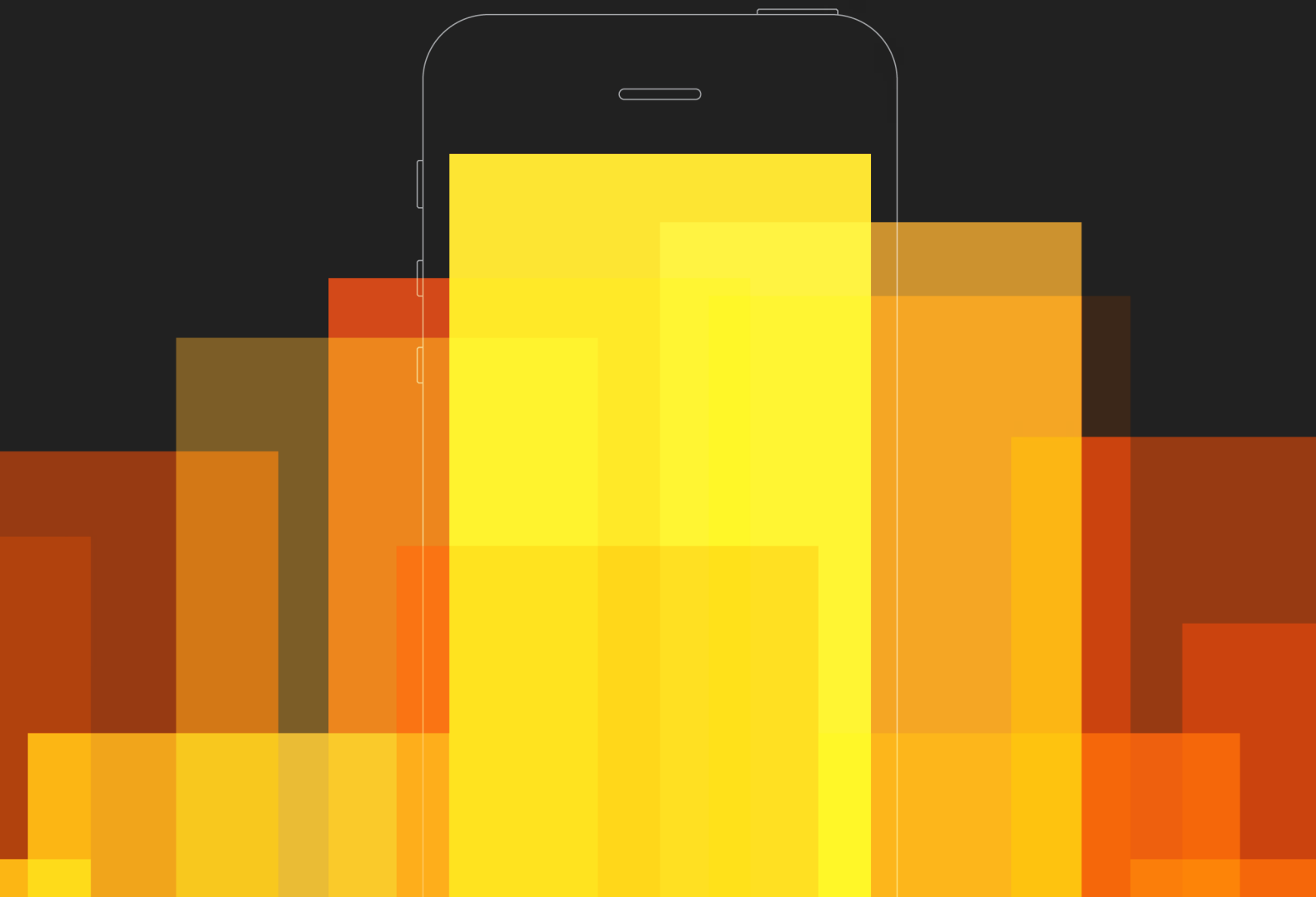
UXPin

# Mobile UI Design Patterns 2014

*A Deeper Look At The Hottest Apps Today*



UXPin

# Mobile UI Design Patterns 2014

*A Deeper Look At The Hottest Apps Today*

If you like Uber, Pinterest, Tinder, OKCupid, Spotify, Yelp, Facebook, Instagram, Dropbox, Dropbox Carousel, Facebook Messenger, Secret, Quora, LinkedIn, RelateIQ, Flipboard, Snapchat, or Mailbox...

You'll love what you see next.

designed by
Dominik Pacholczyk

# INDEX


1. INTRODUCTION
    - What Are UI Design Patterns?
    - How Should I Use UI Design Patterns?
2. THE INTERACTIVE IMPERATIVE
    - Gestures
    - Animations
3. GETTING INPUT
    - Smart Keyboards
    - Default Values & Autocomplete
    - Immediate Immersion (or “Lazy Signups”)
    - Action Bars
    - Social Login
    - Huge Buttons
    - Swiping for Actions
    - Notifications
    - Discoverable Controls
    - Expandable Inputs
    - Undo
4. NAVIGATION
    - Walkthroughs & Coach Marks
    - Overflow Menus
    - Sliders
    - Content-Based Navigation
    - Morphing Controls
    - “Sticky” Fixed Navigation
    - Vertical Navigation

Popovers
Slideouts, Sidebars & Drawers
Links to Everything
Advanced Scrollbars
Swipe Views

4. SOCIAL
Activity Feeds
Friend Lists
Follow
Vote to Promote
Direct Messaging
Single Share Button
Like
Find & Invite Friends

5. DATA & CONTENT MANAGEMENT
Full-Screen Modes
Interactive Content Layers
Inline Expanding Areas
Circles
Transparency
Maps As Backgrounds
Group Friends & Content
Full-Bleed Images
Grids
Cards
Hidden Information
Empty States
Direct Manipulation of Content & Data
Draggable Objects
Pull to Refresh

6. LEARN FROM THE BEST

7. ABOUT UXPIN

Share this ebook
with friends!

# 1

# Introduction

*A quick note from the author*

*"Creativity involves breaking out of established patterns in order to look at things in a different way."*

Edward de Bono

UI design patterns are frequently misinterpreted and misused concepts. I've learned a lot about them in writing this e-book and hope you do too.

Too often, they're confused with specific elements (or features) that can simply be developed as-is from one of the popular Design Pattern and Wireframing Libraries. While those examples are a great start, they're not enough. And although UI design patterns overlap with development patterns - which you can learn more about in A Beginner's Guide to Design Patterns - they're focused more on solving common user problems and less on product problems.

The UXPin team recently scoured the interwebs for mobile UI design patterns and elements being used by the most forward-thinking mobile application companies and adopted eagerly by many others as the product design and development process becomes even more lean - and competitive. We've seen a beautiful assortment of mobile applications outfitted with touch, press, swipe and drag, and pinch-enabled visual and functional solutions to the user's everyday problems.
Below, I've shared an abundance of these standard reference points with you to help you as you brainstorm, sketch, wireframe, mockup, and prototype.
To name a few, we've summarized patterns and elements of **Uber, Pinterest, Tinder, OKCupid, Spotify, Yelp, Facebook, Instagram, Dropbox, Dropbox Carousel, Facebook Messenger, Secret, Quora, LinkedIn, RelateIQ, Flipboard, Snapchat, Mailbox and many, many more**. We've even included matching wireframe examples to help you visualize these patterns and elements at different stages in the design process - you can use them directly in our wireframing and prototyping tool, UXPin. Some of the UI design patterns and elements are also covered in UXPin's free Web Design Trends, Mobile Design Trends, and The Guide to Wireframing e-books - bookmark them for another time since there's a lot to catch up on.

We'd love your thoughts on what I've written. And feel free to include anyone else in this discussion by sharing this e-book.

For the love of mobile,
Chris Bank
*(co-written by Waleed Zuberi)*

## What Are UI Design Patterns?

Generally speaking, a UI design pattern is a reusable solution to a commonly occurring problem you might encounter every day.

It is not a feature that can be plugged into your product design and it's not a finished design that can simply be coded. Rather, it is a formalized best practice, a guide or template, that designers, developers, and product managers (and anyone else who touches product) can use to solve common problems when designing a mobile application or system. Although it has to be utilized in the correct situation, it is generally language, device, and platform agnostic - although there may be technological limitations depending on how the designs are ultimately implemented.

And, of course, if implemented in the wrong context, they do more harm than good - but we'll focus on the positives for now.

## How Should I Use UI Design Patterns?

Despite the glut of Design Pattern and Wireframing Libraries out there, few online resources actually break down how to think about these patterns with supporting examples. Below, I've broken out the general format and provided a brief example to get you started.

Here's what you need to think about when evaluating a design pattern and adapting it to your own needs:

- **Problem Summary:** What user problem are you solving? Stay focused, and phrase it like a user story - in one sentence only.

- **Solution:** How have others solved this problem? Among others, few things to detail include user navigation (including shortcuts), getting user inputs, dealing with data and integrations with other services or applications, and displaying

information and content (including defaults).

- **Example:** Great, can you show me? Sometimes a screenshot or mockup is sufficient; other times, a user flows and/or additional notes are necessary to clearly communicate the pattern.

- **Usage:** When should this pattern (not) be used? Among others, a few things to detail include product architecture, interface layout, device(s), programming language, absence or existence of other design patterns, type of user, and primary use cases.

For the sake of brevity, we'll highlight the first three details of the design patterns in this e-book below. In our next e-book, however, we intend to dive deeper into the solution details and use cases so you have a deeper understanding of when and how to use some of these design patterns.

2

# The Interactive Imperative

*At the heart of many of these new UI design patterns is the evolution of gestures and animations in mobile development.*

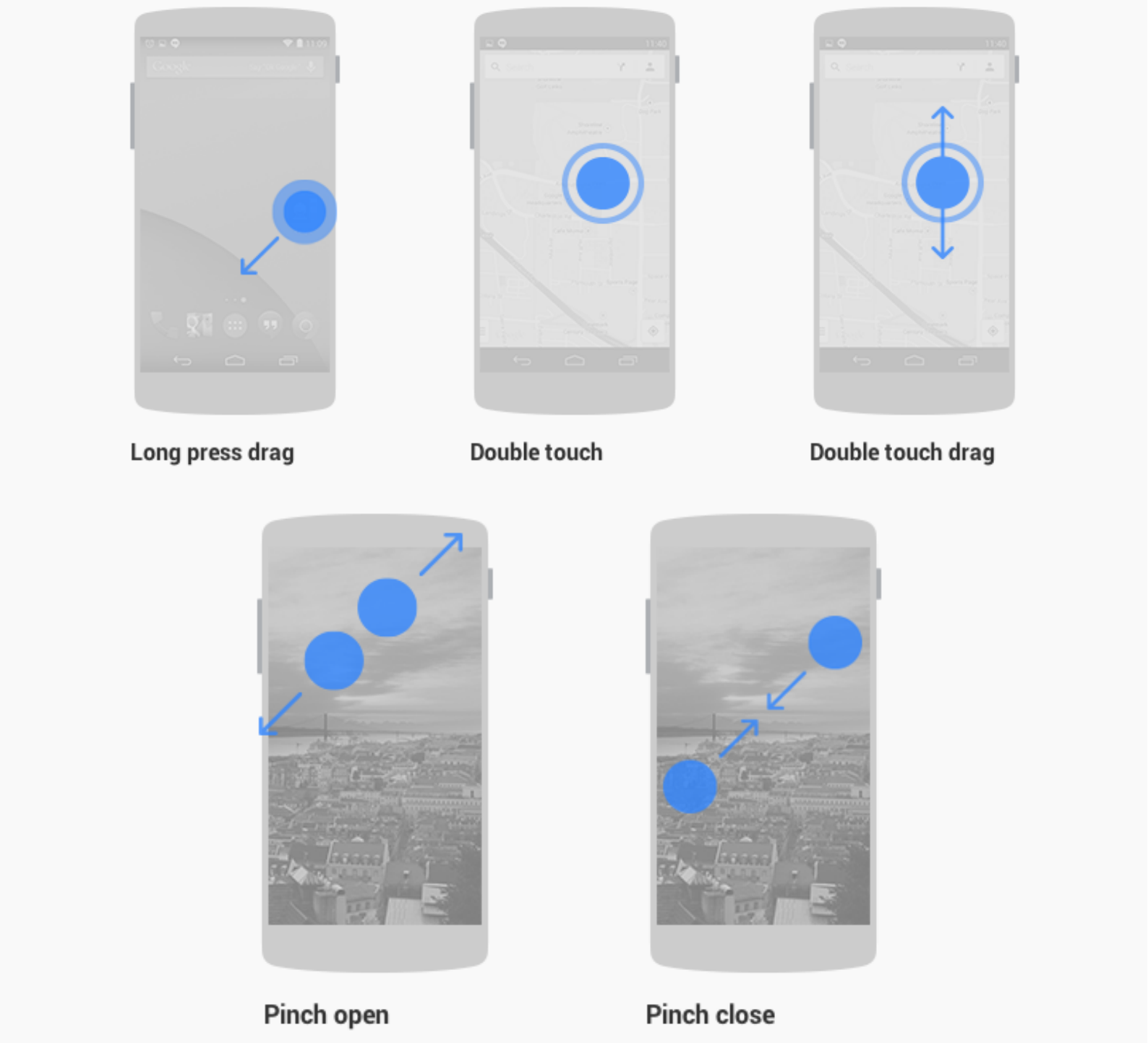
# Gestures

Remember the days when clicking, hovering, scrolling, and keyboard shortcuts were the only triggers for web interactions? Although they're not the only triggers today, they certainly dominate the user experience across the web.

Mobile application design, on the other hand, has exploded with new design patterns and their implementations. Made possible by advancing hardware and software capabilities, the mobile space is developing with unprecedented levels of human-computer interactions. These solutions are largely empowered by new gestures. And, marked by responsive design advancements, web and mobile design is rapidly converging so applications may be build for all device shapes and sizes - this will have a dramatic and re-invigorating impact on design of the web experience.

If we look at Android's current gestures, you get a sense of how many ways someone could interact with a pattern as simple as a calendar, message or picture gallery.

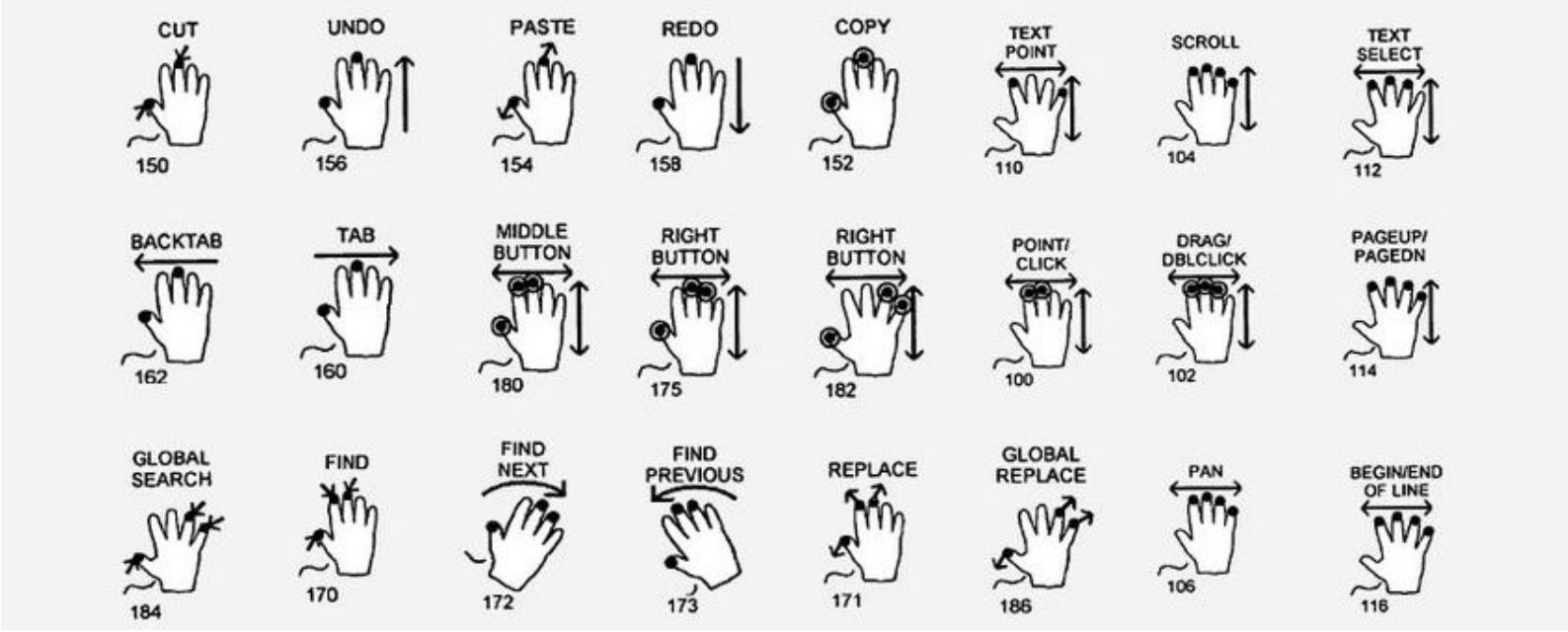
source: *developer.android.com*




source: *developer.android.com*

If we glimpse into Apple's treasure trove of multi-touch gesture patents, your mind might explode with the possible solutions you could design. While the gesture examples below are from a 2010 patent filing, many have yet to be implemented and Apple's archive of gesture patents extends far beyond these examples. Consider it a glimpse into the "Minority Report" future.

Android, iOS, and other mobile platforms will continue to provide an increasing




source: *patentlyapple.com*

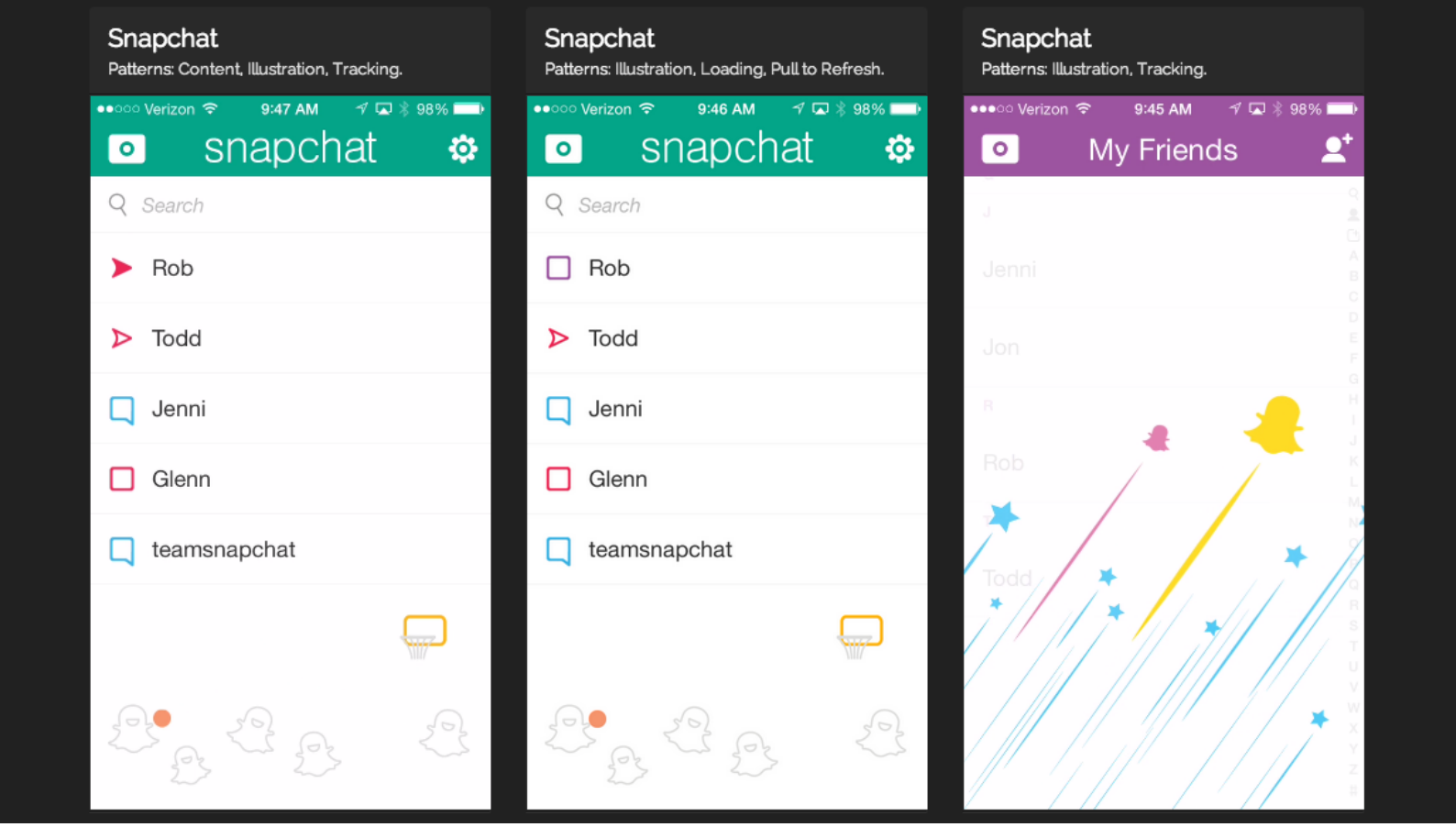
variety of gestures natively that can be implemented for taking particular actions or navigating through the app. Not only does this help preserve screen real-estate by eliminating some of the on-screen buttons, but it also makes the experience intuitive and fun. Combine this with various animations, and you can have a field day with the ways you can implement UI design patterns in your mobile applications.

## Animations

The best thing about using gestures on mobile is that they feel so intuitive and responsive in the same way a real object would. Animations play a very important role in maintaining that illusion for users and keeping them grounded in the UI.

Beyond the tangibility of animations, they're often used to delight users by manipulating the laws and lens of nature. Timing, velocity, bounciness, image and color transitions, scrolling, and myriad other settings allow designers to tweak the user experience in seemingly limitless ways that can be both surprising and amazing.

The sheer variety of animations coupled with gestures not only adds an element of fun to the experience, but are invaluable in providing visual feedback and affordance to the user. We touch on it briefly here but will go in-depth later on our blog and design library.




Captivate is a great site to see animations from popular mobile apps.

| | | | |
|---|---|---|---|
| **In:** rotateSlideIn **Out:** rotateSlideOut | **In:** rotateSidesIn delay200 **Out:** rotateSidesOut | **In:** rotateCarouselBottomIn **Out:** rotateCarouselBottomOut ontop | **In:** rotateCarouselTopIn **Out:** rotateCarouselTopOut ontop |
| **In:** rotateCarouselRightIn **Out:** rotateCarouselRightOut ontop | **In:** rotateCarouselLeftIn **Out:** rotateCarouselLeftOut ontop | **In:** rotateCubeBottomIn **Out:** rotateCubeBottomOut ontop | **In:** rotateCubeTopIn **Out:** rotateCubeTopOut ontop |
| **In:** rotateCubeRightIn **Out:** rotateCubeRightOut ontop | **In:** rotateCubeLeftIn **Out:** rotateCubeLeftOut ontop | **In:** rotateRoomBottomIn **Out:** rotateRoomBottomOut ontop | **In:** rotateRoomTopIn **Out:** rotateRoomTopOut ontop |

If you want to see a ton of transitions without the frills, Dan Silver has some inspiration.

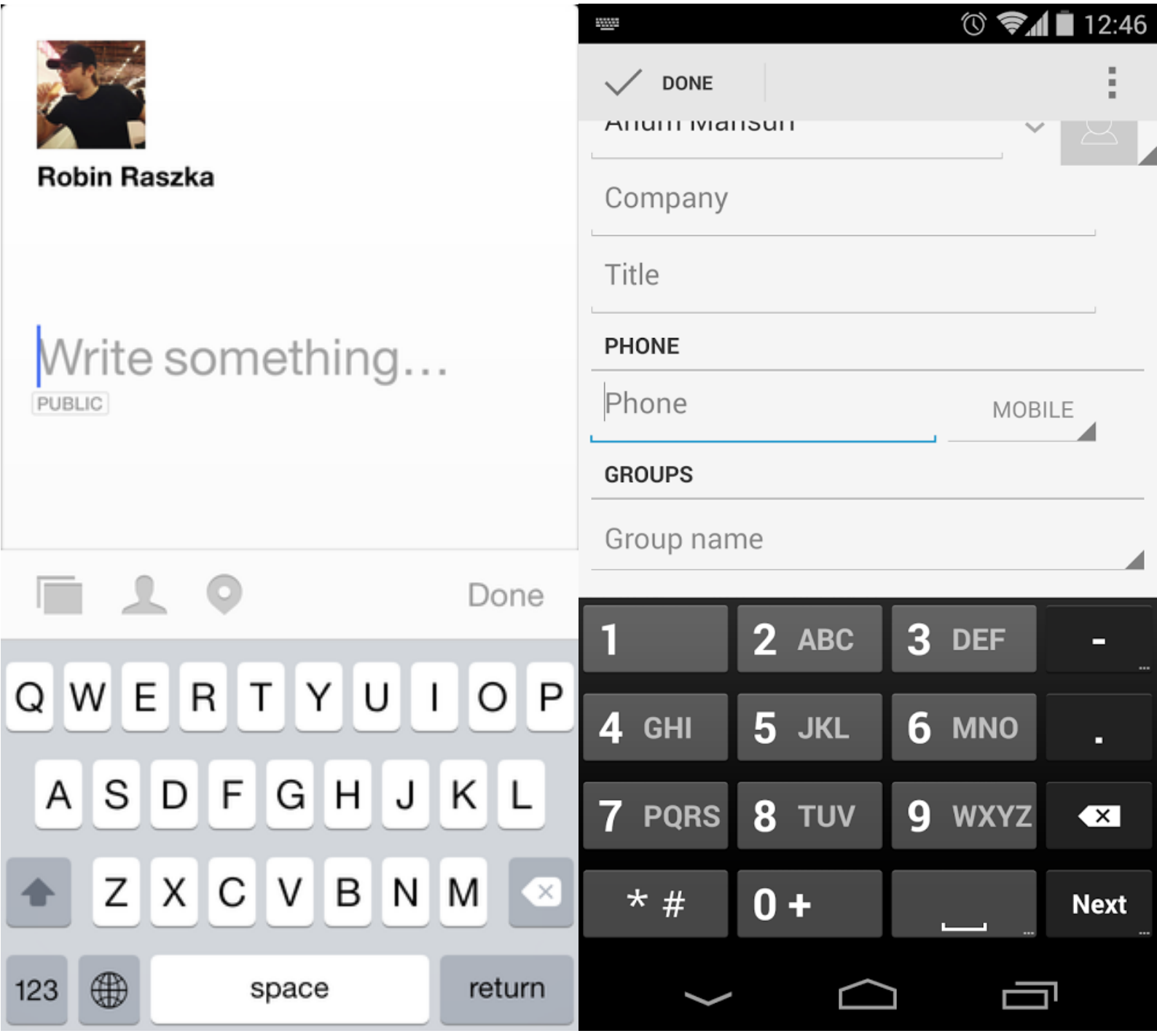
# 3

# Getting Input

# Smart Keyboards

Examples

Facebook Paper, Android Contacts




Problem

The user wants to enter information quickly.

## Solution

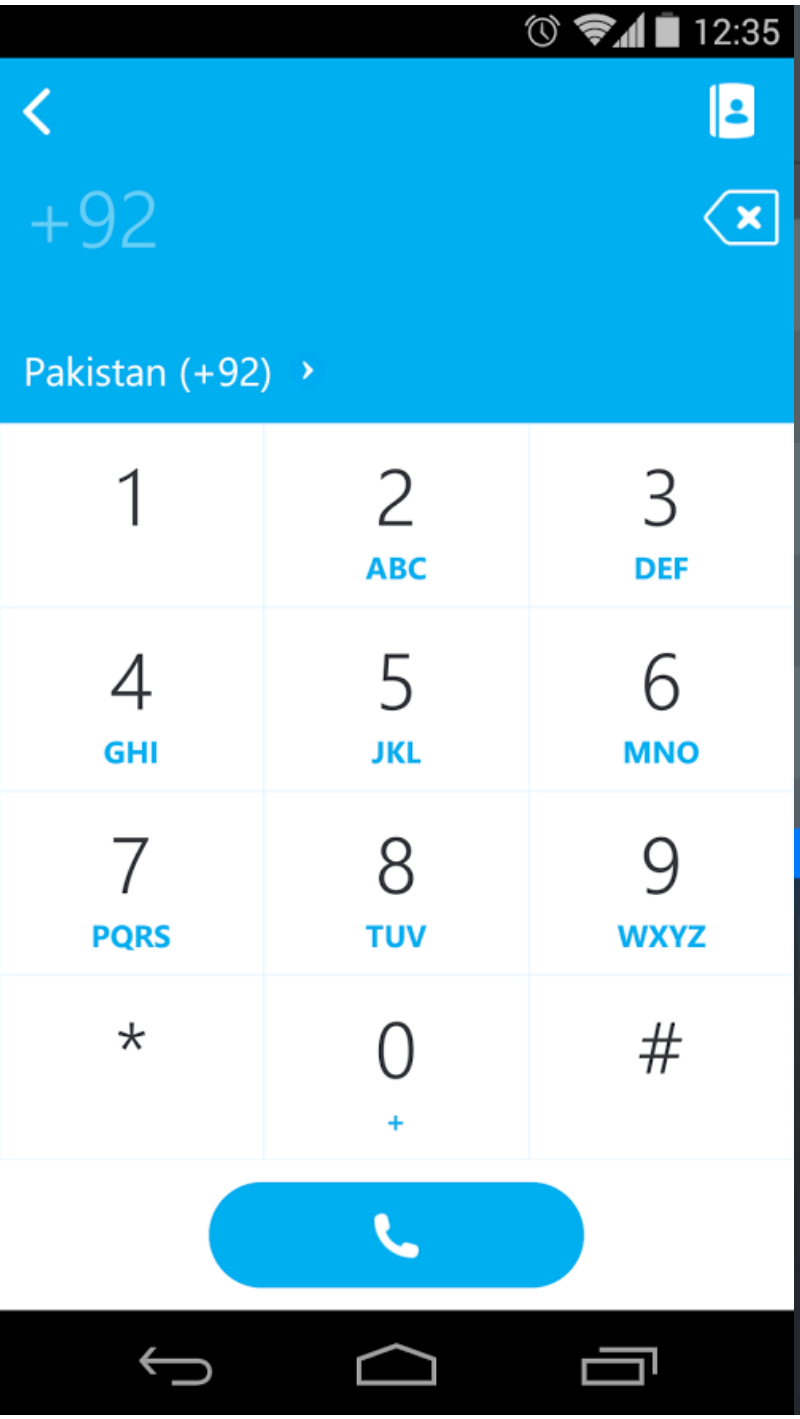
Give users the keyboard that's relevant to the data they are entering when they tap into a section of the app that allows for entering information. This saves them from having to move between the alphanumeric screens to find the right buttons, or taking an extra step to access the keyboard. Not only is this convenient for the user, it also serves as an indication of what kind of input is expected from them. Mobile platforms allow text fields to be marked accordingly and this allows for some flexibility in terms of which buttons are displayed more prominently.

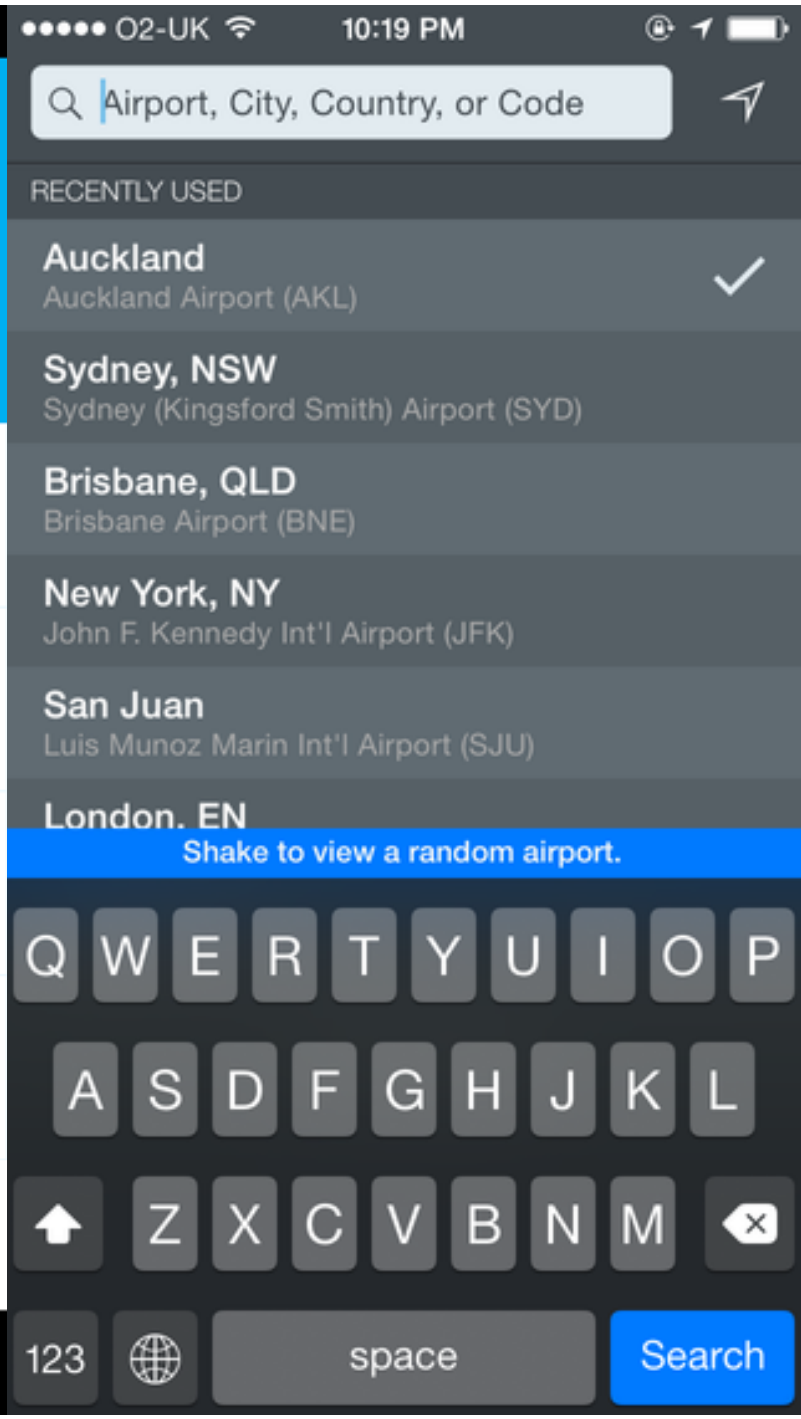
For example, when entering phone numbers in address books or dialers, the user doesn't need the full keyboard. When they tap on these fields, the numeric keypad pops up instead of the full keyboard, making the process more streamlined by doing away with the distraction of unnecessary buttons. Similarly, tapping on a URL bar in the browser brings up a slightly modified keyboard in which the "/" and ".com" buttons are displayed next to the spacebar instead of behind the symbols key. By hooking into these smart keyboard types provided by the system, your UI can adapt according to what the user is currently trying to do.

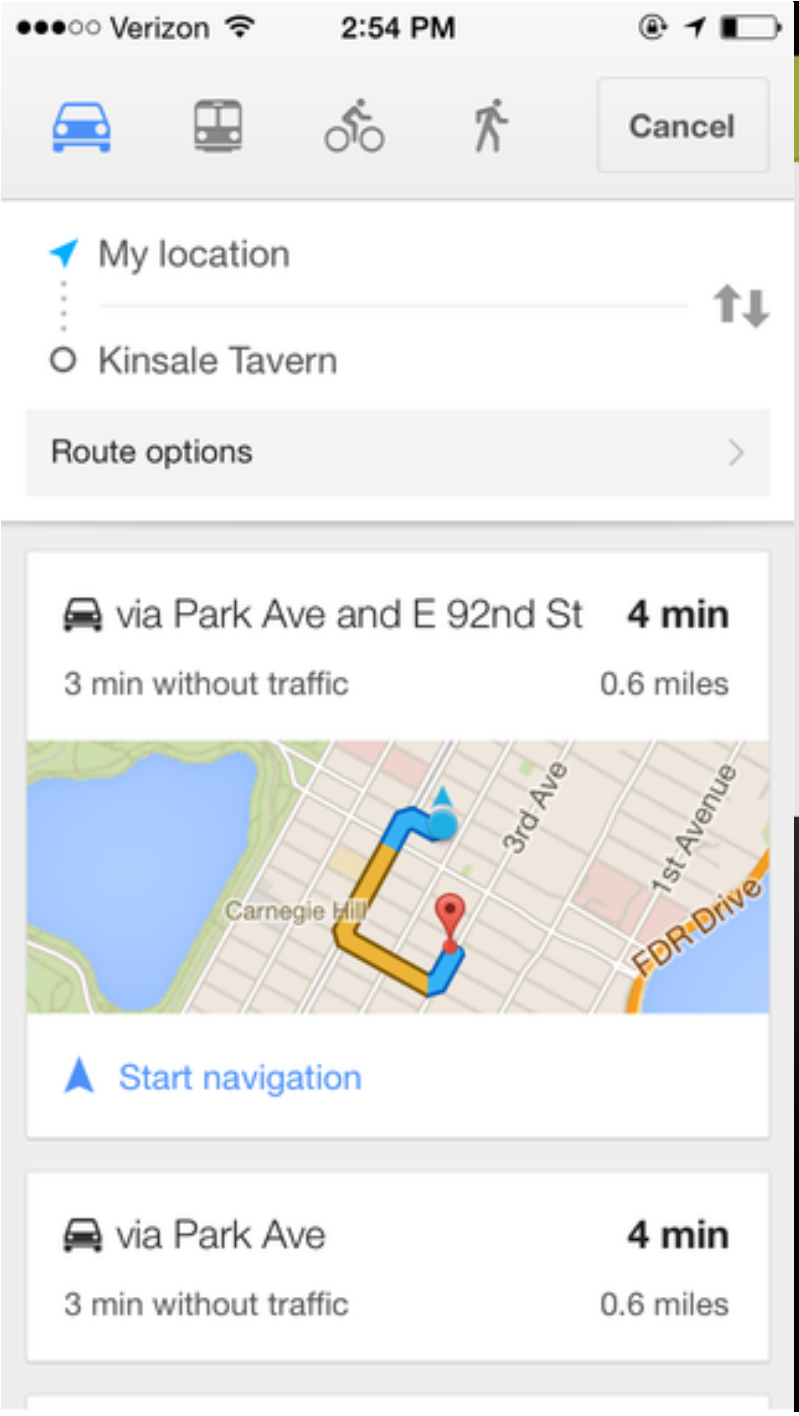
# Default Values & Autocomplete

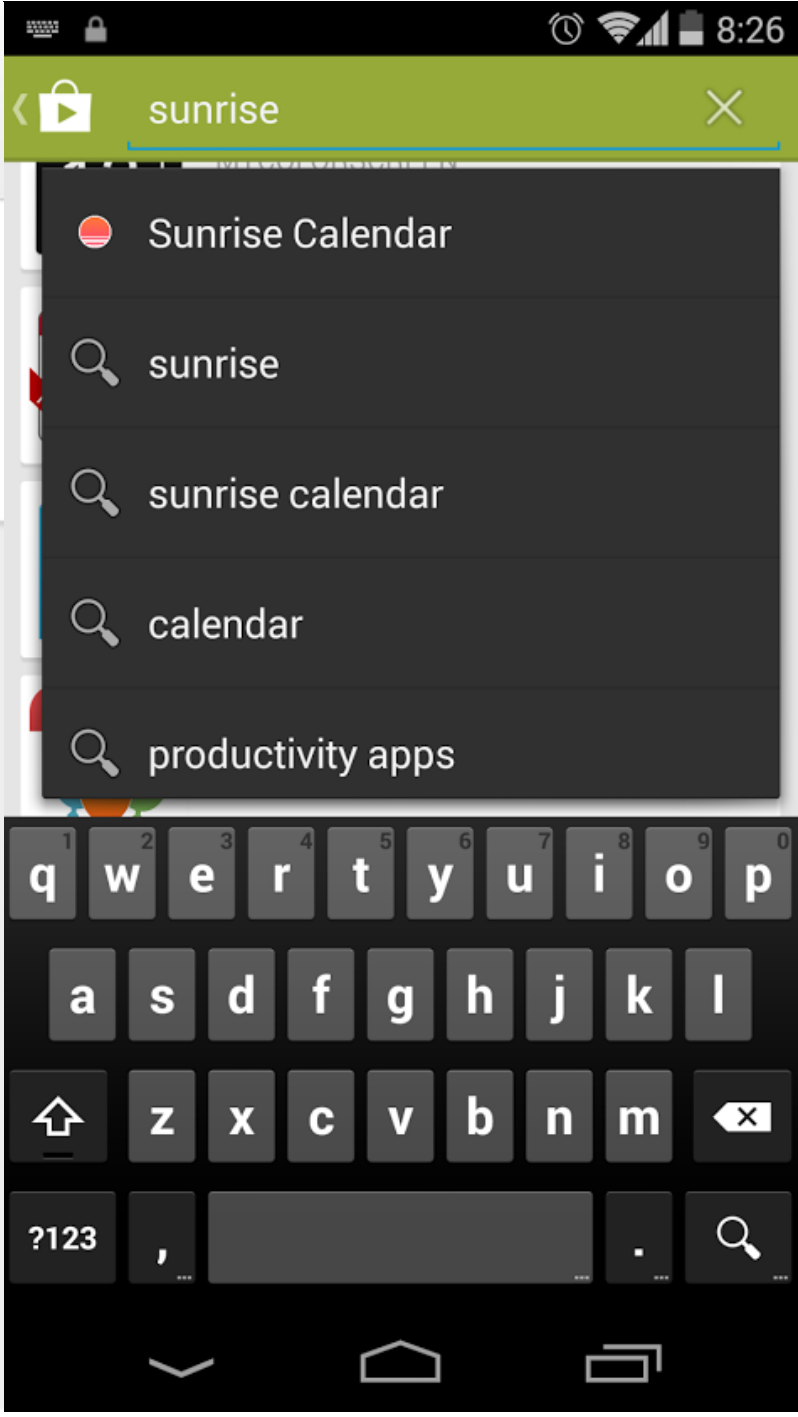
Examples

Skype, Flightboard, Google Maps, Google Play Store






## Problem

The user wants to complete actions quickly.

## Solution

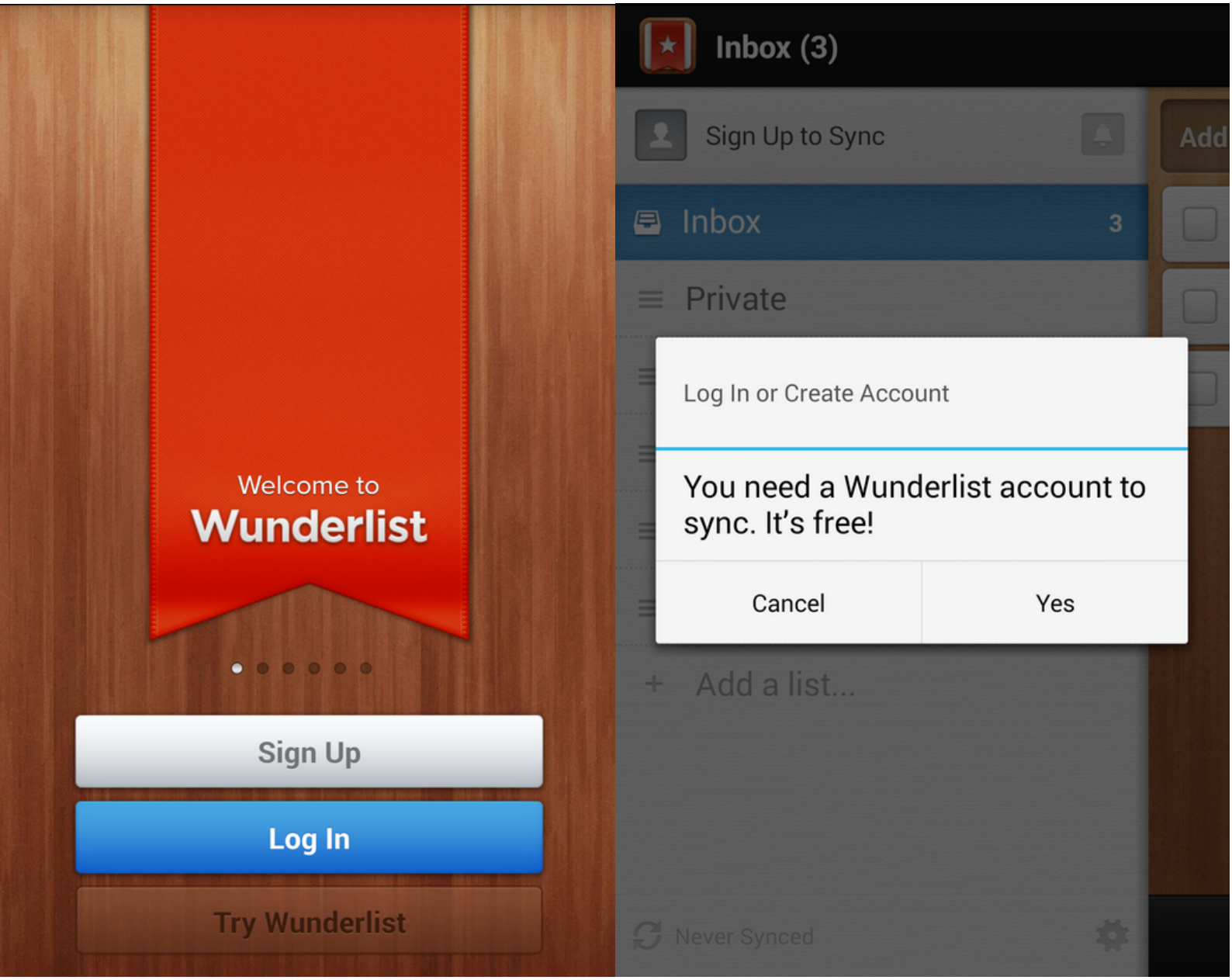
Anticipate frequently selected items and make data entry easier for the user by providing them with pre-populated default values or prompts based on previously entered data. This can be paired with autocomplete functionality like in the Google Play Store search, significantly improving the user experience by speeding things up. This pattern can be particularly useful in standardizing user input and anticipating problems before they occur. Skype, for example, automatically prepends

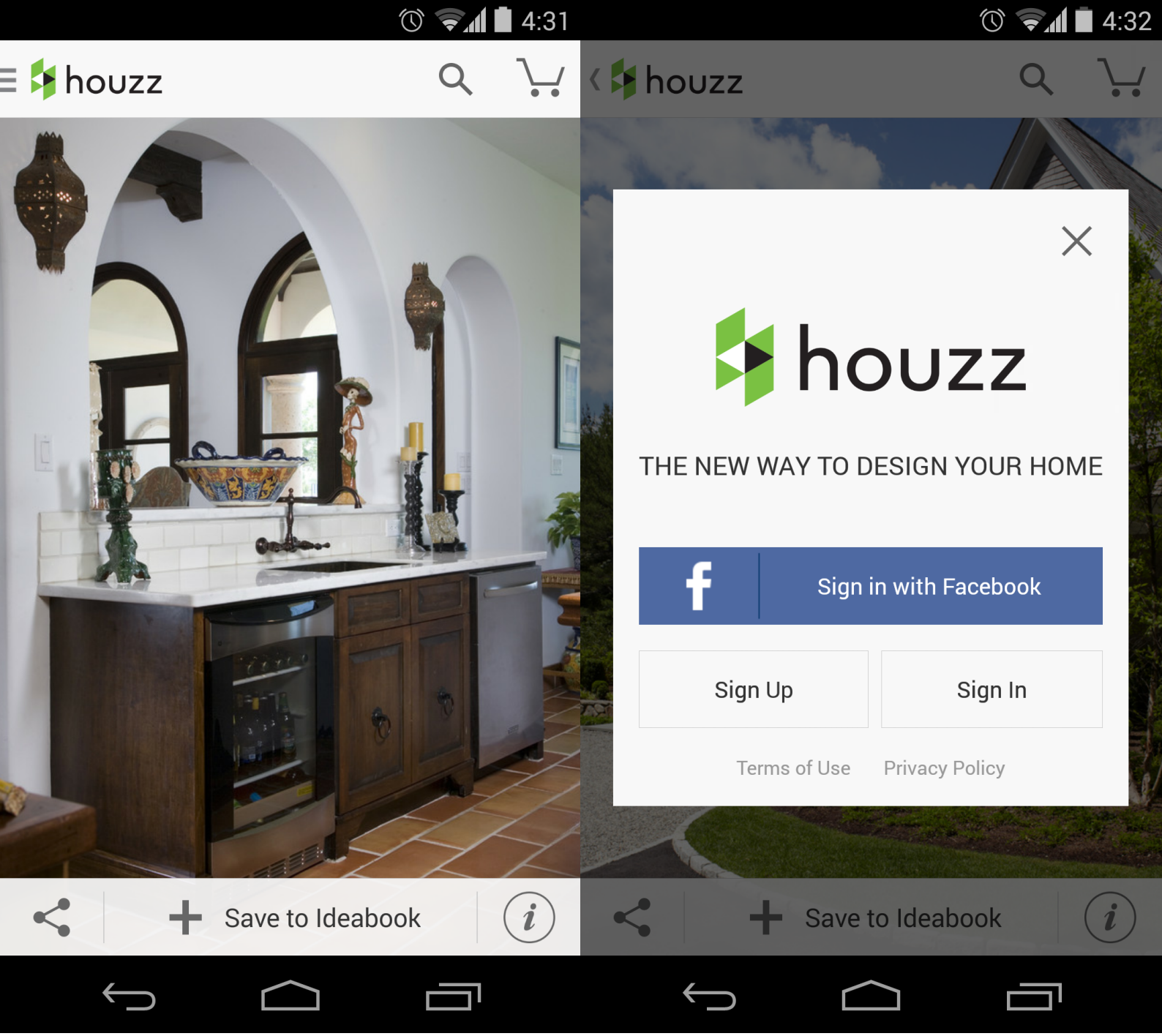
entered phone numbers with the country code. This makes sense from the user's perspective because they aren't used to entering this information on a regular basis, but it's important in this context because Skype is an international calling app. Another way of implementing this is by saving the last item entered by the user and presenting these recently used items when the user goes to enter or search again. For example, Flightboard lists previously used locations below the search box to save users from having to type it in again. Most map or directions apps also implement this pattern, saving the user a few taps by automatically entering the user's current location when searching for directions because that is simply the most common occurrence.

# Immediate Immersion (or “Lazy Signups”)

Examples

Wunderlist, Houzz






## Problem

The user wants to try things out before making a commitment.

## Solution

More applications are letting users immediately immerse themselves in an app before anything else - even signing up or logging in.

Remember, they can only do one thing at a time, and have limited time to test every new product out. With the growing specialization of apps, it's increasingly important that you find quality user or customer leads before nurturing them - they may hate

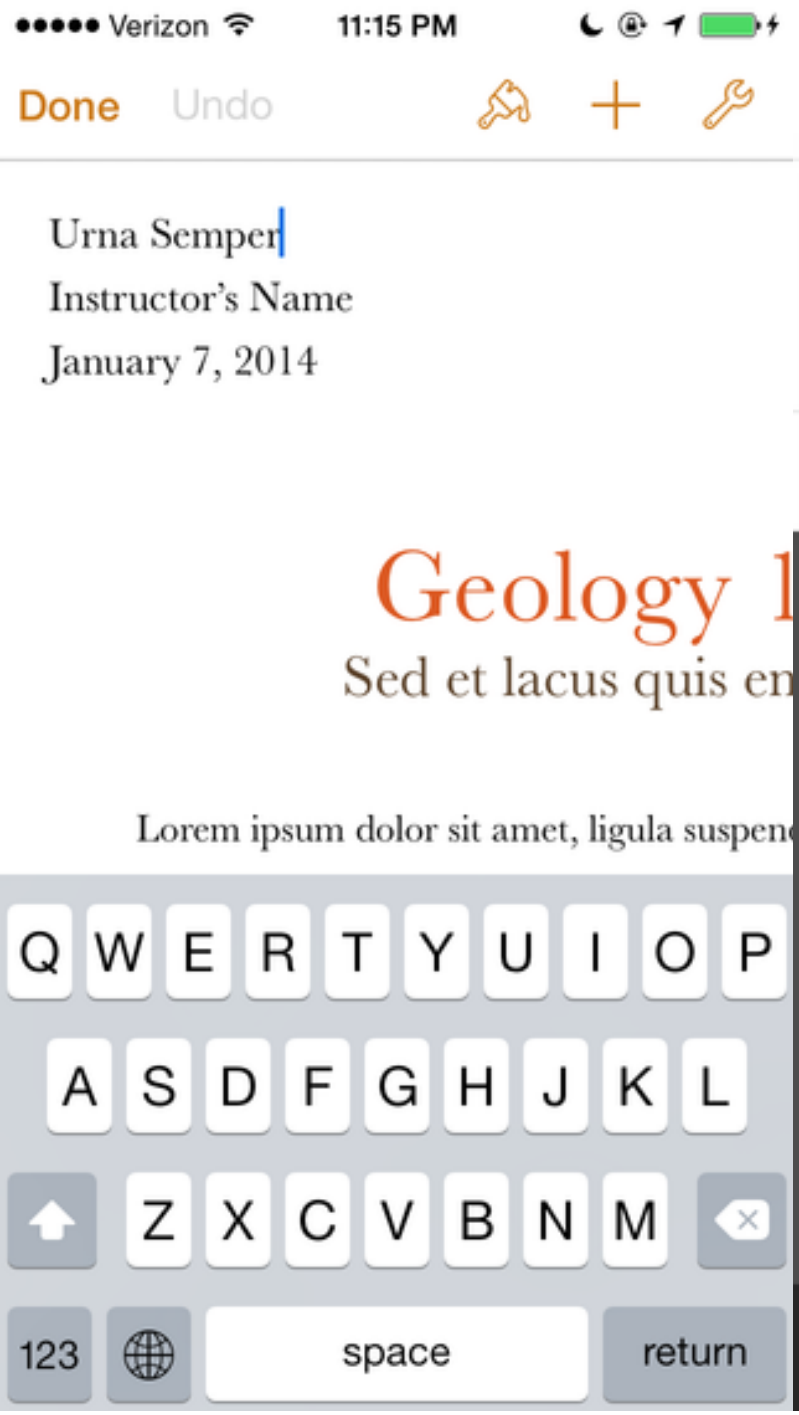
your product or quickly realize it's not what they wanted. Asking users for the information you need to register their accounts can be a tough ask, and lower signup conversion rates even for qualified visitors. On a positive note, by letting them immediately experience your product, they may get more hooked because of how deeply they were able to explore the app on the first experience. This can work better than the onboarding walkthrough UI pattern we cover next, because it shows the user instead of telling them how things work.

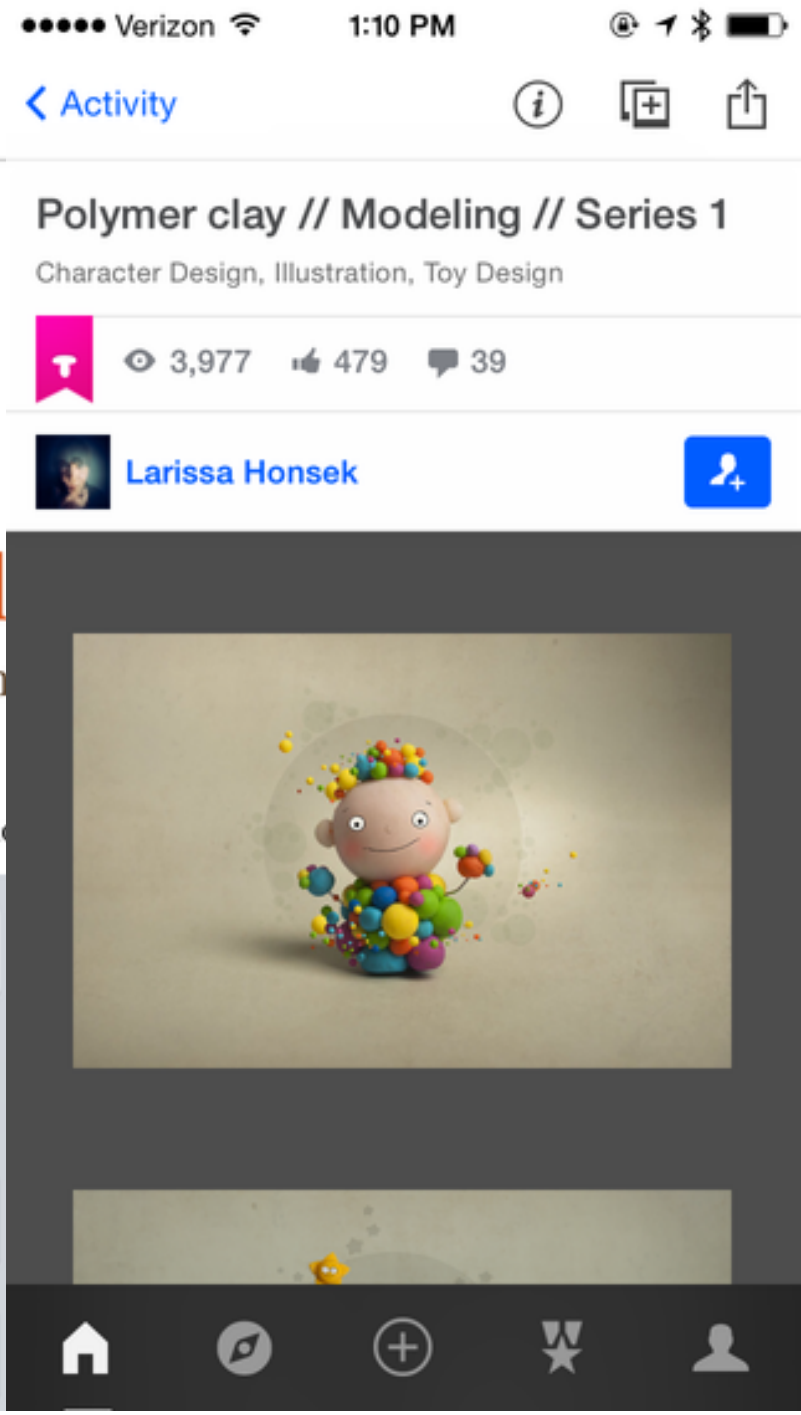
Allowing late registrations doesn't make sense for apps like Carousel or Duolingo, which rely on user data to function, but apps like Wunderlist or Houzz can allow their users to come in and use the app before asking them to identify themselves. Oftentimes, registration comes with an added benefit which makes it more attractive, like cross-device syncing in Wunderlist or creating an idea book in Houzz. Late registrations may not always be a good idea, but the option to "try-before-you-register" can a great way to increase engagement with your app.

# Action Bars

Examples

Facebook Paper, Behance



## Problem

The user wants quick access to frequently used actions.

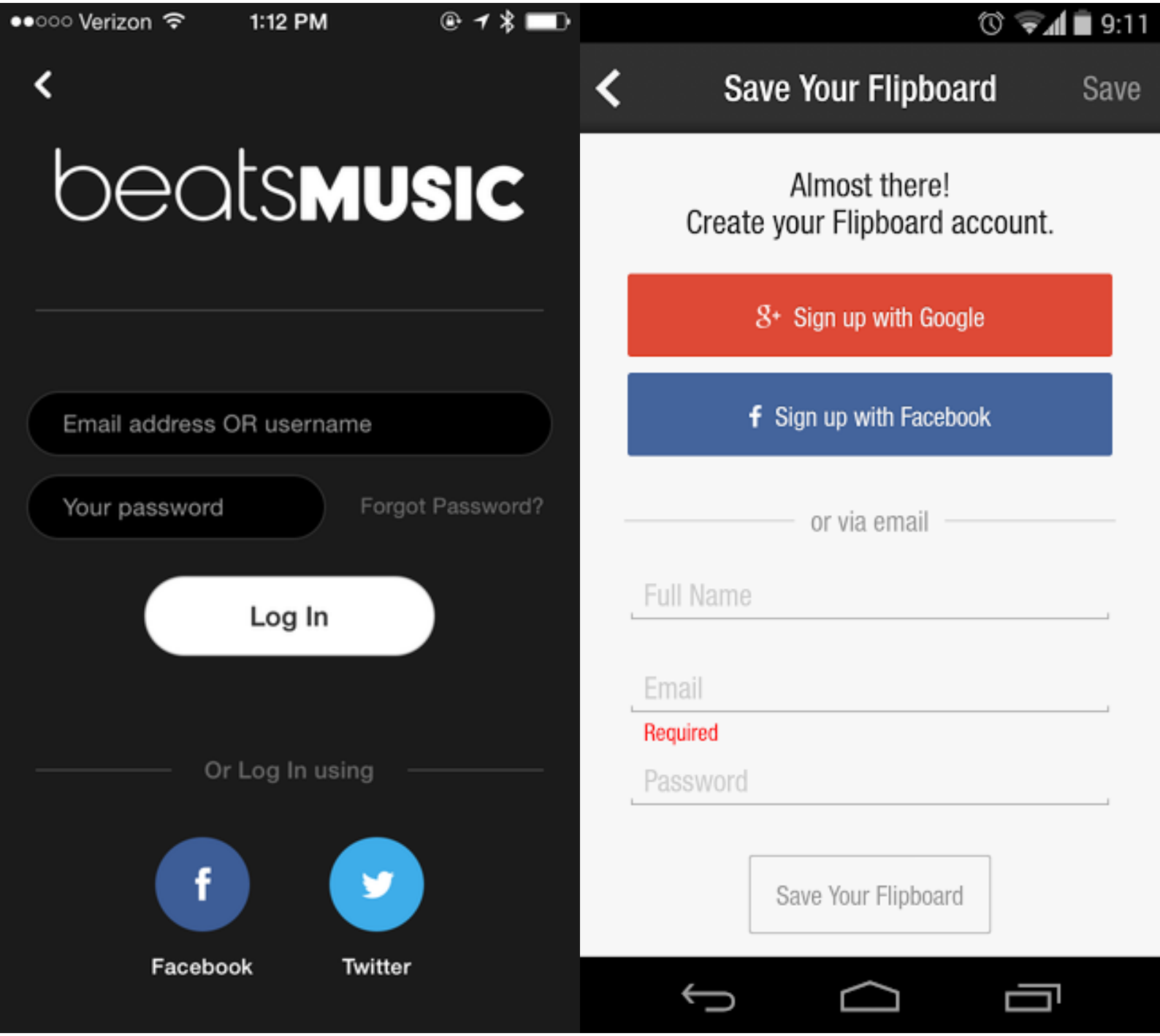
## Solution

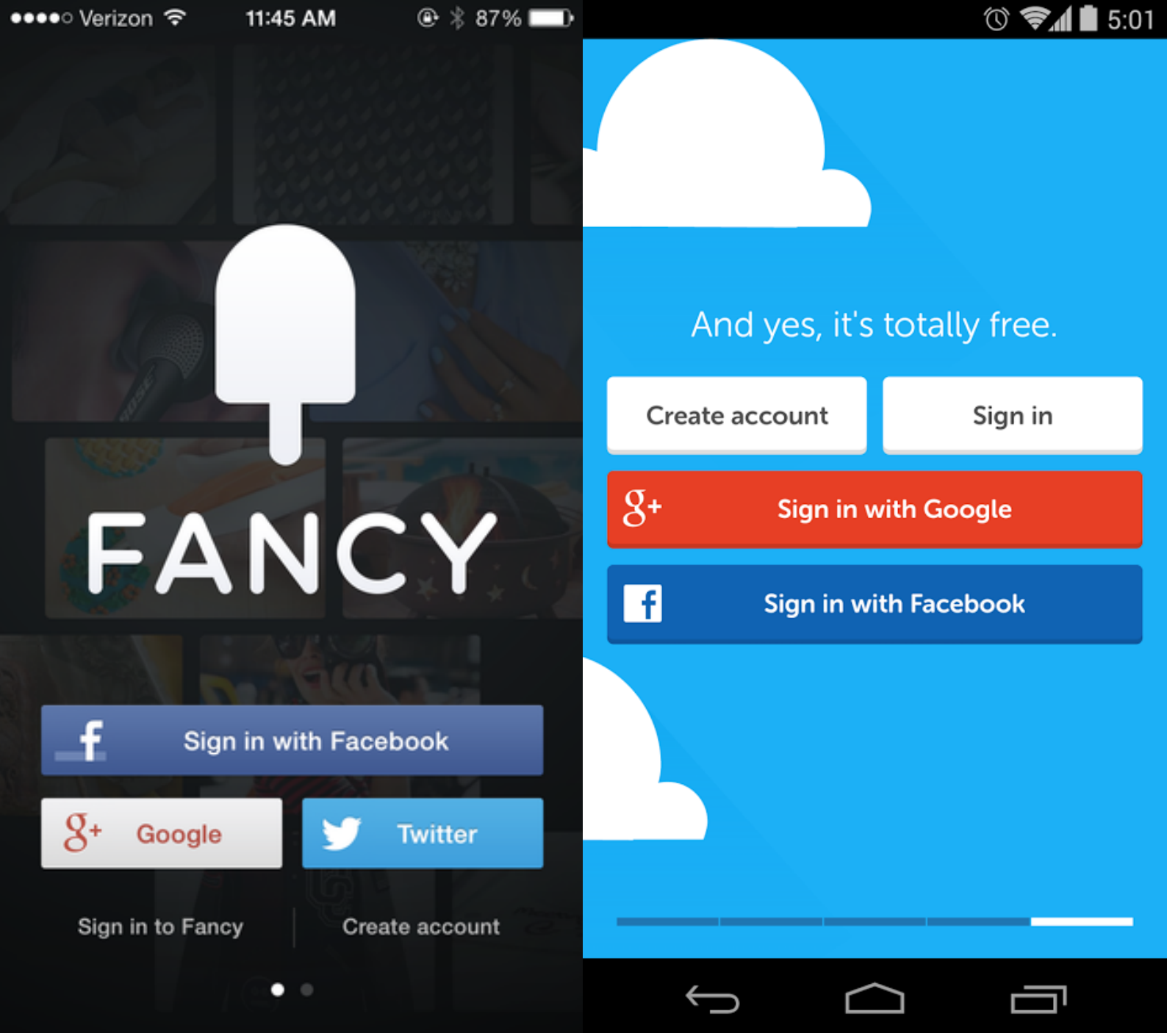
Provide quick access to important actions from the app's action bar (or "toolbar" in iOS terminology). While navigation bars have dominated web and early mobile application design, the use of other patterns like drawers, slideouts & sidebars, links to everything, button transformations, vertical and content-based navigation have allowed for more simple app views that can focus on primary and secondary actions, and less on secondary navigation. Common actions are search, share and creating new content within the app. This persistent menu helps users become familiar with the UI but also clears away some clutter by focusing on the important actions that are relevant to the user.

# Social Login

Examples

Beats Music, Flipboard, Fancy, Duolingo






## Problem

The user wants an easier way of signing up and logging in.

## Solution

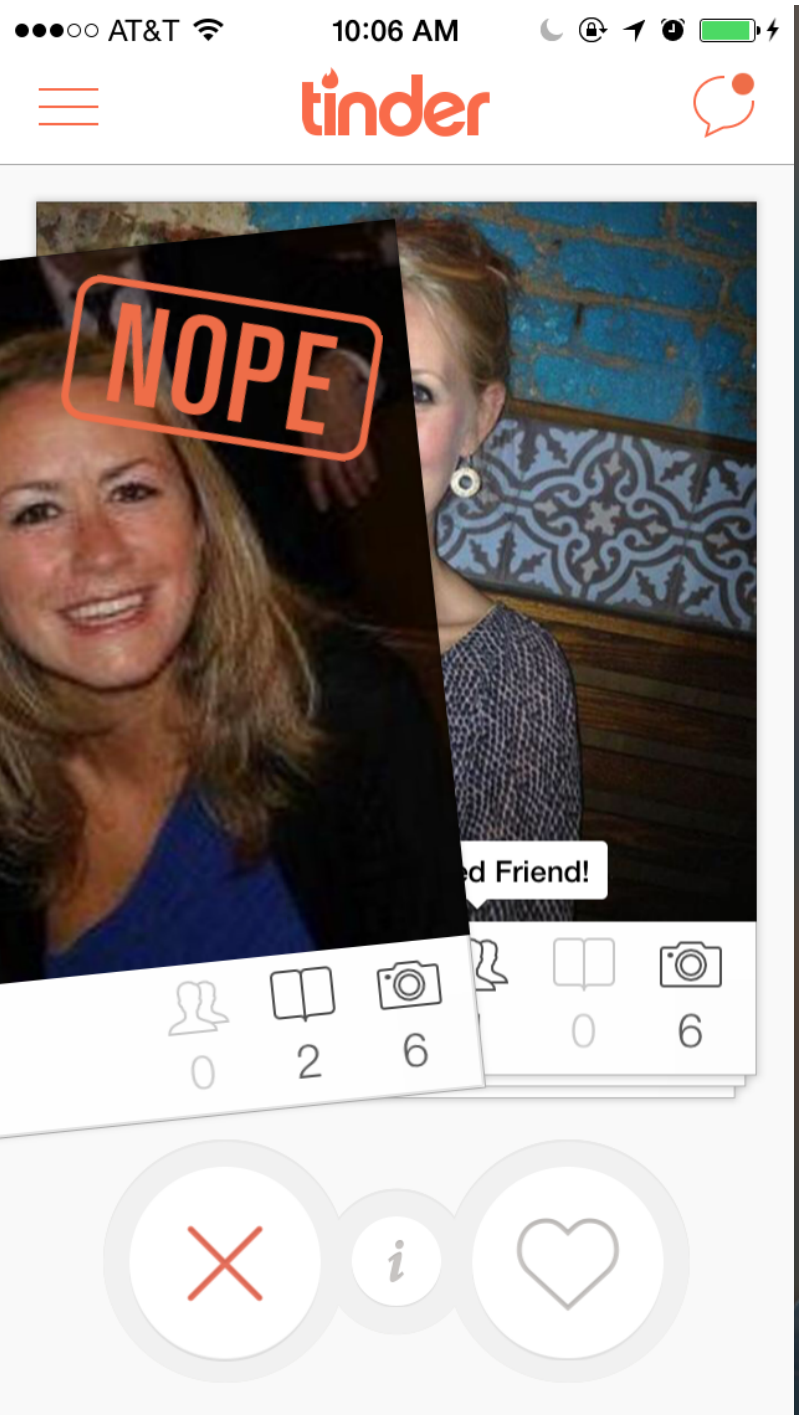
Integrate social sign in methods that allow users to login through their existing accounts. This means they have one less username/password combination to worry about, and at the same time, you don't have to worry about password security as much. Facebook, Twitter and Google are the major OAuth login providers and depending on the platform and target audience, you can implement all or either of these in your app instead of having users set up a separate account that they may

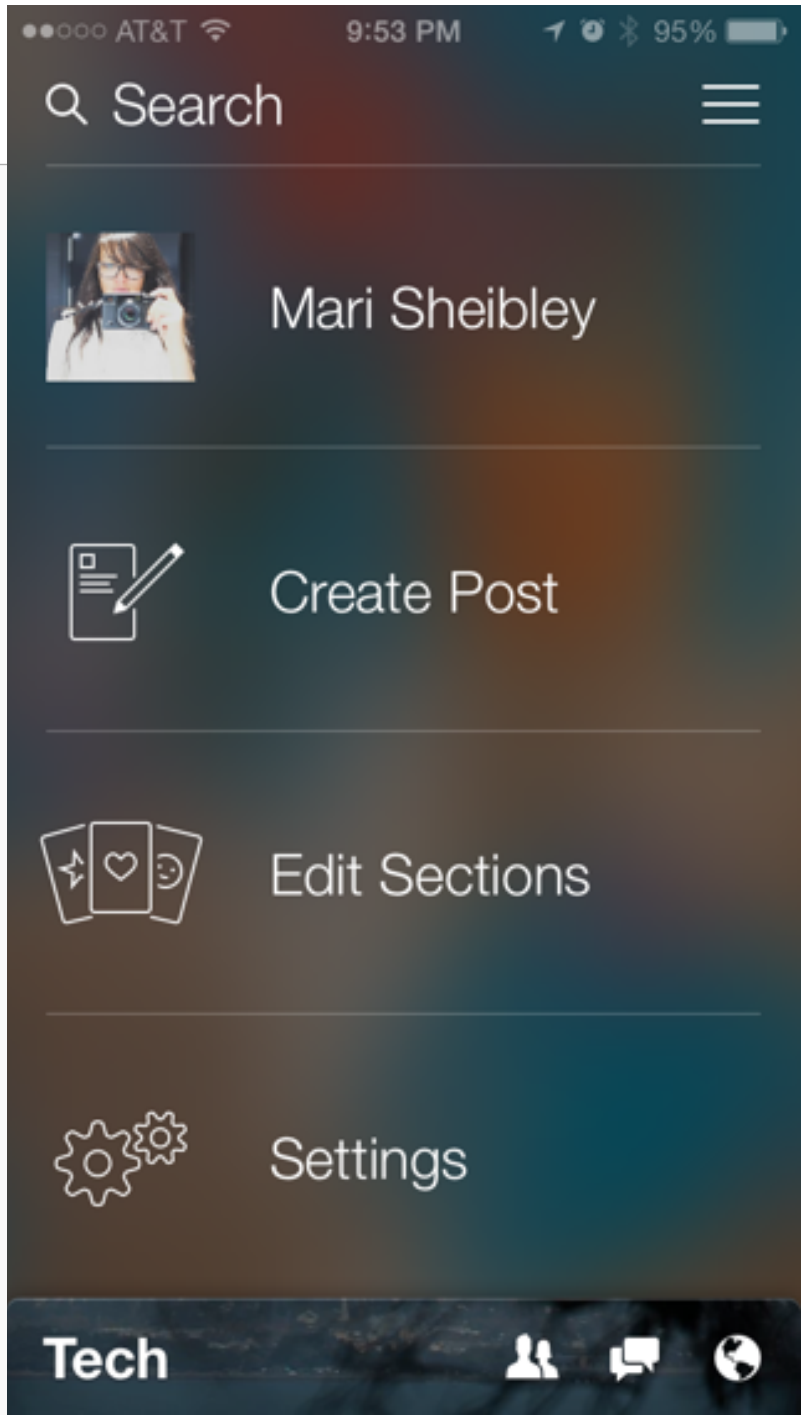
or may not end up using in the future. Using this signup and login pattern can also provide you with some basic data about the user (which feeds into data auto-population as they use the application), all the while making it easier on them by not forcing them to type their details into the strange new app they just downloaded. This simple feature can go a long way in drastically improving your UX, and no wonder this pattern is well on its way to becoming an expectation.

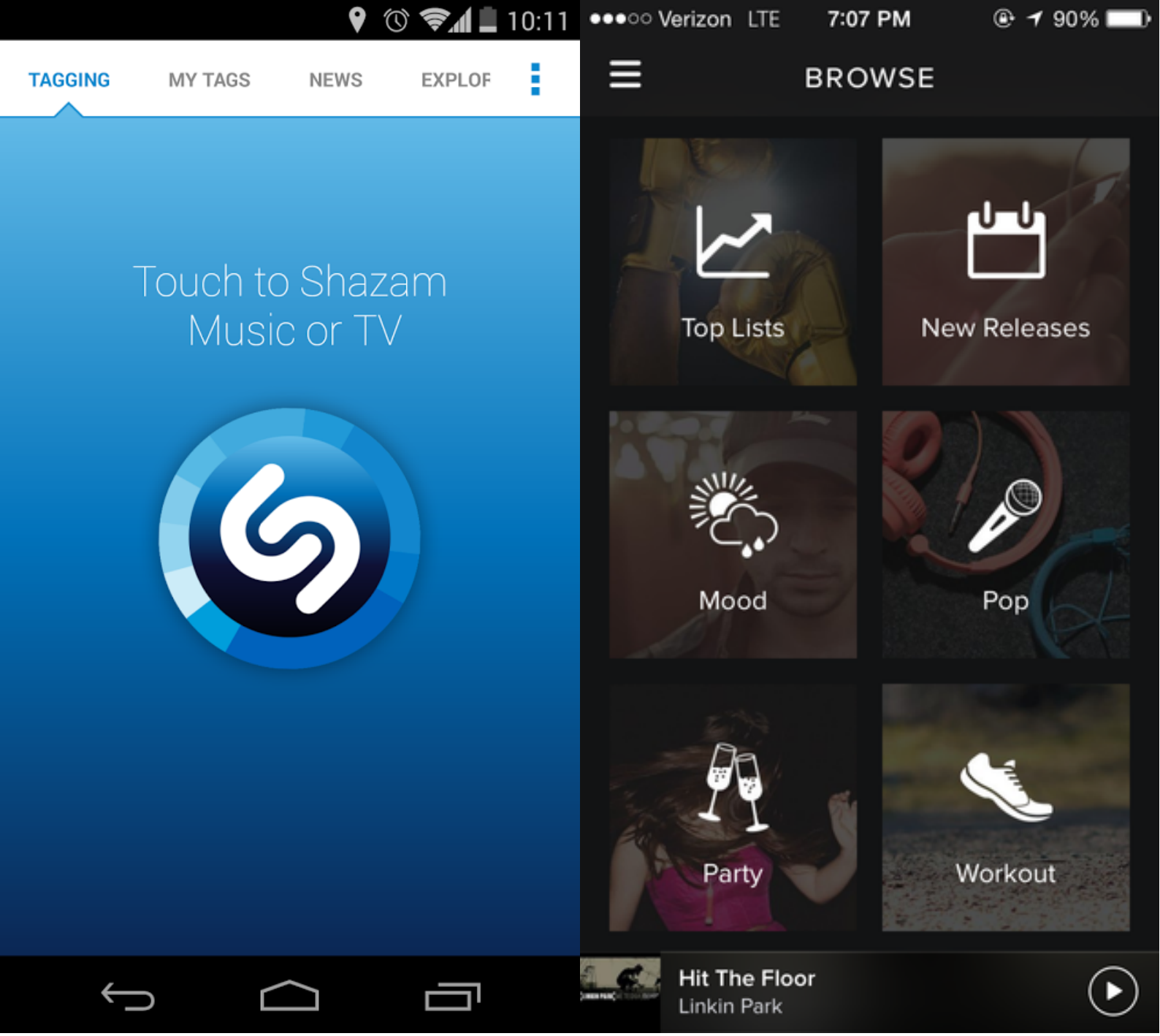
# Huge Buttons

Examples

Tinder, Facebook Paper, Shazam, Spotify






## Problem

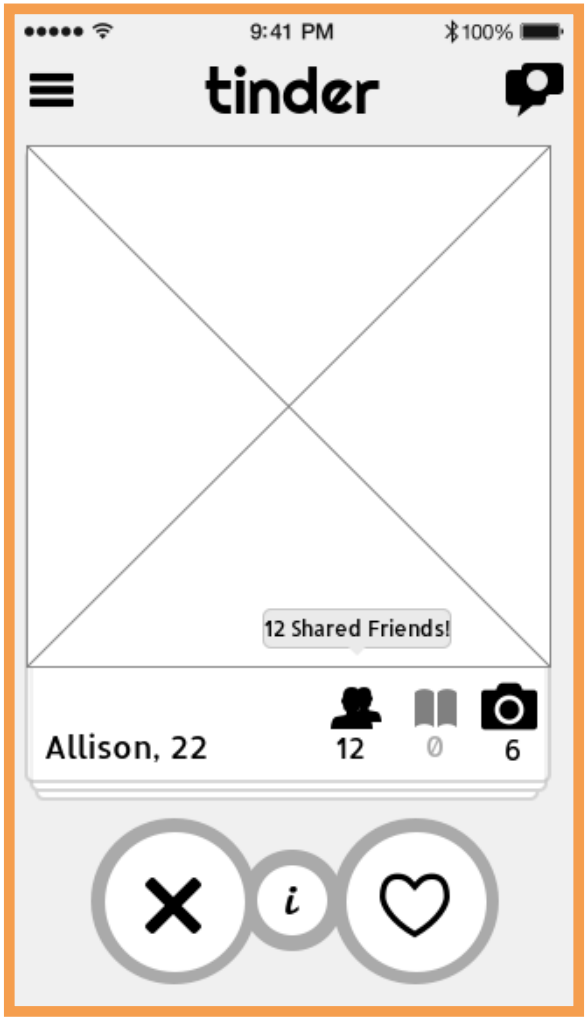
The user wants to know immediately which actions they can take.

## Solution

The ideal touch screen tap target size may be 72px, but some apps like Tinder also give you huge buttons so you know exactly what to do and can do it quickly wherever you are and whatever you're doing - it's pretty hard to miss these massive buttons, even if you're not looking. This is particularly valuable in more simple applications where there are limited actions a user needs to take and, thus, more reason to make it easier for them to take those actions in various contexts. Shazam for exam-

ple, is meant to be used while watching TV or listening to music, and it really only does one thing. The huge buttons are a great improvement for the user who's trying to multitask in this distracted state.

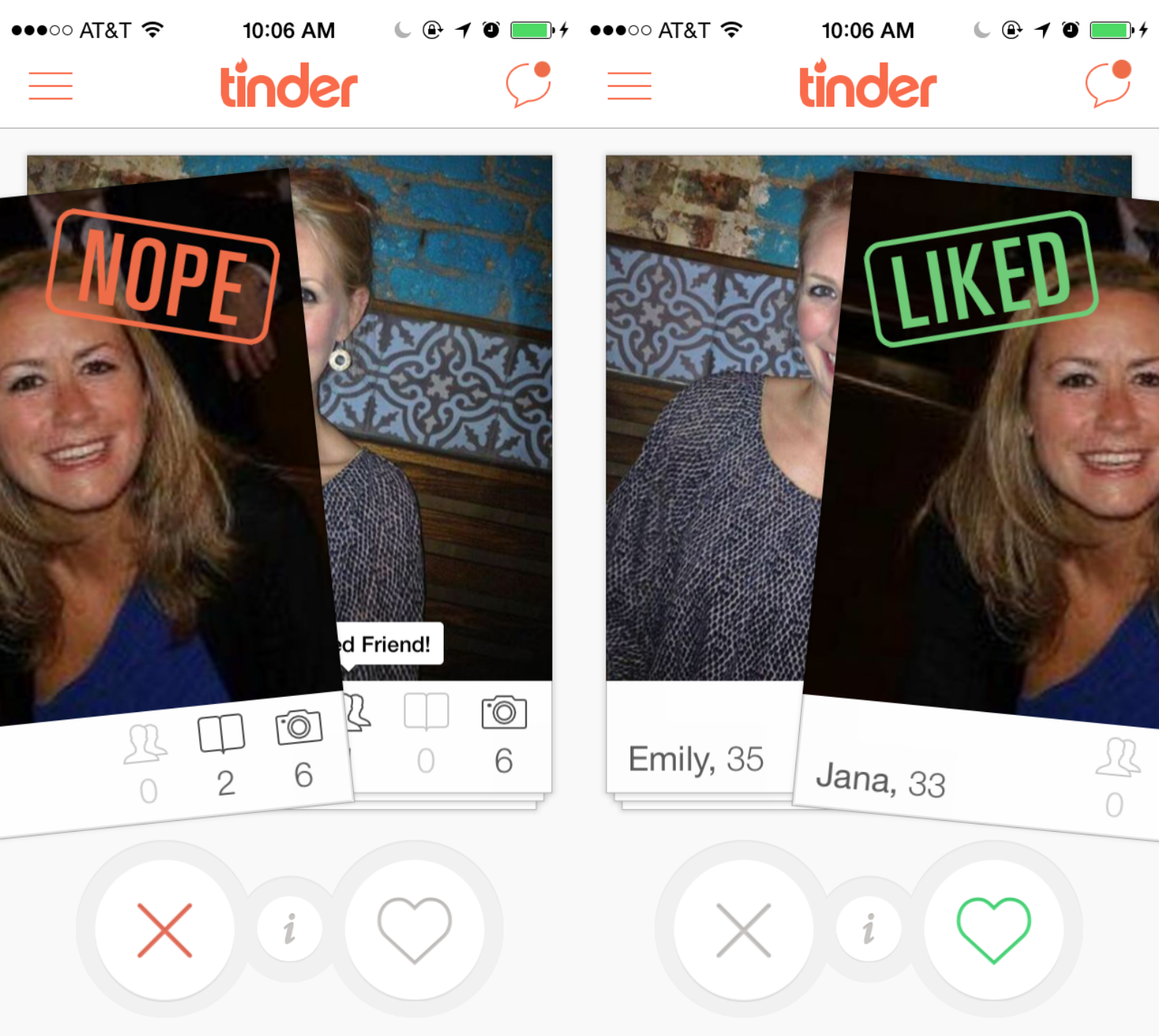
Click to use this wireframe in UXPin

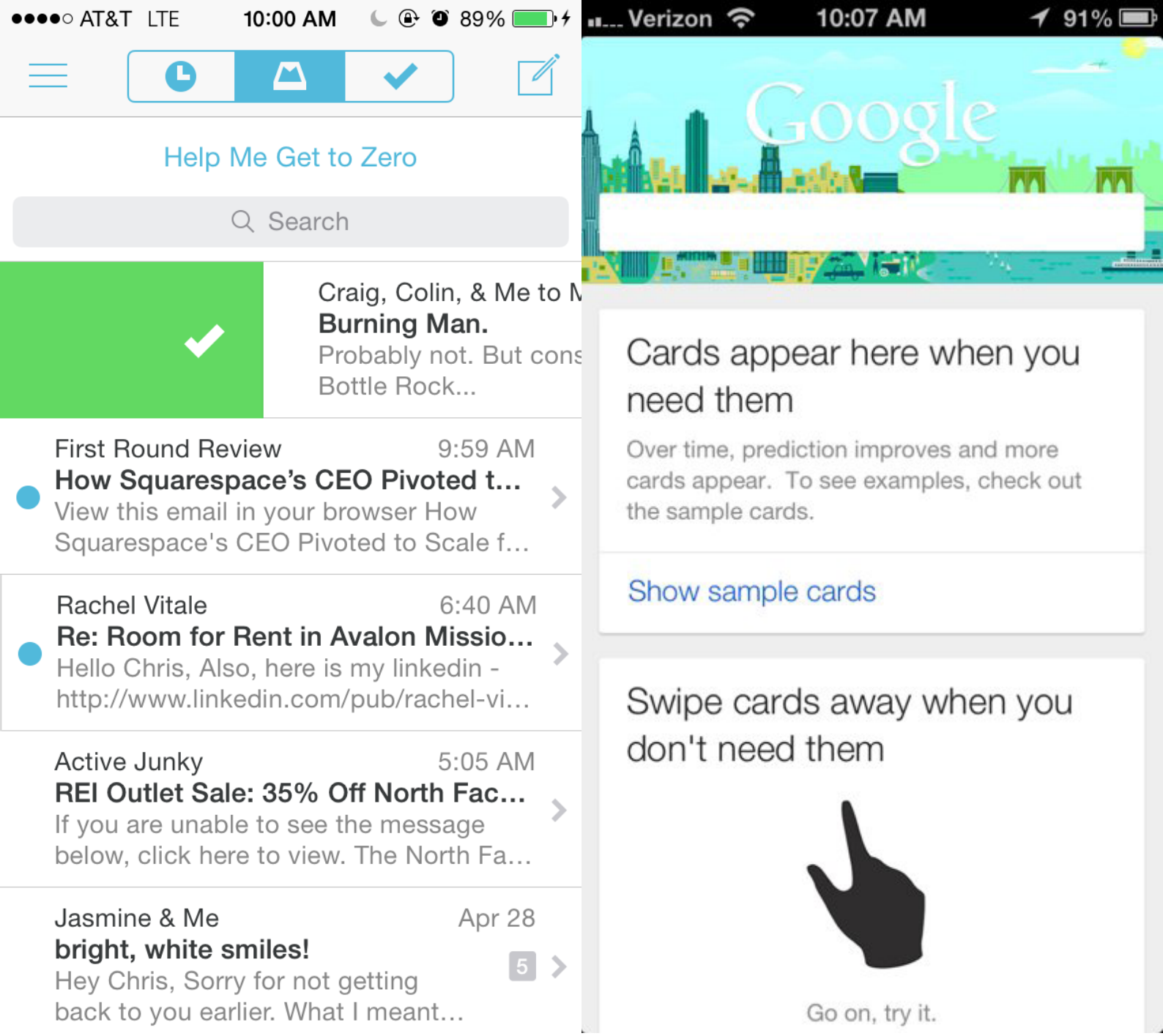


# Swiping for Actions

Examples

Tinder, Mailbox, Google Now






## Problem

The user wants to focus on particular content.

## Solution

Allow content to be swiped or moved out of the way. This provides users with a very intuitive way of handling the information on screen. For example, the “cards” in Google Now can be swiped away when you don’t need them to clear up the clutter; similarly, profiles in Tinder can be swiped to the right or left to indicate a positive or negative response. This pattern is different from the swipe views we talked about in navigation patterns. Here, the swipe gesture is being used for an action rather than

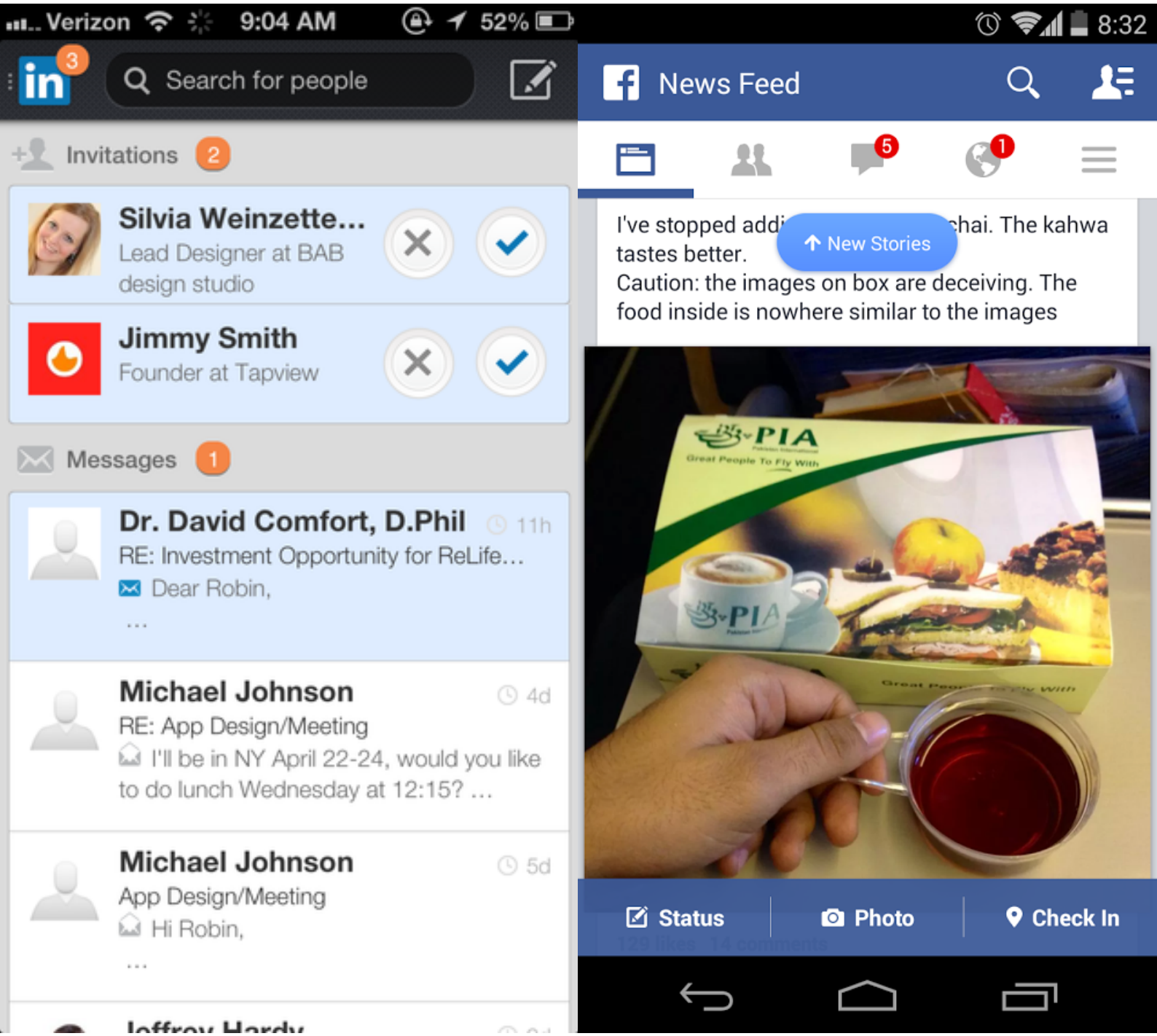
just browsing.

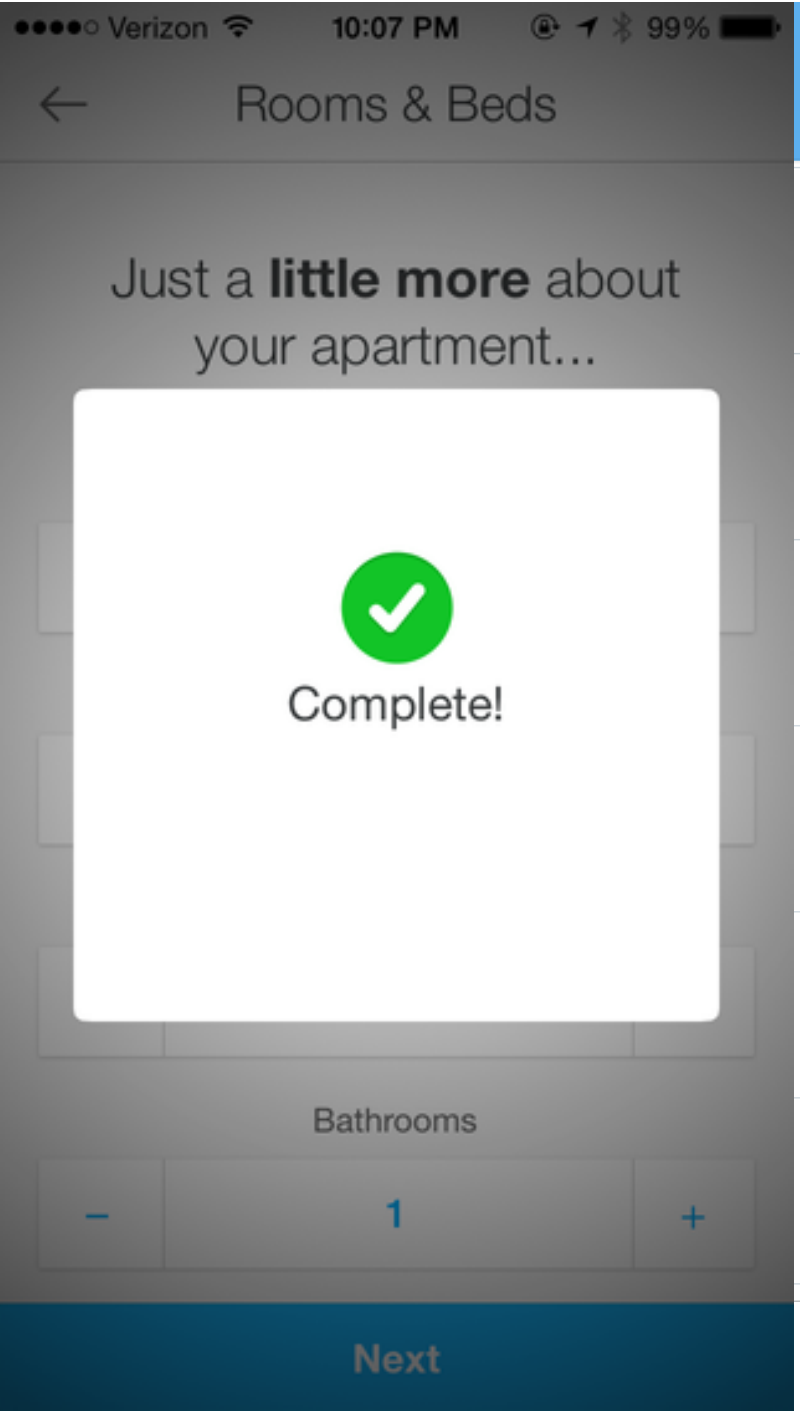
Some apps combine the two kinds of swipe patterns, for example Carousel, which lets you browse through multiple photos by sliding them to the side, as well as manipulating them by swiping upwards or downwards to share or hide them. Mailbox popularized the side-to-side swiping actions for email clients, allowing you to mark emails as read and schedule them for follow-up by swiping right or left, respectively. Secret let's you discover new actions the way it let's you discover new menus. Swipe left on a secret and you like it.

# Notifications

Examples

LinkedIn, Facebook, Airbnb, Twitter





## Problem

The user wants to know about new activity or actions they should take at a glance.

## Solution

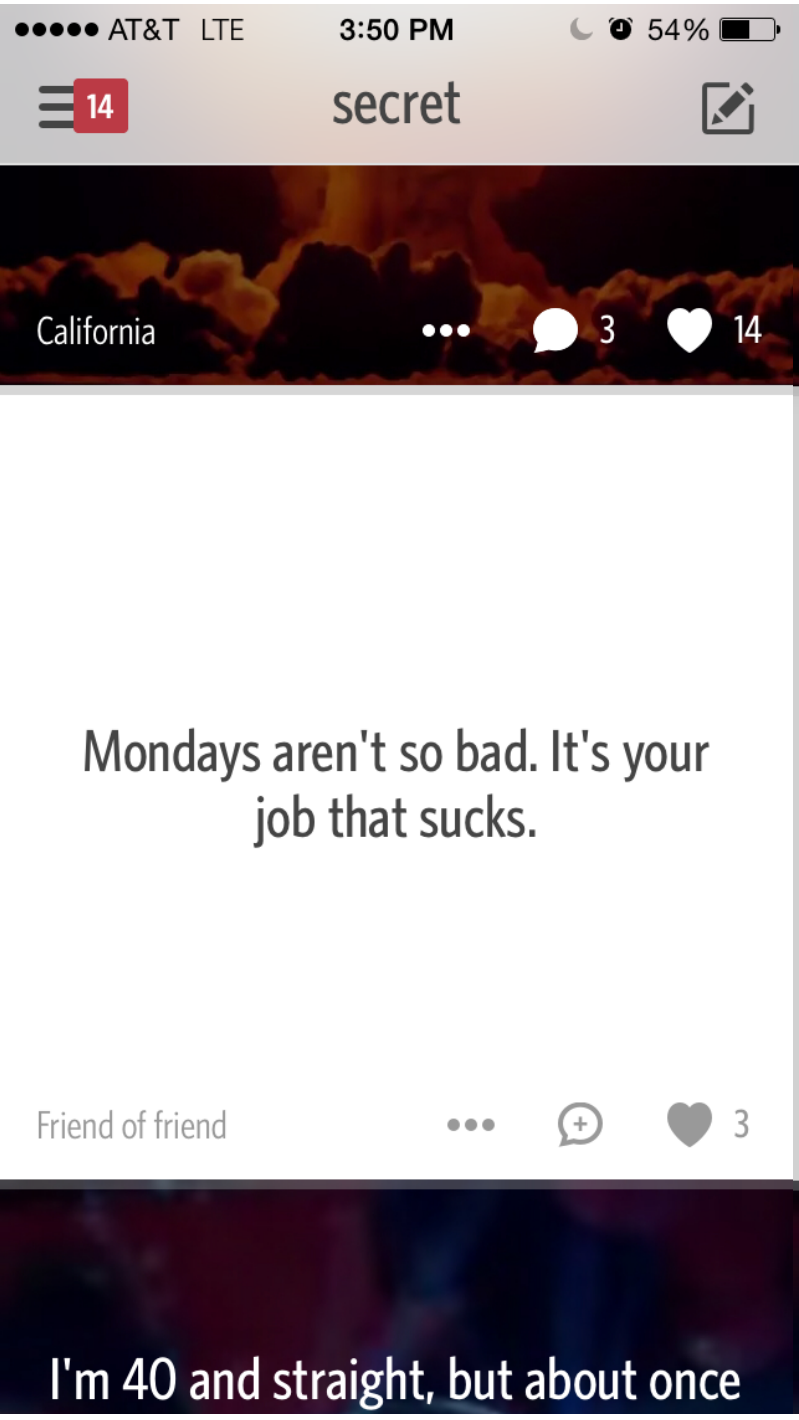
Highlight recent activity by visually marking new content. There are several implementations of this pattern. For example, placing a numbered badge on the label was popularized by iOS but can be seen in many other apps now such as LinkedIn, Facebook or Quora. Twitter does this as well for the notifications button but also has a small dot on top of the timeline icon to indicate new activity in a more subtle way. Another way to display notifications is with a banner that drops down within the

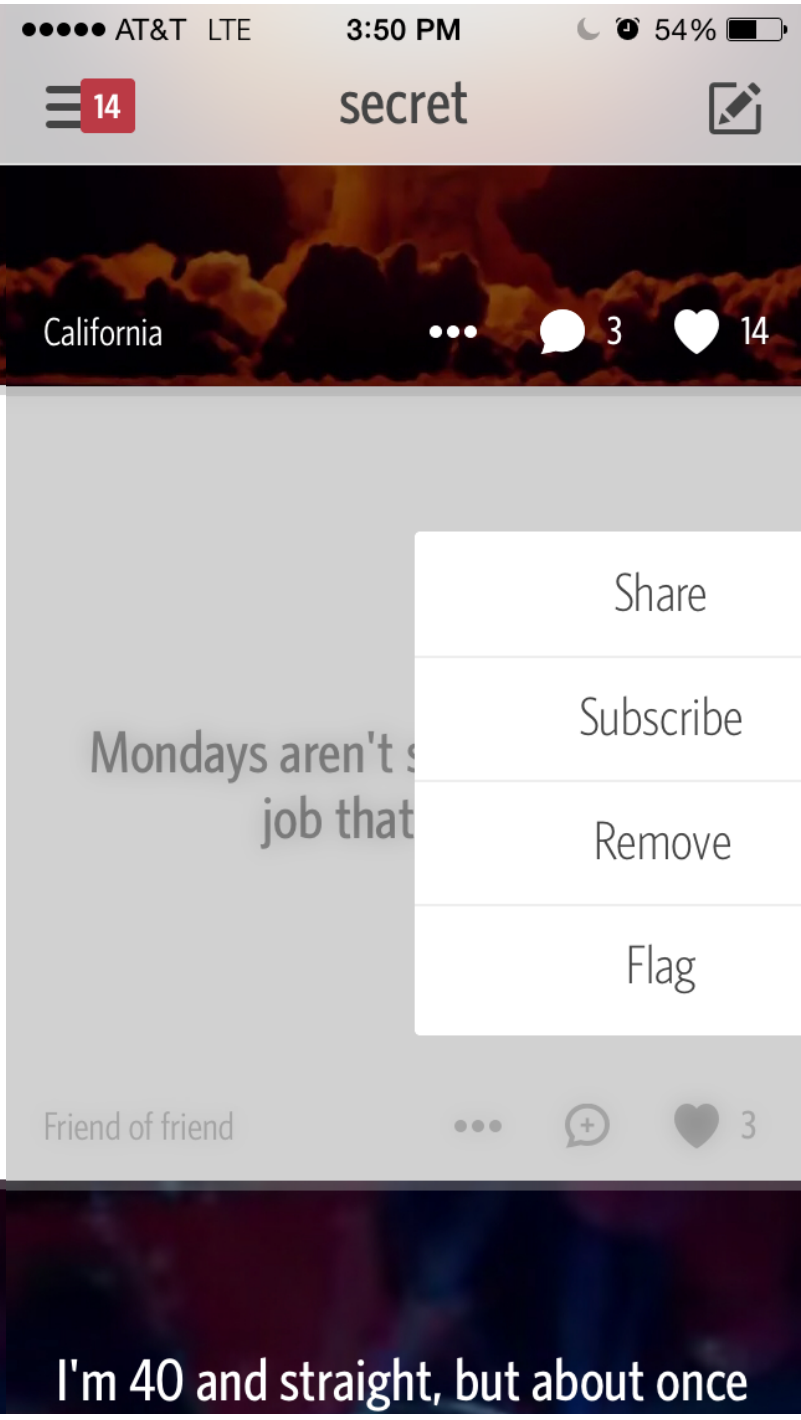
app to show new activity. The Facebook app does this as well, showing a small pop-up when there are new items in the newsfeed.

# Discoverable Controls

Examples

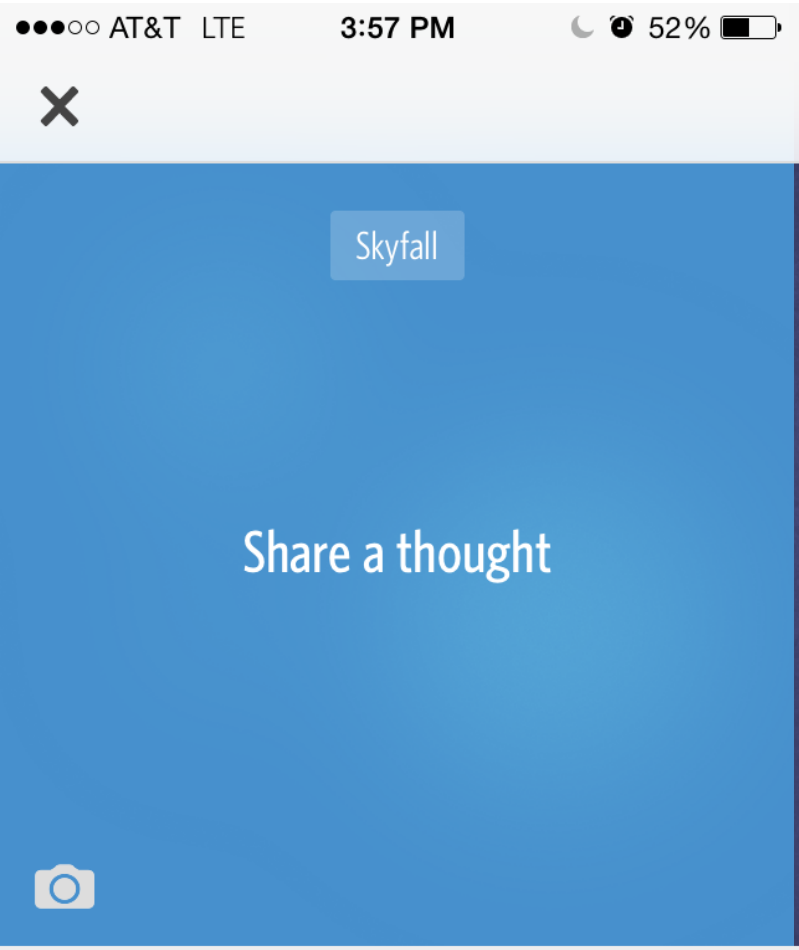
Secret

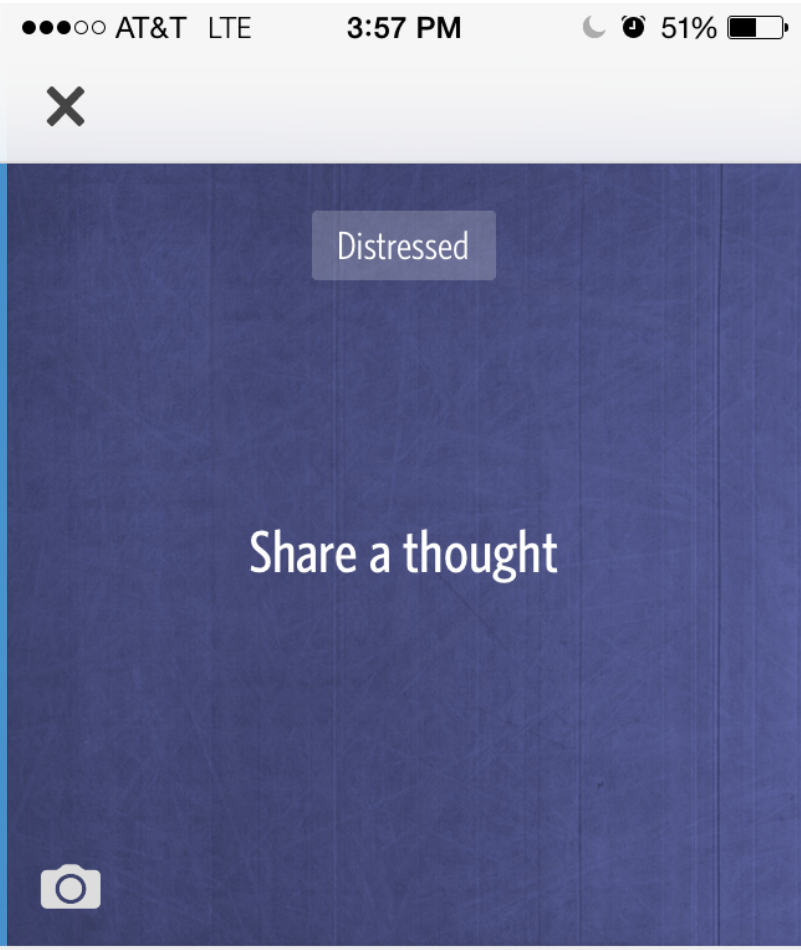


Examples

Secret



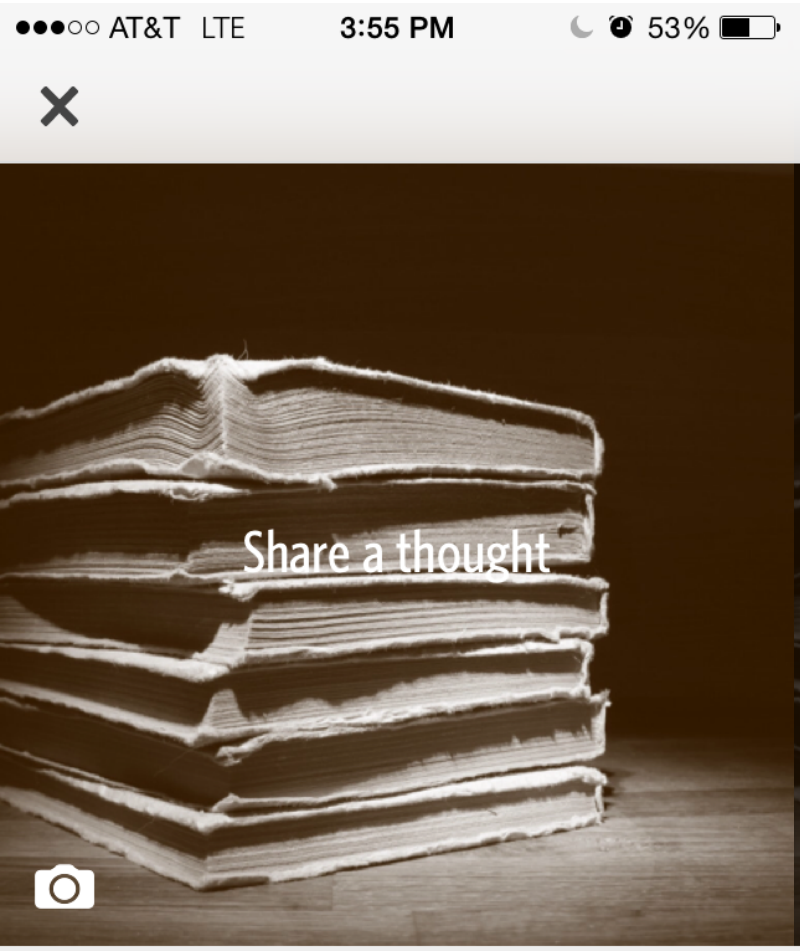




Who will see this?

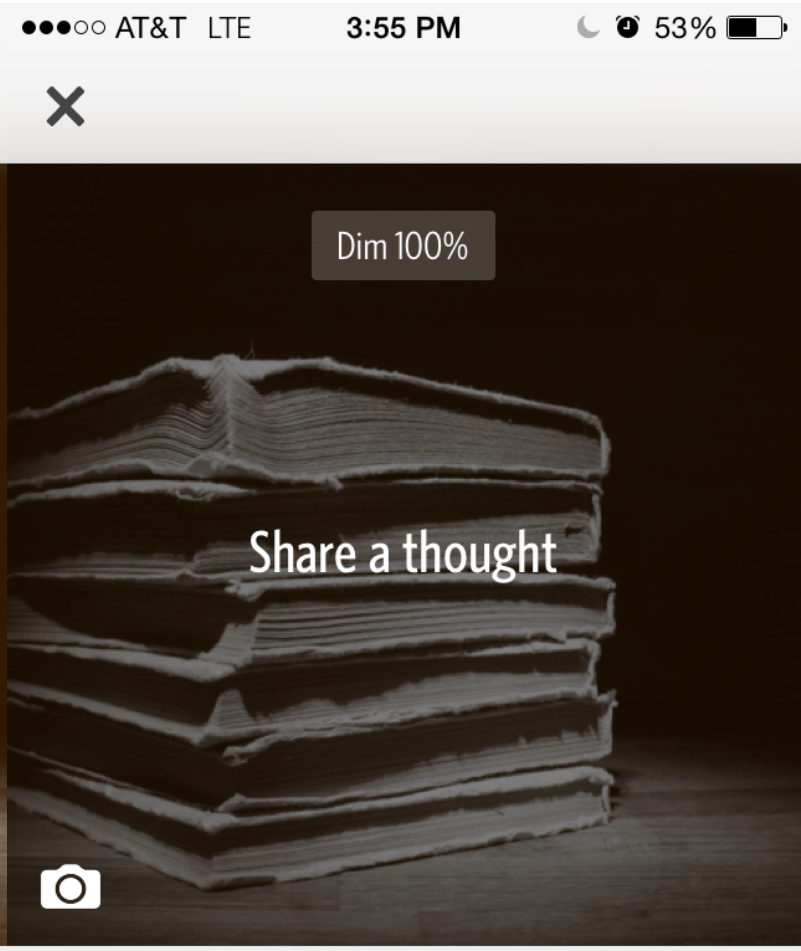
Who will see this?

Examples

Secret



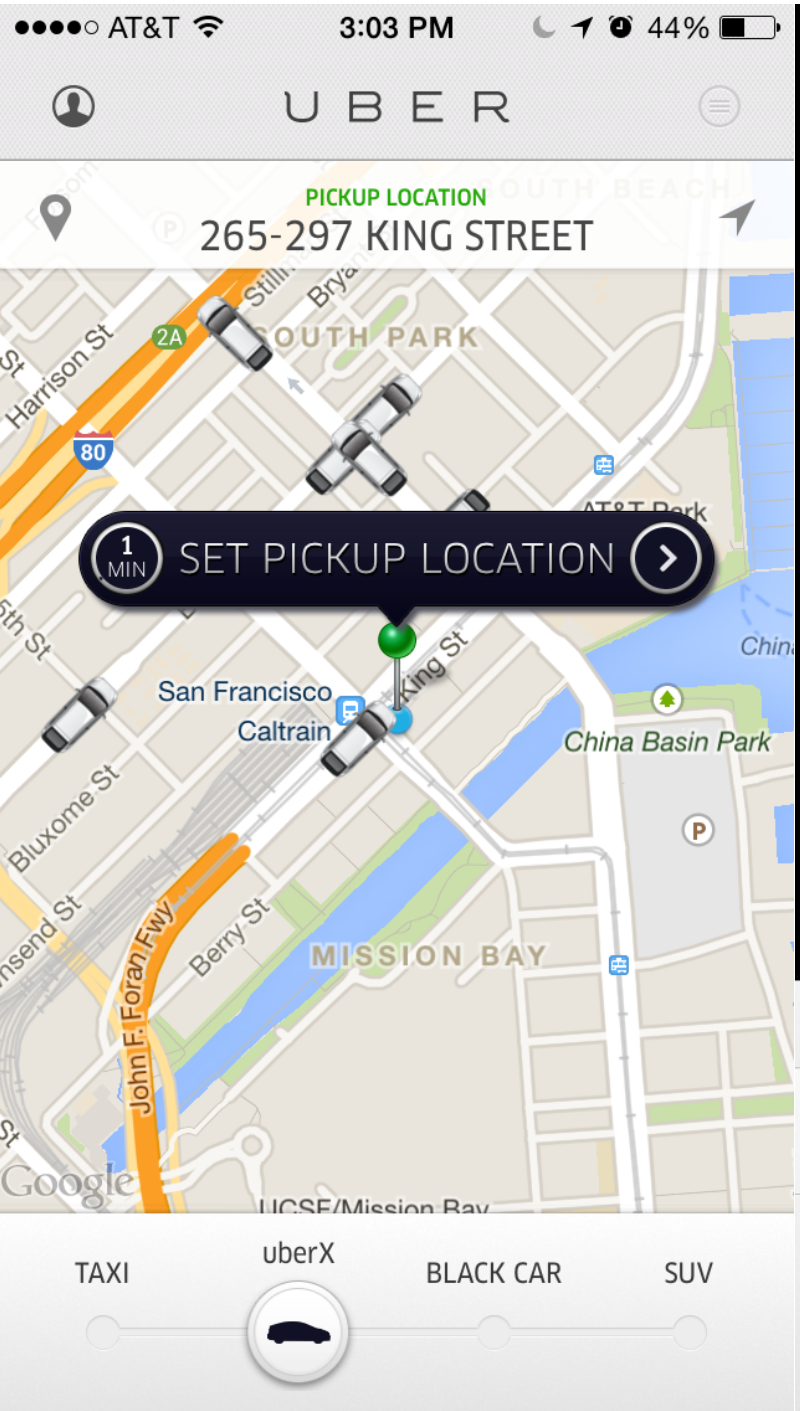

Who will see this?

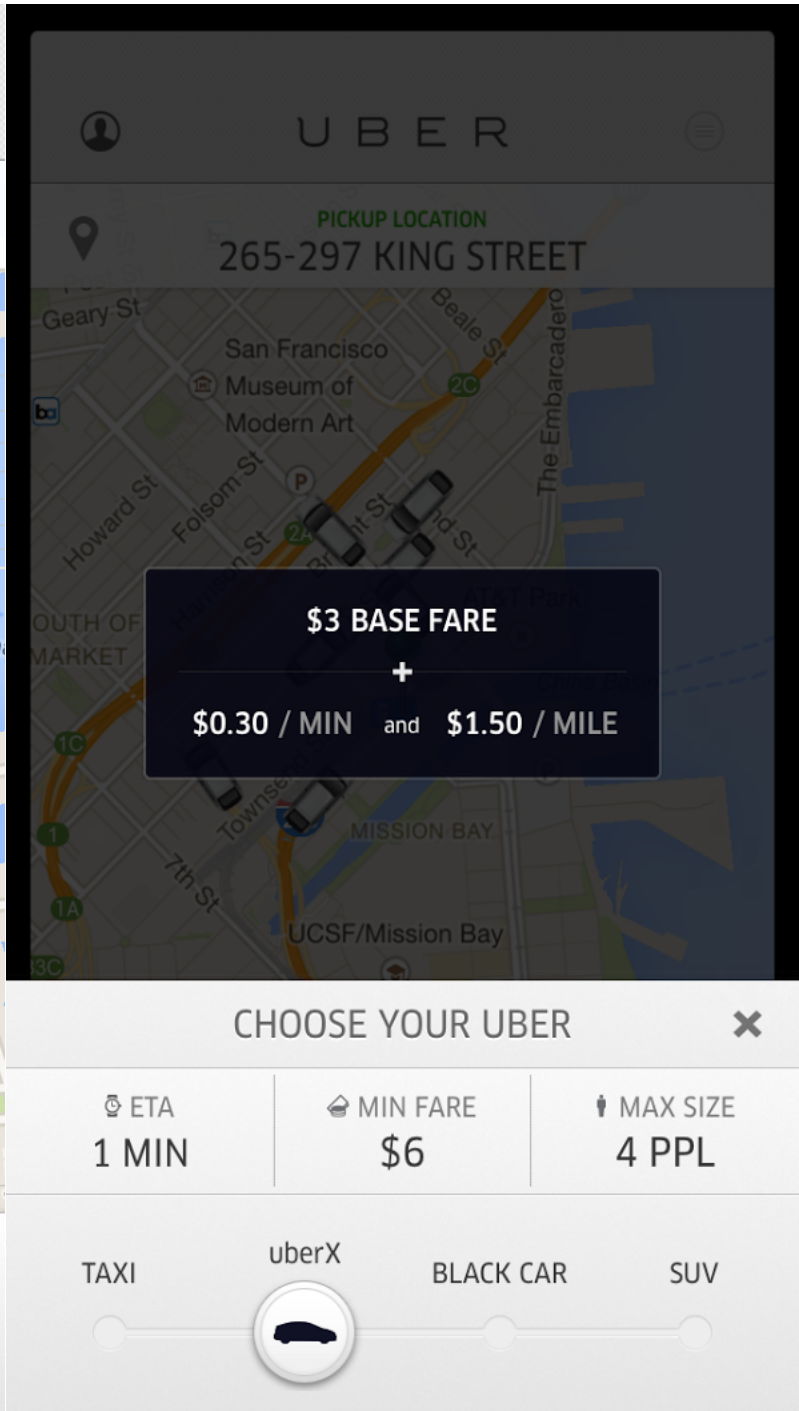



Who will see this?

Examples

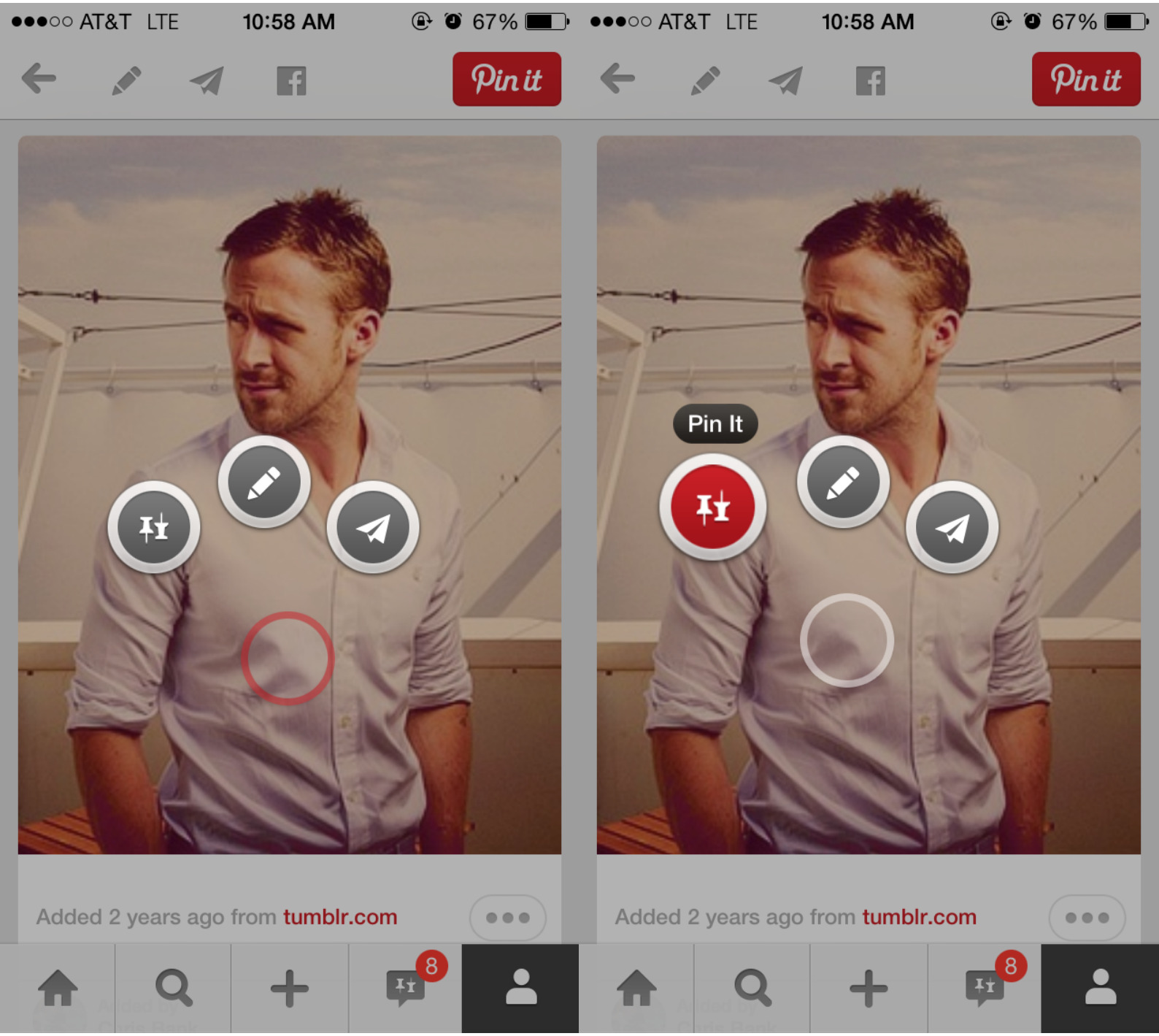
Uber



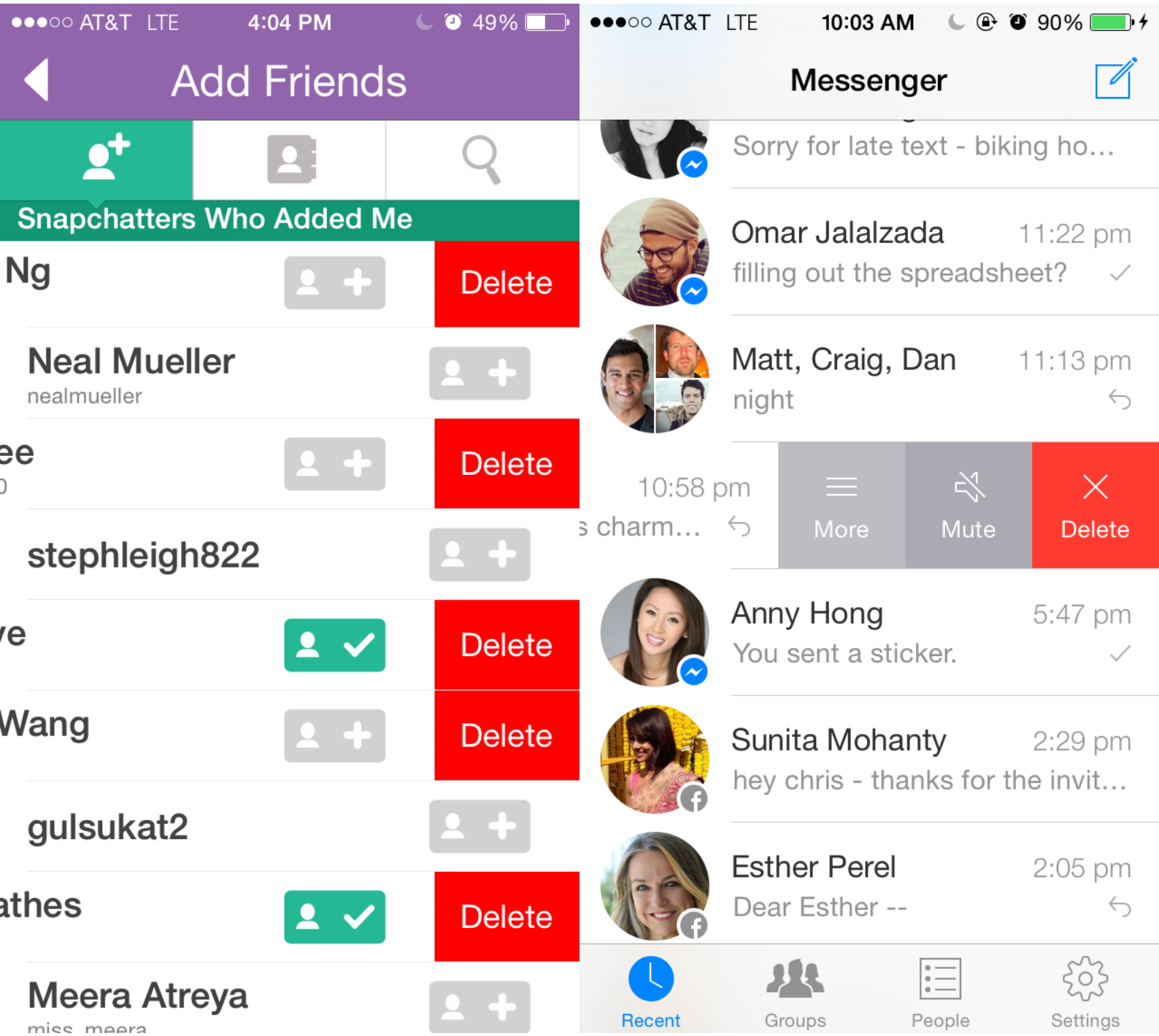
Examples

Pinterest



Examples

Snapchat, Facebook Messenger



## Problem

The user wants quick access to controls that are secondary or only relevant to specific sections or content in the application.

## Solution

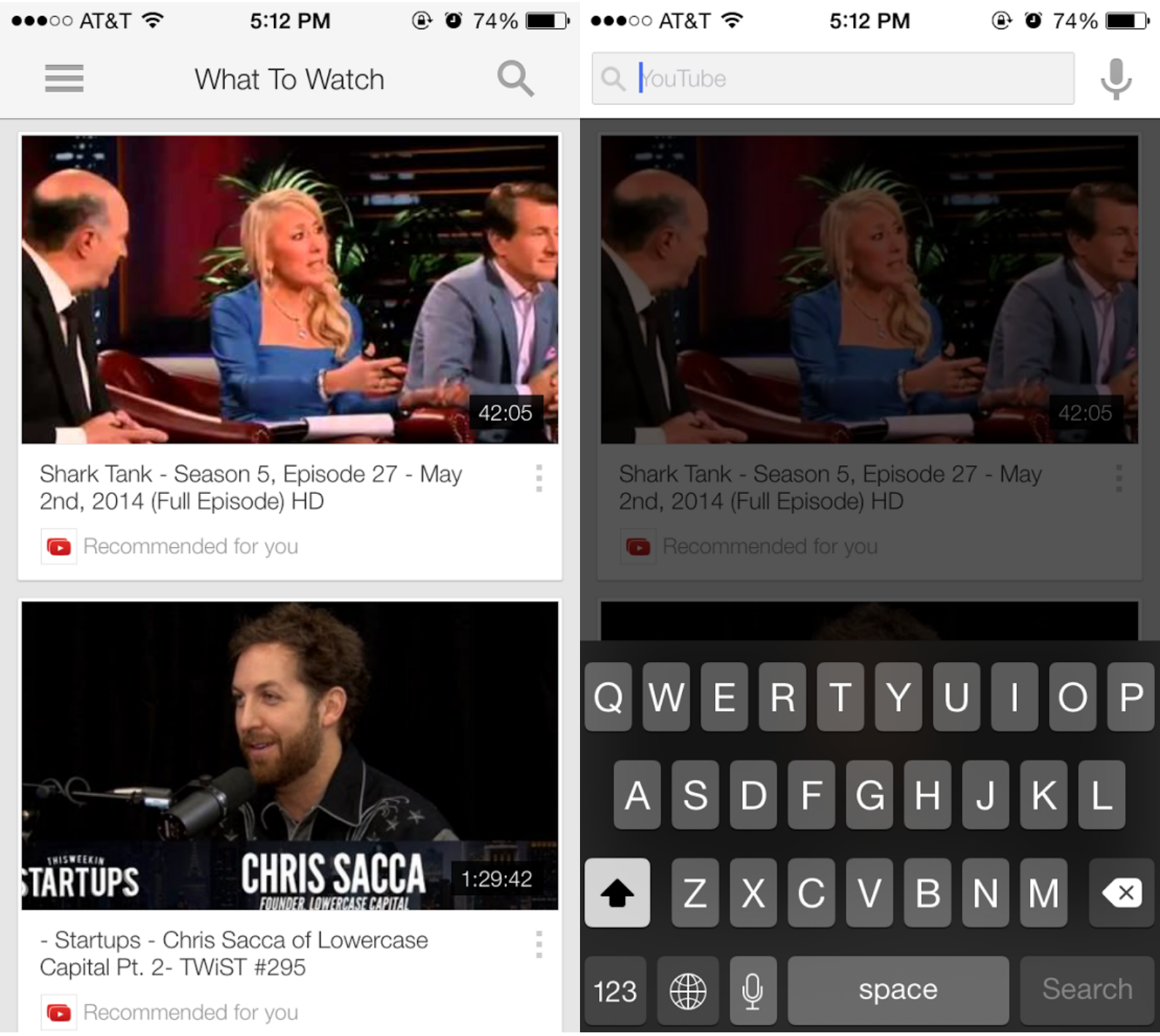
Clear up the clutter and let users discover particular actions only when they need them. These invisible controls can be accessed by any gesture - swipe, tap, double--tap, long-press etc. (which we talk about in the gestures pattern). This gives you the ability to keep these actions off-screen, saving some valuable real estate. Secret, for example, uses gestures instead of visible controls. Swipe right and you'll expose an action menu, which is a minimalistic version of a drawer pattern which we've covered earlier. When creating content, users can swipe horizontally or slide their finger vertically across the background to change its color and pattern, or in case a picture is being used, its brightness, saturation or blur. There are no other controls that let you do this - nor should there be. This UI design pattern is so intuitive and clean that you're bound to see a lot more of this type of interaction. Pinterest is another app that uses gestures to hide action buttons. A long-press on an image reveals buttons that let users pin or comment on it by dragging the pop-up control to the button.

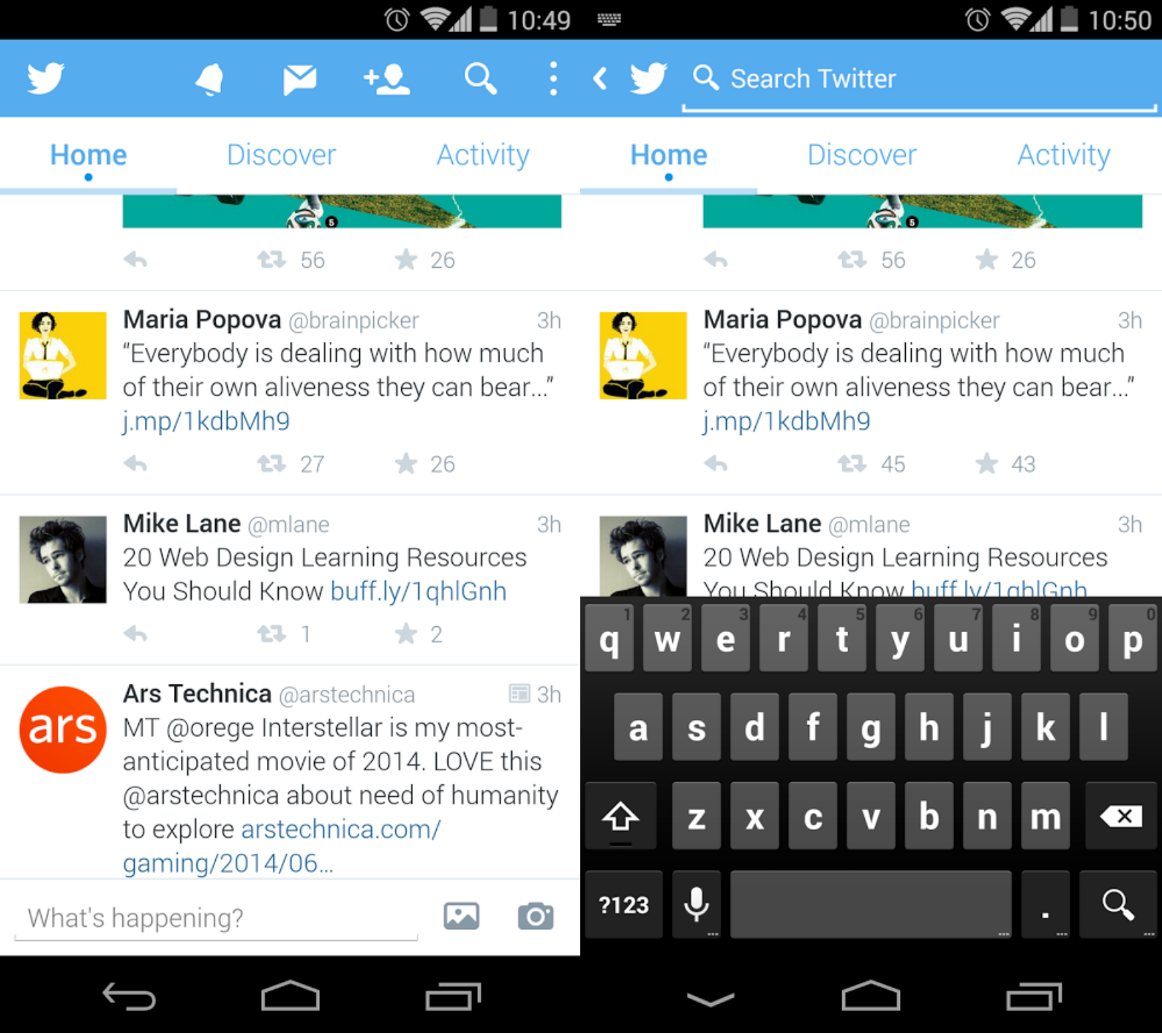
Uber is an alternative implementation of this design pattern. Uber also let's you toggle between booking a ride and seeing the fare estimation by tapping the slider button once you've chosen which ride type you want. This is a simple yet important UI design pattern that makes me smile every time I'm doing five things while trying to get a ride somewhere, but want to make sure Uber isn't ripping me off with surge pricing. Snapchat and Facebook Messenger let you access features when you need them by swiping any friend left.

# Expandable Inputs

Examples

YouTbe, Twitter






## Problem

The user wants to focus on the content instead of sacrificing screen real estate to controls.

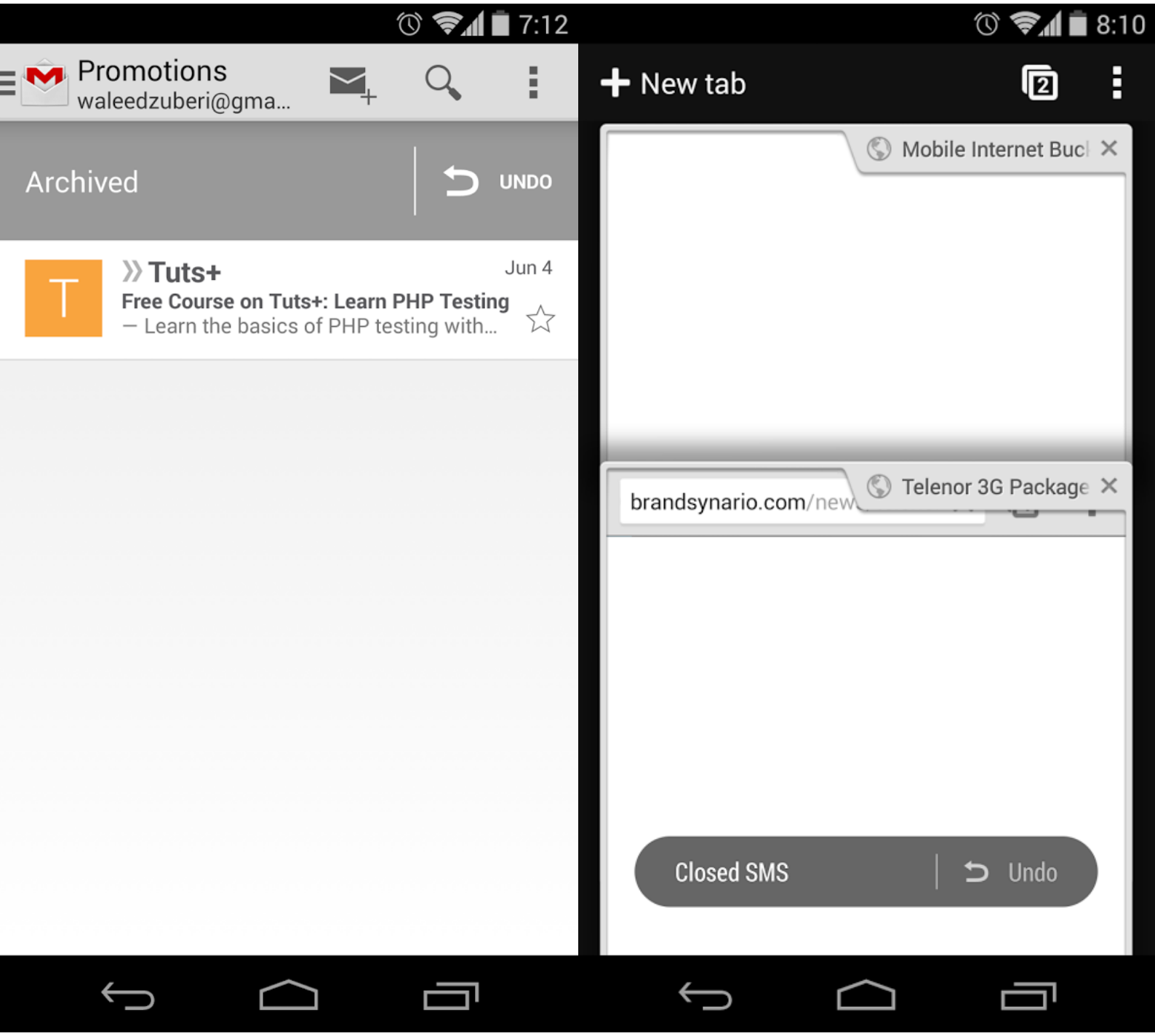
## Solution

Design controls that expand when the user taps on them. This keeps most controls out of the way until the user needs them. For example, YouTube and Facebook conserve screen space by hiding the search bar behind an icon that expands into a search bar when the user taps on it.

# Undo

Examples

Gmail, Chrome



### Problem

The user wants to take actions quickly without interruptions (ex: confirmations) but with the option of reverting accidental actions.

### Solution

Provide an easy way for users to undo their actions instead of just asking them to confirm beforehand. Situations where an action can cause inconvenience or loss of data if done by accident or in haste, for example deleting an email or editing some text. The user may have completed an action because they didn't know what to expect; a forgiving UI that let's them experiment can be more engaging and friendly. The ability to undo is also great for power users, who will appreciate feeling more in control without the UI holding their hand throughout the process repeatedly asking if they're sure they want to proceed. A confirmation popup can be useful at explaining what's about to happen, but user's may not understand the implications until they see the result of their action. In cases like these it's best to get out of the way while providing a safety net in case of mistakes.

Share this ebook
with friends!

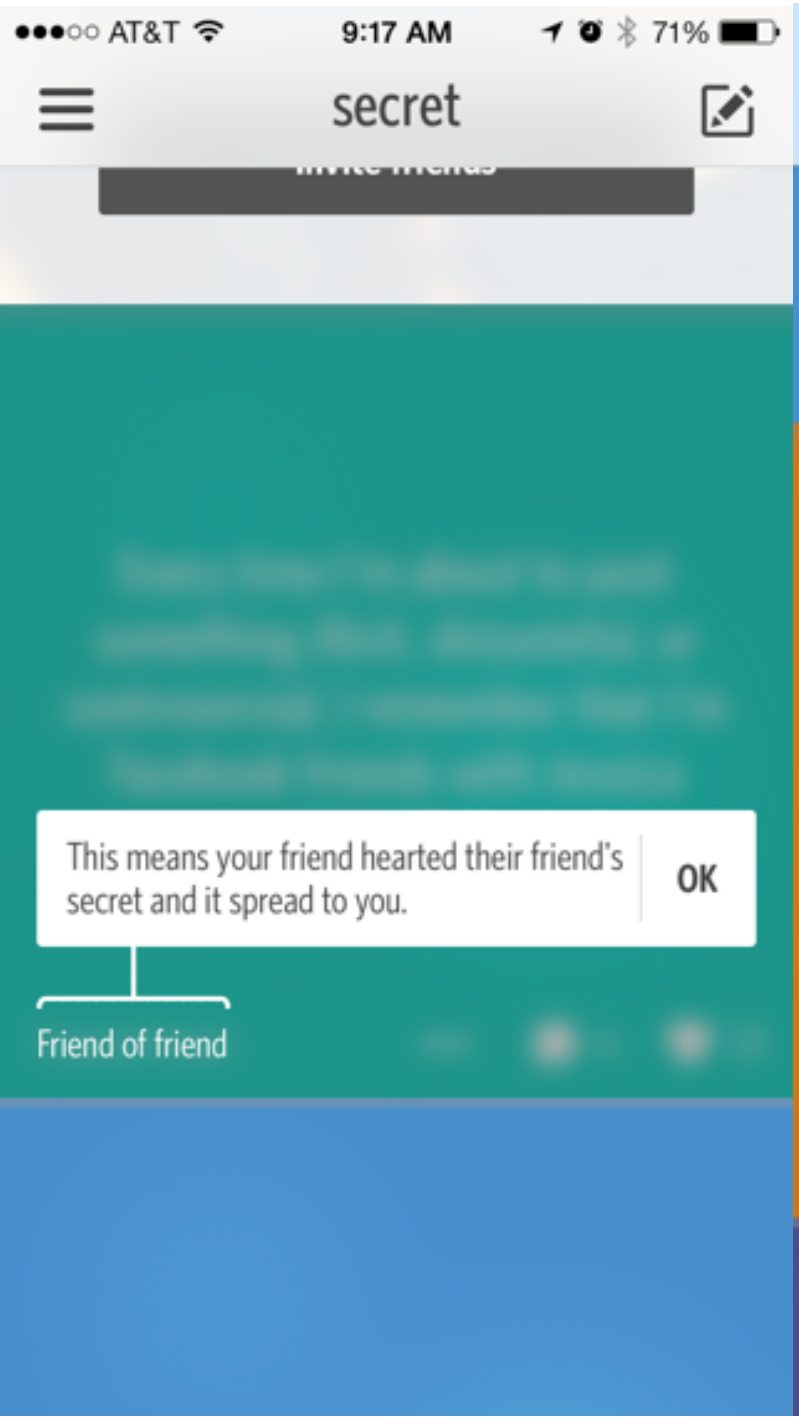
# 4

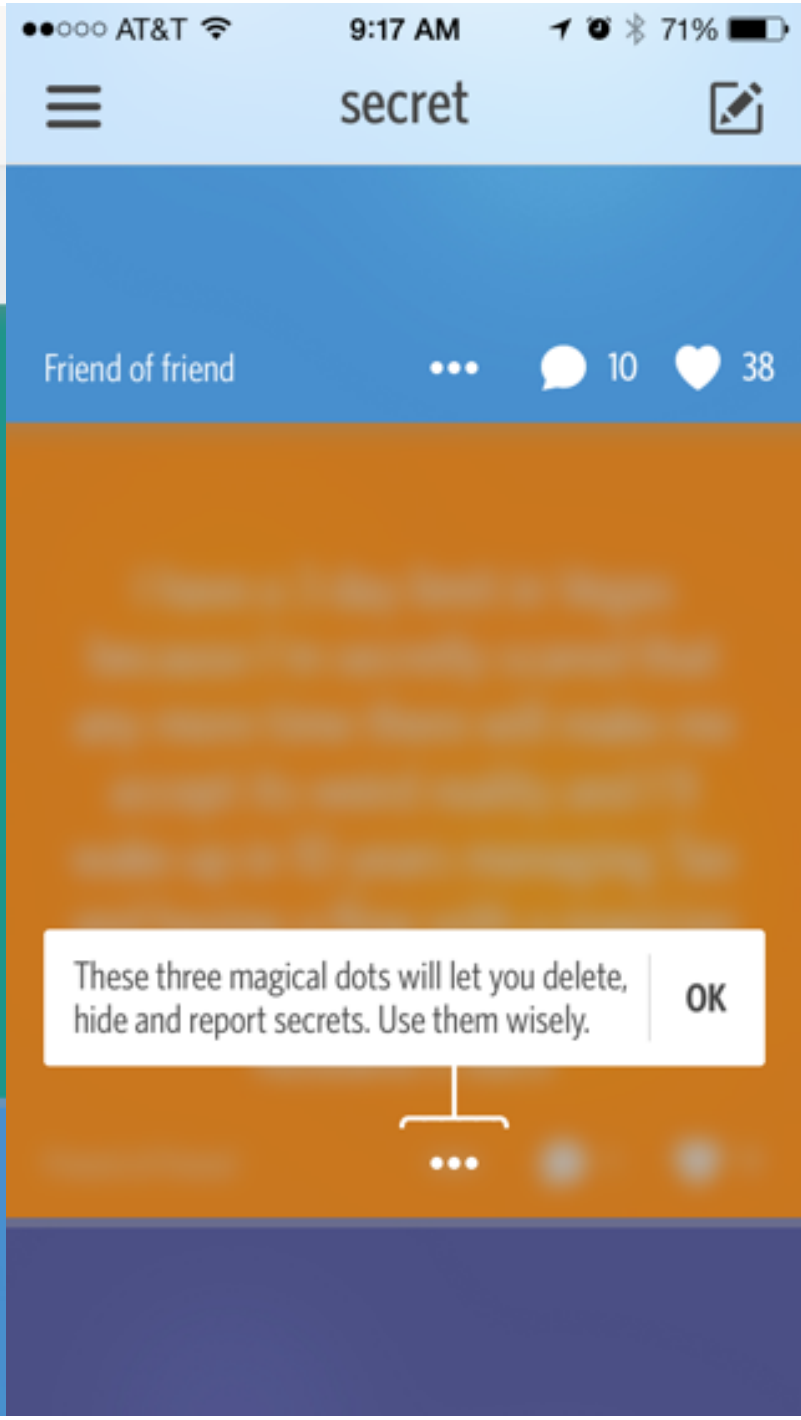
# Navigation

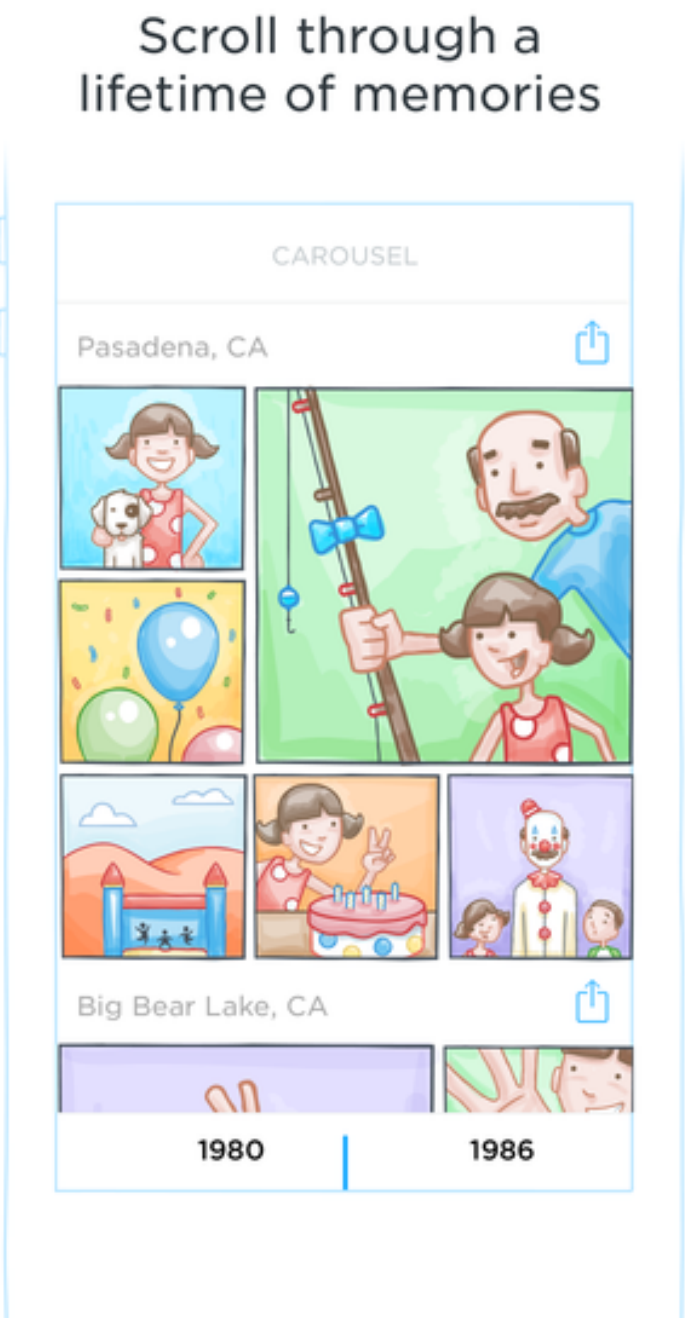
# Walkthroughs & Coach Marks

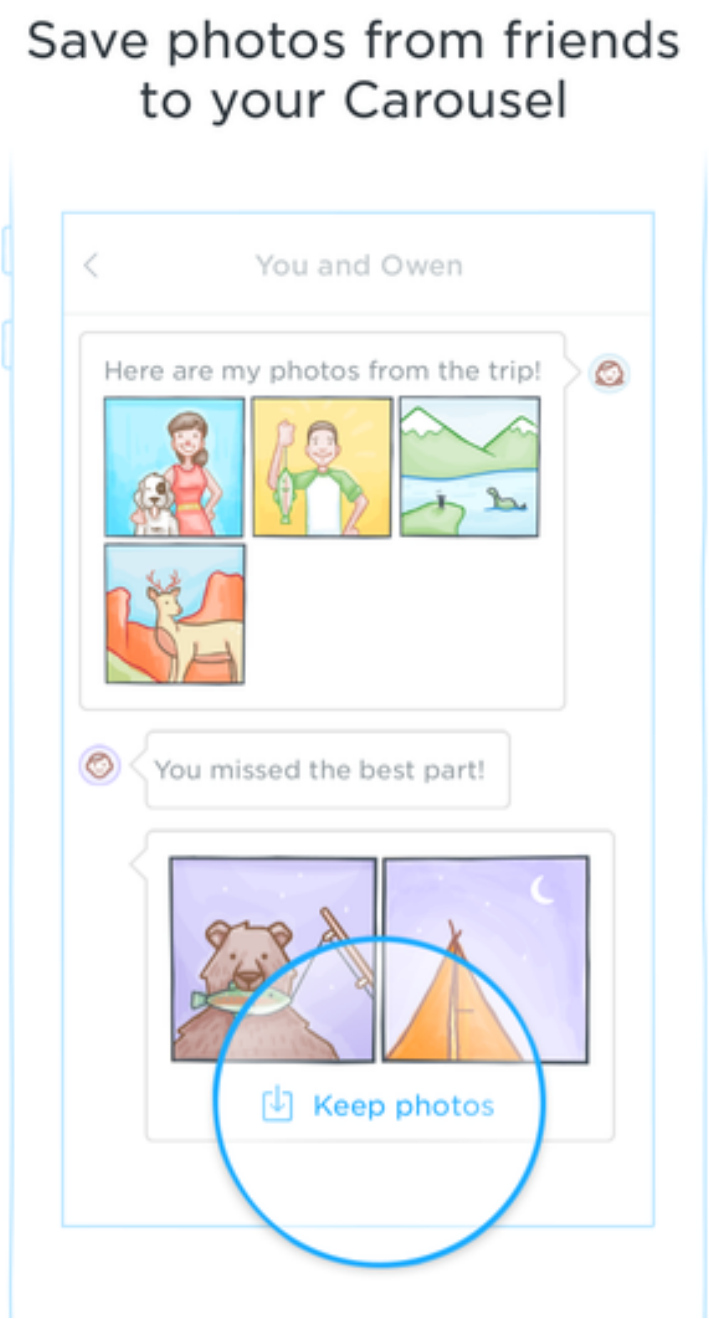
Examples

Secret, Carousel






## Problem

The user wants to know how to use the different features of the application.

## Solution

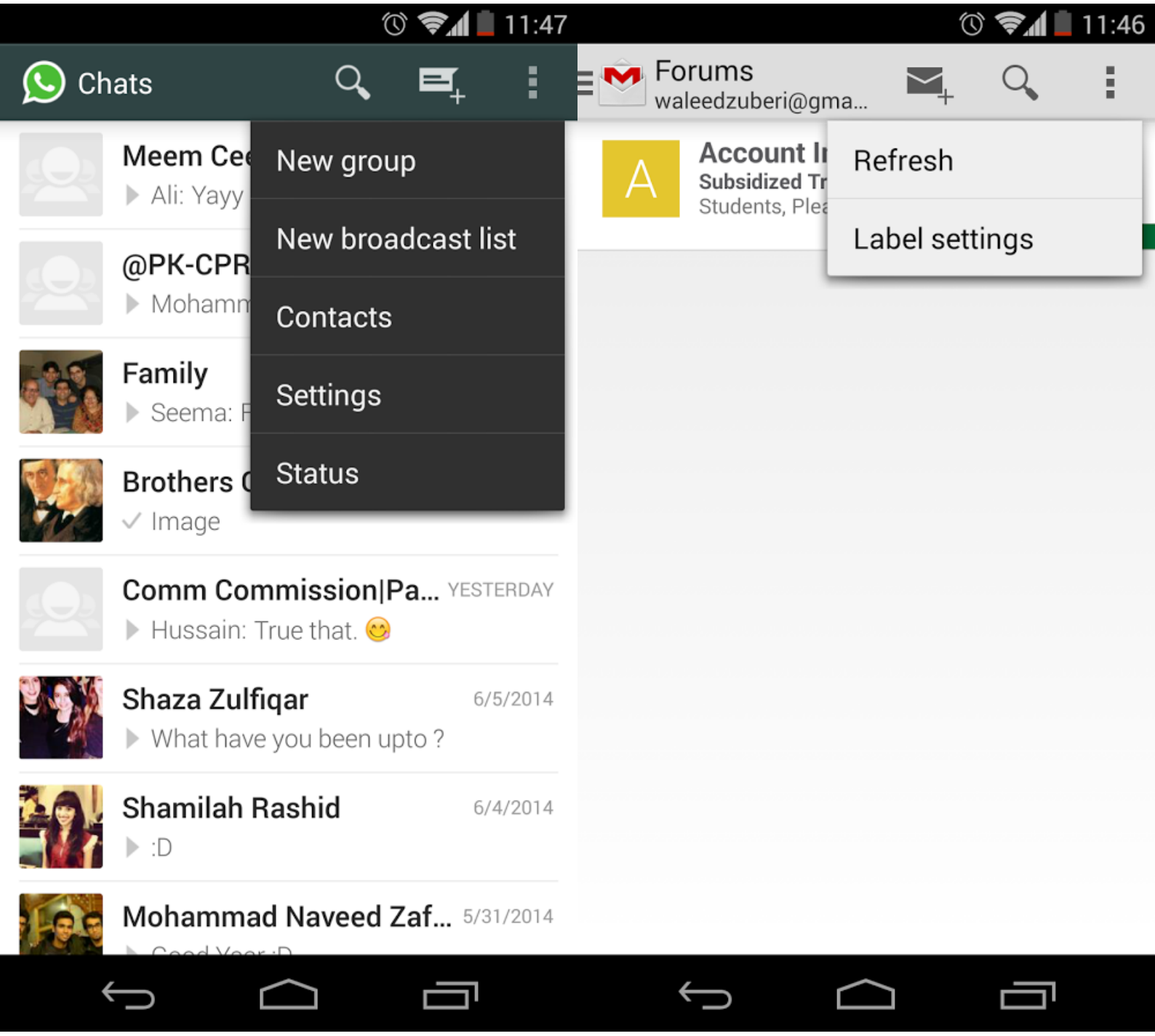
Design a walkthrough or tutorial that demonstrates how each function works. A lot of apps have begun using this technique to show users around when they first launch and there are two basic ways of doing this. Some apps , like Secret or YouTube, go the route of overlay instructions, highlighting important parts of the UI with “coach marks” to explain what they do. Alternatively, some apps like Carousel and Duolingo use the first launch to show a slideshow that walks users through the entire experience, effectively explaining what the user can accomplish with the app.

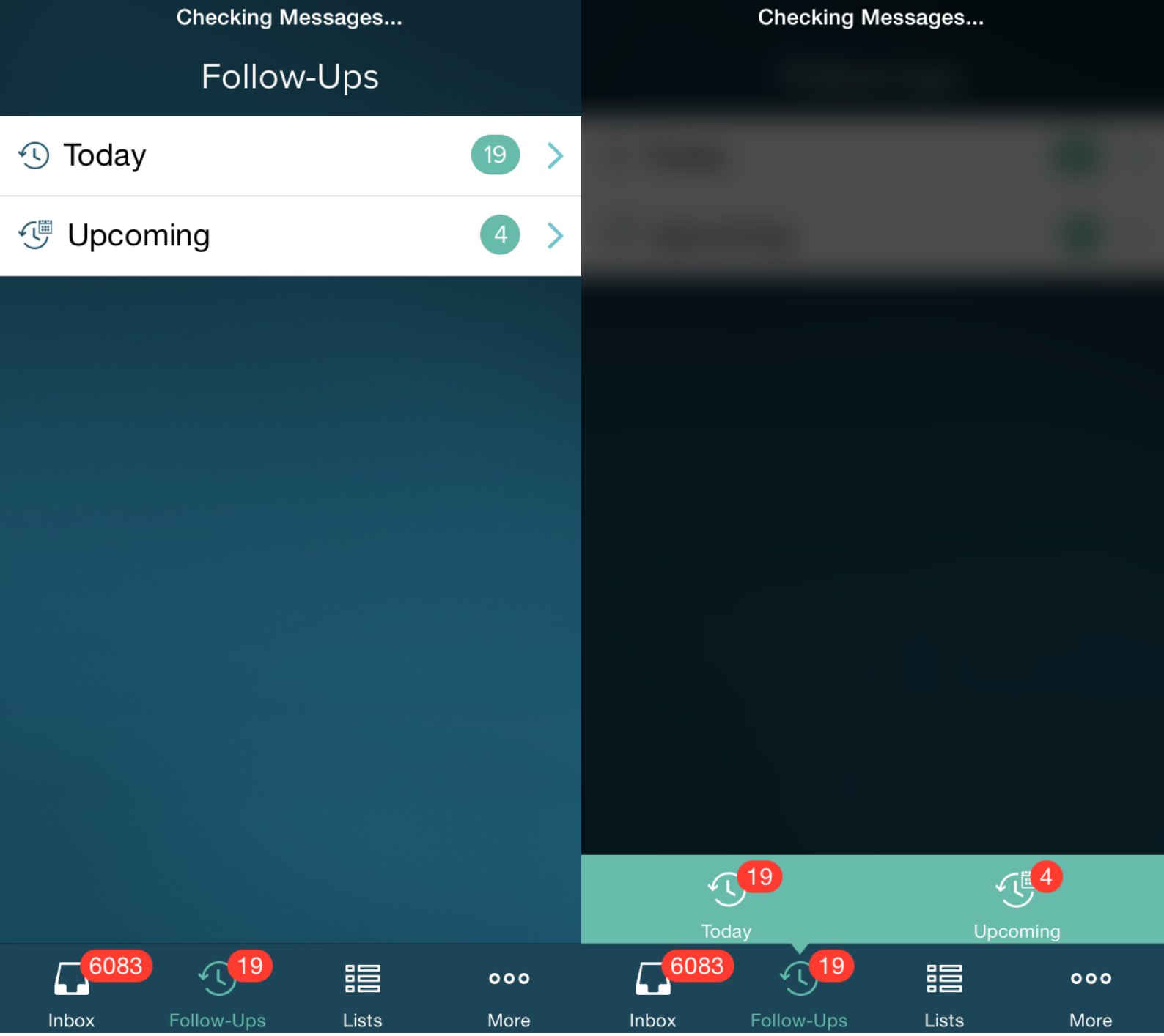
This walkthrough is also a great time to collect important information that goes beyond simple registrations, much like a setup wizard. The importance of this pattern cannot be stressed enough for any mobile application that isn't immediately intuitive because the more a user knows about your product, the more reasons they'll have to come back.

# Overflow Menus

Examples

Whatsapp, Gmail, RelateIQ






## Problem

Users want quick access to additional options or actions they can perform.

## Solution

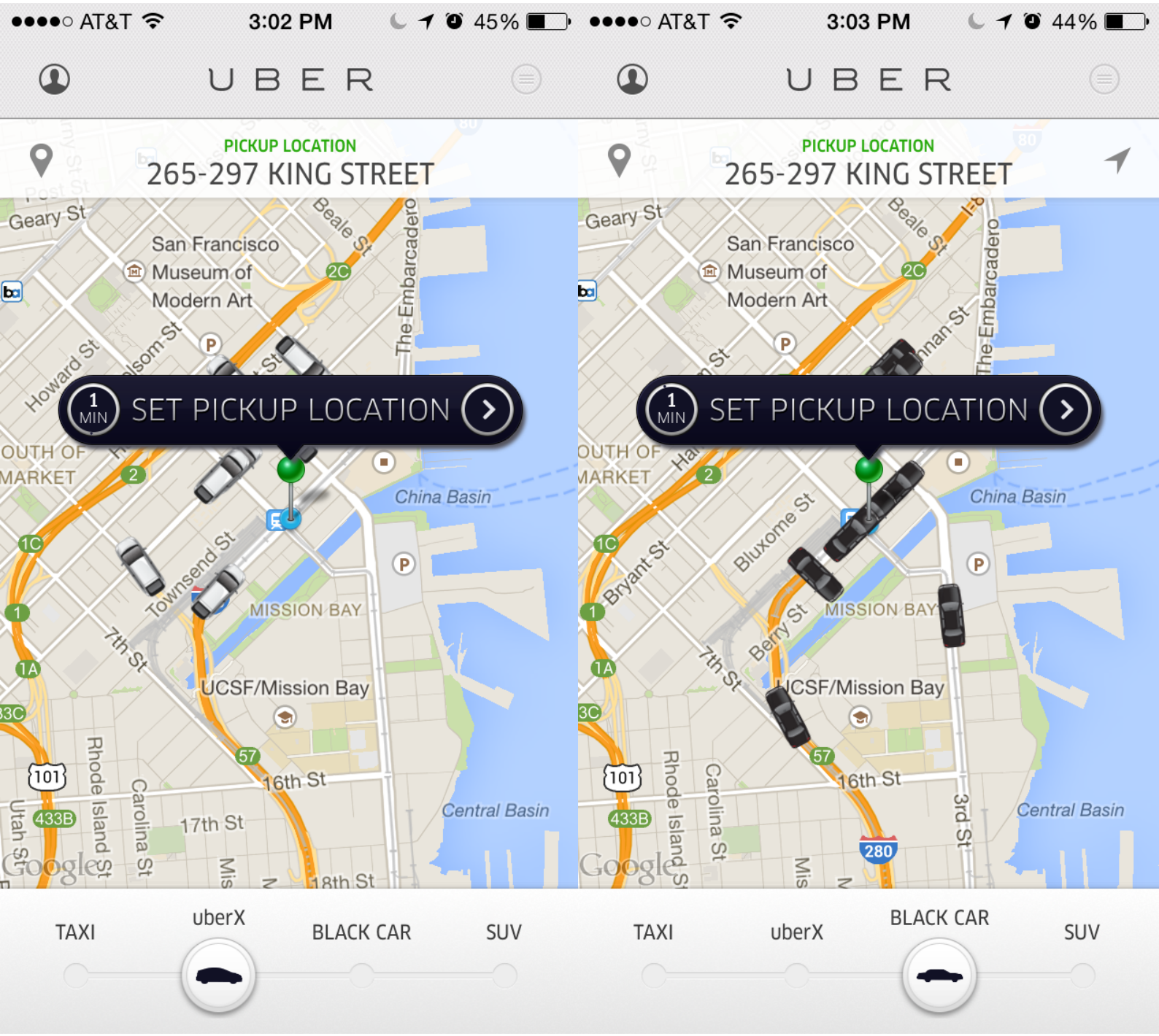
Hide extra options and buttons in an overflow menu so that they don't clutter the main interface. Overflow menus are extensively used in Android to stow away options and menu items in the action bar that aren't often used but are relevant to the current context. Apps like Whatsapp and Gmail use it for menu items like re-fresh and setting a status - added features of the app that user's should have quick access to, but would otherwise get in the way if put in more prominent positions.

RelateIQ let’s you hold down main menu items to see submenus for faster navigation to views.

# Sliders

Examples

Uber



## Problem

Users want to seamlessly move between options.

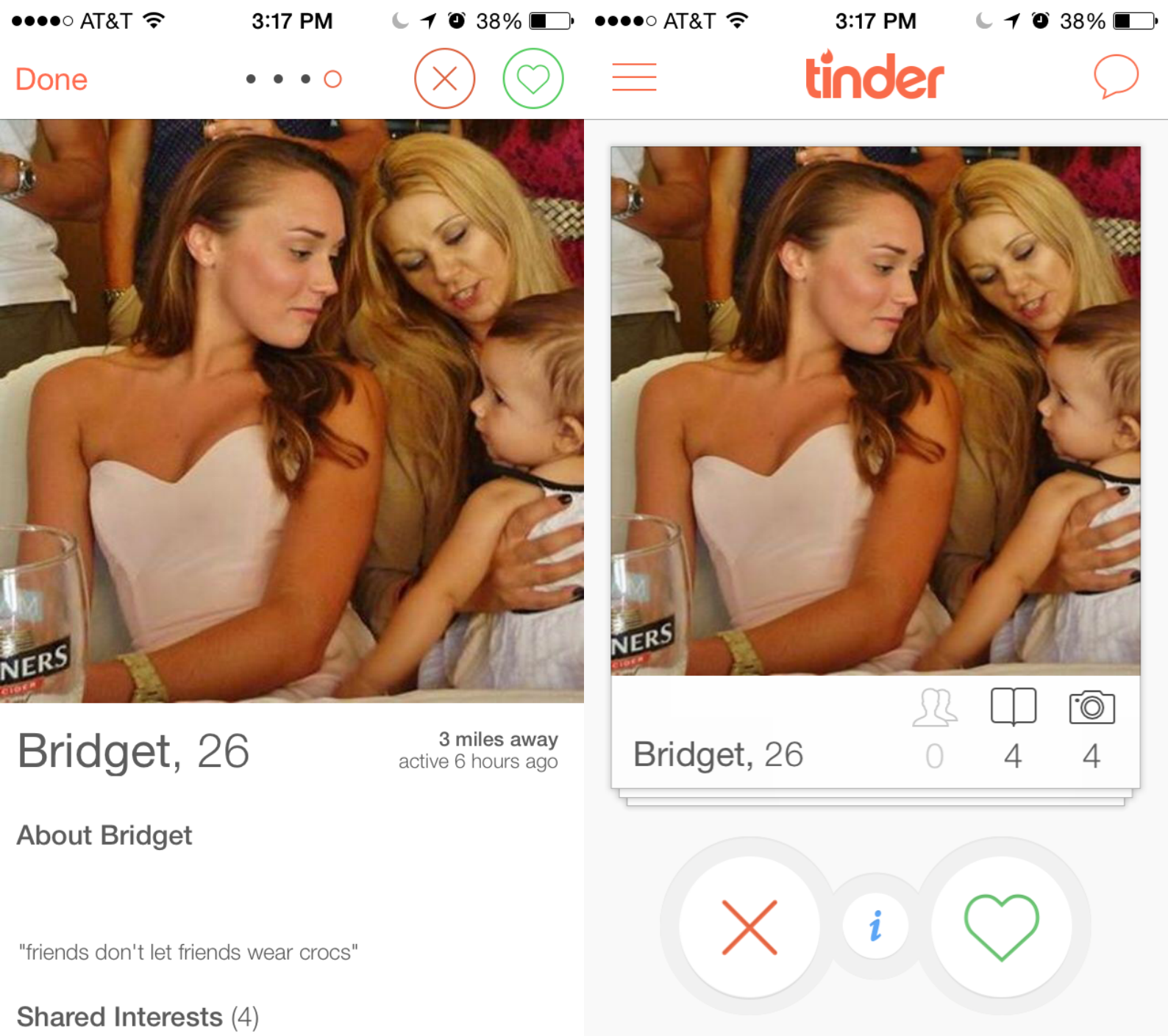
## Solution

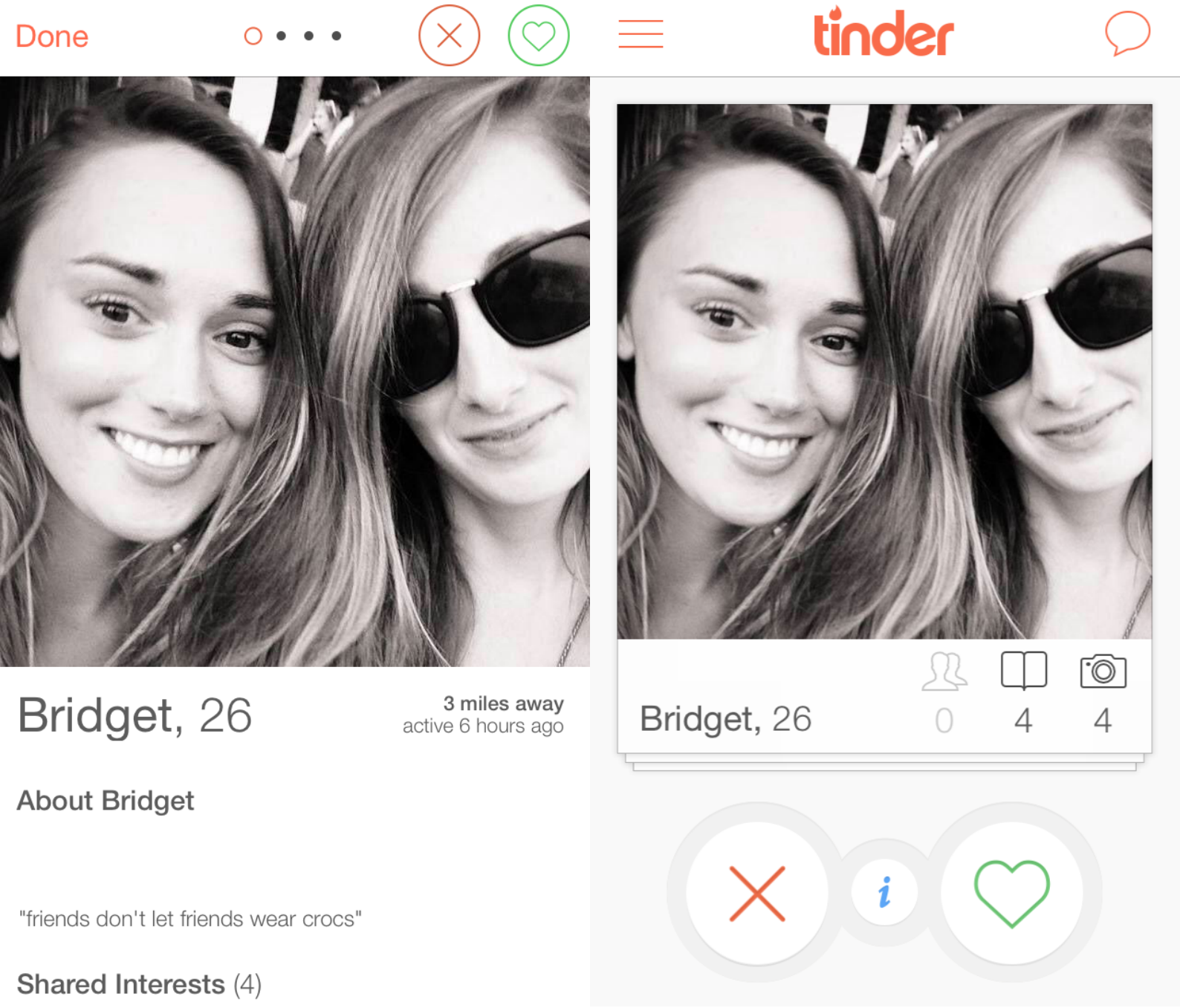
Make transitions between selections obvious and easy with the swipe of a finger. Uber lets you toggle between four types of ride services seamlessly by dragging a slider side-to-side. In this UI design pattern, they even zoom in and out to give you a similar level of density of cars nearby so you can see an acceptable number of options automatically.

# Content-Based Navigation

Examples

Tinder






## Problem

Users want to explore details of specific content easily and intuitively.

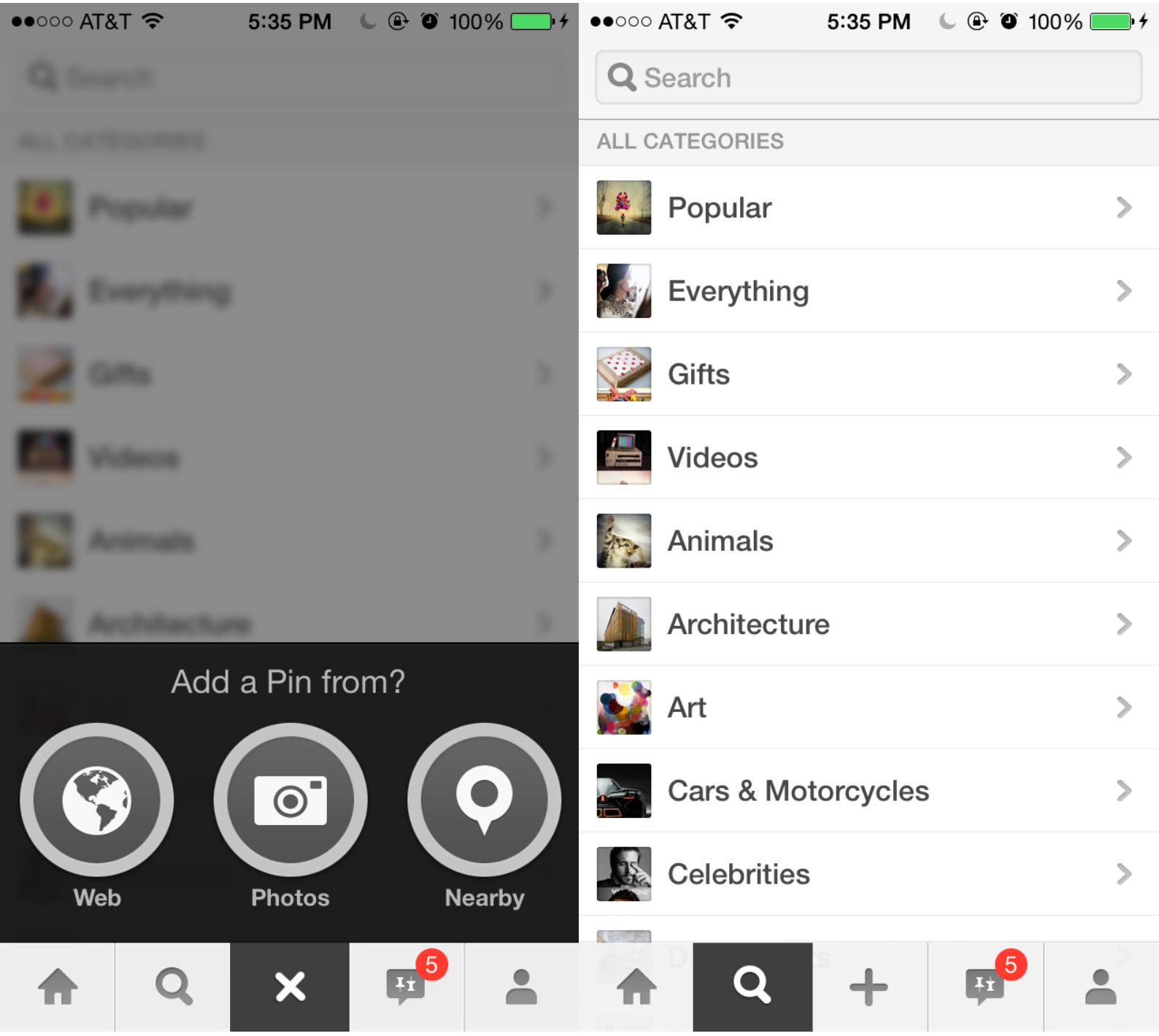
## Solution

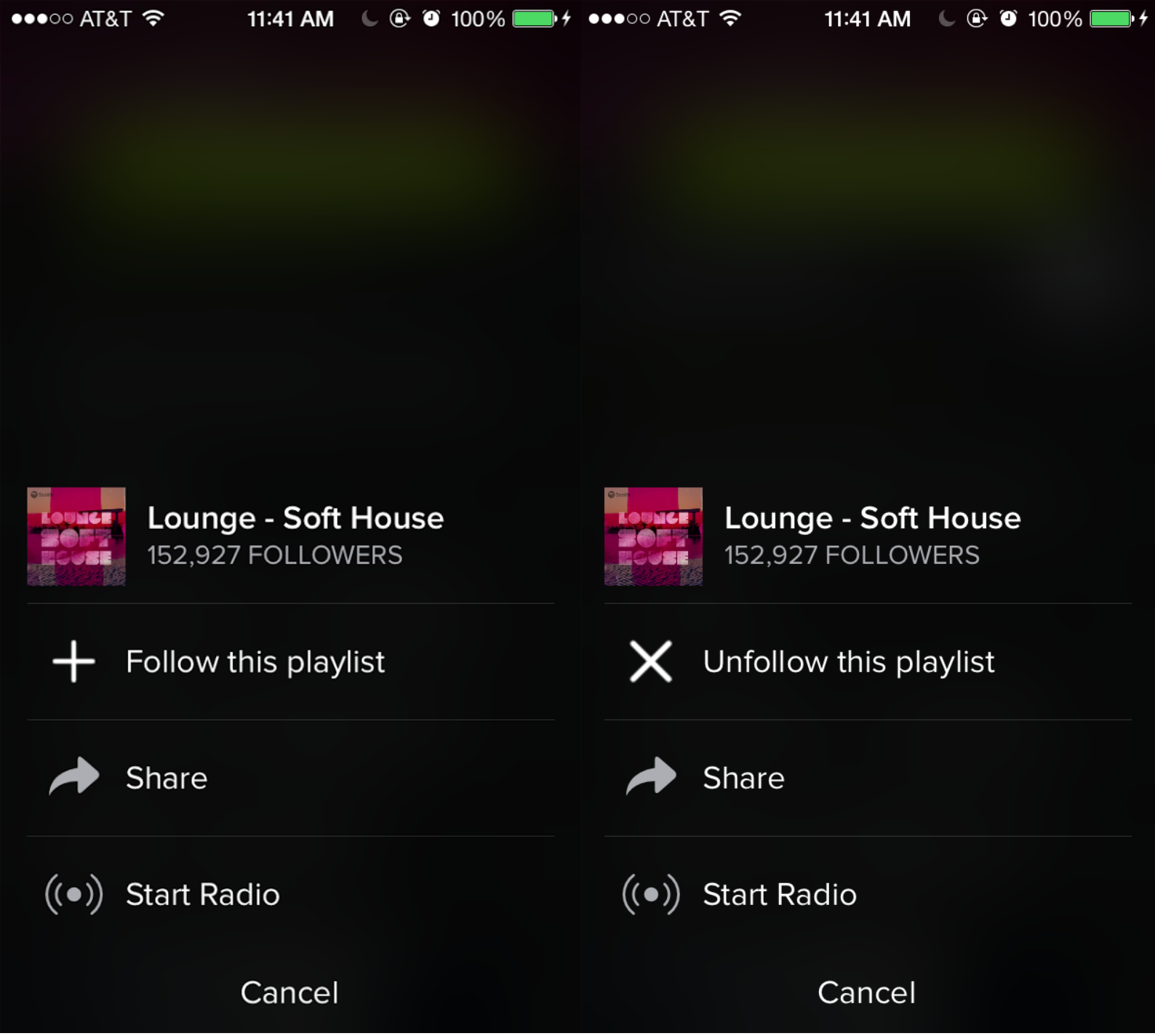
Make transitions between overview and detail states seamless. Tinder and 9Gag have made this seamlessly responsive. In Tinder, this UI design pattern lets you toggle between 2 states of a user's profile simply by clicking on the main picture in each view. But they go one step further. If you swipe through pictures in the detailed view of a user profile then click on the picture to go back to to the basic view, it stays on the picture you clicked on. This creates an extremely fluid and intuitive user experience and flow.

# Morphing Controls

Examples

Pinterest, Spotify






### Problem

The user wants to perform different types of actions, but there's limited screen real estate to show all these controls.

### Solution

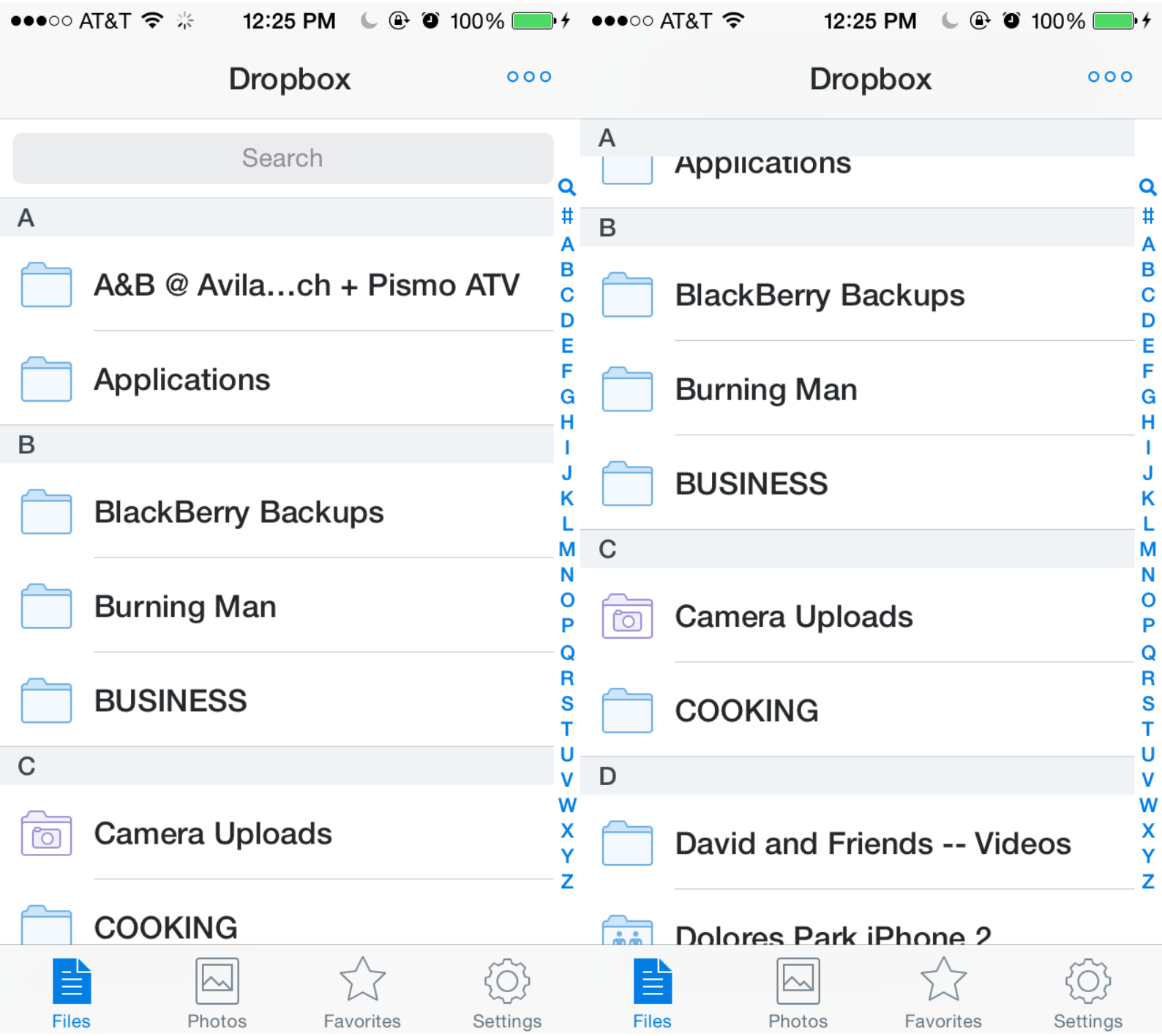
Replace buttons and on-screen controls with alternative functionality. Depending on what the user is currently doing, the UI could entirely replace an element with another, e.g. "do" and "undo" or "add" and "delete." This makes sense when the alternating actions are related in some way.

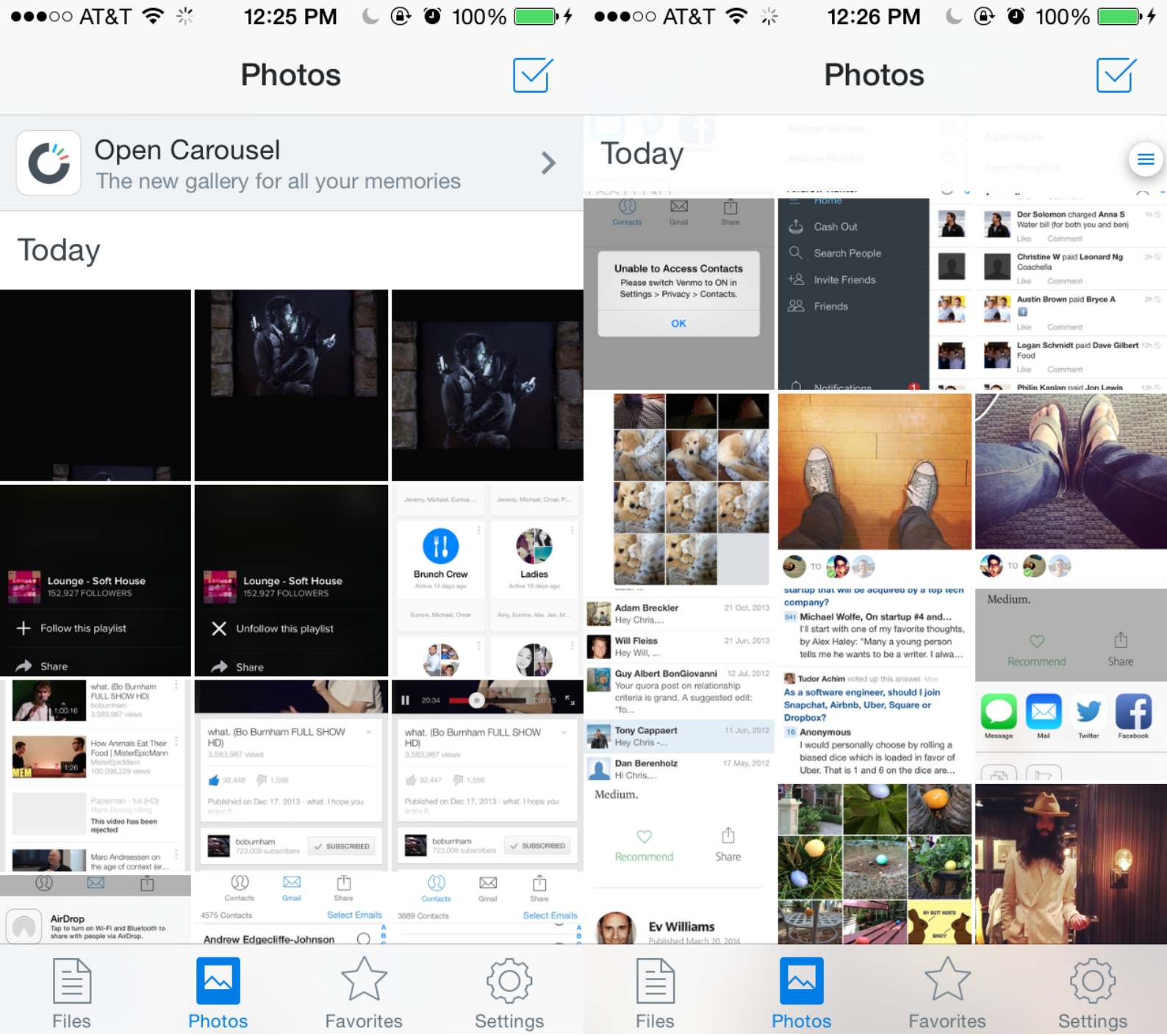
Pinterest and Spotify let you know you can cancel adding a pin or following an album, respectively, by transforming the “+” into an “x” button. This UI design pattern saves real estate, makes undoing any action quick and clean, and is an overall playful solution.

# “Sticky” Fixed Navigation

Example

Dropbox






## Problem

The user wants to have access to the menus anytime while in the application.

## Solution

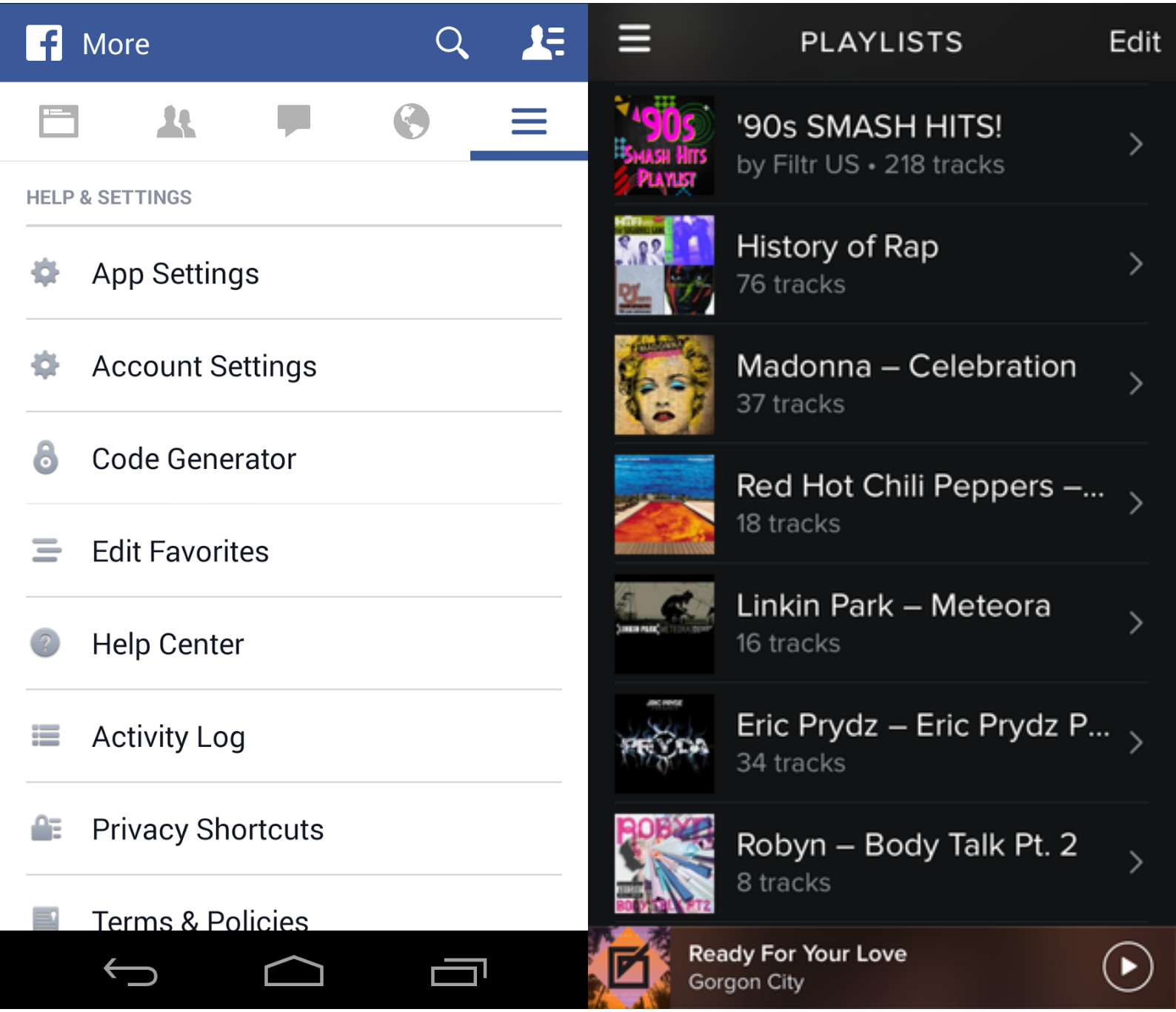
The top, side, or bottom navigation stays in place while a page is scrolled. In some cases, headings from sub-sections may also become fixed while scrolling and replace or be appended to the existing fixed navigation. Address books are a great example where each alphabetical section (“a”, “b”, “c”, etc.) stick below the top navigation when you scroll past that section header. Photo galleries and file folders tend to utilize this same design pattern. In other cases, menus disappear when scrolling

in one direction and become fixed when you scroll in the other direction. Pinterest is a great example of that where the menu disappears at the bottom when scrolling up and appears when scrolling down. This is different from an action bar (a pattern that's used heavily in Android) to store commonly used app functions.

# Vertical Navigation

Examples

Facebook, Spotify



### Problem

The user needs a way to navigate between different sections of the app, but there's limited space to show this information.

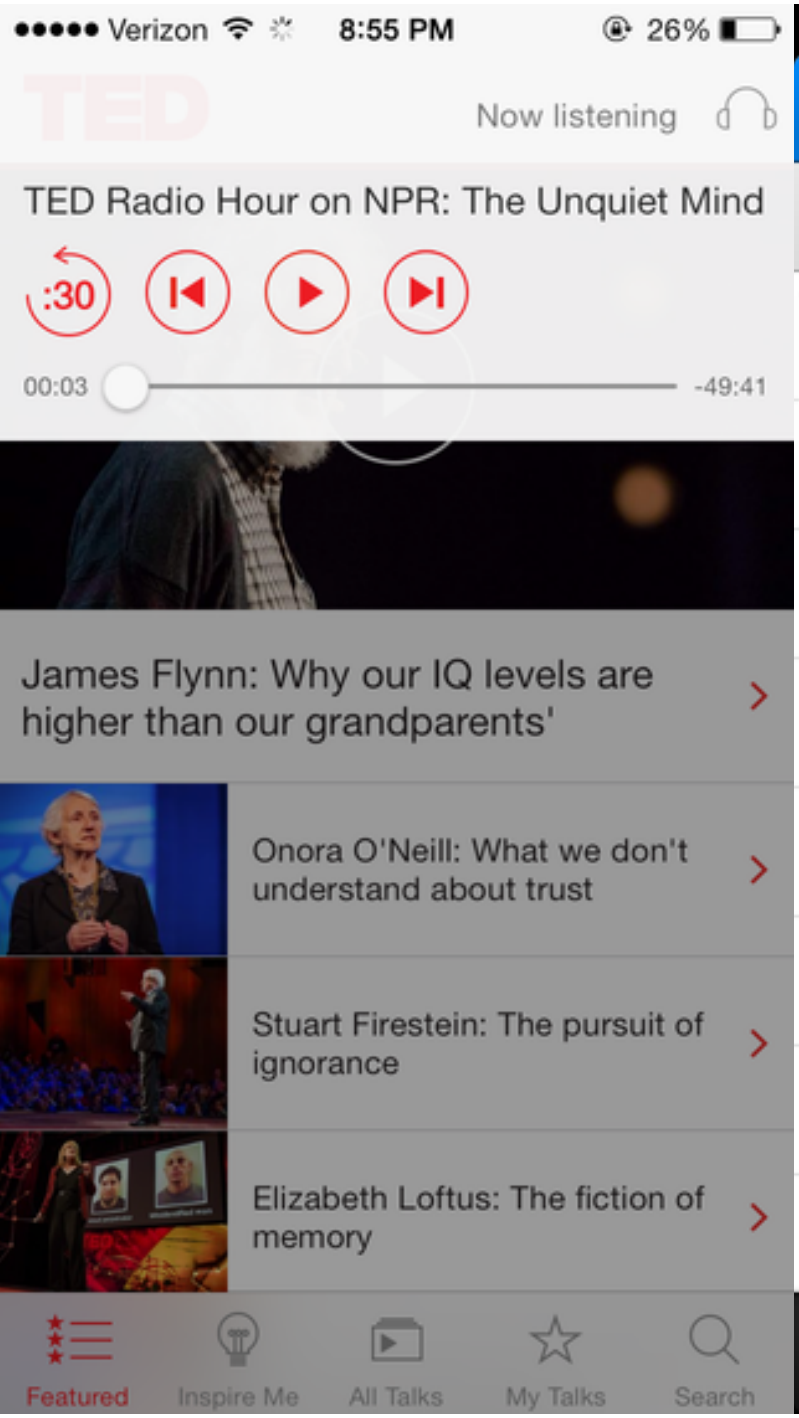
### Solution

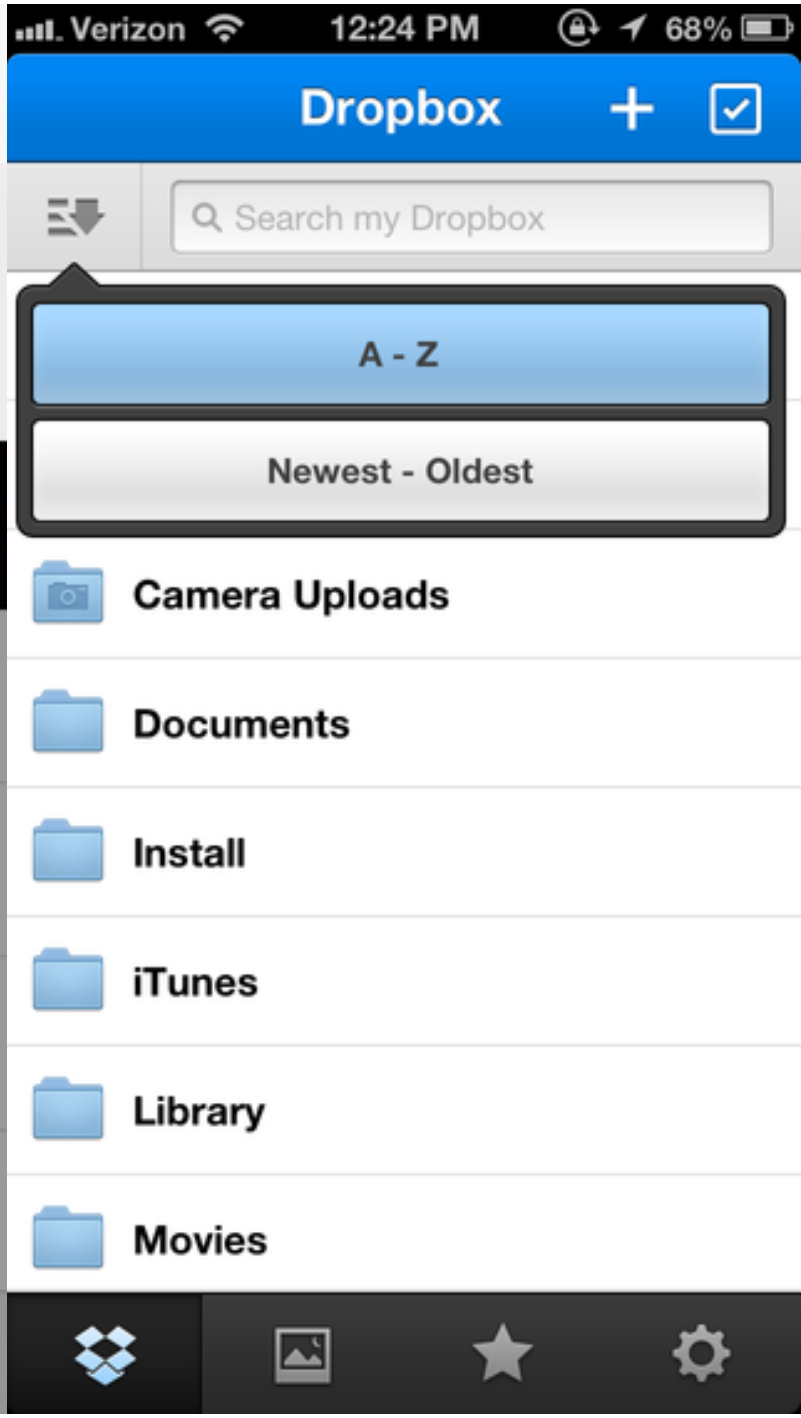
Important sections of the UI are presented in a list, which the user can scroll through to get what they want. Scrolling in this way is a standard mobile gesture, so it makes sense for apps to adopt this for their navigation layout. This also leaves the header and footer of the UI free for more "universal" navigation, such as action bars. You'll see varying implementations of vertical navigation across all kinds of apps ranging from music players like Spotify to news readers like Yahoo! Digest.

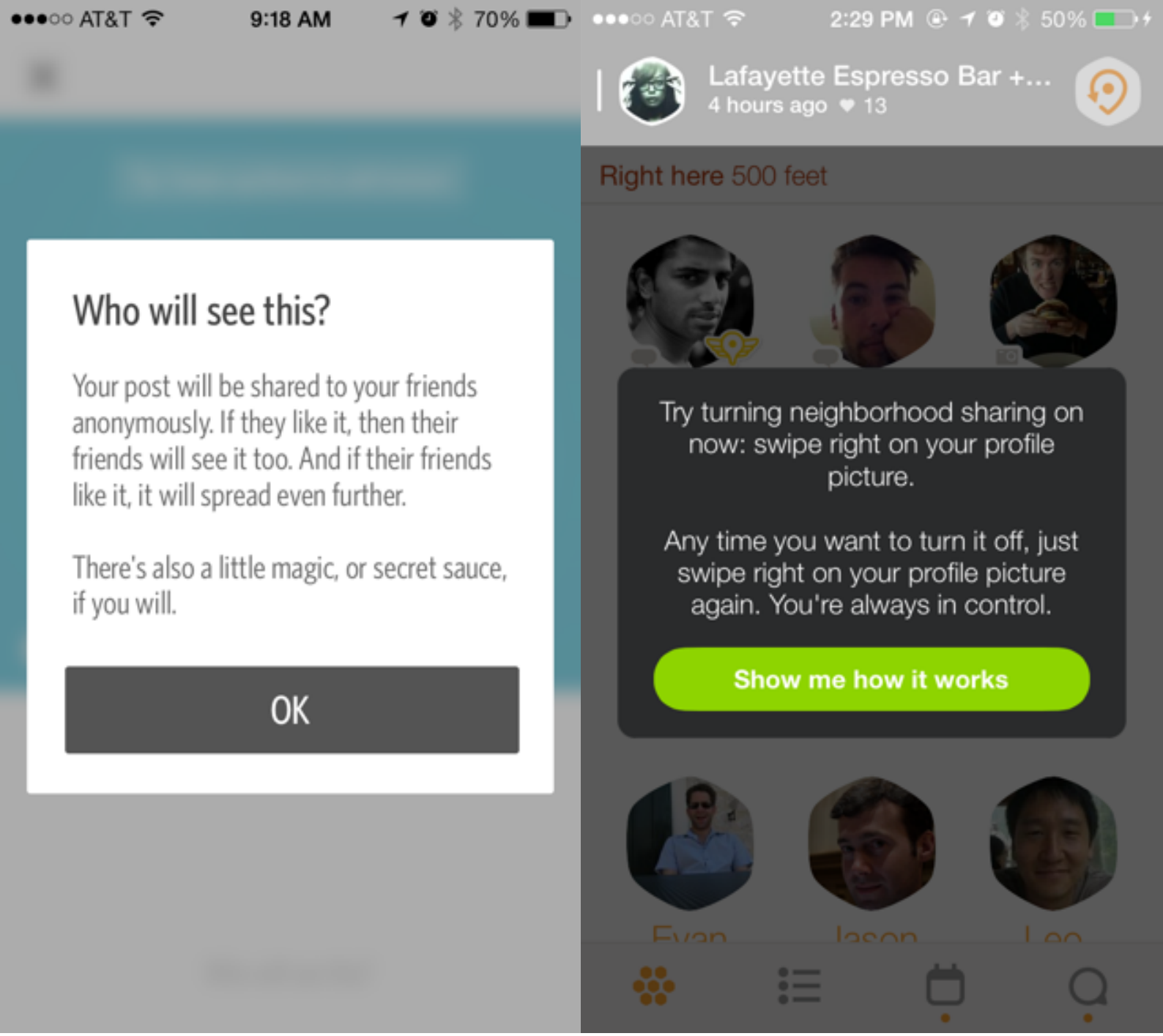
# Popovers

Examples

TED, Dropbox, Secret, Swarm






## Problem

The user wants to view relevant information without losing their current place in the UI.

## Solution

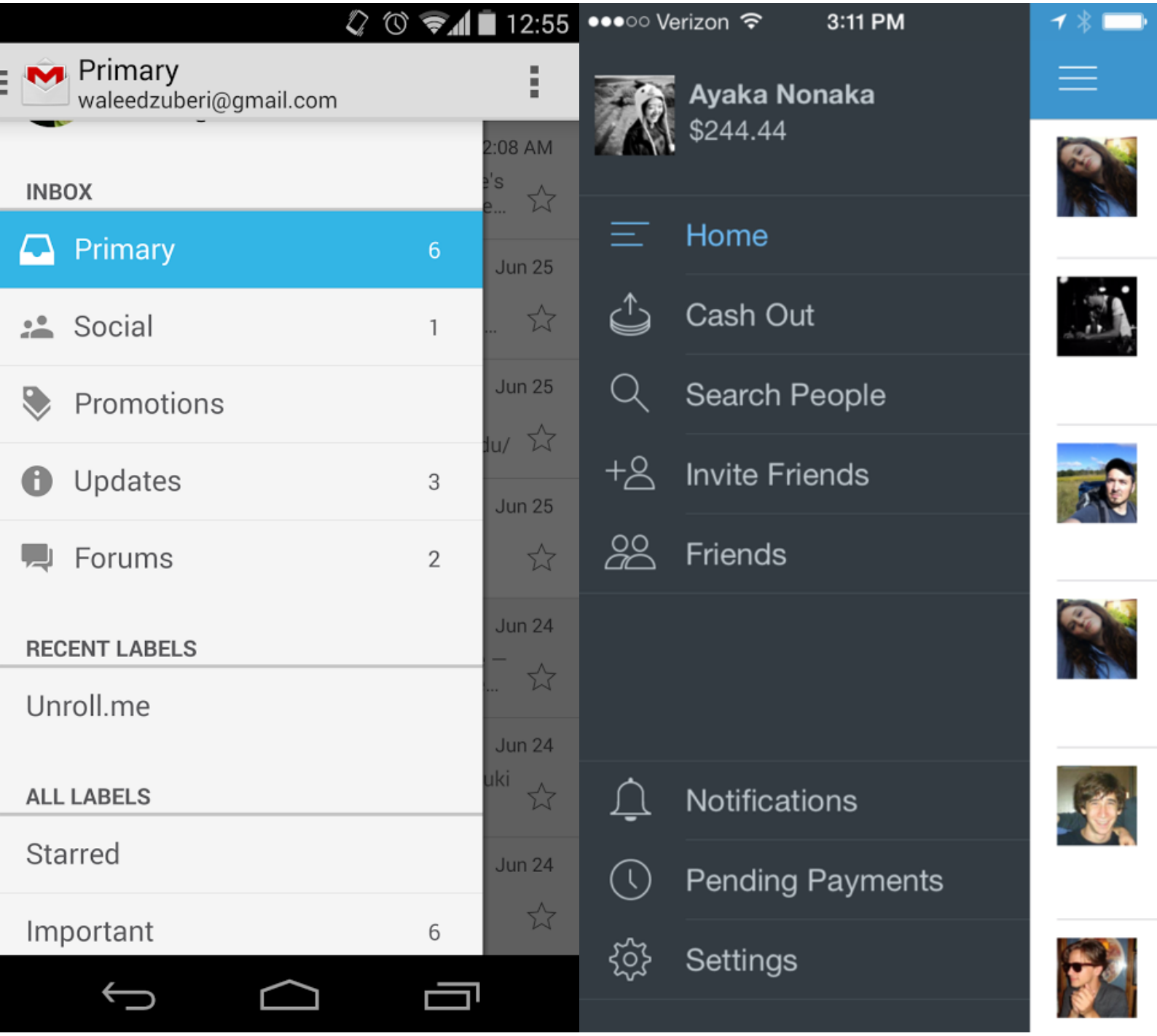
Show important notifications and additional information in popovers. This UI pattern has the advantage of providing a lightweight and straightforward way of viewing additional information or taking a particular action, but they do so without pulling the user out of their current activity. The official TED app puts playback control in a popover with the background faded out so that the user is always aware that

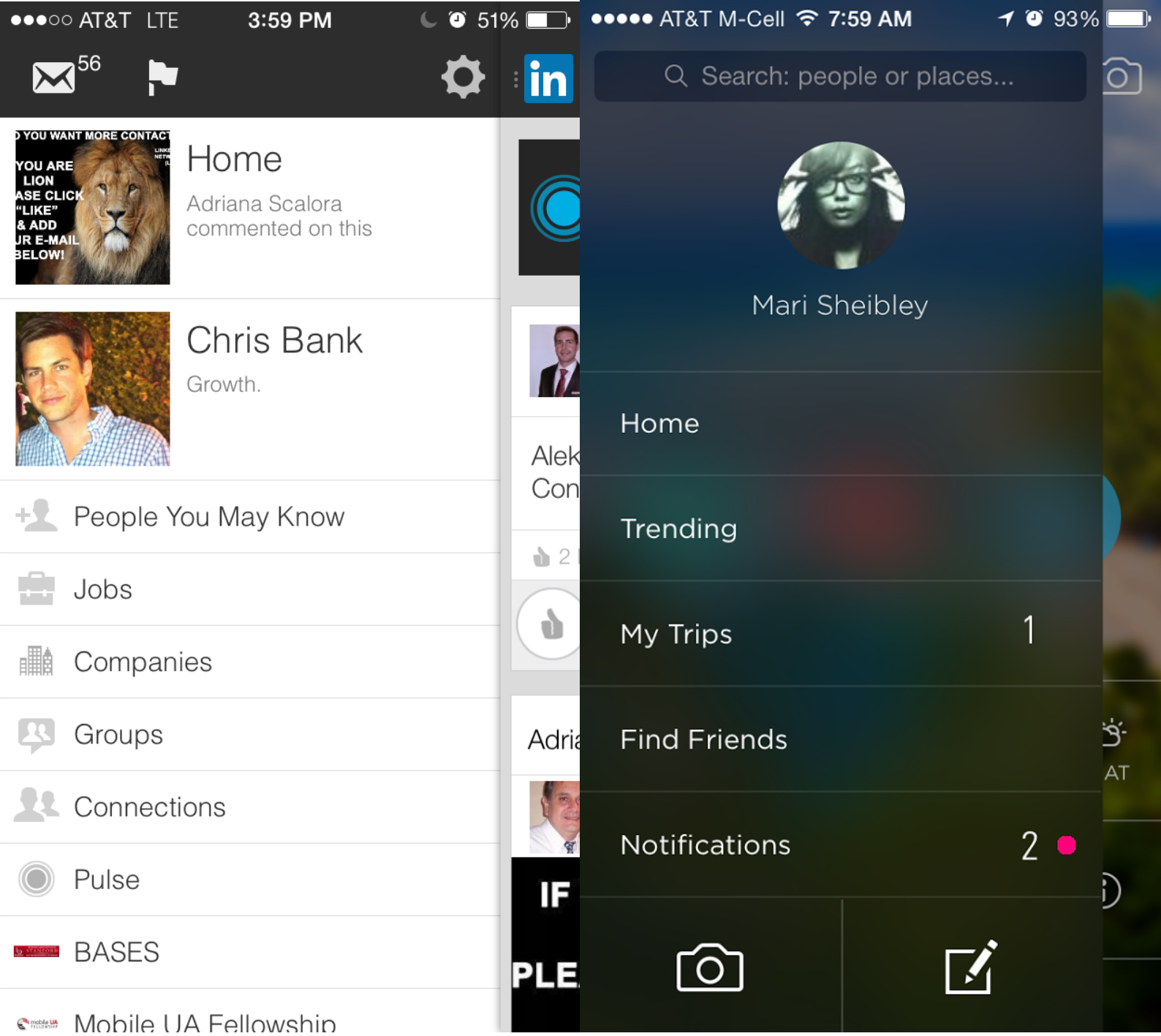
this quick interaction with the player is not going to interfere with their browsing through other content. Dropbox and Kindle also place controls in a popover. The popover UI pattern is important for actions like these because they are being performed on the data and this way users always know what these controls apply to. With the content still visible in the background, the user can tweak sorting options or change the font size without having to go back and forth between the views - it all happens right there. Popovers and modal windows can also be used to display important notifications or notices where it's essential to get the user's attention because dismissing them requires a tap or swipe. For example Secret and Swarm use a popover to explain what will happen if the user continues with their action.

# Slideouts, Sidebars & Drawers

Examples

Gmail, Venmo, LinkedIn, Gogobot






## Problem

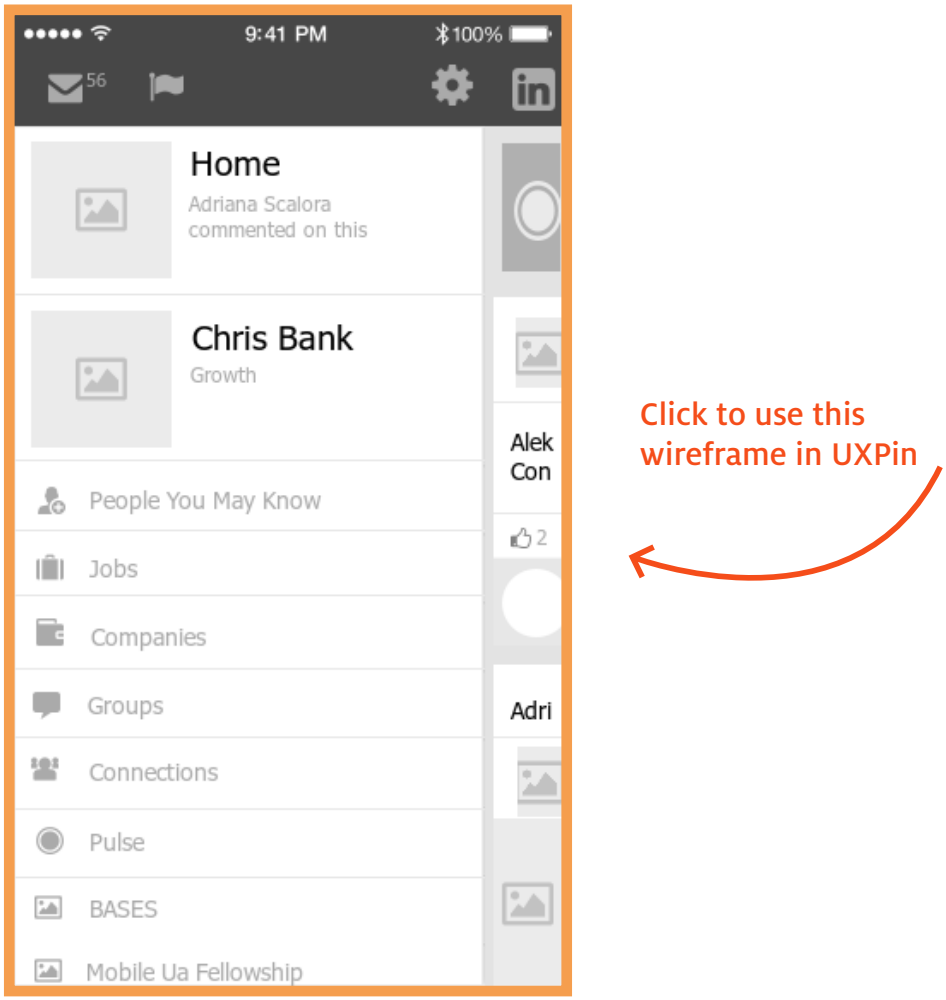
The user needs a way to navigate between different sections of the app without being distracted in each individual section.

## Solution

A secondary section of the application - such as navigation, chat, settings, user profiles, etc. - is tucked away in a collapsible panel hidden under the main section when it is not needed. When accessed, it usually either moves the main section aside or slides over it. Since the slideout is in a separate layer from the main content in the application, there's a lot of flexibility in terms of how content can be laid out

inside the drawer - icons, text, and even simple controls are viable options to provide quick access to important actions here. Often times, the drawer can be hidden under a “hamburger menu” or a simple arrow that indicates there’s more content there. It’s an easy way to hide all the less important things in a “side drawer” and not worry about how a mobile application should distill the most important information. Instead, you only have to focus on how to distill the most important information in each view that’s accessible from the side drawer.

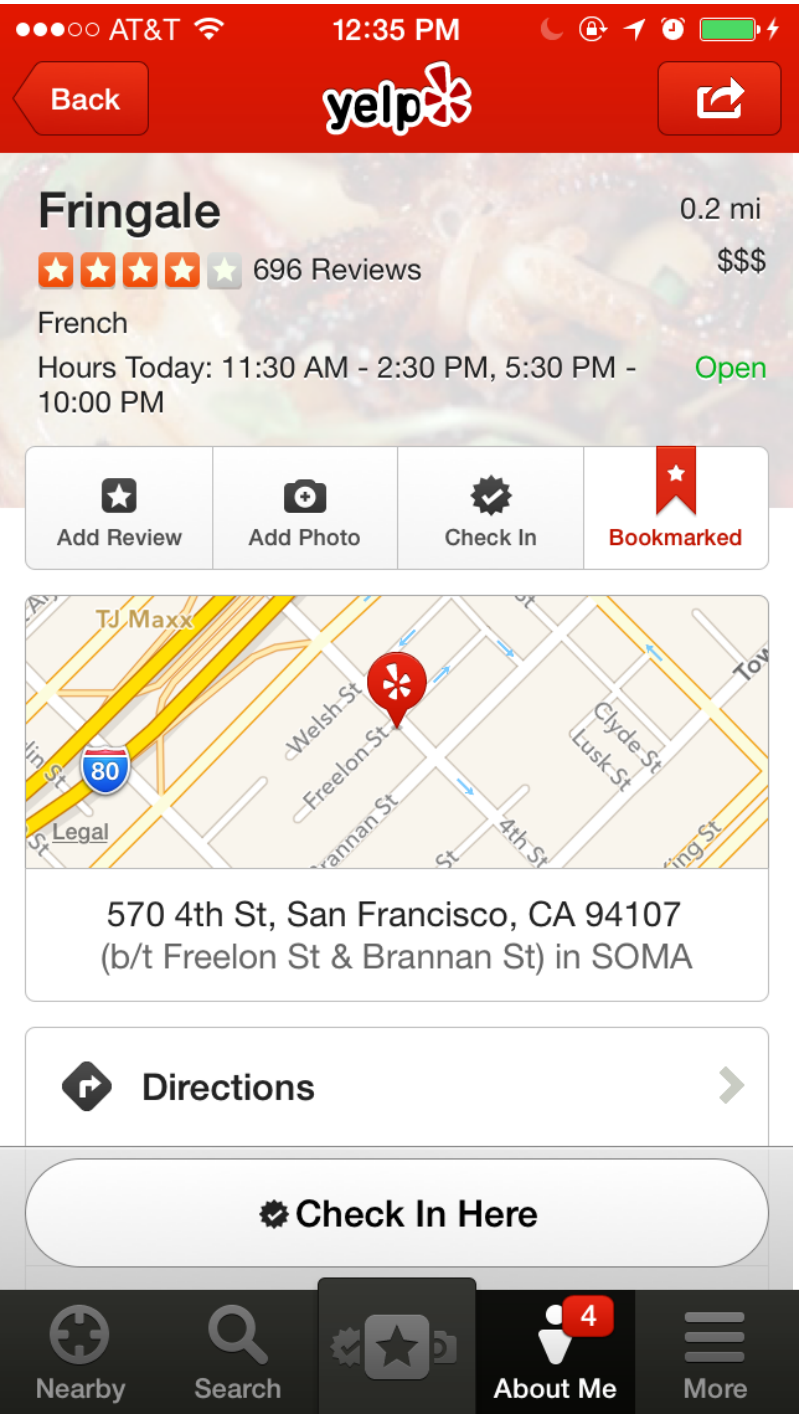
We’ve included a wireframe example of this design pattern below using UXPin.

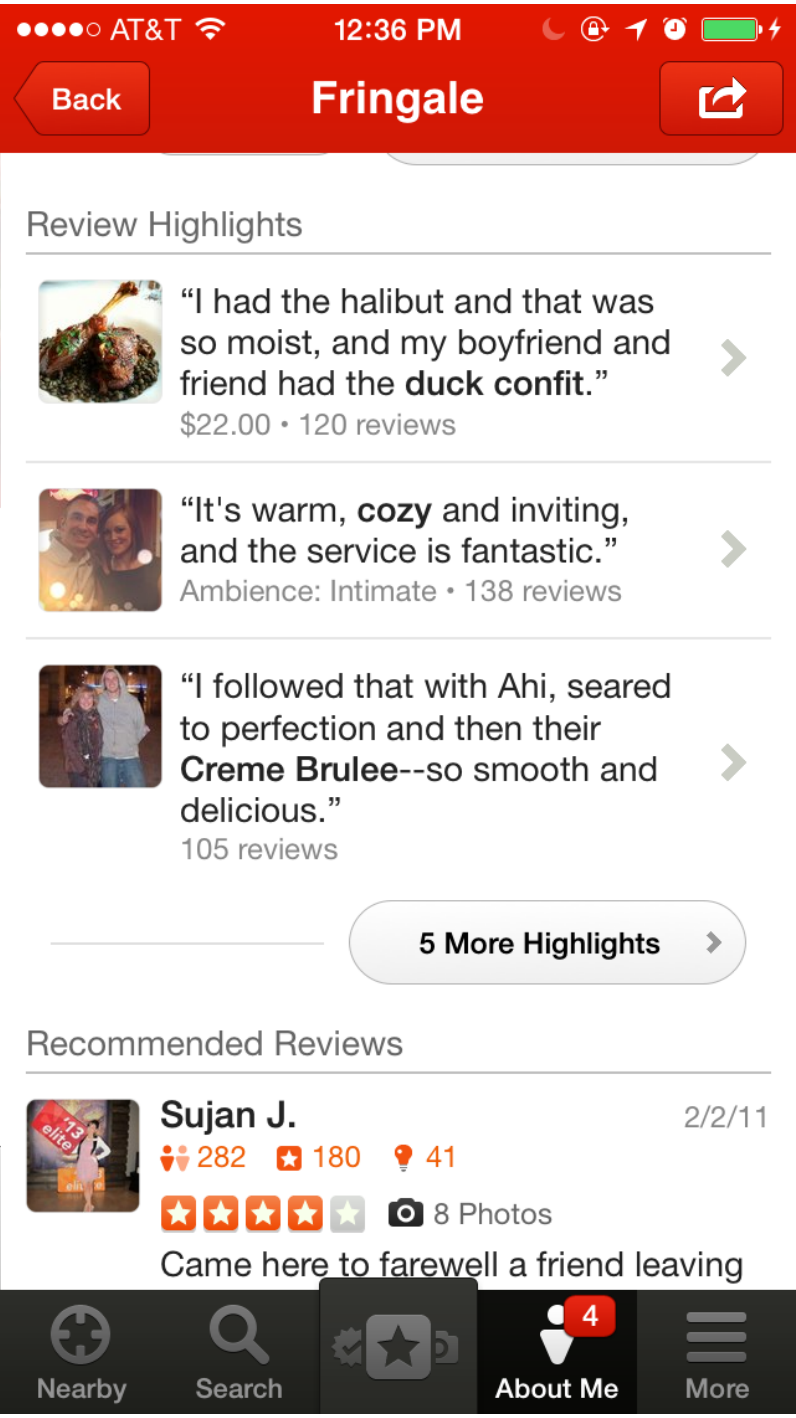


# Links to Everything

Example

Yelp



## Problem

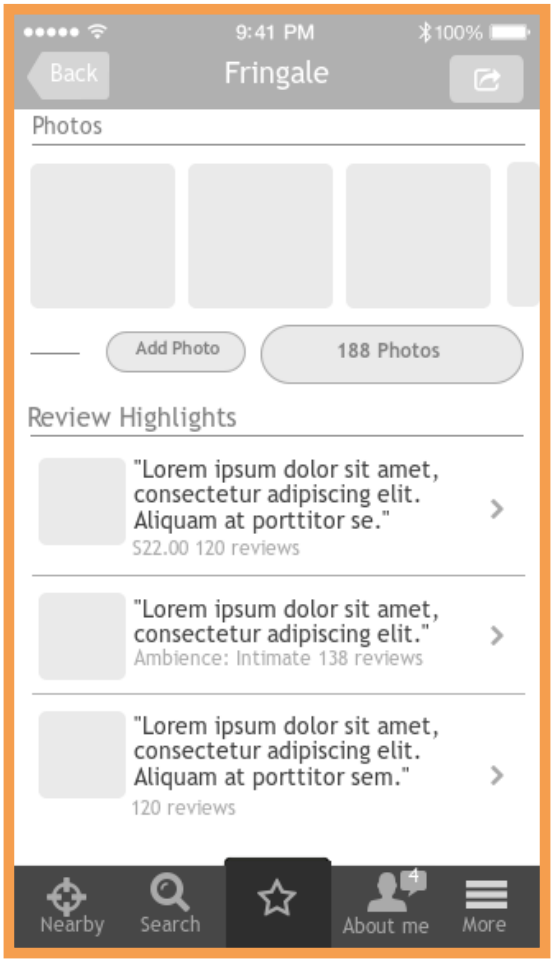
The user needs a consistent way of navigating through content without being distracted by additional content.

## Solution

Most or all user content within the app is linked, giving users the freedom to explore and find the exact information they're looking for without hitting dead-ends or being distracted by a litany of hyperlinked text, additional buttons, calls to action, etc. that you would normally see on a website. If they want to interact with a piece of content in the app, odds are that they can tap on it and go to a new view for a more detailed experience. For example, in Yelp, users have lots of options - they can tap on the buttons across the bottom or instead explore by tapping on the content itself, like maps or comments. This makes for an easier navigational pattern than, say, Flipboard, with endless ways you can swipe, tap, x-out, undo, and go-back as you navigate through it's digital magazine.

We've included a wireframe example of this design pattern below using UXPin.

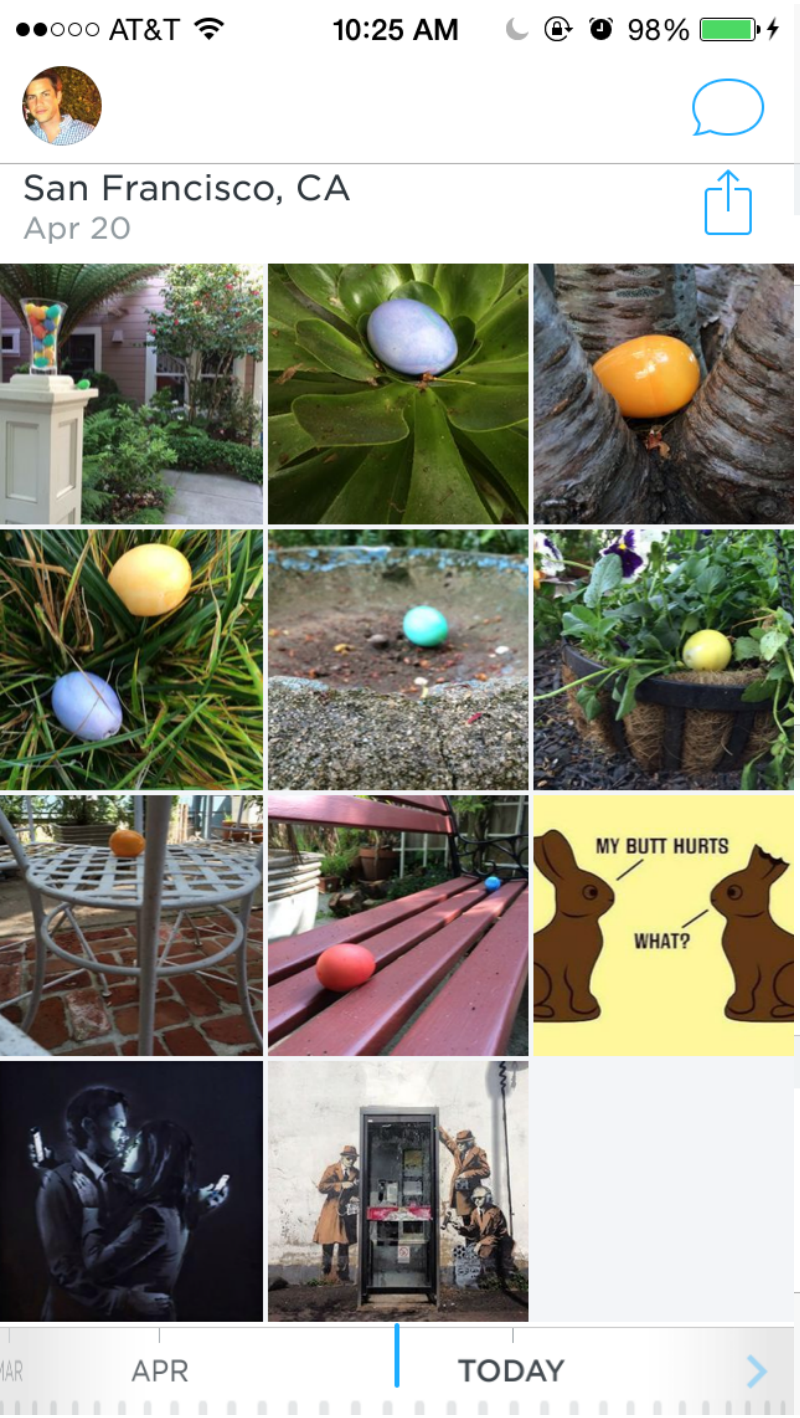
Click to use this wireframe in UXPin

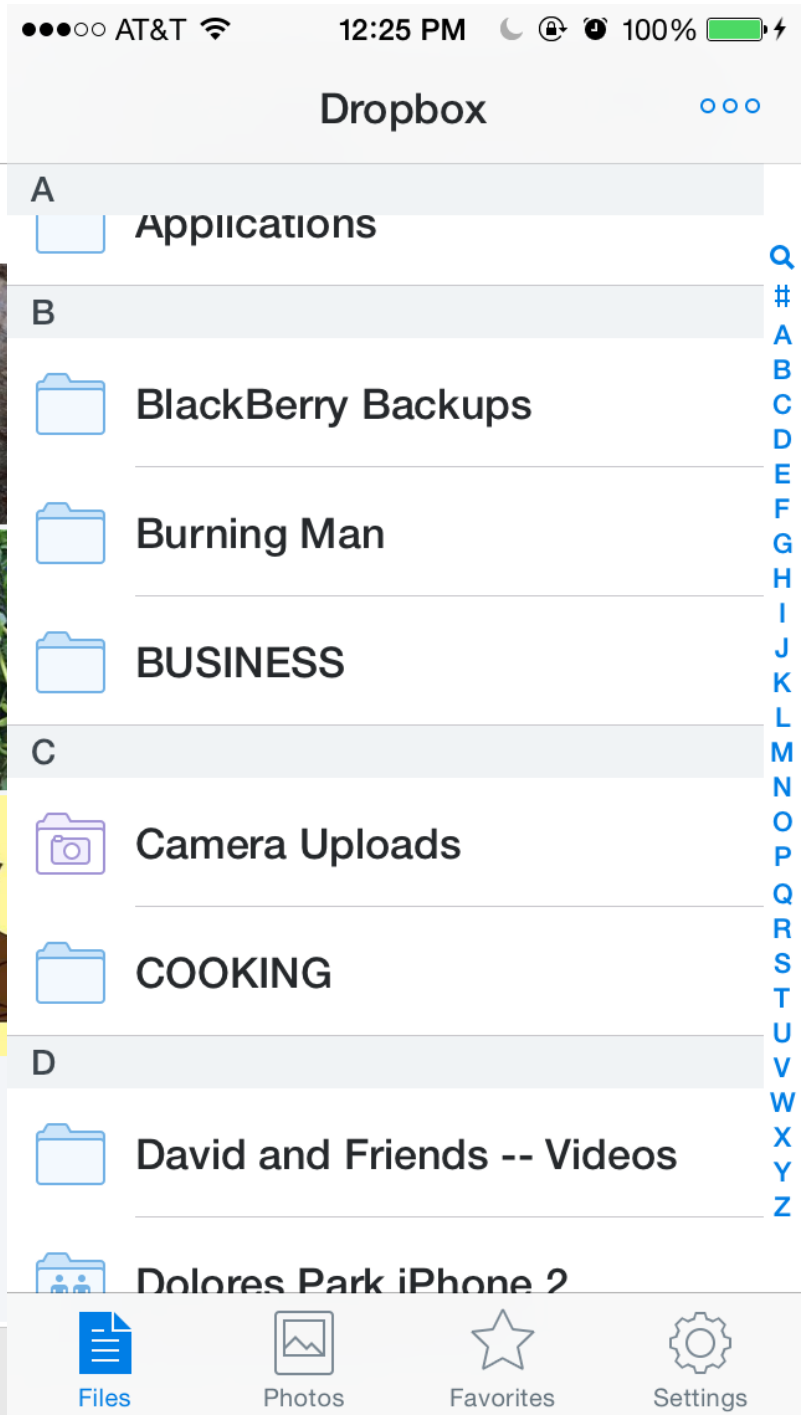


# Advanced Scrollbars

Examples

Carousel, Dropbox



## Problem

The user needs to see their current position in the context of an entire content set, or move to a specific section of a long list or gallery more quickly.

## Solution

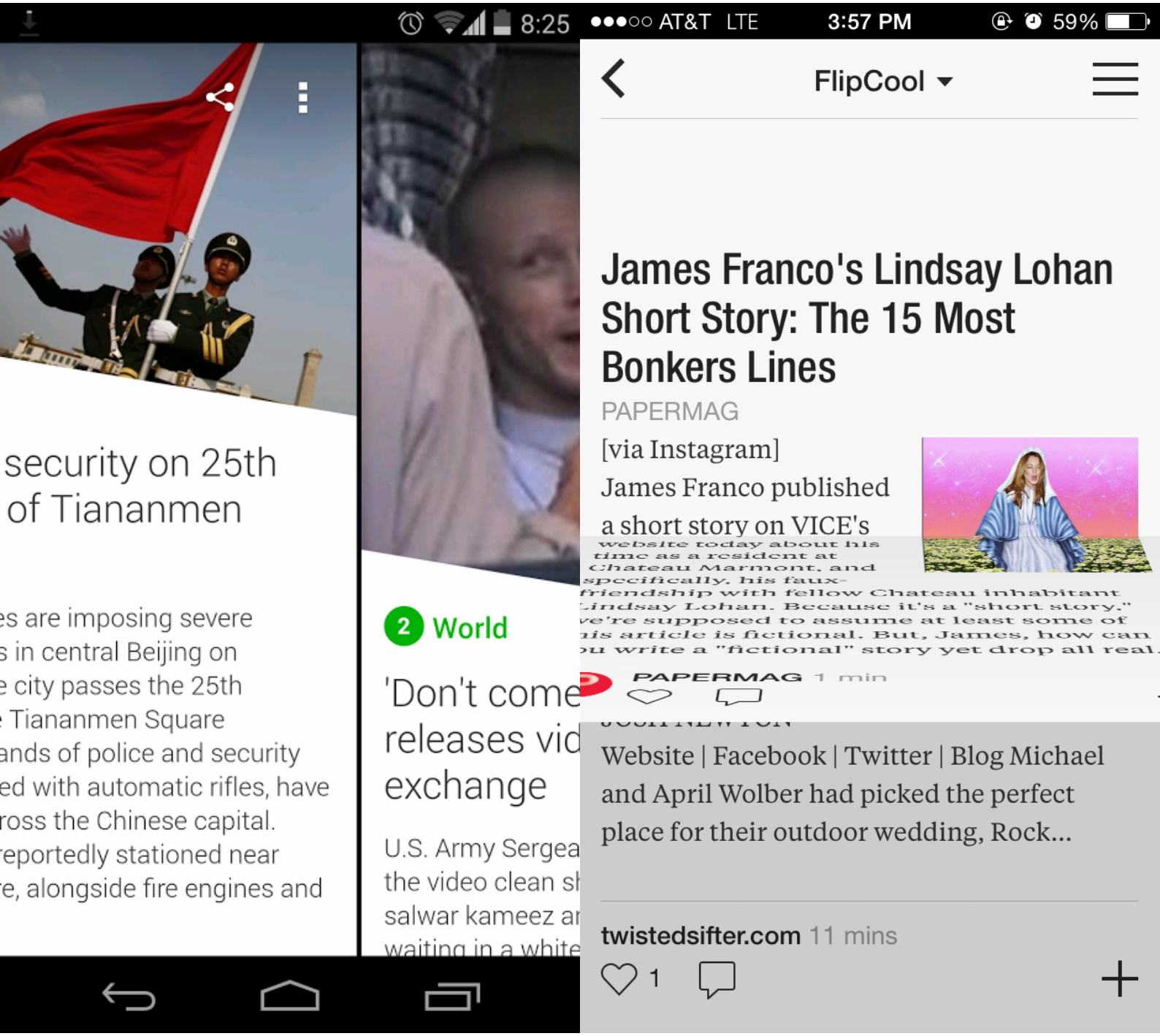
Beyond scrolling with a swipe gesture, mobile lists and galleries often have a scroll bar that is persistent or temporarily appears when scrolling. Often, the scroll bar is complemented by a scrolling index - dates, alphabetical letters, categories, locations, etc. With index scrolling, a scroll indicator is typically persistent so it appears even when the user isn't scrolling. Touching or dragging it causes the current section to pop up in a prominent way. However, the scroll bar and index can be coupled and only appear during scrolling to save extra screen real estate and reduce distractions. In cases where scrolling and indexing are even more critical, the scroll bar is more likely to be more prominent and persistent.

For example, in Carousel, we not only have a visible scrollbar, but a power scrollbar at the bottom so you can blaze through your 1 million hosted photos with ease. As user-generated content, feeds, groups, lists, etc. keep growing, we'll see even more innovative UI design patterns that allow users to find what they're looking for beyond search and scroll bars.

# Swipe Views

Examples

Yahoo! Digest, Flipboard



### Problem

The user wants to navigate from one piece of content to the next without having to go back to the index.

### Solution

Allow users to move from item to item by swiping through content. This pattern should be familiar from browsing through photo albums, but more and more apps are starting to implement this for their content as well, like Yahoo! Digest and Flipboard. This helps maintain the immersive experience for users when they can simply swipe through content. This pattern can also be used for an organizational purpose, separating different section of the app into “tabs” that the user can access by swiping. When implementing this pattern it’s also a good idea to think about how well the interface shows its ability to be swiped.

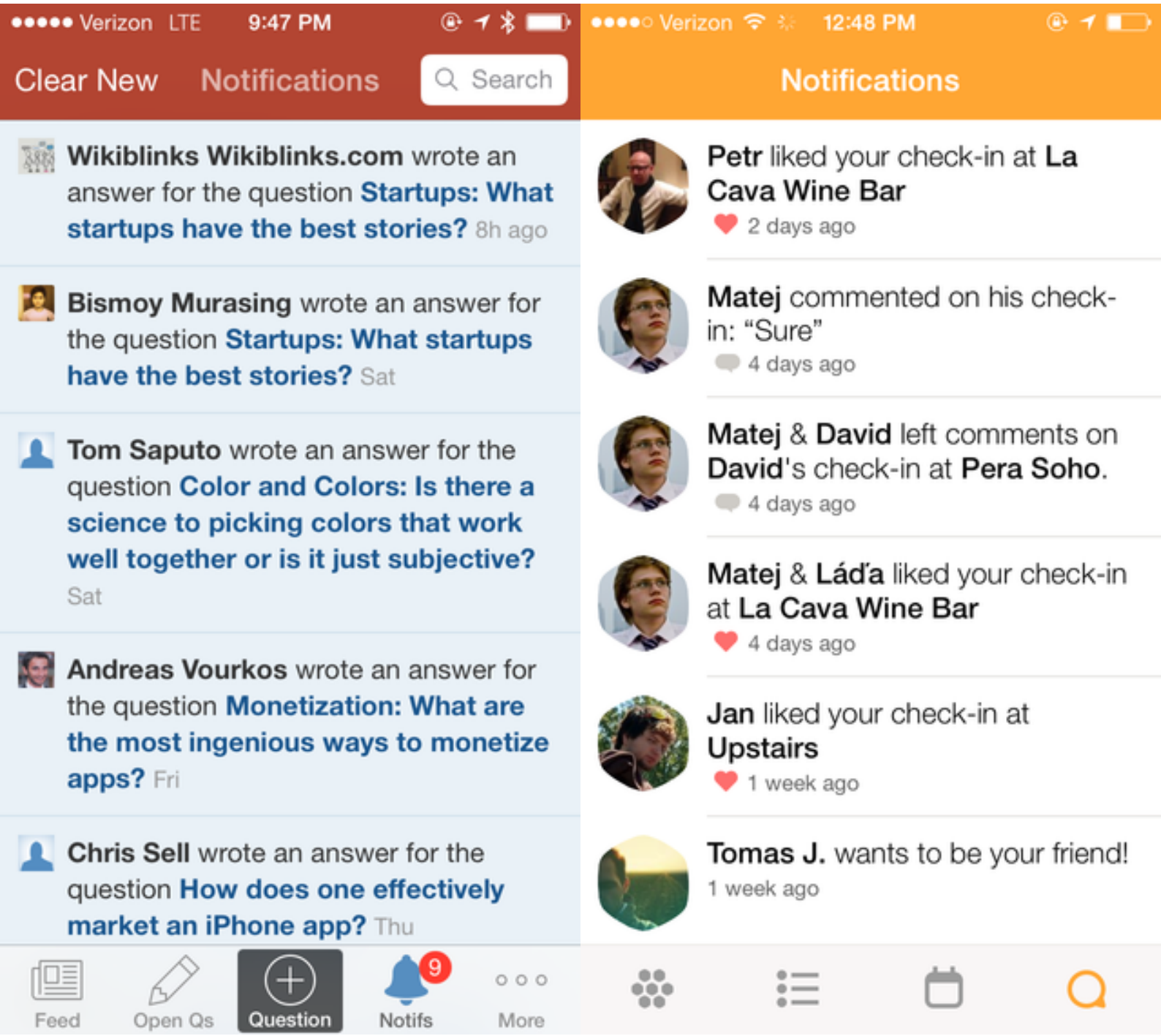
# 4

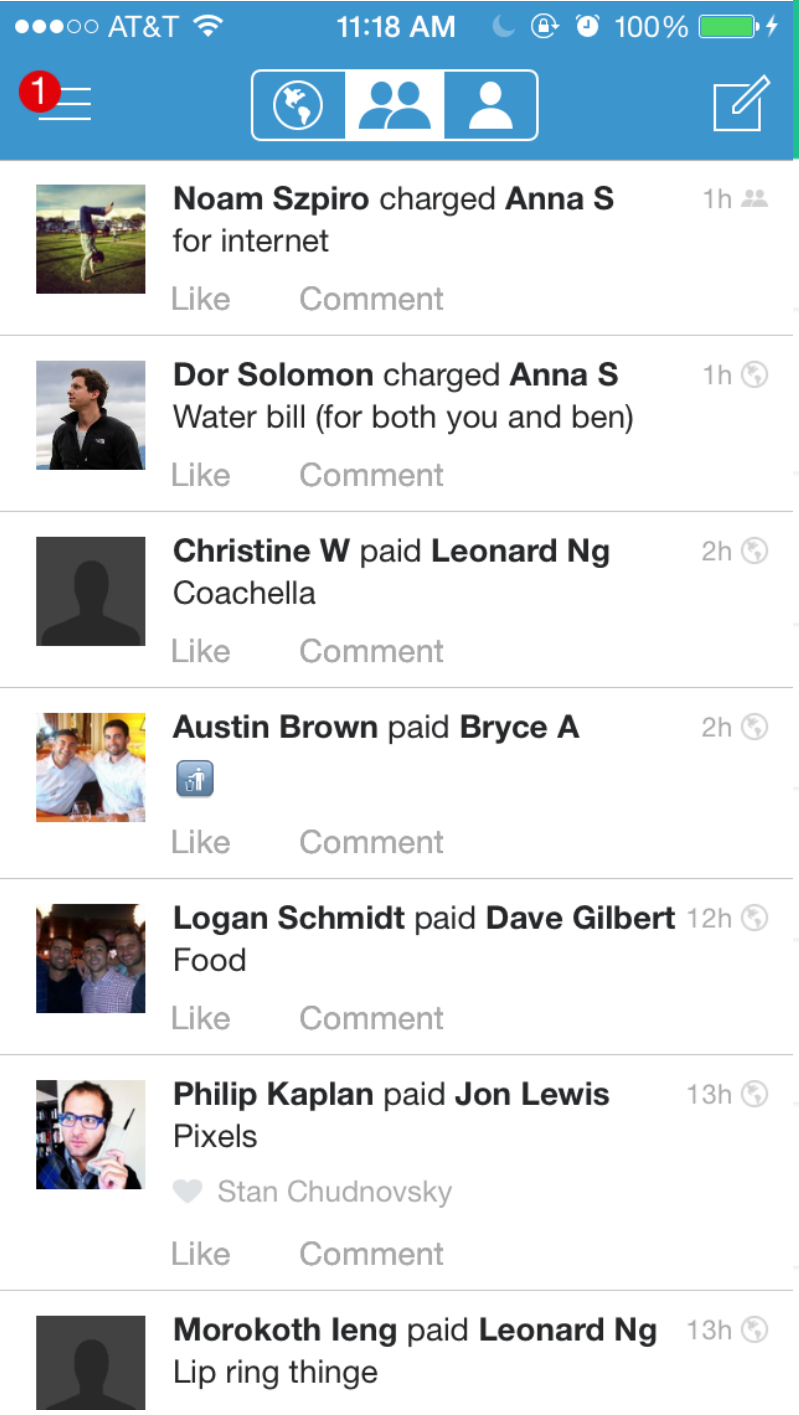
# Social

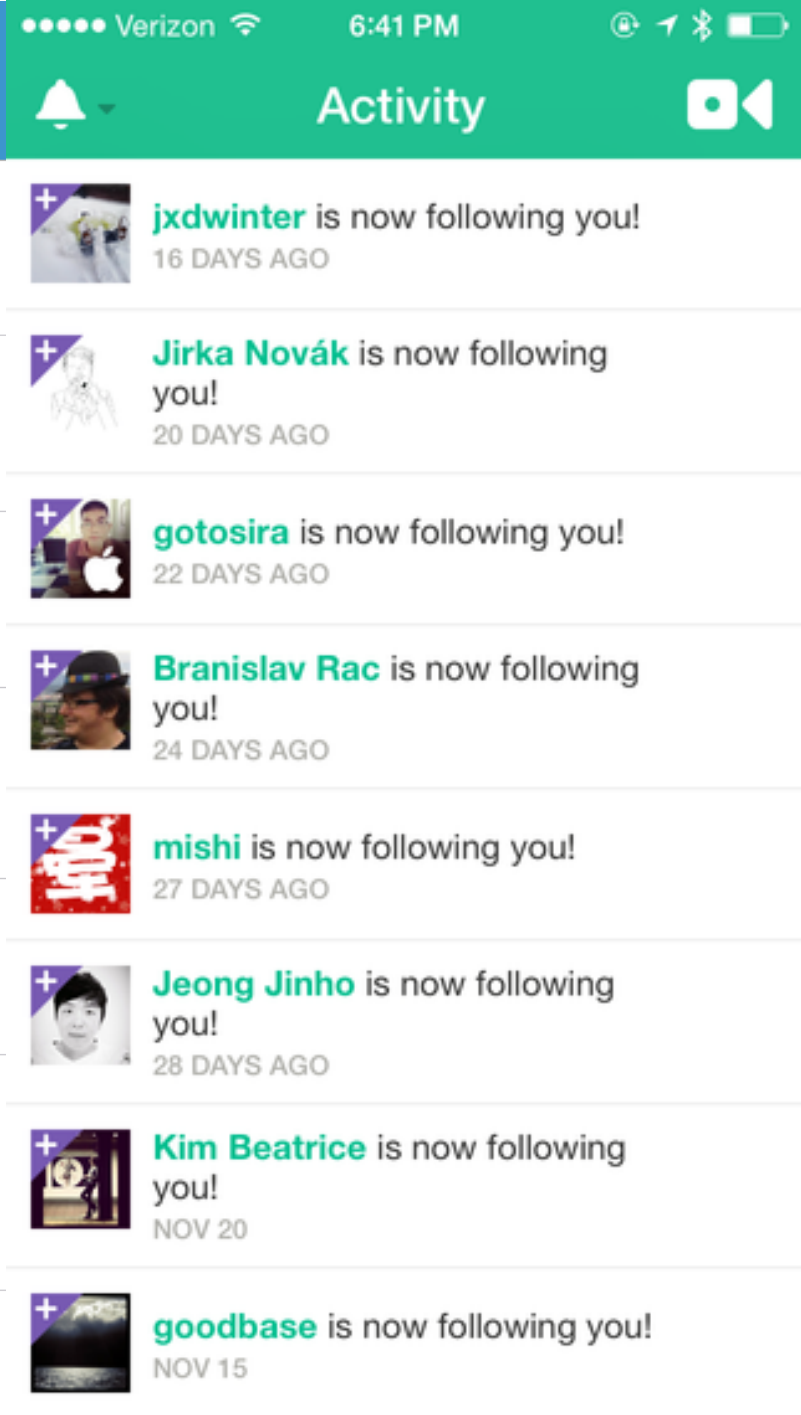
# Activity Feeds

Examples

Quora, Swarm, Venmo, Vine






## Problem

The user wants to keep up with what's happening around them and get quick updates on recent activity.

## Solution

Show recent activity that's relevant to the user within the app. Aside from the obvious Facebook or Twitter news feeds, other apps that contain an element of social interaction, like Quora or Swarm have implemented activity feeds that provide users with an overview of recent activity from their friends or people they follow. The activity stream can be used to aggregate recent actions by an

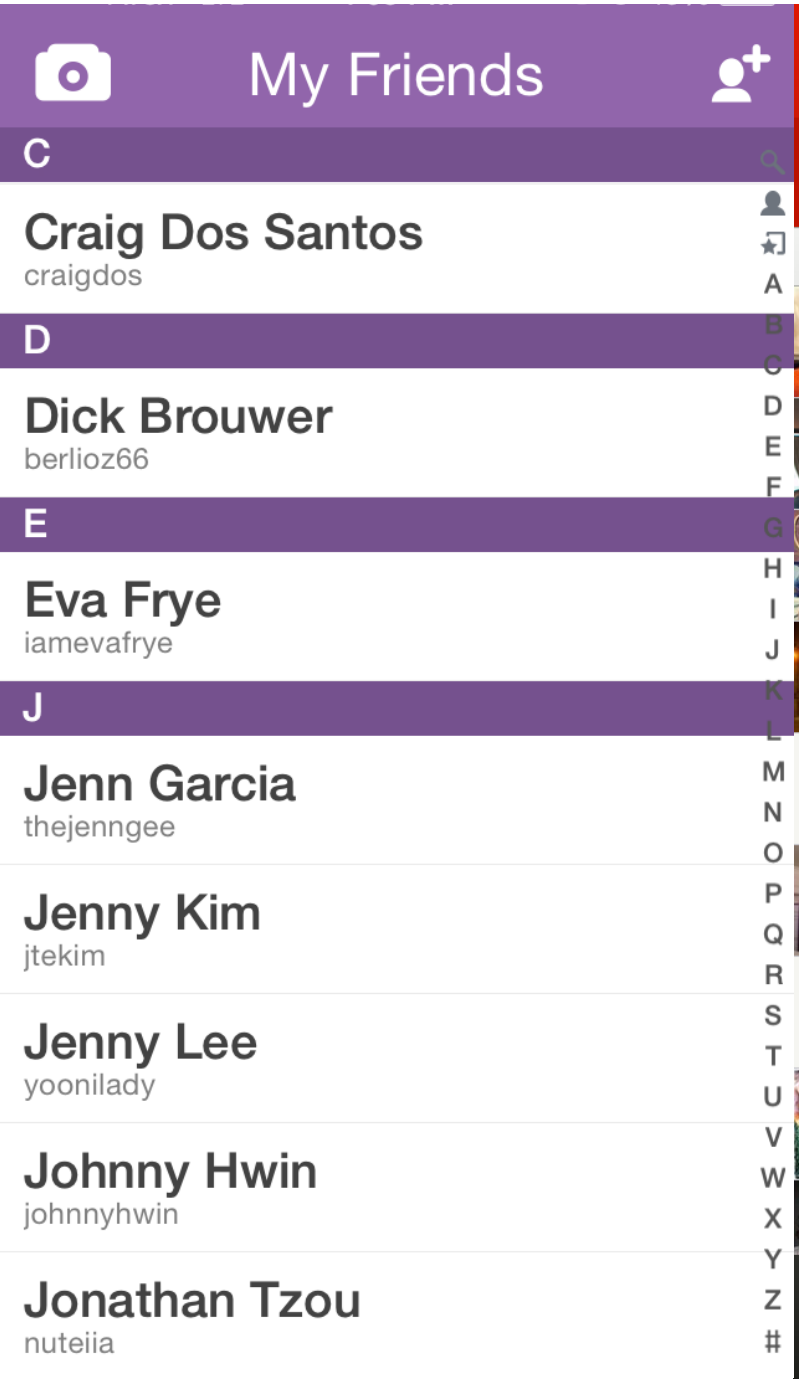
individual user, commonly used on profile pages; more commonly however, activity feeds are used to aggregate multiple users from the perspective of one user. These feeds are extremely useful in demonstrating different features of the UI by showing how other users are interacting with it, and in this also plays a great word-of-mouth role.

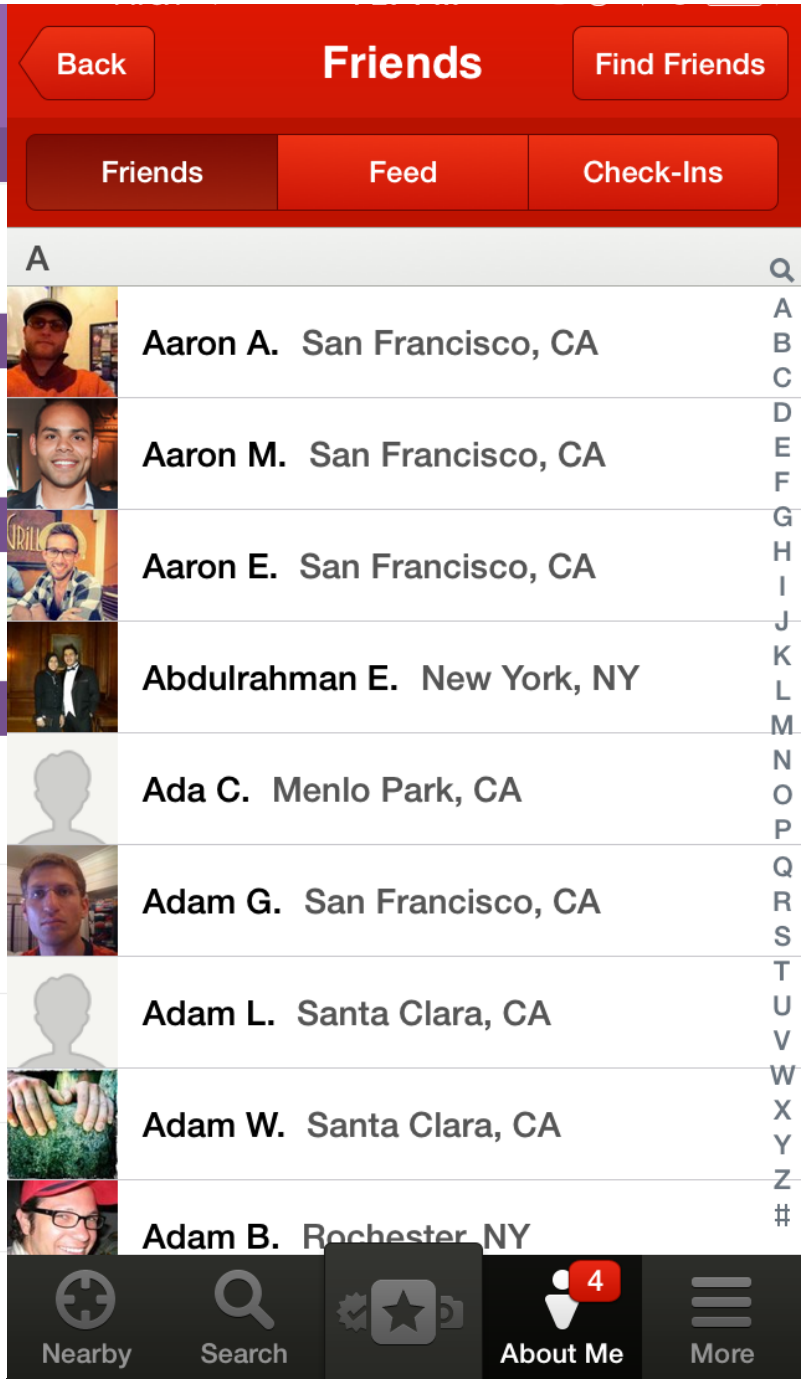
Quora and Venmo are two of my favorite activity feeds because “learning” and “earning” are two of the primary things people do in life. It’s fascinating to passively see people I know provide meaningful answers about self-improvement while others are spending their hard-earned money on pixels and lip rings.

# Friend Lists

Examples

Snapchat, Songkick



### Problem

The user wants to keep track of their friends and contacts within the app.

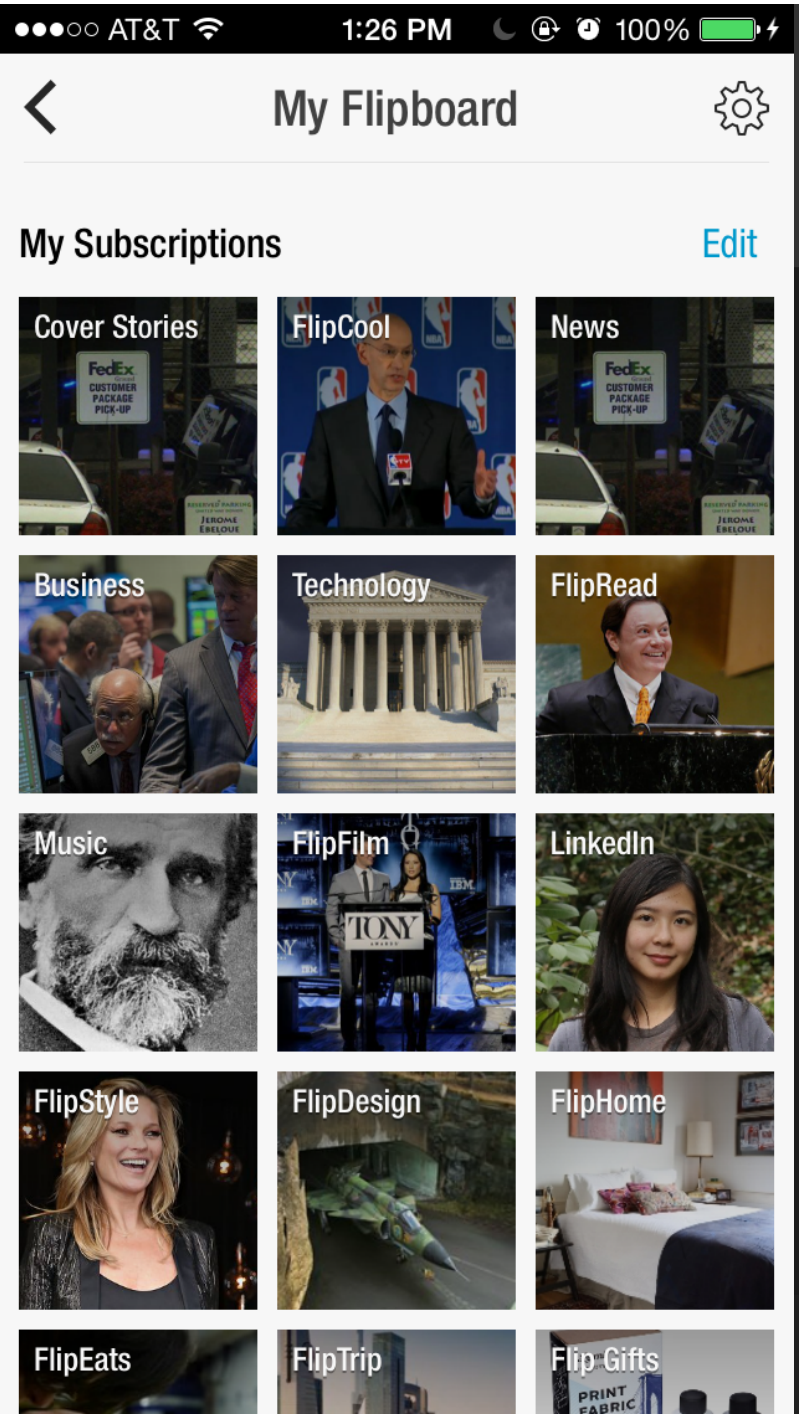
### Solution

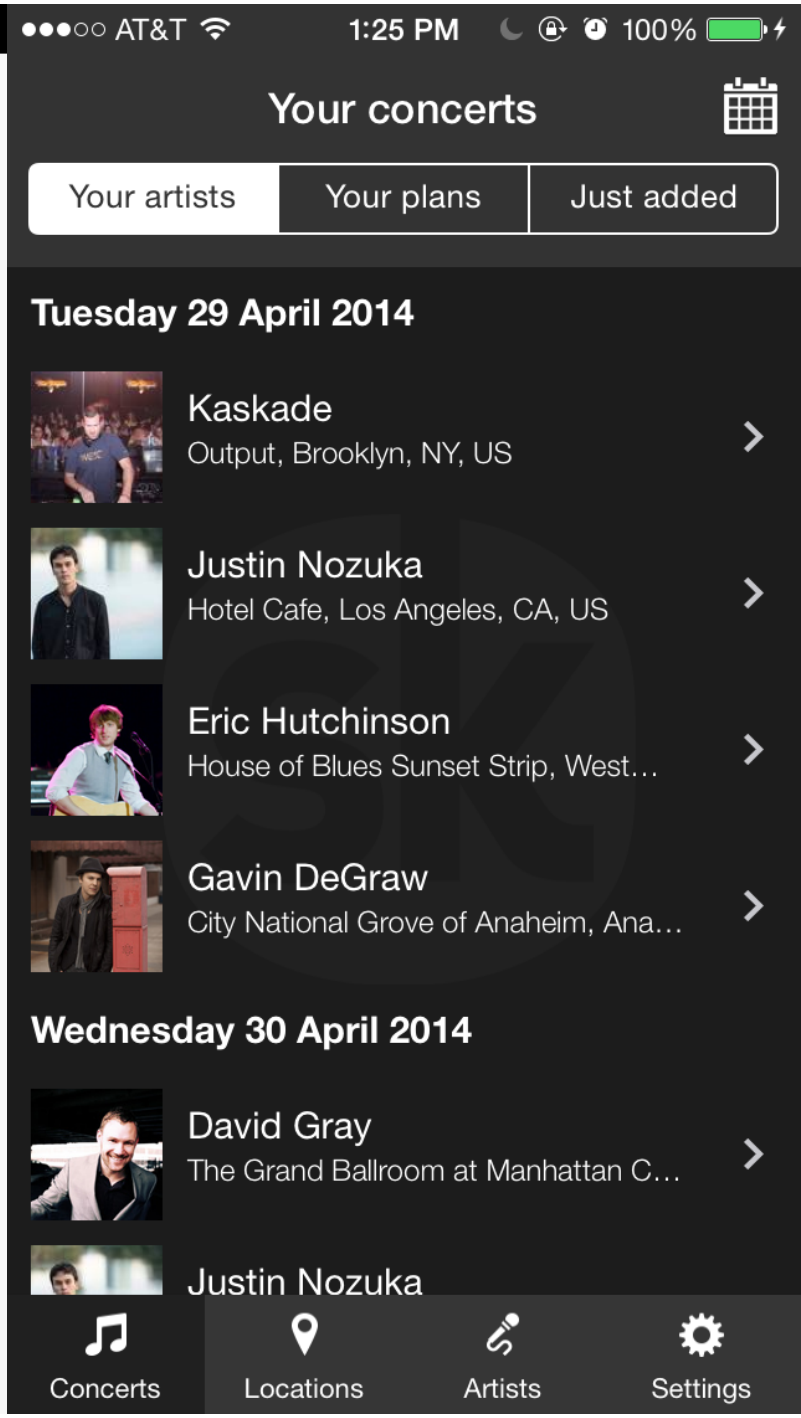
Show all the user's connections or friends in a list. Snapchat and Yelp are part of the growing number of apps that give you friend lists. Whether it's one-on-one communication or keeping track of someone's tastes and preferences, the way users explore their blossoming friend groups will become increasingly contextual, requiring friends to become a more integral part of the web and mobile experience.

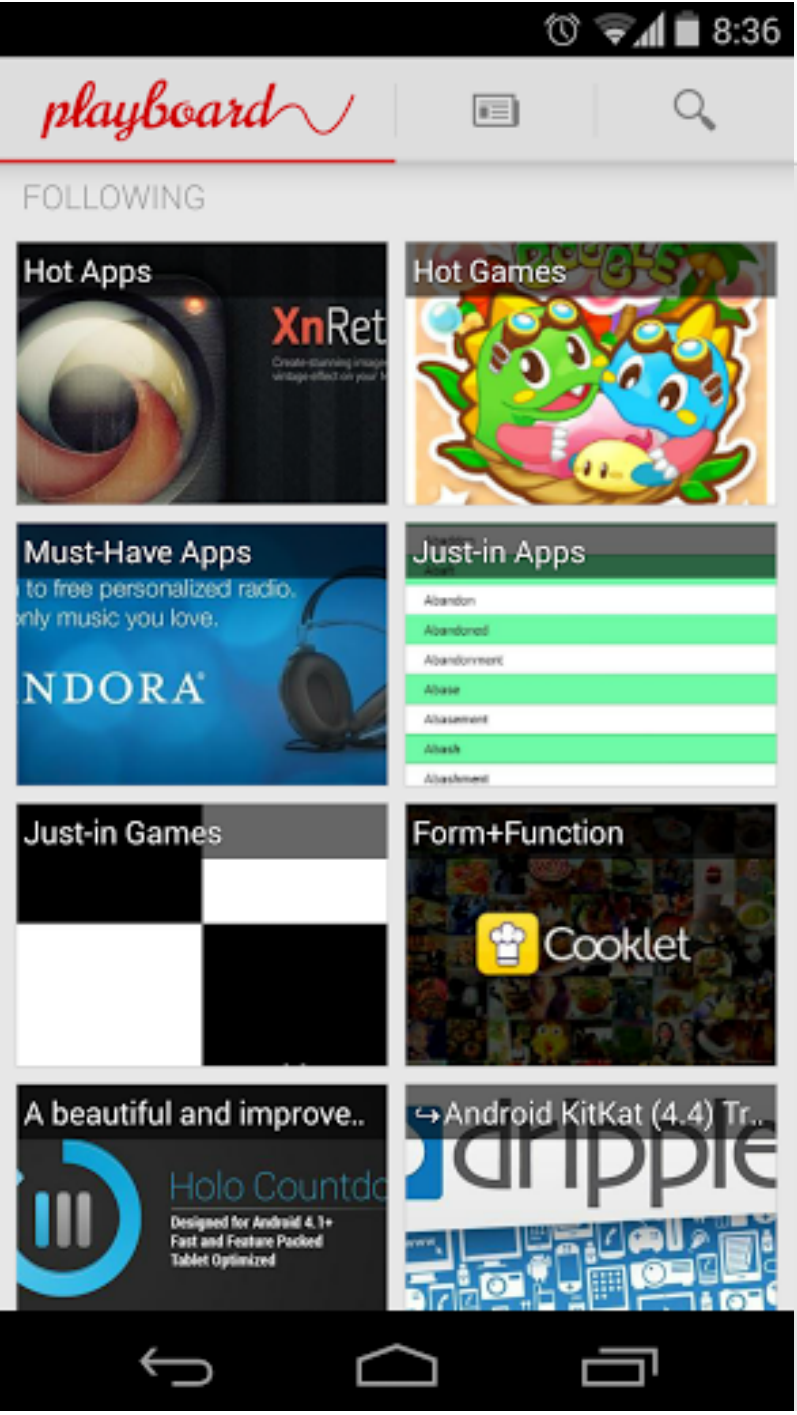
# Follow

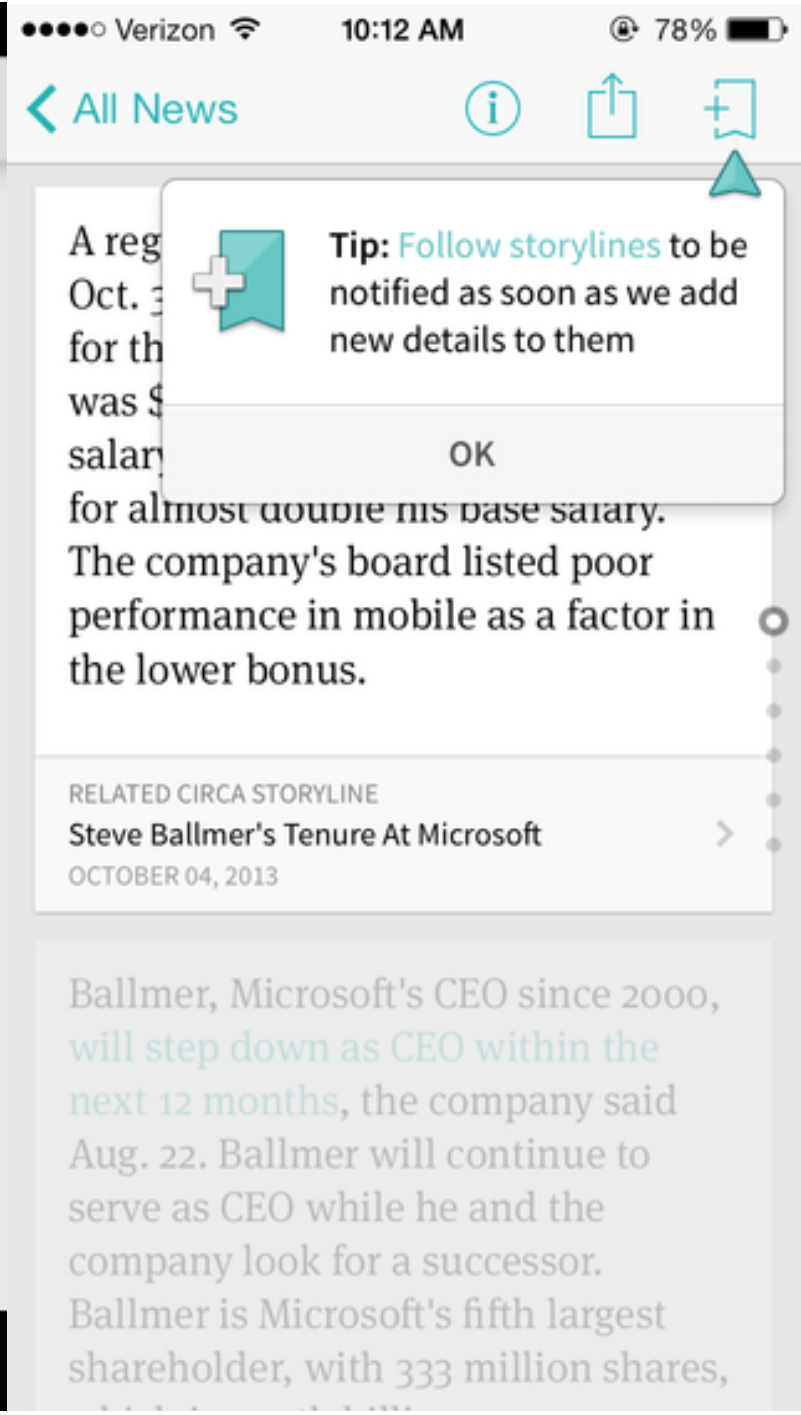
Examples

Flipboard, Songkick, Playboard, Circa






## Problem

The user wants to track and keep up to date with activity on topics or themes, not just people.

## Solution

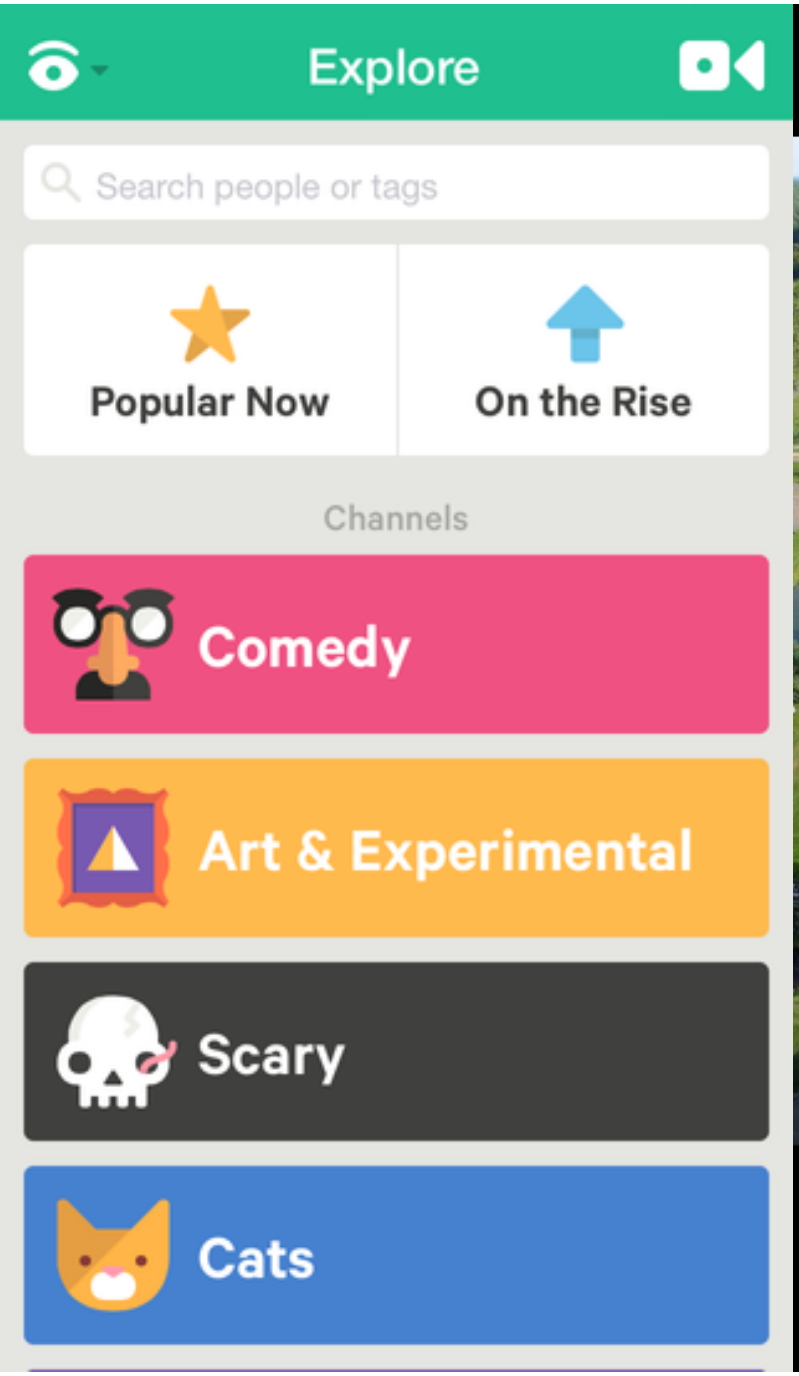
Let users select items that they want to stay up to date with. Aside from the purely social apps like Twitter, other apps like Circa, Playboard, Flipboard and Songkick let you select channels or artists that you want to keep track of, and updates are shown in the user's newsfeed. Whether you have friends or not, there's endless user-generated content to keep you busy. For the same reason friend lists will become an

increasingly important UI design pattern, so will following.

# Vote to Promote

Examples

Vine, 9Gag



### Problem

The user wants to endorse and share content they like.

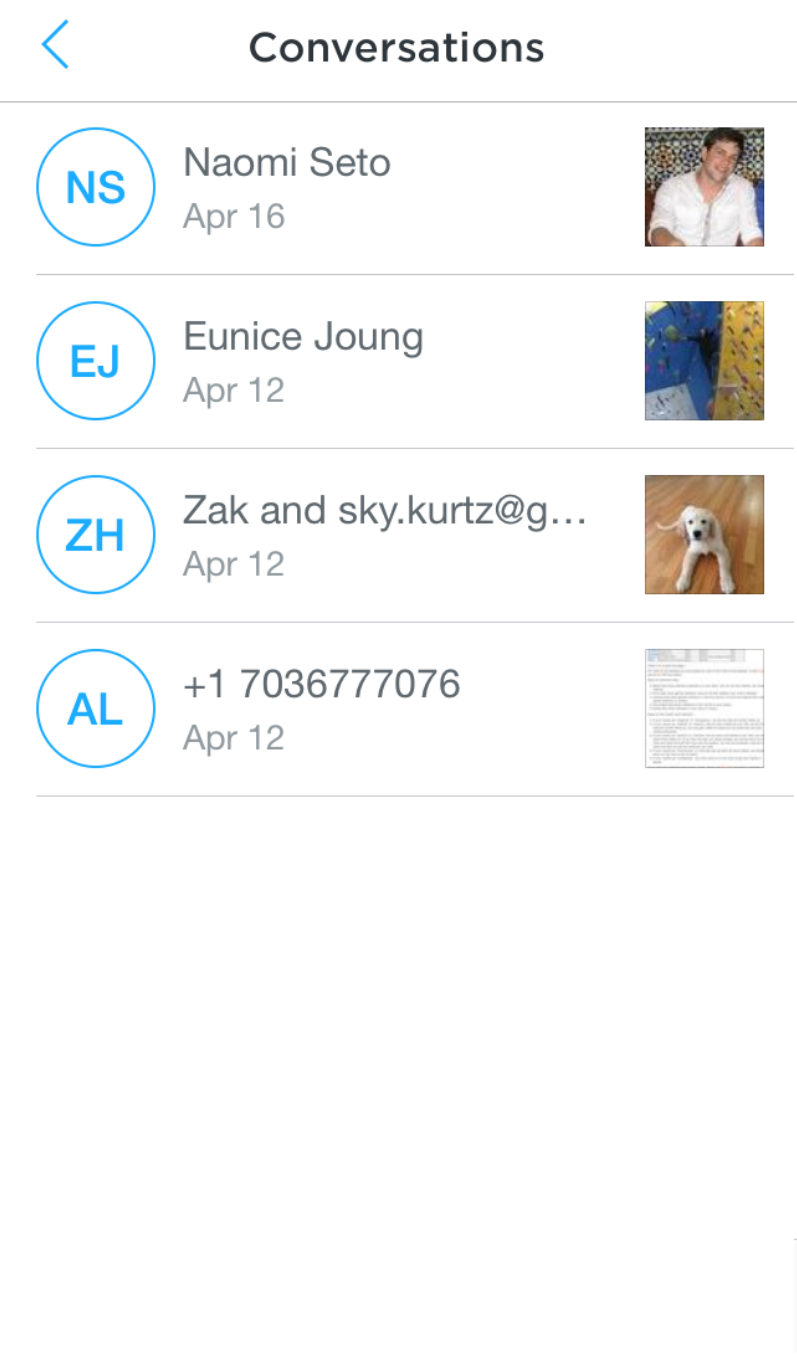
### Solution

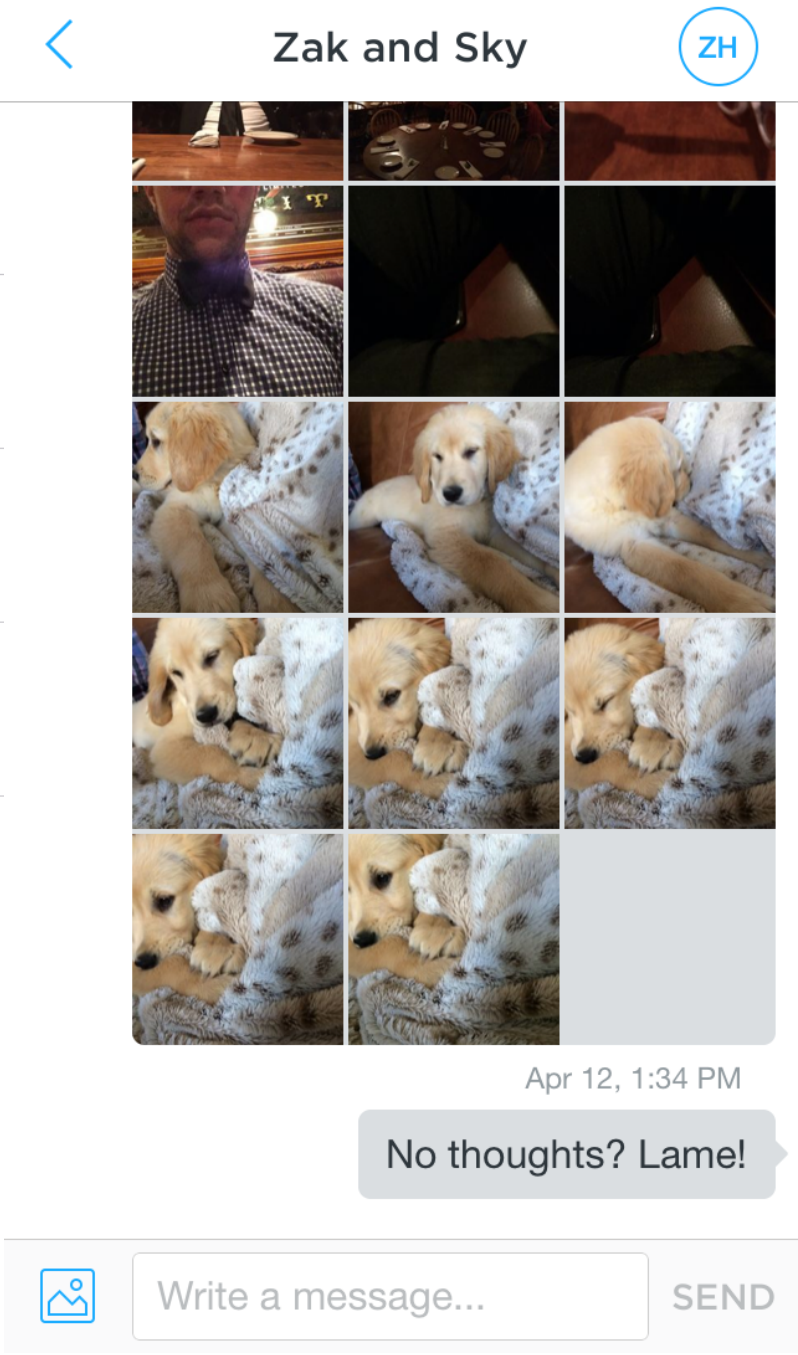
Let users participate in content curation by designing a voting system, where content they like can be promoted. The idea of crowd-sourced content curation was popularized by the likes of Digg and Reddit, and today we see almost every app that has user generated content integrate this pattern to bring up the best.

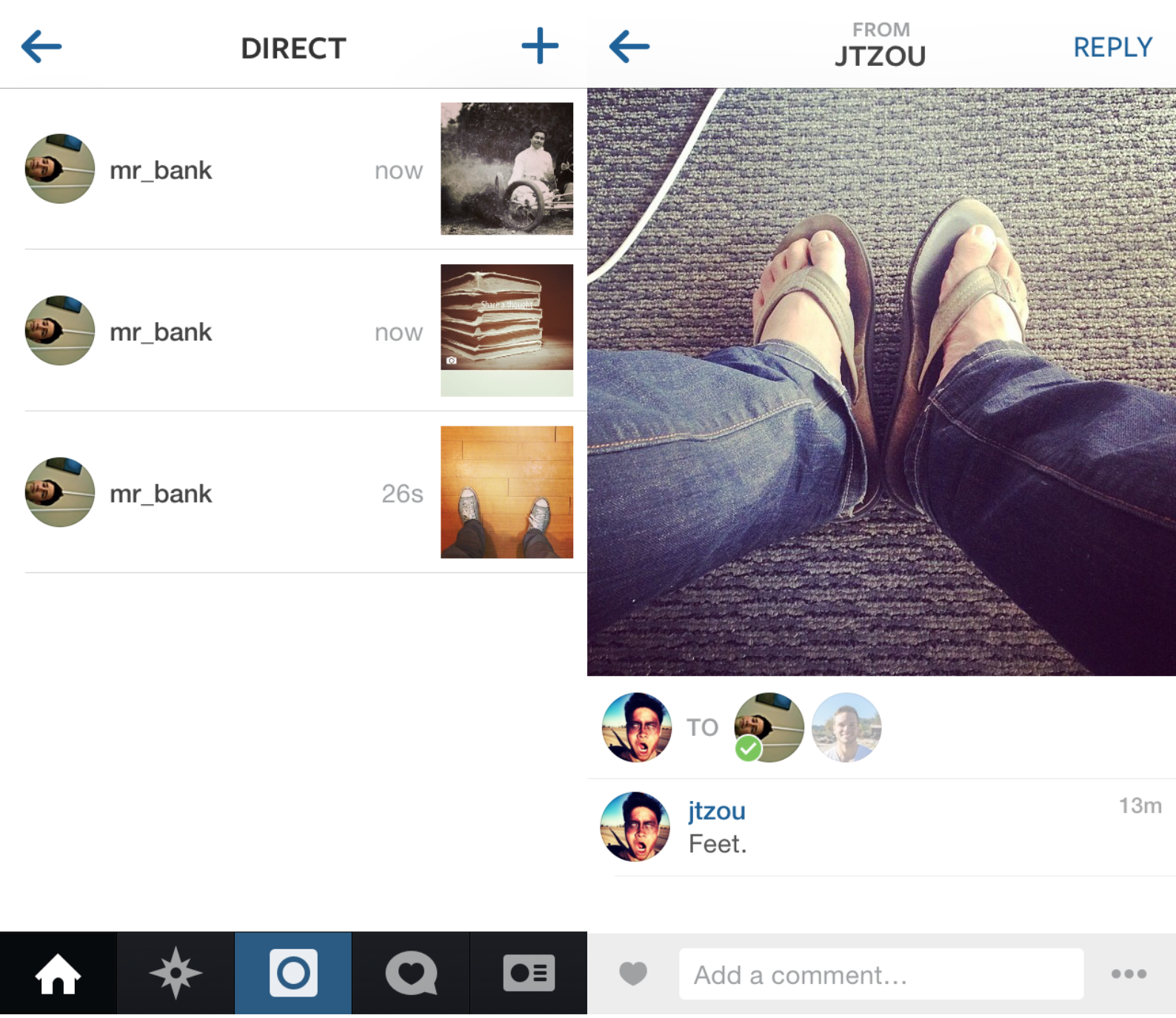
# Direct Messaging

Examples

Carousel, Instagram






## Problem

The user wants to send private messages to their friends from within the system.

## Solution

Allow users to interact with each other in private messages alongside their other interactions. Carousel and Instagram and many other apps offer chat or direct messaging as an integral part of their experience. Private chat UI design patterns will continue to blossom across many apps, not just traditional “social networks” now that users are finally comfortable sharing more private things online and they have substantial breadth in the content they’re generating online, even their financial transactions on apps like Venmo.

# Single Share Button

Examples
Medium, 9Gag

If you're in the app, go ahead and swipe left to your next story right now, and enjoy the best of Medium.

Recommend Share

**Ev Williams**
Published March 20, 2014

**Beautiful Stories**
Edited by Medium

### Problem

The user wants an easy way to share interesting content via various channels.

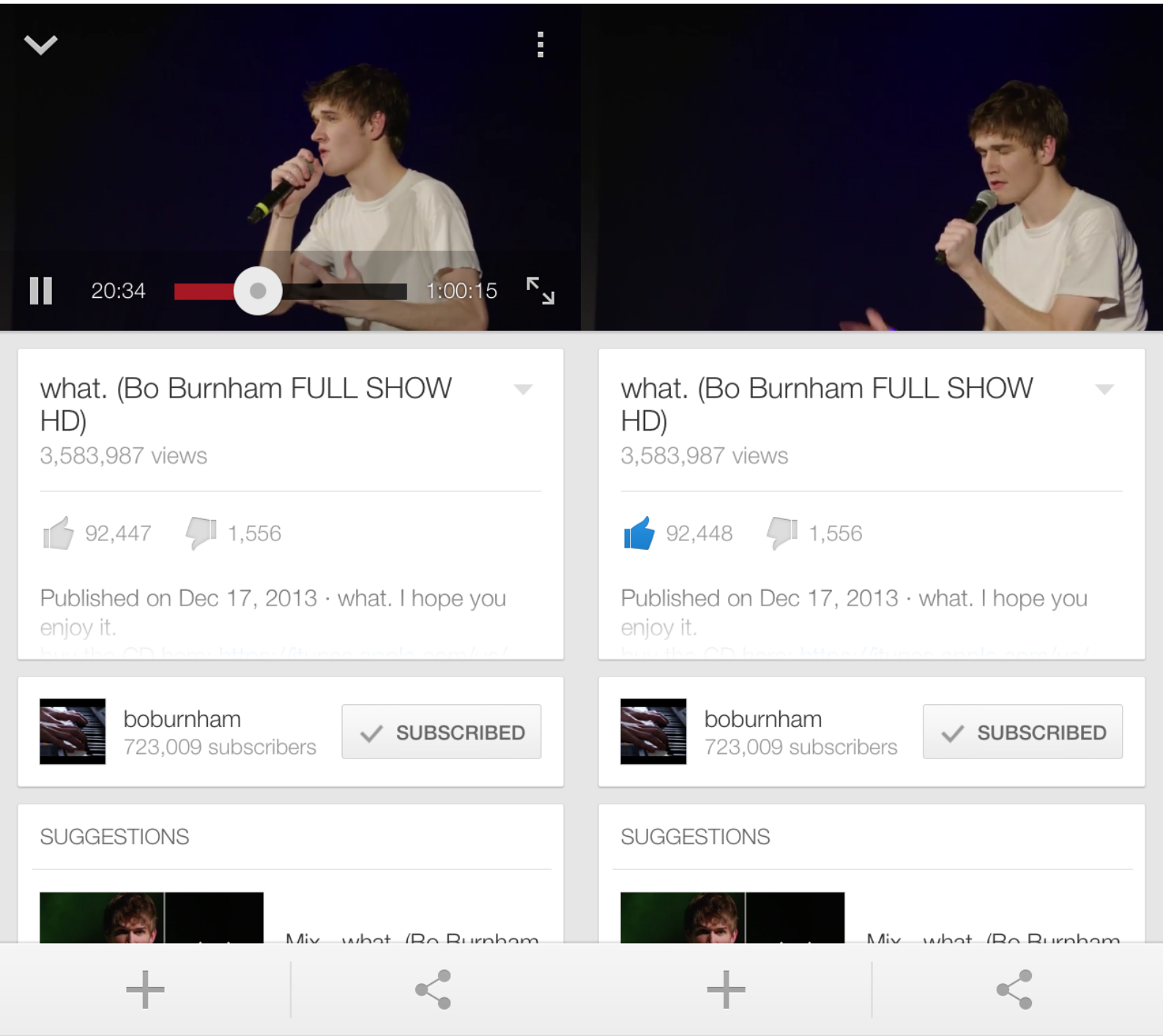
### Solution

Provide an easily accessible share feature that can be accessed through a button (or gesture). Because of the limited space on mobile UIs, most apps consolidate this into a single button instead of showing all the options up front. Medium, like many other apps, has consolidated the ugly “share” widgets with a single share button to give you a beautiful experience as well as a clear action to share the content, regardless of where you want to share it. The UI can integrate with the platform to provide greater flexibility to the user, for example iOS integrations with Facebook and Twitter and the Android share actions which allow transferring content through to any other app installed on the system.

# Like

Examples

YouTube

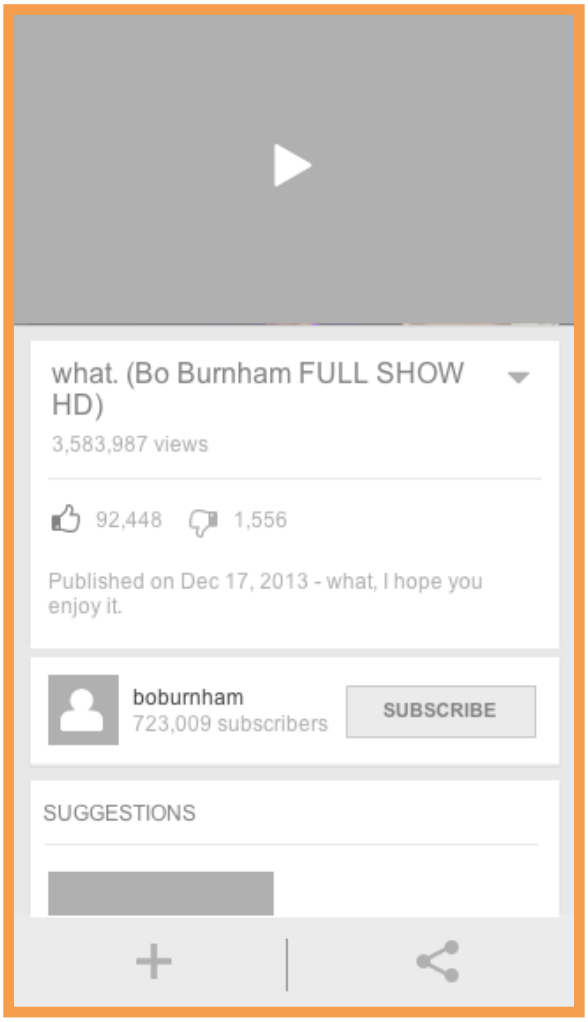


## Problem

The user wants to rate content in a simple way without having to worry about the degrees to which they like it.

## Solution

Simplify rating controls by making them binary choices - the user either likes it or dislikes it. Eliminating the fine-grain of stars and rating scores, this makes rating things easier for users as well as interpreting them. If I liked a video, should I rate it 4 stars or go all the way with 5 stars? YouTube and almost every application lets you like (or even dislike) everything in a binary way instead. A lot of apps provide a way of showing appreciation by simply “liking” or “hearting” content.

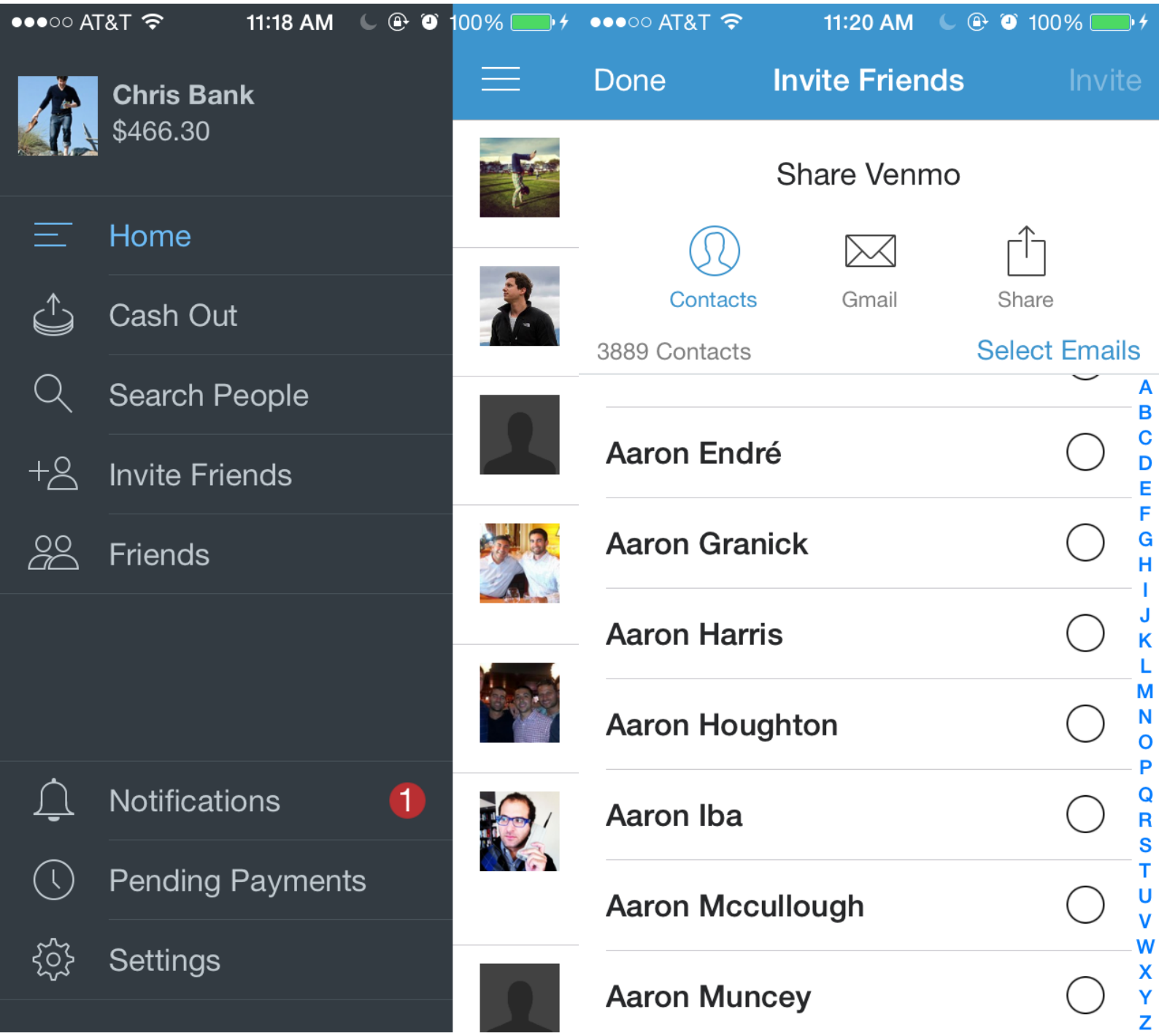


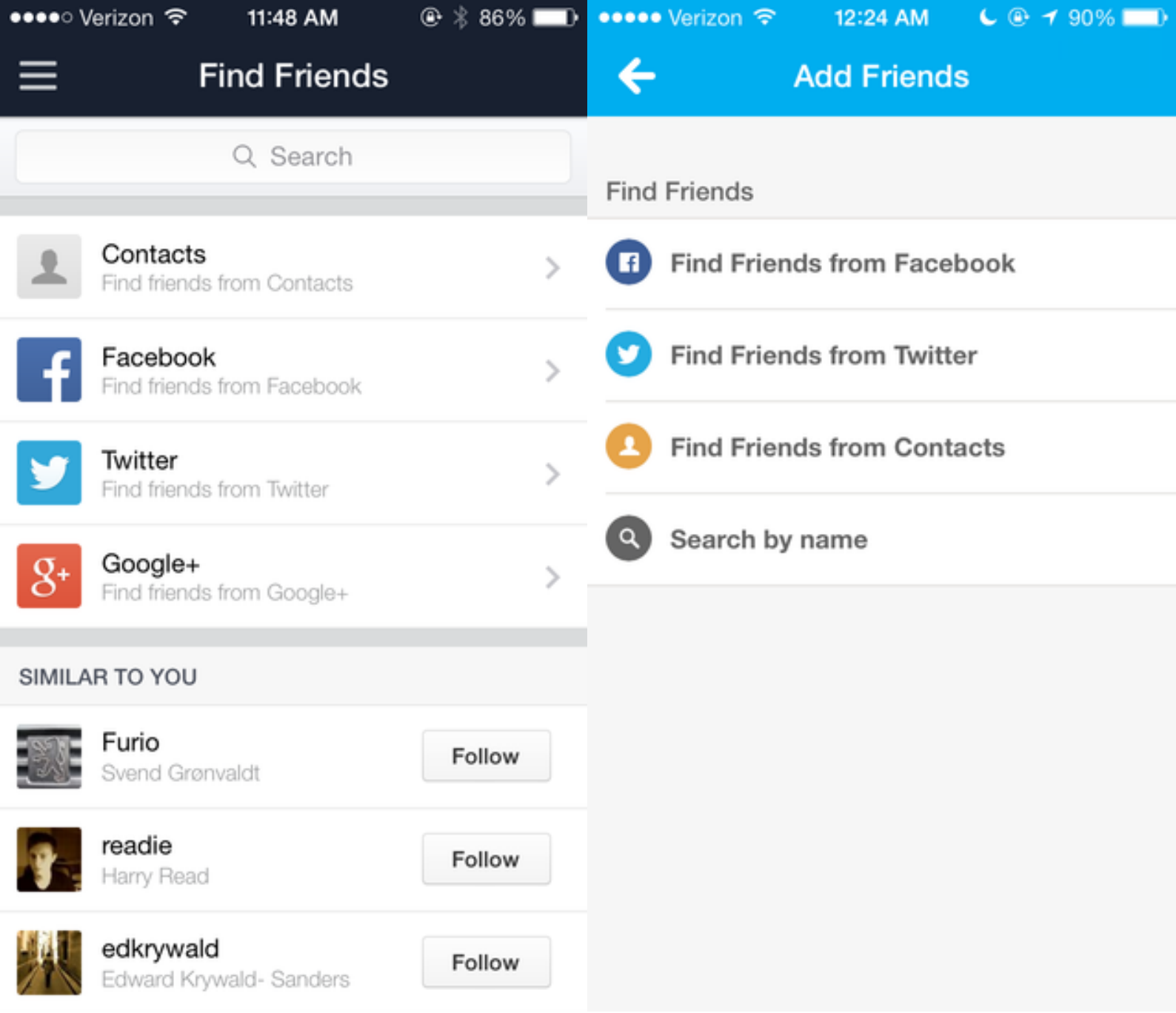

Click to use this wireframe in UXPin

# Find & Invite Friends

Examples

Venmo, Foursquare, Fancy






## Problem

The user wants to experience the application with their friends.

## Solution

Make the invitation process simple and easy to complete. Venmo, for example, makes it really easy to invite others through social, mobile contacts, and email integrations. Since word-of-mouth and referrals are a huge driver of growth especially in consumer applications, you'll see this UI design pattern proliferate and evolve even more. The invite feature can be built into the onboarding pattern or even as the empty state design, both of which we've covered earlier.

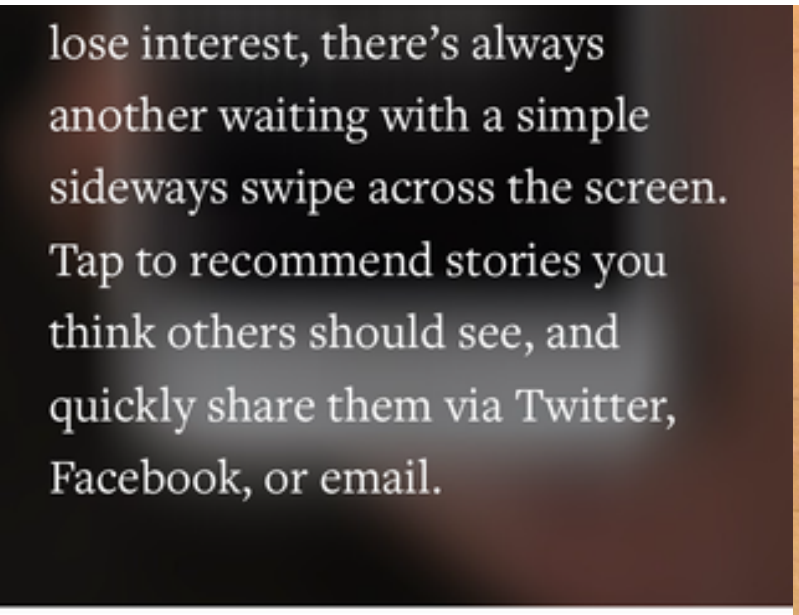
# 5

# Data & Content Management

# Full-Screen Modes

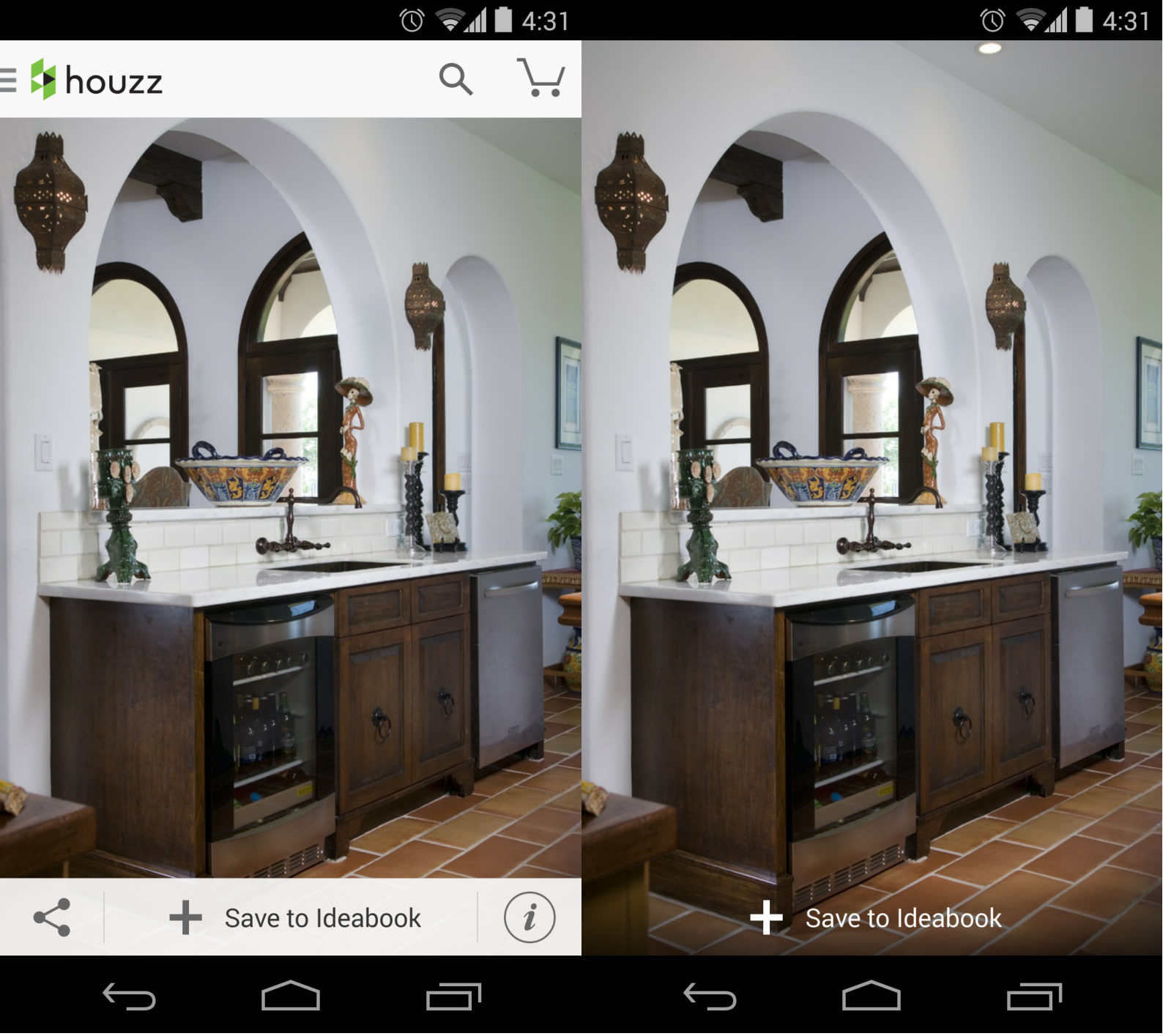
Examples

Medium, Snapchat, Houzz



I've been carrying around the beta version of Medium for iPhone for the last couple months, and it has become not only my favorite way to experience Medium, but my reading app of choice wherever I am.

If you're in the app, go ahead and swipe left to your next story right




## Problem

The user wants to focus on content instead of being distracted with the UI.

## Solution

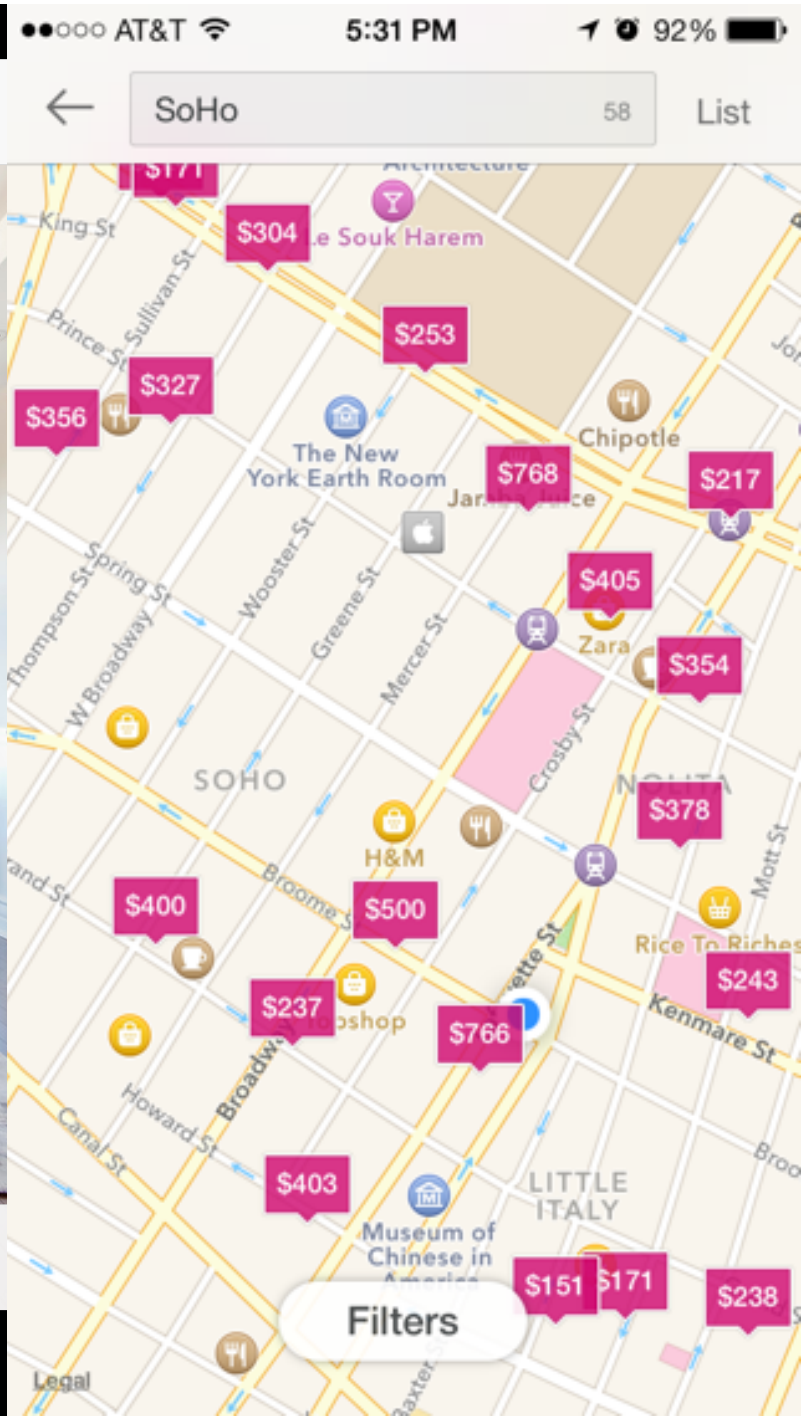
Design a full-screen mode that hides or minimizes the UI clutter around content. This helps users focus on what matters, rather than being distracted by the clutter of the UI. Besides being an essential for video players, this pattern works particularly well for multimedia apps like Medium, Houzz and Kindle, which let users minimize the UI "frame" by tapping on the main content. This minimizes the navigation and other buttons on screen, making for a more immersive experience with the

content when users need it. Snapchat implements this in their camera as well, getting rid of most of the UI “chrome” in favor of a minimalistic navigation, showing you the 1 or 2 more important buttons and change these primary buttons depending on what view you’re in. To get between views, you can either click one of these primary buttons or swipe left-or-right.

# Interactive Content Layers

Examples

Houzz, Airbnb



### Problem

Users want to know which items within a content view they can interact with in further detail.

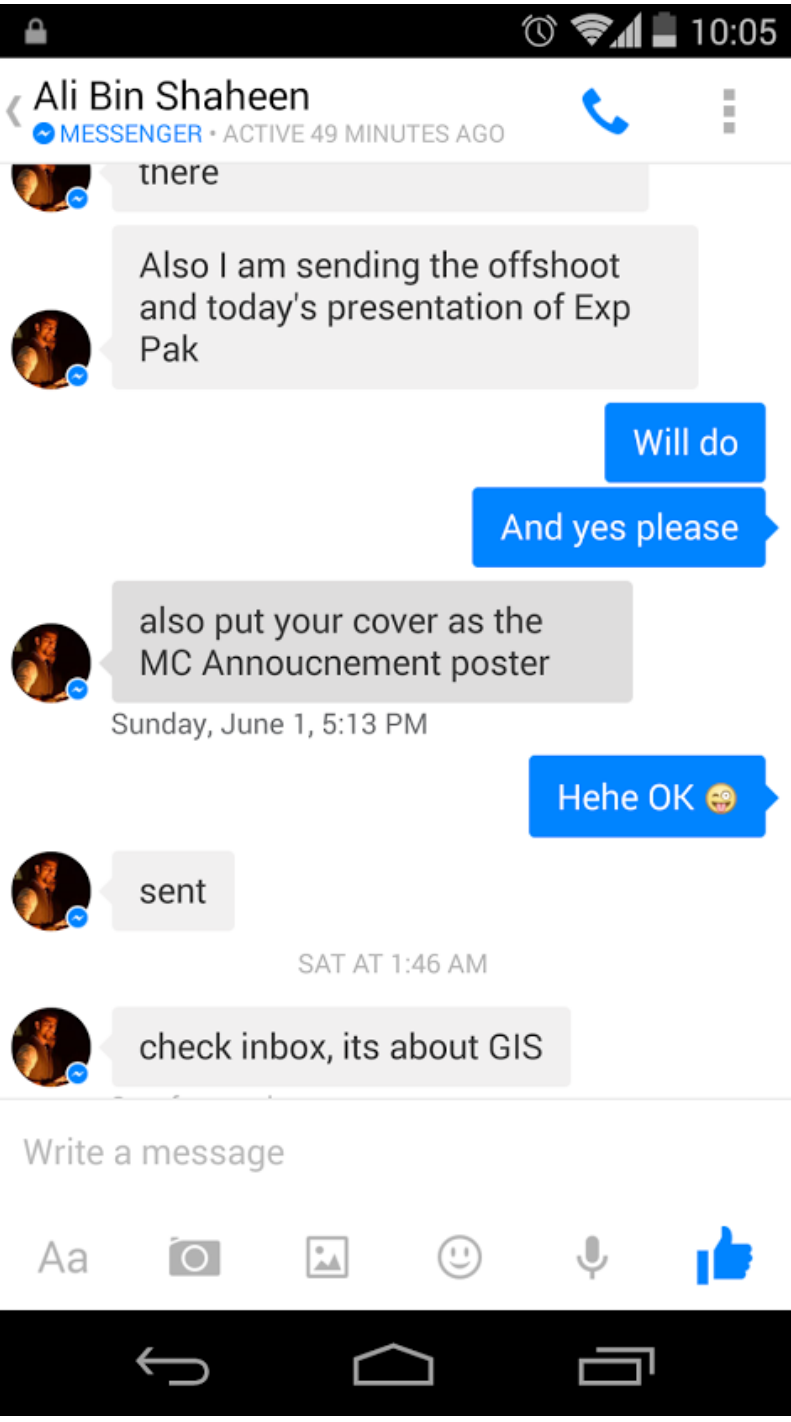
### Solution

Layer interactive items to provide an “augmented reality” approach to your content. Apps like Houzz do this well by placing a price tag icon on individual items in the pictures that are being sold, and tapping on them provides further information. Not only is this a great way of highlighting content within the picture that the user can interact with, but it also adds an element of playfulness by having the tags swing around based on movement and orientation of the device.

# Inline Expanding Areas

Examples

Facebook Messenger



### Problem

The user wants access to relevant secondary details without cluttering the main UI.

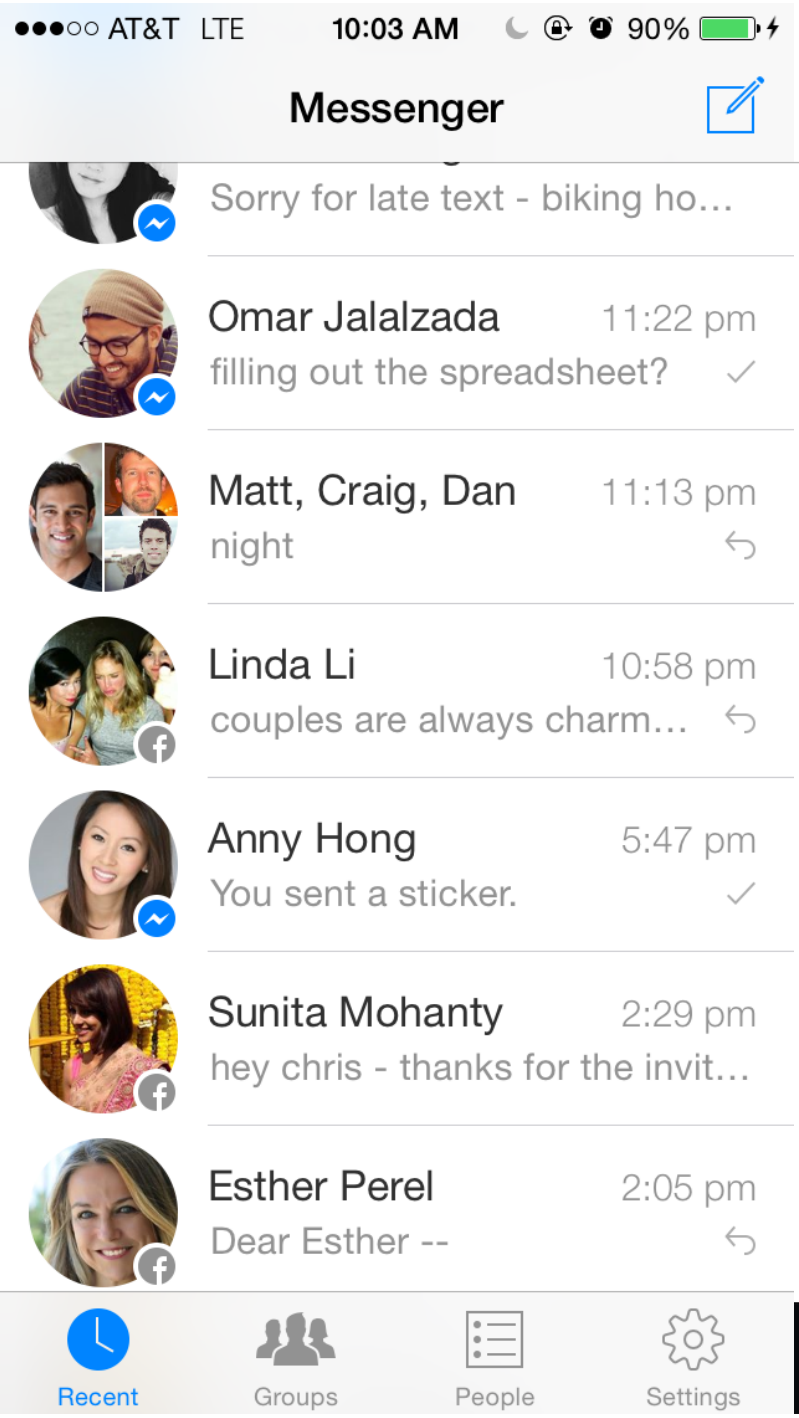
### Solution

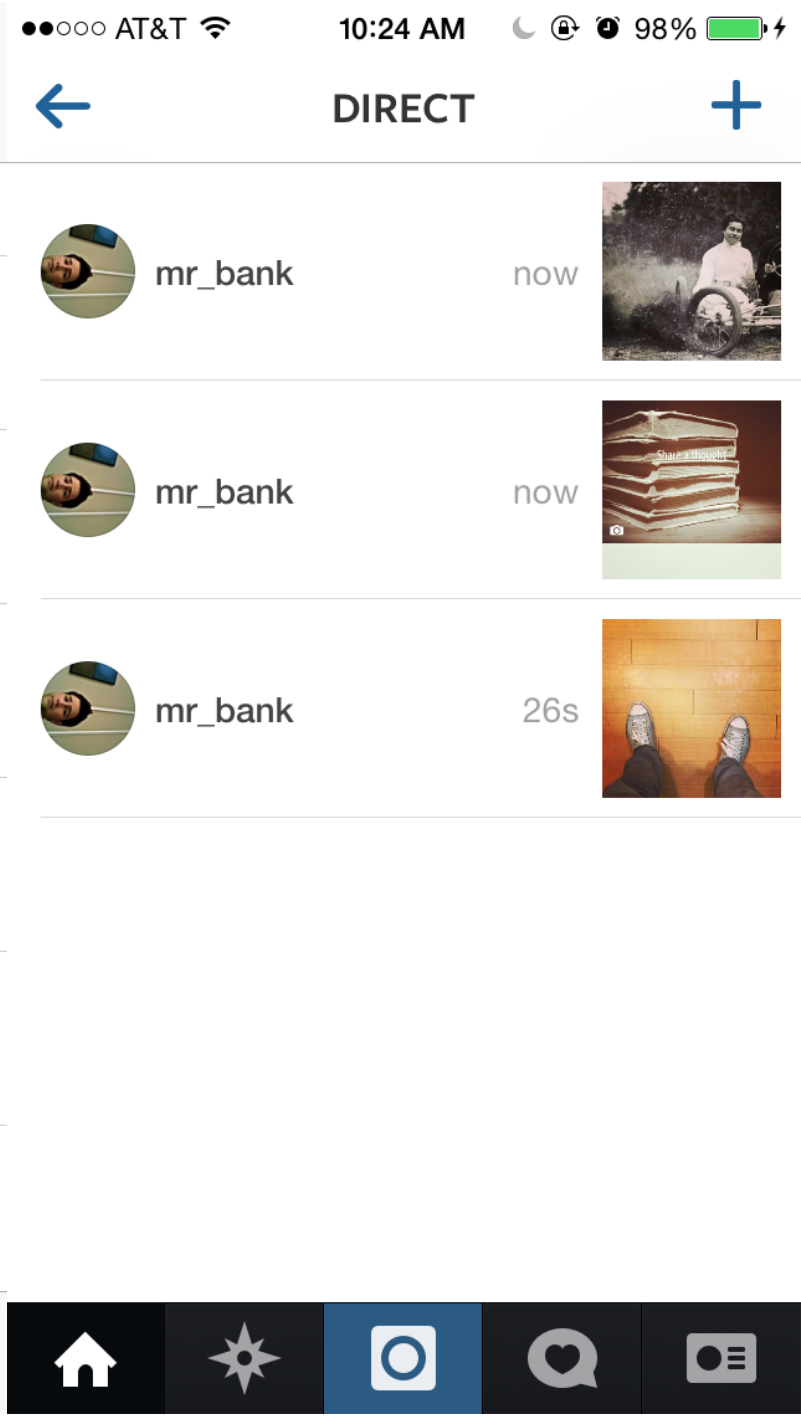
Make metadata invisible unless the user explicitly wants to see it. Facebook Messenger for example hides individual timestamps and location data, making it visible only if the user taps on the particular message.

# Circles

Examples

Facebook Messenger, Instagram



## Problem

The user wants to quickly distinguish between buttons and media.

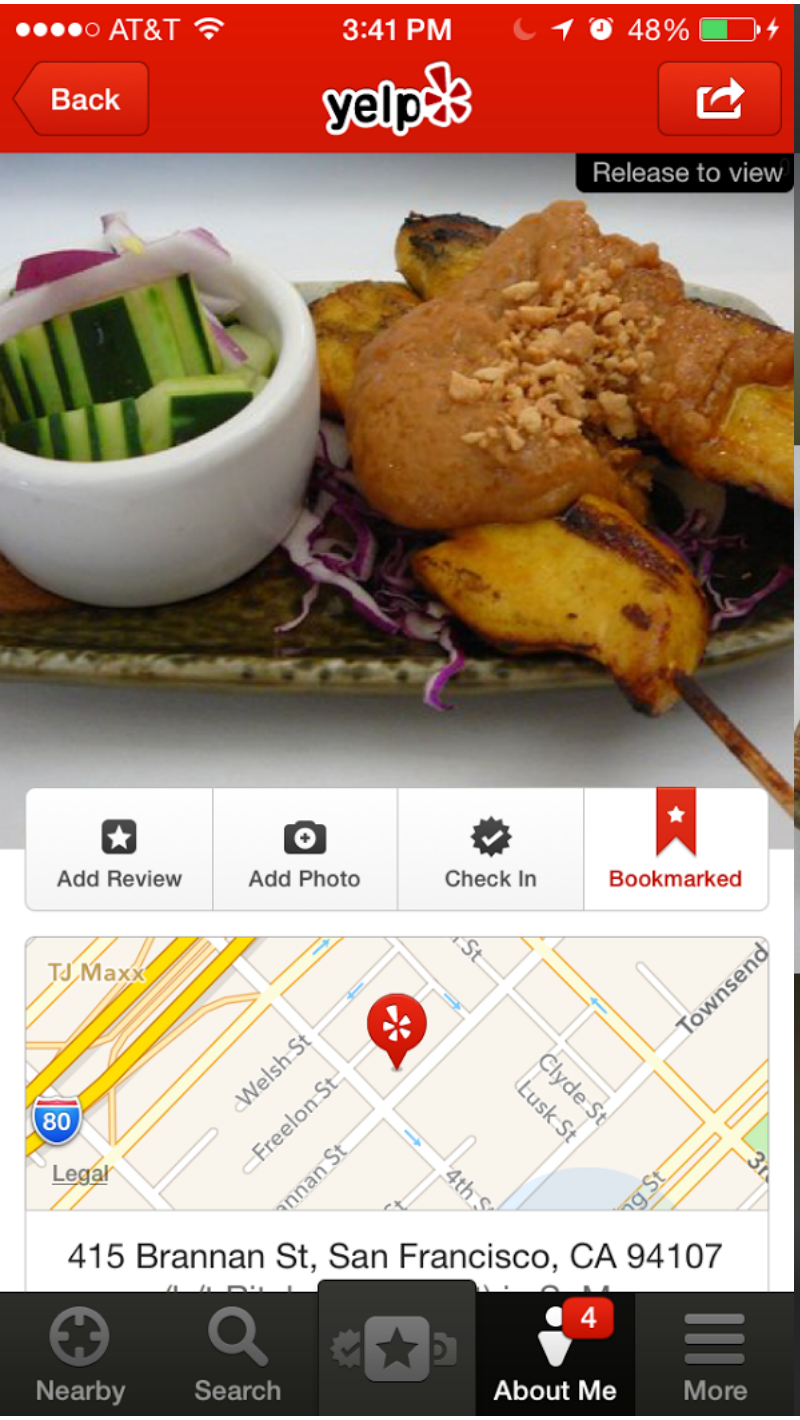
## Solution

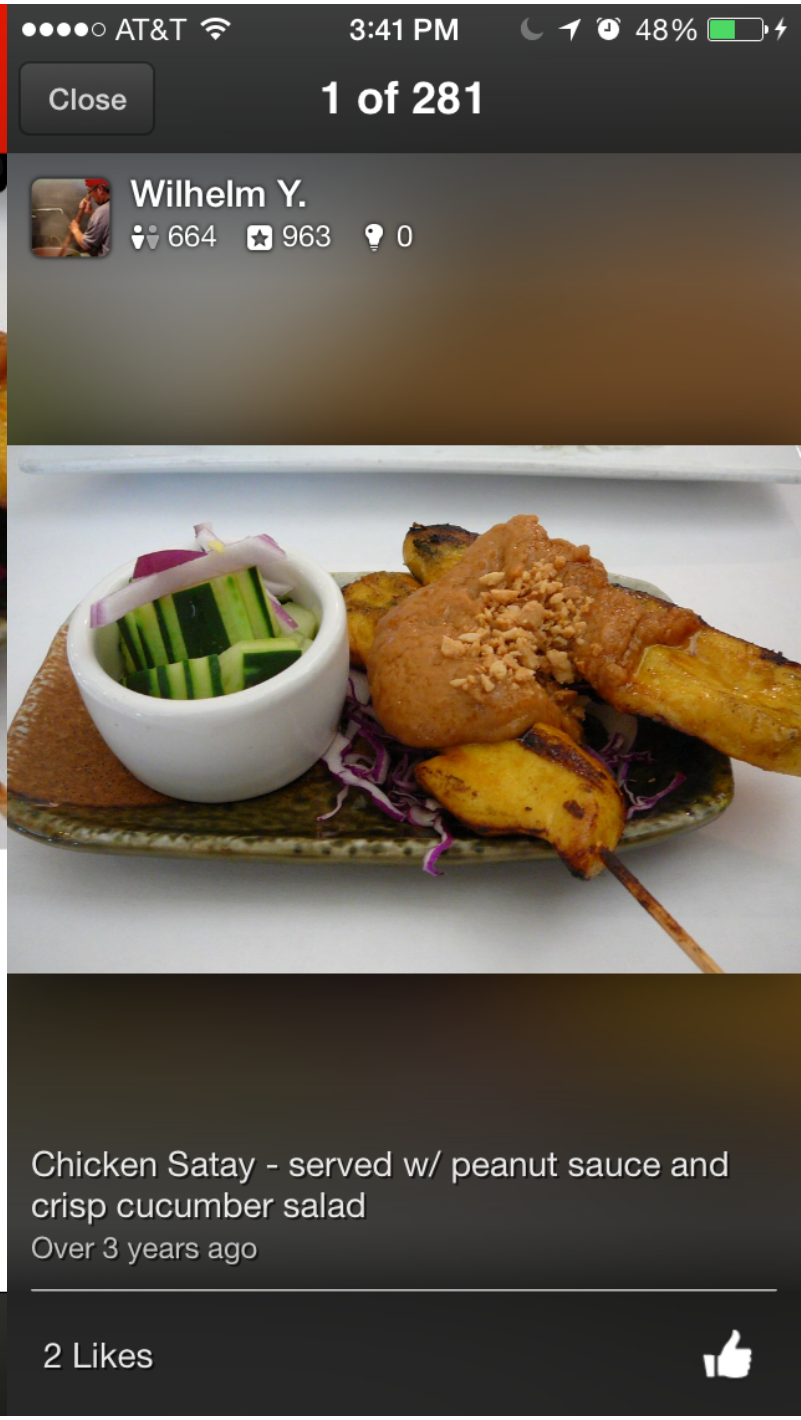
While the traditional photo thumbnail has always been rectangular both on the web and mobile, we see a lot of apps breaking that mold and opting for circles instead. both because it's the optimal tap target for a touch screen but cleans the UI up a bit with the extra white space between adjacent content. Some apps like LinkedIn use different shapes to visually distinguish between action buttons and other media. Some apps like Tinder and Swarm are using circles exclusively. Facebook Messenger and Instagram shows all user thumbnails in circles. Popularized by Google+ and improved by Path in some respects, this UI design pattern is gaining popularity although it's benefit over the traditional square thumbnail is not clear other than adding variety, the unequivocal "spice of life."

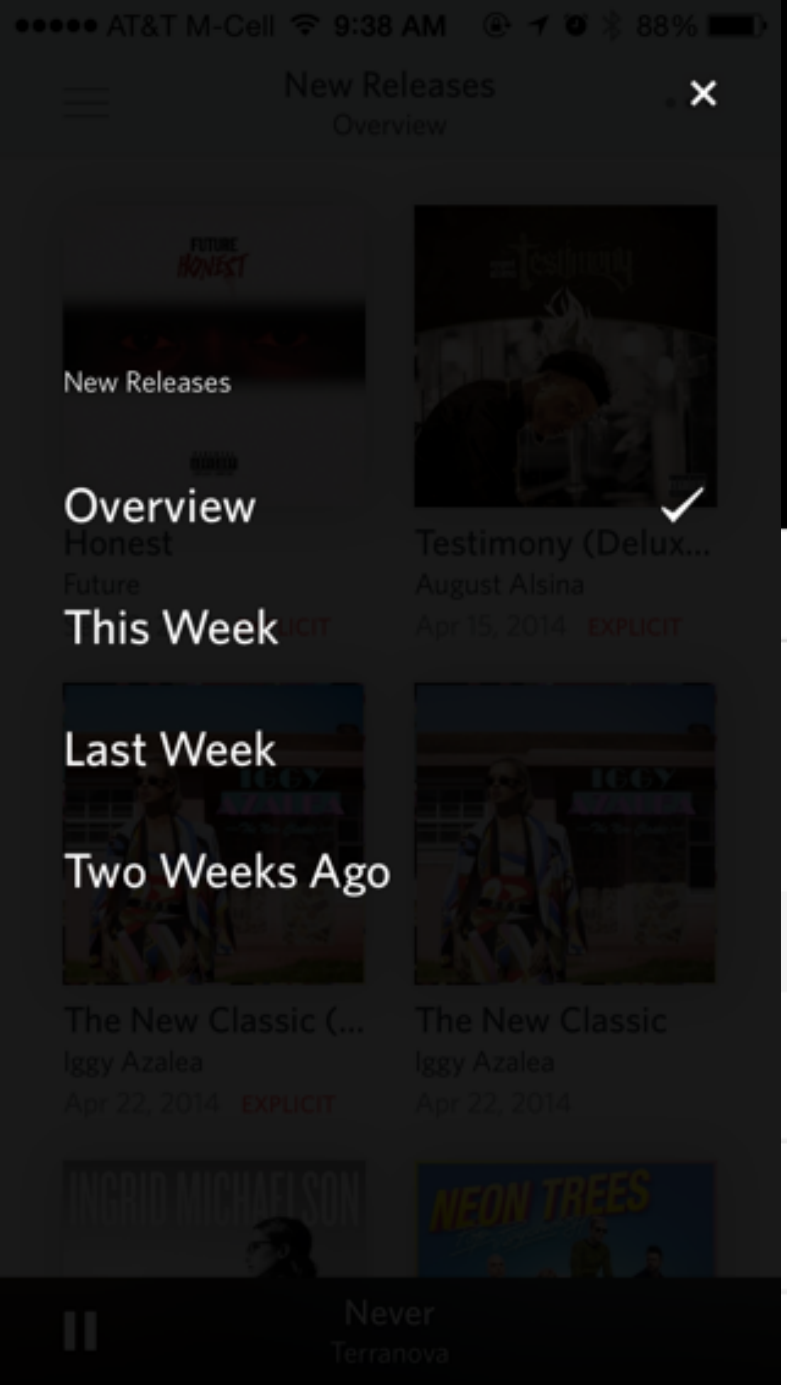
# Transparency

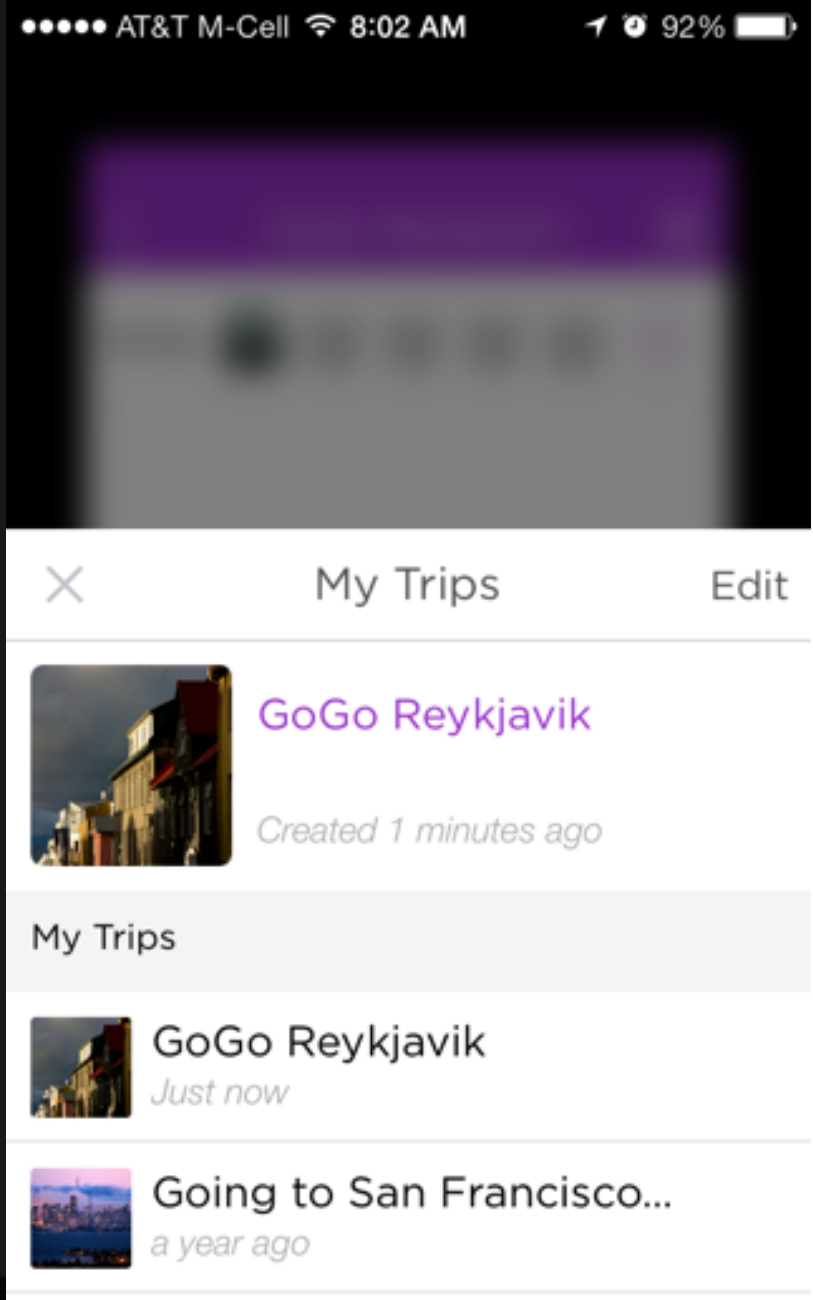
Examples

Yelp, Rdio, Gogobot






## Problem

The user wants to know if there is content behind an overlay.

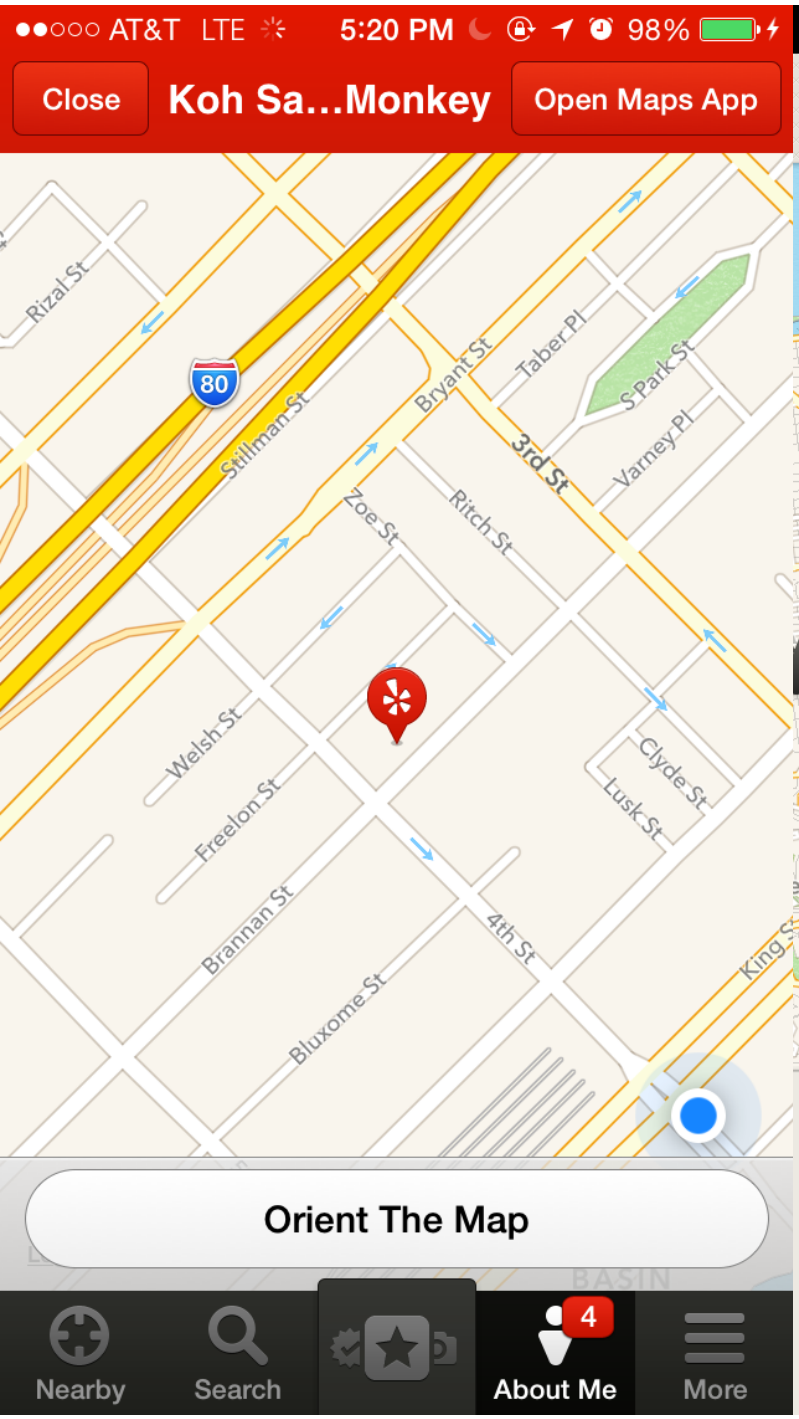
## Solution

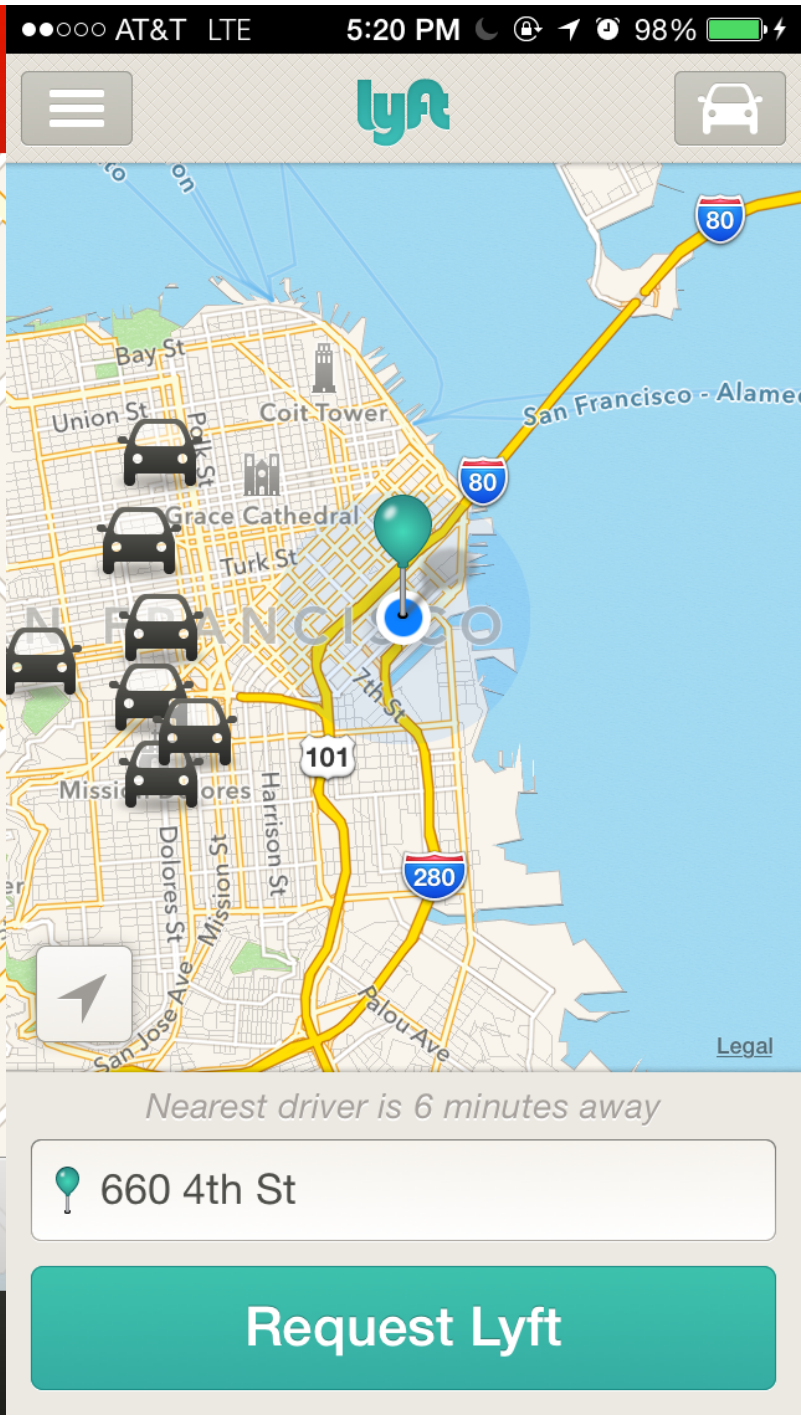
Use gradients and fading overlays to show that there is content layered below. Yelp let's you go between listing details and the photo gallery when you drag downward to further expose the photo hidden behind a semi-transparent listing header. The use of semi-transparency and responsive content creates a wonderful experience here. Rdio and Gogobot use transparency and blur to achieve the same effects, not only providing context to the user about where they are but also making interaction menus look attractive.

# Maps As Backgrounds

Examples

Yelp, Lyft



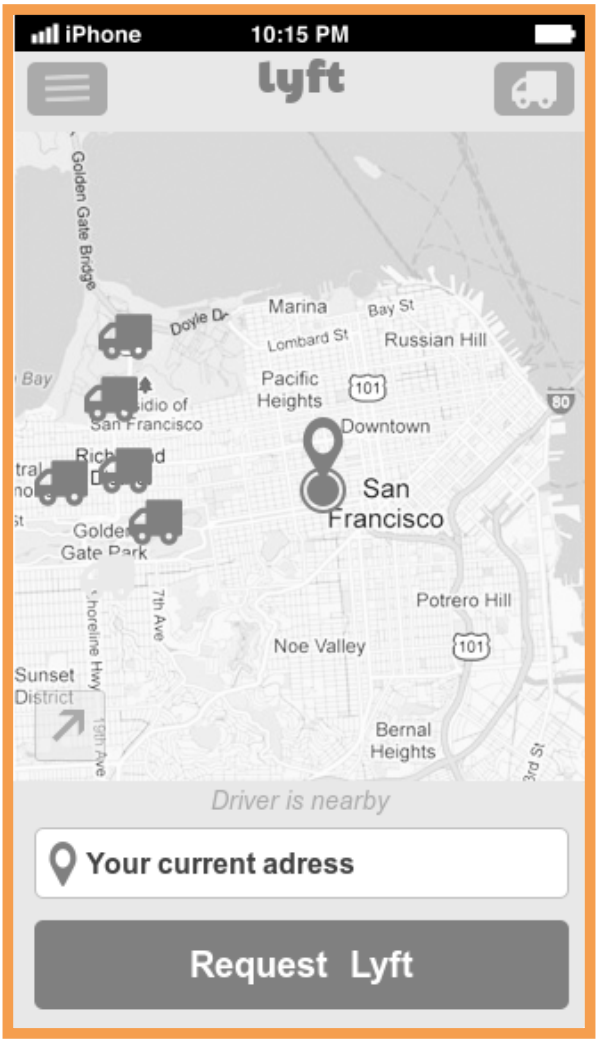
## Problem

Users want to spatially place content on a map to see what's going on around them.

## Solution

Lyft and Yelp provide maps as backgrounds, which makes sense given their local nature. This will become an increasing trend as local applications become more prominent and more information can be layered onto the map view, making maps a full experience not just for one-off directions on web or mobile. You'll also see a lot more UI design patterns that blossom from videos, pictures, and other media as backgrounds.

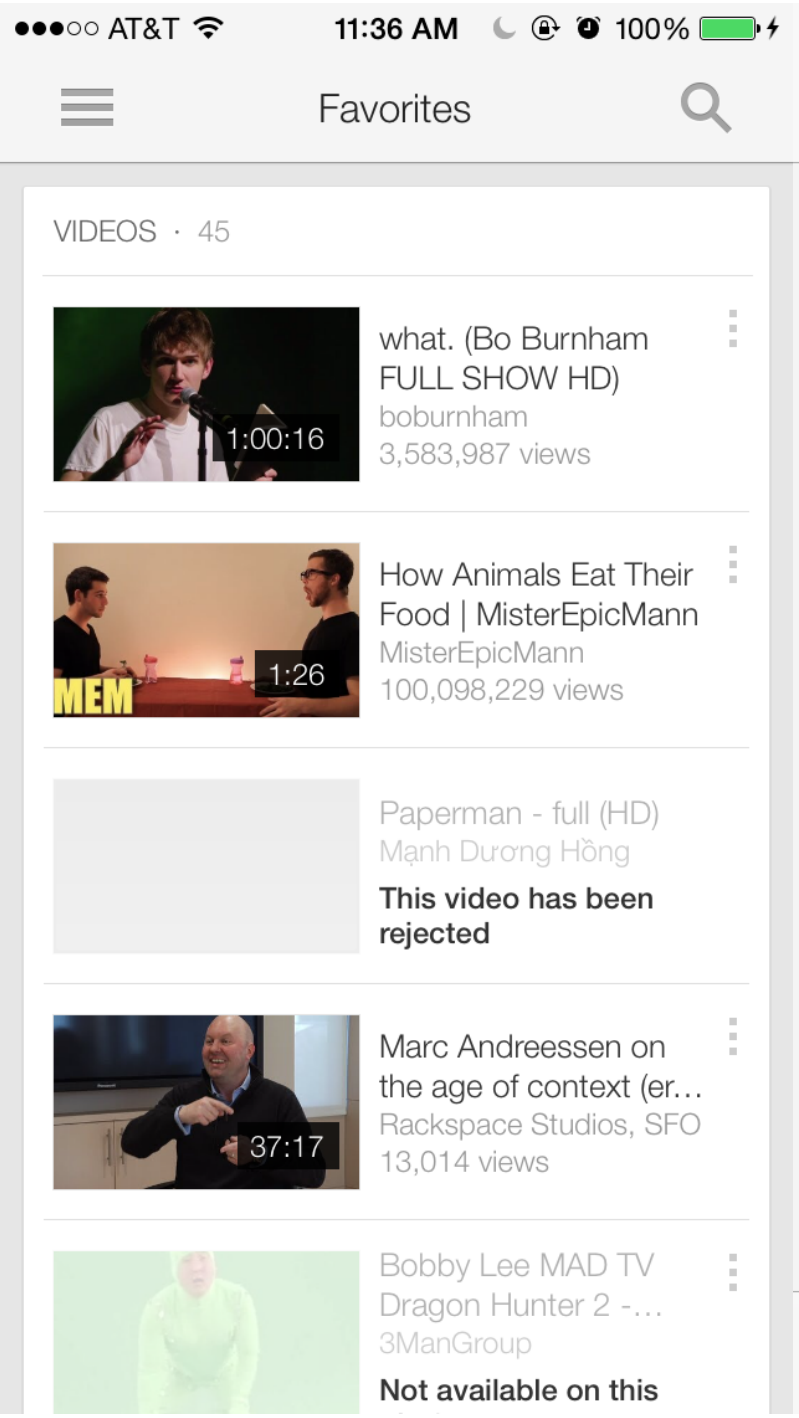
Click to use this wireframe in UXPin

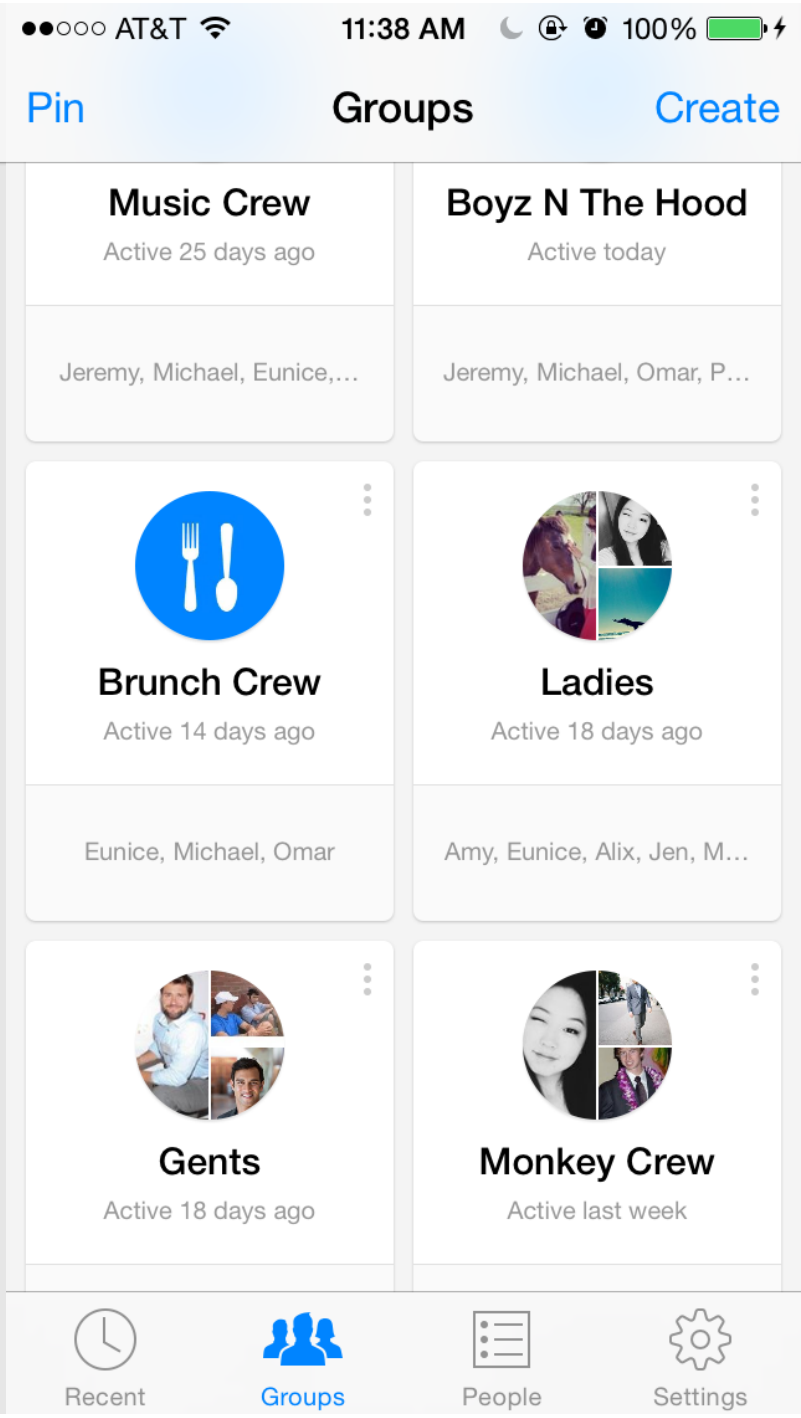


# Group Friends & Content

Examples

YouTube, Facebook Messenger



## Problem

The user wants to organize content  according to their own groupings.

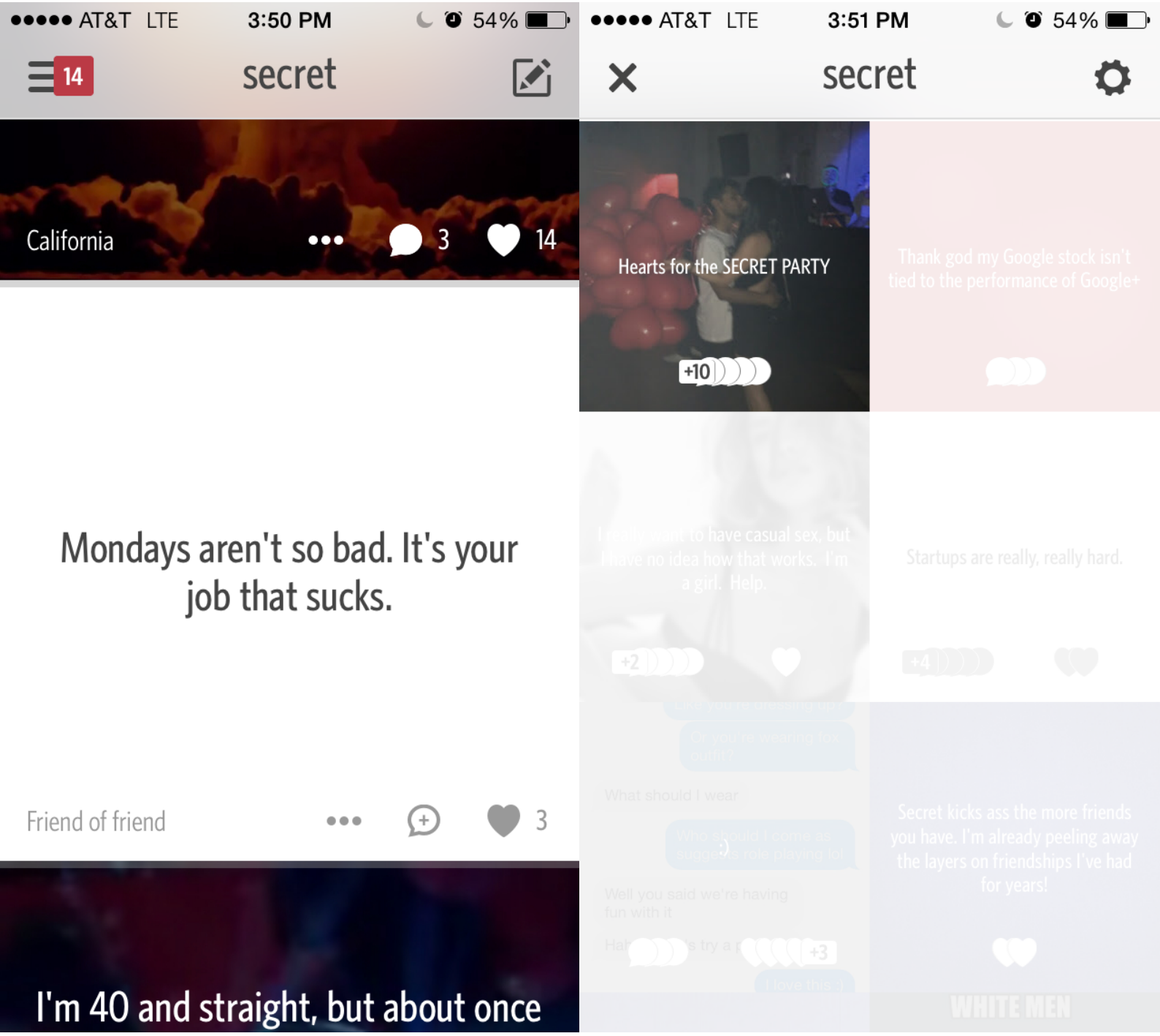
## Solution

Allow users to  sort and organize friends and followers inside the app. YouTube and Facebook Messenger allow you to group your friends and content alike. As content of all forms - including friend profiles - continues to proliferate, the ability for users to curate and organize things in a way that makes sense to them becomes more important.

# Full-Bleed Images

Example

Secret



### Problem

The user wants to focus on the content with minimal distractions.

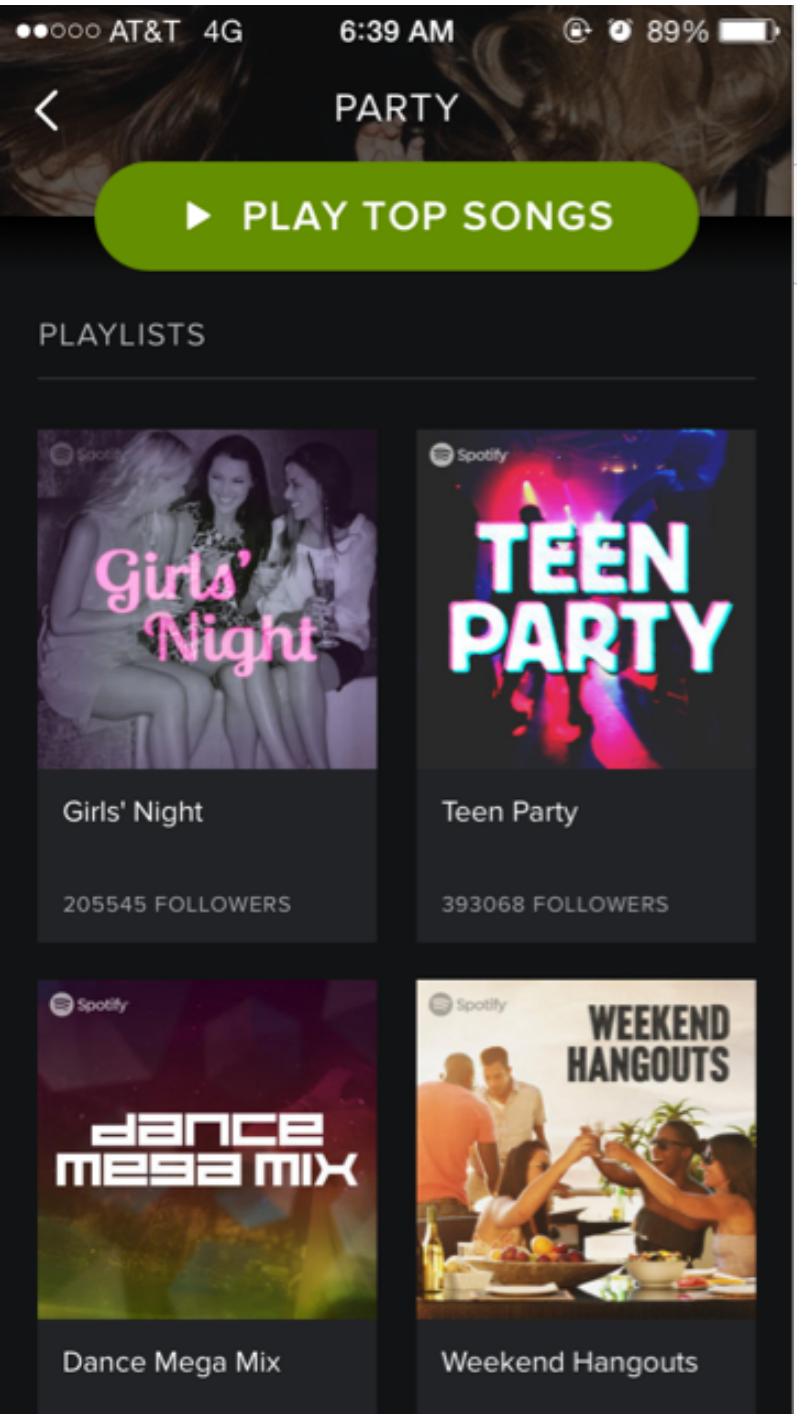
### Solution

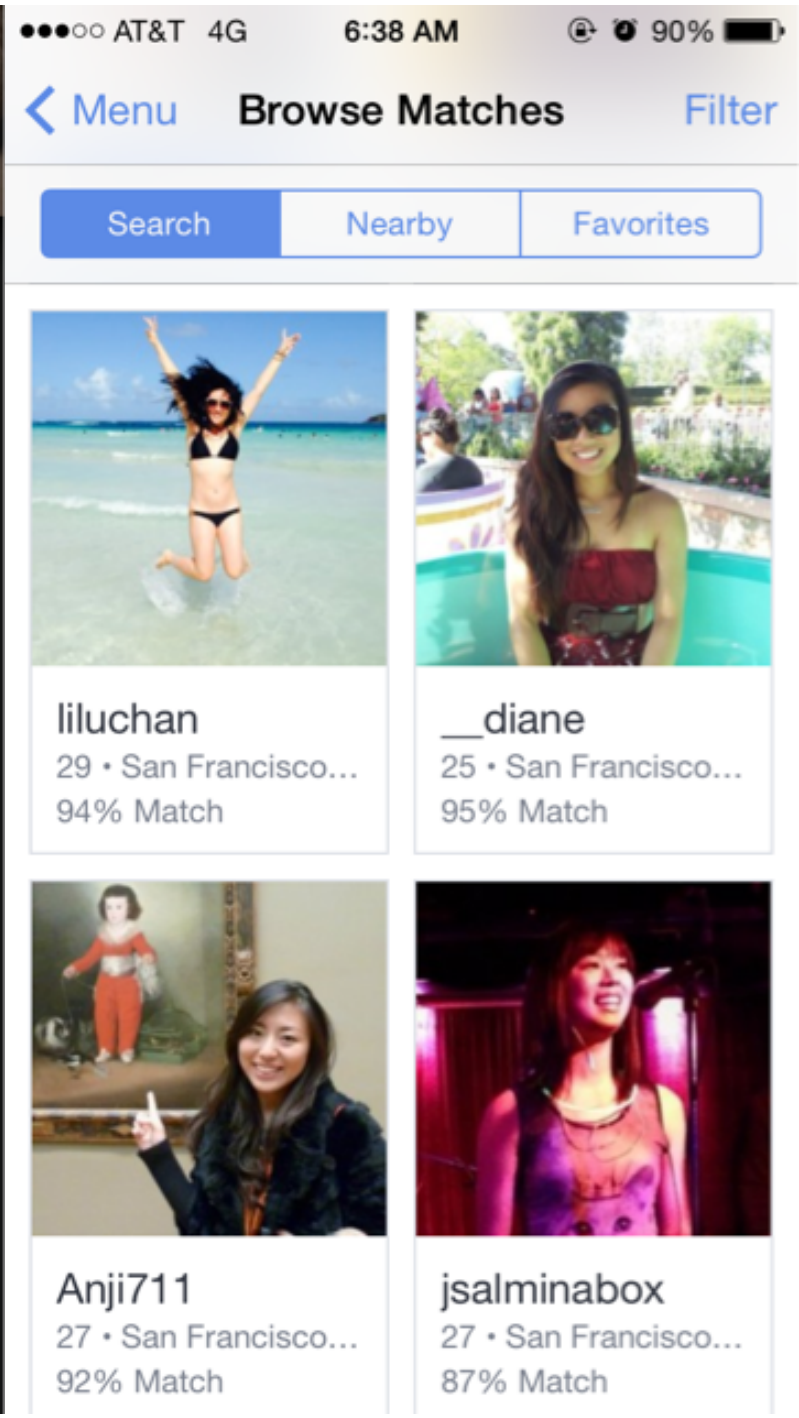
Eliminate fluff from the UI, leaving just the absolute essentials. Apps like Secret take no whitespace for granted, stacking full-bleed images on top and next to each other while layering some important information on top of them to make the best use of space. These images act as a background with relevant information over them in an overlay. This UI design pattern keeps the user extremely engaged with even less white space and distracting design details than Pinterest.

# Grids

Examples

Spotify, OKCupid



### Problem

The user wants content to be organized.

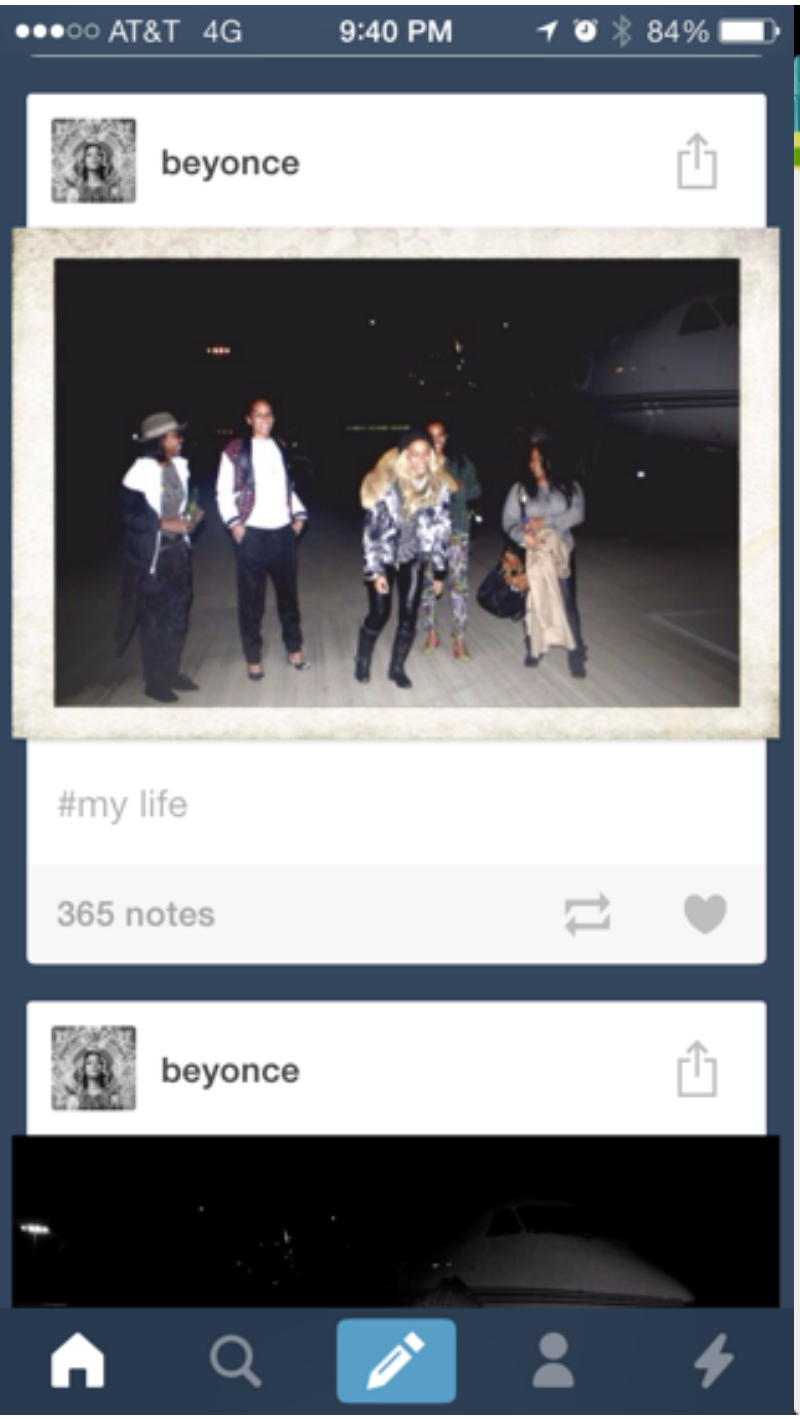
### Solution

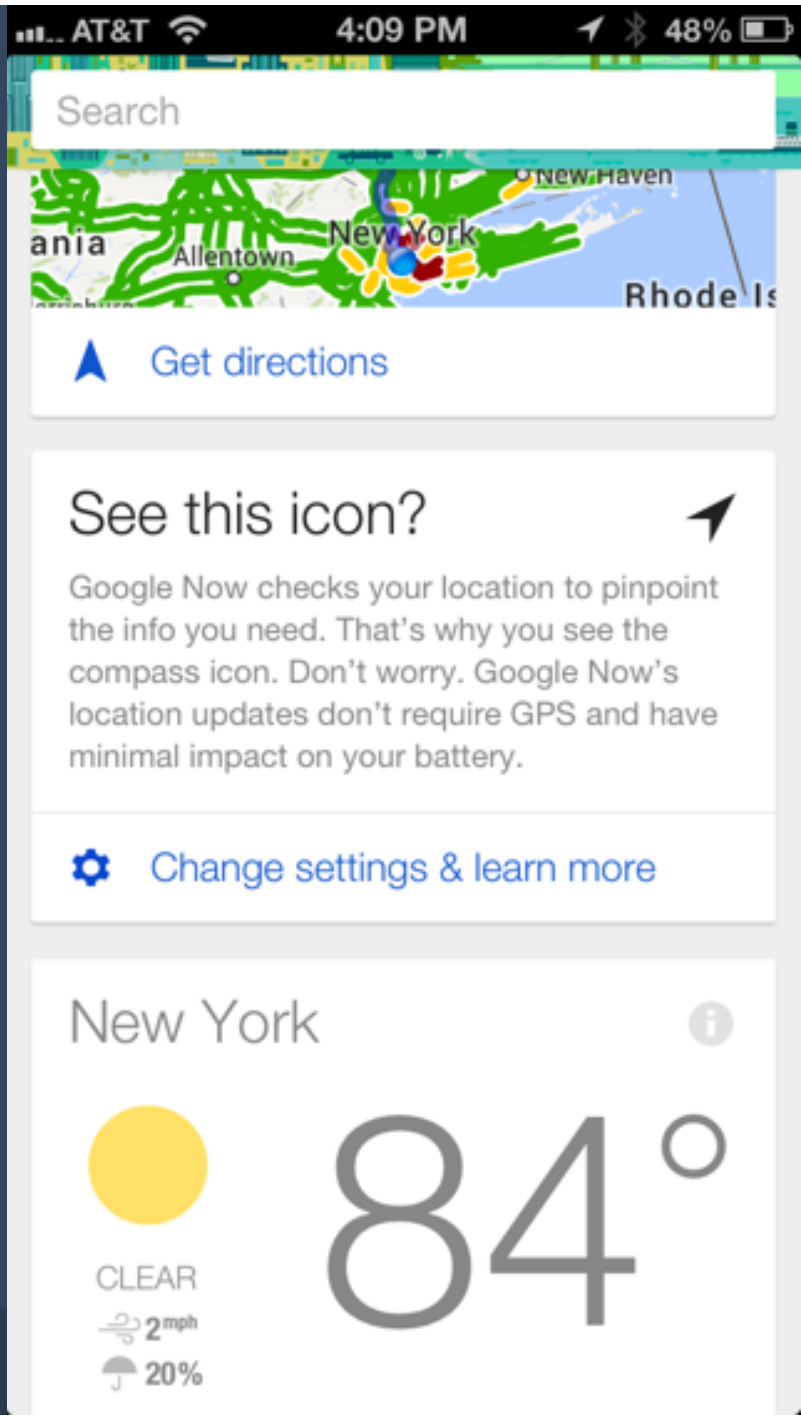
Show snippets of content in a grid. Apps like OKCupid and Spotify present all their content in a grid, effectively separating each item from the other while maintaining a structure. Grids are a great alternative to the simple list views and work extremely well for content that can be represented visually, making it much more enjoyable for users to scroll through lots of content.

# Cards

Examples

Tumblr, Google Now



## Problem

The user wants to browse through content quickly and interact with it, without the detail views cluttering up the UI.

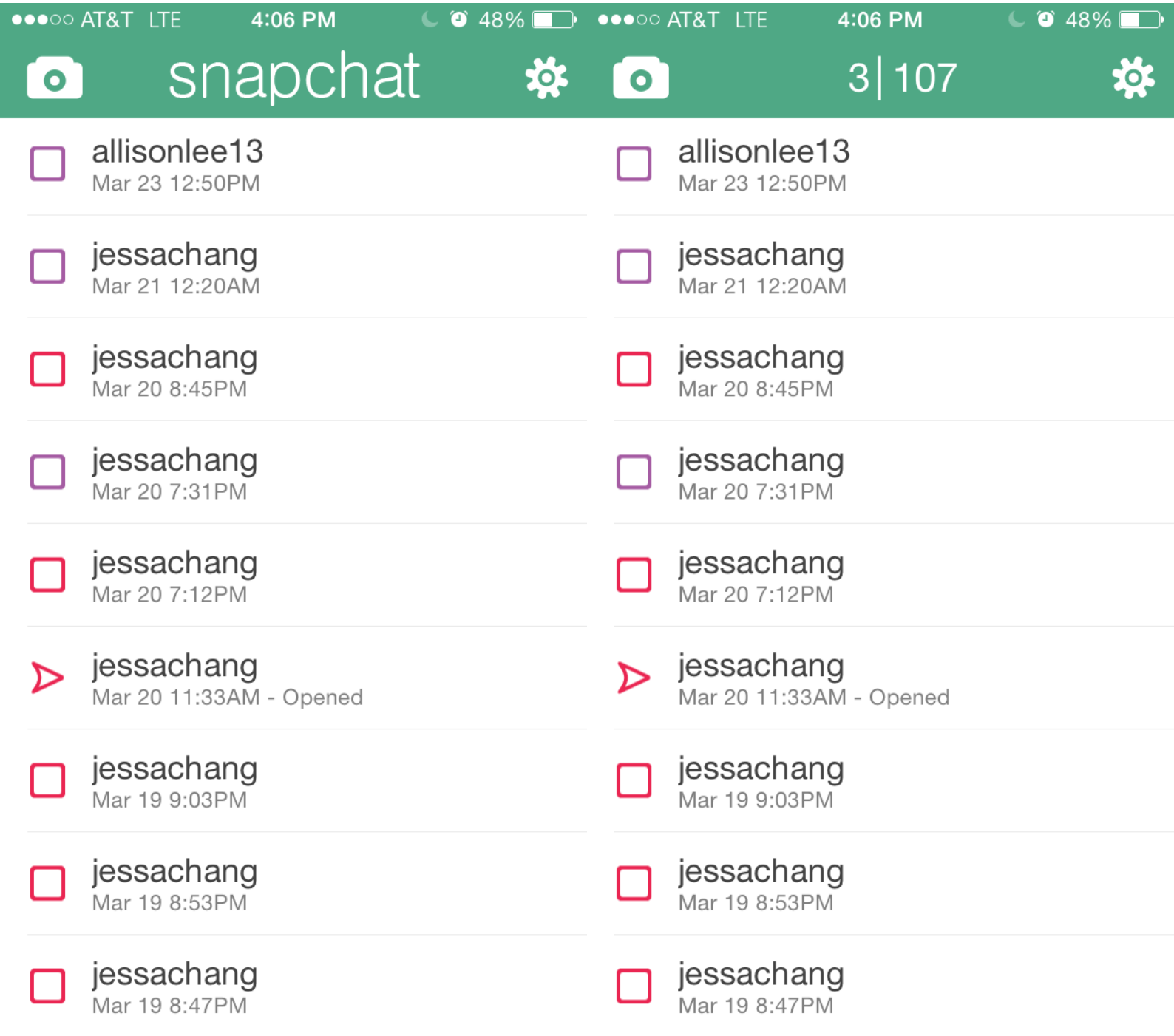
## Solution

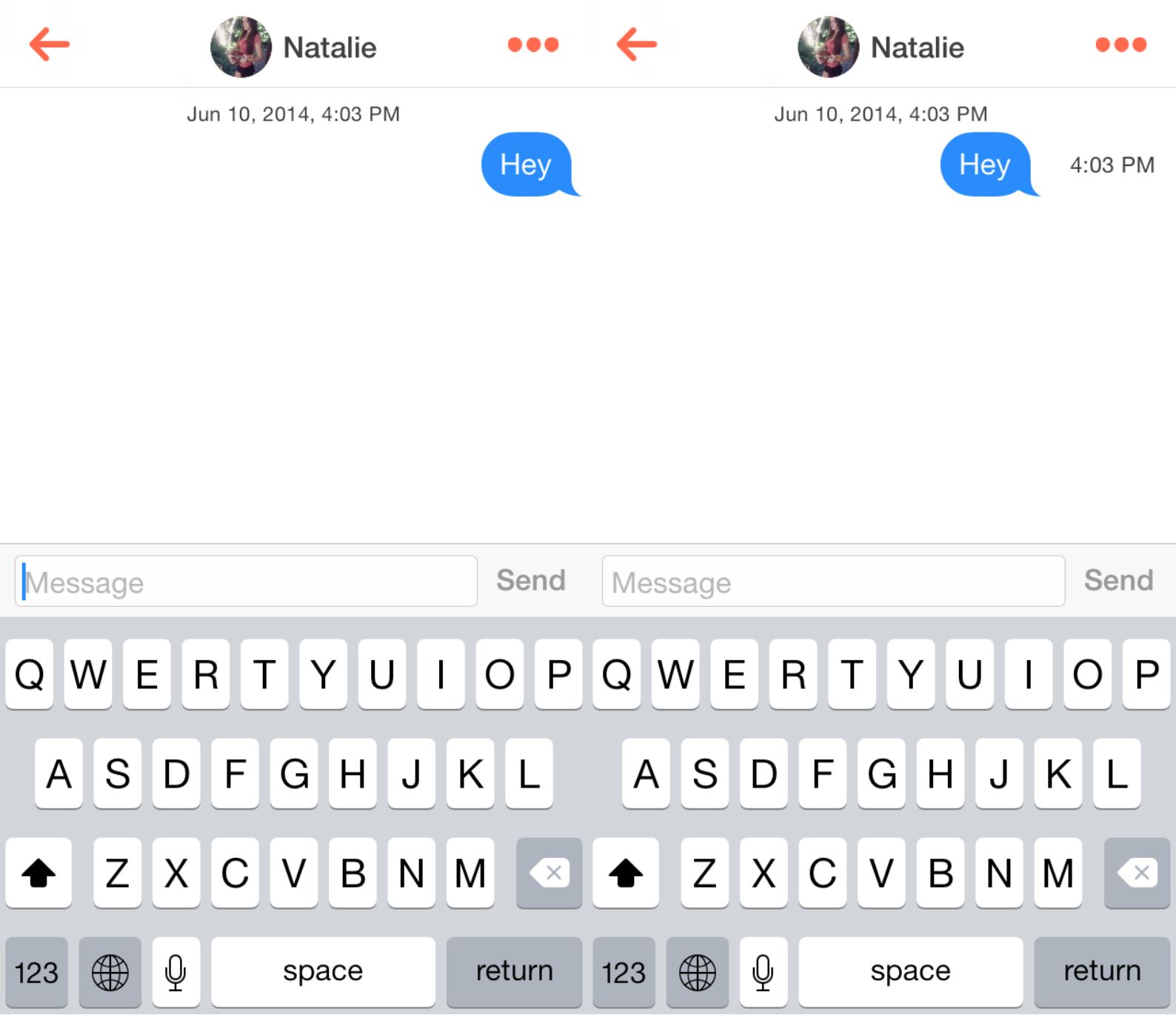
Present snippets of information in bite-sized cards that can be manipulated to show more information if the user wants it. Popularized by the likes of Pinterest to show large image thumbnails in a compact layout, we see “card” views now being implemented in a variety of apps beyond video and photo galleries on the web. This pattern works best for “modules” of data that can be viewed or manipulated individually, like posts on Tumblr or Facebook. Cards are a way to allow users to browse and discover all kinds of content in a more engaging way while accommodating responsive design trends, as well as social feed patterns.

# Hidden Information

Examples

Snapchat, Tinder






## Problem

The user wants quick access secondary information that's not usually necessary to show.

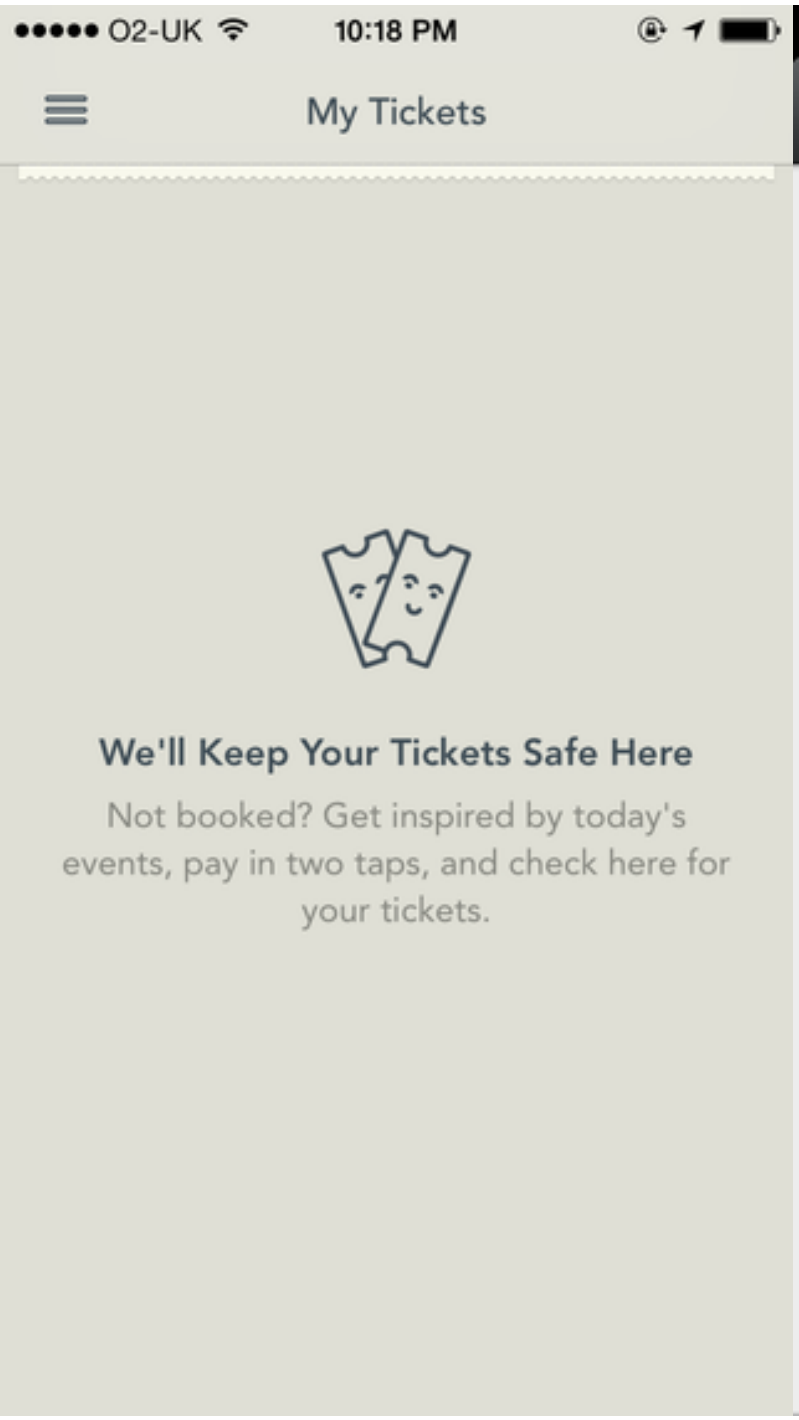
## Solution

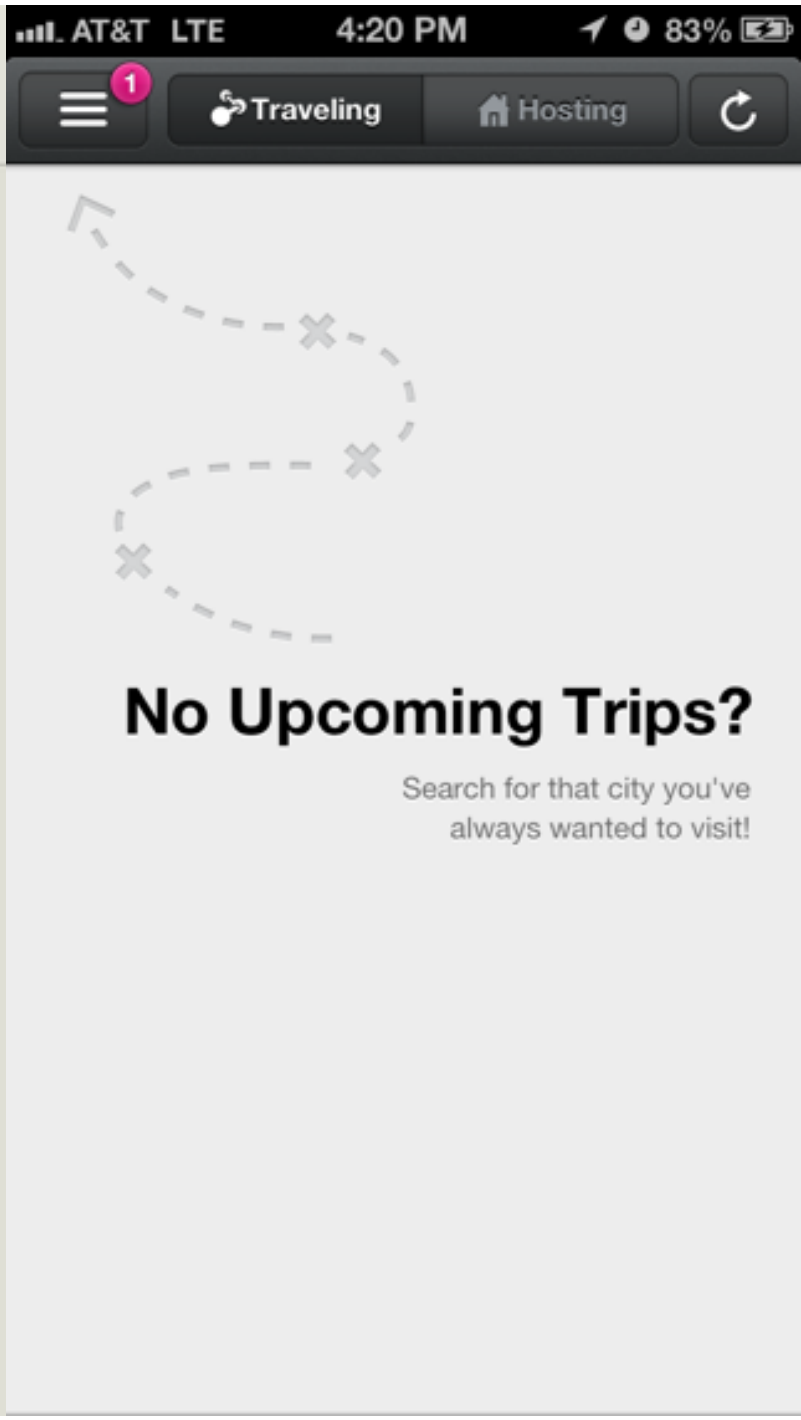
Hide contextual information that's not essential behind the UI but make it accessible for power users. Snapchat let's you see hidden information - the number of messages received and unread - by clicking on the Snapchat header. Tinder let's you see timestamping by dragging texts to the left, which is also how Apple's native Messages app works.

# Empty States

Examples

YPlan, Airbnb



### Problem

The user needs to know why a section of the application is empty and what to do next.

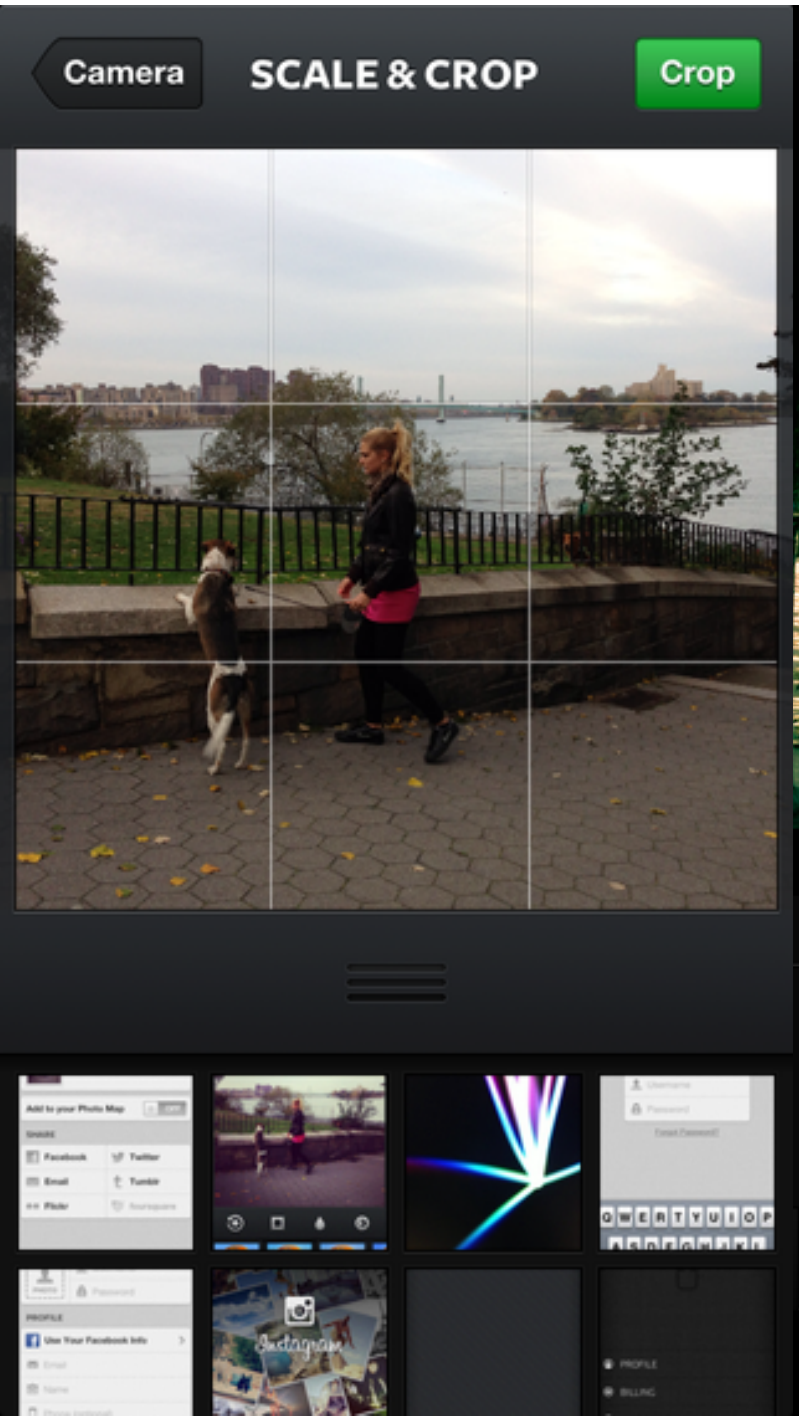
### Solution

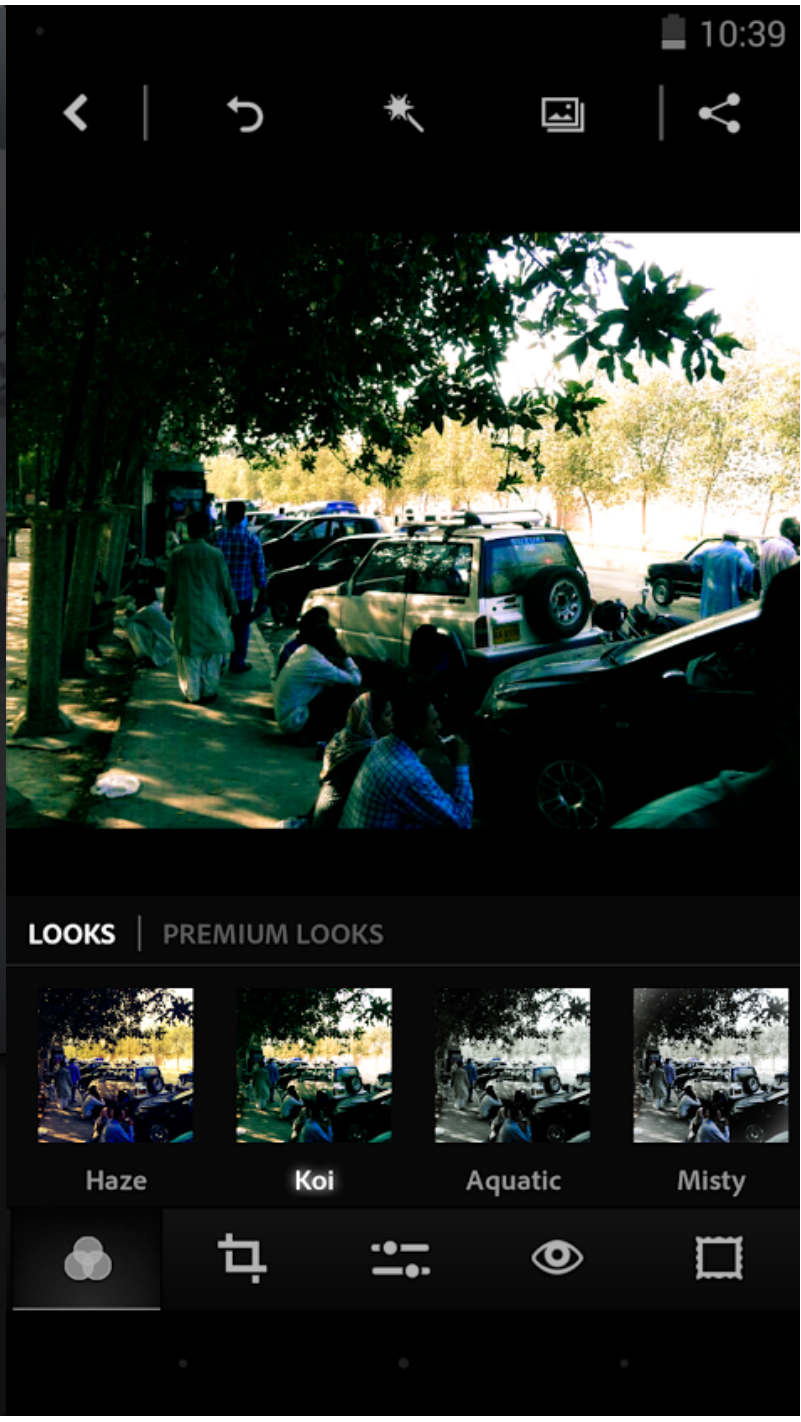
Make sure your UI provides a good first impression by designing for the “blank state,” that is the condition when there is no user data. This is the natural state of your UI and the first thing a user sees. It is also the point where many users decide whether its worth it to continue, so designing the empty state is very important. This is a great place to show some examples that will help users get started or simply to show them instructions on how to proceed.

# Direct Manipulation of Content & Data

Examples

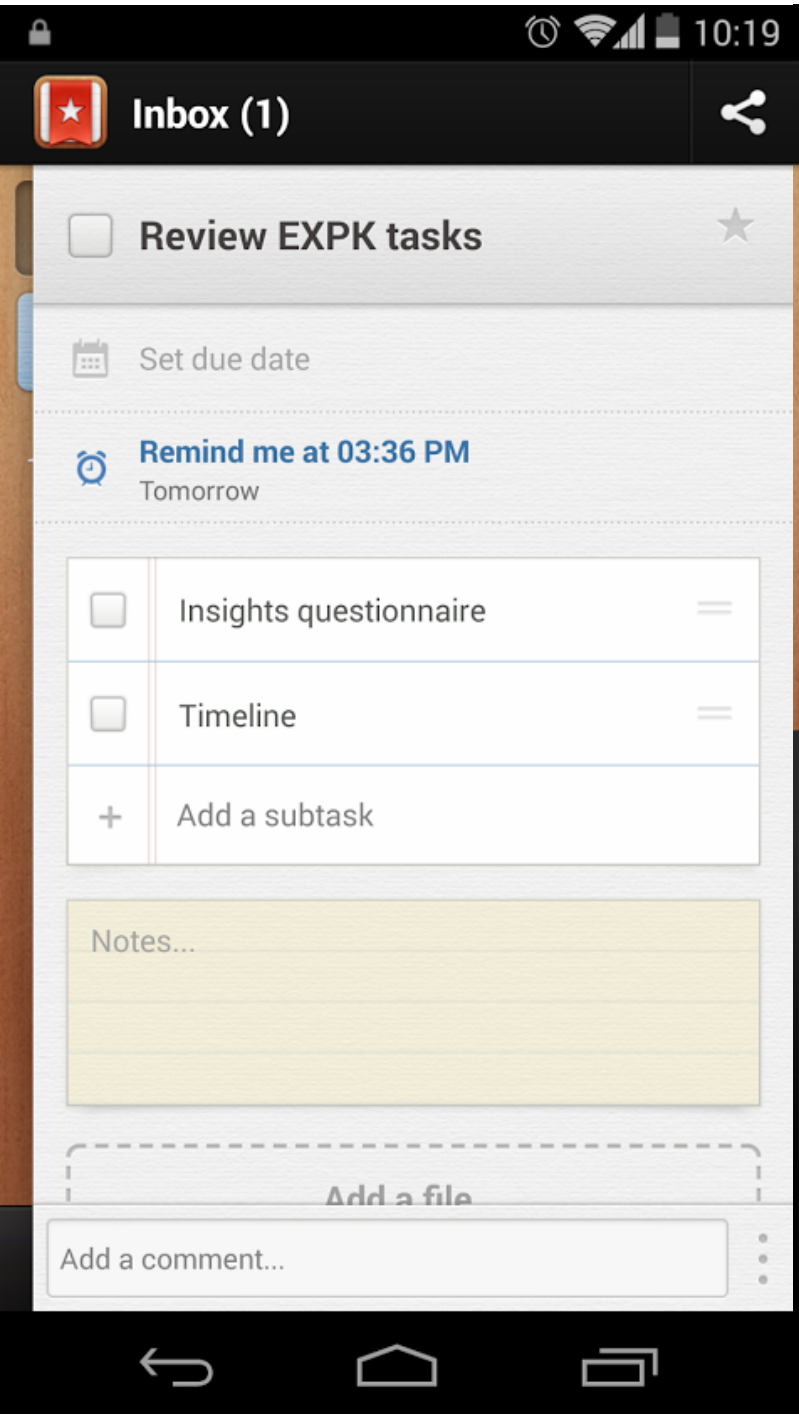
Instagram, Photoshop Express

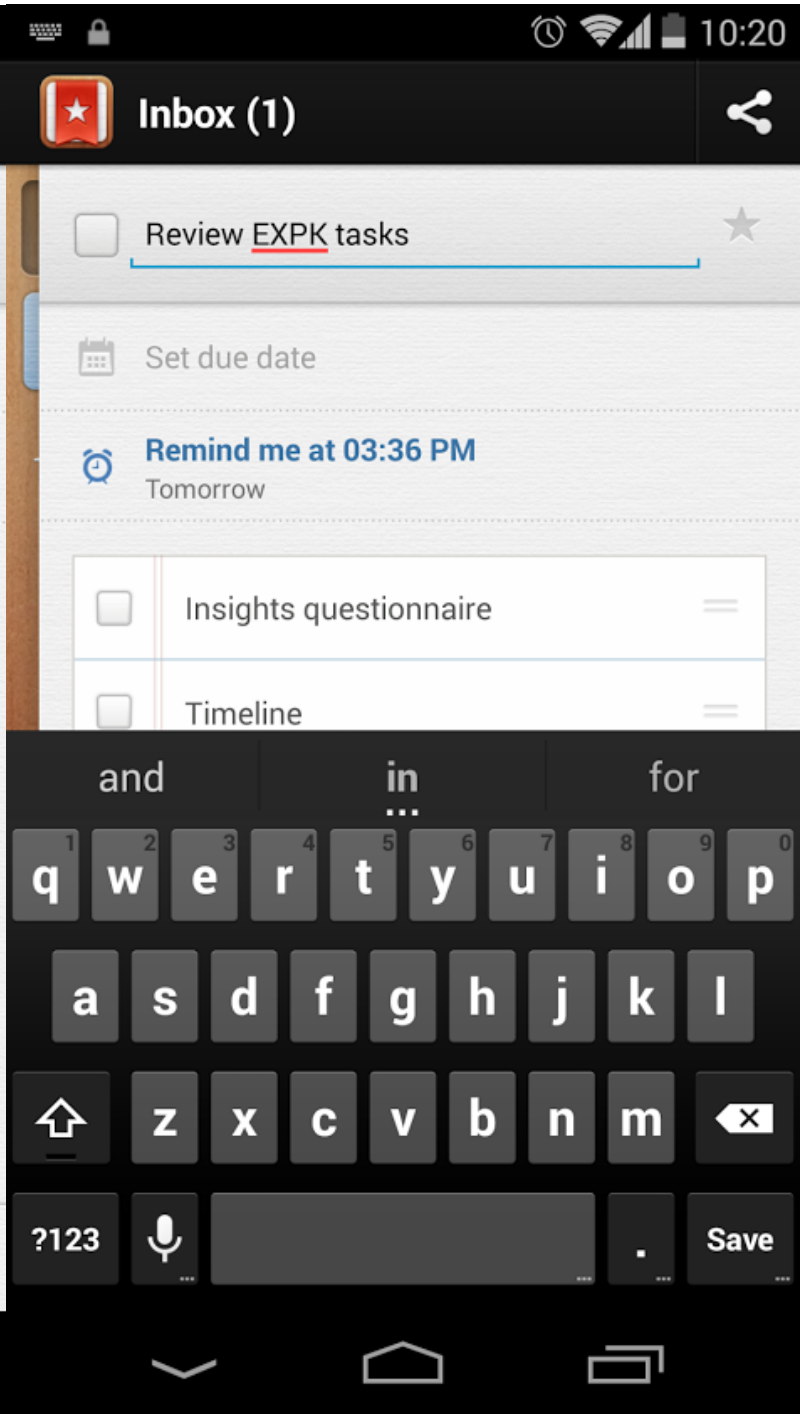


Examples

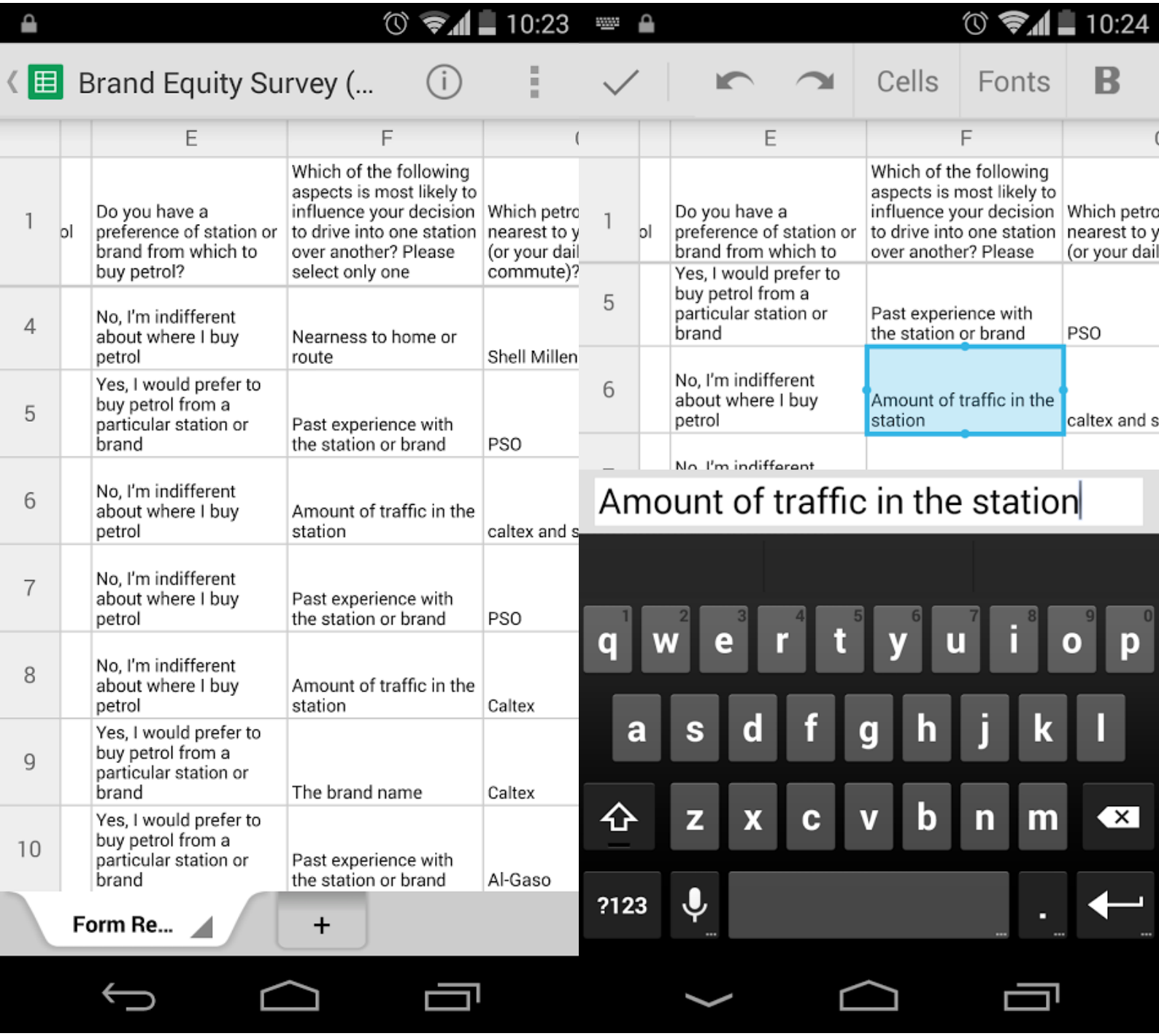
Wunderlist



Examples

Google Sheets



## Problem

The user wants to interact with entered content or data in a direct and intuitive way.

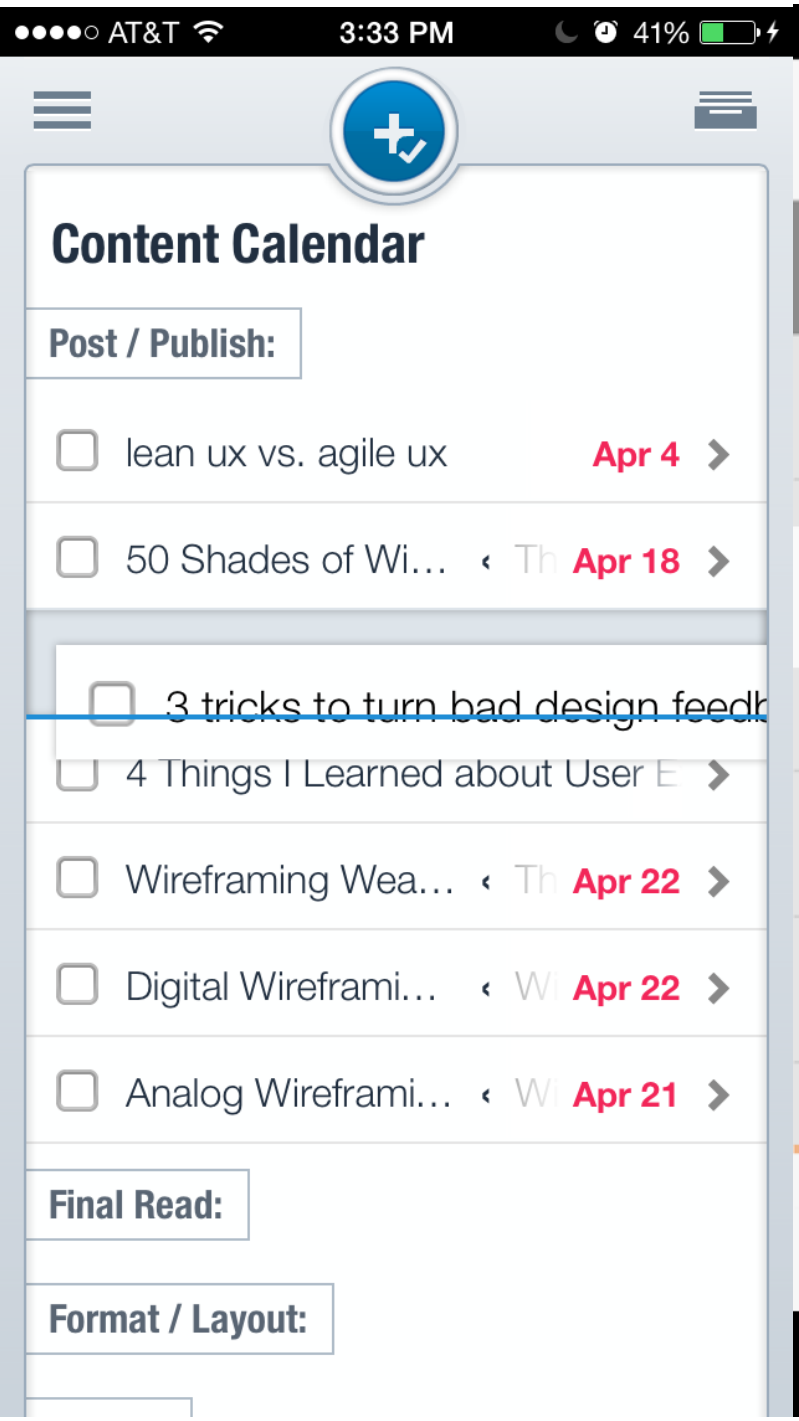
## Solution

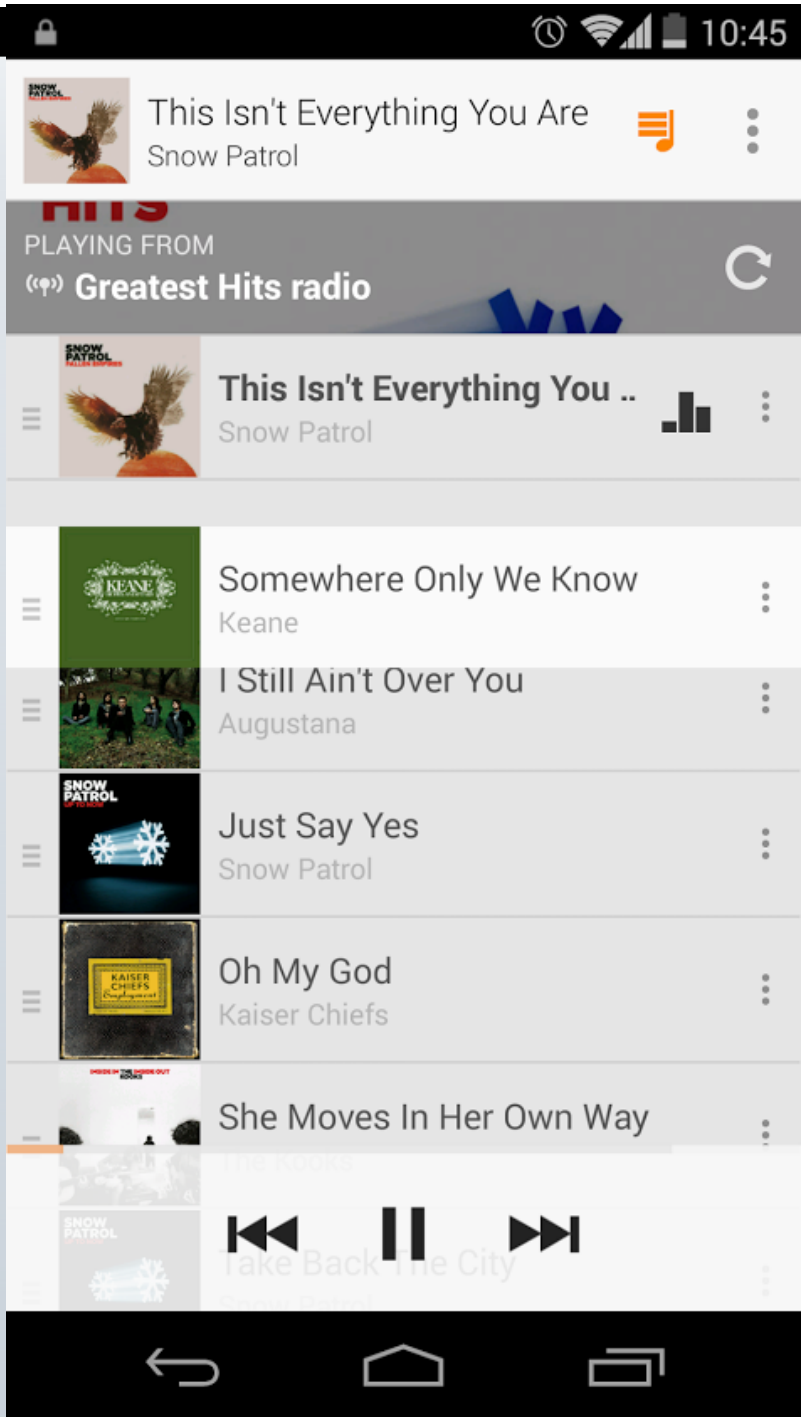
Allow for content to be edited directly without having to transition between editing or deleting modes. Letting users work with data directly on the screen can make your UI more engaging by eliminating the extra layer of interaction provided by a button or context menu. Instead of selecting the item and then toggling between individual CRUD (Create, Read, Update, Delete) states, users of Wunderlist for example can directly tap on task names to edit or delete them. Photo editing apps like Instagram and Photoshop Express also follow this pattern, allowing users to directly see the results of a filter on the selected photo instead of choosing from a list and hoping for the best. In most map apps, there's no button to zoom in or turn around, you just do it!

# Draggable Objects

Examples

Asana, Google Play Music



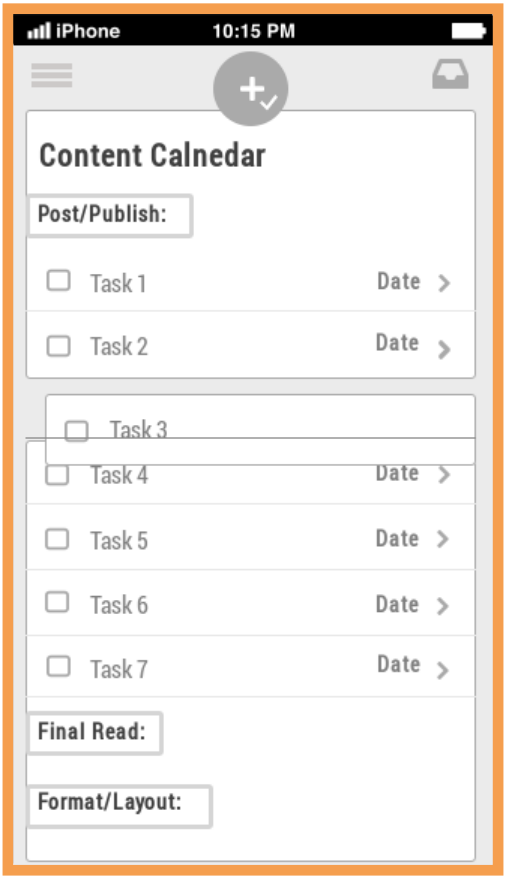



## Problem

The user wants to sort and organize items in a way that makes sense to them in the current view without pogo-sticking between master and detailed views of content.

## Solution

Asana for example let's you move tasks around by pressing-and-holding then dragging-and-dropping them wherever you want; you may want to move an item into different categories or days, and this drag and drop ability puts this in an intuitive gesture. Similarly, Google Play Music lets you drag and drop songs in a playlist to rearrange the order in which they're played.

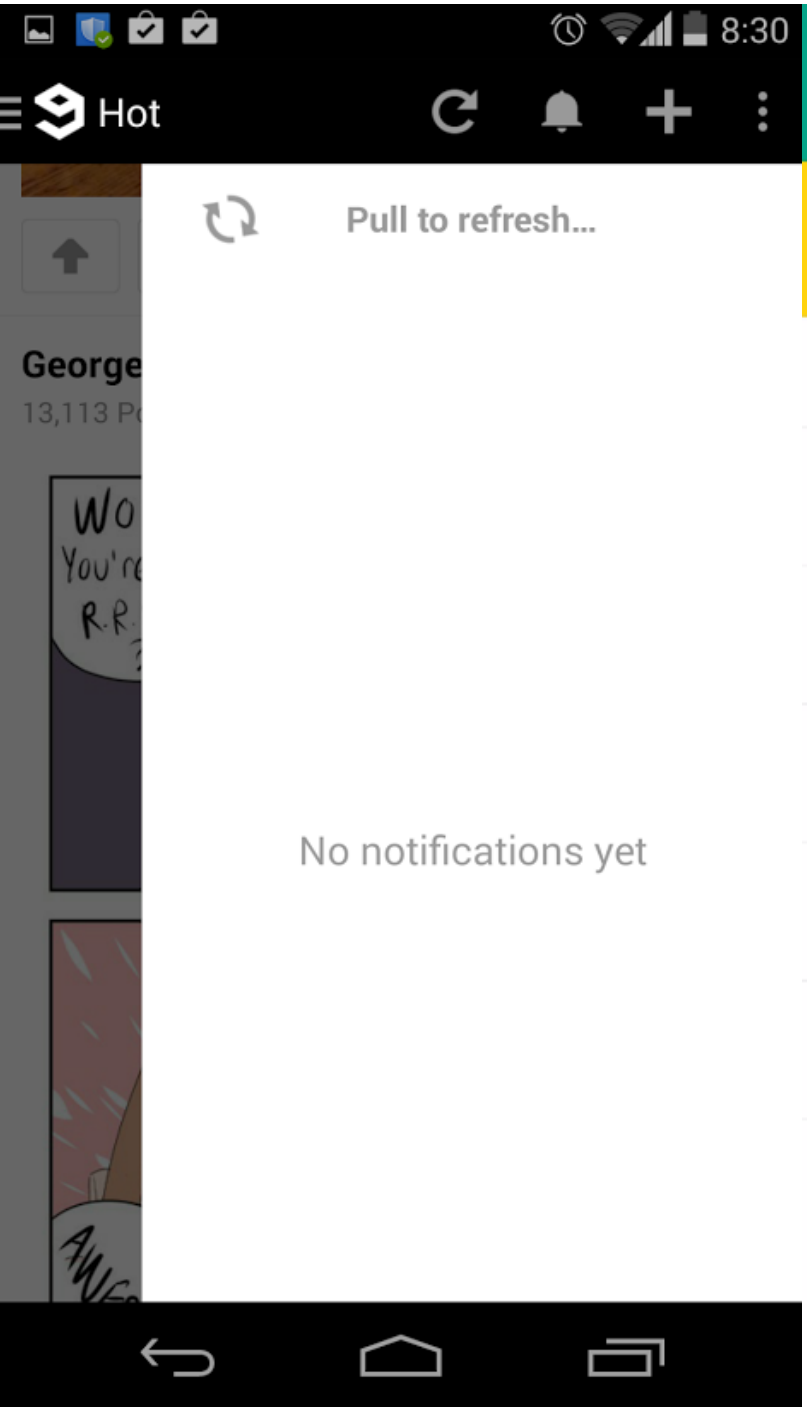


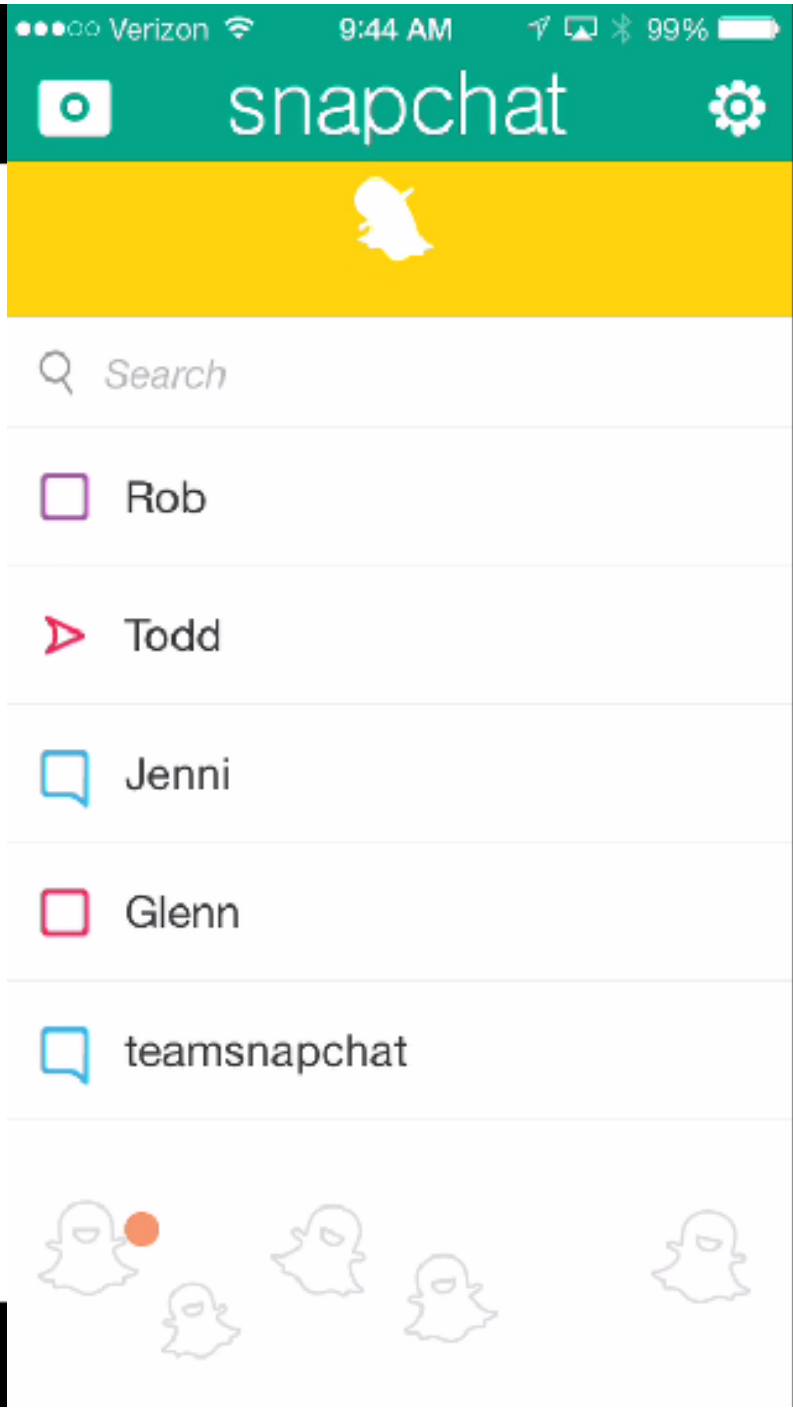

Click to use this wireframe in UXPin

# Pull to Refresh

Examples

9Gag, Snapchat



### Problem

The user wants to be able to refresh content manually.

### Solution

Instead of relying on just a refresh button, allow the main content window to be dragged down to refresh it. This is another very interactive pattern and the amount and kind of visual feedback the UI offers during and after the action is important to let users know something is happening. For example, the pull to refresh implementation in Gmail is accompanied by a horizontal colored activity indicator, while the Twitter implementation shows a circular loading animation. Snapchat shows a dancing animation. This pattern is great for lists with content that needs updating, for example a timeline or activity feed. It's an intuitive gesture to go alongside a standard button for manual refreshing, but doesn't altogether replace the automatically refreshing interfaces.

Share this ebook
with friends!

# 6

# Learn From The Best

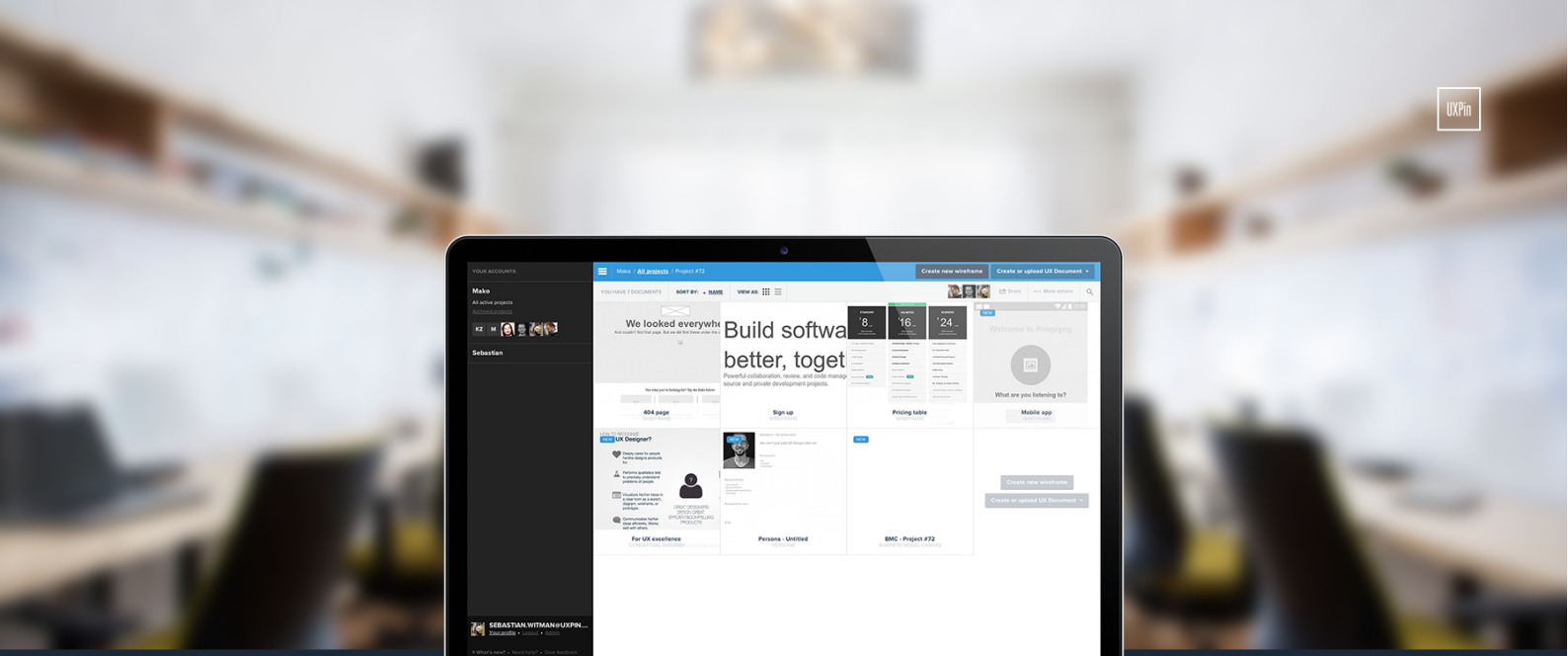
As Picasso and Steve Jobs and many other designers have said before, „Good artists borrow, great artists steal." Take these design patterns and elements, and tailor them to solve your own problems and, most importantly, those of your users.

Armies of designers, developers, and product managers have slaved over solving many of the same problems you're trying to solve today. Piggy-back off of their hard work to move faster and smarter - just don't be lazy about it. Your product, users, and team are unique, even in the slightest ways, and need a unique final solution even if the general solutions are the same as thousands of others.

Remember, there is no one-size-fits-all solution when it comes to user experience design. Many of the patterns we've covered here work extremely well when mixed and matched with each other.

We're constantly uncovering new design insights on our blog and design library so stay tuned for more. In the meantime, UXPin's current free e-books - Web Design Trends, Mobile Design Trends, and The Guide to Wireframing - and, of course, The Design Pattern and Wireframing Libraries Guide have a bunch of examples as well.

Good luck!



- ✔ Rich UI element libraries, including iOS, Windows Phone and Android for mobile design
- ✔ Built-in UX Patterns Library with inspiring solutions for your designs
- ✔ Responsive design ready

UXPin

www.uxpin.com