

User Experience for Mobile Applications and Websites

Design Guidelines for Improving the Usability of Mobile Sites and Apps

By Raluca Budiu and Jakob Nielsen

3rd edition



WWW.NNGROUP.COM

48105 WARM SPRINGS BLVD., FREMONT CA 94539-7498 USA

Copyright © Nielsen Norman Group, All Rights Reserved. To buy a copy, download from:
<http://www.nngroup.com/reports/mobile-website-and-application-usability>

Copyright Notice

Please do not post this document to the internet or to publicly available file-sharing services.

This report required hundreds of hours of planning, recruiting, testing, analyzing, writing and production. We sell these reports to fund independent, unbiased usability research; we do not have investors, government funding or research grants that pay for this work.

We kindly request that you not post this document to the Internet or to publicly available file-sharing services. Even when people post documents with a private URL to share only with a few colleagues or clients, search engines often index the copy anyway. Indexing means that thousands of people will find the secret copy through searches.

If someone gave you a copy of this report, you can easily remedy the situation by going to <http://www.nngroup.com/reports> and paying for a license.

We charge lower fees for our research reports than most other analyst firms do, because we want usability research to be widely available and used.

Thank you!



Report Authors: Raluca Budiu and Jakob Nielsen

Contents

Executive Summary	6
User Research	7
Basic Characteristics of the Mobile User Experience.....	7
Mobile Usability Is Harder to Achieve.....	9
Terminology	11
What's New in the Third Edition	12
Research Overview.....	13
Mobile Activities	15
Mobile Information Needs.....	16
Mobile Activities.....	18
What Tasks Are Important on Mobile	19
Mobile Limitations and Strengths	21
Small Screen	21
Portable Means Interruptible	22
Single Window	22
Touchscreen.....	23
Variable Connectivity	23
GPS, Camera, Accelerometer, Voice, and Other Phone Features.....	24
Mobile Website or Mobile App?	25
When Should You Build an App?.....	27
Mobile Websites	28
Mobile Apps.....	36
Mobile Usability Guidelines.....	39
Making Your Mobile-Dedicated Website Findable	39

Making Your App Findable	49
Mobile-App Functionality.....	57
Logo and Company Name	65
Touch Targets	69
Icons.....	90
Typing and Input Fields.....	94
Spinners, Sliders, and Other Selectors	121
Menus: Drop-downs, Accordions, Overlays, Sliding Menus, Pie Menus	127
Carousels	156
Buttons and Button Placement	166
Errors and Alerts	173
Notifications.....	188
Progress Indicators.....	193
Orientation.....	197
Gestures.....	202
Forms.....	223
Navigation.....	278
Search, Filter, Sort	326
Lists and Scrolling.....	381
Content	389
Location Information	443
Instructions and Help.....	457
Initial Experience	469
Methodology	475
Diary Studies.....	475
Usability Testing	477

Design Reviews	485
List of Guidelines	487
Acknowledgments	505
About the Authors	506

Executive Summary

This report presents user-experience guidelines for designing for mobile devices, as well as the findings from our user research with these devices. In contrast to much other writing on mobile design, these guidelines are not based on our personal opinions or preferences. Very few companies target an audience of Ph.Ds. who immigrated from Europe but now live in Silicon Valley and have decades of experience in human-computer interaction. So it's irrelevant what we like. That's why we base the recommendations on direct observation of real user behavior, as usability-test participants interact with a wide variety of user-interface designs. We can see what works and what doesn't work, and what this much broader audience likes and dislikes. Such behavioral data is the only sound basis for UX advice, so that's what this report is built on.

The next several hundred pages focus on mobile devices, for which we now have a fairly narrow definition that targets the currently popular models of smartphones with a touchscreen that's about 3.5–6.5 inches large. This includes iPhone, Android, Windows Phone, and other similar phones. A few of the guidelines are specific to an individual platform, but, since user experience is a matter of *the user's* experience, it follows that most of the UX guidelines are not determined by the exact model of technology.

Touchscreens that are 7 inches or bigger are covered in our separate *Tablet User Experience*¹ report and not here. We also don't cover smaller devices, such as smartwatches, feature phones, or the first generations of smartphones that allocated about half of their surface area to a small screen and the other to a small keyboard. Feature phones and these early smartphones were covered in previous editions of this report and were included in our early research studies, but we know from teaching our *Mobile UX* training courses² that almost our entire audience has been targeting solely full-screen touch devices in recent years. We have targeted this new report edition and our recent research rounds accordingly.

This is the third edition of our report. The following table shows the count of design guidelines in each of the three editions:

Report Edition	1 st	2 nd	3 rd
Number of UX Guidelines	85	237	335

It's clear that as time goes by, we have more and more guidelines to follow in mobile design. This is partly because we conduct new user research for each new report edition. These new studies can't help but uncover new insights, and yet the findings from past studies often remain valid (or, at most, in need of minor modification in tune with recent developments). There's also an increasing body academic research for us to draw on.

More research has definitely led to more insight, but that's not the only reason for the increasing number of design guidelines. Over the years, mobile devices are being used for more and more tasks by an ever broader segment of the population. We have more websites, more mobile apps, and more interaction styles to consider, as inventive designers dream up new ideas that sometimes work and sometimes cause

¹ Please see <http://www.nngroup.com/reports/tablets>

² Please see <http://www.nngroup.com/courses/topic/mobile-and-tablet-design>

new usability problems. Furthermore, users are becoming more demanding of the mobile user experience. The original iPhone was released in June 2007 as the first device falling within the range we target for this report (3.5 inch touchscreen). When we conducted the research for the first edition of this report two years later, most people still had primitive mobile devices with horrible user experience. At the time, anything that was half-way decent was a pleasant experience compared with the misery caused by most mobile devices and most mobile sites. Users couldn't really be that demanding back then. Today they most certainly can demand a good mobile user experience — or they'll go elsewhere.

USER RESEARCH

The majority of the research presented in this report was done as one-on-one usability testing. We recruited a broad range of test participants and had them use their own phones during their test sessions. Where applicable, we asked participants to show us the apps that they had installed on their phones, and then we gave them tasks to complete using either mobile apps or the web.

We carried out 15 separate usability-testing studies over seven years. The majority of the studies were done in the US; we also ran testing sessions in the UK, Hong Kong, Australia, Netherlands, and Romania.

Sessions lasted between 60 and 90 minutes. Overall, we had 151 participants, out of which 124 were from the US. We studied 109 touch phones (57 iPhones, 36 Android phones, 9 Windows phones, and 7 other), 28 nontouch smartphones, and 14 feature phones.³

To understand what activities are normally done on mobile devices, we carried out two diary studies: one international and one in the US. The international diary study involved people from six different countries in Europe, Asia, Australia, and America, who owned different types of phones. For the second diary study, we focused on iPhone owners in the United States. Participants logged all the activities that they carried out on their mobile devices for approximately one week. The diary gave us information about the range of mobile activities done by advanced users and informed our choice of tasks for the traditional usability testing.

BASIC CHARACTERISTICS OF THE MOBILE USER EXPERIENCE

Mobile phones come with strengths, but also limitations. These strengths and limitations are the underlying reasons for the mobile user experience guidelines presented in this report.

Small Screen

In spite of the modern trend towards larger-screen phones, what makes mobile phones so convenient and portable is their small size. Compared with desktop and even laptop screens, phone screens accommodate a lot less content. As a result, screen size is a serious limitation for mobile devices. The content displayed above the fold on a 30 inch monitor requires 5 screenfuls on a small 4-inch screen.

³ Feature phones were only included in our first research rounds and the findings were included in the first edition of this report. Nontouch phones were included in some of the early research rounds, and findings were included in the first and second editions of this report. The current edition of the report only covers findings from touchscreen users.

Whenever you include a new design element or a new piece of content on the mobile screen, something else gets pushed out (or below the fold). Think hard of the opportunity cost of each new element: what does it mean for the users if you leave out element B in order to include element A? Is element A more important than element B? Content and feature prioritization are key. Although we provide general guidelines in this report, your answer likely depends on the kinds of users and tasks that you have.

Portable = Interruptible

Mobile phones are portable: most fit easily in a pocket or purse and they tend to follow us everywhere. Because we use phones in a variety of contexts and situations, we are more likely to be interrupted when using such devices: an external event in the outside environment may demand our attention and require us to stop whatever we were doing on the small screen. As a result, attention on mobile is often fragmented and sessions on mobile devices are short. In fact, the average mobile session duration is 72 seconds. In comparison, our own studies show that on desktop, the average session is 150 seconds: more than twice as large.

Short mobile sessions mean that we must design for interruptions: save state for users and allow users to save state.

But designing for interruptions doesn't only mean saving state. It also means prioritizing the essential and simplifying tasks and interactions. Because attention is fragmented, strive to show users what they need as soon as possible. Flooding them with details and asking them to parse walls of text for relevant facts is not interruption friendly. The gist should always come before the minutiae. A simple task is easier to finish quickly. It's also easier to resume than one with many steps and alternatives.

Single Window

Although some phone manufacturers are trying to accommodate multiple windows on the screen at the same time, the limited size of the mobile screen makes that goal quite unpractical, even with today's larger-screen phones. The vast majority of users only see a single window (and thus a single application or website) at a time; they cannot split the screen (as on the desktop) and work with two different apps simultaneously.

The single-window constraint means that design should be self-sufficient: Any mobile tasks should be easy to complete in a single app or on a single website. Users should not have to leave an app (or website) to find information that the app requires, but that it doesn't provide. Remember that pen and paper, even if available, are often unusable on the go. If users must move information from one app to another, it's likely that they will need to copy-and-paste it (or worse, rely on their memory and increase their cognitive load); the interaction will become more complex and error prone. Apps and websites should be self-sufficient and should not necessitate any external props, be they physical or virtual.

Touchscreen

Touchscreens come with many blessings and sins. Gestures represent a hidden, alternate user interface (UI), that, when built with the right affordances, can make the interaction fluid and efficient and can save screen real estate. But they also suffer from low memorability and discoverability. On the other hand, it's hard to type

proficiently on a tiny virtual keyboard and it's easy to accidentally touch the wrong target.

Typing is simply painful on mobile.

Another difference between touch and mouse input is that target sizes must be considerably larger for touch than for mouse to avoid errors and facilitate speedy use. So not only is the screen *smaller* than a laptop or desktop screen, but design elements need to be *larger* than those on a regular monitor.

Variable Connectivity

Even in the era of fast cellular networks and ubiquitous Wi-Fi, coverage is not universal or equally good. Phone users frequently complain about connectivity problems. Every new page load translates into a significant waiting time when the network does not cooperate.

If you want users to finish their tasks on your mobile site or in your app, mind the waiting time. Design pages that are light, yet contain as much information as possible, to avoid many back-and-forth trips from client to server. Minimize the number of steps and, ultimately, the number of page loads.

MOBILE USABILITY IS HARDER TO ACHIEVE

As noted above, users' expectations for the mobile user experience have definitely increased, as people use their mobile devices for more tasks and as they have more competing services to turn to when they get disappointed.

But the limitations inherent in small devices make it harder for designers to deliver on this demand. Many of the usability guidelines for mobile are similar to guidelines for desktop user interfaces, but because of the platform limitations, the mobile guideline is stricter and allows less scope for nonconforming designs before they become too unpleasant to use.

As an example, it's always been a guideline when designing web forms to limit the number of optional (i.e., non-essential) form fields. Users don't like completing long forms, whatever the platform. Similarly, since our very first e-commerce usability studies⁴ in 2000 we have had a guideline to offer guest checkout and not require customers to register to be allowed to buy.

We present similar guidelines in this report, but with two substantial differences:

- What's a slightly annoying, slightly long form for a desktop user would be an overwhelmingly long form for a mobile user. And a superfluous question that may slow down a desktop user by a few seconds of extra typing will often be the straw to break a mobile camel's back (i.e., to cause a mobile user to abandon the site).
- Whereas a desktop user might prefer guest checkout, a mobile user would often be completely unable to log in as a registered user because he or she would have forgotten any existing login credentials or find it too convoluted to create new ones. Thus, an e-commerce site without guest checkout would

⁴ For the current (much expanded) version of the e-commerce user experience guidelines, please see <http://www.nngroup.com/reports/ecommerce-user-experience/>

lose a few percent of sales to desktop users but a huge percent of sales to mobile users.

Other times, we have different guidelines for desktop and mobile design. For example, breadcrumbs are recommended to assist navigation when using a large screen. However, desktop sites should never rely on breadcrumbs as their only navigation aid, and we recommend showing a fair amount of global navigating options on all web pages that are seen on a desktop-sized monitor.

In mobile designs, these design guidelines reverse: most sites should not have breadcrumbs, because they would consume too large a percentage of a small screen. And even though global navigation remains useful, mobile sites should often make the trade-off to hide the global navigation menu under a small icon, so that users only see it when they request that it be shown.

The truly fundamental human-factors principles certainly remain the same for desktop and mobile user interfaces, since people don't change. It's always good to help users understand their navigational location and their navigation choices. And it's also always good to show users the content they're interested in. However, you can't show everything on every screen. (And if you could, you would violate the principle of avoiding information overload.) Thus, for different screen sizes, our trade-offs will differ, and we should prioritize the different design elements differently.

Terminology

This report is concerned with touchscreen **smartphones**. Unless otherwise noted, the phrases **mobile device** and **mobile phone** refer to mobile smartphones that have a touchscreen.

What's New in the Third Edition

This third edition of the mobile report is a completely revised and reorganized version of our second edition that reflects the changes in the mobile landscape. Some of the topics discussed in the second edition have become less relevant, as companies are already convinced of the importance of having a mobile presence. And since most mobile phone nowadays are touchscreen based, we no longer distinguish between guidelines that apply to all phones and guidelines for touchscreen phones; instead we simply assume that our readers will design for a touchscreen smartphone. Moreover, to reflect the needs of our audience, we completely eliminated the section on feature phones.

The additions in this report reflect many of the new mobile topics and patterns that emerged or were revived in the recent years since the second edition. They include, but are not limited to responsive design and how it relates to mobile design in general, findability of apps, flat design, accordions and overlays, hamburger menus, gestures, pagination and infinite lists, flattening navigation hierarchies for mobile, secondary navigation and submenus, in-page table of contents and page mini-IAs, instructional overlays.

All of the guidelines in this report apply to the larger-screen phones commonly known as **phablets**. And many of them also apply to touch-based tablets. However, there are two big differences between tablets and mobile phones: (1) the tablet screen is bigger; (2) tablets are used in different ways than mobile phones. (Whereas many of the tasks carried out on mobile phones need to be quick, tablet users are not usually "in a rush", to quote one of our users.) Because of that, we do not address tablets in this report. Readers interested in that topic are referred to our separate report with tablet user experience research & guidelines.⁵

⁵ See <http://www.nngroup.com/reports/tablets/>

Research Overview

The main purpose of our research was to understand how people access the web on their mobile phones and what challenges they face when they use websites and applications on their phones. The result of this research effort is a set of design guidelines for websites and applications that are usable on mobile devices. We based these guidelines on methodical observations, interviews, user diaries, as well as expert reviews.

In this section we present a brief overview of our research project. For details about the methodology, please refer to the Methodology section in this report, starting on page 475. The research project encompassed three different methods:

- diary studies,
- usability testing,
- expert reviews.

Diary studies. To understand what activities are normally done on mobile phones, we carried out two diary studies: one international and one in the US. The international diary study involved people from six different countries in Europe, Asia, Australia, and America, who owned different types of phones (feature phones and smartphones, including touchscreen phones). For the second diary study, we focused on iPhone owners in the US. Participants used their mobile phones at least several times a week. They logged all the activities that they carried out on their mobile devices for approximately one week. The diary gave us information about the range of mobile activities done by advanced users and informed our traditional usability testing.

Usability testing. We carried out 15 separate usability-testing studies over seven years. The majority of the studies were done in the US; we also ran testing sessions in the UK, Hong Kong, Australia, Netherlands, and Romania. In all these studies, participants brought their own mobile phones into our lab. While older studies included all types of phones (ranging from touchscreen phones, nontouch smartphones, and feature phones), more recent studies focused on touch phones only, reflecting the current market-penetration figures. Moreover, the last of our studies exclusively involved phablets (touchscreen smartphones with screens larger than 5.3 inch). Where applicable, we asked participants to show us the apps that they had installed on their phones, and then we gave them tasks to complete using either mobile apps or the web.

The tasks could be grouped in two different categories:

- Specific tasks, in which users were asked to go to a given website or use a specific app to find specific information. These tasks specified the app name or the website URL. The URL could have been a mobile website (designed specifically for mobile devices — e.g., *"Use mobile.fandango.com to find out what movies are shown at your local movie theater"*) or just a domain name (e.g., *"Use fandango.com to find out what movies are shown at your local movie theater"*).
- Open-ended information-gathering tasks, in which users were free to use any website or application in order to find specific information (e.g., *"Find how many calories a slice of cheese pizza typically has"*).

Please refer to the *Methodology* section for examples of tasks.

We observed users as they worked and encouraged them to think aloud. As part of the sessions, we also interviewed participants about their common mobile practices and asked them to demonstrate some of their favorite sites and applications.

Sessions lasted between 60 and 90 minutes. Overall, we had 151 participants, out of which 124 were from the US. We studied 109 touch phones (57 iPhones, 36 Android phones, 9 Windows phones, and 7 other), 28 nontouch smartphones, and 14 feature phones.

While our very first studies looked at both full and mobile sites on mobile devices⁶, all other studies focused on how participants use mobile apps and mobile sites. Most of the time, when users were not able to find a mobile site on their own, we directed them to the mobile site.

Expert Reviews. The last source of information for our guidelines came from expert reviews. Over the course of three years we reviewed many apps and mobile websites on a variety of platforms. The list of these has become too long to include in this report, but the Methodology section contains a subset of sites and apps that were reviewed.

⁶ We included full sites in our initial studies for two reasons: (1) there were simply too few mobile sites when we first started our research and most users encountered full sites on their devices regularly; (2) we wanted to see how people react to full versus mobile sites and whether they can and do access mobile sites on their devices.

Mobile Activities

Mobile devices are small and portable; as a result, people have come to depend on them⁷ and carry them around at all times. While users use phones for a variety of tasks, these can be grouped into four big categories:

1. **Killing time.** In an attempt to make the time pass faster when they're waiting at the doctor's office or in a bus stop, or to seem busy while having lunch by themselves⁸, users browse around (using games, news, email, or social networking apps) and consume the content that is readily offered by apps and websites.
2. **Searching for specific information** is typically very focused: we don't see people planning their career, or researching hotels for their next vacation on the phone. The single window and small screen are strong limitations for any complex information-gathering tasks that require putting together multiple sources of information. When, in usability studies, we require users to perform such complicated tasks, the first thing many of them say is "I would never do this task on my phone." (Therefore, we don't include many complex tasks in our mobile studies.) Instead, when they are on mobile, users typically need to answer questions relevant to the here and now: where the closest gas station is, where to find a good place for lunch, what kind of reviews the product in front of them has, or even if they have enough money in their checking account to go to the store and make a purchase. Many of these questions have a strong local component: users are interested in answers around them. A Pew Internet survey shows that 49% of the cell-phone owners use their phone to get location-based information⁹.
3. **Communicating with others.** In surveys, the top most-frequently performed mobile activities are messaging, email, and social networking¹⁰.
4. **Transactions.** Although in the beginning we saw some reluctance to perform purchases and banking on mobile devices, these activities have now become more popular. According to Pew Internet¹¹, in 2011 18% of phone owners used their phone for banking activities, while in 2013 that number had already increased to 35%. A report by Nielsen (a company not associated

⁷ The cell phone is the technology that would be the second hardest to give up, after the Internet (Susannah Fox and Lee Rainie. "The Web at 25 in the U.S. Pew Internet," Pew Internet, February 2014.)

<http://www.pewinternet.org/2014/02/27/the-web-at-25-in-the-u-s/>

⁸ 13% of American cell phone owners pretended to be using their phones in order to avoid interaction with other people (Aaron Smith. "Americans and Their Cell Phones," Pew Internet, August 2011).

⁹ Maeve Duggan. "Cell Phone Activities 2013," Pew Internet, 2013.

http://www.pewinternet.org/files/old-media//Files/Reports/2013/PIP_Cell%20Phone%20Activities%20May%202013.pdf

¹⁰ See for instance the same Pew Internet survey cited above.

¹¹ Susannah Fox. "51% of Adults Bank Online," Pew Internet, 2013.

<http://www.pewinternet.org/2013/08/07/51-of-u-s-adults-bank-online/>

with Nielsen Norman Group) found that in the U.S. 48% of smartphone owners engage in mobile shopping and 38% do mobile banking¹².

MOBILE INFORMATION NEEDS

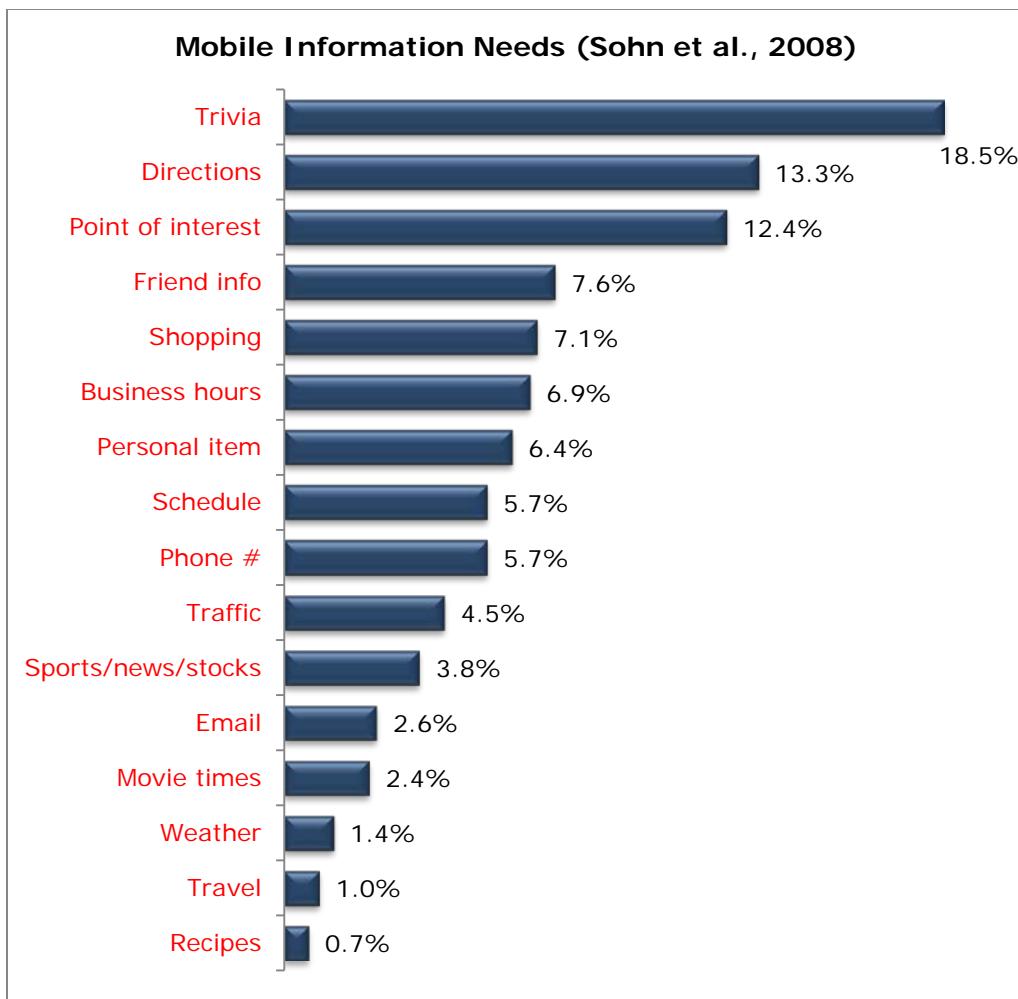
In the first edition of this report, we discussed a study published in CHI 2008¹³ that investigated what kinds of needs people have when they are away from home or work. In this study users added an entry to their diary each time they had a question (or an **information need**), whether they were able to answer that question using their mobile device or not. (Note that the majority of the users in that study did not have mobile Internet access.) By design, this study focused on the second type of activity that we normally see on mobile — searching for specific information — and ignored the kill-time aspect.

The figure below shows what kinds of questions people had when they were away from home or work. Even though the study may be considered “old” by now, these are fairly universal information needs that won’t change much over time. In fact, it can be considered an advantage of the 2008 study that it was based on identifying intrinsic information needs as opposed to measuring what people did with the devices they had at the time.

¹² Nielsen. “The Mobile Consumer: A Global Snapshot,” 2013.

<http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2013%20Reports/Mobile-Consumer-Report-2013.pdf>

¹³ Sohn, T., Li, K. A., Griswold, W. G., Hollan, J. D. “A diary study of mobile information needs,” *CHI ’08 Conference Proceedings*.



The most-frequent information needs were trivia: questions such as “Where did Bob Marley die and why?” Such questions occur often in conversations or are prompted by some contextual element. After trivia, the next most-frequent questions concern relatively simple facts such as directions, business hours, schedule, and contact information.

Sohn et al. also looked at how often the information needs were addressed. Their data showed that only 45% of the information needs were addressed at the time, by a variety of means that included the Internet (18%), but also making a phone call, printing beforehand the information that was needed, or going to a specific location. Therefore, in that study, most of users’ information needs remained unaddressed or were addressed using means other than the Internet.

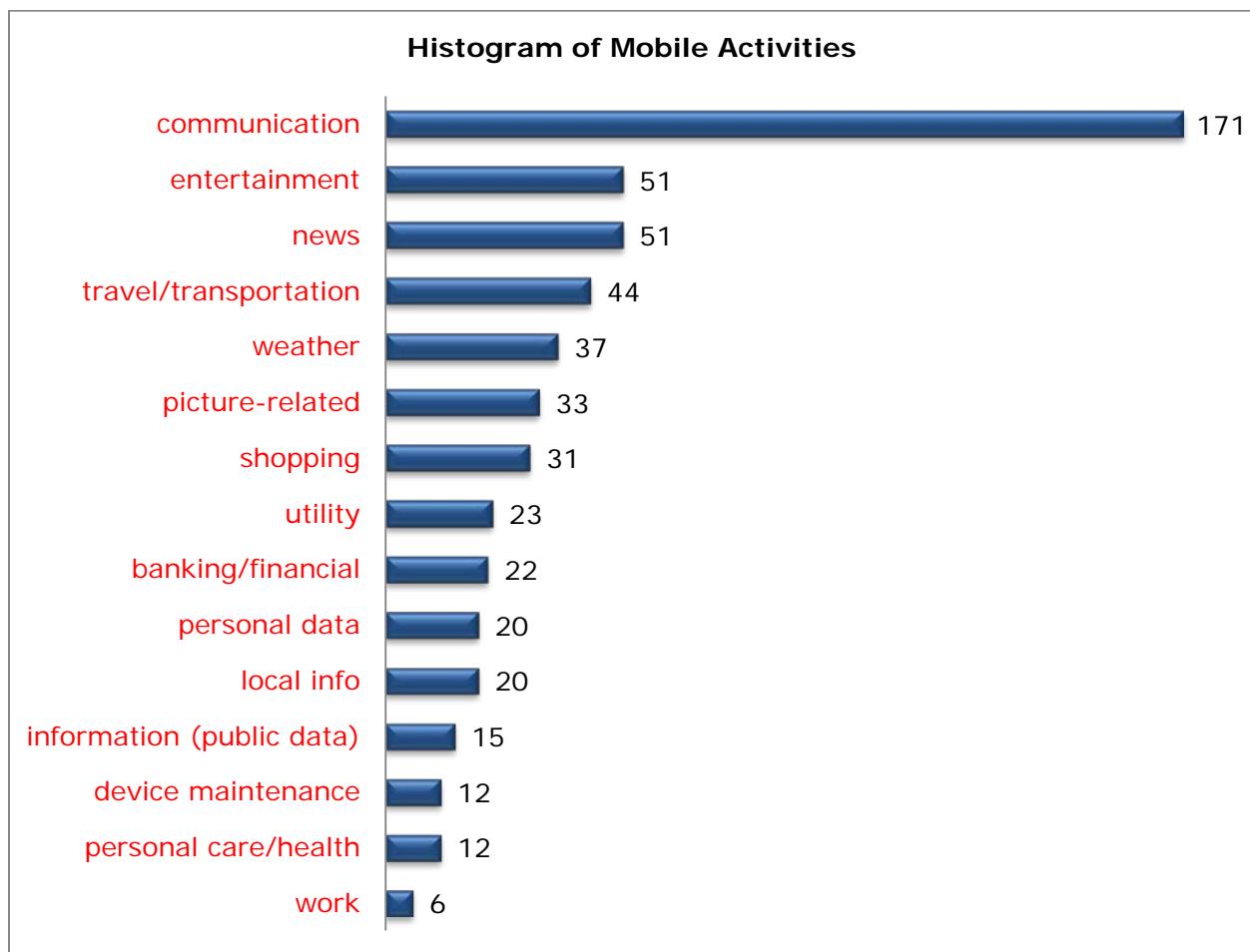
Sohn et al. was replicated later by Heimonen (2009)¹⁴. The gallery of information needs mirrored that from the earlier study (and thus confirmed the durability of the

¹⁴ Heimonen, T. Information needs and practices of active mobile Internet users. Mobility 2009.

research into intrinsic information needs); however, in Heimonen's study, 98% of the needs were addressed at the time of the need. Moreover, 94% of the needs were addressed using the mobile Internet. That is a huge difference (an increase by a factor of 5 compared to Sohn et al.'s users), and is a testimony to the tremendous growth of the mobile web in a short span of time. It is also a testimony to users' trust in their ability to find an answer using the mobile web.

MOBILE ACTIVITIES

The following graph shows a histogram of the mobile activities in our two diary studies, combined:



In this chart, the numbers indicate how many times each mobile activity was performed across the 172 person-days for which we collected a diary. Shopping includes purchasing, but also activities like price comparison and checking order status.

By far, the most frequent activity for our users was communication (including email, social networking, and forums) — consistent with other third-party large-scale

surveys¹⁵. Out of that category, email had the biggest chunk: checking email is still the preferred way to kill time on the go. The next two most frequent activities (entertainment and news) are all **consumption** related and reflect the kill-time type of mobile use.

WHAT TASKS ARE IMPORTANT ON MOBILE

If you start with an existing desktop site, you can use site analytics and surveys to map out the goals and needs of your mobile users by using the segmentation feature in your analytics package to separate out visitors by platform. Figure out what mobile users are already doing on your desktop. Also, take a look at what is done most frequently on your full website. What are your top ten tasks on the desktop? Which of those are relevant on mobile, but maybe currently too difficult to keep mobile-users' attention?

Most of the time, your tasks will be highly particular to your industry, or even to your company, so you'll have to figure them out by yourself. Here is a list of activities and types of information likely to be needed on the go; these have to be included and prioritized on mobile:

- **Location information.** It's common knowledge that getting directions is a frequent mobile activity. Third-party surveys show that 49% of smartphone owners use location-based services¹⁶.
- **Rapidly changing content.** Examples of such tasks are sports scores, traffic, flight information, order status, movie schedules, or product-availability information (i.e., whether a product is available at a given store).
- **Event-related content.** Anything that may be needed when participants attend an event (e.g., conference, commencement, concert) should be available on a mobile device.
- **Emergency information.** Emergencies occur in all circumstances, and, often, the phone is the only available source of information for those involved. Content that pertains to emergency situations (e.g., what to do when you are stung by a bee, or when your child accidentally swallowed dishwasher detergent) should be readily accessible and readable on mobile.
- **Information related to contacting businesses** (location, business hours, phone numbers). The tasks that our diary participants did involved finding business hours or business locations, finding particular types of stores (e.g., pizza), finding the phone number of a restaurant.

A Dutch diary participant was trying to find stores that sell glasses on his mobile phone:

"We were going to buy glasses, so we wanted to know where the stores were located. [We] wanted to get an idea where we should go

¹⁵ For instance: Nielsen. "The Mobile Consumer: A Global Snapshot," 2013.

<http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2013%20Reports/Mobile-Consumer-Report-2013.pdf>

¹⁶ Maeve Duggan. "Cell Phone Activities 2013," Pew Internet, 2013.

http://www.pewinternet.org/files/old-media//Files/Reports/2013/PIP_Cell%20Phone%20Activities%20May%202013.pdf

to buy the glasses, and know the best route to walk to make it an efficient way to spend our time today."

An UK diary participant used her phone to look for bars that stayed open late:

"I was looking for bars/pubs that had a late Sunday license. [...] I managed find bars/pubs in the area and was able to call them to find out what time they closed."

- **Tasks that have a deadline.** In our diary studies, participants were more likely to do transactions on the phone when they were under time pressure. Examples include buying a gift at the last minute, paying bills during vacations, checking bank-account balances before writing a check.

A diary participant recalled how she used her bank site on the way to the grocery store:

"I was sitting in my car and I checked my bank account, and made sure I had money and didn't need to transfer from the other account I have [...] [The site] came up and showed that I had money, and, if it hadn't, I could have made the transfer right on the phone."

- **Tasks that require privacy.** According to data from 2007, the most frequent searches done on the Internet involve adult content¹⁷. While this finding reflected the infancy of the mobile web at the time, it also pointed out that small-screen devices are ideal for activities that we want to keep private. Although none of our participants mentioned adult content, several did note that they used their phone at work as a discreet way of checking personal email or doing nonwork-related tasks.

This is how one diary participant put it:

"I was at work ... I read some email and visited some [news] websites [on my phone]. My work usually tracks our web usage, so I try not to do some personal things on my work computer."

- **Communicating with others.** In our diary studies, we found that the most frequent activities performed on mobile devices involved some form of communication with others (see the section *Mobile Activities*). The most-encountered examples involved email, social network sites, and forums. In some situations, the communication was around a product: users emailed pictures of that product to friends or relatives in order to decide what to buy.

One of our male diary participants used email to decide which boot to buy:

"I was shopping for boots... I took photos and sent them to my wife to see which style of snow boot she preferred."

¹⁷ Kamvar and S. Baluja. "Deciphering trends in mobile search," *Computer*, 2007. See also Church, K., Smyth, B., et al. "A large scale study of European mobile search behavior," *MobileHCI '08*.

Mobile Limitations and Strengths

Mobile devices have transformed the way we live and conduct everyday activities. Not only can we access almost any type of content on mobile, but with most mobile smartphones today we can deposit checks, accept credit cards, order food and pay for groceries, sign digital documents, and even lock our house door and start the car. Some of these tasks have become easier because of the existence of this new technology: the smartphone.

Mobile phones come with strengths, but also limitations. These strengths and limitations play out in good mobile user experiences.

SMALL SCREEN

In spite of the modern trend towards larger-screen phones, what makes mobile phones so convenient and portable is their small size. Compared with desktop and even laptop screens, phone screens accommodate a lot less content. As a result, screen size is a serious limitation for mobile devices. The content displayed above the fold on a 30-inch monitor requires 5 screenfuls on a small 4-inch screen. Thus, mobile users must (1) interact more with the device in order to access the same amount of information; (2) rely on their memory to keep accessible information that is not visible on the screen.

- Consider the opportunity cost of each design element.

Whenever you include a new design element on the mobile screen, something else gets pushed out (or below the fold). Think hard of the opportunity cost of each new element: what does it mean for the users if you leave out element B in order to include element A? Is element A more important than element B? Although we provide general guidelines in this report, your answer likely depends on the kinds of users and tasks that you have.

- Prioritize content over chrome on mobile.

“Chrome” denotes the user-interface elements that are instrumental in using a site or application¹⁸ such as buttons and task bars. Users come to a site to find information that they need or to accomplish a task, not to contemplate the beauty of buttons, navigation, menus, and other design elements. Content is always of interest (whether on mobile or on desktop), but whereas on desktop there is enough screen space for both content and chrome to coexist, often, on mobile, designers must downplay the chrome to make space for essential content.

That doesn’t mean that chrome should disappear from mobile. In fact, as we discuss in our section Gestures, it’s hard to create a usable interface with no chrome. However, designers need to accommodate a high **content-to-chrome ratio**¹⁹ on the mobile screen.

¹⁸ More on chrome: Jakob Nielsen. “Browser and GUI Chrome.”

<http://www.nngroup.com/articles/browser-and-gui-chrome/>

¹⁹ See also Raluca Budiu. “Maximize the Content-to-Chrome Ratio, Not the Amount of Content on Screen.” <http://www.nngroup.com/articles/content-chrome-ratio/>

PORABLE MEANS INTERRUPTIBLE

Mobile phones are **portable**: most fit easily in a pocket or purse and they tend to follow us everywhere. Because we use phones in a variety of contexts and situations, we are more likely to be **interrupted** when using such devices: an external event in the outside environment may demand our attention and require us to stop whatever we were doing on the small screen. As a result, **attention on mobile is often fragmented** and **sessions on mobile devices are short**. In fact, a CHI study²⁰ found that the average mobile session duration is 72 seconds. In comparison, our own studies show that on desktop, the average session is 150 seconds: more than twice as large.

Short mobile sessions have significant design implications:

- Design for interruptions. Save state for users and allow users to save state.

Designers should save context and make it easy for users to recover context and resume an interrupted task. The mobile app or website must save state at all times and be prepared for such interruptions. It should also try to do the transition back to the app/website as smooth as possible, so that the user doesn't have to redo work already done before the interruption.

Moreover, mobile users don't always make definitive decisions, but may want to return to certain content in contexts with larger bandwidth or screen. Allow users to save history, as well as to email or share information with themselves or others. And also allow them to return to their data on other platforms and access any actions they may have carried out on mobile.

- Prioritize the essential.

Because attention is fragmented, strive to show users what they need as soon as possible. Flooding them with details and asking them to parse walls of text for relevant facts is not interruption friendly. The gist should always come before the minutiae.

- Simplify tasks and interactions.

A simple task is easier to finish quickly. It's also easier to resume than one with many steps and alternatives.

SINGLE WINDOW

Although some phone manufacturers are trying to accommodate multiple windows on the screen at the same time, the limited size of the mobile screen makes that goal quite unpractical, even with today's larger-screen phones. The vast majority of users only see a single window (and thus a single application or website) at a time; they cannot split the screen (as on the desktop) and work with two different apps simultaneously.

²⁰ M. Bohmer, B. Hecht, J. Schoning, A. Kruger, G. Bauer. "Falling asleep with Angry Birds, Facebook and Kindle—A large scale study on mobile application usage," Mobile HCI 2011.

- Design for a single window. Don't ask users to go elsewhere for information.

Any mobile tasks should be easy to complete in a single app or on a single website. Users should not have to leave an app (or website) to find information that the app requires, but that it doesn't provide. Remember that pen and paper, even if available, are often unusable on the go. If users must move information from one app to another, it's likely that they will need to copy-and-paste it (or worse, rely on their memory); the interaction will become more complex and error prone. Apps and websites should be self-sufficient and should not necessitate any external props, be they physical or virtual.

TOUCHSCREEN

Touchscreens come with many blessings and sins. Gestures represent a hidden, alternate user interface (UI), that, when built with the right affordances can make the interaction fluid and efficient and can save screen real estate. On the other hand, it's hard to type proficiently on a tiny virtual keyboard and it's easy to accidentally touch the wrong target.

- Minimize typed input.

Perhaps the biggest problem is related to typing: on a soft keyboard, users need to continuously divide attention between the content that they are typing and the keypad area. Touch typing is impossible in the absence of haptic feedback; plus, keypads themselves are small and keys are crowded.

- Create large, spaced touch targets.

Another difference between touch and other types of input such as mouse is that the target size required to optimize the reaching time and minimize errors is considerably larger for touch than for mouse. So not only is the screen smaller than a laptop or desktop screen, but buttons and other targets need to be larger than those on a regular monitor.

- Give users a way to undo any action.

Because on a touchscreen there can be many target areas, it is easy to make accidental touches. Some can leave the user disoriented and unsure of what happened. Undo is one of the original 10 usability heuristics²¹, and it is even more important on touch devices.

VARIABLE CONNECTIVITY

Even in the era of fast cellular networks and ubiquitous Wi-Fi, coverage is not universal or equally good. Phone users frequently complain about connectivity problems. Every new page load translates into a significant waiting time when the network does not cooperate.

- Minimize the number of page loads and the load size.

²¹ See Jakob Nielsen. "10 Usability Heuristics for User Interface Design." <http://www.nngroup.com/articles/ten-usability-heuristics/>

If you want users to finish their tasks on your mobile site or in your app, mind the waiting time. Design pages that are light, yet contain as much information as possible, to avoid many back-and-forth trips from client to server.

GPS, CAMERA, ACCELEROMETER, VOICE, AND OTHER PHONE FEATURES

Phones come with many limitations, but also with strengths. The camera, microphone, and GPS are conveniently integrated into the device and can be easily used²² to make input easier and get around some of the difficulties of typing. Photographs can transmit more nuanced information that often cannot be easily captured in text (think of describing a product that you are looking at). Notifications enable users to be updated immediately of events that are relevant to them. Touch ID allows users to log in using a fingerprint, without typing passwords. And Apple Pay and Google Wallet enable users to use their phones to pay in real life or online, without entering a credit card.

- Take advantage of the phone features to save user effort and add convenience.

If the phone comes with a camera, don't ask users to type in barcodes. If the phone has a GPS feature, don't have them enter zip codes. Use the phone features as much as possible to lessen users' work.

²² Some of these features (e.g., camera, notifications) can only be used in an app. Others such as the GPS are also available on websites.

Mobile Website or Mobile App?

Once you have decided to create a mobile platform, the next question that often arises is: should you build an application (or maybe even multiple apps) or should you create a mobile website?

There are pros and cons for each option²³.

- **Applications may allow users one-touch access to your service.** Usually one click is enough to start an app (compare that with entering a website address in the browser). This is mostly true for frequently used apps or apps positioned on the first phone screen.

This is what a user in our diary study had to say about a YouTube application:
"I like that it's right on my screen, [you can] hit a button and search what you're looking for. I don't have to go to [my] Safari [browser] to find it."

However, accessing a rarely used app can be as difficult as finding a website. One reason is that people tend to have a lot of apps on their phone. When asked to use an app that they had installed on their phone, many of our study participants did not remember that they had it or confessed that they had never used the app before. Even if people may have used the app before, they often have trouble finding it in an app list that spans several screens. When apps can be grouped in folders, the extra level of organization can paradoxically make the finding process more tedious. Was this app under "Utilities" or was it under "Lifestyle"? Often people end up resorting to search in order to find an app on their phone. And search is also the main method of access for websites.

- **Applications provide closer integration with phone features.** For instance, they can take advantage of GPS, compass, accelerometer, camera, and call on utility apps such as maps, phone, and contacts. Some of these features (e.g., GPS, orientation info) have become available to websites as well, but it's still unlikely that mobile websites will achieve the same level of integration with the device as mobile apps.

One of our study participants talked about the ShopSavvy application for Android. ShopSavvy, a price comparison application, saved typing by allowing users to take a picture of a product barcode.

"[I would not use] the web directly, I would not use my browser [to shop for a better price]... I would actually use my ShopSavvy app. I don't have the product with me [...] I could search for a product ...It's a pretty cool interface, see I could scan a barcode or I could use the keypad to search for it [...] And, then, I could go browse the website or I could email myself a link right off the bat."

- **Applications can work when the phone is offline.** As people move around their day, the quality of the cell signal or the availability of a wireless network varies. Applications often pull data from a server when the phone is connected and use it offline. Some apps store data on the phone and require few (if any) server accesses. Users appreciate being able to use their phone when they have poor or no connectivity.

²³ See also Raluca Budiu. "Mobile: Native Apps, Web Apps, and Hybrid Apps." <http://www.nngroup.com/articles/mobile-native-apps/>

- **Applications require platform-specific development.** As companies develop applications for various devices, they need to train staff and have potentially different development teams and processes for different platforms. It can be more costly to create several mobile apps versus a single mobile website: the same website can work on an iPhone, on an Android phone, and on a Windows phone.
- **Users need to install and update apps.** Although there are app aficionados who may fish for apps in app stores, the majority of users don't like installing and maintaining apps (and also wasting space on their device), and will install an app only if they expect to use it often. Most people are fairly familiar with installing apps on their phone, but they don't necessarily update their apps religiously and often think of app maintenance as a chore. In contrast, websites do not need to be downloaded, upgraded, or updated by users; the maintenance work is entirely delegated to the website.

Maintaining a native app can be complicated not only for users, but also for developers (especially if they have to deal with multiple versions of the same information on different platforms): Changes have to be packaged in a new version and placed in the app store.
- **Apps are less discoverable than websites.** Users have to know first about an app in order to install it; moreover, they need to search for the app in an app store (that often has questionable usability) and install it. In contrast, everyone can access a website — whether they had heard about it or not (via a search engine, which tends to have great usability).
- **The content on websites is also more discoverable than the content in apps.** When people have a question or an information need, they go to a search engine, type in their query, and choose a page from the search results. They do not go to the app store, search for an app, download it, and then try to find their answer within the app. For instance, a search for "The Amazing Spiderman" on Google will return a list including the Wikipedia page of the movie, the IMDb page, the Fandango page, and so on. Even if the user had all three apps on her phone, she would still have to launch them separately and navigate to "The Amazing Spiderman" page in each of the apps. On the web, all these are directly accessible from the search-engine results page²⁴.
- **Development cost is usually higher for apps.** It's arguably cheaper to develop web sites, as these require skills that build on previous experience with the web. We often find that going fully native is a lot more expensive, as it involves more specialized talent. But, on the other hand, HTML5 is fairly new, and good knowledge of it, as well as a good understanding of developing for the mobile web, also requires fairly advanced skills.
- **User interface.** If one of your priorities is providing a user experience that is consistent with the operating system and with the majority of the other apps available on that platform, then native apps are the way to go. That doesn't mean that you cannot provide a good user experience with a mobile website — it just means that the graphics and the visuals will not be exactly the same as those with which users may be already accustomed.

²⁴ Search engines are starting to present deep link to apps that users have installed on their phone, so this advantage of websites is likely to vanish in the future.

- **iOS applications need to go through an approval process before they become public.** iPhone apps need to be approved by Apple before becoming available in the Apple's app store. Although users potentially benefit from this process (by experiencing a consistent look-and-feel across apps), in practice, it also means that the app's launch in the app store can be delayed and possibly denied.
- **iOS apps need to abide by Apple's in-app purchasing rules, as well as by acceptable content rules.** In iOS apps, Apple receives 30% of the price of any in-app purchases of digital content²⁵. Many content publishers dislike having to share part of their revenue with Apple, so they sometimes try to create web apps that replicate more closely the experience of a regular app, but are, in fact, webpages.

WHEN SHOULD YOU BUILD AN APP?

- **Apps are for repeat users.** Installation, in particular, forms a high barrier to adoption. Because the cost of installing an app is quite high (in terms of time, but also space taken on the device), users will not bother to do it unless the estimated benefit from the app is on par with the cost.
Even if a user may have heard of an app, it's unlikely they will install it if they estimate a one-time use. Mobile information needs rarely lead to apps being installed: when people have specific information needs (as they often do when they are on mobile), they look for answers on the web. The exception is when users expect to have that same need over and over again. For instance, if every week they order pizza from the same place, an app is probably more convenient than a website.
Because of that, we usually advise to design apps for repeat users: for people who are familiar with your company and will engage with your app repeatedly. Consequently, it's highly important to personalize the app to the user and save history and state, so that users can perform their usual tasks as fast as possible.
- **Applications are best when offline access is needed.**
Although some caching is available in HTML5, if functionality needs to be provided when the phone is not connected to a cellular or Wi-Fi network, applications provide the best solution. Thus, it makes more sense to build an application and not a website for services provided during a natural disaster or for services likely to be necessary in areas with spotty coverage (e.g., ski resorts, national parks).
- **Applications are better for supporting in-context or in-the-field use.**
To some extent, all mobile use is in-context use. Yet, if you provide functionality that supports users as they interact with the external world (for instance, during exercising, cooking, repairing a car or an appliance, or interacting with patients), an app is usually more appropriate than a website. An app allows designers to create interfaces that are specific to the device, while also taking advantage of phone features. It also enables designers to focus on a few tasks that can be well supported by a simplified interface.

²⁵Apple. "Getting Started with In-App Purchase on iOS and OS X"
<https://developer.apple.com/in-app-purchase/In-App-Purchase-Guidelines.pdf>

- **Apps typically have better UI.** In our own research we observed that overall apps tend to have more usable UIs than mobile websites. This should be no surprise: whereas websites are generally intended to cover a multitude of platforms, apps are platform specific and their interface tends to be more tailored to the device and to better incorporate platform conventions and features. Also, apps tend to have simpler, more streamlined functionality and fewer tasks that they incorporate.
- **Apps are better suited for integration with external devices.**

With the advent of smartphones, we have seen a plethora of technology that interacts with them either occasionally or on a repeated basis. Devices used during exercising (e.g., Fitbit) or intended to maintain posture (e.g., Lumo Lift), credit-card readers (e.g., Square), thermostats (e.g., Nest), smart locks (e.g., August) and others are better supported by an app, especially if there is frequent communication between the device and the phone that needs to be supported independent on network connectivity.

- **Apps are better for taking advantage of phone features and personalization.**

Although features such as GPS and voice are available in the browser, others (e.g., camera, notifications, fingerprint authentication, defining complex gestures) can be fully taken advantage of only in the context of an app. Moreover, apps can also access data stored on the phone (email, contact list, calendar) and thus create a better, more personalized user experience.

- **Apps are better if speed is of essence.** Apps win the speed competition. In 2012 Mark Zuckerberg declared that Facebook's biggest mistake had been betting on the mobile web and not going native. Up to that point, the Facebook app had been a hybrid app with an HTML core; in 2012 it was replaced with a truly native app. (See our section *Hybrid Apps* for more information about hybrid apps.)

MOBILE WEBSITES

On the web, not all websites are created equal. On a mobile device, users can encounter one of the following types of sites:

- **Full sites** or **desktop sites** are designed for the desktop and are not mobile optimized.
- **Mobile-dedicated sites** are designed for mobile phones.
- **Web apps** are a special type of mobile-dedicated site that looks and feels like an app.
- **Responsive-design sites** are sites designed for a multitude of devices with different screen sizes; they automatically adjust the layout of their content to the available screen size.

Full Sites on Mobile?

Users sometimes say that they'd rather go to a desktop site than to a mobile site. That's mostly due to their prior experience with mobile-optimized content: In an attempt to make the content more digestible, some sites include only a tiny subset of the full-site offerings on their mobile site. And sometimes people may be so used

to the full site that they can use this prior knowledge to figure out their way around on a small screen. (For instance, a Yahoo! user told us that he preferred the full site on mobile because he had been visiting it every day on his computer and knew where each piece of content was.)

Finally, users occasionally declare that the mobile site is dumbed down: it's too simple and impoverished. One of our participants was trying to make a reservation on MGM Grand Hotel's mobile site. The first thing she said when she saw the site was that it was very barebones, and she expected a flashier website from MGM (the biggest casino hotel in Las Vegas). However, she was able to finish the reservation quickly. In the end, she came to appreciate the simplicity of the site and was pleasantly surprised at how easy it was for her to complete the task.

The bottom line is: You should not listen to what users say, but rather look at what they do²⁶. When people use mobile sites, they typically are more efficient and more successful. But when you ask them whether they prefer mobile sites, they might tell you otherwise.

A Note on Phablets

One of our studies involved exclusively phablets, that is phones with screens larger than 5.3 in. We were curious whether people's behavior or sites and apps' usability issues are different on larger screen. The larger screen did enable participants to read better, and also allowed some of them to use desktop site more often on mobile. Whereas some of our phablet participants consistently preferred desktop sites, the usability of these sites on the (still small) screen is far from good, and people struggled with small targets as well as with the tiny font.

Overall, while on large-size tablets (iPad like), full sites work decently and a small number of minor adjustments can make them quite usable, on phablets they are very much a strain. For this reason, we don't recommend that you send your phablet users to your desktop site.

Mobile-Dedicated Sites

Mobile-dedicated sites are sites designed specifically for mobile phones. They often live under a separate URL (e.g., m.site.com) and are completely distinct from the full site. They contain features or content that have been deemed appropriate for mobile; frequently, these are just a subset of what is available on the desktop. They are often contrasted with responsive sites, which typically contain the same content and functionality for mobile and desktop, but rearrange these features differently on mobile.

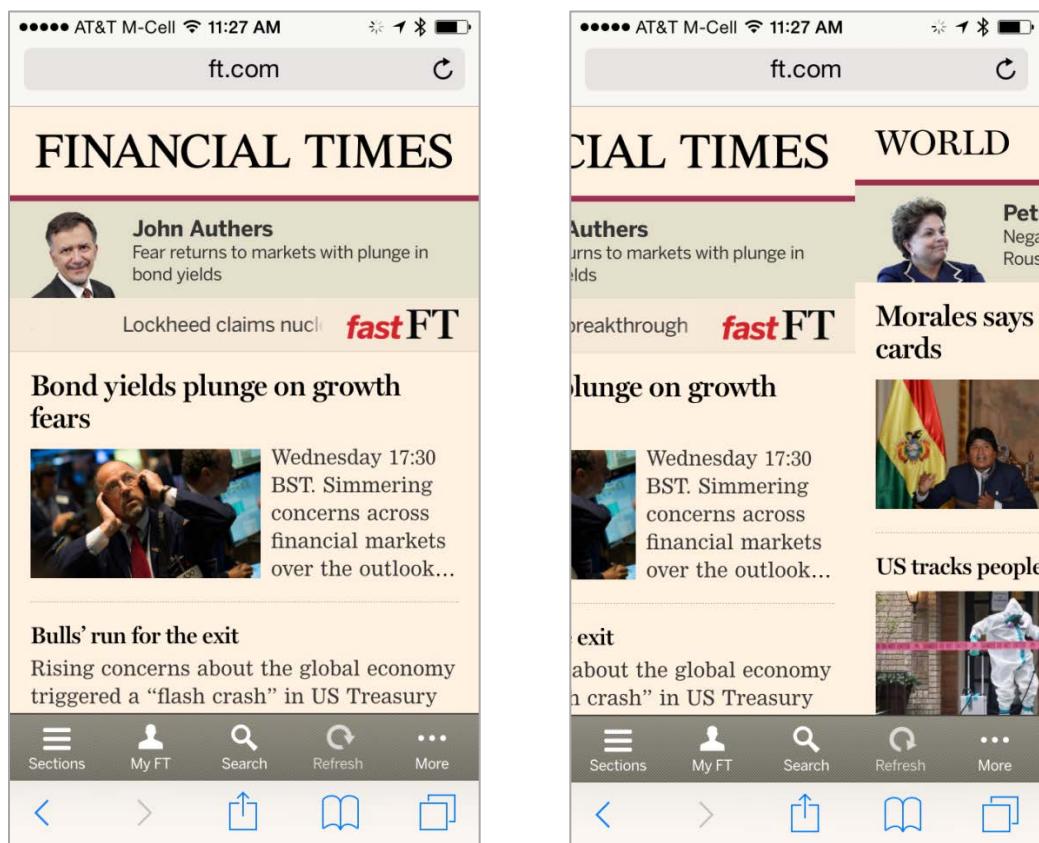
Web Apps

Web apps are not real applications; they are really **websites** that, in many ways, *look and feel* like native applications, but are not *implemented* as such. They are run by a browser and typically written in HTML5. Users first access them as they would access any web page: they navigate to a special URL and then have the option of "installing" them on their home screen by creating a bookmark to that page.

²⁶ See also Jakob Nielsen. "First Rule of Usability? Don't Listen to Users." <http://www.nngroup.com/articles/first-rule-of-usability-dont-listen-to-users/>

Web apps became really popular when HTML5 came around and developers realized that they could obtain native-like functionality in the browser. Today, as more and more sites use HTML5, the distinction between web apps and regular web pages has become blurry.

In 2011 *Financial Times* withdrew its native app from Apple's App Store to circumvent subscription fees and maintain closer connection with their subscribers. Instead, FT came out with an iPhone web app (app.ft.com). This web app is, in many ways, hard to distinguish from a native app. For instance, the app-specific tab bar is shown at the bottom of the page and users can swipe horizontally to move on to new sections of the app. And, due to browser caching, it's even possible to read the newspaper offline.



Financial Times web app for iPhone uses a tab bar and horizontal swiping.

Responsive Design

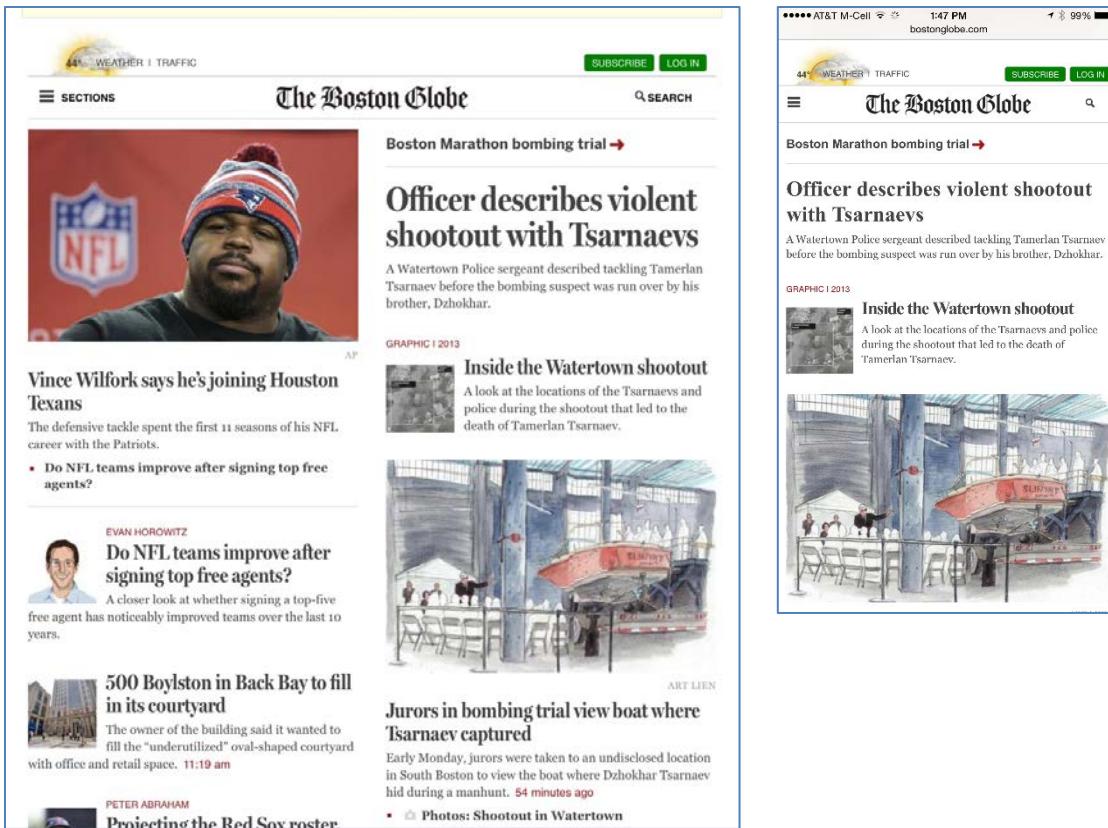
Responsive design is a development technique that detects the client type and dynamically adjusts the layout of a site according to the size of the screen on which it is displayed²⁷. Thus, the same content may be displayed in a three-column format

²⁷ More on responsive design: Amy Schade. "Responsive Web Design (RWD) and User Experience." <http://www.nngroup.com/articles/responsive-web-design-definition/>

on a desktop, two-column format on a tablet, and one-column format on a smartphone.

The screenshot shows the desktop version of The Boston Globe website. At the top, there's a navigation bar with links for BOSTON.COM, CARS, JOBS, REAL ESTATE, weather (44° Mostly cloudy), and LOGIN. Below the header, there's a main news article about Vince Wilfork joining the Houston Texans, featuring a large photo of him wearing a Patriots beanie. To the right of this, there's a sidebar for the Boston Marathon bombing trial, a travel advertisement for Expedia, and a link to a Marathon trial podcast. The central column contains articles about the Watertown shootout, 500 Boylston, and the Red Sox roster. The right column has sections for opinion pieces by Joan Vennochi and John E. Sununu, and a letter from Kevin Cullen. The bottom of the page features a footer with copyright information and a red navigation bar containing the number 31.

Bostonglobe.com is a responsive site. On the desktop the content was formatted across 3 columns.



The tablet (left) and mobile (right) versions of BostonGlobe.com showed the desktop content in two columns and one column respectively.

One of the complaints against mobile-dedicated sites is that they often exclude content and functionality that may prove relevant at least to some users occasionally. Responsive design tackles that objection by striving for content and feature parity across different versions of a site.

In practice responsive design is often a continuum: many responsive sites are not “fully” responsive and do not have a 100% feature or content parity; instead, they do remove functionality that is rarely needed on mobile.

authentic jobs®
since 2005

ALL JOBS POST A JOB PRICING ABOUT US OUR GUARANTEE Contact Us Sign In

ALL JOBS (choose a category...)

Full-time Contract Part-time Freelance Moonlighting Internship

Jr. Front End Web Developer/ Designer
1105 Media A leading provider of integrated B2B information & media. Irvine, CA Full-time NEW

Production Designer Intern
Petite Alma Multidisciplinary design studio in NYC New York City Internship NEW

Web Developer with back-end experience
SNworks A digital media start-up inside a 100-year-old company East Lansing, MI Full-time NEW

PHP and Mobile Developer(Titanium)
Easy Mortgage Apps LLC Providing an ecosystem for the lending industry Salem, MA Contract NEW

Graphic/Web designer
WDStone & Associates We believe in what you do. Cookeville, TN Full-time NEW

Senior Web Developer
Quidsi One of the fastest growing ecommerce companies in thecountry Greater New York Area Full-time NEW

UX Strategist
The Advisory Board Company - Crimson Thinkers wanted. Austin, TX Full-time NEW

Mobile Engineer
CoSchedule Do what you love, love what you do! Bismarck, ND Full-time NEW

Usability Consultant
UW-Madison, Division of Information Technology Provides IT support and services to the UW-Madison & System. Madison, WI Full-time NEW

Senior Interaction Designer
CarMax A better way to buy used cars Richmond, VA Full-time NEW

Senior Visual Designer
Slacker Powered by Humans San Diego, CA Full-time NEW

Associate, E-mail Marketing
Lincoln Center for the Performing Arts, Inc. n/a New York, NY Full-time NEW

Web Developer
n/a Brooklyn, NY Full-time

POST A JOB
Starting at \$99 for 30 days

*** 100% MONEY-BACK GUARANTEE**

The job board for web professionals. Authentic Jobs is where companies and creative professionals meet to make a better web. Since 2005, qualified candidates have been applying for great opportunities at Apple, Facebook, ESPN, Sony, The New York Times, and many other companies big and small.

Sign up for highlighted jobs and discounts.

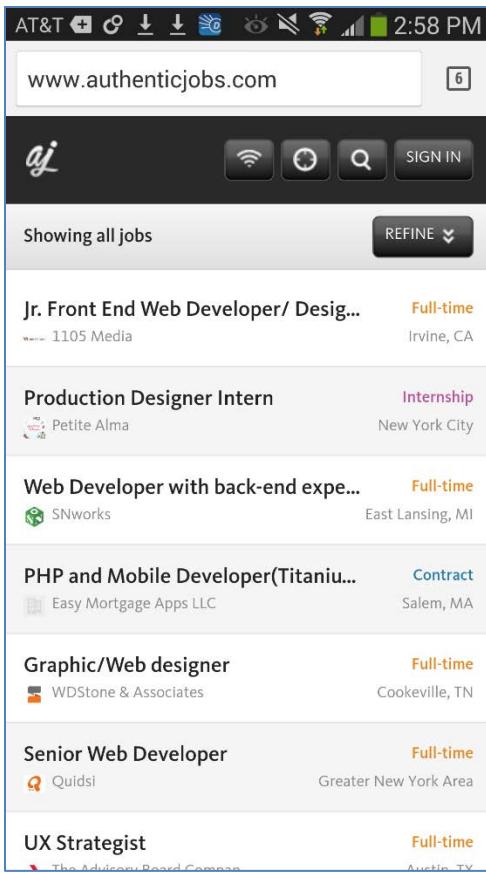
Perks
Just the perks from our listings.

AUTHENTIC JOBS for iPhone

Follow @AuthenticJobs

Like Share

Authenticjobs.com's responsive site: the desktop version included the ability to post a new job opening, which was missing from the mobile version.



Authenticjobs.com's mobile version: although the site is responsive, the mobile version did not include all the features available on the desktop.

For users there is no distinction between mobile-dedicated sites and responsive sites. They do not recognize responsive sites as being different than mobile sites, and they don't treat them differently. (The only way you can tell as an end user that a site is implemented in responsive design is to compare the same page on two or more different devices side by side. And that's not something anybody would do unless they're participating in our seminar on *Scaling User Interfaces*.)

Although their implementations may be poles apart, both responsive and mobile-dedicated sites need to follow the same mobile-usability principles and guidelines in order to be usable.

Mobile-Dedicated vs. Responsive Sites

Here are some of the relative advantages and disadvantages of these two approaches.

- **Responsive sites can support a variety of devices and screen sizes with a single implementation.** Dedicated sites are device specific: companies must build separate sites for mobile and for desktop. In contrast, the same responsive site can work well on a range of devices and screen sizes, from smartphones to tablets to desktops.

- **Responsive sites offer content and feature parity (at least to some extent).** Unlike for mobile-dedicated sites, the same content and functionality is available on all versions of a responsive site. No more need to decide which features are important on mobile and which should be left out. (Though you still need to prioritize features and decide on their placement on smaller screens.)
- **Responsive sites are easier to find with a search engine.** Mobile sites have a different URL than desktop sites, and they do not automatically inherit a high search rank from their sister desktop site. As a result, mobile sites may appear lower on search-engine page results. And even if desktop sites detect mobile clients and redirect the users to the corresponding mobile site, the redirect can take extra time and impair the mobile user experience (plus, it can also affect SEO).
Since a single URL corresponds to all versions of a responsive site, responsive sites can take advantage of the good ranking of an original desktop site and do not need to redirect users.
- **Responsive sites save content and feature maintenance.** A single site and a single content repository are easier to maintain than several separate sites. However, any interface change must be tested across all devices.
- **Responsive sites tend to be more expensive to develop.** Our clients report that the process of building an entire responsive site from scratch is more costly than just creating a separate mobile site. Also, the development skills required tend to be more advanced for responsive sites.
- **Responsive sites tend to be slower.** Although there are techniques for improving the performance of responsive sites, because the same content is delivered to all types of devices, loading a responsive page can take longer than loading a mobile-dedicated page.
- **Responsive sites work less well for complex tasks and content.** Complex tasks are hard to accommodate on all devices equally well. Complex spreadsheets, comparison tables, and visualizations are often difficult to rescale well on small mobile screens. Mobile-dedicated sites may often decide to leave out such content, especially since users tend to avoid doing complicated tasks on smartphones.
- **Responsive sites do not integrate well with existing third party services.** If you're building a site that relies on a separate independent backend service (e.g., booking engine on a hotel site), it's often hard to integrate the interface for that service into a responsive site.

Adaptive Design

Adaptive design is a version of responsive design in which the server detects the capabilities of a client device and only sends content and features that can be appropriately displayed on that device. More powerful devices receive more complex content that is enriched with CSS and JavaScript features accommodated by those devices. Less powerful devices on poor network connections are sent nimble, light versions of the page — stripped down to core functionality. This technique is sometimes called **progressive enhancement**.

The main advantage of adaptive design is that it solves the problem of slow response times that often plagues responsive design.

MOBILE APPS

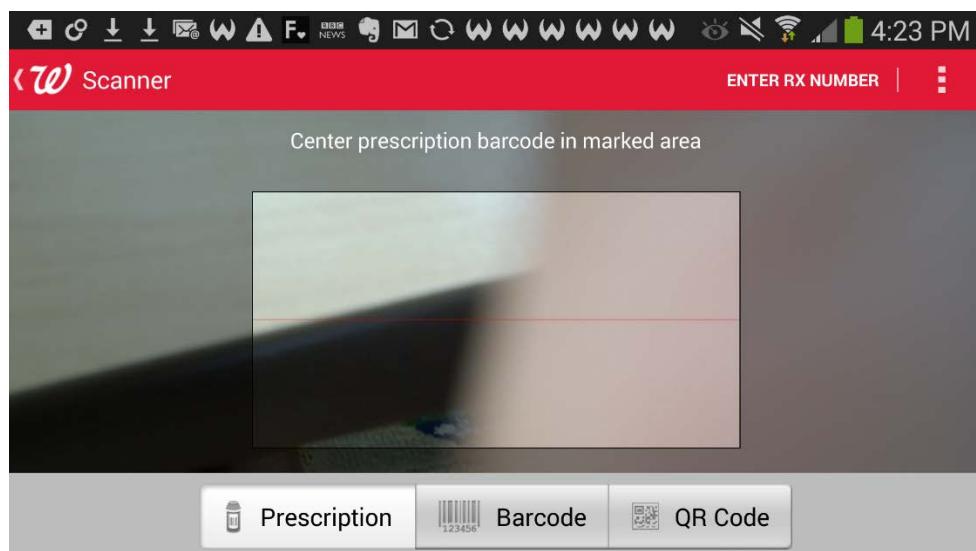
Native apps live on the device and are accessed through icons on the device home screen. Native apps are installed through an application store (such as Google Play or Apple's App Store). They are developed specifically for one platform, and can take full advantage of all the device features — they can use the camera, the GPS, the accelerometer, the compass, the list of contacts, and so on. They can also incorporate gestures (either standard operating-system gestures or new, app-defined gestures). And native apps can use the device's notification system and can work offline.

Hybrid Apps

Hybrid apps are part native apps, part web apps. (Because of that, many people incorrectly call them "web apps"). Like native apps, they live in an app store and can take advantage of the many device features available. Like web apps, they rely on HTML being rendered in a browser, with the caveat that the browser is embedded within the app.

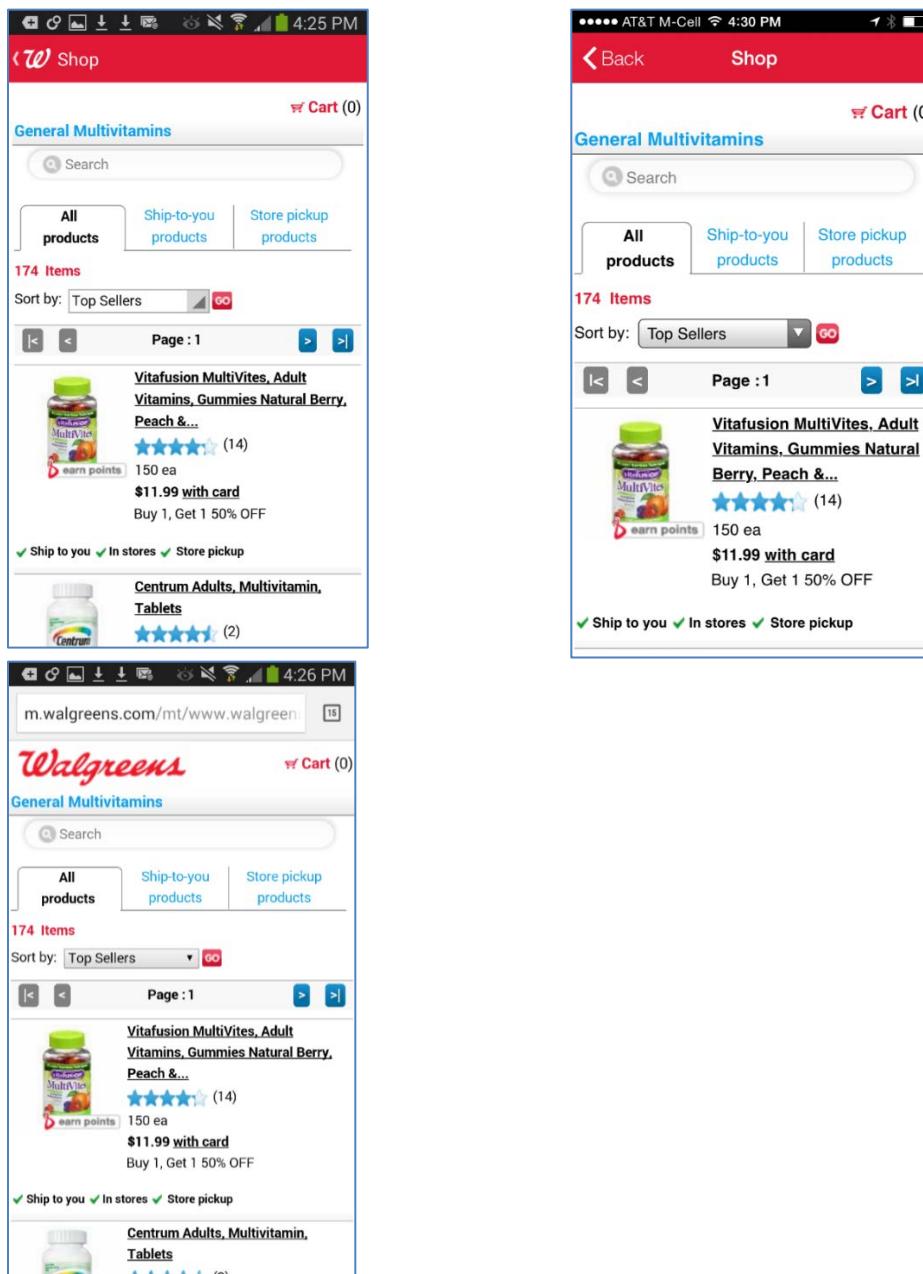
Often, companies build hybrid apps as wrappers for an existing web page; in that way, they hope to get a presence in an app store, without spending significant effort for developing a different app. Hybrid apps are also popular because they allow cross-platform development and thus significantly reduce development costs: that is, the same HTML code components can be reused on different mobile operating systems. Tools such as PhoneGap and Sencha Touch allow people to design and code across platforms, using the power of HTML.

Walgreens provides two very similar hybrid apps — one for Android and the other for iPhone. Both apps have multiple sections and many native features such as access to notifications and a *Refill by scan* feature that uses the phone camera to refill prescriptions:



Walgreens's hybrid app for Android includes native features such as the use of the camera for scanning bar codes.

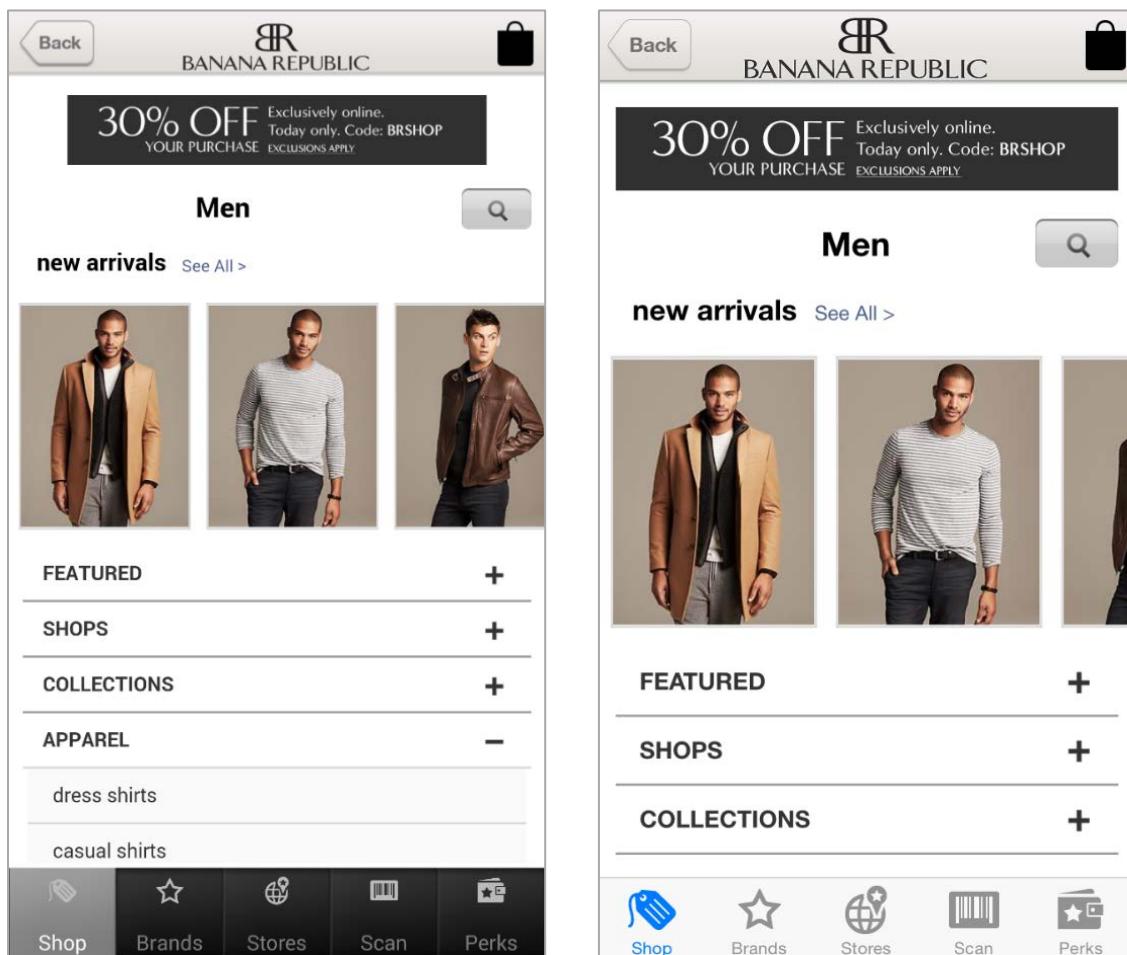
However, the *Shop* section in both the Android and iPhone apps uses a browser view that renders the corresponding page of the Walgreens mobile website. Here are three pages displaying the same content in the Android app, iPhone app, and mobile website:



Walgreens's Android app (top left), iPhone app (top right), and mobile website (bottom) all used the same HTML 5 core. The corresponding screenshots are identical, except for the top header in the two apps. Notice how the iPhone app appropriately showed a *Back* button in the top left corner.

As you can see, all these pages were the same, except for the top header, which was platform specific. The *Back* button on iOS was translated into a caret on Android; the logo was present on the web page, but not in the app. (The designers have correctly assumed that on the web people need the logo to orient themselves, since they are likely to land on a deep page without navigating through the homepage. In contrast, in their apps all navigation has to go through the homepage.)

Banana Republic is another example of a hybrid app; it has used the exact same design on Android and iPhone:



Banana Republic's hybrid app for Android (left) and iPhone (right)

However, the *Back* button in the Android app ignores the fact that, unlike iPhones, Android devices come with a *Back* button. The tab bar at the bottom of the page works well in the iOS design, but is clunky and clearly nonnative on Android.

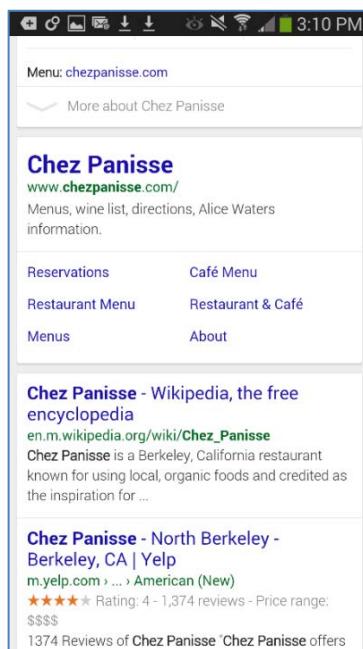
Mobile Usability Guidelines

MAKING YOUR MOBILE-DEDICATED WEBSITE FINDABLE

People use search engines to access sites, both on desktop and on mobile. They do so even in the rare situation when they know the exact URL of the site. Thus, instead of typing “amazon.com” in the URL bar, they go to Google and type in “amazon”. (This trend is further strengthened by the dying distinction between a URL bar and a search bar in most browsers, mobile or otherwise.) There are two big advantages to using a search engine instead of typing a URL:

- The interaction cost²⁸ is lower: It’s simply more efficient to type just a word and let the search engine do the rest of the work. As one user put it:

“I typically like to use Google or something like that to [get to a website] — just so that I type less ... You probably noticed I always do that instead of just trying to go to the website directly. I just try to minimize the typing strokes. And, then, there’s a better chance that I get what I want sometimes — when I’m texting I might misspell something, but Google usually interprets that pretty quickly.”
- Search-engine results often include deep links into the site. These help users get to the page they need faster. For instance, when searching for the website of the restaurant Chez Panisse, many of our users were able to get directly to the café-menu page from the search-engine results page (see the figure below).



Google search result for “chez panisse” included links to the different sections of the site. Users could click directly on the link labeled *Café Menu*, without going to the actual site.

²⁸For a definition of interaction cost, see Raluca Budiu. “Interaction Cost.”

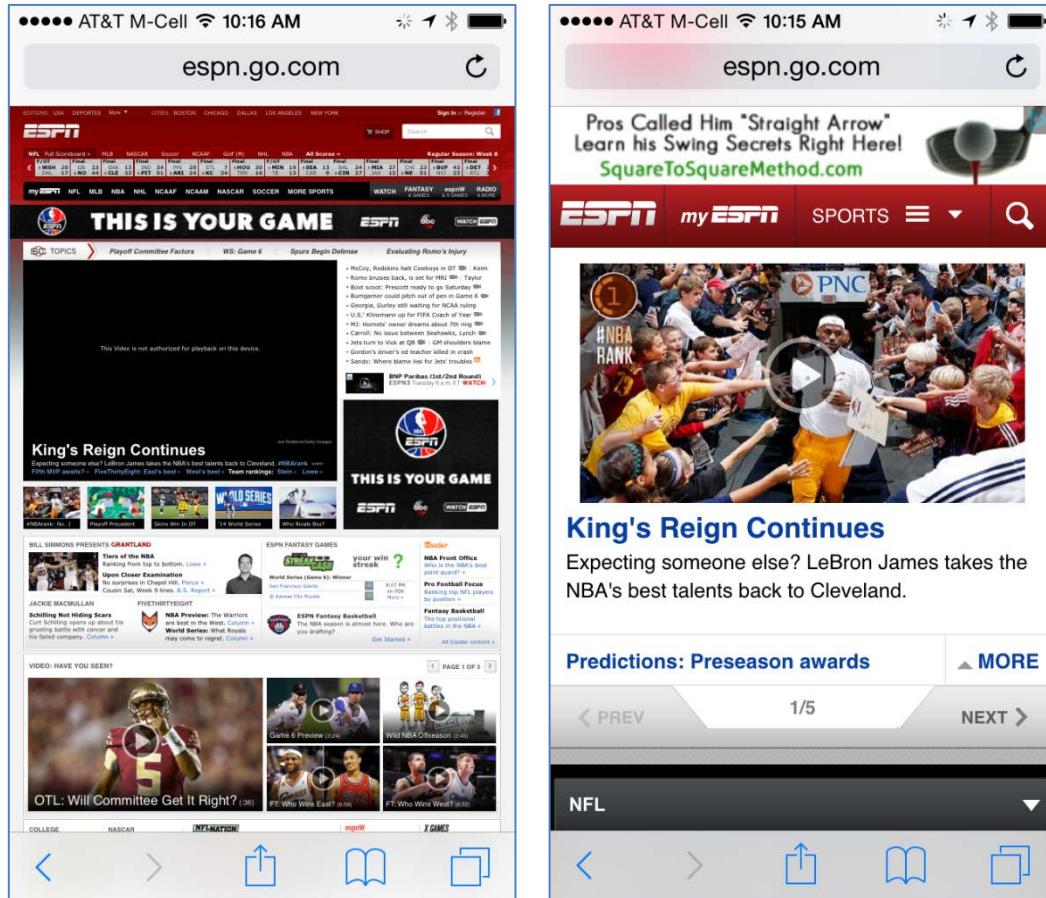
<http://www.nngroup.com/articles/interaction-cost-definition/>

Regardless of how mobile users access your site, you need to make sure that they get to a version that is appropriate for their device. If your site is responsive, the pages will automatically reflow to the size of the screen on a mobile device, so no special effort is required to get the user to the mobile version of the site.

However, if you have a mobile-dedicated site, it will have a separate URL than the URL of your desktop site and you will have to make sure that mobile users are taken to that version of the site.

- [Web] Redirect users to the mobile site if they access your site on a mobile phone.**

It is surprising that there are still sites that do not redirect mobile users to their mobile variant. Although much money and effort may have gone into that mobile site, users are left to struggle with the full site on their phone.



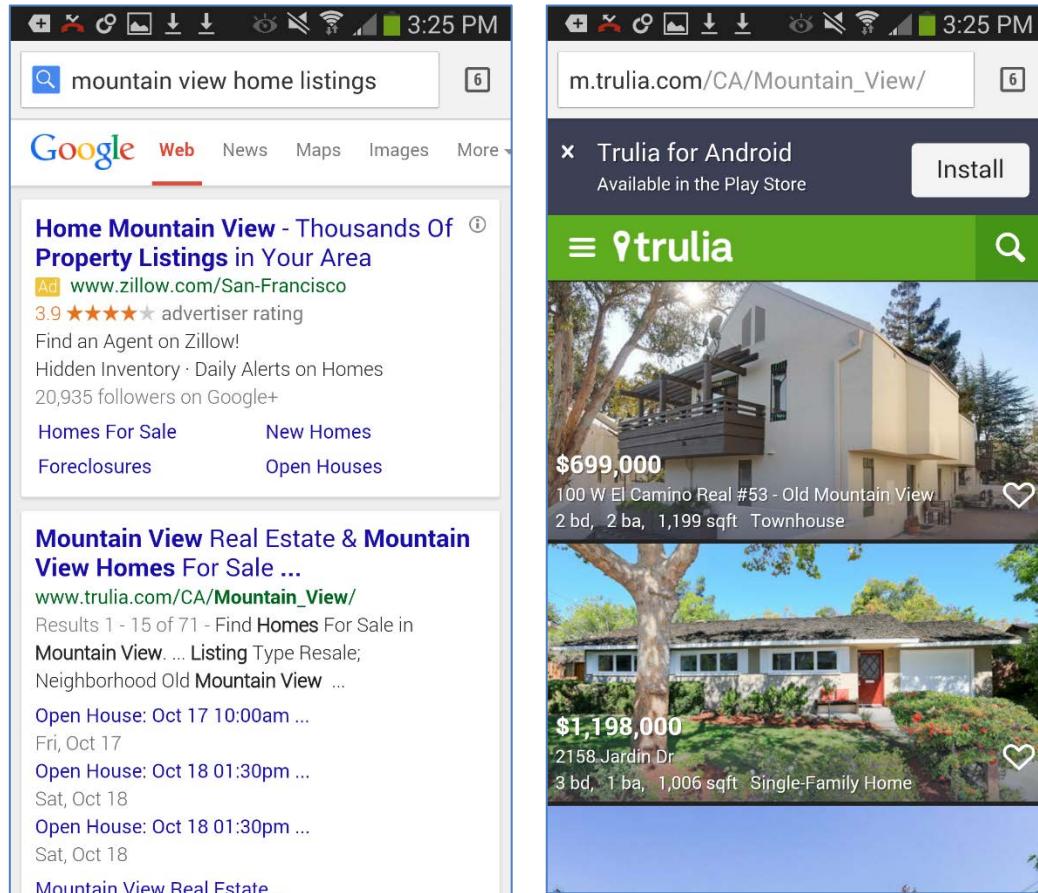
Although ESPN had a mobile site (m.espn.go.com), when mobile users typed “espn.go.com” in the browser, they were delivered the full site (left) and they were not redirected to the mobile site (right).

We are happy to report that this is an issue that has seen substantial improvement over time. It used to be that lack of redirects to mobile was the norm; nowadays, most sites have understood the importance of taking users to their mobile-dedicated versions. Cooperation from search engines might have

also sweetened the deal: search engines detect redirects to mobile and, in such cases, replace the full URL with the mobile URL in the search results. Users can click directly on the mobile page: they don't have to first navigate to the full page and then wait for the redirect.

2. [Web] Give users the mobile version of the requested page; do not always redirect users to the mobile homepage.

On mobile and desktop users depend on search to access websites. Often the search results will include deep links to a site. If that is the case, make sure than when users select one such result, they are redirected to the mobile version of that deep page, and not to the mobile homepage.



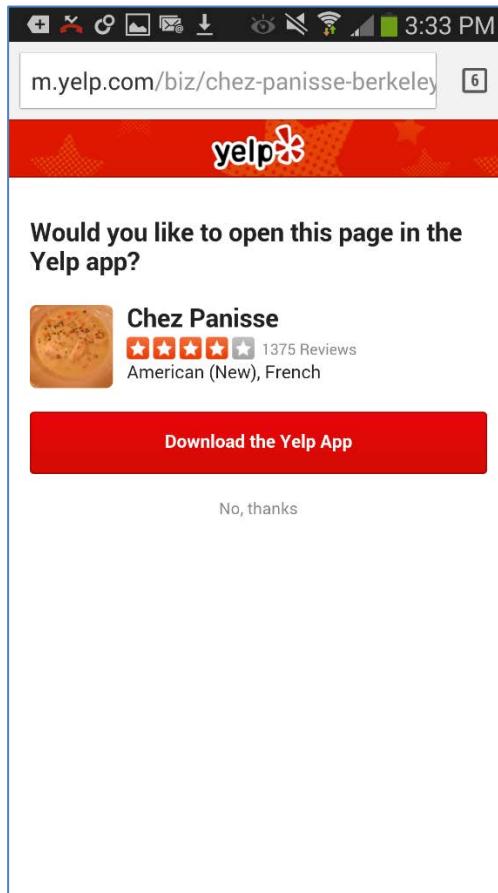
When users clicked on the second result on the Google search-results page, they were correctly redirected to a deep page on m.trulia.com that responds to their original query ("mountain view home listings").

3. [Web] Don't make users decide if they should use the full site, the mobile site, or mobile app. Always direct the users to the mobile site.

Most of the time, when users are on mobile, they want to get to the content as soon as possible. Deciding whether they need the mobile site, full site, or to

the app is counterproductive because it disrupts the continuity of the experience and increases users' cognitive load²⁹: users must stop what they were doing, make an unrelated decision, and, once they've made that decision, recover context in order to finish their task.

Instead of asking users' opinion, always take them to the mobile site and once on that site, give them ways to access the app or the full site (see the section *Linking to Full Site*).

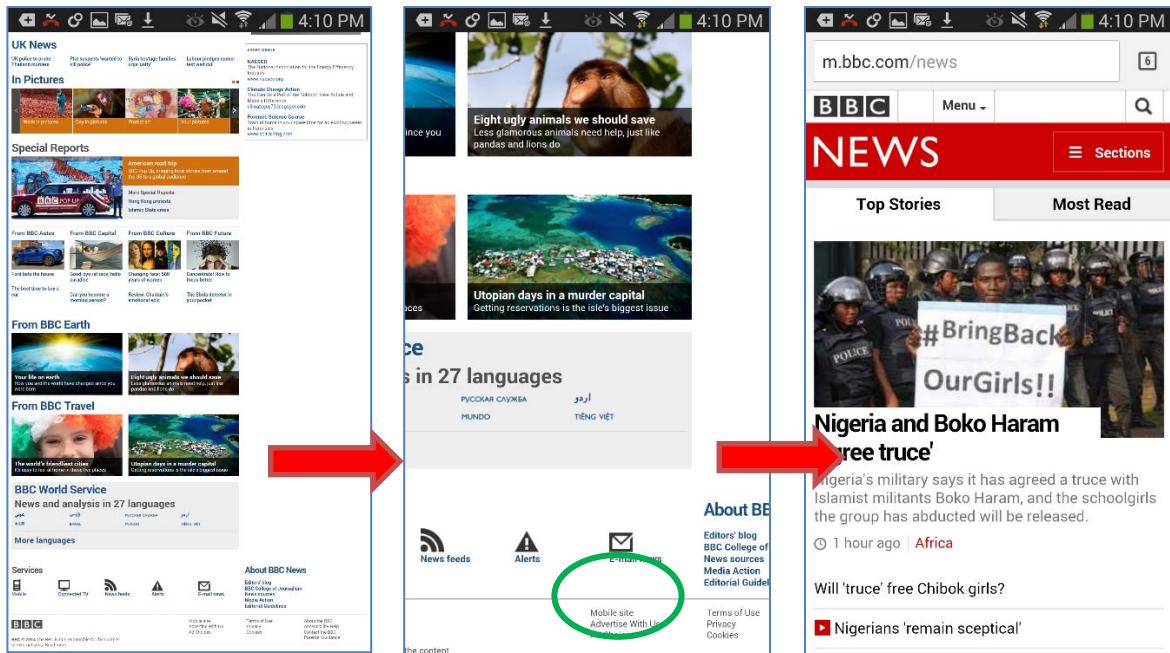


Yelp.com on Android: Don't force users to choose among different platforms.

4. [Web] On your desktop site include a link to your mobile site.

The link to your mobile site can help two categories of mobile users: (1) those who, for whatever reason, were not redirected to your mobile site; (2) those who chose to visit the desktop site (see guidelines 8 and 11) and who may need to navigate back to the mobile site.

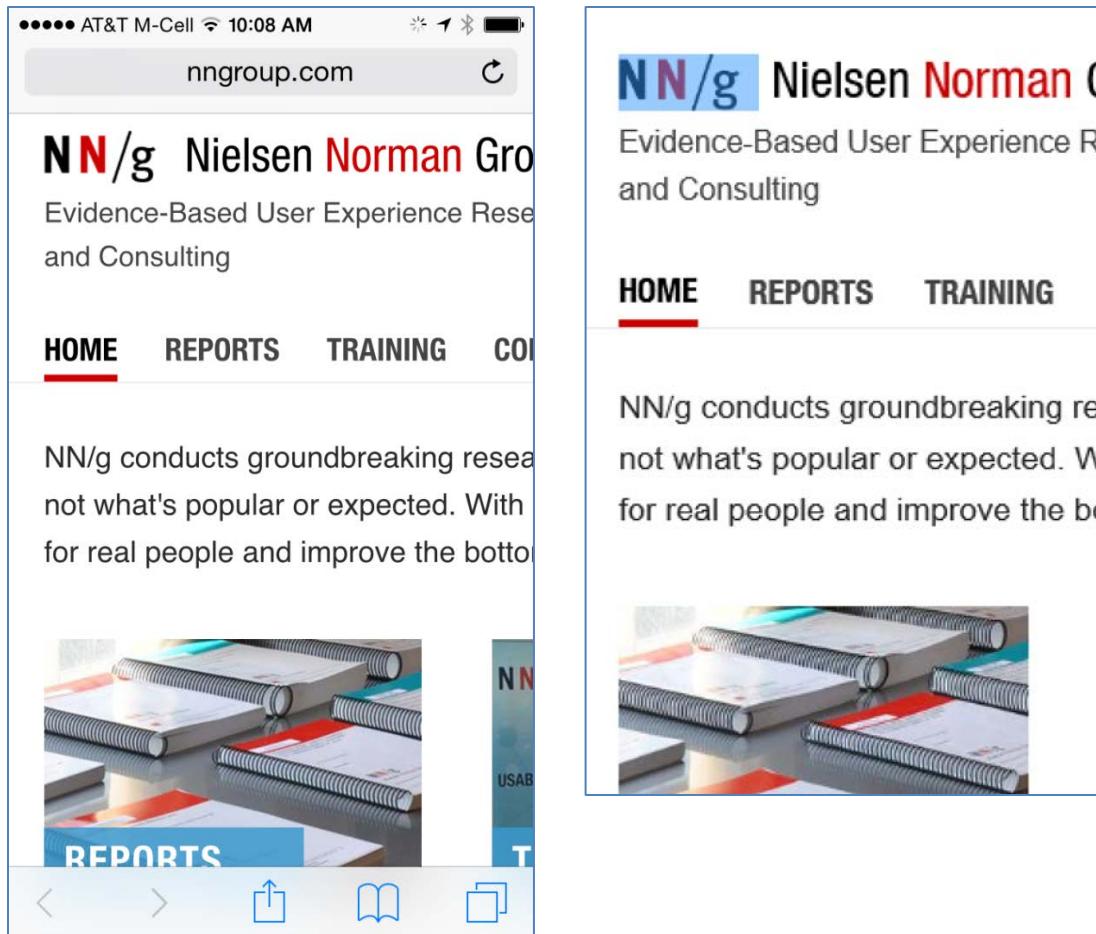
²⁹See Kathryn Whitenton. "Minimize Cognitive Load to Maximize Usability." <http://www.nngroup.com/articles/minimize-cognitive-load/>



BBC had a link labeled *Mobile site* on their full site. The link led to the mobile BBC site.

There is an additional, unexpected benefit from linking to your mobile site from your full site: mobile sites may make the desktop user experience better for people with accessibility issues. Full sites displayed on smartphones are not unlike full sites displayed through a screen magnifier. In both cases, to get access to details, the user needs to zoom in; in doing so, she loses the panoramic view of the page and she can easily get disoriented (see below).

Thus, access to mobile sites on desktop may benefit people with disabilities by allowing them to interact with a simpler version of the site, more suited for the limited viewport of the screen magnifier.



Left: NNgroup.com on iPhone, after pinch-zoom. Right: NNgroup.com through a screen magnifier on a regular PC (as would be seen by a low-vision user).

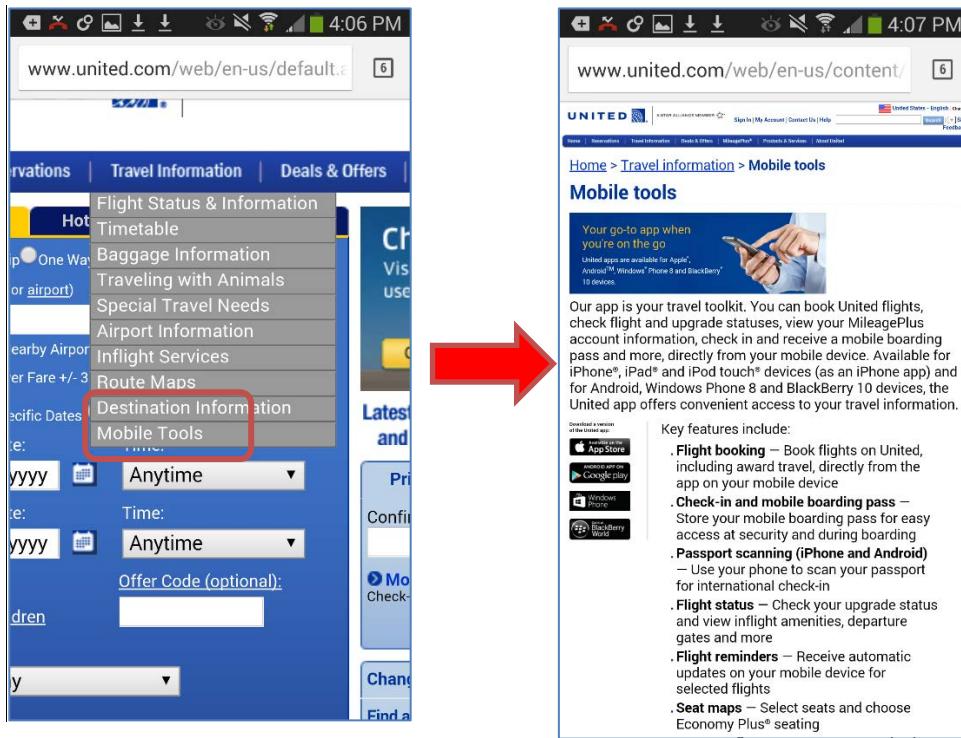
5. **[Web] On your full site, the link to the mobile site should be placed either in a very salient location (e.g., navigation at the top of the page) or in the page footer.**

With lack of redirects to mobile being such a common problem in the past, we used to recommend that the link to the mobile site be placed close to the top left corner. We now believe that accessing the mobile site has become easier, and that many desktop sites do a good job of directing the users to the mobile version. As a result, while we still recommend that you have a link to your mobile site and that this link not be hidden under menus or in deep pages, it's acceptable to have the link to the mobile site in the footer of the page (as in the BBC example above).

6. **[Web] The link labeled *Mobile* on your full site should not point to information about your mobile site.**

The link should correctly point to your mobile website.

United.com did not have a link to its mobile site on its homepage. There was a *Mobile tools* option as a second-tier navigation category hidden under a drop-down menu. That link pointed to a page with information about United's mobile apps and website. While it's a good idea to have a page detailing the different apps, the link to the mobile site should not have been buried on this page only. Users need to have an easier way to access the mobile site.



The full site United.com did not contain an easy-to-find link to its mobile site. Their *Mobile tools* link was buried under a menu item and led the users to a page with information about United's mobile apps and mobile webpage. The paragraph about the mobile site did contain a link to that site, but it was buried in the text.

7. **[Web] Standard domain names and URLs ([m.site.com](#), [mobile.site.com](#), [site.mobi](#), [www.site.com/mobile](#)) should all point to your mobile site. If you can afford only one of these domains, use [m.site.com](#).**

Some users may be tempted to go to the mobile variant of your site and type [mobile.yoursite.com](#), expecting this address to work. Do not force people to guess the name of your mobile site. Typing on a mobile device is hard already.

For some of our usability-testing tasks, we actually gave users the URL of a mobile site. We sometimes saw people abbreviate "mobile" to "m" — for instance, one of our users, instead of typing "mobile.fandango.com" (the URL that we gave to him), typed "m.fandango.com", expecting to get to the same site. Whenever the abbreviations did not work, users were annoyed to have to spell the entire word "mobile". That is one of the reasons for which we recommend using the URL [m.site.com](#), if only one domain can be used.

If you follow the guideline to make all the addressing variants work, autofocus all of them to one canonical URL. No matter what people type, they should end up in the same place. This can also be helpful for SEO and will reduce confusion.

Linking to Full Site

The guidelines below apply mostly to websites; if, however, your mobile app has content similar to your desktop site, follow the same guidelines for those apps as well.

8. **Include a link to the full site on the mobile-dedicated homepage.**

Because mobile-dedicated sites often support only a limited functionality compared to full sites, occasionally, some users will be looking for content that is not included on your mobile site, but does appear on your full site. To help people in those situations, link to your full site from your mobile homepage.

We've seen some users being disgruntled and suspicious of mobile sites, in general; such users prefer full sites because they think mobile sites have less content (see section *Full Sites on Mobile?*). Although we don't believe that the user experience is better on full sites on mobile, a link to the full site has the potential to satisfy these users.

Below are examples of how two different sites have included a link to their full page. On Southwest, an airline site, the link to the full site was visible with very little scrolling. On L.L.Bean's site (e-commerce site), the link was at the bottom of a long page.

The image displays two mobile website screenshots side-by-side for comparison.

Southwest.com (Left):

- The top navigation bar includes a user profile for "Gary Kelly", the "Southwest Rapid Rewards" logo, and a menu icon.
- The main content area features a large call-to-action button labeled "Enroll Now" with the text "Join Rapid Rewards® and start earning today." above it.
- A vertical sidebar on the left lists travel-related services: Check In, Flight Status, Book Travel, Manage Travel, and Special Offers.
- At the bottom, there are links to "Visit southwest.com" and "Privacy Policy", with "Visit southwest.com" circled in green.
- Copyright notice: ©2014 Southwest Airlines Co. All Rights Reserved.

LLBean.com (Right):

- The top navigation bar includes a menu icon, the "L.L.Bean" logo, and a "CALL" link.
- A modal window titled "Have an llbean.com account?" is open, containing a "Log In" button, a "create an account" link, and a "Close" button.
- The main content area features a search bar with the placeholder "Enter item # or keyword".
- A promotional banner for the "2014 OUTERWEAR COLLECTION" featuring an orange jacket is displayed.
- A sidebar on the right lists categories with plus signs for expansion: Men's, Women's, Kids', Footwear, Outdoor Gear, Hunting & Fishing, Luggage, Home, Outdoor Discovery Schools, Sale, Gift Cards, and Gift Shop.
- Links at the bottom include "Customer Service", "Find a Store", "Log In", "L.L.Bean® Visa® Card Coupon Lookup", and "Purchase History & Order Tracking".
- A footer section contains social media icons for Facebook, Twitter, and YouTube, and security logos for Norton Secured and PayPal.
- At the very bottom, there are links for "Security | Privacy | Promotional Terms | Recalls" and a phone number "888-797-3880 | Full Site", with "Recalls" and "Full Site" circled in green.

Southwest.com (left) and LLBean.com (right) both contained a link to the corresponding full sites. On Southwest.com the link to the full site was visible without almost any scrolling. However, the name of the link was ambiguous and did not clearly indicate that users will be taken to the corresponding desktop page. On L.L. Bean's site, the link was at the bottom of a long page, under the fold. The link label was more descriptive.

9. The link to the desktop site should be labeled *Desktop site* or *Full site*.

The name of the link tells users what to expect; the more precise the label, the better. A label such as “yoursite.com” for the full-site link (as in the Southwest example above, where the full-site link was labeled *Visit southwest.com*) may leave users wondering as to whether they are currently on the right page.

In one of our Hong Kong testing sessions, a participant was utterly confused by a similar link on the Virgin Atlantic website. He thought that the mobile page was not Virgin Atlantic’s and that he had accidentally navigated to the wrong site. He then proceeded to the full site and struggled with checking flight status, a task that would have been a lot easier had he used the mobile site. (Since then, Virgin Atlantic has changed the label for its full-site link.)

10. [Web] A responsive site should also have a link to a “full” site in its lower-screen version.

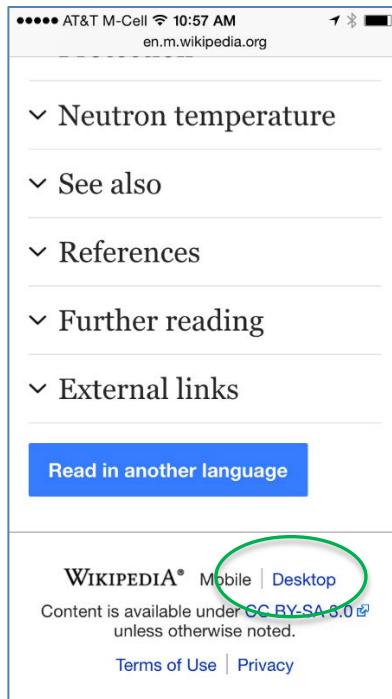
The distinction between full and mobile site makes sense for mobile-dedicated sites, but, to developers, it can seem unnecessary or even ridiculous for a responsive site. There are several reasons for which the link to a “full” site should exist on responsive sites:

- If the site looks different on mobile than on desktop, users will perceive it as a mobile-dedicated site. They are not aware (nor do they care) of the technology behind site. Those frequent-desktop users may still want the desktop site on mobile because they are familiar with the desktop layout.
- Many responsive sites do not have 100% content and feature parity (see the section *Responsive Design*). Users may still need to occasionally access functionality that was left out from the mobile version of a responsive site.

11. On deep pages of your mobile site, include a link to the corresponding page on the full site.

If the full version of a mobile page contains more information or information presented in a different layout (e.g., a larger table or spreadsheet), allow users to access the full version of that page on mobile by including a link at the bottom of the page. The link will enable users to directly navigate to the page of interest, without having to go back to the homepage, select the desktop-site link, and then struggle to find their way to the page of interest on the desktop site.

For example, all deep pages on the mobile version of Wikipedia had a link pointing to their desktop version.



Wikipedia.com allowed users to see the desktop version of any page.

MAKING YOUR APP FINDABLE

App stores have hundreds of thousands of apps. In an ocean of apps, what can you do to make your app easy to find and recognize? And, once users have installed your app, how can you ensure that they will easily find it on their phone? (Remember, people install many apps, and they often forget about most of them. Plus, apps can be buried in folders and their owners may never see the icons.)

12. [App] Choose app names that are unique, recognizable, and memorable.

First, make sure that your app has a recognizable, unique, but also memorable name. Names that are too common are not identifiable — if someone recommends your app to a friend (and many apps are discovered through word of mouth), you want that friend to find your app easily.

13. [App] If your brand name is recognizable, make it part of the app name.

If you have a brand that people know of, it's very likely that it will be the one most salient thing that users will associate with your app. They may not remember the exact name that you chose for the app, but they may remember whose apps it is. Especially if apps are ordered alphabetically (as on Android phones), people will look for your brand name and not for the name that your app may start with.

For instance, names such as *YP* (for Yellow Pages) and *Express* (for Google Shopping Express) don't mean anything to most users. Yet they have behind

them well-known brands that users are more likely to remember than the specific name that the company has decided to use for the app.

This recommendation seems to go against another established scannability guideline: start your name with a keyword that conveys the maximum amount of information. Thus, one could argue that *Shopping* or *Express* are better names than *Google Shopping Express*, for instance, because they tell users what the app is about. Wrong. In this situation *Google* is a stronger associate³⁰ than both *Shopping* and *Express*³¹. (How many shopping apps are there out there? A lot. And how many Google apps? A lot fewer.)



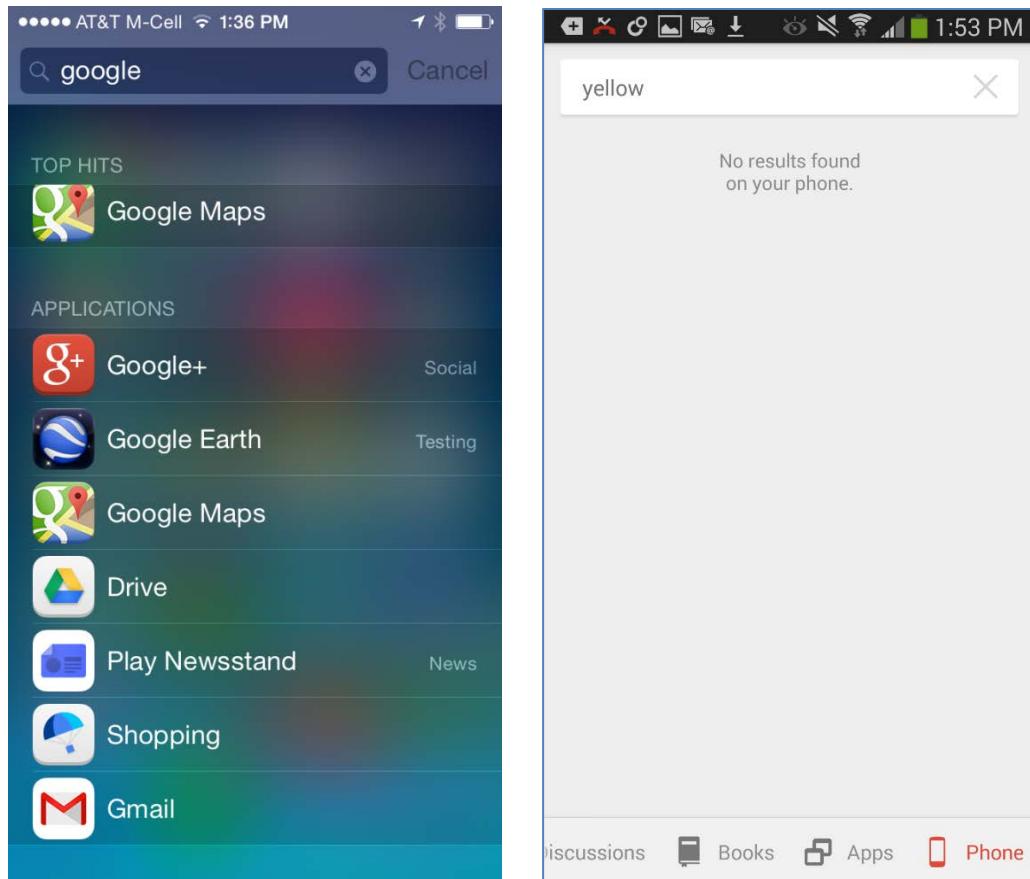
Various AT&T applications include the company name in the label.

14. **[App] If your brand name is not part of the app name, your app should be one of the search results in the app store and on the phone when people search for your brand name.**

Allow users to find your app if they search for your brand name on their phone or in an app store.

³⁰ More about associations in memory: Raluca Budiu. "Memory Recognition and Recall in User Interfaces." <http://www.nngroup.com/articles/recognition-and-recall/>

³¹ Google Shopping Express was initially named *Shopping* and then *Express*.



Searching for a brand name on the phone should return all the apps associated with that brand, whether their name contains the brand or not. Left: An iPhone search for “google” returned apps such as *Shopping* and *Drive*, which were made by Google but did not include that word in their names. Right: An Android search for “yellow” did not return the YP (Yellow Pages) app. (The app was installed on the respective phone.)

15. **[App] If your brand name is not part of the app name, consider incorporating your logo in the app icon.**

If the logo is part of the app icon, users will be more likely to recognize your app if they see its icon, even if the app name does not explicitly include your brand.

16. **[App] Choose an application icon that is descriptive and easy to recognize.**

Application icons that are too generic or too obscure don’t help users recognize your app any faster when they’re looking for it. The icon should also be indicative of the app’s functionality, so that users can “guess” what the app is about, even if they don’t remember using it.



The YP (Yellow Pages) icon was not memorable or recognizable: it hardly created any association with the brand or the type of service that it represents. In contrast, the Amazon icon included the Amazon logo and a shopping cart evocative of their e-commerce business.

17. [App] Use the same (or almost identical) icons for all versions of your app if you offer different apps for phone and tablets.

If a user normally uses your app on one device (say, a tablet), that user should be able to immediately identify the app on those rarer occasions it's used on another device (say, a phone).



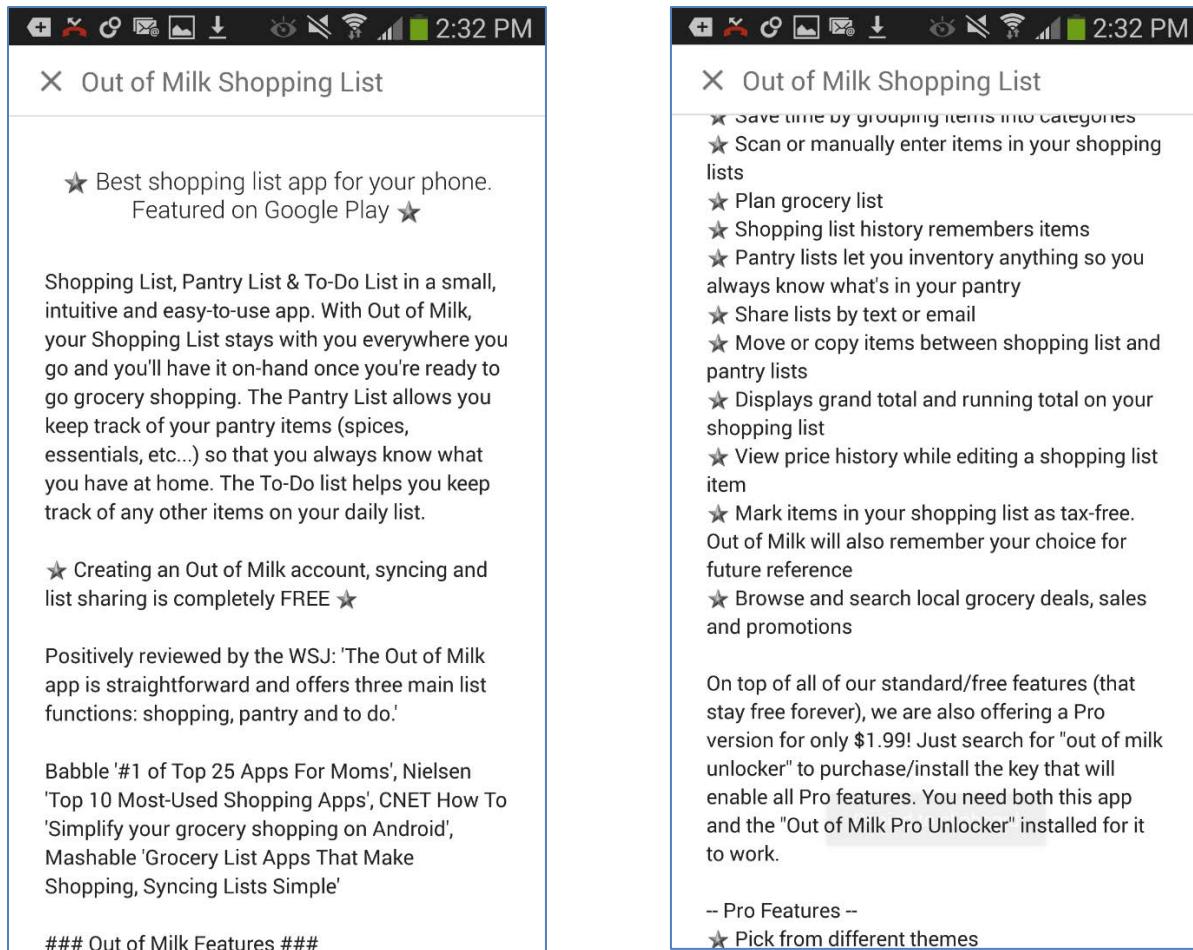
In these examples, the phone and tablet icons for The Wall Street Journal and Fidelity Investments have very different colors. The icon on one device will not be immediately recognized by someone mainly using the app on another device. The two icons for The Economist are so similar that one could only tell the difference when viewing the two next to each other. The icons for Full Fitness and KAYAK are the same, but the tablet version is denoted by a small "HD" label which will not impede immediate recognition.

- 18. [App] Your app description in the app store should clearly explain what the purpose of the app is and if it has any special features compared with other similar apps (and especially, compared with the free or paid version of the same app).**

- 19. [App] Your app description should be concise and scannable.**

Users are not going to sit in awe, reading carefully pages and pages of app descriptions. Limit the description to the key points and format it according to

the rules of writing for mobile (see section *Content*). If the app is special in any way, prioritize the description of that feature. If a new version includes updates that were requested by users, make sure to mention that in the description. If you have several versions of the same app, it's important to explain the differences so people know which to get.



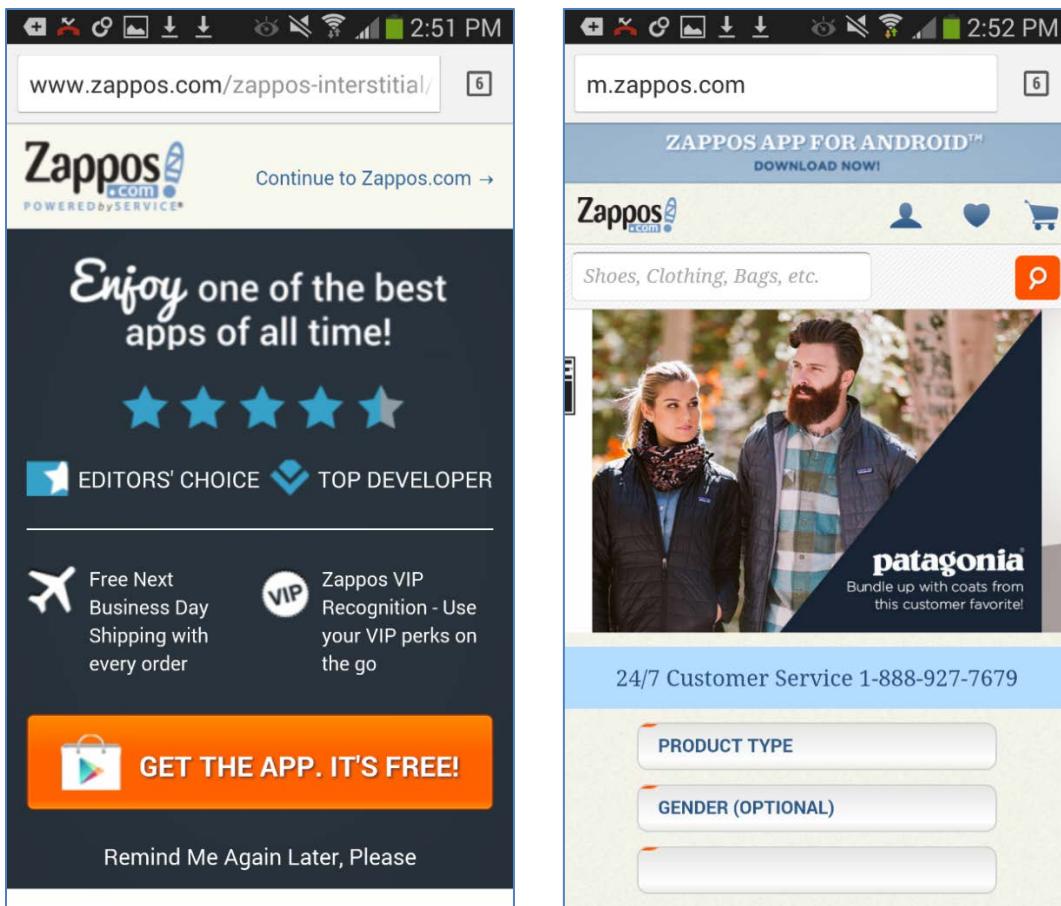
Out of Milk Shopping List app description in the Google Play store ran over seven 5.7-in screenfuls. The first paragraph stated the obvious ("With Out of Milk your Shopping List stays with you everywhere you go and you'll have it on hand once you're ready to go grocery shopping.") The description did explain the differences between the paid and the free versions of the app and did emphasize the awards received.

Advertising Your App on the Mobile Web

20. Advertise the app on your mobile website. Make sure that you promote the right platform on the right device.
21. Advertise the app as a link on your page rather than on a separate page.

If users come to your site and you have an app, let them know. But don't force them to choose between continuing to the mobile site and downloading an app. Users want the answer as soon as they can, and downloading the app demands a detour from their current goal and also an increase in interaction cost.

The best way to advertise an app is as a banner at the top of the mobile page that was requested by the user, and not as an interstitial page that interrupts the current task and requires an extra wait period (for the interstitial and also for the requested page).



Zappos.com stopped users in their track by showing an interstitial that advertised the app (left). Then it included another ad for the app on the homepage of the mobile site. The ad took users to the Play Store instead of recognizing that the app was already installed on the device.

Sometimes users may already have your app, but may still choose to go to the web to get more information from several sources at once. For instance, searching for a movie in a search engine will give pointers to both Wikipedia and IMDb; a movie buff who wants to see both sources may prefer the web for that particular task, rather than carrying the search twice in two separate apps.

And never cut access to the website and direct users to your apps, like Taco Bell did. It's no guarantee that users will bother to download your app; most likely you will succeed in annoying them and losing precious traffic.

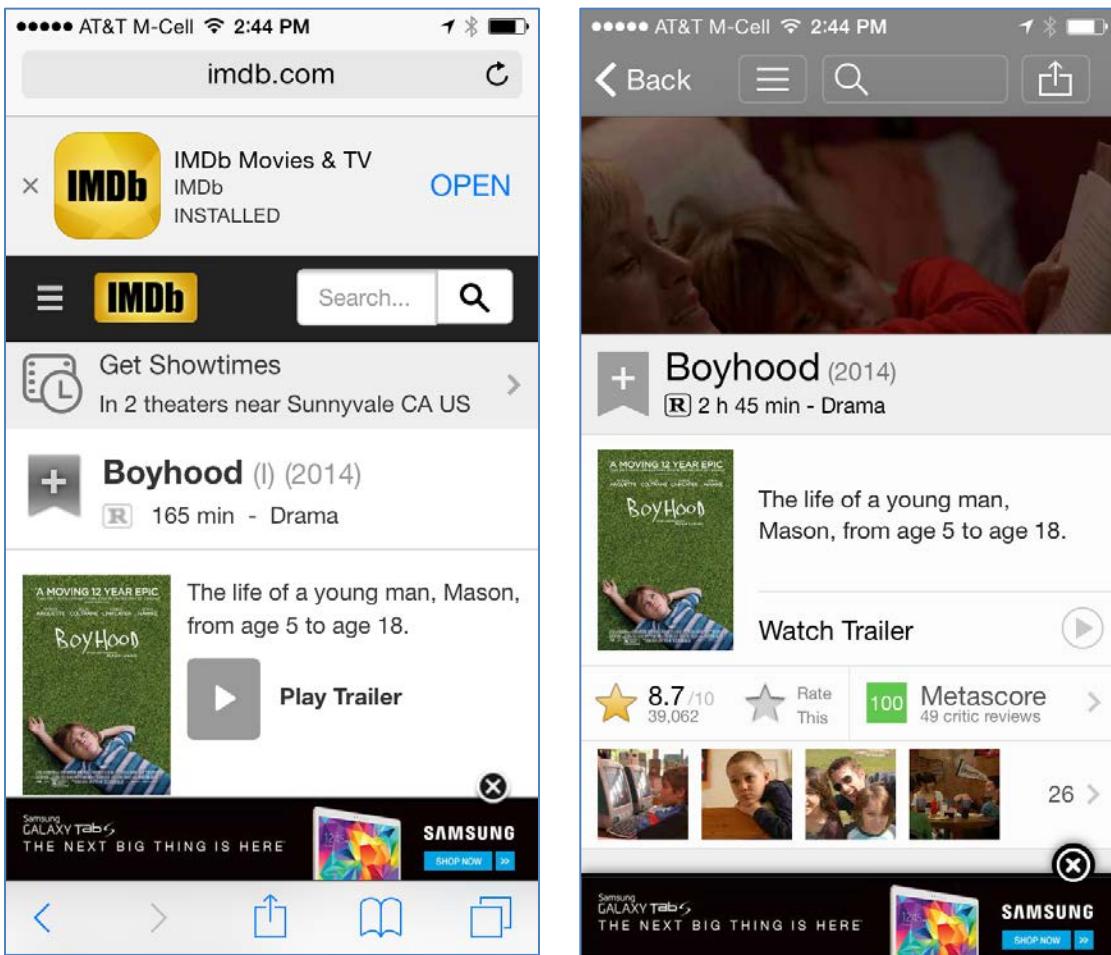


Taco Bell cut off access to their site on mobile and directed users to download its app in the app store. (Note that it did not bother detecting that the request had come from an Android phone and they included links to both Apple's App Store and Google Play.)

22. If users have the app installed on their phone, the link advertising the app should take them to the corresponding page in the app.

Sometimes users may have forgotten that they own an app or they may want to carry out their task across multiple sites (as in the movie-buff scenario mentioned above). In those situations it can be helpful to remind them in a nonintrusive way that they own the app and they can potentially improve their

experience by using it. To do so, help users transition seamlessly between the mobile site and the app by using the in-page app ads. These app ads (normally shown at the top of the page) should point the user to the corresponding page in the app (and not to the app homepage), as in the IMDb example below.



IMDb website (left) and corresponding iPhone app (right): tapping on the app ad at the top of the left screenshot correctly opened the same page in the installed iPhone IMDb app.

Realize however that users may choose to stay on the mobile web instead of using the corresponding mobile app. The interaction cost of switching to a different context can be high (wait for the app to launch, use the app, and then potentially go back to the web to continue the task), and in some situations it may not substantially improve the user experience or offer any other benefits.

MOBILE-APP FUNCTIONALITY

After watching a lot of people use apps on their phones, we can confidently say that the secret of a good app is keeping the functionality simple. We find that, whenever features are exposed and easy to access, users are successful. Trouble appears when the tasks involve some complex feature that's hidden under a chain of buttons.

When we ask people to show us how they use an app that they have on their phone, they usually mention 1–2 things. We sometimes ask them to carry out a task that is supported by the app, but that they don't normally do. Almost always they stumble and it takes them a while to figure out how to do it.

23. [App] Find a core function and design your app around it.

24. [App] Keep the app functionality simple.

25. [App] Don't add features that are unrelated to your core function.

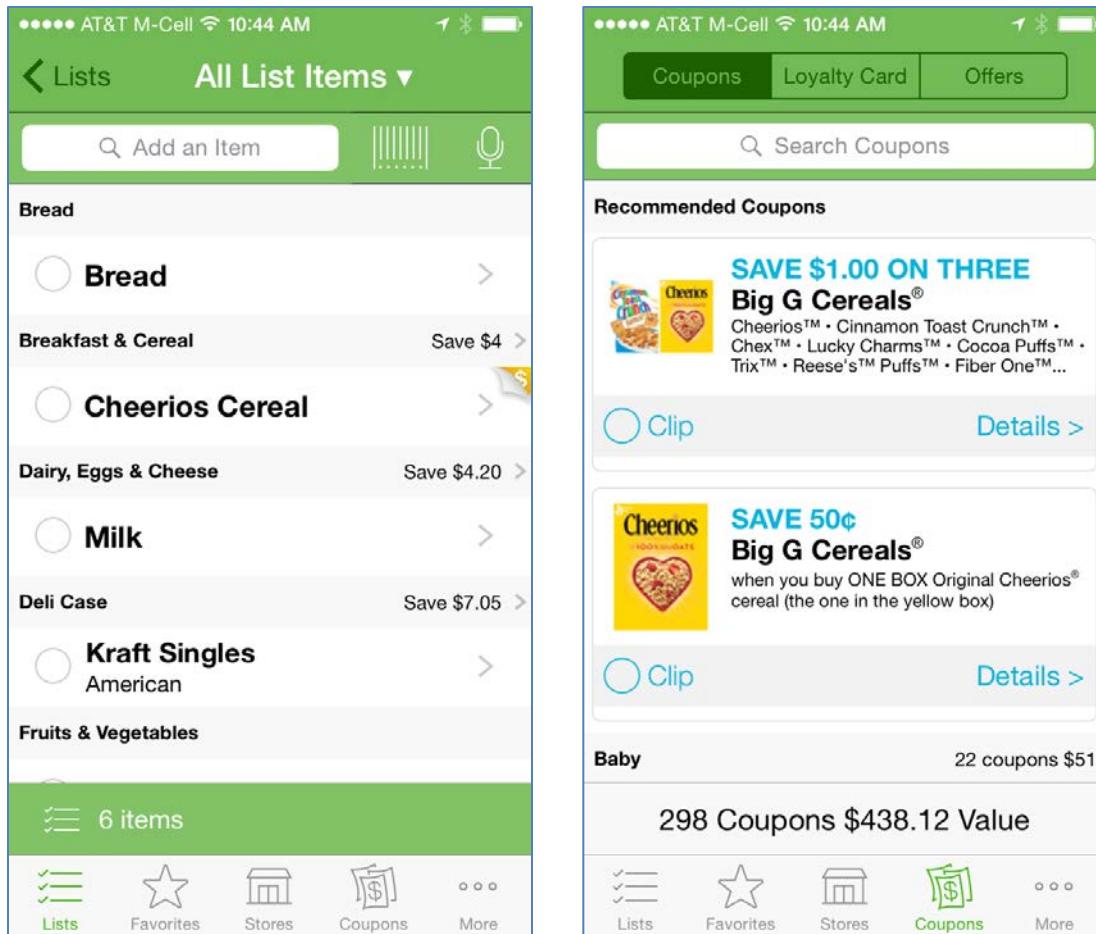
A 2009 study of information needs on the go showed that even then people were able to address 94% of their questions using their mobile devices (see section *Mobile Information Needs*). Does it mean that in the near future we'll be ready to get rid of our computers and replace them with smartphones? No. In fact, most of the tasks that users perform on phones are a lot simpler than the ones that people typically do on desktops. The screen size is too small to support complex information exchange or processing³². And most mobile apps support only a very small number of tasks that are fairly simple in nature. And that's how it should be.

However, the temptation of bigger and better things arises every so often. We see apps that attempt to fit any possible (sometimes remote) feature in their app. If a single user requested that feature in a review, developers feel compelled to oblige — with ten users asking for ten different things, they have the chance to bloat their app to the point of becoming unusable.

Grocery IQ is an example of app that strayed from the basic tenet of designing around a single functionality. The core function of the app was to create shopping lists. However, on top of the shopping-list task, the app supported the task of clipping and selecting coupons. The coupons got added to the shopping list. It's almost as if there were two different apps within the same app: one for coupons and one for grocery lists.

The result was that the shopping list appeared cluttered with nonessential items that came from the clipped coupons. Instead of being able to quickly figure out if they needed to go to the store or not, users saw a long list of coupons added to the list.

³² For more on how screen size affects information exchange between people and devices, see Raluca Budiu. "Scaling User Interfaces: An Information-Processing Approach to Multi-Device Design." <http://www.nngroup.com/articles/scaling-user-interfaces/>



Grocery IQ app, a shopping-list app, had a coupon section where users could add coupons to their shopping list. The final shopping list got cluttered with unnecessary content. And the coupon and the shopping list features were not tightly integrated.

The Weather Channel app has suffered many iterations over time, both on Android and on iPhone. An older version did a lot more than simply report the weather forecast: it allowed users to see tweets related to the weather or post their weather on a variety of channels, it showed dew point, visibility, as well as sunrise and sunset information, and also weather alerts, videos of the weather, photos shared by others in the area, allergy indicators, info about their TV channel, and a radar map of the weather. In a recent version of the app all those features were still available, but at least the interface prioritized the most important piece of information: the current weather.



Two versions of The Weather Channel app for iPhone: older version (left) and more recent version (right). The older version had had an abundance of features unlikely to be needed and that cluttered the interface. The current version still had these features (hidden below the fold), but appropriately prioritized the most important information. (The newer version wasted the screen space however; there is too little information available on the screen.)



The iPhone Weather app did a better job than both versions of The Weather Channel: it focused on only the relevant weather information (temperature and forecast, plus a few other metrics below the fold) and it appropriately used the screen space to deliver valuable content.

In contrast, the iPhone Weather app showed only the current temperature and the forecast for the next few hours and days. More weather-related metrics such as humidity and visibility were hidden below the fold, and no social features or videos were included.

Your app should be as simple as possible. Hidden features are hard to reach, and they increase visual clutter. The more users have to choose, the more they have to think about it and the longer it takes.

Multiple Apps or One App?

We often get asked: if my company supports multiple kinds of tasks, should I build a single app for everything or should I divide the functionality across multiple apps, each with a narrow focus? Most of the time the answer is to split the app into several apps, especially if the tasks are disparate. There are however advantages and disadvantages of having multiple apps.

- **Focus on one task at the time.** Having multiple apps allows companies to streamline the interface and create a better, simpler user experience.
- **Focus on one audience at a time.** If your company serves multiple, discrete audiences (e.g., faculty, students, alumni for a university), then creating multiple apps allows designers to give deserved focus to each of these audiences

individually. It also allows them to include only those tasks relevant for a particular audience.

- **App interference.** On the minus side, multiple apps make it difficult for users to know (1) what the right app is for them; and (2) when to use what app, if they had installed multiple apps on their phone. Companies need to work harder to individualize the apps that share the same brand name.
- **Context switching.** If two tasks are normally performed at the same time, separating them in different apps can increase the interaction cost and can disorient users as they'd need to move from one app to the other.

26. [App] Use separate apps if the tasks that you plan to support are different and performed at different times.

For instance, Amazon has multiple apps that serve different main tasks; among them are:

- Amazon: the e-commerce app
- Kindle: the book reader
- Flow: an augmented-reality app that lets you identify objects by photographing them
- Amazon Local: deals and offers (similar to Groupon)
- Amazon Seller for managing your sales on Amazon.com

Each of these apps has a distinct audience, and some of these audiences overlap. Having all of the tasks embedded in a single app would create a too complicated and cumbersome interface. Note also that most of these tasks are separate enough that it's unlikely that the user would need to perform two at the same time. For instance, it's safe to assume that most Amazon shoppers do not normally sell products on the Amazon Marketplace.

In contrast, eBay users must use the same app for both selling and buying. Keeping them within the same app may make sense if most eBay users do both these activities at the same time, in a single eBay session. (For instance, they may check all auctions in which they are involved in both capacities at the same time). If however the tasks were mostly performed at different time or by different users, having two separate apps would keep each of the tasks more efficient.

27. [App] Use separate apps if the audiences that you plan to support are distinct (that is, most people will be in a single audience category).

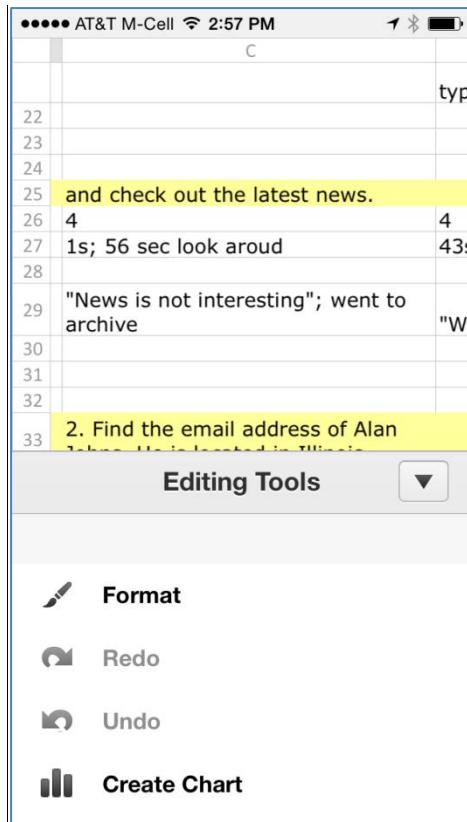
Similarly, if your audiences are discrete and they do not overlap, it is better to create a separate app for each audience. For instance, a health-insurance company may have different apps for physicians and patients.

From Desktop to Phone

What if you already have a desktop app and you want to move it to mobile? Should you provide the same functionality on the phone? The answer depends on the complexity of the app. If the app is complex, it's likely that it cannot be accommodated well on the small screen and that you will need to support only a subset of features on mobile. If the app is simple (for instance, a time-tracking app), it may be however easy to fully move it to mobile.

28. [App] For the phone version of your desktop app, prioritize those features that revolve primarily around consuming rather than producing content.

Think hard at the types of functions supported by your app that are likely to be needed on the go. If your app is a productivity app that requires a lot of input from the user, chances are that the mobile users won't need the same plethora of editing functions. Look for instance at Office Mobile for iPhone. From all the features that Microsoft Excel implements on the desktop, they picked a very limited subset that essentially allows users to do some basic formatting and, most importantly, lets them view Excel documents on their phone.



Microsoft Office for iPhone: Only a few features from the desktop were supported by the phone version of Excel.

Similarly, PS Express, an iPhone version of Photoshop, an application notoriously difficult on the desktop, had many fans among our participants. The app supported only a few basic photo-editing features that enabled users to quickly correct the pictures taken by their phone camera.

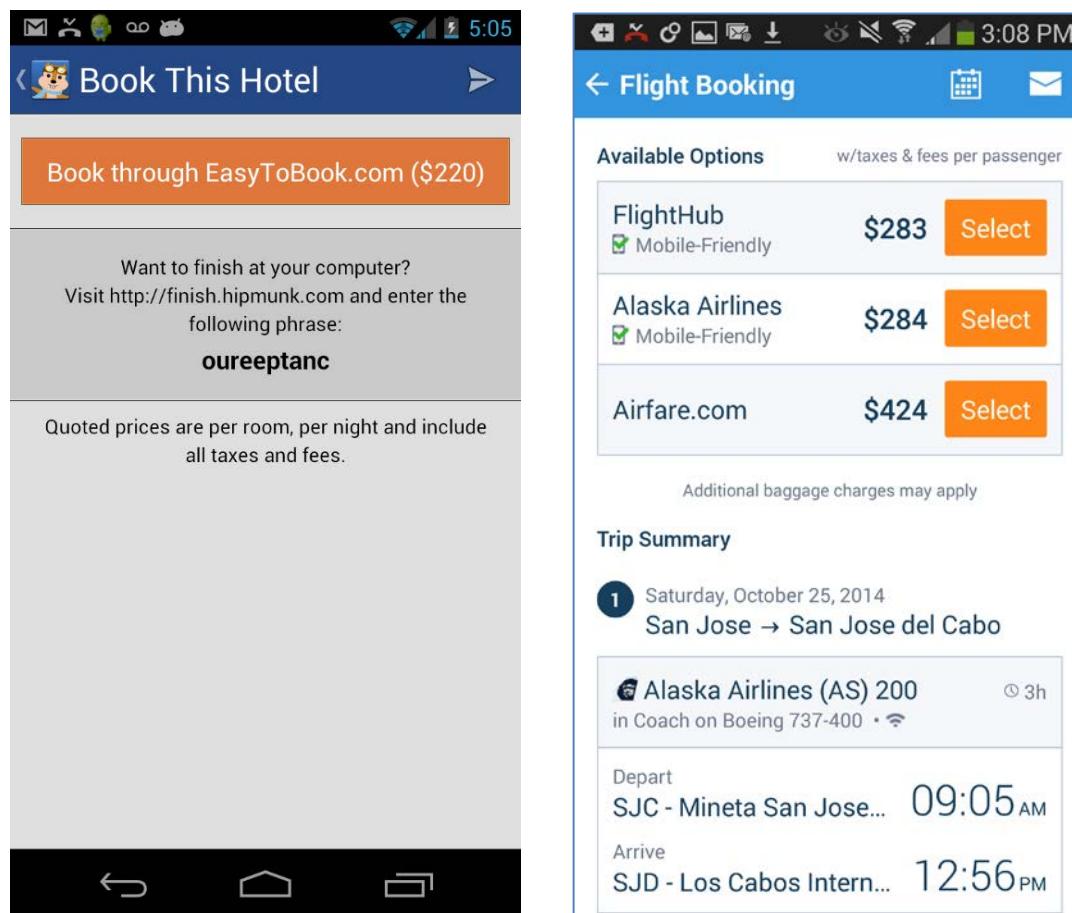
29. Allow users to resume their task on a different platform.

If you have multiple platforms (e.g., a website or a desktop app and a phone app), users' data should be consistent across platforms. Users shouldn't have to worry about syncing their data — they should have access to them wherever they are.

Mint, for instance, allows users to see the transactions entered on their phone when they use their regular computer. Similarly, Epicurious, a cooking app, allows users to sync their recipe box (on the phone, on their tablet, and on the website) if they log in to their account.

30. Consider emailing a link to users to let them finish on a different platform a task started on mobile.

Hipmunk, the travel-search engine, used to have a particularly innovative approach to syncing: the mobile app gave users a code word that could have been used on the desktop site to resume a task started in the app. A more recent version of the app still had this feature, but unfortunately buried it under the share menus in the Android app.



Hipmunk for Android: An older version (left) included a clear code word for finishing up the hotel booking on a different platform. A newer version (right) made this feature a lot less salient. (If users decided to email this itinerary, a link in the email would allow them to resume the task.)

LOGO AND COMPANY NAME

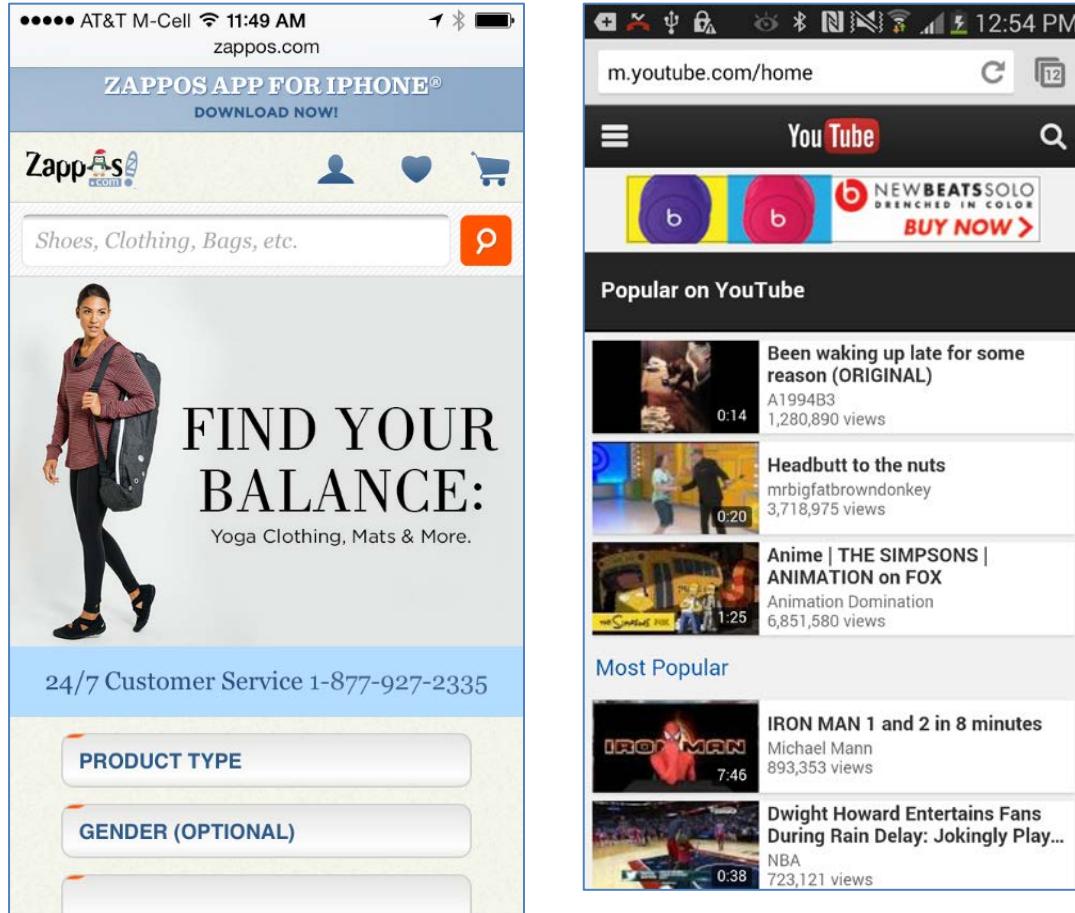
The company logo (or name) serves an important function: it tells users where they are. It shouldn't take too much space on the small mobile screen, but it should be present in order to help users get oriented.

- 31. [Web] On websites, include the company logo or name in a salient location at the top of the mobile homepage.**
- 32. [Web] On websites, include the logo or name on every deep page of your site.**
- 33. [Web] The logo should take a minimum amount of space on the small mobile screen.**

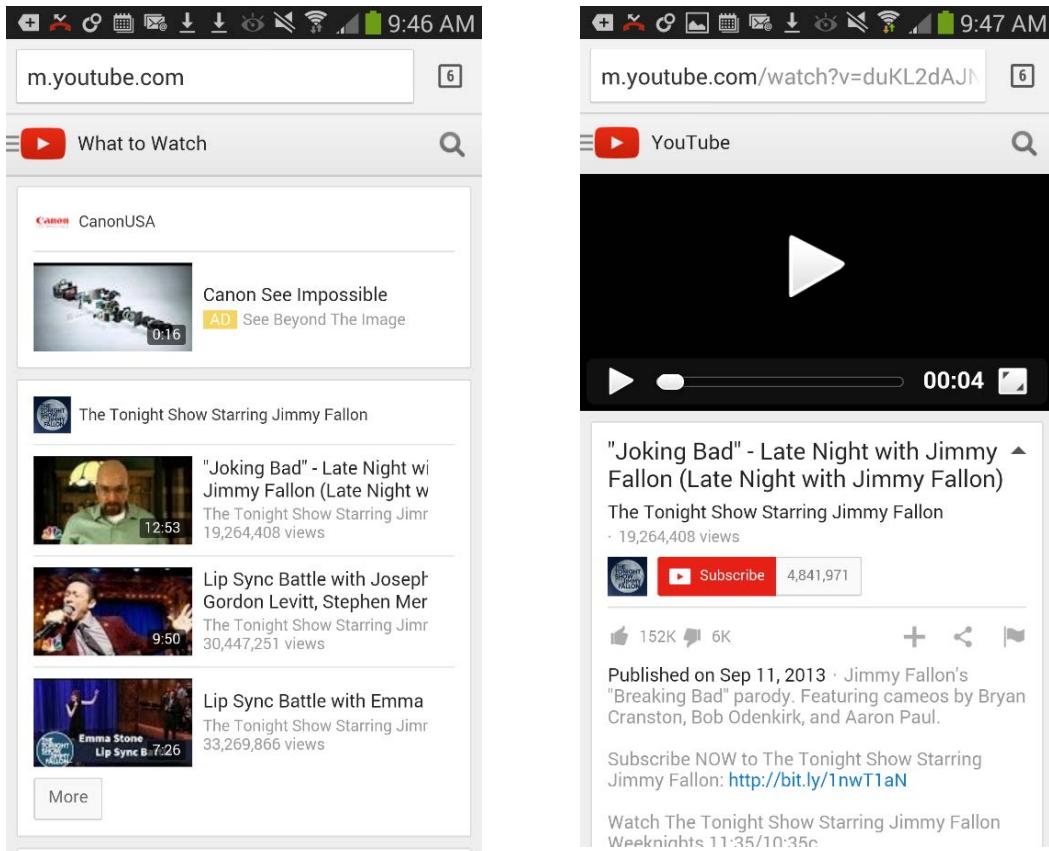
The mobile screen is small, so it may seem like a waste to include a logo on each page. Wrong! Unfortunately, on a small screen people often get disoriented and confused; they may ignore or not see the context in which the information is presented. Many times they reach the webpage deep inside a site via a search engine and have no idea where they are. The URL bar and the title bar may not be displayed in some mobile browsers, so unless you tell them explicitly where they are, it may be hard for users to find out.

Still, do make the logo fairly small, as long as you keep it recognizable. Big headers are only for the desktop web.

The screenshots below show how the logo can be incorporated on the page, without taking too much space.



The logo for the mobile sites of Zappos (left) and an older version of YouTube (right) appeared on the homepage, without taking up too much space.



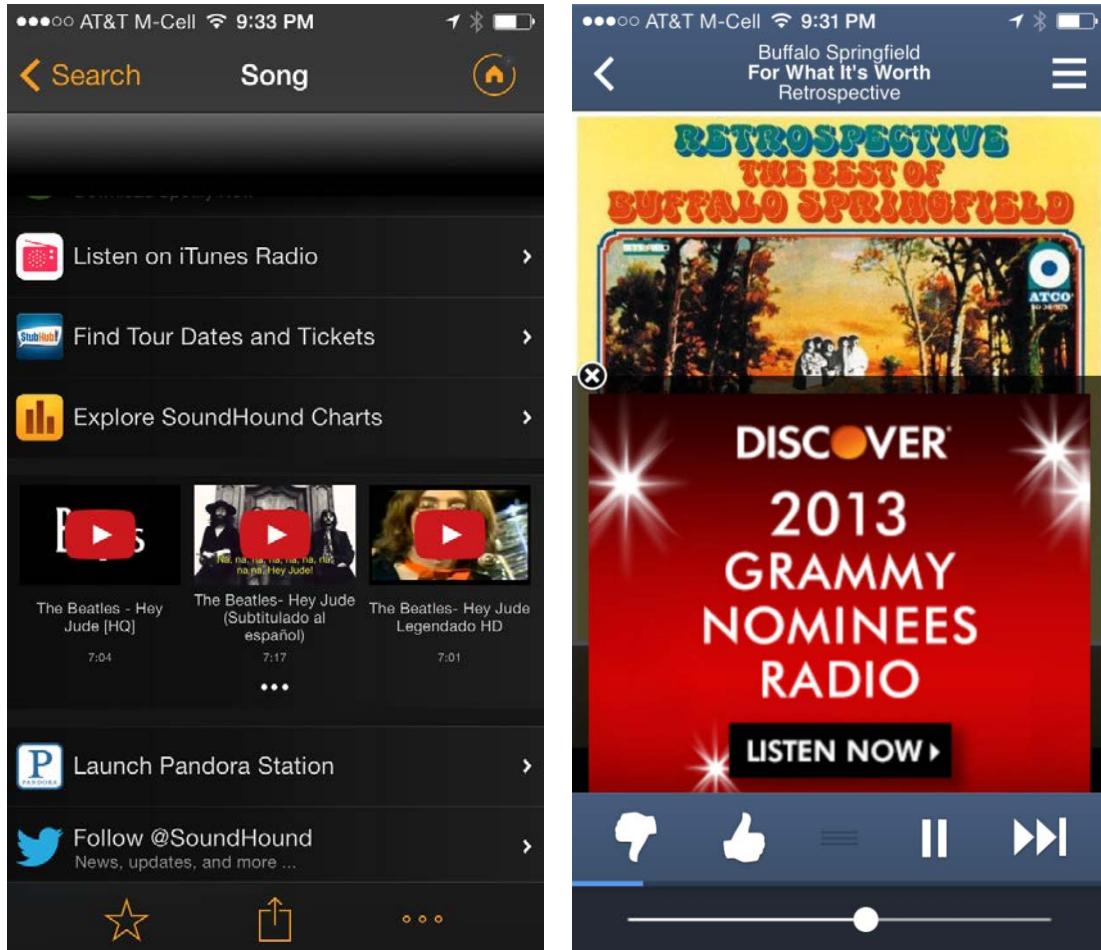
A recent version of YouTube's mobile site did not include the site name on the homepage, but it did include it on deep pages.

- 34. [App] An app generally does not need a logo on the homepage or elsewhere.**
- 35. [App] If you estimate that other apps will give control to your app, consider including the logo on every page of your app.**

Note that for most apps it's still not so important to have a logo. With apps, users are much more in control: they decide which app to use by launching it from their home screen. So, most often, they already know what the app is. It's also still rare that the user be taken to a random page in your app while using a completely different mobile app, as might be the case when using the web.

However, as more and more apps start to communicate with each other, it's likely that in the near future we will see users jump more often from one app to another. If you expect that the control be given to your app from within another app (e.g., because that other app invokes your app), then the logo (or company name) will help orient the user.

For instance, SoundHound for iPhone, an app that recognizes music and plays it, can launch several other music apps such as Pandora, Rdio, Spotify, and iTunes Radio. Luckily, the links that launch these apps are quite explicit, but even with these links, users may get occasionally disoriented if they don't clearly see in which apps they are at a given moment.



A sound found in the SoundHound iPhone app (left) could be played in a different app, for instance in Pandora (right). The second app was launched from SoundHound. The lack of logo in the second app could occasionally confuse users, as some may have not realized that they were taken to a different app.

36. The company logo or name should link to the homepage.

This guideline is valid on mobile and on the desktop. Although not every user knows to navigate back home by clicking on the logo, this web convention has been around for a long time, and many users are familiar with it by now.

TOUCH TARGETS

Size and Placement

One of the bigger problems with touch interfaces is that touch targets are hard to hit precisely. Parhi, Karlson & Bederson (2006)³³ showed that, for touchscreen portable devices operated with one hand, the optimal target size was a 9.6 mm square (approximately 1cm²) for discrete tasks (that is, tasks that involved an isolated tap, as opposed to a series of taps; a series of taps is more representative of typing).

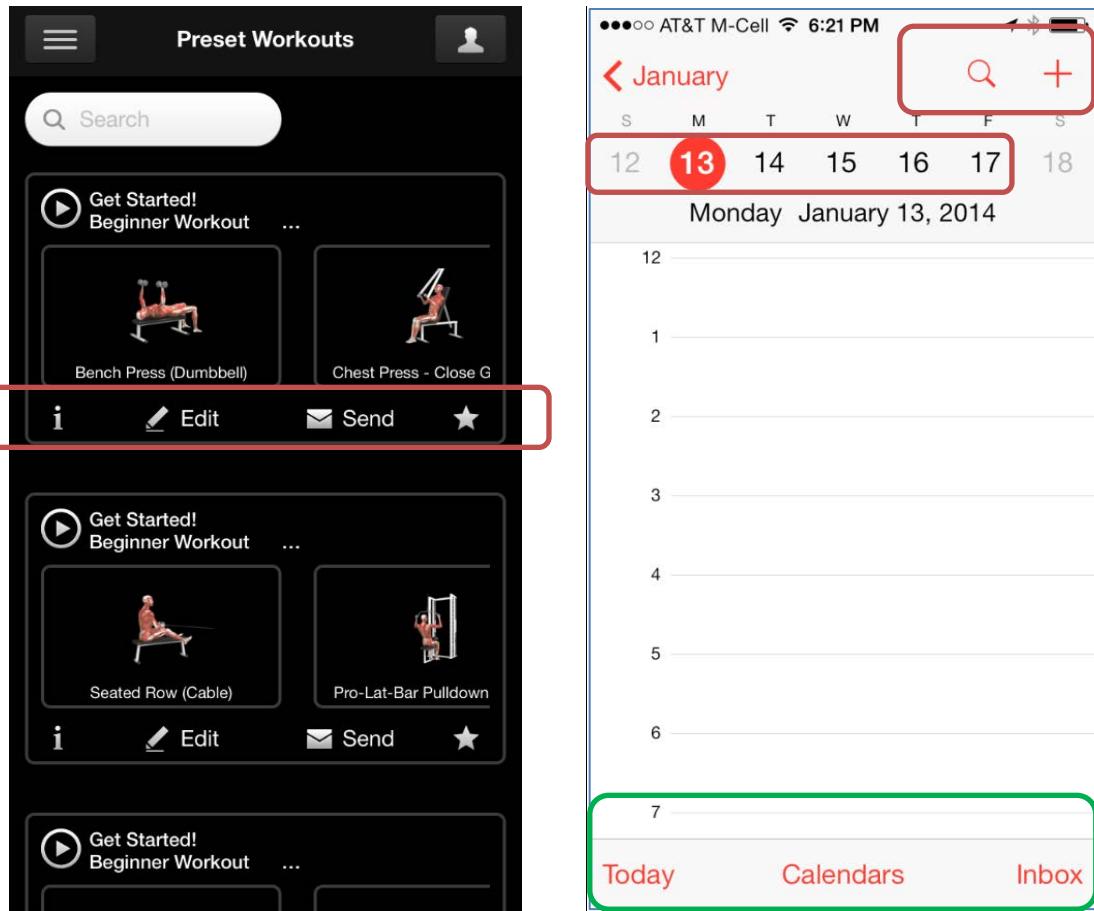
Therefore, all these interfaces must build in some tolerance to error by (1) leaving generous amounts of space around UI elements that need to be tapped (buttons, arrows for drop-down boxes, links, scrollbars), and by (2) increasing the target size of these elements. The first condition ensures that people will not accidentally click on the wrong element. The second condition builds in some room for reaching errors.

These considerations are contained in the following guidelines:

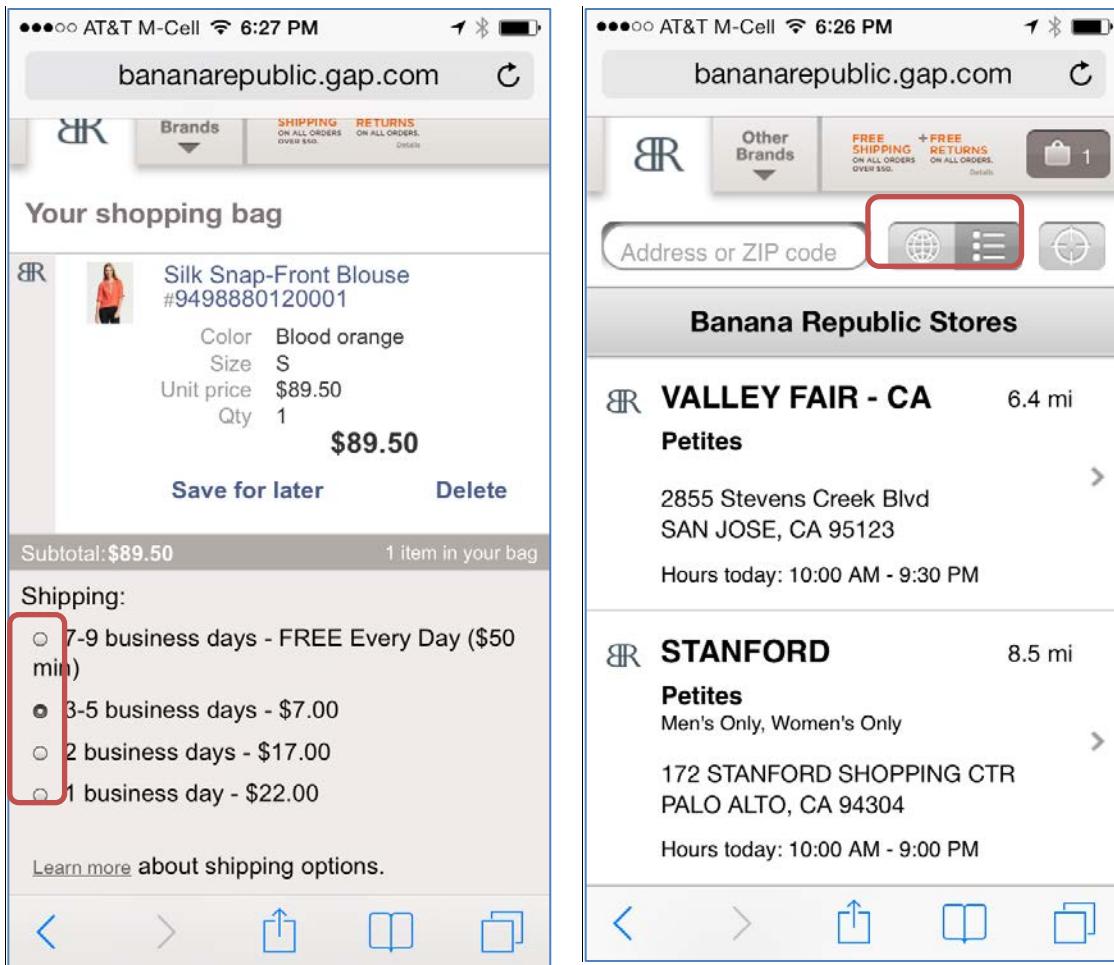
- 37. The target area (i.e., tappable area) for touchscreen devices should be at least 1cm X 1cm.**
- 38. Do not crowd targets. Leave generous amounts of space around UI elements such as radio buttons, arrows for drop-down boxes, checkboxes, scrollbars, and links.**

Although intuitively everyone agrees that small targets are hard to hit, many designers persist in creating smaller and smaller targets, in the attempt to fit more functionality on the small screen. That leads to apps with crowded, hard-to-use interfaces. Below are a few examples of targets that have just too little space around them.

³³ Parhi, P., Karlson, A. K., and Bederson, B. B. 2006. "Target size study for one-handed thumb use on small touchscreen devices." In *Proceedings of the 8th Conference on Human-Computer interaction with Mobile Devices and Services* (Helsinki, Finland, September 12–15, 2006). MobileHCI '06, vol. 159. ACM, New York, NY, 203-210. DOI= <http://doi.acm.org/10.1145/1152215.1152260>



iMuscle2 for iPhone (left): The icons under each workout were too small and hard to touch. Calendar for iPhone (right): The dates were too small. The elements at the top right corner were bigger, but still too small and close to each other. Contrast that with the buttons at the bottom of the page, which were much easier to reach.



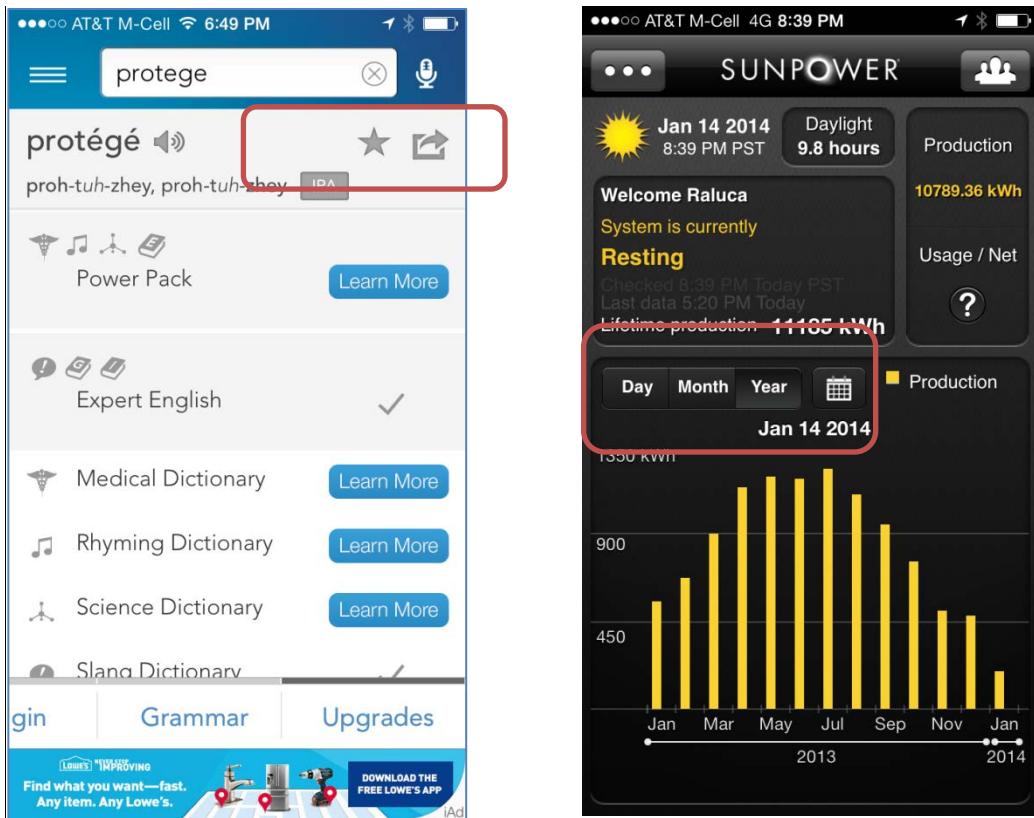
The mobile site of Banana Republic (m.bananarepublic.com): (Left) The shipping radio buttons were very small (and also close to each other). The buttons at the bottom of the page were of a better size. (Right) The map and list buttons next to the *Address or ZIP code* box were too small and close to each other.

One of our usability-testing participants struggled for about 2 minutes to select his state (California) in a drop-down menu. The states were just too close together and he kept hitting the wrong one, so he exclaimed:

"At this point I wouldn't even care [to find locations]... Sometimes with the keypad it's actually more handy..."

Indeed, although typing is generally painful, typing two letters to select a state can be easier than using a badly designed drop-down menu in a touch interface. Even for nontouch interfaces, if there are many choices in a drop-down box, the cost of scrolling down to the right option needs to be weighed against the cost of actually typing the choice or its first few letters.

When targets are too close to each other, it's easy to accidentally hit the wrong one. In the Dictionary and SunPower examples below, the targets were too small and simply too close for adult fingers.



Crowded targets are hard to touch reliably. Dictionary app for iPhone (left): The top icons for favorite and share were small and close to each other. SunPower app for iPhone (right): The *Day*, *Month*, and *Year* buttons (as well as the calendar icon) were hard to hit individually.

39. Do not rely on small padded targets.

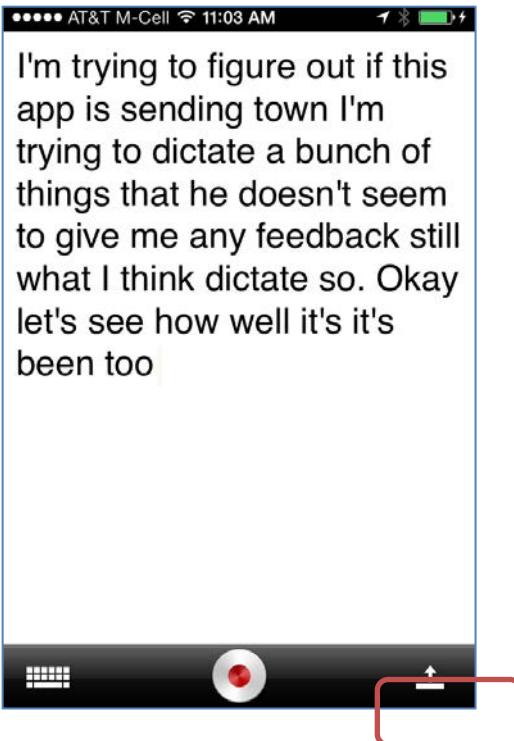
Padding targets is a practice that many apps use to get away with smaller targets. Padding means that the real tappable size of the target is bigger than the visible area of the target. Thus, even if users are slightly imprecise and don't hit exactly on target, the tap is still counted as hitting that target.

Padding helps, but it's not as effective as it might seem. The main problem is that, with few exceptions that we discuss below, users have no way of knowing which targets are padded. Because of that, they will still strive to reach the target precisely, taking more time to do so than if the target was bigger.

In human-computer interaction, Fitts's law says that the time to reach a target depends on the finger's distance to the target, but also on the (perceived) size of the target. Users' movement is typically composed of two movements: a rough one in the direction of the target, and, once the finger has come close to the target, a slow, refined one, focused on hitting the target accurately. If the target looks small, users will still take a long time trying to hit it precisely, without knowing that in fact they have room for error.

In other words, padding is not an excuse for making targets smaller: users are still going to struggle trying to hit them.

The share icon in the Dragon Dictation screenshot was padded: the target was small, but the app left some room for error. Unfortunately, users had no way of knowing it: they programmed their movements carefully and slowed down to reach the target.

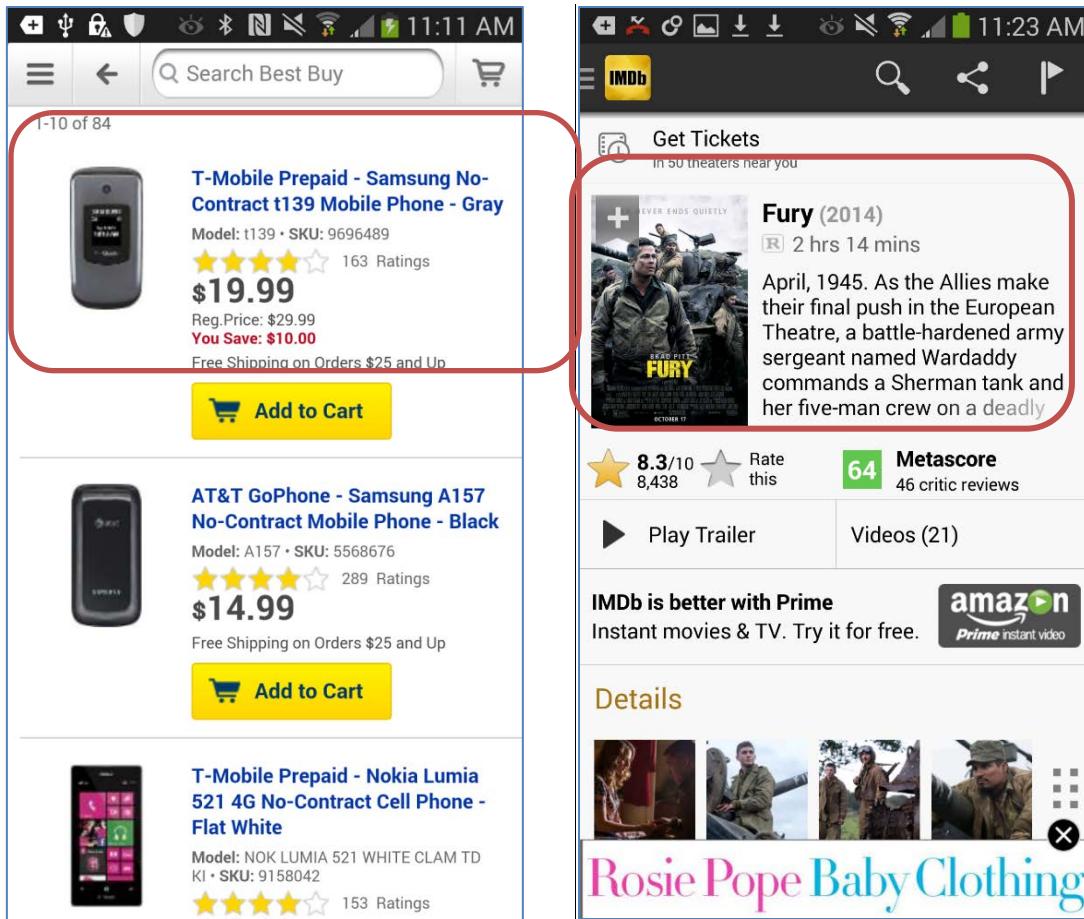


Dragon Dictation app for iPhone: The share icon was padded, but users had no way of knowing that they did not have to slow down to hit that target precisely.

40. Use padding for tabular views.

There are some situations where padding is expected. They usually occur in a tabular view. When users see items in a table, they assume that they can hit anywhere in a row and get the same result (rather than hit only on the right or only on the left side of the row).

Problems occur when apps and websites compartmentalize the row space, and assign different functionalities to different parts. For instance, in the IMDb example below, tapping on the movie thumbnail has a different outcome than tapping on the text in the same row.



Best Buy for Android (left): Users needed to tap on the image or on the blue link to navigate to the product page. It would have been better if tapping in the whole region marked on the screenshot had taken users to the product page. IMDb for Android (right): Tapping on the image led to a bigger picture of the movie poster; tapping on the text led to a summary of the movie.

Target Affordances

Affordances³⁴ in design refer to those physical features of an object that invite certain actions and reveal functionality. Thus, people grab a kettle by the handle because it has one; and they may turn a knob and push a button (but not rotate a button and push a knob).

³⁴ In recent years, Don Norman, who originally introduced the term “affordance” in UX design, has argued for replacing it with “signifier”. While his motivations are well justified, we stick to the word “affordance” since it is more familiar to the majority of our audience.

Similarly, in touch interfaces, people will touch a target if it invites touching. The traditional signals of touching are skeuomorphic³⁵ (see also section *Skeuomorphism*): targets that look like real-life buttons are normally touched because this is what we normally do in real life with buttons. Thus, 3D cues can easily convey tappability.

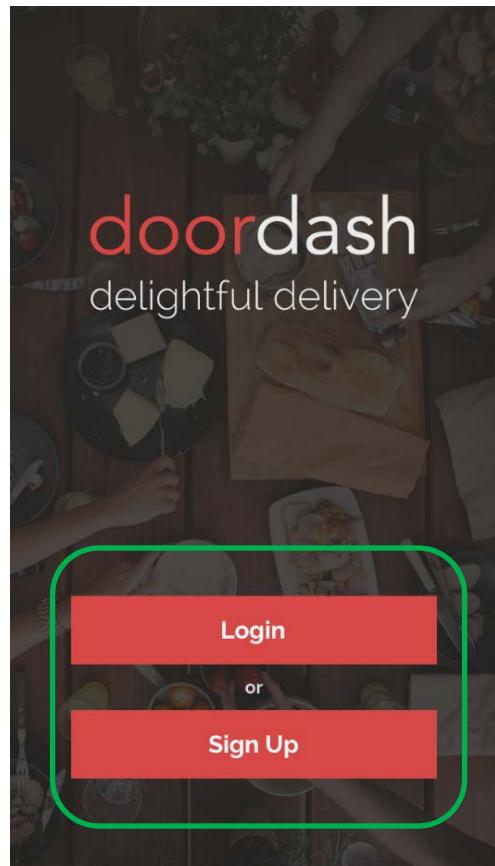
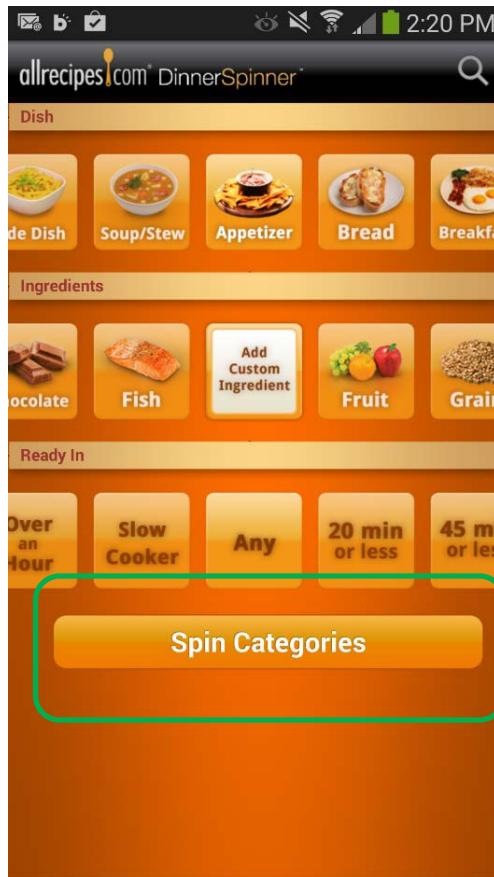
The current trend towards flat design makes signaling targets more difficult. Yet, borders, boxes for calls to action, and consistency within a site or app (as well as consistency with existing conventions) can go a long way.

Sometimes people can figure out targets right away, even if they don't have a 3-D look. For example, simply by placing targets where they are supposed to be, you can build in affordance. Thus, almost every user will know that the icons in a navigation bar are tappable because they have seen navigation bars before. They will know that a cell in a table view (see the discussion under guideline 40) is tappable because they are familiar with table views. But whenever your design departs from conventions (as it is the case with the examples below), make sure that all the targets have the right visual affordance.

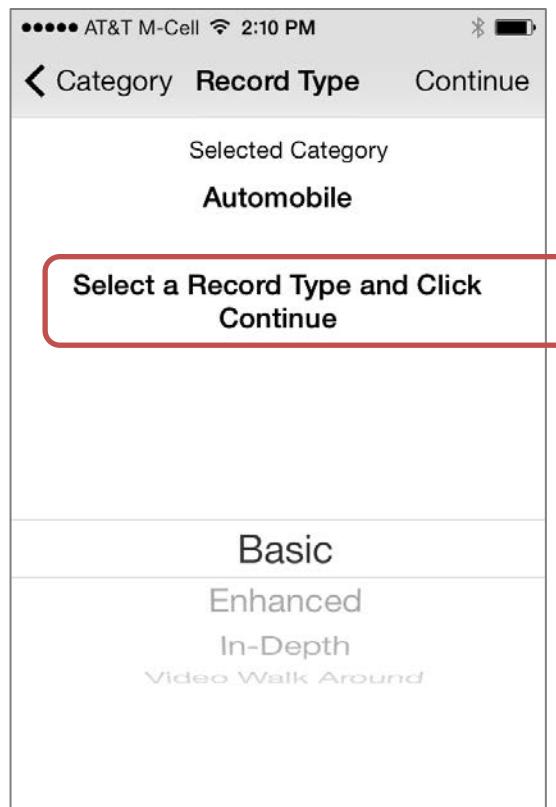
- 41. You can use 3D cues to make targets look tappable.**
- 42. You can use borders and color to make targets look tappable.**
- 43. To signal tappability follow target-placement conventions on your platform.**
- 44. Be consistent in your treatment of targets within your app or website.**

Here are a few examples of how these four recommendations can help signal tappability.

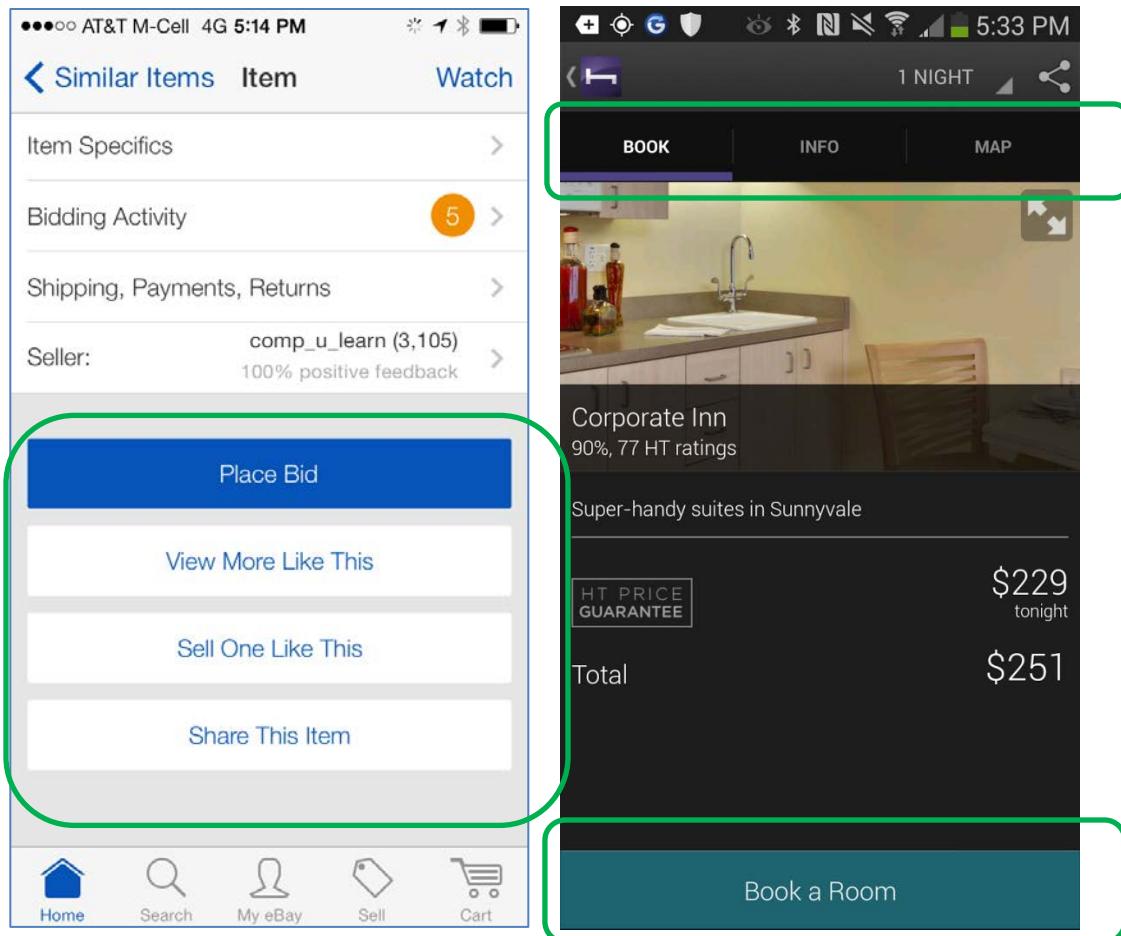
³⁵ Skeuomorphism in UX design refers to mimicking real-life objects in user interfaces. For instance, skeuomorphic design for a telephone pad may imitate a rotary dial.



Good target affordances: All Recipes for Android (left) used 3D cues, as well as color and borders around targets to indicate tappability. DoorDash for iPhone (right) used borders and colored background around the two main buttons (*Login* and *Sign Up*) to indicate tappability without using 3D cues.



Targets lacking affordance: In Weather for iPhone (left), the temperature number was a target; tapping it showed additional weather information (humidity, chance of rain, etc.). In Mobile Inspect for iPhone (right) some of the text in the top part of the page was tappable — but it did not look so.



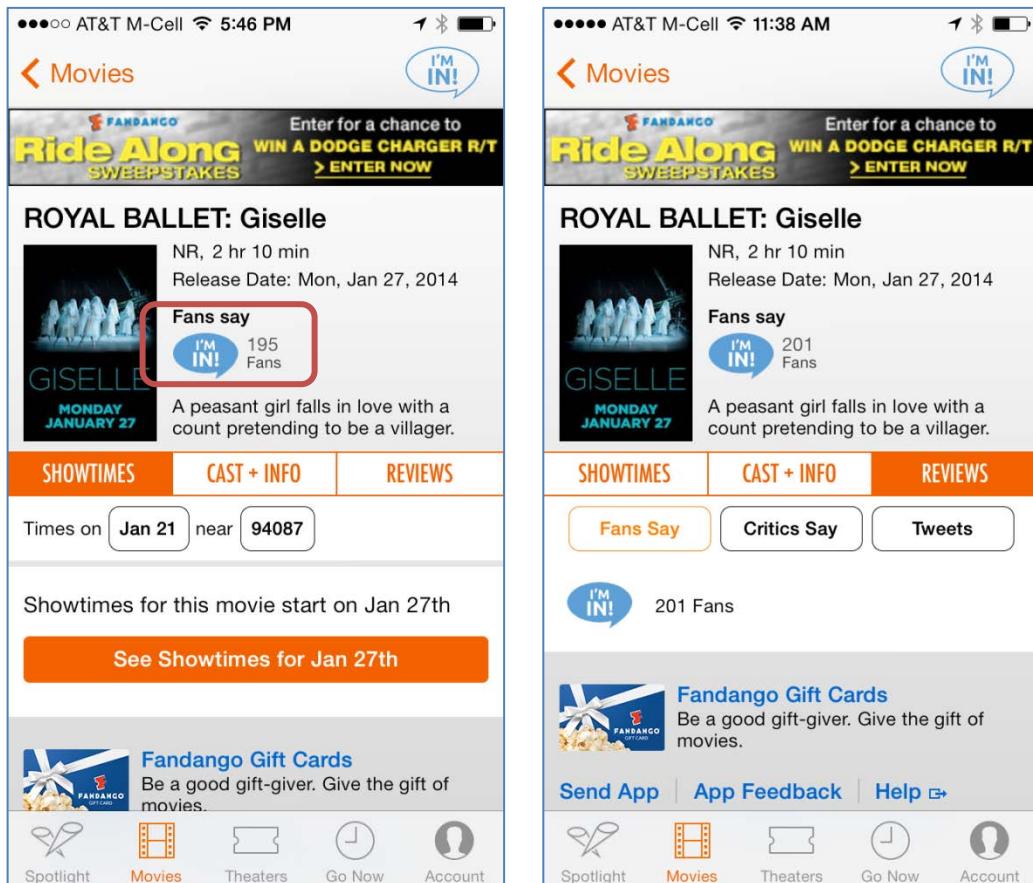
Color and position-based cues: In eBay for iPhone (left), the main calls to action were enclosed in boxes and placed at the bottom of the auction page. In Hotel Tonight for Android (right), the Book a Room button was enclosed in a blue box. The links Book, Info, and Map (at the top of the page) didn't have a different background, but the position, as well as the vertical delimiters between them, made them recognizable as tabs.

45. Use different visual styling for tappable and nontappable design elements.

When a design element looks like a button, users will think that it is a button. So make sure that you don't use 3D or other tappability cues inappropriately in your app.

In one of our studies we asked users to create a list of movies that they wanted to watch using the Fandango app. One user proceeded to press the blue *I'm in* sign next to the movie and completely ignored the colorless *I'm in* button in the navigation bar (top right corner). She thought that the movie had been added to her list of movies (in spite of the lack of feedback) and repeated the process for several soon-to-be-released movies. When, at the end, we asked her to look at her list of saved movies, she discovered that it was empty.

After our study, Fandango changed the behavior of the *I'm in* icon to lead to the reviews page for the movie. Although it's not what the user in our study expected, the new design represented an improvement because (1) the icon looked and was tappable; (2) upon tapping, there was immediate feedback (a new page loaded), which helped the user figure out the functionality of that button.



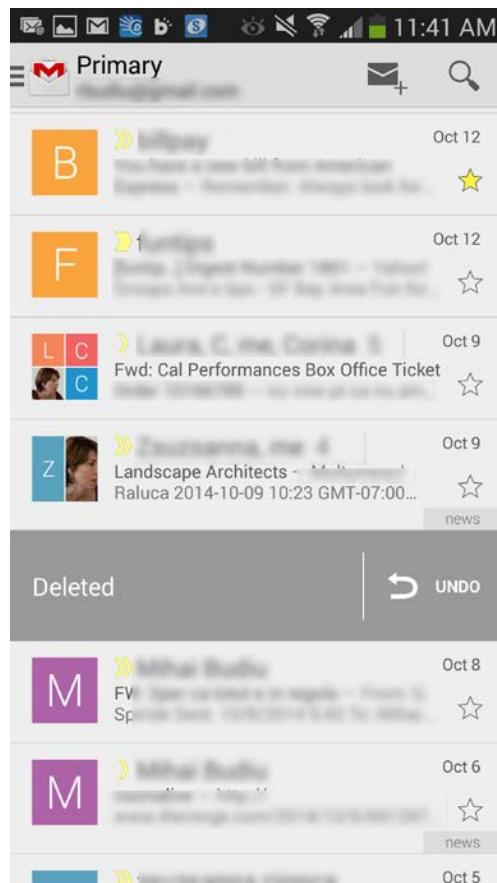
Two versions of Fandango for iPhone: (Left) Pressing the *I'm in* icon next to the movie thumbnail used to have no results, but users thought that it would save the movie to a list of movies. (Right) In a later version, pressing the icon changed the current tab to *Reviews*.

Accidental Touches and Undo

One of the original usability heuristics is to give users control and support undo and redo. This is even more important on touchscreens because it's so easy to accidentally touch something that you didn't mean to. We've seen many users get lost because of an unintended touch, and then losing all their hard work or being completely disoriented because they didn't have a way to go back. Please refer also to our more extensive discussion on the role of the *Back* button in the subsection *Back Button* from the section *Navigation*.

46. Give users a way to undo the last action.

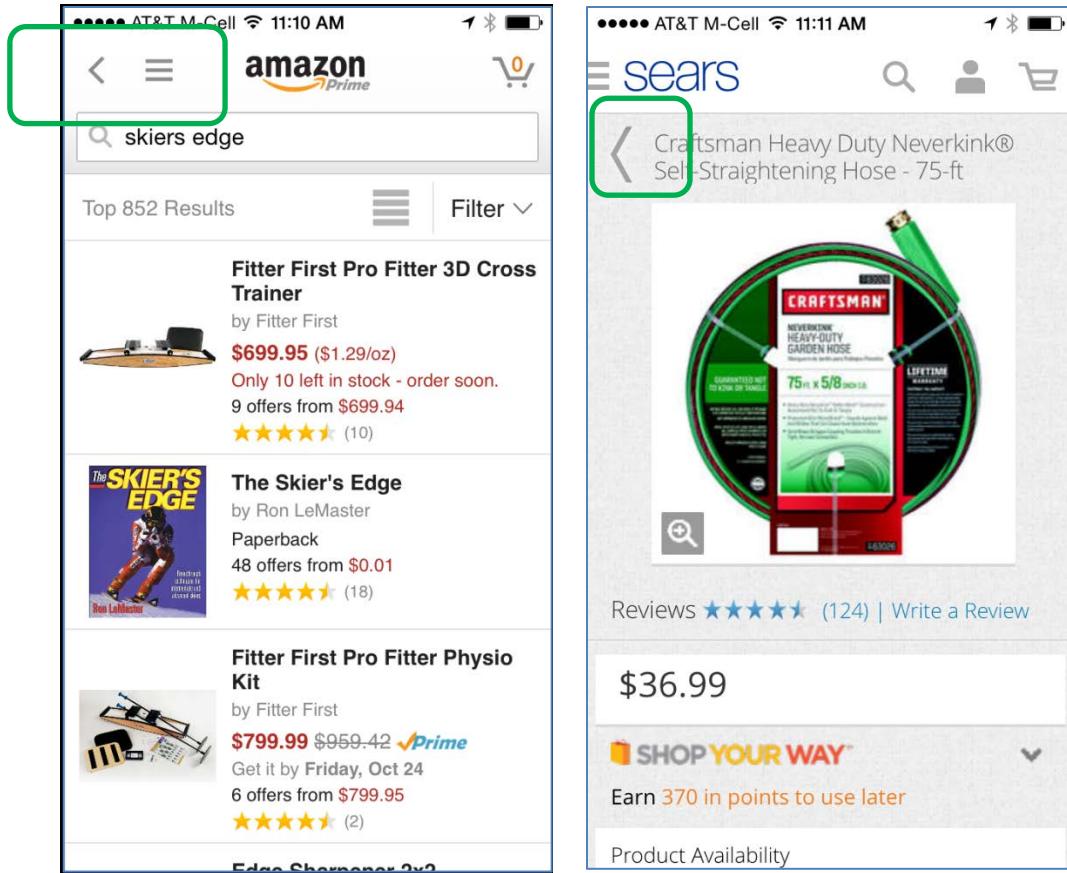
It's important that people be able to revert back to the state before an accidental action. Always allow them to either navigate back to a previous page, or, if the action is another type of command (e.g., delete, move), to undo it.



Gmail for Android: Users could swipe on a message to delete it. Once a message was deleted, users had the option to undo the delete.

47. [iOS, Apps] Include a *Back* button into your iOS app.

Phones running Android and Windows Phone have a physical *Back* button that (sometimes) saves the day: if properly used by the app, it can take the user back to the previous screen. But the iPhone has no *Back* button, so the interface must make up for it: most apps should include a *Back* button to allow the user to go back to the previous state.

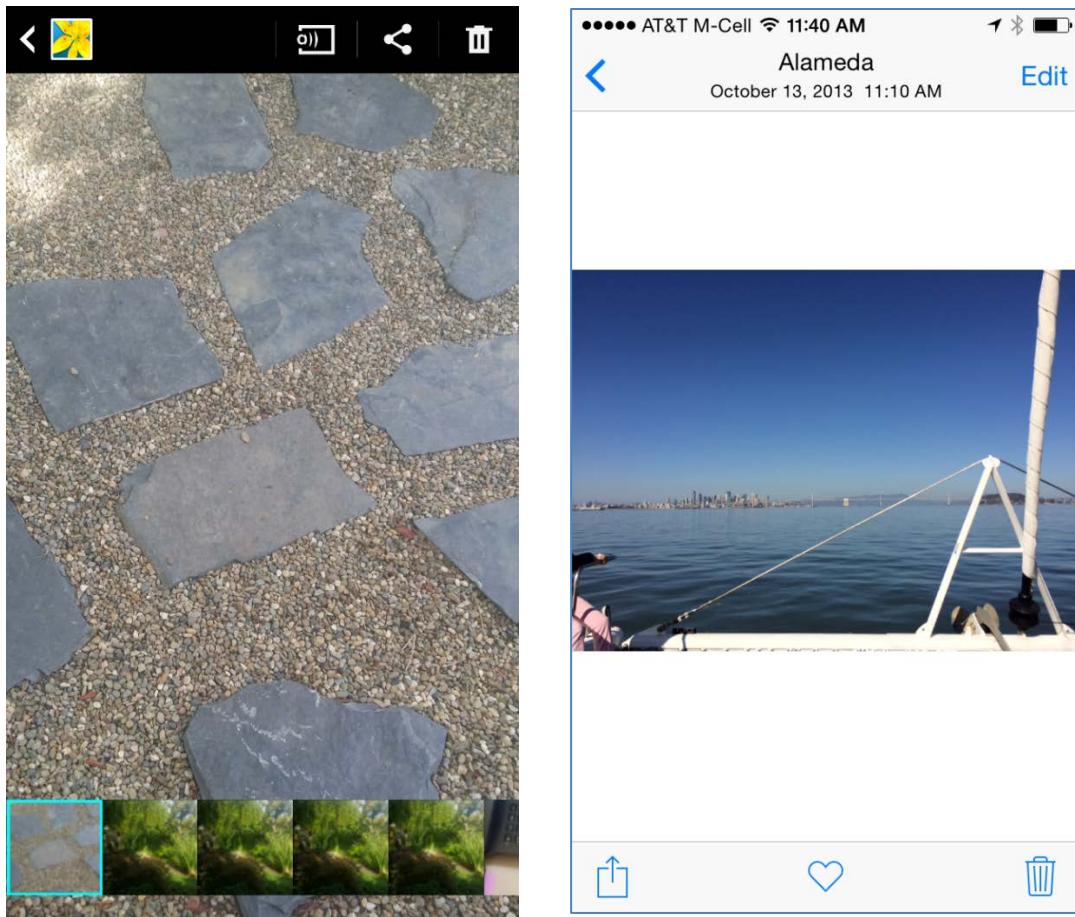


Amazon (left) and Sears (right): Both iPhone apps have a *Back* button that enables users to go back to the previous screen.

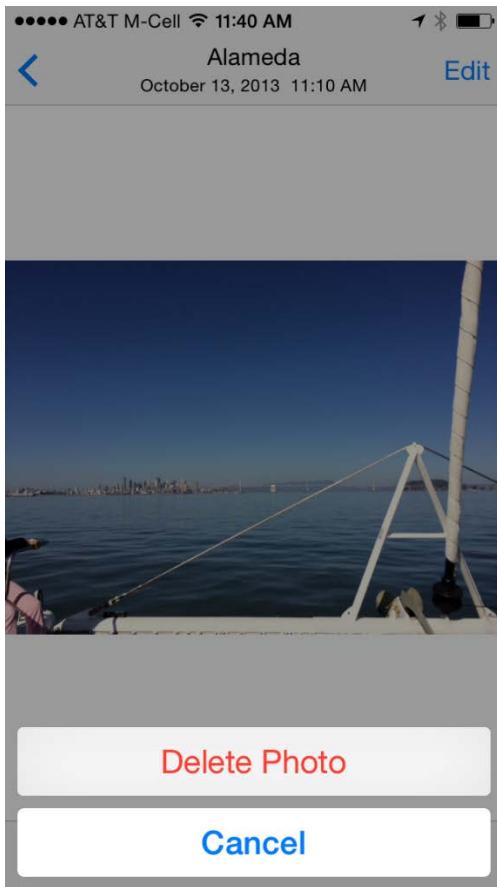
48. Consider placing destructive buttons far away from the physical buttons.
49. Always ask for confirmation before carrying out a destructive action.

The physical buttons tend to be used a lot. Any targets placed in the proximity of those buttons could be easily hit by mistake. Buttons that trigger destructive actions should be farther away to prevent accidental reach.

Moreover, you want to make sure that a destructive action (such as delete) is not initiated by accident. Wherever you place your destructive buttons, it's good practice to confirm the action with the user.



Two camera apps: Gallery for Android (left) correctly placed the *Delete* wastebasket button far away from the physical buttons (located at the bottom of the screen). Camera for iPhone (right) placed the *Delete* button at the bottom of the screen, closer to the iPhone's physical *Home* button. Both apps correctly asked users for confirmation when the *Delete* button was pressed.

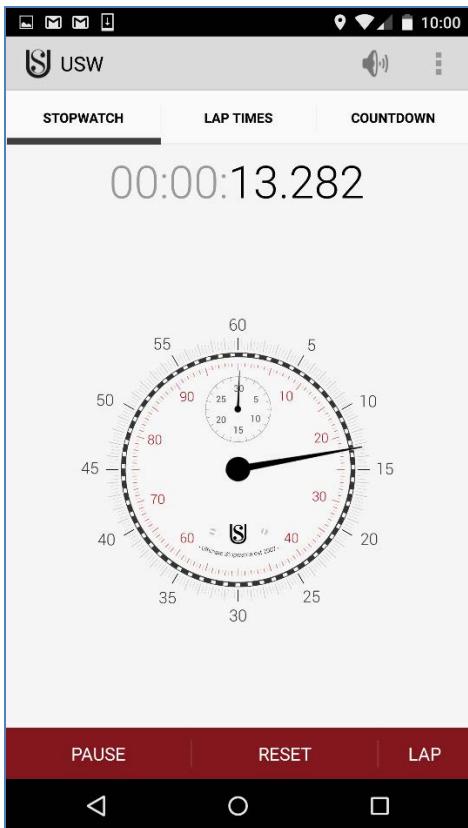


The Camera app for iPhone asked for confirmation once the user tapped on the *Delete* wastebasket icon (see previous example). Note also that the *Delete Photo* button was placed farthest from the iPhone's *Home* button (the *Cancel* button was the one closer to the physical button).

Skeuomorphism

Skeuomorphism in UX design refers to mimicking real-life objects in user interfaces. For instance, skeuomorphic design for a collection of books in a book-reading app may imitate a real bookshelf. Skeuomorphic interfaces do away with the traditional interface conventions and attempt to replicate an object with which people are already familiar. The assumption of a skeuomorphic interface is that users bring their prior experience with the physical object and thus, already know how to use the app without any explanation or training.

Skeuomorphism was a big trend in iOS designs prior to iOS 7; with the introduction of iOS 7 it became out of fashion and was replaced with flat design. Still, many designs retain skeuomorphic elements.

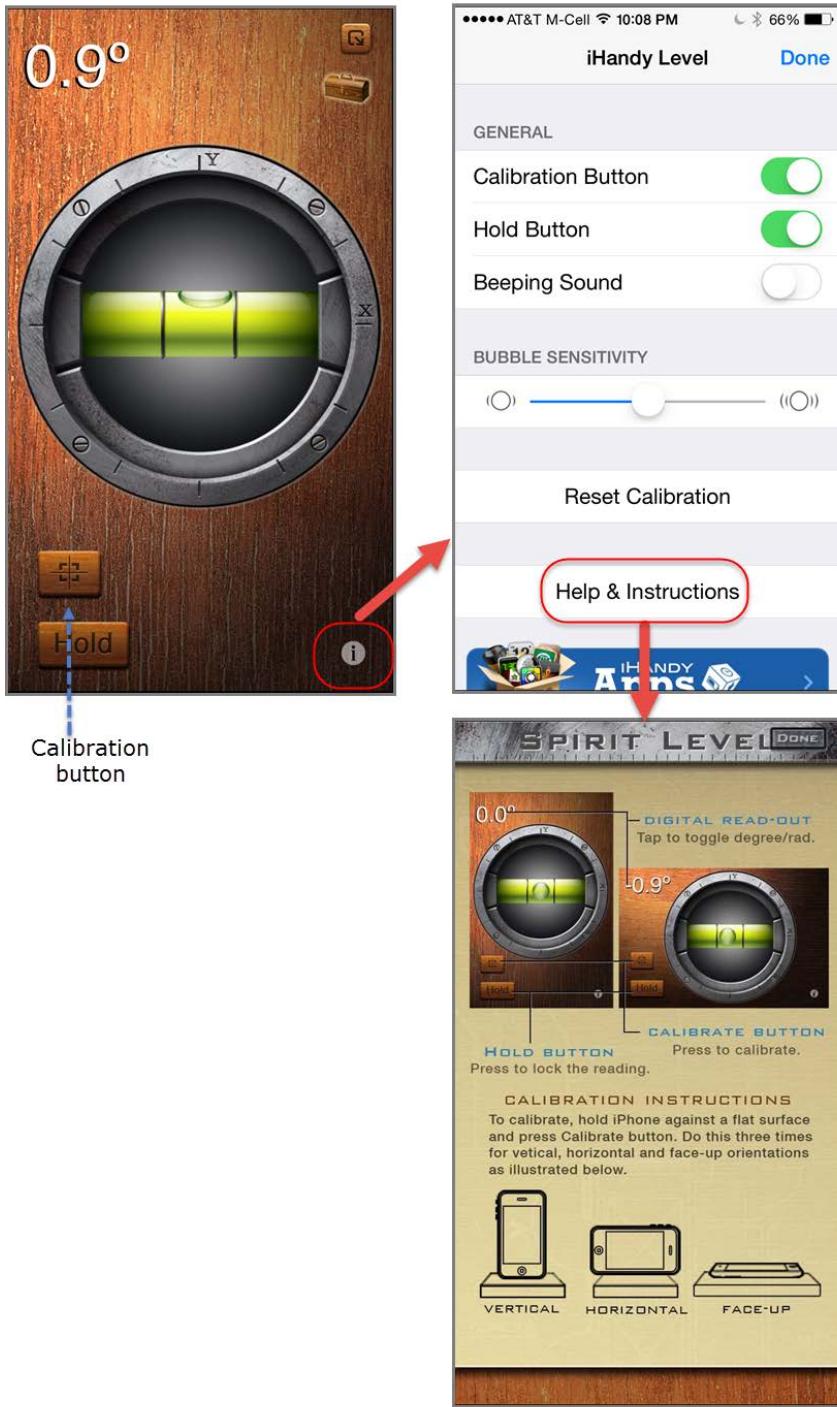


Skeuomorphic apps such as Ultimate Stopwatch for Android and iHandy Level for iPhone resembled real-world objects.

50. Don't use a skeuomorphic app if you cannot make it completely coherent with the users' expectations.

A successful skeuomorphic app should be easier to use than a different app with similar functionality that uses a traditional interface. If the skeuomorphic quality of the app makes any part of the task harder to accomplish, then the task is not a good candidate for a skeuomorphic app.

Let's take the example of iHandy Level. The app looked fairly intuitive — you were supposed to use it as regular leveler. Unfortunately, there was one part with no correspondence in the real world — namely, calibration. To use the app, users had to calibrate the leveler. Since they had no prior knowledge on leveler calibration, they had to figure out how to do it using the existing interface.



To calibrate the iHandy Level for iPhone, users had to click the *i* button, find the *Help and instructions* link, then study and memorize the instructions, go back to the main page, find the calibration button, then apply the instructions. A tedious process like this invalidates the whole purpose of a skeuomorphic app.

Even if people discovered the calibration button, that would not be enough. They would then have to figure out how to use the button, by getting to a page of instructions, memorizing the instructions ("To calibrate, hold iPhone against a flat surface and press *Calibrate* button. Do this three times for vertical, horizontal, and face-up orientations as illustrated below"), and then coming back to the main app page. One of our study participants went back and forth between the main screen and the instructions several times until he was able to put the instructions in practice.

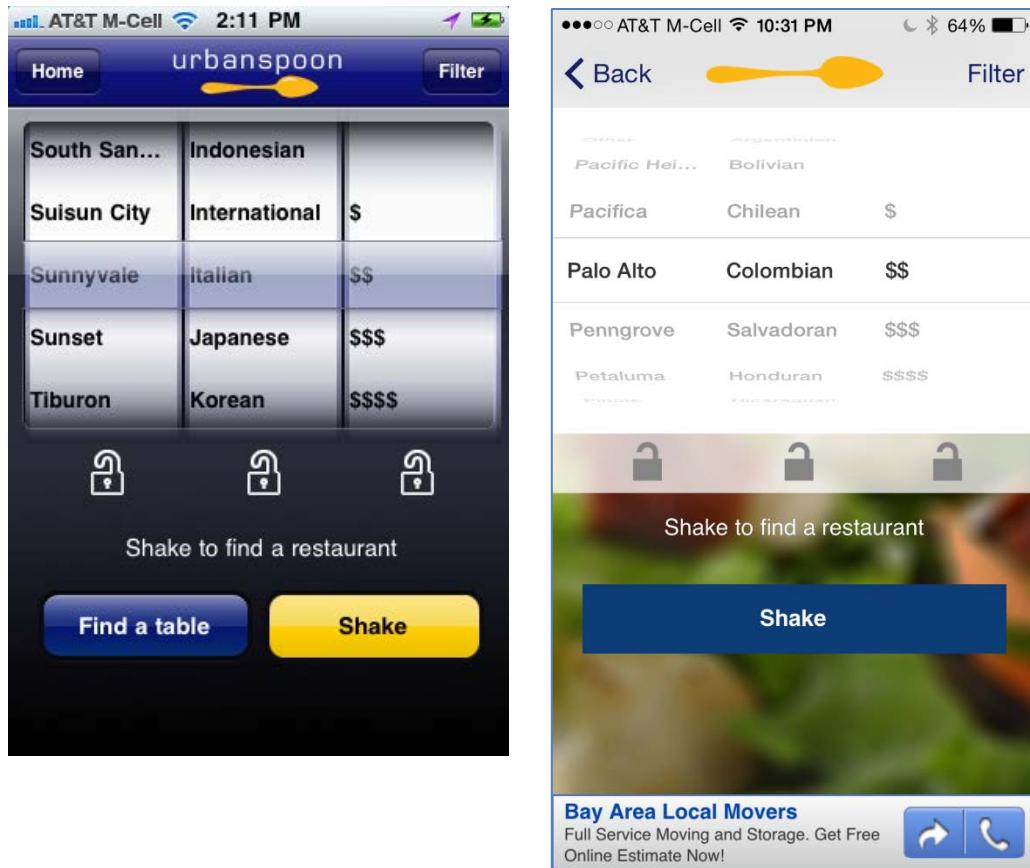
The conclusion is to be cautious when designing a skeuomorphic app. Make sure that the tasks supported by the app have a good mapping onto what people already know from their interaction with the real-world object. Any clashes between the application's mental model and the users' mental model can be fatal.

- 51. Even if the app is not completely skeuomorphic, well-selected skeuomorphic elements can make it more enjoyable.**
- 52. Make sure that the skeuomorphic elements make the users' task easier rather than harder.**

Skeuomorphism is not all or nothing. Apps like Urban Spoon or Web MD were not completely skeuomorphic. In fact, the majority of their screens were quite standard. But in their attempt to make the app more unique, they added skeuomorphic elements. Some were more successful than others.

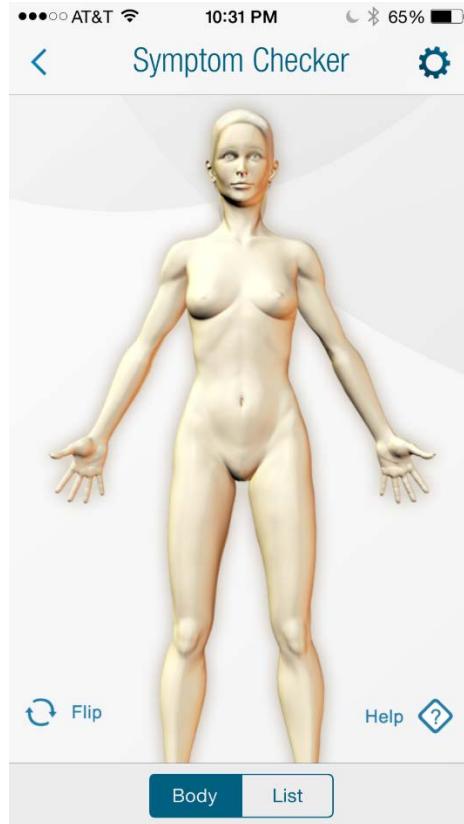
Thus, selecting a restaurant in Urban Spoon was meant to look like playing a slot machine. However, unlike with slot machines, users had to select the values in each of the spinning columns. They also had to lock the locks underneath each column. And then they could either shake or just press one of the two buttons. Their action triggered a noise similar to the slot machine, and the result was a restaurant.

If you think about this process, you will realize that it's not quite like a slot machine — it works more in the opposite way (you get one result, but set three inputs). Most importantly, the locks under the columns have no place in the slot-machine mental model. In fact, almost every time we tested that app, users forgot to lock the locks. Even people who said they use the app regularly forgot to do so.



Urban Spoon for iOS (older version on the left and newer one on the right) had skeuomorphic design elements inspired by slot machines.

In WebMD, another partially skeuomorphic app, in order to diagnose a disease, users had to select a body part where the disease symptoms were localized. Then, they needed to choose their symptoms from a list. It turned out that some users did not necessarily localize their symptoms in the same way as the app (for instance, users placed "fever" in the head, whereas the app thought it was a general symptom that apparently had no mapping on the body). Again, just using a regular list interface would have probably been more effective in this situation.



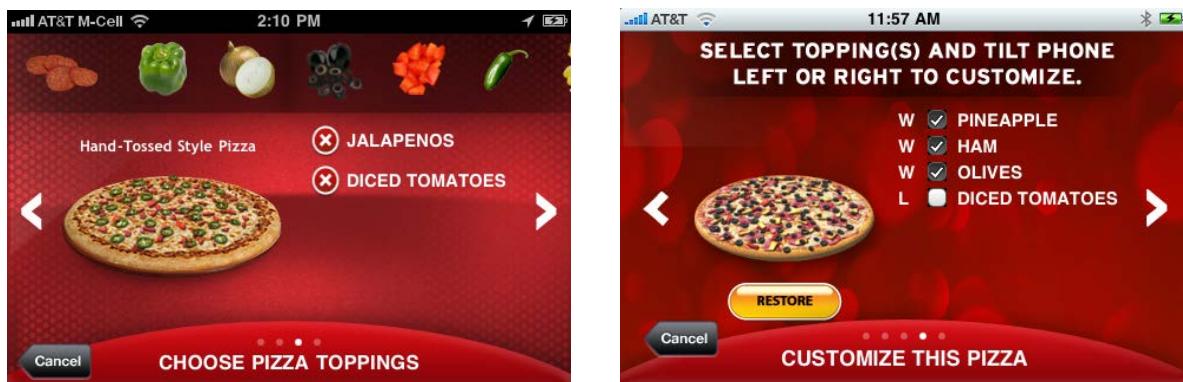
WebMD users had to select a body part to indicate the location of their symptoms.

A prior version of Motor Trend, an app for car-related news, had a unique homepage that fit the profile of the app — the homepage resembled a dashboard, with different controls corresponding to different topics. The mapping was quite simple, and, since the only task available to the user was selecting one of the topics, it worked.



An older version of Motor Trend for iPhone had a homepage that resembled a car dashboard.

Finally, an older Pizza Hut version allowed users to customize their pizza by dragging toppings on it. It was very intuitive (all our participants quickly figured it out) and mirrored the real-life process. Unfortunately, Pizza Hut also had a screen where people were asked to “select toppings and tilt phone left or right to customize.” None of our participants understood what they were supposed to do and what the outcome was supposed to be: although they all tilted their phones, none checked the different toppings to indicate which of them would slide on half the pizza. One departure from the real-world mental model caused the app to fail.



Pizza Hut app for iPhone (older) required users to tilt their phone to slide the toppings on one side of the pizza. Users did not understand what tilting the phone would achieve.

The Barnes and Noble's Nook for Android used a page-turning animation, presumably to follow the real-time image of a turning page and to make the app more similar to a real book. One participant commented:

"Here you get all that turning of the page, which I guess, esthetically [it's pleasing] — but to me, I just want to read, 'cause you don't have very much on the page, so when I want to get to the next page I don't want to wait for this [animation]."

That's what the problem is with many skeuomorphic elements: they need to make the app aesthetically pleasing AND not get in the way in doing so.

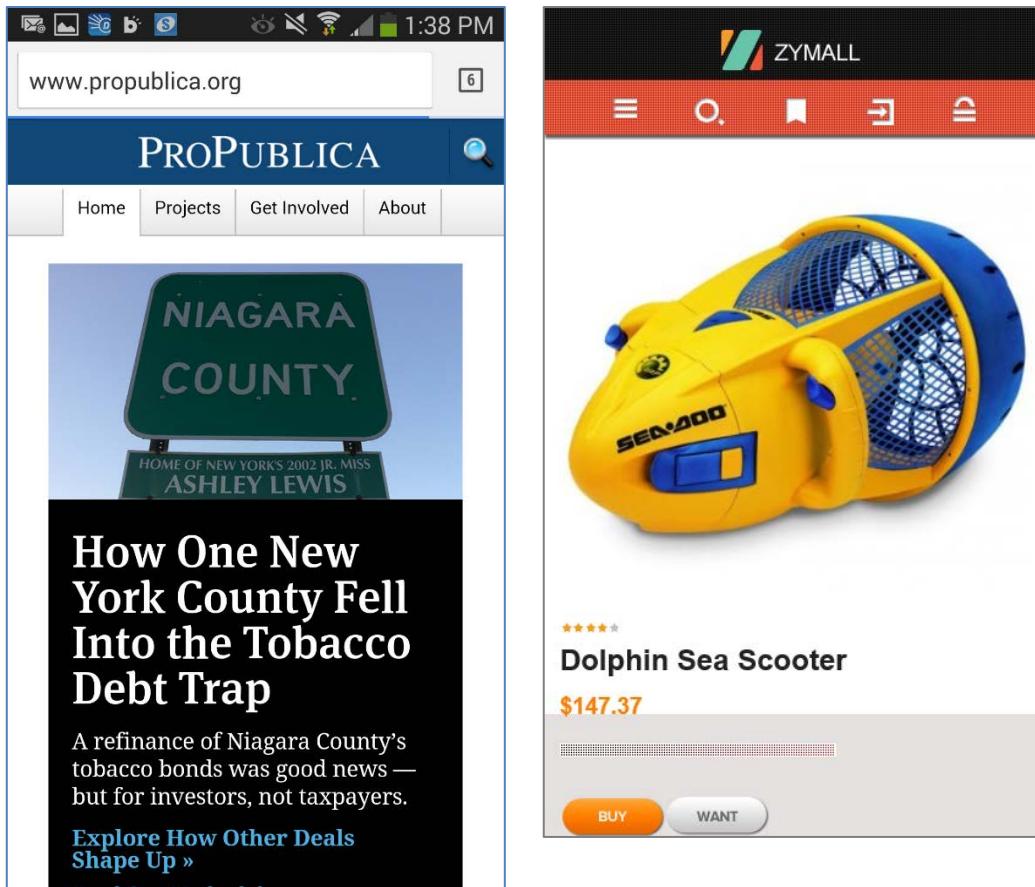
ICONS

Icons are attractive, can convey brand personality, they usually make for good targets on a touchscreen, and can sometimes save space on the small mobile screen. However, when poorly chosen, they can cause serious usability issues³⁶.

53. Use standard icons whenever possible.

Instead of designing your own icons, prefer preexisting, OS- or browser-specific icons. It's likely that your users are already familiar with these icons and will recognize them a lot faster than any new (albeit more beautiful) icon that you may come up with. Don't be tempted to take a standard icon and embellish it with a more styled graphical design: you will lose the advantage of fast recognition that common icons have.

³⁶ More on the usability of icons: Aurora Bedford. "Icon Usability."
<http://www.nngroup.com/articles/icon-usability/>

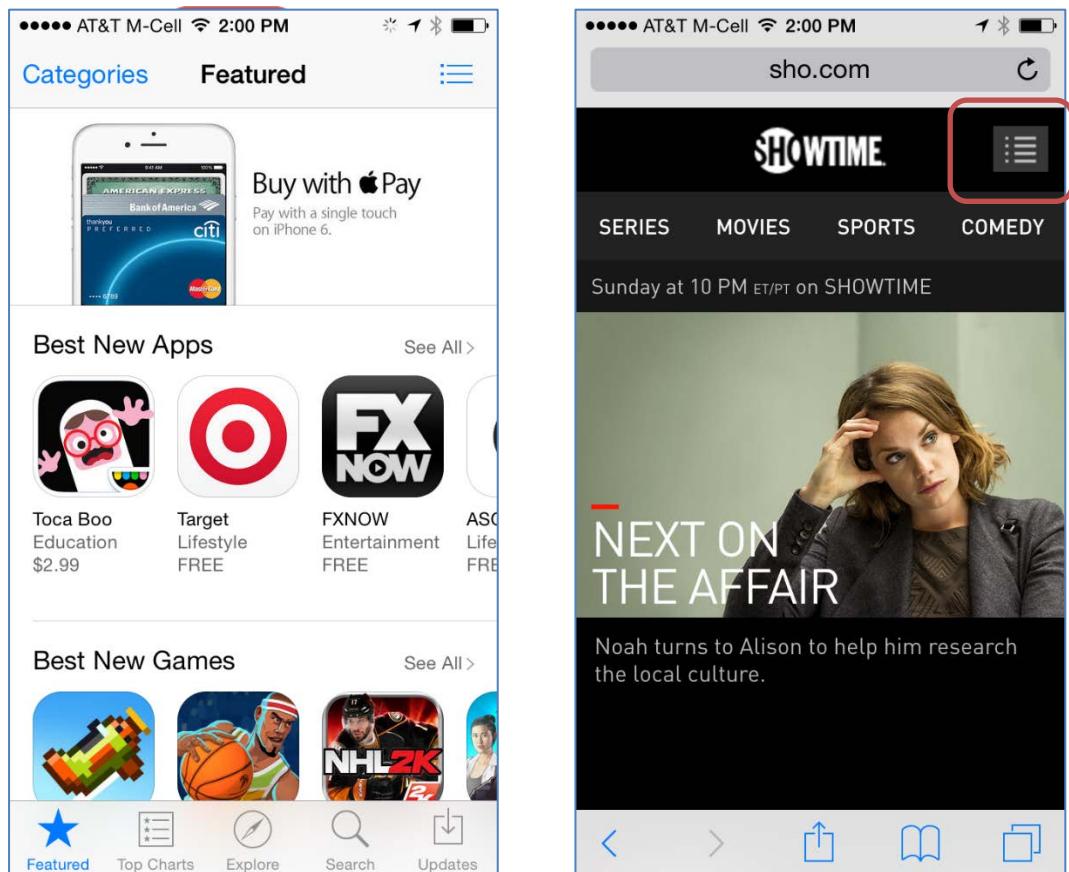


Self-defined icons replacing standard icons: On Propublica.org (left) users had trouble recognizing the magnifier glass in the top right corner as a search tool. On Zymall.com (right) the icons at the top of the page were stylized versions of standard icons and were hard to interpret and recognize.

54. Use standard icons in standard ways.

Users expect standard icons to behave the same everywhere. If you end up using standard icons in your app or on your website, make sure you preserve their standard meaning. Don't mislead people by hiding unexpected functionality under a well-known icon.

This is just one incarnation of a basic usability principle: consistency. By using familiar icons in a familiar way, you take advantage of users' knowledge and make it easy for them to figure out the interface of your app.



The same icon is used differently: in the iOS App Store app (left) it stands for wish list, while the Showtime website (right) uses it for the main navigation. (Showtime presumably was confused by the resemblance between the list icon and the hamburger icon.)

55. Use icons with good information scent³⁷.

There are three types of icons: **resemblance icons** (that look like the real-life object that they depict), **reference icons** (that are based on a metaphor or analogy), and **arbitrary icons** (based on convention and prior knowledge)³⁸. When creating a new icon, resemblance icons may seem the safest, but for more abstract concepts it can be hard to come up with a resemblance icon that has a good information scent.

- (a) There is no good recipe for creating icons that convey meaning appropriately. The best thing to do is to come up with several alternatives and test them by

³⁷ Information scent of an icon or link refers to those cues that tell users what to expect to happen when they clicked on that icon or link. For a more detailed discussion see Jakob Nielsen. "Information Foraging: Why Google Makes People Leave Your Site Faster." <http://www.nngroup.com/articles/information-scent/>

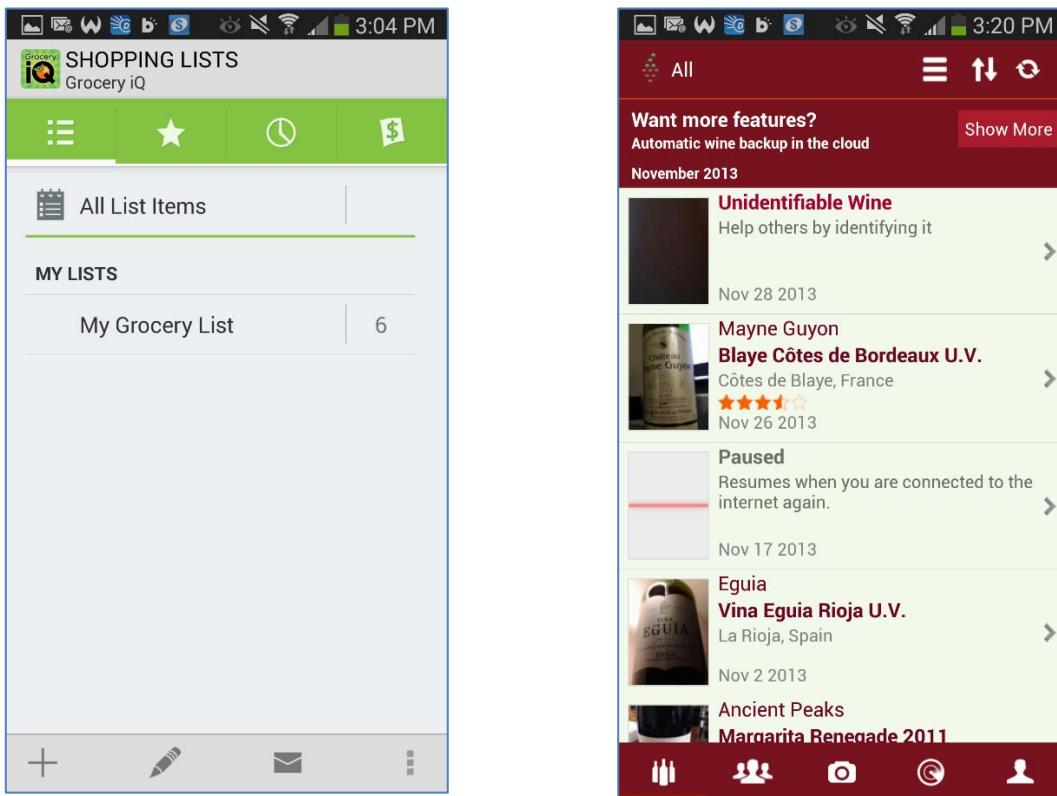
³⁸ For a more extensive discussion of these types of icons, see Jakob Nielsen. "Icon Classification: Resemblance, Reference, and Arbitrary Icons."

<http://www.nngroup.com/articles/classifying-icons/>

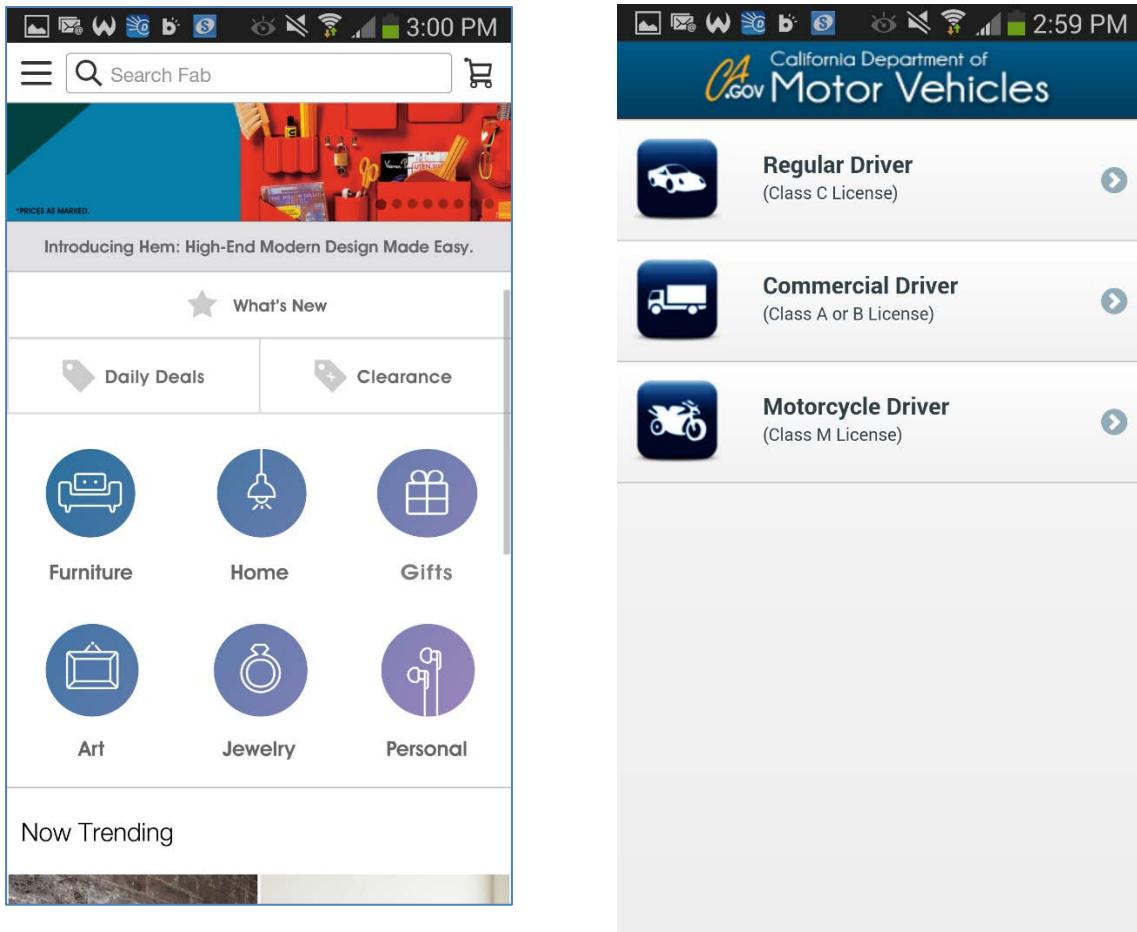
asking people what each of these candidate icons could stand for. This doesn't have to be a full-fledged usability test, although having the right context cannot hurt. But even an online survey where you ask people what they expect to find under those icons can be indicative of whether your icons are on the right track.

56. Include labels with your icons.

Because finding icons with good information scent can be difficult (and also because the same image may suggest different things to different people), it's always best to enrich the icon by attaching a label to it. In that way, your information scent is automatically increased by the words. (Of course, you must choose good words for the labels.)



Insufficiently explanatory labels: Grocery IQ for Android (left): Some of the icons in this app would have benefited from a label (for instance, it was harder to figure out what the clock icon stood for). Vivino for Android (right): Even though the objects depicted may have been identifiable, it was hard to know what the different icons meant in this app's context. Labels would have improved the information scent of the icons.



Fab for Android and CA.gov DMV for Android correctly used icons AND labels for their product categories and different types of drivers, respectively.

TYPING AND INPUT FIELDS

As we discussed in the section *Mobile Limitations and Strengths*, typing on touchscreens is difficult for two reasons: (1) the relatively small keys on the soft keypad make it easy to commit mistakes; and (2) in the absence of haptic feedback, the user's attention must be divided between the keyboard and top part of the screen that shows the text being typed.

Our general recommendation for small-screen devices is to limit the amount of typing and do as much of the users' work as possible. Next, we make specific recommendations intended to minimize typing.

57. **For any input field, consider if you could eliminate it and save the user some work.**
58. **Consider eliminating optional fields from mobile forms to make the form more compact.**

Many forms contain fields that are not absolutely necessary, but nice to have. Even when those nice-to-have fields are optional, they still make forms look long and daunting, and increase the chance of people abandoning the task.

••••• AT&T M-Cell 10:14 AM

mobile.southwest.com

Personal Information
First, Middle and Last name must match your government issued photo identification.

First Name

Middle Name (Optional)

Last Name

Preferred Name (Optional)

Suffix (optional)

Social Security # (Last 4 digits) (Optional)

Gender

Date of Birth

Month Day

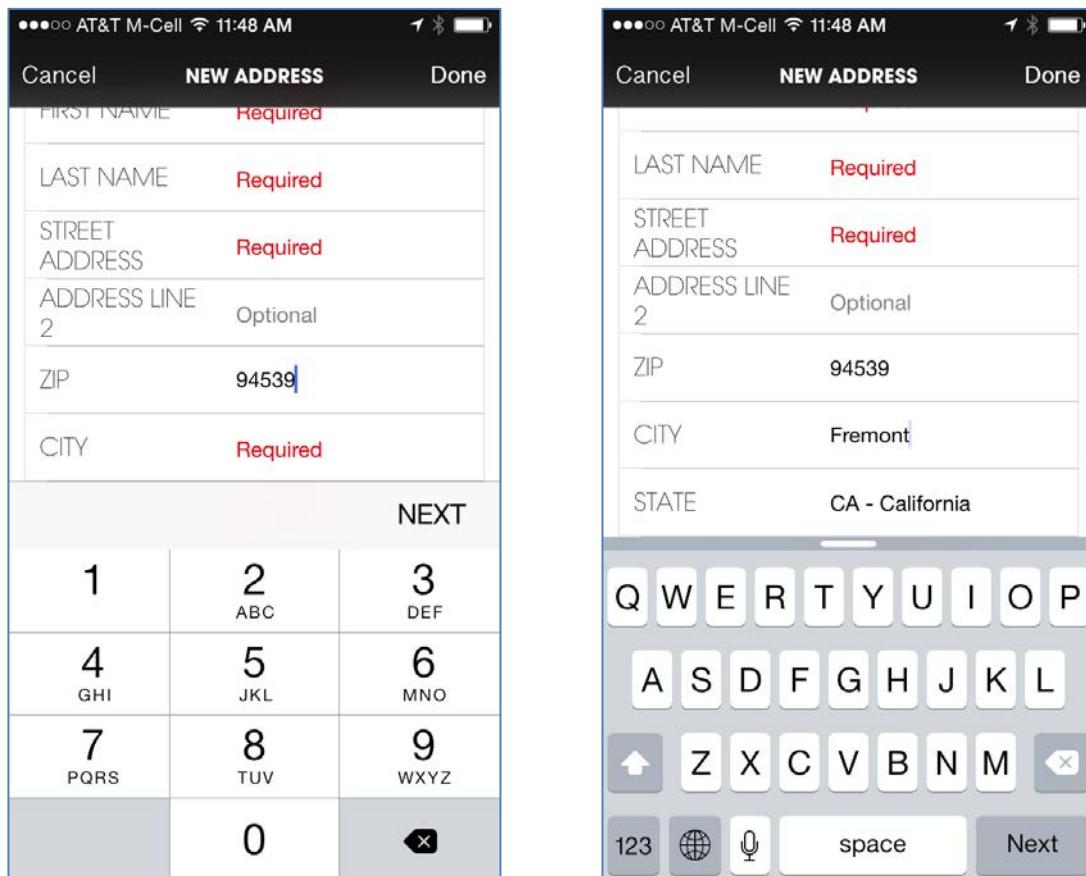
Year

< > ⌂ ⌃ ⌁

Southwest.com: Enrolling for the frequent flier program had several fields that were optional and made the form look long. Instead, it would have been better if only the essential information had been requested (first and last name, possibly birth year if any age verifications were needed). The other fields (*Preferred name, Suffix, Social Security Number, Gender*) should have been delegated to a profile that the user could fill in at a later date.

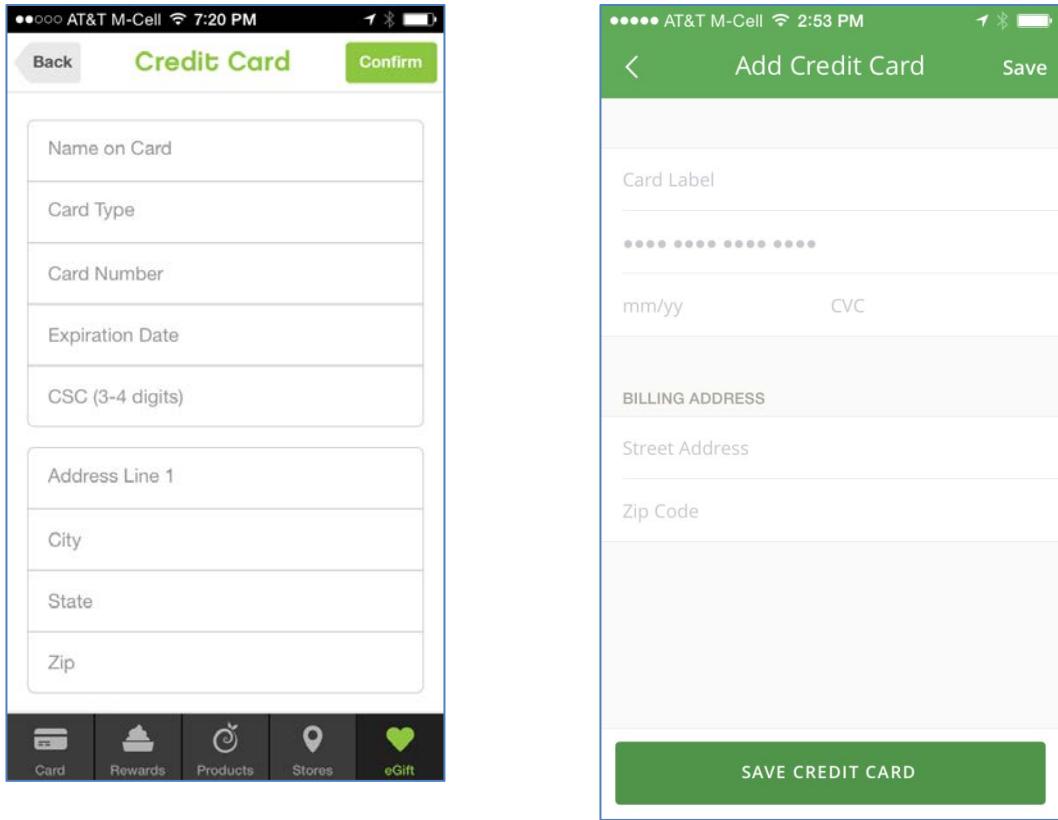
59. Where possible, compute field values rather than asking the users to enter them.

A lot of information can actually be computed and filled in for the user. Credit-card type can be inferred from the credit-card number; city and state can often be determined based on zip code³⁹; the default country of destination can match the country where the phone is being used.



Sephora for iPhone: After the user filled in the zip code (left), the city and state were automatically filled in by the app (right).

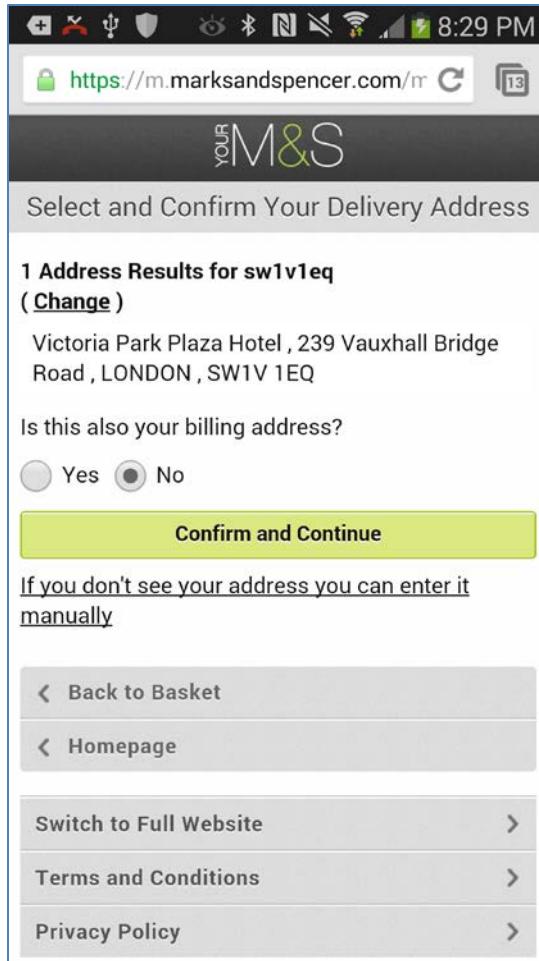
³⁹ Even if there were several cities that correspond to the same zip code, selecting from a short list would be easier than typing the whole city name.



Don't ask for redundant information: Pinkberry for iPhone (left) required users to enter the credit-card type in a separate field instead of computing it for them. Instacard for iPhone (right) did not require users to enter a credit-card type, or a city or state for their credit-card billing address. (The design could have been improved by dispensing of the *Card Label* field and by placing field descriptions above fields — see guideline 63.)

In UK we had the pleasure of seeing this idea at work: on all the UK sites that we tested, once users entered their postcode, they could select their complete street address from a short list. (It turns out that in the UK, there are just a few addresses that have the same postcode.)

The following example is from Marks and Spencer's mobile website. Once the delivery postcode was entered, the user was prompted to select the correct address.



Marks and Spencer asked users to enter a delivery postcode and computed the delivery address based on it.

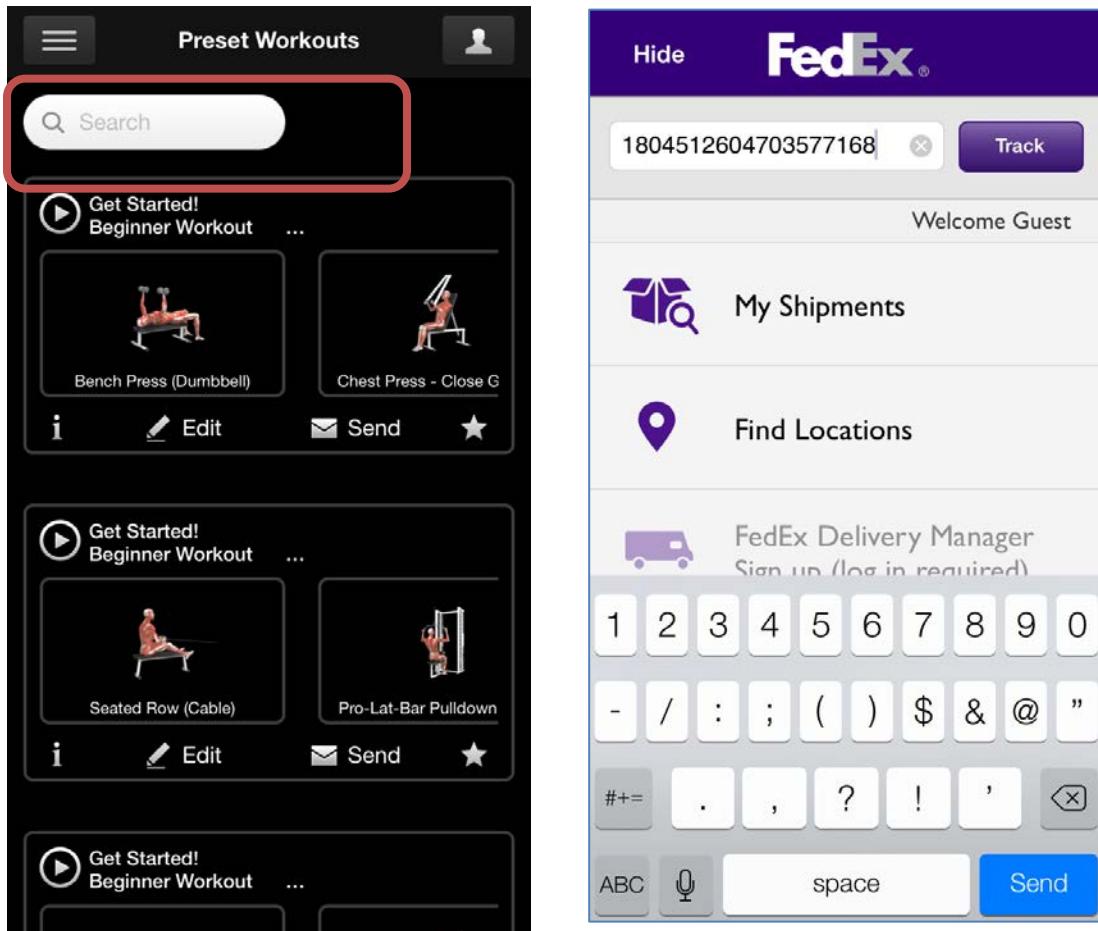
60. Make textboxes long enough so that users don't have to scroll within.

Anybody who has tried to scroll within a textbox on a touchscreen with no trackball knows it's no easy task. Because scrolling within the textbox is so tedious, if people want to modify the input string, they simply delete the last characters of the string until they get to where they want to make the change; then, they type again. But even so, hiding part of the text in the textbox is conducive to mistakes — users don't see all the characters that they have entered and can easily overlook a typo.

One of our iPhone users was typing a long search string and was annoyed that he couldn't easily scroll within the textbox:

"I look at the results, and it returned no articles in the past 30 days... Hmm... well, that's because I misspelled 'salmonella'... Now I need to go back, check my typing... I hate it when that happens... yes, there's the 'i', should be an 'o': 's-a-l-m-o-n' ... I've got to tell you, I don't like it when you get to these windows where you can't necessarily see all the text you've typed in, it's kind of annoying... I want to advance to

the part where my error is and I can't do that because the window is too small, so I'm going to make a guess that I've correctly spelled 'salmonella' but..."



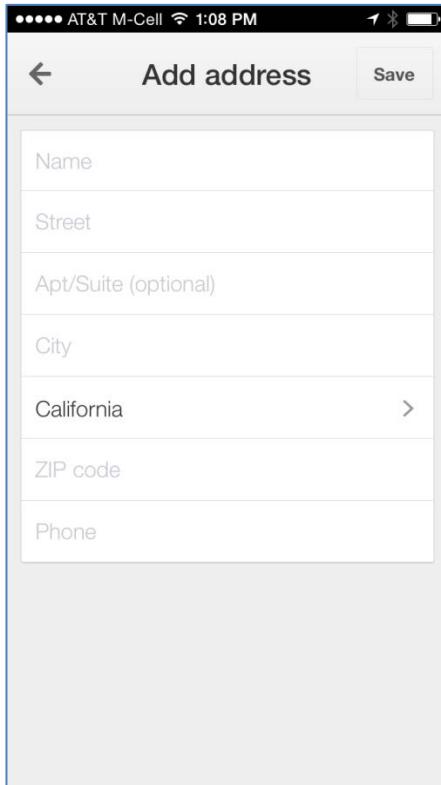
Textboxes should be long enough to accommodate the string that goes inside. iMuscle2 for iPhone (left): The search box was too short, although there was plenty of empty space next to it that could have been used to make it wider. FedEx for iPhone (right): the tracking-number box was also too short: there was not enough space for all the digits of a FedEx tracking number. The design should have pushed the *Track* button under the input box to make space for the full tracking number.

61. If you know how long a certain field is going to be, make it that size.

The length of the field can give users cues about the content that goes within and help them quickly assess how much typing they have to do. It can also prevent them from making errors.

For instance, US zip codes have 5 digits, so a zip-code field doesn't need to be wider than that. For many fields (e.g., names) it is hard to predict the actual

length, but for others (phone numbers, credit-card numbers), the length is known at design time.



Google Shopping Express for iPhone: All fields were the same length. It would have been better if the ZIP code and the phone fields had been sized according to the correct input length. Note also how the field descriptors are placed inside the field instead of above the field (see guideline 63).

Note that this is not a show stopper: users will likely fill in the form successfully even if all the input boxes have the same size. The main benefit of following this guideline is that users will likely be faster in completing the form.

62. Place description text above the textbox, not in line with the textbox.

Of course, the size of the textbox is limited by the screen width: you cannot (or shouldn't) design textboxes that are wider than the screen width. But be generous with the textbox length, especially if you have extensive variability in the possible input strings.

You can allocate more space for the textbox by placing the field descriptions above them, and not in line with them.

The image displays two mobile application screens side-by-side, illustrating poor form design practices regarding field descriptions.

Left Screen (MyFitnessPal):

- Create Food** screen.
- Brand Name:** Input field containing "Isn't believe it's not butter".
- Description:** Input field containing "ex. Chicken Soup".
- Servings** section.
- Serving Size:** Input field containing "ex. 1 cup".
- Servings per container:** Input field containing "1".

Right Screen (Walgreens):

- Register** screen.
- Registration Information** section.
- First Name:** Input field.
- Last Name:** Input field.
- Email:** Input field containing "emailaddress@email.address.com".
- Password:** Input field containing "Min 8chr 1#, CaseSensitive".
- Re-Type:** Input field.
- Zip Code:** Input field.
- Mobile #:** Input field.
- D.O.B.:** Input field.
- Save Username:** Checkbox checked.
- Save Password:** Checkbox unchecked.
- NEXT** button at the bottom.

Field descriptions should not be placed above the input box, not in line with it. My Fitness Pal for Android (left) and Walgreens for Android (right) both restricted the size of visible input by placing descriptions next to the input boxes.

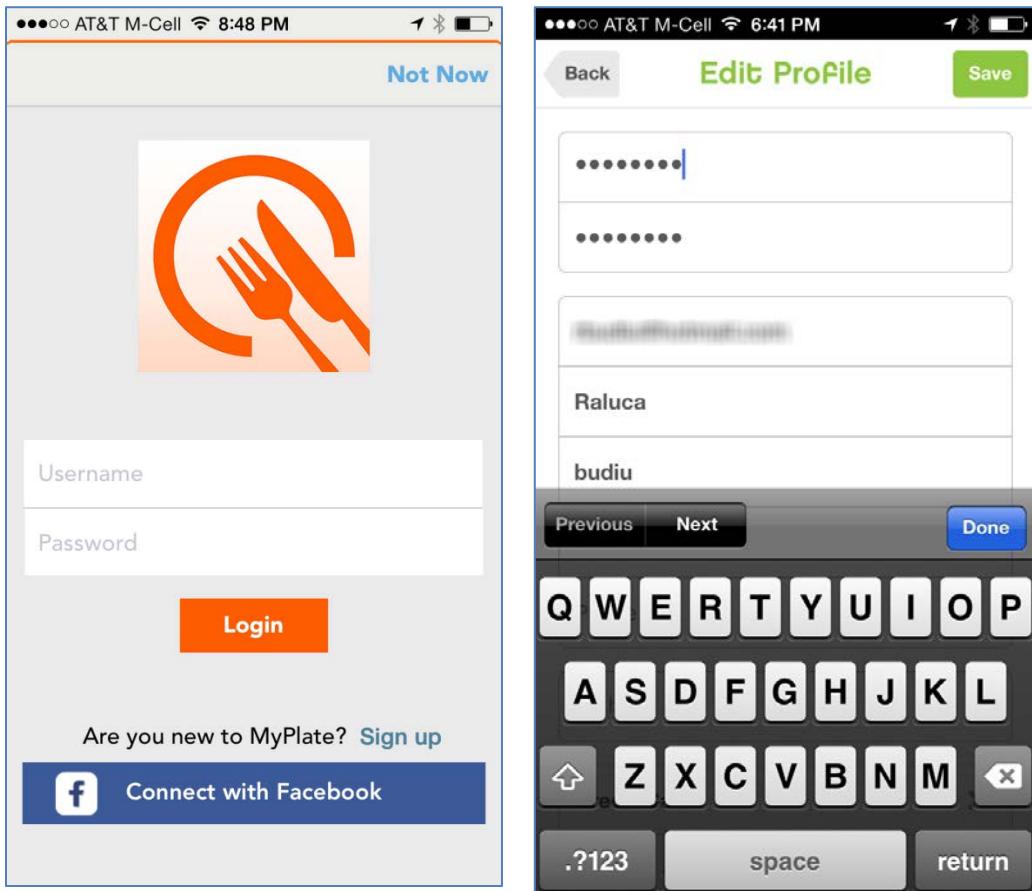
63. Do not use placeholders inside textboxes.

Whenever possible, placeholders (descriptions of fields or examples of field values placed inside a textbox) should not be used⁴⁰.

It's better to show the description of the field or an example of input either above or next to the field. This will also allow the users to see the field description or the example while they are typing, instead of having to remember it. Especially on mobile, it's easy for users to get interrupted and forget what fields they were filling later on when they resume the activity. And

⁴⁰ This guideline also applies on desktop. Here is a more detailed discussion of why placeholders should not be used: Katie Sherwin. "Placeholders in Form Fields Are Harmful." <http://www.nngroup.com/articles/form-design-placeholders/>

sometimes information such as “Passwords must be 6-8 characters and contain a capital letter” may be hard to remember even if it was just read.



Placeholders in form fields are harmful: MyPlate for iPhone (left): If users started typing and got interrupted, they might forget whether the first field was a username or an email address. Pinkberry app for iPhone (right): The lack of field descriptors made editing the profile challenging. What were the first two fields? Was one of them a username or were they both passwords? (Normally, if the form had not been filled at all, placeholders with the field descriptors would have appeared inside the textboxes.)

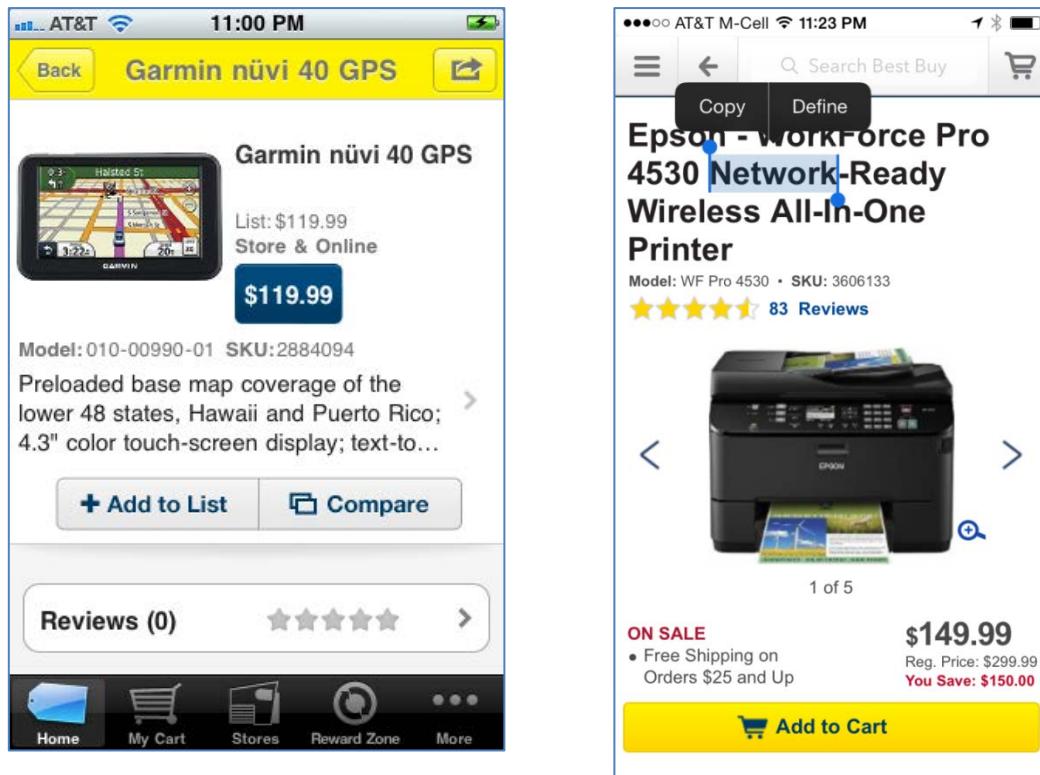
- 64. Allow users to copy any content that you may present to them.**
- 65. Allow users to paste information into an input field.**

Cut-and-paste can save precious typing effort; make sure you support it throughout your app.

One of our participants was using the Salesforce app on his iPhone to manage his many contacts at work. He shared that his company had an agreement with Skype, so that whenever he needed to call internationally he could save money by using Skype. The participant proceeded to show us how he would call a

contact: he called in Salesforce (using the regular Phone app), then he quickly canceled the call and copied the number that he had just called. Next, he opened Skype and wanted to paste the number into the dial box; unfortunately, that version of Skype no longer supported the paste operation. (Subsequently Skype went back to supporting it.)

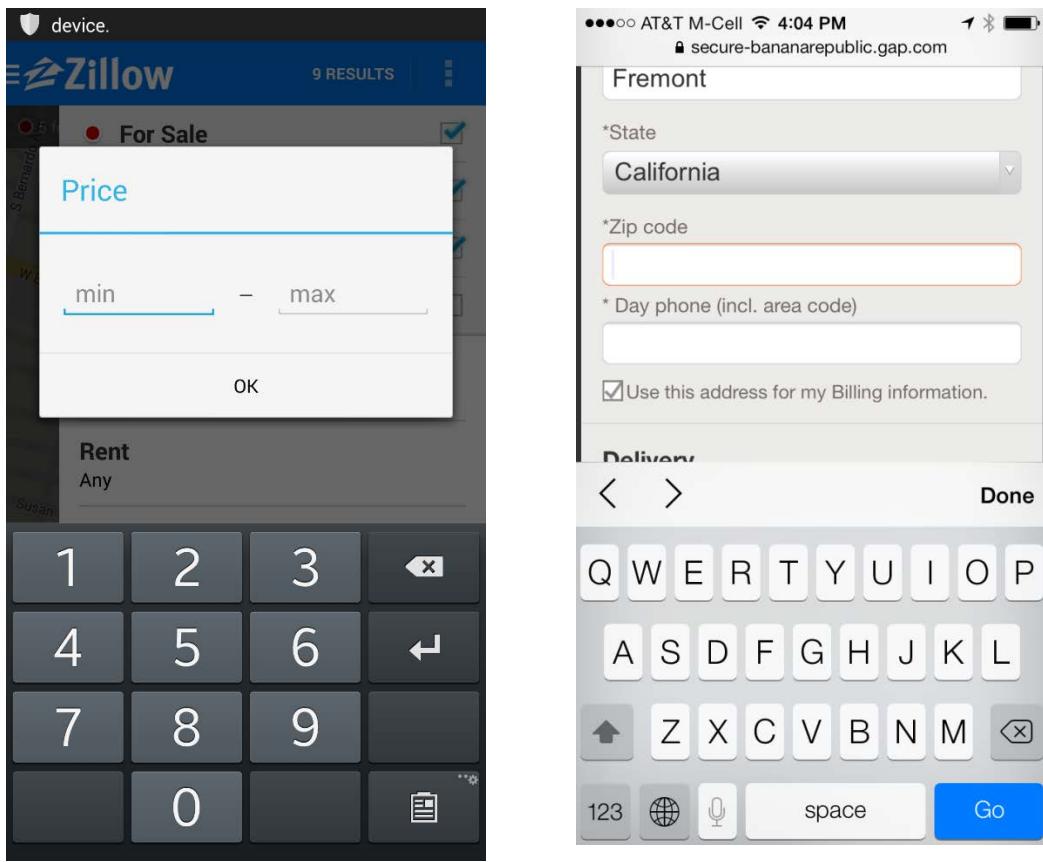
Another participant searched for car navigators in his Best Buy Windows Phone 7 application. The application had no reviews for the product that he was interested in, so he decided to go to Amazon to find reviews. He wanted to copy the product name and paste it into the Amazon app — however, that was not possible because Best Buy did not allow its content to be copied.



An older version of Best Buy (left) did not allow users to copy content such as product names from his app. Newer versions (right) allow copying information from the product page.

66. Present users with the keyboard appropriate for the input field.

If a certain field can only contain a limited set of characters, use a keyboard that only supports those characters. Don't force users to switch keyboards. Examples of fields with restricted characters include numerical fields, zip codes, telephone numbers, and credit-card numbers.



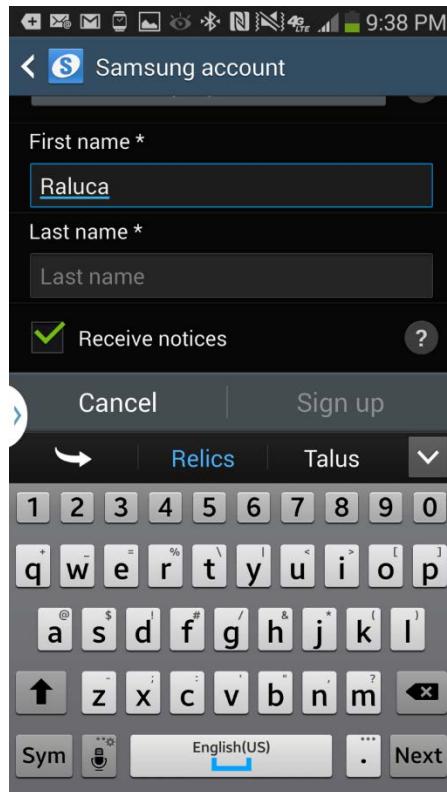
Show users the keyboard appropriate for the type of input field: Zillow for Android (left) appropriately presented a numerical keyboard for fields such as *Price*. Banana Republic's website (right) forced customers to switch keyboards when entering the zip code.

67. Use autocomplete and suggestions whenever users fill in a textbox.

Most modern phones already incorporate language-based suggestions in the keyboard. You can take this one step further by using knowledge about the input field (e.g., based on site analytics or on possible values for that field) to make further informed suggestions and completions.

68. Do not use autocomplete and suggestions for fields that are highly singular — such as email or names. (Do use history or other information that you may have for those fields.)

As much as autocomplete and suggestions are useful for general words, they can be utterly annoying when it comes to names, emails, addresses, or other fields that contain fairly unique information. Unless you can make an educated guess for the value of that field (based, for instance, on user history), it's better to let the user type it without help.

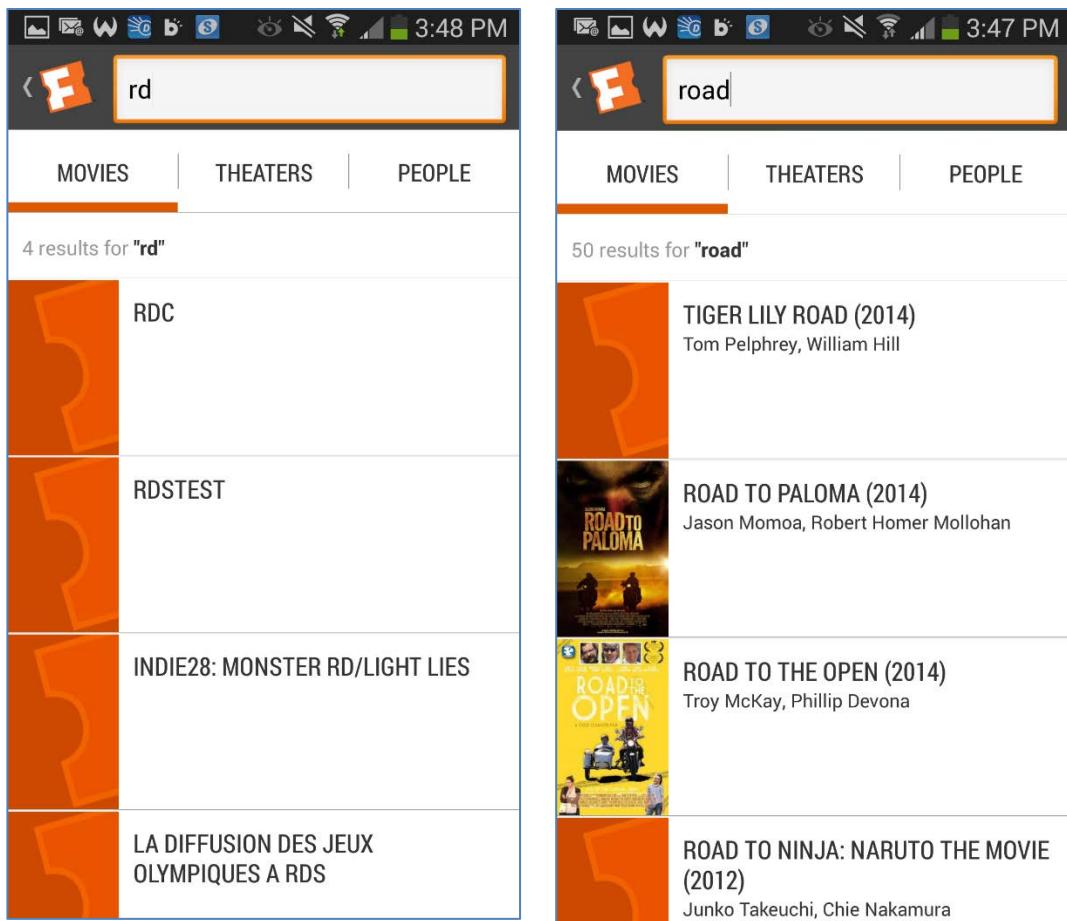


Setting up a Samsung account on a Samsung phone. The app uses autosuggest for names, and does an embarrassing job at it.

69. Allow for typos and abbreviations.

The harder it is to type, the more likely users are to make mistakes. Instead of exact string matches, you should do partial matching on the input strings whenever the entered strings are used for a query in a database. You should also watch for common abbreviations.

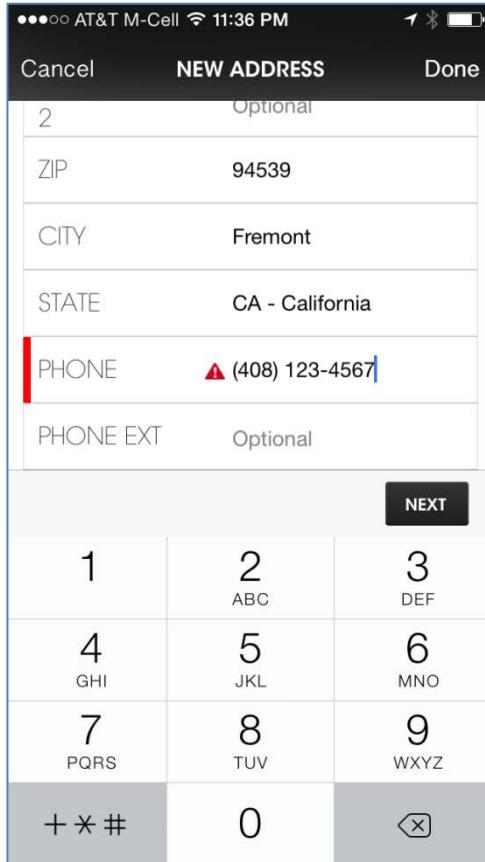
In one of our usability sessions, a participant searched for the movie "Revolutionary Road", which he spelled "Revolutionary rd". He was not able to find any results, because "rd" was not expanded to "road". The figure below shows the Fandango search-results page when the search string was "road" versus "rd".



Mobile apps and websites should tolerate typos and abbreviations: (Left) Fandango app for Android returned four results when we searched for "rd". (Right) The same app found 50 matches when we searched for "road".

70. Autoformat fields for users. Don't force them to use characters such as dash or space to achieve a specific format.

Fields such as phone numbers or credit-card numbers often contain delimiters (such as the dash or space character) to make them easier to read. However, delimiters also make these fields harder to type. Don't force the users to format a field in any way; let them type the entire string and then properly format it so that it's readable.



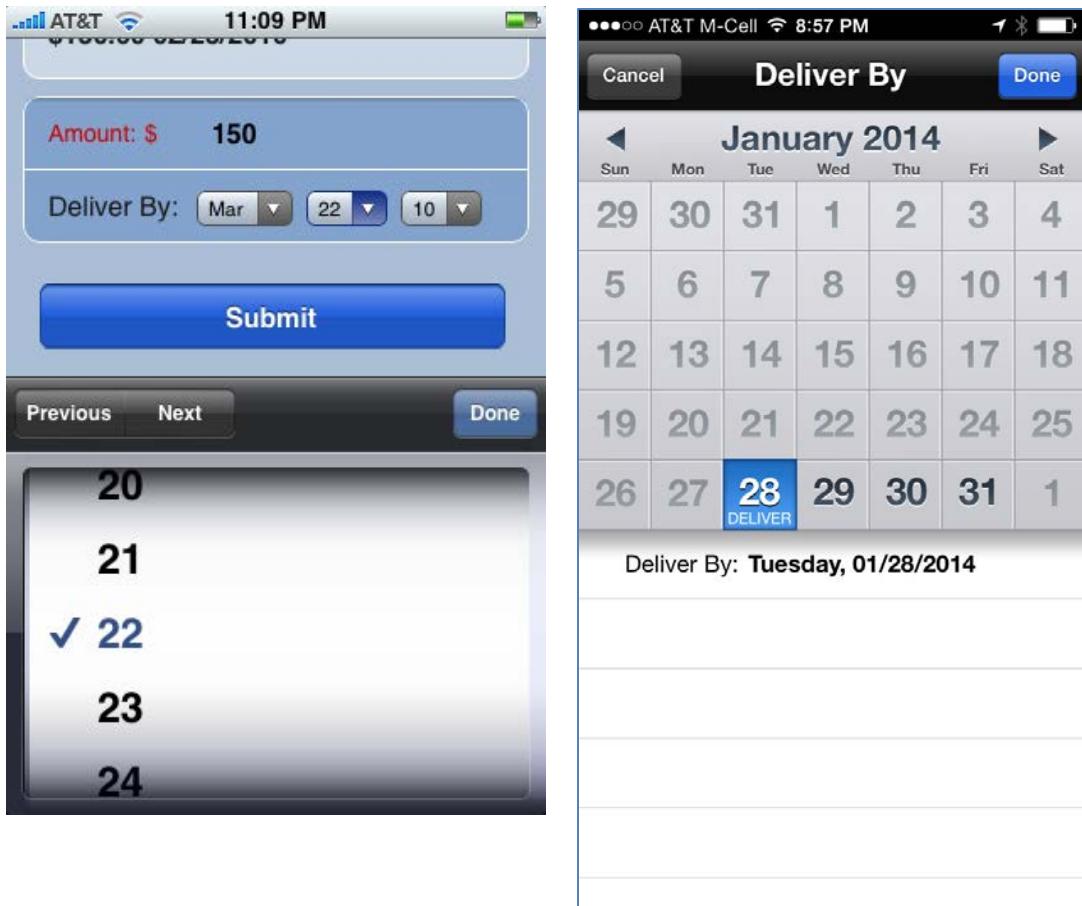
Sephora for iPhone autoformats the phone number; the user does not have to type any of the delimiter characters.

71. Do not make people memorize information from one page to another.

On a desktop, people usually have access to multiple windows; they can easily move back and forth between them and transfer information from one window into another. If they need to refer to a list of instructions while they fill in some complicated form, they can always keep the two windows next to each other and refer to one while working in the other.

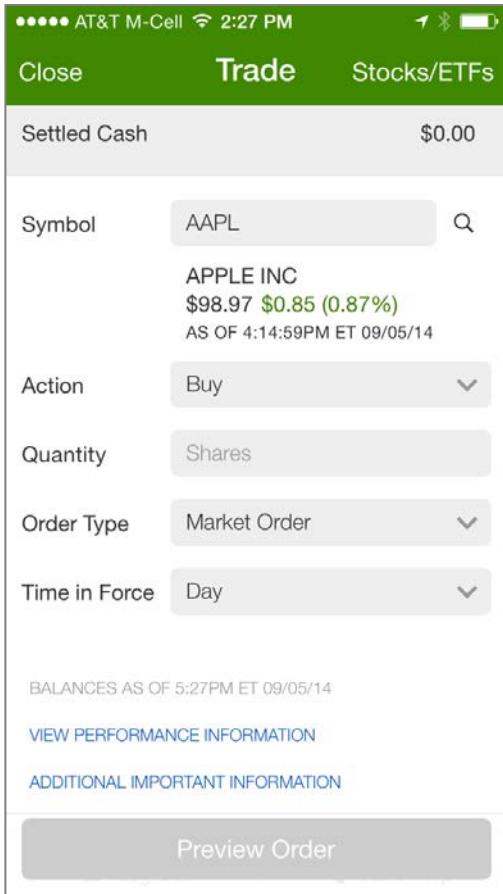
Unfortunately, as discussed in the section *Mobile Limitations and Strengths*, on mobile devices users only have one window available to them at any given time. Because of that reason, that window needs to be self-sufficient — users should not need to refer to other sources of information.

An earlier version of the Bank of America application asked users to select a business day for an electronic-bill delivery date, but it did not tell them which days were business days. People had to quit the app, find a different calendar app, then come back, log in again into their Bank of America account (because the app did not preserve login between sessions for security reasons), go through the schedule-a-payment process once more, and finally enter a good date (provided that they could still remember it). Luckily, Bank of America has fixed this problem in newer versions.



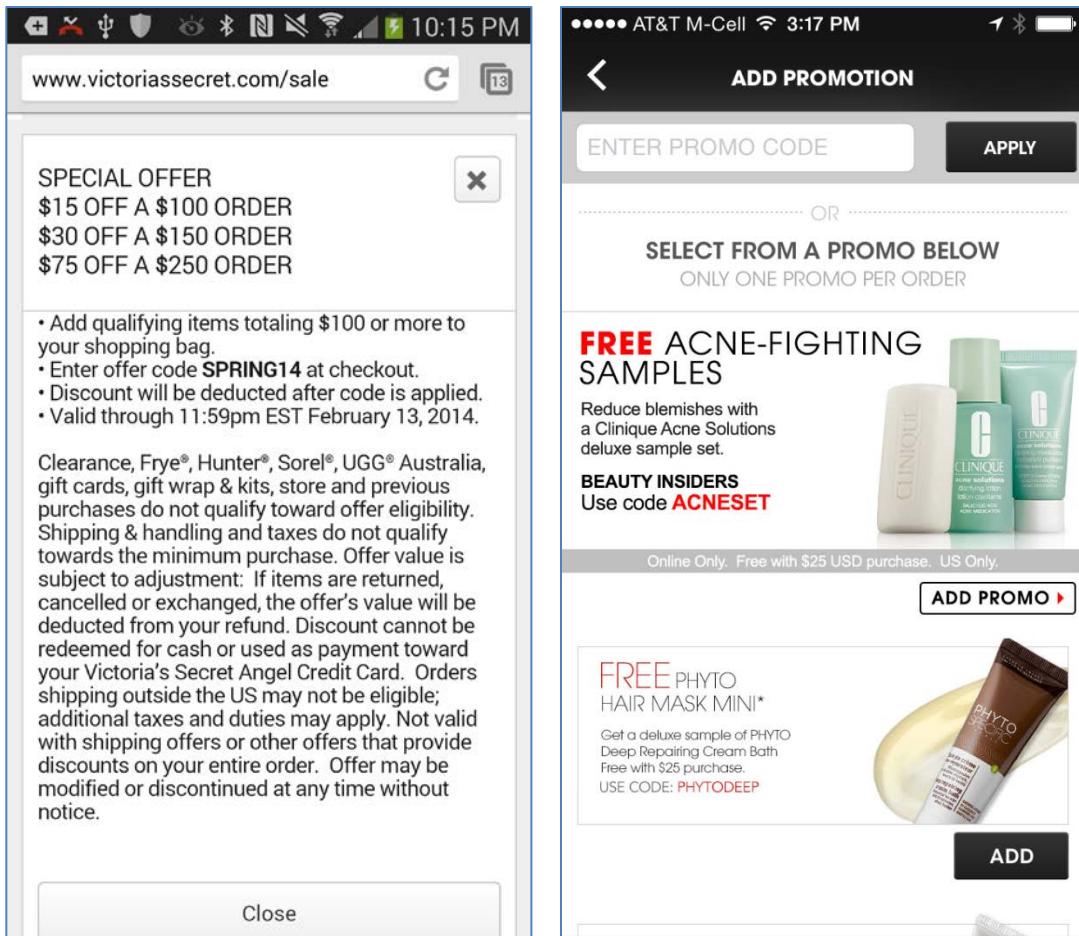
When setting up a bill payment in the Bank of America app, users were asked to enter a delivery date that was a business day. The older version of the app (left) did not tell users which days were business days. The newer version (right) corrected that problem by displaying a calendar where non-business days are not selectable.

One of our users was using a financial app to buy \$5000 worth of Apple share. The app requested the user to specify the number of shares that he wanted to buy and it displayed the price per share, but did not include a way to find out how many shares could be bought with \$5000. The user had to quit the app and start a calculator, figure out how many shares he could buy, keep the result in his short-term memory, and then return to his financial app to finish the transaction. It would have been a lot easier if he could just calculate the number of shares right there in the app.



The Fidelity app for iPhone only allowed users to specify how many shares they wanted to buy. One participant wanted to spend \$5000 on shares; he had to go to a different calculator app to figure out how many shares he could buy, and then come back and enter that number into the financial app.

We still see mobile websites offering coupon codes on one page, and then asking users to fill in the coupon code on a different page. In the best case, users would have to copy the coupon code and paste it into the appropriate field at checkout (assuming that cut-and-paste were supported); why not automatically populate the coupon code for the users or at least show it on the checkout page so they could select the one they wanted?

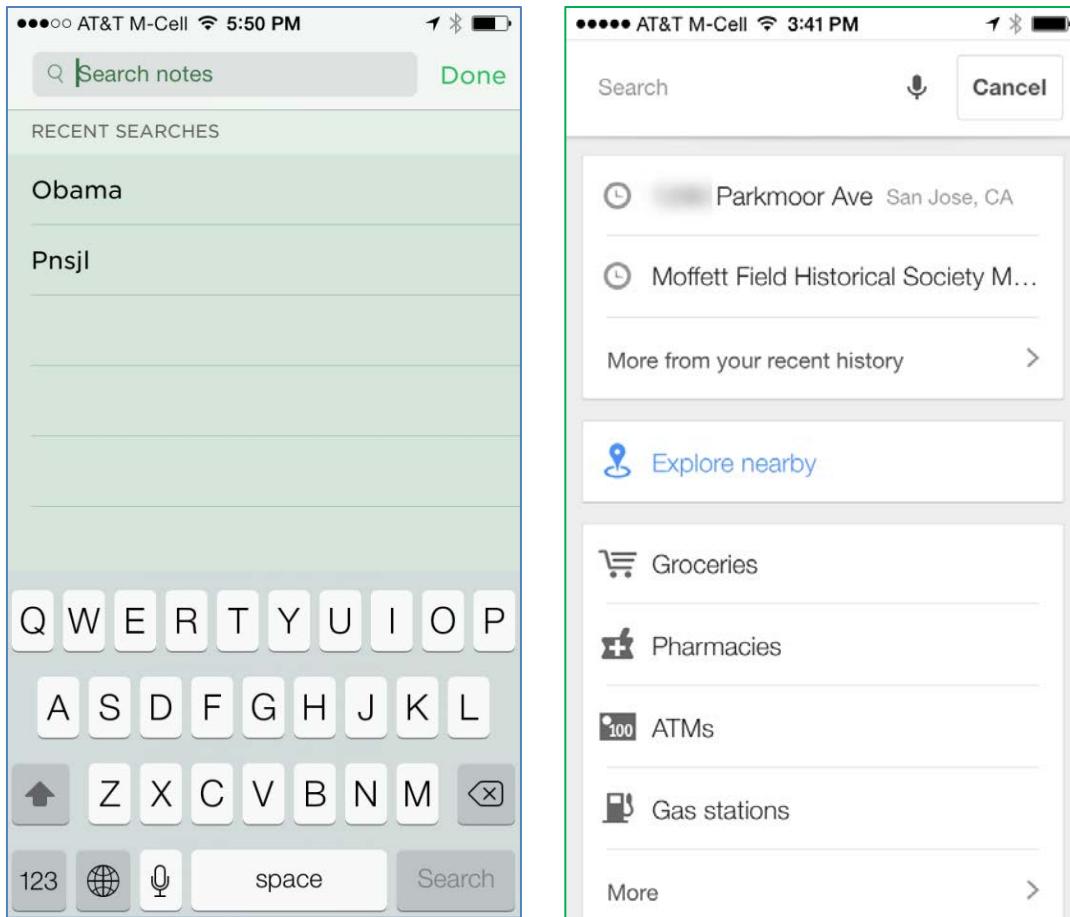


Coupons should be applied automatically: On Victoria Secret's site (left), users had to memorize the coupon code and enter it at checkout. In Sephora for iPhone (right), users could select a coupon from a list of offerings and have it automatically applied to their order.

72. Use personalization and history to provide good defaults and suggestions for text that needs to be input.

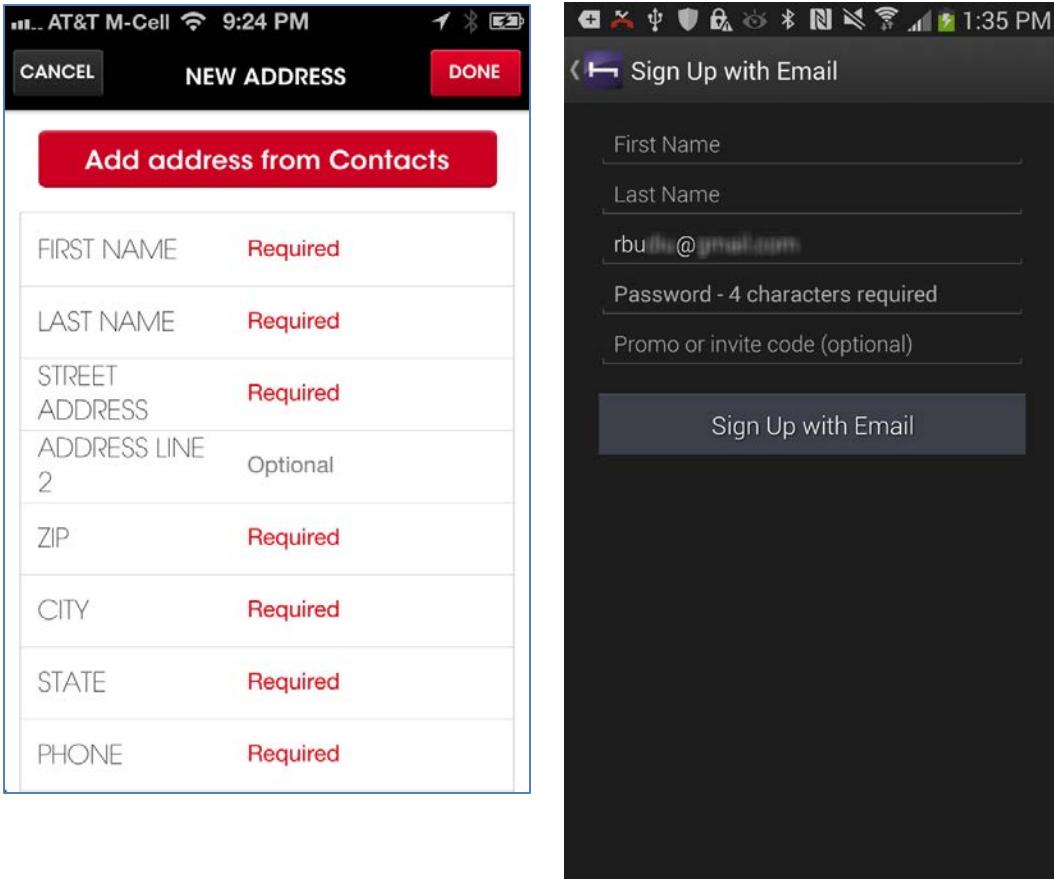
One way to save users some typing is to produce good defaults. Sometimes the defaults can be based on what the user has typed in the past (e.g., zip codes, names, addresses), or on other data that she has previously submitted.

The Time and Google Maps apps remembered recent searches and allowed users to easily perform them again.

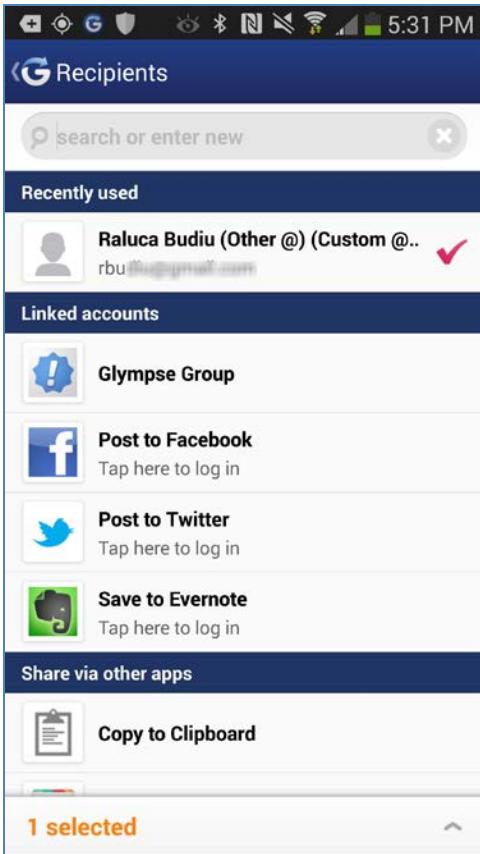


Access to recent history can reduce typing. Both Evernote for iPhone (left) and Google Maps for iPhone (right) remembered recent searches and made it easy for the users to input them again.

Besides using phone features and history, an app can use information already stored on the phone (e.g., in the contacts or Settings apps) to fill in field values. Sephora for iPhone allowed users to use their contacts list to select a shipping or billing address, while Hotel Tonight used the Google email address associated with the Android phone.



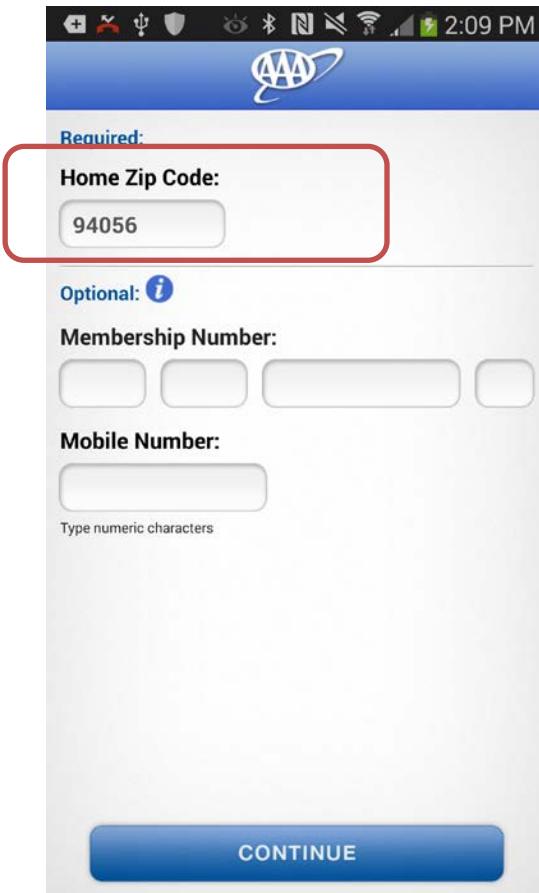
Information stored on the phone can be used to speed up the completion of forms on mobile: Sephora for iPhone (left) let users import an address from their contacts to fill in the shipping or billing address for an order. Hotel Tonight for Android (right) prepopulated the email field with the email address associated to the phone.



Glympse for Android (a location-sharing app): Users could choose with whom to share their location from a list of previously used recipients or from other channels associated with the user (e.g., Facebook account).

73. Allow users to easily delete default field values.

Because defaults are bound to be wrong occasionally, there should be a way to easily delete them. Users shouldn't have to manually erase the text field character by character, by clicking the delete key. Instead, they should be able to erase the defaults by pressing a single button that clears the entire field.



AAA for Android prepopulated the home zip code based on the current location, but did not let users erase it easily: they had to delete the characters one by one.

Sometimes textboxes are filled “by default” with a placeholder describing the content that should be typed in there. We recommend against placeholders for reasons described in guideline 63. If you must use a placeholder, the placeholder text should disappear when the user moves the cursor into the textbox to start typing.

A user was trying to plan a journey, and had to enter the start point and the destination:

“One thing I don’t like on this site, when I click this box, where it says ‘Address, intersection, or landmark’ — when I click on it, it doesn’t disappear and I have to delete it.”

74. Use QR codes to allow users to easily access websites and content on mobile.

We see more and more paper magazines, street advertisements, product tags, and desktop websites featuring QR codes. More and more users take advantage of them to quickly access a website.



QR codes like this one, featured in an ad in the Dwell magazine, help users find the apps more easily.

We definitely encourage QR codes as a way to deliver information about a topic or a product with little effort on users' part. Here's a quote from one of our participants:

"[I use QR codes] even in magazines — you know, a lot of magazines have it. Just shopping and seeing something ... if I want to get more information than what I can see on it, I just hit the website and go."

QR codes should lead to content that is relevant to the user and associated with the context in which the code is scanned. One of our coworkers scanned a QR code on a bread bag bought in a fancy restaurant, attracted by the promise of a recipe. Instead, she got a long video showing two cooks engaged in producing a completely different recipe that had no connection with the original bread product.

Phone Features Help with Input

Because typing is so error prone on touchscreens, it's a good idea to offer users an alternative whenever possible. Device features such as camera, GPS, or microphone are some of the strengths of mobile devices (see the section *Mobile Limitations and Strengths*) and can be effective input methods; use them whenever possible.

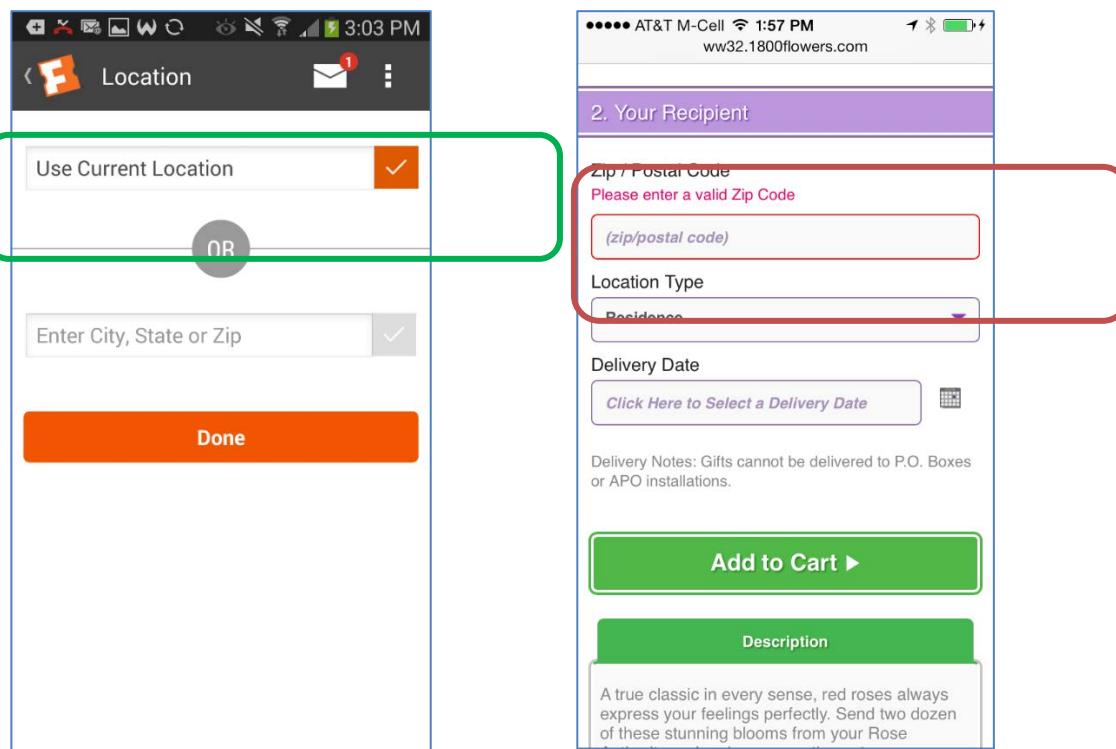
75. Use GPS information to detect location automatically and allow the use of current location.

Many times mobile users look for information around them. Since phones are equipped with GPS, most users expect that sites and apps compute location information for them. Plus, when people are traveling, they may not be able to come up with a zip code or even city name for their current location.

Even on the web, it is possible to use the GPS to detect the current location and save users the trouble of typing a city or a zip code. Applications have no excuse for requiring users to type in their location. (However, as explained in our guidelines about locators, users should be able to also enter a location different from their current one — see guideline 306).

One of our users was upset when he had to enter his zip code to find movie-theater locations:

"My phone has built-in GPS... I'm kind of mad that now I have to go and do another type, I have to click on this [textbox] and [type] 94555 and get show times... I know there's technology nowadays [...] [that will] locate where I am ... I don't know why they don't do that. I'd expect them to do that... it's like a feature, so I don't know, that's just weird to me, I guess..."



Users should be able to use GPS information instead of typing a location: Fandango for Android (left) made it easy for users to enter the current location by using the GPS. 1800 Flowers (right) asked users to type their zip code explicitly and did not give them the option to use their current location.

Users are grateful for any work that you can save them. One of our participants was using the Movies application on his Android phone, and, although his GPS was not working at the time, the application filled in his zip code based on past sessions:

"[The message says] 'Could not determine your location', but it did give me what it thinks it's the closest zip code and, automatically, without me having to type in the zip code, it brought in the movies for that zip code."

76. Allow users to edit the location provided by the GPS or other info captured via phone features.

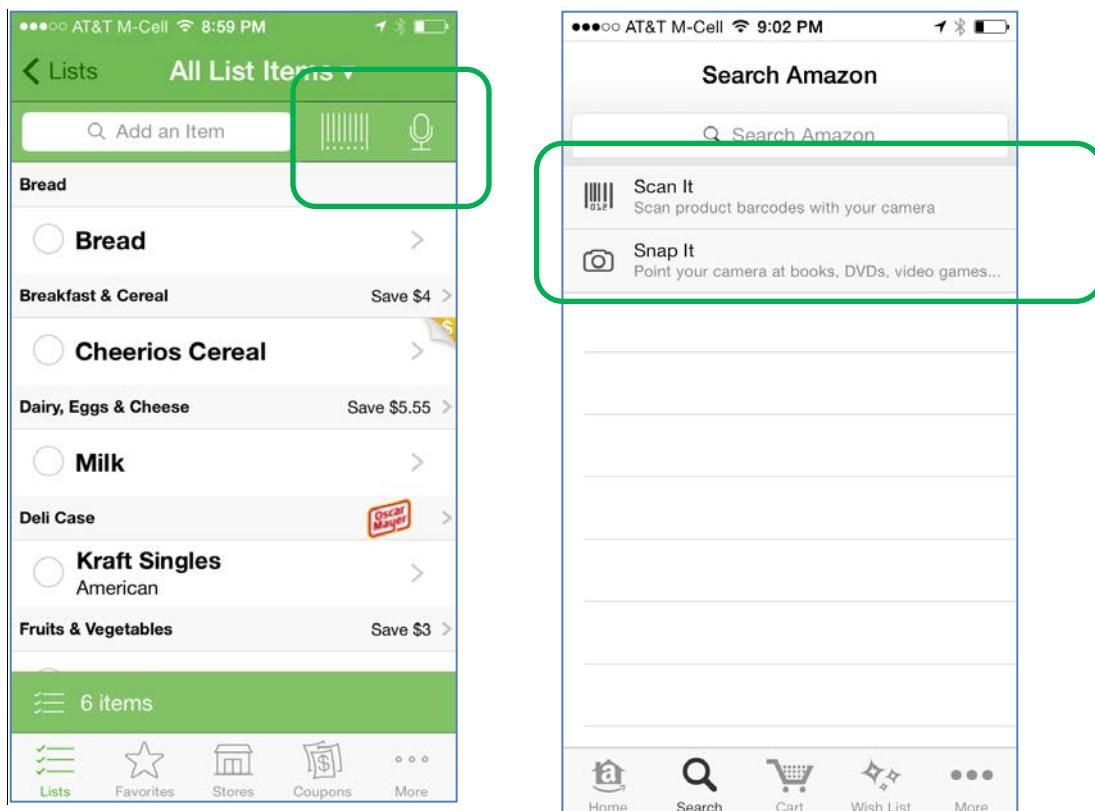
For tasks like finding movies or the current zip code, GPS accuracy will suffice. But for exact addresses, like ordering a taxi pickup, the GPS sometimes results in the neighbor's address or in the address across the street. For such exact tasks, the UI should as a minimum provide feedback with the address it derived from the GPS and allow users to edit it. Even better might be to provide one-click access to a list of addresses surrounding the GPS coordinates.



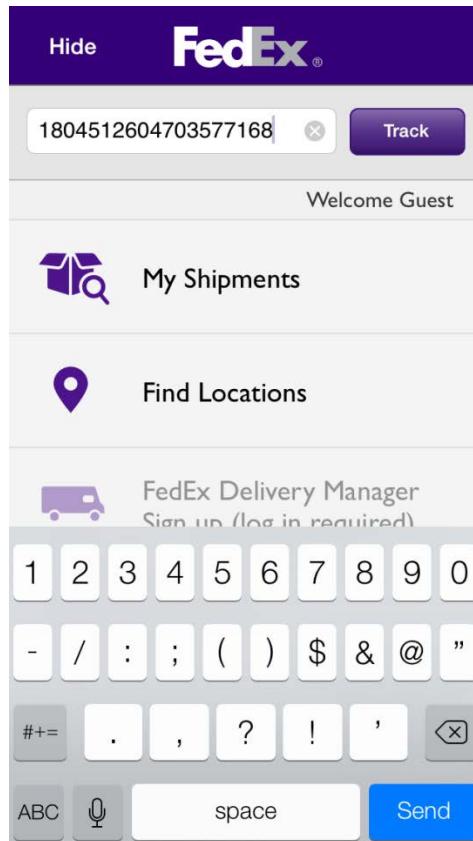
Uber for iPhone allowed users to edit the current location detected by the GPS or selected by touching a point on the map.

77. Use the camera to fill in information where possible.

Most phones come with other features that can be employed to speed up user input. The camera can easily be used to take a picture of an object or of a barcode. There is a plethora of apps that incorporate barcode scanners: some, such as RedLaser and Bar Code Scanner were dedicated to reading barcodes and searching prices for the corresponding products, but many other apps (e.g., Target, Amazon) included a search-by-barcode feature. And taking a picture of a credit card is a lot more convenient than typing in the 16 or more digits.

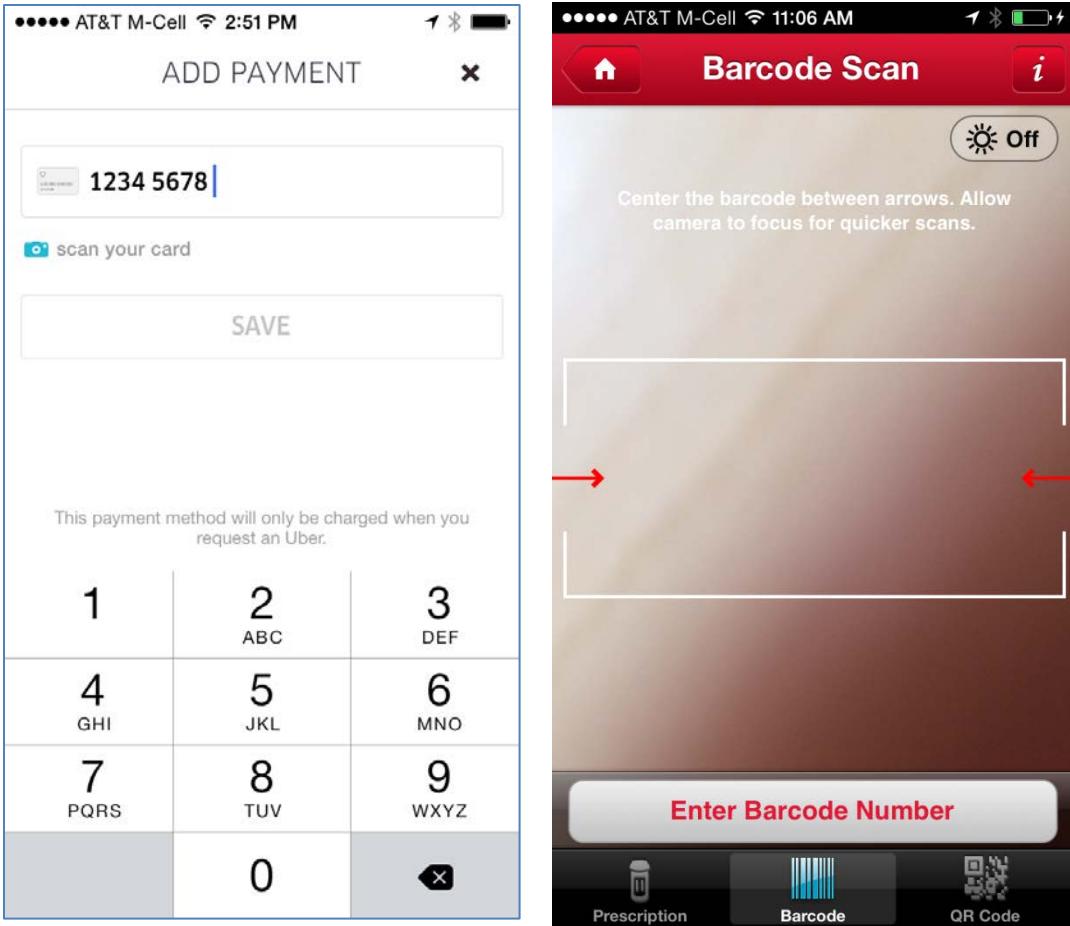


Apps such as Grocery IQ (left) and Amazon (right) used the built-in device camera to recognize barcodes. Amazon even allowed users to take a picture of an object and search for it. (Unfortunately, that feature was sometimes unreliable. A picture of a bracelet was recognized as an egg ring.)



FedEx required users to type the tracking number. A barcode-scanner feature would have made the task easier, at least for those situations when users had access to the mailing label (e.g., when they were sending a package).

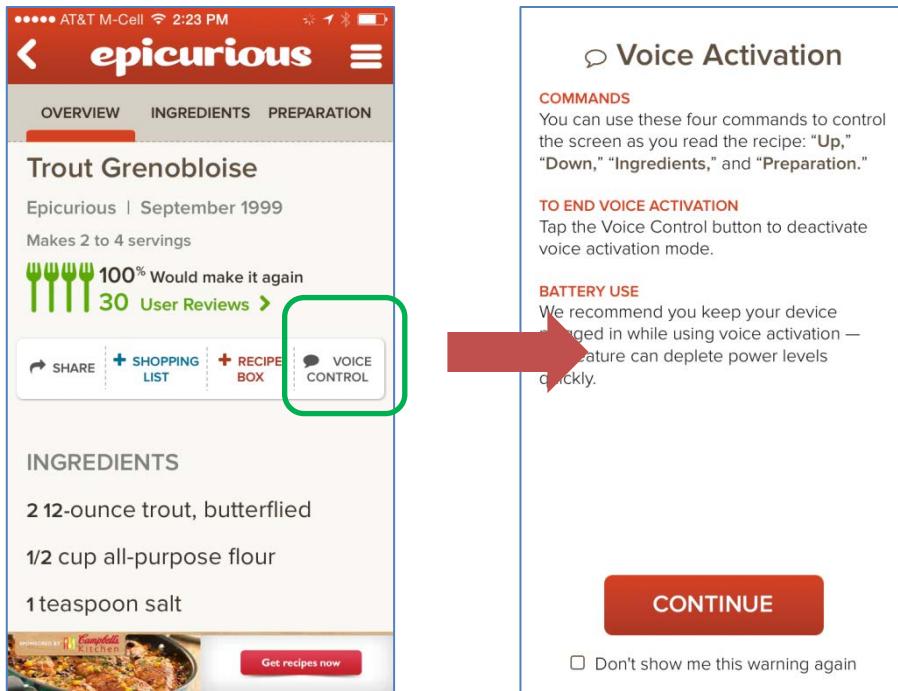
Sephora and Uber users could take a picture of their credit card and have details such as card number, name, or expiration date automatically filled in based on that picture. The Walgreens app used the camera to allow people to scan their prescriptions barcodes and easily refill prescriptions.



Taking a picture of a credit card or barcode is a better input alternative than typing the corresponding numbers: In Uber for iPhone (left), users could take a picture of a credit card and have it scanned automatically by the app. In the Walgreens app (right), users could refill prescriptions by taking a picture of the prescription barcode.

78. Use voice recognition to help users input information.

Voice recognition is well integrated in most mobile platforms: it's easy to say a word and have the phone recognize it. Standard keyboards include a microphone key. Many of our users rely on Siri or a similar voice-recognition system to initiate tasks in situations where touching the screen is cumbersome. And some apps even accept voice control. For example, in Epicurious (a cooking app), instead of having to use dirty hands to interact with the app, users could use their voice to go back and forth between different parts of a recipe.



Epicurious for iPhone allowed users to use their voice to interact with a recipe.

Even if users may not be able to use voice commands in all circumstances, voice input can be significantly faster than typing; apps and websites should take advantage of it whenever possible.

SPINNERS, SLIDERS, AND OTHER SELECTORS

79. Do not use sliders for inputting precise values. Only use sliders when the exact value is not important.

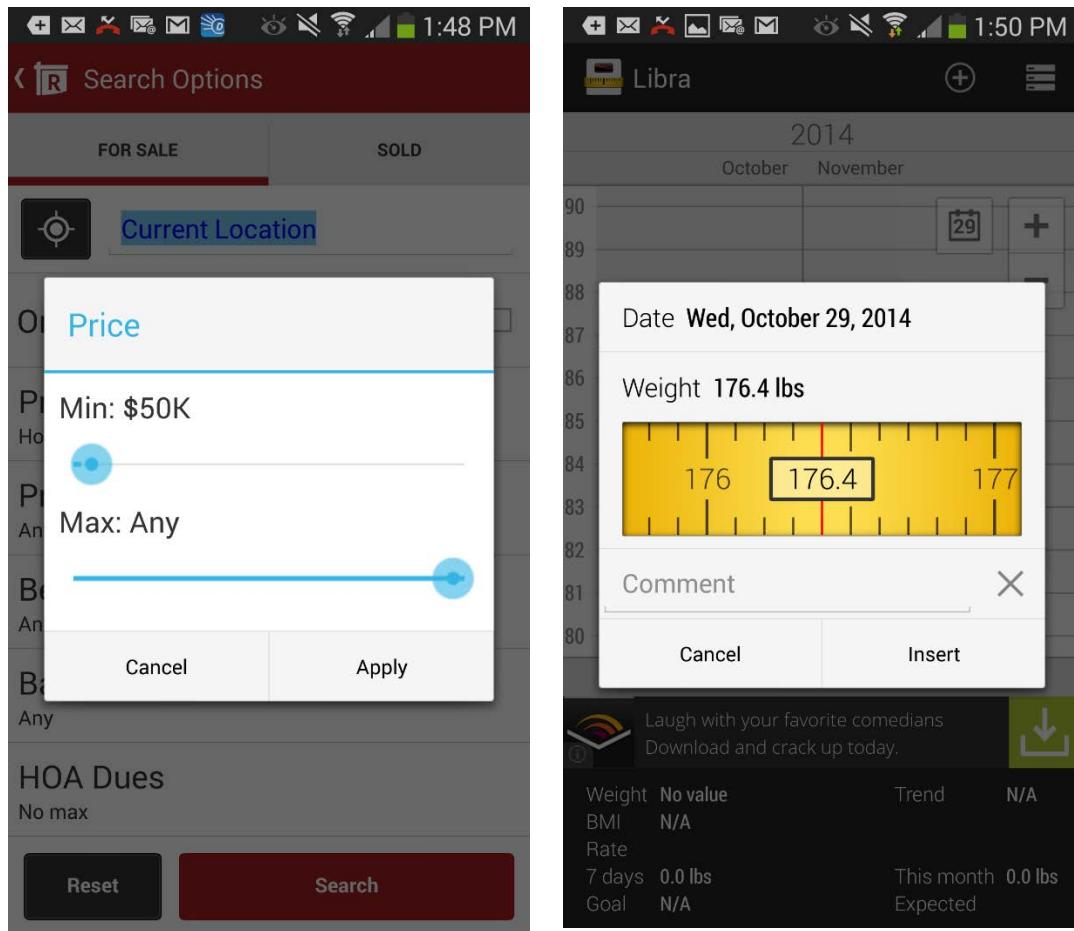
Sliders are good in situations where an exact value is not important, but they are hard to use when you need to indicate a precise value: users have to move back and forth to make sure they get the exact value that they need. They're also not recommended if the range of values is very big: (a) because the app will often need to compress the slider (as in the case of Redfin, below) to span the entire range, making it harder for users to hit an exact value; and (b) because, even if the app doesn't compress the range (as is the case with the Libra app below), there needs to be a lot of scrolling to reach the right value on the slider.

Redfin asked users to enter a price range for a home by using sliders. House prices are definitely a domain where the exact value does matter: there's a difference between looking for a house of \$1 million versus one of \$900k. It would have been more appropriate to just ask users to enter their price, instead of having them fiddle with the slider.

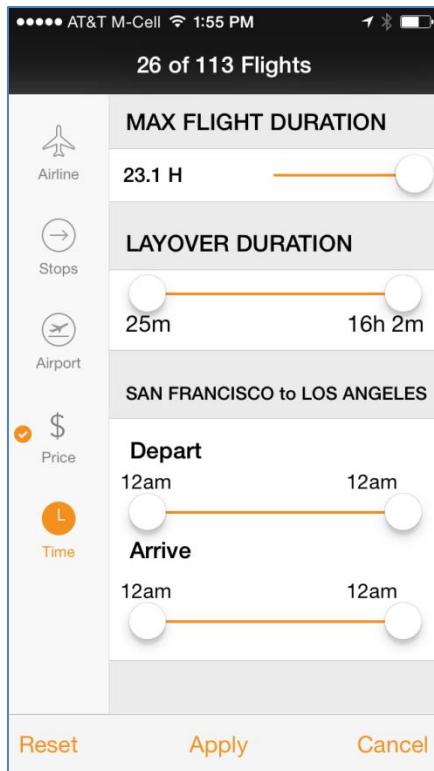
Libra allowed users to track their weight; in order to enter their weight on any given day, they needed to move a slider that showed increments of 0.1

pounds. By default, the slider was positioned at 176 lbs. Imagine how much one would have needed to slide if their weight had been 130lbs or 200lbs!

In contrast, Kayak appropriately used a slider for entering flight departure times in a flight search engine: most users don't care for the exact time, but for a range (late morning, early evening, etc.). Note, however, that sliders are not always appropriate for selecting time: if the context were entering an appointment in a calendar app, a slider would not be right (because there the exact time to the minute is precisely what matters).



Inappropriate use of sliders: Redfin for Android (left) used a slider to select house prices in a real estate app. Libra for Android (right) had users enter weight using a slider.



Appropriate use of sliders: In Kayak for iPhone the flight departure time was selected with a slider; the departure time doesn't need to be precisely indicated when users search for a flight.

80. Do not use spin wheels for items that don't have a natural order.

In a spin wheel, values in a list are projected on a circle: the first element follows the last one. For digits from 0-9, users can easily see that they've got through all the options if they reach small numbers after big numbers. In other words, there's a natural order that helps them figure out with practically zero effort that they've exhausted all the values in a list. However, if, instead of digits, the values are "previous orders", "most popular", "pizza", "pasta", "wings", "sides and desserts", "drinks", "specials", users must remember them in order to check whether they've seen them before and have gotten to the end of the list.

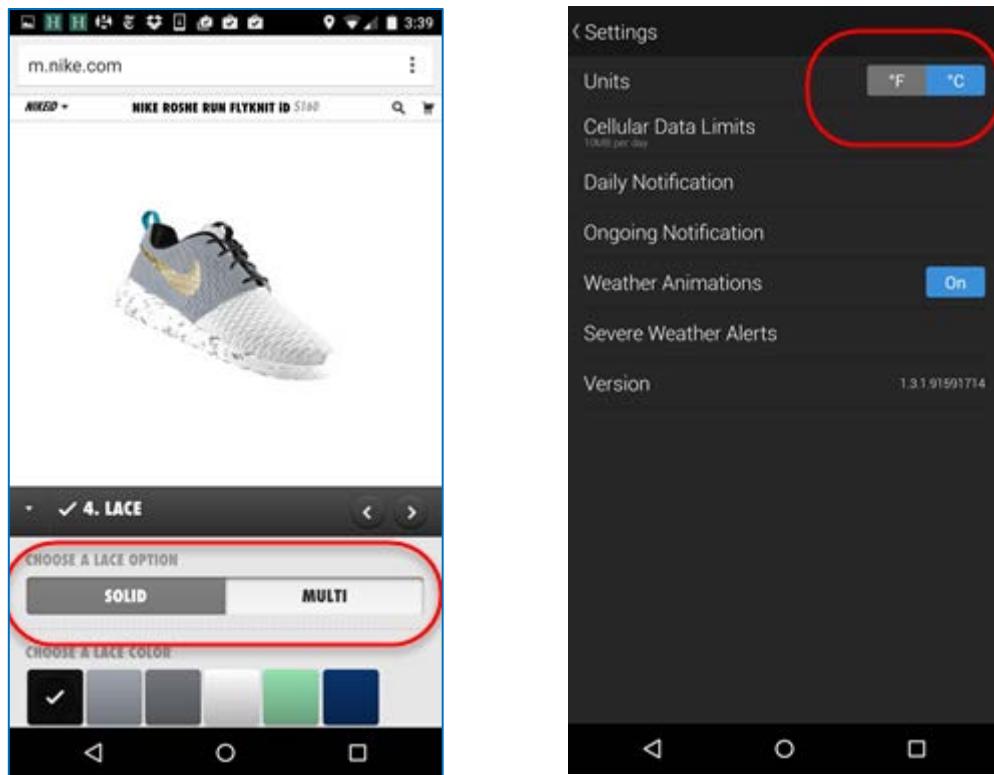
It's a lot easier to use a tabular view for lists like that, because people don't have to memorize the beginning and the end of the list: the end of the page is the same as the end of the list.



An older version of Pizza Hut for iPhone used a spin wheel for different menu choices. Users had to remember which values they had seen before.

81. Do not use color to communicate the active choice.

A toggles, switch, checkbox, or radio button can all be successfully used to indicate a choice out of two alternatives. However, their visual design can impact how usable they are. A popular design trend is to use color-based toggles to indicate selection. Unfortunately, color is an unreliable indicator for several reasons: (1) 6–8% of men are color blind and may not distinguish between different colors; (2) variable light conditions (a typical problem with smartphones) may make the color look different in different circumstances, and may impoverish the contrast between the on and off colors; (3) color meaning for selection is not used consistently on the web — for instance, does the lighter/brighter color mean on (selected) or off (unselected)? The answer to that question depends on the site.

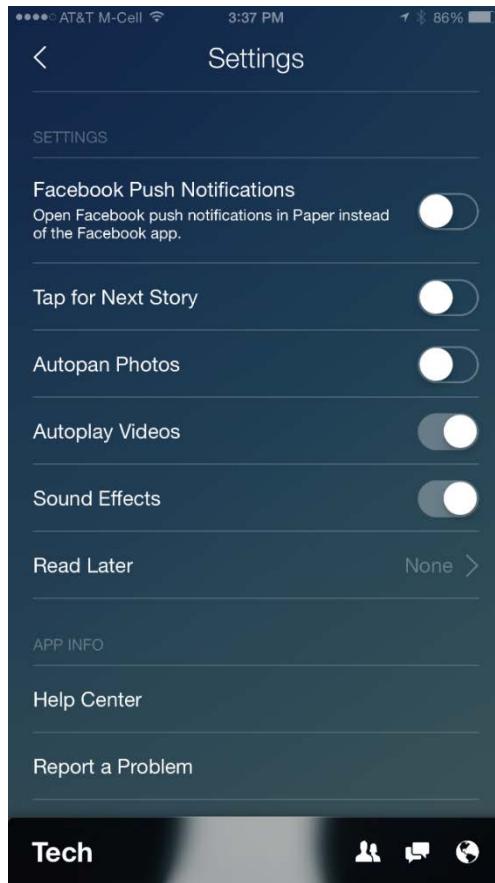


On Nike.com (left), users complained that it was hard to figure out whether the default selection was *Solid* or *Multi*. In this case, the context could be used to figure out the answer, but it was more work than necessary. In Yahoo! Weather for Android (right) the color blue was used to indicate the active choice of temperature unit; this visual representation is ambiguous. Note that Nike used the darker color to indicate the selected option, while Yahoo! Weather used the brighter one.

Instead, a checkbox or a pair of radio buttons would work better in many of these cases. Note that the problem is significant when there are two choices, as when there are more than two, users can usually infer from context which of the choices has been selected. For instance, in the Pages app below, the selected styling was easy to recognize because it was the only one (among the 3 possible choices) that was visually different.



Pages for iPhone also used color to indicate selection; however, because there were more than two options it was easy to determine which was selected. For example, it was clear that the bold font B was selected because all the other options looked the same, and B was the only one that looked different.



Paper for iPhone: Which of these settings were on and which were off?

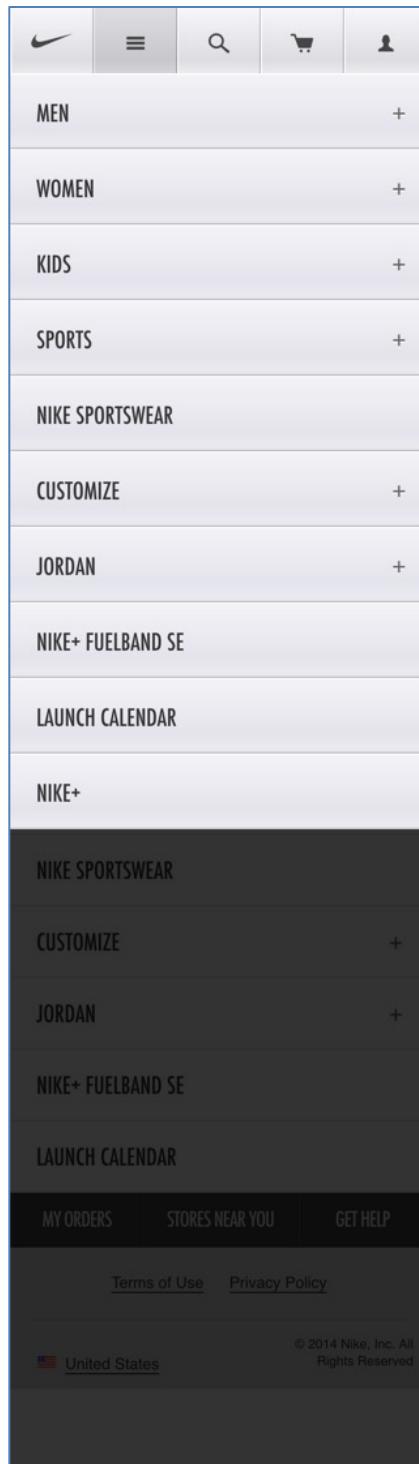
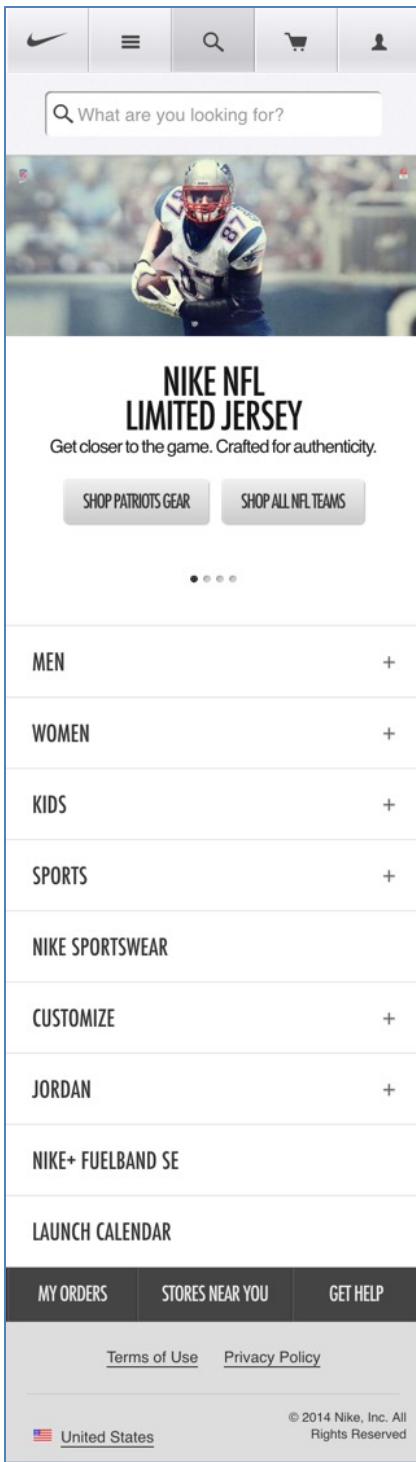
MENUS: DROP-DOWNS, ACCORDIONS, OVERLAYS, SLIDING MENUS, PIE MENUS

All these types of elements are highly popular on mobile, for good reason: they increase the information density on the screen by hiding content that is nonessential yet might be necessary to the users.

They all suffer from the same big problem: out of sight is out of mind. In other words, what is hidden under a menu has a higher chance of being ignored by users. In the section *Navigation* we discuss these types of elements in the context of navigation design; here we focus on their advantages and disadvantages regardless of the function that they serve.

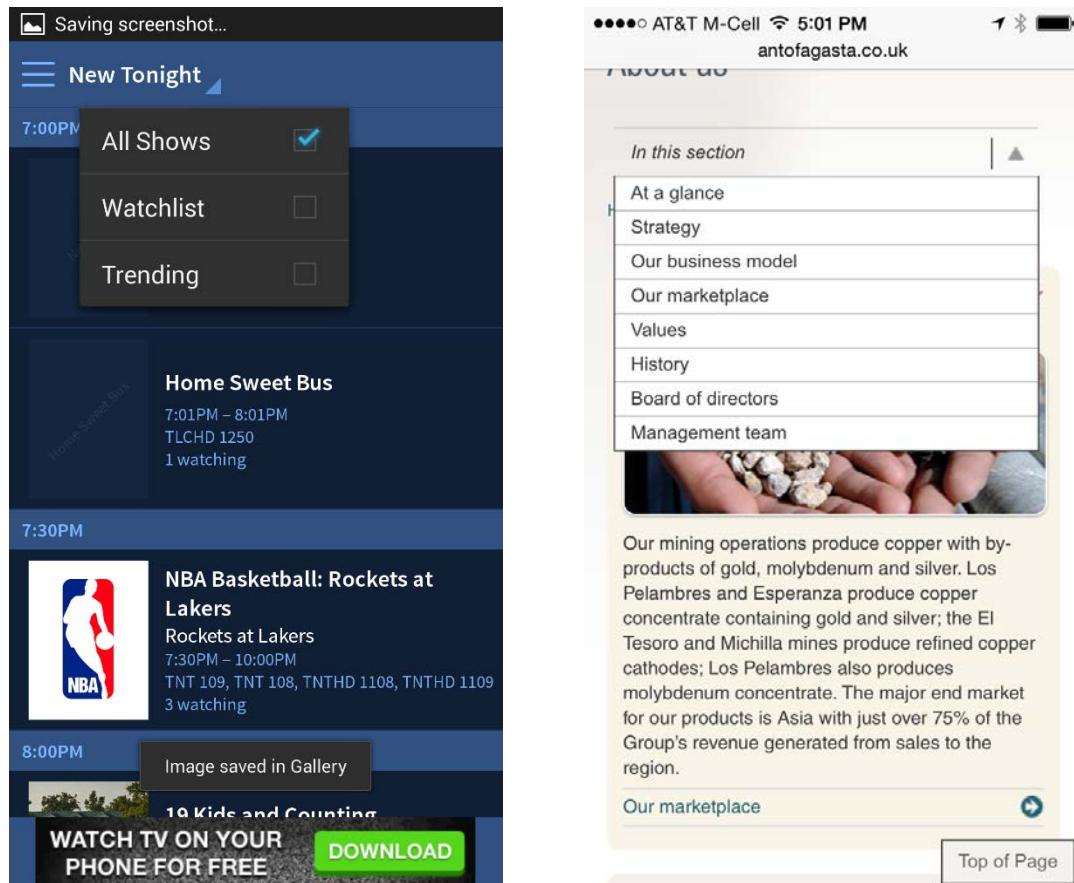
Drop-downs are often encountered in forms, when users have to choose from a set of alternatives hidden under the drop-down.

Overlays appear in place on top of the background content and cover some of that. They can be **modal** or **nonmodal**, depending on whether users can still interact with the background (if they can, then the overlay is nonmodal).



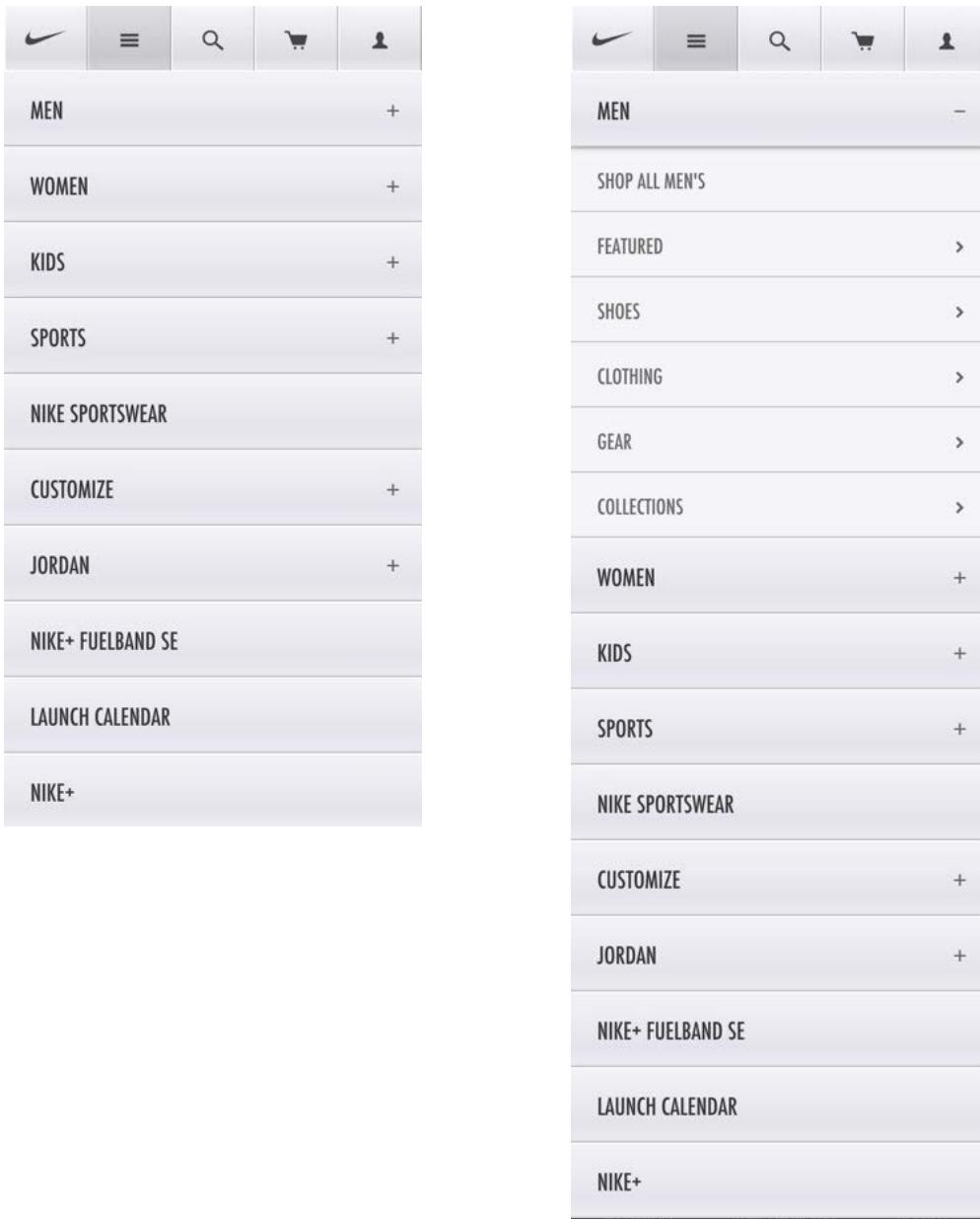
Nike.com: The main-navigation menu was implemented with an overlay (right) that appears on top of the homepage content (left).

Overlays themselves can be modal (when the user cannot interact with any element on the page before dismissing the overlay —like in the Nike example above) or nonmodal.



Modal and nonmodal overlays: TV Guide for Android (left) used a modal dialog: users couldn't interact with the background page before the dismissal of the overlay. Antofagasta.co.uk (right) used a nonmodal overlay: users could scroll down the page or even select a link from the page (e.g., *Our marketplace*) with the menu open.

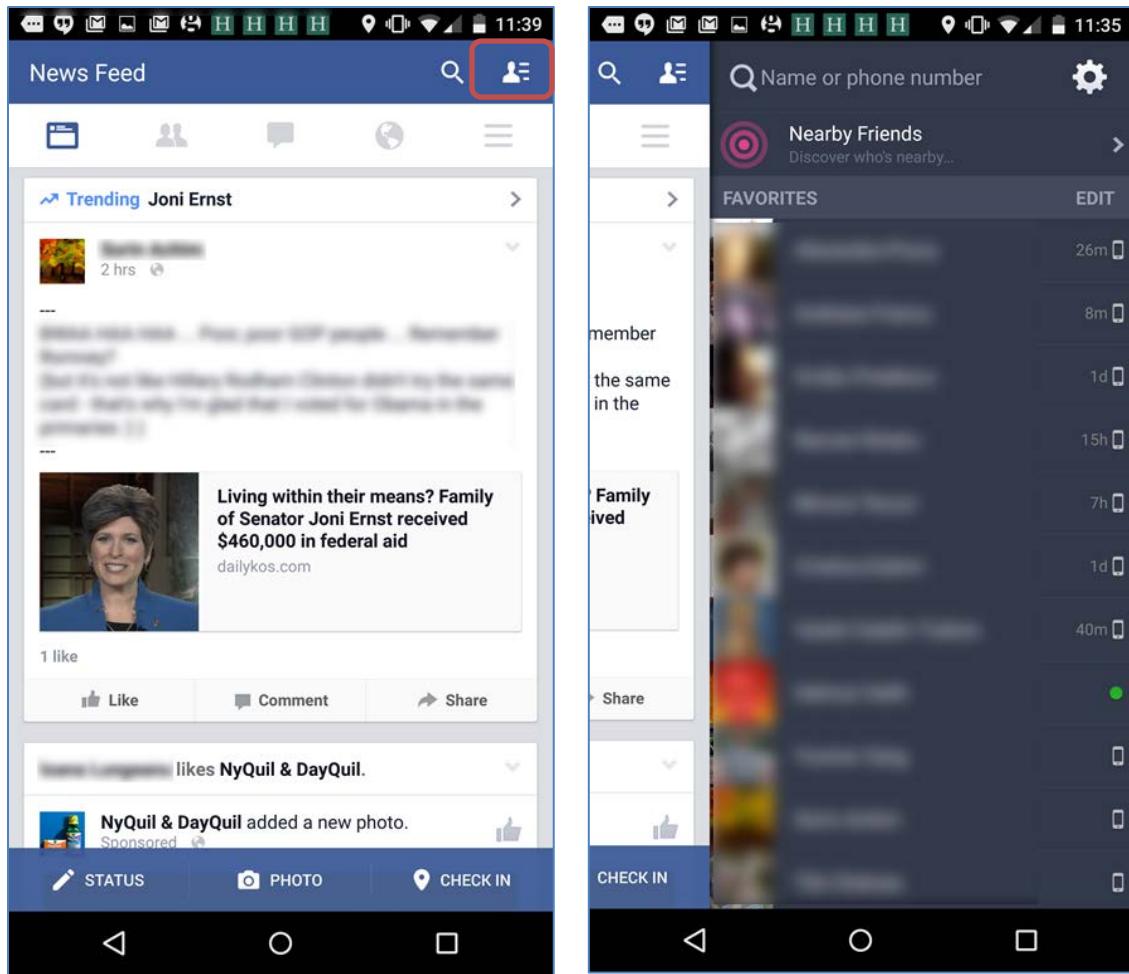
Accordions also appear in place, but instead of being superposed on top of page content, they push the page content down.



Nike.com: Each of the categories in the main-navigation menu (left) was implemented as an accordion that expands the secondary navigation (right). The main-navigation options were pushed down the page.

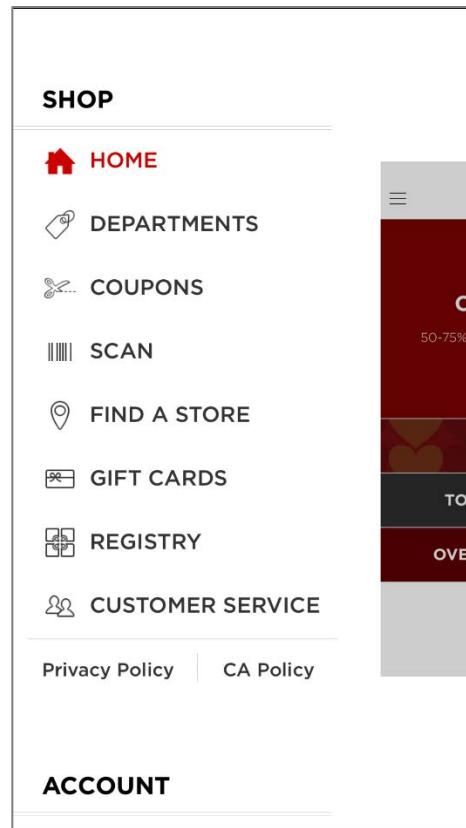
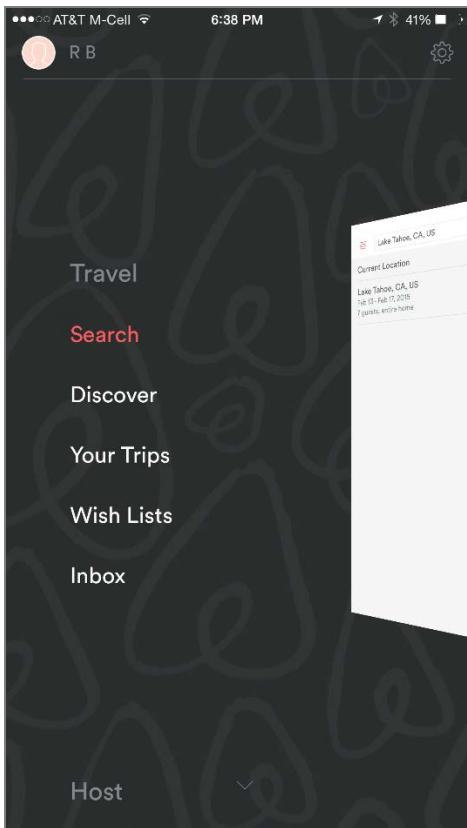
Overlays and accordions can also be used for selection, as well as for hiding additional informative content that may not be needed by everybody.

Sliding menus are similar to accordions, but instead of pushing the content down, they push it laterally (either left or right). This menu was originally made popular by Facebook, which has since moved back to the tab bar for the main navigation.



Facebook for Android used a sliding menu for the contact list.

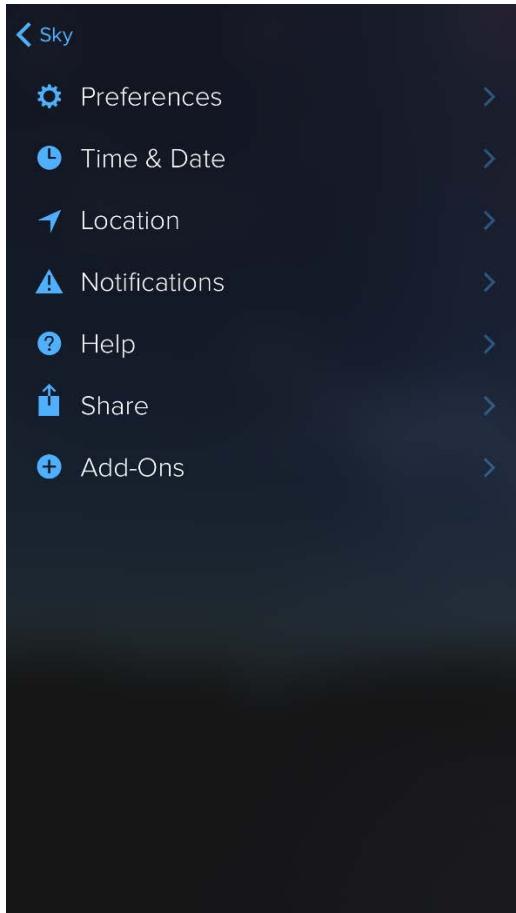
The thumbnail sliding menu is a variation of the sliding menu that has become popular more recently. It shows a thumbnail of the main page on the side of the menu. Sometimes (like in the JC Penney's example below) the thumbnail changes depending on the section selected in the menu.



Airbnb (left) and JC Penney's (right) for iPhone used a thumbnail sliding menu, in which the main page was shown as a thumbnail to the right of the menu. The JC Penney's app changed the thumbnail as users scrolled down to see another section of the menu (e.g., if they scrolled to *Account* the thumbnail would be that of the account page).

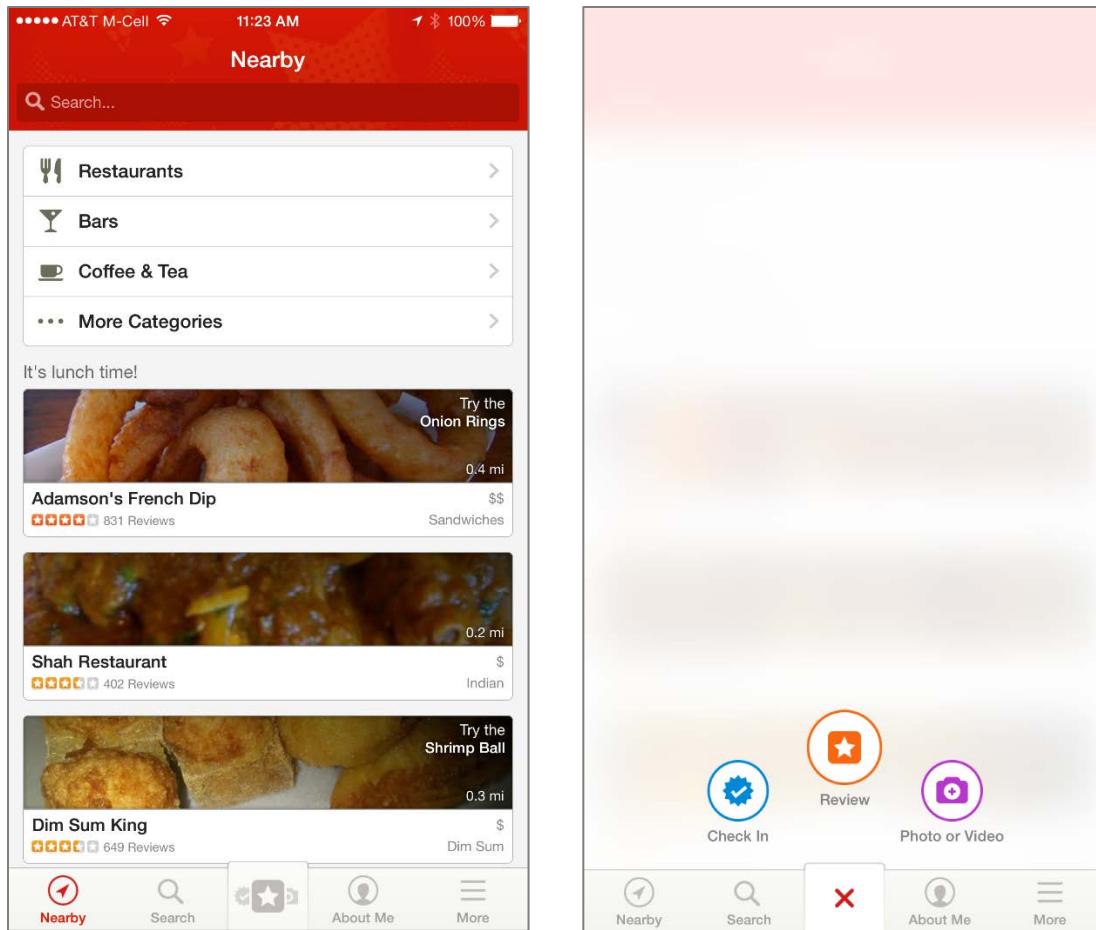
Like overlays, sliding menus can be modal or nonmodal, depending on whether the user can still interact with the main-page content. The thumbnail sliding menus are modal usually, as the user can go back to the main page but not interact with the content of the page.

Separate-page menus take the user to a new page where they display only the menu options, with none of the original page content.



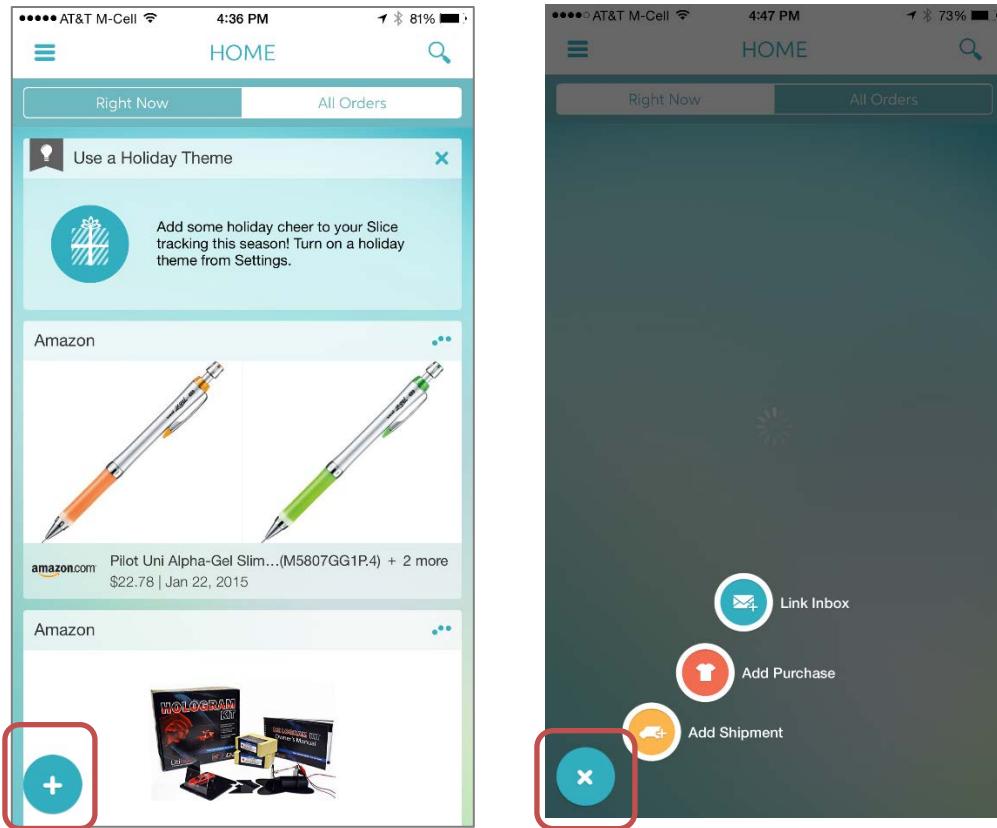
Sky Guide for iPhone: The hamburger menu (left screenshot) is an example of separate-page menu as it expanded on a new page.

Pie menus appear in place around the icon that triggered them. They are usually used with icons instead of text only and have a nice advantage: they minimize the time to reach any of the options inside the menu. However, they are still quite unfamiliar to most users, which may annihilate this aforementioned advantage.



Yelp for iPhone: The icon in the middle of the tab bar expanded into a pie menu with 3 options.

Note also that pie menus must have their options placed **on a circle** centered in the menu icon (to make sure that all options are equally distant to that icon)—otherwise, they have no reaching-time advantage over regular menus. Thus, a menu like the one used in the Slice app for iPhone is not a real pie menu.



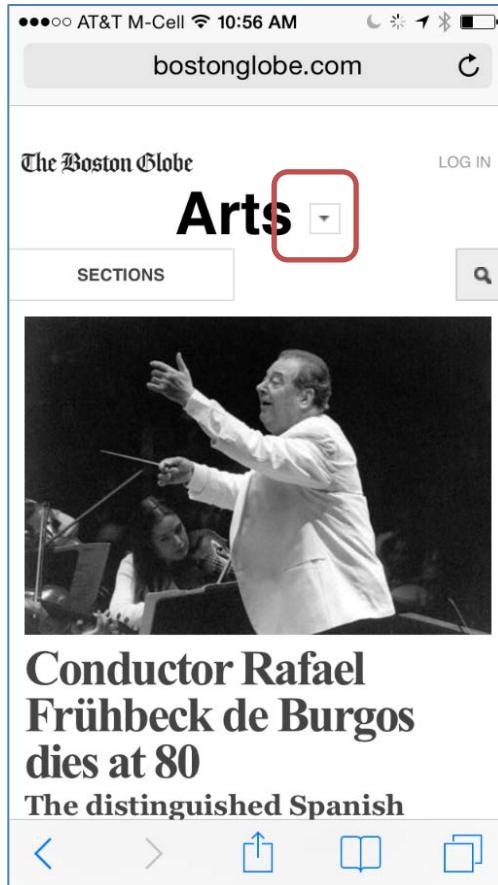
Slice for iPhone: The plus expanded into a menu that did not benefit from the advantages of the pie menu, because the options were not equidistant from the menu icon.

82. Make sure that your menus look expandable.

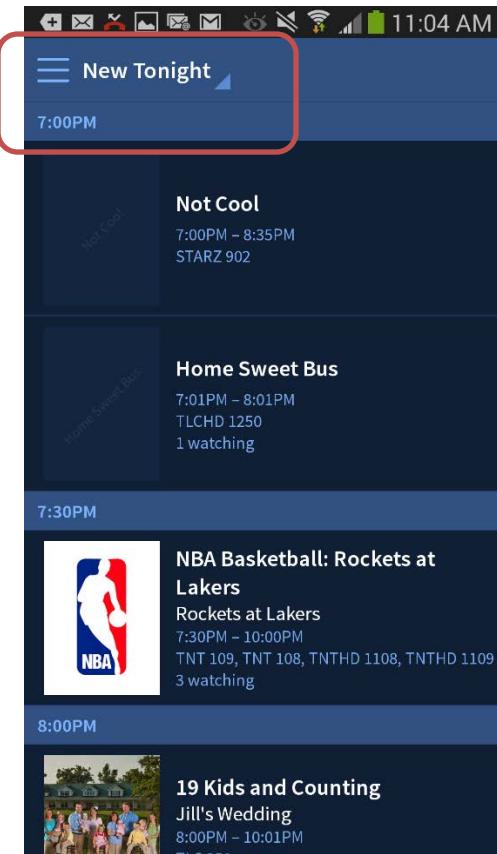
Using drop-down menus and other expandable menu formats is especially attractive on mobile devices because they allow some space saving: a group of links is compressed under one button. Unfortunately, users frequently ignore menus, and, hence, the links hidden under them. The main reason is that menus are not salient enough — often, they don't look like something may be hidden under them, and users don't think to press them.

The standard way to signal all these elements is to have an arrow at the right of a label. Sometimes the arrow may be too light, or have poor contrast. In other situations, the arrow is too small. Remember that anything that is hard to perceive visually is unlikely to be noticed or used.

The arrow hiding the secondary navigation on the mobile version of Boston Globe (a responsive site) was so tiny that none of our users noticed it.



Bostonglobe.com: The arrow next to the section title (*Arts*) was never noticed by our participants who struggled to find secondary-navigation options.



TV Guide for Android: The little arrow next to *New Tonight* is fairly standard for Android, yet users still fail to notice it. Go for higher contrast between the arrow and the background color to make it more noticeable.

83. Use large targets for arrows associated with drop-downs or other menus.

Not only do the arrows next to menus have to be visible, but they must also be easy to touch. As any other touch targets, drop-downs target area should be approximately 1cm x 1cm (see guideline 37).

Note that sometimes people insist on tapping the little arrow next to a drop-down instead of the whole drop-down area. Making the arrow larger can speed up the interaction.



Wikipedia.org: The drop-down for changing the language had too small a target area. Moreover, the language choice should not have been implemented with a drop-down (since there were a lot more than 6 languages supported — see guideline 87). (Note also that the search box was too short and forced users to scroll inline — see guideline 60, and that the label for the language menu was improperly set to one of the alternatives —see guideline 85.)

84. Use menu labels with good information scent.

85. Do not label a menu with one of the options inside it.

Not only should you signal that a drop-down or another type of menu can be expanded, but you should also choose a label that has good information scent — that is, it describes all the options inside the menu. Don't label the menu with one of the options that it offers: people might not realize that there are other related alternatives available to them.

For instance, on ESPN's website, a menu was used to switch between different kinds of results (women's, men's, singles, etc.) in a tennis tournament (see figure below). The menu was not salient (it blended with the background and looked more like a title than like a menu — the arrow next to it was really small

and barely visible). The menu label (*Men's Singles*) did not suggest that there were alternative options to explore.

The image shows two side-by-side screenshots of the ESPN.com mobile website. The left screenshot is from an older version of the site, displaying the '2011 ATP Winston-Salem Open' results. It features a prominent 'Men's Singles' dropdown menu at the top, which is highlighted with a red box. Below it, a date navigation bar shows 'AUG 24', 'AUG 25', and 'AUG 26'. The right screenshot is from a newer version of the site, showing the 'ATP BNP Paribas Masters' results. It also has a 'Men's Singles' dropdown menu at the top, which is also highlighted with a red box. Below it, a date navigation bar shows 'OCT 27', 'Tue OCT 28', 'Wed OCT 29', and '>'. Both screenshots show detailed tournament results for men's singles matches.

ESPN.com: An older version of the website (left) used menus to allow people to switch between women's and men's results in a tennis tournament. Unfortunately, users usually ignored the menu. A newer version (right) preserved the same poor menu label, but increased the contrast between the arrow and the background.

When we tested this website, users had a hard time finding a way to get to the women's singles results, since they were hidden under a menu labeled *Men's Singles*:

"It's the men's singles. I would assume, since they don't have the women's singles up here, that they're not even playing today [...] We'll go with [reading link names] *Daily Results*, *Men's Singles*, *Results*. I think only men played today, and I'm trying to back into — okay, today is the 27th, let's go back to the 26th [after clicking on link]. They still do men's coverage only here ... Ok, let's see if we can change this to women's coverage, [finds menu] ok, *Women's Singles*, ok, go to *Women's Singles*, ok."

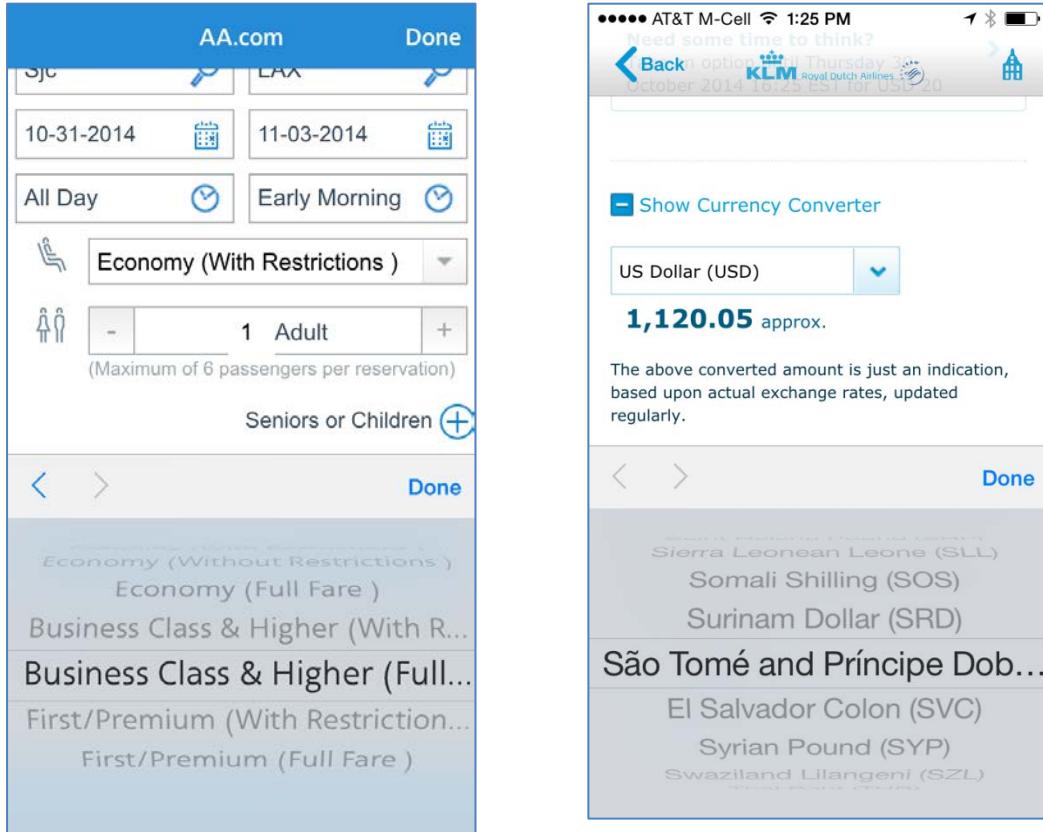
Android menus often suffered from low discoverability, because many apps used an Android-specific pattern that labels the drop-down with one of the options and displays a tiny arrow next to the menu label.



Weather Channel for Android: Some users never figured out that the label *Milpitas, CA* was actually an overlay menu that allowed them to quickly switch between previously set locations. Other took a long time to discover this feature. (The small, far-away arrow was another cause of the low discoverability of this menu.)

86. Make sure that all the text of each option in a menu is visible.

Another problem that occasionally appears with menus is that options that are too long get truncated, and users have no idea what they're choosing, like in the two examples below. In such situations it would have been better to use another type of display that allowed long names to wrap on another line.



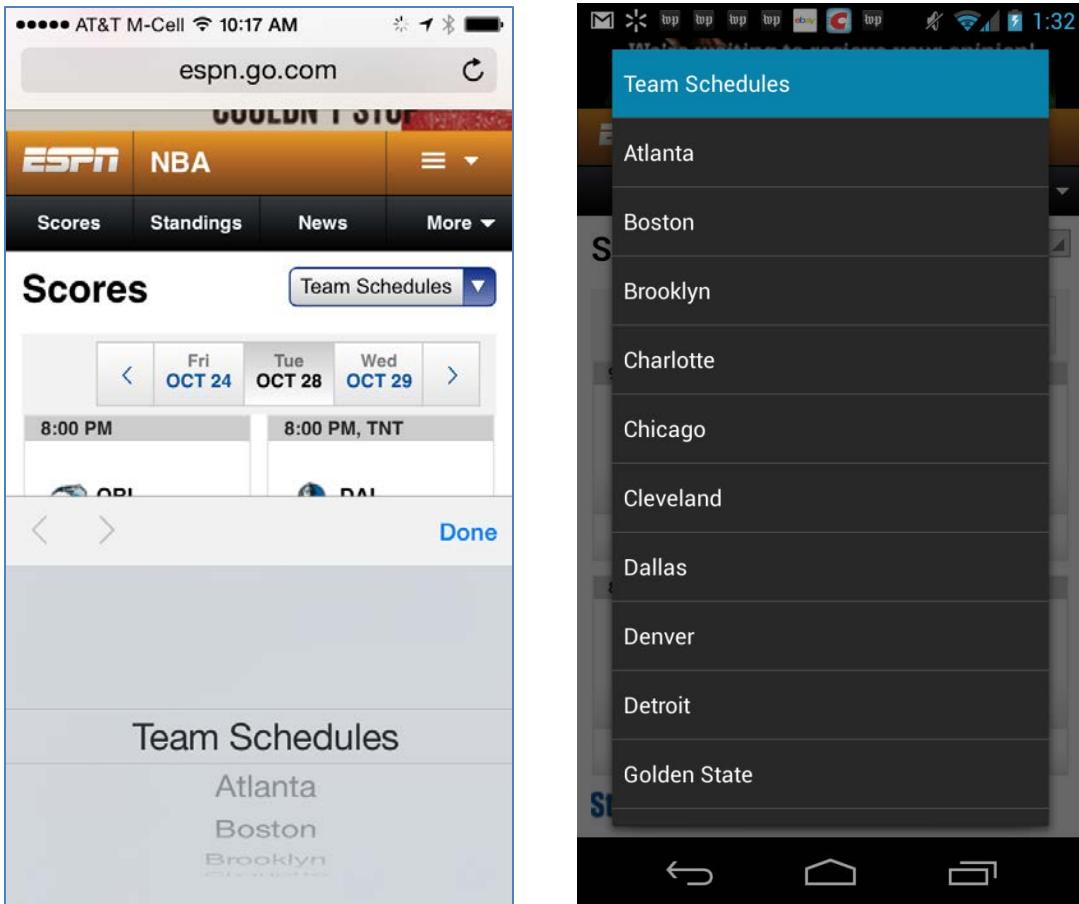
Option names should always fit in the drop-down or popover. AA for iPhone (left): Types of tickets were too long and had to be truncated in the drop-down. KLM for iPhone (right): Currency names were too long for the drop-down.

Drop-Downs

87. **[iOS] On iOS use drop-downs only for short lists of 4-6 items.**
88. **[Android] Minimize scrolling in a drop-down box by using the whole screen to display long lists of options.**

Unlike on Android and Windows Phone, on iOS drop-downs take only half the screen space. Scrolling through screens and screens of information in a tiny window is painful already; doing it in a window half the screen size is twice as painful.

If you have a long list (for instance, US states, countries, years), instead of using a drop-down, consider displaying it in-situ or on a separate page.

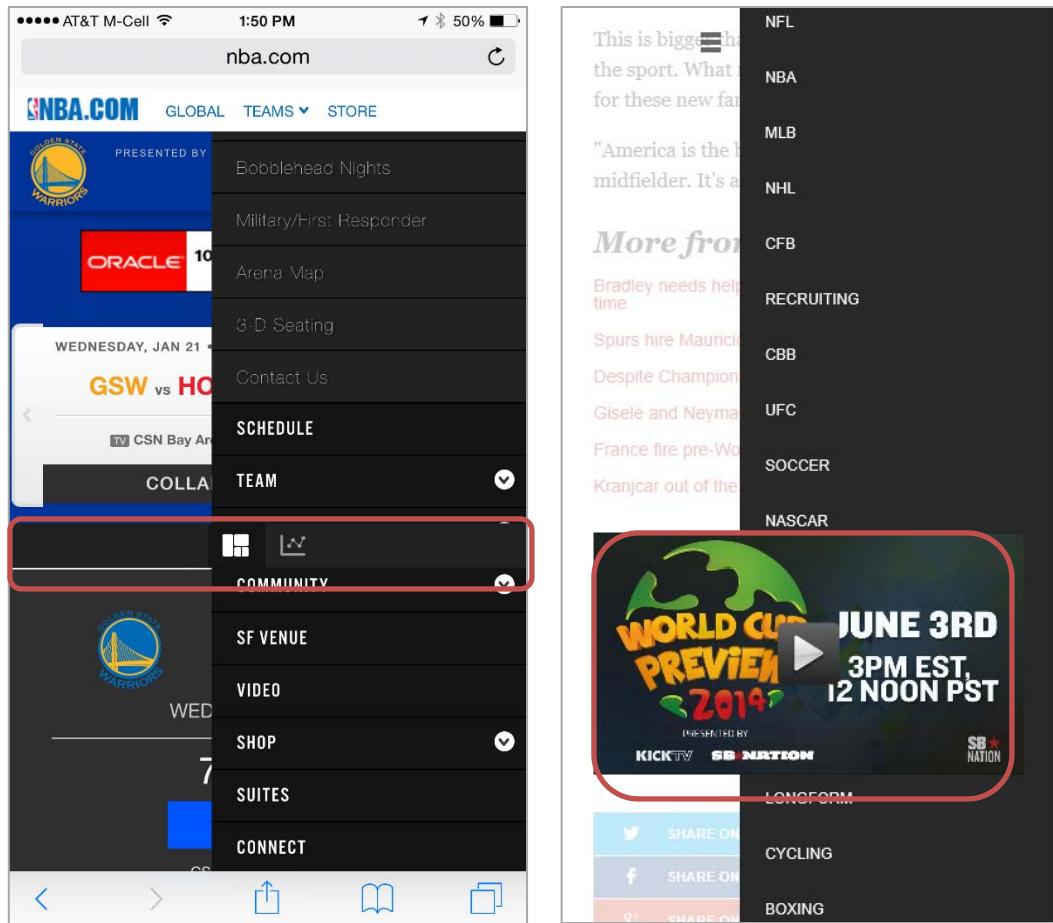


Long lists are not appropriate for drop-downs on iOS. The same drop-down menu occupies only half the screen in iOS (left), but can take up the whole screen on Android (right).

Overlays

89. Avoid using overlays.

Overlays can suffer from three problems: (1) people may not realize the separation between the overlay and the background; (2) for nonmodal overlays, people may trigger actions unintentionally by touching elements of the screen that are not part of the overlay; (3) even modal overlays tend to be buggy and occasionally display background elements that are supposed to be hidden (like in the NBA and SB Nation examples below).



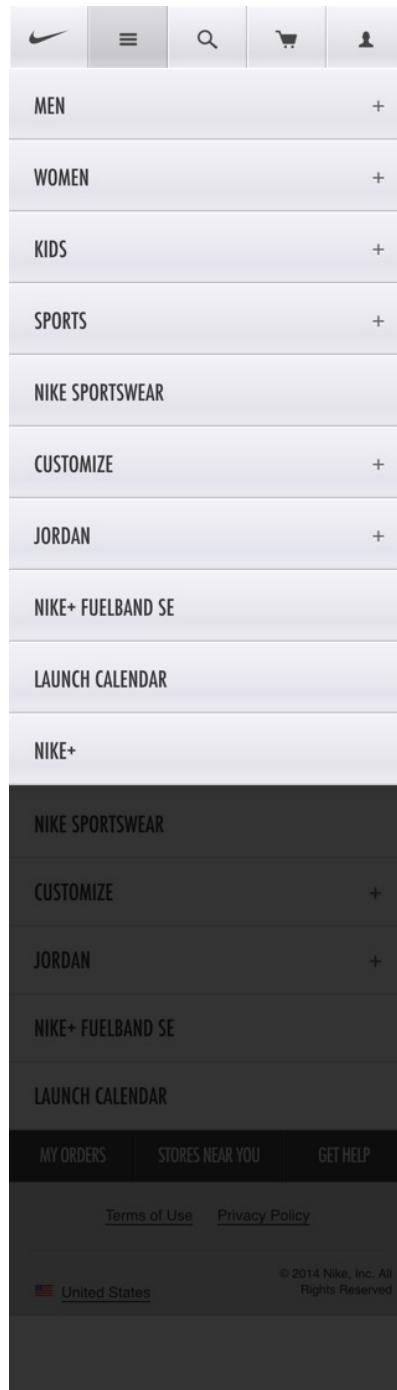
Modal overlays were occasionally buggy and allowed background elements to show on top of the overlay content, as in the screenshots above from NBA.com (left) and SBNation.com (right).

Although it is possible to design usable overlays, we find that it's safer to employ accordions or other types of design elements that displace content on the screen (e.g., sliding menus) instead of appearing on top of existing content. In case you insist on using overlays, here are a few guidelines.

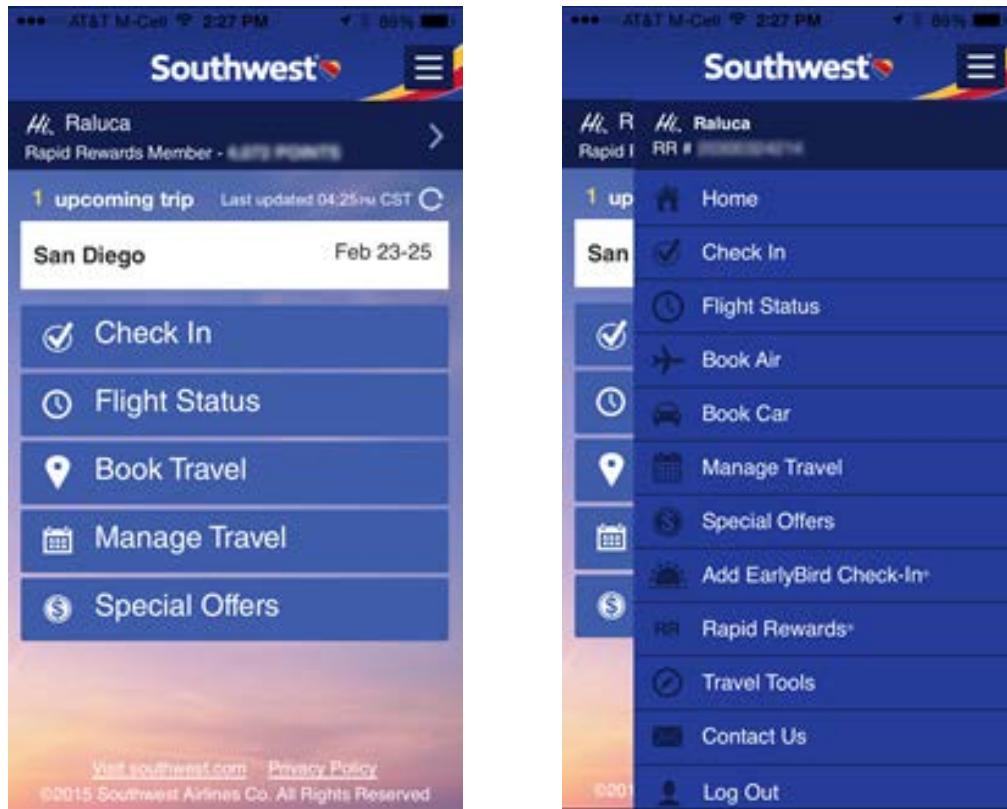
90. Make the content in the overlay clearly distinguishable from the content in the background.

Users may not always realize where the overlay ends and where the actual page content begins. Shading is sometimes used successfully to create that demarcation, but sometimes this is not enough.

In the Nike example at the beginning of the section (repeated below for convenience), the main navigation appeared in an overlay on top of existing content. The rest of the screen was darker as if shadowed; yet because of the similar design elements on the homepage and in the overlay, it was unclear whether the categories on the homepage were also part of the menu.



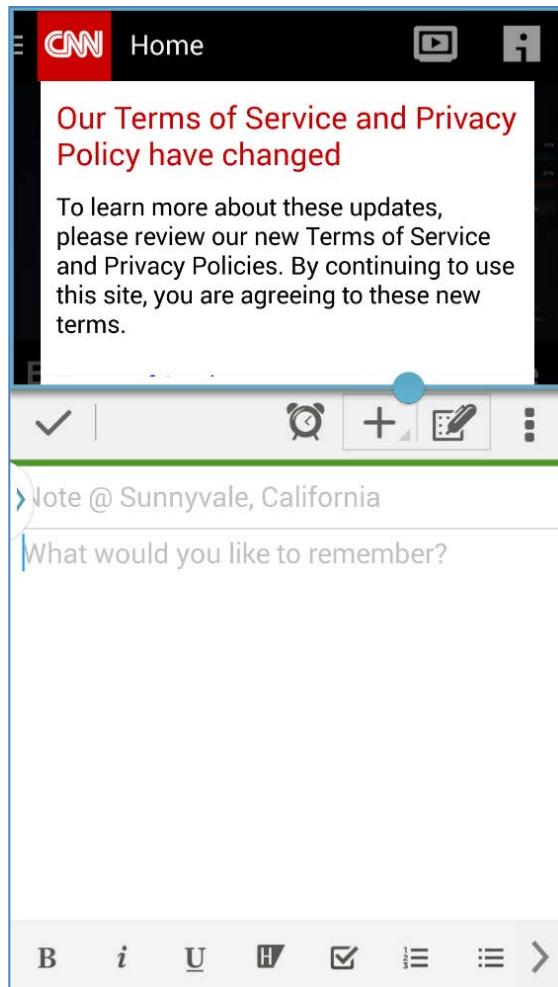
Nike.com: Due to the similar styling, the shadowed categories from the homepage appeared as if they were part of the menu, only darker.



On Southwest.com the background (left) looked as if it were part of the expanded menu (right). People accidentally tapped the icons in the background thinking they were selecting the options in the menu.

91. For accordions and overlays, make the menu content scrollable to ensure that the entire content is accessible on a variety of screen sizes and screen orientations.

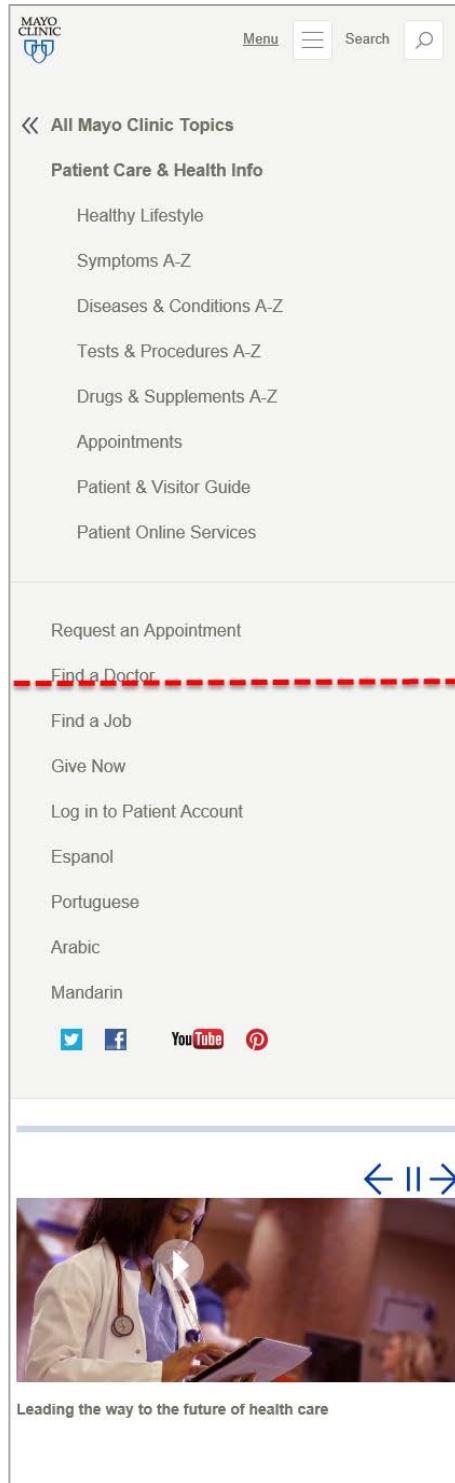
Another problem with overlays (and sometimes accordions) is that they may appear truncated on the screen and may prevent the user from selecting an action. The following example came from a Samsung Galaxy Note phone—one of the few supporting multiple windows. When the screen was divided into 2 windows, the button at the bottom of the modal was not visible, yet the modal dialog was not scrollable. Although we have not seen many users take advantage of this split-screen feature on Samsung phones, it does expose a real danger with modals in a varied landscape of screen sizes: they sometimes may not display fully on smaller-screen phones.



A split-screen view of two apps on a Samsung Galaxy Note phone: the top window ran CNN and the bottom ran Evernote. The CNN modal dialog was cut off above the *Close* button and was not scrollable. Users could dismiss the dialog only if they increased the size of the CNN window.

In other situations, scrolling down in a long menu was interpreted as a scroll of the page content instead of the menu, and the menu closed unexpectedly.

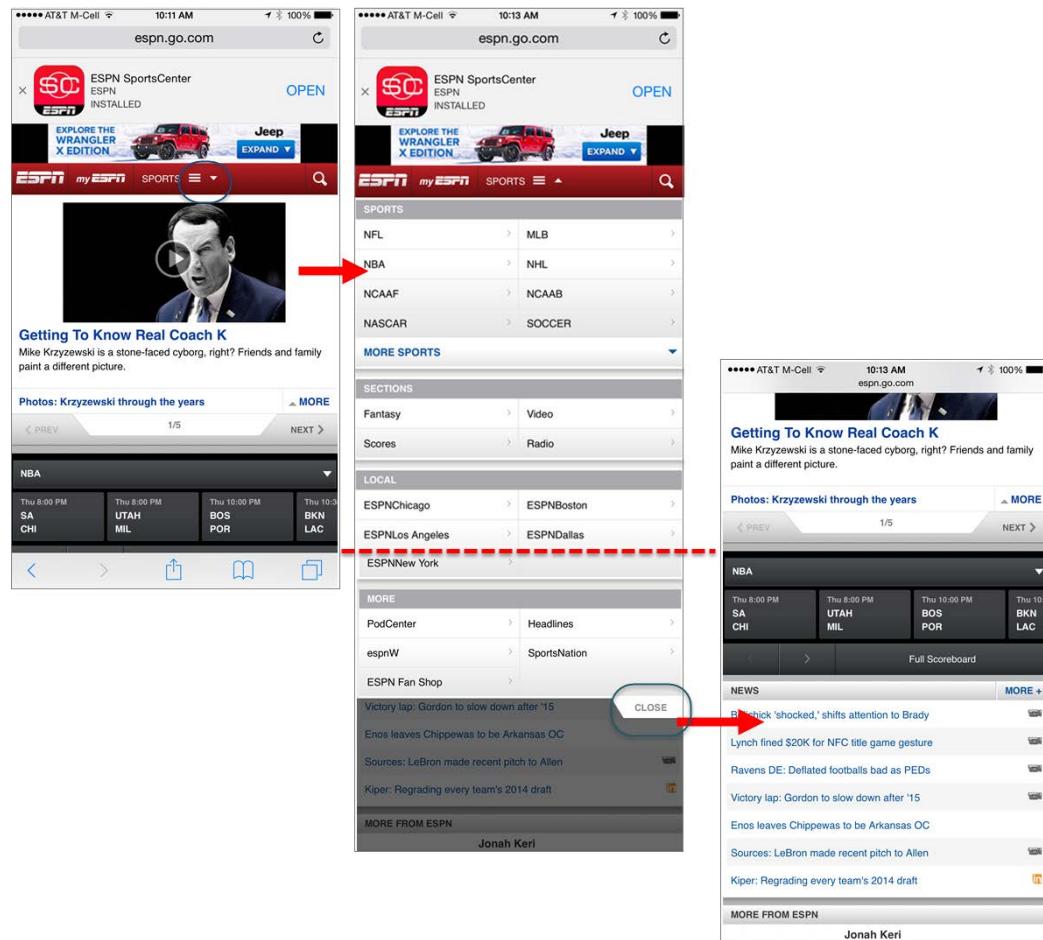
In Mayo Clinic's case (a responsive site), swiping down the long global-navigation accordion menu caused the menu to disappear, and prevented users from reading any of the menu options that fell below the fold.



Mayoclinic.org: the navigation menu was an accordion spanning 2 screens. However, whenever users tried to scroll down to read the menu content that fell below the fold (red line), the menu closed automatically.

92. When the overlay is closed, take users back to the place on the page where they were when they opened the overlay.

Users often feel disoriented when a site takes them unexpectedly to a place where they did not ask to go. Sometimes this problem occurs with long overlays that span more than one screenful and have a *Close* button at the bottom, like in the ESPN example below. The presence of such a button at the bottom of the overlay is laudable, as it saves the user some scrolling. However, when users pressed it, the overlay disappeared and they saw the second screen of the ESPN homepage, which they had not requested and made little sense without the context provided by the first screen.



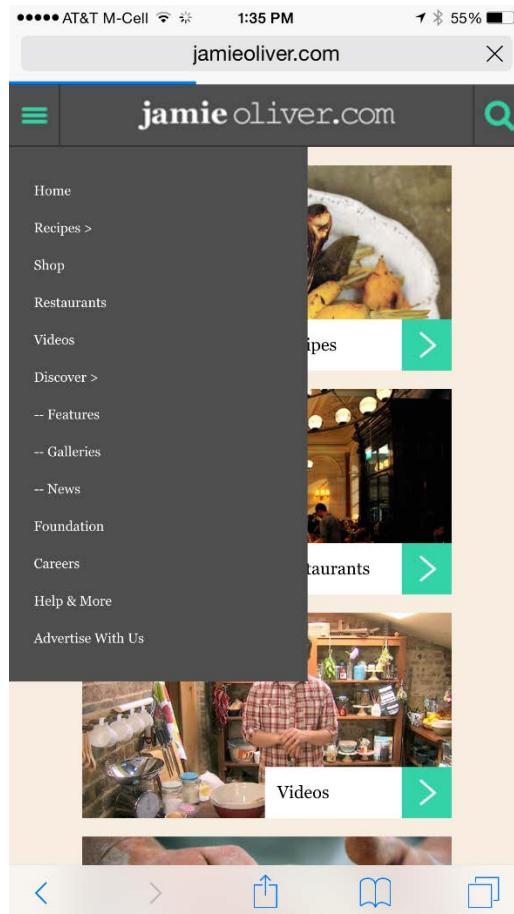
ESPN.go.com: When users closed the long overlay menu, they saw the contents of the second screen of the homepage, instead of being taken back to the top of the page. This created confusion.

93. Consider forcing people to tap on a *Close* icon or button to dismiss the overlay, in order to ensure that the menu is not closed by accident.

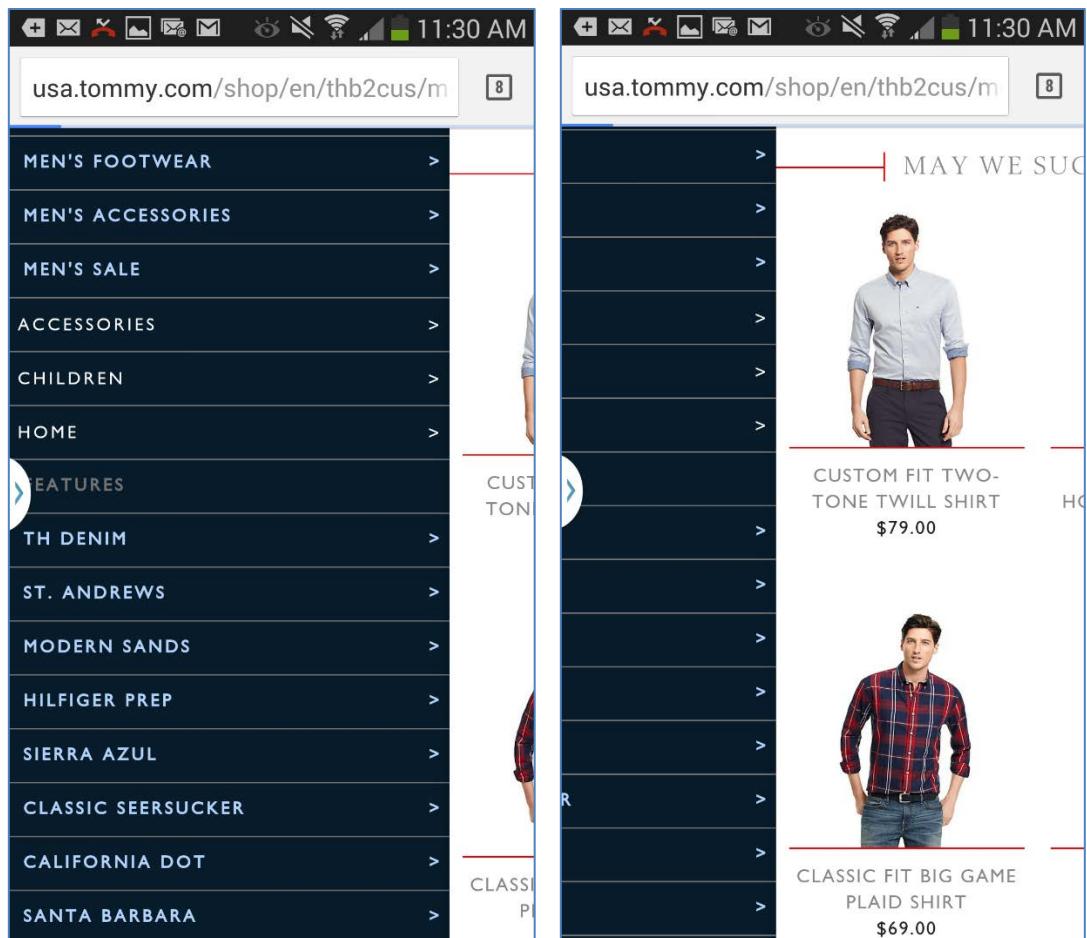
Using a dedicated *Close* button or even letting people close the menu by tapping back on the same button that expanded it minimizes the chance of unintentionally closing the menu by touching something else on the screen.

94. Prefer modal to nonmodal overlays or sliding menus to avoid accidental touches.

When an overlay is nonmodal, people can still interact with elements of the screen that do not belong to the overlay. Although modal dialogs are often considered intrusive on the desktop, the combination of the touch input with small screen size makes nonmodal dialogs impractical on small screens: users may accidentally tap an area of the screen outside the nonmodal dialog and trigger an unintended action with no relation to the overlay.



jamieoliver.com: The hamburger menu was implemented as an nonmodal overlay, and some of the areas behind it were still active and could trigger a new page load if tapped by accident.



Tommy Hilfiger's site used a nonmodal sliding menu for hosting its main navigation. The menu was scrollable together with the page (scrolling in the white region scrolled both the page and the menu). However, accidentally tapping outside the menu counted as a tap on the targets present on the actual page (e.g., on one of the products apparent in the right screenshot).

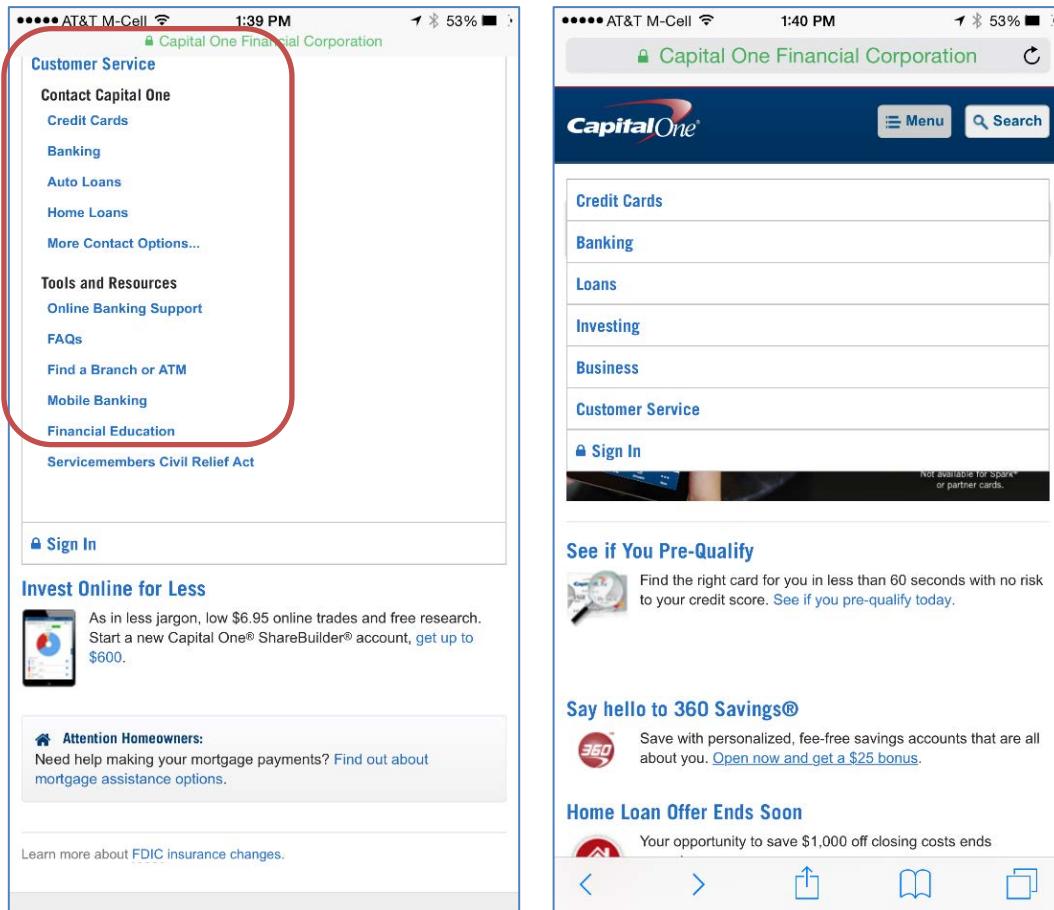
Not only can the user accidentally tap on an element outside a nonmodal menu and generate an unexpected action, but sometimes even touching inside the menu can count as if an element in the background was tapped.

Accordions

95. For accordion menus, make the menu content scrollable to ensure that the entire content is accessible on a variety of screen sizes and screen orientations.

See discussion under guideline 91.

96. Do not close the accordion when people touch other areas of the screen that are not part of that accordion.



CapitalOne.com: Touching outside the global-navigation menu closed the *Customer Service* accordion (inside the global-navigation menu), but left the global-navigation menu open.

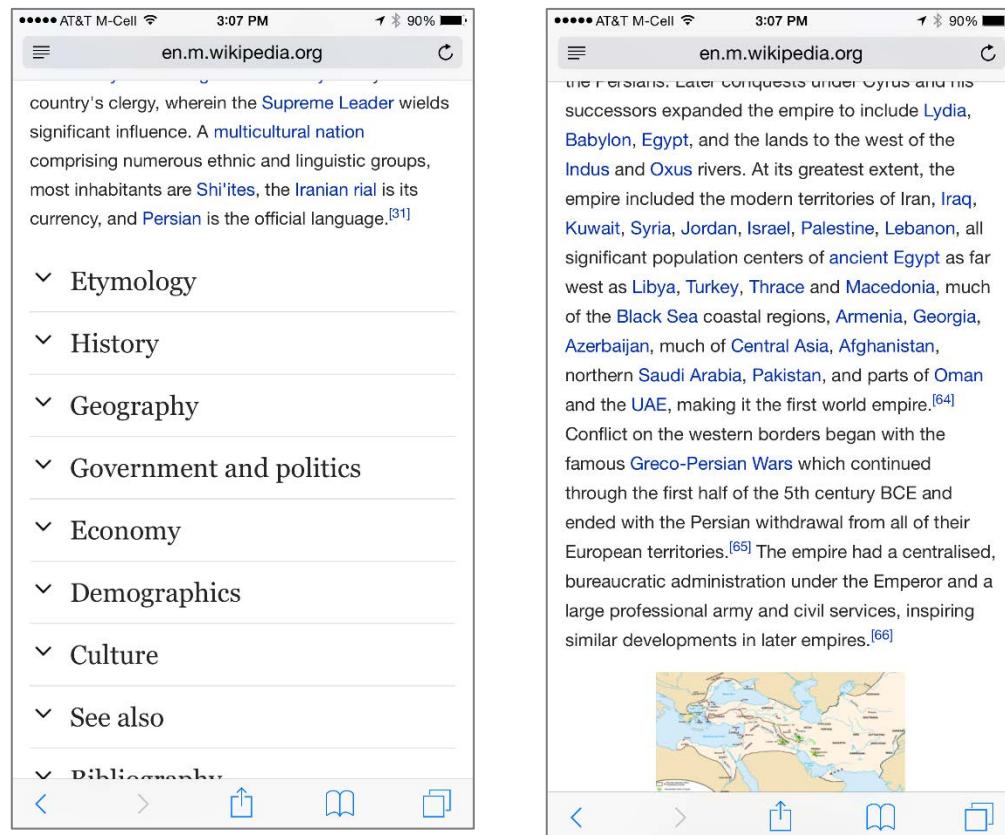
97. Consider forcing people to tap on a *Close element or button* to dismiss the accordion, in order to ensure that the menu is not closed by accident.

See also the discussion under guideline 93.

98. If the content under an accordion is really long (spanning several pages) consider allowing users to quickly close the accordion without scrolling back to the beginning.

Most accordions are closed simply by pressing the same element that opened them (although there is more variation when accordions are used for navigation menus).

Sometimes the content under an accordion can be really long, and as a result, it can be beneficial to allow quick access to the close button. For instance, in the Wikipedia example below, the content of the *History* accordion was very lengthy, and if readers had decided to quit reading it and jump to another subtopic, they would have had to either scroll down to find the next accordion, or scroll up to close the *History* accordion.



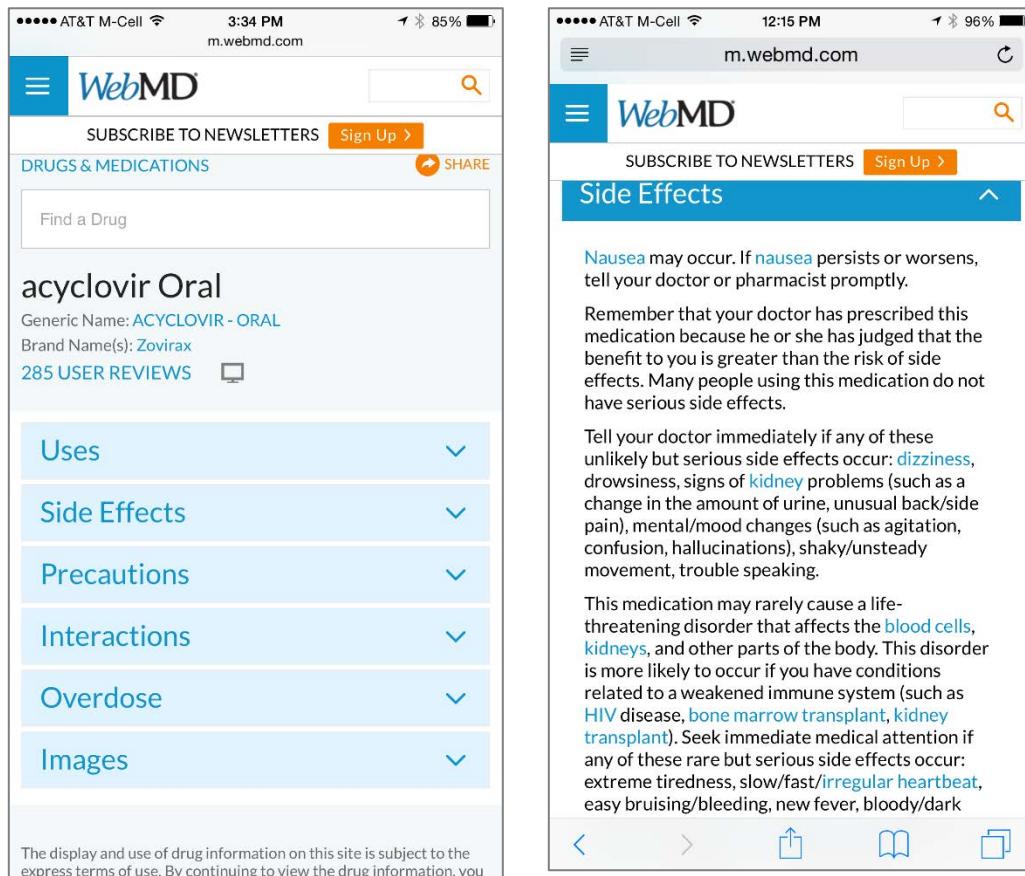
Wikipedia.org: The History accordion expanded to many screenfuls of content. Readers deciding to read for a while and then move on to another topic had to scroll up to close the History accordion or to scroll down to navigate to the other accordions in the list.

In situations like this, a persistent accordion header or another floating element that allows users to quickly close the accordion can speed up the interaction and save people some effort. In the browser, using *Back* to undo the accordion expansion (instead of taking people to the previous page) can also speed up the interaction.

See also our discussion of using accordions for filters under guideline 254.

99. [Web] Allow users to close the accordion using the browser *Back* button.

When users expand an accordion, designers may move the accordion to the top of the screen (as in the WebMD example below) to maximize viewing of the accordion content. Unfortunately that has the disadvantage of making the users think that they navigated to a new page. As a result, they will often try to use the *Back* button to go back to the view with the closed accordion, but instead they will be taken away from the current page.



WebMD.com: When users expanded the *Side Effects* accordion, the content was pushed to the top of the page, making it look as if a new page had been loaded. The expectation was that by tapping the *Back* button users would be able to go back to the closed-accordion view.

To prevent this confusion and also to allow users to quickly go back to the view with the closed accordions, consider using the *Back* browser button as an accordion-collapse button: if the last user action was expanding an accordion, then tapping *Back* should take the user to the closed-accordion page view. Essentially this means treating accordions as if they were in-page (anchor) links.

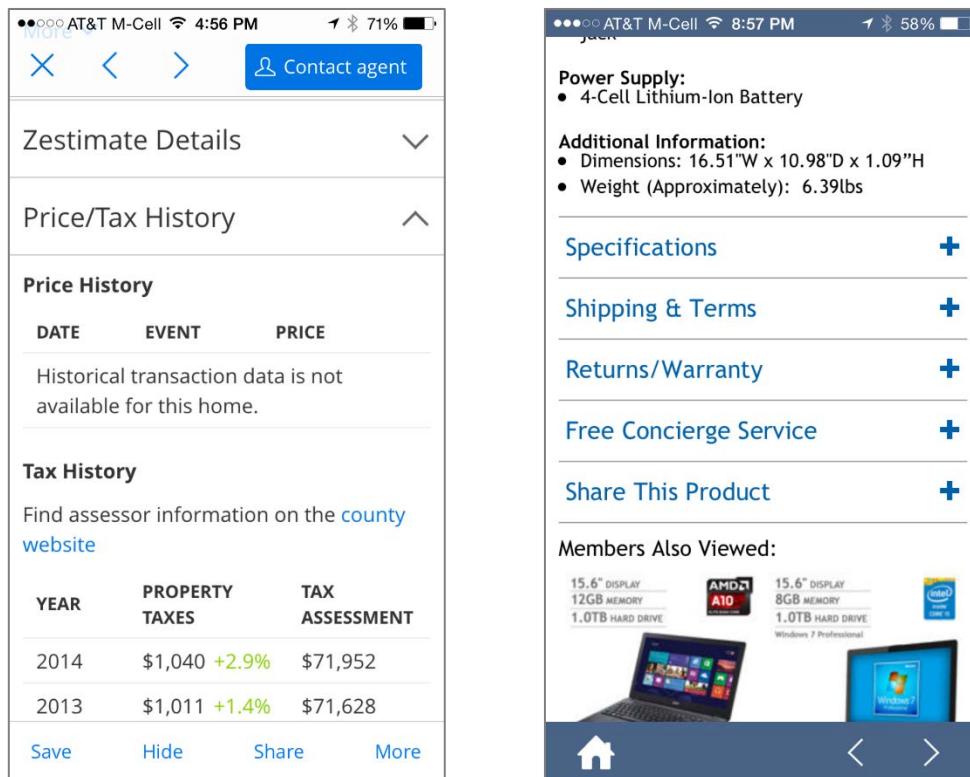
100. By default accordions should be closed.

The label of an accordion should be descriptive enough so that people can guess whether they want to expand it or not. Especially when multiple accordions are present on a page, showing them closed in the beginning gives the user a general idea of the types of content and topics included on the page. If the accordions start open, users will have to scroll down to hunt for new material.

In the Wikipedia example presented under guideline 98, the closed accordions help the user form a mental model⁴¹ of the page content and represent a **mini-information architecture (mini-IA)** of the page — see section *Miniature Information Architecture (Mini-IA)*. They also facilitate direct access to any of the topics of interest, without the user having to scroll down in search of the desired section.

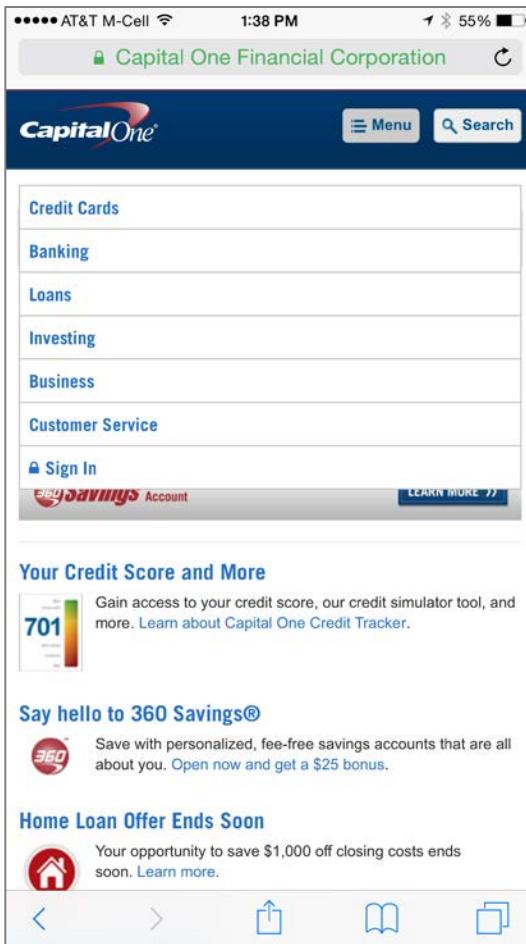
101. Use a plus or down arrow icon to indicate to users that the accordion will expand in place.

The down arrow and the plus signs have become standard for indicating accordions. Avoid using just text.



Down arrows (like the ones used by the Zillow iPhone app in the left screenshot) and pluses (used by Microsoft.com in the right screenshot) signal accordions well.

⁴¹ For a definition of mental models see Jakob Nielsen. "Mental Models." <http://www.nngroup.com/articles/mental-models/>



Some of the links in Capital One's menu were accordions and others (*Sign In*) took people to a different page, but there was no visual differentiation between them. An arrow or a plus sign would have helped manage users' expectations.

Sliding Menus

Sliding menus usually occupy just a section of the screen, so they are not appropriate whenever the options within the menu have long names that need to be truncated or shrunken down so that they fit the width of the menu (see also guideline 86).

Usually they offer no usability advantages over more conventional separate-page menus.

102. Prefer modal to nonmodal sliding menus to avoid accidental touches.

See the discussion under guideline 93.

CAROUSELS

The carousel⁴² has always been a popular way to stick content on the front page without taking up too much space and has seen a resurgence with the advent of the iPad⁴³. (Original iPad designs were fascinated by the etched-screen aesthetic and wanted to control the layout in the tiniest detail. As a result they often forewent vertical scrolling in favor of a card or carousel-like design.)

Like menus and accordions, carousels are popular on mobile because they fit a lot of content into a relatively small footprint. Carousels have many advantages, but one big disadvantage is that they are based on **sequential access**⁴⁴: users must go through all the items in the carousel one by one in order to get to the last one. This interaction is inefficient and provides little information scent⁴⁵: users generally have no information about what comes next. Although carousels may solve content-priority quarrels within the organization, they slow users down (at least in their more traditional incarnations).

103. Users should reach the last item in the carousel in 3-4 steps (e.g., taps or swipes).

Going through items one by one with the hope of possibly finding one of interest is no fun: most people stop after viewing 3-4 different pages in the carousel. If you have a high number of items, use a tabular view instead and allow people to directly access any of the items on the page (see also guideline 260).

Note that the recommendation of 3 to 4 steps to reach the last item does not necessarily mean that the carousel should have only 3 or 4 items. If multiple items per page are displayed, then the carousel may fit more. In the Hulu Plus example below, the hero carousel displayed just one item per page and required 15 swipes to reach the 15th item in the carousel, but the other carousels displayed 2 items per page (thus requiring 13 swipes to reach the last item in the list of 25).

⁴² Carousels on desktop are discussed in this article: Kara Pernice. "Designing Effective Carousels: Create a Fanciful Amusement, Not a House of Horrors."

<http://www.nngroup.com/articles/designing-effective-carousels/>

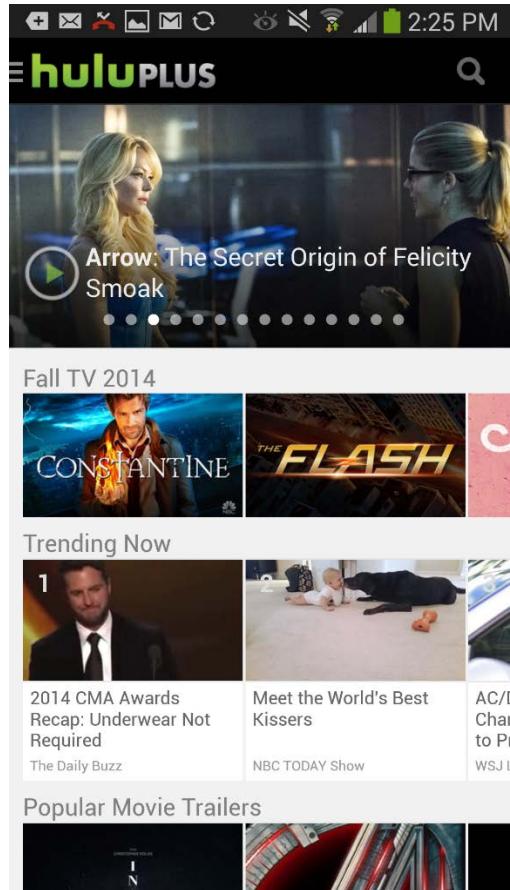
⁴³ See Jakob Nielsen. "iPad Usability: First Findings From User Testing."

<http://www.nngroup.com/articles/ipad-usability-first-findings/>

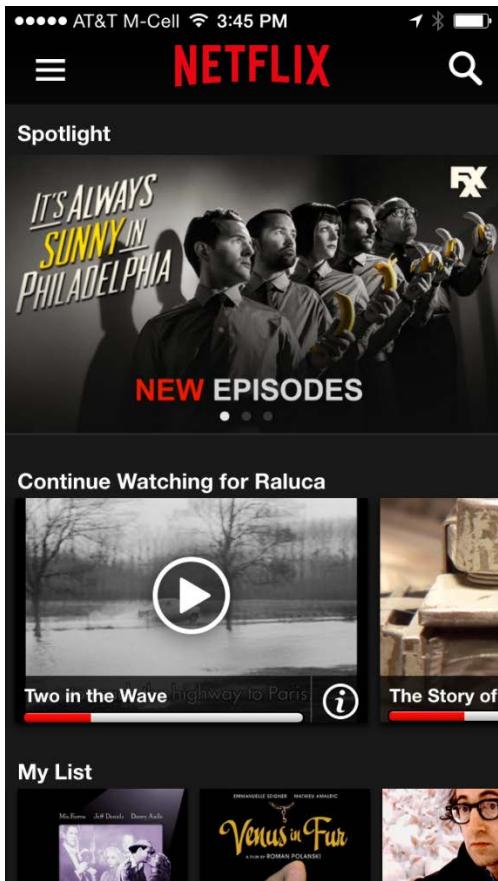
⁴⁴ More information about sequential access and direct access can be found in this article: Raluca Budiu. "Direct Access vs. Sequential Access: Definition."

<http://www.nngroup.com/articles/direct-vs-sequential-access/>

⁴⁵ See Jakob Nielsen. "Information Foraging: Why Google Makes People Leave Your Site Faster." <http://www.nngroup.com/articles/information-scent/>



Hulu Plus for Android incorrectly used carousels to display long lists: the top hero carousel had 15 items; the other carousels on the page had 25 items and required the user to scroll 12 times to get to the last item.

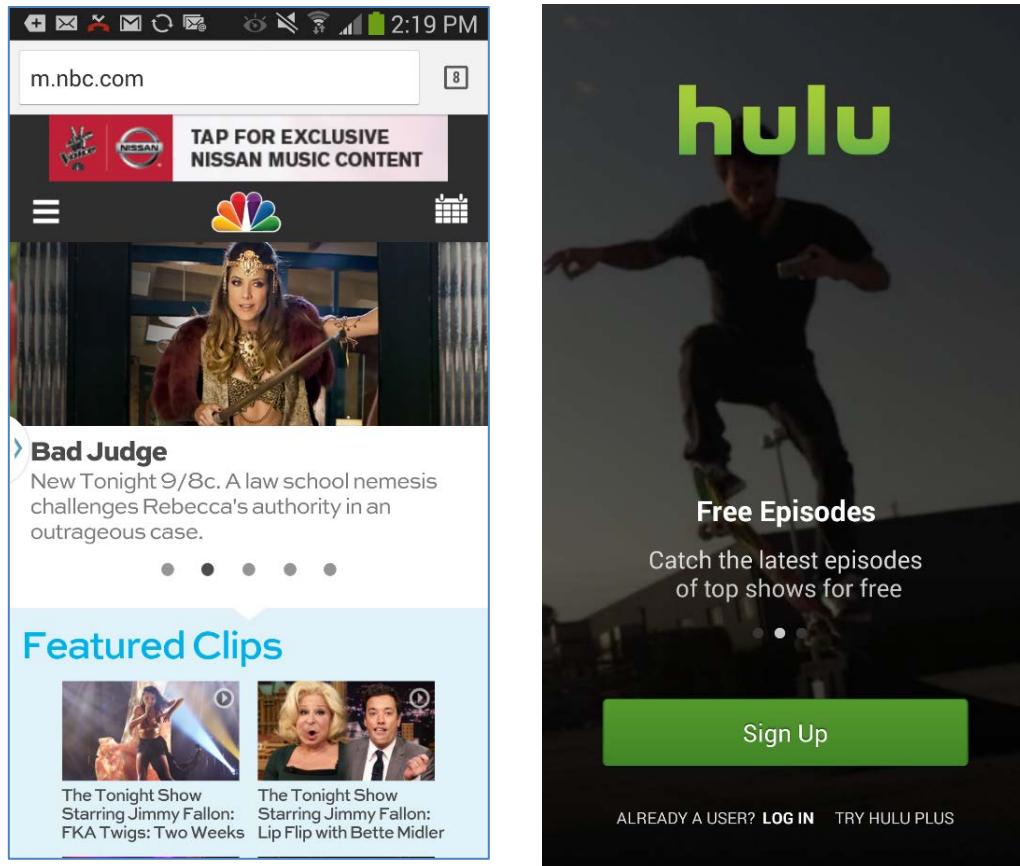


Netflix for iPhone: The hero carousel at the top of the page appropriately contained only 3 items. (Note however that the other two visible carousels — *Continue watching* and especially *My List* — had the potential to contain a lot of items and would have been better accommodated by vertical lists.)

104. Do not use animated carousels. Use carousels that can be controlled by users.

Animated carousels take more time to download. Because the screen is so small on mobile devices, chances are that the users won't even notice them — by the time the carousel has changed, users are likely to have scrolled past them.

Some sites use animated carousels that fill up the whole page, like in the example below from Hulu Plus for Android. Even in these situations where scrolling is not available, by the time the carousel changes the user may have already decided to press one of the available buttons.



Animated carousels are a waste of resources: When NBC's hero carousel changed (left), users had scrolled down and did not longer see it. Hulu Plus for Android (right) displayed a self-looping carousel when the app was first launched. By the time the carousel changed, the user may have already pressed one of the available buttons or links.

Because animated carousels tend to slow down the page and the user may not even notice them, we recommend against them: let users decide when and if they want to move to the next section of the carousel.

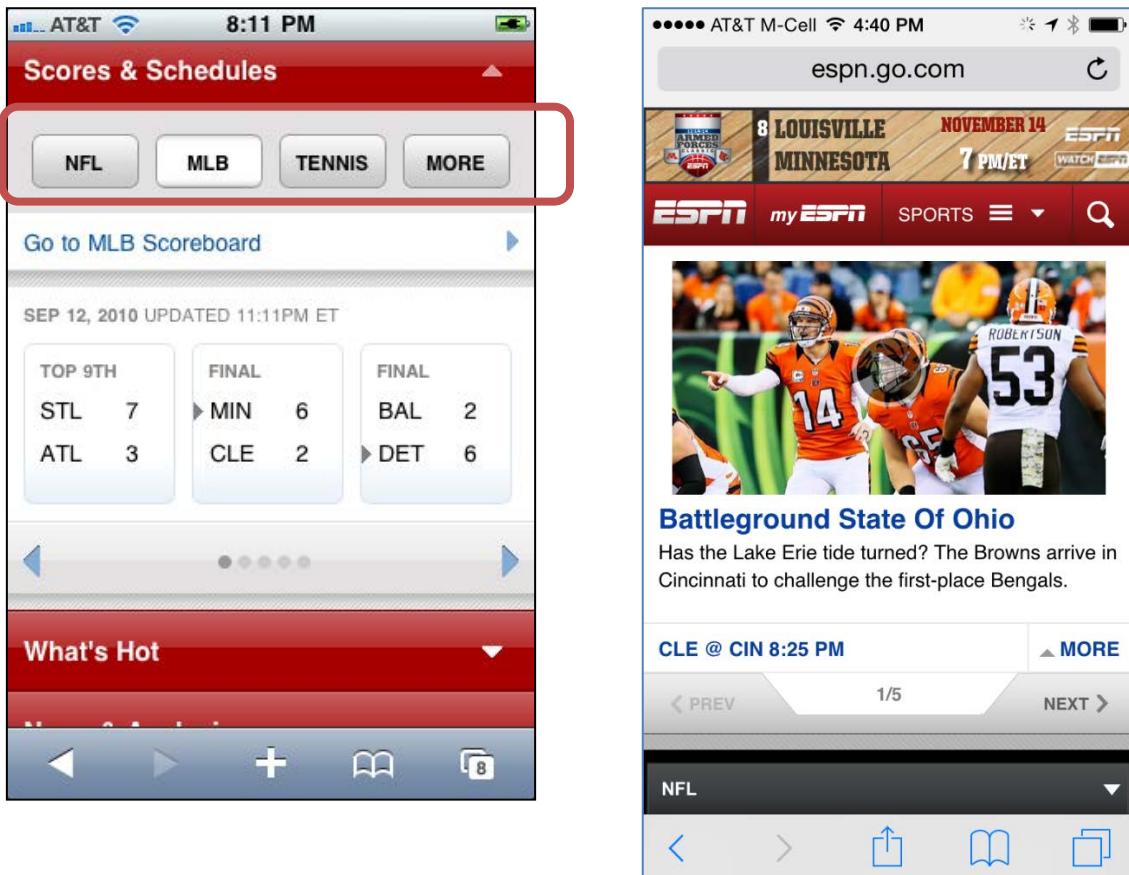
105. For hero carousels that are used for featured content on the homepage, make sure that users can access that content in some other way as well, and not only through the carousel.

As explained in guideline 104, chances are that the carousel won't get noticed on mobile. Or, even when it is noticed, if the page displayed by default is not interesting enough, people may not bother to swipe through the rest of the carousel. Because of that, if the content is in any way critical, there should be some other path to it.

106. Group carousel controls with the carousel content.

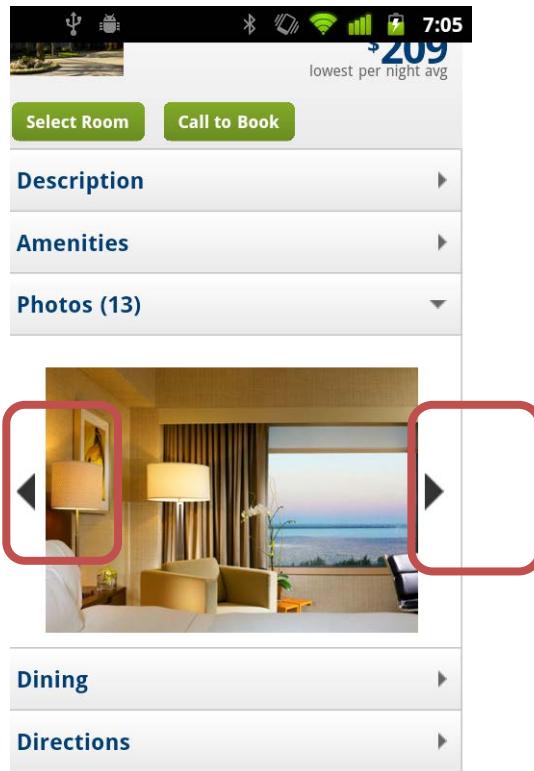
An older version of ESPN's mobile website used an animated carousel (that could also be manually controlled) to display sports scores (see below). Our users missed the automatic changing of the screen. The ESPN carousel also

had a set of controls (NCAAB, NBA, MBL, NHL, MORE) that allowed the users to select a different sport and see scores related to that sport. Unfortunately, many users did not understand that those buttons controlled the carousel.



Carousel controls separated from the carousel: In an older version of ESPN.com (left), the buttons at the top controlled the sport displayed in the carousel. Many users did not realize that the buttons were related to the carousel. A newer version of ESPN (right) placed the carousel-controlling buttons *Next* and *Prev* too far away from the content of the hero carousel.

Travelocity had a better implementation of a carousel: the arrows that moved to the next picture were part of the carousel and could not be separated from the carousel by scrolling.



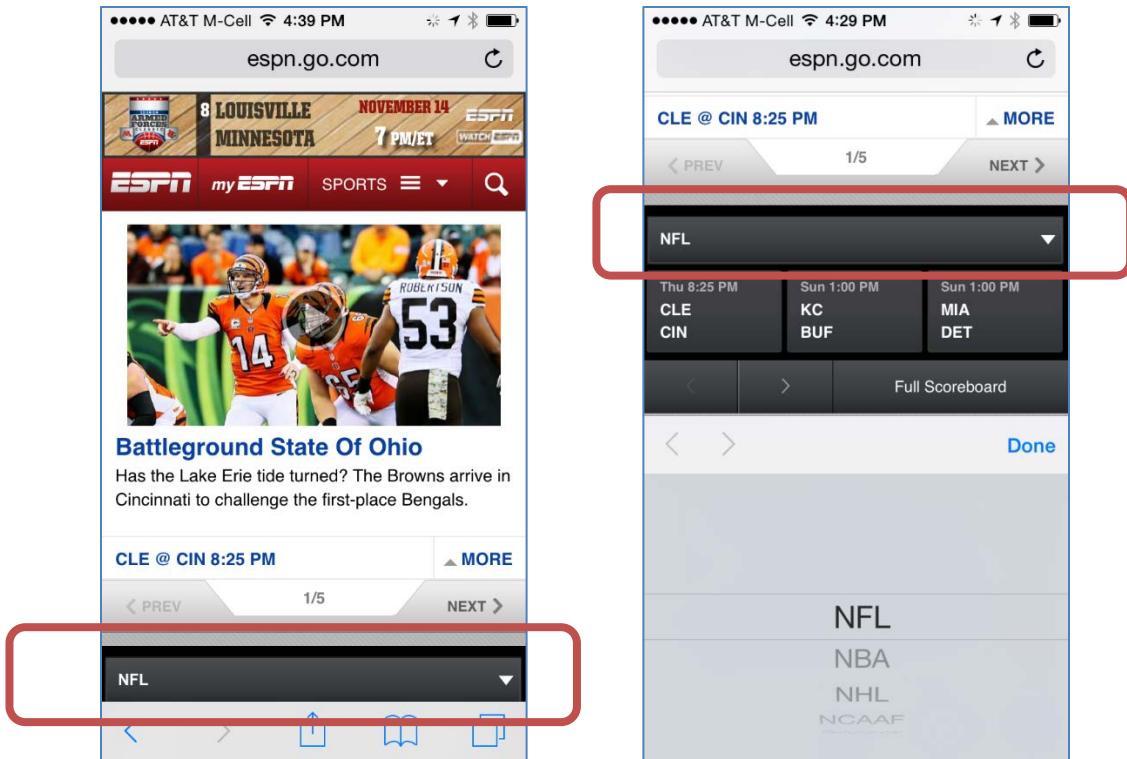
Travelocity used a carousel to show pictures of individual hotels. Their version was more usable because the controls that moved to the next picture were part of the carousel. Such a design is unlikely to run into the scrolling issues that we noted with ESPN.

107. When users select a control that acts upon the information displayed in the carousel, make sure that they can see the carousel content.

When one of our users tested the older version of ESPN in the previous example and tried to see scores for his favorite sport, only the different sport buttons were visible above the fold (the carousel was below the fold). Changing the buttons resulted in no visible effect other than confusing the user (who had expected that the buttons take him to a different page):

"Where does it say 'basketball'? Usually if you got to ESPN it gives you choices — where is it? Oh, here it is, *Scores & Schedules* [clicks on *Scores and Schedules*], *NBA* [clicks on *NBA*]. [Nothing happens] You know what I really would do at this point? And it's not even moving, I wonder why ... Let' me try these other ones to see if there's something else going on [tries to press *MLB*]... Maybe there's no ESPN game tonight, I don't see anything scheduled..."

A newer version of the ESPN mobile page still used a carousel; the drop-down menu at the top of the carousel allowed users to change the sport and see different scores. The new carousel still separated the controls from the carousel; however, if the user selected one of the sports, the page would automatically scroll down so that the user would see the carousel.



ESPN.com: the sports drop-down that controlled the type of scores displayed in the carousel was separated by the carousel (which could appear below the fold). However, when the user tapped on the drop-down to change the sport, the page scrolled down to make the carousel visible.

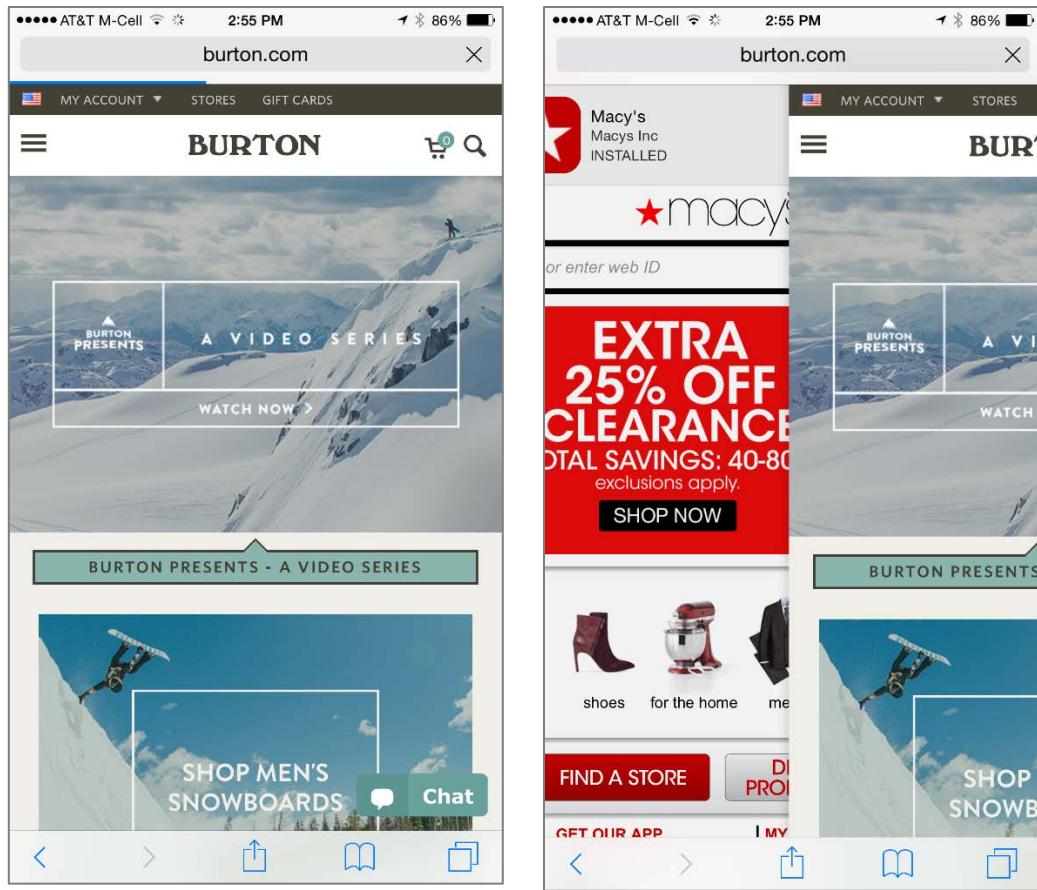
108. Allow users to use the swipe gesture to navigate through the carousel.

By now most users have become familiar with this gesture for horizontal navigation, and they expect to use it to advance through the carousel items. Make sure your carousel supports that gesture.

In the ESPN example above (see guideline 107), the carousel did not support the swipe and could be operated only through the arrows.

109. [iOS] Allow some swipe space (e.g., a page gutter) around carousels to avoid swipe ambiguity.

Swipe ambiguity refers to the fact that the same swipe gesture can be interpreted to mean different things depending on the precise location where it is executed (see more about this in the section *Gestures*). Since iOS 7, swipe ambiguity has become a constant danger with iOS. For instance, in the Safari browser, a horizontal swipe on the left edge is synonymous with "back": it takes users back to the previous page. Unfortunately, the same gesture is also used to navigate through carousels when initiated in a slightly different spot on the screen.



Burton.com: The carousel occupied the entire screen width; users accidentally went back to the previous browser page when they used the swipe gesture.



Net-a-porter.com made it easier for users to swipe through the image carousel by leaving some white space between the carousel and the screen edges.

Carousel Cues

One of the problems of the nonanimated carousels is that they can have low discoverability: users may never guess that they can get more content. Therefore, it's important to signal to users that they are dealing with a carousel.

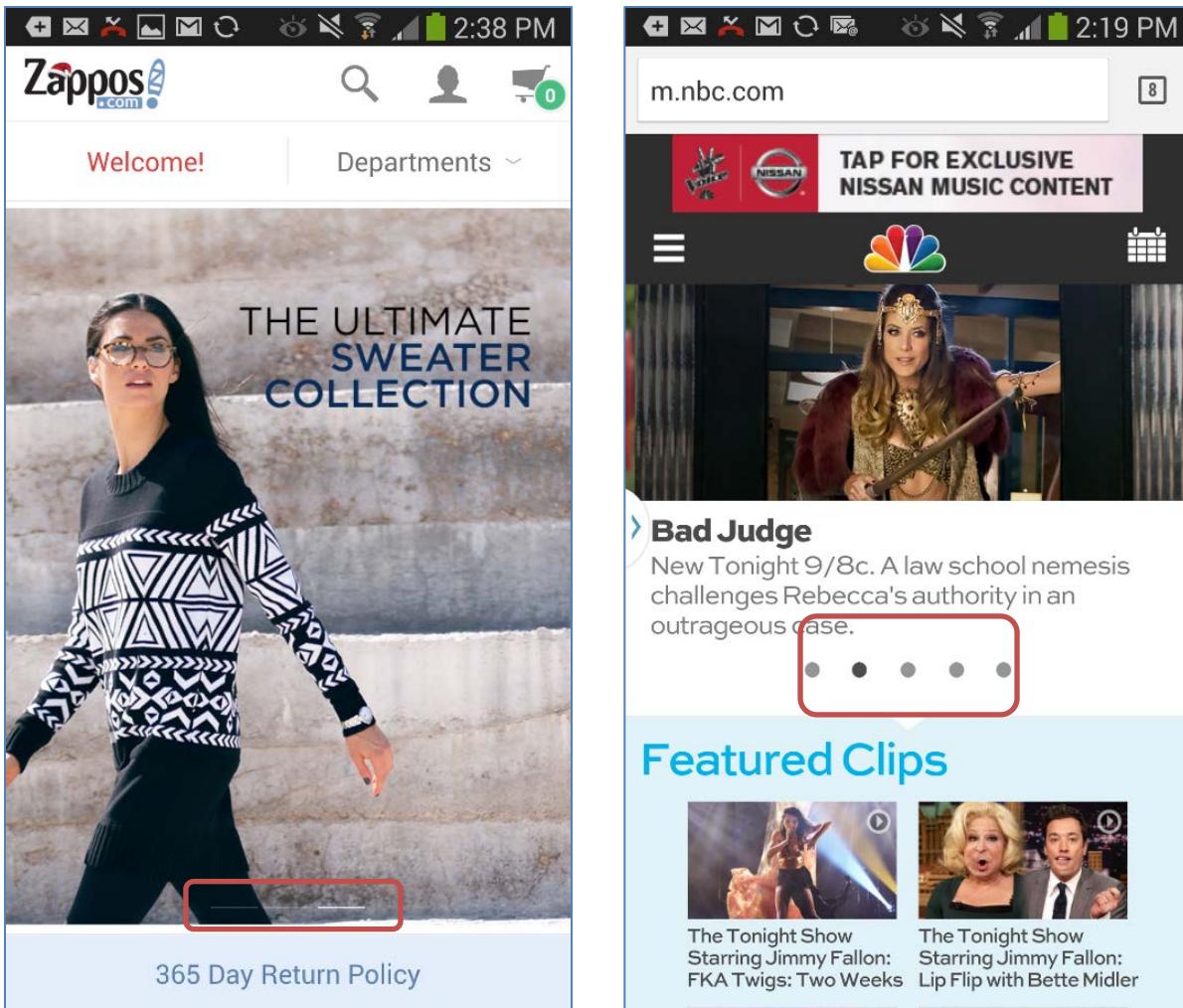
Some carousel cues are stronger than others. As we discussed before, arrows placed in line with the carousel image(s) are better cues than arrows place under or above the carousel.

110. Use arrows or the illusion of continuity to show users that they can see more items in a carousel.

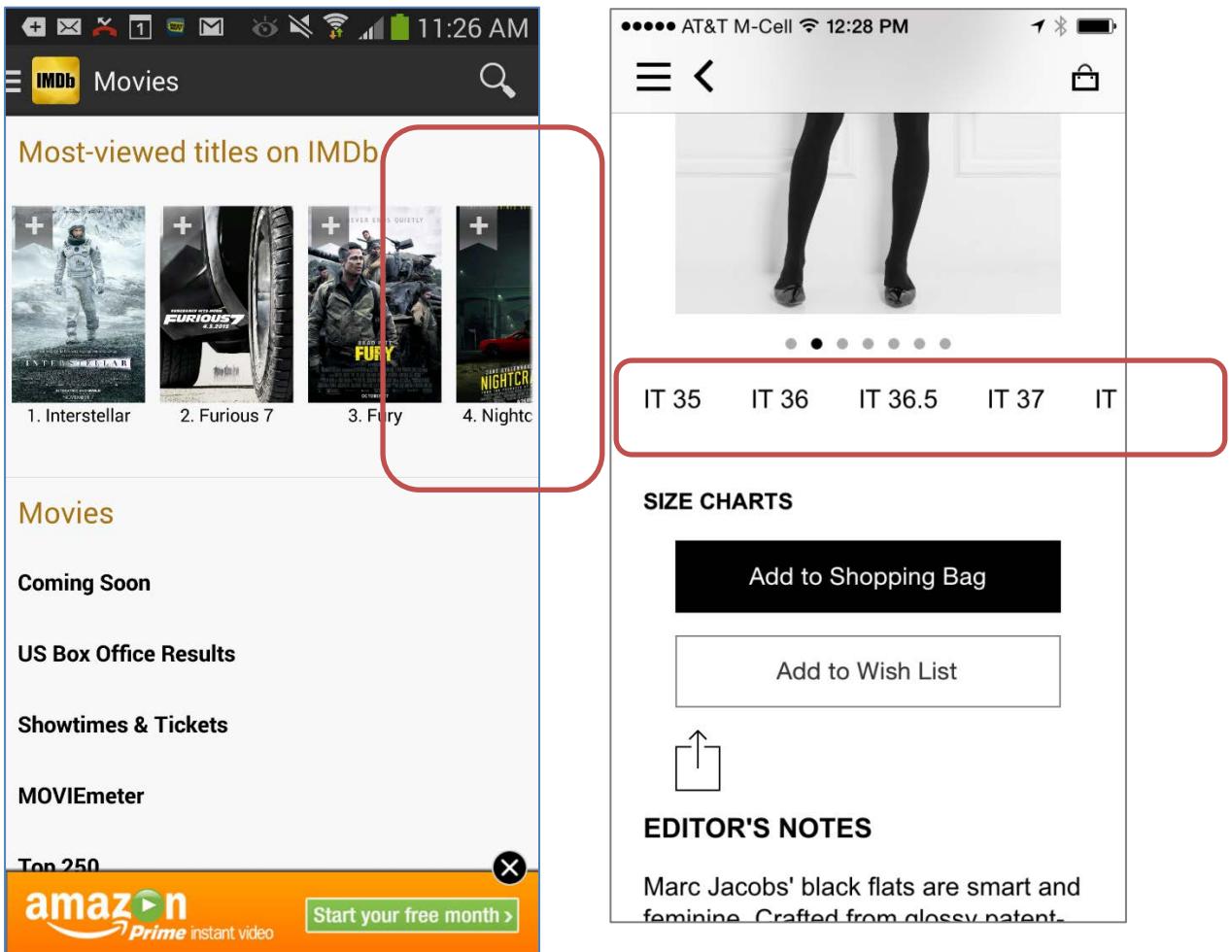
111. Do not use dots or lines to indicate pages in a carousel.

Dots are generally weak cues (because people often don't notice them, especially if they overlap with busy image backgrounds). Half images or text that looks like it's continued to the right of the image are strong cues — users quickly understand that they can get more content.

Below are a few examples of designs with either strong or weak carousel cues.



Weak carousel cues: Zappos (left) showed dashes on top of the carousel item. The dashes blended with the image and were hard to notice. NBC.com (right) used gray dots on white background; although these dots were more conspicuous than the dashes on Zappos.com, they were still a weak carousel cue.

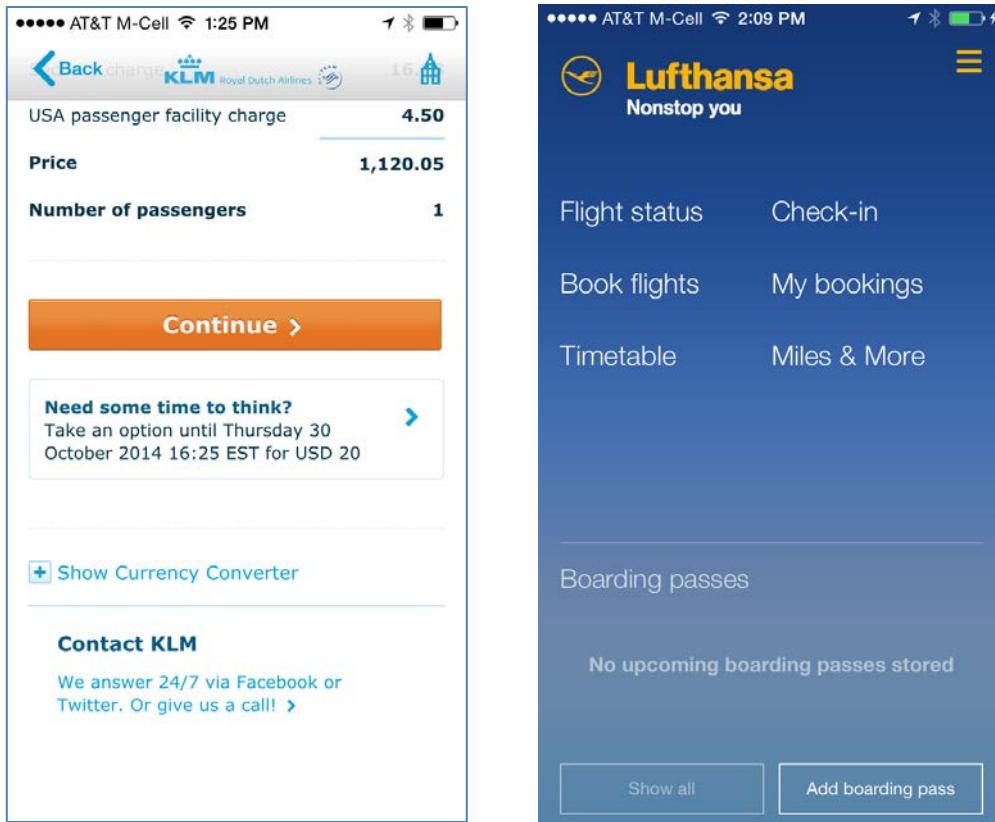


Strong carousel cues: Half images (like in the IMDb app for Android on the left) and incomplete words (like in the Net-a-porter app for iPhone on the right) signaled users that there was more content to the right or left. (The dots used by Net-a-porter for the image carousel were less good cues, however.)

BUTTONS AND BUTTON PLACEMENT

112. Label buttons in a descriptive way.

As with link names (see guideline 273), a button label needs to be explicit and tell users what actions the corresponding button will trigger. Labels such as *Continue*, *Next*, or *More* are too vague and do a poor job of telling people what to expect.

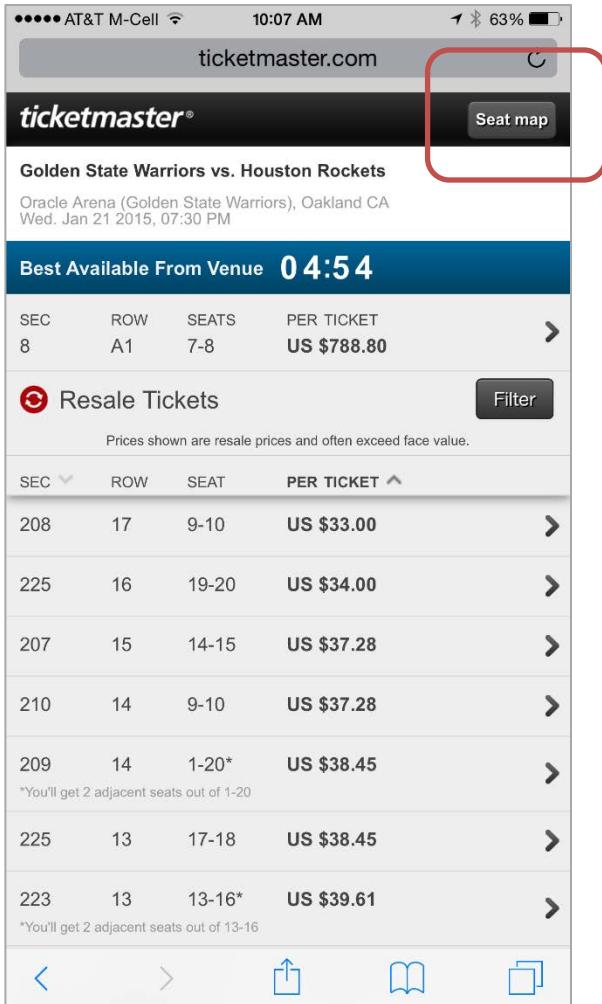


Button labels need to be explicit: The *Continue* button on klm.com (left) had a vague label; a better one would have been *Book this flight*. The *Add boarding pass* label in Lufthansa's iPhone app was clear enough, but *Show all* could have been made more explicit if it read *Show all boarding passes*.

113. Group related controls (and related information).

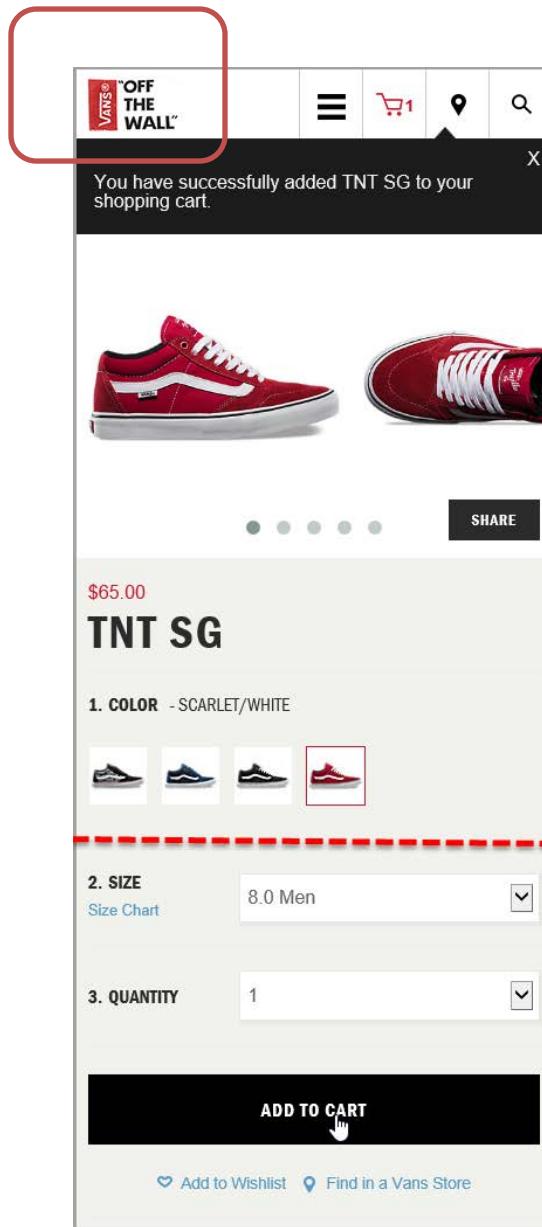
One of the principles of Gestalt psychology is the law of proximity, which says that people tend to perceive items that are close together as related, or part of the same whole. Conversely, things that are far apart are perceived as separate or unrelated. In interface terms, elements that are far apart will be considered unrelated, whereas those close together look connected.

Take advantage of this law, and make sure that all the buttons relevant to completing a task are placed in the same region of the screen, close together. Because the screen is so small, one might argue that the law of proximity does not count, since everything is close on a phone screen. Wrong! While it's true that the problem is even bigger on a larger tablet screen or on a computer, we've witnessed many cases where people had difficulty finding the right button because it was "out of the way".



Ticketmaster.com: The *Seat map* button was far away from the ticket information and prices, and some users had a hard time finding it.

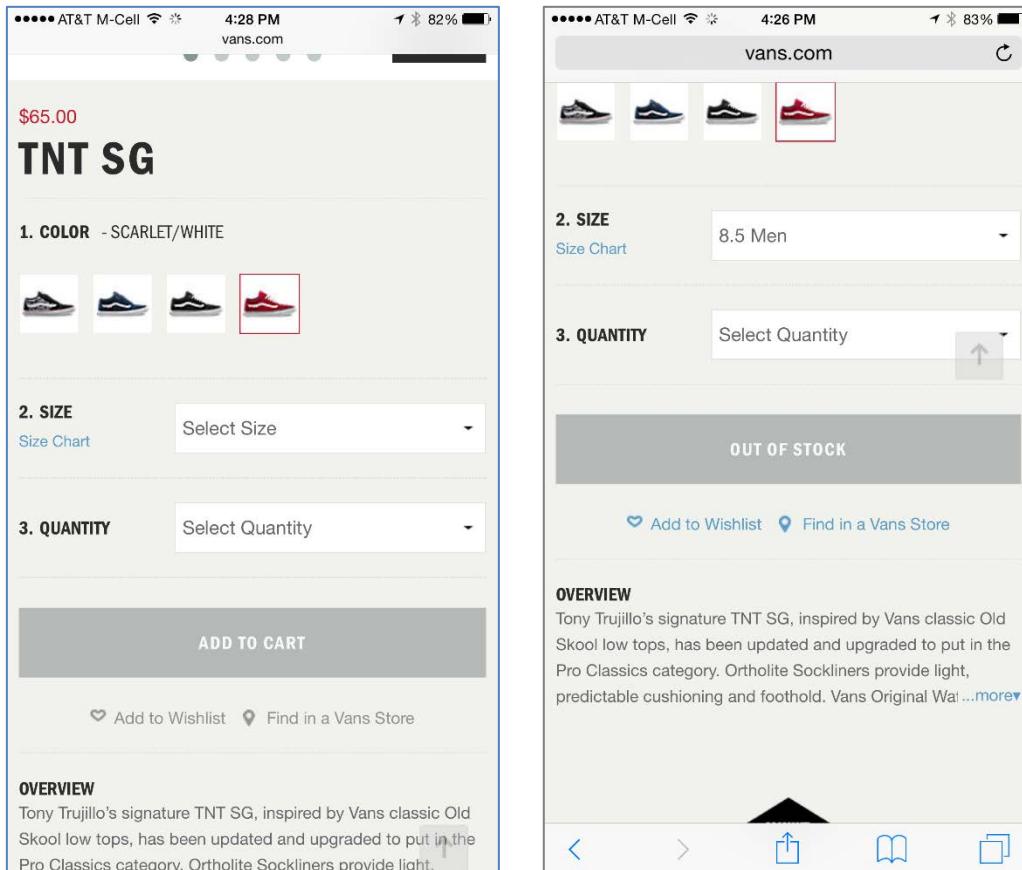
On the Vans website, users trying to purchase shoes were not sure whether they had successfully added the product to the cart: although the cart icon displayed a small badge on top of the cart, indicating that the cart contained an item, this icon was too far away from the *Add to cart* button (in fact, it was at the very top of the page and not even visible to users).



Vans.com: The *Add to Cart* button was not on the same screen as the cart icon and the feedback message at the top of the screen (the red dashed line indicates the fold on an iPhone 6 Plus). When people added an item to cart, they did not see the notification and the badge on the shopping cart icon.

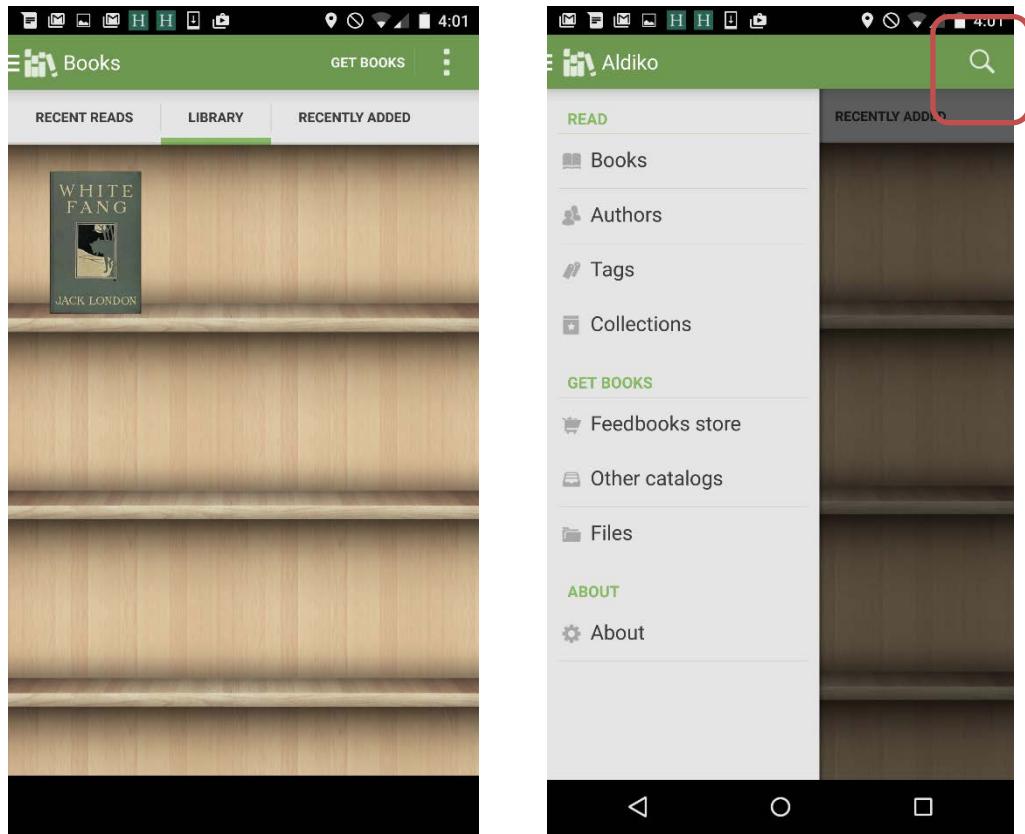
Also on Vans.com, some users did not notice that their desired shoes were out of stock also because that information was disconnected with the choice of a size. When our study participants selected their sizes, the *Add to cart* button changed into *Out of stock*. Users however did not notice that change because that button was too far away from their focus of attention. They went on to select a quantity, and they were puzzled because they could only choose a quantity of 0. Only after that did they see the new *Out of stock* label. Instead,

the site should have shown a message next to the size immediately after the user had selected it, or even better, should have shown only in-stock sizes.



Vans.com: Users did not notice that their size was out of stock because the change in label of the submit button (from *Add to cart* to *Out of stock*) was too subtle and too far from where their attention was focused (that is, too far from the *Size* and *Quantity* fields).

On Aldiko for Android, the search tool was hidden unless the user chose to expand the main-navigation menu. (Hiding the search under a menu is acceptable in certain conditions — see guideline 231.) However, the search tool was not inside the menu — it was on the opposite corner of the screen. Users did not notice that other parts of the screen had changed because their attention was focused on the menu.

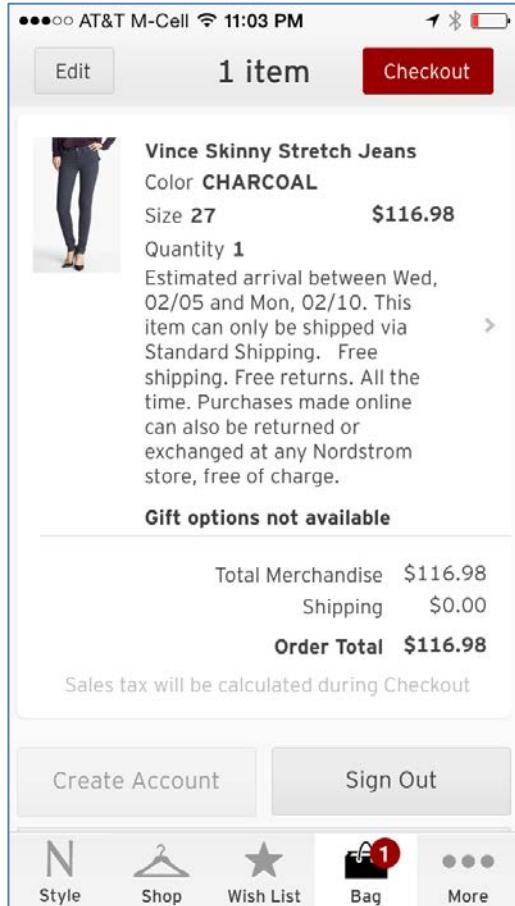


Aldiko for Android: When users expanded the menu, the search icon appeared in the top right corner of the page. Users did not notice it because it was far away from their focus of attention (and from the menu button).

114. On a page, do not place main calls-to-action at the top of the page, in the tool bar or navigation bar.

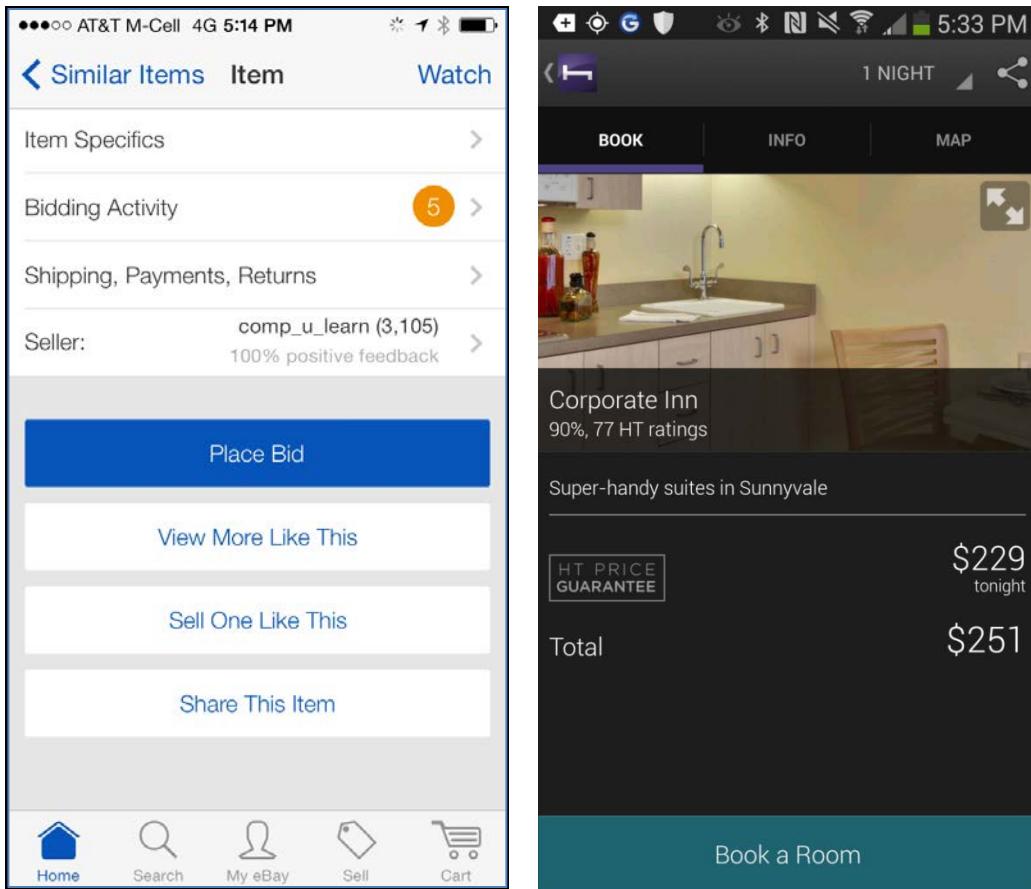
Even if the page is not a form, important calls-to-action should not be placed at the very top of the page (for instance in the tool bar or navigation bar), before relevant content.

When users get to a page, they normally scan the page down for information. They may reach the bottom of the page or stop when they think that they have gathered enough information to make a decision. If they want to take an action based on content on the page, they expect to do it right there and rarely think to look up at the top of the page.



Nordstrom for iPhone: The main call to action *Checkout* was placed at the top of the screen and went against the natural workflow on this page. Users normally scan the information on the page and then expect to proceed to checkout. Instead, on this page, in the logical position for checkout, they found an invitation to create an account or sign out.

Contrast this top placement with the one in the examples below. In both these examples (eBay and Hotel Tonight), the main call to action was placed at the bottom of the page, in a position that matched the natural workflow on the page.



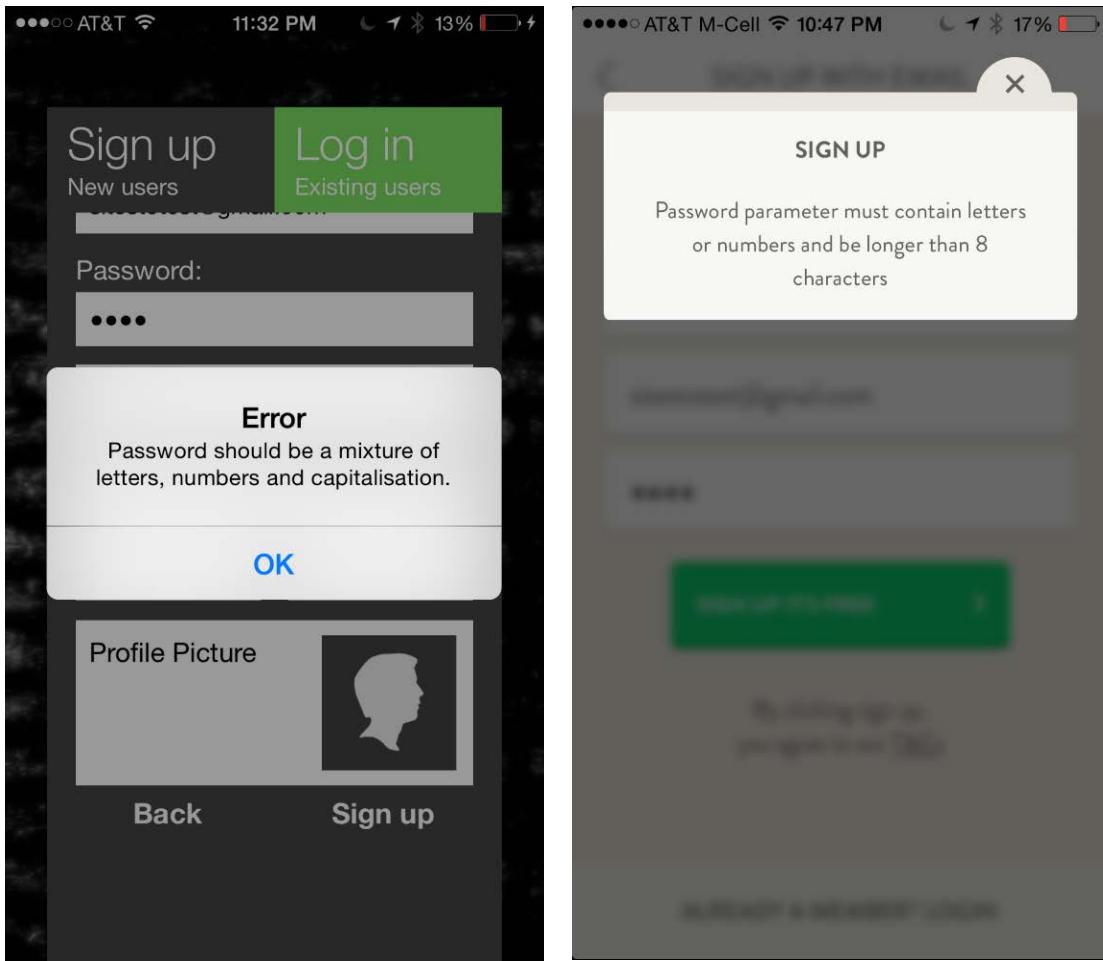
Both eBay for iPhone (left) and Hotel Tonight for Android (right) placed the main calls-to-action at the bottom part of the page, taking advantage of users' natural tendency to scan top-down.

ERRORS AND ALERTS

Alerts are usually used for communicating something important to the user. When using alerts, keep in mind that the alert content is not easily accessible and, if users are to follow any instructions contained in the alert, they will need to memorize them. As a result, it's best to not use alerts for any information upon which users must act.

115. Don't use alerts, fading popups, or separate pages to signal an error in a form.

Most of the time instructions need to be visible at the time when the user fills in the form; otherwise, users must memorize them and then act upon them. Unless it's content that you expect most users to know, instructions should be displayed next to the field that they refer to. Once the alert or popup disappears, the user will need to remember what the error was in order to fix it. Especially on mobile, where interruptions are frequent, it's easy for people to forget even the simplest instruction.



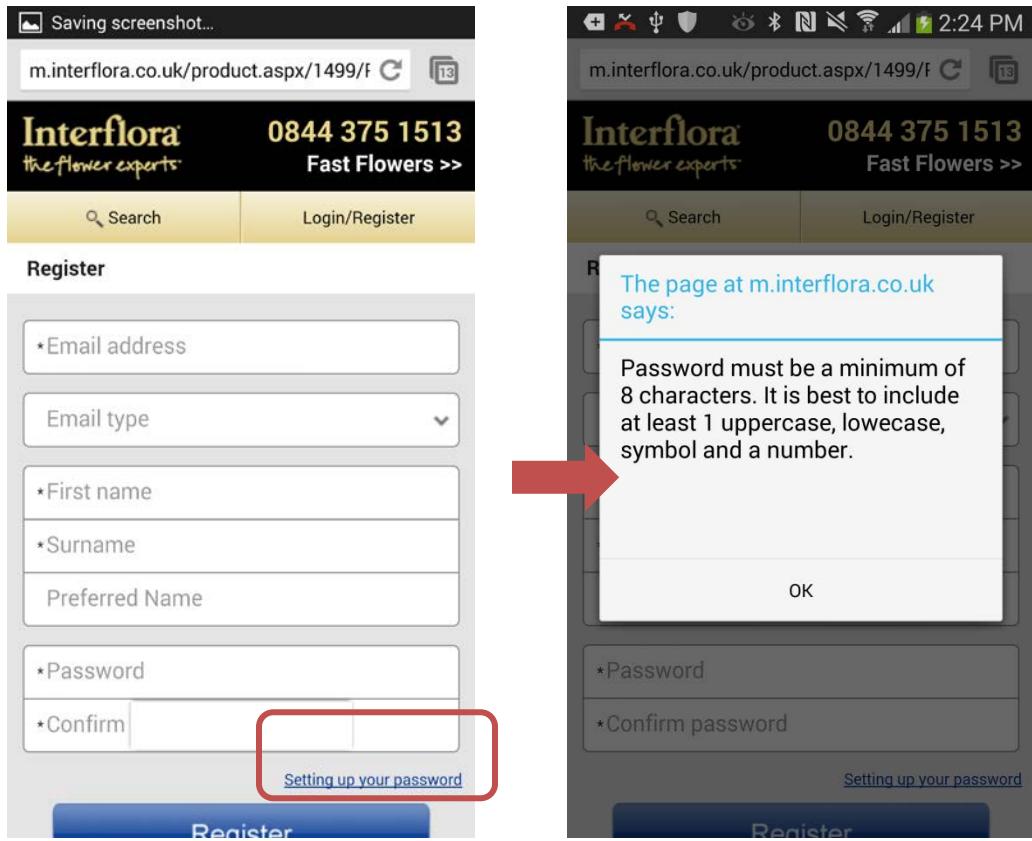
Quikkkly and Headspace for iPhone: When users created a wrong password, the error message appeared in an alert. Users had to dismiss the alert and remember the password requirements while choosing a new password.

The image contains two screenshots of mobile interfaces. The left screenshot shows a numeric keypad with a central overlay displaying 'Member Number Incorrect' and the instruction 'Please enter your full 16 digit member number.' An 'Ok' button is at the bottom of the overlay. The right screenshot shows a sign-in form for 'm.drugstore.com/checkout/'. It features an orange error box at the top stating 'Could not sign up - Passwords require at least one number.' Below it are fields for 'email' and 'password', a 'forgot password?' link, and a 'sign in' button. A horizontal line labeled 'OR' separates this from a 'NEW CUSTOMERS' section with fields for 'first name', 'last name', 'address line 1', 'address line 2 (optional)', and 'zip'.

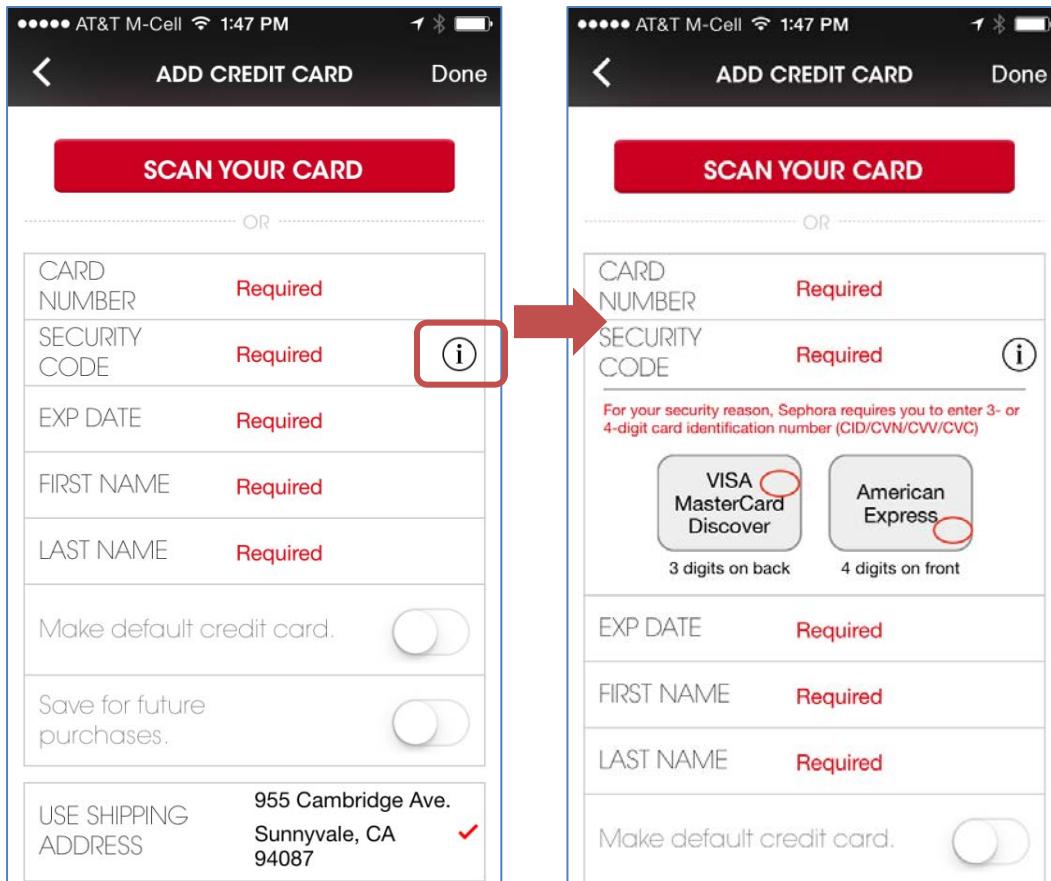
Alerts (used by AAA for Android — left) and popovers (used by drugstore.com) are not recommended for displaying errors. In the drugstore.com example, the orange box fades away after a few seconds.

116. Don't use alerts, popovers, or separate pages to display instructions for filling in the form, or other information that the user must refer back to.

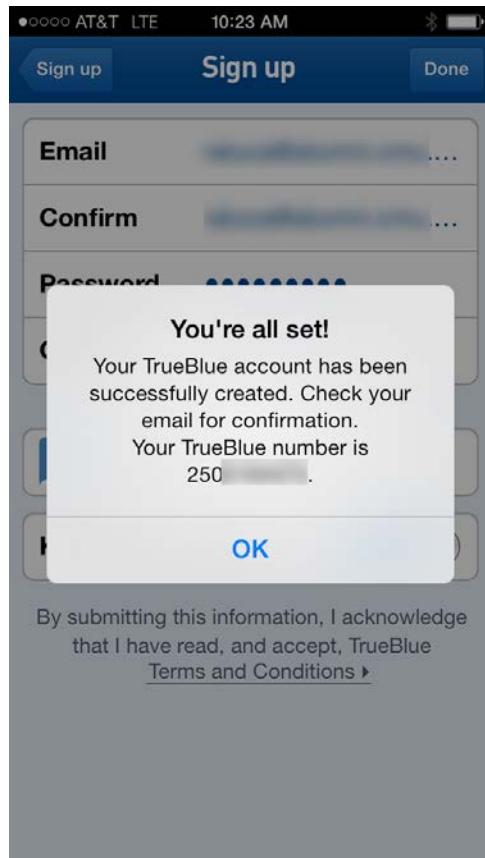
You can use accordions to expand instructions or clarifications in situ without cluttering unnecessarily the design for those people who don't need them (like Sephora does in the example below).



Interflora.co.uk: Instructions for setting up the password were displayed in an alert box. Users had to memorize all the password constraints, dismiss the alert, and then create their password.



Sephora for iPhone: The information regarding the credit-card security code was displayed in situ once the *i* icon was pressed.

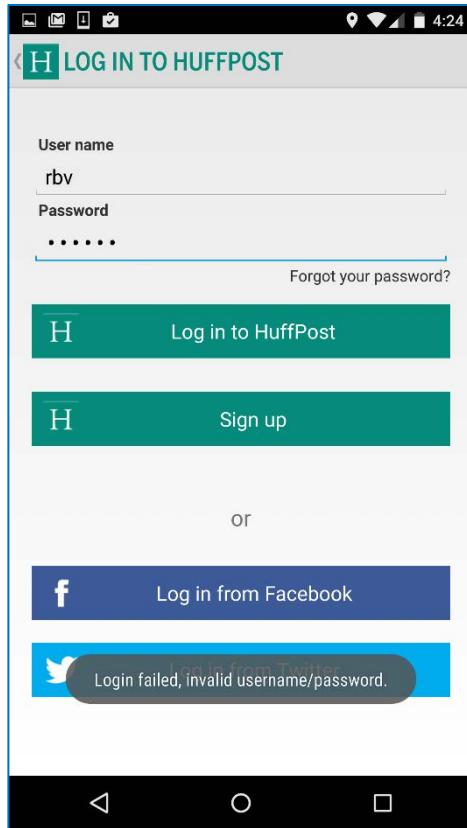


JetBlue for iPhone displayed the new frequent-flier number in an alertbox; users could not refer back to that number or copy and paste it.

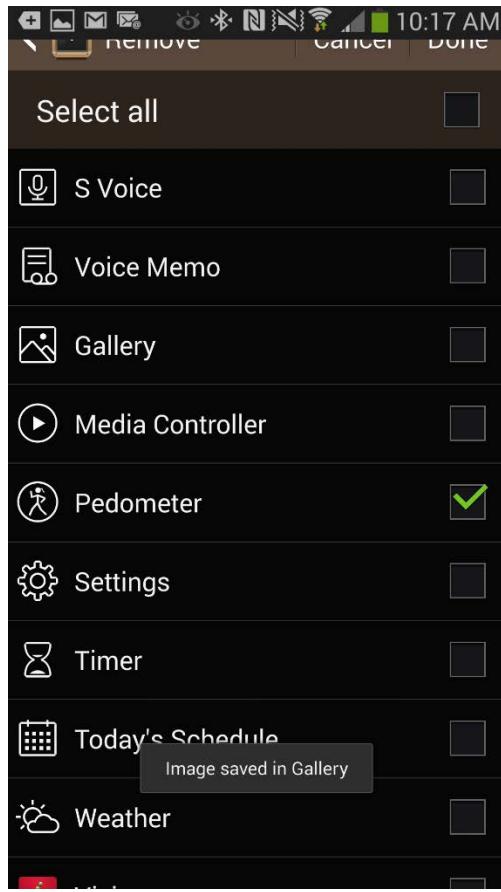
117. Use toasts only for feedback that is not crucial; do not use them for error messages.

Some apps use **toasts** — alerts or messages that fade away after a while — to report errors in forms. We recommend against this practice for three reasons: (1) sometimes users do not notice these toasts; (2) even if they notice them, users might not get enough time to read the message; (3) as with the other alert boxes, the information in the toast needs to be memorized in order to fix the error.

Toasts are nonintrusive, however, and can be instrumental for delivering subtle, noncrucial feedback.



Huffington Post for Android used a toast to indicate that the login failed. Due to its position (far away from the *Log in to HuffPost* button) and to the fact that it disappeared after a few seconds, such a toast ran a high risk of being unnoticed by users.

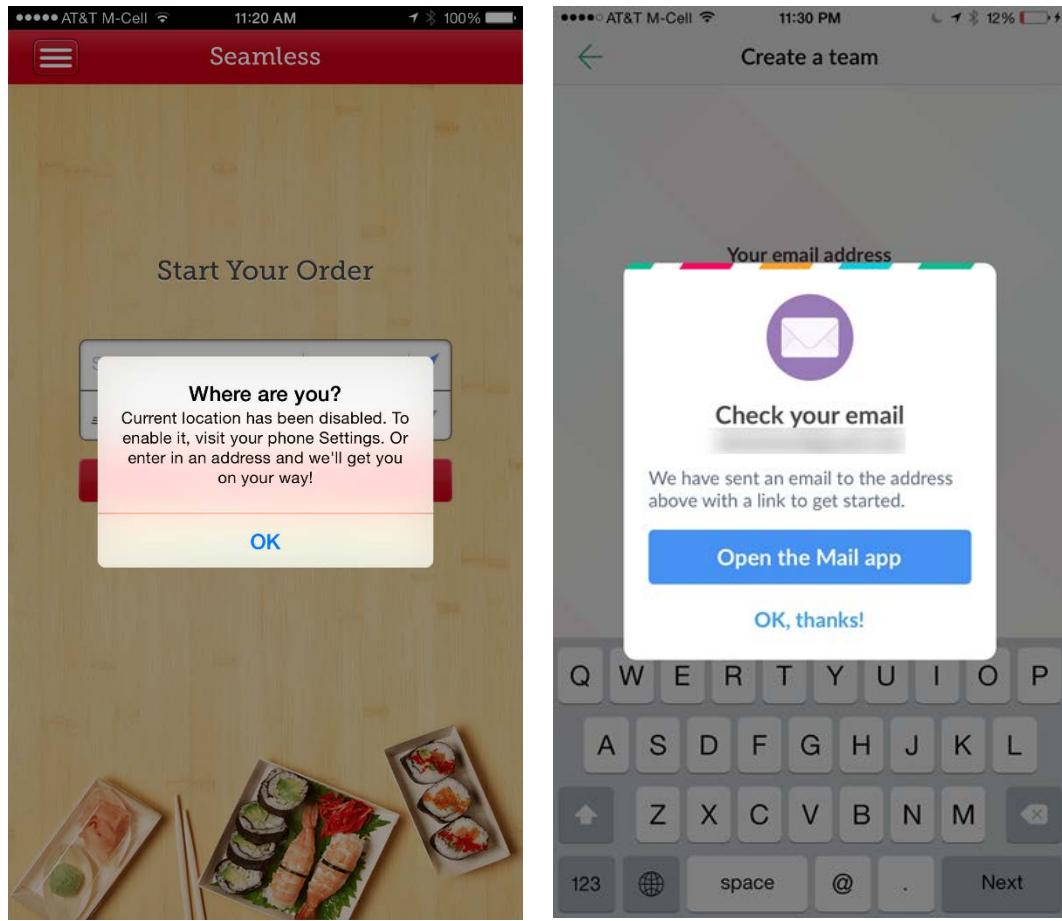


The toast used to indicate that a screenshot has been saved was small, and hard to notice; however, given that this toast was not the only feedback received by the user, it was entirely appropriate.

118. When telling users how to fix a problem, always offer them a link to where they can do it.

When an application reports a problem to users and tells them to how to fix it, the instructions should be really easy to follow and remember. The app should lead users by hand as much as possible.

Thus, if an app complains that it does not have permission to use location services or some other phone feature, instead of telling users how to turn that feature on, it should offer them an easy way to turn it on. For instance, it could include a button that turns on location services right away.



Instead of telling users where to go and forcing them to remember the steps they must take (as Seamless for iPhone did, in the left screenshot), offer a link to the page where they must go to fix the problem (like Slack for iPhone, on the right).

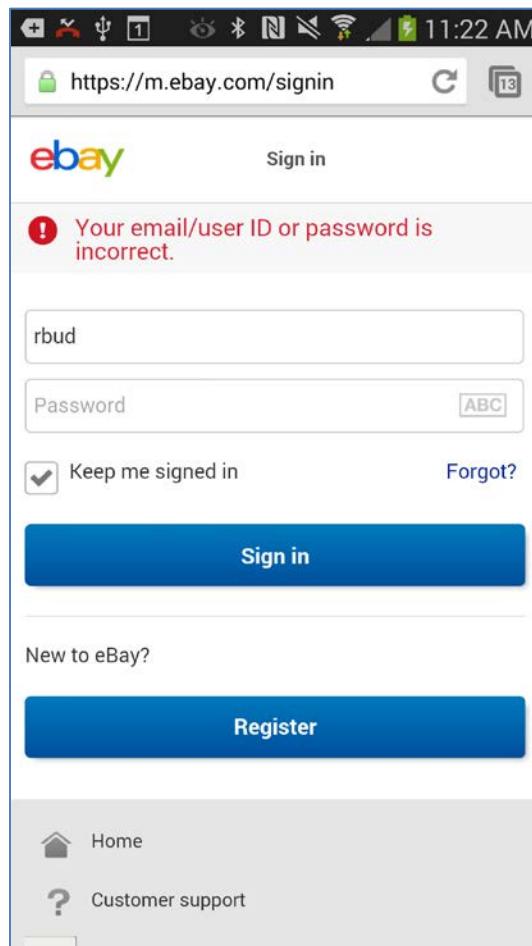
119. In a form, mark fields that need to be corrected to signal an error.

When users have made a mistake in a form, the textbox that needs to be corrected has to be clearly marked. Users should be taken directly to the field that they need to change. Otherwise, if the error is displayed only at the top of the screen, user may not understand or remember precisely where the error was and might even try to fix a correct entry.

A user who was looking to check a flight status on United's website got an error regarding the *From* field. Unfortunately, the textbox that she needed to change was not marked, so she modified the *To* field instead. Users often do not read the text of the error message in detail, and if they see that an error has occurred, they will make their best guess as to what the error was.

One of our participants was unsuccessful when she tried to log in on eBay's mobile site. She got an error message, but no incorrect field was marked, and the message did not point out which of the fields (username or password) needed to be changed. Our user kept changing the password, although there

was an error in the username. It is possible that the designers of the webpage chose not to signal where the mistake was for security purposes; however, the login process on a mobile device is so tedious, that, for many sites, making it even more so in the name of security is not worth it.



Logging in on eBay.com: the error message did not indicate which field needed to be changed, nor was the field marked on the screen.

If an error occurs in a longer form, it's better to show the error message right next to the invalid field rather than at the top of the form. The users should be shown the exact place where the first error occurred, to help them easily locate the error and also see a description of the problem in front of their eyes, as they are correcting their input.

●●●○○ AT&T M-Cell 8:09 PM

Cancel NEW ADDRESS Done

Please enter your first name
 Please enter your last name
 Please enter your street address
 Please enter your city
 Please select your state
 Please enter your phone number

Add address from Contacts

FIRST NAME	▲ Required
LAST NAME	▲ Required
STREET ADDRESS	▲ Required
ADDRESS LINE 2	Optional
ZIP	94539
CITY	Fremont

●●●○○ AT&T M-Cell 9:03 PM

Back Shop In Safari

TEAVANA STORE LOCATOR MY BAG (2)

Checkout BILLING ADDRESS

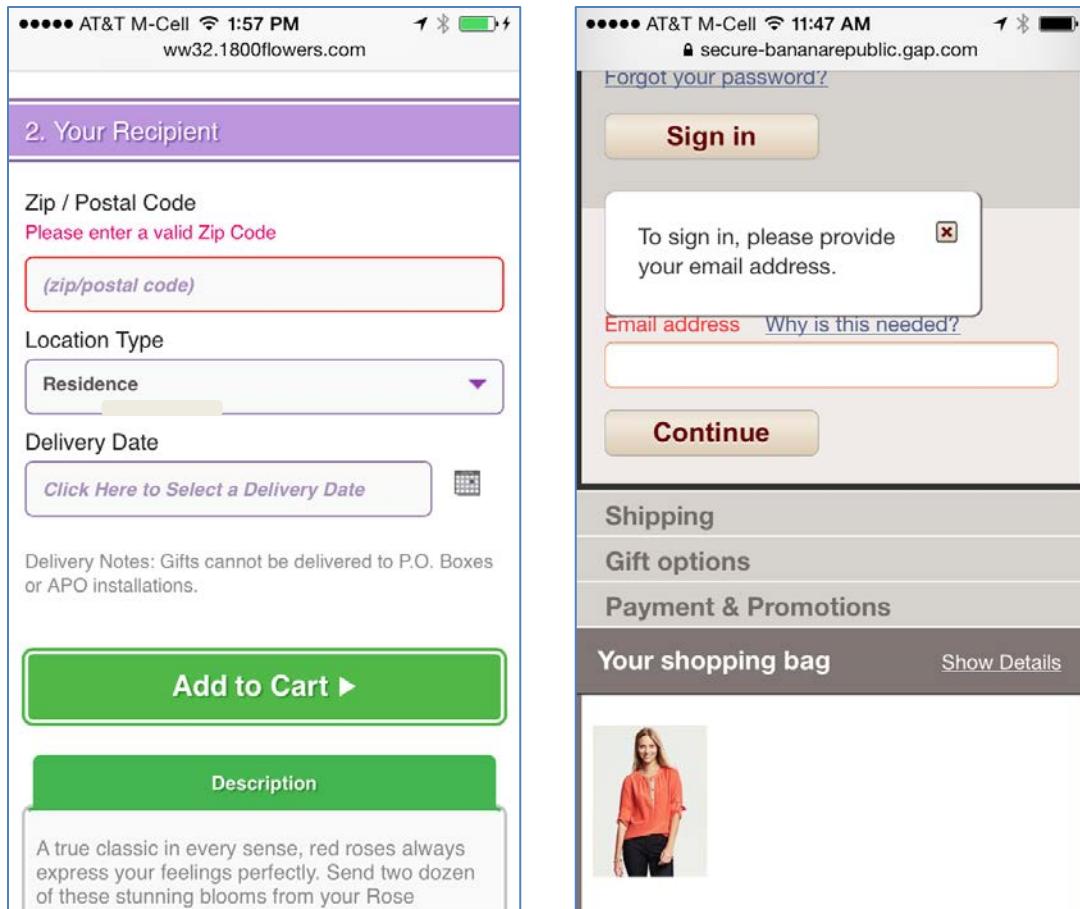
Already have an Account
 Please correct the following errors
 First name can only be up to 100 characters
 Last name can only be up to 100 characters

Email Rbudi@hotmail.com

Password

< Tea Timer Tea Blender Stores Shop

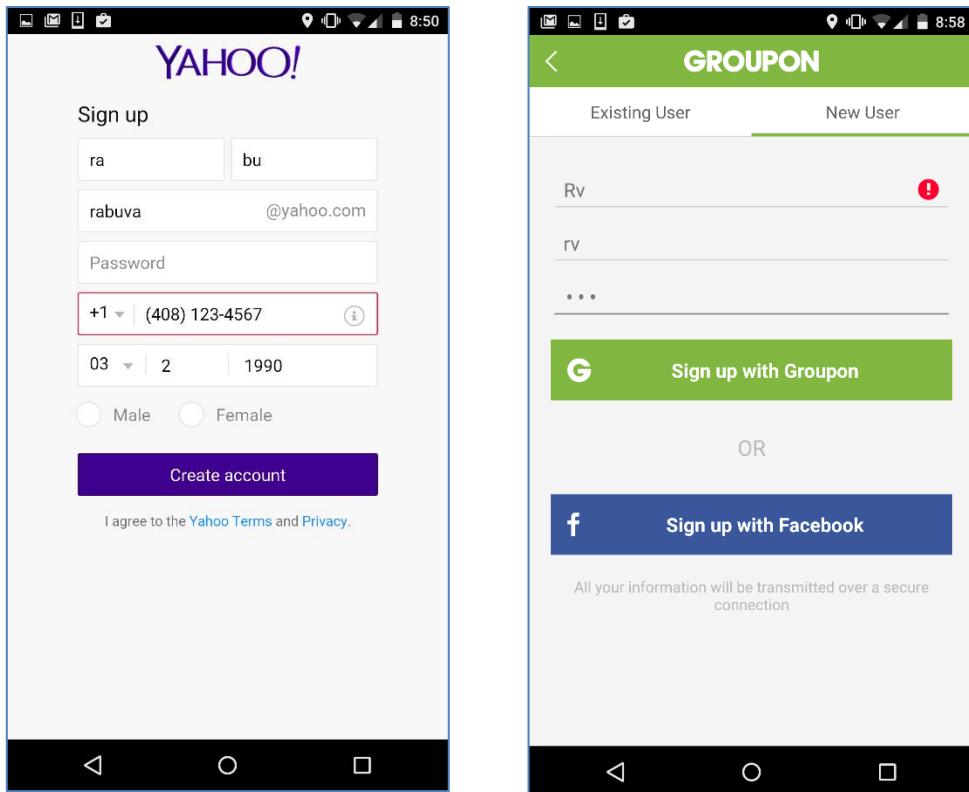
Both these examples show error messages placed at the top of the form, instead of directly above the fields that caused the errors. (Left: Sephora for iPhone; Right: Teavana for iPhone).



The errors were correctly marked in situ on both 1800flowers.com (left) and BananaRepublic.com (right); the user was taken to where the error occurred.

120. In a form always show the error message. Do not hide the error message and expect the users to uncover it.

Instead of showing users what the problem was upfront, some apps only indicated where the error was and exposed the error message once the user took an action (for instance, tapped in the highlighted field). Unfortunately, this is suboptimal for two reasons: (1) the signaling may be too subtle and users may not think to tap in the marked field; and (2) the interaction cost is increased: users must think to tap and tap in the fields to expose the error message.



Yahoo! Weather (left) and Groupon (right) for Android only marked the fields that caused an error, without showing any error message. To see the error message, users had to tap the field (Yahoo! Weather) or the little ! icon (Groupon).

121. An error message should tell users (1) what the problem is, and (2) what they can do to fix it.

We often see error messages that people do not understand or ignore. In order to be noticed, error messages need to stand out and be presented in a simple language that tells the user where the error happened (browser, website, application level, etc.), what does not work, and what the user needs to do.

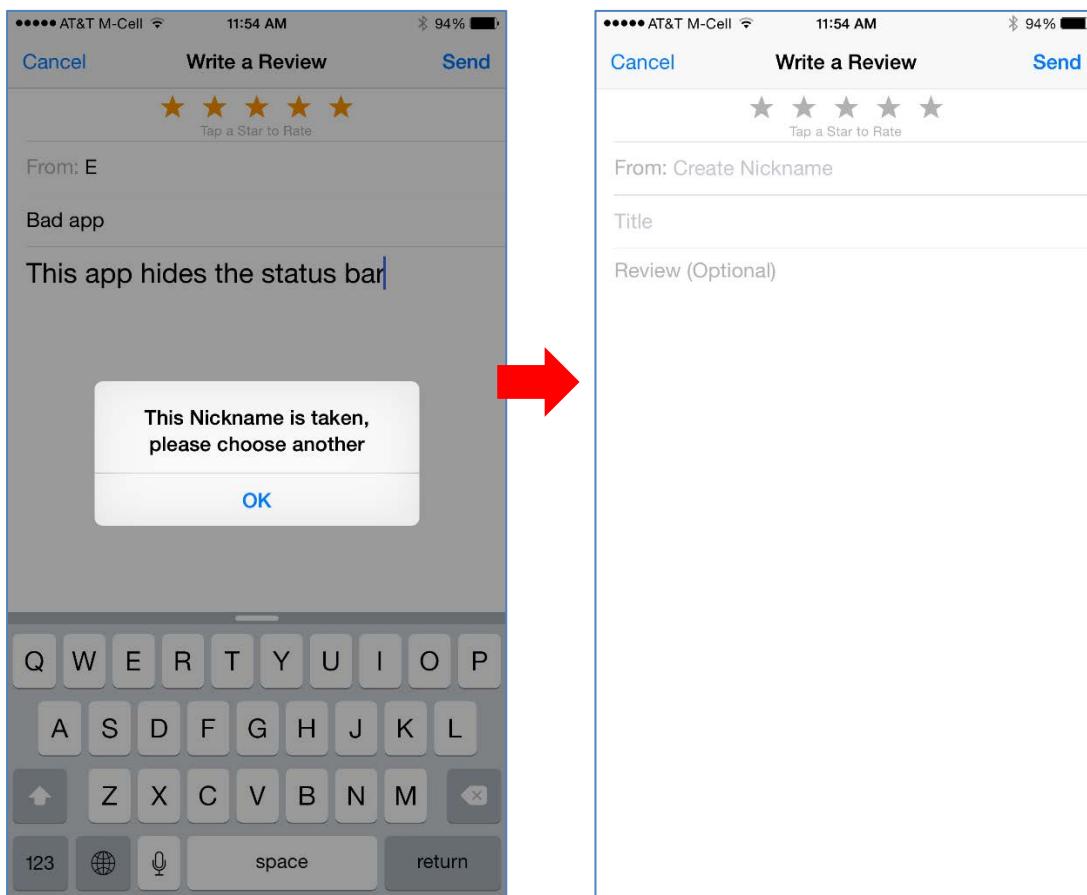
We've seen many users read an error message and then simply reload the page. When we ask them what they think the error means, they usually just raise their shoulders and say that, when they get such errors, it usually works if they try a second time.

122. After reporting an error, return to the state before the error.

Always respect the work that the user has put into filling in a form and, if an error is reported, keep all the information that users have entered and show it to them so that they can correct it.

One of our participants was trying to review an iPhone app in Apple's App Store. To do so, he had to select a username. He picked his name and entered a long review (he really liked the app). However, after he clicked "Send", he

got an error message, telling him that the username was already in use. Alas, when he tried to change it, he discovered that his review was lost and that he was supposed to type it again.

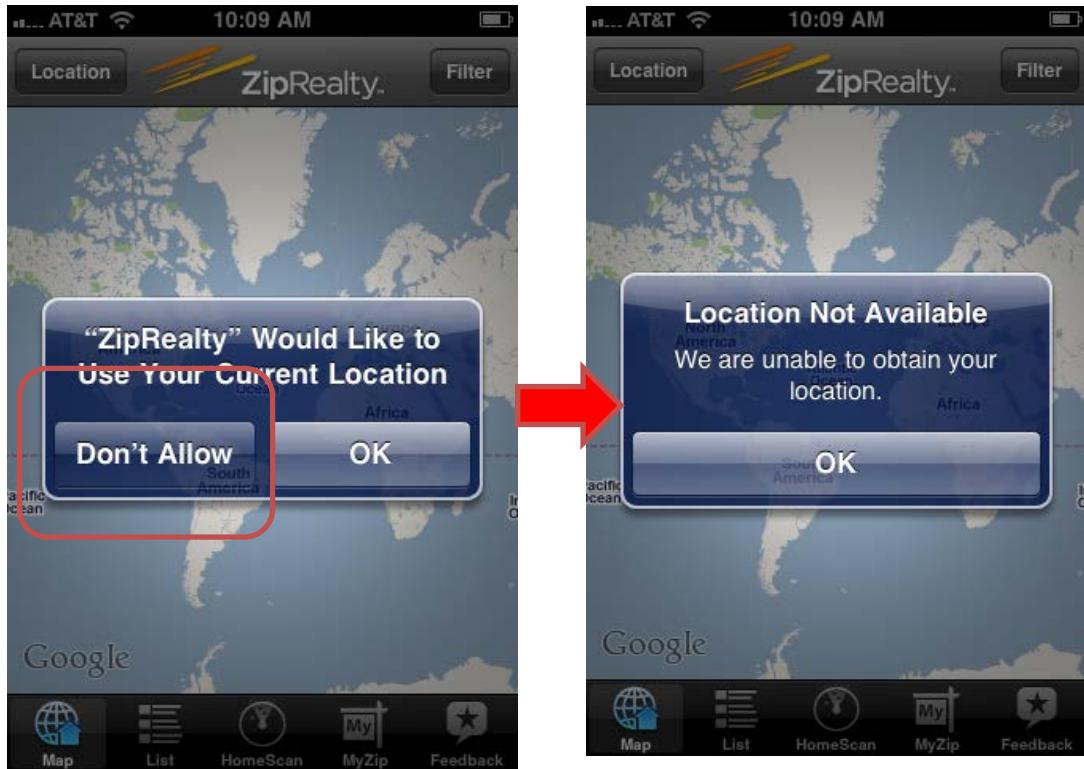


App Store for iPhone: When an app review generated an error, the entire review was lost and the user was supposed to enter it again.

123. Don't report an error regarding a feature that is not in use.

Users' mental model of an app is often very simple and does not match that of a developer. When users do not use an app feature such as current location or camera input, any error regarding that feature looks unjustified. (Luckily, this problem seems to have become less frequent in recent years.)

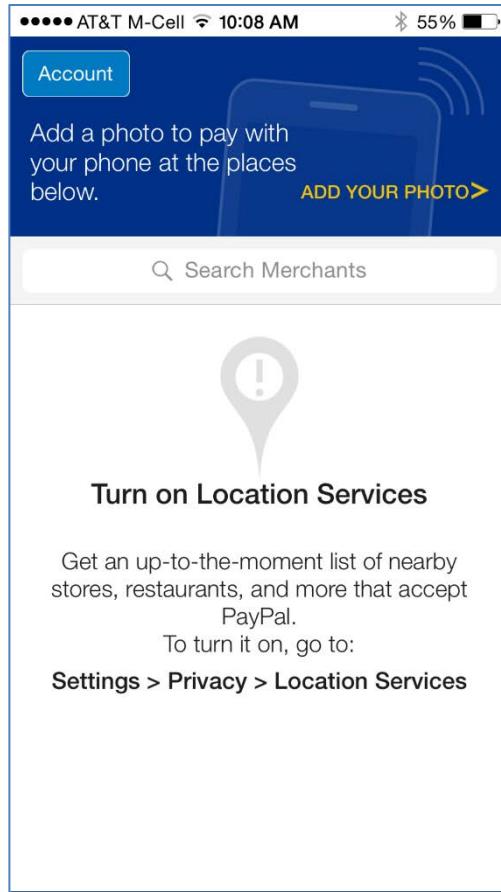
In the example below, the participant denied the use of the current location. The app ZipRealty reported an error that the participant considered groundless: he was not interested in searching for houses around the current location, but he rather wanted to search in a specific town.



Zip Realty app for iPhone : The error reported by the app felt unjustified once the user had denied the use of the current location.

Redfin ran into a similar problem. A user was trying to find houses for sale in Palo Alto, so she denied the use of the current location (which was in a different city). However, throughout the session she kept getting errors reporting the inability to use the current location. She was puzzled by this error, since she felt that the app should not need her current location in order to find houses in a completely different one.

More recently, PayPal did not allow users to search for merchants that accept PayPal unless they gave permission for the use of the current location. This permission should have been unnecessary for users who wanted to search in a location other than their current one.

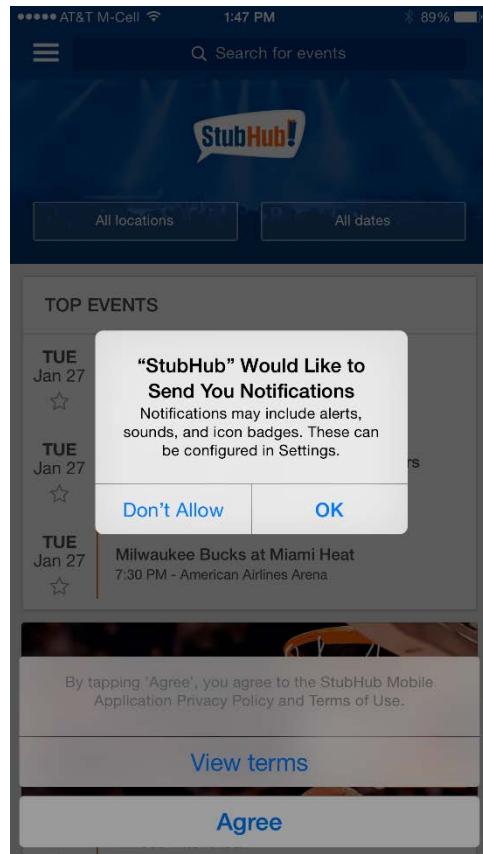


PayPal for iPhone: Users could not search for merchants if they did not want to use their current location.

NOTIFICATIONS

- 124. [App] Do not ask people to accept push notifications the first time they launch an app.**

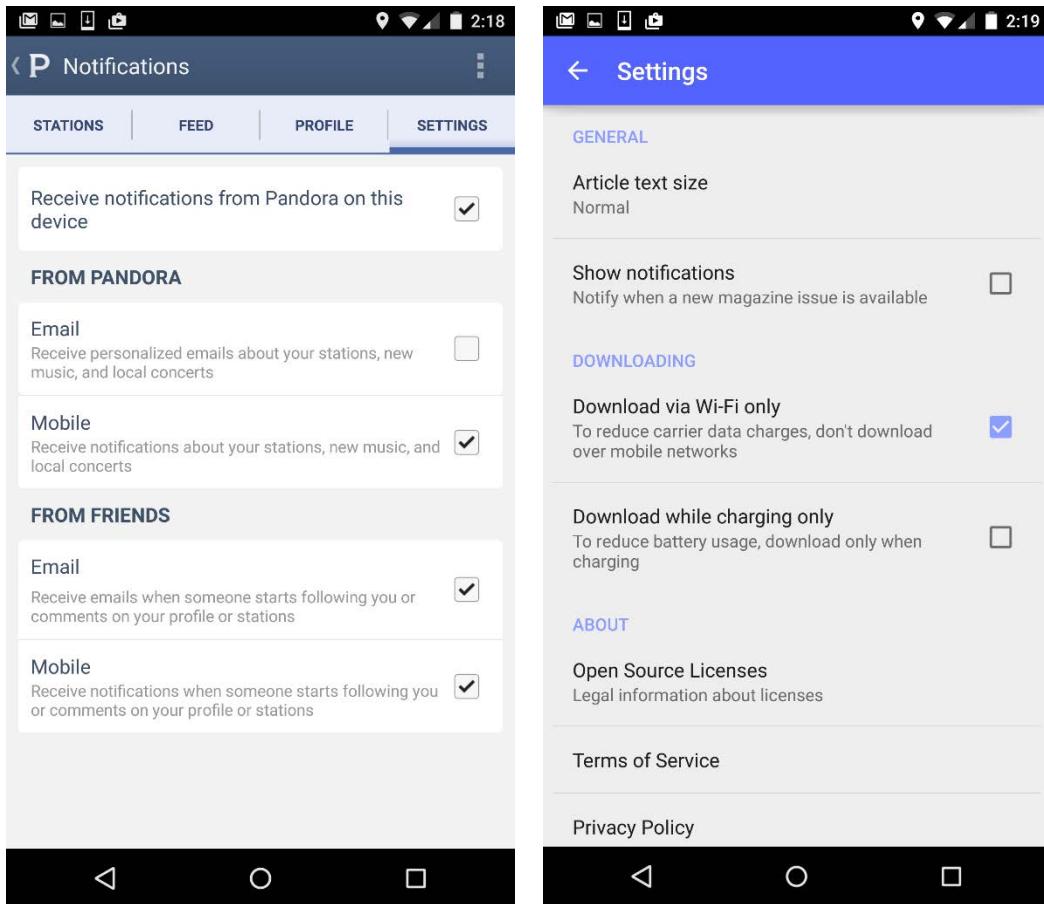
When users have just downloaded an app, they are unlikely to know what the app can offer and what kind of notifications they can receive from the app. It's wrong to have them decide if they want notifications at a time when they have insufficient information to make a decision. In fact, most of the users we've seen denied that request.



StubHub for iPhone asked users to accept push notifications when it was first launched.

125. [App, Android] Do not turn on notifications by default.

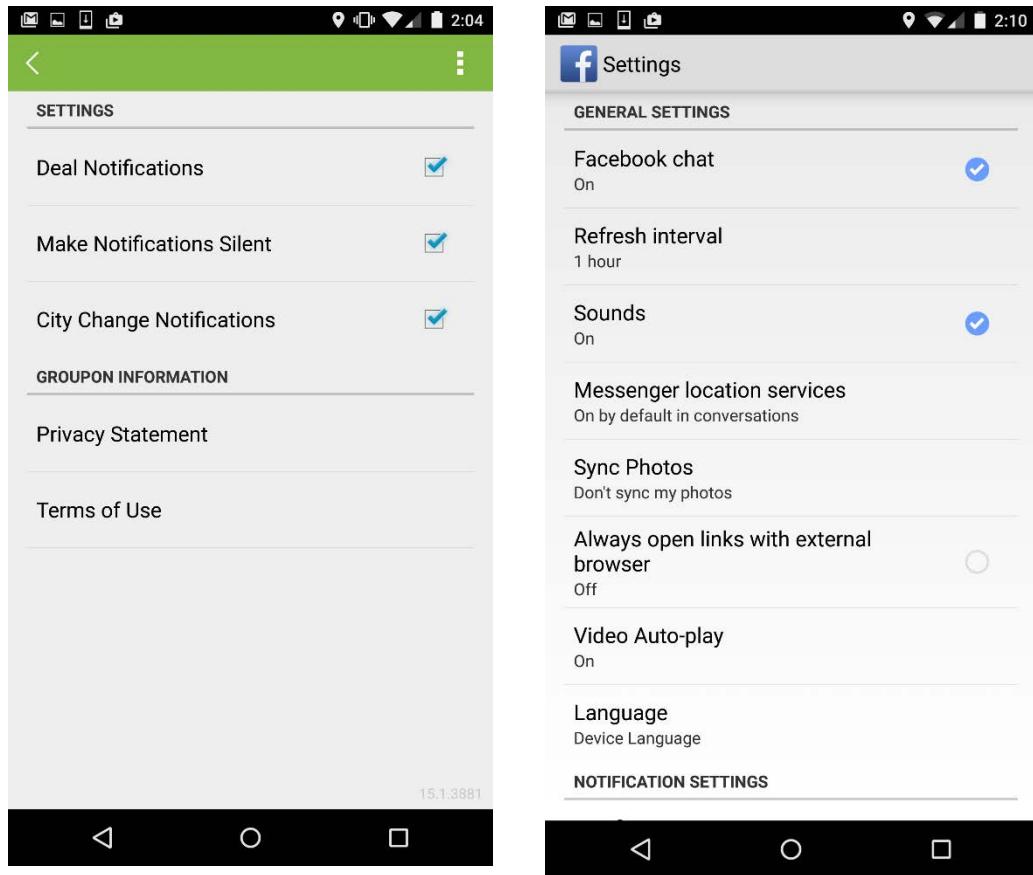
On Android apps can turn on notifications without asking users for permission. The practice is intrusive; instead apps should make sure that users consent explicitly to notifications.



By default Pandora on Android (left) enabled notifications without asking users' permission to do so. In contrast, Play Newsstand (right) did not turn on notifications by default.

126. [App] Avoid turning on sound notifications unless people explicitly agree to them.

If users accept a generic request for push notifications, make sure you don't turn on sounds, too. Enable sound notifications only when users explicitly agree to that (i.e., they say they want sound notifications, not just "push notifications" in general). Most users don't realize that by accepting push notifications they may have turned on sound notifications as well and may not appreciate being waken up in the middle of the night by a new news story or by a new offer. One participant in our studies recalled such an experience and proceeded to tell us that she took great care in identifying the app that had sent the notification and deleting it from her device.

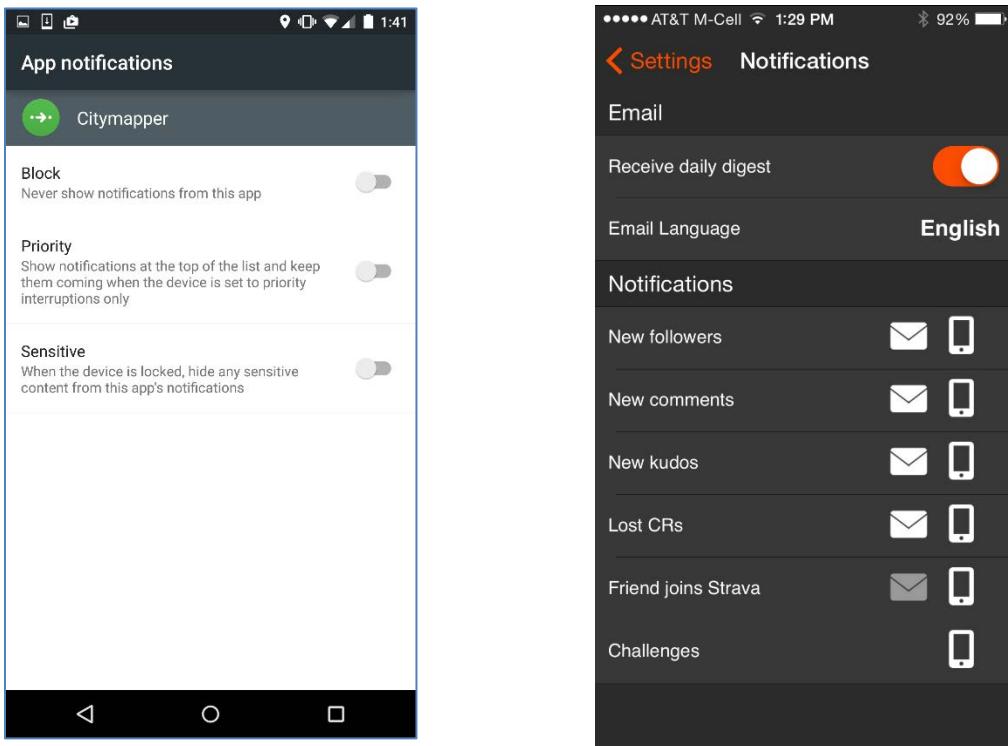


Groupon for Android (left) turned on notifications by default; however, it was considerate enough to make them silent. Facebook for Android (right) turned sounds on by default.

127. [App] Make sure that people can easily turn notifications on or off.

128. [App] Allow users to turn notifications on or off within the app.

Users should not be stuck with receiving notifications from an app if they no longer need them, nor should they have to be denied forever the opportunity of receiving notifications if they said “no” one time. The best place for opting in or out of notifications is the settings section of your app. Don’t make users go to a global notification setting to turn notifications on or off — many users are not aware of that feature of the phone and expect to be able to turn on notifications within the app. Plus, going to the phone’s *Settings* is disruptive (as it requires users to change contexts) and has a high interaction cost.

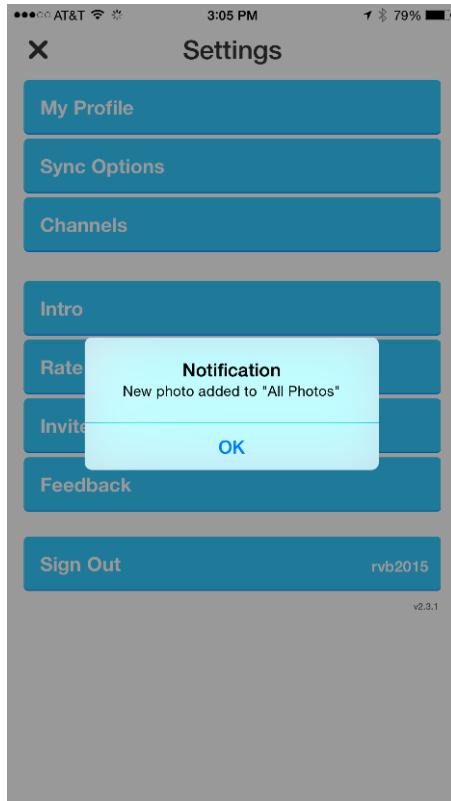


In Citymapper for Android notifications could only be controlled from the global notification setting in the Settings app. Notifications in Strava for iPhone could be turned on or off within the *Settings* section of the app.

129. [App] Do not send notifications in burst.

If you have more than 5 notifications that you need to send at once, combine them into a single notification.

IFTTT, an app that allows users to create “recipes” (macro-like operations that can be run if certain conditions are met), had an option to send a notification each time a recipe was applied. We created a recipe to upload any new photo taken with the phone camera to Dropbox. However, IFTTT cannot run at all times in the background on iOS, so it did not apply this rule until we went back and used it again. At that time there were already many photos to be uploaded, and as a result the app sent a flurry of notifications, one for every single photo.



IFTTT for iOS sent many identical notifications at once, overwhelming the user.

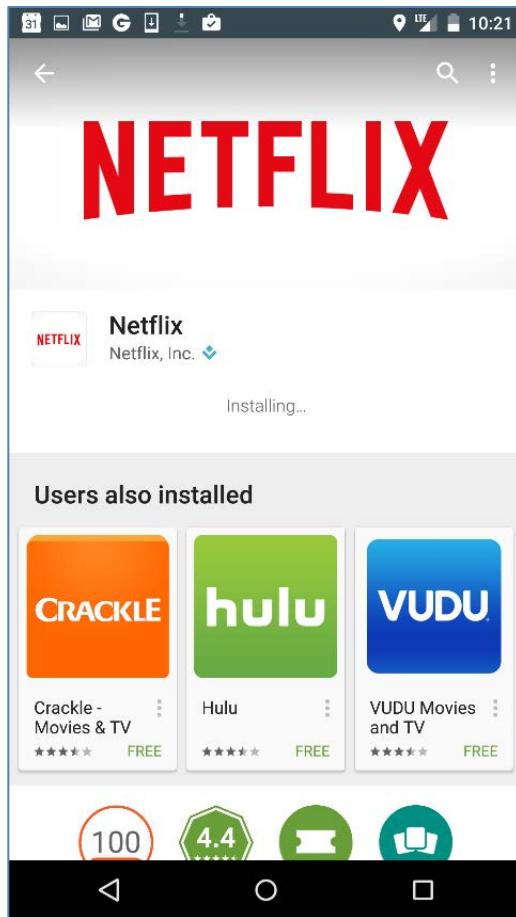
PROGRESS INDICATORS

The main job of progress indicators is to tell users that something's going on and that the app is working. When users don't see a progress indicator, they have no way of knowing whether the app is stuck, whether there's a network or a phone problem (and they'd be better off postponing the task), or whether the app is making progress⁴⁶. Progress indicators also give users something to watch while waiting for the system to update state.

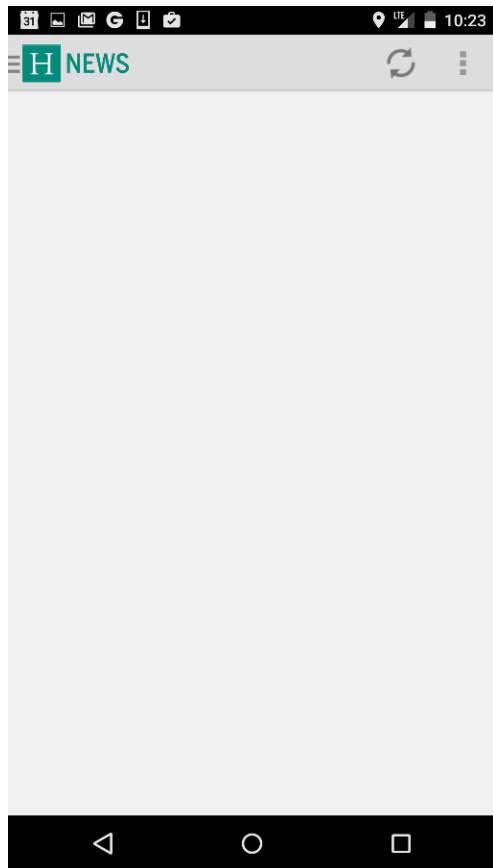
130. Do not use static progress indicators.

Static progress indicators communicate very little information (essentially just that the system has received your command); they should not be used, as often they give the impression that the system is stuck and does not progress.

⁴⁶ For more information on progress indicators in general (not specific to mobile), Katie Sherwin. "Progress indicators make a slow system less insufferable." <http://www.nngroup.com/articles/progress-indicators/>



Google Play used a static progress indicator (the word *Installing...*) to indicate that the app was being installed. Because this operation might take longer depending on the app, it would be better to use a spinning gear or another type of wait animation.



Huffington Post did not use any progress indicator to signal that the content was being loaded. The loading however took long enough to enable us to capture a screenshot of the empty page. A progress indicator would have been appropriate.

131. Use a dynamic progress indicator for every action that takes longer than 1 second.

Mobile users are often impatient and pressured by many variables in their environment. In the absence of a progress indicator people will think that the site or app is broken and might quit the task.

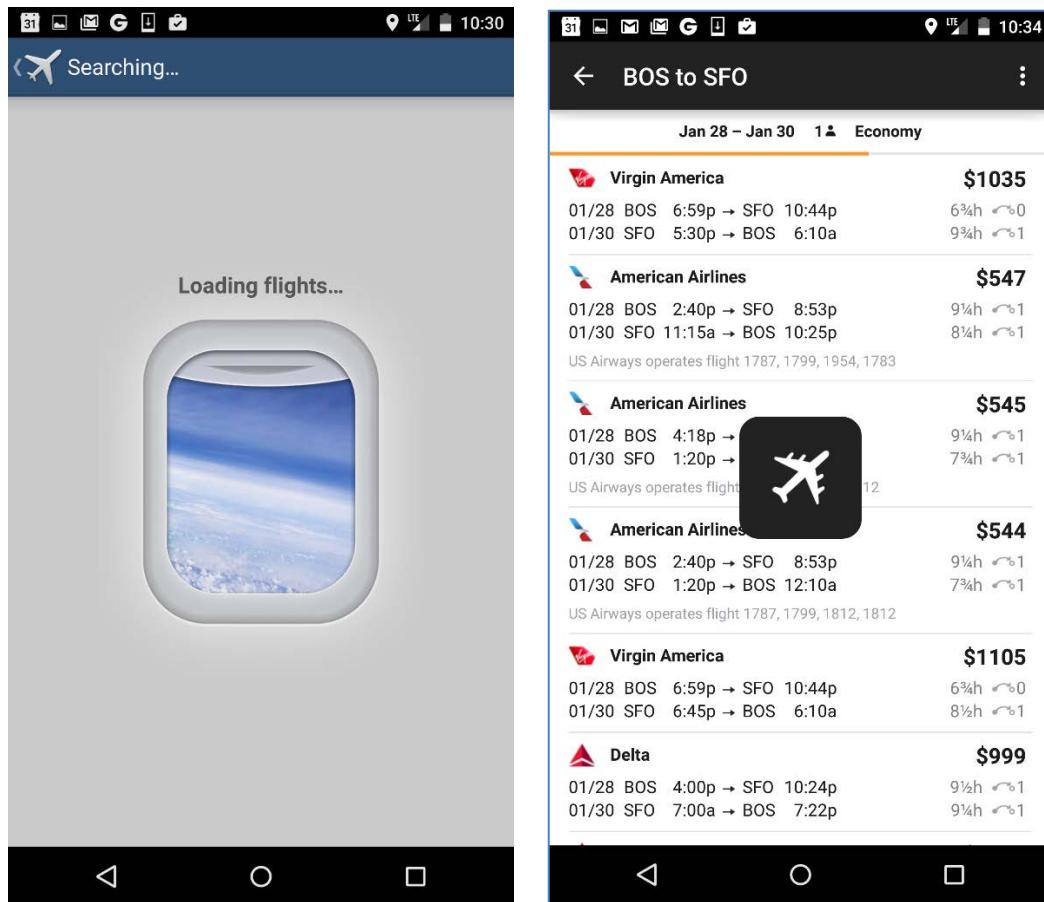
132. Whenever your app is downloading content such as magazine articles, videos, slideshows, show a progress indicator.

Especially since the speed of download can vary across different network conditions, it's important to accommodate that variability by showing a progress indicator, even though under normal conditions the download should be quick.

133. Whenever the action might take more than 5-10 seconds, use a progress bar instead of a spinning gear or other indicators that don't show real-time progression.

Unlike spinning gears or other wait animations which signal only that the system is not stuck, progress bars also give a sense of the progress made and of how much more work needs to be done. Their downside is that they demand more user attention and are more disruptive, so that's why they are generally not needed for quick actions that take around 5-10 seconds.

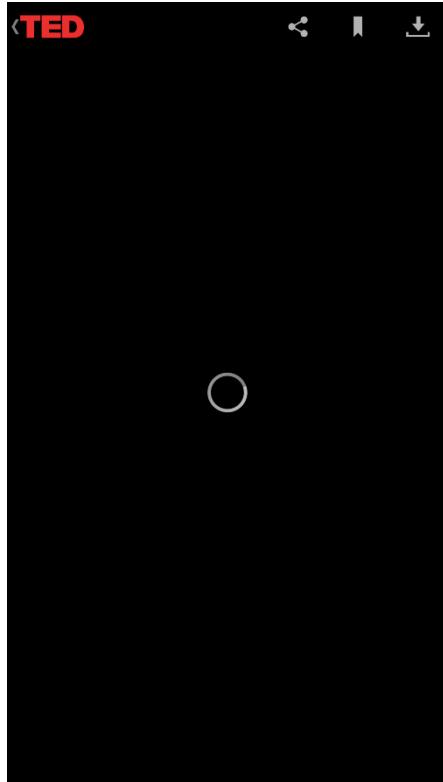
Users also often express their preference for progress bars.



Unlike Expedia for Android (left), which only used a wait animation (and a static progress indicator), Kayak used both a wait animation (the rotating plane) and a progress bar at the top of the page to tell users how much more work needs to be done.

134. For larger downloads (e.g., videos, magazines), give users an estimate of how long the download it's going to take.

Some apps use progress bars only for tasks that they expect will take longer. The problem is that, if the app gets stuck, a task that was supposed to take 2 seconds may take forever, and users may have no idea that there is a problem with the app.



Ted for Android showed only a spinning gear instead of a progress indicator while the video was loading.

Ideally, the progress indicator should tell users how much time has elapsed and, most importantly, how much time they still have to wait. Even an estimation of those amounts can be useful, since it can let users know whether they have the time to wait or not.

ORIENTATION

135. Support both phone orientations (landscape and portrait) whenever possible.

Although we recommend that your app support both the portrait and the landscape orientations, we've seen apps that forced users to change orientations (because, for instance, they did not support the portrait orientation) without causing major discomfort.

136. Don't make users switch between orientations often.



iVerse Comics app for iPhone (older): Comic panels alternated between portrait and landscape, forcing users to turn the phone back and forth.

What does cause major discomfort, however, is when the app forces users to frequently change orientations. For instance, a comic-viewer app (iVerse Comics for iPhone) alternated between portrait panels and landscape panels. Although, technically, the portrait panels were visible in landscape mode (and vice versa), the text was too small, and the user could not read it properly unless the panel orientation matched the phone orientation. She bitterly complained about it:

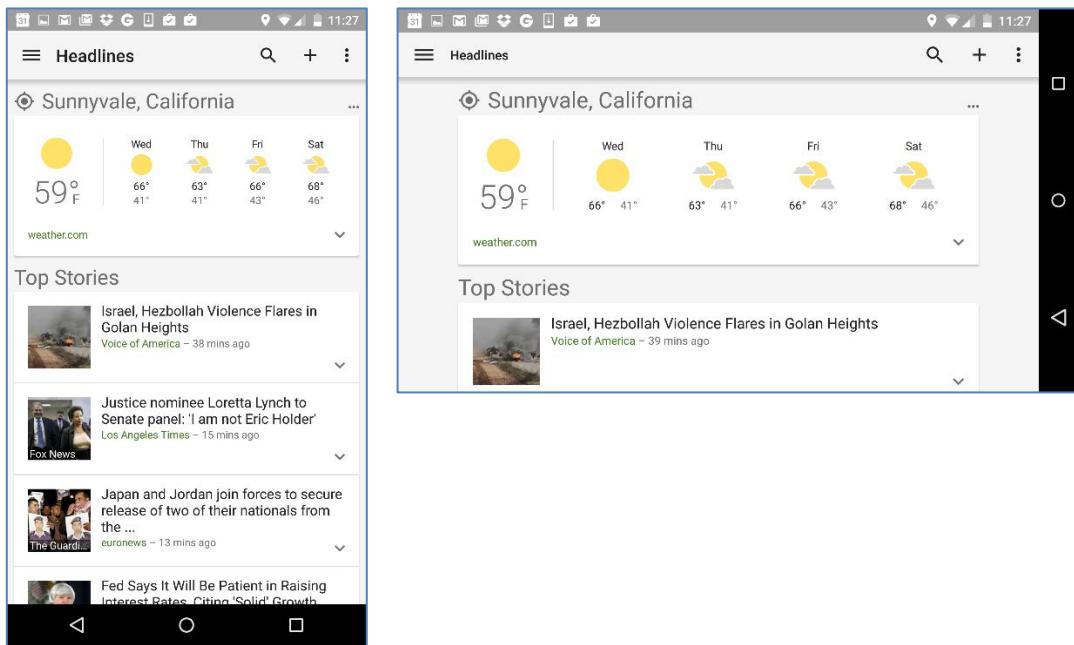
"One thing I do not like is that I turn it sideways so I can read it, and the comics keep changing direction. So then I have to [turn it] back and forth so I can read it".

137. Do not display different content (or different interface elements) in different orientations.

For most apps, however, we don't recommend displaying different content in the two orientations — for at least two reasons. First, users don't naturally change orientations just to check if the interface or the content will be different. And second, because most orientation changes happen when there's a problem, changing content exactly at the moment when users want to inspect it better can cause a lot of frustration.

An example of interface change that causes users trouble is the change of the phone buttons (*Back*, *Home*, *Recent Apps*) on some Android phones. In portrait orientation, these buttons were normally at the bottom of the screen; however,

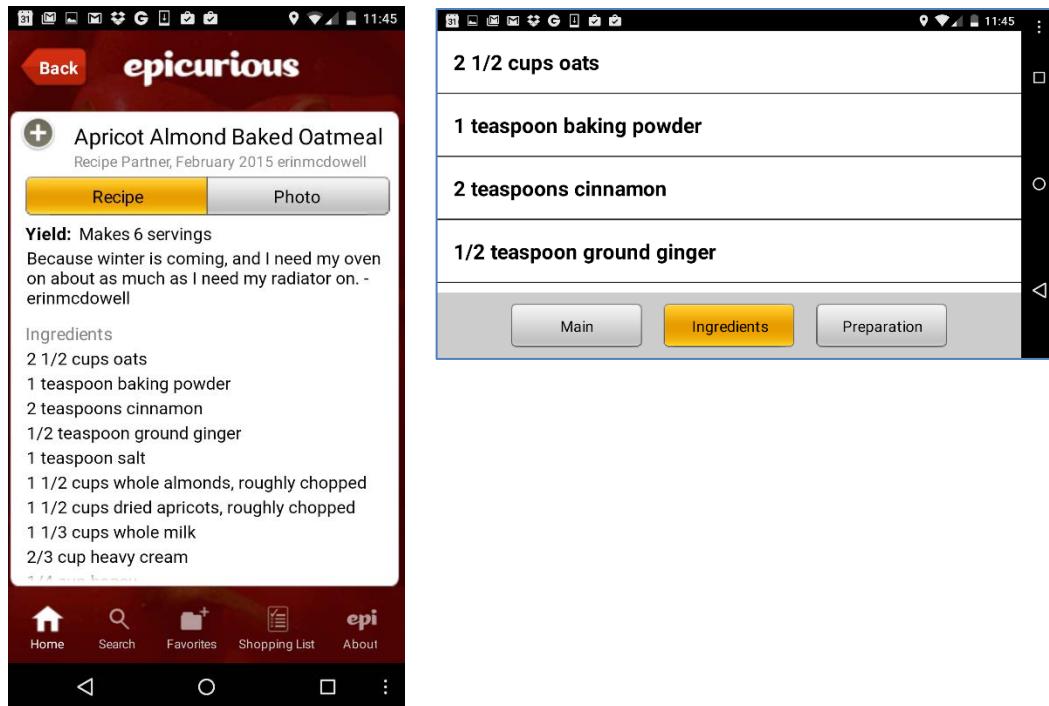
in landscape orientation they migrated to the right edge of the screen. If you think of these buttons as true physical buttons with fixed location, that position makes sense (since the buttons stay in the same position relative to the phone⁴⁷ —it is only the location relative to the user that changes). However, we've seen Android users forgetting about the interface change and struggling to find the *Back* button in landscape mode.



The phone buttons on Android phones were at the bottom of the screen in the portrait orientation, but on the right in landscape.

In Epicurious recipes were displayed differently depending on the orientation. The interface in landscape was not the same with the interface in portrait. While both layouts have their merits, a change of orientation is not the right way to enable users to take advantage of both.

⁴⁷ Technically, they will be on the right only if the phone is rotated counterclockwise; if the rotation was clockwise, the buttons would be on the left.

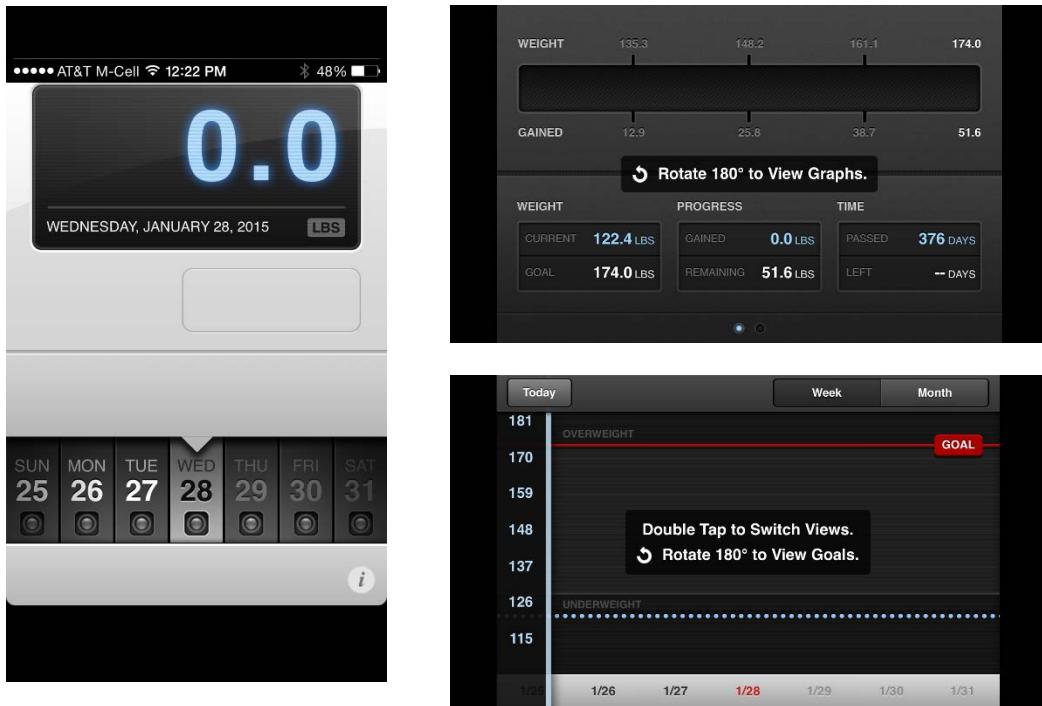


Epicurious for Android: The recipe layout was very different in the two orientations.

138. Do not ascribe different content to the two possible landscape orientations (physical buttons left or right).

We've seen that turning the phone around is not something that people remember to do — they just do it because they're told to or because they need to see something better.

Not only did WeightBot violate guideline 137 and displayed totally different content in landscape and portrait, but it also attached two different meanings to rotating the phone, depending on whether it was done clockwise or counterclockwise. Asking users to memorize what kind of content different landscape orientations may accommodate is confusing and too demanding.



WeightBot for iPhone displayed different content in the portrait and the two landscape orientations.

- 139. You can use the landscape orientation to fit extra detail in a data table, spreadsheet, or other type of graphic.**
- 140. Whenever the landscape orientation is used for extra information, you should let users know (e.g., by displaying a tip or a message).**

Landscape often can fit more detailed information than portrait. While most apps and websites should have consistent content and interface in the two orientations, we do agree that certain pieces of information, such as graphs and complex data, can be displayed in more detail in landscape mode. If that is the case, we recommend that you explicitly tell users that they can get more in the landscape view. The message, however, should be easily noticeable, so that users don't miss it.

CNBC displayed more-in-depth graphs in landscape orientation. An older version of CNBC showed a grey-font message next to the chart. In our testing, most users paid no attention to it. In a newer version, CNBC moved the message in a more salient position. It was better, but still not perfect.



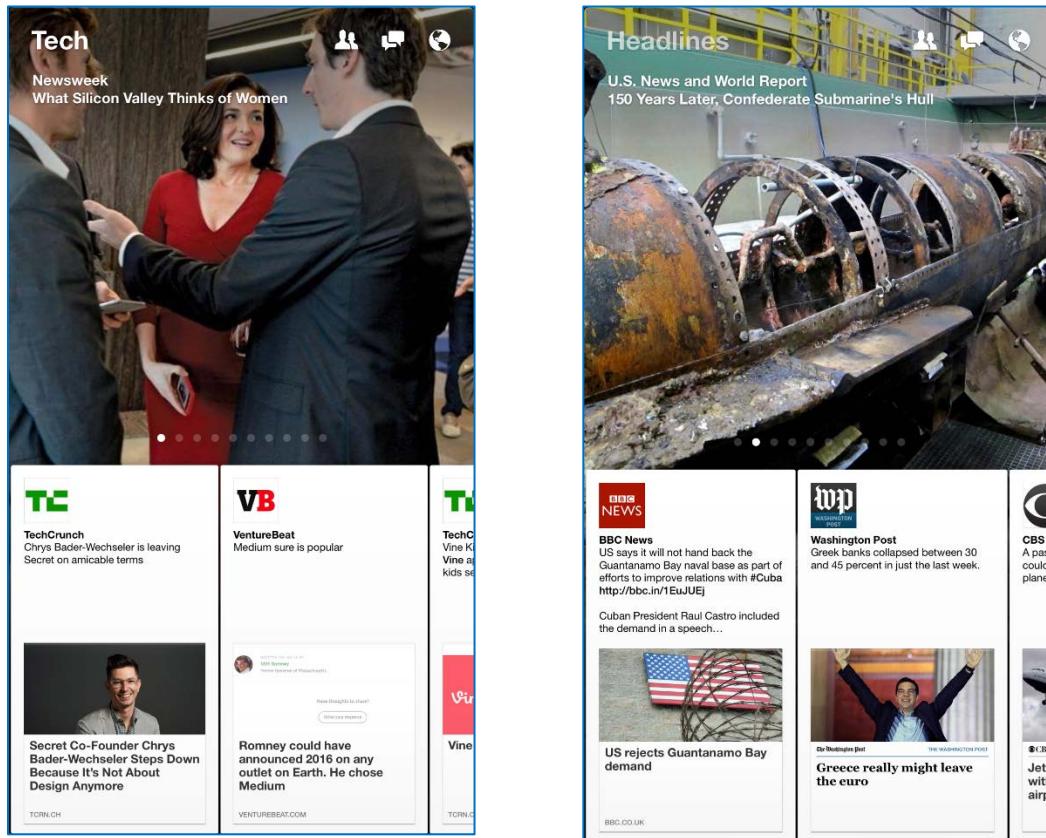
Different ways of signaling extra details for charts in landscape orientation. A prior version CNBC for iPhone (top left) was the least effective, followed by a newer CNBC for iPhone (bottom row).

GESTURES

Using gestures instead of visible interface controls can free up space for content, but it unfortunately has risks. First and foremost, gestures are not easily **discoverable**

and most of them have no natural affordances — it can be quite hard for users to remember to do a certain gesture. Another big problem with gestures is **memorability**: when many are used at the same time, people mix them up and forget what each is supposed to do. And last but not least — gestures can be difficult to produce and replicate reliably.

The horizontal-swipe gesture is popular with designers of touchscreen apps and websites. It is often used for carousels and **deck-of-card navigation** (which refers to a special type of carousel that takes up the whole page; the user is supposed to “flip” through the cards in the carousel), and, while quite discoverable in certain contexts, it is less so in others. Apps sometimes abuse the horizontal swipe by assigning it different meanings, depending on where it is done on the page.



Paper for iOS used a deck-of-card model: users had to “flip” (that is, swipe horizontally) through the different types of news sections.

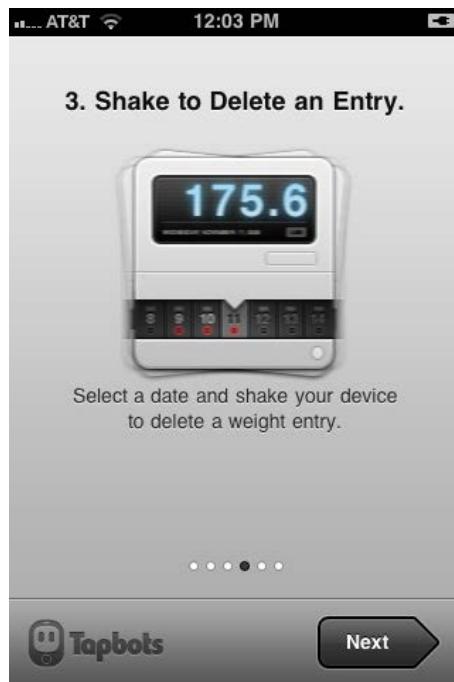
141. Use only the more familiar gestures. Do not rely on users knowing and remembering more complicated gestures.

Although mobile operating systems support many gestures, it doesn't mean that your app needs to take advantage of all them. Although different platforms support different gestures, users across platforms seem to be comfortable and aware of only a subset. In our tests we observed that people rarely employ more than 2-3 more common gestures (such as tap, swipe for horizontal navigation, and pinch open/pinch close for zooming in or out). We

never saw people spontaneously trying other gestures to figure out if the interface supports them.

For instance, shake in iOS can mean undo in applications such as Mail. Yet many well-seasoned iOS users are unaware that they can use this gesture. Similarly, in Google Maps the shake gesture was used for sending feedback to the app designers. Yet, most Google Maps users did not know of that functionality.

And iOS phablet users can get the top part of the screen closer to their thumbs by double tapping the home button; however, we've never observed users do it purposefully.



WeightBot for iPhone used an unfamiliar gesture to delete an entry.

Even when people are instructed about less frequent gestures, they tend to forget about them or about what they are supposed to mean.

142. Multi-finger gestures (other than pinch) should only be used as accelerators for features that can also be accessed in other ways or are not needed by non-expert users.

Pinching with two fingers to zoom in-or-out is a generic gesture known (and frequently used) by virtually all users. Other gestures that require more than one finger should be considered for expert users only, since most users won't know them and won't learn them because they are so different from the single-finger gestures that dominate all other mobile use.

Multi-finger gestures, such as swiping with two fingers moving in parallel, can be used as interface accelerators. That is, they can serve as shortcuts for an expert user to more quickly achieve an operation that can also be executed in a



Sky Guide for iPhone: Dragging a finger on the screen moved the screen in the corresponding direction. This type of gesture had a natural, strong affordance.



Post-it Plus for iPhone is an app for creating and organizing post-it notes. Users could drag the post-it notes around freely.

slightly more tedious way by simpler means that all users will know. See guideline 151 for more on adding redundancy to difficult gestures.

(This is an equivalent to the way desktop applications use command-key shortcuts to allow expert users to bypass the menus. If less-experienced users don't know the required command key, they can simply go to the menus and will still be able to access the feature in question.)

143. Take advantage of the natural affordances for gestures.

Beside swipe, drag around can be also fairly intuitive in a context where you move yourself in reference to a landscape (like in the Sky Guide app, in which you can "move" on the sky map to see different stars and constellations) or where you need to move items around (like in the Post-it Plus example below, where you can position post-it notes around the screen and group them as they wished).

- 144. Use contextual tips to make gestures more discoverable.**
- 145. Use progressive disclosure to make gestures more discoverable and avoid overwhelming people.**

See also our section *Contextual Tips*.

Gently reminding users of the existence of a gesture can ease them into using gestures. But, in order to be effective, such tips must be contextual: instead of presenting them when the app is first launched and when users do not know yet which actions will be useful and which can be safely ignored, present tips *in context* — when users first attempt such an action in the interface. (And if you followed our other guideline about redundancy in the interface, it shouldn't be difficult to guess users' minds.)

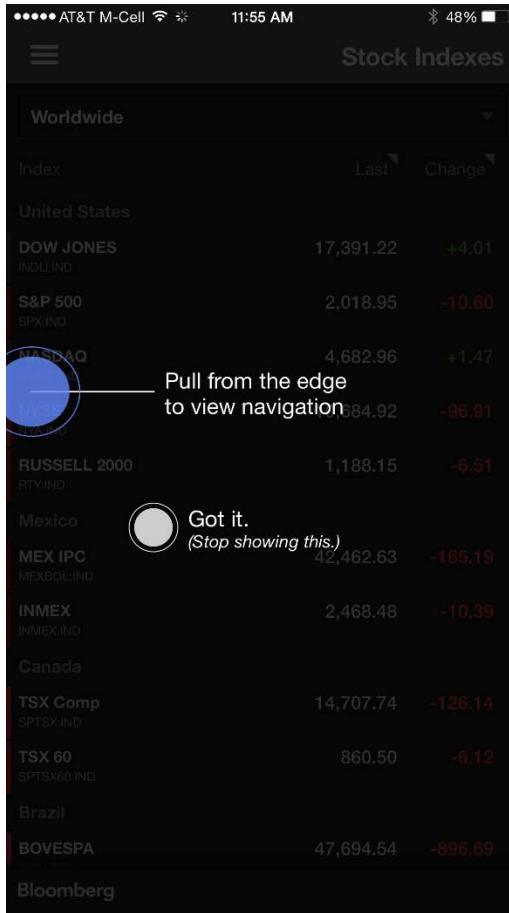
In video games users are able to juggle a plethora of complex gestures and key combinations. The secret ingredients of success in that case are (1) the use of contextual tips or suggestions, and (2) **progressive disclosure**⁴⁸. Progressive disclosure is a concept widely used in human-computer interaction and can be applied to gestural interfaces as well. If gestures are introduced gradually, users are less likely to get confused and mix up the different possible gestures. They also get the opportunity to practice the gesture a little and establish it into their memory before moving on to a next one.

Paper, an app that delivers social news, used audio tips to explain users that they could tilt their phone to "explore" a picture. While the audio modality is not recommended, the app delivered that tip at the right time: not when the user was reading a news article or when the app was first launched, but when she was inspecting a photograph.

⁴⁸ See Jakob Nielsen. "Progressive disclosure."
<http://www.nngroup.com/articles/progressive-disclosure/>



Paper for iPhone presented an audio tip instructing users to tilt their phone in order to "explore" this image.



Bloomberg used a tip to make users aware of a gesture.

- 146. In most situations, users need cues to figure out that they're supposed to use the horizontal-swipe gesture for horizontal navigation.**

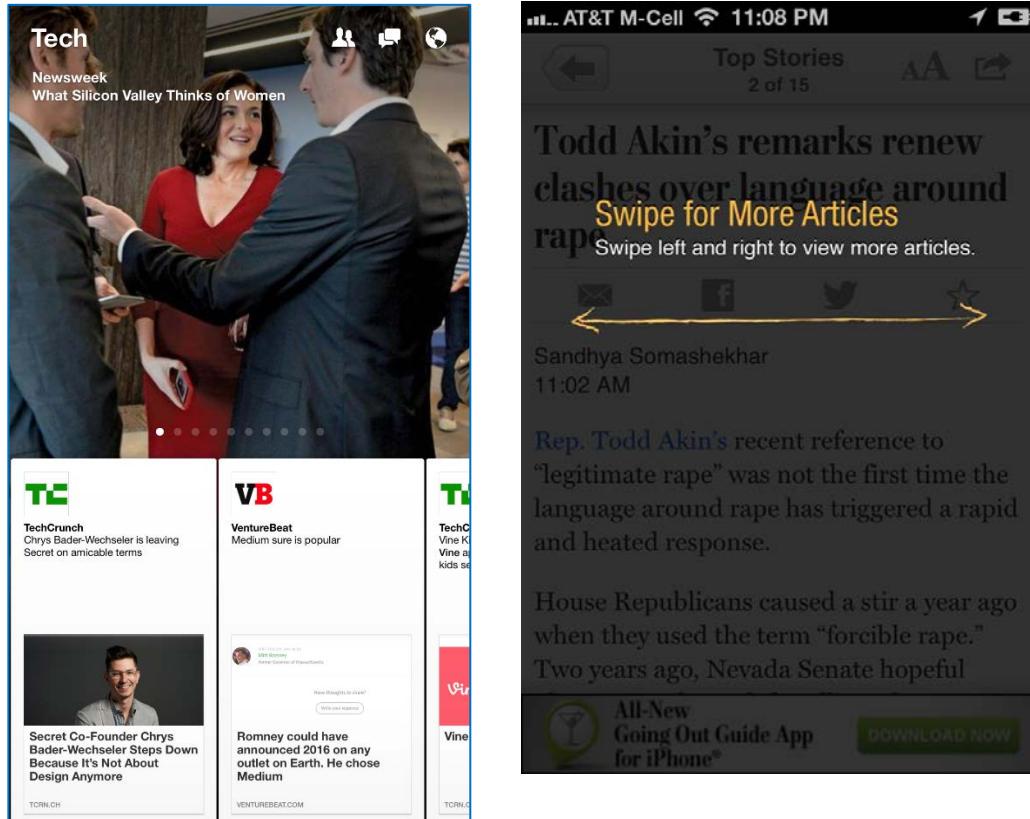
Users are quite familiar with the mental model of a physical book, and whenever they read a magazine or a book on a phone, they discover relatively fast that using the swipe gesture can turn pages. Some users need more time figuring it out, so even there, a quick tip that tells first-time users how to turn the page can be helpful.

In most other situations, the swipe gesture needs some external cues such as an arrow or truncated content to the right. Guidelines 110 and 111 talk about using cues for carousels, so that users know they are supposed to navigate horizontally. The same kinds of cues are instrumental for deck-of-cards.

When we talked about carousels, we said that dots are a weak cue. In spite of being present in the iOS and, more recently, Android design of the app home screen, we find that users don't necessarily notice dots and may ignore content that is on different pages.

Paper for iPhone used the deck-of-card navigation and employed dots to suggest horizontal navigation to users. It also used audio tips when the user

first launched the app (see guideline 144). Similarly, Washington Post tipped users that they could swipe horizontally to view more articles.



Paper (left) and Washington Post (right) for iPhone used tips to tell users that they could flip through different pages of content. Paper also used dots to signal the deck-of-cards navigation.

147. To invite users to scroll down a page, avoid creating false floors.

While horizontal scrolling has become popular since the advent of touch screens, vertical scrolling has been around since the advent of the web. People are certainly used with vertical scrolling. So it may come as a surprise to some of us that there are situations where people do not scroll vertically⁴⁹. Yet, whether people scroll or not is fairly predictable: if the page invites them to scroll and tells a continuous and coherent story, users scroll down. If the page builds “false floors” and the illusion of completeness, then the affordance for scrolling will lack and users will not scroll down.

⁴⁹ More about the fold: Amy Schade. “The Fold Manifesto: Why the Page Fold Still Matters.” <http://www.nngroup.com/articles/page-fold-manifesto/>

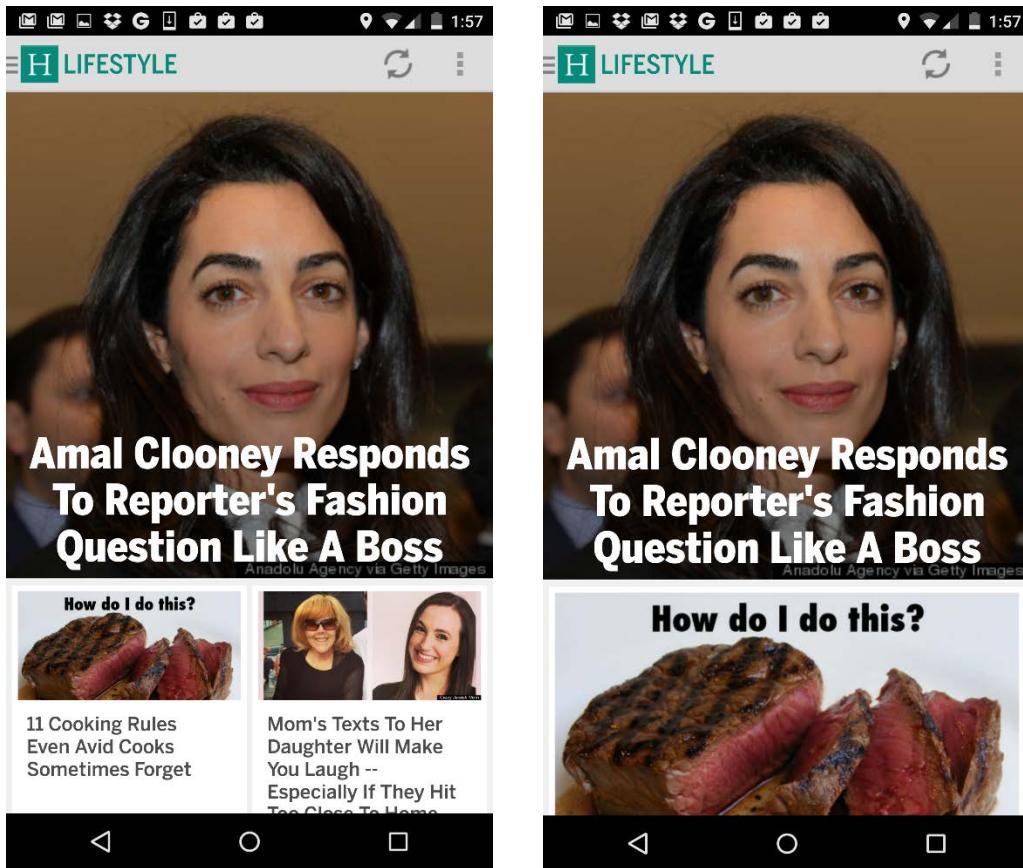


Weather Channel for iPhone created a “false floor” and the illusion that the page was complete and there was no more information below the fold. It thus discouraged users from scrolling down.

148. Do not assign unconventional meanings to standard gestures.

If a gesture has a strong association with an action (e.g., pinch-open for zooming in, swipe for moving through a carousel), many users will have a hard time remembering a different use for that gesture. So creating new, weaker associations with actions that have no relationship with the original one is unlikely to succeed.

Huffington Post attempted to use the pinch-open/pinch-closed gestures for changing the layout of its main pages from “grid” to “list”. Aside from the fact that users are unlikely to need to switch between these two interfaces, having people remember that gesture is unrealistic, even with the little tip that the app displayed when it was first launched.



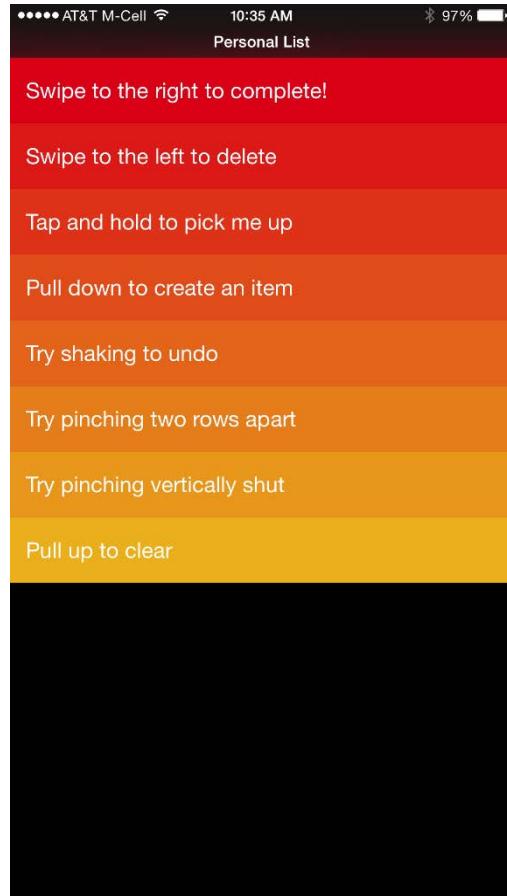
Huffington Post for Android allowed users to switch between two different interfaces using the pinch-open/pinch-closed gesture. This is a gesture normally associated with zooming in/out of a page, and it's unlikely to be remembered by users when used in a nonstandard way.

149. [App] Avoid using many different gestures within the same app.

Because most gestures have few affordances, they also suffer from **low learnability**: they are hard to learn and remember. The more gestures people have to learn, the more likely that they will mix them up or forget some.

That is the primary reason for which apps based entirely on gestures have not been successful (at least among audiences that did not consist of designers or gesture enthusiasts): there was just too much information to remember.

Clear, a to-do-list app with a gesture-only interface and no visible interface elements, caused a lot of grief to our study participants. In spite of trying hard to figure it out, they were not able to complete even the most basic tasks. Participants were overwhelmed and confused by the many different gestures that they had to remember in order to interact with the interface.

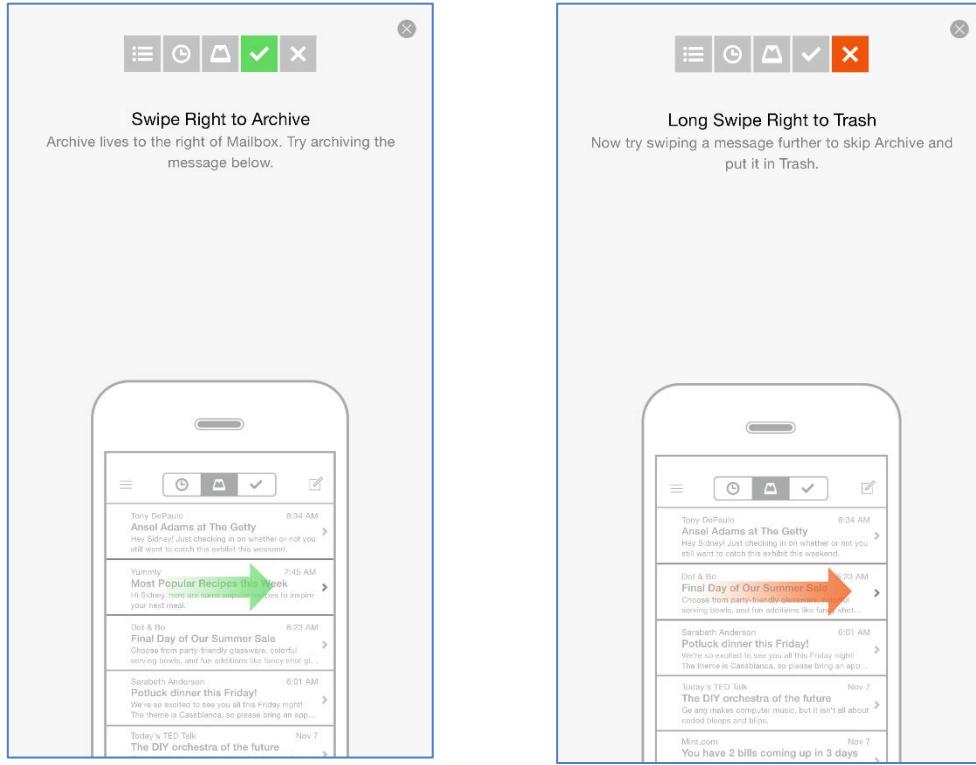


Clear for iPhone used a variety of gestures to compensate for the lack of visual interface elements. These gestures were hard to remember.

150. [App] Avoid using many similar gestures with different functionalities within the same app.

Some gestures are fairly similar, and assigning different actions to them can create difficulty for the users. For instance, (long) swipe and flick (short swipe) are fairly close. When apps use the two gestures to implement different functionality, users can easily produce the wrong action by accident.

Mailbox, a mail app, used a long swipe right for deleting a message and a short swipe right for archiving it. These gestures are too similar — even if people remember which is for what, they may still not be able to reliably produce one over the other.

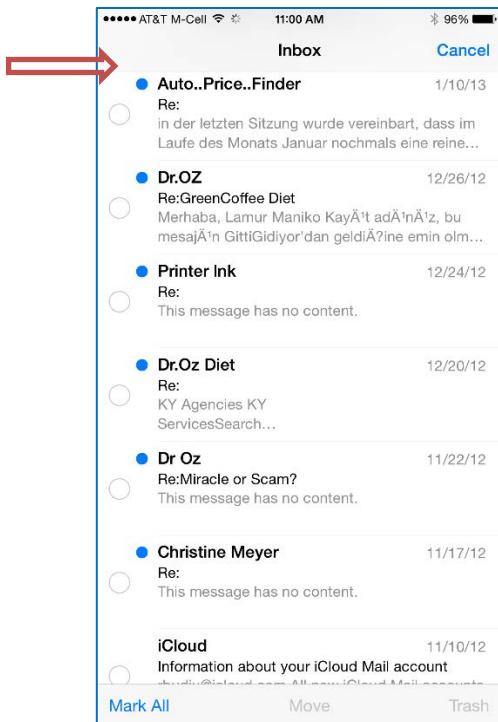
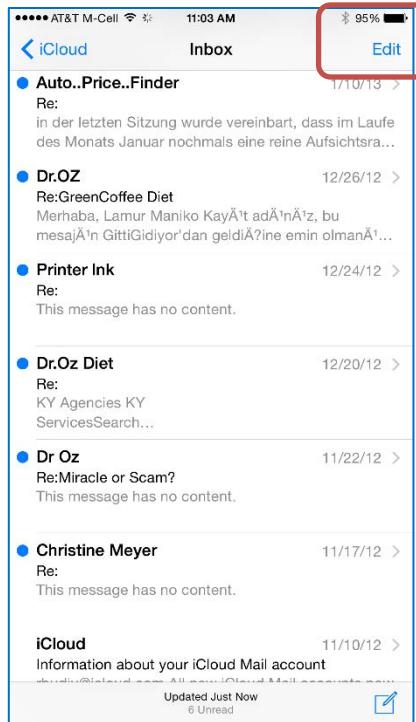
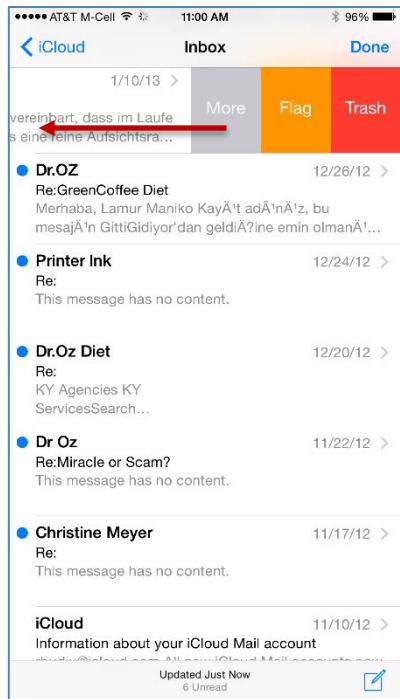


Mailbox for iPhone: The app used two similar gestures for different actions: (1) swipe right to archive a message, and (2) long swipe right was used to delete it.

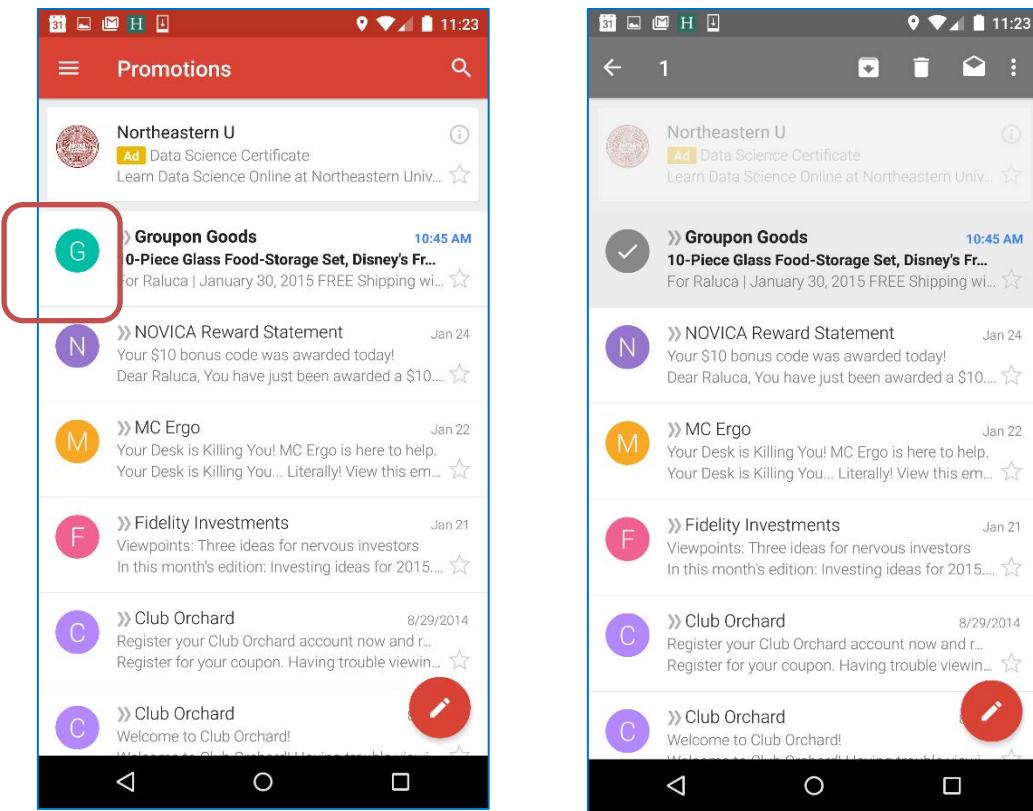
151. If you must use less familiar or less discoverable gestures, add some redundancy in the interface.

If you must use one of the less familiar gestures or you plan on defining new gestures, we recommend that you also allow users to perform the same actions in a different way, preferably by using interface buttons.

For instance, in the iOS Mail app, users could delete items either by swiping over each individual email or by using the edit button at the top of the screen. Similarly, Yowza, a coupon app, allowed people to delete favorite stores with the same swipe gesture, or using the interface. The Gmail app on Android let users delete messages either by checking individual items or by tapping and holding a message (a gesture that has lower discoverability).

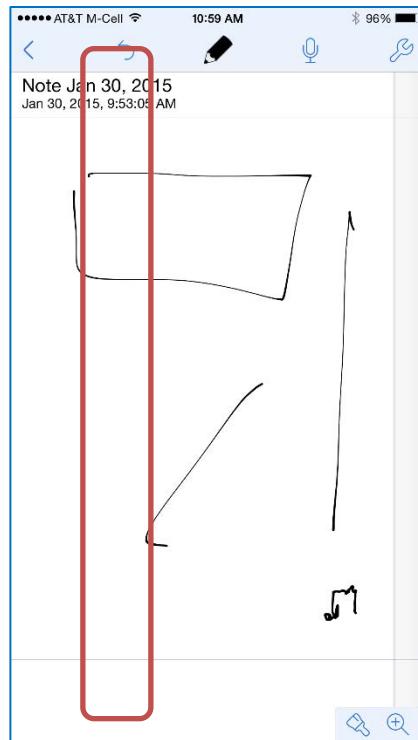


Mail for iPhone: The swipe gesture over any of the individual messages allowed users to delete the message (top), but so did the *Edit* button (bottom).



Gmail for Android also used redundancy for the action of selecting a message: users could either tap the little icon next to the message or tap and hold the message heading. Note however that none of these actions was very easily discoverable.

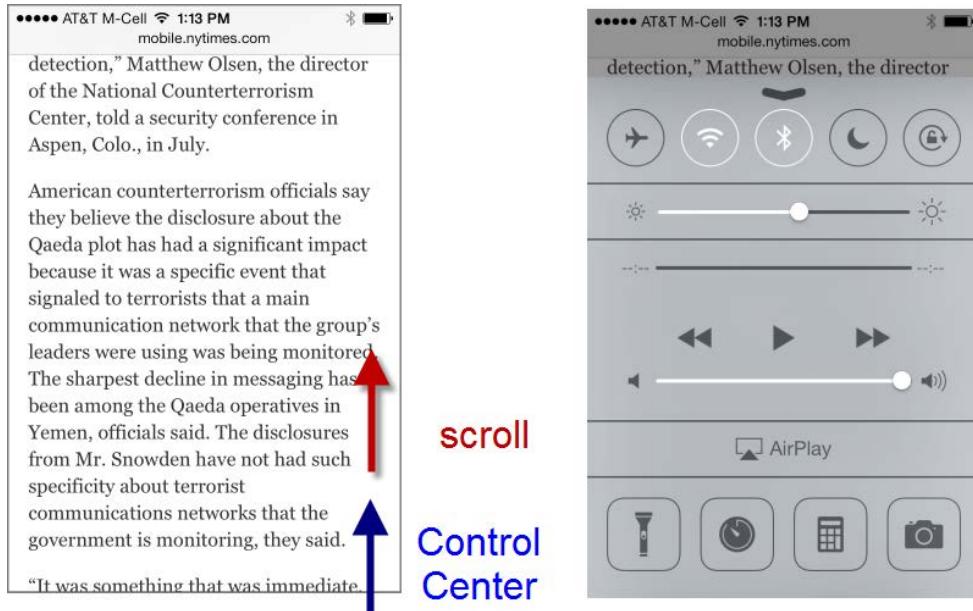
Notability, a drawing and note-taking app, faced a dilemma encountered by all such apps: how to distinguish between drawing (naturally implemented by swiping the finger across the screen) and scrolling down the page. They used a two-finger swipe for scrolling, but, because this gesture is rarely used in apps, they also provided a visible scrollbar that allowed users to move up and down the page.



Notability for iPhone: users could scroll either by using a two-finger swipe down the page or by using a single-finger swipe on the right scrollbar.

Gesture Ambiguity

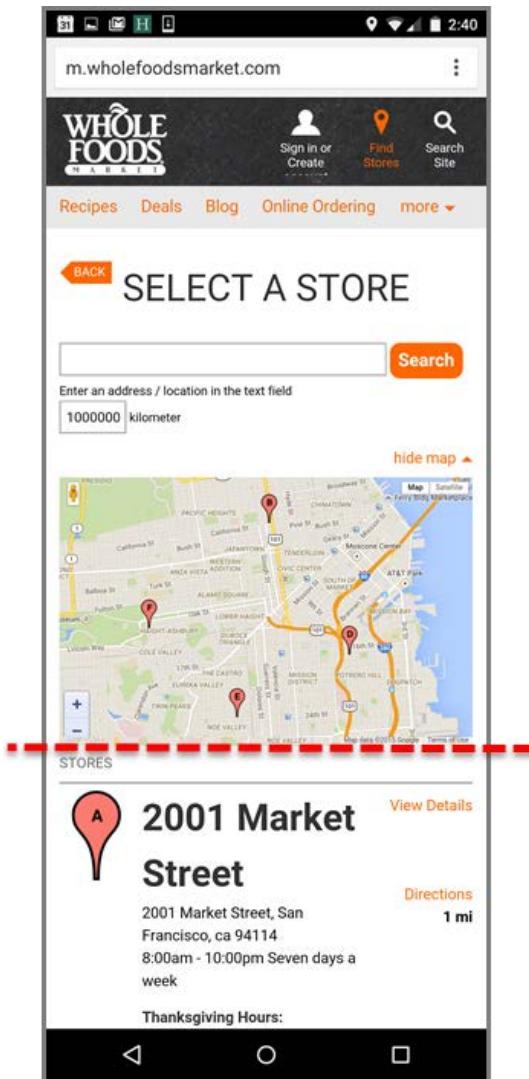
Gesture ambiguity refers to assigning several possible meanings to the same gesture on the same page. A small difference in the location of the gesture decides what action will be triggered. It's most common with the swipe gesture and is encountered in a variety of contexts, especially in iOS. Swiping near the left, bottom, or upper edges of an iPhone can cause problems if the swipe is not precisely executed. For instance, in Safari swiping on the bottom edge might either expose the Control Center, a place where some frequently used functions of the phone are grouped together, or it might scroll the page down. And when we talked about carousels we saw that it's advisable to leave some gutter space between the carousel and the side of the screen (guideline 109) to avoid accidentally navigating to the previous page in the browser when swiping to move the carousel.



Scrolling up in Safari could accidentally expose the iPhone's Control Center.

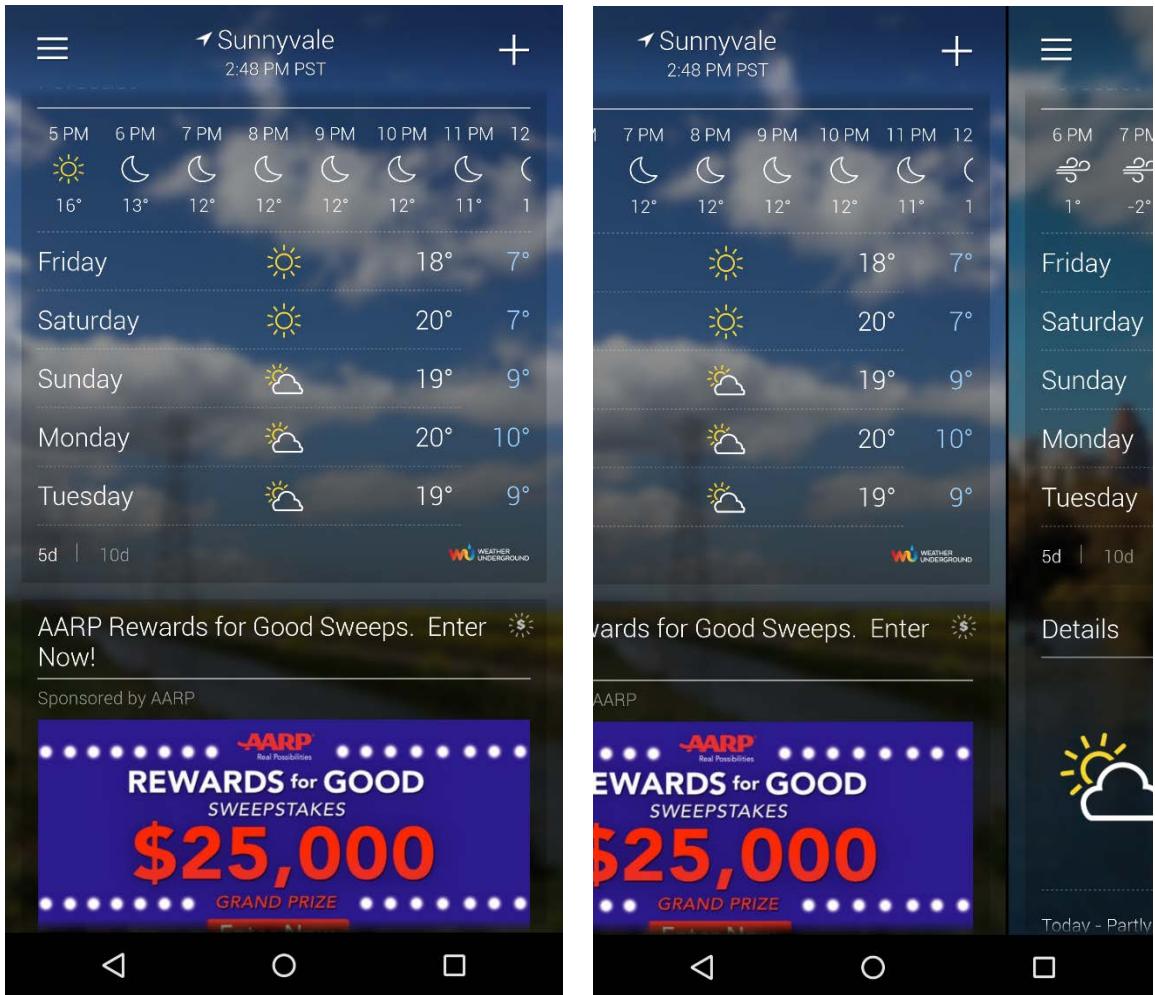
152. Avoid swipe (or other gesture) ambiguity: Do not assign multiple meanings to the same gesture depending on where the gesture is deployed on the page.

There are countless examples of swipe ambiguity on touch screens. Beside carousels (see guideline 109), some of the most frequent contexts for swipe ambiguity are maps that are embedded on a page. Often designers allow users to scroll the map without realizing that the map scrolling interferes with the scrolling of the page on which the map is placed.



The placement of the scrollable map prevented users from scrolling down the page. The same swipe down scrolled the map and the page.

The Yahoo! Weather app allowed users to flip horizontally to see the weather in different cities, and also used a carousel for the hour-by-hour temperature. Someone trying to see a different location could easily swipe over the carousel and be surprised to get a different result than expected. This is because, unfortunately, when users swipe to turn the page, they rarely think of localizing the gesture on a part of the screen (in real life, it doesn't matter where you touch the page in order to turn it). Although they will not expect to swipe outside the carousel to make the carousel move, they do expect to swipe anywhere on the page to move to the next location. The bigger the carousel on the page, the more serious the issues.

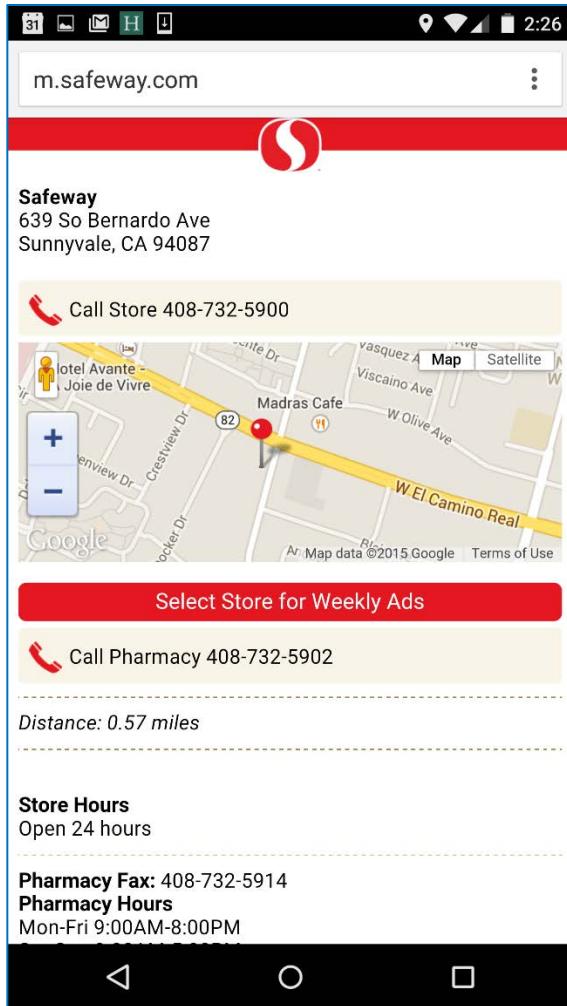


Yahoo! Weather for Android: A horizontal swipe could either swipe the hour-by-hour temperature carousel or move users to the weather for a different location. The designers tried to mitigate swipe ambiguity by leaving a narrow gutter between the hour-by-hour carousel and the screen edge.

The Yahoo! Weather designers tried to reduce the problem by leaving some space between the carousel and the edge of the screen. While this solution is not optimal, it does invite users to localize their carousel swipe farther away from the screen edges. (The optimal solution would be to not have two actions associated with the swipe gesture.)

- 153. To minimize swipe ambiguity, separate the location of the swipes as much as you can. For carousels or maps, place them far from the vertical or horizontal edges of the screen.**

In the Yahoo example under guideline 152 we saw that leaving gutters around carousels alleviates swipe ambiguity. Here's another example from Safeway, which placed the map far away from the top and bottom edge of the screen and left users plenty of room for scrolling. It also left a vertical gutter on the sides of the map to minimize horizontal-swipe ambiguity.



Safeway placed the map far away from the top and bottom edges of the screen and also left a gutter on the vertical edges to minimize both vertical and horizontal swipe ambiguity.

Tapping and Content

The most common gesture on touch screens is, of course, tap. Here we describe two circumstances where tap is commonly used: (1) tap the margins of a page to turn the page in an e-reader (or magazine) app; and (2) tap a full-screen image (or video) to show the interface controls.

154. In content-reader apps, it should be possible to turn pages both by tapping the page margins and by swiping horizontally.

The first gesture — tapping to turn pages — is almost a contraction of the swipe gesture: when you swipe again and again, the gesture becomes more economical and essentially transforms into a tap. Although users are not necessarily aware of it when they first start using content-reader apps, they eventually discover it.

We talked before about the affordance of swiping for turning pages (see the discussion under guideline 146). Because of the close relationship with real books, that gesture has better affordance in content readers. Therefore, we recommend that the two gestures coexist within the same app to support the same function: turning the page.

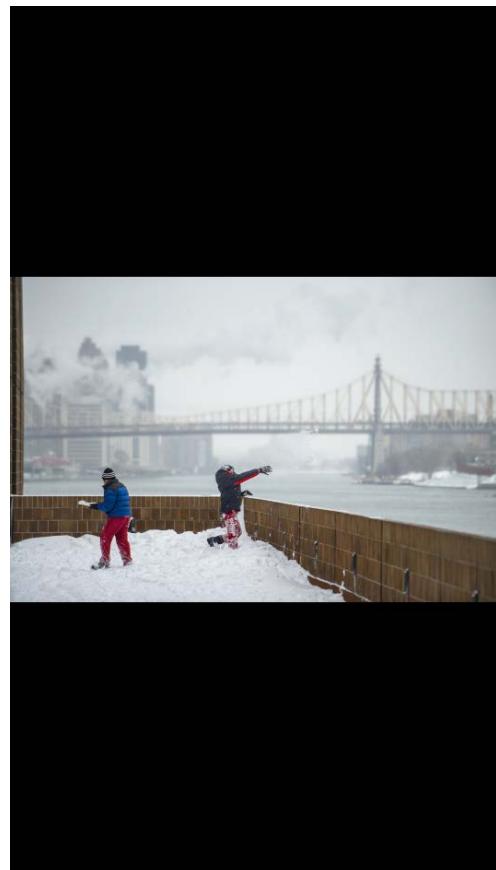
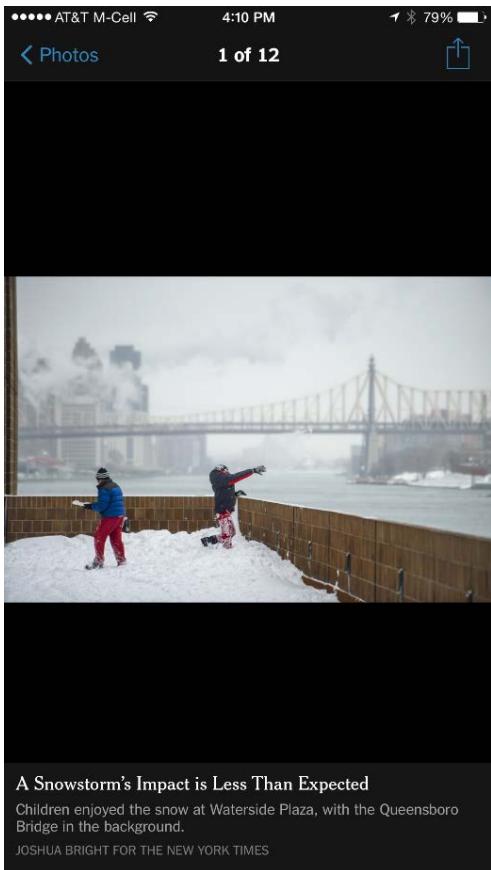
- 155. When displaying images (or video) full screen, allow users to get to both controls and captions by tapping on the image.**

- 156. When displaying images full screen, start by showing both controls and caption, then fade them away.**

Tapping an image to show interface controls is less discoverable. We've seen many people not realizing what they have to do when they are confronted with a picture and no interface buttons. However, we still think it's very important to be able to show pictures (or videos) full screen, to take full advantage of the already-small screen real estate. The solution is simple: whenever showing an image full screen, display the controls for a few seconds and then fade them away. In that way, users become aware of the existence of the controls.



Gilt for iPhone: When the image was first shown full screen, the controls were displayed. Tapping on the image made them disappear. (Note: this image did not display the title of the product, which is contrary to our recommendations.)



NYTimes for iPhone showed captions and controls first; on tap they were hidden.

FORMS

157. In a form always specify whether a field is required (preferably by using an asterisk).

It's best to have such a specification next to every single required field, instead of having a message such as *All fields are required* at the top of the page. People rarely read general instructions, and even when they read them, they might forget the instructions, given the interruptible nature of mobile interactions.

You might think that it may be more efficient to just mark the optional fields, since they are fewer. Unfortunately, to determine that a field is required, people would have to scan the entire form and check for any optional fields. That is not going to happen — instead, they will use common sense to assume which fields are mandatory, and users' expectations can differ significantly from the designers'.

The asterisk has become the standard for indicating mandatory fields and it has the advantage that it does not take up much space, so use it.

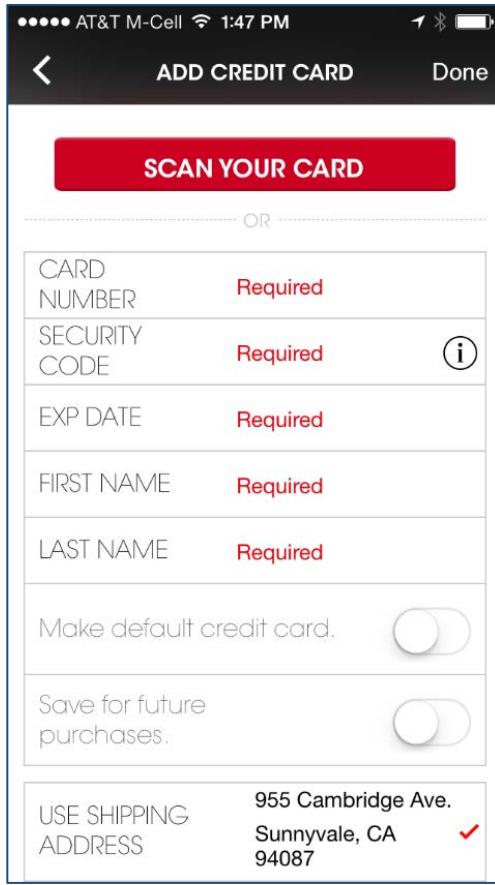
Left Screenshot (Older Version):

- Header: AT&T M-Cell, 1:45 PM
- Section: manage my account
- Fields:
 - *last name: [Text input]
 - *email address: [Text input]
 - *password: [Text input]
 - *confirm password: [Text input]
- Text: Your password must include ALL of the following
 - At least 8 characters, but no more than 20 characters
 - At least one number or one special character or one capital letter
- Checkboxes:
 - please send me emails about special offers, exclusives, and promotions from Target.
 - [Privacy Policy](#)
- Buttons:
 - create an account** [Red button]
 - By creating an account, you are agreeing to our [Terms & Conditions](#), [Delivery Details](#), and that you are at least 13 years old.

Right Screenshot (Newer Version):

- Header: AT&T M-Cell, 4:27 PM
- Section: sign up
- Fields:
 - @ email address
 - password
 - name
- Text: By creating an account, you are agreeing to our [Terms](#) and [Privacy Policy](#).
- Checkboxes:
 - receive Target email offers
- Keyboard:

Target for iPhone: An older registration form (left) specified which fields were required using a red asterisk. A newer version of the form (right) was simpler, yet it left out this important information.



Sephora for iPhone used the word *Required* to mark mandatory fields. Unfortunately, once the user moved the cursor to that field, the word disappeared.

158. In a form specify whether a field is optional.

If the required fields are specified, then why should you also specify the optional fields? While it's not absolutely obligatory to do it, it does simplify the user's cognitive load: in the absence of that word, the user must look around and infer that the field is optional based on the other fields being marked as required. If the word *optional* is next to the field descriptor, that task becomes a tad easier.

Not specifying that a field is optional is not a deal breaker, but doing so is a nice perk.

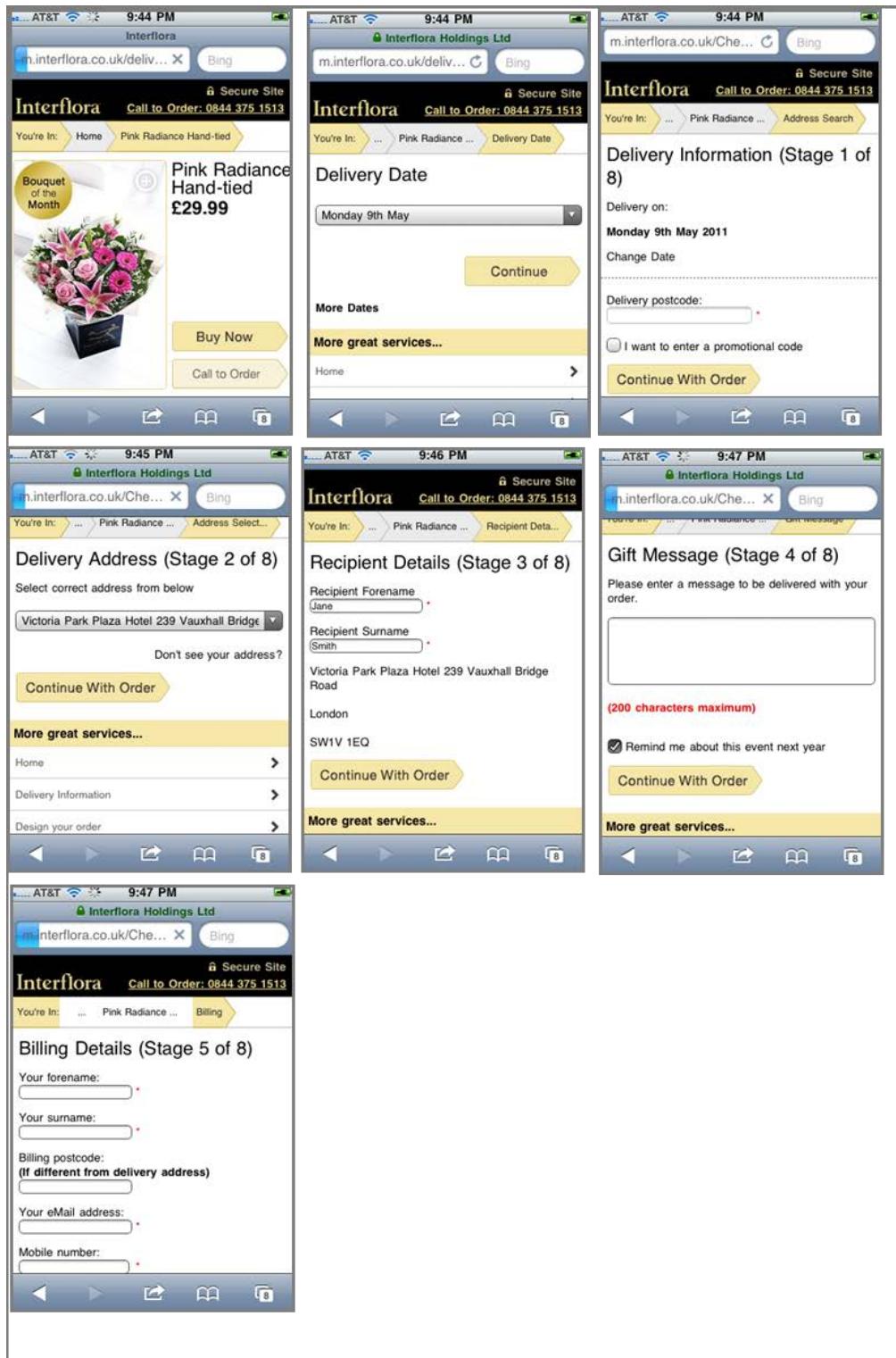
159. Minimize the number of page loads in a form.

Each time users need to submit information to a server, they have to wait for a response. Moreover, the chance of the connection being dropped increases, and the users may lose all their hard work. That's why it's important to minimize the number of steps that make server requests and incur wait time (see the section *Variable Connectivity*).

Note that applications suffer less from this problem: if an application keeps data on the phone, an extra click or tap will not necessarily incur a request to the server and it will not take any significant wait time. That said, there are applications that keep most of their data on a server; for them, this guideline still applies.

A user ordering flowers at Interflora.com complained repeatedly about having to complete 8 steps in order to be able to order flowers:

"You've got your options for color, yes, that's moved in quite quickly... There was only 6 or 7 options on the last screen [where flowers were chosen], you could've made your color selections on the same screen, instead of having a screen just for color selection ... [Next step] add something extra — that's fair enough, they're trying to make money ... Maybe a fast track would be useful ... Delivery code, post code ... I have to read all that because we're moving towards the nitty-gritty [...] Ok, there's some excess page here ... When I've put the postcodes, the options [for the addresses] should have come up right there... Even when you've got pages coming up every two seconds, you're at work, you're not meant to be doing this, you know, it'd need to be sped up lot and you also want the simplest website you can get ... [Entering the recipient] is unavoidable, but choosing flowers and choosing color, that's too much. Now a message... My reaction is the process is dragging, I'm halfway through, yes, I have to put in my credit card and so on, but, it's essential, if you could get it down to six screens or something... Right, billing details... Again, to be honest, there you have a telephone number you can ring, and it's usually quicker to call someone ... I've got a billing confirmation — is this necessary? Surely we could have all confirmations... you could have the flowers, where it's going to, you don't get to make that much of a confirmation, you could have made a mistake early on..."



Interflora.co.uk (older version) used 8 different steps (and many more page loads) to implement the checkout process.

Several steps on the Interflora website could have been compressed together — all the information needed from the user (delivery date, name and post code of recipient, gift message, and buyer information) could easily have been placed together on a single screen. A newer version of Interflora compressed the checkout flow to only 3 steps (and 5 different page loads).

The figure consists of three screenshots of the Interflora mobile website's checkout process, illustrating how the number of steps was reduced from five to three.

- Screenshot 1: Step 1 - Check Availability**
This step shows a delivery address (239 sw1veq), a delivery date (Thursday 23rd Aug), and a delivery option (Standard Delivery (£5.99)). It includes a promotional code field and a note about selecting a delivery address.
- Screenshot 2: Step 2 - Confirm Delivery Address**
This step shows the confirmed delivery address (Flat 10 Russell House Cambridge Street LONDON SW1V 4EQ). It includes a "Select another delivery address" link and a "Confirm and Continue" button.
- Screenshot 3: Step 3 - Recipient Details**
This step shows recipient details (Recipient Information: r, b; Gift Message: a message box); it also includes a note about sending a gift message and a "Need assistance or want to order?" section with contact info.

The screenshots also show the "Order Line: 0844 375 1513" and "Interflora" branding at the top of each page.

A newer version of Interflora.co.uk compressed the checkout process in 3 steps (and 5 page loads).

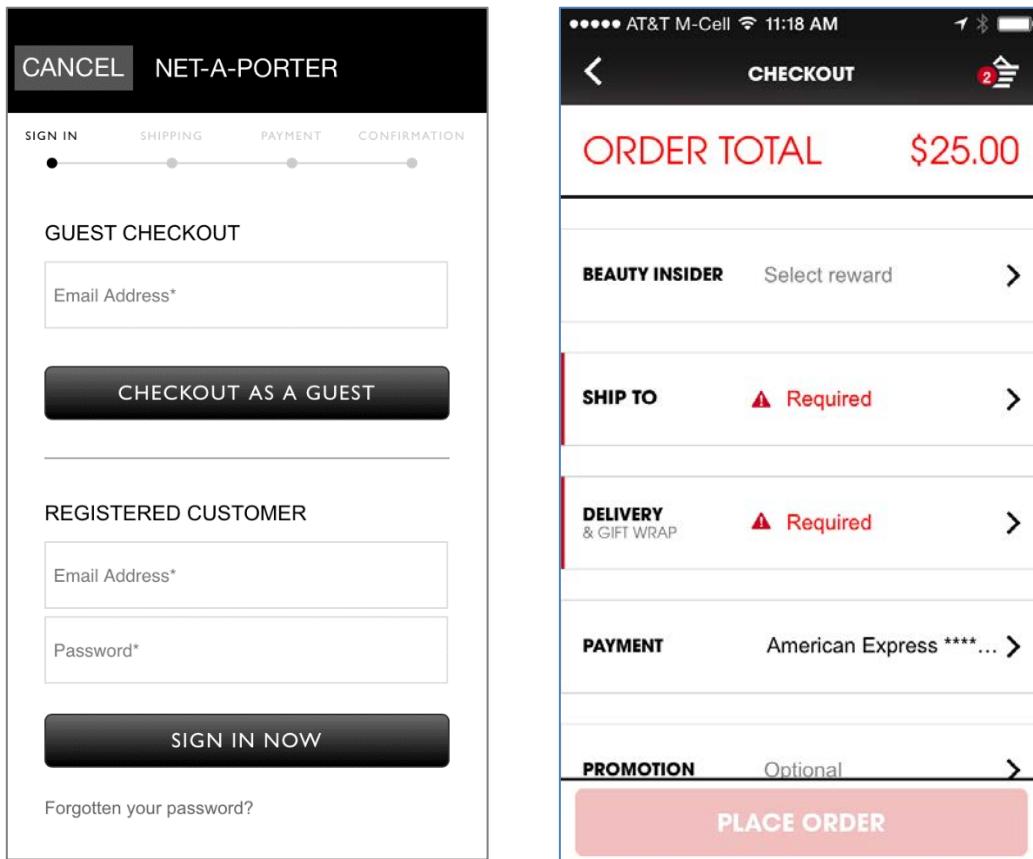
On an older version of Fandango's website, to search for a movie, users had to enter the movie title and then enter their zip code on a separate page. One user observed that these two steps could have been easily compressed into one:

"Now I actually have to go and search for this movie...And now I'm doing all this again, where I have to enter my zip code, and this is like, oh, my god, why couldn't have I put my zip code and the movie title at the same time — that's one less webpage I have to visit, that's one less loading time..."

- 160. For multistep forms, give users a mental model of the form by showing an overview of the form steps (e.g., a workflow diagram or a list of the steps).**

Seeing all the steps that make up a form helps users form a mental model of the process and keeps them oriented within the task flow. If people know what to expect, they know how much time they need to allocate to the process.

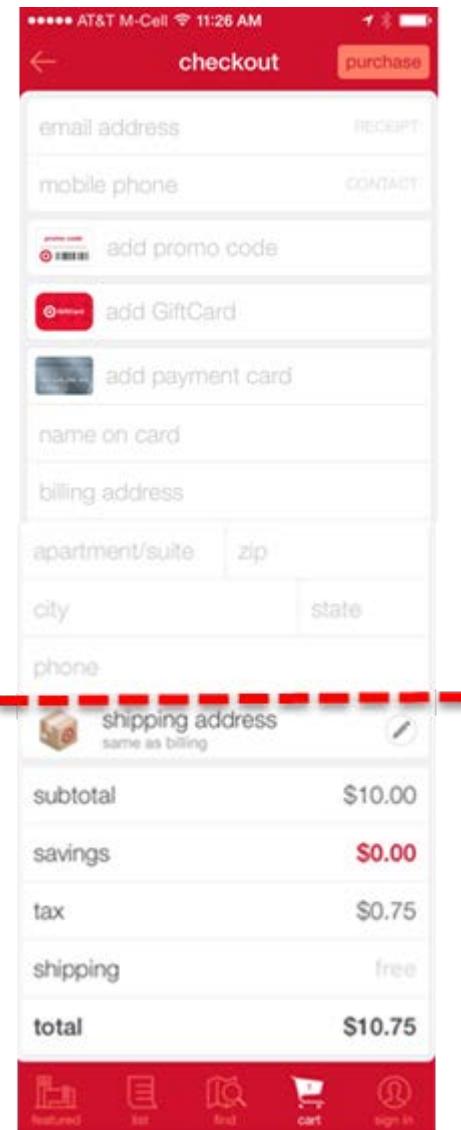
The mental model can be rendered either as a workflow diagram (as in the Net-a-porter example below) or as a list of the different steps that make up the form (as in Sephora's case).



Mental models for the checkout process: Net-a-porter for iPhone (left) showed a workflow diagram at the top of the checkout form. Sephora for iPhone (right) showed a list of the steps that make up the checkout process. Each step took users to a different page.

Target attempted to streamline its checkout process and minimize the number of page loads by merging all the different checkout steps in a single-page form.

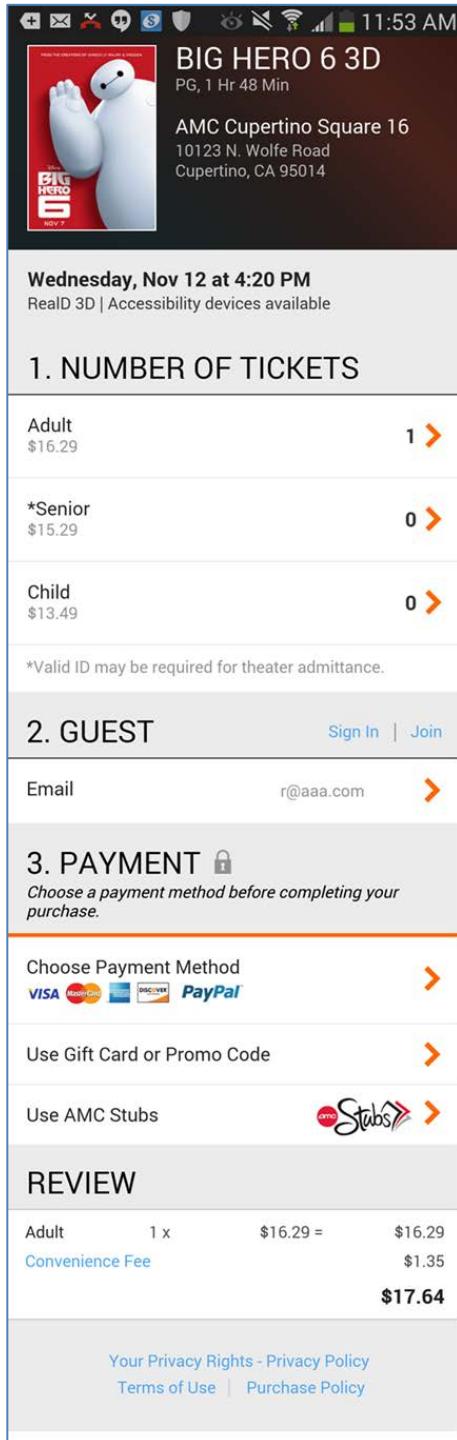
While the initiative is laudable, the resulting form is too unstructured and looks overwhelming. The same low page load could have been achieved using accordions to enclose the different checkout steps.



Target for iPhone: All the different steps of the checkout form were merged into one long form, with no apparent structure. Although Target did the right thing in minimizing the number of page loads, the lack of structure on this checkout page made it overwhelming. (The red line indicates the fold.)

Fandango also used a single page for the checkout form; in this case, the different steps were marked with clear headings. Unfortunately, the long page (relative to the small mobile screen) made it hard for people to get an overview

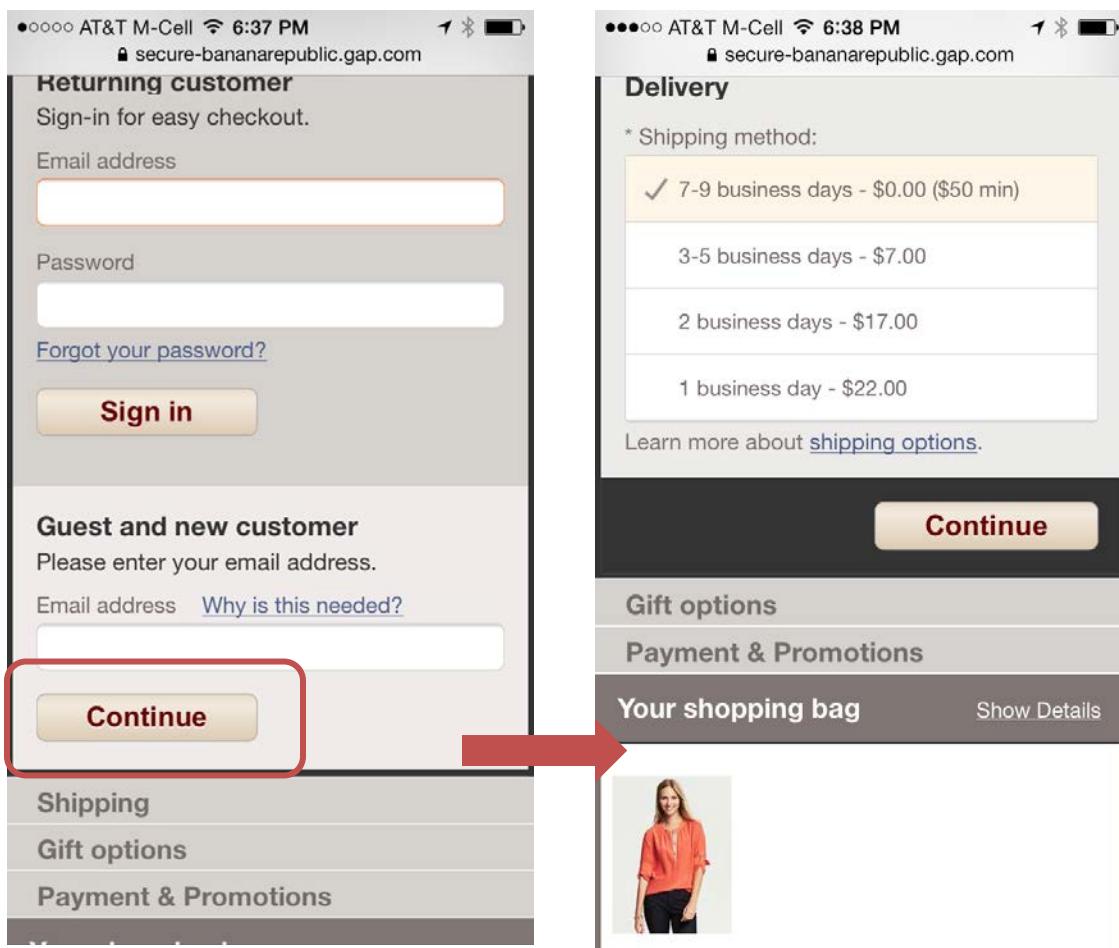
of the information that they needed to enter. Again, accordions would have been a more effective design choice.



Fandango.com: The steps were nicely illustrated and separated by the all-capital headings. However closed accordions would have better conveyed the mental model of the workflow.

161. Use collapsed accordions to represent different steps of a form.

Accordions, when closed, successfully convey a mental model of the form process (see the previous guideline 160) and are a convenient way of compressing steps without making the page excessively long or overwhelming for users. The Banana Republic website used a single page for the checkout process, with all the different types of information (billing, shipping, coupons) grouped under accordions.



Banana Republic's site: The *Shipping*, *Gift options*, *Payment & Promotions* links were expanded on the same page once the users filled in all the previous steps. Thus, after they entered their email and clicked the *Continue* button, the *Shipping* section gets expanded.

The screenshot shows the Skinnyties.com checkout process. At the top, there's an orange header bar with 'SPECIAL OFFERS' and 'Free shipping + tie bar!'. Below it is the website's logo 'SKINNY TIES' and a navigation bar with links for 'Collection', 'Color', 'Width', 'Fabric', and 'Pattern'. A search icon is also present. The main content area has a blue header 'Checkout'. The first step, '1 Checkout Method', is active. It contains fields for 'Email *' and 'Password *', along with 'SIGN IN' and 'Forgot password?' buttons. Below this are sections for 'Create An Account' (with 'Register and checkout together' and 'Checkout without registering' options) and 'Guest Checkout' (with a 'CONTINUE' button). To the right of the steps, there's a vertical list of the remaining steps: '2 Billing Address', '3 Shipping Address', '4 Shipping Method', '5 Payment Method', and '6 Order Review'. Each step is preceded by a numbered circle.

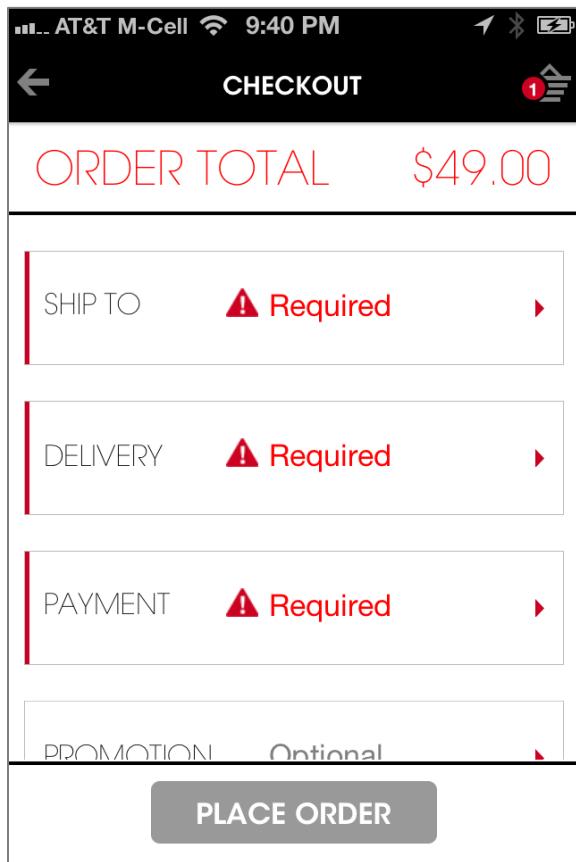
Skinnyties.com's checkout form also used accordions that allowed users to see the entire workflow without being overwhelmed by a really long form.

162. When using a list of steps in a form, force users to complete the different steps in a fixed order.

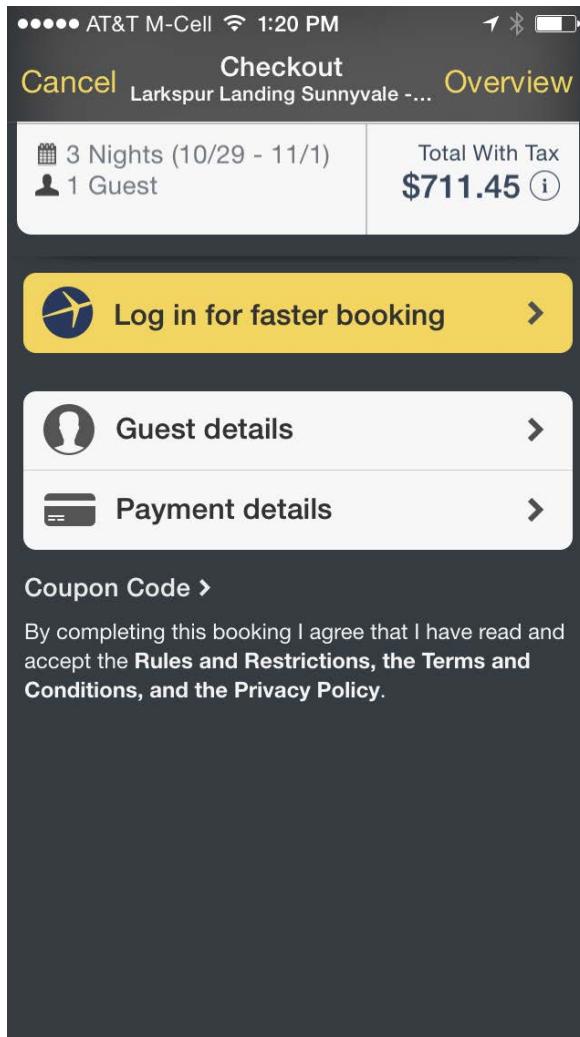
Do not allow users to start with step number 3, even though technically it is ok. The reason is simple: people sometimes get confused if they can fill in steps in any order (and they might also skip or forget certain steps, or think that not all

the steps are mandatory). It's best to make a step available for expansion only after all the prior steps have been completed.

Skinny Ties (shown in the example under guideline 161) did exactly as we recommended: users had to go through the checkout steps in a fixed order (and the steps were numbered accordingly). In contrast, Sephora allowed users to complete the different steps in whatever order they wanted, causing occasional confusion and disorientation.



Sephora for iPhone allowed users to complete the steps in the checkout process in whatever order they wanted; for instance, they could start with the payment step.



Expedia for iPhone: Users could choose any of the three calls to action first (*Log in for faster booking*, *Guest details*, and *Payment Details*). Users were confused as they were not sure which steps were mandatory and which were optional, and what needed to be done first.

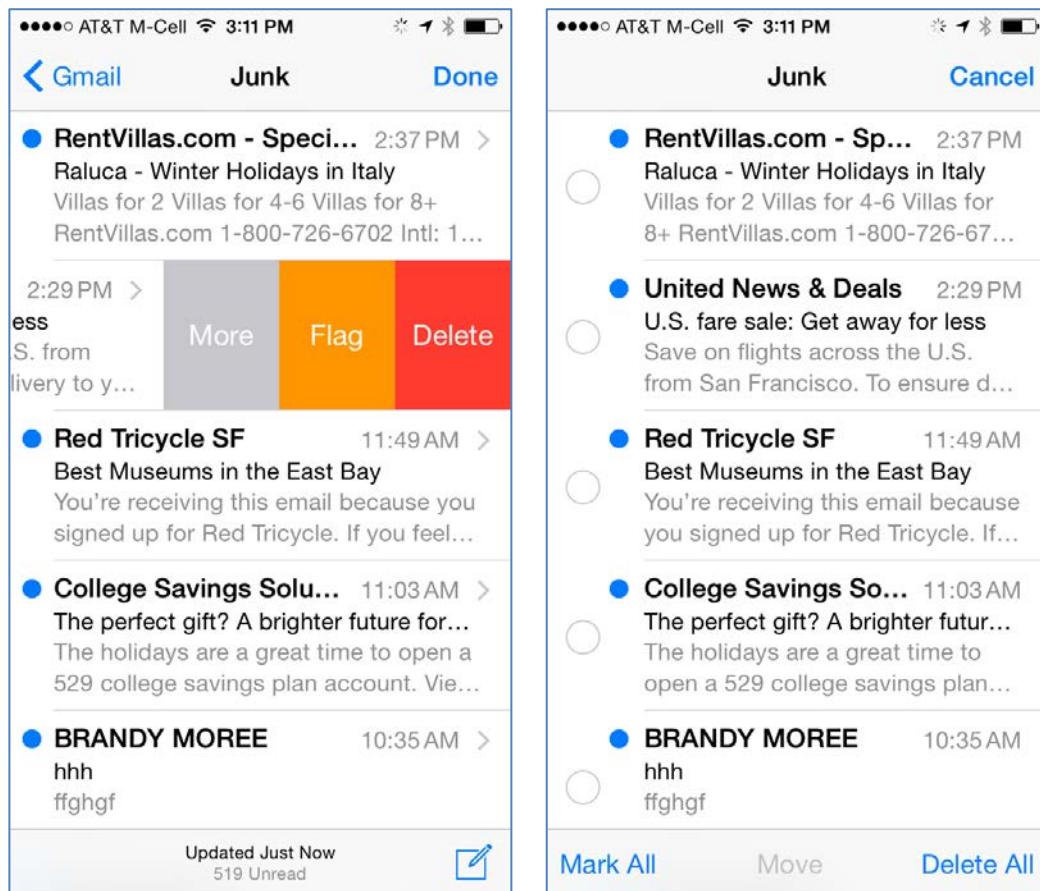
- 163. The form *Submit* button (or equivalent) should be displayed under the form fields rather than at the top of the page.**

See also guideline 114 above.

On iOS, the *Done* button is often displayed in a navigation bar at the top of the page. Sometimes the form *Submit* button (whether called *Submit* or something else — for instance, *Place Order*) is also placed at the top of the form. This pattern has started to trickle into Android apps as well.

Even in iOS apps we recommend against following this pattern for the simple reason that it goes against the natural top–bottom workflow on the page. As users fill in the form, they usually do it top to bottom. When they get to the end of it, they expect to find a *Submit* button right there, next to the last field

they've completed. Most of the times, when people don't find it there, they get confused and start looking around, not knowing what to do.

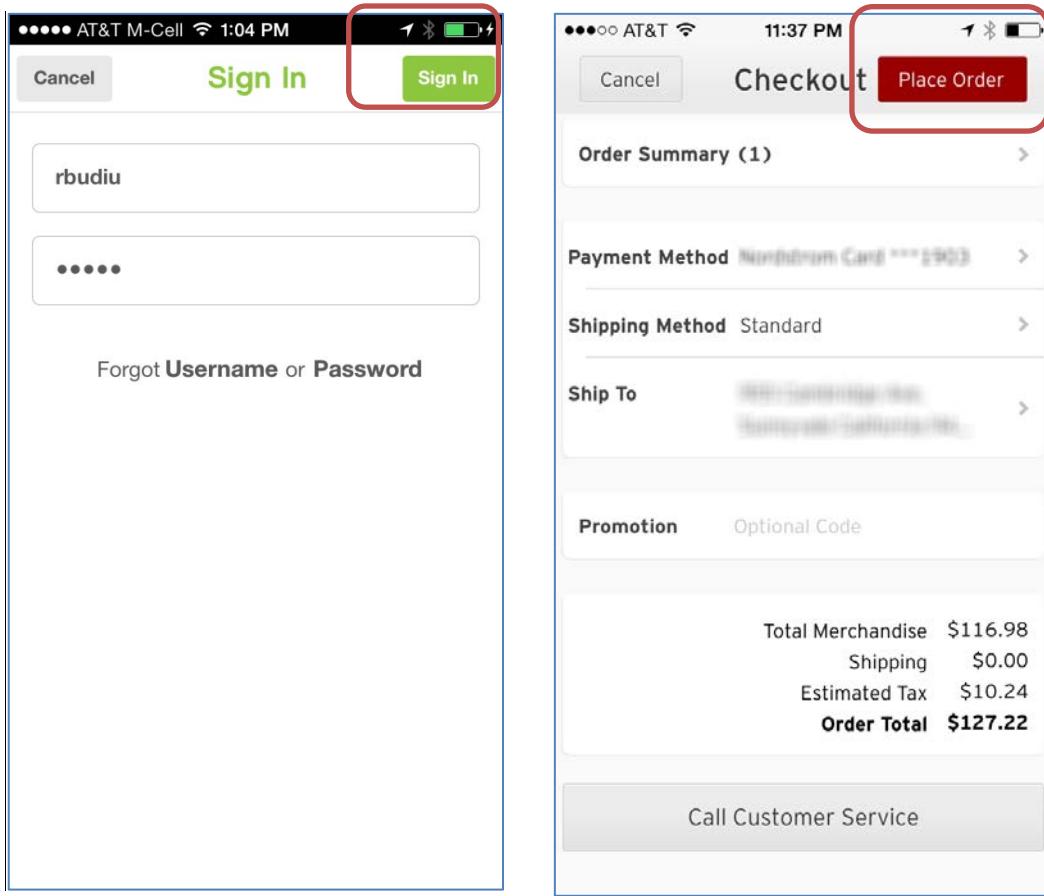


Mail app for iPhone: The *Done* and *Cancel* button were ungrouped with the other buttons related to the task (*Delete* and *Move*).

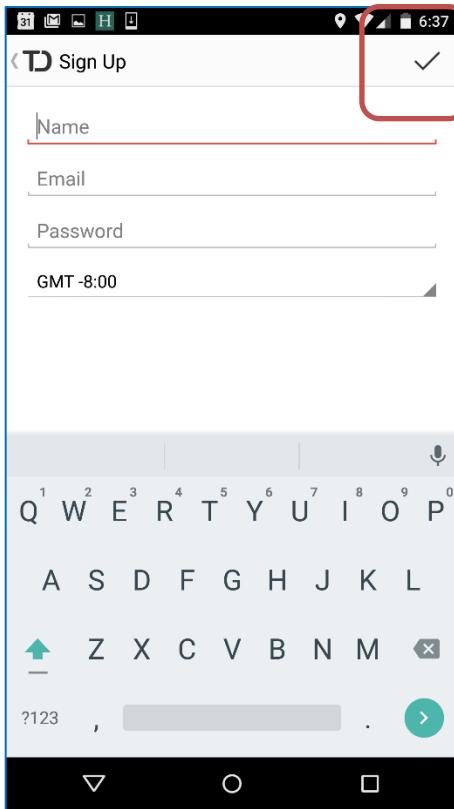
In Mail, the *Cancel* and *Done* buttons appeared at the top of the screen. However, the *Cancel* and *Done* buttons were not the most important ones with respect to the task they support (which is deleting messages). And one could argue that, in the case of Mail, the button at the top applies to the entire view (essentially, it quits the current view), rather than to one individual message, in which case it is ok to place them at the top.

Unlike in Mail, in the examples below (Pinkberry and Nordstrom), the *Sign In* and *Place Order* buttons were essential, because they needed to be pressed to complete the task. Moreover, they had a strong spatial and/or temporal connection to one other area of the screen: they needed to be pressed after the user had filled in the form. Placing these buttons at the top of the screen was against the natural flow of filling in the form: once done with all the fields of the form, users found themselves at the bottom of the page, left with nothing to do next. Looking for and finding the corresponding *Submit* button required some unnecessary extra effort that could have been saved if that

button had been placed at the bottom of the page, under the last field, as it is customary on most forms on the desktop.



The working area in Pinkberry for iPhone (left) and in Nordstrom for iPhone (right) was disconnected from the *Submit* buttons (labeled *Sign In* and *Place Order*, respectively), which were placed at the top of the screen.



Todoist (a to-do-list app) for Android borrowed the iOS pattern and placed the *Submit* button (the checkmark icon) in the top right corner. (Note also that the form fields violated many of our guidelines: the labels were inside the field, and can you guess what *GMT-8:00* stands for? Answer in the footnote⁵⁰.)

164. In a form always include a *Submit* button.

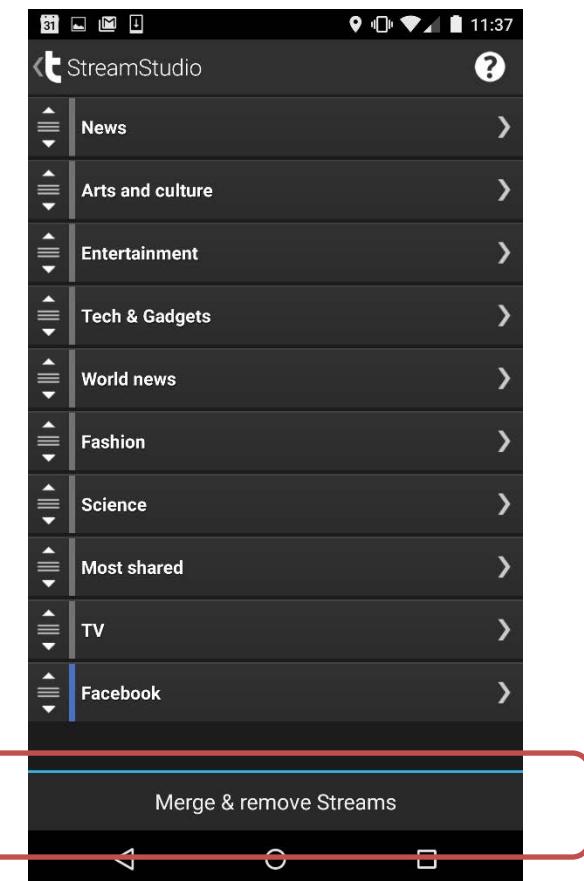
Sometimes apps are too clever and submit a form once the user has filled in all fields and made all selections. Unfortunately, the lack of a *Submit* button puzzles people and makes them wonder about the correct way to interact with the page. We recommend that you always present users with a *Submit* button, even though you may think that technically such a button is not necessary.

165. Distinguish between link buttons and *Submit* buttons. *Submit* buttons are usually placed at the bottom of the form; when link buttons are positioned there, they can create confusion.

Some forms contain buttons that lead to subforms. The placement and labeling of these buttons is important because it can mislead users into correctly understanding the purpose of the initial form.

⁵⁰ It referred to the current time zone: by default the app had correctly selected GMT-8:00 (that is, 8 hours before Greenwich Mean Time, or, in other words, US Pacific Time).

In Taptu for Android, a topic-customization page had a big button labeled *Merge and remove Streams*. Our participants thought it was a *Submit* button and couldn't figure out how to indicate which streams to merge on this page. In fact, they couldn't. The *Merge and remove Streams* button was a link to a page where they could merge and remove streams, and not a *Submit* button. This page only allowed users to change the stream order.



Taptu for Android. The button at the bottom of the screen (*Merge and remove Streams*) was incorrectly interpreted by the participants to be a *Submit* button, although it was in fact a link to a page where they could merge and remove streams. As a result, they thought that the form in the screenshot was for merging streams when in fact it was just for reordering them.

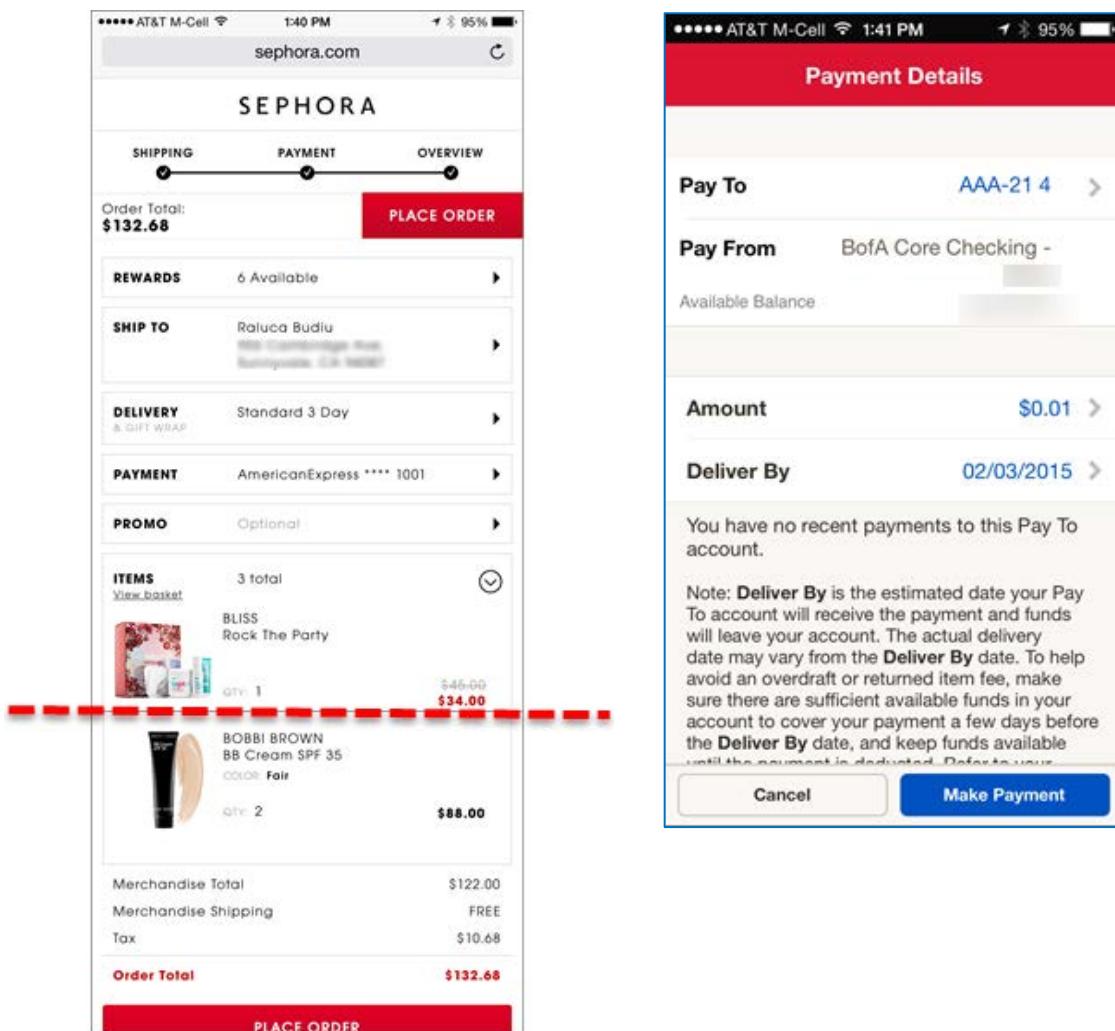
From this example you can also see why it's important to include a *Submit* button in the form (see guideline 164) — when users did not see one, they assumed that the only button present on the page was the *Submit* button.

166. For transactional forms, show a one-screen summary of the transaction before the user needs to commit the transaction.

On mobile it's easy to overlook information or to make a mistake. Because of that, people are often cautious and reluctant to make transactions with significant consequences: not only because they are somewhat unsure of the

security of the medium, but also because it's so easy to type something wrong or to overlook an important piece of information.

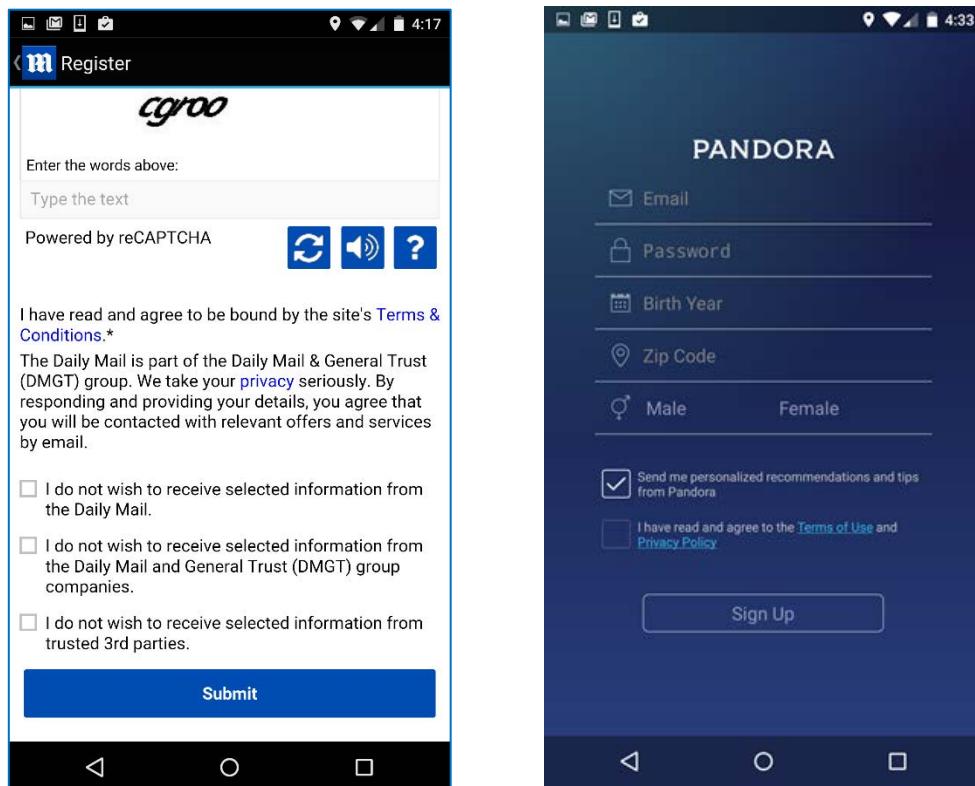
The app or mobile website needs to work hard to reassure the user that all the filled-in information is correct. When the information related to a transaction does not fit on one screen, users cannot be certain that they haven't made a mistake. If a summary of the transaction is visible, users will not have to keep anything in their memory; they will see all that is relevant to the transaction in front of them and confidently press *Submit*.



Sephora.com (left) split the confirmation page over 2 screens, while Bank of America (right) contained all the information pertaining to the transaction on a single screen, and thus allowed users to easily make sure that the form did not contain any mistakes.

167. Do not automatically subscribe users to newsletters or other email. Allow users to opt in, instead of requiring them to opt out.

People are wary of companies trying to send them email and carefully study all the checkboxes in a registration or checkout form. (If there is one situation where users are careful and not rushing, this is it! They've been burned too many times with spam and unwanted emails, and they are usually circumspect.) So by checking by default the checkbox that subscribes them to your email most likely you'll only succeed in losing their trust and making them suspicious.



By default both Daily Mail (left) and Pandora (right) subscribed users to emails. Daily Mail required users to check several boxes to opt out, while Pandora forced them to uncheck a checkbox.

- 168. Whenever users make a transaction on the phone, email them a confirmation number for that transaction. Inform them that they will receive an email and tell them what it will contain.**
- 169. Allow users to “soft print” the transaction confirmation screen by taking a screen capture.**

Transactions often involve a confirmation number, and that is hard to write down when people are away from their desk. Unless users can be certain they will have later access to that confirmation number, they may be reluctant to commit to a transaction whose traces may be hard to recover.

A user tried to accomplish a funds transfer between two bank accounts on his phone, but stopped short of doing it because he was concerned about losing his confirmation:

"The other thing that happens, at least on my laptop, is that, once I do [the transfer], it gives me a confirmation number that I could save... I don't know what to do about the confirmation that I get here. And it's important to me because it's happened before: I know I transferred something, it didn't occur at Bank of America, they charged me saying 'you don't have enough funds'. I kept the record and I emailed the document to them, 'this is my screenshot of this confirmation', and they said, 'oh, you know, there must have been a problem on our end, we're sorry'."

To prevent worries that the transaction may not go through, send users a confirmation email with the transaction information; also let them know to expect that email.

Another way to help users preserve an accurate level of history is by taking a screenshot. A screen capture is the mobile phone equivalent of "Print Screen".

(Don't assume that users will already know how to take a screenshot of the screen using phone buttons: most have no idea.)

Login and Registration Forms

One of the most common forms that people encounter on mobile devices is the login form, where they are asked to enter a username and a password. We have observed many users repeatedly trying to enter usernames and passwords, and failing to successfully do so. Typing on a phone is hard, but passwords and usernames are even harder than regular typing. Part of the reason is that passwords and usernames often contain a mixture of upper-case and lower-case characters, as well as special characters. Accessing these kinds of characters can be tricky even on large-screen smartphones and touch phones. To add insult to injury, passwords are often masked — people cannot see what they type and they cannot check for mistakes.

The other problem with passwords on mobile is that users may not remember them. That's not a problem unique to mobile — it often happens on the desktop, as well. However, on the desktop users have more recourse available to them: they can check their password file (if they store their passwords in a file, as many users do), or they can use the *Forgot Password?* link. That last option is, of course, available on mobile too, but the process of recovering the password is a lot more tedious in a single-window environment.

170. Don't have a login wall when the app is first launched. Deliver value before asking users to log in.

Login walls⁵¹ are pages that ask the user to log in or register before proceeding. Although a website can offer to users the option to register once they have entered a lot of information about themselves, it should never ask for registration upfront, before delivering any value to the user. Most people want immediate gratification, and they need to be highly motivated and patient to want to work before seeing what they need to work for. Apps and websites

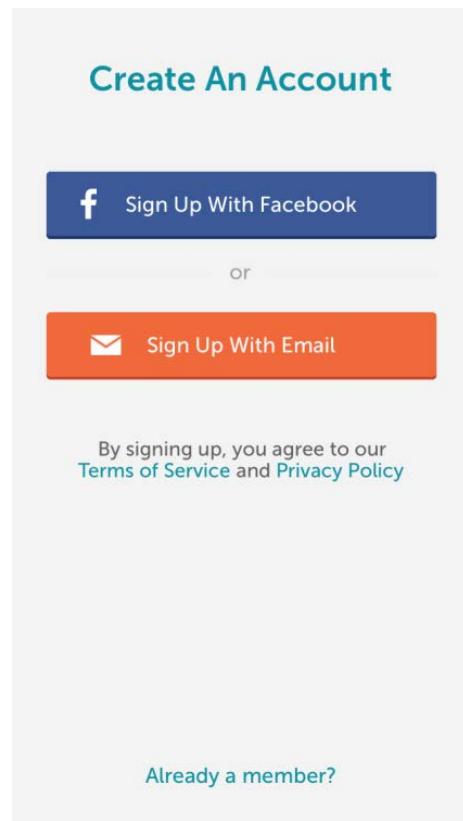
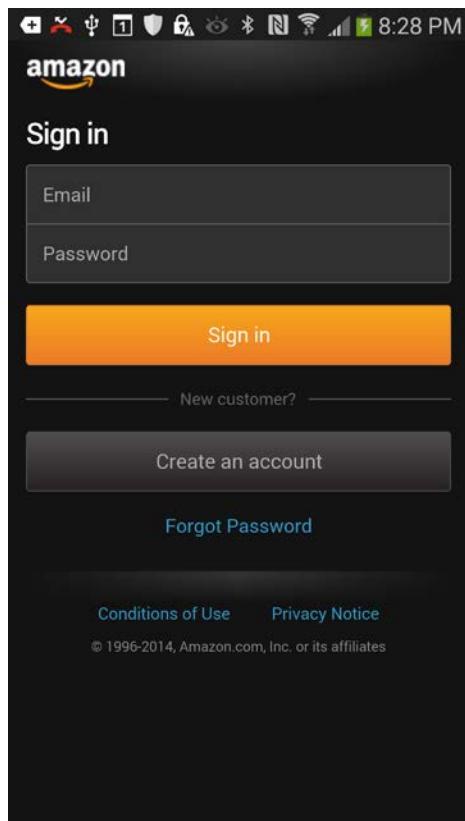
⁵¹ See also Raluca Budiu. "Login Walls Stop Users in Their Tracks."

<http://www.nngroup.com/articles/login-walls/>

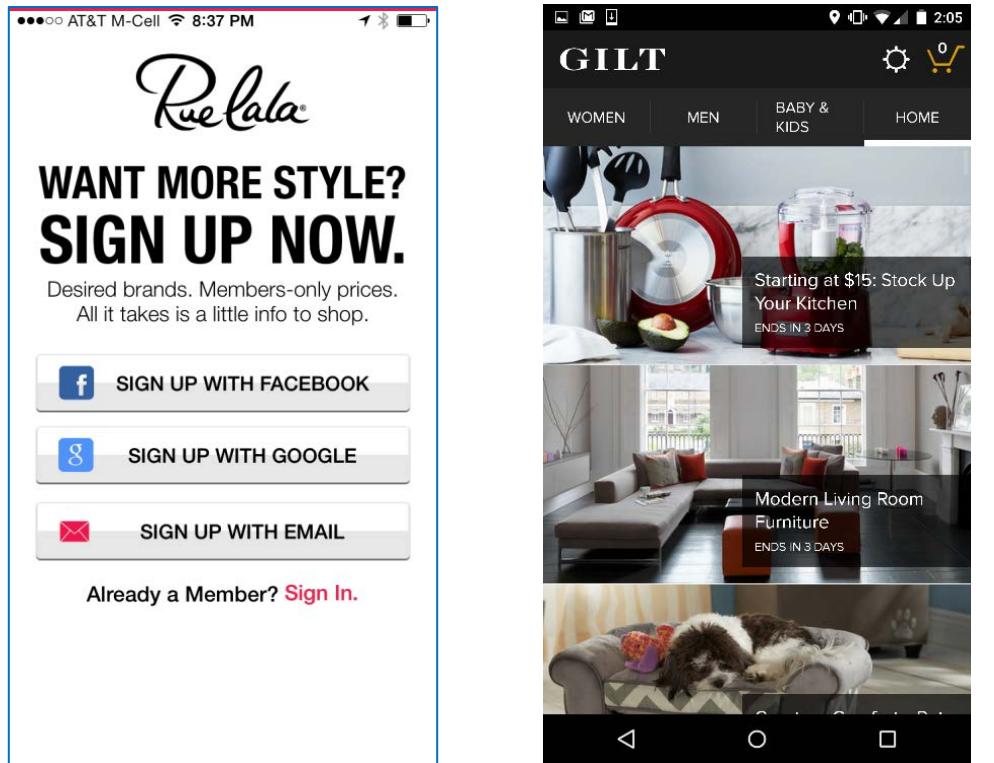
that ask people to register before offering any content slam a door in front of their users, and users don't know if it's worth opening it. The work involved in registering is considerable enough, and the benefit unknown. Often users simply want to try out the app and decide if they can use it — they should be allowed to do so without having to register.

Kindle asked users to register or log in right away, before showing them anything. Many of our users were put off by the registration; some had Amazon accounts but did not remember their passwords. Why not show them a few sample books right away to whet their appetite and motivate them to create an account?

Rue La La, a flash-sale app, also required users to first log in before seeing any of the offered products. By contrast, Gilt, a similar flash-sale app, allowed users to see its content and asked them to log in only when they wanted to add items to their cart.



Kindle for Android (left) and Lumosity for iPhone (right) raised a login wall and did not allow users to try out the app before having to log in or register.



Rue La La for iPhone asked users to log in before they could see any content; in contrast, Gilt let them shop and had them log in only after adding an item to the cart.

Even for companies such as Hulu and Netflix, which provide content to subscribers only, it makes sense to show a preview of the content available: the quality of the content may convince users that it's worth signing up for the service (although they may do it at a later time on their desktop).

- 171. Do not ask users to log in or register unless absolutely necessary for security reasons.**
- 172. Avoid forcing users to log in or register for authentication purposes only.**

Many websites and apps need login walls for authentication rather than security: that is, they want to know who their users are. One common reason is syncing users' data across different platforms. They may not realize that they lose business when they ask users to log in on mobile devices, simply because, many times, logging in is so painful that people give up. Registration and login incur a large interaction cost even on desktops; the cost is greater on mobile devices, where typing can be tedious and every extra click takes a lot of extra time.

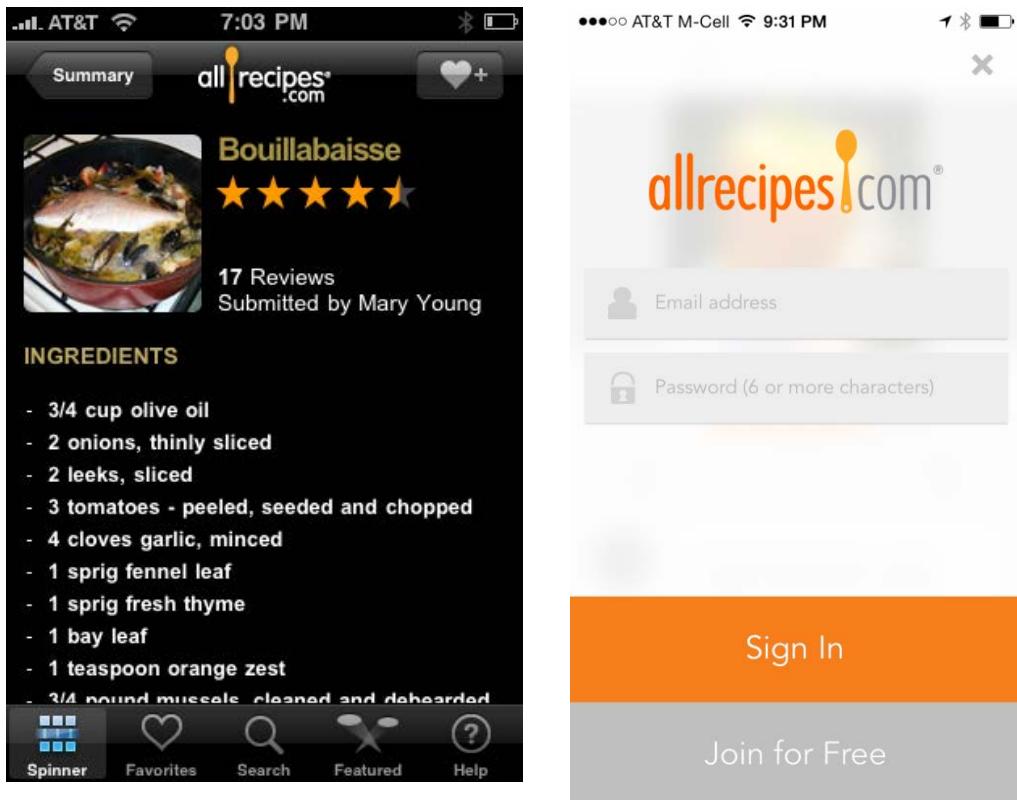
Users are annoyed at sites that enforce registration, especially if they think that they are not going to use the site again. As one of our users put it:

"Another thing that gets me is registering on a thousand different websites. It annoys me to no end, because you're going there for very basic information, so what's the point?"

173. [App] Do not ask people to register or log in to save information such as favorites.

In an app, it's easy to store favorites on the phone. Users should be allowed to sign in or register in case they want that the information on the phone be synced with a desktop site, but they should always be told if this is the only benefit of registering and allowed to opt out if they don't want it. As discussed before, registration and login are painful on mobile, and users shouldn't have to go through any of them.

All Recipes showed content to users without having them log in. However, while an older version let users save their recipes on the phone without registering, a later version required signup in order to store users' data: that should not be necessary!



An older version of All Recipes for iPhone (left) allowed users to save favorites without login. A newer version (right) required them to log in to save favorites.

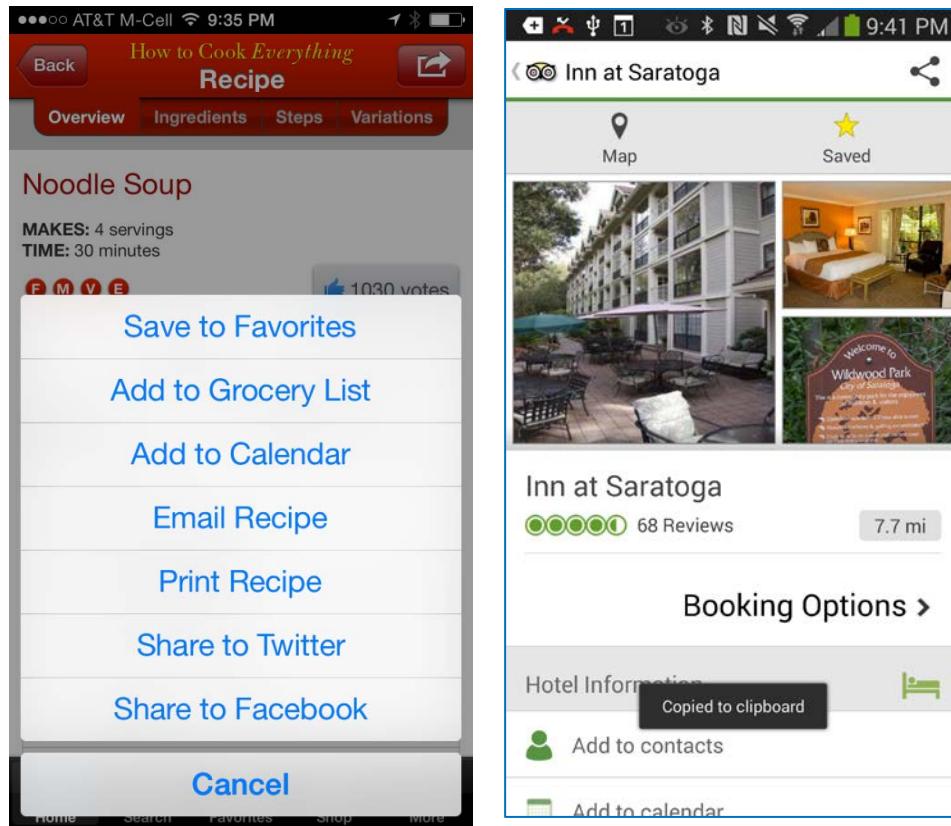
One of our Hong Kong participants was using Open Rice — an app similar to Yelp that offered restaurant information. He wanted to save a restaurant in his list of favorites, but, unfortunately, the app required him to have an account to

do so. He tried several times to create an account, but eventually he discovered that there was an account associated with his email. He had to recover his password and log in. Here is a screenshot from his session:



Open Rice app for iPhone: In order to add a restaurant to his favorites, the user had to have an account.

Unlike Open Rice and All Recipes, How to Cook Everything, another recipe app, allowed users to add a recipe to their list of favorites without logging in.



Both How to Cook Everything for iPhone (left) and Trip Advisor for Android (right) let users save information without requiring them to register.

174. Do not force people to sign in or register to check out. Allow them to use a guest checkout.

Often e-commerce sites and apps use login walls at checkout. Designers think that, by logging in, users will be able to take advantage of the prior account information and thus won't need to type in information such as address and credit card.

That may be true for many users in many circumstances; however it's not always true. Every now and then, users will shop on sites where they are unlikely to go back. (This is one of the 5 main classes of e-commerce user behavior⁵².) They would rather finish their task as soon as they can and be done with it; the benefits of creating a new account for a site that they are unlikely to use in the future are slim. As one user put it:

"I hate registering... I have to spend time on the site to register instead of buying it. It's simple enough, but I'll never have to go back... It's frustrating..."

⁵² See Amy Schade. "Designing for 5 types of e-commerce shoppers." <http://www.nngroup.com/articles/ecommerce-shoppers/>

And even people who have an account may forget their credentials or may not have them handy — a situation frequently encountered on mobile. (Yes, recovering the password is an option, but a tedious and time-consuming one.) To give users maximum flexibility, we recommend that sites implement a guest-checkout alternative that permits users to complete the purchase without creating an account or registering.

Amazon was guilty as charged: its mobile site did not have a guest-checkout option. Probably the majority of Amazon's customers are frequent shoppers, which is why Amazon's UX requirements are different than those of most sites⁵³. Nonetheless, that doesn't mean that these shoppers can remember their passwords. In fact, we have observed many Amazon users who could not log in to their account on their phone. They were often users of Amazon on a different channel⁵⁴ (e.g., desktop browser) and their passwords had been automatically stored in that channel, so they did not get a chance to use their credentials often and thus remember them.



Amazon.com did not allow its users to check out as guests.

⁵³ Jakob Nielsen. "Amazon: No Longer the Role Model for E-Commerce Design." <http://www.nngroup.com/articles/amazon-no-e-commerce-role-model/>

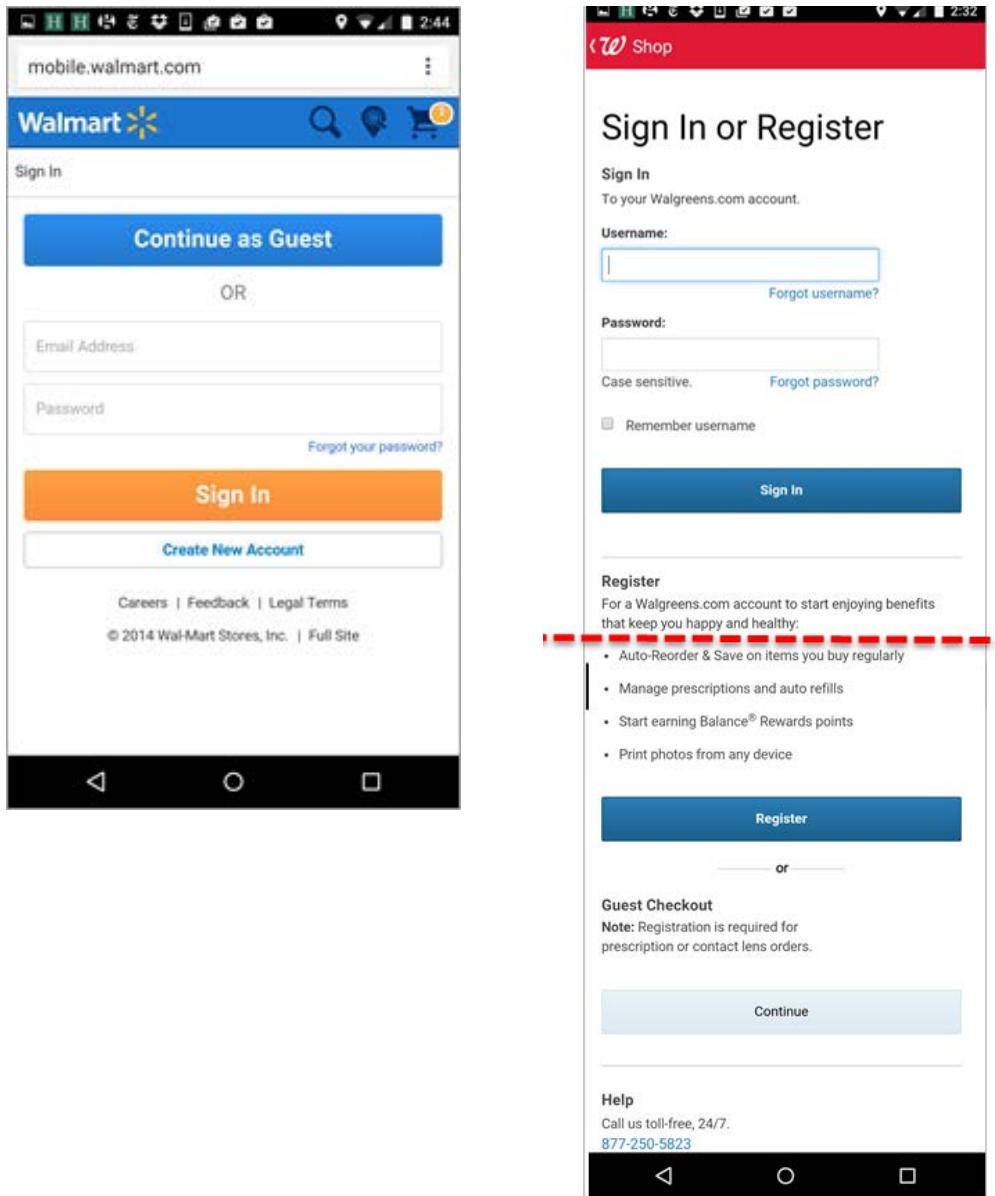
⁵⁴ Nielsen Norman Group. "Consistency in the Cross-Channel Experience." <http://www.nngroup.com/articles/cross-channel-consistency/>

Here is another user commenting on guest checkout:

"I like that they have an area for guest checkout, so that if you don't have an account with them you don't have to create one, which is sometimes easier, especially if you have (like I do) a bunch of accounts [...] I keep my passwords on a piece of paper next to my computer."

175. The guest checkout option should always be the first option (before log in or register) displayed on the checkout page.

The guest checkout should be the most salient option on mobile, since that is the alternative most likely to be chosen by users. Even when users have an account with the site, they are unlikely to sign in on mobile, not only because they hate typing user names and passwords, but also because, most of the time, they cannot remember their account information and they don't have access to the piece of paper or file where they've stored it.

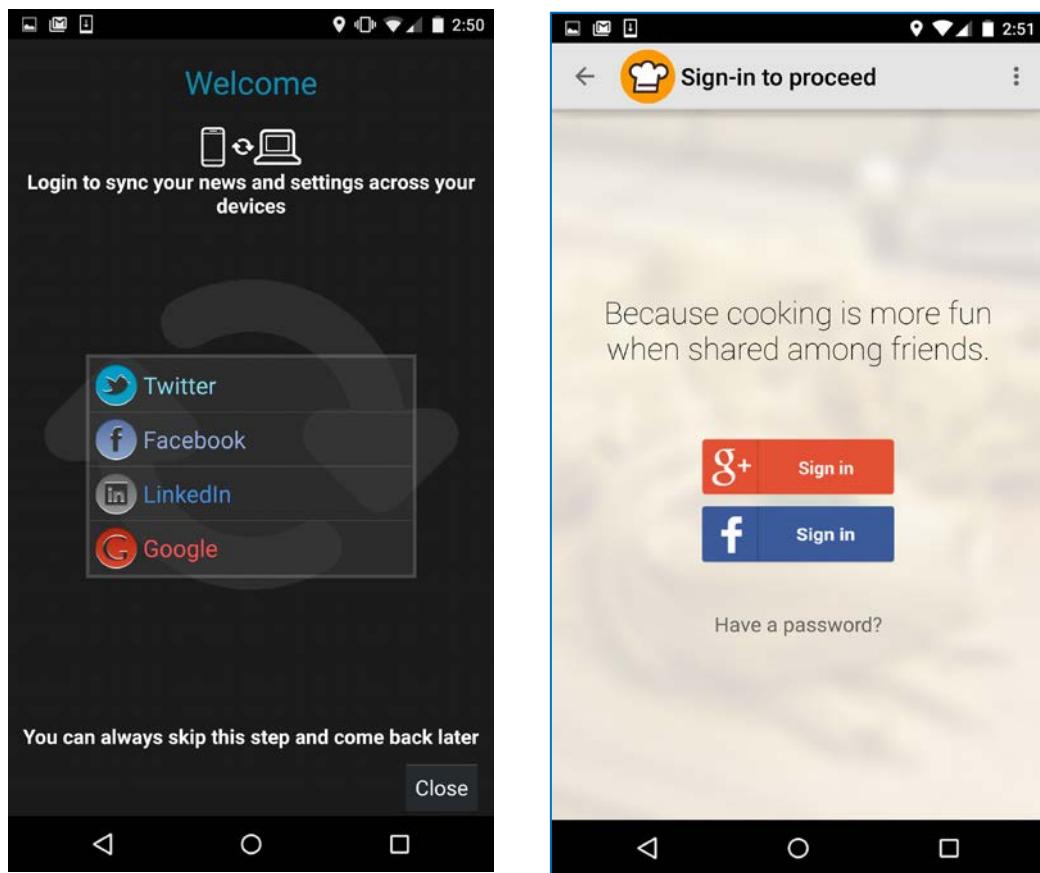


Both Walmart.com (left) and Walgreens.com (right) allowed users to skip signing in. However, Walmart rightly made it the most salient choice by listing it first. Walgreens listed the login and registration options first.

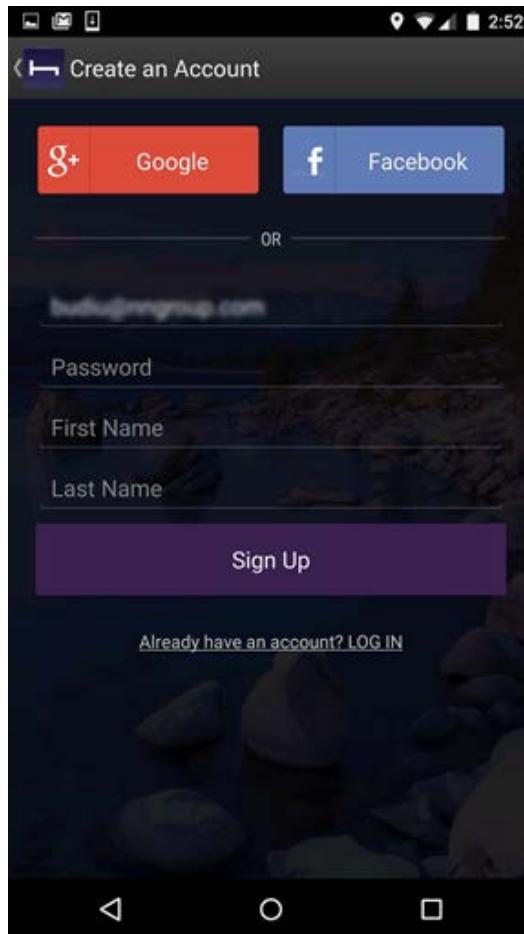
176. Offer people the option to sign in with a Facebook or Google account, but don't make those be the only options available.

Many users will never use a social login for privacy reasons, but some will. For those people who are willing to use a social login, the option to do so can save them a lot of grief because (1) they can skip a long registration process; (2) they probably use their Facebook or Google credentials more often and they will likely remember them.

Because some people are suspicious when it comes to sharing their Google or Facebook (or Twitter, or Amazon) identities, it's never a good idea to make social login the only way into an app. And although Google and Facebook may be used a lot, some people rarely log in or out of these accounts, and may still be unable to remember their password even if they wanted to. For instance, Paper for iPhone (a news app designed by Facebook) required users to sign in with their Facebook account. However, some of our users mentioned that they normally don't use Facebook login on other sites and had a hard time remembering their password.



Taptu, a news app (left), and Cookpad, a recipe app (right) for Android did not provide an option to sign in with email.

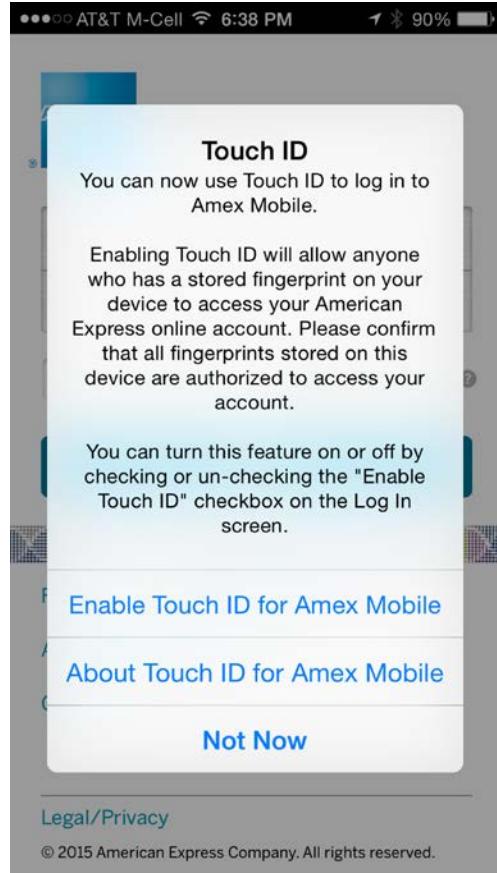


Hotel Tonight for Android allowed users to either use a social sign in or create a separate account.

177. [iOS] Offer users the option to log in using their Touch ID.

178. [iOS] Do not force users to log in using their Touch ID.

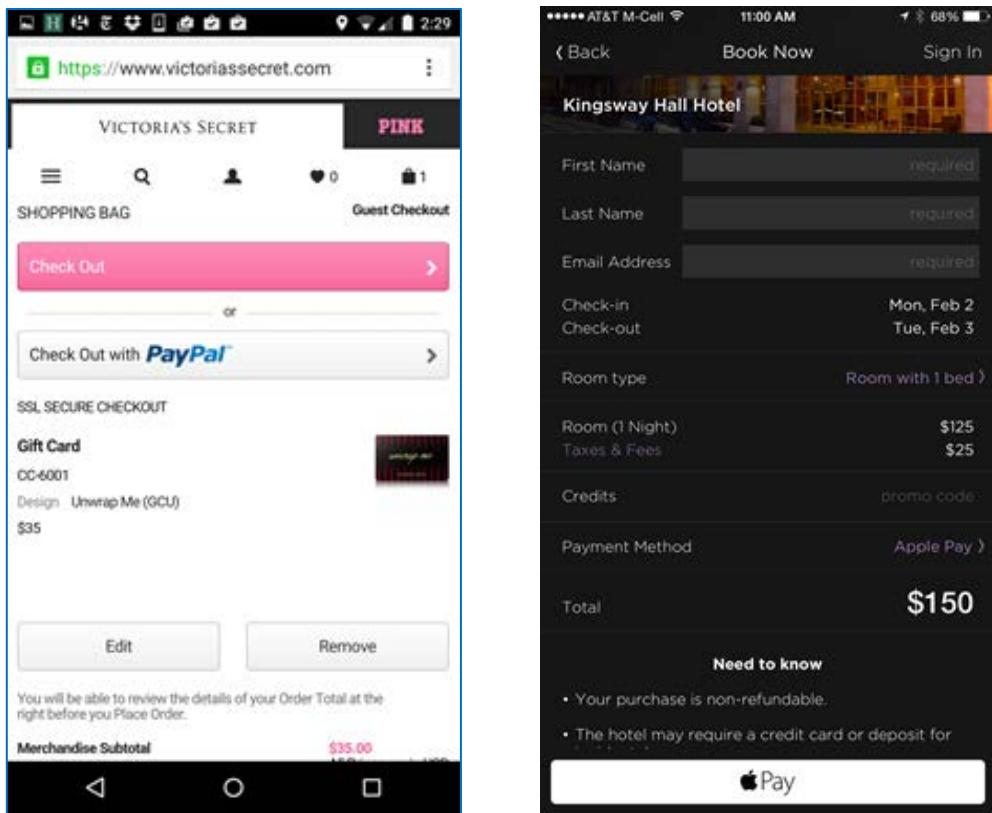
A finger press is incomparably easier than typing a username and password. Although, as with the social login, many users are still reluctant of allowing access to their fingerprints, some will happily take advantage of that feature. While it's definitely minimizing the interaction cost, it should however not be the only registration/login option available to users.



American Express for iPhone allowed users to authenticate with their Touch ID.

179. Allow users to checkout with PayPal, Apple Pay, or Google Wallet, but do not make those the only options.

For e-commerce sites, checking out with one of these options is a huge convenience for users, as they are spared the effort of entering their information, as well as long credit-card numbers. Allow users to take advantage of these options, but don't assume that everybody will use them or will be able to use them.



Victoriassecret.com (left) and Hotel Tonight for iPhone (right) sped up the checkout by allowing users to use PayPal or Apple Pay.

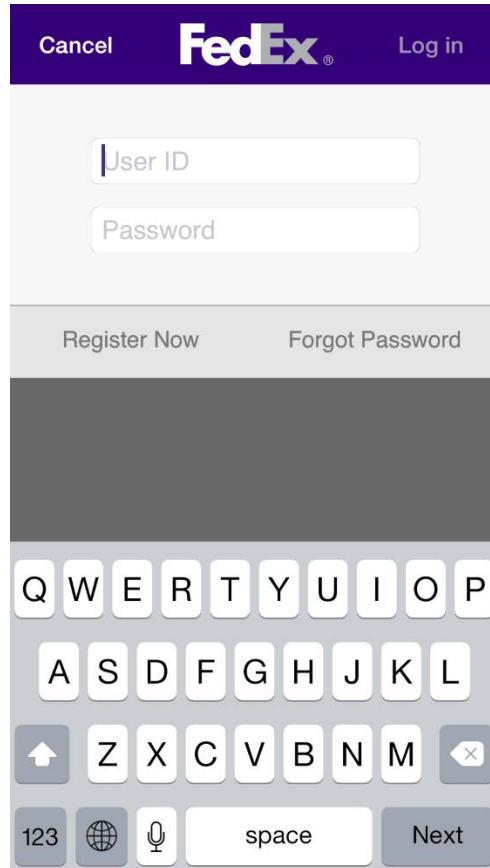
180. Place both the *Sign In* and *Register* buttons under the login form, one next to each other on the same line.

This guideline derives directly from the Gestalt law of proximity and from guideline 113. We've seen countless users clicking the wrong button when they needed to sign in or register. Invariably, users who don't have an account end up attempting a login and, vice versa, users who have an account end up registering again. The problem is typically that the two main options (register and sign in) are not given the same priority, and users just pick the one that is more salient.

When presented with a login form, most users don't think of whether they already have an account — they just put in an email and password and click *Sign In*. The form acts as a user magnet. To quote one of our users:

"I always forget creating an account."

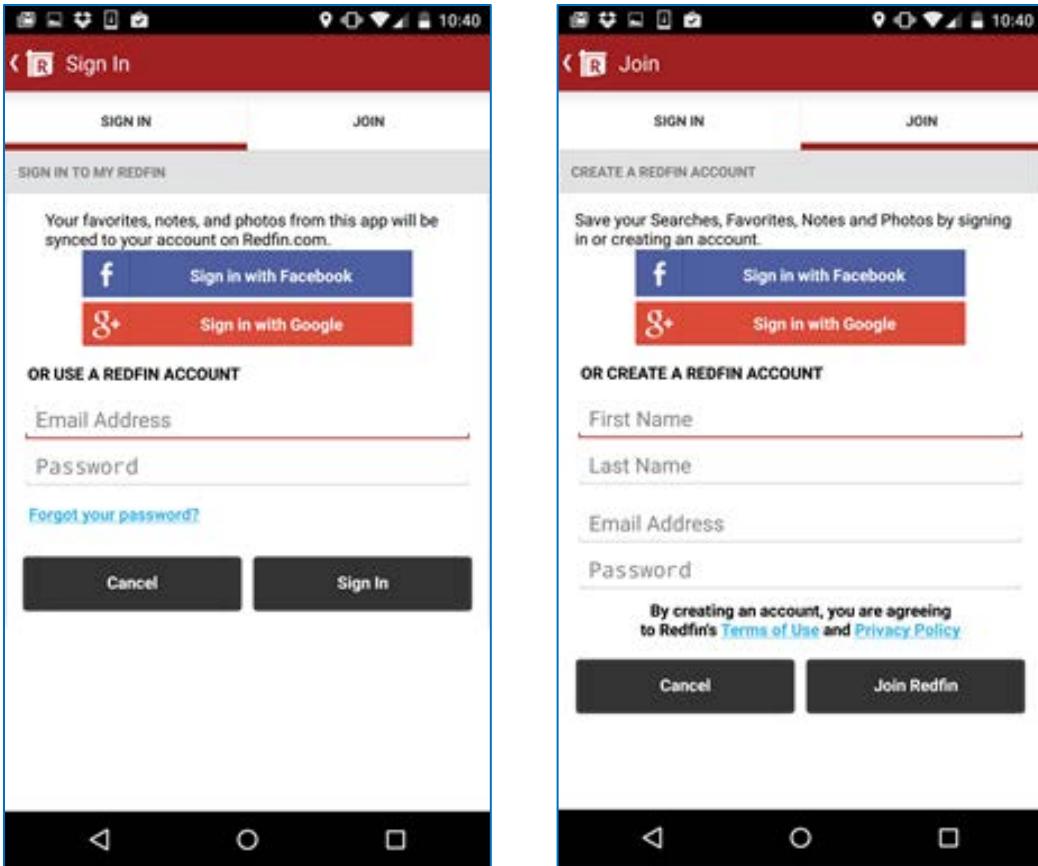
A study participant created an account the first time he used the FedEx app, but then, when he tried to log in with his new account, he kept clicking the *Register now* button and being taken back to the registration process. The register button was under the login form, while the login button was farther away, in the top left corner.



FedEx placed the *Log in* button that submitted the login form at the top of the screen, and the *Register now* button just below the form.

Because the *Log in* button was far away, customers with an account would not notice it and press the *Register now* button, only to be asked to register again.

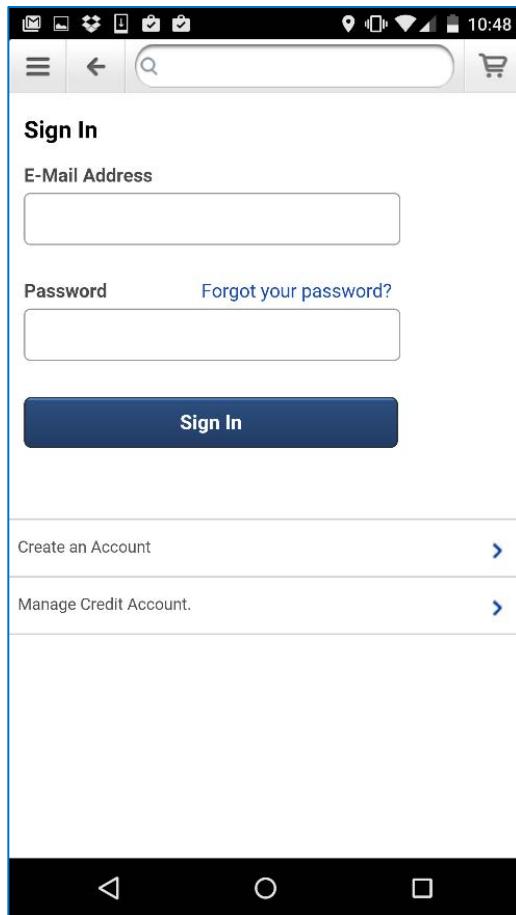
Redfin below directed users automatically to the sign in form. The option to join was presented as a tab at the top. This design suffered from the same problems as FedEx, but it hurt first-time users rather than account holders.



Participants using Redfin for Android ignored the tabs at the top and tried to sign in even though they did not have an account.

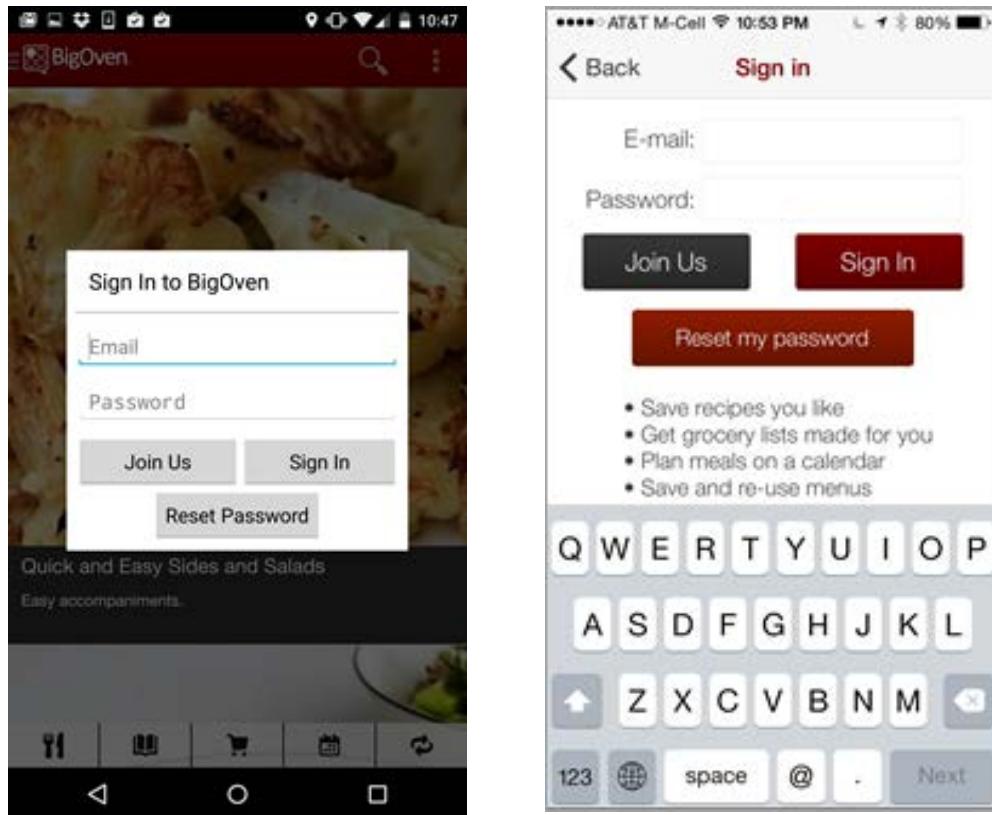
The correct solution in this situation should give equal priority to both login and register.

Unfortunately, just putting the *Register* button under the *Login* button (like Best Buy did in the example below) doesn't always cut it: users tend to press the button right under the login form, without bothering to notice that there are more options below.



Best Buy for Android: Placing the *Create an Account* option under the *Sign In* button made it less salient and was not a good strategy for minimizing user errors.

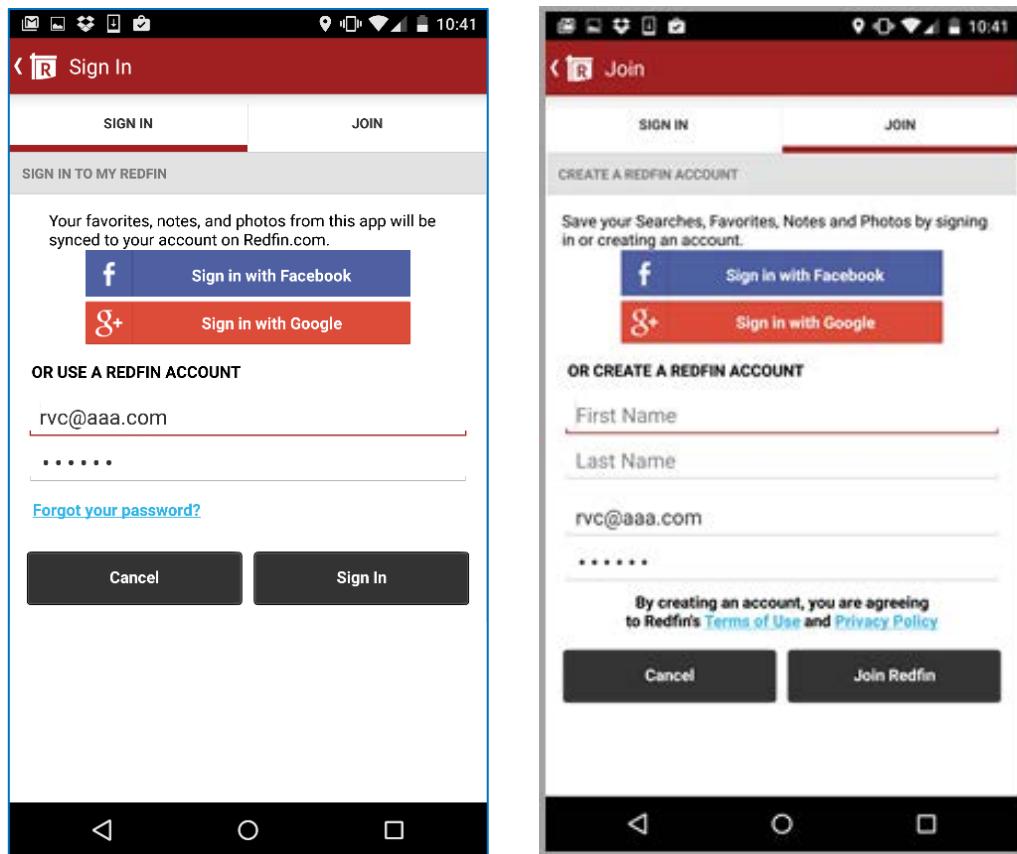
The design used by Big Oven is the preferred one, where the register and login options are placed next to each other.



Big Oven for Android (left) and iPhone (right) placed the *Join Us* and *Sign In* buttons next to each other, making it easier for users to pick the right choice.

181. If users make an error trying to sign in when they don't have an account, take them to the registration form, but save the information that they had entered in the login form.

This is such a common error, that it's worth paying attention to it and minimizing its impact. If a user accidentally enters an email and password in the login form, although she does not have an account, save that information to prepopulate the corresponding email and password fields in the registration form. This is exactly what Redfin did: the same email and password information erroneously entered in the login form were reused in the registration form.



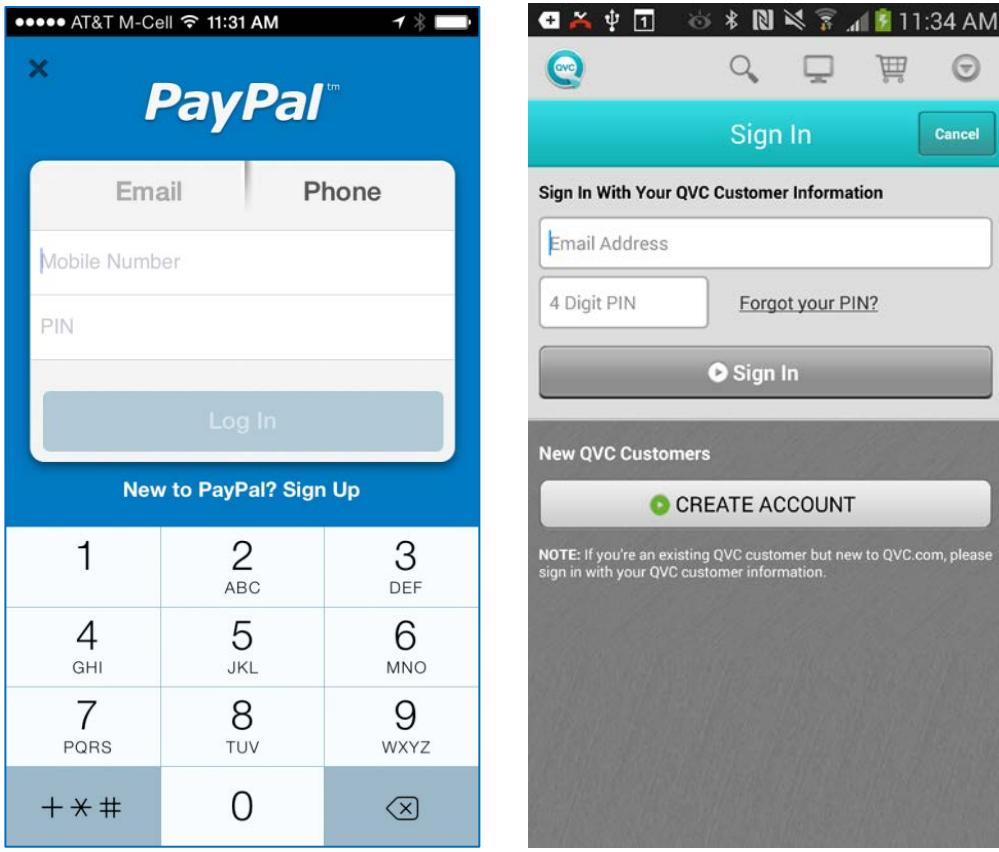
Redfin for Android: The information erroneously entered in the *Sign In* form was saved and transferred to the *Join* form.

Passwords

- 182. When logging in and out must be done for each session, let users define a numerical pin.**

Sometimes users need to log in. If that is the case, you can make password typing easier by lessening the constraints normally imposed on passwords: if users only had to enter 4 numerical characters, they would not have to juggle multiple keyboards and the chance of making a mistake would be smaller.

PayPal allowed users to define a PIN to be used only to log in to their mobile app; so did QVC (both on their mobile apps and on their website).

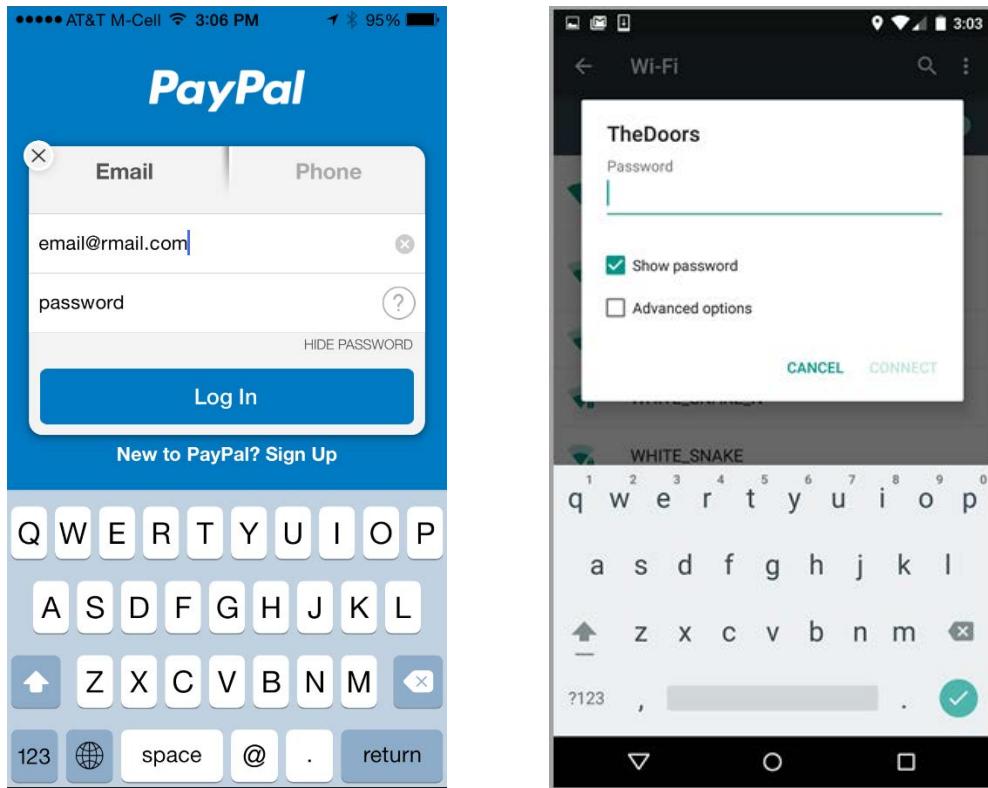


PayPal for iPhone (left) and QVC for Android (right) allowed users to use pins as passwords.

Unfortunately, one of the disadvantages of defining a PIN (on top of a regular password) is that users have to remember one more thing.

183. When logging in must be done, have an option that allows the user to see the password in clear.

What makes it even harder to get passwords right on mobile is that users cannot see what they type. There is no real reason for masking passwords. Advocates of password masking argue that someone else may be snooping over the user's shoulder and may read the password. That problem has an easy fix: allow users to toggle between masking the password and typing it in the clear — they can judge whether they have enough privacy to unmask the password or not.



Users should decide if they want to see their password. Both PayPal for iPhone and the Wi-Fi Settings in Android allowed users to see their passwords in clear.

184. When users need to create a password, tell them if the password has to satisfy any constraints (e.g., password length).

Progressive disclosure has its place in a variety of interface-design contexts, but it should not be applied when instructing users how to create a password. Some may think that disclosing password requirements is unnecessary and puts an extra burden on the users (since they have to read them). That last bit is certainly true, but unfortunately revealing them is a necessary evil: password requirements are not standardized enough (some sites are happy with 4 characters while others require at least 6; some insist on capital letters, others don't, and so on so forth), so users don't know what any particular app or website will require.

One of our users was trying to create an account in the Best Buy app. Unfortunately, the registration form did not warn him that the password had to be at least 6 characters long, so when he chose a shorter password he got an error and had to go back and create another password. Not only did the user take an extra step, but he also had to remember what the password format needed to be. If a password must be created on mobile, it's better to make the process as smooth and fast as possible.

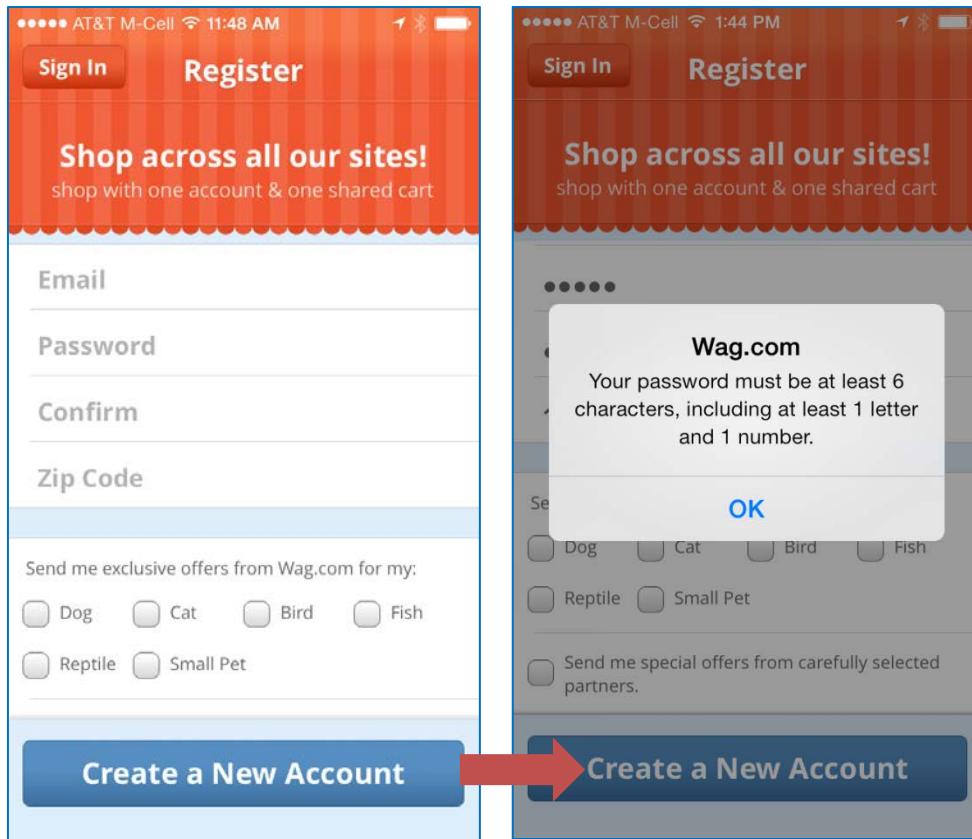
The image displays two side-by-side screenshots of a mobile application's account creation screen. Both screens show the following fields:

- First Name: r
- M.I.: M.I.
- Email: budiu @ [redacted]
- Phone: 94539
- Create a Password**:
Password must be 6-30 characters.
Current password:
Re-enter Password:
- RewardZone®**:
RewardZone® ID: Optional >
- Legal Information
- Create button

In the second screenshot (on the right), a dark gray callout bubble appears over the "Re-enter Password" field, containing the text "Passwords must contain a number". A red arrow points from the first screenshot to the second, highlighting the change in the password requirement.

Best Buy users were first told that they must create a 6-30 character password; after they had typed their new password twice (!), they were also told that the password must contain a number.

Best Buy has since fixed this problem. However there are still many other apps and websites that still do not disclose password requirements unless the user chooses the wrong password.

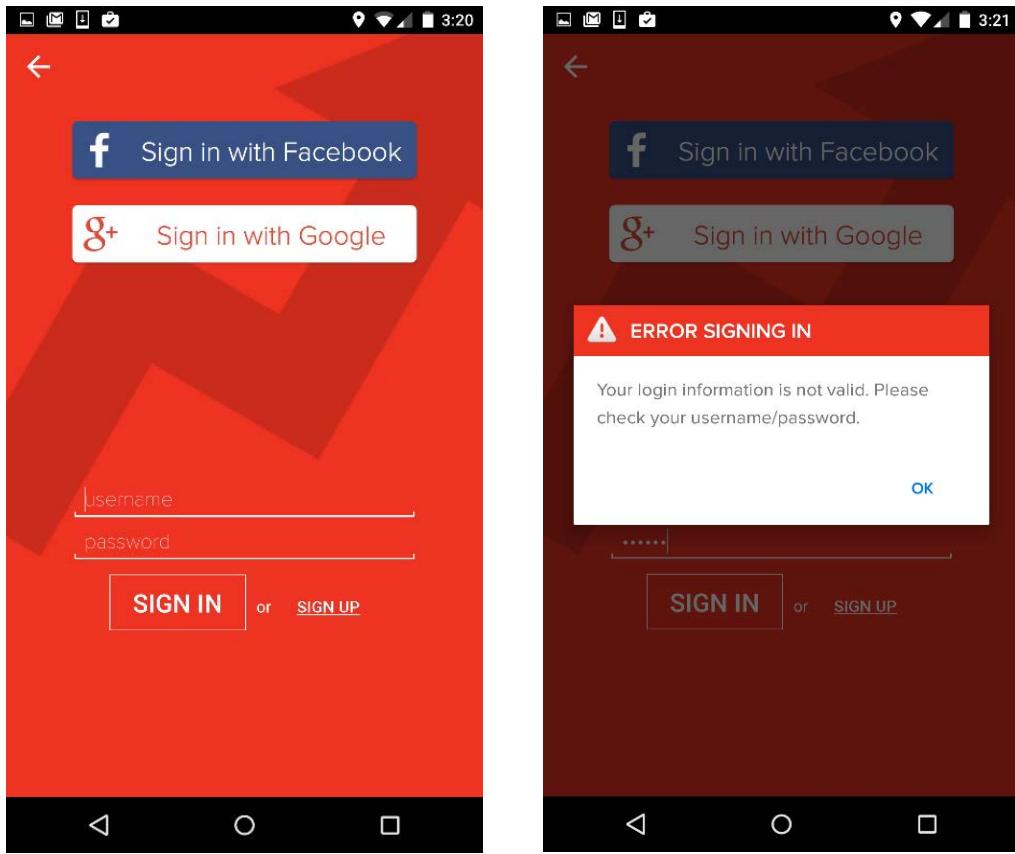


Wag.com did not tell its users what kind of password they had to choose unless they made an error.

185. A login form should include a *Forgot password?* link to allow users to recover their password.

Since users often have a multitude of passwords that they need to keep track of, and since they don't always store passwords on the phone, the *Forgot password?* link should offer a last resort for those people who need to log in but cannot.

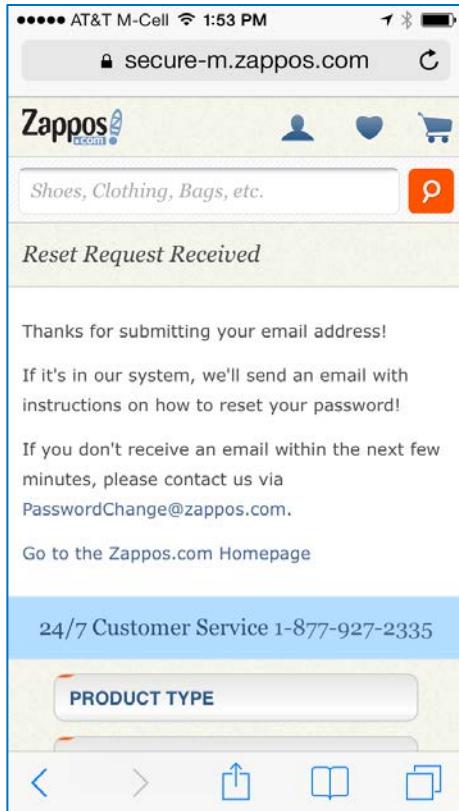
One of our study participants attempted to log in to BuzzFeed, a news app, but could not do so because he did not remember his password. The app did not offer him the option to retrieve a forgotten password.



BuzzFeed for Android: Users could not retrieve their forgotten password.

186. If users must submit an email address to recover their password, tell them if the email is not in your database.

People often have multiple email addresses, and they may not remember which one they had used on your site. Always tell them if an address is not in your email database, so they can try a different email.

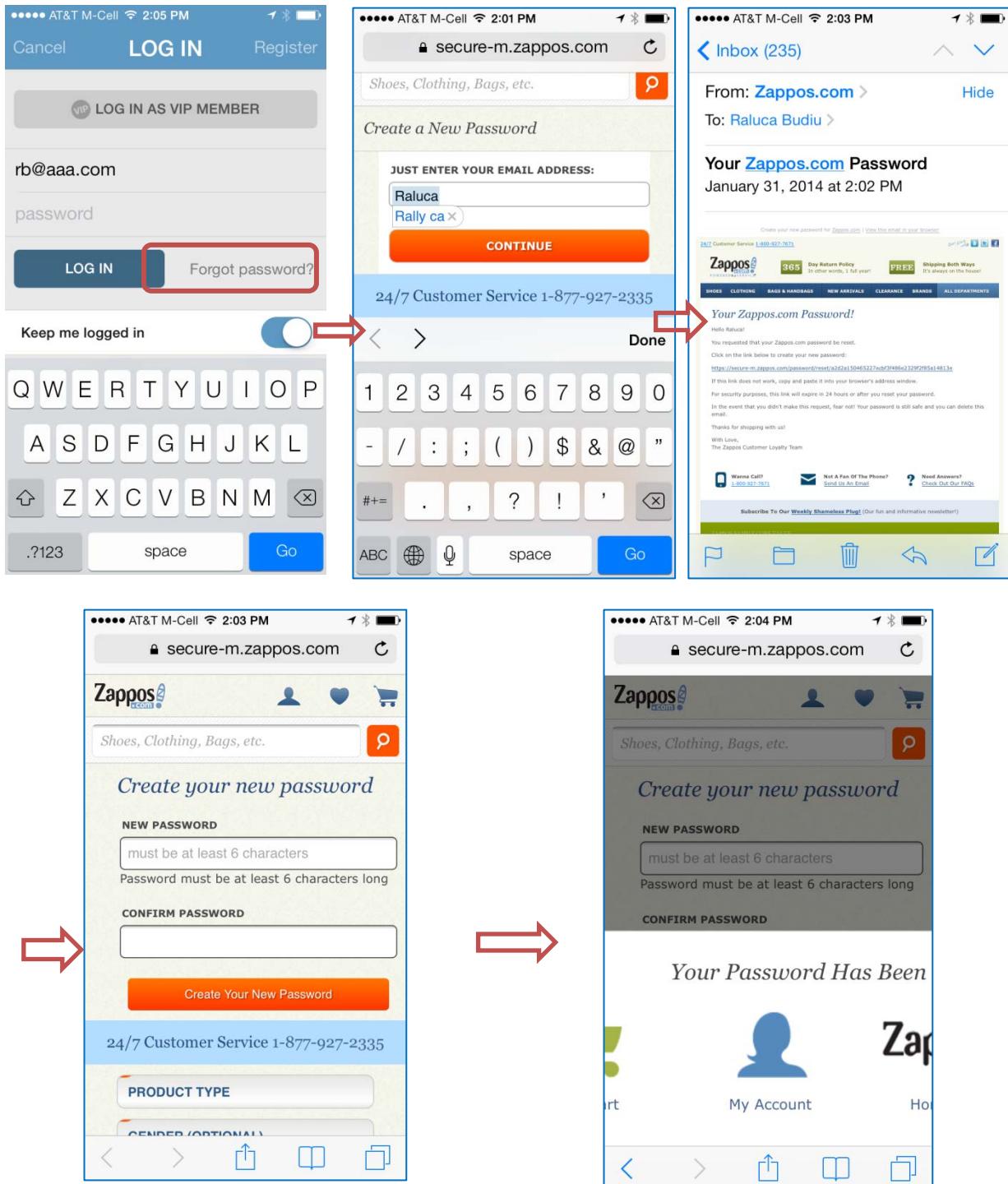


Zappos.com required users to enter an email to reset their passwords. However, the company did not share whether the email was in its database. Instead, users had to wait “a few minutes” and then try again or email technical support.

187. Avoid taking users out of the app or website to recover their passwords.

When users tap *Forgot Password?* in an app, they often have to go through several steps. First, they must go to their email app and wait for an email from the site or app. Second, they need to tap on a link in the email. The link usually takes them to a website where they can reset their password. And, last but not least, users must remember to go back to the app and log in with their new password.

Each of these steps is an opportunity for the users to get lost and give up. For instance, if they started the process in an app and then found themselves in a browser, they may not remember to return to the app (and may not even realize that they are now interacting with the company through a different channel). We often saw people resetting their password in the browser, and then expecting to find the content of their shopping cart (from the app) right there, on the company’s site. When that didn’t happen (unsurprisingly, since the website was not aware of the activity in the app), users got frustrated and annoyed, and frequently they did not realize what caused the problem.

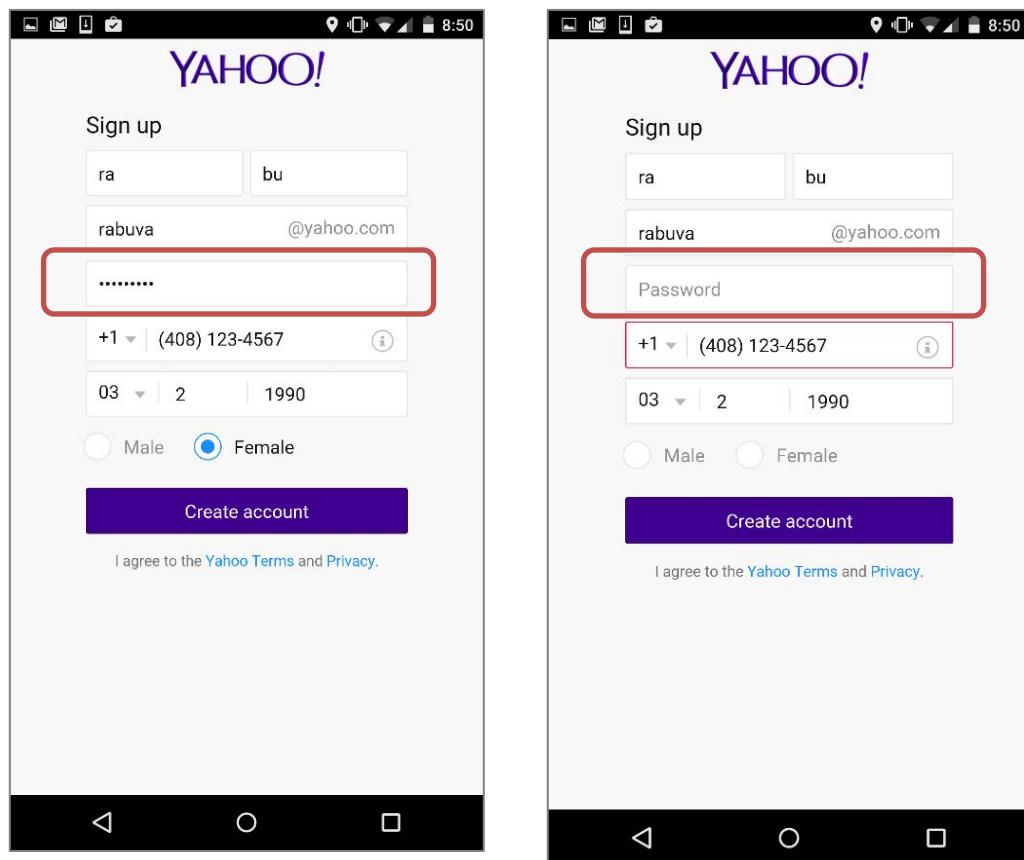


When users clicked the *Forgot password?* link in the Zappos app, they were taken to the Zappos website (in the Safari browser). Next, they had to check their email and click on a link that took them to the website again. Once the new password was set, users were left on the website and had to find their way back to the app.

188. If user made an error in the log in or registration form, do not clear the password field. Do not make them reenter their passwords from scratch.

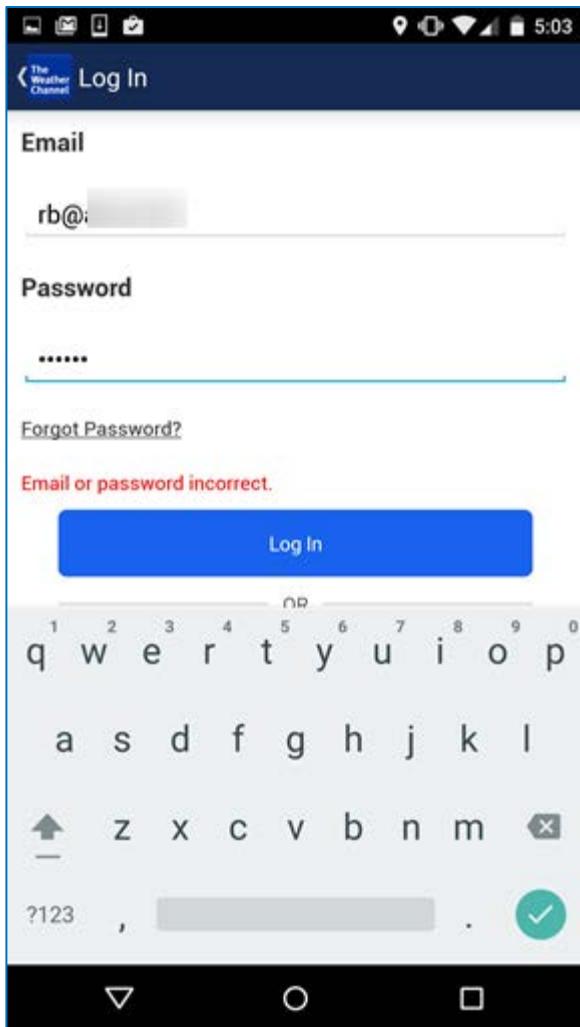
This is a version of guideline 122, which says to always return to the state before the error. Remember that it's easier to modify than to start from scratch. It's even more important to follow this guideline when it comes to password fields, which are so difficult to type correctly.

Yahoo! Weather's registration form cleared the password field each time the user made an error in any of the other fields. Thus, if the phone number was wrong, the app returned an error and also removed the information entered in the password field, forcing the user to type it again.



Yahoo! Weather for Android deleted the password field, although there was no error related to that field. The error was caused by the invalid phone number.

Even if the password field was the one causing the error, it should not be erased. If users have the ability to see the password in clear, it will be easier to edit the field than to type the password again from scratch.

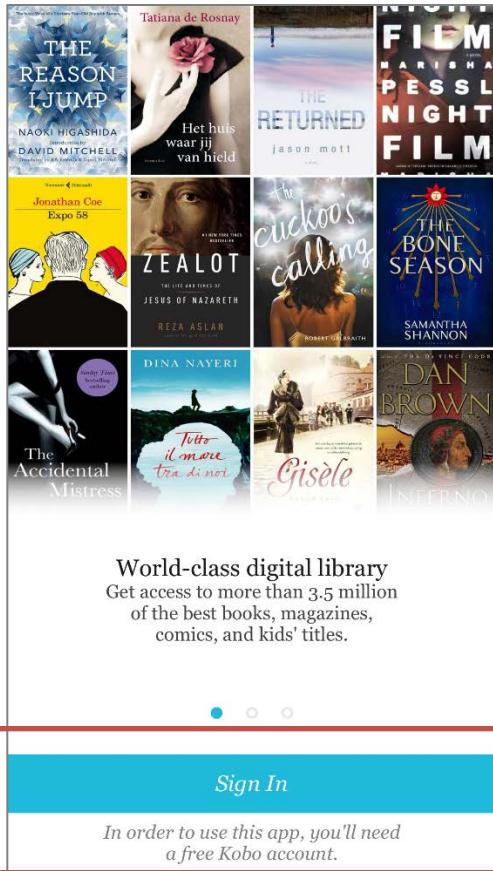


Weather Channel for Android did not erase the password field even though the field was incorrect. It would have been, however, easier to modify it if there was an option to show the password in clear.

Registration

- 189. If you have a login wall, allow people to register within the app or mobile website.**

Sometimes people will be really motivated to use the app in spite of your efforts to deter them by creating a login wall. They may even be willing to register. If you are in that fortunate position, allow them to create an account in order to be able to pass through the login wall. Don't force them to use a computer to create the account.

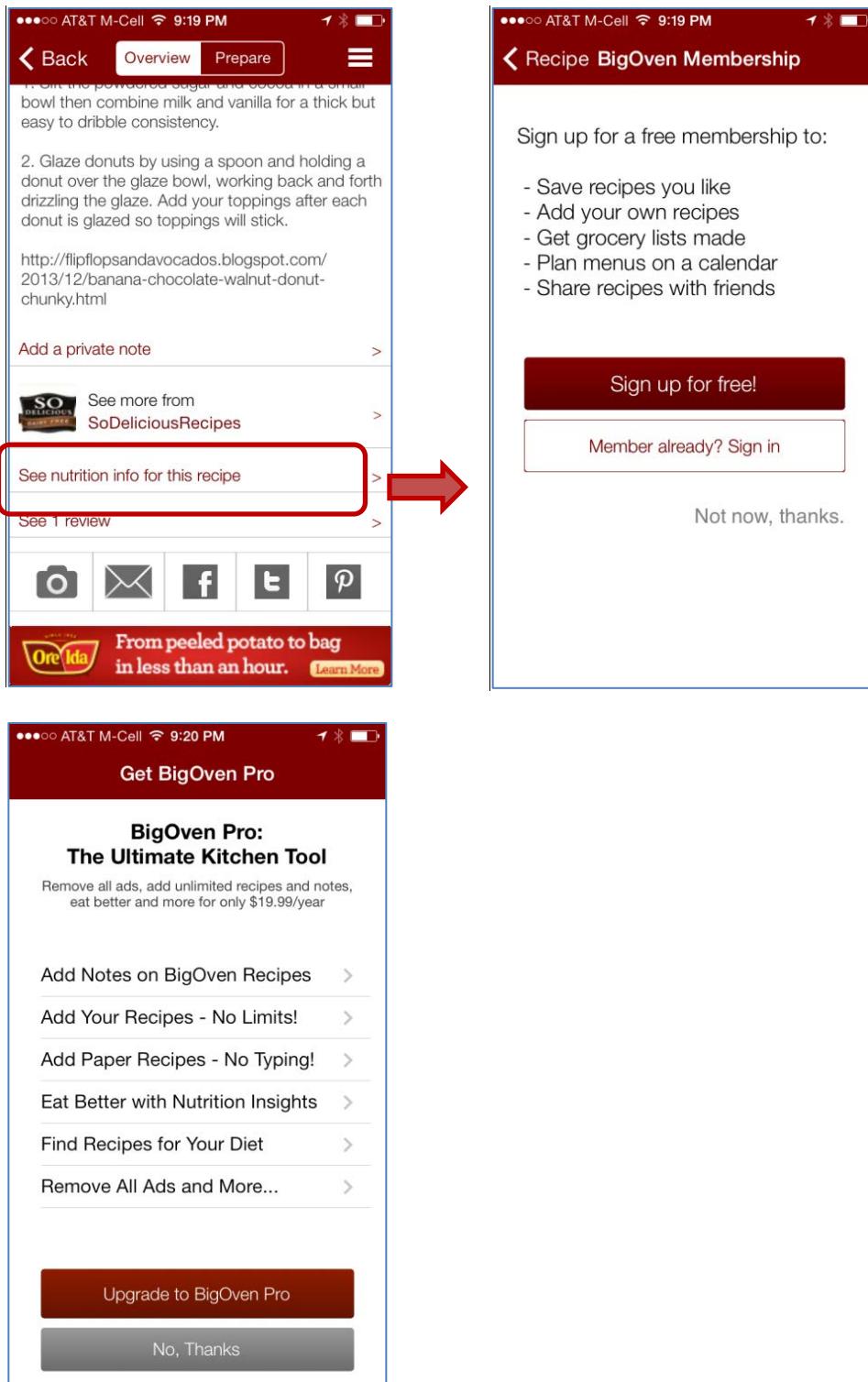


Kobo, a book reader, did not allow people to preview or use the app if they did not have an account. Users could not create an account within the app. (Nor did the app provide them with a link to the webpage where they could create an account.)

190. Be upfront about the benefits that registration brings to the users.

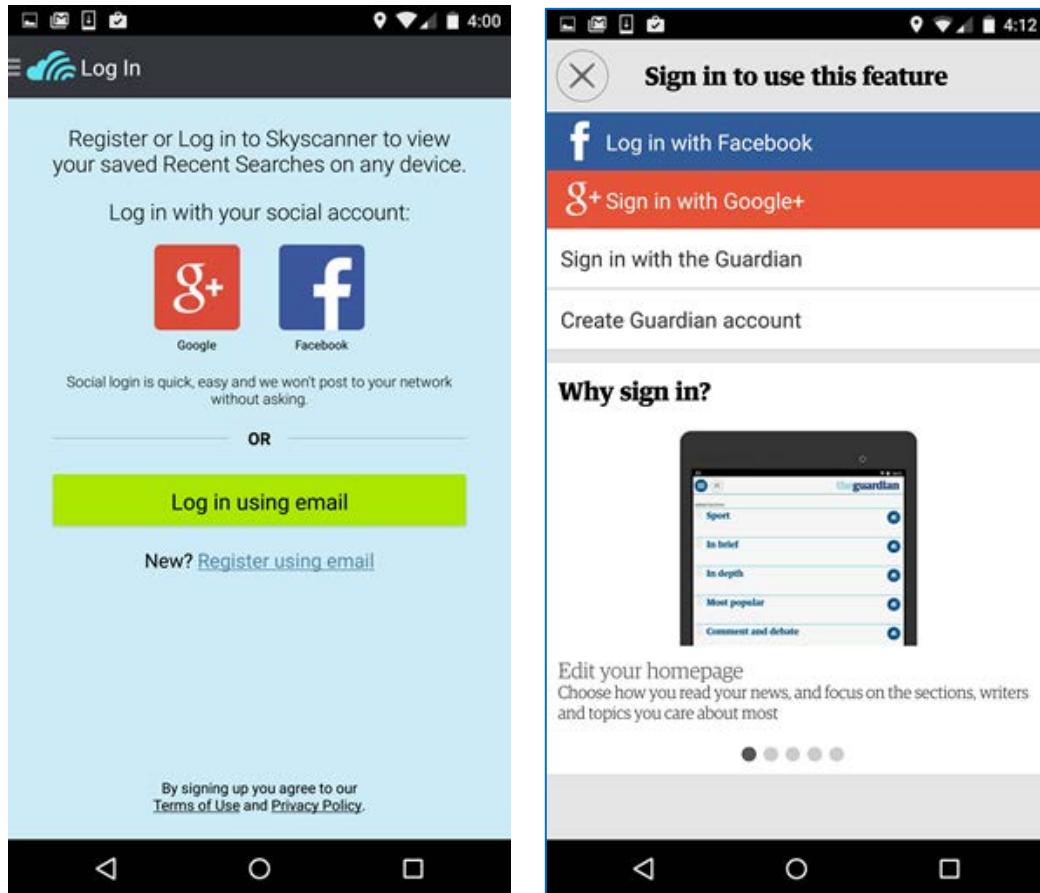
Whenever you ask users to register, tell them what advantages they get by doing so. If there are features available only to registered members, tell them what they are. But don't mislead them into thinking they can get a feature by registering, only to find out later that they need to pay a fee on top of the registration to get it.

When users tried to get calorie information for a recipe in Big Oven, they were shown a login screen, which led them to believe that they could access that feature if they registered. However, once they completed the painful registration process, they discovered that a paid "pro" membership was needed to access the nutritional information.



Big Oven for Android asked users to register “for free” to access caloric information, but then informed them that they need a paid “pro” membership.

The Guardian and Skyscanner were upfront about the advantages offered by the registration.



Skyscanner for Android (left) and The Guardian for Android (right) told users what benefits to expect from registration.

191. [App] Avoid taking people out of the app for registering.
192. [App] If registration is not fully supported in your app, consider using a browser view within the app to display the full site. Once users finish the registration, take them back to the flow they were in before starting it.

For reasons similar to those listed in guideline 187, taking users to a browser to complete registration can be disorienting and can cause them to forget to go back to the app. Whenever possible, allow users to register within the app.

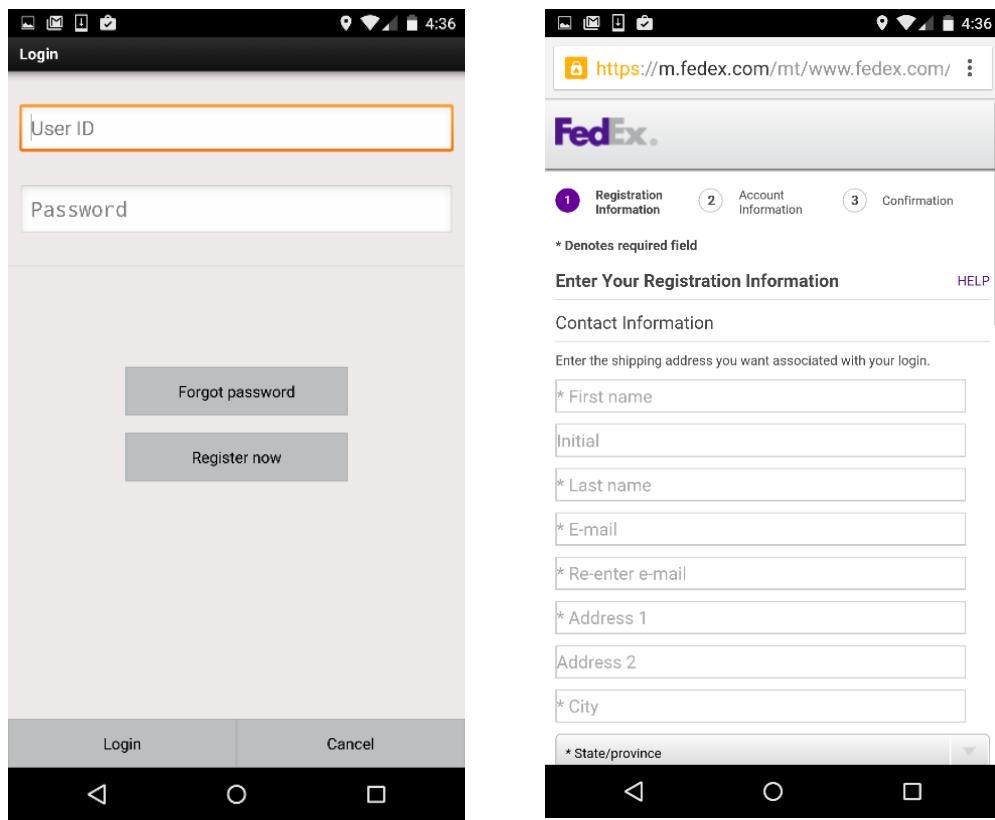
Many apps correctly assume that people will not take the trouble to register on a phone, so don't have any app-optimized pages for registrations. In those situations, they simply send the users to their mobile (or full) site.

This cost-benefit analysis makes this solution reasonable. However, when implementing it, it's best to do it within the app. Namely, instead of sending the user to Safari, Chrome, or Internet Explorer, have a browser view within

the app that will present the web page to the user. Once the registration is completed, take users back to the flow they were originally in (for instance, if they were purchasing something, take them back to the checkout flow).

If the registration is done in a separate browser app, users are burdened with the task of recovering the flow — they need to go back to the app, and enter their new credentials. We noticed that when people were taken to the web to register, they often forgot to come back to the app. Most of the time, they just completed (or failed) their task on the mobile site.

FedEx did not support registration in its Android mobile app and asked users to go to its mobile site to complete a lengthy registration. Once the registration was completed, users had to remember to go back to the app and enter their newly created credentials.



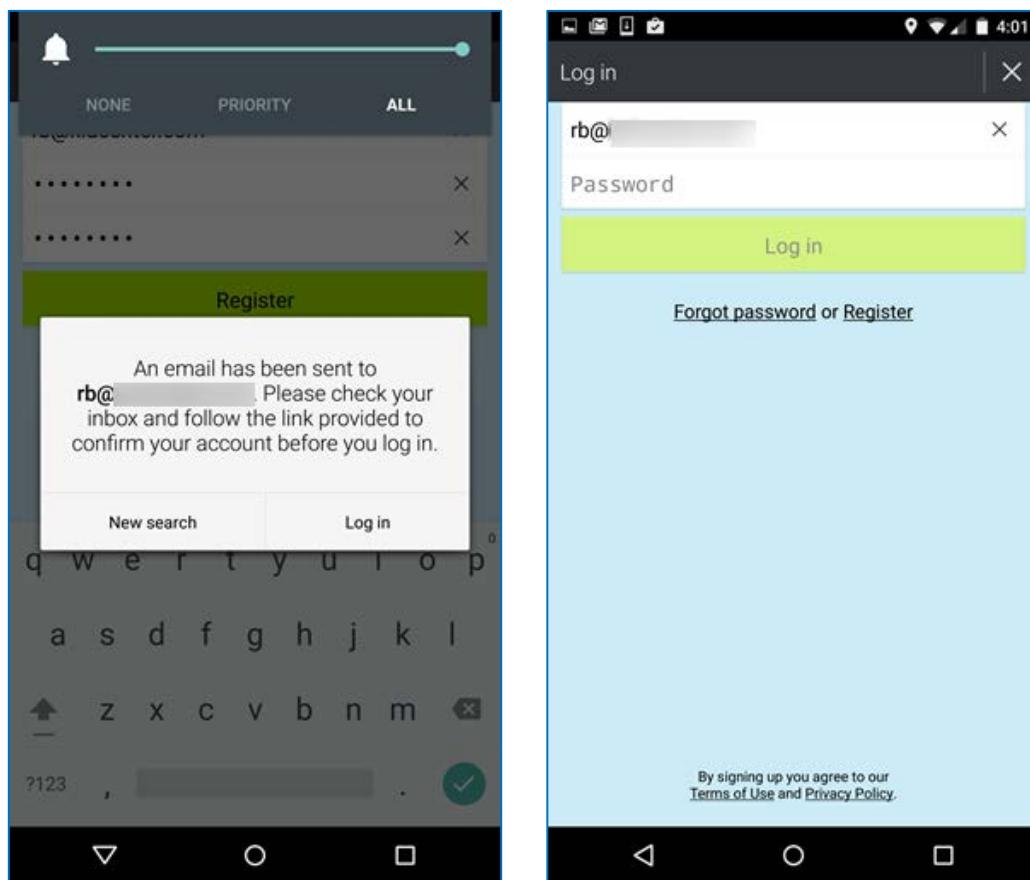
FedEx users who tried to register in the Android mobile app were taken to a mobile web page outside the app and then had to remember to go back to the app.

193. Don't ask users to confirm their registration through email.

Registering on mobile is a tedious process in and by itself. You don't want to add any extra hurdles for users by asking them to confirm their email. Not only does the email confirmation increase the users' work load, but it also disrupts their task flow: they have to go to a different app, read their email, click on a link in the email, and get taken to a new web page.

- 194. If users have just completed registration in an app or on a website, they should be considered logged in and should not have to log in again.**

Occasionally sites and apps separate registration from login and require users to log in immediately after they have registered. This creates extra work for users and is unnecessary.



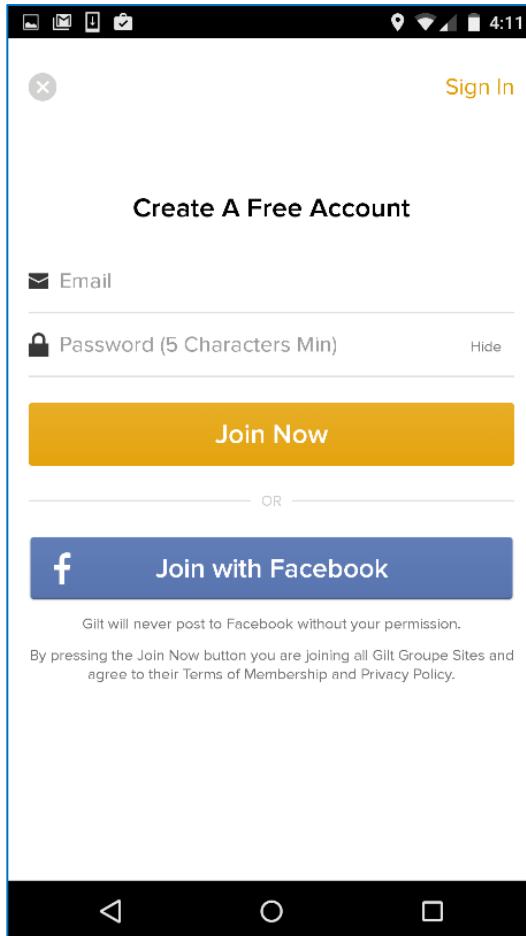
Skyscanner for Android required users to go through an email verification (left) and then log in (right) immediately after registering.

In contrast to Skyscanner, which required users to log in immediately after they had registered, once users registered with The Guardian for Android, they were automatically logged in and allowed to use the app.

- 195. Whenever registration is an option on mobile, require only the minimum information to create an account. Let users add extra information at later times.**

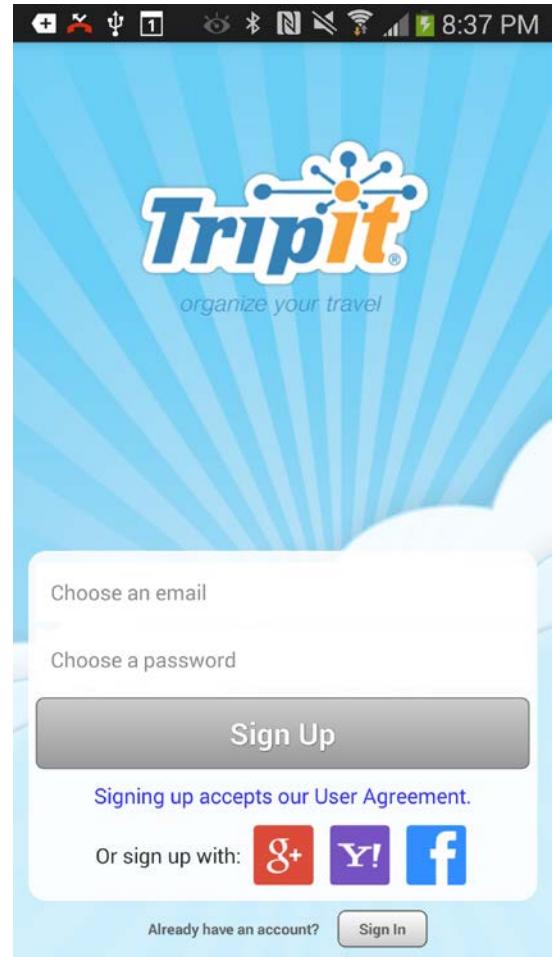
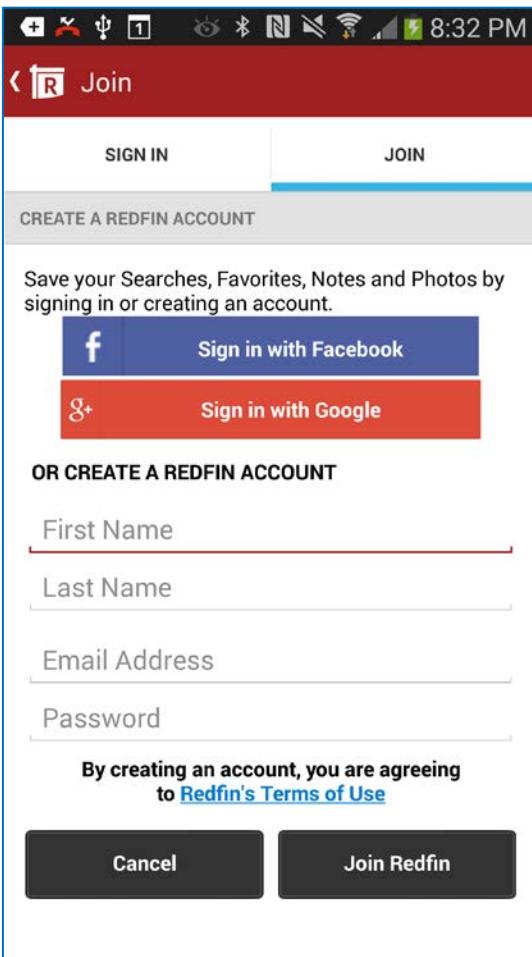
In user testing, we've never seen a user who was happy to have to register to a site or app. The more complex the registration form, the more time and effort will be needed to complete it, and the more likely it is that the user will make an error.

Mobile registration forms should include a minimum amount of fields: ideally just an email and a password (as in Gilt's example below). Any information that is not absolutely necessary should not be included in the registration field; instead, it should be delegated to a profile that the users could complete whenever they want.



Gilt for Android required users to enter only two fields (email and password) in order to create an account.

Gilt's form was among the most economical we've seen. Most sites or apps ask a lot more questions that they are unlikely to use, starting from address and phone number and ending with birth date, security question, or title. Even if that kind of information might be useful, if it's not essential — don't include it. It's better to ask users at a later time (preferably on a desktop) to fill in that information. And, always, if you ask for confidential information such as birth date or social-security number, tell users why you need it and how they may benefit by sharing it with you.



Redfin and Tripit for Android also requested very little information at signup.

In contrast, Yahoo! Weather required users to enter a lot of apparently irrelevant information to register for a weather app: birth date, mobile phone number, and first and last name. Similarly, fedex.com had a 3-step registration form that required a physical address, a username separate from the email address, and a security questions. Some of this information could have been requested on an as needed basis (for instance, if the user wanted to schedule a pickup at the home address or wanted to pay with a credit card).

Yahoo! Sign up

First name Last name

Yahoo username @yahoo.com

Password

+1 | Mobile number

Birthday

Male Female

Create account

I agree to the [Yahoo Terms](#) and [Privacy](#).

FedEx.

Registration Information

* Denotes required field

Contact Information

Enter the shipping address you want associated with your login.

* First name
Initial
* Last name
* E-mail
* Re-enter e-mail
* Address 1
Address 2
* City

* State/province
* ZIP/postal
U.S.A.
* Phone no.

Login Information

* Create a user ID
Use at least 6 characters.
* Create a password
Password must use at least 8 characters and contain one upper case letter, one lower case letter and one numeric character.
* Re-enter password
* Secret question
* Secret answer

Terms and Conditions

I have read, understood and agree to be bound by the [fedex.com Terms of Use](#). I also understand how FedEx intends to use my information. [Privacy Policy](#)

Confirm Your Selected Account Option

Open a FedEx account
Get access to the full array of shipping, billing and tracking services on fedex.com.
[More information](#)

Use my account online
Set up your existing FedEx account number for online access.
[More information](#)

Create a user ID only
Ship right away using your credit card, no account needed.
[More information](#)

Cancel **Continue**

[View Full Site](#) | [Global Home](#)
[Feedback](#) | [fedex.com Terms of Use](#) | [Security](#) | [Mobile Web by UsableNet](#)
All rights reserved. © 1995 - 2015 FedEx

Yahoo! Weather for Android and fedex.com had too long registration forms, with too many fields.

One of our participants who had to register for an account with Pizza Hut using her iPhone app confessed:

"I am not a fast typer on this [phone]... Oh, lord! Let's just say — if I was at home, I wouldn't mind filling in all that stuff, but this seems like a lot to do. [...] To be honest, these kinds of fields — I am not really keen on entering on my iPhone 'cause I don't like typing a lot. Date of birth? I hate using my date of birth; I really don't like giving that information out."

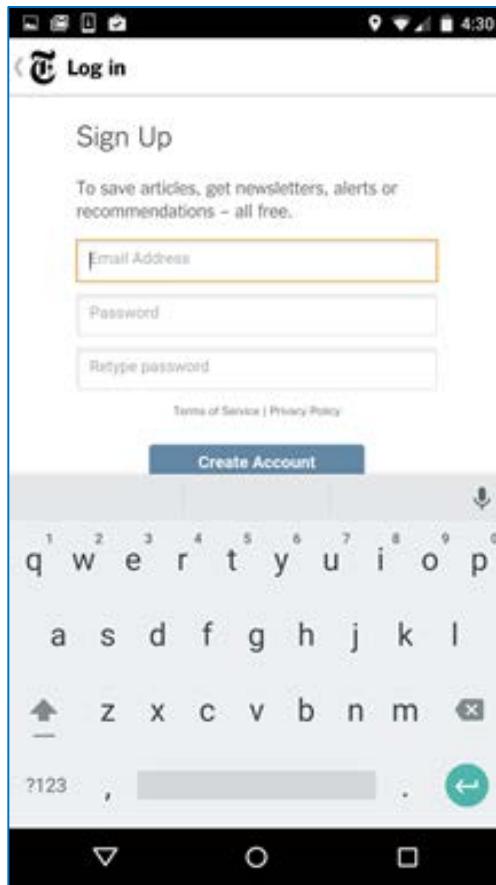
The image consists of two side-by-side screenshots of a mobile application. The left screenshot shows the initial registration screen with a red header and a white background. It contains fields for 'First name', 'Last name', 'Birthday' (with a placeholder 'Select your birthday'), 'Phone number', and 'Email address'. Below these is a section titled 'Default Address Information' with fields for 'Location name', 'Delivery Address', 'City', 'State', and 'Zip Code'. The right screenshot shows the continuation of the registration process. It includes sections for 'Delivery Address' (with 'City' and 'State' fields), 'Sign In Information' (with 'Password' and 'Re-enter password' fields, and a 'Security Question' section with a dropdown arrow), and 'Send Me Deals' (with 'Email Only' and 'Text Message' radio buttons). At the bottom of both screens are a 'Privacy Policy' link and a large yellow 'Submit' button.

Pizza Hut for Android asked users to fill in four sections of mandatory information, including birth date, mobile number, security question, and consent to receive promotions.

196. Remove duplicate fields (email, password).

Some believe that it is necessary to ask users to enter twice fields such as email and password, to remove the possibility of an error. Research has shown that simply asking users to confirm that the email and password that they entered are correct is as effective in preventing errors as typing this information twice. And, it requires a lot less user effort.

(Of course, for users to verify that their password was entered correctly requires it to be displayed as clear text, as recommended in guideline 183.)



Although New York Times for Android had a simple registration form, it could have made it even simpler if it had removed the duplicate password field.

NAVIGATION

Why should we care about navigation if we have search? Now that search has gotten good, why isn't search enough? After all, the success of sites such as Google and Amazon shows that users search; not only that, but they actually love the search box, and even on mobile, where typing is tedious and error prone, they don't shy back from searching.

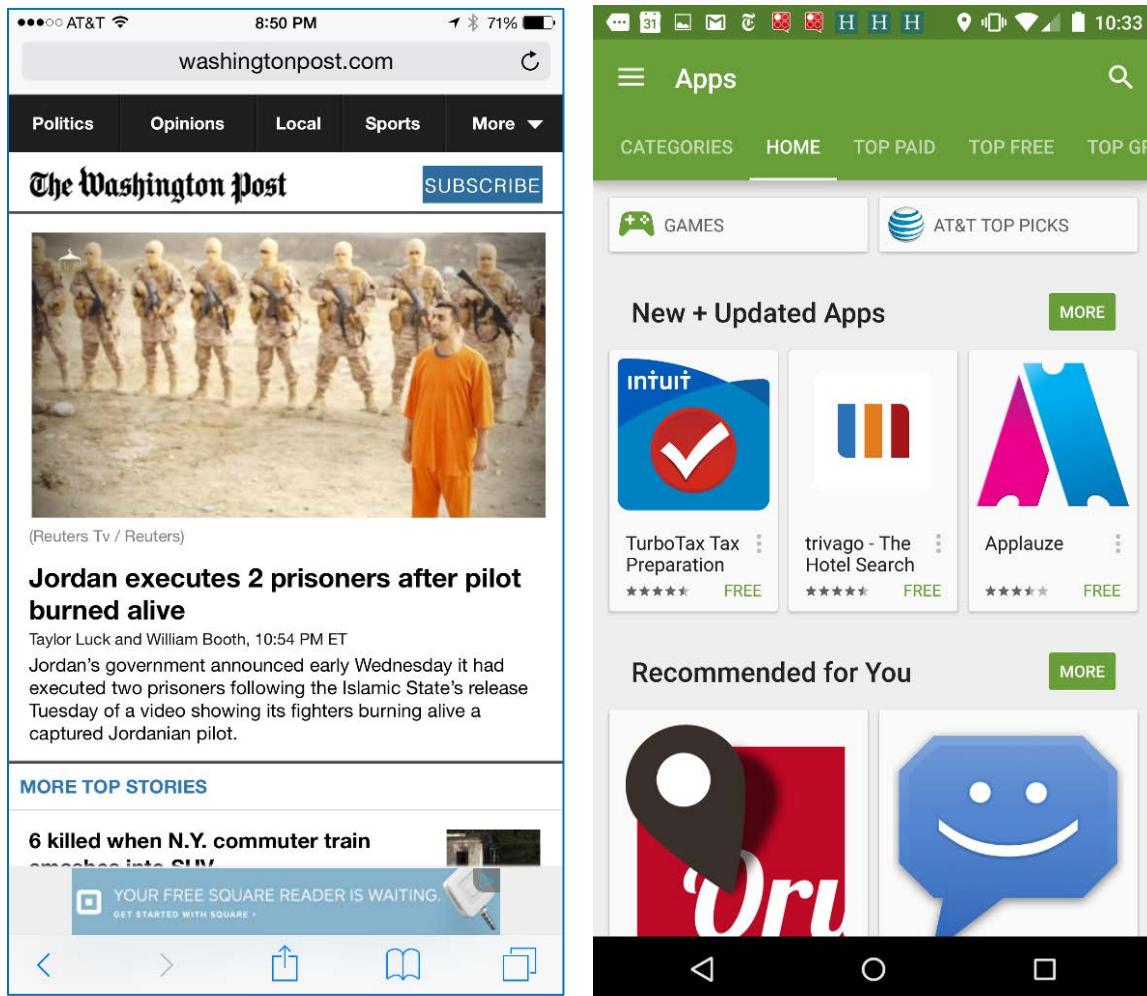
However, navigation⁵⁵ complements search for several reasons: (1) sometimes users don't know what to search for, and need help figuring out the partitioning of the search space; (2) coming up with a good query requires more mental effort and higher interaction cost than tapping a navigation link (and, in fact, users are notoriously bad at formulating good queries); (3) site search often works a lot more poorly than the search-engines users expect it to.

However, both navigation and search come at a price: they occupy screen space and grab users' attention. If the screen space is really scarce, a search box or navigation

⁵⁵ See Raluca Budiu." Search Is Not Enough: Synergy Between Navigation and Search." <http://www.nngroup.com/articles/search-not-enough/>

links at the top of the page can interfere with the users' ability to get to new information fast and may make the user work more. Pay attention to navigation and search, make them accessible and discoverable, but don't forget one of the basic tenets of mobile usability: prioritize content over chrome (see section *Small Screen*). This is in fact one of the big challenges of implementing navigation on mobile: **how to prioritize content while making navigation accessible and discoverable.** Different approaches sacrifice either content prioritization or the accessibility of the navigation.

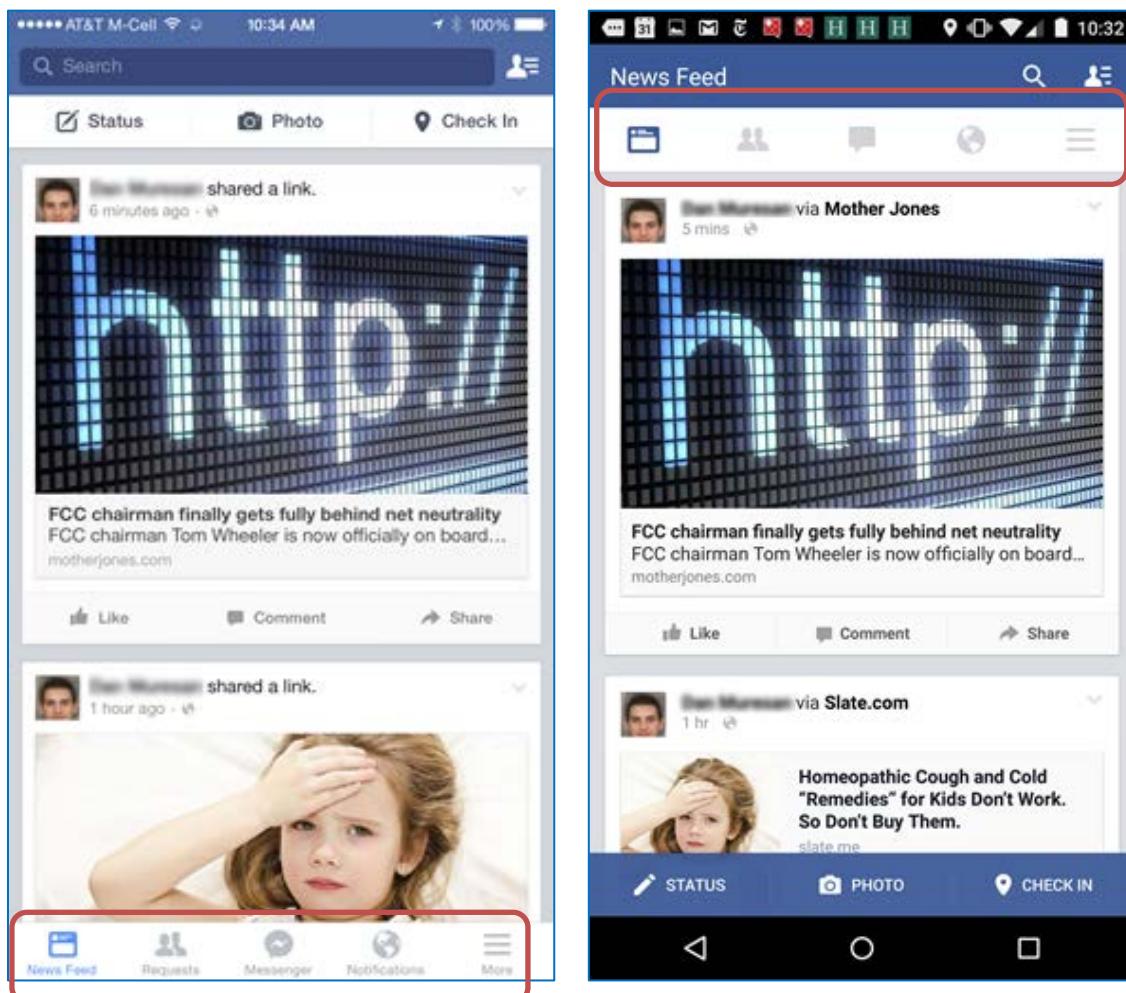
The top navigation bar is essentially inherited from desktop design. It simply involves a bar that enumerates the main navigational options. It is quite efficient, but has two disadvantages: (1) it works well only when there are relatively few navigation options; (2) it takes up valuable real estate at the top of the screen.



The Washington Post's website (left) and Google Play for Android (right) both used a top navigation bar for the main navigation. Google Play was able to fit more items in the navigation bar by using a carousel.

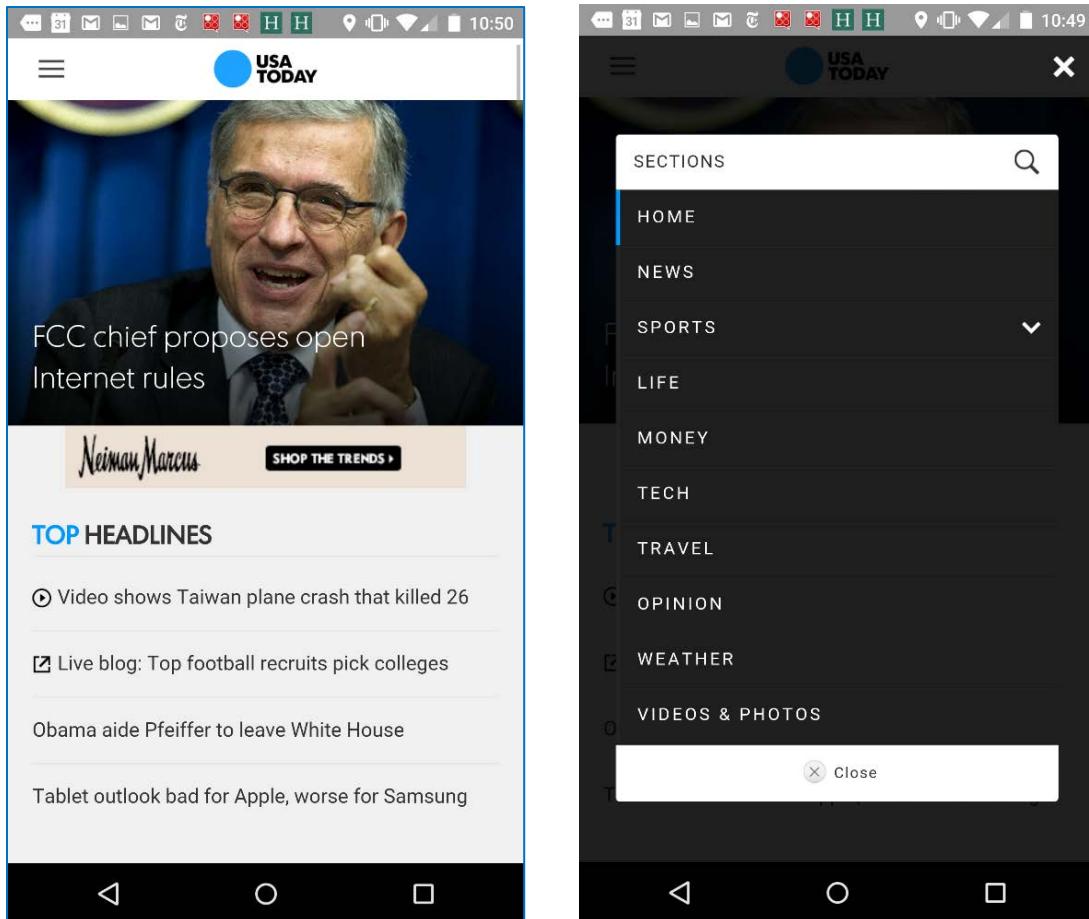
The tab bar is a close relative of the top navigation bar specific to apps. It can appear at the top (Android mostly) or at the bottom of the page (iOS mostly). It is usually present on most pages within an app and has the same disadvantages as the

navigation bar. One important difference between tab bars and navigation bars is that tab bars are **persistent**, that is, they are always visible on the screen, whether the user scrolls down the page or not. Navigation bars are usually present at the top of the page and disappear once the user has scrolled one or more screens down.



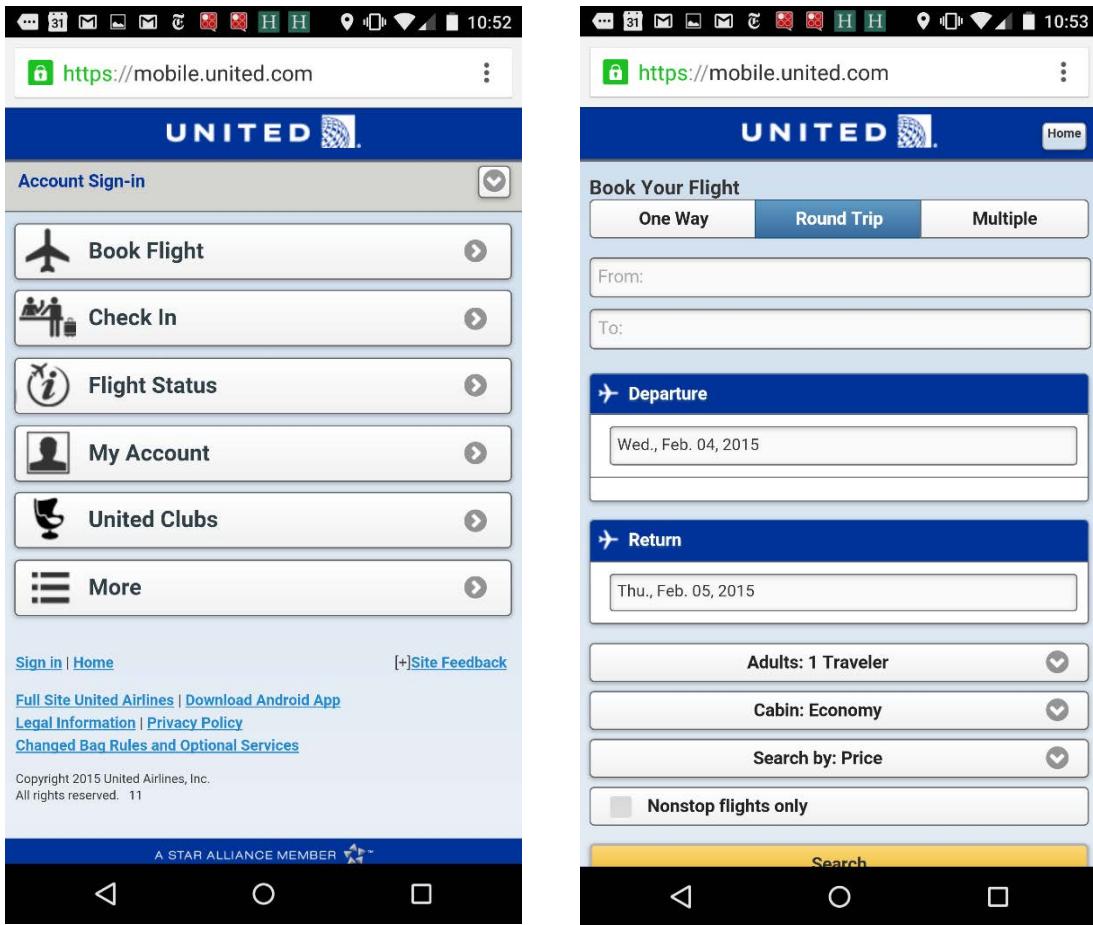
Facebook on iPhone (left) and Android (right) used a tab bar for the main navigation options. The tab was positioned in accordance with official operating-systems guidelines: at the bottom on iPhone and at the top of the page on Android.

The **navigation menu** is a menu that contains the main navigation options in a manner that usually hides the detailed options but makes them visible upon request. While the hamburger menu is perhaps the most popular navigation menu, other types of menus (e.g., pull-up menu) can be used for navigation. The main advantage of the navigation menu is that it can contain a fairly large number of navigation options in a tiny space and can also easily support submenus, if needed; the disadvantage is that it is less discoverable, since, as the old adage says, “out of sight is out of mind.”



A hamburger menu was used for the global navigation options on USA Today's website.

The navigation hub is a page (usually the homepage) devoted to listing all the navigation options. To use navigation, users have to first go back to the hub and then use one of the options listed there. This navigation approach usually devotes the homepage exclusively to navigation (at the expense of content), and incurs an extra step (back to the hub) for each use of the navigation. It can work well in task-based websites and apps, especially when users tend to limit themselves to using only one branch of the navigation hierarchy during a single session.



United used the homepage as a navigation hub. On deep pages, they had to use the Home button to go back to the homepage if they wanted to select a different navigation option.

Some sites and apps implement a hybrid approach, combining several of these basic patterns. For instance, some (e.g., Facebook and Google Play above) may use a combination of hamburger menu and tab or navigation bar. Others may use a hamburger menu that is only present on the homepage (thus combining the navigation menu with the homepage as navigation hub pattern).

The following guidelines reflect the discussion above and summarize the advantages and disadvantages of each of these basic patterns.

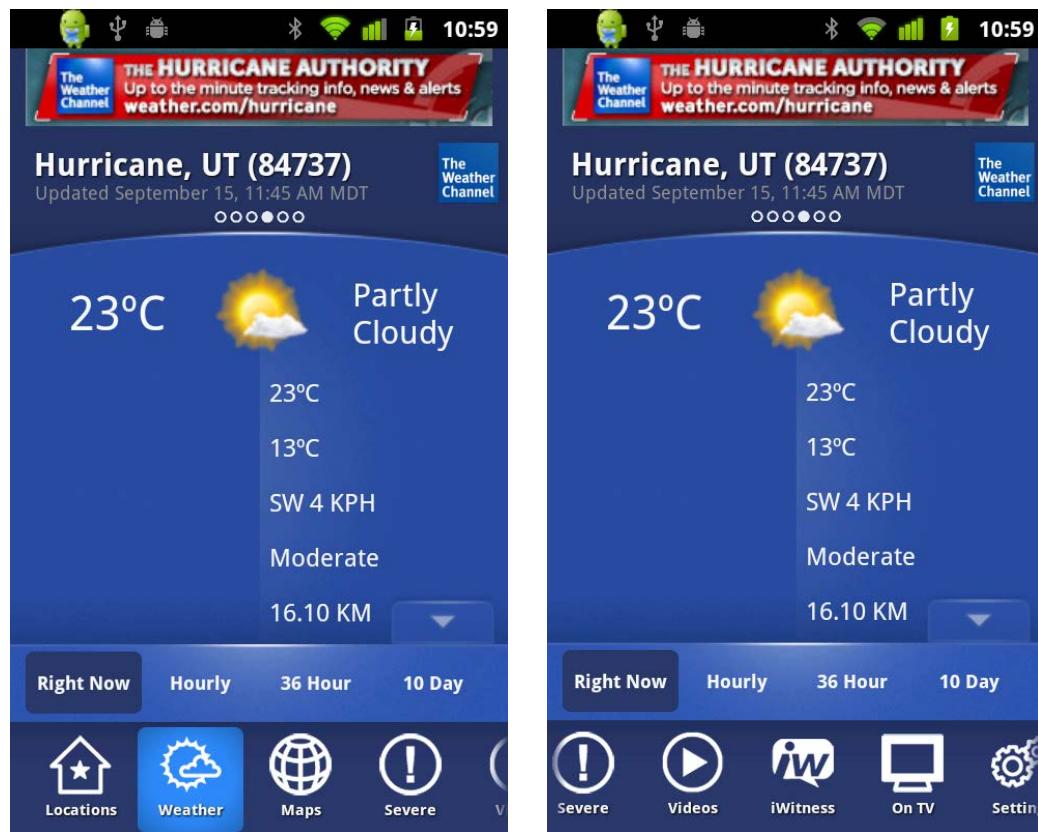
197. Tab bars and navigation bars are well suited for sites with relatively few navigation options.

198. Do not use more than 5 options in a tab bar or top navigation bar.

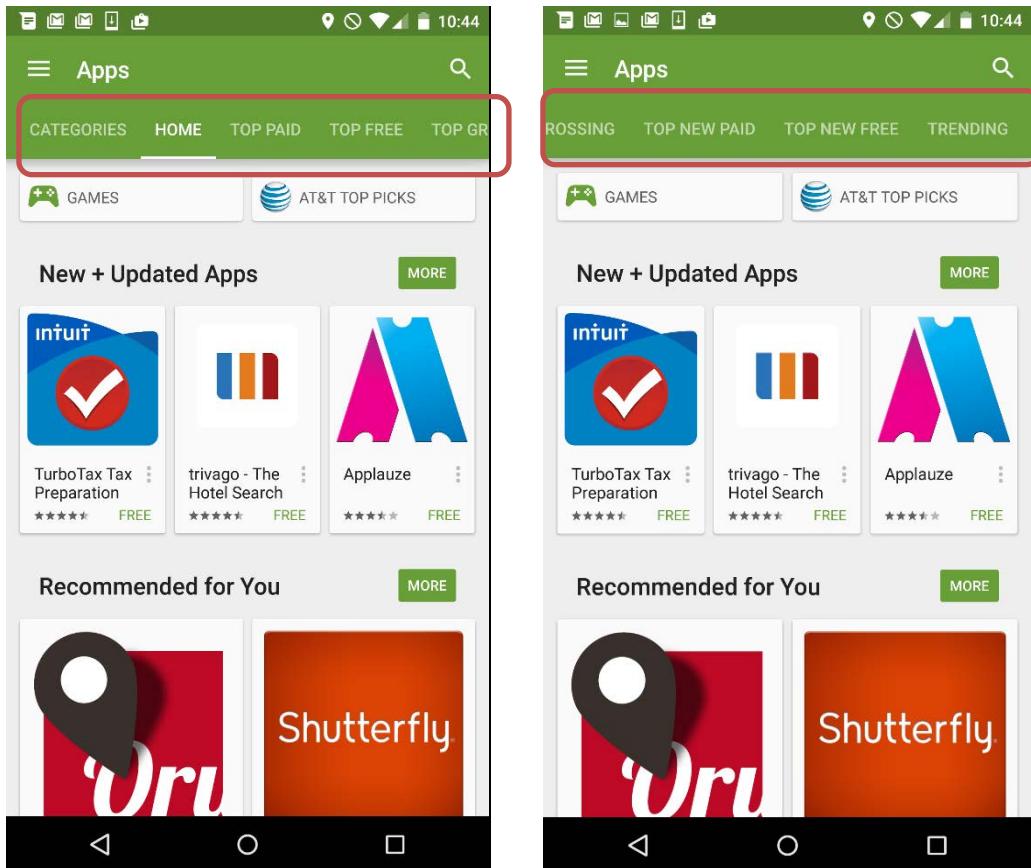
If your site has more than 5 options, it's hard to fit them in a tab or navigation bar and still follow the guideline 69 for the optimum touch-target size. (There are solutions such as using a carousel navigation bar or tab bar, like Google Play did in the example at the beginning of this section, but they are not always appropriate — see next guideline 199.)

199. Use carousels for a navigation bar or a tab bar only if the categories in the bar are similar and highly predictable.

If you have more than 4–5 options in a navigation bar, it may seem like a good idea to use a carousel to fit more categories in a small space. However, remember that out of sight is out of mind. If the categories are widely different (like the case of an older version of Weather Channel illustrated below), it's likely that users won't think to scroll to get to those options. In effect, it'll be the same as hiding some of the navigation under a *More* tab or under an action-overflow button (see our section *Hidden Navigation: More and Action Overflow*).



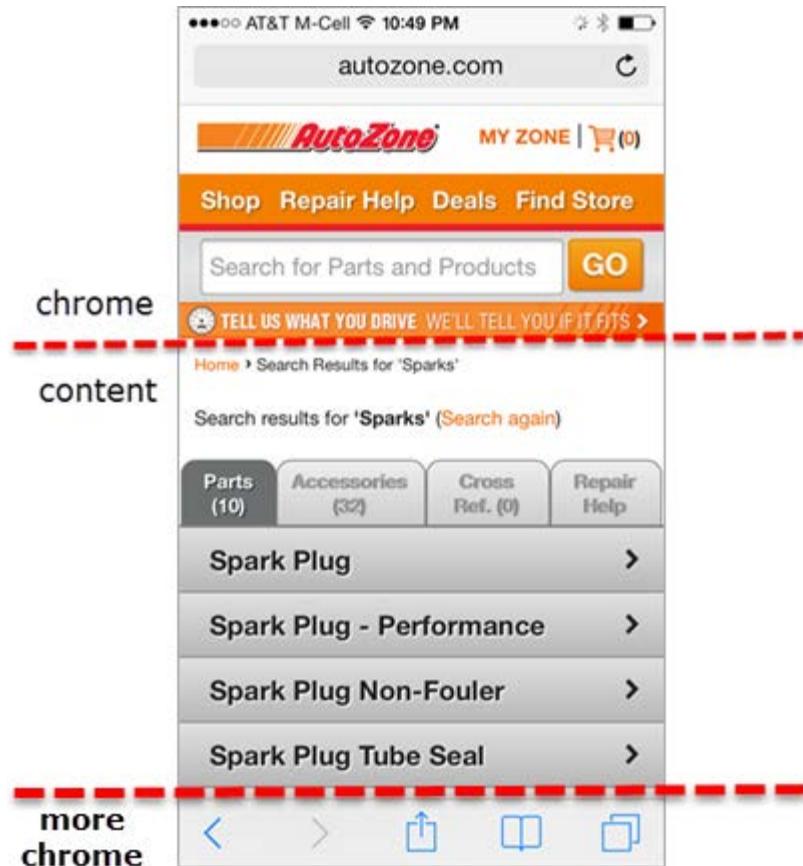
An older version of Weather Channel for Android had a carousel tab bar at the bottom of the screen; the categories in the carousel were not similar or predictable, and thus the ones that were not visible had little discoverability.



The categories in the carousel navigation bar in Google Play for Android were a lot more similar (and thus predictable) than the ones in the carousel tab bar in Weather Channel (above).

- 200. If you decide to use a navigation bar or a tab bar, they should be the main chrome area of the screen and little extra space should be devoted to other utility-navigation options or to search.**

If the site has 4–5 main navigation options, it may make sense to have them all visible on the screen at all times, especially if these are options that will likely be needed. However, keep in mind that navigation needs to be judged in the context of the overall chrome on a page (see the section *Small Screen*): even if a site may only have a few top-tier categories, if other utility-navigation links (e.g., shopping cart, account information) and search must also be included, the overall chrome may add up wasting too much space on the page.



Autozone.com: Although the site only had 4 main-navigation categories (*Shop, Repair Help, Deals, and Find Store*), there were a lot more chrome elements present on the page (logo, shopping cart, *My Zone* link, search box) that all occupied too large an area.

201. Homepage as navigation hub works well for task-based sites or apps where users are unlikely to do more than one task in a single session.

If the homepage is used as a navigation hub, prime real estate will be wasted for chrome and all navigation will have to go through the homepage. While these two may seem as major disadvantages (and they are for most types of sites or apps), they can be less of a problem for those sites and apps used not for browsing and consuming content, but for accomplishing a very specific task (for example, checking in for a flight or changing the settings of the phone). Such sites and apps can take advantage of the homepage-as-navigation-hub pattern, especially if users rarely accomplish more than one task during a single session, and thus they don't need to traverse the navigation tree often (an action that is relatively difficult and annoying if all navigation must go through the homepage).

In the airline example, most likely you want to *either* buy a ticket *or* check in for a flight, but not do both in a single session. Thus, most users won't have to return to the hub in this example, meaning that it rather serves as an efficient distribution point.

202. On homepages of content-heavy, browse-mostly sites or apps, give priority to new content over navigation links.

When users come to a browsing site, they are likely to look around for new content. If your homepage only contains a set of navigation options, users will have to do extra work to get to new content. That is why we recommend that you make the new content highly salient, and, if space allows, also add navigation options at the top or above the fold.

203. The navigation menu makes the navigation options least discoverable and is best suited for content-heavy, browse-mostly sites and apps.

If users rarely care about navigating to specific sections of the site and are mostly content to digest whatever information is presented to them (as is often the case on news sites), then a navigation menu is appropriate. The navigation menu also has the advantage of stealing a minimum amount of space from content, which is the star of browse-mostly sites.

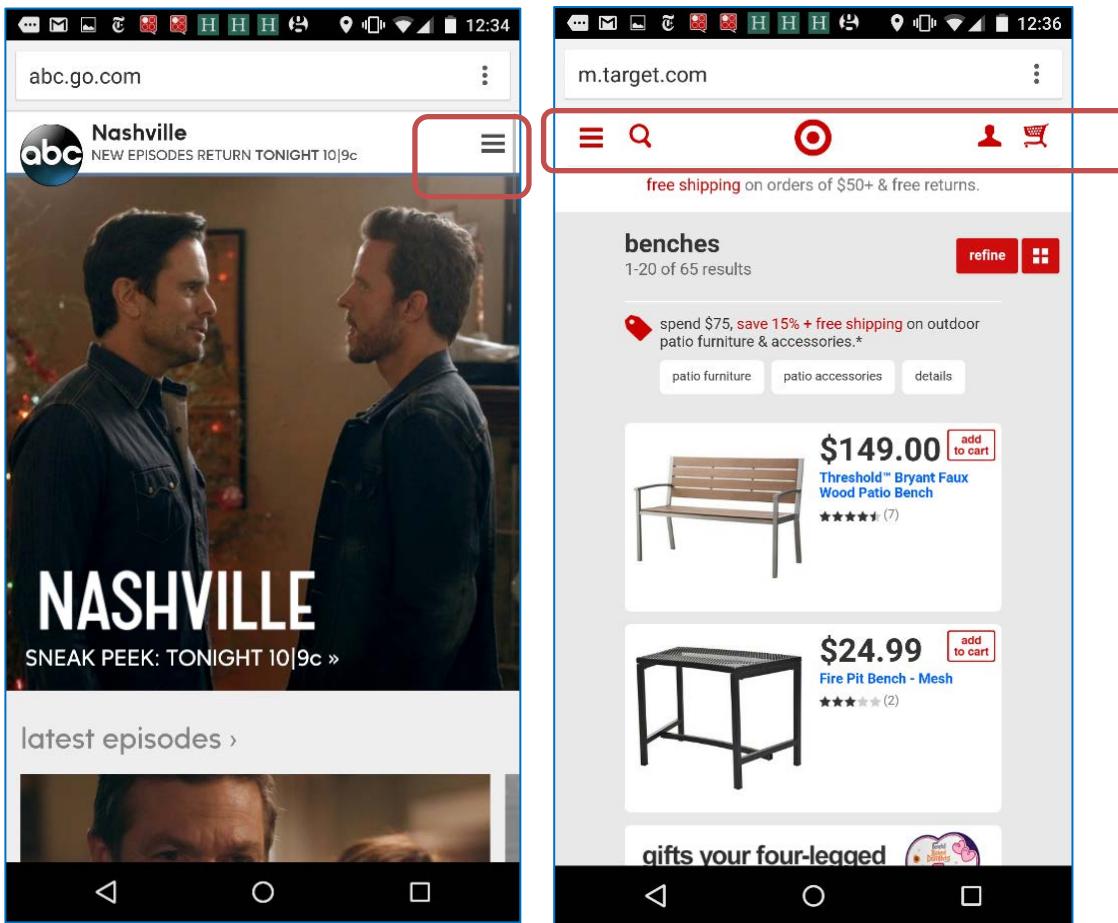
However, keep in mind that when navigation is hidden under a menu, even though that menu as a whole may be salient enough, users will have to make a decision to open it and check whether the individual navigation options are relevant. While the navigation menu is becoming standard, many people still simply don't think to open it. Even users who tried the navigation menu at some point during a session may not remember to do so later on.

Making Navigation Accessible

In order for people to be able to easily jump from one branch of the navigation tree to another, the navigation needs to be readily available, both at the site level (that is on the different pages of the same site or app) or at the page level (that is, as people scroll down a long page). Because of their persistent nature on the page, in apps tabs are usually one of the best ways to make navigation accessible.

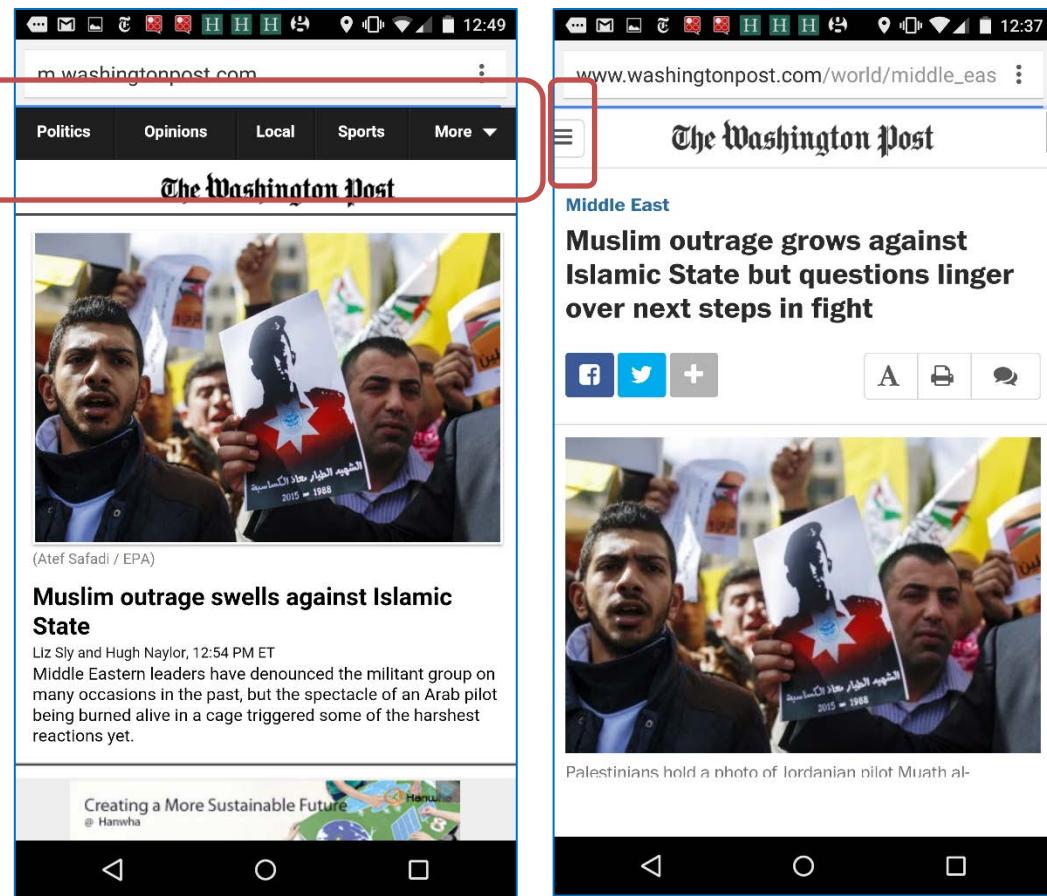
204. Unless your site or app is task based, consider allowing users to access the main navigation on every page.

Users may land on one of your pages via a search engine or they may get there from another page on your website. In both cases it is useful to give them a way to navigate away from that page to another section of your website. However, while this solution works well for navigation menus, having the navigation structure of your webpage repeated on every page can be too space consuming on a mobile screen, especially if you use a navigation bar or a tab bar.



ABC (left) and Target (right) included a main-navigation hamburger menu on every page of their sites. Target also had a navigation bar with navigation menu, search tool, and utility-navigation options on every page.

While Washington Post used a navigation bar on the homepage, it did not include it on the deep pages. However, in order to make navigation accessible throughout the site without forcing users to go back home, they used a hamburger menu for the main site navigation on the deep pages. While this solution works, it is not ideal, since the inconsistent treatment of the navigation throughout the site is going to make the hamburger menu even less likely to be used.

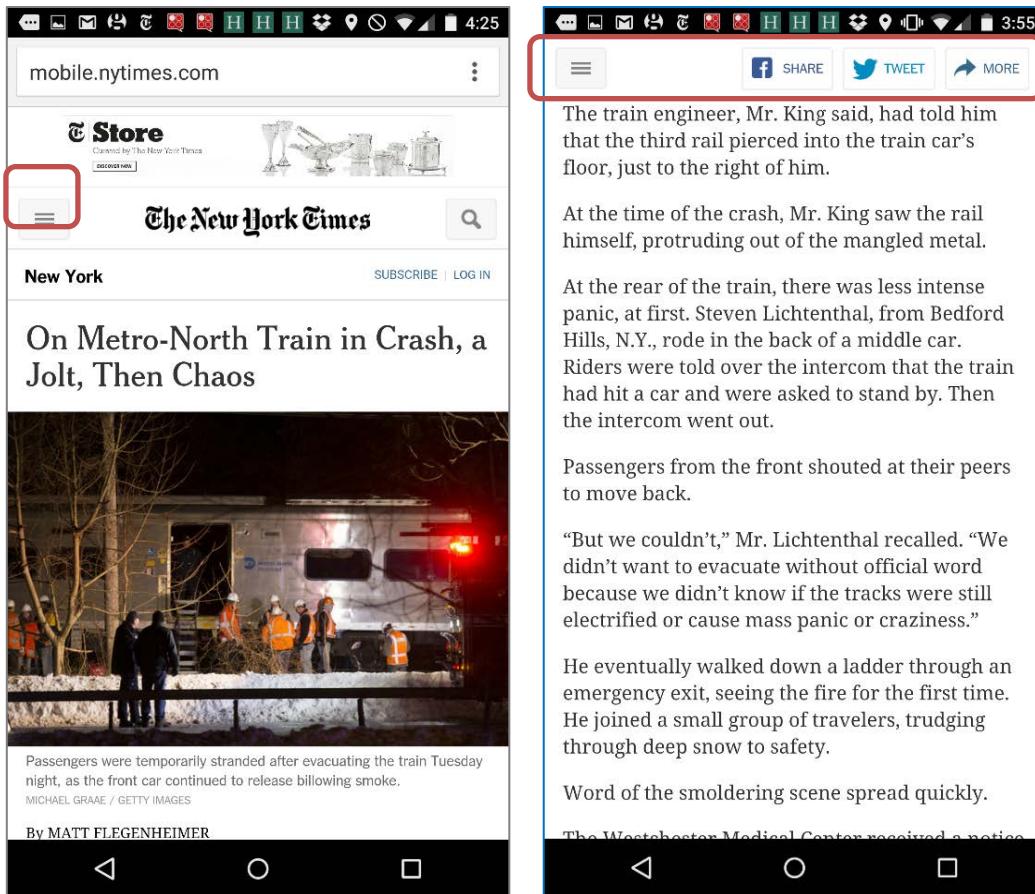


The top navigation bar present on the homepage of The Washington Post (left) became a hamburger menu on article pages (right).

- 205. If your mobile pages are long, consider using persistent or floating navigation at the top of the page or, alternatively, a floating back-to-top control to rapidly take users back to the top of the screen.**

Mobile pages can be quite long, and scrolling back and forth to the top of the page, where navigation or controls may be, can quickly become tedious. To circumvent this problem, many sites and apps have started using techniques for making certain page elements either visible at all times or at least easily accessible at all times.

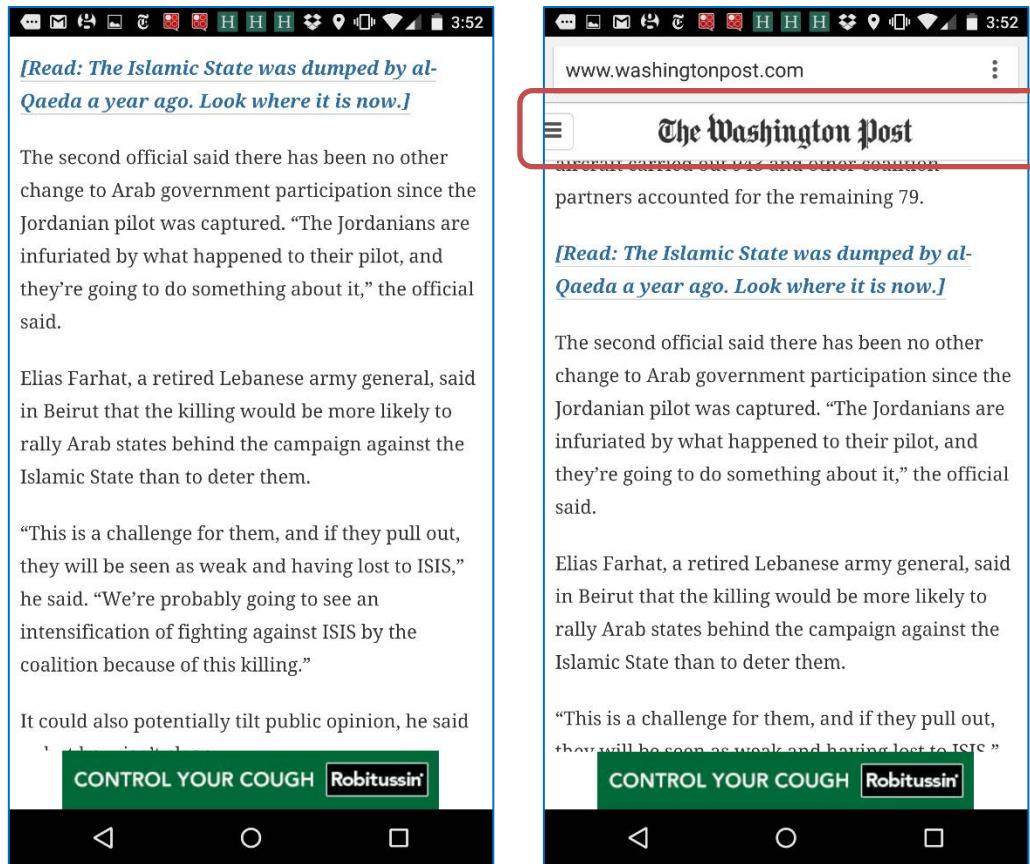
Persistent (or fixed-position) controls are sticky — they stay on the screen as the page scrolls up or down. They have the advantage that they are accessible to users at all times, but the downside is that they continue to take up space and effectively reduce the screen area available for content.



NY Times used fixed positioning at the top of the page for the main-navigation menu and for the social-sharing controls.

As people scrolled down on the NY Times article page, the top bar that included the main-navigation menu and the social-sharing buttons remained fixed.

To get around limiting the screen space for content, some sites and apps use floating navigation: instead of having the navigation visible at all times, they hide it if users move down the page but make it visible again as soon as the user starts scrolling up. The Washington Post example below implemented that idea on its article pages.

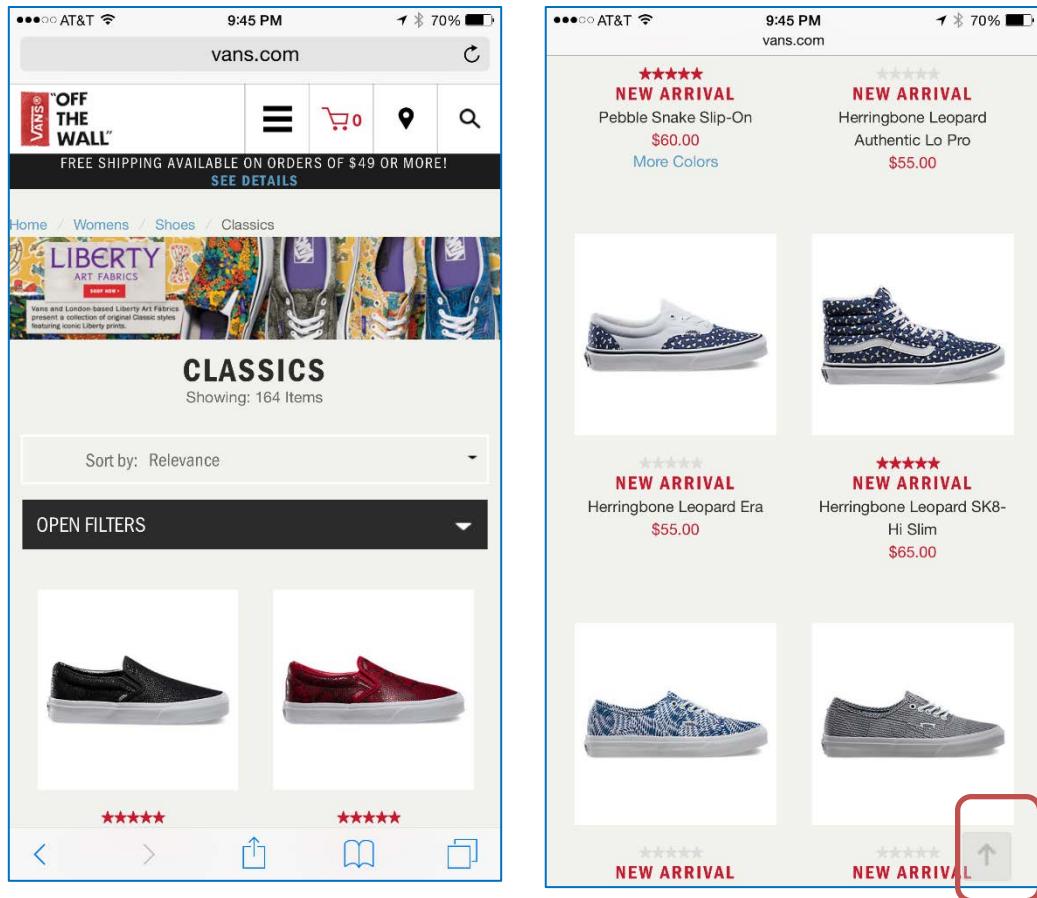


The Washington Post used a floating header for the logo and the navigation menu: as soon as users scrolled up, it displayed the header at the top of the page, in case users wanted to use the navigation options.

There is a slight disadvantage to floating navigation: namely, people may overlook it because it's not always present on the page (again, consistency usually pays off in user experience) and appears as a side effect of an action that does not have that intent (scrolling up).

A different solution that also takes advantages of floating elements uses a floating back-to-top control that appears in the bottom right corner of the page as soon as the user starts scrolling up (or sometimes down)⁵⁶. This is somewhat more noticeable, as it's usually closer from the users' focus of attention, and has the advantage that the user causes directly the scroll up by pressing that button. But not all users know what it is or use it, and, of course, it involves an extra step from the user. And it can obscure content.

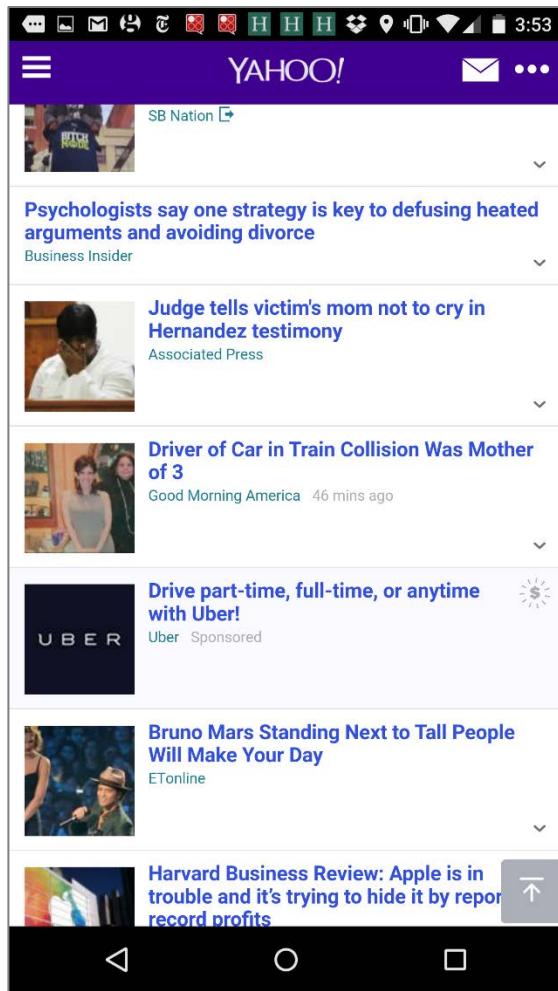
⁵⁶ Sometimes such a button can be persistent and visible at all times instead of only visible when the user starts scrolling up.



Vans.com: The “back-to-top” arrow appeared as soon as the user started scrolling down.

206. Avoid using both persistent navigation and a back-to-top control.

Although in theory the two are not identical, they do serve similar purposes most often. Unless you have strong reasons to believe that users may want to revisit content (other than navigation) that was placed close to the top of the page, it's unnecessary to have both these features on a page.

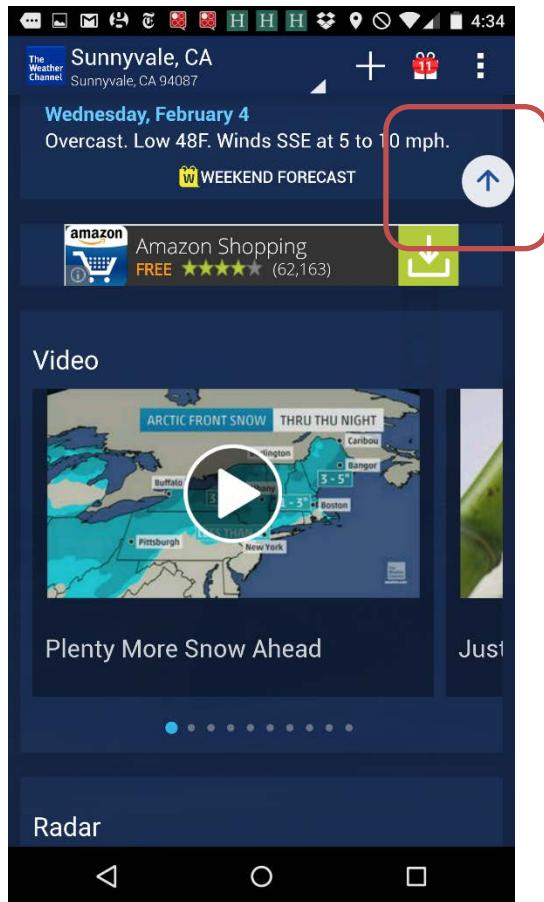


Yahoo! had both a persistent header and a back-to-top floating control on its homepage. In this situation these two features were likely redundant.

207. Place the “scroll-to-top” control close to the bottom right corner of the screen.

It's become almost a standard to place this control in the bottom right corner of the screen. This standard is supported by natural user behavior: as people have reached the fold and decided to scroll up, their attention is still focused in the area closer to the bottom edge of the screen. If a button appears elsewhere on the screen, it's likely that it won't get noticed.

Weather Channel placed the scroll-to-top button close to the right edge of the screen, but more towards the top of the page. This placement went unnoticed in our testing: people did not take advantage of the button.



Weather Channel for Android: The nonstandard placement of the back-to-top button made it less noticeable.

Navigation and IA on Mobile

Information architecture (IA) refers to the information backbone of a site or app: the different categories in which the site or app content can be grouped. **Navigation** refers to those UI elements that allow the users to access the different categories⁵⁷. In many cases, the navigation directly reflects the information architecture. On mobile, a significant design challenge is to translate a deep information architecture into a usable mobile navigation. We've seen many sites and apps struggle with this issue as they were trying to translate a multilevel desktop navigation into a decent mobile-navigation scheme.

Before discussing specific guidelines for multilevel navigation on mobile, we must again emphasize the idea of content over chrome. Navigation is chrome, and people don't go to a site or app to see the chrome; they go there because they are interested in the content.

Deep IA hierarchies that are rendered into multiple navigation steps pose

⁵⁷ See Jen Cardello. "The Difference Between Information Architecture (IA) and Navigation." <http://www.nngroup.com/articles/ia-vs-navigation/>

considerable challenges on mobile, because each of these steps takes time and user effort and are perceived as a detour from the point of interest — the content.

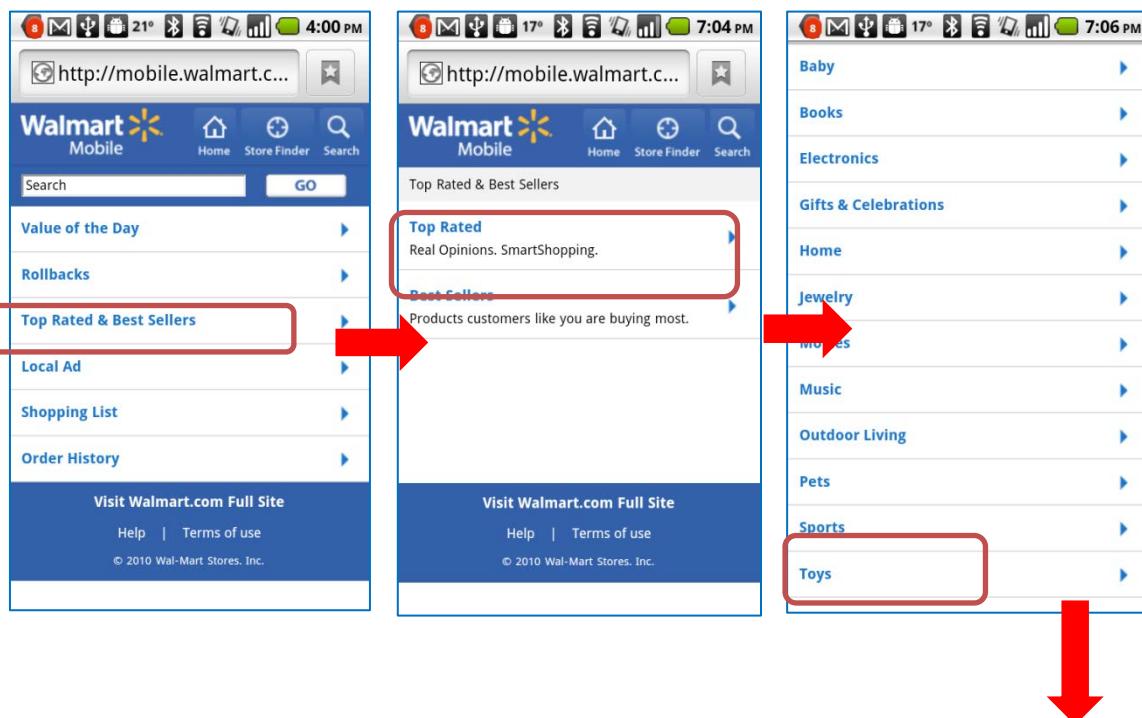
Whereas on the desktop deep navigation hierarchies are usually implemented through a succession of cascading menus, on mobile they typically translate in many successive pages where people have to select among an excess of subcategories. Unfortunately, a deep hierarchy often depletes users' patience: they have to click again and again (often, waiting for the pages to load between clicks) before they get to a real product or item of interest.

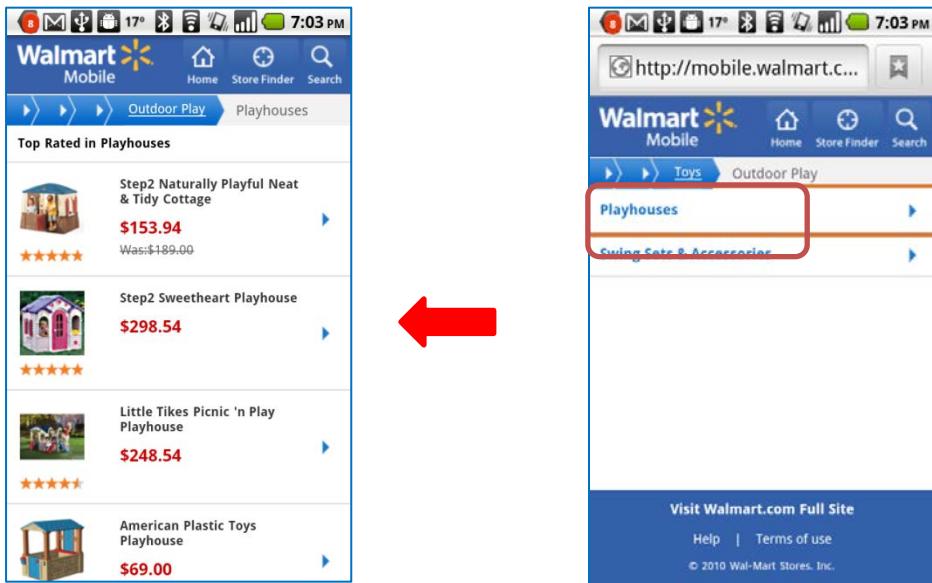
208. Avoid navigation hierarchies deeper than 2 levels on mobile.

209. If your site IA has more than 2 levels, consider flattening the navigation hierarchy on mobile by merging some of the IA levels.

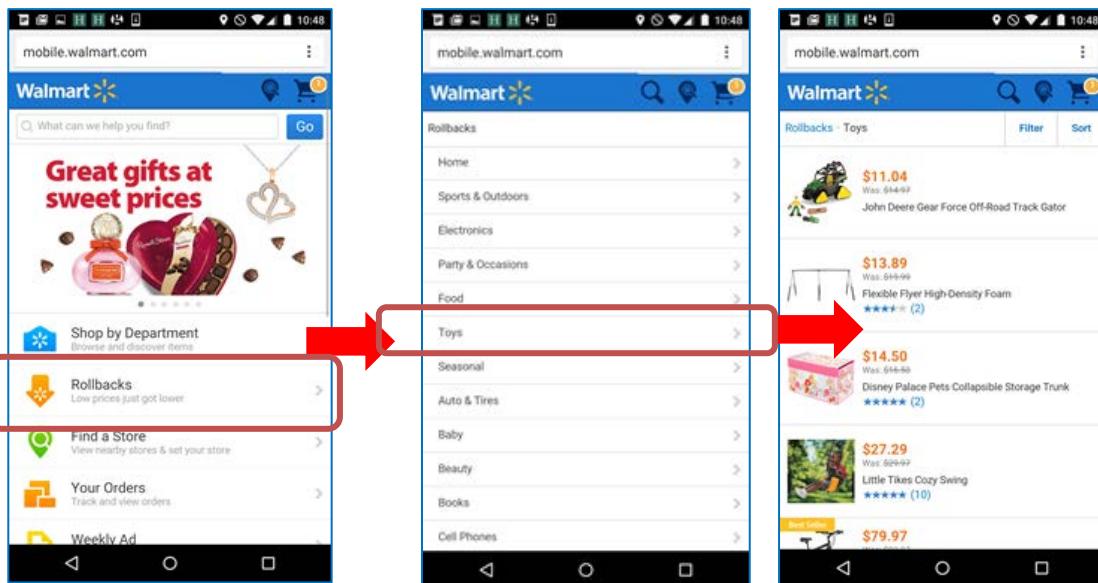
Although the full-site hierarchy may be (appropriately) deep, we recommend flattening it for mobile to avoid too many clicks and wait times.

An older version of Walmart's mobile site used to suffer from exactly this problem: users looking for, say, toys had to click approximately 5 times before they could see a picture of a product: they had to select "Top rated and best sellers", "Top rated", "Toys", "Outdoor toys", "Playhouses". Luckily, Walmart realized that this hierarchy was too deep and compressed it to only two steps in a later release. We show here both the older variant of Walmart and the newer redesign.



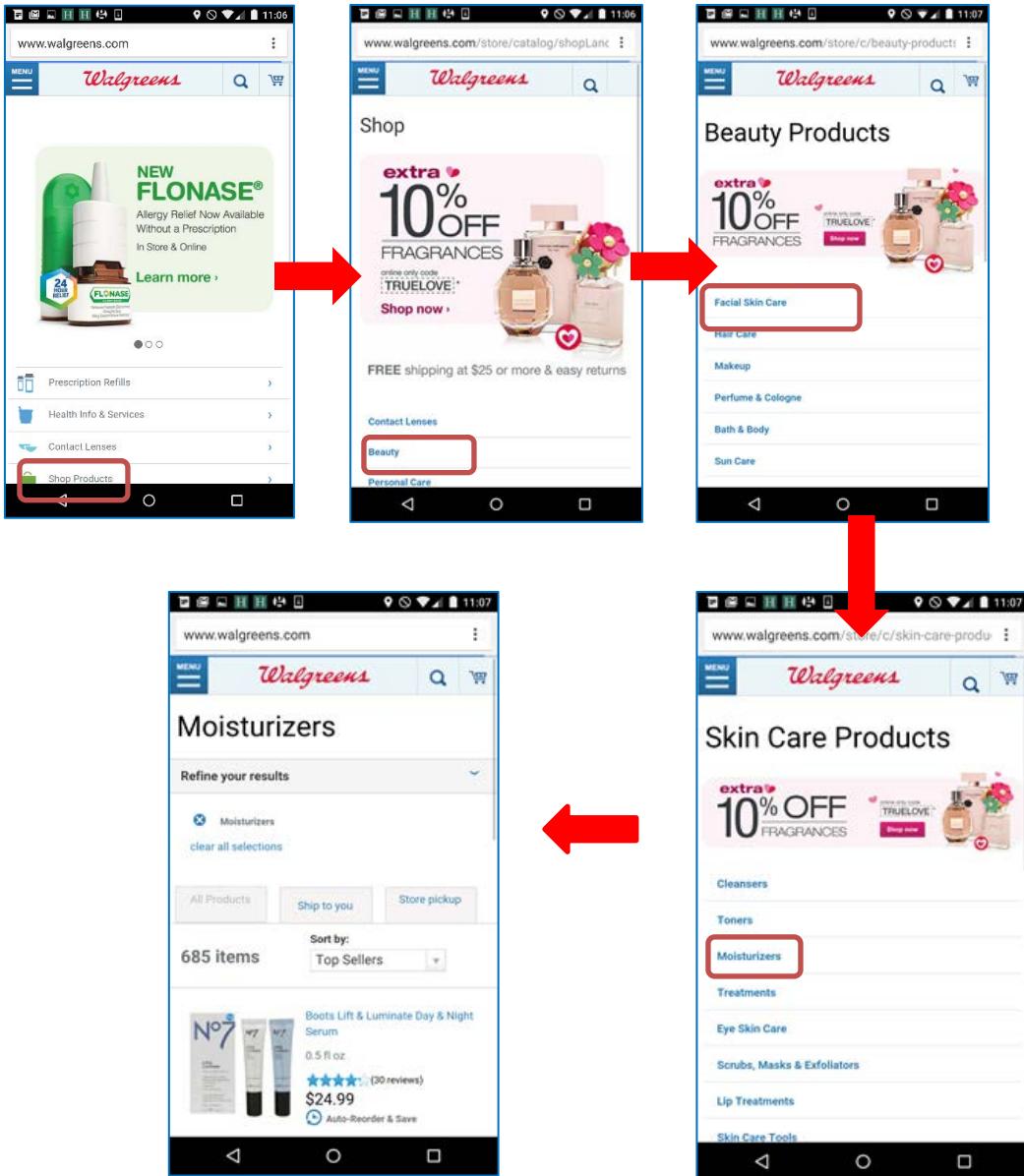


An older version of Walmart.com used a deep product hierarchy: users had to click many times before seeing a product.



A redesign of Walmart dropped many of the subcategories under "Toys", getting users faster to products.

Unfortunately, there are still many e-commerce sites that suffer from an inflation of mobile navigation (Target and Walgreens are two examples). We don't recommend getting rid of categories in the information architecture: they can still be useful when people try to narrow down the available selections. However, with clicks so expensive, it's important to follow the principle of immediate gratification on mobile: show content (rather than site navigation) as soon as possible.



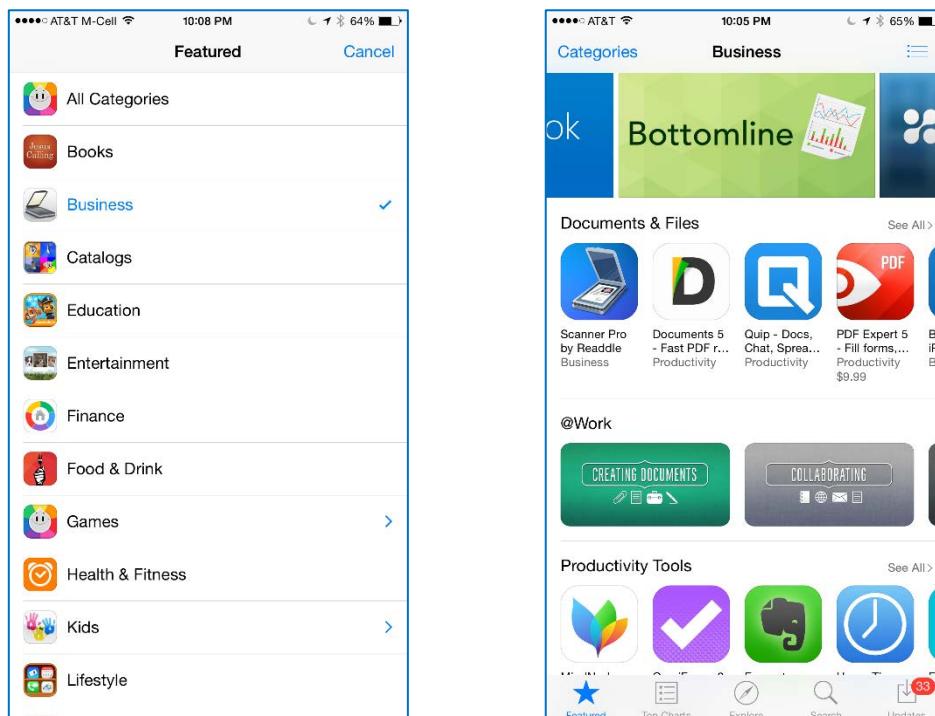
Walgreens.com required users to make 4 category selections before they could see any products.

- 210. If some IA levels have been flattened in the mobile navigation hierarchy, provide filters, facets, and in-page navigation to allow users to recover them should they need to.**

We've recommended that you avoid implementing navigation hierarchies that are deeper than 2 levels on mobile. But what should you do if your users actually want to find items that are more specific than the 2 levels allow you to

specify? The answer is to provide them with filters and facets⁵⁸ (see section *Filtering and Sorting*) that enable people to narrow down the provided items to a subset that is relevant for them. In other words, let the content, and not the category structure, drive the browsing.

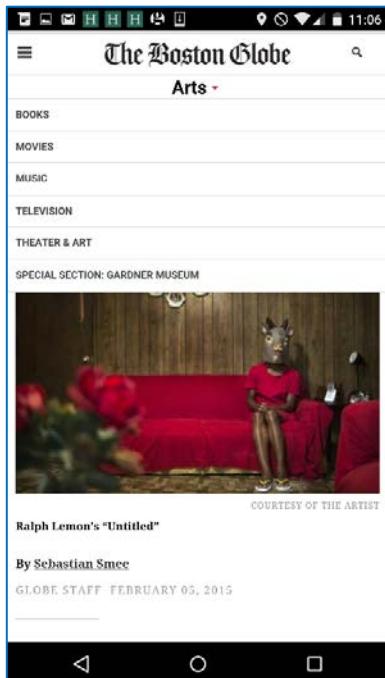
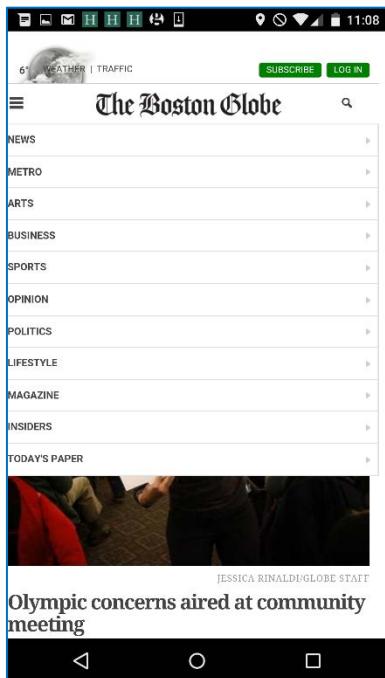
Apple's App Store followed this guideline when it listed samples from all the different types of content for the category *Business*, rather than presenting users with a list formed of the subcategories (*Documents & Files*, *@Work*, *Productivity Tools*, etc.) for that item.



App Store for iPhone: the subcategories under *Business* were indicated by displaying a few items under each. Instead of using a category-based design where users had to decide among different categories, the app rightly used a content-based design.

Boston Globe also flattened its navigation hierarchy on mobile (In fact, it restricted itself to a single-level structure.) When users selected the *Arts* section, they could see the content-heavy category homepage, containing a variety of arts-related content. If they wanted to narrow down this content to a particular subtype (e.g., movies), they could have used the second-level menu available on this page. (Unfortunately that menu was very hard to discover: the idea was good, but the execution suboptimal.)

⁵⁸ See Kathryn Whitenton. "Filters vs. Facets: Definitions." <http://www.nngroup.com/articles/filters-vs-facets/>

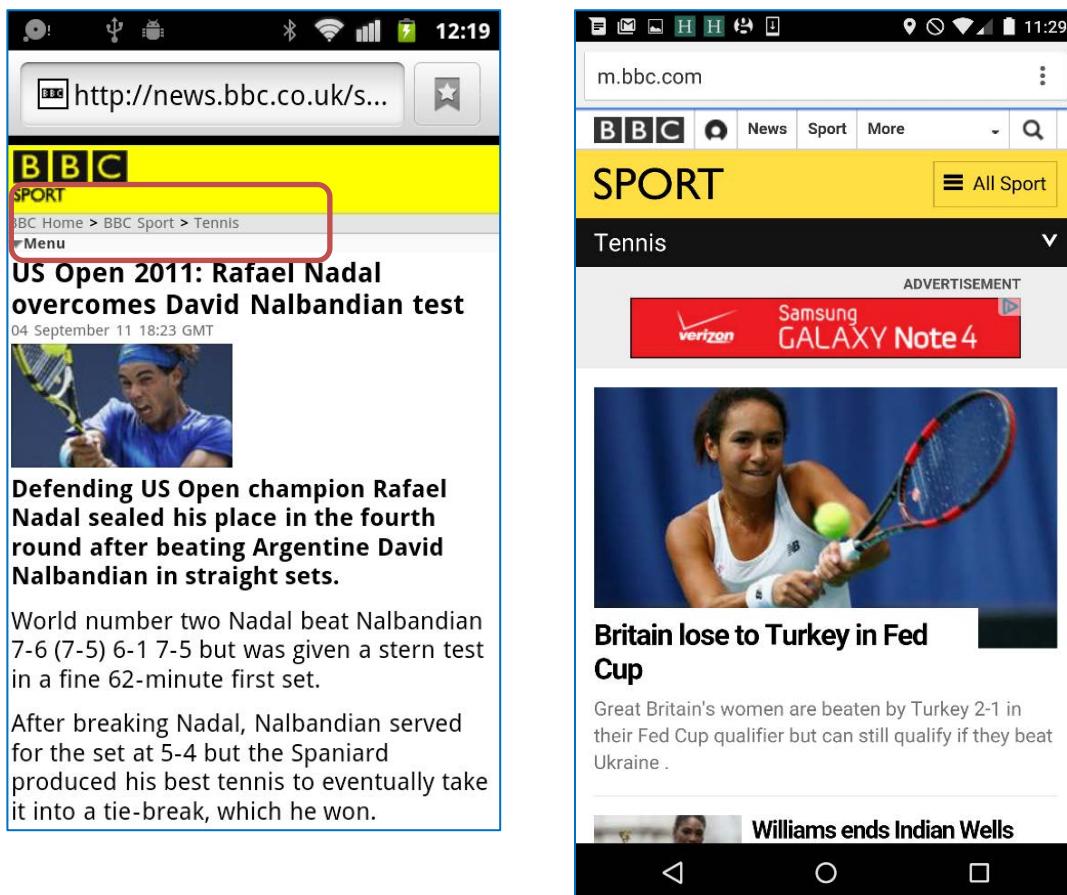


The main navigation menu on Boston Globe contained no submenus. A category in this menu (e.g., *Arts*) led to the category page, which contained an amalgam of articles on different category-related topics. Users could narrow down the content to a subtopic of the category by using the section menu. (Unfortunately, this menu was virtually undiscoverable in our tests due to the very subtle visual styling.)

211. Use breadcrumbs on sites with a deep navigation structure (more than 2 levels). Do not use breadcrumbs on sites with shallow navigation structures.

While breadcrumbs can keep users oriented in a deep navigation hierarchy, they have two disadvantages on mobile: (1) they take space; (2) they are often really small and, as a result (a) they do not form good touch targets, and (b) they often don't get noticed. Sites with a deep navigation structures (many navigation levels, with many options at each level) can benefit from breadcrumbs, but, as argued earlier in this section (guidelines 208 and 209), deep hierarchies do not belong on mobile, so breadcrumbs should be unnecessary for most sites or apps.

For example, a site like BBC should not need breadcrumbs. An older version of BBC did have them, although they were unnecessary: *BBC Home* and *BBC Sport* breadcrumbs duplicated logos already existent on the page. There were too few navigation layers on the website to justify breadcrumbs. BBC realized this and later versions eliminated the breadcrumbs.



An older version of BBC (left) did not have an IA deep enough to warrant the usage of breadcrumbs, plus the breadcrumbs duplicated information already existing on the page. A more recent version of the site (right) eliminated breadcrumbs.

Although we don't recommend using deep hierarchies, sites that do use them — like Walgreens, discussed in the previous example — need to also make sure that they show the current path. Note that Walgreens did not have breadcrumbs, but Walmart did (see the examples discussed under guideline 209).

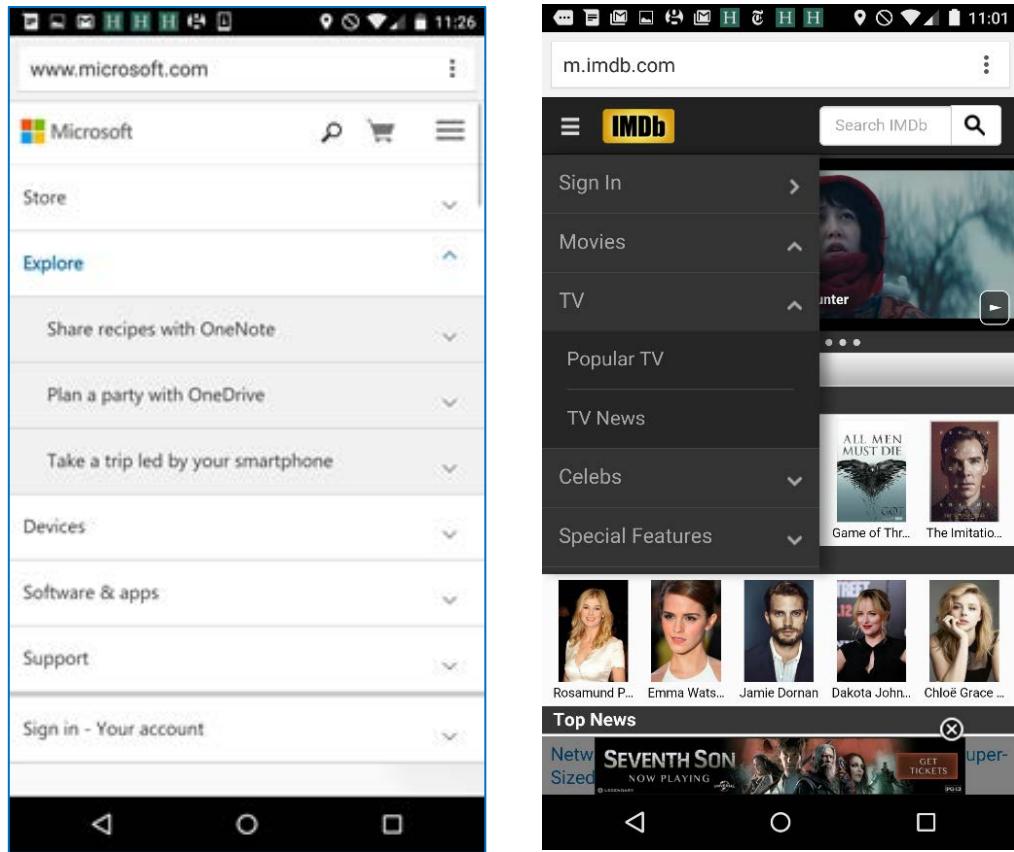
Submenus and Second-Level Navigation

We've seen before that we don't recommend having more than 2-level navigation hierarchies on mobile. Given that cascading menus that overlap (like the ones on the desktop) are hardly an option on mobile, how should sites and apps display second-tier categories?

There are three options available: (1) use a submenu (if the main navigation is implemented using a menu); (2) use a category landing page that contains the second-level categories either as a list or as an in-page menu (similar to the solution offered by the Boston Globe example, in guideline 210); (3) alter the main menu to display the second-level categories instead of the first.

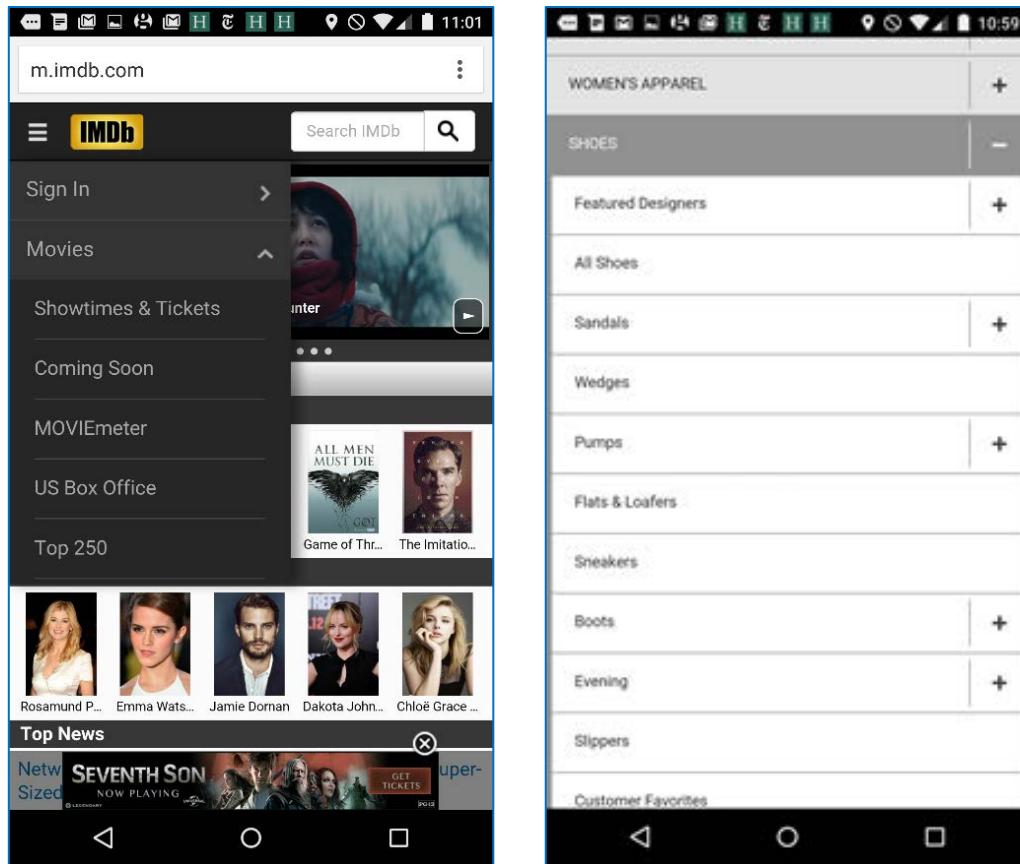
- 212. If your main navigation is implemented as a menu, you can use an in-situ accordion for a short (under 10 items) second-level submenu.**
- 213. If your second-level submenus are long (more than 10 items), consider using a category landing page to display them.**

Accordions have the advantage that they don't require a page load, and, thus, users can quickly explore second-level categories and investigate the type of content available on the site or in the app. When the submenus are long, however, the overall menu length becomes harder to manage and it makes it more difficult to get a quick view of the structure of the site.



Short submenus on Microsoft.com (left) and IMDb (right) were appropriately implemented with an accordion. IMDb, however, should have made the navigation menu longer (especially since some of the submenus included more than two choices).

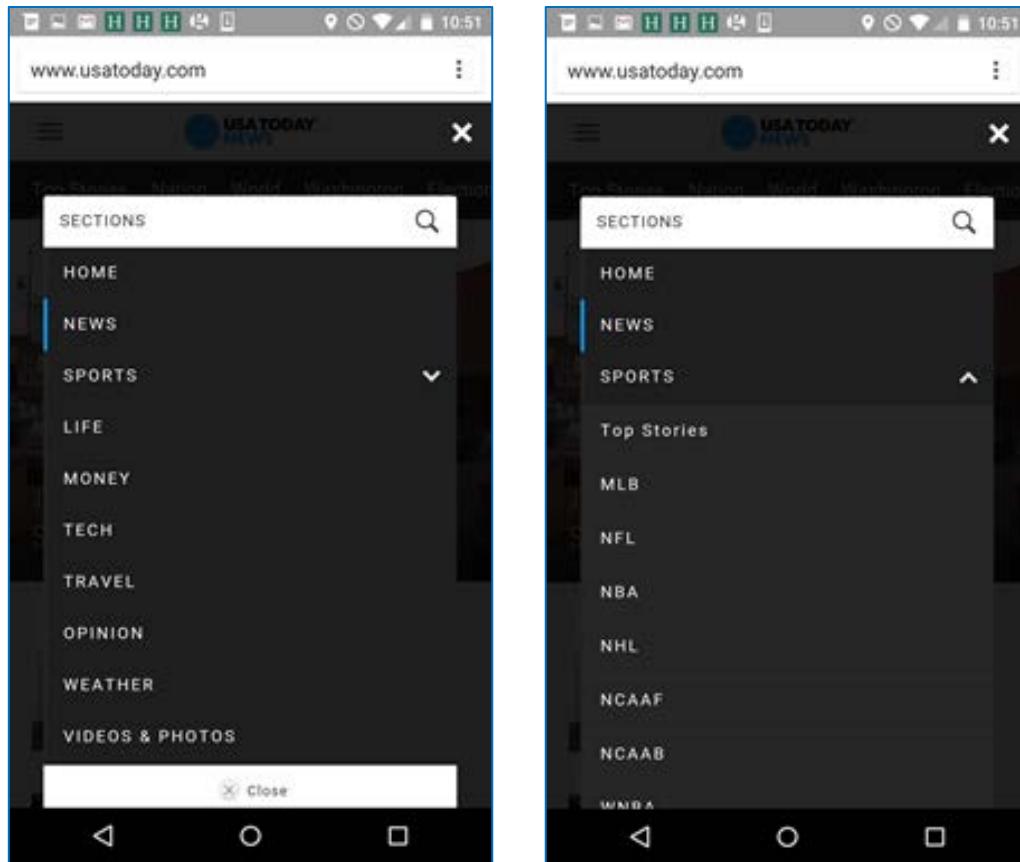
Scrolling in a small area of the screen can be tedious. It can also prevent people from keeping track of which top category the submenu belongs to.



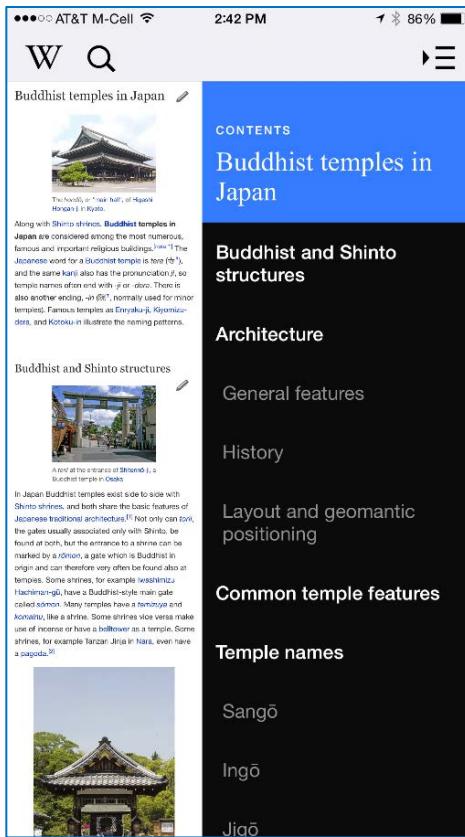
A different submenu on IMDb (left) was too long for the menu window and made it harder for people to keep track of where they were. Neiman Marcus's (right) submenus also had too many options.

214. If you use an accordion to implement a submenu, make sure that the submenu categories are styled differently than the top-level categories.

It's important that the navigation hierarchy translates well visually and people understand the level of each of the items displayed under a navigation menu. If the styling of first- and second-level categories is similar, users may not know which is which and get disoriented.

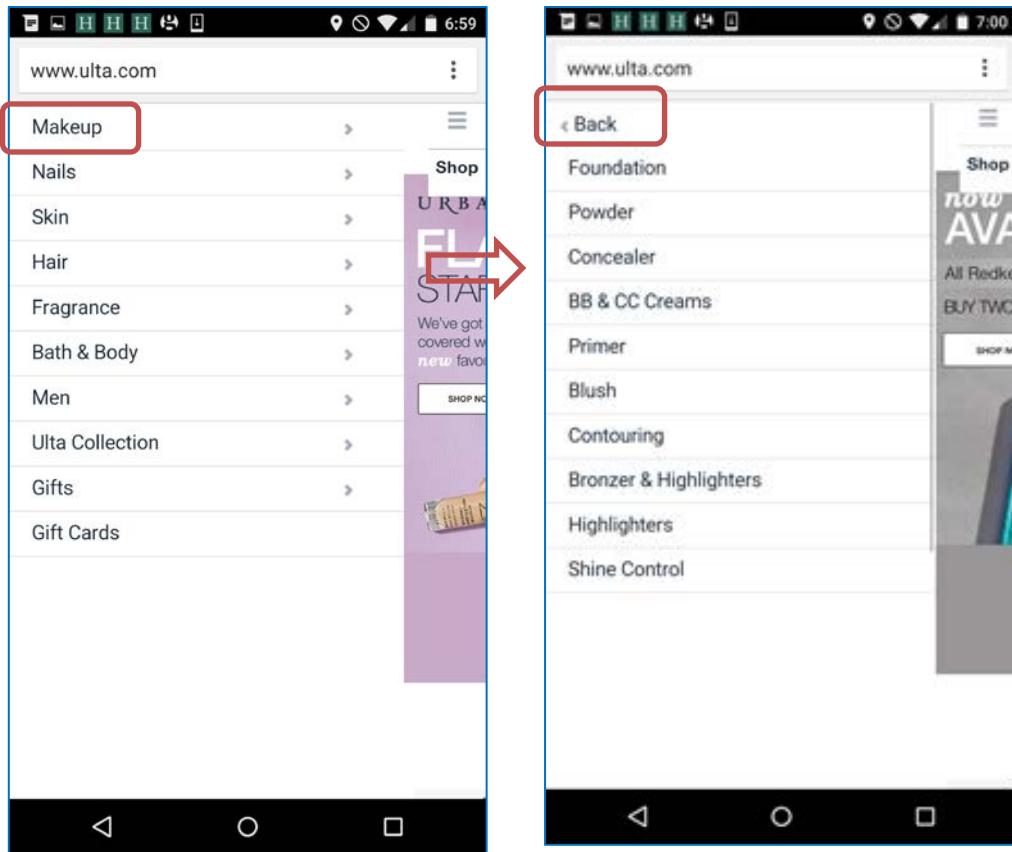


USA Today: the left screenshot shows the main-navigation menu with the *Sports* submenu closed, and the right screenshot shows the *Sports* submenu expanded. The main categories and the second-level categories were styled in a too similar way and made it hard for users to keep track of the navigation hierarchy.

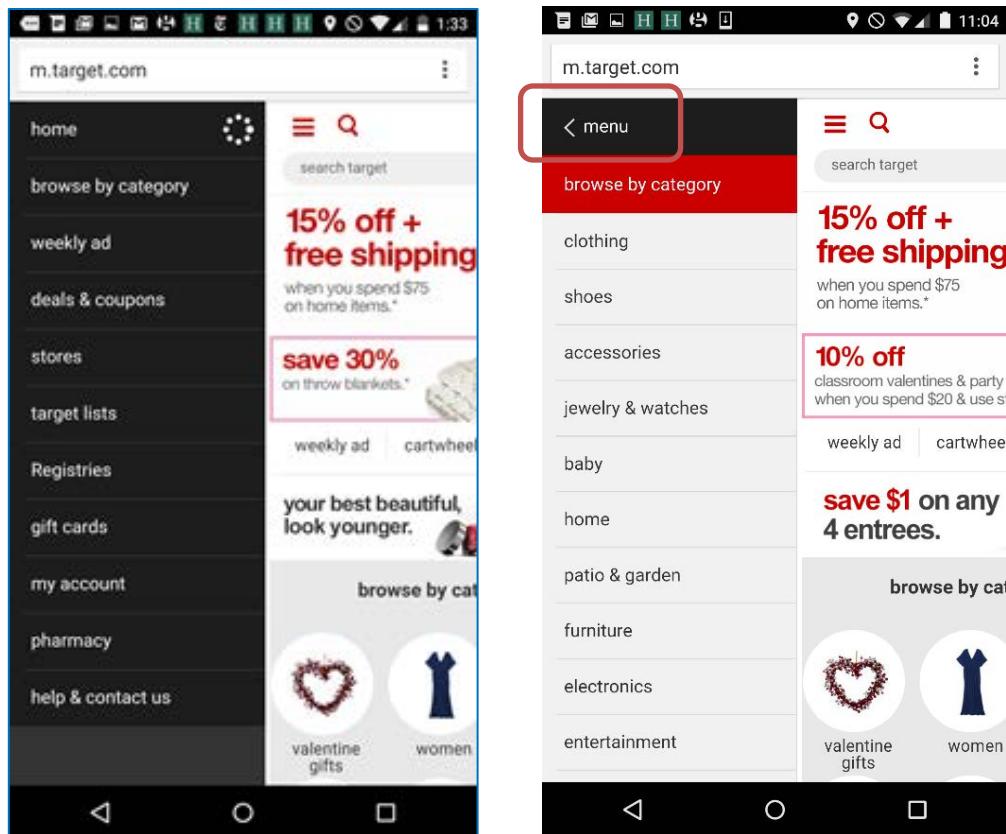


Wikipedia for iPhone used different fonts sizes, color, and indentation to represent the hierarchical structure of the items in its in-page navigation menu.

- 215. Always display the top-level categories in a navigation menu.**
 - 216. Do not require users to use a menu-back functionality to navigate back to the top-level categories in the menu.**
- The third option for implementing second-level navigation is to alter the main menu to display the secondary-level options. This choice has major disadvantages, so we firmly recommend against it:
- (1) It disorients users because the contents of the navigation menu changes as they make their selections; as a result they may forget where they are in the overall navigation hierarchy.
 - (2) It makes at least some of the top-level navigation categories invisible, thus obscuring users' understanding of the site's hierarchy.
 - (3) It requires users to use a menu-specific *Back* button to go back up in the navigation hierarchy. Users are often confused by such a button (or do not notice it), and often try to use the regular *Back* button (when available) to navigate back to the top-tier categories. As a result, in most cases, they are taken to a different page in the app or site, instead of navigating up in the menu.



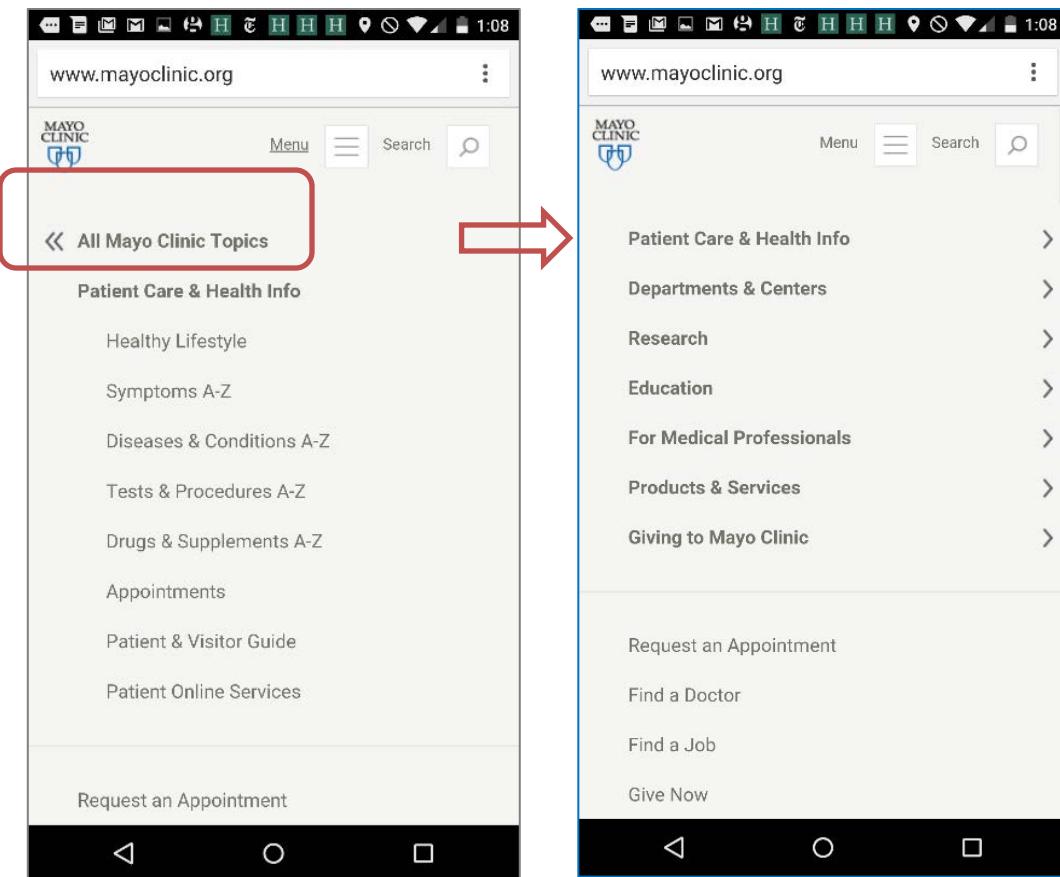
Ulta.com: Selecting the *Makeup* top category in the *Shop* menu replaced that menu with the second-level categories under *Makeup*. If users wanted to go back to seeing the top-level categories they had to tap the menu-specific *Back* button. (Had they tried the phone's *Back* button at the bottom of the screen, they would have been taken to the previous page in the browser's history.)



Target.com: Not only did the main-navigation menu (left) change once people selected *browse by category* (right screenshot), but it remembered state and stayed set to the second-level categories in all subsequent instances when people used the hamburger icon. If they wanted to go back to the main-navigation categories, they would have had to tap the *<menu* button.

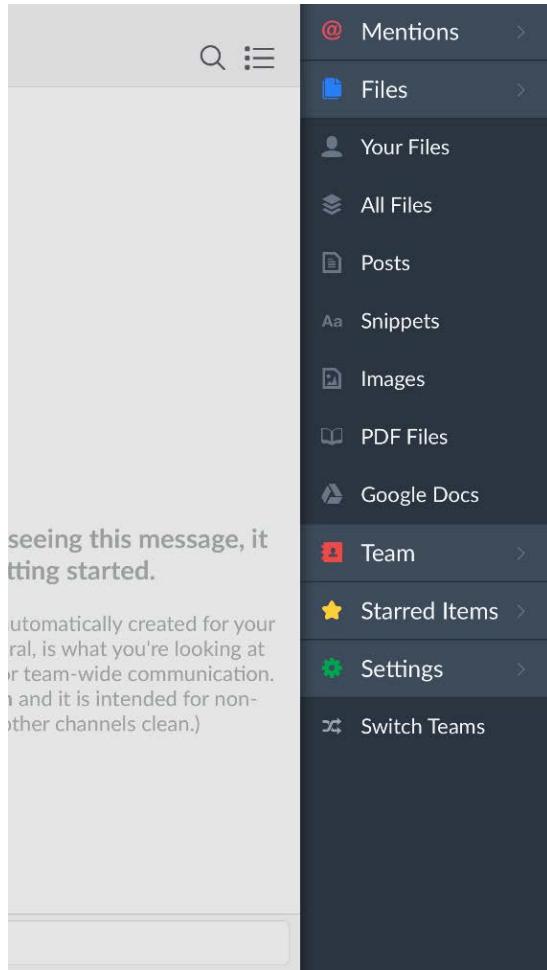
If users are shown second-level categories, they may forget, or, worse, never discover that there are higher-level categories available to them. Moreover, people who may be familiar with the higher-level categories from a desktop site may feel as if the mobile site was completely different, with totally distinct content and navigation categories.

For example, on Mayo Clinic's website, by default users were shown the second-level options under *Patient Care and Health Info*. Participants in our study never discovered that there were higher-level categories available to them.



Mayoclinic.org: The main navigation under the hamburger menu automatically showed users second-level topics under the *Patient Care & Health Info* (left screenshot). Should they have wanted to see the top-level categories (right screenshot), they would have had to select *All Mayo Clinic Topics*.

Mayo Clinic probably made this decision because it considered *Patient Care & Health Info* to be its most important (or most frequently visited) top category. Unfortunately this design decision ended up impacting the usability of the site navigation. They could have used an expanded accordion instead to show the subtopics under *Patient Care & Health Info*, while keeping the top-level categories visible as well (see Slack's example below).

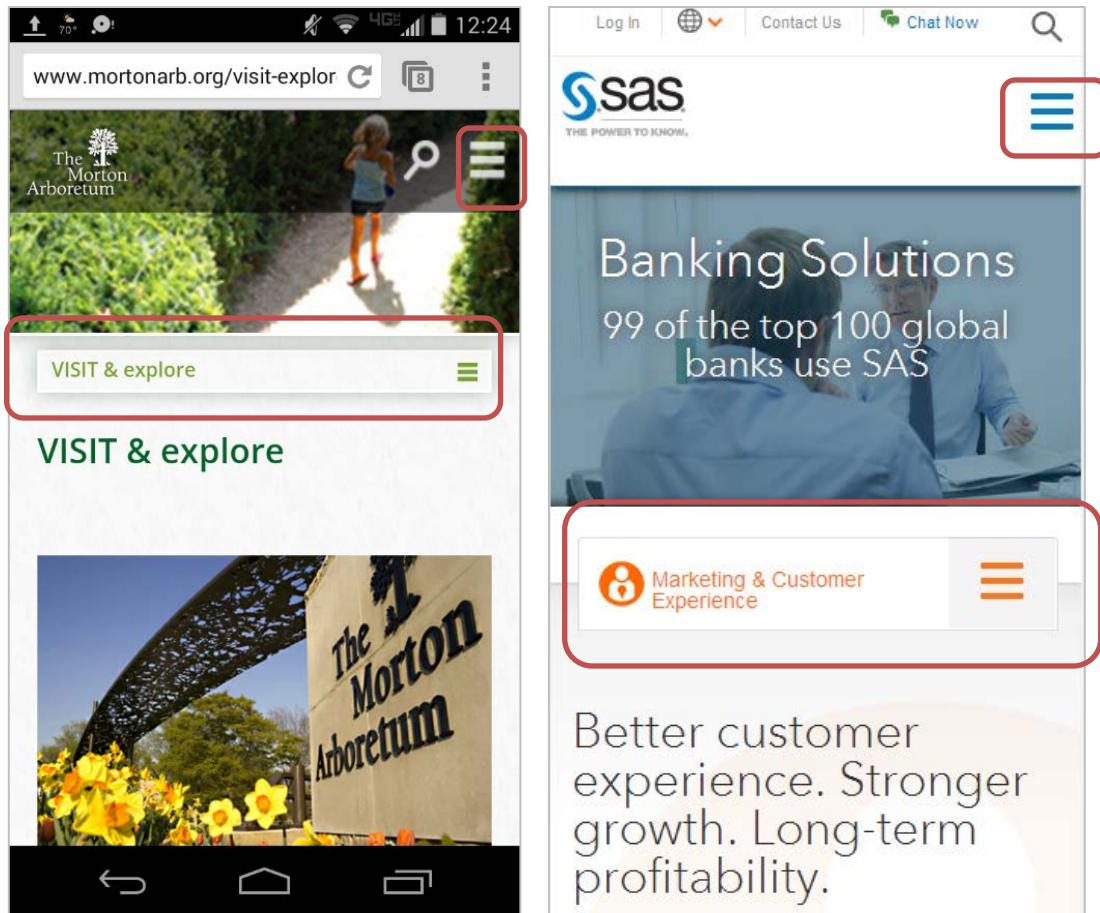


Slack for iPhone: The most important top-level category (*Files*) was expanded and the corresponding second-level categories shown under it. The other top-level categories were still visible. (Note that there should have been a stronger visual-design difference between the top-level categories and the second-level categories.)

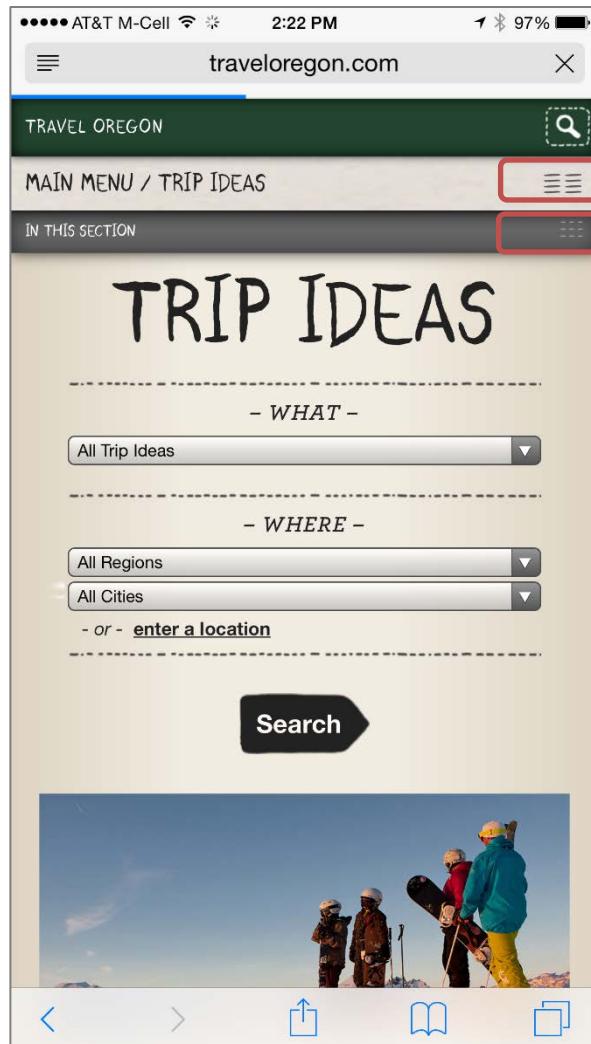
217. If you use two navigation menus on the same page make sure that there is enough physical separation and visual differentiation between them.

Sometimes sites and apps use two menus on a page: one for the main-navigation and the other for either in-page navigation or for section navigation (if, for instance, the page is a landing page for a main category).

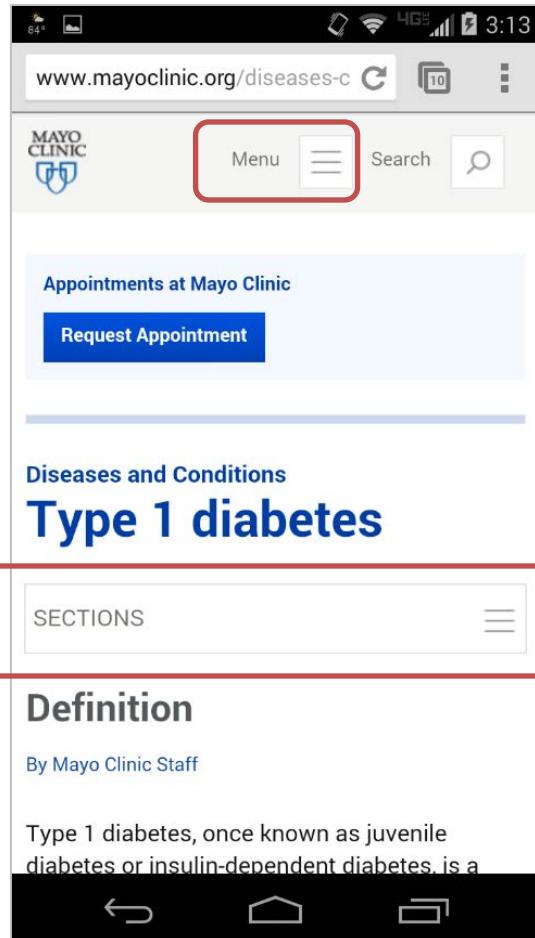
With two menus the challenge is to ensure that their function is clear and that people do not mix them up.



Both Morton Arboretum's site and SAS's site used double hamburger menus: one for the site's main navigation and the other for the different category sections. While the two were fairly far apart in both cases and had different colors, the secondary-navigation menu had a weak label and lacked discoverability.



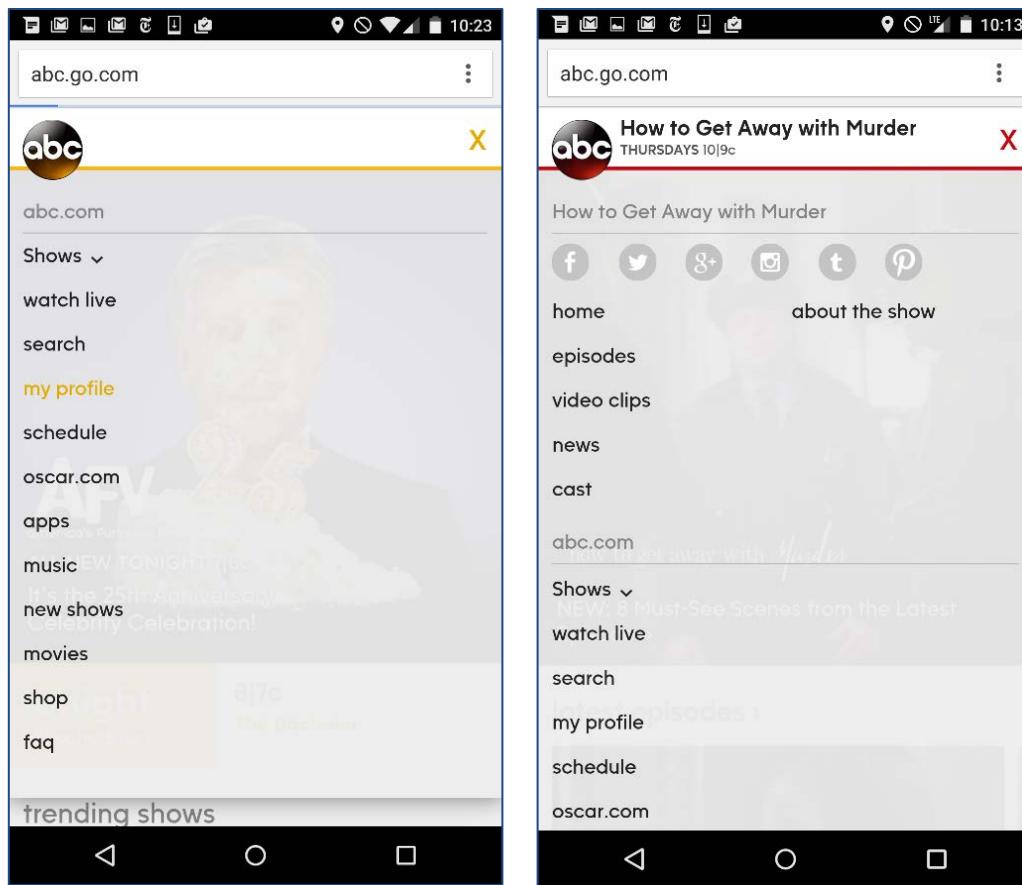
TravelOregon.com used two different menu icons for site navigation and for in-section navigation. While the two were fairly different, they were placed too close to each other and too far away from their respective labels.



Mayo Clinic used two hamburgers: one for the top navigation at the top of the page, and the other for topic navigation. The two menus were far apart, and the *Sections* menu was placed below the section heading, clearly indicating the type of content within. Users had no trouble differentiating between the two menus.

218. Do not change the contents of the main-navigation menu to secondary or in-page navigation.

Once people have expanded a menu, they expect its contents to be the same throughout the site or app. If the contents of the main menu changes depending on the section of the site (like in the ABC example below), the site risks to have those page-specific options stay undiscovered.



ABC.com: The content of the site's hamburger menu depended on the page. The left screenshot shows the content of the menu on the homepage; the right screenshot shows the content of the same menu on a show page.

Back Button

On a touch screen, accidents are bound to happen: people will eventually touch something by mistake and be puzzled by an unexpected effect. The larger the touch screen, the higher the probability of an accidental touch. (Smaller screens can usually be manipulated by just moving fingers around and holding the phone while keeping the hands "fixed" relative to the phone. As screens get bigger, thumbs and fingers no longer can reach all the screen when the hand is fixed; as a result, people need to move their entire hands — not just the fingers — to use the device, and thus can accidentally touch the screen.) When such an unexpected event happens, users reach first for an undo function, traditionally supported by a *Back* button. In the browser, most often *Back* has the intended functionality and people are able to undo an accidental touch. But apps, and especially iOS apps, pose a difficulty.

The *Back* button is often interpreted in one of two ways: (1) undo; (2) up in a hierarchy. Most of the problems in today's (mostly iOS) interfaces stem from implementing *Back* as up instead of as undo.

Back as up will take the user up in the interface, on a page that is at the immediate higher level in the page hierarchy. *Back* as undo will take the user back to the

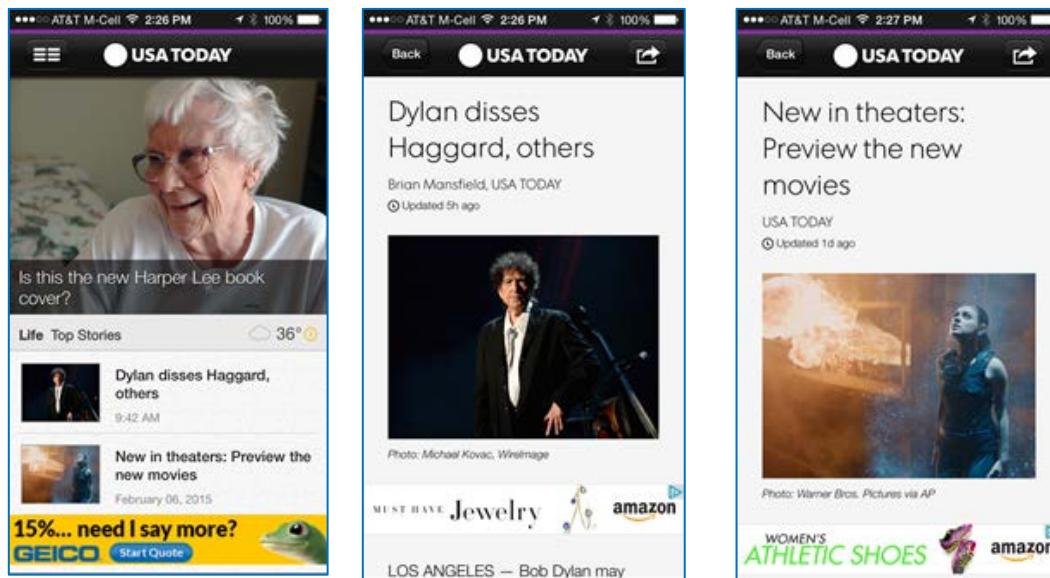
previous page that they were on. *Back* as undo is usually more important than *Back* as up and should be prioritized in interface design.

219. Users should always be able to go back to the previous page.

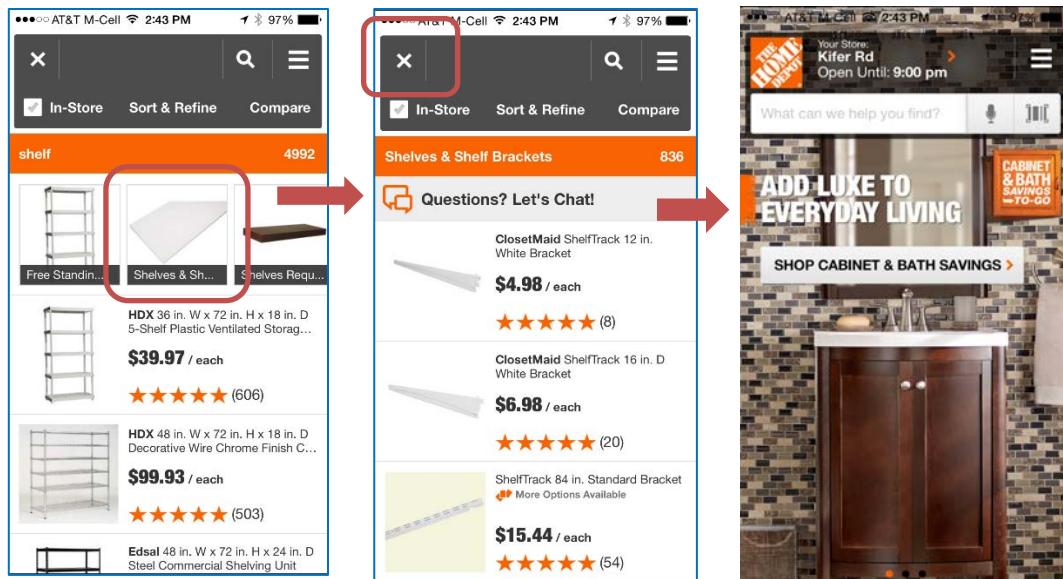
220. [App] Make sure that people can easily undo an accidental touch by providing a *Back* as *undo* functionality.

In Android, this guideline is usually easy to follow: the physical *Back* button on Android phones is most often correctly programmed to undo the most recent user action.

In iOS many apps interpret *Back* as up in the information hierarchy of the app instead of undo. As a result, when people need to go back to the previous page, they may not be able to click *Back* and instead may have to retrace their steps and thus incur a higher interaction cost, as in the examples below.



USA Today: Users could go through these 3 pages in the order in which they are listed above (left to right), but if they tapped the top *Back* button in the 3rd page, they would go back to the 1st page that included the list of all articles instead of being taken to the page they were on before (corresponding to the middle screenshot).



Home Depot for iPhone: When users selected one of the facet links at the top of the screen, they navigated to a page that contained that type of products. However, on that page there was no *Back* button to allow them to revisit the list of products; closing the view took them up in the hierarchy, to the homepage.

- 221. [App, Android, Windows] If the phone has a physical *Back* button, make sure that the physical *Back* button functions as undo and allows users to return to the previous page within the app.**

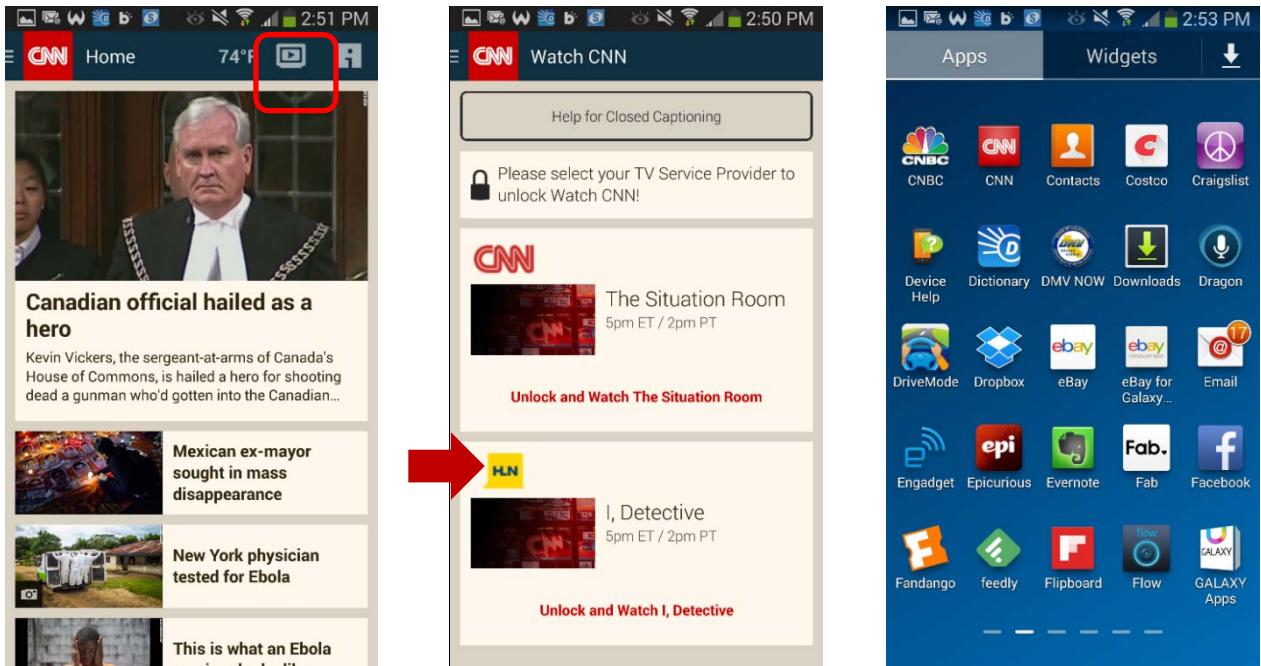
Even platforms that have a physical *Back* button, surprisingly, suffer from problems because it does not always work correctly.

On phones with physical buttons, the *Back* button usually has 2 functions: (1) dismiss a menu; and (2) undo, or go back to a previous page.

Unfortunately the *Back* button is not always used consistently across apps.

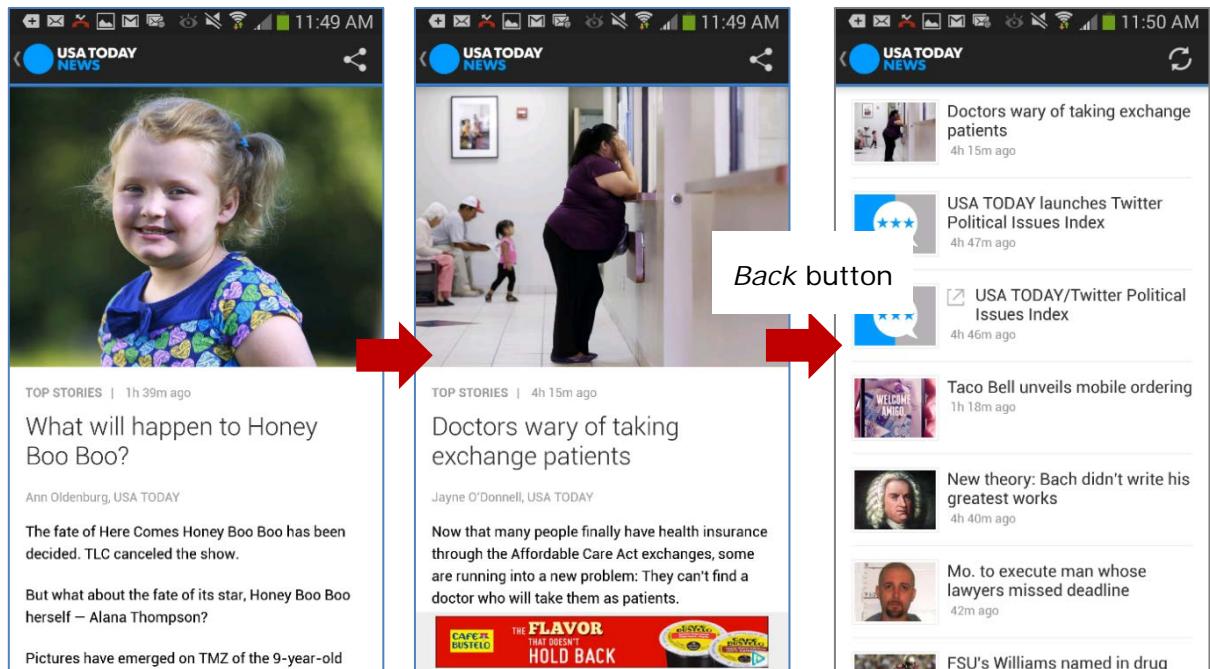
One of our participants was looking for a moisturizer using the Walgreens app for Android. She picked one, looked at it, and then decided she would not buy it. However, when she tapped the physical *Back* button to go back to the search results, she was taken back to the app homepage. (Since then, Walgreens has fixed this problem.)

In the CNN app, we observed a similar issue: tapping *Back* on the video page took the user out of the app instead of leading back to the previous page.



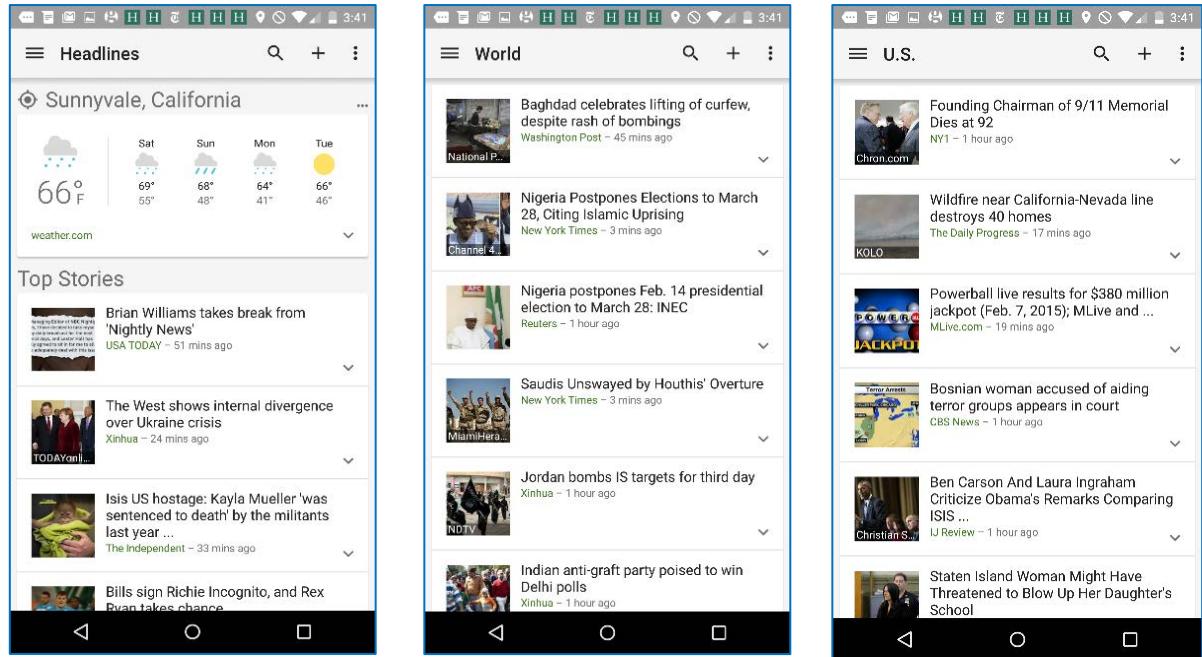
CNN on Android: When the video icon was tapped, the app loaded the video section. There, tapping the *Back* button led to the phone's home screen instead the app's home.

In the USA Today Android app the physical *Back* button acted as an up button instead of undo: it went back to the list of stories, not the previous page.

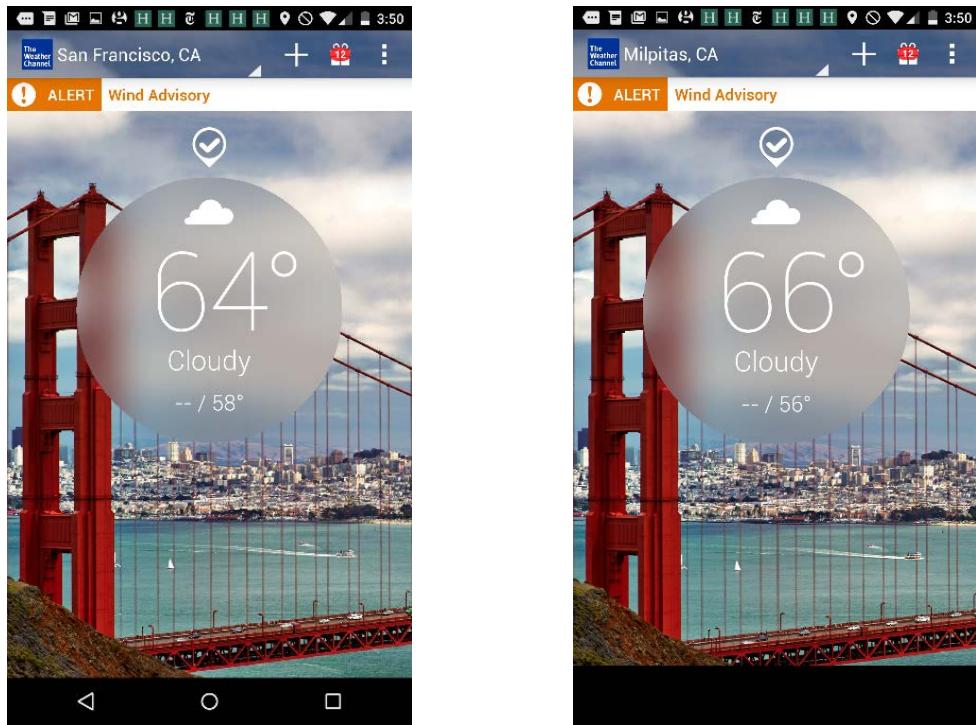


USA Today: The *Back* button went to the article list, not to the previous page.

In the News and Weather Android app, the *Back* button was used to mean up instead of the more traditional undo. Users could swipe through the different sections (from *Headlines* to *World* to *U.S.*), but pressing *Back* did not take them to the previous page; instead, it always lead to the highest page in the app hierarchy (*Headlines*).



News and Weather for Android: The physical *Back* button was designed to mean up instead of undo. When users navigated from *Headlines* to *World* to *U.S.* and pressed *Back*, they were taken up to *Headlines* instead of back to *World*.

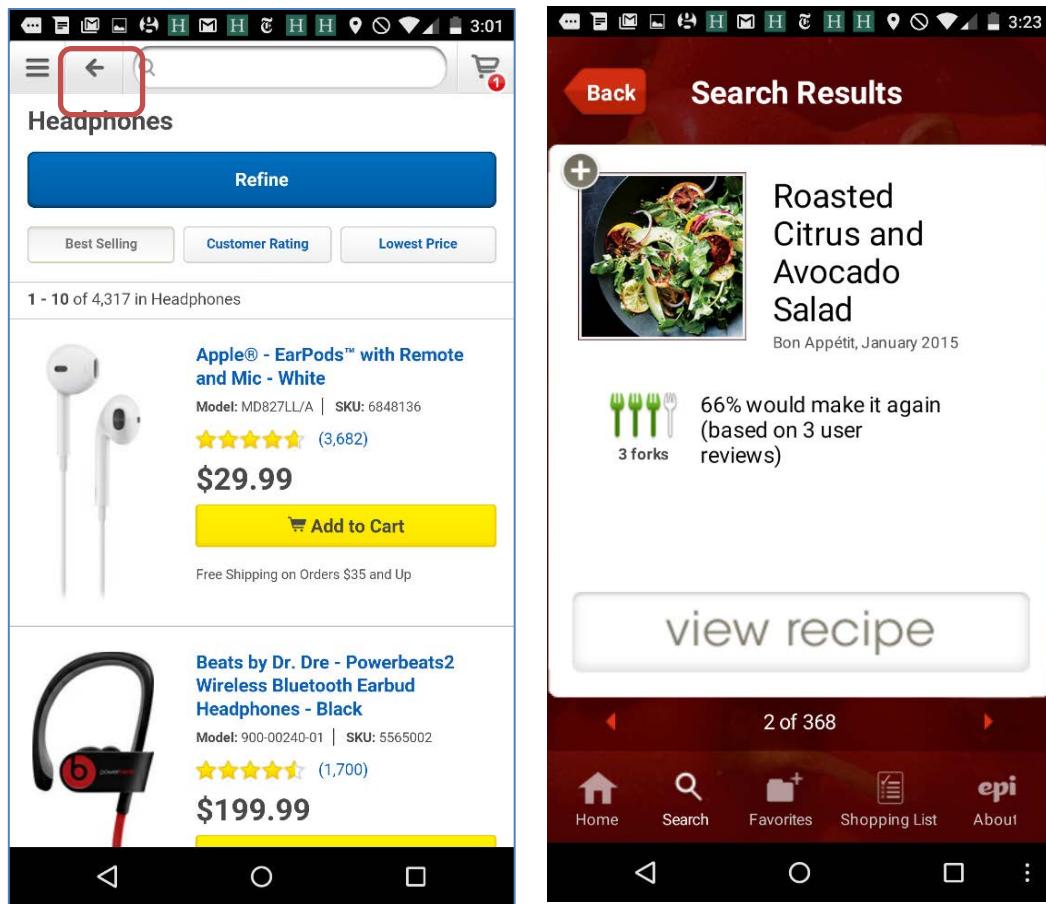


Weather Channel for Android: When users switched from San Francisco to Milpitas and then pressed the phone's *Back* button, they were taken outside the app.

222. [App, Android, Windows] Do not include a separate *Back* button if the phone already provides one.

Multiple *Back* buttons must be avoided at all cost: they always confuse users, who are not sure which must be used for their goal (see also the discussion about menu *Back* buttons under guideline 216). Even if the buttons did the same thing always, it'd still be wrong to have two because their presence would make users ask themselves: why are there two buttons and what's the difference between them? Which is the right one to use?

It seems hard to believe that an app would waste screen space with a *Back* button instead of using a perfectly good one provided by the phone. Often, examples like the one below stem from trying to produce a unique design for both Android and iOS platforms and indicate a hybrid app (see section Hybrid Apps).



Best Buy (left) and Epicurious (right) are Android apps that had a second on-screen *Back* button. Best Buy's *Back* button sometimes replicated the functionality of the phone's *Back*, and in other instances simply took users back to the home page. Epicurious used the on-screen *Back* to mean up. (While that functionality is useful, the button should have been signaled and labeled differently, so people could interpret it correctly.)

We could ask: why would two hamburgers work (like in the Mayo Clinic example in guideline 217), if two *Back* buttons don't? A possible explanation is that *Back* is understood by users to mean "undo", whereas the hamburger icon doesn't communicate *what* features are available, but only that *some* will appear by pressing it. In that sense it's similar with checkboxes or radio buttons, which can turn a feature on or off, but, to know what is being changed, one will have to read the label.

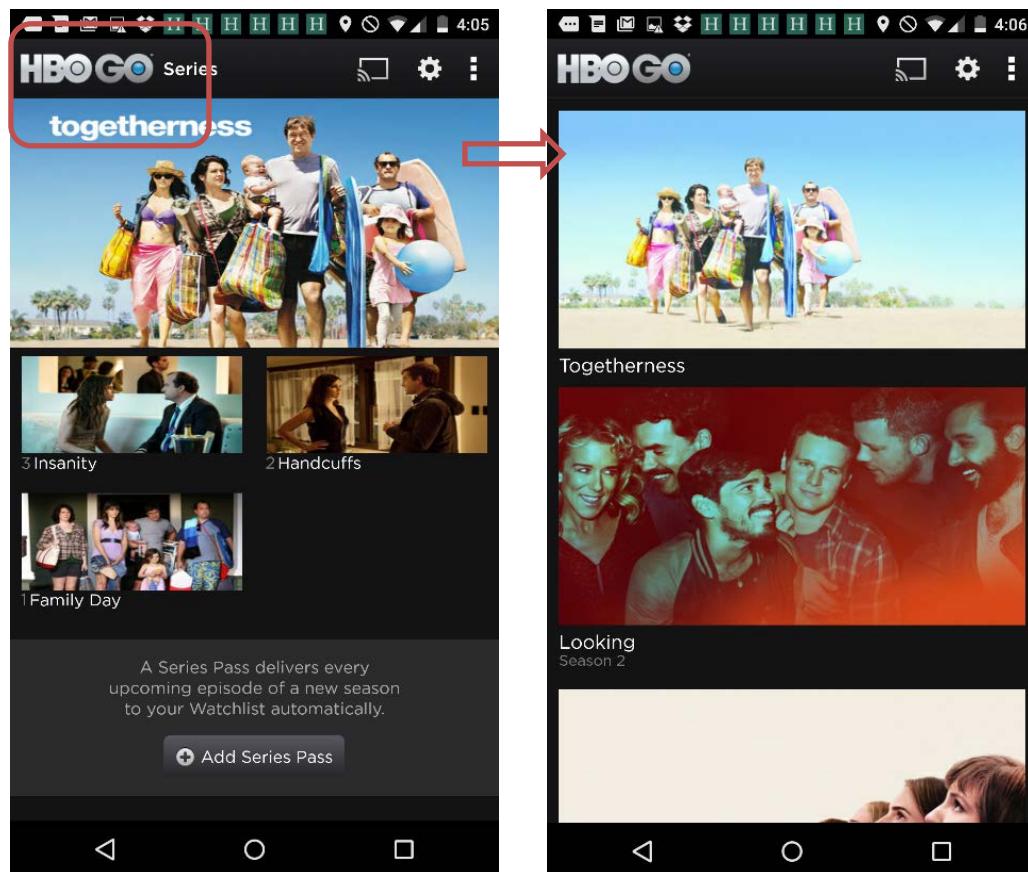
223. [Web] Do not include a *Back* button on a website.

Browsers already have *Back* buttons built in and users know how to use them.

224. [Android] On the homepage, the global *Back* button should take the user back to the previous page and not outside the app.

Even with Android, the implementation of the *Back* button on the homepage is incorrect, as it typically takes the user outside of the app instead of taking them to the previously visited page.

In the HBO Go app, users could navigate to the homepage from anywhere within the app by pressing the logo. However, once on the homepage, pressing the *Back* button took them outside the app instead of allowing them to go back to the previously revisited page within the app.

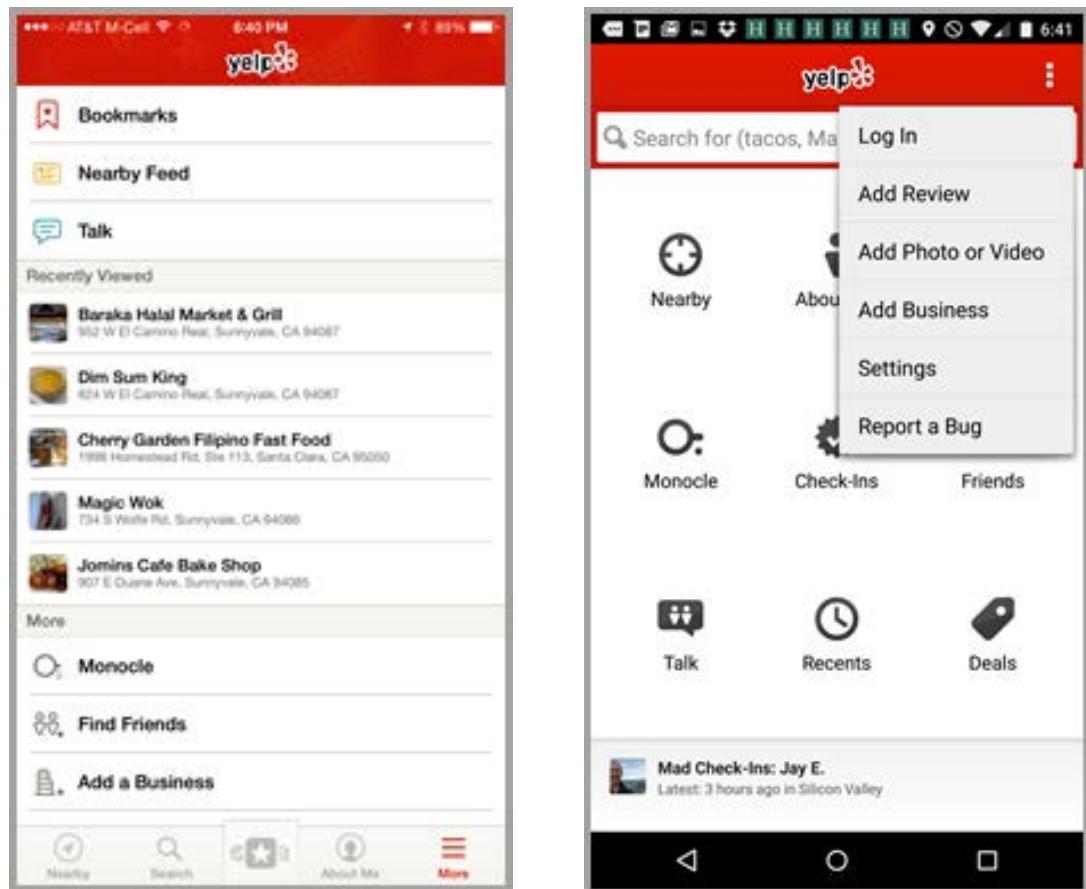


HBO Go for Android: Pressing the logo on a series page took users to the homepage; from there they could not go back to the series page using the *Back* button. Instead the *Back* button took them out of the app.

Hidden Navigation: More and Action Overflow

Literally speaking, any type of navigation placed under a menu is hidden, and, as discussed at the beginning of this section, suffers from poor visibility and discoverability. However, before the hamburger menu became popular for mobile navigation, **hidden navigation** referred to those app-specific navigation options

that were hidden under the action-overflow button on Android or the *More* tab button⁵⁹ on the iPhone. As designers realized that using a hamburger menu has serious disadvantages, they revived the tab bar (and the *More* tab or, in Android, the action overflow button).



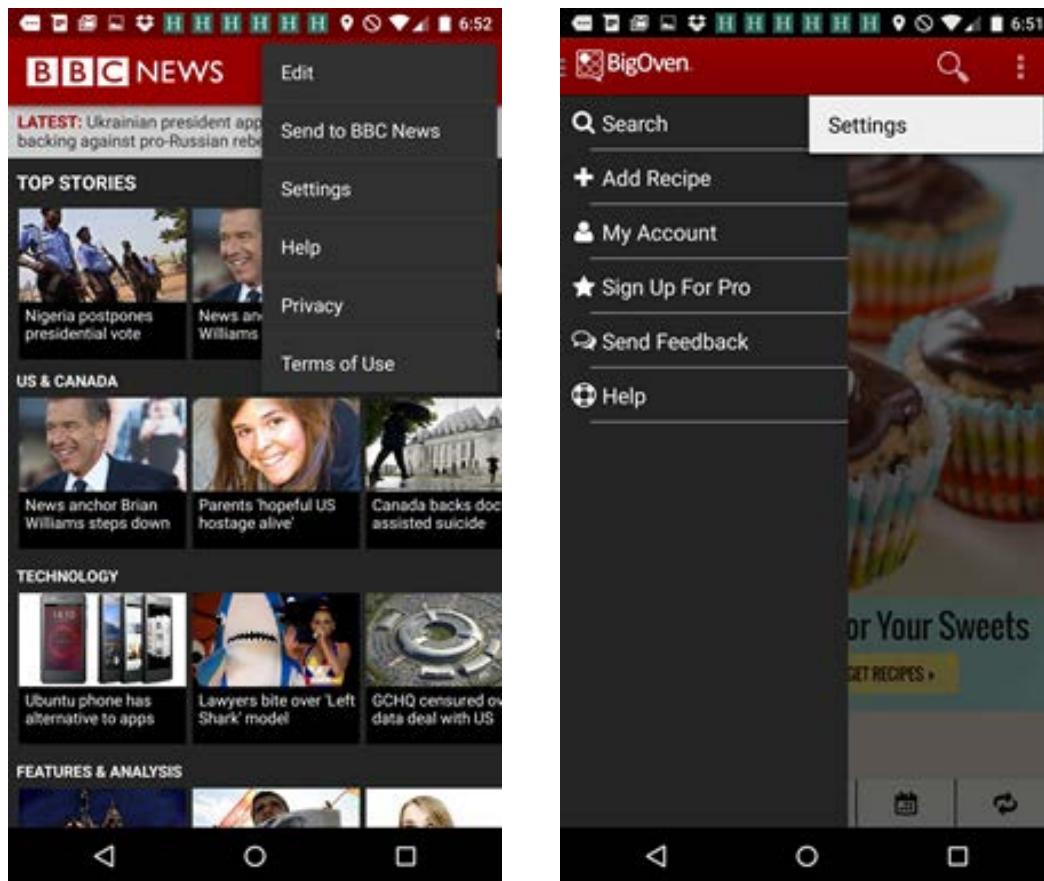
Yelp for iPhone (left) and Android (right) both used hidden navigation; the iPhone version under the *More* tab in the bottom right corner, and the Android version under the action overflow button in the top right corner.

There is one difference though between hiding the main navigation under a menu and using the *More* tab or action overflow: because some navigational choices are visible, users assume that whatever is not visible is going to be of less importance. As a result, they are even more likely to overlook the *More* tab and the action overflow button than they are to ignore the main-navigation menu.

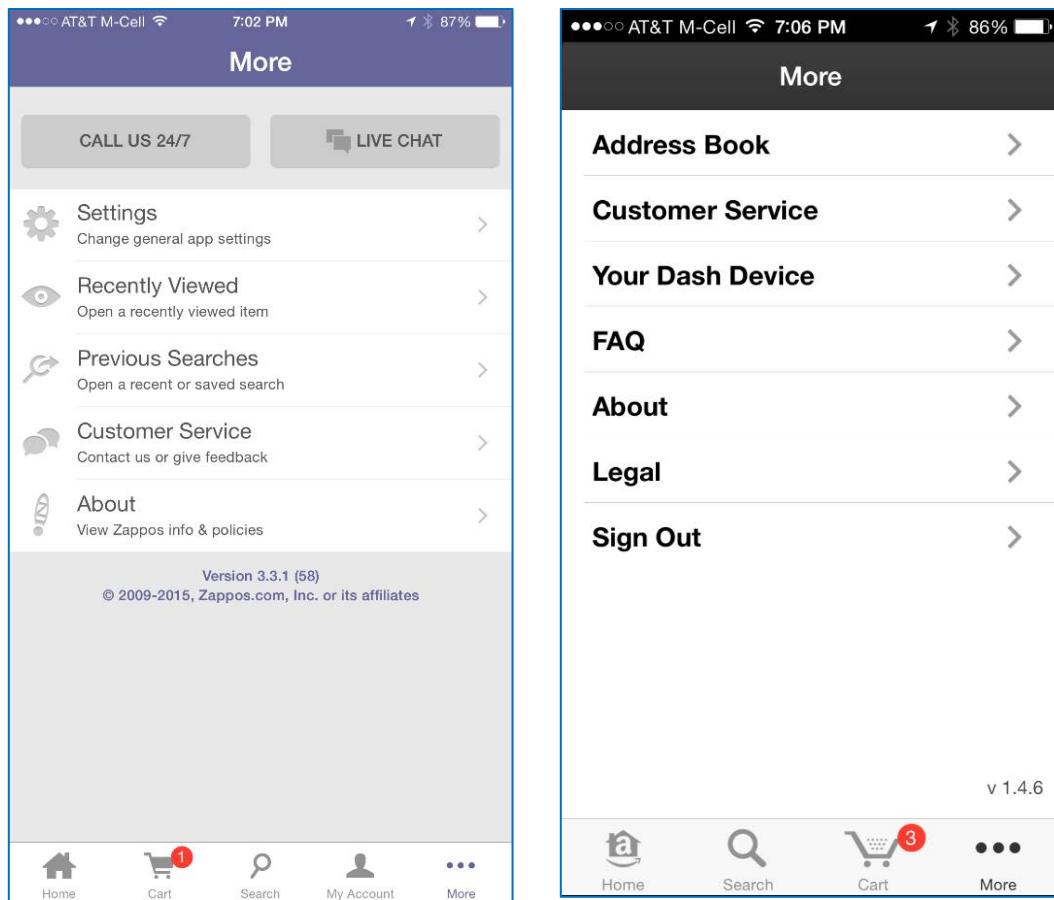
⁵⁹ Because in portrait orientation the tab bar on iPhone only supports 5 options, if an app has more than five tabs, four are displayed, and the rest are all grouped under a *More* tab.

225. [App] No essential functionality (e.g., search) or navigation options should be hidden under the *More* tab or under the action overflow button.

Luckily, there seems to be an emerging standard in this direction: apps usually reserve the action overflow or the *More* tab to those tasks that are least frequently done by users. Things like *Terms and Conditions*, *Reporting a bug*, *Settings* are usually delegated to this area of the app.

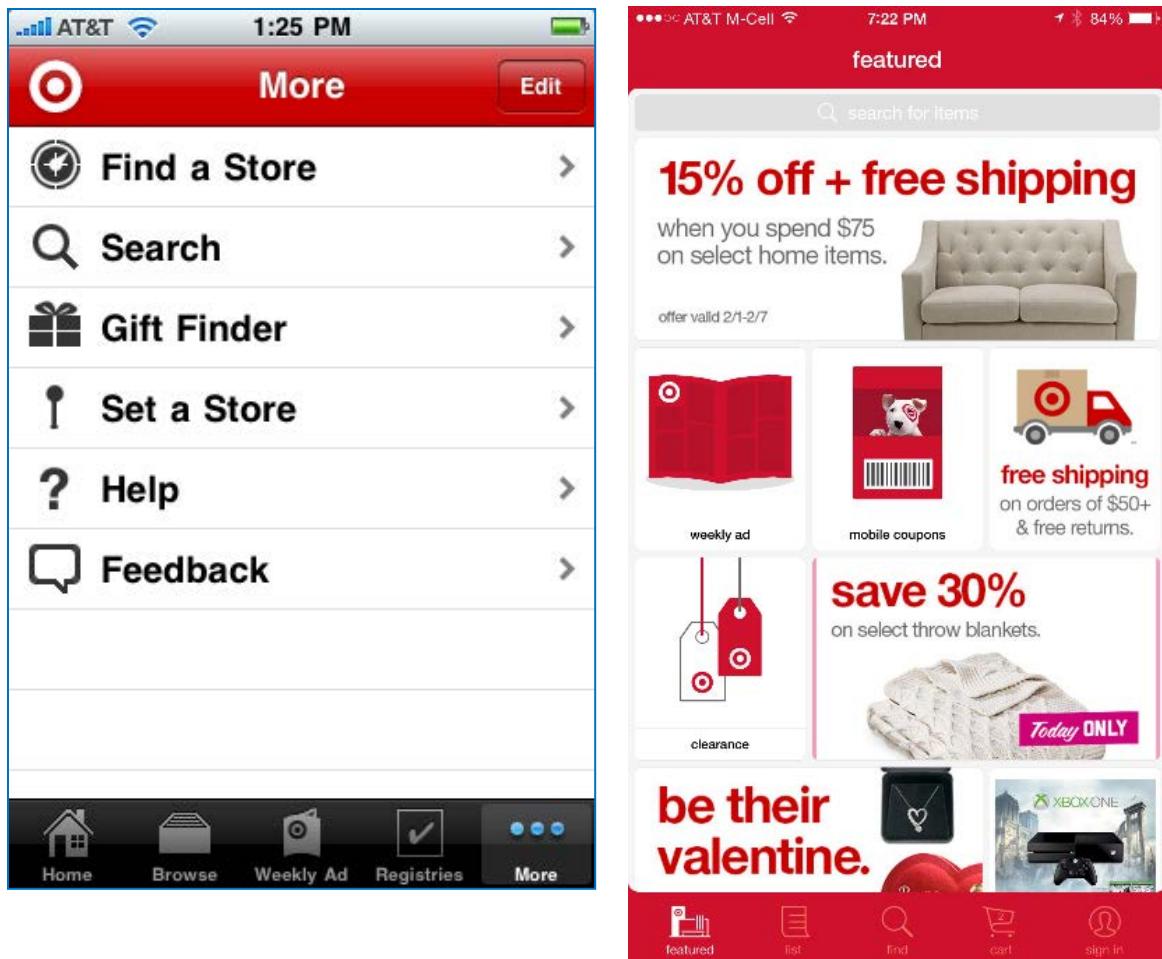


Android apps BBC (left) and Big Oven (right) placed only nonessential functionality under the action overflow button.

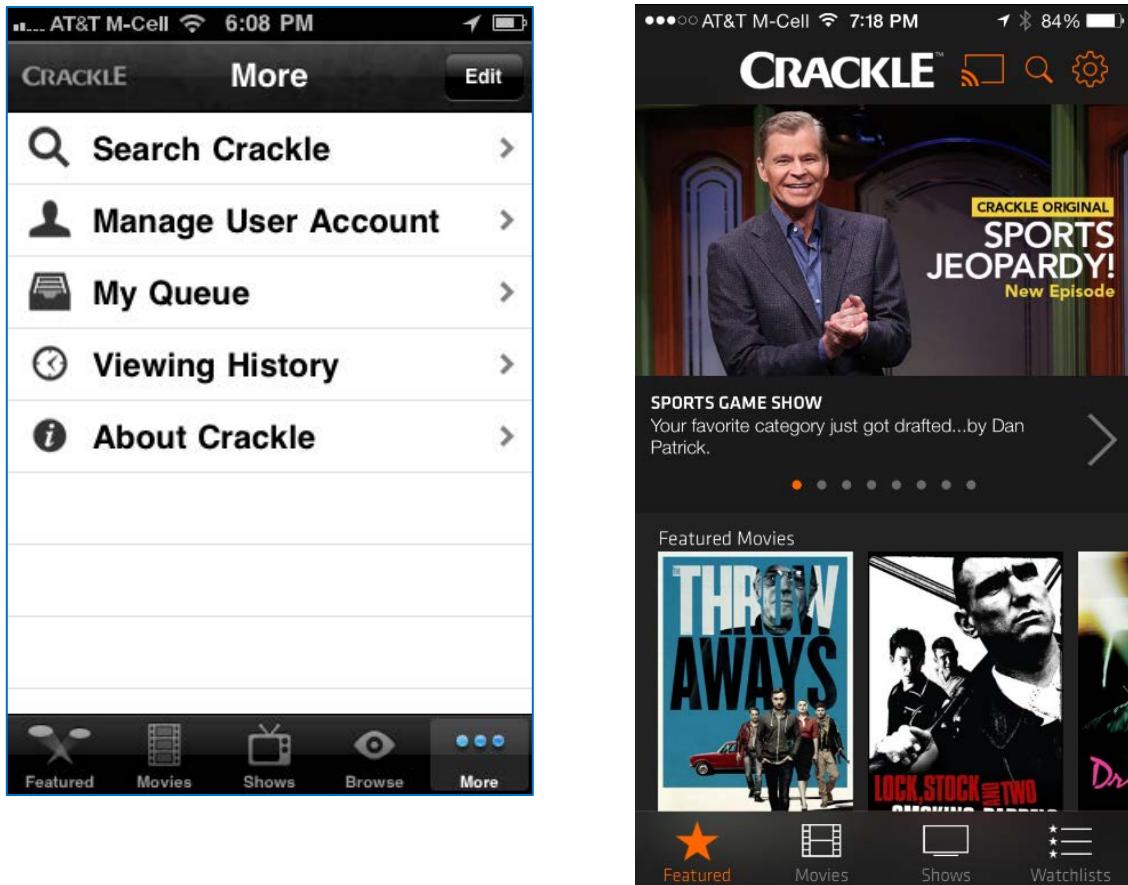


iPhone apps Zappos (left) and Amazon Fresh (right) fit only nonessential functionality under the *More* tab.

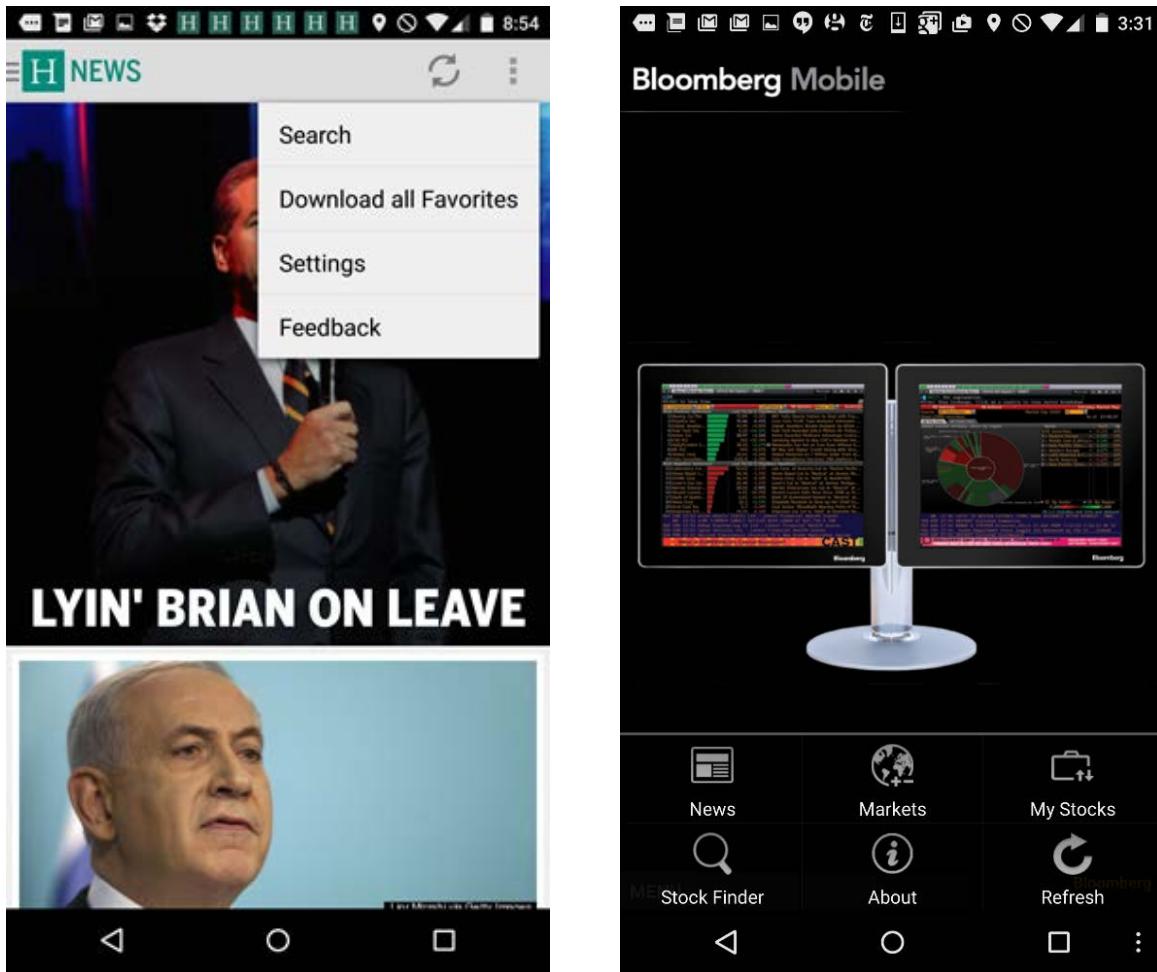
This hasn't always been the case though. One of our users was looking for a cleaner in an older version of Target's iPhone app. In that version of the app, search was under *More*; she never found the search box and had to use the category structure to find the cleaner, putting herself through the ordeal of guessing under which category the cleaner could hide. Since then, Target has fixed their app several times and has given search a more prominent place. Similarly, Crackle for iPhone used to make the same mistake. Even today, apps such as Huffington Post hide search under the action-overflow menu.



Two versions of Target for iPhone: an older version (left) in which the search was hidden under the tab *More*, and a newer version (right) in which the search was one of the tabs.



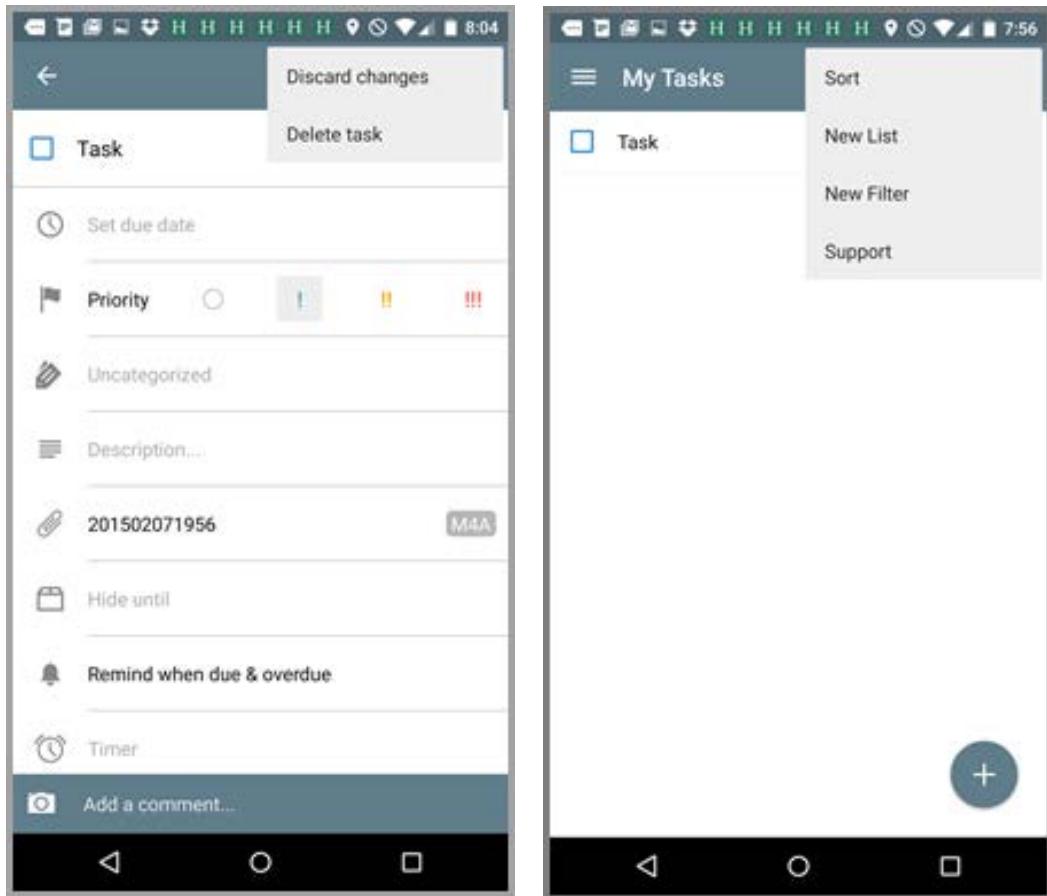
Two versions of Crackle for iPhone: in the older one (left), search was hidden under the *More* tab. The newer one has search at the top of the screen.



Huffington Post for Android (left) hid search under the action-overflow button, and Bloomberg for Android (right) hid all its navigation and search under the same button.

226. [Android] Do not change action-overflow functionality within the app.

Changing what the action-overflow button does from page to page is confusing. Some users may not realize that the action-overflow button behaves differently in different circumstances, and not press it, thinking that the functions that were discovered on a different page are irrelevant on the current page. And, as mentioned above, even if they pressed and discovered a different set of functions, keeping track of the different functionalities on different pages would be too hard.



Astrid for Android assigned different functionalities to the action-overflow button throughout the app.

SEARCH, FILTER, SORT

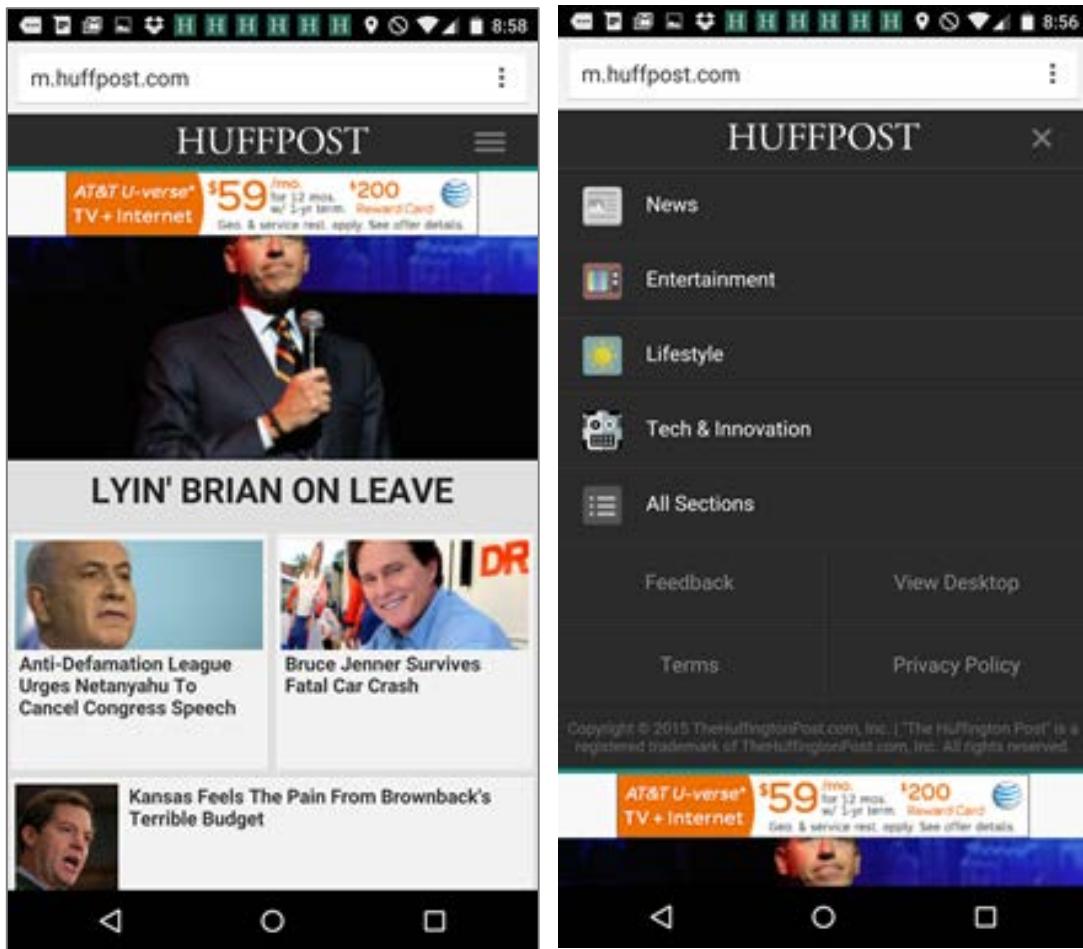
Search engines such as Google and Bing have taught users to rely on search for most tasks. Even on mobile, where typing has a high interaction cost, people use search and expect the same type of accuracy and behavior from site search as from large search engines.

Thus, on mobile (as on the desktop), search needs to be easily accessible and to work well. Most sites benefit from making the search tool visible and placing it in a prominent position on the homepage.

227. Most mobile websites and apps with a significant amount of content benefit from having a search tool on the homepage.

There are very few types of sites or apps that can get away with no search box. Productivity tools such as calculators or restaurant websites probably are among the few that qualify. Most other sites and apps are likely to need a search box, even if the huge majority of their users use search only occasionally.

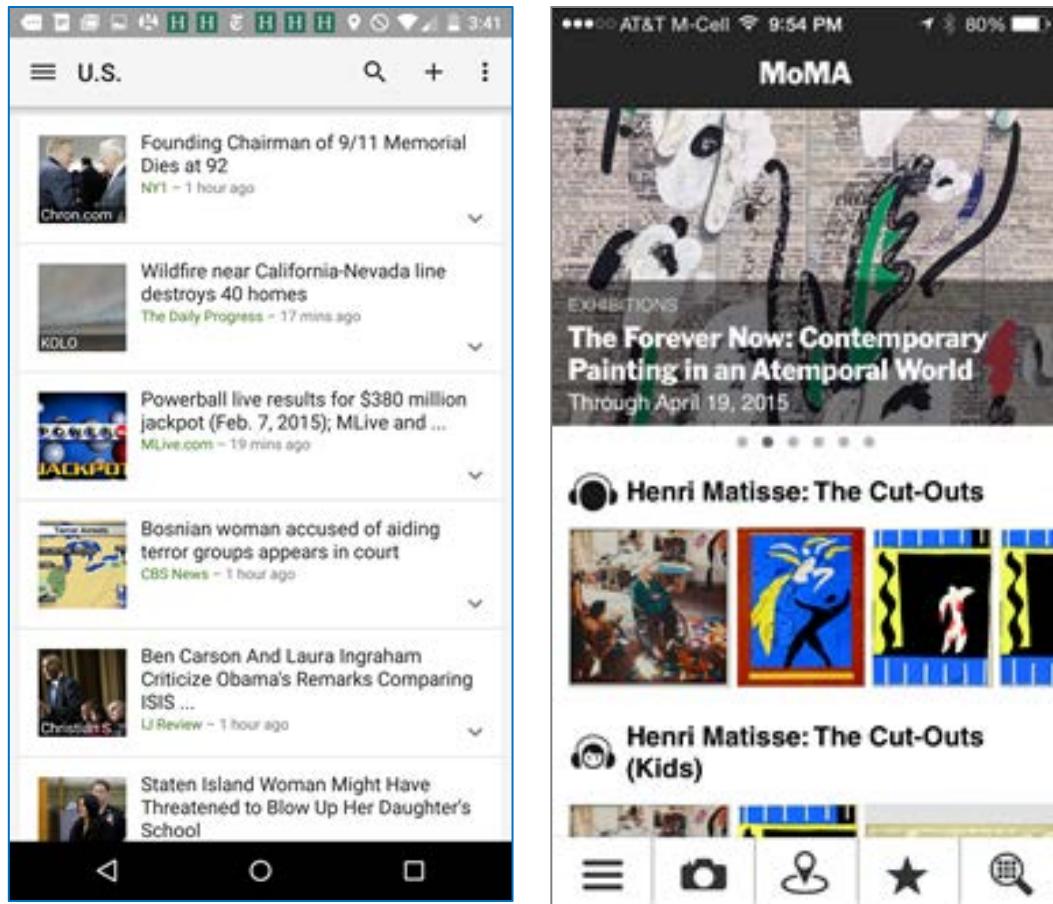
Even news apps or sites (that have an audience composed primarily of browsers) can benefit from having a search box, although perhaps they do not need to devote a large area of the screen to it.



Huffington Post's website did not have a search box.

228. The search functionality should be easy to locate.

As explained in guideline 225, search should not be hidden under the *More* tab or under the action-overflow button. It usually benefits from being present at the top of the screen, in a homepage header, or, for apps, in the tab bar.



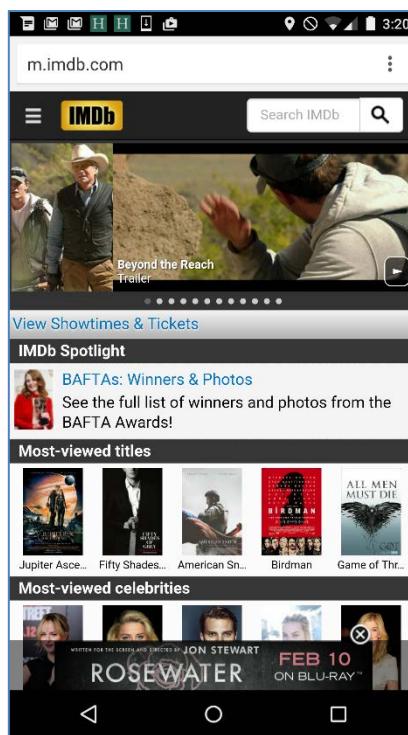
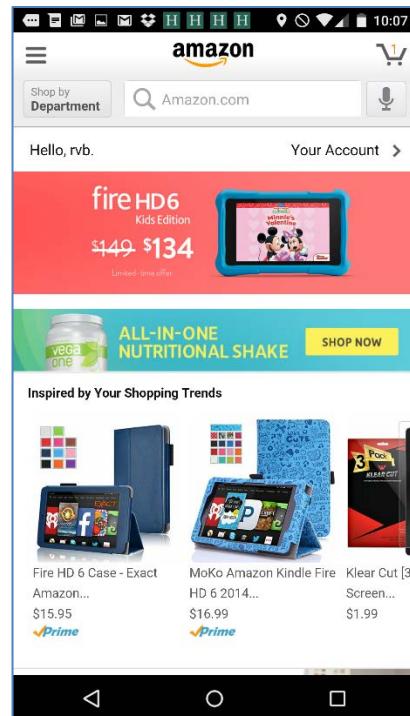
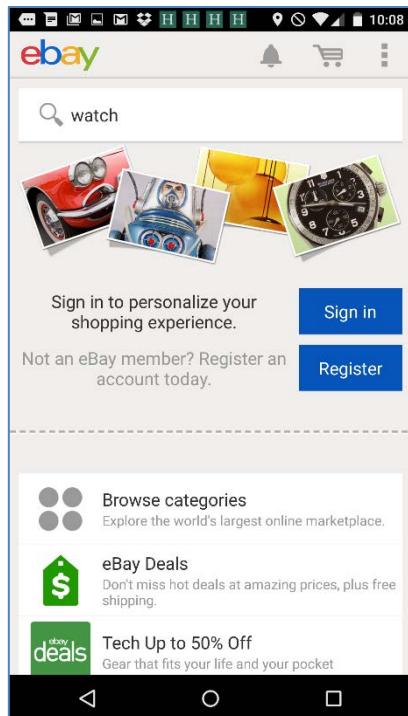
News and Weather for Android (left) placed the search icon in the header at the top of the page. MoMA for iPhone (right) positioned the (perhaps too stylized) search tool in the tab bar at the bottom of the screen.



AirBnb for iPhone placed the search tool in a “utility” bar at the bottom of the screen. This placement was the least discoverable among the examples shown here, since the utility bar was not standard among iOS apps.

- 229. If search is frequently used on your site, make it prominent by using a visible search box placed at the top of the screen.**

The more space devoted to search, the easier it's going to be for users to find it. While the magnifier-glass icon (with no search box) has become standard on mobile and has the advantage of a smaller footprint, it's still less discoverable than a search box. If your users (like Amazon's, IMDb's, or eBay's users) regularly rely on search to access information on your website, you should use a visible search box on the homepage and place it in a salient position (usually at the top of the screen).



Search was implemented with a visible search box whenever it was a primary way for users to access content on the site (top row: eBay and Amazon for Android; bottom row: IMDb.com and Epicurious for iPhone).

230. If search is only occasionally used on your site, use a magnifier tool for the search box and place that close to the top of the page.

Even when most of your users do not engage in search, if search is hard to find, those who need to use search may not be able to locate the information that they need and leave your site or app in favor of a search engine.

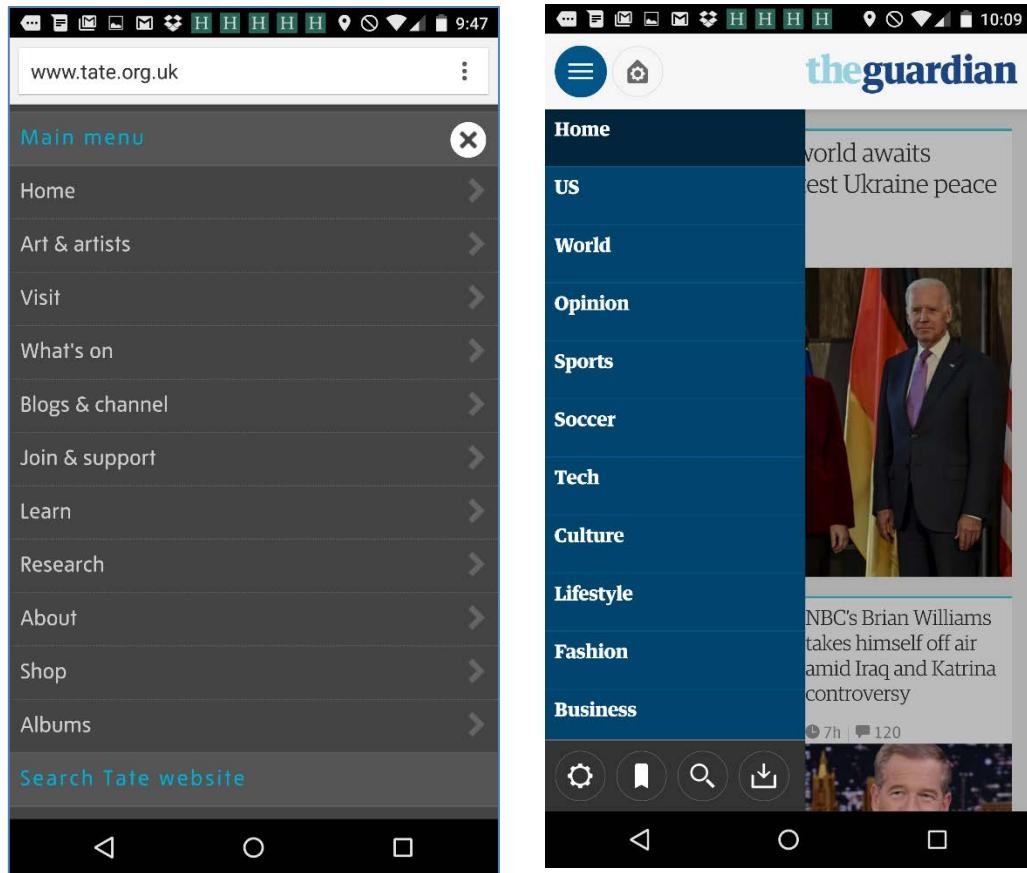
A search icon does not need to take up significant space on the screen and can be a reasonable compromise for those sites and apps whose users rely more on browsing than on search.

One of our users was searching for information about a company on CNN Money's website; the search box on that site was at the bottom of the page. (The website has since changed the design to include a prominent search box at the top of the page.) This placement might have seemed appropriate for CNN Money, since it is primarily a browsing site. However, for this searcher, it was quite tedious to scroll through an entire page of articles in order to find the search box. (In fact, many people may give up if they scroll for a while and do not find what they want.) She was able to find it and use it, but unfortunately the search did not return a satisfactory result. So she hit the back button and, in order to modify the search, she had to scroll again through the entire screen to get to the search box. This example shows the cost that is incurred to the searchers when the search box is not immediately available.

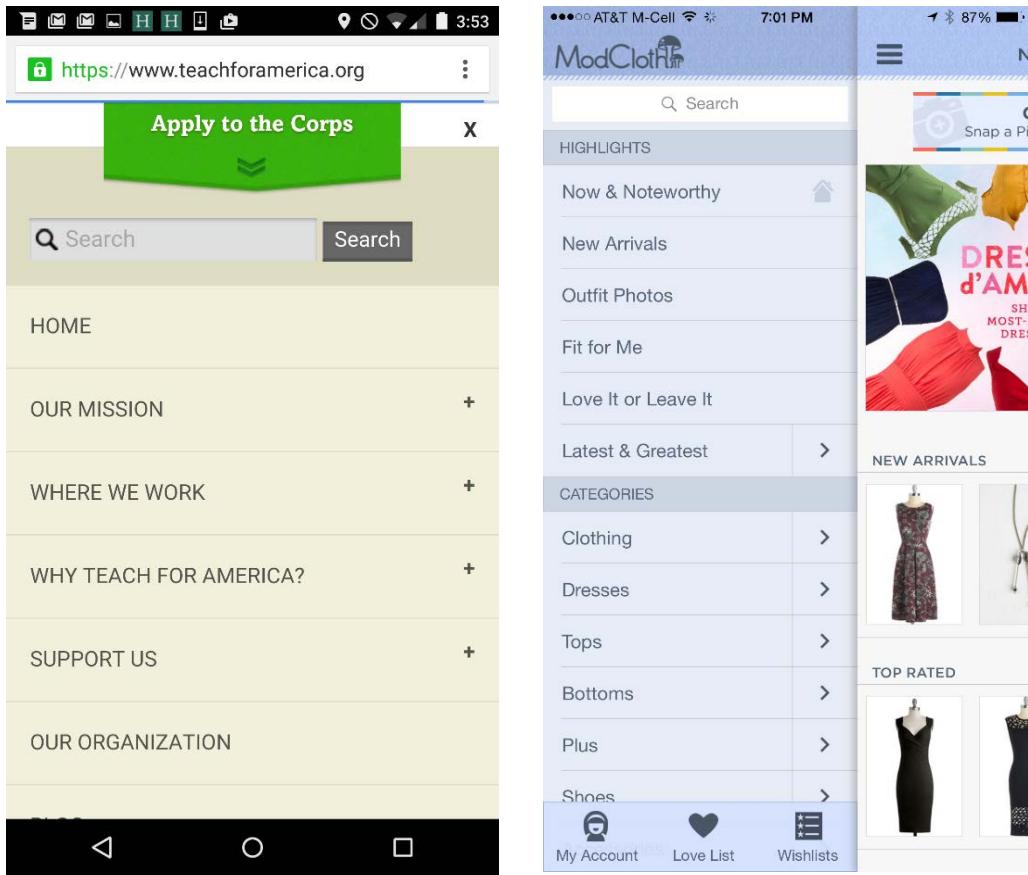
231. If the search tool is inside the navigation menu, place it at the very top of that menu.

Most people are used to search being in a prominent place such as the top of the screen; if they don't find it quickly, they may assume it does not exist on that site or in that app.

Users had a hard time finding the search feature on Tate Britain's website, tucked as it was at the bottom of the main-navigation menu. Even people who discovered it at one point during the session, subsequently had trouble remembering where it was. Hiding search under a menu automatically makes it less discoverable, so at least designers should give it prominence within the menu by placing it at the top.



On Tate's and The Guardian's websites, the discoverability of search was seriously hurt by its placement at the bottom of the navigation menu. Ironically, using a contrasting color for the menu item Search Tate website may have made it even less discoverable, because it didn't look like the other menu items. Instead, it looked like something closing off the menu and thus may not have been attended to as users scanned down the list of menu items to see what commands were available.

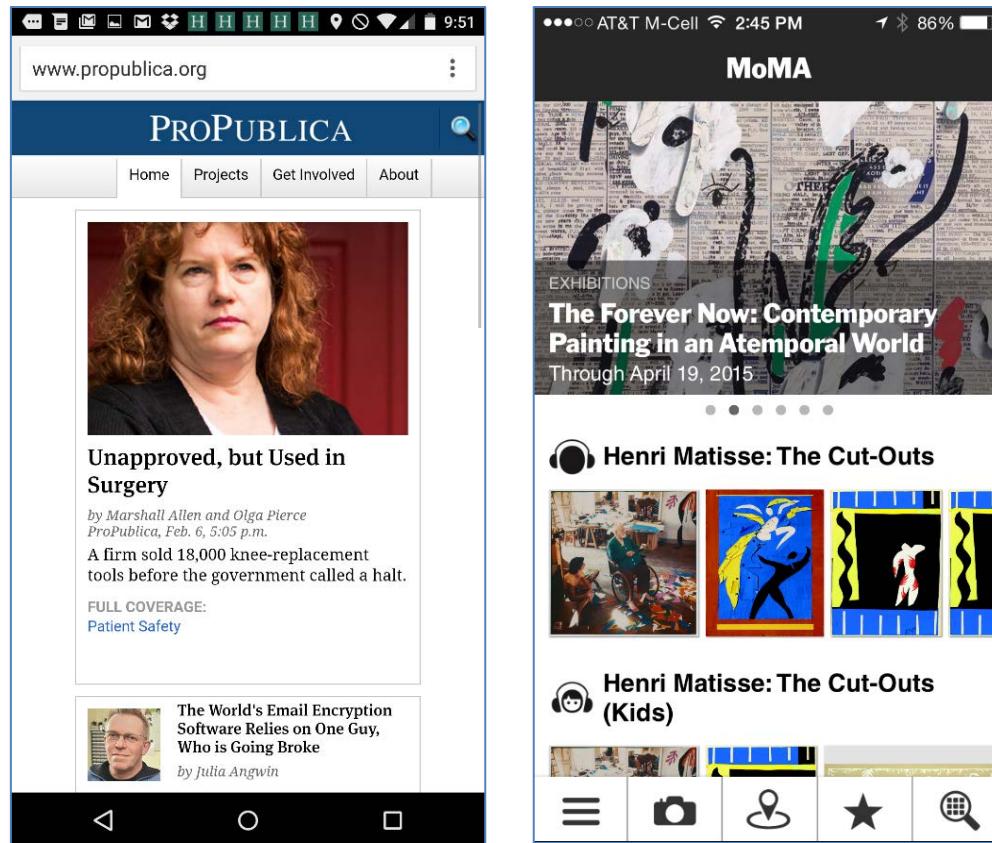


Both Teach for America and ModCloth for iPhone placed the search function inside a menu; however, in both cases it was at the very top of the menu.

232. If no search box is visible, use a standard magnifier-glass icon for search.

The search icon is not the place to show originality: the more standard it is, the faster users will recognize it. Some study participants had trouble finding the search tool on ProPublica's website because it looked too much like a real magnifying glass and too little like a regular search icon. (ProPublica was probably trying to be clever and relate the search tool to their fairly similar logo.)

The recommended magnifying glass serves as a reference icon, referencing the act of looking closely for stuff à la Sherlock Holmes.



ProPublica's website (left) and MoMA for iPhone (right) used search icons that were too styled and harder to recognize.



Zappos for iPhone had two search icons present on the homepage. While it's not necessary to duplicate the search function (see guideline 235 below), the search icons that they used were standard and easily recognizable.

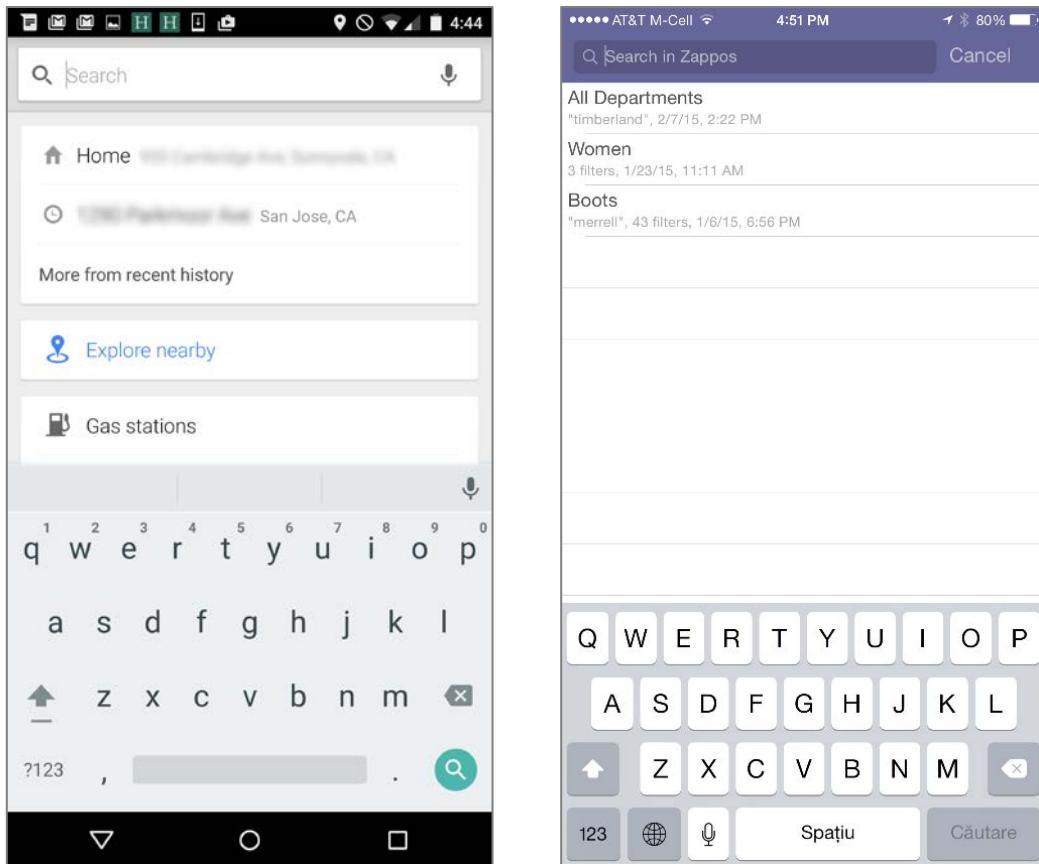
233. The length of the search box should be the largest possible size that will fit on the screen.

This guideline replicates the more general one (see guideline 60) for sizing textboxes in forms. Refer to the discussion for that guideline.

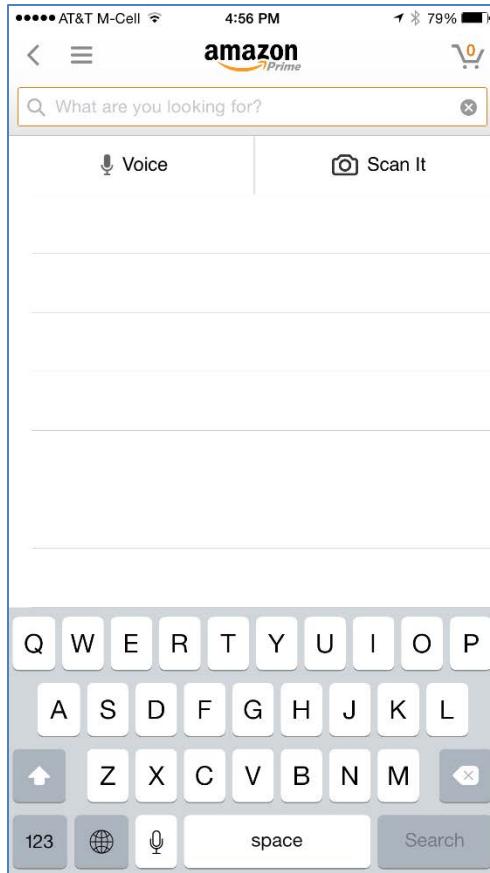
234. Use history-based suggestions for search. Present them to the users when they move input focus to the search field.

This guideline is a search version of the more general guideline for typing (67 and 72), which recommends giving users autosuggestions and history-based defaults to minimize typing.

Remember that users will often refine and reuse old queries, so access to a history of searches is vital. It is however not enough to allow them to access past search history; recent searches should be offered as suggestions as soon as the user moves the input focus into the search field.



Google Maps for Android (left) and Zappos for iPhone (right) made possible to retrieve recent searches. They both offered suggestions based on previous searches when the user moved focus into the search field.



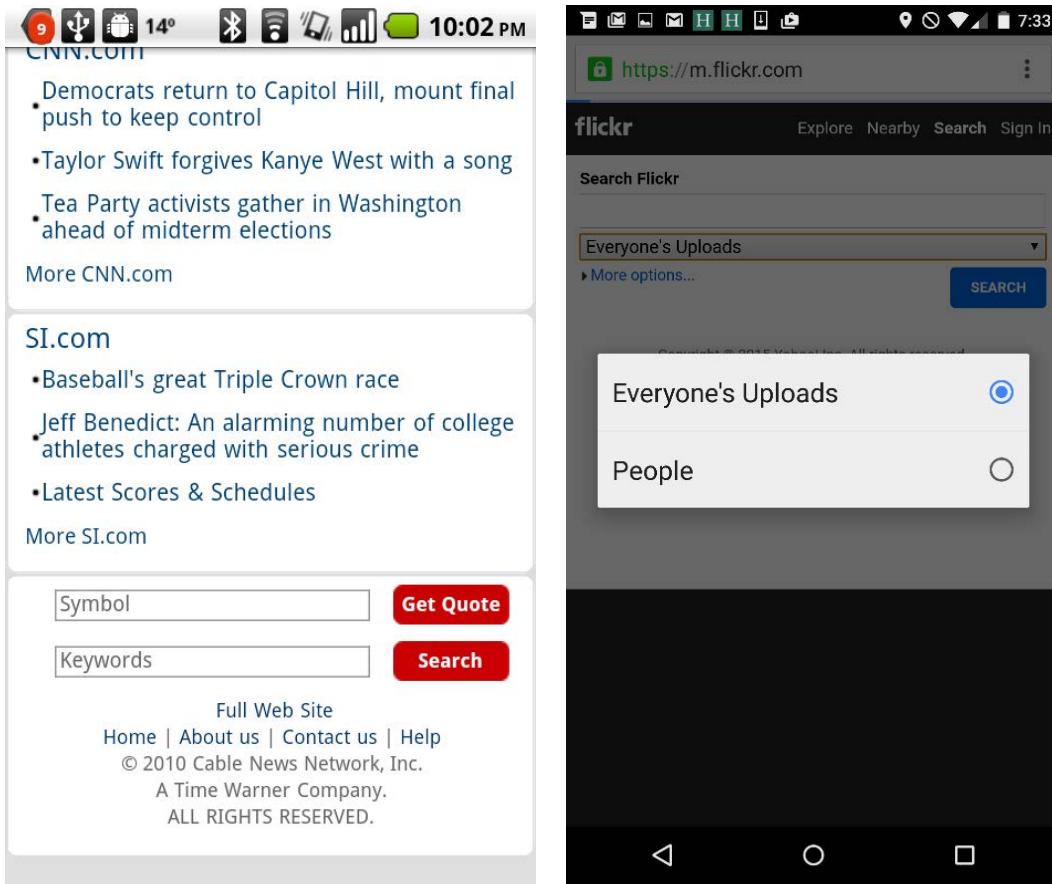
Amazon did not offer search suggestions based on history, even though the user was logged in.

235. Do not use several search boxes with different functionalities on the same page.

We often see mobile users become disoriented and lose sense of context. When there are multiple search boxes, each of them implementing a different type of search, users may confuse one with the other. First, they may not even see all the search boxes at a time, and second, they may not read carefully the descriptions of any of them.

In the example below, an older version of CNN Money used two different search boxes with different functionalities: one box was for searching stock quotes by ticker symbol, another box was for free-text searching within the site's articles. The multiple search boxes were confusing for the users. For instance, one of our users was searching for the trading value of Autodesk, and he entered the company name "Autodesk" in the search-by-symbol search box. He did not get any result (because the website wanted a symbol in that search box, rather than a company name, and was very inflexible about it), so he tried again searching within the search-by-keyword box at the bottom of the page. The new search resulted in a bunch of articles about Autodesk, which confused the user and made him skim through text in order to possibly find some indication of the stock value. Contrast that design with Flickr's, where people

could enter the search string in the same search box, and then use the dropdown to indicate the scope of the search.



An older version of CNN Money (left) used two different search boxes for different types of search: one for stocks by company symbol, and one for search within their articles. Users were confused by the purpose of each of the search boxes. In contrast, Flickr (right) used a single search box, but it allowed users to specify the scope of the search.

Scoped Search

Context often changes quickly on mobile, and people forget the setting in which they last visited an app or a website. Sometimes the site may attempt to save state for them and inadvertently limit the scope of the search to a previously set one. In other situations the scope of the search is automatically set to a most popular category that does not match the user's current needs.

In both these situations, users would be better off if the scope of the search was as general as possible, and the results were grouped by scope instead. This type of

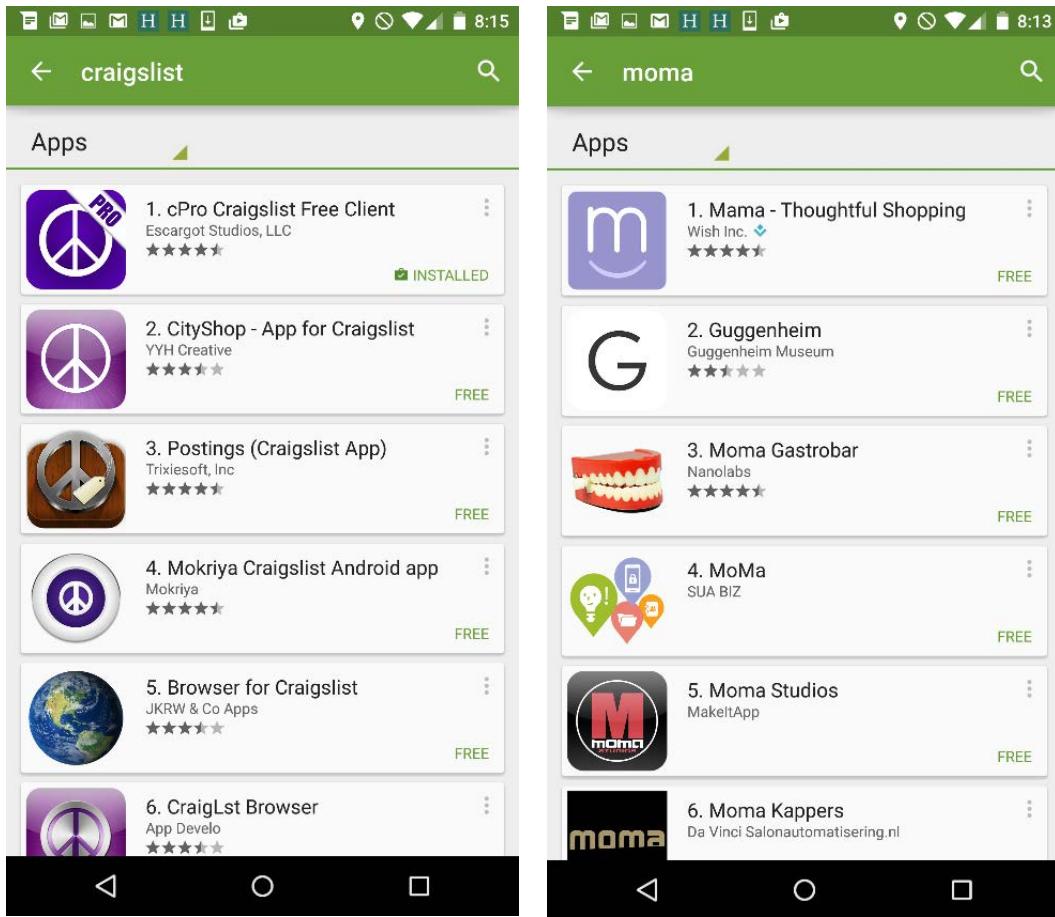
design minimized the burden on users: they don't need to think about scope or explicitly select it⁶⁰.

- 236. Do not limit the scope of the search based on the section of the app that the user is in.**
- 237. If there are multiple types of scopes available, group results by scope and show a few results from each scope.**

In a prior testing, one of our users had trouble finding free mulch on Craigslist. The reason was that prior to the session he had used the app to look for furniture, and thus the app remembered that scope and searched for "mulch" in the furniture category.

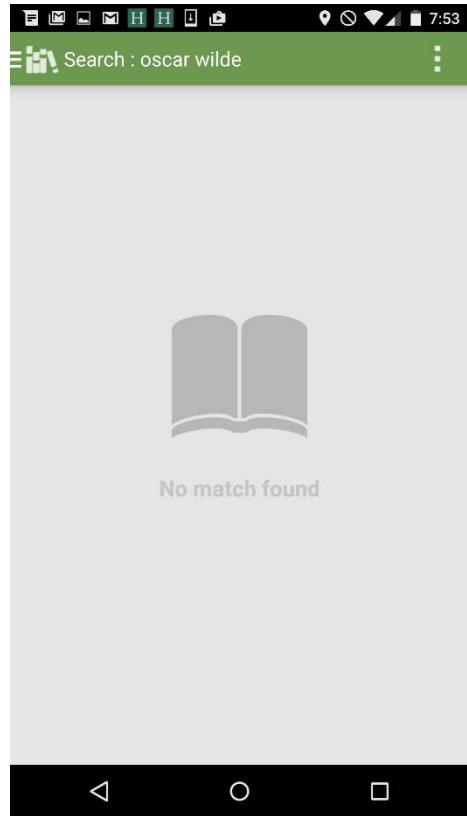
Similarly, if users happened to be in the *Apps* section of Google Play, a new search would automatically return results from that section. As a result, many users would think that there are no, say, music results that match the search query.

⁶⁰ For more information on scoped search see Katie Sherwin. "Scoped Search: Dangerous, but Sometimes Useful." <http://www.nngroup.com/articles/scoped-search/>



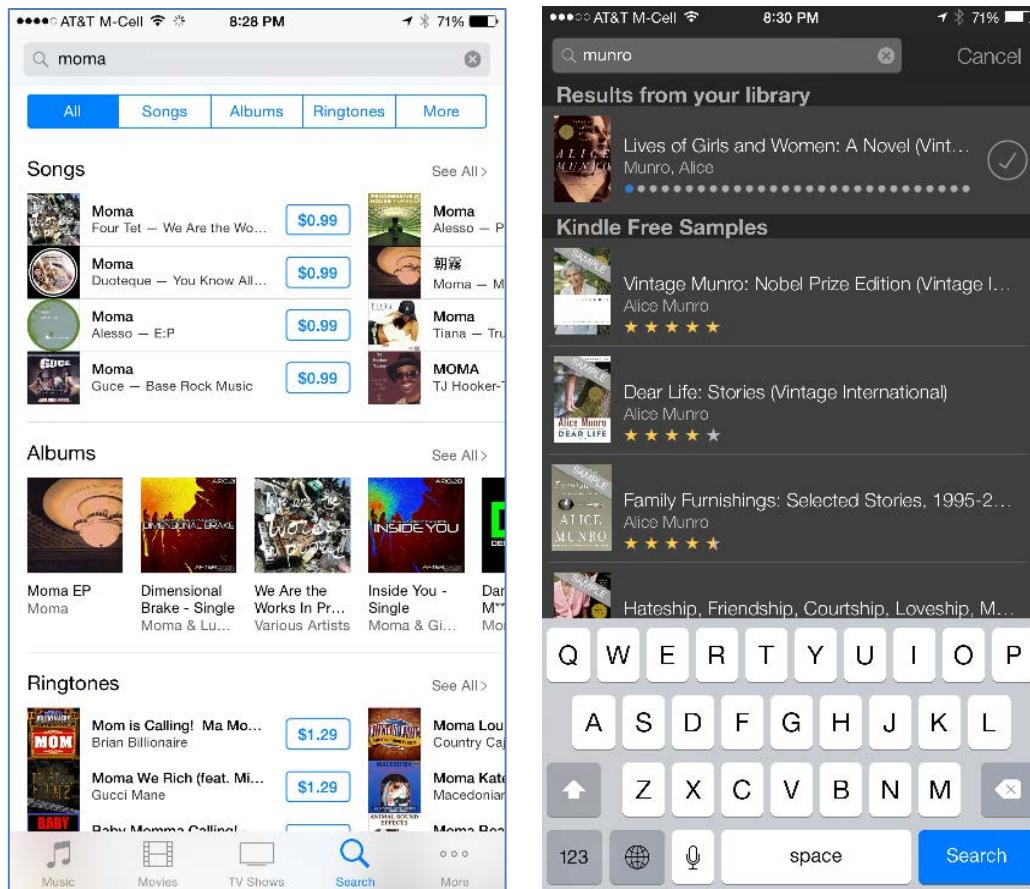
Google Play for Android: The user first searched for the Craigslist app; when she subsequently searched for the string “moma”, automatically only app results were displayed. These results could confuse the user into thinking that there were no other types of results (e.g., music, books).

In Aldiko, a book-reading app, searches are automatically scoped to the user’s library. Thus if users searched for “Oscar Wilde” chances are that their results will be null unless they happened to have had downloaded some of Wilde’s books. Thus, someone attempting to download books by Wilde could easily assume that there were no such books in Aldiko’s database.



Aldiko returned zero search results for “oscar wilde” because the scope of the search was by default set to the user’s library.

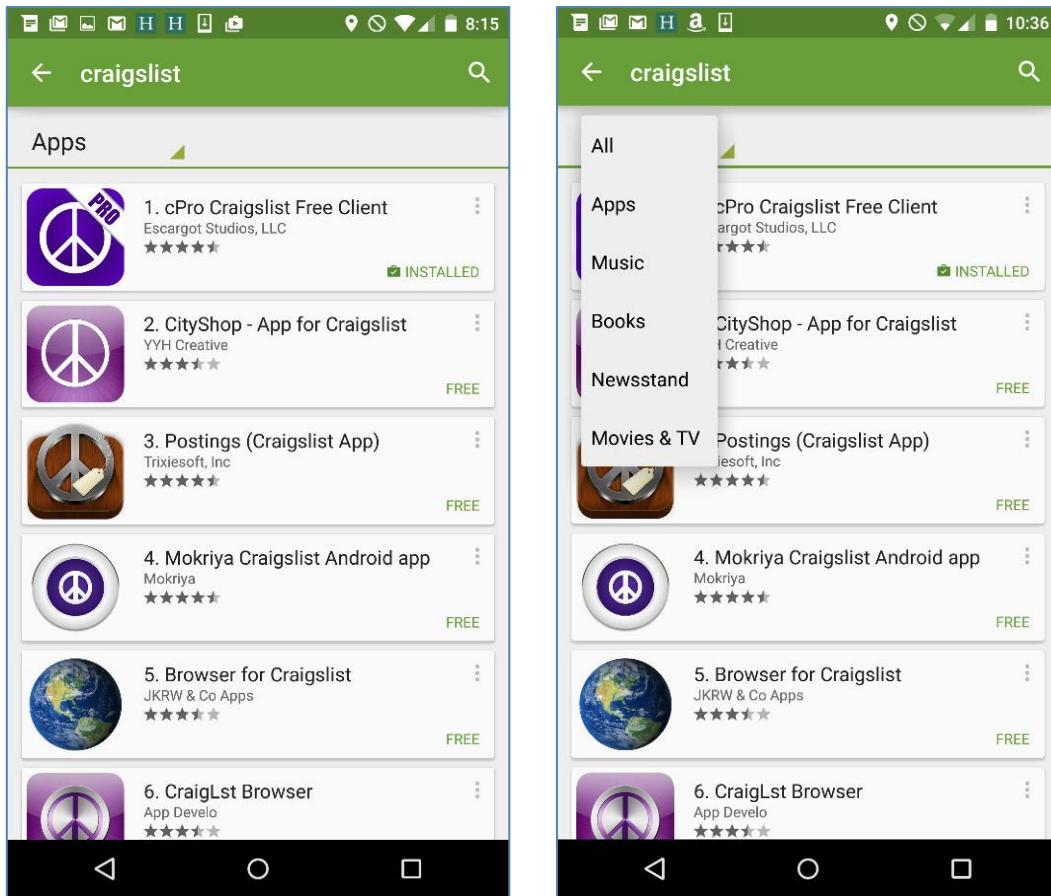
Instead of automatically assuming a certain scope for a search, it’s better to show a few results from each available scope and thus allow users to select the scope that is relevant to them, like in the iTunes and Kindle examples below.



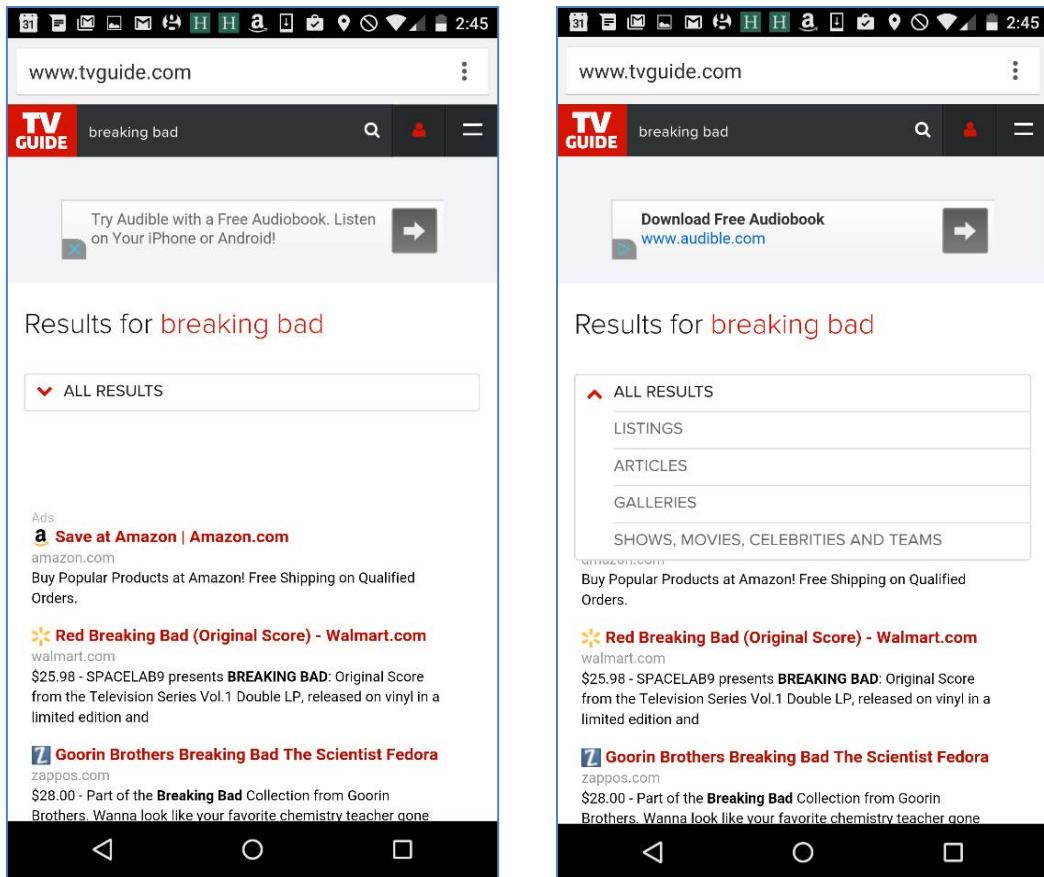
iTunes for iPhone (left) and Kindle for iPhone (right) both grouped search results into different scopes and showed results from each scope. iTunes used both tabs (at the top of the search results) and result groupings to indicate the different types of results. Note that tabs would not be enough for indicating different result categories, as users often ignore them (see guideline 238).

238. Do not use tabs or menus as the only method to indicate different categories of search results.

Often people ignore tabs and menus, so it's unwise to use them as the only way to change search scope. While tabs and menus can be used as an additional method to signal scope (as in the iTunes example under guideline 237), results should also be grouped according to scope (like iTunes does).



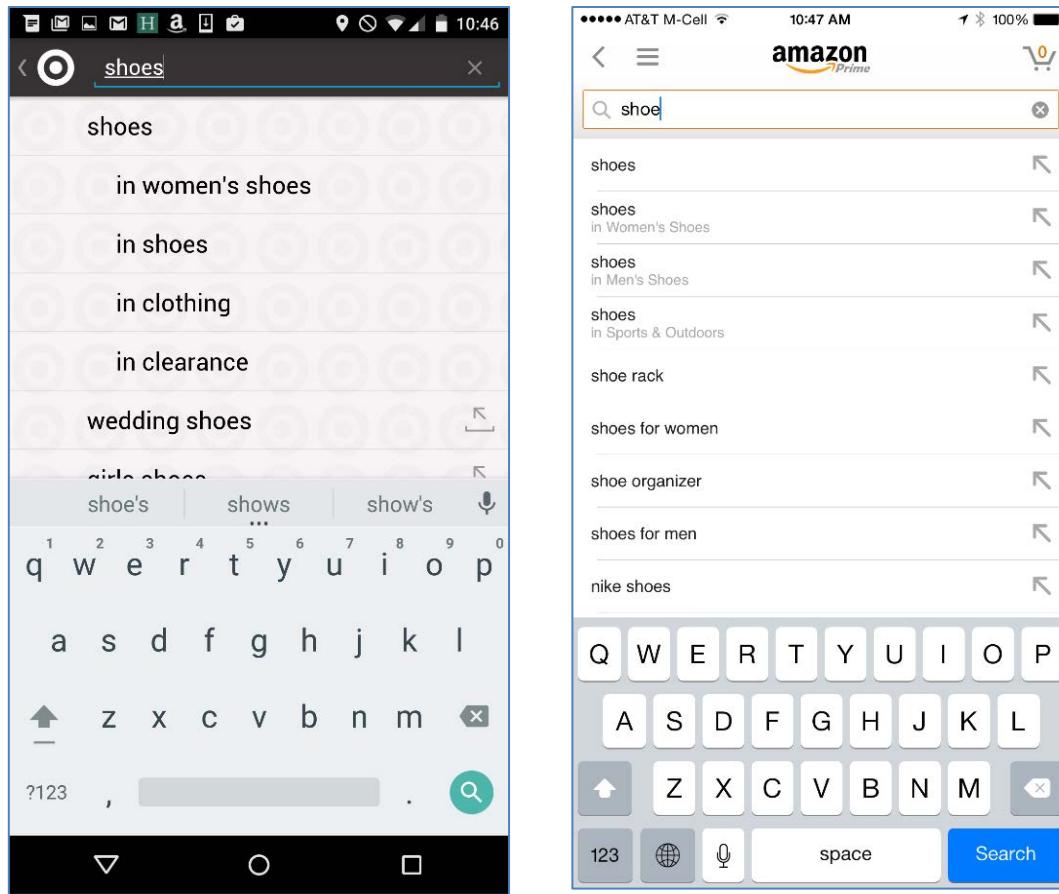
In Google Play for Android the results for the query “craigslist” were apps, but the scope could be changed using the drop-down labeled *Apps* at the top of the screen. Unfortunately that drop-down violated several of our menu guidelines, as it had little discoverability and a poor label (see guidelines 83 and 85).



TV Guide hid the scopes available for search under a drop-down (labeled with one of the options). That made the content types hard to discover. (Luckily, TV Guide did set the default scope to include all types of content.)

239. If scope autosuggestions are made, make sure that you allow users to make a selection that does not have a scope applied to it.

This guideline is valid for both mobile and desktop and it ensures that people are not forced within a scope against their will.

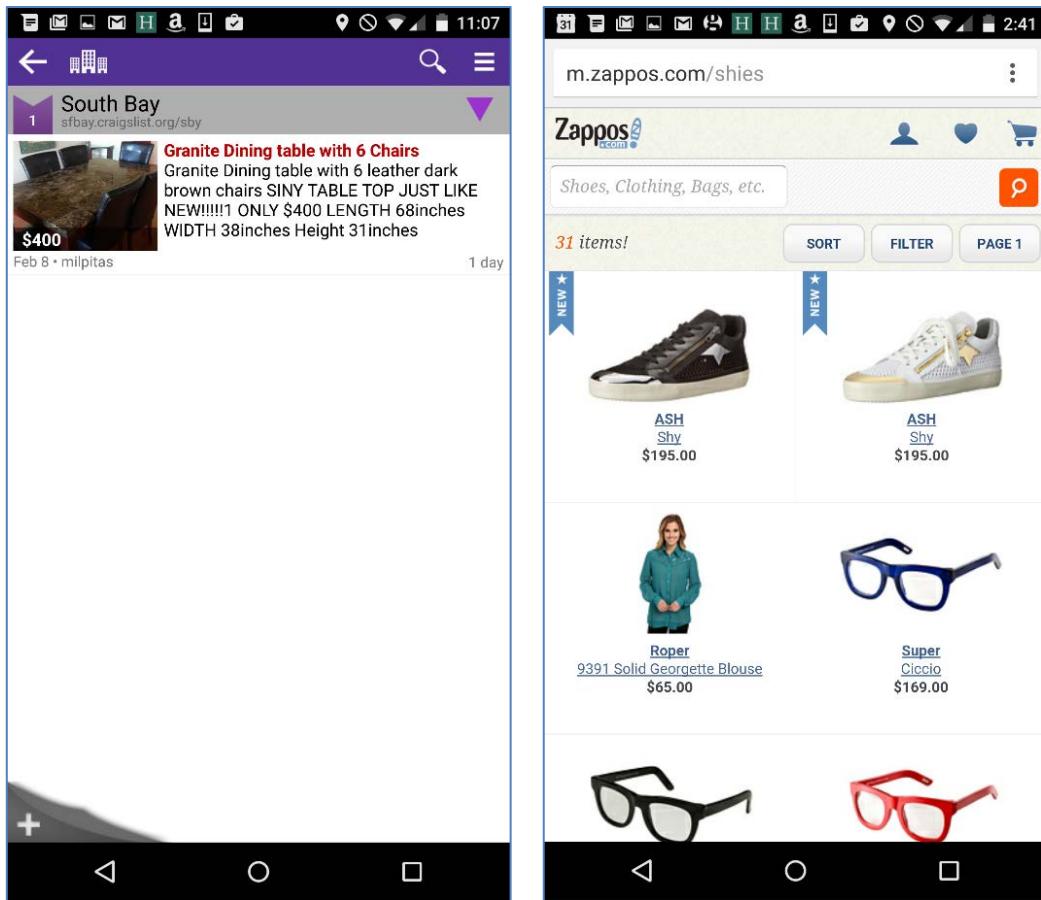


Both Target for Android (left) and Amazon for iPhone (right) provided scope suggestions to allow users to narrow down to a specific category. They both had a no-scope option as well in the list of autosuggestions.

Search Results

240. Always show the search query on the search-results page.

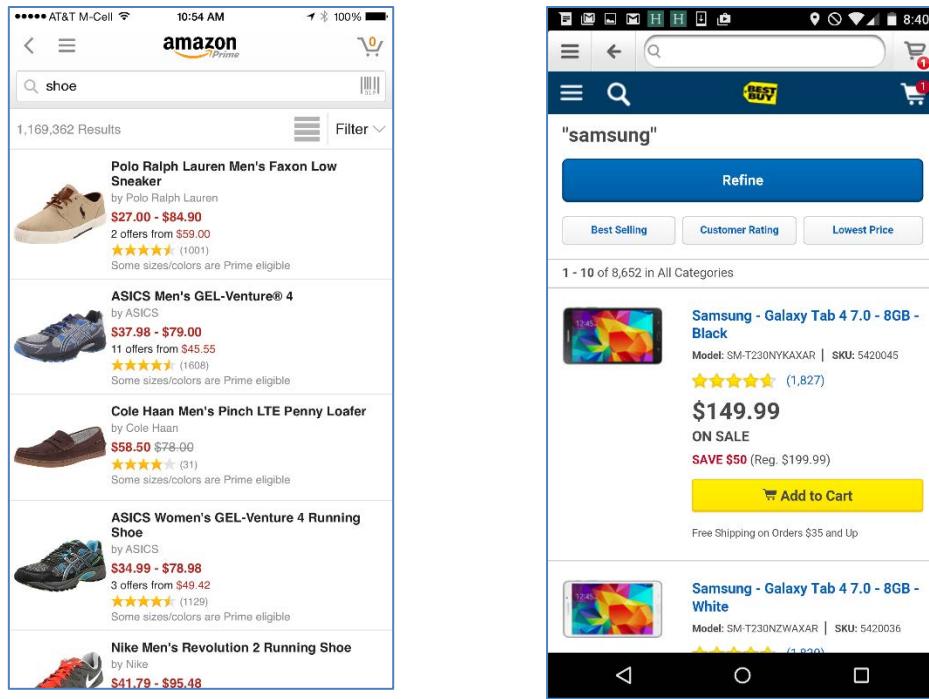
Sometimes users may have accidentally made a typo in their query without realizing it, and sometimes an interruption may have occurred that made them forget what they searched for. Always make sure that the search query is displayed on the search-results page.



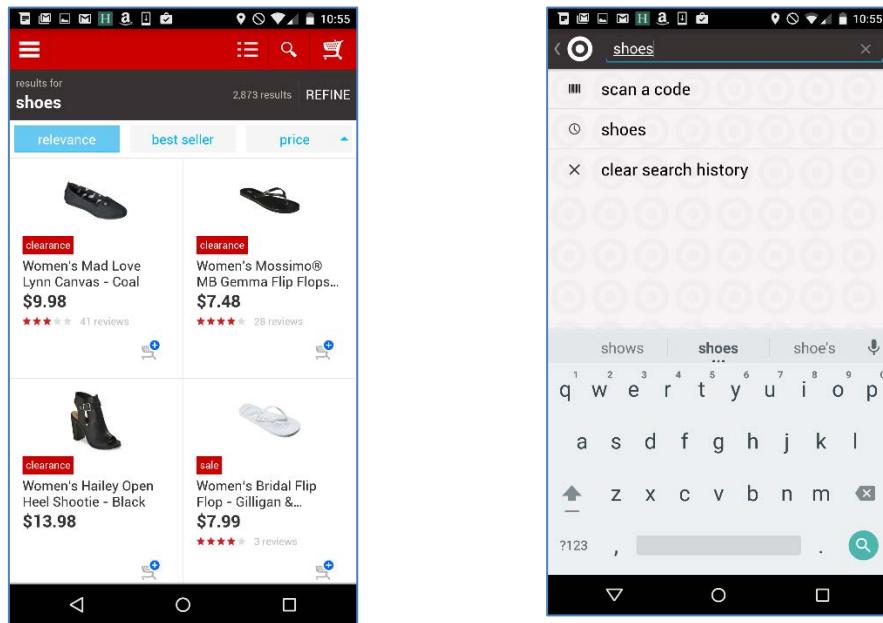
Craigslist for Android or Zappos for Android did not indicate the search query ("siny" in Craigslist's case and "shies" for Zappos). Craigslist did show the location filter applied (*South Bay*).

241. On the search-results page, preserve the search query in the search box.

Since people often edit their previous queries, it's important that the search box not be cleared and that it preserve the string that they typed. This string will serve two purposes: (1) it shows them what they searched for; (2) it saves them extra typing if they need to change or refine their search.



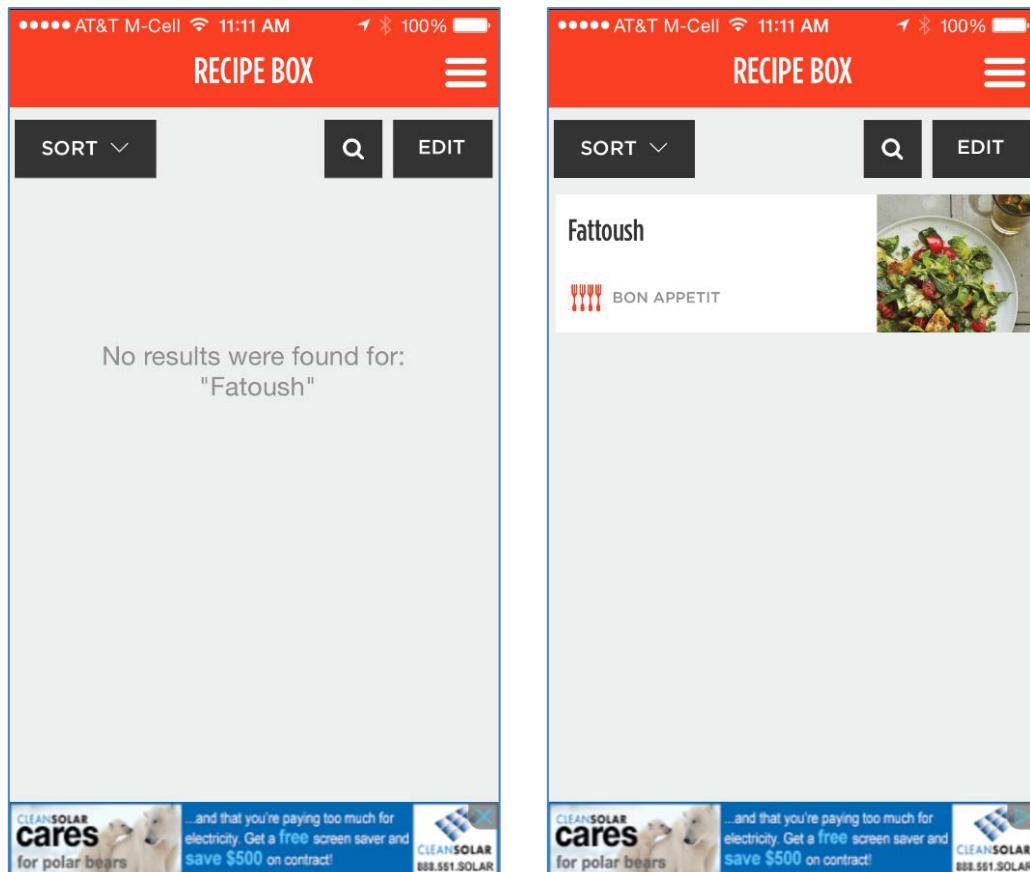
Both Amazon for iPhone (left) and Best Buy for Android (right) preserved the search query on the search-results page, but only Amazon displayed it in the search box.



Target for Android showed the search string on the results page, but did not display it within the search box for the simple reason that its search-results page only included a search icon. However, upon tapping the search icon, the last search query could be edited in the search box.

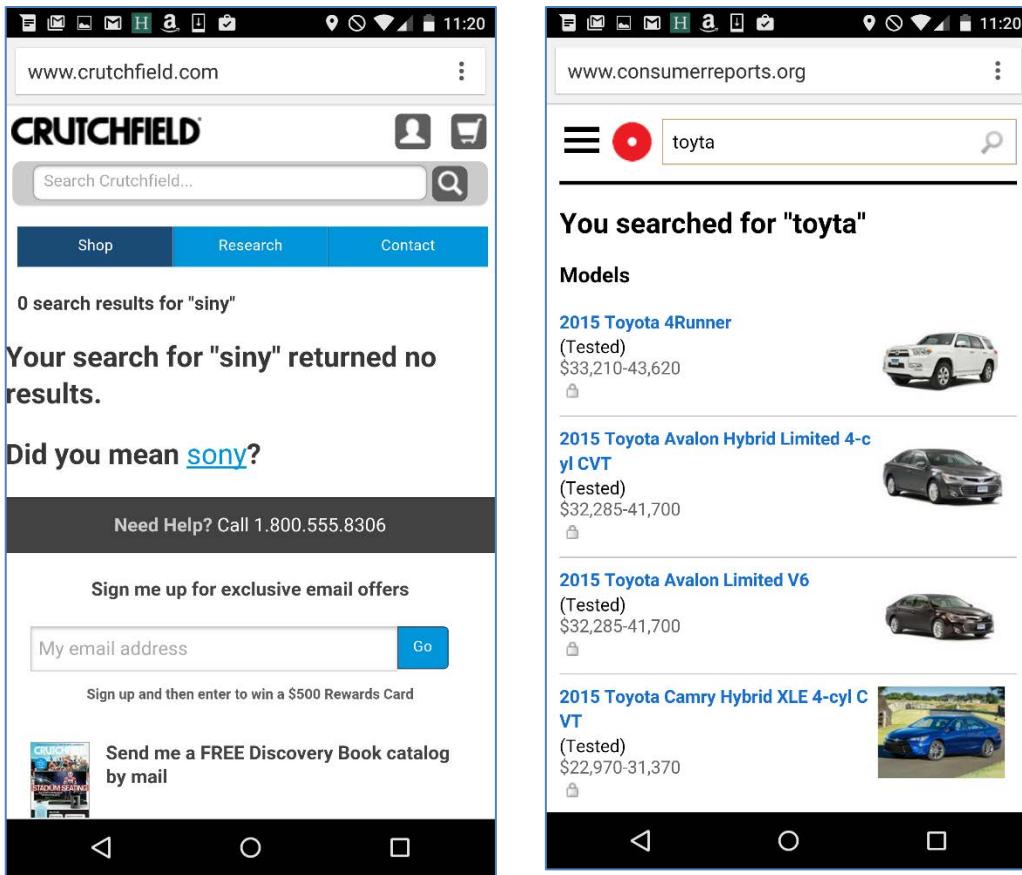
242. If there are no search results, tell people that there were zero results for their search and offer results for alternative searches.

When there are no results for a search, it is often because of a typo in the search query. Whereas users need to be informed of a search failure, the website can save them the interaction cost of revising the search by offering results to alternative searches that partially match the initial query. Moreover, if you have reasons to expect that the search results to the same query be different on your full site, offer a link to the search results on the full site (sometimes, perhaps the user is looking for one of those site features that you have decided not to support on mobile).



Epicurious for iPhone did not offer any alternative searches when the user misspelled the name of a dish (left screenshot), although the database did contain a close match (right screenshot). (Note also that the app did not show the search query in the right screenshot.)

If users are just shown alternative results without being told that their search had zero results, they may not realize that the results shown do not exactly match their query. (In many cases that may not be a problem, especially if a typo was the cause of the null results, but sometimes the users did mean what they had typed.)

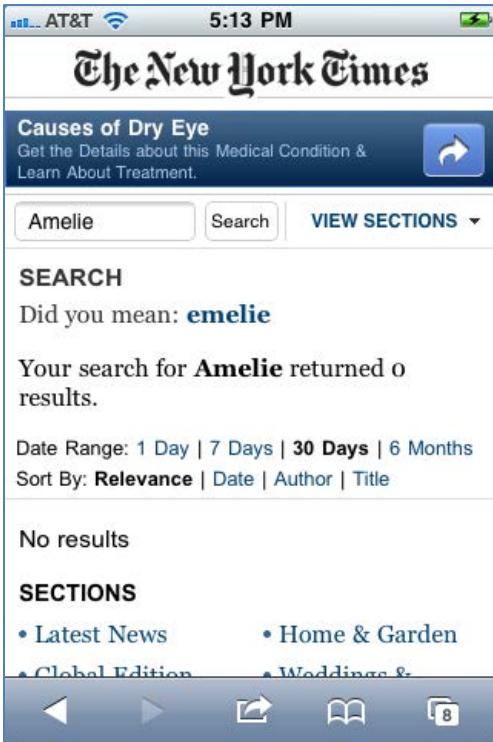


Left: A search for "siny" returned zero results on Crutchfield's website; this fact was clearly communicated and an alternate search was suggested (however, the site should have also presented results for the alternate search). Right: Consumer Reports did offer alternative results for a search for "toyta", but did not tell users that their original search returned no results.

One of our participants was attempting to search for a car using the Edmunds application. Unfortunately the feature combination that he chose led to no search results; in spite of his attempts to tweak the features, the search still returned no results, leading the user to believe that the app was not working.

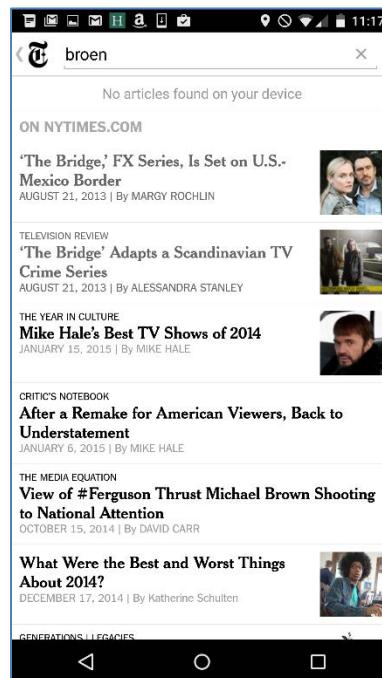
243. If there are no search results in your app or mobile website, run the query on your full site and return matching results if any.

On an older version of New York Times' mobile site, a search for "Amelie" returned zero results. Linking to the full-site search results would have been a better solution. And in fact a recent implementation of the app did exactly that: returned results from the full website when the app did not contain any.



The screenshot shows a desktop search results page for 'amelie' on The New York Times website. The header includes 'The New York Times' logo and a 'Search' button. Below the header, a search bar contains 'amelie'. Navigation links include 'Today', 'Past 7 Days', 'Past 30 Days', 'Past 12 Months', and 'All Results Since 1851'. Below these are 'All Result Types', 'Articles', and 'Multimedia'. On the right, it says '1-2 of 2 Results'. The first result is a book review titled 'Say Her Name and The Long Goodbye - Book Review' by Dwight Garner, dated April 13, 2011. Below the result is a note about 'TIME TOPICS: Raf Simons'. The footer includes a link to 'Advanced Search' and a note that search results are powered by Google Custom Search.

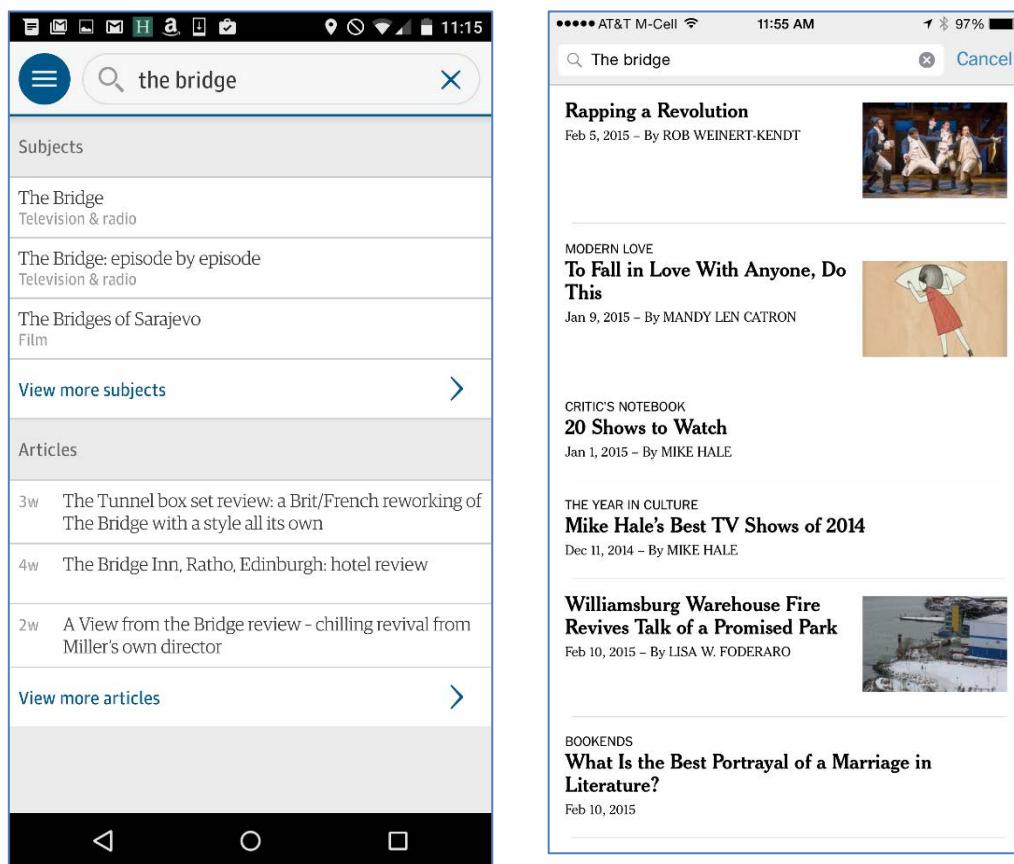
The search for “Amelie” on an older version of NYT’s mobile site (left) returned zero results. The site could have linked to the results from the full site (right).



NY Times for Android did show results from the full site when the app did not contain any matches to the user’s query.

244. Whenever the results belong to different categories, group them and display a few results from each group.

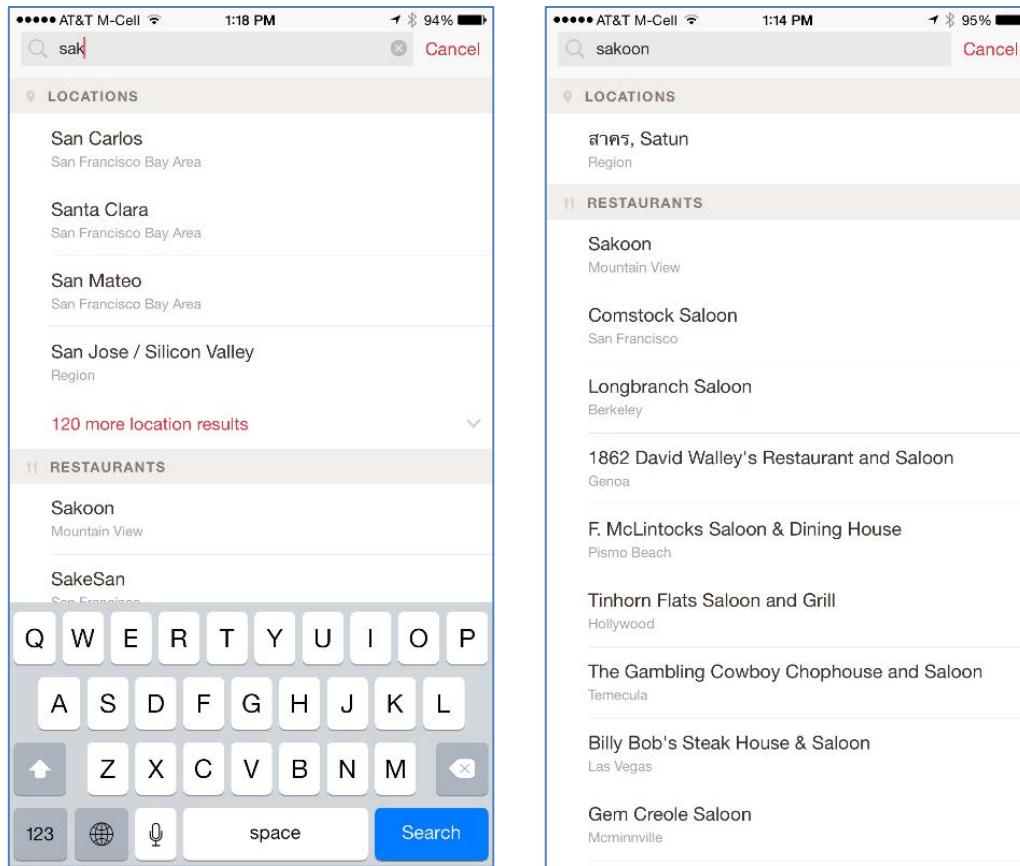
This is a reformulation of the scoped-search guideline 237. The results from a query in The Guardian offer a clear structure of the search space and allow users to select the type of result that best matches their need. In contrast, The New York Times search results are not structured at all.



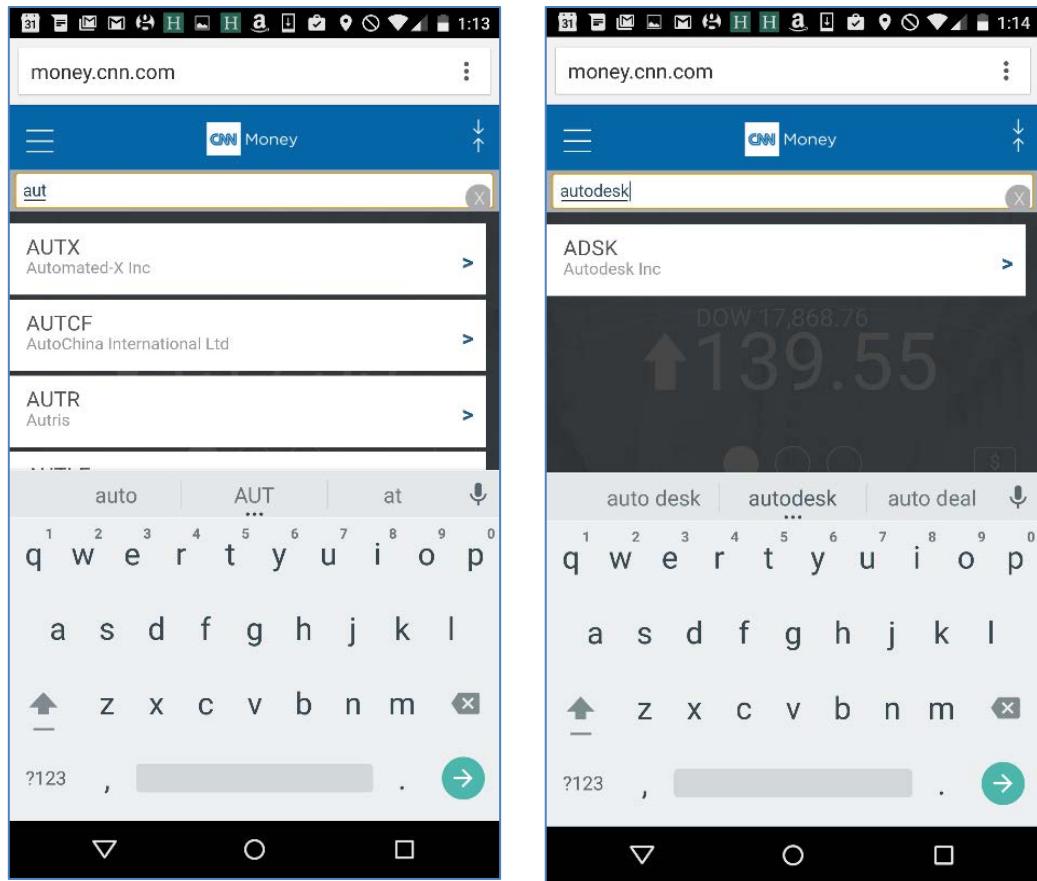
The results from same query in The Guardian for Android (left) and New York Times for iPhone (right) show the benefits of groupings for results. The structure of the search space was a lot clearer in The Guardian's design.

245. Differentiate between search results and search suggestions.

It's important that people recognize search results when they see them. One of our study participants was trying to find a restaurant using Open Table. She typed the name of the restaurant (ignoring all search suggestions) and then she repeatedly hit search to no avail. The app was expecting her to select the name of the restaurant from the list of results (which looked exactly like suggestions), but in fact she had felt no need to do that additional gesture since she had typed the restaurant name already.



Open Table for iPhone: The list of suggestions (left) looked the same as the list of results, so the user repeatedly hit Search on the keyboard without realizing that the results were already in front of her.

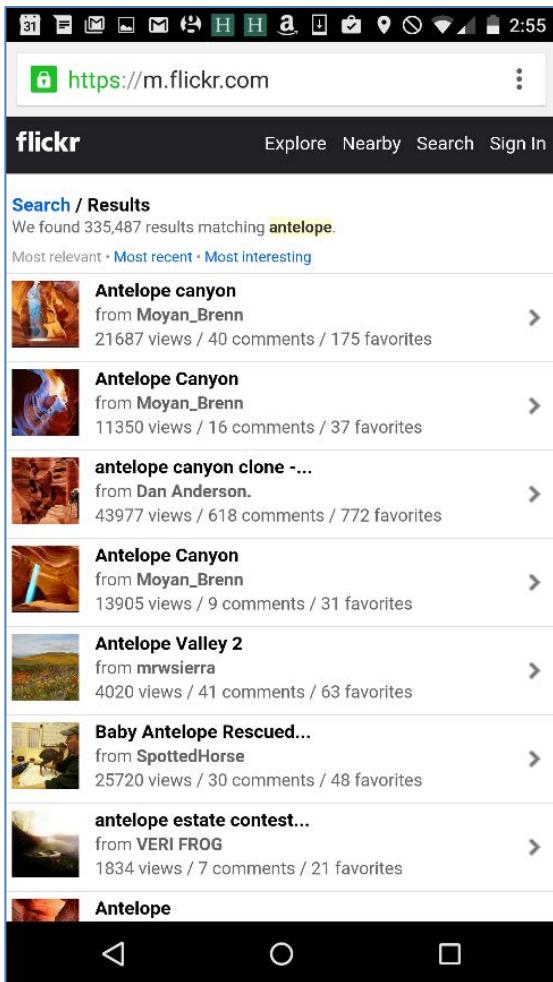


CNN Money's list of results was identical with the autosuggestions.

Filtering and Sorting

- 246. Provide filters. If a list contains items that belong to different categories, provide filters to narrow down the number of elements.**

Filtering limits the number of items that users need to inspect to a manageable subset. It is especially important on mobile since mobile hierarchies tend to be flatter than desktop hierarchies (see guideline 208), and thus lists may be in reality longer than on the desktop.



Although it did offer some sorting options, Flickr provided no filters for narrowing down the long list of results. At the very least it should have provided filters for the type of content (e.g., video or photo).

Our TVGuide users would have benefited if they had been able to filter through the listings and select the movies. One participant spent about 7 minutes searching for a movie starting at 8pm so he could watch it with a friend:

"These are, I know, not movies. Since I didn't see a way to filter it, this is I guess what I have to do... [go through every listing]. See if I can go through help. I hope to find some more filtering... [After reading the help] The help was worthless, so I guess I'll continue ...[He goes through more listings.] This seems like a real pain ... I'm still in the 8th hour. To have to go through all this, and then to have to hit 9 [hours] and 10 [hours] – I'd just tell her 'don't come'..."

On an older version of Schuh's website (a UK shoe store), it was possible to see a list of products on sale, but not to filter them by size. Users were frustrated when they clicked through several pairs of shoes, only to discover that their size was not available.

A UK user described the problem with the Schuh website:

"I'm going to have to go into each shoe to see if they have my size... It'd be nicer to filter by size. A lot of other online companies do that. I'd give up..."

A user was trying to find out if a tennis player had been eliminated from a tournament. The ESPN website allowed filtering by tournament day, but not by player:

"What was interesting about that — I think it should have been a search part where you could have typed her name in there, [...] rather than having to scroll through everything, 'cause, you see, here where it says *Women's Singles* I actually had to use common sense and go back [through dates — January] 26, 25, 24, and finally found it; there may have been a quicker way, but I couldn't find it at all — there should be a search bar, type in her name and it will take you directly to this."

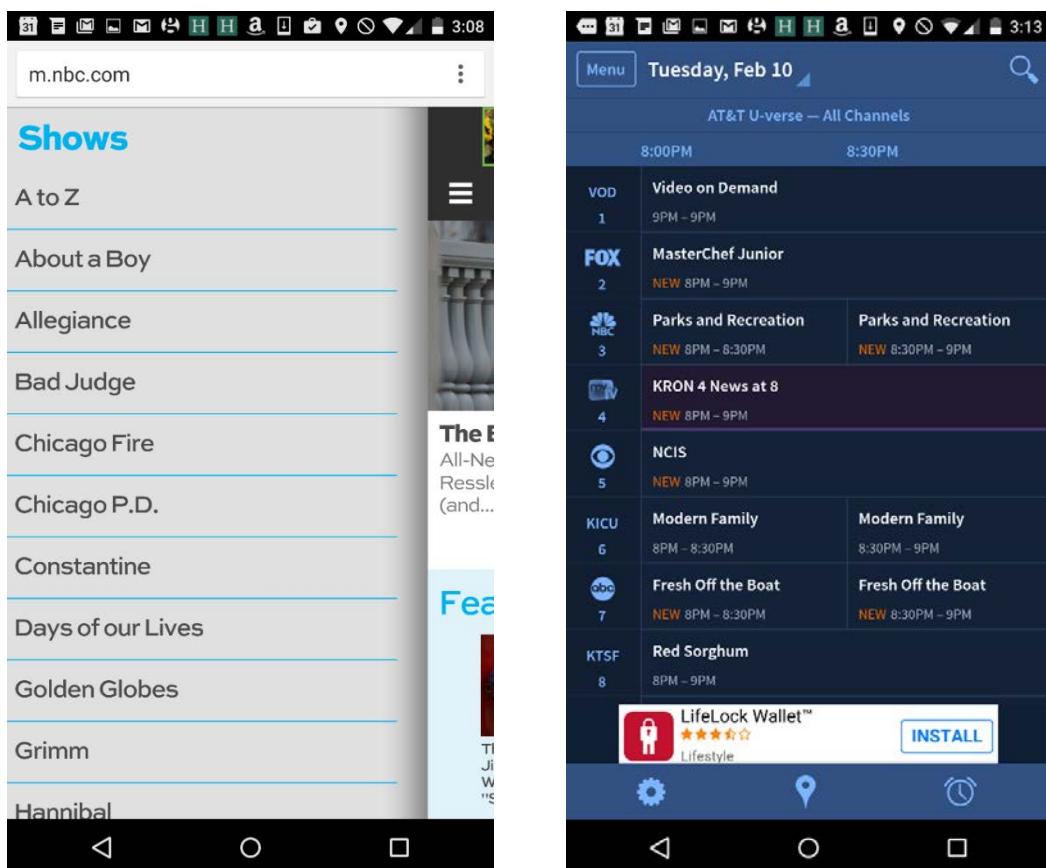
The screenshot shows the ESPN Tennis website for the 2011 US Open. At the top, there's a navigation bar with tabs for Daily Results, Schedule, News, and Rankings. Below this is a date selector showing SEP 2, SEP 3, and SEP 4. A red box highlights this area. Further down, a blue box highlights the "Women's Singles" category under the "2011 US Open" section. The main content area displays the "FULL TOURNAMENT RESULTS" for the "Third Round: Women's Singles". Several matches are listed with their start times and opponents.

Match	Start Time	Opponents
1	11:00 AM ET	(1) C Wozniacki v. V King
2	TBD	(4) V Azarenka v. (28) S Williams
3	11:00 AM ET	S Soler Espinosa v. C Suarez-Navarro
4	11:00 AM ET	[Match details partially visible]

On ESPN.com the user needed to check all tournament dates to find information about a particular player on ESPN's website. There was no way to filter by player.

247. If a list of items can be sorted according to different criteria, provide the option to sort that list according to all those criteria.

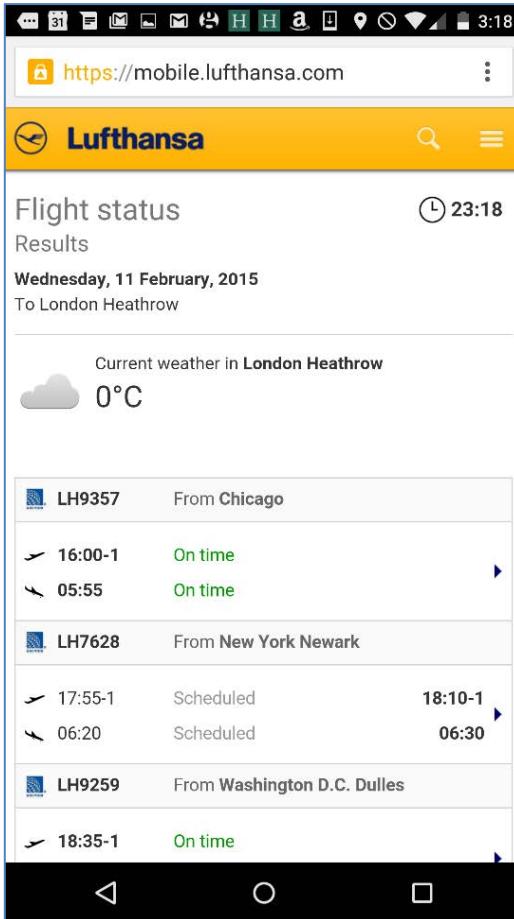
The example below shows the mobile site of NBC. To find a particular show, users had to tap a hamburger menu and scroll down a list of shows ordered alphabetically. To find a show such as "Parenthood", users could easily scroll down to the letter "P" and locate it. Compare this list with the one on TVGuide. In this case, our users were looking for a movie that started at 8pm. There was no way for them to know which of these listings was a movie, so they had to scroll down the list (which was ordered according to channel) and guess whether any of the particular entries was a movie. The task was highly frustrating for the users. The reason the TVGuide list did not work well was because it was not sorted in an order that matched the user's task (instead, it was sorted by hour). Note however, that, if the tasks were different (e.g., find a movie on the NBC website and find what's on channel 5 at 8pm) the results could have been reversed, since the order on the NBC website did not indicate the type of show, and the TV guide was, indeed, sorted by channel.



Left: Alphabetically sorted shows worked well for users looking for the show "Parenthood", but might not work for users looking for shows in a particular category (e.g., comedy). Right: The TVGuide contained a list of programs sorted according to channel; this sorting isn't natural if you are looking for a particular show, but may work for someone who wants to know what's playing on a given channel.

These examples show that, if there are many ways to sort the same list, for certain users and tasks the list will be sorted in the wrong way. That is why it's really important to provide the user with the capability to sort the list after different criteria.

One of our users wanted to find out whether a flight from Munich to London was on time, but she did not have the number of the flight or the exact time. She looked through the list of arrivals at London Heathrow Airport, but the list was sorted by scheduled time of arrival rather than by name of the origin airport. Unfortunately, there was no way to sort the list differently, so she had to go through every flight to find the city she was interested in.

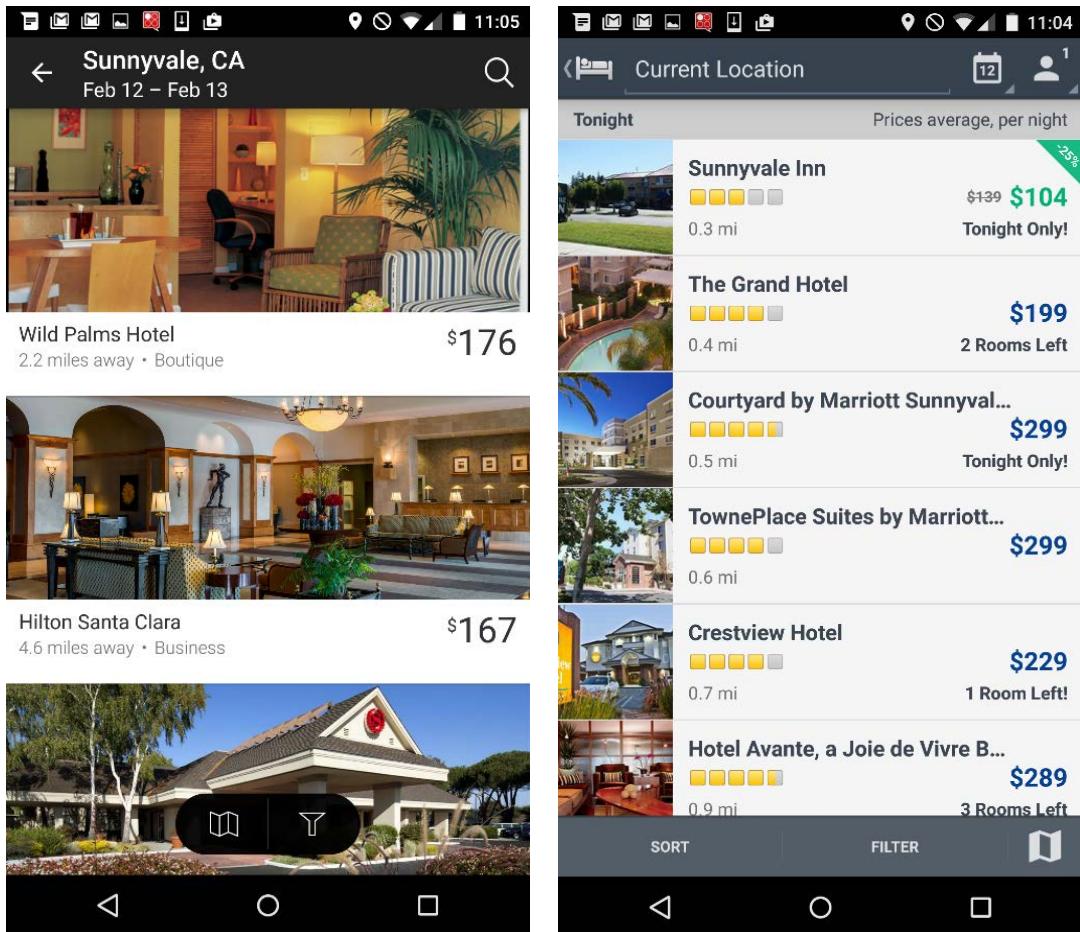


Lufthansa did not allow sorting the list of arrivals according to the departure airport.

Sometimes sorting does not solve the problem. Especially when the results are numerous, it is essential to provide a mechanism for filtering them.

248. Place sort/filter controls at the top of the list.

People are used to finding filters and sort controls at the top of the search-results list. Anywhere else on the page these controls are less discoverable.



Both Kayak (left) and Expedia (right) for Android used nonstandard locations for their filter placement. Kayak's was especially problematic because it was on top of a picture.

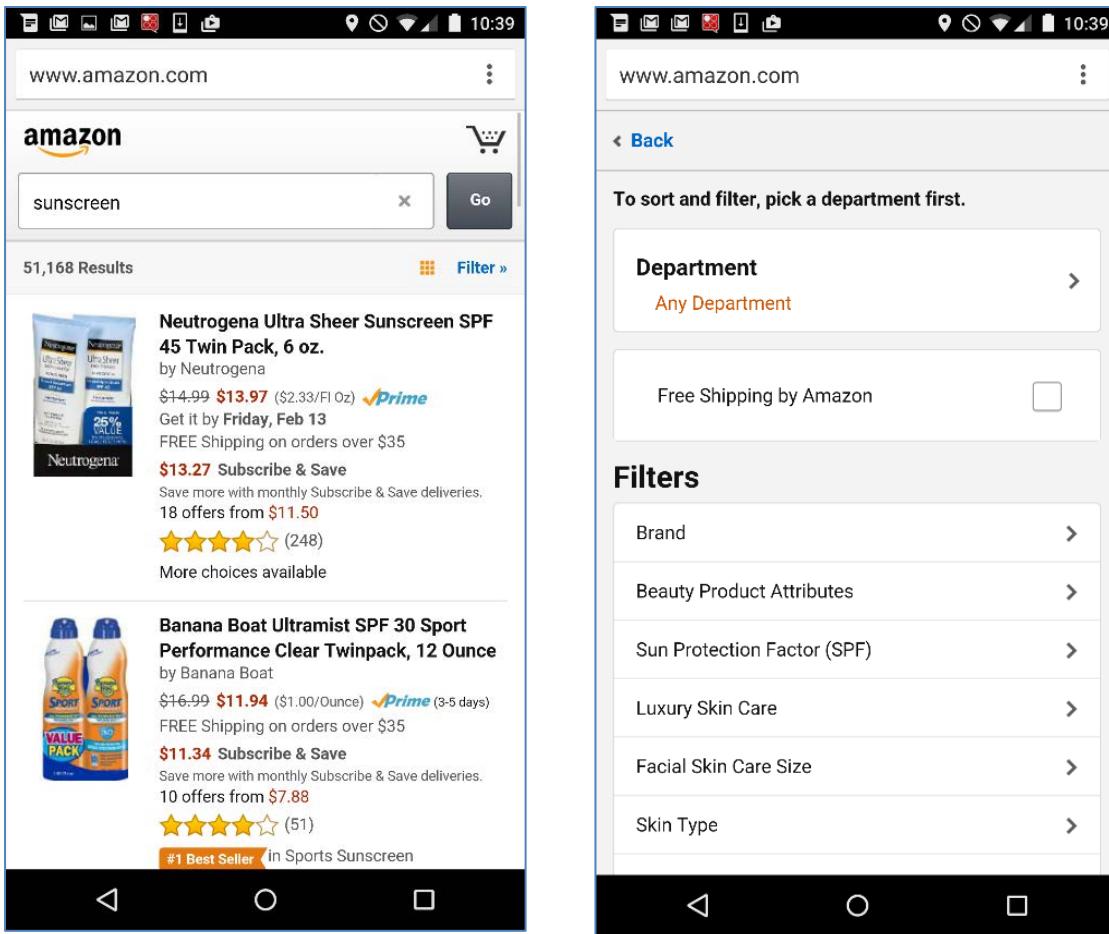
249. Minimize the space allocated to filters above list.

Filters are chrome, and one of the problems with having a lot of filter and sort options at the top of the screen is that they take up precious screen space: instead of seeing the list items, users see the filtering options. The placement at the top of the list can thus conflict with the recommendation to prioritize content over chrome on mobile (see the *Small Screen* section). Thus, we recommend that you place filter and sort options at the top of the list (because that is where users expect them), but allocate them as little space as possible.

If extended space is needed for the sort/filter options, use either an expandable menu or a link to a filtering page. On a website, to avoid extra load times, consider using top-placed jump links to a filtering area at the bottom of the page (like in an older version of Amazon example below), or, alternatively, using JavaScript to present the filtering area without resorting to a new page load (like in the newer version of Amazon).

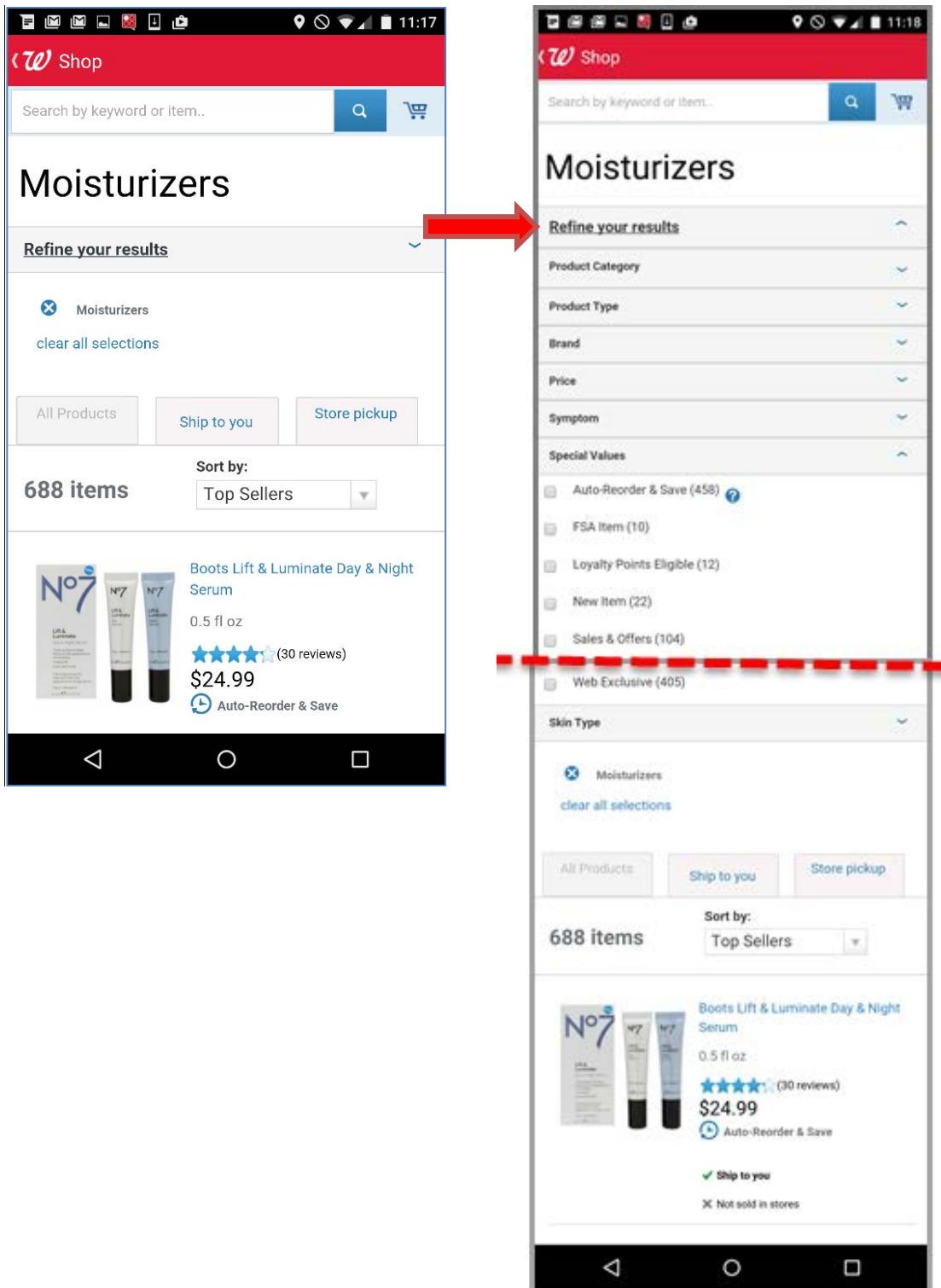
The image consists of two side-by-side screenshots of an old Amazon mobile interface. Both screens show a search for 'Arthur' in the Books category. The left screen displays the main search results with a red box highlighting the 'Filter Results' link just above the results list. The right screen shows the same search results with the 'Filter Your Results' section expanded, and the 'Filter Results' link is now a large, prominent button at the top of that section. A red arrow points from the 'Filter Results' link on the left to the 'Filter Your Results' section on the right.

On an older version of Amazon, jump links were used to take users to the filtering section, at the bottom of the same page. Note that the link *Filter Results* was placed just above the list of results (in a position that is salient, as well as consistent with the expectations created by the full web).



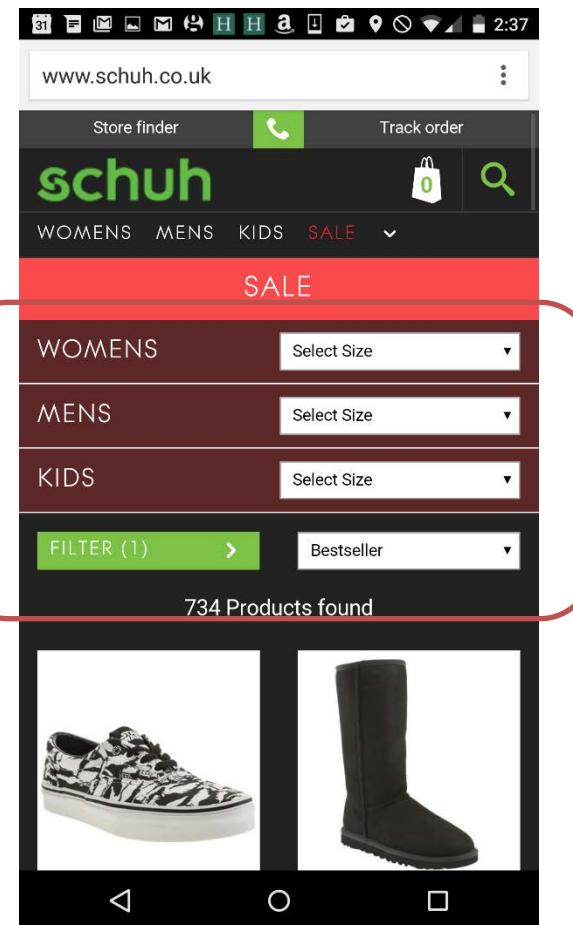
A newer version of Amazon used JavaScript to show the filter area without requiring a new page load. As in the older version, the *Filter* button took up a minimal amount of space above the list of results.

Another solution that saves up space is to use menus or accordions for the filtering/sorting options. This solution is exemplified by Walgreens:



Walgreens used an accordion for the filters in *Refine your results*. (Note however that it wasted a lot of space on the page for the various chrome elements.)

Schuh.co.uk chose to expose filters right away, on top of the search results. Unfortunately, that was not a good choice, since the filters occupied precious real estate (whether users needed them or not), stealing away space from the results.



Schuh.co.uk wasted too much space on the page by exposing some of the filters.

250. Indicate the selected filters/sorting criteria, but do not waste a lot of screen space with them.

When we discussed search, we recommended that the search query be always shown on the search-results page, so people know what they searched for (see guidelines 240 and 241). Similarly, when users have applied filters to a list, they should be able to see the filter values. Otherwise, they cannot be sure if the system interpreted their input correctly and did indeed run a new query based on their criteria.

The image displays two side-by-side screenshots of mobile web browsers. Both show search results for 'moisturizer'.

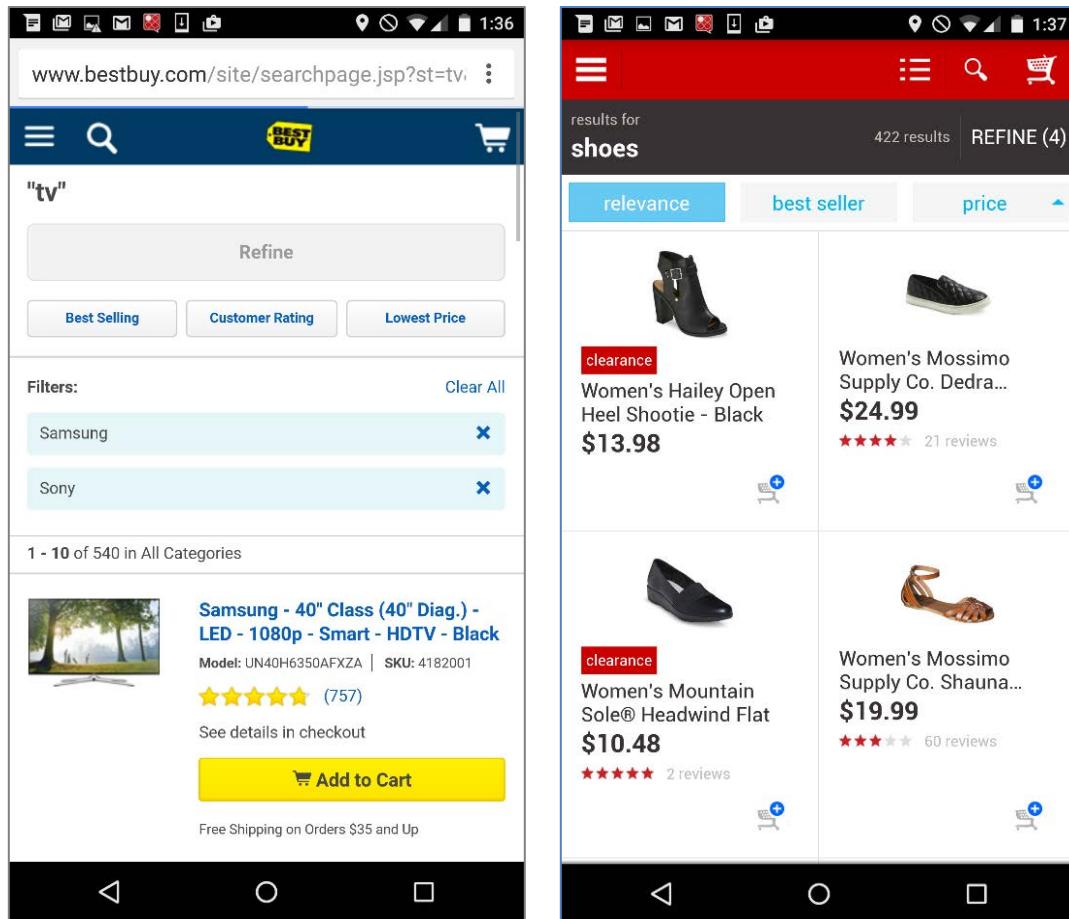
Walmart (Left):

- The search bar at the top contains 'mobile.walmart.com'.
- The search results page shows five products:
 - Vaseline Spray & Go Total Moisture Moisturizer, 6.5 oz**: \$5.97, 4.5 stars (428 reviews)
 - Neutrogena Daily Moisturizer, SPF 30 Visibly Even, 1.7 fl oz**: \$13.97, 4.5 stars (119 reviews)
 - Vaseline Total Moisture Cocoa Radiant Rich Feeling Lotion, 10 fl oz**: \$3.48, 4.5 stars (3 reviews)
 - Olay Total Effects Daily Fragrance Free Moisturizer, 1.7 fl oz**: \$17.94, 4.5 stars (39 reviews)
 - A partially visible product at the bottom starts with '\$1.97'.
- At the bottom of the screen are standard Android navigation icons: back, home, and recent apps.

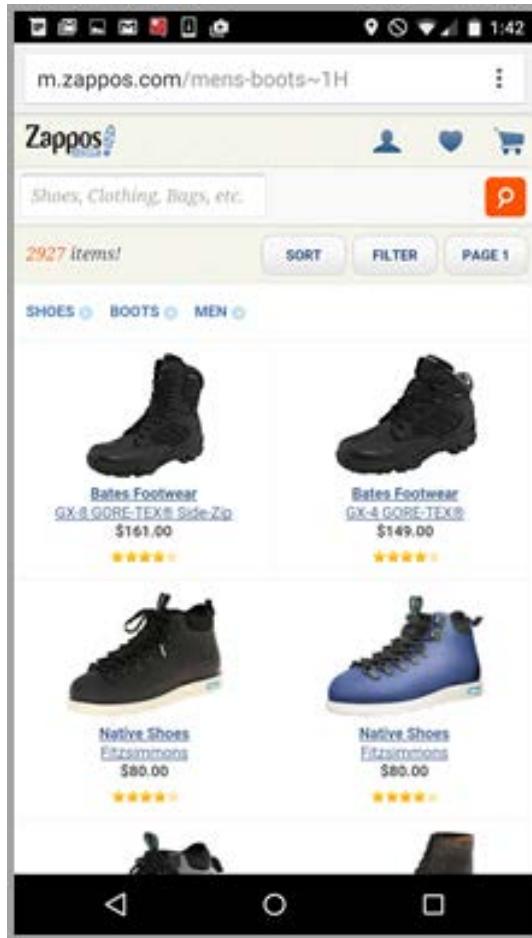
Amazon (Right):

- The search bar at the top contains 'www.amazon.com'.
- The search results page shows two products:
 - Jack Black Line Smoother Oil-Free Moisturizer 8% Glycolic Acid Treatment, 3.3 fl...**: \$30.00 (Prime), 4.5 stars (25 reviews)
 - DDF Daily Protective Moisturizer SPF 15, 1.7 oz.**: \$40.00 (Prime), 4.5 stars (3 reviews)
- On the left side of the Amazon results page, there is a promotional banner for 'Skin Care Sale: Extra 20% Off'.
- At the bottom of the screen are standard Android navigation icons: back, home, and recent apps.

Walmart (left) did not display any of the filtering criteria applied to the list of results (nor did it show the search query). Amazon (right) did display the filters applied as breadcrumbs, but unfortunately they were truncated.



BestBuy (left) and Target for Android (right) did indicate that filters had been applied. Best Buy listed the filters on the page (but unfortunately took more screen space than necessary), while Target only showed that 4 filters were applied by adding the number 4 after the word *Refine*. Users could see the filters applied by tapping the *Refine (4)* button. Target's signaling was too subtle.



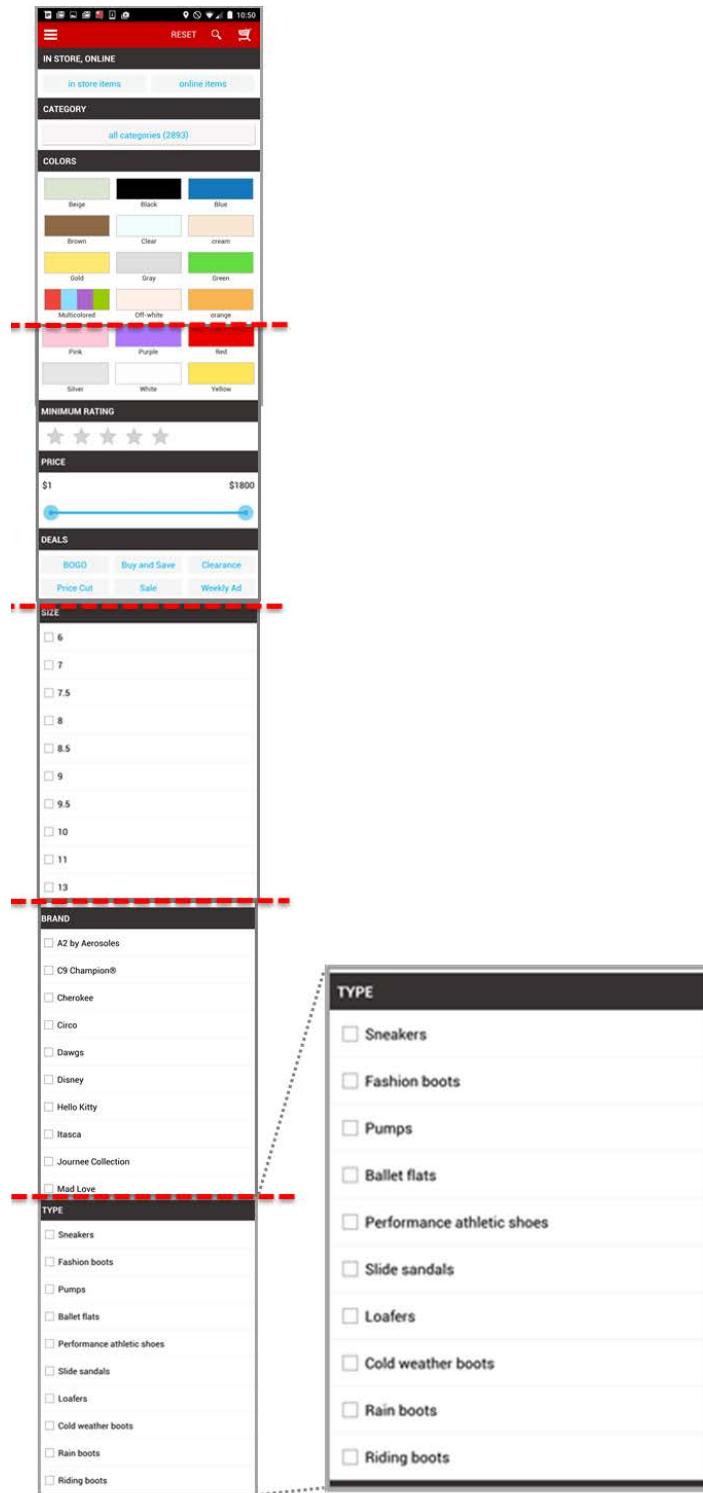
Zappos clearly indicated the filters applied to the list of products while not wasting a lot of screen space.

- 251. Let users quickly navigate to any single filter in your list of filters.**
- 252. Consider using menus or accordions for hiding the possible values for a single filtering criterion.**
- 253. Make sure that the filter menus/accordions are closed by default.**

All these three guidelines ensure that users can quickly access their filter of interest, whether it's first in the list of filters or not.

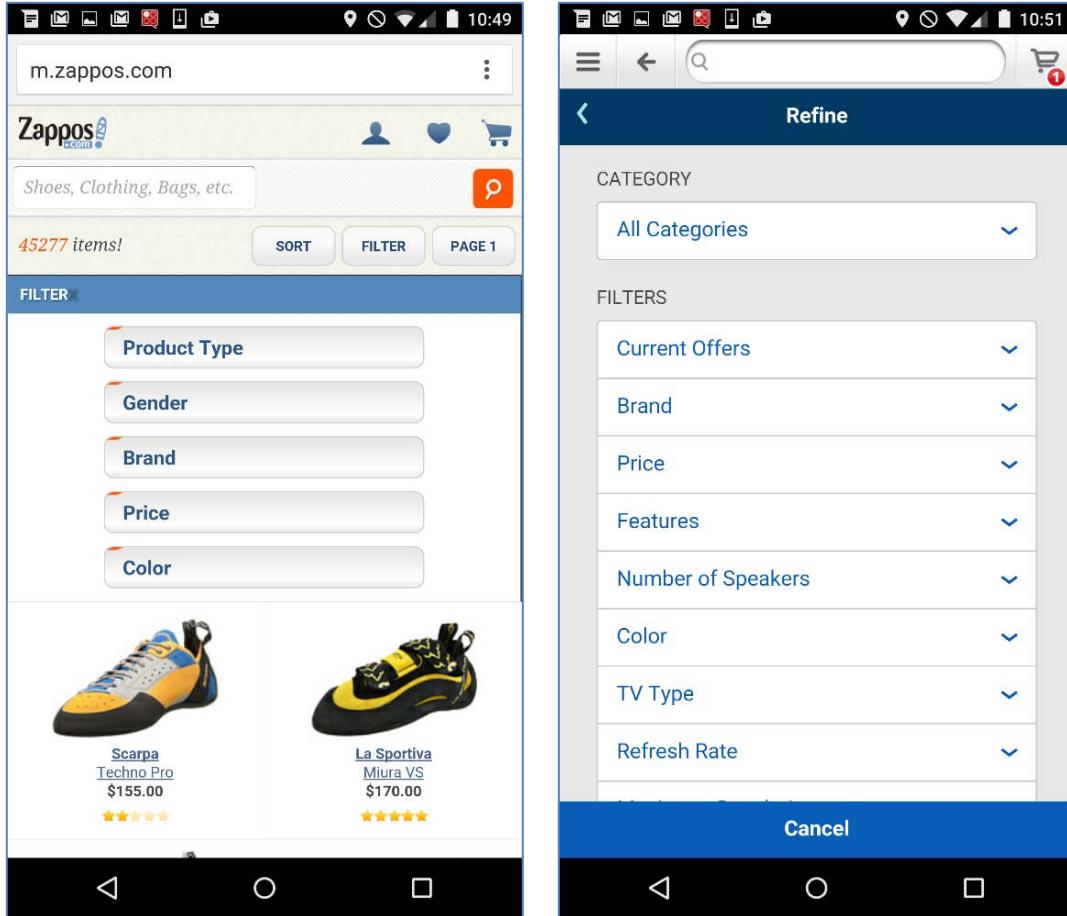
If all the values for all the different filters were explicitly listed on the page, then the user would need to scroll down too much to get to a filter of interest. It's best to display possible values for a filter only after the user has explicitly expressed interest in that filter.

In the example below, Target showed all the possible filter values for all filters on a single page. Thus, if users were interested in selecting a type of shoe (the last criterion listed), they had to scroll down through at least 4 screenfuls of other filter values to get to *Type*.



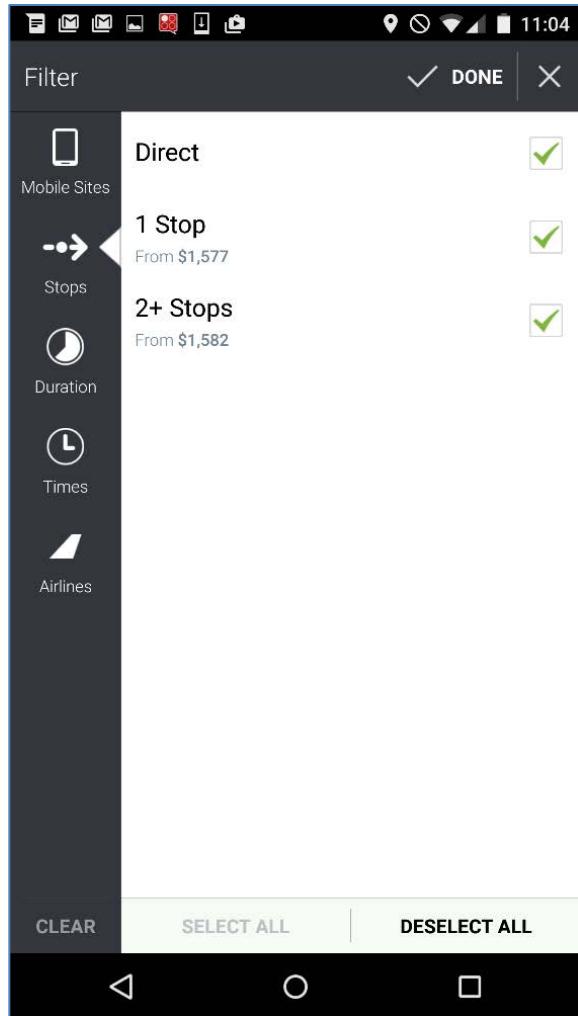
Target for Android: All the filters were expanded and the filter values listed on a single page. Users had to scroll down through many values to get to the criterion of interest.

In contrast, Zappos and Best Buy both listed their filters and expanded the list of filter values only after the user had selected a specific filter.



Both Zappos.com (left) and Best Buy for Android (right) used closed accordions for their filters. This choice enabled users to see all the filters at once, without scrolling down for too long to reach any particular filter.

An alternative solution that did not involve accordions was used by the Skyscanner app: it fragmented the screen into a slim filter column and a larger filter-value window.

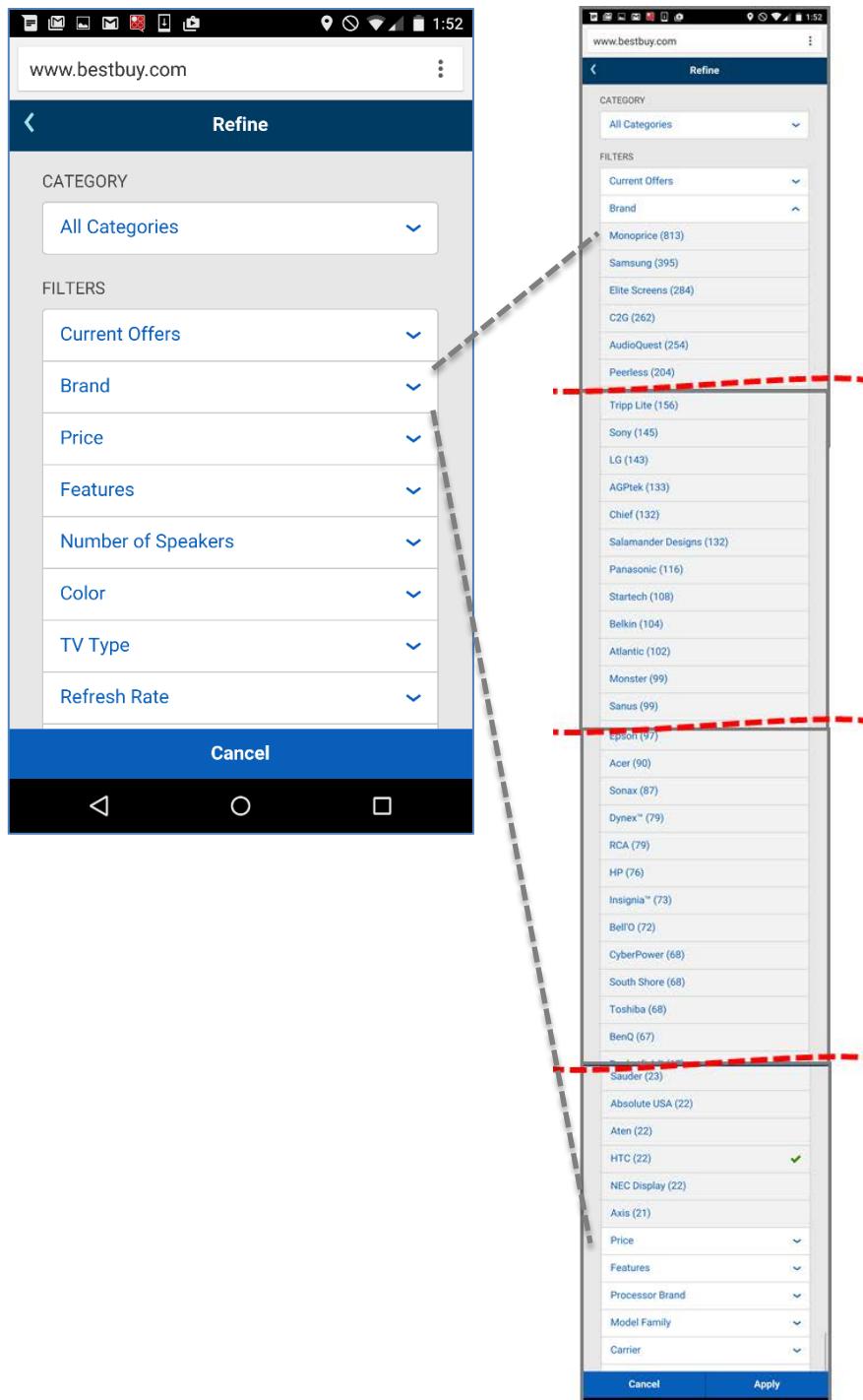


Skyscanner for Android used a vertical filter column and a larger filter-value window.

254. Consider allowing users to quickly close a filter, especially if the number of possible values for that filter is large.

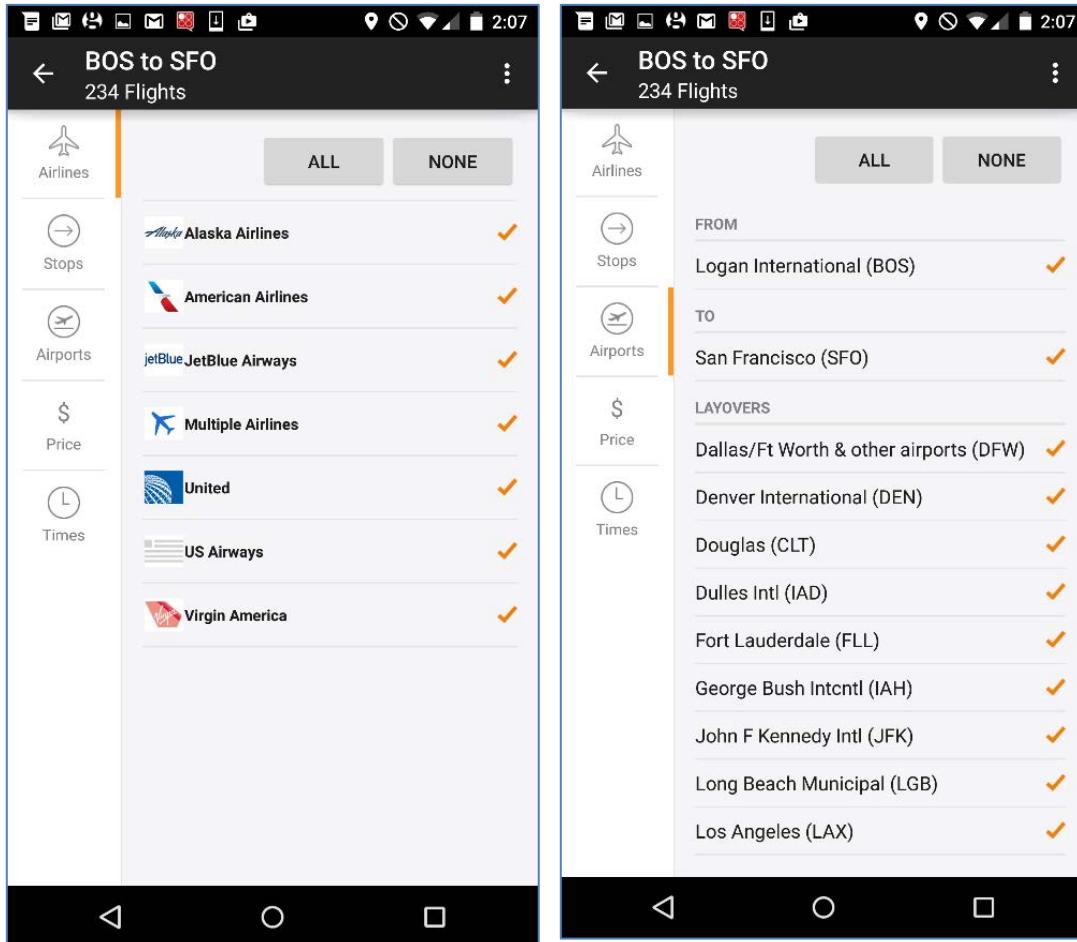
Let's assume the situation where users want to select multiple filters. Once they've chosen one or more values for the first filter, they would need to access the next filter of interest. To do so, they shouldn't have to scroll through the (potentially long) list of values for the first filter.

Let's say that the user needs to set *Brand* and *TV Type* in the Best Buy example below. Once she has selected her preferred brands, she still needs to scroll down the remainder of the *Brand* values to find the next filtering criterion. Alternatively, she could scroll up to the *Brand* accordion, close it, and then look at the filters below *Brand*.



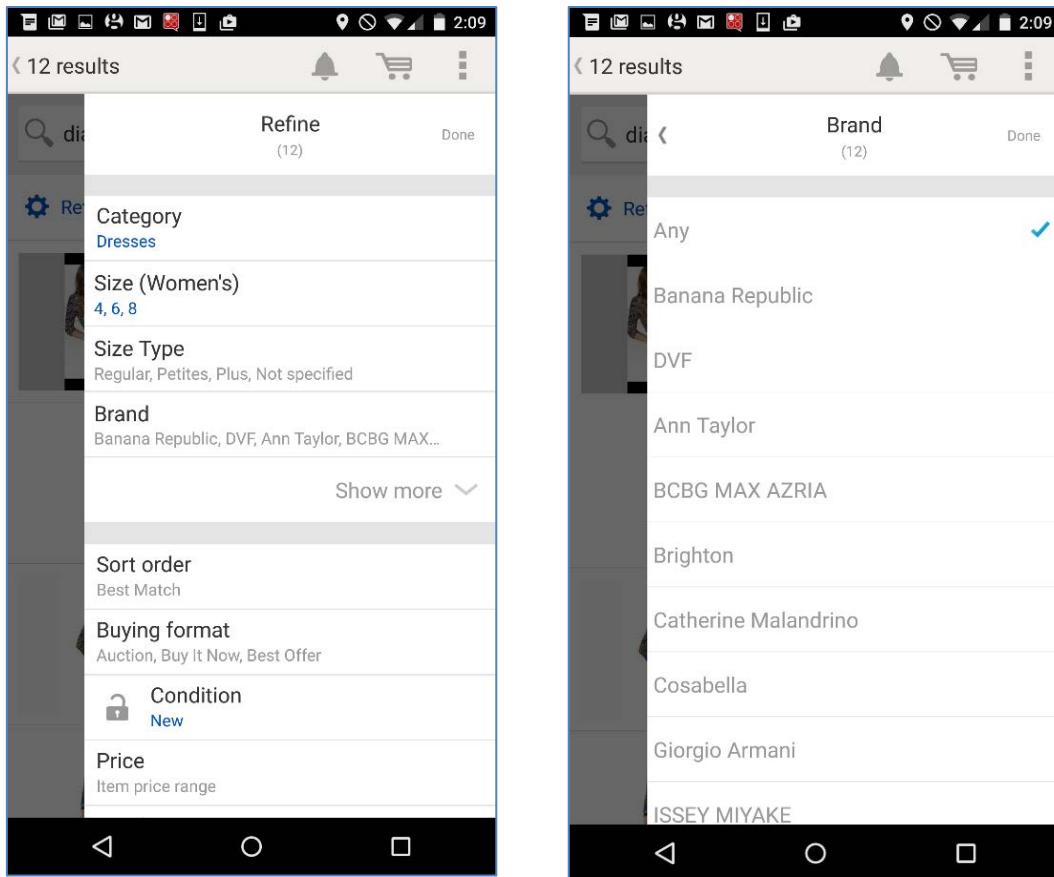
Best Buy: If users wanted to select both a brand and a TV type for their TV, they would have to expand *Brand*, then scroll down through screenfuls of randomly ordered TV Brands, make their selections, then continue scrolling until they got to the end of the *Brand* list in order to find the other filters, including the *TV Type*. (Note that the curved lines indicate that we truncated the list in the interest of space.)

Some designs, such as Kayak's below, naturally avoid that situation by separating the list of filters from the list of filter values:



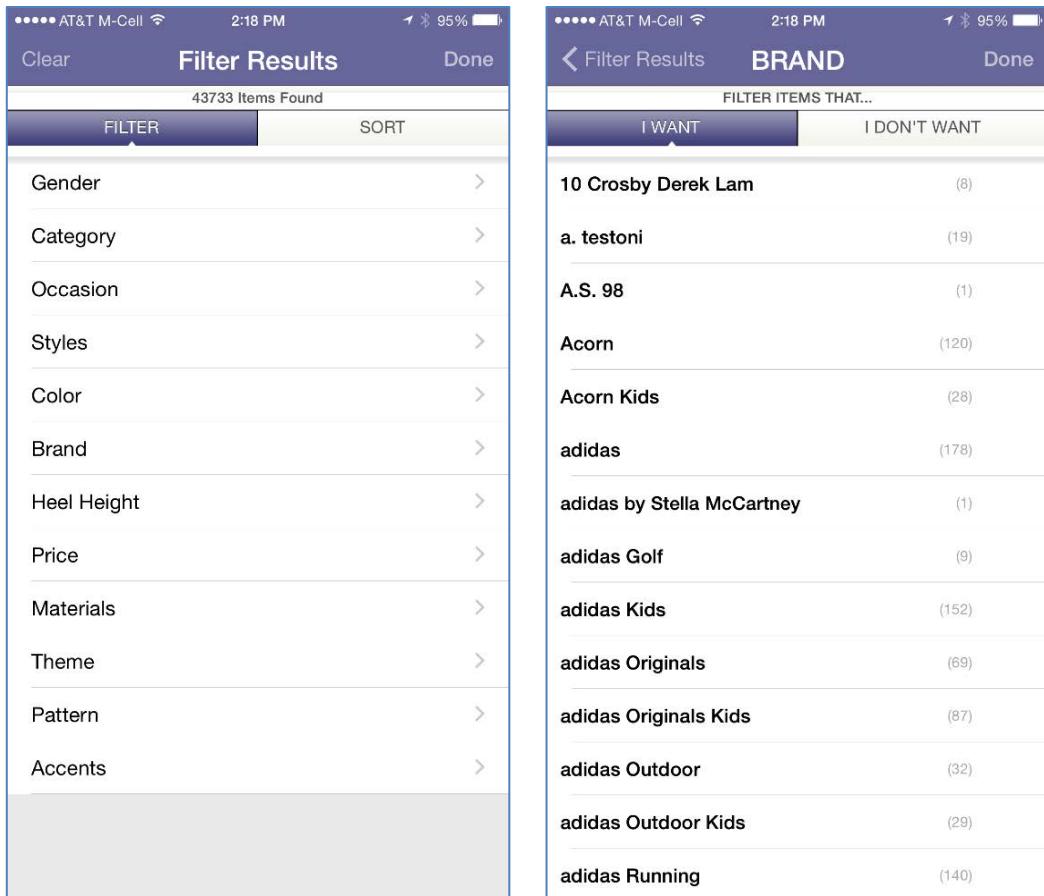
Kayak for Android showed the filters in a column on the left, and the filter values on the right. As soon as users had selected a filter value, they could quickly navigate to a different filter.

Alternatively, the filter values could be displayed on a separate page that users could leave whenever they made their choices, like in the eBay example below.



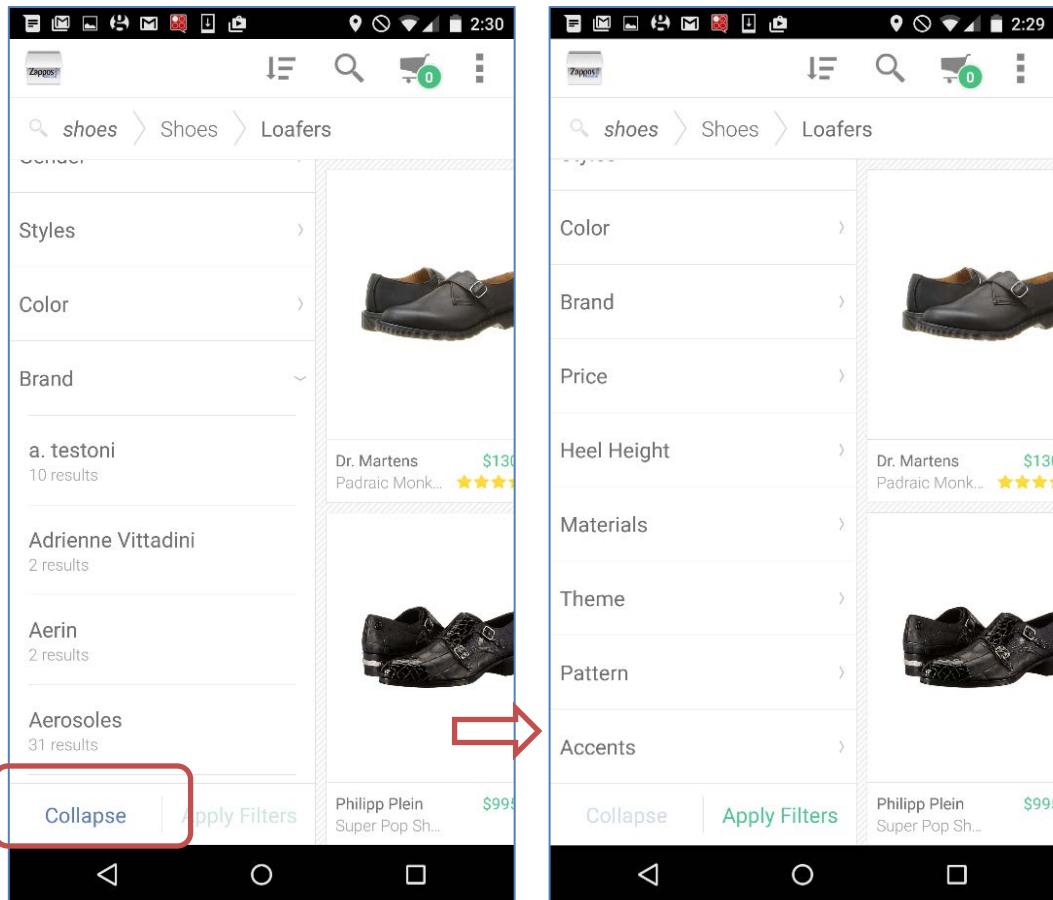
eBay for Android: Filter values were displayed on a separate page (in this case page overlay) with a fixed header; once users selected their preferences, they could tap the arrow at the top of the screen to go back to the previous page with the list of filters.

Note that the overlay implementation was, however, wasteful and error prone; the same idea works well in apps if overlays are replaced with regular pages, as in Zappos for iPhone below.



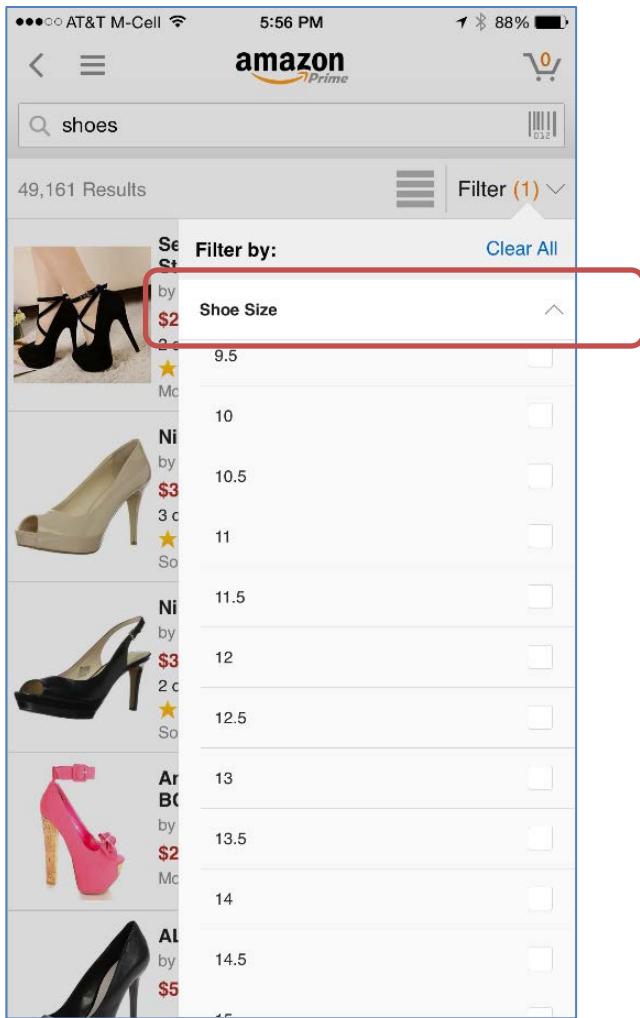
Zappos for iPhone: The list of filters and the filter values were displayed on separate pages. The page navigation bar stayed fixed, so users could quickly navigate to the list of filters once they had made their selections.

Zappos for Android used a different solution that relied on accordions: it expanded the filter values in place, but provided users with a *Collapse* button to let them easily collapse the accordion and get back to the filter-list view. Unfortunately, in testing, no users recognized what that button might do, so they did not use it. (The term “collapse” as applied to a list is meaningful for UX nerds, but for an average user this word is mainly used with a meaning like “he collapsed from heat stroke.”)



Zappos for Android: The *Brand* filter was an accordion that expanded in place (left); if users tapped the *Collapse* button, they could go back to the filter list (right).

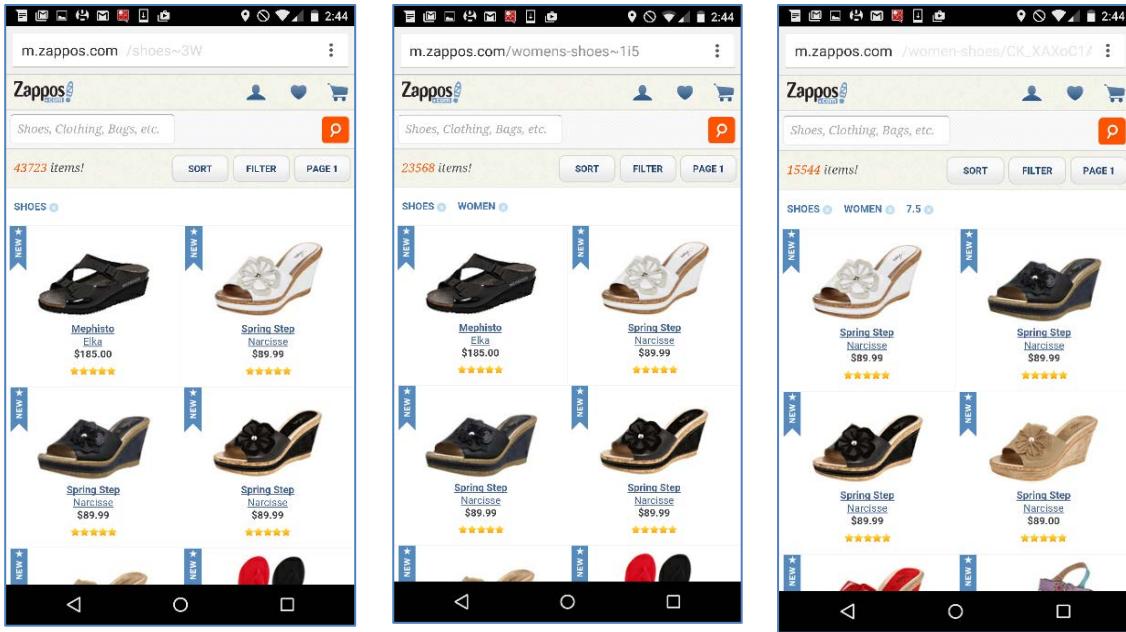
Amazon also used an accordion, but instead of introducing an unfamiliar button, it simply made the accordion control sticky at the top of the value list (see also the discussion under guideline 98 for long accordions).



Amazon for iPhone used a sticky accordion for filters to make sure that users can quickly dismiss a too long filter-value list.

255. Let users specify all filter values before refining results.

If users have in mind a combination of criteria for their search, it's wasteful to apply each criterion separately and have users do multiple filter operations. Thus, if a user knows that she is looking for black high-heeled shoes size 7.5, she should be able to specify all her constraints at once, and get a set of results that satisfies them. She should not have to wait for the app or site to return first a list of black shoes, and then use the filters again to narrow the list down to high-heeled shoes, and apply the filters once more to get only size-7.5 shoes. Unfortunately, this is exactly what the Zappos website did: it increased the interaction cost and forced the user to apply filters repeatedly and wait for multiple lists of results, when one should have sufficed.

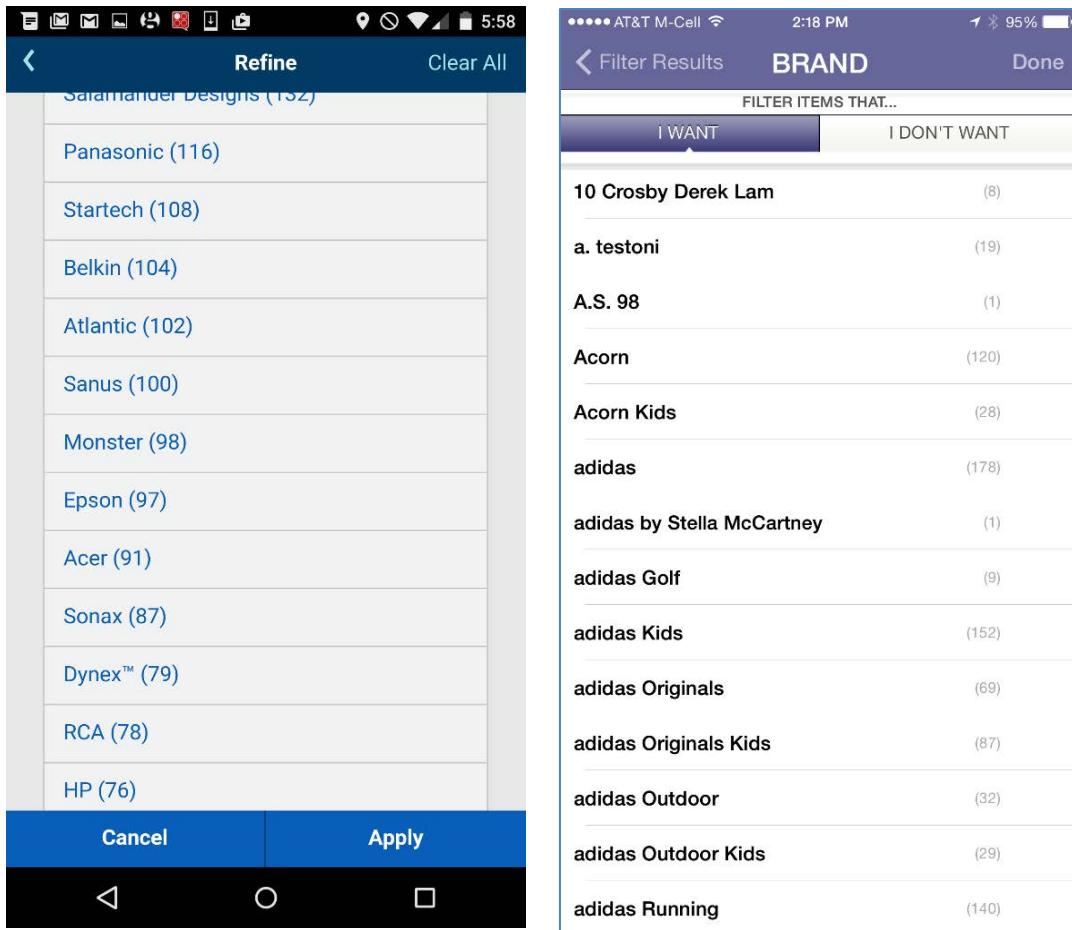


Zappos: Users could only specify one filter value at a time. Thus, to get women shoes size 7.5 they had to run three separate queries (one for each of the three filters).

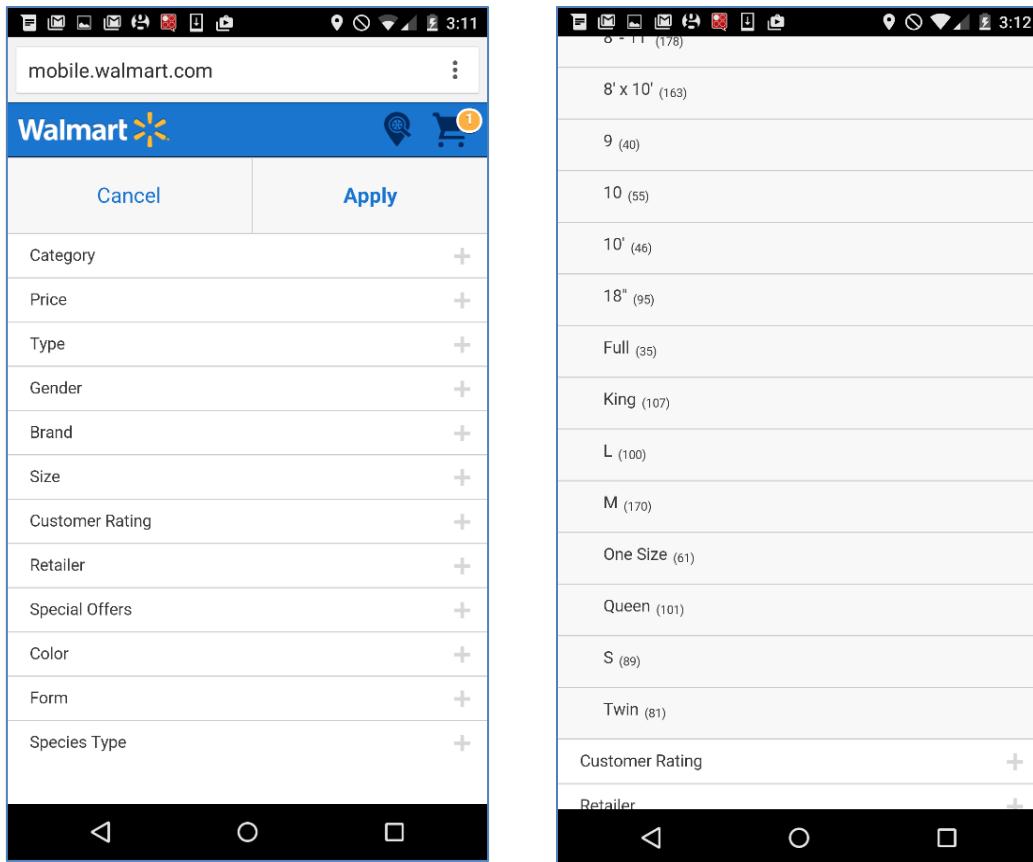
In contrast, the Zappos app did allow users to specify all the desired filters before applying them.

256. Let users quickly access the *Apply Filters* button.

Although the name of that button may vary, it should be easily accessible. It's best when this button is sticky on the screen, and people don't have to scroll up or down to access it.



Filter-submitting buttons were easily accessible on both Best Buy's webpage (left) and Zappos's iPhone app (right). Best Buy used a sticky *Apply* button at the bottom of the screen, and Zappos used the (inappropriately named) *Done* button at the top of the screen, in the page header. Both these buttons stayed on screen as the user scrolled through the list of options.

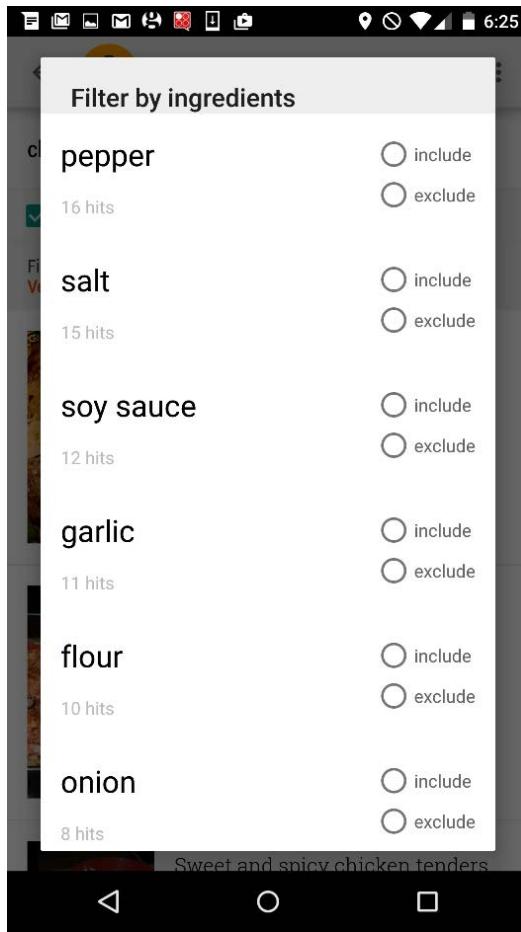


Walmart.com: The *Apply* button (that submitted the filters) was placed at the top of the filter list, but was not sticky (left). As each filter was implemented as an accordion, when people expanded the filters, the page could become quite long. To apply the filters, users would have had to scroll back to the top of the page to find the *Apply* button.

257. Allow users to select multiple values for a filter.

Sometimes users may want to select several values for the same filter (for instance, both Sony and Samsung may be acceptable TV brands, or both low- and medium-height heels might work).

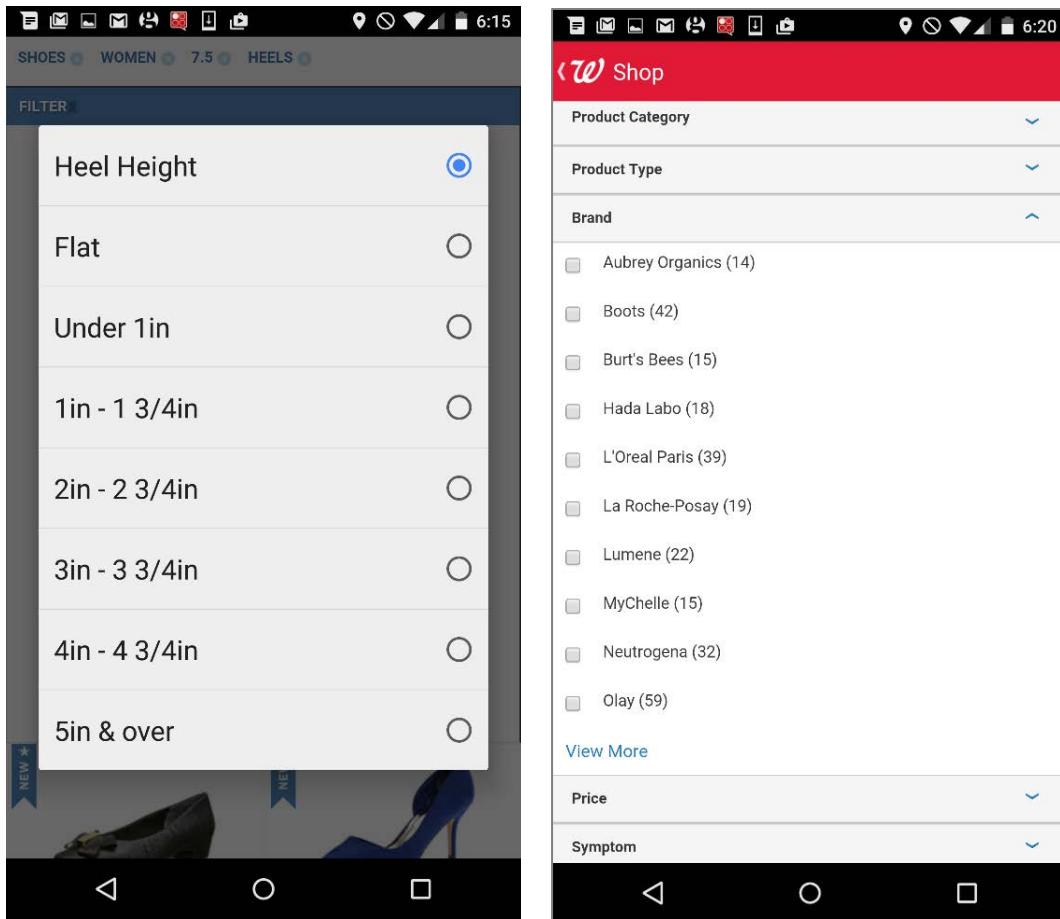
Some apps (like Cookpad for Android below) forced the user to refine the same query multiple times — one for each selection. Thus, if users wanted recipes that did not include onions or soy sauce, they could first specify onions as an ingredient to exclude, would get a list of results, and then could specify soy sauce as a second ingredient to exclude. In the end they would end up with a list of recipes that did not contain onions or soy sauce.



Users of Cookpad for Android could only select one ingredient at the time.

Others websites and apps (like Zappos and Walgreens) did not allow the user to ever specify two filter values for the same query. Thus if a user wanted to see shoes with heels that are either flat or under 1 in., she would need to run two separate queries and deal with two sets of results: one query with the flat-heel filter and the other with the under-1-in. filter.

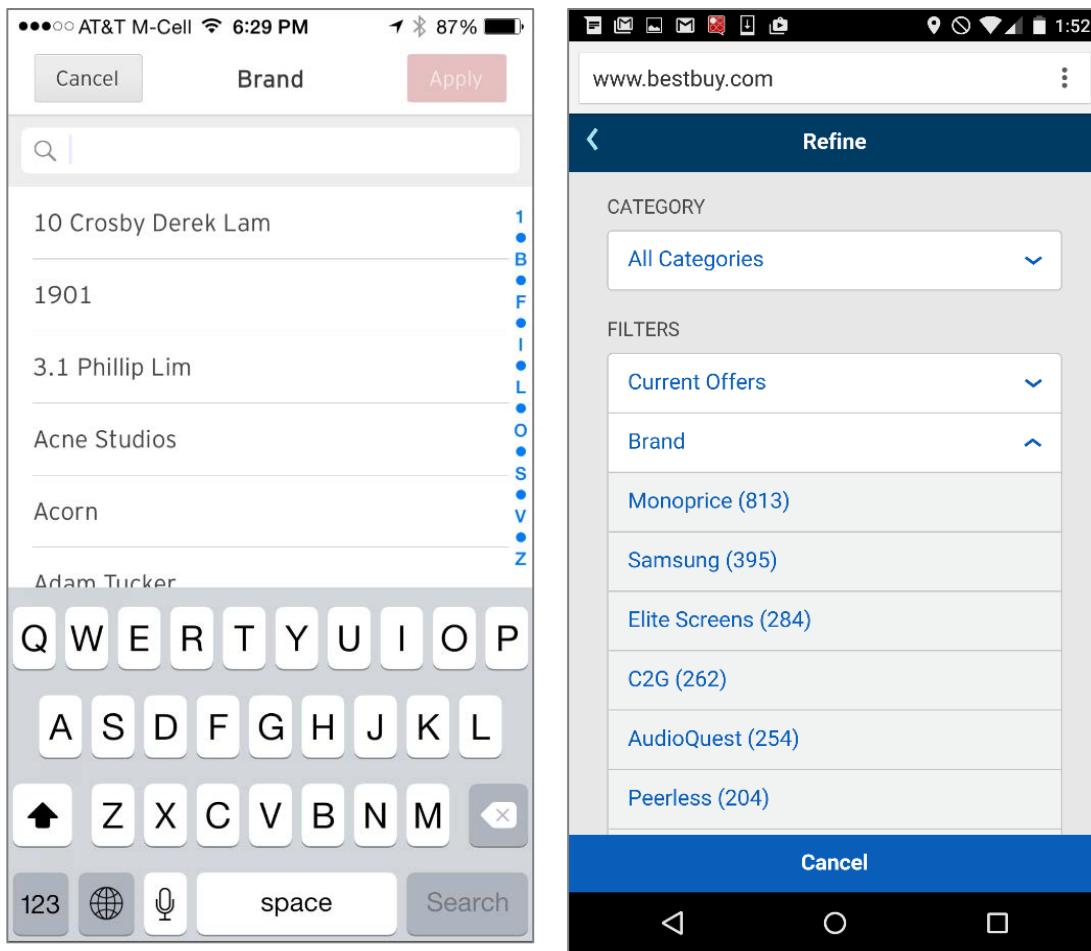
Needless to say, to minimize interaction cost, users should be able to run a single query with as many constraints as they want for a single filtering criterion.



On Zappos.com (left) and Walgreens (right), users could only select one filter value for a filter (e.g., only a heel height or a brand of moisturizer). Once the selection was made, the filter was automatically submitted and the user could not add in a different value.

258. If a filter has many possible values, allow users to type the name of the value to narrow down the list of results.

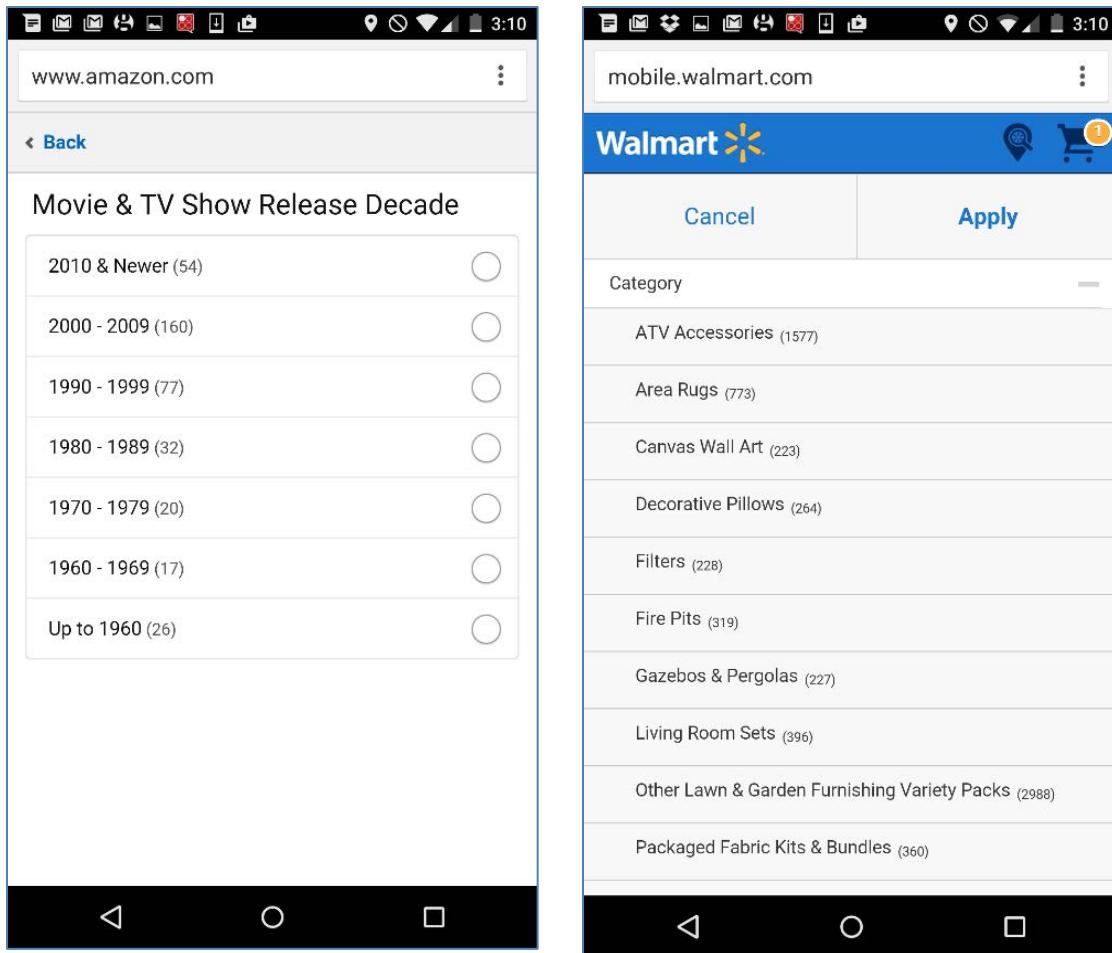
Brand lists are one example of filter that commonly has many possible values. Even if these values were ordered alphabetically, it would still take the user too much time to scroll down to the brand of interest. Instead, provide users with a text-input box where they could type in the name of the brand that they want (or at least the first few letters of that name) and offer suggestions that matched that string.



Unlike Nordstrom for iPhone (left), which allowed users to type the first letters of a brand name, Best Buy (right) forced them to scroll down a long, unordered list.

259. When narrowing down results, show facets and number of items in each category.

Being proactive and telling users how many items are in a given category can help them decide which category to choose next. On the mobile web, where every click counts, it's especially important to let users know what to expect when selecting a category. Moreover, if a category contains no items, seeing that can save users an extra click.



Both Amazon and Walmart showed the number of elements in each facet, so users knew what to expect when selecting any given category.

LISTS AND SCROLLING

260. Do not use a deck of cards (i.e., one item per page) for lists.

See also the discussion under *Gestures*.

Often, when users interact with lists, their goal is to select one element that matches their needs. Many of the lists elements can be filtered out without close inspection. Representing one item per page (besides being completely wasteful for the web, as it forces the user to click many times to get only small pieces of information) forces the user to inspect each item individually, in a fixed order. The items late in the list get fewer chances to be seen (although they may match users' needs better), simply because the users may get tired.

One application that inappropriately uses this model is Epicurious for Android. When users searched for a recipe, they never got a chance to see a list with all the results; instead, they needed to go through individual results one by one.



Epicurious for Android: The list of results for a “cauliflower parmesan” recipe search was spread over 18 pages, with one result per page. Users had to go through all the pages sequentially.

This is a quote from a participant interacting with Epicurious:

“I am not too excited to have to go through every [recipe]... I wish they had a list that said ‘this and this and this and this’ and then I can hit that... because I don’t need to see pictures of them and all that, I’d rather read the list [...], because this is 907 recipes. I am not going to go through 907 recipes! So I don’t know what 907 is, but I’ll never see it. In fact I’ll probably not even see 20.”

261. Use alphabetical ordering for a list if you expect participants to be familiar with the exact names of items in the list.

Brand names, TV shows, movies are good candidates for alphabetical sorting: you can expect users to know their exact names, and thus take advantage of alphabetical ordering. However, with many other items, users may not be able to guess their name as it appears in the list. As a result, in those situations, alphabetical ordering is unlikely to help much.

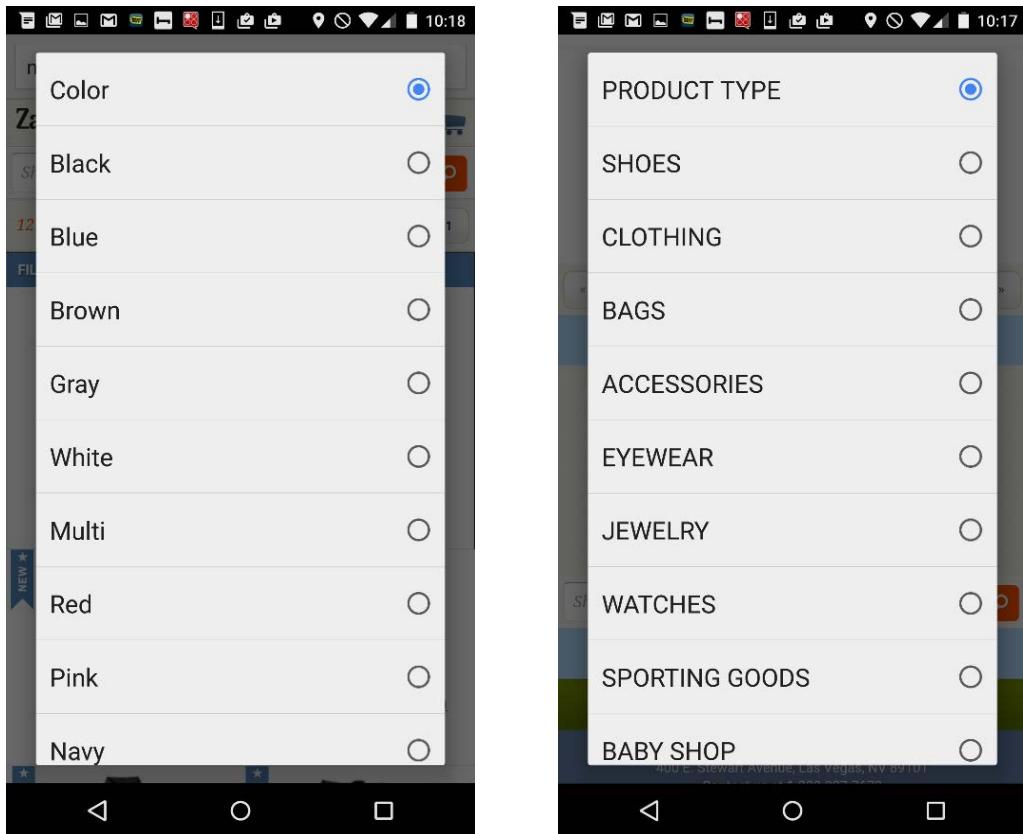
Left Screenshot: Best Buy Brand Refinement List

Brand	Stock Quantity
Tripp Lite	156
Sony	145
LG	143
AGPtek	133
Chief	132
Salamander Designs	132
Panasonic	116
Startech	108
Belkin	104
Atlantic	102
Monster	99
Sanus	99
...	(27)

Right Screenshot: ABC.com TV Show List

Show
American Crime
The Bachelor
Castle
The Chew
Dancing with the Stars
Fresh Off the Boat
General Hospital
Grey's Anatomy
How to Get Away with Murder
Jimmy Kimmel Live
Marvel's Agent Carter
Modern Family
Once Upon a Time

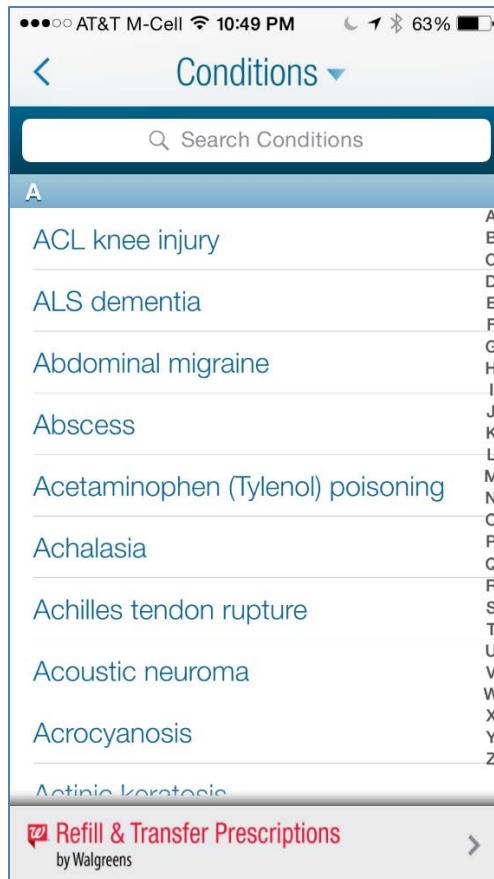
Left: Best Buy did not order brand names alphabetically (instead it chose to order by the number of items in stock, a value that makes little sense to customers). Right: The TV shows were listed in alphabetical order on ABC's website.



Zappos.com: List of colors and product categories were not ordered alphabetically. They didn't need to be, since not all colors have standard names (e.g., "tan" can also be "beige"), and product categories are not standard (that is, someone trying to find ski boots would not automatically think "sporting goods").

262. In a long list organized alphabetically, allow users to easily jump to a certain letter.

It's become almost a standard to see an alphabet scroll bar presented on the right-hand side of a long alphabetical list (see the example below). While the targets on the side are usually small (by necessity), they do allow the users to land closer to the item of interest. In other words, although the user may hit "L" when she wants "N", she will still be closer from her desired goal than if she had to scroll all the way there.



Web MD for iPhone: Although the targets in the alphabet scroll bar on the right side of the screen were small, they did help users get closer to their goal.

View All, Pagination, and Infinite Lists

On mobile every page load is expensive: it takes time, and more importantly, it takes an *unpredictable* amount of time. Even if the first page loaded fast, it does not mean that the next one may also do so, as the connectivity can vary substantially from one location to another.

That's why mobile users don't like pagination when it comes to long lists of similar items, articles, or slideshows. They want to be able to get as much information as possible on a single page to ensure that they won't have to wait again after 1 minute for yet another page load.

The concept of an **infinite (expandable) list** was born out of this observation of the single-page preference among mobile users: the app or website displays the first, say, 30 items in a list on a single page, and if users have reached the end of that, the next 30 items are loaded below, and so on so forth. (Of course, infinite lists do not solve the problem of many unpredictable loading times, since new items are usually loaded only when the user gets to the bottom of the list.)

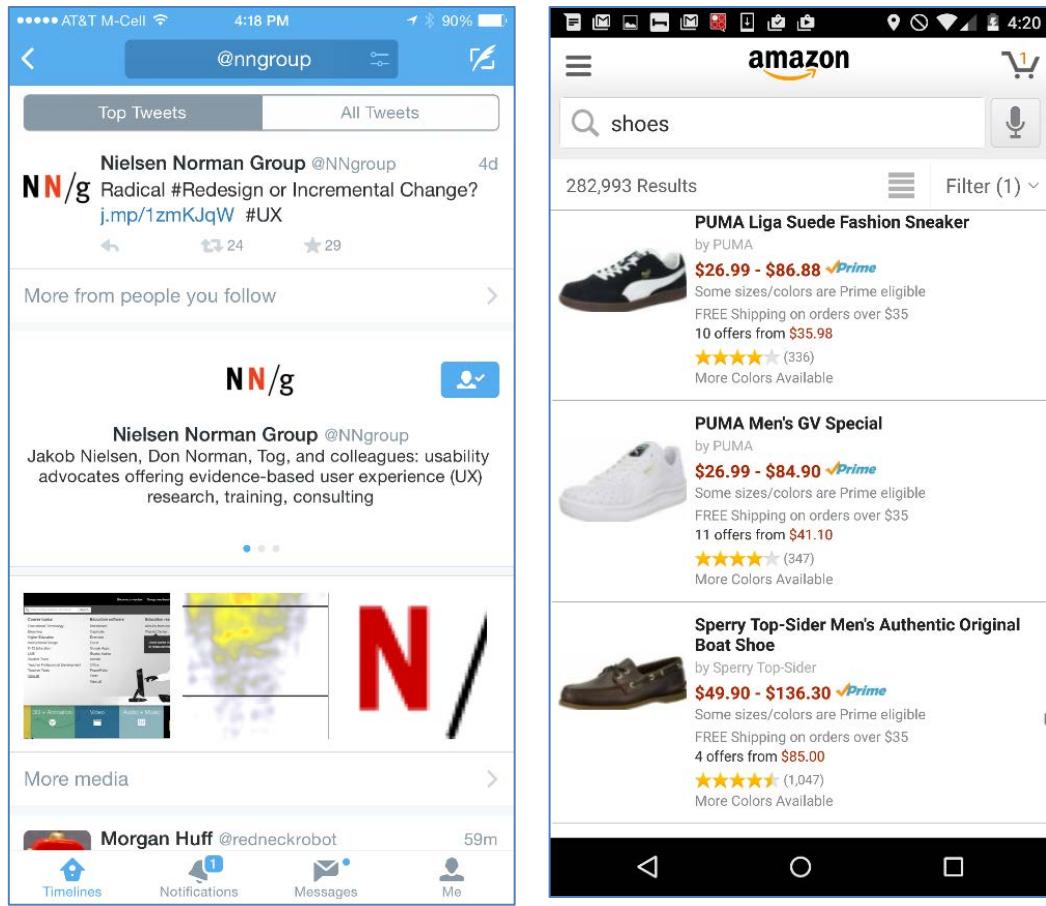
However, really long lists present some problems. First, if a list is excessively long (say, longer than 100 items), it may take too much time to load it at once. (Thus, to avoid that problem, infinite lists load only a few items at a time.) Second, it can be hard to navigate back and forth in such a long lists. The guidelines below are based on these observations.

263. Use infinite (expandable) lists only if users will never have to go back and inspect previous items in the list.

One of the problems with infinite list is that they create overly long pages; thus, navigating back and forth between different elements of the list is hard. Moreover, they also create disorientation, because there are no signposts to help the user remember where an element was in the list. (In contrast, with pages, a user may remember that he saw an item of interest on the first or second page of the list.)

There are very few advantages that infinite lists bring over regular pagination, since in many cases there is a waiting time (similar to a new page load) when a new subset of items is being loaded. One of them is that, provided that the website or app does have a back-to-top button (see guideline 205), it can be easier to navigate back to the top of the list. The other advantage is that sometimes the loading time can be masked, as items can be loaded in the background, before the user has reached the last item in the list.

Infinite lists work well with news app and social-networking apps (such as Twitter and Facebook), where users don't go back and forth between items, and usually act upon an item of interest right away and are done with it. They are less well suited for e-commerce, because often users need to go back and forth between candidate items, and an infinite list makes it difficult to do so.



Social-networking apps such as Twitter (left) can take advantage of infinite lists with no major issues. But e-commerce apps such as Amazon for Android (right) should not use this pattern because it can cause user disorientation and can make product comparison difficult. Note also that Amazon provided a small scroll bar on the right to indicate where the user was in the list. But that location changed as soon as the user reached the last loaded product.

264. For an infinite list, provide users with a back-to-top button or with a sticky header at the top of the list.

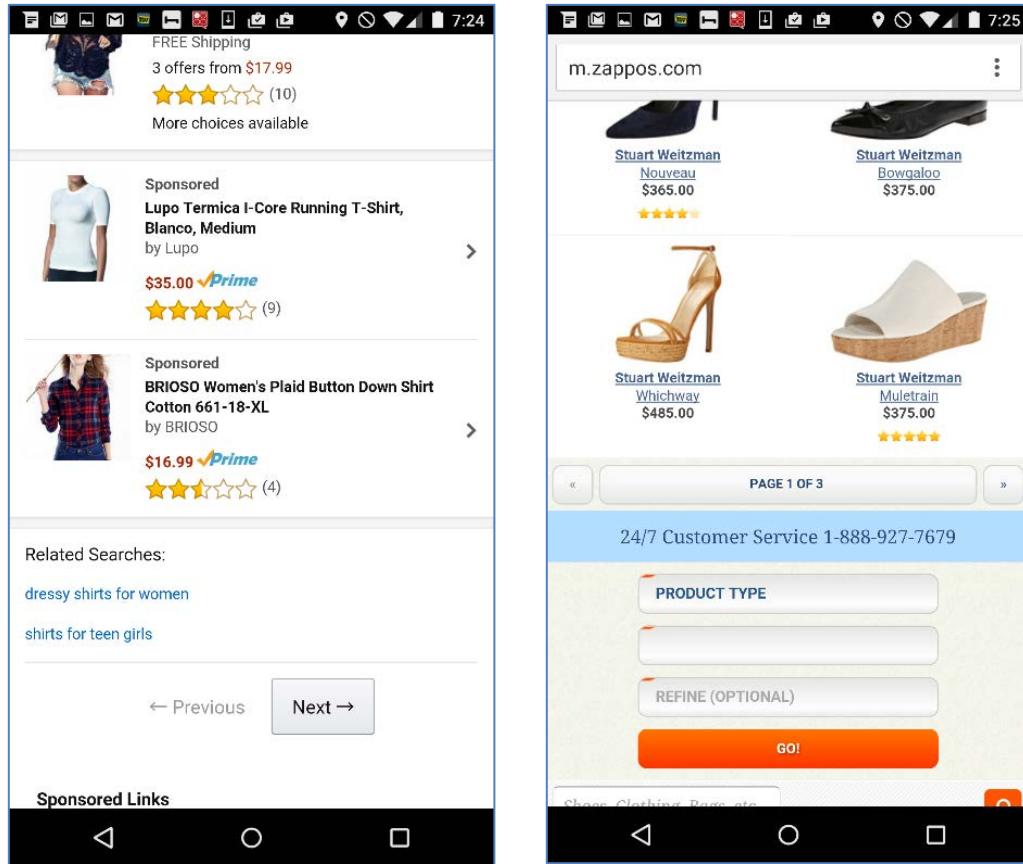
See the discussion under guideline 205.

In the examples above, both Twitter and Amazon used a persistent page header.

265. Prefer *View All* for lists shorter than 100 items; use pagination for longer lists.

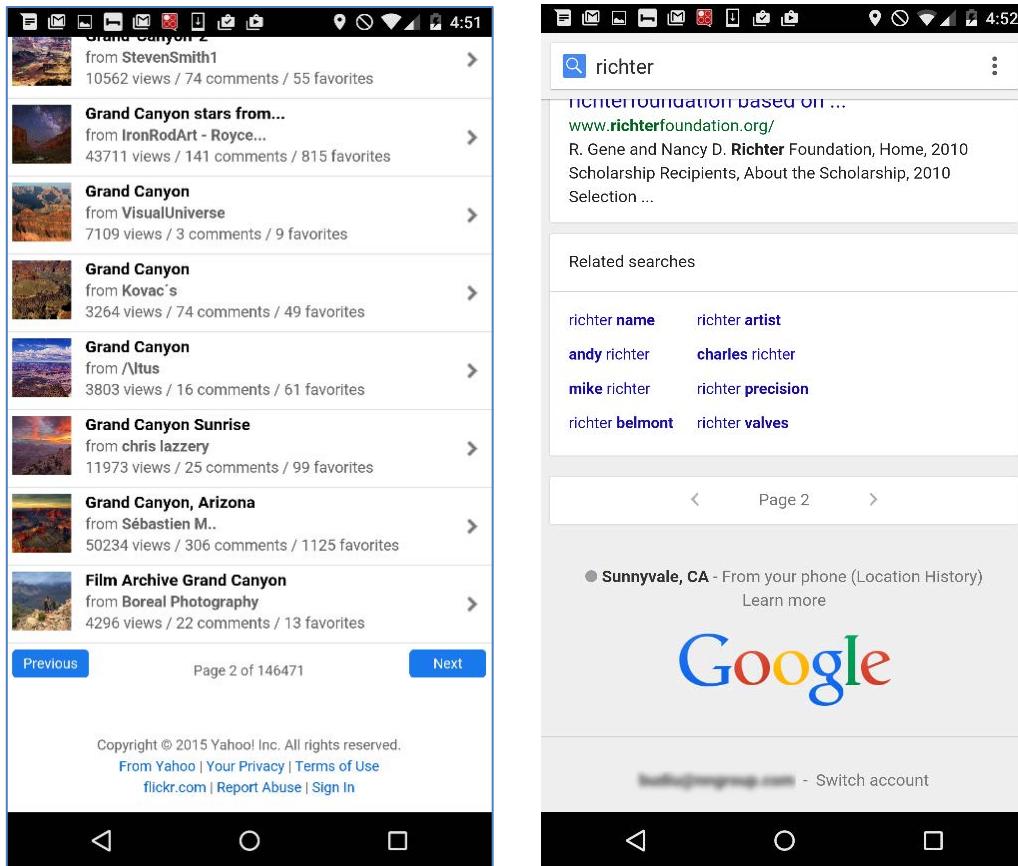
We saw that infinite lists can work well for a subset of apps and websites, but are not for everyone. If you cannot use the infinite-list pattern, then consider showing all the items on a single page in the list if your list is short (less than 100 items); otherwise divide your list into 100-item pages. If a list is shorter

than 100 items, chances are that it's going to be relatively fast to load. That length is also manageable for navigation purposes.



Unlike the corresponding apps, the mobile websites for Amazon (left) and Zappos (right) used pagination for the list of products.

Pagination is also useful when the list is ranked. For instance, most users are aware that with a search engine only the first page of results is usually relevant, so they rarely bother with the next pages.



Flickr.com (left) and Google (right) used pagination to display their lists of results.

CONTENT

It's well known that people rarely read every word on a web page; instead they scan⁶¹. The eye stops and reads words that stand out on the page (either because they are part of a heading, or because they are bolded, or in a different color), and words at the beginning of a paragraph have a higher chance of being read than others. People use the spatial arrangement and visual characteristics of text on the page to figure out what is relevant and worth reading in more detail.

It's probably no big surprise that scanning affects comprehension: a text that was scanned is more poorly comprehended and remembered than if it was read word by word. On mobile, the small screen adds one more complication: even if people find something interesting in the text, chances are they will lose some of the connections with the surrounding material due to the small page. Indeed, mobile content is twice

⁶¹F-Shaped Pattern For Reading Web Content. <http://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/>

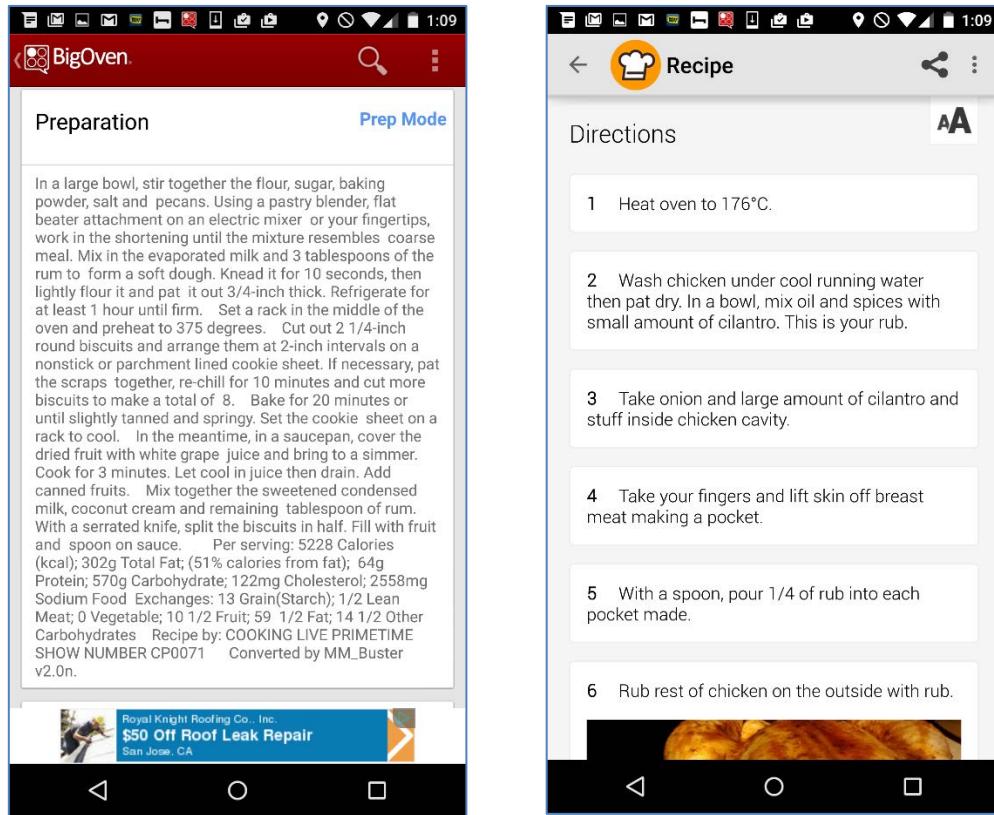
is difficult than desktop content, and comprehension scores on mobile are about half of desktop-based scores⁶².

As a result, many of the content guidelines for mobile enforce scannability and reflect general principles of writing for the web.

Scannability

266. Use bullet points, spacing, and headings to draw users' attention to the gist of your content.

In the example below, the formatting on this Big Oven page was so bad that it's hard to imagine someone following this recipe:



Left: In Big Oven's app for Android, the steps in this recipe were hard to scan and follow. Right: Cookpad for Android did a much better job of formatting the recipe steps for scannability.

In contrast, OSHA should be praised for concise, scannable writing for mobile. Emergency information needs to be read fast and understood easily, and the bullet points and keywords on OSHA's first-aid page helped with that. The illustrations, however, were just decorative: it is unlikely that they can assist users in identifying the corresponding condition.

⁶²Jakob Nielsen. "Mobile Content Is Twice as Difficult."

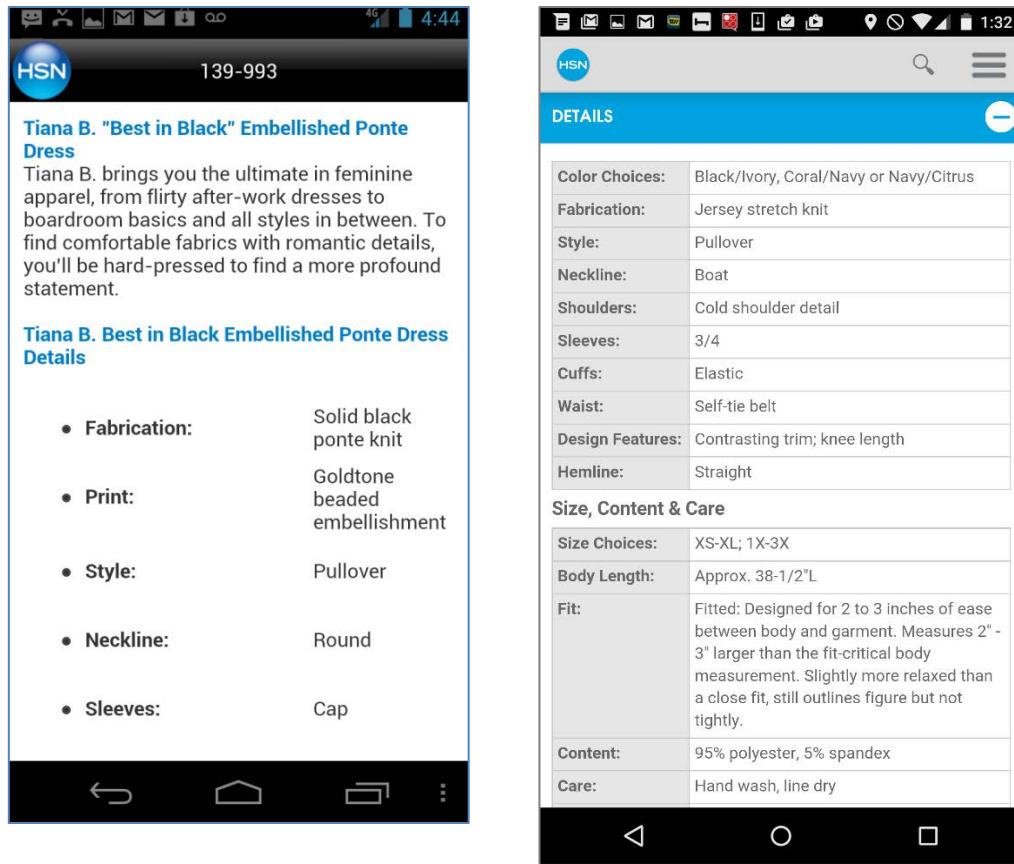
<http://www.nngroup.com/articles/mobile-content-is-twice-as-difficult/>

Illness	First Aid *
<u>Heat stroke</u>	<ul style="list-style-type: none">• HEAT STROKE IS A MEDICAL EMERGENCY. Call 911.While waiting for help:<ul style="list-style-type: none">• Place worker in shady, cool area• Loosen clothing, remove outer clothing• Fan air on worker; cool packs in armpits• Wet worker with cool water; apply ice packs, cool compresses, or ice if available• Provide fluids (preferably water) as soon as possible• Stay with worker until help arrives
<u>Heat exhaustion</u>	<ul style="list-style-type: none">• Have worker sit or lie down in a cool, shady area• Give worker plenty of water or other cool beverages to drink• Cool worker with cold compresses/ice packs• Take to clinic or emergency room for medical evaluation and treatment if

OSHA for Android used bullets, headings, and spacing to support scannability.

An older version of HSN did use bullets and color in an attempt to make the page scannable. Unfortunately, the excess of white space between the bullets, and the table-without-borders format made it hard for the user to know which description went with each option. And the description under the first blue sentence was completely unnecessary on mobile: it contained brand generalities, rather than focusing on the product facts. Brand information is better delegated on a secondary page.

A redesigned version of HSN substantially improved the scannability of the page, by addressing exactly the problems that we've just pointed out.



An older version of HSN for Android (left) had too much white space; the tighter table style used in the redesign (right) represented an improvement in scannability.

267. Prefer brief, concise sentences over convoluted ones.

People rarely have the uninterrupted time to read carefully a whole paragraph on mobile. It's best to strive for short sentences that start with the essential.

One user complained about the text next to a checkbox on Interflora's website:

"These [checkboxes] are annoying because sometimes you read them, and you have to read it because you don't know if it's going to be 'tick this box if you don't want this crap sent to you' [...] [reading] So they'd like to send you some special offers ... 'if you'd prefer not to receive these offers, tick here' — right, I can stop reading it at that point, but I still had to read it... It should have a title "Would you like to receive offers?" or even "Offers?" and then underneath the full lengthy blurb ..."

Billing postcode:
(If different from delivery address)

Your eMail address: *

Mobile number: *

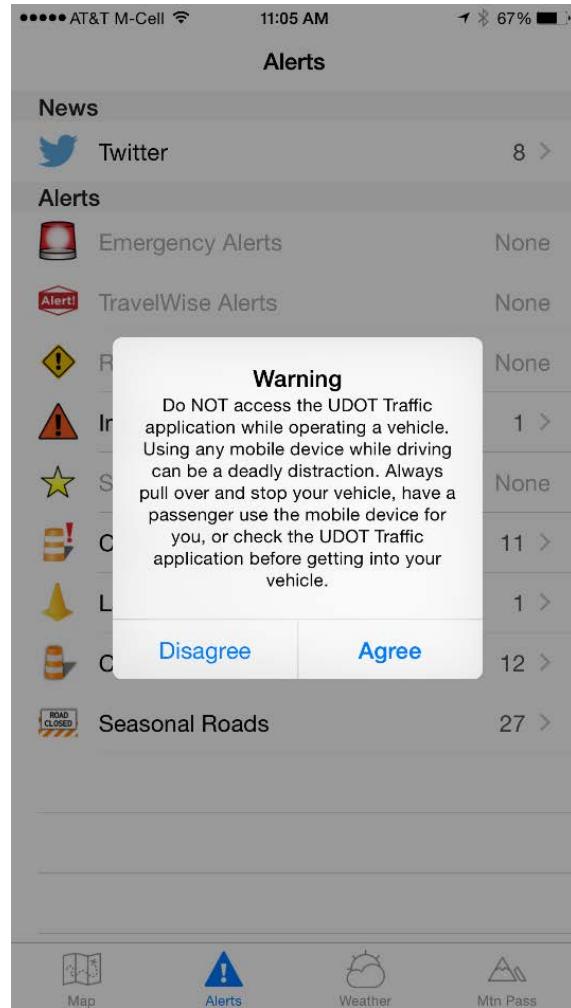
Interflora would like to send you special offers and information about new products. If you would prefer not to receive these offers please tick here. We will not pass your information on to other companies

Continue With Order

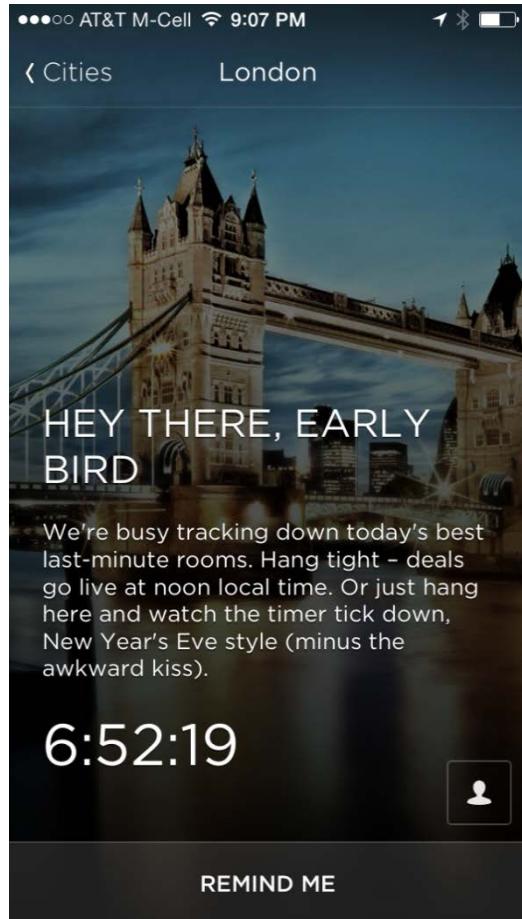
* Required Information

More great services...

Home >



Left: On Interflora.co.uk, the text next to the checkbox was too lengthy and hard to understand even when read carefully. Right: Utah Department of Transportation (UDOT)'s app for iPhone used a verbose alert to explain why the app was not supposed to be used while driving.

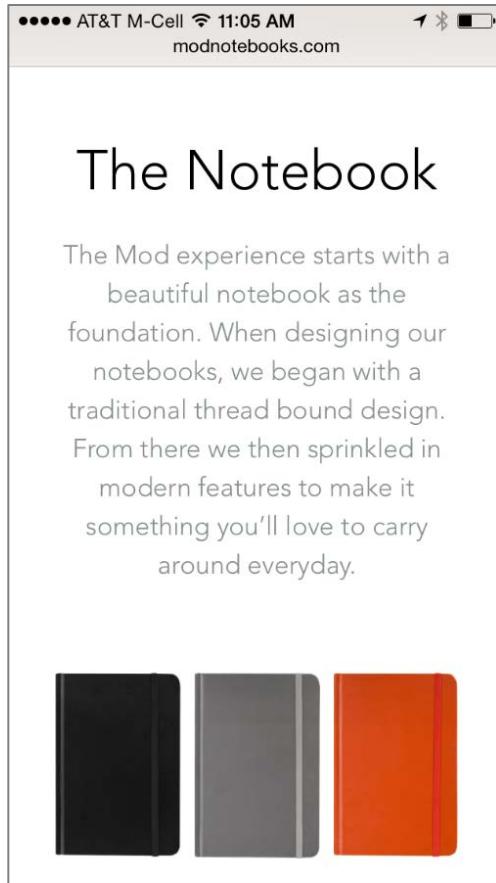


Hotel Tonight for iPhone chose convoluted language to communicate that their deals had not been posted yet.

268. Use left-aligned text.

Left-aligned text is simply easier to read: the eye doesn't have to find a start point for each new line, but instead, the start point is predictable and always the same.

Don't succumb to the temptation to prettify your pages by centering text (or by right-aligning it).

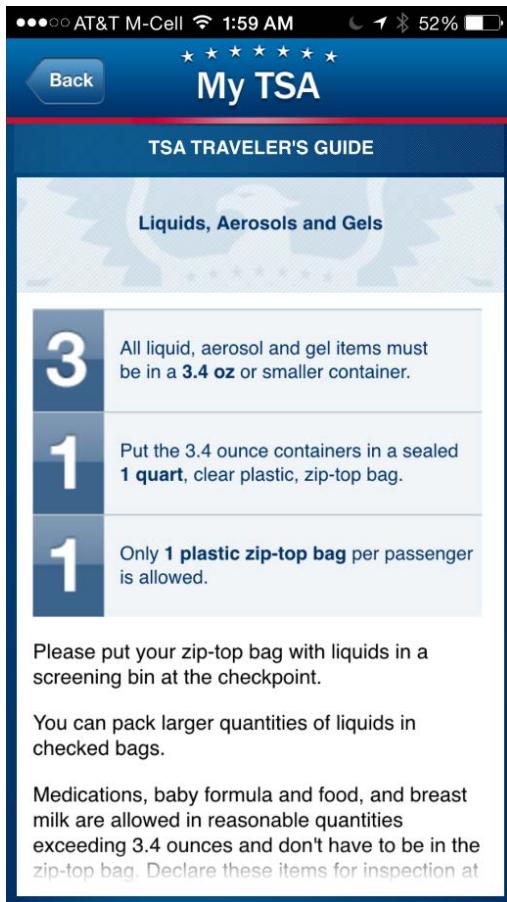


Modnotebooks.com used centered text for their website instead of left-justified text.

269. Stay away from jargon and industry-specific terminology.

This is a guideline that is valid for websites in general, and stays even more valid on mobile, where people don't necessarily have the time to investigate unknown words.

My TSA app used big icons with the digits "3", "1", "1", to explain the 3-1-1 rule referring to what liquids can be brought in a carry-on on board of an airplane (3.4 oz. containers that must fit into a 1-quart plastic bag; a passenger is allowed only 1 bag). Most of our participants did not understand what these icons meant and, although they quickly scanned the explanatory text next to the icons, they never got the connection between the digits and the rule: they simply did not have the time to make that inference.



My TSA app for iPhone: Our participants never understood what the 3-1-1 rule was and what connection there was between the big icons and what they were supposed to bring on the plane.

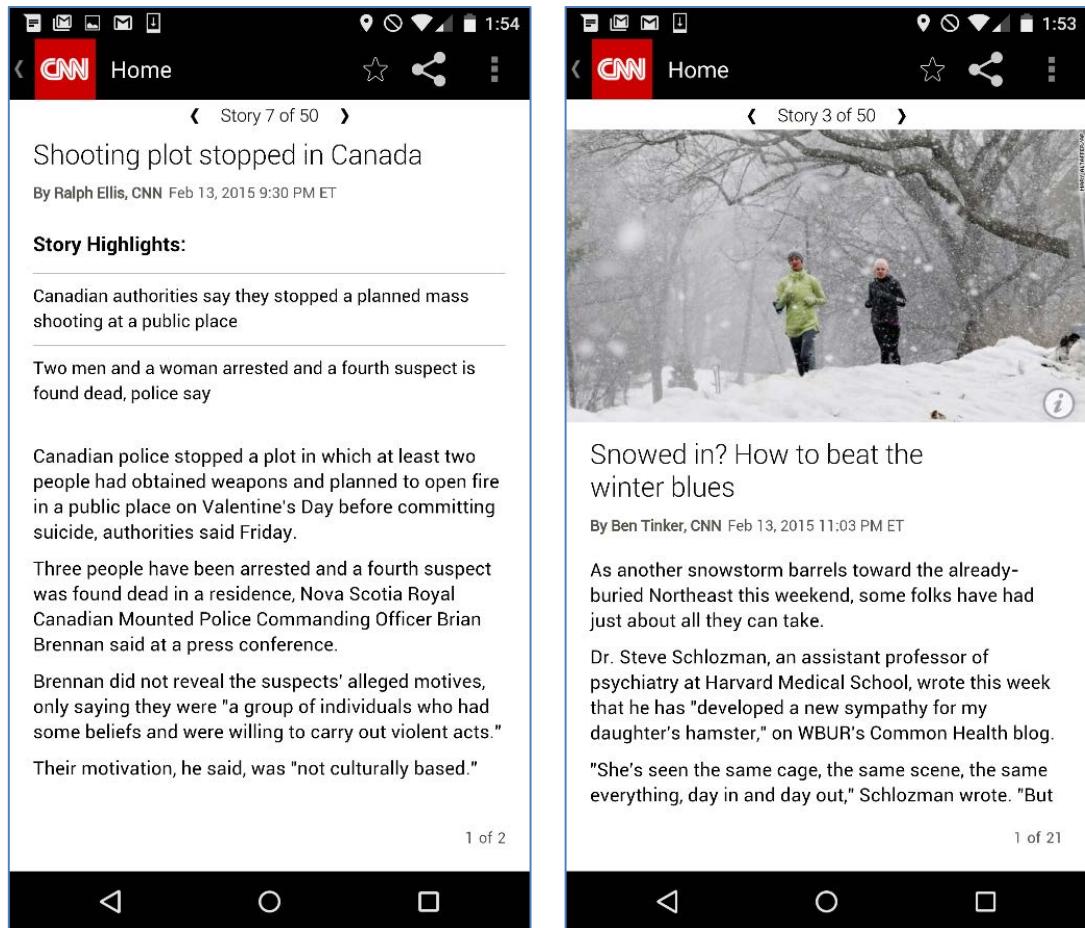
270. Use an inverted-pyramid writing style. Present overviews and summaries at the beginning of the article.

Journalists and writers learn to use the inverted-pyramid writing style, and always start with the most important piece of information. The inverted-pyramid works when writing for the web, as well as for mobile.

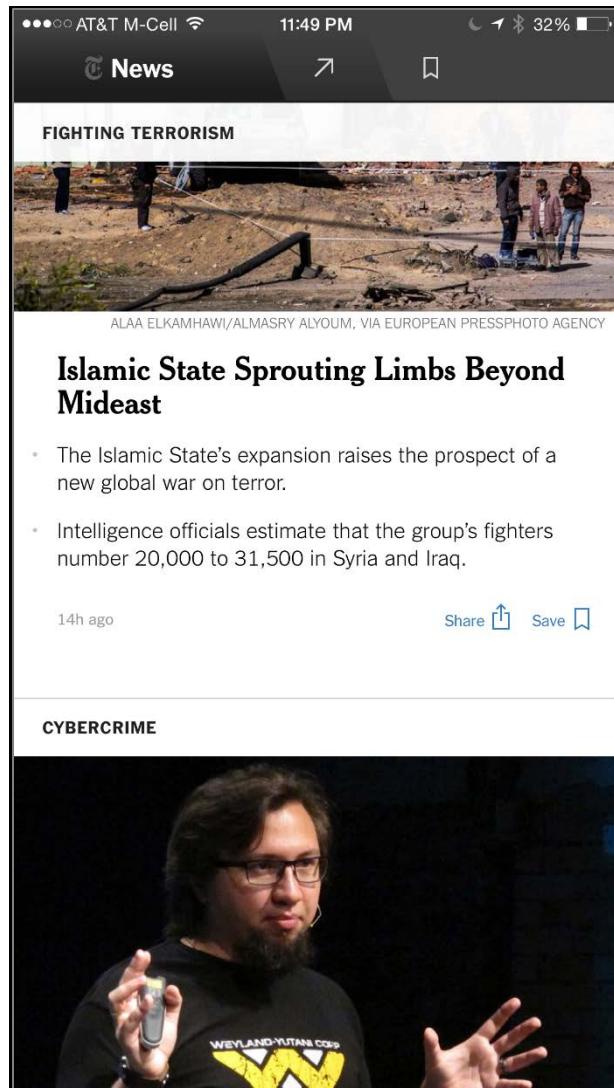
Always try to give a short summary or overview at the beginning of the article to synthesize the main points of the article. Moreover, when developing the article, leave the nonessential details for the very end (or even remove them completely).

In reading a "breaking-news" story about a tornado, one test user found commentary from local people and said, "I don't need to know what everyone else is saying and the event from their point of view. I don't mind a quote from a local leader, but all this to me is just filler, and I wouldn't read it."

She went on to say, "This is what came to me as breaking news? That's too much. It should be: This is what happened, and this is what's going on."



CNN for Android: Some stories contained *Story Highlights* that captured the gist of the story (left). However, unfortunately, not all articles had that feature (right).



NYT Now for iPhone showed a bulleted-point summary of the articles.

271. Defer secondary information to secondary pages or collapse it under accordions.

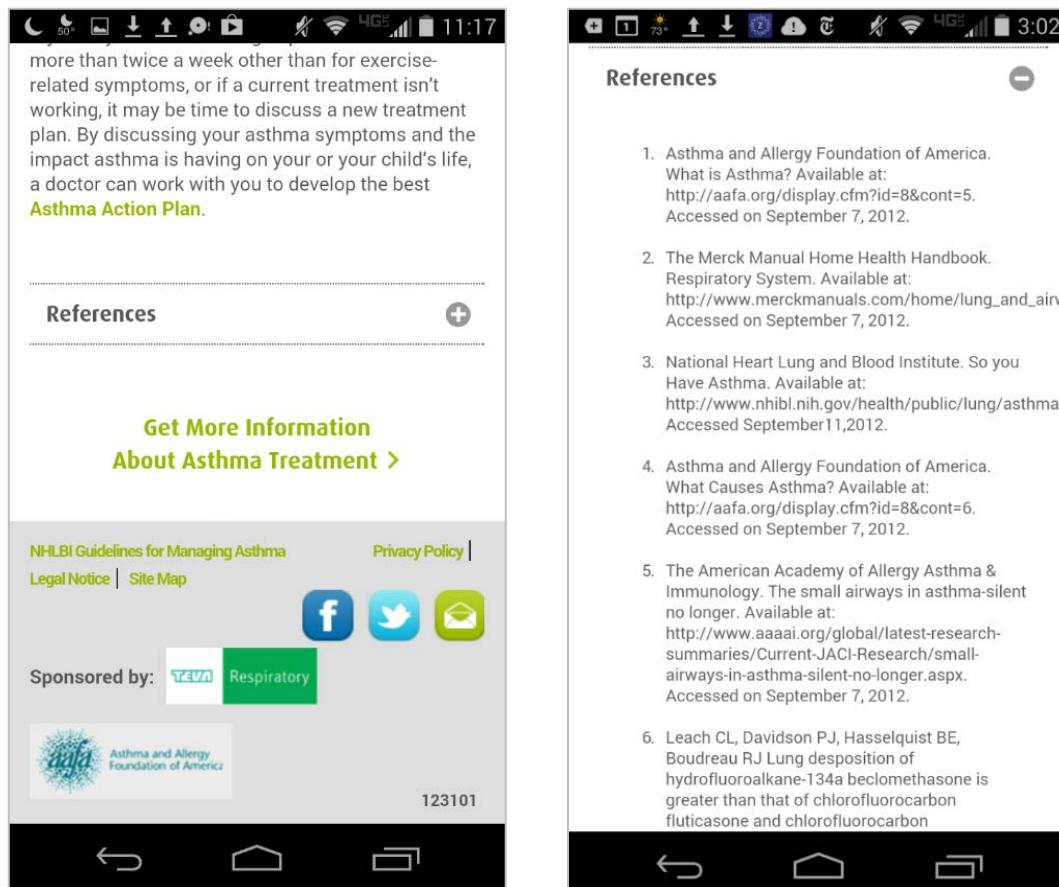
Defer background material to secondary screens shown only to users who explicitly ask for more info⁶³. Such additional content supports people who have extra time on their hands or an exceptional interest in the topic, but it does not overwhelm regular users who are likely interested only in a general overview.

⁶³ See also Jakob Nielsen. "Mobile Content: If in Doubt, Leave It Out." <http://www.nngroup.com/articles/condense-mobile-content/>

It's a tough decision to defer information to secondary screens because many users will never see it there, even though you no doubt consider it very important.

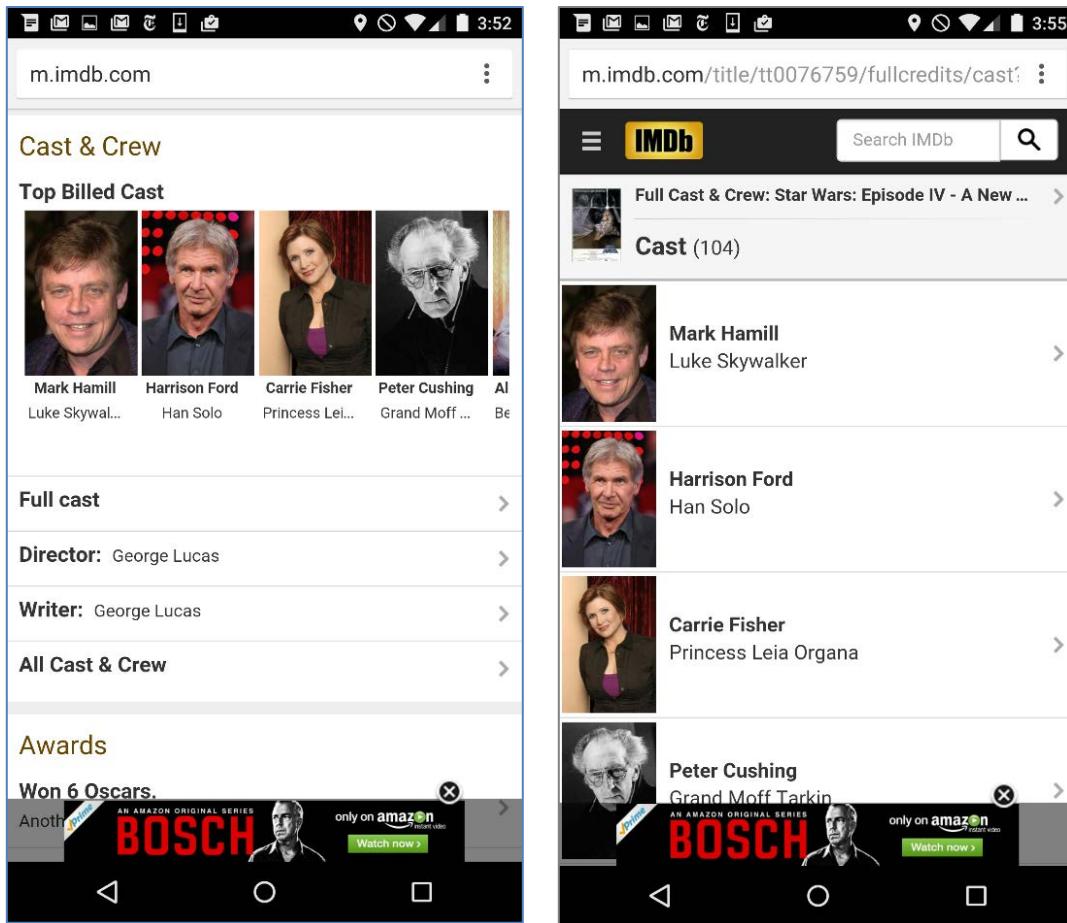
But remember: if you make the first screen too dense, then *nobody* will read anything. Better to focus the initial screen and let those users who're particularly interested dig into the rest. That way, you'll satisfy more customers, get more traffic, and derive more business value from your mobile content.

Deferring information can mean either using a separate page to display it or hiding it into a collapsed accordion that can be expanded in place. In the example below, GetSmartAboutAsthma.com chose to hide the article's references under an accordion; interested users could expand the accordion, but for most users that information was not important enough.



GetSmartAboutasthma.com: The article bibliography was hidden under a References accordion.

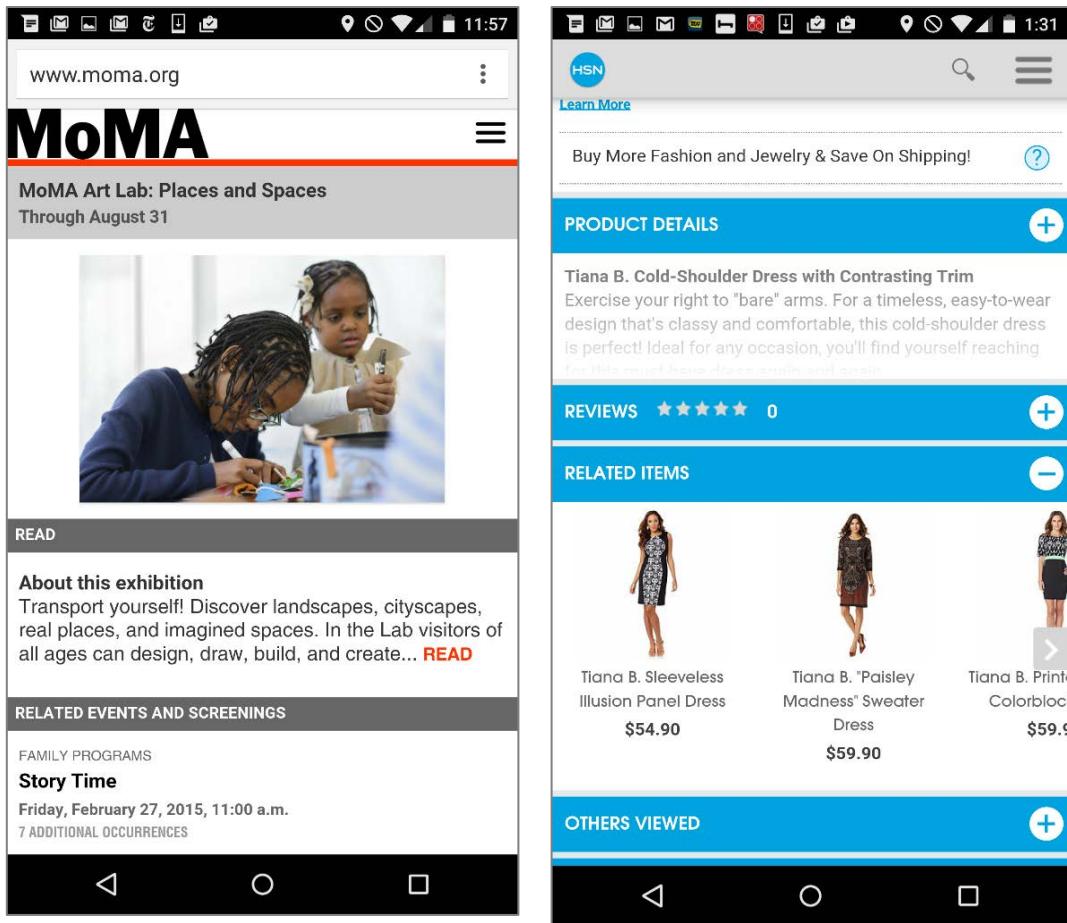
On the movie page, IMDb presented some information about the movie (e.g., top cast, director, writer), but correctly delegated the detailed information to a secondary screen because most users are not likely to be interested in it.



IMDb.com showed only the information likely to elicit interest from most users on the movie page. The complete movie details were delegated to a secondary page.

272. Do not arbitrarily truncate text paragraphs or product information in lists of products.

Although we do recommend that you defer secondary content to secondary pages (see guideline 271), that does not equate with truncating a paragraph to the first few lines and forcing users to go to another page or expand an accordion to read more. The content presented on the main page should truly be the most important part; randomly cutting through a paragraph doesn't usually result in a good, scannable selection of that section's meaning. Instead, it forces most users to incur a higher interaction cost — either by expanding the accordion or by loading a new page.



MOMA (left) and HSN for Android (right): In both cases, truncating a paragraph did not result in a satisfactory summary of the exhibition (MoMA) or of the product (HSN).

273. Use links with good information scent.

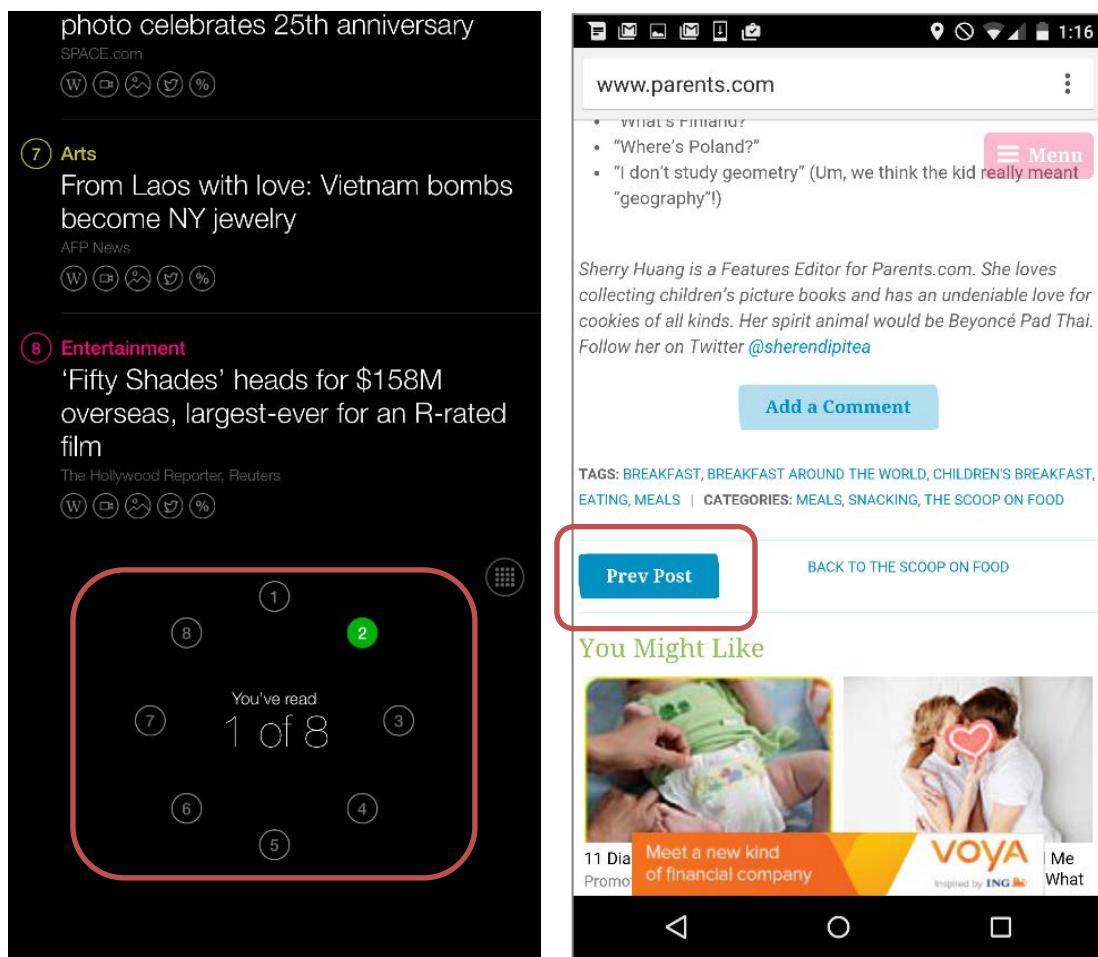
Information scent refers to the level of relevance that a page has with respect to a user goal⁶⁴. People choose to follow a link only if they assume that it will help them satisfy an information need or complete a task. The name of the link is crucial in assessing whether a link is worth following or not; the more descriptive and precise the name, the more likely that users will be efficient and thus satisfied with your website and app.

This guideline is a classic website-design guideline, but it becomes even more important on mobile devices, where every extra click delays the user significantly. The text of the links should be transparent and should set good expectations about the content of the page⁶⁵. Meaningless links suck as *Next*

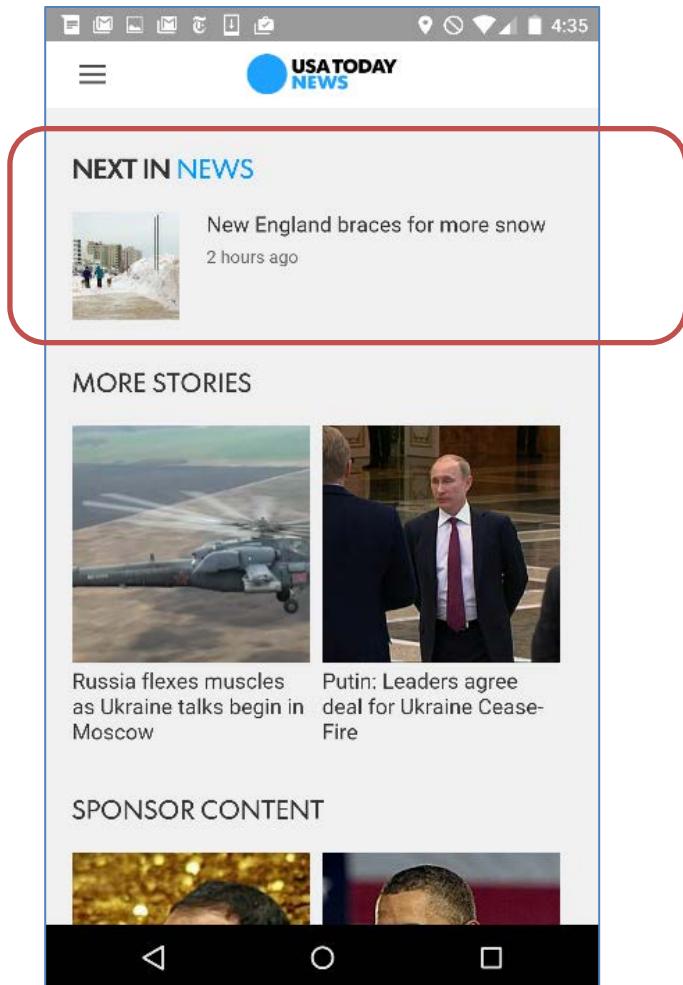
⁶⁴ See also Jakob Nielsen. "Information Foraging: Why Google Makes People Leave Your Site Faster." <http://www.nngroup.com/articles/information-scent/>

⁶⁵ See also Kara Pernice. "A Link is a Promise." <http://www.nngroup.com/articles/link-promise/>

Article or Previous Article may be easy to generate automatically, but are too vague — the links should contain the title of the story and possibly a summary, like USA Today did (see below).



Left: Yahoo News Digest for iPhone showed the links to stories as numbers (no text description), which made it convenient for seeing how many stories you've read, but not for knowing what they were. Right: Parents.com link to *Prev Post* carried little information scent.



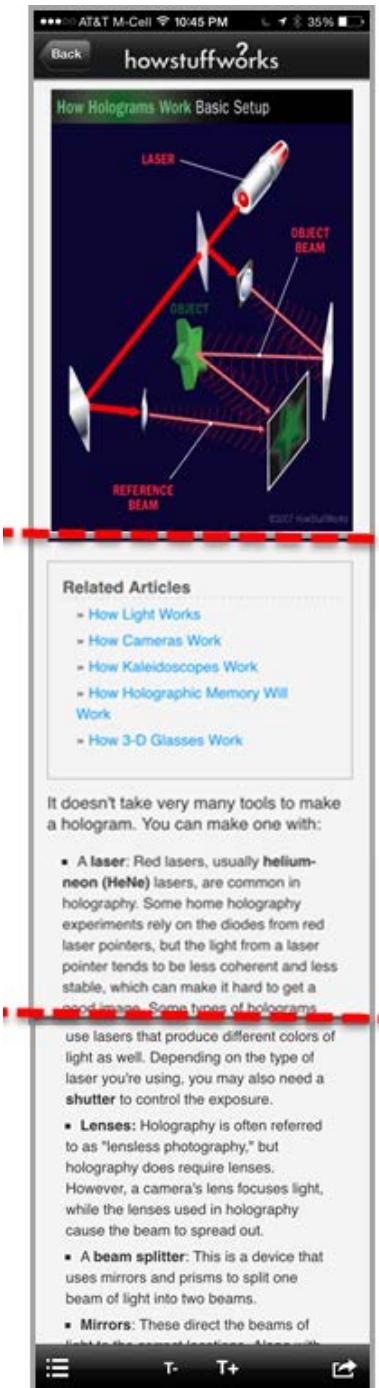
USA Today for iPhone indicated what the title of the next story was, thus helping users decide if they wanted to read it.

Moreover, if secondary information is going to be deferred to secondary pages (as we recommend in guideline 271), the links to these pages need to be explicit enough so that people know what they commit to. A link labeled *Learn More* is too vague, and does not tell users whether the content on that page will be worth of learning.

274. Avoid back referencing in articles; that is, do not force users to go back and forth between different parts of the article.

Compared to a regular monitor, a small screen fits less content and users can see less at any given time. Thus, they must rely on their imperfect memory to tie together different pieces of information that may not fit on the same screen. Moreover, on a small screen users must scroll more; thus, some of their cognitive resources must be spent on interacting with the screen and making sure that they correctly reacquire their previous location once they've scrolled down.

If a paragraph refers back to information presented one screenful away, users may either need to go back and recover context or they may simply sacrifice comprehension instead of paying the higher interaction cost.



How Stuff Works (HSW) for iPhone: A description of how holograms work spread over 3 screenfuls and required users to remember the diagram or go back and forth between the image and the textual description.

Readability

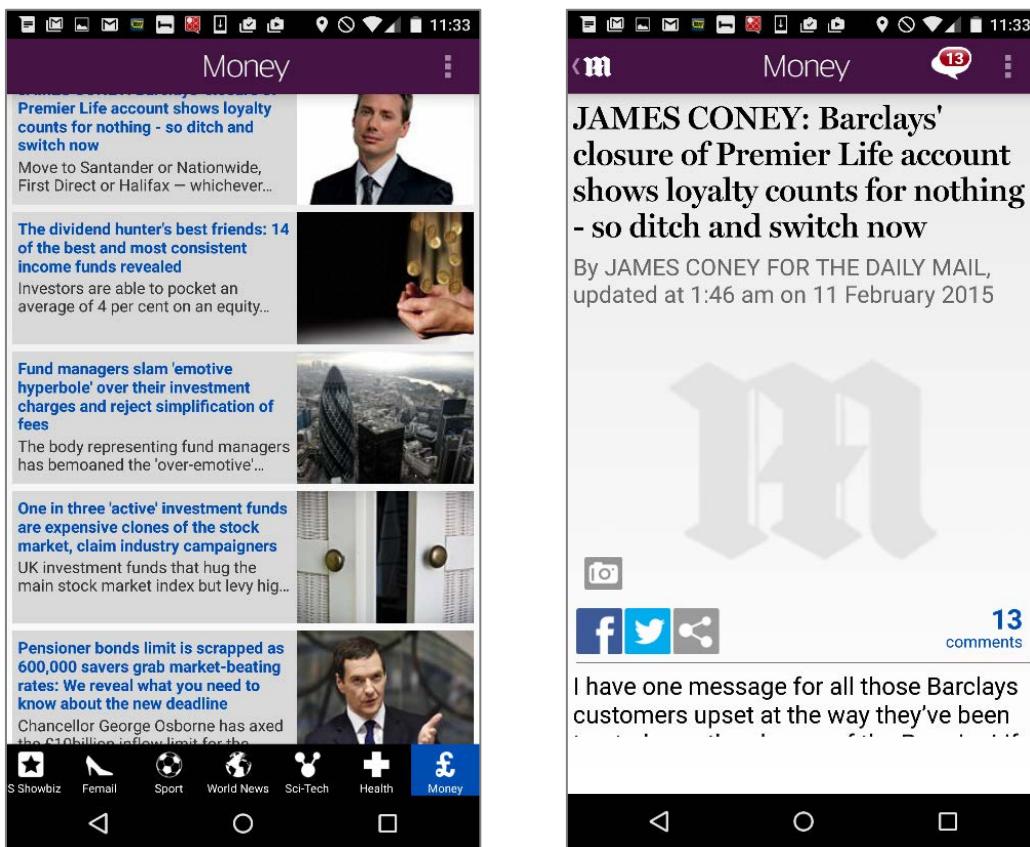
Because the mobile screen is small and mobile devices are used in a wide range of light conditions, reading is generally harder. Therefore, you should take every action possible to make it easier for users to read your content. Our desktop readability guidelines not only stand on mobile, they also become even more stringent.

275. If your site contains a lot of text, include tools for changing font size.

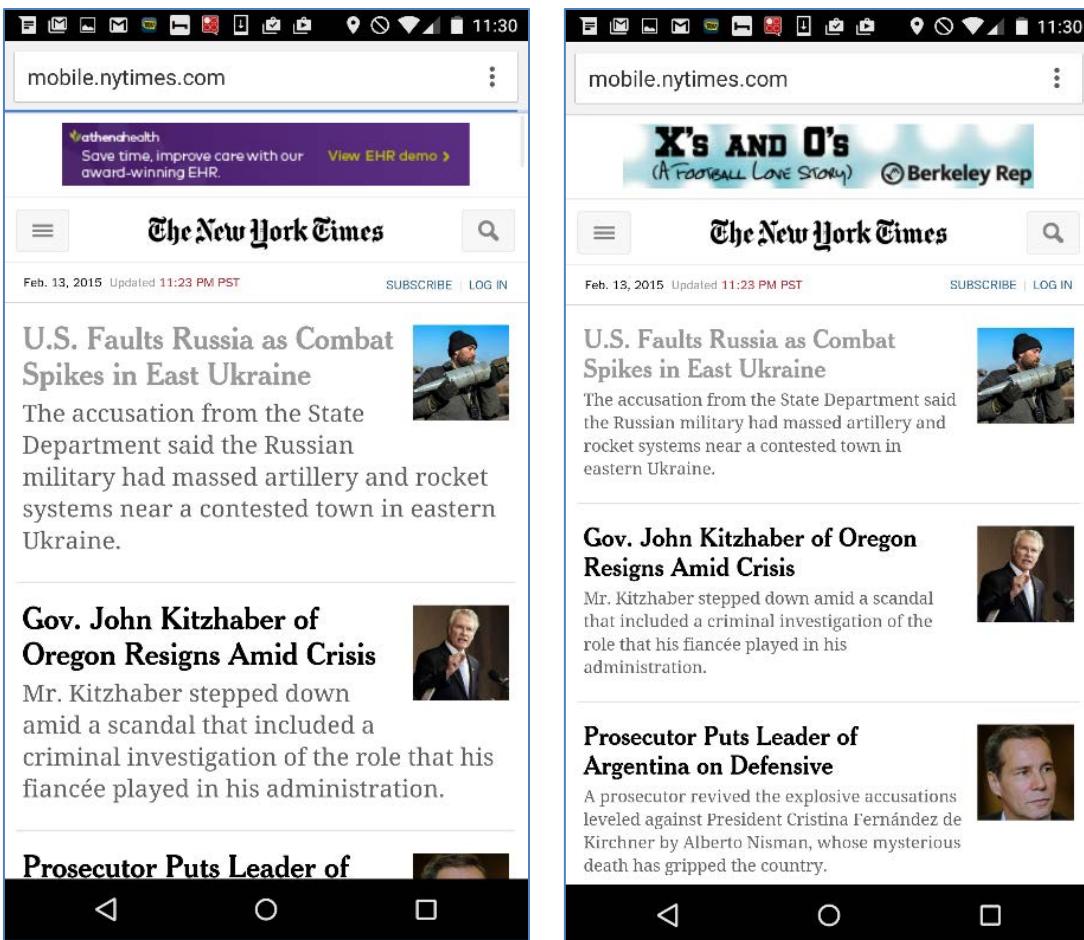
276. Use a font-size default that is readable for a large majority of users.

The font-sizing tools ensure that, if the default font is too small for some users, they can increase it and make it bigger. While we recommend including it on sites that contain a lot of text, most users don't change the defaults. Moreover, many times, the size of the text can be changed for article text, but not for headlines, buttons, icons, or other design elements.

That is why it's really important to start with a good font size throughout your site or app. Because there is a wide variability in screen sizes and resolutions, we cannot recommend an absolute font size that will work on all devices. A relative setting of the font size to "medium" should be a good starting point.



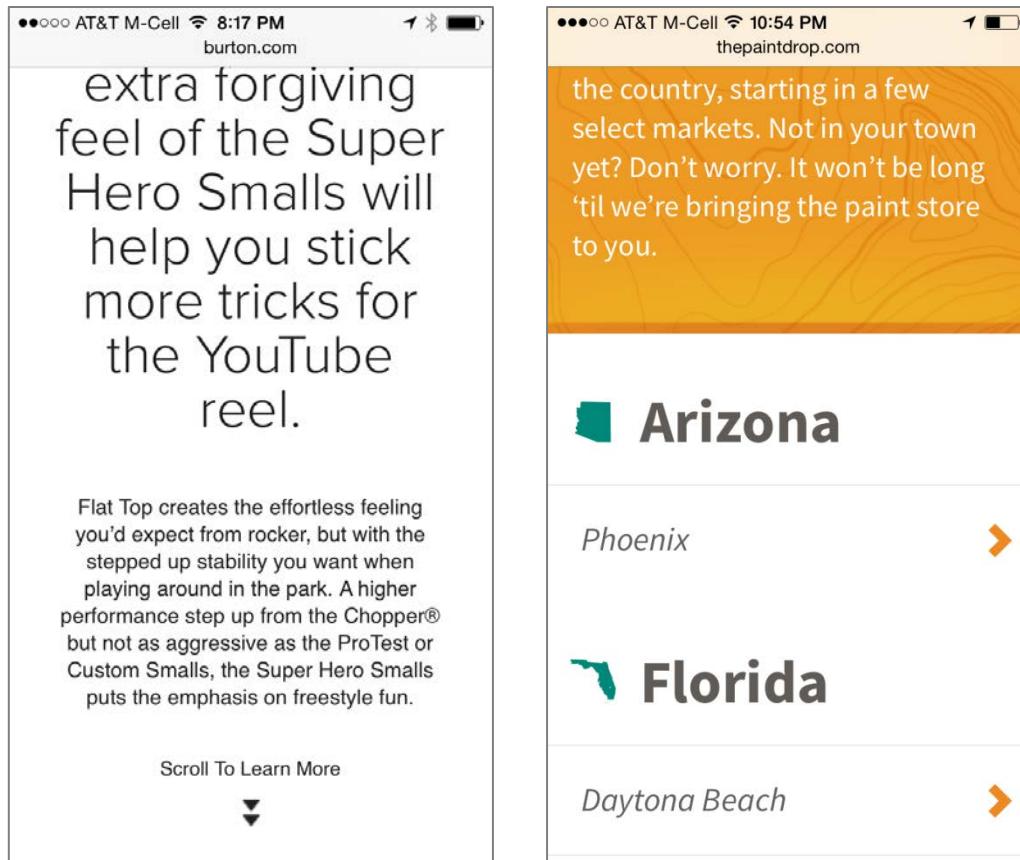
Daily Mail Online for Android: The "X-Large" font did increase the article font size (right), but not the headline-page font.



Nytimes.com increased not only the font of the articles but also the font for the headlines and summaries.

277. Do not use excessively large font.

While too small fonts are not good, fonts that are too large are wasteful: they unnecessarily spend precious screen real estate. In the examples below, the same effect could have been achieved with lower-size fonts.

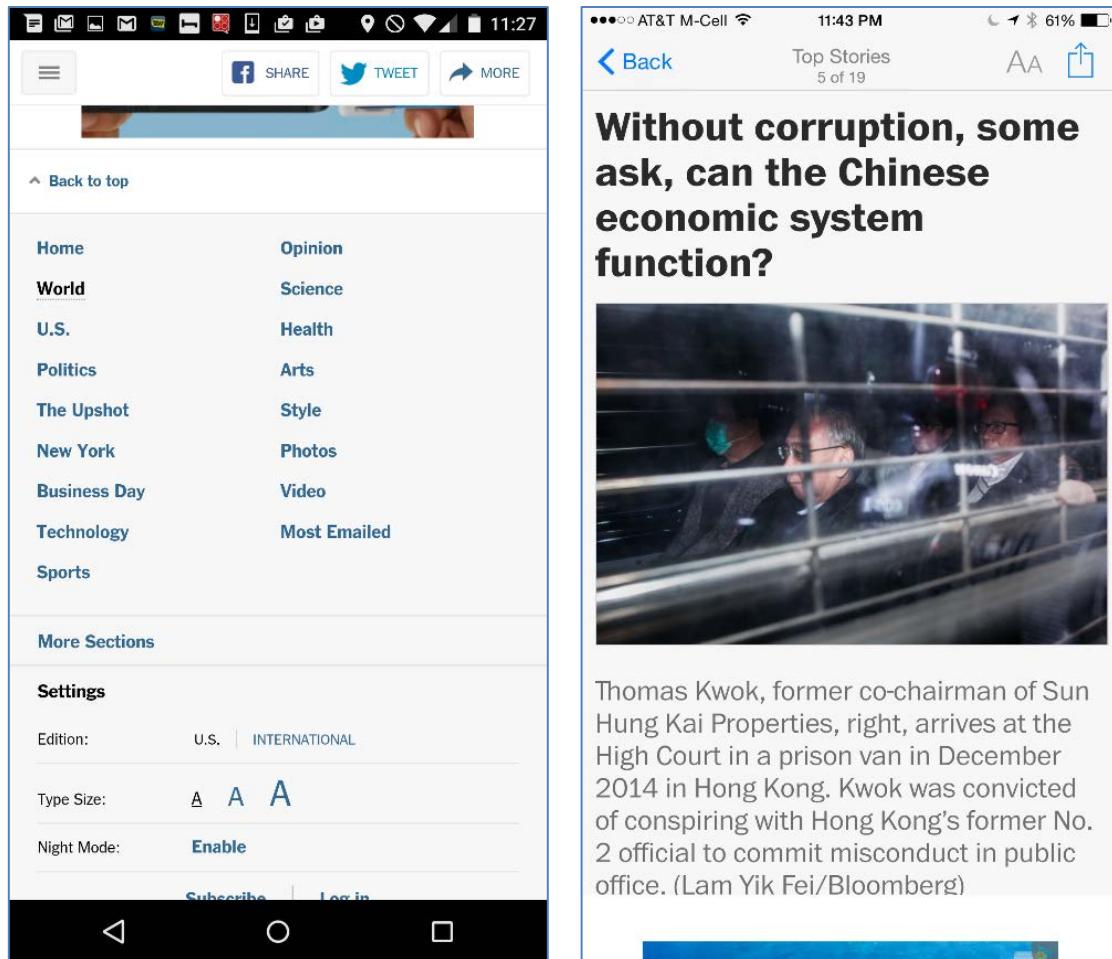


Left: The text in the first paragraph was unnecessarily large. Emphasis could have been achieved using spacing or bolding, but not wastefully large fonts. (Note also that the center-aligned text did not help readability.) Right: The font for the state names was too large.

278. Place the font-size controls in a place where they are easy to find.

The font-size controls are much like the foreign-language option on ATMs and in phone systems: they need to be set first. Thus, it's imperative to have the font control big enough and prominent enough so that users who have trouble with the smaller font can see it.

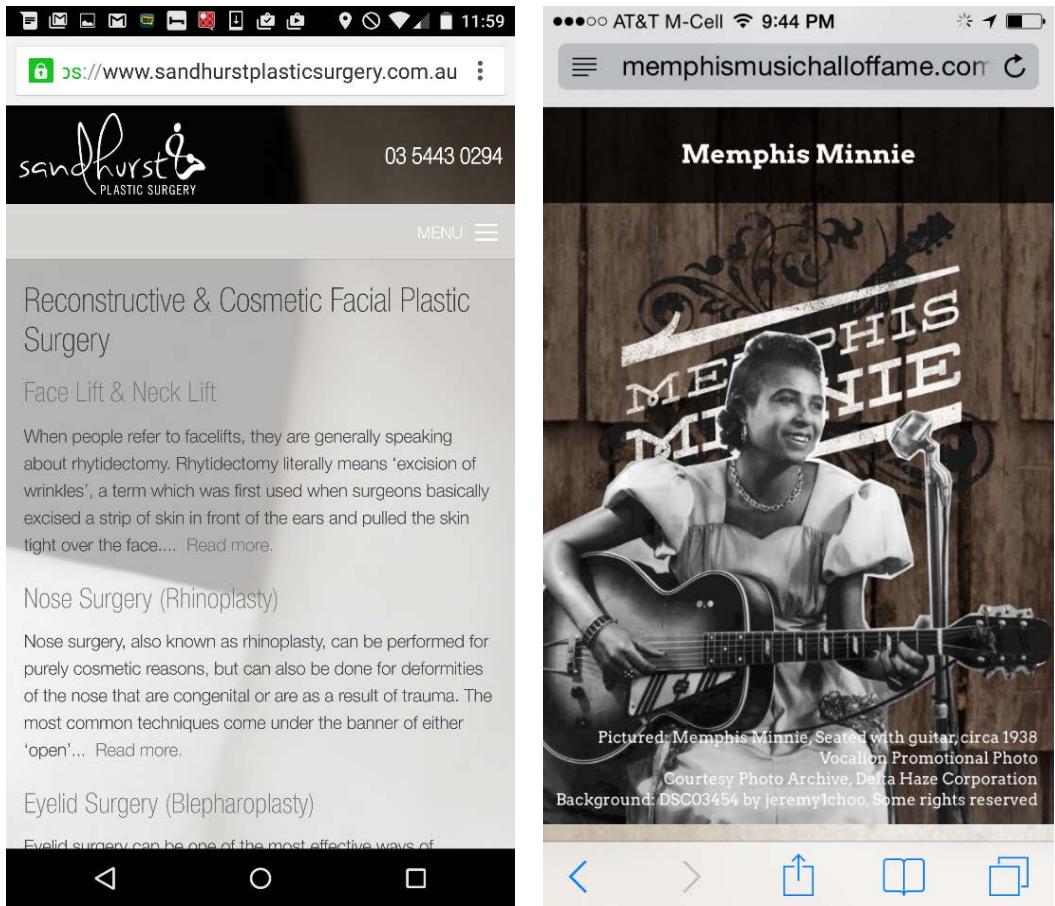
Although NY Times did a good job of accommodating variable font sizes, unfortunately the font controls were placed at the very bottom of the page footer, in an accordion called *Settings*. In contrast, The Washington Post for iPhone correctly placed the font controls at the top of the article, in the page header.



The New York Times mobile site (left) had virtually undiscoverable font controls, placed at the bottom of the page footer. In contrast, The Washington Post for iPhone (right) positioned the font controls at the very top of the page, in a highly discoverable position.

279. Use solid backgrounds.

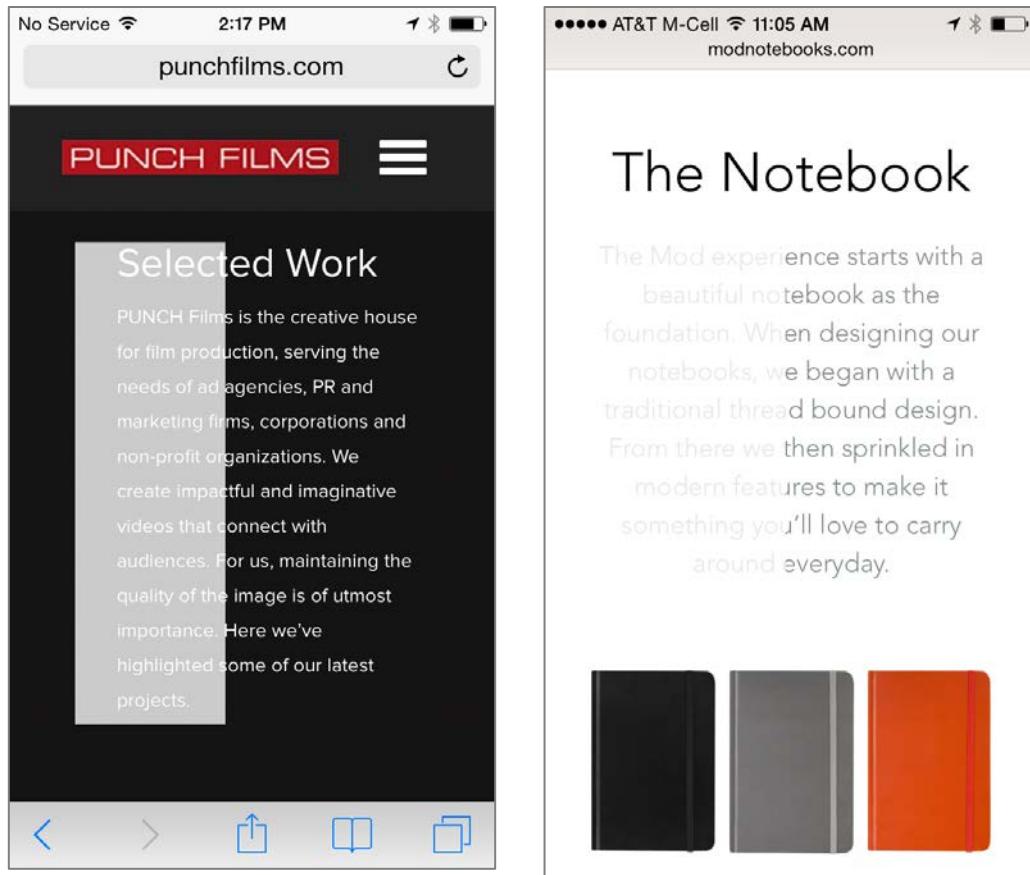
Text on top of an image is hard to read, especially under variable light. We recommend against it in all circumstances. Don't assume that because you've carefully chosen the image it's not going to cause problems —it will. Even under perfect circumstances, the variation in contrast between the background and the text tires the eye and makes reading difficult.



The text on top of the images in these two screenshots was hard to read.

280. Use high contrast between the font color and the page background.

Poor color contrast is never good in a design, but is particularly harmful if users have glare on screens. Remember that mobile phones are used everywhere: indoors and outdoors. In the examples below, we show two screenshots (one with high, one with low contrast) subjected to the same opacity filter, that mimics what we may see when staring at the phone in the sun. It's obvious that the high-contrast text is much easier to read than the low-contrast one, even in the areas where the glare was applied.



A same-opacity filter that mimics glare was applied to a high-contrast text (left) and to a low-contrast text (right). The high-contrast text was more readable than the other one.

Headlines and summaries

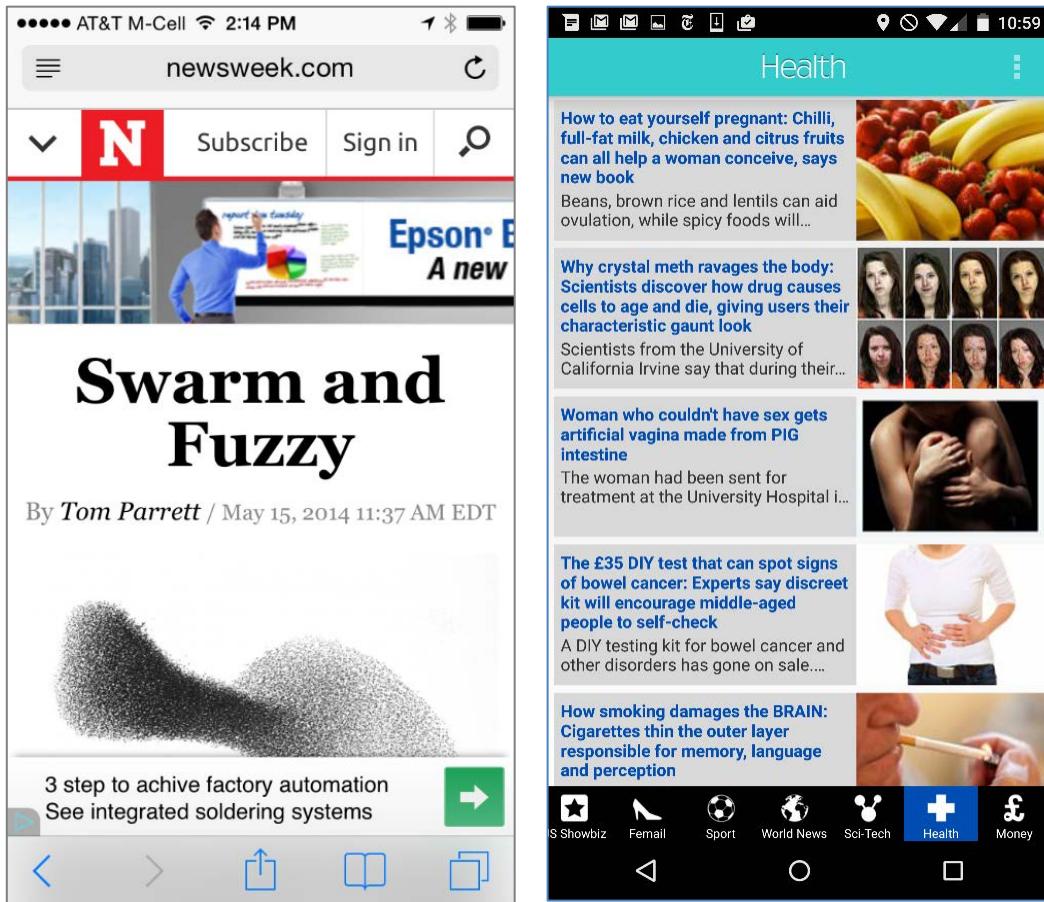
Many websites that contain news stories or other articles have a headline page that enumerates the articles. Sometimes this page also includes a short summary for each article. The summary is highly beneficial to users, as they can get a better understanding of what the story is about and of whether it's worthy of their click.

Article summaries in a list of headlines are an example of layered content, that is of deferring the secondary information to a different page (see guideline 271). Many users get their news by scanning the headlines and the summaries, and never bother to read an article. Because of that, descriptive rather than catchy headlines and "true" summaries, that make sense by themselves and capture the gist of the story, are preferable to sentence fragments or to just echoing the first sentence of the article.

281. Use informative headlines.

Cute headlines may be intriguing, but they often have low information scent. As a result, users may skip tapping on them to avoid risking content that is not

up to their expectations. On mobile, and especially if the headline is going to appear by itself either on a headline page or even on an article page (like in the Newsweek screenshot below), it's important to create headlines that correctly convey the gist of the story.

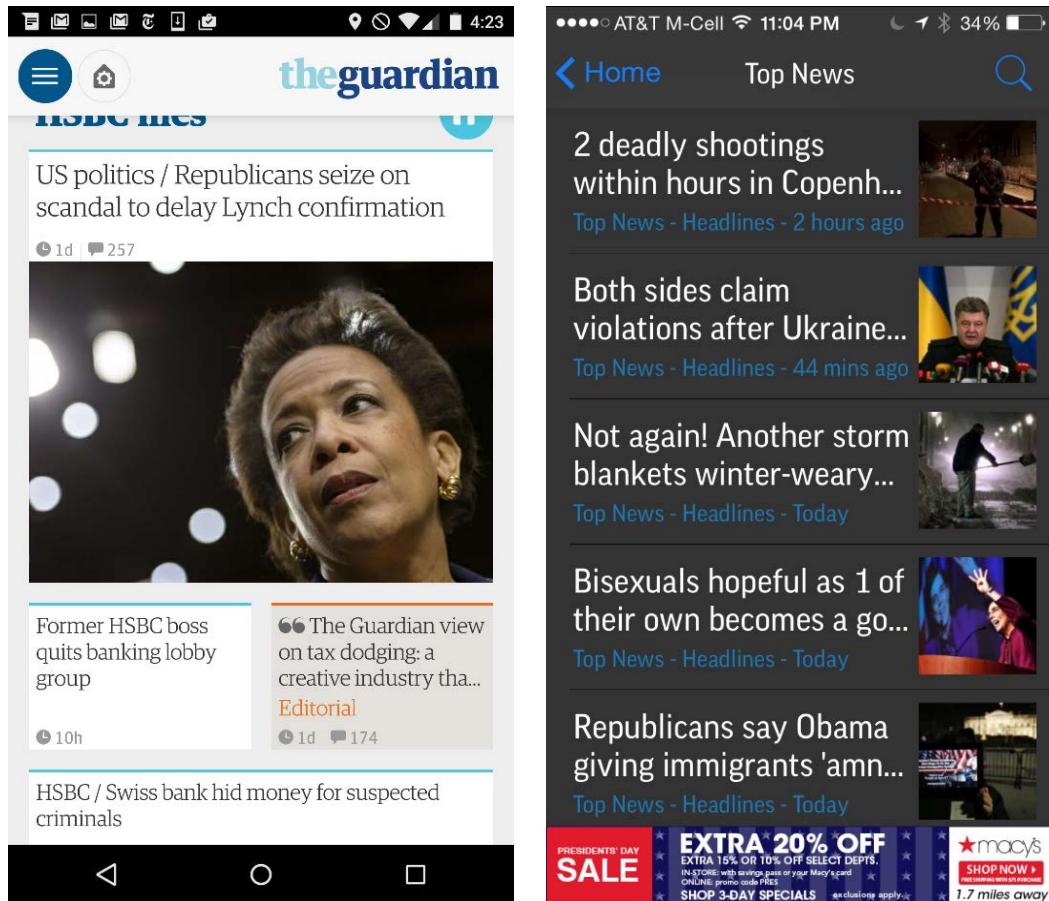


Left: Some of the Newsweek headlines were too vague and did not describe the content of the article. Right: In contrast, the headline of the Daily Mail Online for Android functioned almost like summaries for the articles. (Another, unfortunately truncated, "summary" followed the headline on the headline page; given the amount of detail in the headline itself, that additional summary was probably unnecessary.)

282. Do not truncate headlines or summaries.

Complete summaries and headlines tell people what the article is about and whether it is interesting to them; by truncating the headline or the summary, the amount of information is obviously reduced. Truncated sentences mean impaired information scent (see guideline 272). Remember: people use information scent to guide themselves on the web and decide what they want to read. It's better to give users an accurate, quick impression of the topic of the article than to run the risk of having that story ignored.

Even reputable institutions are guilty of truncating headlines; it's a fatal mistake that stems from the fact that, instead of having different-length headlines and summaries suitable for different media and contexts, many news sources stick with a single headline that may be just too long for mobile. Don't fall into that trap.



Some of the headlines were truncated in The Guardian app for Android (left) and the AP Mobile app for iPhone (right).

This type of logic works in general for situations where people must choose among a set of available links. For instance, if they need to pick a product, it's important that they see the full name of the product, rather than a truncated one that may leave out relevant information.

283. Frontload headlines and summaries with keywords.

As discussed in the beginning of this section, people scan online, and the words at the beginning of a paragraph or at the beginning of a headline have a higher chance of being read than those in the middle. In the Children's Museum page below, many of the exhibit summaries started with words that carried little information; that, combined with the truncated paragraphs, made it difficult to tell what the different offerings were about.

**children's museum
PITTSBURGH**

MENU

Exhibits

There's lots of creative, fun and educational exhibits to explore throughout the Children's Museum!

NOTE:
The Studio and MAKESHOP close at 4:30 pm daily.

children's museum PITTSBURGH
For more information, please contact exhibits@pittsburghkids.org or (412) 322-5058, ext. 229 www.pittsburghkids.org

Coming Soon

The Children's Museum of Pittsburgh and The Eric Carle Museum of Picture Books...

[Read More](#)

XOXO

Now Open [Read More](#)

Attic

Tackle illusions, perception and phenomena. [Read More](#)

Garage

Take our real SmartforTwo car for a test drive, then fill the gas tank and wash the win... [Read More](#)

MAKESHOP®

MAKESHOP® is a space for children and families to make, play and design using "... [Read More](#)

Nursery

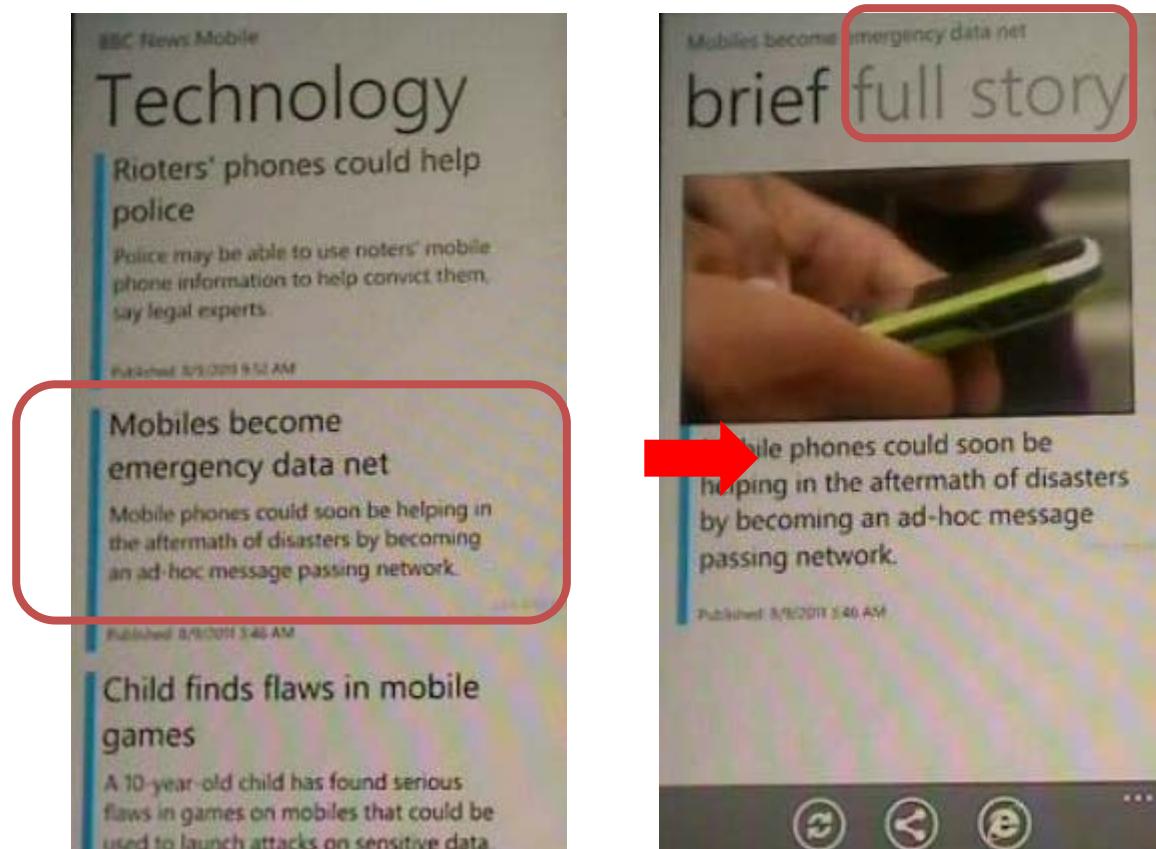
You can enjoy both quiet, thoughtful play and active, creative play in this serene space till... [Read More](#)

Children's Museum in Pittsburgh had poor exhibit descriptions: some were truncated, some had almost no information (e.g., the one for XOXO), and some started with words that carried little information (for instance, the description for "Nursery" started with "You can enjoy both quiet..." and the description for "MAKESHOP" started with the title word "MAKESHOP").

284. Don't repeat summaries from the headlines page on the article page.

When the user has clicked on a new story, they expect immediate gratification, and they shouldn't have to work to get to new content; rather, they should get it immediately. Therefore, information should not be repeated from one page to another.

One participant was using the BBC News application for Windows Phone 7. He noted that one of the things he didn't like about the app was that, when you clicked on a headline, it repeated a "brief" that had been already included on the headline page. The user thought that was "a waste of space." To read the article, the user had to click one more link on the "brief" page. Here are some screenshots from the recording of that session:



BBC News app for Windows Phone: The headline page contained the article title plus a summary. When clicking on the article, the same summary was shown once more. Users had to click on *Full story* at the top to actually read the article.

The screenshot shows a news feed on the SB Nation mobile site. At the top, there's a banner with a dark background and white text: "GET THE MOST RECENT SCORES" and "GO TO THE SB NATION SCOREBOARD". Below this, there are three main article cards:

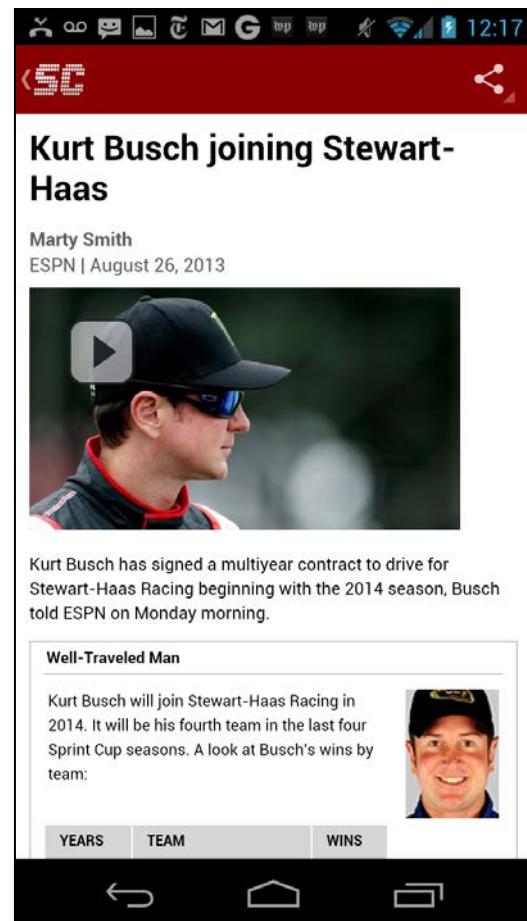
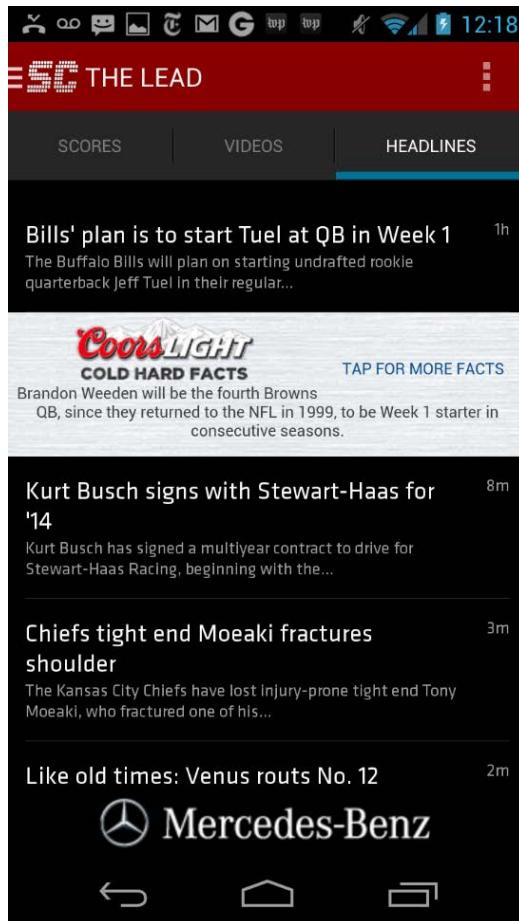
- Behind the scenes at the All-Star fashion show** by Sarah Kogod (Feb 14, 2015, 9:53a). A small thumbnail image shows a person in a costume.
- NBA Relentless positivity pushes George through rehab** by Mike Prada (Feb 14, 2015, 10:30a). A thumbnail image shows Paul George in a white shirt, holding a basketball.
- DeMarcus Cousins fires back at Charles Barkley** by Ricky O'Donnell (Feb 14, 2015, 3:34p). A thumbnail image shows DeMarcus Cousins in a blue jersey.

The screenshot shows a news feed on the SB Nation mobile site. At the top, there's a banner with a dark background and white text: "Relentless positivity is pushing Paul George through his rehab" by Mike Prada (@MikePradaSBN) on Feb 14, 2015, 10:30a. Below this, there's a large image of Paul George shooting a basketball. The caption reads: "A horrible leg injury threatened to derail Paul George's entire career. Now, he's on the cusp of an early comeback, thanks to a sunny disposition that has carried him through his rehabilitation." At the bottom, there's a blue button with the text "SHARE ON FACEBOOK (2527)".

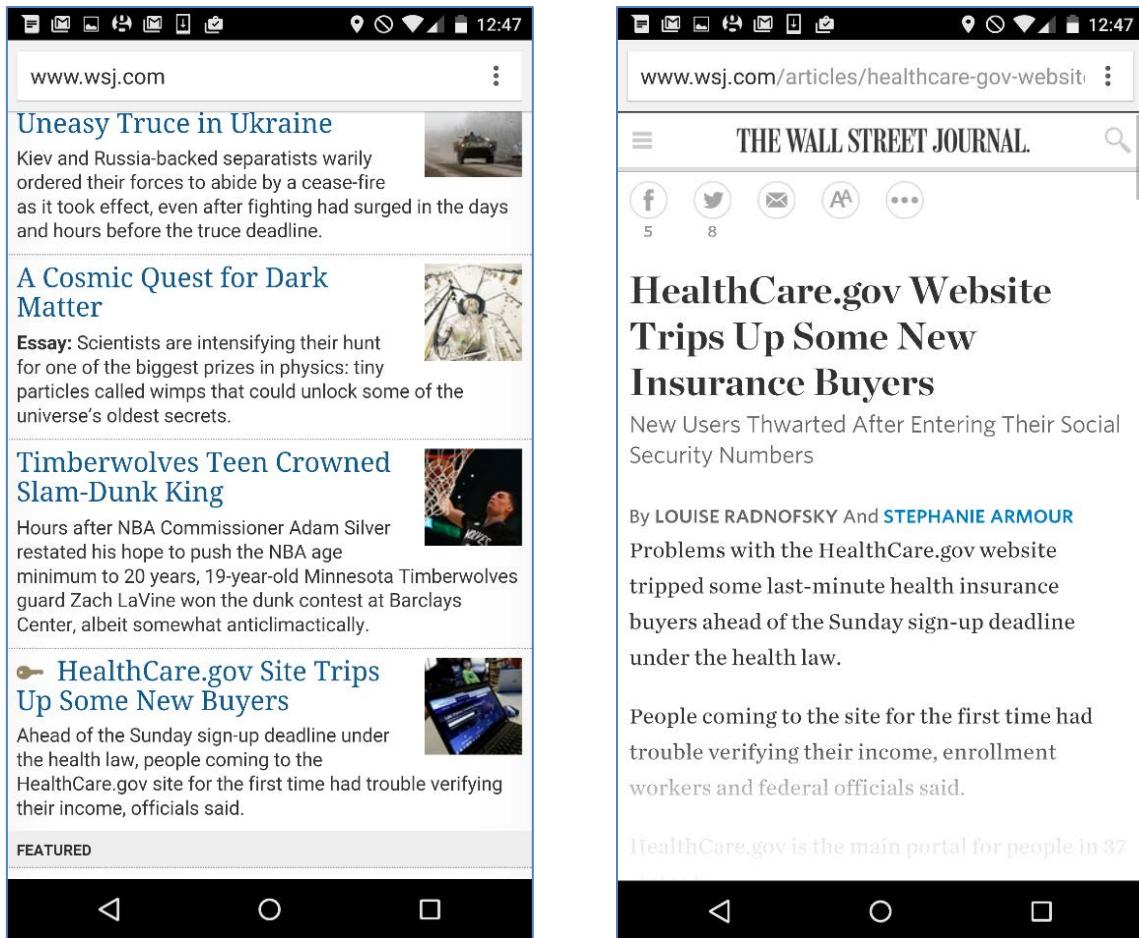
SBNation.com: Although the site did have separate summaries for the articles, the summaries (together with the article title) were repeated on the article page; users had to scroll past the first article page to read new content.

285. Do not use the first few sentences of the article as a summary on the headlines page.

Sometimes sites cheat, and instead of presenting a true summary of the article, they show the first few sentences of the article. We recommend against that, to avoid repeating information: users will have to look again at those first sentences when they click on the article link and see the whole article in front of them. Plus, the first few sentences rarely include the gist of the article.



An older version of ESPN Score Center for Android used the first few sentences in the article as a “summary” on the headline page.



The Wall Street Journal used true summaries on the headline page; these were different from the first paragraph of the article.

Related Content

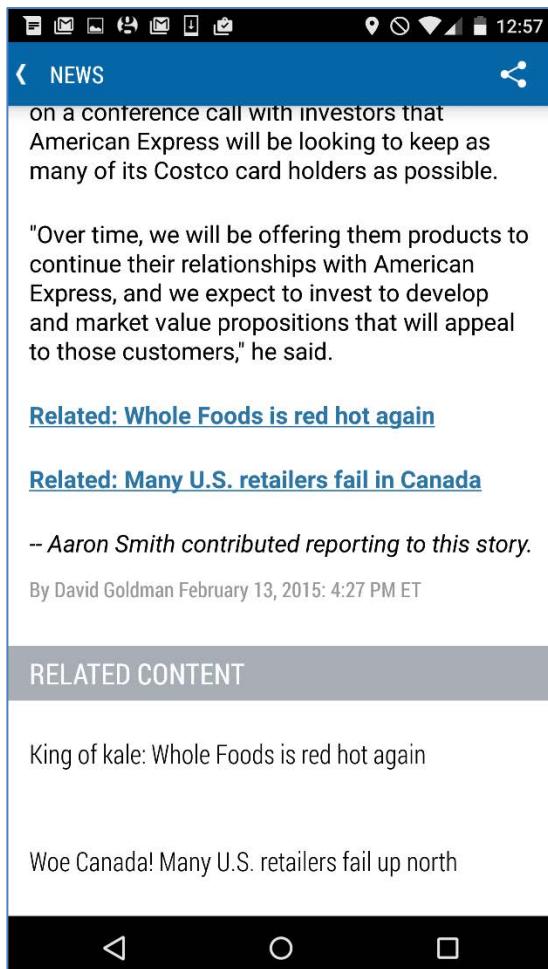
Well-chosen related-content links can be a major way of navigation and of keeping the user on your website⁶⁶ or in your app. On mobile, they also can save extra clicks back to a “headline-only” page: users can directly select another article from the current page. Related content works well when: (1) users find it easily, that is, when it’s placed immediately at the end of the article; (2) when it is indeed relevant and connected with the current articles.

286. Use links to related content to help the user navigate more quickly between similar topics.

⁶⁶ See also Hoa Loranger. “Related Content Boosts Pageviews, When Done Right.” <http://www.nngroup.com/articles/related-content-pageviews/>

287. Make sure that the related links are relevant to the article.

When the “related” content is not relevant to the current topic, users lose trust in the site’s ability to make intelligent suggestions and start ignoring them systematically. So don’t just add in related content for the sake of filling up that section with links or promoting content that you think is important.

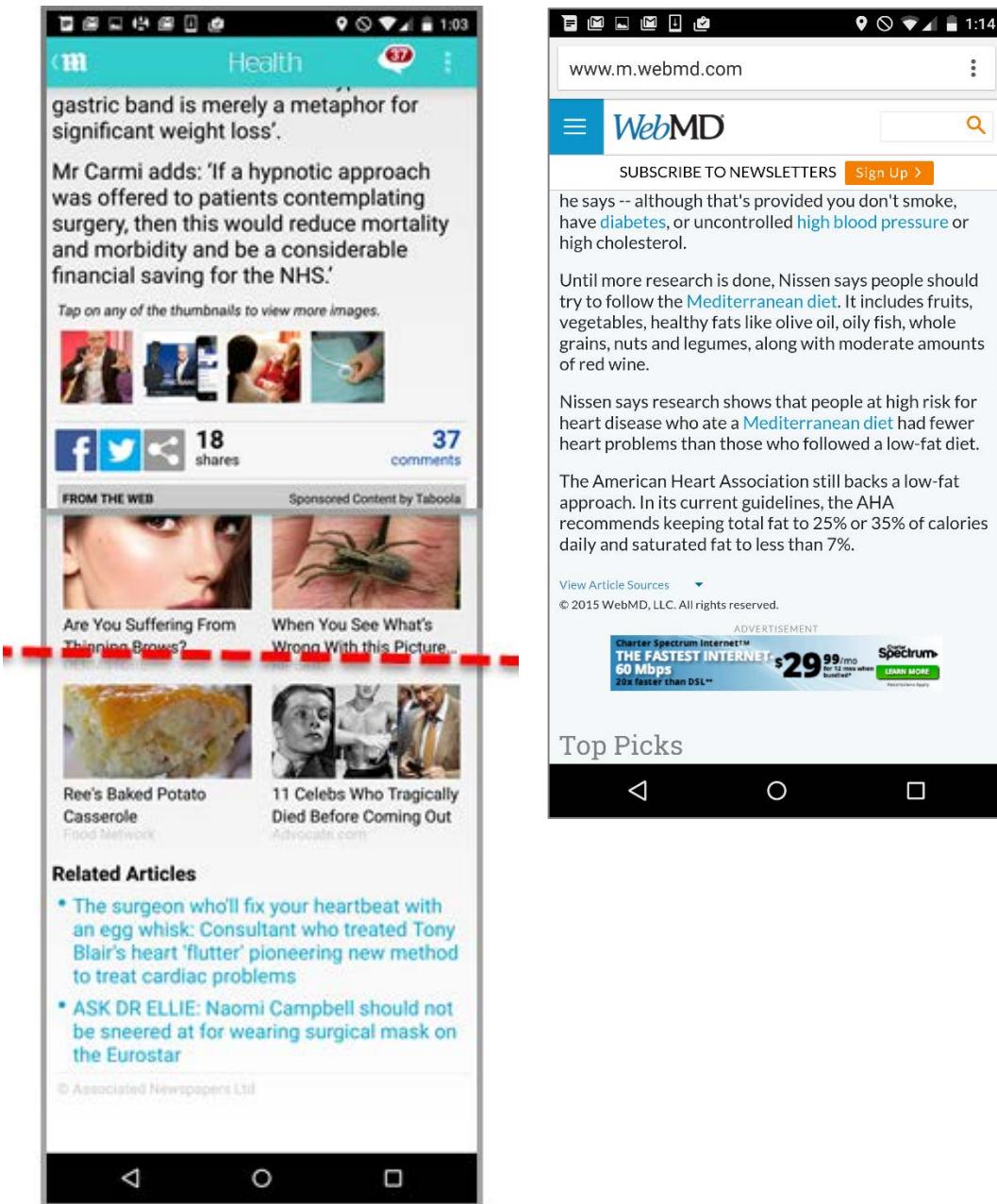


CNN Money for Android: The two related articles seemed completely unrelated to the topic of the current article, which was about American Express discontinuing its Costco credit card.

288. Place related links in close proximity to the end of the article, to make sure that people do not miss them.

289. Do not insert ads between the article and the related content.

If other content such as ads are inserted between the end of the article and the related content, people lose interest or think that there is no more content on the page. To prevent that from happening, place related links immediately after the end of the article.



Left: Daily Mail for Android (left) and Web MD (right) placed sponsored content between the article and the related links, making it less likely for users to scroll down.

Miniature Information Architecture (Mini-IA)

A **mini-IA** of an article or web page refers to how the information on that page is structured. For instance, one way to structure the information would be to place all the different sections of a page one under the other and give the user no other alternative than scrolling sequentially through all the content with the hope that eventually something of interest will come up.

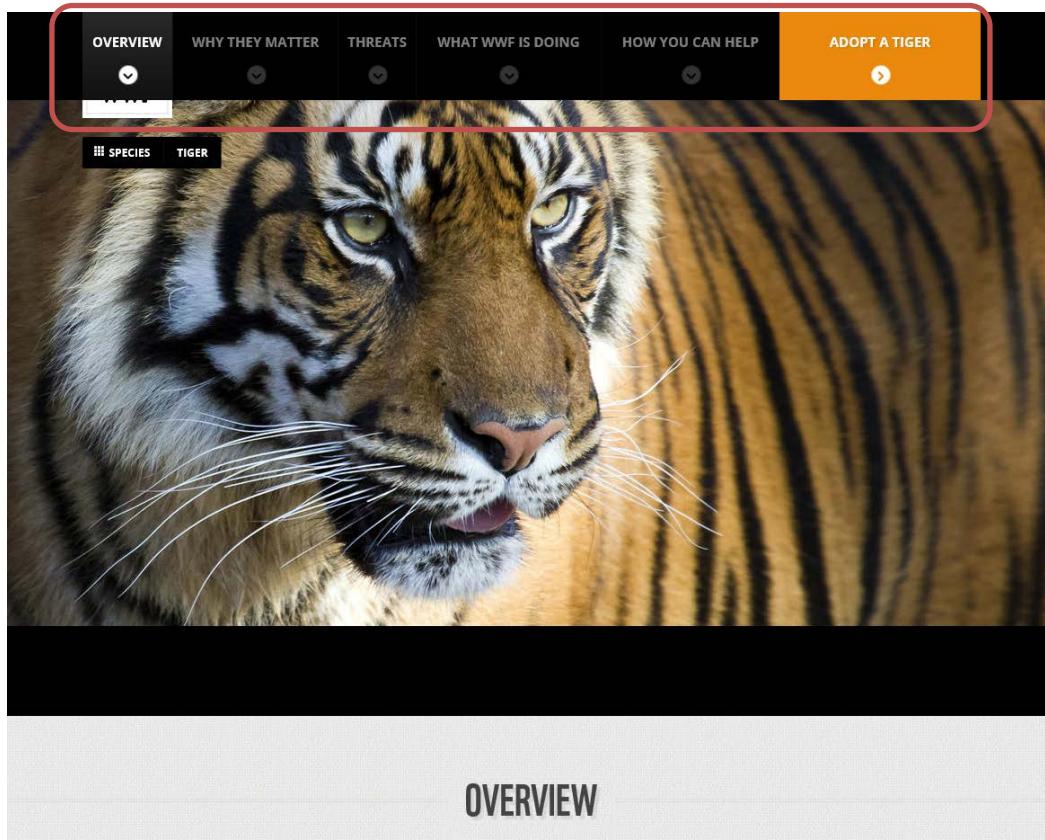
As you can guess, this type of structure (or, better said, lack of structure) is very inefficient on mobile: since the screen is small, there is little chance that users will get a glimpse of the different types of content available unless the page is short or they scroll.

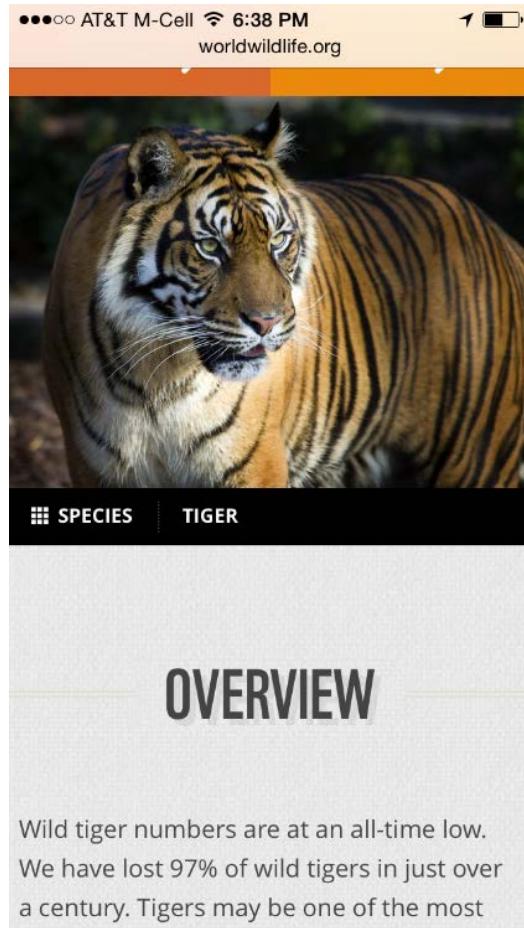
The alternative is to tell people beforehand what the page is about. This is akin with having a table of content. There are three main benefits of such a table of contents:

1. It gives users direct access to a section of interest.
2. It tells users what the page contains and whether the type of information is likely to be relevant for their goals.
3. It helps users form a mental model of the page and the site.

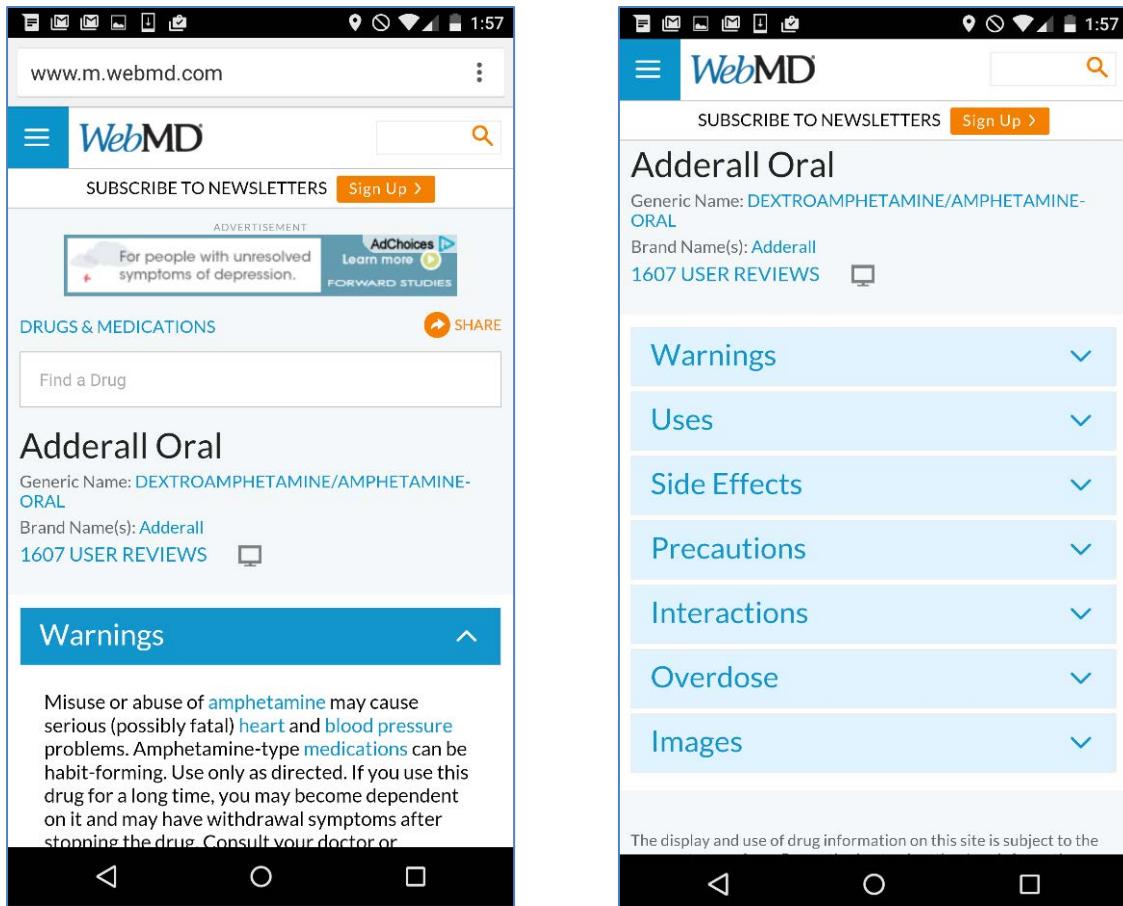
Like a real site IA, a mini-IA or an in-page table of contents can be implemented in several ways:

- **Accordions**, when closed, can let users see the skeleton of the page, while also allowing them to expand sections of interest in situ.
- **Jump links** offer the same advantages as accordions, but they also preserve the *Back* button (see guideline 99 for a discussion of *Back* problems with accordions).
- **Page menus** have a tighter footprint, but also hide information and can get mixed up with main-navigation menus (see guideline 217).
- **Navigation bars** (usually placed at the top of the page) can work well if there are only a few sections available.





World Wildlife Fund: The desktop version of the responsive page (top) included a mini-IA at the very top of the page, but this was left out on mobile. The mobile page showed the different page sections one under the other. Mobile users had no way of knowing what type of content the page included — for instance, they could not have guessed that the page included a description of the threats that tigers faced or a link to donate. The top *Overview* section did not entice to scroll down and find that information.



On drug pages, WebMD showed a mini-IA of the page implemented through accordions (right). Unfortunately, by default the first accordion was expanded and prevented users from getting a quick glance (without scrolling) at the structure of the page. Some users may have thought that the page was only about drug warnings and left before scrolling down.



Epicurious for iPhone used a navigation bar for the in-page table of contents; the categories in the bar were jump links to the different page sections.

- 290. Unless your content has a sequential structure (e.g., a piece of fiction or a news article), all pages can benefit from an easily accessible table of contents.**

Pagination

In the section *View All, Pagination, and Infinite Lists* we explained why mobile users don't like content that spreads over multiple pages. You should use pagination only if your content is too long to be managed on a single page (for instance, it would be impractical to have all the content of a book on a single page).

- 291. Avoid splitting articles in many pages.**

It's better to show articles on a single long page than to split them in many short pages, to minimize the load time between pages. If the article contains many images, it's ok to have the images as a separate feature (e.g., associated slideshow).

On an older version of NBC's mobile page, an episode recap from "Days of Our Lives" was split into 41 pages. Each page contained a big picture and one paragraph of text. It would have been better to fit all the text and 1-2 pictures on a single page, and then include a link to additional pictures.



NBC.com: The episode summary was split into several short pages.

292. Consider including all the images in a slideshow on a single page.

A new trend of including all the photos of a slideshow on a single page (instead of having users swipe through them) was received well by users: people appreciated that they did not have to wait for a new photo to load.

**** AT&T M-Cell 2:26 AM 19%

< Back Jennifer Aniston - Red Carpet Watch

RED CARPET WATCH

Jennifer Aniston

EMMA STONE
PATRÍCIA ARQUETTE
MARION COTILLARD

Ms. Aniston, who plays a woman in chronic pain in "Cake," is more closely associated with her hairdresser, Chris McMillan, than any wardrobe stylist.



At the "Horrible Bosses 2" premiere in Los Angeles, the actress wore a lace bustier and short skirt by the Lebanese designer Zuhair Murad.



NYTimes for iPhone placed all the images in a slideshow on a single page; subsequent images loaded in the background as users looked at the ones before.

Images

Images are worth a thousand words and have their place on mobile websites. As one user put it:

"The thing with websites is that you're always impatient, you don't want to read... I think really picture icons are so much more effective — you want to just see, and stab, and move quickly, and if you've got a good speed, like me at the moment, scrollbars and reading your way through a scrollbar takes seconds..."

With images, we encounter the same recurring theme that we've seen so far in relation to search boxes and navigation: we need to counterbalance the benefit that the images bring to users with the costs they incur (in space and speed). Therefore, each time you consider including an image on your site, we recommend that you seriously evaluate how much information load that image carries.

293. Include images on your website only if they add meaningful content. Do not use images for decoration purposes only.

It's ok not to use images if they do not add extra information; there is no reason to add an image just to prettify the page.

F.A.A. Rules Would Limit Commercial Drone Use

By SCOTT SHANE
FEBRUARY 15, 2015

In an attempt to bring order to increasingly chaotic skies, the [Federal Aviation Administration](#) on Sunday [proposed long-awaited rules](#) on the commercial use of small [drones](#), requiring operators to be certified, fly only during daylight and keep their aircraft in sight.

The rules, though less restrictive than the current ones, appear to prohibit for now the kind of drone delivery services being explored by [Amazon](#), [Google](#) and other companies, since the operator or assigned observers must be able to see the drone at all times without binoculars. But company officials believe the line-of-sight requirement could be relaxed in the future to accommodate delivery services.

The proposed regulations would cover only nonrecreational unmanned aircraft weighing up

Find a plan that's right for you.

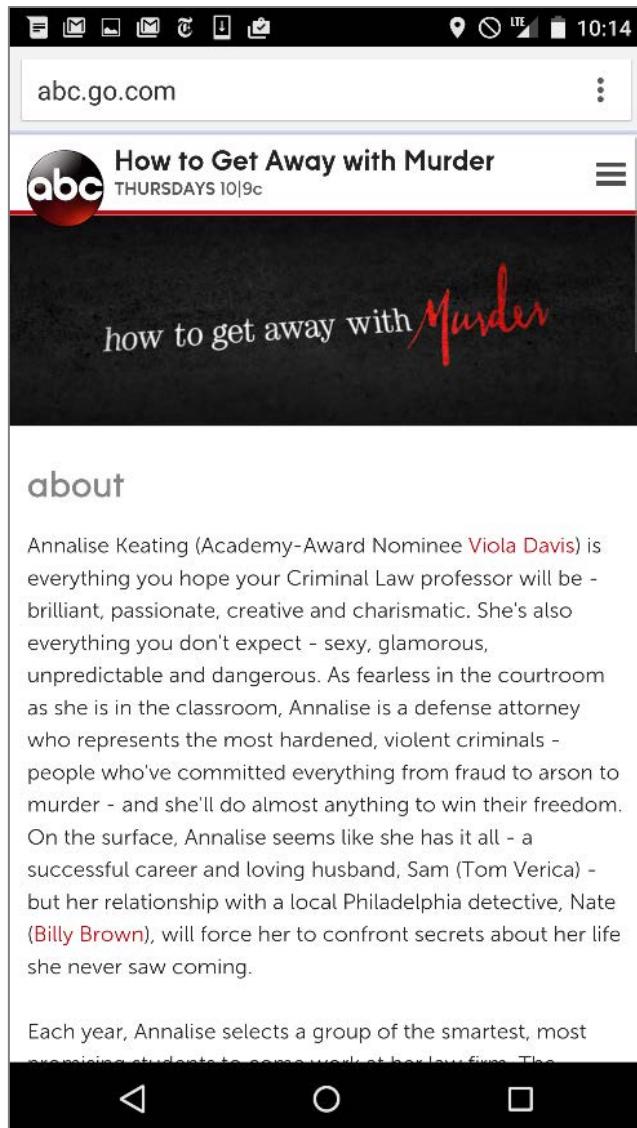
Entrepreneur

Up to 2,000 subscribers

Send 12,000 emails to 2,000 subscribers for free. No contracts, and no credit card required. It's free forever.

Learn More

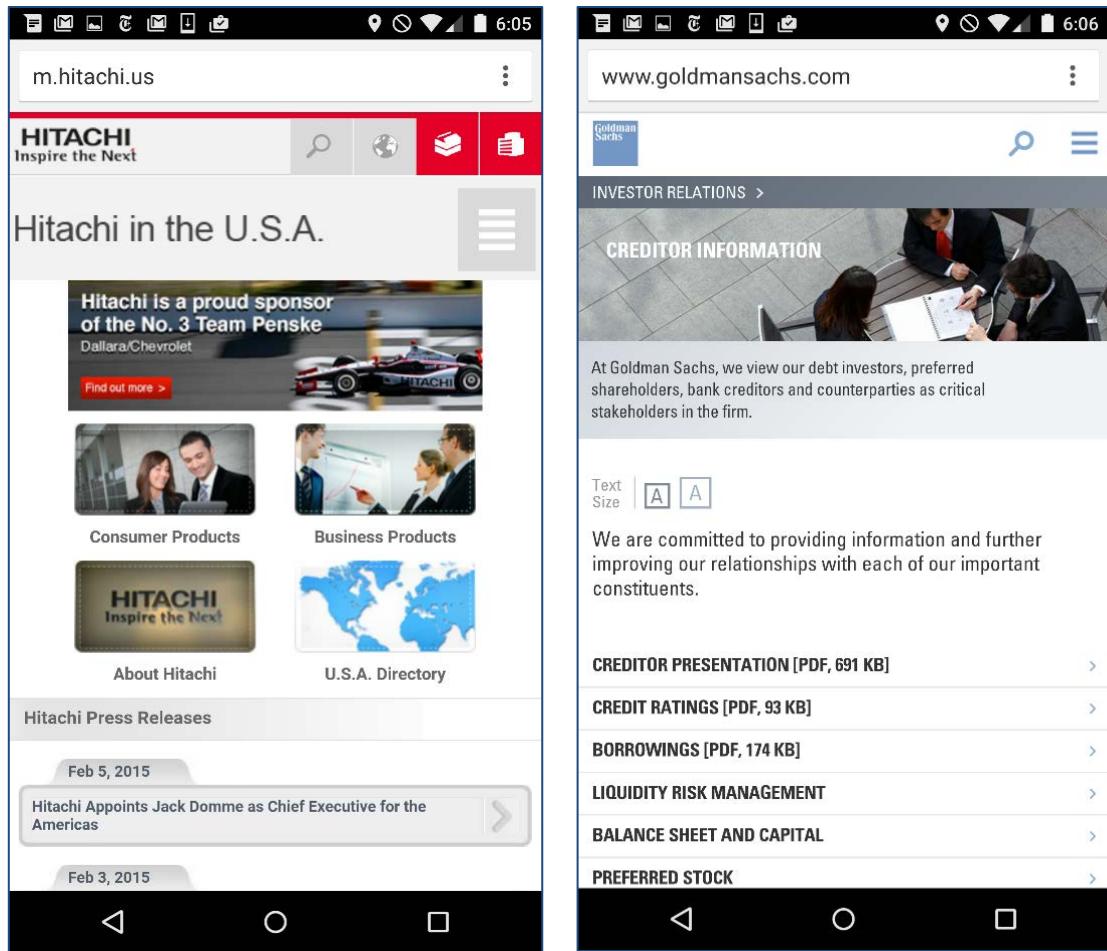
Left: On the New York Times website, some articles did not include any images. Right: On MailChimp's website the images added no meaning to the different plans' information.



ABC used images for show titles. These were completely unnecessary and duplicated the text title of the page.

294. Avoid stock photography or photography used for decoration only.

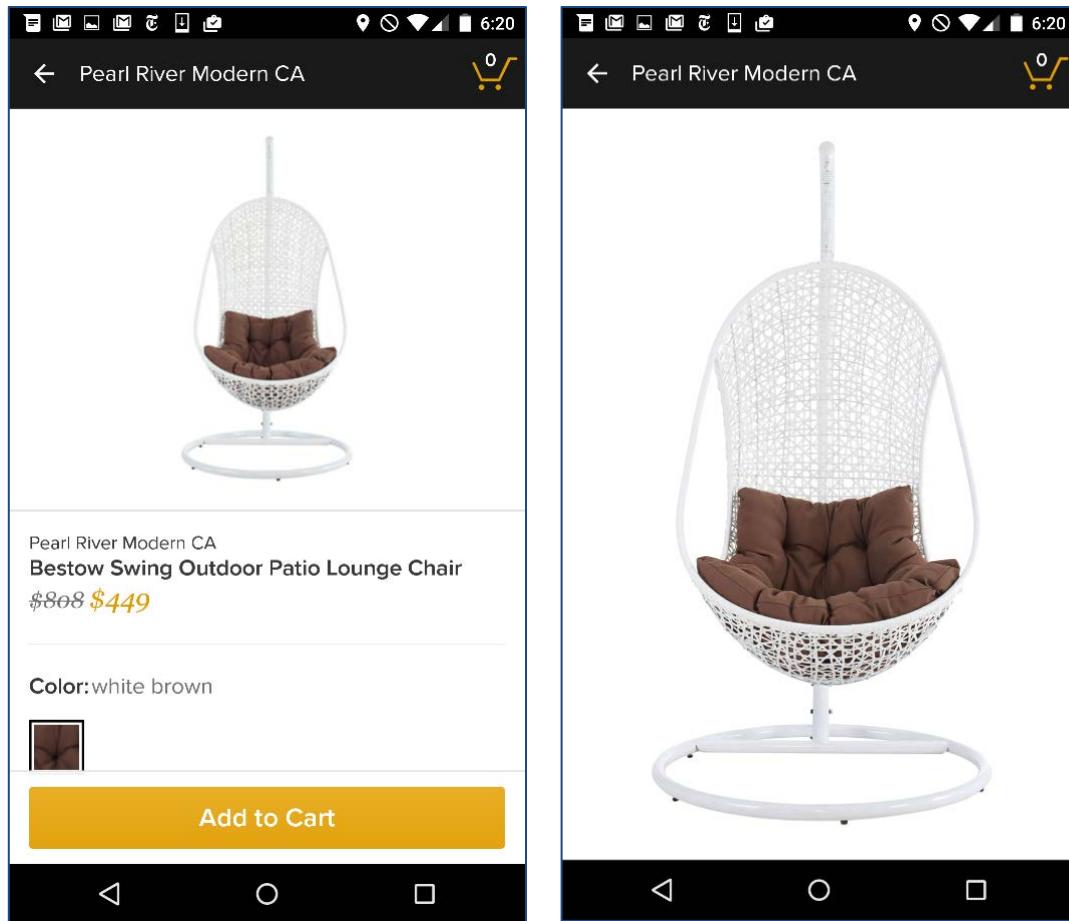
Stock photography is rarely appreciated by users on any medium, as it looks obviously fake. We recommend that you stay away from it on mobile; its only effect is that it's going to slow down your site.



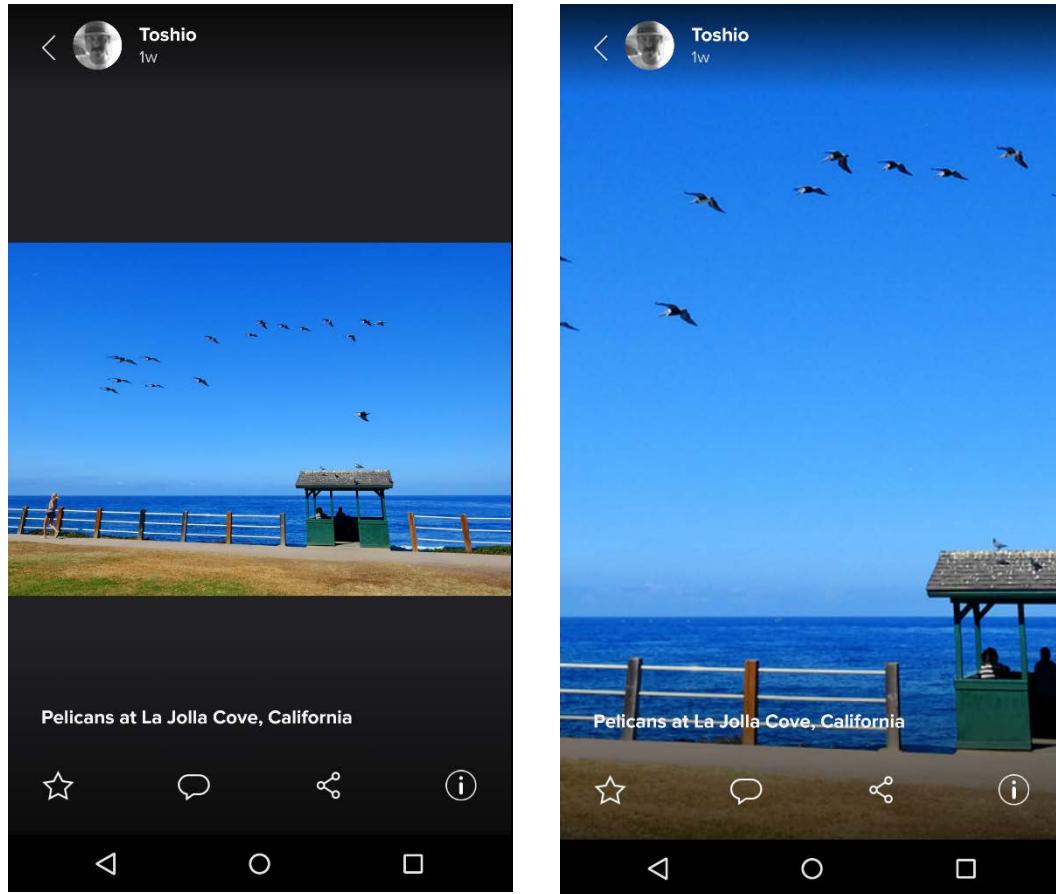
Hitachi and Goldman Sachs used stock photography with low information content to decorate their sites.

- 295. Do not use image sizes that are bigger than the screen. The entire image should be viewable with no scrolling.**
If the image is too big and does not fit on the screen properly, the user cannot get the big picture.
- 296. For cases where users are likely to need access to details in a picture, either (1) add a separate link to a higher-resolution, full screen variant, or (2) allow people to zoom into the picture to check details.**

Typical instances of these situations are shopping sites and sites for photographers. In both these cases, details are important. However, they are important only after the user has established a certain interest level in the picture — that is, they have decided that the product may be worth buying or that the photograph is interesting. The majority of the users are unlikely to be interested in the details of any one particular picture, so for them, waiting for the high-size image to download is a nuisance.



Gilt for Android provided a full-screen, zoomable image for each product.



Flickr for Android provided screen-sized images that could be zoomed in to get more details.

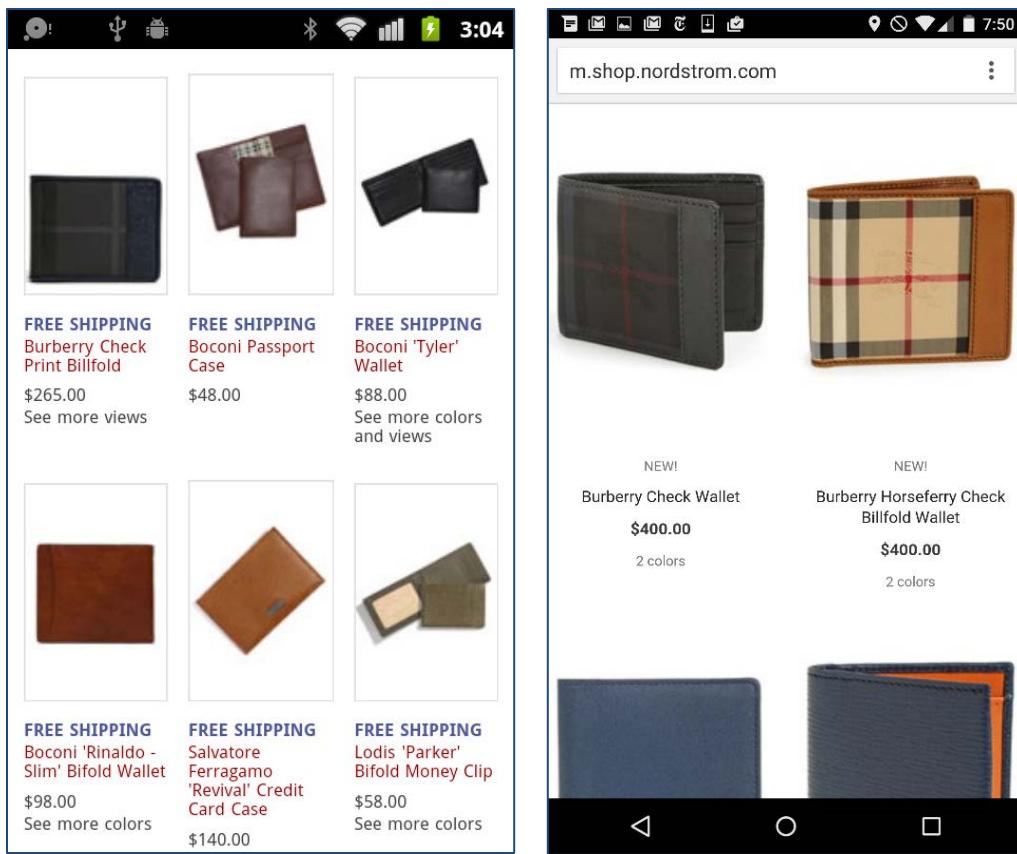
297. Thumbnails should be as big as need to ensure that users can distinguish what the picture is about.

For thumbnails, we need to compromise between size (which affects the speed of loading) and information conveyed (if the picture is too small, people cannot understand what it is about).

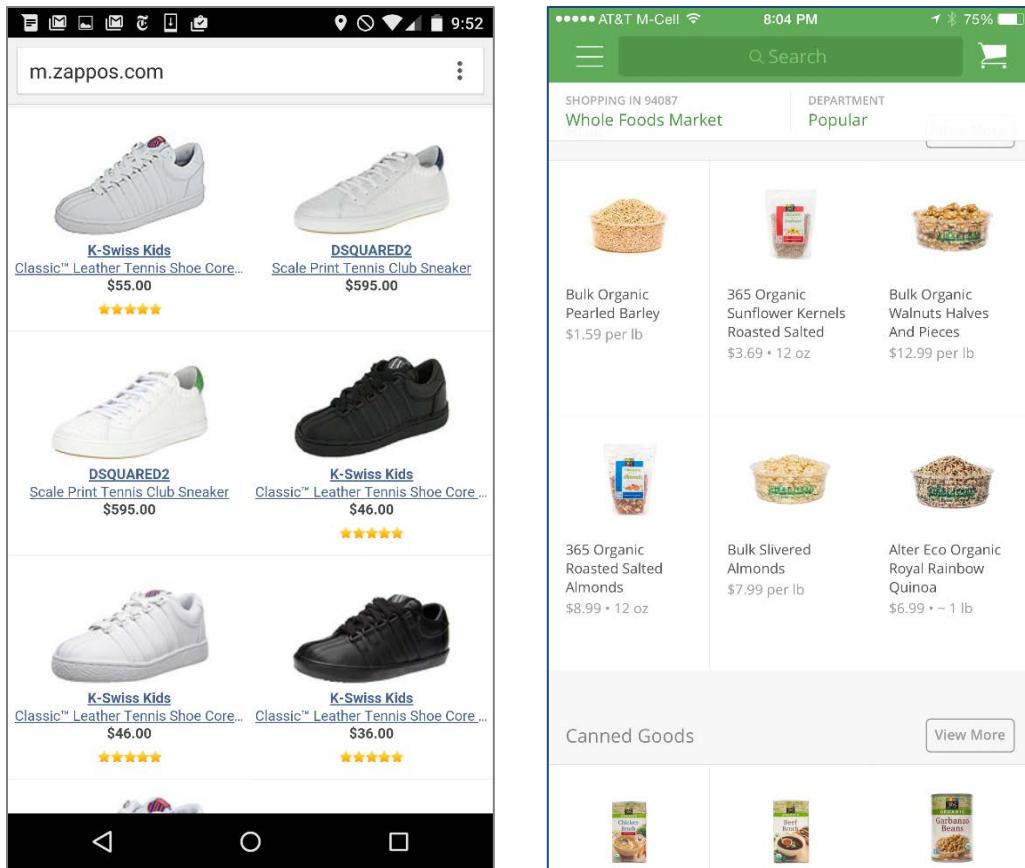
Our users had difficulty on an older version of Nordstrom's website because they were not able to see wallets well enough in a thumbnail to actually decide if they were interested in the product. One of our participants noted:

"Images are too small... You can't see that well".

In contrast, Zappos had really good thumbnails. Users often commented that they loved the quality of images both on Zappos' mobile website and in their app.



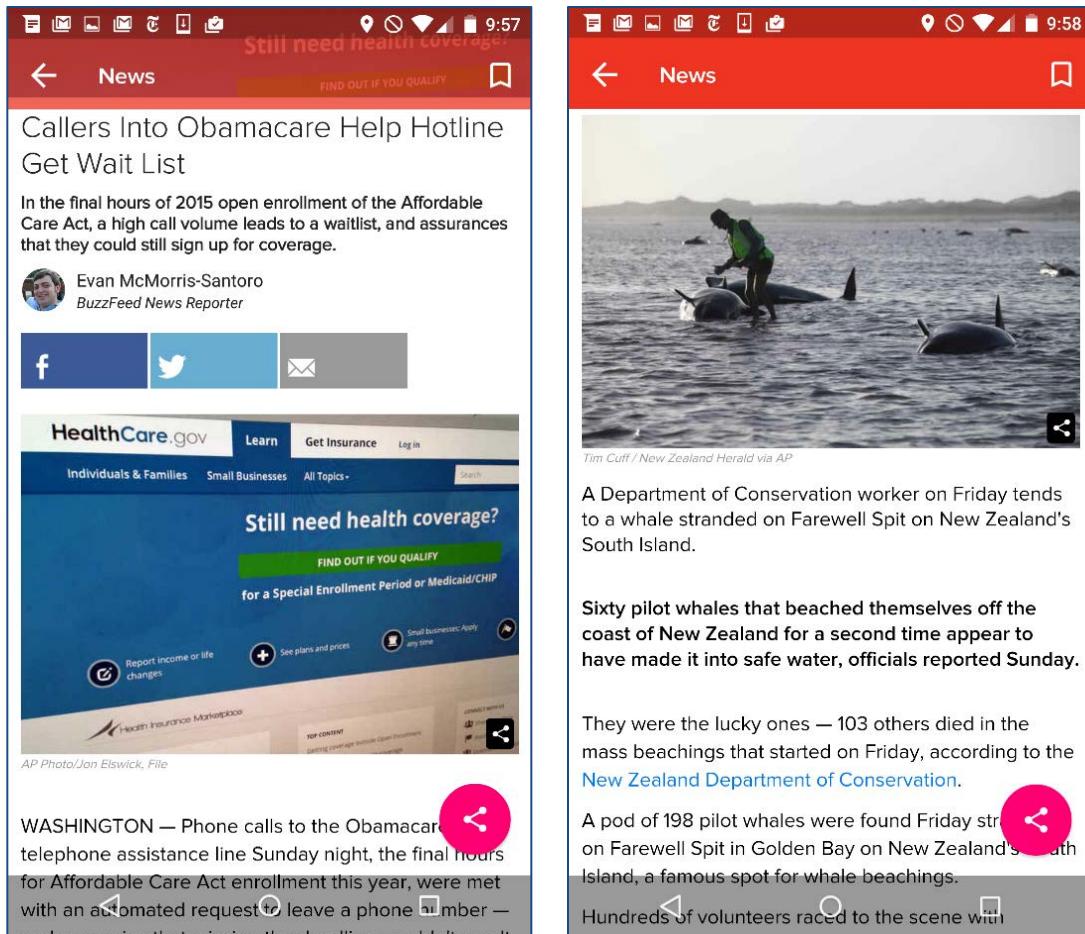
Nordstrom.com: On an older version of the site (left) users complained that the thumbnails were too small; a redesign used substantially bigger thumbnails.



Compare the big thumbnails from Zappos.com (left) with tiny ones from Instacart for iPhone (right).

298. Use captions for images that are part of an article only if their meaning is not clear from the content of the article.

Captions are useful when the photographs bring extra information to the article. If your article is about a TV show, for instance, you probably do not need a caption for a picture from the show. If an article is about a politician, again a caption may not be necessary, unless it shows that politician at an event discussed in the article. But if the picture shows an interesting detail about the story, it is worth a caption.

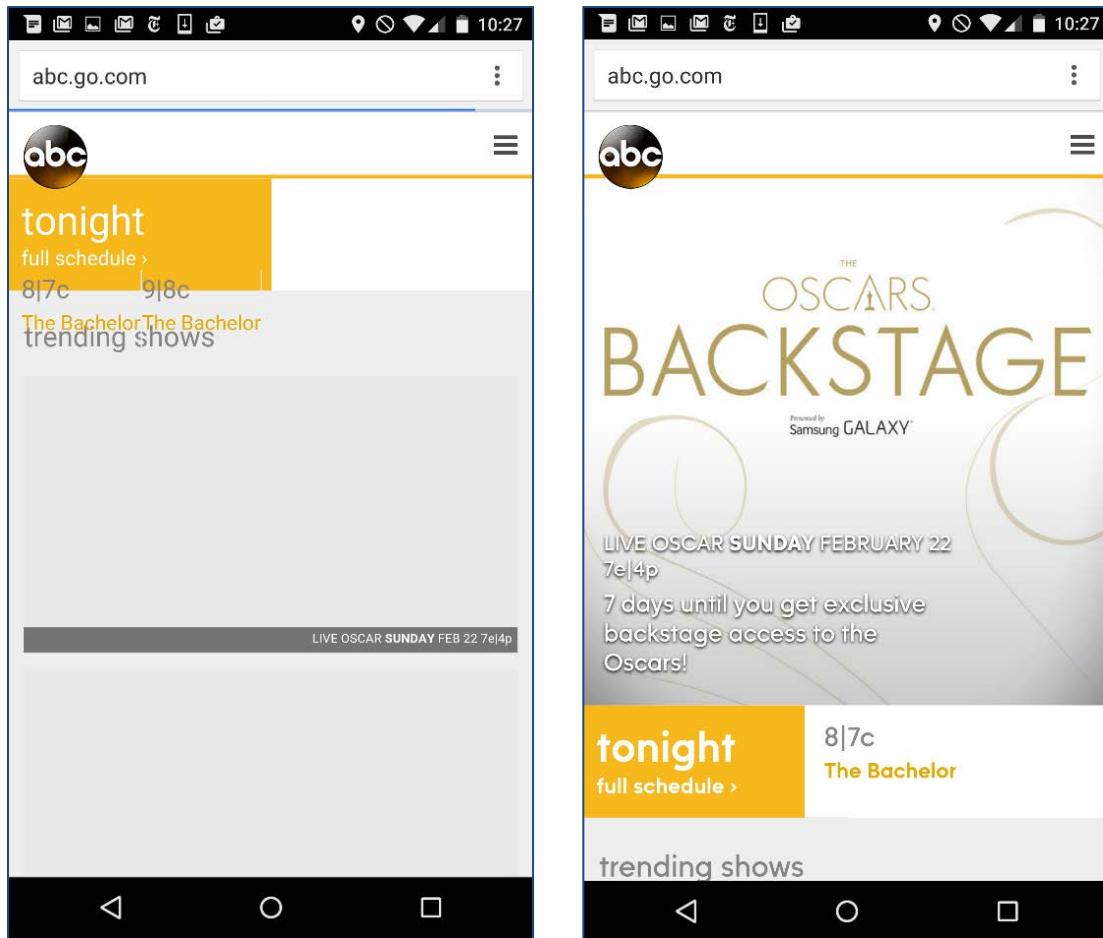


Two articles from BuzzFeed on Android: (Left) No caption was needed for an image from Healthcare.gov in an article about the website. (Right) The image depicted an interesting detail that deserved an explanation in an article about whales on New Zealand's coast. (Note, however, that the formatting of the caption does not differentiate it sufficiently from the rest of the article.)

299. Do not shuffle text around as images get loaded. Use placeholders as images load.

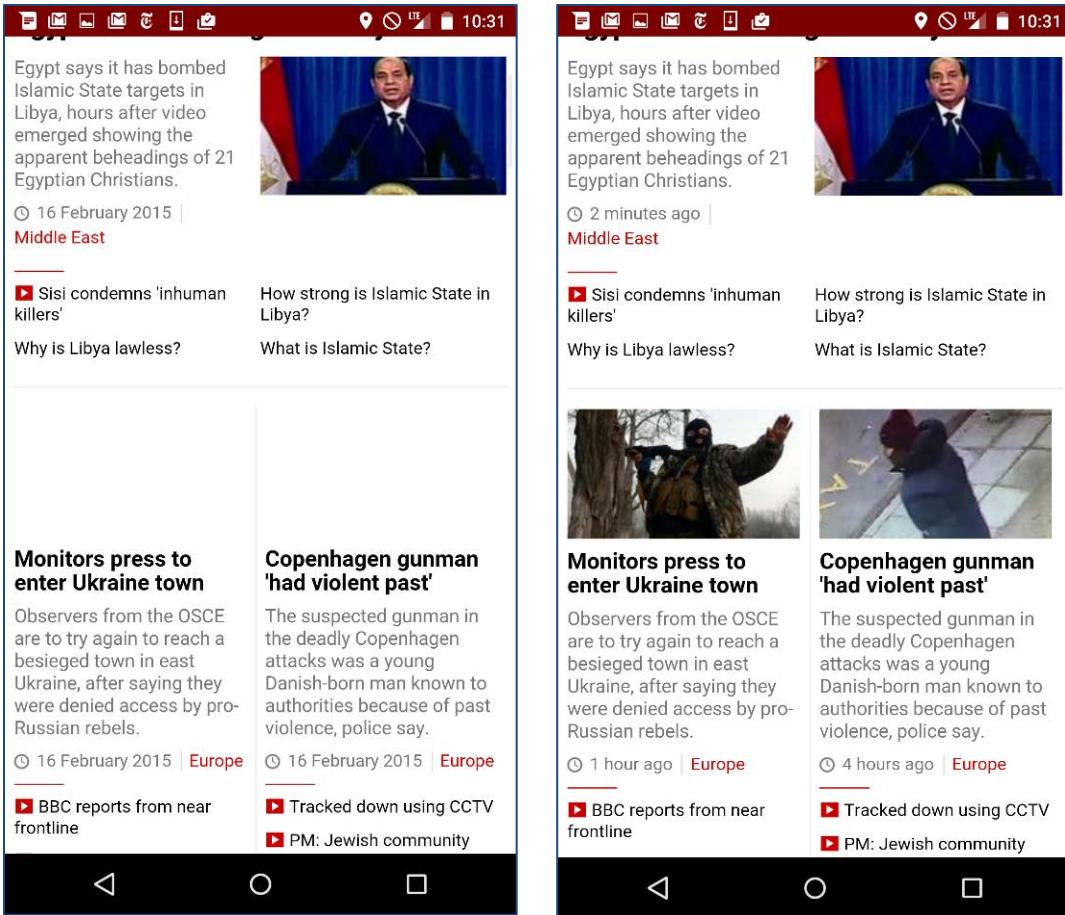
If a page has many images, the textual elements usually load faster and get displayed. Some websites use no placeholders for the images that are still loading, and the text gets quickly displayed, only to be shuffled around once images start becoming available. This is highly disruptive to the users, because they typically start reading as soon as content is visible on the page; when the same content is moved around, they have to find it again on the page in order to continue reading.

The example below is from ABC's website:



The homepage of ABC.com before (left) and after (right) the images got loaded

Contrast that with BBC's website: here, images had their own placeholders where they appeared when they loaded. None of the text on the page was moved around once the pictures became available.

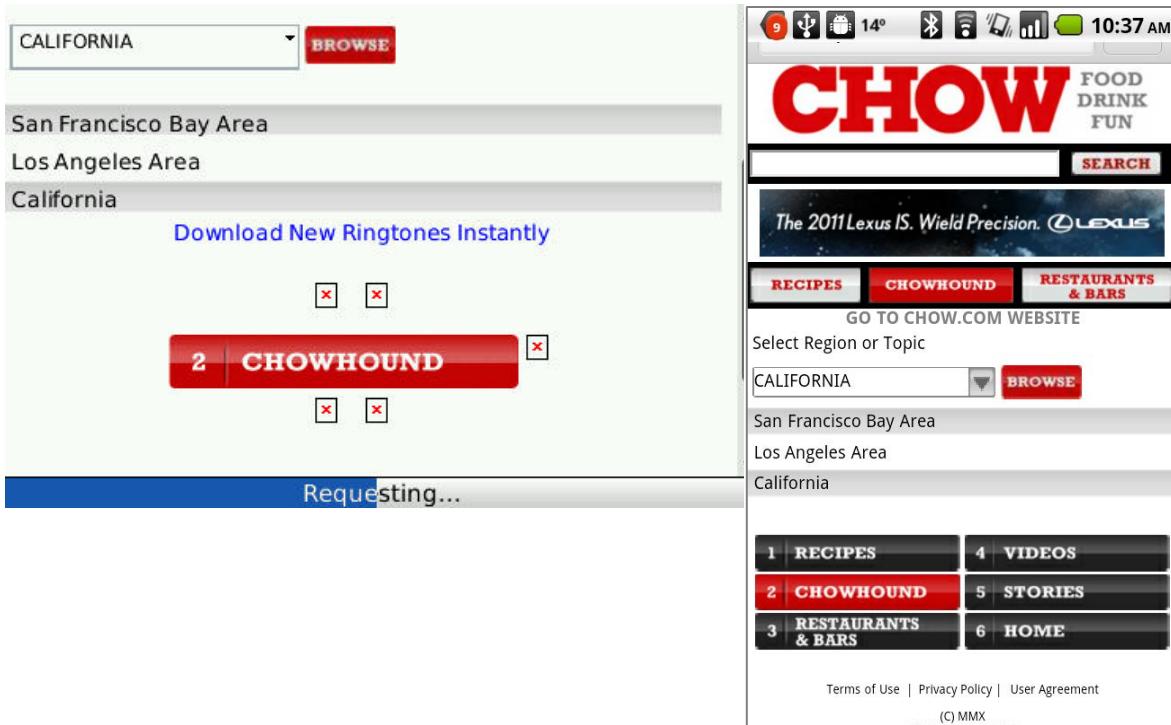


BBC.co.uk: The text did not get shuffled around as the images loaded.

300. Do not use images of text for navigation links.

When instead of text, sites use images to make their navigation options look fancier, they expose themselves to the risk that these images be loaded late. Unfortunately, this can cause problems on slow connections, because users may want to take an action and click on a navigation link before the entire homepage is loaded.

On Chowhound's older mobile site, one of our participants was looking for a recipe. She never got to see the link to recipes, because the page loaded very slowly and the link was an image. This is how the website looked on her Blackberry:



The navigation links on an older version of Chowhound.com took a long time to load on the Blackberry (left), and the user did not find the link to recipes, which was hidden in an image. The screenshot on the right shows how the site was supposed to look like when all images were loaded.

Videos

301. Do not start videos automatically.

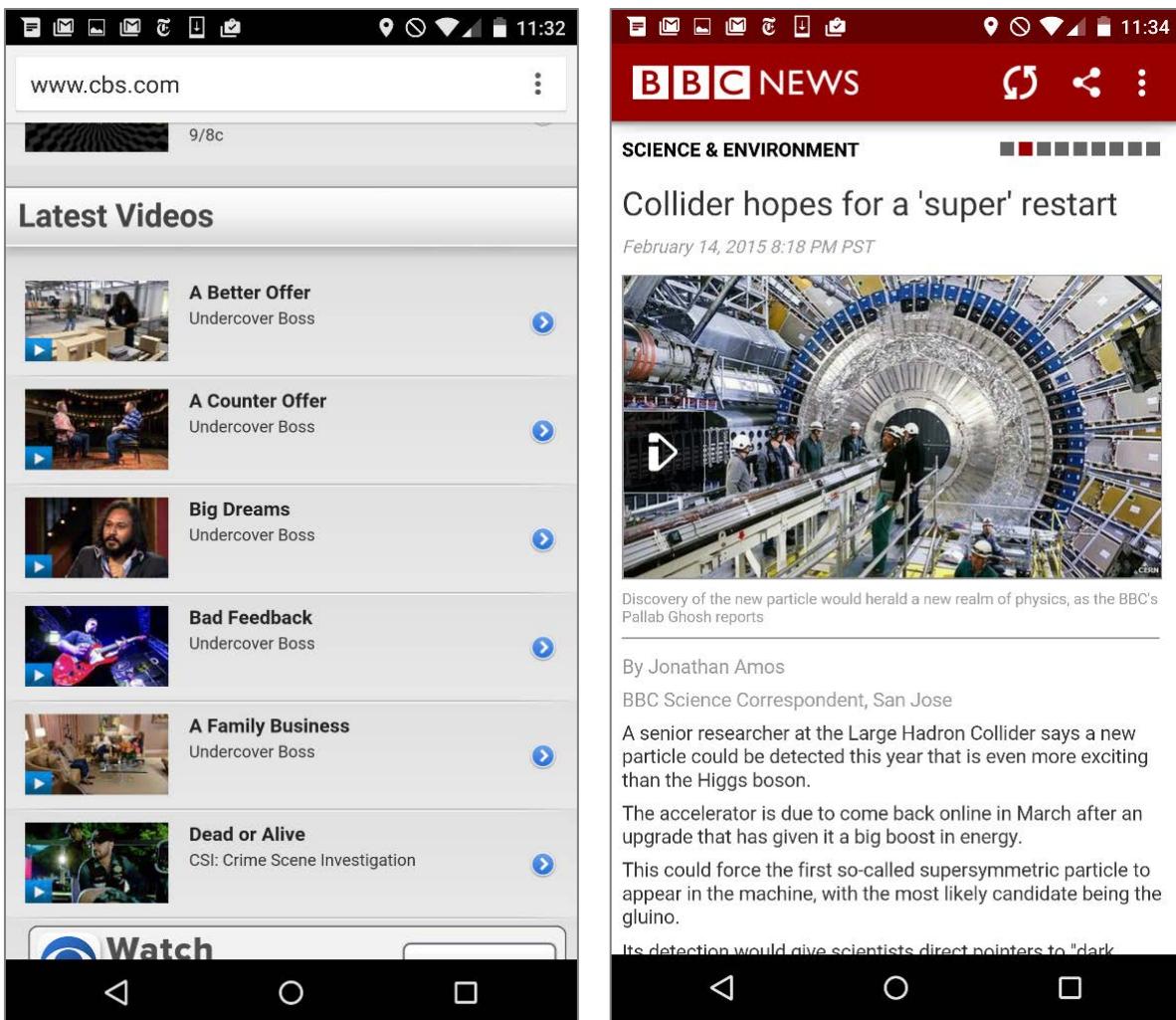
There is nothing more annoying than to have a video play unexpectedly. Even when people do have their device muted, they do not expect a video to start playing unless they have explicitly taken an action to initiate the video.

Beyond the surprise effect, remember that mobile devices are used in a variety of circumstances, so don't embarrass your users by automatically playing a sound or a video when they have not requested it.

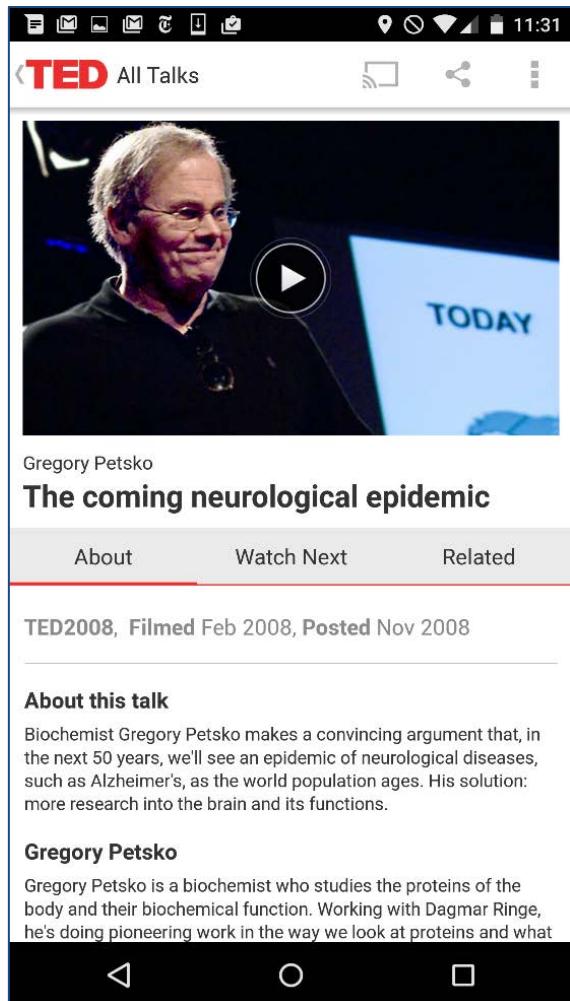
302. If you have videos on your site, offer a textual description of what the video is about.

Videos take a long time to load, so you want to give the user all the information they need to decide whether it's worth watching them. We recommend that next to each video you include a short textual summary.

In the examples below, CBS's mobile site did not offer any summaries for their videos, making it hard to pick one.



Left: CBS had no summaries for their videos. Right: BBC for Android had a “video caption” that summarized the video. Neither indicated the video length.

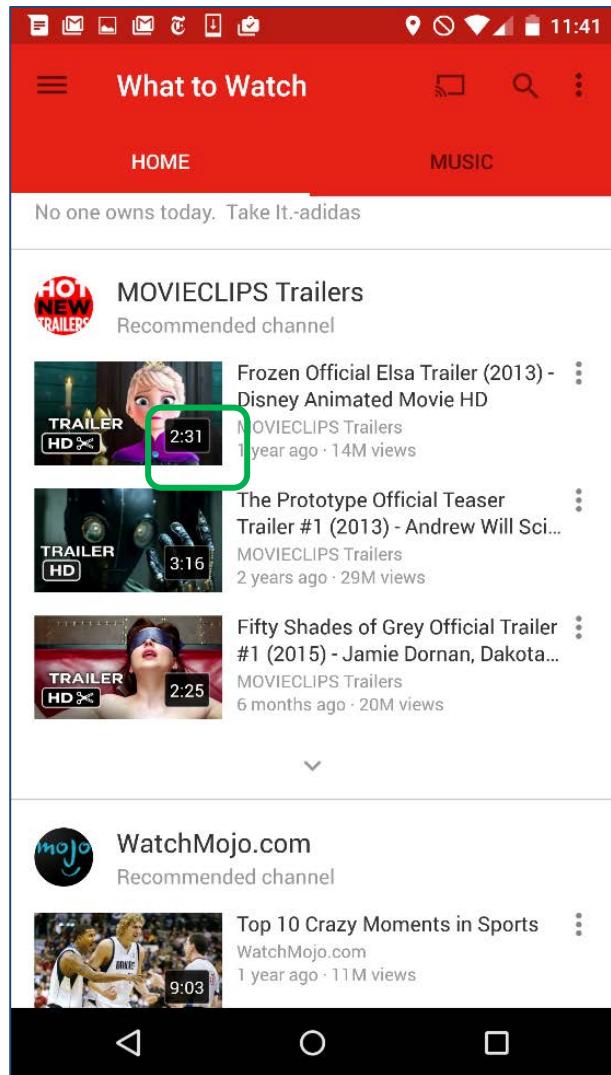


TED for Android did have a longer video summary, as well as information about the speaker.

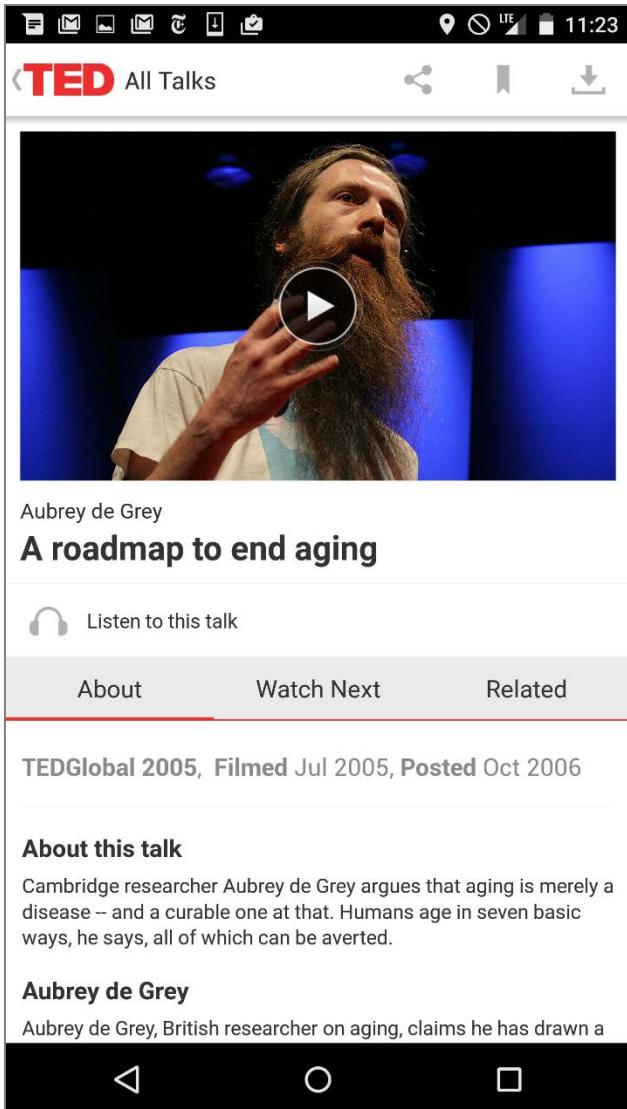
303. Indicate video length.

This guideline is valid for video on a mobile device or on a desktop. However, it is particularly relevant on a mobile device, where a long video can take a lot of time to download. Users often watch videos on mobiles to pass the time while they are waiting for something. The length of the video may give the user an idea about whether they have the time and the connection speed necessary to try watching the video, or it's best to postpone the task for another time.

Among the examples under guideline 301, none showed video length. YouTube for Android did offer this information.



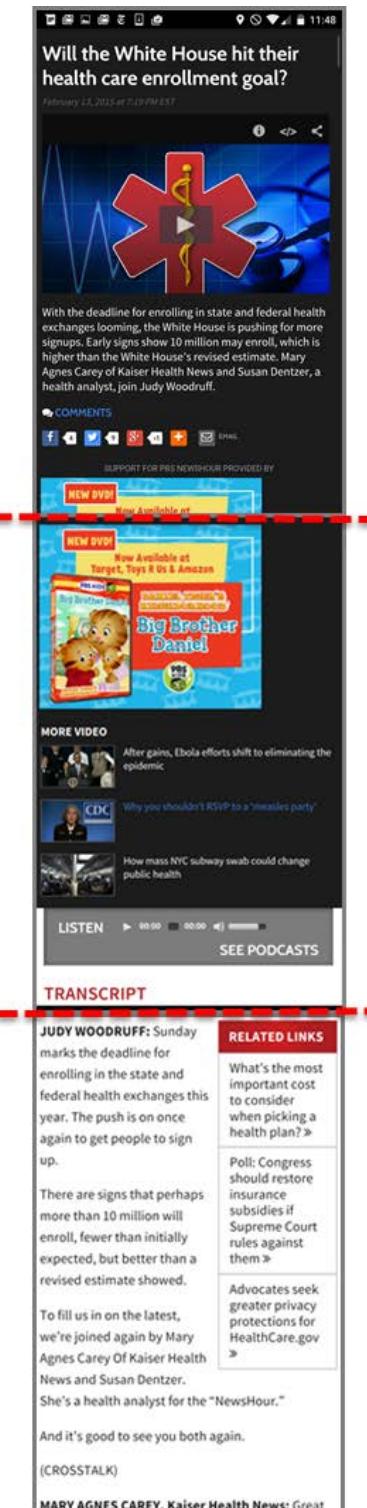
YouTube for Android showed the video length next to each video.



TED for Android had no indication about the video length.

304. Consider offering users a transcript of informational videos.

Watching a video is a sequential operation: users have to watch the entire the video without knowing if and when what it contains will be relevant or not. In contrast, a text can be easily scanned for relevant information. Plus, there are situations where users cannot watch a video but may be able to read an article. (Transcripts also helps with accessibility for deaf users.)



PBS.org had transcripts of the videos on its website. Unfortunately the transcripts were placed too low on the page, and the ad and related content in between made it hard to discover.

305. Consider offering captions to allow users to watch the video in a variety of contexts.

Sometimes people may not be able to have the sound on, but if the video has captions, they could still watch it.



TED Talks for Android offered captions for many of its videos.

LOCATION INFORMATION

Locator forms

Mobile use is often contextual: users want to find information that is related to their current context. That context often includes location: finding a gas station nearby or a good restaurant in a new neighborhood are tasks that are highly local. When we informally asked participants in one of our studies to recall one instance where the phone was particularly useful, they all mentioned maps and navigation — being able to find their way around and to get where they needed.

That's why having a locator on your web page or in your app is essential for most companies that have physical locations. That locator should be easily accessible — users need to be able to quickly find it. Location questions are frequently relative to

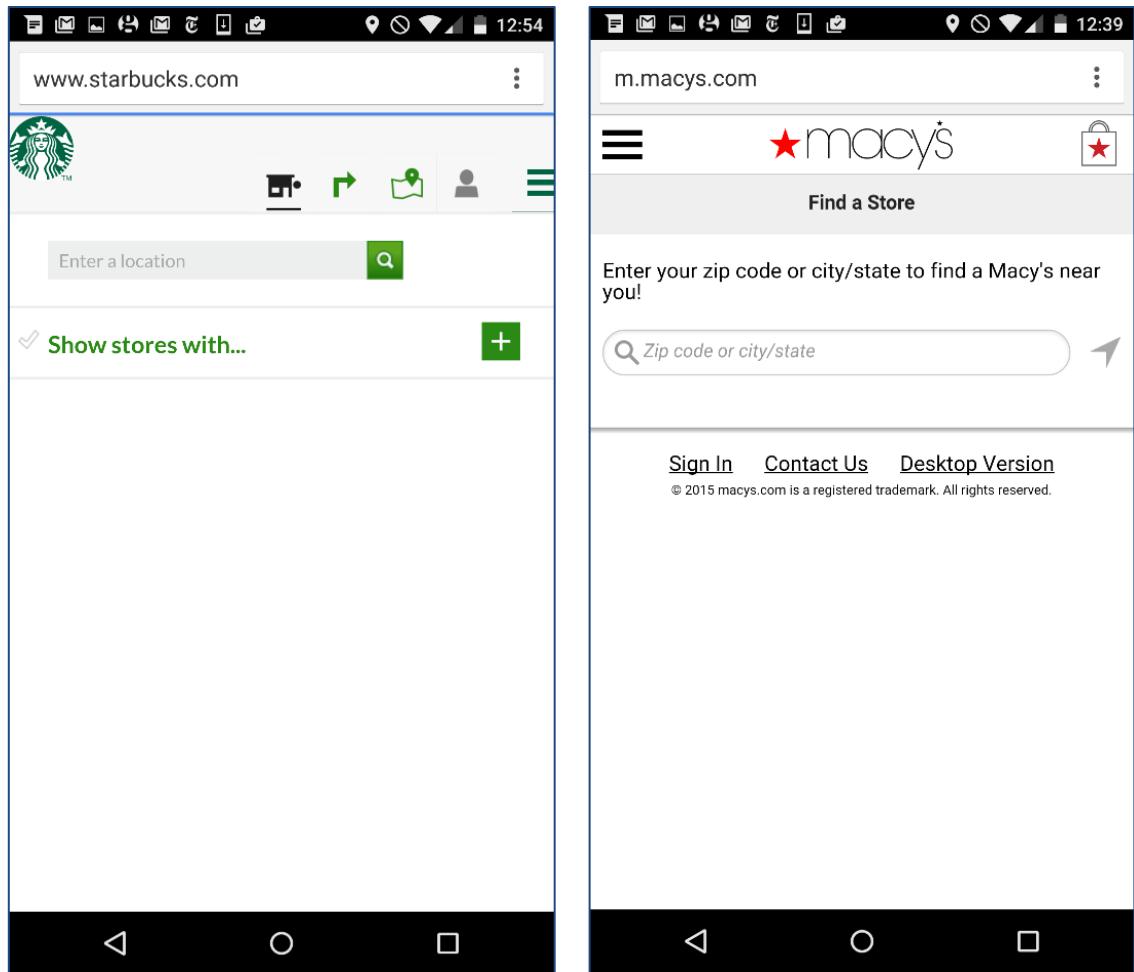
users' current location, so you should take advantage of the GPS phone feature whenever possible to save users the extra trouble of explicitly entering information about it (see guideline 70). Because current location is so often used on mobile, it should be given priority in any forms that require users to enter a location, by making it the default or the first option. That does not mean that users should not be allowed to search for locations in places other than their current location — on the contrary.

- 306. If your company has a single brick-and-mortar location, list it in your app or on your mobile website. If the company has multiple locations, include a locator form in your mobile app or on your mobile website.**
- 307. In a locator, use two possible values for location: automatically detected current location and location specified by user.**
- 308. In a locator form, always give priority to the current location.**

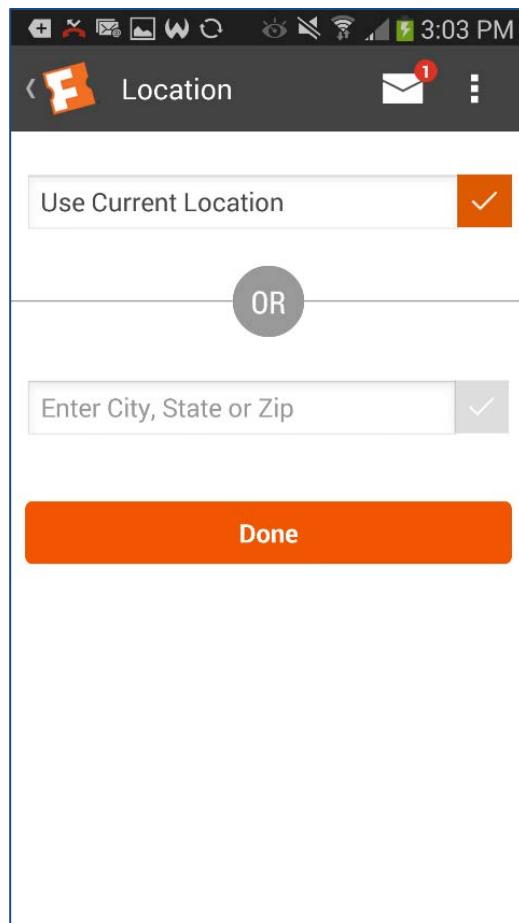
See also the discussion under guideline 75.

Most apps and websites that we've seen do a good job of including locators on their website. The implementation of those locators is, however, often suboptimal, as the examples below show it.

Starbucks.com did not make use of the phone's GPS feature and asked users to enter their zip code. Macys.com did allow use of the current location, but that option was last on the locator form, before the options to search by zip code and to search by city and state. When users see fields that need to be filled, they often start typing right away, without looking at the other elements on the page. That's why it's better to have the current location at the top, before the other options, to enable them to quickly take advantage of it, without inadvertently thinking that they need to enter their zip code. The third example (from Fandango) shows a correct locator — it first listed the current location option, and only then the textbox for the city and state.



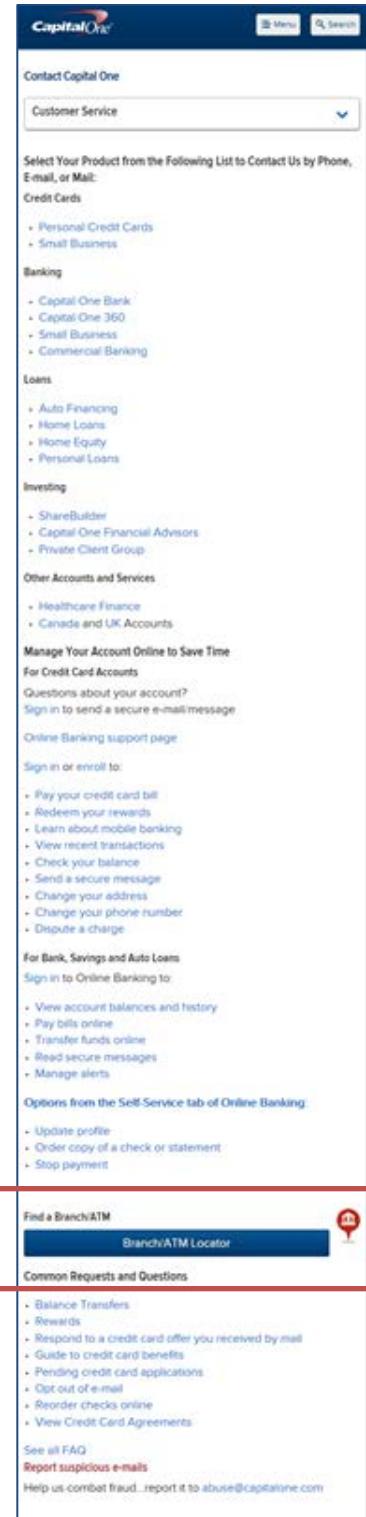
Starbucks.com (left): The current location was not an option. Macy's: The current locator was an option, but it was listed after the *ZIP code or city/state* box.



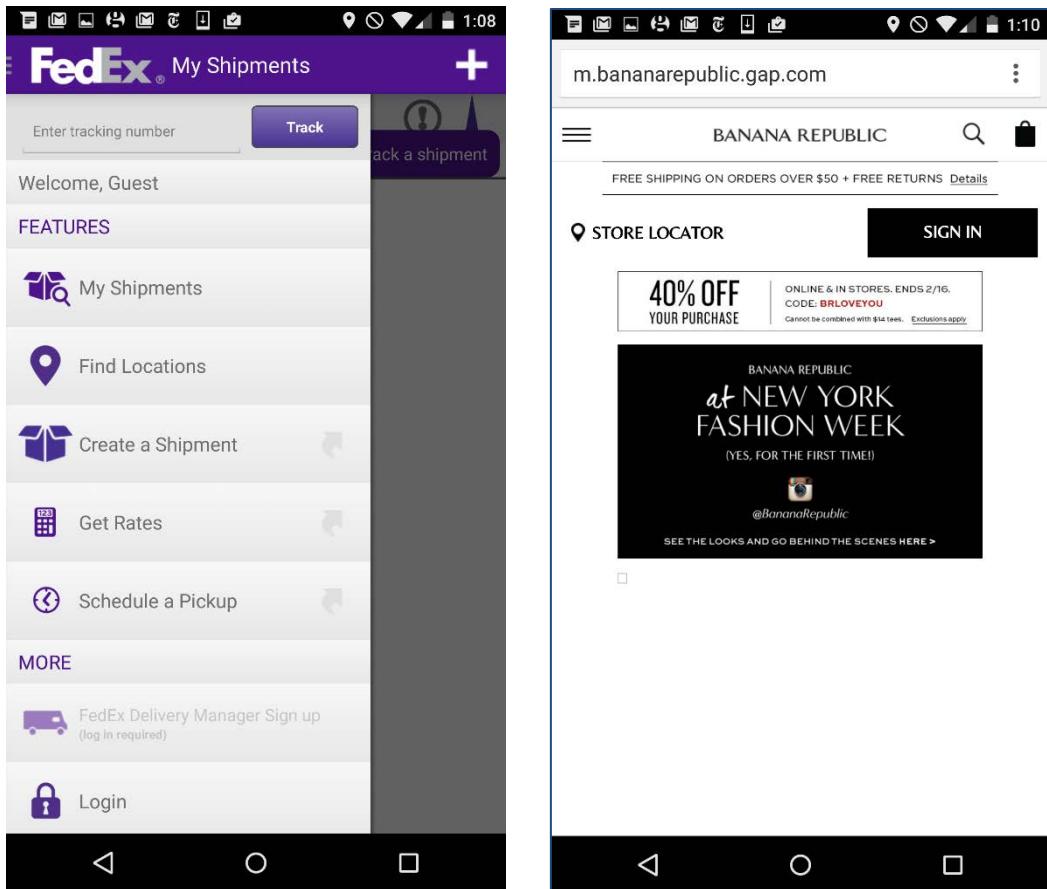
Fandango for Android had a correct locator design, with the current location listed first, followed by the specific-location field.

309. The locator form needs to be highly discoverable.

Since finding location information is among the most frequently performed tasks on mobile, store locators should be prominently placed and easy to find.



On CapitalOne.com, the ATM finder was buried at the bottom of the Customer Service page.



Both FedEx for Android (left) and Banana Republic (right) had discoverable store locators. FedEx's was inside the main-navigation menu, and Banana Republic's was at the top of the homepage.

Displaying Location Information

- 310. Whenever you have information about a single location on your website or in your app, link it to the phone's map application.**

Maps often create problems when embedded on web pages and in apps, as they are prone to swipe ambiguity. Whenever possible, delegate the task of displaying a map to an app that people will already be used to: the native map app⁶⁷.

⁶⁷ See also Aurora Bedford. "Maps and Location Finders on Mobile Devices." <http://www.nngroup.com/articles/mobile-maps-locations/>

The image consists of two side-by-side screenshots of a mobile web browser displaying the Starbucks website. Both screenshots show a search results page for locations in Sunnyvale, CA, United States.

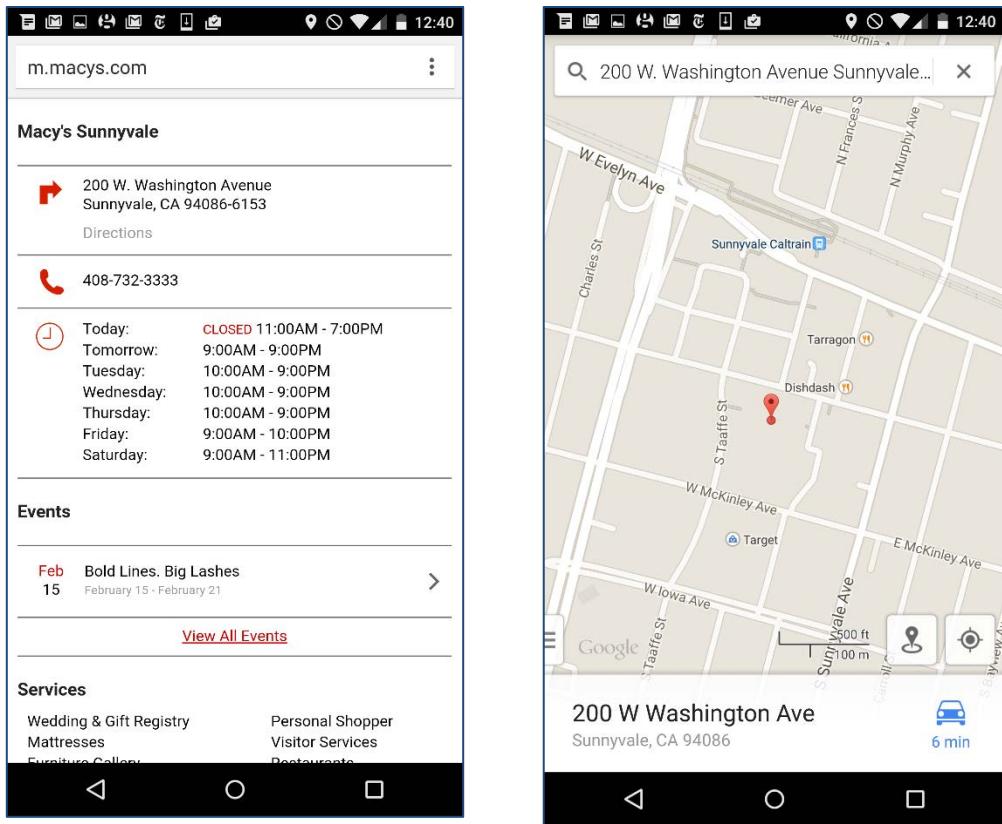
Screenshot 1 (Left): This screen shows a list of nearby businesses. Each item includes the business name, address, phone number, and a 'CLOSED' status indicator. Below the list is a navigation bar with arrows and a magnifying glass icon.

Business Name	Address	Status
El Camino & Mathilda	332 W. El Camino Real Sunnyvale, CA 940871306	CLOSED
Safeway-Sunnyvale #2887	762 Sunnyvale Saratoga Rd Sunnyvale, CA 940871462	CLOSED
Target Sunnyvale T-2584	298 W McKinley Ave Sunnyvale, CA 940866193	CLOSED
Mary & Washington - Sunn...	201 S. Mary Avenue Unit #1 Sunnyvale, CA 94086	

Screenshot 2 (Right): This screen shows a detailed view of the 'El Camino & Mathilda' location. It includes a map of the area, the business name, address, phone number, regular hours (listing days and times), and an 'Amenities' section. A 'Sign In' link is visible at the top right.

Day	Open	Close
Tomorrow	5:00 AM	10:00 PM
Wednesday	5:00 AM	10:00 PM
Thursday	5:00 AM	10:00 PM
Friday	5:00 AM	10:30 PM
Saturday	6:00 AM	10:00 PM
Sunday	6:00 AM	10:00 PM

Starbucks.com did not link the addresses to Google Maps; instead it embedded a map into the store page. It would have been preferable to show a link to Google Maps on the location-listings page.



Macy's did link the address of the store to the Google Maps map of the location and offered directions.

- 311. When displaying location information, show address, distance from current location, link to map app, phone, and business hours (if applicable).**

All these elements are important for users who search for location because they may determine what store users select. Make sure they are all conveniently displayed with the location. In fact, for stores, it is easy to calculate whether they are open at the time; display that information in the listing, together with a link to the complete opening hours.

BANANA REPUBLIC

FREE SHIPPING ON ORDERS OVER \$50 + FREE RETURNS [Details](#)

Address or ZIP code

Banana Republic Stores

- EMBARCADERO CENTER** 0.5 mi
Shoes, Petites [Details](#)
2 EMBARCADERO CENTER
SAN FRANCISCO, CA 94111
Hours today: 10:00 AM - 8:00 PM
- GRANT AVE - SF** 0.8 mi
Shoes, Petites [Details](#)
256 Grant Ave.
SAN FRANCISCO, CA 94108
Hours today: 10:00 AM - 8:00 PM
- SAN FRANCISCO CENTRE** 1.1 mi
Petites [Details](#)
865 MARKET STREET
SAN FRANCISCO, CA 94103
Hours today: 10:00 AM - 8:30 PM
- BAY STREET** 6.2 mi

Store Detail

EMBARCADERO CENTER
Shoes, Petites

2 EMBARCADERO CENTER
SAN FRANCISCO, CA 94111

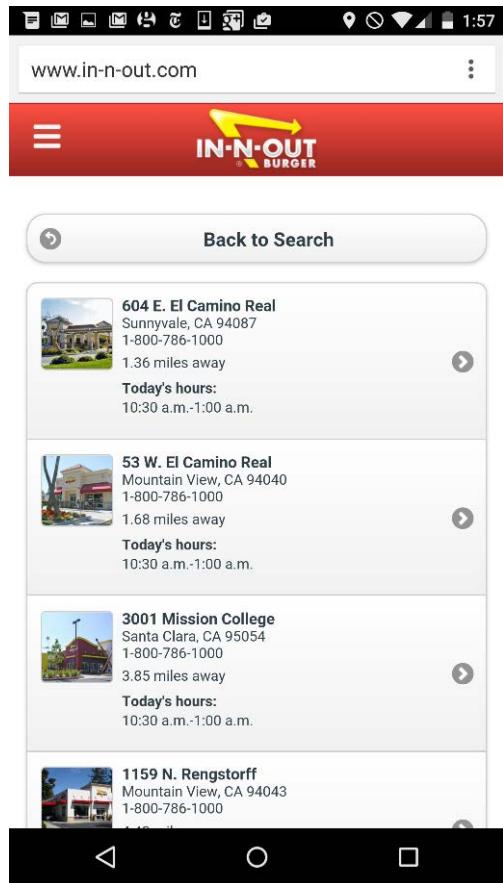
Get Directions

(415) 986-5076

Hours

Mon: 10:00 AM - 8:00 PM
Tue: 10:00 AM - 8:00 PM
Wed: 10:00 AM - 8:00 PM
Thu: 10:00 AM - 8:00 PM
Fri: 10:00 AM - 8:00 PM
Sat: 10:00 AM - 6:00 PM
Sun: 12:00 PM - 6:00 PM

Banana Republic included address and hours in the location listings, but the telephone, and link to a map were available only on a secondary page when the user tapped on the store. If the address was a link to the Google Map it would have saved users one extra step without taking any extra space on the screen.

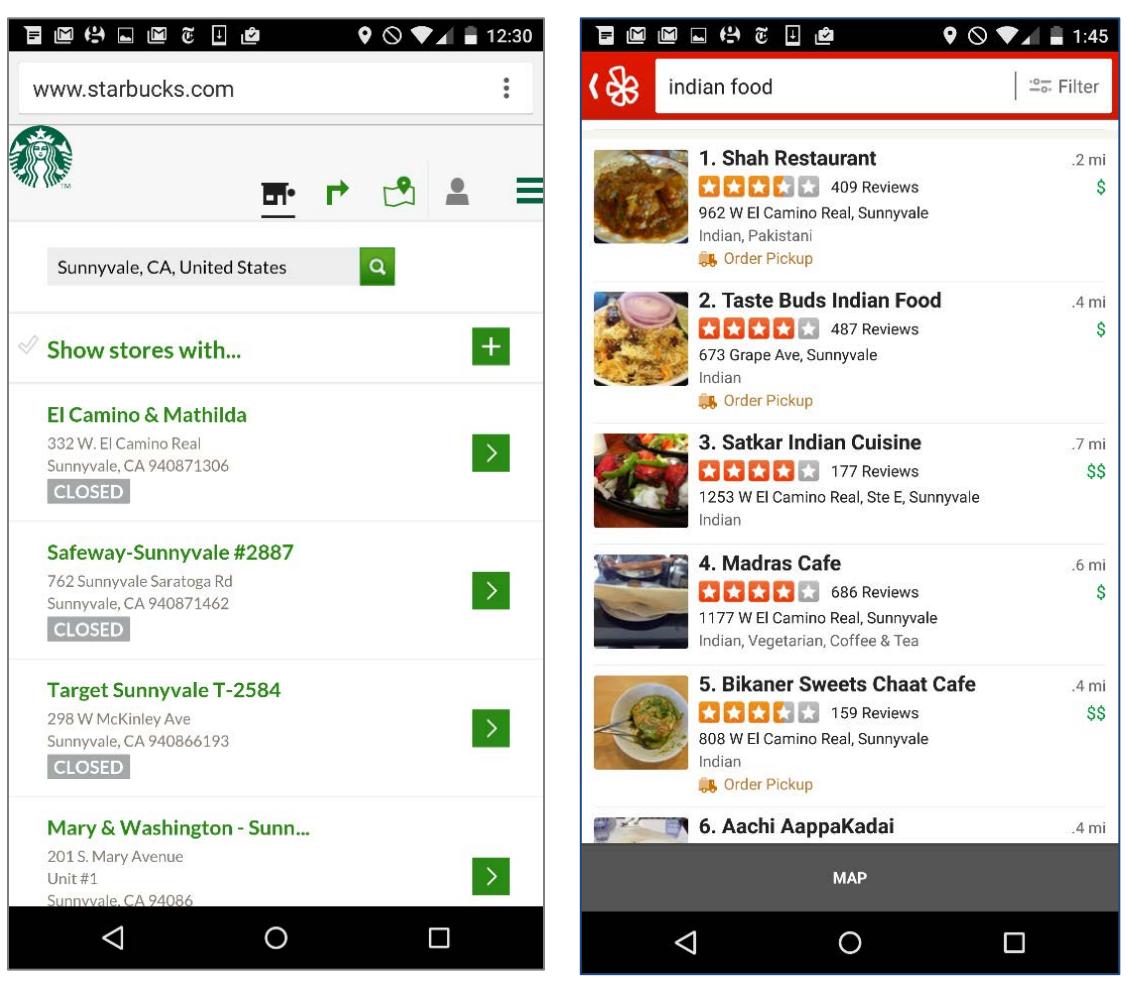


In-N-Out Burger showed the address, phone number, hours, and distance from search location. Tapping on a location took users to the Google Maps app.

312. In a list of locations, show how far locations are from the current location. List locations in the order of distance from current location.

When participants in our studies were using location listings, they often assumed that the locations were listed in the order of distance from the current location, and were surprised to discover that it was not always the case. That's one piece of information that needs to be present in the list view, so users can quickly assess what location is the most convenient.

We also recommend giving priority to locations closer to the current location (or to place where the search was performed, if the user searched somewhere different than the current location).



Starbucks (left) did not show the distance of the different stores from the search location. Yelp (right) displayed the distance and sorted the restaurants from nearest to farthest.

- 313. When displaying multiple locations on a map, start with a zoom level that enables the user to see his current location (or the location specified by the user) and at least one of the locations of interest.**

It does not help the users to see the map of the entire country, nor does it help to see just the very close neighborhood around their current location. Users need to see both the current location and one or more locations of interest. In that way, users can quickly judge where the store is in relation to themselves. Showing just the locations of interest on a map will not help either — the user may not be familiar to the area. (Note that directions usually tell the route between two points, but users may be interested in other, more subtle questions such as — is it worth it to take a detour to a different destination?)

If the location search is based on a specific location (other than the current one), that location and a location of interest should be shown on the map, instead of the current location (e.g., if a user in San Francisco searched for restaurants around Times Square in New York City, he should see Times

Square on the map, as well as restaurants around that point, but the map does not have to include San Francisco).

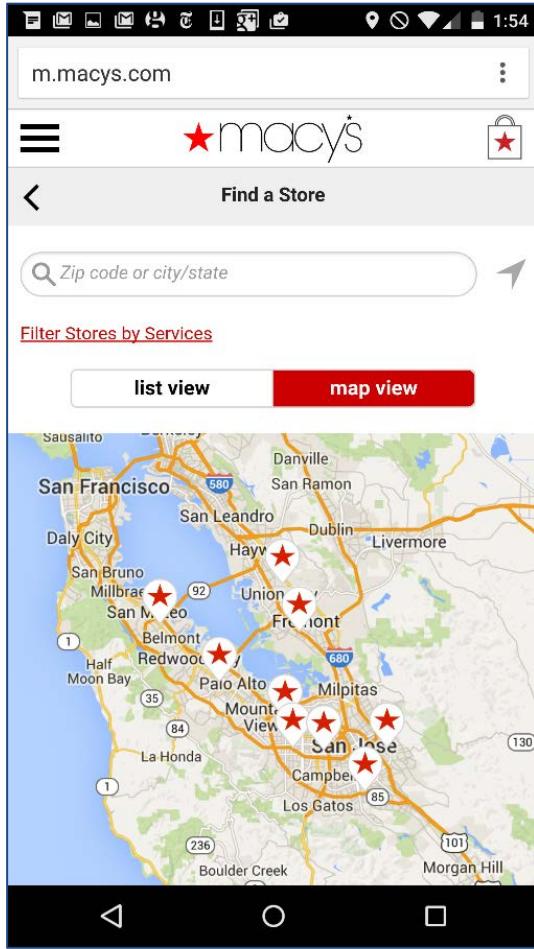
One of our Australian participants was looking for houses in Melbourne using Property Finder Australia. After he entered his search query, he was taken to a map, but no results were displayed on the map. He commented:

"It's just brought up a map with nothing on it."

The user had to slide right or left to find the search results because the zoom level was too high to accommodate any of the returned locations.



Property Finder Australia for Android: The participant searched for houses, but he had to scroll around in order to find them.

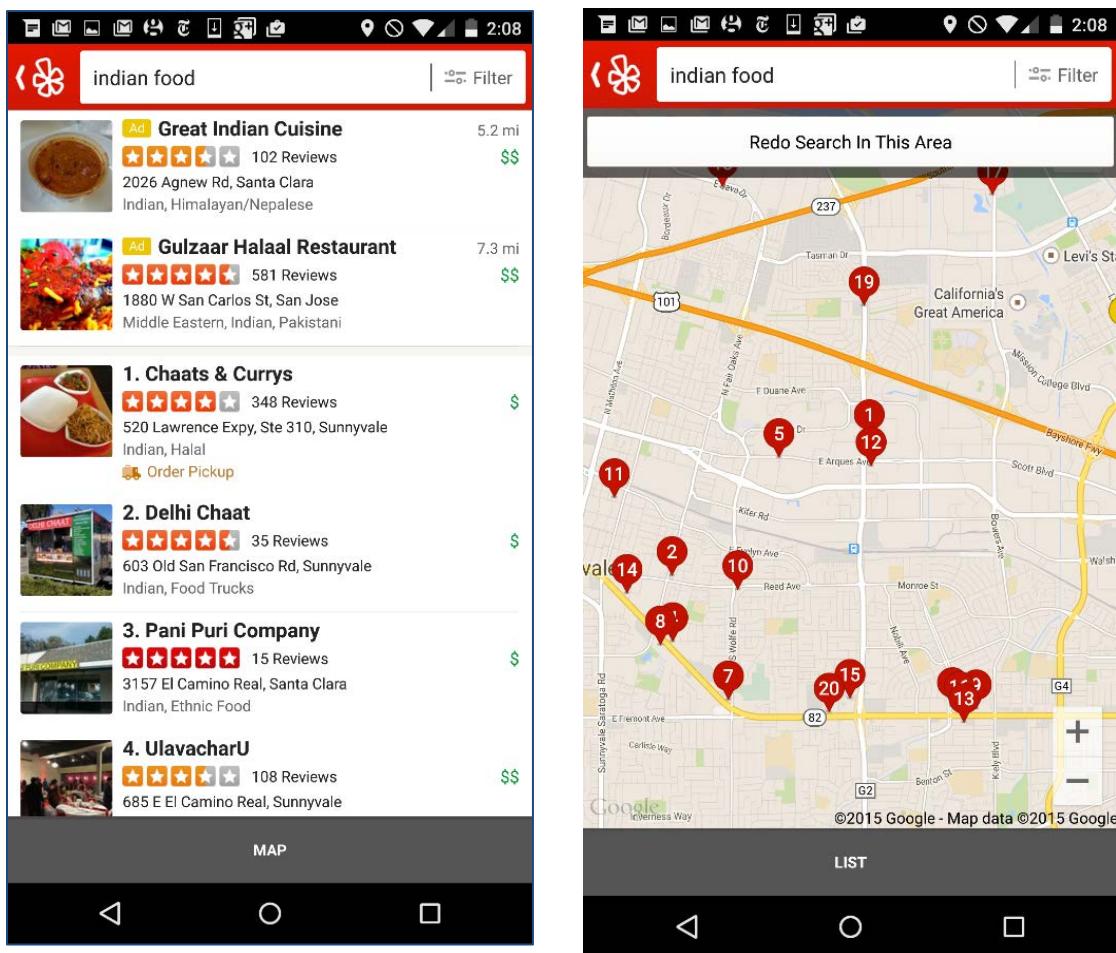


Macy's showed the map with the store locations at an appropriate zoom level: neither were the targets too crowded, nor too sparse, and the map included the search location.

- 314. Allow users to switch between a list view and a map view of locations.**
- 315. If locations have different attributes of interest to the user, provide the list of locations by default.**
- 316. If locations are all equivalent (except for distance), provide a map view of locations by default.**

When there are multiple locations, it often pays to show them both on a map and in a list. The two types of visualizations have different advantages: when they are in a list, it's easy to see details such as business hours, or information about the offerings at the location (e.g., in the case of Yelp below, the type of restaurant). However, maps give a good sense of where the locations are and how far they are from a current location or from a point of interest. Often users may not be interested in the closest location, but rather in the location closest to a certain route.

We recommend making the list of locations the default because it often contains more information. In the case of Yelp below, seeing the locations in a list lets users distinguish between different kinds of restaurants, and allows them to pick one to their taste.



Yelp for Android allowed users to switch between list and map view for locations, and, correctly, defaulted to the list view.

The Property Finder Australia app showed all the search results on a map (it did not have a list view). One of our participants commented:

"I think I would prefer to have a list of properties rather than a map. And then once you elect them you can go to a map view. It seems random and haphazard to be dragging around a map, looking for little house icons. It gives you very limited information: it doesn't give you the price range, or the amount of bedrooms, whereas if they had all properties in a list they could show that information in a fairly concise way."

On the other hand, if all locations were similar (e.g., all Chevron gas stations), then it's ok to use the map view by default.

INSTRUCTIONS AND HELP

Whenever you are seriously thinking to provide interface “tutorials” about people should use your mobile app, think again! When users are asked if they want to read user manuals, they always say “no” (and often laugh). If your app needs a user manual, it’s probably too complex for mobile.

Users may seek out help when they are confused and really motivated to use your app. The last condition is rarer in the real life than in a user-testing lab: in usability studies, we sometimes push people to finish a task that they would have given up in real life. So they are slightly more likely to seek out help. But even then, we don’t see users doing it very often.

Part of the problem with help on mobile is the single-window constraint (see section *Single Window*): users cannot study instructions and apply them at the same time, as they could do on a desktop. Because of that, they have to memorize the instructions and then use them, which puts quite a load on their working memory. If the instructions are also poorly written or formatted, their task is even harder.

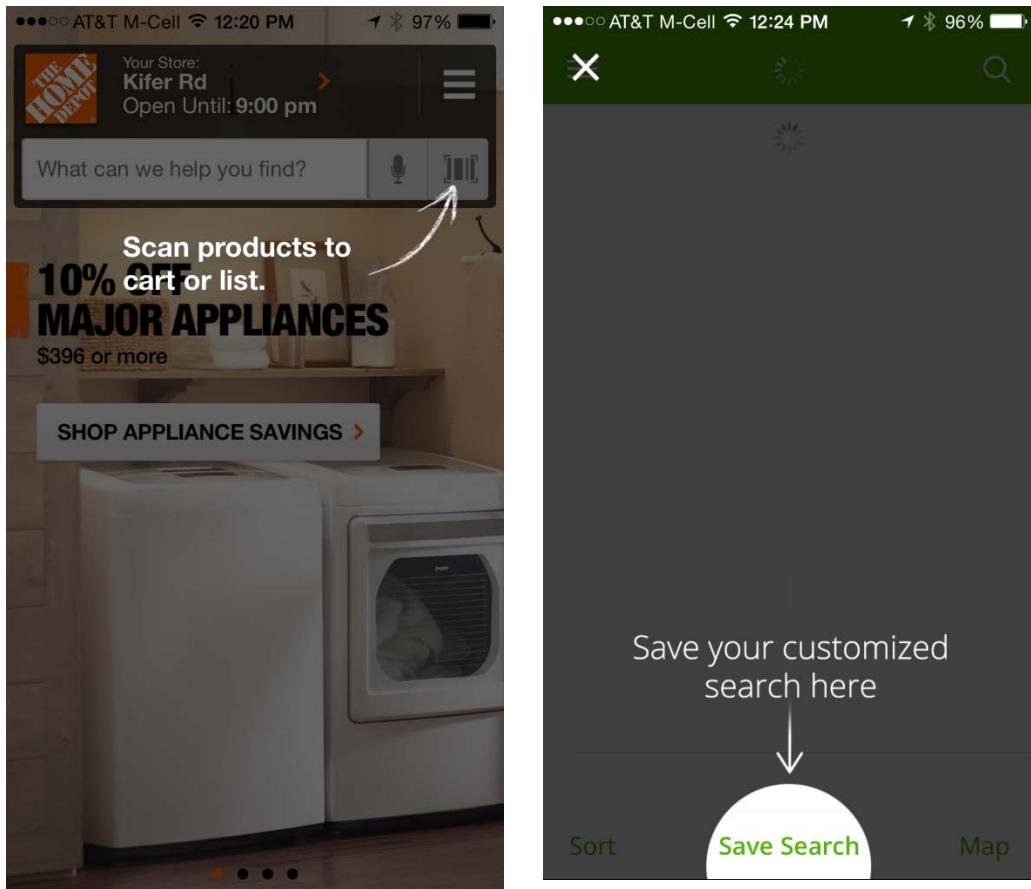
Many apps use a condensed form of a user manual — **the coach mark** or **instructional overlay**⁶⁸ (a transparent overlay of tips shown when the app first launched). These are usually better than a traditional user manual for the simple fact that they are shorter (so there is less to memorize), and are shown directly on top of the existing UI (so it’s easy to understand what they refer to).

Instructional Overlays

317. Instructional overlays should contain only tips that are truly necessary and not obvious.

Explaining that a search box is a search box is counterproductive and wastes user attention. Use the instructional overlays to show only those features that are less discoverable or standard. (But then, it may be a better idea to put the overlay work effort into making the features discoverable.)

⁶⁸ For tips on designing coach marks see also Aurora Bedford. “Instructional Overlays and Coach Marks for Mobile Apps.” <http://www.nngroup.com/articles/mobile-instructional-overlay/>

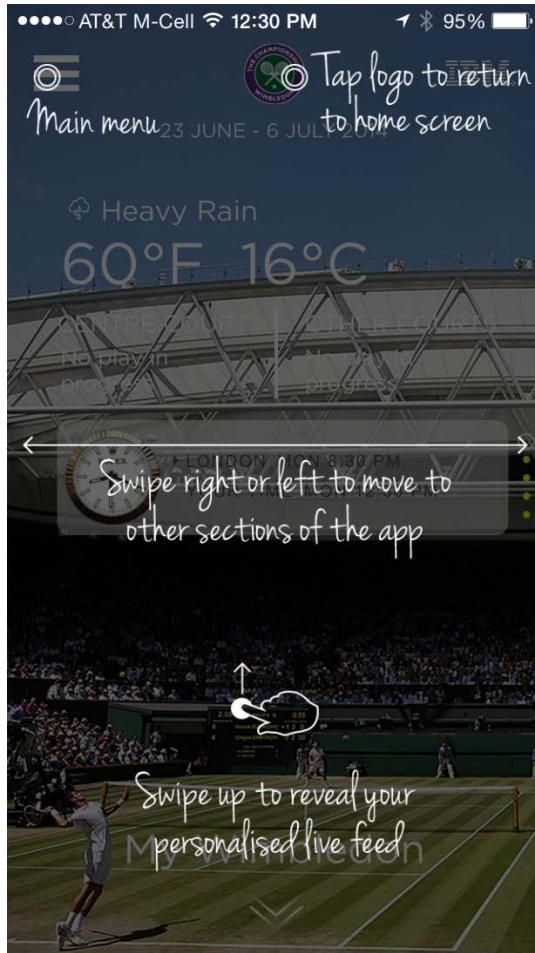


Left: Home Depot for iPhone had a tip to explain the self-explanatory scan icon. Right: Trulia for iPhone explained the quite obvious *Save Search* button.

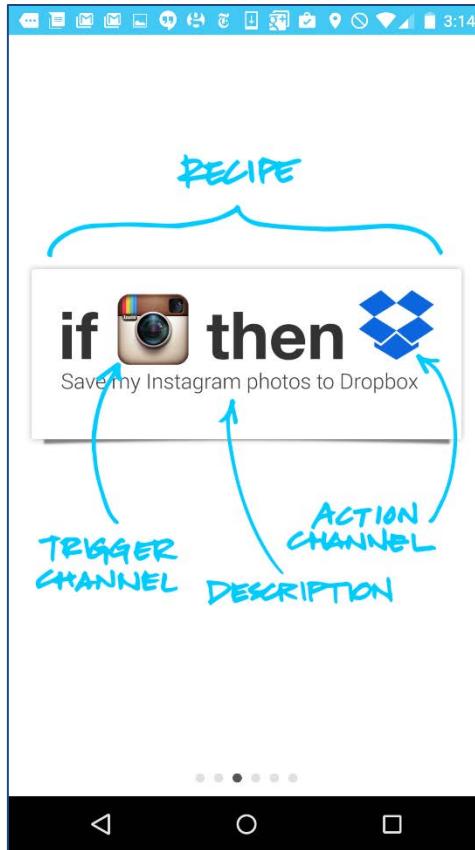
318. Make the coach mark clearly distinct from the page in the background. Signal to users whether the buttons are active or only depictions of the elements in the interface.

Users should be able to tell what is an instructional overlay and what is the actual interface. We've seen people who weren't sure if they can already interact with the app or they were looking at a tutorial. This separation is usually achieved by either using a completely different font for the coach marks, or by dimming the background substantially.

Some apps prefer a font that mimics handwritten annotations for coach marks. That is ok as long as the font is legible — such fonts can be harder to read.



Wimbledon for Android: The “handwriting” font used for the coach marks was clearly distinct from the interface. (Unfortunately fonts like these can also be harder to read; and some of the tips obscured interface elements that they were supposed to explain.)



IFTTT for Android: Although the app used a different font for the tips, because the interface elements were not dimmed, it wasn't clear whether this was an active page (and users could tap the different icons) or just an inactive picture.

Contextual Tips

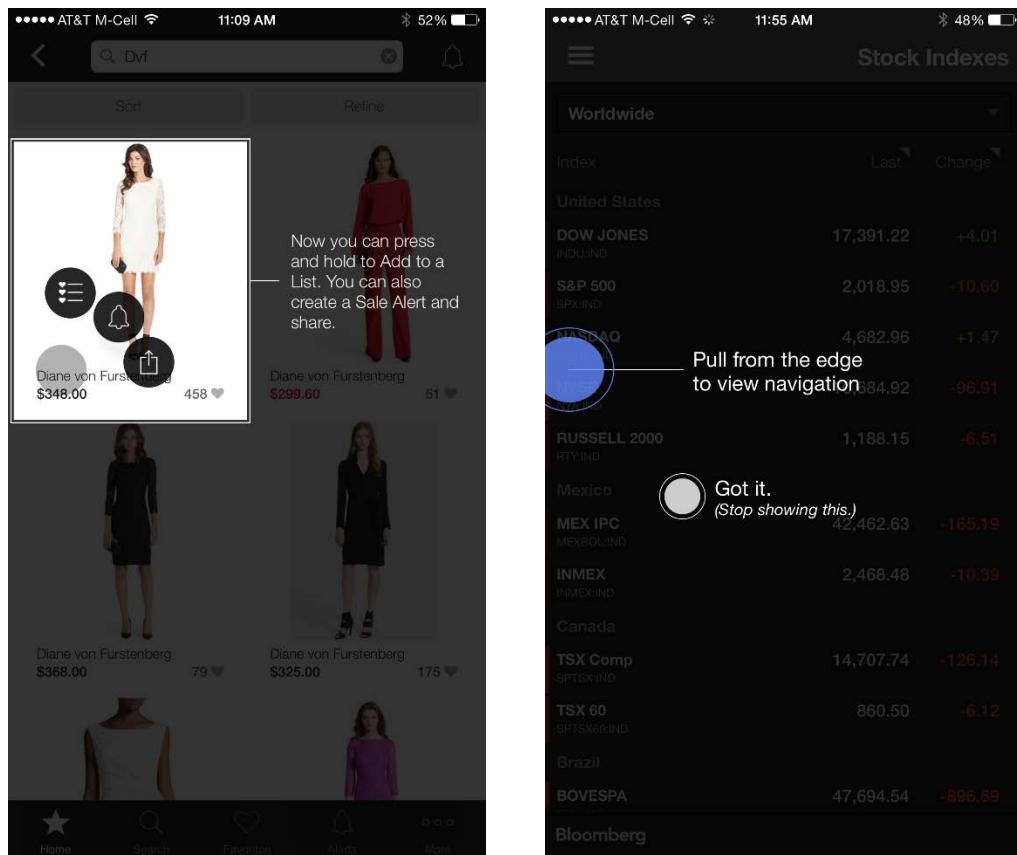
Tips that are shown exactly when the user needs them are the best help system. They are most likely to be attended and remembered by users on the go, as long as they are brief and meaningful.

In user testing, we noticed that, overall, tips are more effective than user manuals. Some users don't read tips, but some do — our observations suggest that tips are more likely to be read if (a) they are related to the task that users are attempting at the time, or (b) they appear at a time when the user is waiting for a download or some other event.

Tips that are too obvious or too general hold little interest for people, and make them more likely to distrust and ignore further tips. Also, tips that occur too often tend to be annoying, so users may start dismissing them more quickly.

319. Meaningful, contextual tips can be helpful.

Here are some examples of tips presented when users were most likely to need them.



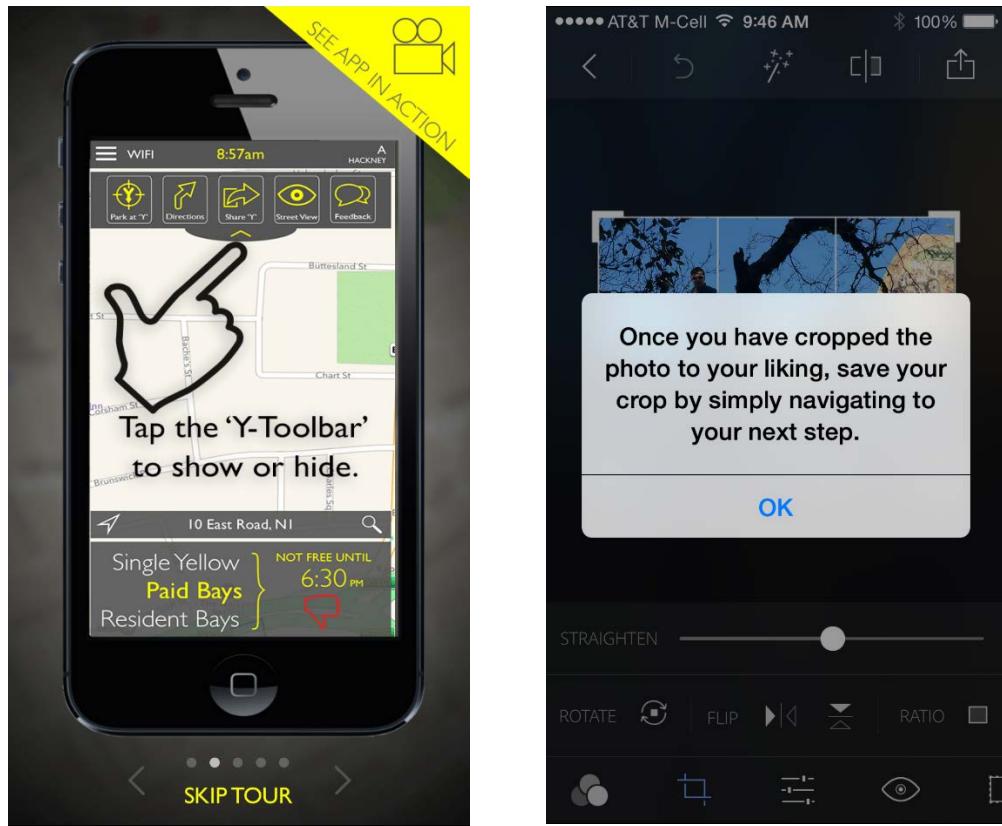
Left: ShopStyle for iPhone showed contextual tips about features that were more advanced (save an item or create an alert for an item) after the user performed her first search and was browsing through the results. Right: Bloomberg for iPhone showed a tip about revealing the navigation menu at the beginning of the session.

320. Tips should have clear, simple language.

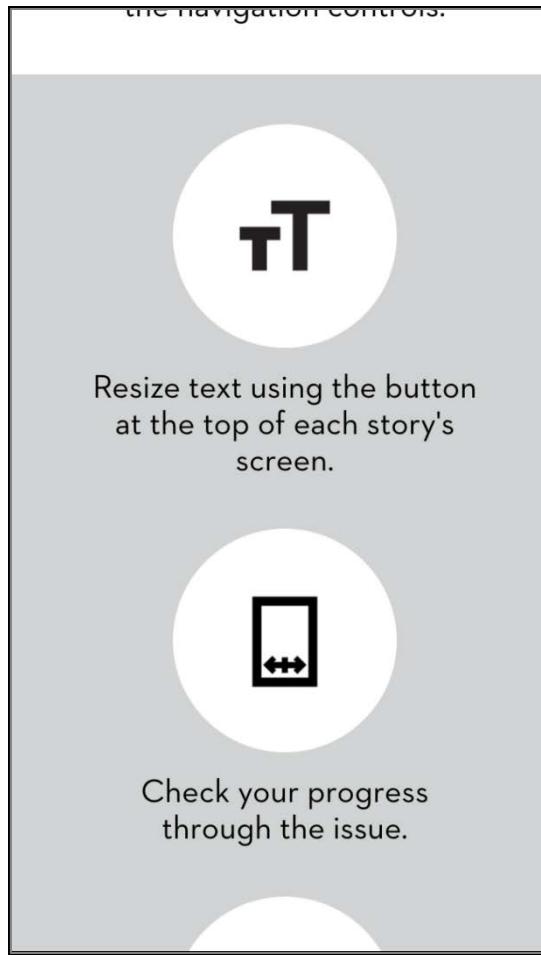
321. Tips should explain the function not only the action.

322. Tips should point directly to the design element instead of mentioning it by name.

One of the challenges of offering help with the UI is that the language that designers use to refer to UI elements is not necessarily the same as the users' language. Coach marks are a great way to get around this problem because they can point directly to the design elements that they refer to.



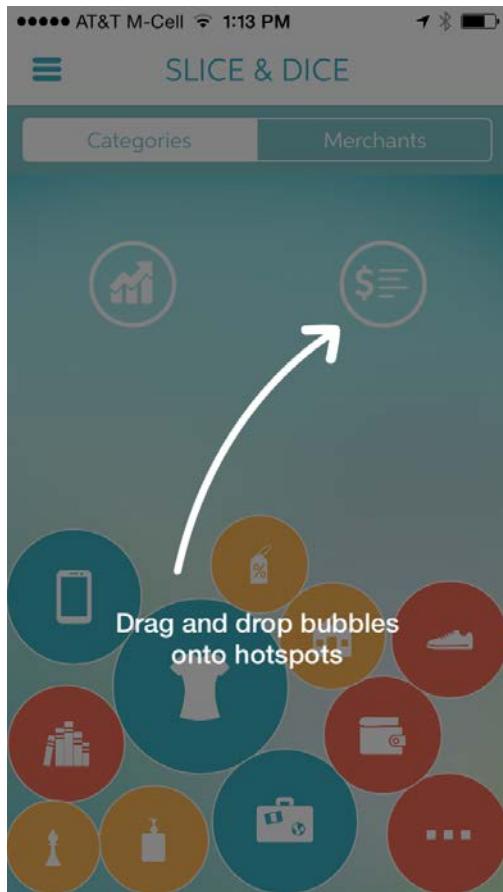
Just Park for iPhone (left) and Adobe PS (right) created tips that were hard to understand. Just Park used designer jargon ("Y toolbar") and Adobe PS had a long way of saying that the photo was automatically saved.



New Yorker for iPhone: The tip page referred to buttons out of context.



Bloomberg for iPhone: Which was the *+WATCHLIST* button?



Slice for iPhone: Why would you “drag and drop bubbles onto hotspots”? Which are the “bubbles” and which are the “hotspots”? Is the arrow pointing to a bubble or to a hotspot?

323. Do not flood users with tips.

324. Phase tips out as users learn the interface.

For those power users who have learned the interface too well, tips can be annoying. Make sure that you don’t show the same tip again (unless the user has not been using the app for a while).

325. Do not use vocal tips.

At least not by default. Paper for iPhone used vocal tips instead of the usual coach marks to remind users of certain gestures. While the tips were contextually appropriate (and highly useful), the delivery modality was not: most users are not yet used with receiving tips that way, plus it makes the app usage inconvenient in a variety of situations, because you never know when the app is going to deliver a potentially disruptive audio tip.

User Manuals

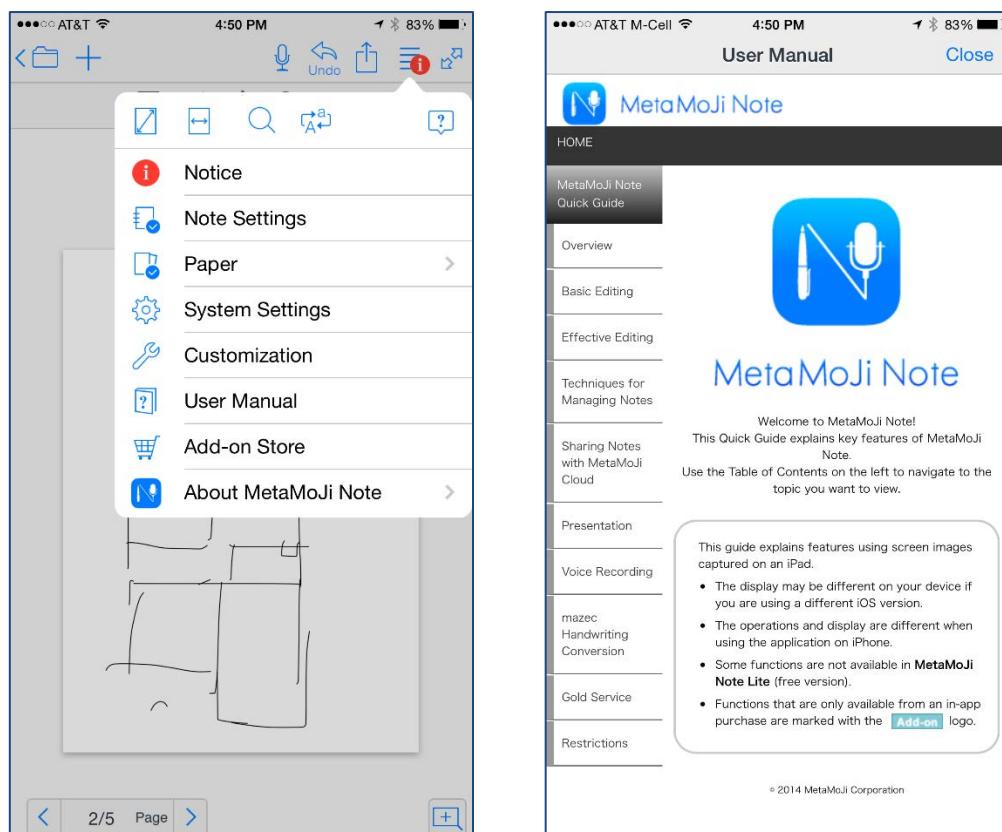
We include in this category the more traditional tutorials, as well as the popular tip screens that often appear when apps are first used. Basically, anything that has more than one page of help or explanations qualifies as a user manual.

User manuals are pretty much useless on mobile: they contain a lot of information that users must memorize. At most, user manuals can be entertaining, but users cannot be expected to remember more than one or two things from them at best. Live manuals (where users get to “practice” interacting with the app) are also not very effective, although they are more so than static ones. Users perceive these interactive manuals as obstacles that need to be moved out of the way in order to get to the more interesting part: using the app. Although they do fixate slightly better in memory, there is still a significant risk of the different actions interfering with each other. There is just too much material presented at once.

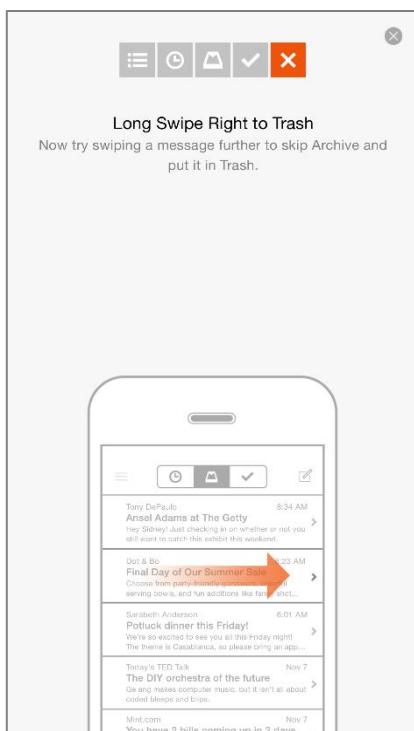
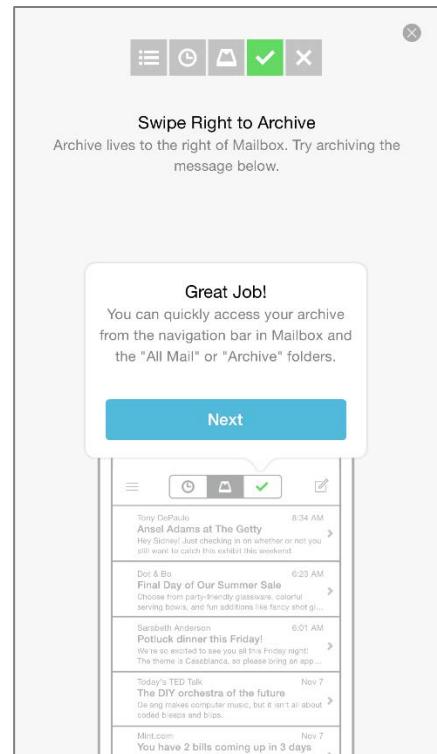
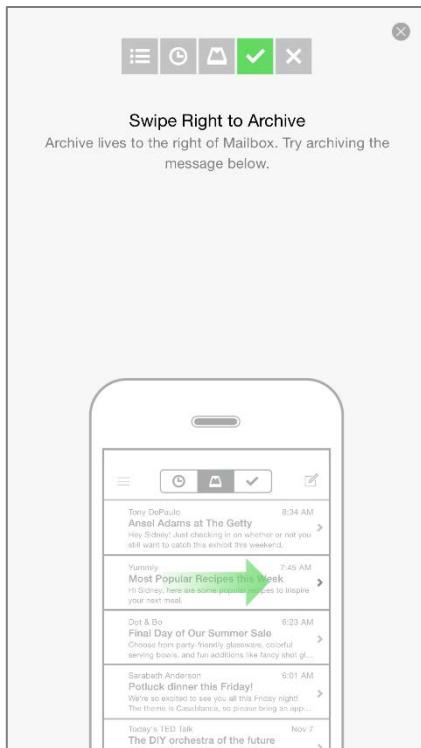
Given that most people won’t remember much out of these manuals, they should not waste users’ time. They can be made available in the app, but should not be mandatory.

Contextual tips (that are presented ideally exactly when the user needs them — see the section *Contextual Tips*) can be much more effective than any user manual.

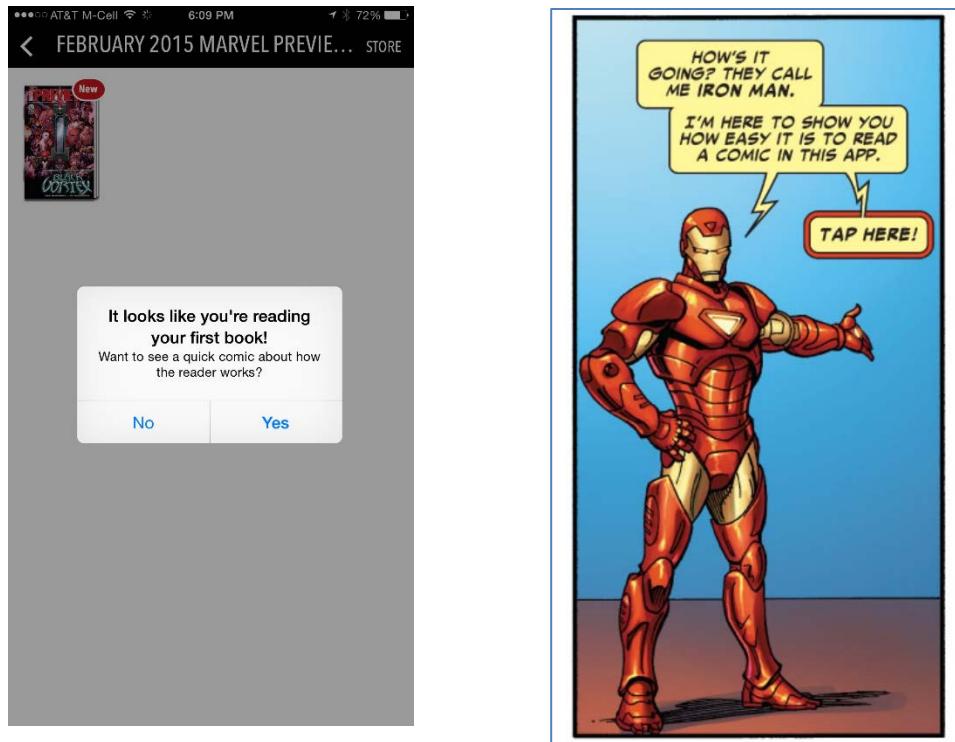
- 326. User manuals are not useful, nor should they be necessary on mobile.**
- 327. If you decide that you absolutely need a user manual, make sure that it's easy to read.**
- 328. If you decide to include an initial tutorial, allow users to skip it and use the app right away.**



MetaMoji Note for iPhone had a link to its user manual, but unfortunately this was not optimized for the device.



Mailbox for iPhone forced users to sit through a tutorial explaining the different gestures. The tutorial was interactive; however, the many gestures and features explained were overwhelming.



Marvel Comics for iPhone: The user manual was itself a comic; however, even this “fun” format cannot ensure great memorability of the material.

INITIAL EXPERIENCE

- 329. Keep launching time to a minimum.**
- 330. If users must wait for more than 20 seconds, give them something to do while waiting.**

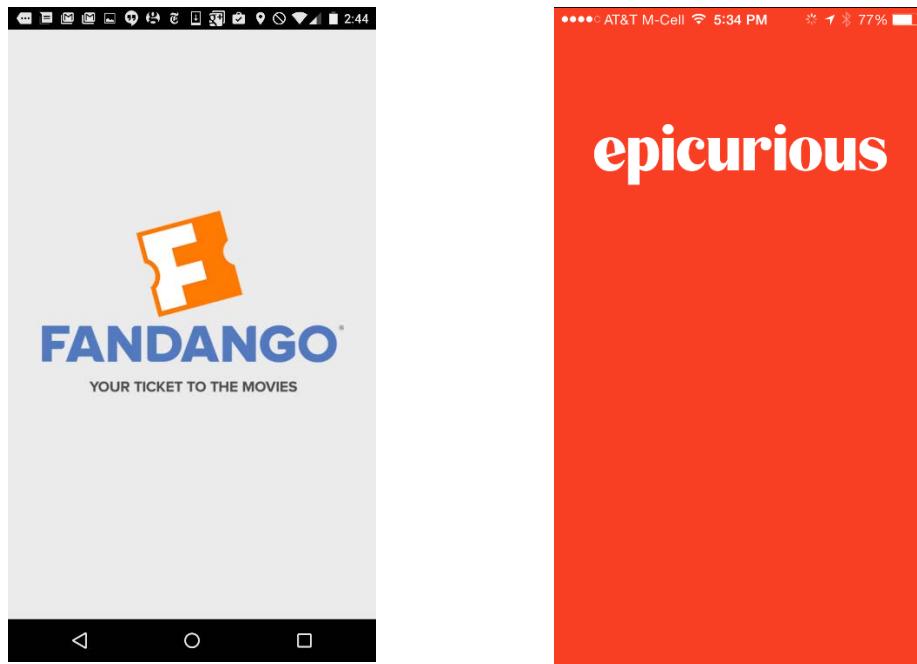
On mobile, users have little patience for long downloads. We noticed that, in the lab, they are willing to wait at most 20 seconds for a piece of content that they know is big (e.g., a magazine issue). If you keep them waiting for more than that, they become impatient and are very likely to leave the app.

In that situation, progress indicators that keep users informed of the state of the download are a must (see guidelines 132–134). Moreover, showing users a preview of the content that is loading or even some tips can keep them engaged with the app and can make them feel that they’re not waiting for such a long time.

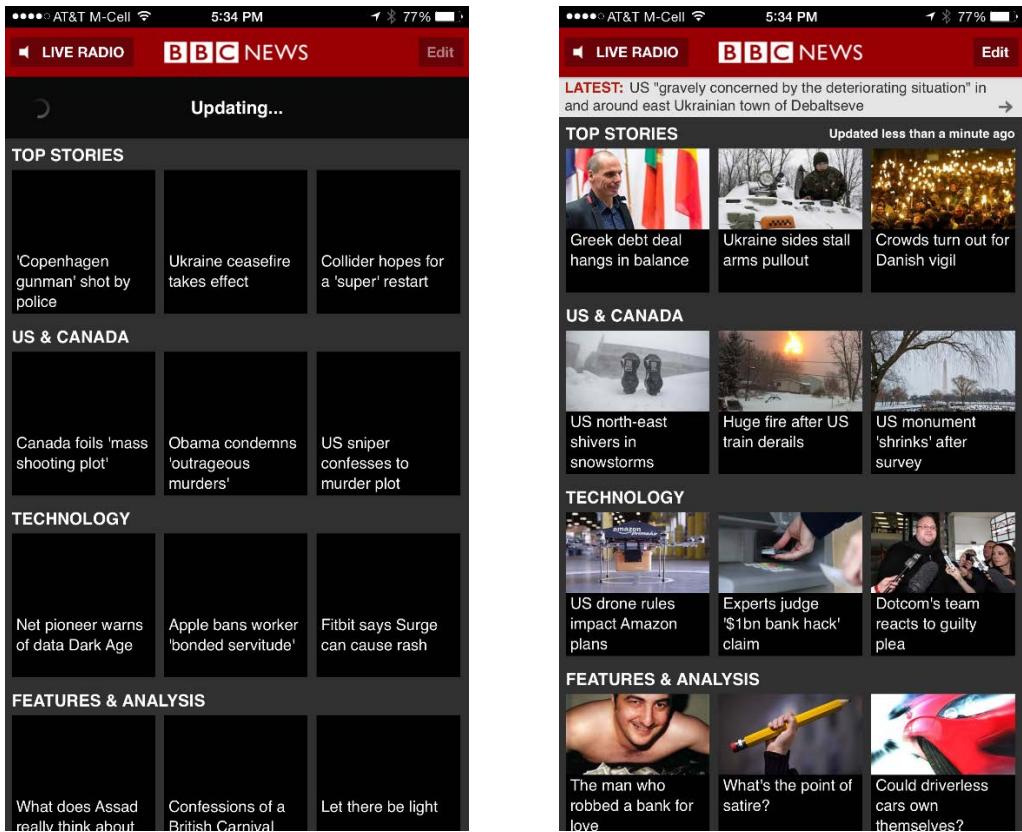
- 331. Don't use a launch screen if possible.**
- 332. If you must use a launch screen, make it similar to the first screen of the app.**

On mobile, it is important for users to get to what they're trying to do as soon as possible. Launch screens are just unnecessary obstacles — users don't care for them, and, as they use the app more and more, having to repeatedly sit through a few seconds where they cannot use the app is simply annoying.

That's why we recommend that you avoid launch screens whenever possible. If your app needs some time to load, show a launch screen that's as similar as possible to the first functional screen of the app (for instance, that could mean showing the interface with no content, like BBC did in the example below). This minimizes the cognitive effort for the users: they don't have to process a new screen once the content has loaded, and can use the extra launching time to familiarize themselves with the interface.



Launch screens do not bring any value to the user and they can give the feeling that they slow the app down.



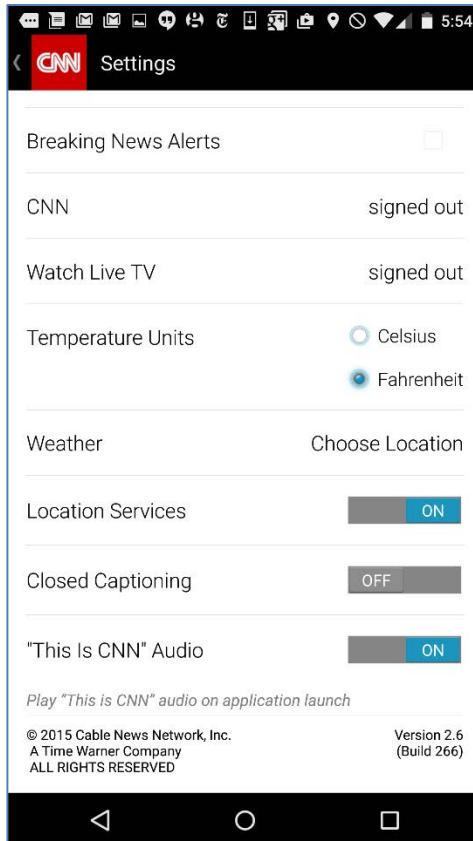
BBC News for Android: Launch screens similar to the first screen make the experience seamless and give users time to get used to the app.

333. Don't start with a video or a sound effect.

Users don't expect sounds when starting a new app — they're often startled by the sounds, at best, and annoyed and disturbed, at worst. (Mobile usage can happen anywhere, and apps should be particularly cautious with noises — think about a sudden noise in the middle of a meeting or late at night, when everybody else in the house is asleep.)

My TSA made a noise when the app was launched. One of our users was surprised and wondered what the noise was. Al Gore's Our Choice started with a video of Al Gore the first time the app was launched; on subsequent sessions, it played music while downloading the data.

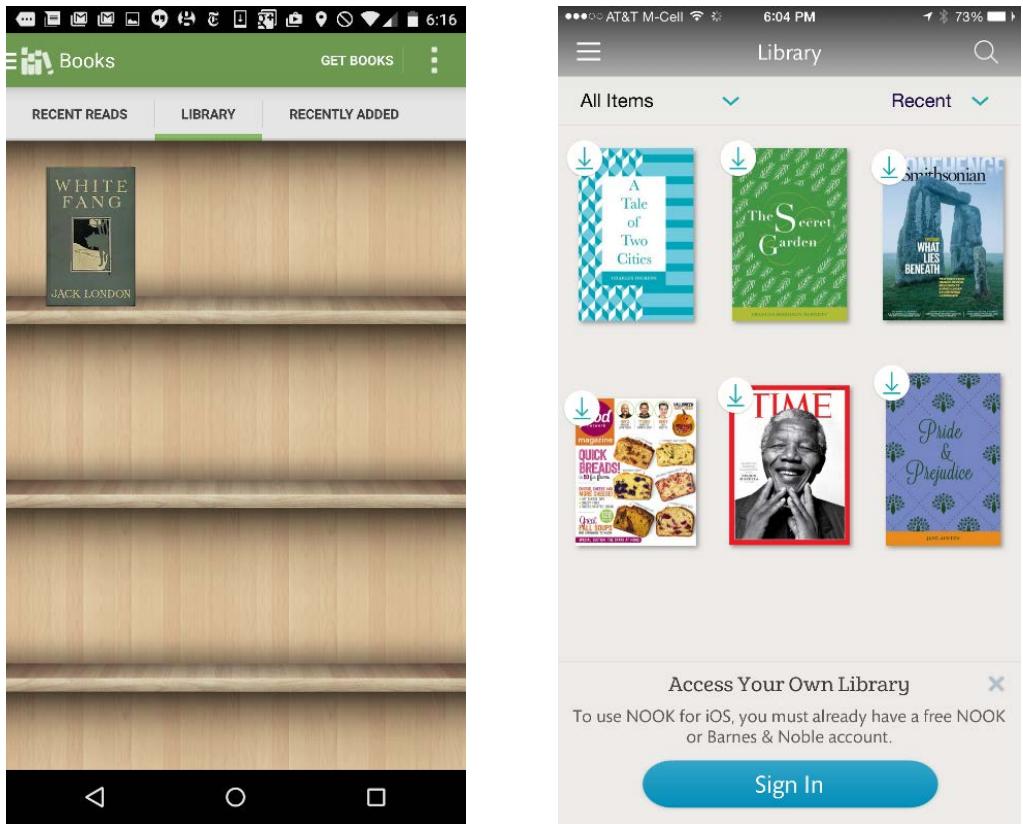
CNN had an option for turning on or off an audio "This is CNN" message upon app launch. This option used to be on by default in previous versions of the app; luckily, in more recent versions, the default has been changed to off. We can hardly imagine anybody wanting to turn it on.



CNN for Android: The app had an option to turn on or off an audio message on app launch.

334. Preload data for the first launch.

When users have just downloaded your app, you have to entice them and convince them that your app is worth keeping and using. Don't make them work by asking them to download content or to register — show them how great your app is by having them check some preloaded content. Book-reader apps such as Aldiko come with 1–2 preloaded books that give users a chance to try the app and see if they like.



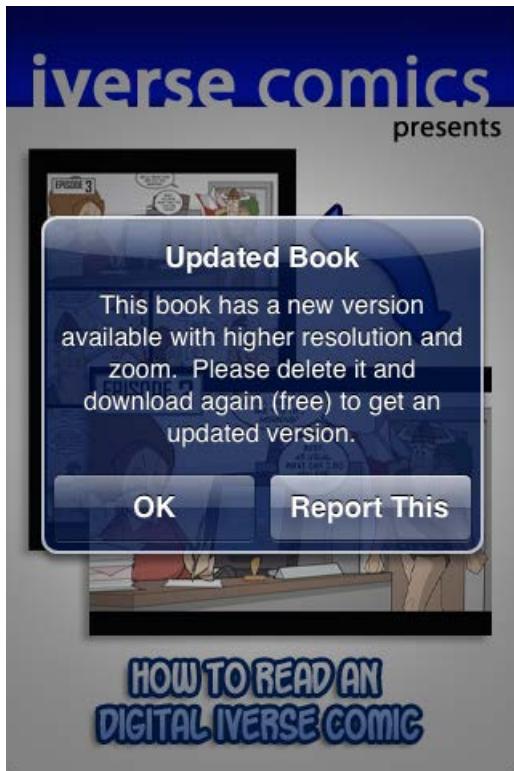
Left: Aldiko for Android had already preloaded some content in the app, so that when people used it for the first time, they could see how they like the app without having to work to download a book. Right: Barnes and Noble came with a suggested “library” for people who were not signed in, but it did not preload any of these books.

335. Update data for the users.

336. Don't ask users to delete an app and install it again.

This is an instance of a basic usability principle: do as much work for the users as you can. Asking users to delete a book and then download it again requires them to put in too much effort — they'll have to find the book in the store again, possibly enter their credentials, and wait for the content to download. It's so much easier if the app allows them to skip at least some of the steps and does the update for them.

It's even worse to ask users to circumvent the regular application update process and delete the app, then install it again. Unless users deeply care about your app, they are very unlikely to do so.



(a) iVerse Comics for iPhone
(older)



(b) Fluent News for iPhone

Do the work for the users: don't ask them to delete a book (left: iVerse comics for iPhone) or an app (right: Fluent News for iPhone), and then reload it.

Methodology

The guidelines discussed in this report are based on several studies using three different methods: diary, traditional usability-testing, and design reviews. Next we discuss each of these types of studies individually.

DIARY STUDIES

Overview

The purpose of the diary studies was to understand the range of activities that people perform on their phones. We carried out two separate diary studies: one international and one in the US. The international diary study involved people from six different countries in Europe, Asia, Australia, and America, who owned different types of phones (feature phones and smartphones, including touchscreen phones). For the second diary study, we focused on iPhone owners in the US.

For the first diary study, we were less interested in usability problems; therefore, we recruited participants with relatively advanced technical skills and did not impose any of the typical occupational restrictions that are used when recruiting for traditional usability testing (e.g., no IT-related occupation). For the second study, we recruited users who did not work in marketing or any IT-related occupation.

Participants recorded every activity that they did on their mobile phone (except for talking or text messaging) for one to two weeks. At the end of each day, they completed a questionnaire that detailed the context of each of the mobile activities that they had performed during the day. Users commented on:

- Where and when they had done the activity
- What they were trying to accomplish and whether they were successful
- What kind of websites or applications they had used to accomplish the activity
- Whether they had encountered any problems or had any comments about their experience

At the end of their participation in the diary study, we conducted a short interview with the participants over the phone. In the second study, instead of the phone interview, participants came to the lab for a regular usability-testing session.

Participants

The first diary study involved 14 different participants from 6 countries (Australia, Netherlands, Romania, Singapore, UK, and USA). The second study involved 13 participants from USA. The table below shows the country and occupation of each participant, as well as the number of days they participated in the study.

COUNTRY	OCCUPATION	DAYS OF PARTICIPATION
First study		
Australia	Business administration	6
Australia	Market research	4

Australia	Director of community innovation	7
Netherlands	Self-employed	10
Netherlands	Student	7
Romania	Business administration	6
Romania	Director	6
Singapore	Engineer	6
Singapore	Bank executive	7
UK	TV producer	3
USA	Bookkeeper	6
USA	Director	5
USA	System administration	8
USA	Sales	6
Second study		
USA	Sales manager	7
USA	Controller	7
USA	Administrative assistant	7
USA	Clinical specialist	7
USA	Football coach	7
USA	Reading Instructor	5
USA	Public safety officer	7
USA	Elementary teacher	7
USA	Probation officer	7
USA	Homemaker	7
USA	Operations support	7
USA	Student	3
USA	Classroom tech specialist	7

Method

We instructed participants to create an account on twitter.com (henceforth called *Twitter*). *Twitter* is a microblogging service that allows each user to post short messages; the messages are further broadcasted to all other *Twitter* users who opted to receive updates from that particular user (i.e., to “follow” that user).

We followed each of the diary-study participants on *Twitter*. We instructed the participants to send a *tweet* each time they used their mobile phone for a nonvoice or nontexting activity. At the end of each day, participants completed a questionnaire that targeted the context of the mobile activities that they had done during the day. These are the questions used in the questionnaire:

1. Date and time of activity (include AM or PM)
2. Where were you at the time?
3. What were you doing at the time?
4. What did you do (visit a website, send an email) and what were you trying to accomplish? Please be specific.
5. What website or application did you use? For a website: how did you get there: did you use bookmarks/favorites, search, typed in the URL, clicked on a link, or some other method (specify)?
6. Have you used that phone application or website before (on your phone)?
7. If applicable, why did you go to this particular site/app versus another similar site/app? Please be specific.
8. Did you accomplish what you wanted to do at the time? Why or why not?
9. What did you like about the site/phone application? We are interested in anything that made it easy for you to accomplish your task. List and briefly explain each thing that was easy to use or that you liked.
10. What did you dislike about the site/phone application? Think of anything that made it hard for you to accomplish your task. List and briefly explain each thing that was hard to use, confusing, or that you did not like.
11. Any other comments?

In the first study, participants emailed the responses to the questionnaire. In the second study, we used Google Spreadsheets to generate an online form that was filled in by the participants each day.

USABILITY TESTING

Overview

We carried out 15 separate usability-testing studies over 5 years. Seven of the studies took place in the US; the other three were done in Australia, Hong Kong, the Netherlands, Romania, and the UK. All of these were traditional usability studies using the think-aloud methodology. The purpose of these studies was to understand the typical usability issues that people encounter when doing web-related tasks on mobile phones. The first two studies involved all types of phones (feature phones, smartphones, and touch phones). Further studies involved a combination of nontouch smartphones and touch phones. In the recent years, the term "smartphone" has become a synonym for "touch phone", and therefore our newest studies only looked at touch phones only. Moreover, the last of our studies exclusively involved phablets (touchscreen smartphones with screens larger than 5.3 inch).

Where applicable, we asked participants to show us the apps that they had installed on their phones, and then we gave them tasks to complete. The tasks involved either mobile apps or the web. For all the tasks, participants used their own phones.

A moderator sat next to the participant, and observed, listened, and took notes. Users commented on:

- What they were looking for or reading
- What they liked or did not like about the site
- What made it easy or difficult for them to accomplish the task

Some of the tasks involved specific websites or apps; others were open ended. The moderator observed users as they worked and encouraged them to think aloud. As part of the sessions, she also interviewed participants about their common mobile practices and asked them to demonstrate some of their favorite sites and applications.

For 14 out of 15 studies, the participants' interaction with the phone was recorded using a document camera. For one study we used a mobile-device camera (kindly provided by Ovo Studios), that was mounted on the phone. Both the document camera and the mobile-device camera allowed the participants to hold the phone in their hand.

Each session lasted between 60 and 90 minutes.

Participants and Devices

A total of 151 people participated in these studies, out of which 124 were from the US and 27 from other countries. The studied phone distribution was as follows:

TOUCH PHONES				SMARTPHONES (NONTOUCH)	FEATURE PHONES
IPHONE	ANDROID	WINDOWS PHONE 7 OR 8	OTHER		
57	36	9	7	28	14

69 participants were males and 65 females. The age demographics was as follows:

20–30	31–40	41–50	51+
55	43	34	19

All participants used their mobile phone several times per week for activities other than texting or talking. We screened out for technical experts and people who worked in usability or marketing, since they were not the target users for the sites that we tested and tend to exhibit atypical behaviors due to their expertise.

Following is a sample of participants' occupations:

- Administrative assistant
- Accountant
- Business development manager

- Business owner
- Consultant
- Criminal investigator
- Director of administration
- Event planner
- Fashion consultant
- General contractor
- Hair stylist
- Homemaker
- Management analyst
- Operations manager
- Patent lawyer
- Programme leader
- Property data analyst
- Real-estate agent
- Recruitment adviser
- Retail manager
- Retired nurse
- Sales engineer
- School counselor
- Server
- Staffing manager
- Student
- Teacher
- Television producer
- Travel agent
- Warehouse worker

Method

Each session was divided in three parts:

1. The participant showed the moderator which applications or sites she was using on a regular basis on the mobile device.
2. The participant demonstrated applications or websites that she used more frequently. (The facilitator chose the applications or websites, based on the information given by the participant at the beginning of the session.)

3. The facilitator gave the participant one task at a time and asked him to carry out each task as far as he would if he was on his own. The participant was encouraged to think aloud while performing the task.
4. Some tasks were specific to an app or website; others were general information-finding tasks (e.g., "Find where the word *dollar* comes from").

Each participant saw a subset of the available tasks. Where applicable, the tasks were balanced across participants so that each task would be performed by participants with all types of phones (feature, nontouch smartphones, or touchscreen smartphones). The order of the tasks was randomized for each participant.

Materials

The following tables show a sample of tasks that we used for these studies.

GENERAL TASKS (NO SPECIFIC APP OR WEBSITE)
Find the symptoms of swine flu and what you should do to avoid getting sick.
Check the local weather forecast for tonight.
Find out what is on BBC 1 TV tonight at 8 p.m.
It's 6pm and you need to get from West Kensington to Tufnell Park. You decide to go by tube. Find out the best way to get there, changing as few tube lines as possible.
Find out which is the tallest building in the world.
You just found the Panasonic digital cordless phone model KXTG1102 in store for GBP 35. Find out if you can find a better price online.
Find some reviews of the same phone (Panasonic digital cordless phone model KXTG1102).
You are traveling to Berkeley, CA and want to make a reservation at the restaurant Chez Panisse. Find out what their menu for the week is.
You are playing Monopoly and having an argument with your friend over how many turns you may stay in jail before you need to pay \$50 to get out. Use the internet to find out the correct answer.
You and your friend are having an argument about the origin of the word "dollar." Find out where it comes from.
Use the web to find out what "carotid stenosis" means.
You are traveling to San Francisco and want to eat out at a restaurant called "Jardiniere", close to the San Francisco Opera. Find some trusty reviews of the restaurant.
Find out the most recent NBA (basketball) scores.

Find which movie got the Golden Globe award for best picture this year.
Your friend's 6-year old daughter has never heard of Tom and Jerry. Find a Tom and Jerry video cartoon to show her.
You and your friend want to watch the movie "Slumdog Millionaire". Find a movie trailer for that movie.
Your friend wants to watch a movie on TV tonight after 8pm. Find a listing of tonight's TV program and identify a movie that she may want to watch.
You and your vegetarian friend want to find a good Indian restaurant nearby. Use the web to locate one that you may want to go to and that serves vegetarian food.
Find out how to get to the Indian restaurant that you just found.
You are in Chicago, IL and need to call a cab for a ride to the airport. Find a phone number for a taxi company that operates in Chicago, IL.
How many calories are there typically in a slice of thin-crust cheese pizza? Find out how that compares with a slice of regular cheese pizza.

WEBSITE-SPECIFIC TASKS	
URL	TASK
lufthansa.com	You are planning a business meeting next week in Frankfurt, Germany. See if you can catch a flight back to London after 6pm.
qype.co.uk/mobile	Find an Italian restaurant in Soho that has good reviews.
	Find where the restaurant is and how you can get there from your current location.
pda.skysports.com	Find out the score in the latest football match played by Newcastle in the Premier League.
interflora.co.uk	Find out the delivery rates for a bouquet of roses that need to be delivered the same day.
	Order a bouquet of flowers under GBP 40 for your friend's birthday; she should get them tomorrow by noon (complete all steps before actually making the purchase).
bloomberg.com	Find the current stock price of Renault (UK).
Homedepot.com	Where is the closest Home Depot from your current location? Find out directions to that store using

	<p>homedepot.com.</p> <p>Use HomeDepot.com to find out how to install a new faucet.</p>
Nordstrom.com	Where is the closest Nordstrom from your current location? Find out directions to that store and opening hours using Nordstrom.com.
fandango.com	You want to see the movie "Revolutionary Road". Use fandango.com to find out if and at what time it's playing tonight at a theater close to your house.
	Find some movie critic reviews of the movie "Revolutionary Road" at Fandango.com.
Wikipedia.org	Find out where the word "spa" comes from, using the site mobile.wikipedia.org.
Abc.com	A friend is a fan of the ABC TV show "Lost". Use abc.com to find out if there are any episodes in Lost Season 5 that can be watched for free.
weather.com	You are planning to leave for Paris, France next weekend. Find out what the weather is going to be like using weather.com.
yelp.com	Use yelp.com to find reviews of the San Francisco restaurant "Absinthe".
	Use mobile.yelp.com to find out a good Mexican restaurant in Fremont CA. 17c. Find directions to the Mexican restaurant that you just found on yelp.com.
espn.go.com	Use espn.go.com to find out if there are any NBA basketball games that you could watch tonight on the ESPN TV channel.
	Use mobile.esp.com to find the latest scores from the Australian Open (tennis).
Adorama.com	You are in an electronics store and consider buying a Canon PowerShot SD1100IS as a present. The camera costs \$220.25 in the store. Check adorama.com to see if you can get a better price online.
WineSpectator.com	You and your friend are celebrating his birthday at an expensive restaurant. He has chosen duck leg with duck confit for the main course and wants to select a Cabernet Sauvignon to go with the dish. Here is the restaurant's selection of Cabernet Sauvignons. Please use mobile.WineSpectator.com to recommend a wine to your friend.

		Look up this bottle of wine in WineSpectator.com.
PLATFORM	APP	TASK
iOS	Fandango	Find a movie that you may want to watch for next Friday. Buy tickets to a theater near you (stop short of actually buying the tickets).
iOS	JC Penney	You have \$50 dollars to spend on a piece of clothing for yourself. Find something that you might like.
iOS Android Windows	Best Buy	Find a car navigator for a friend. Figure out if you could buy it online or pick it up in a nearby store.
iOS	My TSA	You have a flight into Chicago's O'Hare airport. Find out how many flights arrive on time and how many departures are delayed at that airport. Can you find statistics for last year?
iOS Android	1800flowers	Buy a bouquet of white roses for your friend. (Stop short of actually completing the purchase).
iOS Android	Target	Your friend asked you to buy a Method-brand kitchen cleaner for her. Find out if they have it at your nearest Target.
iOS Windows	Huffington Post	Check the latest entertainment news. Email a story to yourself. Can you find any picture-based articles?
iOS Android	Zappos	Find a pair of comfortable shoes for yourself.
iOS Android	Walgreens	Find a moisturizer with SPF 30 or above that is suitable for your skin.
iOS	app.ft.com (Financial Times web App)	Check the latest world news. Read an article that is of interest and email it to yourself.
iOS Android	Edmunds	You are in the market for a new car. You want a 4-door car with 5 seats (or more) and a high mileage per gallon. Figure out what options you have.
Android	Cardio Trainer	Build an exercise schedule: Mon 30 min running at 8pm, Tue 45 min cardio; Wed break; Thu 45 min weights; Fri break; Sat 30 min running.
Android	News and Weather	Check the latest news in technology.

Windows	Amazon	Find a birthday gift for yourself.
Android Windows	ESPN Score Center	Find the latest baseball scores and news.
Windows	Movies by Flixter	Find a movie that you may want to watch for next Friday. Buy tickets to a theater near you (stop short of actually buying the tickets).
Android Windows	170,000 recipes big oven	You have zucchini and tomatoes in your refrigerator. Find a dish that involves both and that you could cook tonight. Save the recipe so you could come back to it easily.
Windows	USA Today	Find the latest news-related photos.
Windows	CNN Money	Check the latest financial news.
iOS Android	HK Movie(s)	You want to watch "Black Swan" next Saturday. Find a movie theater where you can watch it after 7pm.
iOS	Pizza Hut	Buy a pizza of your choice (stop short of actually buying it).
iOS Android	AA stocks	Find the current stock value of China Mobile. How did the stock change during the past month?
iOS Android	China Daily	See if you can find any interesting pictures related to today's news.
iOS	Myer	It's your friend's birthday and you want to buy her a nice bracelet under \$100. Complete all the steps and stop just before actually buying the bracelet.
iOS	Coles	You want to buy some pasta, diced tomatoes and ice cream. Create a list that contains all those items.
iOS	realestate.com.au	Figure out what the range of prices is for a 3 bedroom 2 bathroom apartment in Melbourne. How about for an apartment in this neighborhood?
Android	Australia news	Check the latest sport news.

For many tasks, we did not specify where users should go. Thus, the first part of their task was to find one or more appropriate websites. Some of the website-specific

tasks had URLs associated with mobile sites; others did not. For some websites, we tested both the mobile and the full (desktop) website on the mobile device. We sometimes gave participants the URL of the site without indicating the mobile variant, in order to test whether the site will recognize access from a mobile phone and will redirect the user to a mobile web page. In cases where this redirection did not happen, we still found it valuable to see what users experience when they are not aware of the web address for the mobile site.

In the case of app-specific tasks, users were asked to download the app first and then use it to complete the task. We tested both free and paid apps; participants were reimbursed for the paid apps (unless they already had these apps installed on their phones).

DESIGN REVIEWS

The last source of information for our guidelines came from expert reviews. Over the course of three years we reviewed many apps and mobile websites on a variety of platforms.

The following table contains a subset of the sites and applications included in the review:

m.cnnmoney.com	Financial news from CNN
m.TVGuide.com	TV schedule
Freshdirect.com	Grocery store in NY
mobile.movietickets.com (iphone.movietickets.com)	Movie schedules and movie tickets
mobile.VictoriasSecret.com	Clothing and lingerie store
Wsj.com	Wall Street Journal
Yahoo.com	Yahoo! portal
Flickr.com	Image repository
Zillow.com	Real estate
realestate.com/mobile	Real estate
eBay.com	Online auctions
southwest.com	Airline
craigslist.org	Classifieds
CNet.com	IT News
Facebook.com and Facebook applications for iPhone and Blackberry	Social Networking
Squareup.com	Credit-card swiping device
Nba.com	Sports

1800flowers.com	Flower shop
Sears.com	Sears online store
cnn.com	News
bbc.co.uk	News
ShopSavvy for iOS/Android	Price comparison
Netflix	Entertainment
The Weather Channel for iOS/Android	Weather
ATT U-Verse for iOS	DVR app for subscribers of ATT U-verse
Sears for iOS	Shopping
REI Snow Report for iOS	Weather (ski resort information)
British Museum for iOS	Museum app
Moleskine for iOS	Note taking
Bank of America for iOS	Banking
AP Mobile for iOS/Android	News
Astrid Tasks for Android	To do app
CNBC for iOS/Android	News and stocks
Paper for iOS	Social News
Skyscanner for iOS/Android	Flight search engine
Yummly for iOS	Recipes
Instacart for iOS	Grocery shopping
Sephora for iOS/Android	E-commerce

List of Guidelines

Making Your Mobile-Dedicated Website Findable	39
1. [Web] Redirect users to the mobile site if they access your site on a mobile phone.	40
2. [Web] Give users the mobile version of the requested page; do not always redirect users to the mobile homepage.	41
3. [Web] Don't make users decide if they should use the full site, the mobile site, or mobile app. Always direct the users to the mobile site.....	41
4. [Web] On your desktop site include a link to your mobile site.	42
5. [Web] On your full site, the link to the mobile site should be placed either in a very salient location (e.g., navigation at the top of the page) or in the page footer.	44
6. [Web] The link labeled <i>Mobile</i> on your full site should not point to information about your mobile site.	44
7. [Web] Standard domain names and URLs (m.site.com, mobile.site.com, site.mobi, www.site.com/mobile) should all point to your mobile site. If you can afford only one of these domains, use m.site.com.	45
8. Include a link to the full site on the mobile-dedicated homepage.	46
9. The link to the desktop site should be labeled <i>Desktop site</i> or <i>Full site</i>	48
10. [Web] A responsive site should also have a link to a "full" site in its lower-screen version.	48
11. On deep pages of your mobile site, include a link to the corresponding page on the full site.....	48
 Making Your App Findable	49
12. [App] Choose app names that are unique, recognizable, and memorable.....	49
13. [App] If your brand name is recognizable, make it part of the app name.....	49
14. [App] If your brand name is not part of the app name, your app should be one of the search results in the app store and on the phone when people search for your brand name.	50
15. [App] If your brand name is not part of the app name, consider incorporating your logo in the app icon.....	51

16. [App] Choose an application icon that is descriptive and easy to recognize.....	51
17. [App] Use the same (or almost identical) icons for all versions of your app if you offer different apps for phone and tablets.....	52
18. [App] Your app description in the app store should clearly explain what the purpose of the app is and if it has any special features compared with other similar apps (and especially, compared with the free or paid version of the same app).	53
19. [App] Your app description should be concise and scannable.....	53
20. Advertise the app on your mobile website. Make sure that you promote the right platform on the right device.....	55
21. Advertise the app as a link on your page rather than on a separate page.....	55
22. If users have the app installed on their phone, the link advertising the app should take them to the corresponding page in the app.....	56
Mobile-App Functionality.....	57
23. [App] Find a core function and design your app around it.	58
24. [App] Keep the app functionality simple.....	58
25. [App] Don't add features that are unrelated to your core function.	58
26. [App] Use separate apps if the tasks that you plan to support are different and performed at different times.....	62
27. [App] Use separate apps if the audiences that you plan to support are distinct (that is, most people will be in a single audience category).	62
28. [App] For the phone version of your desktop app, prioritize those features that revolve primarily around consuming rather than producing content.....	63
29. Allow users to resume their task on a different platform.....	63
30. Consider emailing a link to users to let them finish on a different platform a task started on mobile.	64
Logo and Company Name	65
31. [Web] On websites, include the company logo or name in a salient location at the top of the mobile homepage.....	65
32. [Web] On websites, include the logo or name on every deep page of your site.....	65
33. [Web] The logo should take a minimum amount of space on the small mobile screen.	65

34. [App] An app generally does not need a logo on the homepage or elsewhere.	67
35. [App] If you estimate that other apps will give control to your app, consider including the logo on every page of your app.	67
36. The company logo or name should link to the homepage.	68
Touch Targets	69
37. The target area (i.e., tappable area) for touchscreen devices should be at least 1cm X 1cm.	69
38. Do not crowd targets. Leave generous amounts of space around UI elements such as radio buttons, arrows for drop-down boxes, checkboxes, scrollbars, and links.	69
39. Do not rely on small padded targets.	72
40. Use padding for tabular views.	73
41. You can use 3D cues to make targets look tappable.	75
42. You can use borders and color to make targets look tappable.	75
43. To signal tappability follow target-placement conventions on your platform.	75
44. Be consistent in your treatment of targets within your app or website.	75
45. Use different visual styling for tappable and nontappable design elements.	78
46. Give users a way to undo the last action.	80
47. [iOS, Apps] Include a <i>Back</i> button into your iOS app.	80
48. Consider placing destructive buttons far away from the physical buttons.	81
49. Always ask for confirmation before carrying out a destructive action.	81
50. Don't use a skeuomorphic app if you cannot make it completely coherent with the users' expectations.	84
51. Even if the app is not completely skeuomorphic, well-selected skeuomorphic elements can make it more enjoyable.	86
52. Make sure that the skeuomorphic elements make the users' task easier rather than harder.	86
Icons	90
53. Use standard icons whenever possible.	90

54. Use standard icons in standard ways.....	91
55. Use icons with good information scent.....	92
56. Include labels with your icons.....	93
Typing and Input Fields.....	94
57. For any input field, consider if you could eliminate it and save the user some work.....	95
58. Consider eliminating optional fields from mobile forms to make the form more compact.....	95
59. Where possible, compute field values rather than asking the users to enter them.....	96
60. Make textboxes long enough so that users don't have to scroll within.....	98
61. If you know how long a certain field is going to be, make it that size.....	99
62. Place description text above the textbox, not in line with the textbox.....	100
63. Do not use placeholders inside textboxes.....	101
64. Allow users to copy any content that you may present to them.....	102
65. Allow users to paste information into an input field.....	102
66. Present users with the keyboard appropriate for the input field.....	103
67. Use autocomplete and suggestions whenever users fill in a textbox.	104
68. Do not use autocomplete and suggestions for fields that are highly singular — such as email or names. (Do use history or other information that you may have for those fields.)	104
69. Allow for typos and abbreviations.....	105
70. Autoformat fields for users. Don't force them to use characters such as dash or space to achieve a specific format.	106
71. Do not make people memorize information from one page to another.	107
72. Use personalization and history to provide good defaults and suggestions for text that needs to be input.	110
73. Allow users to easily delete default field values.....	113
74. Use QR codes to allow users to easily access websites and content on mobile.	114

75. Use GPS information to detect location automatically and allow the use of current location.....	116
76. Allow users to edit the location provided by the GPS or other info captured via phone features.....	117
77. Use the camera to fill in information where possible.....	118
78. Use voice recognition to help users input information.....	120
Spinners, Sliders, and Other Selectors	121
79. Do not use sliders for inputting precise values. Only use sliders when the exact value is not important.....	121
80. Do not use spin wheels for items that don't have a natural order.....	123
81. Do not use color to communicate the active choice.....	124
Menus: Drop-downs, Accordions, Overlays, Sliding Menus, Pie Menus	127
82. Make sure that your menus look expandable.....	135
83. Use large targets for arrows associated with drop-downs or other menus.....	137
84. Use menu labels with good information scent.....	138
85. Do not label a menu with one of the options inside it.....	138
86. Make sure that all the text of each option in a menu is visible.....	140
87. [iOS] On iOS use drop-downs only for short lists of 4-6 items.....	141
88. [Android] Minimize scrolling in a drop-down box by using the whole screen to display long lists of options.....	141
89. Avoid using overlays.....	142
90. Make the content in the overlay clearly distinguishable from the content in the background.....	143
91. For accordions and overlays, make the menu content scrollable to ensure that the entire content is accessible on a variety of screen sizes and screen orientations.....	145
92. When the overlay is closed, take users back to the place on the page where they were when they opened the overlay.....	148
93. Consider forcing people to tap on a <i>Close</i> icon or button to dismiss the overlay, in order to ensure that the menu is not closed by accident.....	148

94. Prefer modal to nonmodal overlays or sliding menus to avoid accidental touches	149
95. For accordion menus, make the menu content scrollable to ensure that the entire content is accessible on a variety of screen sizes and screen orientations	151
96. Do not close the accordion when people touch other areas of the screen that are not part of that accordion.....	151
97. Consider forcing people to tap on a <i>Close</i> element or button to dismiss the accordion, in order to ensure that the menu is not closed by accident.	151
98. If the content under an accordion is really long (spanning several pages) consider allowing users to quickly close the accordion without scrolling back to the beginning.	152
99. [Web] Allow users to close the accordion using the browser <i>Back</i> button.	153
100. By default accordions should be closed.	154
101. Use a plus or down arrow icon to indicate to users that the accordion will expand in place.	154
102. Prefer modal to nonmodal sliding menus to avoid accidental touches.	155
Carousels	156
103. Users should reach the last item in the carousel in 3-4 steps (e.g., taps or swipes).	156
104. Do not use animated carousels. Use carousels that can be controlled by users.	158
105. For hero carousels that are used for featured content on the homepage, make sure that users can access that content in some other way as well, and not only through the carousel.	159
106. Group carousel controls with the carousel content.	159
107. When users select a control that acts upon the information displayed in the carousel, make sure that they can see the carousel content.	161
108. Allow users to use the swipe gesture to navigate through the carousel.	162
109. [iOS] Allow some swipe space (e.g., a page gutter) around carousels to avoid swipe ambiguity.	162
110. Use arrows or the illusion of continuity to show users that they can see more items in a carousel.....	164

111. Do not use dots or lines to indicate pages in a carousel.....	164
Buttons and Button Placement	166
112. Label buttons in a descriptive way.....	166
113. Group related controls (and related information)	167
114. On a page, do not place main calls-to-action at the top of the page, in the tool bar or navigation bar.	171
Errors and Alerts	173
115. Don't use alerts, fading popups, or separate pages to signal an error in a form.	173
116. Don't use alerts, popovers, or separate pages to display instructions for filling in the form, or other information that the user must refer back to.	175
117. Use toasts only for feedback that is not crucial; do not use them for error messages.	178
118. When telling users how to fix a problem, always offer them a link to where they can do it.	180
119. In a form, mark fields that need to be corrected to signal an error.....	181
120. In a form always show the error message. Do not hide the error message and expect the users to uncover it.	184
121. An error message should tell users (1) what the problem is, and (2) what they can do to fix it.	185
122. After reporting an error, return to the state before the error.....	185
123. Don't report an error regarding a feature that is not in use.	186
Notifications.....	188
124. [App] Do not ask people to accept push notifications the first time they launch an app.....	188
125. [App, Android] Do not turn on notifications by default.	189
126. [App] Avoid turning on sound notifications unless people explicitly agree to them.	190
127. [App] Make sure that people can easily turn notifications on or off.	191
128. [App] Allow users to turn notifications on or off within the app.	191
129. [App] Do not send notifications in burst.	192

Progress Indicators.....	193
130. Do not use static progress indicators.....	193
131. Use a dynamic progress indicator for every action that takes longer than 1 second.	195
132. Whenever your app is downloading content such as magazine articles, videos, slideshows, show a progress indicator.	195
133. Whenever the action might take more than 5-10 seconds, use a progress bar instead of a spinning gear or other indicators that don't show real-time progression.	196
134. For larger downloads (e.g., videos, magazines), give users an estimate of how long the download it's going to take.	196
Orientation.....	197
135. Support both phone orientations (landscape and portrait) whenever possible.	197
136. Don't make users switch between orientations often.	197
137. Do not display different content (or different interface elements) in different orientations.	198
138. Do not ascribe different content to the two possible landscape orientations (physical buttons left or right).	200
139. You can use the landscape orientation to fit extra detail in a data table, spreadsheet, or other type of graphic.	201
140. Whenever the landscape orientation is used for extra information, you should let users know (e.g., by displaying a tip or a message).	201
Gestures.....	202
141. Use only the more familiar gestures. Do not rely on users knowing and remembering more complicated gestures.	203
142. Multi-finger gestures (other than pinch) should only be used as accelerators for features that can also be accessed in other ways or are not needed by non-expert users.	204
143. Take advantage of the natural affordances for gestures.	206
144. Use contextual tips to make gestures more discoverable....	207
145. Use progressive disclosure to make gestures more discoverable and avoid overwhelming people.	207
146. In most situations, users need cues to figure out that they're supposed to use the horizontal-swipe gesture for horizontal navigation.	209

147. To invite users to scroll down a page, avoid creating false floors.	210
148. Do not assign unconventional meanings to standard gestures.	211
149. [App] Avoid using many different gestures within the same app.	212
150. [App] Avoid using many similar gestures with different functionalities within the same app.	213
151. If you must use less familiar or less discoverable gestures, add some redundancy in the interface.	214
152. Avoid swipe (or other gesture) ambiguity: Do not assign multiple meanings to the same gesture depending on where the gesture is deployed on the page.	218
153. To minimize swipe ambiguity, separate the location of the swipes as much as you can. For carousels or maps, place them far from the vertical or horizontal edges of the screen.	220
154. In content-reader apps, it should be possible to turn pages both by tapping the page margins and by swiping horizontally.	221
155. When displaying images (or video) full screen, allow users to get to both controls and captions by tapping on the image.	222
156. When displaying images full screen, start by showing both controls and caption, then fade them away.	222
Forms	223
157. In a form always specify whether a field is required (preferably by using an asterisk).	223
158. In a form specify whether a field is optional.	225
159. Minimize the number of page loads in a form.	225
160. For multistep forms, give users a mental model of the form by showing an overview of the form steps (e.g., a workflow diagram or a list of the steps).	229
161. Use collapsed accordions to represent different steps of a form.	232
162. When using a list of steps in a form, force users to complete the different steps in a fixed order.	233
163. The form Submit button (or equivalent) should be displayed under the form fields rather than at the top of the page.	235
164. In a form always include a Submit button.	238
165. Distinguish between link buttons and Submit buttons. Submit buttons are usually placed at the bottom of the form; when link buttons are positioned there, they can create confusion.	238

166. For transactional forms, show a one-screen summary of the transaction before the user needs to commit the transaction.	239
167. Do not automatically subscribe users to newsletters or other email. Allow users to opt in, instead of requiring them to opt out.	241
168. Whenever users make a transaction on the phone, email them a confirmation number for that transaction. Inform them that they will receive an email and tell them what it will contain.....	241
169. Allow users to “soft print” the transaction confirmation screen by taking a screen capture.	241
170. Don’t have a login wall when the app is first launched. Deliver value before asking users to log in.....	242
171. Do not ask users to log in or register unless absolutely necessary for security reasons.....	244
172. Avoid forcing users to log in or register for authentication purposes only.....	244
173. [App] Do not ask people to register or log in to save information such as favorites.....	245
174. Do not force people to sign in or register to check out. Allow them to use a guest checkout.	247
175. The guest checkout option should always be the first option (before log in or register) displayed on the checkout page.	249
176. Offer people the option to sign in with a Facebook or Google account, but don’t make those be the only options available.	250
177. [iOS] Offer users the option to log in using their Touch ID.	252
178. [iOS] Do not force users to log in using their Touch ID.....	252
179. Allow users to checkout with PayPal, Apple Pay, or Google Wallet, but do not make those the only options.	253
180. Place both the <i>Sign In</i> and <i>Register</i> buttons under the login form, one next to each other on the same line.	254
181. If users make an error trying to sign in when they don’t have an account, take them to the registration form, but save the information that they had entered in the login form.	258
182. When logging in and out must be done for each session, let users define a numerical pin.	259
183. When logging in must be done, have an option that allows the user to see the password in clear.	260
184. When users need to create a password, tell them if the password has to satisfy any constraints (e.g., password length).....	261

185. A login form should include a <i>Forgot password?</i> link to allow users to recover their password.	263
186. If users must submit an email address to recover their password, tell them if the email is not in your database.	264
187. Avoid taking users out of the app or website to recover their passwords.....	265
188. If user made an error in the log in or registration form, do not clear the password field. Do not make them reenter their passwords from scratch.	267
189. If you have a login wall, allow people to register within the app or mobile website.....	268
190. Be upfront about the benefits that registration brings to the users.	269
191. [App] Avoid taking people out of the app for registering.	271
192. [App] If registration is not fully supported in your app, consider using a browser view within the app to display the full site. Once users finish the registration, take them back to the flow they were in before starting it.	271
193. Don't ask users to confirm their registration through email.....	272
194. If users have just completed registration in an app or on a website, they should be considered logged in and should not have to log in again.....	273
195. Whenever registration is an option on mobile, require only the minimum information to create an account. Let users add extra information at later times.	273
196. Remove duplicate fields (email, password).....	277
Navigation.....	278
197. Tab bars and navigation bars are well suited for sites with relatively few navigation options.....	282
198. Do not use more than 5 options in a tab bar or top navigation bar.	282
199. Use carousels for a navigation bar or a tab bar only if the categories in the bar are similar and highly predictable.....	283
200. If you decide to use a navigation bar or a tab bar, they should be the main chrome area of the screen and little extra space should be devoted to other utility-navigation options or to search.	284
201. Homepage as navigation hub works well for task-based sites or apps where users are unlikely to do more than one task in a single session.....	285

202. On homepages of content-heavy, browse-mostly sites or apps, give priority to new content over navigation links.....	286
203. The navigation menu makes the navigation options least discoverable and is best suited for content-heavy, browse-mostly sites and apps.....	286
204. Unless your site or app is task based, consider allowing users to access the main navigation on every page.....	286
205. If your mobile pages are long, consider using persistent or floating navigation at the top of the page or, alternatively, a floating back-to-top control to rapidly take users back to the top of the screen.....	288
206. Avoid using both persistent navigation and a back-to-top control.	291
207. Place the “scroll-to-top” control close to the bottom right corner of the screen.....	292
208. Avoid navigation hierarchies deeper than 2 levels on mobile.....	294
209. If your site IA has more than 2 levels, consider flattening the navigation hierarchy on mobile by merging some of the IA levels.	294
210. If some IA levels have been flattened in the mobile navigation hierarchy, provide filters, facets, and in-page navigation to allow users to recover them should they need to.	296
211. Use breadcrumbs on sites with a deep navigation structure (more than 2 levels). Do not use breadcrumbs on sites with shallow navigation structures.....	299
212. If your main navigation is implemented as a menu, you can use an in-situ accordion for a short (under 10 items) second-level submenu.	300
213. If your second-level submenus are long (more than 10 items), consider using a category landing page to display them.	300
214. If you use an accordion to implement a submenu, make sure that the submenu categories are styled differently than the top-level categories.	302
215. Always display the top-level categories in a navigation menu.	304
216. Do not require users to use a menu-back functionality to navigate back to the top-level categories in the menu.	304
217. If you use two navigation menus on the same page make sure that there is enough physical separation and visual differentiation between them.	308
218. Do not change the contents of the main-navigation menu to secondary or in-page navigation.	311
219. Users should always be able to go back to the previous page.....	313

220. [App] Make sure that people can easily undo an accidental touch by providing a <i>Back</i> as undo functionality.	313
221. [App, Android, Windows] If the phone has a physical <i>Back</i> button, make sure that the physical <i>Back</i> button functions as undo and allows users to return to the previous page within the app.	314
222. [App, Android, Windows] Do not include a separate <i>Back</i> button if the phone already provides one.	317
223. [Web] Do not include a <i>Back</i> button on a website.	318
224. [Android] On the homepage, the global <i>Back</i> button should take the user back to the previous page and not outside the app.	319
225. [App] No essential functionality (e.g., search) or navigation options should be hidden under the <i>More</i> tab or under the action overflow button.	321
226. [Android] Do not change action-overflow functionality within the app.	325
Search, Filter, Sort	326
227. Most mobile websites and apps with a significant amount of content benefit from having a search tool on the homepage.	326
228. The search functionality should be easy to locate.	327
229. If search is frequently used on your site, make it prominent by using a visible search box placed at the top of the screen.	329
230. If search is only occasionally used on your site, use a magnifier tool for the search box and place that close to the top of the page.	331
231. If the search tool is inside the navigation menu, place it at the very top of that menu.	331
232. If no search box is visible, use a standard magnifier-glass icon for search.	333
233. The length of the search box should be the largest possible size that will fit on the screen.	335
234. Use history-based suggestions for search. Present them to the users when they move input focus to the search field.	335
235. Do not use several search boxes with different functionalities on the same page.	337
236. Do not limit the scope of the search based on the section of the app that the user is in.	339
237. If there are multiple types of scopes available, group results by scope and show a few results from each scope.	339

238. Do not use tabs or menus as the only method to indicate different categories of search results.	342
239. If scope autosuggestions are made, make sure that you allow users to make a selection that does not have a scope applied to it.	344
240. Always show the search query on the search-results page.	345
241. On the search-results page, preserve the search query in the search box.	346
242. If there are no search results, tell people that there were zero results for their search and offer results for alternative searches.	348
243. If there are no search results in your app or mobile website, run the query on your full site and return matching results if any.	349
244. Whenever the results belong to different categories, group them and display a few results from each group.	351
245. Differentiate between search results and search suggestions.	351
246. Provide filters. If a list contains items that belong to different categories, provide filters to narrow down the number of elements.	353
247. If a list of items can be sorted according to different criteria, provide the option to sort that list according to all those criteria.	356
248. Place sort/filter controls at the top of the list.	357
249. Minimize the space allocated to filters above list.	358
250. Indicate the selected filters/sorting criteria, but do not waste a lot of screen space with them.	362
251. Let users quickly navigate to any single filter in your list of filters.	365
252. Consider using menus or accordions for hiding the possible values for a single filtering criterion.	365
253. Make sure that the filter menus/accordions are closed by default.	365
254. Consider allowing users to quickly close a filter, especially if the number of possible values for that filter is large.	368
255. Let users specify all filter values before refining results.	374
256. Let users quickly access the <i>Apply Filters</i> button.	375
257. Allow users to select multiple values for a filter.	377
258. If a filter has many possible values, allow users to type the name of the value to narrow down the list of results.	379
259. When narrowing down results, show facets and number of items in each category.	380

Lists and Scrolling	381
260. Do not use a deck of cards (i.e., one item per page) for lists.	381
261. Use alphabetical ordering for a list if you expect participants to be familiar with the exact names of items in the list.	382
262. In a long list organized alphabetically, allow users to easily jump to a certain letter.	384
263. Use infinite (expandable) lists only if users will never have to go back and inspect previous items in the list.	386
264. For an infinite list, provide users with a back-to-top button or with a sticky header at the top of the list.	387
265. Prefer <i>View All</i> for lists shorter than 100 items; use pagination for longer lists.	387
Content	389
266. Use bullet points, spacing, and headings to draw users' attention to the gist of your content.....	390
267. Prefer brief, concise sentences over convoluted ones.	392
268. Use left-aligned text.	394
269. Stay away from jargon and industry-specific terminology.	395
270. Use an inverted-pyramid writing style. Present overviews and summaries at the beginning of the article.....	396
271. Defer secondary information to secondary pages or collapse it under accordions.	398
272. Do not arbitrarily truncate text paragraphs or product information in lists of products.	400
273. Use links with good information scent.	401
274. Avoid back referencing in articles; that is, do not force users to go back and forth between different parts of the article.....	403
275. If your site contains a lot of text, include tools for changing font size.	405
276. Use a font-size default that is readable for a large majority of users.	405
277. Do not use excessively large font.....	406
278. Place the font-size controls in a place where they are easy to find.	407
279. Use solid backgrounds.	408

280. Use high contrast between the font color and the page background.....	409
281. Use informative headlines.....	410
282. Do not truncate headlines or summaries.	411
283. Frontload headlines and summaries with keywords.....	412
284. Don't repeat summaries from the headlines page on the article page.....	414
285. Do not use the first few sentences of the article as a summary on the headlines page.....	415
286. Use links to related content to help the user navigate more quickly between similar topics.....	417
287. Make sure that the related links are relevant to the article.....	418
288. Place related links in close proximity to the end of the article, to make sure that people do not miss them.	418
289. Do not insert ads between the article and the related content.....	418
290. Unless your content has a sequential structure (e.g., a piece of fiction or a news article), all pages can benefit from an easily accessible table of contents.	423
291. Avoid splitting articles in many pages.....	423
292. Consider including all the images in a slideshow on a single page.	424
293. Include images on your website only if they add meaningful content. Do not use images for decoration purposes only.....	426
294. Avoid stock photography or photography used for decoration only.	428
295. Do not use image sizes that are bigger than the screen. The entire image should be viewable with no scrolling.	429
296. For cases where users are likely to need access to details in a picture, either (1) add a separate link to a higher-resolution, full screen variant, or (2) allow people to zoom into the picture to check details.	429
297. Thumbnails should be as big as need to ensure that users can distinguish what the picture is about.	431
298. Use captions for images that are part of an article only if their meaning is not clear from the content of the article.	433
299. Do not shuffle text around as images get loaded. Use placeholders as images load.	434
300. Do not use images of text for navigation links.	436
301. Do not start videos automatically.	437

302. If you have videos on your site, offer a textual description of what the video is about.	437
303. Indicate video length.	439
304. Consider offering users a transcript of informational videos.	441
305. Consider offering captions to allow users to watch the video in a variety of contexts.	443
Location Information	443
306. If your company has a single brick-and-mortar location, list it in your app or on your mobile website. If the company has multiple locations, include a locator form in your mobile app or on your mobile website.	444
307. In a locator, use two possible values for location: automatically detected current location and location specified by user.	444
308. In a locator form, always give priority to the current location.	444
309. The locator form needs to be highly discoverable.	446
310. Whenever you have information about a single location on your website or in your app, link it to the phone's map application.	448
311. When displaying location information, show address, distance from current location, link to map app, phone, and business hours (if applicable).	450
312. In a list of locations, show how far locations are from the current location. List locations in the order of distance from current location.....	452
313. When displaying multiple locations on a map, start with a zoom level that enables the user to see his current location (or the location specified by the user) and at least one of the locations of interest.	453
314. Allow users to switch between a list view and a map view of locations.	455
315. If locations have different attributes of interest to the user, provide the list of locations by default.	455
316. If locations are all equivalent (except for distance), provide a map view of locations by default.	455
Instructions and Help.....	457
317. Instructional overlays should contain only tips that are truly necessary and not obvious.	457
318. Make the coach mark clearly distinct from the page in the background. Signal to users whether the buttons are active or only depictions of the elements in the interface.	458

319. Meaningful, contextual tips can be helpful.....	461
320. Tips should have clear, simple language.....	461
321. Tips should explain the function not only the action.....	461
322. Tips should point directly to the design element instead of mentioning it by name.....	461
323. Do not flood users with tips.....	465
324. Phase tips out as users learn the interface.....	465
325. Do not use vocal tips.....	465
326. User manuals are not useful, nor should they be necessary on mobile.....	467
327. If you decide that you absolutely need a user manual, make sure that it's easy to read.....	467
328. If you decide to include an initial tutorial, allow users to skip it and use the app right away.....	467
Initial Experience	469
329. Keep launching time to a minimum.....	469
330. If users must wait for more than 20 seconds, give them something to do while waiting.....	469
331. Don't use a launch screen if possible.....	470
332. If you must use a launch screen, make it similar to the first screen of the app.....	470
333. Don't start with a video or a sound effect.....	471
334. Preload data for the first launch.....	472
335. Update data for the users.....	473
336. Don't ask users to delete an app and install it again.....	473

Acknowledgments

We thank several people for helping us gather the data for this research:

- Our participants for delivering all the data on which this report is based.
- Janelle Estes for providing success rates and participant information for one of the studies.
- Hannah Kain, Julia Toy, Aaron Katske, and Brandon Marugg for offering us the space and the assistance necessary to run our user-testing sessions.
- Mihai Budiu for lending us his Windows Phone 7 to plan our mobile testing with Windows Phone 7 users.
- Susan Pernice for helping us with the recruitment of participants in Australia.
- Eric Chow and Luice Hwang for helping us recruit participants in Hong Kong.
- Our colleagues from Nielsen Norman Group, and in particular Amy Schade, Aurora Bedford, Hoa Loranger, Kara Pernice, Katie Sherwin, and Kathryn Whitenton for providing some of the examples and insights.

About the Authors

Raluca Budiu, Ph.D. is a Senior Researcher with Nielsen Norman Group. At NN/g she consults for clients from a variety of industries and presents tutorials on mobile usability, measuring UX, cognitive psychology for designers, and principles of human computer interaction. She coauthored the NN/g reports on tablet usability, mobile usability, iPad usability, and the usability of children's websites. Budiu is also coauthor of the book *Mobile Usability* and she regularly writes articles on UX-related topics for NNgroup.com. Budiu served as a judge for GSMA Global Mobile Awards 2012-2015. Before NN/g, Budiu worked at Xerox PARC, doing research in human-computer interaction and at Microsoft. Budiu has authored more than 20 peer-reviewed articles and conference presentations on human-computer interaction, psychology, and cognitive science.

Jakob Nielsen, Ph.D. is principal of Nielsen Norman Group. He is the founder of the "discount usability engineering" movement, which emphasizes fast and efficient methods for improving the quality of user interfaces. Nielsen, noted as "the world's leading expert on Web usability" by *U.S. News and World Report* and "the next best thing to a true time machine" by *USA Today*, is the author of the best-selling book *Designing Web Usability: The Practice of Simplicity* (2000), which has sold more than a quarter of a million copies in 22 languages. His other books include *Hypertext and Hypermedia* (1990), *Usability Engineering* (1993), *Usability Inspection Methods* (1994), *International User Interfaces* (1996), *Homepage Usability: 50 Websites Deconstructed* (2001), *Prioritizing Web Usability* (2006), *Eyetracking Web Usability* (2010), and *Mobile Usability* (2012). Nielsen's [Alertbox](#) column on Web usability has been published on the Internet since 1995 and currently has about 200,000 readers. From 1994 to 1998, Nielsen was a Sun Microsystems Distinguished Engineer. His previous affiliations include Bell Communications Research, the Technical University of Denmark, and the IBM User Interface Institute.

Evidence-Based User Experience Research, Training, and Consulting

Since 1998 Nielsen Norman Group has been a leading voice in the user experience field.

- Conducting groundbreaking research
- Evaluating interfaces of all shapes and sizes
- Guiding critical design decisions to improve the bottom line

We practice what we preach

We don't just talk about the importance of testing with real users, on real tasks, in real life business situations: we do it. Every week, somewhere around the globe, NN/g team members are conducting research that informs the three pillars of our business: training, consulting and research. In that work we have:

- Tested over 2,000 different interfaces
- Observed more than 4,000 users—in person—in 18 countries and on 5 continents
- Analyzed thousands of hours of recorded user observations sessions
- Conducted countless diary studies, focus groups and remote user tests

Our collective experience will save you time... and money

Making technology easier to use is no longer a nice-to-have. Useful, usable products make money. And our expertise can help your team achieve their design goals quicker and easier than going it alone. Choosing NN/g means you benefit directly from our:

- Finely tuned methodology: We have an arsenal of proven tools at our disposal and know how and where to apply each one, taking the guesswork out of how to achieve the optimal design solution to meet your business goals.
- Comprehensive body of knowledge: We've taken the results of our decades of research and testing and distilled it down into actionable guidelines, best practices and proven methodologies. Our research library, containing over 50 published reports, X books and an email newsletter archive dating back to 1995 is unrivaled.
- Practical approach: Our approach is 100% practical, useful and actionable. Whether you attend one of our Usability Week events or invite us to consult at your place of business, the training you will receive can be put into action immediately so that you can see the results.

Our people are the best in the business

At NN/g there is no “B Team”. When we dispatch consultants to work with you and your team, or when you attend a Usability Week course, you are learning directly from some of the best-educated and most experienced minds in the business.

- Our principals are considered pioneers in the fields of user research and interface design.
- Our researchers and consultants tackle the most recent and relevant topics in usability, from evergreen challenges such as information architecture and intranet usability to emerging trends in social media and mobile usability.

Stay Informed**Jakob Nielsen's Alertbox Newsletter**

Summaries of our latest research and insights published twice per month.

To subscribe: <http://www.nngroup.com/articles/subscribe>

TRAINING

Usability Week Events

Usability Week training events are offered in the U.S., Canada, the U.K., Europe, Asia and Australia.

Each week-long event features full-day, immersive training courses where attendees learn practical skills directly from experienced practitioners so they can solve complex UI problems and create better interface designs.



Over 40 courses offered in these categories:

- Agile
- Applications
- Content Strategy
- Credibility & Persuasion
- Email
- Information Architecture
- Interaction Design
- Intranets
- Mobile & Tablet
- Non-Profit Websites
- Prototyping
- Social UX
- User Testing
- Visual Design
- Web Usability
- Writing for the Web

Available courses and upcoming locations: www.nngroup.com/training

In-house Training

Many of our courses can be taught at your location and customized to fit your unique offerings, methods and resources.

In-house training is ideal for:

- Large teams that want to spread user experience perspective throughout the group
- Teams working on large projects that need to kick start the creative process and head in the right direction

In-house training information: www.nngroup.com/consulting

REPORTS

NN/g has published over 60 reports that detail thousands of evidence-based design guidelines derived from our independent research studies of websites, intranets, application, and mobile interfaces.



Over 60 reports addressing these topics:

- Agile
- Applications
- Audience Types (e.g., children, college students, seniors, people with disabilities)
- B2B Websites
- Corporate Websites
- Ecommerce
- Email
- Information Architecture
- Intranets
- Mobile & Tablet
- Non-Profit Websites
- User Testing
- Social UX
- Strategy
- Web Usability

Shop for reports here: www.nngroup.com/reports

CONSULTING

The same experts who conduct our research and teach Usability Week training courses are available for custom consulting including:

- **Evaluating your website, application, intranet or mobile interface** (average cost \$38,000 USD)
- **Usability testing** (average cost \$25,000 - \$35,000 USD)
- **On-site training with your team** (average cost \$10,000 USD, plus travel per day)

Consulting details: www.nngroup.com/consulting