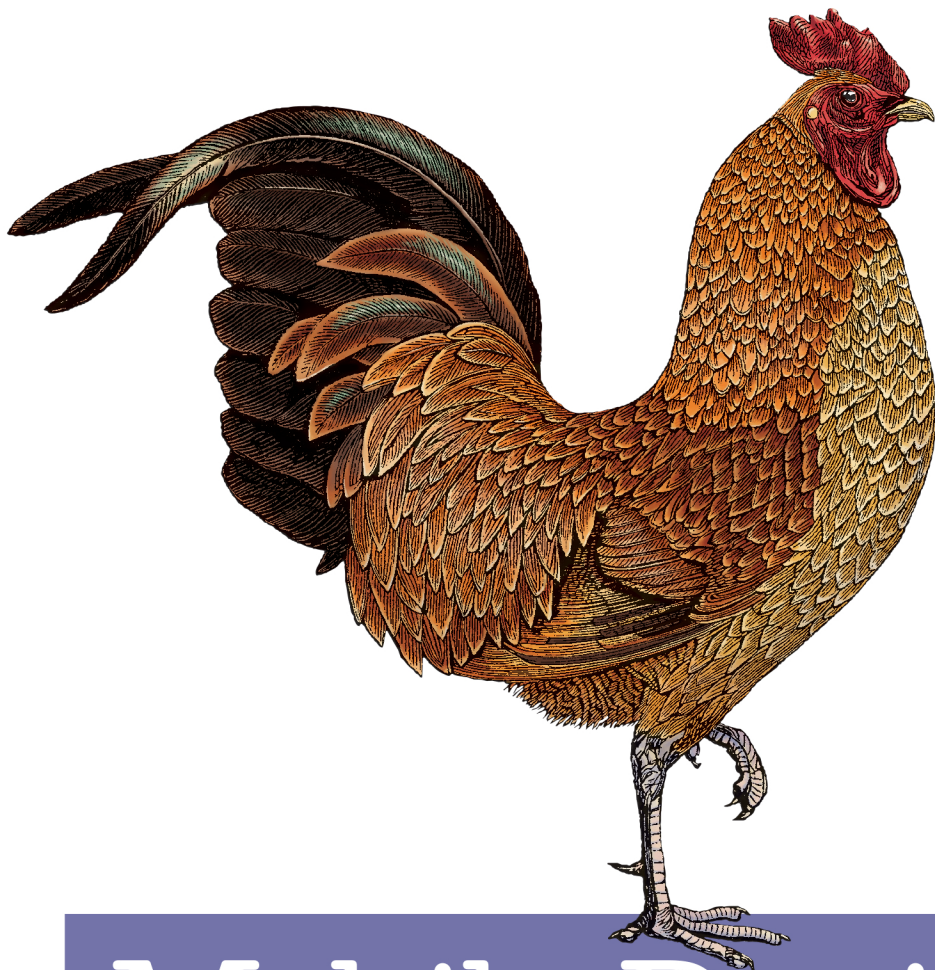


O'REILLY®

2nd Edition



# Mobile Design Pattern Gallery

UI PATTERNS FOR SMARTPHONE APPS

Theresa Neil

*Foreword by Jenifer Tidwell*

# Mobile Design Pattern Gallery

When you're under pressure to produce a well-designed, easy-to-navigate mobile app, there's no time to reinvent the wheel—and no need to. This handy reference provides more than 90 mobile app design patterns, illustrated by 1,000 screenshots from current Android, iOS, and Windows Phone apps.

Much has changed since this book's first edition. Mobile OSs have become increasingly different, driving their own design conventions and patterns, and many designers have embraced mobile-centric thinking. In this edition, user experience professional Theresa Neil walks product managers, designers, and developers through design patterns in 11 categories:

- **Navigation:** get patterns for primary and secondary navigation
- **Forms:** break industry-wide habits of bad form design
- **Tables:** display only the most important information
- **Search, sort, and filter:** make these functions easy to use
- **Tools:** create the illusion of direct interaction
- **Charts:** learn best practices for basic chart design
- **Tutorials & Invitations:** invite users to get started and discover features
- **Social:** help users connect and become part of the group
- **Feedback & Accordance:** provide users with timely feedback
- **Help:** integrate help pages into a smaller form factor
- **Anti-Patterns:** what not to do when designing a mobile app

**Theresa Neil** is an internationally recognized design expert who is passionate about making products that look good and work well. Her books and talks have helped thousands of IT professionals advance their design skills and create better user experiences.

“Mobile design evolves at a fast pace, so it's great to see a new edition of Theresa Neil's essential book. The initial set of patterns has proven remarkably resilient, as patterns should, and updates and improvements make the book both useful and timely. I can't wait to share the on-boarding chapter with my team.”

—**Christian Crumlish**  
Director of Product Management,  
CloudOn

MOBILE DESIGN / USER EXPERIENCE

US \$49.99

CAN \$52.99

ISBN: 978-1-449-36363-5



Twitter: @oreillymedia  
facebook.com/oreilly

# Mobile Design Pattern Gallery

Second Edition

*UI Patterns for Smartphone Apps*

Theresa Neil

**O'REILLY®**

Beijing • Boston • Farnham • Sebastopol • Tokyo

## Mobile Design Pattern Gallery, Second Edition

by Theresa Neil

Copyright © 2014 Theresa Neil. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc.,  
1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*oreilly.com/safari*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

**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: Second Edition.

### Revision History for the Second Edition:

2014-04-15	First release
2014-06-06	Second release
2016-12-22	Third release

See <http://www.oreilly.com/catalog/errata.csp?isbn=0636920029311> for release details.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *Mobile Design Pattern Gallery, Second Edition* and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

Although the publisher and author have used reasonable care in preparing this book, the information it contains is distributed “as is” and without warranties of any kind. This book is not intended as legal or financial advice, and not all of the recommendations may be suitable for your situation. Professional legal and financial advisors should be consulted, as needed. Neither the publisher nor the author shall be liable for any costs, expenses, or damages resulting from use of or reliance on the information contained in this book.

ISBN: 978-1-4493-6363-5

[LSI]

# [ Contents ]

	<i>Foreword</i> .....	ix
	<i>Preface</i> .....	xiii
<b>Chapter 1</b>	<b>Navigation</b> .....	1
	Primary Navigation Patterns, Persistent .....	2
	Springboard .....	7
	Cards .....	12
	List Menu .....	14
	Dashboard .....	17
	Gallery .....	17
	Tab Menu .....	19
	Skeuomorphic .....	26
	Primary Navigation Patterns, Transient .....	30
	Side Drawer .....	30
	Toggle Menu .....	41
	Pie Menu .....	44
	Secondary Navigation Patterns .....	46
	Page Swiping .....	48
	Scrolling Tabs .....	51
<b>Chapter 2</b>	<b>Forms</b> .....	55
	Sign In .....	57
	Multi-Step .....	75
	Checkout .....	82
	Tip #1: Include Sign In, Register, and Guest Options .....	83
	Tip #2: Streamline the Flow .....	84

	Tip #3: Provide Time-Saving Shortcuts . . . . .	84
	Tip #4: Offer Express Checkout . . . . .	86
	Tip #5: Forget the Web . . . . .	88
	Calculator Forms . . . . .	90
	Search Forms . . . . .	93
	Long Forms . . . . .	96
<b>Chapter 3</b>	<b>Tables . . . . .</b>	<b>99</b>
	Basic Table . . . . .	101
	Headerless Table . . . . .	102
	Fixed Column . . . . .	104
	Overview plus Data . . . . .	105
	Grouped Rows . . . . .	107
	Table with Visual Indicators . . . . .	108
	Editable Table . . . . .	111
<b>Chapter 4</b>	<b>Search, Sort, and Filter . . . . .</b>	<b>115</b>
	Search Patterns . . . . .	116
	Implicit Search . . . . .	116
	Explicit Search . . . . .	121
	Search with Auto-Complete . . . . .	126
	Dynamic Search . . . . .	128
	Scoped Search . . . . .	130
	Saved, Recent, and Popular Search . . . . .	131
	Search Form . . . . .	133
	Search Results/View Results . . . . .	136
	Sort Patterns . . . . .	139
	Onscreen Sort . . . . .	141
	Sort Overlay . . . . .	143
	Sort Form . . . . .	145
	Filter Patterns . . . . .	149
	Onscreen Filter . . . . .	149
	Filter Overlay . . . . .	152

	Filter Form . . . . .	154
	Filter Drawer . . . . .	156
	Gesture-Based Filtering . . . . .	158
<b>Chapter 5</b>	<b>Tools . . . . .</b>	<b>161</b>
	Toolbar . . . . .	163
	iOS . . . . .	163
	Android. . . . .	165
	Windows Phone. . . . .	166
	OS-Neutral Pattern: Contextual Toolbar . . . . .	168
	Toolbox. . . . .	168
	Call to Action Button. . . . .	172
	Inline Actions . . . . .	177
	Multi-State Button . . . . .	180
	Contextual Tools . . . . .	181
	Bulk Actions . . . . .	185
	Lock Screen Controls . . . . .	188
<b>Chapter 6</b>	<b>Charts . . . . .</b>	<b>191</b>
	Chart with Filters . . . . .	196
	Interactive Timeline. . . . .	200
	Data Point Details. . . . .	202
	Drill Down . . . . .	208
	Overview plus Data . . . . .	208
	Interactive Preview. . . . .	211
	Dashboard. . . . .	213
	Zoom. . . . .	216
	Sparklines. . . . .	218
	Integrated Legend. . . . .	220
	Thresholds . . . . .	220
	Pivot Table . . . . .	222
	Pulling It All Together . . . . .	223

<b>Chapter 7</b>	<b>Tutorials and Invitations</b> .....	<b>227</b>
	Tutorial Rules .....	228
	Rule #1: Use Less Text .....	230
	Ness Compared to Foodspotting .....	231
	Boomerang Compared to Mailbox .....	233
	DigiCal Compared to Fantastical .....	234
	Catch Compared to Clear .....	235
	SlideStory Compared to Vine .....	236
	Rule #2: No Frontloading .....	236
	Phoster Compared to Creative Studio .....	238
	Dooo Compared to Todoist .....	239
	Buy Me a Pie! Compared to OneNote .....	240
	Clipchat Compared to Kik .....	241
	Rule #3: Make It Rewarding .....	242
	NBC News Compared to Flipboard .....	242
	Noom Compared to DailyBurn Tracker .....	244
	Rule #4: Reinforce Learning Through Use .....	246
	Rule #5: Listen to Your Users .....	248
	Invitation Patterns .....	250
	Tips .....	250
	Persistent Invitations .....	252
	Discoverable Invitations .....	254
	Chapter Extra: Invitations—Rolling Out the Welcome Mat.....	255
	Iterating on the Welcome Experience .....	256
	Summary .....	263
<b>Chapter 8</b>	<b>Social Patterns</b> .....	<b>265</b>
	Social Registration .....	265
	MapMyFitness Compared to We Heart It .....	266
	Connecting .....	267
	Following .....	270
	Profiles .....	272

	Groups .....	274
	Gamification .....	278
<b>Chapter 9</b>	<b>Feedback and Affordance .....</b>	<b>283</b>
	Feedback Patterns.....	283
	Error Messages .....	284
	Confirmation .....	285
	System Status.....	293
	Affordance .....	298
	Tap .....	298
	Swipe/Flick .....	299
	Drag .....	304
<b>Chapter 10</b>	<b>Help.....</b>	<b>309</b>
	How-Tos.....	313
	User Guide/Help System .....	315
	FAQs .....	319
	Feature Tours .....	320
	Tutorials.....	323
	Contextual Help.....	326
	Capture Feedback.....	328
<b>Chapter 11</b>	<b>Anti-Patterns.....</b>	<b>331</b>
	Novel Notions .....	332
	Needless Complexity.....	338
	Metaphor Mismatch .....	342
	Control Mismatch .....	342
	Icon Mismatch.....	345
	Gesture Mismatch.....	345
	Mental Model Mismatch.....	347
	Idiot Boxes .....	348
	Chart Junk .....	351
	Oceans of Buttons.....	355
	Square Peg, Round Hole.....	359

Chapter Extra: Let Them Pee—Avoiding the Sign-Up/Sign-In Mobile Anti-Pattern.....	361
<b><i>Appendix A: Additional Resources</i></b> .....	<b>365</b>
<b><i>Index</i></b> .....	<b>367</b>

## [ *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.

## O'Reilly Safari

Safari (formerly Safari Books Online) is a membership-based training and reference platform for enterprise, government, educators, and individuals.

Members have access to thousands of books, training videos, Learning Paths, interactive tutorials, and curated playlists from over 250 publishers, including O'Reilly Media, Harvard Business Review, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Adobe, 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, and Course Technology, among others.

For more information, please visit <http://oreilly.com/safari..>

## 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](mailto: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.

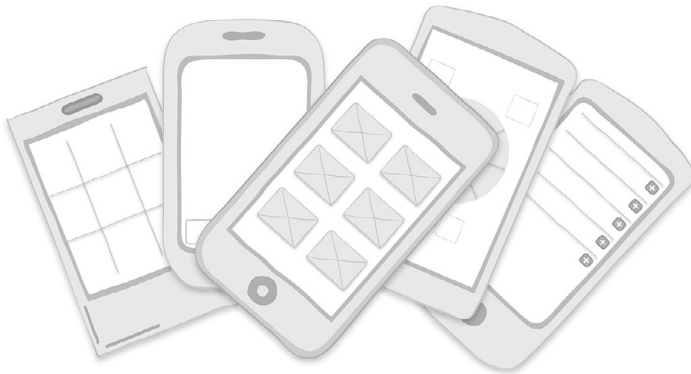
This is my third book for O'Reilly Media, and it has been a pleasure to work with Mary Treseler and her team again.

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)!



[ 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.

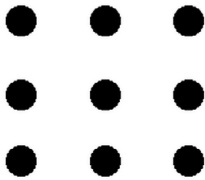
## Primary Navigation Patterns, Persistent

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.

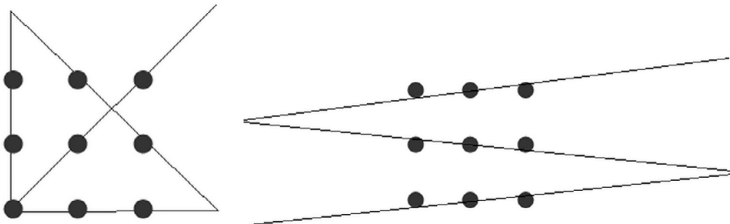
Transient navigation, however, must be explicitly revealed with a tap or gesture. These patterns arise from the constraints of smartphone screen sizes, which have pushed mobile designers to think “outside of the box,” literally.

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

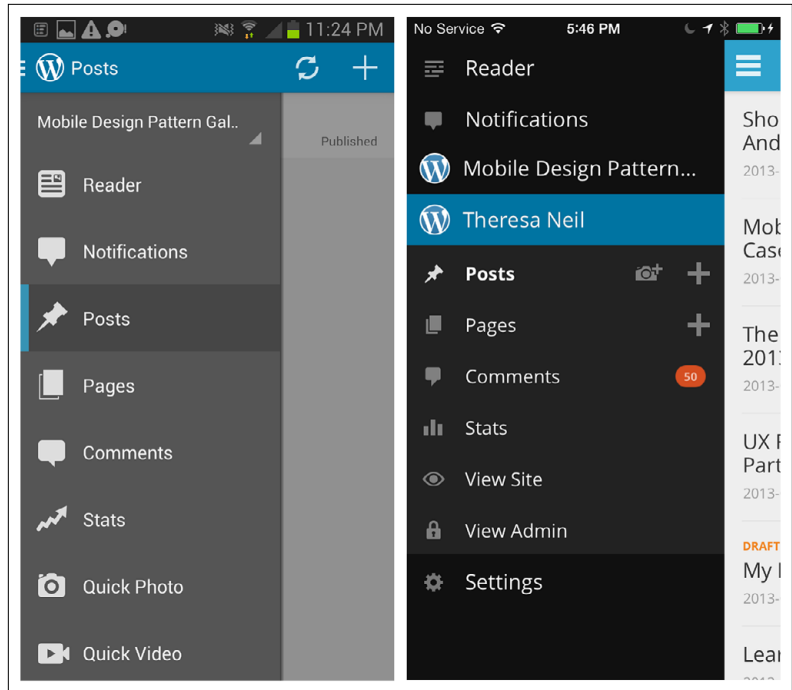
How’d you do? You probably figured out that the only way to solve this puzzle is to break free of the artificial boundaries. In the mobile world, it’s called thinking “off-canvas.”



**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.

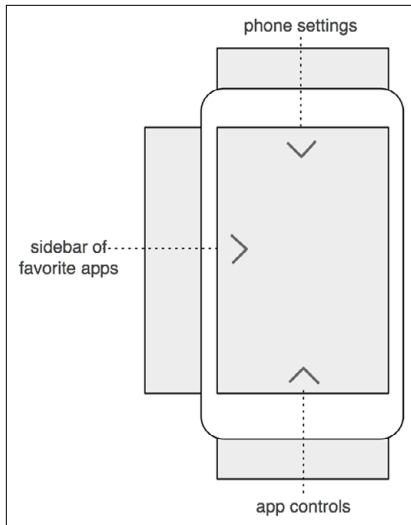
**FIGURE 1-3.**  
WordPress for Android  
and iOS: Side Drawer  
represents “off-canvas”  
thinking



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





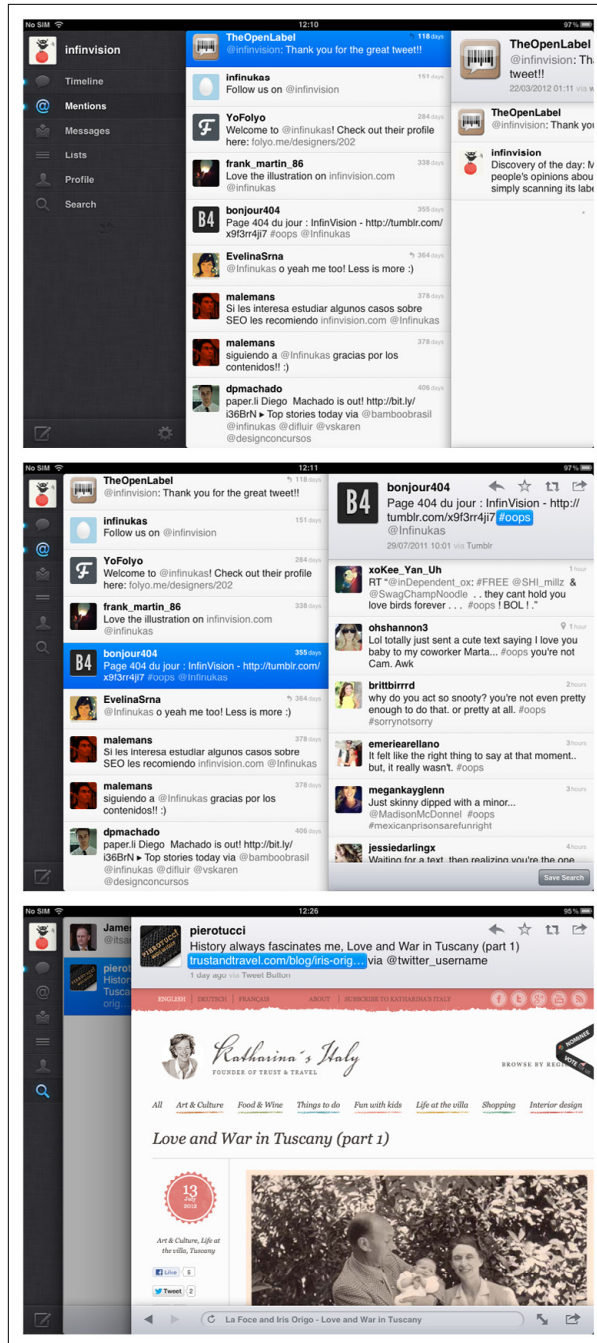
**FIGURE 1-5.**

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.

**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?

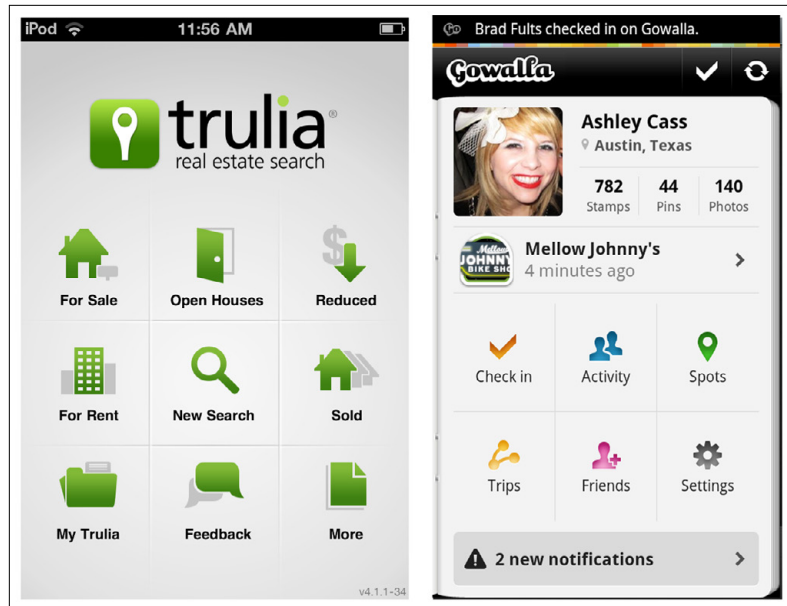
If you answered “yes” to one or more of these questions, it’s probably best to stick with persistent navigation. Now let’s take a look at those patterns.

### 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

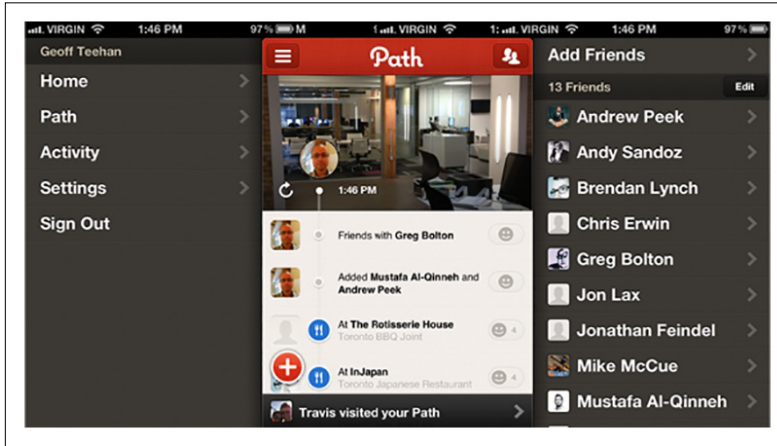


**FIGURE 1-8.**  
Facebook for iOS and  
LinkedIn for Android:  
Springboard designs  
from 2011



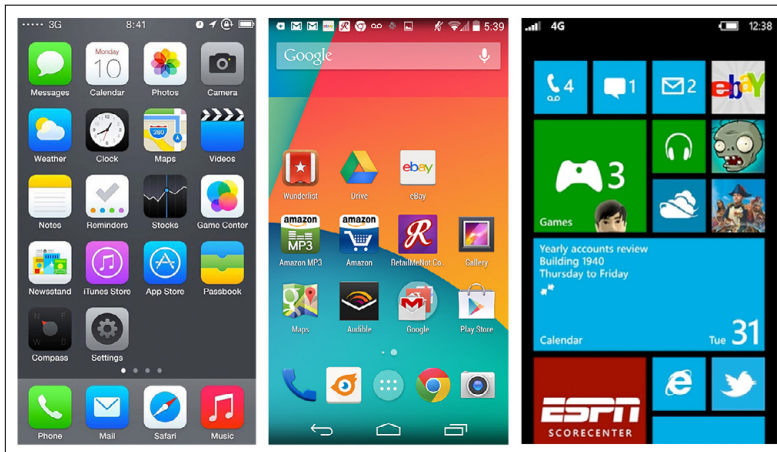
The main drawback of the Springboard pattern is that it flattens all options to the same level of importance. Enter the Side Drawer pattern, first designed by Aza Raskin for Firefox Mobile (<http://www.azarask.in/blog/post/firefox-mobile-concept-video/>), and adopted by

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.**

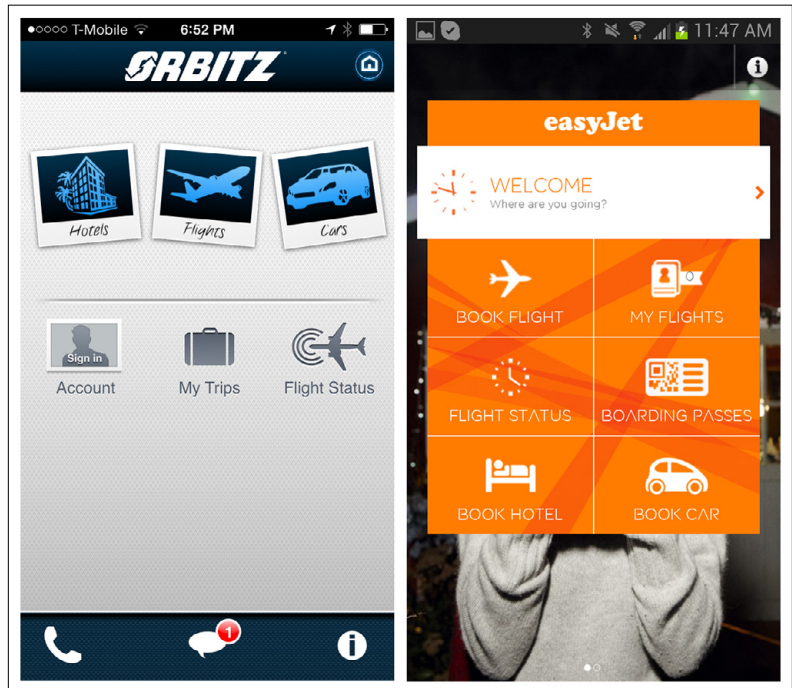
LearnVest for iOS, BBC Radio for Windows Phone, and Vimeo for Android: traditional Springboard alive and well within apps



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

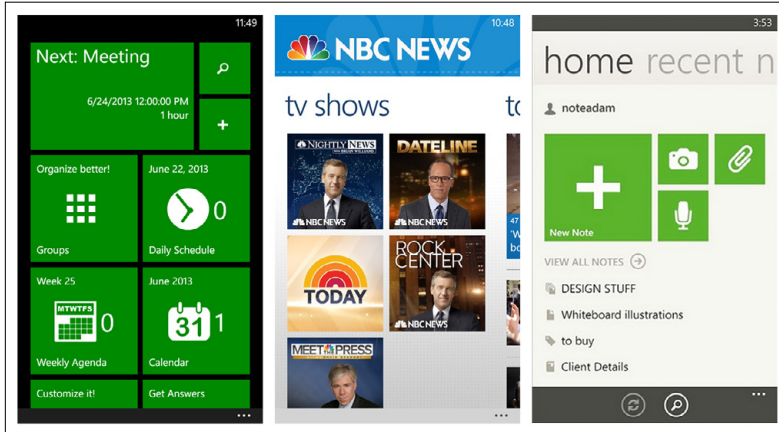
**FIGURE 1-12.**

Orbitz for iOS and EasyJet for Android: graphic treatment and layout imply a hierarchy



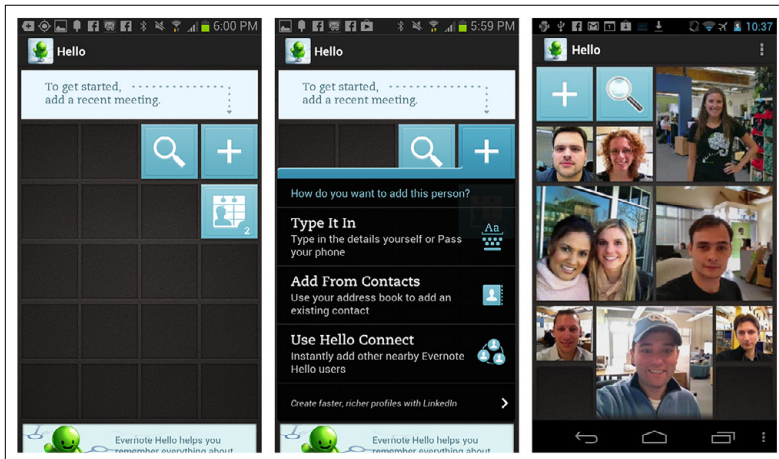
Windows Phone has pushed the Springboard pattern the farthest with *tiles*. Tiles can be live or static, and come in three different sizes. Live tiles convey dynamic information like number of calls missed, details of your next appointment, or the avatar of your last caller. Read the Windows Design Guide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

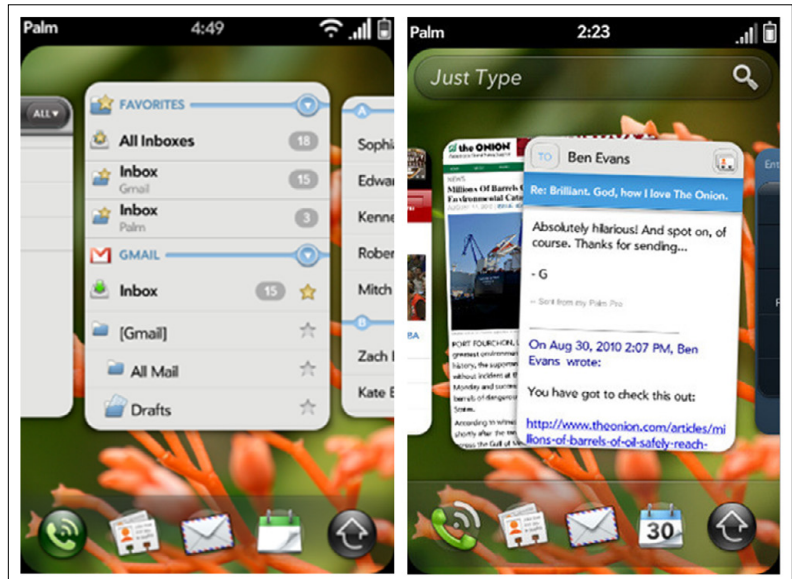


**FIGURE 1-14.** Evernote Hello for Android: tile-inspired customizable Springboard

## Cards

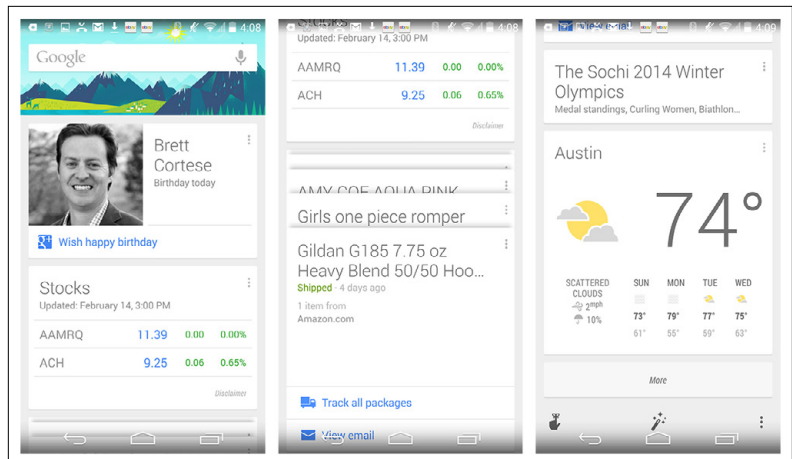
Cards may seem familiar to those of us who had a Palm in 2010–2011. Card navigation is based on a card deck metaphor, including common card deck manipulations such as stacking, shuffling, discarding, and flipping.

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

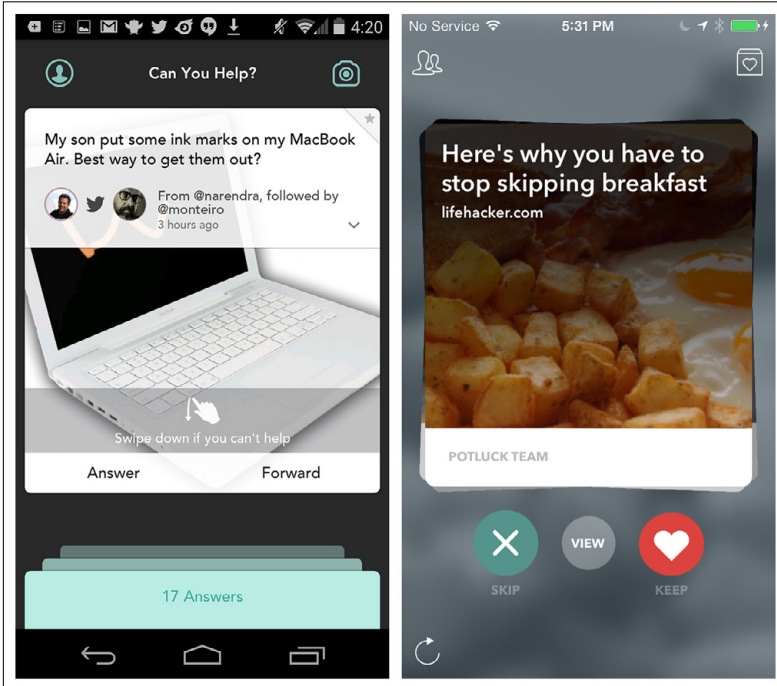


This pattern has become popular again with the release of Google Now, which stacks information-rich cards vertically to display a long list of launch points into the app, or quick actions in context.

**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](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

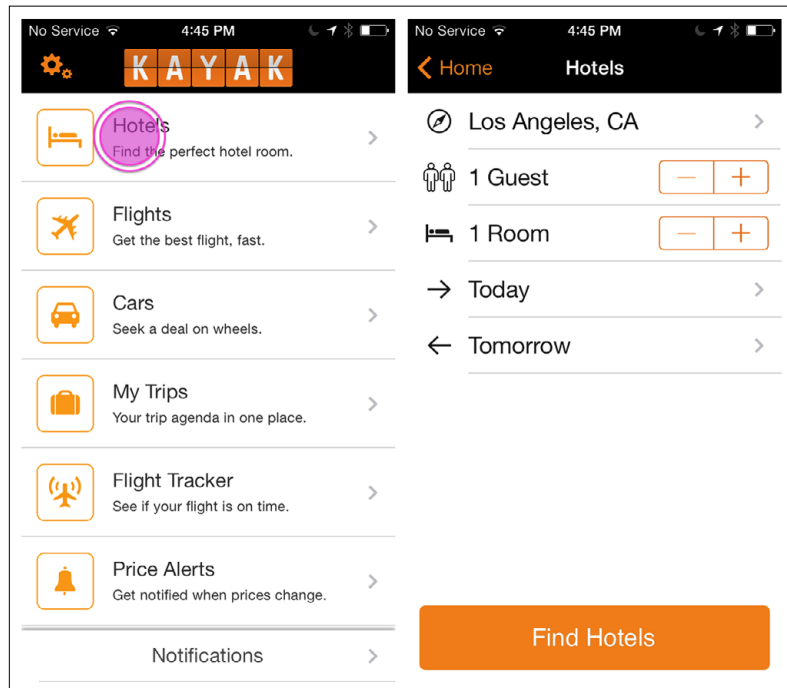
The List Menu pattern is similar to the Springboard in that each list item is a launch point into the application, and switching modules requires navigating back to the list. Apple (<http://bit.ly/1dZDU8>) calls this *hierarchal navigation*:

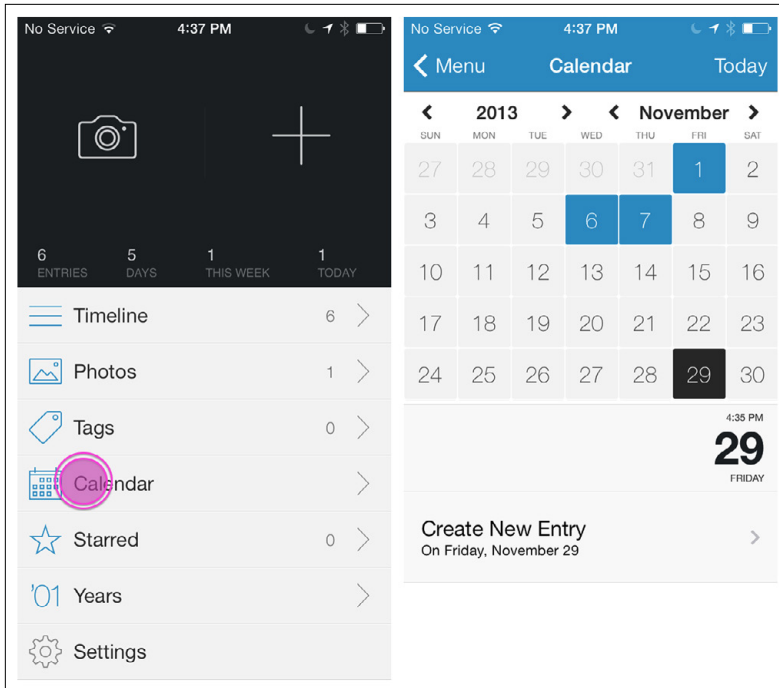
In a hierarchical app, users navigate by making one choice per screen until they reach their destination. To navigate to another destination, users must retrace some of their steps—or start over from the beginning—and make different choices. Settings and Mail are good examples of apps that use a hierarchical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

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

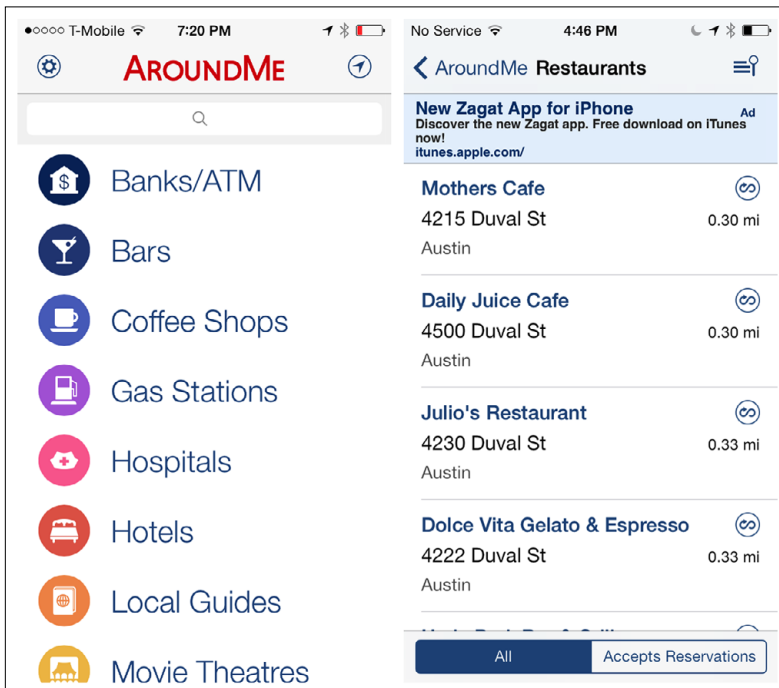
**FIGURE 1-18.**

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



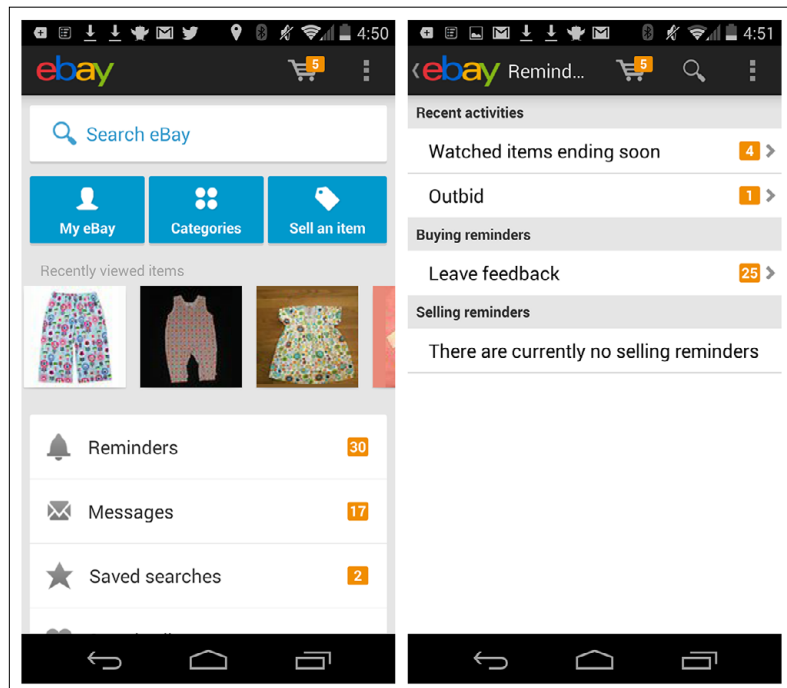
**FIGURE 1-20.** AroundMe for iOS: “Home” or “Menu” might’ve been a better choice for the Back button label

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 top-most 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 ]

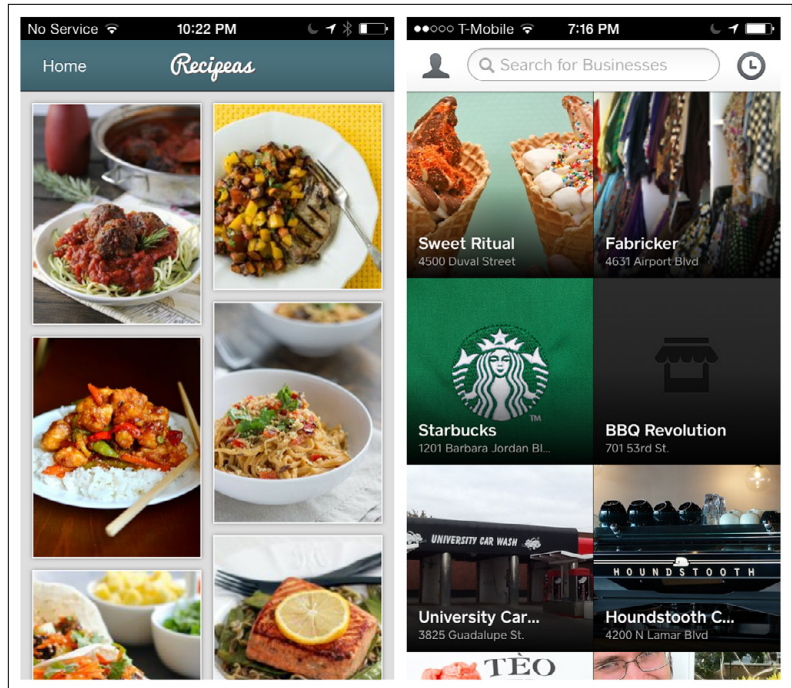
Use a Dashboard when it makes sense to use key metrics or data as launch points into the app. But don't overload the Dashboard; conduct research to determine which key metrics or data to include.

## 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.

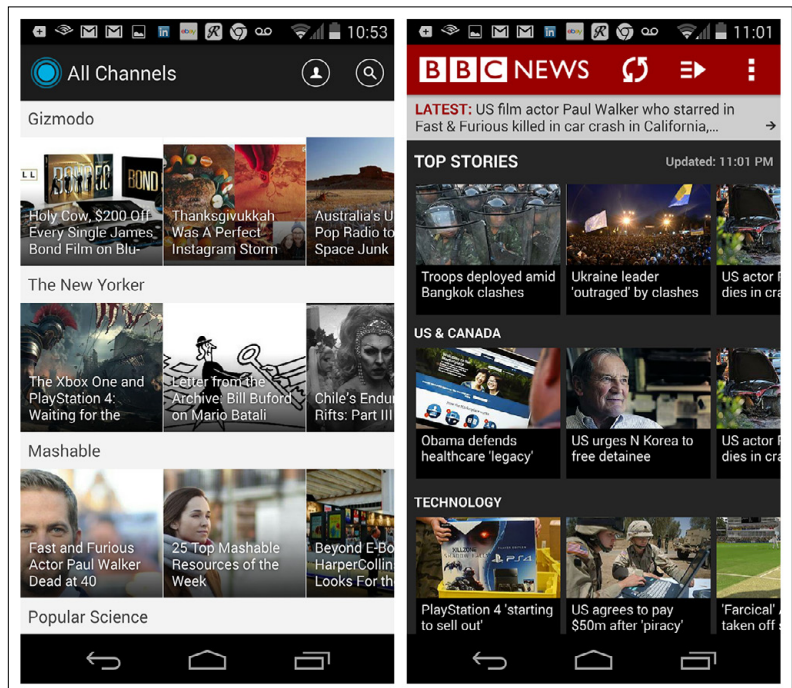
**FIGURE 1-23.**

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.

## 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

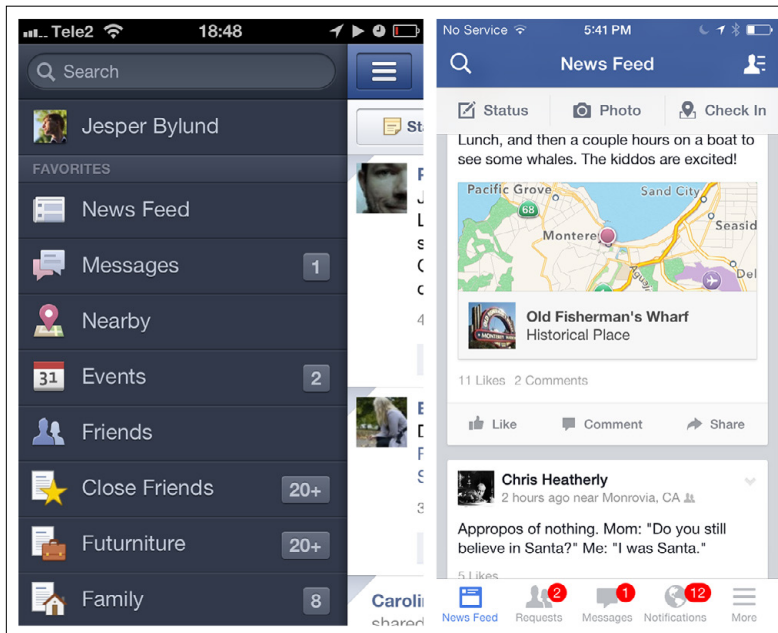
Since the first release of iOS, Apple has recommended (<http://bit.ly/1dZF9Vt>) the Tab Bar for navigating *flat apps*:

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/1dZFIUF>.

## [ NOTE ]

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



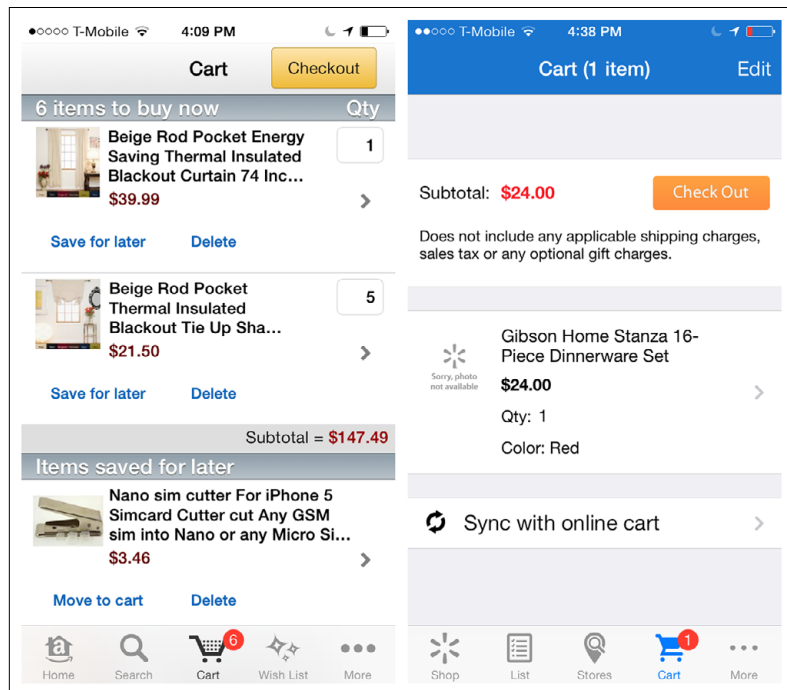
**FIGURE 1-25.**

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

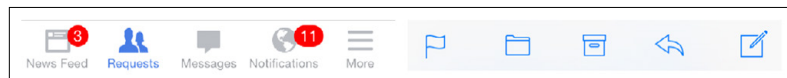
The iOS Tab Bar is restricted to five menu items. If the application has more than five primary categories, a More option can be provided as the fifth tab on the right.

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

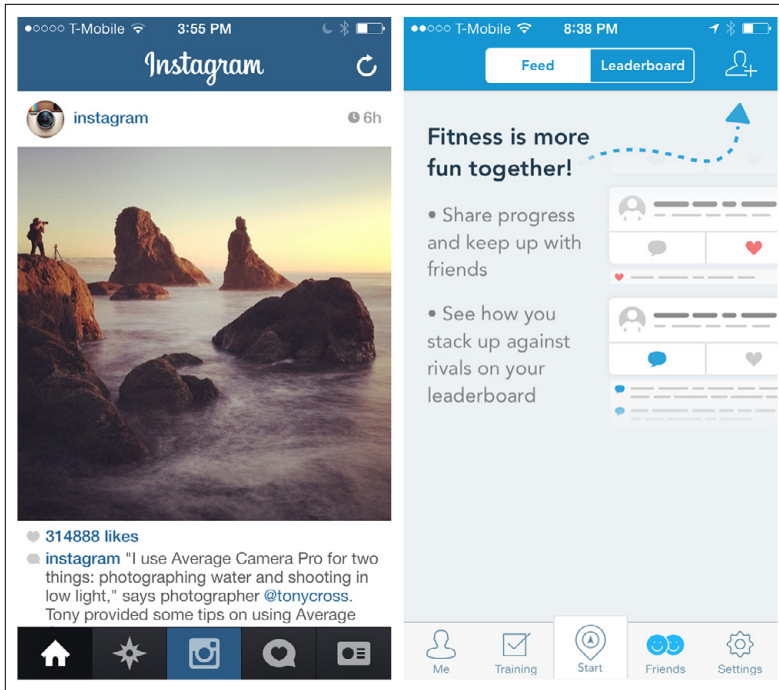


**FIGURE 1-27.**  
Tab Bar (left) has menu  
items; Toolbar (right)  
has tools for actions



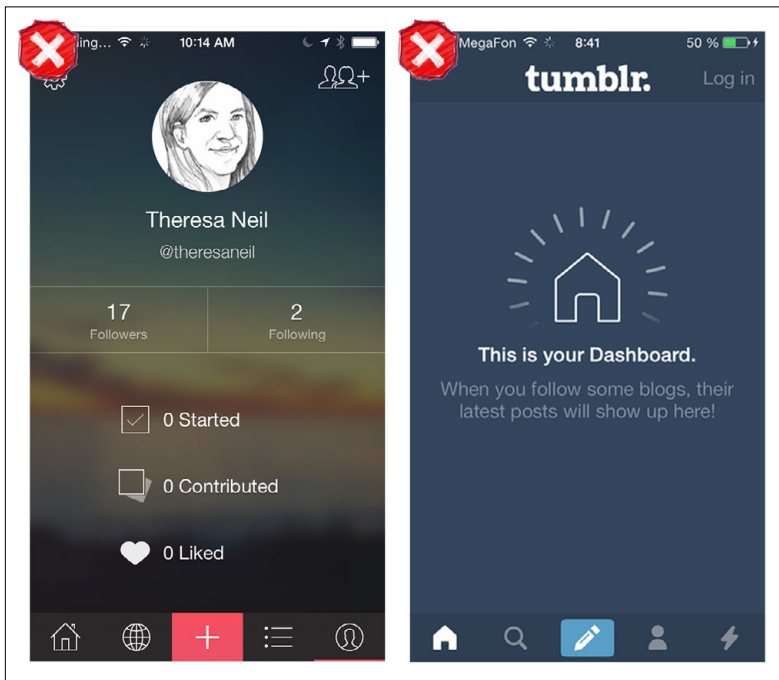
Some applications, like Instagram and RunKeeper, rely so heavily on the user taking a single action (like taking a picture or starting a run) that they place *calls to action* (a single action more prominent than the rest) in their Tab Bars.

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:

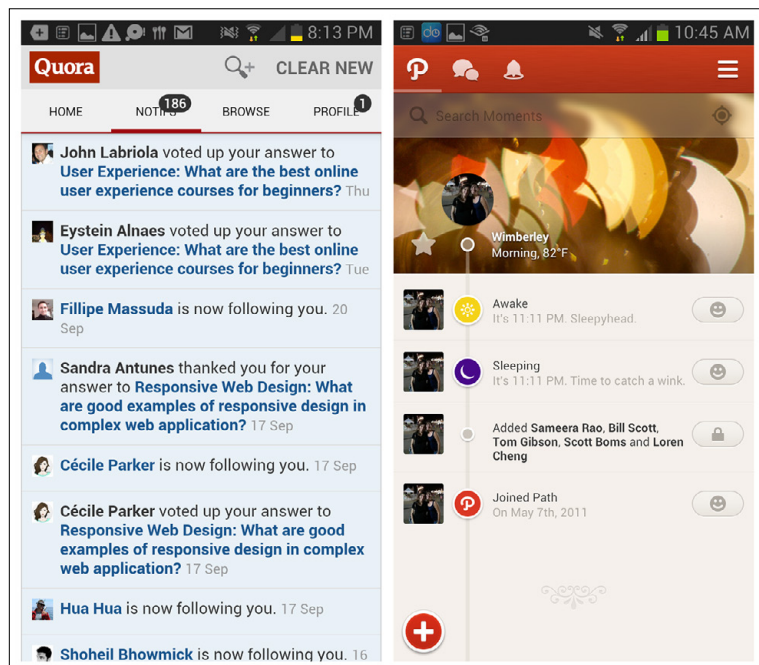
- You expect your app's users to switch views frequently.
- You have a limited number of up to three top-level views.
- You want the user to be highly aware of the alternate views.

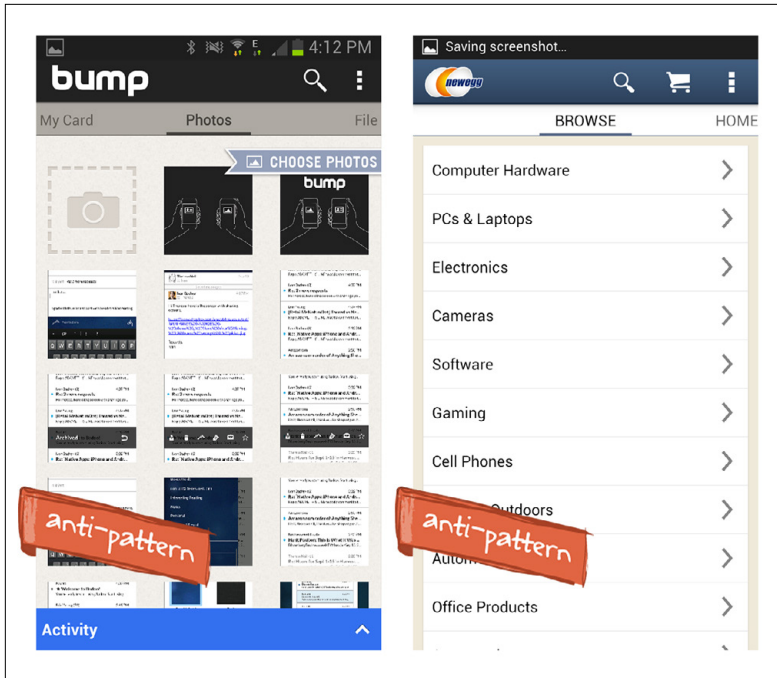
Path makes Fixed Tabs work by using icon-based tabs, while Quora pushes the limit, squeezing in four text-based tabs. More often than not, designers incorrectly use the Scrolling Tab control for primary navigation when they should be using a Spinner or Navigation Drawer instead.

**FIGURE 1-30.**

Quora for Android pushes the Fixed Tabs limit by squeezing in four items; Path for Android uses icons for

Fixed Tabs



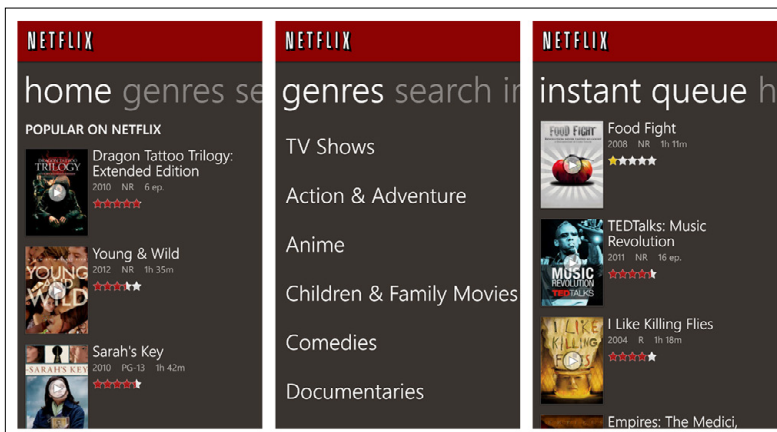


**FIGURE 1-31.** Bump and Newegg for Android: incorrect use of Scrolling Tabs for primary navigation

## 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