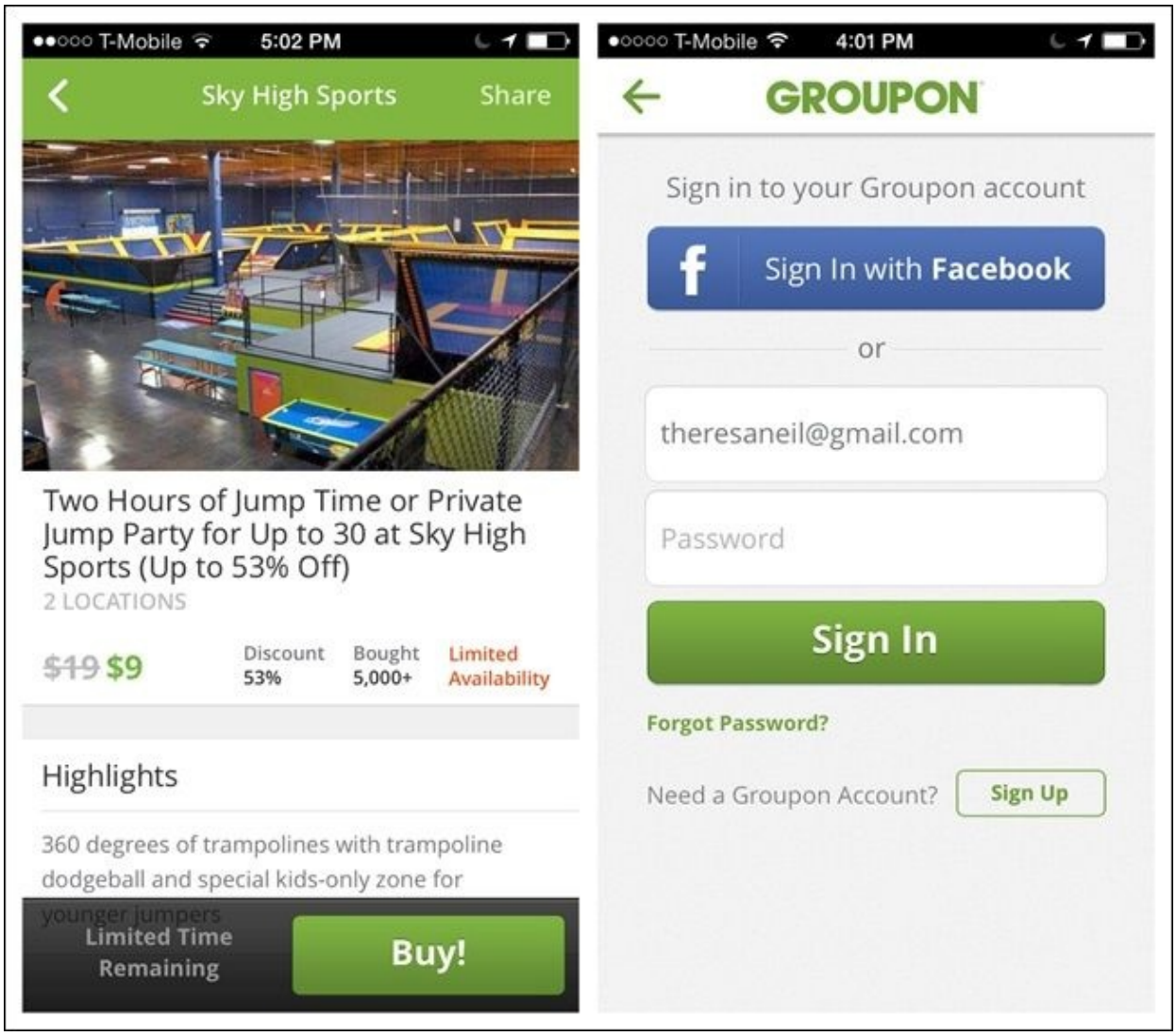



*Figure 2-10. Groupon for iOS: browse freely — you'll be prompted to sign in to purchase*

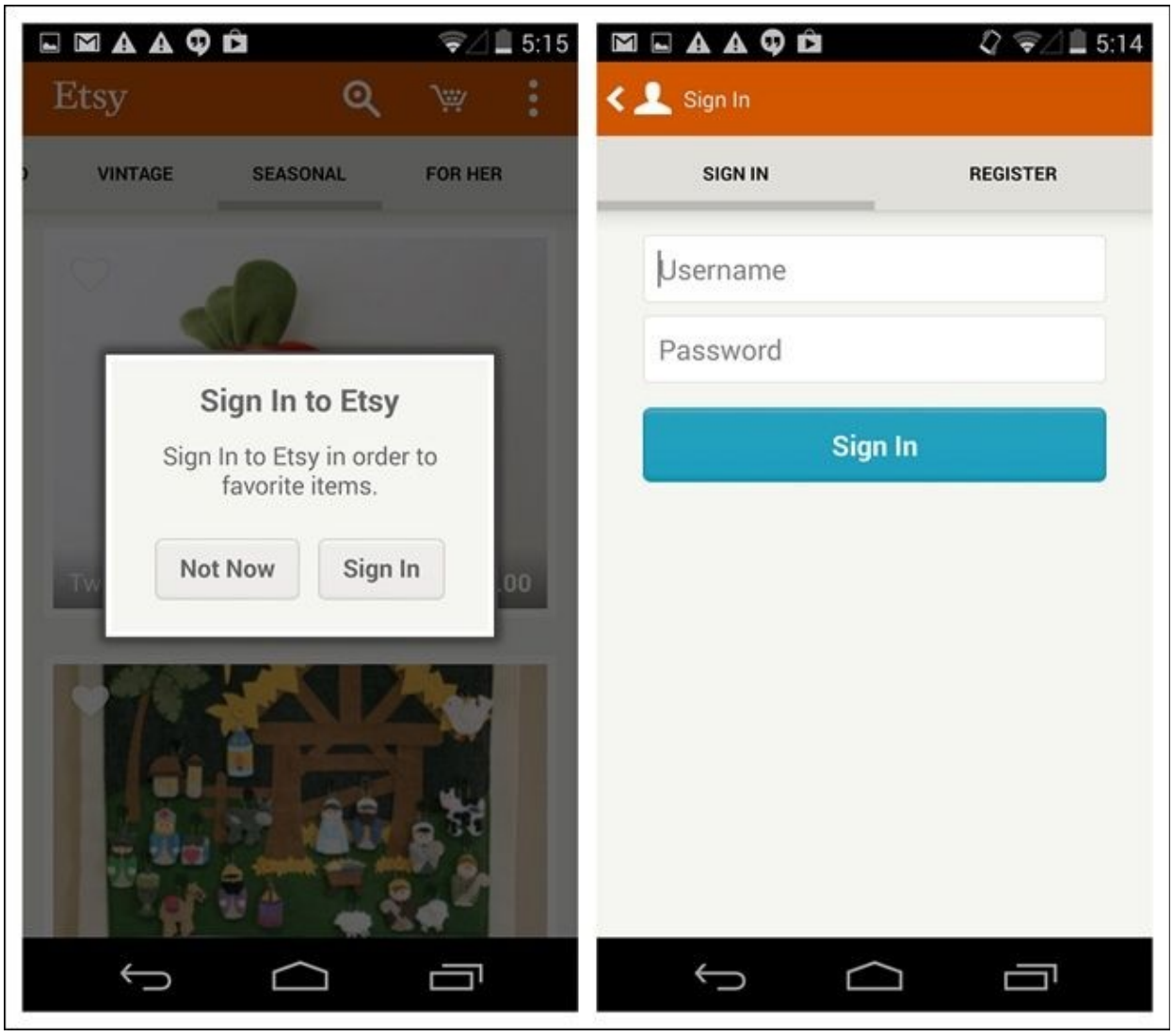



*Figure 2-11. Etsy for Android prompts for Sign In when user "favorites" an item*

# Registration

Businesses love customer data, so they often try to get as much of it as possible at user registration. But Registration forms should have a minimal number of inputs. Since every additional form field demonstrably lowers Registration form conversion, UX consultant and author Chui Chui Tan recommends ruthlessly editing “elements which do not carry important functions.” One easy target has been validated by a lot of user testing: eliminating the redundant Confirm Email and Confirm Password fields found in apps such as NOOK and LearnVest.

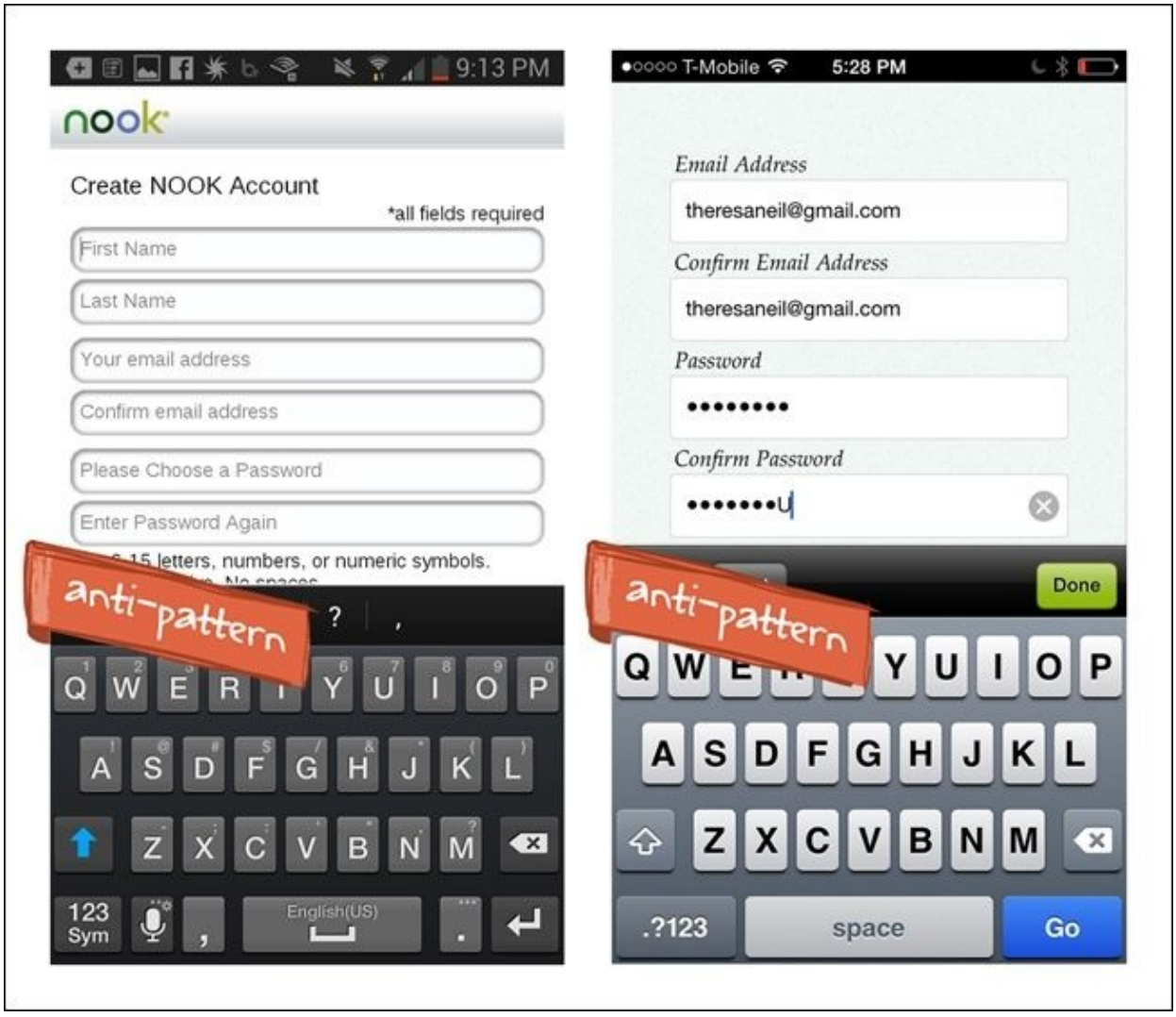



*Figure 2-12. NOOK for Android and LearnVest for iOS: redundant email and password fields*

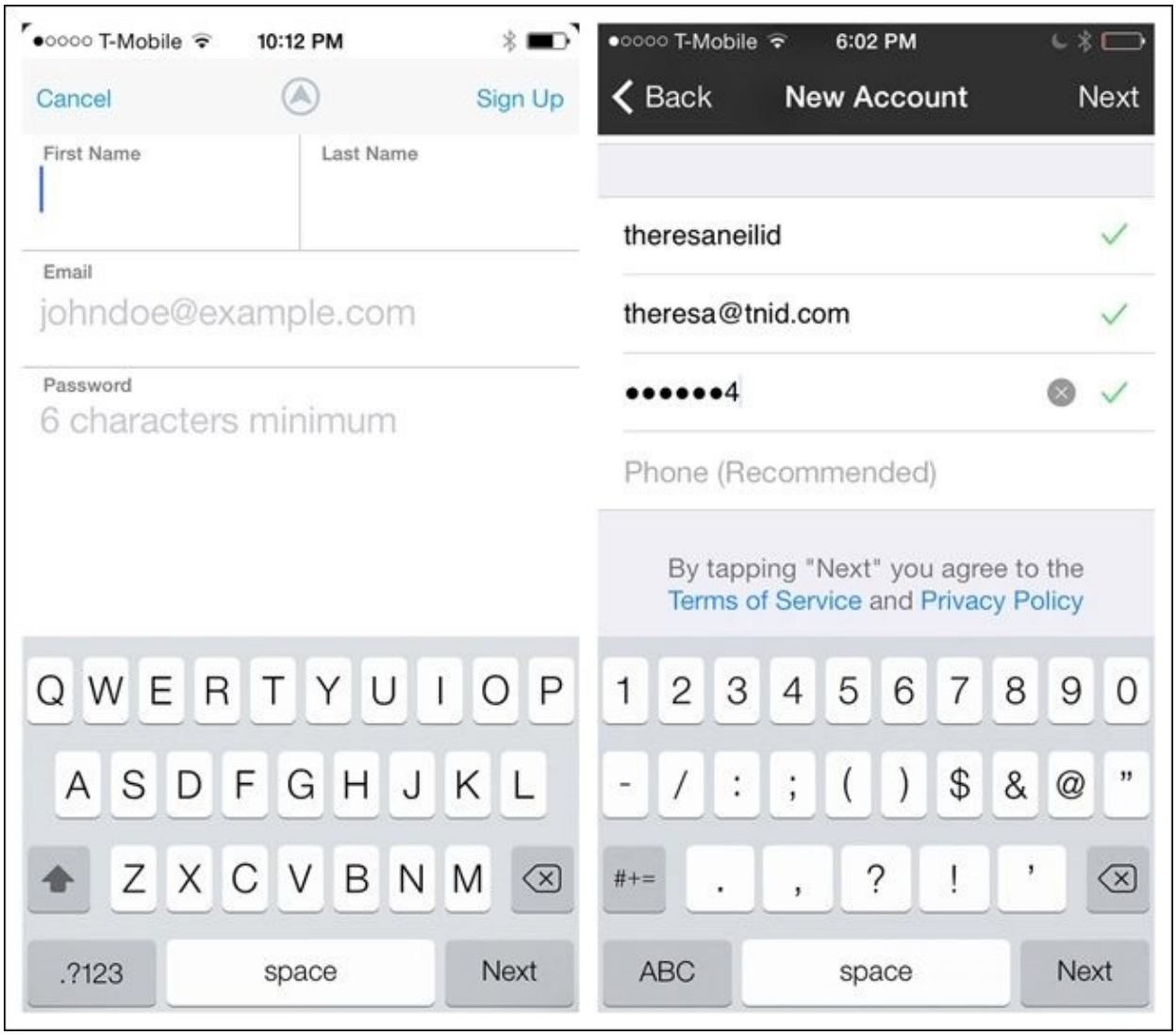



*Figure 2-13. Rove and Kik for iOS: testing validates use of single fields for email and password*

Can’t live without email confirmation? Path’s Registration form is a bit longer, but it clearly marks the optional fields, and instead of asking for the email twice, it prompts users to double-check it.

Since this may be the first form you design for your application, establish a labeling convention that is easy to read. Polar, the fun polling app from form guru Luke Wroblewski provides a good example. Note that each field also has inline feedback and the password field has a Hide/Show option, which further makes redundant password entry unnecessary.

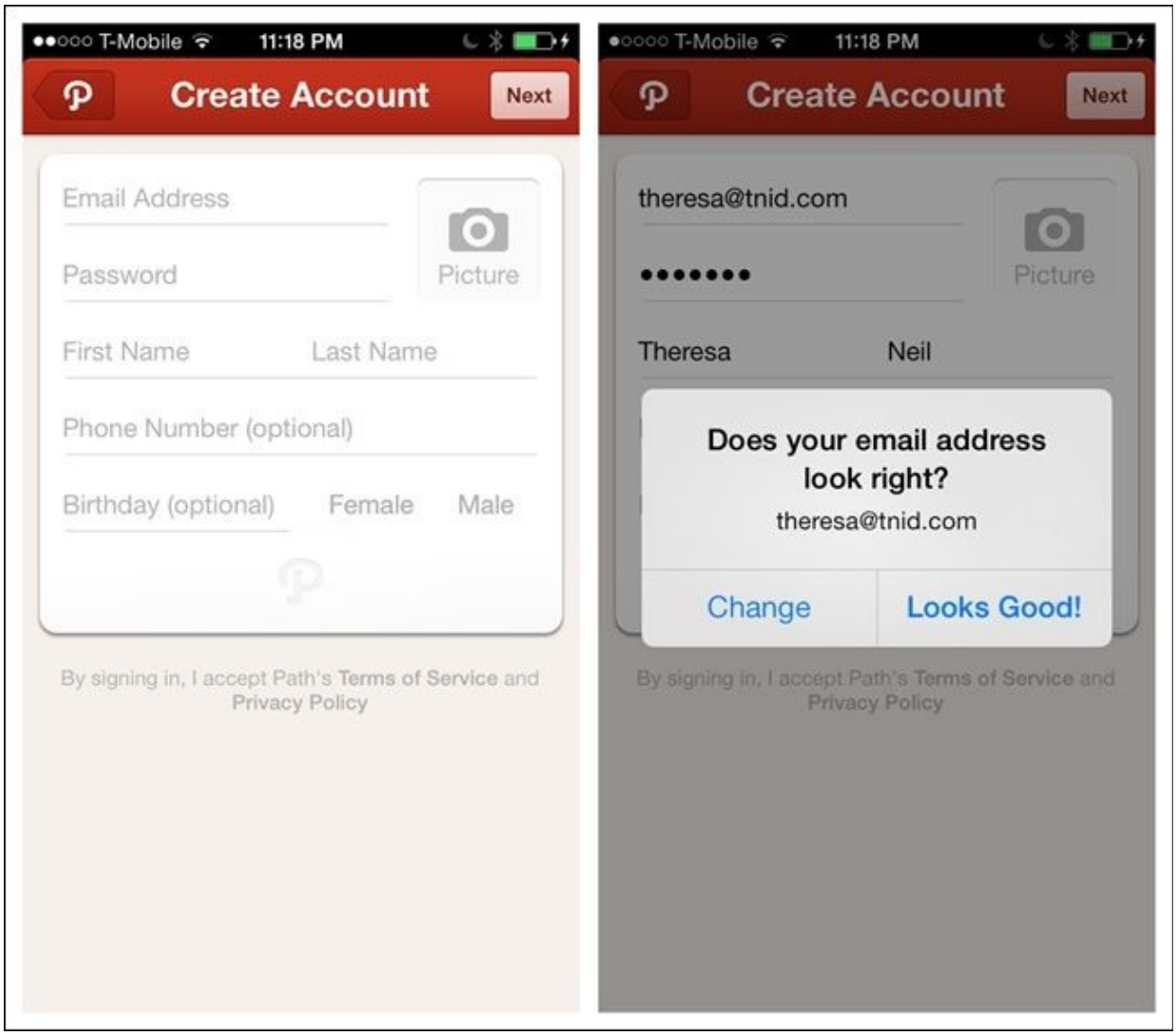



*Figure 2-14. Path for iOS clearly marks optional fields and uses a prompt to confirm email address*

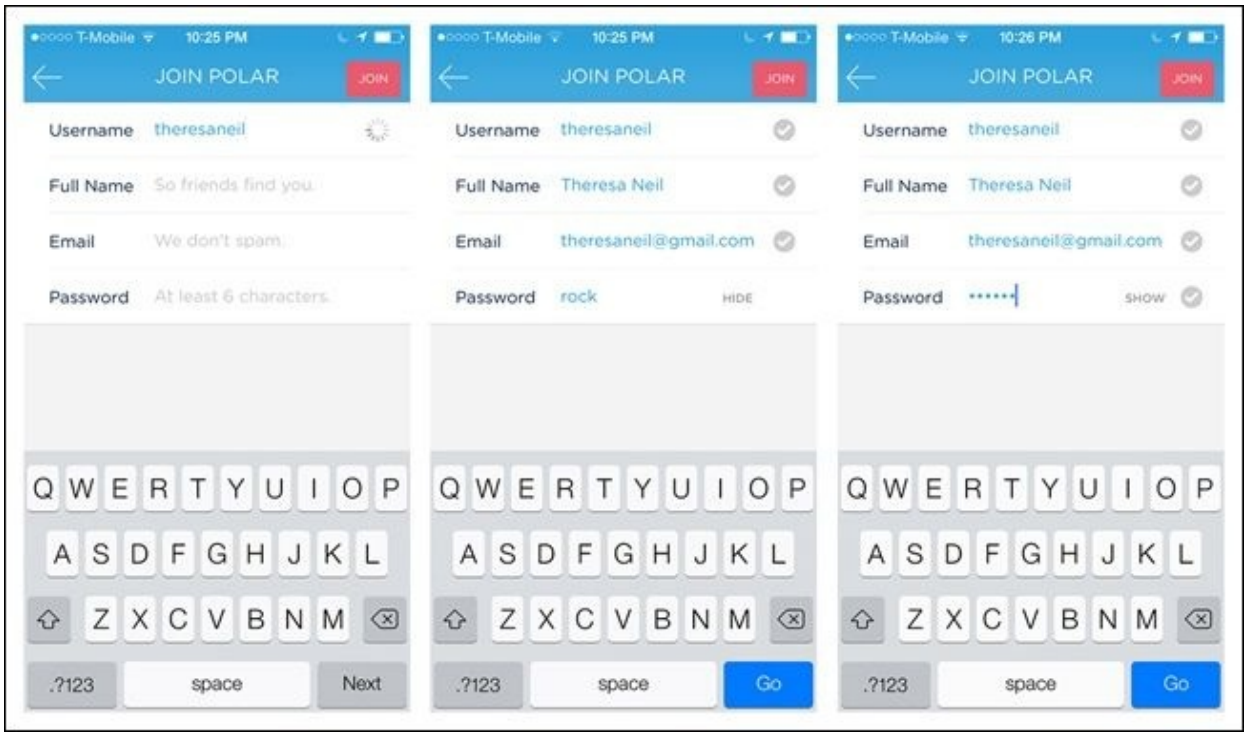



*Figure 2-15. Polar for iOS is a good example of form layout and interaction design*

Horizontally aligned form labels might not be the best choice, since they can end up truncated or simply cut into the space needed to enter the value. Consider vertically aligned labels, like in Fancy and Remember The Milk.


*Figure 2-16. Horizontal labels can end up truncated or overlapped*

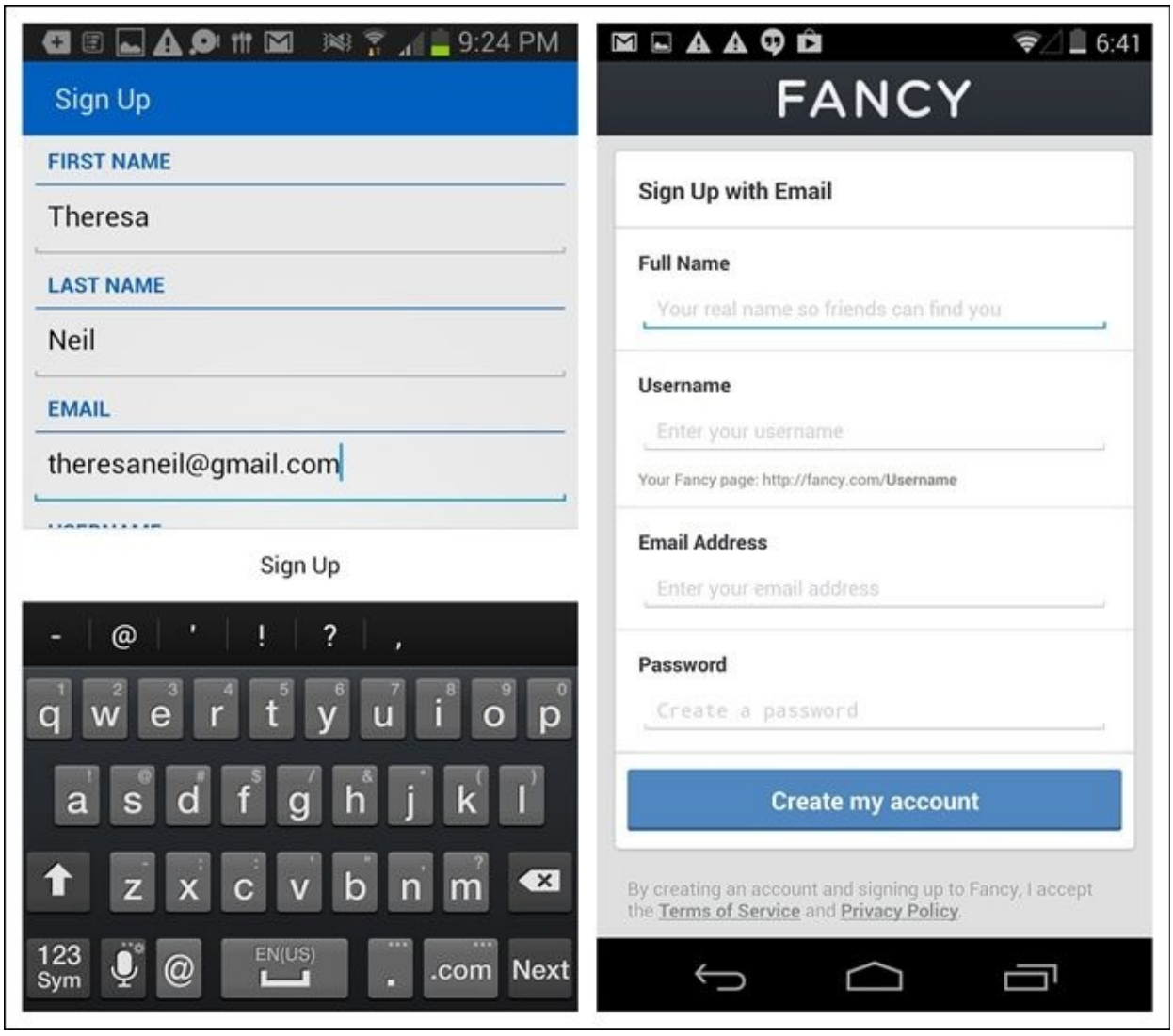



*Figure 2-17. Remember The Milk and Fancy for Android: vertically aligned labels often work better*

Another option is to use *hint text* (i.e., faint text prepopulated in the field that prompts users for the required info). But this can create a problem, because it disappears once the user starts entering text. A distracted user can easily forget what she's supposed to enter in that field. The Float Label is an emerging pattern that may address some of the usability problems around watermark labels. See more on this from Matt Smith, the pattern's creator, at *http://mattdsmith.com/float-label-pattern/*.

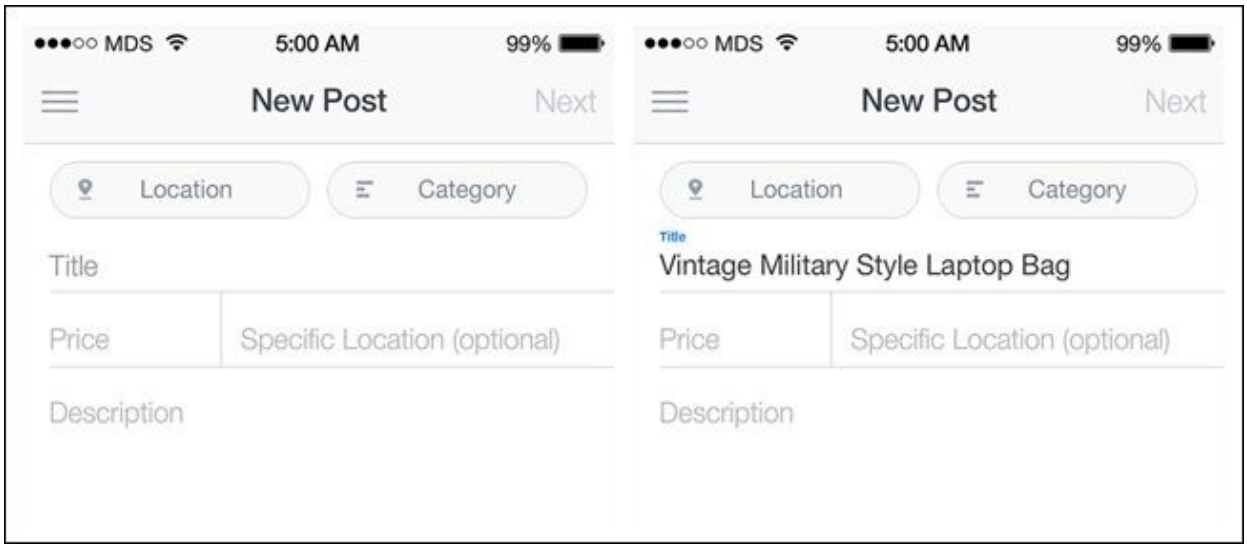



*Figure 2-18. With the Float Label pattern, watermarks become labels "floating" above the field once text is entered*

Another good practice is to offer inline feedback wherever possible and appropriate, like Instagram does for Account Name. See Chapter 8 for more tips.

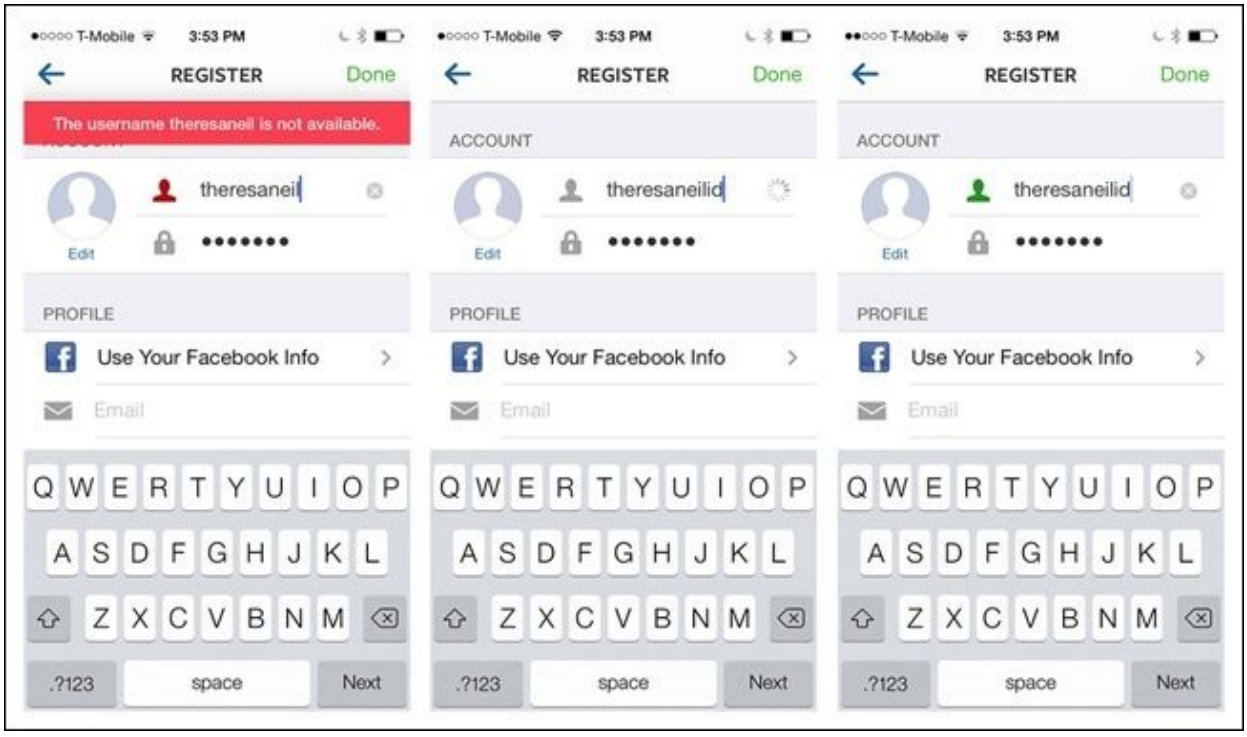



*Figure 2-19. Instagram for iOS: real-time inline feedback*

Will you have certain requirements for password security, like upper-and lowercase letters and so forth? Tell users that *before* they tap the Submit or Done button, not in an error message afterward. HelloWallet's instructions are bit

much; Level is more succinct (although it could improve its registration experience overall, in my opinion).

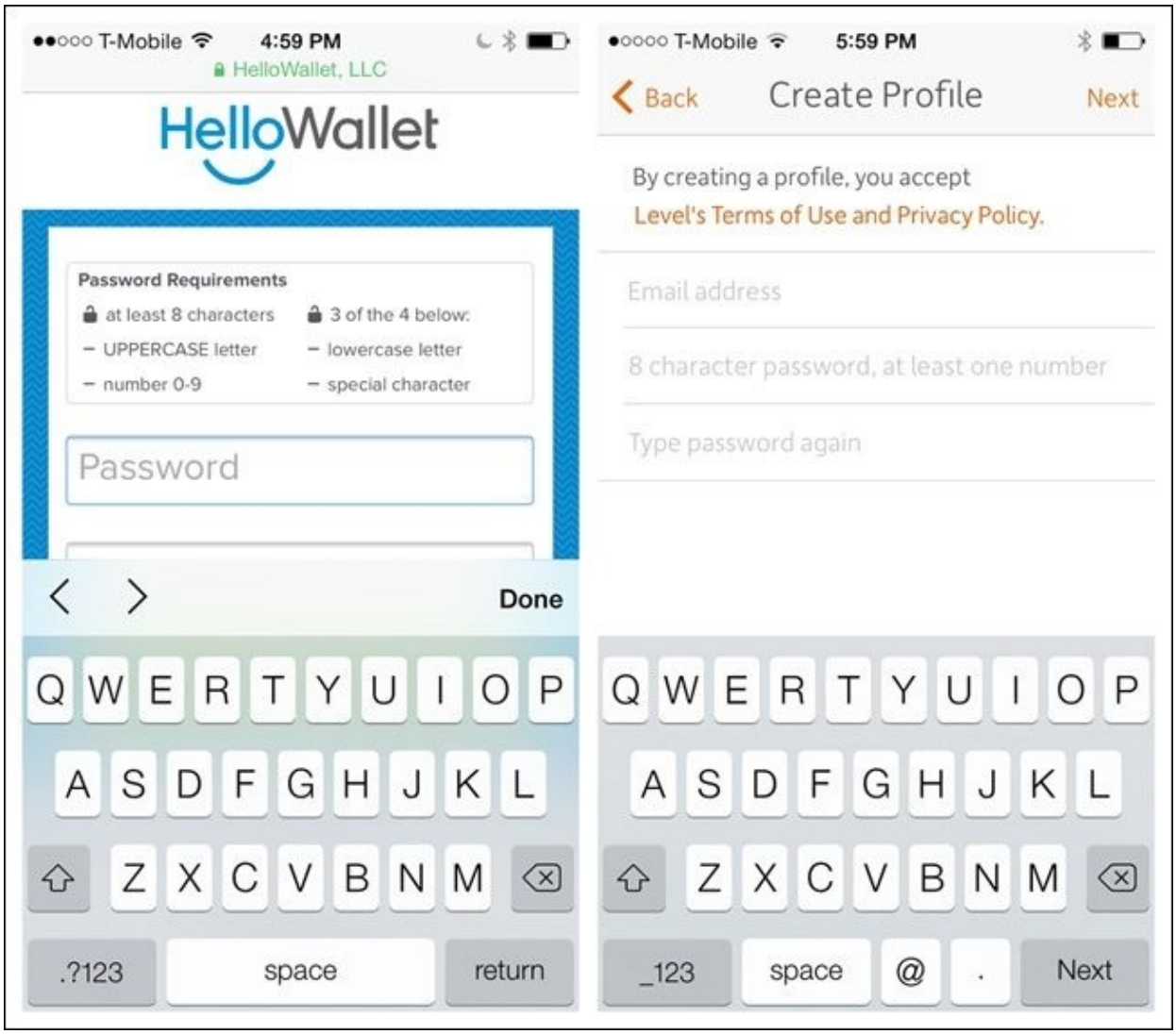



*Figure 2-20. HelloWallet and Level for iOS: show password requirements up front*

And once the user does submit a Registration form, offer feedback on progress, like in the Etsy and Pageonce apps for Android.

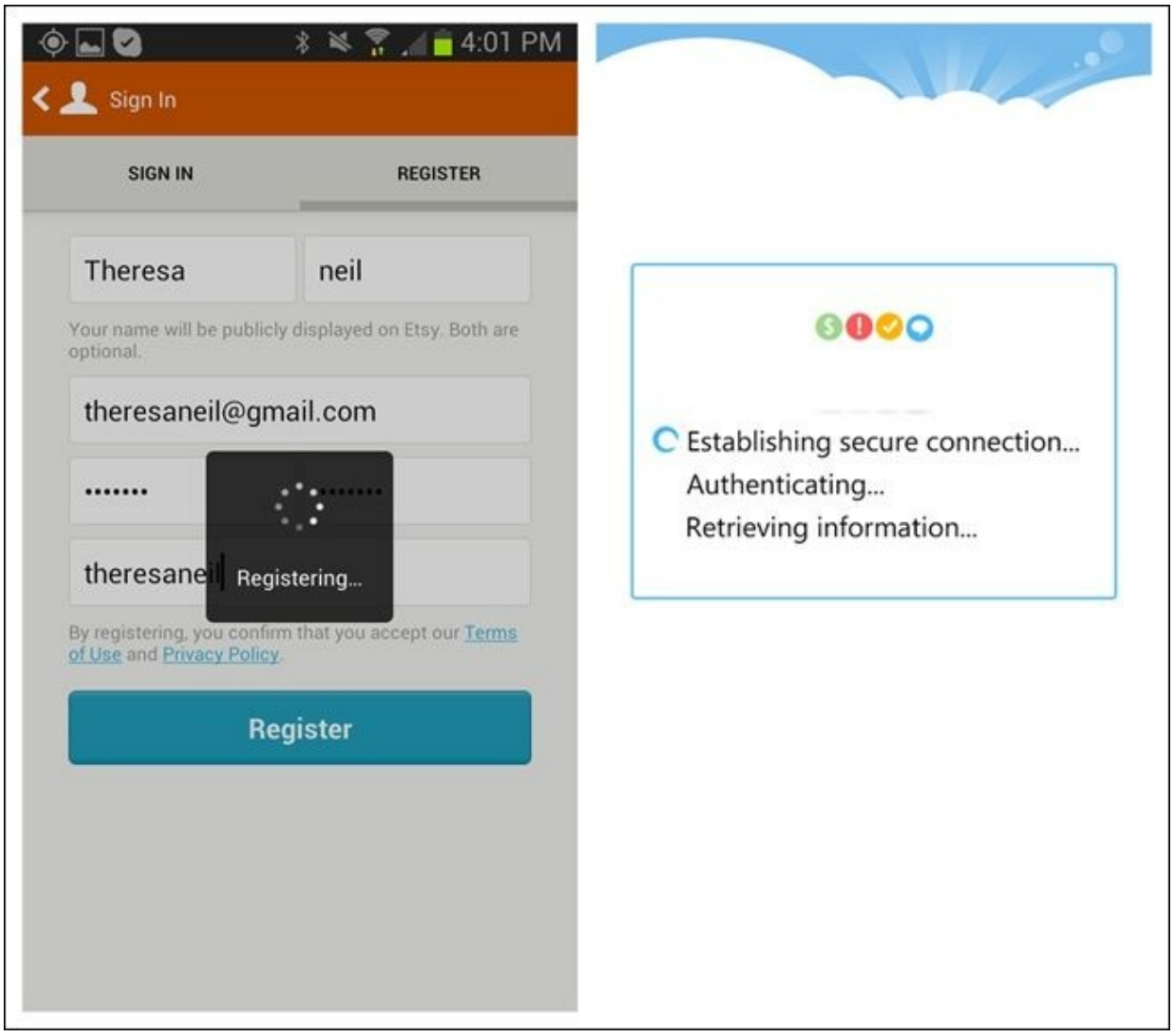



*Figure 2-21. Etsy and Pageonce for Android: once the form is submitted, offer the user feedback on progress*

During the registration process, keep the user within the application if at all possible. If you must link out to a form on a web page, at least test the form in mobile browsers to make sure it is usable. The designers of Fitbit for Windows Phone apparently didn't do this.

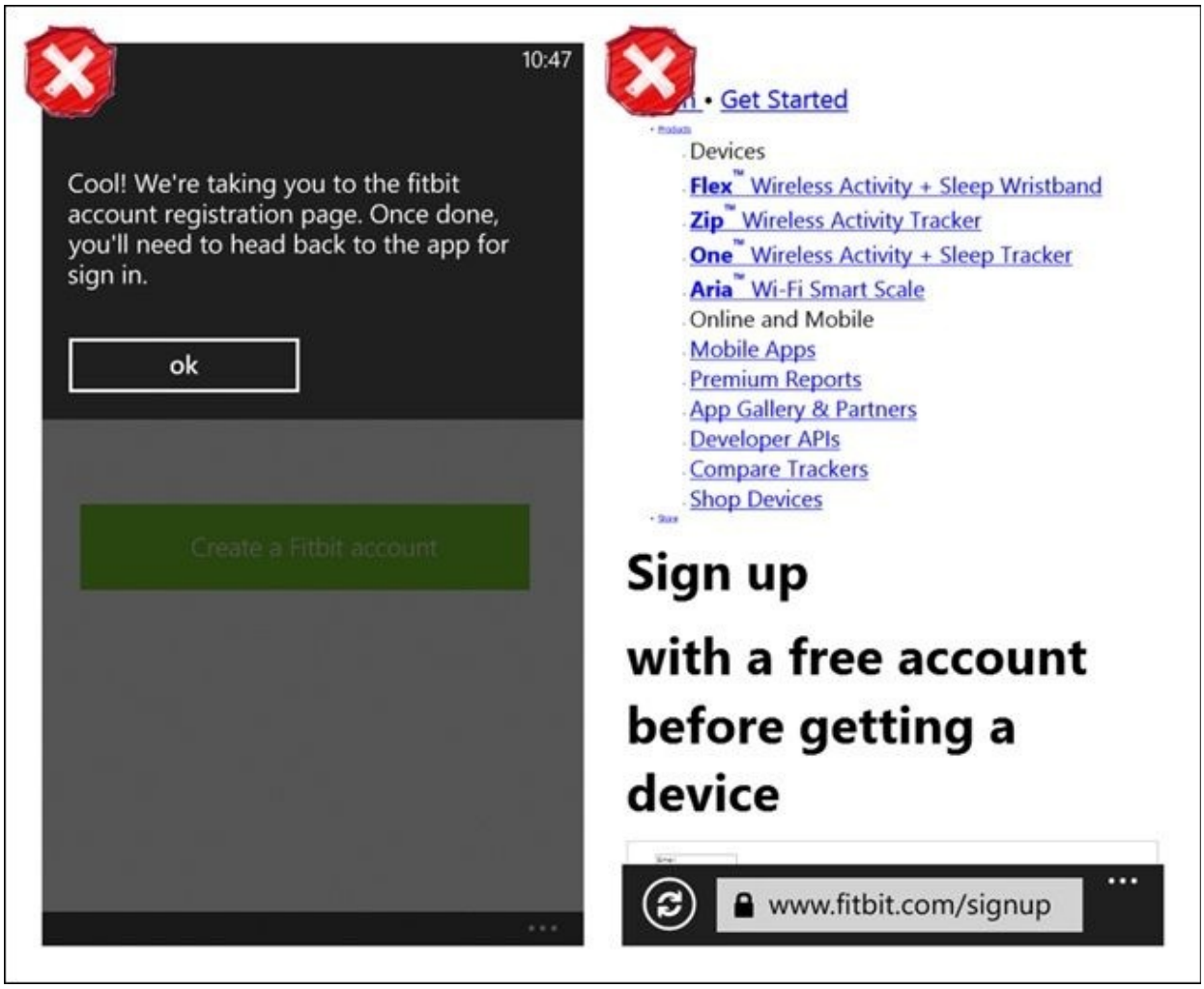



*Figure 2-22. Fitbit for Windows Phone: miserable browser-based registration UX*

Think of registration as a flow, and account for all possible user inputs. In another example from Level, I "accidentally" filled out the Registration form again, even though I already had an account. Even though there was a checkmark next to my email address — which would apparently indicate that it was acceptable — I received an error message because the email address was already in use. My only option was to tap "Bummer."

Why not tell me the email address is already in use when I enter it in the form field? Or how about at least redirecting me to the Sign In screen with my email address prepopulated instead of serving me a lame error message that gives me no clue as to what step to take next?

**NOTE**

Keep Registration forms short, preferably one screen, with the command button in plain sight. Clearly state field entry requirements and use inline feedback to speed up the process and let the user know

what's going on. And account for all user scenarios.

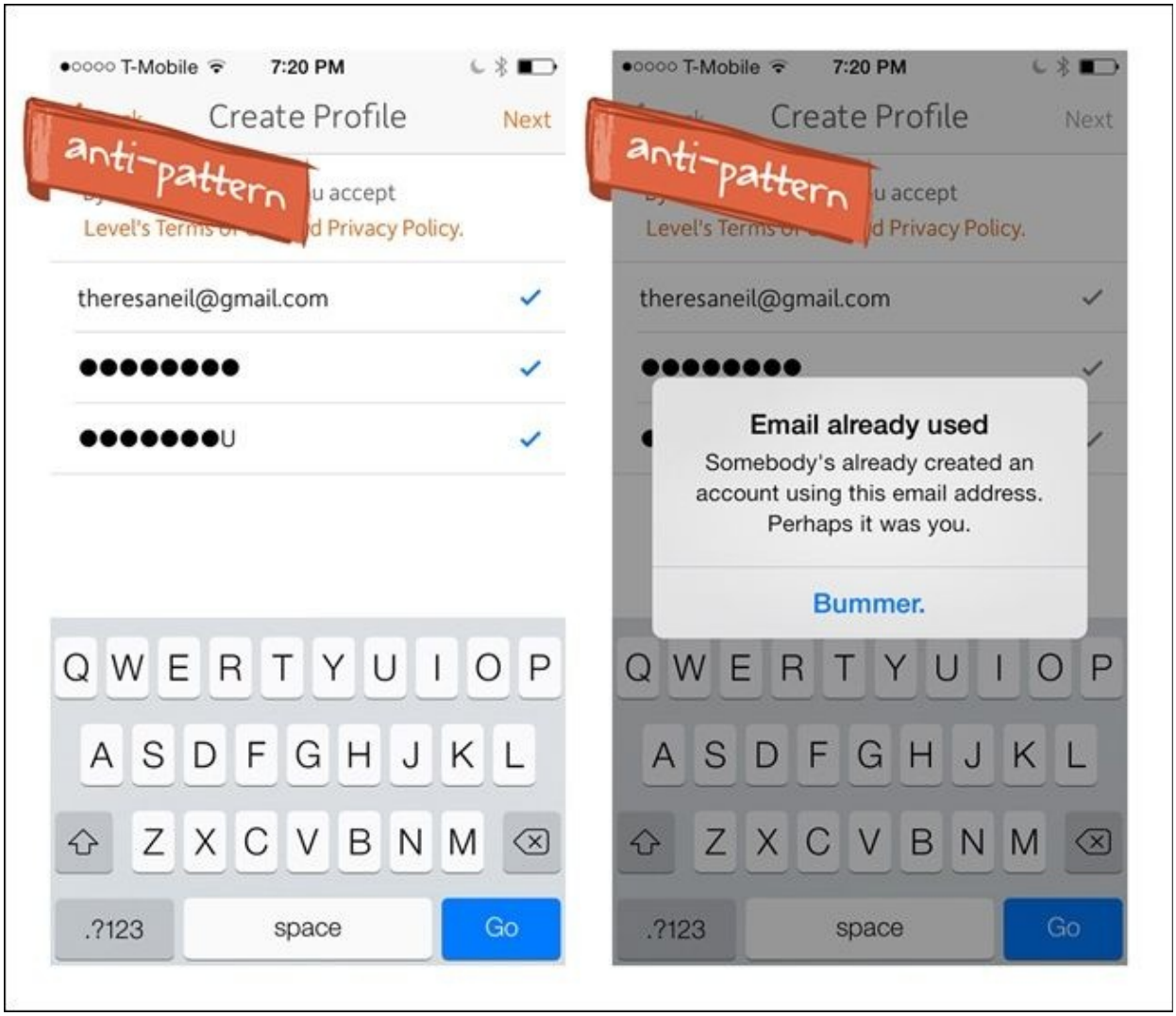



*Figure 2-23. Level for iOS: my email address has a checkmark by it, even though it's already in use*

Dropbox offers an elegant solution for this same scenario. Once it recognizes my email address it assumes that I would like to sign in, not reregister. Since I most likely need recovery instructions for my password, it prepopulates my email address on the recovery screen, so I only have to tap Send and I'm on my way.

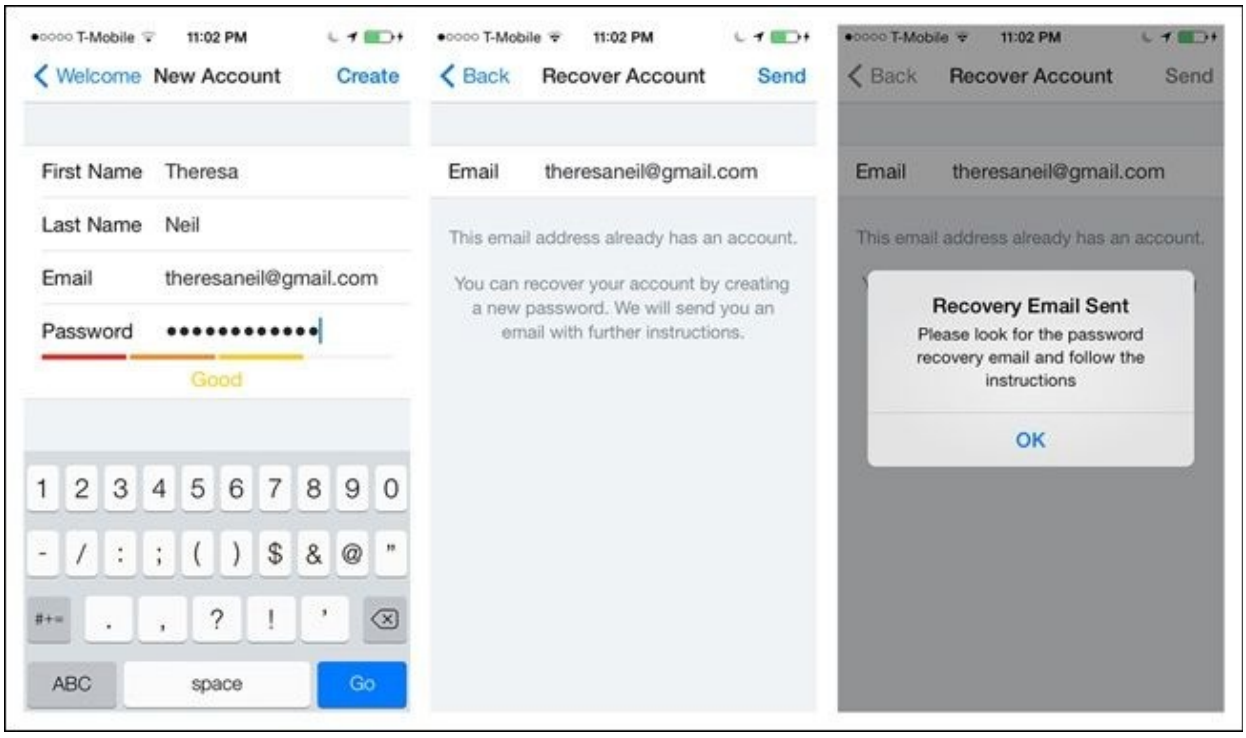



*Figure 2-24. Dropbox for iOS: elegant solution for the duplicate registration scenario*

## Registration with Personalization

This is a relatively new pattern designed to either offer more personalized content to the user (as shown with Ness and News360), enhance the experience through social connections (Path and Instagram), or capture data required for setup (Personal Capital).

The Multi-Step pattern we'll look at next offers examples of more elaborate Registration flows.

Facebook's recent Paper app veers toward the Needless Complexity anti-pattern (see Chapter 11), with its drag-and-drop UI for selecting topics of interest. Swiping and tapping like shown in Ness and News360 is more efficient and very easy to do with just a thumb. For more on that concept, see Luke Wroblewski's article "Designing for Thumb Flow" (*http://www.lukew.com/ff/entry.asp?1734=*).

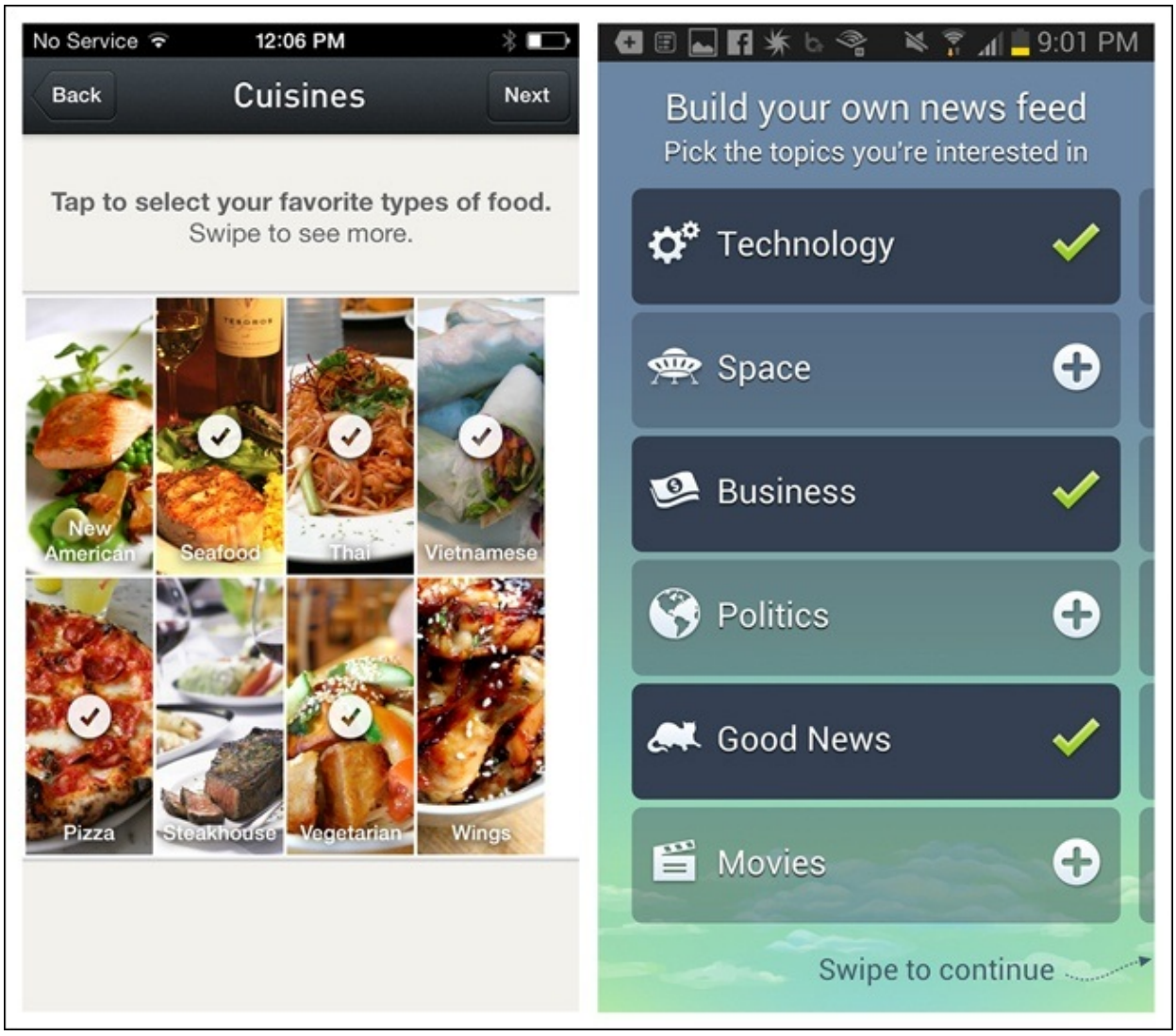



*Figure 2-25. Ness for iOS and News360 for Android: prompt for preferences in the Registration flow*

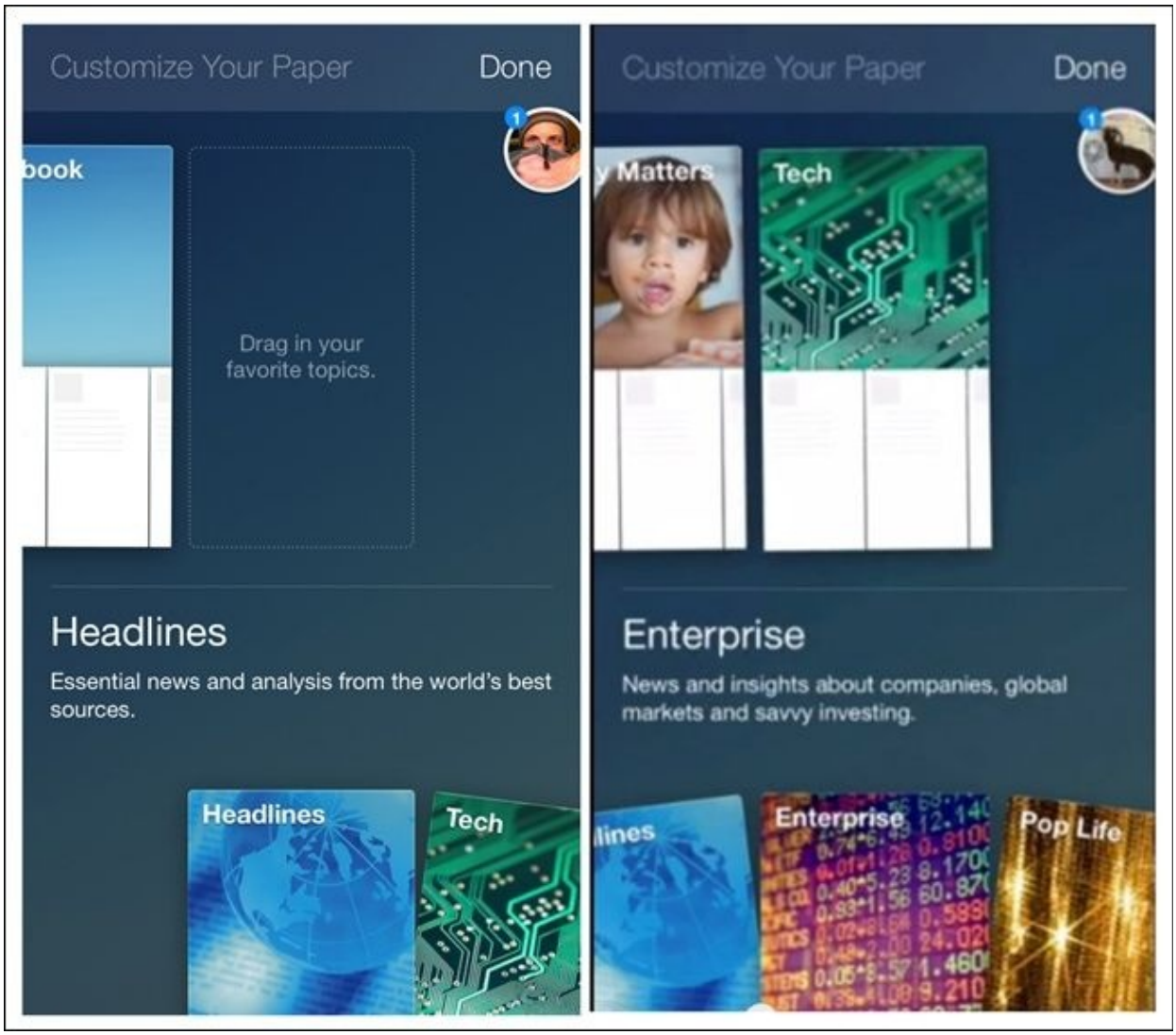



*Figure 2-26. Facebook's Paper for iOS: drag-and-drop UI requires extra user effort (http://youtu.be/UpZFw2wQr58)*

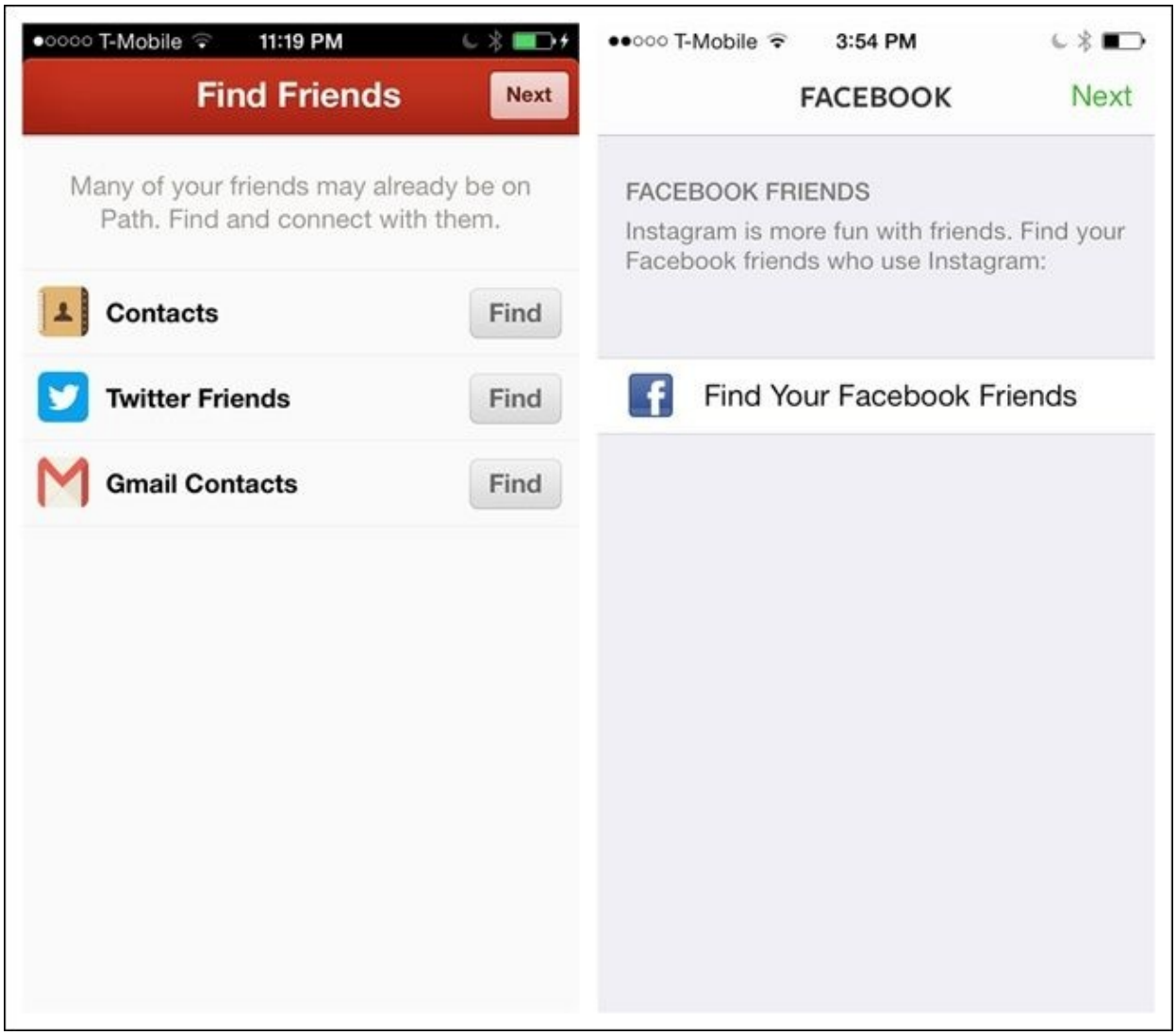



*Figure 2-27. Path and Instagram for iOS: encourage friend connections in the Registration flow*

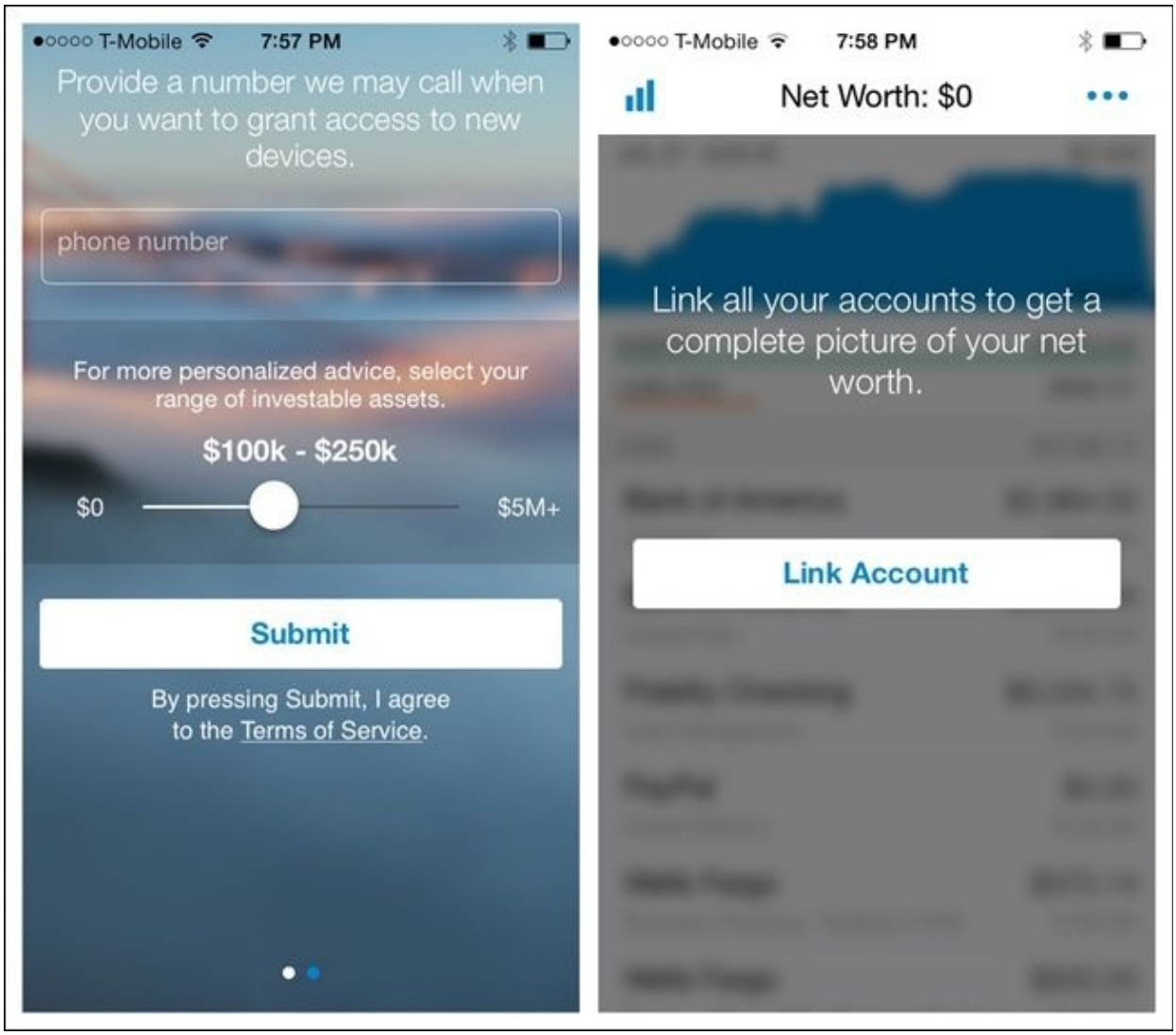



*Figure 2-28. Personal Capital for iOS: prompts for phone number and asset range and provides an opportunity to link accounts*

## Multi-Step

On the Web, Multi-Step registration processes often use process bars to show users where they are and what's next. But with their smaller screens, mobile devices just don't have the real estate to display bulky process bars. You might be able to cram one in the title bar, but there are better ways to guide mobile users through a series of steps.

For an early version of the PayPal iOS app, stakeholders insisted on a process bar. Sending money was a process, after all. But compare the 2010–2011 version with the current, much cleaner evolution.


*Figure 2-29. PayPal for iOS, circa 2010–2011: Send Money process bar*

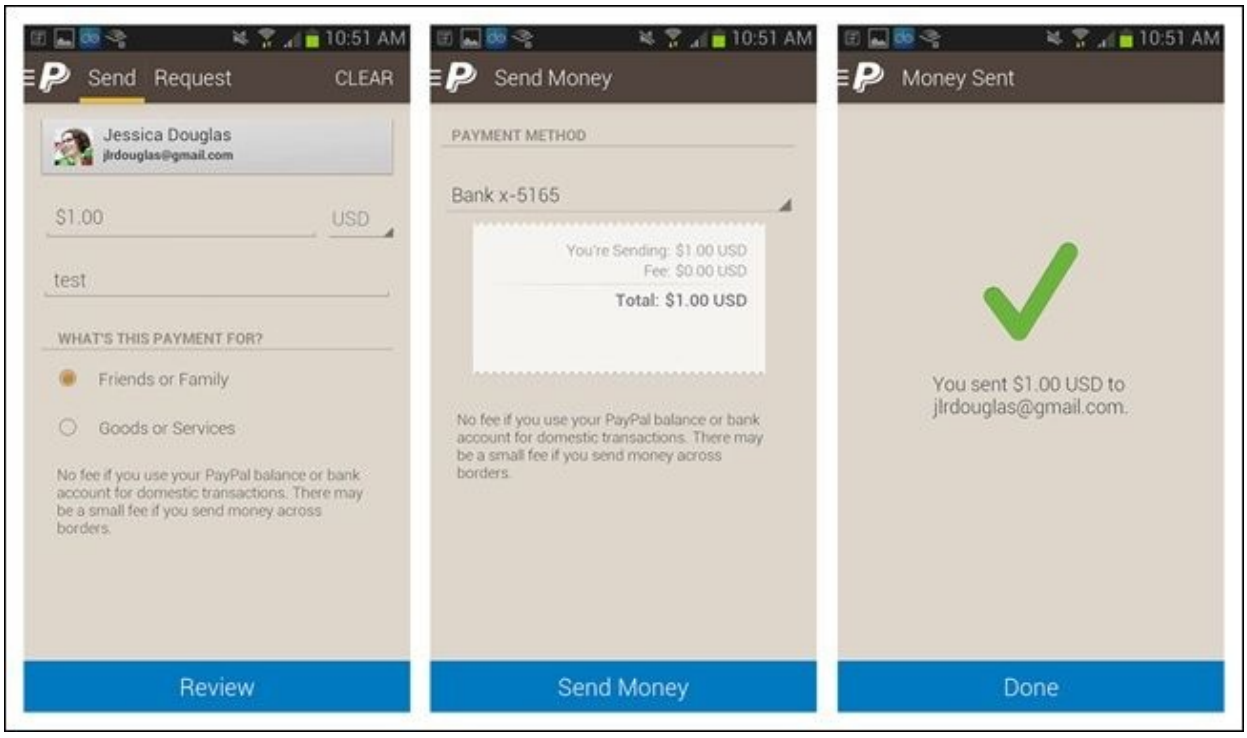



*Figure 2-30. PayPal circa 2013: losing the process bar is part of a cleaner overall look*

If steps have to be broken across multiple screens, skip the heavy process bar and consider displaying the current step among the total number of steps, like SnipSnap (displayed in the title bar) and Priceline (displayed in the footer).

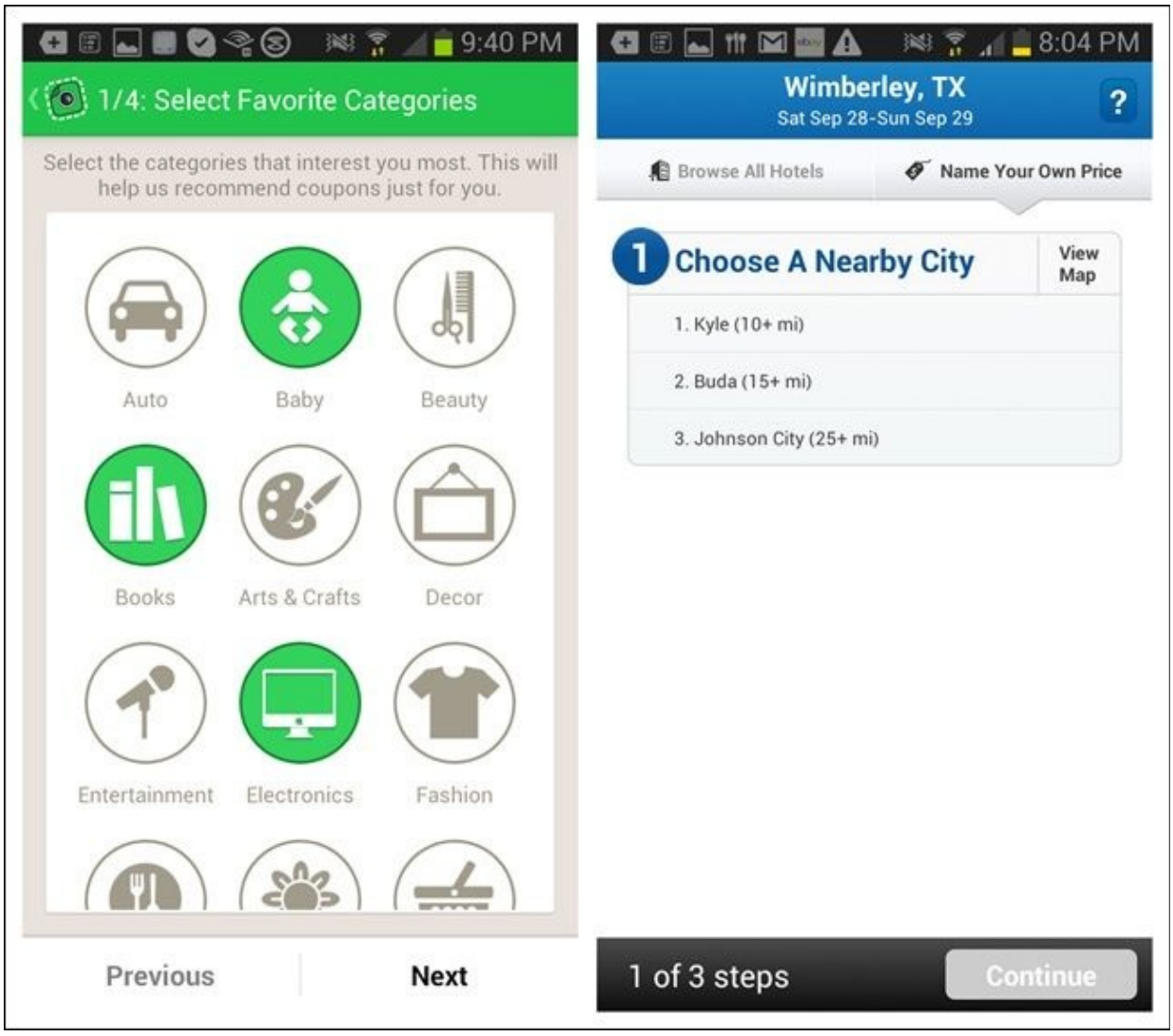



*Figure 2-31. SnipSnap and Priceline for Android: indicate the current step among the total number*

In the first edition of the book, Home Depot's checkout process was included as an anti-pattern. The process bar was crowded with tiny text, and the checkout experience felt tedious. But it now offers a much-improved single-screen design.

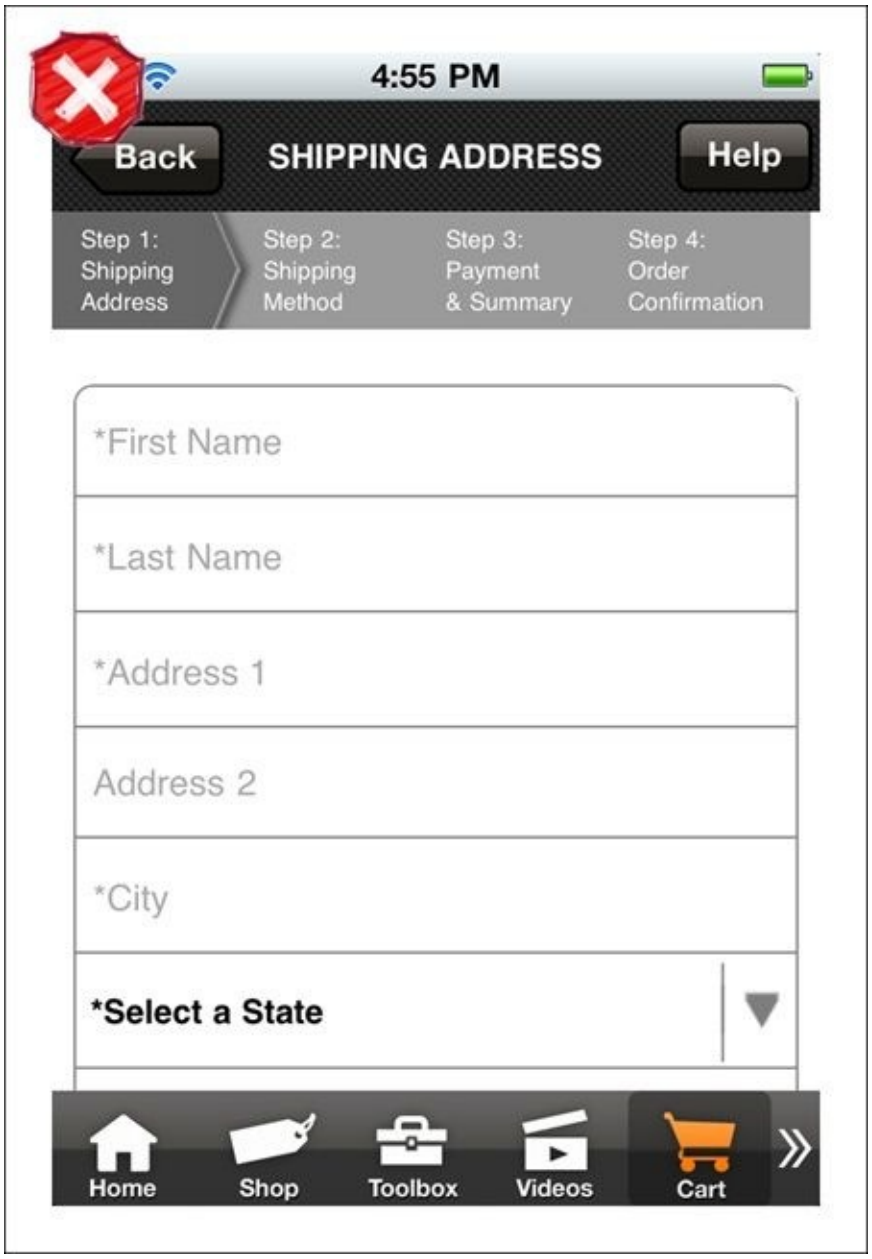



*Figure 2-32. Home Depot for iOS (2011): overcrowded process bar*

The secure checkout process first captures the delivery address. Upon tapping Next, you then select the delivery date. The delivery address is still shown, but in a read-only mode, with an Edit button.

Once the delivery date is selected, tapping Next reveals the payment form. Upon completing the payment form, you can review the total, and all of the information entered previously — while never leaving this screen — and proceed to the next step.

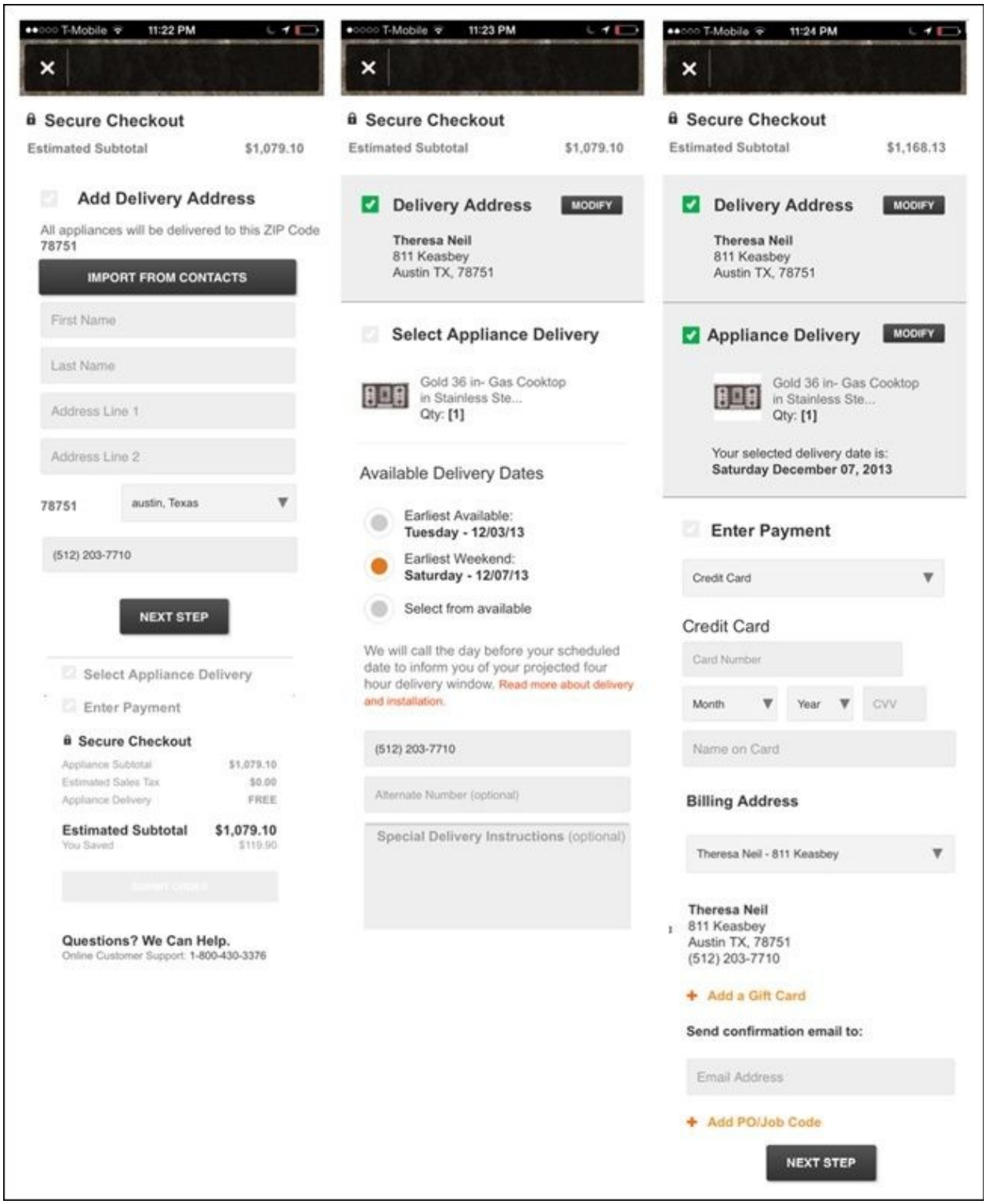



*Figure 2-33. Home Depot for iOS (2013): much-improved checkout flow*

TurboTax SnapTax uses a similar approach. However, the details for each step are collected in a drill-down screen instead of inline on the same screen. This

single-page approach provides valuable navigation information (there are three steps in the process, two of which are remaining), and it also doubles as a high-level summary.

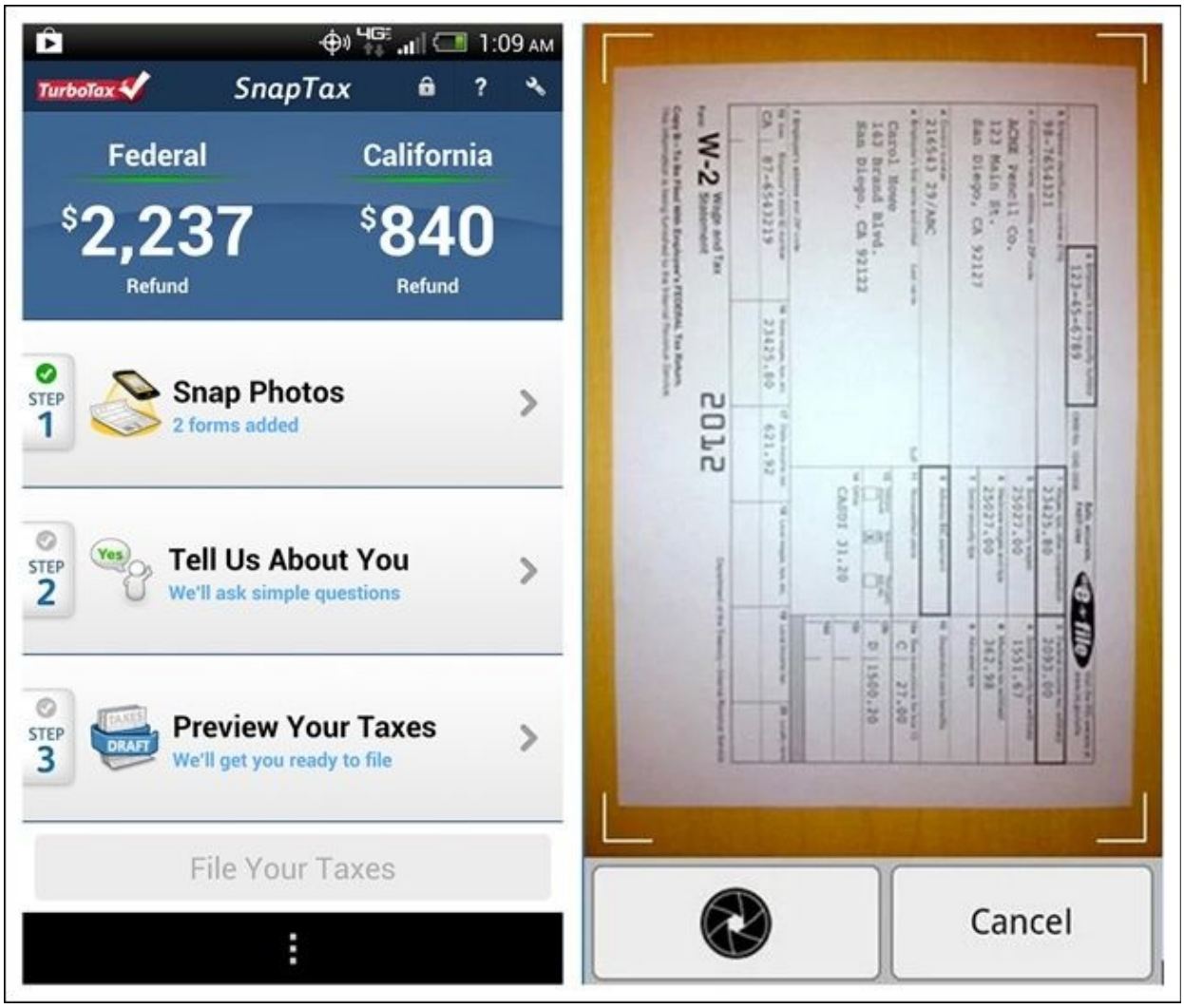



*Figure 2-34. Intuit's TurboTax SnapTax for Android and iOS: drill down to the details*

Product configurators are a unique type of Multi-Step form that may need to allow for back-and-forth navigation, not just the typical linear flow. To enable this in its Drink Builder tool, Starbucks just uses Next-and Back-style buttons.

This works fine, but it is hard to see what's been specified in the customization screens without opening every one of the options and scrolling through the list.

A simple fix could be a persistent Summary or Details button in the footer, near the Nutrition tab and Info icon. The Audi Configurator does exactly that, with a Details button that opens to reveal all of the selected preferences so far.

**NOTE**

Show users where they are and where they can go, but skip the bulky process bar. Eliminate unnecessary fields and minimize the number of pages and steps.

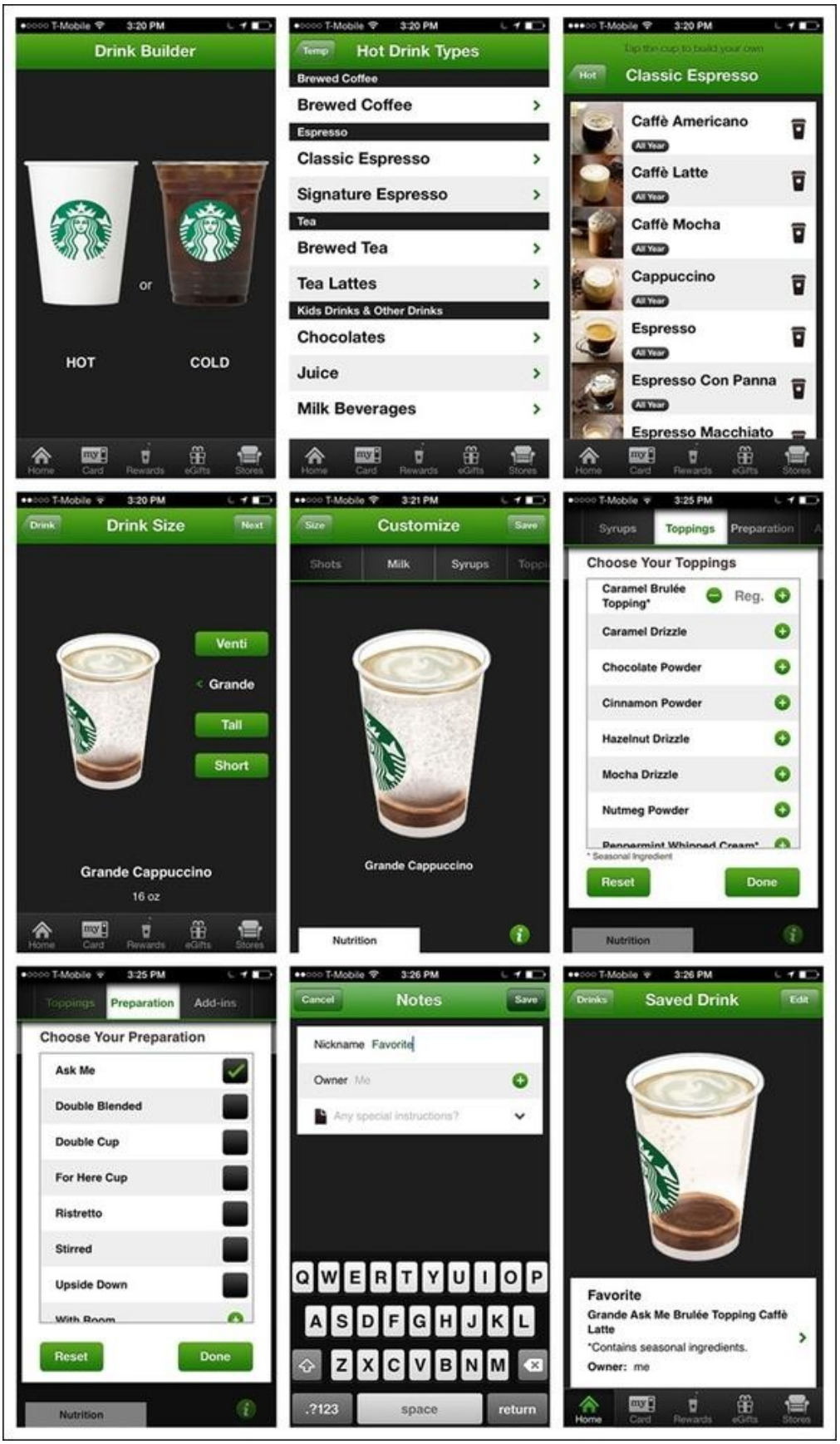

Drink Builder
or
HOT
COLD
Hot Drink Types
Brewed Coffee
Brewed Coffee
Espresso
Classic Espresso
Signature Espresso
Tea
Brewed Tea
Tea Lattes
Kids Drinks & Other Drinks
Chocolates
Juice
Milk Beverages
Classic Espresso
Caffè Americano
Caffè Latte
Caffè Mocha
Cappuccino
Espresso
Espresso Con Panna
Espresso Macchiato
Drink Size
Venti
Grande
Tall
Short
Grande Cappuccino
16 oz
Customize
Shots
Milk
Syrups
Grande Cappuccino
Nutrition
Choose Your Toppings
Caramel Brulée Topping*
Reg.
Caramel Drizzle
Chocolate Powder
Cinnamon Powder
Hazelnut Drizzle
Mocha Drizzle
Nutmeg Powder
* Seasonal Ingredient
Reset
Done
Choose Your Preparation
Ask Me
Double Blended
Double Cup
For Here Cup
Ristretto
Stirred
Upside Down
Notes
Nickname Favorite
Owner Me
Any special instructions?
Saved Drink
Favorite
Grande Ask Me Brulée Topping Caffè Latte
*Contains seasonal ingredients.
Owner: me

*Figure 2-35. Starbucks Drink Builder for iOS: Next-and Back-style buttons (http://www.youtube.com/watch?v=Bvke0dXD2SM)*

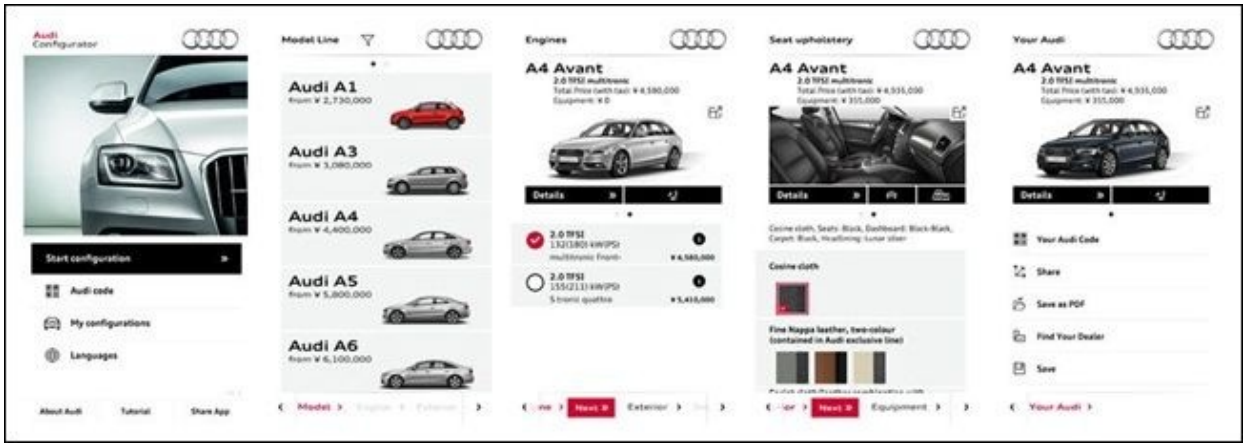



*Figure 2-36. Audi Configurator for iOS: Persistent Details button reveals selected options*

# Checkout

As of January 2013, about 15% of ecommerce was being conducted via mobile devices, with the expectation that this will exceed 25% in a couple of years. However, even when the shopping experience starts on the phone, most people choose to finish the transaction, or checkout, via desktop or tablet or at the retailer.

In many cases, this is likely due to the incredibly bad checkout experiences many retailers include in their native apps. In others, it's because many retailers only let users shop in their apps and redirect them to a mobile-optimized site for checkout.

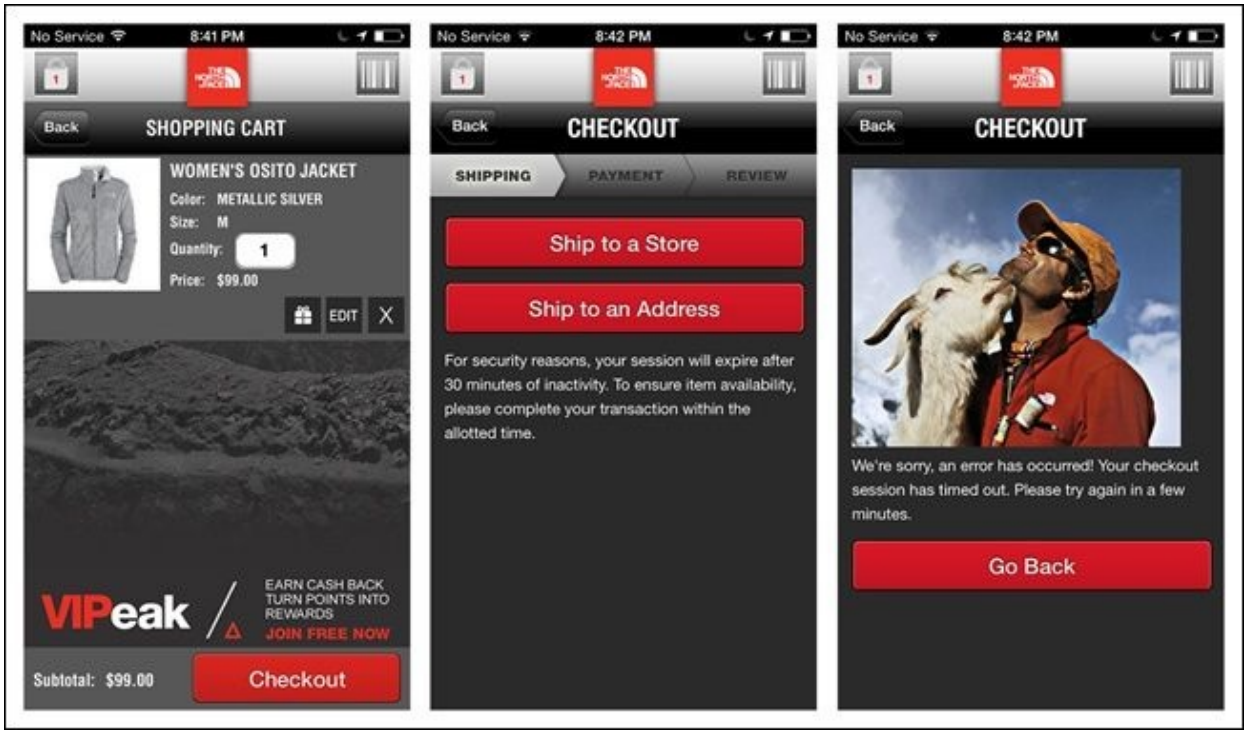



*Figure 2-37. The North Face for iOS times out when I try to check out — every time! Grr! Why have a timeout anyway?*

Help your users buy right where they are, right when their interest is highest. Don't let a poor UX deter your customers from checking out on their phones, preferably right within the app they are shopping from. Here are some design tips for creating a high-conversion checkout flow.

## Tip #1: Include Sign In, Register, and Guest Options

Allow customers to save their account information with their profiles for faster

checkout in the future, but also offer guest checkout. You can always prompt the user to create an account after checkout is complete, when he is more comfortable with the experience.

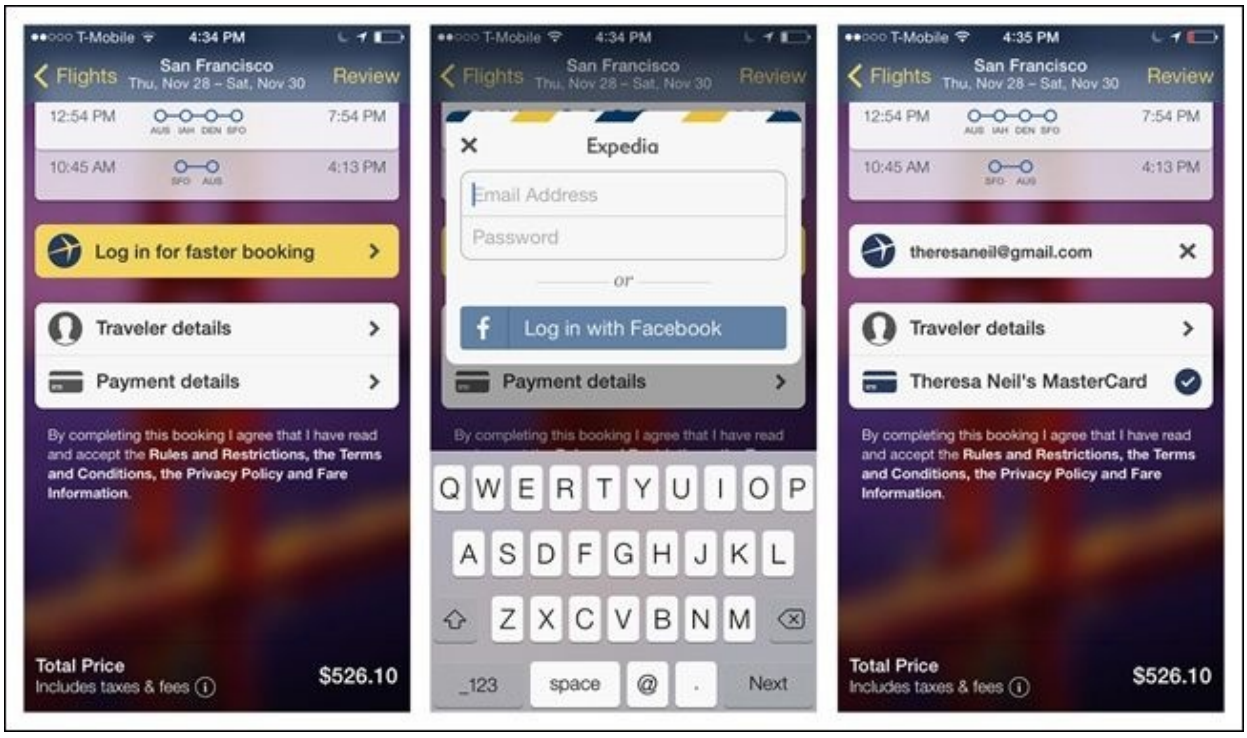



*Figure 2-38. Expedia for iOS: signing in prepopulates the checkout form with traveler and payment details*

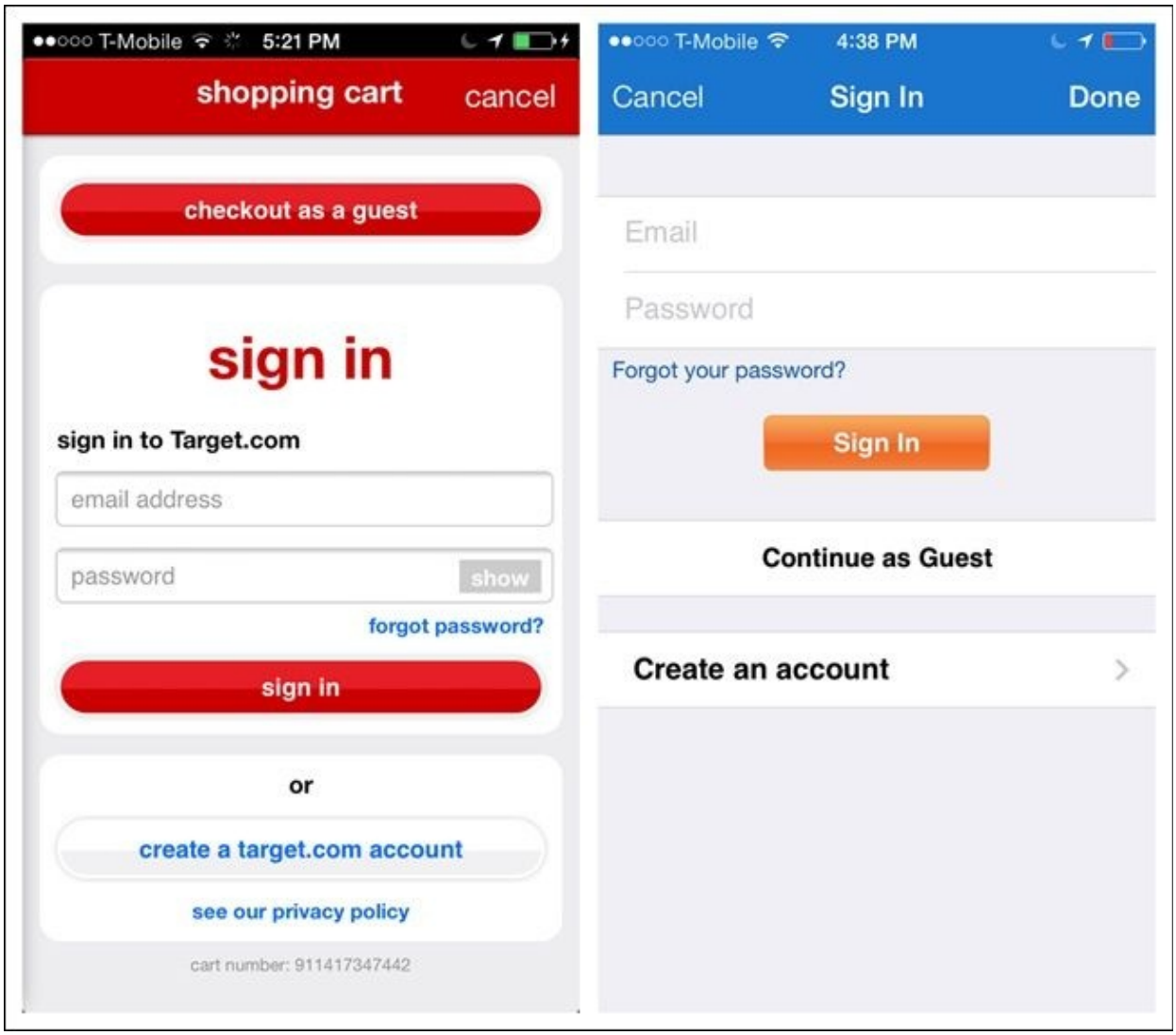



*Figure 2-39. Target and Walmart for iOS: offer guest checkout as well as Sign In/Register options*

## Tip #2: Streamline the Flow

Skip the process bar and consolidate multiple screens into one checkout page (see the earlier discussion of the Multi-Step pattern for a good example from Home Depot). Retailers like Haute and Apple present a single checkout form with sections users can drill down into to complete.

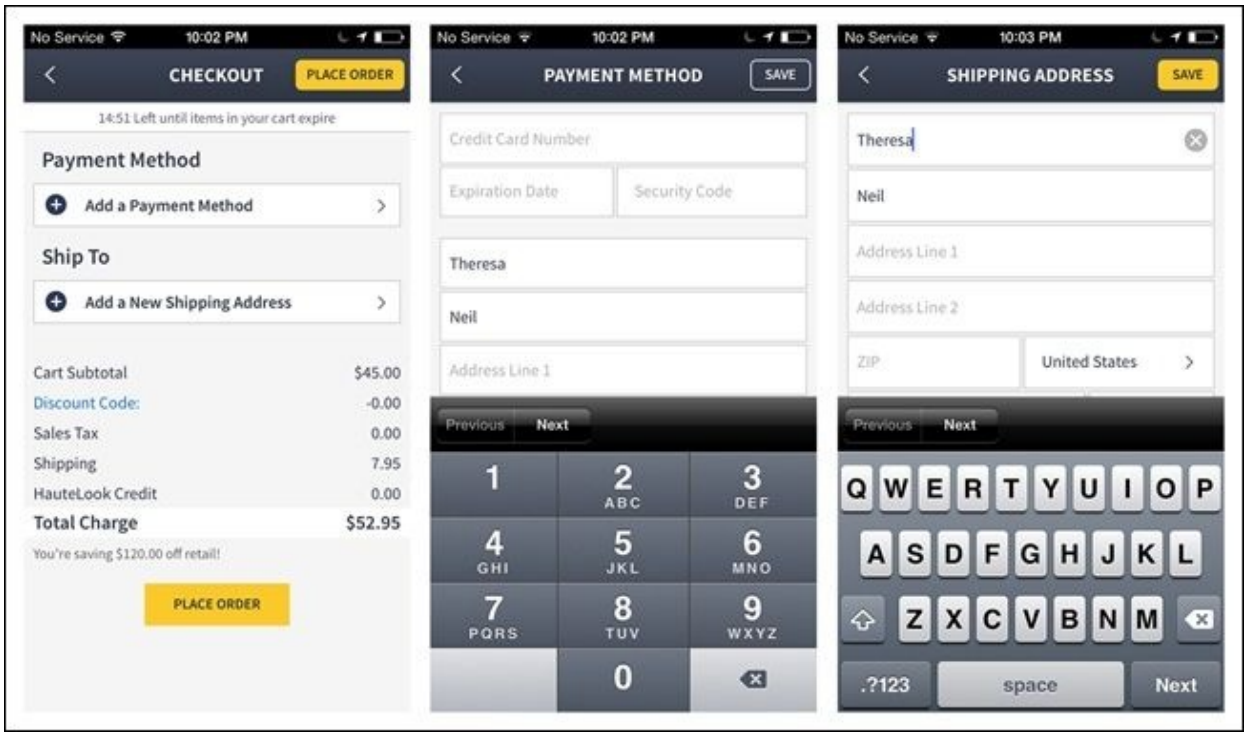



*Figure 2-40. Haute for iOS: single checkout screen with additional screens to add details*

## Tip #3: Provide Time-Saving Shortcuts

Offer an “Import Address from Contacts” option to quickly populate the user’s shipping and billing addresses. Also, consider a Scan Card feature to save the user the trouble of typing in all of her credit card info.

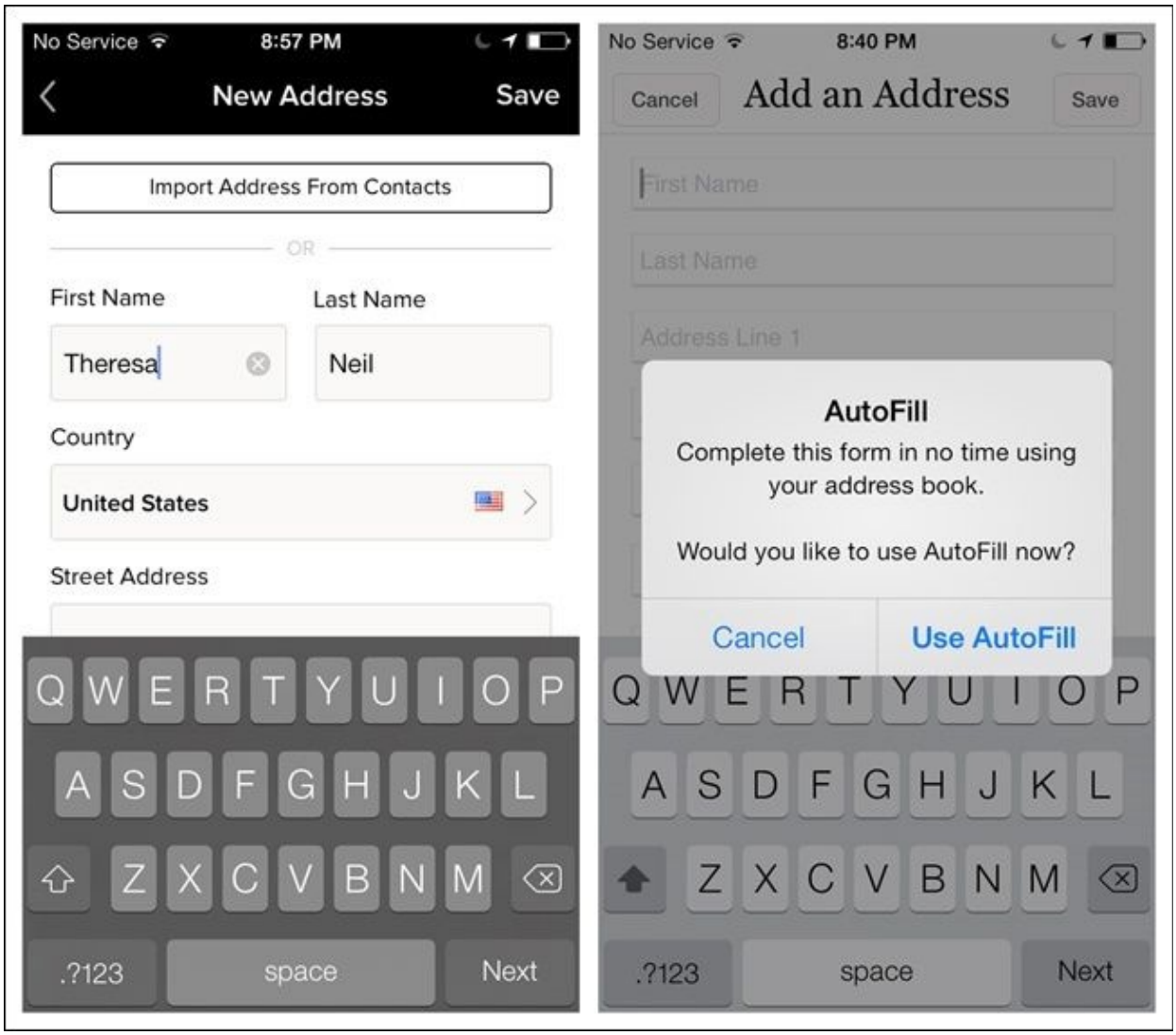



*Figure 2-41. Gilt and RuLaLa for iOS: optional import/autofill for entering addresses*

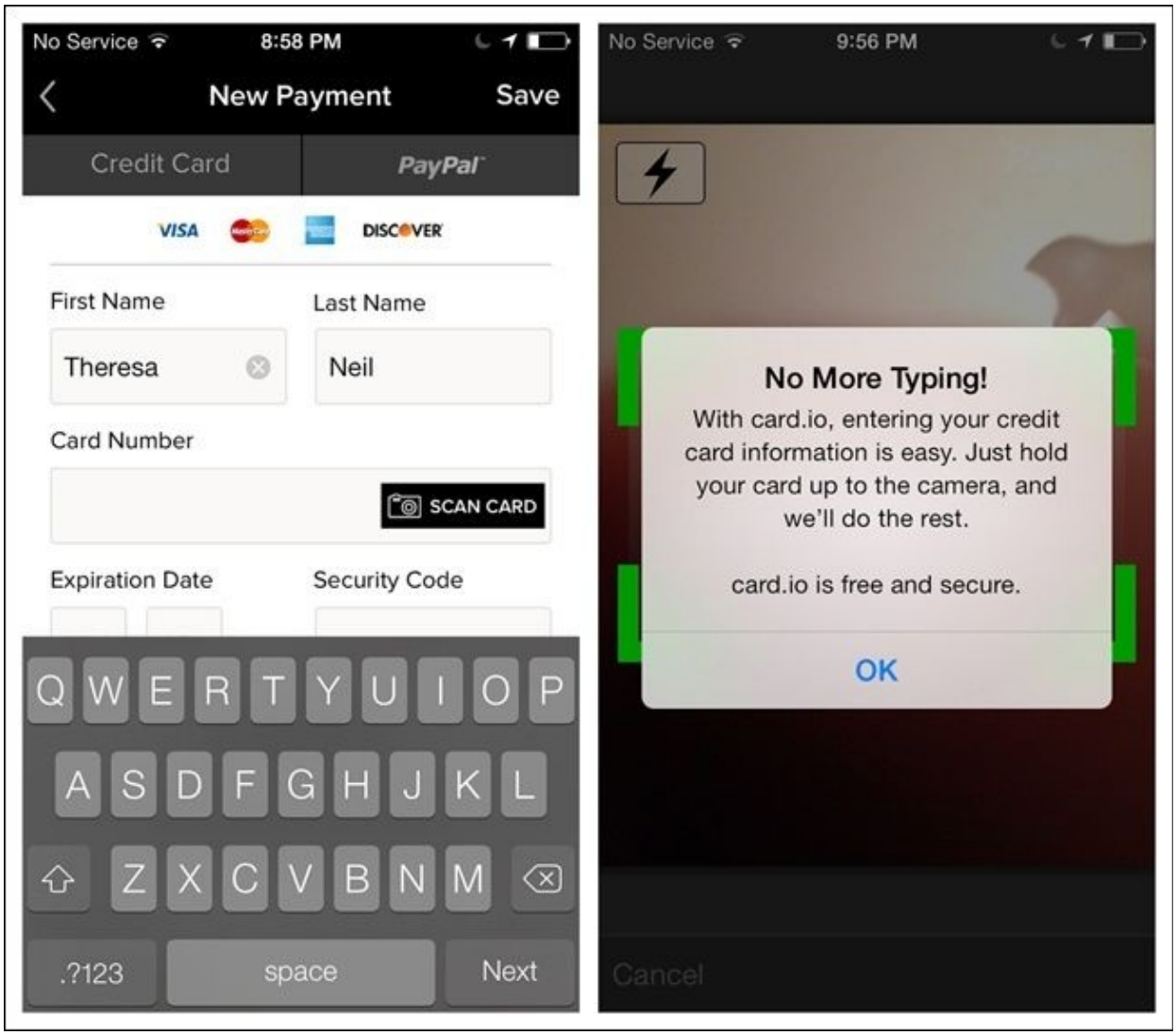



*Figure 2-42. Gilt for iOS: a Scan Card option via card.io*

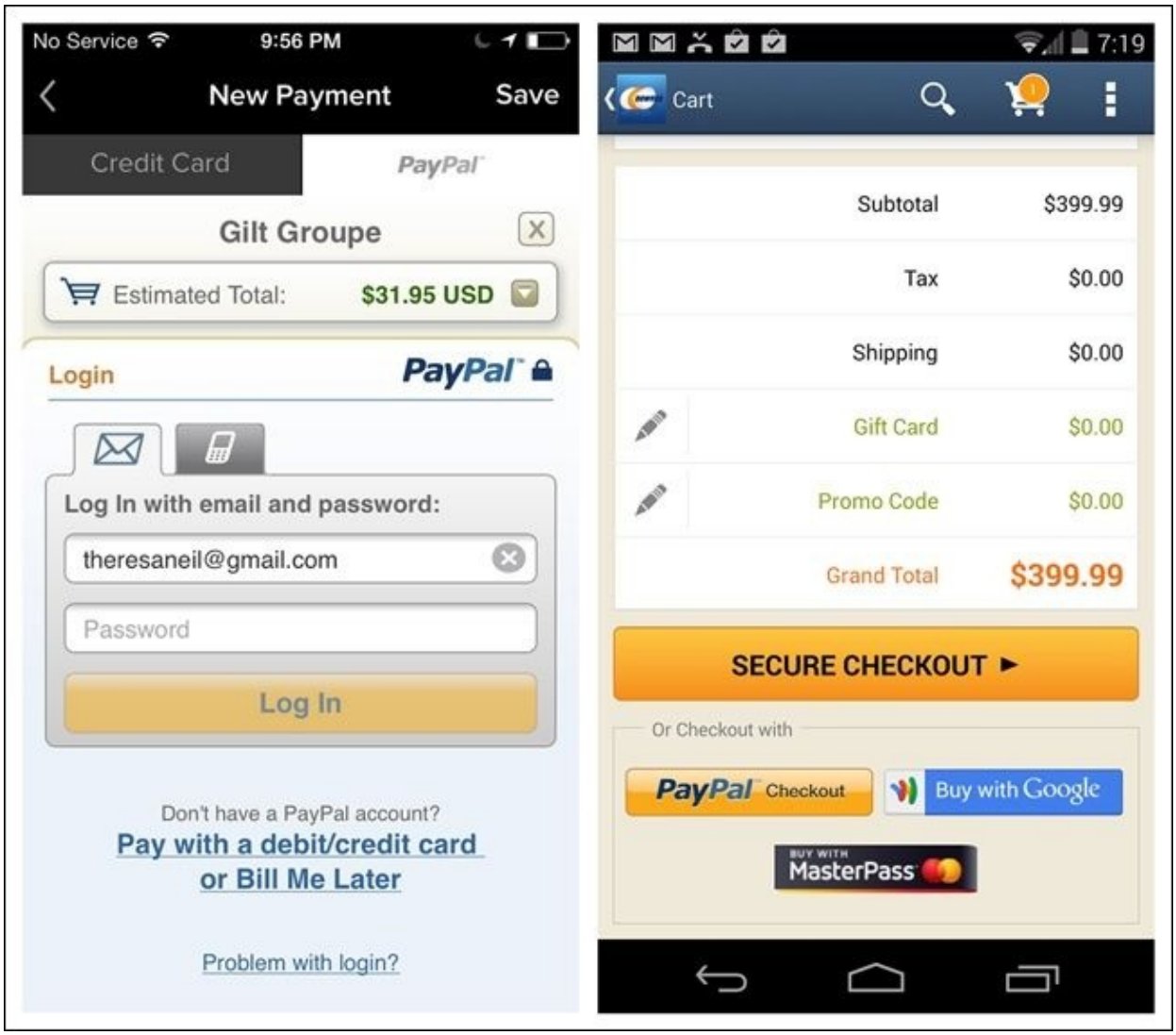



*Figure 2-43. Gilt for iOS and Newegg for Android: alternative payment options via PayPal and Google Wallet*

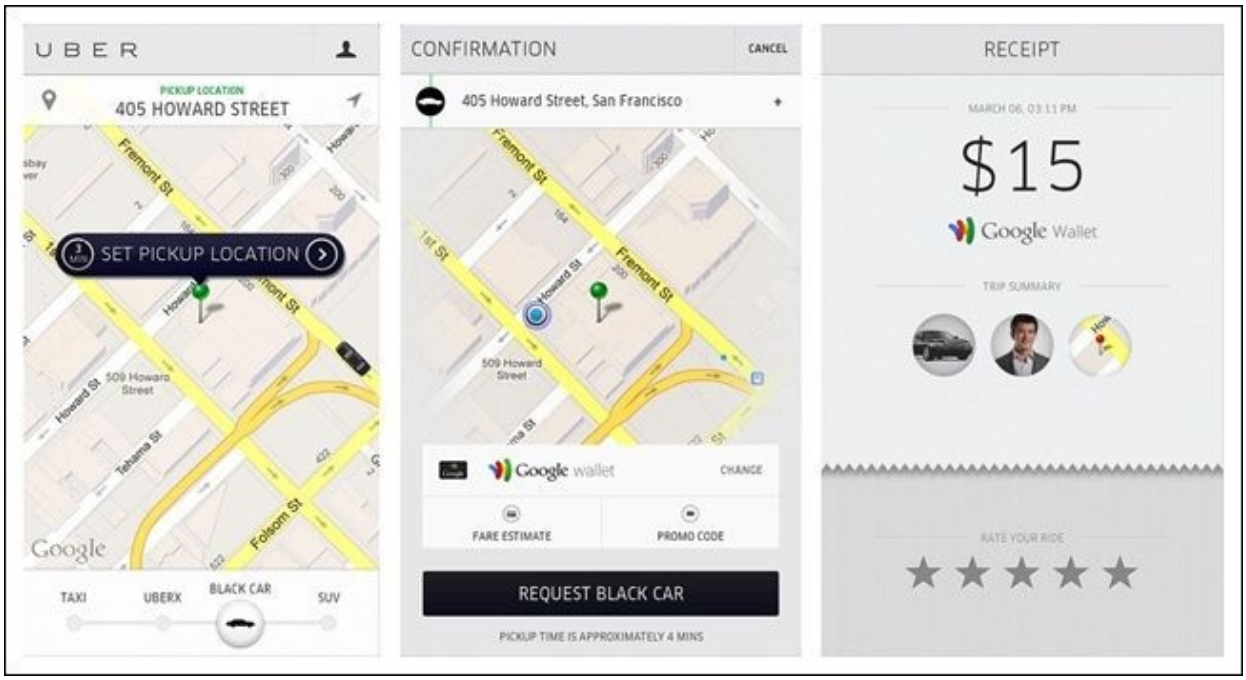



*Figure 2-44. Uber for iOS integrates Google Wallet to simplify booking a car*

## Tip #4: Offer Express Checkout

Consider offering an ultra-streamlined checkout process for registered users. The Apple Store prompts users with an option to set up Express Checkout. Amazon offers 1-Click checkout right on the product details page. In the example shown, there are two buttons for Mobile 1-Click, both in bright, call-to-action orange and sporting the 1-Click icon.

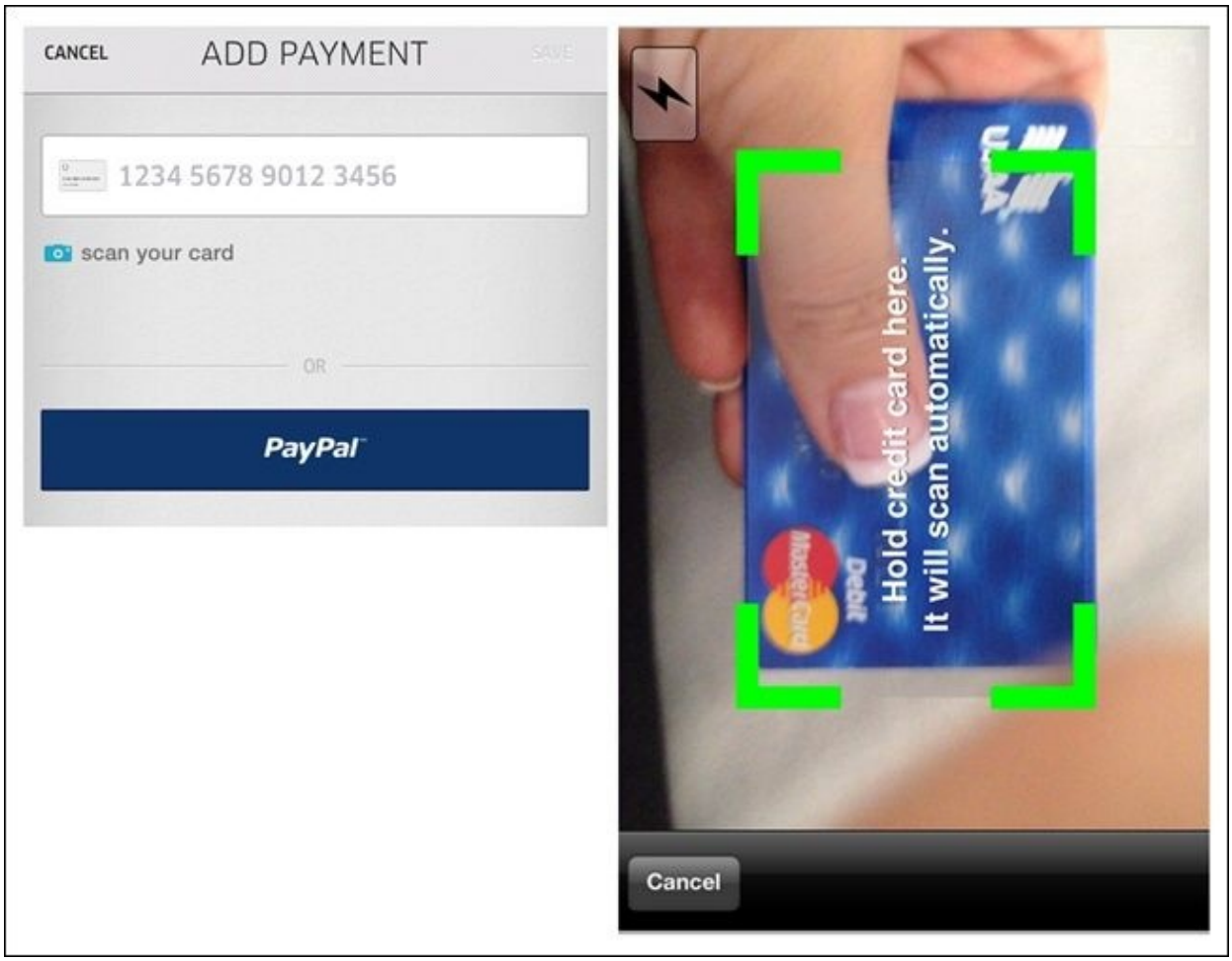



*Figure 2-45. Uber for iOS also offers credit card scan and PayPal options*

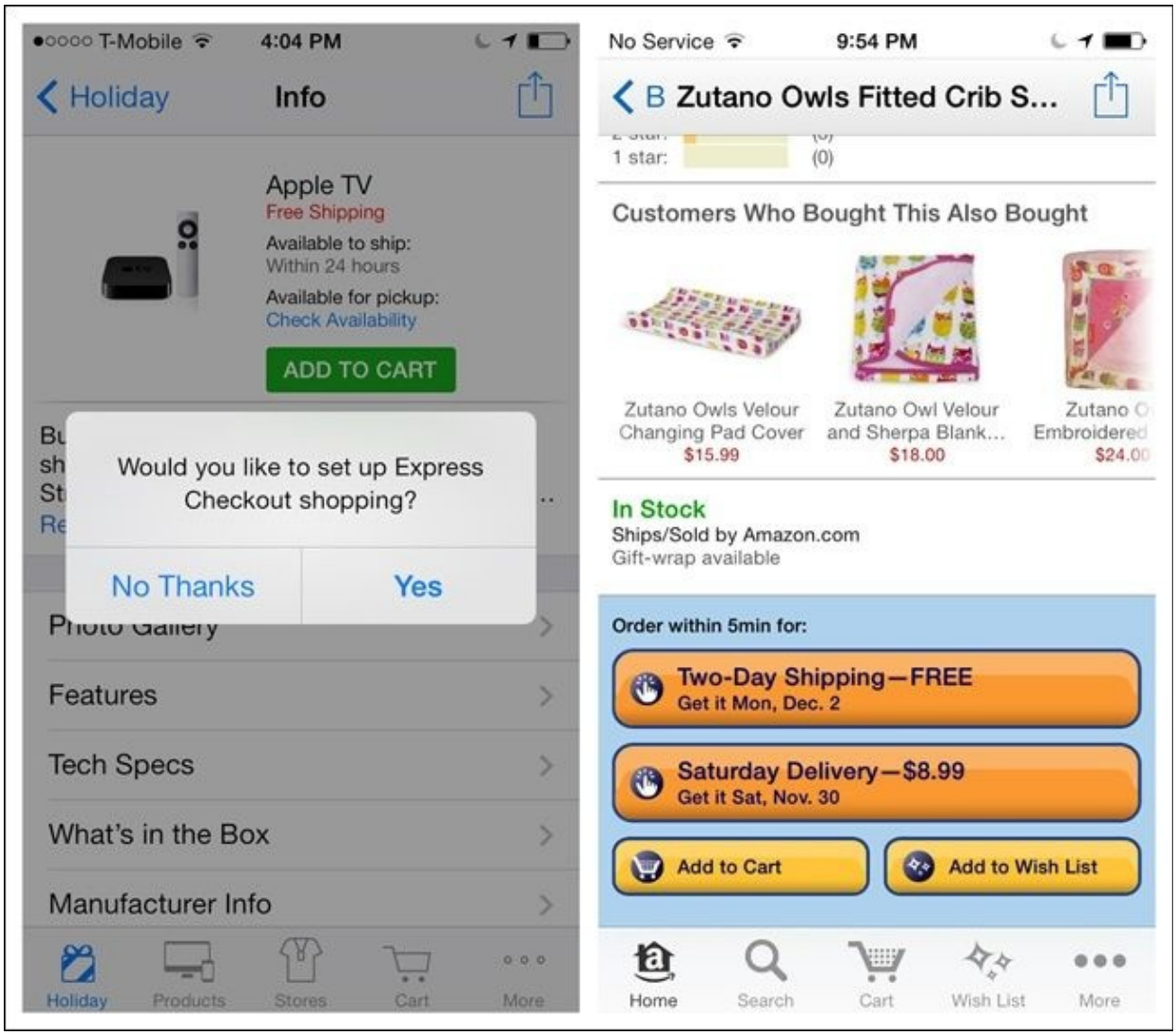



*Figure 2-46. Apple Store and Amazon for iOS: express checkout options*

## Tip #5: Forget the Web

Shopping on the Web has familiarized us all with the shopping cart metaphor and the multi-step checkout process, but the mobile retail space is ripe for innovation. Apple, Walmart, Target, and other retailers are experimenting with new checkout experiences both in and outside of the stores, with an eye to converting more sales on mobile.

Walmart's Scan & Go feature, for instance, lets in-store customers skip the checkout line by scanning their own items as they add them to their shopping carts. Walmart reports that over half of all first-time users eventually use the feature again.

**NOTE**

Design for mobile conversion by streamlining the checkout flow, eliminating unnecessary steps, and providing shortcuts. Look past web shopping conventions and take advantage of what native mobile technology can do.

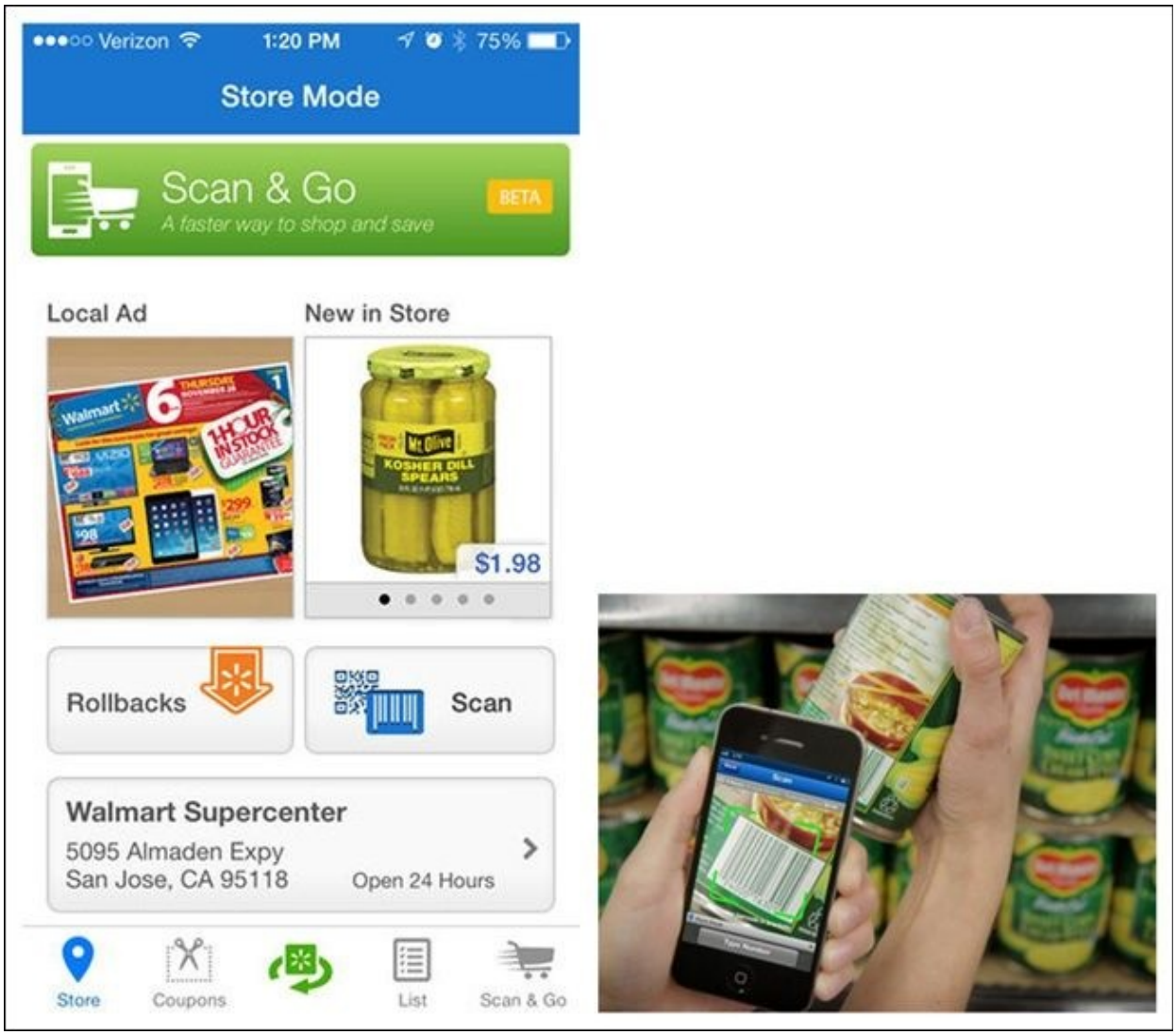



*Figure 2-47. Walmart for iOS: Scan & Go feature lets you skip the checkout line by scanning your own items*


*Figure 2-48. The Apple Store includes the EasyPay self-checkout feature*


*Figure 2-49. Walmart Canada and Mattel's Virtual Toy Store (http://bit.ly/1g3f9Ld): scan a QR code to purchase an item*


*Figure 2-50. Target and CBS Outdoors (http://bit.ly/1g3fi14): a bus stop becomes a "digital holiday experience"*


*Figure 2-51. ShopKick's ShopBeacon (http://tcrn.ch/1g3fLR2) for finding deals and PayPal Beacon (http://bit.ly/1g3fQUG) for accepting payments both use proximity-based Bluetooth to personalize and simplify mobile purchases*

# Calculator Forms

Calculator-style apps, like weight trackers, tax estimators, and loan calculators, all require user input. And although these forms can accommodate custom controls and layouts, it is still important to follow basic conventions to make sure they are readable.

Alignment, labels, fonts, button placement, contrast, and colors all affect the usability of mobile forms. For example, compare the readability of the Valspar Paint Calculator, with its well-aligned fields and labels, with that of the Behr Paint Calculator.


*Figure 2-52. The Valspar Paint Calculator, with well-aligned fields, is easier to read than the Behr Paint Calculator*

The best calculation apps tightly correlate the input with a visual result. TaxCaster uses a gauge-style chart to visualize the amount of taxes due or to be refunded. The gauge updates as data is entered.

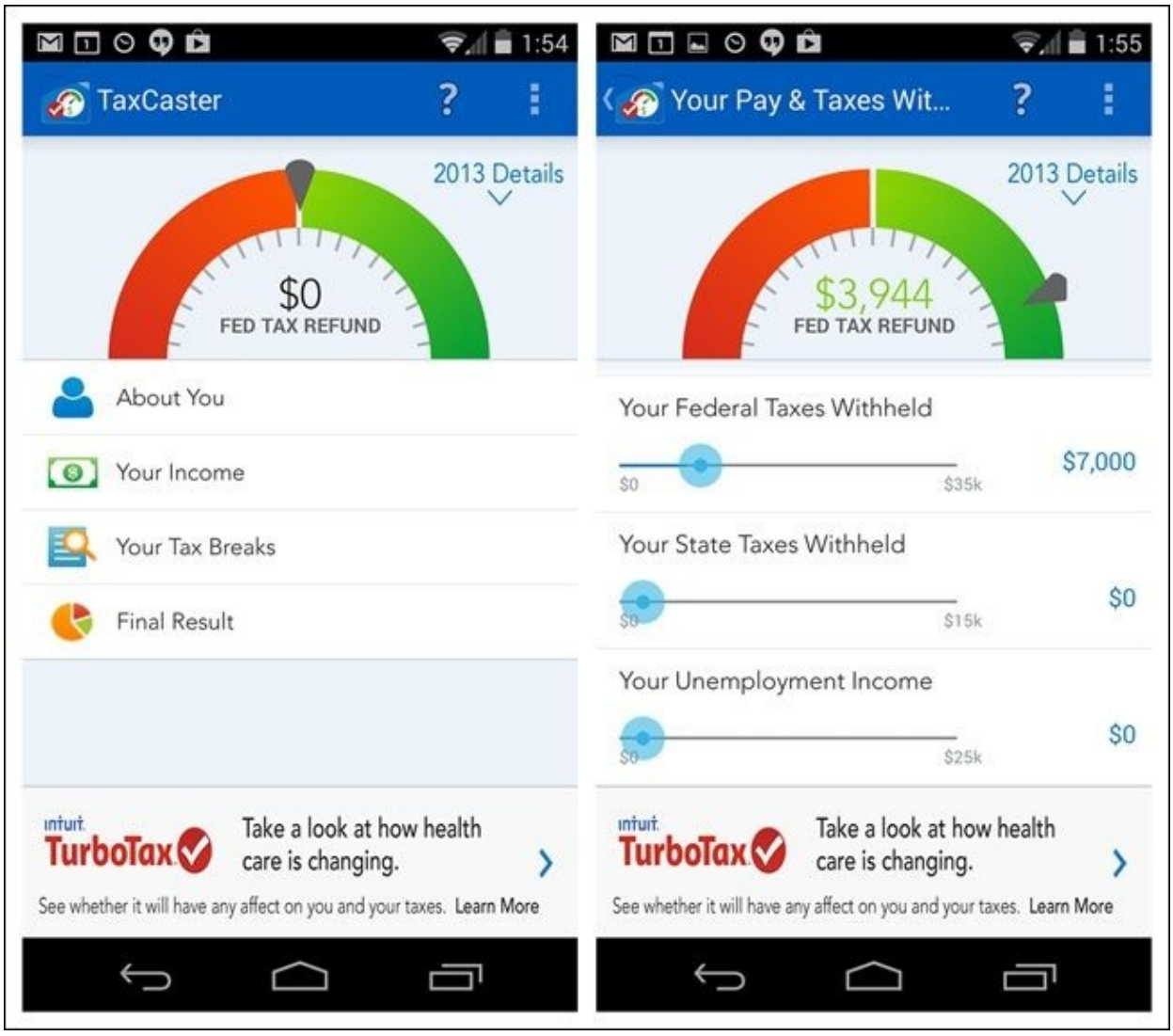



*Figure 2-53. TaxCaster for Android: gauge updates as input is changed*

The Zillow and Trulia Real Estate Calculators also provide a real-time view of data inputs.

**NOTE**

Use standard form conventions for calculator form layout. Add interactive visualizations if they are appropriate for the way the app will be used.

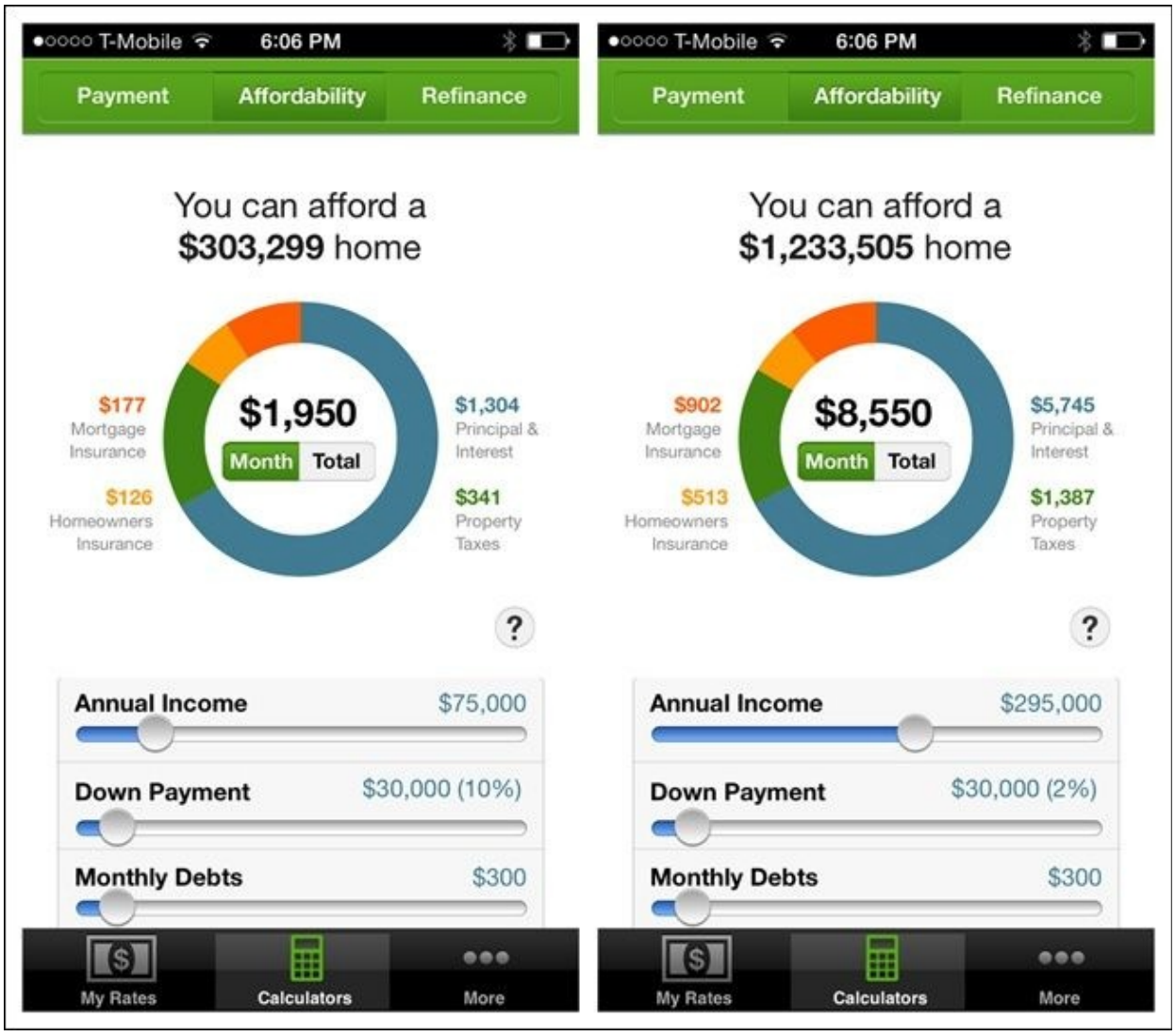



*Figure 2-54. Trulia Real Estate Calculator for iOS: real-time feedback*

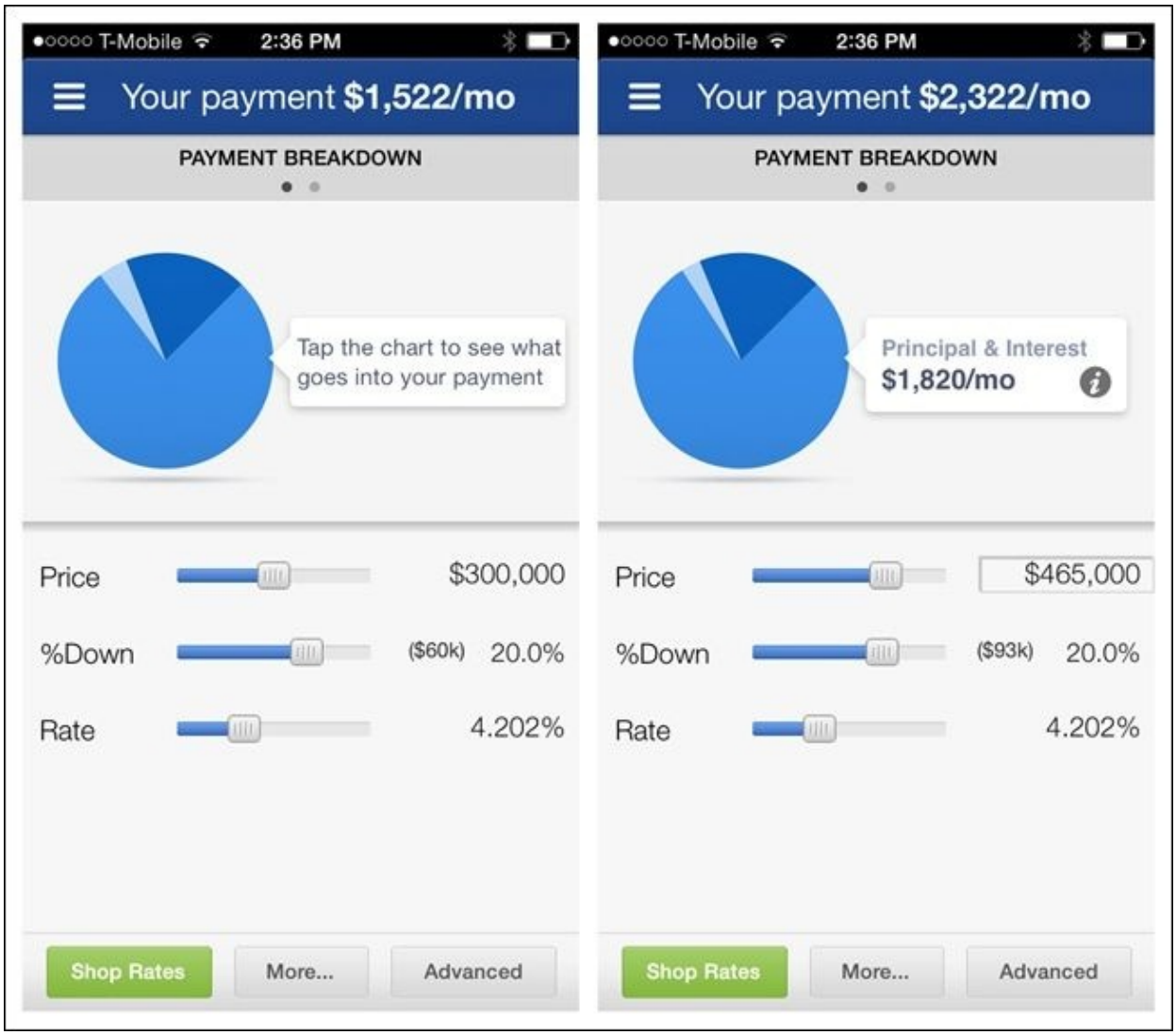



*Figure 2-55. Zillow Mortgage Calculator for iOS: display changes with data input*

# Search Forms

Some searches require multiple inputs to generate results. Like the other form patterns, Search forms should have only the essential or most requested fields and provide sensible defaults. You can always offer a filter option on the results page to let people refine the results list. I'll focus on the form itself here; see Chapter 4 for more on search, sort, and filter patterns.

Compare the confusing layout of the American Airlines Search form with the clean design in Kayak.

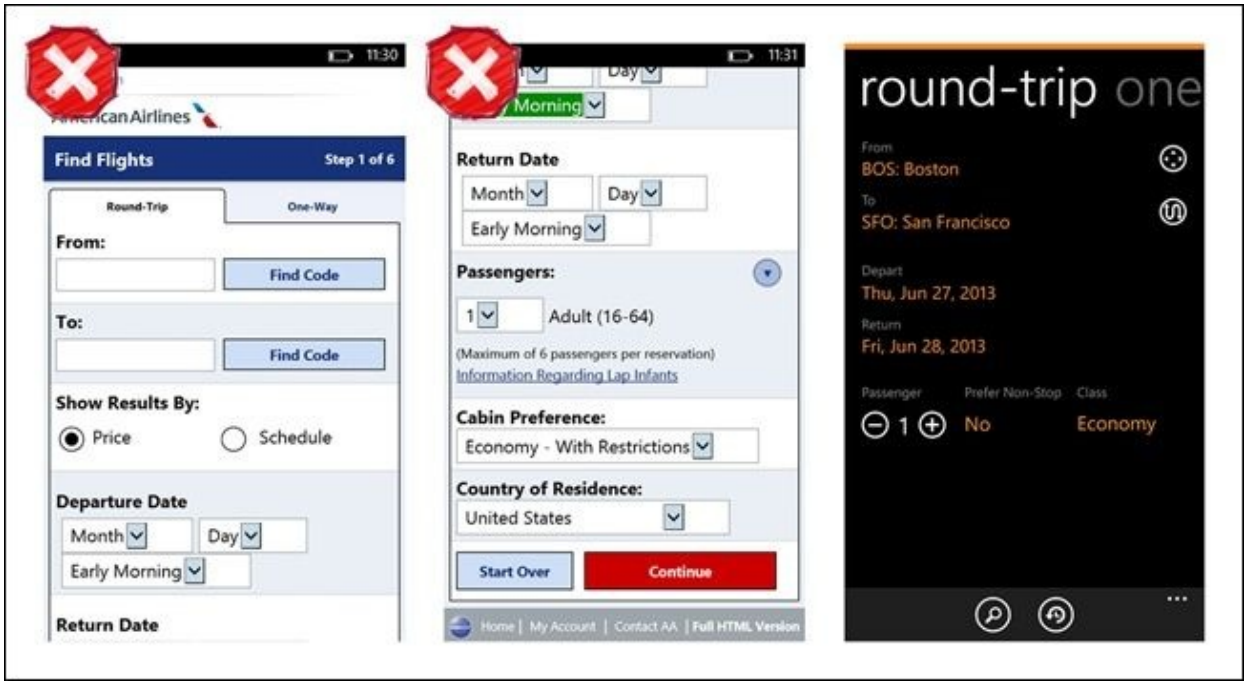



*Figure 2-56. American Airlines (2013) and Kayak for Windows Phone: clutter versus clean*

Zillow has a solid design that offers a preview of the number of results directly on the form. If you register with Zillow, you can also save this search, to save time and effort in future searches.

Hipmunk offers a concise Search form, as well as an added bonus of fare alerts. Fare alerts are similar to saved searches in that they provide an extra level of efficiency for the mobile user by saving the search criteria; but unlike a simple saved search, fare alerts provide in-app notification when fares change. This is a critical part of Hipmunk's success, since the alerts bring the users back into the app again and again.

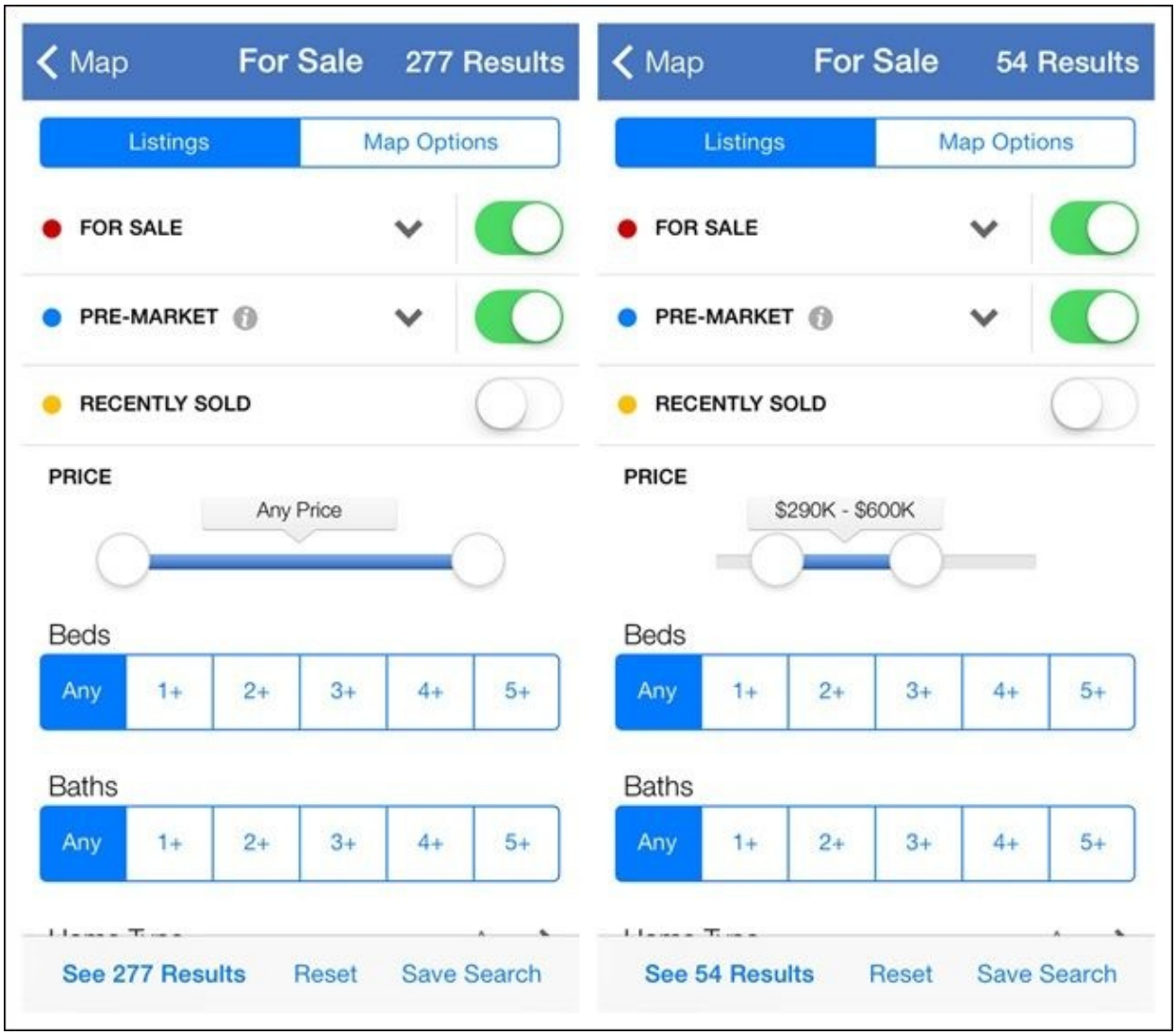



*Figure 2-57. Zillow for iOS displays the number of results as inputs are entered on the form*

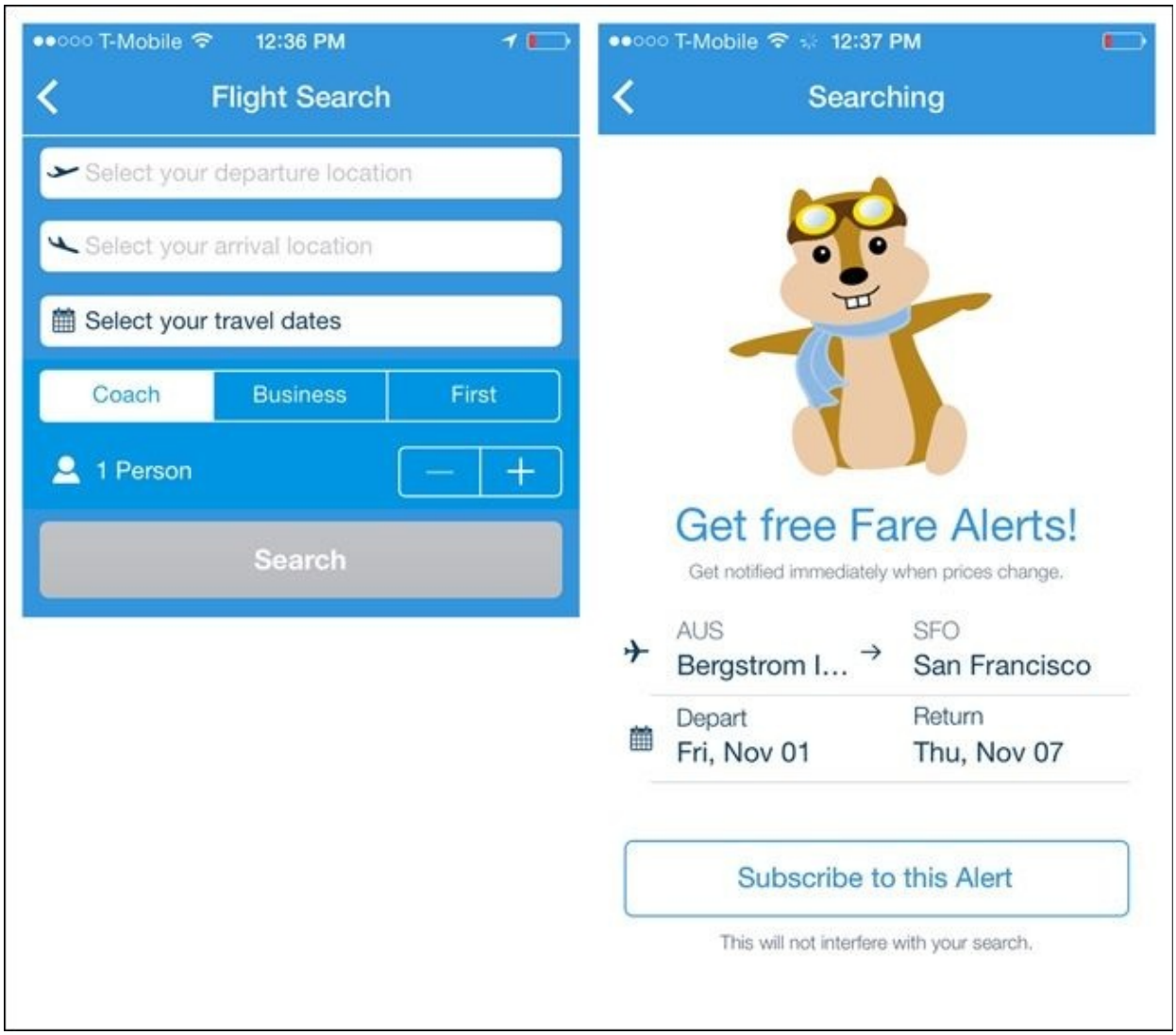



*Figure 2-58. Hipmunk for iOS: fare alerts add money-saving utility to saved searches*

Let's look at the OpenTable Search form from 2011 and compare it with the more recent iOS 7 app. The early design is a full Search form with four to five fields and a large red call-to-action button.

The iOS 7 design opens up with a list of nearby restaurants and possible seating times for today. This is an example of an Implicit Search, as discussed in Chapter 4. Search is now just an overlay with three fields, with autosuggest enabled for the restaurant name field.

**NOTE**

Keep Search forms short and simple; aim for a single page of criteria or less. Offer reasonable defaults and the option to save the search for later.

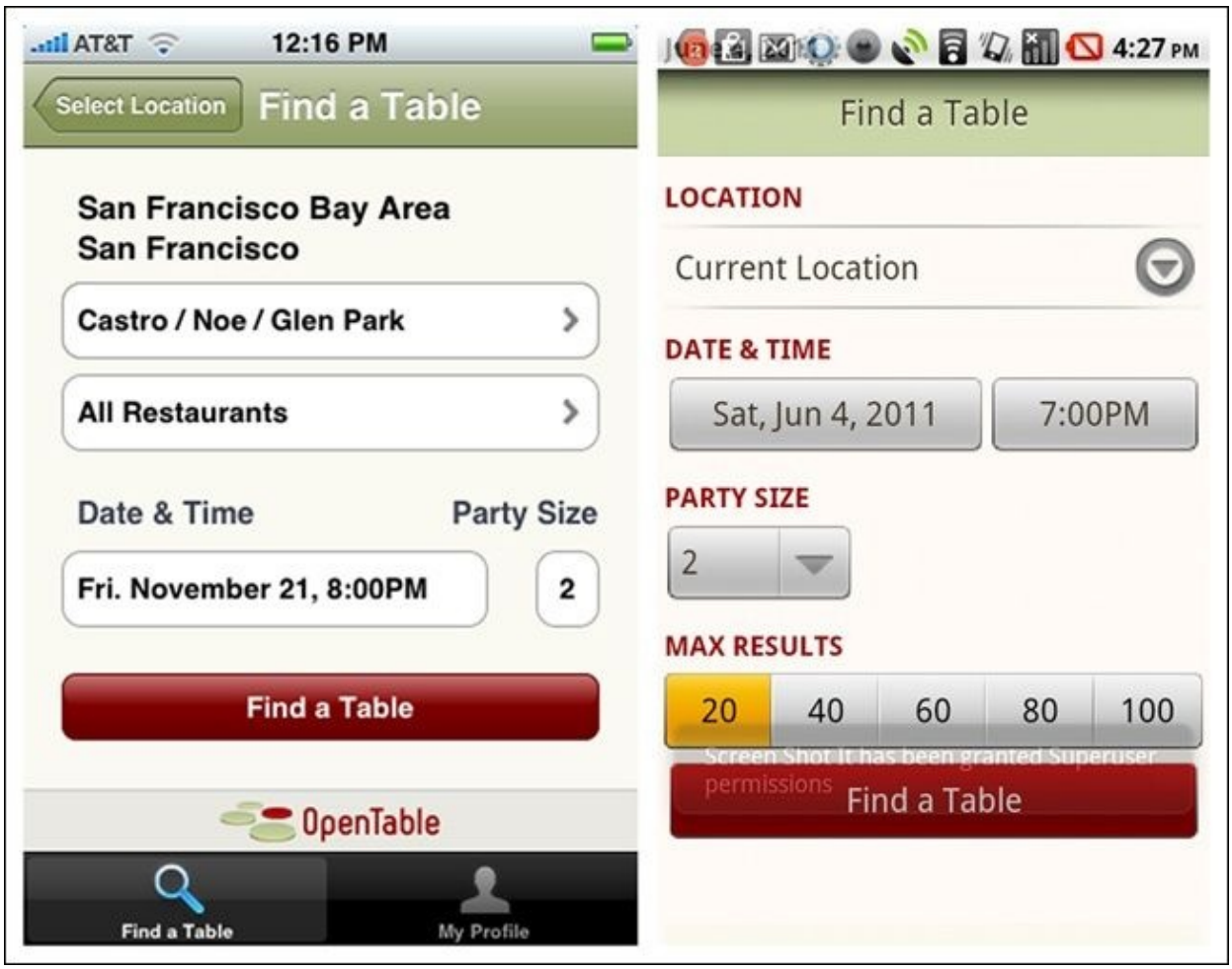



*Figure 2-59. OpenTable for iOS and Android (2011): user must enter all Search criteria*

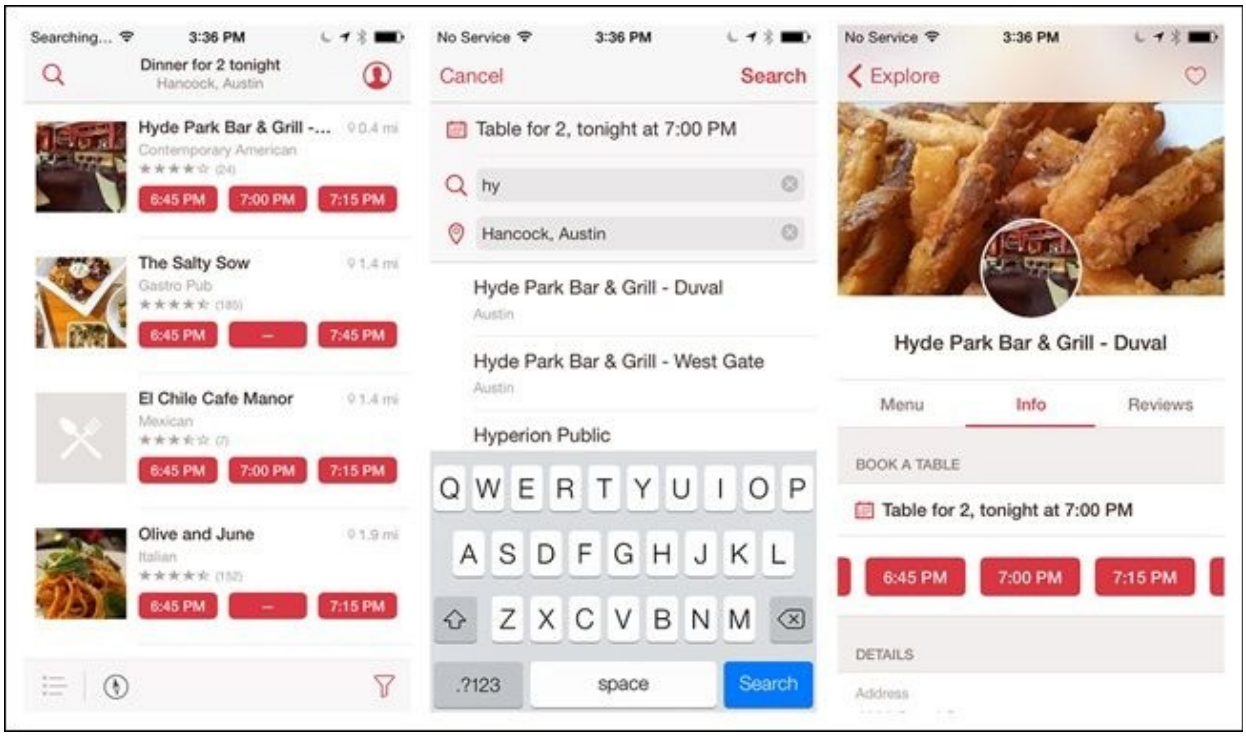



*Figure 2-60. OpenTable for iOS 7: Implicit Search autosuggests nearby options*

# Long Forms

Some forms can't fit on a single screen. In these cases, a long scrolling page is vastly preferable to a form broken up across multiple pages. The trickiest part of the Long Form is where to put the Command and Escape buttons.

In most iOS apps, forms are presented in a modal context. The iOS 7 design guidelines (*http://bit.ly/1g3h0j5*) recommend that the Command button reside at the top right and the Escape button (typically Back or Cancel) at the top left.

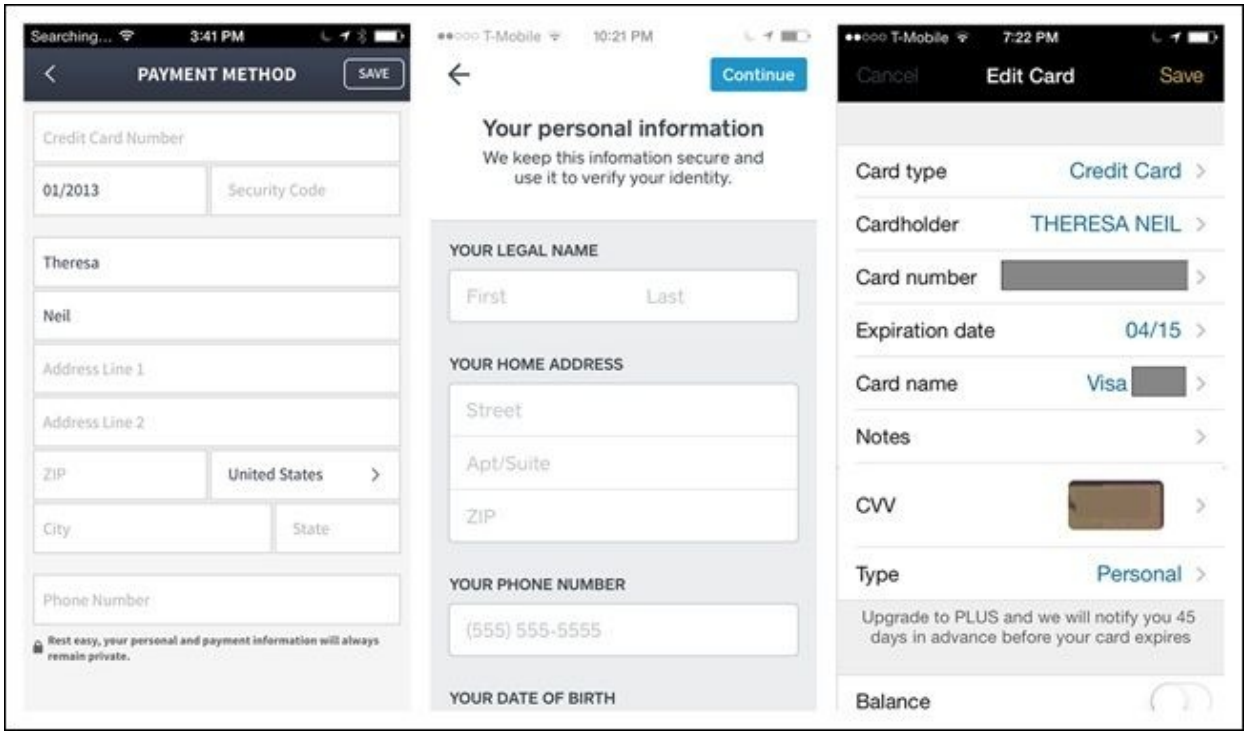



*Figure 2-61. HauteLook, Square, and Lemon Wallet for iOS: Command button at top right*

Windows Phone 8 design guidelines (*http://bit.ly/1g3h4PM*) suggest that the Command button should reside in the App Bar (which runs across the bottom of the screen).

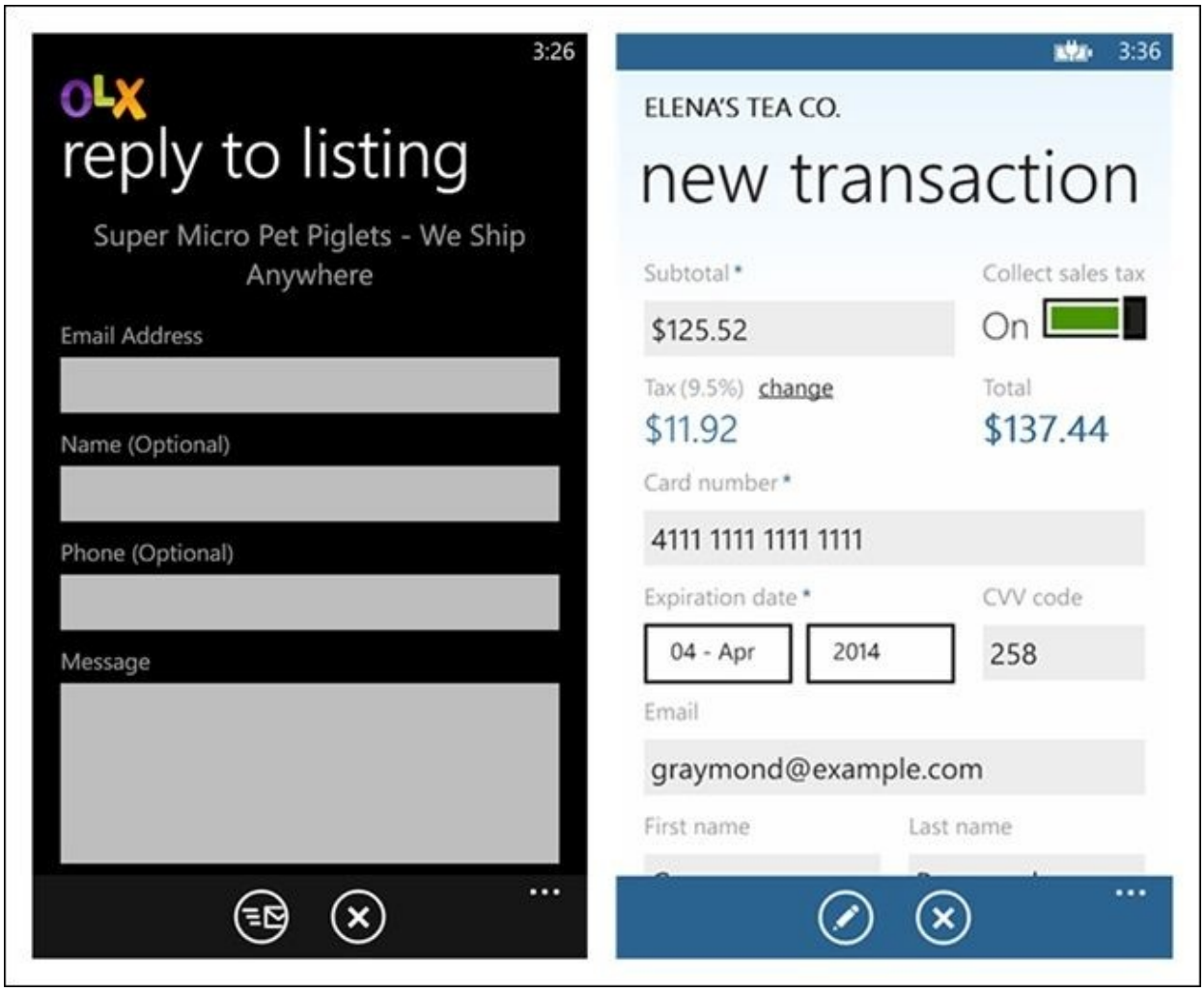



*Figure 2-62. OLX and Innerfence for Windows Phone: Command button in App Bar*

And, yes, Android has a completely different recommendation (*http://developer.android.com/design/patterns/actionbar.html*). Since the release of Jelly Bean (Android 4), Android's design guidelines recommend that the Command button be located in the Action Bar (across the top).

Unfortunately, as I write this in late 2013, this Android design guideline doesn't always test well. In user testing for a number of our clients, a majority of users didn't see or couldn't find the buttons in the Action Bar. This usability issue will likely decline over time as more users upgrade their devices and OS and get familiar with the new navigation conventions. In the meantime, some Android apps rely on a fixed footer to display the Command button.

**NOTE**

Don't artificially break the form into multipage steps — scrolling on one page is preferable. But do ruthlessly edit any unnecessary fields. Follow OS standards for button placement.

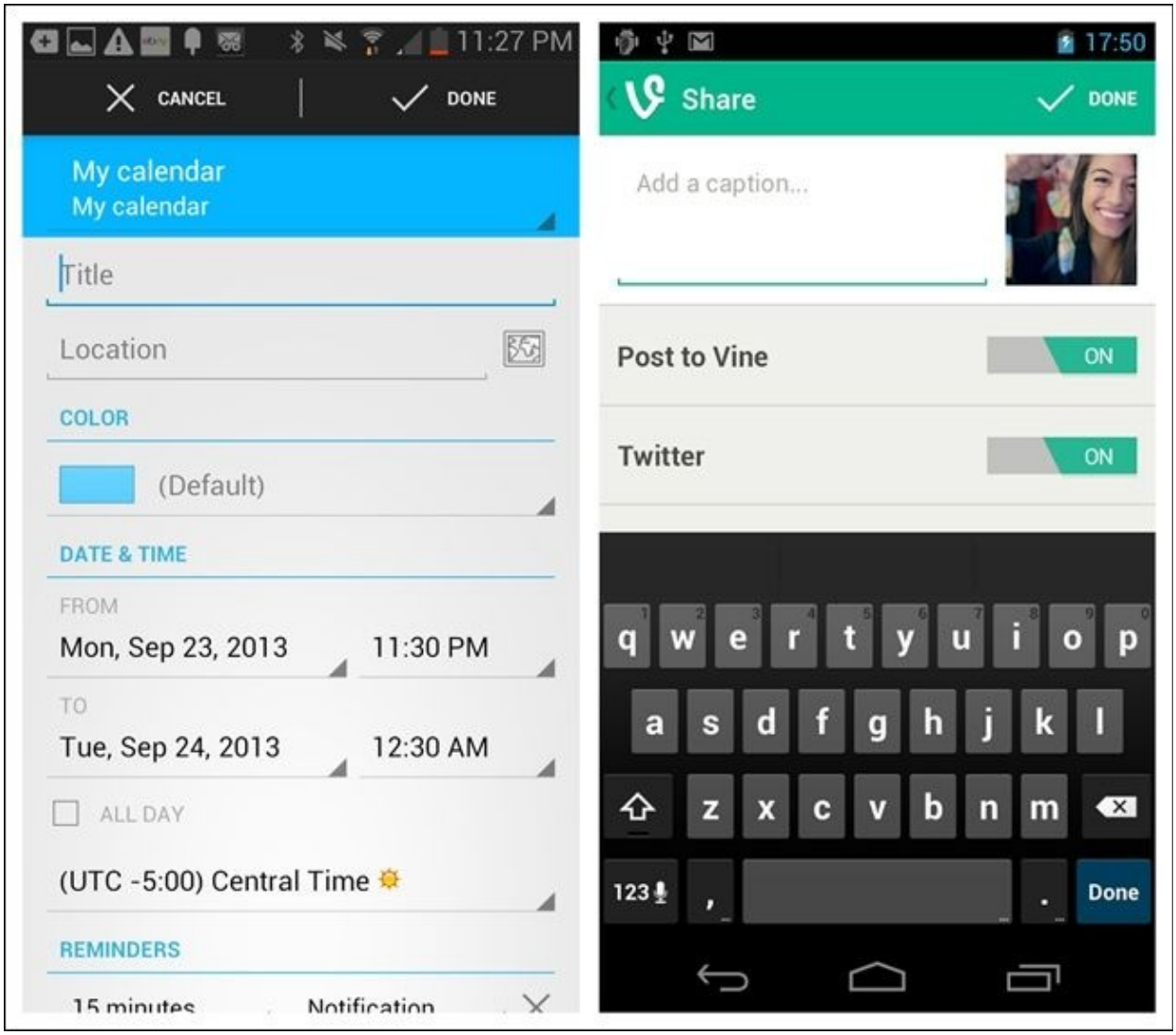



*Figure 2-63. DigiCal and Vine for Android: examples of modal form and simple share form, respectively, with Command button in Action Bar*

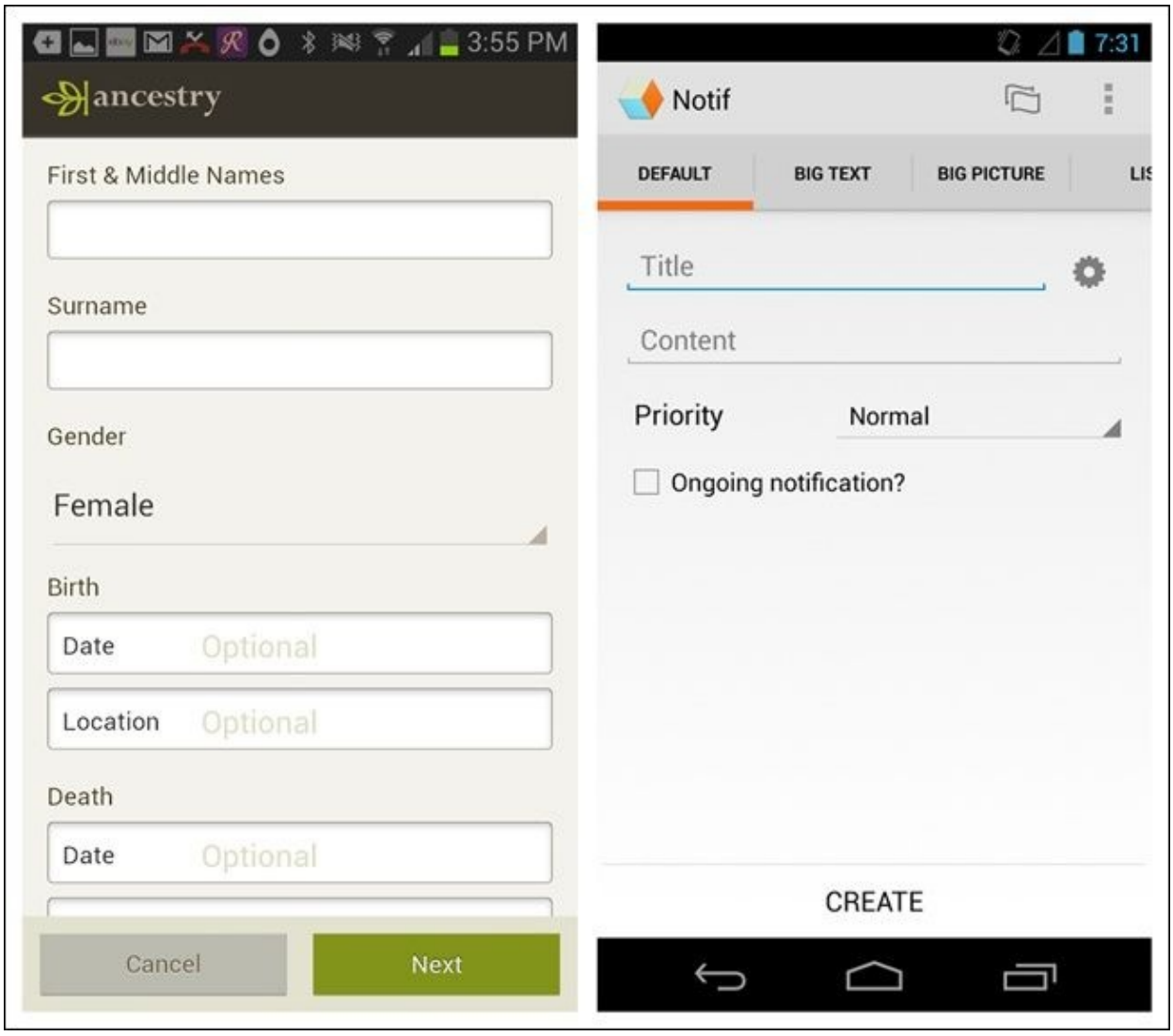



*Figure 2-64. Ancestry and Notif for Android: fixed Command button in footer*

# Chapter 3. Tables


*Patterns*

> Basic Table, Headerless Table, Fixed Column, Overview plus Data, Grouped Rows, Tables with Visual Indicators, Editable Tables

In the first edition of this book, I introduced this chapter by mentioning that many of my company's clients have enterprise applications that require dense displays of data. Since they are familiar with using tables to read this data, they are curious to know how we're going to make those tables work on a mobile device. And my answer, then as now, is "We're not." At least, not without accounting for the demands the mobile form factor imposes on designers.

Displaying tabular data in mobile presents a challenge *and* an opportunity. The challenge is finding usable ways to display the data on small screens. The opportunity is that we can take a more critical view of the data to determine what the user really needs to take away from it, and then develop creative solutions to display the data so that it will be even more useful.

To revisit my example from last time, a long table of student test results can be better represented as a chart that users can drill into to see all students who scored in a certain band of the bell curve.

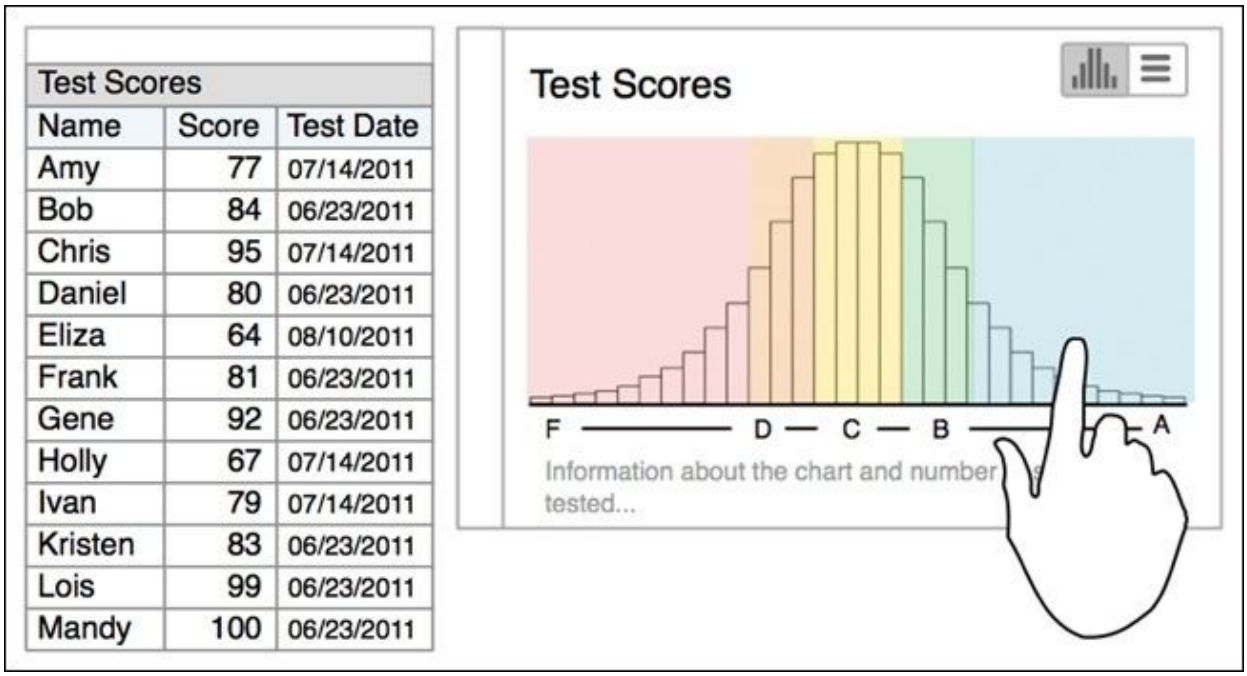



*Figure 3-1. Redesigned table: converting to an interactive chart may be more useful*

And for an alternate view of the information, we can provide a Headerless Table with Dynamic Search for quickly accessing a specific student's results.

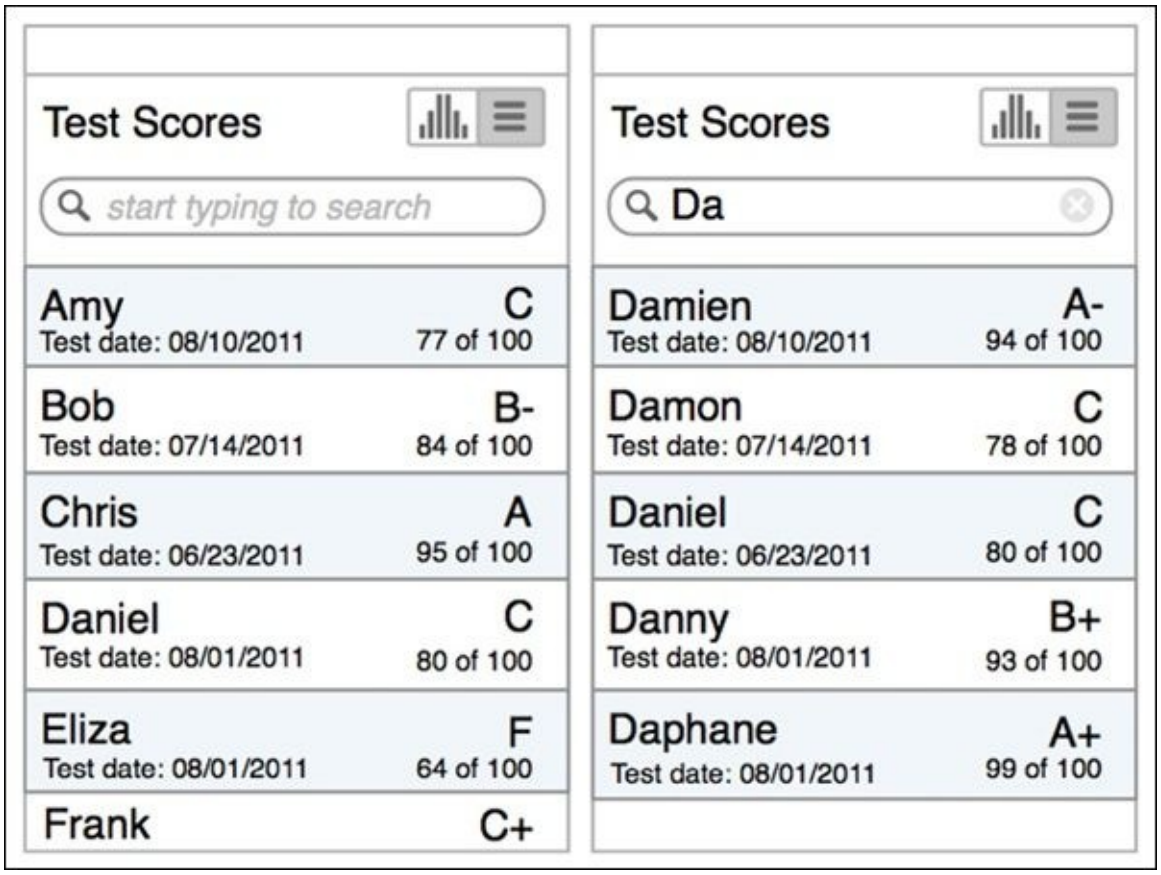



*Figure 3-2. Headerless Table with Dynamic Search: alternative to the bell curve chart*

Though the patterns have largely stayed the same from a couple of years ago, designers have become more adept at using them to display tabular data clearly and elegantly. Once you've identified the key data your app needs to display, check out the following table patterns for inspiration.

# Basic Table

The Basic Table pattern is just a standard table with fixed column headers and a grid layout. Alternating table row colors, also called *zebra striping*, or placing a thin horizontal line between rows may enhance readability.

It's now common to omit the vertical gridlines to reduce visual noise. That's fine, but be sure to maintain column alignment. To see why, compare the readability of the ESPN SportsCenter app with the table in Stock Trainer.

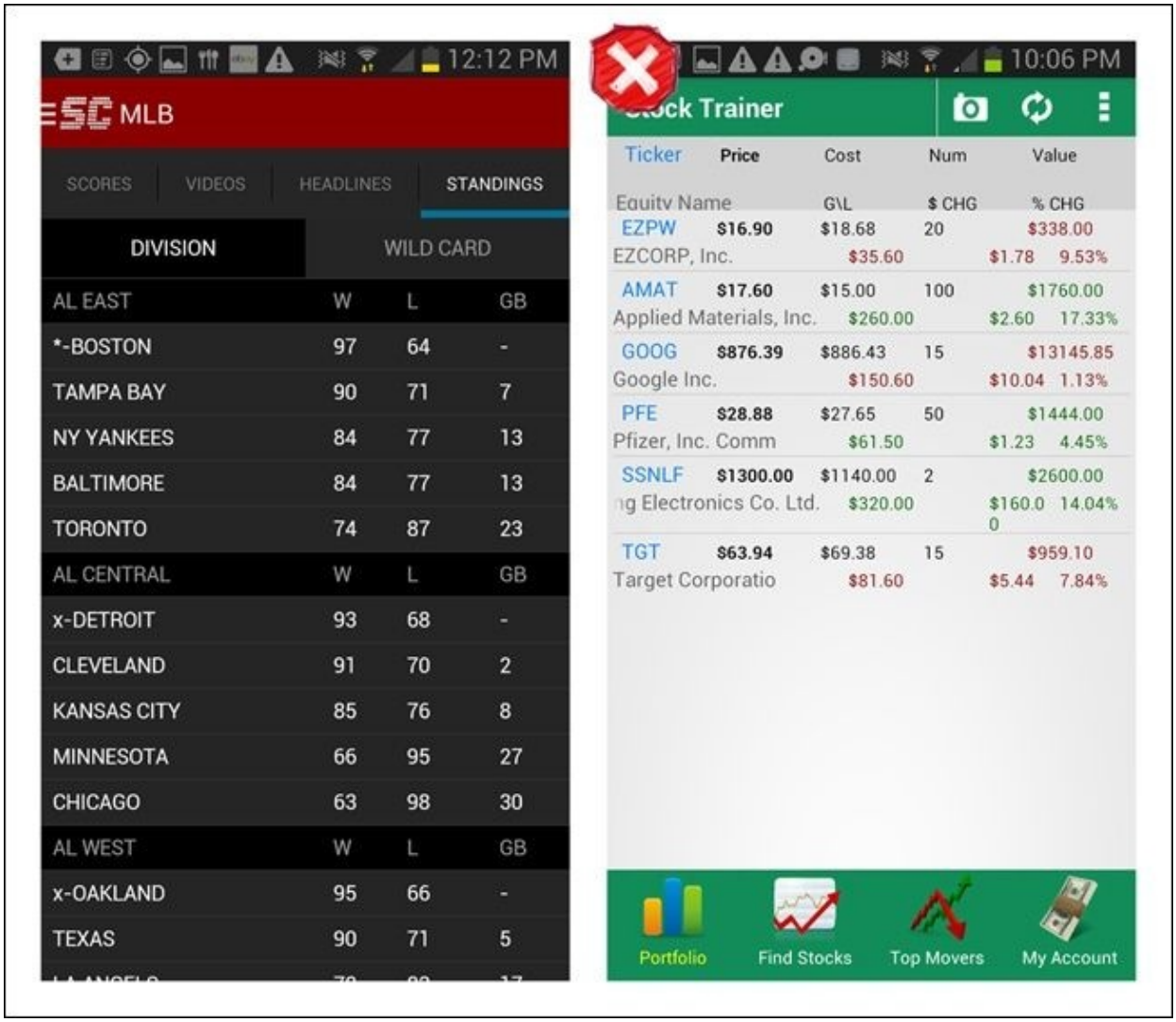



*Figure 3-3. ESPN SportsCenter and Stock Trainer for Android: if skipping vertical gridlines, keep columns aligned*

**NOTE**

Avoid using dark gridlines; if you're leaving out vertical gridlines, make sure the columns are aligned.

Left-align text and right-align numbers. Consider an alternate pattern if there is too much information to fit on a single screen.

## Headerless Table

The Headerless Table is characterized by fat rows displaying multiple variables for an object, and no column labels. It is common practice to make the row identifier — a key piece of information — stand out. BillGuard does this with icons, while Gas Guru uses a larger, darker font.

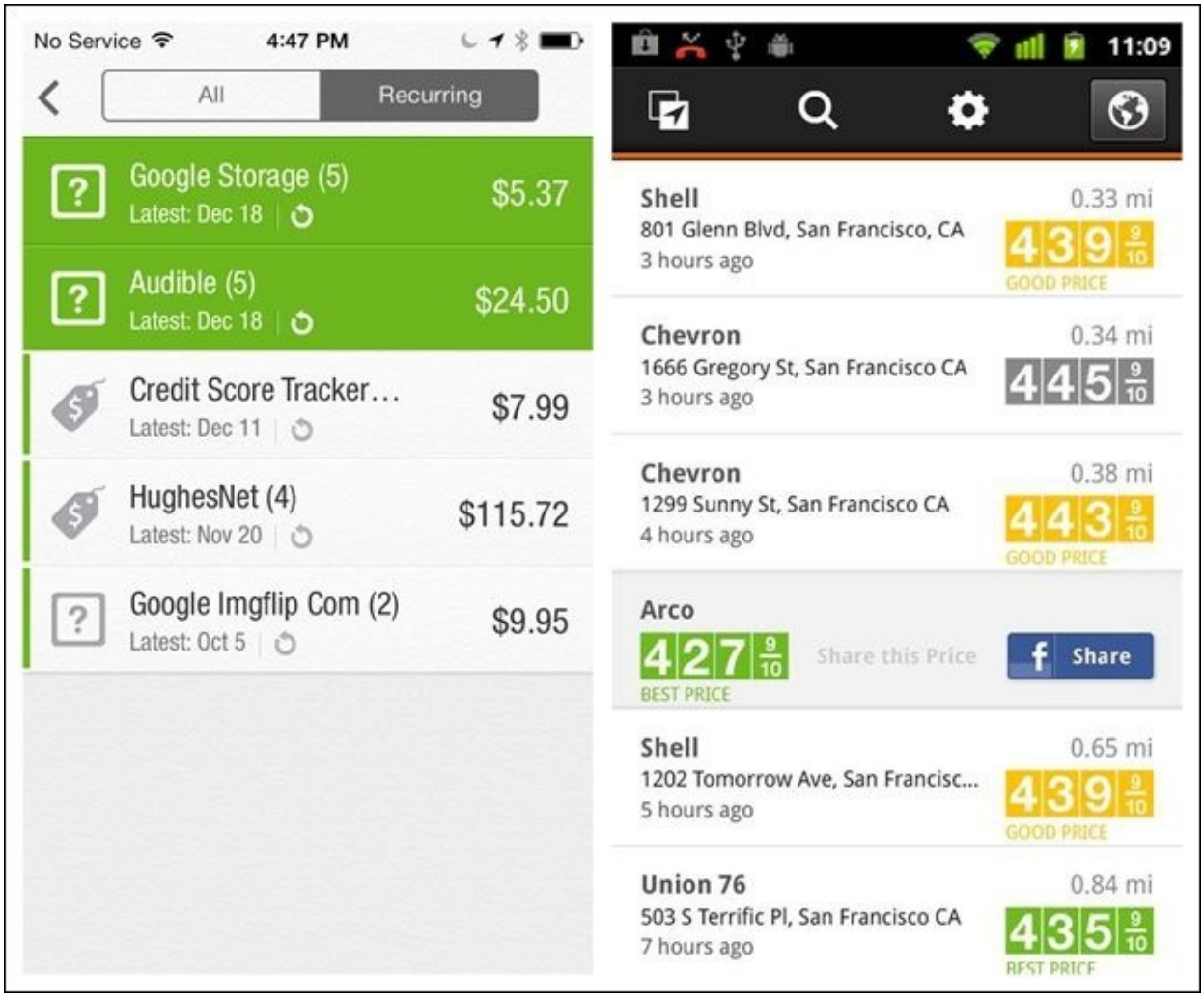



*Figure 3-4. BillGuard for iOS and Gas Guru for Android: conspicuous row identifiers*

Headerless Tables are ideal for displaying collections of items (like inventory, recipes, albums, etc.) and search results. FlightBoard shows the flights for the selected airport. When users tap on a row, the flight details are shown inline instead of on a new screen.

Like a list, Headerless Tables are meant for quick scanning and action. The visual design elements of alignment, font, and color are critically important. See

how the Realtor.com table is easier to scan than the one in Redfin, for instance.

> **NOTE**
>
> Skip the noise (extra visual elements like icons and borders), and create strong alignment for easy scanning. Use a smaller and/or lighter font for less important details. Don't guess what the most important information is — ask your customers, then validate the designs through testing.

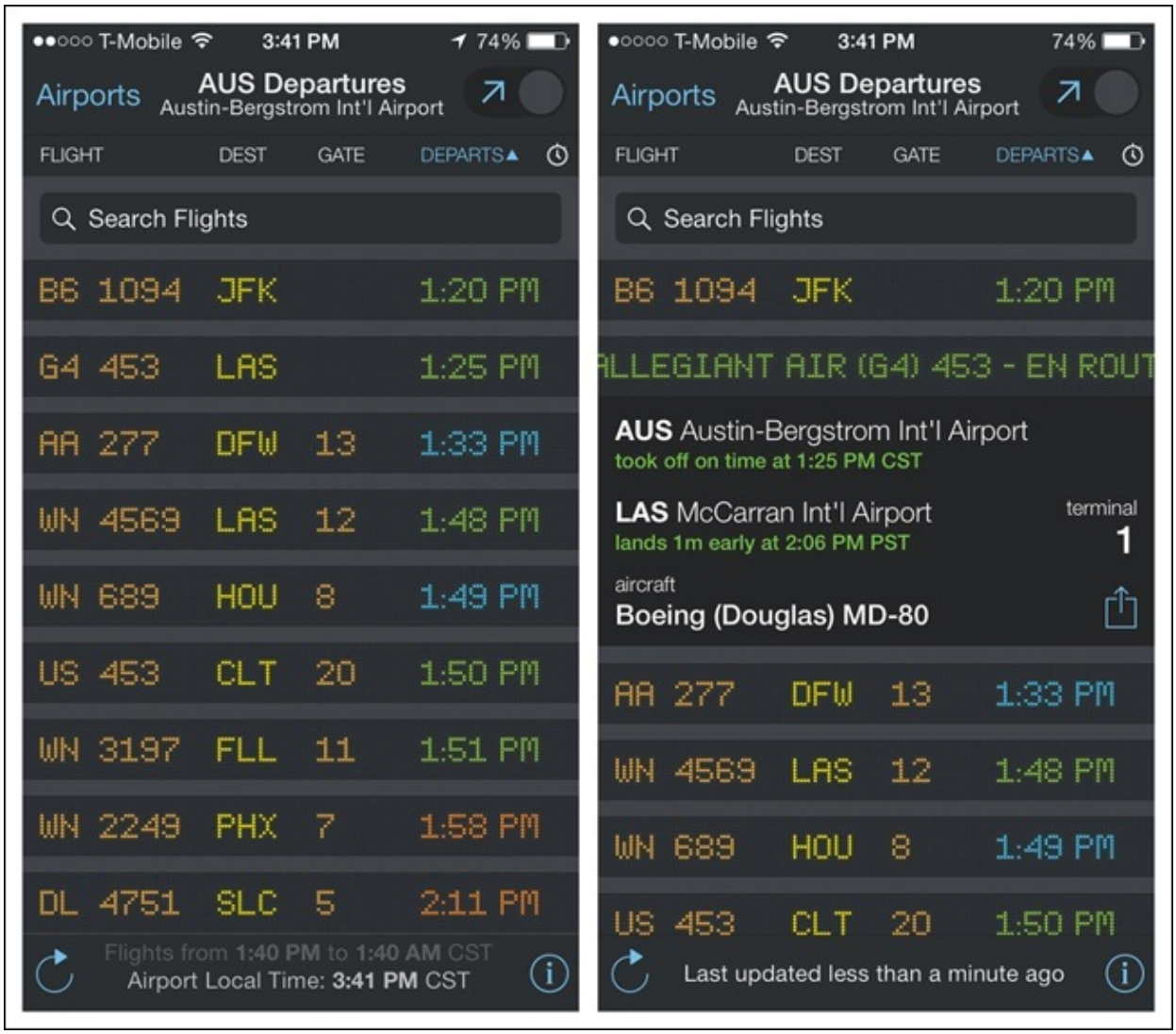



*Figure 3-5. FlightBoard for iOS: Headerless Table with inline expand on tap*

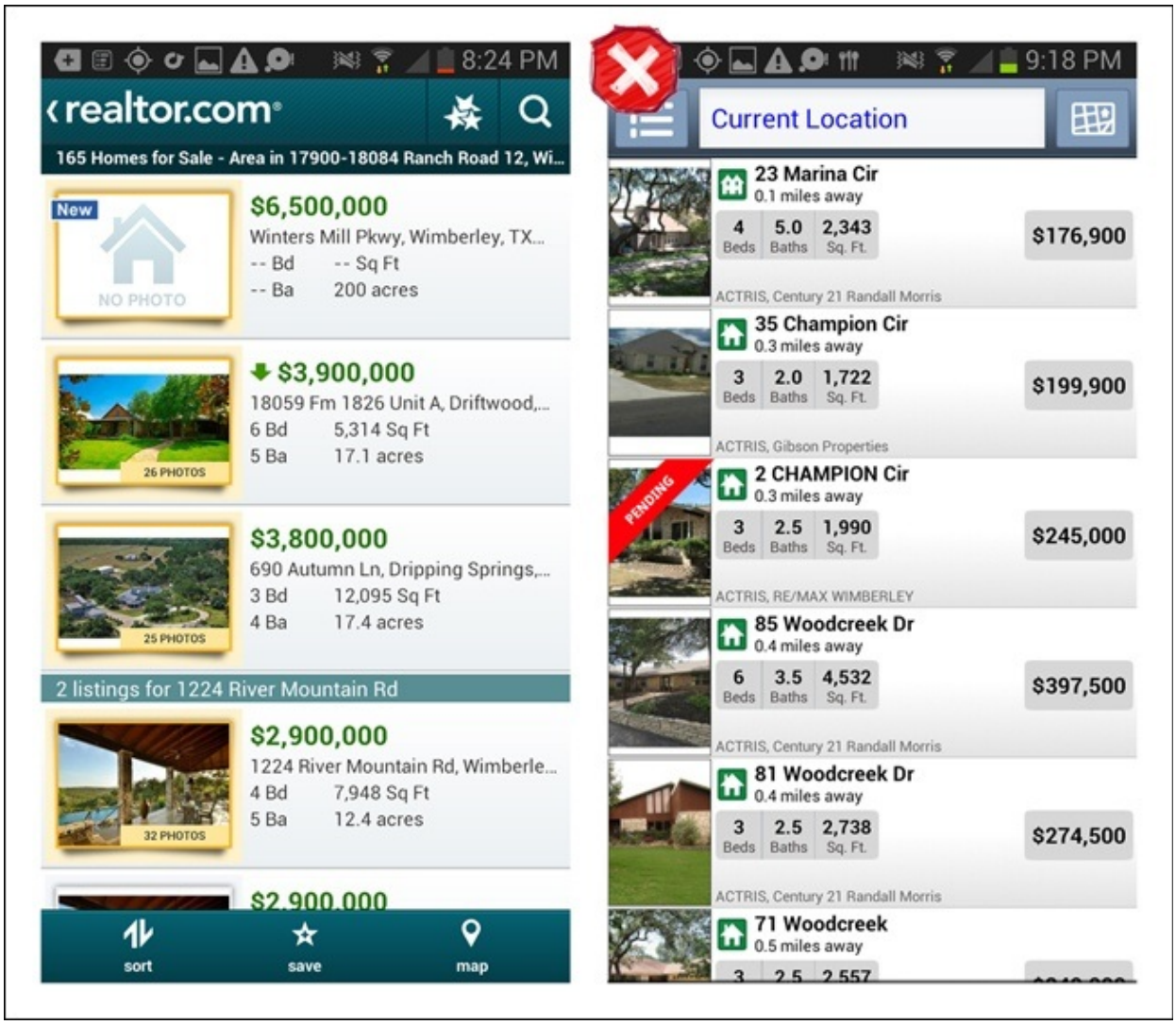



*Figure 3-6. Realtor.com compared to Redfin for Android: design and alignment are critical in Headerless Tables*

## Fixed Column

For larger tables, the Fixed Column pattern may be a viable solution. In the examples from Roambi and Fidelity, when the user is not scrolling vertically, the leftmost column is fixed and the other columns swipe side-to-side. In Roambi's ultra-flat design, at first glance it's a little difficult to see that the left column is fixed, but you can see that more information is available to the right. Fidelity's design helpfully includes visual affordance that the first column is fixed and sorted.

**NOTE**

Provide visual affordance that more columns are visible on swipe. Show the most important columns on the screen by default.

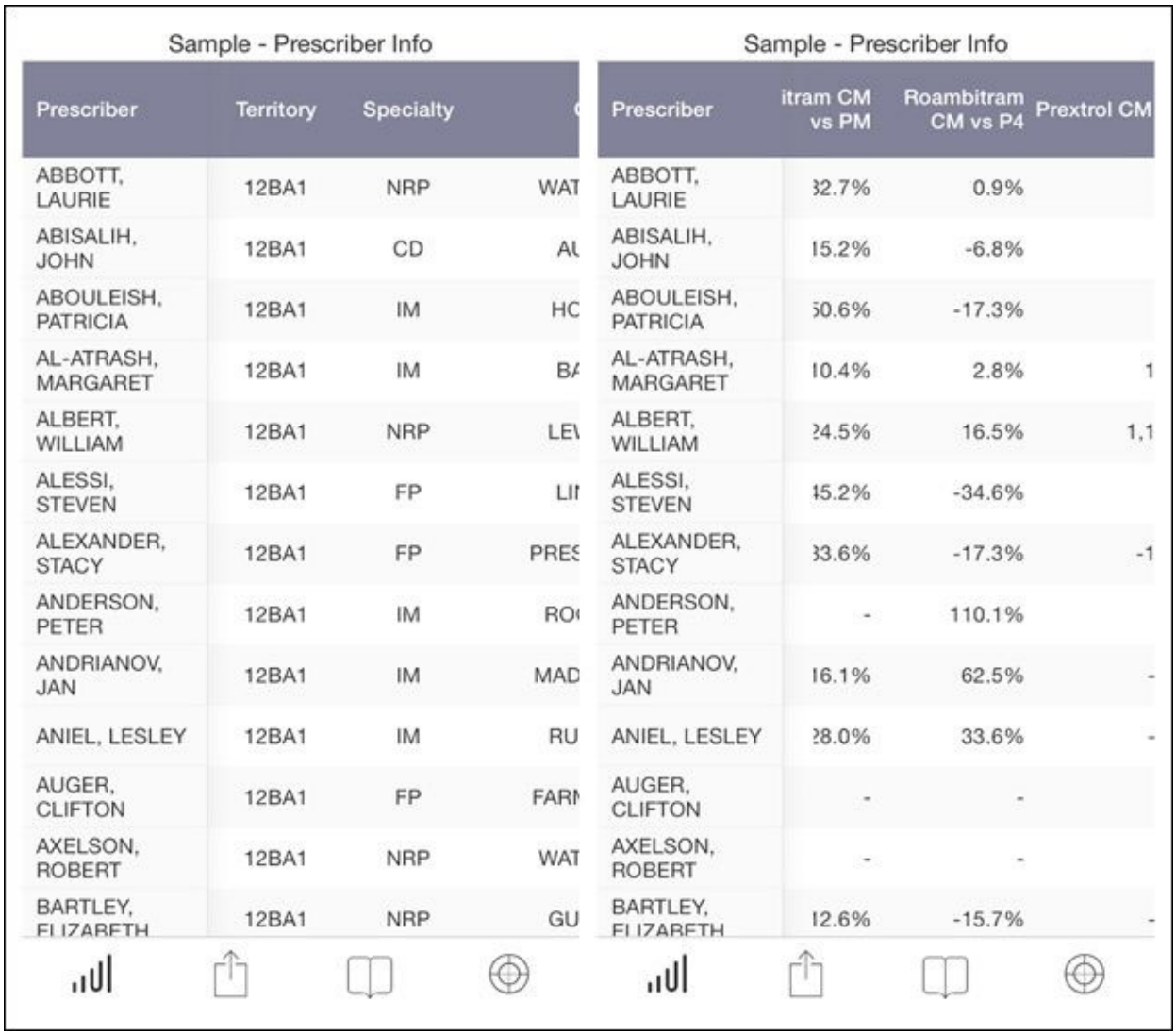



*Figure 3-7. Roambi for iOS: note that leftmost column is fixed (indicated with subtle shadow)*

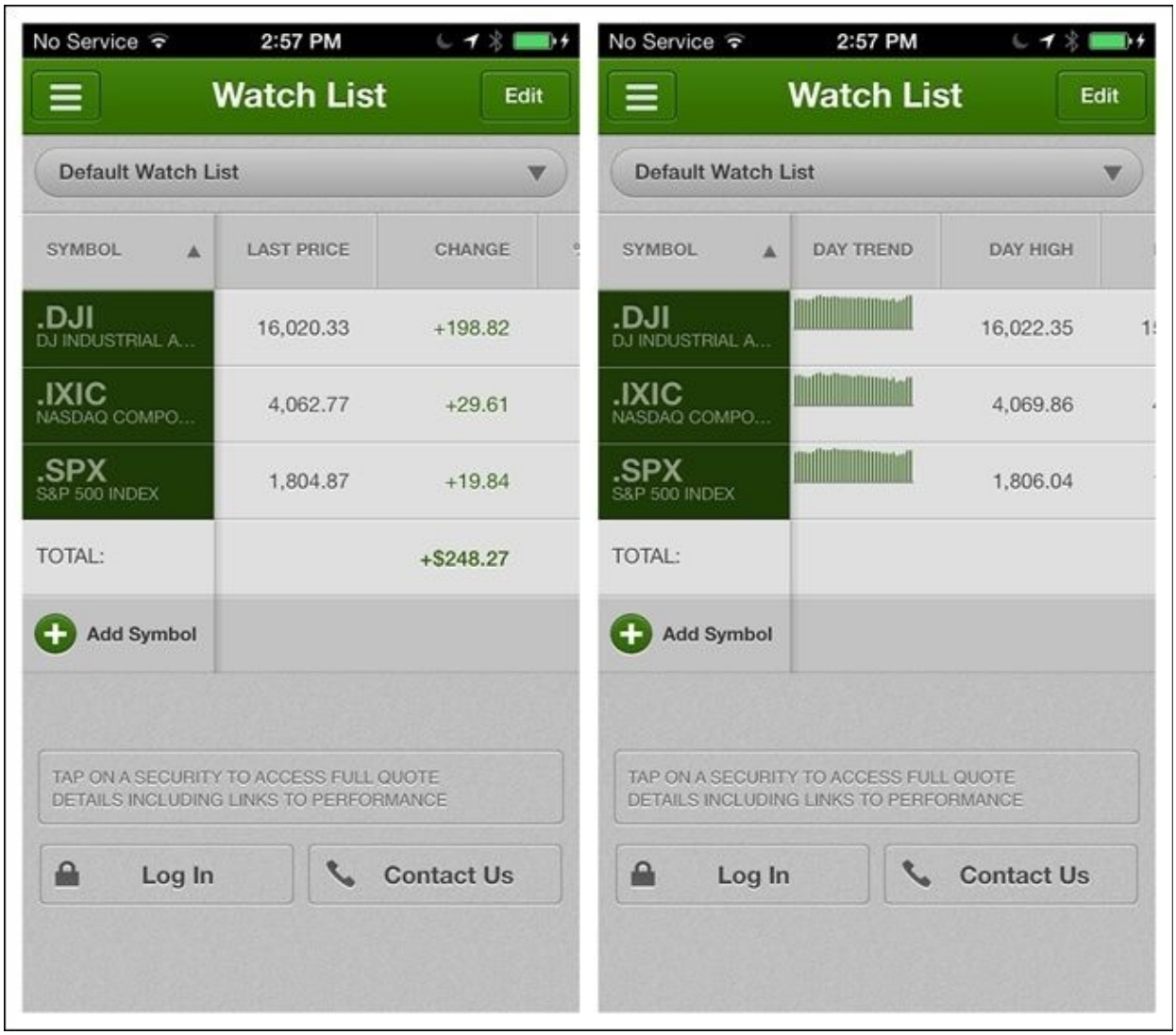



*Figure 3-8. Fidelity for iOS: affordance reveals that left column is fixed, while swiping reveals more columns*

# Overview plus Data

The Overview plus Data pattern features a summary of the table's content displayed above the rows of data. This is a common pattern in financial apps, but it's useful in any app where the user wants to see totals or trends over time.

The overview can be text but is most often a chart-style graphic, as shown in the following examples.

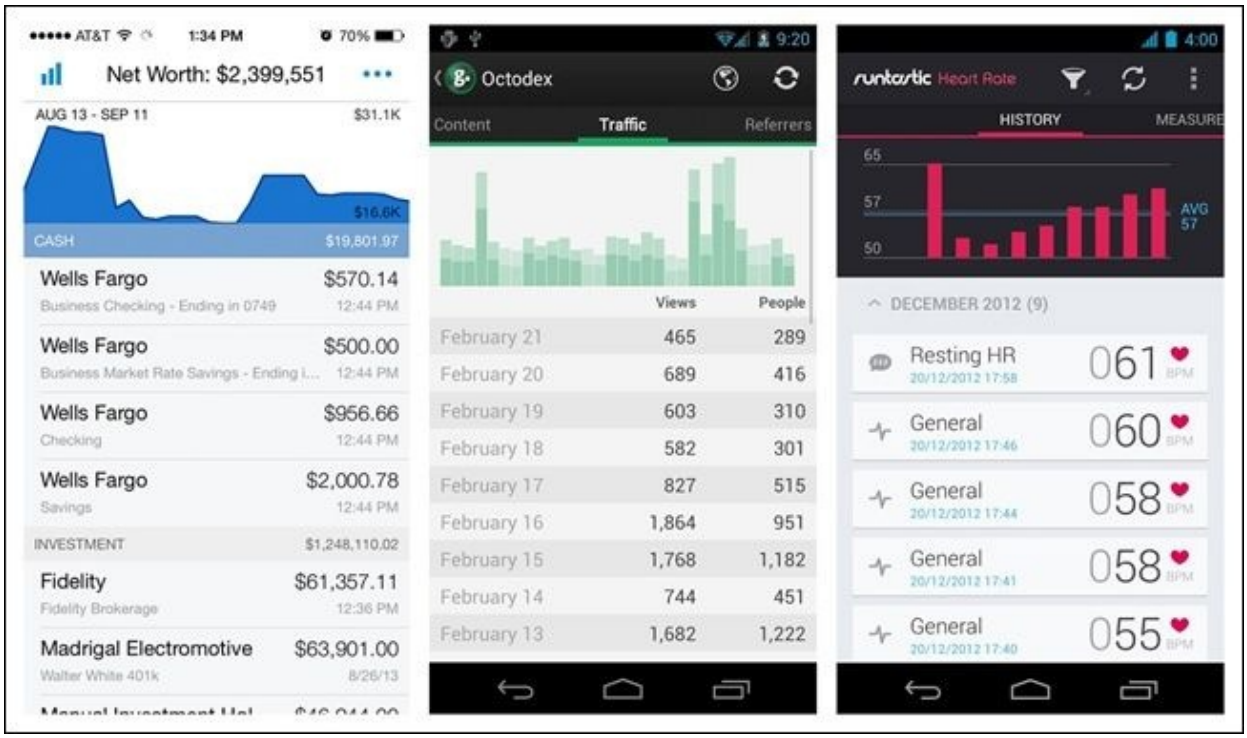



*Figure 3-9. Personal Capital for iOS and Runtastic Heart Rate for Android: Overviews plus Data*

BillGuard's overview is interactive. In the example, tapping "VS. AUG" draws the comparison line and shows the difference in the amount spent.

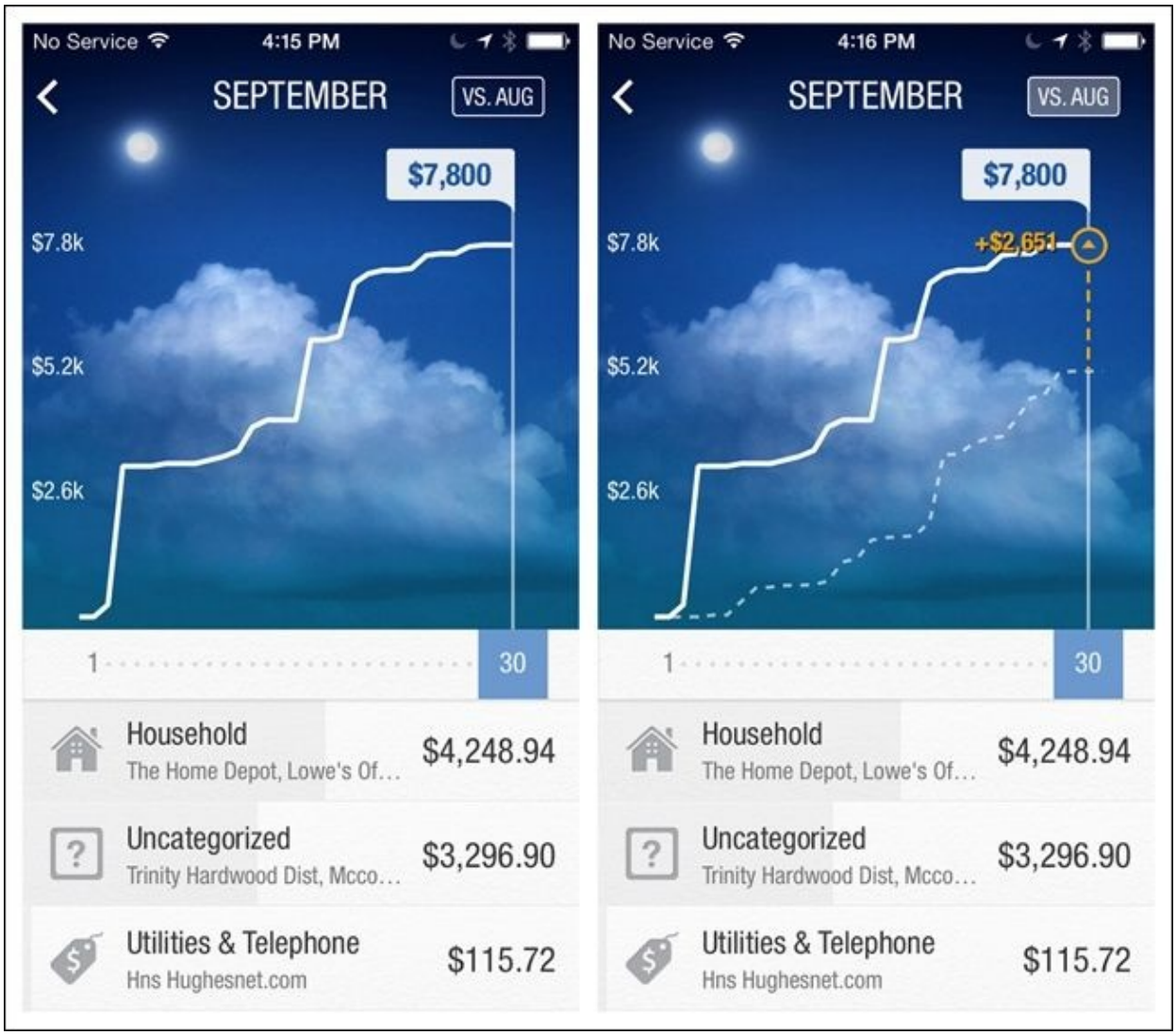



*Figure 3-10. BillGuard for iOS: interactive overview*

If the overview is visualized as a pie or donut chart and the chart and legend take up the whole screen, the Drill Down pattern may work better, as shown with Roambi. See Chapter 6 for more examples.

**NOTE**

Position the overview above the individual rows of data. If the chart overview is too large, consider the Drill Down pattern instead. See Chapter 6 for tips on designing effective charts.

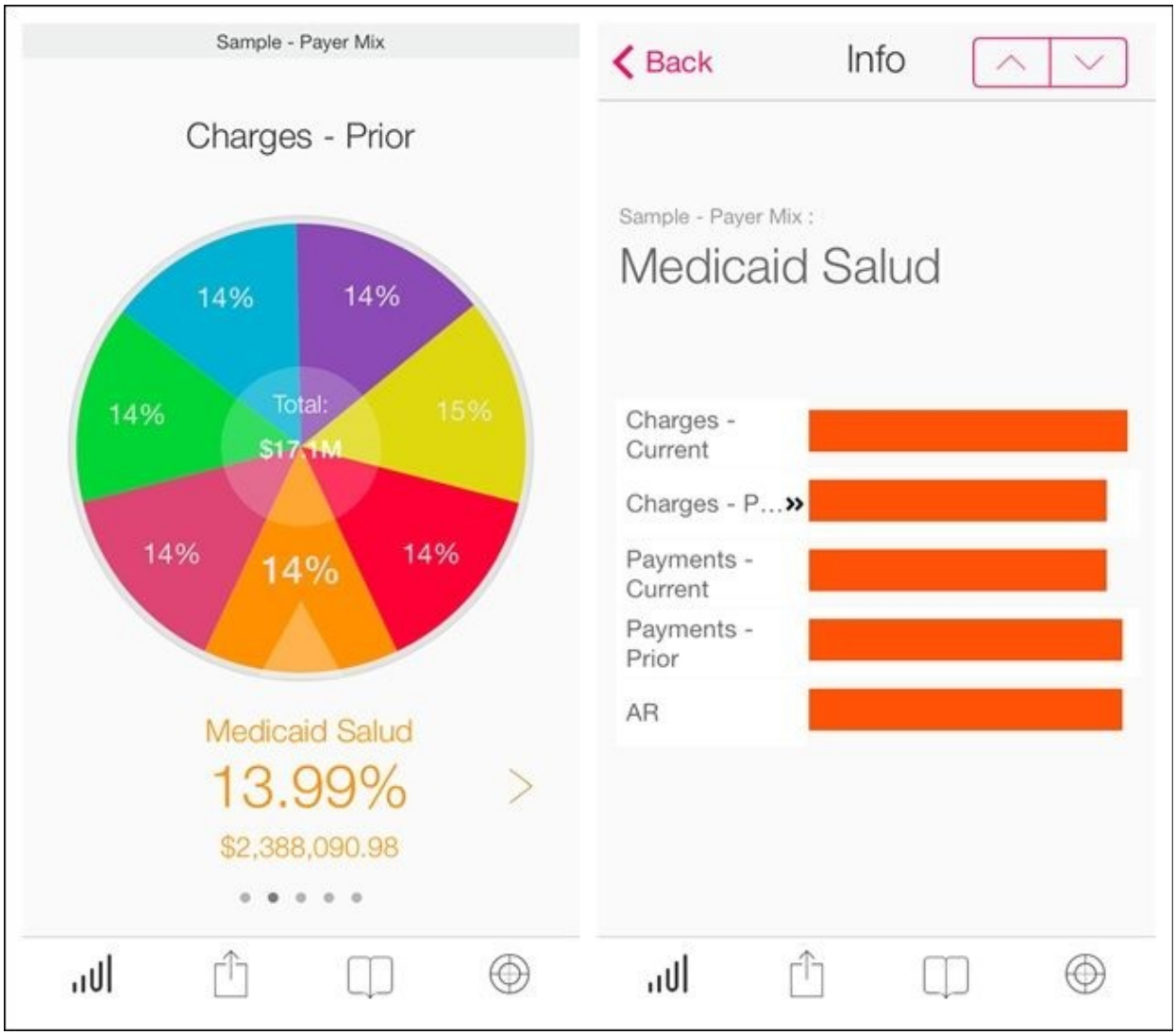



*Figure 3-11. Roambi for iOS: chart consumes entire screen, so Drill Down works better than Overview plus Data*

# Grouped Rows

Row grouping can make it easier to read a table's data. Grouped Rows might serve as section headers, like transactions grouped by year (as in the Zillow Mortgage Calculator) or by category (as in Mint).

> **NOTE**
>
> Visually differentiate the summary rows from the other rows in the table. In general, the summary rows should be thin and subtle.

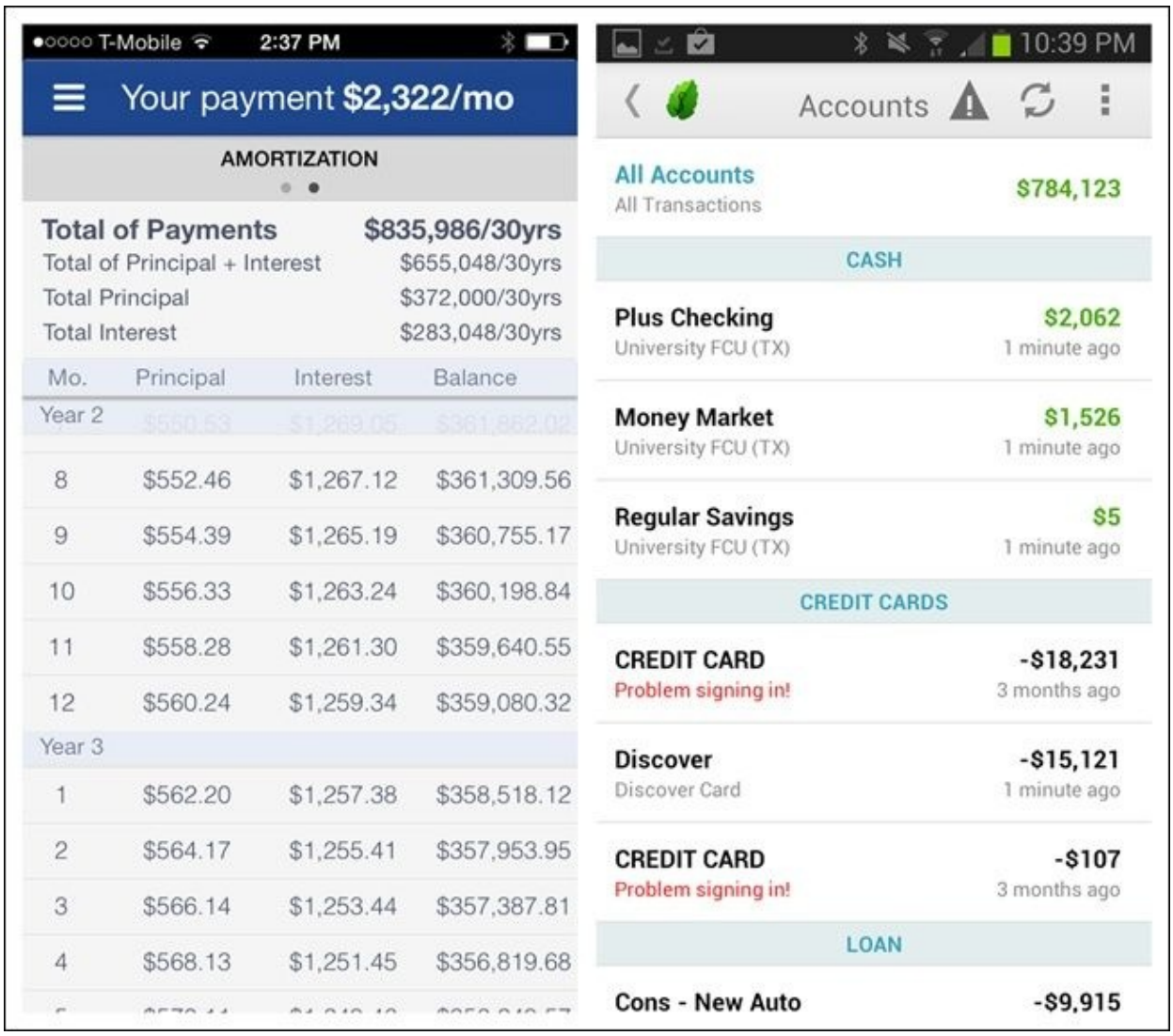



*Figure 3-12. Zillow Mortgage Calculator and Mint for Android: Grouped Rows*

## Table with Visual Indicators

In the Table with Visual Indicators pattern, Sparklines and icons can enhance the table's information display, but you must be careful not to overload the table with chart junk, like in the Sleep Charts design.


*Figure 3-13. Sleep Charts for Android: don't overload designs with chart junk*

On the other hand, the visual design of Roambi Sales Reports enhances the data rather than obscuring it. See Chapter 6 for more information about using Sparklines in mobile apps.

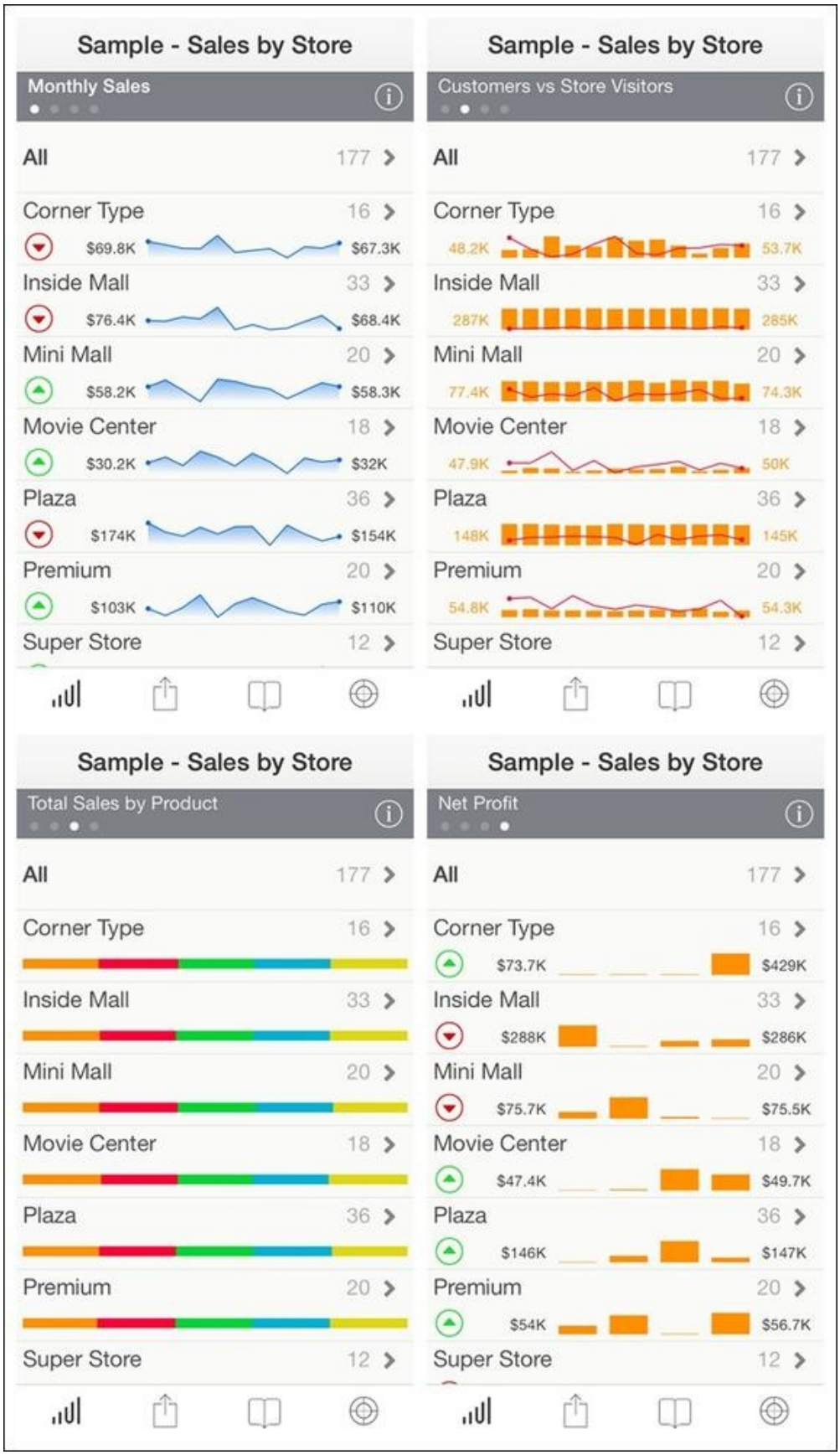

Sample - Sales by Store
Monthly Sales
All 177
Corner Type 16
$69.8K $67.3K
Inside Mall 33
$76.4K $68.4K
Mini Mall 20
$58.2K $58.3K
Movie Center 18
$30.2K $32K
Plaza 36
$174K $154K
Premium 20
$103K $110K
Super Store 12
Sample - Sales by Store
Customers vs Store Visitors
All 177
Corner Type 16
48.2K 53.7K
Inside Mall 33
287K 285K
Mini Mall 20
77.4K 74.3K
Movie Center 18
47.9K 50K
Plaza 36
148K 145K
Premium 20
54.8K 54.3K
Super Store 12
Sample - Sales by Store
Total Sales by Product
All 177
Corner Type 16
Inside Mall 33
Mini Mall 20
Movie Center 18
Plaza 36
Premium 20
Super Store 12
Sample - Sales by Store
Net Profit
All 177
Corner Type 16
$73.7K $429K
Inside Mall 33
$288K $286K
Mini Mall 20
$75.7K $75.5K
Movie Center 18
$47.4K $49.7K
Plaza 36
$146K $147K
Premium 20
$54K $56.7K
Super Store 12

*Figure 3-14. Roambi Sales Reports for iOS: design elements, including Sparklines, enhance the data*

Choose Visual Indicators that are immediately recognizable, and once again, avoid using icons gratuitously. Compare the last two versions of PayPal for iOS. The newer iOS 7 design has more space for the transaction content without the excess icons.

> **NOTE**
>
> Use Visual Indicators only if they will enhance the table data; avoid them otherwise.

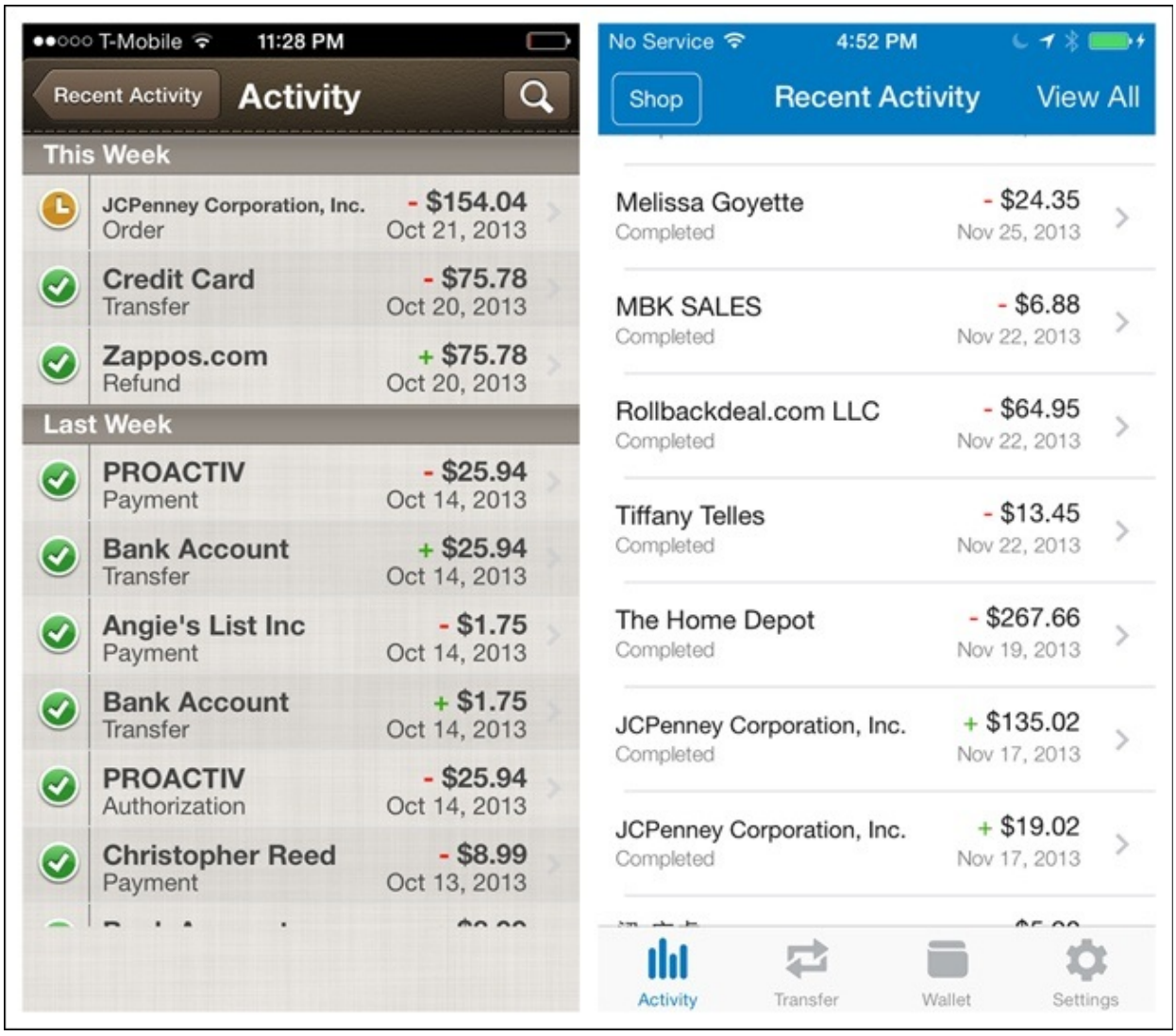



*Figure 3-15. PayPal for iOS 6 and iOS 7: newer design loses the icons and gains readability*

# Editable Table

Editable Tables in mobile interfaces are found almost exclusively in spreadsheet applications like Microsoft Office, Google Drive, and Numbers. Many of the same guidelines for Editable Tables on the Web apply to mobile:

- Clearly indicate the cell and/or row selected.
- If there is a specific format for the cell, offer the appropriate editor (selector, spinner, color picker, date picker, etc.).
- It's not necessary to provide confirmation feedback with each change, unless the change prompts an error message; otherwise, provide confirmations just on Save.

In Office for Windows Phone, single-tapping a cell presents cell menu options. Tapping the function icon (the *fx* at the top left) opens the functions menu.


*Figure 3-16. Office for Windows Phone: tapping fx opens functions menu*

Unlike on the desktop, however, Editable Tables on mobile devices are not appropriate for extensive data entry, mainly because mobile keyboards don't support screen navigation. Specifically, you can't tab from cell to cell.

> **NOTE**
>
> Follow web and desktop design best practices for Editable Tables where appropriate, but avoid using Editable Tables for bulk data entry, or where extensive editing could be necessary.

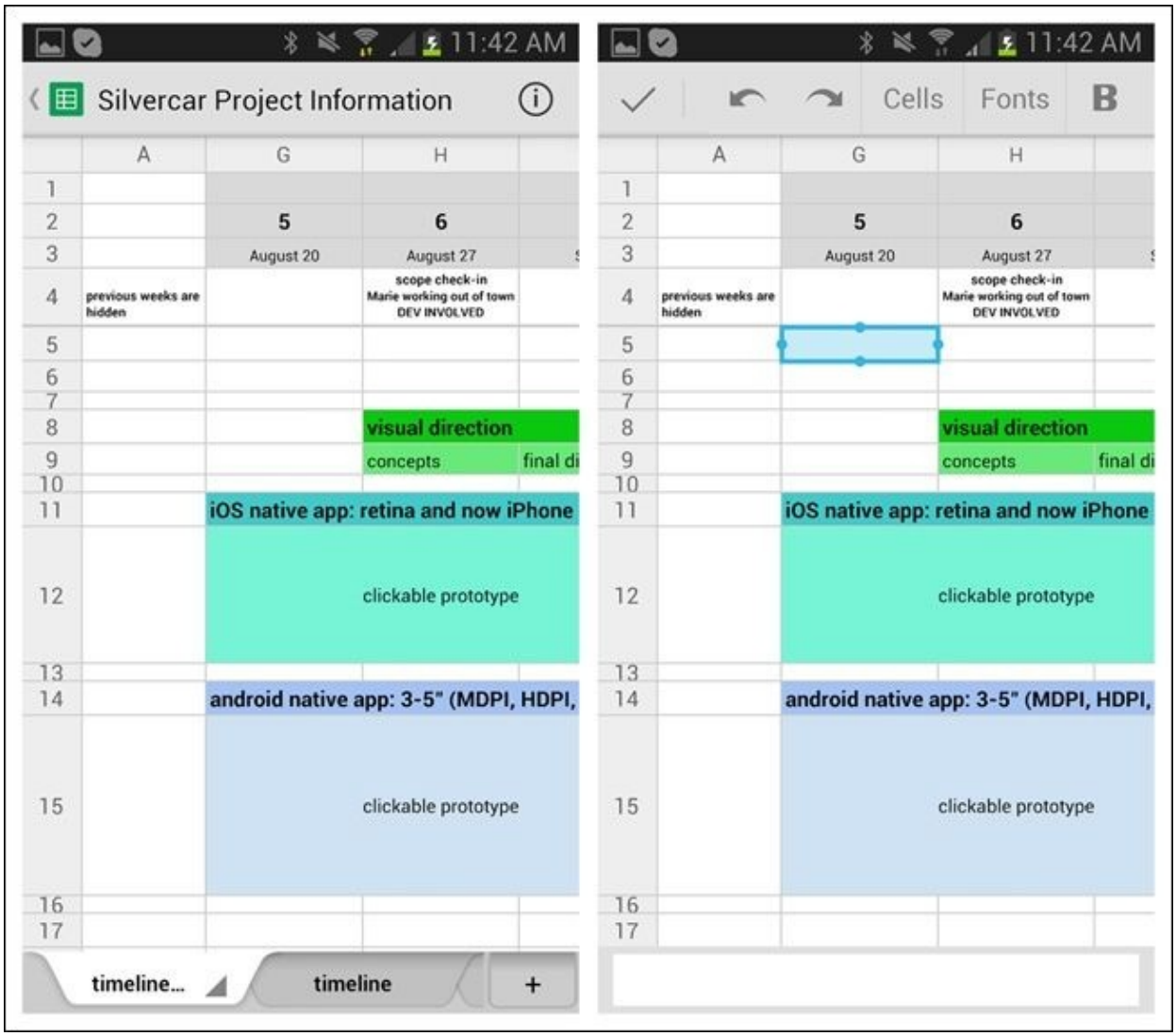



*Figure 3-17. Google Drive for Android: selecting a cell reveals the input field across the footer*

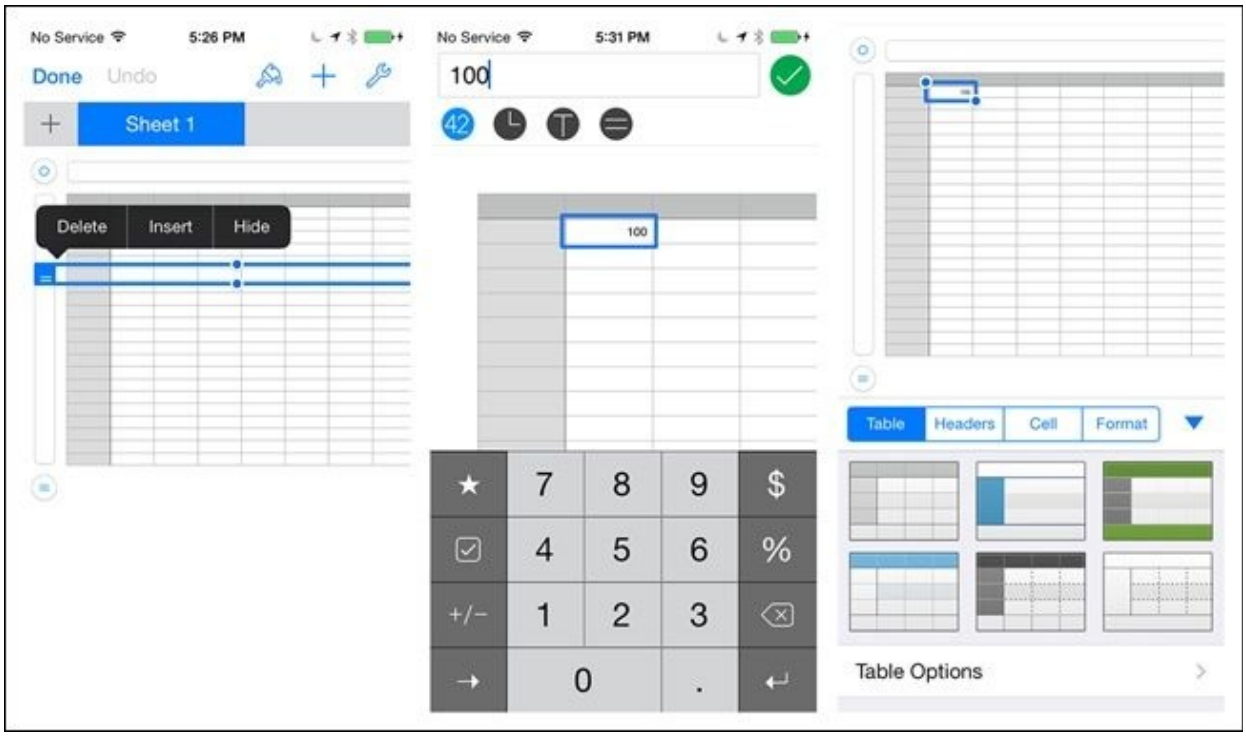



*Figure 3-18. Numbers for iOS: separate view and edit modes for data entry and formatting*

# Chapter 4. Search, Sort, and Filter


*Search Patterns*

Implicit Search; Explicit Search; Auto-Complete; Scoped Search; Dynamic Search; Saved, Recent, and Popular; Search Form; Search Results

*Sort Patterns*

Onscreen Sort, Sort Overlay, Sort Form

*Filter Patterns*

Onscreen Filter, Filter Overlay, Filter Form, Filter Drawer, Gesture-Based Filter

Have you looked at AutoDirect, Greensheet, or a similar freebie print ad circular lately? You'll see page after page of classified ads organized in very broad categories, like Trucks, Cars, and Appliances. You practically have to read the whole rag from beginning to end to see if there are any ads that interest you. This should give you a new appreciation for three basic online tools: search, sort, and filter.

We take them for granted in our digital world. But there are some nuances to getting search, sort, and filter right in a mobile application. This chapter explores

more than a dozen different approaches.

# Search Patterns

Josh Clark, author of *Tapworthy: Designing Great iPhone Apps* (O'Reilly, 2010), says that in touch interface design, "buttons are a hack." That's because touch gives us more direct ways of interacting with content, like *pull to refresh* and *pinch to zoom*. Why use an abstract button when a physical gesture would be more intuitive?

We should think of mobile search the same way. As our apps become more "aware" of our individual needs, they can reduce instances where we have to explicitly search for what's relevant to us.

For instance, Google "knows" that I have a meeting downtown and tells me when to leave and how to get there. It also knows that I often take the kids to the movies on Saturday, so it "suggests" some shows at our favorite theater. And when I am near a mall that has some of my favorite stores, RetailMeNot shows me all the deals they are running.

This emerging pattern is called Implicit Search. We'll explore it further, and take a fresh look at the classic search, sort, and filter techniques covered in the earlier edition.

In this section we'll look at search patterns specific to mobile applications, including patterns for Implicit Search; Explicit Search; Auto-Complete; Scoped Search; Dynamic Search; Saved, Recent, and Popular; Search Forms; and Search Results.

## Implicit Search

One night before I had a meeting with a new client, I Googled the client's address on my PC. As I prepared to leave the next morning, I opened the new version of Google Maps and typed in the first number of the address, and Maps immediately knew where I wanted to go.

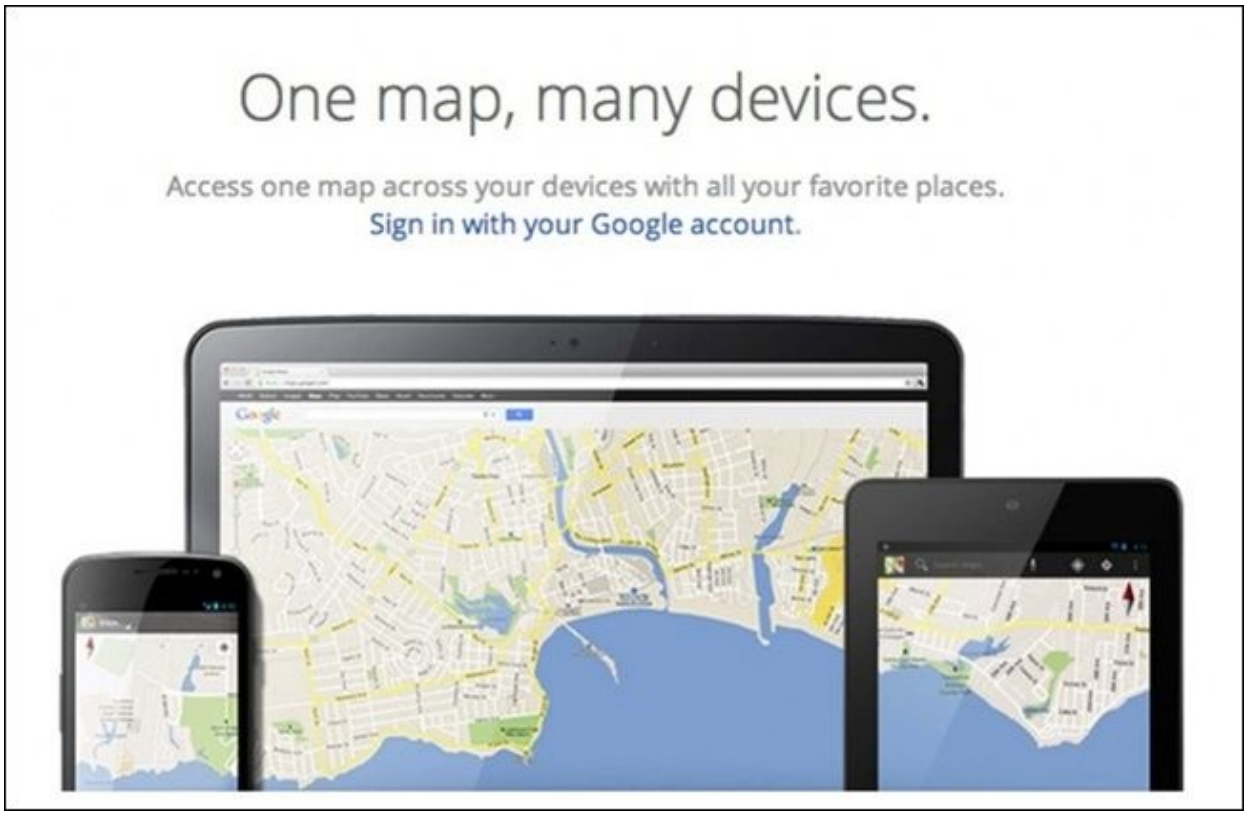



*Figure 4-1. Google Maps: cross-platform integration*

With the new app, Google synchronized my information across channels and performed an Implicit Search for me. *Wow*, I thought, *that was helpful*. But then Google Now really took Implicit Search to a new level, showing me where my following meeting was located, when I should leave, and how to get there. I rarely ever need to explicitly search in Google Maps now.

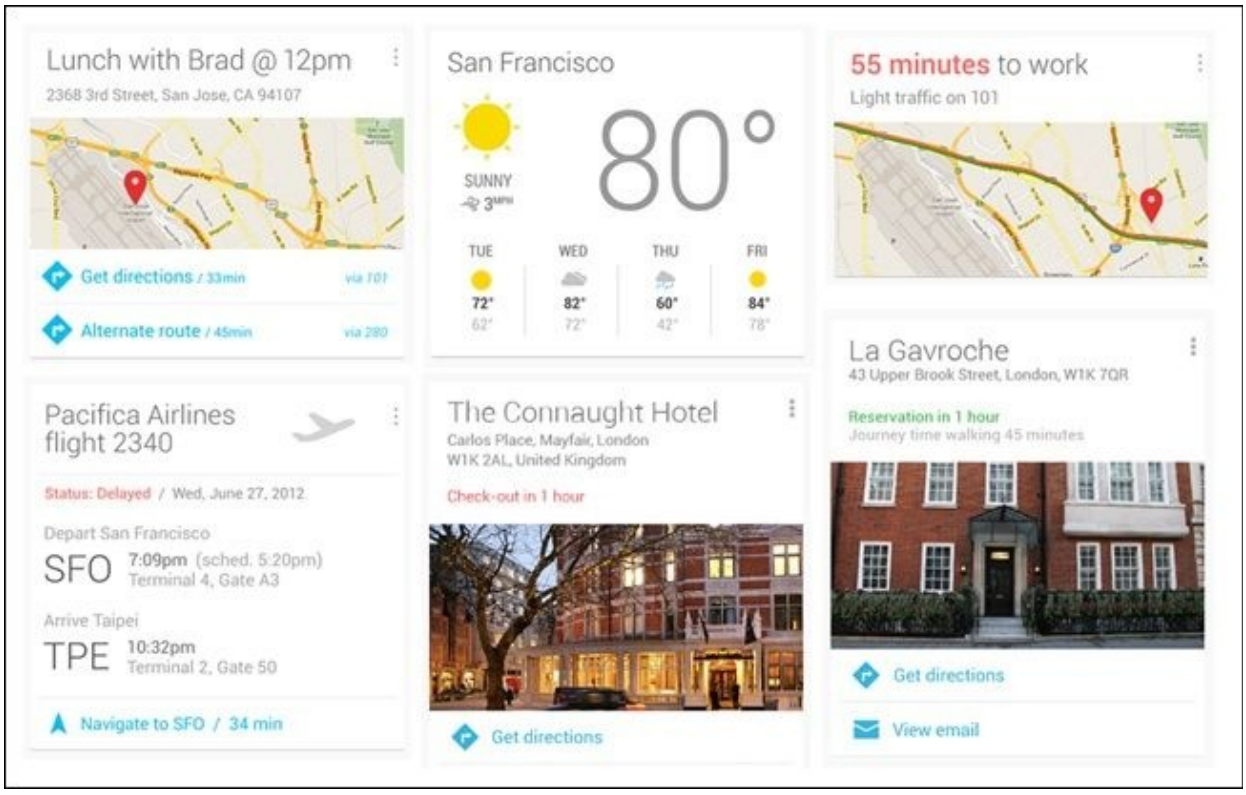



*Figure 4-2. Google Now: taking Implicit Search to a new level*

I had a similar delightful experience with the RetailMeNot application: as I was driving around running errands, I heard a “cha-ching” from my purse. Wondering why my phone sounded like a cash register, I saw a RetailMeNot notification telling me that my favorite stores at the mall nearby were running some great deals. Once I was in a store, the app surfaced in-store coupons I could use to save on my purchase. This relevant information found me, without my having to look for it.

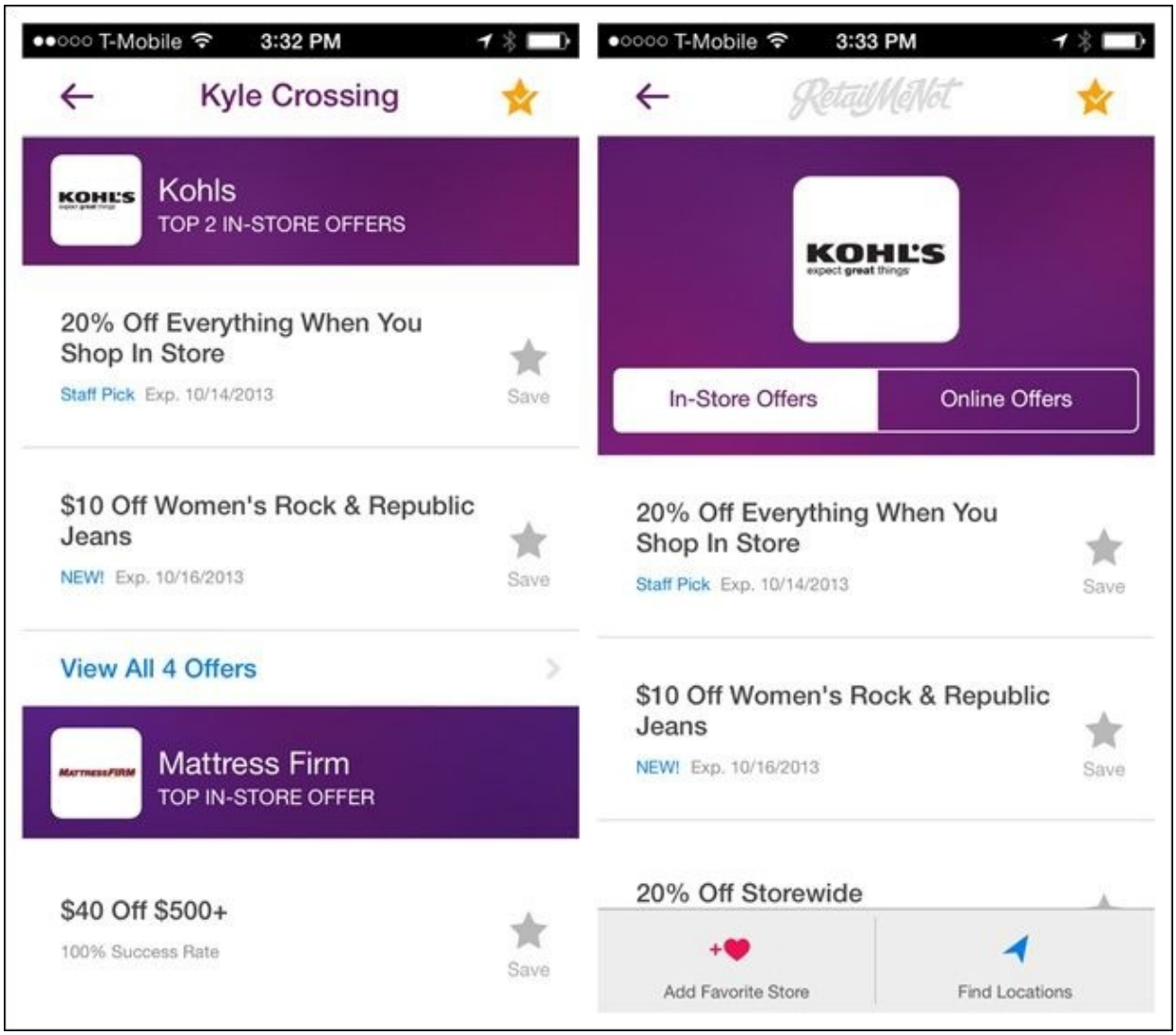



*Figure 4-3. RetailMeNot for iOS: proximity-driven Implicit Search*

For me, this is a much more efficient way to find deals than explicitly searching for a store and then searching to see if it is running any special offers. And it is definitely working for RetailMeNot, who saw a fourfold increase in engagement with the introduction of this "geofencing" feature.

There are many other examples of Implicit Search. If location-based services are enabled in Zillow and similar real estate apps, the apps immediately reveal all the properties for sale or rent in the area. Badoo, a social meetup tool, uses the same technique to show nearby people looking to connect.

Foodspotting and Hungry Now highlight nearby restaurants on their landing pages. The former uses a carousel to show pictures and details about each restaurant, encouraging the user to start rating her dining experiences, whereas

the latter shows the restaurants on a map.

> **NOTE**
>
> Consider the possibilities of Implicit Search before choosing another search pattern. It could introduce an element of delight to the user experience.

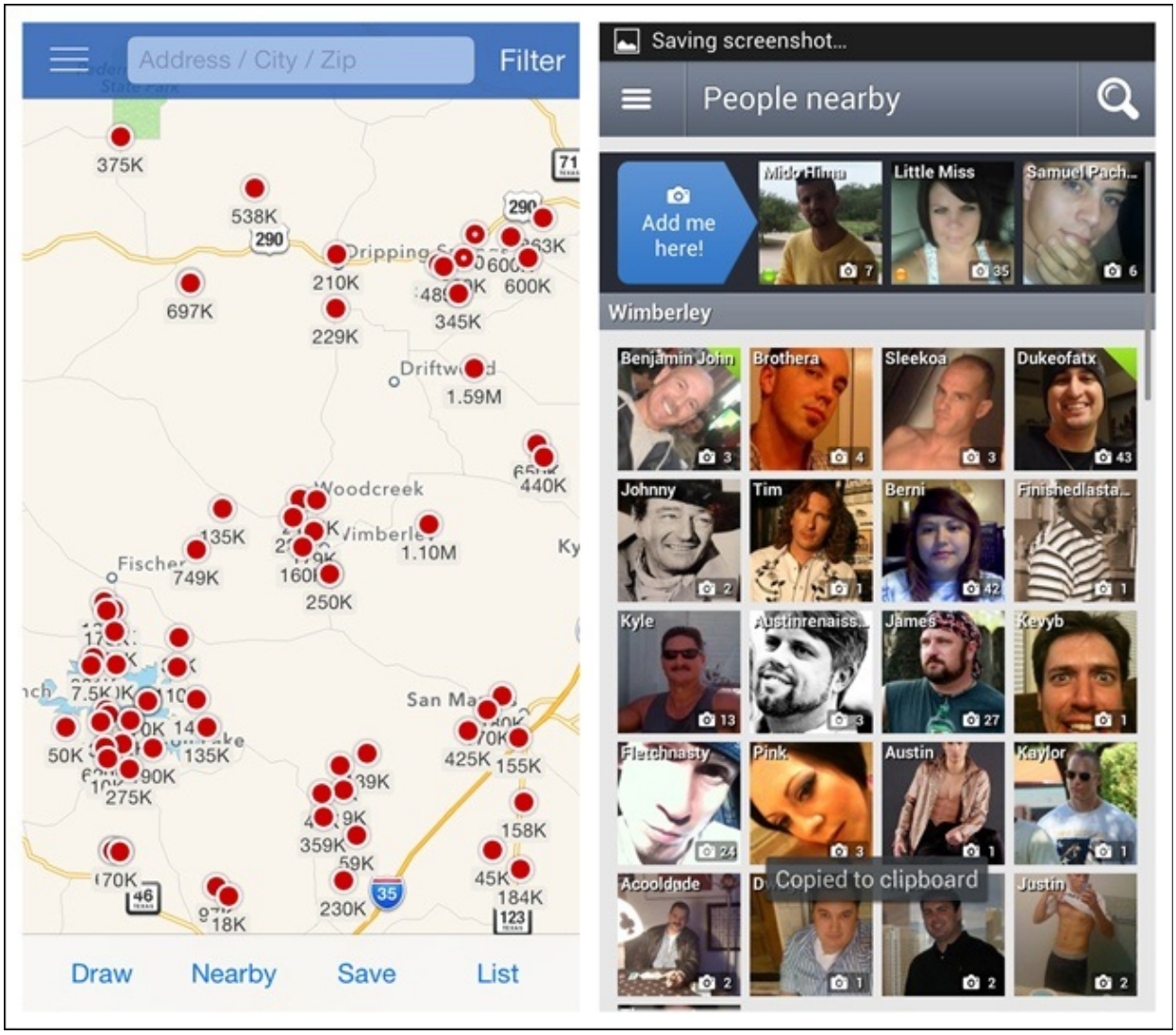



*Figure 4-4. Zillow for iOS and Badoo for Android: location-based Implicit Search*

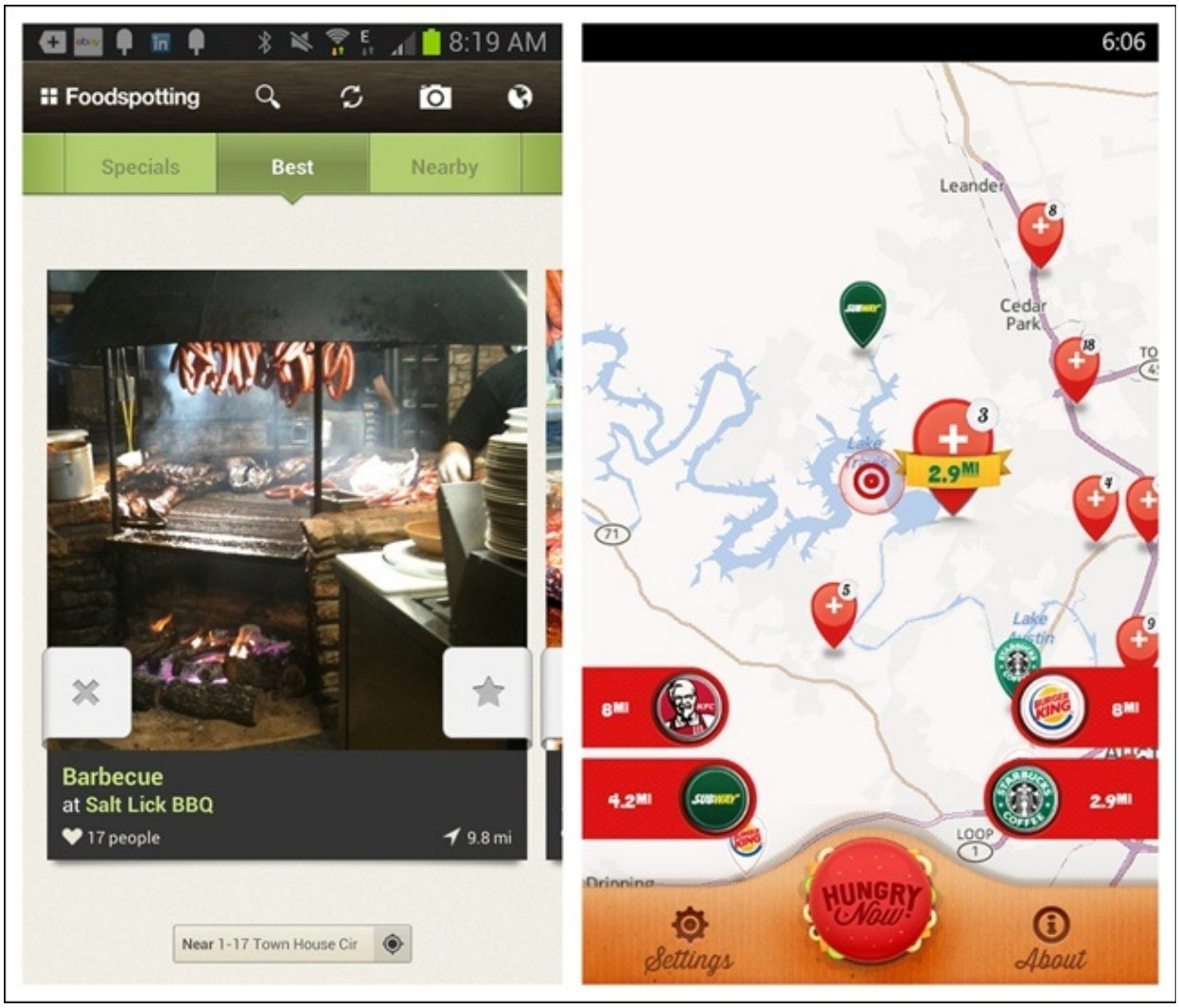



*Figure 4-5. Foodspotting for Android and Hungry Now for Windows Phone: more location-based results*

Apps like Wikitude and Layar use a combination of the device's location and the accelerometer to create an "augmented reality" for the user — no search terms necessary.


*Figure 4-6. Wikitude World Browser: Implicit Search uses augmented reality*

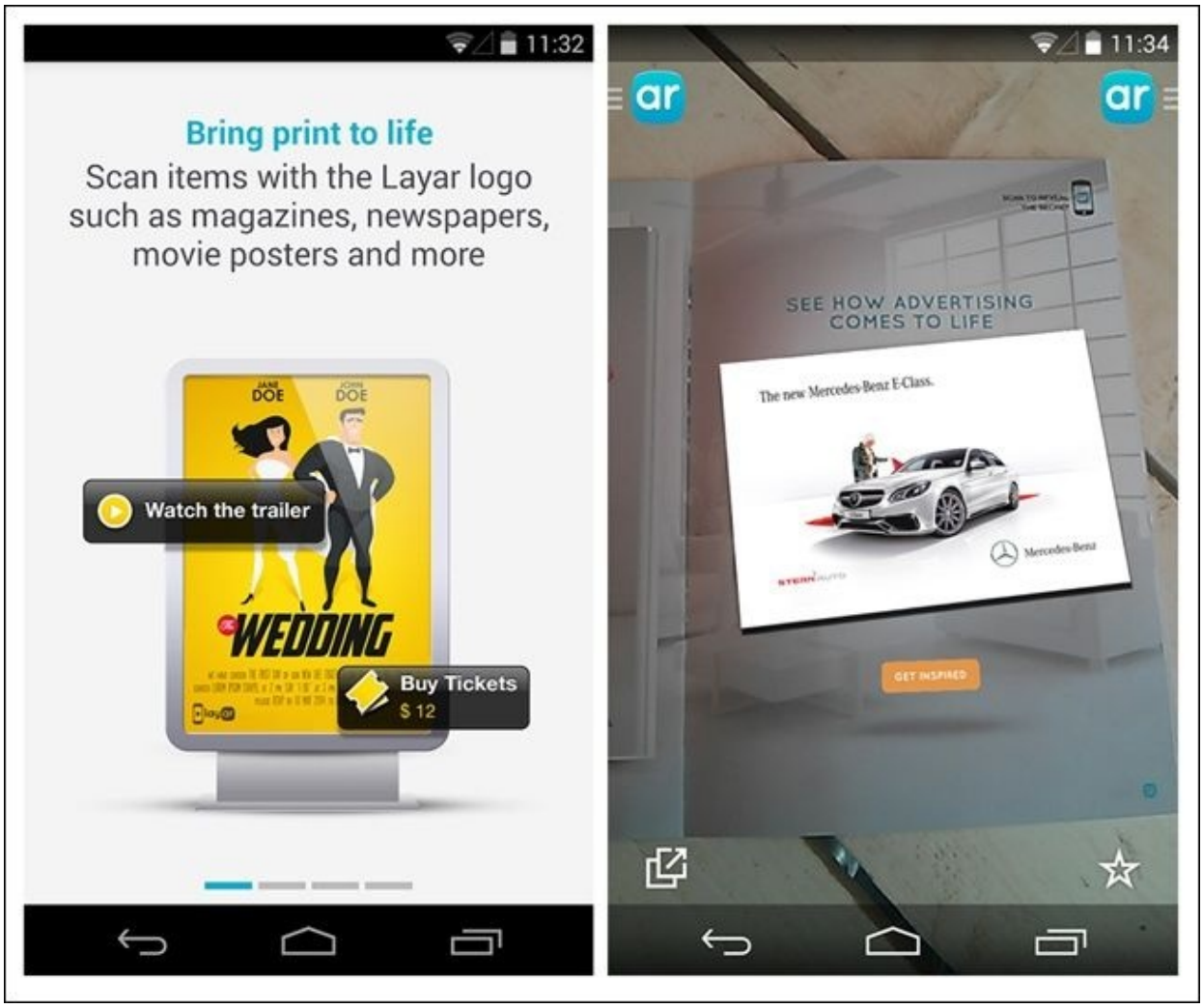



*Figure 4-7. Layar for Android: another way to use augmented reality for Implicit Search*

## Explicit Search

While Implicit Search can create richer user experiences, there are still valid cases for the Explicit Search pattern. As you might guess, Explicit Search relies on explicit user action to perform the search and get results.

The eBay app on Windows Phone 8 is a simple example. Type the search term, then tap on the search icon or hit Return on the keyboard to see the results. But by being just a little smarter, eBay for Android manages to be a lot more helpful.

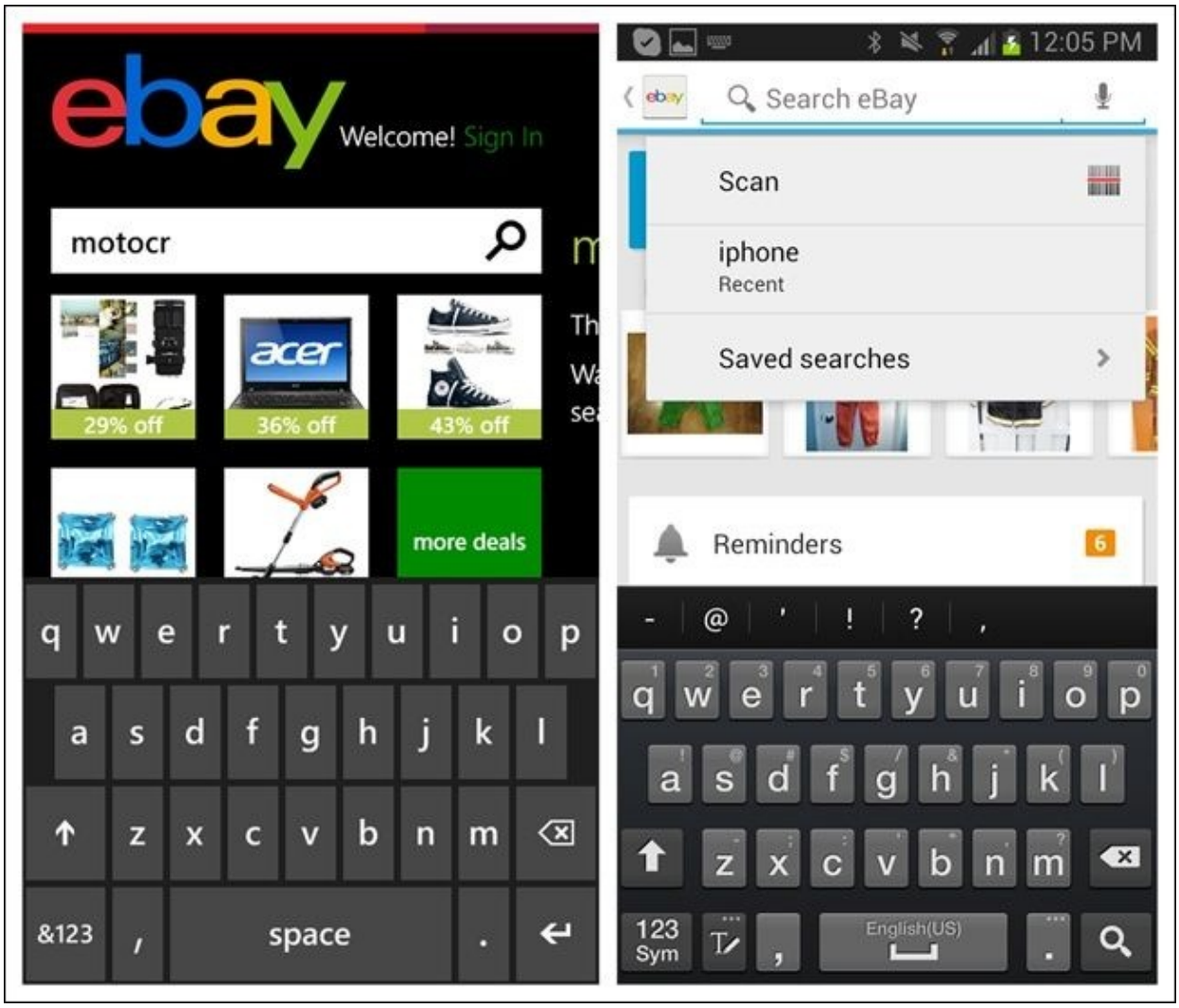



*Figure 4-8. eBay for Windows Phone and Android*

Tapping in the search field in the Android app provides a number of options: scan an item instead of typing it, reuse a recent search, or access saved searches. Once the user starts typing, the app surfaces suggestions that match the text entered. Minimizing the need to type in this way increases search efficiency and reduces the chance of error. Also see the next pattern, Search with Auto-Complete.

As eBay for Android suggests, Explicit Search needn't be confined to text entry. Amazon and RedLaser offer photo and barcode scanning options for quick in-store lookups. These mobile-friendly search features have helped drive the showrooming phenomenon (*http://en.wikipedia.org/wiki/Showrooming*) that has changed the retail shopping dynamic in recent years.

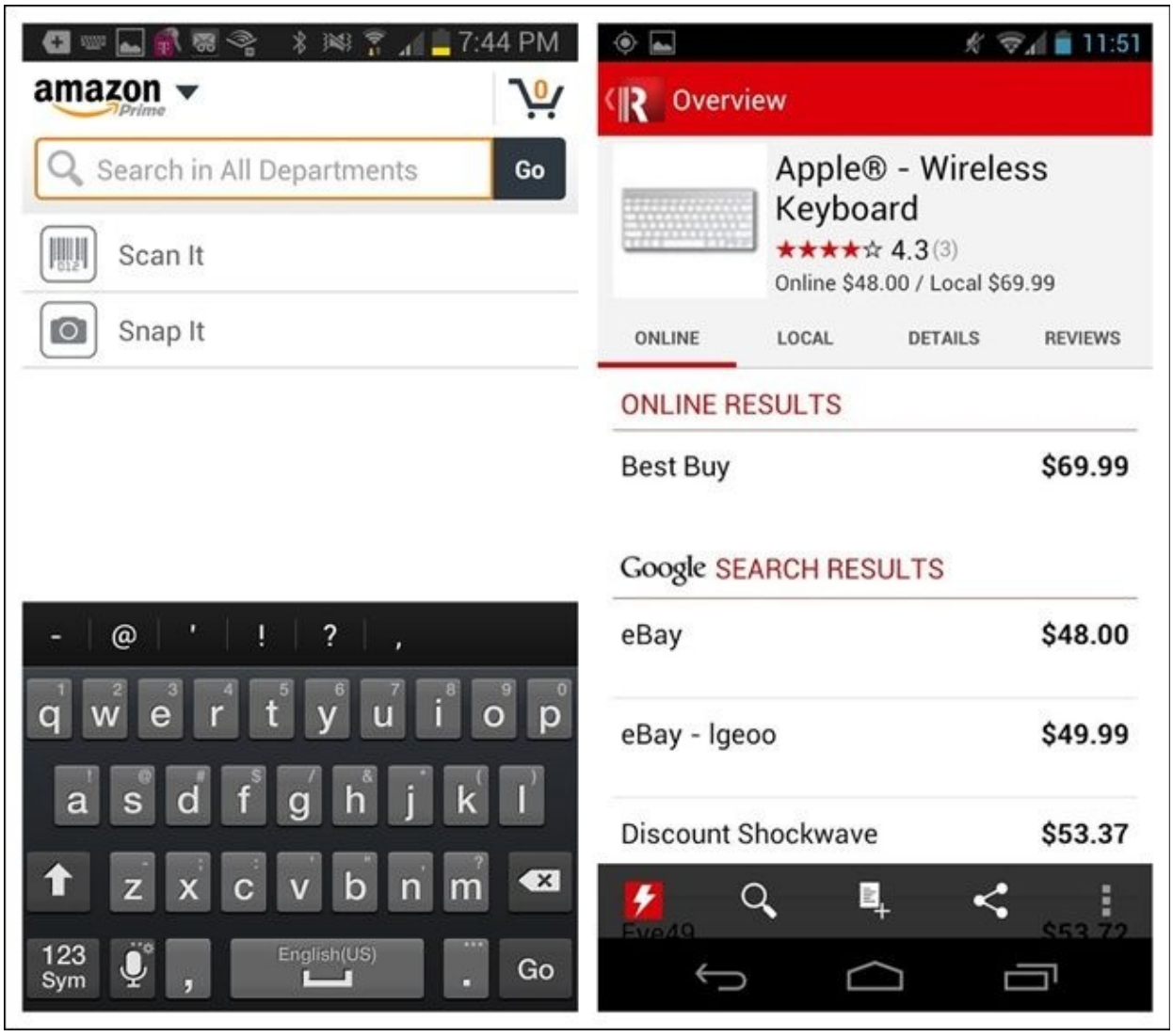



*Figure 4-9. Amazon and RedLaser for Android: Explicit Search via barcode scanning*

Voice-based search also can be useful, but it is dependent on the quality of the voice recognition program. We've all heard funny stories of Siri wildly misunderstanding user questions (see the screenshots from WhySiriWhy.com), but voice search has already come a long way, and I expect its refinement to continue.


*Figure 4-10. WhySiriWhy.com: a site devoted to voice searches gone wrong*

Another option for Explicit Search is via direct manipulation. Trulia and Soundwave both offer options to physically draw search boundaries on a map.

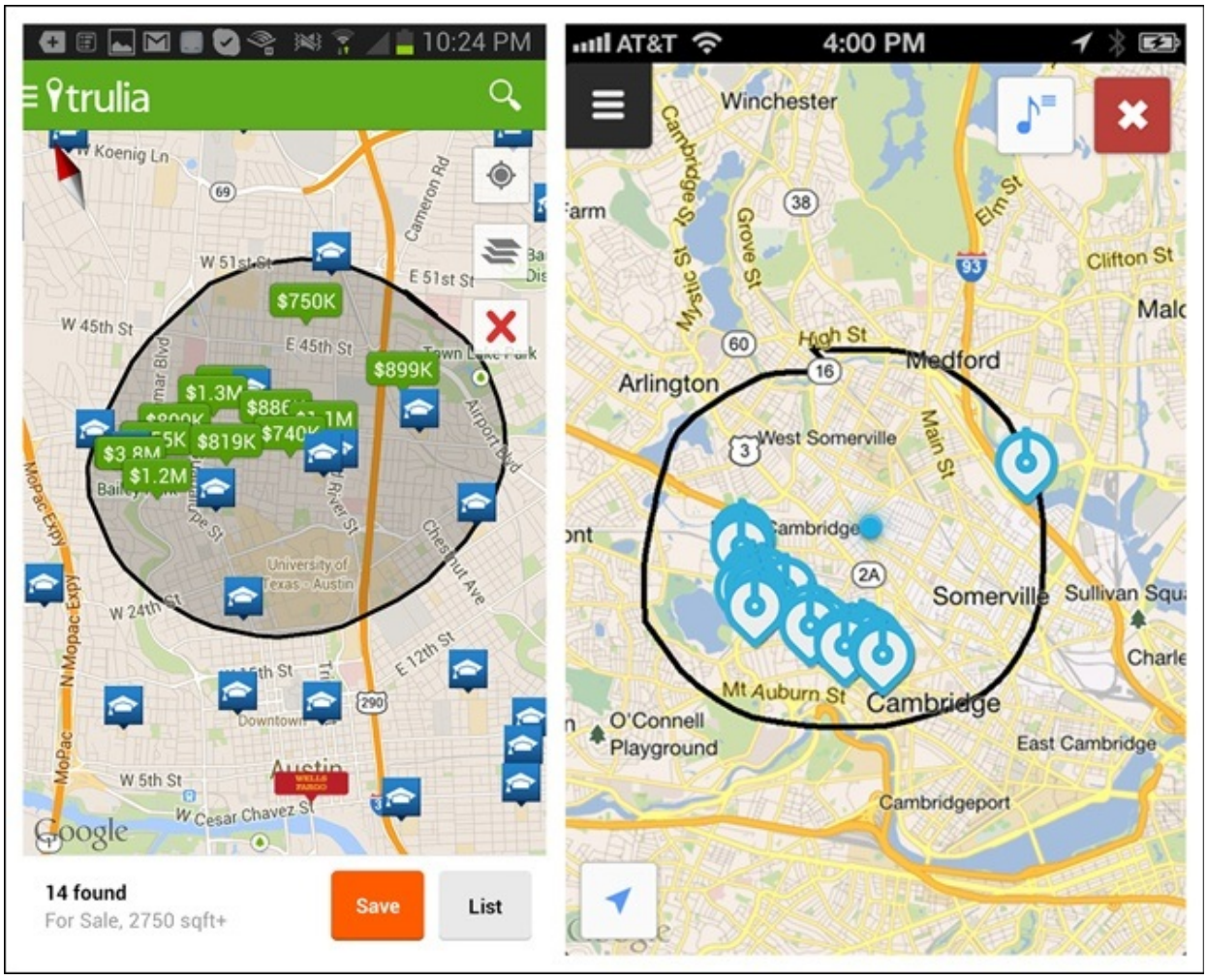



*Figure 4-11. Trulia for Android and Soundwave for iOS use gestures to draw search boundaries*

One more emerging pattern in Explicit Search is Modal Search. In this example from Target, the search field is shown in the navigation drawer, but as soon as it is tapped, “search mode” displays a full search screen with keyboard.

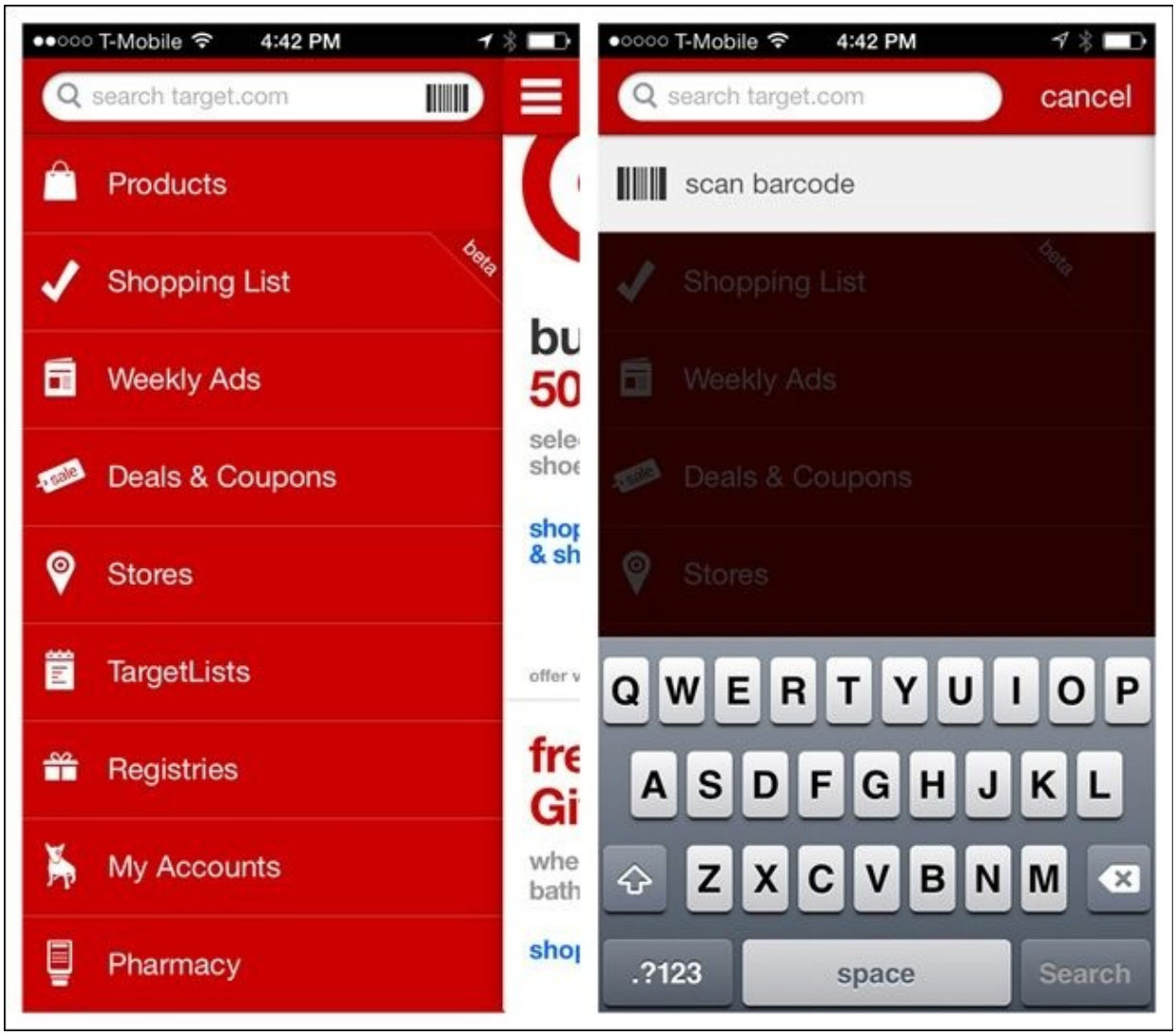



*Figure 4-12. Target for iOS: tapping on the search field opens a new search screen*

TripAdvisor and RetailMeNot use the same pattern, but in these examples, search lives on the home page instead of in a navigation drawer.

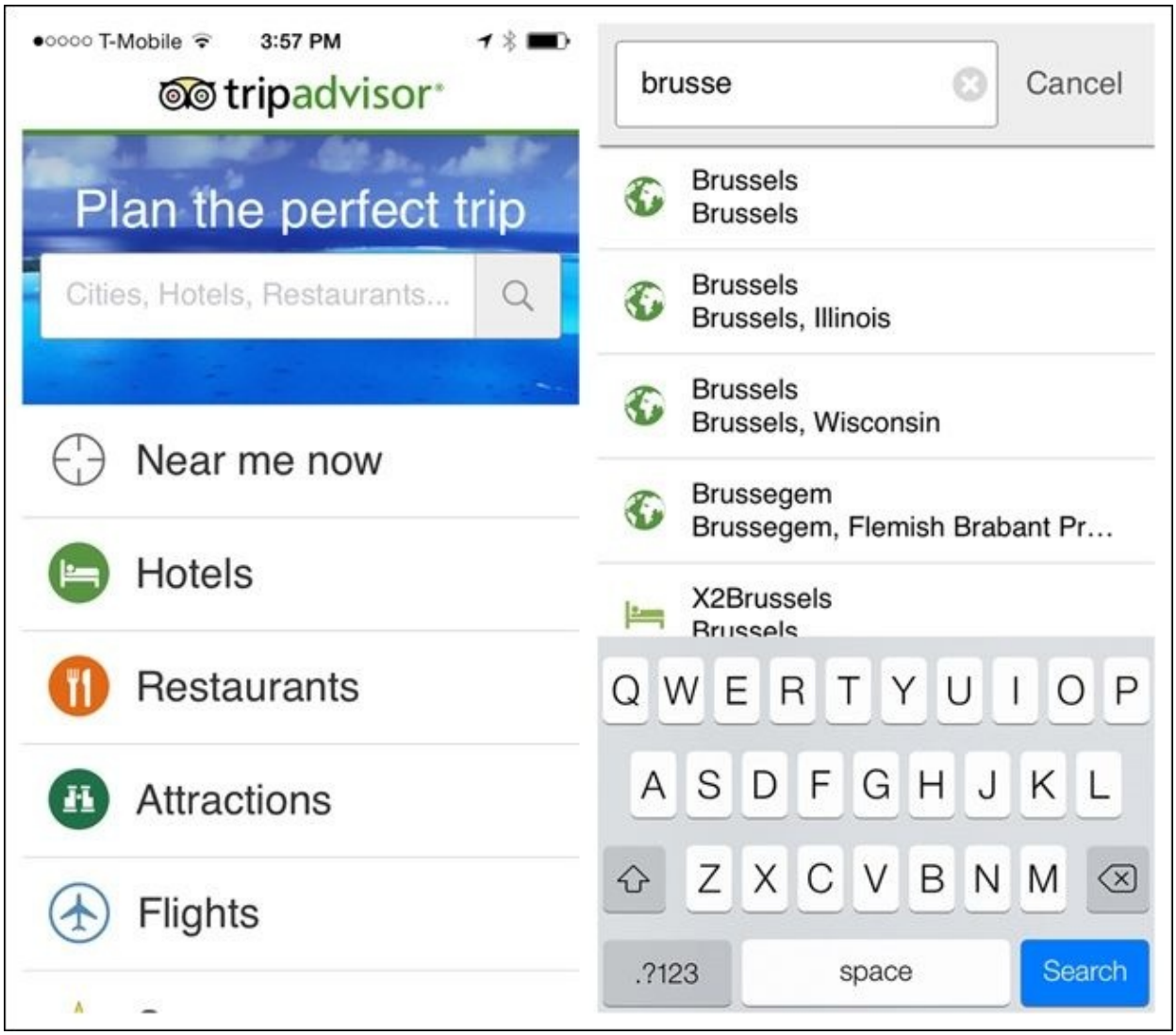



*Figure 4-13. TripAdvisor for iOS: Modal Search slides a search screen in from the right*

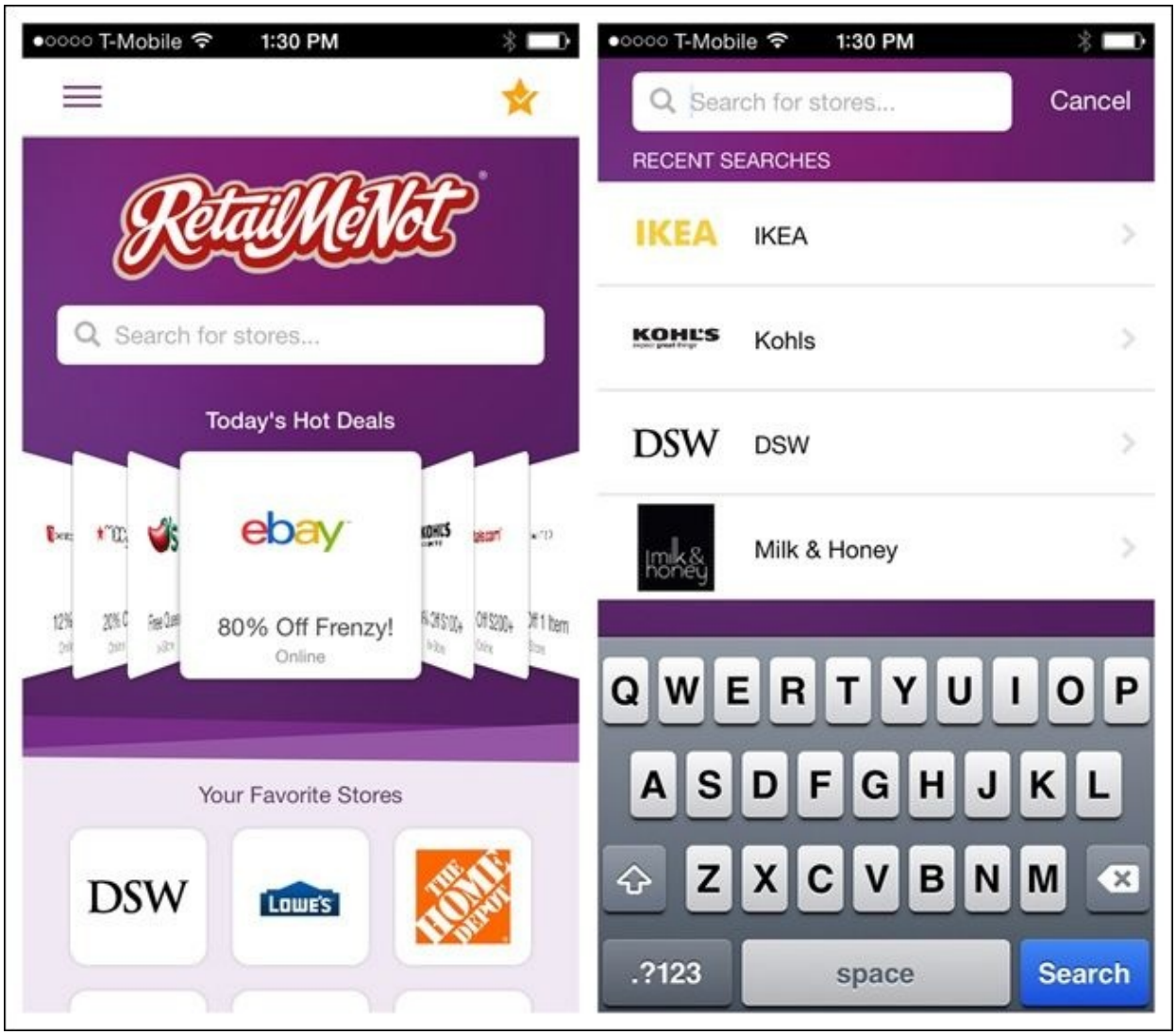



*Figure 4-14. RetailMeNot for iOS: the Modal Search screen slides up*

The main benefit of Modal Search is the additional real estate the full search screen offers. Just remember to offer a way back (e.g., Cancel) for users who decide not to search after all.

> **NOTE**
>
> Offer a Clear button and an option to cancel a Modal Search. Use feedback to show that the search is being performed (see Chapter 8).

## Search with Auto-Complete

At this stage in mobile design's maturity, Auto-Complete should be standard for Explicit Search. I was stunned that fewer than 10% of the Windows Phone 8 apps I researched had this functionality, even though these design patterns have

been around since 2004 on the Web, and even longer on desktop. Shockingly, even Dictionary.com on Windows Phone doesn't offer Auto-Complete, and who needs it more than someone using a dictionary app?


*Figure 4-15. Dictionary.com for Windows Phone and iOS: Auto-Complete should be standard*

Auto-Complete should be implemented so that as text is entered, a set of possible results surfaces. As the user continues to enter text, the search results narrow accordingly. A tap on any result initiates the search. Tapping a button (on the screen or keyboard) should also be an option to initiate search on the currently highlighted result.

Ideally, Auto-Complete results will display immediately, but a progress indicator ("Searching...") should be used to give feedback. Fidelity displays feedback where the results will eventually be displayed.

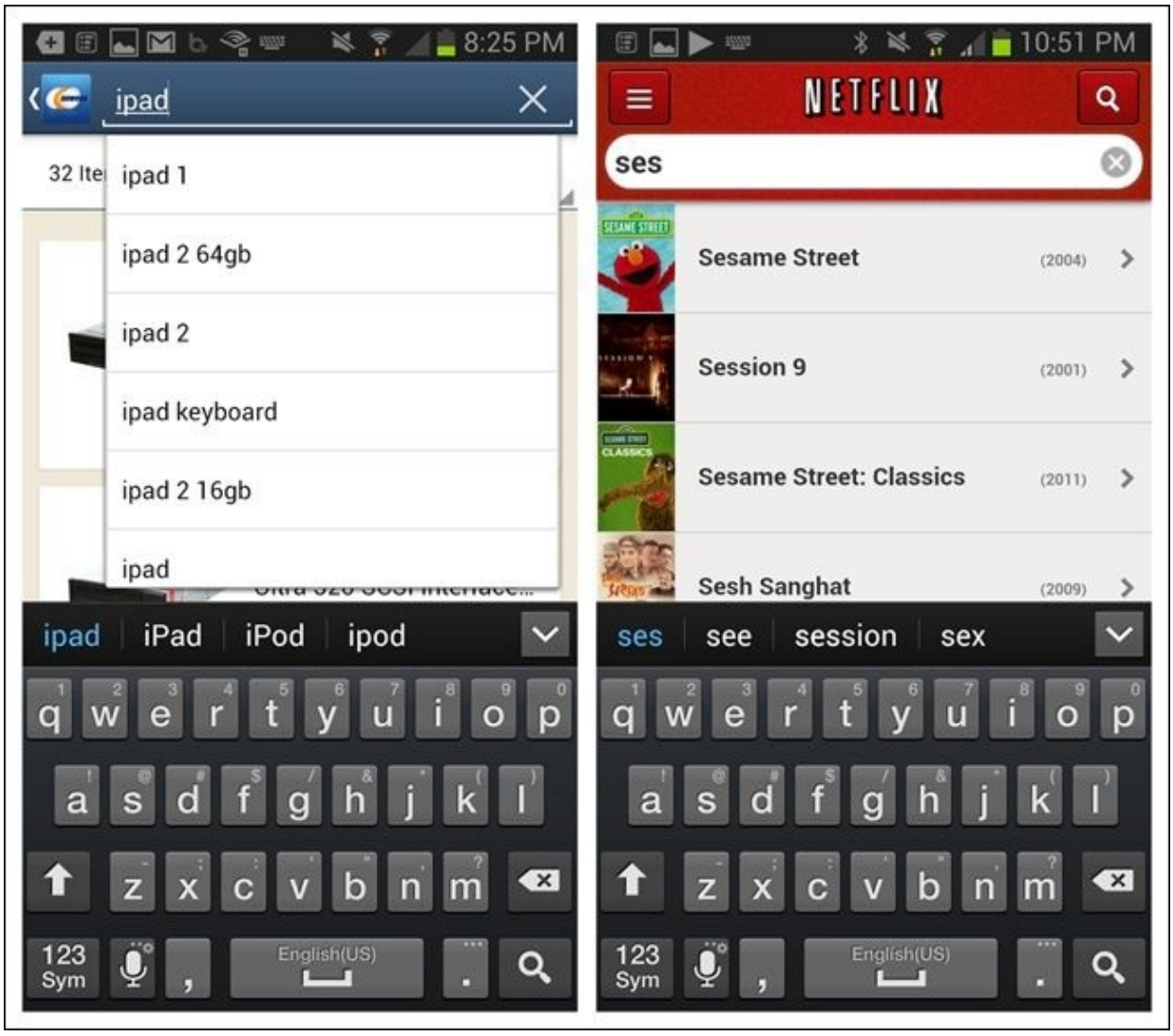



*Figure 4-16. Newegg and Netflix for Android: Auto-Complete can offer multiple ways to initiate search*

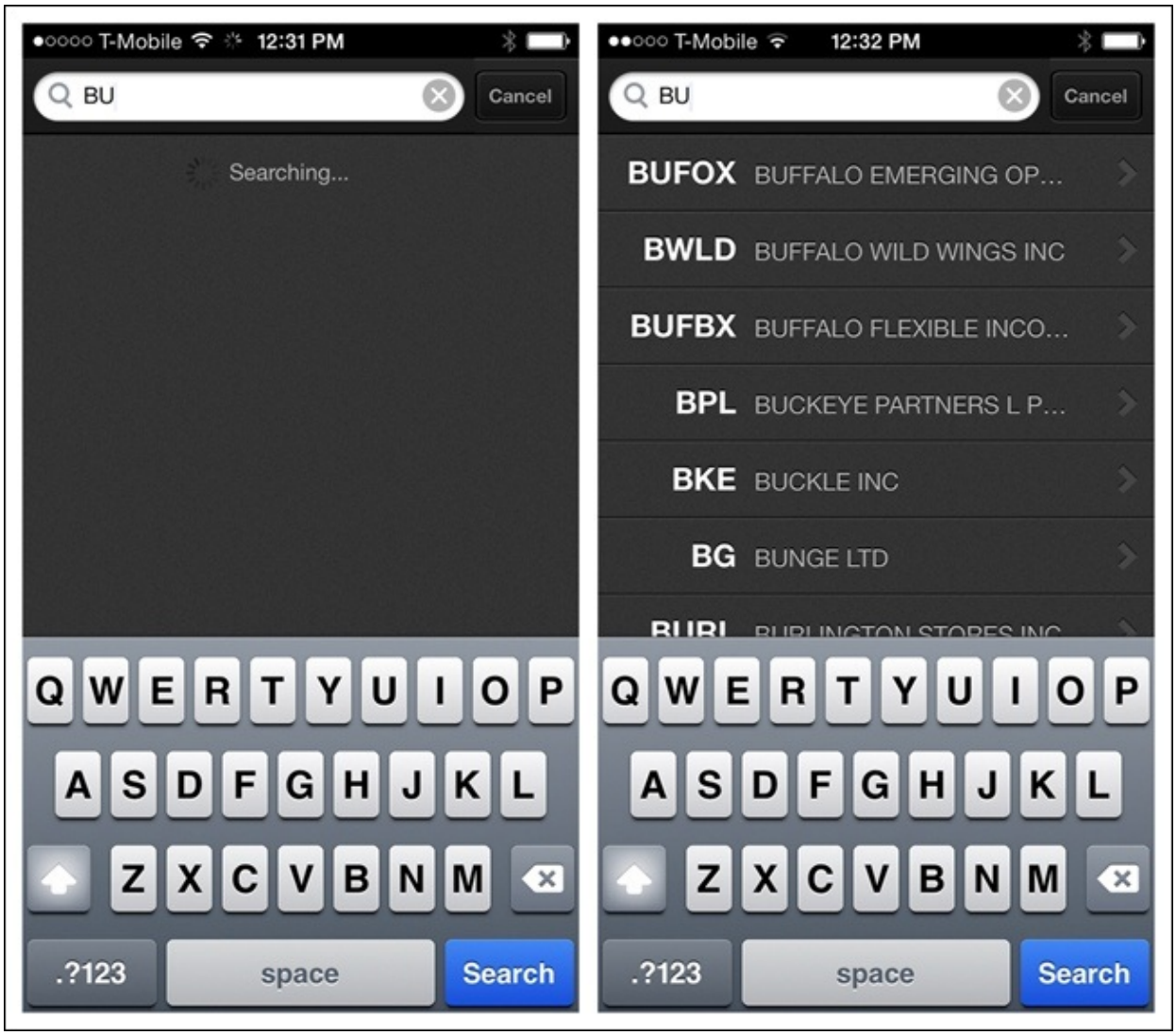



*Figure 4-17. Fidelity for iOS provides feedback for Auto-Complete progress below the search field*

Target's Android app provides an enhanced Auto-Complete that shows the possible categories for each suggestion. This makes it easier for users to navigate directly to the most relevant search results. Songza also has an enhanced Auto-Complete that shows groupings.

**NOTE**

Show a loading indicator if there could be a delay in displaying Auto-Complete results. Consider grouping the suggestions to improve search efficiency.

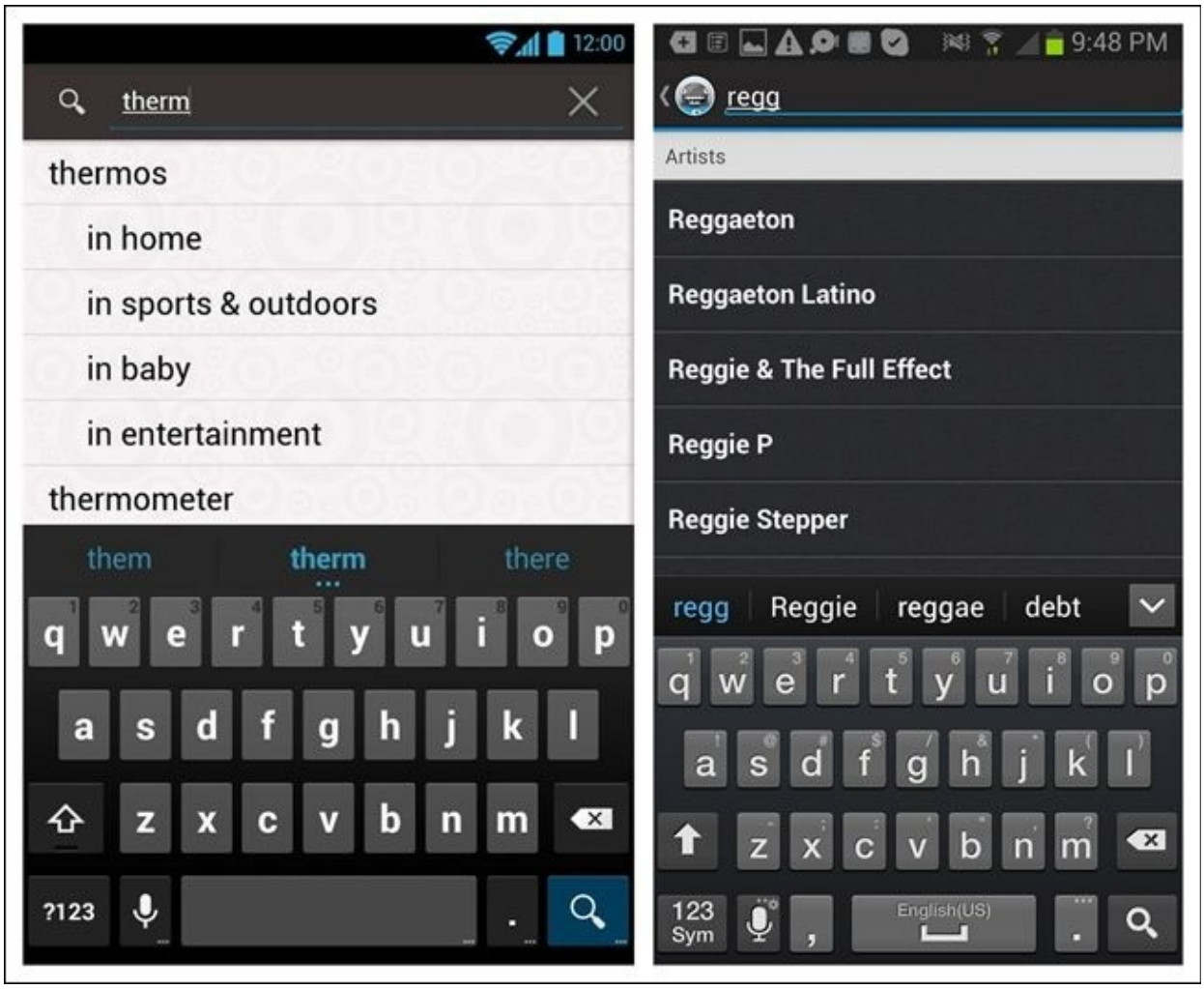



*Figure 4-18. Target and Songza for Android: enhanced Auto-Complete options*

## Dynamic Search

The Dynamic Search pattern may also be considered dynamic filtering. Here, a blank search field returns all possible category results, and entering a search string dynamically filters down the list. In the Audible example, you can see that my entire audiobook library is shown in the background. When I start to type, only the matching titles remain.

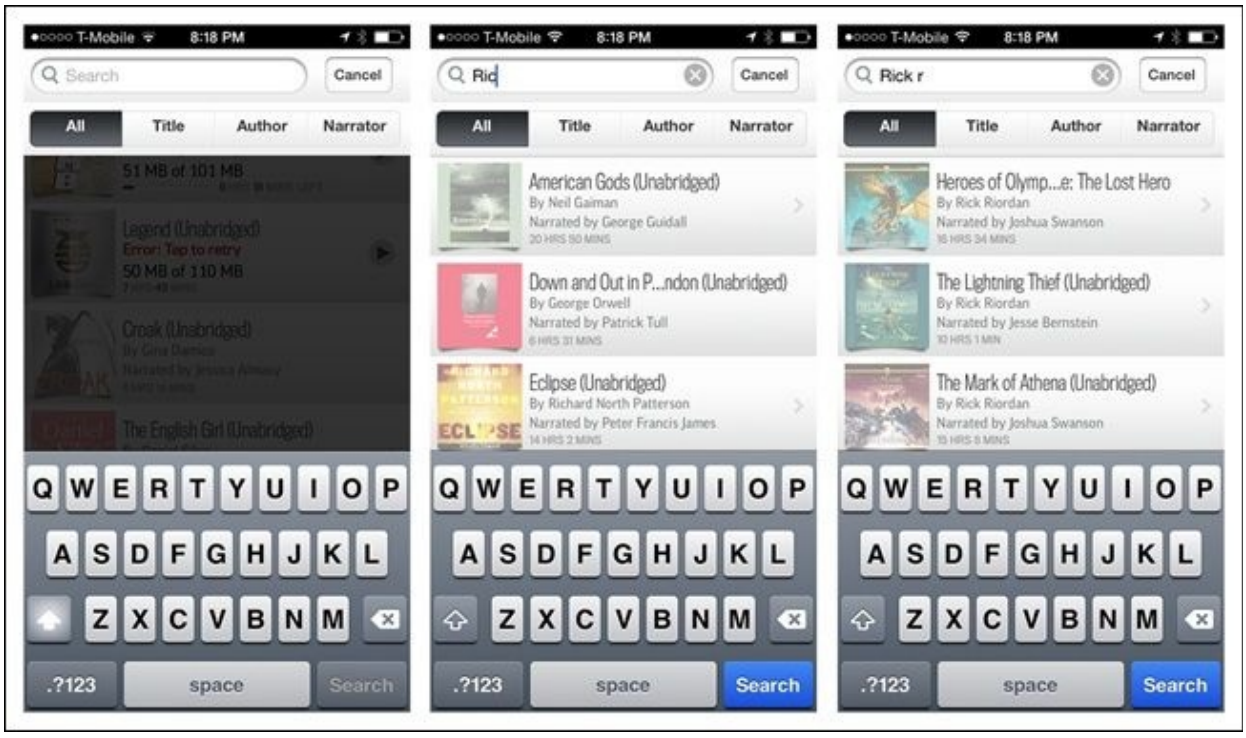



*Figure 4-19. Dynamic Search in Audible for iOS narrows results as users type*

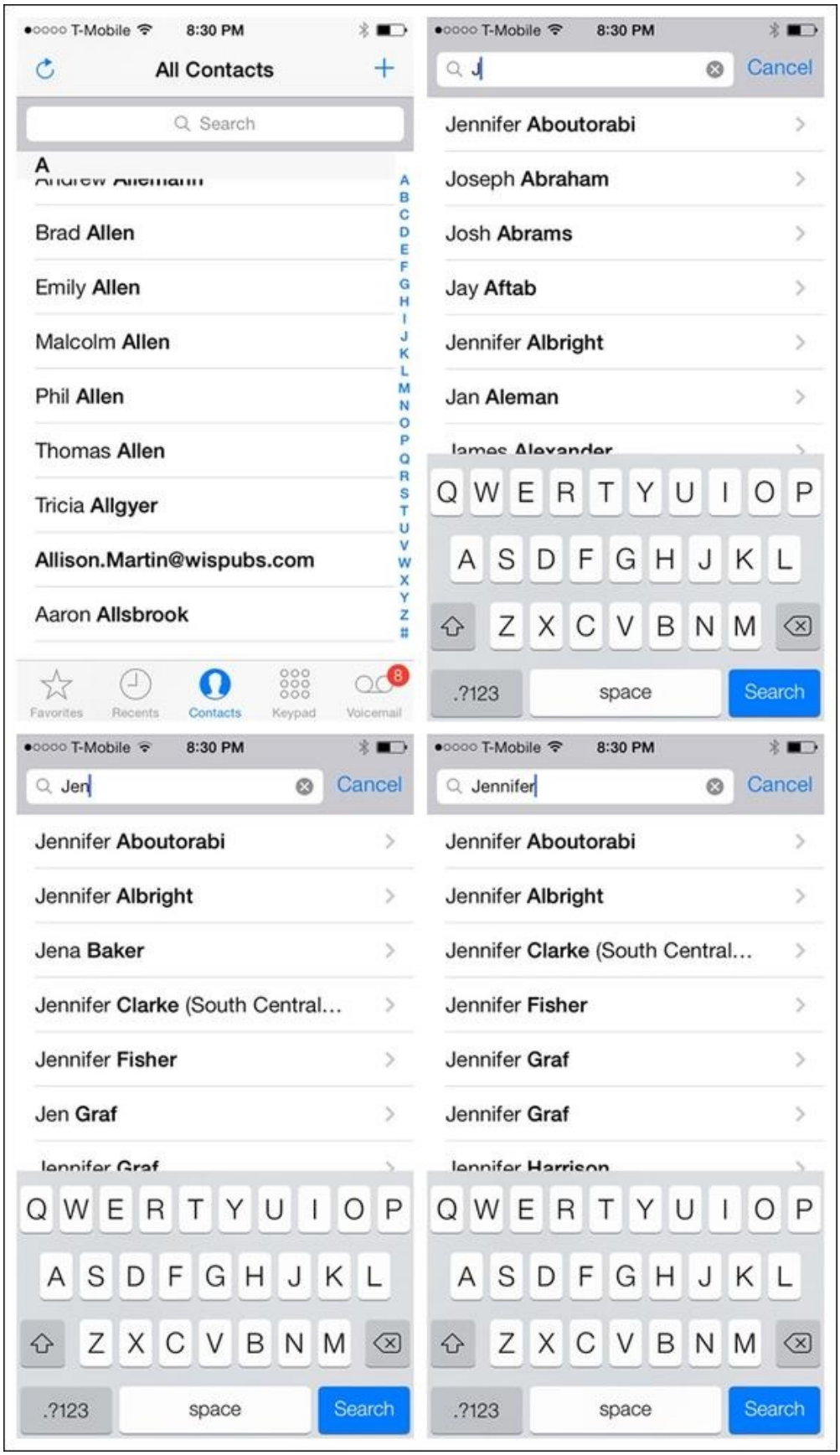

●○○○○ T-Mobile 8:30 PM
All Contacts
Search
A
Brad Allen
Emily Allen
Malcolm Allen
Phil Allen
Thomas Allen
Tricia Allgyer
Allison.Martin@wispubs.com
Aaron Allsbrook
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z #
Favorites
Recents
Contacts
Keypad
Voicemail
8
●○○○○ T-Mobile 8:30 PM
J
Cancel
Jennifer Aboutorabi
Joseph Abraham
Josh Abrams
Jay Aftab
Jennifer Albright
Jan Aleman
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
.?123
space
Search
●○○○○ T-Mobile 8:30 PM
Jen
Cancel
Jennifer Aboutorabi
Jennifer Albright
Jena Baker
Jennifer Clarke (South Central...
Jennifer Fisher
Jen Graf
●○○○○ T-Mobile 8:30 PM
Jennifer
Cancel
Jennifer Aboutorabi
Jennifer Albright
Jennifer Clarke (South Central...
Jennifer Fisher
Jennifer Graf
Jennifer Graf

*Figure 4-20. Dynamic Search in Contacts for iOS quickly winnows results of search*

> **NOTE**
>
> Dynamic Search works well for constrained data sets, like an address book or a personal media library, but it should not be used for searching huge data sets, like a retailer's inventory.

## Scoped Search

Sometimes users can get search results faster and easier if you let them narrow the scope of the search before they actually initiate it. Google on Windows Phone 8 offers a menu for specifying a category, while the Google Play Store for Android uses a Spinner control nestled under the search field.

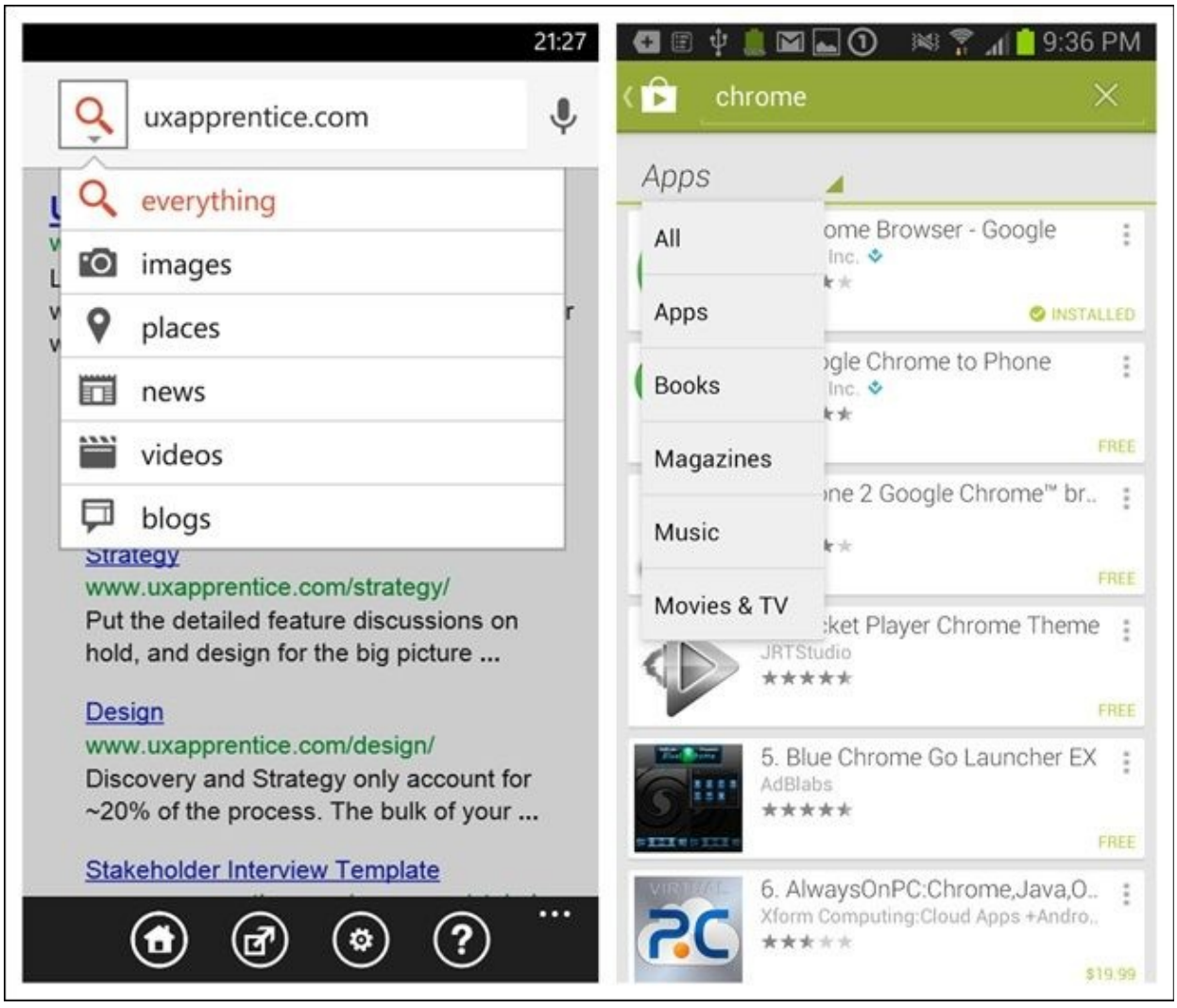



*Figure 4-21. Google for Windows Phone and Google Play Store for Android: menus to scope searches*

It is common to see a segmented control in iOS for Scoped Search, as in iTunes, but there are other UI options, like Scrolling Tabs or a Pill Bar.

> **NOTE**
>
> Offer reasonable scoping options based on the data set. Three to five scoping options is plenty. Consider a Search Form to provide advanced search capabilities.

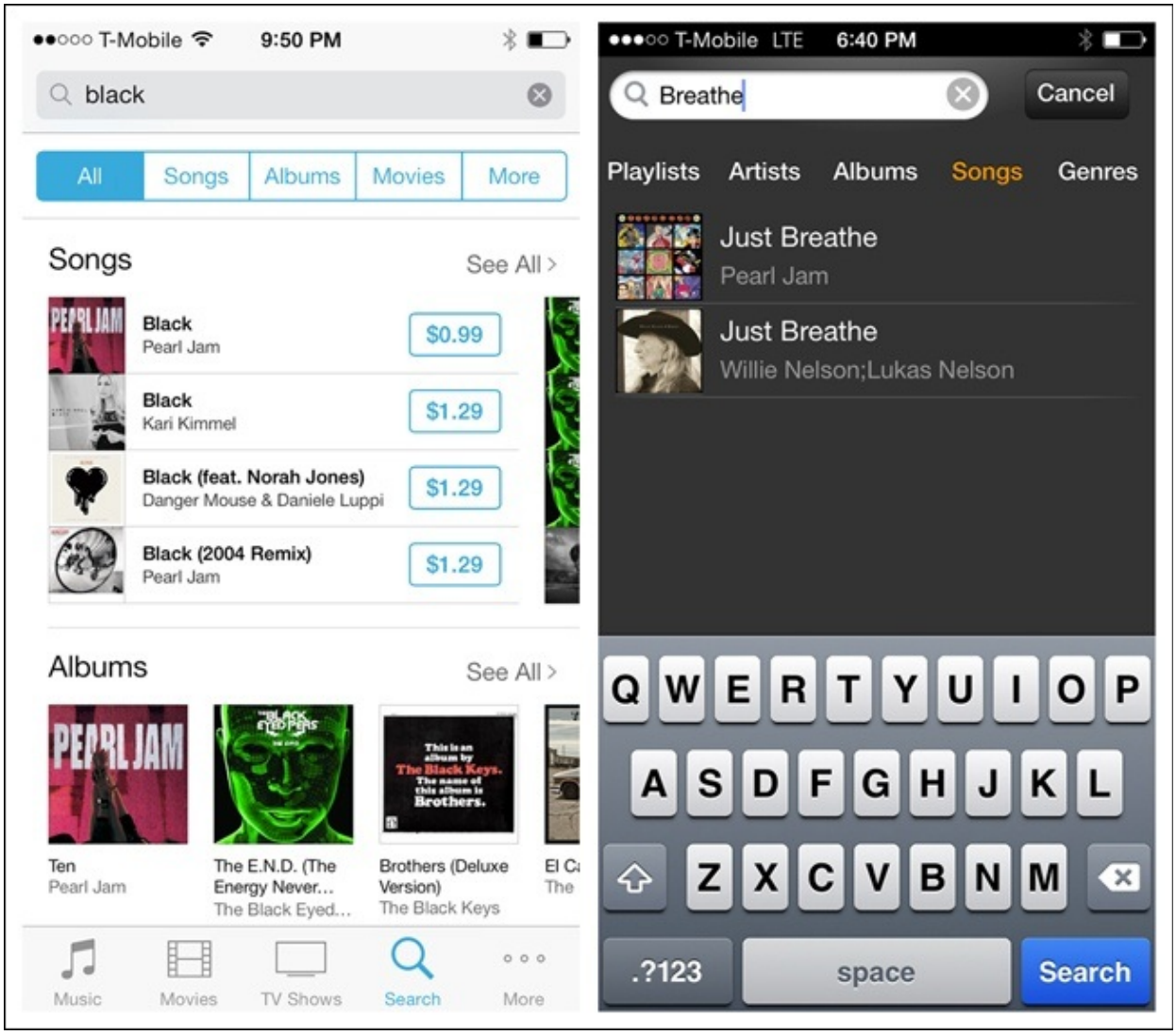



*Figure 4-22. iTunes and Amazon Cloud Player for iOS: two options to narrow search scope*

## Saved, Recent, and Popular Search

Well-designed mobile interfaces follow a basic usability maxim: respect the user's effort. Saved, Recent, and Popular Searches do this by making it easy to select from previous searches instead of retyping the same keywords or search criteria.

**NOTE**

Saved Search typically requires an additional step to name the search for later use, whereas a Recent Search is automatically saved and surfaced. Consider which one will best serve the needs of your users.

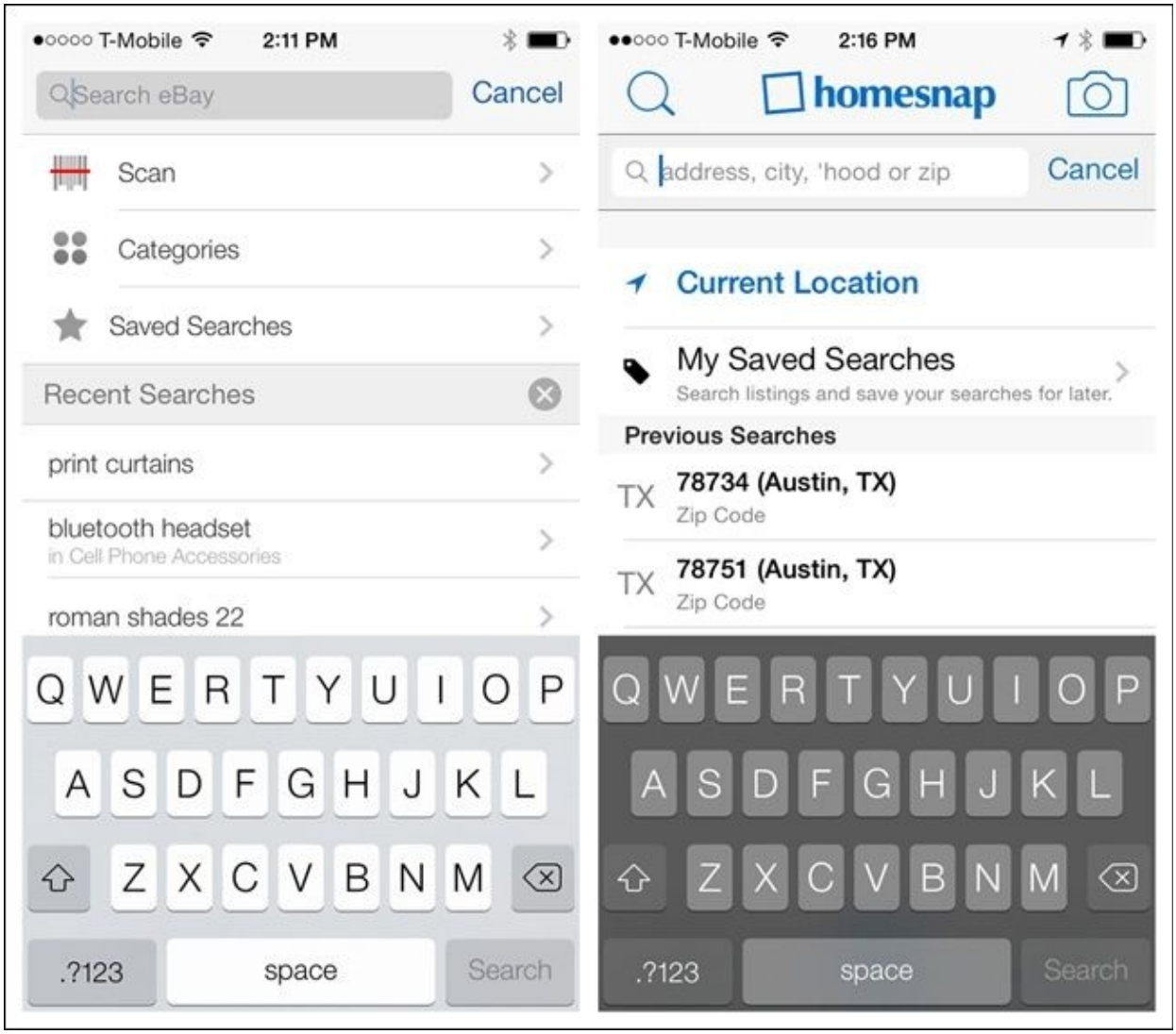



*Figure 4-23. eBay and Homesnap for iOS: Saved and Recent Searches*

Groupon and CheckPay surface users' most popular searches and most popular search options, respectively.

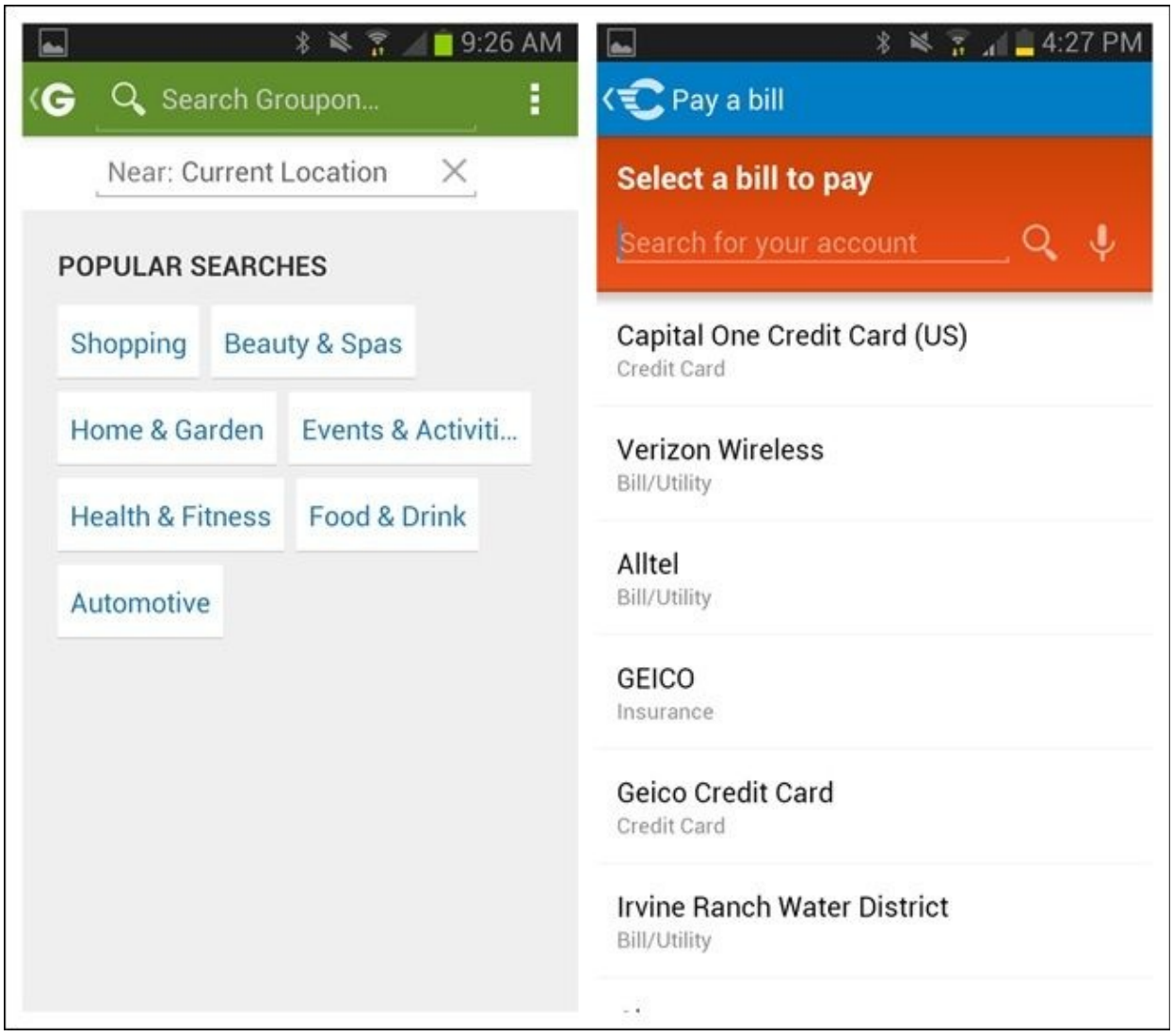



*Figure 4-24. Groupon and CheckPay for Android: surfacing popular searches*

Other techniques to respect the user's effort include location-based searching, as in Airbnb and Foursquare.

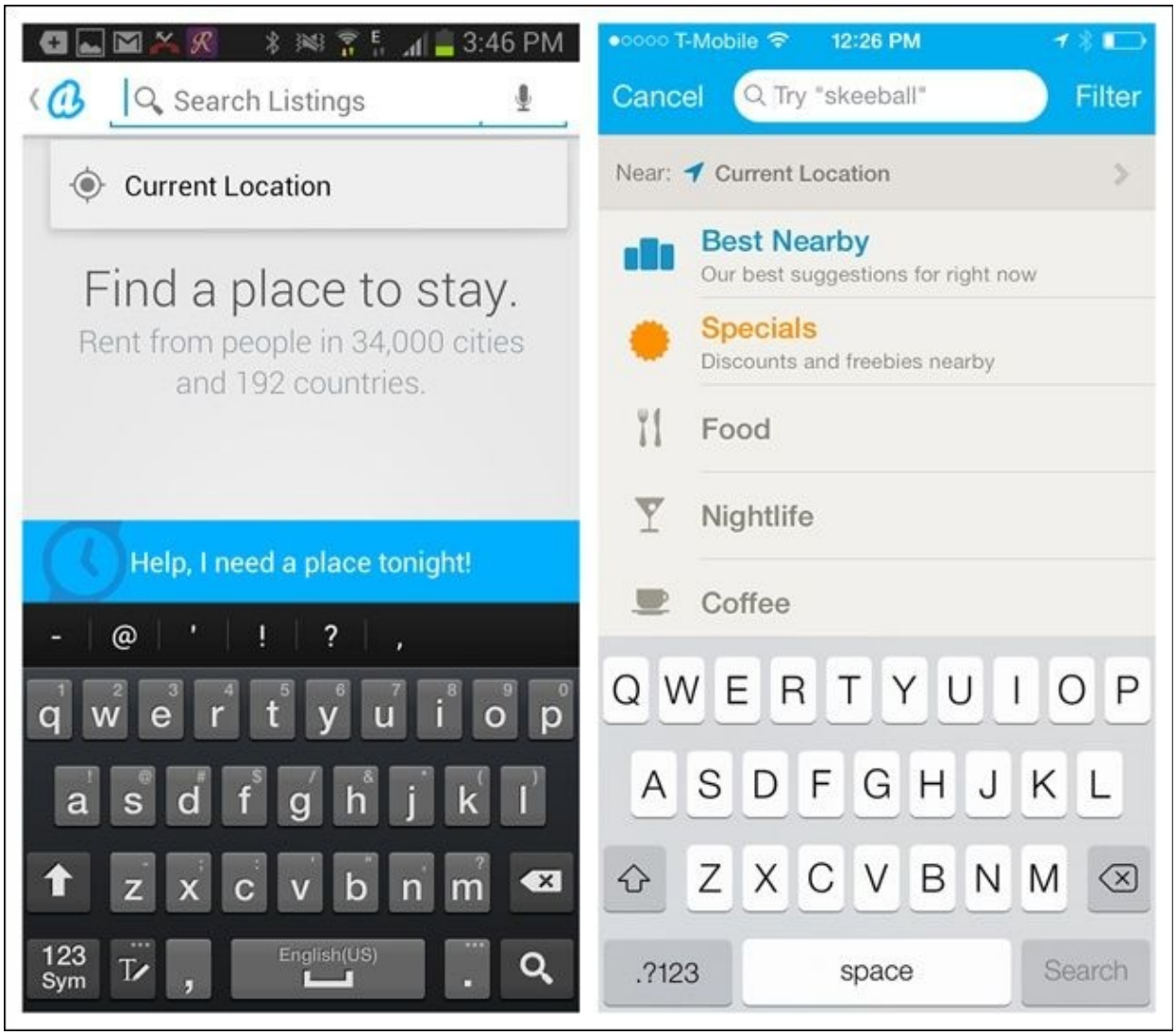



*Figure 4-25. Airbnb for Android and Foursquare for iOS: location-based searching*

## Search Form

This pattern is characterized by a separate form for entering multiple criteria, along with an explicit search button. Hipmunk and Skyscanner use Search Forms to set the necessary criteria for searching flights and hotels. See more examples in Chapter 2.

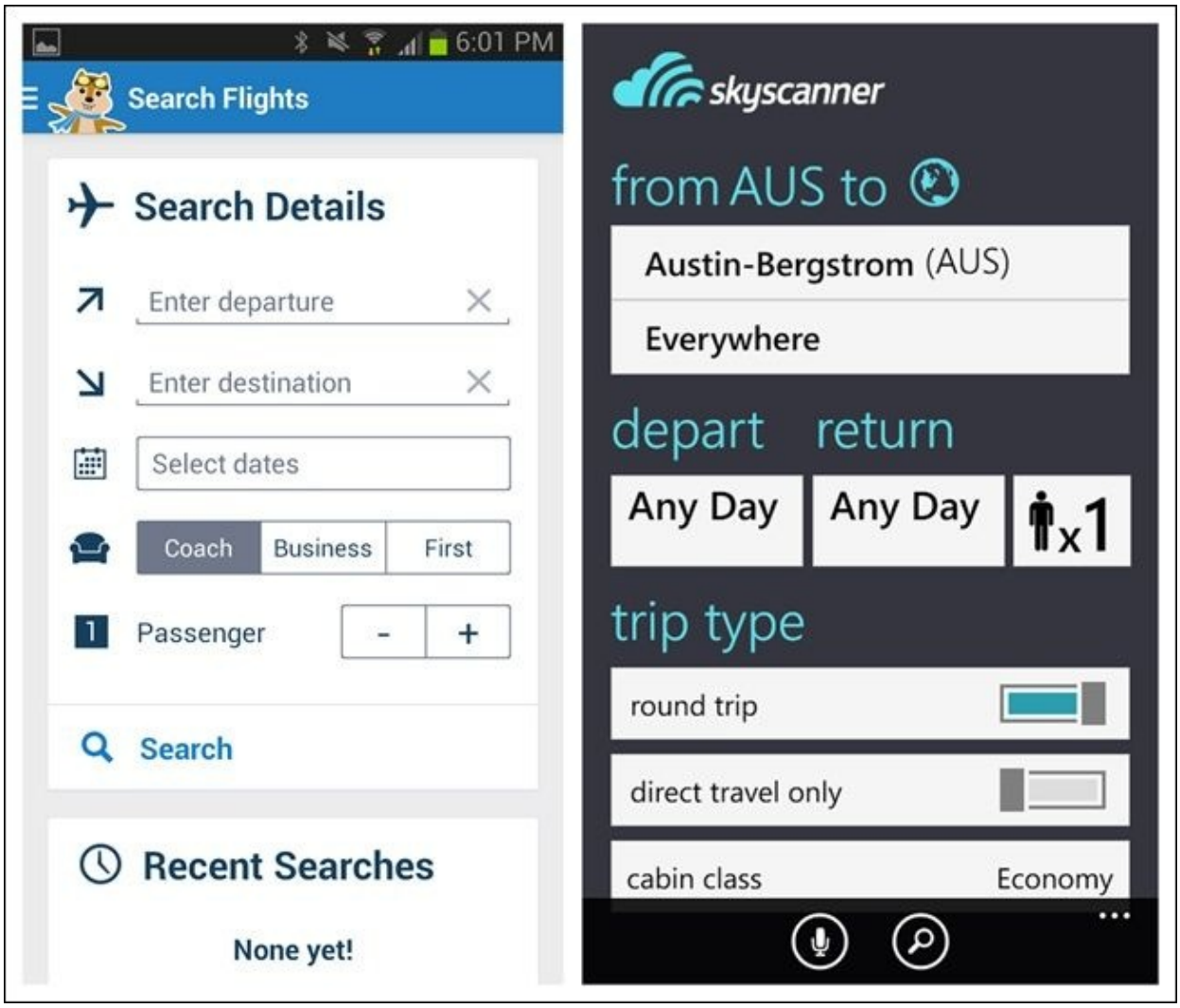



*Figure 4-26. Hipmunk for Android and Skyscanner for Windows Phone: Search Form examples*

Expedia for both iOS and Android uses an elegant design that animates the origin and destination fields. Both versions also include only the most essential search fields.

**NOTE**

Minimize the number of input fields. Use OS-specific input controls properly. Follow form design best practices for alignment, labels, size, and so on.

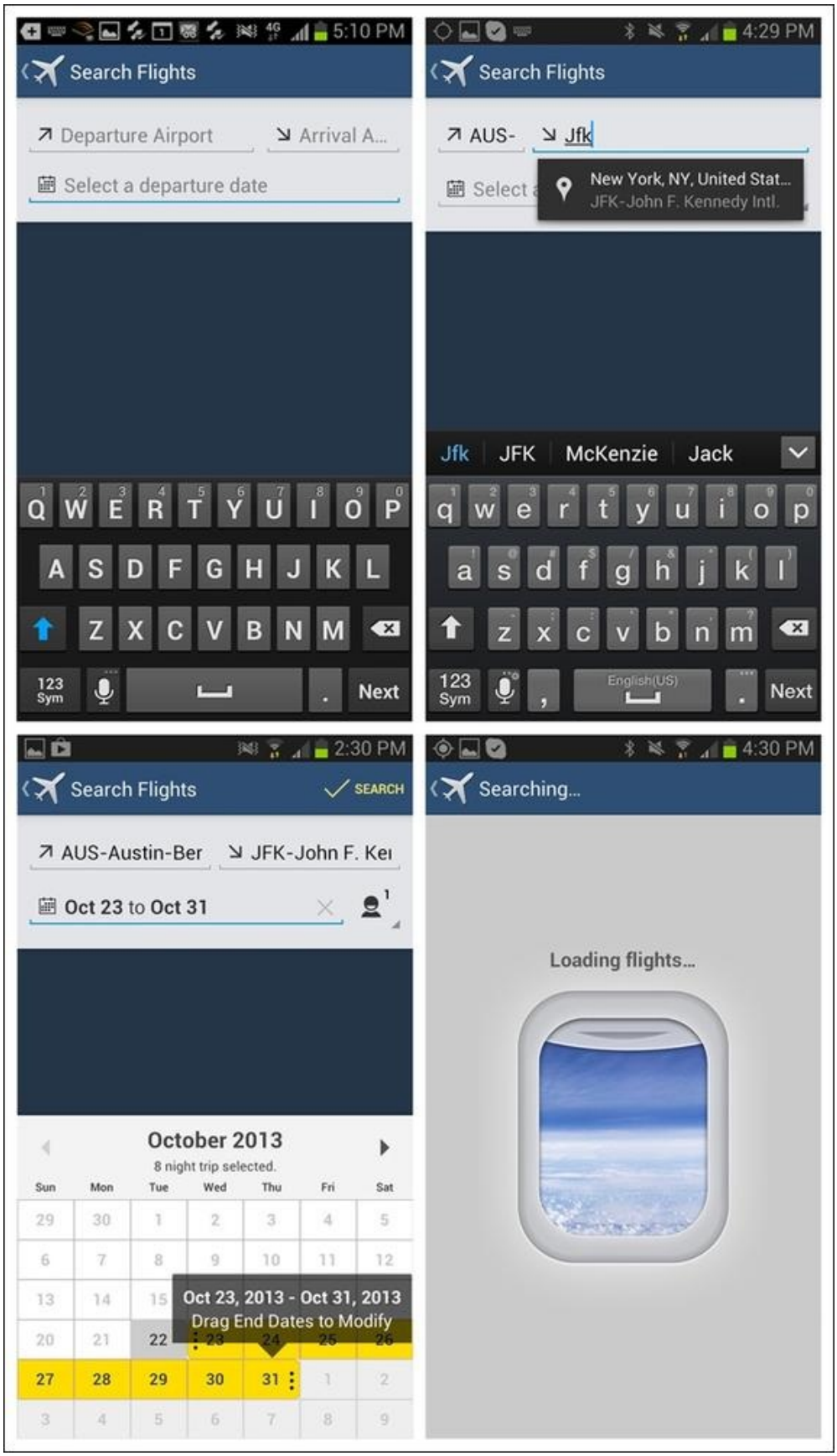

5:10 PM
Search Flights
Departure Airport
Arrival A...
Select a departure date
4:29 PM
Search Flights
AUS-
Jfk
New York, NY, United Stat...
JFK-John F. Kennedy Intl.
Jfk
JFK
McKenzie
Jack
Next
2:30 PM
Search Flights
SEARCH
AUS-Austin-Ber
JFK-John F. Kei
Oct 23 to Oct 31
October 2013
8 night trip selected.
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Oct 23, 2013 - Oct 31, 2013
Drag End Dates to Modify
4:30 PM
Searching...
Loading flights...

*Figure 4-27. Expedia for Android: a helpful, elegant Search Form design*

## Search Results/View Results

Once a search is performed, the results can be displayed on the same screen or on a dedicated results screen. Results may be displayed in a table or list, on a map or satellite image, or as thumbnails.

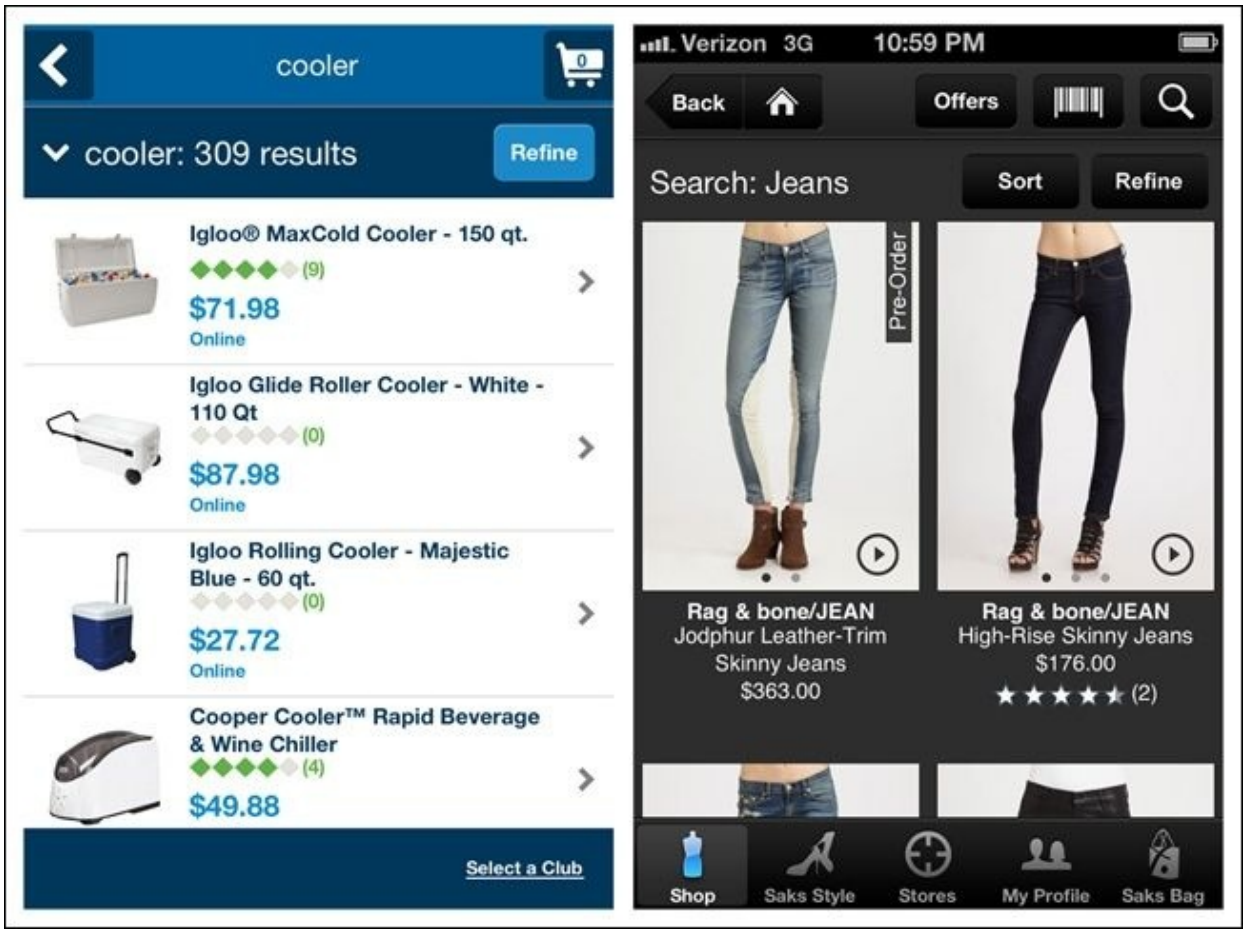



*Figure 4-28. Sam's Club and Saks for iOS: list view and tile view Search Results, respectively*

Multiple View Results options are available depending on the type of results and user preferences.

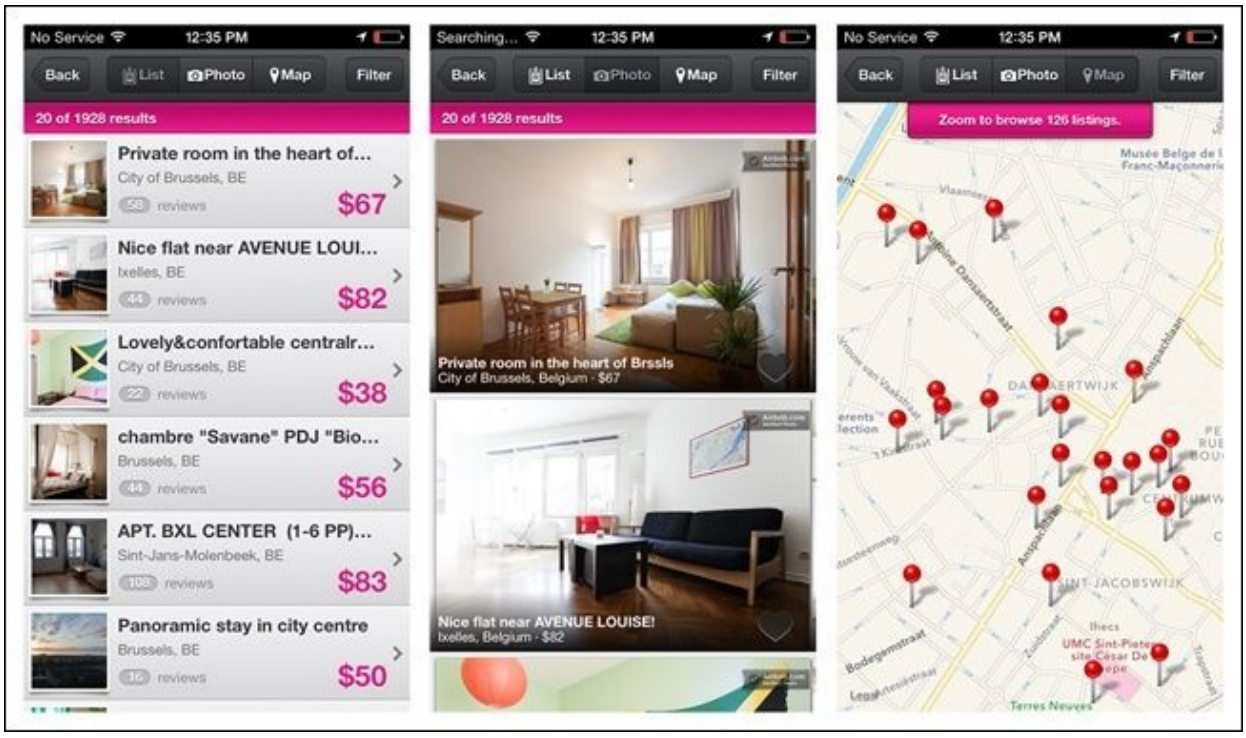



*Figure 4-29. Airbnb for iOS: list, photo, and map Search Result views*

Another emerging trend for displaying Search Results is to show each result on a card, slide, or tile. Users can navigate these by swiping through them.

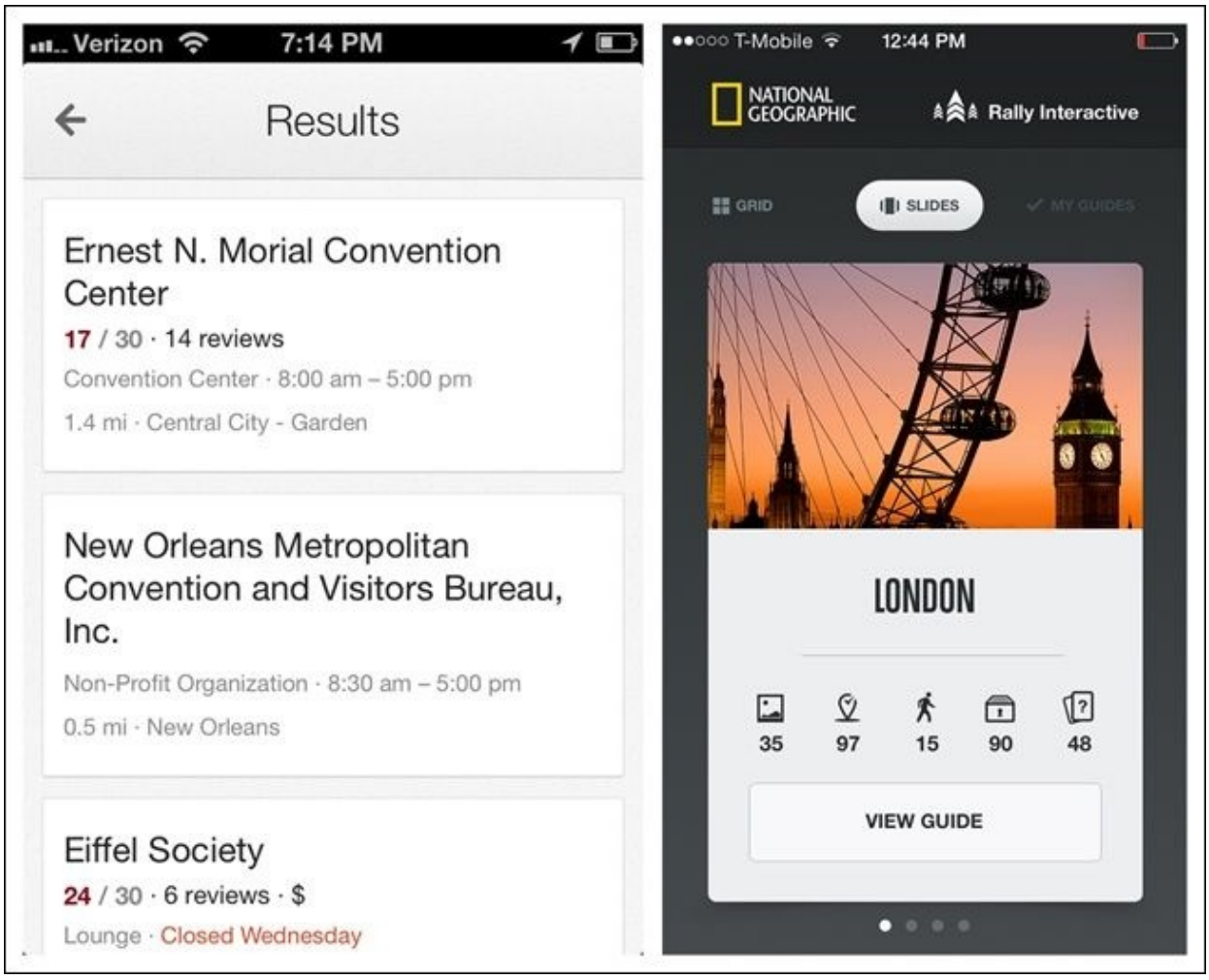



*Figure 4-30. Google Now's Search Results on cards; National Geographic's on slides*

Hipmunk has pioneered visualizing flight results with a Gantt chart–style timeline. Expedia's take on it helps visualize the flight duration in context with the ticket cost and departure time.

Hipmunk also offers an alternative way to view hotel search results. The results are shown on a map — pretty standard, right? But there's a heat map overlay option for viewing the hotels with respect to area crime, shopping, and nightlife.

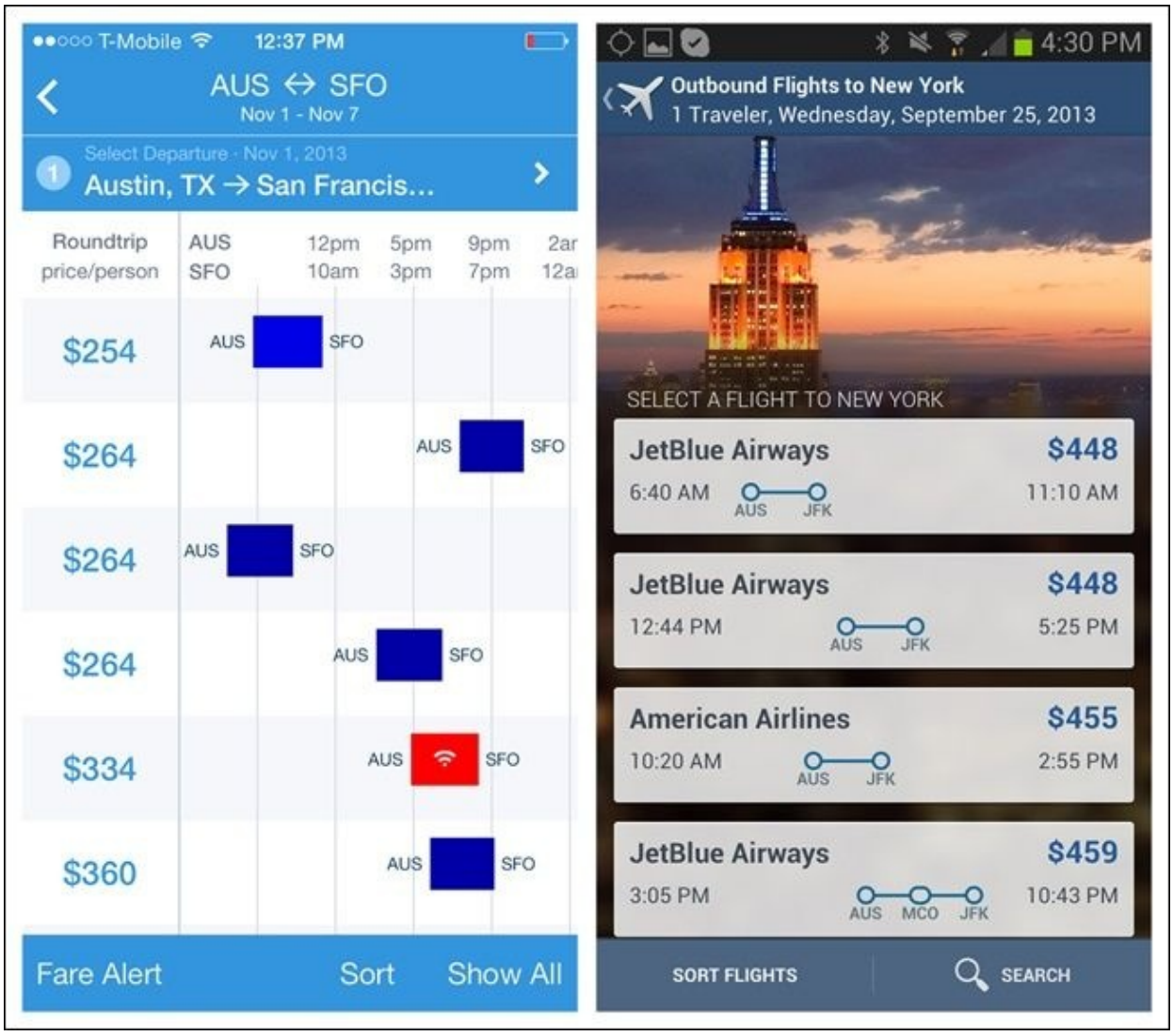



*Figure 4-31. Hipmunk for iOS and Expedia for Android*

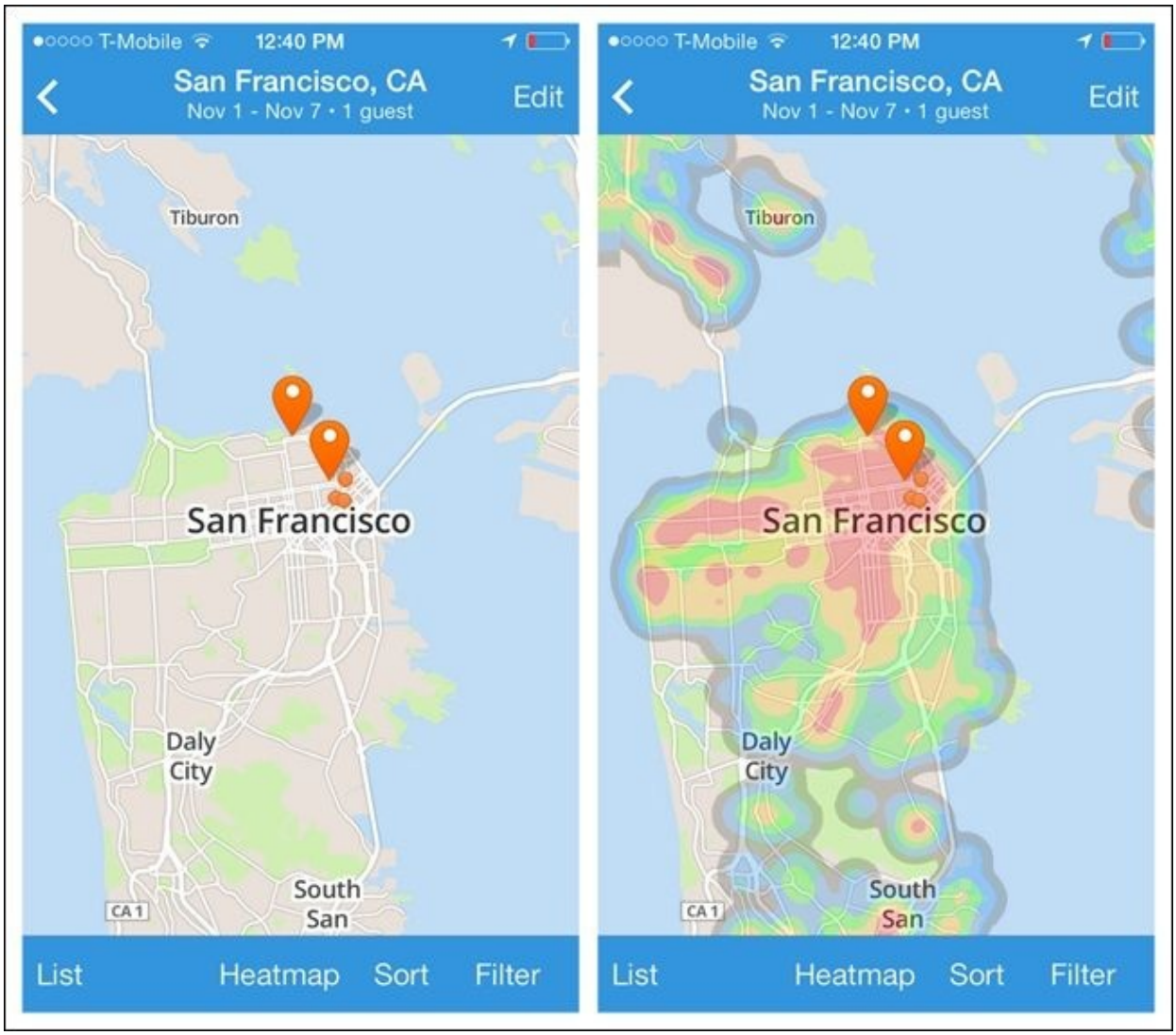



*Figure 4-32. Hipmunk for iOS: showing a heat map overlay for various statistics*

*Lazy loading* is a common technique for displaying partial results while the rest are being loaded. Many applications, like Allthecooks, automatically load more results when the screen is swiped. Alternatively, the Apple Store provides a link to load more items.

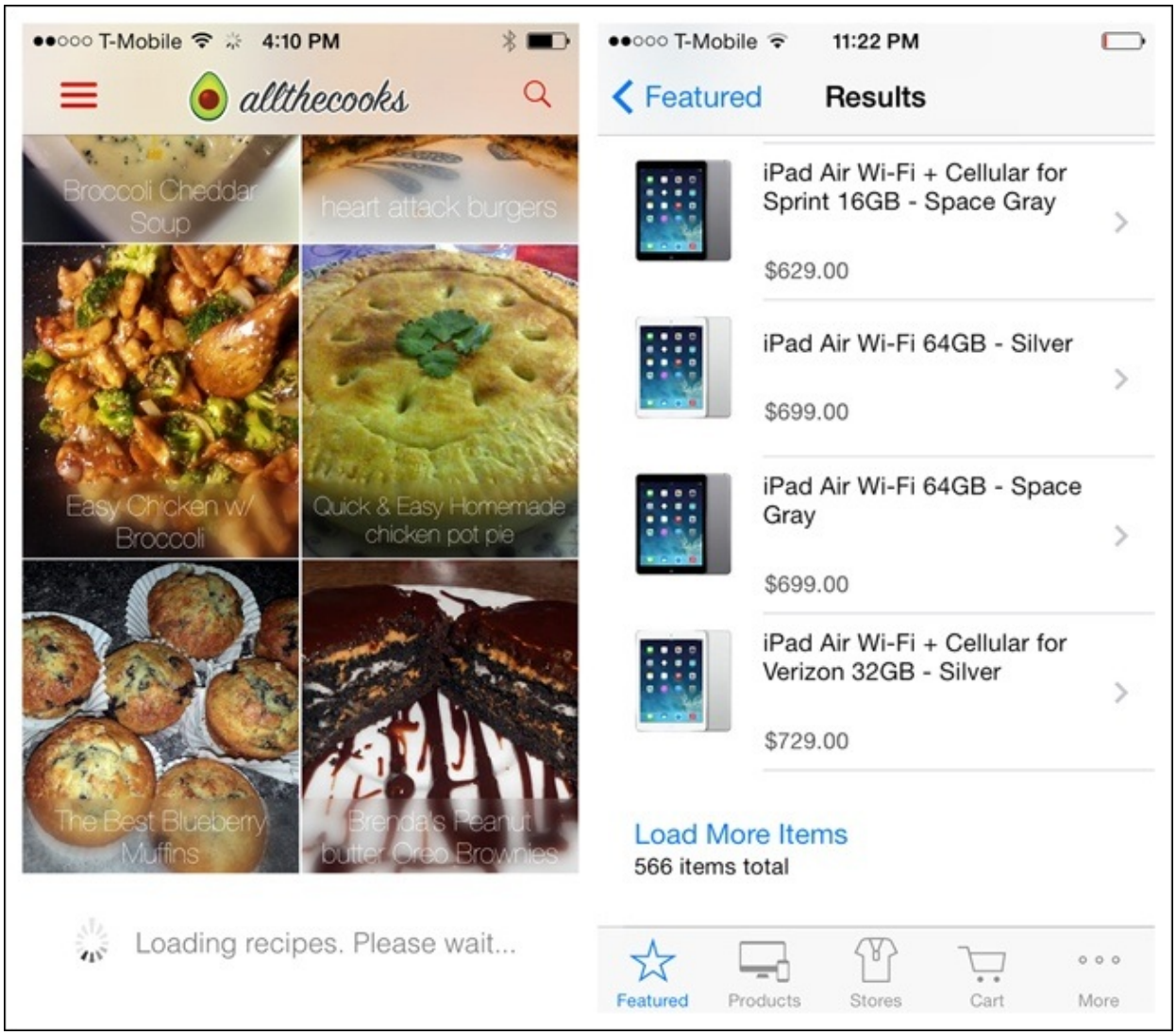



*Figure 4-33. Allthecooks and Apple Store for iOS: lazy loading of results*

Avoid paged tables — they break the natural interaction model of viewing information on a mobile device.

> **NOTE**
>
> Label the results with the number of items returned. Use lazy loading instead of paging. Apply a reasonable default sort order.

## Sort Patterns

Apple earns praise for some of its user-centric designs, but I'd argue the App Store search function shouldn't be one of them. Search "finance," for example, and you'll get 2,200+ results. As you start flicking through hundreds of apps, you might wonder, are these in any kind of order, like most popular, highest rated, or recently released? There's no way to tell. Odds are good you'll need to go elsewhere to research which financial apps will work best for you, and then search for those specific app names.


*Figure 4-34. App Store for iOS: 2,200 unsorted results for finance apps*

Likewise, having the ability to manipulate search results in Google Maps would be a huge help. I assumed the default result for "gas," say, would be the gas station closest to my current location, but I found that typically it isn't.

Instead, I'm forced to open the results list to check the distances of each result. Of course, even this is less than perfect, because distance and drive time are not the same thing. Having the ability to arrange the list by highest rated, distance, or drive time would be super useful. Alas, no such luck.

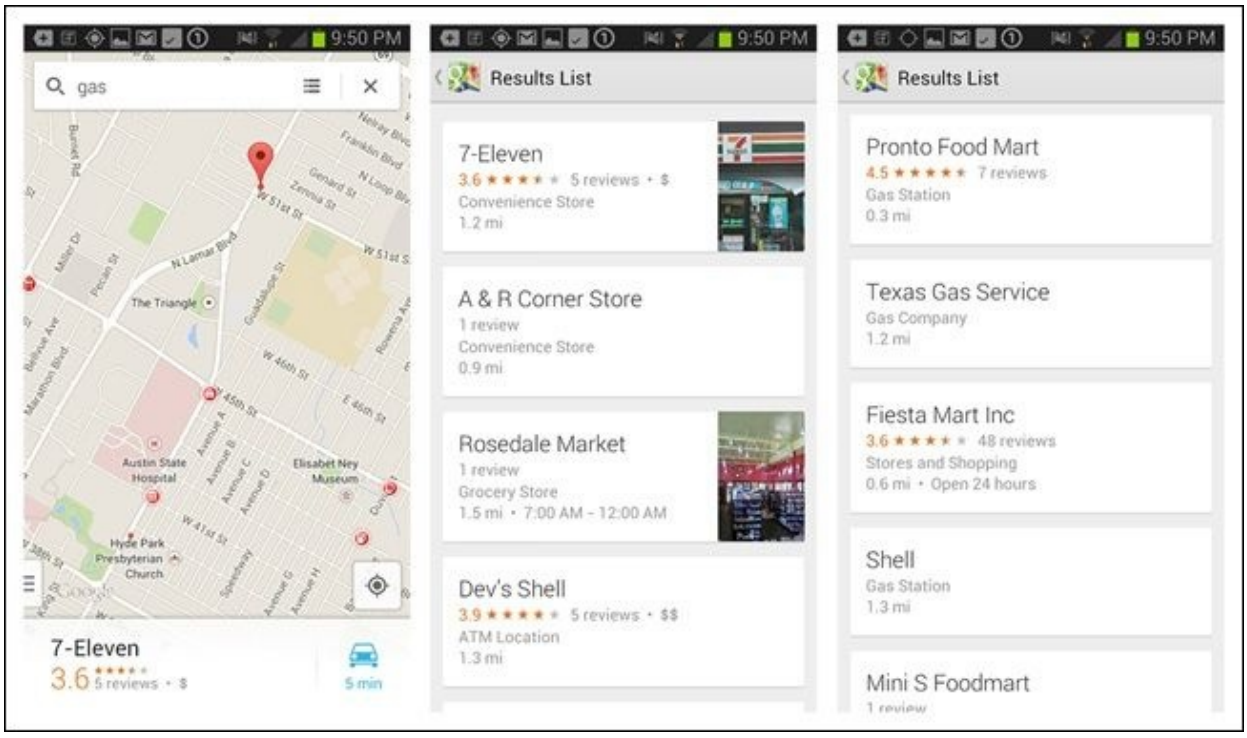



*Figure 4-35. Google Maps for Android: the closest and highest-rated result is hidden on page 2 (Pronto Food Mart)*

Although in these instances Apple and Google seem to have forgotten the value of it, the sort pattern can greatly enhance search usability. Consider integrating one of the following patterns with your search results:

- Onscreen Sort
- Sort Overlay
- Sort Form

## Onscreen Sort

When you need only a few sort options, an Onscreen Sort can provide a simple one-tap solution. But as we'll see, many iOS apps have moved away from this pattern in favor of a more space-saving option, like the Sort Overlay.

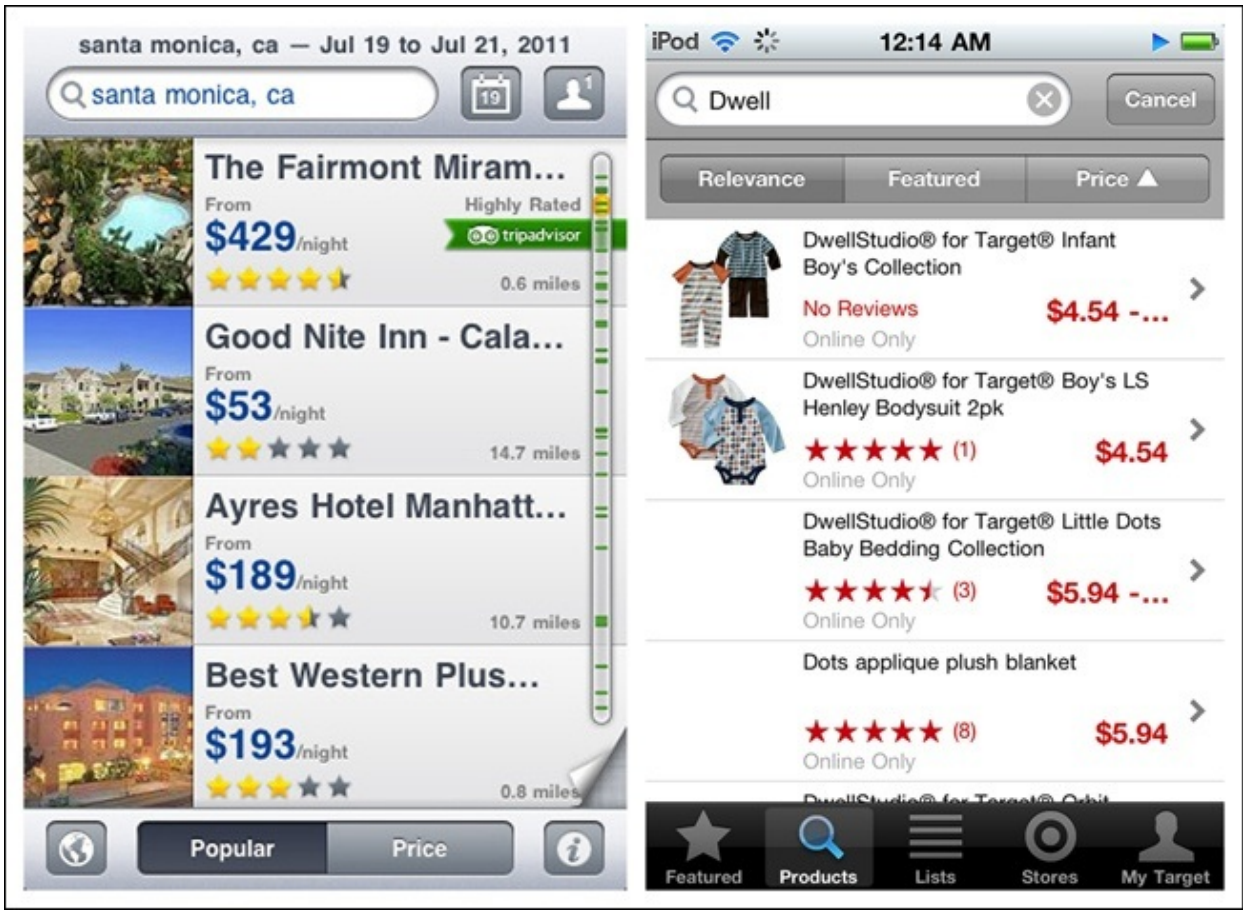



*Figure 4-36. Onscreen Sort examples from the first edition of this book: Expedia and Target for iOS*

The email app Boxer reveals Onscreen Sort options only after the sort icon has been tapped, and keeps them visible until the user closes them.

In Android apps, the Spinner control can work nicely for selecting the sort order. Though it doesn't show all sort options at a glance, it clearly shows which option is active.

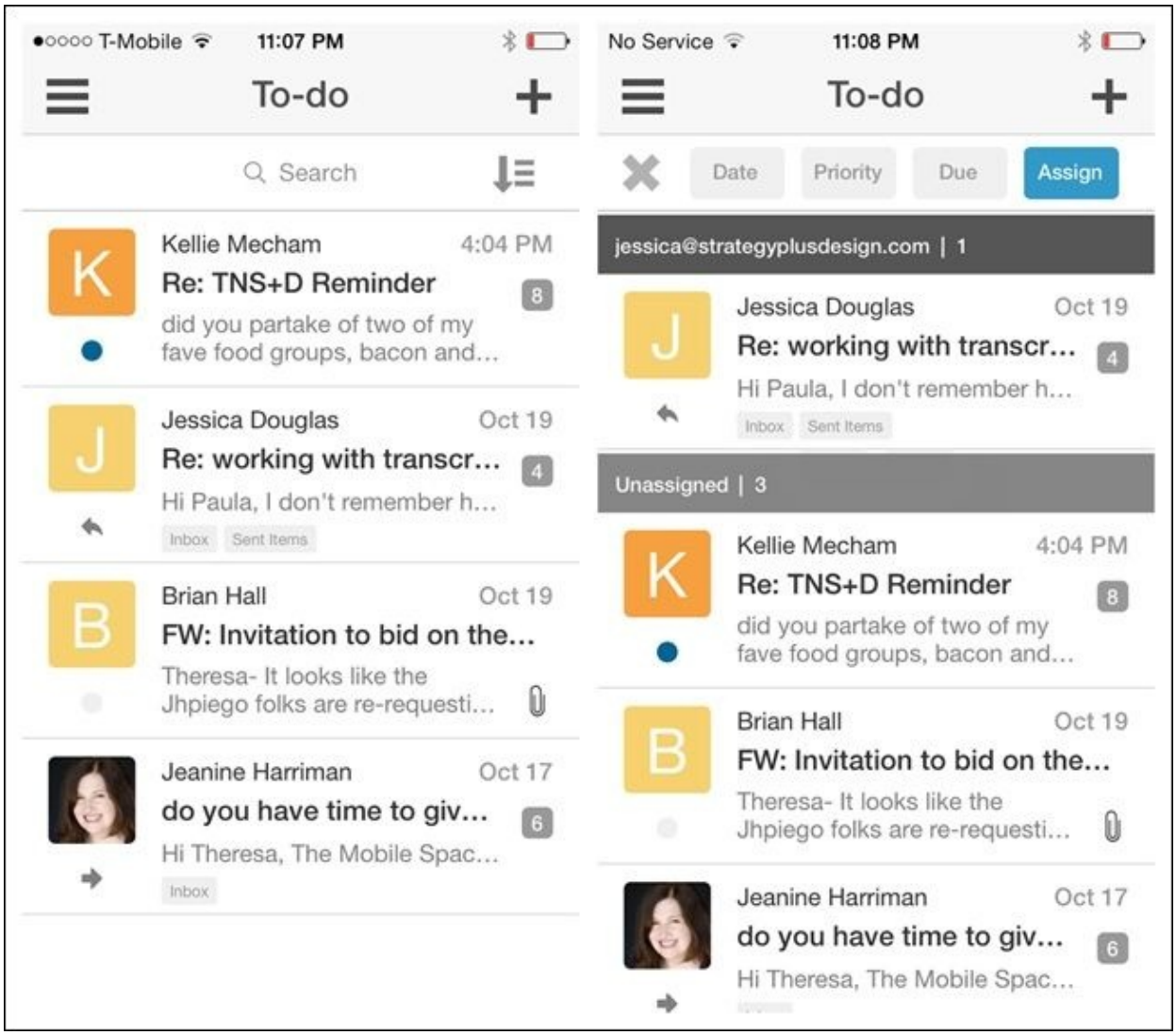



*Figure 4-37. Boxer for iOS*

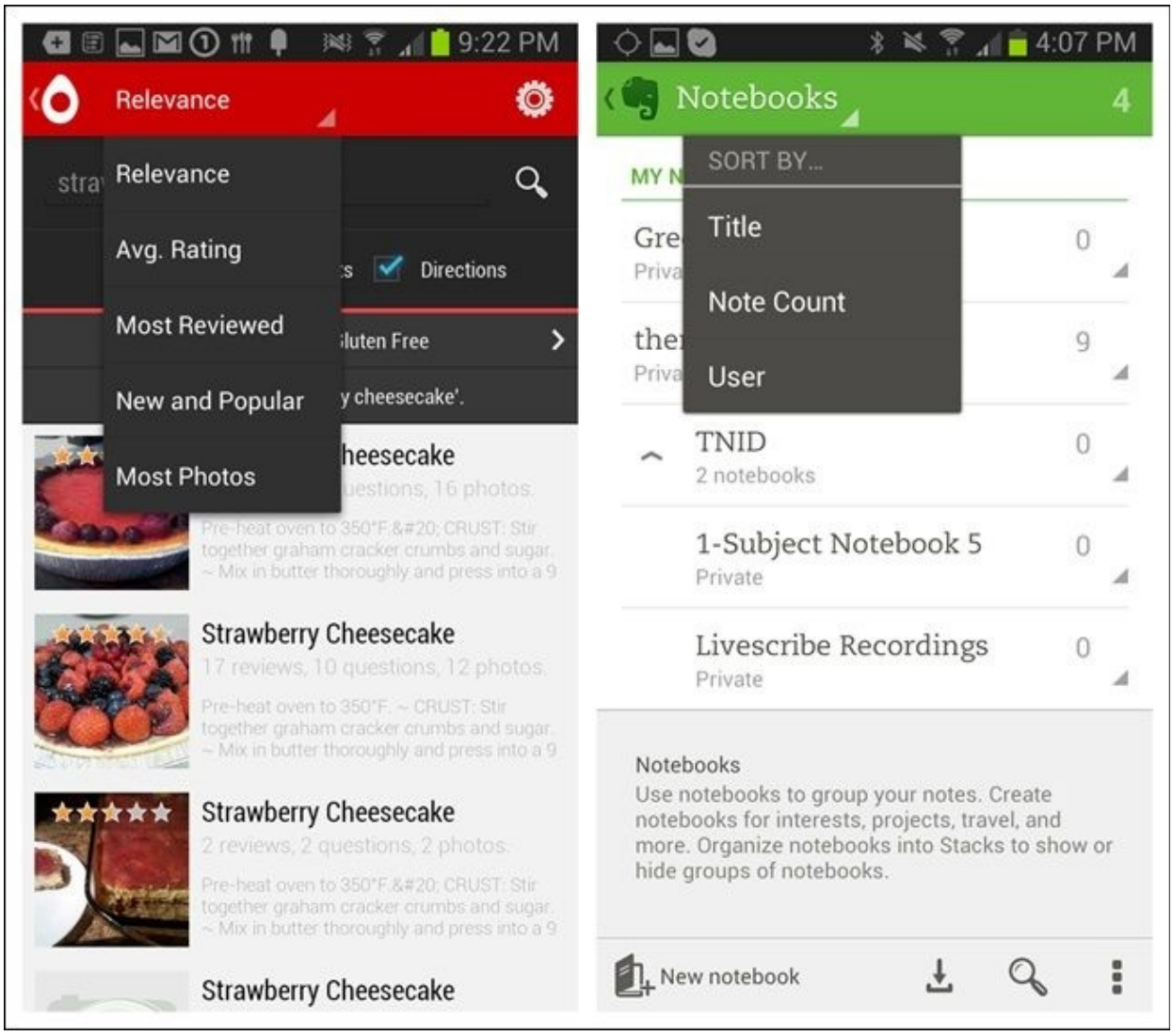



*Figure 4-38. Allthecooks and Evernote for Android: Spinner control for sort options*

## Sort Overlay

The Sort Overlay pattern is a good alternative to Onscreen Sort. It doesn't take up valuable screen real estate or force a truncated label when a longer, more explicit one would work better. The user can initiate the overlay by tapping a sort label or icon, and it can be placed at the top, middle, or bottom of the results screen. Audible's design is one of the most helpful since it shows which sort option is active, even when the overlay is closed.

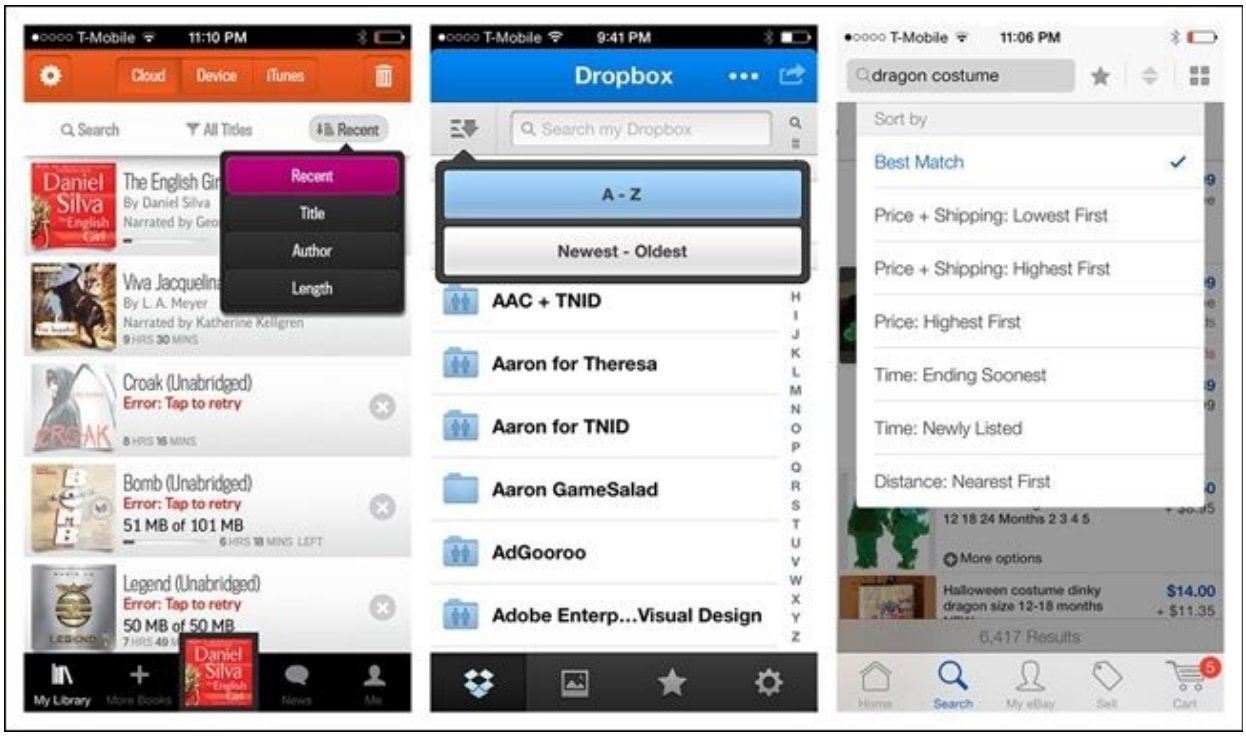



*Figure 4-39. Audible, Dropbox, and eBay for iOS: top-sort overlays*

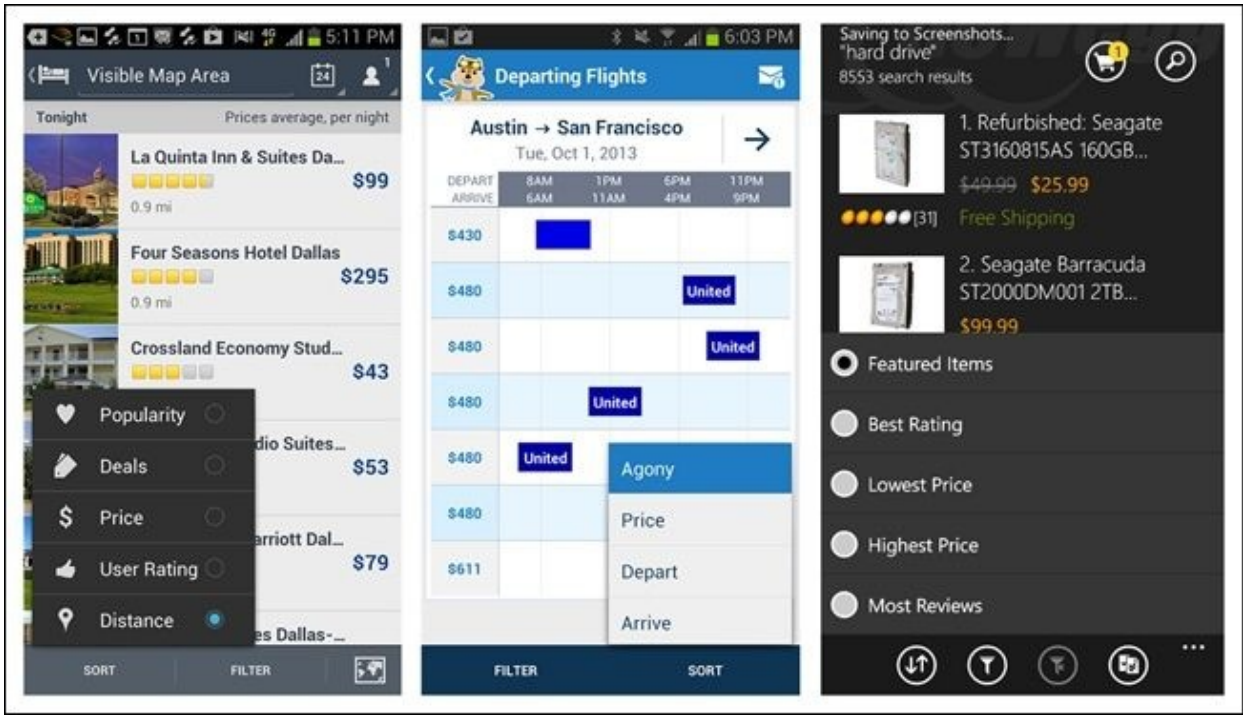



*Figure 4-40. Expedia and Hipmunk for Android, and Newegg for Windows Phone: bottom-sort overlays*

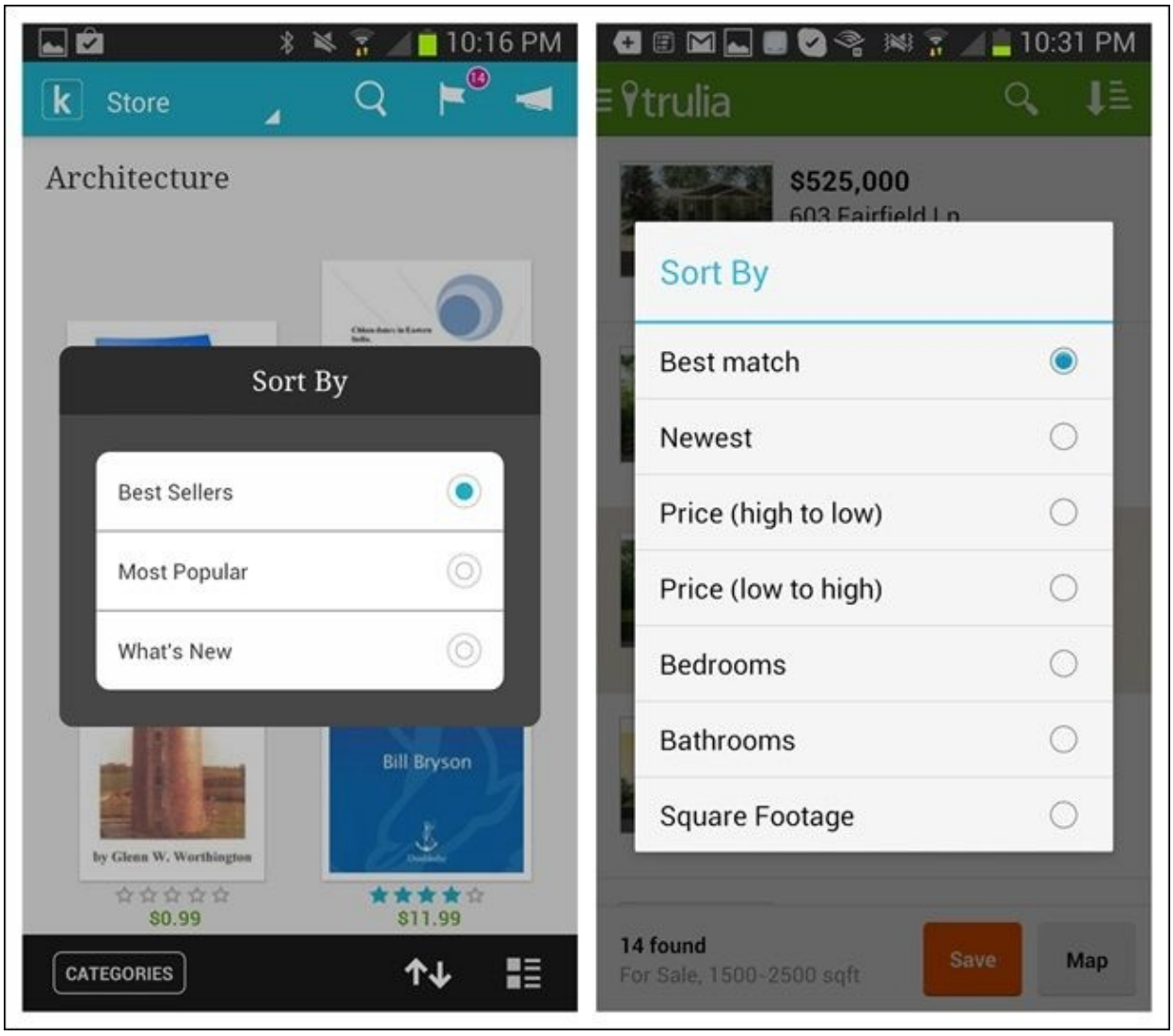



*Figure 4-41. Kobo and Trulia for Android: Sort Overlay dialogs*

Wunderlist offers contextual tools, including a sort option, accessible from the bottom-right corner.

**NOTE**

Sort options should be located near other tools for manipulating the results list, like filters, view toggles, and the “clear search” button. Clearly show which sort option is active.

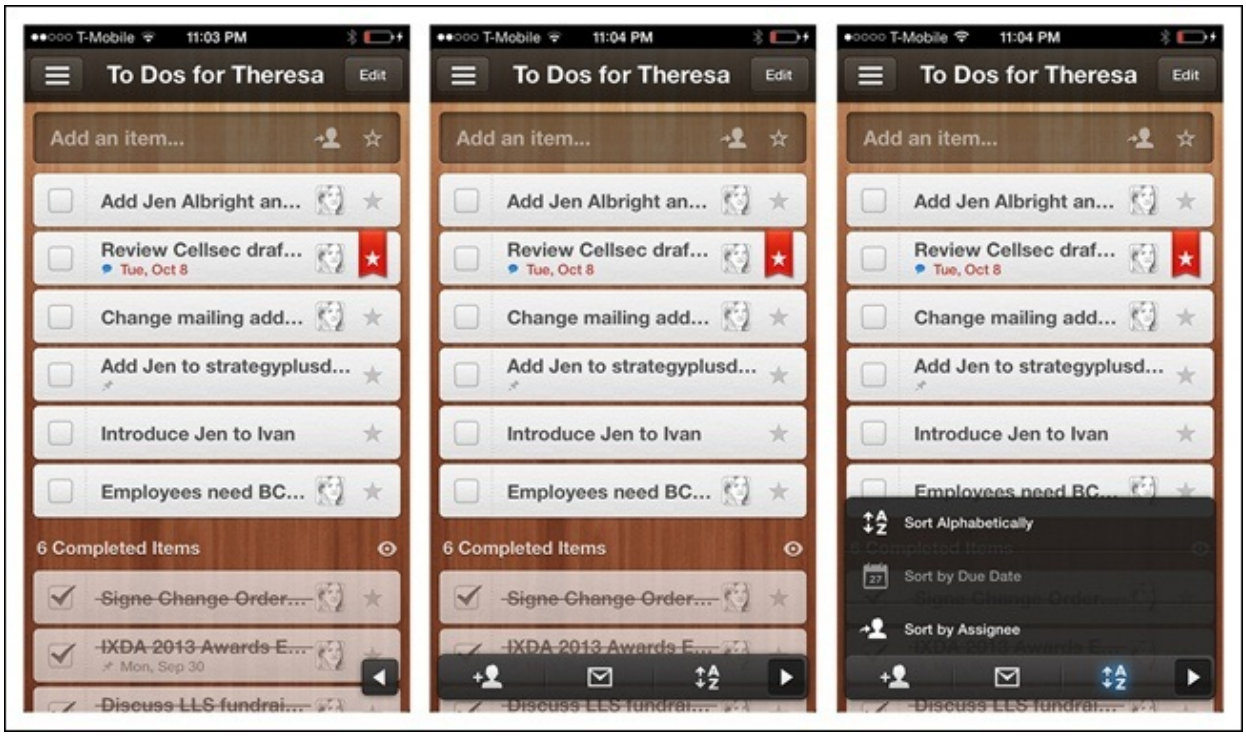



*Figure 4-42. Wunderlist for iOS*

## Sort Form

The Sort Form may require more effort from the user, who must open the form, select an option, and potentially tap “done,” “apply,” or “back.”

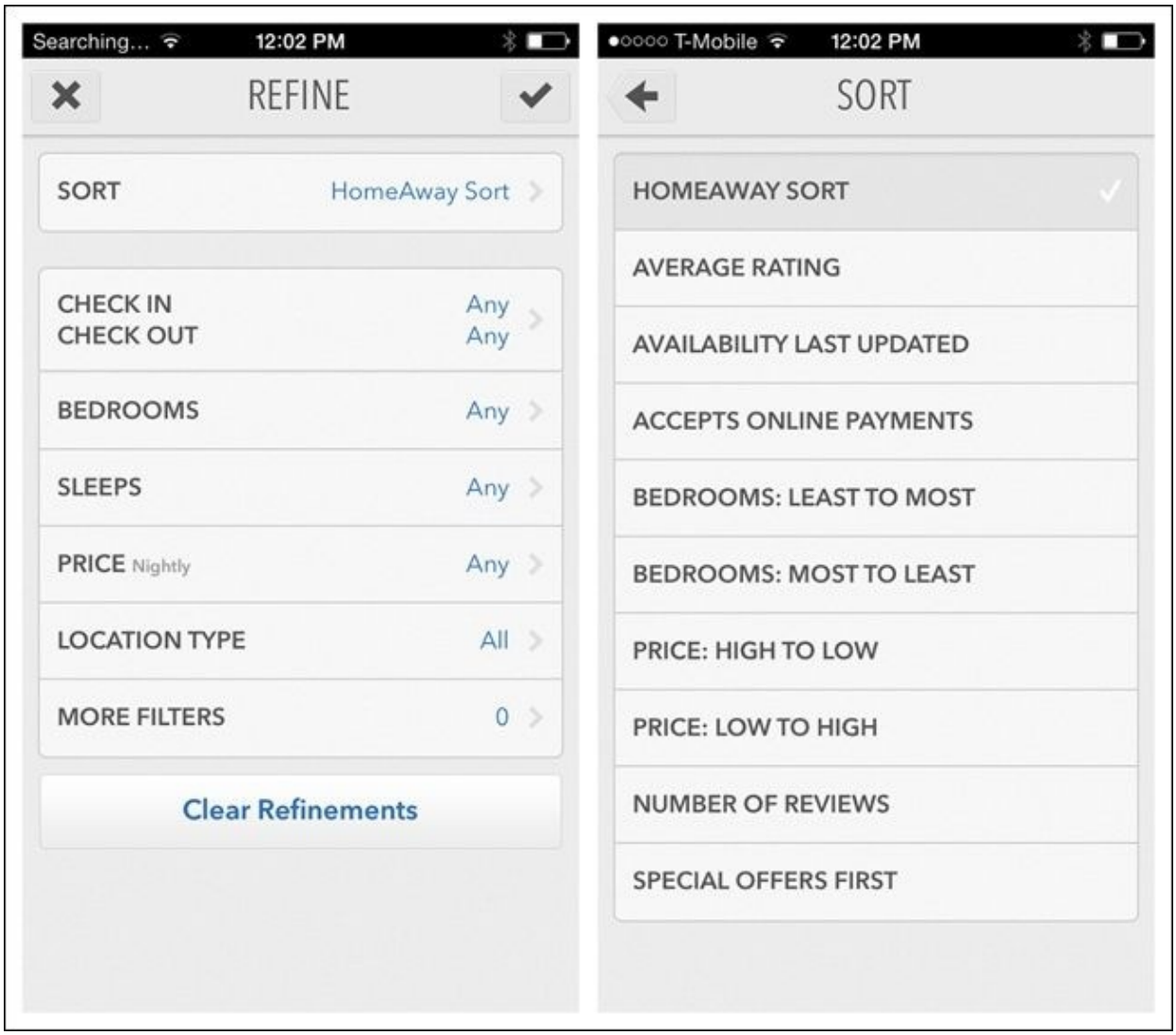



*Figure 4-43. HomeAway for iOS*

Hipmunk uses the Sort Overlay pattern for Android, but chose a Sort Form for iOS. The sort selection is applied as soon as it is tapped — no “apply” or “done” button required.

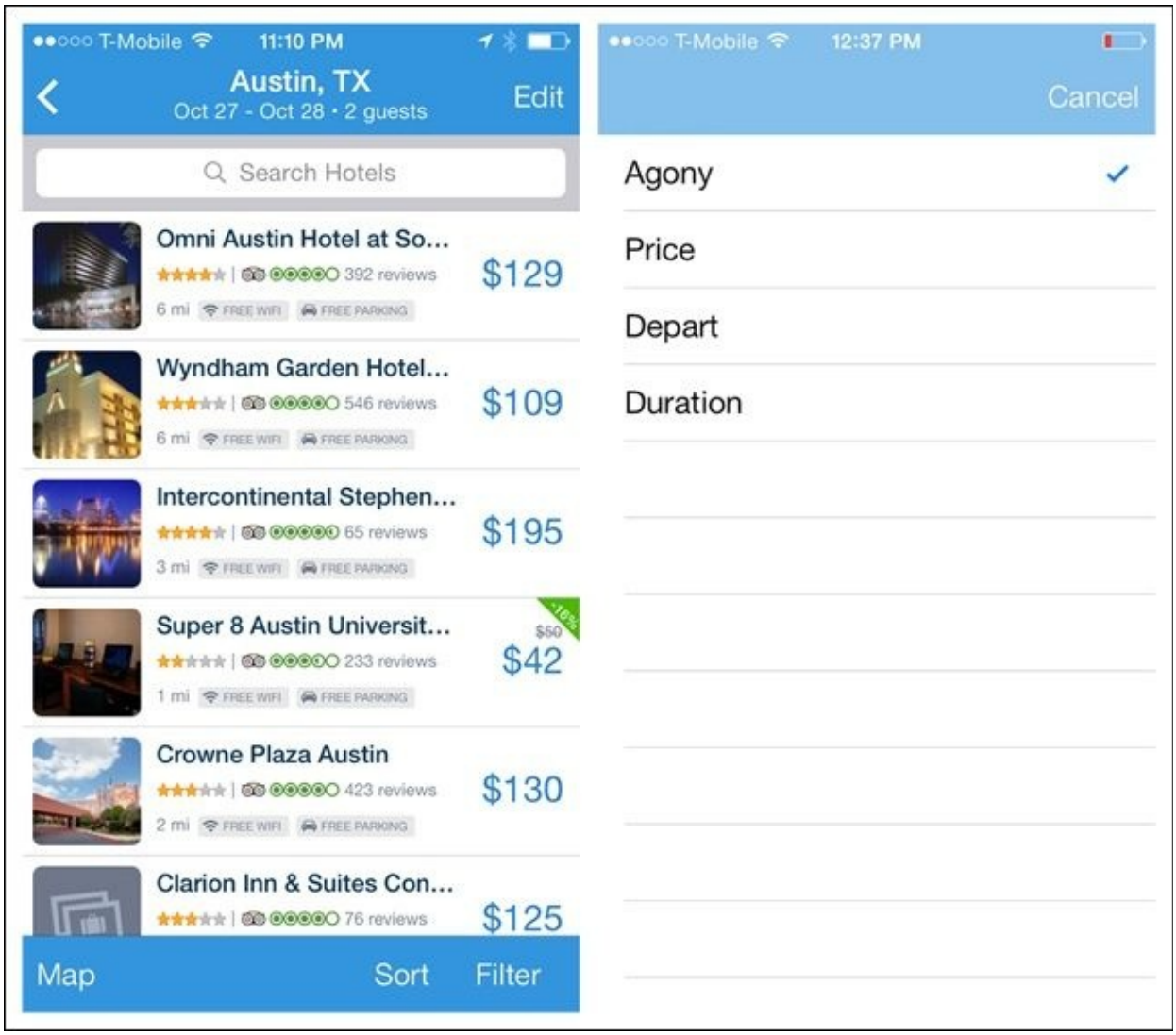



*Figure 4-44. Hipmunk for iOS*

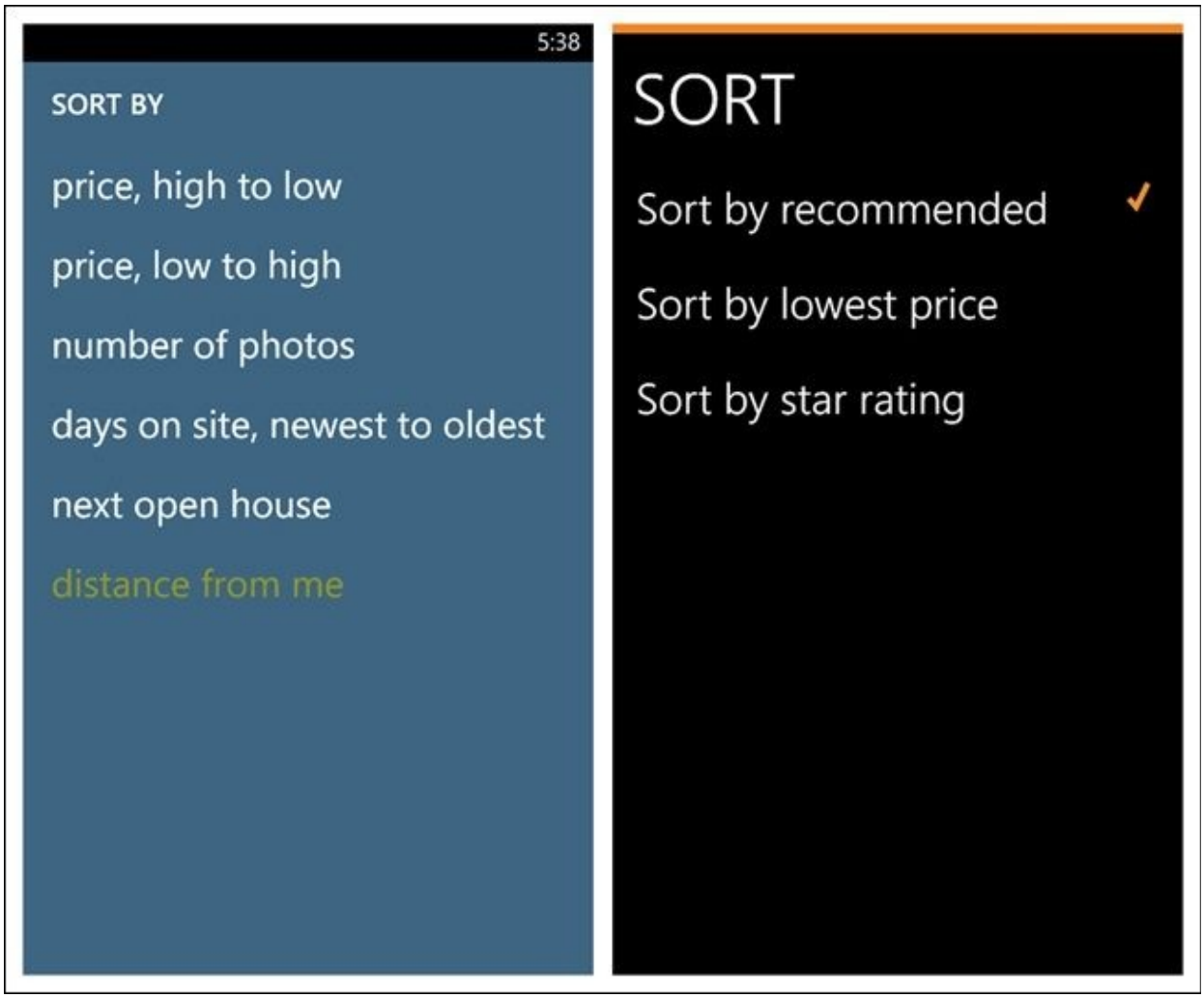



*Figure 4-45. Realtor.com and Kayak for Windows Phone*

Unfortunately, in Windows Phone apps the Sort Form seems to be the predominate pattern, with Newegg, which uses a Sort Overlay, being a welcome exception. Another option to consider for Windows Phone apps is the Pivot control, as shown in the Travelocity example.

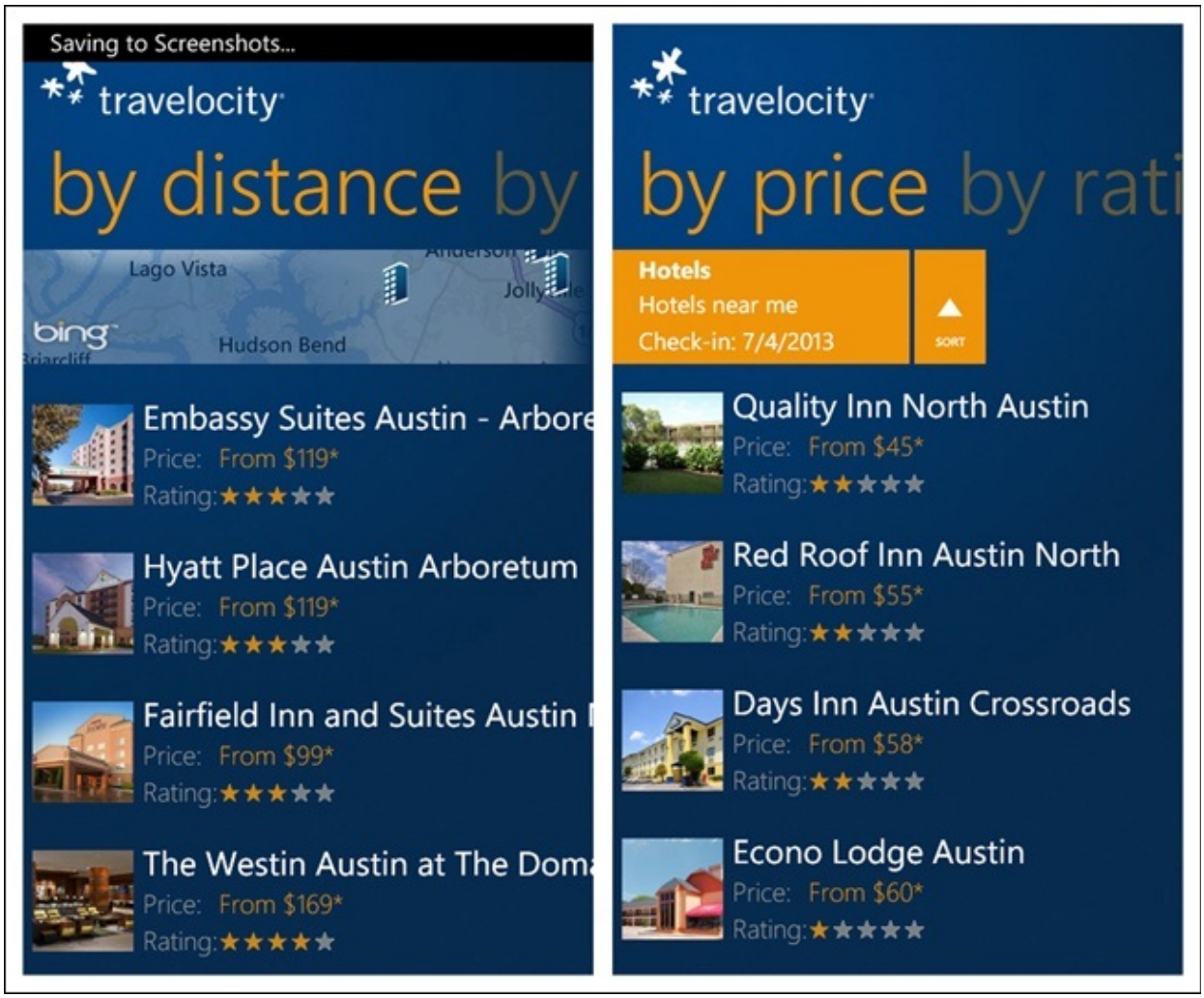



*Figure 4-46. Travelocity for Windows Phone: Pivot control for sort options*

Some apps combine sort and filter patterns in a single form. This can seem to make sense, as both are search refinement options. But as with Priceline and Yelp for iOS, it can result in long and crowded screens that veer into anti-pattern territory (see Chapter 11). Yelp for Windows Phone and Foursquare for iOS offer cleaner, single-page designs.

**NOTE**

Consider the more efficient Onscreen Sort or Sort Overlay patterns before opting for a Sort Form. If combining Sort and Filter in a single form, keep the screen as short and uncluttered as possible.

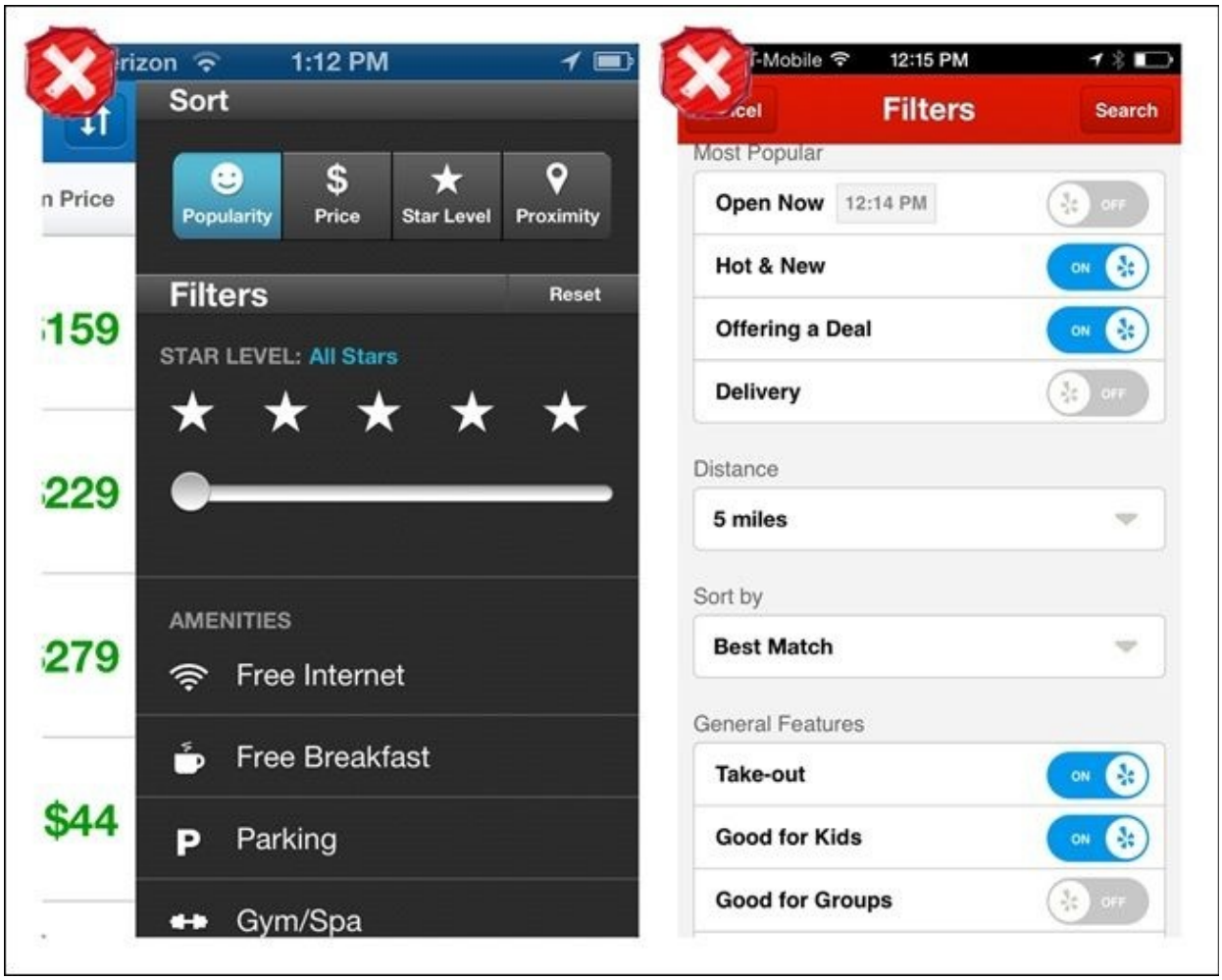



*Figure 4-47. Priceline and Yelp for iOS*

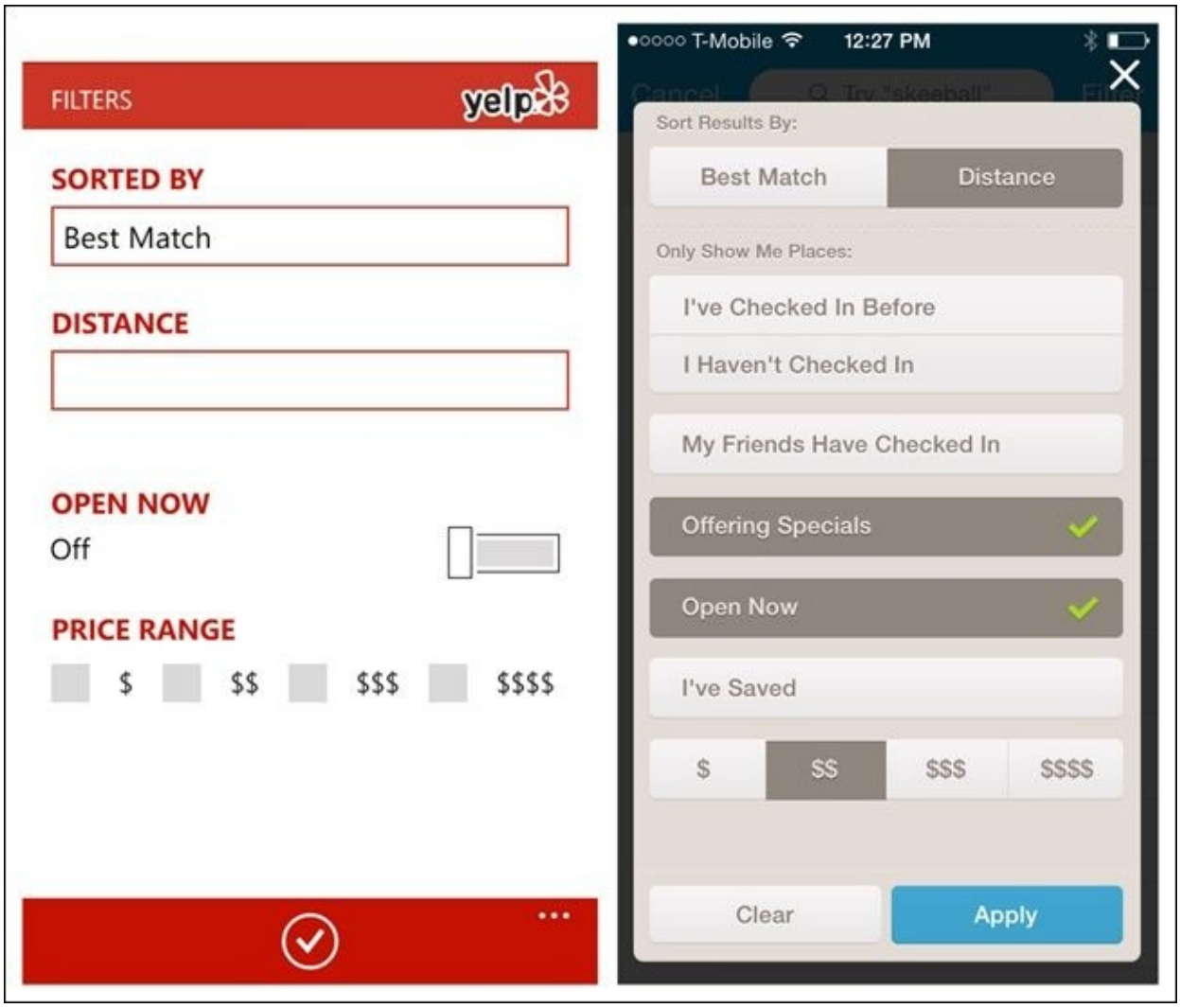



*Figure 4-48. Yelp for Windows Phone and Foursquare for iOS*

# Filter Patterns

Large sets of data can be so unwieldy as to be almost useless. To make sense of them, users often need to filter, or refine, the results. Filters let users select criteria to reduce data sets to their most manageable, relevant results. Common filtering patterns include:

- Onscreen Filter
- Filter Overlay
- Filter Form
- Filter Drawer
- Gesture-Based Filtering

Also see the earlier search pattern, Scoped Search, for more filtering techniques.

## Onscreen Filter

Like the Onscreen Sort, the Onscreen Filter is displayed on the Search Results screen. With one tap, the filter is applied.

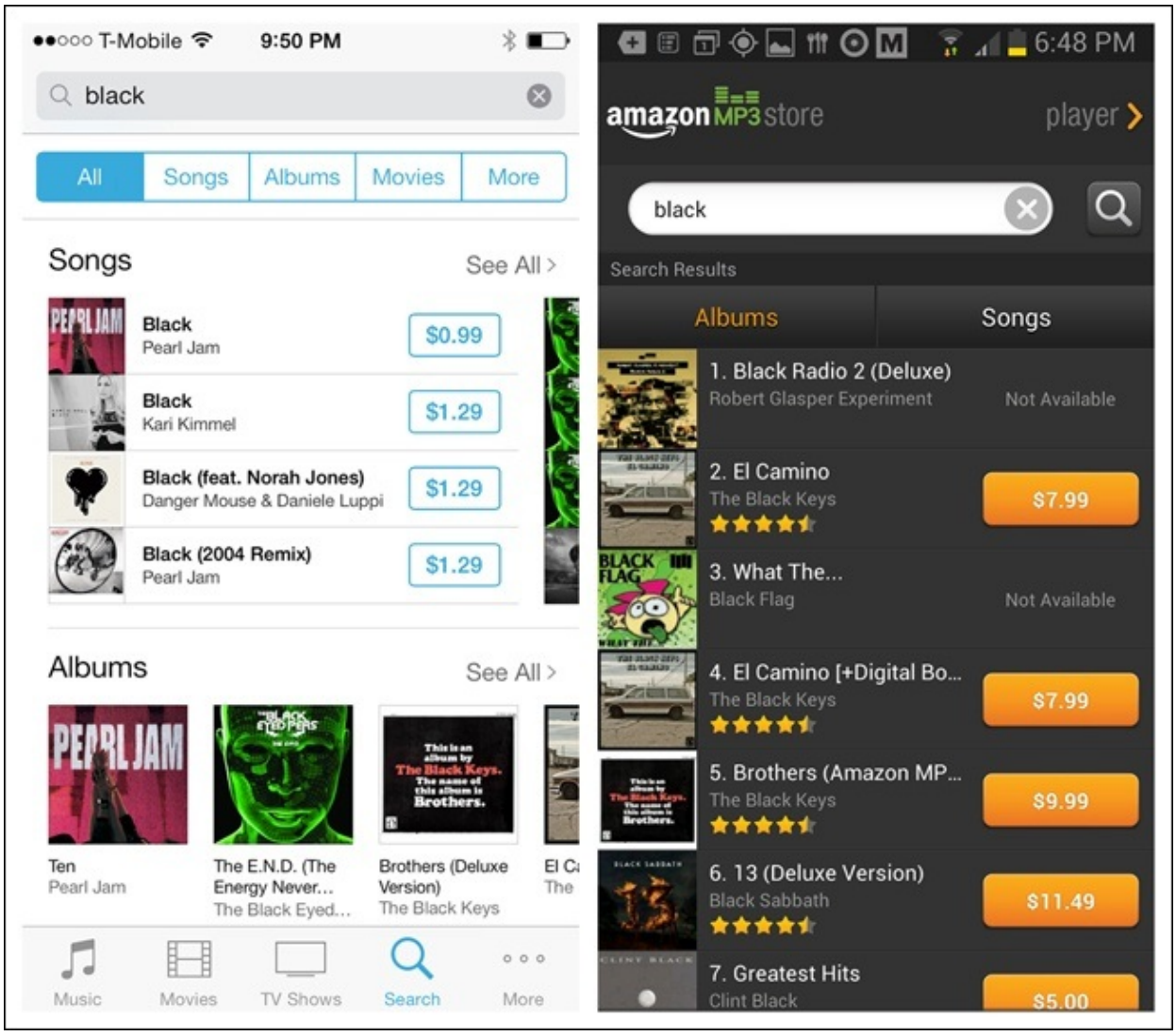



*Figure 4-49. iTunes for iOS and Amazon Cloud Player for Android: Onscreen Filters at top*

Google for Android shows its three most common Onscreen Filters across the bottom of the screen. Tapping More reveals other filter options. As a user scrolls the results, the filters hide, so more of the results are visible.

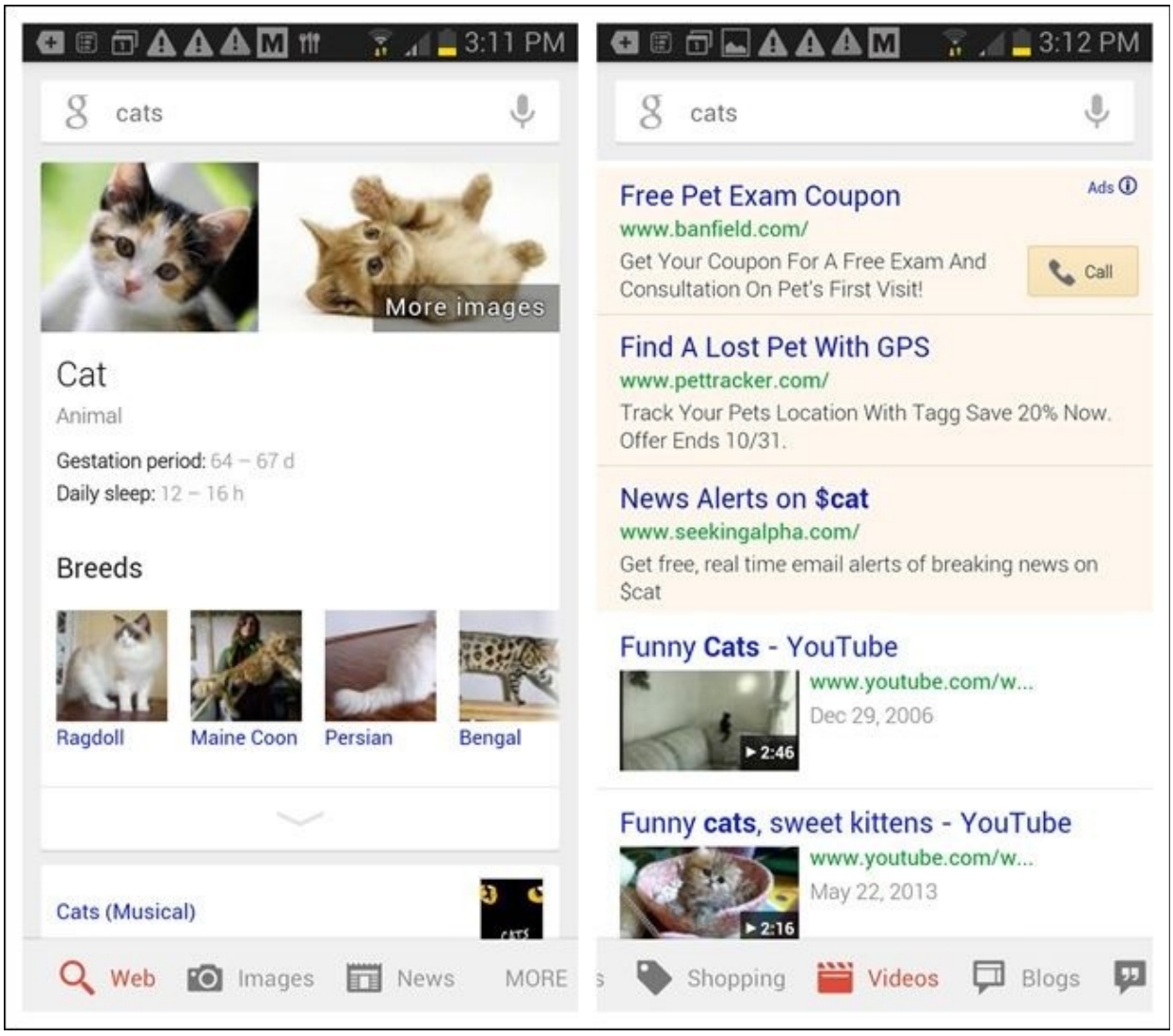



*Figure 4-50. Google for Android: Onscreen Filters across the bottom*

The Android versions of Groupon and the Play Store use a side-scrolling filter bar to let users quickly find certain types of coupons and apps, respectively. Android calls this control *Scrolling Tabs*, whereas Windows refers to it as a *Pivot Header*.

SXSW GO has a two-row filter for selecting the date, session type, and other advanced filter options. You can see when advanced filters are on because the color changes and the text is updated from Filter Off to Filter On.

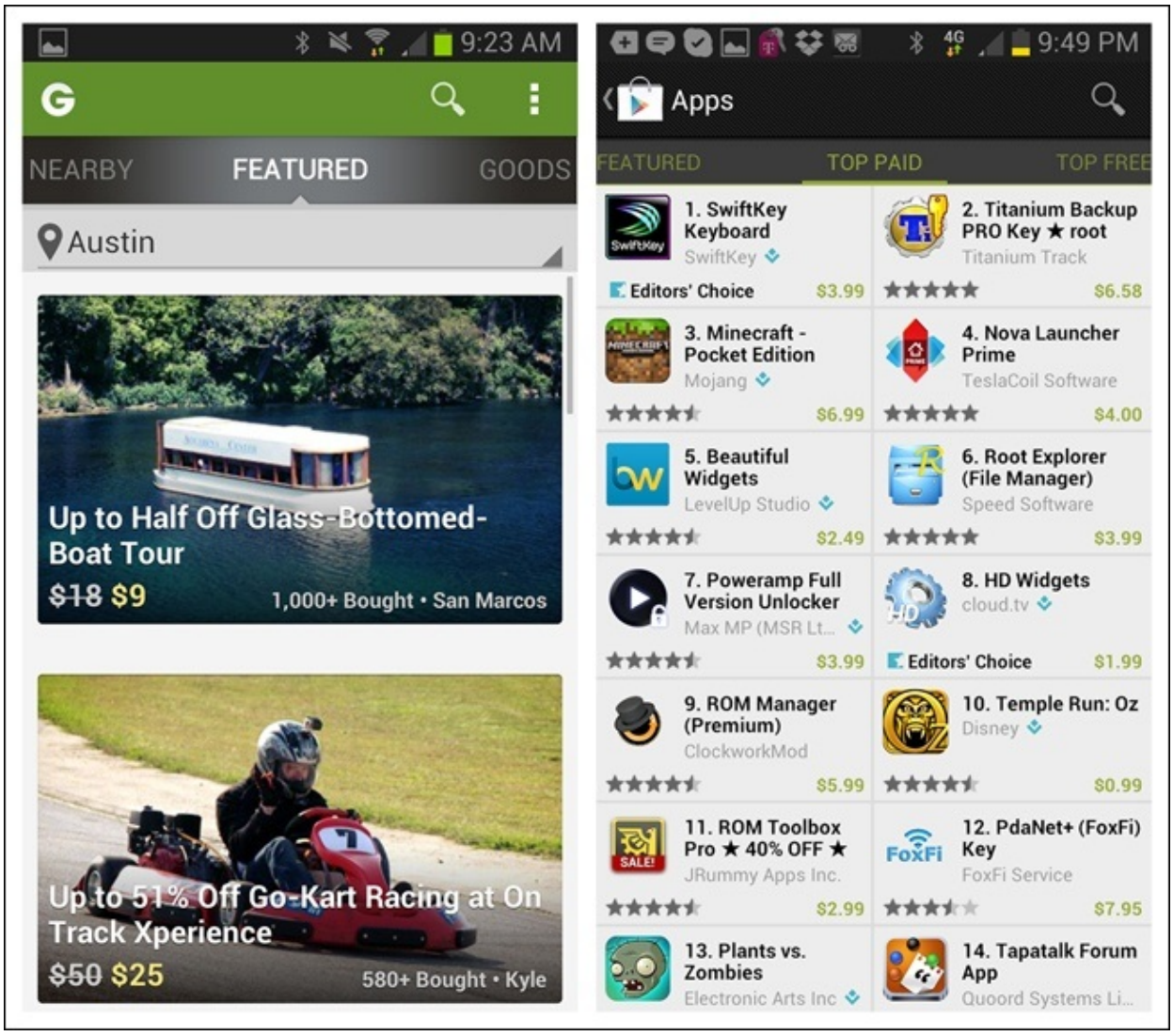



*Figure 4-51. Groupon and Play Store for Android: Scrolling Tabs provide filtering*

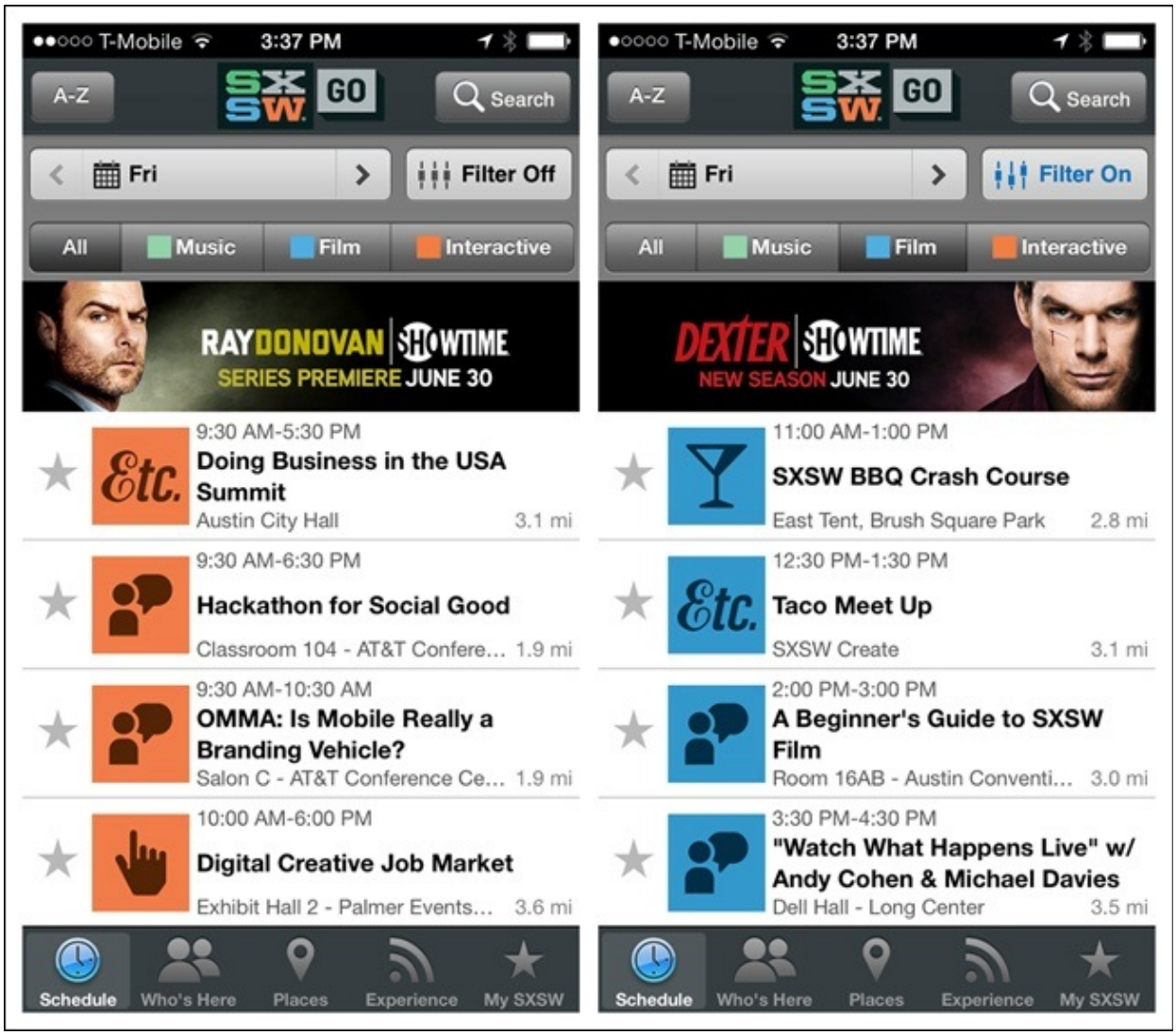



*Figure 4-52. SXSW GO for iOS: double-decker filter*

In Android apps, the Spinner control can work nicely for filtering a list. Though it doesn't show all filter options at a glance, it can clearly show what is selected.

**NOTE**

Filtering options should be clearly worded and easy to understand. When possible, provide visual cues to show the filters that are applied or "on," especially if the filter controls are sometimes hidden.

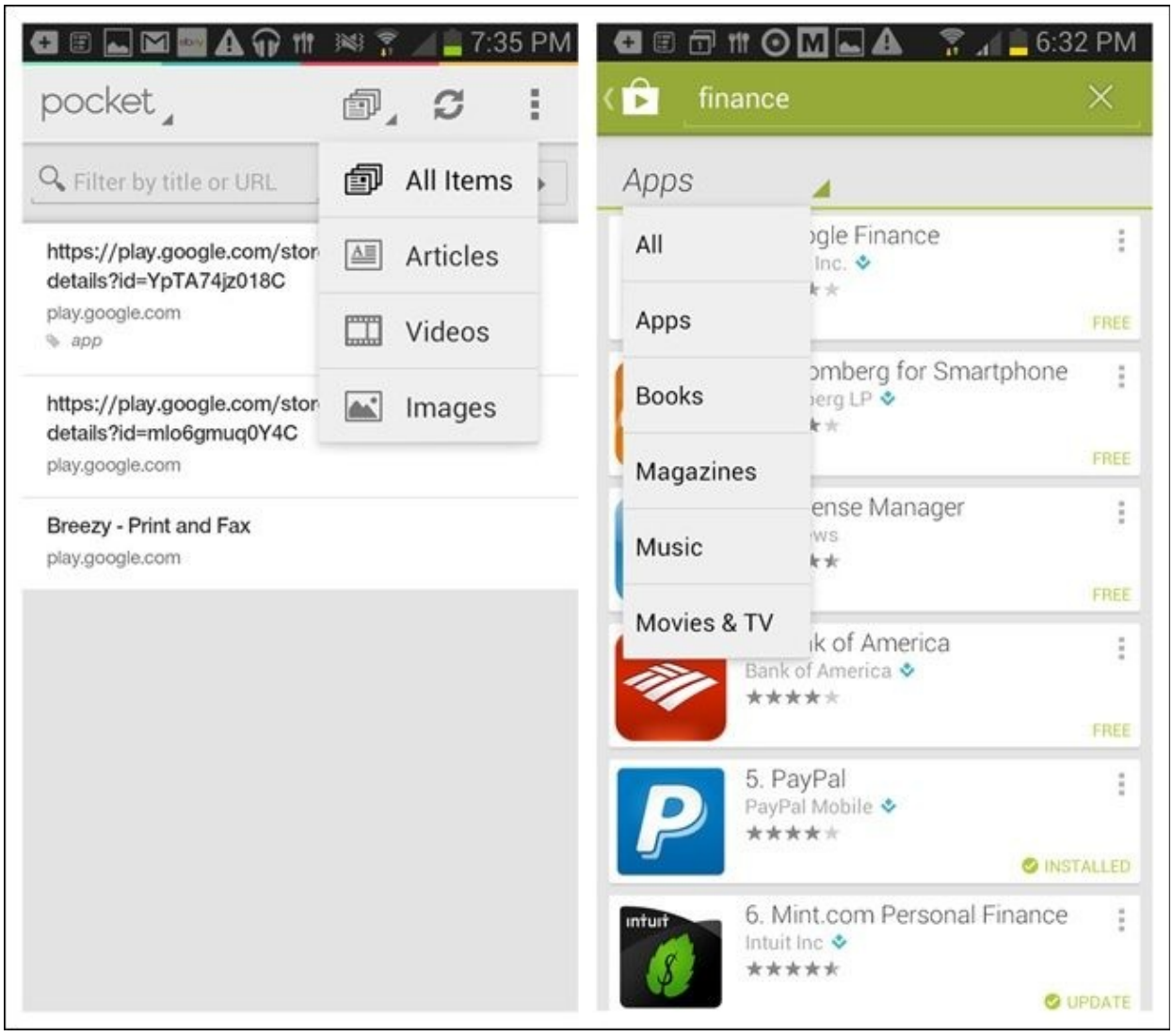



*Figure 4-53. Pocket and Play Store for Android: Spinner control filter*

## Filter Overlay

The Filter Overlay pattern is a viable alternative to the Onscreen Filter. It doesn't take up valuable screen real estate or force a short label where a longer, more explicit one would work better. The Overlay may be initiated from a Filter or Refine label, or icons that clearly depict the filtering criteria that can be applied.

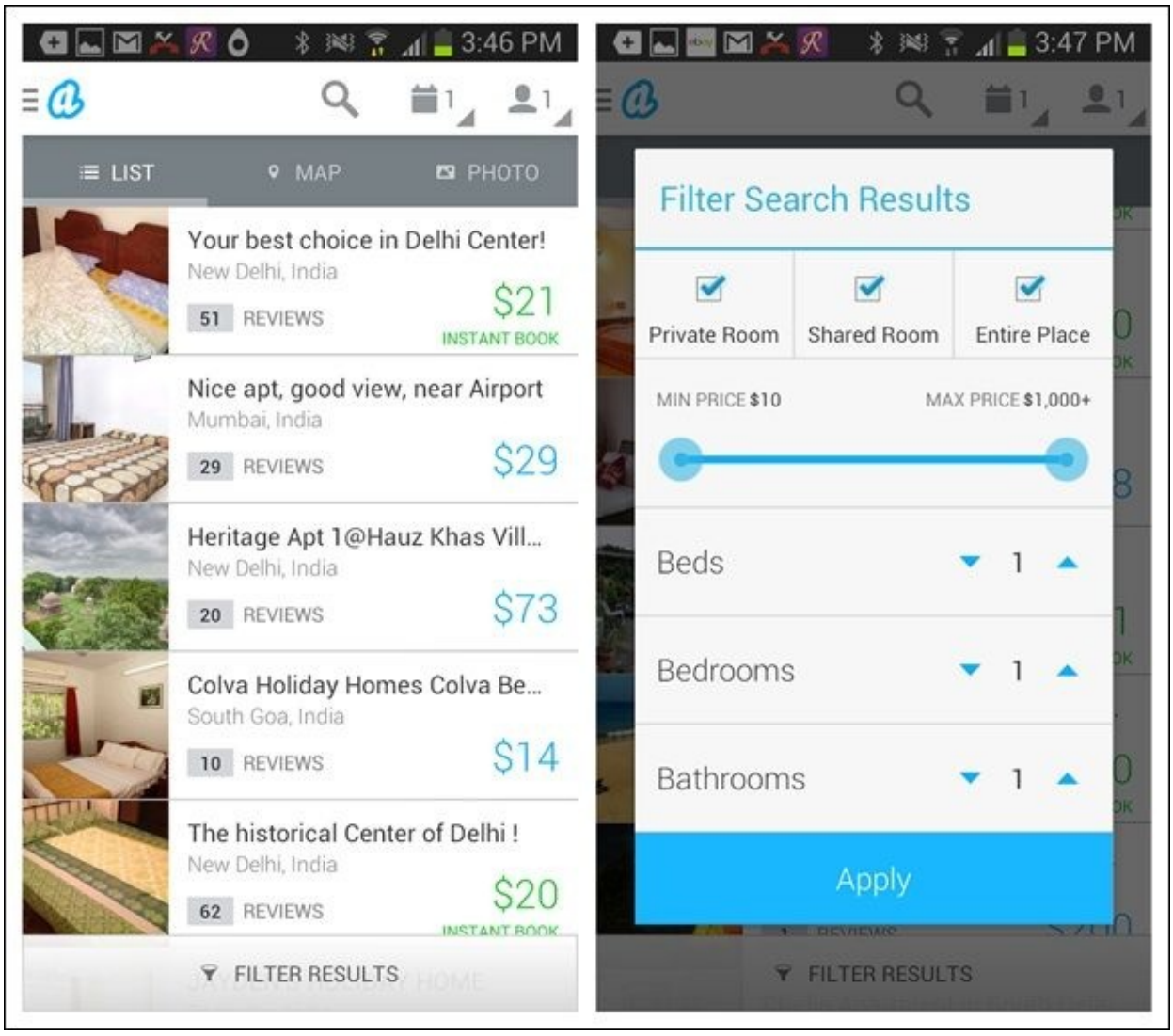



*Figure 4-54. Airbnb for Android: Filter Overlay*

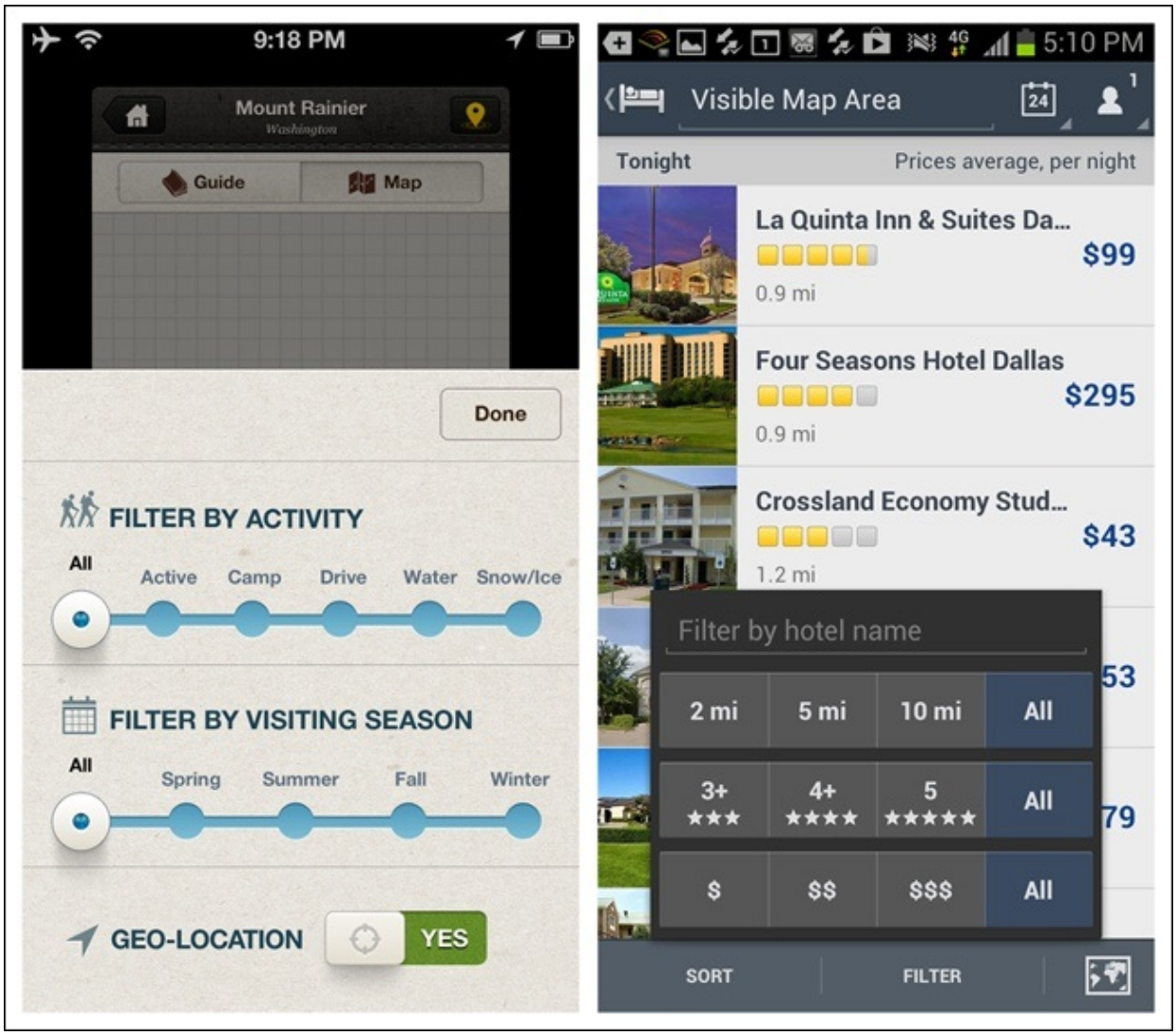



*Figure 4-55. National Parks for iOS and Expedia for Android: Filter Overlays*

Ness uses icons in the title bar to offer three filtering criteria, with each having its own distinct Overlay. The refined selections display beneath the segmented control.

**NOTE**

Filter Overlays are useful for preserving screen real estate and including more advanced filtering options than Onscreen Filters. But they require more user time and effort, so consider the trade-offs before choosing this pattern.

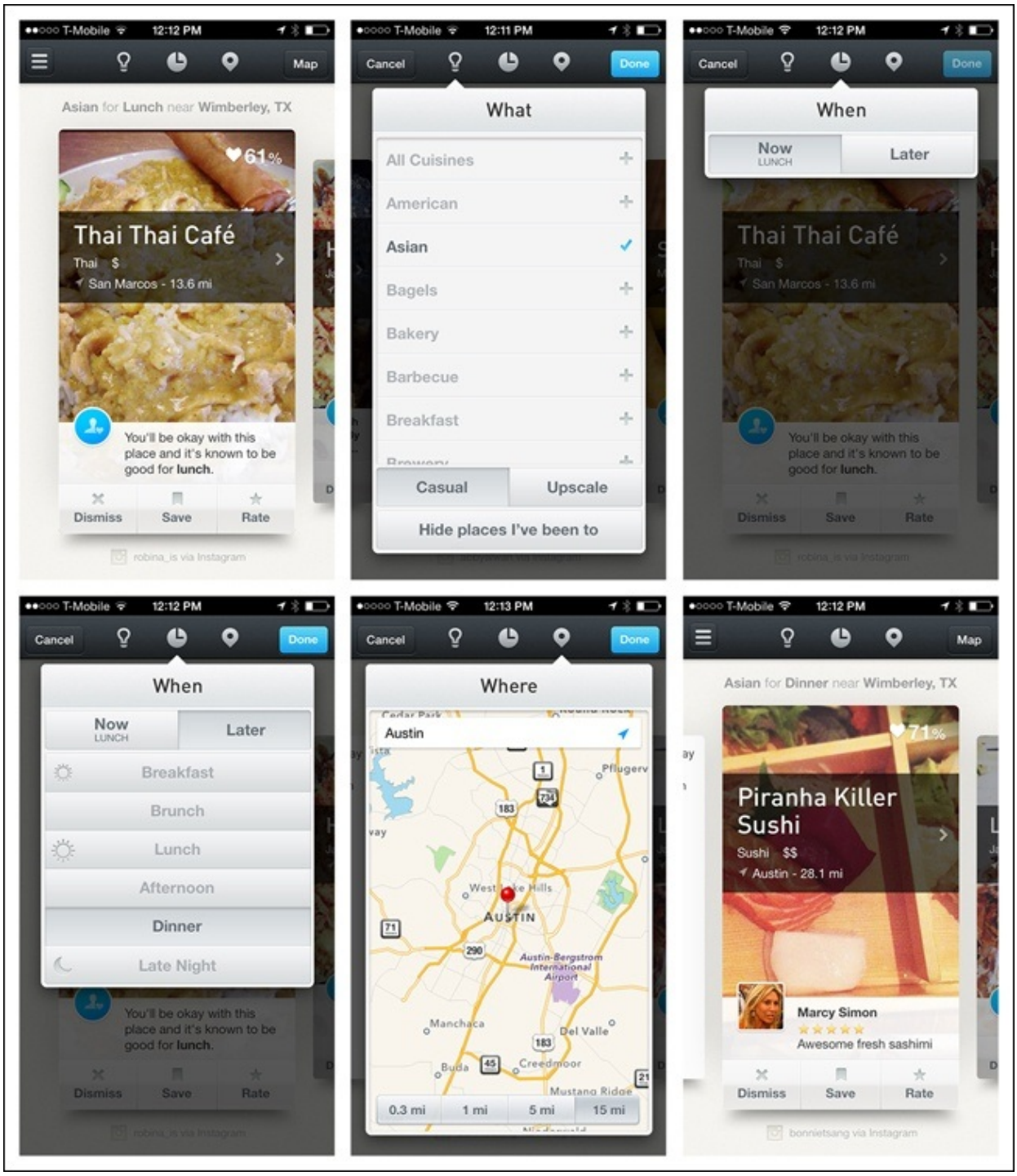



*Figure 4-56. Ness for iOS: Filter Overlays for each filtering option*

## Filter Form

The Filter Form also requires more effort from the user, who must open the form, select one or more options, and explicitly "apply" them. Filter Forms might be the best fit where you have a large number of filter options that lend

themselves to being grouped visually.

By necessity, Filter Forms momentarily hide the results being acted on. Note how Zillow for iOS dynamically updates the number of results as selections are made — this helps the user determine when the results list is manageable.

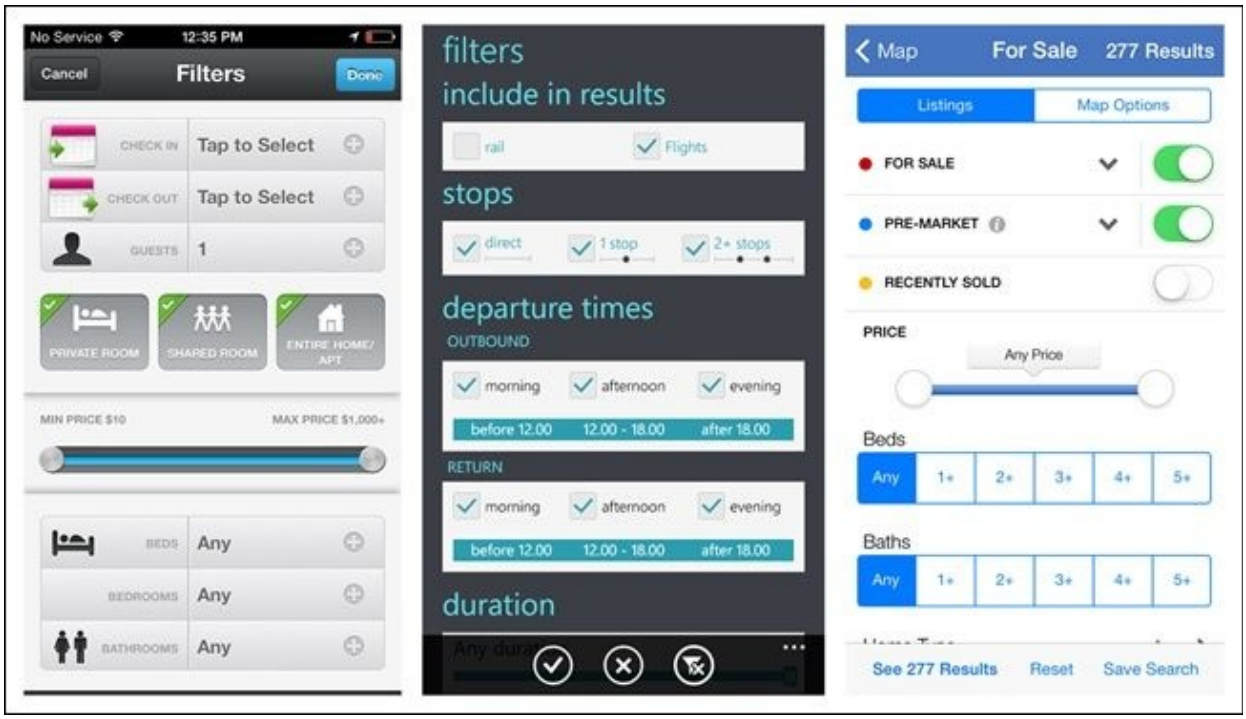



*Figure 4-57. Airbnb for iOS, Skyscanner for Windows Phone, and Zillow for iOS 7: Filter Forms*

The Filter Form also works well for nested, or hierarchal, selections, as illustrated in the Walmart app for iOS 7.

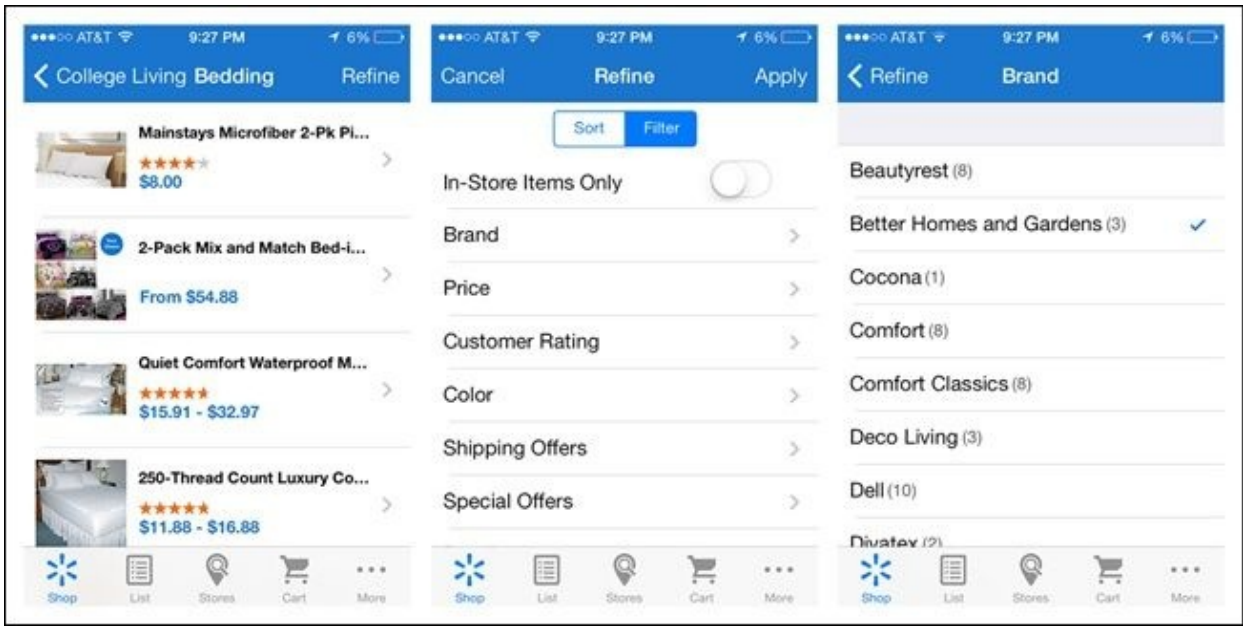



*Figure 4-58. Walmart for iOS 7: nested/hierarchal Filter Form*

Allrecipes for Windows Phone offers a Filter Form in context with the search field, similar to the Scoped Search pattern illustrated earlier in this chapter.

**NOTE**

As with Search Forms, keep Filter Forms as spare as possible and follow OS guidelines for form design. Resist the urge to innovate new, "clever" types of filter controls to avoid creating an anti-pattern (see Chapter 11).

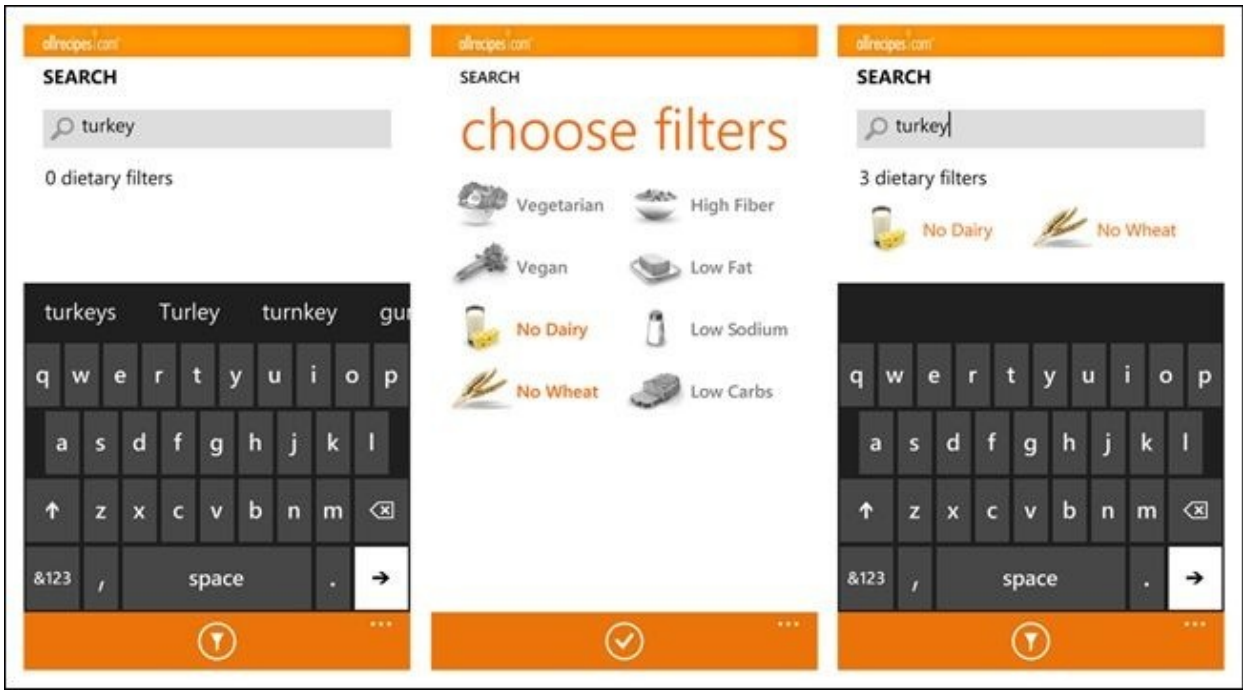



*Figure 4-59. Allrecipes for Windows Phone: Filter Form in context with search field*

## Filter Drawer

The Filter Drawer is very similar in some ways to both the Filter Overlay and the Filter Form. Like the Overlay and Form, it requires an extra tap of “apply” or “done” to get the filtered results. Its main distinction is that it appears to slide out from the current screen, thus allowing the user to stay in context with the search results being refined. Ideally, the Filter Drawer will reveal a glimpse of the results behind it.

In the Target app for iOS, the Filter Drawer slides down, completely obscuring the search results. Once the drawer is closed, there is no indication which filters have been applied. The only feedback that filtering is active is that the label “filter” is now blue.

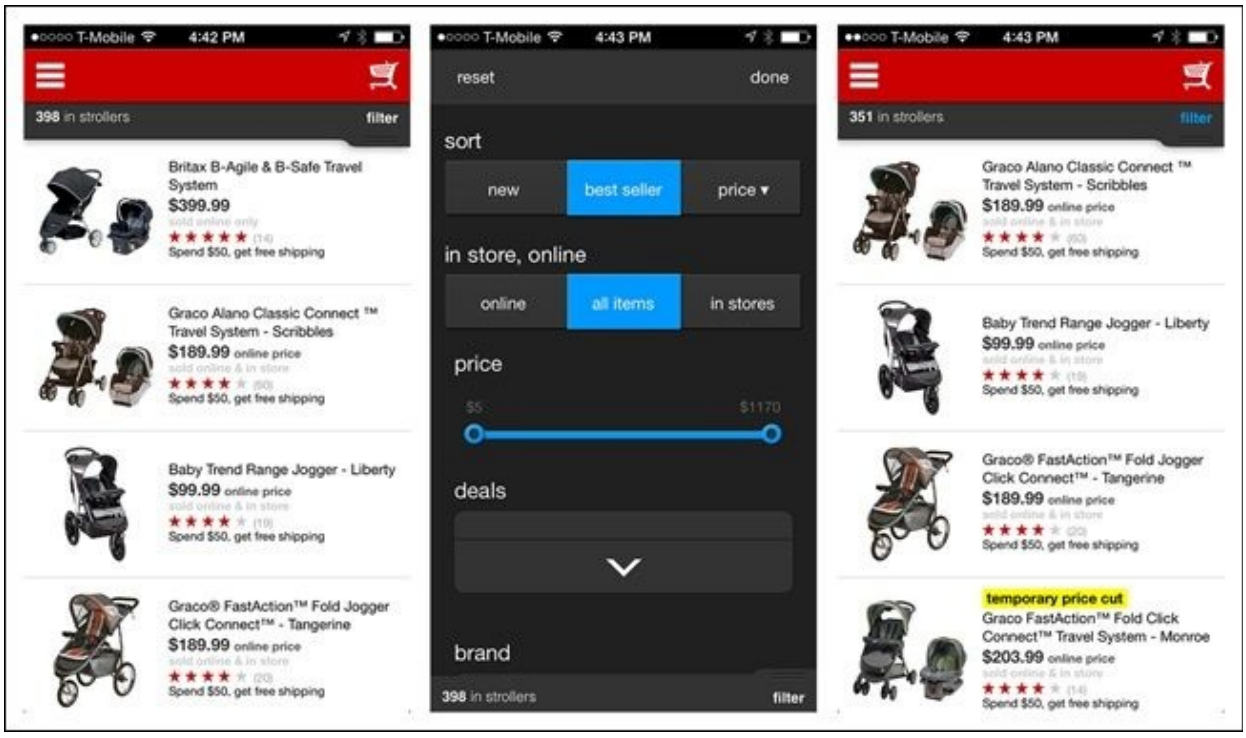



*Figure 4-60. Target for iOS: Filter Drawer*

Here's an example of a redesign that follows best practices for a Filter Drawer:

- Symmetrical actions to open and close the drawer
- Dynamic update of the number of search results
- Search results still visible in the background
- Explicit feedback when filters are applied

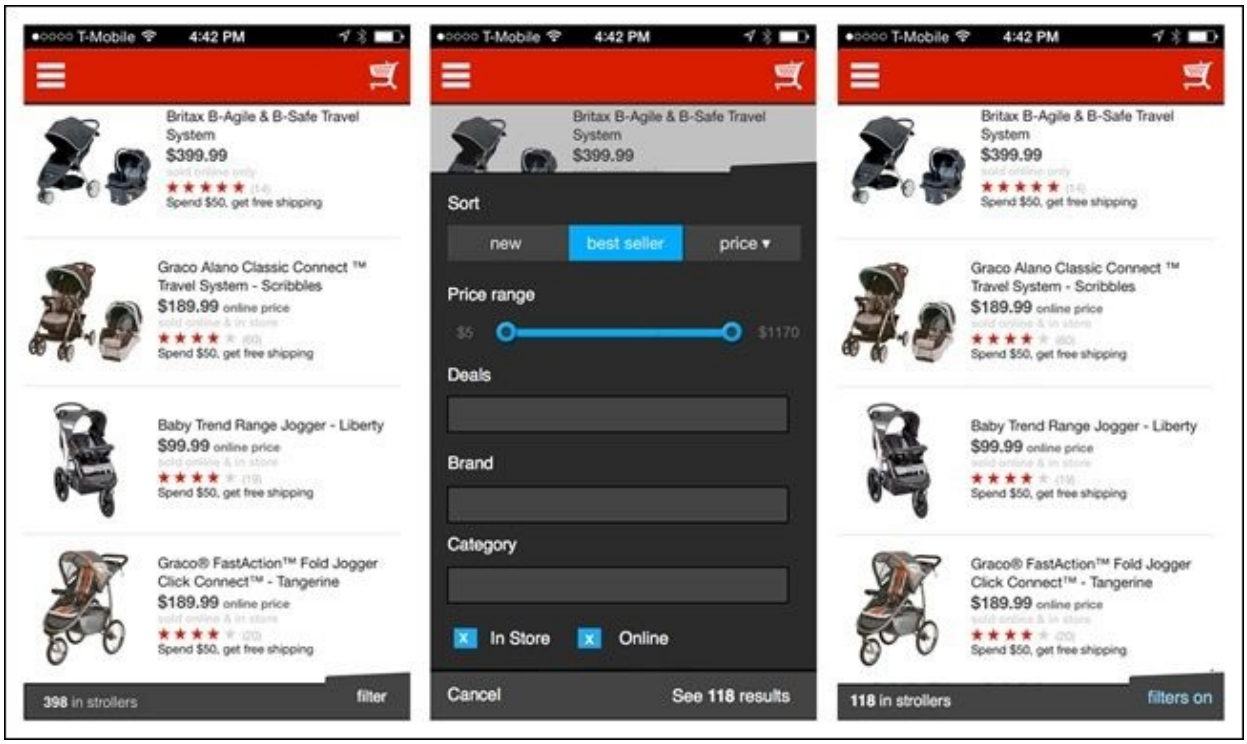



*Figure 4-61. Target for iOS: Filter Drawer, redesigned*

ToDo for iOS offers filtering in the Navigation Drawer. This will work only for simple filtering, and should be user tested. It could introduce inefficiencies if users would rather open a list and filter it directly.

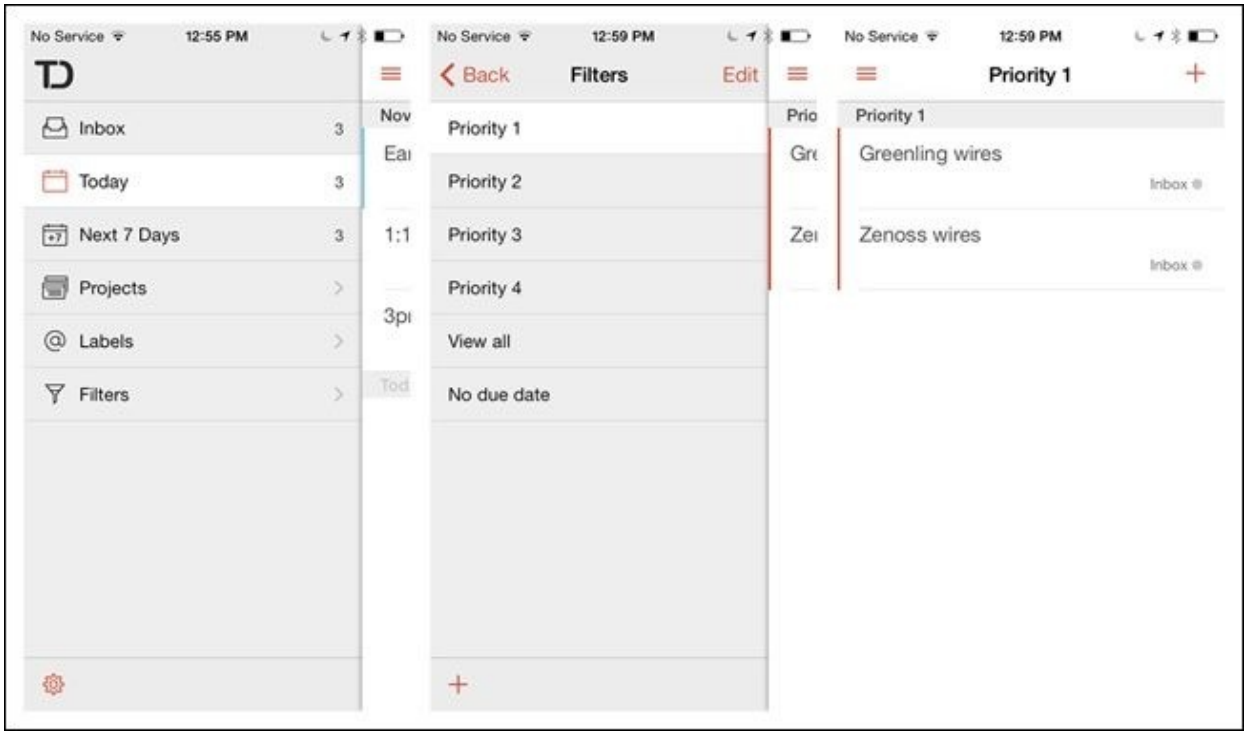


*Figure 4-62. ToDo for iOS: filters in the Navigation Drawer*

eBay for iOS has a robust, multilevel Filter Drawer. Tapping Refine on the Search Results screen slides in the Filter Drawer from the right, showing all the possible refinement options.

From the Search Results screen, you can also directly refine specific options — like Category, Platform, and Buying Format — by tapping on those labels, which will reveal the Filter Drawer open at the appropriate level.

The number of results is always displayed at the top of the drawer, and when it closes, the control bar indicates which filters are currently applied. Take a look at the video to see exactly how the interaction design works.

**NOTE**

The Filter Drawer can help users stay in context with the results being refined. Show the updated number of results as options are applied, and provide a visual indication of whether filtering is active in the drawer's closed state.

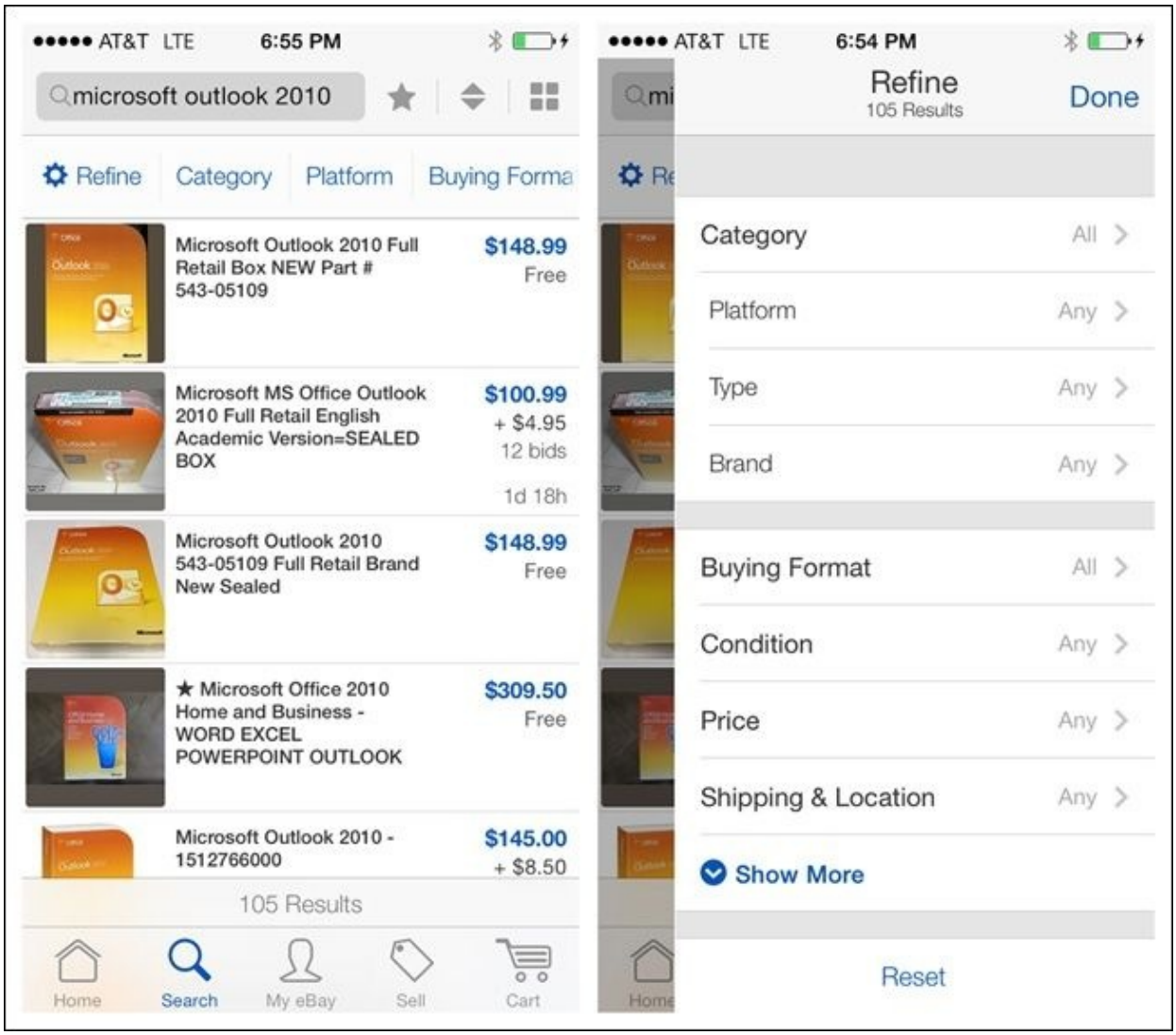



*Figure 4-63. eBay for iOS: multilevel Filter Drawer (http://www.youtube.com/watch?v=M_R3eSFcML0)*

## Gesture-Based Filtering

Pinch, zoom, and pan gestures can be used for filtering results whose relevance is dependent on location, as on a map, chart, or graph. HomeAway for iOS shows a default view of a certain radius. Pinch–zoom in to reduce the radius and thus the results shown; pinch–zoom out to widen the radius and show more results. Kayak for Windows offers the same functionality.

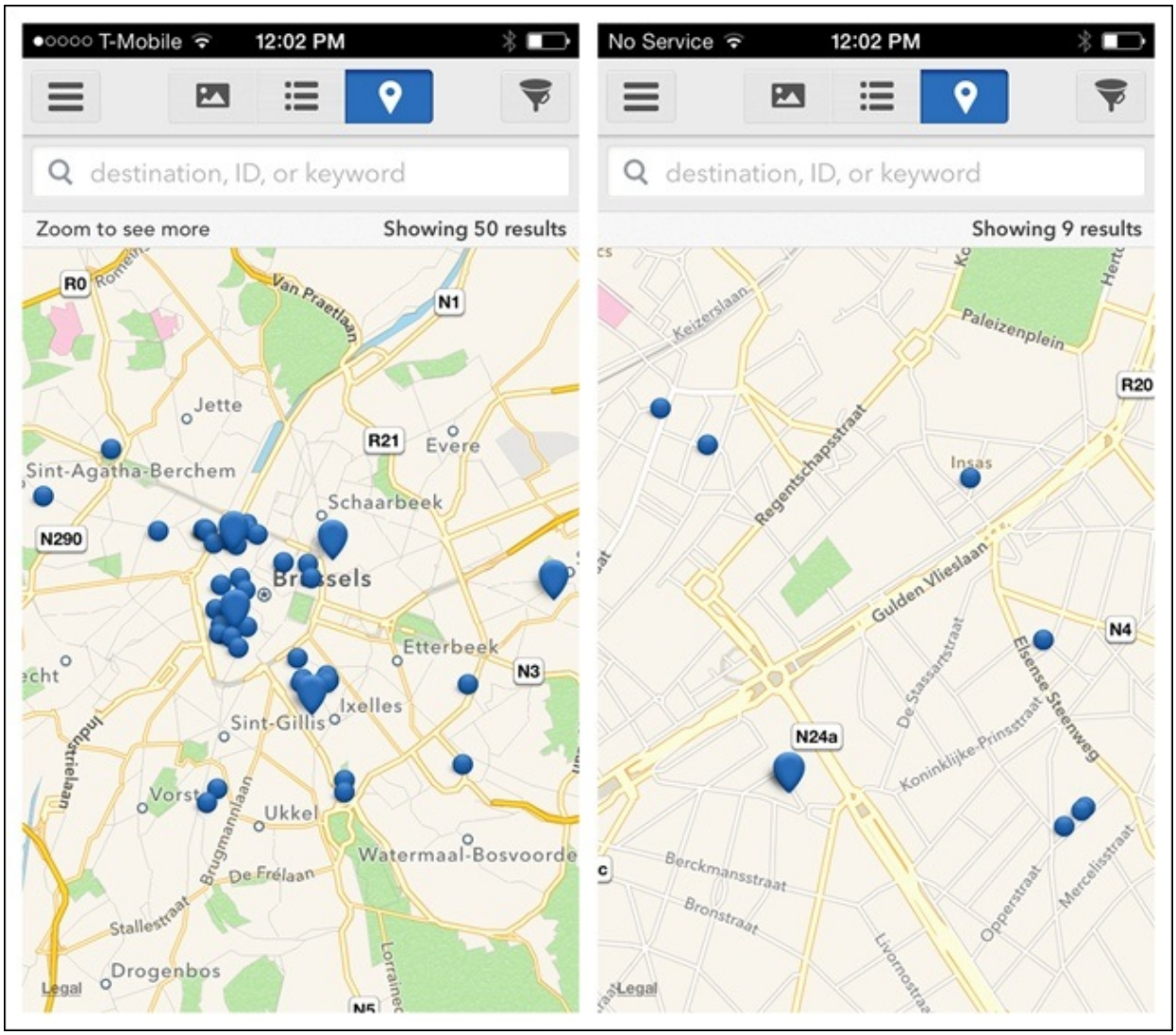



*Figure 4-64. HomeAway for iOS: Gesture-Based Filtering*

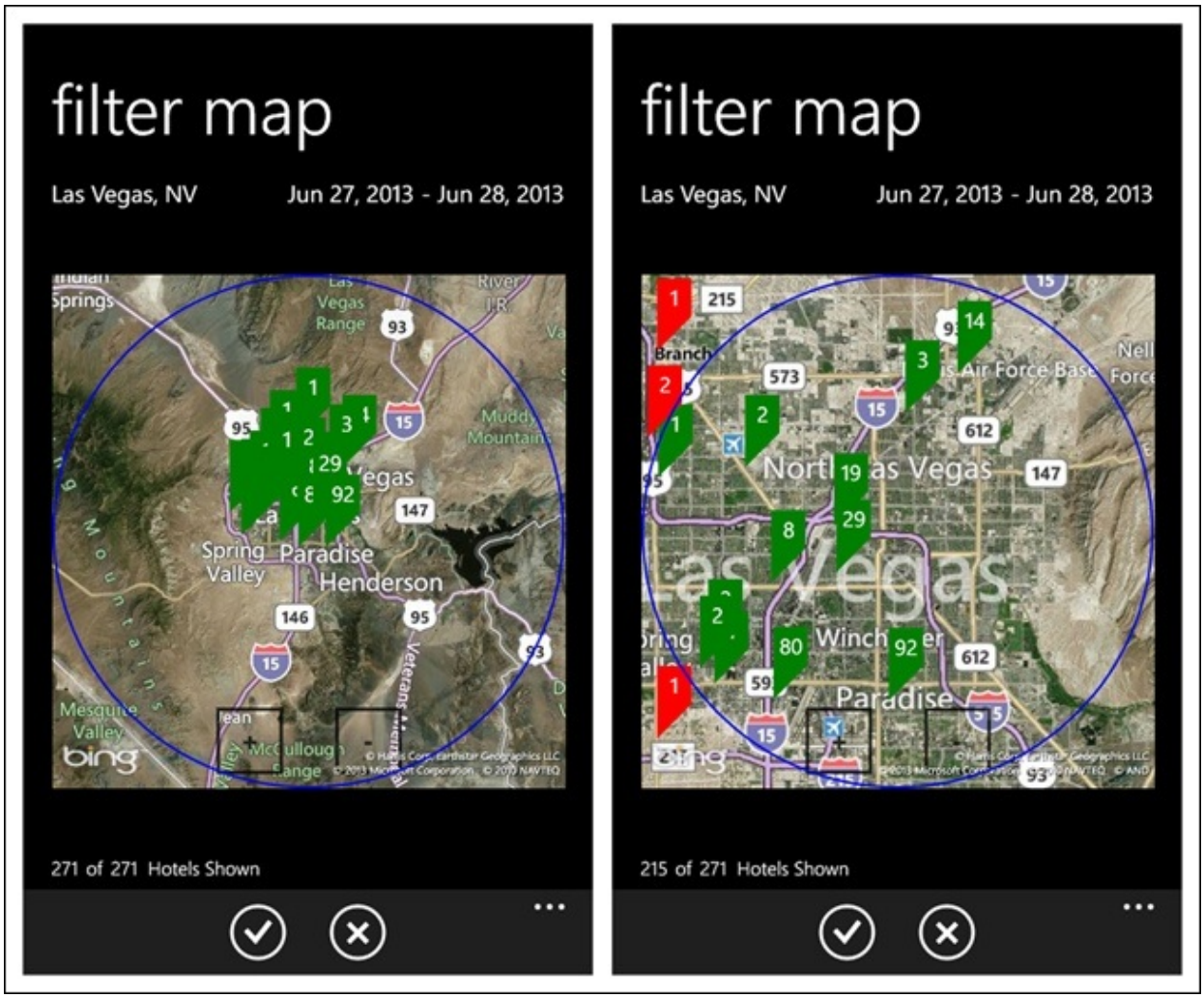
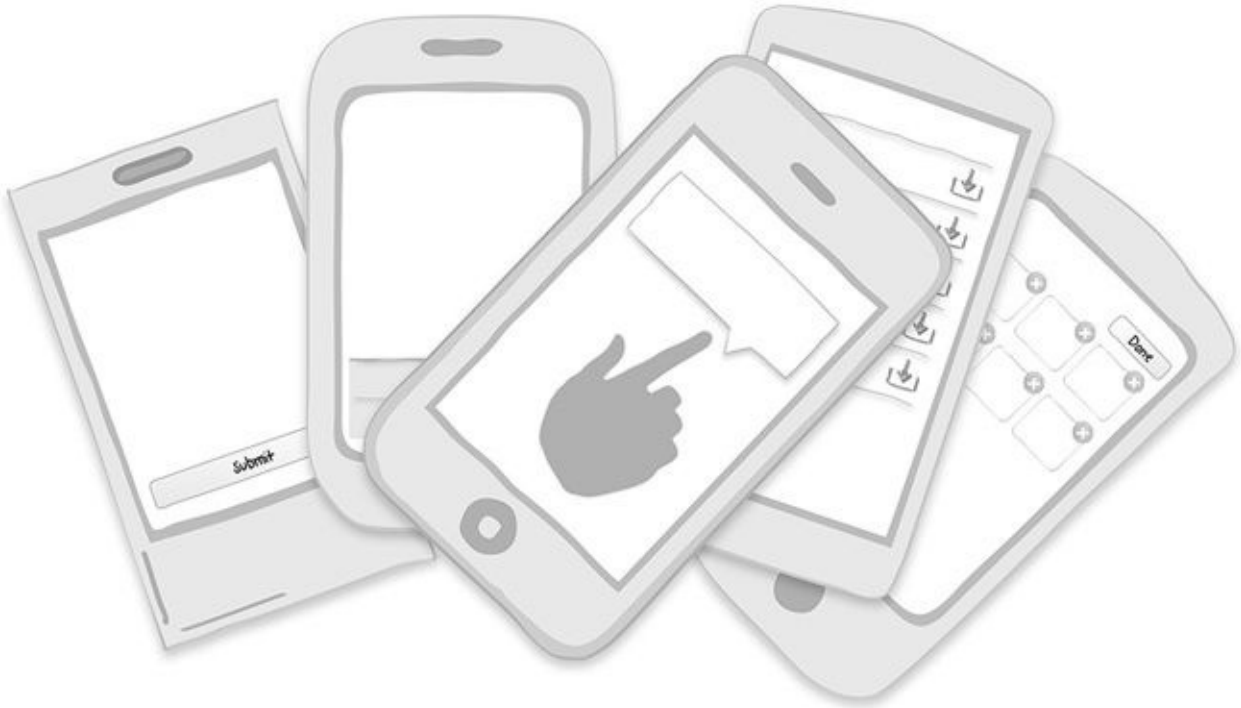



*Figure 4-65. Kayak for Windows Phone: Gesture-Based Filtering*

**NOTE**

Gesture-Based Filtering provides an intuitive way to refine results whose relevance to the user depends on their location. Consider combining it with other filtering options, since location may be only one of the user's criteria.

# Chapter 5. Tools




*Patterns*

> Toolbar, Toolbox, Call to Action Button, Inline Actions, Multi-State Buttons, Contextual Tools, Bulk Actions, Lock Screen Controls

In Chapter 1, we looked at how screen size constraints are forcing designers to think off-canvas and explore new solutions. Before diving into this chapter, I encourage you to embrace another challenge — shed three decades of abstract desktop metaphors and start designing, really designing, for touch interfaces. For inspiration, start with Josh Clark's "Buttons Are a Hack" campaign (*http://globalmoxie.com/blog/buttons-are-a-hack.shtml*):

> **Buttons are a hack.** As in the real world, they're often necessary, but they work at a distance — secondary tools to work on primary objects. A light switch here turns on a lightbulb there. These indirect interactions must be learned; they're not contextually obvious. The revolution that touchscreen devices are working is that they allow us, more and more, to use primary content as a control, to create the illusion of direct interaction.
> I don't mean to suggest that we throw out all of our familiar buttons entirely. Light switches shall remain necessary, after all, and so shall buttons, especially where it's necessary to trigger abstract actions ("share via Twitter," for example). But it's important to recognize those devices for what they are: necessary hacks for moments when direct interaction isn't possible. Touchscreen interfaces allow that direct interaction in many more contexts. As new solutions arise, we should be open to putting our time-tested workarounds aside. When designing an interaction for touch, always ask: do I really need

another button or control for this?

Let’s look at examples from TweetCaster and Kayak. In TweetCaster, the main interface is a list of tweets. Tapping one opens it in a new screen. To navigate to the previous or next tweet, you use two buttons (up and down arrows) in the navigation bar. But do we really need buttons for navigating to next or previous items, or is this just a holdover from desktop design? Page Swiping is a more efficient and natural way to navigate content on a touchscreen, as you’ll see when we look at that pattern later in this chapter.

And what about the plus/minus magnifying glasses on the Kayak map? I think I remember those from my first computer in the ’90s. Ditch the zoom buttons, people — the pinch-to-zoom gesture’s got this covered.

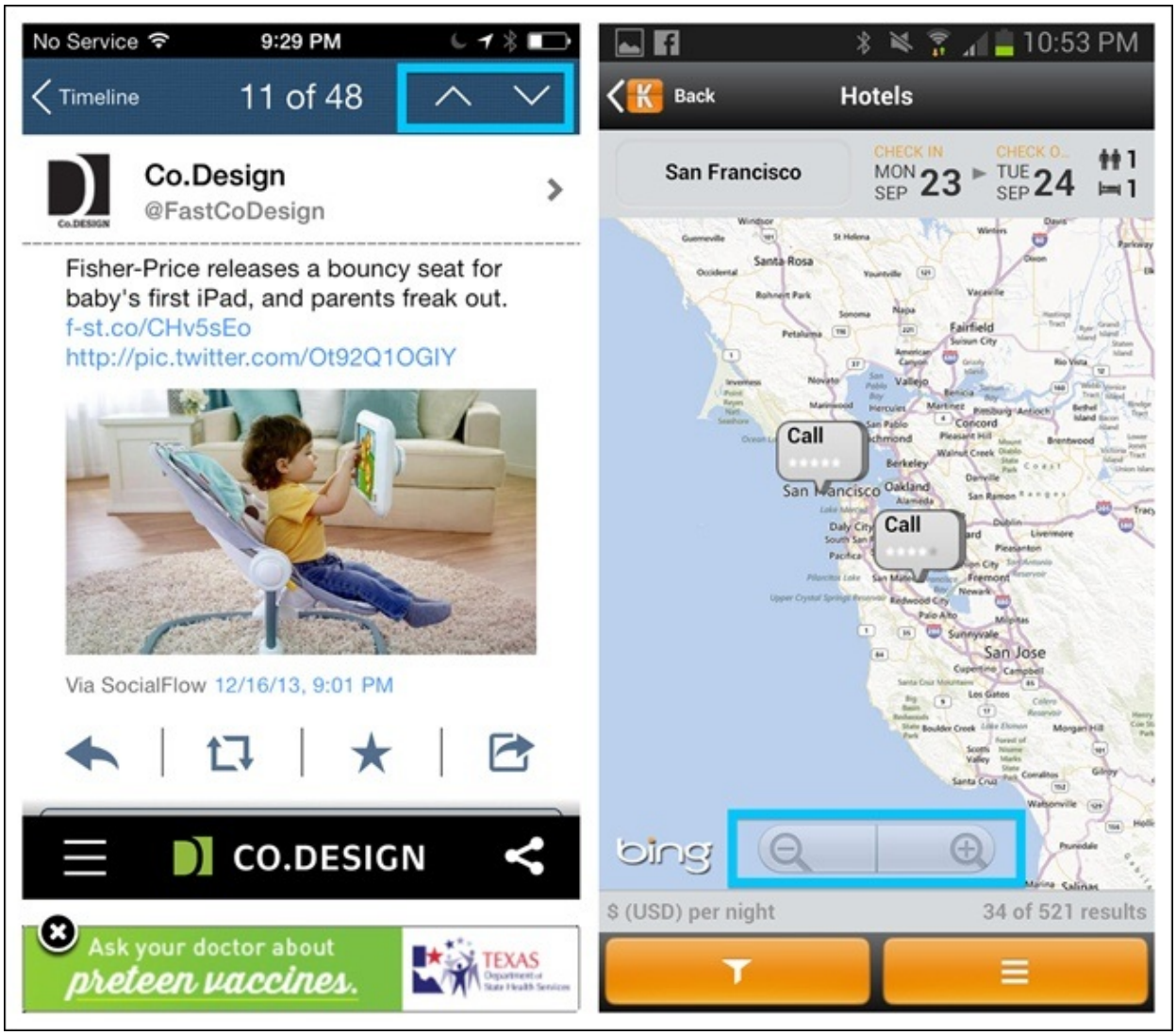



*Figure 5-1. TweetCaster for iOS and Kayak for Android: outdated desktop metaphors*

I’m fairly certain that by the time the next edition of this book comes out, some new, creative, touch-based tool patterns will have emerged. For now, we’ll look at the most common ones today. True, some of these patterns are versions of the old desktop metaphors — they’re simply triggered by a touch instead of a click. But others incorporate touch to respond to users in ways that desktop interfaces can’t.

# Toolbar

The Toolbar is one of the most common ways to provide screen-specific actions. Each operating system has its own terminology and guidelines for the Toolbar.

## iOS

The iOS Human Interface Guideline (*http://bit.ly/NoLrna*) says:

> Although a toolbar looks similar to a navigation bar or a tab bar, it doesn't enable navigation. Instead, a toolbar gives users controls that act on the contents of the current screen. It always appears at the bottom edge of a screen or view on iPhone.

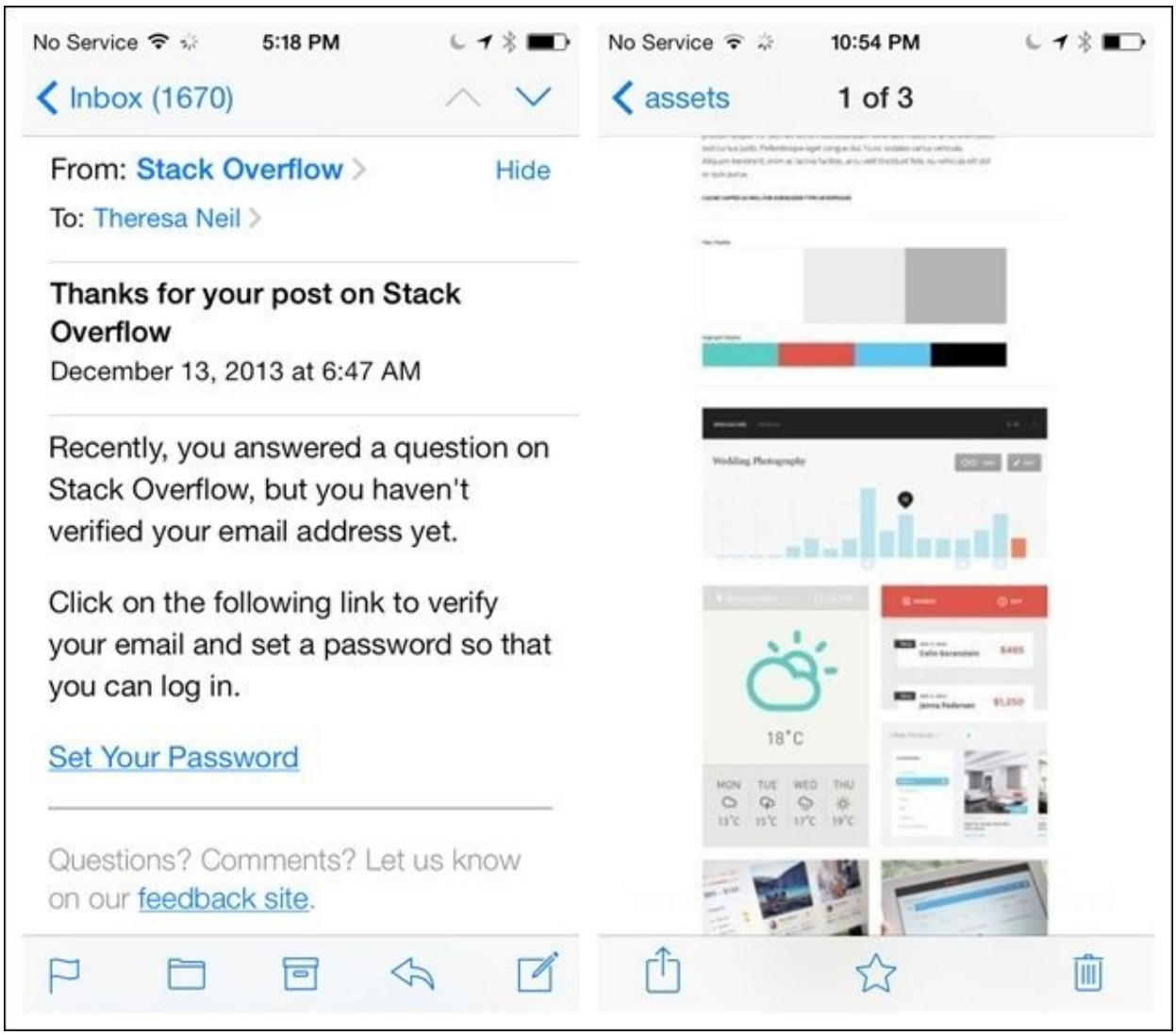



*Figure 5-2. Mail and Dropbox for iOS: don't confuse Toolbars with navigation bars*

Expedia has created a cascading menu from the base Toolbar, and Wunderlist designed a custom collapsing Toolbar to save real estate.

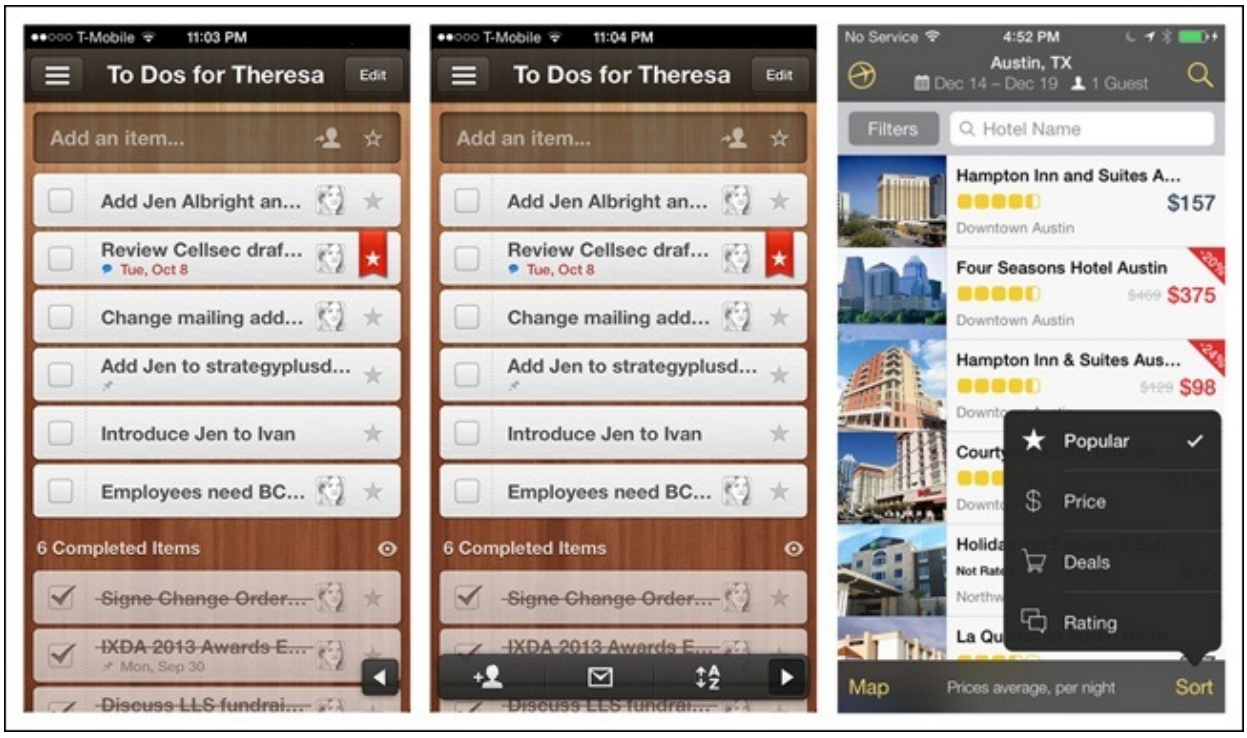



*Figure 5-3. Expedia and Wunderlist Custom Toolbars*

Alternatively, an action button may launch an Action Sheet, which displays additional options for the selected button. In Narrato, for instance, the photo button opens a standard Action Sheet for choosing the photo source, while the smiley face button opens a custom Action Sheet for selecting a mood. This is an example of a smart design that stays within the spirit — if not the letter — of the iOS Design Guide.


*Figure 5-4. Narrato for iOS: two different Action Sheet designs*

## Android

The Jelly Bean release brought some changes to the Android interface (*http://developer.android.com/design/patterns/actionbar.html*). There is a not a dedicated Toolbar, as in iOS. Instead, action buttons are assigned to the top right of the Action Bar, a dedicated piece of real estate at the top of each screen that is generally persistent throughout the app.


*Figure 5-5. The Action Bar is split into four functional areas that apply to most apps*

The action buttons should be specific to the current page context, like in Gmail. Primary actions should be visible in the Action Bar at the top, but secondary actions belong in the Overflow Menu, which can be accessed via the vertical ellipsis icon.

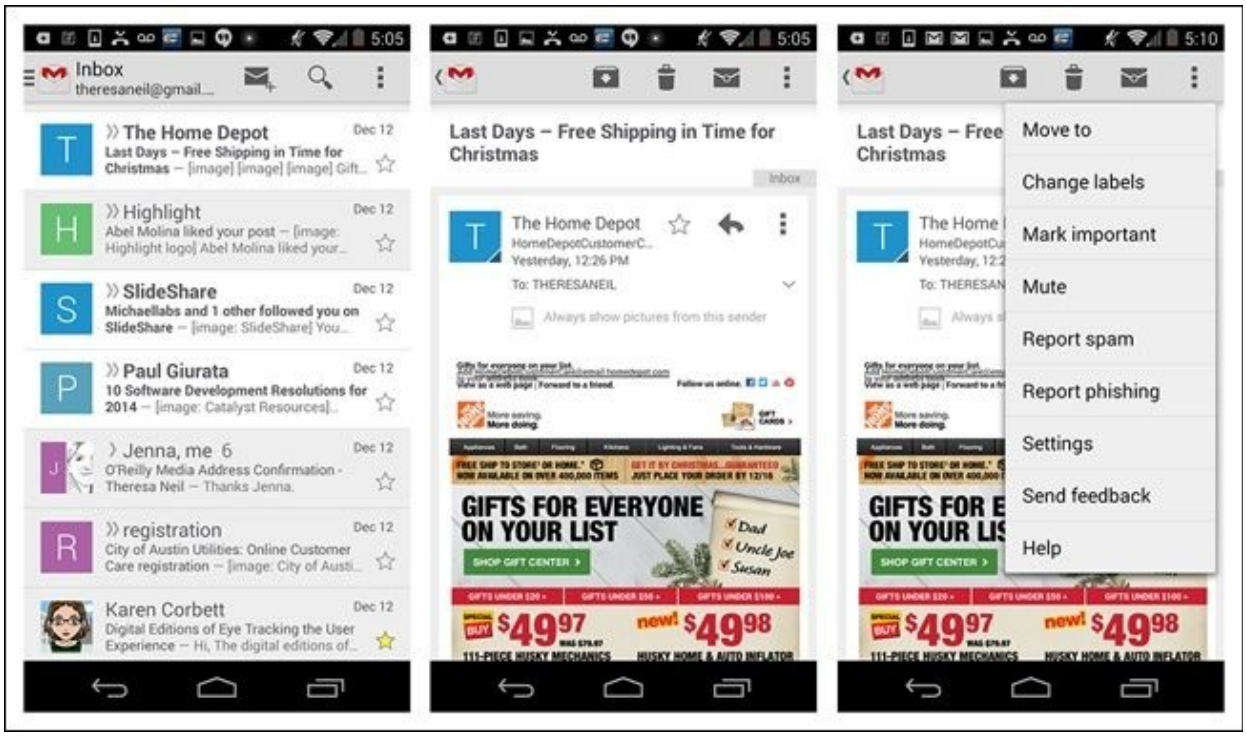



*Figure 5-6. Gmail for Android: Action Bar buttons are determined by screen context*

One of the nice features of Android is that a long press on an action button brings up a tooltip explaining what the button is for. One of the not-so-nice features is that the Action Bar has to hold any necessary navigation controls (title, drawer icon, up button, spinner, etc.), as well as the action buttons, so it can get crowded very quickly.

## Windows Phone

The Toolbar in Windows Phone apps is called the Application Bar (*http://bit.ly/NoMtPY*). The primary actions should be displayed as labeled icons, and secondary actions (called Menu Items) can be tucked under the ellipsis icon.

I was surprised to see that Amazon is trying to make the Application Bar a primary navigation control (see the Home, Search, and Cart icons at the bottom of every screen). It makes me think that the product design was driven by iOS aficionados, rather than a team familiar with Windows Phone design paradigms.

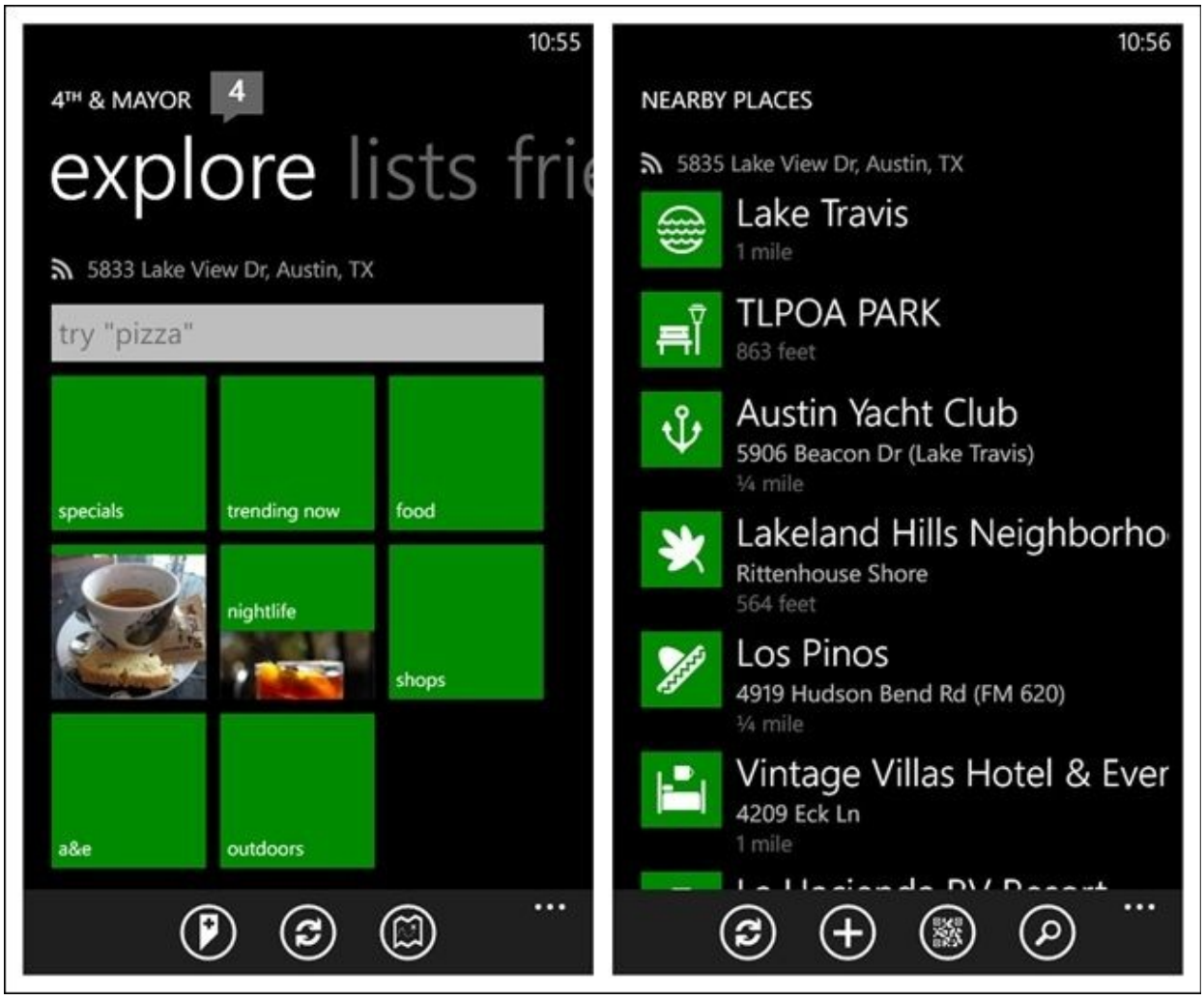



*Figure 5-7. 4th & Mayor for Windows Phone: the Application Bar is for primary actions*

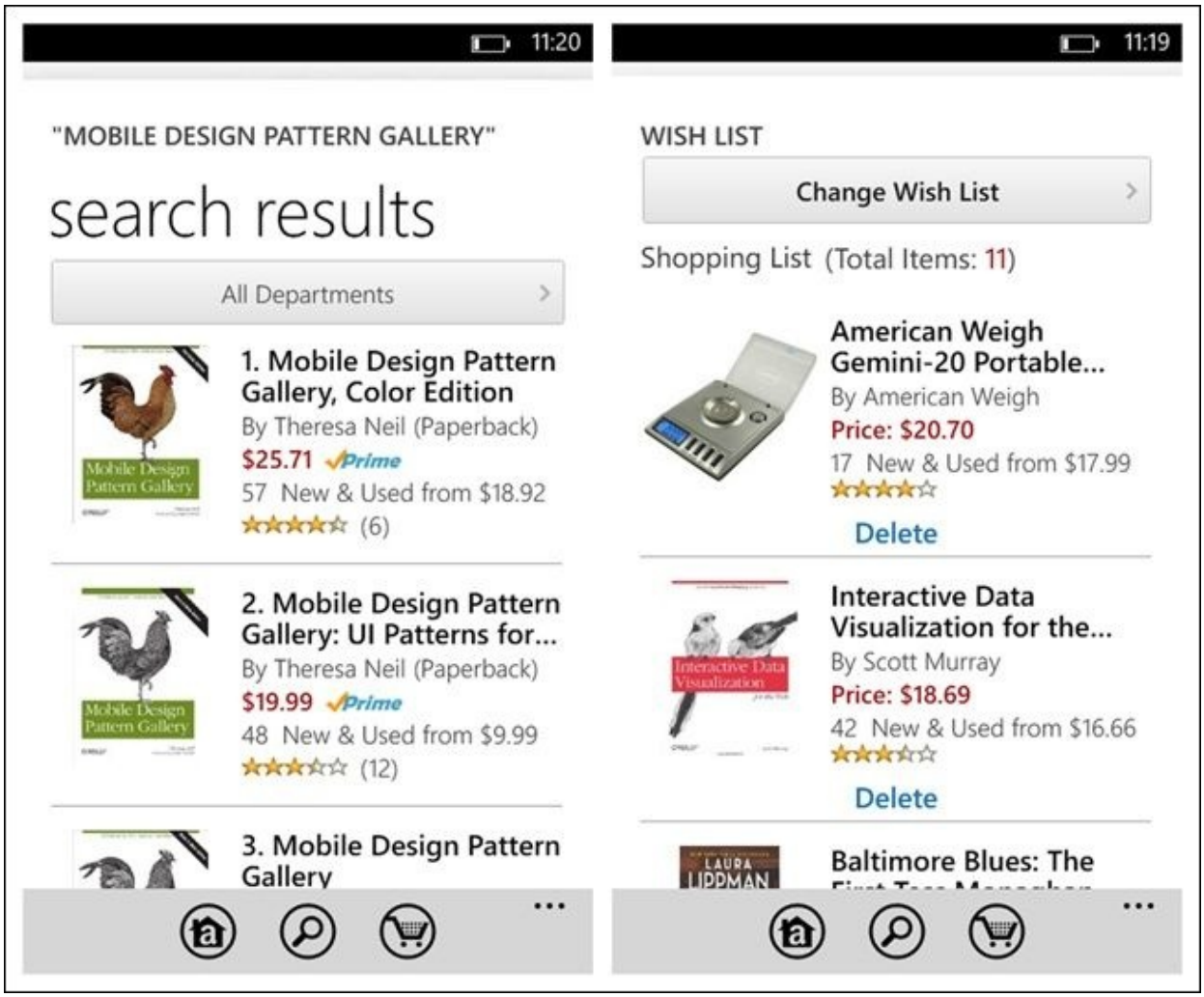



*Figure 5-8. Amazon for Windows Phone: incorrectly using the Application Bar for primary navigation*

Like Android, Windows Phone also offers a way to expose the icon labels. Tapping the ellipsis icon heightens the Application Bar just a bit to reveal the labels below the icons.


*Figure 5-9. Windows Phone Application Bar: tapping the ellipsis reveals labels*

## OS-Neutral Pattern: Contextual Toolbar

A Contextual Toolbar can also serve the action buttons for a text field or text area, like the formatting and attachment options shown in the Twitter, Evernote, and OneNote Toolbars.

> **NOTE**
>
> Follow OS-specific recommendations for Toolbar placement and behaviors. Choose icons that are familiar and easy to recognize.

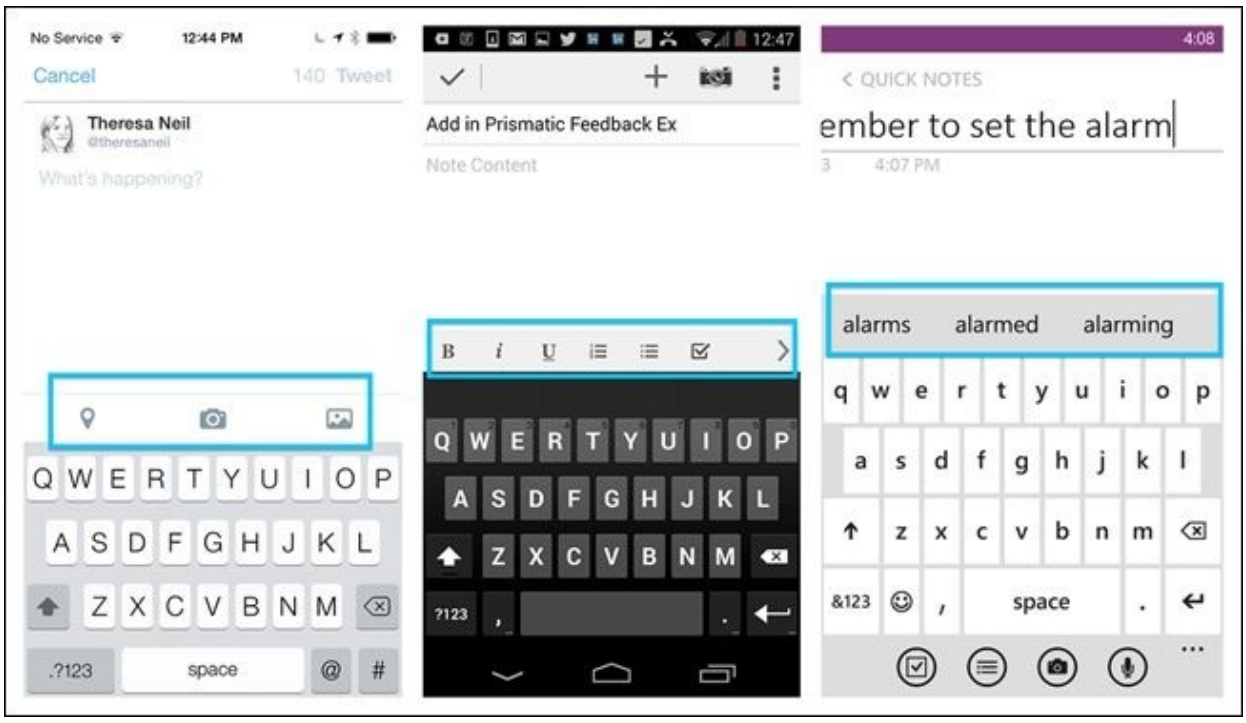


*Figure 5-10. Twitter for iOS, Evernote for Android, and OneNote for Windows Phone: Contextual Toolbars for text entry actions*

## Toolbox

A Toolbox is like a Toolbar on steroids. You'll see this pattern most often in photography and design apps that offer a lot of tools for effects and treatments.

Typically, the main Toolbox is positioned across the bottom of the screen. The options may exceed the screen width and scroll off the screen. Selecting an option from the main Toolbox can reveal a new set of options specific to the selected tool. In Pixlr, for example, selecting Adjustment shows all the possible Adjustment options, like Whiten, Touch Up, Focal Blur, and more.


*Figure 5-11. Pixlr for Android: tool options are shown when a tool is selected from the Toolbox*

Repix and Oggl use the same pattern, but instead of an overlay menu of options, a second row of tools is displayed to show the selected tool's options.


*Figure 5-12. Repix for iOS: tool options appear as a second row above the Toolbox*

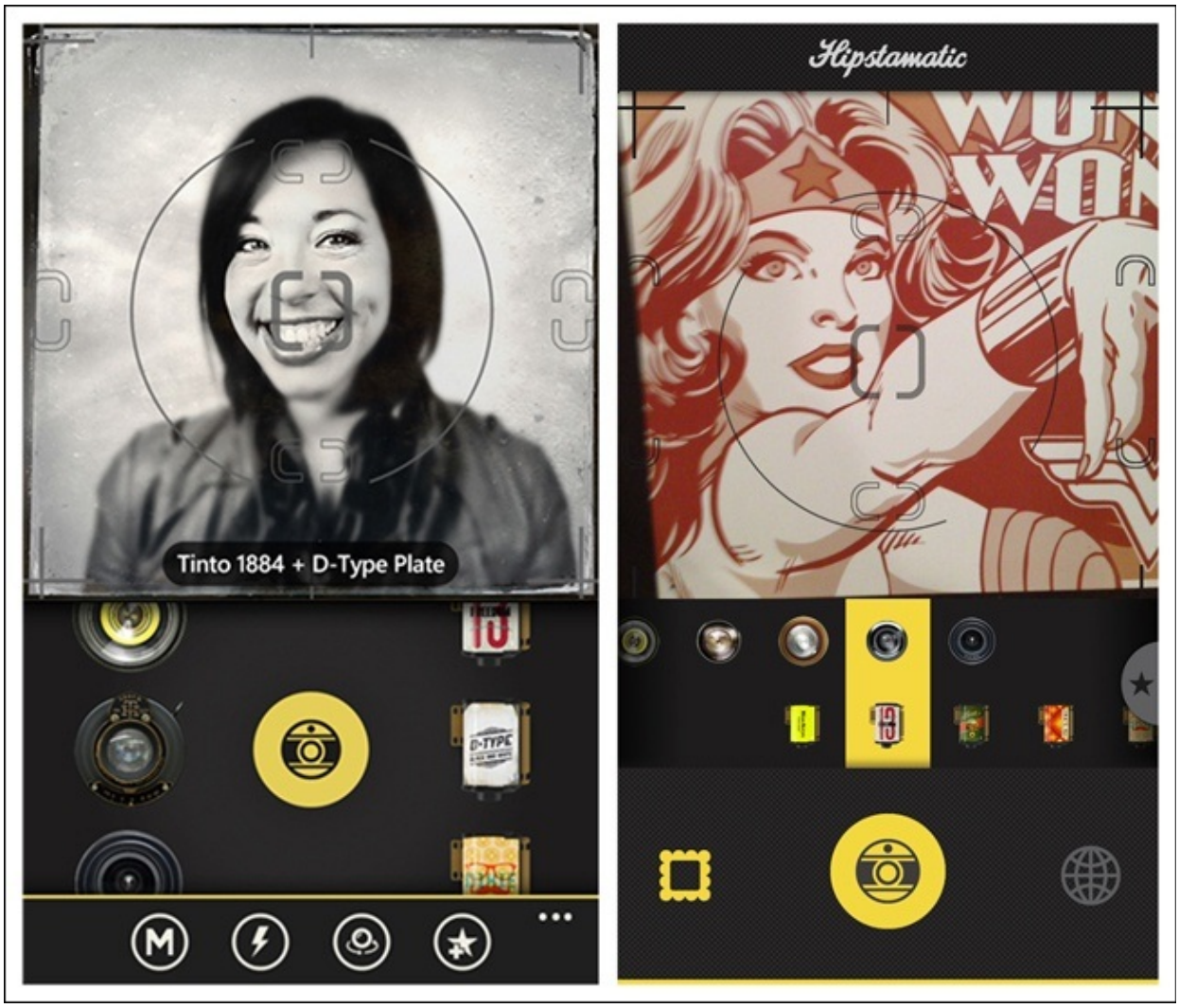



*Figure 5-13. Oggl for Windows Phone and iOS: Toolbox design varies with OS*

Another implementation shows the primary set of tools and, once one is selected, reveals the suboptions for that tool, hiding the primary options. In the example from Aviary, orientation is selected first, and then the orientation options are displayed. If you use this pattern, keep two things in mind: it's best to restate which menu item is in effect since the main Toolbar is now out of sight, and you should offer an escape from this mode, such as a Cancel button.

**NOTE**

Labels plus icons improve the usability of Toolbars and Toolboxes. If tool options hide the main Toolbox, use some means (like a screen label) to let users know where they are, and be sure to include a Cancel button.

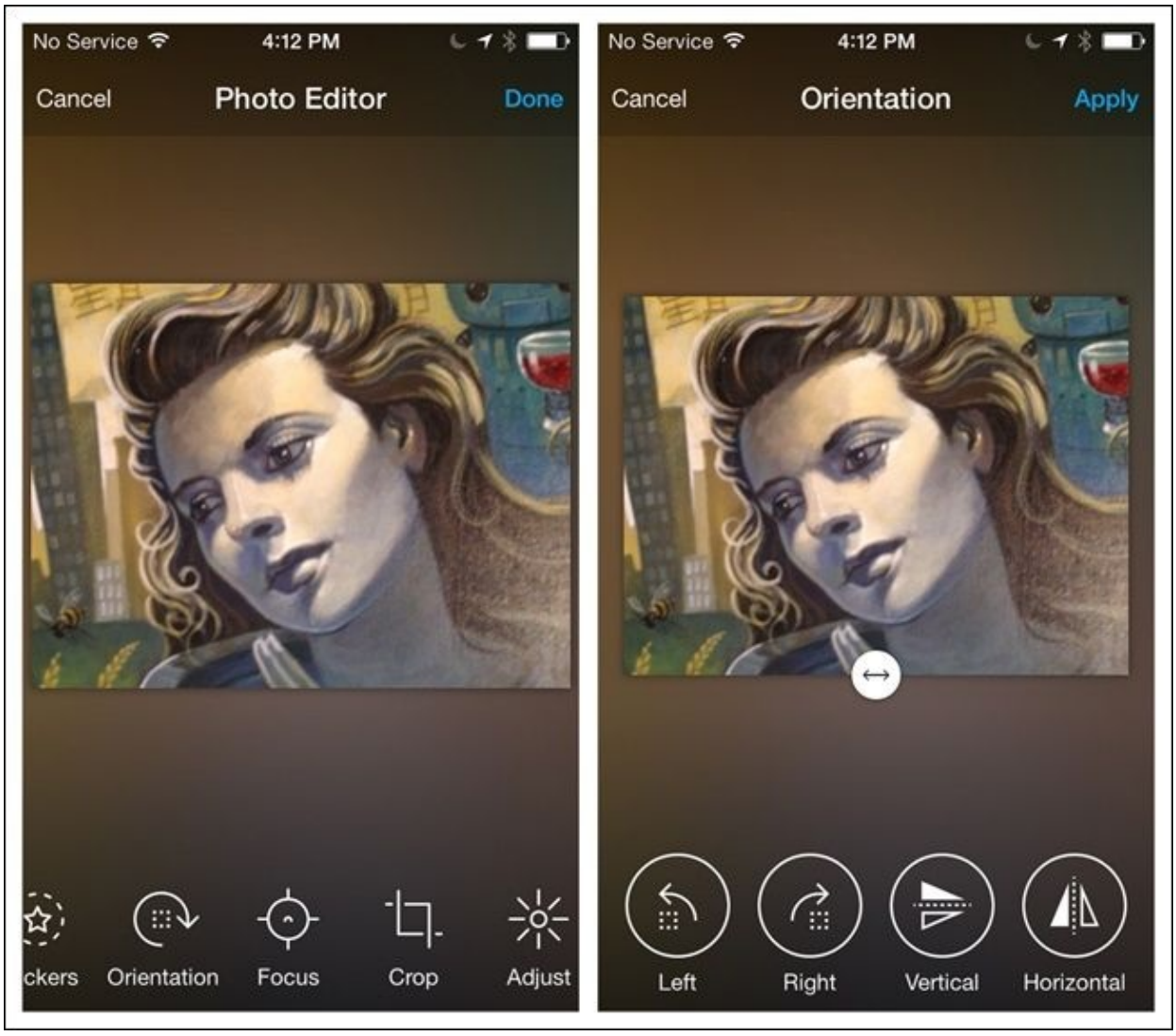



*Figure 5-14. Aviary for iOS: options for selected tool (orientation) are shown on right*

Virtual Makeover uses this pattern to present many different makeup tools. Its pie-style version of the Toolbox is surprisingly easy to navigate.

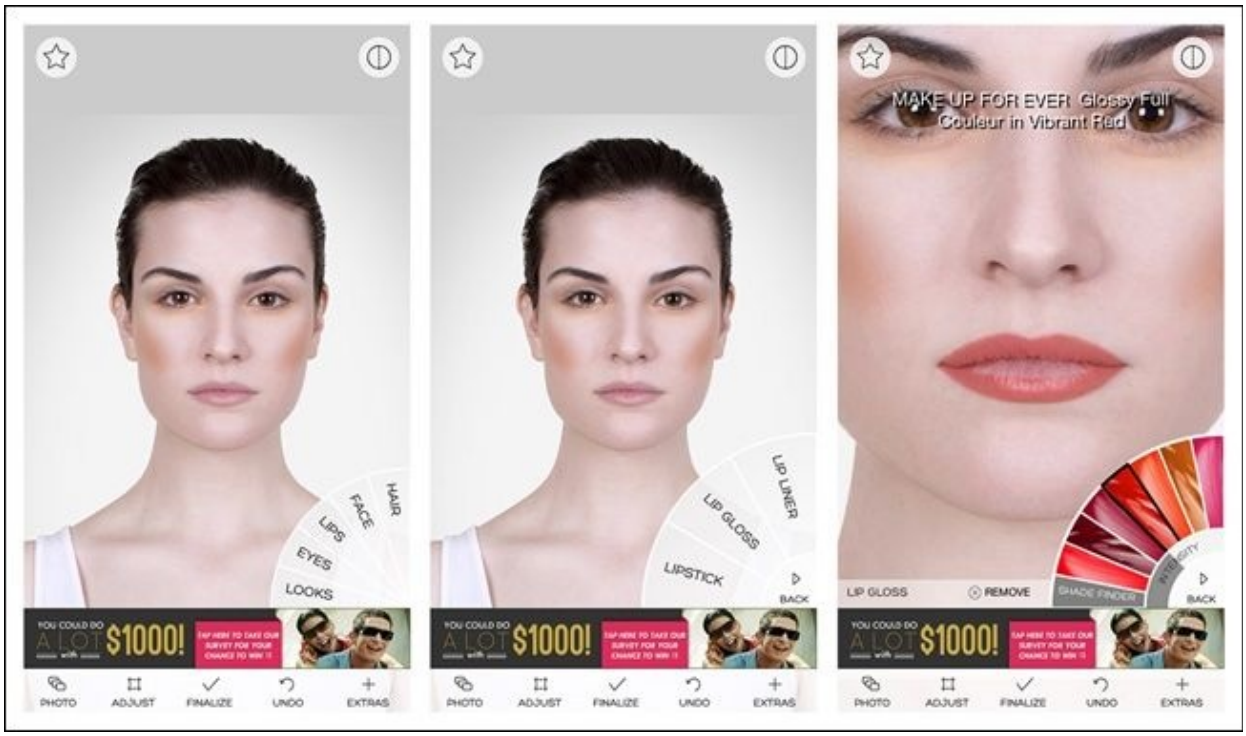



*Figure 5-15. Virtual Makeover for iOS: pie-style Toolbox*

## Call to Action Button

Sometimes a Call to Action (CTA) Button may be a better option than a Toolbar when you are presenting a single action on the screen. Frank & Oak, Etsy, and Hipmunk are examples.

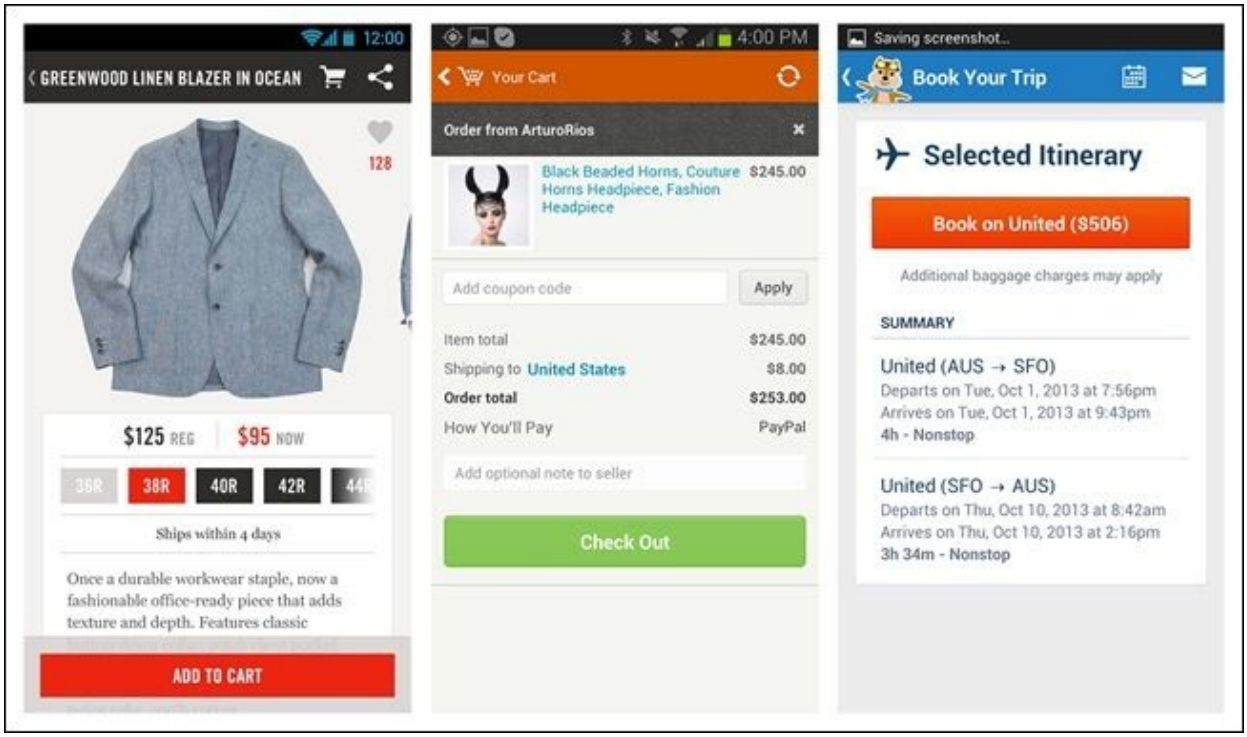



*Figure 5-16. Frank & Oak, Etsy, and Hipmunk for Android: prominent CTA Buttons*

This pattern may also work for screens with one primary and one ancillary CTA, as in Foursquare and Fancy. Just make sure to differentiate the primary CTA visually from the other buttons.

In those examples, the CTAs are screen-specific and, more importantly, flow-specific. Over the years, designers have struggled with where and how to position a global CTA Button; that is, one that is always accessible throughout the app. In Chapter 1, we looked at variations of the standard Tab Bar that included a single Action Button in the middle of the bar, like Instagram and RunKeeperPro.

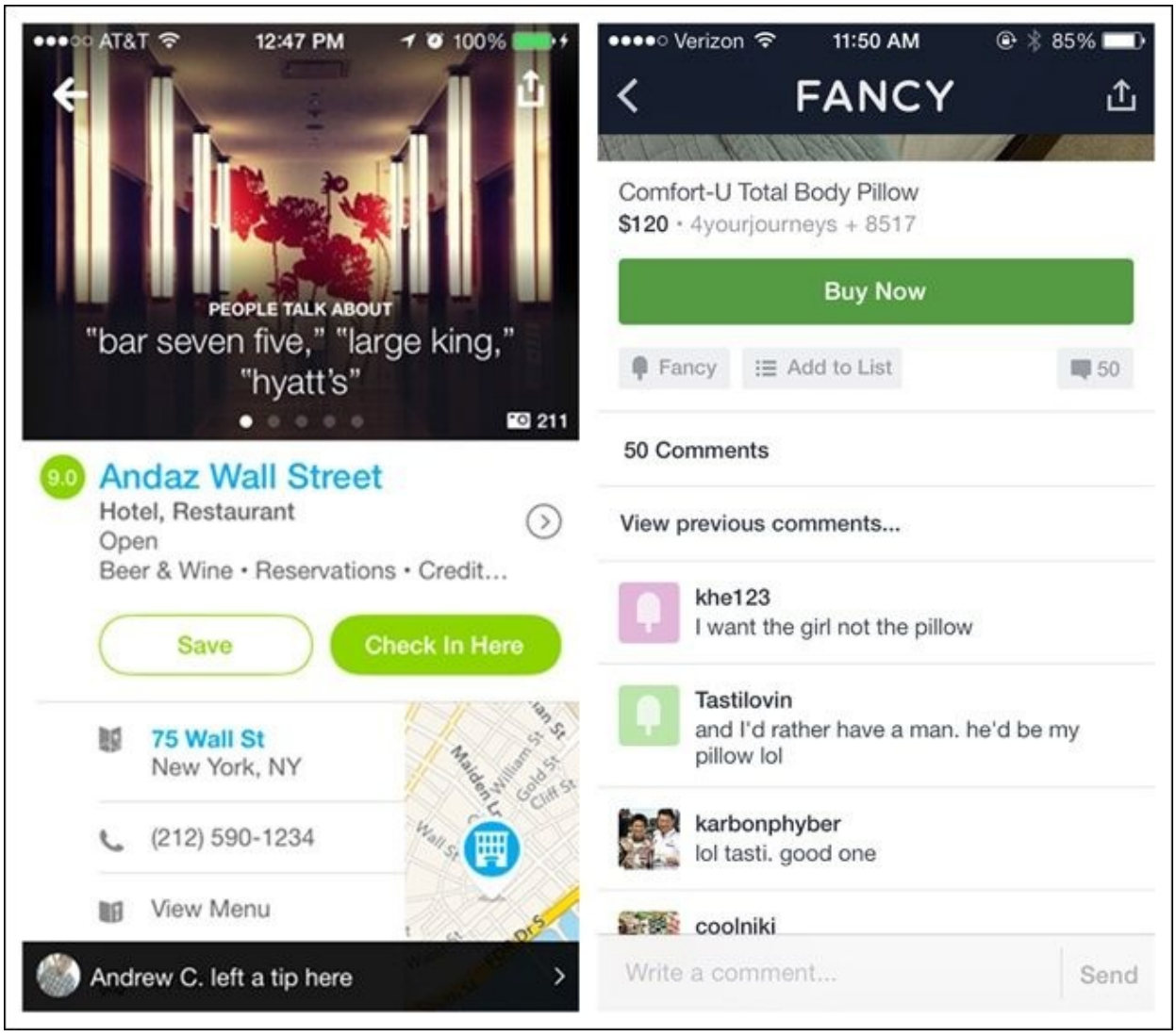



*Figure 5-17. Foursquare and Fancy for iOS: primary CTA is visually different from secondary CTAs*

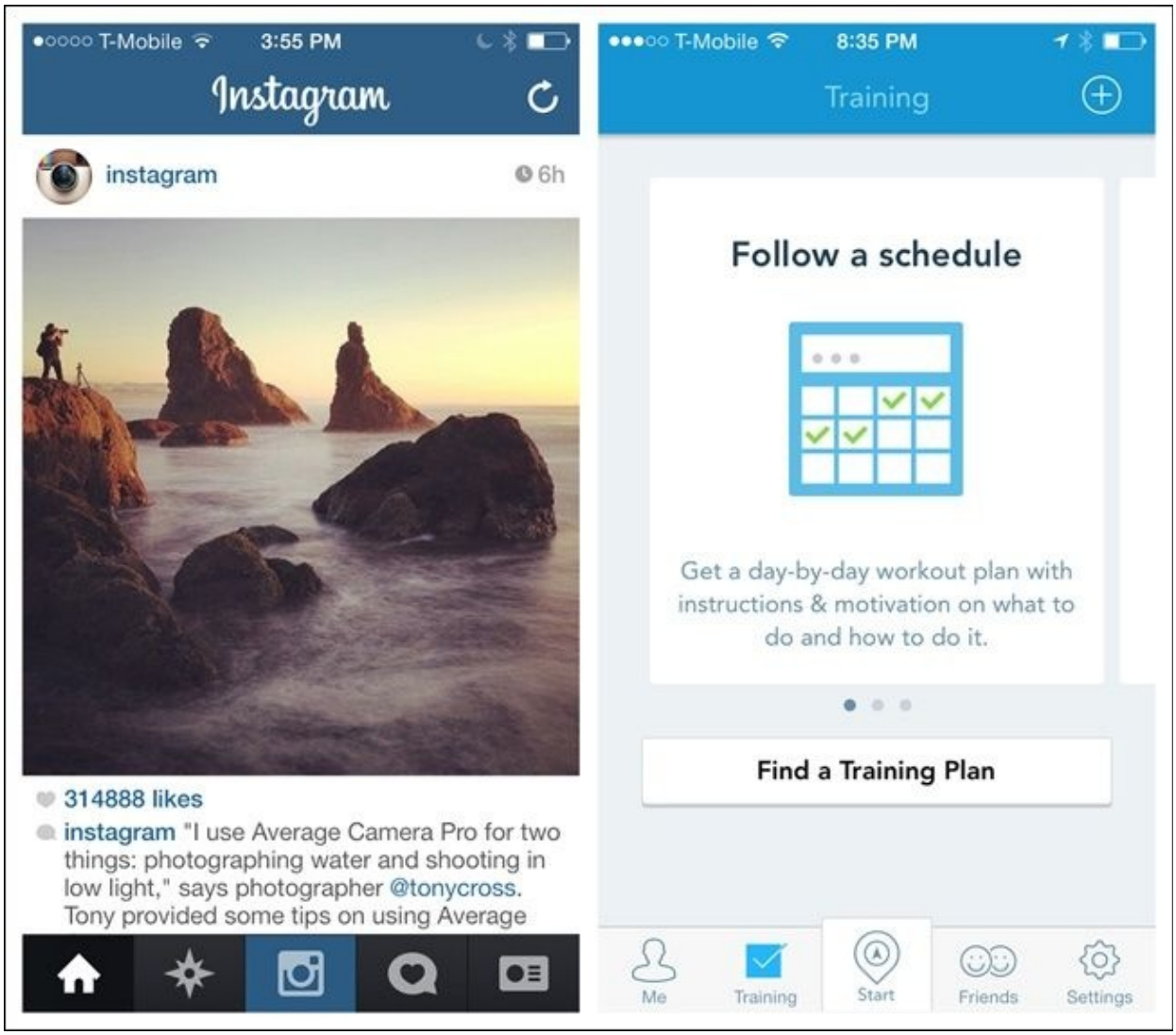



*Figure 5-18. Instagram and RunKeeperPro for iOS: global CTAs in Tab Bar*

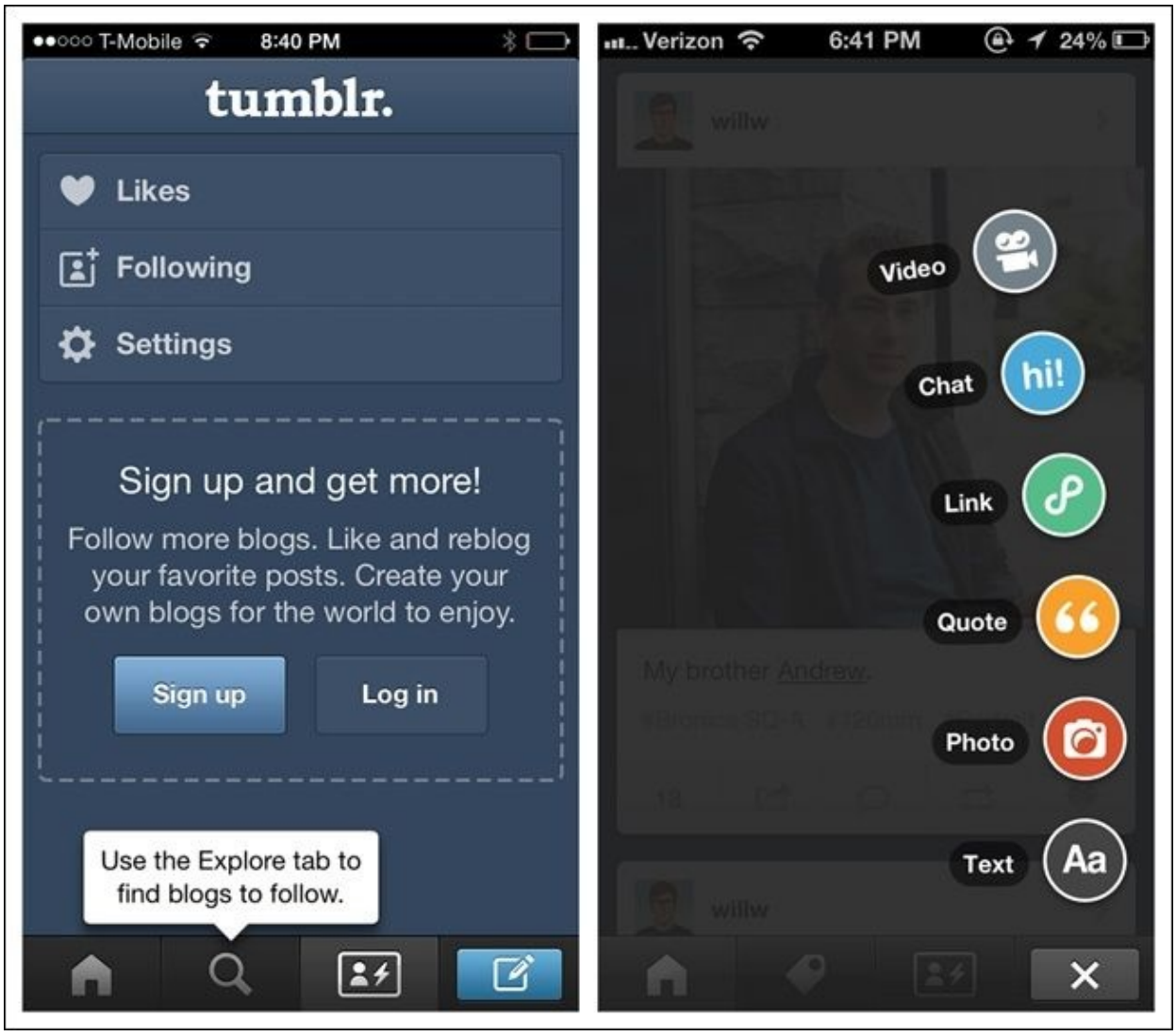



*Figure 5-19. Tumblr for iOS 6: unbalanced right-side global CTA Button with flyout options*

The Tumblr app for iOS 6 revealed a cascading menu of options from the global CTA Button. However, in the iOS 7 design, the global CTA Button opens a full screen of options.

One popular trend is a floating CTA Button positioned at the bottom of the screen. Foursquare, Sphere, and Rove's floating CTA Buttons are examples.

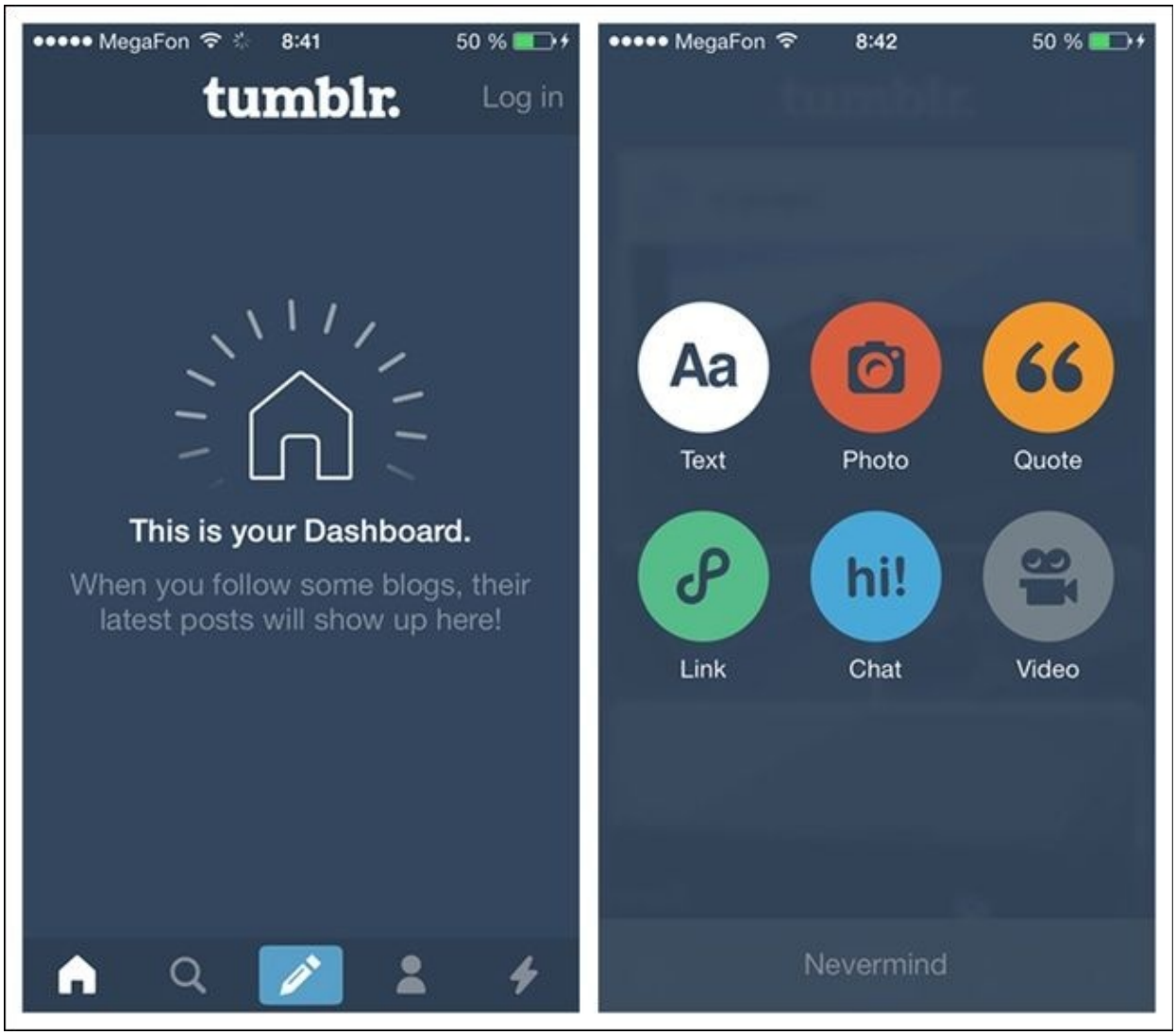



*Figure 5-20. Tumblr for iOS 7 centers the global CTA Button, and presents a full screen of action options*


*Figure 5-21. Foursquare, Sphere, and Rove for iOS: floating CTAs*

Path, Myspace, and QuickBooks reveal additional options when the floating CTA Button is tapped. In all three, the CTA Button shifts from a + to an x. By the way, this is the proper way to design symmetrical interactions (e.g., tap to open and tap to close in the same location).

**NOTE**

Consider using the CTA Button to help users complete a flow successfully, or quickly engage with applications that have a specific global driving action (like taking photos, sharing, or recording entries).


*Figure 5-22. Path for Android: when tapped, the floating CTA reveals floating options*

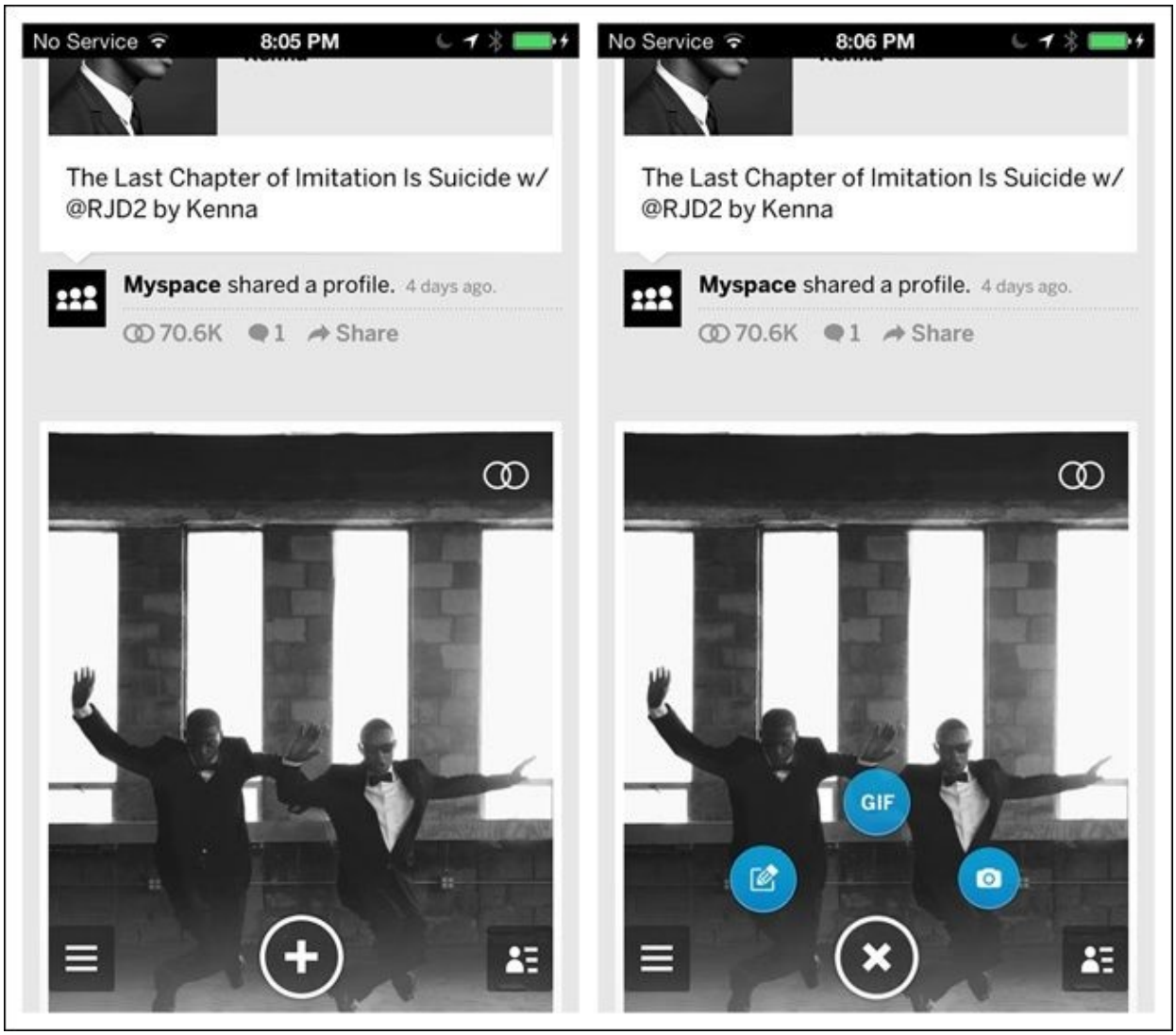



*Figure 5-23. Myspace for iOS: note that the CTA icon changes to an x (close) in its opened state*

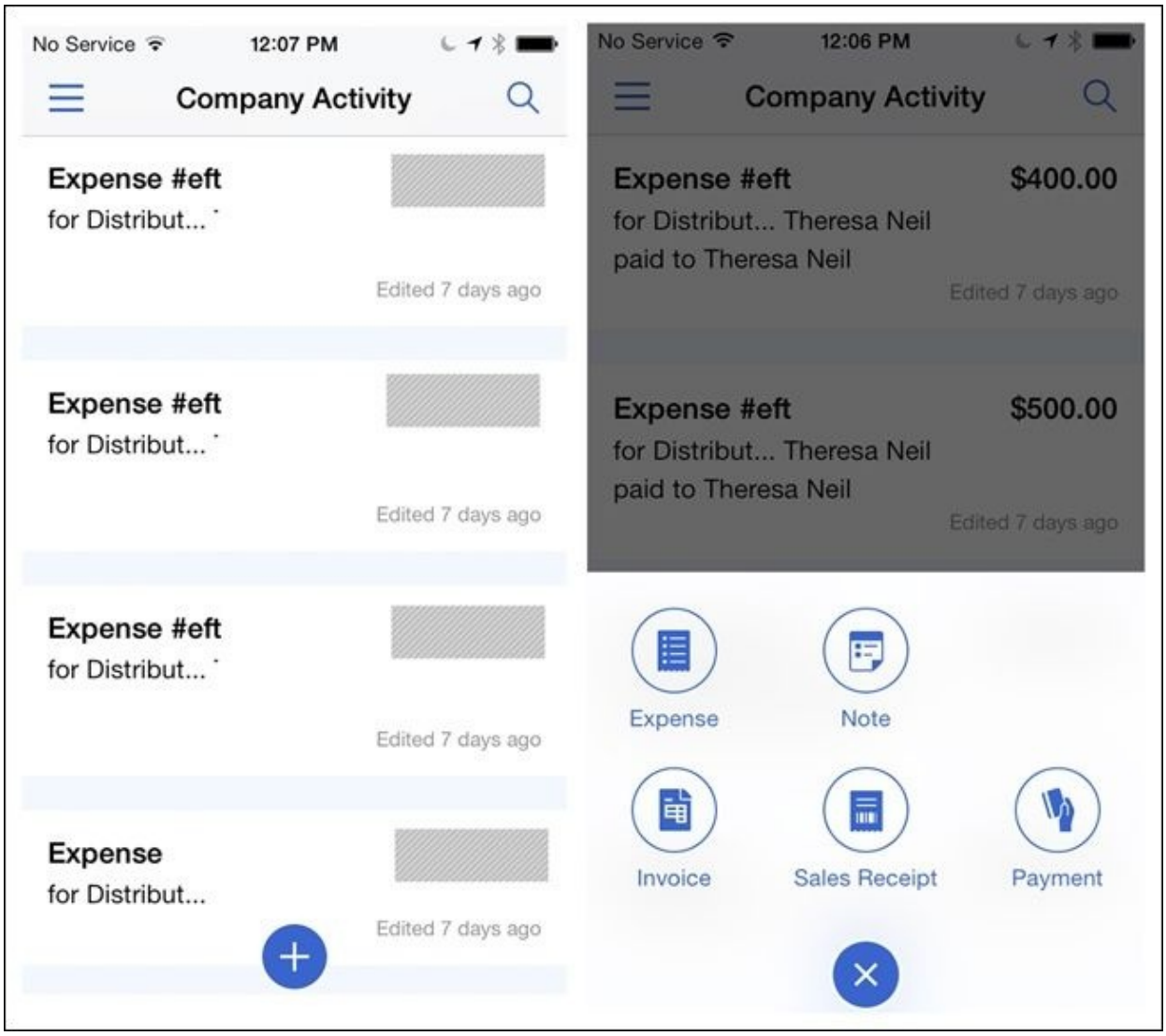



*Figure 5-24. QuickBooks for iOS: another variation of the open/close floating CTA*

# Inline Actions

Inline Actions refer to action buttons that are inline with the object they affect, as opposed to being at a global or screen level. In The Nearby and Foodspotting, for example, each item in the list has an Inline Action button that can show two states (follow/following).

Many applications rely on stateful buttons for the Inline Actions, like the star buttons that can be toggled on and off in RetailMeNot and Gmail. See the next pattern, the Multi-State Button, for incorporating more than two states.

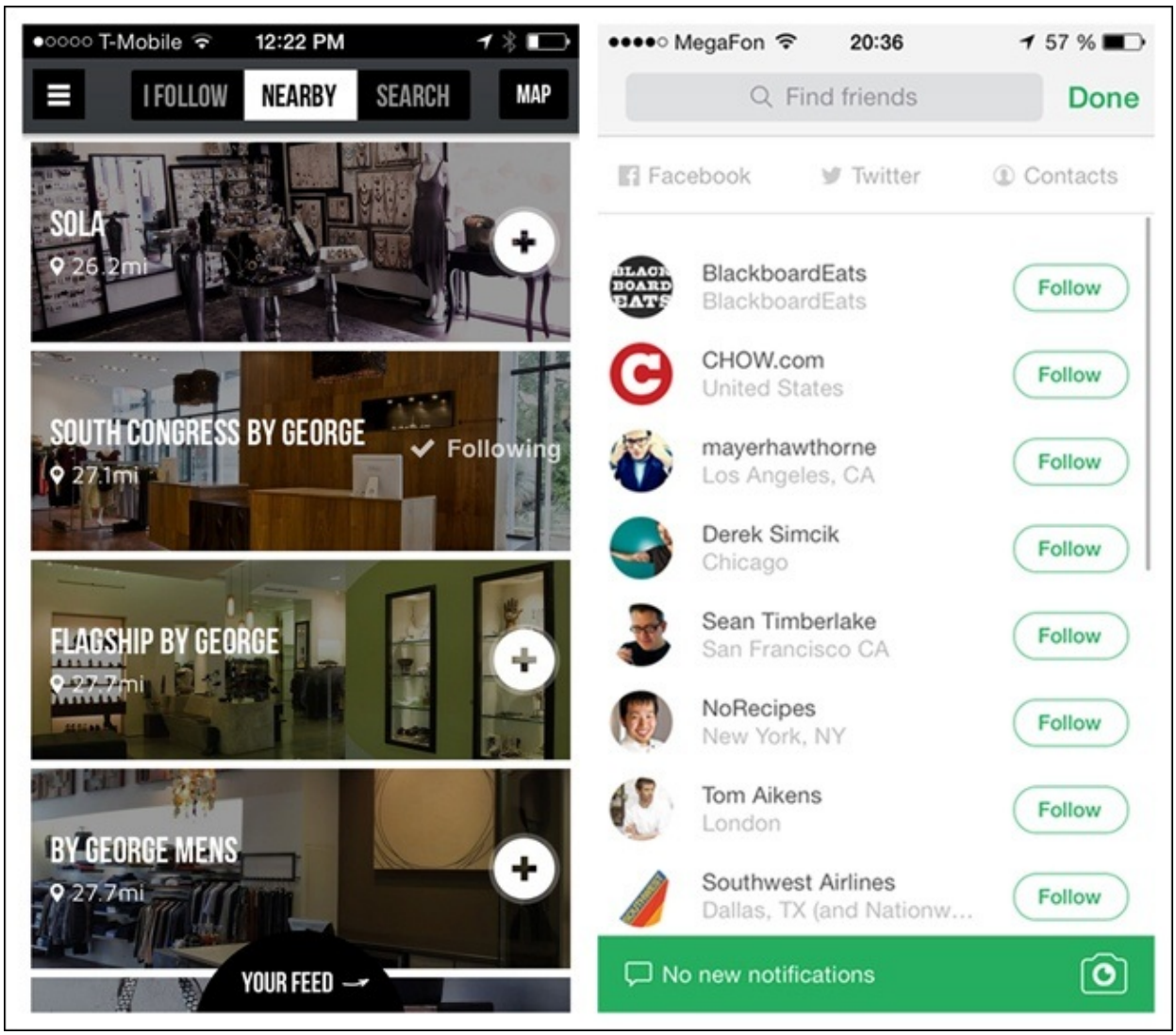



*Figure 5-25. The Nearby and Foodspotting for iOS: Inline Action buttons that can show two states*

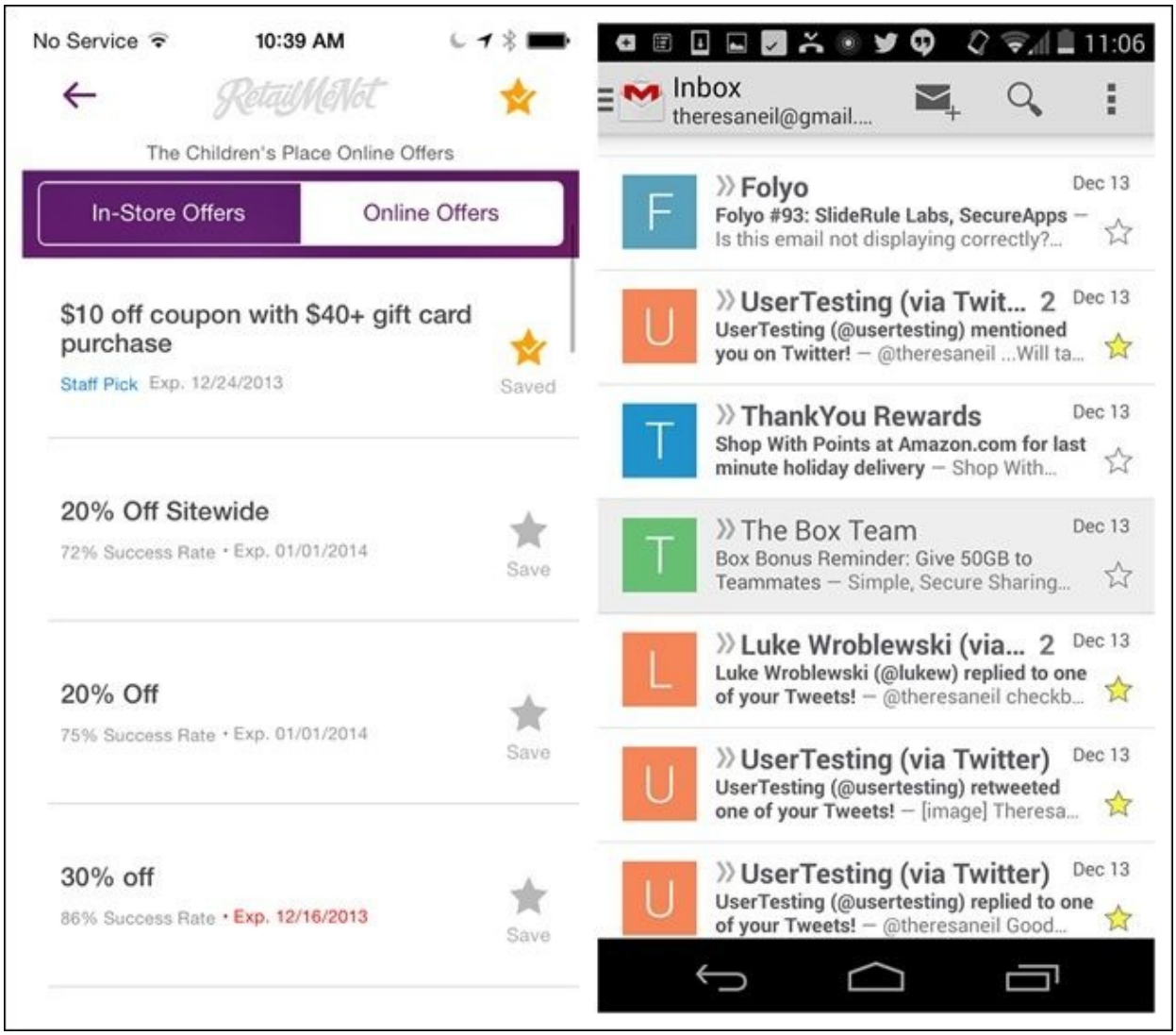



*Figure 5-26. RetailMeNot for iOS and Gmail for Android: stateful star buttons*

The examples from LinkedIn and Facebook have two Inline Actions per row. If you want to display more than two actions per row, consider one of the Reveal patterns discussed later in this chapter.

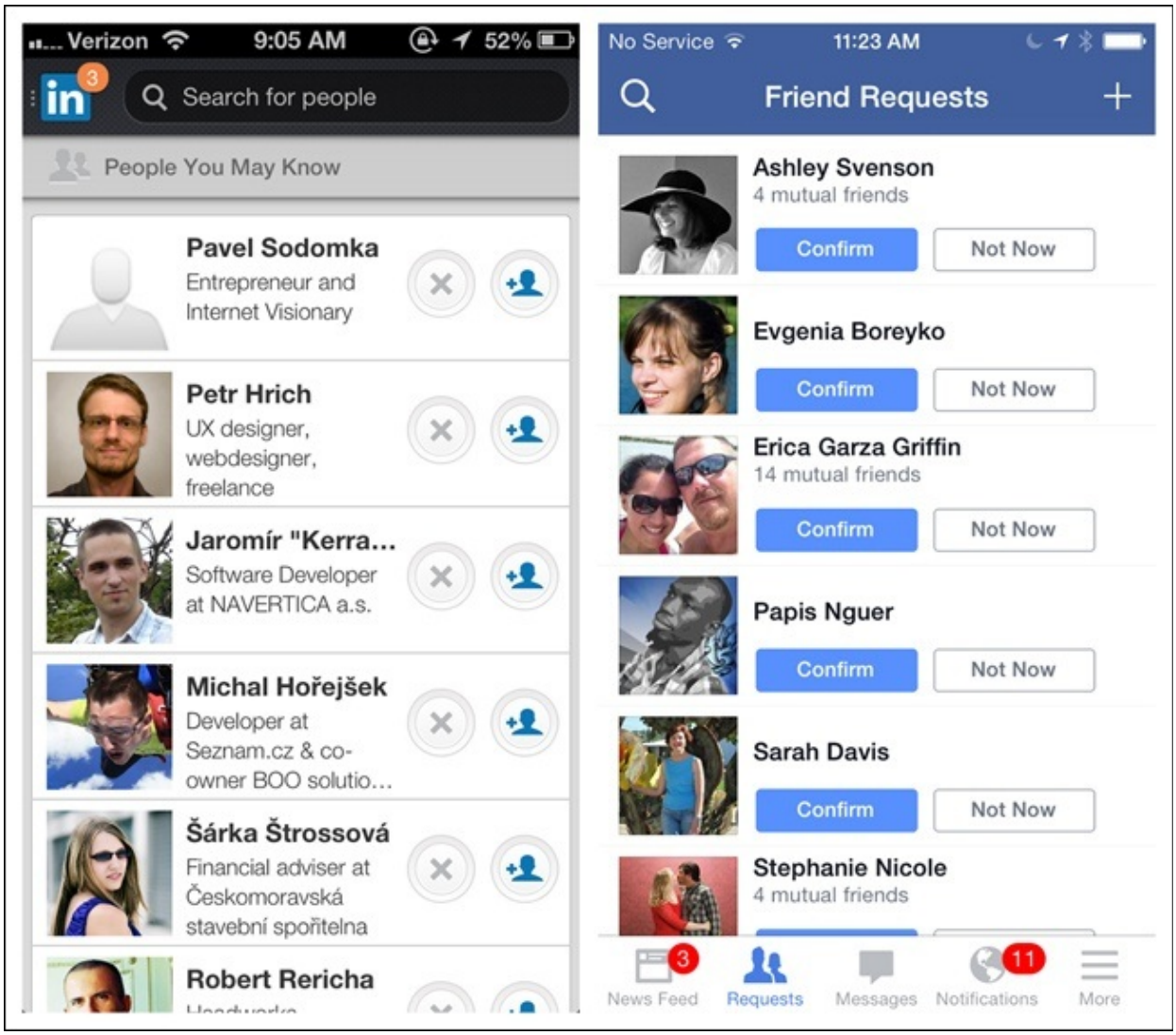



*Figure 5-27. LinkedIn and Facebook for iOS: two Inline Actions per row*

Google Maps, Soundwave, and Pulse provide Inline Actions at the object detail view.

**NOTE**

Inline Actions should be near the actionable object. Choose familiar icons and consider adding a text label.

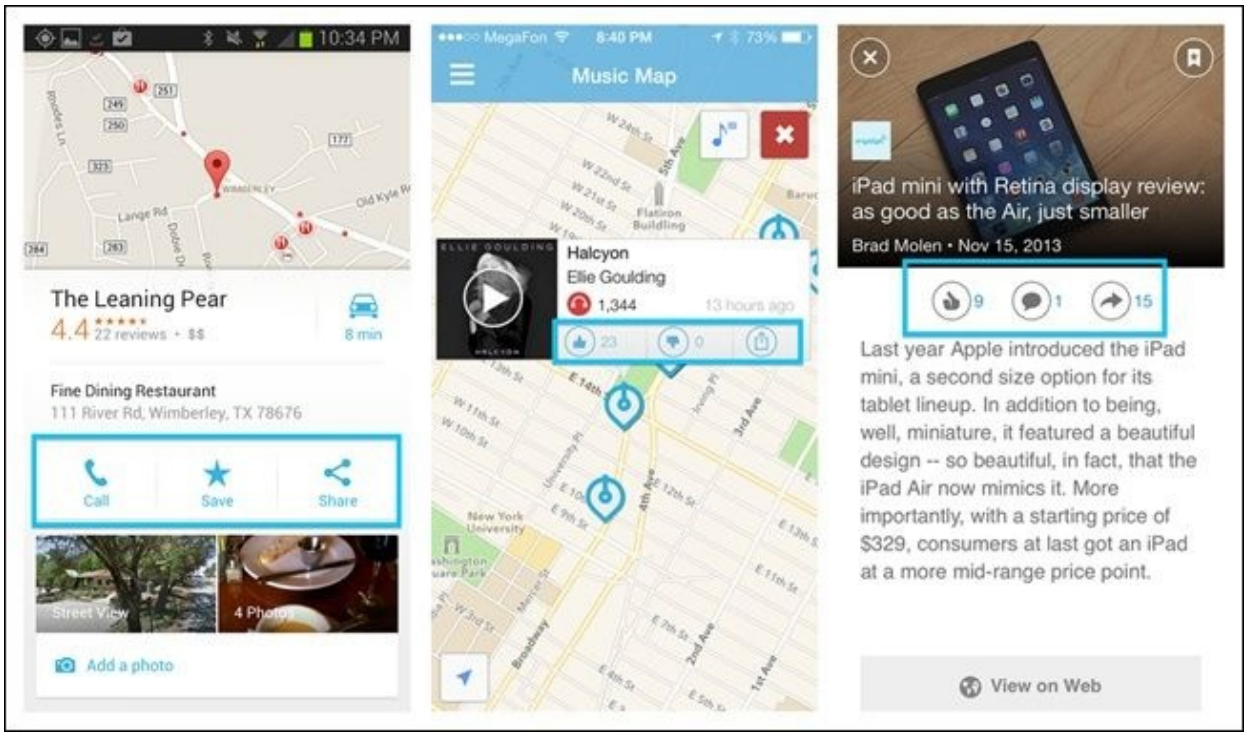



*Figure 5-28. Google Maps for Android, and Soundwave and Pulse for iOS: Inline Actions in object detail views*

## Multi-State Button

Normally, you shouldn't assign multiple actions to a UI control — it can overload the screen and the user. The Multi-State Button is an exception. This type of button acts as both an action trigger and a feedback mechanism.

For example, take the experience of downloading an app in the App Store for iOS. First you'll see the Inline Action to buy. Tap it again to install. Then it becomes an indicator for the install progress, and finally, once the app has downloaded, the button shows the Open option. See Chapter 9 for more examples.


*Figure 5-29. App Store for iOS: Multi-State Buttons show buy, install, progress, and open options*

Another common use case for a Multi-State Button is a delete control. In many cases, popping open a delete confirmation dialog will break the user's flow (see Idiot Boxes in Chapter 11). But it makes sense to prompt for confirmation before completing an irreparable action, so the Multi-State Button serves nicely. (By the way, archiving an email is not an irreparable action, but emptying the trash is.)

**NOTE**

Multi-State Buttons work well for a series of tightly correlated processes that will be performed in rapid succession. Also consider using them instead of a dialog to prompt for confirmation before an irreparable action.

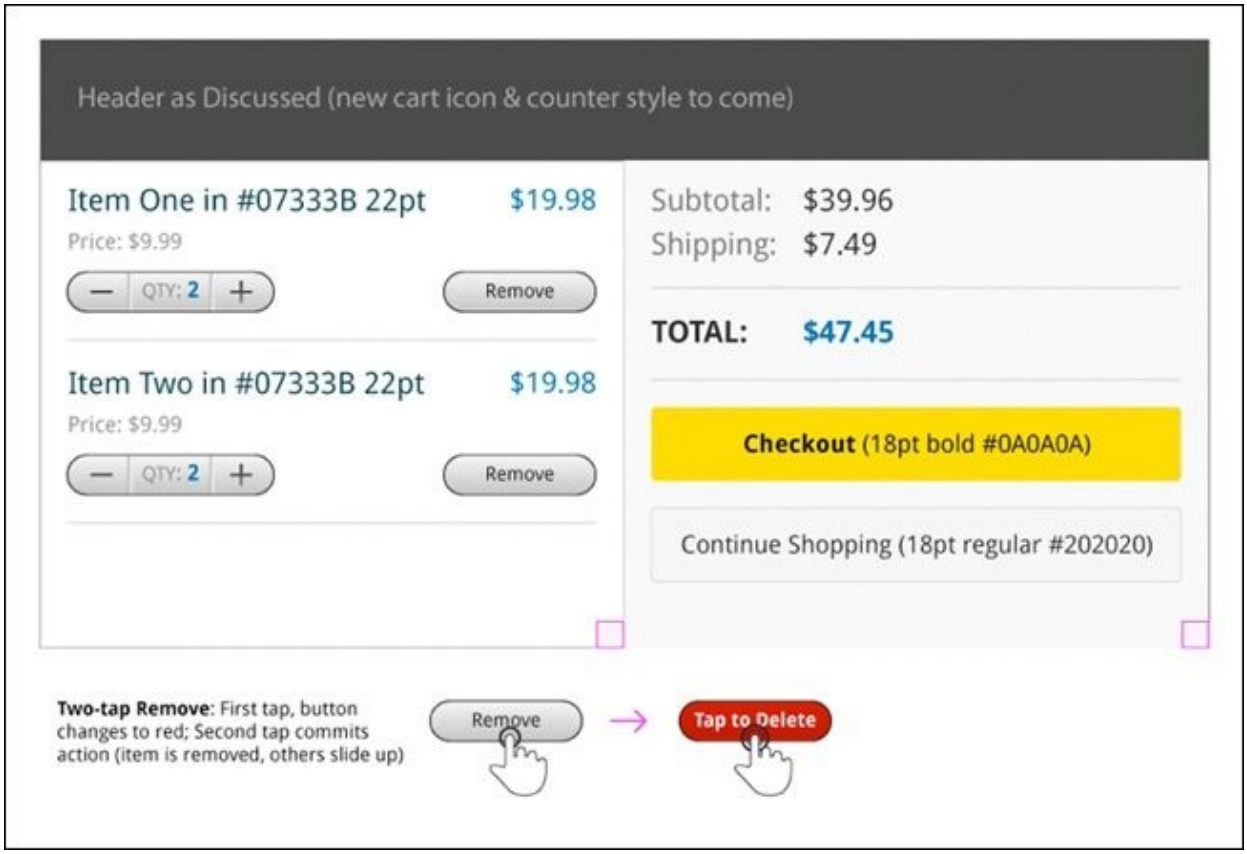



*Figure 5-30. Wireframe and interaction notes from an Android application with a Multi-State Button for delete*

# Contextual Tools

Contextual Tools are specific to an object or task. It makes sense (and declutters the interface) if we reveal these tools only after the context has been established.

For example, tapping the ellipsis button on any of the pictures in Luvocracy reveals the Contextual Tools for the photo. This design is far more discoverable than the long-press-and-slide gesture Pinterest implemented. Of course, Pinterest looks cool with the animation, but watch the video of someone trying to use it for the first time.

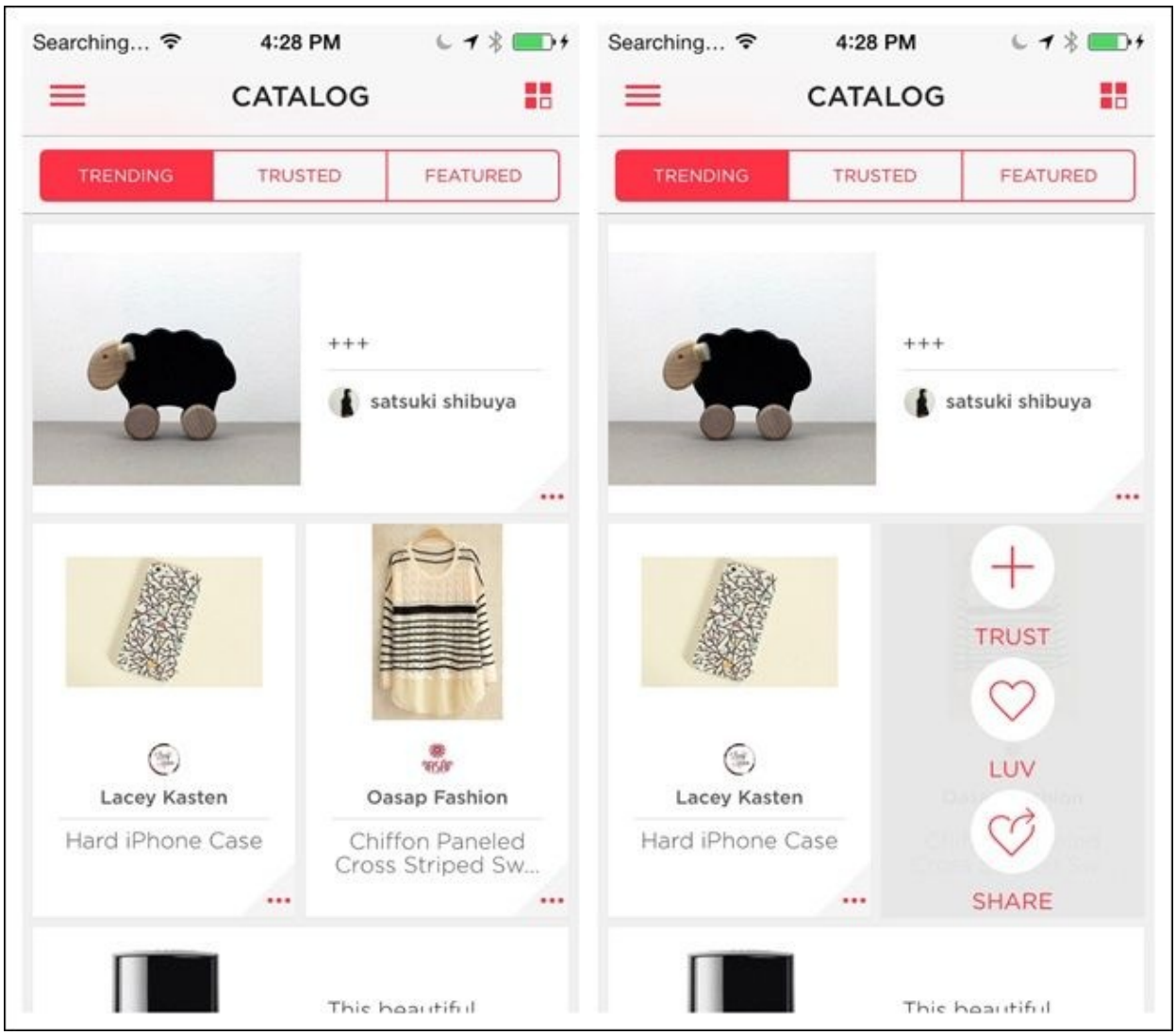



*Figure 5-31. Luvocracy for iOS: tapping ellipsis reveals Contextual Tools*

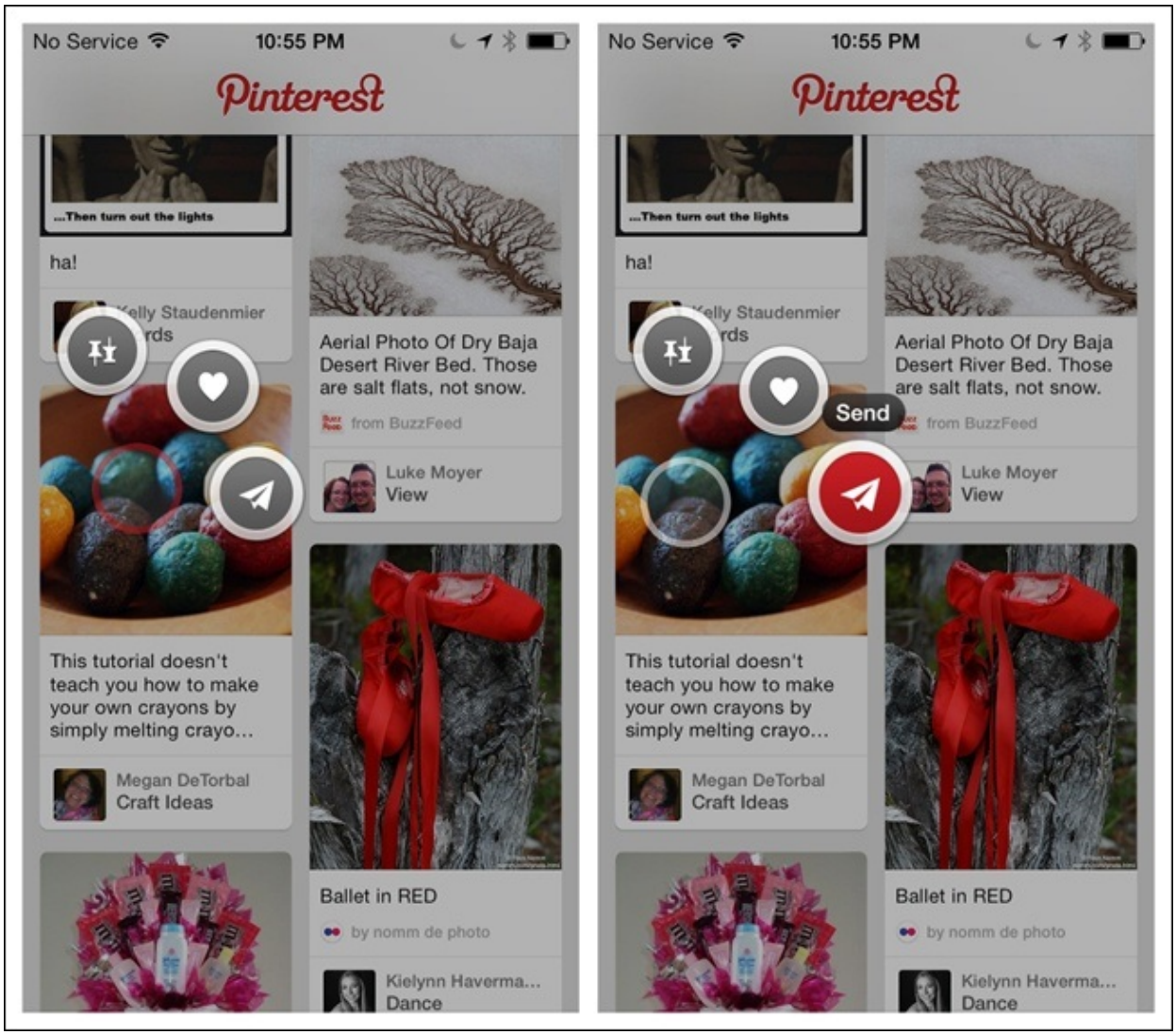



*Figure 5-32. Pinterest for iOS: long-press a Pin, then hold and slide to a button (http://www.youtube.com/watch?v=dvXYztMT-bI)*

Any.do also uses a simple tap to reveal Contextual Tools including set priority, add reminder, and share.

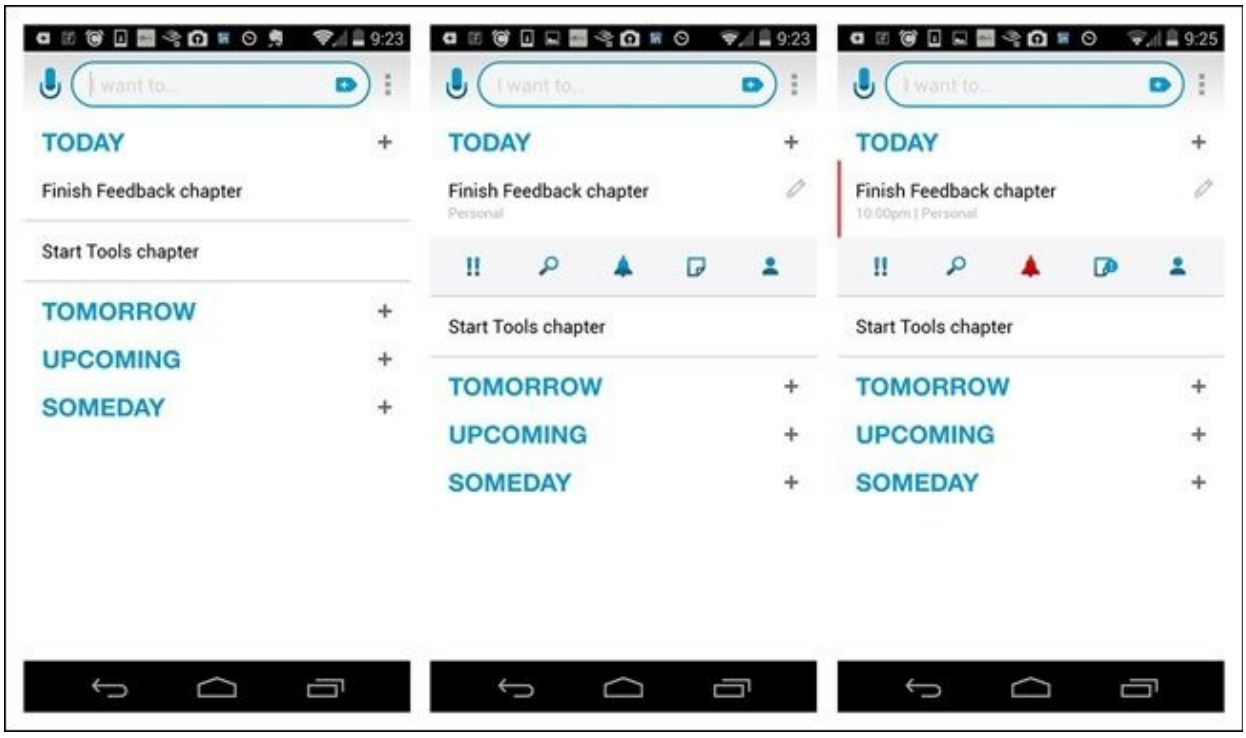



*Figure 5-33. Any.do for Android: tap to reveal Contextual Tools*

What if the rows are already touch targets, like the folders in Dropbox? One idea is to add a “more” icon that, when tapped, will reveal the Contextual Tools. Dropbox uses an overlay to display the options, but you could also go with an inlay that pushes the other rows down.

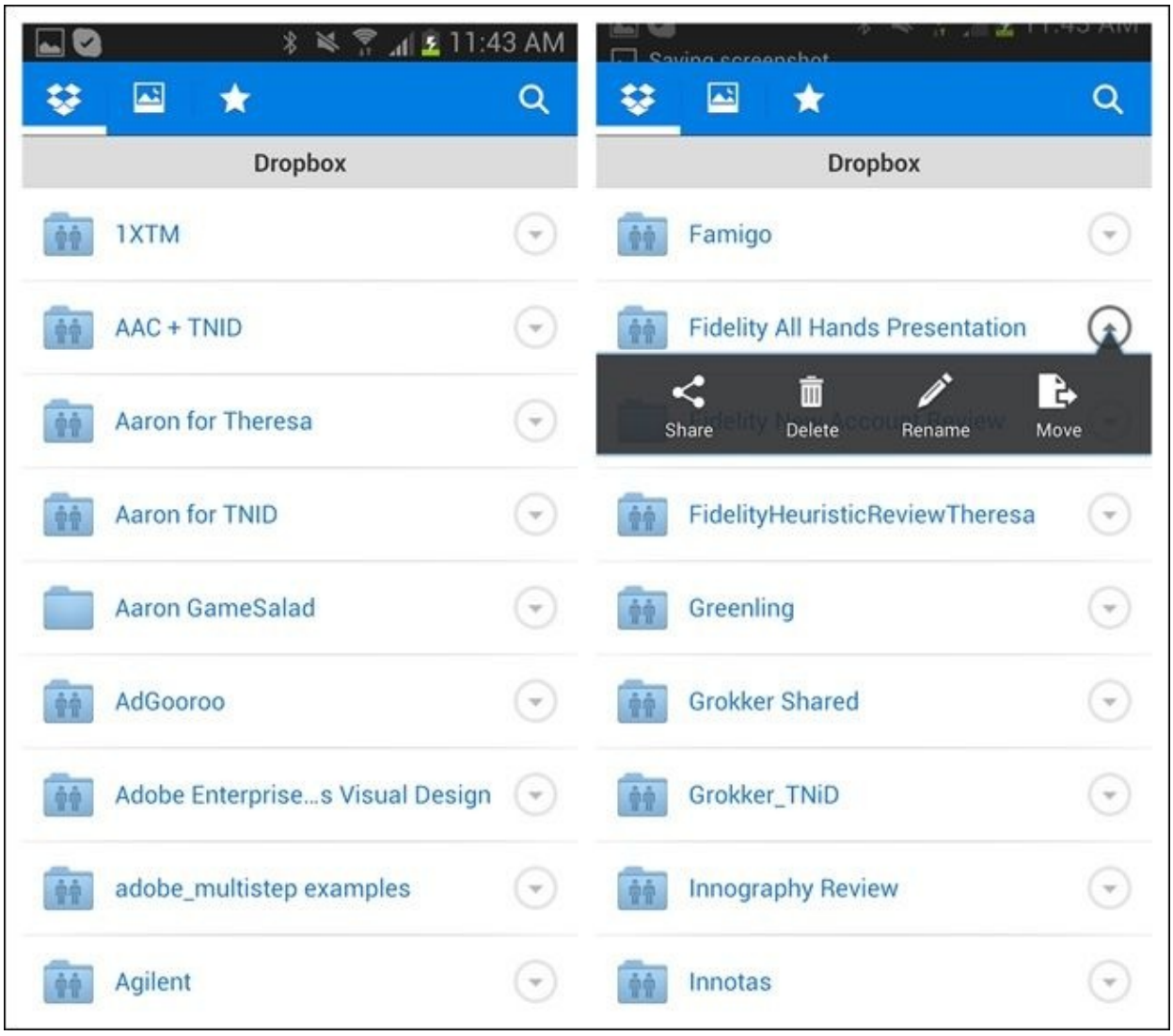



*Figure 5-34. Dropbox for Android: tap the "more" icon to reveal the Contextual Tools overlay*

Another option is to reveal the tools with a gesture other than tapping, like swiping. Flixster and BaconReader reveal Contextual Tools on horizontal swipe.

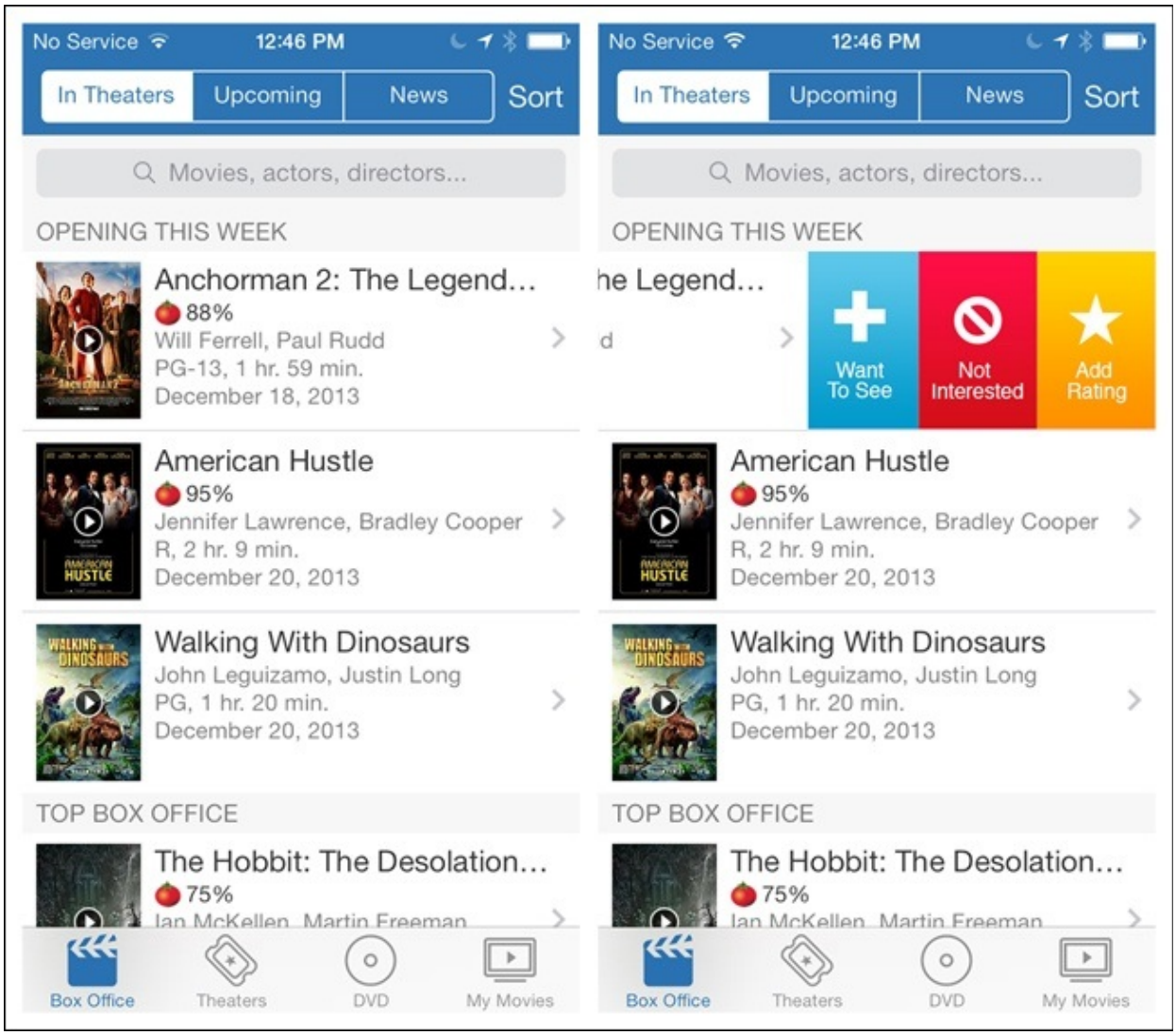



*Figure 5-35. Flixster for iOS: swipe left to reveal Contextual Tools*

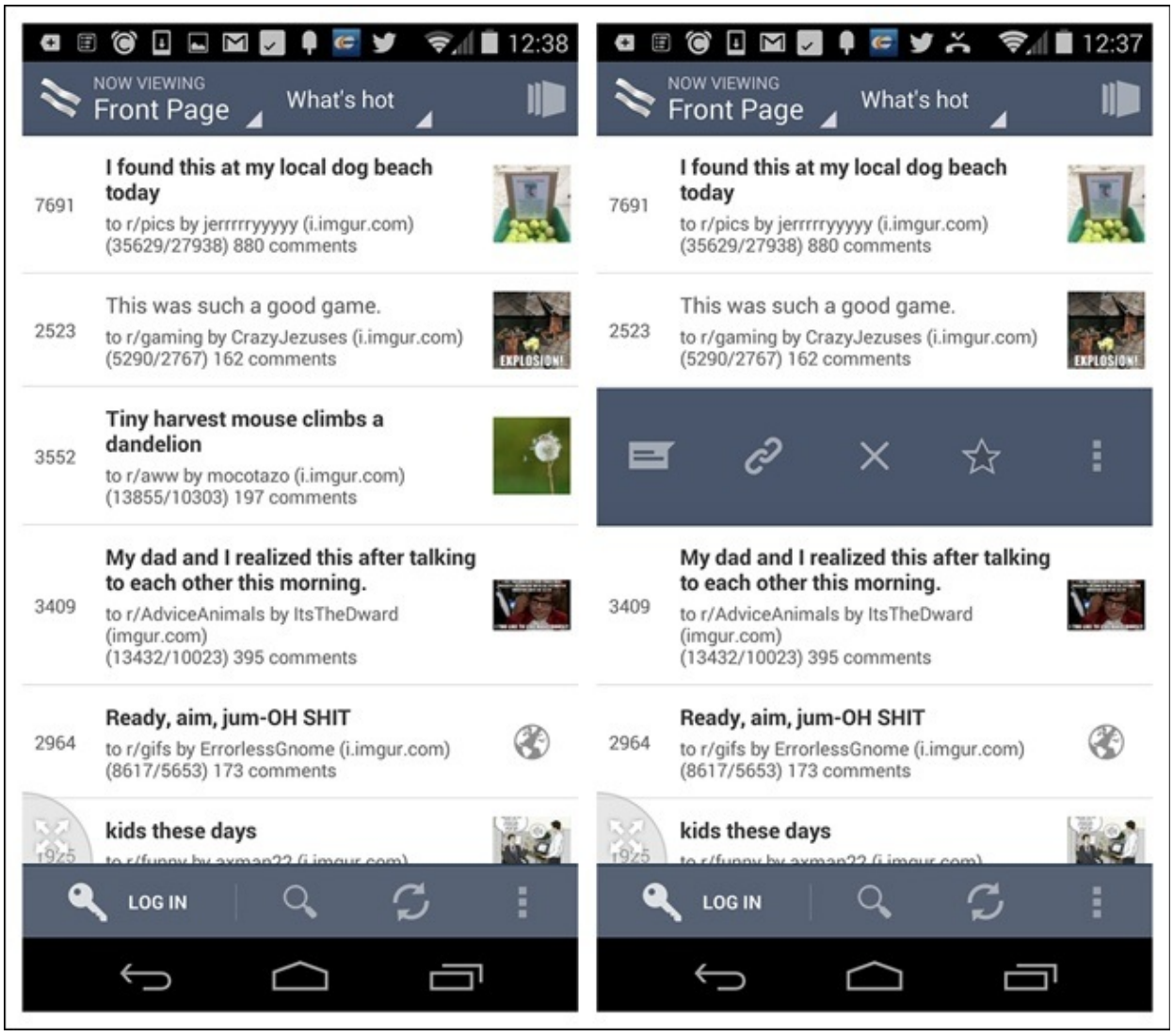



*Figure 5-36. BaconReader for Android: swipe left or right to reveal Contextual Tools*

This approach is very popular in productivity tools, like email and task apps. But beware — it is not as easy as it seems to design an intuitive and efficient gesture-based interface.

A long press can also be used to select an object and present Contextual Tools for it. In Box, long-pressing a file triggers the selection mode with the pressed object already selected. More objects can be selected, and the Contextual Tools (download, share, copy, delete, etc.) replace the Action Buttons. See the next pattern, Bulk Actions, for alternate ways to select multiple objects.

**NOTE**

When you choose a gesture to reveal Contextual Tools, start with the simplest solution first and test for discoverability and user efficiency.

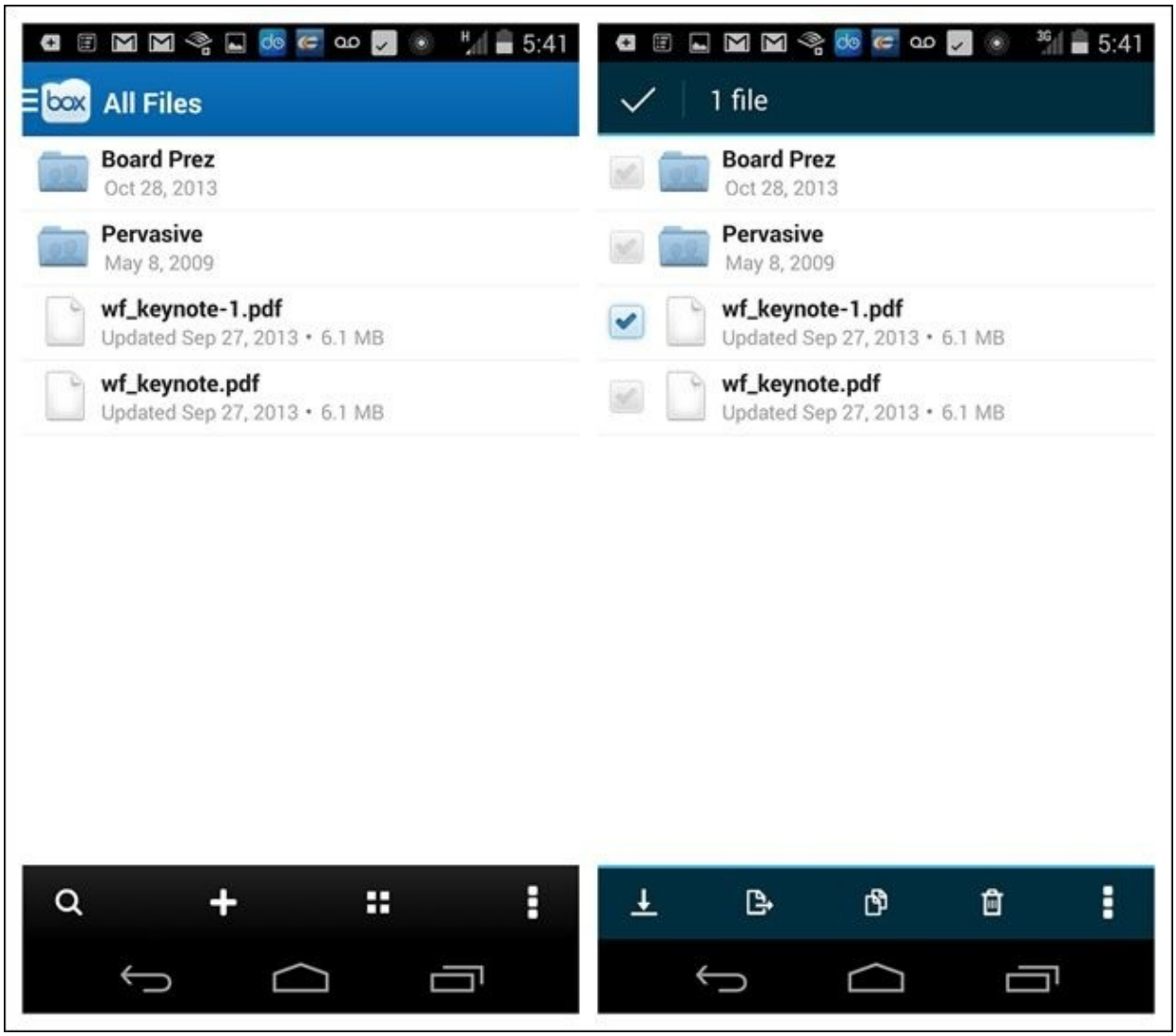



*Figure 5-37. Box for Android: long-pressing a file selects it and presents Contextual Tools; other items can be selected, too*

# Bulk Actions

Common Bulk Actions include selection, deletion, and reordering. Instead of cluttering the main screen with all of these options listed for every item, provide a mode for Bulk Actions.

The native Photos app for iOS offers a “selection” mode for choosing photos from the camera roll to share. Once an item is selected, the Contextual Tools (share and delete) are revealed, and more items can be selected.

The iOS Stocks app offers an edit mode for reordering, deleting, and sorting stocks. Per iOS design standards, tapping Done in the top-right corner saves the changes and exits edit mode.

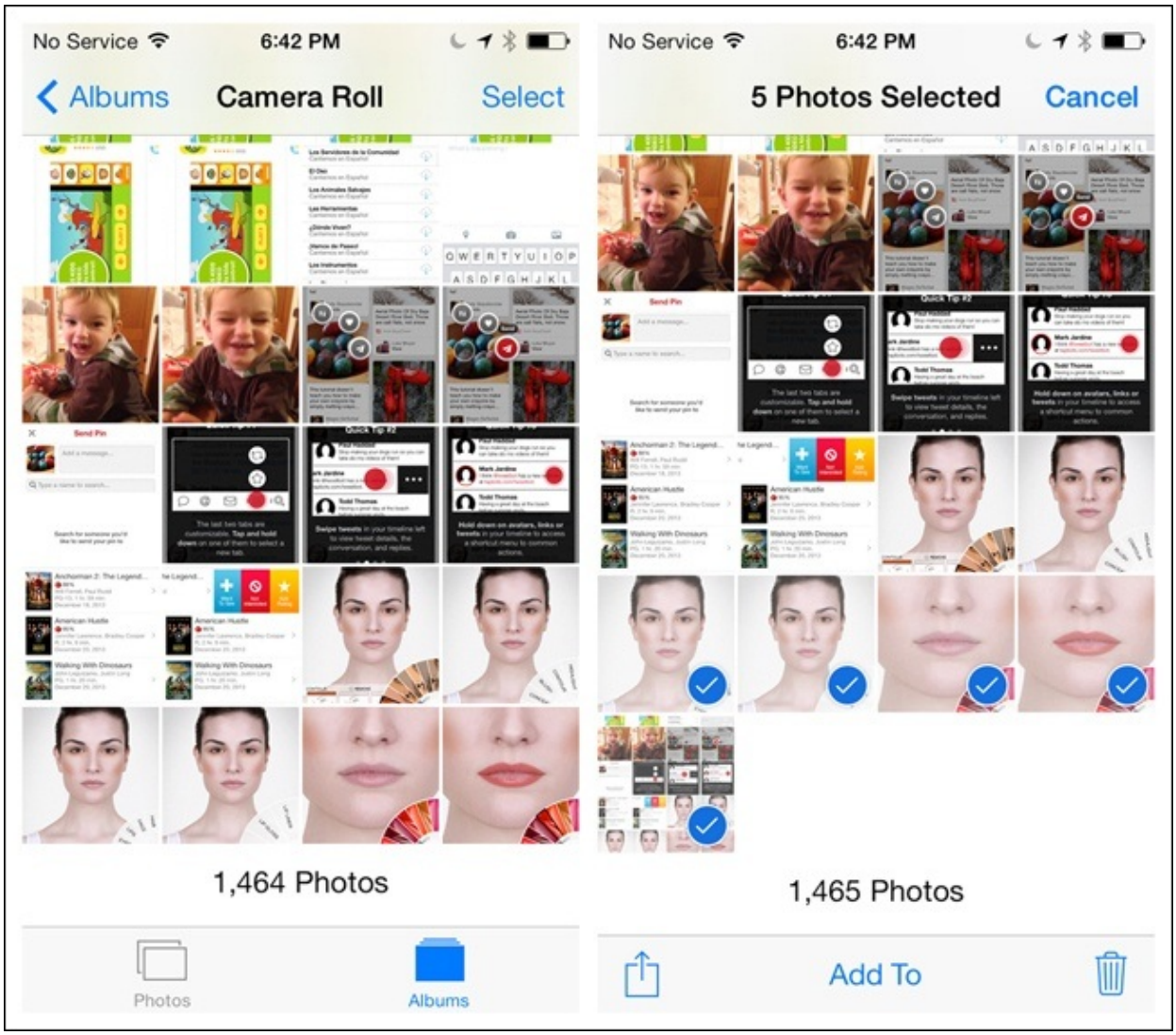



*Figure 5-38. Photos for iOS: Contextual Tools enable Bulk Actions for all selected items*

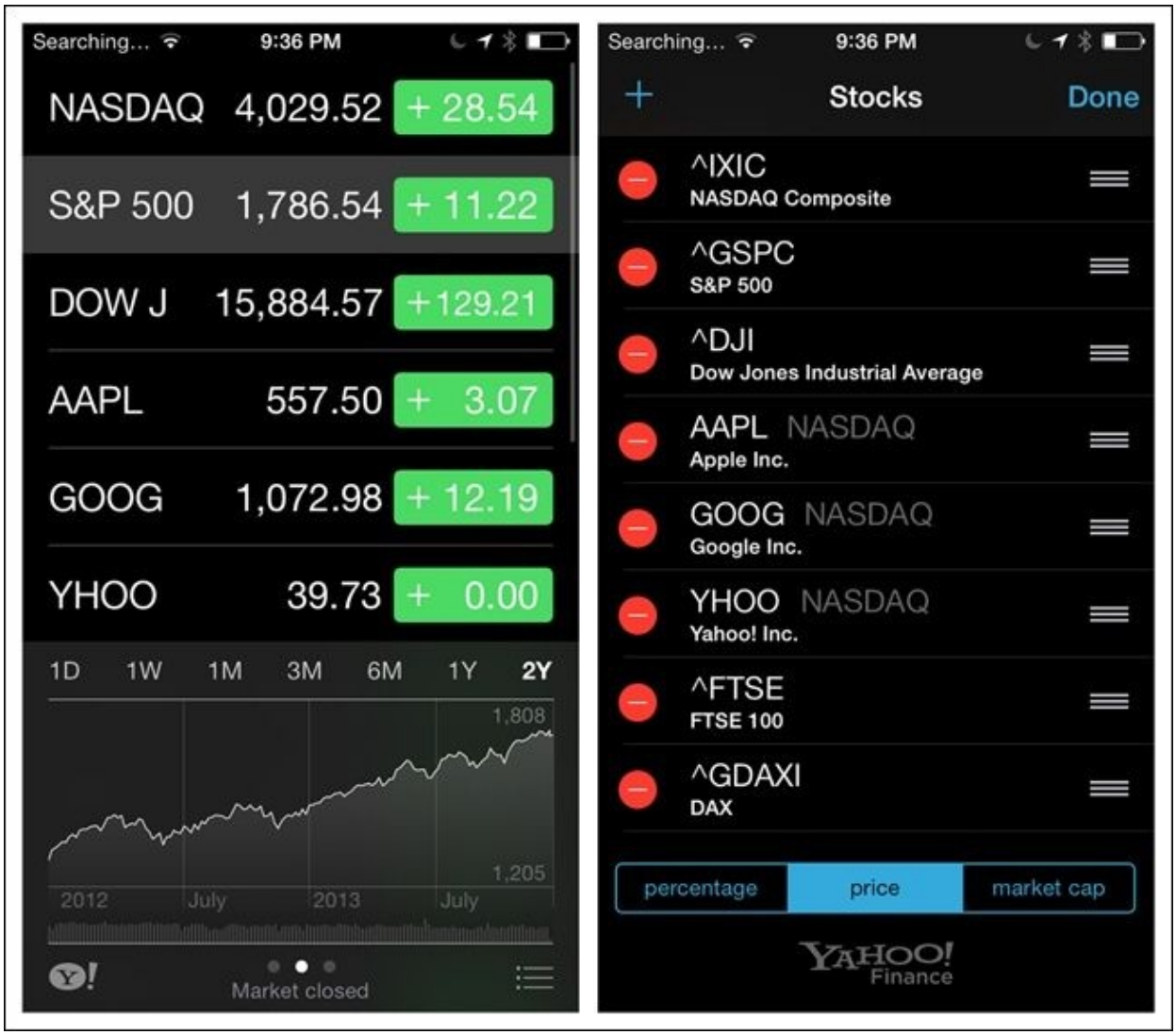



*Figure 5-39. Stocks for iOS: Bulk Actions in edit mode*

Fidelity for Android uses the same pattern, both for selecting stocks and for selecting and ordering the columns displayed in the Watch List.

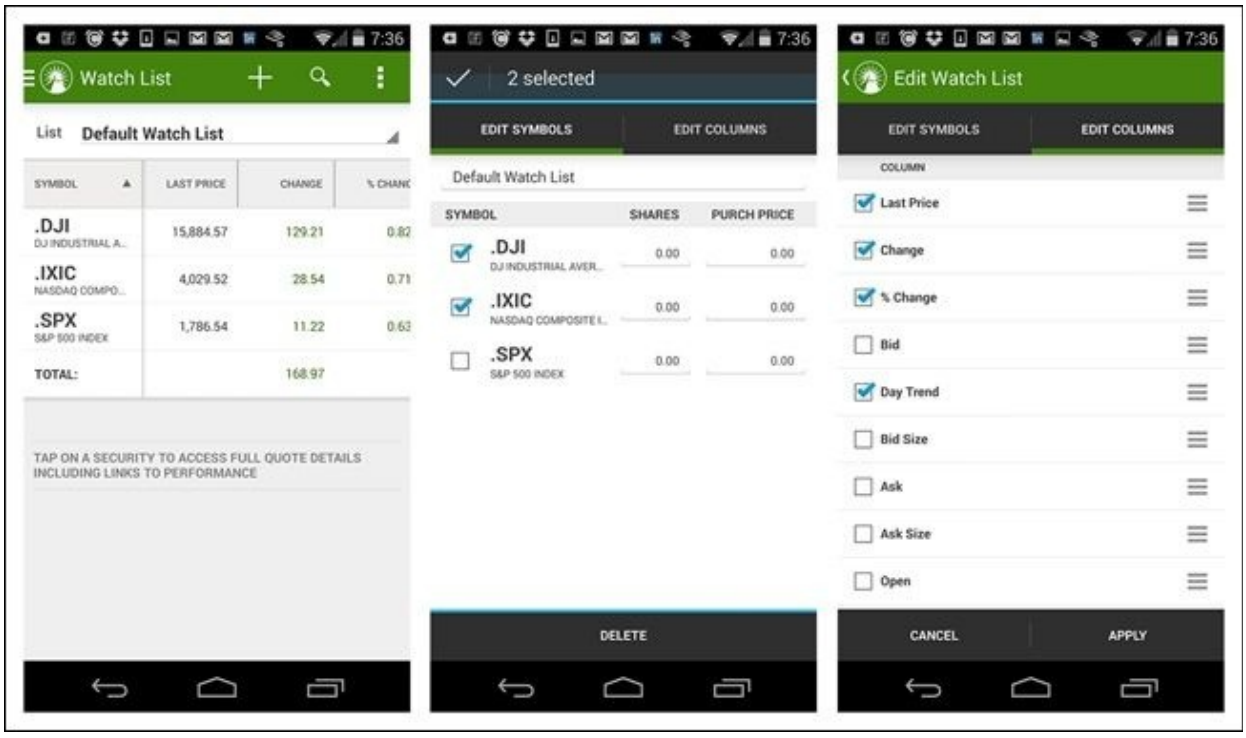



*Figure 5-40. Fidelity for Android: bulk selection of stocks and Watch List criteria*

Path for iOS also offers Bulk Actions in edit mode, like add, remove, and assign a contact to your inner circle (the star icon).

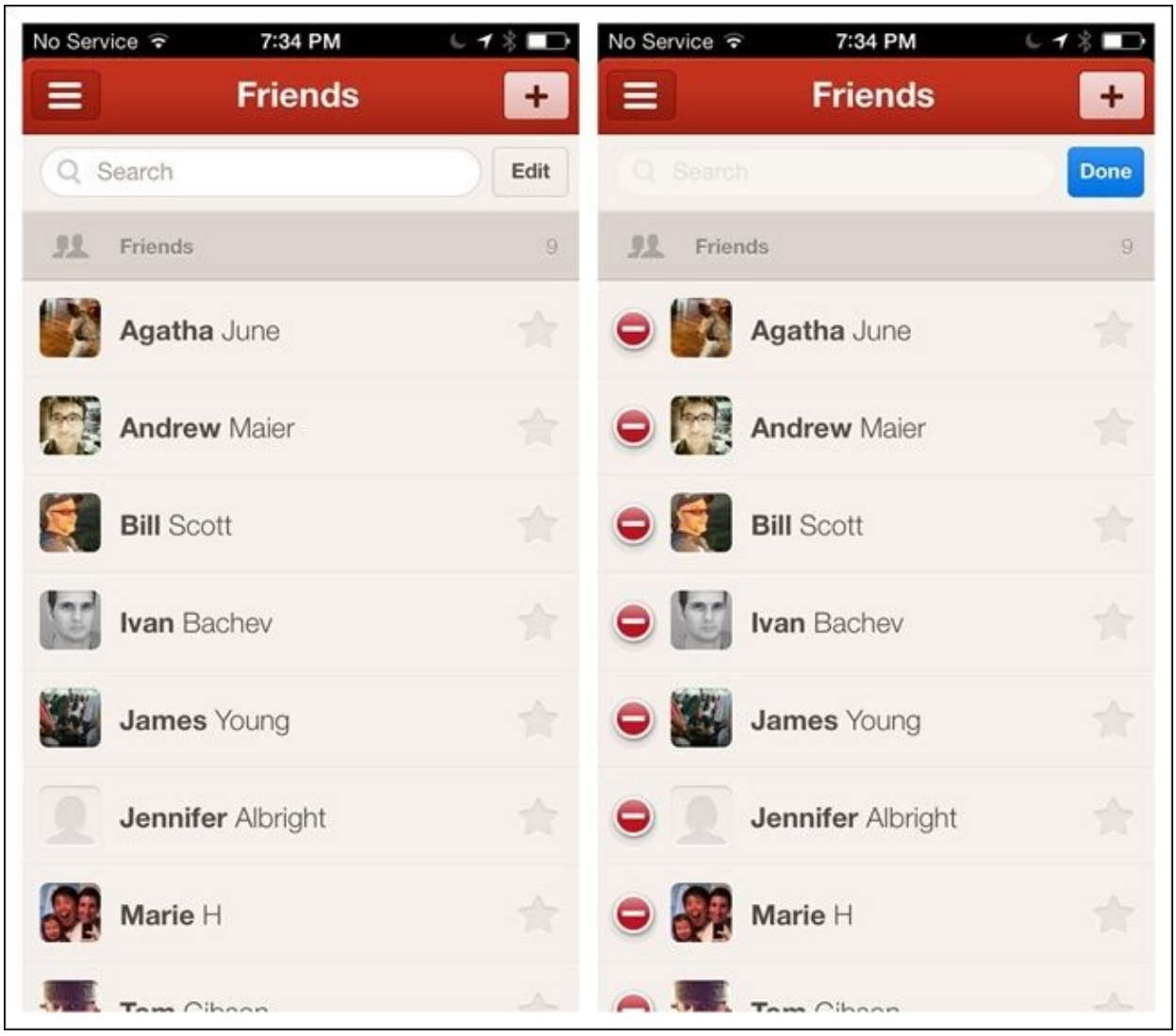



*Figure 5-41. Path for iOS: Bulk Actions in edit mode can act on multiple items at once*

To trigger this mode in Path for Android, you long-press any friend in the list. Path could improve this UI design by following Android design conventions and showing the number of selected items, instead of just shading them.

**NOTE**

Bulk Actions like delete and reorder are best handled in a discrete edit mode. Provide an obvious option for exiting the mode.

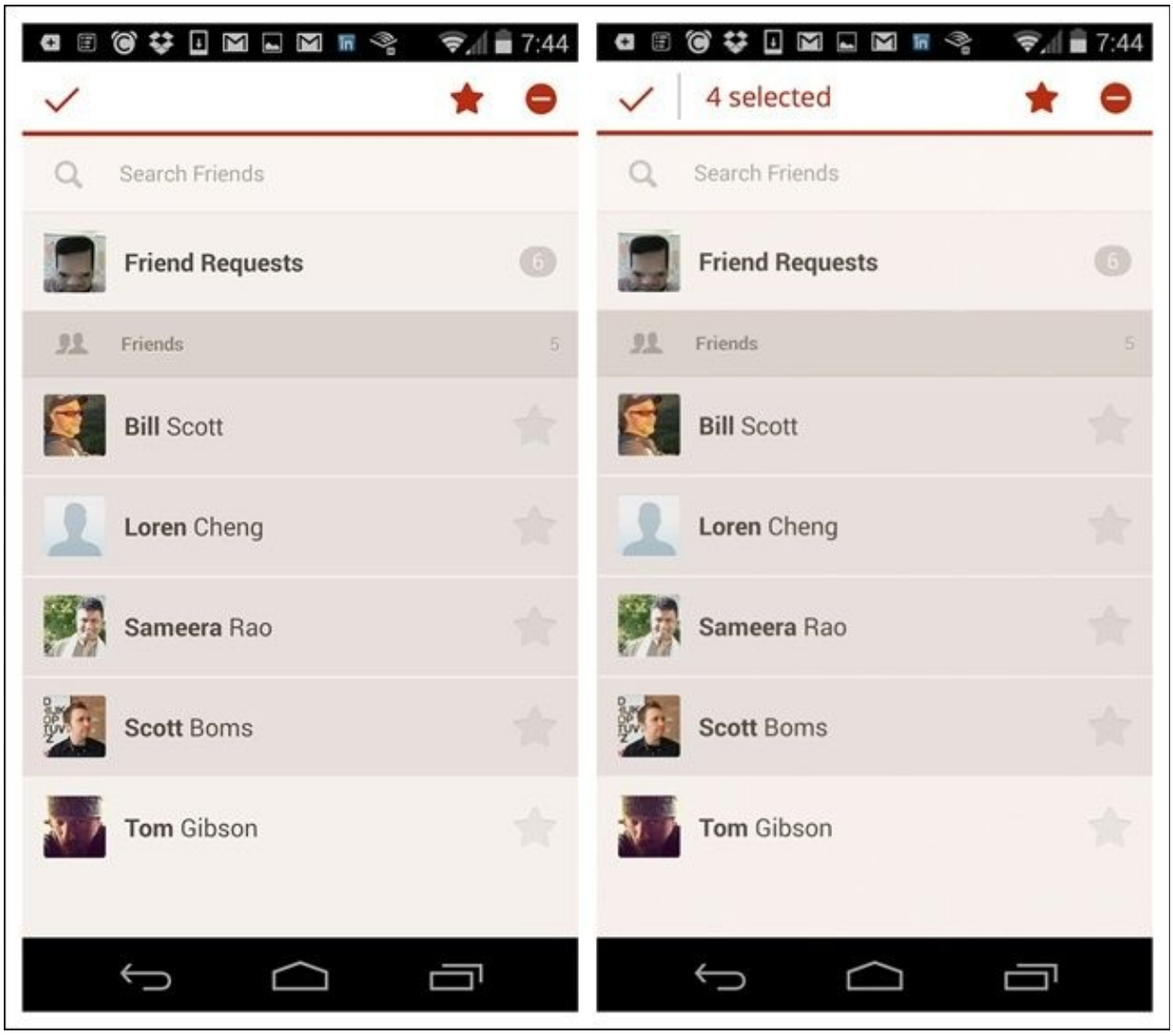



*Figure 5-42. Path for Android: current design on left; suggested improvement, showing selected number of items, on right*

## Lock Screen Controls

Both iOS and Android offer the option to display tools on the lock screen. This is seen most often in music apps, but not exclusively. If your application is running in the background and users may need quick access to pause, stop, or adjust it, consider designing tools for the lock screen.

The controls do not need to be limited to playing and pausing audio and video. Any.do for Android uses a widget to prompt app usage first thing in the morning, and also to help with tasks during the day.

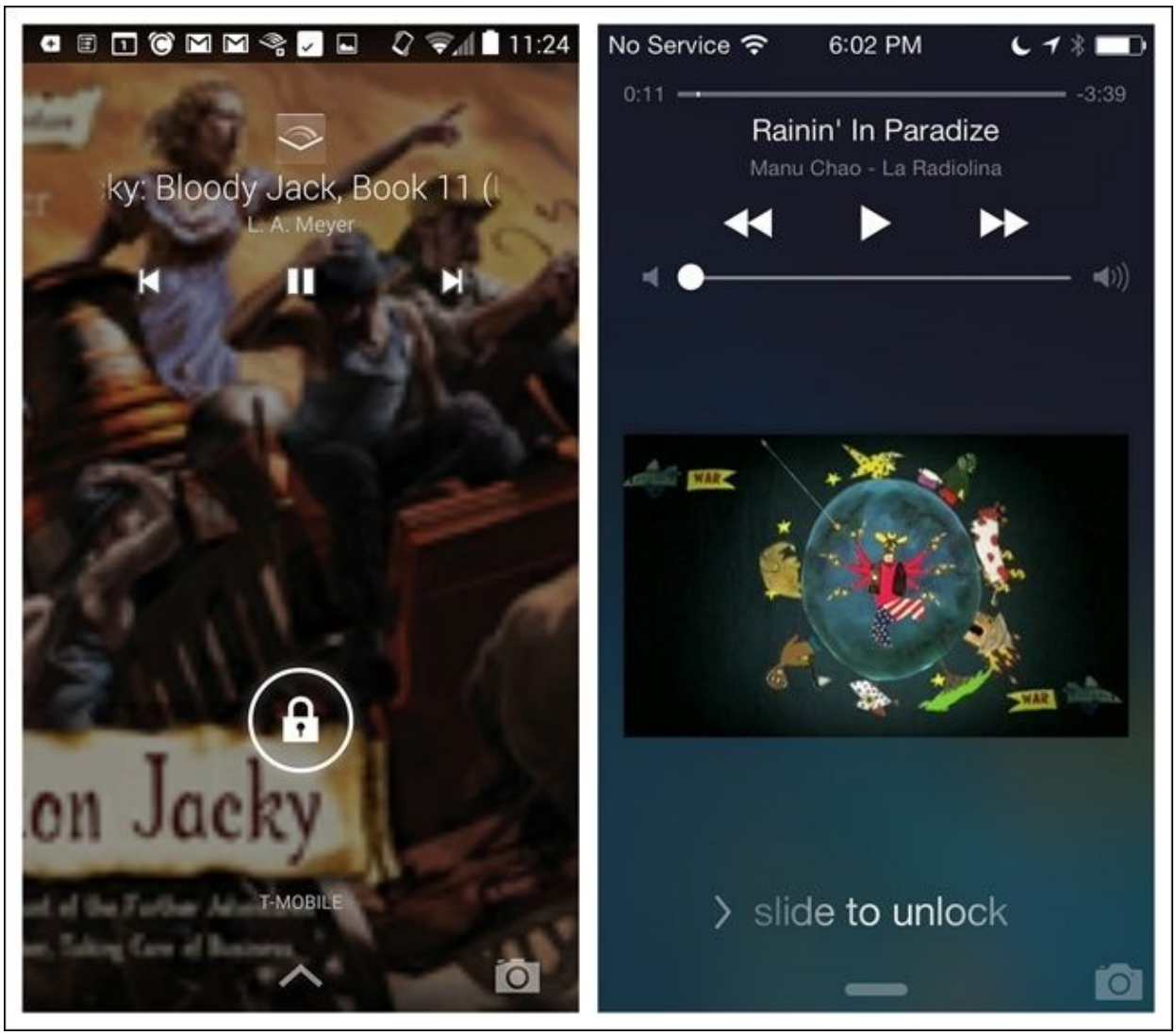



*Figure 5-43. Audible for Android and iTunes for iOS: lock screen controls*

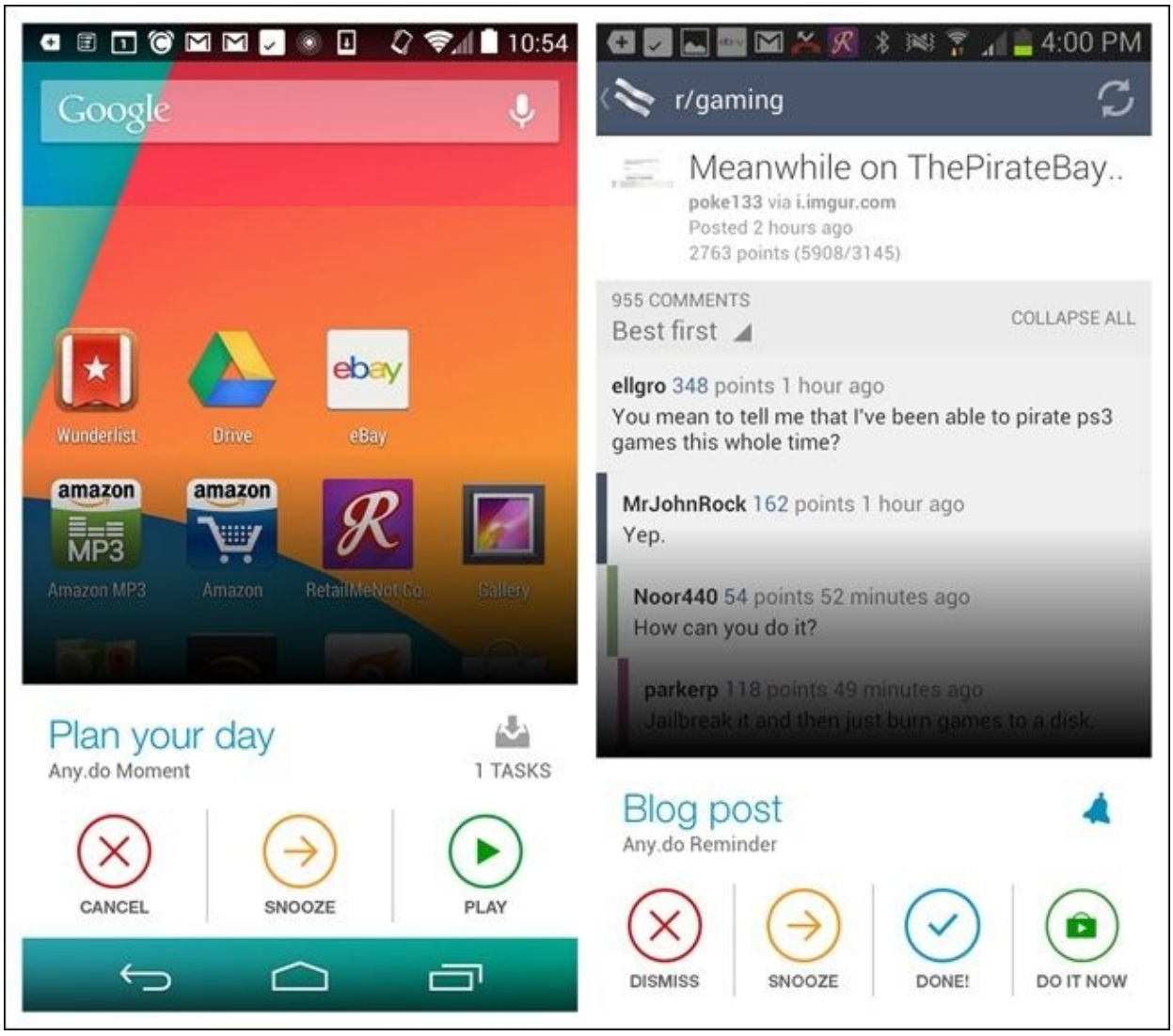



*Figure 5-44. Any.do for Android: widgets offer deeper engagement than notifications*

**NOTE**

The mobile user experience may extend past the typical borders of the application. Consider lock screen controls or widgets to enhance the UX.

# Chapter 6. Charts

*Patterns*

Chart with Filters, Interactive Timeline, Data Point Details, Drill Down, Overview plus Data, Interactive Preview, Dashboard, Zoom, Sparklines, Integrated Legend, Thresholds, Pivot Table

Designing charts for the mobile form factor is a challenge. However, it also provides an opportunity to get to the very core of the best practices for data visualization. A chart should tell a story by depicting important relationships in data. If it isn’t doing that, it is just a waste of space.

In the first part of this chapter, I offer some tips to help you make sure the charts you design are clear and meaningful to your audience. Along the way, we’ll look at some good and bad examples to illustrate these tips. In the second part of the chapter, we’ll look at common chart design patterns. Finally, we’ll wrap up by looking at a couple of apps that pull everything together in exemplary fashion.

Tip 1: Learn the basics

Many of the guidelines and best practices for print and desktop chart design apply to mobile chart design as well. A great introductory book on this topic is *The Wall Street Journal Guide to Information Graphics* by Donna Wong

(Norton, 2013). “Data Visualization Best Practices” (*http://slidesha.re/1fFbP2W*), a presentation on SlideShare by business information analyst Jen Underwood, is a very quick and easy way to learn the basics.

There are many chart styles you can use to present data. Some that you may be familiar with include the bar chart, the line chart, and the pie chart. There is a lot of contention among experts about which charts are best. For instance, many pros, like Jen Underwood, strongly discourage the use of pie charts; others strongly defend them. I’m not going to take sides either way. I’ve seen pie charts that served their purpose well (some of which are included in this chapter), and I’ve seen bar charts that were so poorly designed they were all but indecipherable.

If you look at online design galleries like Dribbble or Behance, you’ll see many beautiful mobile chart designs. Unfortunately, many of them are artistic exercises that aren’t very helpful at communicating information. The two examples shown here look nice, but they are both missing titles that provide context for the data being shared. The designers’ use of color also inadvertently creates correlations in the data where none exist.

The key is to learn the basics so that you use the right chart for the job, and your chart includes all the necessary information and nothing more. At a minimum, the necessary information will include a chart title, clearly labeled axes, and the data.


*Figure 6-1. While pretty, these design examples found on Dribbble are uninformative and confusing*

## Tip 2: Keep it simple

The simpler your chart, the better. Beware the urge to spice up your charts with extra data or visual effects that decrease their effectiveness. Remember the experts I referred to earlier? They call these excess frills “chart junk,” and there’s nothing they like better than finding examples of it to ridicule. So do yourself and your users a favor, and don’t provide any! See Chapter 11 for more examples of chart junk.

Argus uses basic column and line charts to communicate progress against specific user goals for walking and running.

Other simple read-only charts can be found in Trulia, which uses a donut chart (a variant of the pie chart) to show the breakdown of a mortgage payment, and Zillow Mortgage Calculator, where a line chart shows trends in mortgage

rates.


*Figure 6-2. Argus for iOS 7: simple column and line charts communicate the user's progress toward fitness goals*

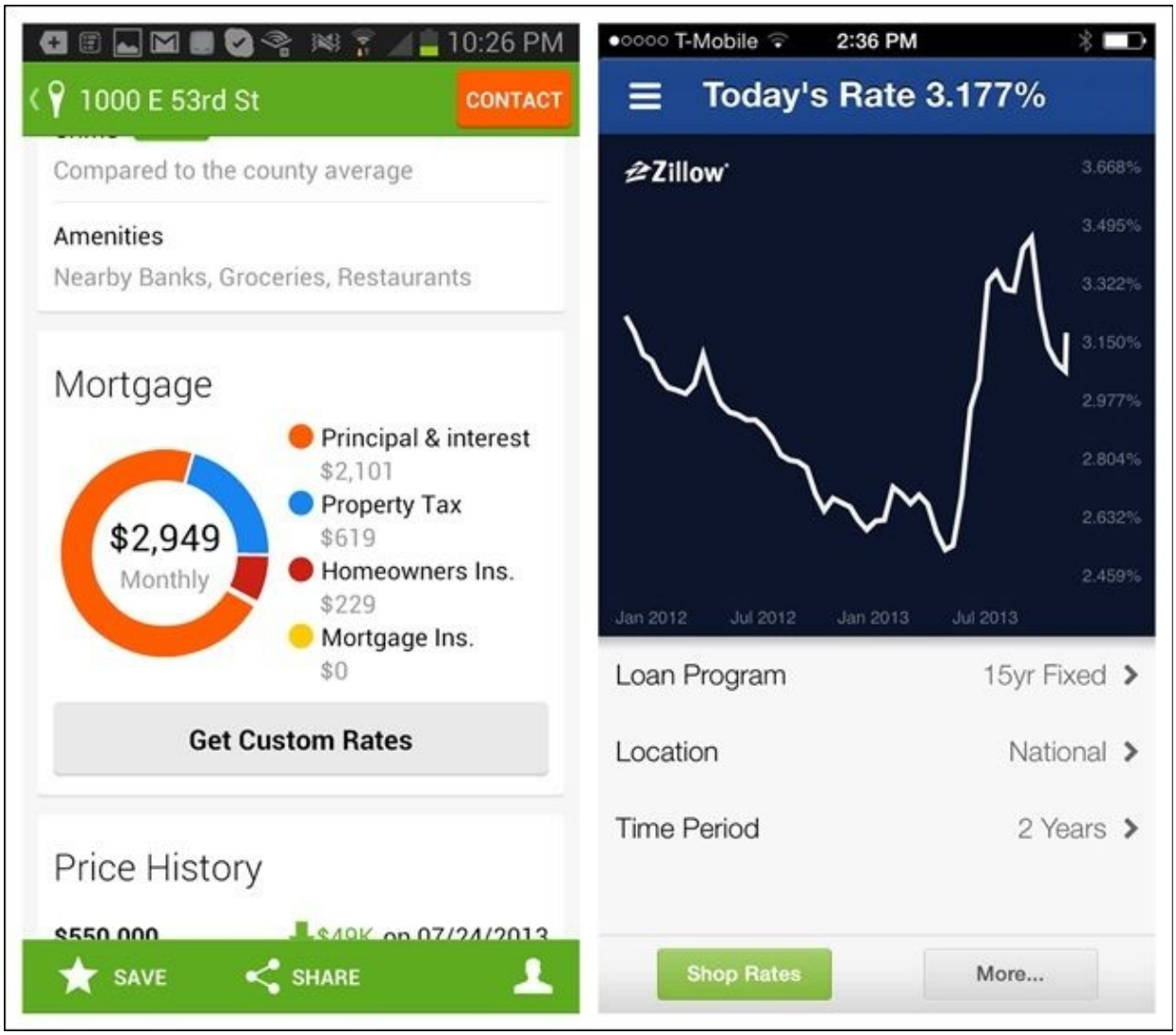



*Figure 6-3. Simplicity rules: donut chart in Trulia for Android and line chart in Zillow Mortgage Calculator for iOS*

Tip 3: Surprise — math quiz!

So your chart looks sexy and everyone on your team likes it. Now the question is, will your users understand it? To find out, when you conduct your user testing, write the usability script like a math test. Ask the participants specific questions about the data. If they can't answer them, the chart is either the wrong type, too complicated, or missing critical labels — or possibly all three.

For instance, in testing the chart from 1Weather for Android, questions I might ask the participant would include:

- What time period does this chart cover?

- What was the chance of rain on Friday? What about on Monday?
- Which day this week will be the hottest? The coolest?

During testing, it would probably become apparent that participants are unsure of the difference between the solid line and the dashed line. A simple label change could remedy this, but test again to be sure. I like to sketch along as participants respond (or ask the participants to sketch) during test sessions and iterate on the chart design until they can answer all the questions correctly.


*Figure 6-4. 1Weather for Android: missing legend on left, added legend on right*

Tip 4: Know your toolset

Before cracking open Photoshop, you should get familiar with the charting library your dev team is using. There are many different options available; here are some popular ones worth exploring:

- Core Plot (*https://code.google.com/p/core-plot/*) for iOS

- ShinobiControls (*http://www.shinobicontrols.com/*) for iOS and Android
- TeeChart (*https://components.xamarin.com/view/TeeChart*) for Xamarin
- Telerik RadControls (*http://bit.ly/1fFcbGI*) for Windows Phone
- A JS charting library like AnyChart (*http://www.anychart.com/products/anychart/gallery/*) or amCharts (*http://www.amcharts.com/*)

Ask your developers to show you the available features and consult with them on the chart design. Aside from user testing, early collaboration between the designer and developer is the best way to ensure your users will get the charts they need.

Okay, now that you've learned some basic tips, it's time to look at the most common design patterns used with the charts in mobile applications.

## Chart with Filters

A simple chart can be enhanced with time controls or other filters. In apps like Fidelity and StockPlus, the data can be explored for different time periods.

When you design an Onscreen Filter, provide plenty of space for the touch target. In Chaikin, the filter controls are in the chart, where they are difficult to even see. Since the first release of this book, Chaikin has redesigned this screen and moved the time filters to the left, but that didn't fix the problem at all.


*Figure 6-5. Fidelity for iOS and StockPlus for Windows Phone: time period filtering*


*Figure 6-6. Two generations of Chaikin for iOS: filter controls are still hard to see*

The DailyBurn Tracker app provides a contextual filter; tapping the chart reveals the actions across the top and the time filters across the bottom.


*Figure 6-7. DailyBurn Tracker for iOS: tapping the chart reveals filters*

Time, of course, is not the only possible filtering option. In Fidelity for iOS, for example, data sources can be selected for comparisons.

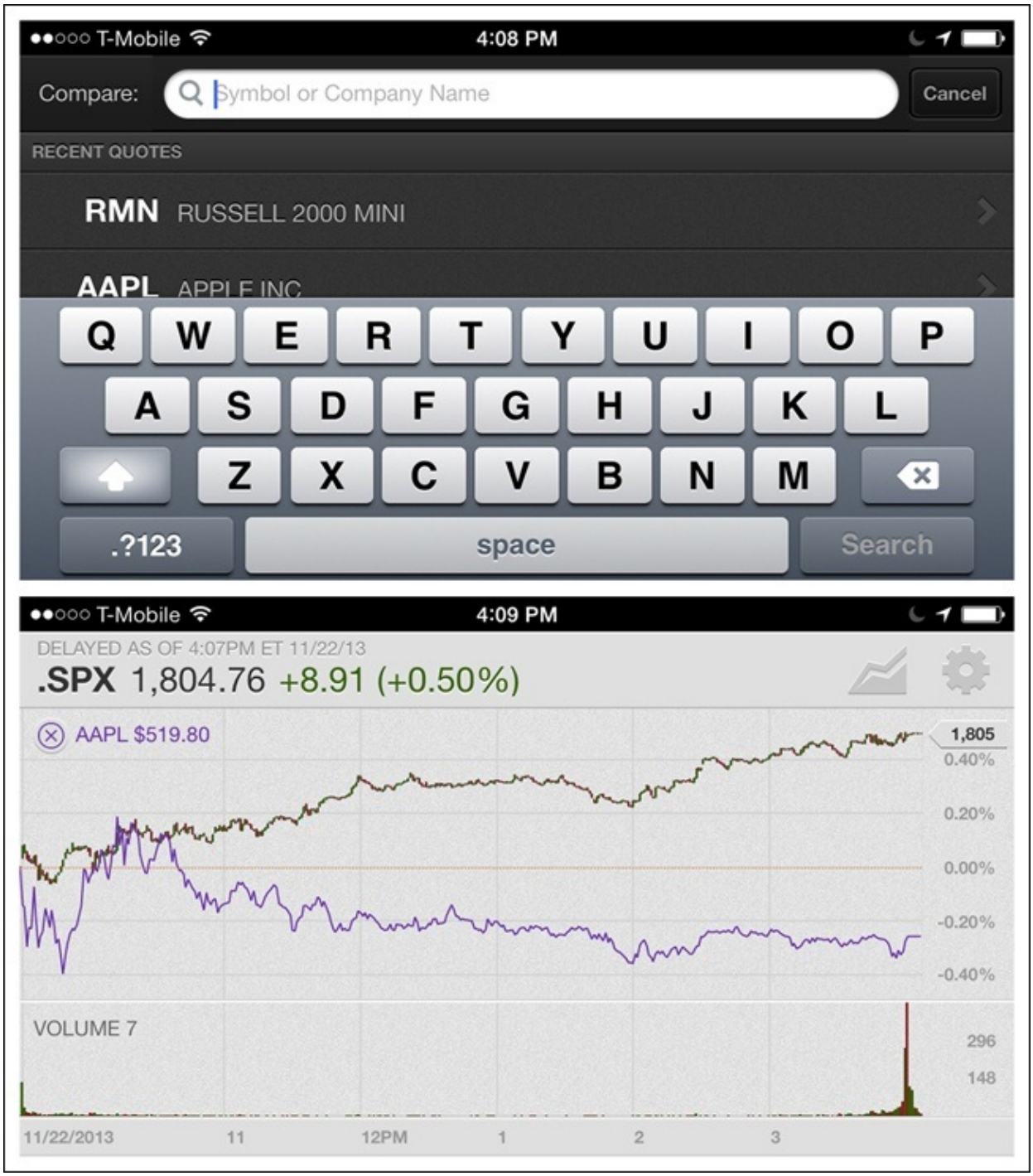



*Figure 6-8. Fidelity for iOS: data source comparisons*

The Fidelity app also offers configuration options for changing the chart type, thresholds, and pricing axis.

**NOTE**

Use common patterns for filtering, like the OnScreen Filter, Filter Drawer, and Filter Form (see

Chapter 4).

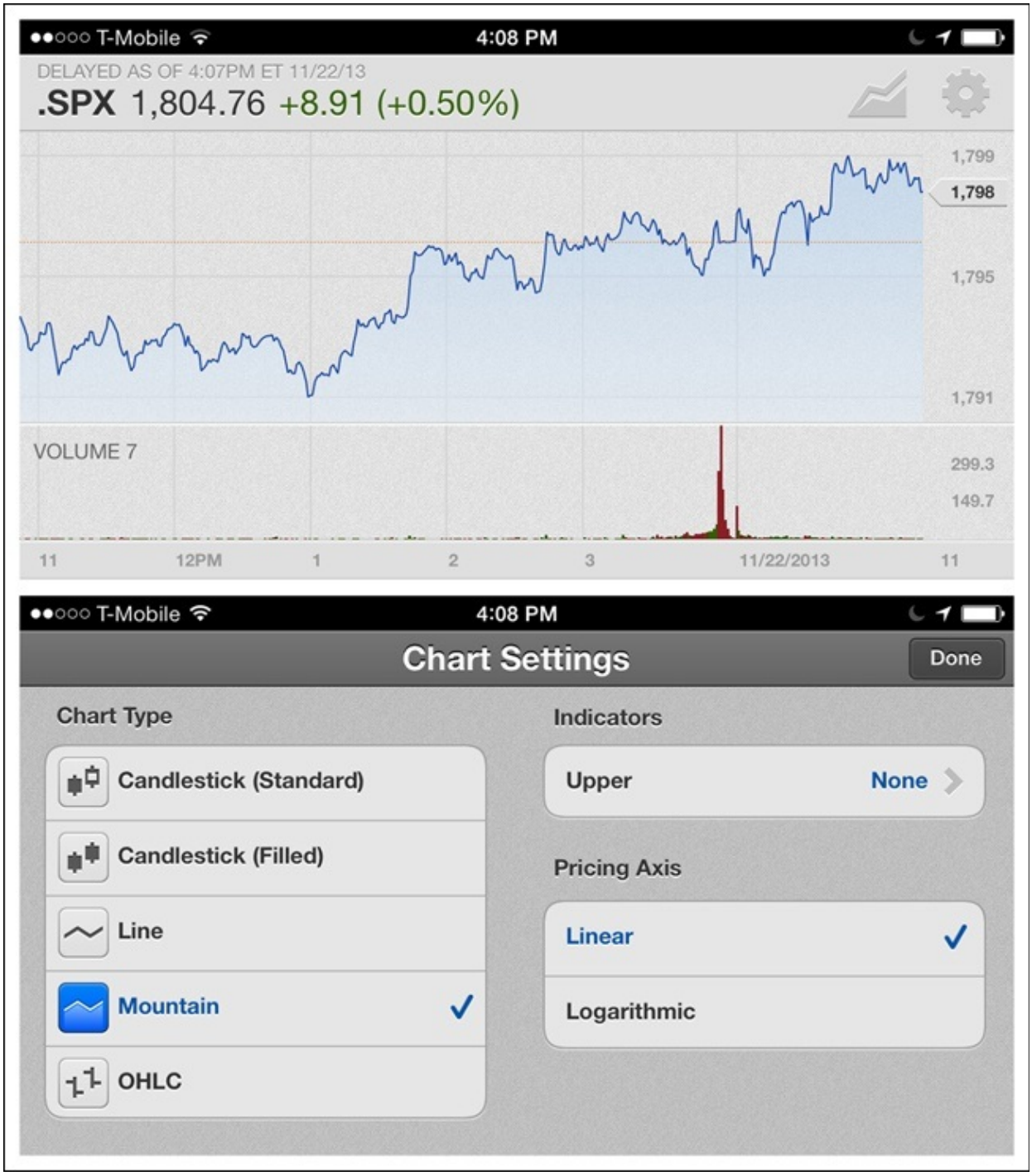



*Figure 6-9. Fidelity for iOS: comparisons and configuration*

## Interactive Timeline

In the first edition of this book I included a pattern called Scrolling Window, exemplified by Roambi for iOS. I've expanded the pattern to include any type of Interactive Timeline combined with a chart or data visualization.

In the Roambi example, the small chart across the bottom of the screen provides an interactive control (two drag handles) for selecting the point in time displayed in the top, much larger chart.

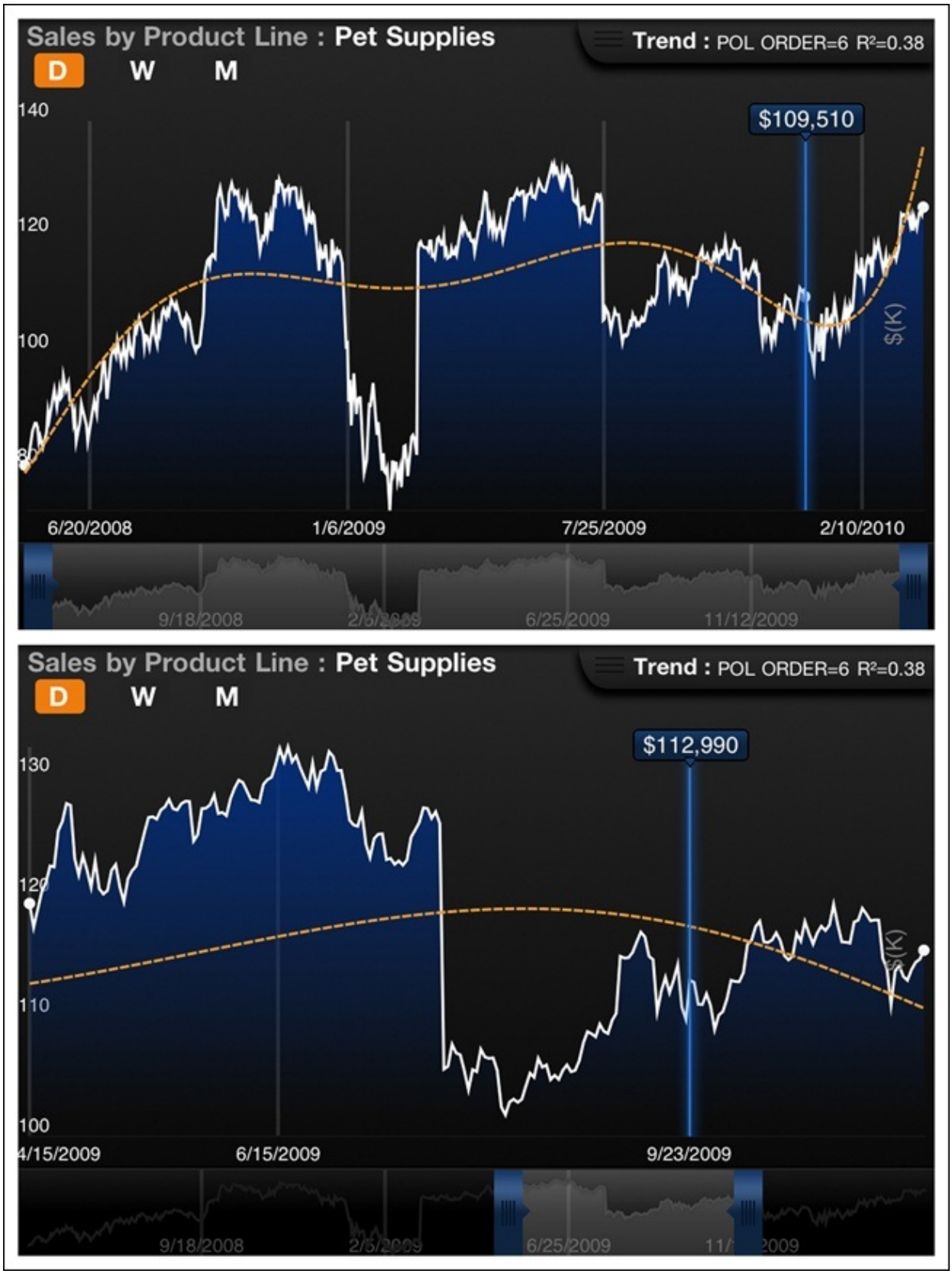



*Figure 6-10. Roambi for iOS: handles for selecting a window in time*

The Dark Sky weather app provides a timeline, plus a Play button to animate the change over time. It has also added a Now option to pop back to the current point in time.

**NOTE**

An Interactive Timeline can provide users with more freedom and control in selecting context than a static chart or visualization.

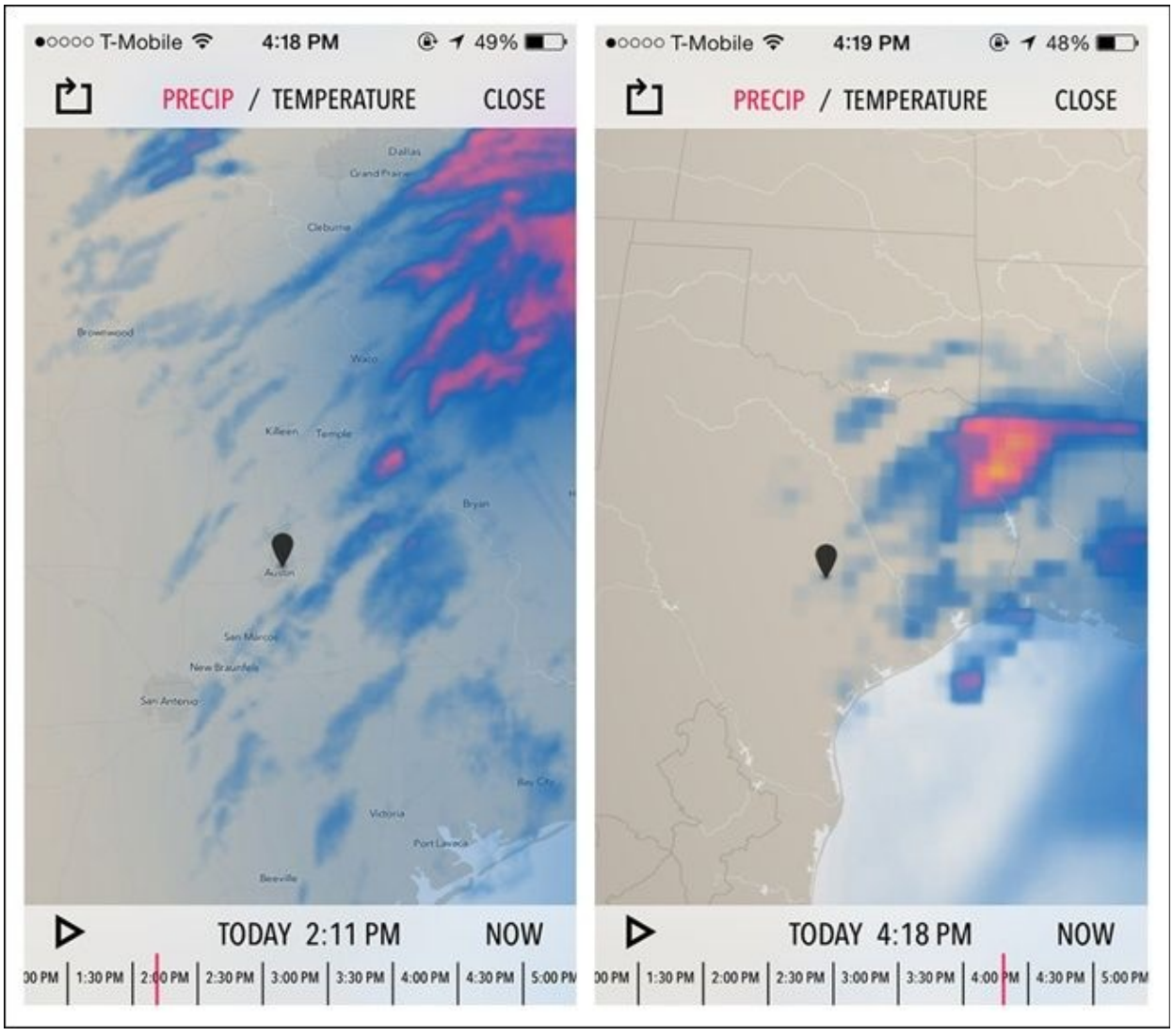



*Figure 6-11. Dark Sky for iOS: Interactive Timeline plus Play button (http://youtu.be/d_ZJYVPxRe8)*

## Data Point Details

Since hover doesn't exist in mobile and long press isn't very intuitive, a different interaction is needed to show details for a specific data point. In the Zillow Mortgage Calculator for iOS, touching a wedge in the pie chart displays details about the payment breakdown.

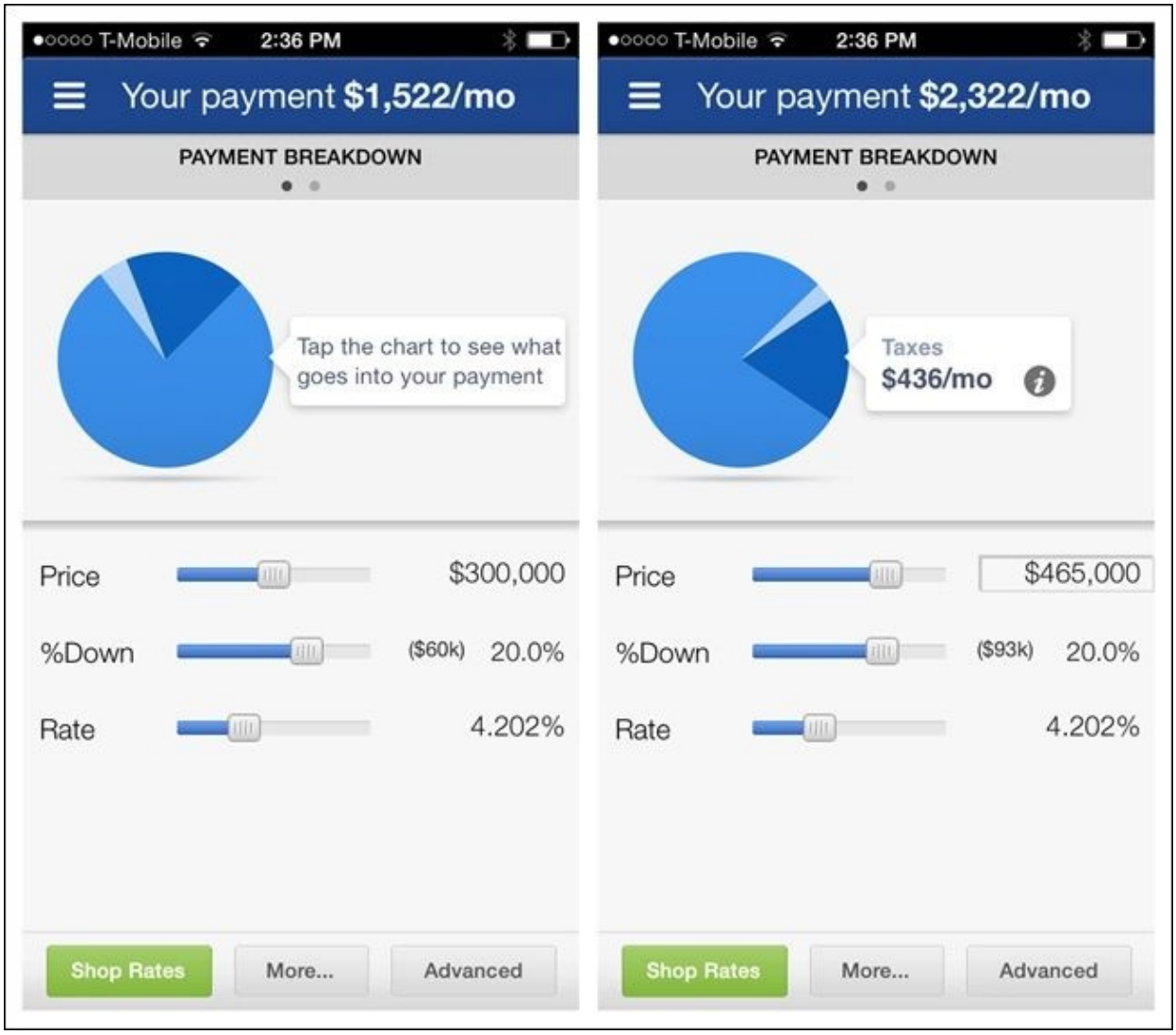



*Figure 6-12. Zillow Mortgage Calculator for iOS: touching the chart area reveals Data Point Details*

In Apple Stocks and Yahoo! Finance, tapping a data point highlights it in the Preview Window at the bottom of the chart (although you have to look pretty closely to see it). In these apps, the whole chart is a touch target, so you can tap anywhere, and then drag to zero in on the specific date.


*Figure 6-13. Apple Stocks and Yahoo! Finance for iOS: tap data points for more info*

In the Apple Stocks app, a single touch shows the details for that data point, but touching two points (multitouch) shows the difference between the two points.

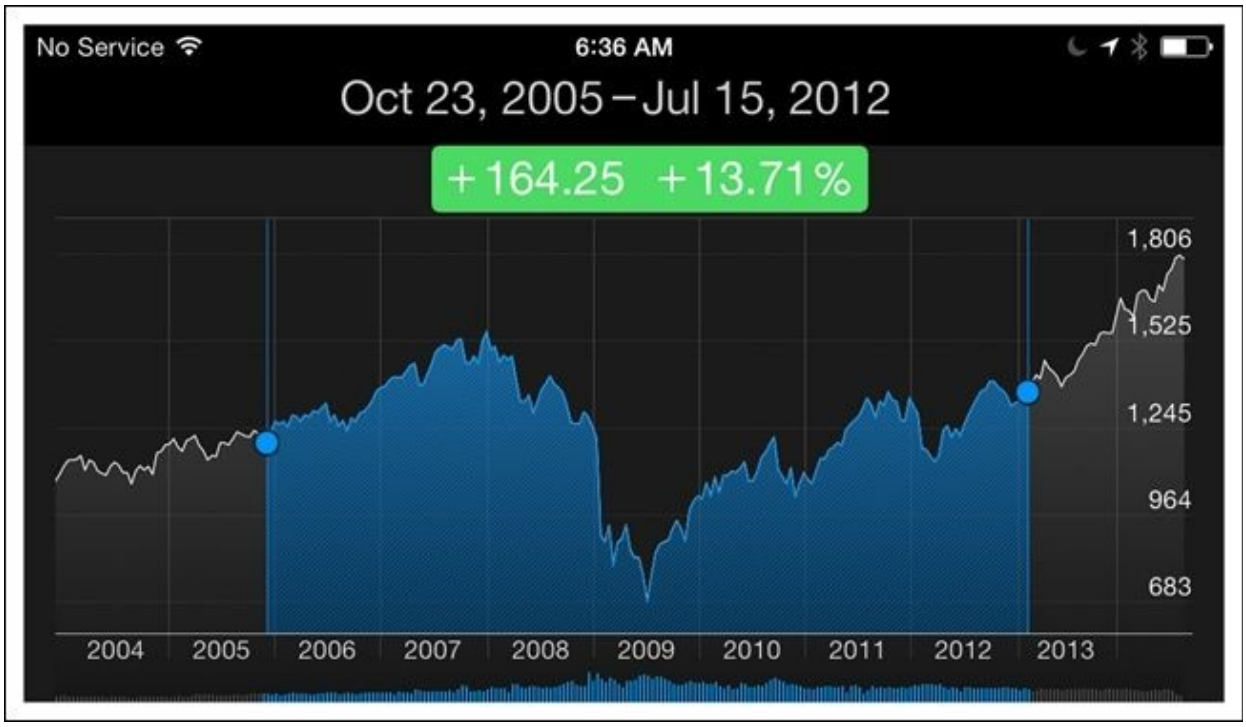



*Figure 6-14. Apple Stocks for iOS: single and multitouch Data Point Details*

In Traffic for Google Analytics, the data points are large enough to tap, but they also automatically highlight the data for the month you swipe to. As I scroll back in time, first September is highlighted, then August, then back to May, and so on.


*Figure 6-15. Traffic for Google Analyticsfor iOS: gestures let you explore Data Point Details and change time periods (http://www.youtube.com/watch?v=UqZkxNT3Ggs)*

In its iOS 7 release, Roambi has introduced a clever way to toggle between multiple data sources to see their details by just tapping on the drag handle.


*Figure 6-16. Roambi for iOS: Data Point Detail toggle (http://www.youtube.com/watch?v=XBipN_7XMnU)*

Dark Sky encourages users to swipe to see the details. When the week view loads, the first row "bumps" to invite a swipe action. Once the row is swiped, that day's weather details are revealed.

Withings uses a combination of Data Point Details plus the Drill Down pattern, which we'll look at in more detail next. Tapping on the data point shows the details; then, tapping on the details overlay drills down into a new screen for editing or taking other action.

**NOTE**

Our web experiences have set the expectation that Data Point Details will be displayed on hover, which isn't available in mobile apps. Consider serving details when data points are tapped to provide the additional information users want.

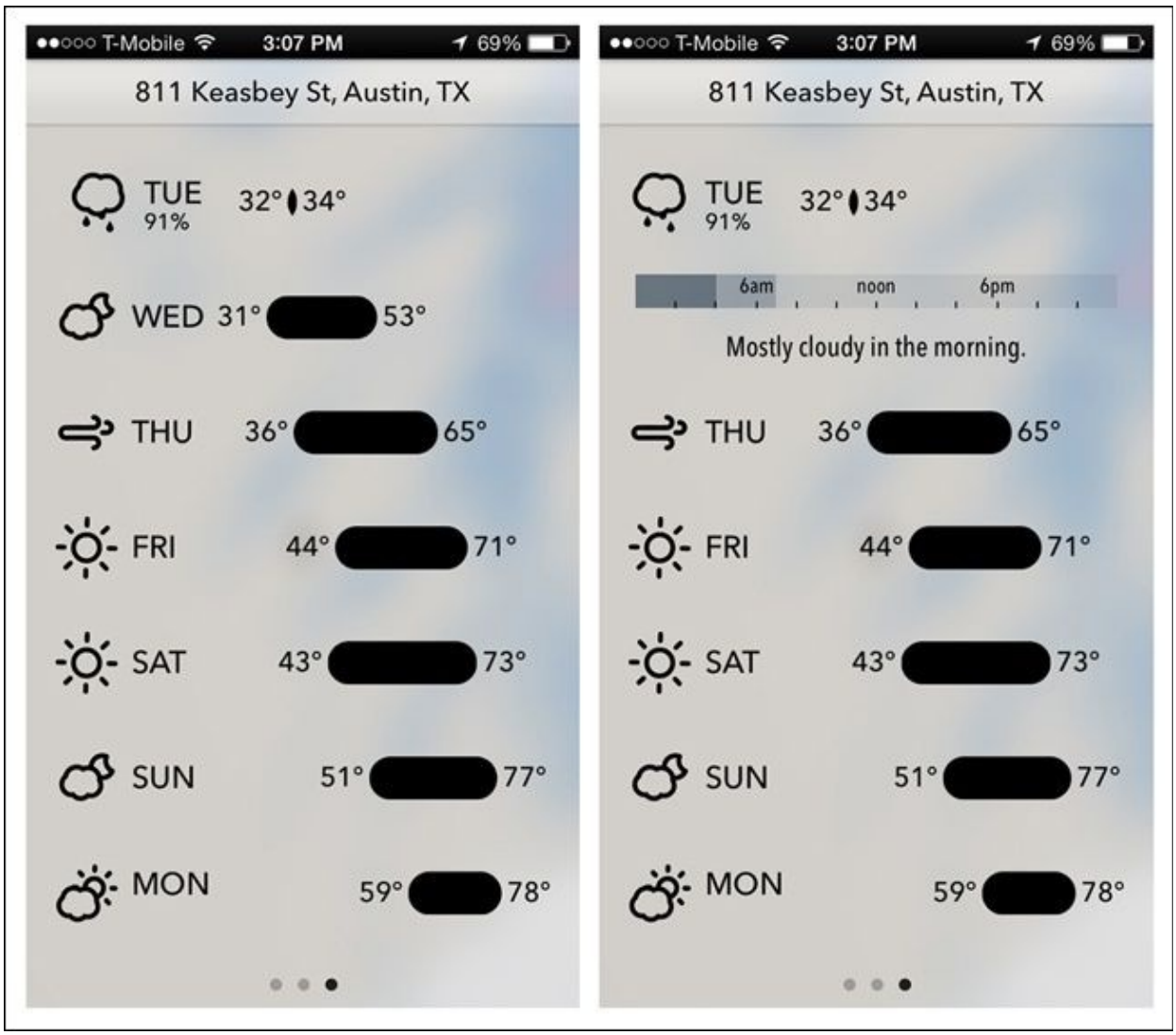



*Figure 6-17. Dark Sky for iOS: swipe row for details (http://www.youtube.com/watch?v=d_ZJYVPxRe8)*


*Figure 6-18. Withings for iOS: tapping the chart shows the Data Point Details; tapping the details navigates to a new screen with even more details*

# Drill Down

The Data Point Details pattern is usually sufficient for showing a bit more information in context. But sometimes users need to explore the data more deeply. That's when the Drill Down pattern can be useful.

In this example from Mint, the top-level view is the Dashboard, which drills down into Budgets, which drills down again to a specific budget for more detailed data (and an option to edit the threshold).

> **NOTE**
>
> Invite the user to Drill Down for more data. Use the OS-appropriate control for navigating back up the chain (i.e., the Back button for iOS and the Up button for Android).

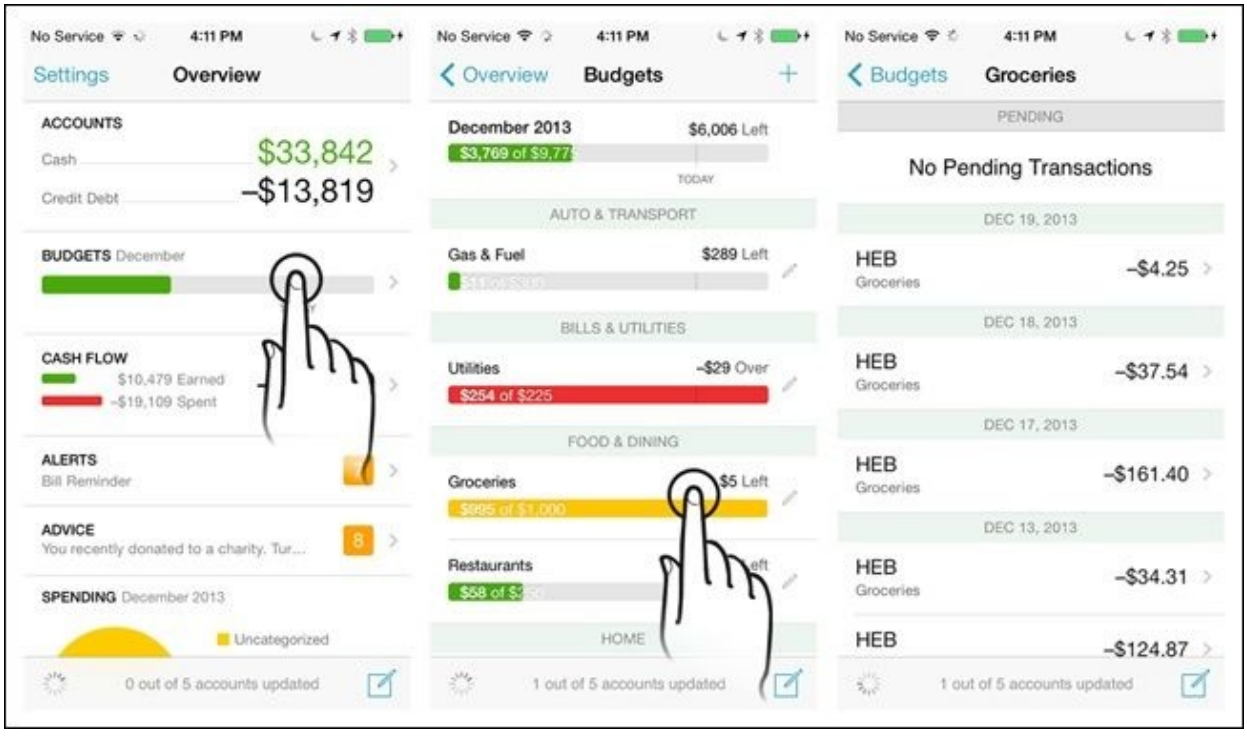



*Figure 6-19. Mint for iOS: Drill Down reveals layers of details*

If there is no clear visual cue that Drill Down is possible, you should use an invitation to encourage first-time users to tap to Drill Down. (See Chapter 8.)

## Overview plus Data

The Overview plus Data pattern is also covered in Chapter 3, but in this chapter it refers specifically to using a chart for the overview. This is one of the chart patterns that has evolved the most in the past couple of years.

In the first edition, the focus was on designs that leveraged a trend chart on top and a detailed data table below, like Gaug.es and Personal Capital.

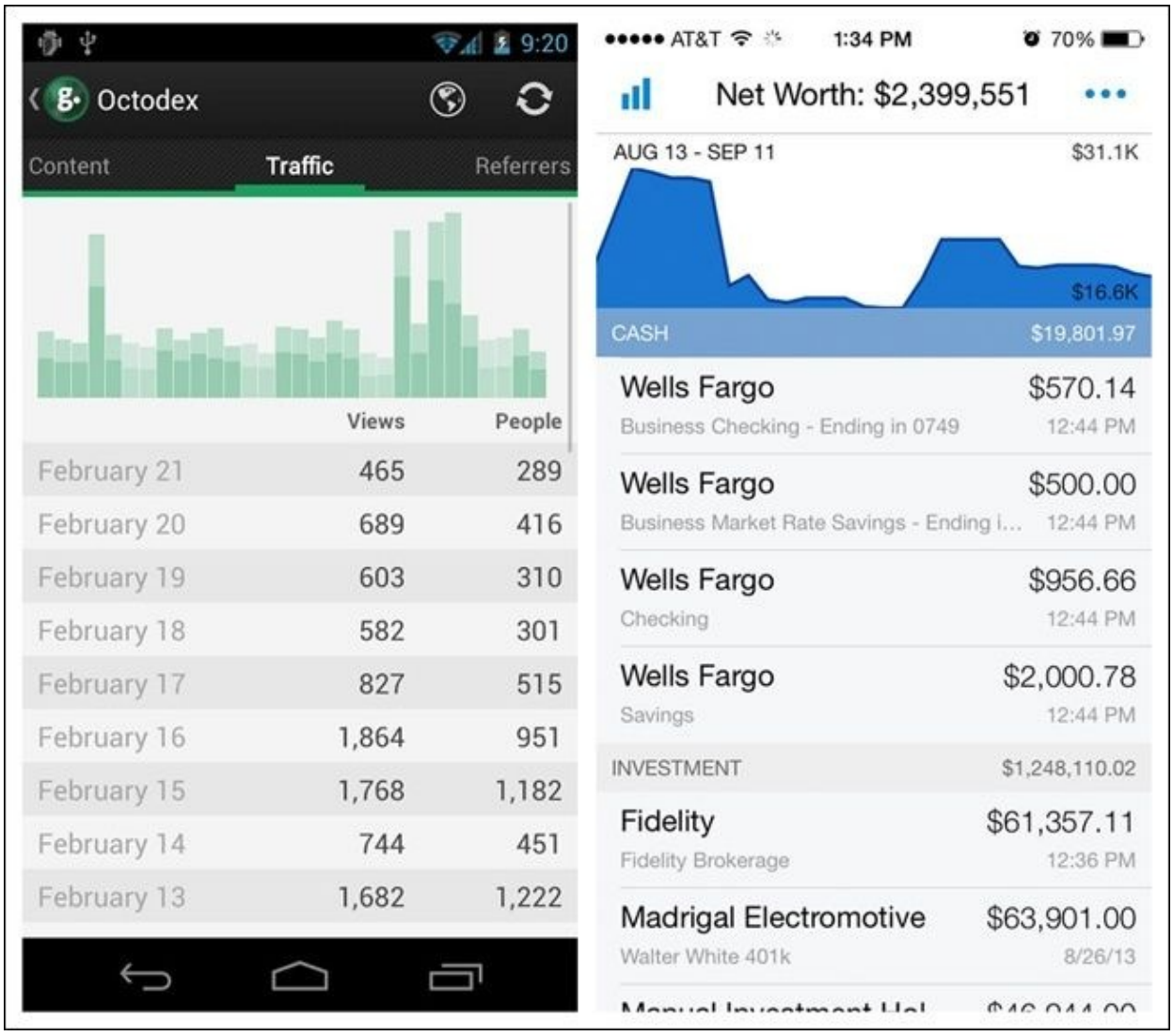



*Figure 6-20. Gaug.es for Android and Personal Capital for iOS: trend chart on top, detailed table below*

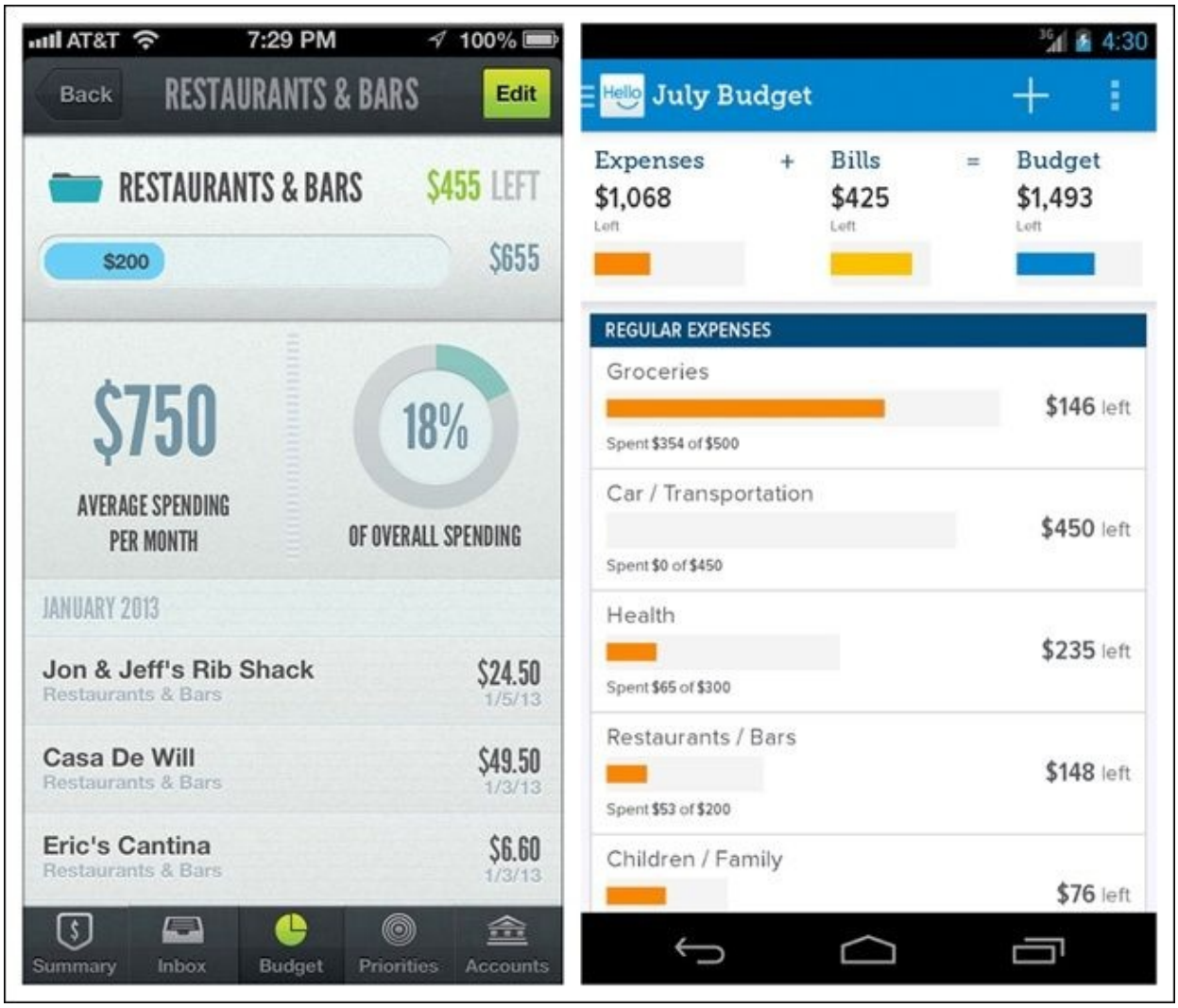



*Figure 6-21. LearnVest for iOS and HelloWallet for Android*

But this pattern is not restricted to read-only charts and their respective data tables. In Personal Capital, for example, an interactive donut chart is used for the overview. Tapping on a wedge highlights the corresponding category and total in the table below.

An earlier version of the app featured a Sparkline (a type of very small chart, covered later in this chapter) in the middle of the donut, affording a bit more context around spending for this category. This has since been replaced with a number indicating the percent increase or decrease, plus an area chart in the background behind the donut showing the trends over time.

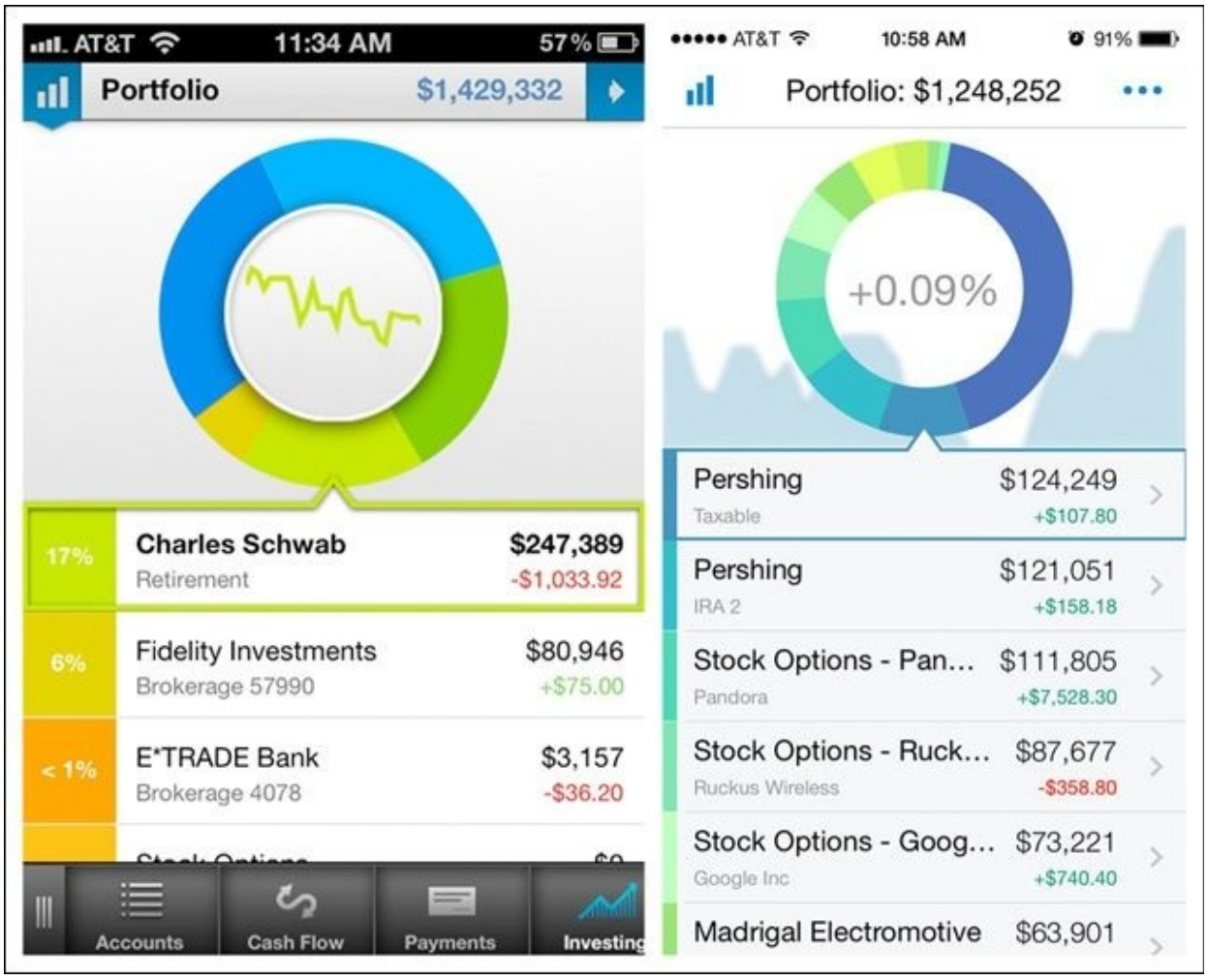



*Figure 6-22. Two iterations of Personal Capital for iOS: Sparkline replaced by background area chart*

Another variation on this pattern is from Move for Android and iOS. The overview is a bubble chart and the detailed data is displayed below on a timeline.

**NOTE**

Use this pattern where a visual overview accompanying the detailed data will be more valuable to your users than just a list of transactions with no summary.


*Figure 6-23. Move for Android: a different take on Overview plus Data*

## Interactive Preview

This pattern is also covered in Chapter 2. In the example from the Trulia Mortgage Calculator, you can see that increasing my annual income from 75K to 295K using the slider on the bottom half of the page redraws the pie chart and updates the legend.

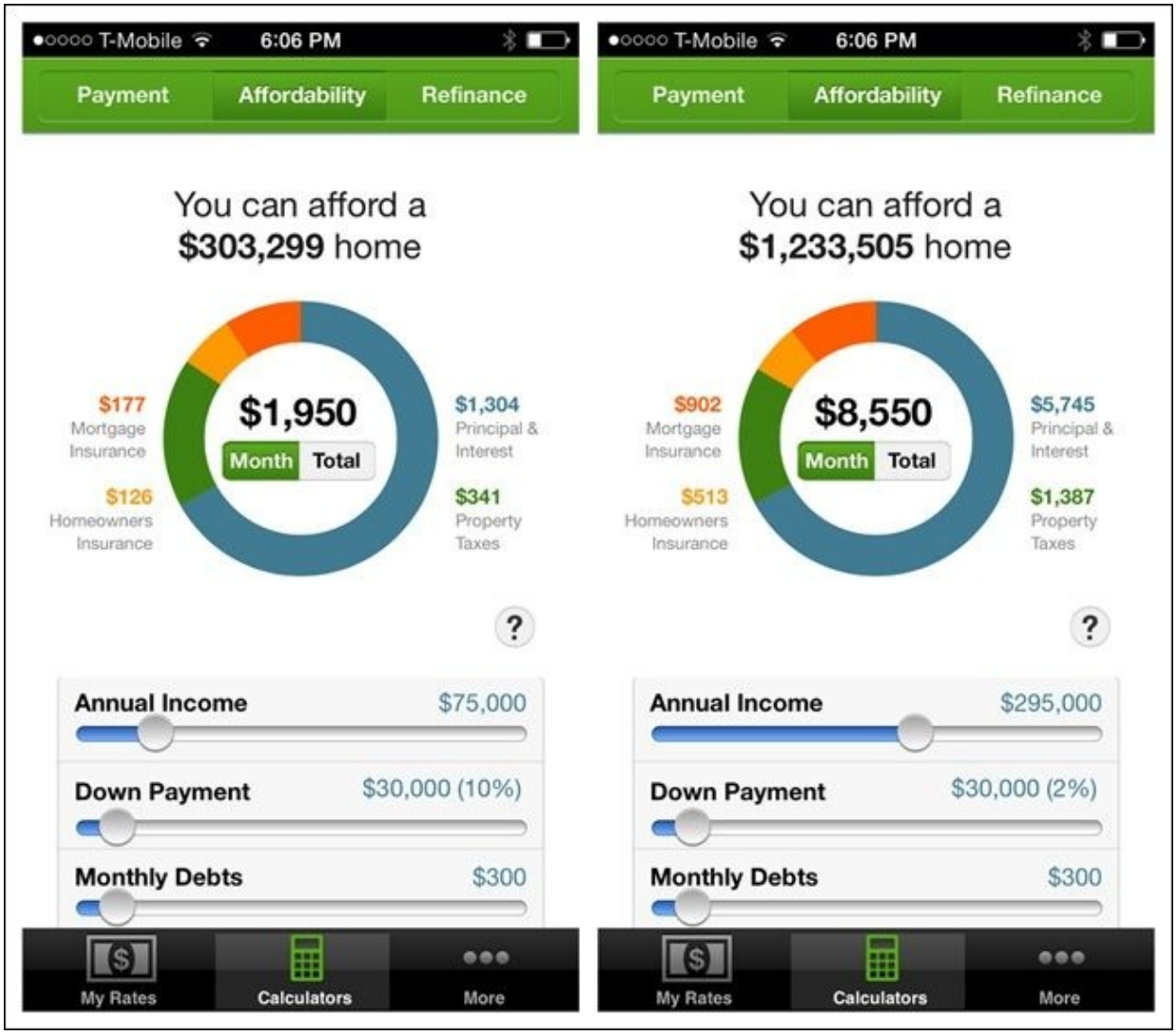



*Figure 6-24. Trulia Mortgage Calculator for iOS: chart with Interactive Preview*

The Zillow Mortgage Calculator and Intuit's TaxCaster utilize the same interaction design.

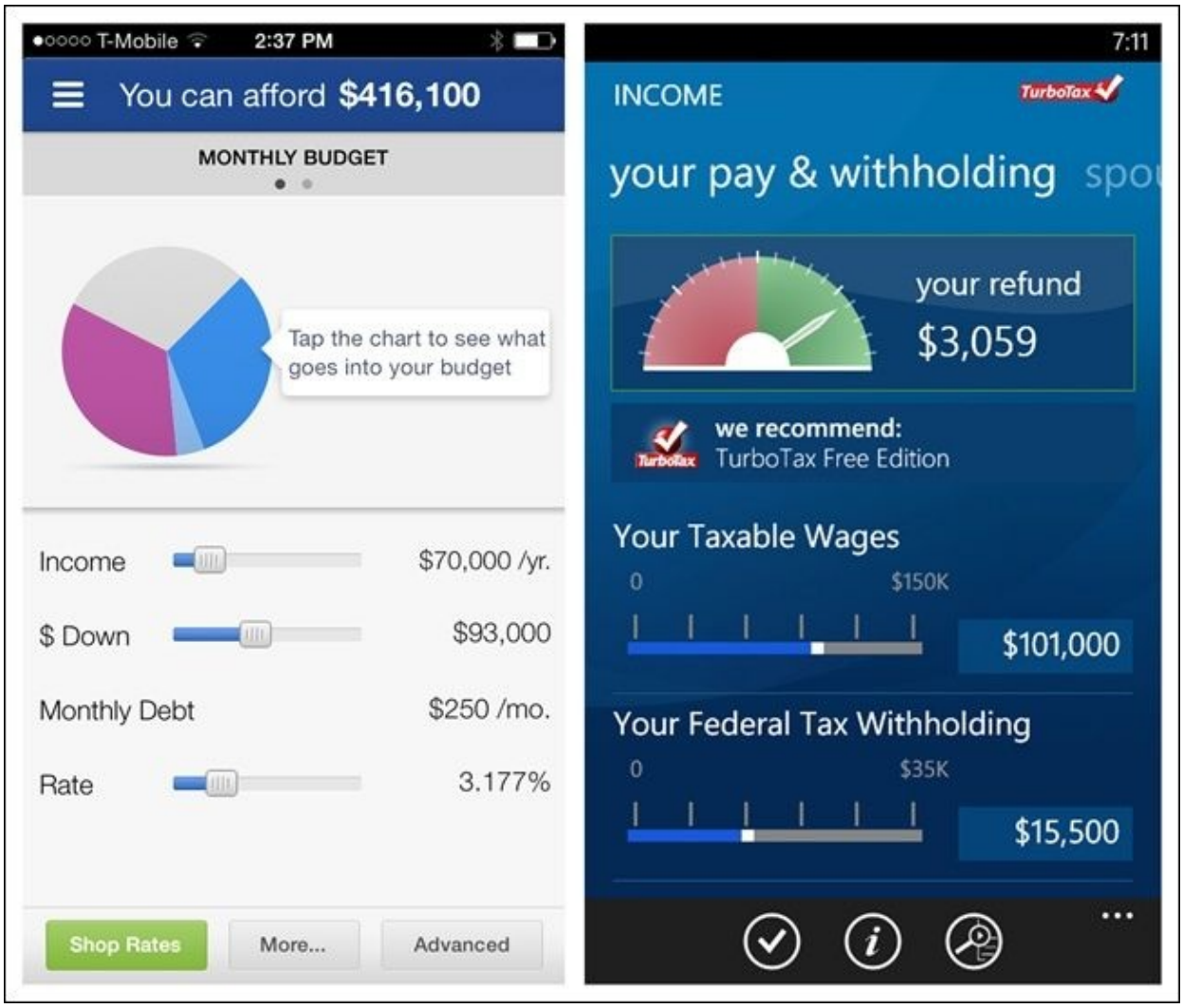



*Figure 6-25. Zillow Mortgage Calculator for iOS and TaxCaster for Windows Phone: Interactive Previews*

Mint offers an Interactive Preview in the Budget screen. The screen initially shows a line chart of the past six months of spending for a selected category. An invitation (see Chapter 8) is also displayed, encouraging the user to "tap or drag to change the budget." If the handle is adjusted, a preview of the new budget is displayed at the top of the screen until the handle is released.

**NOTE**

Interactive Previews are useful for displaying multiple forecast scenarios, as in personal finance applications.

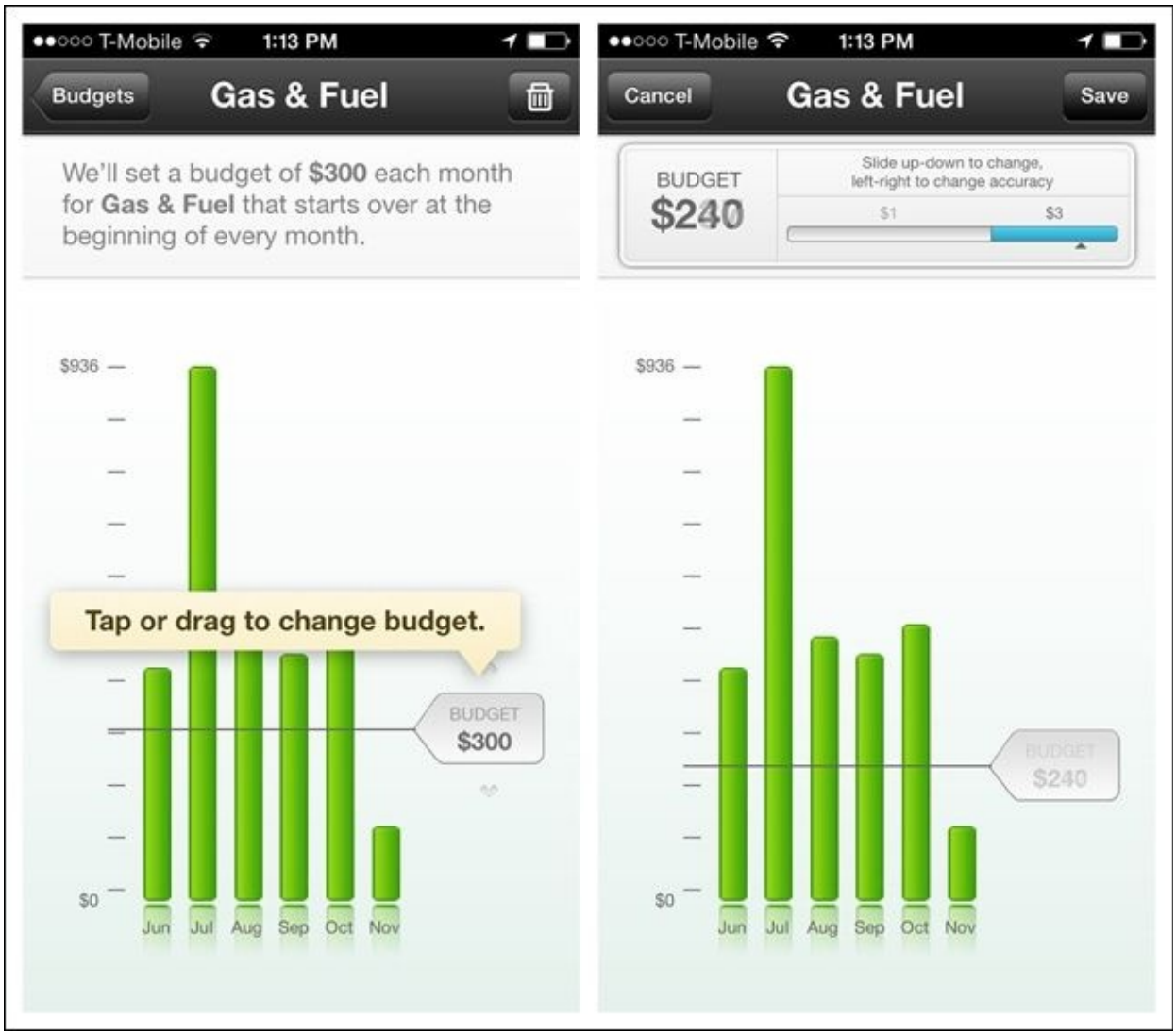



*Figure 6-26. Mint for iOS: interactive chart for setting a budget*

# Dashboard

This pattern is also covered in Chapter 1. A Dashboard can be used for primary navigation, as in Withings and Mint, which each use a Dashboard as the hub for navigating to specific modules.


*Figure 6-27. Withings for Android and Mint for iOS: Dashboard as primary navigation*

Gekoboard offers a customizable dashboard for tracking sales data. Widgets can be added or removed, and reordered.

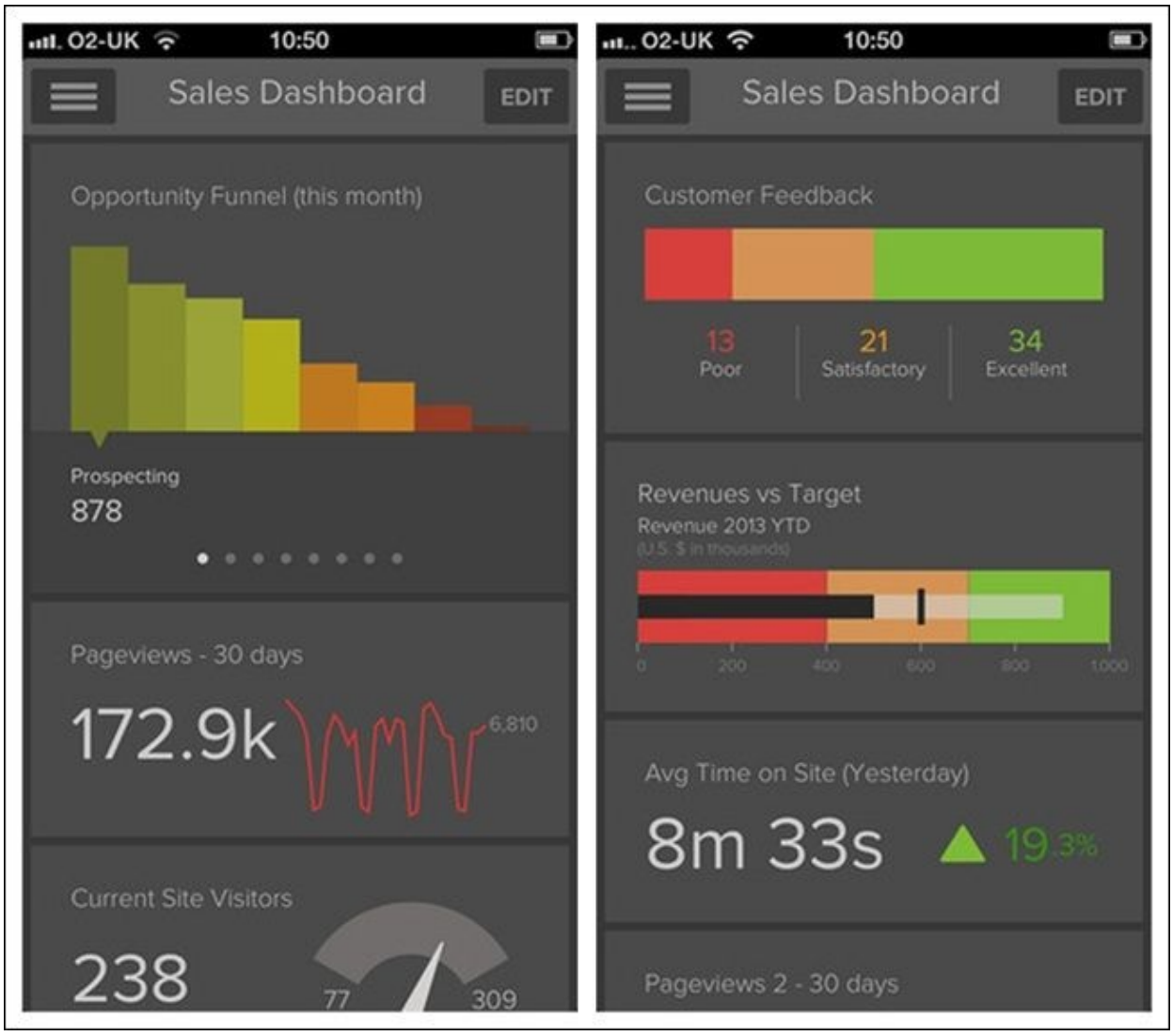



*Figure 6-28. Gekoboard for iOS: customizable Dashboard*

The Dashboard also can be used for secondary navigation, as in RunKeeper for iOS and Lose It! for Android, which use multiple stacked charts to show progress against a set of goals.

Then there's Analytiks, which uses Dashboards for primary *and* secondary navigation. The home screen shows the website's key performance indicators, and you can navigate to a second dashboard with detailed metrics around demographics, browsers, audience, and time on site.

**NOTE**

Use the Dashboard pattern if displaying multiple charts together helps tell an overarching story. Avoid it if it's just consolidating disparate, unrelated data.

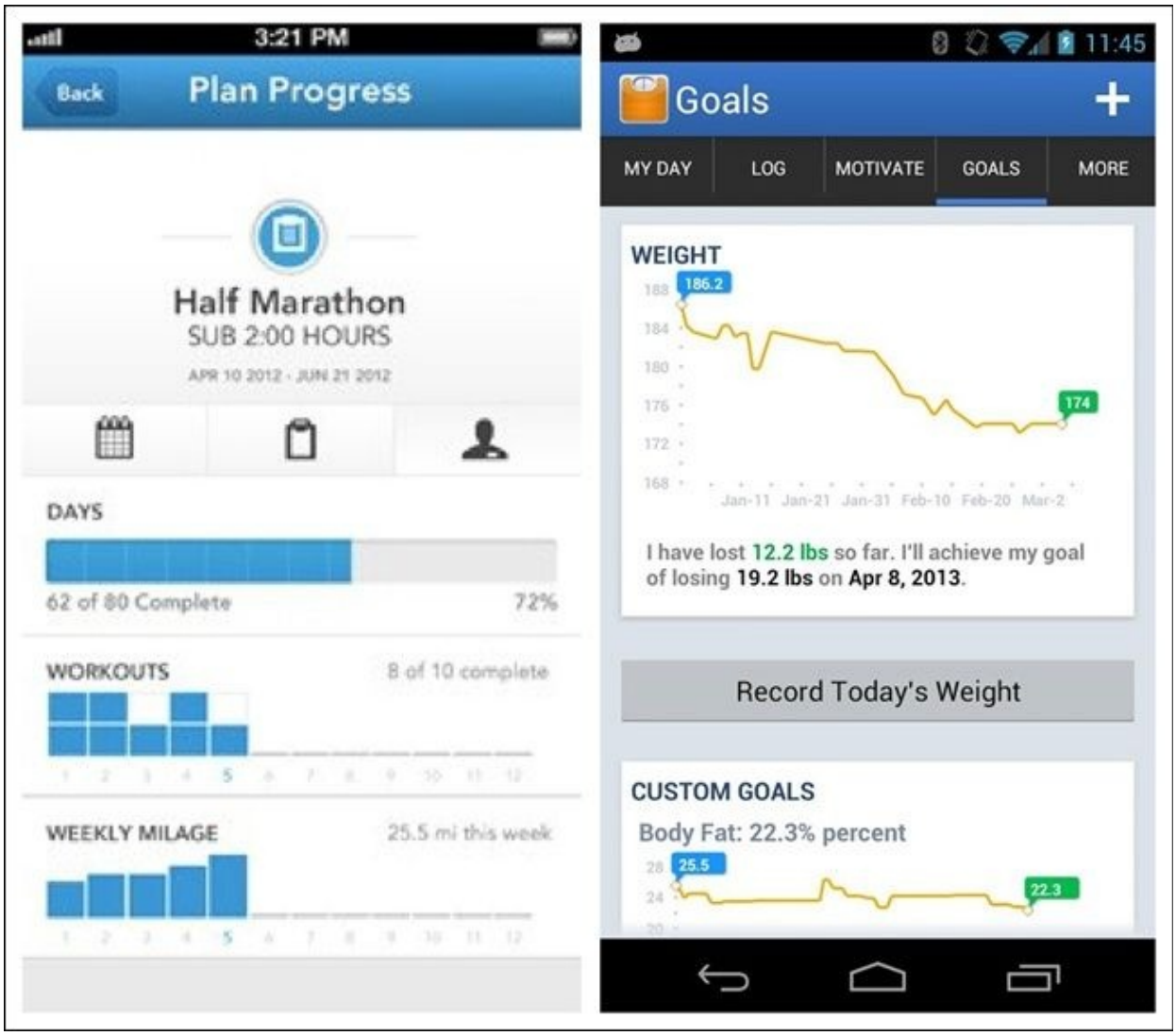



*Figure 6-29. RunKeeper for iOS and Lose It! for Android: Dashboard as secondary navigation*

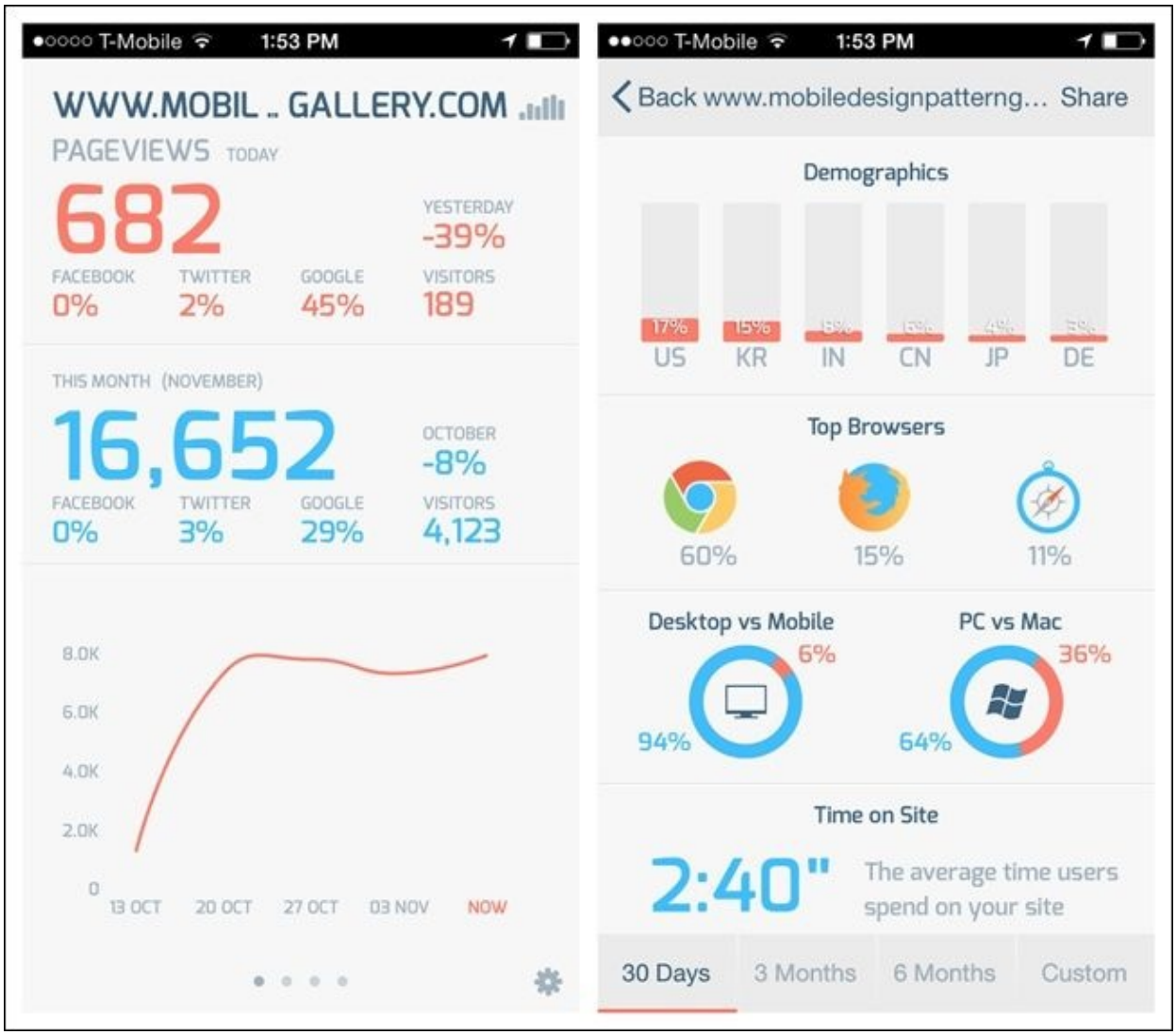



*Figure 6-30. Analytiks for iOS: well-designed Dashboards as primary and secondary navigation*

# Zoom

A chart might be only one of many elements on the screen, making it difficult to read. It is common practice to invite the user to rotate her phone to get a full-screen, landscape view. The title and navigation elements are typically hidden to give the chart as much space as possible. Rotating the device back to portrait mode restores the navigation and other controls.

It is important to provide an invitation to rotate the device for the immersive chart view. NASDAQ QMX uses a little illustration of a rotated phone, whereas SigFig shows a prompt to rotate the device when the chart is tapped.

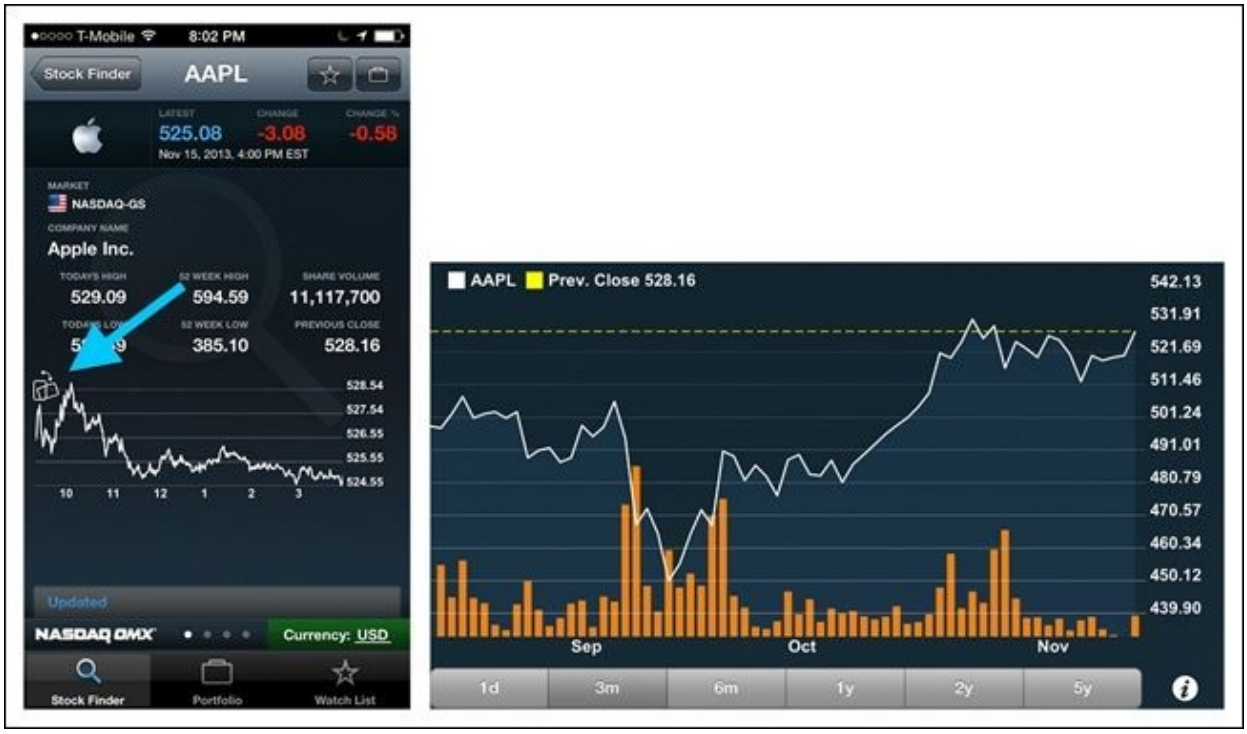



*Figure 6-31. NASDAQ QMX for iOS: illustration invites user to rotate phone for detailed view*

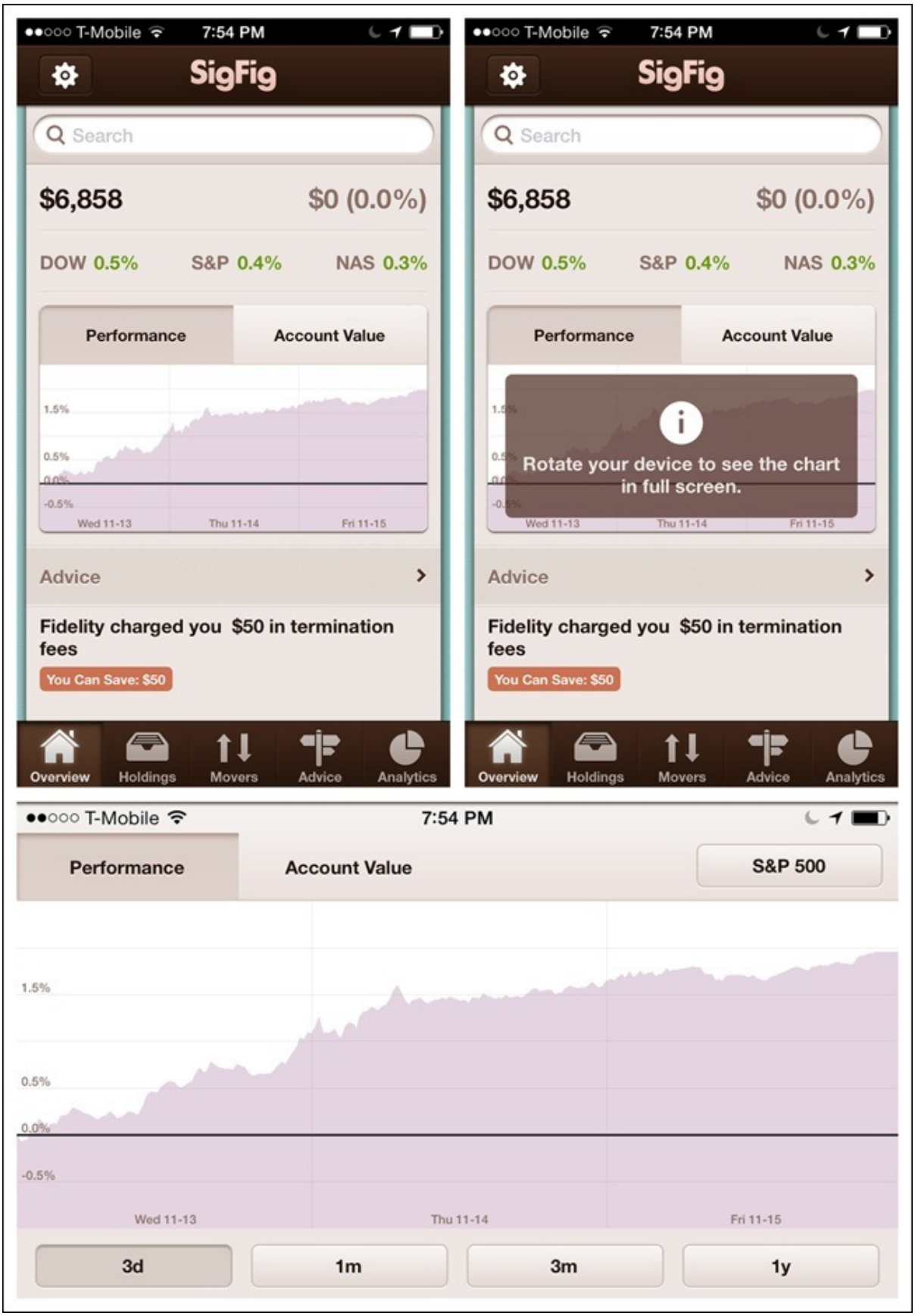

●●○○○ T-Mobile
7:54 PM
SigFig
Search
$6,858
$0 (0.0%)
DOW 0.5%
S&P 0.4%
NAS 0.3%
Performance
Account Value
1.5%
0.5%
0.0%
-0.5%
Wed 11-13
Thu 11-14
Fri 11-15
Advice
Fidelity charged you $50 in termination fees
You Can Save: $50
Overview
Holdings
Movers
Advice
Analytics
Rotate your device to see the chart in full screen.
S&P 500
3d
1m
3m
1y

*Figure 6-32. SigFig for iOS: tapping chart brings up prompt to rotate for detailed view*

Once the chart is in landscape mode, the user may still want to look more closely at a specific time frame. In the example from Yahoo!, my first view covered two weeks, then zoomed into one week, then two days, and finally from 12:30 p.m. to 2:20 p.m. on a single day. To see a broader time frame again, I can either tap one of the time filters or pinch the screen.

**NOTE**

Invite the user to rotate to landscape for a full-screen, more detailed view of the chart; automatically restore navigation when the device rotates back to portrait. Try gestures like pinch to zoom or panning for expanding or narrowing a chart's focus.

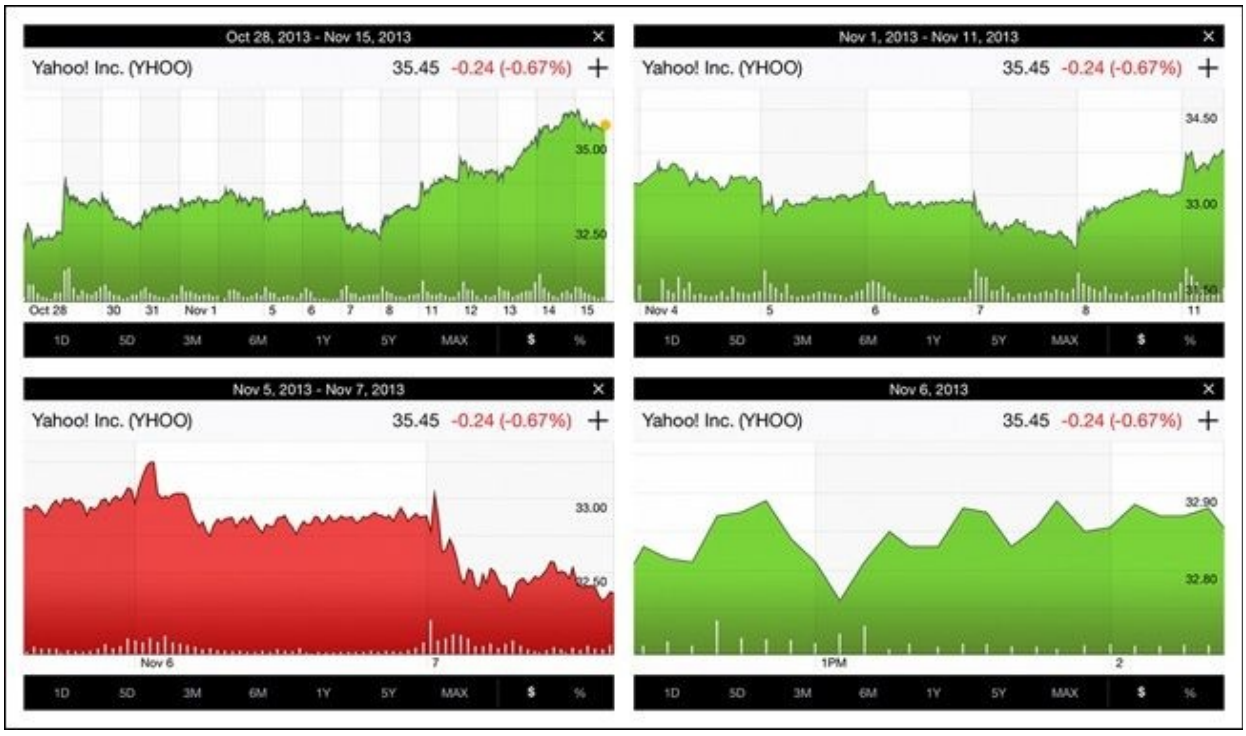



*Figure 6-33. Yahoo! for iOS: pinch/zoom in landscape view allows for adjusting time frame focus*

## Sparklines

The Sparkline, also called a microchart, is credited to visual information design guru Edward Tufte, who defines it as “a small intense, simple, word-sized graphic.”

Sparklines are particularly well suited for mobile since they provide an overview of a trend or variation in the data without using much space. They can be, Tufte says, “embedded in a sentence, table, headline, map, spreadsheet, graphic.” I think of them as bite-sized charts.

In the Analytics Tiles for iOS example, the top two tiles use Sparklines to communicate the trend in page views and visits. The tiles are an entry point to drill down into the larger and more detailed charts. Similarly, Gaug.es uses one Sparkline per row to communicate the trend in the data.

In the example from Fitbit, a bullet-style Sparkline indicates progress against a preset goal. And in Noom, another health app, micro–donut chart Sparklines indicate the balance of each meal.

**NOTE**

Use Sparklines to provide a quick visual hit of information, but always validate their clarity in user testing. Consider using Sparklines as the entry point to a Drill Down into a more detailed view.


*Figure 6-34. Analytics Tiles for iOS and Gaug.es for Android: Sparkline examples*

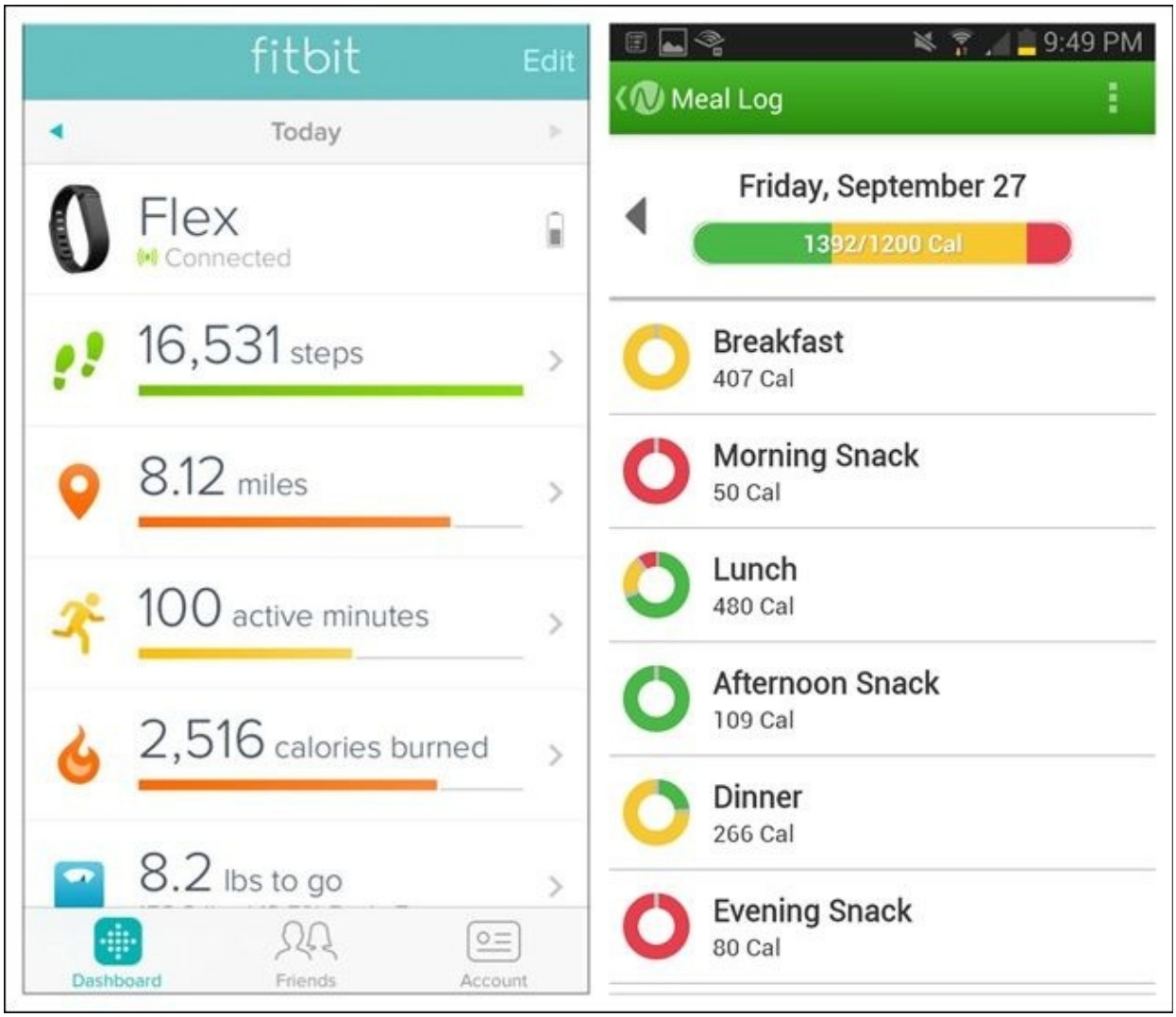



*Figure 6-35. Fitbit for iOS and Noom for Android: more examples of Sparklines*

# Integrated Legend

Another technique to save space is to integrate the legend into either the title of the chart or the data table below. Personal Capital for iOS provides some good examples.

> **NOTE**
>
> Integrated Legends can save precious screen real estate, but consider accessibility guidelines before using this pattern instead of a standalone legend.

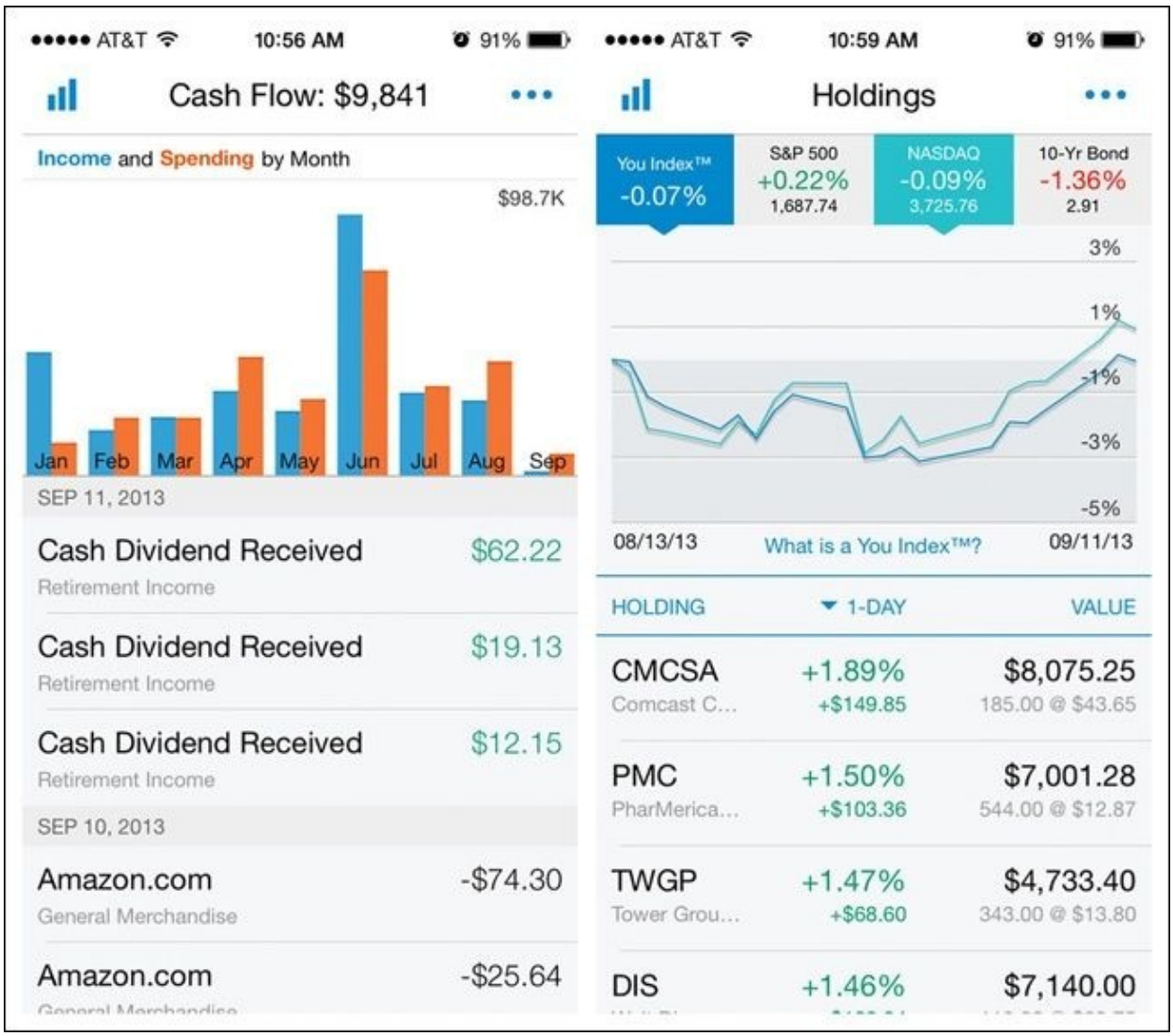



*Figure 6-36. Personal Capital for iOS: Integrated Legends in title (left) and data table (right)*

# Thresholds

Many charts use this pattern to communicate if data is in or out of range, or on or off target. Thresholds can be communicated by use of color, bands, or lines on charts.

In Battery Graph Notification for Android and the MySugr Diabetes Companion app for iOS, the common red/yellow/green color palette is used to indicate if the data is in or out of the acceptable or target range.

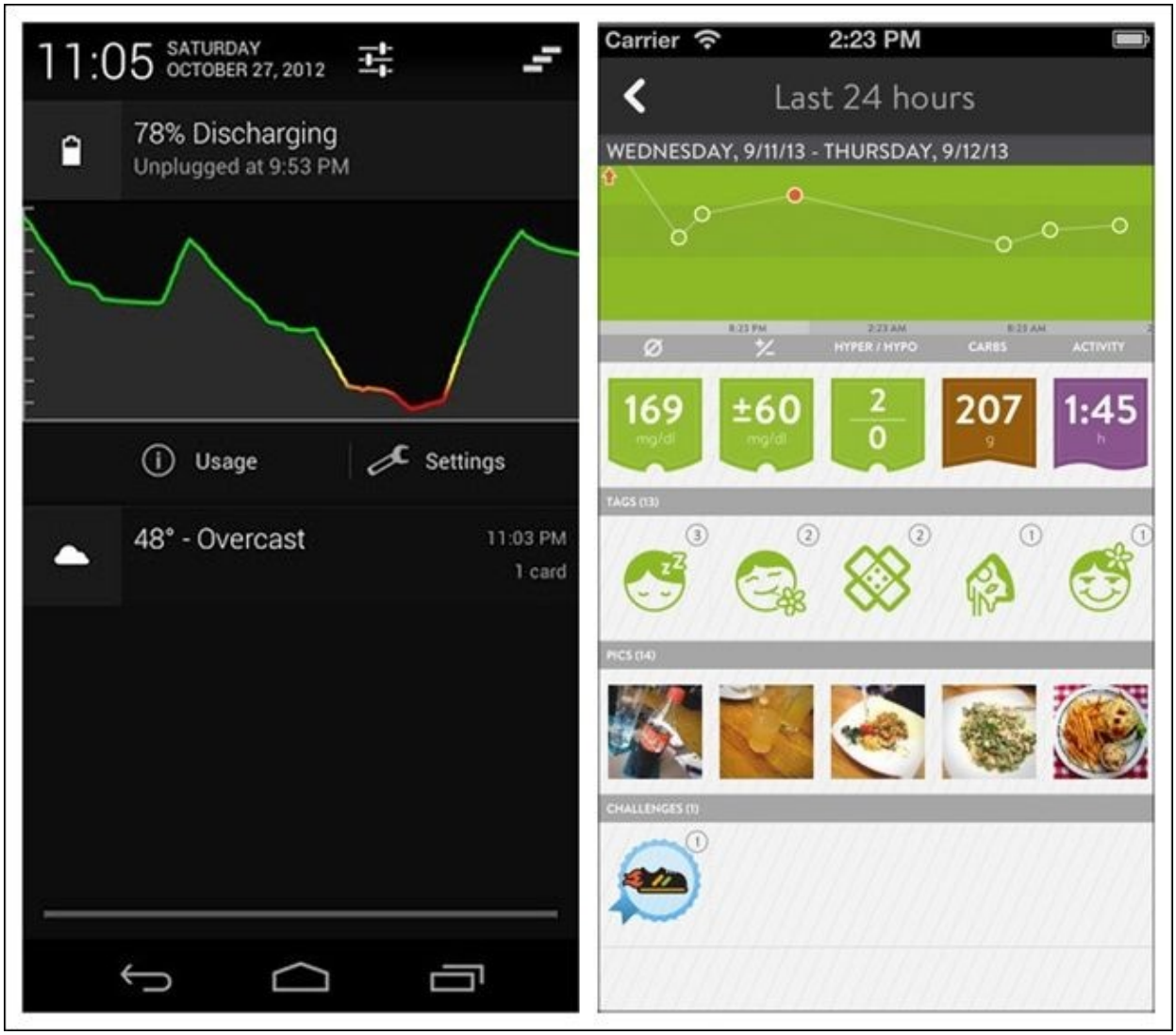



*Figure 6-37. Battery Graph Notification for Android and MySugr for iOS: colors indicate data in or out of target range*

As examples of bands or lines to indicate Thresholds, in Data Usage, the orange line indicates a warning, and the red line at the top would obviously be critical;

Anthony Lagoon's app for monitoring diabetes shows a band for the normal range, and any data points falling above or below that are considered problematic.


*Figure 6-38. Data Usage for Android and Diabetes concept app by Anthony Lagoon: Threshold line and band, respectively*

Bullet chart examples from Toshl and Lose It! use dotted and colored lines, respectively, to indicate Thresholds.

**NOTE**

Thresholds are particularly useful for marking personal user goals. Consider making the Threshold marker interactive (i.e., adjustable within the chart).

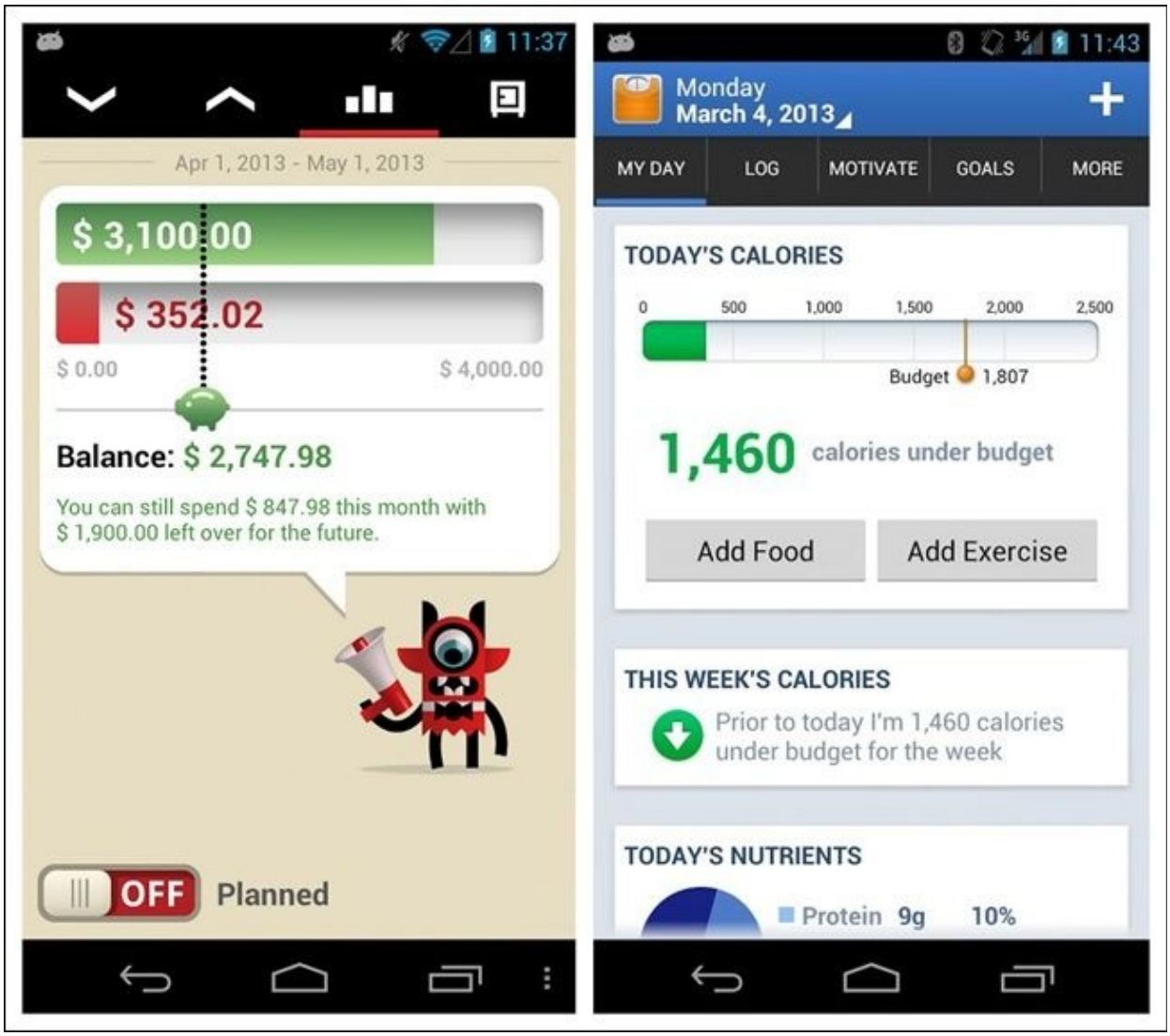



*Figure 6-39. Toshl for Windows Phone and Lose It! for Android: more Threshold designs*

# Pivot Table

Pivot Tables are small summary tables within larger tables. In popular desktop and web spreadsheet applications, for example, the user can build a Pivot Table by dragging and dropping the values to be included in the summary. By moving these values around, the user can “pivot” the table to display the data in multiple ways.

Roambi has developed a simple filtering interface for exploring data in a style similar to the Pivot Table. The y-axis remains fixed; however, any dimension can be selected for the x-axis (the selection is highlighted in an orange frame). For each dimension, a specific category can also be selected. The example here shows how easy it is to explore the quarterly profit for all cities, and then quickly refine the chart to just see this info for a specific product line: books. With one more selection, we can narrow this down to just the online channel.


*Figure 6-40. Roambi for iOS 6: Pivot Table–style filtering*

## Pulling It All Together

Now let’s look at some examples from apps that really use charts well.

The Nike+ FuelBand app does a fantastic job of presenting clean and clear data visualizations to motivate users. Its designers selected the best chart types for communicating progress on different goals, and avoided the temptation to overload the screens with chart junk. Each chart is clearly titled and the axes are clearly labeled. The app establishes a Threshold color palette and uses it consistently throughout. Regardless of the chart type, the low end of the progress and performance spectrum is always shown in red, and the top end is always green.


LTE 10:04 PM
TUESDAY, OCT 15
New York, NY
3172
3500
THIS WEEK
3 Goals in a row starts a streak.
M T W T F S S
LTE 10:04 PM
ACTIVITY
DAY WEEK MONTH YEAR
TUESDAY, OCT 15
4387
Avg Tue: 3435
12am 12pm 12am
998 Calories
9595 Steps
10:35p Goal Hit
7 Hours Won
NikeFuel Breakdown
LATE NIGHT 134
MORNING 846
AFTERNOON 543
EVENING 325
Peak Hour: 8-9am, 324
LTE 10:04 PM
Edit
Done
TENNIS
658
1:30pm
2:38pm
01:08:00 Duration
22/min Intensity
Took on Lola today... and STILL remain queen of the court!
– with Lola O. at Neighborhood Swim & Tennis
SHARE
9:41 AM 100%
YOUR MOVEMENT
MONDAY, SEPTEMBER 2 - SUNDAY, SEPTEMBER 8
This Week's Daily Avg: 2234
4000
3000
2000
1000
Avg
S M T W T F S S M
17,438 NIKEFUEL THIS WEEK
THAT'S 4% MORE THAN YOU EARNED LAST WEEK.

*Figure 6-41. Nike+ FuelBand for iOS: clean and clear data visualizations throughout*

Weathertron is another fine example of well-designed data visualization. While the first-time tutorial experience could benefit from a redesign featuring embedded invitations (as outlined in Chapter 8), the app UX itself is simple and intuitive. The compact bar chart in the drawer accessed at lower right, for example, is the perfect way to show the highs and lows for the seven-day forecast.

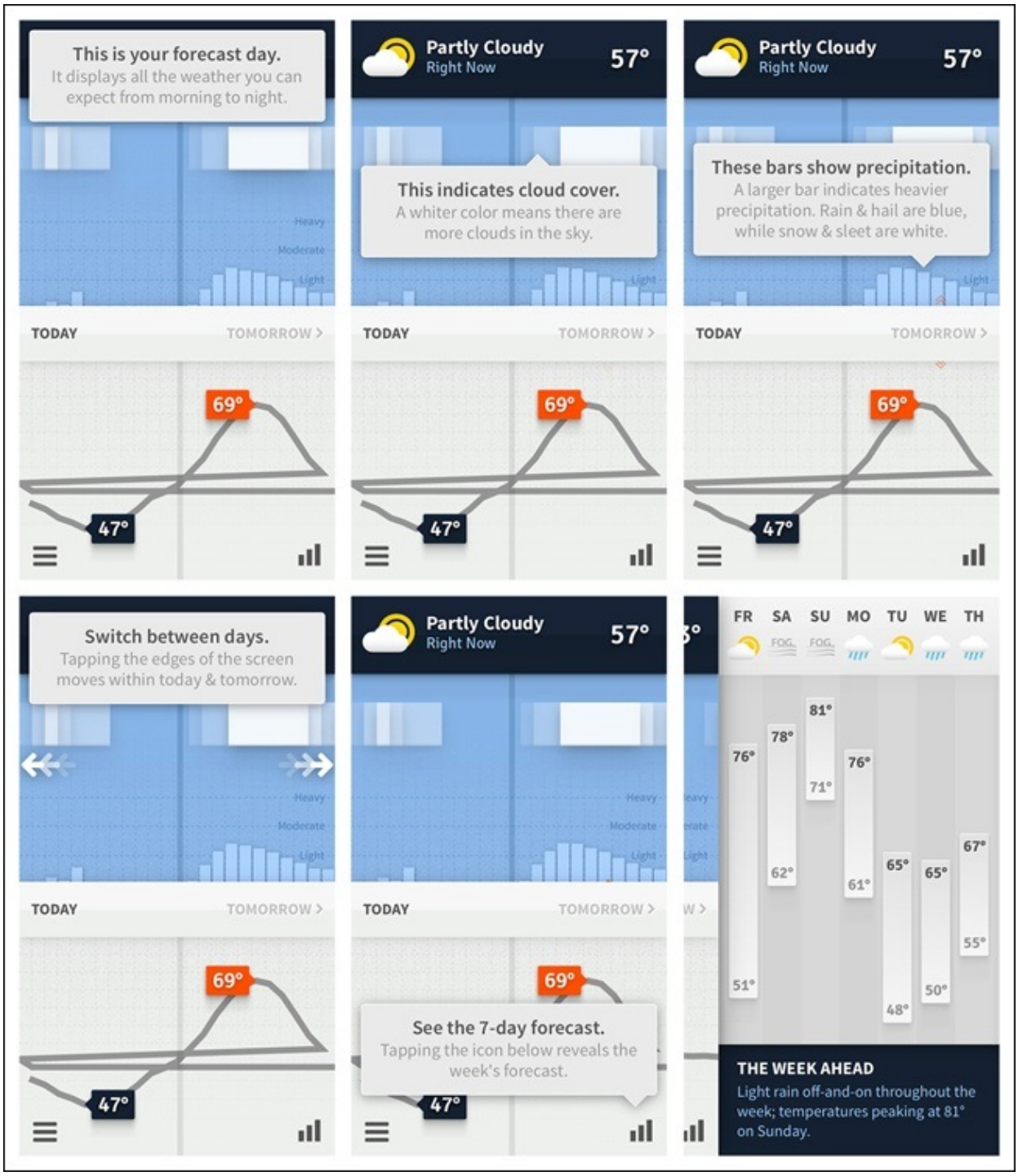



*Figure 6-42. Weathertron for iOS: well-designed, simple, and intuitive data visualizations*

# Chapter 7. Tutorials and Invitations


*Tutorial Rules*

Use Less Text, No Frontloading, Make It Rewarding, Reinforce Learning, Listen to Your Users

*Invitation Patterns*

Tips, 1st Time Through, Persistent Invitations, Discoverable Invitations

In the first edition of *Mobile Design Pattern Gallery*, I covered eight common invitation patterns. But we now know that even though these patterns are commonplace, few of them are very useful. When I had the opportunity to meet some of the leading designers at Intuit in 2012, they shared findings revealing that over half of these patterns tested poorly with their users. This corresponded with the poor results my team and I had recently gotten testing these patterns in a variety of applications, including the RetailMeNot iOS and Android apps.

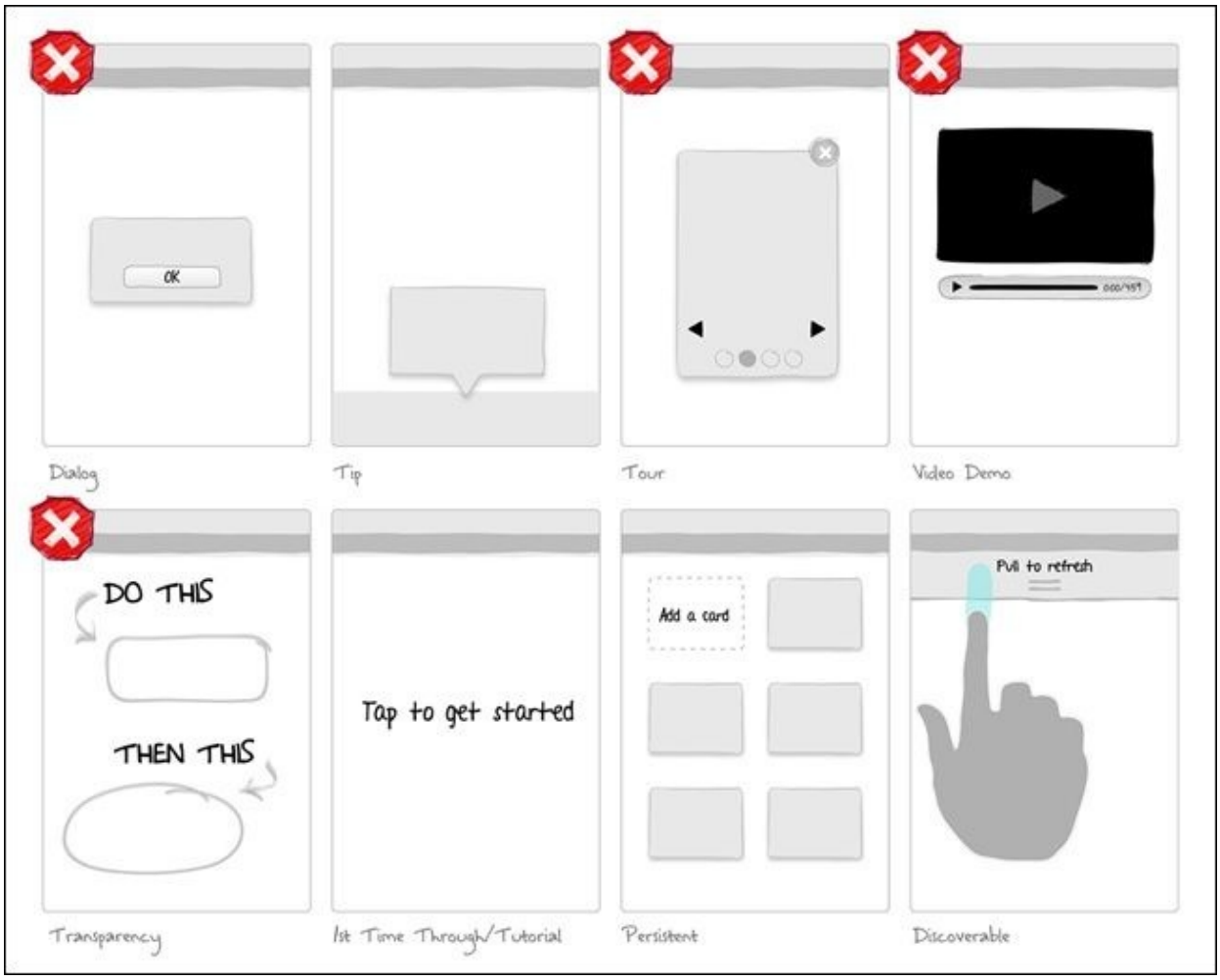



*Figure 7-1. First edition patterns for invitations*

In a nutshell, testing showed that users tend to skip or otherwise ignore Dialogs, Tours, Video Demos, and Transparencies. At best, users find them a minor inconvenience. At worst, they can cause significant aggravation to new users who are trying to get into the app. As one RetailMeNot test participant put it, “I just want to get in the app and start exploring.”

So with this new knowledge, we begin a new chapter. The first half of the chapter is devoted to explaining how to create an effective “first-time through” experience, also known as a tutorial or onboarding. The second part of the chapter looks at the invitation patterns that have been proven to work: Tips, 1st Time Through, Persistent Invitations, and Discoverable Invitations.

## Tutorial Rules

So why don't Dialogs, Tours, Transparencies, and Video Demos work well for onboarding new users? I turned to the field of game design for answers. Game designers have always known that you can't drop new players into the middle of a firefight and expect them to enjoy the experience. Most players would be dead before they figured out how to fire their weapons and fight back.

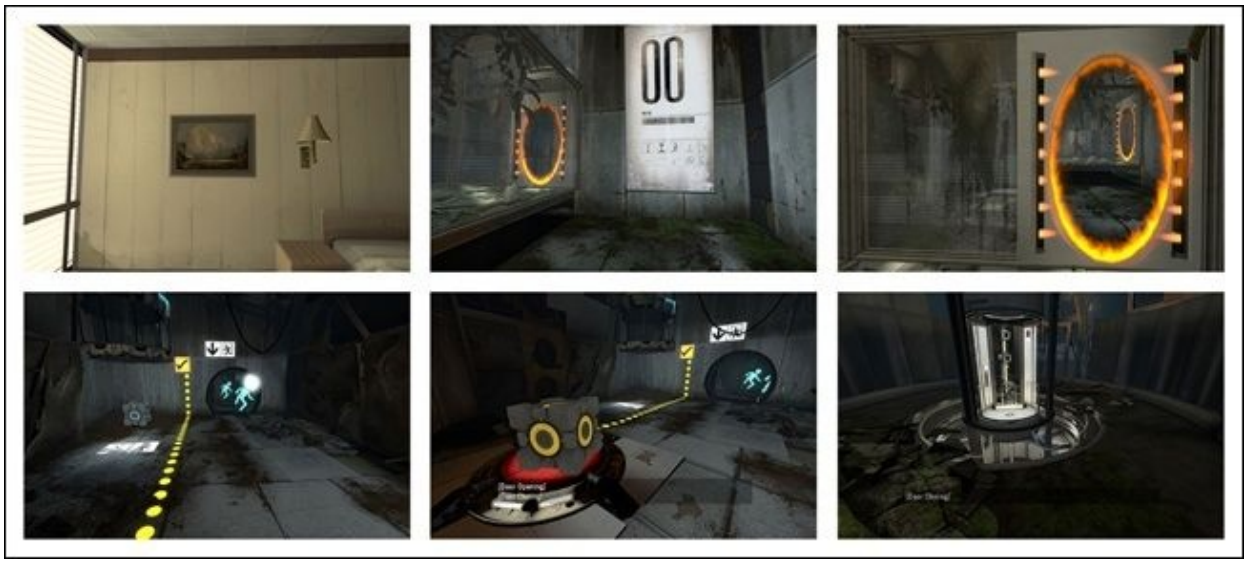


*Figure 7-2. Portal offers a safe environment for players to figure out the controls while advancing in the game*

In game design, some methods of driving deeper engagement are much more effective than others. The same holds true for mobile apps. While the stakes may not be (virtual) life or death, the frustration users experience when they don't know how or what to do is the same. And when that happens to too many of your users, it's game over for your app.

The Extra Credits online video series thoughtfully looks at various aspects of the gaming world from an insider's perspective. Its design team created a fantastic video called Tutorials 101 (*http://www.youtube.com/watch?v=BCPcn-Q5nKE*) that I believe every app designer would do well to watch. (Get past the annoying chipmunk voice narration — it's worth it.)

Tutorials 101 lays out some basic rules for crafting an effective game tutorial:

1. Use less text.
2. No frontloading.
3. Make it fun.

4. Reinforce learning through play.
5. Listen to your players.

Instead of design patterns, we're going to use these rules as our outline for creating a great tutorial (after first making some minor modifications as to how they apply to apps). As we look at these rules in depth, it should become apparent why most of the invitation patterns from the previous edition of this book are less effective in helping users to have deeper, richer experiences with our apps.

## Rule #1: Use Less Text

When we want to explain something, words often seem like the easiest tool to use. But when we want to *learn* something, the written word is often a turnoff.

According to Tutorials 101, we should avoid relying only on text because “it kills pacing, it destroys immersion, and it will often be skipped by the players who need the tutorial the most.”

Too much text makes for a “tell, don’t show” approach that works against the strengths of mobile apps. Instead, tutorials should “show, not tell” — they should be interactive, so users learn by doing. If we actually practice an action beforehand, we’re more likely to remember how to do it when we need to than if we’d merely been told how to do it.

The “Use less text” rule doesn’t mean that text is always wrong, or even that fewer words are always better. In general, it means that when you seek to invite deeper engagement, you should seek first to use less text and more interactivity.

Following are a number of comparisons between apps that break this rule and apps that embody the right approach. The first comparison, for instance, looks at two apps for foodies. Ness combines customization screens with a tutorial that tells rather than shows, creating a lengthy flow before the user even gets into the app. With Foodspotting, on the other hand, the user gets right down to business. With only a few calls to action on the screen, it is easy to start engaging — no tutorial needed.

The other examples similarly illustrate apps that have gotten this right and apps that have been less successful.

### Ness Compared to Foodspotting

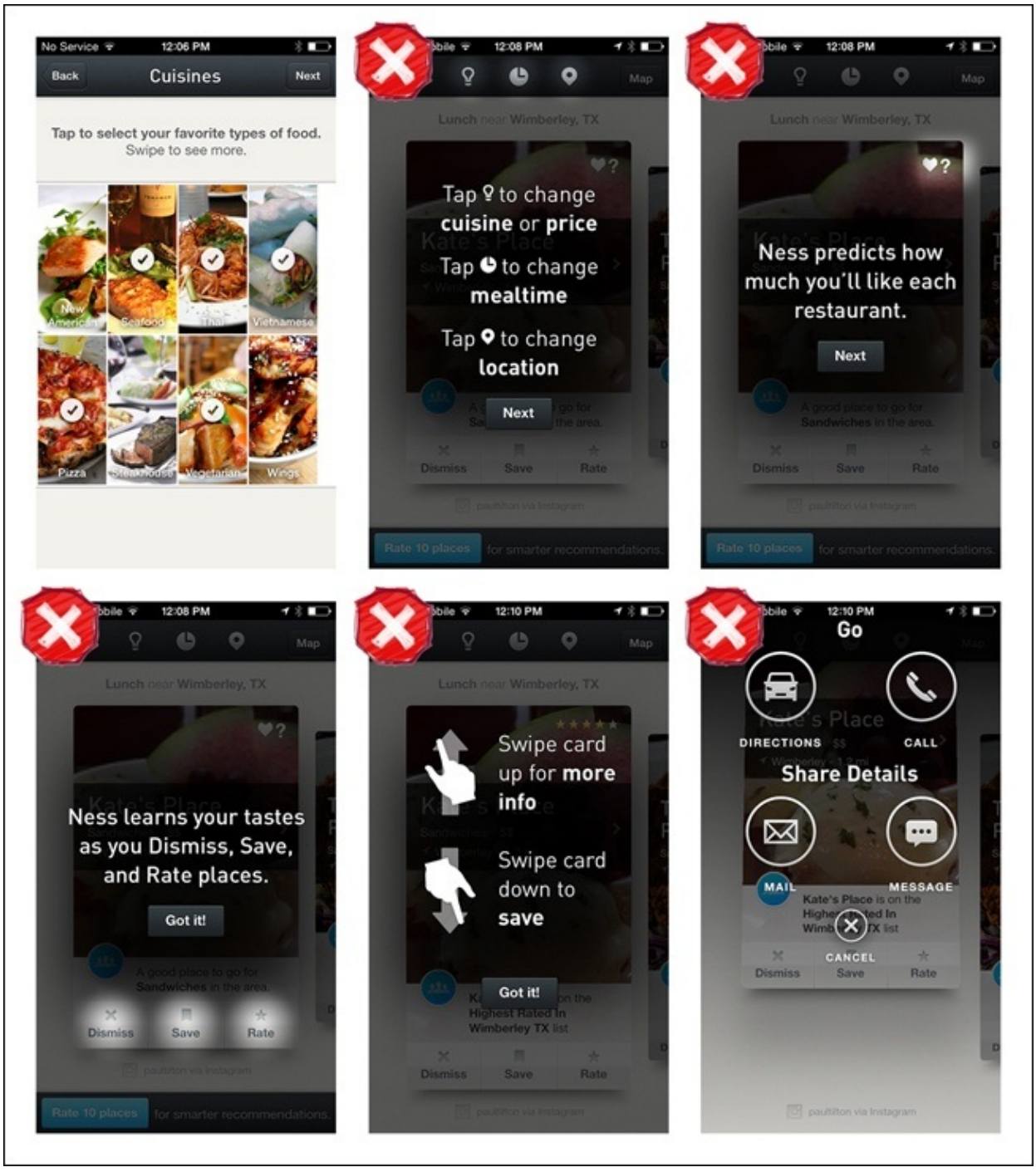



*Figure 7-3. Ness for iOS: note that some prompts explain cause and effect rather than inviting the cause and letting users observe the effect*

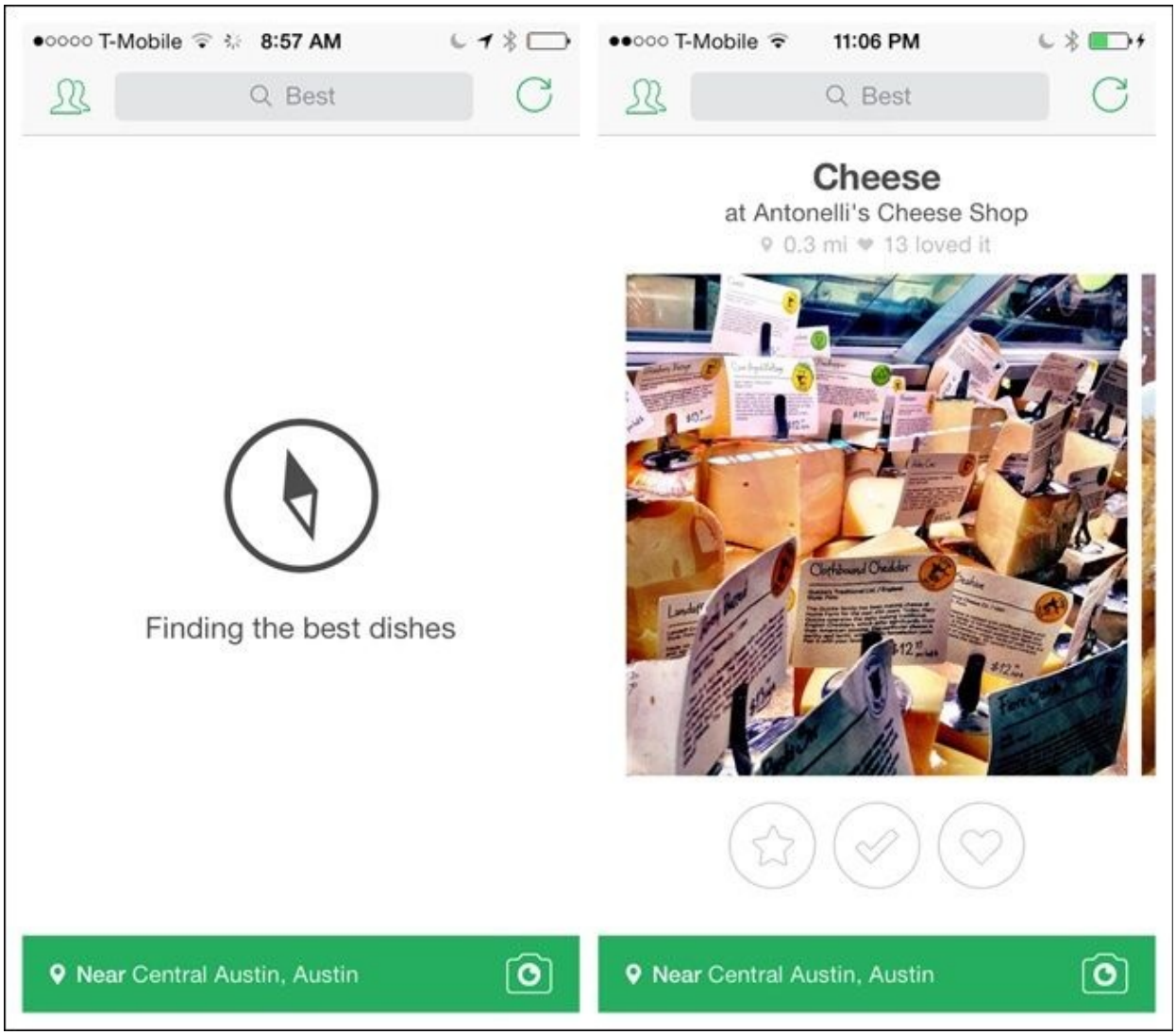



*Figure 7-4. Foodspotting for iOS: an app so simple, it invites deeper engagement without users needing a tutorial*

## Boomerang Compared to Mailbox


*Figure 7-5. Boomerang for Android: too much text (also see "Rule #2, No Frontloading")*

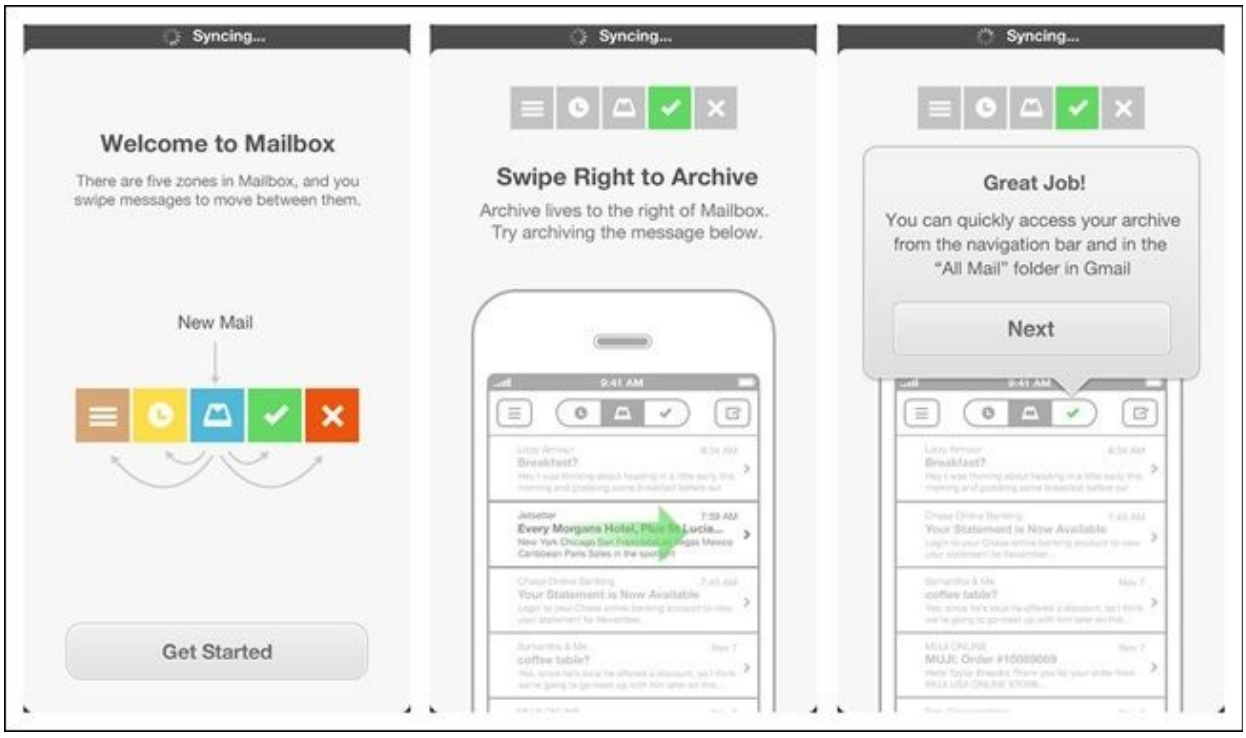



*Figure 7-6. Mailbox for iOS: the tutorial text encourages the user to learn by doing (http://www.youtube.com/watch?v=URd0j5vEsHE)*

## DigiCal Compared to Fantastical

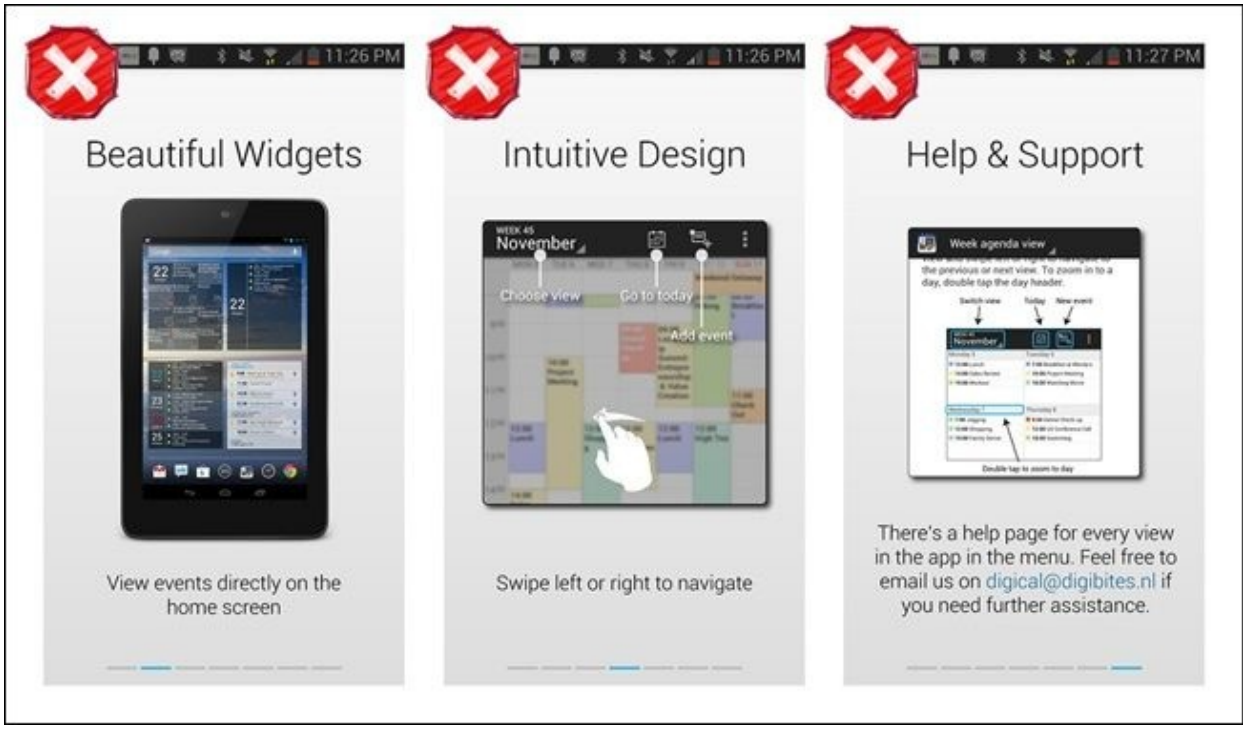



*Figure 7-7. DigiCal for Android: the text is abstracted from the activities it describes*

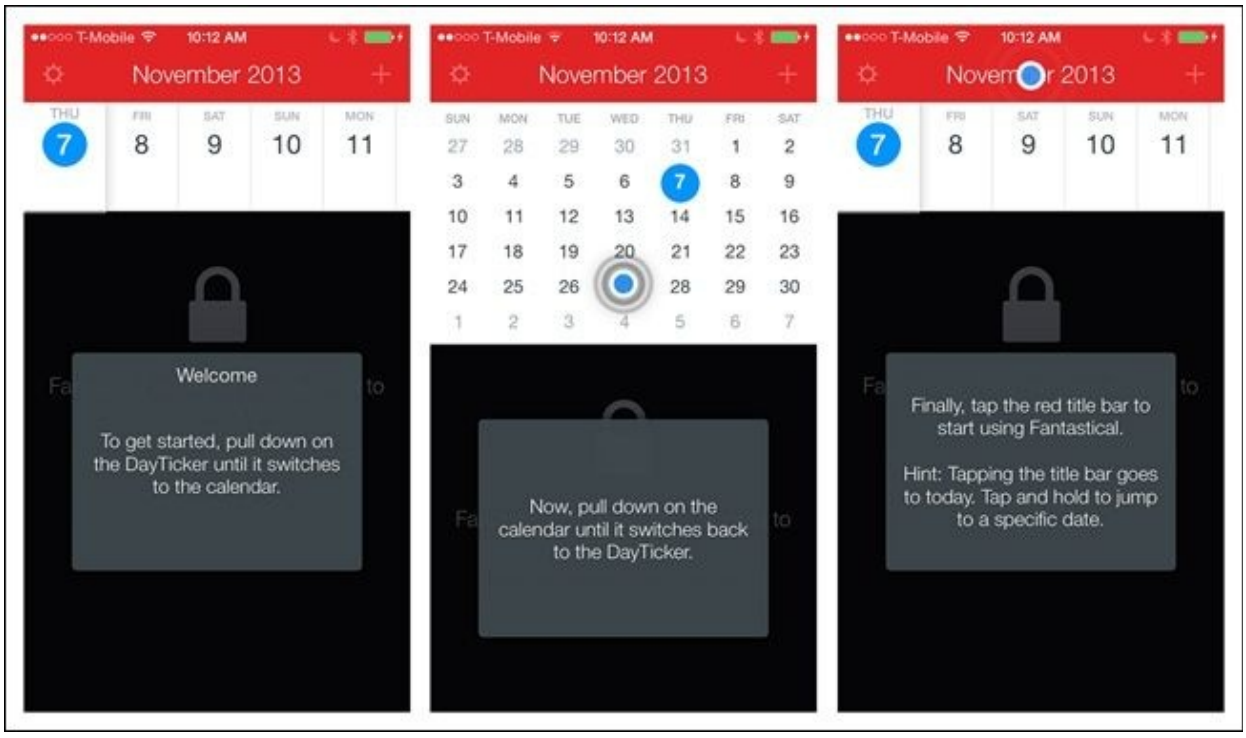



*Figure 7-8. Fantastical for iOS: the tutorial invites the user to work through the gestures he needs to learn (http://www.youtube.com/watch?v=e6Q8FbuNwiQ)*

## Catch Compared to Clear

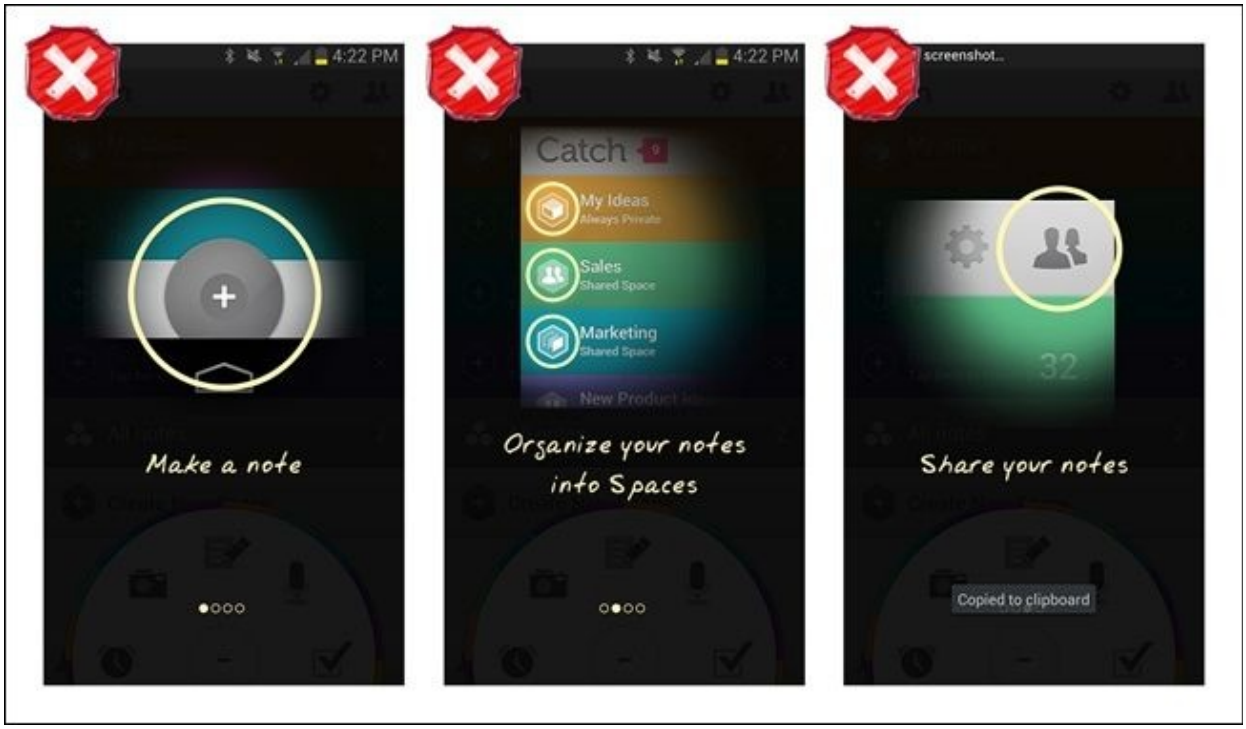



*Figure 7-9. Catch for Android: the tour describes features and actions, but doesn't let the user try them*

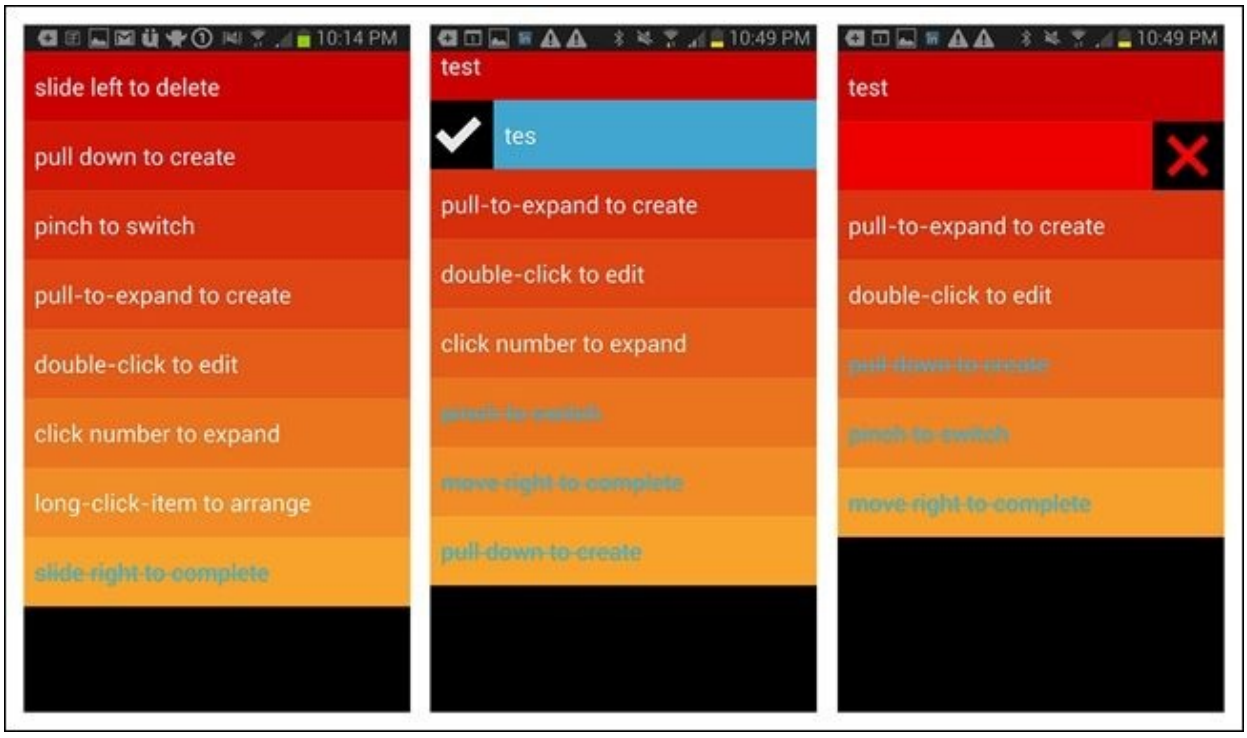



*Figure 7-10. Clear for Android: the default view is preloaded with tasks that the user learns by doing (http://www.youtube.com/watch?v=v3gGTG003_A)*

## SlideStory Compared to Vine

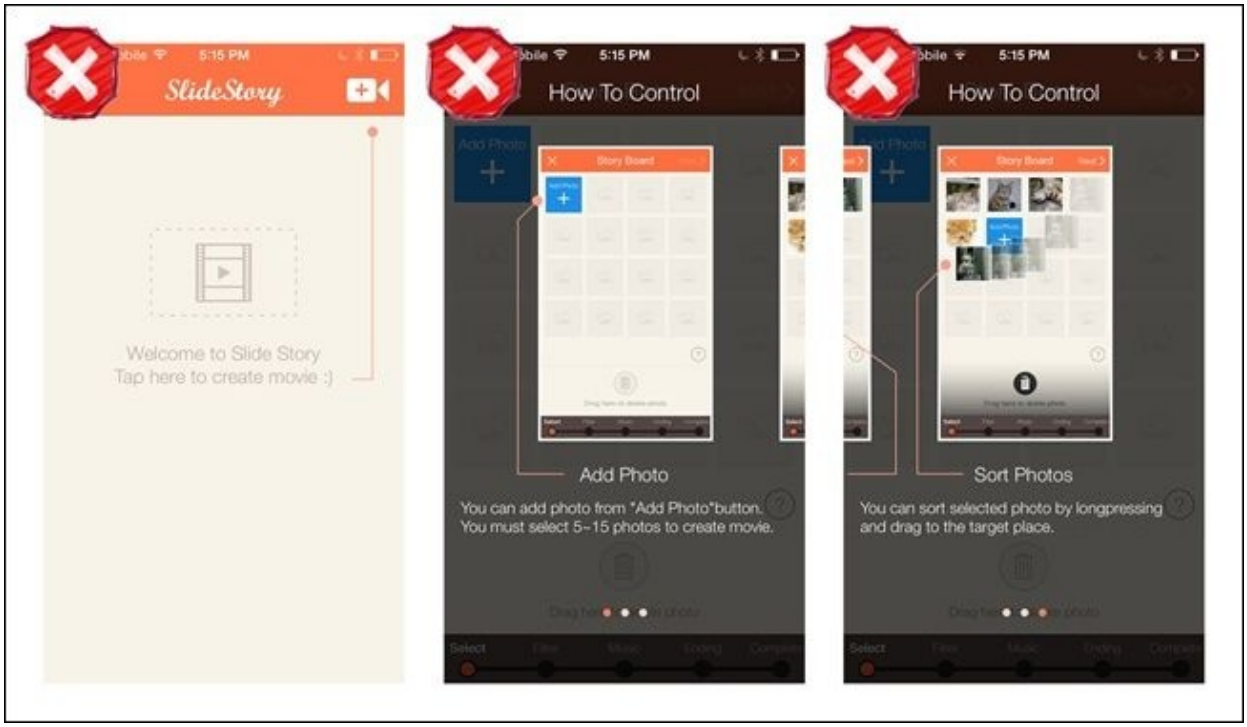



*Figure 7-11. SlideStory for iOS: this tutorial shows and tells, but doesn't let the user do*

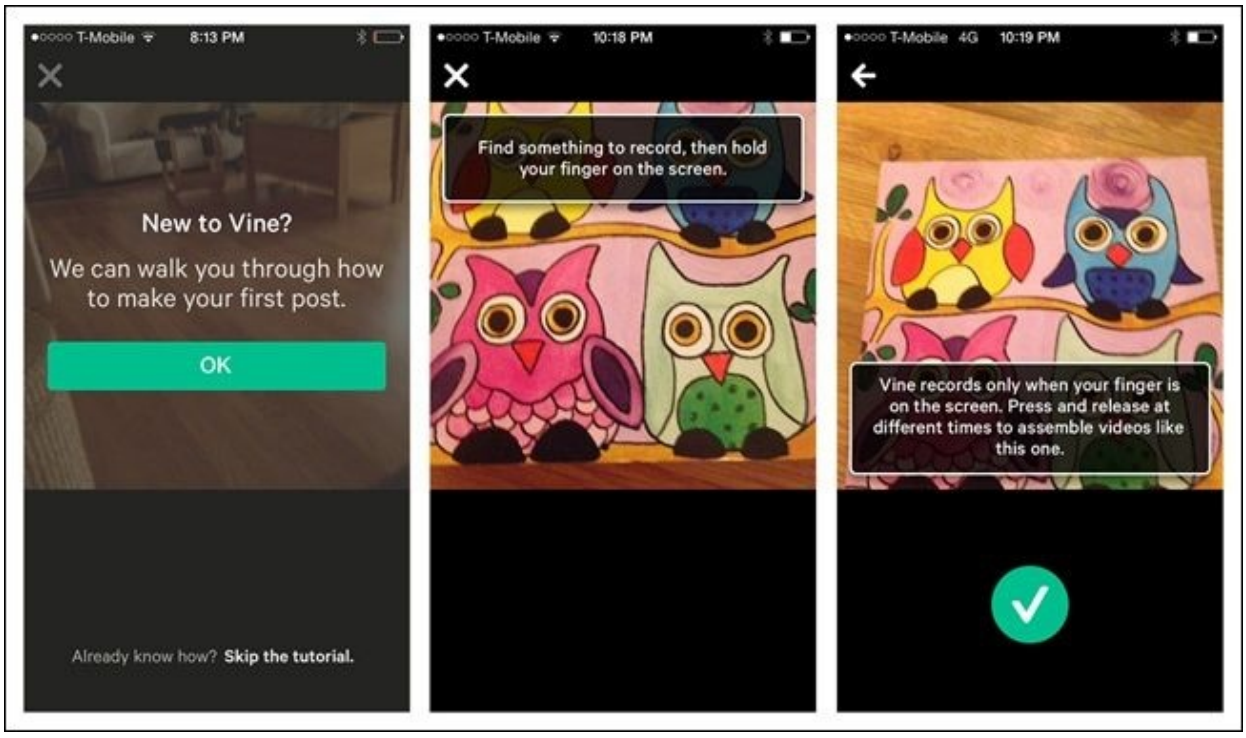



*Figure 7-12. Vine for iOS shows, tells, and encourages users to try for themselves (http://www.youtube.com/watch?v=vBOC9DpC5ns)*

**NOTE**

Resist the urge to use text if there are ways of showing rather than telling. Use text to prompt the "cause" and let the user observe the "effect" by doing.

## Rule #2: No Frontloading

“If you frontload your tutorial and teach the player everything at the beginning,” Extra Credits says in Tutorials 101, “they’ll be overwhelmed with information and undersupplied with engagement.” All you have to do is replace the word *player* with *user* in that sentence, and its relevance is obvious to app designers.

This is a tough lesson for many of us to learn. After all, we work so hard to add useful features to our designs that we naturally want to let the user know everything the app can do, right away.

Resist that urge. As the Tutorials 101 video rightly points out, frontloading will turn off users and cause them to skip your tutorial. And a user who skips your tutorial isn’t likely to get much out of your app or recommend it to others.

Rather than overwhelming users with information that they will probably forget by the time they need it, provide the information in short, easily digestible chunks precisely *when* they need it. Remember, you’re making a first impression here. Wouldn’t you rather make a first impression that leaves users wanting more, rather than less?

By limiting up-front tutorials to just the “getting started” essentials, you allow your users to dive right in and start using your app. Invitations they encounter later on can highlight more advanced features and functions.

Let’s compare some apps that get this right with some that miss the boat.

To start, I’ve paired the app Phoster for iOS with Creative Studio for Windows Phone. Phoster invites initial engagement with a tutorial that tries to cram everything the user will ever need to know onto one chaotic page. Any time the user needs to refer back to remember how to do one thing, she has to wade through instructions for how to do everything.

By contrast, when a user is manipulating a photo in Creative Studio, instructions for completing an action are displayed only once he selects a tool. In the example shown, I opened a photo and chose the Focus effect, and was then guided through a two-step process. Once I previewed my work, I decided to edit it some more, and when I tapped Edit, I was shown another contextually relevant tip.

### Phoster Compared to Creative Studio


*Figure 7-13. Phoster for iOS: though only one page, this chaotic layout presents too much information up front*

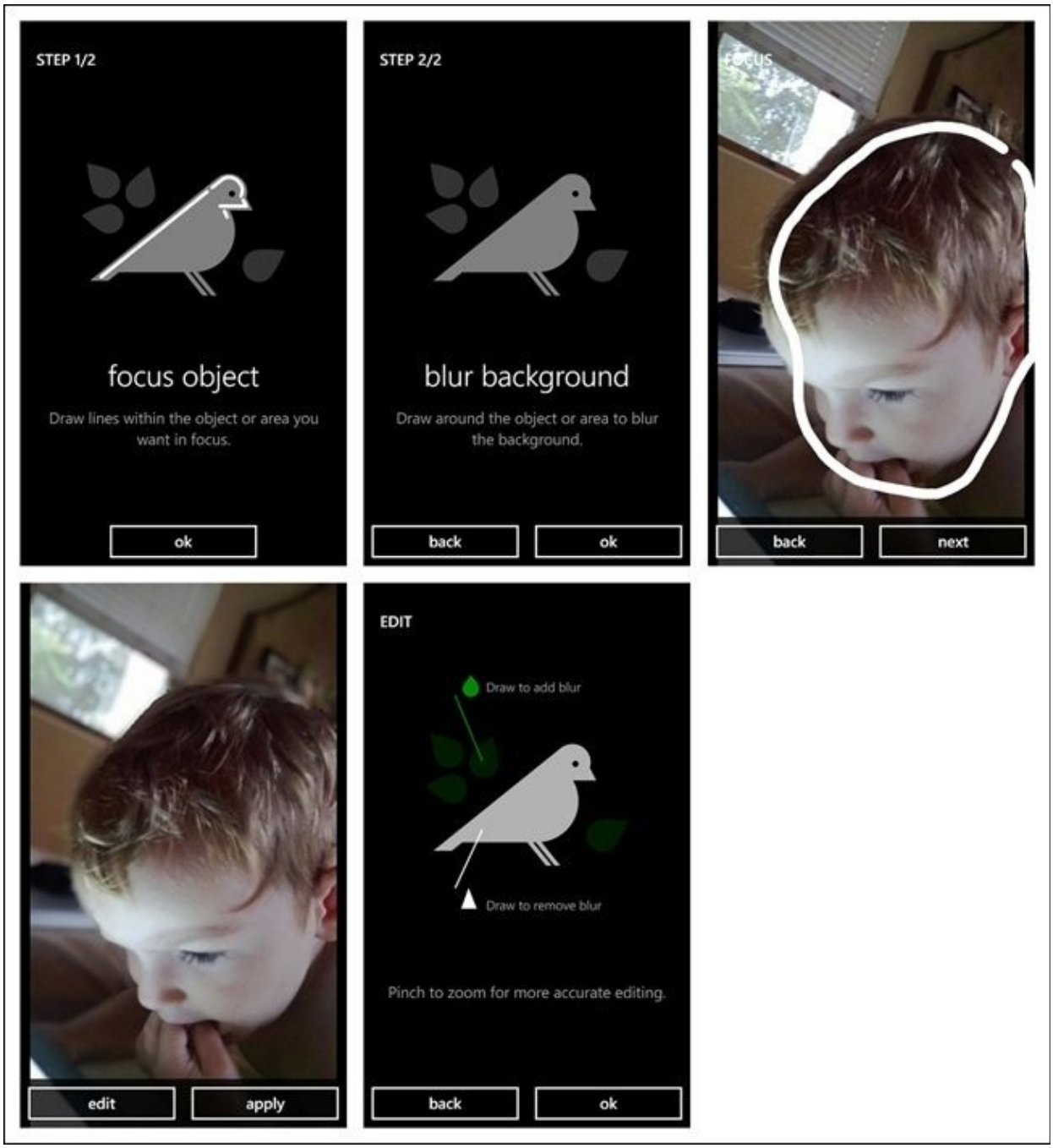



*Figure 7-14. Creative Studio for Windows Phone offers contextual help when and where the user needs it*

## Dooo Compared to Todoist


WELCOME!
DOOO allows you to quickly Create Tasks & Notes with PHOTO, VOICE, DRAWING, LOCATION
Swipe to Begin
TODAY
UNSOLVED
After 24hours, Your tasks will automatically leap sideways to the "UNSOLVED" column. ( You can change the term '24hours~Month' when you add a task )
USING ' ' TO ATTACH THE LONG TERM TASKS IN 'TODAY' COLUMN
EDIT
MOVE
NOTE
LONG PRESS TO "EDIT" "MOVE" "NOTE"
SWIPE RIGHT ON A TASK TO SET PRIORITY
00:27
DONE
USING VOICE RECORDER WHEN YOU'RE DRIVING OR URGENT SITUATION
DOOO PROVIDES MANY OPTIONS FOR INPUTTING THOUGHTS, INCLUDING PHOTOS, LOCATIONS , DRAWINGS, & VOICE RECORDINGS
IDEA
WORK
TO BUY
ORGANIZE YOUR THOUGHTS EFFICIENTLY
RENAME A LIST
DELETE A LIST
DONE
LONG PRESS THE LIST TO EDIT, DELETE AND REORDER
GET STARTED!

*Figure 7-15. Dooo for iOS: overwhelming frontloaded information goes on and on (11 pages total)*

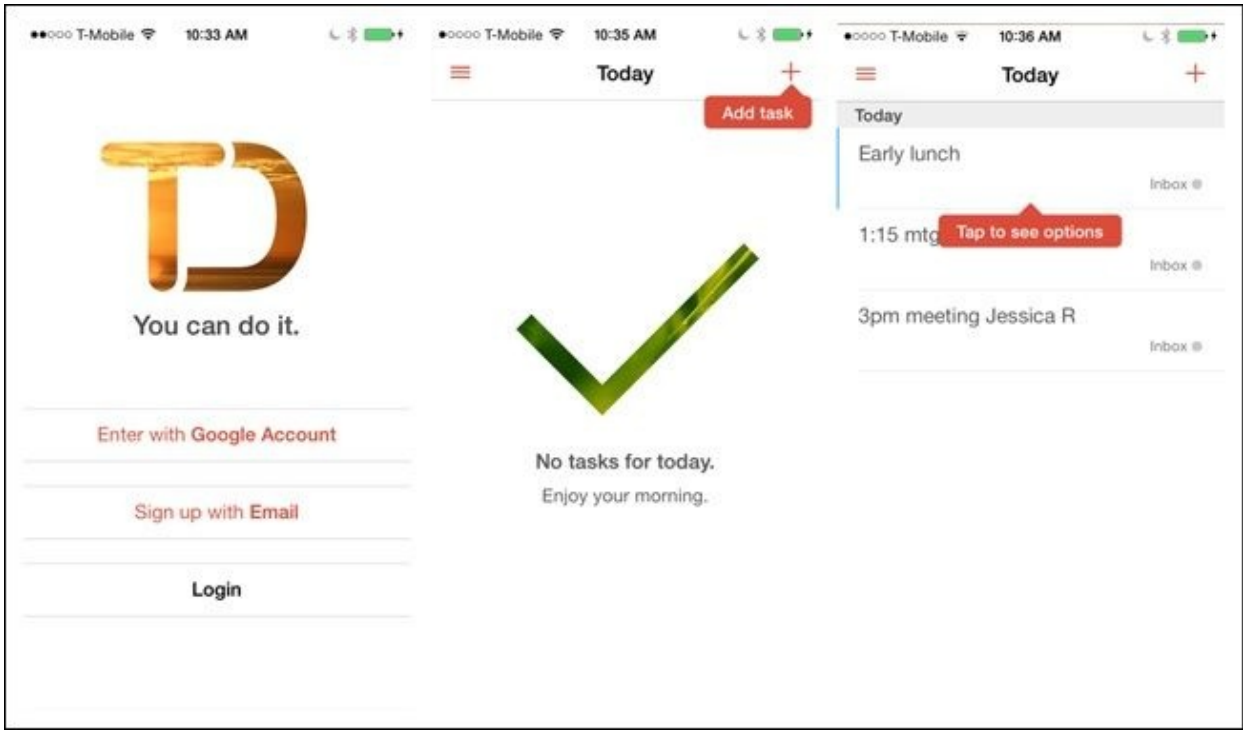



*Figure 7-16. Todoist for iOS: a tip invites the user to add a first task, and then another prompt introduces the options menu*

## Buy Me a Pie! Compared to OneNote

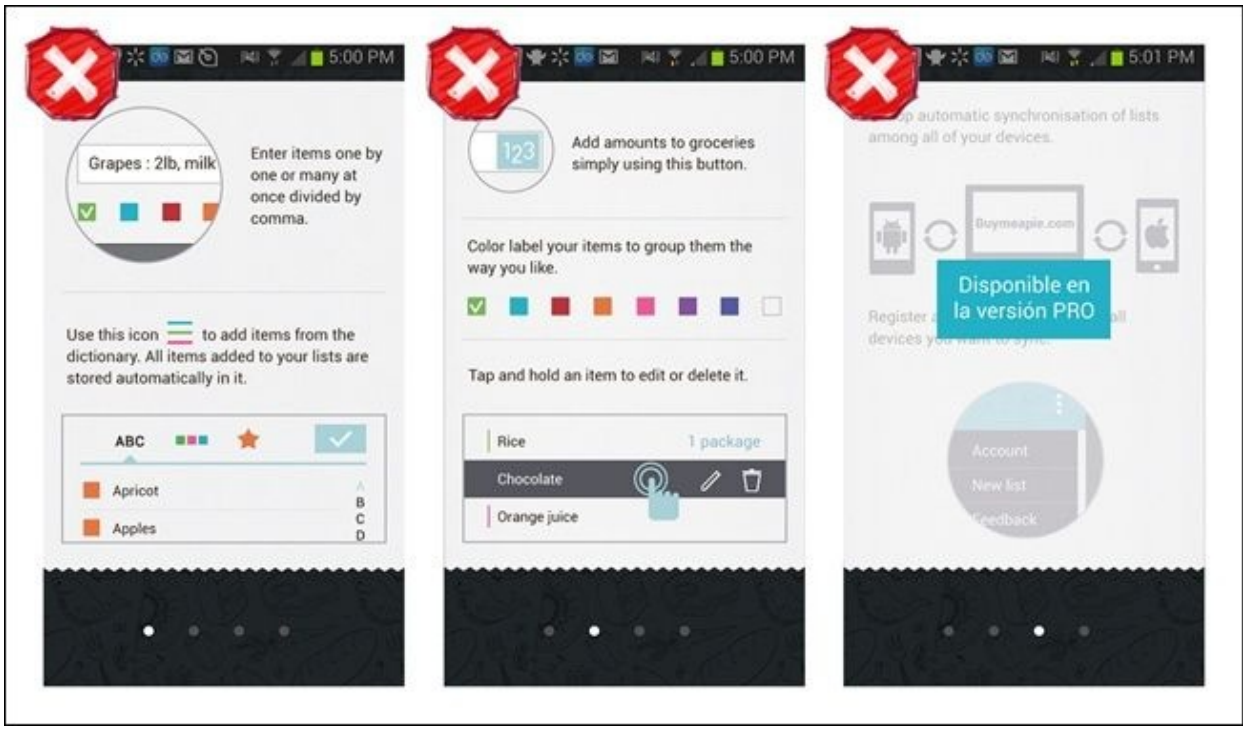



*Figure 7-17. Buy Me a Pie! for Android: frontloaded, confusing, and out-of-context tour*

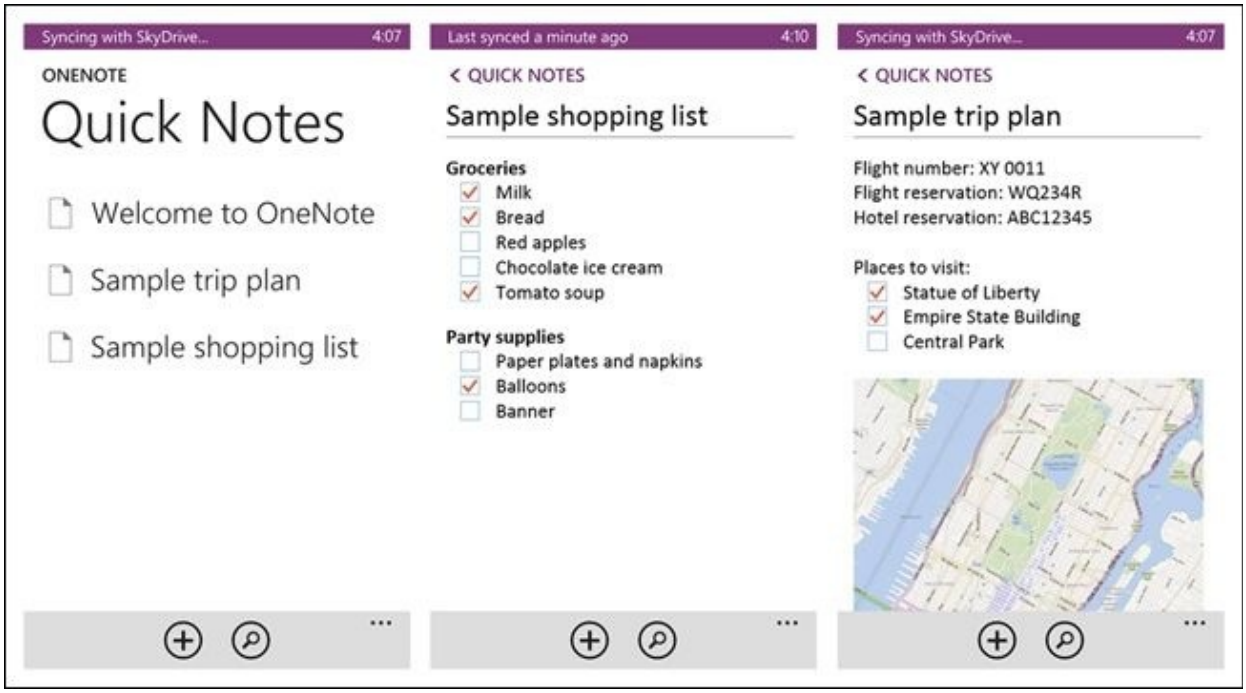



*Figure 7-18. OneNote for Windows Phone: two sample lists allow users to learn the ropes by doing*

## Clipchat Compared to Kik

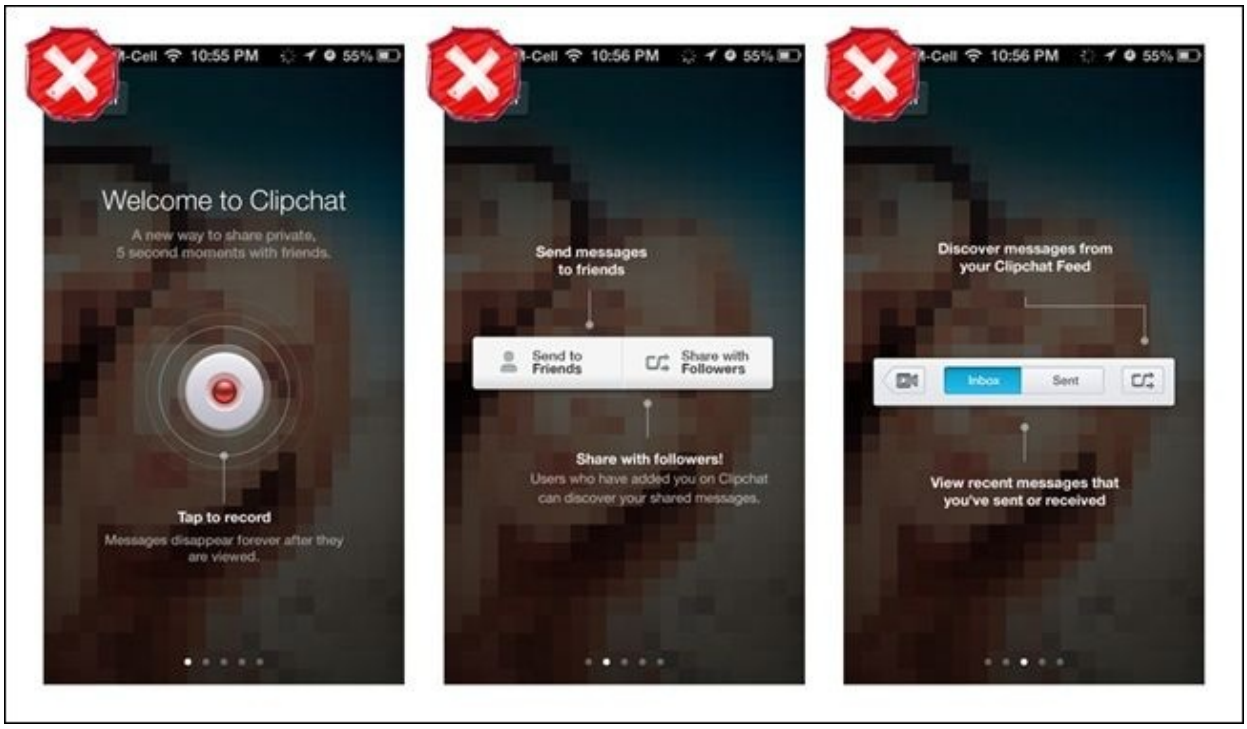



*Figure 7-19. Clipchat for iOS: too much up-front information, abstracted from usage context*

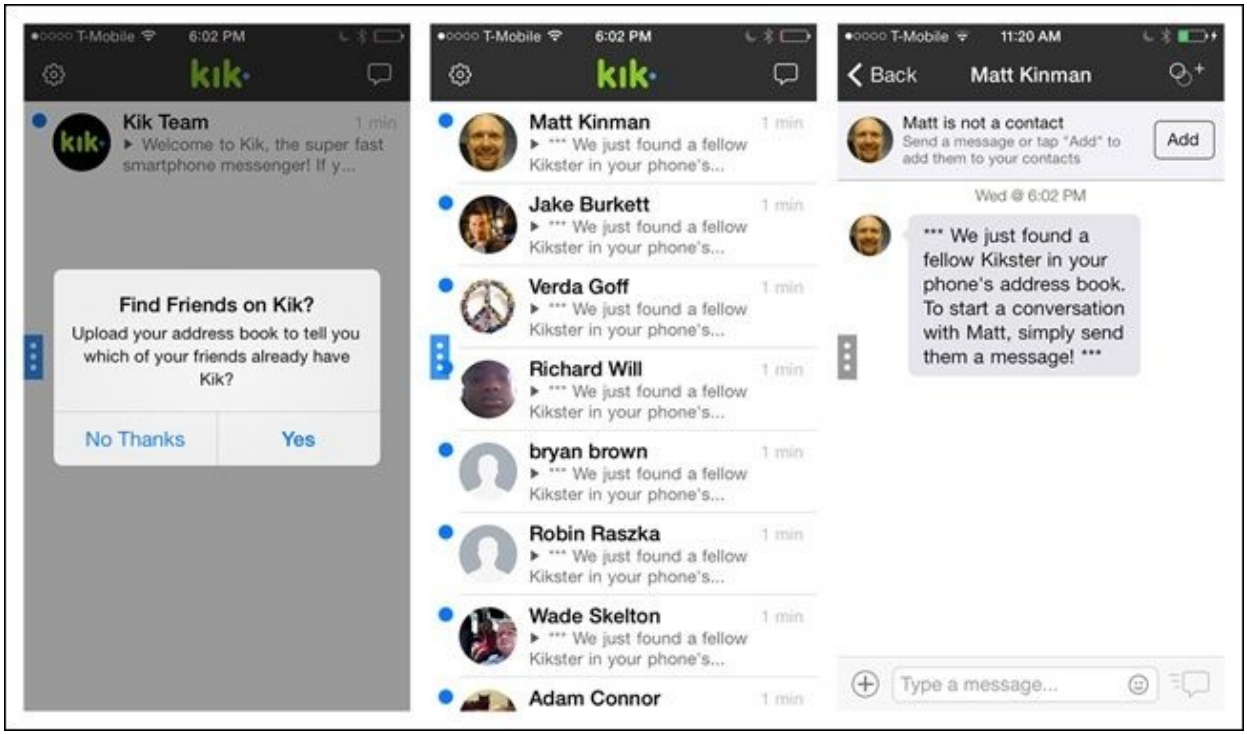



*Figure 7-20. Kik Messenger for iOS: import contacts and see a tip for how to start chatting with them*

**NOTE**

Don’t overload your users with all the information they will ever need up front; give it to them in bite-sized chunks, just where and when they need it.

# Rule #3: Make It Rewarding

Actually, you may recall that the rule from Tutorials 101 is “Make It Fun.” But let’s face it, “fun” is not appropriate for all apps. Still, when the narrator of the video says, “Your tutorial should be as engaging as any other part of your game,” we can see that in every other sense, this rule applies to our world.

Even if we can’t make learning our apps fun, there are certainly ways to make it rewarding and make it flow seamlessly into the overall app experience. One good way to do that is with interactivity that actually lets the user accomplish things. This provides a sense of “agency” that reinforces what we learn.

And even when “fun” is not the right tone to strike, sometimes instilling a sense of playfulness can still be appropriate. Our first comparison here is between NBC News and Flipboard: these apps take different approaches to this, with the latter exemplifying the right approach.

## NBC News Compared to Flipboard

Despite its “playful” font on the five transparency screens, the NBC tutorial still feels like a lecture. The first time you open Flipboard, on the other hand, it feels dramatically different.

There is no instructional text, but the bottom half of the screen playfully flips up a bit, teasing you with a peek at some content before flipping down again. After it does this once more, you’ll probably get the hint that you need to swipe up to reveal the underlying content. But if by then you still don’t swipe, you’ll get an embedded invitation that says “Swipe up to continue.”

Every subsequent flip transition reinforces the swipe gestures a new user needs while exploring the content in her Flipboard. Playful *and* rewarding.

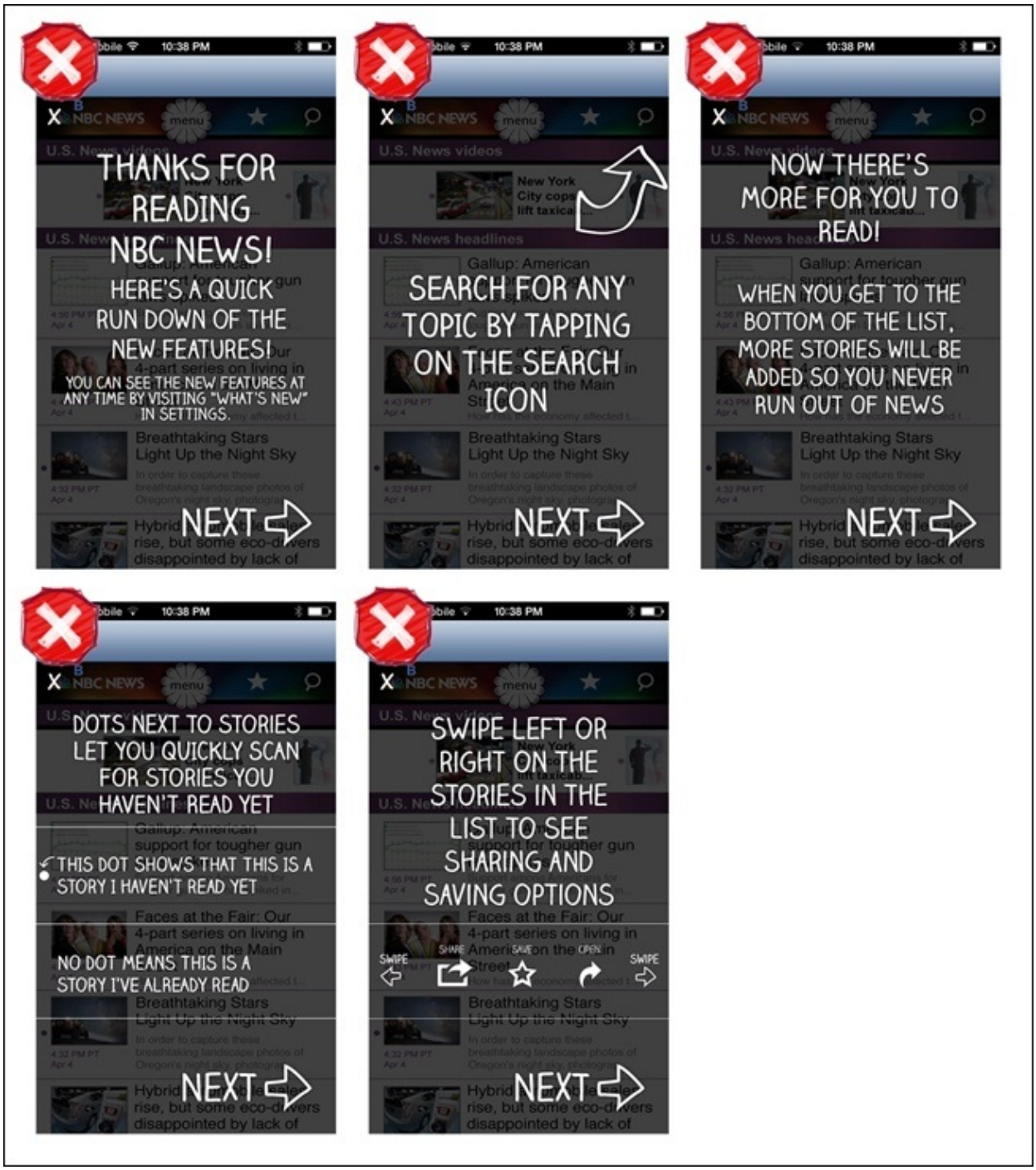



*Figure 7-21. NBC News for iOS: a playful font, but a long lecture isn't rewarding*

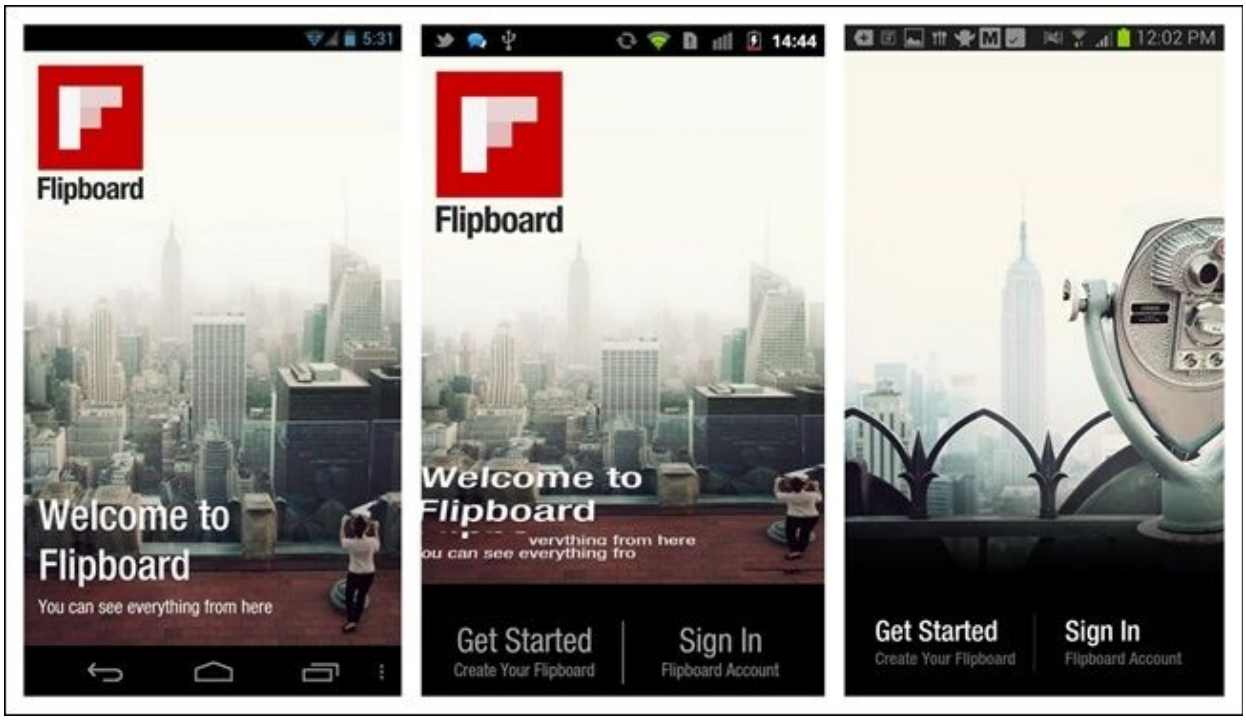



*Figure 7-22. Flipboard for Android uses playful embedded prompts to engage the user and reinforce the key gestures needed to navigate the app (http://www.youtube.com/watch?v=rQASZtuvBCs)*

## Noom Compared to DailyBurn Tracker

Noom for Android is an incredible app and does a great job of using rewards to invite deeper engagement. The experience is designed from start to finish to engage the user and keep him coming back. Noom uses a Gamification technique (see Chapter 8) called “leveling up” to encourage the user to successfully complete a series of tasks and reach the next level of proficiency.

It also includes an element of surprise to keep things interesting. Just as I completed the setup process, I was rewarded with an “overachiever bonus” that inspired just a little more engagement.

In contrast, setting up DailyBurn Tracker was a chore. It started with a half-dozen configuration screens, then dropped me straight onto the home page with a list of 10 options. No guidance, no motivation, and no surprises. (Motivational apps are one instance where surprises are a good thing.)

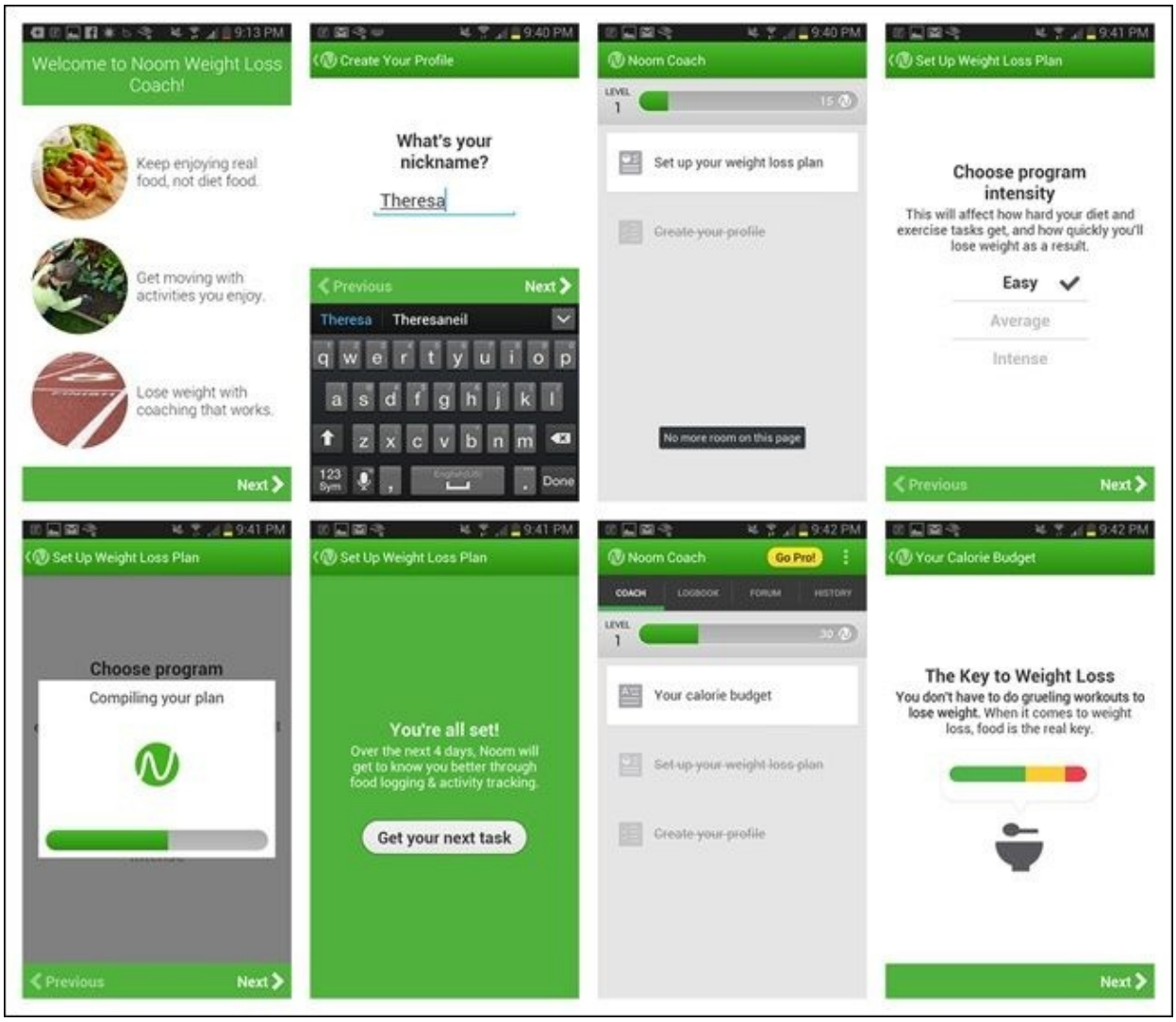



*Figure 7-23. Noom for Android: guided setup is easy and friendly, with positive feedback along the way*

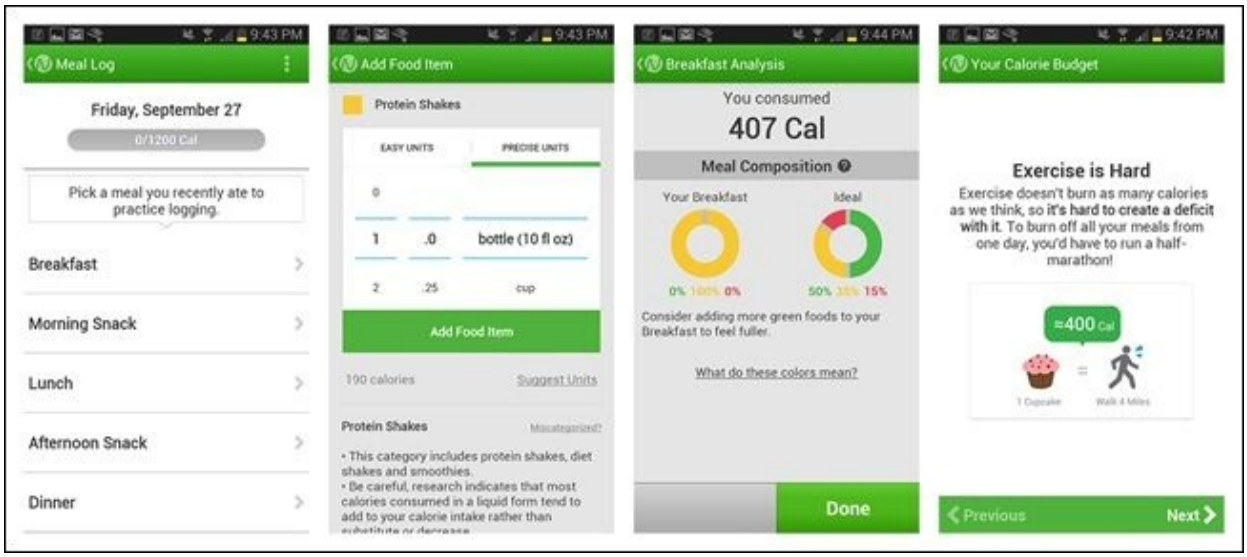


*Figure 7-24. Noom for Android: the tasks are designed for participation to ensure comprehension (like practice logging a meal)*

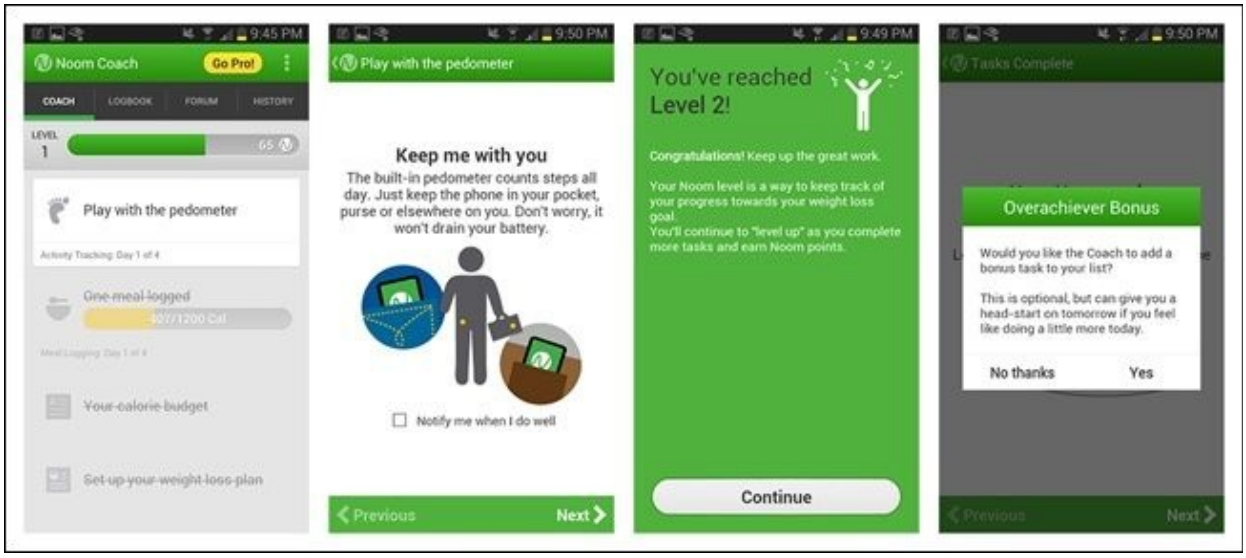



*Figure 7-25. Noom for Android: tasks start off easy to complete, but get harder as you progress; leveling up "gamifies" the experience and makes it feel more rewarding*

**NOTE**

If you download Noom for Android today, you may notice some differences between the screenshots shown here (taken in October 2013) and the newest release. I was curious about the catalyst for these changes, so I reached out to Eli Holder, Noom's VP of Engagement.

According to Eli, user testing showed that the 2013 design left participants feeling "exhausted." The redesign is intended to shorten the onboarding process into bite-size chunks and make it more about the user's weight loss story and less like a task list.

Keep an eye on Noom this year as Eli and his team evolve the onboarding flow to more quickly give users a joyful experience and an emotional connection with Noom.

**NOTE**

Make deeper engagement with your app rewarding. Consider adding elements of playfulness where appropriate, and let users learn by doing to create a sense of agency.

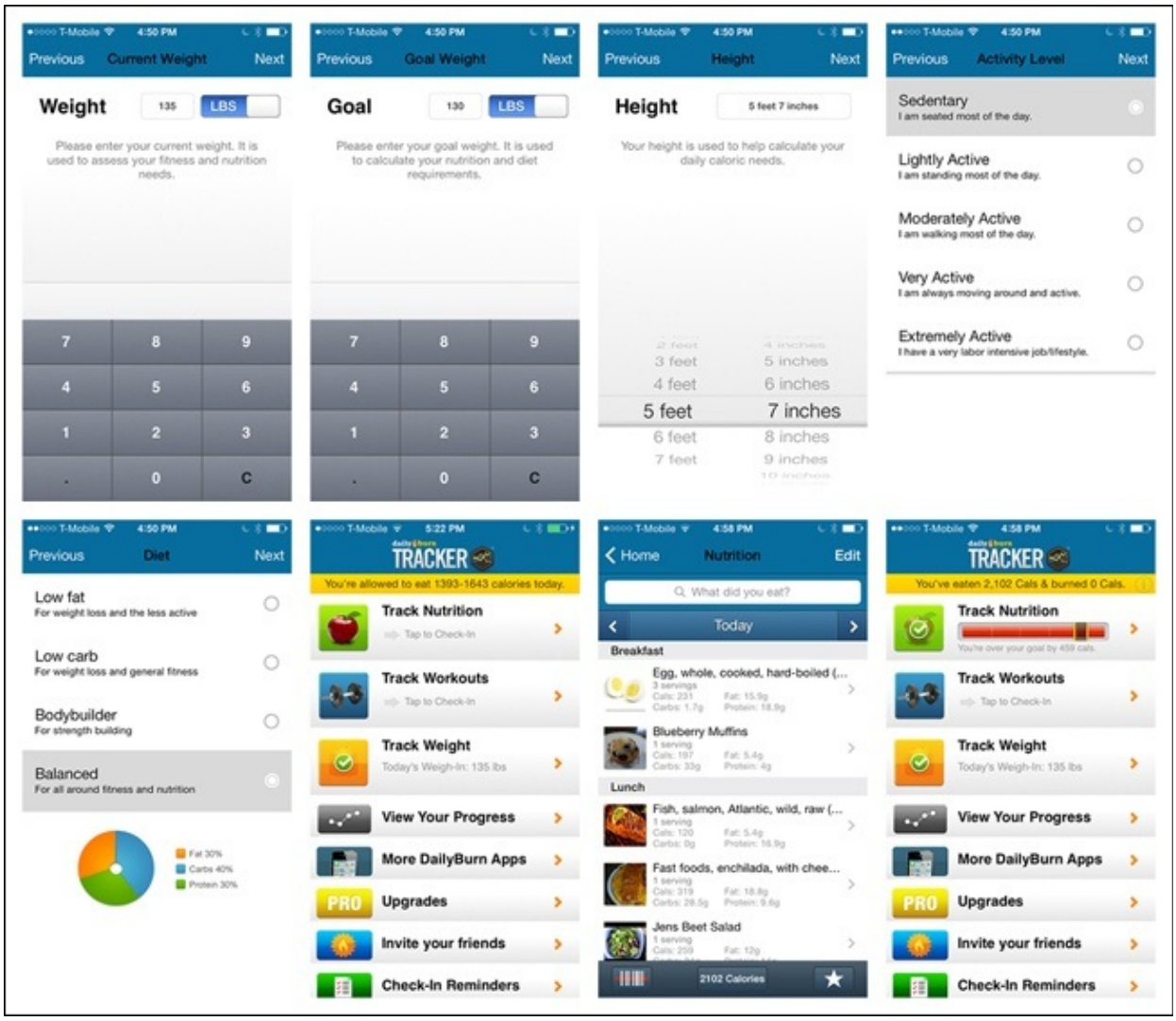



*Figure 7-26. DailyBurn Tracker for iOS: setup feels like an impersonal, sterile chore*

## Rule #4: Reinforce Learning Through Use

Remember those “a-ha!” moments you’d get in science class when you came to understand a concept by demonstrating it in a simple experiment? That’s what we’re talking about here. Of course, the teacher had *told* you the concept, but it was through doing the experiment that you actually *learned* it.

The same idea applies with tutorials. And if you follow the first three rules, this reinforcement will largely take care of itself. It can be as simple as accompanying an action that is demonstrated in a tutorial with a very subtle bit of visual or aural feedback. Then, when the user subsequently performs the action, the same feedback reinforces what she has learned. (If you do use audio feedback, offer a way to turn it off so that it doesn’t become annoying once the user is familiar with the app.)

This rule complements the “No frontloading” rule. Rather than trying to show off everything in your app at once, consider crafting an experience that invites the users progressively deeper into the app. By revealing more advanced features over time, or by giving the users unexpected “rewards” as they progress, you’ll help reinforce what they learn as they use the app.

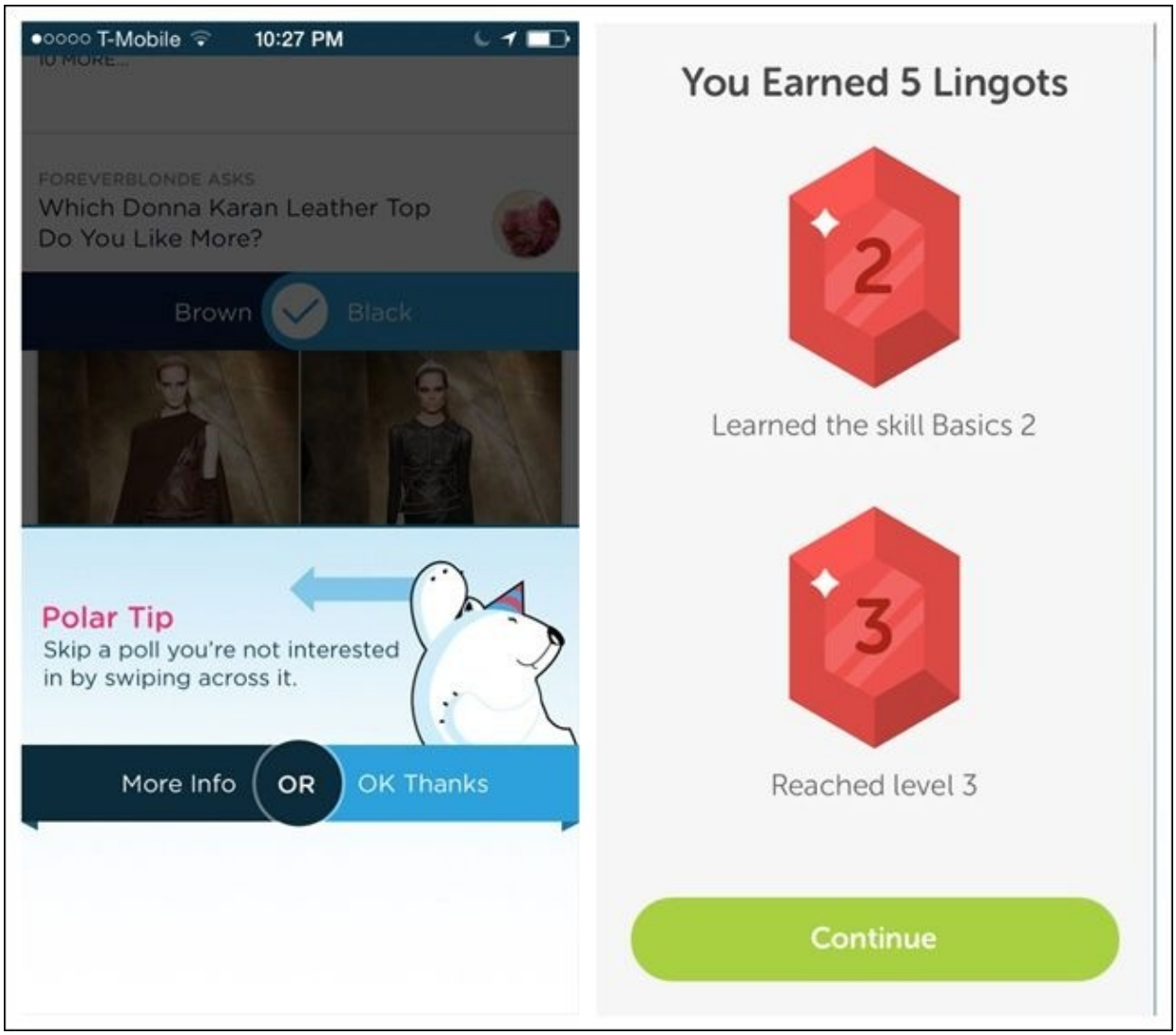



*Figure 7-27. Polar for iOS: after answering a few polls, I get a tip; DuoLingo for iOS: rewards participation with Lingots*

**NOTE**

Learning isn't a onetime thing; using an app should reinforce what the tutorials have helped users learn.

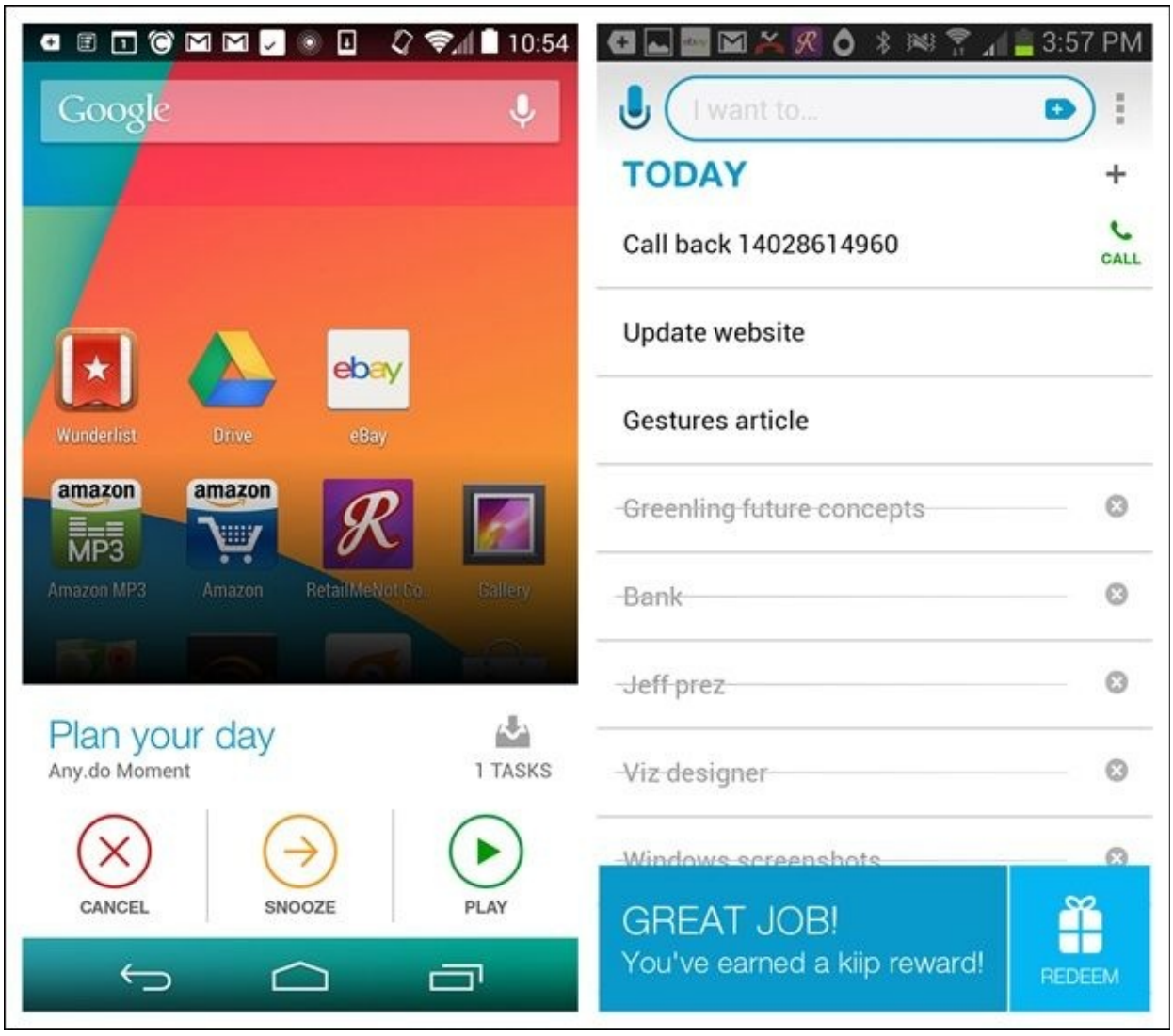



*Figure 7-28. Any.do for Android uses a widget to encourage engagement and spontaneous rewards for completion*

## Rule #5: Listen to Your Users

Considering that you've been deep inside your app building and refining it for months, who's the best person to explain to users how it works?

Probably not you, according to the narrator of Tutorials 101, who says: "When you're a designer who's been working on a project for a year or two, it's very easy to think things are intuitive or obvious that are actually totally incomprehensible."

Long-term exposure to an app blinds us to what users will need to be taught in order to use it. We'll touch on this later, in Chapter 10. But for now, the key to keep in mind is that your users (or players in the game world) are the only resource that counts when it comes to measuring the effectiveness of your tutorials.

Simple, properly conducted user testing will reveal any stumbling blocks. Observe your users to see where they get stuck and where they have problems. Listen to their comments as they interact with the app. Don't question them until later; if you question them as they are using the app, you are likely to unconsciously lead them to the answers you want to hear.

> **NOTE**
>
> Familiarity blinds you to what users need to learn. Let users show you themselves through unbiased user testing.

That brings us to a few last tips from the experts at Extra Credits:

6. Don't wait until the end to design your tutorial.
   You must integrate your tutorial into the overall design process from the very beginning. Too many teams relegate the tutorial to an afterthought, and that couldn't be more wrong or potentially damaging to the fate of your app. Many of the tutorial screens I marked as anti-patterns in this chapter probably got to be that way because they were afterthoughts in otherwise well-designed apps. Tutorials should be treated as one of the most important elements of your app. If they fail, your app fails.
7. Tutorials should be skippable.
   Have you ever used an app that forced you to go through a tutorial every time you opened it? If so, you probably didn't use that app for very long.
8. Anything conveyed in the tutorial should be accessible at all times.
   By the same token, any help provided in your tutorials should always be

available for reference. For instance, Fidelity lets you relaunch the tutorials from Settings. See Chapter 10 for more examples.

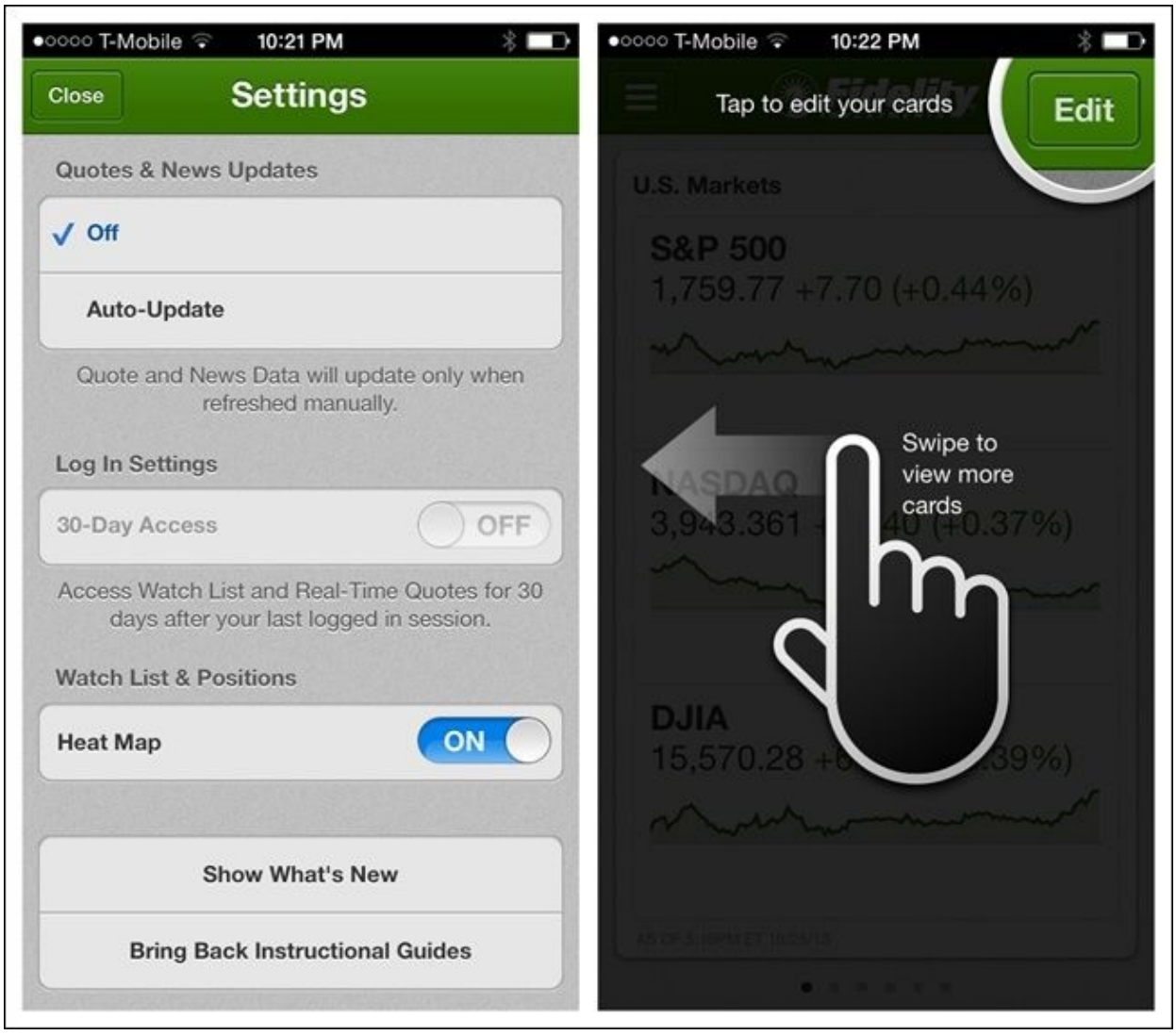



*Figure 7-29. Fidelity for iOS: Bring Back Instructional Guides lets you relaunch tutorials*

# Invitation Patterns

Whether or not your application requires a tutorial for onboarding, there are likely opportunities throughout the app to invite users to engage with a specific feature. Invitations are also helpful to introduce new features within a specific flow, as opposed to just showing a dialog on launch with a list of new features.

## Tips

Tips should be specific to a particular tool or action on the screen. Tips in the form of Transparencies (Zappos) or Dialogs (HoursTracker) should be avoided, as should multiple tips per screen, as in myAppFree. For smart Tip design, see the examples from Evernote Food and Mint.

> **NOTE**
>
> Place Tips in proximity to the action or menu item they apply to, keep the content short, and remove the Tip once interaction begins (i.e., when the screen is touched).

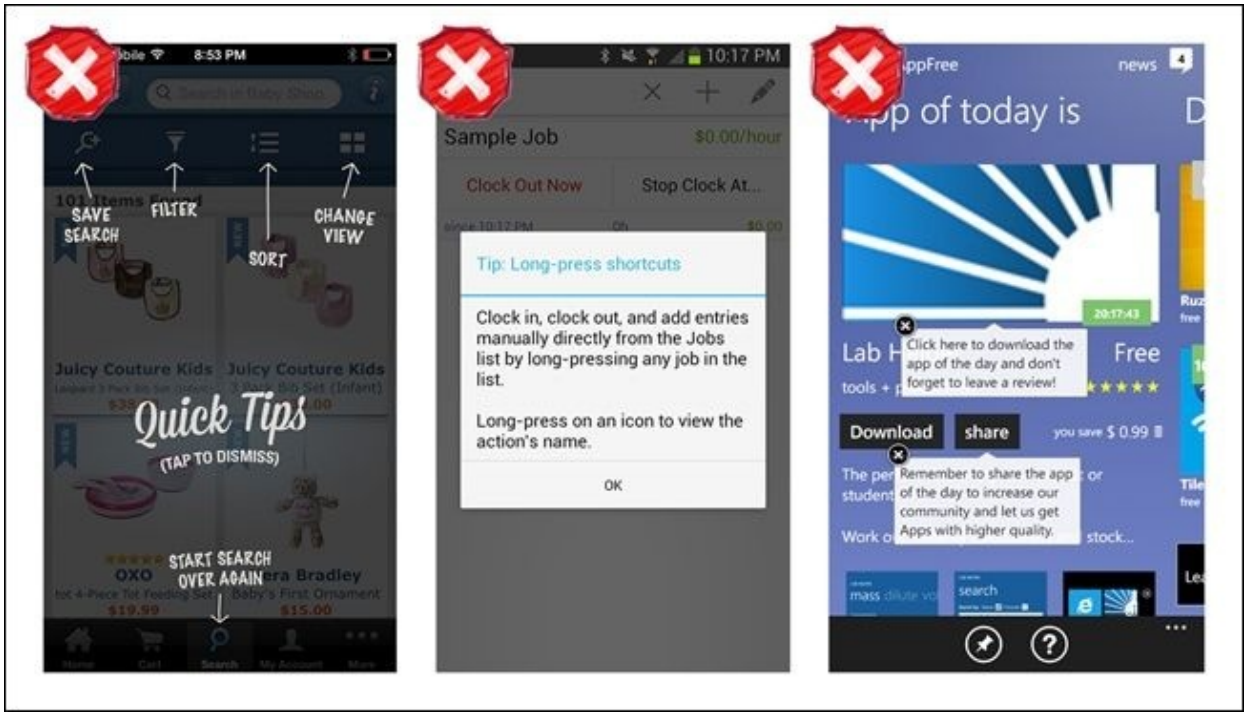



*Figure 7-30. Zappos for iOS, HoursTracker for Android, and myAppFree for Windows Phone: poorly designed Tips*

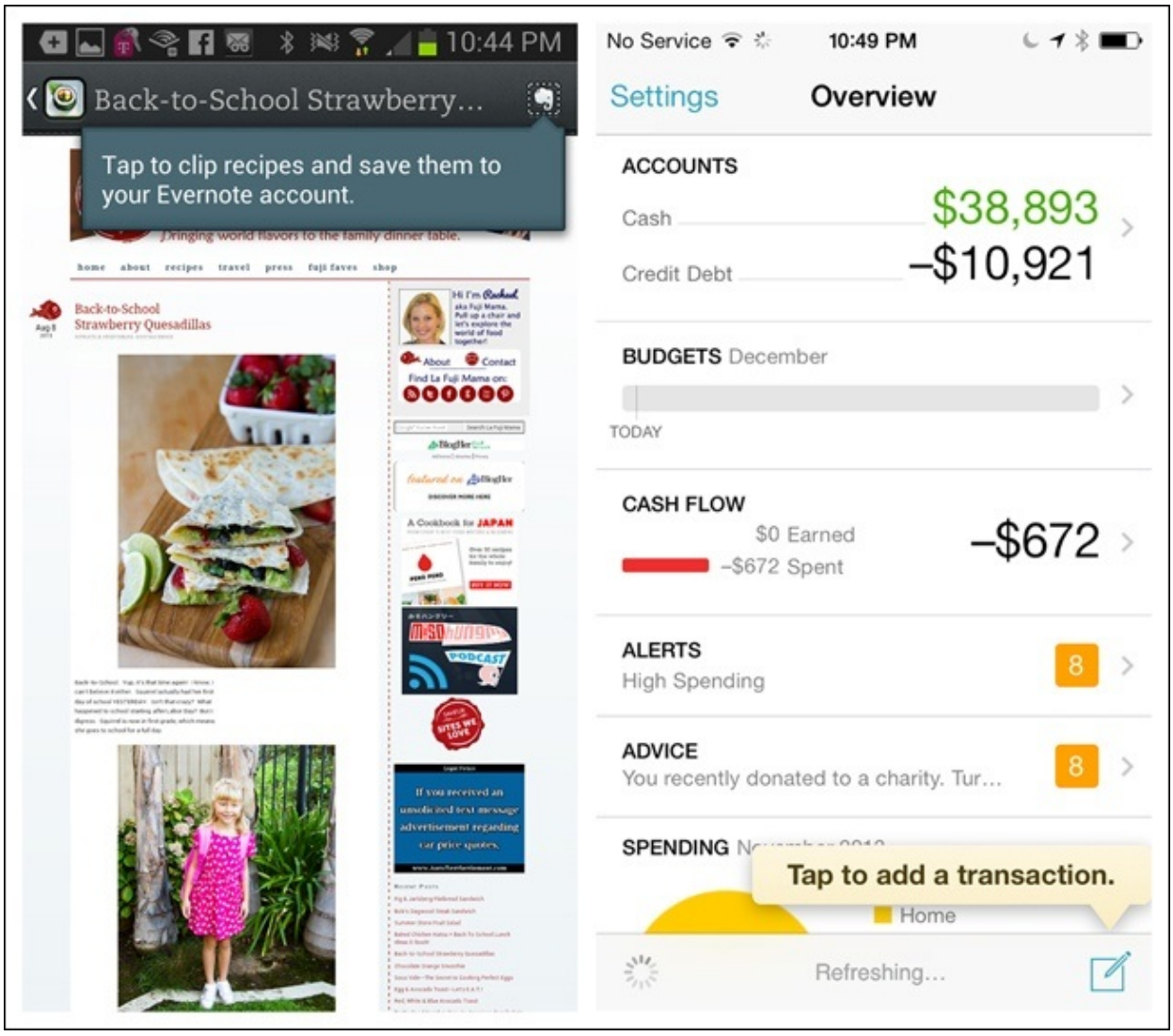



*Figure 7-31. Evernote Food for Android and Mint for iOS: well-designed Tips*

For a detailed look at designing effective Tips, take a look at the Intuit case study in Chapter Extra: Invitations — Rolling Out the Welcome Mat at the end of this chapter.

## Persistent Invitations

Persistent Invitations might look like Tips in some instances, but they differ in their, well, persistence. These invitations are either always visible, or remain visible until the user takes the specified action.

In the examples from RetailMeNot, Expense Manager, and Pandora, the invitations will persist until the user takes the requested action (save coupon, record expense, and add favorite artists, respectively). Then they disappear for good.

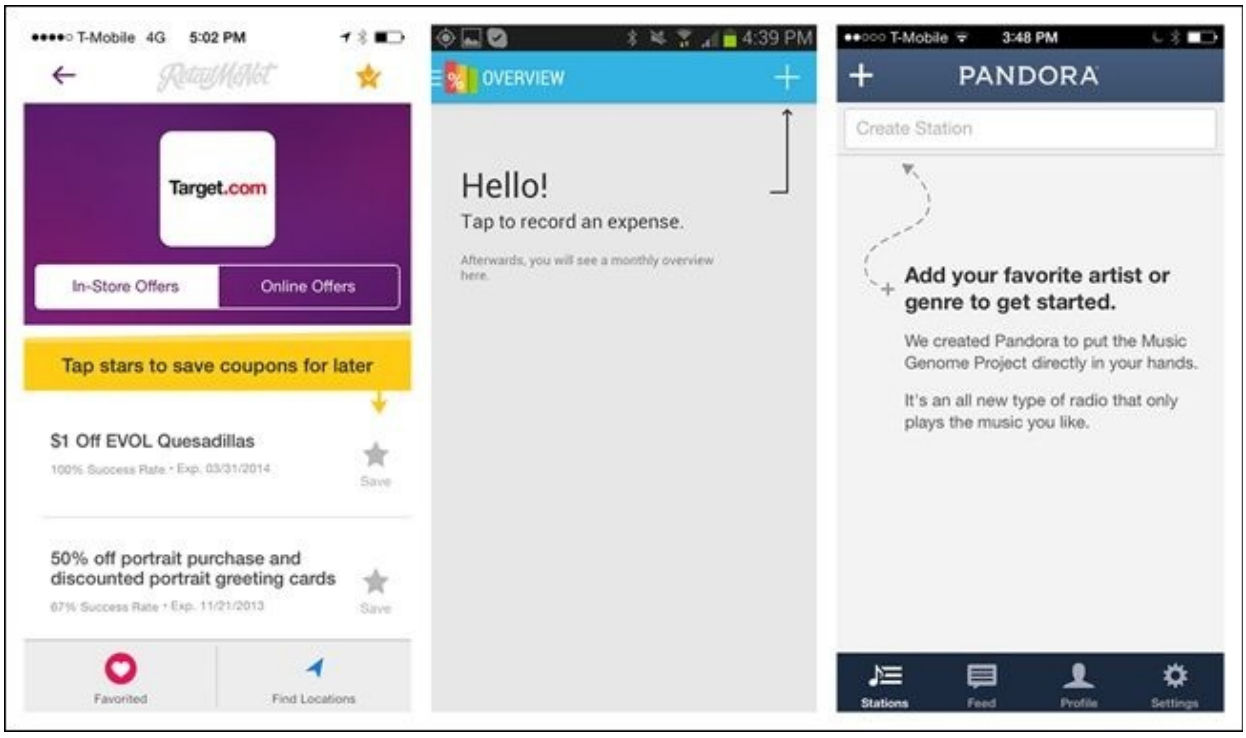



*Figure 7-32. RetailMeNot for iOS, Expense Manager for Android, and Pandora for iOS: Persistent Invitations*

With Pinterest, however, the invitation is always extended. No matter how many boards you add, there will always be an invitation to add another. The verdict is out on where to place the invitation: in Do It (Tomorrow), it's at the top of the list; in Flipboard, it's at the end.

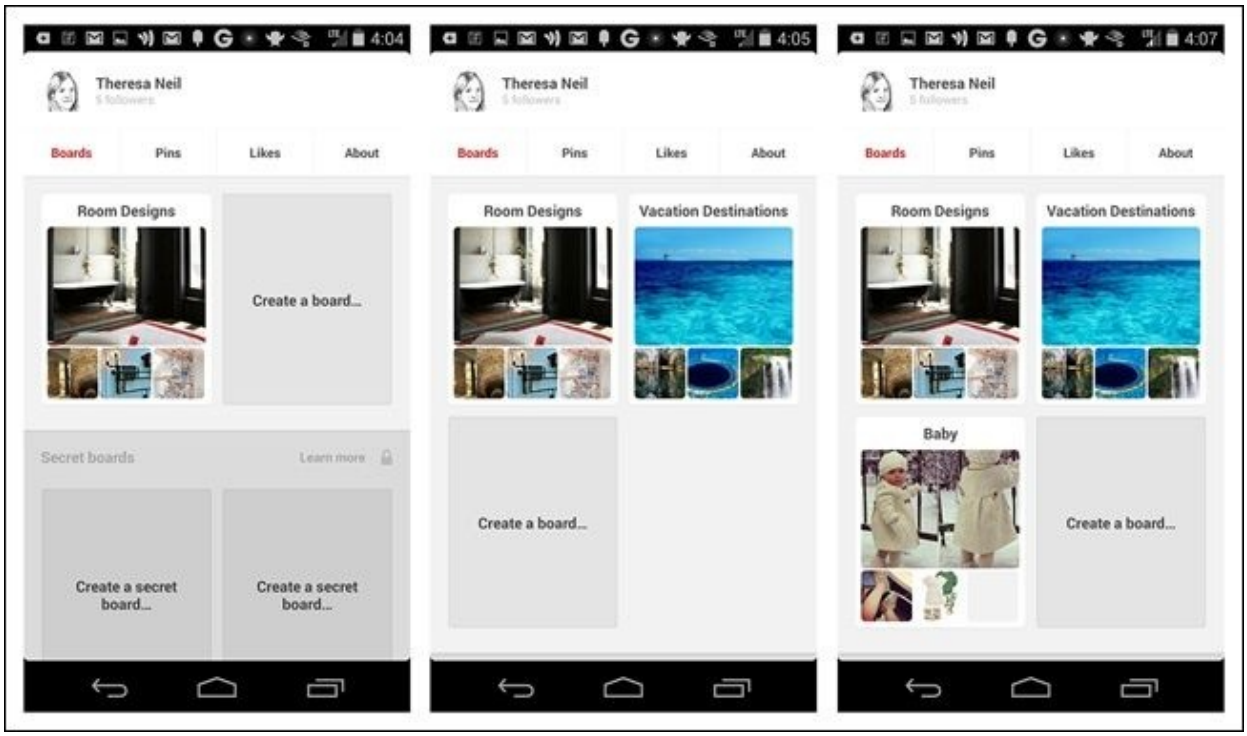



*Figure 7-33. Pinterest for Android: Persistent Invitation*

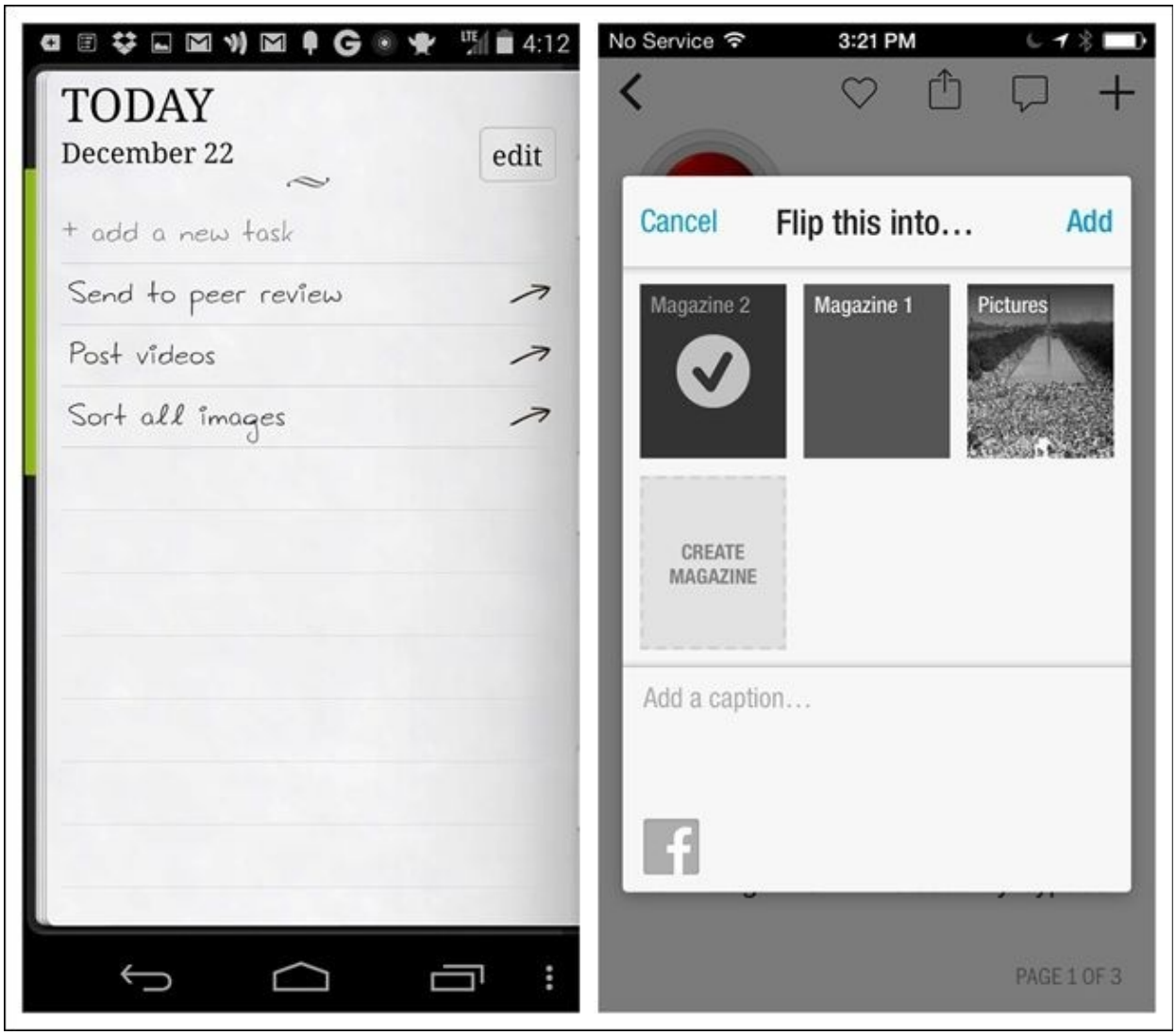



*Figure 7-34. Do It (Tomorrow) for Android and Flipboard for iOS: invitations can go at the beginning or end of a list*

Argus has built-in logic to show Persistent Invitations in context with the user's preset exercise and nutrition goals. For example, the "Tap to set weight" hexagonal invitation will persist until I log my weight for the day, and then it will be replaced with a hexagon showing my weight. And I'll get that same invitation again every week.

**NOTE**

Put some thought into the right level of persistence for the invitation; should it be always visible, visible until acted upon, or visible in a specific context? Test this with users to make sure the invitation is helping them, and not annoying them.


*Figure 7-35. Argus for iOS: invitation to set weight reappears based on user's goal settings*

## Discoverable Invitations

A Discoverable Invitation might seem like an oxymoron, but it is an effective way to encourage specific interactions without cluttering the screen. These invitations are meant to be discovered when the user performs a common gesture, like tapping, pulling, or swiping.

In Polar, like many data feed apps, pulling the content list down reveals a "Release to refresh" invitation. Tapping on the chart in SigFig (a natural gesture for wanting to see a larger version of the chart) shows an invitation to rotate the device to landscape view.

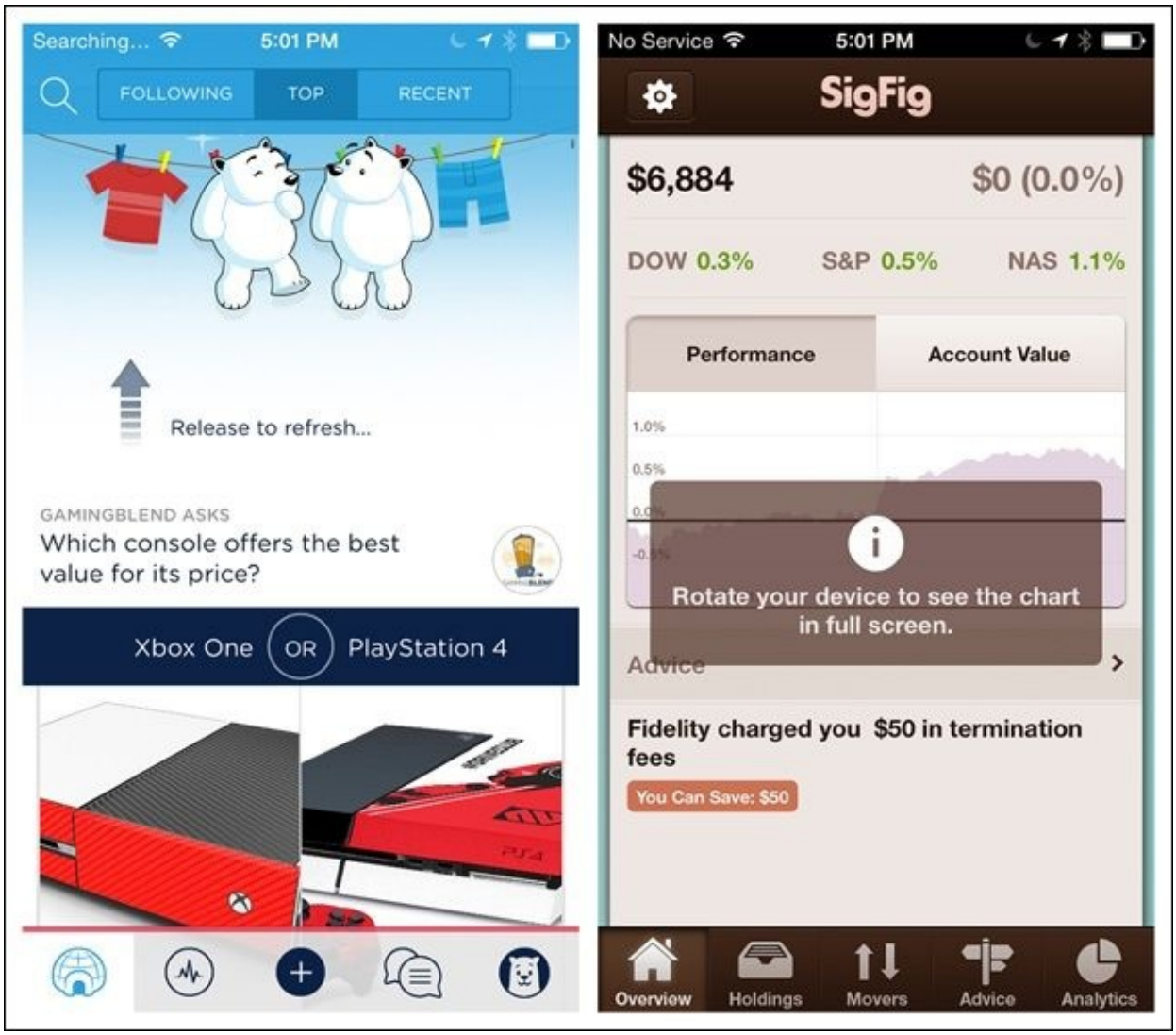



*Figure 7-36. Polar and SigFig for iOS: Discoverable Invitations*

# Chapter Extra: Invitations — Rolling Out the Welcome Mat

**ALISSA BRIGGS, UX MANAGER AND PRINCIPAL DESIGNER AT INTUIT**

Imagine that a new acquaintance has invited you to a dinner party. You show up at the door, bottle of wine in hand. But you hesitate before going in. Will you know anyone there? What if you make a mistake or embarrass yourself?

Just then, your host opens the door, gives you a big hug, and welcomes you in. He takes your coat and introduces you to the other guests. Your hesitation washes away and you smile, feeling confident and ready to enjoy the evening ahead.

Designing a satisfying first-use experience is not so different from being a great host. The goal is the same: to help your guests quickly overcome their anxiety, and to make them feel welcome and confident.

When we began designing Intuit Snap Payroll, a smartphone-based paycheck calculator, one of our top priorities was to create a welcoming first-use experience. Our customers are small-business owners new to the world of payroll. They have a lot of fear and worry when it comes to calculating paychecks accurately. To convey that Intuit Snap Payroll would let them master this intimidating task, we had to make the user's first experience welcoming and confidence-inspiring.

Here's a look at how we found a solution and what we learned along the way.

## Iterating on the Welcome Experience

To understand which design approach would be most effective for inspiring confidence, we got customer feedback on a variety of design concepts. The images that follow are actual wireframes that we tested.

### Concept #1: Direct Immersion

Our first concept was to ditch the “welcome mat” altogether and drop users directly into the app menu that they would rely on during ongoing use.


*Figure 7-37. No welcome mat: just drop users right in*

Customers found this confusing and felt lost. They weren’t sure which option they should choose first, and that added to their anxiety about doing payroll on their own. To continue the dinner-party metaphor, it was as if our guests magically appeared in the center of a party where they didn’t know anyone and the host was not in sight. To welcome them in, we needed to provide additional context.

### Concept #2: Tour

We called upon a common invitation pattern — the Tour — to provide additional context. Curious to see which format was most effective, we tested the four versions shown.


*Figure 7-38. Benefit statement: an image, short benefit statement, and clear call to action*


*Figure 7-39. Feature list: a benefit statement and short description of core functionality*

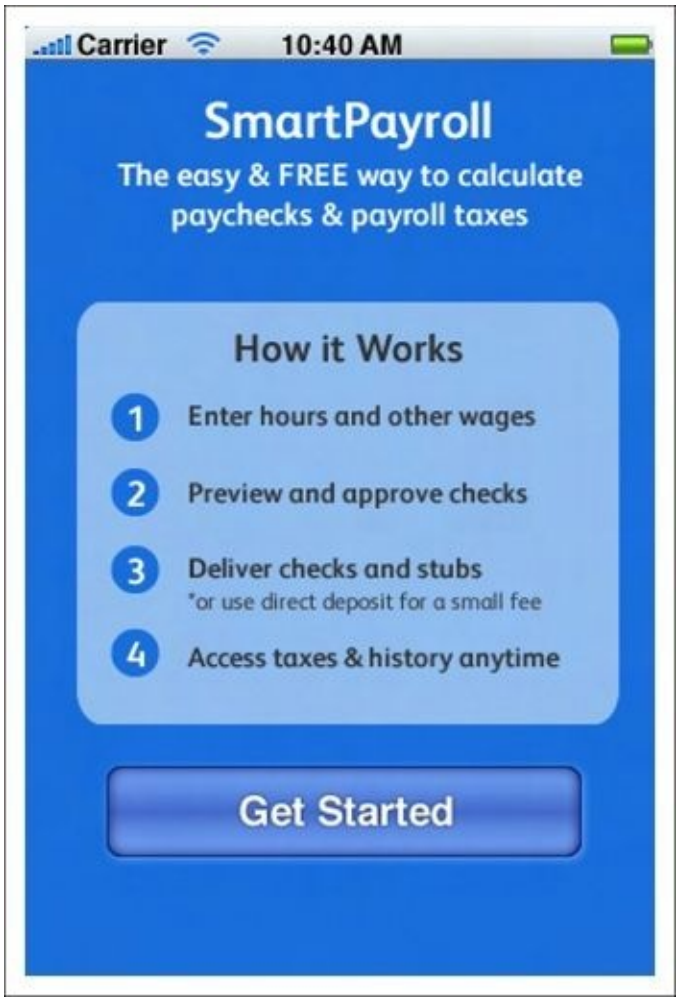



*Figure 7-40. Step list: a step-by-step list explaining how to use the app*


*Figure 7-41. Multi-step tour: a series of introductory screens that users could swipe through to learn more about the app*

Users responded consistently to all of these: no one read the screens or swiped

through the feature lists. Instead, they immediately skipped over them and tapped Get Started. Users explained that the screens got in their way and that they'd rather just jump into the experience immediately.

Providing context via an up-front informational message did not aid user confidence. It was like the dinner guests reached our door and found a note explaining how a dinner party works, and inviting them to let themselves in. We needed to set the proper context — without getting in the way.

### Concept #3: Multi-Step Transparency

To provide context without interrupting user flow, we tried another common pattern: Transparency. With this pattern, the user can see the actual app interface, but with additional context overlaid on top.

We tried several approaches to overlays, two of which are shown here.


*Figure 7-42. Multi-step transparency: multiple features are called out on the main navigation screen*


*Figure 7-43. Multi-step transparency tour: a series of steps that the user needs to tap through, each describing a navigational element*

Although users liked being able to see the app in the background, they were irritated by how many instructions were on the screen. Comments included: “I don’t have time to try to understand this,” “I just want to make this get out of my way,” and, of course, “Do I have to read all this?”

We were getting closer, but showing too many instructions at one time made customers feel like there was too much to learn. It was like our dinner host had welcomed them in, suggested 10 different guests they ought to meet, and then left them standing in the hall with their heads spinning. How could we make the experience even easier?

### Concept #4: Single-Step Transparency

We adapted our transparency concept to focus on just the *one* thing a user should do upon first entering the app.


*Figure 7-44. Single-step transparency: a focus on the most important thing*

Customers responded more favorably to this. They liked being able to get a sense of all of the options that would be featured in the app, but also liked having a clear recommendation about where to start.

But the concept also made users a little nervous. Customers asked, "If you have a tutorial, does this mean the app is too complicated for me to figure out for myself?" We also observed users trying to tap on the gray background to close the overlay and get a better view of what was behind it. We saw an opportunity to simplify even more and further get out of the user's way.

### Concept #5: Tip

In this concept, we kept what was working well about the single-step transparency (one clear call to action and a visible menu), while fixing some of the problems (tutorial link and gray overlay). In this concept, the user saw one call to action in the form of a tip, and could tap elsewhere on the screen in order

to hide it.


*Figure 7-45. Tip: one call to action, with the rest of the screen tappable*

This tested very well. Customers had a good sense of where they were and what was available to them, as well as a clear first step. At last, we could make our dinner-party guests feel welcome and confident, while introducing them to “a menu item you simply must get to know!”

To craft a great first experience for our users, we’ve learned to keep the following principles in mind:

Get out of the way

Don’t obscure elements of the UI or force users to sit through an unwanted tour; it feels like you’re hiding something.

Provide guidance

Remove friction and uncertainty by suggesting the first step in the workflow.

These findings led to our final design for the Intuit Snap Payroll first-use experience. As in Concept 5, we provided a Tip that points to the first action a customer should take.

When a customer first opens the app, he sees the menu screen overlaid with a friendly sticky note that points to the recommended menu item. But he can view other parts of the menu, too, and hide the note, if he chooses.


*Figure 7-46. Pre–iOS 7 implementation: hideable sticky note*

This has been very successful in transforming our customers' fears into confidence. Users quickly understand the functionality the app offers, while also feeling that they know how to proceed in order to have a positive experience.

With the introduction of iOS 7, we used subtle translucency and animation effects to strike the right balance of openness and guidance. In this example, the menu loads first and then the sticky note fades, conveying helpfulness while staying out of the way.

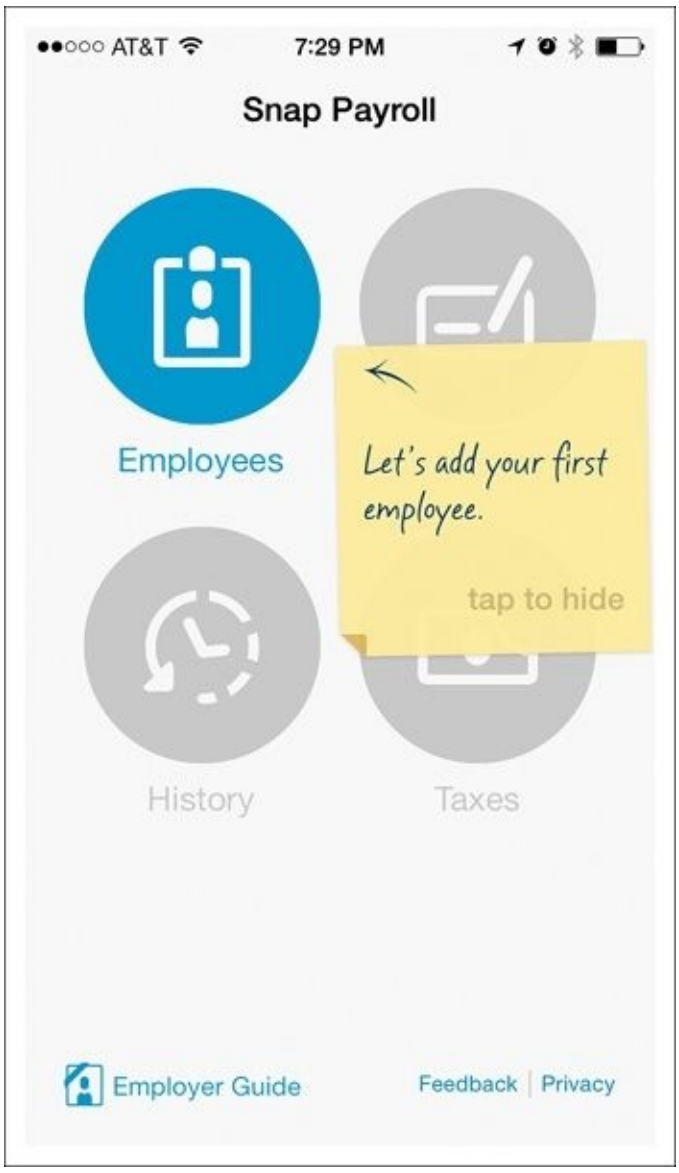



*Figure 7-47. iOS 7 implementation: helpfulness that doesn't get in the way*

## Summary

When it comes to inviting your customers into your app, don't just point to the door and expect them to let themselves in. Take the time to craft a thoughtful experience that makes them feel welcome and confident. You'll leave them looking forward to their next visit.

# Chapter 8. Social Patterns


*Patterns*

Social Registration, Connecting, Following, Profiles, Groups, Gamification

Services that provide social connections have continued to proliferate and specialize. Now we can connect with other people through our shared affinities for food, shopping, fitness, and much more. In this new chapter, we'll examine patterns that can help your users make these connections through their mobile devices.

# Social Registration

In Chapter 2 we looked at Registration form patterns. Here, we'll take a look at Registration user flows specifically for social apps.

## MapMyFitness Compared to We Heart It

MapMyFitness offers Sign Up with Facebook, and uses the Facebook API to show a single authorization screen. If authorized, it pulls pertinent registration information from your Facebook profile to create your MapMyFitness profile for you.

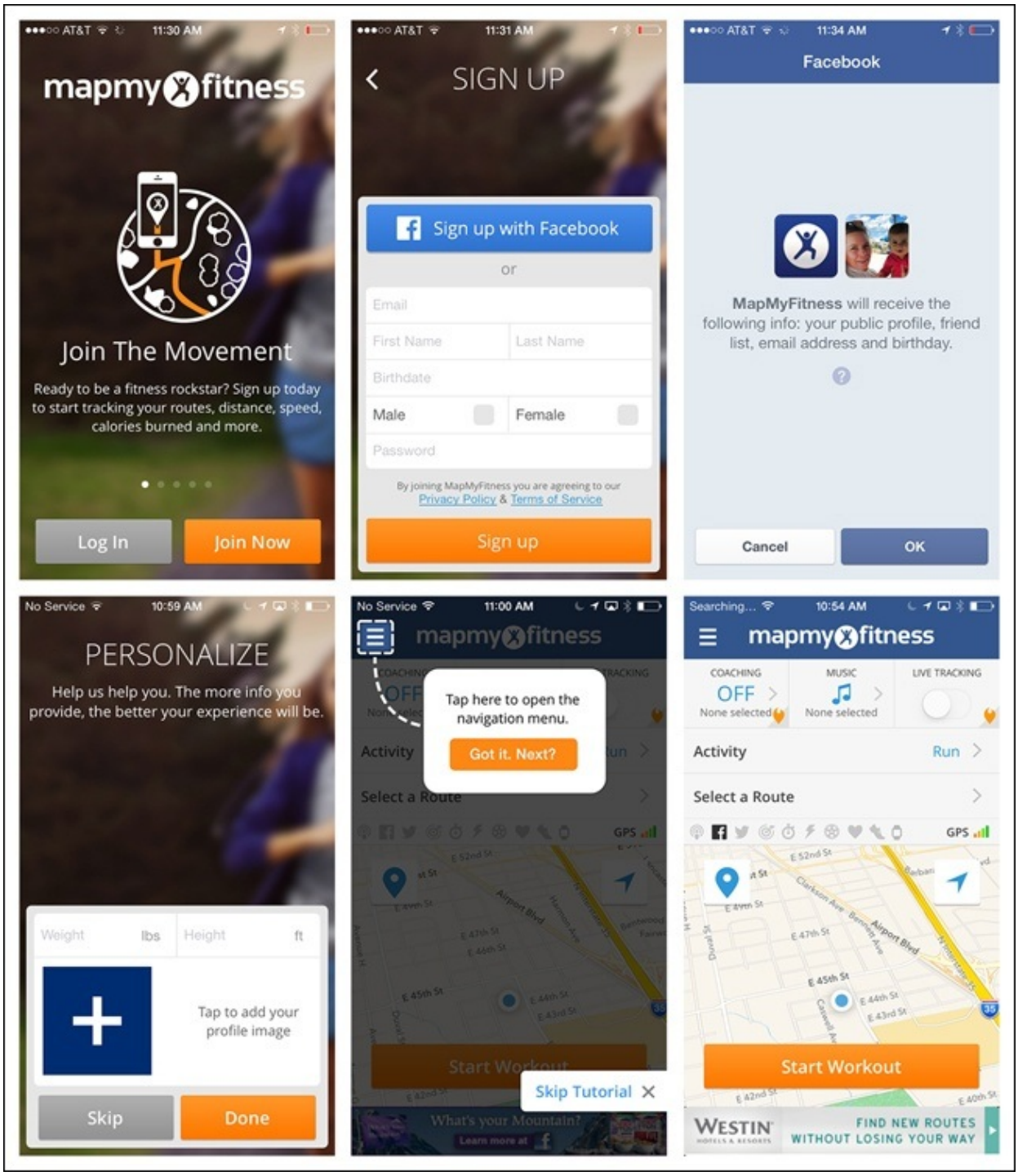



*Figure 8-1. MapMyFitness for iOS streamlines Registration with Facebook and an optional personalization screen*

Unfortunately, We Heart It, like many apps, has implemented the Facebook API incorrectly. After I authorized the use of my Facebook identity to register, it asked me to sign in to Facebook (I already had), and only then told me that if I

wanted to use Facebook to register, I needed to adjust my Safari settings. What the heck? And to top it off, there was a bug: I was presented the option to “skip posting to Facebook” multiple times in a row. It’s these kinds of confusing extra steps and glitches that cause user frustration and abandonment.

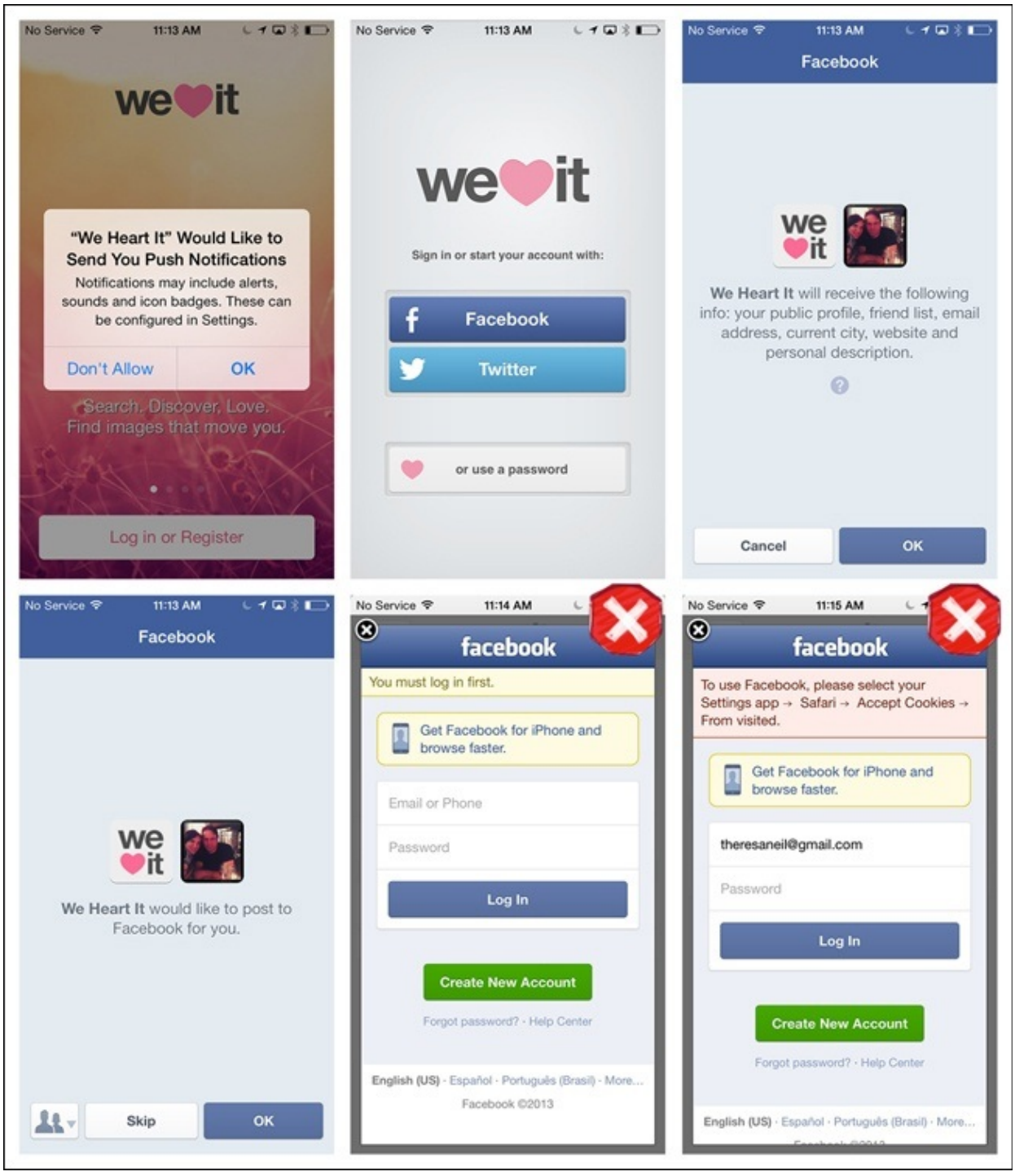



*Figure 8-2. We Heart It for iOS: I don’t “heart” all of these authorization screens and errors*

**NOTE**

Use social network APIs properly to streamline the Registration experience. Make sure to perform quality assurance testing on all paths of the Registration flow.

## Connecting

Part of the beauty of using APIs for other social platforms is that it lets users access their existing social connections. Oggl offers a simple connection flow as an optional part of the Registration process. In this example, I registered using my Twitter account, and then Oggl created a prefilled profile for me; I only needed to add a password and email address. A couple of taps later, I was able to connect with everyone I follow on Instagram, Twitter, and Facebook.

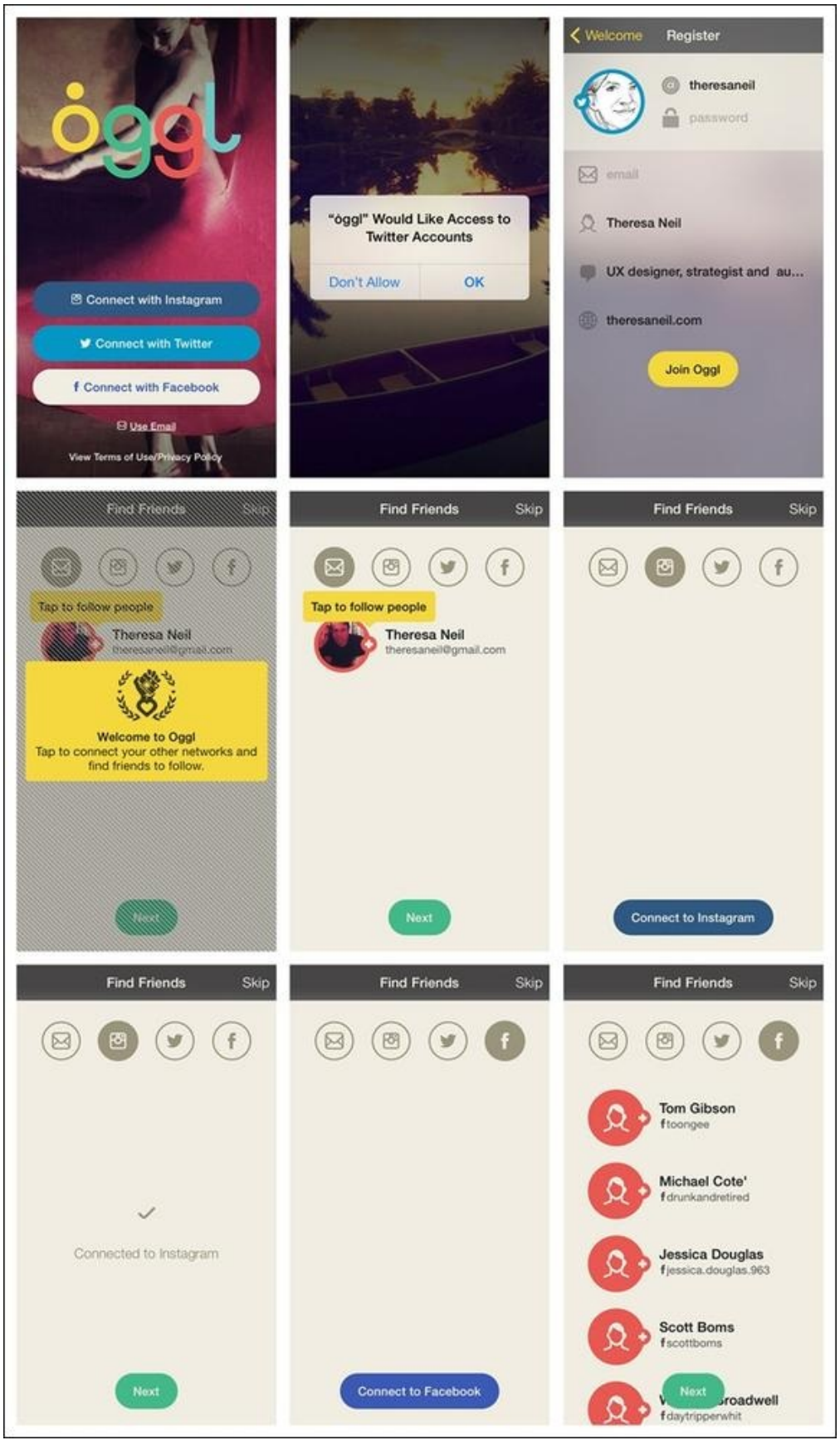

oggl
Connect with Instagram
Connect with Twitter
Connect with Facebook
Use Email
View Terms of Use/Privacy Policy
"oggl" Would Like Access to Twitter Accounts
Don't Allow
OK
Welcome
Register
theresaneil
password
email
Theresa Neil
UX designer, strategist and au...
theresaneil.com
Join Oggl
Find Friends
Skip
Tap to follow people
Theresa Neil
theresaneil@gmail.com
Welcome to Oggl
Tap to connect your other networks and find friends to follow.
Next
Connect to Instagram
Connected to Instagram
Connect to Facebook
Tom Gibson
ftoongee
Michael Cote'
fdrunkandretired
Jessica Douglas
fjessica.douglas.963
Scott Boms
fscottboms
fdaytripperwhit

*Figure 8-3. Oggl for iOS: adding connections from other social networks as an optional flow in Registration*

LinkedIn has introduced a new flow in its web and mobile apps that has had a significant impact on driving additional connections. Any time you accept an invite, there is a full-screen prompt to “See who you already know on LinkedIn.”

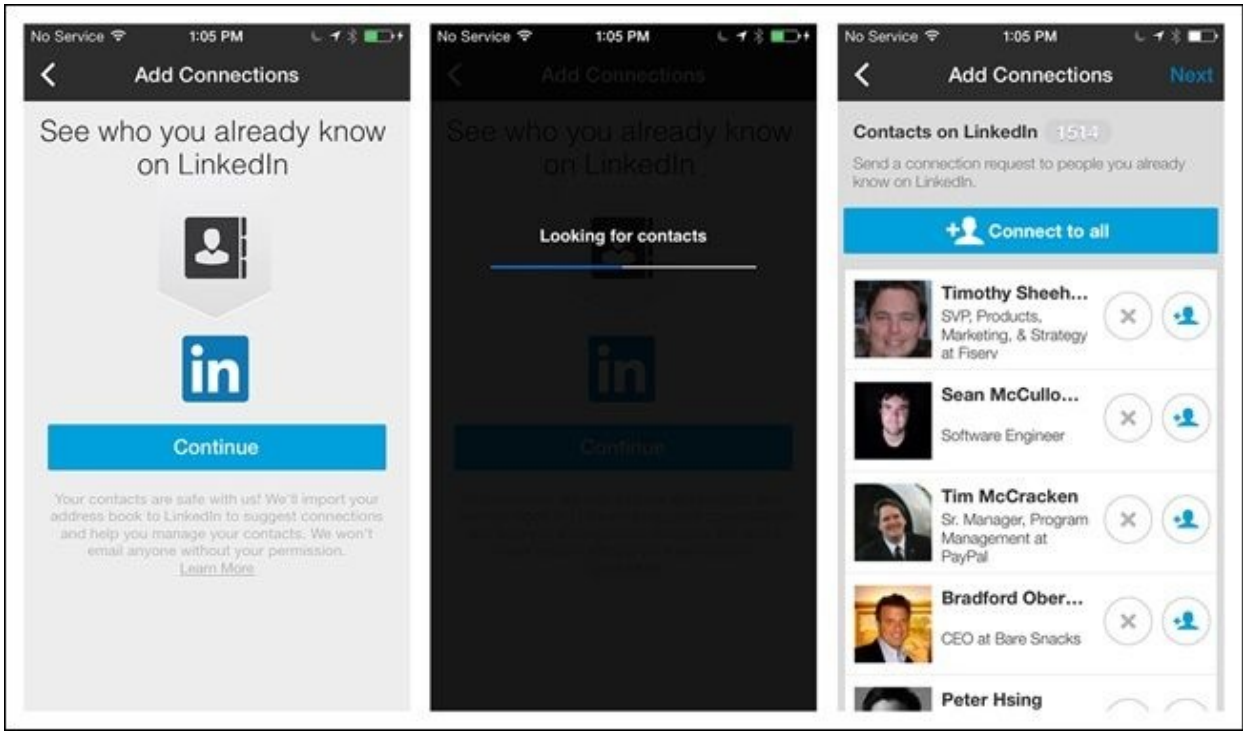



*Figure 8-4. LinkedIn for iOS: accepting a connection serves up suggestions for others to connect with*

Looking again at MapMyFitness, I gave its Social Registration high marks for being short and sweet, but connections aren’t part of that flow. That’s not really an issue, though, because the app organically prompts for connection within normal user flows: from the main menu, through invitations before or after a workout, and through invitations in the Activity Feed.

**NOTE**

Offer connection points throughout the application. It might make sense to offer connection opportunities during Registration, and/or as part of one or more other user flows.

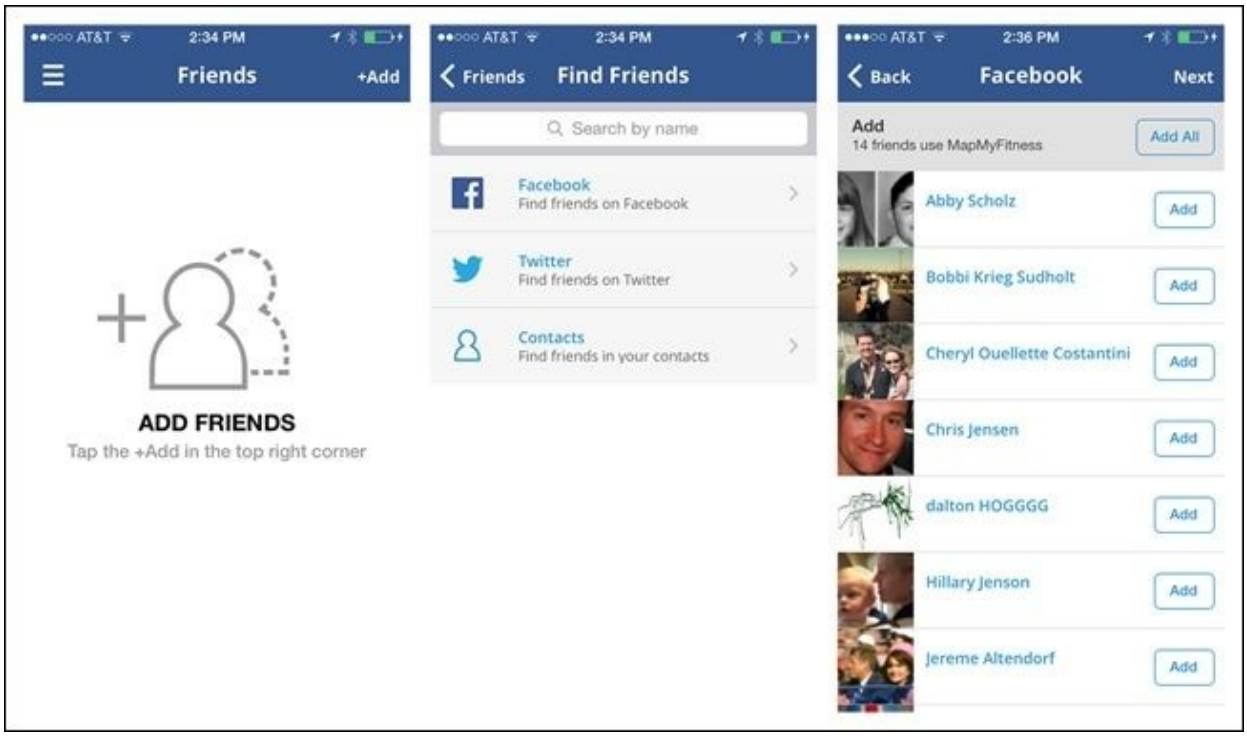



*Figure 8-5. MapMyFitness for iOS: find friends from main menu*

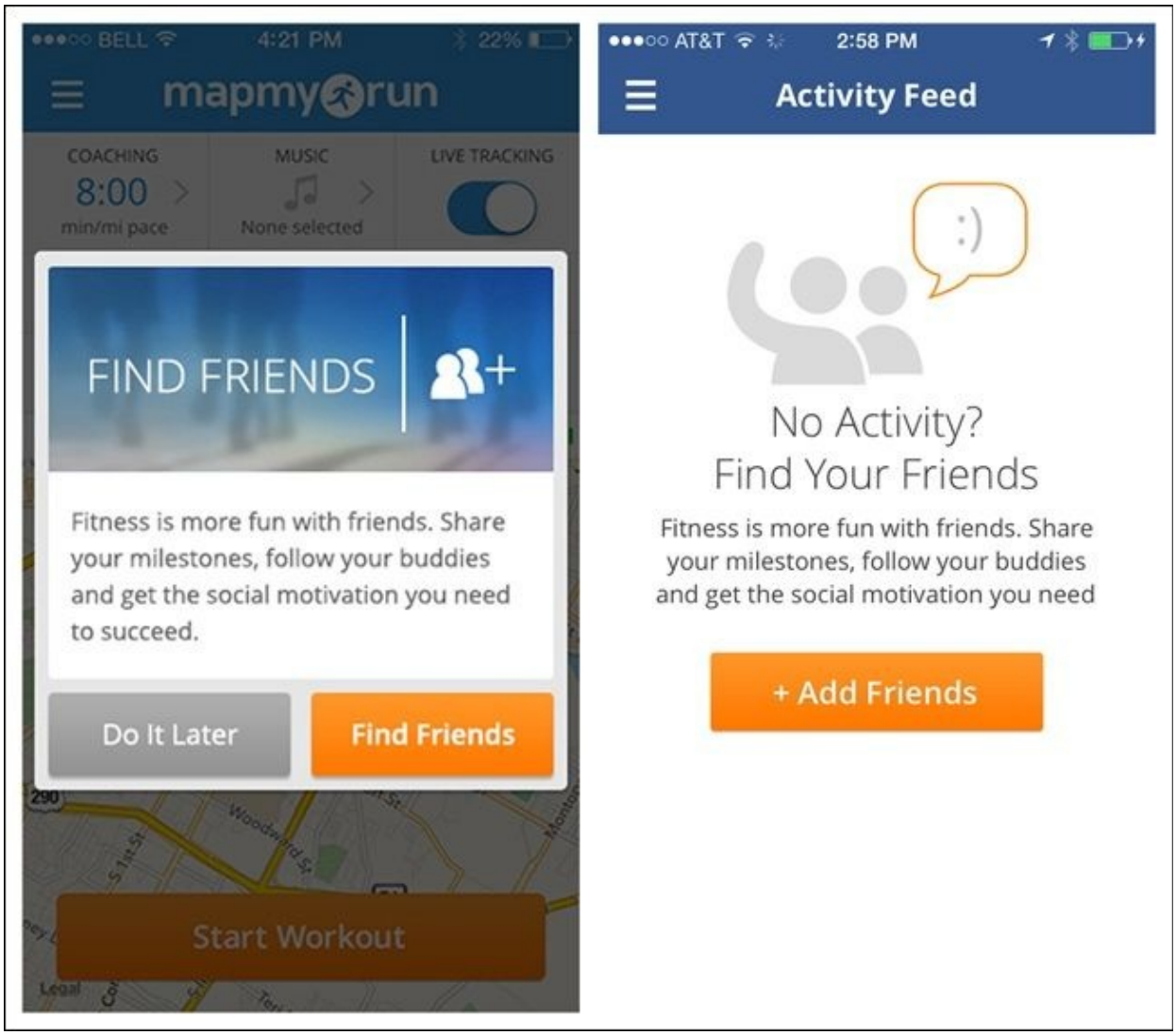



*Figure 8-6. MapMyFitness for iOS: invitations to connect pre-or post-workout, and in the Activity Feed*

# Following

All of the previous examples show how to enable bulk connections. But you'll probably also want to offer a one-to-one connection option. Thanks to Twitter, *Following* has become the generic term for this. In the Twitter and Zoomingo apps, the Following flow works like this: I'm browsing, I see something interesting (a tweet or a coupon), I look at who shared it, I check out what else that person has shared, and if what I see appeals to me I follow that person.


*Figure 8-7. Twitter for iOS: the Following flow lets you see what else a user has shared*

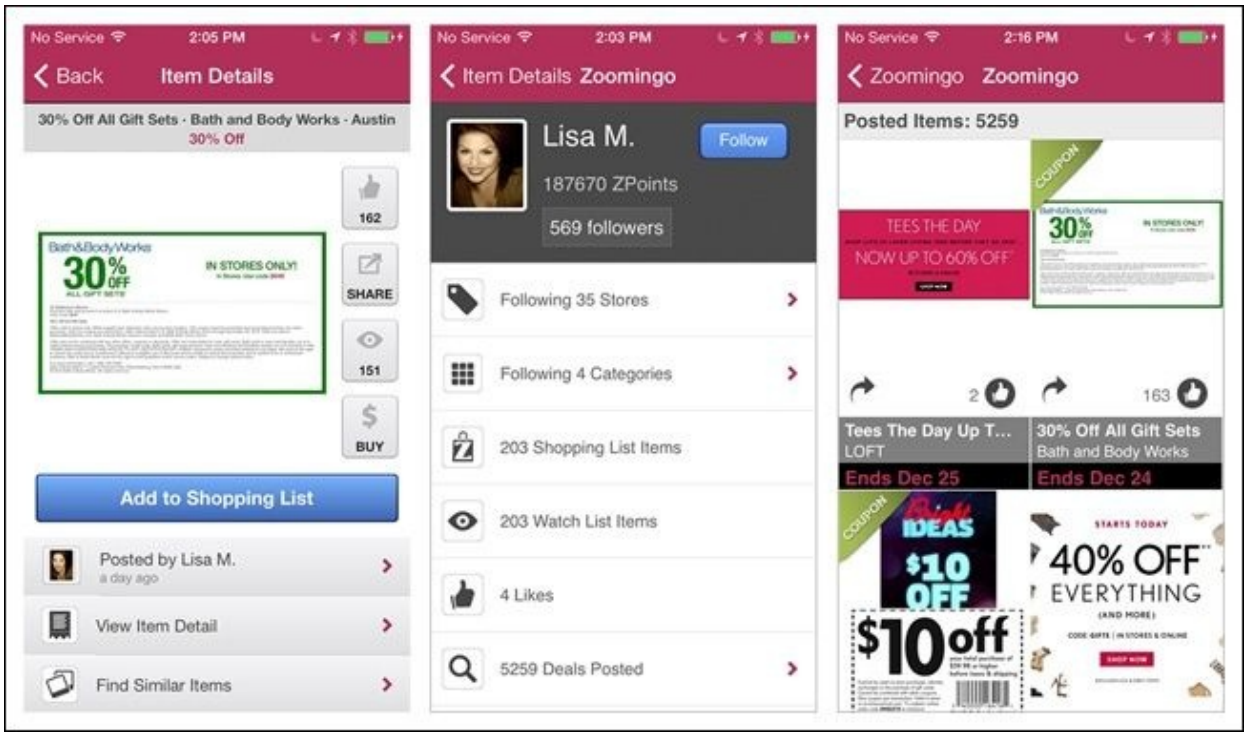



*Figure 8-8. Zoomingo for iOS: the Following flow highlights the user's reach within the network*

Pose makes Following a feature that drives signups. In its flow, the home page has a Featured Poser and a Follow call to action. If I want to see the Featured Poser's collection and profile, I'll need to register, if I haven't already.

**NOTE**

Offer Following at key points in the browsing flow to increase user engagement. Make the Follow button a prominent call to action on the Profile screen.

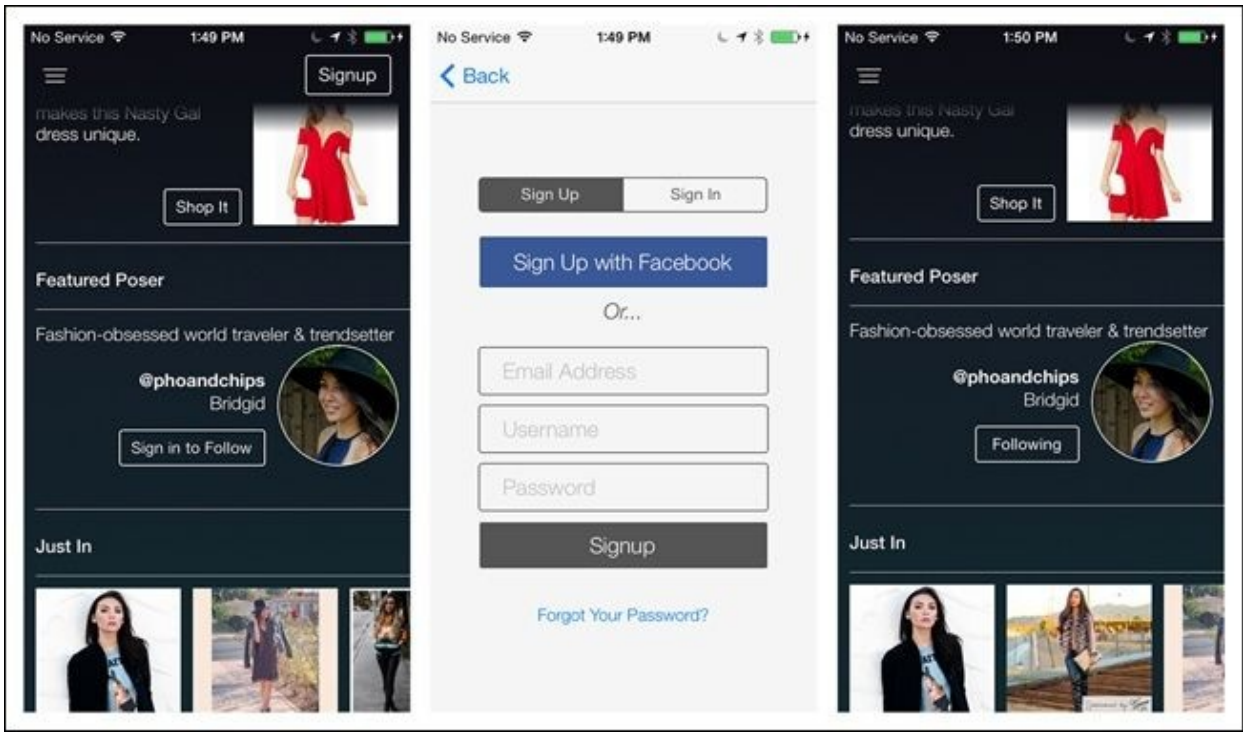



*Figure 8-9. Pose for iOS: Featured Poser makes Following a conversion tactic*

# Profiles

Although Profiles aren't unique to social platforms, they are one of the key elements of all social networks. The best profile pages "introduce" the person with a photo, short bio, and lists of interests. They include an activity feed or another indicator of the profilee's level of engagement, show that person's reach (how popular, influential, or active she is in the network), and offer a way to follow, add, or otherwise connect with her.

Apps like Foodspotting and Foursquare use statistics to show how engaged the person is in the community. This makes sense, since the relationships in these two apps are based on credibility and trust.


*Figure 8-10. Foodspotting and Foursquare for iOS: statistics communicate engagement and reach*

Pose and Pinterest are more visual networks, based on members' shared tastes. No surprise, then, that photos of shared content are the primary focus in their Profile screen designs.

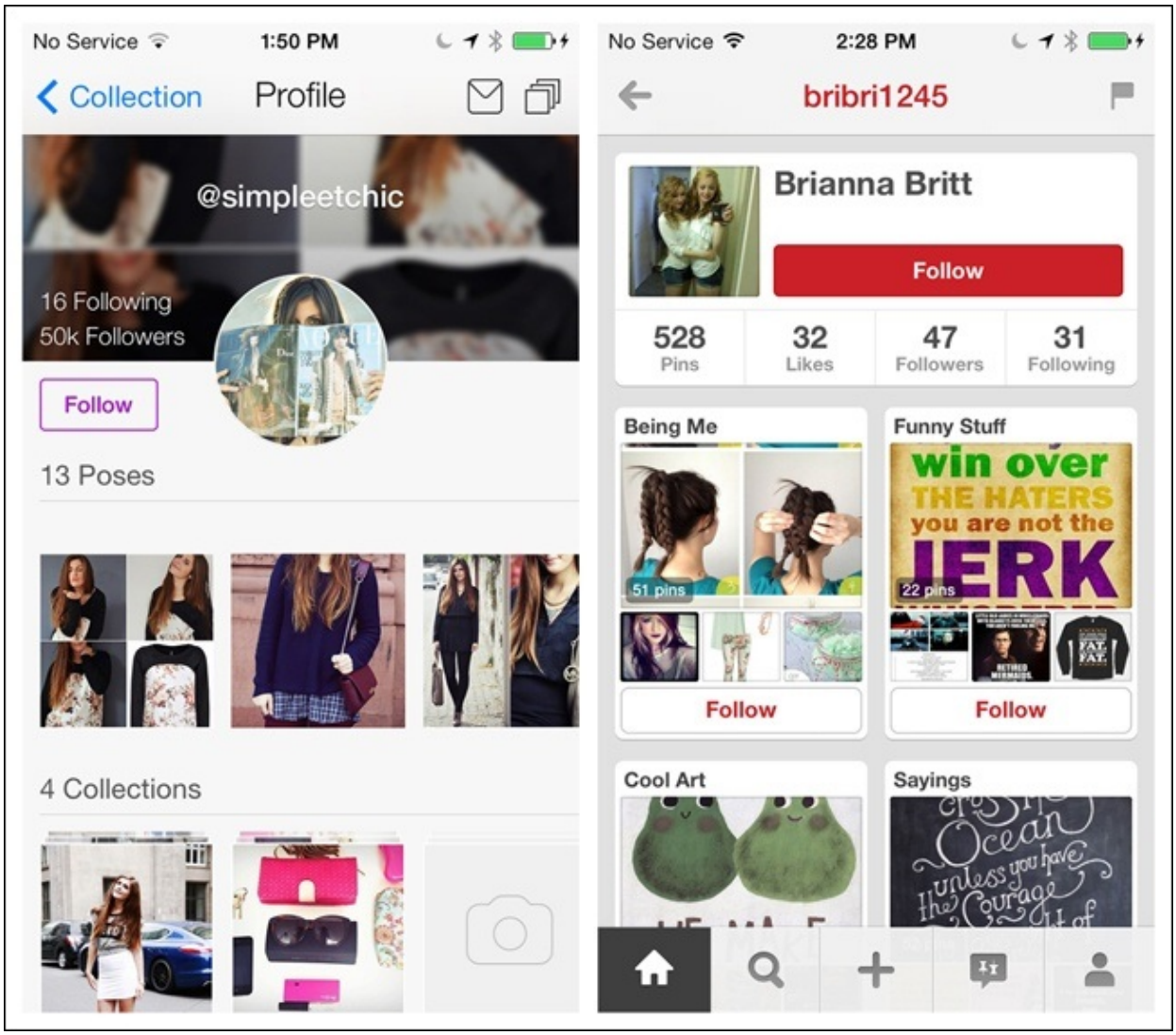



*Figure 8-11. Pose and Pinterest for iOS: shared content is prominent in Profiles*

Sometimes a network touches so many aspects of a person's life that a Profile can't be limited to a single screen. Instagram and Google+ use tabs for the different facets of a Profile, whereas LinkedIn uses a list to drill into more details.

**NOTE**

Profiles are the heart of social networks. Ultimately, the design should surface just enough information to make people want to connect with one another and build relationships within the network.

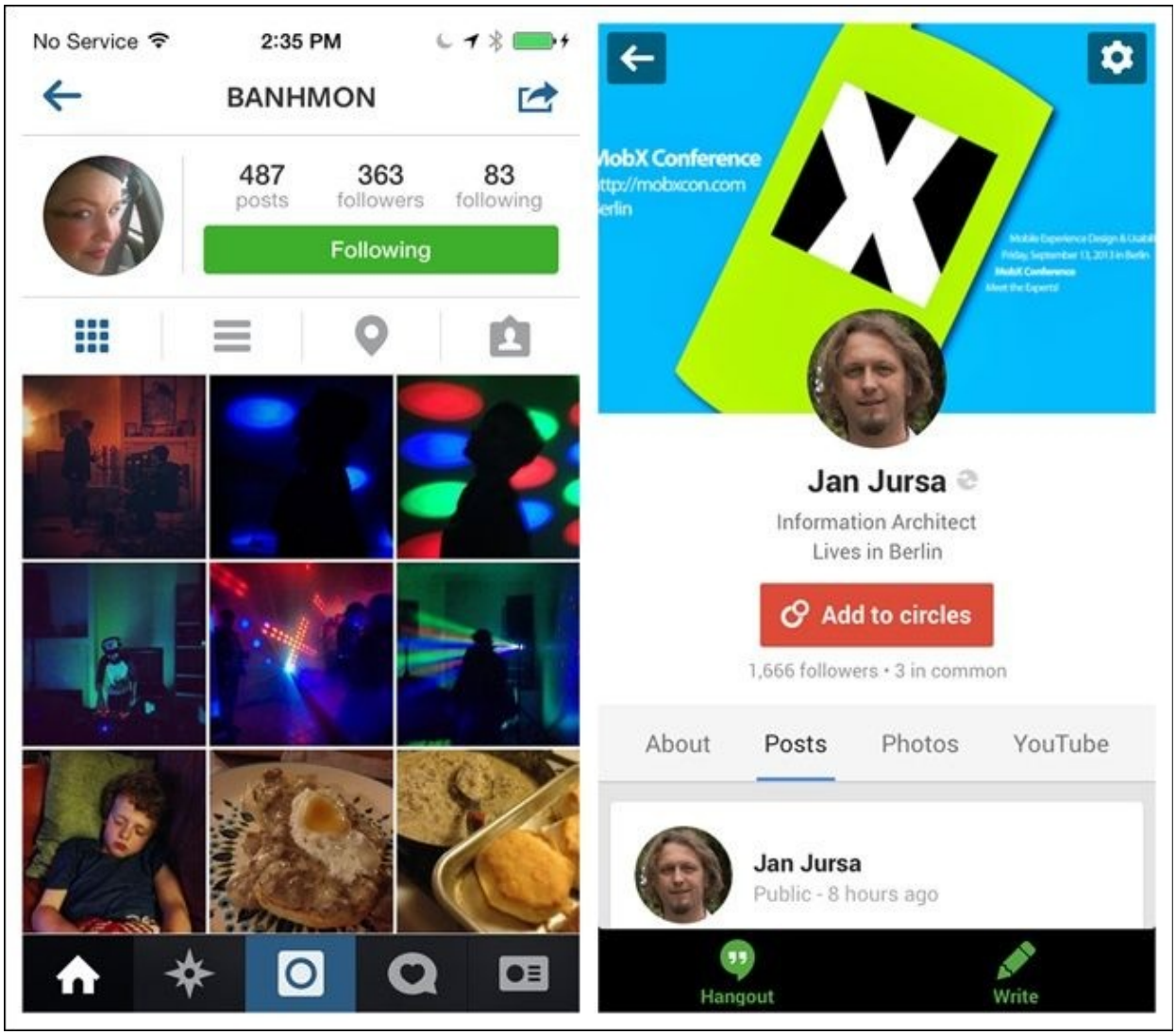



*Figure 8-12. Instagram and Google+: tabbed Profiles*

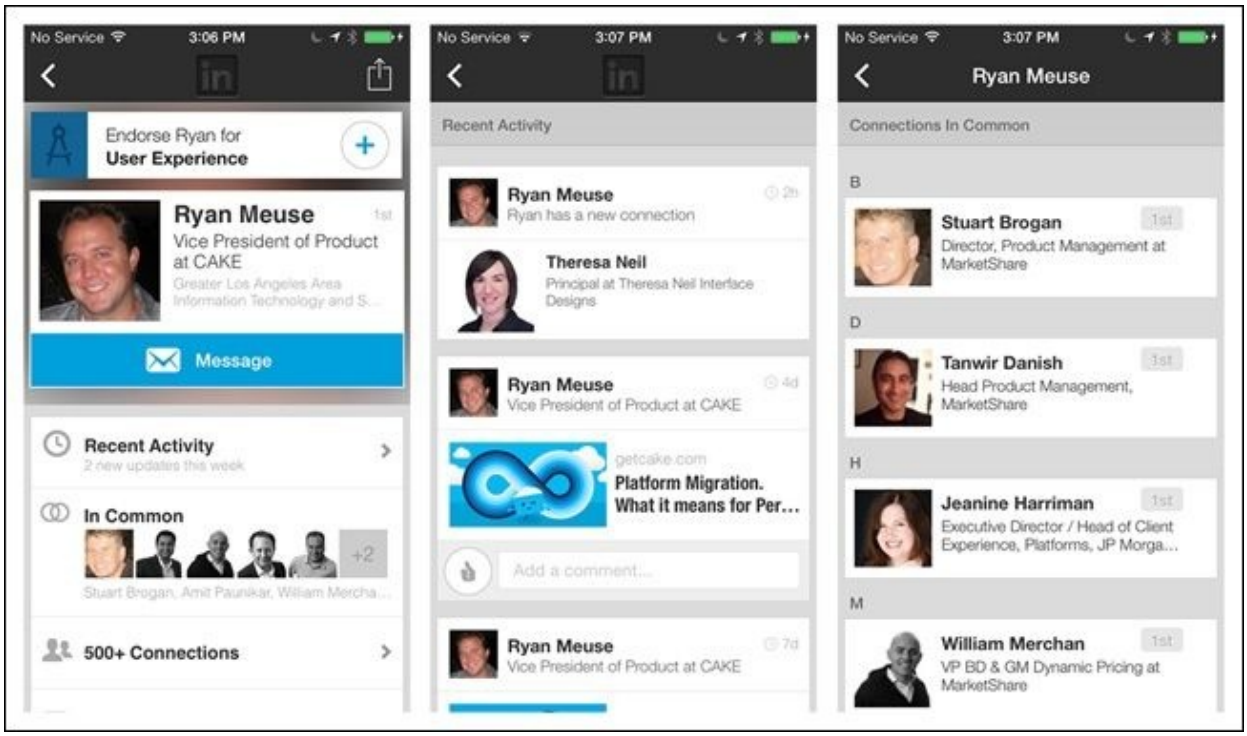



*Figure 8-13. LinkedIn for iOS: list design for more Profile details*

## Groups

Some Groups are implicit within social networks, while others need to be created by users. Group.me has a super simple interface for creating a Group. Start with a Group name, and then use dynamic search to start adding contacts.

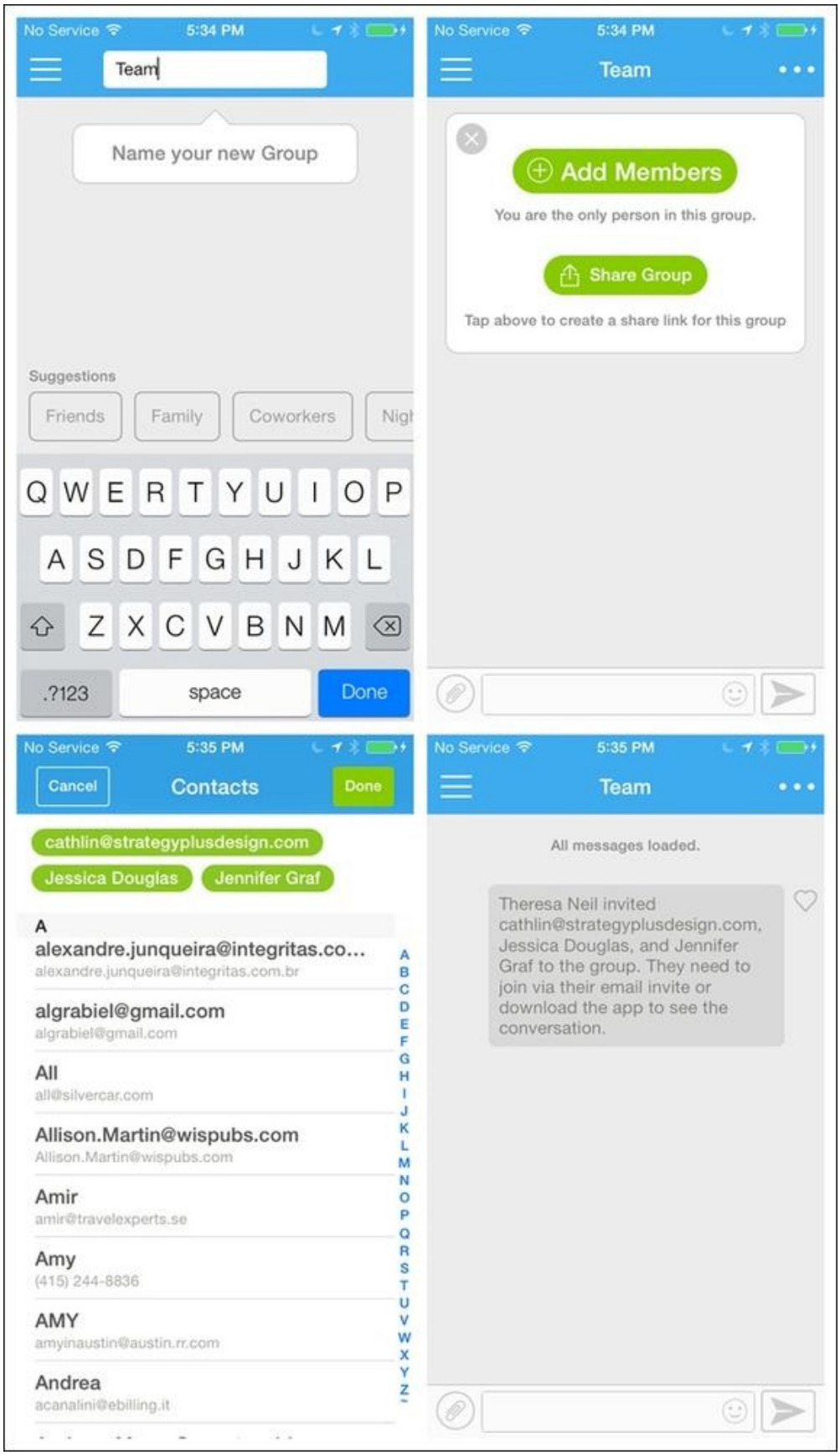

No Service
5:34 PM
Team
Name your new Group
Suggestions
Friends
Family
Coworkers
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
.?123
space
Done
No Service
5:34 PM
Team
Add Members
You are the only person in this group.
Share Group
Tap above to create a share link for this group
No Service
5:35 PM
Cancel
Contacts
Done
cathlin@strategyplusdesign.com
Jessica Douglas
Jennifer Graf
A
alexandre.junqueira@integritas.co...
alexandre.junqueira@integritas.com.br
algrabiel@gmail.com
algrabiel@gmail.com
All
all@silvercar.com
Allison.Martin@wispubs.com
Allison.Martin@wispubs.com
Amir
amir@travelexperts.se
Amy
(415) 244-8836
AMY
amyinaustin@austin.rr.com
Andrea
acanalini@ebilling.it
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
No Service
5:35 PM
Team
All messages loaded.
Theresa Neil invited cathlin@strategyplusdesign.com, Jessica Douglas, and Jennifer Graf to the group. They need to join via their email invite or download the app to see the conversation.

*Figure 8-14. Group.me for iOS: dynamic search aids Group creation*

Once added to the Group, an invitation will be sent to the members you selected. In this example from Facebook, my invitation appeared as an in-app notification.

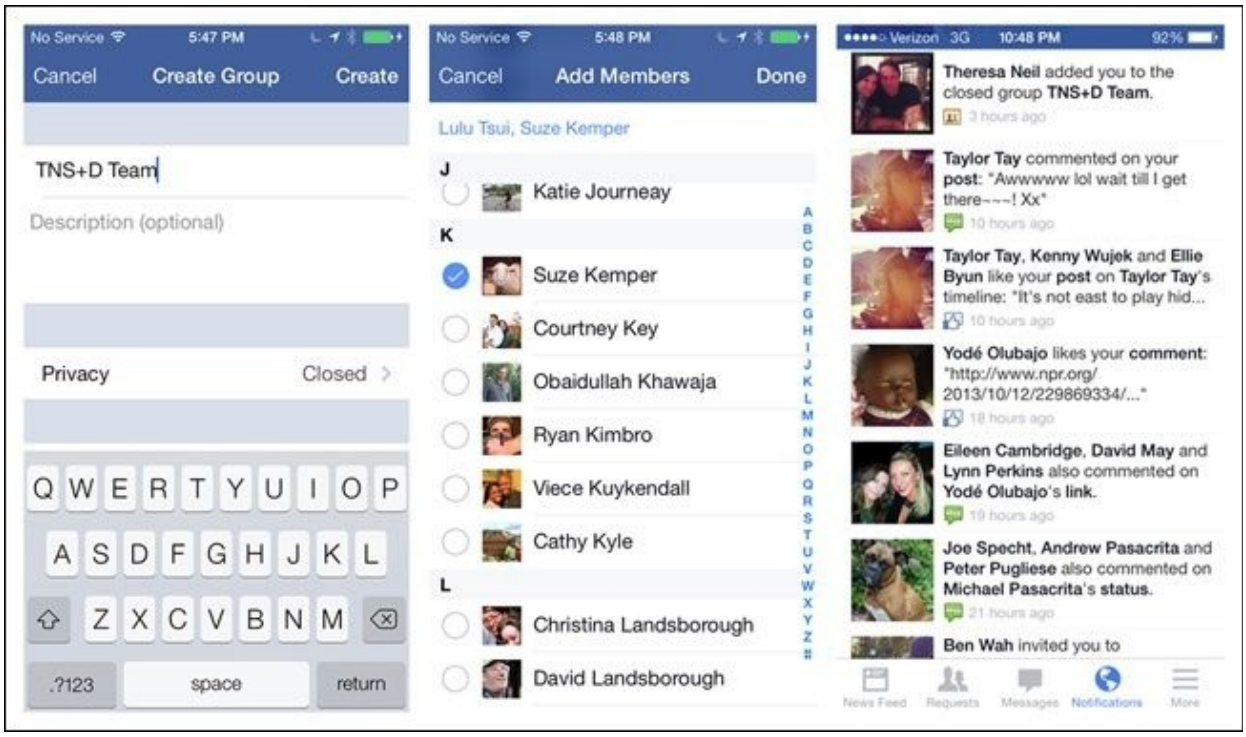



*Figure 8-15. Facebook for iOS: create Group and add members, who then receive in-app notifications*

LinkedIn provides recommendations for “Groups You May Like” and allows you to join a Group by tapping the + button. Many Groups require a Group administrator’s approval to join. In the example shown here, I added the Boxes and Arrows Group and received feedback that my request was being processed. On my Profile page, I can easily access the 13 Groups I already belong to.

If your network relies on explicit Group creation, make the flow super easy and integrate the invitation into the application.

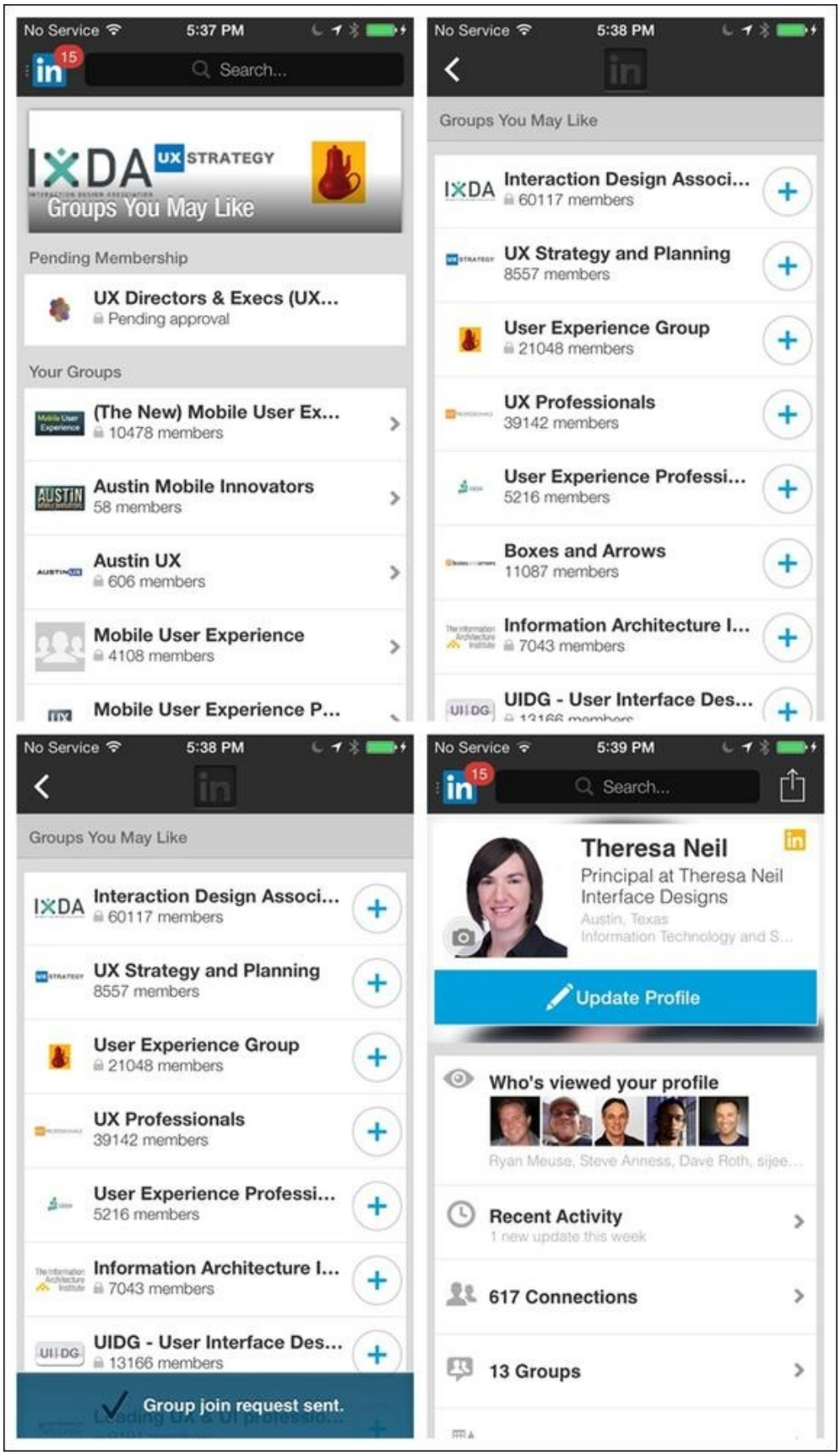

No Service
5:37 PM
15
Search...
IXDA
UX STRATEGY
Groups You May Like
Pending Membership
UX Directors & Execs (UX...
Pending approval
Your Groups
(The New) Mobile User Ex...
10478 members
Austin Mobile Innovators
58 members
Austin UX
606 members
Mobile User Experience
4108 members
Mobile User Experience P...
No Service
5:38 PM
Groups You May Like
Interaction Design Associ...
60117 members
UX Strategy and Planning
8557 members
User Experience Group
21048 members
UX Professionals
39142 members
User Experience Professi...
5216 members
Boxes and Arrows
11087 members
Information Architecture I...
7043 members
UIDG - User Interface Des...
No Service
5:38 PM
Groups You May Like
Interaction Design Associ...
60117 members
UX Strategy and Planning
8557 members
User Experience Group
21048 members
UX Professionals
39142 members
User Experience Professi...
5216 members
Information Architecture I...
7043 members
UIDG - User Interface Des...
13166 members
Group join request sent.
No Service
5:39 PM
15
Search...
Theresa Neil
Principal at Theresa Neil Interface Designs
Austin, Texas
Information Technology and S...
Update Profile
Who's viewed your profile
Ryan Meuse, Steve Anness, Dave Roth, sijee...
Recent Activity
1 new update this week
617 Connections
13 Groups

*Figure 8-16. LinkedIn for iOS: a more formal process for joining Groups*

## Gamification

It seems like everyone wants Gamification (game-like elements in a nongame) in their applications these days. What they really want is a *sticky* app — one that people return to time and again. But true Gamification takes more than offering a couple of badges. To understand the science and psychology behind Gamification, I highly recommend you read three books: *Designing for Behavior Change*, by Stephen Wendel (O'Reilly, 2013); *How to Get People to Do Stuff*, by Susan Weinshenk (New Riders, 2013); and *Seductive Interaction Design*, by Stephen Anderson (New Riders, 2011).

Beware: Gamification can backfire when implemented poorly. Let's take a look at Audible and what I call the Forget-Me-Not anti-pattern. I'm a long time Audible subscriber, having purchased hundreds of audio books. I listen to Audible in my car, on my computer, at the gym, on a variety of devices. But despite my loyalty and longevity, look at my progress: I've got no badges and the app ranks me at the "newbie" level!

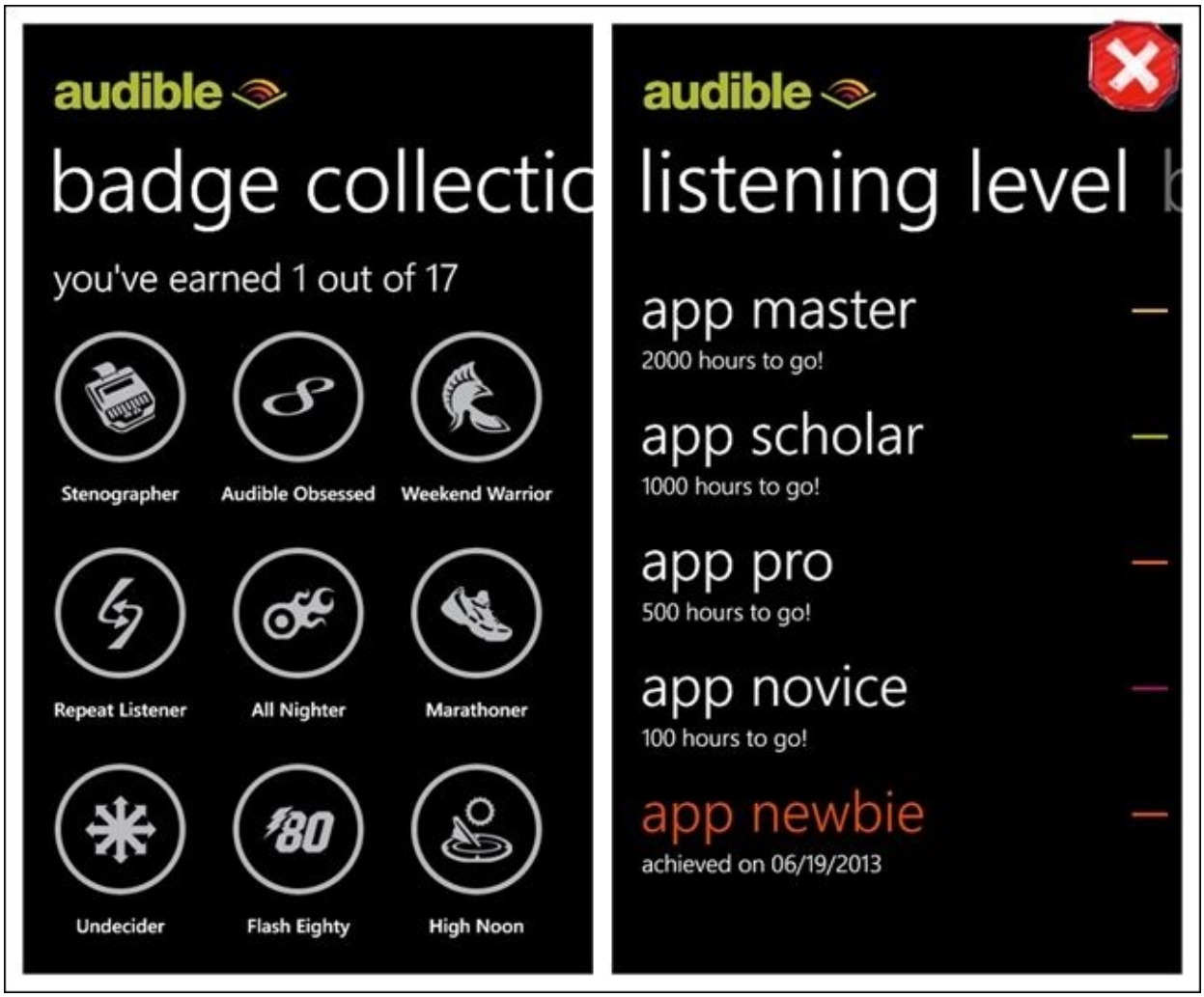



*Figure 8-17. Audible for Windows Phone: a newbie despite long-term loyalty*

Why? Because this takes into account only what I have done with the mobile app on one particular device. With Gamification, it is critical to meet users' expectations of interconnectivity and synchronization across devices. Kobo does a nice job of this with its Reading Life program, and also integrates the Connecting design pattern.

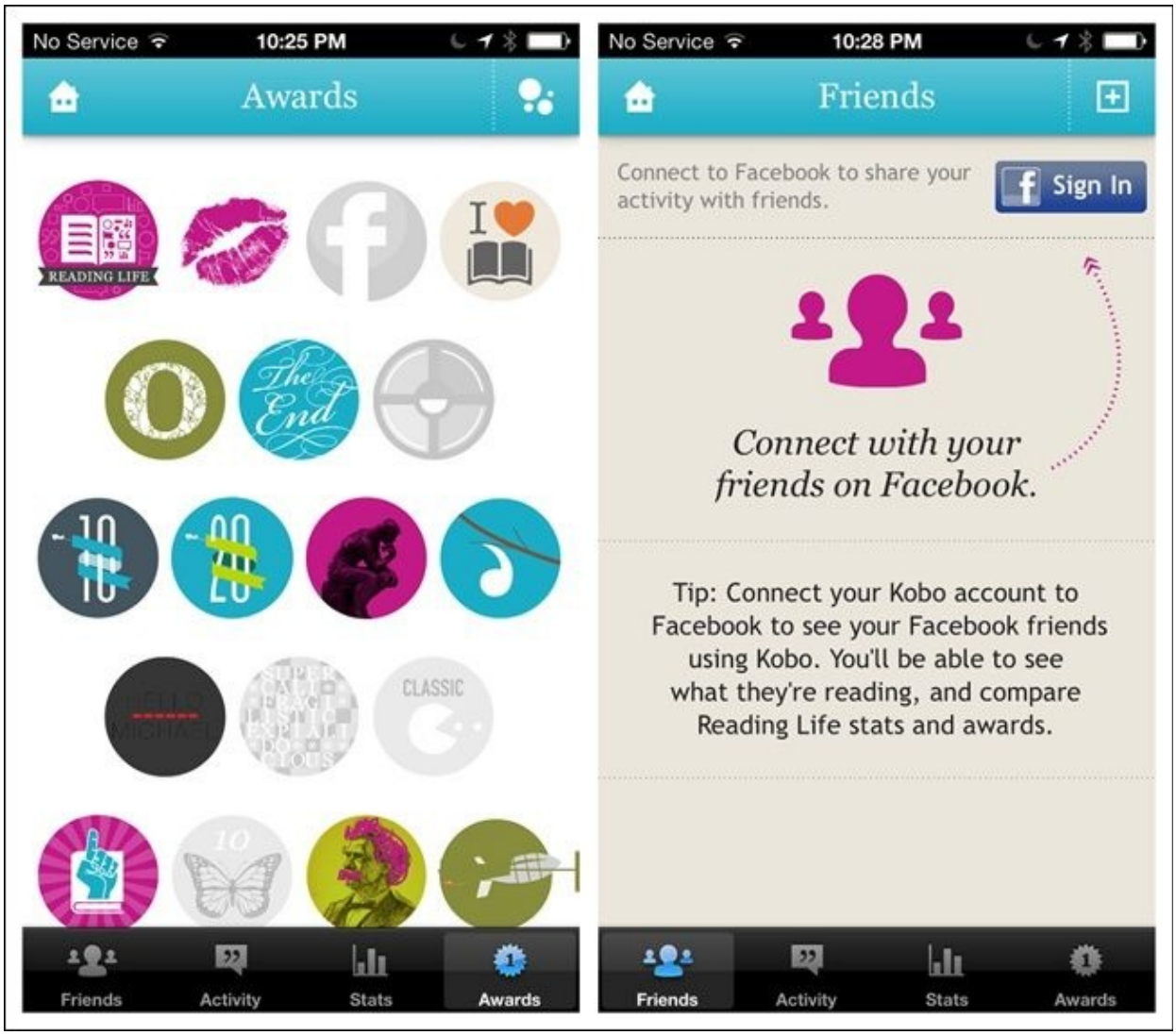



*Figure 8-18. Kobo for iOS: Reading Life rewards work across platforms*

Nike Training Club has dialed back the badges but continues to offer rewards, which are tied directly to self-specified training goals.

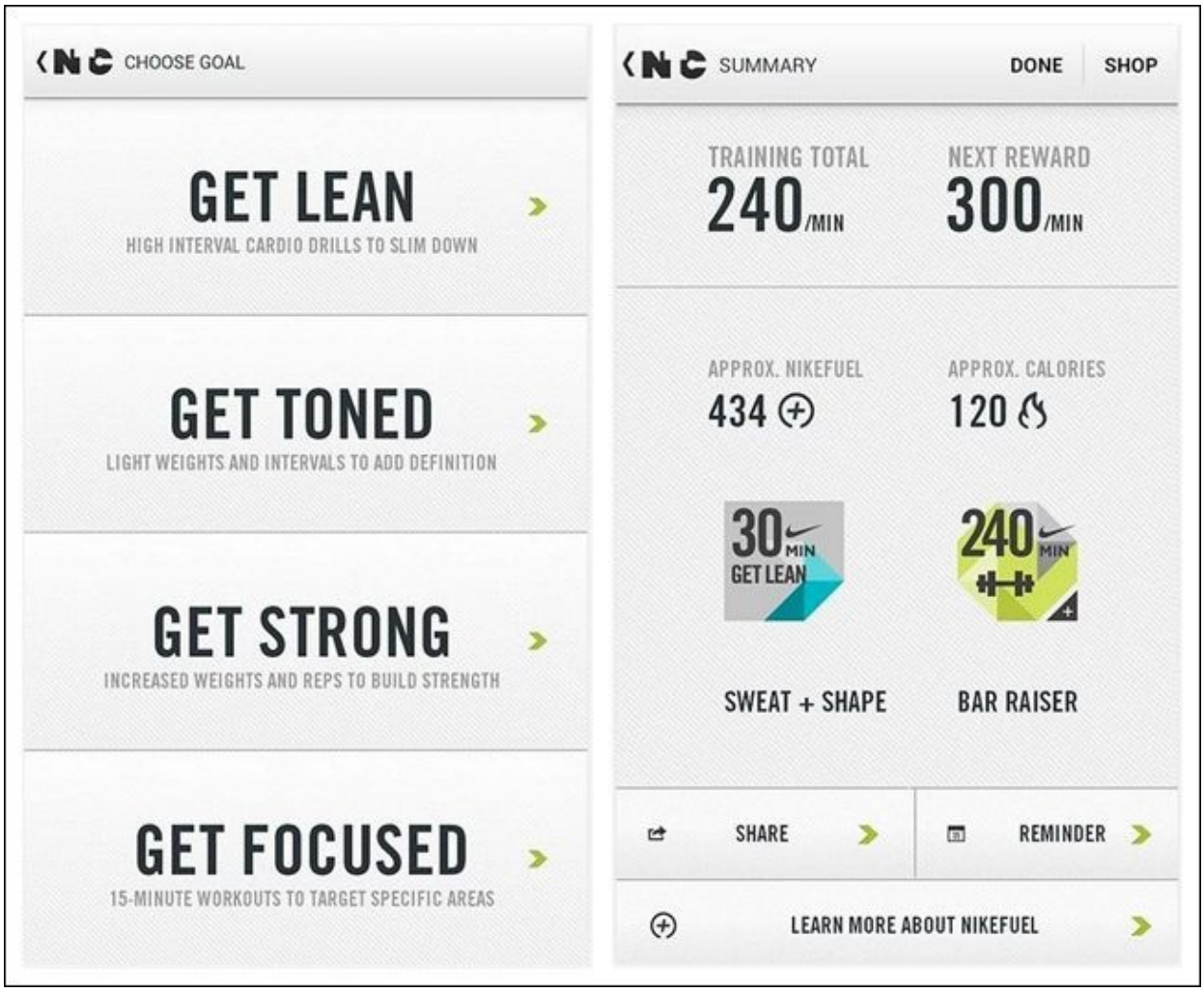



*Figure 8-19. Nike Training Club for Android: meet self-specified goals, and get rewards*

Noom uses the game concept of *leveling up* to encourage continued engagement and forward progress. It also uses a simple, well-known behavior modification technique: the element of surprise. Just as I completed the setup process, I was rewarded with an “overachiever bonus,” which inspired me to engage just a little more.

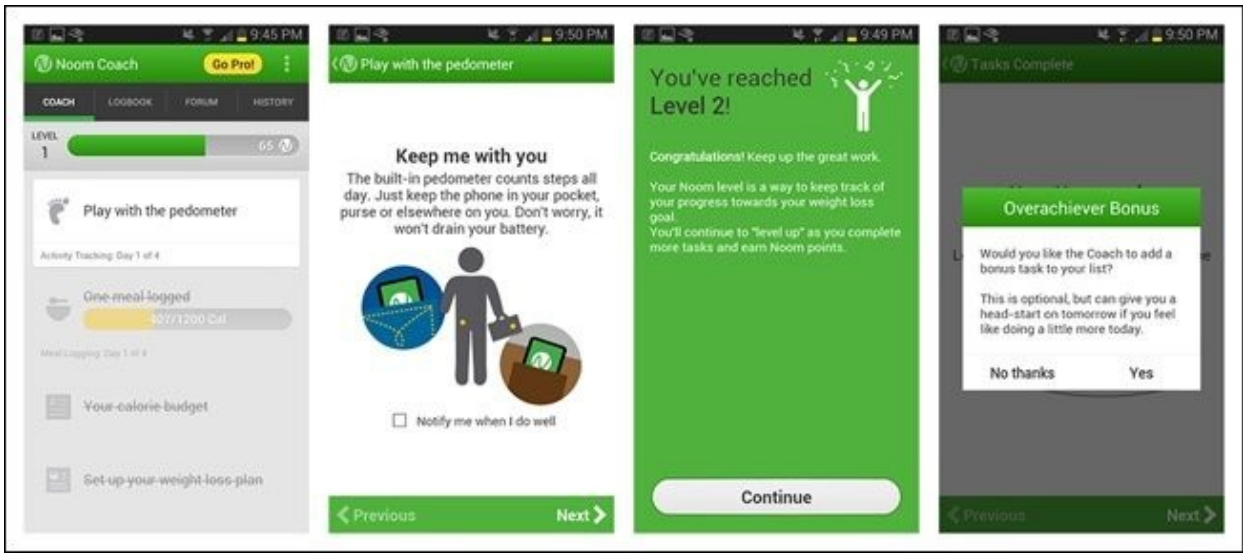



*Figure 8-20. Noom for Android: leveling up and surprises "gamify" the experience*

Other motivators in gamified apps include Leaderboards and Points. GasBuddy prominently features Top Spotters, and Duolingo shows stats for top performers.

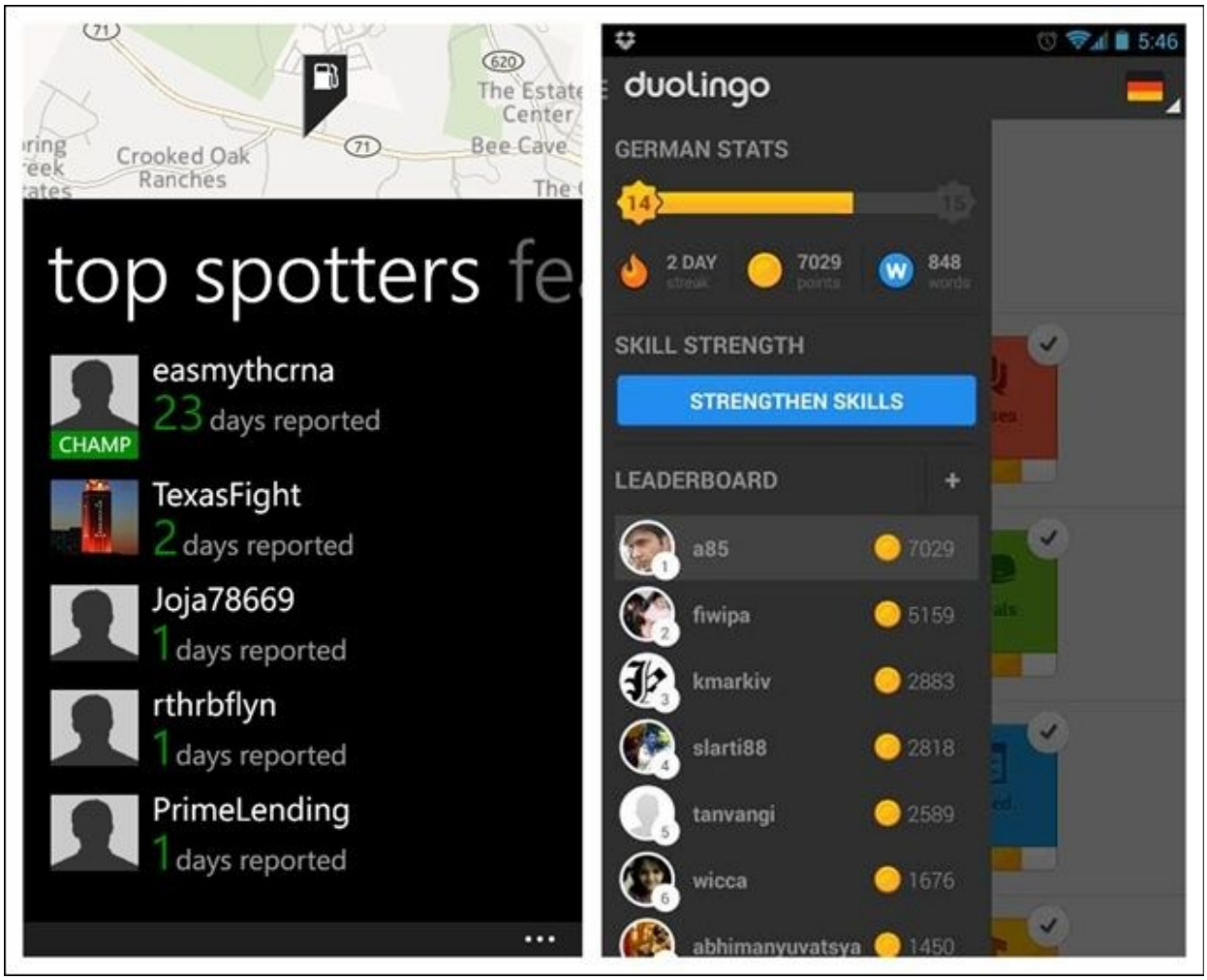



*Figure 8-21. GasBuddy and Duolingo for iOS: Leaderboards and reputation Points*

With Duolingo, users earn “Lingots” for successfully completing a skill set. They can then use the Lingots for shopping — say, to purchase a new outfit for the cute little owl instructor.

Compare the simplicity of Duolingo to Foodspotting’s lengthy and confusing reputation system (*http://www.foodspotting.com/about/faq*), where users have to earn points in order to award restaurants with blue ribbons (called “noms”). If you can’t explain the system in a sentence or two, it may be too complex to motivate users as much as you’d like.

**NOTE**

Understand the principles of Gamification and behavior change before trying these patterns. Keep it simple and fun, and don’t force Gamification where it isn’t appropriate.

## Lesson Complete! +10 XP
## Heart bonus! +2 XP


You're on track today. Good job!

Continue

## You Earned 5 Lingots


Learned the skill Basics 2


Reached level 3

Continue

## Spanish Skills

Basics 1

Phrases

Basics 2

Food
0/5

Animals
0/4

Plurals
0/1

Practice Weak Skills

## Shop

CLOSE

Earn lingots for achievements. Get fun things with them.

Balance 12

**Heart Refill**

Allows you to regain one heart lost during a lesson.

4 LINGOTS

**Formal Attire**

Learn in style, sporting accoutrements that any gentleman or lady would admire.

20 LINGOTS

**Champagne Tracksuit**

Learn in luxury. Duo will love the feel of 24 carat gold silk on his feathers.

30 LINGOTS

*Figure 8-22. Duolingo shows progress against goals and rewards success with Lingots*

# Chapter 9. Feedback and Affordance


*Feedback Patterns*

Error Messages, Confirmation, System Status

*Affordance Patterns*

Tap, Swipe/Flick, Drag

# Feedback Patterns

In its “Principles for Usable Design” (*http://www.usabilitybok.org/principles-for-usable-design*), the User Experience Professionals Association says that well-designed user interfaces should “provide appropriate, clear, and timely feedback to the user so that he sees the results of his actions and knows what is going on with the system.” Paraphrasing an old movie tagline, I’d shorten this to: “Feedback means users never have to say, ‘What’s happening?!’”

Feedback can vary from simple progress indicators and confirmation messages to more sophisticated animations and effects. Mobile feedback patterns include Error Messages, Confirmation, and System Status.

## Error Messages

Error Messages should be expressed in plain language (no codes), precisely indicate the problem, and suggest a constructive solution. Best practice is to make Error Messages highly visible onscreen, like in Soundwave and Pandora. These approaches are preferable to modal dialogs, since they may literally cover up the issue, as in Square Register and Pinterest.

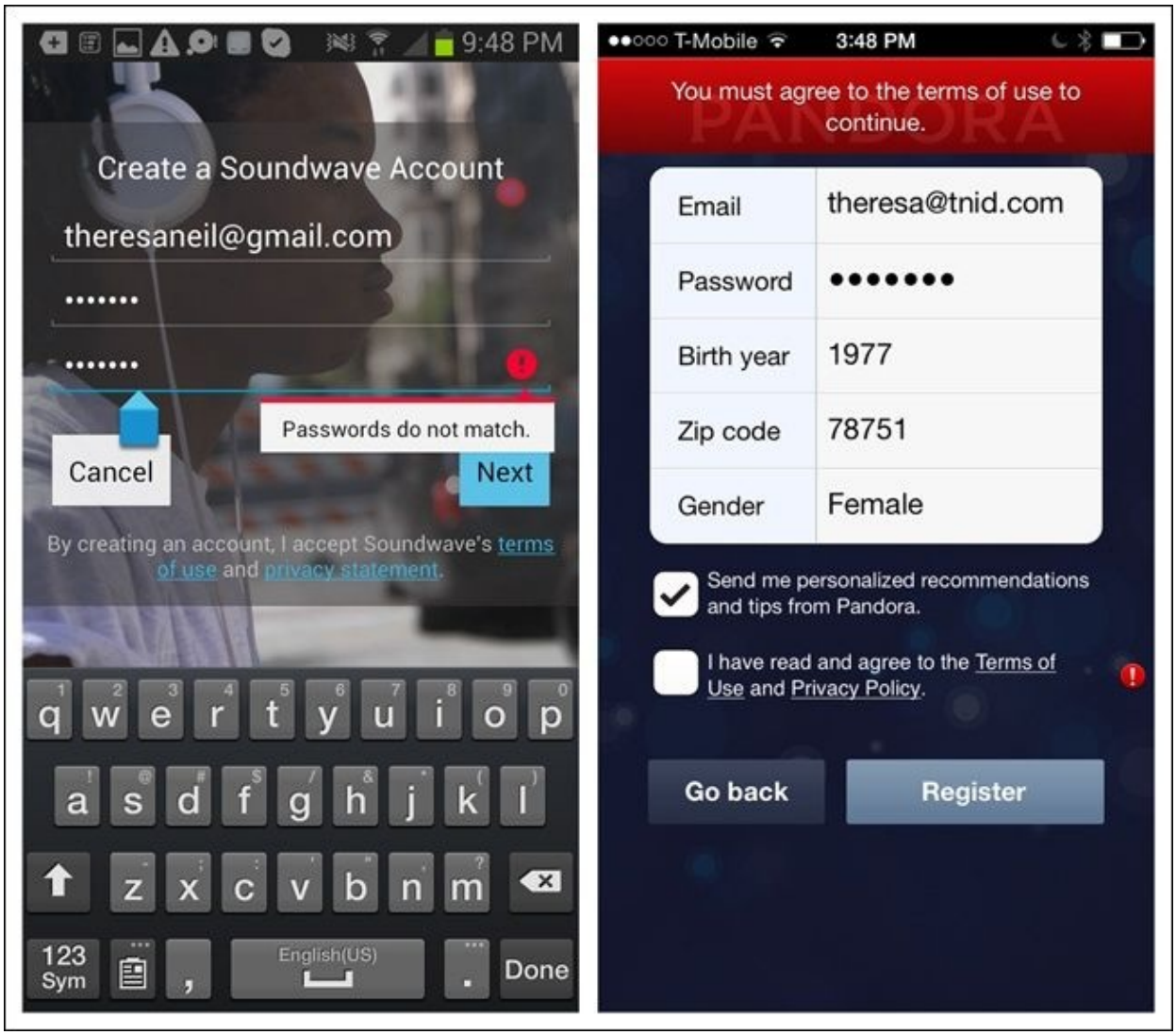



*Figure 9-1. Soundwave for Android and Pandora for iOS: highly visible Error Messages*

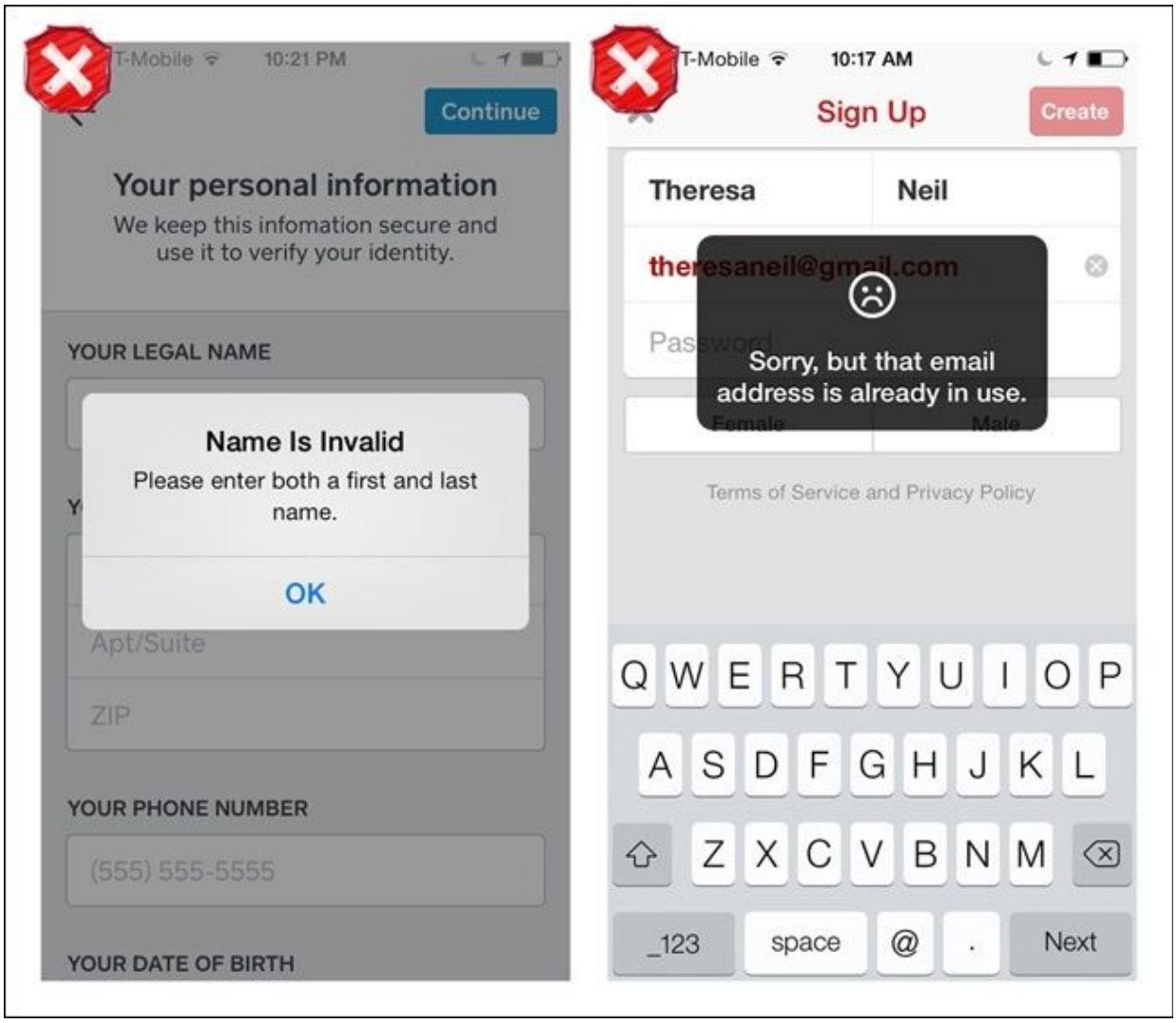



*Figure 9-2. Square Register and Pinterest for iOS: error dialogs hiding the errors*

If you can prevent the error, that's even better. See the Confirmation pattern for suggestions on providing inline feedback that prevents errors — and Error Messages.

**NOTE**

Explain the error in plain language and offer a solution for resolving it. Integrate error messaging into the screen design so messages leave errors visible, instead of covering them up.

## Confirmation

Do provide Confirmation when an action is taken. But don't use the Idiot Box anti-pattern, like Amazon for Android and HauteLook. Instead, look for ways to provide Confirmation feedback that don't disrupt the user flow. (See Chapter 11

for more on the Idiot Box anti-pattern.) For example, Amazon for iOS and Zappos both use animation to show an item being added to the cart, with the number of items in the cart updating.

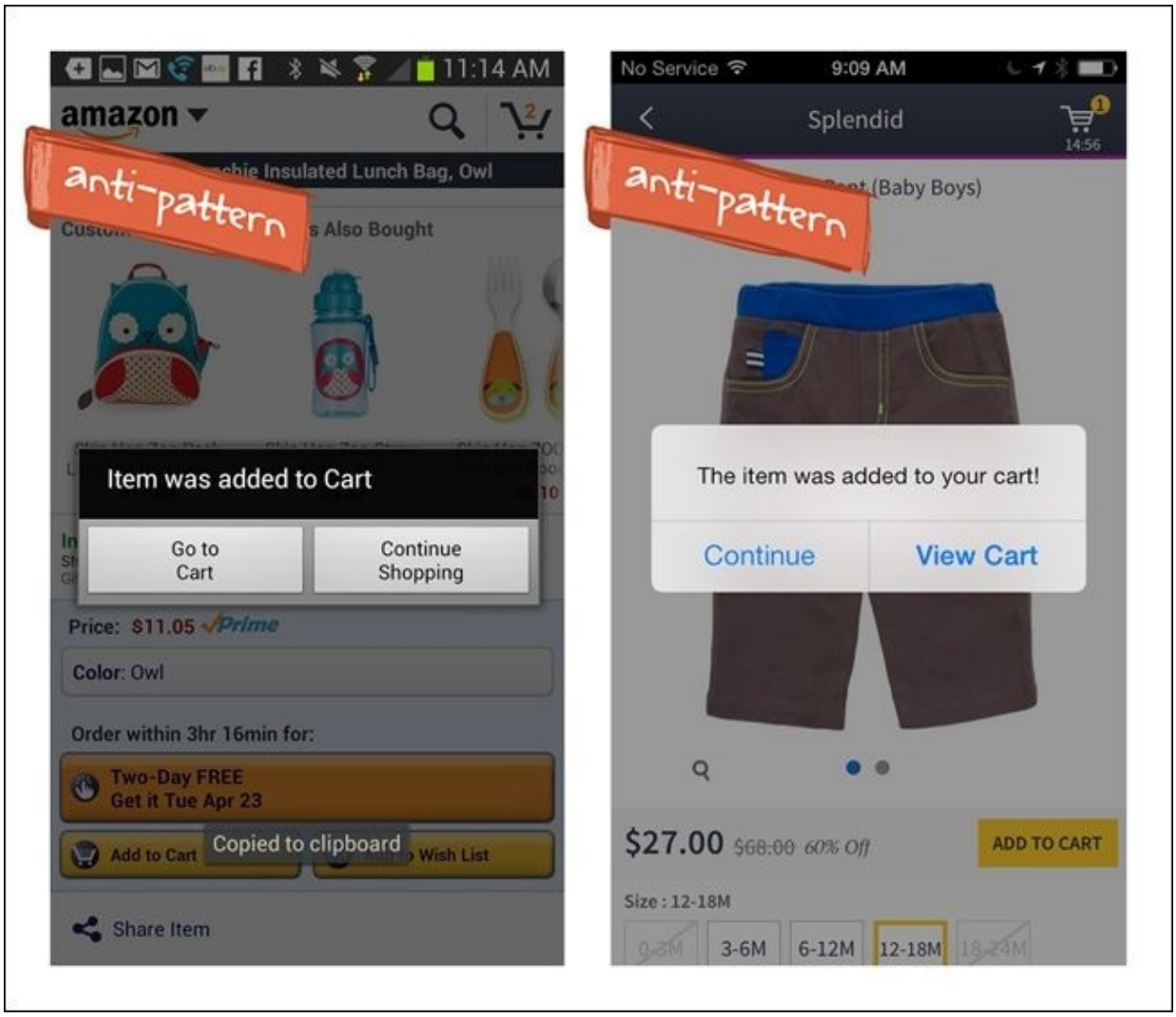



*Figure 9-3. Amazon for Android and HauteLook for iOS: Idiot Box Confirmation interrupts shopping flow*

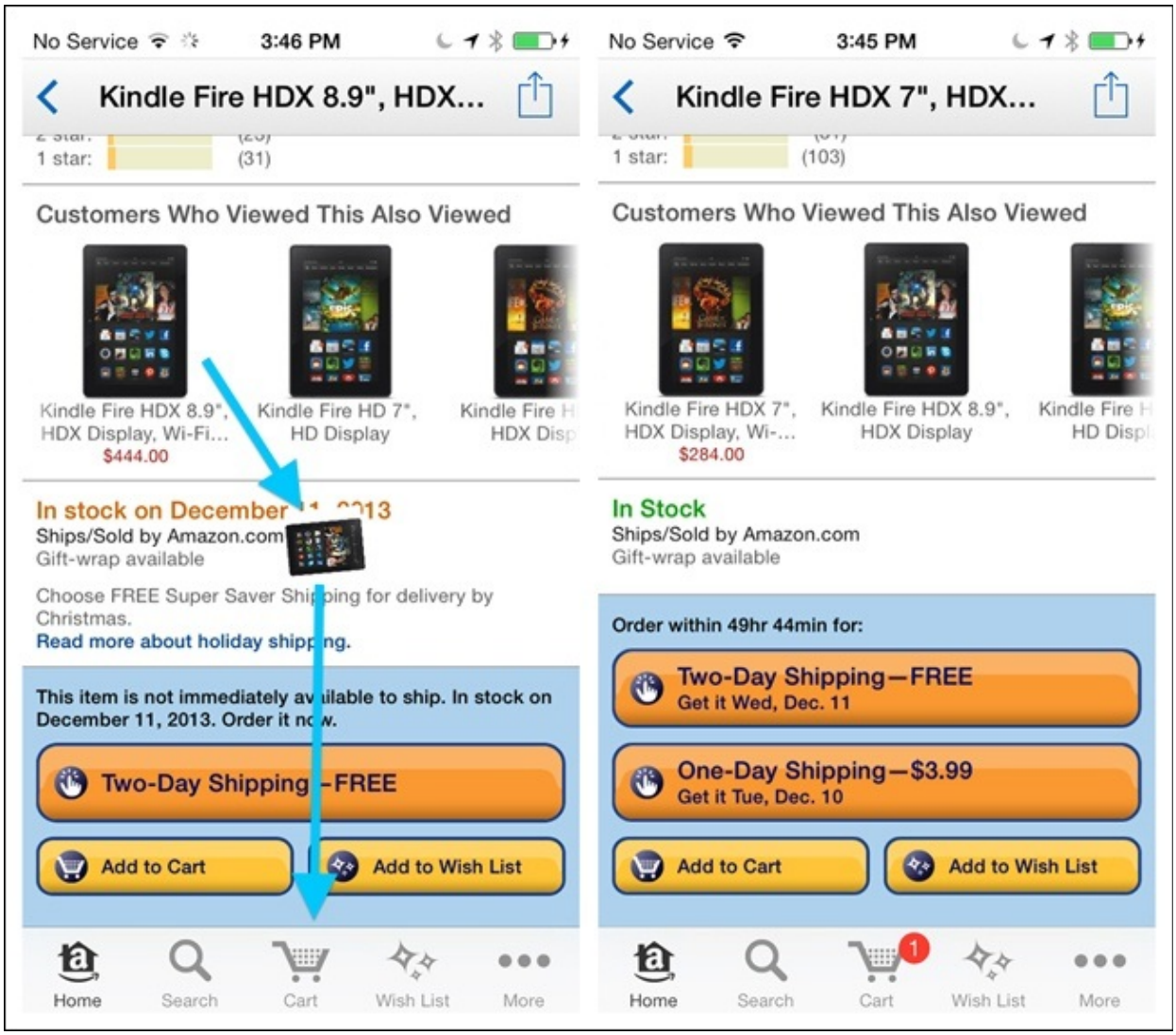



*Figure 9-4. Amazon for iOS: Kindle "falls" into the cart and the cart item number updates (http://www.youtube.com/watch?v=--JzNARO_eM)*

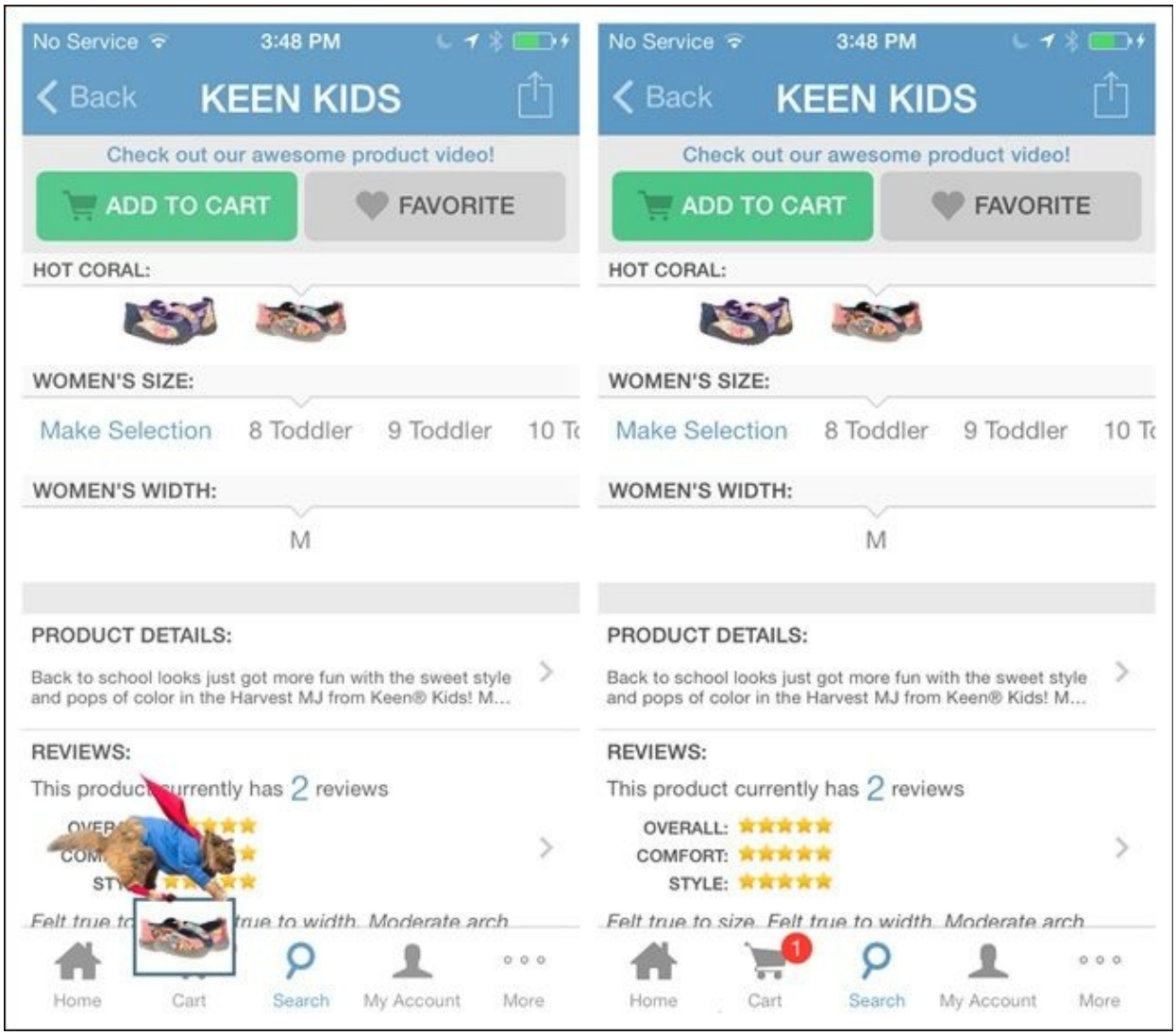



*Figure 9-5. Zappos for iOS: Super Cat "flies" an item into the cart, and the item number updates (http://www.youtube.com/watch?v=JxADy-3HOjc)*

Another option, used by Gilt and BestBuy, is to show a message near the cart, in addition to updating the number of cart items.

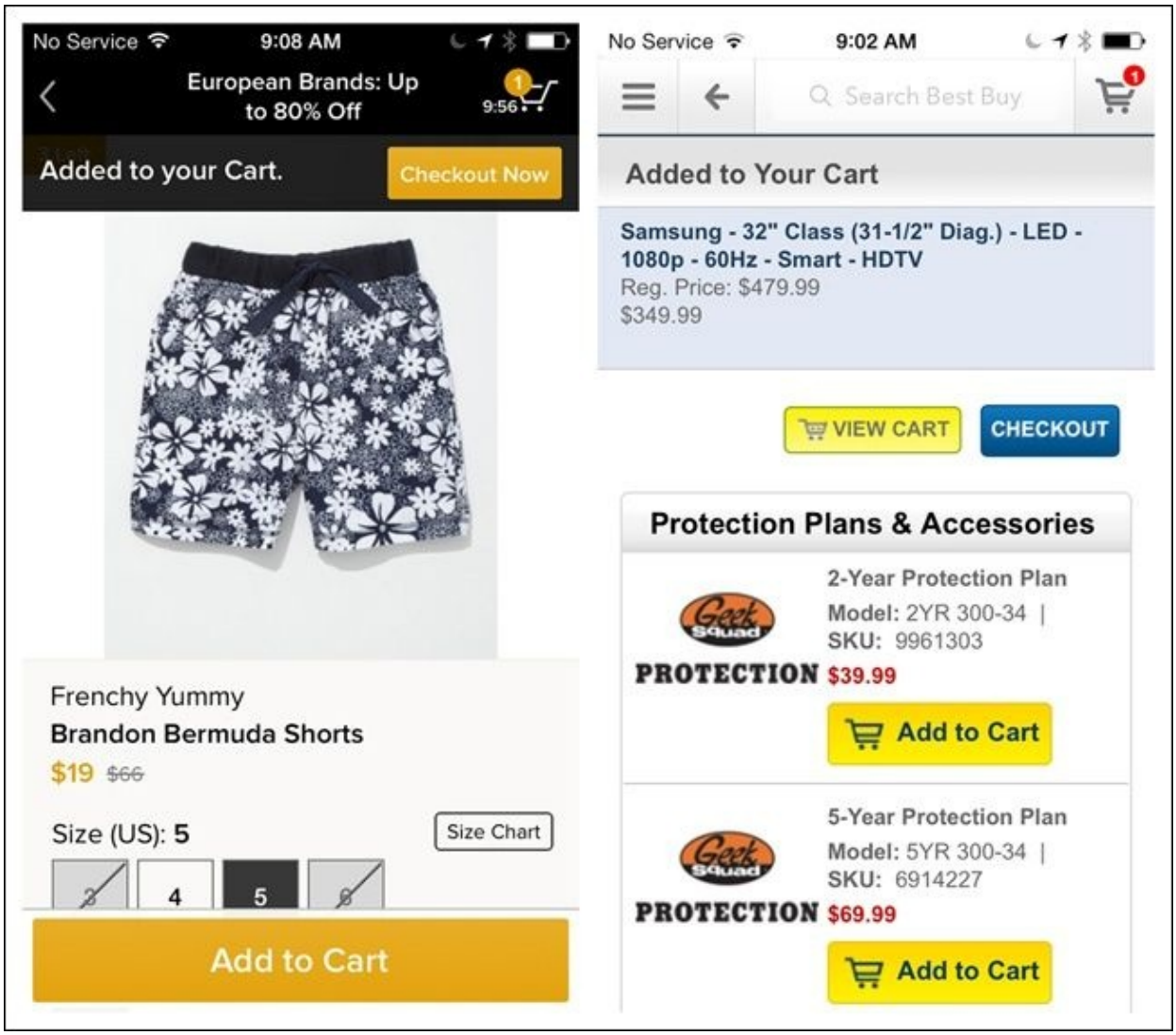



*Figure 9-6. Gilt and BestBuy for iOS: an in-screen message near the cart confirms items added*

These are shopping cart examples, but adding items to any type of list requires Confirmation. For example, RetailMeNot uses animation to show where the object is saved and, just as importantly, where it can be found later.

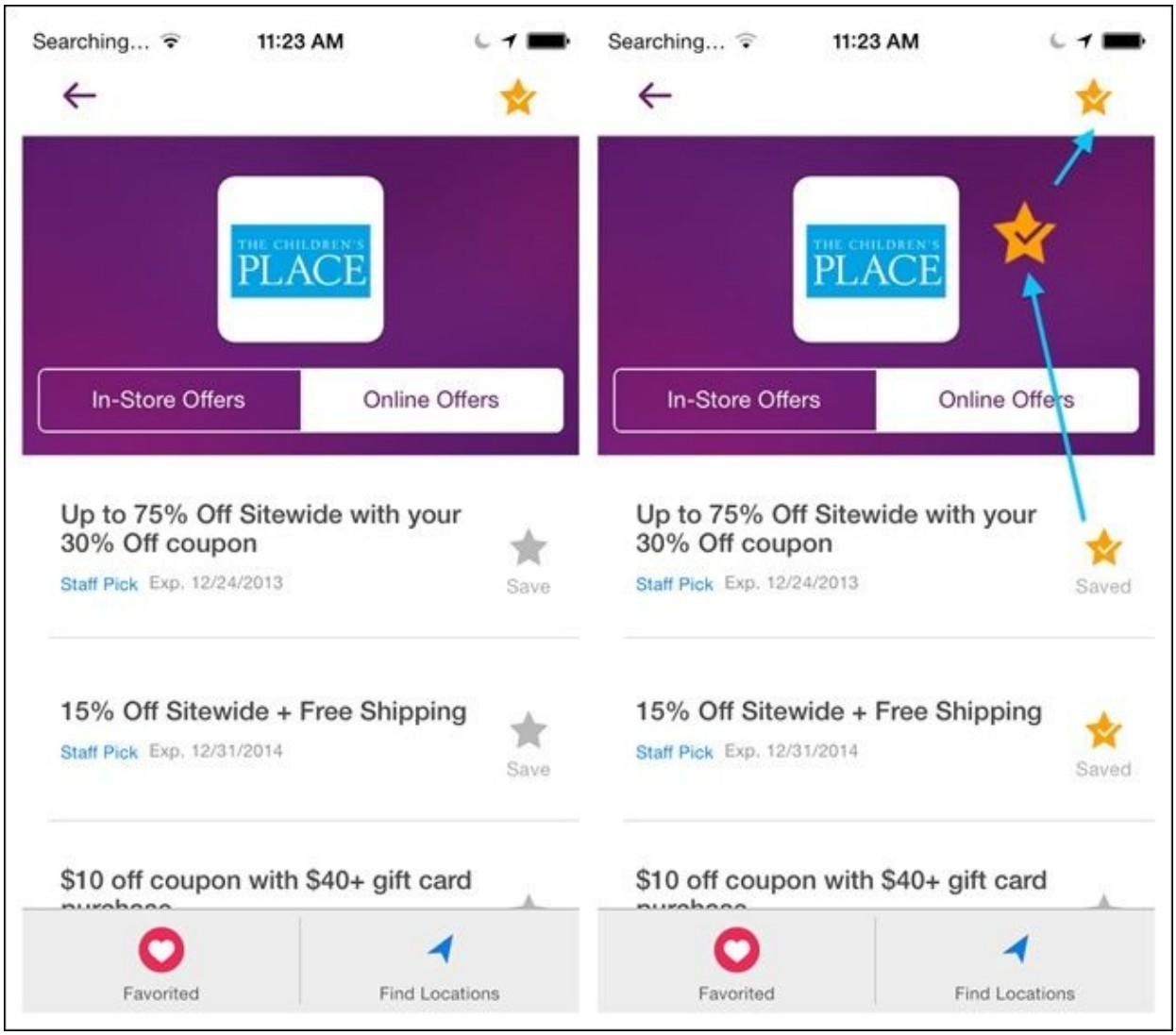



*Figure 9-7. RetailMeNot for iOS: animations show save plus where saved items can be accessed (http://www.youtube.com/watch?v=wFqxj1o4ueE)*

Inline feedback, as seen in Kik for iOS, is a great way to provide Confirmation. It's especially useful for forms, where submitting invalid data can trigger multiple Error Messages.

Inline feedback should also be used for transactional gestures, like *swipe to delete*. If you are unfamiliar with the Android design conventions for swipe to delete, they are fundamentally different from those in iOS.

In Android, let's take Gmail as an example: swiping a message immediately archives it. This is an example of a *swipe-to-perform* gesture. Inline feedback shows that the message has been archived and offers the undo option.

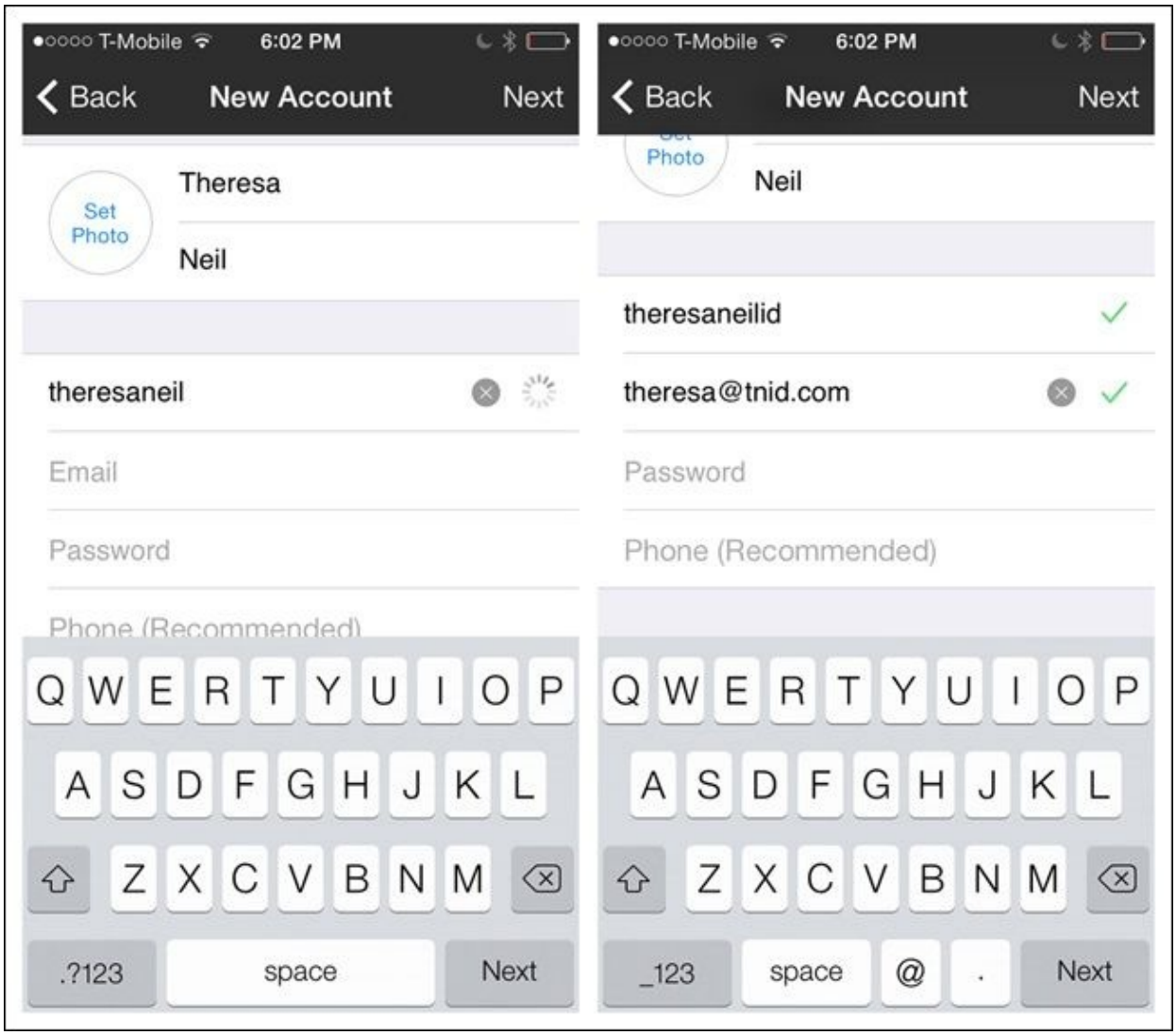



*Figure 9-8. Kik for iOS: inline feedback confirms valid data as form fields are filled*

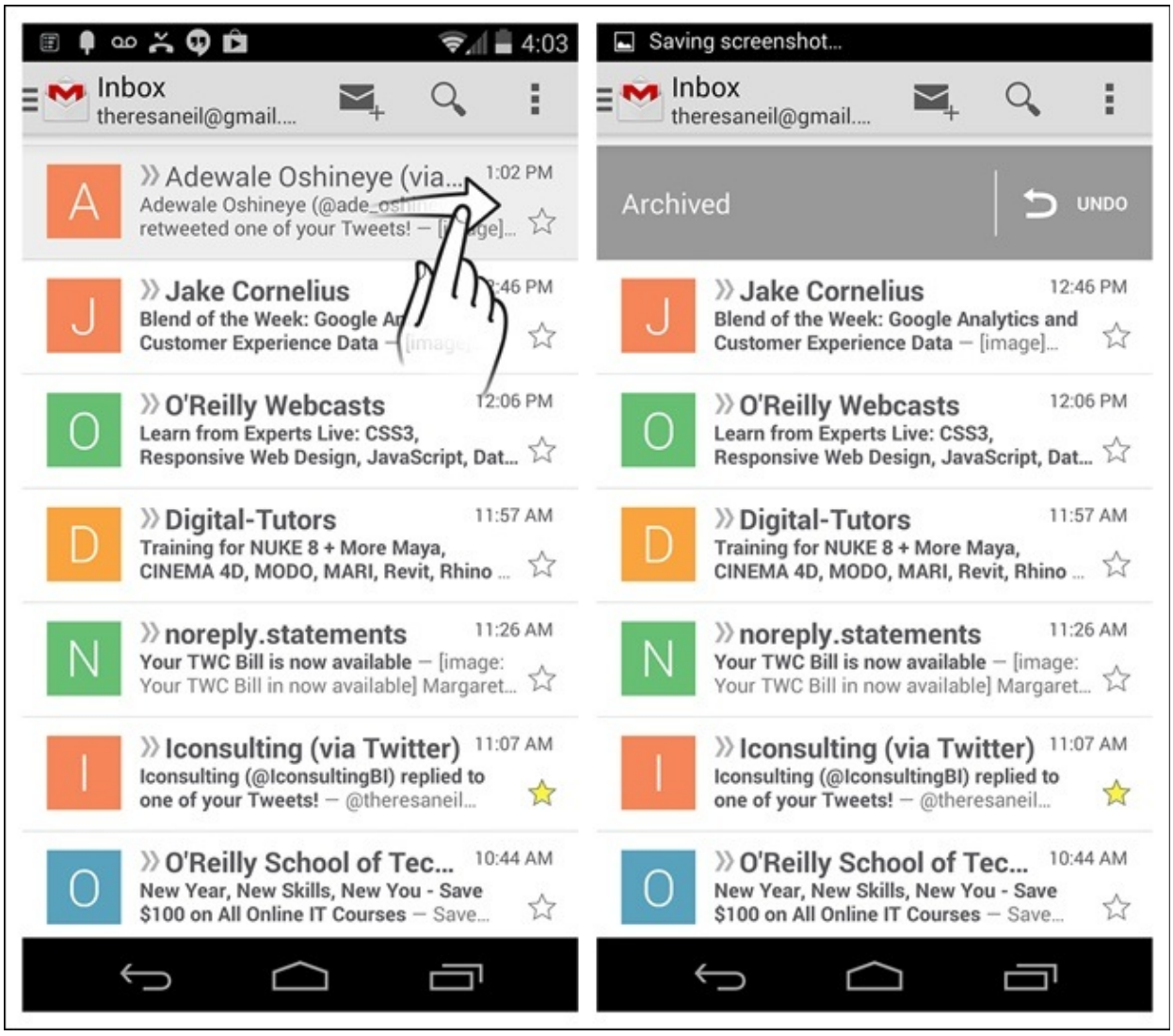



*Figure 9-9. Gmail for Android: inline Confirmation with undo on swipe-to-delete gesture*

By contrast, in Mail for iOS, swiping a message is an example of a *swipe-to-reveal* gesture. Swiping a message only reveals the possible actions; no action is performed yet. A second gesture — in this case, tap — is required to archive the message. Inline Confirmation feedback following the tap is not necessary here because the message disappears from the list with a self-healing transition.

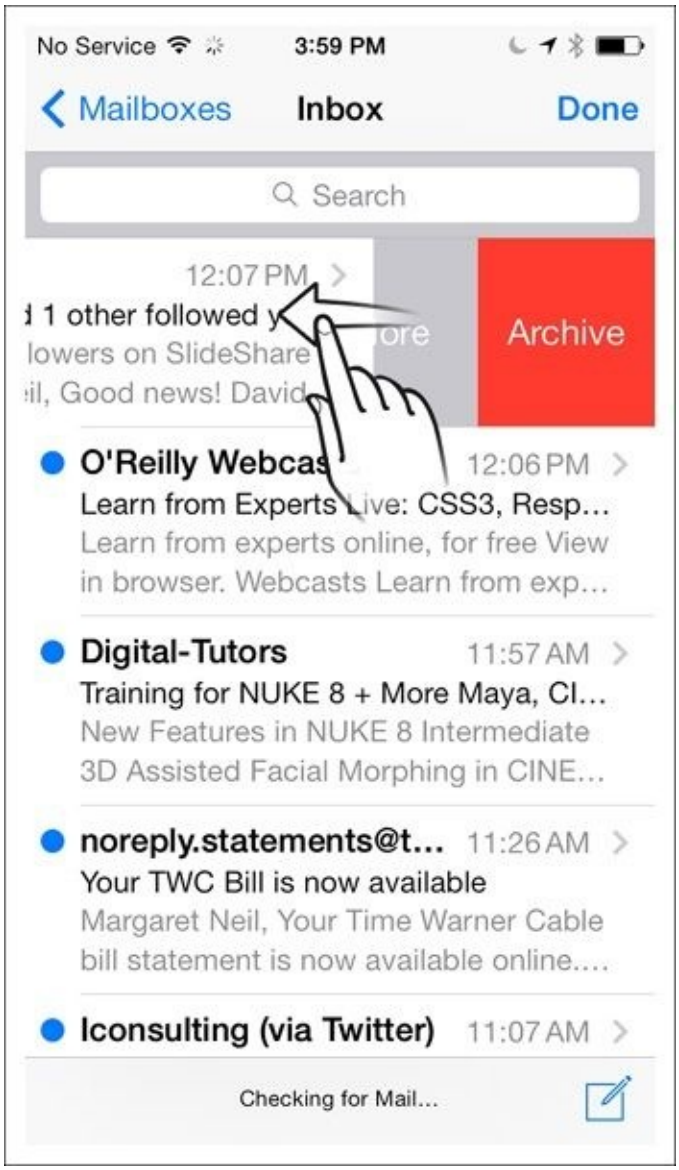



*Figure 9-10. Mail for iOS: swipe to reveal, tap to archive; inline Confirmation not required*

Another option for providing Confirmation is a transient message called a *toast*. This message appears briefly (five seconds or less) on the screen, then disappears — no additional taps from the user required. A toast confirms the action just performed and can provide an undo option, as shown in Gmail and Boxer.

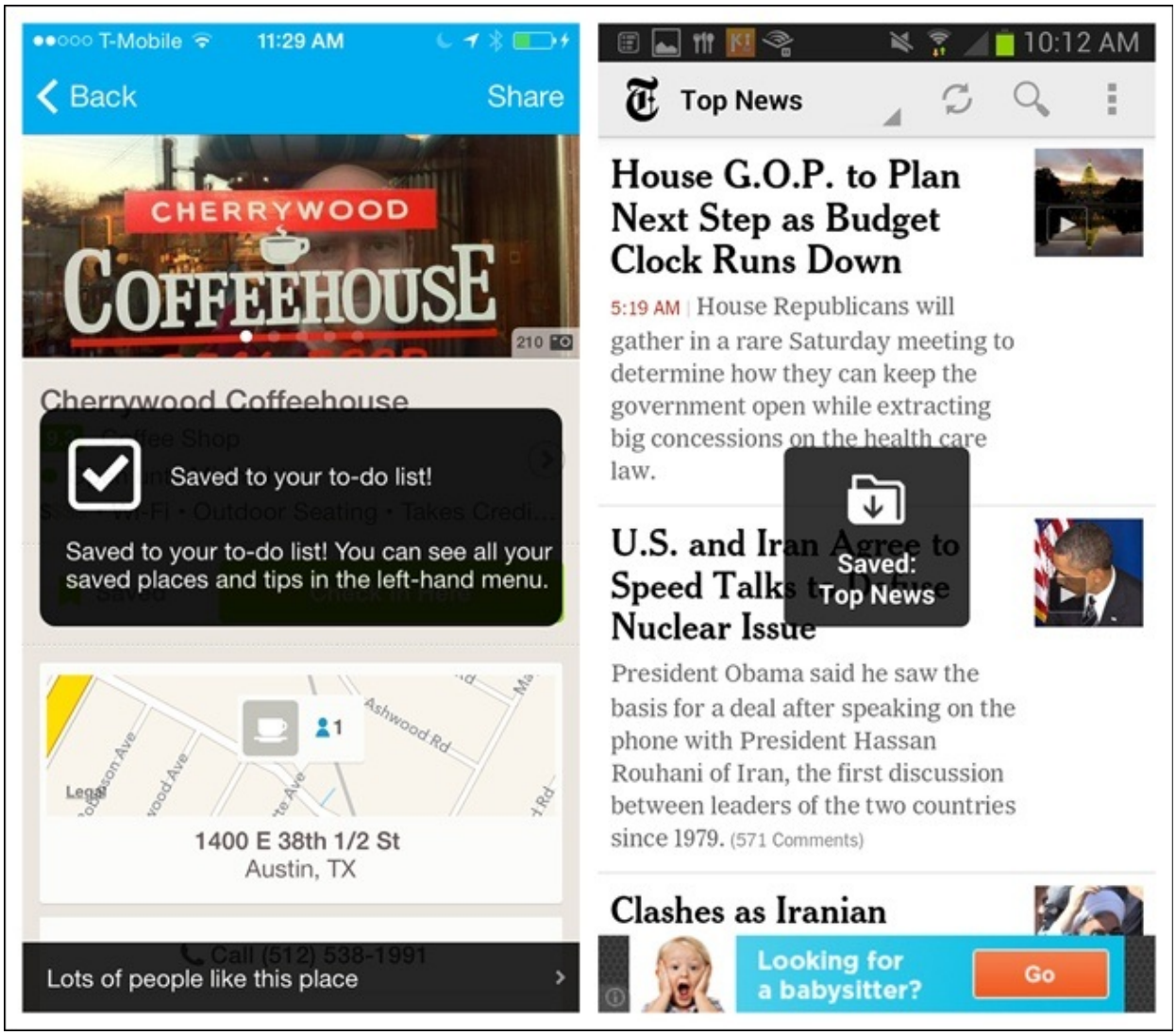



*Figure 9-11. Foursquare for iOS and NYTimes for Android: simple toasts*

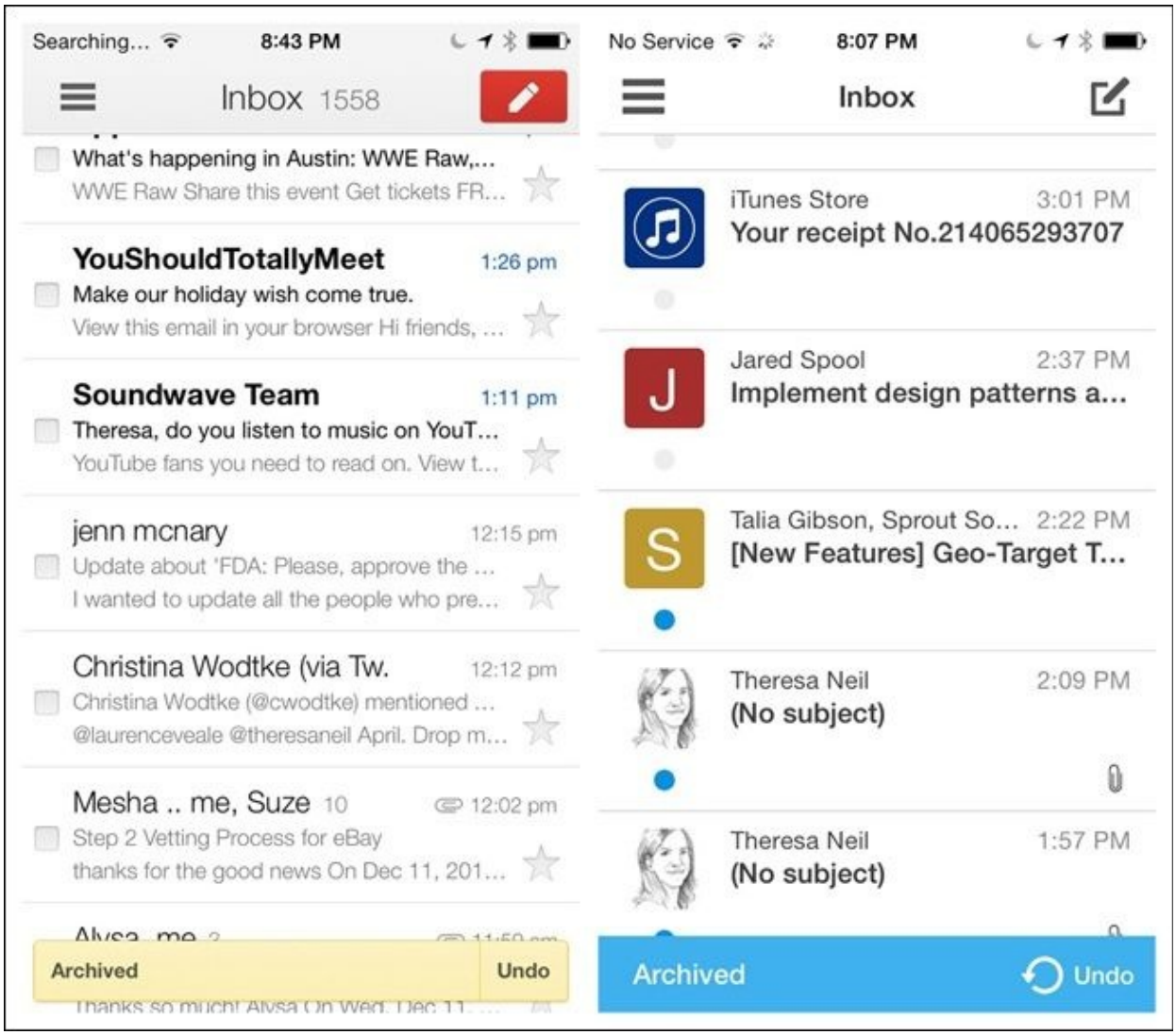



*Figure 9-12. Gmail and Boxer for iOS: transient toast message provides Confirmation and an undo option*

Multi-State Buttons, like the cloud download icon in the App Store, and progress meters, like the download progress indicator in Audible, are other unobtrusive ways to provide Confirmation without interrupting the flow. Notice that Audible shows the message “Tap to play” as soon as part of the story is downloaded, just to let you know you don’t have to wait for everything to download before you start listening.

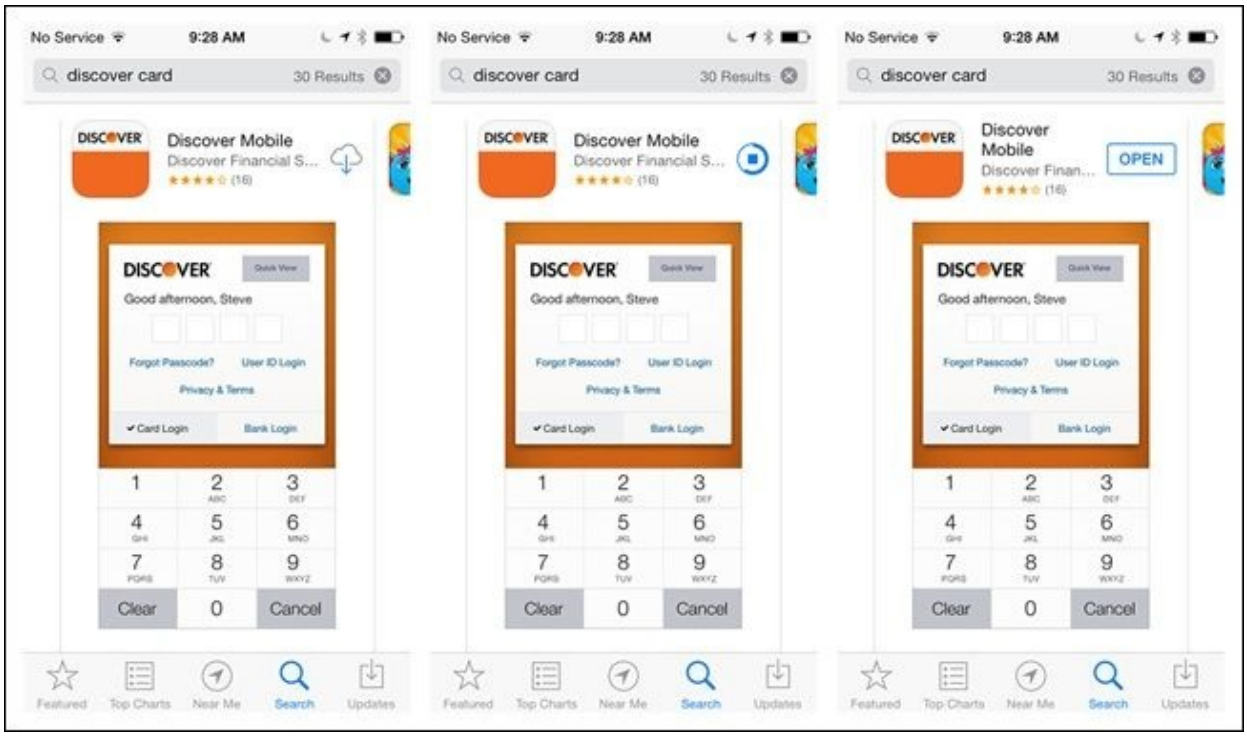



*Figure 9-13. App Store for iOS: Multi-State Button changes from Download to Progress to Open*

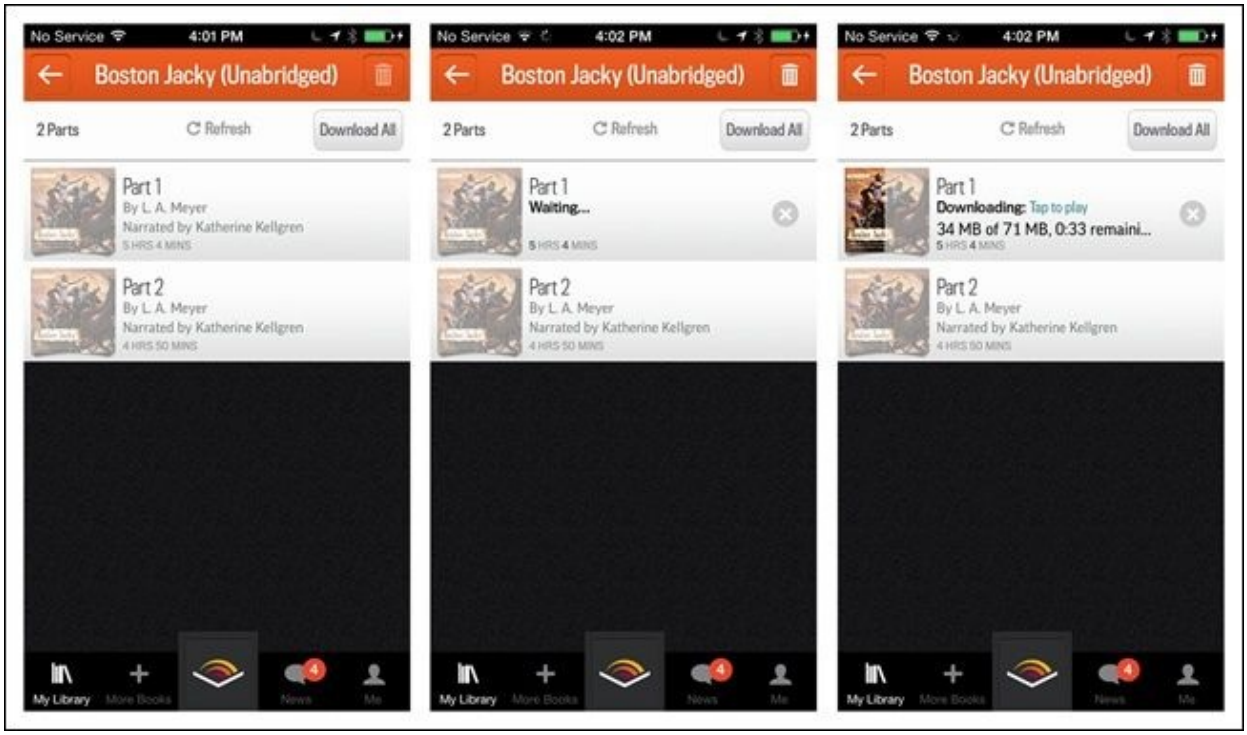



*Figure 9-14. Audible for iOS: progress meter provides feedback and a call to action*

When a transaction is completed, like an order being placed or a form being submitted, more prominent feedback is usually expected. Since these events

typically occur at the end of a flow, a full screen or dialog can be used for Confirmation, as with Check and Google Play.

**NOTE**

Provide Confirmation when actions are taken, but don’t be disruptive. Design for Confirmation elements that are part of the overall flow.

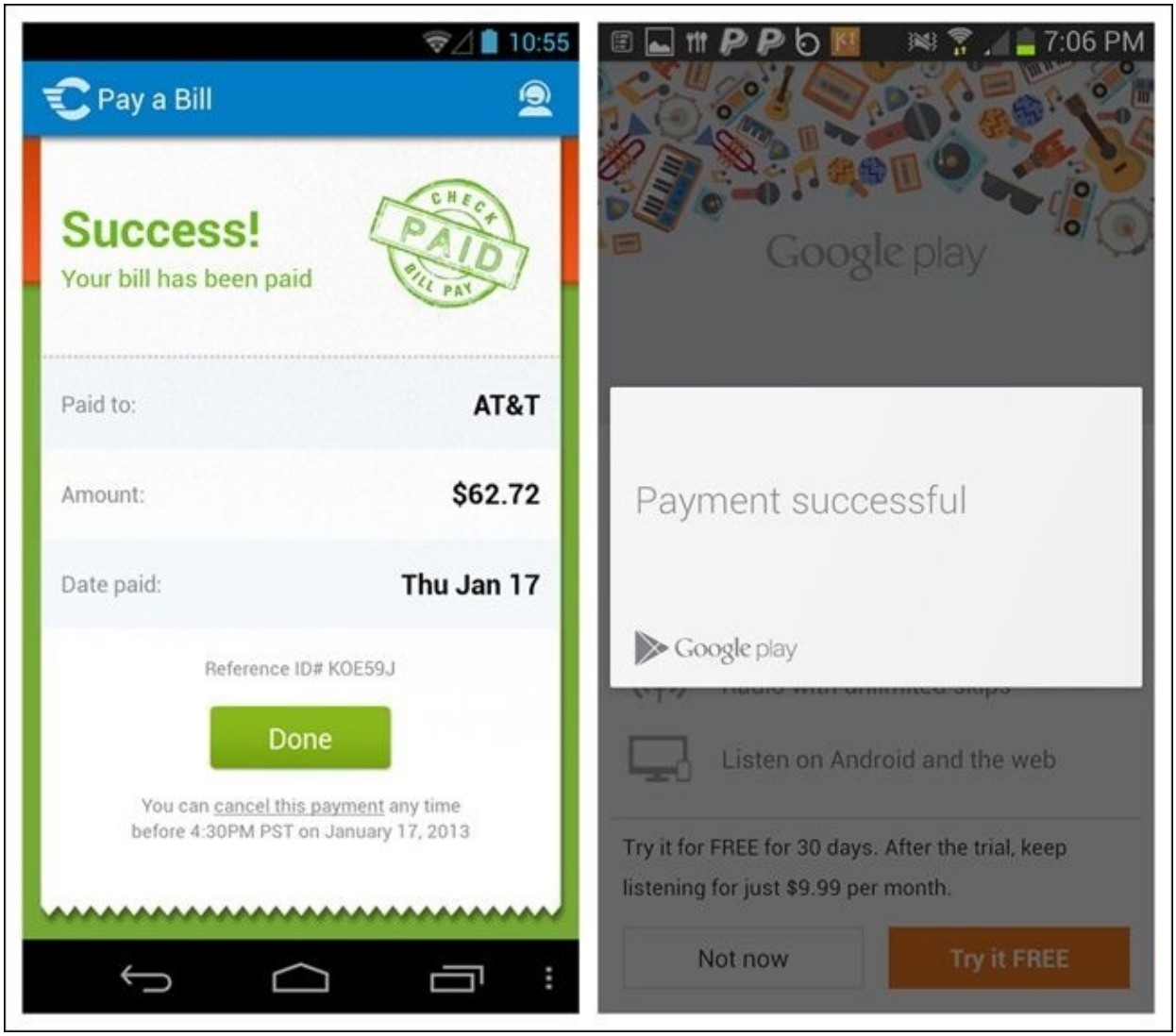



*Figure 9-15. Check and Google Play for Android: completed transactions warrant conspicuous feedback*

## System Status

In the first edition of this book, I encouraged mobile designers and developers to use the same techniques for showing System Status that were used on the Web and in desktop software. Typically, these were loading indicators.

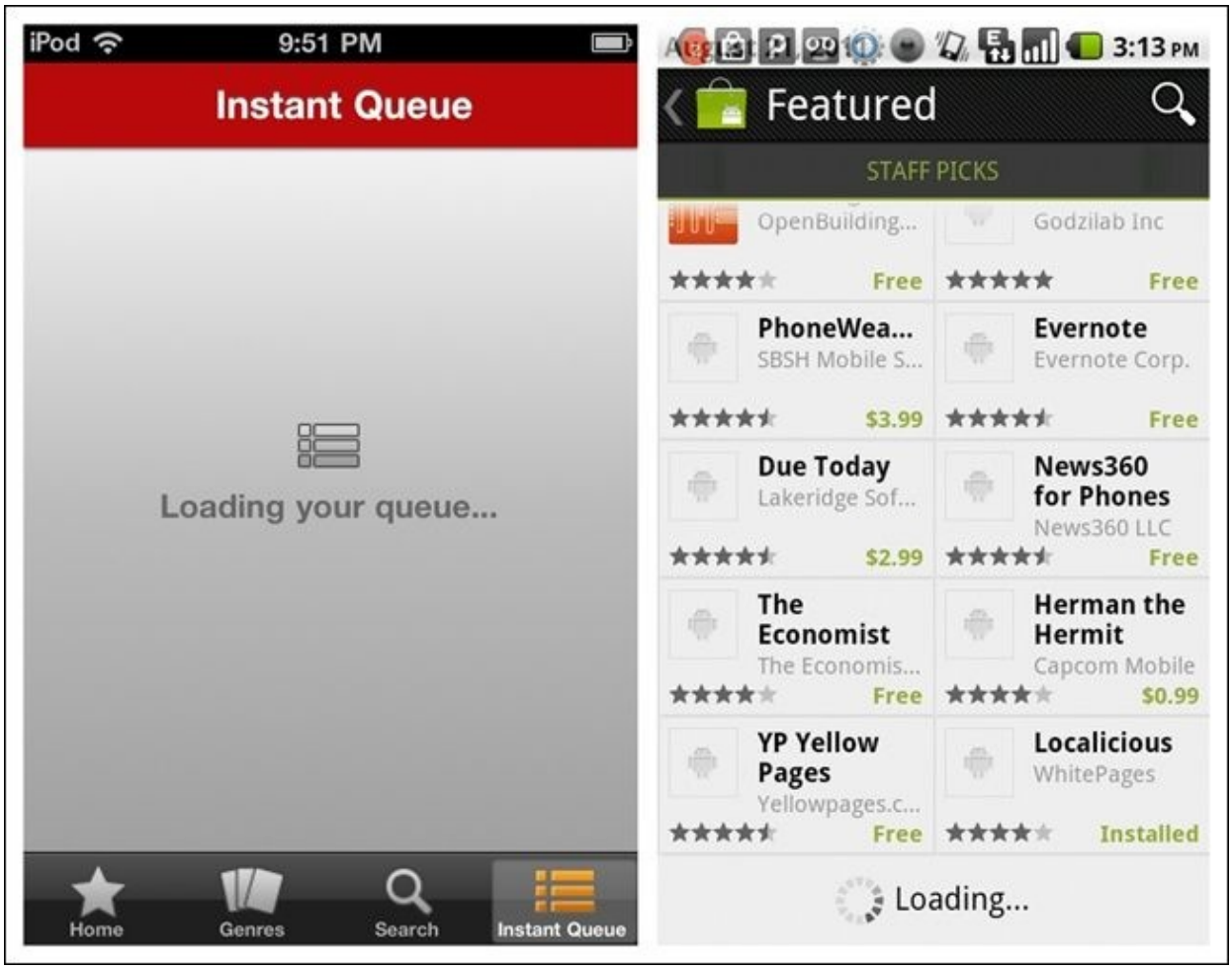



*Figure 9-16. Loading indicator examples from the first edition*

But then Luke Wroblewski, who we met in Chapter 2, posted an interesting article entitled “Avoid the Spinner” (*http://www.lukew.com/ff/entry.asp?1797=*). When he introduced the loading spinner pattern in his app Polar, he reports he started getting more user feedback like this:

> There seems to be an excessive amount of waiting around for pages to refresh and load — it doesn’t seem as quick as the previous version.

If Luke received gripes like that because of a simple spinner, I can only assume Level’s design showing a snail very slowly inching across the screen elicits similar complaints.

Luke concluded that:

> With the introduction of these progress indicators, we had made people watch the clock. As a result, time went slower and so did our app. We focused on the indicator and not the progress; that is, making it clear you are advancing toward your goal, not just waiting around.

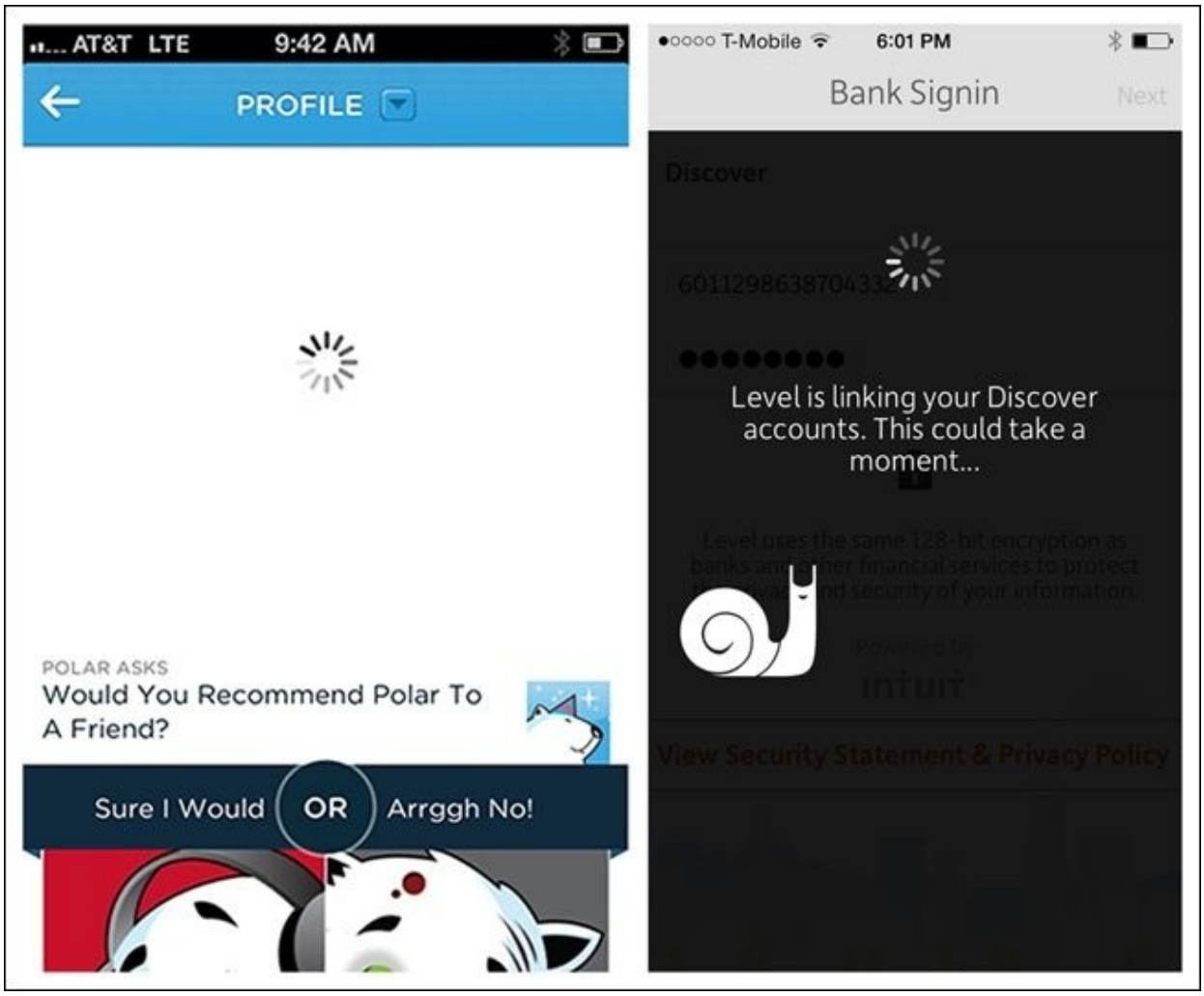



*Figure 9-17. Polar and Level for iOS: asking users to watch the clock*

There are good alternatives that focus users on the progress, not the clock. Google loads pages from the side, making it feel like content is loading immediately even when it isn't. The transition is, in effect, the loading indicator.

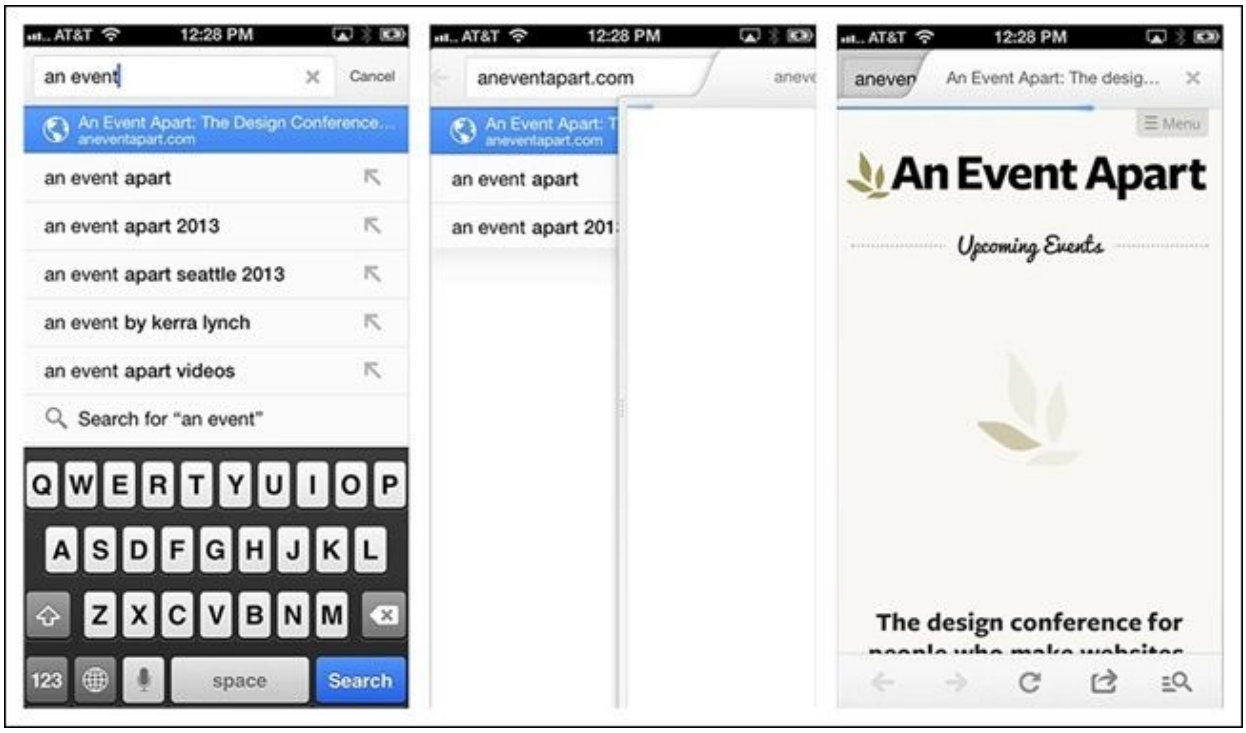



*Figure 9-18. Google for iOS: side-loading page transitions serve as loading indicators*

Polar now uses *skeleton screens* — essentially, blank versions of pages into which information is gradually loaded.

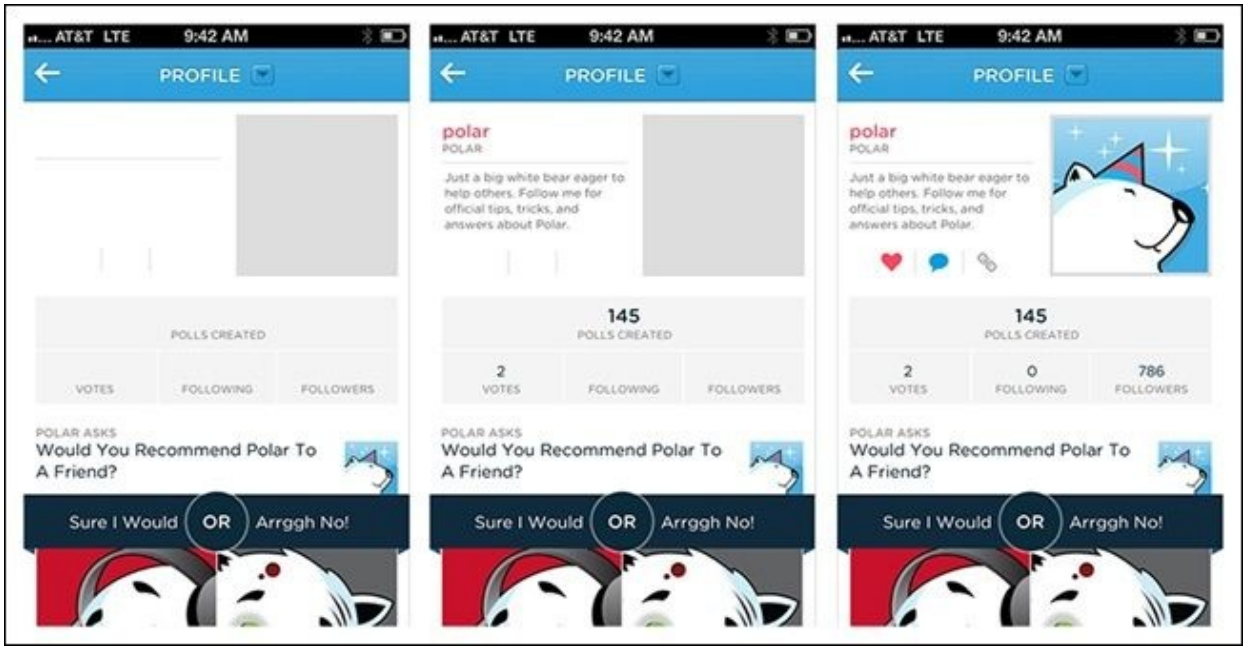



*Figure 9-19. Polar for iOS: gradually loading skeleton screen as progress feedback*

Some apps have a significant wait time for initial configuration or signup.

Examples from SigFig and Smartr show two different approaches to keeping users informed of the app's progress.

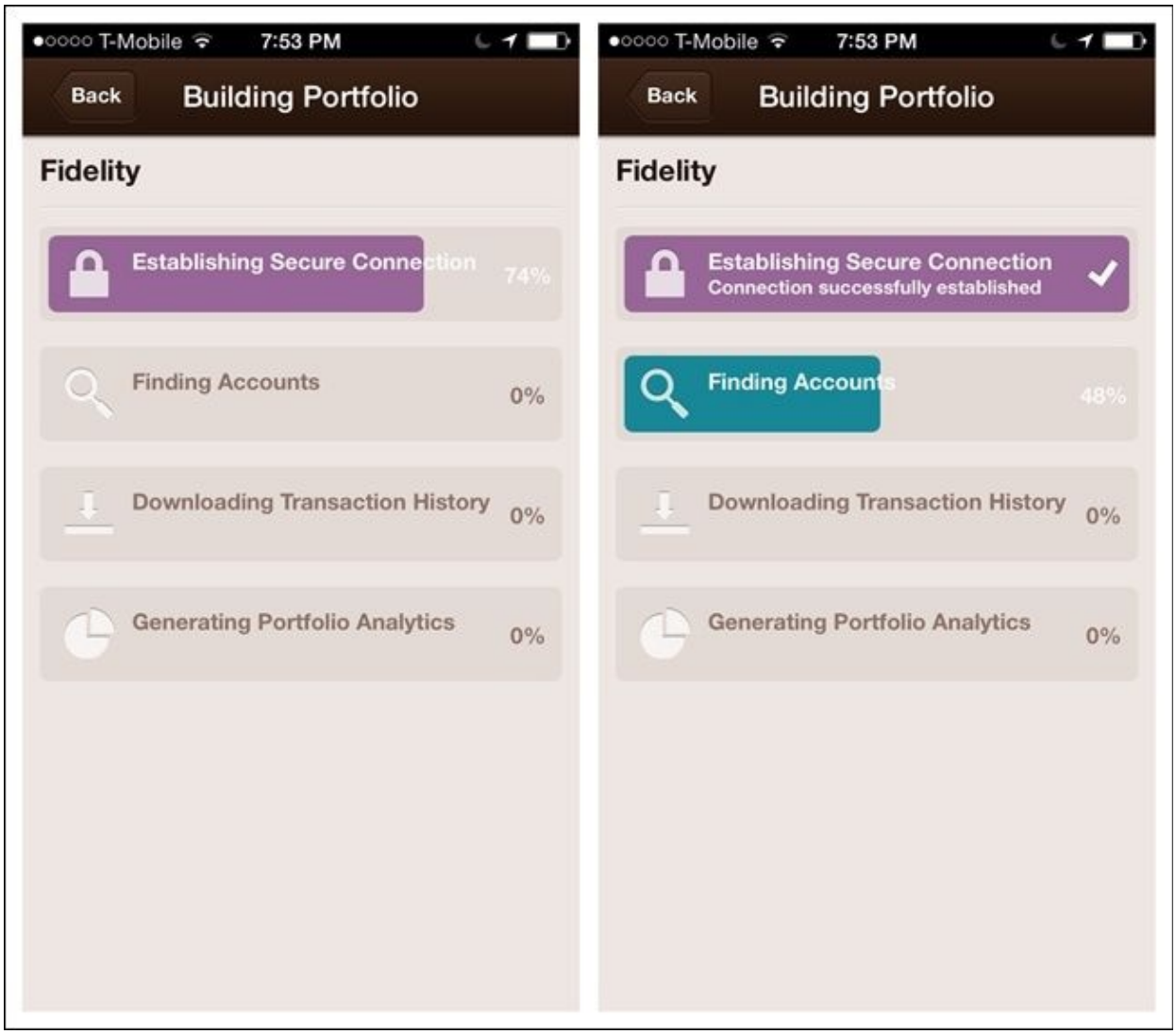



*Figure 9-20. SigFig for iOS: live feedback shows completion percentage of each task as portfolio builds*

For lengthy processes, offer a cancel option, like Story Board does for generating a movie.

**NOTE**

Do provide feedback for System Status, but focus the user on the progress, rather than the wait. Offer a cancel option for potentially lengthy operations.

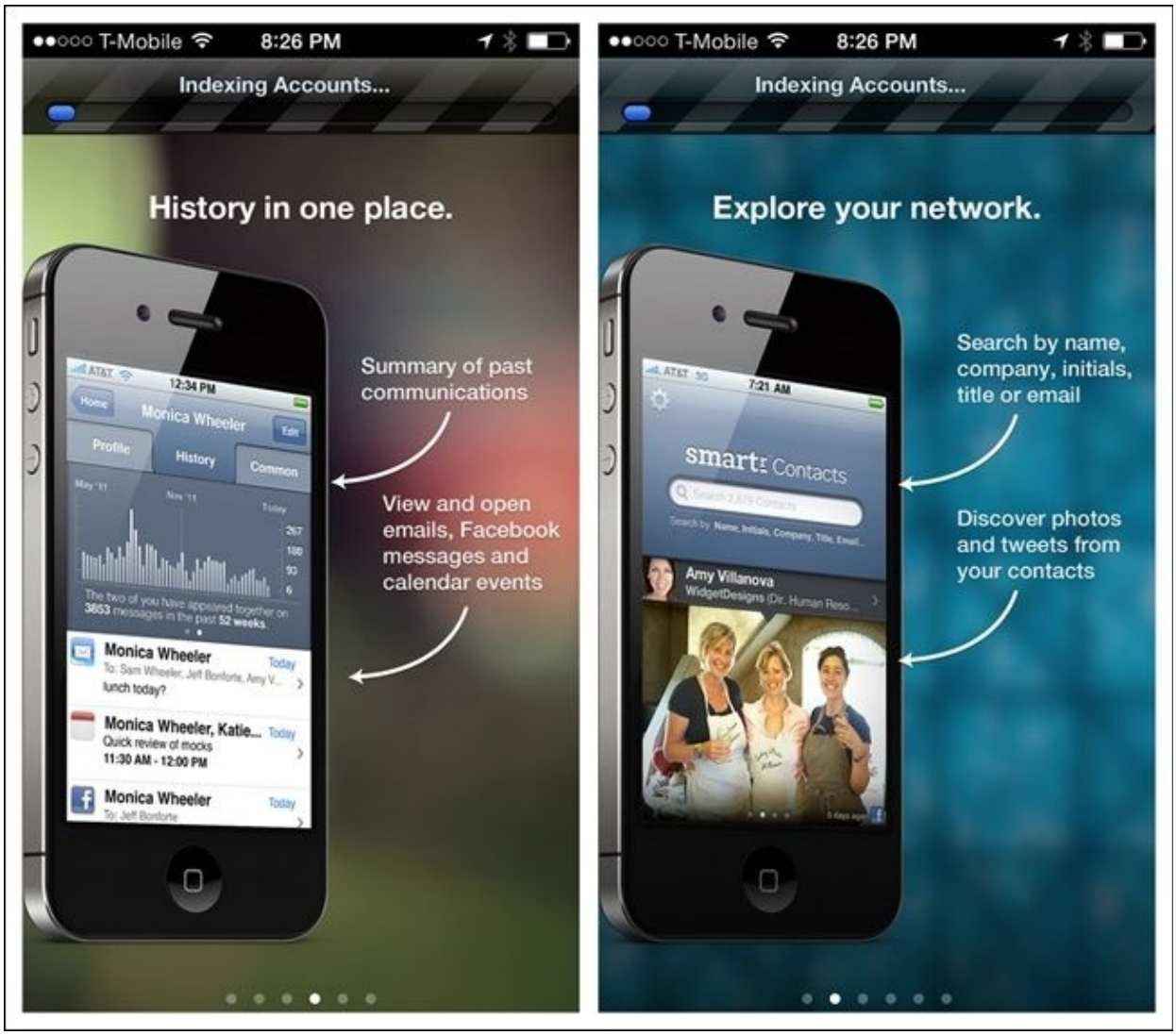



*Figure 9-21. Smartr for iOS: time spent indexing contacts (see progress bar at top) is used to introduce product features*

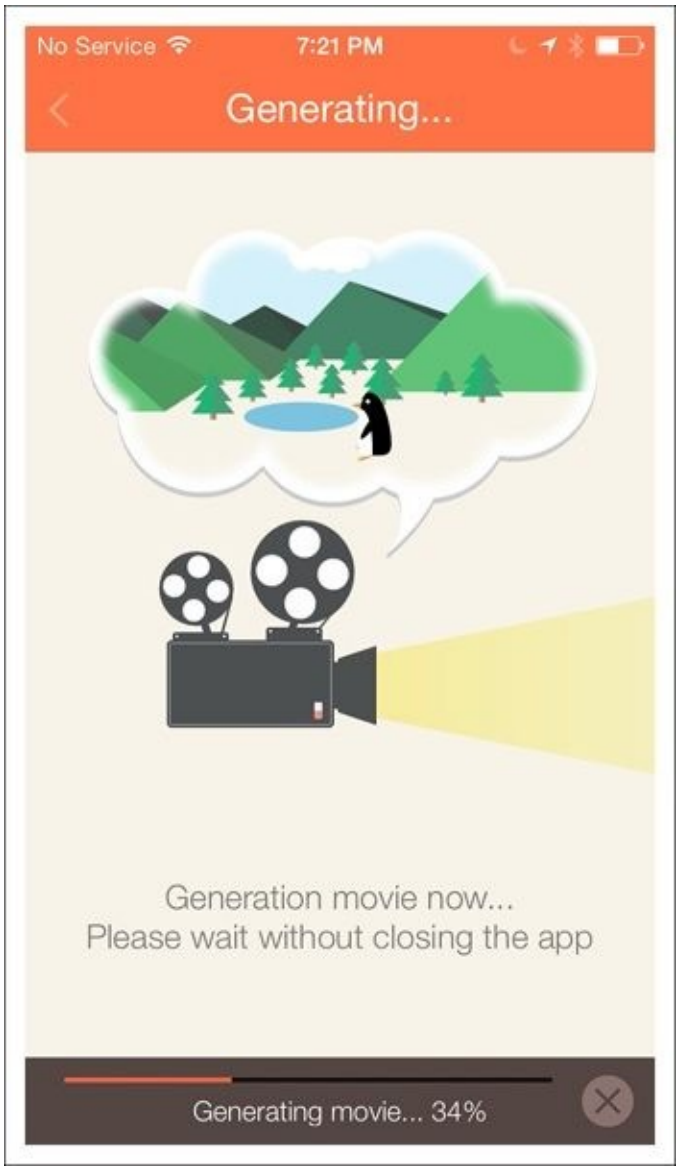



*Figure 9-22. Story Board for iOS: long process can be canceled*

# Affordance

Affordance is the quality of an object that lets a user know that the object can be used to perform an action. Drawer handles and doorknobs are good real-life examples. Common examples of affordance in software include drag handles, page peels, and 3D controls like buttons and sliders.

Since mobile devices make use of single-and multitouch gestures, the list of possible affordances is long and the range of techniques for implementing them is broad. We'll look at some common affordances for Tap, Swipe/Flick, and Drag.

## Tap

Visual design techniques like beveling and shadows can make elements appear tappable. The rising popularity of flat design, however, has made these techniques less accessible to designers. In fact, some people think flat design has decreased the usability of many applications.

Jakob Nielsen, a well-known UX consultant and author, ran a usability test on Windows 8 (*http://bit.ly/Otx5TA*). He found that the operating system's flat design makes it hard for users to know whether things are clickable. In the example shown, many of Nielsen's test subjects mistook "Change PC settings" as a static label for the icons above it, but it's really a clickable command.

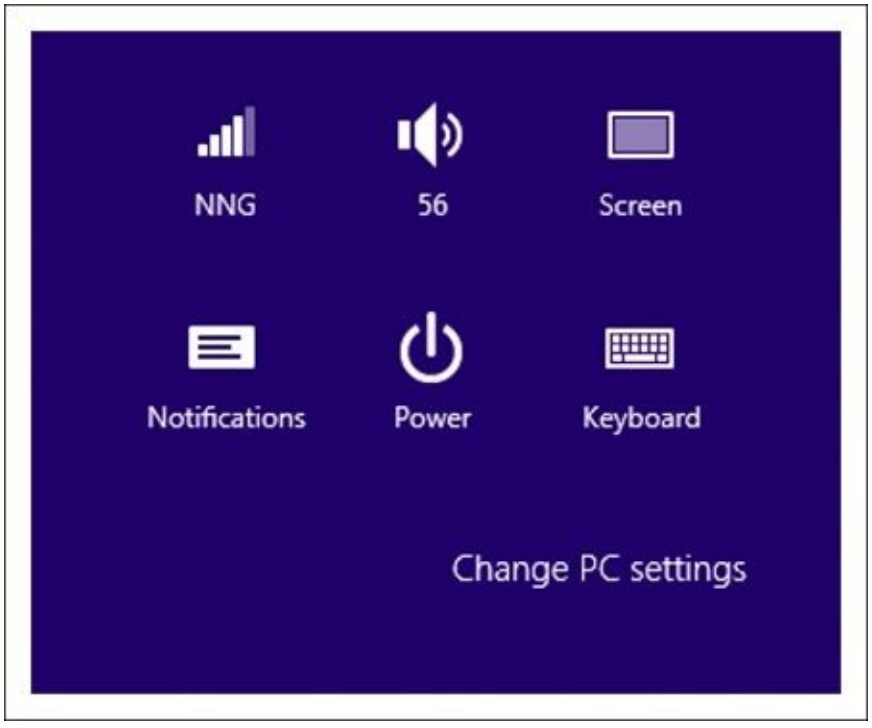


*Figure 9-23. Microsoft Windows 8: flat design may decrease usability*

This is not to say that flat designs can't provide affordance, just that it requires more design skill to get it right. Luke Wroblewski wrote another great article about his experience updating the visual design of Polar for iOS 7 (*http://www.lukew.com/ff/entry.asp?1800*), where one of the major takeaways was this:

> Removing texture and depth forced the rest of the visual design to work harder to create meaningful distinctions between the various elements on screen. I think this is a key reason why [flat design] is harder. It forces you to simplify in order to provide the same clear visual communication using less visual relationships.

**NOTE**

Poor implementation of flat design can harm an application's overall usability. Be prepared for more iterations during the design phase to ensure that actionable elements are not lost in a sea of flatness.

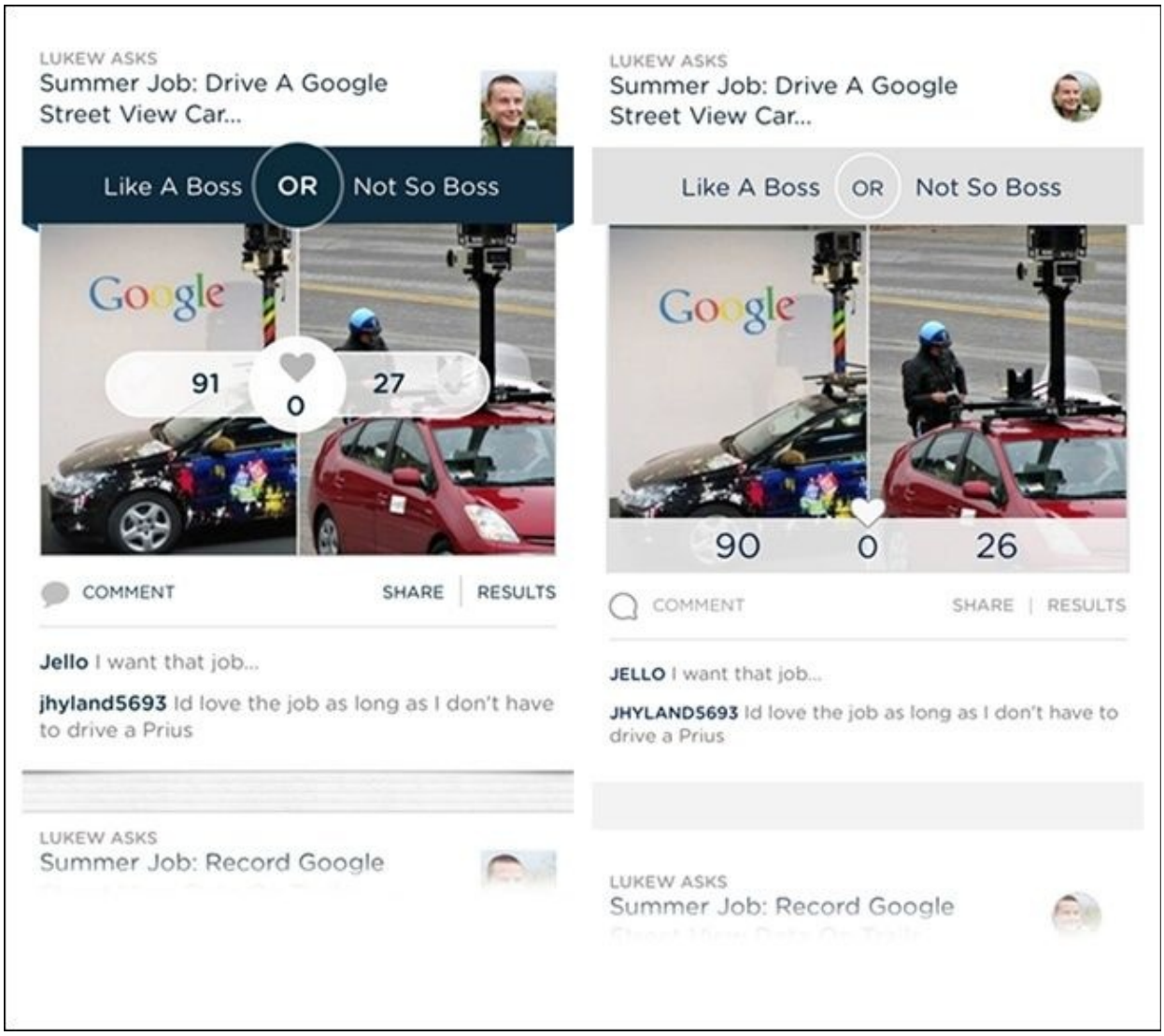



*Figure 9-24. Early design iterations for Polar iOS7*

## Swipe/Flick

There are many ways to provide affordance that a swipe action is available. The iOS page indicator control — aka “those little dots at the bottom of the screen” — has been widely adopted by web and mobile apps alike.

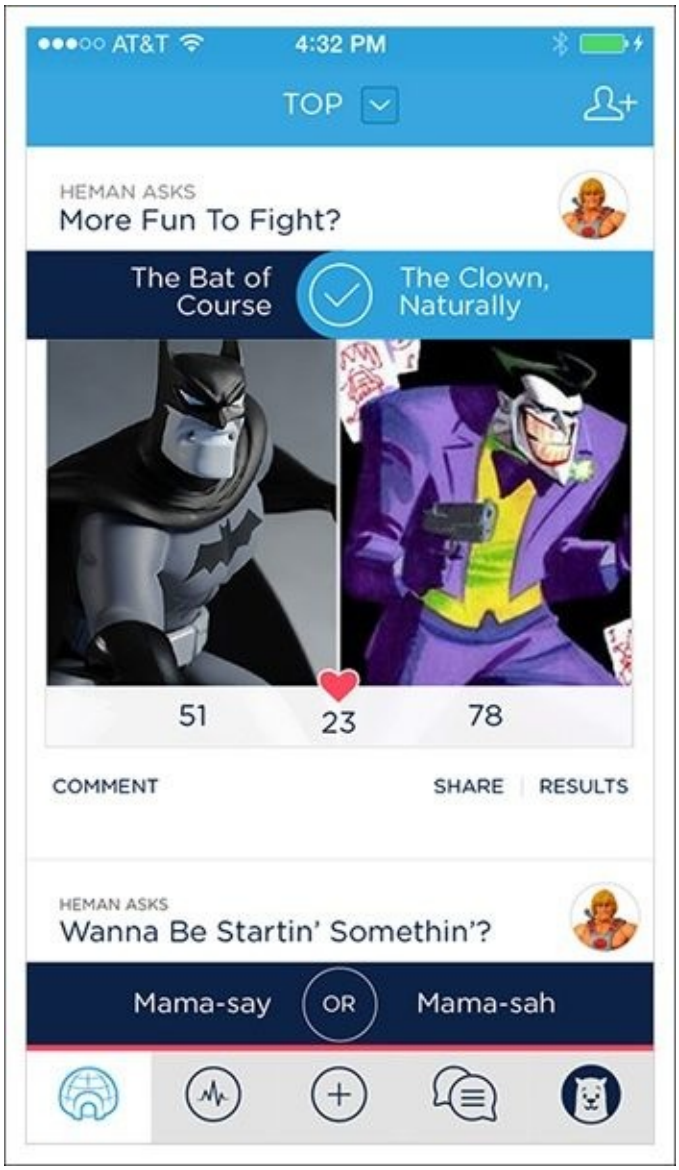



*Figure 9-25. Polar for iOS 7: a later iteration makes different screen elements more distinct*

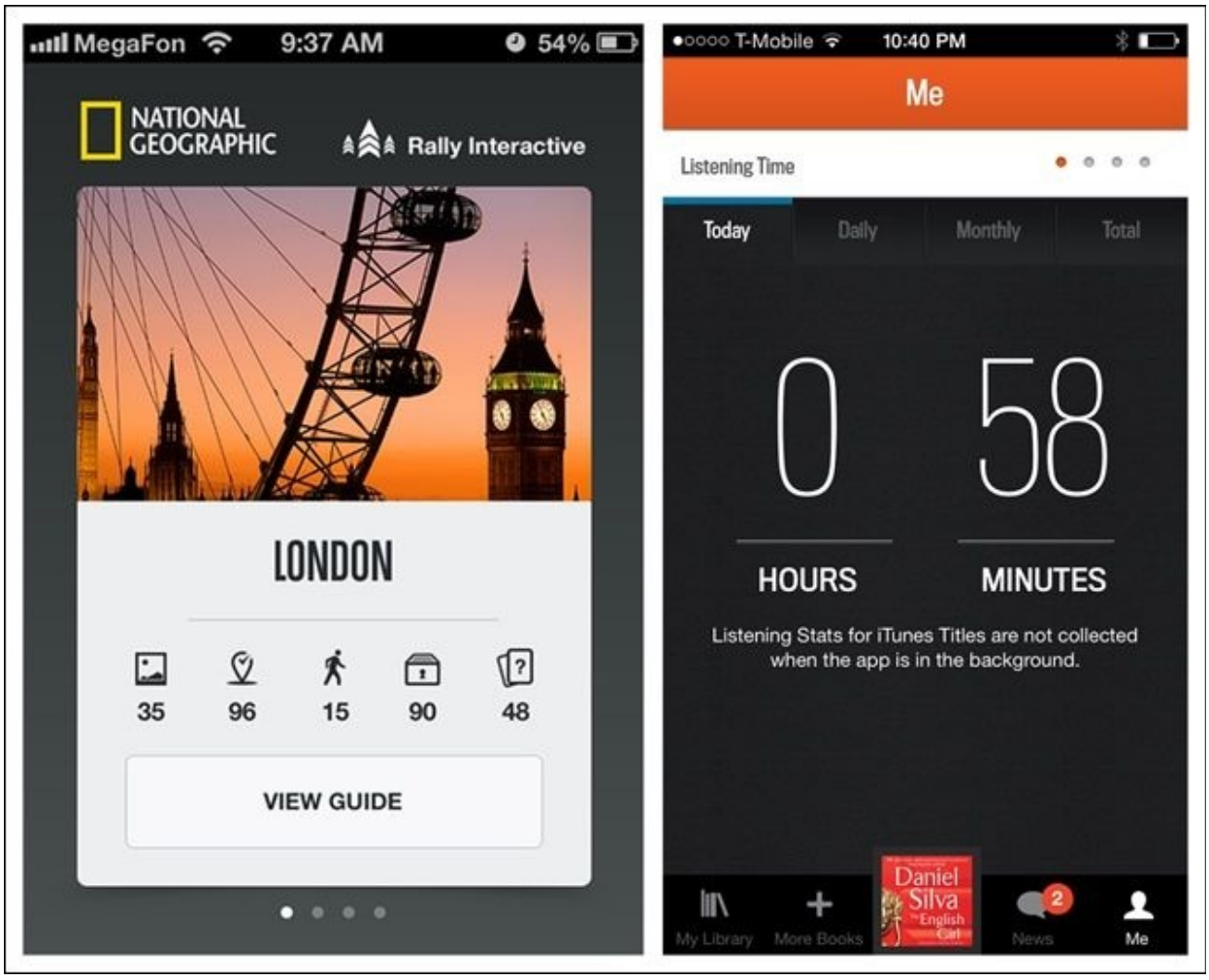



*Figure 9-26. City Guides by National Geographic and Audible for iOS: page indicators at the bottom and top, respectively*

Another option is to show content beyond the borders of the screen to imply that there is more to see off to the sides or below.

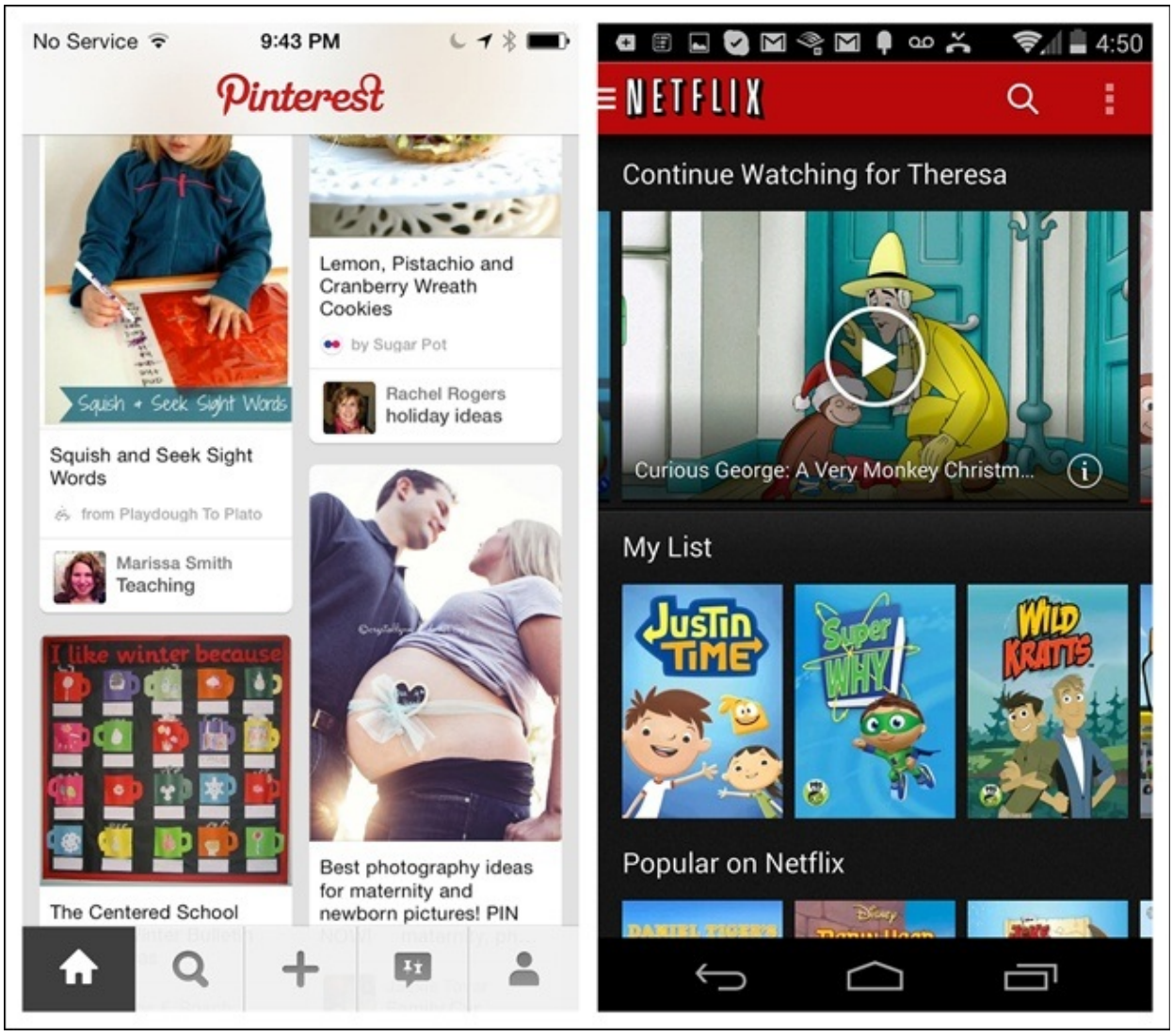



*Figure 9-27. Pinterest for iOS and Netflix for Android: screen overflow hints at how to get at offscreen content*

Scrollbars are also an option, although they are less common in mobile apps. The scrollbar may also act as a touch target for quickly jumping to a certain spot. Skype for iOS uses an alphabet scrollbar that lets you jump to a specific section; Skype for Android uses a retracting scrollbar with a drag handle.

Other design metaphors that provide visual affordance for swiping include the Coverflow/Carousel, Rolodex, and Photo Stack.

Potluck uses stacked Cards to encourage swiping. Swipe left to skip the story; swipe right to keep it. I didn't notice this upon first use, though, and wondered how I could get a story back if I swiped it off the screen. Then I happened to try dragging the story instead of swiping. Dragging revealed a tip that clued me in as to how swiping worked, but I wonder if most people would try dragging first

or swiping first.

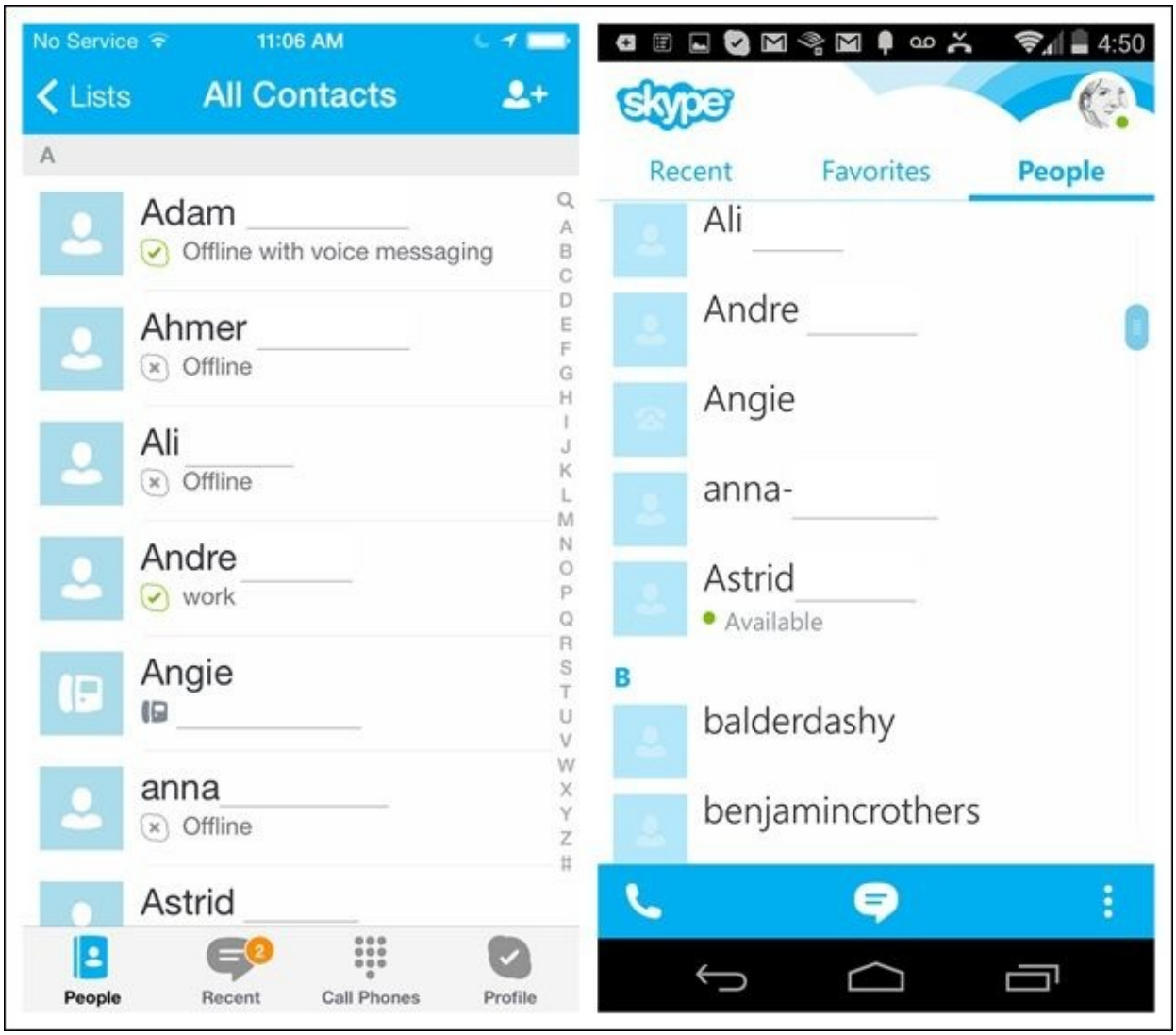



*Figure 9-28. Skype for iOS and Android: different scrollbar designs*

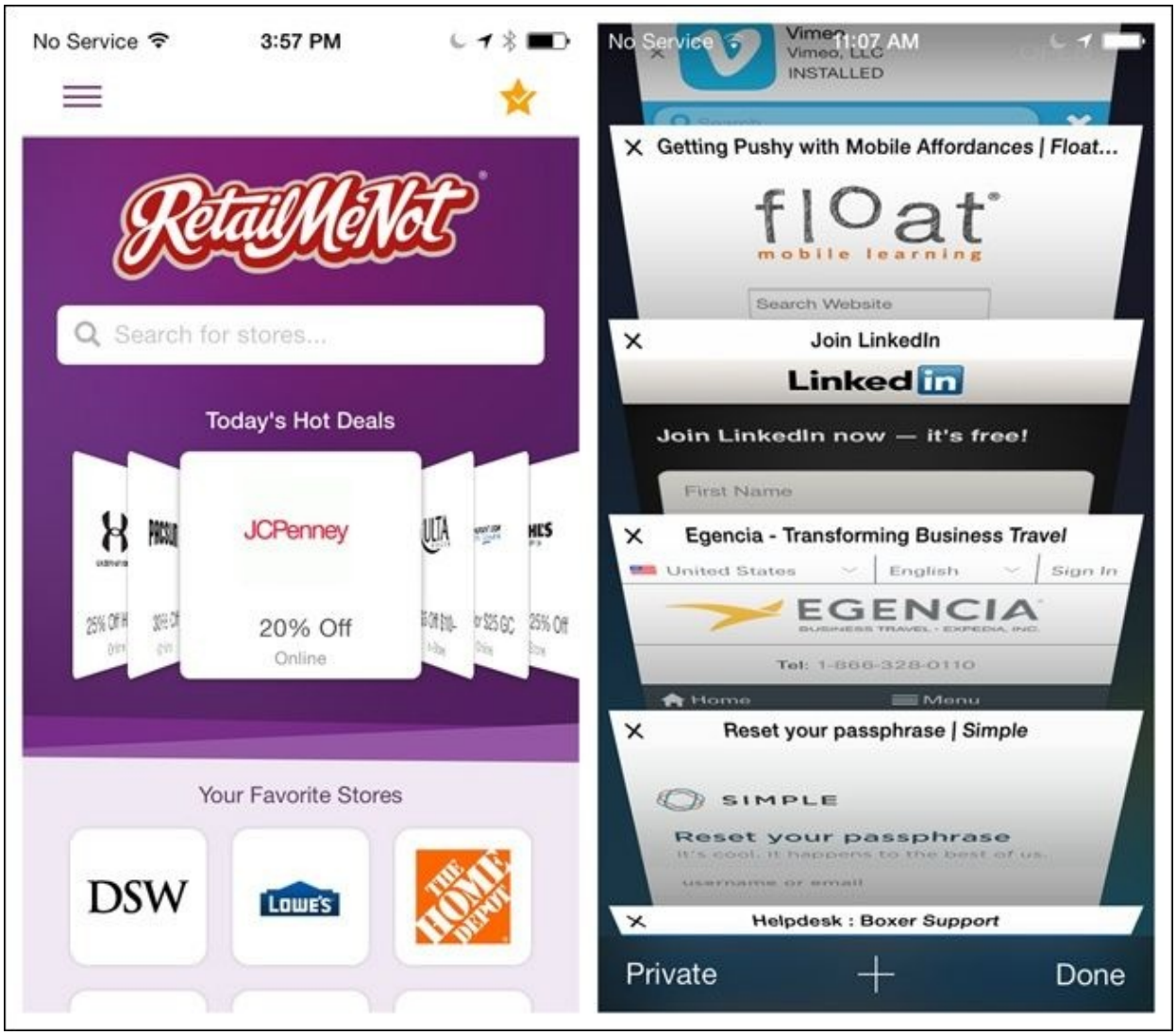



*Figure 9-29. RetailMeNot and Safari for iOS: Coverflow is affordance for horizontal swipe and Rolodex for vertical swipe, respectively*

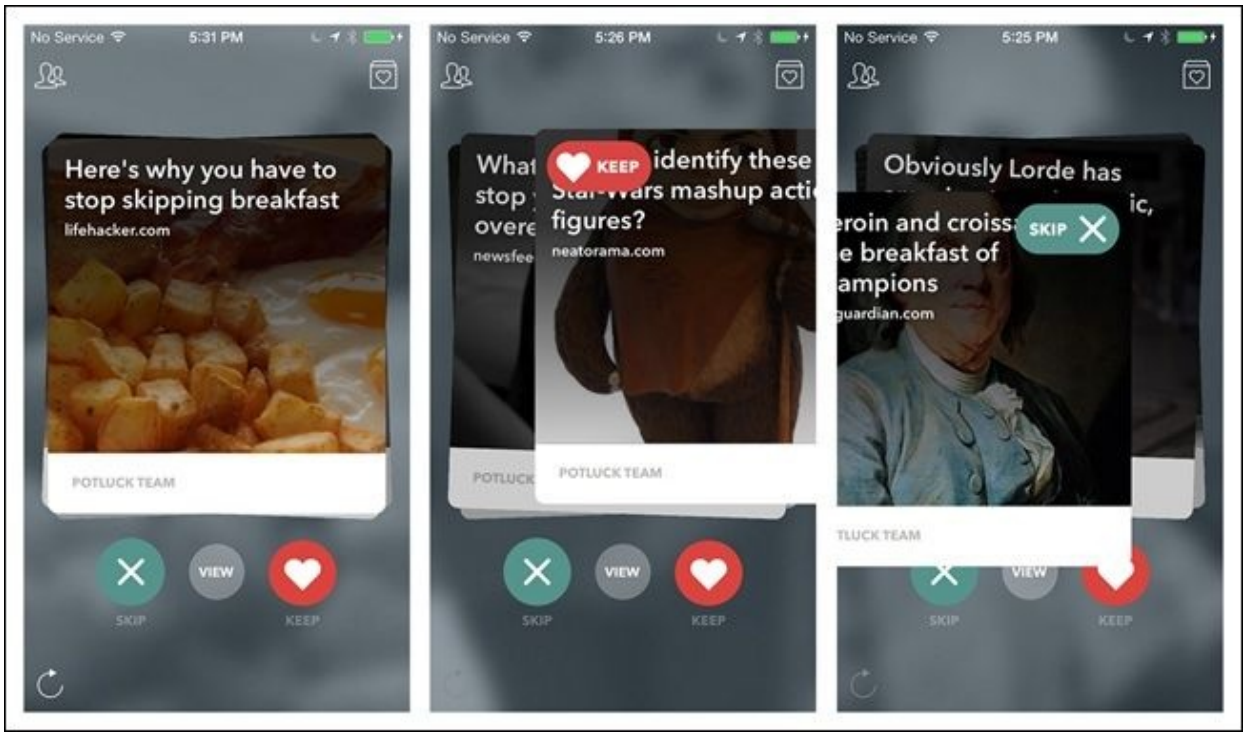



*Figure 9-30. Potluck for iOS: affordance to swipe is good, but an indication of where the story is going could be better (http://www.youtube.com/watch?v=pcfNFuvvdrA)*

Animation can also entice users to try a swipe gesture. Allthecooks for Android "bumps" the side menu open to prompt that a horizontal swipe gesture reveals the full menu. Flipboard flips the bottom edge of the screen up and down to invite the vertical page-swipe gesture the application is known for.

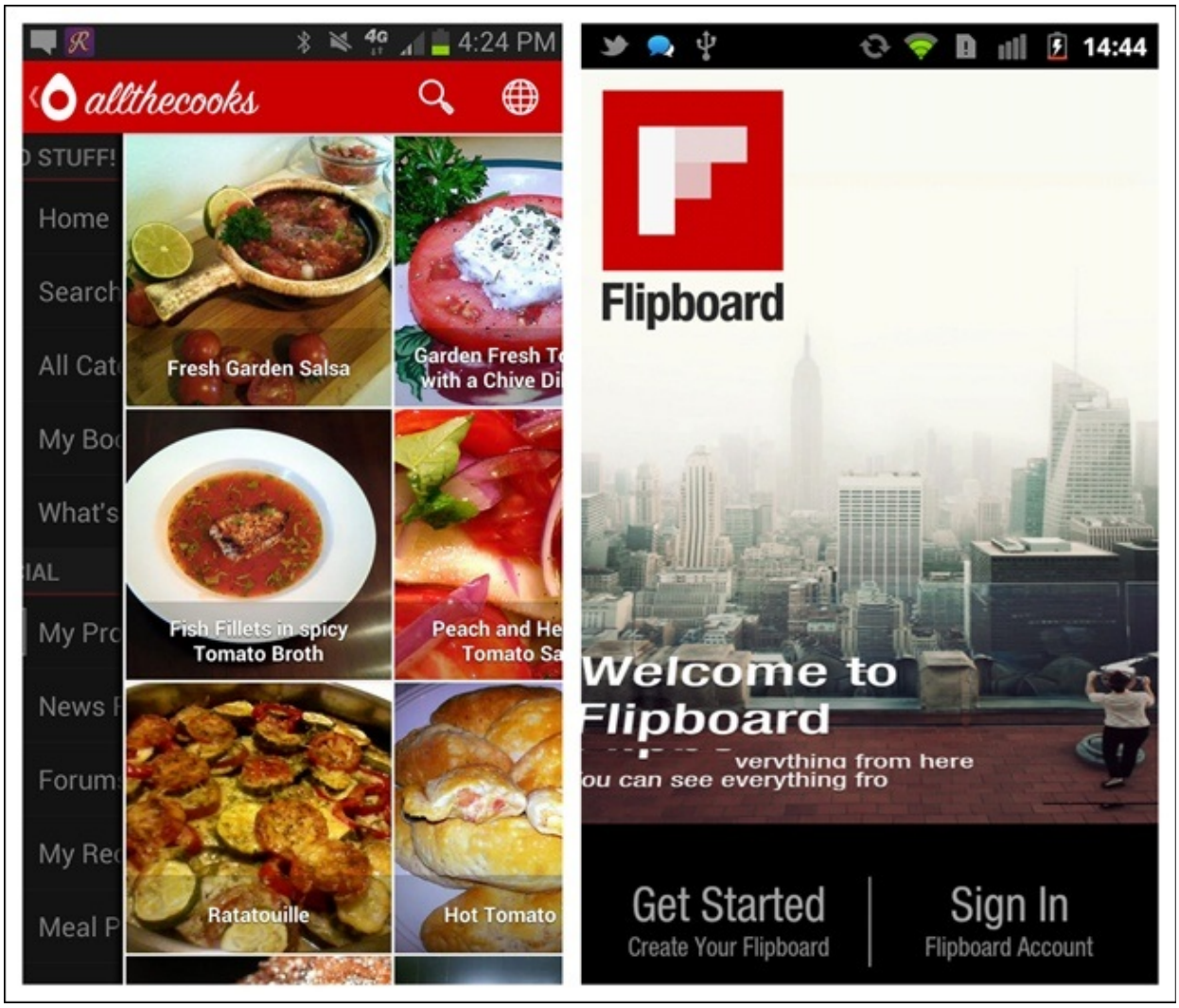



*Figure 9-31. Allthecooks for Android (http://youtu.be/iLlCs-gyNaE) and Flipboard for iOS (http://www.youtube.com/watch?v=rQASZtuvBCs): animations as affordance*

There are additional examples in Chapter 7 from Mailbox and Fantastical that illustrate how to create an interactive tutorial that introduces gestures at the optimal time for users to learn them.

Text has proven to be very ineffective at enticing users to swipe, so skip that option and try other techniques. An early version of Silvercar had a text prompt to swipe to the side. A later version replaced it with a vertical swipe. Content that extended off the bottom of the screen provided the affordance that swipe was possible.

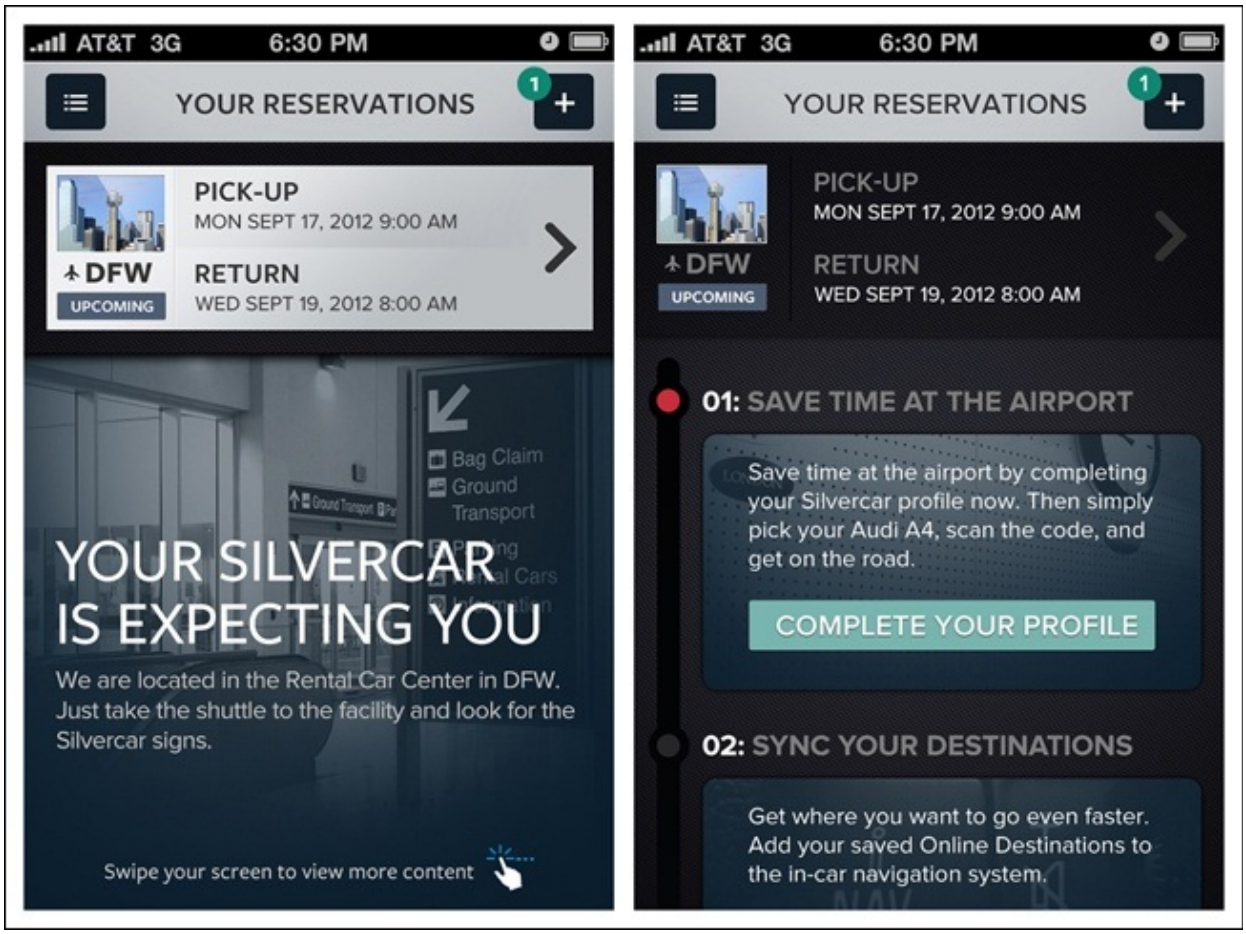



*Figure 9-32. Silvercar for iOS: early design with horizontal swipe and text prompt; redesigned with vertical swipe*

## Drag

The Drag action in mobile is usually accessed via a mode, like the edit mode, or on long press. In iOS, you can long-press any app icon to access the edit mode and drag apps to reorganize them. Apps can also be deleted in this mode.


*Figure 9-33. iOS 7: long press allows app icons to be dragged or deleted*

In Android, long-pressing an app icon reveals “drop zones.” In this state, apps can be moved to a new screen or dragged to the Uninstall drop zone. Like the trash can icon on a desktop, this concept is outdated. But the real anti-pattern in this design is that to see information about the app, the user must drag the app icon to the App Info drop zone. Talk about inefficient and unintuitive.

A better solution would be a long-press menu with Move, Uninstall, and App Info as top-level options. Tapping Move would activate the edit mode and reveal the page drop zones, like a long press in the current design does.

Neither of these examples of Drag has any affordance. They have to be discovered. Users are willing to play with the OS and learn the tricks, but don’t assume they’ll give your app as much leeway.

Within applications, the most common affordance for dragging is the handle. Handles are usually exposed after explicitly entering an edit mode. They work well for reordering list items, as shown in Wunderlist and Lemon Wallet.

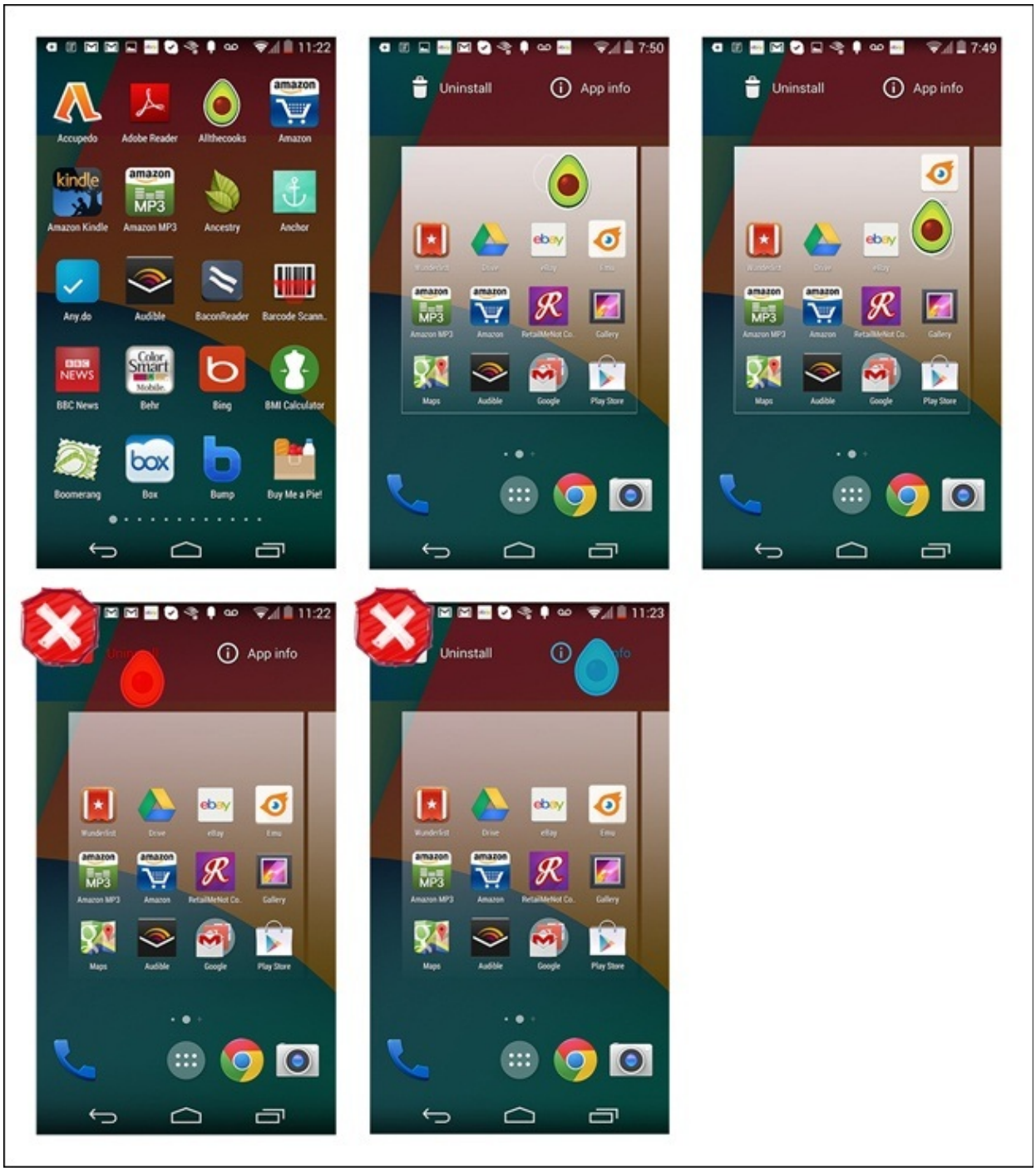



*Figure 9-34. Android OS: Drag to move a selected app to a new screen, or to Uninstall or App Info "drop zones"*

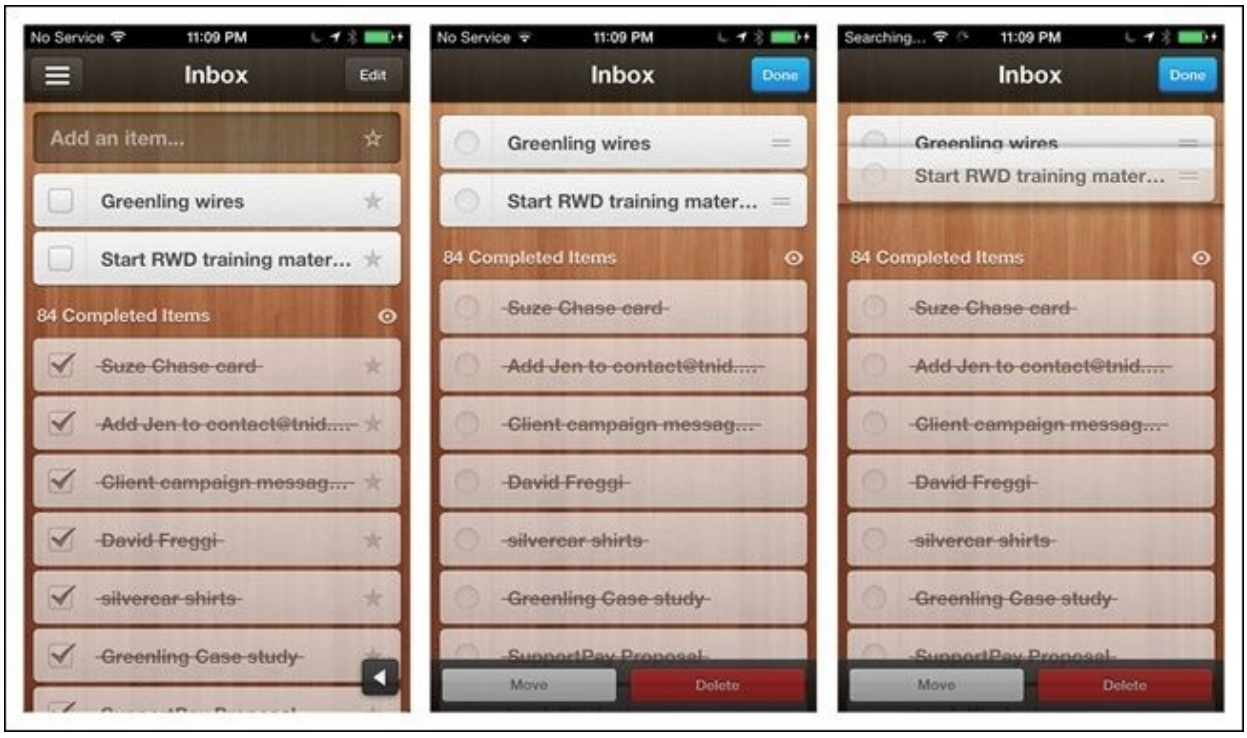



*Figure 9-35. Wunderlist for iOS: task list edit mode reveals drag handles on right*

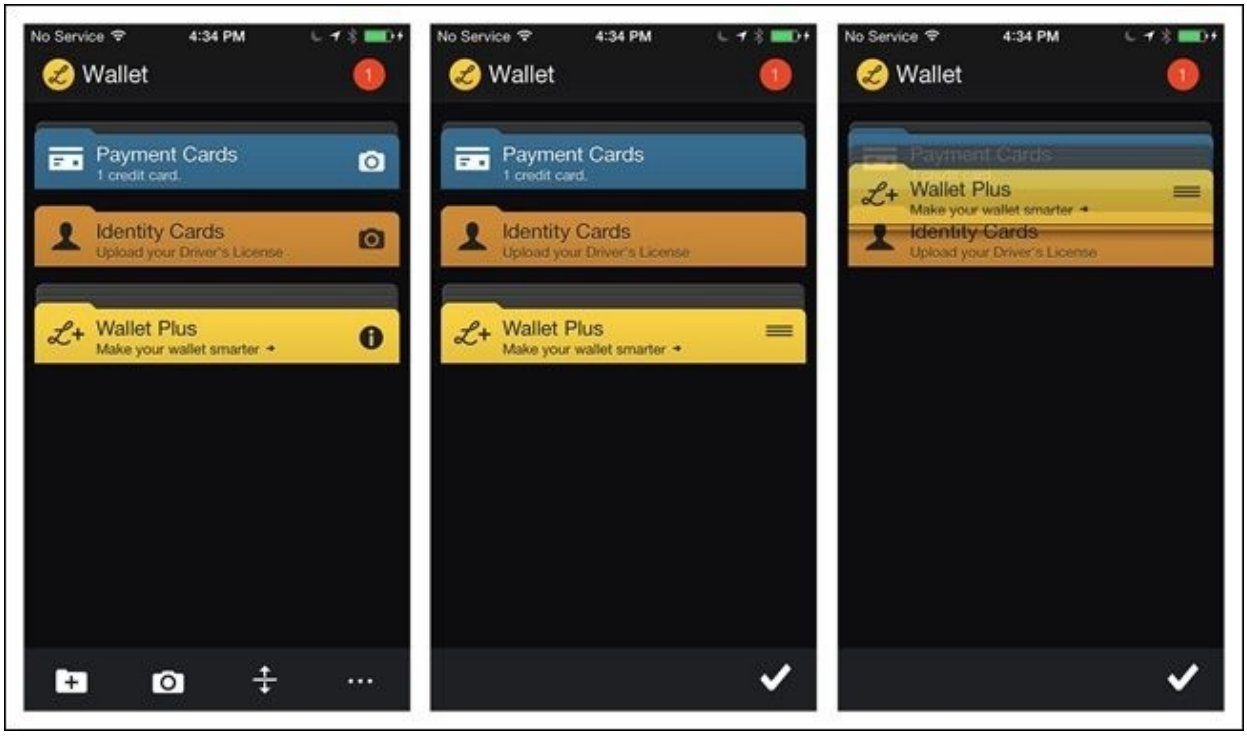



*Figure 9-36. Lemon Wallet for iOS: tapping Drag icon in the Action Bar enters edit mode and shows drag handles*

Some drag handles are visible by default, like those on slider controls and

drawers, as shown for Expedia and Lootsy.

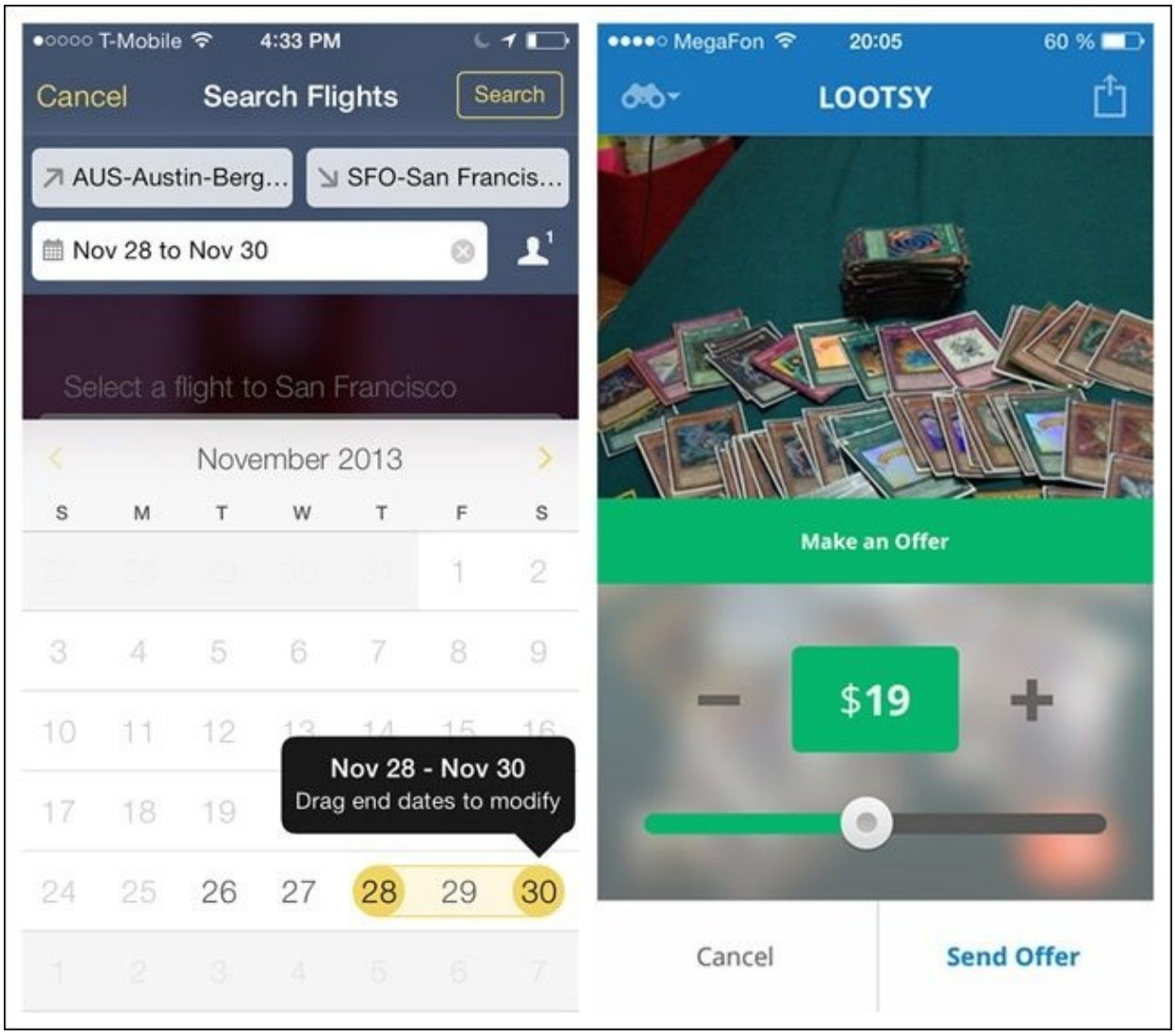



*Figure 9-37. Expedia and Lootsy for iOS: some drag handles, as on sliders, are always visible*

Design and configuration apps have more complex Drag interactions, like dragging to rotate and resize as well as to move. Aviary and Homestyler are worth exploring for good ideas on implementing Drag affordance.


*Figure 9-38. Aviary for iOS: multiple handles for crop; single handle for resizing, rotating, and placing the sticker*

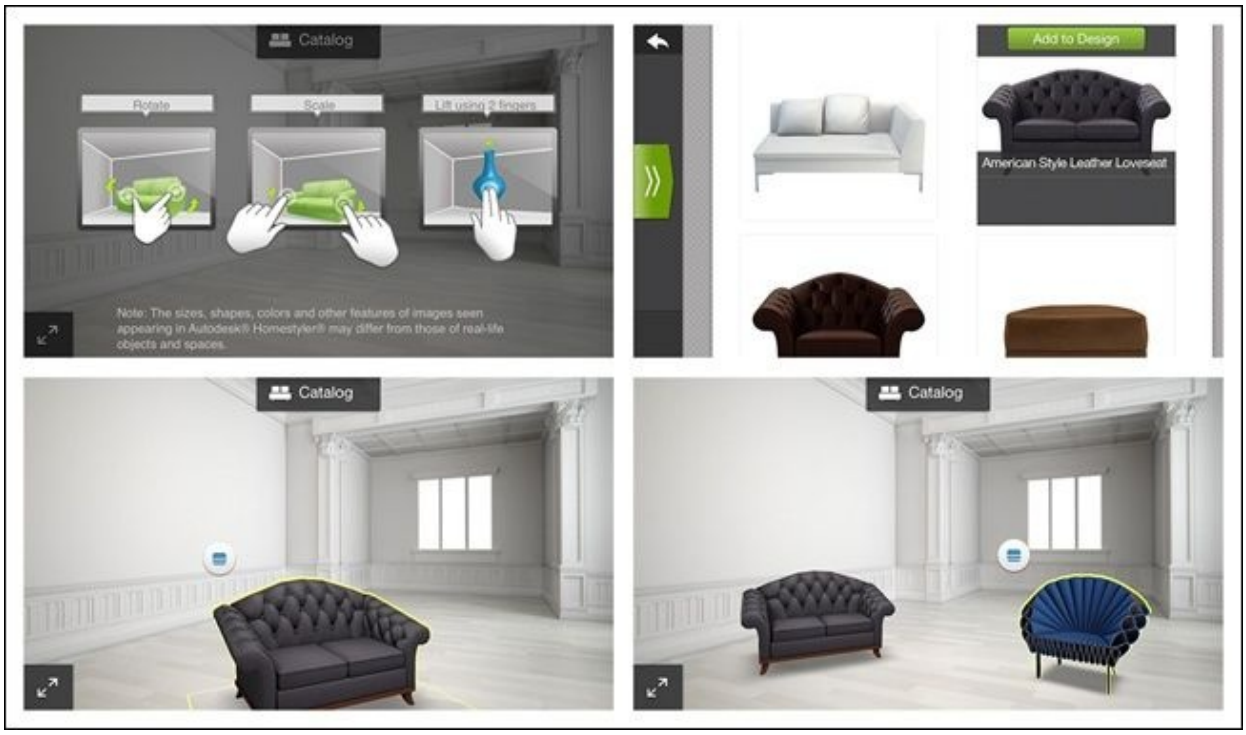
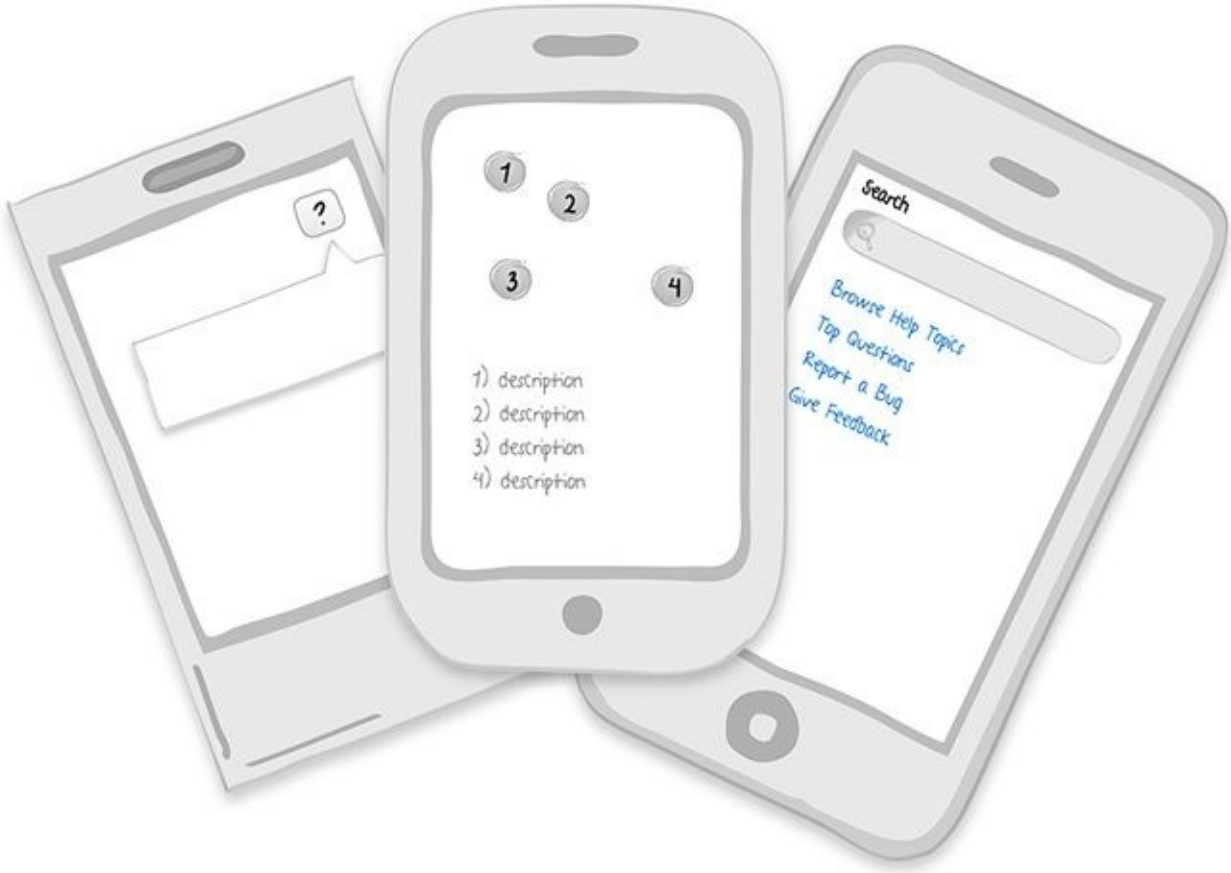



*Figure 9-39. Homestyler for iOS: clear affordance for items selected for repositioning*

**NOTE**

Use a recognizable treatment for the drag handle, selected object, and drop zone. Test to determine if an invitation is also needed to introduce users to the feature.

# Chapter 10. Help




*Patterns*

> How-Tos, User Guide/Help System, FAQs, Feature Tours, Tutorials, Contextual Help, Capture Feedback

We expect that great mobile applications will be easy to learn and quick to master. But that doesn't mean app designers and developers should assume users will simply figure things out on their own.

In a post on VentureBeat.com (*http://bit.ly/1hsU2B6*), mobile strategist Michael Mace cautions against this laissez-faire approach to app onboarding. "Mobile app developers need to assume that users will get stuck, and give them a path to graceful recovery," he says. "Time and again, we see users struggle to figure out how to perform even straightforward functions with mobile apps."

Sure, you understand your app, but you've spent weeks designing and coding it. Don't assume it will be so intuitive for your users. The help patterns discussed

here can keep your users on track, or at least let them recover gracefully:

- How-Tos
- User Guide/Help System
- FAQs
- Feature Tours
- Tutorials
- Contextual Help
- Capture Feedback

A single pattern for user help is probably not enough. Many successful apps use a combination of invitations and help patterns to aid users in uncovering features and getting the most from the tool. Waze, for instance, offers a Welcome Tour, a video How-To, and an interactive “Ask a Question” FAQ.

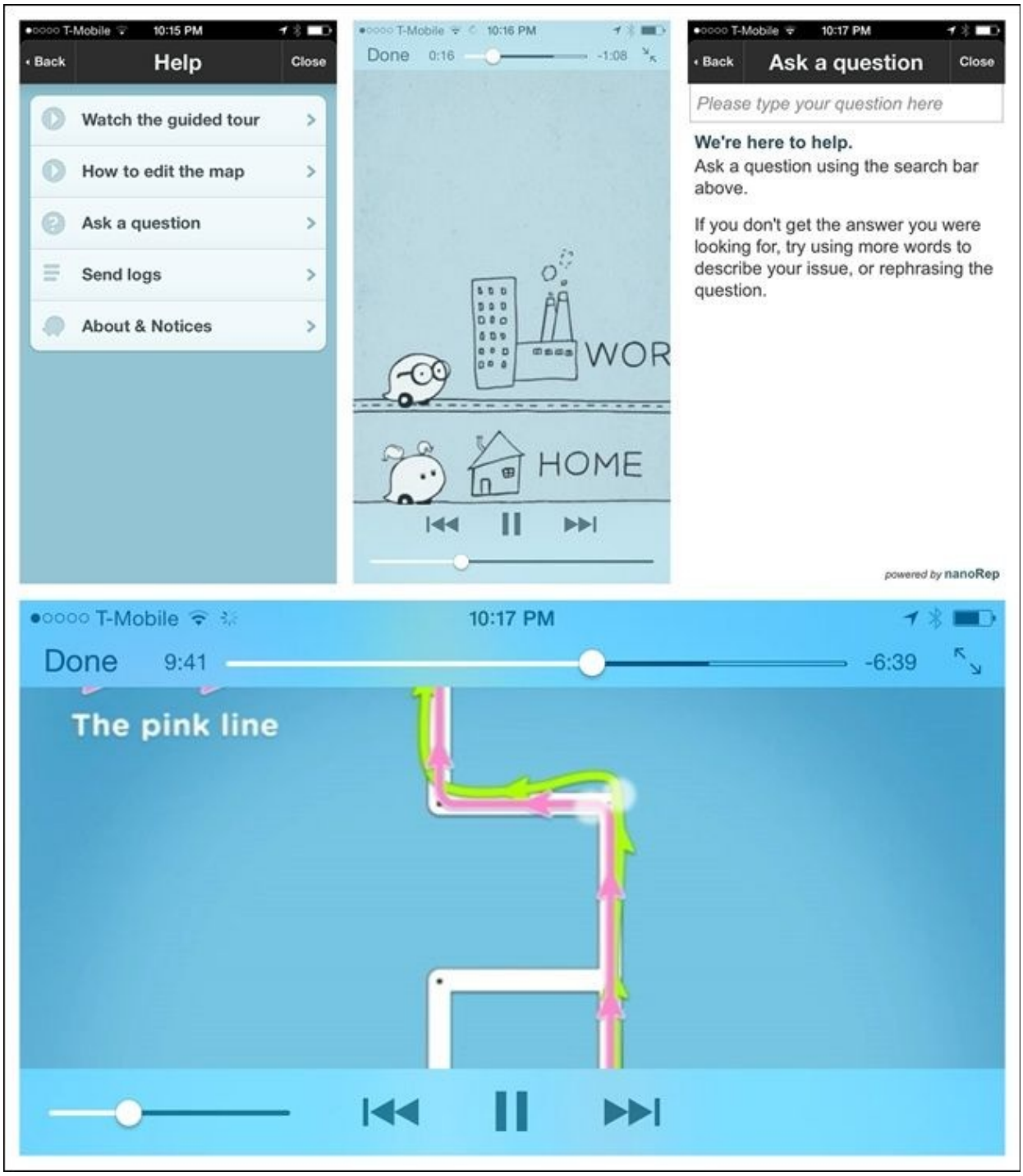



*Figure 10-1. Waze for iOS combines multiple patterns to provide in-app help*

Whenever possible, embed help content directly into the app rather than redirecting to an external website. If you must redirect to an external site, make sure its content is optimized for mobile, like Google Maps and SnipSnap. Kindle and Runtastic's Six Pack app get this wrong, causing even more frustration for

users.

> **NOTE**
>
> Embed help content directly into the app. If redirecting to a website, make sure the content is optimized for mobile consumption.

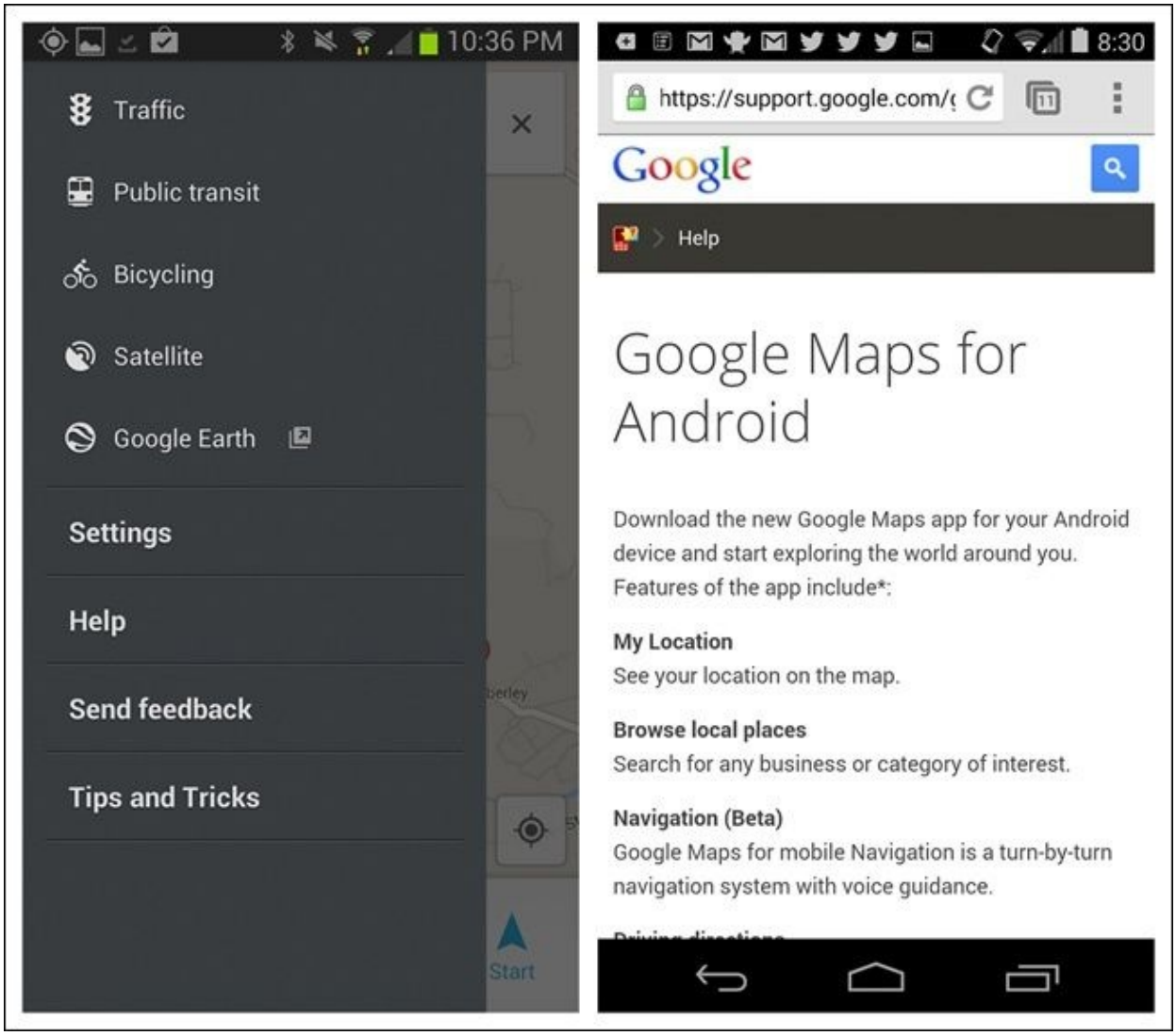



*Figure 10-2. Google Maps for Android opens Help in the browser*

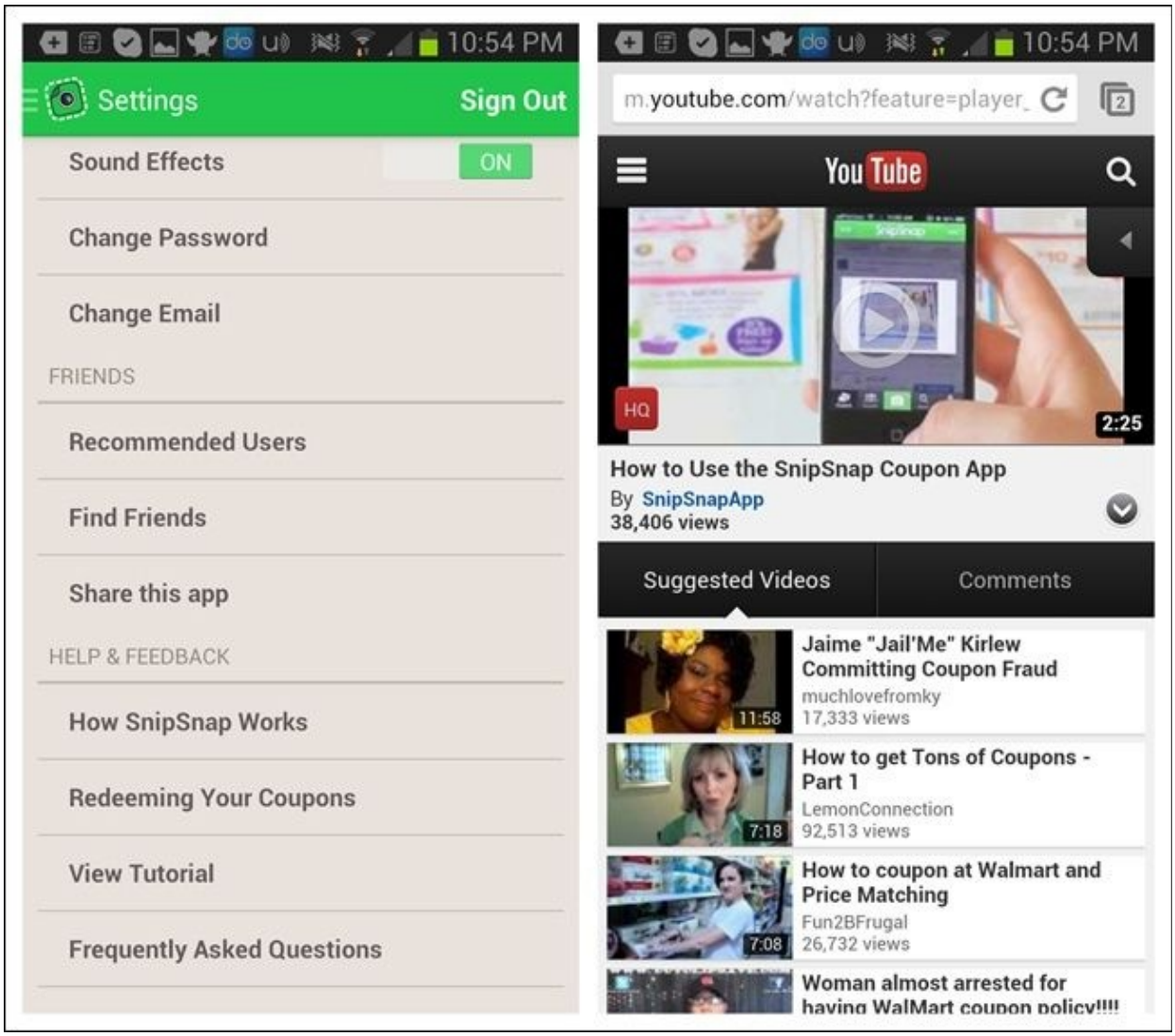



*Figure 10-3. SnipSnap for Android opens a tutorial video in the browser*

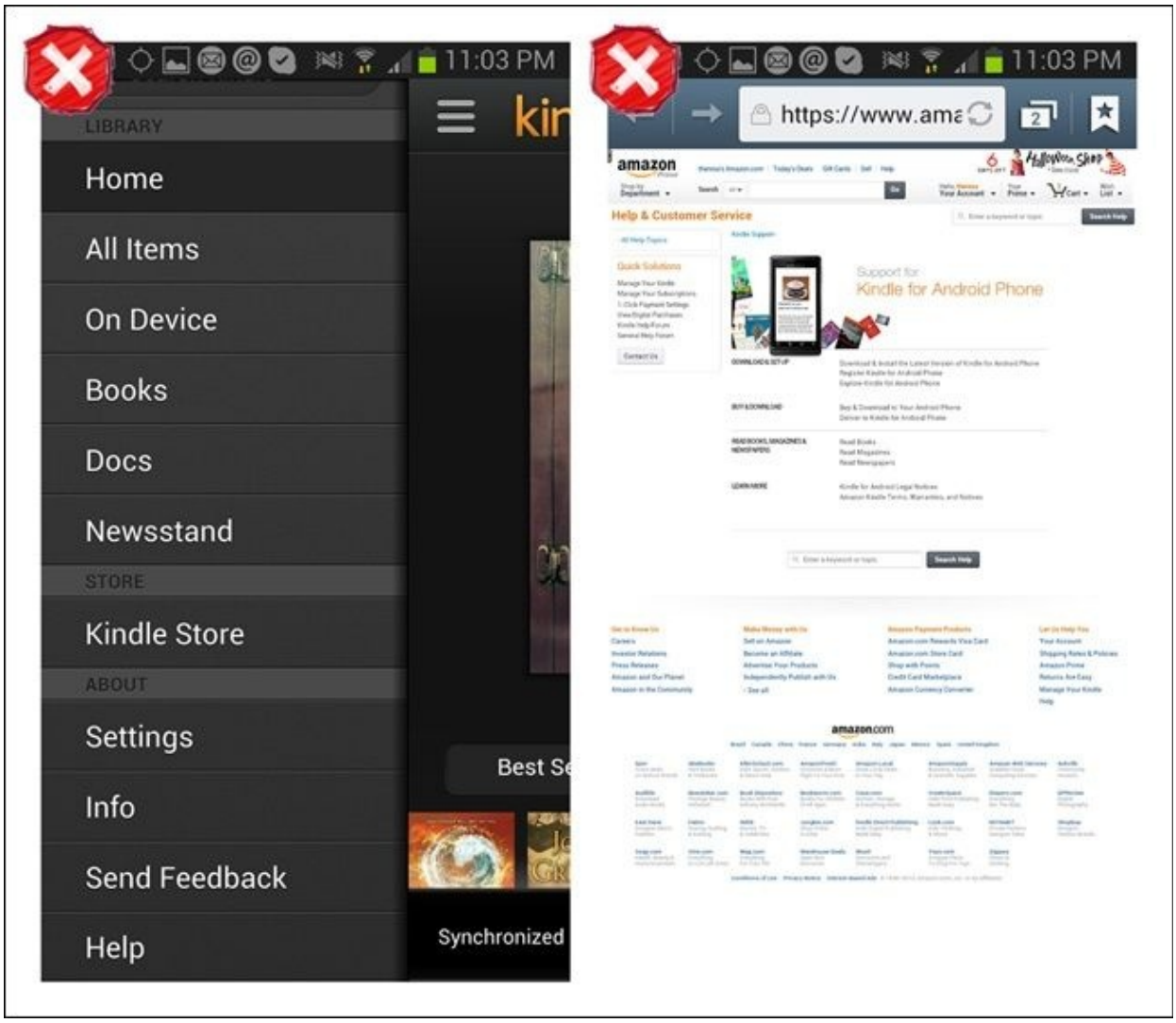



*Figure 10-4. Kindle for Android opens the classic website instead of a mobile-optimized version*

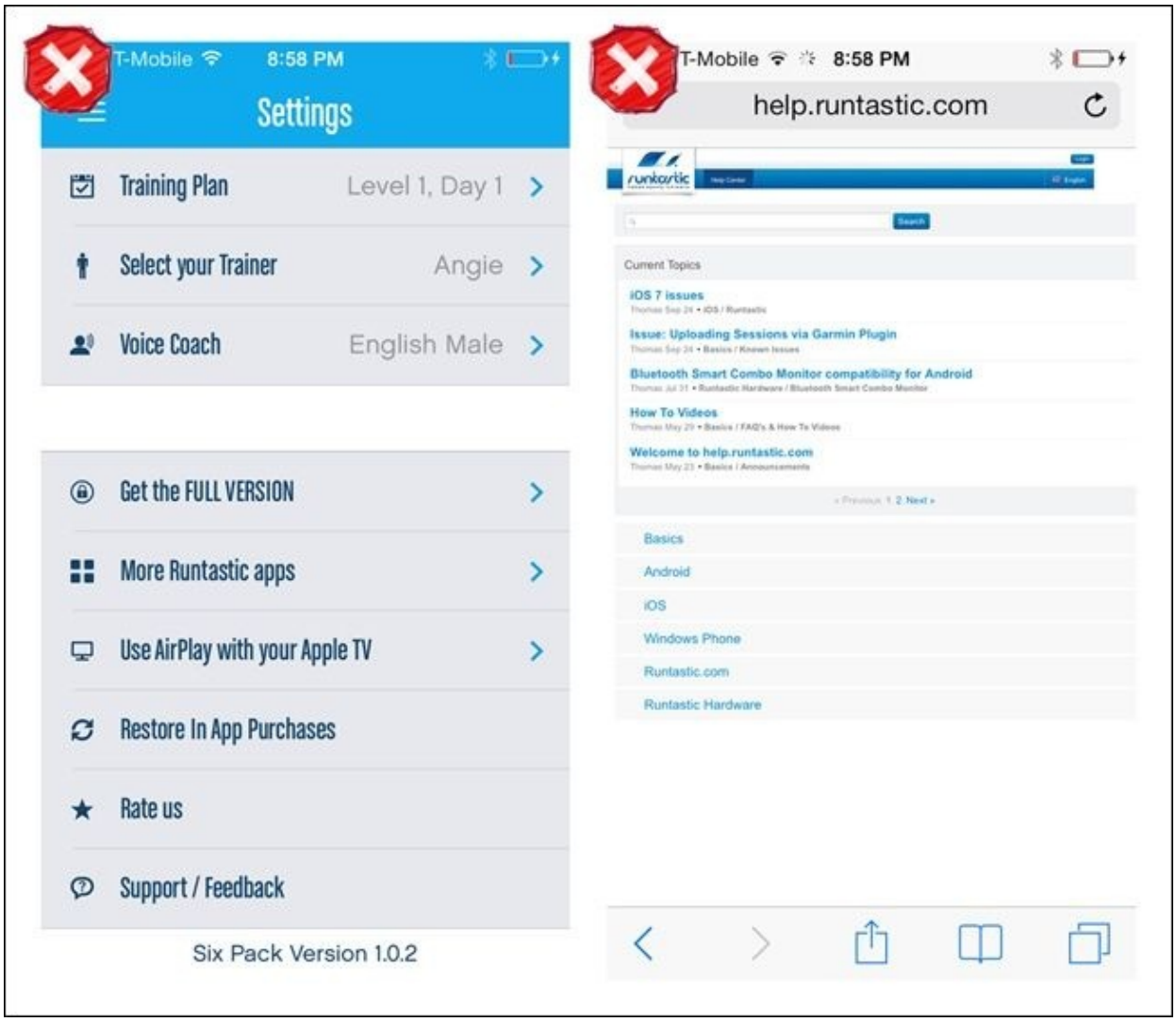


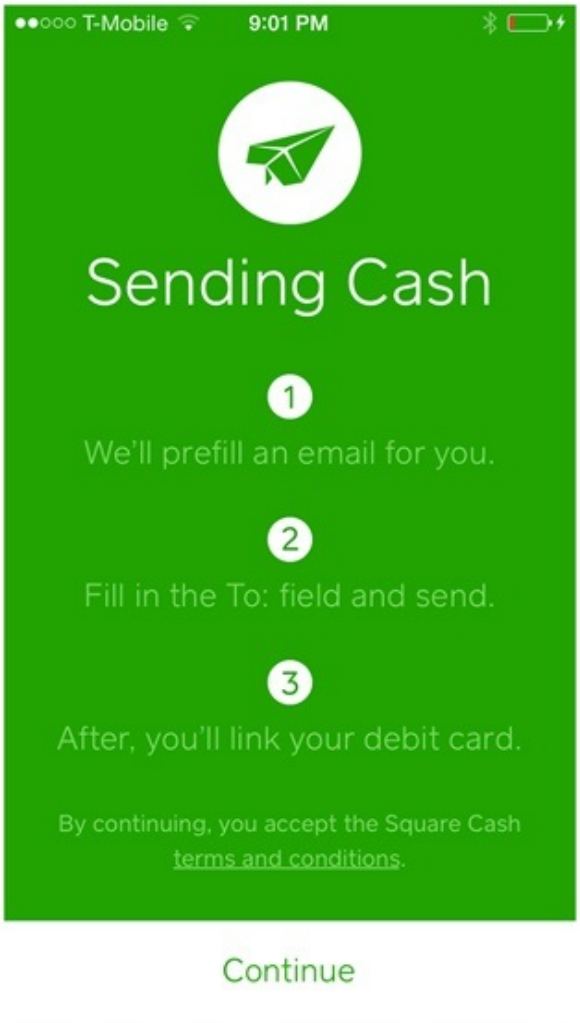

*Figure 10-5. Runtastic's Six Pack app for iOS opens a classic website instead of a mobile-optimized version*

## How-Tos

How-Tos are simple, succinct instructions that explain, well, how the application works. They can be a single page, as in the Gyft and Cash apps; a video demo, like in Bump; or part of a larger help system, like in Pocket.

> **NOTE**
>
> The How-To pattern works great for communicating step-by-step workflows. Keep it brief.


5:33 PM
gyft
How Gyft works
1
UPLOAD GIFT CARDS
Add and store all your physical gift cards securely in your mobile Gyft wallet.
2
SEND & RECEIVE
Buy and send gift cards to friends via Facebook, email or text, or regift any card from your Gyft wallet.
3
REDEEM
Simply flash your phone at the cashier in-store. (Some retailers support online redemption.)


T-Mobile
9:01 PM
Sending Cash
1
We'll prefill an email for you.
2
Fill in the To: field and send.
3
After, you'll link your debit card.
By continuing, you accept the Square Cash terms and conditions.
Continue

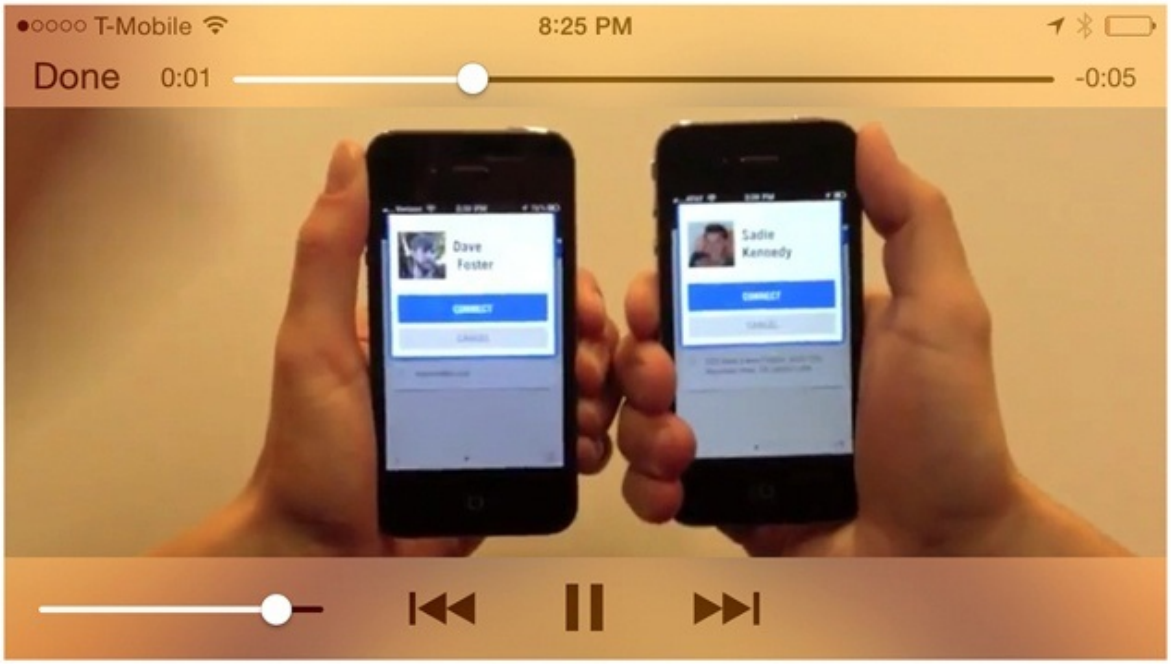

T-Mobile
8:25 PM
Done
0:01
-0:05
Dave Foster
Sadie Kennedy

*Figure 10-6. Gyft for Android and Cash for iOS: Single How-To page; Bump for iOS: How-To video*

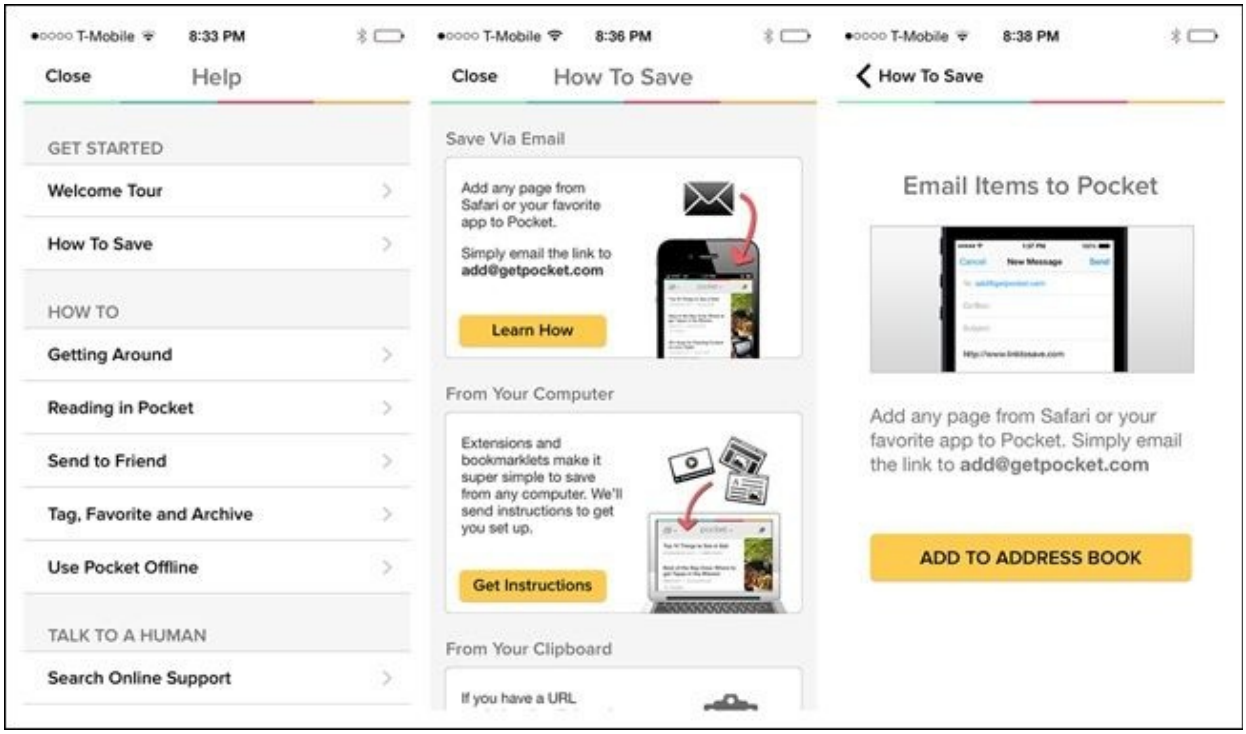



*Figure 10-7. Pocket for iOS: How-To combines screenshots, illustrations, text, and video*

Take a look at Chapter 7 for tips on how to best engage users in your application.

## User Guide/Help System

A User Guide will likely be more in-depth than a simple How-To. Pages and Facebook have extensive multilevel User Guides embedded in the apps.

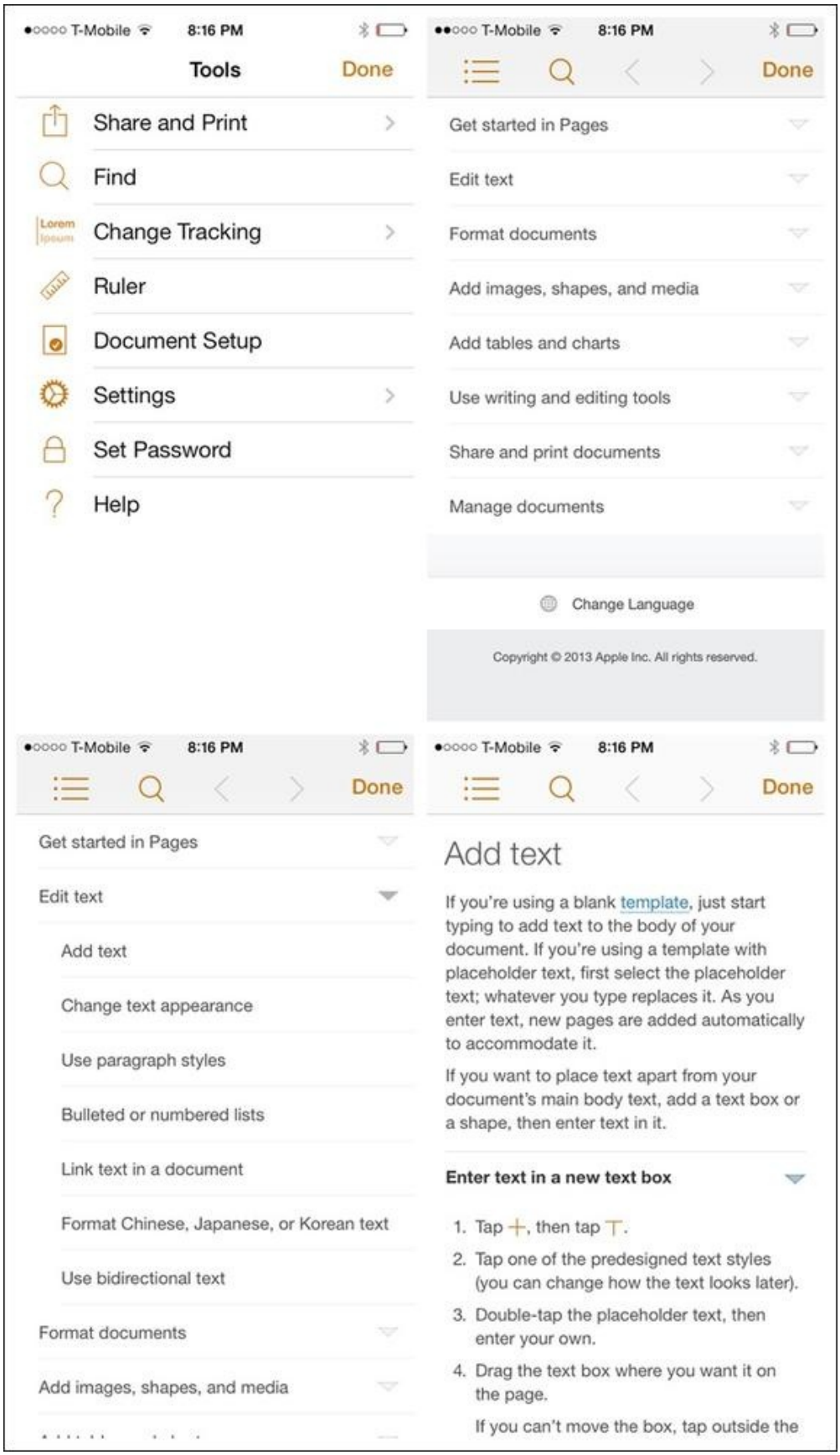

●○○○○ T-Mobile 8:16 PM
Tools
Done
Share and Print
Find
Change Tracking
Ruler
Document Setup
Settings
Set Password
Help
●●○○○ T-Mobile 8:16 PM
Done
Get started in Pages
Edit text
Format documents
Add images, shapes, and media
Add tables and charts
Use writing and editing tools
Share and print documents
Manage documents
Change Language
Copyright © 2013 Apple Inc. All rights reserved.
●○○○○ T-Mobile 8:16 PM
Done
Get started in Pages
Edit text
Add text
Change text appearance
Use paragraph styles
Bulleted or numbered lists
Link text in a document
Format Chinese, Japanese, or Korean text
Use bidirectional text
Format documents
Add images, shapes, and media
●○○○○ T-Mobile 8:16 PM
Done
Add text
If you're using a blank template, just start typing to add text to the body of your document. If you're using a template with placeholder text, first select the placeholder text; whatever you type replaces it. As you enter text, new pages are added automatically to accommodate it.
If you want to place text apart from your document's main body text, add a text box or a shape, then enter text in it.
Enter text in a new text box
1. Tap +, then tap T.
2. Tap one of the predesigned text styles (you can change how the text looks later).
3. Double-tap the placeholder text, then enter your own.
4. Drag the text box where you want it on the page.
If you can't move the box, tap outside the

*Figure 10-8. Pages for iOS: multilevel User Guides*

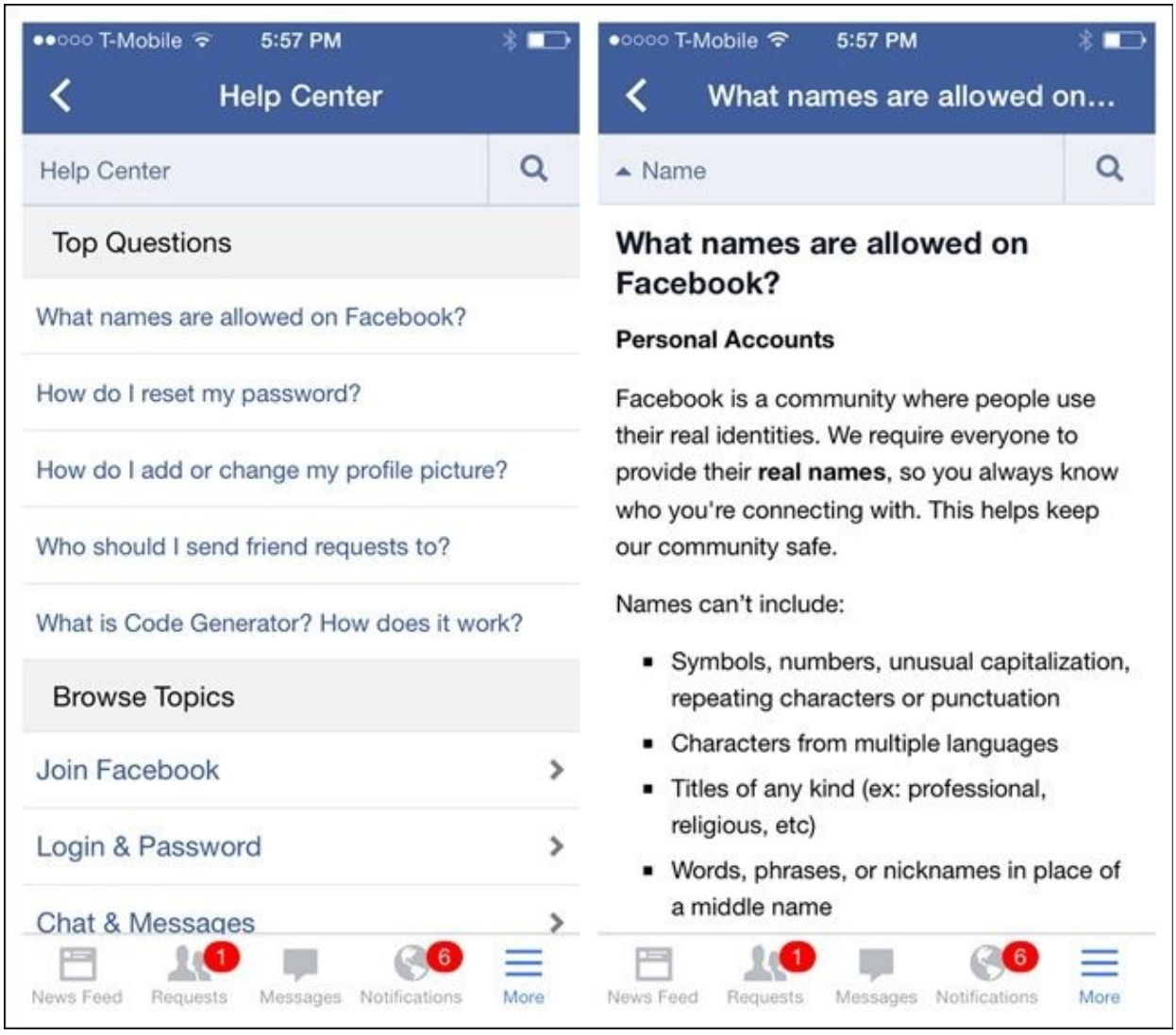



*Figure 10-9. Facebook for iOS: multilevel User Guides*

Like Facebook, Flava offers a User Guide and FAQs, both of which are well organized and easy to navigate.

**NOTE**

Organize the User Guide in logical categories or topics, and offer the ability to search its content.

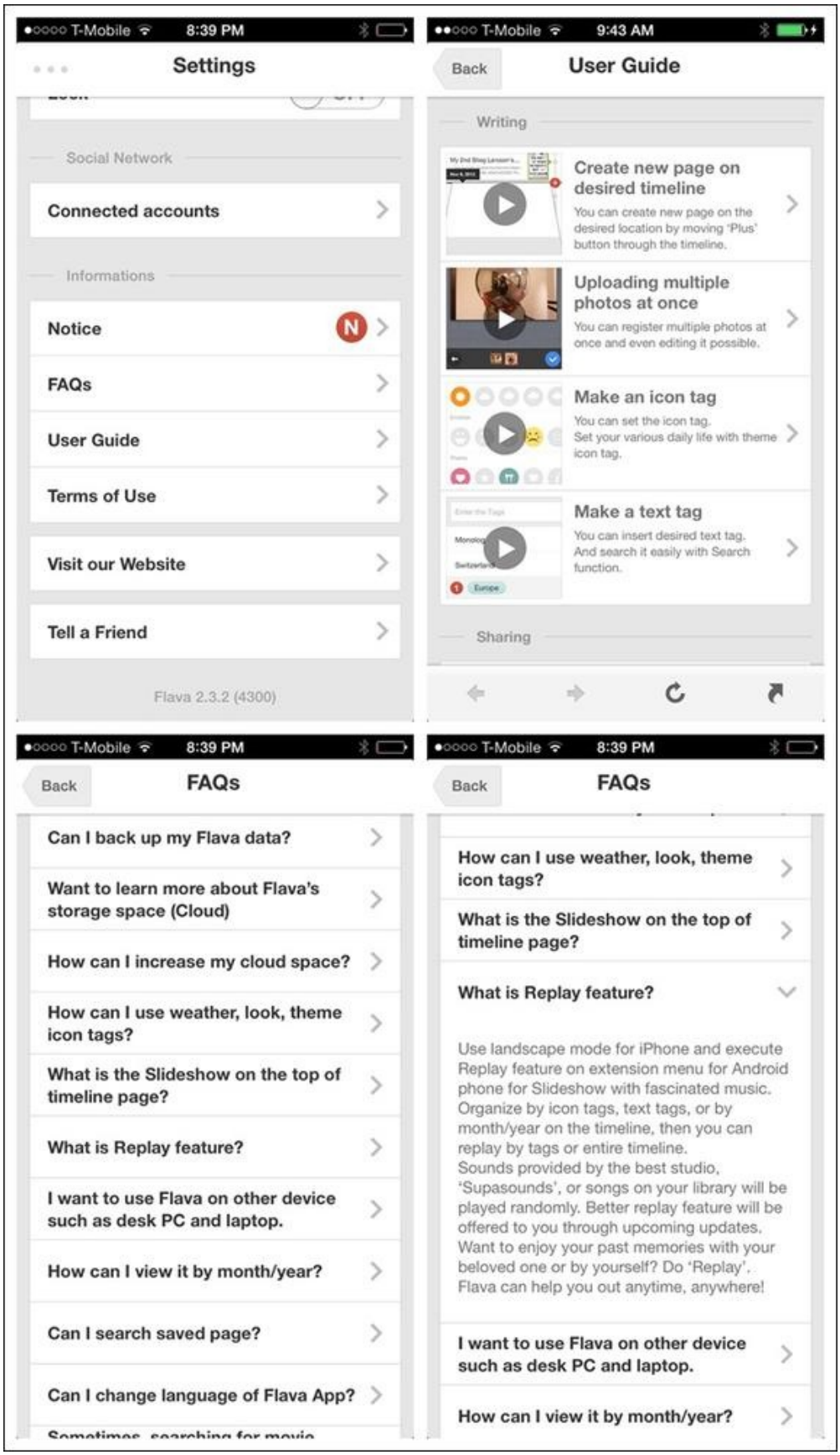

●○○○○ T-Mobile 8:39 PM
Settings
Social Network
Connected accounts
Informations
Notice
N
FAQs
User Guide
Terms of Use
Visit our Website
Tell a Friend
Flava 2.3.2 (4300)
●●○○○ T-Mobile 9:43 AM
Back
User Guide
Writing
Create new page on desired timeline
You can create new page on the desired location by moving 'Plus' button through the timeline.
Uploading multiple photos at once
You can register multiple photos at once and even editing it possible.
Make an icon tag
You can set the icon tag.
Set your various daily life with theme icon tag.
Make a text tag
You can insert desired text tag.
And search it easily with Search function.
Sharing
●○○○○ T-Mobile 8:39 PM
Back
FAQs
Can I back up my Flava data?
Want to learn more about Flava's storage space (Cloud)
How can I increase my cloud space?
How can I use weather, look, theme icon tags?
What is the Slideshow on the top of timeline page?
What is Replay feature?
I want to use Flava on other device such as desk PC and laptop.
How can I view it by month/year?
Can I search saved page?
Can I change language of Flava App?
●○○○○ T-Mobile 8:39 PM
Back
FAQs
How can I use weather, look, theme icon tags?
What is the Slideshow on the top of timeline page?
What is Replay feature?
Use landscape mode for iPhone and execute Replay feature on extension menu for Android phone for Slideshow with fascinated music.
Organize by icon tags, text tags, or by month/year on the timeline, then you can replay by tags or entire timeline.
Sounds provided by the best studio, 'Supasounds', or songs on your library will be played randomly. Better replay feature will be offered to you through upcoming updates.
Want to enjoy your past memories with your beloved one or by yourself? Do 'Replay'. Flava can help you out anytime, anywhere!
I want to use Flava on other device such as desk PC and laptop.
How can I view it by month/year?

*Figure 10-10. Flava for iOS includes a User Guide and FAQs*

# FAQs

Got more information than a simple How-To can impart, but not enough for a full User Guide? A list of FAQs might be just the solution. FAQs are typically ordered by most common questions first. (These can be identified through pre-release user testing, by the way.) Dropbox uses the standard iOS pattern for drilling down into FAQ answers.

**NOTE**

Make the list of FAQs visually easy to scan and ensure that revealing answers is intuitive. Order the most common questions first.

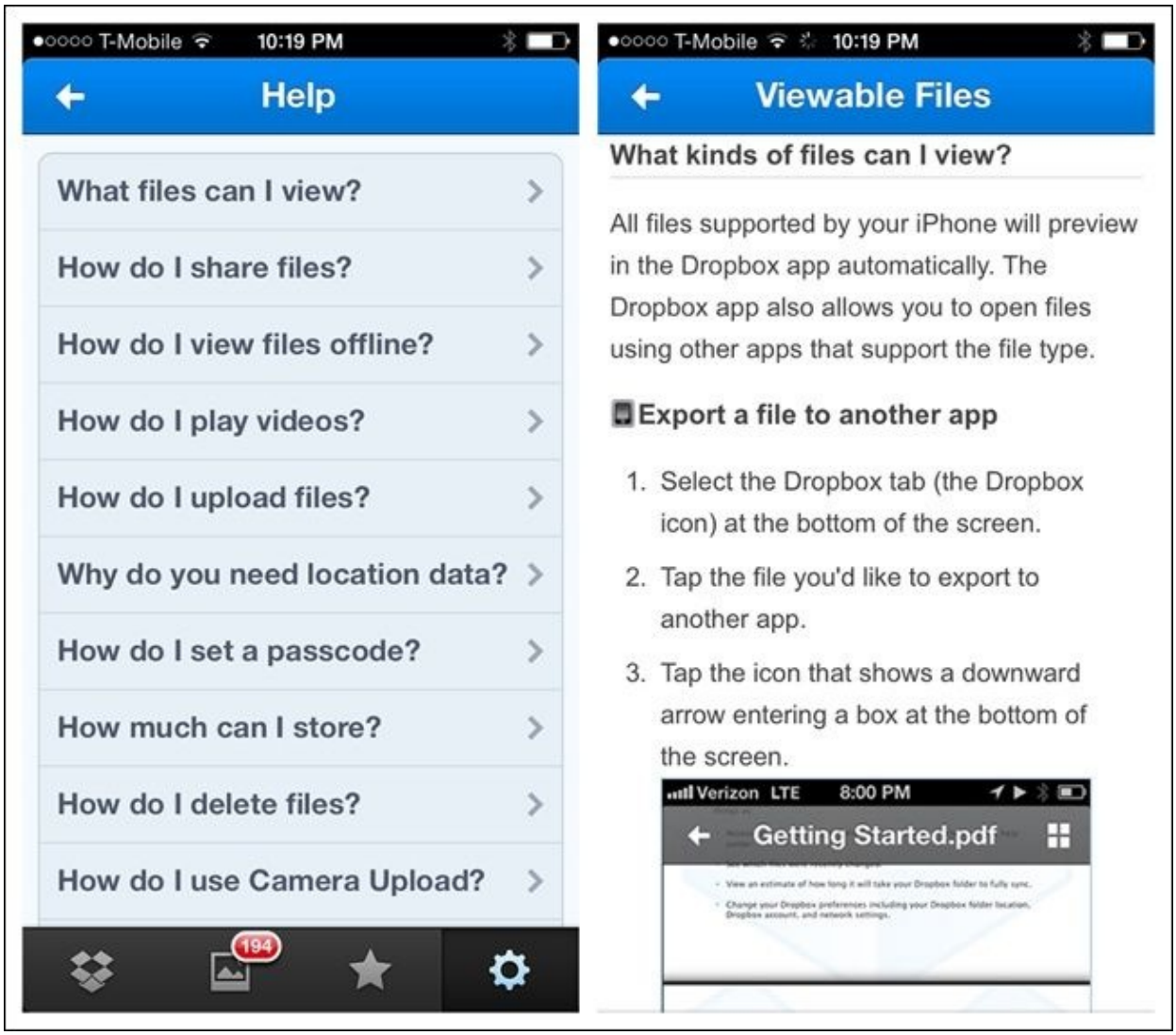



*Figure 10-11. Dropbox for iOS: drill-down FAQs*

Audible uses inline expansion instead of drill-down to reveal answers to Frequently Asked Questions.

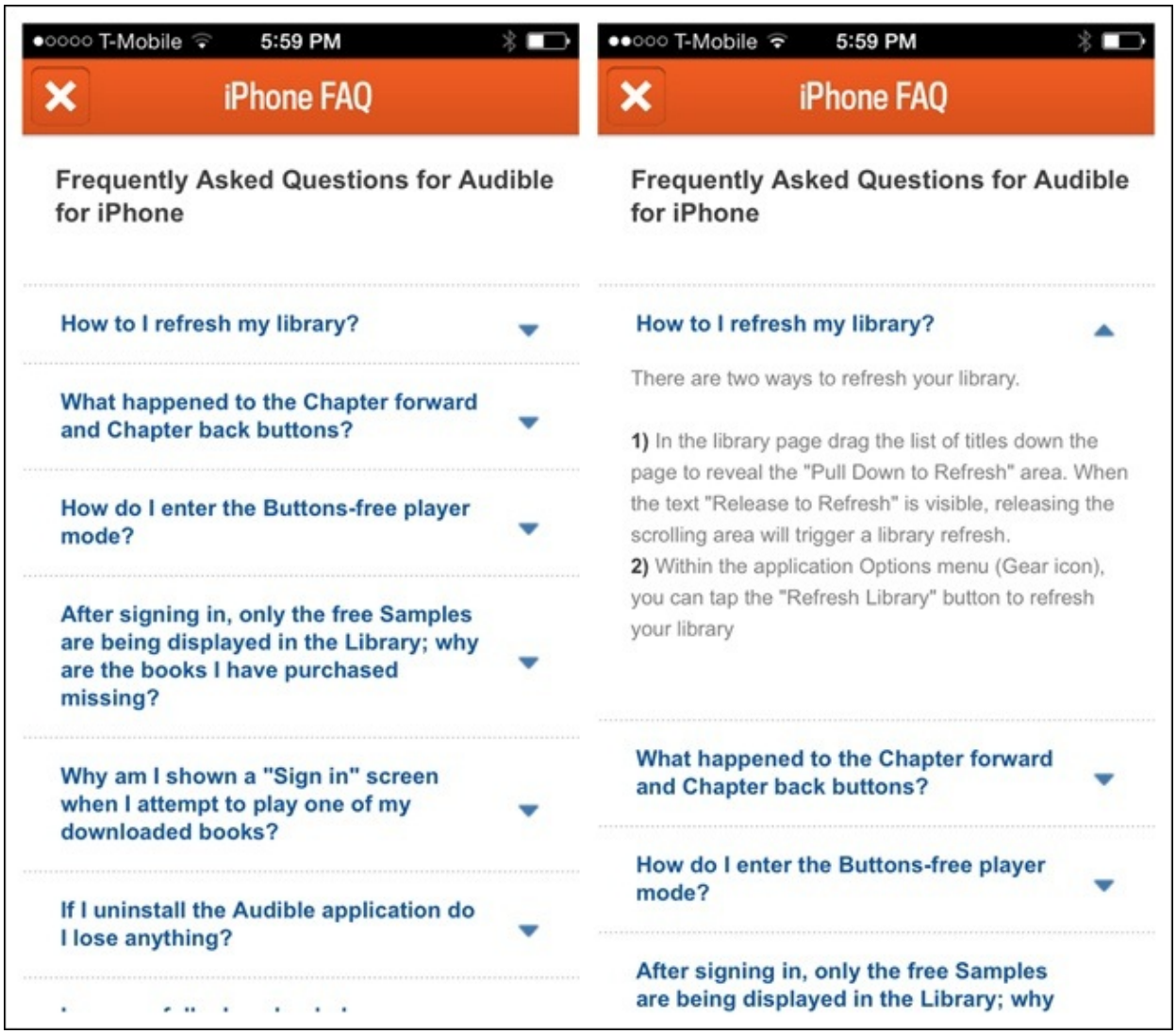



*Figure 10-12. Audible for iOS: FAQs expand onscreen*

# Feature Tours

A Feature Tour may be used to highlight specific features and functionality in an app. This is not to be confused with a Product Tour, which highlights the app's value proposition, as shown in the PayPal example.

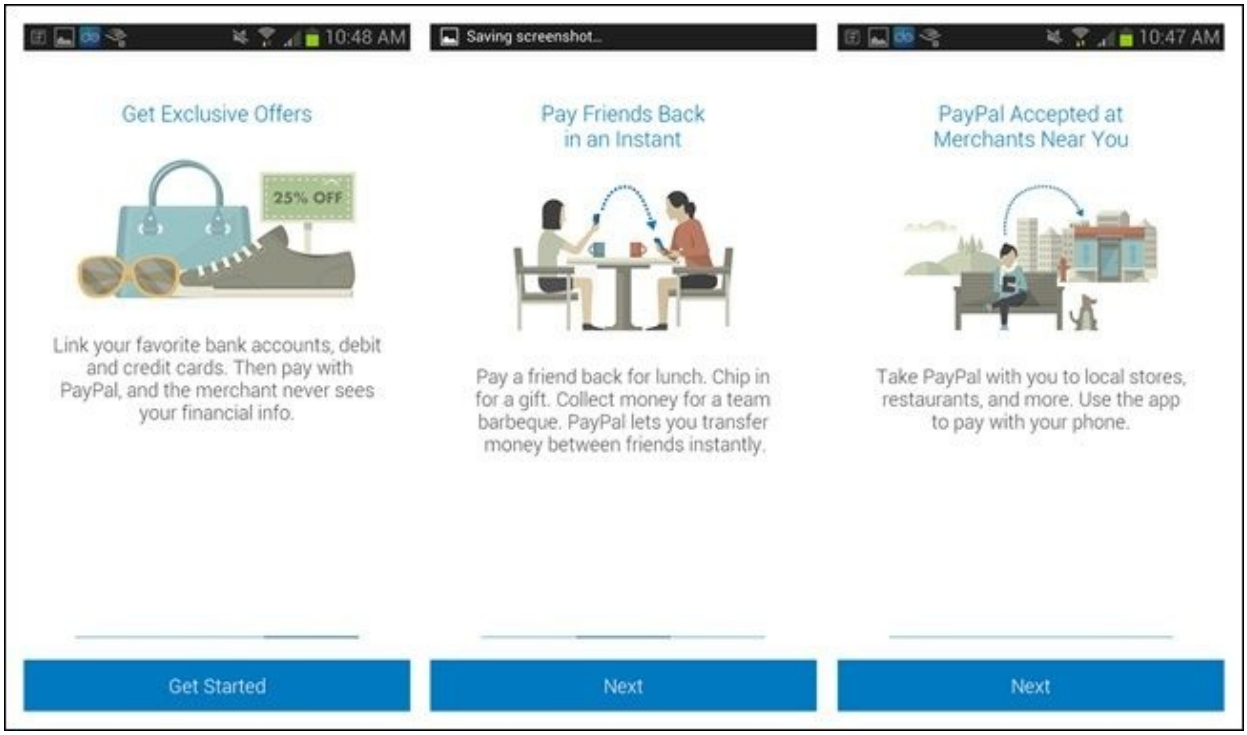



*Figure 10-13. PayPal for Android: a Product Tour to communicate the app's value proposition*

Feature Tours are typically presented on first use but can be accessible at a later time. Fidelity enables this with a Bring Back Instructional Guides setting.

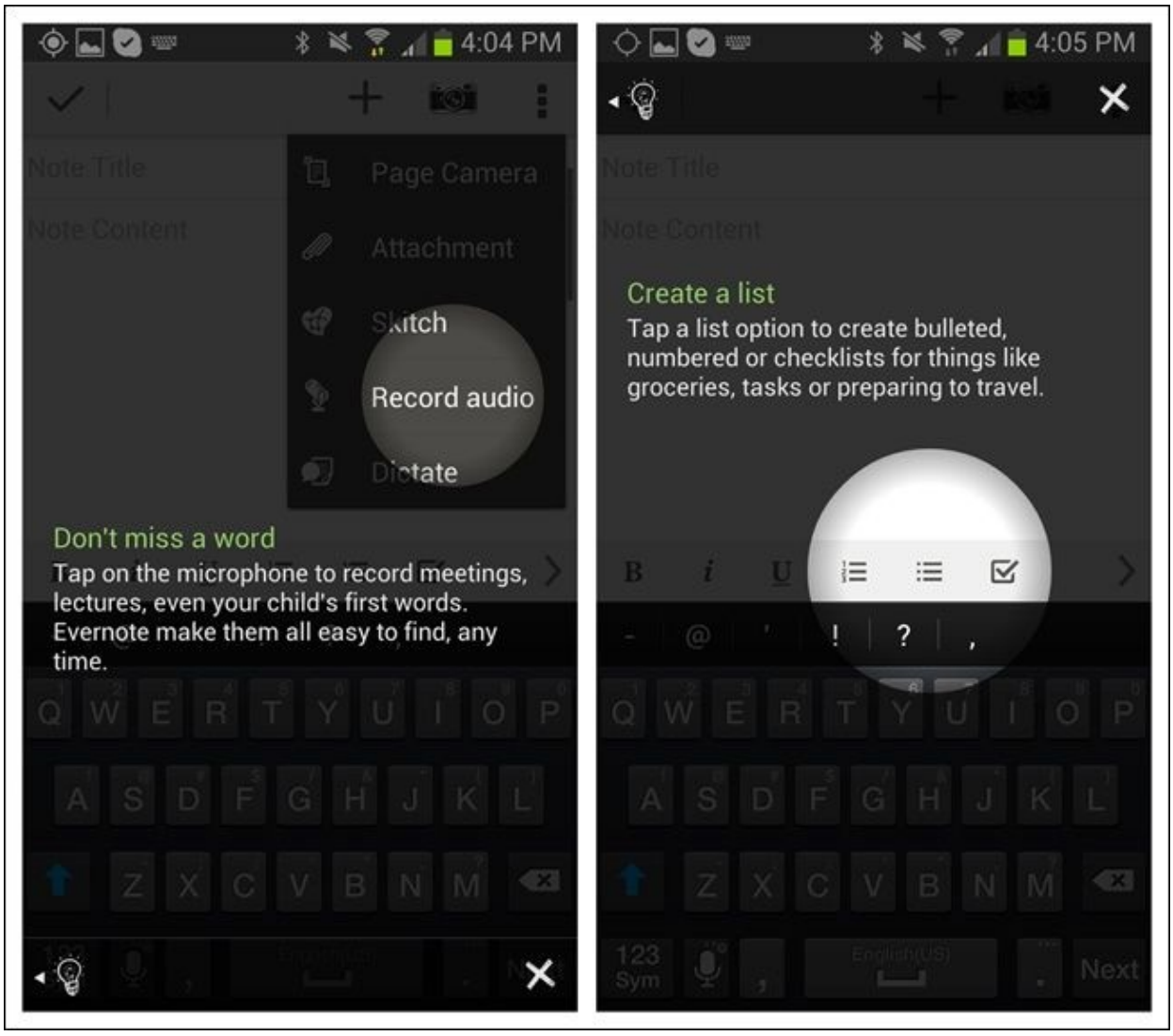



*Figure 10-14. Evernote for Android: Feature Tour highlights specific features and UI controls*


*Figure 10-15. Fidelity for iOS: Tour relaunch option — Bring Back Instructional Guides*

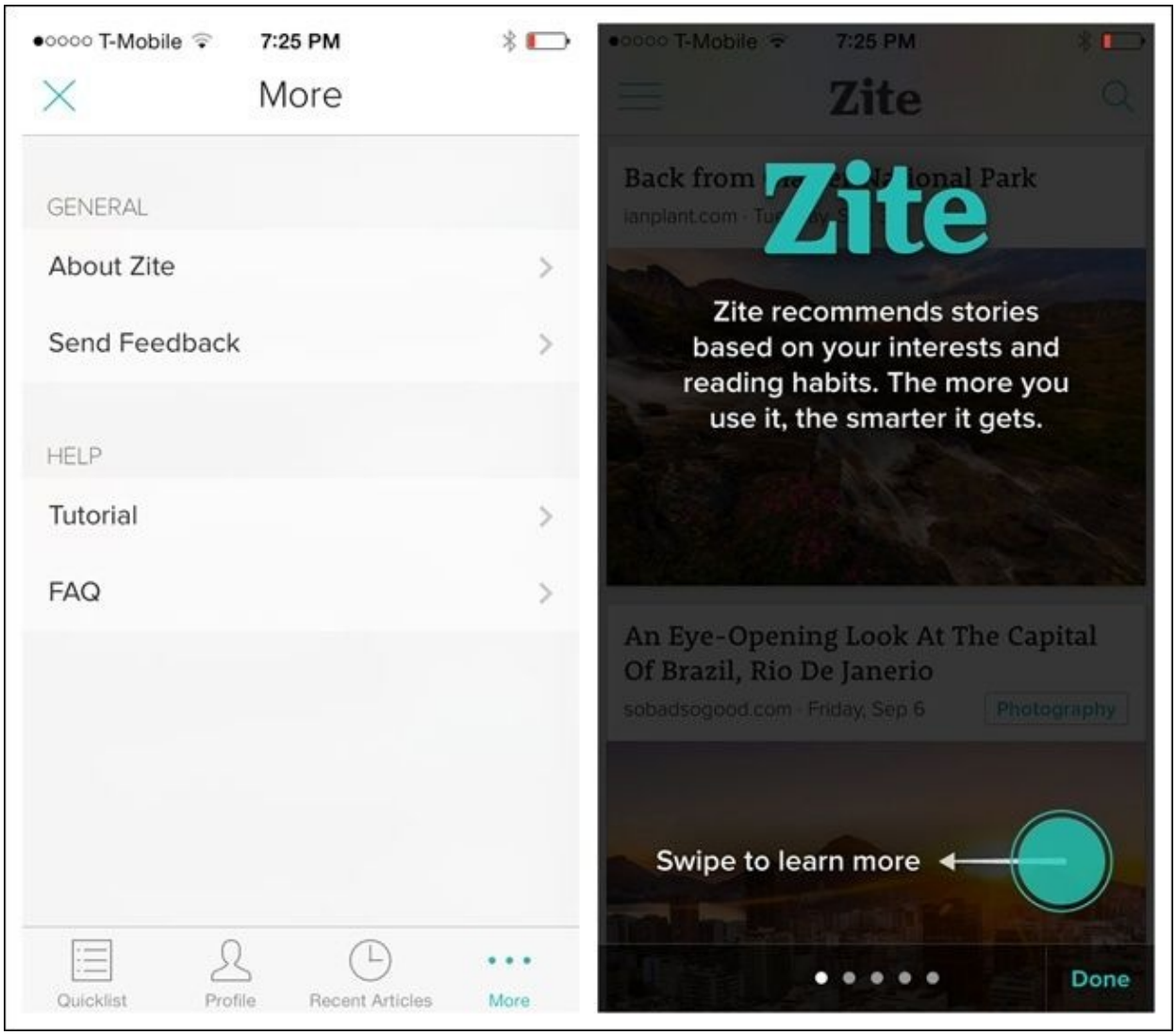



*Figure 10-16. Zite for iOS: the Tour can be launched again from the More menu*

As nifty as they may seem, user testing has found that even well-designed Feature Tours are usually skipped. They are also a source of irritation when users want to jump straight into the app.

> **NOTE**
>
> Introducing a Feature Tour can have a negative impact on conversion and adoption; review Chapter 7 for best practices in designing interactive Tutorials.

The best help option varies depending on the application and the audience, so make sure to test a variety of help patterns. Review the rules in Chapter 7 to avoid creating an ineffective Tutorial.

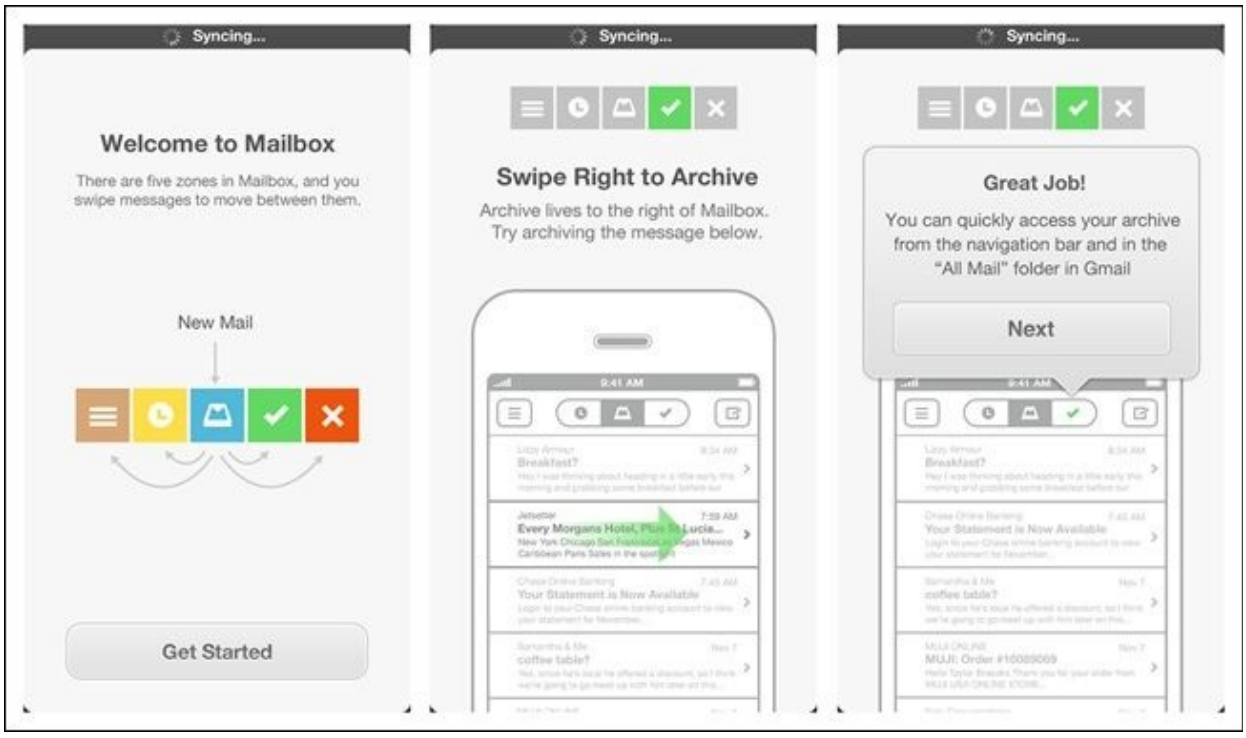



*Figure 10-17. Mailbox for iOS skips the Feature Tour and instead creates an interactive Tutorial (http://www.youtube.com/watch?v=URd0j5vEsHE)*


*Figure 10-18. By comparison, the Feature Tour in Boomerang for Android is long and overwhelming*

# Tutorials

As we discussed in Chapter 7, a well-designed Tutorial will get users invested in and comfortable with a product. Though the two patterns are similar in some ways, a Tutorial differs from a Feature Tour in that users are actually accomplishing something within the app as they learn about it. This pattern works well in apps that require initial configuration or rely on a multi-step process for engagement, like Pocket and Vine.

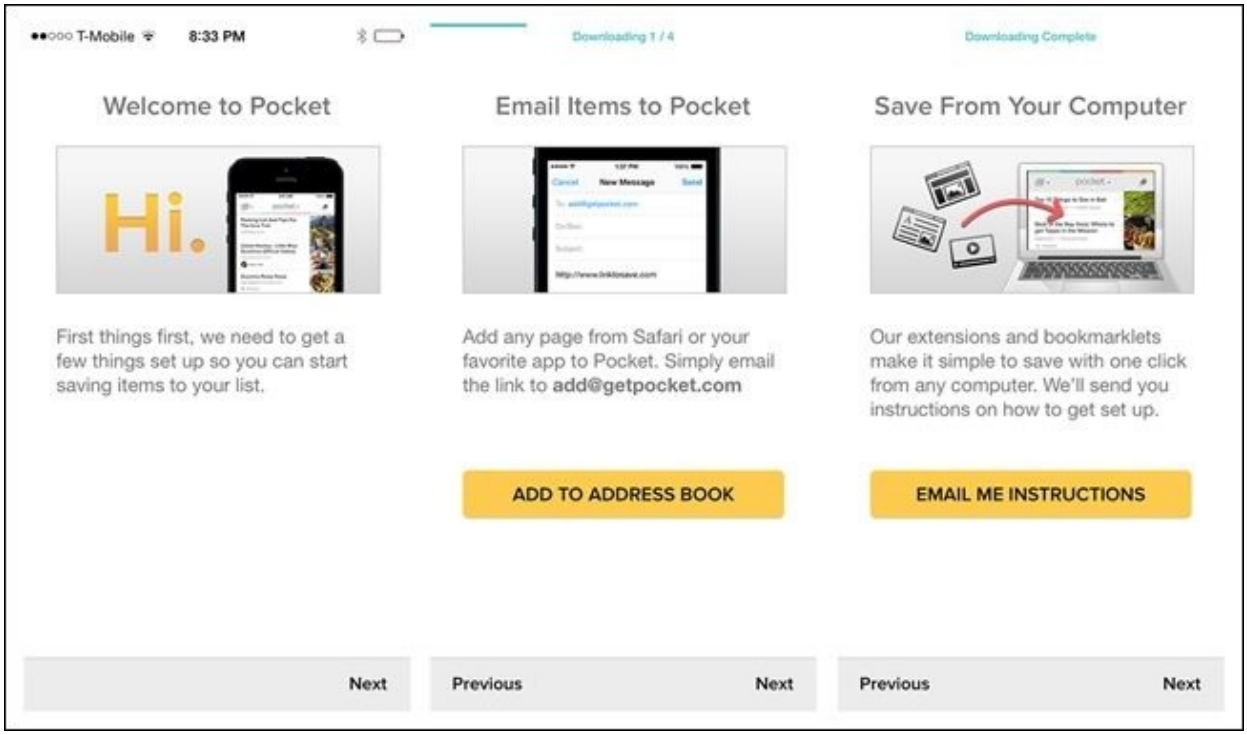



*Figure 10-19. Pocket for iOS: Tutorial*

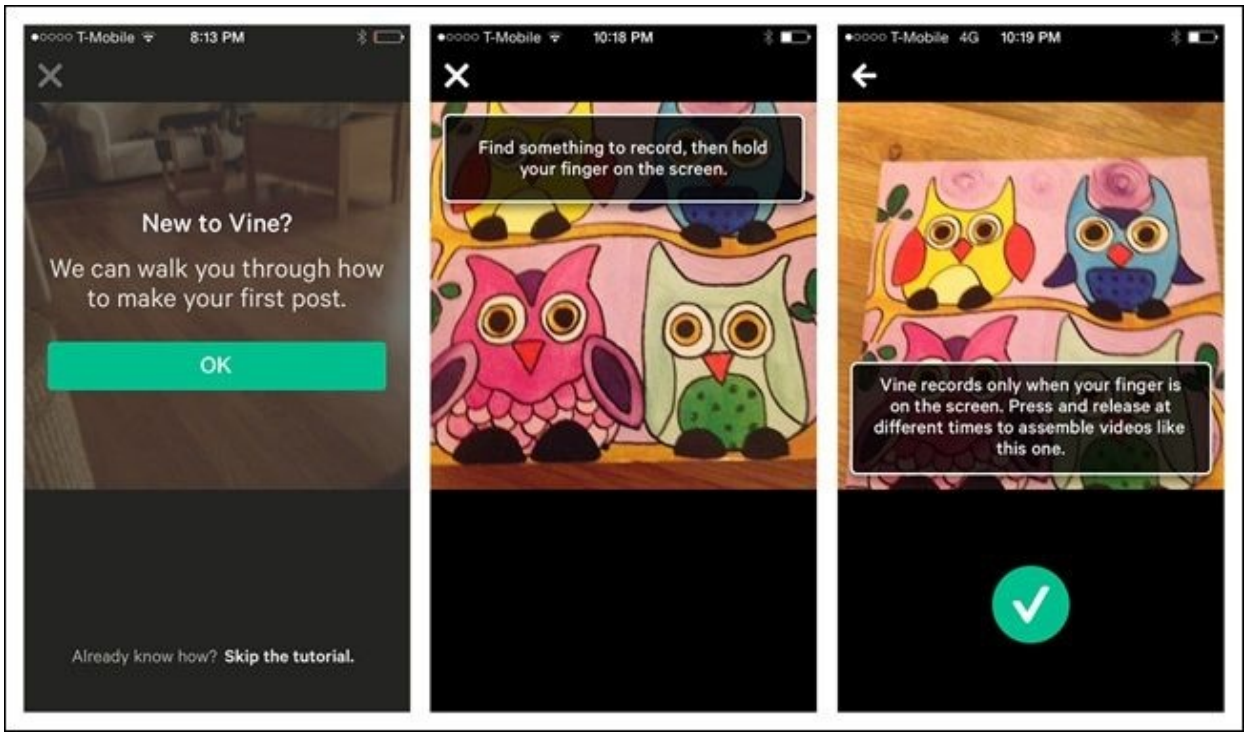



*Figure 10-20. Vine for iOS: general usage Tutorial (http://www.youtube.com/watch?v=vBOC9DpC5ns)*

Goal-based or behavior modification apps like Noom and Everest rely on their Tutorials for both onboarding and retaining users.

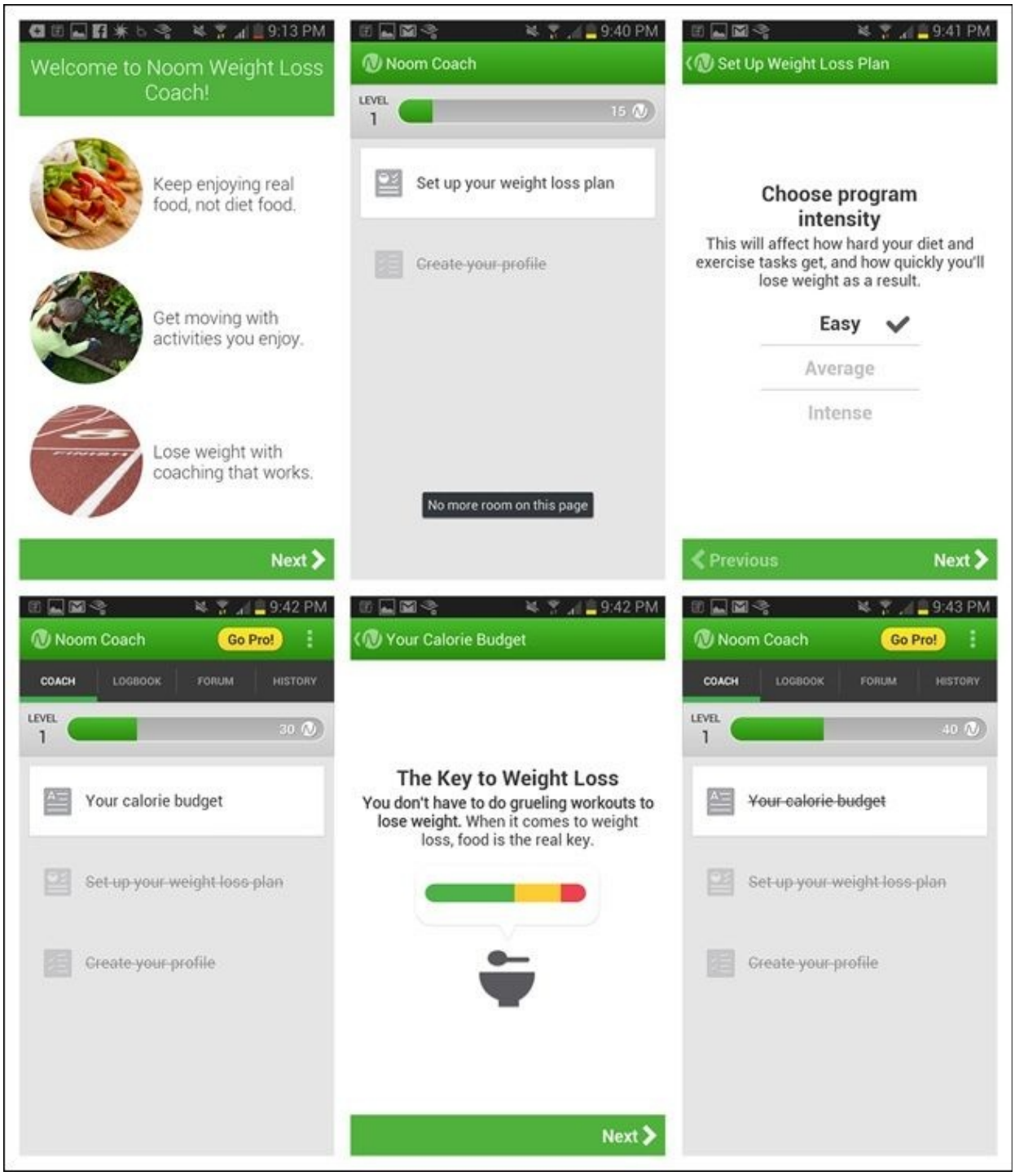



*Figure 10-21. Noom for Android: a noteworthy example of a Tutorial*

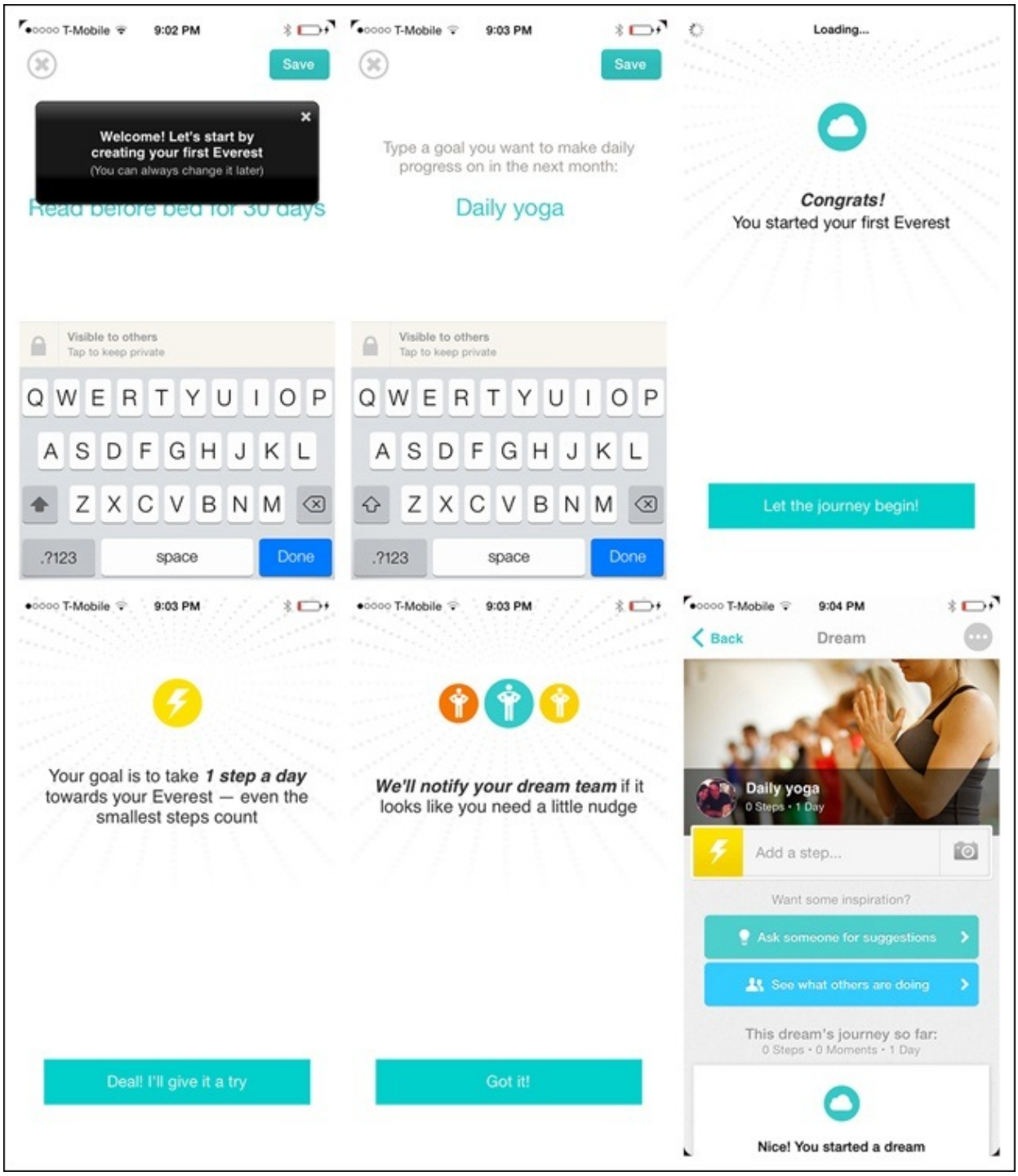



*Figure 10-22. Everest for iOS: Tutorial promotes engagement*

# Contextual Help

Contextual Help is help obtained at a specific point in the application. Unlike with a Feature Tour, there is no frontloading of information; the help is accessible at the screen level and/or is revealed during a task.

In Argus, tapping on a water droplet reveals the trick to adding and removing glasses of water.

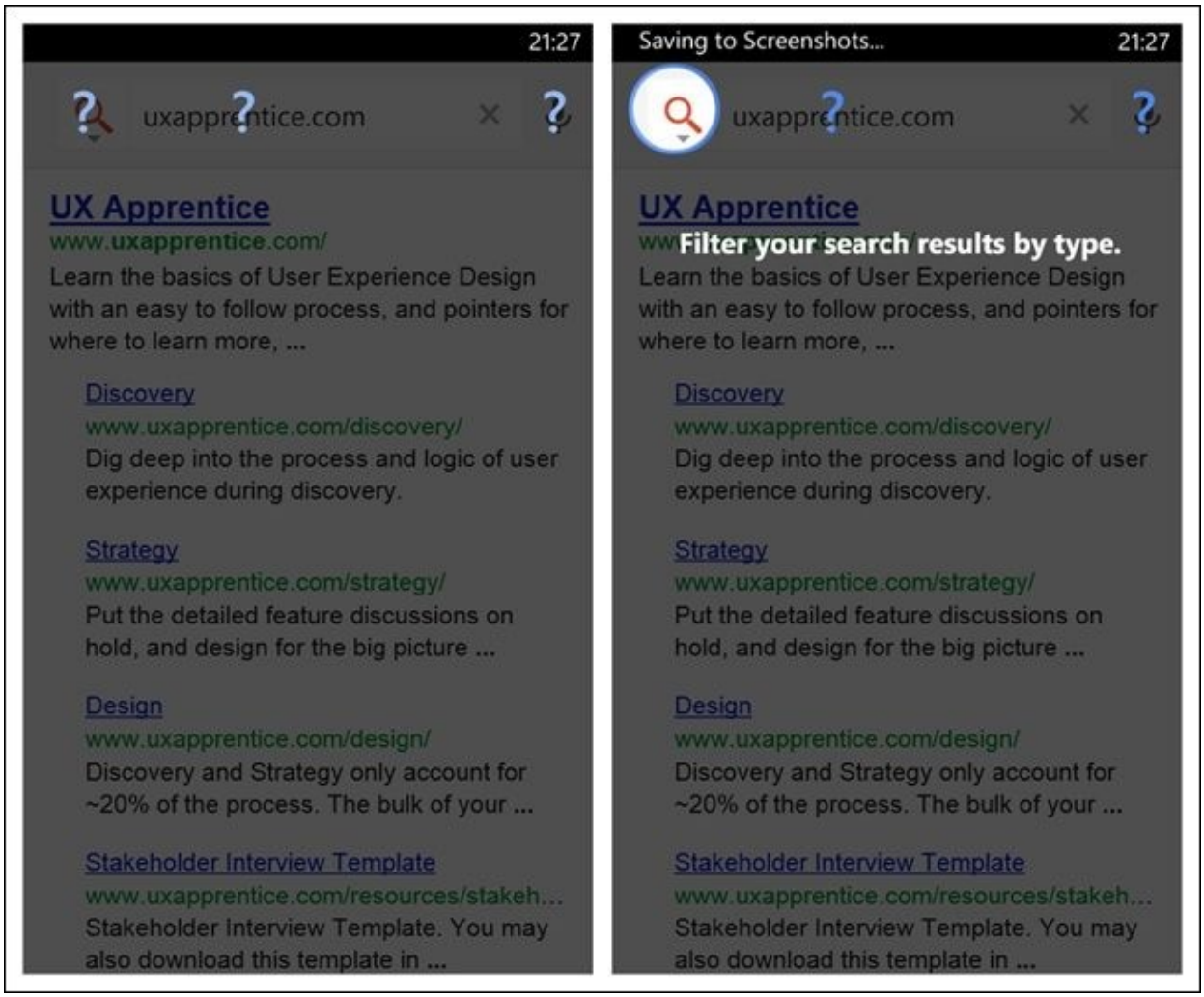



*Figure 10-23. Google for Windows Phone: user can reveal Contextual Help for specific buttons*


*Figure 10-24. Argus for iOS: tapping the water droplet reveals Contextual Help (http://youtu.be/FxvqBhs-2Co)*

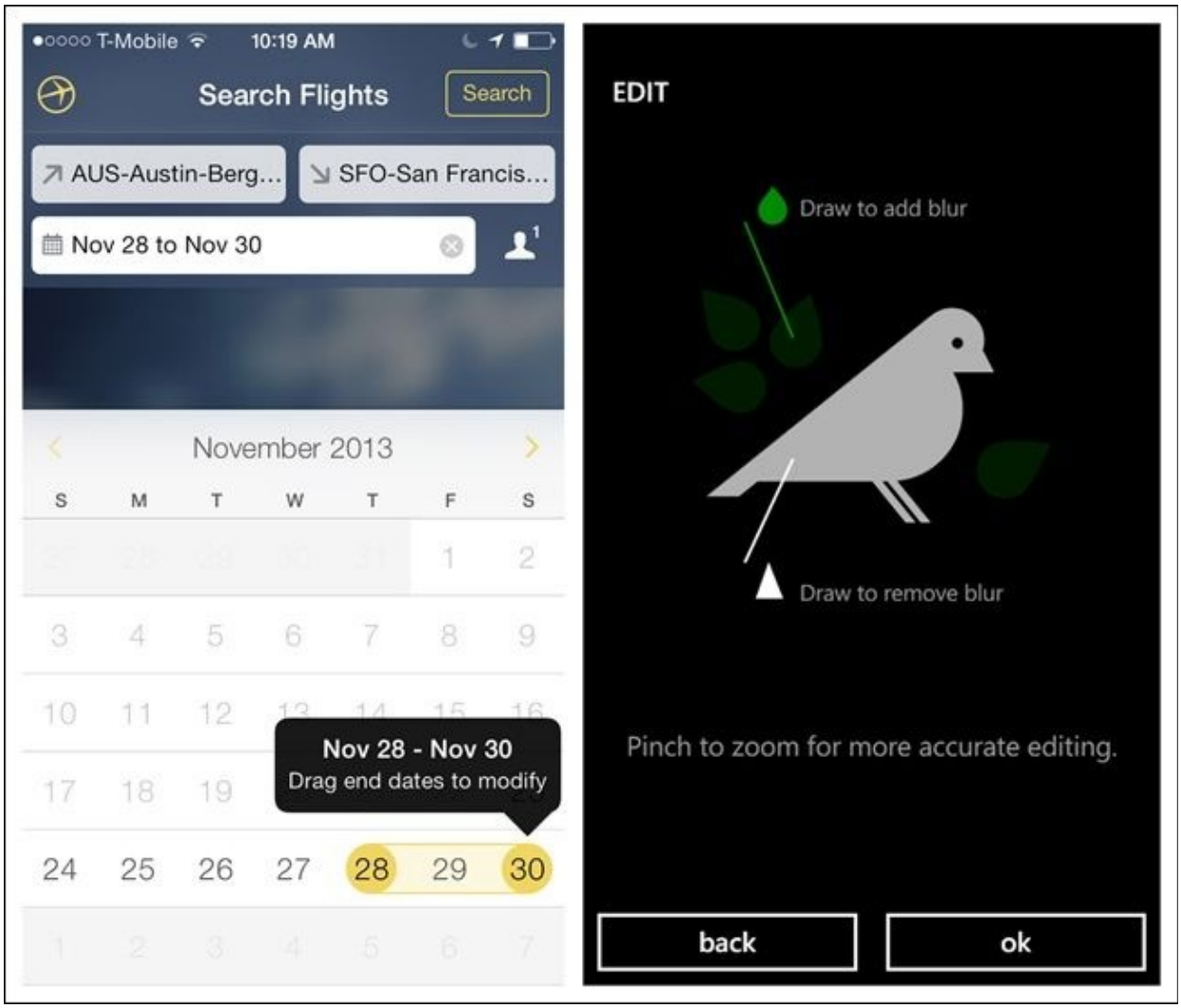



*Figure 10-25. Expedia for iOS and Creative Studio for Windows Phone offer help during the task flow*

## Capture Feedback

Bugs, errors, feedback, suggestions, feature requests — if you're lucky, your users will want to share all of these with you. Make sure you provide a simple mechanism for capturing their feedback.

Most apps have a Feedback option under Settings, which opens the mobile email client or a web-based mail form. Ideally a web form should be embedded in the app, rather than launching a mobile browser.

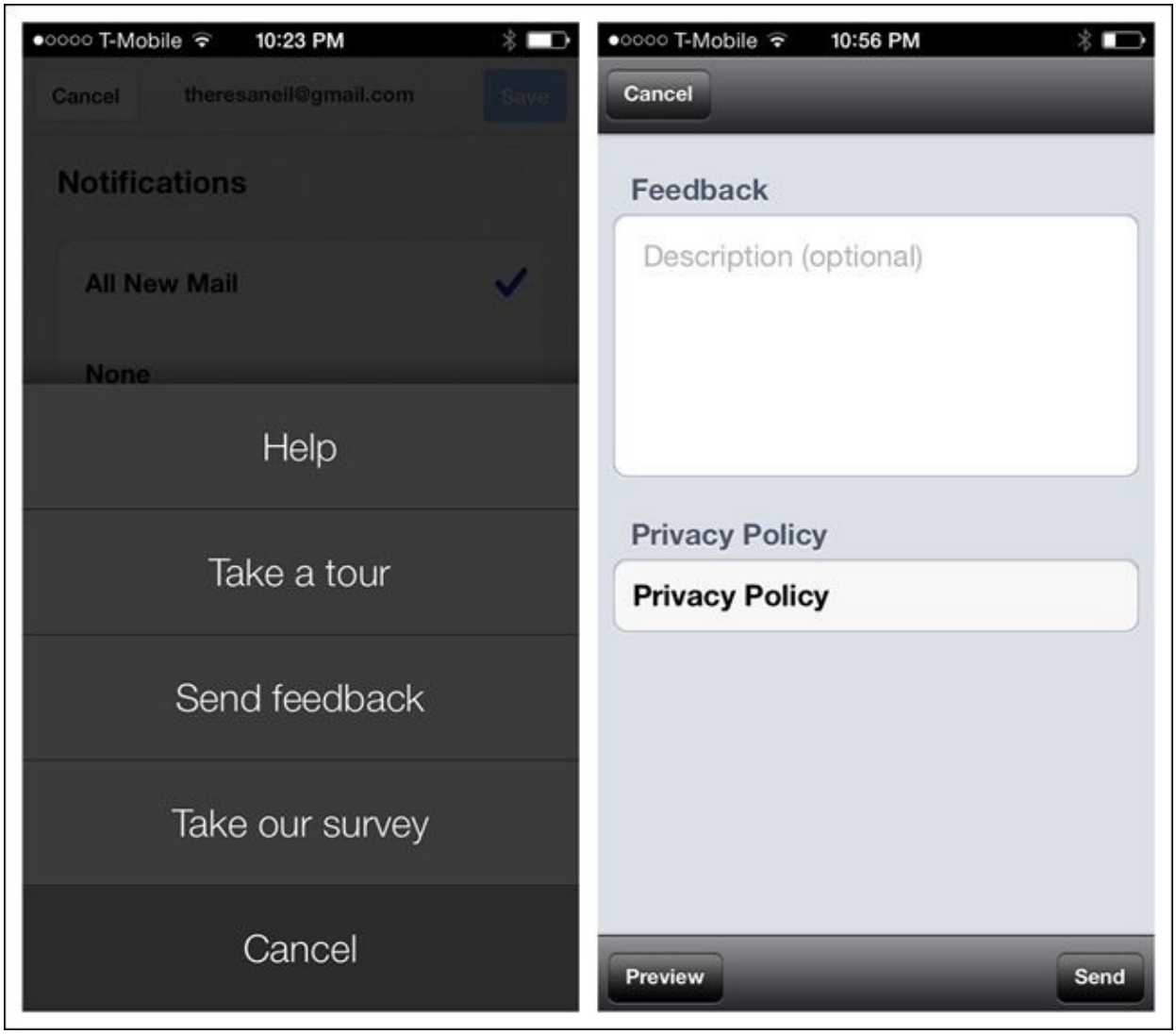



*Figure 10-26. Google Apps for iOS: embedded Feedback Capture form*

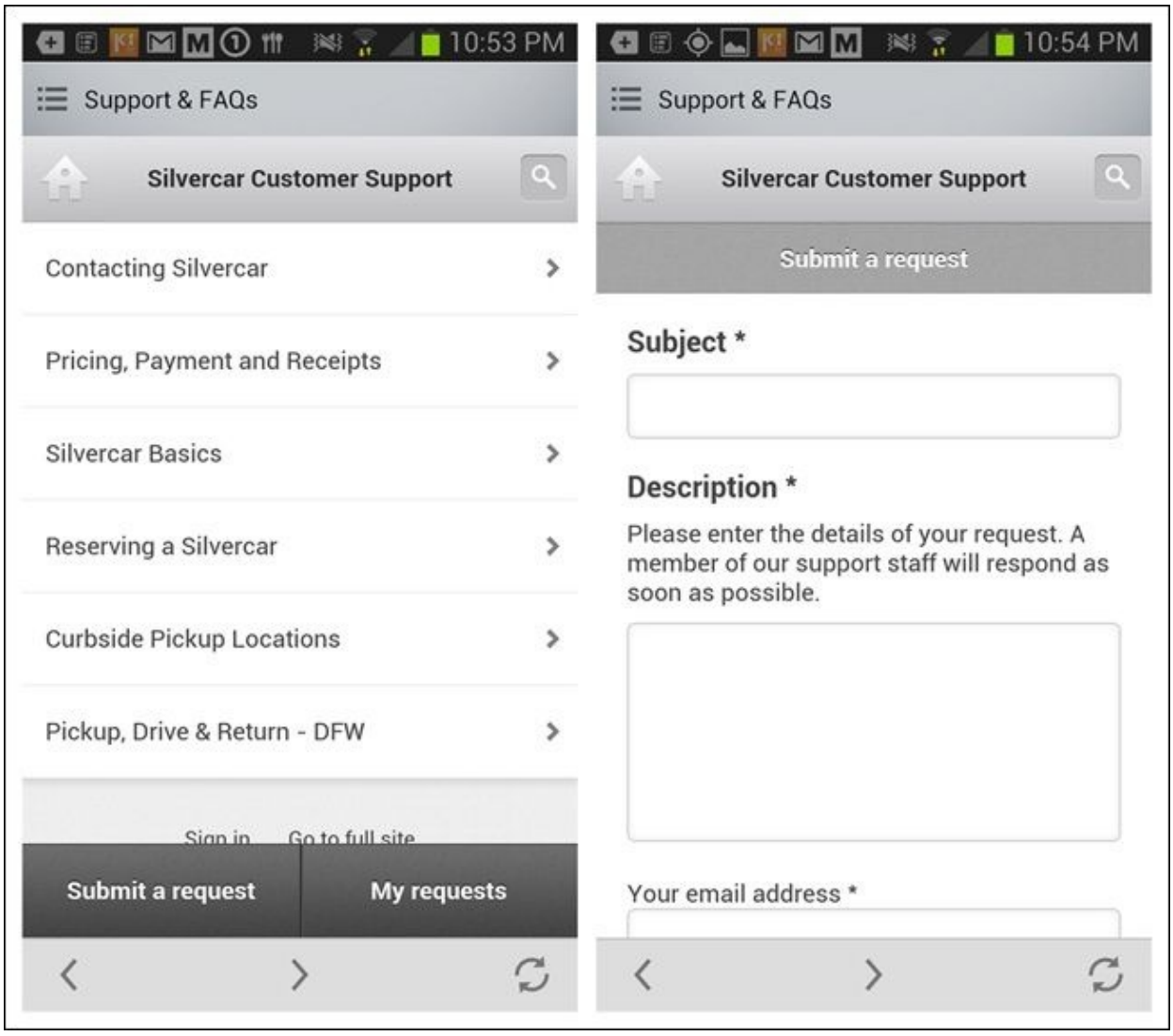



*Figure 10-27. Silvercar for Android: embedded Feedback Capture form (Zendesk)*

Wunderlist has incorporated UserVoice, a popular web-based tool, to build an active community of users who suggest and vote on feature requests. Get Satisfaction is another web-based user feedback tool that can be integrated into native applications.

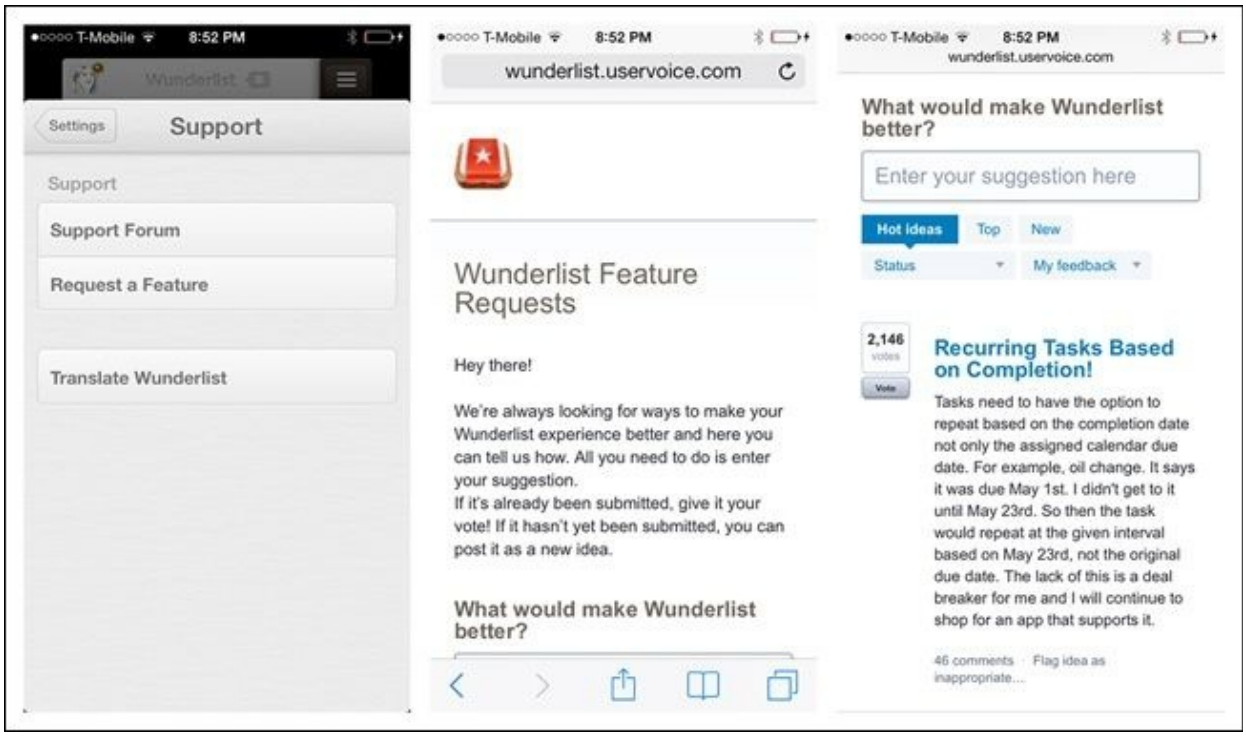



*Figure 10-28. Wunderlist for iOS integrates UserVoice to use Feedback Capture as a way to build community*

**NOTE**

If you ask users for feedback, be prepared to get it. Have a plan in place beforehand that will help you prioritize and act on user issues and feature requests.

# Chapter 11. Anti-Patterns

*Anti-Patterns*

> Novel Notions, Needless Complexity, Metaphor Mismatch, Idiot Boxes, Chart Junk, Oceans of Buttons

What are anti-patterns? Wikipedia says:

> Anti-patterns, also called pitfalls, are classes of commonly reinvented bad solutions to problems. They are studied as a category so they can be avoided in the future, and so instances of them may be recognized when investigating non-working systems. The term originates in computer science, apparently inspired by the Gang of Four's book *Design Patterns*, which displayed examples of high-quality programming methods.

Like their computer science counterparts, the following design anti-patterns are common pitfalls to avoid. Since each reinvention adds its own wretched novelty, I can't provide generic anti-pattern diagrams, so let's just dive right in.

# Novel Notions

New designers try out novel design ideas to be edgy, creative, and innovative. But most of the time the designs are just bad, hard to understand and harder to use. And sometimes anti-patterns result from laziness or habit. According to usability gadfly Richard Gunther, “You can usually tell when a mobile app development team comes from an old web development background. They often attempt to translate old user interaction models to the new platform and assert their ‘creativity’ by introducing nonstandard UI elements.”

Novel Notions can be found anywhere in an application, from primary navigation down to an individual control, interaction, or gesture. NBC News is at the top of my list for Novel Notions with its peacock pie chart menu. Thank goodness most applications don’t try to create menus that match the shape of their logos.


*Figure 11-1. NBC News for iOS: navigation inspired by the company logo*

So why exactly is this an anti-pattern? Let us count the ways:

- The menu location is nonstandard, for mobile in general and iOS specifically. The navicon for a Toggle Menu, for instance, should be placed in the corner of the navigation bar, not in the center.
- Rotated text is hard to read. A simple left-aligned list of categories would take less time to process.

- Too many colors! The overstuffed color palette is confusing and degrades the legibility of the menu options.
- The design fails at the basic function of navigation, which is to show users where they are and where they can go. Neither the selected topic titles nor the show titles persist on the screen once the menu is closed.

Now this isn't as exciting, but it's much easier to use: USA Today has designed a colorful yet functional navigation system using the Side Drawer pattern. Each category of news is represented by a different color, but they aren't shown all at once, thereby avoiding the exploding peacock effect.

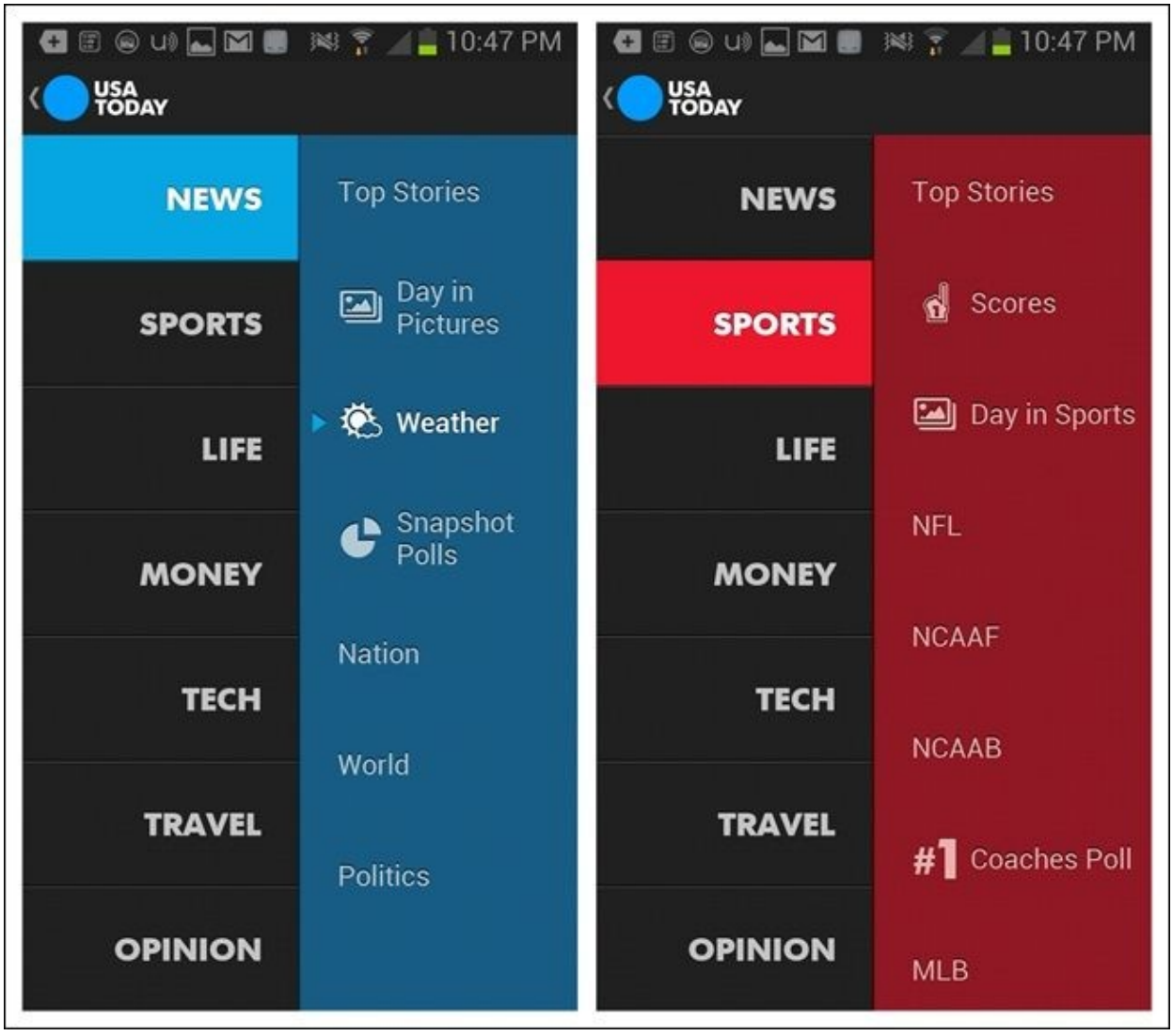



*Figure 11-2. USA Today for Android: color used judiciously in the menu*

Alaska Airlines has also gone off the grid with its menu design. It features a drawer that can be pulled down from the top of the screen. I eventually realized

that this is supposed to mimic the shade on airplane windows! This is a perfect example of taking a skeuomorphic design (i.e., a real-life metaphor) too far. Does it help the user? No, it's cuteness for cuteness's sake.

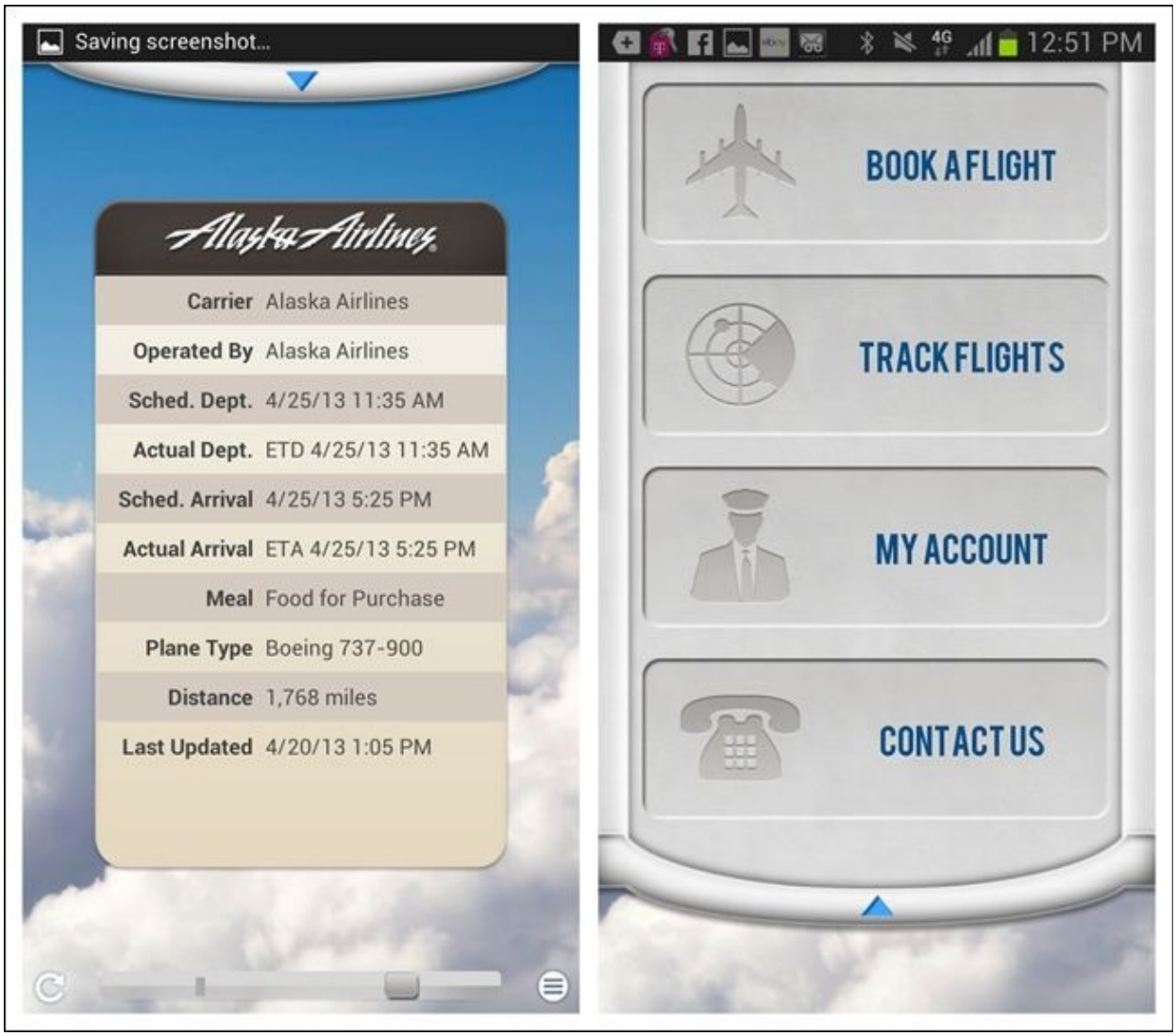



*Figure 11-3. Alaska Airlines for Android: window shade menu doesn't help the user*

But the window shade menu isn't the only Novel Notion. Flight search results are displayed as floating cards. A scrollbar well below the cards lets you navigate through them — assuming you see it. Dragging its handle reveals additional information about each option, but its placement doesn't seem to correlate with the cards.

For a much simpler and faster booking experience, refer to Expedia. It has standard navigation, a succinct search form, and an innovative (and usable) design for reviewing and selecting flights.


*Figure 11-4. Alaska Airlines for Android: unusual Search Results view; navigate through cards with bottom scrollbar*

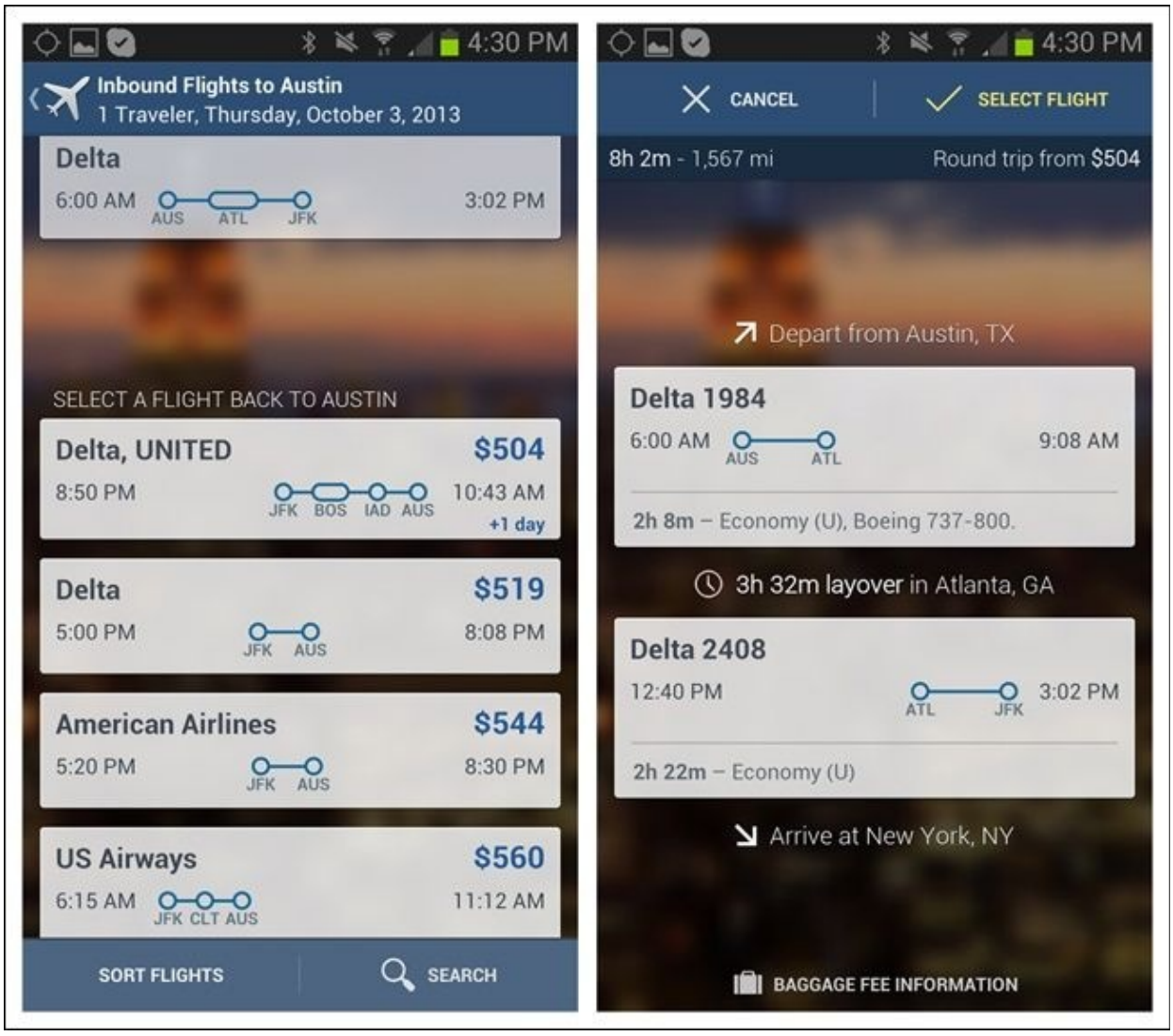



*Figure 11-5. Expedia for Android: nice visualization for the flight legs and connections*

For another innovative approach to flight booking, take a look at Hipmunk. Notice that the innovation is at the UX level (providing “agony” as a search filter), not at the UI control level (a window shade menu doesn’t improve my booking or travel experience in the slightest).

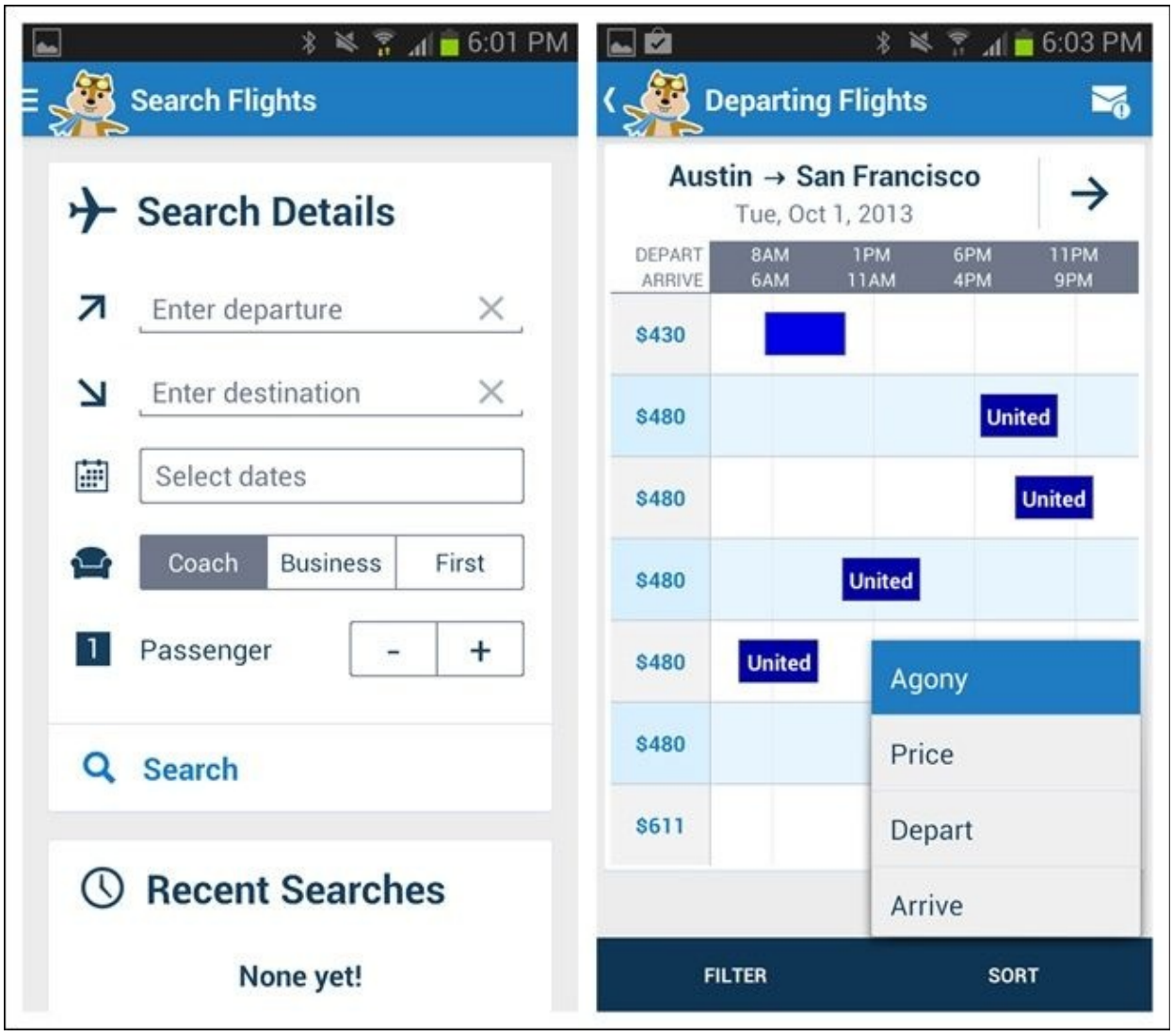



*Figure 11-6. Hipmunk for Android: innovative experience (filter by agony level), but familiar controls*

Here's one more example from Twitter @music. The designers combined a couple of navigation devices to create an "innovative" interface. In this context, I am using the term *innovative* sarcastically to mean they spent extra time and money to create custom controls that decrease usability.

The navigation bar includes a custom spinner, and the footer has page indicators (the four dots). Surprisingly, the spinner and the paging indicators are tied to each other, so selecting an option in the spinner will advance one "page." But wait — there's more! Swiping left or right will update the spinner to show the newly selected module.

Do you think they knew this "spinner plus paging indictor" control was a bad idea? At least that would explain the ultra-low-contrast, black-on-black

navigation bar design — maybe they were trying to hide it!

> **NOTE**
>
> If you are looking for a way to innovate with your application, look for opportunities to help the user by improving the overall UX conceptual model and flows. Understand the OS design guidelines and get familiar with existing design patterns before designing the UI.


*Figure 11-7. Twitter @music for iOS: Android-style spinner + paging indicator = overkill*

For true innovation in music apps, take a look at Songza and Soundwave. Songza's concierge metaphor is brilliant, and Soundwave lets you explore a Music Map that combines social and location elements.

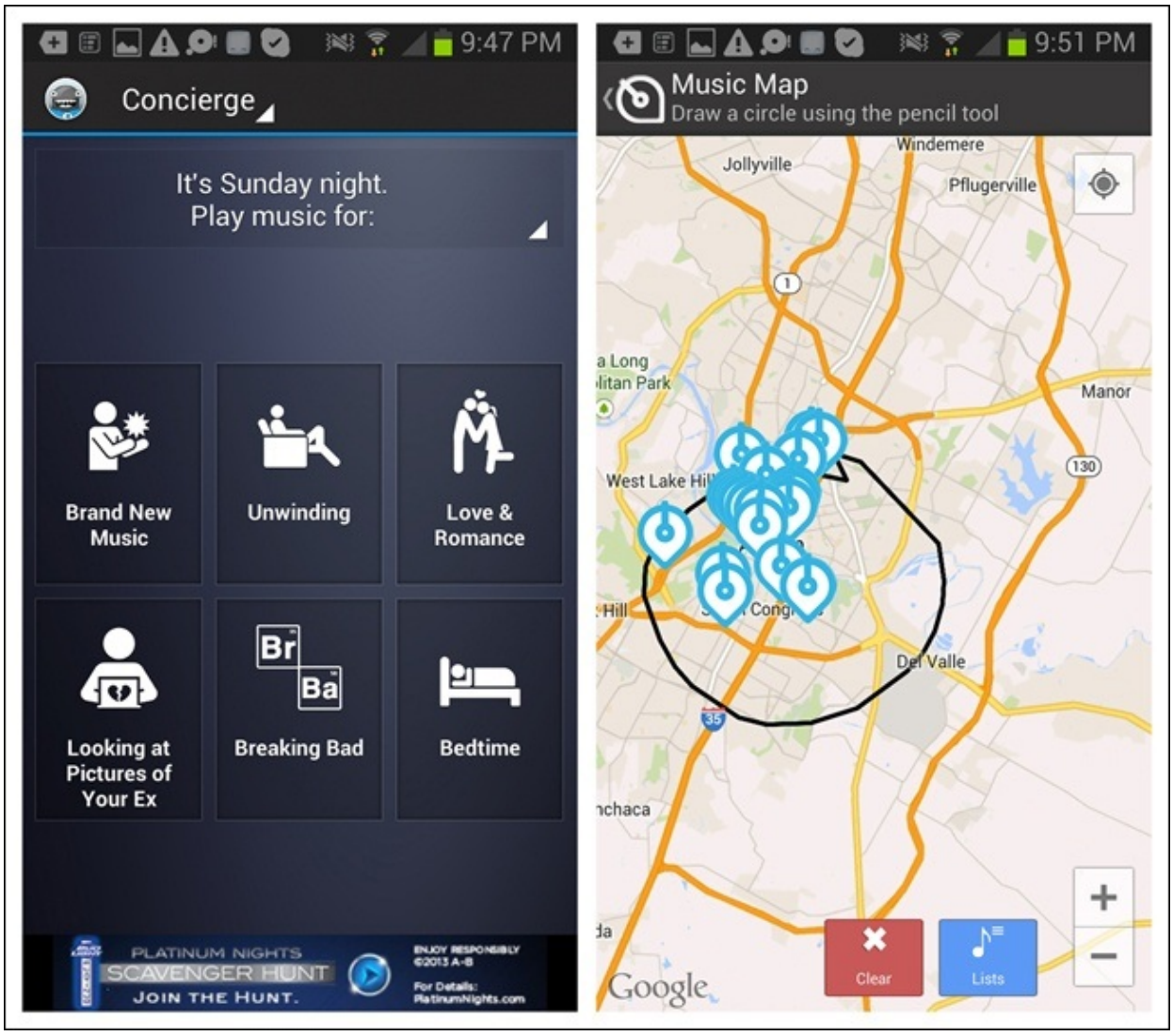



*Figure 11-8. Songza and Soundwave for Android: innovations without anti-patterns*

## Needless Complexity

One of the differentiators between a great user experience and a poor one is efficiency. User experience coach Whitney Hess says, “Efficiency allows for productivity and reduced effort, and a streamlined design allows more to get done in the same amount of time. Creating efficiency demonstrates a great deal of respect for your customers, and they’ll be sure to notice.”

Let’s look at some apps that have needlessly complex interfaces that hamper the user’s efficiency.

Eureka Offers is a pretty simple application for finding local and national deals. Users can save the deals they like. But unlike Google Offers, Shopular, SnipSnap, RetailMeNot, and most other apps of this type, which use a simple Inline Action for saving, snipping, and favoriting deals, Eureka requires a long-press, drag-and-drop interaction to save. Not only does this require more user effort, but it is also less discoverable than an Inline Action (see Chapter 5).

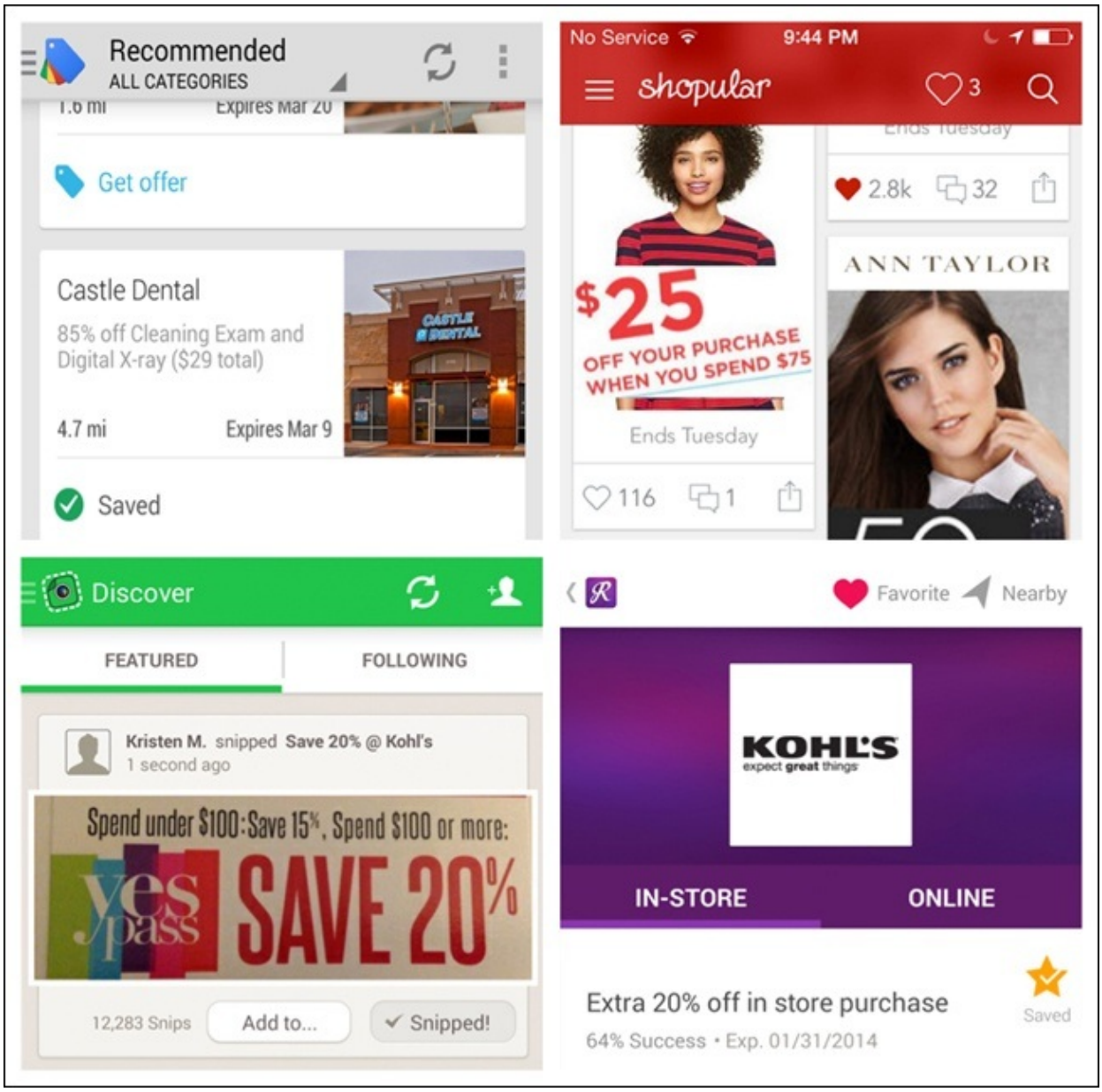



*Figure 11-9. Google Offers, Shopular, SnipSnap, and RetailMeNot all offer Inline Actions for saving, snipping, and favoriting*

Another example of introducing drag and drop without considering the extra effort it requires of users is Facebook Chat Heads. How about a “long press to see/hide” option, or swipe off the screen to remove, or just maybe not have floating heads on the screen?

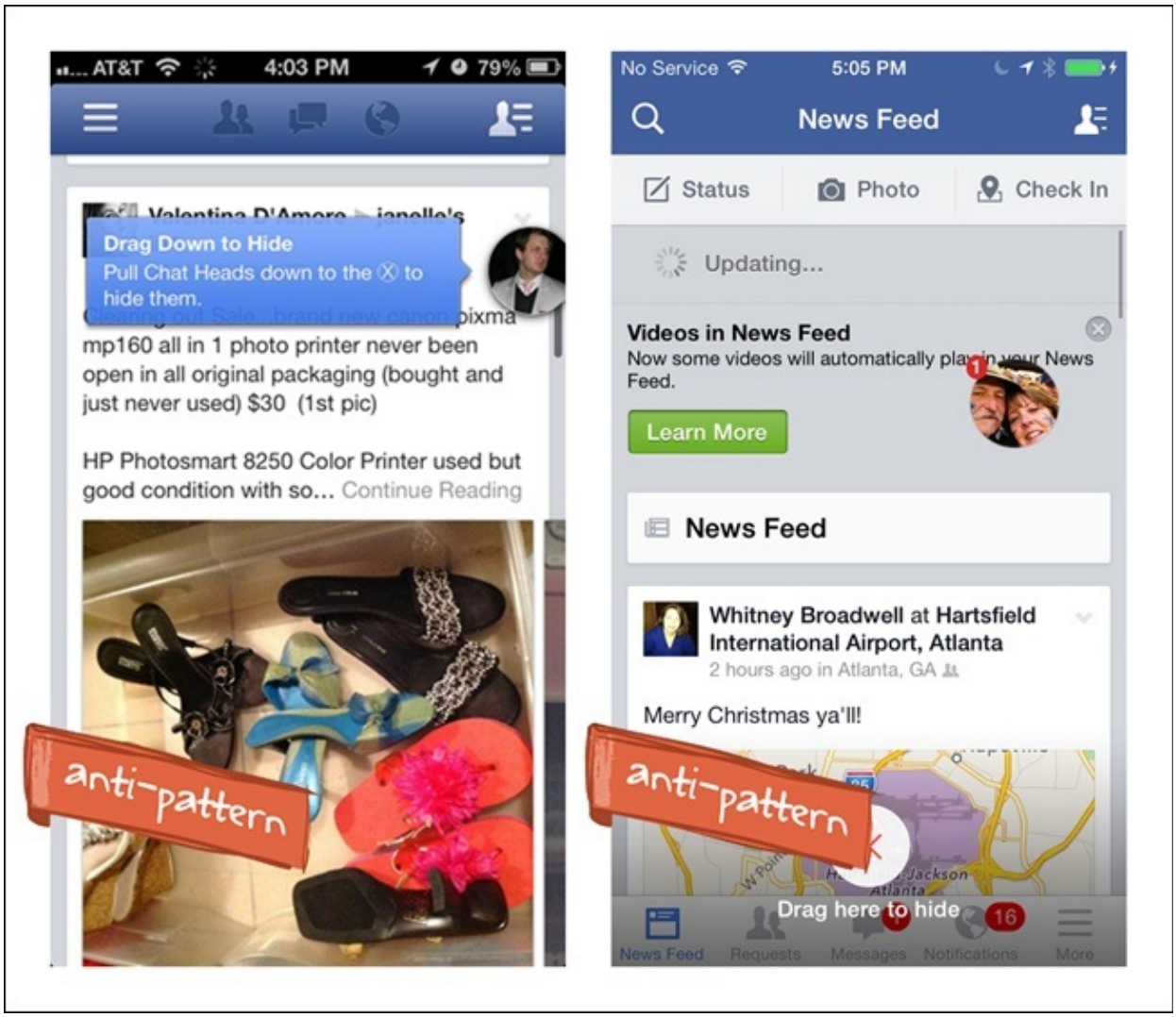



*Figure 11-10. Facebook for iOS: Chat Heads are nonintuitive and require extra effort to remove from the view*

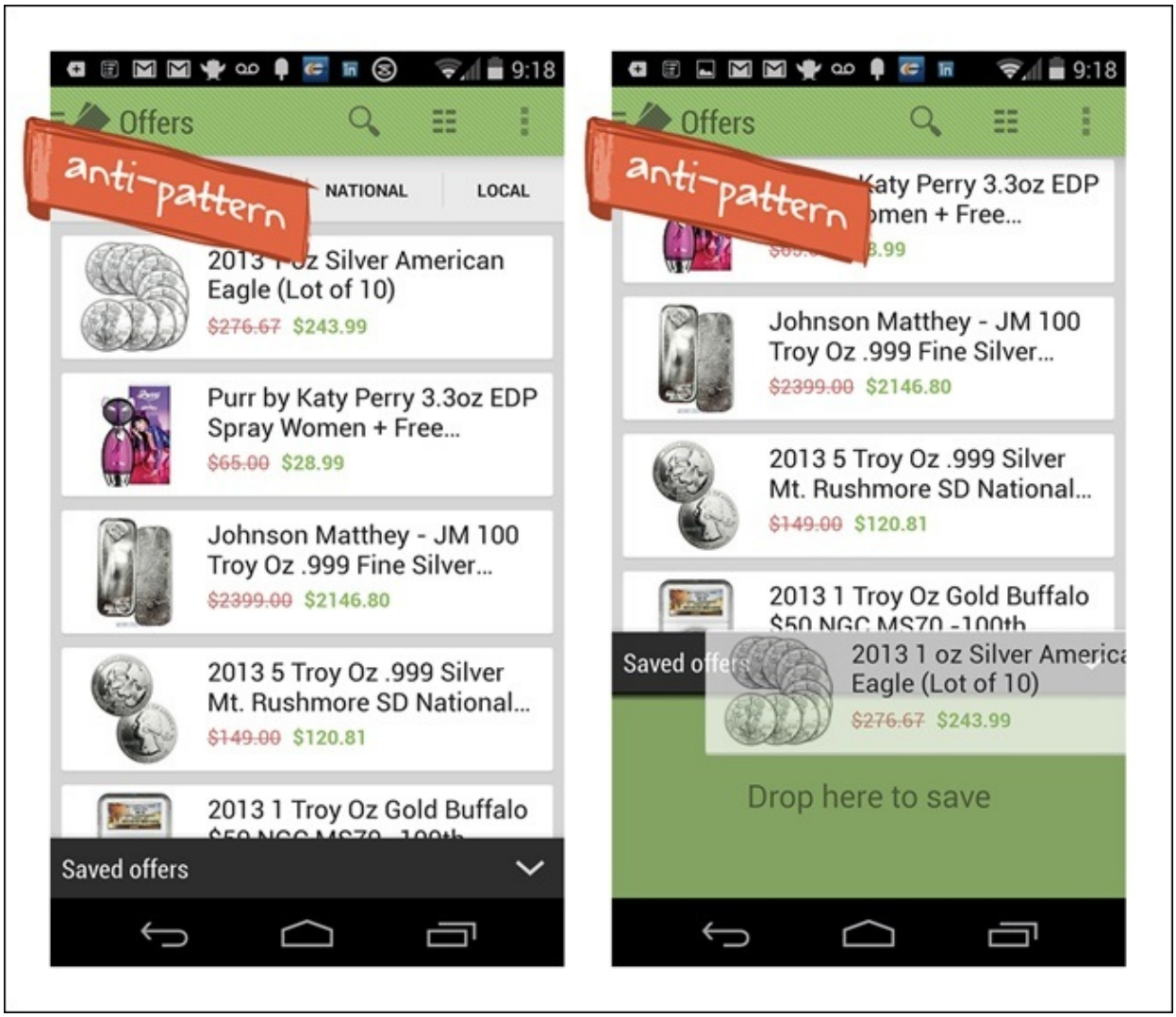



*Figure 11-11. Eureka Offers for Android requires a long-press drag-and-drop to save*

iTranslate's designers appear to have fallen in love with the novelty of gesture-based UIs. I say that because the app is overloaded with nonintuitive gestures. Short-swipe right to paste from clipboard, long-swipe right to use voice input — huh?

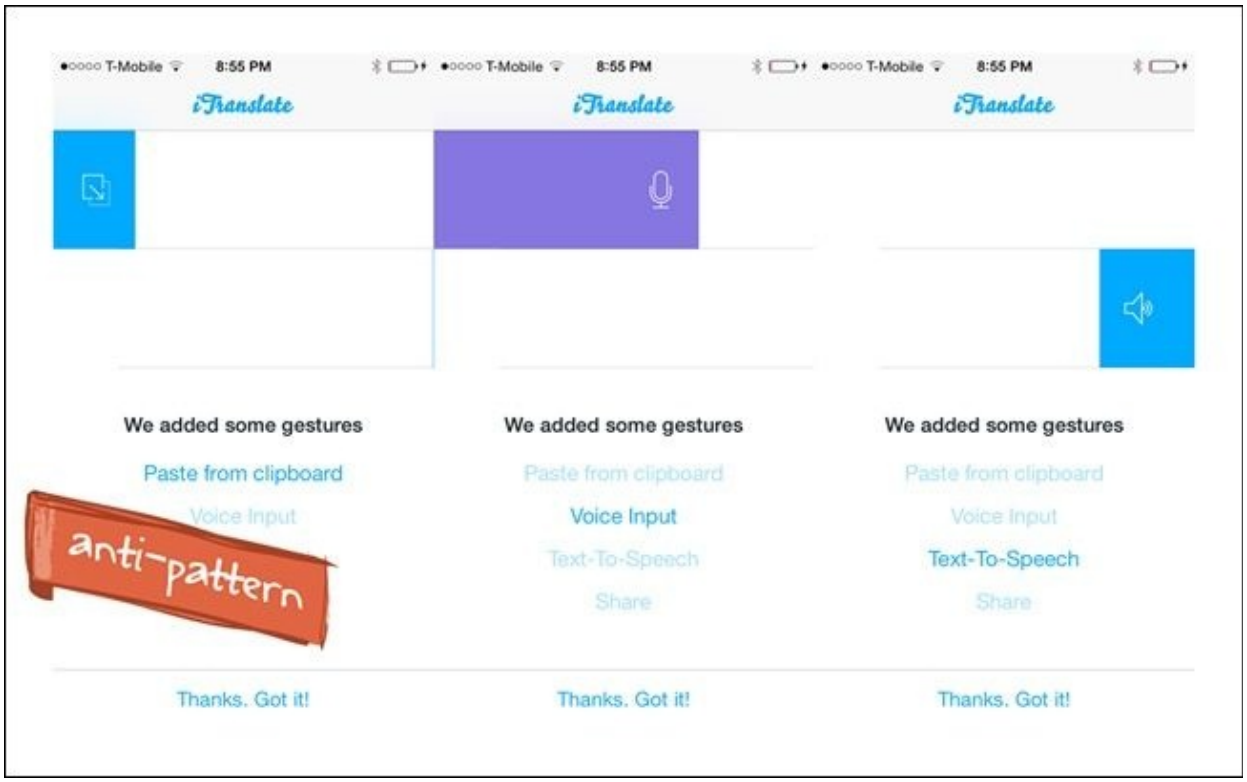



*Figure 11-12. iTranslate for iOS: UI is overloaded with nonintuitive gestures*

To let users choose the input mode, the designers could have offered something simple like the “Scan It, Snap It, Say It” picker in Amazon Price Check. In fact, an iTranslate competitor, Google Translate, serves a “Snap It, Say It, Write It” contextual toolbar under the text entry field.

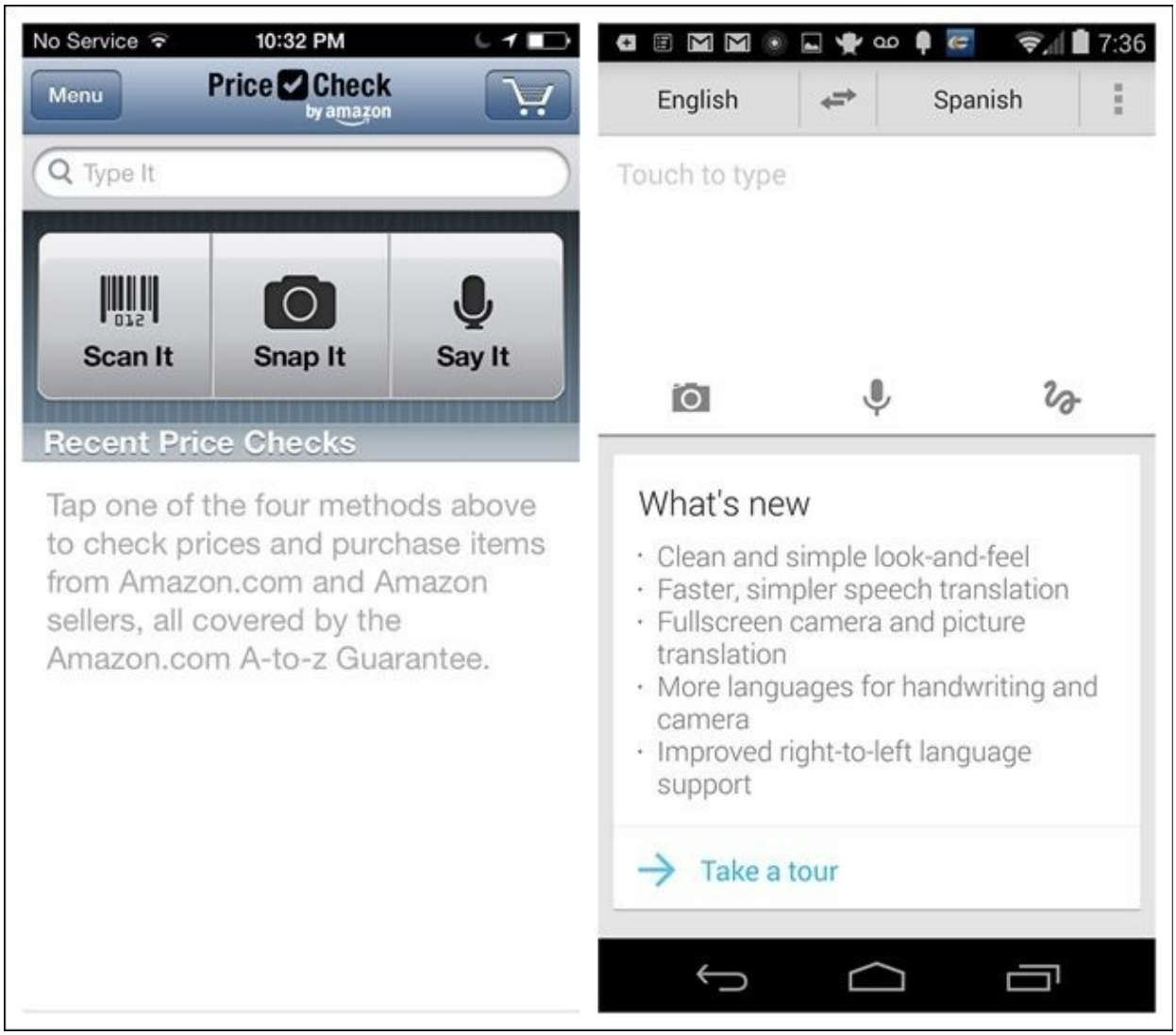



*Figure 11-13. Amazon Price Check for iOS and Google Translate for Android: intuitive input mode choosers*

In the next example, from Over, the designers really tried to make a pie menu work. From the home screen, the menu can be swiped in from the right. This concept has possibilities, but not the way it is designed currently. The pie menu options are a combination of primary navigation and actions, and they are *always changing*. It is truly disconcerting and probably the least efficient photo-editing tool I've ever tried.

Always ask yourself, "Could I make this simpler?" If the answer is yes, do it — even (or maybe especially) if it means sacrificing novelty. Your users will appreciate it.

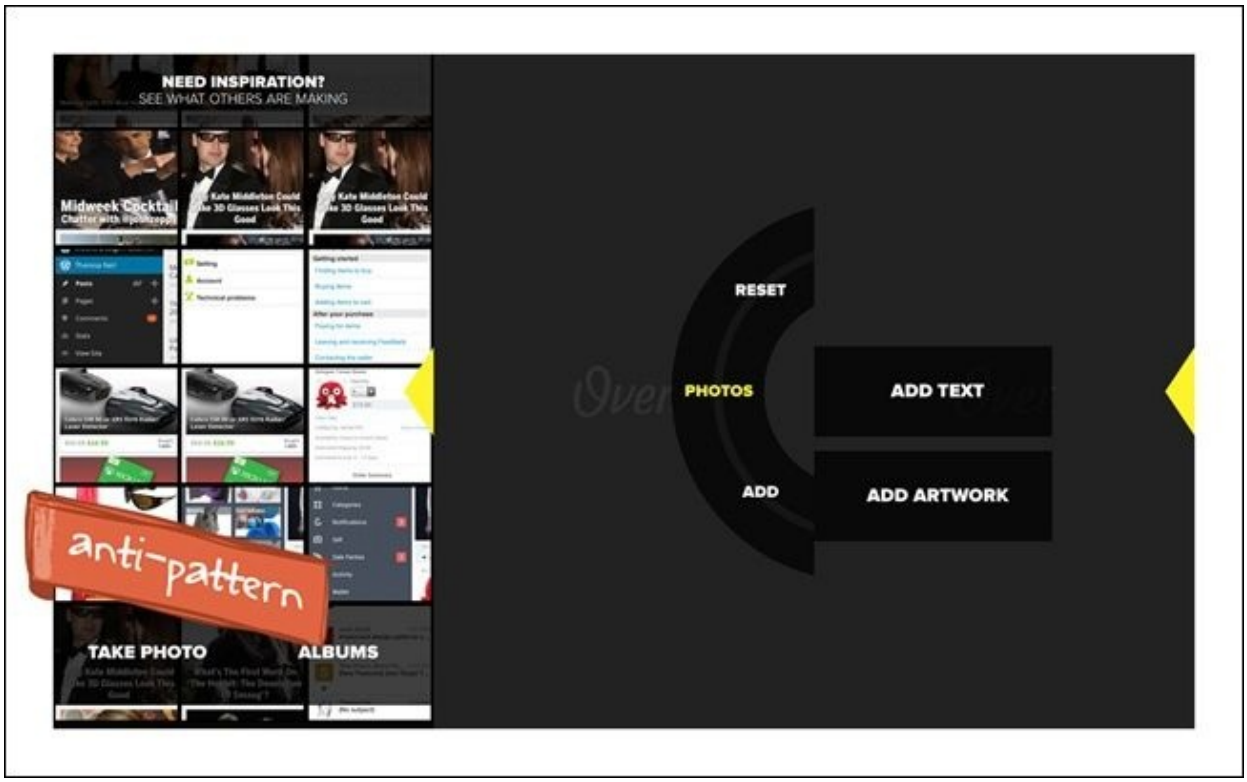



*Figure 11-14. Over for iOS: an unpleasant adventure to navigate (http://youtu.be/rMG0ftTdSy4)*

# Metaphor Mismatch

This anti-pattern results from picking the wrong metaphor for the interface. Metaphor Mismatch can occur at a low level, when a control or icon is used inappropriately, or at a high level, where the conceptual model for the application doesn't match the user's mental model.

## Control Mismatch

Conqu and DailyBurn provide examples of low-level Control Mismatches. For reasons unknown, the designers of both have chosen to create custom controls instead of using the default Android controls. Conqu uses an iPad popover control, and DailyBurn uses an iOS-style selector.

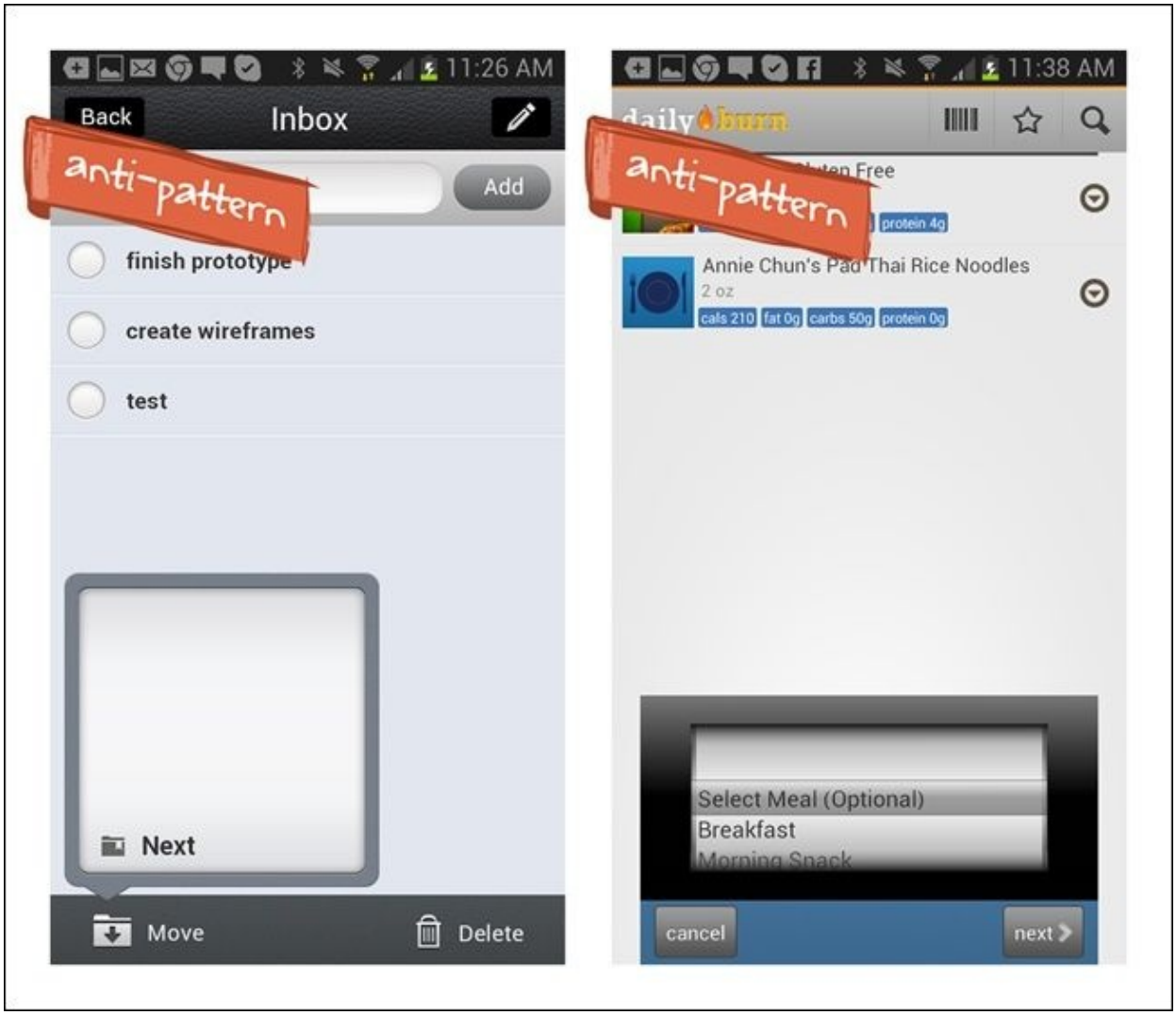



*Figure 11-15. Conqu and DailyBurn for Android: unexpected iOS-style controls in Android apps*

The Edmunds app for iOS and Android has nonstandard controls for primary navigation. If users can't easily see the main modules in the application, it can hurt app adoption.

The home screen for the Edmunds app is almost completely empty by default. It took me a while to realize it was a search form. It is also easy to overlook the other modules in the application, because they are tucked under the spinner in the Android app, and under an arrow in the iOS app.

An easy fix would be to switch to tabs for navigation (see Chapter 1). And the Implicit Search pattern might work well for home screen vehicle searches — better than a blank home screen, anyway. (See the Implicit Search examples for Zillow and Trulia in Chapter 4.)

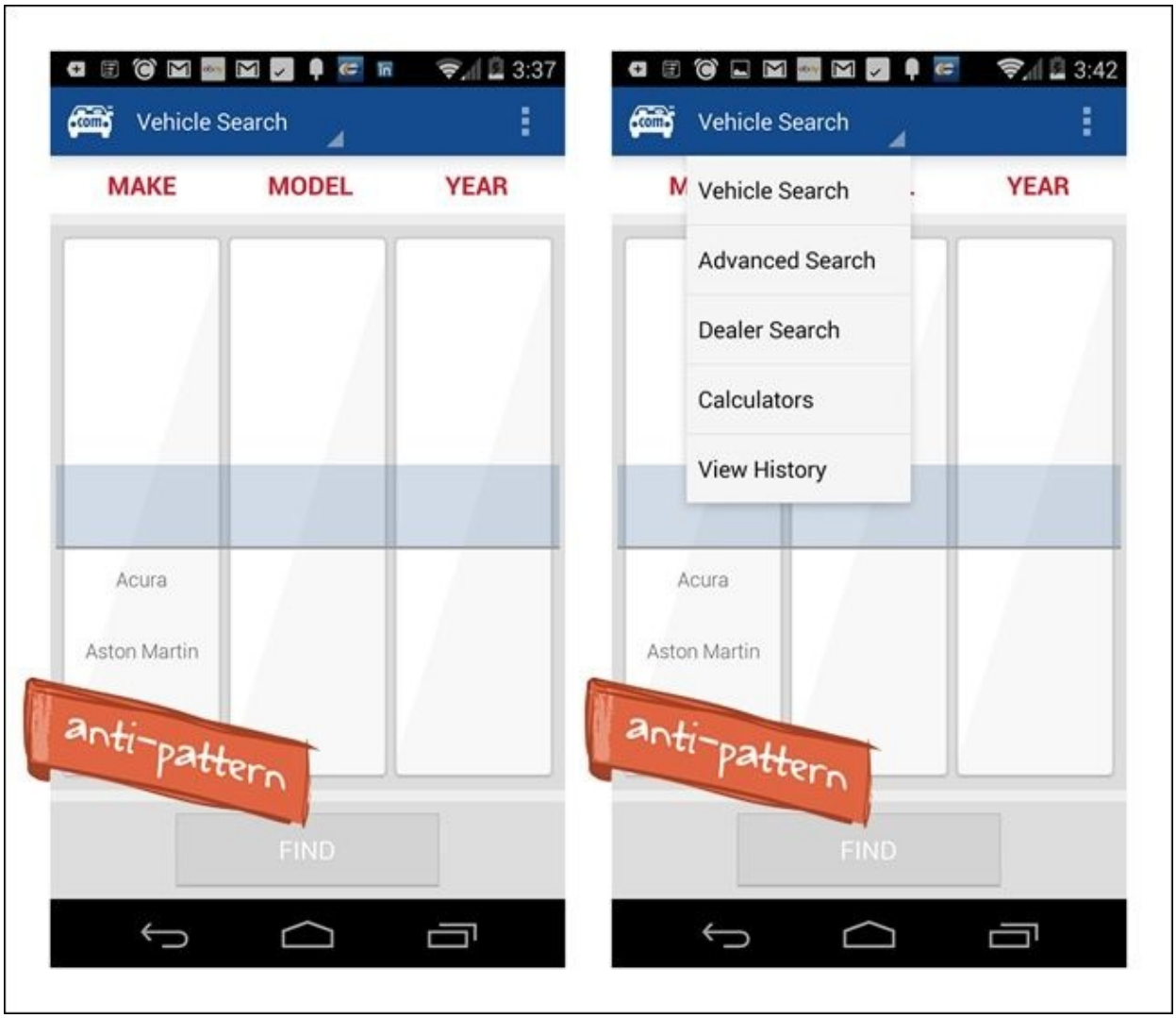



*Figure 11-16. Edmunds for Android: home screen, and spinner used incorrectly for primary navigation*

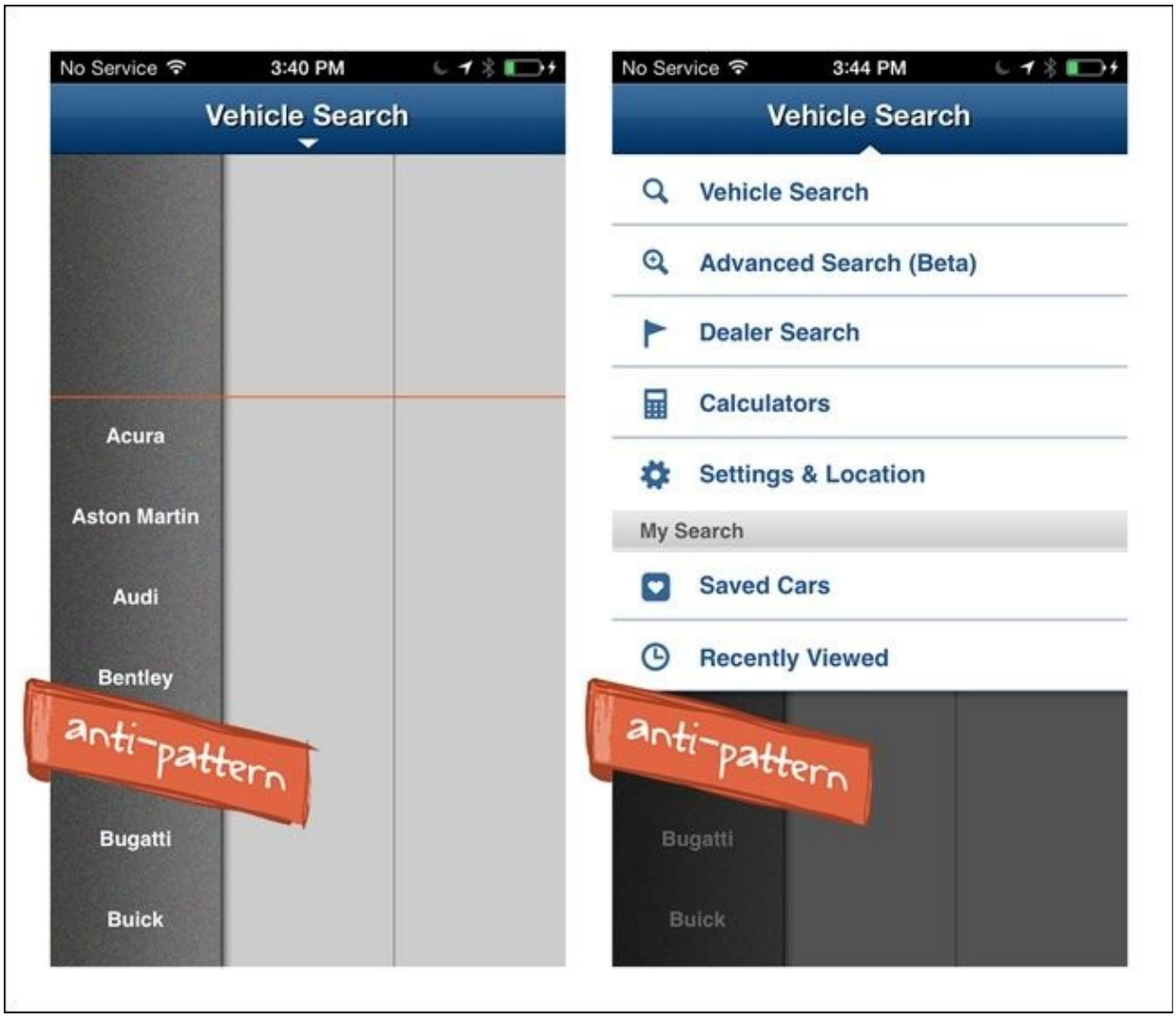



*Figure 11-17. Edmunds for iOS: blank home screen is really a search form with a nonstandard menu*

## Icon Mismatch

Users have specific expectations for common icons: a gear is for settings, a star is for saving or designating favorites, a magnifying glass is for search, and so on. Using a familiar icon for anything other than its familiar purpose will lead to confusion. For example, in Kindle for iOS, I was surprised to finally find the brightness adjustment control under the font properties icon.

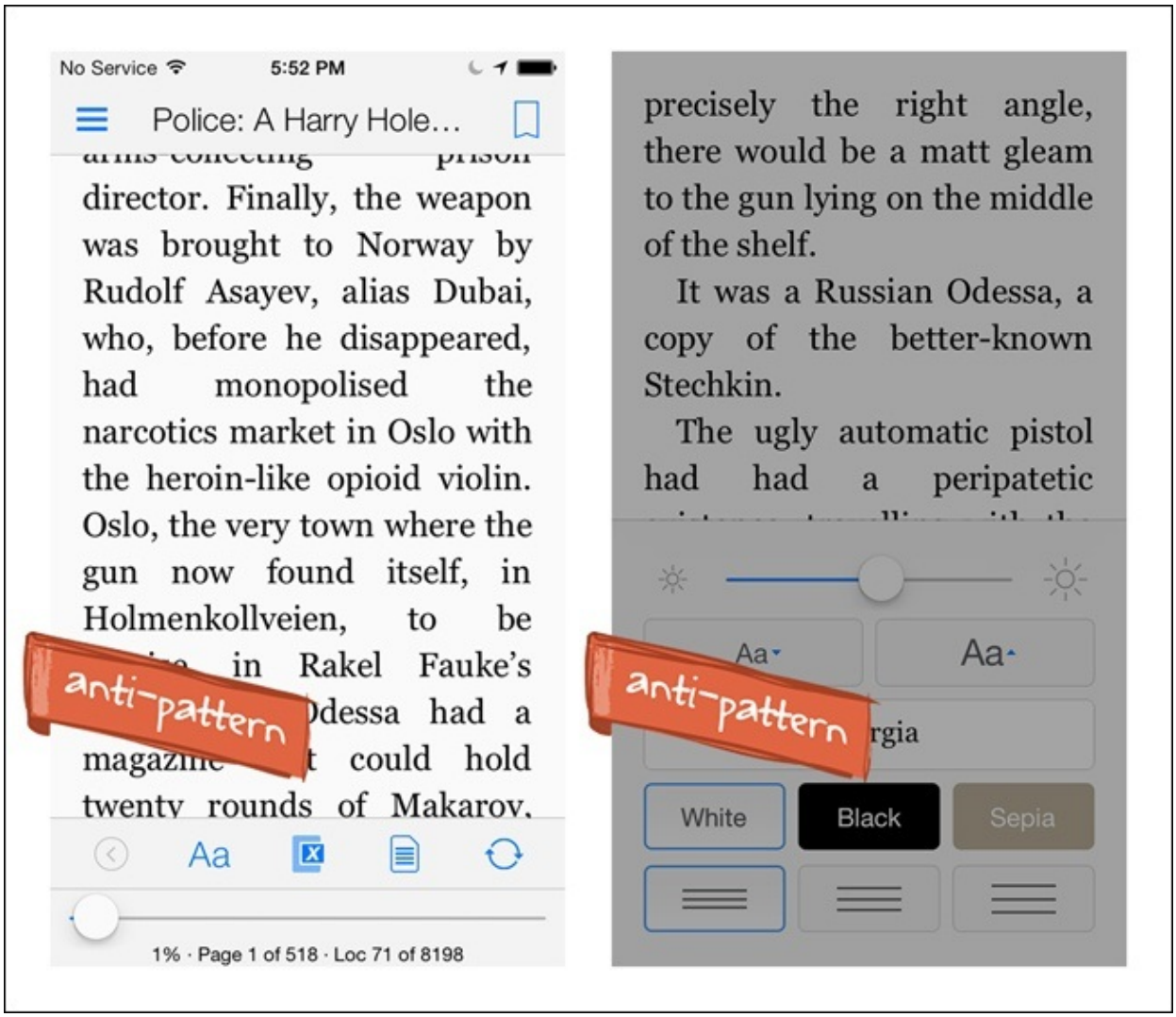



*Figure 11-18. Kindle for iOS hides the brightness control under the font properties icon*

## Gesture Mismatch

*Pull to refresh* is a pretty standard gesture at this point. We see it in productivity tools, social and entertainment apps, and in retail, to name a few categories. Tweetbot 3 and Audible show typical examples of its use.

Seed, an email app, tried to teach us a new trick: pull to add a reminder. Predictably, it didn't work. We all know what pull is for — or think we do — and it isn't for adding a reminder to an email. Guess someone figured that out, since the newest release now has a familiar reminder icon, the clock face, instead.

**NOTE**

Pull to refresh may be disappearing shortly, as most devices today are powerful enough to handle

background updates (source: *http://bit.ly/1fFC1uc*).

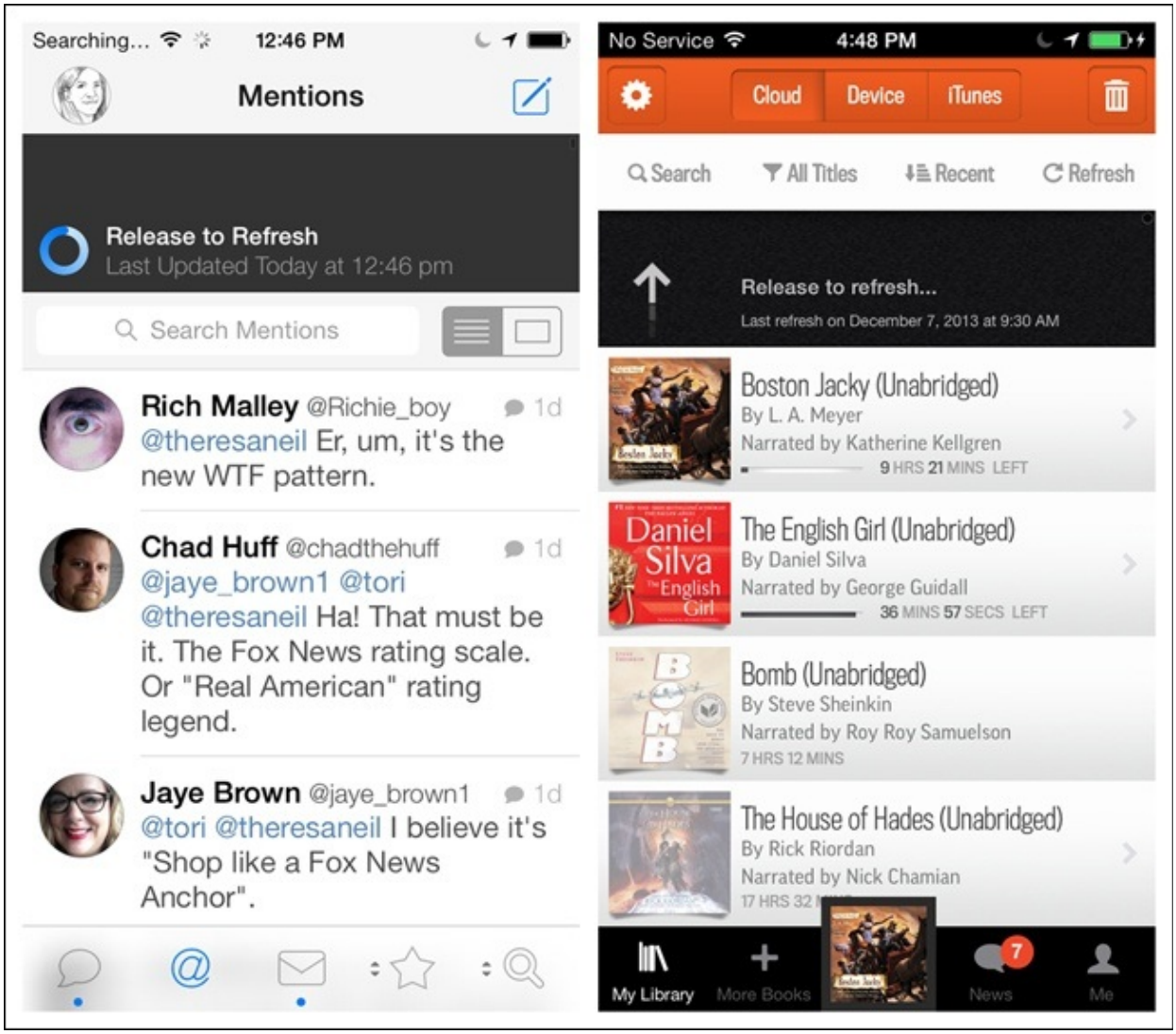



*Figure 11-19. Tweetbot 3 and Audible for iOS: pull to refresh has become a standard gesture*

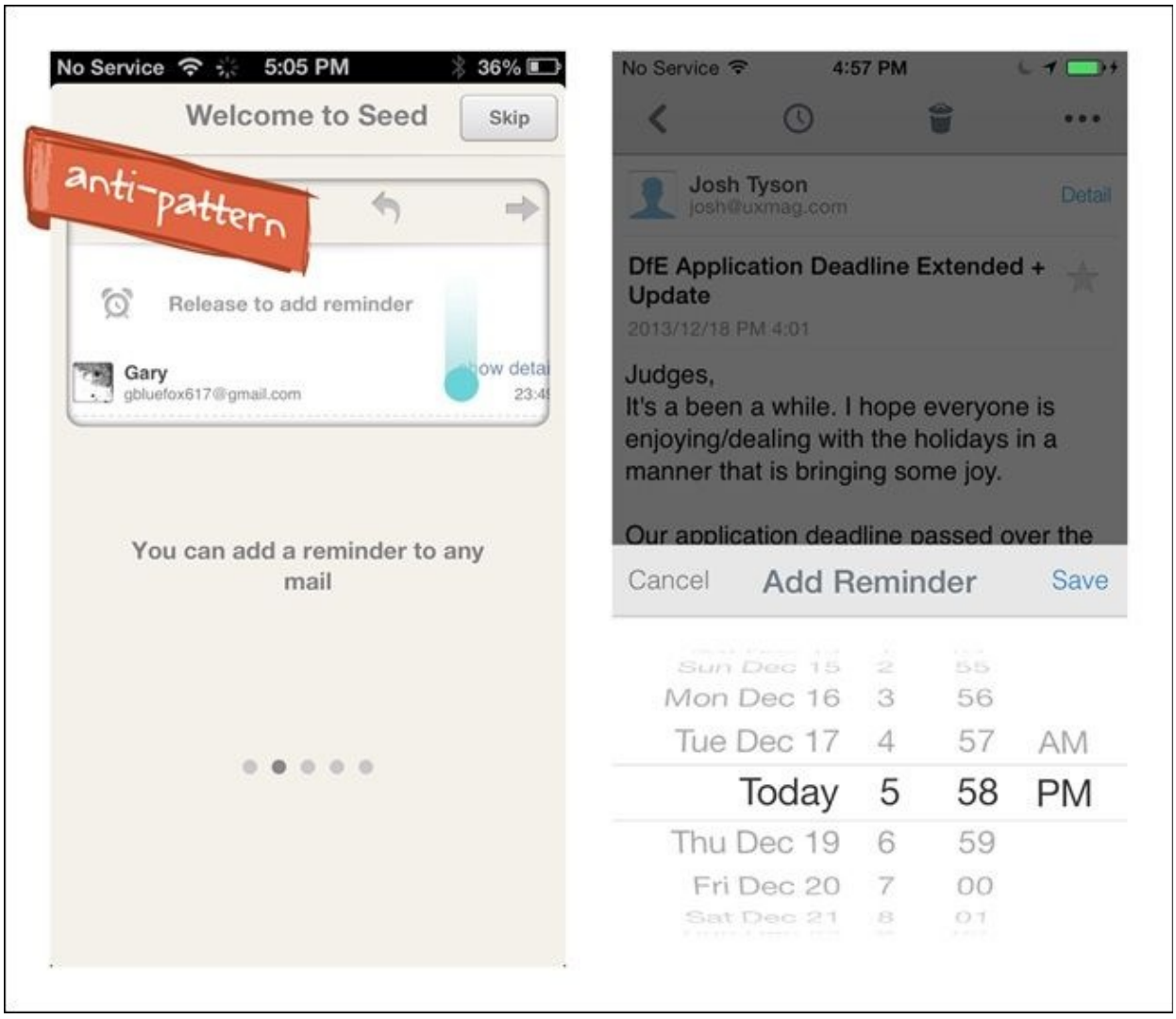



*Figure 11-20. Two versions of Seed for iOS: pull to add reminder (left), dropped from redesign in favor of a reminder icon (right)*

## Mental Model Mismatch

One of my favorite examples of this anti-pattern is a trouble ticket system my previous employer rolled out. When we had a technical issue, whether it was a forgotten password or a dead laptop, we had to sign in to the system, browse the list of issues, select an issue, add it to a shopping cart, and then check out. Great design for a retail shopping app — total mismatch for a technical support application!

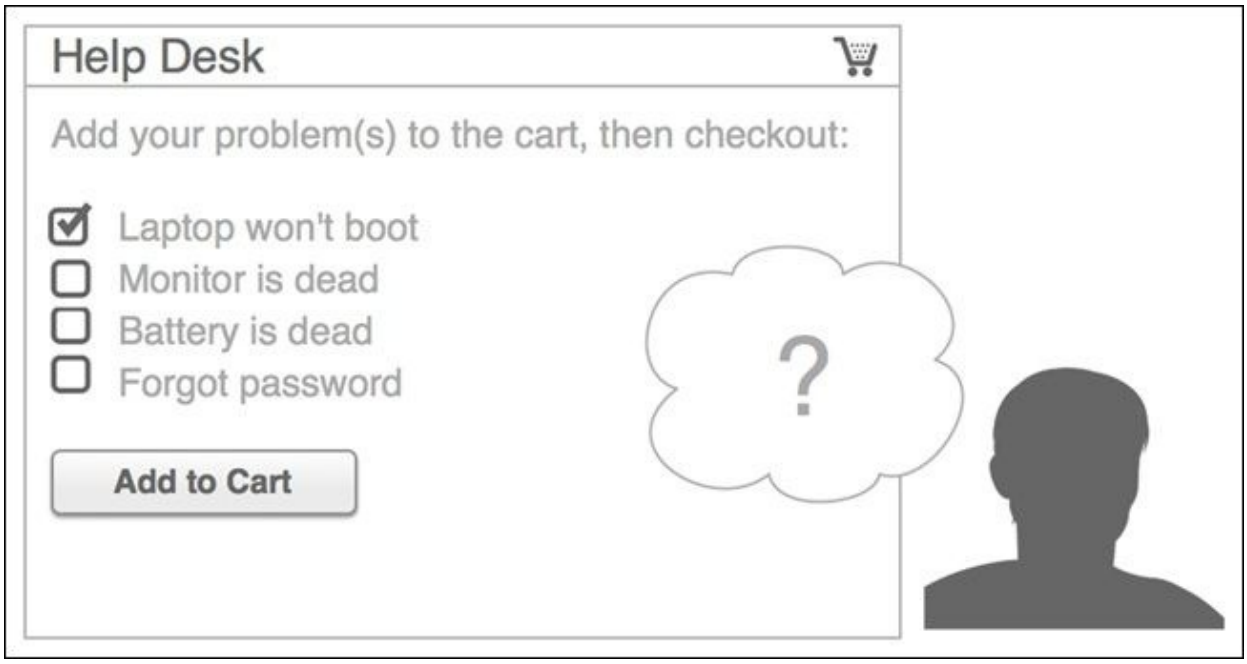



*Figure 11-21. Metaphor Mismatch: treating a request for tech support like a shopping spree*

Another example is from the ABC News iPad app, circa 2011. A globe for navigating news content? I guess that would make sense if stories were surfaced from specific geographic locations — but they weren't. In fact, the ABC News globe had nothing to do with the globe we live on. It was just a spinning sphere of news stories that was hard to read and even harder to browse.

Compare it with the globe in Geo Walk, an educational application for exploring facts from around the globe. Geo Walk has fact cards placed at different locations on the globe, so you can explore facts for a specific region.

**NOTE**

Metaphors, from iconography and controls to conceptual models, are visual shortcuts that can simplify and enhance an experience. But using them improperly will make the application frustrating and difficult to learn.


*Figure 11-22. ABC News for iPad (2011) used a globe for navigating news content; in Geo Walk for iOS, a globe more properly suggests geographic context*

## Idiot Boxes

In *About Face 3: The Essentials of Interaction Design* (Wiley, 2007) (*http://amzn.to/1fFC7SK*), author Alan Cooper says, “A person enters a highly productive mental state by working in harmony with her tools. Interrupting a user’s flow for no good reason is stopping the proceedings with idiocy.”

Bill Scott ran with the “stopping the proceedings with idiocy” concept and christened it the Idiot Box anti-pattern.

I think the very worst experiences I’ve had with Idiot Boxes are with the apps that on first use prompt you to rate the app, then enable location-based services, then allow push notifications, then ask one more time for a rating. Forcing four dialogs in a row on me makes me want to rate the app, all right — a big, fat ZERO!

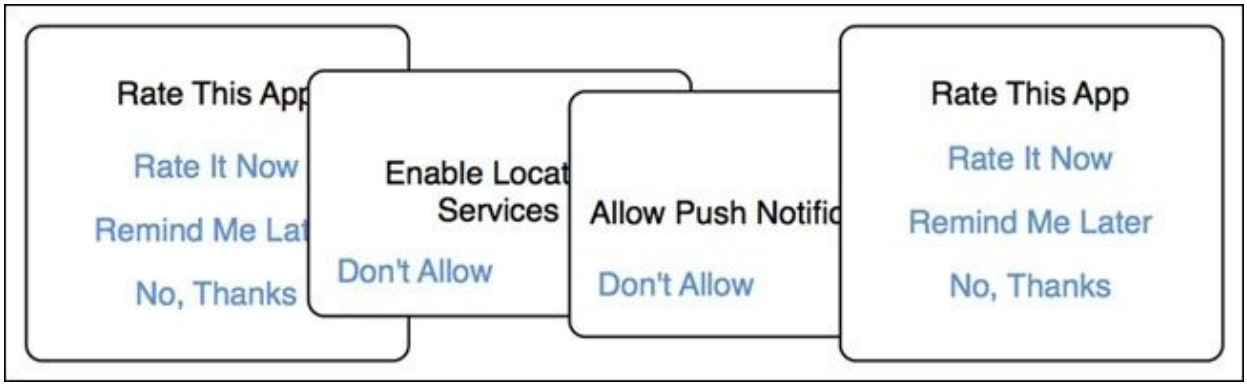



*Figure 11-23. Idiot Boxes: multiple dialogs on first use disrupts the flow before it even gets started*

Even worse are the apps that present you with an Idiot Box prompting you to register the first time you use them. See the upcoming “Chapter Extra” from Greg Nudelman for more on this annoying practice and why it is unnecessary.

Jamie Oliver’s Recipes provides another Idiot Box example. When I tap “Add to shopping list,” a dialog asks, “Do you want to add all of these ingredients to your shopping list?” Well, yeah, that’s why I tapped the button.

The Delete Confirmation dialog in the second screenshot is okay. It is good practice to ask users for confirmation before they perform an irreparable action, like deleting everything in a list or permanently emptying the trash. But there is no need to require explicit confirmation for every action a user takes. For adding to a list, for instance, it’s fine to provide transient feedback that the items were added. (See Chapter 9.)

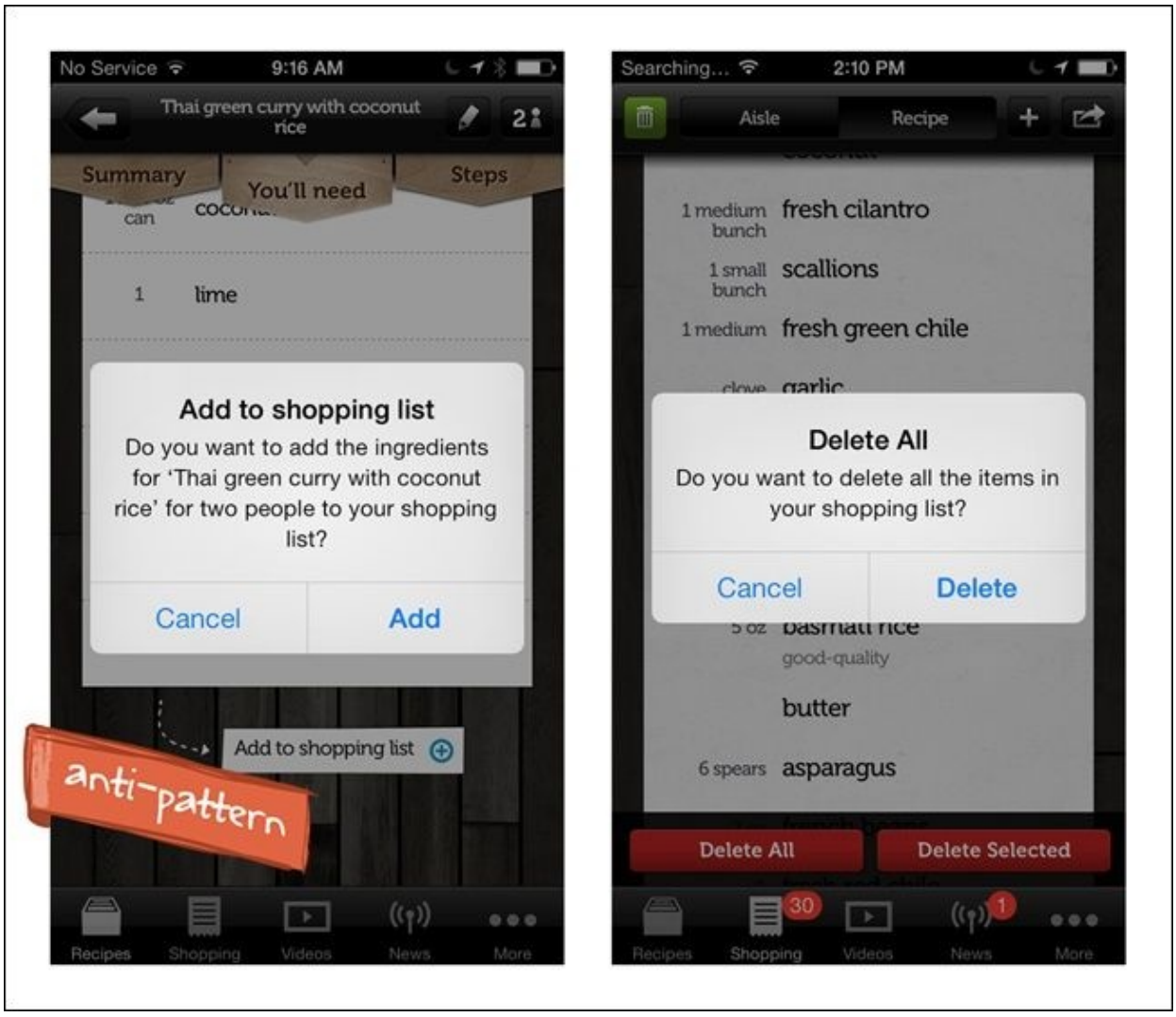



*Figure 11-24. Jamie Oliver's Recipes for iOS: confirming every action is unnecessary*

Chipotle provides another Idiot Box example: I am ready to check out when a Register/Sign In dialog pops up. Why not just take me directly to the Sign In screen? And Zulily can't imagine that I really do want to sign out, so it won't let me without double-checking. Idiotic.

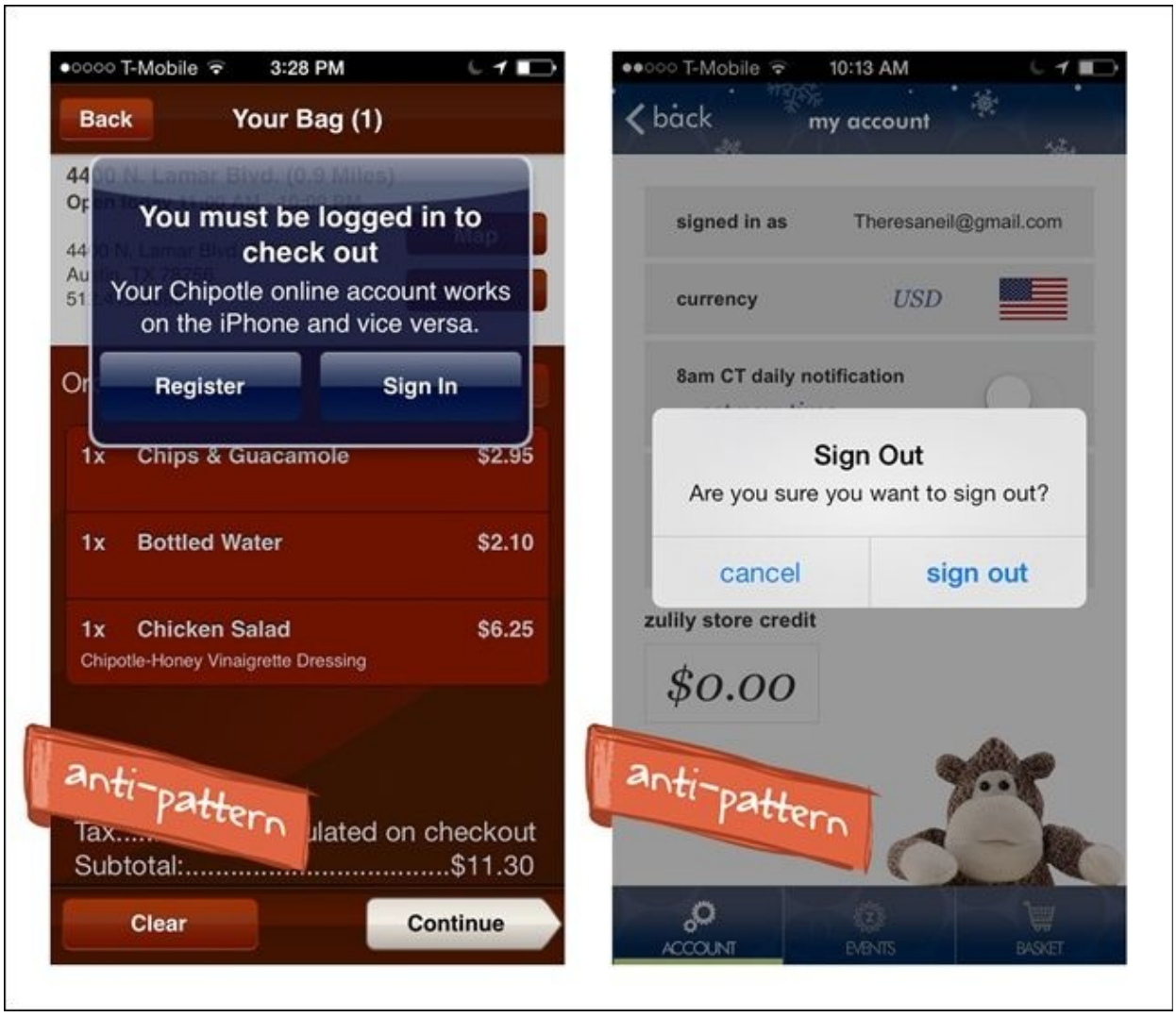



*Figure 11-25. Chipotle and Zulily for iOS: Idiot Boxes doing what they do best, disrupting user flow*

Many retailers rely on Idiot Boxes to give feedback that an item has been added to the shopping cart, like Haute and Rent the Runway. A less disruptive, more effective technique is to simply increase the number indicating how many items are in the cart. You could also show an inline confirmation message or an animation of the item being added to the cart. See Chapter 9 for examples.

**NOTE**

Avoid disrupting the workflow. Only show a dialog that requires user confirmation when an irreparable action is being taken (like a permanent delete). And if you must prompt users to rate your app, at least wait until they've had a chance to use it.

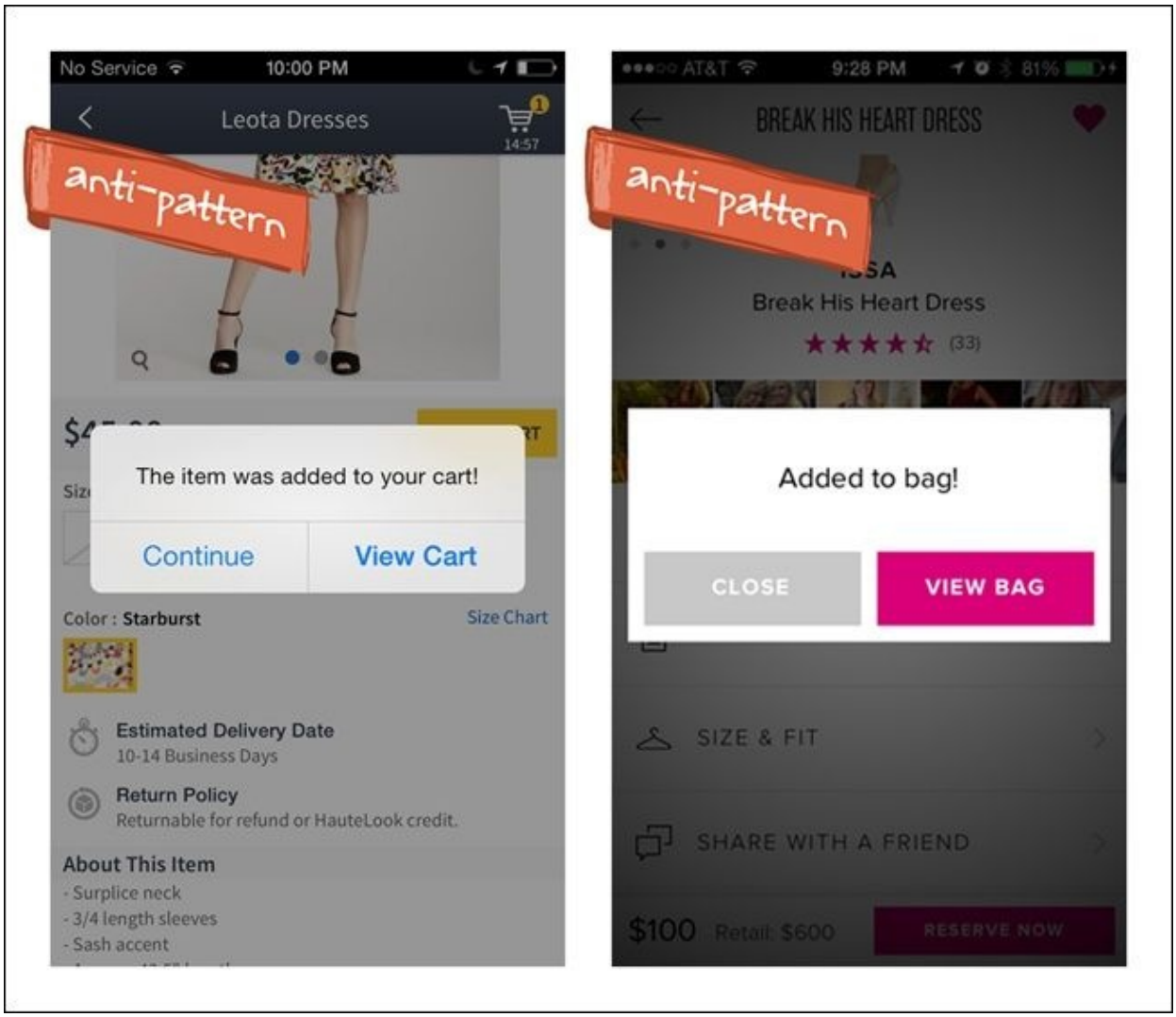



*Figure 11-26. Haute and Rent the Runway for iOS: Idiot Box confirmation for adding items to cart*

# Chart Junk

Edward Tufte coined the term “chart junk” in his 1983 book *The Visual Display of Quantitative Information* (Graphics Press), in reference to charts like the following one.

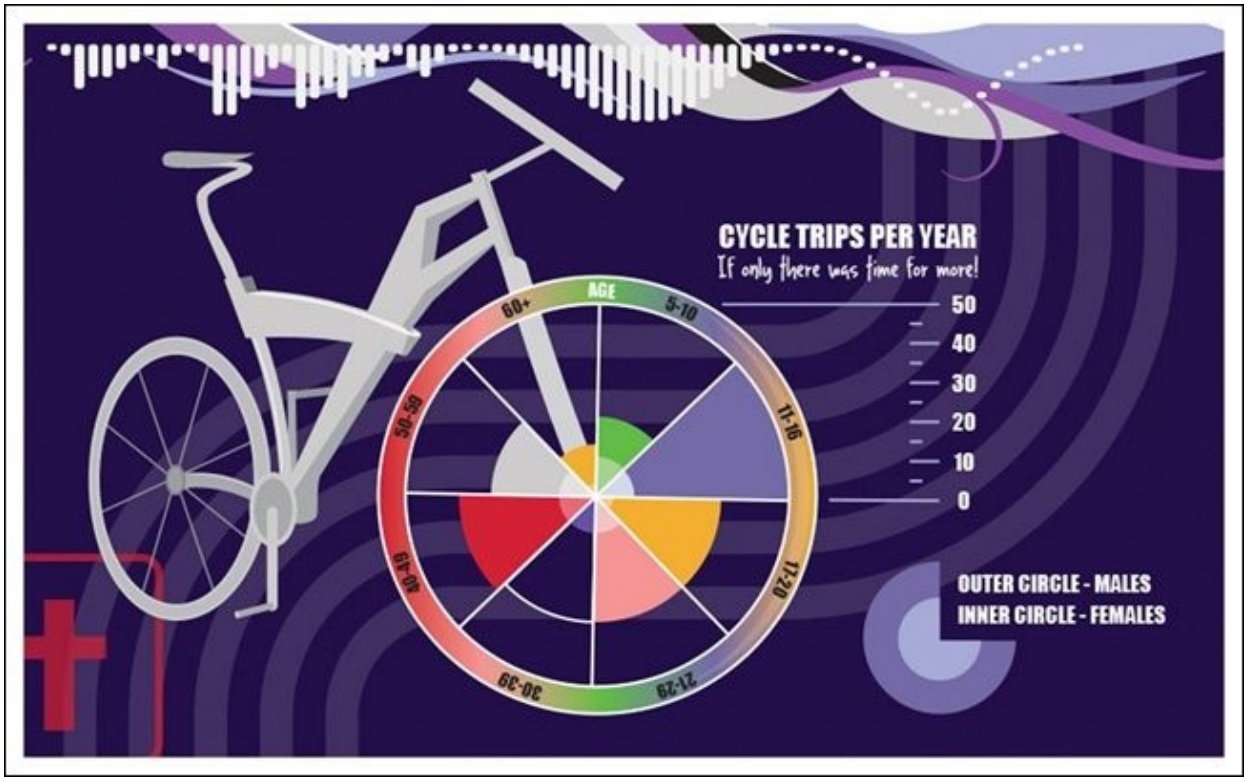



*Figure 11-27. Chart Junk via WTFViz*

According to Wikipedia, Chart Junk is “all the visual elements in charts and graphs that are not necessary to comprehend the information represented on the graph, or that distract the viewer from this information.”

Analytics3D is the poster child for Chart Junk. The visual elements in these charts are so distracting that it is impossible to even begin to decipher the data. But, hey, it’s in 3D!

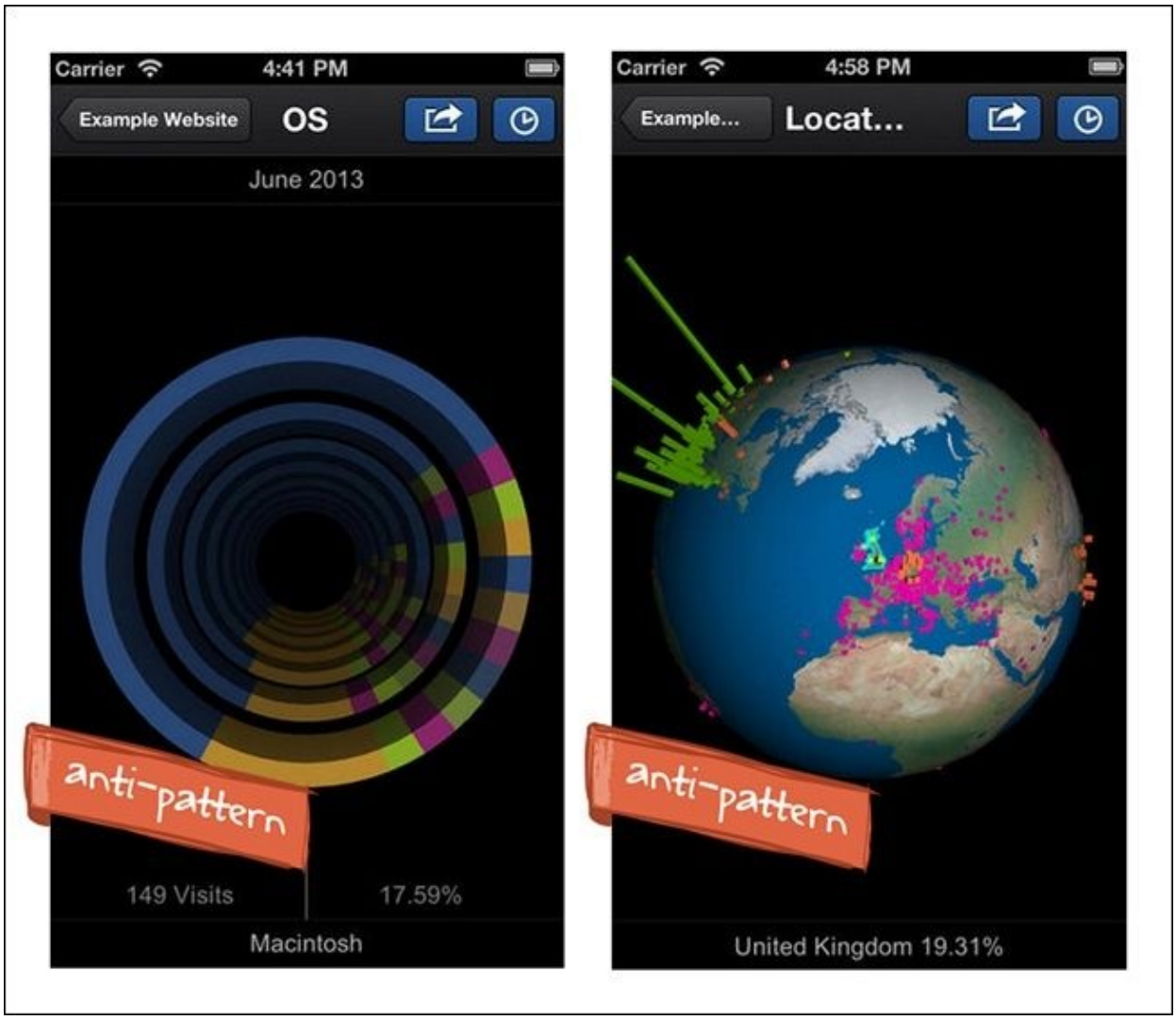



*Figure 11-28. Analytics 3D for Android: this poster child for Chart Junk includes a pie chart/wormhole and a spiky globe*

In the anti-pattern example from OOKLA Speedtest, the glowing, iridescent design of the gauge and the blue background of shifting sands distracts from the data. Cashish is even more of a challenge to read, with a full rainbow color palette overlaid on a dollar bill.

Even a design element as seemingly harmless as gridlines can be Chart Junk. In two health apps, Glucose Buddy and Heart Pal, bold horizontal and vertical gridlines actually make the data harder to read.


*Figure 11-29. OOKLA Speedtest for Android and Cashish for iOS: Chart Junk-o-rama*


*Figure 11-30. Glucose Buddy and Heart Pal for iOS: gridlines can detract from the data*

The next two examples just have too much information on the page. Pie charts are tricky in mobile apps. Because of limited screen real estate, it is hard to show both the chart and its legend.

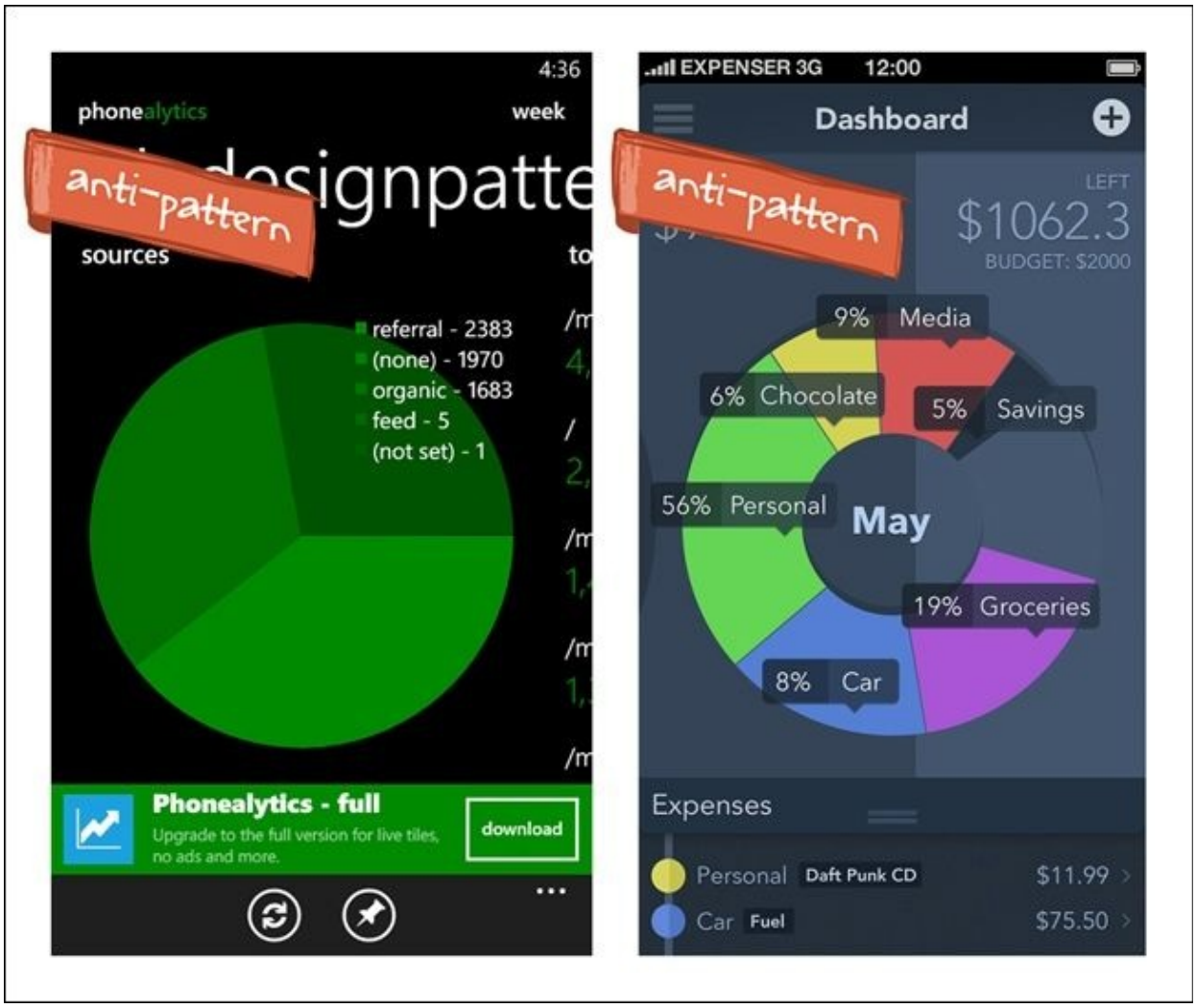



*Figure 11-31. Phonealytics for Windows Phone, and chart design concept for iOS: pie charts are a challenge in mobile*

But Roambi has come up with an innovative and usable design. You can spin the chart, and a fixed indicator shows the details for the wedge it points to.

> **NOTE**
>
> In charts and graphs, use only the visual elements that are absolutely necessary to communicate the information being represented. See Chapter 6 for chart design best practices.

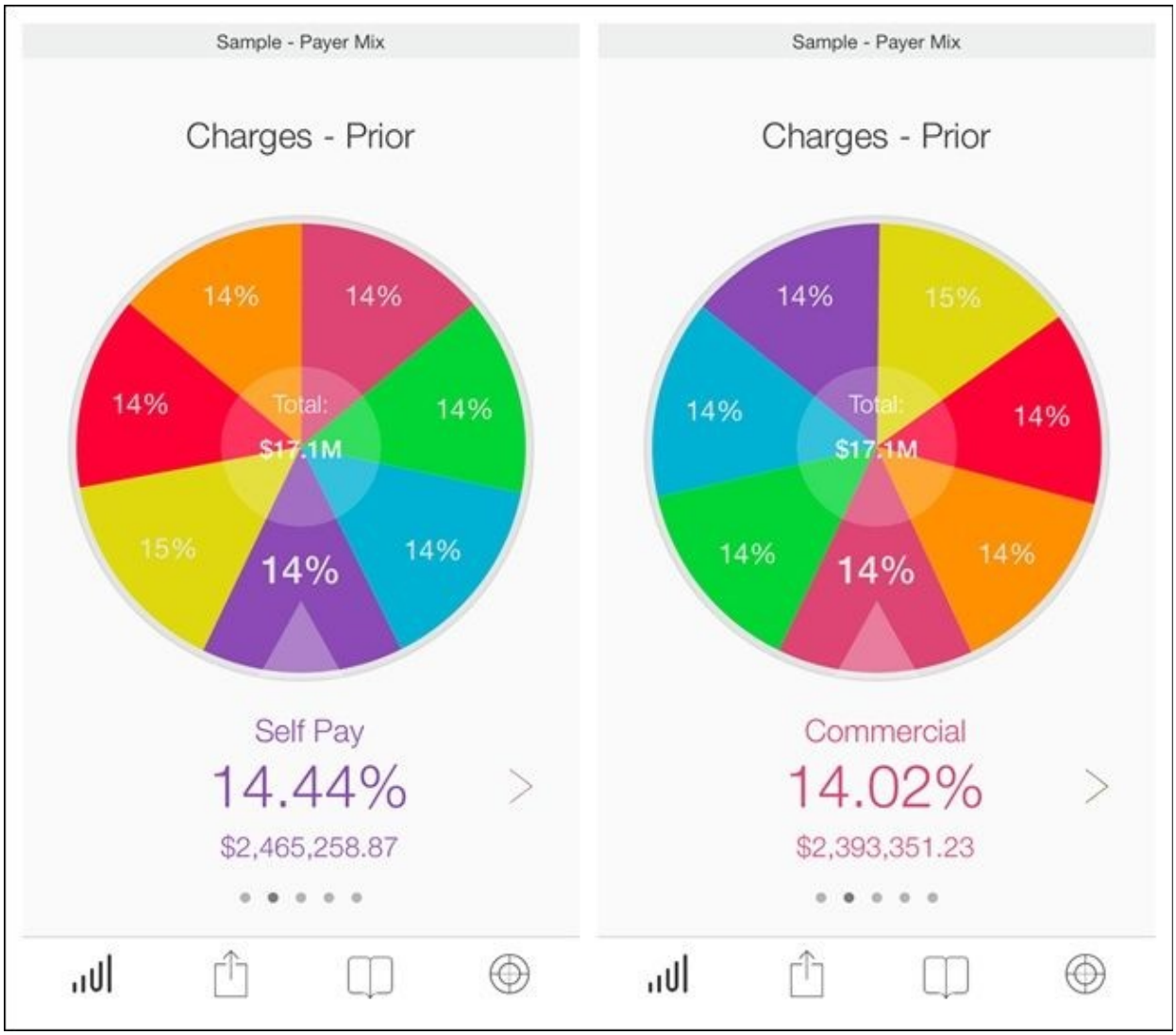



*Figure 11-32. Roambi for iOS: pie chart design that works on a mobile screen (http://youtu.be/WMmMkGF64XY)*

# Oceans of Buttons

Bill Scott and I first named the Oceans of Buttons anti-pattern while conducting usability reviews at Sabre Airline Solutions in 2002. We tasked our team with evaluating dozens of web and desktop applications to find the ones most in need of a UI redesign. Many of the applications had Java Swing or Visual Basic UIs and included a button bar, either down on the right or across the bottom. Sometimes in the bottom bar there were more buttons than could fit horizontally, so two rows were displayed. All the buttons were the same size and color and basically the same visual weight, so it was difficult to determine which one to click without reading all of them.

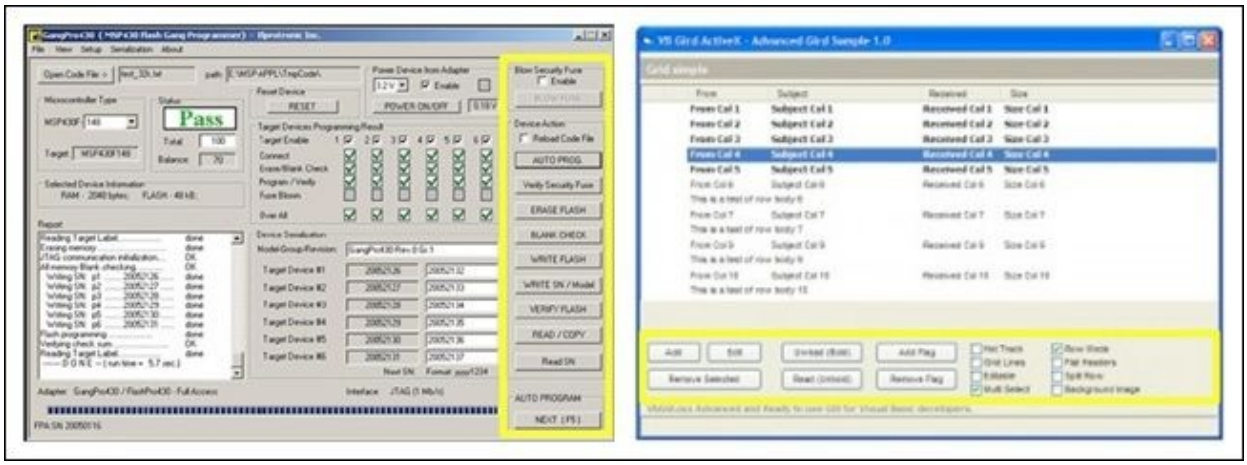



*Figure 11-33. Visual Basic and Java desktop applications: drowning in Oceans of Buttons*

My favorite example for this anti-pattern is Visual KPI. Since the first edition of this book, this app has managed to add in even more controls and buttons! One hundred percent more buttons, in fact. I would normally sketch up a quick redesign here, but this app needs a complete information architecture overhaul.

Another good example is Redfin: there are so many buttons (and elements that look like buttons) on the screen that it is hard to focus on the content (the pictures and descriptions of houses). Zillow's property page is much cleaner; the Toolbar shows the primary actions and tucks the secondary actions under the More option.


*Figure 11-34. Visual KPI for iOS: circa 2010–2011 on the left, and circa 2013 on the right — now with more buttons!*

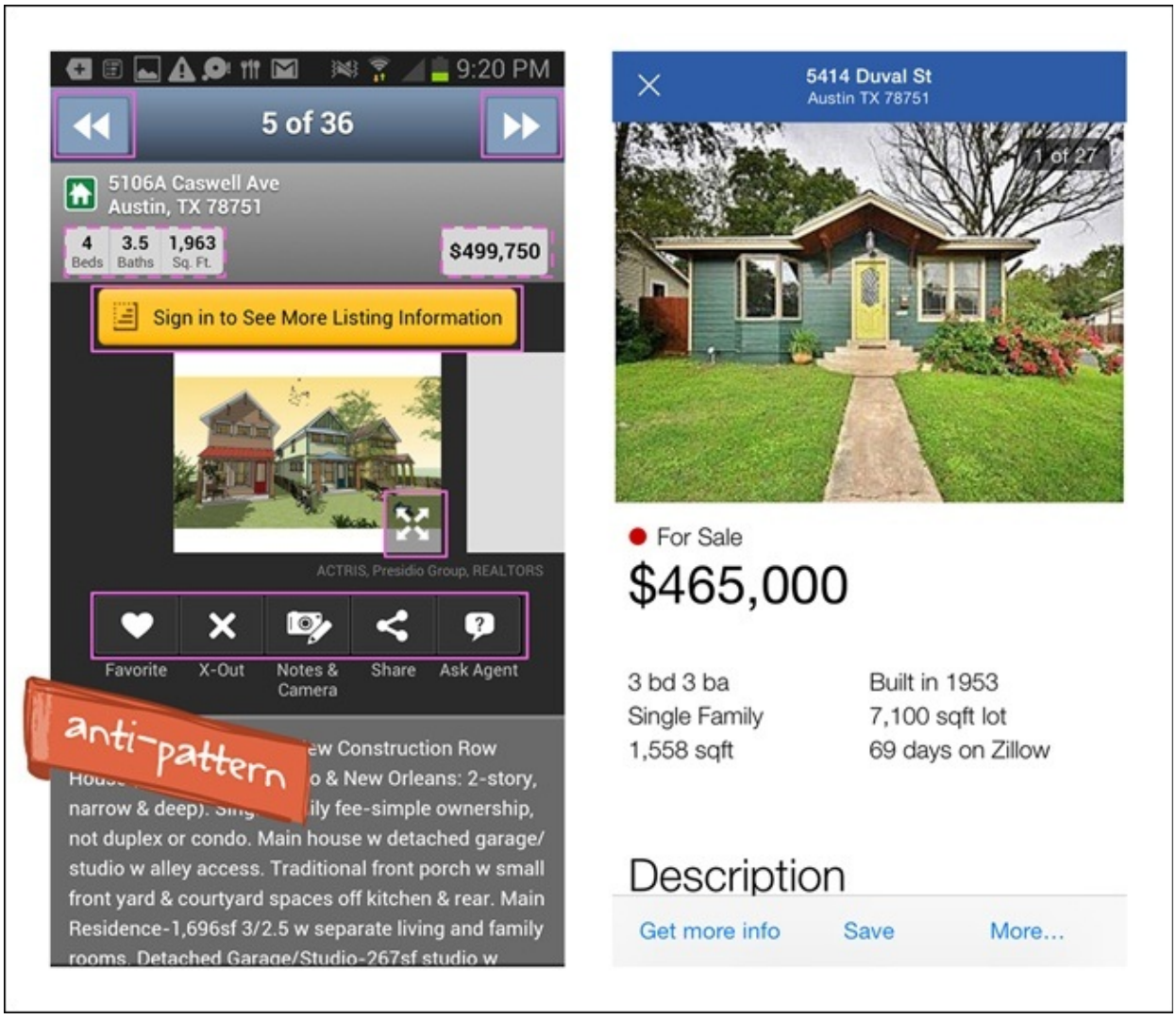



*Figure 11-35. Redfin for Android and Zillow for iOS: Oceans of Buttons versus clean design with Toolbar*

CarMax doesn’t have an ocean of buttons, but its four-corners design is suboptimal for button placement. The Android Action Bar would provide an easy fix, as my redesign illustrates. Save and Share could be shown, while the Call and Email options could be accessed from the More menu.

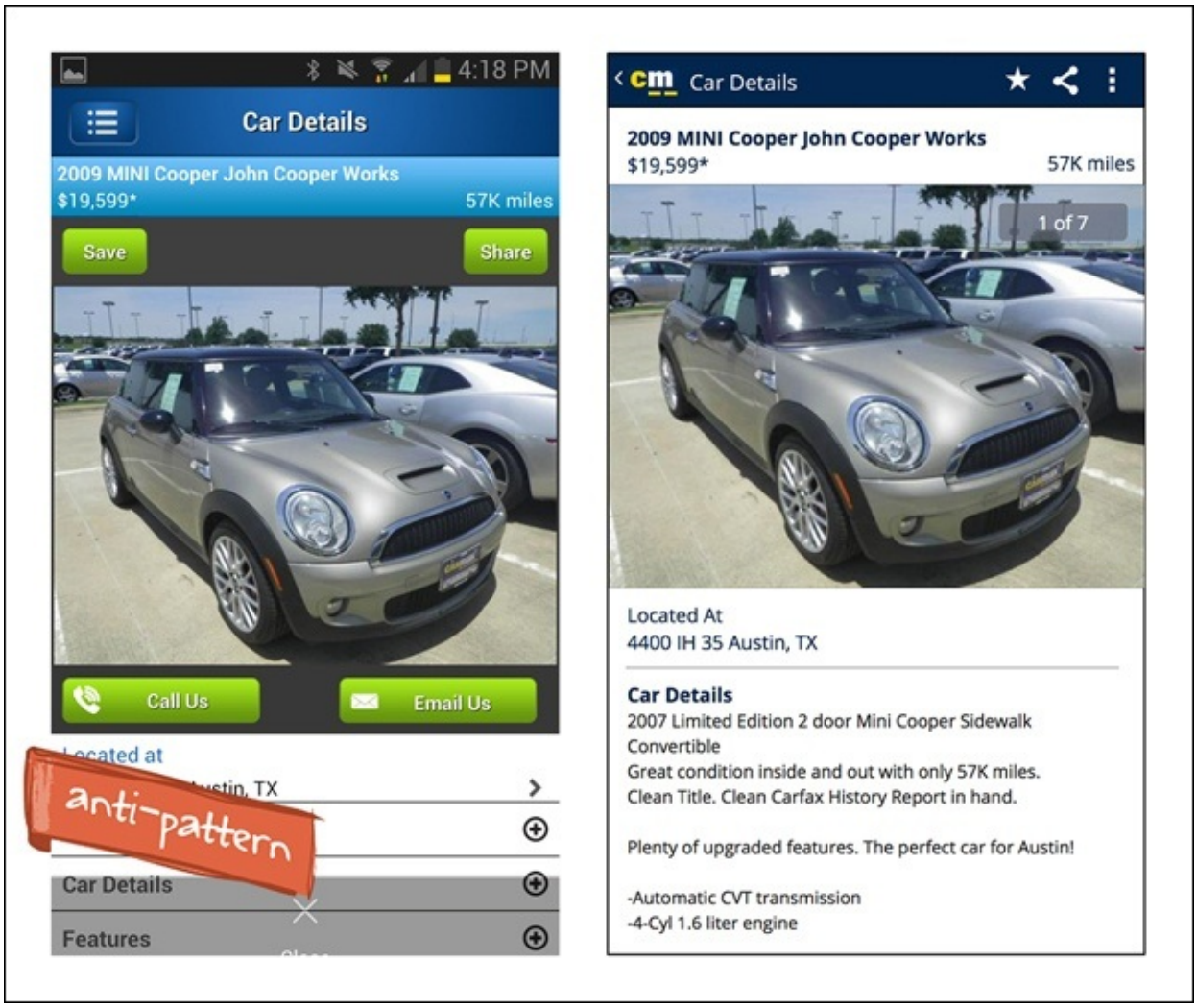



*Figure 11-36. CarMax for Android: suboptimal four-corners button placement, and my low-fidelity redesign with the Android Action Bar*

Plume uses the Contextual Tools pattern to reveal options when a tweet is tapped. But using tap for revealing tools instead of showing the full conversation thread created an interaction design problem. How does the user view the full thread? I would not have done what Plume did, which was to add another row for the “View full conversation” button. Instead, I would have rethought the interaction design.

Tweetbot solved this problem in its iOS design. It also uses the tap gesture to reveal the Contextual Tools, but you swipe to the left to open the message details.

**NOTE**

Use OS-specific controls to provide page-level actions. That way, buttons stay out of the way until there

is a context to use them. See Chapter 5 for best practices.

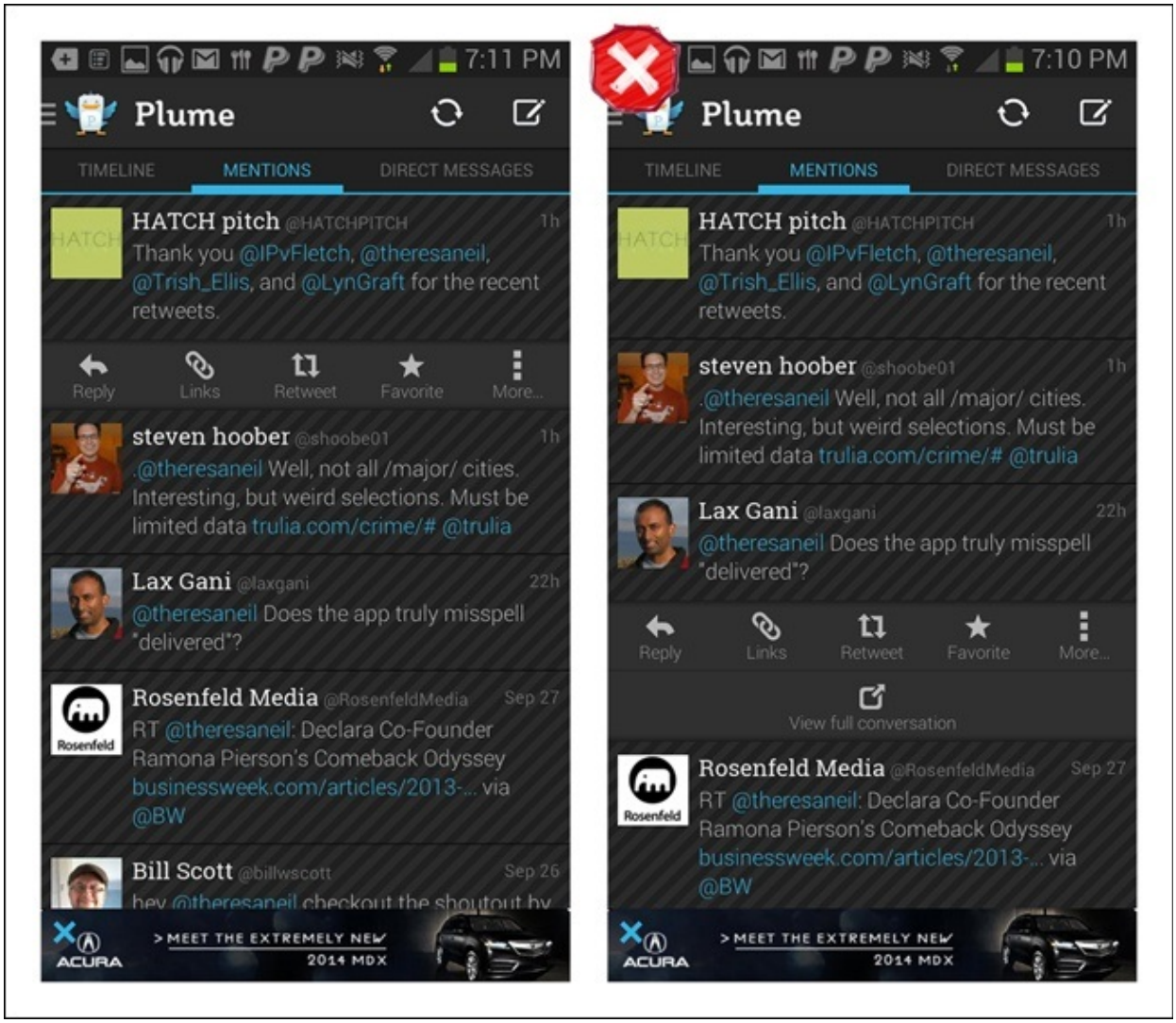



*Figure 11-37. Plume for Android: one row of buttons is fine, but two rows is too many*

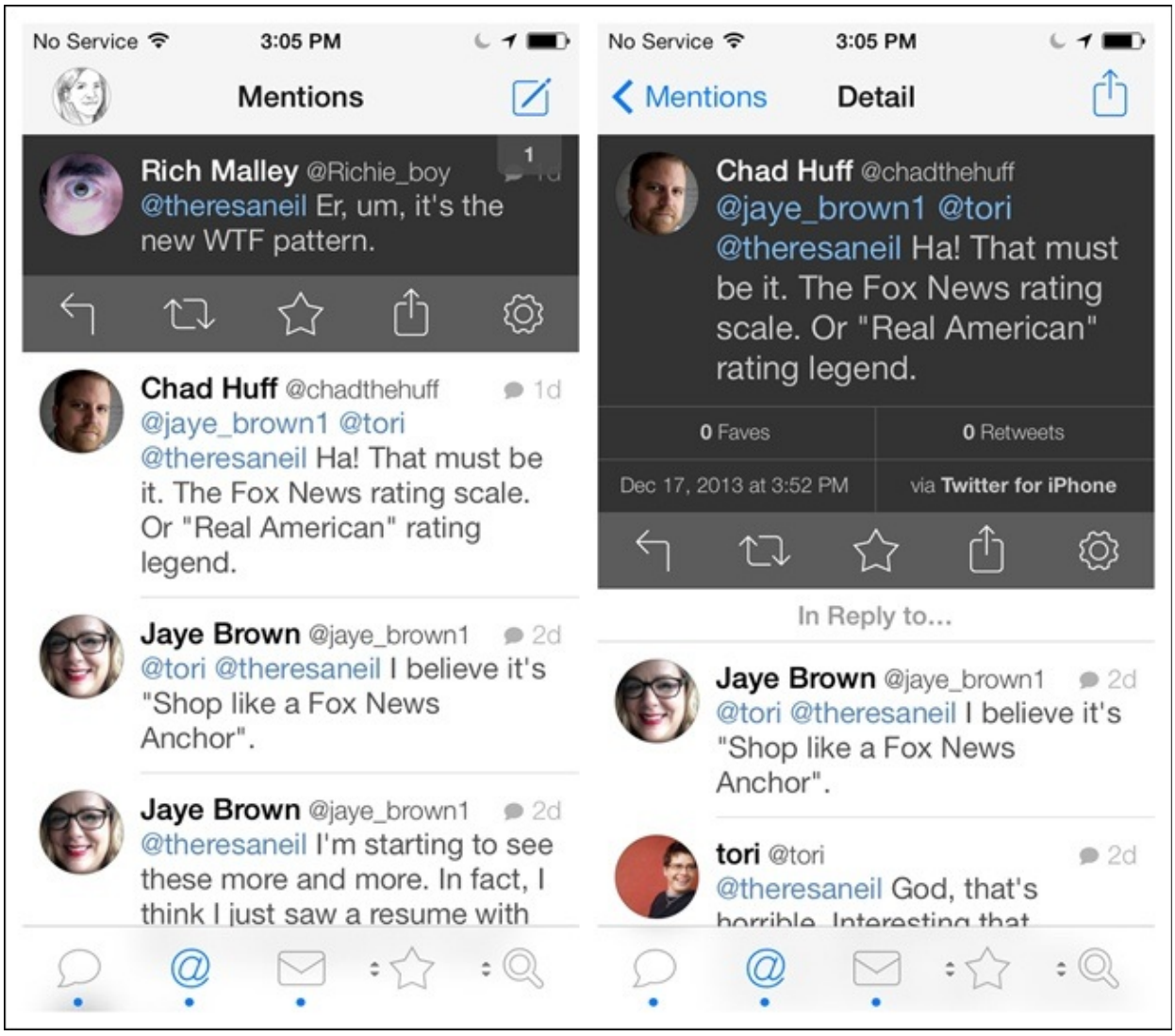



*Figure 11-38. Tweetbot: tap message for Contextual Tools; swipe left for message details*

## Square Peg, Round Hole

The Square Peg, Round Hole anti-pattern is usually the result of an uninformed decision at the executive level or a rush to get to market quickly, or both. It involves taking a design made for one platform and forcing it into another platform.

Behr Paint is just one of many companies that ported their iOS design straight to Android with no modifications. Not only does this look funny (the graphics are the wrong resolution), but it also breaks Android design conventions (see Pure Android for the list at *http://developer.android.com/design/patterns/pure-android.html*).

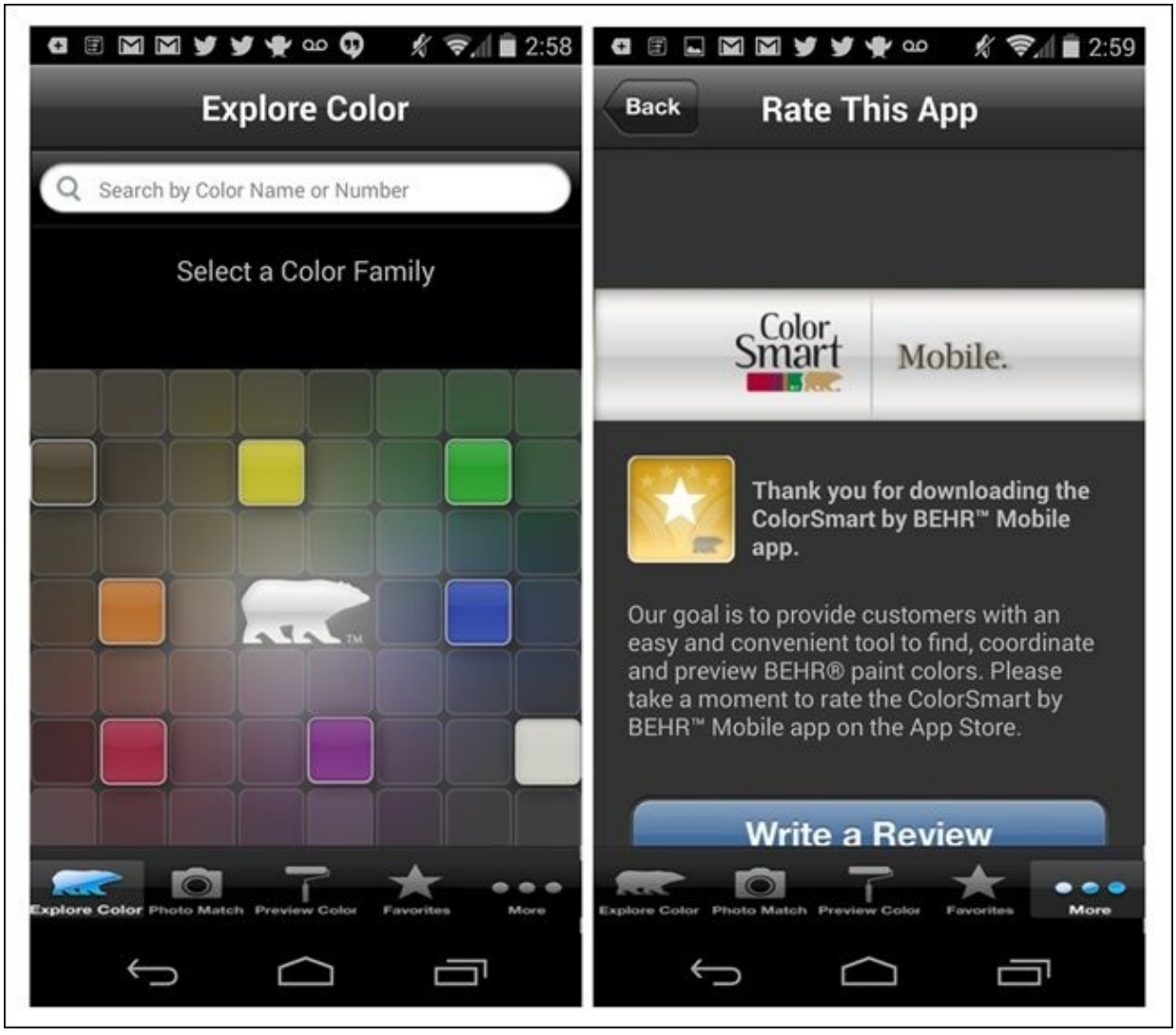



*Figure 11-39. Behr Paint for Android: iOS design ported one-for-one to Android*

Gap simply wrapped its mobile site and called it an app (*http://bit.ly/1fFCCfx*). But users aren't fooled — see the reviews (*http://bit.ly/1fFCBs4*).

I think SmartyPig did the same thing. The first page you see is a Sign In screen that is clearly not optimized for my Nexus 5. I expect the adoption of this app is fairly low.

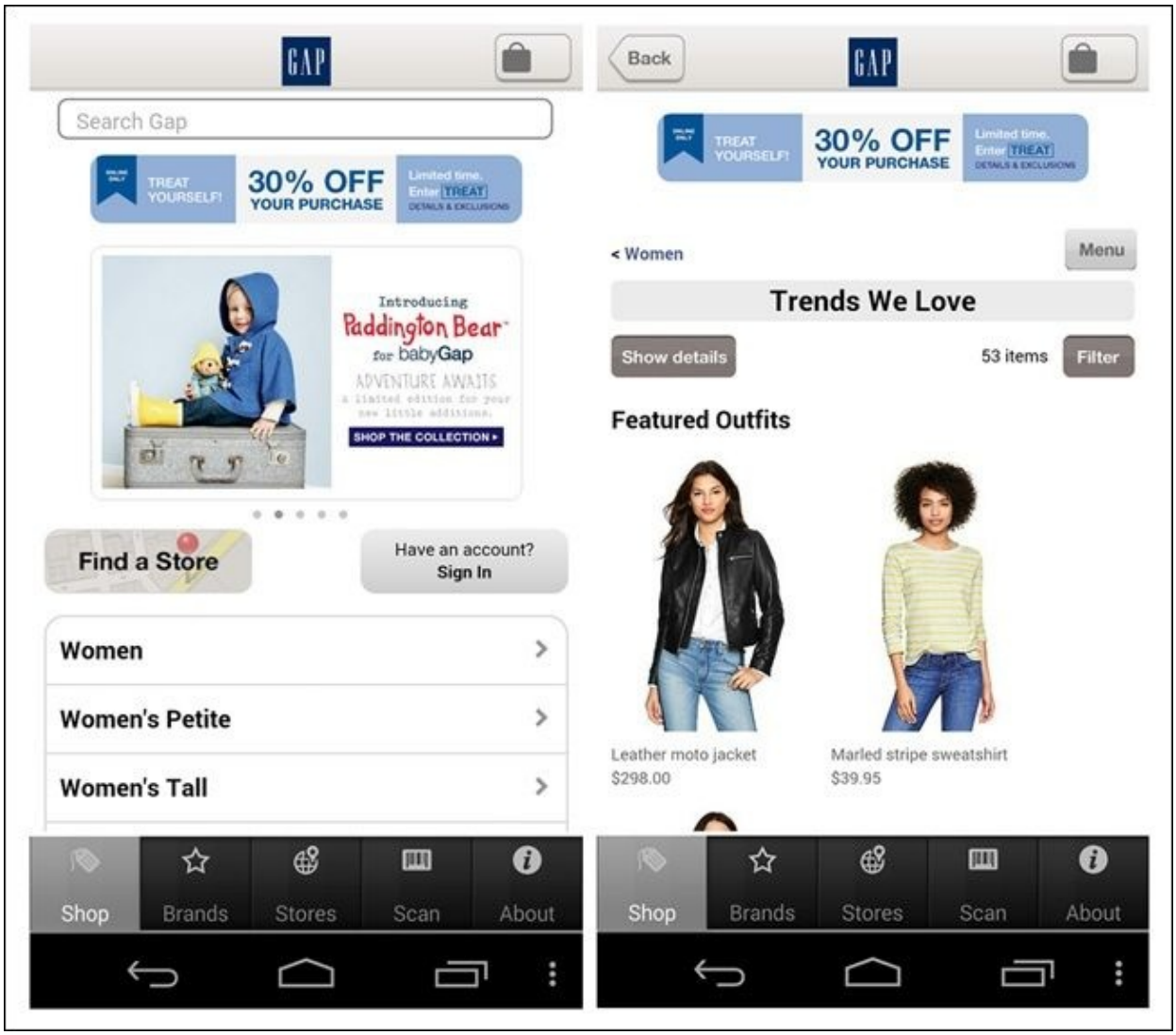



*Figure 11-40. Gap for Android and iOS: mobile website in a native app wrapper*

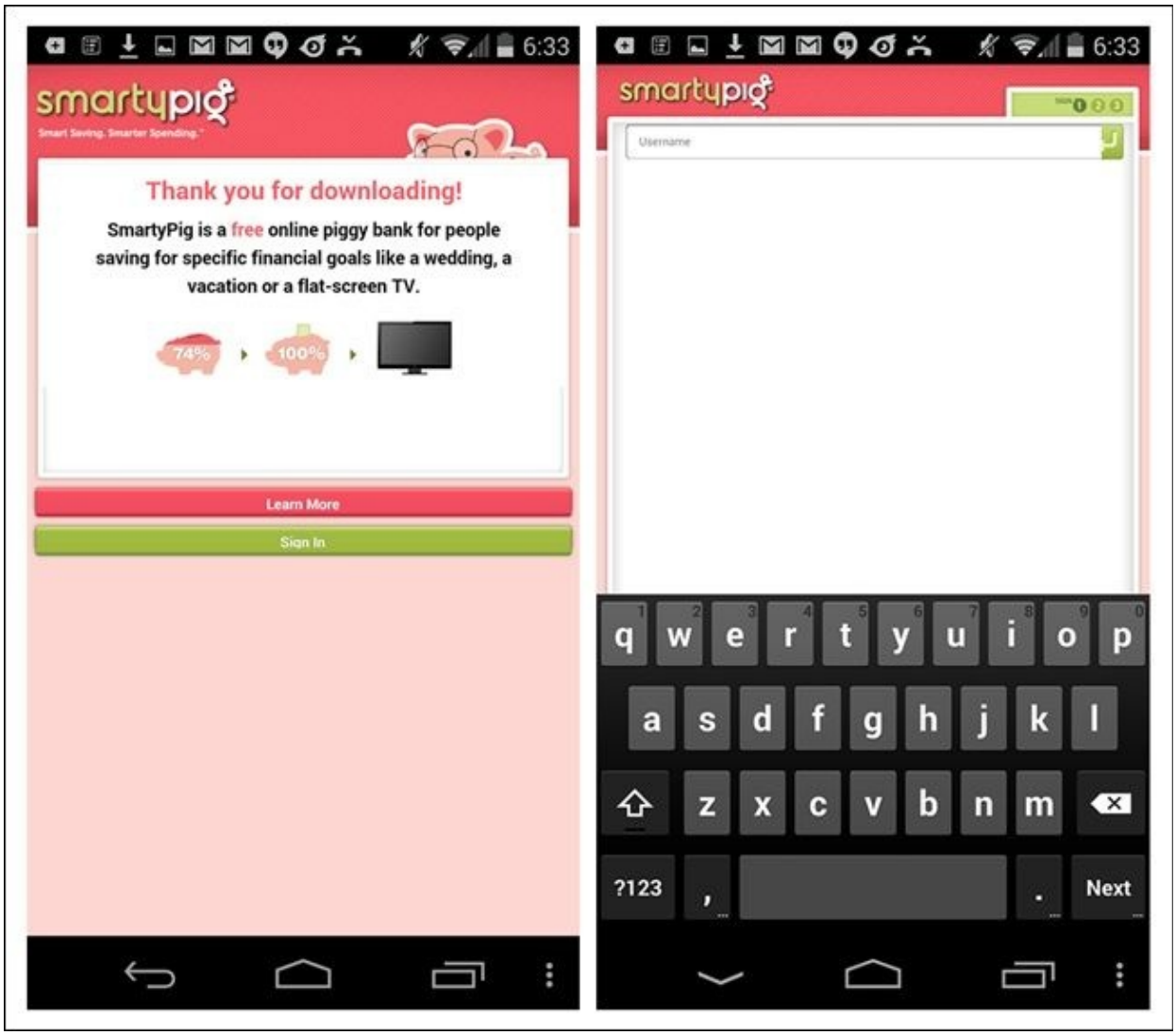



*Figure 11-41. SmartyPig for Android*

Avoid the Square Peg, Round Hole anti-pattern by:

- Thoroughly reading and understanding the design guidelines for the target OS
- Using the device you're designing/developing for on a daily basis until you are familiar with the nuances of the platform
- Designing distinct UIs for each platform, using those platform's conventions

Also, beware of platforms out there that promise you can "code it once, deploy it everywhere." If you're going to use one of these platforms, choose one that lets you create a custom UI for each OS, like Xamarin (*https://xamarin.com/*) or PhoneGap (*http://phonegap.com/*).

**NOTE**

Each platform is distinctly different: a great iOS design will be suboptimal for Android, and vice versa. Design and test to create a great experience for each platform.

# Chapter Extra: Let Them Pee — Avoiding the Sign-Up/Sign-In Mobile Anti-Pattern

**GREG NUDELMAN, FOUNDER OF DESIGN CAFFEINE**

**ANTI-PATTERN TERM COURTESY OF TAMARA ADLIN**

**FROM *ANDROID DESIGN PATTERNS: INTERACTION DESIGN SOLUTIONS FOR DEVELOPERS* (WILEY, 2013) REPUBLISHED ON BOXESANDARROWS**

Anything that slows down customers or gets in their way after they download your app is a bad thing. That includes sign-up/sign-in forms that show up even before potential customers can figure out if the app is actually worth using.

Long sign-up screens detract from the key mobile use case: quick, simple information access on the go. This anti-pattern seems to be going away more and more as companies are beginning to figure out the following simple UX equation:

> Long sign-up form before you can use the app = Delete app

However, a fair number of apps still force customers to sign up, sign in, or perform some other useless action before they can use them.

For example, the application SitOrSquat is a brilliant little piece of social engineering software that enables people to find bathrooms on the go, when they gotta go.

But this urgency is all but unfelt by Procter & Gamble, the company that acquired the app. Not content with the business of simply “Squeezing the Charmin” (that is, simple advertising), P&G executives decided for some

unfathomable reason to force people to sign up for the app — in multiple ways.

The entire app outside of registration consists of basically four screens (if you count the functionality to add bathrooms!). However, if you include all the sign-up anti-pattern screens, it takes seven screens of “preliminary” garbage before the content you are looking for finally shows up. If you count the number of taps necessary to enter my birthday, that is almost 50 taps!

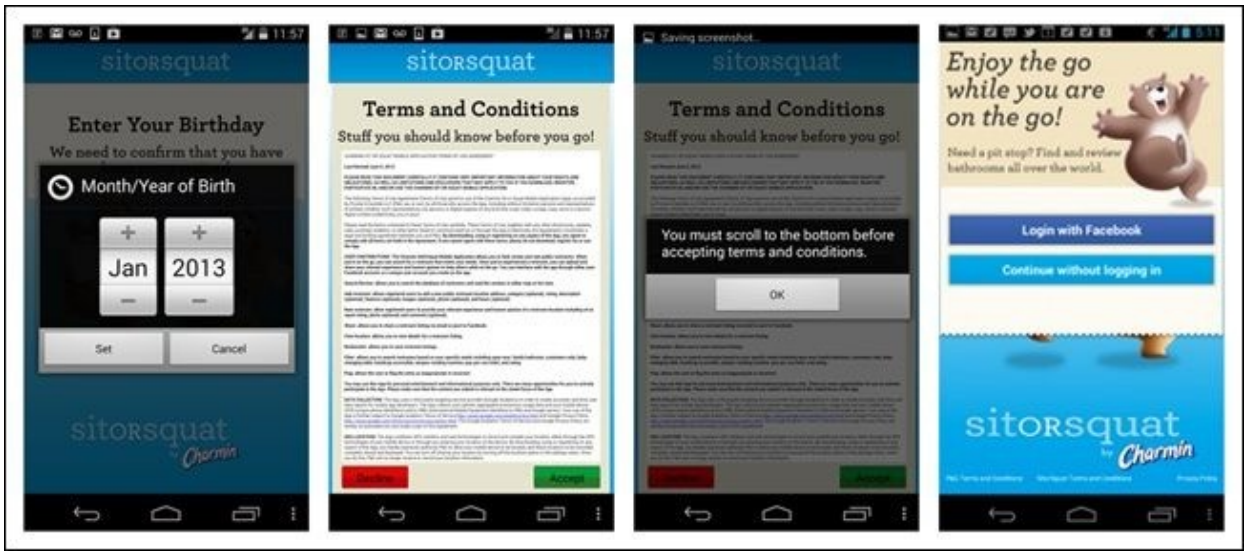



*Figure 11-42. Charmin SitOrSquat for Android (2013)*

When deciding whether to force the customer to perform an action, consider this: if this were a web app, would you force the customer to do this? When Internet connectivity is available, you can save everything the customer does and connect it back to his device using a simple session token and a guest account. And even when he is off-network, today’s smartphones have plenty of on-board storage you can use for syncing with your servers when the mobile network later becomes available.

This means there is simply no reason to force anyone to register for anything, other than if he wants to share the data from his phone with other devices. As a general rule, rather than forcing registration upon download or at the first opportunity, it is much better to allow the customer to save a piece of information locally on the phone without requiring that he log in. Wait until the customer asks for something that requires registration, such as sharing the information with another device or accessing information already saved in his account; at that point, completing the registration makes perfect sense. And even then, ask only for what you strictly need to proceed to the next step and omit extraneous information.

# Appendix A. Additional Resources

*Websites*

*http://www.mobiledesignpatterngallery.com*
*http://www.theresaneil.com*

*Flickr*

*http://www.flickr.com/photos/mobiledesignpatterngallery/sets/*

*Twitter*

*https://twitter.com/theresaneil*

# Index

**A NOTE ON THE DIGITAL INDEX**

A link in an index entry is displayed as the section title in which that entry appears. Because some sections have multiple index markers, it is not unusual for an entry to have several links to the same section. Clicking on any link will take you directly to the place in the text in which the marker appears.

## Symbols


1-Click checkout (Amazon), Tip #4: Offer Express Checkout

1Weather for Android (charts), Charts

4th & Mayor for Windows Phone (Application Bars), Windows Phone

9-dot puzzle, Primary Navigation Patterns, Persistent

The $300 Million Button (article), Sign In


## A


ABC News for iPad (Mental Model Mismatches), Mental Model Mismatch

About Face 3: The Essentials of Interaction Design, Idiot Boxes

Accordion pattern (secondary navigation), Scrolling Tabs

Action Bar, Android, Android, Oceans of Buttons

action buttons (Tumblr/Everlapse), Android

Action Sheets, iOS

Affordance patterns

- Drag, Swipe/Flick
- overview, Affordance
- Swipe/Flick, Tap
- Tap, Affordance

Airbnb for Android

Filter Overlays, Filter Overlay

location-based searches, Saved, Recent, and Popular Search

Side Drawer pattern, Toggle Menu

Airbnb for iOS

Filter Forms, Filter Form

Search Results, Search Results/View Results

Side Drawer pattern, Emerging Pattern

Alaska Airlines for Android (Novel Notions), Novel Notions

Allrecipes for Windows Phone (Filter Forms), Filter Form

Allthecooks for Android

animation, Swipe/Flick

Onscreen Sorts, Sort Overlay

Side Drawer pattern, Side Drawer

Allthecooks for iOS (Search Results), Search Results/View Results

Amazon Cloud Player

for Android (Onscreen Filters), Filter Patterns

for iOS (Scoped Search), Saved, Recent, and Popular Search

Amazon for Android

Confirmation patterns, Confirmation

Explicit Search, Explicit Search

Amazon for iOS

Confirmation patterns, Confirmation

Express Checkout, Tip #5: Forget the Web

Tab Menu pattern, iOS
Amazon for Windows Phone (Application Bars), Android
Amazon Price Check for iOS (Needless Complexity), Needless Complexity
American Airlines
for iOS (Side Drawer pattern), Emerging Pattern
for Windows Phone (Search forms), Search Forms
Analytics 3D for Android (Chart Junk), Chart Junk
Analytics for iOS (Dashboard pattern), Dashboard
Analytics Tiles for iOS (Sparklines pattern), Sparklines
Ancestry for Android (Command Button), Tables
Android
1Weather for (charts), Charts
Action Bar, Android, Oceans of Buttons
Airbnb for (Filter Overlays), Filter Overlay
Airbnb for (location-based searches), Saved, Recent, and Popular Search
Airbnb for (Side Drawer pattern), Toggle Menu
Alaska Airlines for (Novel Notions), Novel Notions
Allthecooks for (animation), Swipe/Flick
Allthecooks for (Onscreen Sorts), Sort Overlay
Allthecooks for (Side Drawer pattern), Side Drawer
Amazon Cloud Player for (Onscreen Filters), Filter Patterns
Amazon for (Confirmation patterns), Confirmation
Amazon for (Explicit Search), Explicit Search

Analytics 3D for (Chart Junk), Chart Junk

Ancestry for (Command Button), Tables

Any.do for (Contextual Tools), Contextual Tools

Any.do for (learning reinforcement), Rule #5: Listen to Your Users

Any.do for (Lock Screen Controls), Bulk Actions

ArkMail for (Page Swiping), Page Swiping

Audible for (Lock Screen Controls), Lock Screen Controls

BaconReader for (Contextual Tools), Contextual Tools

Badoo for (Implicit Search), Implicit Search

Battery Graph Notification for (Thresholds pattern), Integrated Legend

BBC News for (Tab Menu pattern), Gallery

Behr Paint for (Square Peg, Round Hole), Square Peg, Round Hole

Boomerang for (Feature Tours), Tutorials

Boomerang for (tutorial text), SlideStory Compared to Vine

Box for (Contextual Tools), Contextual Tools

Box for (Sign Up & Log In), Sign In

Bump for, Windows Phone

Buy Me a Pie! for (tutorial frontloading), Clipchat Compared to Kik

CarMax for (Oceans of Buttons), Oceans of Buttons

Cashish for (Chart Junk), Chart Junk

Catch for (tutorial text), SlideStory Compared to Vine

Charmin SitOrSquat for (sign-up/sign-in forms), Chapter Extra: Let Them Pee — Avoiding the Sign-Up/Sign-In Mobile Anti-Pattern

Check for (Confirmation patterns), Confirmation

CheckPay for (Popular Searches), Saved, Recent, and Popular Search

Clear for (tutorial text), SlideStory Compared to Vine

Conqu for (Control Mismatches), Metaphor Mismatch

DailyBurn for (Control Mismatches), Metaphor Mismatch

design guidelines, Side Drawer

DigiCal for (Modal Forms), Tables

DigiCal for (tutorial text), SlideStory Compared to Vine

Do It (Tomorrow) for (Persistent Invitations), Persistent Invitations

Dropbox for (Contextual Tools), Contextual Tools

EasyJet for, Springboard

eBay for (Explicit Search), Explicit Search

eBay for (Up button), List Menu

Edmunds app for (Control Mismatches), Control Mismatch

ESPN SportsCenter for (tables), Tables

Etsy for (CTA Buttons), Call to Action Button

Etsy for (Registration feedback), Registration

Etsy for (Sign In), Registration

Eureka Offers for (Needless Complexity), Needless Complexity

Evernote Food for (Tips design), Tips

Evernote for (Contextual Toolbars), Windows Phone

Evernote for (Feature Tours), Feature Tours

Evernote for (Onscreen Sorts), Sort Overlay

Evernote Hello for (tiles), Springboard

Expedia for (Filter Overlays), Filter Overlay

Expedia for (Novel Notions), Novel Notions

Expedia for (Search Forms), Search Results/View Results

Expedia for (Search Results), Search Results/View Results

Expedia for (Sort Overlays), Sort Overlay

Expense Manager for (Persistent Invitations), Persistent Invitations

Fancy for (Registration labels), Registration

Fidelity for (Bulk Actions), Bulk Actions

Flipboard for (tutorial interactivity), Noom Compared to DailyBurn Tracker

Foodspotting for (Implicit Search), Implicit Search

Foodspotting for (Sign In/Sign Up tabs), Sign In

Frank & Oak for (CTA Buttons), Call to Action Button

Gap for (Square Peg, Round Hole), Square Peg, Round Hole

Gas Guru for (tables), Headerless Table

Gaug.es for (Overview plus Data pattern), Overview plus Data

Gaug.es for (Sparklines pattern), Sparklines

Gmail for (Action Bars), Android

Gmail for (Confirmation patterns), Confirmation

Gmail for (Inline Actions), Inline Actions

Google Drive for (Editable Tables), Search, Sort, and Filter

Google for (Onscreen Filters), Onscreen Filter

Google Maps for (Help), Help

Google Maps for (Inline Actions), Inline Actions

Google Maps for (Page Swiping), Page Swiping

Google Maps for (Sort Results), Sort Patterns

Google Play for (Confirmation patterns), Confirmation

Google Play for (ScrollingTabs), Scrolling Tabs

Google Play Store for (Scoped Search), Scoped Search

Google Translate for (Needless Complexity), Needless Complexity

Groupon for (Onscreen Filters), Onscreen Filter

Groupon for (Popular Searches), Saved, Recent, and Popular Search

HelloWallet for (Overview plus Data pattern), Overview plus Data

Hipmunk for (CTA Buttons), Call to Action Button

Hipmunk for (Novel Notions), Novel Notions

Hipmunk for (Search Forms), Saved, Recent, and Popular Search

Hipmunk for (Sign In), Sign In

Hipmunk for (Sort Overlays), Sort Overlay

HoursTracker for (Tips design), Tips

Instagram for (Sign In & Register), Sign In

Kayak for (outdated desktop metaphors), Tools

Kindle for (Help), How-Tos

KitKat for (Springboard pattern), Springboard

Kobo for (Sort Overlays), Sort Overlay

Layar for (Implicit Search), Implicit Search

LinkedIn for (Springboard design), Springboard

LinkedIn Pulse for (Tab Menu pattern), Gallery

Lose It! for (Dashboard pattern), Dashboard

Lose It! for (Thresholds pattern), Thresholds

Mint for (Grouped Rows tables), Table with Visual Indicators

Move for (Overview plus Data pattern), Overview plus Data

Netflix for (Auto-Complete), Search with Auto-Complete

Netflix for (Swipe/Flick), Swipe/Flick

Newegg for (Auto-Complete), Search with Auto-Complete

Newegg for (Checkout forms), Tip #4: Offer Express Checkout

Newegg for (Scrolling Tabs), Windows Phone

News360 for (Registration with Personalization), Registration with Personalization

Nike Training Club for (Gamification patterns), Gamification

Nook for (Registration), Registration

Noom for (Gamification patterns), Gamification

Noom for (Sparklines pattern), Sparklines

Noom for (tutorial interactivity), Noom Compared to DailyBurn Tracker

Noom for (tutorials), Contextual Help

Notif for (Command Button), Tables

NPR for (Toggle Menu pattern), Toggle Menu

NYTimes for (Confirmation patterns), Confirmation

NYTimes for (Toggle Menu pattern), Toggle Menu

OOKLA for (Chart Junk), Chart Junk

OpenTable for (Search forms), Search Forms

PageOnce for (Registration feedback), Registration

Path for (Bulk Actions), Bulk Actions

Path for (floating CTAs), Call to Action Button

PayPal for (Feature Tours), Feature Tours

Pinterest for (Persistent Invitations), Persistent Invitations

Pixlr for (Toolbox), Toolbox

PlayStore for (Expand/Collapse Panel), Accordion

PlayStore for (Onscreen Filters), Onscreen Filter, Onscreen Filter

Plume for (Oceans of Buttons), Oceans of Buttons

Pocket for (Onscreen Filters), Onscreen Filter

Priceline for (MultiStep registration), MultiStep

Realtor.com for (Headerless Tables), Headerless Table

Redfin for (Headerless Tables), Fixed Column

Redfin for (Oceans of Buttons), Oceans of Buttons

RedLaser for (Explicit Search), Explicit Search

RememberTheMilk for (Registration labels), Registration

RememberTheMilk for (Sign In), Forms

RetailMeNot for (Side Drawer pattern), Skeuomorphic

Runtastic Heart Rate for (Overview plus Data tables), Overview plus Data

Silvercar for (Capture Feedback), Capture Feedback

Skype for (Swipe/Flick), Swipe/Flick

Sleep Charts for (Table with Visual Indicators), Table with Visual Indicators

SmartyPig for (Square Peg, Round Hole), Square Peg, Round Hole

Sniper Ghost Warrior 2 for (Skeuomorphic pattern), Skeuomorphic

SnipSnap for (Help), How-Tos

SnipSnap for (MultiStep registration), MultiStep

Songza for (Auto-Complete), Search with Auto-Complete

Songza for (Novel Notions), Novel Notions

Songza for (ScrollingTabs), Scrolling Tabs

Soundwave for (Error Messages), Feedback Patterns

Soundwave for (Novel Notions), Novel Notions

Stock Trainer for (tables), Tables

Tab Menu patterns for, Android

Target for (Auto-Complete), Search with Auto-Complete

TaxCaster for (Calculator forms), Calculator Forms

Trulia for (charts), Charts

Trulia for (Explicit Search), Explicit Search

Trulia for (Sort Overlays), Sort Overlay

TuneIn for (ScrollingTabs), Scrolling Tabs

TurboTax SnapTax for (MultiStep registration), MultiStep

Uber for (Sign In), Sign In

Up button in (List Menu navigation), List Menu

USA Today for (Novel Notions), Novel Notions

Vimeo for, Springboard

Vine for (simple share forms), Tables

Walmart for (Toggle Menu pattern), Toggle Menu

Withings for (Dashboard pattern), Dashboard

anti-patterns
Chart Junk, Chart Junk
Idiot Boxes, Idiot Boxes
Metaphor Mismatch, Metaphor Mismatch
Needless Complexity, Needless Complexity
Novel Notions, Novel Notions
Oceans of Buttons, Oceans of Buttons
overview, Anti-Patterns
Square Peg, Round Hole, Square Peg, Round Hole
Any.do for Android
Contextual Tools, Contextual Tools
learning reinforcement, Rule #5: Listen to Your Users
Lock Screen Controls, Bulk Actions
APIs, social network, MapMyFitness Compared to We Heart It
App Store
for iOS (Confirmation patterns), Confirmation
for iOS (Multi-State Buttons), Multi-State Button
for iOS (Sort Results), Sort Patterns
App Tabs (Windows Phone), Windows Phone
Apple hierarchical navigation, List Menu
Apple Stocks for iOS (Data Point Details), Data Point Details
Apple Store
EasyPay self-checkout feature, Calculator Forms

for iOS (Checkout forms), Tip #5: Forget the Web
for iOS (Search Results), Search Results/View Results
Application Bar (Windows Phone), Android
Argus for iOS
charts, Charts
Contextual Help, Capture Feedback
Persistent Invitations, Persistent Invitations
ArkMail for Android (Page Swiping), Page Swiping
AroundMe for iOS (List Menu), List Menu
Audi Configurator (MultiStep forms), MultiStep
Audible for Android (Lock Screen Controls), Lock Screen Controls
Audible for iOS
Confirmation patterns, Confirmation
Dynamic Search, Search with Auto-Complete
FAQs, Feature Tours
Gesture Mismatches, Icon Mismatch
Page Swiping, Page Swiping
Sort Overlays, Sort Overlay
Swipe/Flick, Swipe/Flick
Audible for Windows Phone (Gamification patterns), Gamification
augmented reality (Implicit Search), Implicit Search
authenticating users, Sign In
Auto-Complete, Explicit Search

Aviary for iOS

Drag affordance, Drag

Toolbox, Toolbox

Avoid the Spinner (article), System Status

Awesome Note for iOS (Skeuomorphic pattern), Skeuomorphic

## B

BaconReader for Android (Contextual Tools), Contextual Tools

Badoo for Android (Implicit Search), Implicit Search

Basic pattern (tables), Tables

Battery Graph Notification for Android (Thresholds pattern), Integrated Legend

BBC News for Android (Tab Menu), Gallery

BBC Radio for Windows Phone, Springboard

Behr Paint Calculator, Calculator Forms

Behr Paint for Android (Square Peg, Round Hole), Square Peg, Round Hole

BestBuy for iOS (Confirmation patterns), Confirmation

BillGuard for iOS

Overview plus Data tables, Overview plus Data

primary & secondary patterns, Secondary Navigation Patterns

tables, Headerless Table

Boomerang for Android

Feature Tours, Tutorials

tutorial text, SlideStory Compared to Vine

Box for Android

Contextual Tools, Contextual Tools

Sign Up & Log In, Sign In

Boxer for iOS

Confirmation patterns, Confirmation

Onscreen Sorts, Sort Patterns

Briggs, Alissa, Chapter Extra: Invitations — Rolling Out the Welcome Mat

Brit & Co. for Windows Phone (secondary navigation), Secondary Navigation Patterns

Bulk Actions, Bulk Actions

Bump for Android, Windows Phone

Buttons Are a Hack campaign, Tools

Buy Me a Pie! for Android (tutorial frontloading), Clipchat Compared to Kik

## C

Calculator forms (Checkout), Calculator Forms

CalendarPRO (tiles), Springboard

Call to Action (CTA) Buttons, Call to Action Button

Capture Feedback (Help patterns), Capture Feedback

Card pattern (persistent navigation), Cards

CarMax for Android (Oceans of Buttons), Oceans of Buttons

Cashish for Android (Chart Junk), Chart Junk

Catch for Android (tutorial text), SlideStory Compared to Vine

CBS Outdoors, Calculator Forms

Chaikin for iOS (Chart with Filters), Charts

Charmin SitOrSquat for Android (sign-up/sign-in forms), Chapter Extra: Let

Them Pee — Avoiding the Sign-Up/Sign-In Mobile Anti-Pattern

charts

basics, Charts

Chart Junk (anti-patterns), Chart Junk

Chart with Filters pattern, Charts

charting libraries, Charts

Dashboard pattern, Dashboard

Data Point Details pattern, Data Point Details

Drill Down pattern, Overview plus Data, Drill Down

Integrated Legend pattern, Integrated Legend

interactive, Tables

Interactive Preview pattern, Interactive Preview

Interactive Timeline pattern, Interactive Timeline

overview, Charts

Overview plus Data pattern, Drill Down

pie charts, Chart Junk

Pivot Table pattern, Thresholds

simplifying, Charts

Sparkline pattern, Sparklines

testing, Charts

Thresholds pattern, Integrated Legend

well-designed apps using, Pivot Table

Zoom pattern, Zoom

Check for Android (Confirmation patterns), Confirmation

Checkout forms

Calculator forms, Calculator Forms

Express Checkout, Tip #4: Offer Express Checkout

mobile retail innovations, Tip #5: Forget the Web

overview, Checkout

Sign In/Register/Guest options, Tip #1: Include Sign In, Register, and Guest Options

streamlining flow of, Tip #2: Streamline the Flow

time-saving shortcuts, Tip #2: Streamline the Flow

CheckPay for Android (Popular Searches), Saved, Recent, and Popular Search

Chipotle for iOS (Idiot Boxes), Idiot Boxes

Clark, Josh, Search, Sort, and Filter, Tools

Clear for Android/iOS (tutorial text), SlideStory Compared to Vine

ClipChat for iOS (tutorial frontloading), Clipchat Compared to Kik

column alignment (tables), Tables

Command button

Android, Long Forms

iOS guidelines, Long Forms

Configurable Tabs design, Emerging Patterns

Confirm Email/Password fields (forms), Registration

Confirmation patterns (Feedback), Error Messages

Connecting patterns (Social networks), MapMyFitness Compared to We Heart It

Conqu for Android (Control Mismatches), Metaphor Mismatch

Contacts for iOS (Dynamic Search), Dynamic Search

Contextual Help patterns, Contextual Help

Contextual Toolbars, Windows Phone

Contextual tools, Contextual Tools

Control Mismatches (anti-patterns), Metaphor Mismatch

Cooper, Alan, Idiot Boxes

Coverflow/Carousel (Affordance), Swipe/Flick

Creative Studio for Windows (tutorial frontloading), Clipchat Compared to Kik

Creative Studio for Windows Phone (Contextual Help), Capture Feedback

Cross DJ for iOS (Skeuomorphic pattern, Skeuomorphic

## D

DailyBurn for Android (Control Mismatches), Metaphor Mismatch

DailyBurn Tracker for iOS

- Chart with Filters, Chart with Filters
- tutorial interactivity, Noom Compared to DailyBurn Tracker, Noom Compared to DailyBurn Tracker

Dark Sky for iOS

- Data Point Details, Data Point Details
- Interactive Timelines, Interactive Timeline

Dashboard pattern

- charts, Dashboard
- persistent navigation, Dashboard

Data Point Details (charts), Data Point Details

Data Visualization Best Practices, Charts

Day One for iOS (List Menu), List Menu

Delete Confirmation dialog box, Idiot Boxes

delete controls, Multi-State Button

Designing for Thumb Flow (article), Registration with Personalization

Diabetes concept app, Thresholds

Dictionary.com for Windows Phone/iOS (Explicit Search), Explicit Search

DigiCal for Android

  Modal Forms, Tables

  tutorial text, SlideStory Compared to Vine

direct immersion (invitations case study), Chapter Extra: Invitations — Rolling Out the Welcome Mat

Discoverable Invitations, Discoverable Invitations

Do It (Tomorrow) for Android (Persistent Invitations), Persistent Invitations

Dooo for iOS (tutorial frontloading), Clipchat Compared to Kik

Drag patterns, Swipe/Flick

Dribbble online design (charts), Charts

Drill Down pattern (charts), Overview plus Data, Drill Down

Drink Builder on iOS (MultiStep registration), MultiStep

drop zones, Drag

Dropbox for Android (Contextual Tools), Contextual Tools

Dropbox for iOS

  duplicate Registration, Registration

  FAQs, FAQs

  Sort Overlays, Sort Overlay

  Toolbars, Tools

Duolingo for iOS (Gamification patterns), Gamification

Dwell (Tab Menu design), Emerging Patterns

Dynamic Search patterns, Search with Auto-Complete

Dynamic Search, Headerless Table with, Tables

## E

EasyJet for Android, Springboard

EasyPay self-checkout feature (Apple Store), Calculator Forms

eBay for Android, List Menu

eBay for iOS

- Filter Drawers, Filter Drawer
- Saved/Recent Searches, Saved, Recent, and Popular Search
- Sort Overlays, Sort Overlay

eBay for Windows Phone/Android (Explicit Search), Explicit Search

Editable Tables pattern, Table with Visual Indicators

Edmunds app for iOS/Android (Control Mismatches), Control Mismatch

Elevatr for iOS (Expand/Collapse Panel), Accordion

Error Message patterns, Feedback Patterns

Escape button (iOS guidelines), Long Forms

ESPN SportsCenter for Android (tables), Tables

Etsy for Android

- CTA Buttons, Call to Action Button
- Registration feedback, Registration
- Sign In, Registration

Eureka Offers for Android (Needless Complexity), Needless Complexity

Everest for iOS (Tutorials), Contextual Help

Everlapse for iOS (action buttons)), Android

Evernote Food for Android (Tips design), Tips

Evernote for Android

- Contextual Toolbars, Windows Phone
- Feature Tours, Feature Tours
- Onscreen Sorts, Sort Overlay

Evernote for Windows Phone (tiles), Springboard

Evernote Hello for Android (tiles), Springboard

Expand/Collapse Panel pattern (secondary navigation), Scrolling Tabs

Expedia for Android

- Filter Overlays, Filter Overlay
- Novel Notions, Novel Notions
- Search Forms, Search Results/View Results
- Search Results, Search Results/View Results
- Sort Overlays, Sort Overlay

Expedia for iOS

- Checkout forms, Tip #1: Include Sign In, Register, and Guest Options
- Contextual Help, Capture Feedback
- Drag actions, Drag
- Onscreen Sorts, Sort Patterns

Expense Manager for Android (Persistent Invitations), Persistent Invitations

Explicit Search patterns, Explicit Search

Express Checkout (forms), Tip #4: Offer Express Checkout

Extra Credits online video series, Tutorial Rules

## F

Facebook

- beta for Windows Phone (Side Drawer pattern), Side Drawer
- Card pattern, Cards
- Paper for iOS (Registration with Personalization), MultiStep
- Tab Menu pattern for, Gallery

Facebook for iOS

- Group patterns, Groups
- Inline Actions, Inline Actions
- Needless Complexity, Needless Complexity
- Springboard design, Springboard
- User Guides, User Guide/Help System

Fancy for Android (Registration labels), Registration

Fancy for iOS

- CTA Buttons, Call to Action Button
- Side Drawer pattern, Side Drawer

Fantastical for iOS (tutorial text), SlideStory Compared to Vine

FAQs patterns, FAQs

Feature Tours (Help), Feature Tours

Feedback patterns

Confirmation, Error Messages
Error Messages, Feedback Patterns
overview, Feedback and Affordance
System Status, Confirmation
Fidelity (Auto-Complete), Search with Auto-Complete
Fidelity for Android (Bulk Actions), Bulk Actions
Fidelity for iOS
Chart with Filters, Charts
Feature Tours, Feature Tours
Fixed Column Tables, Fixed Column, Overview plus Data
invitations, Invitation Patterns
Filter patterns
Chart with Filters, Charts
Filter Drawer, Filter Drawer
Filter Form, Filter Overlay
Filter Overlay, Onscreen Filter
Gesture-Based Filtering, Filter Drawer
Onscreen Filter, Filter Patterns
overview, Filter Patterns
Fitbit for iOS (Sparklines pattern), Sparklines
Fitbit for Windows Phone (browser-based Registration), Registration
Fixed Column pattern (tables), Fixed Column
Fixed Tabs pattern (Android), Android

flat design, Affordance

Flava for iOS

Expand/Collapse Panel, Accordion

User Guides, FAQs

Flightboard for iOS

Headerless Tables, Headerless Table

Skeuomorphic pattern, Skeuomorphic

Flipboard for Android (tutorial interactivity), Noom Compared to DailyBurn Tracker

Flipboard for iOS

animation, Swipe/Flick

Persistent Invitations, Persistent Invitations

Flixster for iOS (Contextual Tools), Contextual Tools

Float Label pattern (forms), Registration

floating CTA Buttons, Call to Action Button

Following patterns (social networks), Following

Foodspotting for Android

Implicit Search, Implicit Search

Sign In/Sign Up tabs, Sign In

Foodspotting for iOS

Inline Actions, Inline Actions

Page Swiping, Page Swiping

Profiles patterns, Profiles

tutorial text, SlideStory Compared to Vine

forms
Checkout (see Checkout forms)
labels, alignment of, Registration
Long, Long Forms
MultiStep registration, MultiStep
overview, Forms
Registration, Registration
Registration with Personalization, Registration with Personalization
resources for designing, Forms
Search, Search Forms
Sign In, Forms
Sign-up/Sign-in forms anti-pattern, Square Peg, Round Hole
Foursquare for iOS
Confirmation patterns, Confirmation
CTA Buttons, Call to Action Button
floating CTAs, Call to Action Button
location-based searches, Saved, Recent, and Popular Search
Sort Forms, Filter Patterns
Frank & Oak on Android (CTA Buttons), Call to Action Button
Frequency for iOS (Configurable Tabs), Emerging Patterns
frontloading, avoiding (tutorials), SlideStory Compared to Vine
Frost Bank for iOS
Side Drawer pattern, Side Drawer

Sign In PIN, Sign In

## G

Gallery pattern (persistent navigation), Dashboard

game designing, Tutorials and Invitations

Gamification patterns, Gamification

Gap for Android/iOS (Square Peg, Round Hole), Square Peg, Round Hole

Gas Guru for Android (tables), Headerless Table

GasBuddy for iOS (Gamification patterns), Gamification

Gaug.es for Android

Overview plus Data pattern, Overview plus Data

Sparklines pattern, Sparklines

Gekoboard for iOS (Dashboard pattern), Dashboard

Geo Walk (Mental Model Mismatches), Mental Model Mismatch

Gesture Mismatches (anti-patterns), Icon Mismatch

Gesture-Based Filtering, Filter Drawer

Gilt for iOS

Checkout forms, Tip #4: Offer Express Checkout

Confirmation patterns, Confirmation

Glucose Buddy for iOS (Chart Junk), Chart Junk

Gmail for Android

Action Bars, Android

Confirmation patterns, Confirmation

Gmail for iOS

Confirmation patterns, Confirmation

Side Drawer pattern, Side Drawer

Google Apps for iOS (Capture Feedback), Capture Feedback

Google Drive for Android (Editable Tables), Search, Sort, and Filter

Google for Android (Onscreen Filters), Onscreen Filter

Google for iOS (System Status), System Status

Google for Windows (Contextual Help), Capture Feedback

Google for Windows Phone (Scoped Search), Scoped Search

Google Maps (Implicit Search), Implicit Search

Google Maps for Android

Help, Help

Inline Actions, Inline Actions

Page Swiping, Page Swiping

Sort Results, Sort Patterns

Google Now

Card pattern, Cards

Implicit Search, Implicit Search

Search Results, Search Results/View Results

Google Offers (Needless Complexity), Needless Complexity

Google Play for Android

Confirmation patterns, Confirmation

ScrollingTabs, Scrolling Tabs

Google Play Store for Android (Scoped Search), Scoped Search

Google Translate for Android (Needless Complexity), Needless Complexity

Google Wallet, Tip #4: Offer Express Checkout

Google+ (Tabbed Profiles), Groups

gridlines (tables), Tables

Group patterns (social networks), Groups

Group.me for iOS (Group patterns), Groups

Grouped Rows pattern (tables), Overview plus Data

Groupon for Android

  Onscreen Filters, Onscreen Filter

  Popular Searches, Saved, Recent, and Popular Search

Groupon for iOS (Sign In), Forms, Registration

Guest option (Checkout forms), Tip #1: Include Sign In, Register, and Guest Options

Gunther, Richard, Novel Notions

## H

handles, drag, Drag

Haute for iOS

  Checkout forms, Tip #2: Streamline the Flow

  Idiot Boxes, Idiot Boxes

HauteLook for iOS

  Command button, Long Forms

  Confirmation patterns, Confirmation

Headerless pattern (tables), Tables, Headerless Table

Heart Pal for iOS (Chart Junk), Chart Junk

HelloWallet for Android (Overview plus Data pattern), Overview plus Data

HelloWallet for iOS (Registration), Registration

Help patterns

- Capture Feedback, Capture Feedback
- Contextual Help, Contextual Help
- FAQs, FAQs
- Feature Tours, Feature Tours
- How-Tos, How-Tos
- overview, Help
- Tutorials, Tutorials
- User Guides/Help System, How-Tos

hierarchical apps, List Menu

hint text (forms), Registration

Hipmunk for Android

- CTA Buttons, Call to Action Button
- Novel Notions, Novel Notions
- Search Forms, Saved, Recent, and Popular Search
- Sign In, Sign In
- Sort Overlays, Sort Overlay

Hipmunk for iOS

- Search forms, Search Forms, Search Forms
- Search Results, Search Results/View Results
- Sort Forms, Sort Form

Hipstamatic for iOS (Skeuomorphic pattern), Skeuomorphic

Home Depot for iOS, MultiStep

Home Depot for Windows Phone (Toggle Menu pattern), Toggle Menu

HomeAway for iOS

  Gesture-Based Filters, Filter Drawer

  Sort Forms, Sort Form

Homesnap for iOS (Saved/Recent Searches), Saved, Recent, and Popular Search

Homestyler for iOS (Drag), Drag

HoursTracker for Android (Tips design), Tips

How-To instructions, How-Tos

HuffPost for iOS (Page Swiping), Page Swiping

Hungry Now for Windows Phone (Implicit Search), Implicit Search

## I

Icon Mismatches (anti-patterns), Icon Mismatch

Idiot Boxes (anti-patterns), Error Messages, Idiot Boxes

IfThisThenThat for iOS (Side Drawer pattern), Side Drawer

Implicit Search patterns, Search Forms, Search, Sort, and Filter

Import Address from Contacts option (Checkout forms), Tip #2: Streamline the Flow

inlay Side Drawers, Skeuomorphic, Emerging Pattern

Inline Actions, Inline Actions

inline feedback

  for transactional gestures, Confirmation

  Registration, Registration

Innerfence for Windows Phone (Long Forms), Long Forms

Instagram (Tabbed Profiles), Groups

Instagram for Android (Sign In & Register), Sign In

Instagram for iOS

  CTA Buttons, Call to Action Button

  Registration, Registration

  Registration with Personalization, MultiStep

  Tab Menu pattern, Android

Integrated Legend pattern (charts), Integrated Legend

interactive charts, Tables

Interactive Preview pattern (charts), Interactive Preview

Interactive Timelines (charts), Interactive Timeline

invitation patterns

  Discoverable Invitations, Discoverable Invitations

  overview, Invitation Patterns

  Persistent Invitations, Persistent Invitations

  Snap Payroll case study

    direct immersion, Chapter Extra: Invitations — Rolling Out the Welcome Mat

    multistep transparency, Concept #3: MultiStep Transparency

    overview, Chapter Extra: Invitations — Rolling Out the Welcome Mat

    single-step transparency, Concept #3: MultiStep Transparency

    Tips, Concept #4: Single-Step Transparency

    Tour, Concept #2: Tour

Tips design, Invitation Patterns

iOS

Airbnb for (Filter Forms), Filter Form

Airbnb for (Search Results), Search Results/View Results

Airbnb for (Side Drawer pattern), Emerging Pattern

Allthecooks for (Search Results), Search Results/View Results

Amazon Cloud Player for (Scoped Search), Saved, Recent, and Popular Search

Amazon for (Confirmation patterns), Confirmation

Amazon for (Express Checkout), Tip #5: Forget the Web

Amazon for (Tab Menu pattern), iOS

Amazon Price Check for (Needless Complexity), Needless Complexity

American Airlines for (Side Drawer pattern), Emerging Pattern

Analytics for (Dashboard pattern), Dashboard

Analytics Tiles for (Sparklines pattern), Sparklines

App Store for (Confirmation patterns), Confirmation

App Store for (Multi-State Buttons), Multi-State Button

App Store for (Sort Results), Sort Patterns

Apple Stocks for (Data Point Details), Data Point Details

Apple Store for (Checkout forms), Tip #5: Forget the Web

Apple Store for (Search Results), Search Results/View Results

Argus for (Contextual Help), Capture Feedback

Argus for (Persistent Invitations), Persistent Invitations

Argus for iOS 7 (charts), Charts

AroundMe for (List Menu), List Menu

Audible for (Confirmation patterns), Confirmation

Audible for (Dynamic Search), Search with Auto-Complete

Audible for (FAQs), Feature Tours

Audible for (Gesture Mismatches), Icon Mismatch

Audible for (Page Swiping), Page Swiping

Audible for (Sort Overlays), Sort Overlay

Audible for (Swipe/Flick), Swipe/Flick

Aviary for (Drag), Drag

Aviary for (Toolbox), Toolbox

Awesome Note for (Skeuomorphic pattern), Skeuomorphic

BestBuy for (Confirmation patterns), Confirmation

BillGuard for (Overview plus Data tables), Overview plus Data

BillGuard for (primary & secondary patterns), Secondary Navigation Patterns

BillGuard for (tables), Headerless Table

Boxer for (Confirmation patterns), Confirmation

Boxer for (Onscreen Sorts), Sort Patterns

Chaikin for (Chart with Filters), Charts

Clear for (tutorial text), SlideStory Compared to Vine

ClipChat for (tutorial frontloading), Clipchat Compared to Kik

Contacts for (Dynamic Search), Dynamic Search

Cross DJ for (Skeuomorphic pattern, Skeuomorphic

DailyBurn Tracker for (Chart with Filters), Chart with Filters

DailyBurn Tracker for (tutorial interactivity), Noom Compared to DailyBurn

Tracker, Noom Compared to DailyBurn Tracker

Dark Sky for (Data Point Details), Data Point Details

Dark Sky for (Interactive Timelines), Interactive Timeline

Day One for (List Menu), List Menu

Dictionary.com for (Explicit Search), Explicit Search

Dooo for (tutorial frontloading), Clipchat Compared to Kik

Drink Builder for (MultiStep registration), MultiStep

Dropbox for (duplicate Registration), Registration

Dropbox for (FAQs), FAQs

Dropbox for (Sort Overlays), Sort Overlay

Dropbox for (Toolbars), Tools

Duolingo for (Gamification patterns), Gamification

eBay for (Filter Drawers), Filter Drawer

eBay for (Saved/Recent Searches), Saved, Recent, and Popular Search

eBay for (Sort Overlays), Sort Overlay

Edmunds for (Control Mismatches), Icon Mismatch

Elevatr for (Expand/Collapse Panel), Accordion

Everlapse for (action buttons)), Android

Expedia for (Checkout forms), Tip #1: Include Sign In, Register, and Guest Options

Expedia for (Contextual Help), Capture Feedback

Expedia for (Drag), Drag

Expedia for (Onscreen Sorts), Sort Patterns

Facebook for (Group patterns), Groups

Facebook for (Inline Actions), Inline Actions

Facebook for (Needless Complexity), Needless Complexity

Facebook for (Springboard design), Springboard

Facebook for (User Guides), User Guide/Help System

Fancy for (CTA Buttons), Call to Action Button

Fancy for (Side Drawer pattern), Side Drawer

Fantastical for (tutorial text), SlideStory Compared to Vine

Fidelity for (Chart with Filters), Charts

Fidelity for (Feature Tours), Feature Tours

Fidelity for (Fixed Column Tables), Fixed Column, Overview plus Data

Fidelity for (invitations), Invitation Patterns

Fitbit for (Sparklines pattern), Sparklines

Flava for (Expand/Collapse Panel), Accordion

Flava for (User Guides), FAQs

Flightboard for (Headerless Tables), Headerless Table

Flightboard for (Skeuomorphic pattern), Skeuomorphic

Flipboard for (animation), Swipe/Flick

Flipboard for (Persistent Invitations), Persistent Invitations

Flixster for (Contextual Tools), Contextual Tools

Foodspotting for (Inline Actions), Inline Actions

Foodspotting for (Page Swiping), Page Swiping

Foodspotting for (Profiles patterns), Profiles

Foodspotting for (tutorial text), SlideStory Compared to Vine

Foursquare for (Confirmation patterns), Confirmation

Foursquare for (CTA Buttons), Call to Action Button

Foursquare for (floating CTAs), Call to Action Button

Foursquare for (location-based searches), Saved, Recent, and Popular Search

Foursquare for (Sort Forms), Filter Patterns

Frequency for (Configurable Tabs), Emerging Patterns

Frost for (Side Drawer pattern), Side Drawer

Frost for (Sign In PIN), Sign In

Gap for (Square Peg, Round Hole), Square Peg, Round Hole

GasBuddy for (Gamification patterns), Gamification

Gekoboard for (Dashboard pattern), Dashboard

Gilt for (Checkout forms), Tip #4: Offer Express Checkout

Gilt for (Confirmation patterns), Confirmation

Glucose Buddy for (Chart Junk), Chart Junk

Gmail for (Confirmation patterns), Confirmation

Gmail for (Side Drawer pattern), Side Drawer

Google Apps for (Capture Feedback), Capture Feedback

Google for (System Status), System Status

Google Now for (Card pattern), Cards

Groupon for (Sign In), Forms, Registration

Haute for (Checkout forms), Tip #2: Streamline the Flow

Haute for (Idiot Boxes), Idiot Boxes

HauteLook for (Command button), Long Forms

HauteLook for (Confirmation patterns), Confirmation

Heart Pal for (Chart Junk), Chart Junk

HelloWallet for (Registration), Registration

Hipmunk for (Search forms), Search Forms, Search Forms

Hipmunk for (Search Results), Search Results/View Results

Hipmunk for (Sort Forms), Sort Form

Hipstamatic for (Skeuomorphic pattern), Skeuomorphic

Home Depot for, MultiStep

HomeAway for (Gesture-Based Filters), Filter Drawer

HomeAway for (Sort Forms), Sort Form

Homesnap for (Saved/Recent Searches), Saved, Recent, and Popular Search

Homestyler for (Drag), Drag

HuffPost for (Page Swiping), Page Swiping

IfThisThenThat for (Side Drawer pattern), Side Drawer

Instagram for (CTA Buttons), Call to Action Button

Instagram for (Registration with Personalization), MultiStep

Instagram for (Registration), Registration

Instagram for (Tab Menu pattern), Android

iOS 7 design guidelines, Long Forms

iOS Control Center, Side Drawer

iOS Design Guide, Emerging Pattern

iOS Human Interface Guideline, Tools

iOS Toolbar, Tools

iTranslate for (Needless Complexity), Needless Complexity

iTunes for (Lock Screen Controls), Lock Screen Controls

iTunes for (Onscreen Filters), Filter Patterns

iTunes for (Scoped Search), Saved, Recent, and Popular Search

Jamie Oliver's Recipes for (Idiot Boxes), Idiot Boxes

Kayak for (List Menu), List Menu

Kik for (Confirmation patterns), Confirmation

Kik Messenger for (tutorial frontloading), Clipchat Compared to Kik

Kindle for (Icon Mismatches), Icon Mismatch

Kobo for (Gamification patterns), Gamification

LearnVest for (Overview plus Data pattern), Overview plus Data

LearnVest for (Registration), Registration

LearnVest for (Springboard pattern), Springboard

Lemon Wallet for (Command button), Long Forms

Lemon Wallet for (Drag), Drag

Level for (Registration), Registration, Registration

Level for (System Status), System Status

LinkedIn for (Group patterns), Gamification

LinkedIn for (Inline Actions), Inline Actions

LinkedIn for (Profiles patterns), Groups

LinkedIn for (Side Drawer pattern), Side Drawer, Side Drawer

LinkedIn for (social connections), Connecting

Lootsy for (Drag action), Drag

Luvocracy for (Contextual Tools), Contextual Tools

Luvocracy for (Side Drawer pattern), Emerging Pattern

Mail for (Confirmation patterns), Confirmation

Mail for (Toolbars), Tools

Mailbox for (tutorial text), SlideStory Compared to Vine

MapMyFitness for (social connections), Connecting

MapMyFitness for (Social Registration), MapMyFitness Compared to We Heart It

Mint for (Dashboard pattern), Dashboard, Dashboard

Mint for (Drill Down), Drill Down

Mint for (Interactive Preview pattern), Interactive Preview

Mint for (Tips design), Tips

Move for (Overview plus Data pattern), Overview plus Data

Myspace for (floating CTAs), Inline Actions

Narrato for (Action Sheets), iOS

NASDAQ QMX for (Zoom pattern), Zoom

National Geographic Guides for (Swipe/Flick), Swipe/Flick

National Geographic Parks for (primary & secondary patterns), Secondary Navigation Patterns

National Parks for (Filter Overlays), Filter Overlay

NBC News for (tutorial interactivity), Noom Compared to DailyBurn Tracker

Ness for (Filter Overlays), Filter Overlay

Ness for (Page Swiping), Page Swiping

Ness for (Registration with Personalization), Registration with Personalization

Ness for (tutorial text), SlideStory Compared to Vine

News360 for (Page Swiping), Page Swiping

Newsstand for (Skeuomorphic pattern), Skeuomorphic

Nike+ FuelBand for (charts), Pivot Table

North Face for (Checkout forms), Checkout

Numbers for (Editable Tables), Search, Sort, and Filter

Oggl for (social connections), Connecting

Oggl for (Toolbox), Toolbox

om finder for (Side Drawer pattern), Side Drawer

OpenTable for (Search forms), Search Forms

Orbitz for (Springboard pattern), Springboard

Over for (Needless Complexity), Needless Complexity

Pages for (User Guides), User Guide/Help System

Pandora for (Error Messages), Feedback Patterns

Pandora for (Persistent Invitations), Persistent Invitations

Paper for (Registration with Personalization), MultiStep

Path for (Bulk Actions), Bulk Actions

Path for (Registration with Personalization), MultiStep

Path for (Registration), Registration

Path for (Side Drawer pattern), Side Drawer

Path for (Springboard pattern), Springboard

PayPal for (MultiStep registration), MultiStep

PayPal for (Table with Visual Indicators), Table with Visual Indicators

Personal Capital for (Integrated Legends), Integrated Legend

Personal Capital for (Overview plus Data), Overview plus Data, Overview plus Data, Overview plus Data

Personal Capital for (Registration with Personalization), MultiStep

Personal Capital for (Sign In PIN), Sign In

Phoster for (tutorial frontloading), Clipchat Compared to Kik

Photos for (Bulk Actions), Bulk Actions

Pinterest for (Contextual Tools), Contextual Tools

Pinterest for (Error Messages), Error Messages

Pinterest for (Retracting Tab design), Emerging Patterns

Pinterest for (Swipe/Flick), Swipe/Flick

Pocket for (How-To), How-Tos

Pocket for (Toggle Menu pattern), Toggle Menu

Polar for (Discoverable Invitations), Discoverable Invitations

Polar for (flat design), Tap

Polar for (learning reinforcement), Rule #4: Reinforce Learning Through Use

Polar for (Registration), Registration

Polar for (System Status), System Status

PortalWebBrowser for (Pie Menu pattern), Pie Menu

Pose for (Following patterns), Following

Pose for (Profiles patterns), Profiles

Potluck for (Swipe/Flick), Swipe/Flick

Priceline for (Sort Forms), Sort Form

Pulse for (Inline Actions), Inline Actions

QuickBooks for (floating CTAs), Inline Actions

qwiki for (Toggle Menu pattern), Toggle Menu

Recipeas for (Gallery pattern), Gallery

Rent the Runway for (Idiot Boxes), Idiot Boxes

Repix for (Toolbox), Toolbox

RetailMeNot for (Coverflow/Rolodex), Swipe/Flick

RetailMeNot for (Implicit Search), Implicit Search

RetailMeNot for (Inline Actions), Inline Actions

RetailMeNot for (Modal Search), Explicit Search

RetailMeNot for (Persistent Invitations), Persistent Invitations

Roambi for (Chart Junk), Chart Junk

Roambi for (Fixed Column Tables), Fixed Column

Roambi for (Interactive Timelines), Interactive Timeline

Roambi for iOS 6 (Pivot Table pattern), Pivot Table

Roambi for iOS 7 (Data Point Details), Data Point Details

Roambi Sales Reports for (Table with Visual Indicators), Table with Visual Indicators

Rove for (floating CTAs), Call to Action Button

Rove for (Registration), Registration

RuLaLa for (Checkout forms), Tip #4: Offer Express Checkout

RunKeeper for (Dashboard pattern), Dashboard

RunKeeper for (Tab Menu pattern), Android

RunKeeper Pro for (CTA Buttons), Call to Action Button

Runtastic's Six Pack for (Help), How-Tos

Safari for (Coverflow/Rolodex), Swipe/Flick

Saks for (Search Results), Search Results/View Results

Sam’s Club for (Search Results), Search Results/View Results

Seed for (Gesture Mismatches), Icon Mismatch

SigFig for (Discoverable Invitations), Discoverable Invitations

SigFig for (System Status), System Status

SigFig for (Zoom pattern), Zoom

Silvercar for (Swipe/Flick), Swipe/Flick

Skype for (Swipe/Flick), Swipe/Flick

Slidestory for (tutorial text), SlideStory Compared to Vine

Smartr for (System Status), System Status

Sniper Ghost Warrior 2 for (Skeuomorphic pattern), Skeuomorphic

Soundwave for (Explicit Search), Explicit Search

Soundwave for (Inline Actions), Inline Actions

Sphere for (floating CTAs), Call to Action Button

Springboard pattern for iOS 7, Springboard

Square for (Command button), Long Forms

Square Register for (Error Messages), Error Messages

Square Wallet for (Gallery pattern), Gallery

Starbucks for (secondary navigation), Secondary Navigation Patterns

Stocks for (Bulk Actions), Bulk Actions

Story Board for (System Status), Affordance

Sugr for (Thresholds pattern), Thresholds

SXSW GO for (Onscreen Filters), Onscreen Filter

Tab Menu pattern for, Gallery

Target for (Checkout forms), Tip #1: Include Sign In, Register, and Guest Options

Target for (Filter Drawers), Filter Drawer

Target for (Modal Search), Explicit Search

Target for (Onscreen Sorts), Sort Patterns

The Nearby for (Inline Actions), Inline Actions

ToDo for (Filter Drawers), Filter Drawer

ToDoist for (tutorial frontloading), Clipchat Compared to Kik

TripAdvisor for (Modal Search), Explicit Search

Trulia for (Springboard design), Springboard

Trulia Mortgage Calculator for (Interactive Preview pattern), Interactive Preview

Trulia Real Estate Calculator for, Calculator Forms

Tumblr for (action buttons), Android

Tumblr for iOS 6 & 7 (CTA Buttons), Call to Action Button

TurboTax SnapTax for (MultiStep registration), MultiStep

Tweetbot 3 for (Gesture Mismatches), Icon Mismatch

TweetCaster for (outdated desktop metaphors), Tools

Twitter for (Contextual Toolbars), Windows Phone

Twitter for (Following patterns), Following

Uber for (Checkout forms), Tip #4: Offer Express Checkout

UltraVisual for (Toggle Menu pattern), Toggle Menu

Vine for (tutorial text), SlideStory Compared to Vine

Vine for (Tutorials), Tutorials

Virtual Makeover for (Toolbox), Toolbox

Visual KPI for (Oceans of Buttons), Oceans of Buttons

Walmart for (Checkout forms), Tip #1: Include Sign In, Register, and Guest Options

Walmart for (Scan & Go), Tip #5: Forget the Web

Walmart for (Tab Menu pattern), iOS

Walmart for iOS 7 (Filter Forms), Filter Form

We Heart It for (Social Registration), MapMyFitness Compared to We Heart It

Weathertron for (charts), Pulling It All Together

Wish for (Side Drawer pattern), Emerging Pattern

Withings for (Data Point Details), Data Point Details

Wunderlist for (Capture Feedback), Capture Feedback

Wunderlist for (Drag action), Drag

Wunderlist for (Sign Up & Log In), Sign In

Wunderlist for (Sort Overlays), Sort Overlay

Yahoo! Finance for (Data Point Details), Data Point Details

Yahoo! for (Zoom pattern), Zoom, Sparklines

Yelp for (Sort Forms), Sort Form

Zappos for (Confirmation patterns), Confirmation

Zappos for (Tips design), Tips

Zillow for (Filter Forms), Filter Form

Zillow for (Implicit Search), Implicit Search

Zillow for (Oceans of Buttons), Oceans of Buttons

Zillow for (Search forms), Search Forms

Zillow Mortgage Calculator for (charts), Charts

Zillow Mortgage Calculator for (Data Point Details), Data Point Details

Zillow Mortgage Calculator for (Interactive Preview pattern), Interactive Preview

Zillow Mortgage Calculator for (real-time data inputs), Calculator Forms

Zillow Mortgage Marketplace for (Side Drawer pattern), Side Drawer

Zite for (Feature Tours), Feature Tours

Zoomingo for (Following patterns), Following

Zulily for (Idiot Boxes), Idiot Boxes

iPad

ABC News for (Mental Model Mismatches), Mental Model Mismatch

early Twitter design for, Primary Navigation Patterns, Persistent

Twitter for (tabs), Emerging Patterns

Yammer for, Emerging Patterns

iTranslate for iOS (Needless Complexity), Needless Complexity

iTunes for iOS

Lock Screen Controls, Lock Screen Controls

Onscreen Filters, Filter Patterns

Scoped Search, Saved, Recent, and Popular Search

## J

Jamie Oliver’s Recipes for iOS (Idiot Boxes), Idiot Boxes

Java desktop applications (Oceans of Buttons), Oceans of Buttons

Jelly (Card pattern), Cards

## K

Kayak for Android (outdated desktop metaphors), Tools

Kayak for iOS (List Menu), List Menu

Kayak for Windows Phone

  Gesture-Based Filters, Gesture-Based Filtering

  Search forms, Search Forms

  Sort Forms, Sort Form

Kik for iOS (Confirmation patterns), Confirmation

Kik Messenger for iOS (tutorial frontloading), Clipchat Compared to Kik

Kindle for Android

  Help, How-Tos

Kindle for iOS (Icon Mismatches), Icon Mismatch

Kobo for Android (Sort Overlays), Sort Overlay

Kobo for iOS (Gamification patterns), Gamification

## L

labels, alignment of (forms), Registration

Launchpad (see Springboard pattern)

Layar for Android (Implicit Search), Implicit Search

lazy loading (Search Results), Search Results/View Results

Leaderboards (Gamification patterns), Gamification

learning through usage (tutorials), Noom Compared to DailyBurn Tracker

LearnVest for iOS

Overview plus Data, Overview plus Data
Registration, Registration
Springboard pattern, Springboard
LemonWallet for iOS
Command button, Long Forms
Drag action, Drag
Level for iOS
Registration, Registration, Registration
System Status, System Status
libraries, charting, Charts
LinkedIn for Android (Springboard design), Springboard
LinkedIn for iOS
Group patterns, Gamification
Inline Actions, Inline Actions
Profiles patterns, Groups
Side Drawer pattern, Side Drawer, Side Drawer
social connections, Connecting
LinkedIn Pulse for Android (Tab Menu pattern), Gallery
List Menu pattern (persistent navigation), List Menu
live tiles, Springboard
loading indicators, System Status, System Status
location-based searching, Saved, Recent, and Popular Search
Lock Screen Controls, Bulk Actions

Long forms, Long Forms

long-press gesture, Contextual Tools, Swipe/Flick

Lootsy for iOS (Drag action), Drag

Lose It for Android

  Dashboard pattern, Dashboard

  Thresholds pattern, Thresholds

Luvocracy for iOS

  Contextual Tools, Contextual Tools

  Side Drawer pattern, Emerging Pattern

## M

Mace, Michael, Help

Mail app (hierarchical structure), List Menu

Mail for iOS

  Confirmation patterns, Confirmation

  Toolbars, Tools

Mailbox (Interactive Tutorial), Feature Tours

Mailbox for iOS (tutorial text), SlideStory Compared to Vine

MapMyFitness for iOS

  social connections, Connecting

  Social Registration, MapMyFitness Compared to We Heart It

Mental Model Mismatches (anti-patterns), Mental Model Mismatch

Metaphor Mismatch (anti-patterns), Metaphor Mismatch

Microsoft Windows 8 (flat design), Affordance

Mint for Android (Grouped Rows tables), Table with Visual Indicators

Mint for iOS

Dashboard pattern, Dashboard, Dashboard

Drill Down pattern, Drill Down

Interactive Preview pattern, Interactive Preview

Tips design, Tips

mobile design, background of, Foreword

mobile PINs as passwords, Sign In

mobile retail innovations (Checkout), Tip #5: Forget the Web

Modal Search (Explicit Search), Explicit Search

Move for Android/iOS (Overview plus Data), Overview plus Data

Multi-State Buttons

Confirmation patterns, Confirmation

use of, Multi-State Button

MultiStep registration (forms), MultiStep

multistep transparency (Snap Payroll case study), Concept #3: MultiStep Transparency

myAppFree for Windows Phone (Tips design), Tips

Myspace for iOS (floating CTAs), Inline Actions

mystery meat navigation (Dwell), Emerging Patterns

## N

Narrato for iOS (Action Sheets), iOS

NASDAQ QMX for iOS (Zoom pattern), Zoom

National Geographic (Search Results), Search Results/View Results

National Geographic Guides for iOS (Swipe/Flick), Swipe/Flick

National Geographic Parks for iOS (primary & secondary patterns), Secondary Navigation Patterns

National Parks for iOS (Filter Overlays), Filter Overlay

navigation, Dashboard

(see also primary navigation patterns; secondary navigation patterns)

navigation bars vs. toolbars, Tools

Navigation Drawers Tab Menu pattern (Android), Android

patterns, Navigation

NBC News (tiles), Springboard

NBC News for iOS (tutorial interactivity), Noom Compared to DailyBurn Tracker

Needless Complexity (anti-patterns), Needless Complexity

Ness for iOS

Filter Overlays, Filter Overlay

Page Swiping, Page Swiping

Registration with Personalization, Registration with Personalization

tutorial text, SlideStory Compared to Vine

Netflix for Android

Auto-Complete, Search with Auto-Complete

Swipe/Flick, Swipe/Flick

Netflix for Windows Phone, Windows Phone

Newegg for Android

Auto-Complete, Search with Auto-Complete

Checkout forms, Tip #4: Offer Express Checkout

Scrolling Tabs, Windows Phone

Newegg for Windows (Sort Overlays), Sort Overlay

News360 for Android (Registration with Personalization), Registration with Personalization

News360 for iOS (Page Swiping), Page Swiping

Newsstand for iOS (Skeuomorphic pattern), Skeuomorphic

Nielsen, Jakob, Affordance

Nike Training Club for Android (Gamification patterns), Gamification

Nike+ FuelBand for iOS (charts), Pivot Table

Nook for Android (Registration), Registration

Noom for Android

Gamification patterns, Gamification

Sparklines pattern, Sparklines

tutorial interactivity, Noom Compared to DailyBurn Tracker

Tutorials, Contextual Help

North Face for iOS (Checkout forms), Checkout

Notif for Android (Command Button), Tables

Novel Notions (anti-patterns), Novel Notions

NPR for Android (Toggle Menu pattern), Toggle Menu

Nudelman, Greg, Idiot Boxes, Square Peg, Round Hole

Numbers for iOS (Editable Tables), Search, Sort, and Filter

NYTimes for Android

Confirmation patterns, Confirmation

Toggle Menu pattern, Toggle Menu

## O

Oceans of Buttons (anti-patterns), Oceans of Buttons

Office for Windows Phone (Editable Tables), Editable Table

Oggl for iOS

social connections, Connecting

Toolbox, Toolbox

Oggl for Windows Phone (Toolbox), Toolbox

Oliver, Jamie, Idiot Boxes

OLX for Windows Phone (Long Forms), Long Forms

om finder for iOS (Side Drawer pattern), Side Drawer

OneNote for Windows (Contextual Toolbars), Windows Phone

OneNote for Windows Phone (tutorial frontloading), Clipchat Compared to Kik

online resources, Additional Resources

Onscreen Filter patterns, Filter Patterns

Onscreen Sort patterns, Sort Patterns

OOKLA for Android (Chart Junk), Chart Junk

OpenTable for Android/iOS (Search forms), Search Forms

Orbitz for iOS (Springboard pattern), Springboard

Over for iOS (Needless Complexity), Needless Complexity

overlay Side Drawers style, Skeuomorphic

Overview plus Data pattern

charts, Drill Down

tables, Overview plus Data

## P

Page Swiping (secondary navigation), Page Swiping

Pageonce for Android (Registration feedback), Registration

Pages for iOS (User Guides), User Guide/Help System

Palm webOS (Card pattern), Cards

Pandora for iOS

Error Messages, Feedback Patterns

Persistent Invitations, Persistent Invitations

panorama control (Windows phone), Primary Navigation Patterns, Persistent

Paper for iOS (Registration with Personalization), MultiStep

Paranoid Android OS, Pie Menu

passwords

Confirm Password fields (forms), Registration

requiring up front (forms), Registration

Sign In forms option, Sign In

Path for Android

Bulk Actions, Bulk Actions

floating CTAs, Call to Action Button

Path for iOS

Bulk Actions, Bulk Actions

Registration, Registration

Registration with Personalization, MultiStep

Side Drawer pattern, Springboard, Side Drawer

Path, Fixed Tabs pattern in, Android

PayPal Beacon (Bluetooth purchases), Calculator Forms

PayPal for Android (Feature Tours), Feature Tours

PayPal for iOS

MultiStep registration, MultiStep

Table with Visual Indicators, Table with Visual Indicators

Persistent Details button (Audi Configurator), Checkout

Persistent Invitations, Persistent Invitations

persistent navigation

Card pattern, Cards

Dashboard pattern, Dashboard

Gallery pattern, Dashboard

List Menu pattern, List Menu

Skeuomorphic pattern, Emerging Patterns

Springboard pattern, Primary Navigation Patterns, Persistent

Tab Menu pattern (see Tab Menu pattern (persistent navigation))

vs. transient navigation, Navigation

Personal Capital for iOS

Integrated Legends pattern, Integrated Legend

Overview plus Data, Overview plus Data, Overview plus Data, Overview plus Data

Registration with Personalization, MultiStep

Sign In PIN, Sign In

phonealytics for Windows (Chart Junk), Chart Junk

Phoster for iOS (tutorial frontloading), Clipchat Compared to Kik

Photos for iOS (Bulk Actions), Bulk Actions

pie charts, Chart Junk

Pie Menu pattern (transient navigation), Pie Menu

Pinterest (Card pattern), Cards

Pinterest for Android (Persistent Invitations), Persistent Invitations

Pinterest for iOS

  Contextual Tools, Contextual Tools

  Error Messages, Error Messages

  Retracting Tab design, Emerging Patterns

  Swipe/Flick, Swipe/Flick

pivot controls, Windows Phone, Sort Form

pivot headers (Windows), Onscreen Filter

Pivot Table pattern (charts), Thresholds

Pixlr for Android (Toolbox), Toolbox

Play Store for Android

  Expand/Collapse Panel, Accordion

  Onscreen Filters, Onscreen Filter, Onscreen Filter

Plume for Android (Oceans of Buttons), Oceans of Buttons

Pocket for Android (Onscreen Filters), Onscreen Filter

Pocket for iOS

  How-To, How-Tos

Toggle Menu pattern, Toggle Menu

Tutorials, Tutorials

Polar for iOS

Discoverable Invitations, Discoverable Invitations

flat design, Tap

learning reinforcement, Rule #4: Reinforce Learning Through Use

System Status, System Status

Popular Search patterns, Saved, Recent, and Popular Search

Portal game, Tutorial Rules

PortalWebBrowser for iOS (Pie Menu pattern), Pie Menu

Pose for iOS

Following patterns, Following

Profiles patterns, Profiles

Potluck (Card pattern), Cards

Potluck for iOS (Swipe/Flick), Swipe/Flick

Priceline for Android (MultiStep registration), MultiStep

Priceline for iOS (Sort Forms), Sort Form

primary navigation patterns

persistent navigation (see persistent navigation)

transient navigation (see transient navigation)

Principles for Usable Design, Feedback and Affordance

product configurators (MultiStep forms), MultiStep

Profiles patterns (social networking), Profiles

progress meters (Confirmation patterns), Confirmation

Pulse for iOS (Inline Actions), Inline Actions

## Q

QR code scanning, Calculator Forms

QuickBooks for iOS (floating CTAs), Inline Actions

Quora (Fixed Tabs pattern), Android

qwiki for iOS (Toggle Menu pattern), Toggle Menu

## R

Raskin, Aza, Springboard

Realtor.com for Android (Headerless Tables), Headerless Table

Realtor.com for Windows Phone (Sort Forms), Sort Form

Recent Search patterns, Saved, Recent, and Popular Search

Recipeas for iOS (Gallery pattern), Gallery

Redfin for Android

Headerless Tables, Fixed Column

Oceans of Buttons, Oceans of Buttons

RedLaser for Android (Explicit Search), Explicit Search

Register option (Checkout forms), Tip #1: Include Sign In, Register, and Guest Options

Registration form patterns, Registration

(see also Social Registration)

Registration with Personalization (forms), Registration with Personalization

Remember The Milk for Android

Registration labels, Registration

Sign In, Forms

Rent the Runway for iOS (Idiot Boxes), Idiot Boxes

Repix for iOS (Toolbox), Toolbox

reputation points (Gamification patterns), Gamification

resources, online, Additional Resources

RetailMeNot (Confirmation patterns), Confirmation

RetailMeNot for Android (Side Drawer pattern), Skeuomorphic

RetailMeNot for iOS

Coverflow/Rolodex, Swipe/Flick

Implicit Search, Implicit Search

Inline Actions, Inline Actions

Modal Search, Explicit Search

Persistent Invitations, Persistent Invitations

Retracting Tab design, Emerging Patterns

rewarding with interactivity (tutorials), Clipchat Compared to Kik

Roambi (Drill Down chart), Overview plus Data

Roambi for iOS

Chart Junk, Chart Junk

Data Point Details, Data Point Details

Fixed Column Tables, Fixed Column

Interactive Timelines, Interactive Timeline

Pivot Table pattern, Pivot Table

Sales Reports (Table with Visual Indicators), Table with Visual Indicators

Rolodex (Affordance), Swipe/Flick

Rove for iOS

floating CTAs, Call to Action Button

Registration, Registration

RuLaLa for iOS (Checkout forms), Tip #4: Offer Express Checkout

RunKeeper for iOS

Dashboard pattern, Dashboard

Tab Menu pattern, Android

RunKeeper Pro for iOS (CTA Buttons), Call to Action Button

Runtastic Heart Rate for Android (Overview plus Data tables), Overview plus Data

Runtastic Six Pack for iOS (Help), How-Tos

## S

Safari for iOS (Coverflow/Rolodex), Swipe/Flick

Saks for iOS (Search Results), Search Results/View Results

Sam’s Club for iOS (Search Results), Search Results/View Results

Saved Search patterns, Saved, Recent, and Popular Search

Scan & Go feature (Walmart), Tip #5: Forget the Web

Scan Card feature (Checkout forms), Tip #2: Streamline the Flow

Scoped Search patterns, Scoped Search

Scott, Bill, Idiot Boxes, Oceans of Buttons

scrollbars in mobile apps, Swipe/Flick

Scrolling Tabs, Android, Scrolling Tabs, Onscreen Filter

Scrolling Window pattern, Interactive Timeline

search forms, Search Forms, Saved, Recent, and Popular Search
Search patterns
Dynamic Search, Search with Auto-Complete
Explicit Search, Explicit Search
Implicit Search, Search, Sort, and Filter
overview, Search, Sort, and Filter
Saved/Recent/Popular Searches, Saved, Recent, and Popular Search
Scoped Search, Scoped Search
Search Form, Saved, Recent, and Popular Search
Search Results, Search Results/View Results
View Results, Search Results/View Results
secondary navigation patterns
Accordion pattern, Scrolling Tabs
Expand/Collapse Panel pattern, Scrolling Tabs
overview, Secondary Navigation Patterns
Page Swiping, Page Swiping
Scrolling Tabs, Scrolling Tabs
using primary navigation patterns as, Secondary Navigation Patterns
secure checkout process (Home Depot for iOS, MultiStep
Seed for iOS (Gesture Mismatches), Icon Mismatch
Settings app (hierarchical structure), List Menu
ShopBeacon (Bluetooth purchases), Calculator Forms
Shopular (Needless Complexity), Needless Complexity

showrooming phenomenon, Explicit Search

Side Drawer pattern, Primary Navigation Patterns, Persistent, Springboard, Skeuomorphic

sidebars in web apps, Emerging Patterns

SigFig for iOS

  Discoverable Invitations, Discoverable Invitations

  System Status, System Status

  Zoom pattern, Zoom

SigFig for Windows Phone (Sign In/Sign Up tabs), Sign In

Sign In forms, Forms

Sign In option (Checkout forms), Tip #1: Include Sign In, Register, and Guest Options

Sign-up/Sign-in forms anti-pattern, Square Peg, Round Hole

Silvercar for Android (Capture Feedback), Capture Feedback

Silvercar for iOS (Swipe/Flick), Swipe/Flick

single-step transparency (Snap Payroll case study), Concept #3: MultiStep Transparency

skeleton screens, System Status

Skeuomorphic pattern (persistent navigation), Emerging Patterns

Skype for iOS/Android (Swipe/Flick), Swipe/Flick

Skyscanner for Windows Phone

  Filter Forms, Filter Form

  Search Forms, Saved, Recent, and Popular Search

Sleep Charts for Android (Table with Visual Indicators), Table with Visual Indicators

Slidestory for iOS (tutorial text), SlideStory Compared to Vine

Smartr for iOS (System Status), System Status

Smith, Matt, Registration

Sniper Ghost Warrior 2 for iOS/Android (Skeuomorphic pattern), Skeuomorphic

SnipSnap for Android

  Help, How-Tos

  MultiStep registration, MultiStep

Social patterns

  Connecting, MapMyFitness Compared to We Heart It

  Following, Following

  Gamification, Gamification

  Groups, Groups

  overview, Social Patterns

  Profiles, Profiles

  Social Registration, Social Patterns

Social Sign In path (forms), Sign In

Songza for Android

  Auto-Complete, Search with Auto-Complete

  Novel Notions, Novel Notions

  ScrollingTabs, Scrolling Tabs

Sort patterns

  Onscreen Sort, Sort Patterns

  overview, Search Results/View Results

  Sort Form, Sort Form

Sort Overlay, Sort Overlay

Soundwave for Android

Error Messages, Feedback Patterns

Novel Notions, Novel Notions

Soundwave for iOS

Explicit Search, Explicit Search

Inline Actions, Inline Actions

Sparkline patterns

charts, Sparklines

Roambi Sales Reports for iOS, Table with Visual Indicators

Sphere for iOS (floating CTAs), Call to Action Button

Spinner controls (Android)

Avoid the Spinner (article), System Status

filtering lists with (Android), Onscreen Filter

Google Play Store, Scoped Search

primary navigation, Toggle Menu

selecting sort order with, Sort Patterns

Spinners Tab Menu pattern (Android), Android

Spool, Jared, Sign In

spreadsheets (Editable Table pattern), Table with Visual Indicators

Springboard pattern (persistent navigation), Primary Navigation Patterns, Persistent

Square for iOS (Command button), Long Forms

Square Peg, Round Hole anti-pattern, Square Peg, Round Hole

Square Register for iOS (Error Messages), Error Messages

Square Wallet for iOS (Gallery pattern), Gallery

stacked Cards (Potluck), Swipe/Flick

Starbucks for iOS (secondary navigation), Secondary Navigation Patterns

Starbucks.com (Toggle Menu pattern), Toggle Menu

static tiles, Springboard

Stock Trainer for Android (tables), Tables

StockPlus for Windows Phone (Chart with Filters), Charts

Stocks for iOS (Bulk Actions), Bulk Actions

Story Board for iOS (System Status), Affordance

streamlining flow of Checkout forms, Tip #2: Streamline the Flow

Sugr for iOS (Thresholds pattern), Thresholds

swipe-to-perform gestures, Confirmation

Swipe/Flick patterns, Tap

SXSW GO for iOS (Onscreen Filters), Onscreen Filter

Symbian, Yahoo! for (Springboard design), Springboard

System Status patterns, Confirmation

## T

Tab Menu pattern (persistent navigation)

  Configurable Tabs design, Emerging Patterns

  for Android, Android

  for iOS, Gallery

  for Windows Phone, Windows Phone

overview, Gallery

Retracting Tab design, Emerging Patterns

tables

Basic pattern, Tables

Editable Tables pattern, Table with Visual Indicators

Fixed Column pattern, Fixed Column

Grouped Rows pattern, Overview plus Data

Headerless pattern, Headerless Table

overview, Tables

Overview plus Data pattern, Overview plus Data

Table with Visual Indicators pattern, Table with Visual Indicators

Tan, Chui Chui, Registration

Tap patterns, Affordance

Tapworthy: Designing Great iPhone Apps, Search, Sort, and Filter

Target for Android (Auto-Complete), Search with Auto-Complete

Target for iOS

Checkout forms, Tip #1: Include Sign In, Register, and Guest Options

Filter Drawers, Filter Drawer

Modal Search, Explicit Search

Onscreen Sorts, Sort Patterns

Target mobile purchases, Calculator Forms

TaxCaster for Android (Calculator forms), Calculator Forms

TaxCaster for Windows Phone (Interactive Preview pattern), Interactive Preview

testing charts, Charts

text, economizing (tutorials), Rule #1: Use Less Text

The Nearby for iOS (Inline Actions), Inline Actions

Thresholds pattern (charts), Integrated Legend

tiles, live and static, Springboard

time-saving shortcuts (Checkout forms), Tip #2: Streamline the Flow

Tips design

  invitations, Invitation Patterns

  Snap Payroll case study, Concept #4: Single-Step Transparency

toast messages (Confirmation patterns), Confirmation

ToDo for iOS (Filter Drawers), Filter Drawer

ToDoist for iOS (tutorial frontloading), Clipchat Compared to Kik

Toggle Menu pattern (transient navigation), Toggle Menu

Toolbars

  Android Action Bar, Android

  Contextual, Windows Phone

  iOS, Tools

  Toolbar in Tab Menu pattern, iOS

  Windows Phone Application Bar, Android

Toolbox pattern, Windows Phone

tools

  Bulk Actions, Bulk Actions

  Call to Action (CTA) Buttons, Call to Action Button

Contextual tools, Contextual Tools

Inline Actions, Inline Actions

Lock Screen Controls, Bulk Actions

Multi-State Buttons, Multi-State Button

overview, Tools

Toolbar (see Toolbars)

Toolbox, Windows Phone

Toshl for Windows Phone (Thresholds pattern), Thresholds

Tour (invitations case study), Concept #2: Tour

Traffic Google Analytics (Data Point Details), Data Point Details

transient navigation

overview, Skeuomorphic

Pie Menu pattern, Pie Menu

Side Drawer pattern, Skeuomorphic

Toggle Menu pattern, Toggle Menu

vs. persistent navigation, Navigation

Transparency pattern (Snap Payroll case study), Concept #3: MultiStep Transparency

Travelocity for Windows Phone (pivot control), Sort Form

TripAdvisor for iOS (Modal Search), Explicit Search

Trulia for Android

charts, Charts

Explicit Search, Explicit Search

Sort Overlays, Sort Overlay

Trulia for iOS (Springboard design), Springboard

Trulia Mortgage Calculator for iOS (Interactive Preview pattern), Interactive Preview

Trulia Real Estate Calculator for iOS, Calculator Forms

Tufte, Edward, Sparklines

Tumblr for iOS

action buttons, Android

CTA Buttons, Call to Action Button

TuneIn for Android (ScrollingTabs), Scrolling Tabs

TurboTax SnapTax for Android/iOS (MultiStep registration), MultiStep

tutorial rules

avoiding frontloading, SlideStory Compared to Vine

design tips from Extra Credits, Rule #5: Listen to Your Users

economizing text, Rule #1: Use Less Text

learning through usage, Noom Compared to DailyBurn Tracker

listening to users, Rule #5: Listen to Your Users

overview, Tutorials and Invitations

rewarding with interactivity, Clipchat Compared to Kik

Tutorials (Help patterns), Tutorials

TweetBot 3 for iOS (Gesture Mismatches), Icon Mismatch

TweetBot for iOS (Oceans of Buttons), Oceans of Buttons

TweetCaster for iOS (outdated desktop metaphors), Tools

Twitter @music (Novel Notions), Novel Notions

Twitter for iOS

Contextual Toolbars, Windows Phone

Following patterns, Following

Twitter for iPad

early design, Primary Navigation Patterns, Persistent

labeled tabs, Emerging Patterns

## U

Uber for Android (Sign In), Sign In

Uber for iOS (Checkout forms), Tip #4: Offer Express Checkout

Ubuntu Touch, Primary Navigation Patterns, Persistent

UltraVisual for iOS (Toggle Menu pattern), Toggle Menu

Underwood, Jay, Charts

unmasked password field (Sign In forms), Sign In

Up button (List Menu in Android), List Menu

USA Today for Android (Novel Notions), Novel Notions

user authentication, Sign In

User Experience Professionals Association, Feedback and Affordance

User Guides/Help Systems, How-Tos

users, listening to (tutorials), Rule #5: Listen to Your Users

## V

Valspar Paint Calculator, Calculator Forms

View Search Results patterns, Search Results/View Results

Vimeo for Android (Springboard pattern), Springboard

Vine for Android (simple share forms), Tables

Vine for iOS

tutorial text, SlideStory Compared to Vine

tutorials, Tutorials

Virtual Makeover for iOS (Toolbox), Toolbox

Virtual Toy Store (QR scanning), Calculator Forms

Visual Basic applications (Oceans of Buttons), Oceans of Buttons

Visual Indicators pattern (tables), Table with Visual Indicators

Visual KPI for iOS (Oceans of Buttons), Oceans of Buttons

voice-based searches, Explicit Search

## W

The Wall Street Journal Guide to Information Graphics, Charts

Walmart for Android (Toggle Menu pattern), Toggle Menu

Walmart for iOS

Checkout forms, Tip #1: Include Sign In, Register, and Guest Options

Filter Forms, Filter Form

Scan & Go feature, Tip #5: Forget the Web

Tab Menu pattern, iOS

watermark labels, Registration

Waze (in-app Help), Help

We Heart It for iOS (Social Registration), MapMyFitness Compared to We Heart It

Weathertron for iOS (charts), Pulling It All Together

websites for further information

About Face 3: The Essentials of Interaction Design, Idiot Boxes

Allthecooks for Android (animation), Swipe/Flick

Amazon for iOS (Confirmation patterns), Confirmation

Android design guidelines, Side Drawer

Android Jelly Bean interface, Android

Android Tab Menu patterns, Android

Apple hierarchical navigation, List Menu

Application Bar (Windows Phone), Android

Argus for iOS (Contextual Help), Capture Feedback

Avoid the Spinner (article), System Status

Buttons Are a Hack campaign, Tools

CBS Outdoors, Calculator Forms

charting libraries, Charts

Clear for Android (tutorial text), SlideStory Compared to Vine

Dark Sky for iOS, Data Point Details, Drill Down

Data Visualization Best Practices, Charts

Designing for Thumb Flow (article), Registration with Personalization

Drink Builder on iOS, Checkout

emerging Side Drawer style, Emerging Pattern

Facebook navigation testing, Gallery

Fantastical for iOS (tutorial text), SlideStory Compared to Vine

Filter Drawers, Filter Drawer

Firefox Mobile, Springboard

Flipboard for Android (tutorial interactivity), Noom Compared to DailyBurn Tracker

Flipboard for iOS (animation), Swipe/Flick

Float Labels, Registration

Foodspotting reputation system, Gamification

form designs, Forms

Gap for Android/iOS (Square Peg, Round Hole), Square Peg, Round Hole

inlay Side Drawer pattern, Emerging Pattern

iOS 7 design guidelines, Long Forms

iOS Design Guide, Emerging Pattern

iOS Human Interface Guideline, Tools

Jelly for iOS, Cards

Mailbox (Interactive Tutorial), Feature Tours

Mailbox for iOS (tutorial text), SlideStory Compared to Vine

mystery meat navigation (Dwell), Emerging Patterns

Over for iOS (Needless Complexity), Metaphor Mismatch

Paper for iOS, MultiStep

PayPal Beacon (Bluetooth purchases), Calculator Forms

Phone Gap, Square Peg, Round Hole

PIE menus, Pie Menu

Pinterest for iOS (Contextual Tools), Contextual Tools

Pivot control (App Tabs), Windows Phone

Potluck for iOS, Cards, Swipe/Flick

Principles for Usable Design, Feedback and Affordance

RetailMeNot (Confirmation patterns), Confirmation

Retracting Tab design, Emerging Patterns

Roambi for iOS, Data Point Details, Chart Junk

ShopBeacon (Bluetooth purchases), Calculator Forms

showrooming phenomenon, Explicit Search

The $300 Million Button (article), Sign In

Traffic Google Analytics, Data Point Details

Tutorials, Tables, Tutorial Rules

Vine for iOS (tutorial text), SlideStory Compared to Vine

Virtual Toy Store QR scanning, Calculator Forms

Windows Design Guide, Springboard

Windows Phone 8 design guidelines, Long Forms

Xamarin, Square Peg, Round Hole

Zappos for iOS (Confirmation patterns), Confirmation

WhySiriWhy.com, Explicit Search

widgets, Bulk Actions

Wikitude World Browser (Implicit Search), Implicit Search

Windows

Creative Studio for (tutorial frontloading), Clipchat Compared to Kik

Design Guide, Springboard

Google for (Contextual Help), Capture Feedback

Newegg for (Sort Overlays), Sort Overlay

OneNote for (Contextual Toolbars), Windows Phone

phonealytics for (Chart Junk), Chart Junk

Windows Phone

4th & Mayor for (Application Bars), Windows Phone

Allrecipes for (Filter Forms), Filter Form

Amazon for (Application Bars), Android

American Airlines for (Search forms), Search Forms

Application Bar, Android

Audible for (Gamification patterns), Gamification

BBC Radio for (Springboard pattern), Springboard

Brit & Co. for (secondary navigation), Secondary Navigation Patterns

Creative Studio for (Contextual Help), Capture Feedback

design guide, Windows Phone

Dictionary.com for (Explicit Search), Explicit Search

eBay for (Explicit Search), Explicit Search

Evernote for (tiles), Springboard

Facebook beta for (Side Drawer pattern), Side Drawer

Fitbit for (browser-based Registration), Registration

Google for (Scoped Search), Scoped Search

Home Depot for (Toggle Menu pattern), Toggle Menu

Hungry Now for (Implicit Search), Implicit Search

Innerfence for (Long Forms), Long Forms

Kayak for (Gesture-Based Filters), Gesture-Based Filtering

Kayak for (Search forms), Search Forms

Kayak for (Sort Forms), Sort Form

myAppFree for (Tips design), Tips

Netflix for, Windows Phone

Office for (Editable Tables), Editable Table

Oggl for (Toolbox), Toolbox

OLX for (Long Forms), Long Forms

OneNote for (tutorial frontloading), Clipchat Compared to Kik

Realtor.com for (Sort Forms), Sort Form

SigFig for (Sign In/Sign Up tabs), Sign In

Skyscanner for (Filter Forms), Filter Form

Skyscanner for (Search Forms), Saved, Recent, and Popular Search

StockPlus for (Chart with Filters), Charts

Tab Menu pattern for, Windows Phone

TaxCaster for (Interactive Preview pattern), Interactive Preview

tiles, Springboard

Toshl for (Thresholds pattern), Thresholds

Travelocity for (pivot control), Sort Form

Windows Phone 8, Primary Navigation Patterns, Persistent, Springboard, Long Forms

Yelp for (Sort Forms), Filter Patterns

Wish for iOS (Side Drawer pattern), Emerging Pattern

Withings for Android (Dashboard pattern), Dashboard

Withings for iOS (Data Point Details), Data Point Details

Wroblewski, Luke, Registration, System Status

Wunderlist for iOS

Capture Feedback, Capture Feedback

Drag action, Drag

Sign Up & Log In, Sign In

Sort Overlays, Sort Overlay

## Y

Yahoo! Finance for iOS (Data Point Details), Data Point Details

Yahoo! for iOS (Zoom pattern), Zoom, Sparklines

Yahoo! for Symbian (Springboard design), Springboard

Yammer for iPad, Emerging Patterns

Yelp for iOS (Sort Forms), Sort Form

Yelp for Windows Phone (Sort Forms), Filter Patterns

## Z

Zappos for iOS

Confirmation patterns, Confirmation

Tips design, Tips

Zillow for iOS

Filter Forms, Filter Form

Implicit Search, Implicit Search

Oceans of Buttons, Oceans of Buttons

Search forms, Search Forms

Zillow Mortgage Calculator for iOS

charts, Charts

Data Point Details, Data Point Details

Grouped Rows tables, Overview plus Data

Interactive Preview pattern, Interactive Preview

real-time data inputs, Calculator Forms

Zillow Mortgage Marketplace for iOS (Side Drawer pattern), Side Drawer

Zite for iOS (Feature Tours), Feature Tours

Zoom pattern (charts), Zoom

Zoomingo for iOS (Following patterns), Following

Zulily for iOS (Idiot Boxes), Idiot Boxes

## About the Author

**Theresa Neil** is a user experience consultant in Austin, Texas, where she designs rich applications for start-ups and Fortune500 companies.

# Special Upgrade Offer

If you purchased this ebook from a retailer other than O’Reilly, you can upgrade it for $4.99 at oreilly.com by clicking here.

# Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps

Theresa Neil

Editor
Mary Treseler



**Mobile Design Pattern Gallery, Second Edition**

by Theresa Neil



**Editor:** Mary Treseler
**Production Editor:** Kara Ebrahim
**Copyeditor:** Rachel Monaghan
**Proofreader:** Rachel Head
**Indexer:** Ron Strauss
**Cover Designer:** Randy Comer
**Interior Designers:** Ron Bilodeau and Monica Kamsvaag
**Illustrator:** Rebecca Demarest
**Compositor:** Kara Ebrahim

May 2014: First Edition.

**Revision History for the First Edition:**

2014-04-15 First release

See *http://www.oreilly.com/catalog/errata.csp?isbn=0636920029311* for release details.



[T]

O’Reilly Media
1005 Gravenstein Highway North

Sebastopol, CA 95472

2014-04-22T11:51:37-07:00

# Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps

Table of Contents


- Special Upgrade Offer
- Foreword
- Preface
  - Intended Audience for This Book
  - Safari® Books Online
  - How to Contact Us
  - Acknowledgments
- 1. Navigation
  - Primary Navigation Patterns, Persistent
    - Springboard
    - Cards
    - List Menu
    - Dashboard
    - Gallery
    - Tab Menu
      - iOS
      - Android
      - Windows Phone
      - Emerging Patterns
    - Skeuomorphic
  - Primary Navigation Patterns, Transient
    - Side Drawer
      - Emerging Pattern
    - Toggle Menu
    - Pie Menu
  - Secondary Navigation Patterns
    - Page Swiping
    - Scrolling Tabs

- Accordion
- 2. Forms
  - Sign In
  - Registration
  - Registration with Personalization
  - Multi-Step
  - Checkout
    - Tip #1: Include Sign In, Register, and Guest Options
    - Tip #2: Streamline the Flow
    - Tip #3: Provide Time-Saving Shortcuts
    - Tip #4: Offer Express Checkout
    - Tip #5: Forget the Web
  - Calculator Forms
  - Search Forms
  - Long Forms
- 3. Tables
  - Basic Table
  - Headerless Table
  - Fixed Column
  - Overview plus Data
  - Grouped Rows
  - Table with Visual Indicators
  - Editable Table
- 4. Search, Sort, and Filter
  - Search Patterns
    - Implicit Search
    - Explicit Search
    - Search with Auto-Complete
    - Dynamic Search
    - Scoped Search
    - Saved, Recent, and Popular Search
    - Search Form

- Search Results/View Results
- Sort Patterns
  - Onscreen Sort
  - Sort Overlay
  - Sort Form
- Filter Patterns
  - Onscreen Filter
  - Filter Overlay
  - Filter Form
  - Filter Drawer
  - Gesture-Based Filtering

5. Tools
- Toolbar
  - iOS
  - Android
  - Windows Phone
  - OS-Neutral Pattern: Contextual Toolbar
- Toolbox
- Call to Action Button
- Inline Actions
- Multi-State Button
- Contextual Tools
- Bulk Actions
- Lock Screen Controls

6. Charts
- Chart with Filters
- Interactive Timeline
- Data Point Details
- Drill Down
- Overview plus Data
- Interactive Preview
- Dashboard

Zoom
Sparklines
Integrated Legend
Thresholds
Pivot Table
Pulling It All Together
7. Tutorials and Invitations
Tutorial Rules
Rule #1: Use Less Text
Ness Compared to Foodspotting
Boomerang Compared to Mailbox
DigiCal Compared to Fantastical
Catch Compared to Clear
SlideStory Compared to Vine
Rule #2: No Frontloading
Phoster Compared to Creative Studio
Dooo Compared to Todoist
Buy Me a Pie! Compared to OneNote
Clipchat Compared to Kik
Rule #3: Make It Rewarding
NBC News Compared to Flipboard
Noom Compared to DailyBurn Tracker
Rule #4: Reinforce Learning Through Use
Rule #5: Listen to Your Users
Invitation Patterns
Tips
Persistent Invitations
Discoverable Invitations
Chapter Extra: Invitations — Rolling Out the Welcome Mat
Iterating on the Welcome Experience
Concept #1: Direct Immersion
Concept #2: Tour

Concept #3: Multi-Step Transparency
Concept #4: Single-Step Transparency
Concept #5: Tip
Summary
8. Social Patterns
Social Registration
MapMyFitness Compared to We Heart It
Connecting
Following
Profiles
Groups
Gamification
9. Feedback and Affordance
Feedback Patterns
Error Messages
Confirmation
System Status
Affordance
Tap
Swipe/Flick
Drag
10. Help
How-Tos
User Guide/Help System
FAQs
Feature Tours
Tutorials
Contextual Help
Capture Feedback
11. Anti-Patterns
Novel Notions
Needless Complexity

- Metaphor Mismatch
  - Control Mismatch
  - Icon Mismatch
  - Gesture Mismatch
  - Mental Model Mismatch
- Idiot Boxes
- Chart Junk
- Oceans of Buttons
- Square Peg, Round Hole
- Chapter Extra: Let Them Pee — Avoiding the Sign-Up/Sign-In Mobile Anti-Pattern

A. Additional Resources

Index

About the Author

Special Upgrade Offer

Copyright