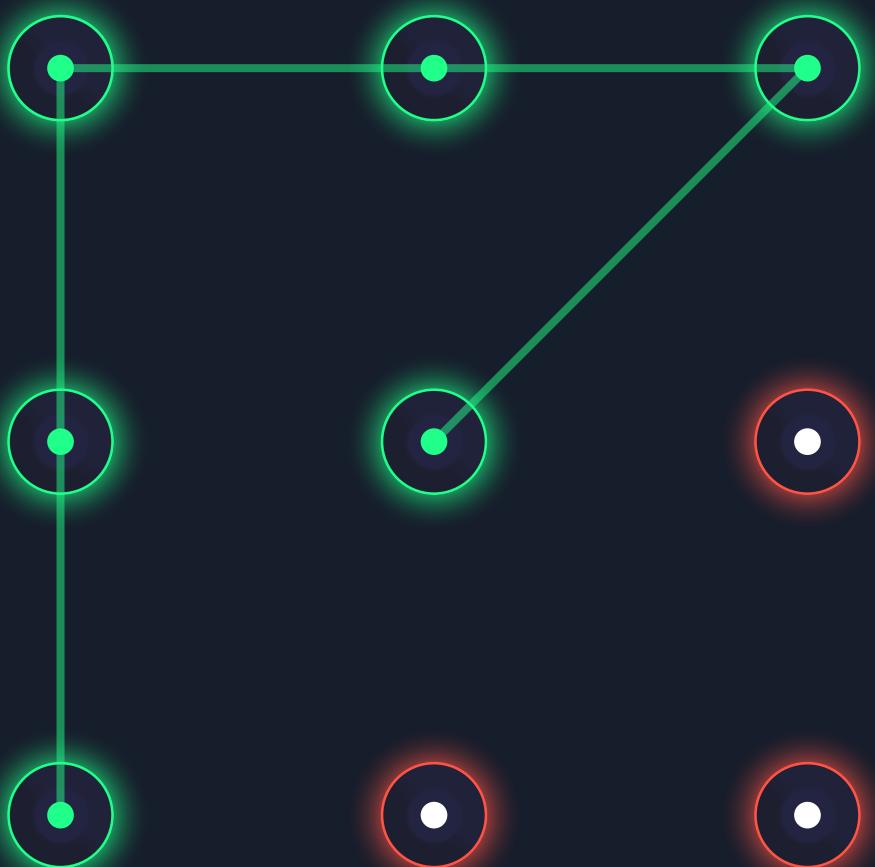


UXPin

Mobile Design Book of Trends

2015 & 2016





Mobile Design Book of Trends

2015 & 2016

Copyright © 2015 by UXPin Inc.

All rights reserved. No part of this publication text may be uploaded
or posted online without the prior written permission of the publisher.

For permission requests, write to the publisher, addressed “Attention:
Permissions Request,” to hello@uxpin.com.

Index

Clever Gestures	8
Primer on Common Gestures	9
Connect Gestures and Animation	11
Apple and 3D Touch	14
Material Design Refines Gesture Actions	16
Complex Gestures	17
Clever Gestures Are More Than Actions	19
What's Next?	21
Takeaways	24
10 Resources and Tools	24
Layered Mobile Interfaces	26
Let's Talk About Material Design	27
Understanding the “Tactile Surface”	28
Material is Made for Adaptive Design	30
Material and Other Mobile Design Trends	32
What's Next?	36
Takeaway	40
11 Excellent Resources and Tools	40

Meaningful Typography	42
Typeface Selection is Important	42
Give Words Some Space	46
Pay Particular Attention to Alignment	49
Create Contrast	51
Design Hierarchy (But Maybe Fewer Levels)	54
Don't Force It	57
What's Next?	58
10 Resources & Tools	62
Flat Mobile Design	63
First, a Flat Design Primer	64
Flat Was Made for Mobile	67
The Evolution of Mobile Flat Design	70
Flat and Other Design Techniques	72
Benefits of Flat Design for Mobile Interfaces	75
What's Next?	76
10 Excellent Resources	81

Delightful Micro-interactions	82
What Are Micro-interactions?	83
What Do Micro-interactions Really Do?	85
Four Elements of a Micro-interaction	86
Are You Considering Feedback?	87
Designing Details is at the Heart of Micro-interaction	89
Ways to Incorporate Micro-interactions into the Design	91
What's Next?	94
Takeaways	98
10 Resources and Tools	99
Bite-Sized Cards	100
Card-Style Interfaces 101	101
Two Distinct Uses for Card-Style Design	103
Cards Work Well With Content	106
Cards and Behavior Patterns	108
Cards Are Easier to Organize	113
Thinking Aesthetically	117
What's Next for Mobile Card Design?	118
10 Resources and Tools	123



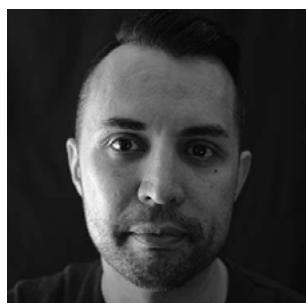
Jerry Cao is a content strategist at UXPin where he gets to put his overly active imagination to paper every day. In a past life, he developed content strategies for clients at Brafton and worked in traditional advertising at DDB San Francisco. In his spare time he enjoys playing electric guitar, watching foreign horror films, and expanding his knowledge of random facts.

[Follow me on Twitter](#)



Carrie Cousins has more than 10 years experience in the media industry, including design, editing, and writing for print and on-line publications. Carrie is also a sports fanatic and spends way too much time planning football and basketball trips and obsessing over stats.

[Follow me on Twitter](#)



Ryan Thomas Riddle is a UX Content Strategist at UXPin. He is an award-winning writer for his work as a mild-mannered reporter for the Bay Area newspaper, The Daily Post. His previous work has appeared in the San Francisco Bay Guardian. Formerly, he worked for ZURB, a digital product design company, as their Lead Editor. You can read all about his adventures through time and space on Twitter, @ryantriddle. You can find more of his writing on his [personal website](#).

Clever Gestures

Gestures are the new clicks.

Every app, game or tool you open on your phone must include a swipe, tap or pinch to function. These hand gestures are the secret to making great mobile apps work. And there's a lot that goes into it.

With clicks, designers and developers really only had to think about where they wanted the action to appear on the screen. With gestures, you have to consider the type of physical action, its location on the screen, and whether users can intuitively find and touch it.



Gestures can help pave the way for more beautiful interfaces because many of these actions can be hidden within the interface. Gestures enable speed in user action and are a comfortable way for users to interact with devices of varying shapes and sizes.

With that, let's dive deep into successful gesture techniques as expressed by some of today's popular apps.

Primer on Common Gestures

Gestures are any physical movement by the user to activate a specific control within the design. Gestures are primarily hand movements, but can also include shaking, tilting or moving the device.

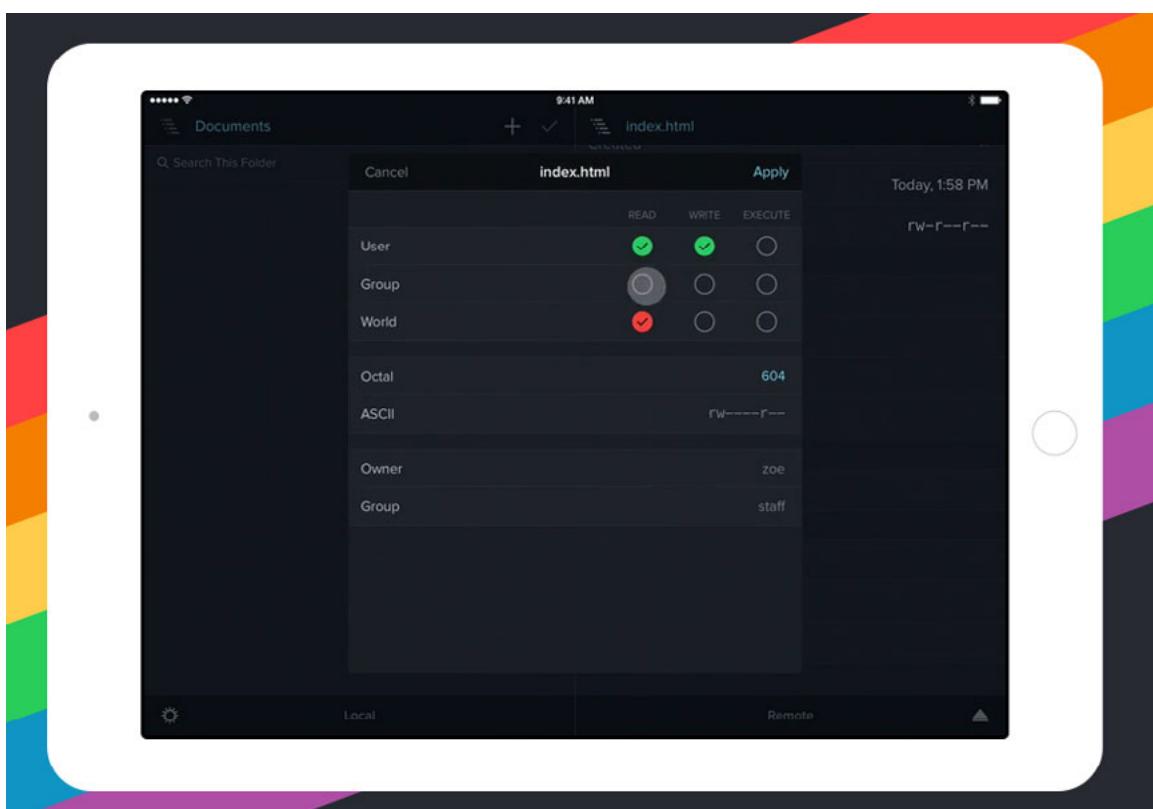


Photo credit: [Transmit](#)

The [definitive guide on gestures](#), created by Craig Villamor, Dan Willis, and Luke Wroblewski, details dozens of actions and how they work. The guide was compiled from a study of people using mobile devices and performing specific tasks.

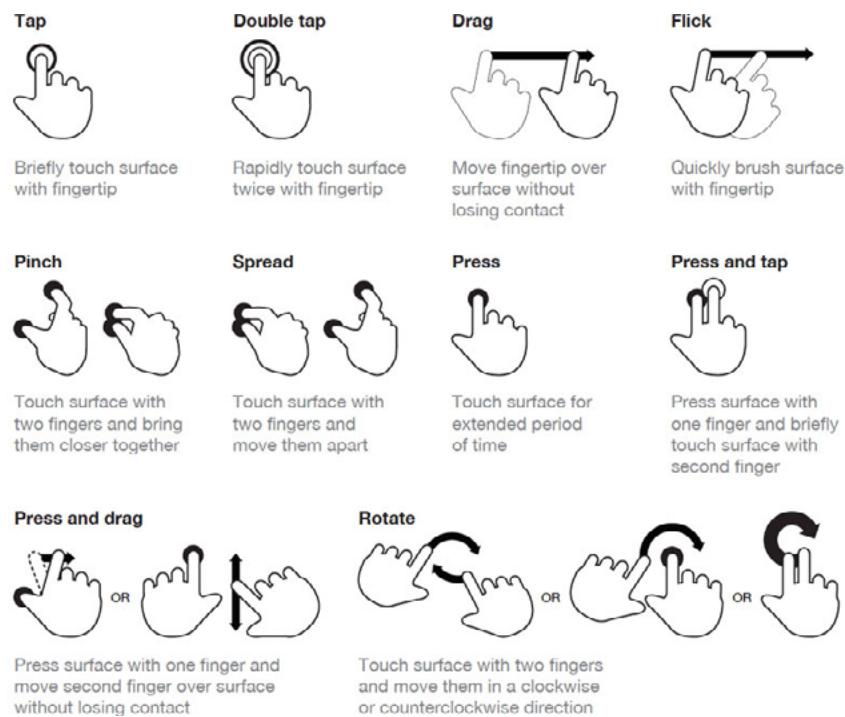


Photo credit: Touch Gesture Reference Guide. Creative Commons.

These are some of the most common user gestures:

- **Tap:** Touch surface briefly
- **Double tap:** Touch surface with two quick motions (often to zoom)
- **Drag:** Move along surface without breaking contact
- **Pinch/spread:** Touch surface with two fingers to move in (pinch) or out (spread)
- **Press:** Touch surface and hold
- **Flick:** Scrolls quickly

Each of these gestures needs to be within reach for the user as well. And we needn't be afraid of gestures now that [larger phones seem to be selling more and more](#). One thing to keep in mind when designing for gestures is thumb placement – something that's known as the "[thumb zone](#)." And while it's not always the case, most of us hold our phones with [one hand and use one thumb to navigate](#). The most common actions happen with a thumb movement – that's about how much space you have to work with across the device.

The size of "buttons," for lack of a better description of gesture-based controls, is vitally important. Each button needs to be large enough to activate with ease, regardless of finger size. (Many designers often reference this as the "fat finger" rule.)

According to this rule, gesture targets should be at least 44 points wide and tall (our [own experiment](#) showed that 30 points is the absolute minimum). With bigger screens, the fat finger rule expands to 70 points, which also makes thumb reach that much easier. Moreover, buttons and swipe gestures are designed to fill entire portions of the screen, such as a button that can be touched anywhere across the design, from left to right.

Connect Gestures and Animation

Gestures are invariably linked to animations in mobile design. Animation is the primary way that user interfaces signal users to complete an action or task.

The five primary functions of animation, from the free e-book *Interaction Design and Complex Animations* include:

1. Animated notifications
2. Revealing information
3. Highlighting content
4. Collapsing forms and menus
5. Scrolling

Think of how many of these animations you use in the apps you touch daily. Here's a short case study using the [Dark Sky weather app](#).



Photo credit: Dark Sky

Dark Sky uses tiny gestures and animations throughout the app to create an enjoyable user experience that extends beyond how warm or cool the day might be.

1. **Animated notifications:** Is it going to rain today? A light animation shows when the rain is expected to fall while another shows how heavy the precipitation will be. A simple swipe will get you more details.
2. **Revealing information:** Tap-based gestures will allow you to expand the daily weather forecasts so you can see detailed hourly information.
3. **Highlighting content:** The app has multiple screens, which you can access with a swipe left or right..
4. **Collapsing forms and menus:** You also can collapse elements so that you can see additional information. For instance, you can take the menu at the top, collapse the current view and change location.
5. **Scrolling:** The globe in the radar view spins on a realistic axis so you can look at the weather anywhere in the world. Scroll left to right on the slider at the bottom of the screen to change the time as well.

Animations and gestures go hand-in-hand (pun fully intended). Without animations, users wouldn't have feedback informing them if they've successfully completed a gestural action. So it's good to get in the mindset of thinking of gestures as complementary elements rather than separate ones.

Of course, with new technologies, how we get feedback from gestures will change. For instance...

Apple and 3D Touch

Gestures are so important that the major mobile platforms – Apple and Android – have developer guidelines for them. Meanwhile, Apple introduced a new type of gesture control with the release of the iPhone 6s in late 2015 called [3D Touch](#).

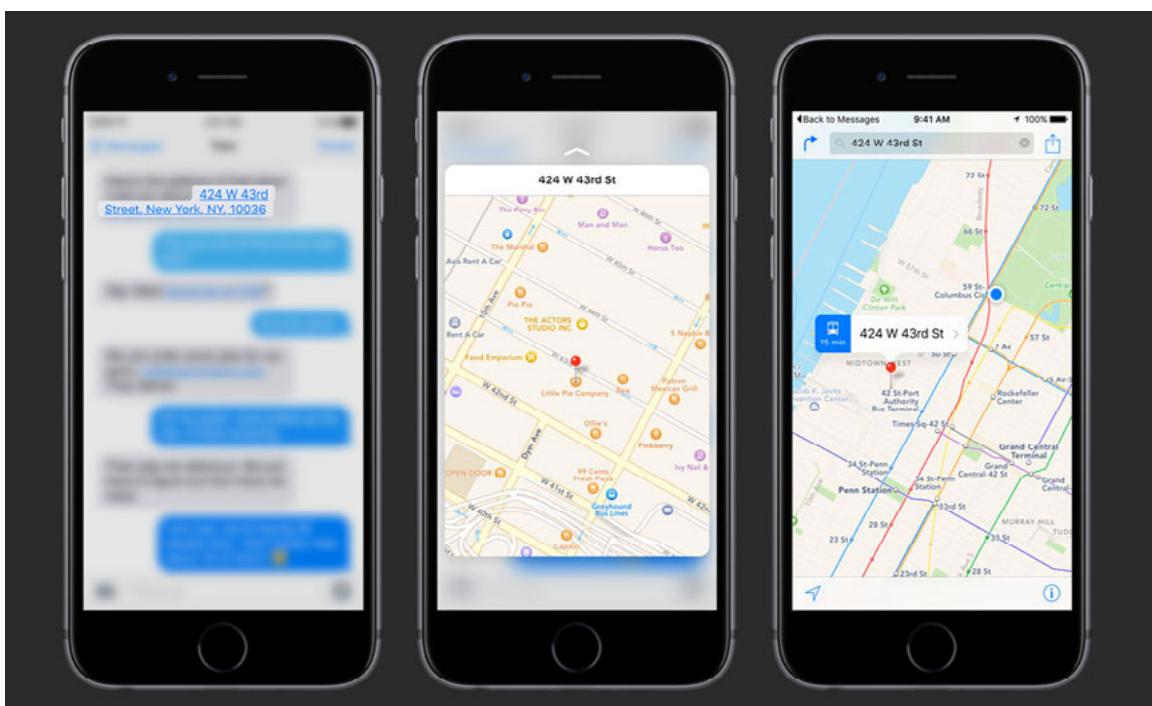


Photo credit: [Apple](#)

Apple's 3D Touch – what's also been previously called Force Touch – takes gestures to the next level. The device will sense the amount of pressure on the screen and relates that to gestures such as tap, swipe and pinch. The major difference in this gesture functionality is part of the layered interface, where pressure prompts a preview action before you actually go on to the next screen.

In the map example above from Apple, someone sends you an address via text. It shows as a link – press lightly and the location pops up.

When you remove pressure, the pop up disappears. But if you apply even more pressure, the full location link opens in a map. There's even a physical "**bump**" of feedback. This is the perfect example of how elements across apps should work seamlessly as a single gesture-based interface.

This intuitive use of gestures perfectly meets user expectations because it works with little or no explanation.

In addition to new features, [Apple iOS Human Interface Guidelines](#) set rules for thinking about gestures and how to use them.

- Avoid associating different actions with standard gestures.
- Avoid creating custom gestures that invoke the same actions as standard gestures.
- Use complex gestures as shortcuts to expedite a task, not as the only way to perform it.
- Avoid defining new gestures unless your app is a game.
- In a regular environment, consider using multi-finger gestures.

While this will change how we interact with our iPhones, there certainly will be a period of adjustment. So it's best to take that into consideration when designing 3D touch gestures because the same gesture won't apply to older phones.

Material Design Refines Gesture Actions

While Apple is working to help us think differently about how gestures make things work, Google is clarifying the mechanics and actions associated with them. In the ever-evolving [Material Design documentation](#), Google outlines the exact details of how these actions should work.

The guidelines go a long way to create a set of rules for actions to simplify usage. One of the common flaws of gestures is that the same action accomplishes multiple things. The documentation thoroughly explains how to use each gesture as a toggle so that you can have them do multiple things.

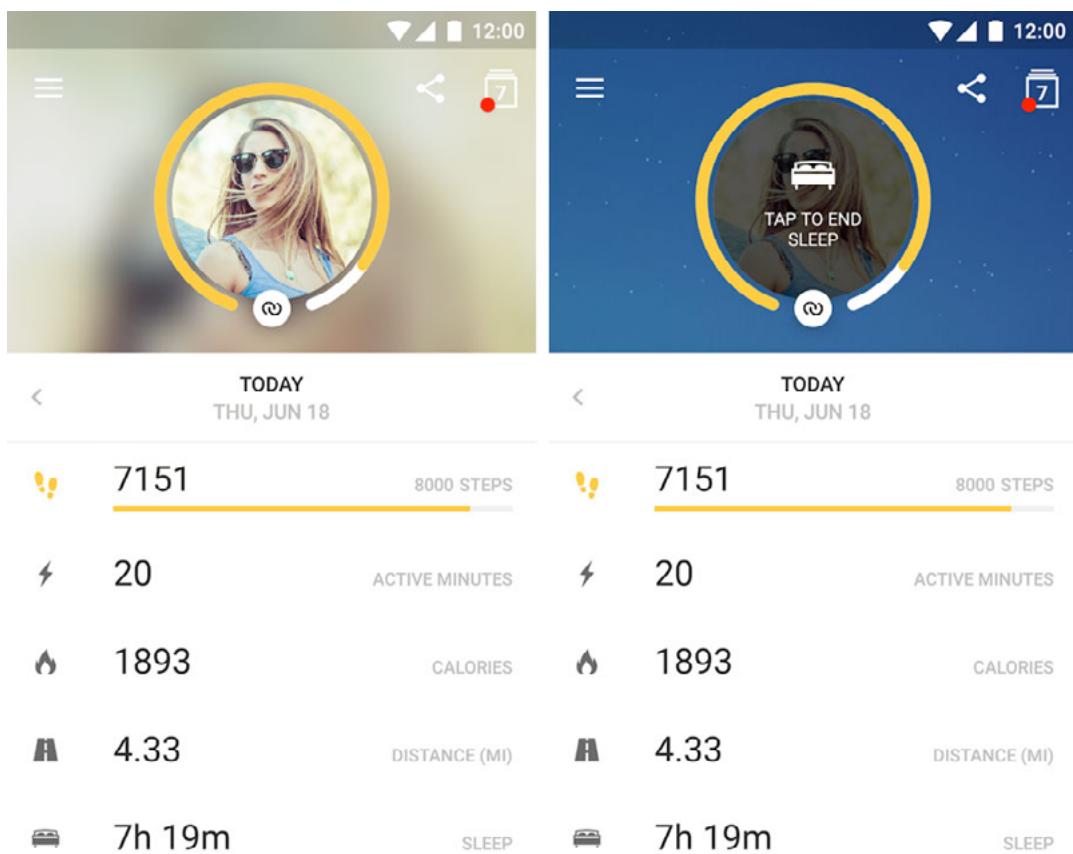


Photo credit: [Runtastic](#)

When it comes to gestures, Material Design covers these three: drag, swipe and fling.

- Drag is a more controlled gesture used in conjunction with a touch target.
- Swipe is a speedy gesture that doesn't require a touch target.
- Fling has no touch target at all and is a snappier gesture.

These common gestures also incorporate velocity as a critical component. Depending on the speed, an action taken with these gestures may or may not be reversible.

Here's how that breaks down:

- A swipe becomes a fling of the finger based on ending velocity and whether the element has crossed a point past which an action can be undone.
- A drag maintains contact with an element, so reversing the direction of the gesture will drag the element back.
- A fling moves at a faster speed and removes contact with the element, preventing an action from being undone.

Complex Gestures

More and more gestures are breaking the “one action equals one reaction” pattern and stringing multiple gestures together.

The result is a complex gesture. The technique is most-commonly used in gaming, but has begun to extend to other apps.



Photo credit: Angry Birds

Complex gestures include three levels of interaction to complete a task.

- 1. Initiation:** Initial contact to start an action, such as a press or tap.
- 2. Continuation:** Apply multiple gestures in a specific sequence, such as tapping in a specific pattern or multiple drags across the screen to make something happen.
- 3. Termination:** End of gesture as contact with the screen is ended.

Think about this usage in the popular game, [Angry Birds](#). To toss a bird to destroy an object, you must complete the following complex gesture:

- Tap, hold and move the bird on the screen to position it for the right trajectory
- Release your finger to toss the bird towards the object.

Clever Gestures Are More Than Actions

Clever gestures do more than perform a task or complete an action. As explained in *Interaction Design Best Practices*, they should also delight users. They may also serve as a teaching tool.

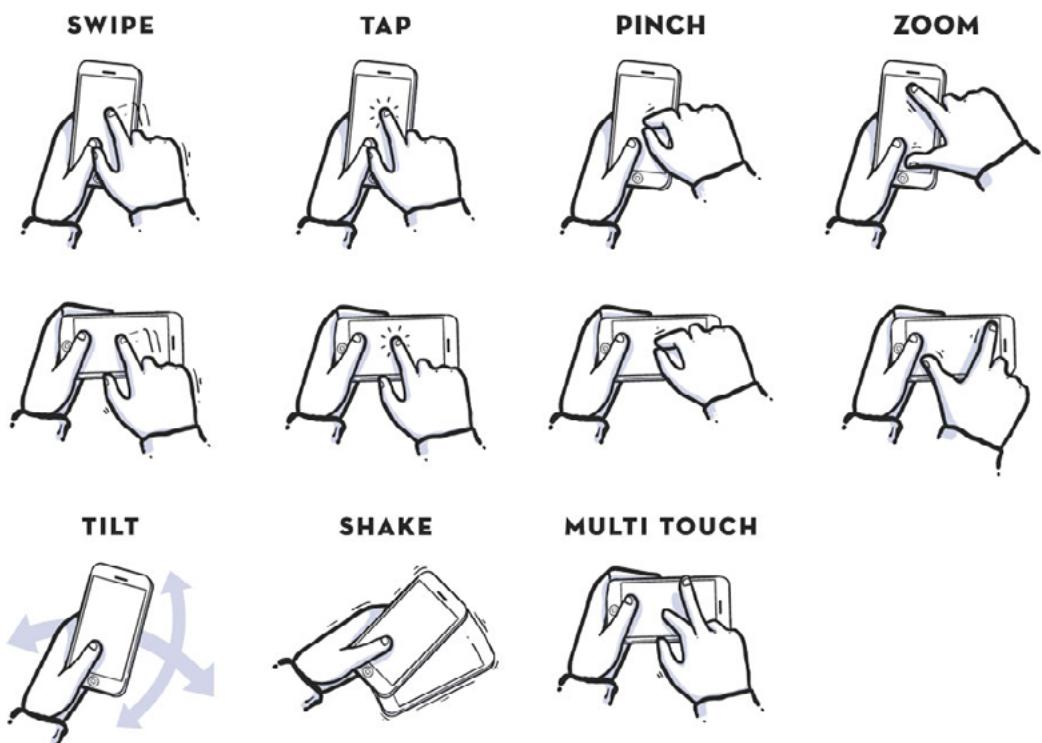


Photo credit: Julian Burford

The best gesture-based interfaces take into account all the ways a user might hold a device:

- Vertically with one hand (could be left or right)
- Vertically with both hands
- Horizontally with one hand (left or right)
- Horizontally with both hands
- With no hands at all (use it laying on a table)

Once you know what users are going to do with your app, how will they know what to do with it? How do you teach them to use gestures effectively? This is more important than it might sound. While common gestures all start with the same actions, the reaction is often different based on the app.

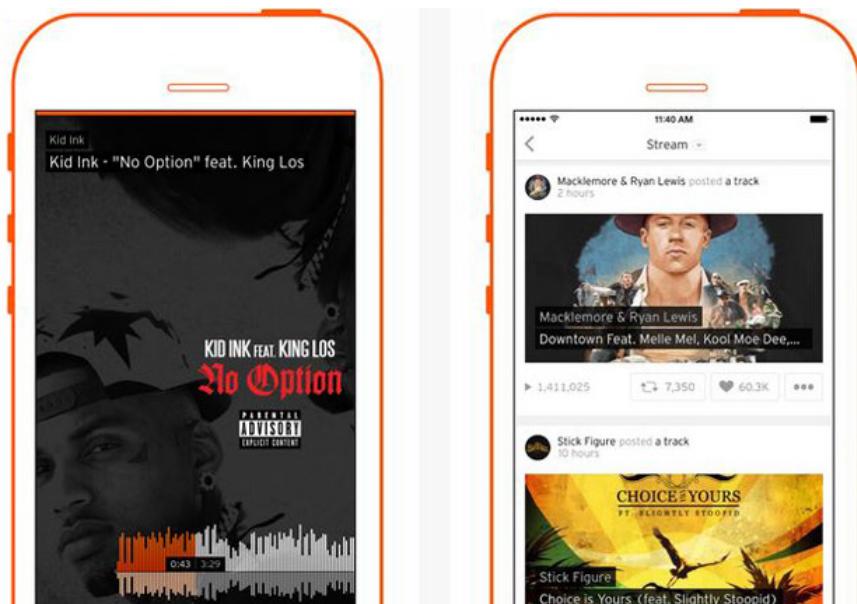


Photo credit: Soundcloud



Photo credit: Clear

Let's compare the two examples above.

[Soundcloud](#) and [Clear](#) both use the same gestures to perform different actions within the apps. Both use taps and swipes. In Soundcloud, gestures help users play and explore music (a tap can open a song or playlist and a swipe helps you explore new things). Clear uses taps and swipes to add and check off items from your to-do list.

The gestures are common and similar in the scope of using taps and swipes to complete actions, but the specific actions themselves are quite different.

The key is that these actions happen pretty seamlessly for users because of simple visual signifiers. The card-based interface of Soundcloud is instantly understood thanks to its popularity, which means no explanation is required for the tap and swipe gestures. Clear's unorthodox rectangular layers creates a sense of modularity, hinting that each rectangle can be swiped away as if it were a sheet of paper.

Finally, when thinking about gestures, it is important to be aware of gesture overload. Too many gestures can be overwhelming. Remember to keep gestures simple, consistent and easy to use.

What's Next?

So where do gestures go from here?

Gestures are beginning to reach their potential. While most gestures are fingertip-based, physical usage is likely to expand in ways we

can't imagine yet. Gestures will also start to integrate more fully with other design trends.

Most of the gestures we see right now are actions performed on top of static design. With the emergence of video and even more animation in the background, gestures will layer more and more animation.



Photo credit: [Haze](#)

The [Haze weather app](#) is already doing this. The app has beautiful layers of colored animation that relate to temperature. There are also plenty of gesture-related controls, from a top-down slide-out to simple swipe actions. The overall effect is visual stunning and highly functional.

Gestures will better link apps to one another. For example, drag-and-drop actions between different apps.

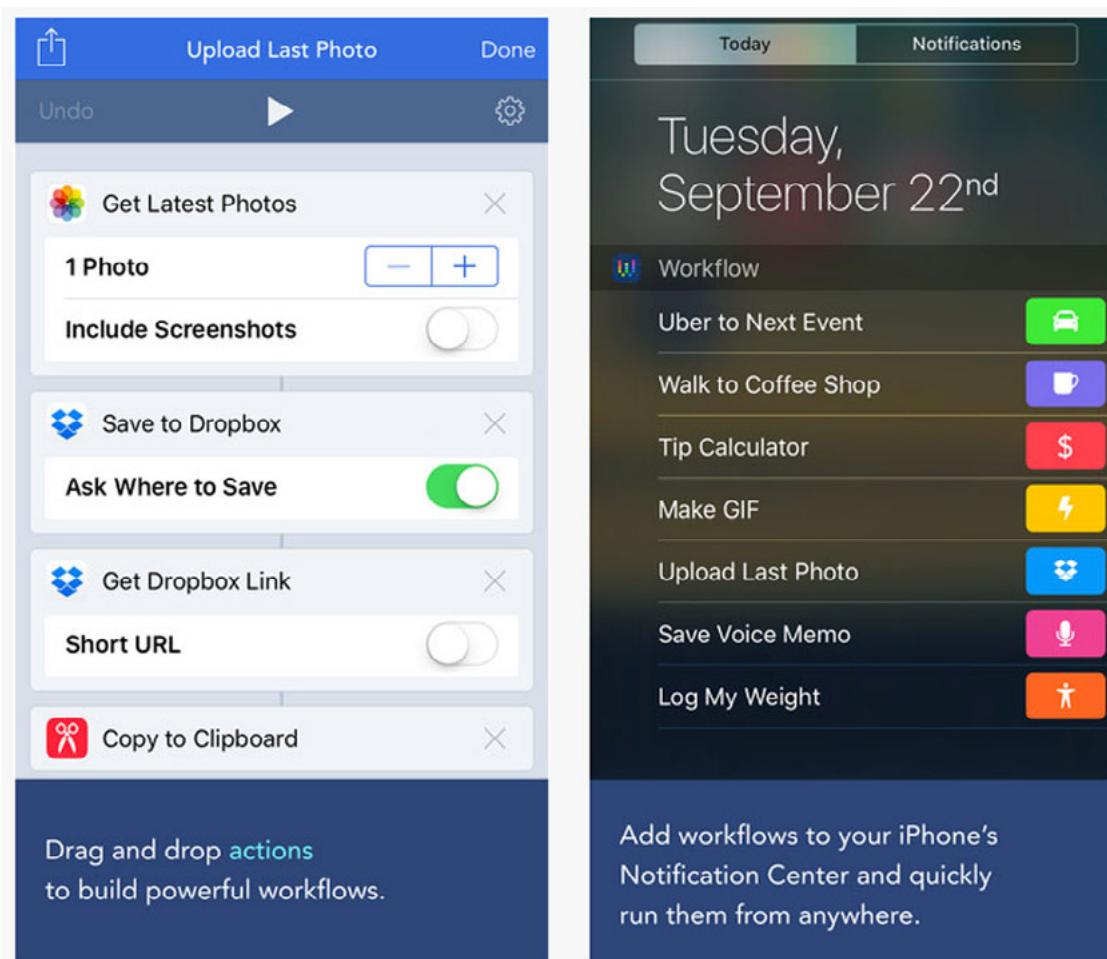


Photo credit: [Workflow](#)

[Workflow](#) is an iOS app that is already demonstrating this functionality. You can take action between apps with drag-and-drop tools (tap and drag). The automated tools are clever and are completely rooted in animation, gesturing and functionality.

When it comes to wearables, gestures will expand to include physical events.. Swing your arm to send a text or flick your wrist to send a call to voicemail. It's very likely that physical actions beyond the touch of a finger will start to dictate how gestures work for other types of devices.

Takeaways

There's no way to design a mobile app without thinking about gestures. These tiny touches are an integral part of the design process and will continue to influence how we design for users.

The secret, though, is the same as with approaching any other design project: new gestures must be intuitive in order for users to embrace them. Do this well and you'll have an app that works well and has a higher likelihood of adoption.

[Prototype mobile apps faster with UXPin \(free trial\)](#)

10 Resources and Tools

1. [Touch Gesture Reference Guide](#) by Craig Villamor, Dan Willis, and Luke Wroblewski (Download)
2. [25 Free Gesture Icon Packs for Your Mobile App](#)
3. [Apple iOS Human Interface Guidelines: Interactivity and Feedback](#)
4. [Google Material Design: Gestures](#)
5. [Designing for Intel RealSense Technology](#)
6. [300+ Icons for Mobile Applications](#)

7. “Google: Your Hand May Be the Only Interface You’ll Ever Need”
8. Accessibility and How Guidelines Apply to Mobile
9. Free Vector iPhone Gestures (as seen in this chapter)
10. Tutorial for Using Gestures in App Design by MIT App Inventor

Layered Mobile Interfaces

Google's [Material Design](#) is one of the most influential visual philosophies in design. It's shaping the way people see and interact with interfaces because of clear design and usability guidelines.

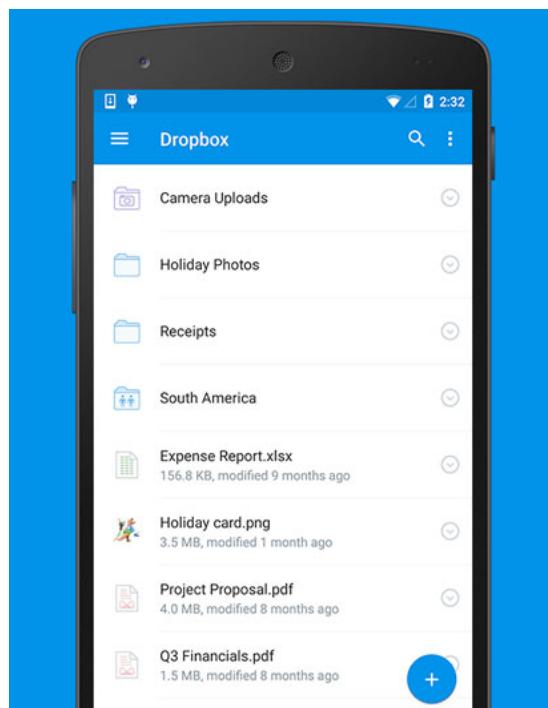


Photo credit: [Dropbox](#)

However, Material Design goes beyond Google and Android apps. Designers are using the design philosophy in a number of ways – especially the concept of layered interfaces. Just as the name suggests,

layering involves stacking multiple elements, like a deck of cards, to create a single, unified experience that is both functional and aesthetically pleasing.

The idea of layering elements in an interface isn't a new concept. However, Material Design takes it a step further – combining a tactile experience with a bold and vibrant aesthetic.

Let's Talk About Material Design

Before we go forward, let's make sure we're all on the same page as to what exactly is Material Design. Let's review:

[Material Design](#) is a Google's conceptual design philosophy that outlines how apps should look and work on mobile devices. It breaks down everything – such as animation, style, layout – and gives guidance on patterns, components and usability. According to Google:

We challenged ourselves to create a visual language for our users that synthesizes the classic principles of good design with the innovation and possibility of technology and science. This is material design.

Material starts with mobile but extends to any other device. It is rooted in a few principles:

- 1. Realistic visual cues:** The design is grounded in reality and actually inspired by design with paper and ink.

2. **Bold, graphic and intentional:** Fundamental design techniques drive the visuals. Typography, grids, space, scale, color and imagery guide the entire design. Elements live in defined spaces with a clear hierarchy. Color and type choices are bold and deliberate.
3. **Motion provides meaning:** Animation is a key component of Material Design, but it can't just be there for the sake of movement. Animations need to happen in a single environment, serve to focus the design and include simple and easy transitions. Movements and actions should mirror the physical world.

Understanding the “Tactile Surface”

One of the things that comes up quite often when talking about layered interfaces is the “tactile surface.” Think of this as having multiple sheets of paper that are stacked together to create a framework for how everything within the design works. These sheets are a little different from physical sheets of paper in that they can change shape and form – such as stretch or bend – but work in a way that is seemingly realistic.

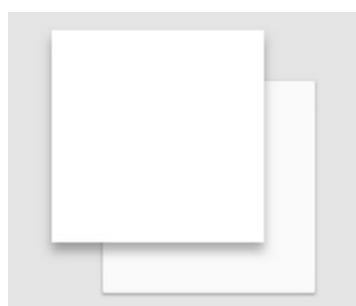


Photo credit: Google Material Design

The tactile surface is a container for content and information. The container is flat in design but has a faint shadow to separate it from other containers and layers. Other techniques to create separation between layers – such as textures, gradients or strokes – are unnecessary.

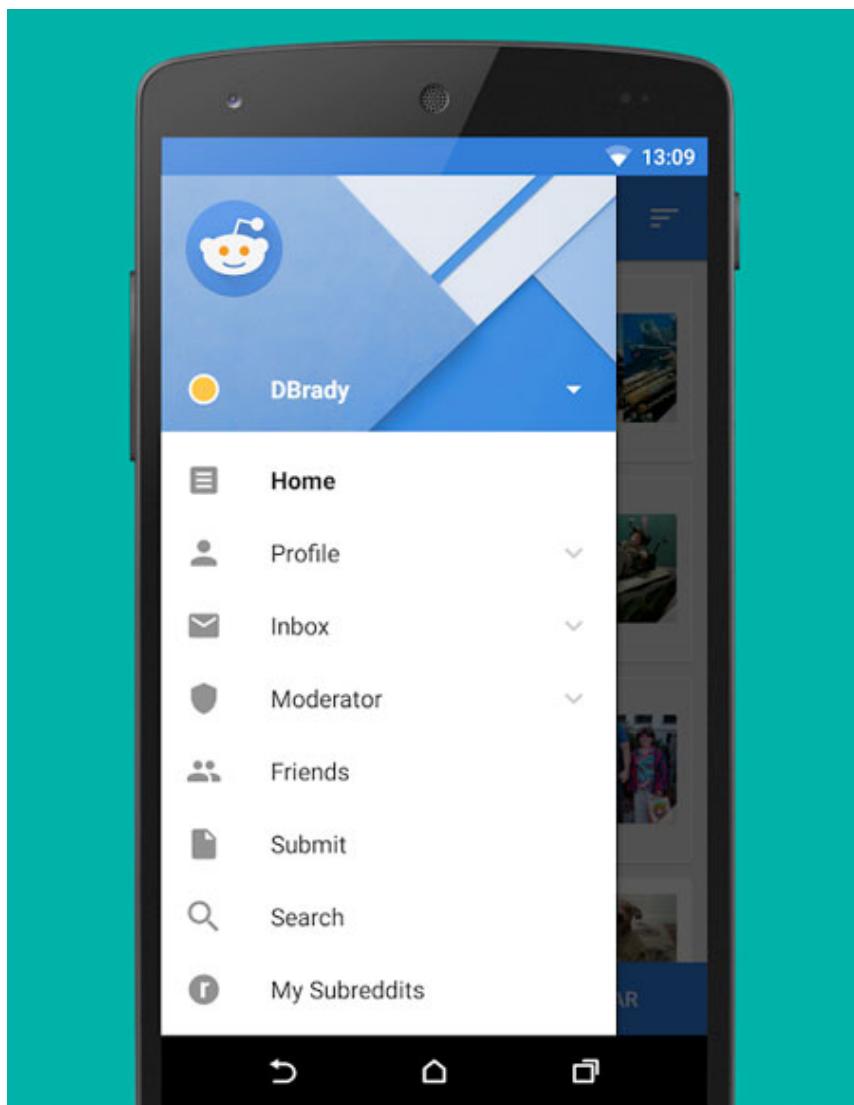


Photo credit: [Reddit](#)

You can see the separation in the layers for the [Reddit app](#), above. There is an obvious top menu layer covering a grayed out main content layer. Even the main header image contains elements of layering and shading that emphasize a three-dimensional tactile surface.

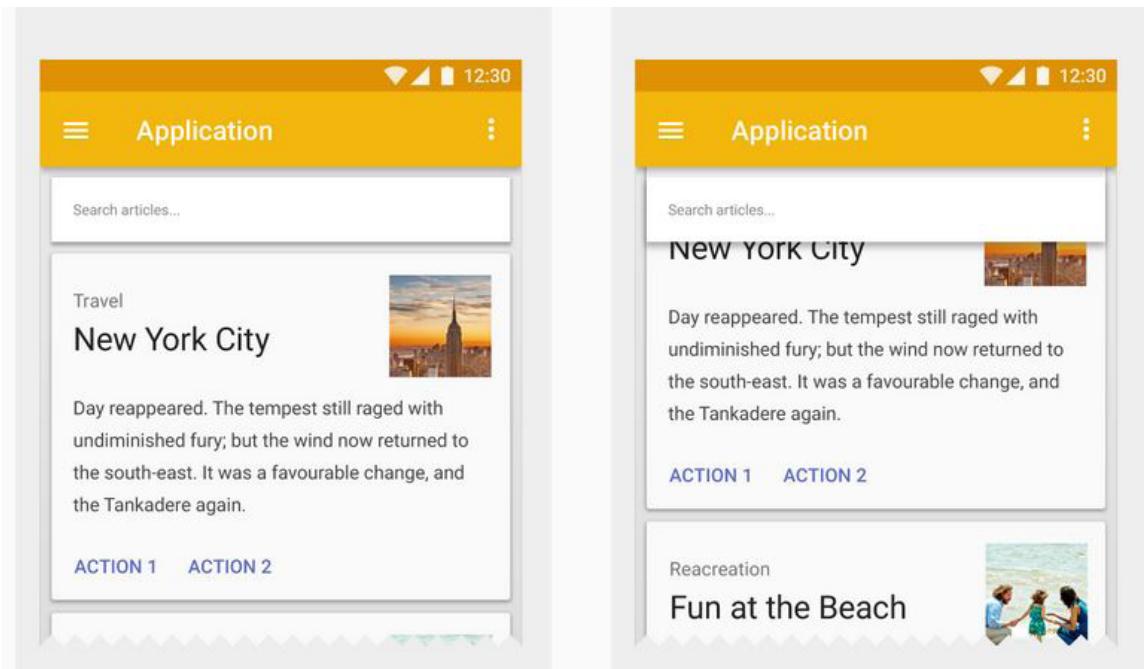


Photo credit: Google Material Design

A tactile surface clearly established the relationship and function of content within a design. (Each container often has one job, such as a link or video player.) This approach also establishes depth, as elements in other containers are layered, creating a nearly three-dimensional world.

Material is Made for Adaptive Design

Layered interfaces are inherently made for adaptive design. All of the design guidelines actually encourage a designer to work with an adaptive layout (whether you prefer **adaptive** or **responsive** is up for debate, however.)

When thinking about layered interfaces, it is important to consider how all the elements relate to one another.

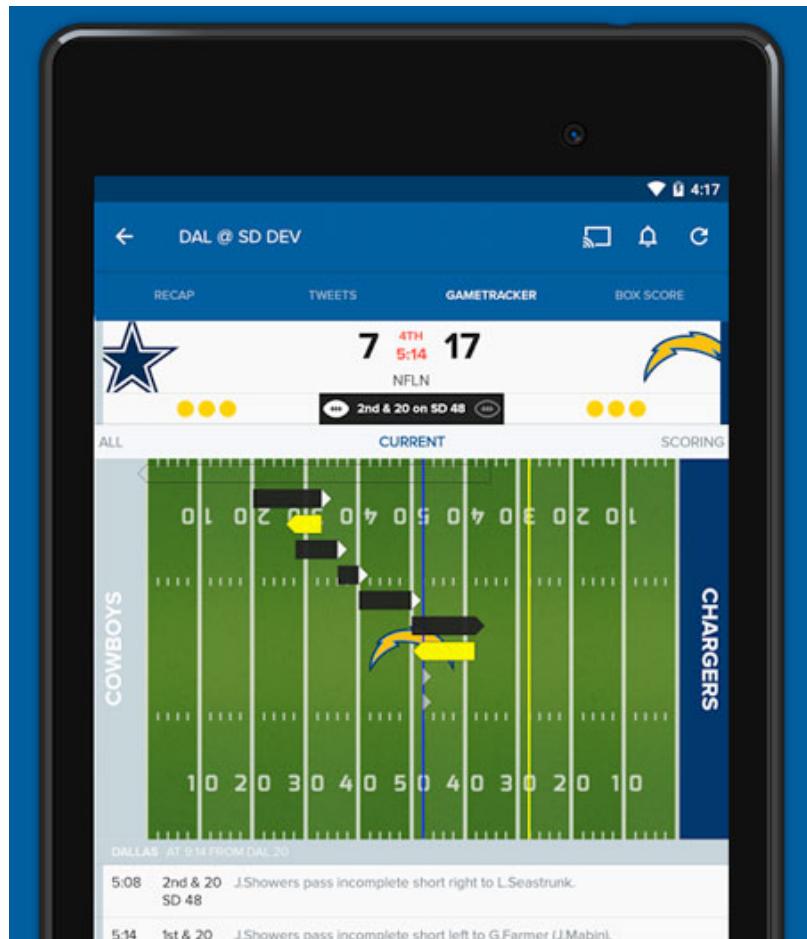


Photo credit: CBS Sports

Google recommends its standards because of a “flexible grid that ensures consistency across layouts, breakpoint details about how content reflows on different screens, and a description of how an app can scale from small to extra-large screens.”

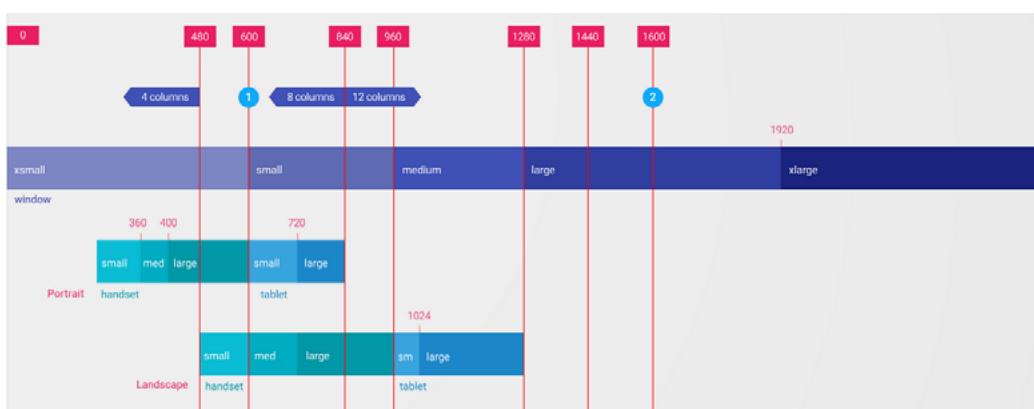


Photo credit: Google's Material Design Guidelines

Considerations include:

- **Breakpoints:** Widths include 480, 600, 840, 960, 1280, 1440 and 1600 pixels
- **Grid:** 12-column layout with margins and gutters (8, 16, 24, or 40 pixels) and a baseline grid
- **Surface behaviors:** UI adapts to the type of screen so that surfaces are visible or toggled to hide
- **Patterns:** Function is based on screen size, including reveal, transform, expand, reflow and divide

These considerations make it easy for designers to ensure their interfaces adapt for any device in any situation. They provide a baseline to help designers as they construct layouts for desktop, tablet and smartphone.

Material and Other Mobile Design Trends

When it comes to creating layered interfaces, other trends also come into play. Material Design has borrowed plenty of design concepts from the flat aesthetic and other trending techniques. In fact, some would argue that Material Design is a close cousin to [Flat Design 2.0](#) because many of the visual treatments are quite similar.

What separate layered interfaces from totally flat design is that effects are needed to create more three-dimensional spaces and to mimic

lighting. In essence, designers are bringing back some of the design tricks eliminated with flat design. The difference is that they're using these tricks to improve usability rather than simply as decorative accents.

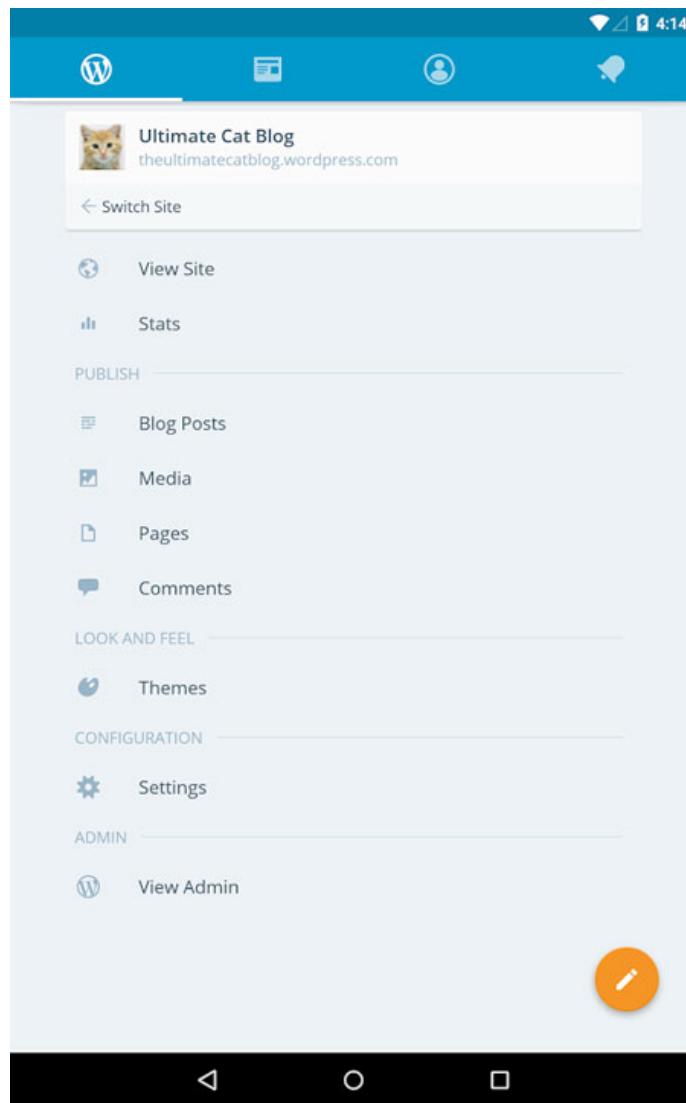


Photo credit: [WordPress for Android](#)

The colors most closely associated with layered interface design nearly fall within the flat aesthetic. The big difference is the vast number of color options that Google provides. Palette options are in the same vein though – bright, bold and fully-saturated hues. While many designers opted for blues and reds when it comes to flat design, more

layered interfaces seem to feature deep purples and yellows. That's likely because each of these hues is easy to pair with contrasting white or black text.

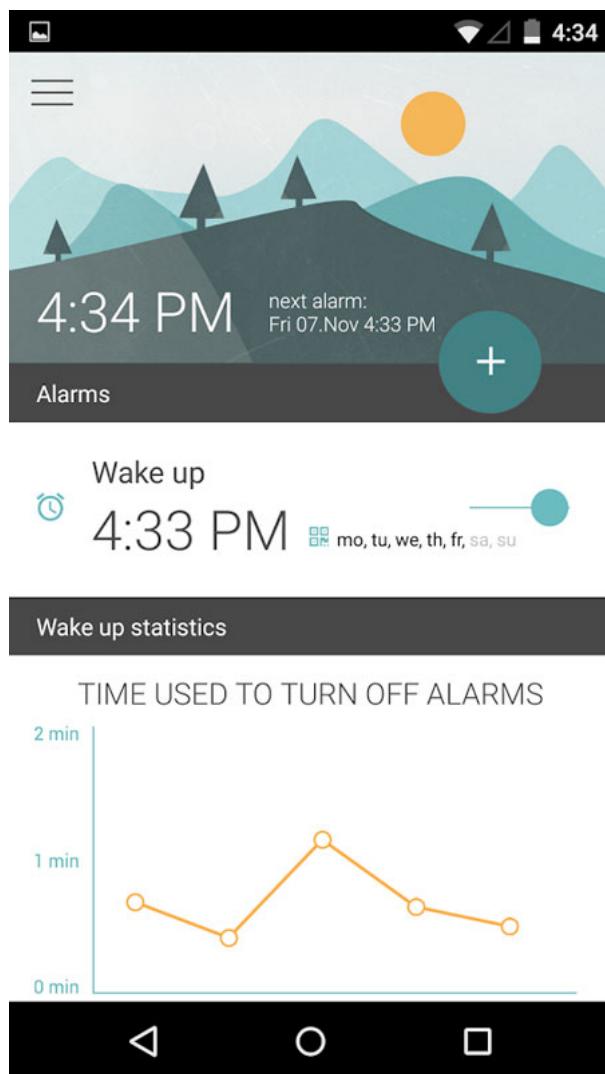


Photo credit: Morning Routine

Layered interfaces also work well in the space of minimalism, particularly when it comes to typography. Type is stacked with clear hierarchy and sans serif options are the preferred choice. Google suggests using **Roboto** as the dominant typeface (which comes with plenty of choices, from thin to bold to italic to condensed). The variety helps create levels of type that guide users between elements. In the

true spirit of minimalism, one font can pretty much do everything you need with the proper sizing and scaling.

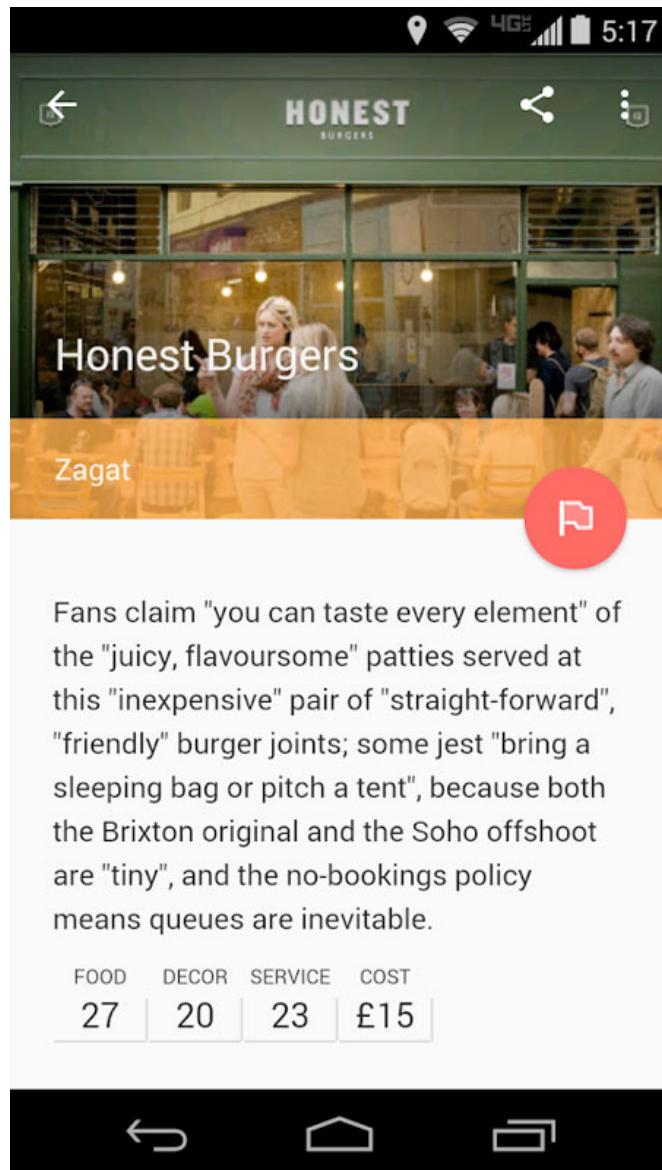


Photo credit: [Field Trip](#)

You'd be hard-pressed to find an app without a full-screen image these days and layered interfaces further emphasize the use of vivid, intentional imagery.

The [Field Trip](#) app, above, makes the most of a simple photo for this very purpose, showing that Material Design does not just use color,

photos and effects solely for visual impact – they are an essential part of the design.

Finally, layered interfaces are perfectly made for cards, which we discussed in the previous chapter. Looking through the examples showcased, almost every design includes something with a card-like element. From smaller cards to full-screen options, these trends go hand-in-hand.

What's Next?

In July 2015, Google released a few new guidelines ([Material Design Lite](#)) so that any website or app developer could make their products look more like Android apps. This is something we'll see more of, maybe even in Apple's iOS and other non-app areas.

Material Design looks nice and works well in a variety of places. Designers will want to take advantage of that and the lite version provides the perfect level of guidance. Material Design Lite is also a good tool for designers and developers that want to create a unified app experience across the Android and iOS platforms, so that apps look, feel and function consistently regardless of device.

Layers are definitely going to stick around, but the overall look may be a little more “layered” and a little less material, so that the design falls somewhere between Material Lite and iOS standards.



Photo credit: [The Weather Channel](#)

The [Weather Channel](#) iOS app is already using this approach. The app layers cards, colors and images. Where the design concepts overlap most is in the use of cards and the placement of geometric shapes. Where the design is “less Material” is in the lack of depth and shadowing so that the overall look is flatter and streamlined.

Gradients and monochromatic color layers are another way layered interfaces can continue to grow visually. [Monochromatic color palettes](#) are a classic design technique that make it easy to create sharp elements to fit almost any type of content.

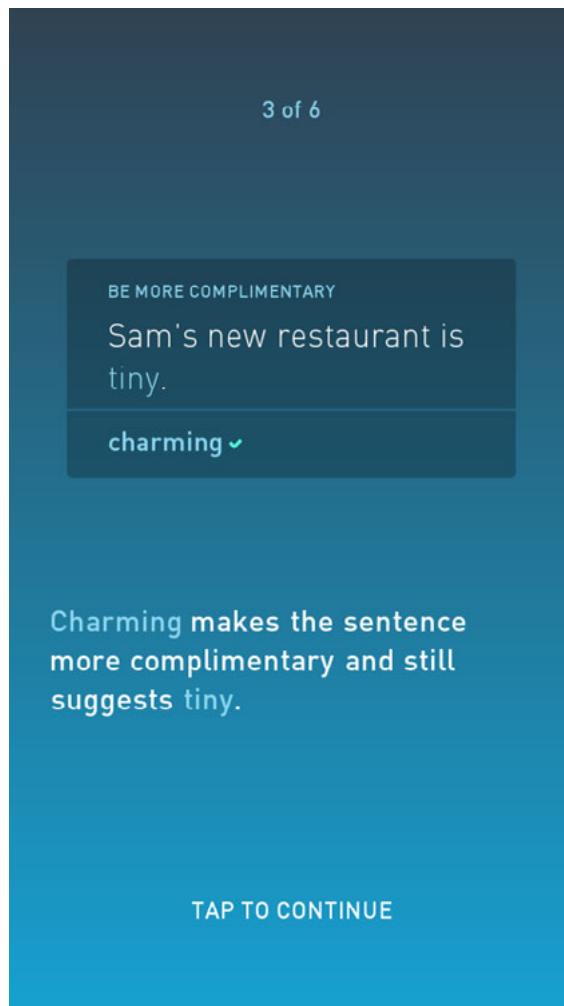


Photo credit: [Elevate](#)

The [Elevate](#) iOS app uses a gradient background with multiple levels of cards in its design. The animations and movements are very Material Design in nature but the use of a gradient is not. This simple evolution highlights how designers will start to break the visual rules of layered interfaces while continuing to leverage its more functional aspects.

Designers will continue to evolve layered interfaces and Material Design concepts with darker colors and hues. Most of the apps available right now feature light and white color schemes, but darker colors will start to emerge.

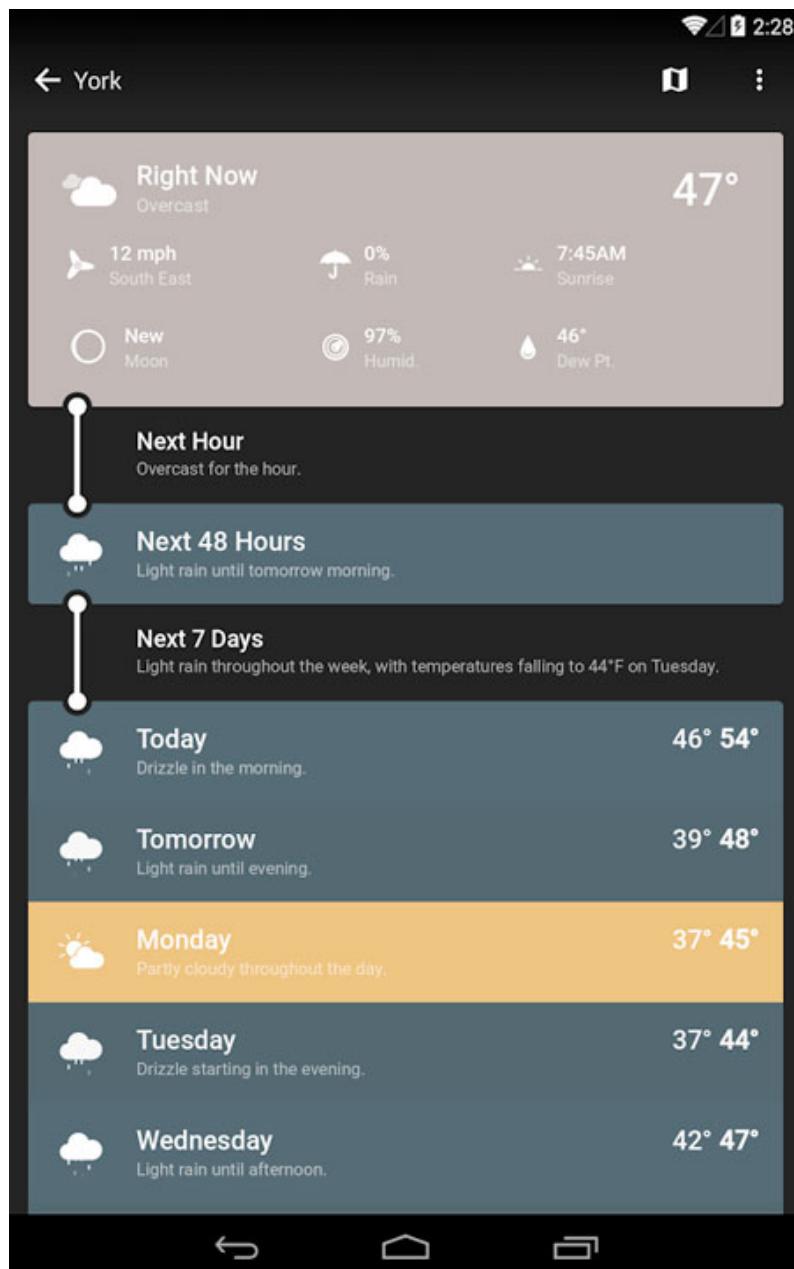


Photo credit: [Weather Timeline](#)

[Weather Timeline](#) is a perfect example of this. The simple change to the color palette is enough to really make this app stand out from all the others available. It still uses a style that's distinctly layered, but the darker interface is simple and elegant. The colors for the entire design are less saturated and toned to match the darker aesthetic.

Takeaway

Today's layered interfaces are just the start.

The simple visual style and high usability of this design style will continue to emerge and grow as designers -- not just for Android -- will latch on the concepts. What may be even more interesting is that the look of layered interfaces is really just an extension of a lot of the design techniques that have been growing in popularity for several years, including flat and minimalism.

At some point the pendulum may swing back to more “realistic-looking” interfaces, but until then this concept appears to have quite the foothold.

Design faster wireframes & prototypes with UXPin (free trial)

11 Excellent Resources and Tools

1. [Google Material Design](#)
2. [Free Lollipop UI Kit](#)
3. “[How Material Design Sparked Evolution of Web Design](#)” by Material Design Blog
4. [Angular Material Framework](#)

5. Downloadable Material Design Color Palette Swatches
6. “The Interface Layer: Where Design Commoditizes Tech” by Scott Belsky
7. Material Up: Daily Design Inspiration
8. Material Design Icon Template Download
9. Materialize Framework
10. Material Design Typography Classes
11. Angular Material vs. Material Design Lite

Meaningful Typography

Let's be clear about something: typography for small screens does not mean small type.

The smaller the screen, the more important it becomes to use typography with specific intent and meaning. It starts with a good understanding of the [foundation of good lettering](#). And extends to applying those concepts to smaller screen sizes and thinking about how responsive typography comes into play. (When it comes to type, one size does not fit all.)

In this piece, we'll show you how to use typography effectively in your mobile UI designs.

Typeface Selection is Important

When it comes to lettering, typeface selection is everything. This includes everything from the type family you choose to work with to size and to how typefaces are incorporated into the overall design.

While there really aren't any perfect rules – with a good eye and idea of your audience behavior, you can make almost anything work – there are a few common threads when it comes to typefaces for smaller screens.

1. Body copy is often sans serif

While the trend in mobile skewed toward thin, condensed typefaces with an uber-modern feel, that concept was rather short-lived. Nowadays, typefaces are somewhat thin- to medium-weight with uniform stroke widths. This style of type is easy to read at a variety of sizes and against a number of backgrounds.

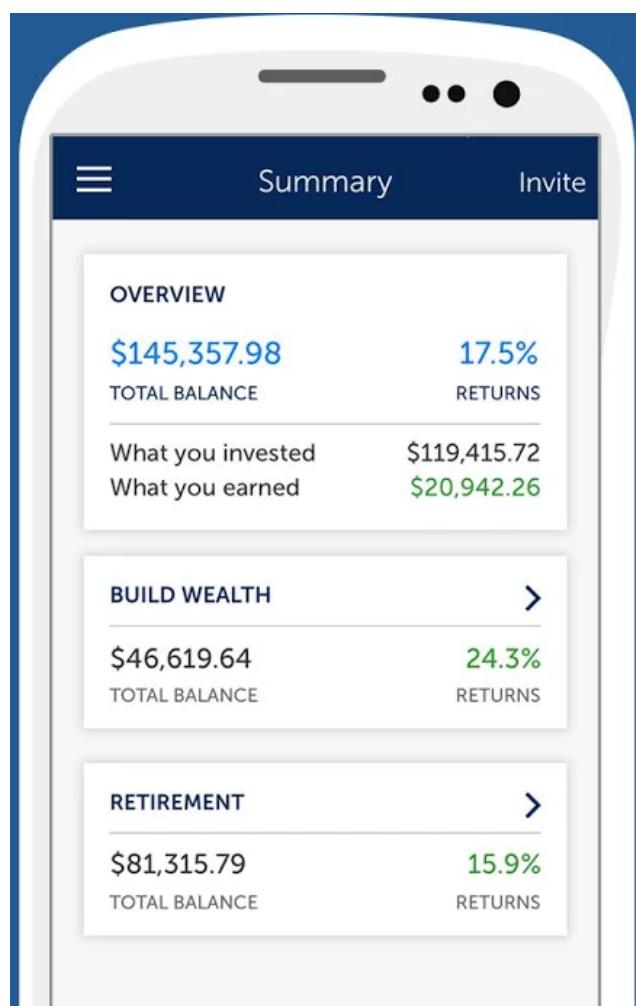


Photo credit: Betterment

One of the arguments for using sans serif is that it is easier to read on screens. Another way to ensure that your copy is readable is to keep it to **30-40 characters per line**. Most text is one of three colors – black, white or gray

Again, this comes back to readability. Text needs to be separate from the background whether it is a color or image or video. Avoiding a lot of colored type is an easy way to do that. The common color values used are **#0D0D0D**, **#0F0F0F** and **#141414**.



Photo credit: [Fitbit](#)

Look at the example above with **Fitbit**. The gray lettering is easy to read, even against some of the more subtle gray backgrounds.

This contrast without “extreme” contrast can be easy on the eyes, especially when it comes to digesting a lot of information. Font size and readability matters.

If you’re building a website, responsive typography is a must because you’ll want whatever typeface you choose to scale properly across devices. After all, the same 16-point type looks quite different on an HD monitors and smartphone screens.

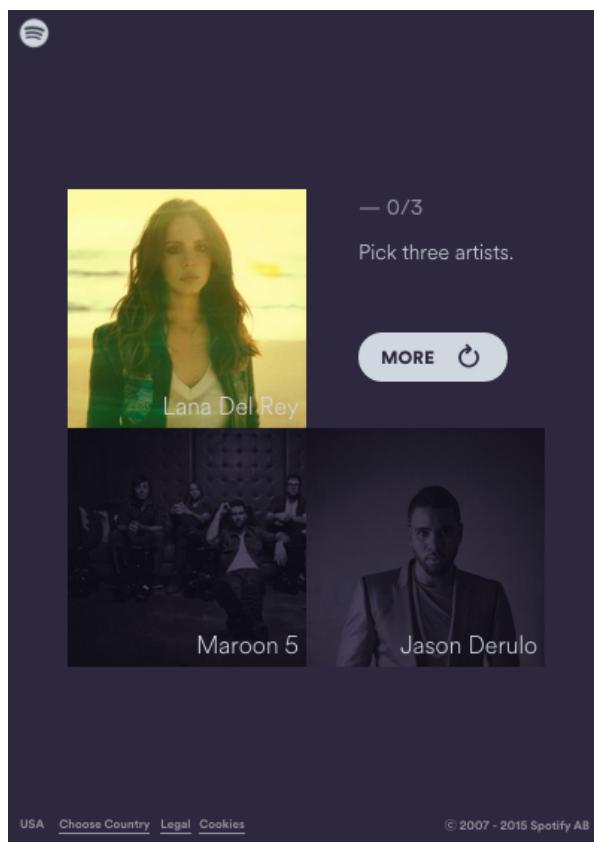


Photo credit: Spotify

Select a typeface with smooth edges for optimal readability. Avoid styles that are overly elaborate, such as scripts or novelty styles. Another thing to consider is web font integration. Many tools, including [Google Fonts](#) and [Adobe’s Typekit](#), have built in integration for apps and responsive websites. Take advantage of these tools

when building your sites and opt for high quality typefaces, which will most likely scale better.

With these common threads in mind, let's dive deeper into using typefaces in mobile.

Give Words Some Space

When it comes to type, space is a make-or-break element. Too much space can make type feel minuscule or out of place on the screen. Too little space and text becomes difficult to read and also feels small and cramped.

You have to find that sweet spot in the middle.

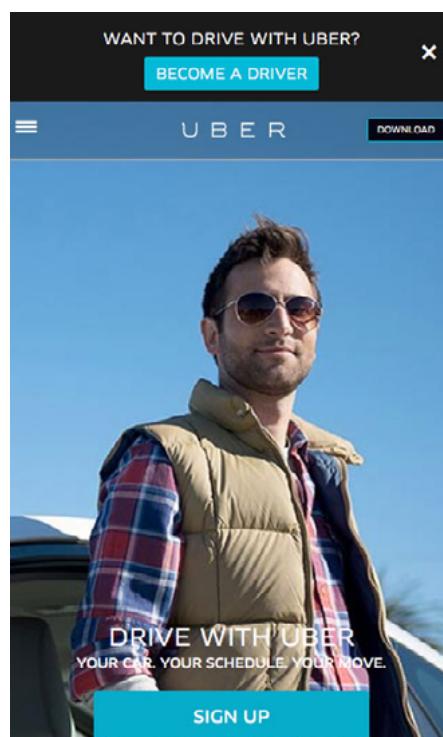


Photo credit: Uber

Unfortunately, that's not something that's easy to define. You almost need to see it to know it. While [Uber](#) and [Twitter](#) responsive sites use space in two very different ways, they are equally efficient and increase the readability of the typography. Uber (above) goes the route of giving text plenty of space, almost to the point of excessive space, which brings very distinct attention to its calls-to-action.



Photo credit: Twitter

Twitter, on the other hand, allows text to practically fill the screen, so users can access as many 140-character messages at a time as possible. Both are very different, but both make quite efficient use of nice typography in conjunction with other design elements to make it work. Just imagine the two designs flipped, how would you feel about the typography then?

Here's some practical takeaways:

- The smaller the font size, the more space your text needs to be perfectly legible. Many designers like to start with 1.5 times the point size of a specific font for the space between lines, or leading. (Fans of the golden ratio extend that to 1.618.)
- But don't stop there – consider additional space between paragraphs or different kinds of text elements.
- When it comes to space and typography, space should provide the right amount of buffer for each readable thought. Think of space as the place where readers pause before moving on to the next idea. If you want elements to read together, keep the space a little tighter, but when the idea or concept changes, add a little extra space and give users time to digest what they have read before moving on to the next thing.
- Finally, when it comes to building a typography framework, think of the words as bricks. Each word is a brick and the space between is mortar; even if the blocks are of varying sizes the space between each is rather uniform. This visualization can help you establish visual guidelines for how words should look, rather than what they say.

Pay Particular Attention to Alignment

Left, right, justified? When it comes to alignment, don't overthink it. In most cases text should be aligned to the left margin with a ragged right side or centered.



Photo credit: <https://spotify-tasterewind.com/>

Avoid justified text. It can get hard to control, with odd (and distracting) spaces that form between words or letters. Ragged text has a more natural feel to it and helps maintain consistency in lettering. It's easy and comfortable to read, and won't leave odd design gaps due to varying column widths.

How do you know when to center or when to left-align text?

- **Use left alignment for larger blocks of text.** Anything that requires actual reading should be left-aligned. Left alignment also works nicely for text elements that fall into a grid, card-style containers or in chunky blocks of text that are supposed to be read together. (Think Twitter: Each text element is 140 characters or less, but most users read multiple text elements back to back.)

THE ONE THING I ALWAYS FORGET TO DO WHEN I TRAVEL



You guys, I consider myself to be a pretty organized traveller. I compartmentalize my clothes into packing cubes so everything will be

Photo credit: <http://thatbackpacker.com/>

- **Use center alignment for special, short bits of text.** Centering can be the perfect solution for a headline or simple call to action. Center placement is also a popular choice for buttons because users can easily reach them with either thumb.

Another option: try mixing centered and left-justified options, especially when there's very few words. This technique works well with a centered headline and ragged body text. The contrasting alignments make for a natural and easy reading experience. This is a popular

technique with blogs, such as [That Backpacker](#), a popular travel blog. When text is aligned well, users will easily be able to consume your content. If it's out of whack, users won't waste their time trying to read what's on the screen.

Create Contrast

This might go against everything you have ever heard – tone down the contrast a bit. You don't have to use solely black type on a white background to make content readable. One of the reasons for this is backlighting in relationship to the amount of text on the screen.



Photo credit: http://www.amazon.com/dp/B00I15SB16/ref=ods_fs_kb

Think about Amazon's Kindle e-reader for a moment. It is the standard in readability among digital devices. And it uses shades of gray to display text and backgrounds.

This same concept applies to many of your favorite websites as well. [Eventbrite](#) and [Airbnb](#) use a color and contrast frameworks that emphasizes subtle grays, rather than stark black and white for backgrounds and text. Both sites are super easy to read – so much so that you probably don't even think about it. Both websites use white text only when it is on top of a photo or video background that provides ample contrast, but without being too stark.

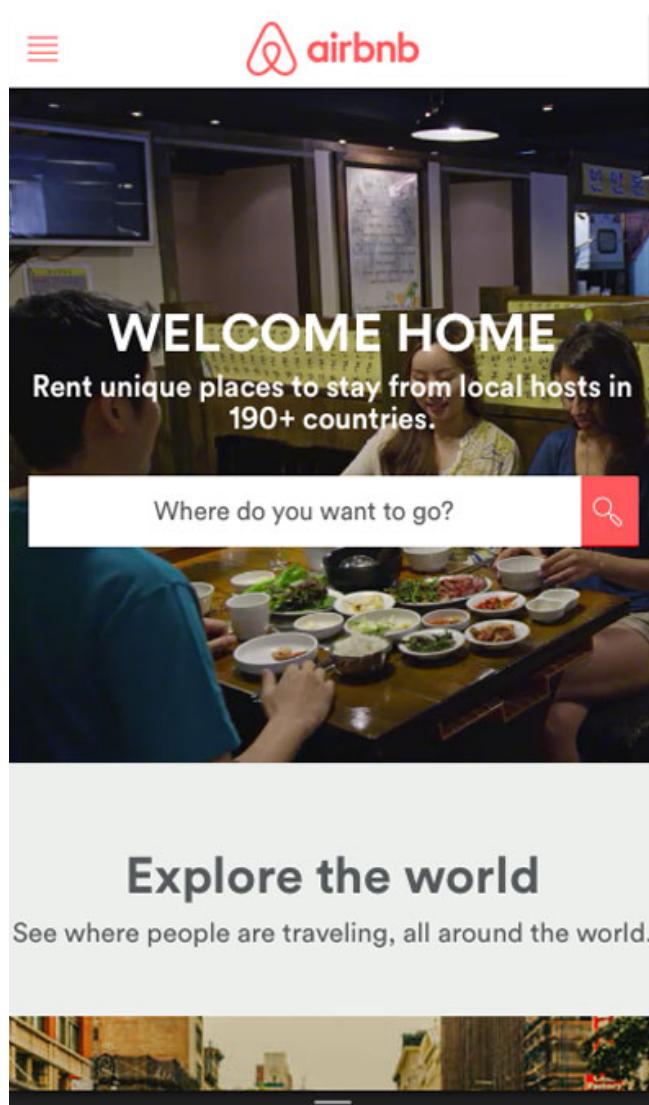


Photo credit: [airbnb](#)

In terms of contrast, the most important consideration may actually be size. When it comes to size, there are no simple answers. That's

because it hinges on the size of the device canvas. It is best to think about typography size in terms of how wide the canvas is in relationship to how many characters fit across the screen. While standard font sizes have been creeping upward, about 40 characters per line is a good place to start, as we said earlier.

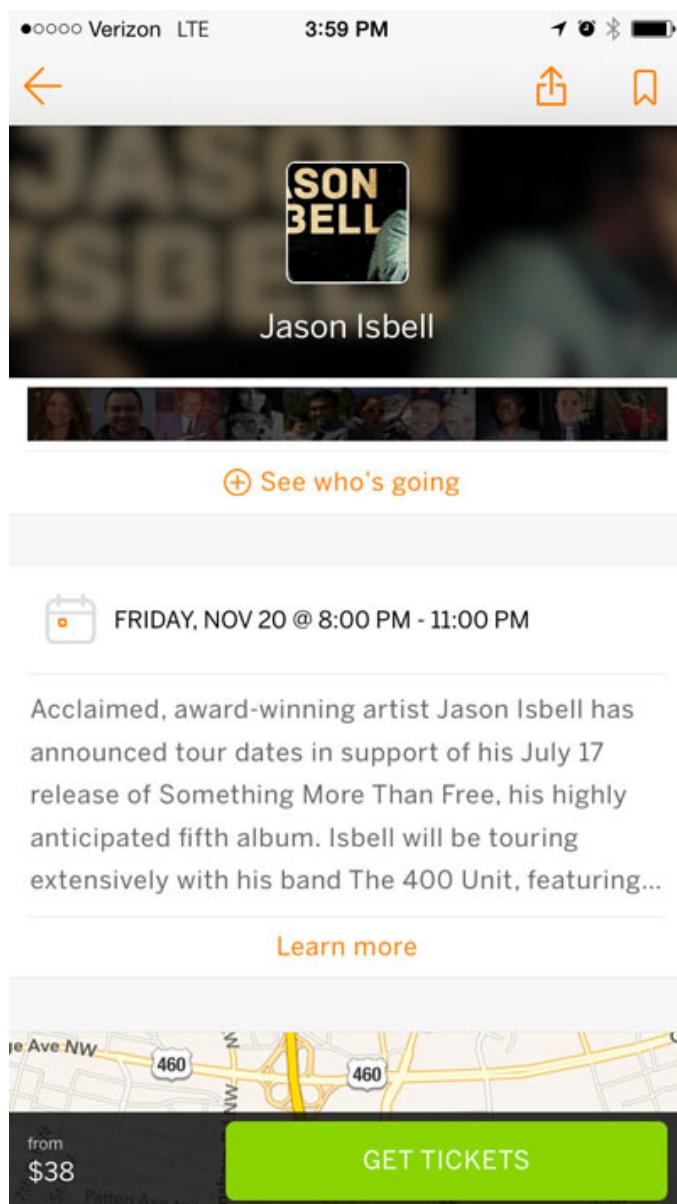


Photo credit: <https://www.eventbrite.com/>

Users should not have to pinch and zoom to read the words on the screen.

This character count mirrors the flow and readability standards of physical items, such as books and magazines. The smaller the canvas, the smaller the character to width ratio.

This concept is why many larger designs, such as newspapers and magazines, use multiple columns to maintain a reader-friendly experience. And the optimal character to column width for body text is around 40 characters, proving that everything old is new again.

Design Hierarchy (But Maybe Fewer Levels)

To further enhance the reading experience, designers are downsizing the levels of text in their designs. On mobile, the hierarchy tends to focus on two type styles and sizes – a headline and body text. (Secondary elements, such as links or subheads, are becoming less common or tend to mirror one of the main two styles.)

When it comes to creating hierarchy for most types of design, you look from left to right and top to bottom. The primary concern on mobile is vertical rhythm. Few designs contain more than one block or column of text, so the user can ease through one bit of typography to the next in the course of scrolling.

You need text that is easy to read and establishes a single thought without the user having to do a lot of swiping or tapping to understand the message.

Should you buy a house or keep renting?

Updated by Timothy B. Lee on August 31, 2015, 8:10 a.m.

ET  tim@voxy.com

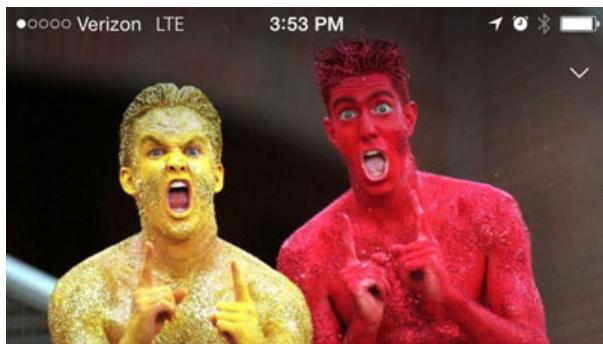


Marta Iwanek/Toronto Star via Getty Images

There's a widespread belief that you should always own a home instead of renting, because if you rent you're "throwing your money away," while buying a home means you're building

Photo credit: Vox.com

Generally, typographical hierarchy is rooted in general [visual hierarchy](#). In fact, standard HTML include typographical tagging for scale fall under this: H1, H2, H3, H4, H5 and so on. When it comes to small screens, you can forget all those levels. In many cases, two levels of type are enough. As stated before, you need headlines and body copy. Deep levels of hierarchy only really work if you can see all of those elements at once. The visual scale for typographical hierarchy – based on size, color, weight and space – is designed so the eye knows what to read first. On smaller screens, there is only one (maybe two) thing to view or read at a time, making all those typographic levels unnecessary.



The Worst People in College Football

Rolling Stone · By Rolling Stone

Crazy coaches, contrived committees and contrarian commissioners – as we kick off a new season, let's celebrate the best of the bad College football is a corrupt and terrifying pastime that masks its flaws by producing an eminently beautiful product year after year. Some of the awfulness is the fault of a deeply flawed system, but it's not hard to find people who deserve to be called out for their egos and errors. Such is the purpose of the list that follows; as mentioned...



Photo credit: <https://flipboard.com/>

The [Flipboard](#) app does a nice job with type hierarchy. The website actually uses three levels of typography – headline, byline and body text – but the middle one is dropped on some screen depending on the type of content. The type design uses a serif body font, which is elegant and somewhat unusual for this style of website, along with a sans-serif headline. The end result is highly readable, and almost book-like.

All this boils down to keeping hierarchy simple with mobile typography. A headline and body copy is all you really need.

Don't Force It

Typography will look different on smaller screens. It's just that simple. So don't try to force it to conform to what you are planning for larger screens. This is an easy thing to say but it's more difficult to do, especially when you want to create a uniform experience across all devices.

When it comes to typography these elements will be different based on device:

- **Type size:** Think about typography in terms of ems, rather than point sizes. This is especially important for responsive design since you want to think about [type that's scales responsively](#).
- **Line breaks:** Different numbers of characters per line for different screen sizes practically ensure that lines will break in different places. Don't obsess over it unless there is a concern about reader flow.
- **Colors:** Sometimes the perfect color combination on a large screen just does not work for smaller sizes. You can make a change. Remember to consider grays.
- **Font selection:** Because of the likelihood of a more limited font palette, some elements will be streamlined to include the same typefaces, and that's ok.
- **Orientation:** Be wary of turning or flipping text to make it work. Users find text that is horizontally oriented easier reads.

- **Kerning:** Just don't touch it when it comes to working in responsive frameworks. It's near impossible to make changes that work well universally. You are better off to invest the time in selecting a high-quality typeface that auto kerns for you.

What's Next?

For responsive sites and mobile apps, thin is out. Medium and medium condensed are in. MyFonts keeps a running list of the [most popular fonts](#). Among the top 10 are Proxima Nova, Museo Sans, Pluto Sans, Helvetica Neue and Futura. All are medium weight sans serifs with somewhat condensed letterforms.

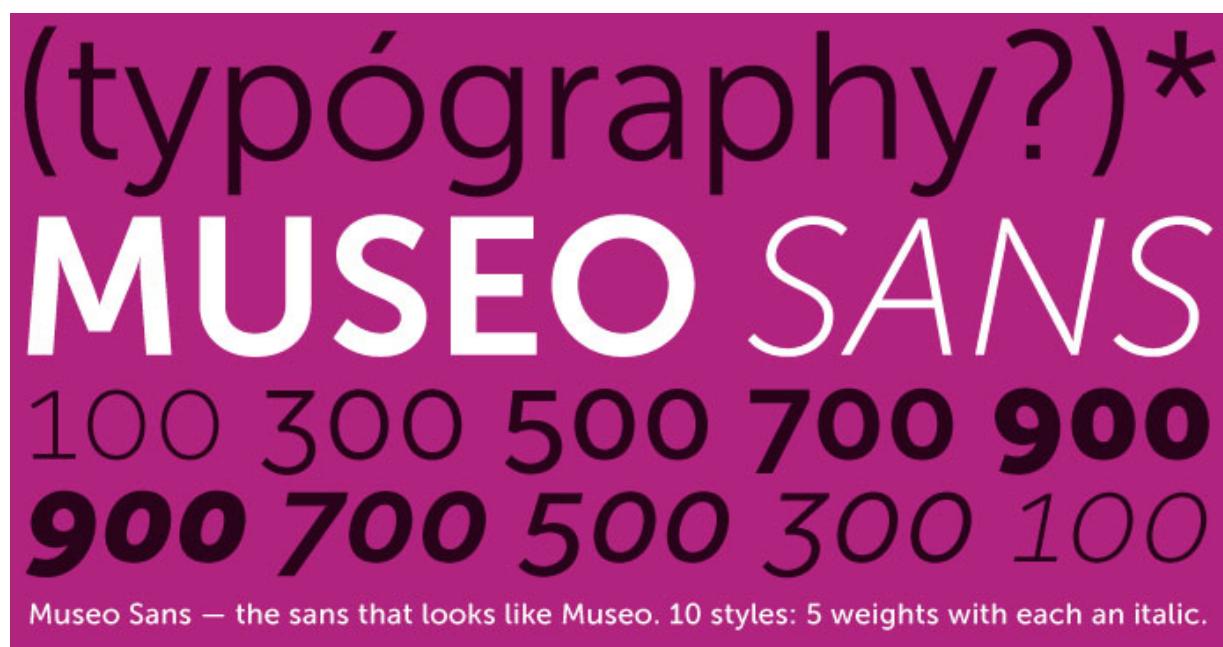


Photo credit: <https://www.myfonts.com/fonts/exljbris/museo-sans/>

The pendulum shifts to smaller typefaces again. When the trend line included thinner, more condensed typefaces, text was often used in more exaggerated ways to make up for a lack of readability. But as

the focus has shifted back to typefaces that are highly readable at almost any size, designers are opting to go a little smaller.

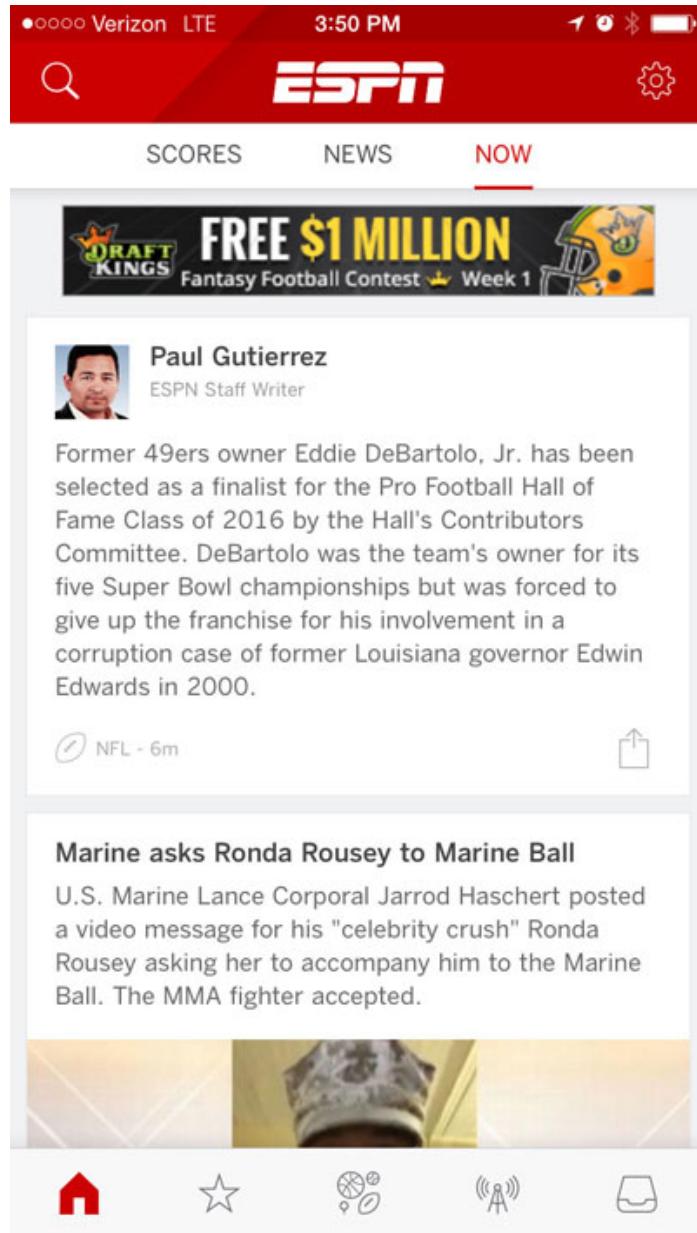


Photo credit: <http://espn.go.com/>

ESPN (above) is one site doing just that. The text inside the “News” section of the app is somewhat small, but remains readable. The text features darker bolder headlines and lighter gray text with plenty of space between lines and a subtle hierarchy. The typography is designed so that it’s easy to read in blocks.

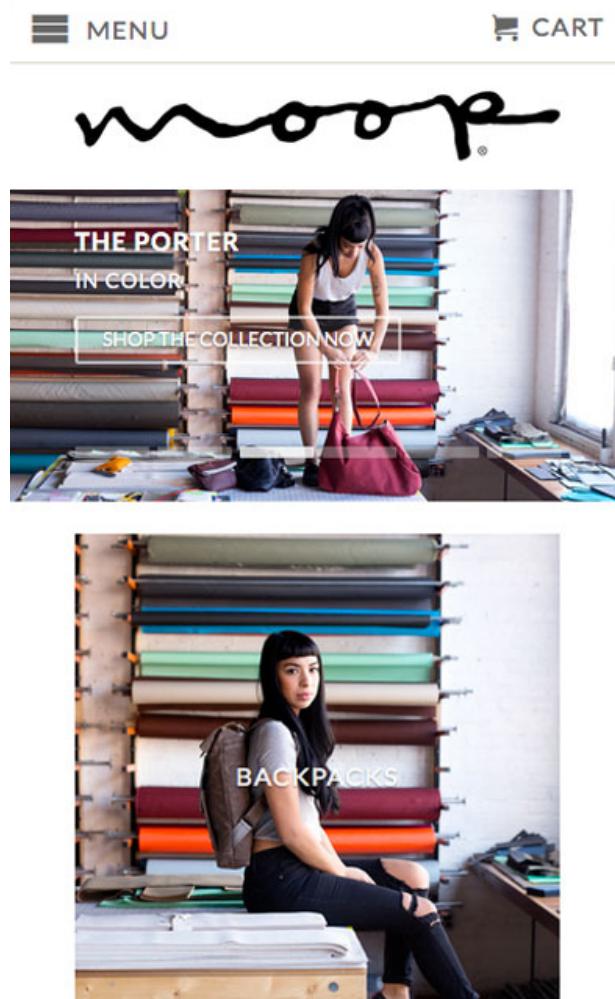


Photo credit: <http://www.moopshop.com/>

Link management has always been an issue on mobile devices. For a long time, inline links have just been too hard to activate. Designers are starting to figure out nice ways to work around this. And while it is less about typography, it is a typography-based characteristic. Card-style interfaces are starting to emerge as a solution to this. Each card includes a well-designed bit of type with the entire block serving as a link.

Moop (above) does this well. Every block is a dedicated link. This includes the ghost button in the header and every block showcasing one of their bags in subsequent blocks. While the design is not per-

fectly card style per se, it uses those concepts to manage links with ease to create an experience that is easy to navigate with thumbs.



Photo credit: <https://www.waze.com/>

App typography will set the tone for everything else. Apps are leading the way when it comes to design and typography. Look at things, such as the popular **Waze**, which features a rounded, almost bubbly sans serif in the logotype and a smaller medium weight sans serif for everything else.

Apps have started to change the design culture in other ways as well. Some of the things designers are using for e-commerce and information

sites started in apps and games. Elements – such as more monotone background and colored type combinations – have long been used to help increase usability and playability in these types of sites.

Prototype mobile apps faster with UXPin (free trial)

10 Resources & Tools

1. [Modular Typography Scale](#)
2. “[Responsive Typography with Sass Maps](#)” by Smashing Magazine
3. [FlowType.JS plugin](#) for managing type based on container width
4. “[Responsive Typography: Using Type Well on the Web](#)” by Jason Pamental
5. [Apple iOS Color and Typography Guidelines](#)
6. [Google Material Design Typography Guidelines](#)
7. [CSS with Vertical Rhythm](#)
8. “[A More Modern Scale for Web Typography](#)” by Typecast
9. [Golden Ratio Typography Calculator](#)
10. “[The 10 Best Google Fonts for Print, Web and Mobile](#)” by Creative Bloq

Flat Mobile Design

There were doubters when flat design started emerging, but this design trend isn't going anywhere anytime soon. Flat design is only getting started and is more popular than ever when it comes to mobile UI design.

The principles behind this design technique are so universally – and aesthetically – pleasing that some of the biggest companies in the digital sphere, including Apple and Google, have adopted design frameworks with flat philosophies in mind.

Flat design works, simply because it is adaptable, which makes it perfect for [mobile-first design](#). Unlike in the early days, where designers argued that flat design had to follow a more strict set of guidelines, flat design is now much more usable with tasteful touches of skeuomorphism (see [Flat Design 2.0](#)).

With that, let's dive into how flat design is a match made in mobile heaven.

First, a Flat Design Primer

By now, flat design has become an almost “household” term. But in case you aren’t 100 percent familiar with the concept let’s take a quick review of what makes for a flat design.

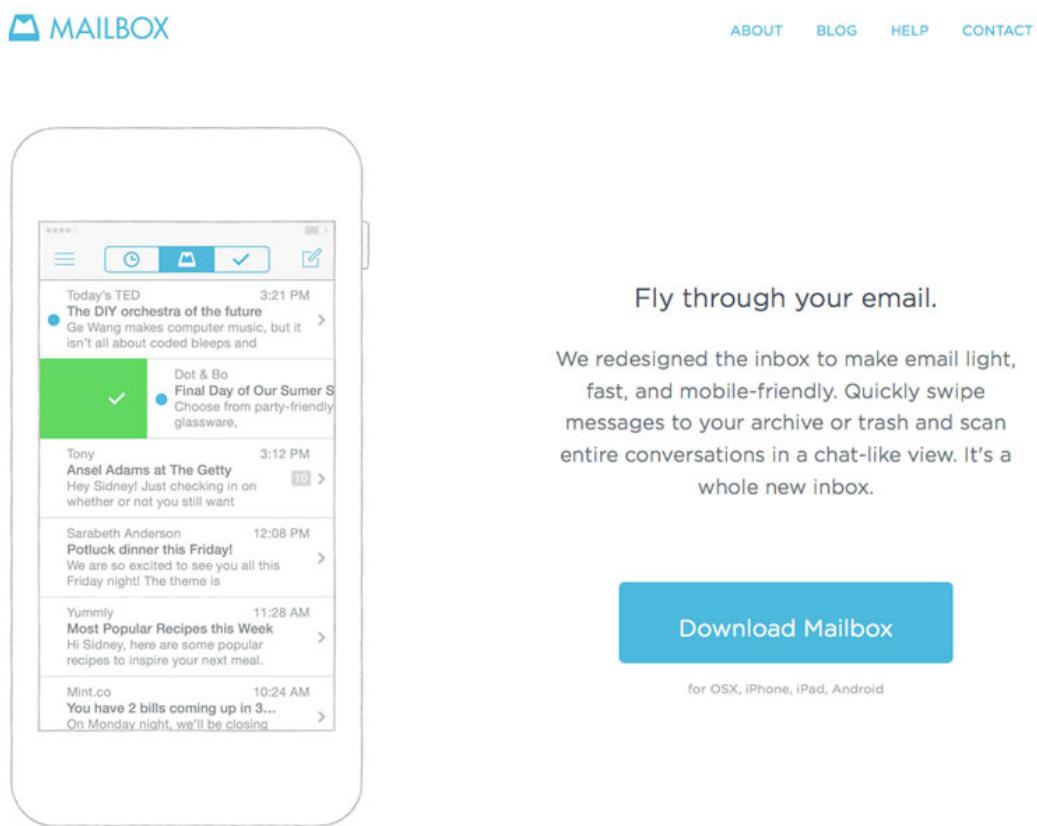


Photo credit: <http://www.mailboxapp.com/>

Flat design started as a reaction against the skeuomorphic aesthetic that dominated mobile design, particularly the [Apple iOS of the early iPhones](#).

Back in those early smartphone days, it was necessary to have things, like buttons, more accurately mirror their physical world counterparts so that we understood how to interact with the device. Nowadays it's old hat and we no longer need metaphorical visual cues everywhere

to navigate our way around a mobile device. Hence, flat design grew in popularity – more so once [Apple made the switch with iOS7](#).

But, as we first described in the free e-book [Web Design Trends 2015-2016](#), flat design is evolving. Some of the characteristics listed below have been part of flat design from the beginning. What has changed and is still continuing to change is that stringency of these design rules. We are now seeing “almost flat” design, or what Ryan Allen terms “[flat 2.0](#)”. All the flat but with some eye-catching enhancements, such as hints of shadows and color.



Photo credit: <http://playspecimen.com/>

The mobile game [Specimen](#) is a good example of flat design with these bits of shadow and color. While the design is certainly flat, it is far from skeuomorphic even though it isn't afraid to embrace a few select elements.

Let's take a closer look at some mobile flat 2.0 characteristics.

- **Lack of over-embellishment:** Design techniques such as drop shadows, gradients and other decorations are usually left off of elements. Although with “flat 2.0,” as seen in the example above, we’re seeing these return in more subtle ways.
- **Distinct color palette:** Flat design colors are characterized by bright, fully saturated hues to make up for the austere visual embellishments. Palettes are commonly monotone or fall to the other end of the spectrum with a wide array of color choices.
- **Simple elements:** Elements are designed in simple shapes and with simple structures, including rectangles, circles and squares. Minimalism is the order of the day. This allows for more content-focused design and “invisible” interfaces that don’t distract from user tasks.
- **Strong focus on typography:** Because flat interfaces are inherently minimalist, typography becomes a dominant visual element. For a bold yet elegant aesthetic, type choices are often simple sans serifs with thin to medium weights. Oversized typography and all caps are popular choices to create more contrast and emphasis.
- **Less is more approach:** Everything about flat design follows a philosophy of “less is more.” These design often lack photographs and flourishes. They include plenty of space, but not a lot of elements.

You can find a full primer on flat design concepts and techniques in our free e-book, *Web UI Trends Present & Future: The Evolution of Flat Design*.

Flat Was Made for Mobile

Why flat? What makes it such a popular option when it comes to mobile design?

Because it works.

Regardless of which device or interface you are designing for, flat design is a tool that helps you check many of the boxes when it comes to the dozens of [mobile design specifications](#) out there. Because of the smaller screens (although they seem to be getting larger, elements such as space and contrast are of utmost concern.

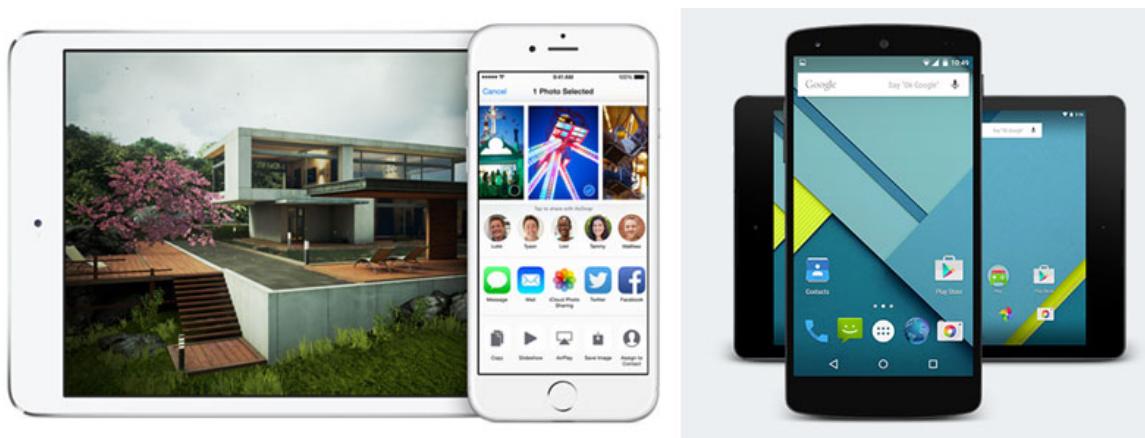


Photo credit: [Apple](#) and [Google](#)

Flat design helps alleviate concerns about limited space, especially since the focus is simplicity and minimalism. This is arguably part of the reason why all the major phone operating systems use a flat design aesthetic. Just look at Apple's iOS, Android and even Microsoft – all flat. The use of this design style and patterns creates a common visual language for users, so that they can move easily from device to device, operating system to operating system.

The most obvious use of flat characteristics are space, typography and color. Both Apple and Google aesthetic guidelines come back to flat design without saying it. Because some of the biggest names in the industry are doing this, designers are inclined to follow suit.

Flat design also fits well into responsive frameworks and [mobile-first design](#). Mobile-first design is a process for fulfilling [responsive](#) or [adaptive design](#) in which you first tackle the smallest screen experience. By scaling up from mobile rather than degrading downwards, you ensure a consistent experience regardless of device.

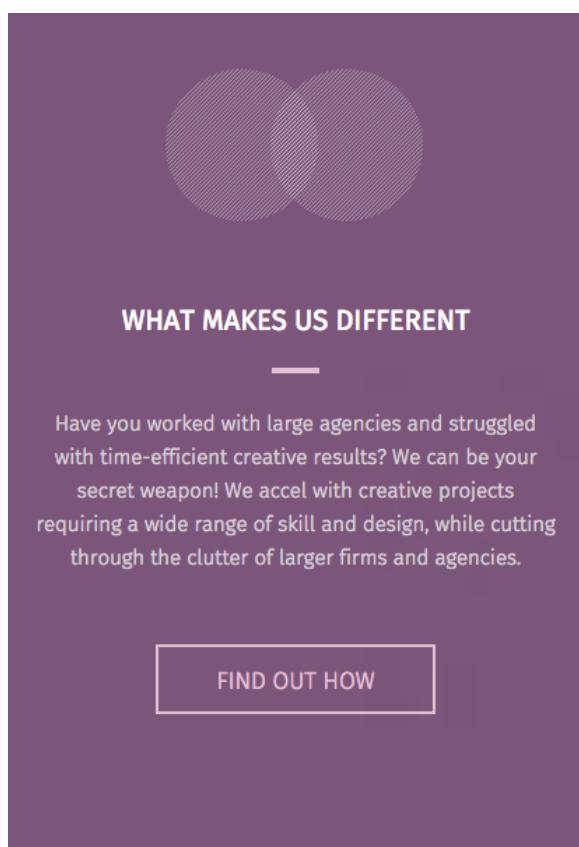


Photo credit: <http://www.micelistudios.com/>

There are usability perks as well. Because flat outlines tend to be rather simple and lightweight, load times aren't as much of a concern. Fewer and smaller elements won't bog down a site. When you arrange a

flat interface using a common cards-style pattern, you also find that the layout is much easier to adapt to multiple breakpoints (you're essentially playing around with rectangular content containers). Users adapted to the style quickly and now also understand how elements such as [ghost buttons](#), icons, simple color and text buttons work.

One last point, flat design has been shown to help with leading users to take action. When a flat button was used [conversions for Underwater Audio jumped 35 percent](#).

Let's briefly recap the advantages of flat design for responsive and mobile-first design:

- Colors appear at a better resolution since they are generated in-browser (very important considering the rise of HD devices and monitors like Retina)
- Content-driven interface is perfect for mobile users with short attention spans. Less obstructive interface allows for faster skimming.
- Simpler to re-arrange for breakpoints. With more skeuomorphic interfaces, you create a “physical” interface that must be broken down and rebuilt at different screen sizes. With flat interfaces, it’s more a game of moving around elements in a fluid medium.
- Easy for users to understand and navigate thanks to the inherent grid-like visual organization
- Speedy loading times since elements are simpler and browsers generate colors instead of downloading to the device

It all comes together for a simple reason – flat design is a trend that has been evolving from Day 1. Designers have found ways to use it for complete frameworks, incorporate flat concepts into other major design styles or use elements of flat design as accent pieces.

The Evolution of Mobile Flat Design

Flat design is not nearly as flat as it was in 2013. As we said before, tweaks made to the concept have made it more functional and aesthetically desirable for designers.

Flat 2.0 is fixing some of the early complaints about flat design and giving designers more breathing room and visual maturity.

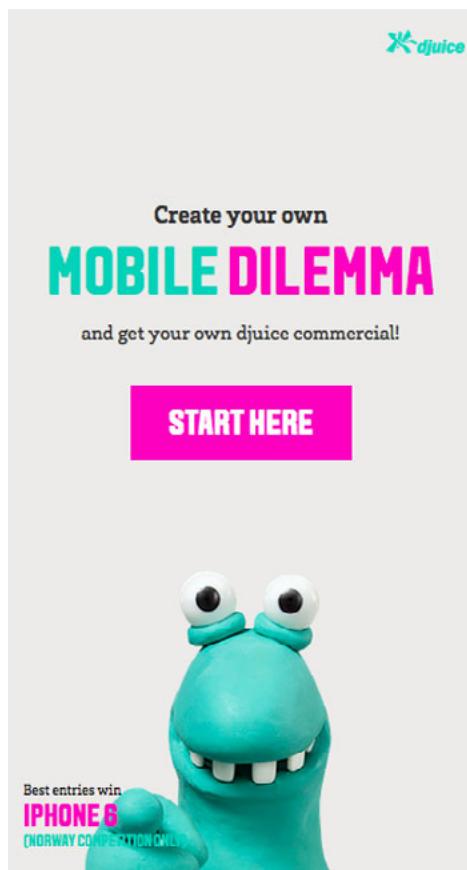


Photo credit: <http://mobildilemma.djuice.no/en>

Flat design really began to open designers up to using more space again. This is vital when it comes to mobile design:

- **Users need room to interact with elements.** They need to be big enough, and separated enough, from other elements so that the user engages with the right element each time. Remember to [design for fat fingers](#).
- **Refocus on readability.** Simple typography always helps improve readability. This might be one of the biggest benefits of the trend long-term. Font selection and contrast is a major topic of consideration with every design project. Is the typeface too thin, too small, too condensed? At the end of the day, if your design isn't readable, then it's not usable.



Works



Photo credit: Omega R

Today's flat design does include some slight embellishment. Subtle gradients and drop shadows are part of many flat outlines. Color palettes have evolved to include more colors and not just those that were deemed flat. Flat design has grown to include elements such as photography and animation as those elements have grown in popularity, as well.

Flat and Other Design Techniques

Flat design has evolved to integrate with a number of other design styles and patterns. It no longer exists in a vacuum.

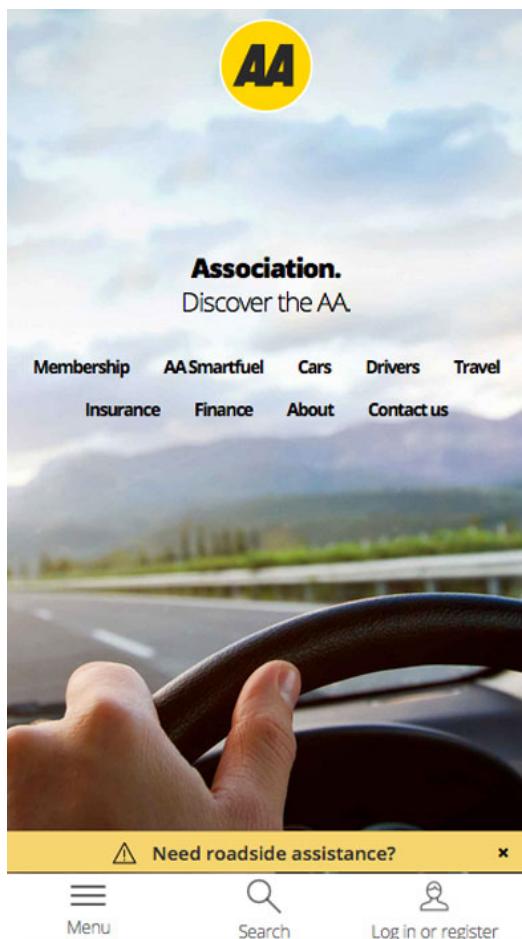


Photo credit: AA

- **Flat and hero images:** Full-screen images are hot. Paired with flat concepts such as bold typography and simple button structure, you get a design that layers flat elements over strong photography. The key to full-screen images is adding simple typography. [Discover the AA](#) (above) uses a strong image with plenty of room for text elements. It also adds a colored bar as a quick link for what is likely their most important tool.



A GLOBAL DESIGN DIRECTION FOR

metrogroup

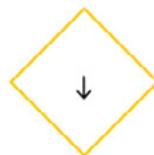


Photo credit: <http://tapunit.com/>

- **Flat & minimalism:** Flat design is a minimal concept in its own right, and works exceptionally well when stripped down even further in minimally-designed frameworks. In the above example from [the Metro Group](#), the white space tells users where to look and what to do. Simple color choices further emphasize calls to action.



Photo credit: <http://www.gude-stoff.de/>

- **Flat and animation:** Many gestures trigger some type of animation, creating a **delightful microinteraction**. From animations as simple as hover effects to interactive elements that move dynamically in game play, animation and flat work together seamlessly. **Gude** (above) provides simple visual feedback by expanding the image when you tap upon it. If you'd like to create your own mobile microinteractions, feel free to play around with our **custom animations editor**. Always follow the **12 rules of animation**.

Benefits of Flat Design for Mobile Interfaces

The benefits of flat design are pretty universal whether you are designing for a mobile site, app or desktop site, as detailed in *The Evolution of Flat Design*.

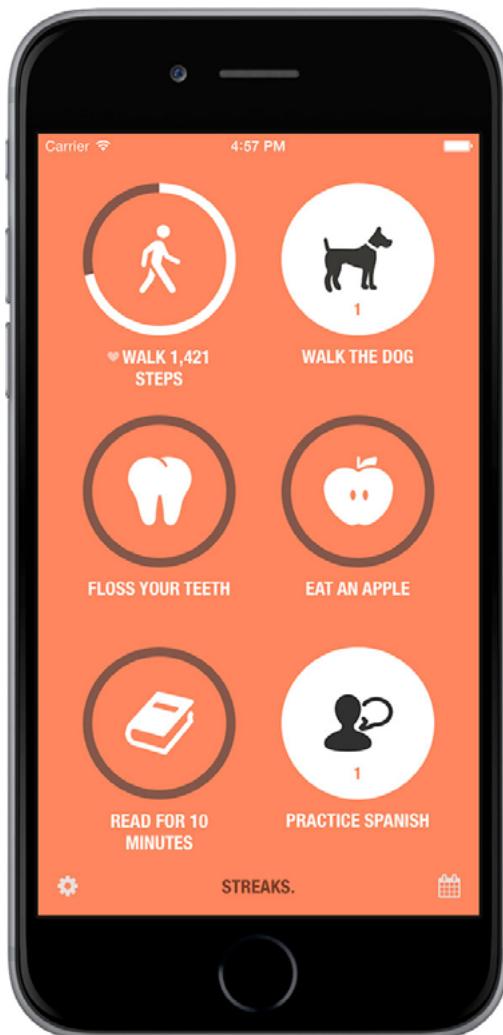


Photo credit: <http://streaksapp.com/>

- The visually organized interface works well with responsive frameworks
- Minimalist, simple style is easy to browse for users
- Clean yet vibrant color palette is highly engaging

- Graphic icons are fun and easy to understand
- Bold lines and shapes express a logical, almost architectural feel
- Simple pieces and elements are quick to load
- Typography is designed for readability

We can expand the list to include a few more distinct benefits specific to mobile devices and users, as well.

- Style translates well to small screens while retaining its impact on bigger devices such as phablets
- Space provides plenty of room for touch and gestures
- Simple buttons make calls to action easy to distinguish
- Designing for vibrant colors and sharp contrast make elements highly visible in different environments (such as outside lighting)
- Planned simplicity for usability, such as designing one call to action for the smaller screen (one design, one click)

What's Next?

Where does flat design for mobile go from here?

With as much as it has changed since taking the design world by storm a couple years ago, flat design will continue to be shaped by other trends in mobile design.

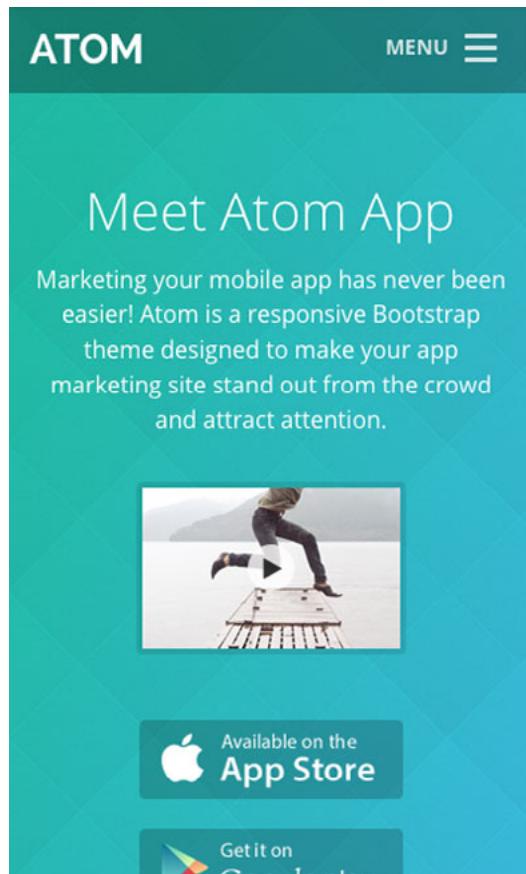


Photo credit: <http://wrapbootstrap.com/preview/WB065NR19>

Designers will create websites and apps that are even more “less flat.” This includes everything from the subtle textures in “flat color” backgrounds to drop shadows or gradients that lean a little more toward skeuomorphism. Less-flat applies more to the overall layout while smaller elements (like icons) remain deeply rooted in pure-flat visuals. Just look at the background for the [Atom App](#) above: the overall color scheme is certainly based on the flat palette, but is formed from individual squares of graduated color.

[Material design](#) will continue to impact how designers approach to element layering along the Z-axis. There are already entire websites and plenty of Dribblers dedicated to creating and reimagining websites with material design.



Dousing The Blaise

Photo credit: BuzzFeed App

For [Buzzfeed](#)'s app (above), you can see how material design gives the interface a more paper-like feel reminiscent of paper magazines. Because each card is “raised” above the grey background, the content feels inviting and fun to click. It’s the perfect blend of skeuomorphic familiarity and flat design elegance.

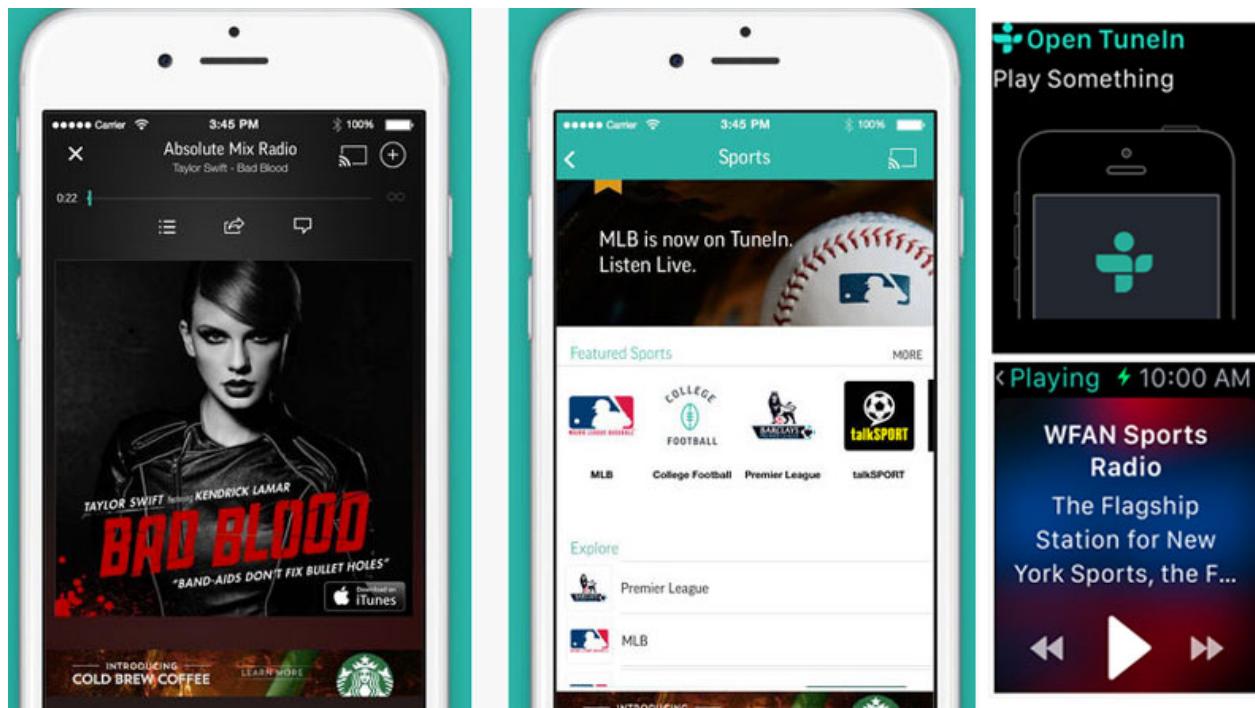


Photo credit: <http://tunein.com/>

And while it seems like many mobile devices are actually getting larger, flat design for mobile applications is getting smaller thanks to wearables. The flat aesthetic is practically made for this super small, super simple screen type since there's zero room for skeuomorphic clutter. Flat might even see more growth as designers try to create wearable, mobile design interfaces that visually correlate to their responsive and app counterparts.

Because flat design is simple – we're going all the way back to one thought, one screen here – it allows wearable users to get all the information they need at one glance. The [TuneIn radio app](#) (above) uses a highly visual interface with flat elements for its main app, but slims ways down for the wearable version with only an icon or simple alert text on the screen.



Photo credit: <http://www.mugakofestival.com/>

In the same way flat elements are used with photos, they can be used with video. More access to high-speed internet connections is making video a fast-growing mobile option. Adding flat elements, buttons or icons to video-based responsive sites or applications is stylish and functional.

The video elements for the [Mugako Festival](#) (above) are fun, fast and flat. (Just take a look at the aqua color paired with a ghost button.) Incorporating video in the background is an easy way to merge these

trends without being overwhelming, and with as much video as we are seeing in website design as a whole, expect plenty more video in app design as well.

Flat design, like everything else in design, will continue to evolve, especially as more and more of us begin to use wearables.

Prototype mobile apps faster with UXPin (free trial)

10 Excellent Resources

1. [Apple iOS Design Guidelines](#)
2. [Android Design Guidelines](#)
3. [Material Up: Daily Material Design Gallery](#)
4. [“10 Principles of Mobile Interface Design” by Creative Bloq](#)
5. [Flat UI Colors](#)
6. [11,136 Flat Graphics](#) by freepik
7. [Free Flat UI Kit for Mobile](#)
8. [Tutorial: How to Design and Animate a Bicycle in Flat Design](#)
9. [*Elegant Web UI Design Techniques: Flat Design & Colors*](#) (free e-book)
10. [Code It: Inspiration for Text Styles and Hover Effects](#)

Delightful Micro-interactions

You understand and have created an app that looks amazing. It has a cool story and the visual design is impressive. But that's not enough – it needs more, something that really connects with users and feels alive.

That's where micro-interactions come in.

Micro-interactions are the secret ingredient when it comes to creating an addictive app. These design moments keep users engaged, create unexpected delight and are almost invisible to everyone but the designer.

You come across micro-interactions hundreds of times a day. Each time you end an alarm, see a text message flash across the screen, are reminded of your turn in a game, skip a song in a music player or even as you change course based on traffic alerts during your morning commute. Every one of these tiny moments forms a micro-interaction. And it's likely that you don't think about any of them, but each one contributes to why you continually use specific apps day after day.

What Are Micro-interactions?

A micro-interaction is any single task-based engagement with a device. Most of these engagements are things users rarely think about when it comes to how they work or look (as long as they are working properly).

As described in Dan Saffer's [excellent book](#) (highly recommended), micro-interactions help fulfill three specific functions:

- Quickly communicate status or feedback
- Visualize the result of an action
- Help the user manipulate something on-screen

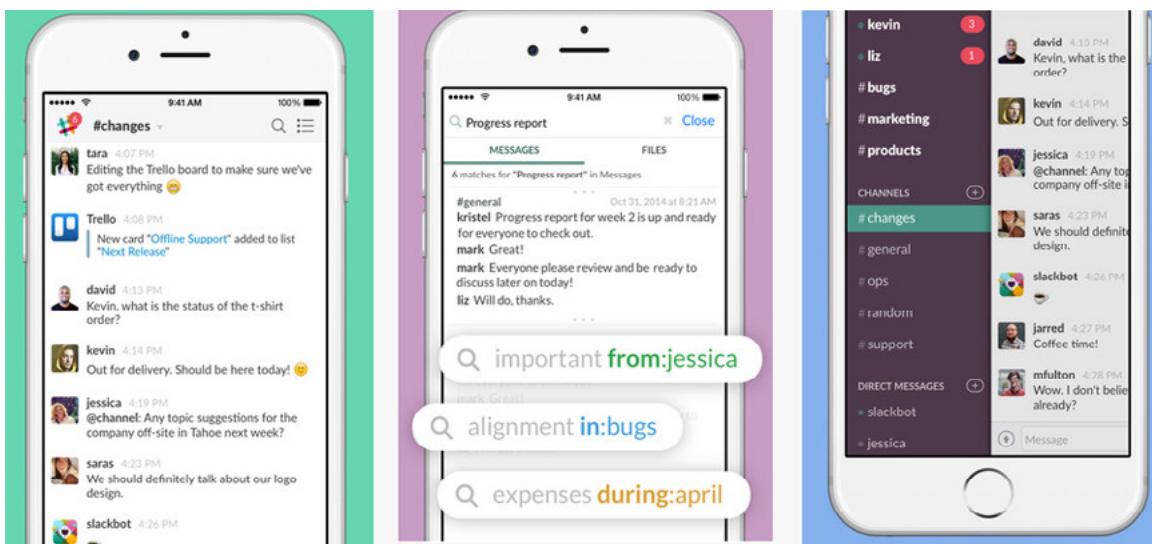


Photo credit: [Slack](#)

[Slack](#), above, is a great app that does all three of these jobs in one package. The app allows users to communicate in a closed-loop chat, attach documents and tag one another. All the while, the app provides

real-time updates (such as marking messages as read) and helps users navigate around. It also uses notifications and other small actions to keep users in sync with what's being communicated.

Microinteractions impact the user and function of the app in a variety of ways:

- Turning things off or on
- Commenting in any digital medium
- Changing a setting or process
- Viewing a notification or message
- Sliding down the “screen” on a mobile device to refresh content
- Interacting with a data element, such as checking the weather
- Accomplishing any single task
- Connecting devices, such as those for multi-player games, or printing from your laptop
- Sharing or liking a photo or video on a website

Simply put, a micro-interaction is an action from the user that triggers another action on the part of the device. Each of the interactions is based on a **human-centered design** concept, where the digital tool works and functions mirrors how a person would actually do something. And that's the secret sauce to usability – interactions that behave as expected and in a “human way.”

What Do Micro-interactions Really Do?

As you can see from the list above – and in no way is it inclusive of every micro-interaction – micro-interactions serve multiple roles. But speaking less specifically, they should always create engagement.

When you are considering how to design a micro-interaction, think not only about the specific action or task at hand but also what it should “do” as part of the overall user experience.



Photo credit: [Elevate](#)

In essence, micro-interactions should make the user feel or do something physically:

- Touch the screen
- Smile with delight
- Learn or understand something
- Connect to another user

The Elevate app above uses each of these components in the game-style learning interface. Users must touch the screen in order to answer the questions and correct answers while nifty animations serve to delight users. The app allows users to train their brains (the pictured game is designed to help you with grammar and usage). Users can share scores and results with friends and other app users. It sounds like a lot for a simple game to do, right? But these are the kinds of things users have come to expect from almost every app.

These actions are at the heart of why micro-interactions work. These tiny, lightweight “digital moments” feel like and are perceived as an emotional linkage. The micro-interaction allows the user to feel something or reach out to another user or object.

Four Elements of a Micro-interaction

At the forefront of micro-interactions is Dan Saffer.

Just Google “micro-interaction” and his name is linked to almost all of the definitive information on the topic in some way. He [wrote the book](#) – quite literally – on this topic. Saffer focuses on a [four-part structure for micro-interactions](#) and it is truly the best way to understand how to create them.

- **Trigger:** Initiates the microinteraction. For example, I click on a heart icon to favorite a page.

- **Rules:** What happens in the interaction. The user cannot “see” the rules, but only understands them through feedback (the next stage). In this case, clicking the heart icon will add that page to the user’s feed.
- **Feedback:** How you know what’s happening in the microinteraction. The heart icon filling with color and bouncing, accompanied by a dissipating “Saved to feed!” message informs users of what happened.
- **Loops and Modes:** Determines the length of microinteraction and how it repeats or evolves with time. For example, the micro-interaction we described now evolves to deliver us content from our favorited page, and might even remind us in 6 months when we first liked the page.

Each of these parts is present in every interaction to create a cycle for how things work. According to Saffer, most people don’t even know or think about micro-interactions unless something goes wrong.

Are You Considering Feedback?

Feedback is the most important part of the micro-interaction cycle. This is the stage of the micro-interaction where the user and interface connect. Feedback determines exactly how a micro-interaction will work.

Think about it this way. You need to get up at 7 a.m. and set an alarm on your phone. What happens when that alarm goes off? Do you get up and turn off the alarm? Or do you hit snooze? This simple action tells the app what micro-interaction comes next – reset the alarm for the next cycle or go off again in 9 minutes. The feedback loop between the user and system is now complete.

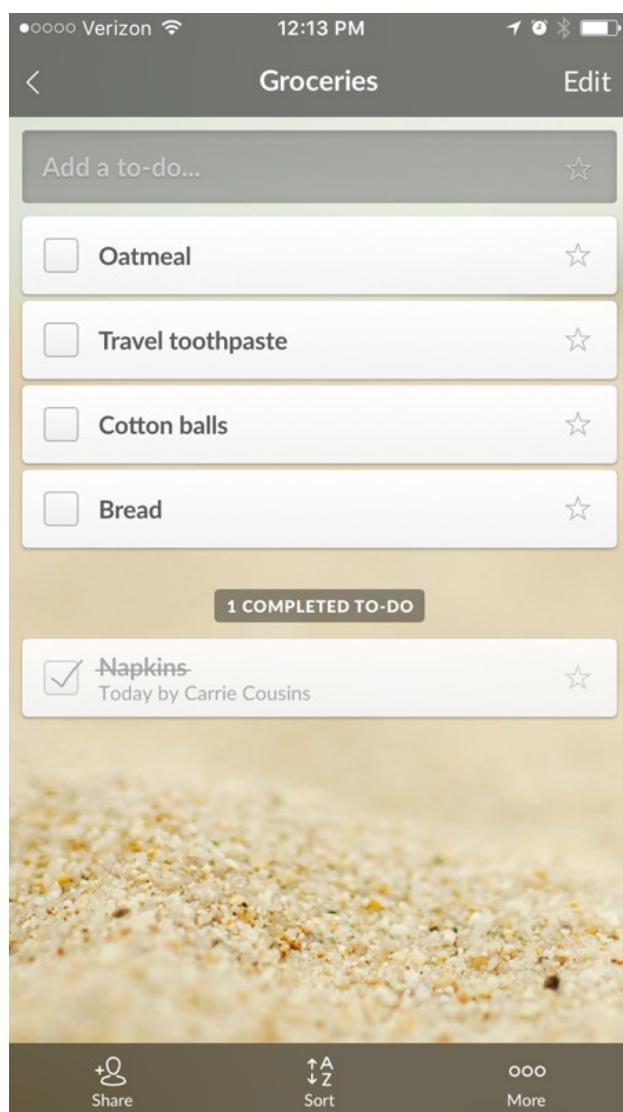


Photo credit: [Wunderlist](#)

Without the initial feedback from the user, the sound of the alarm never stops. It does not reset. The open cycle allows it to work again over and over.

It's the same when checking off items on a to-do list, like the one above from Wunderlist. By ticking a check mark next to an item, you tell the app to trigger the micro-interaction of striking the item off your to-do list. In an instant, you now know which items are complete and which are pending. The satisfaction of completing a task further encourages you to complete more tasks, thus interacting further with the app.

While this is a really simple example, there is a lot to learn from it. Primarily, it shows us that users want to engage with a micro-interaction in a way that clearly creates a next step (the loop) and want it to work intuitively enough that it can evolve over time and usage.

Designing Details is at the Heart of Micro-interaction

The way a micro-interaction is designed and how you handle the details will make or break your projects. Interaction is a must-have design element that you can't ignore.

But how do you design it? What should you think about?

- **Micro-interactions must live through repeated use.** What's fun and quirky on first use might become annoying after 100 uses. Be cautious of gimmicks or odd design cues.
- **Simplicity rules.** Simple type, simple language, simple colors, and simple design. Don't make the design any more complicated than the action.

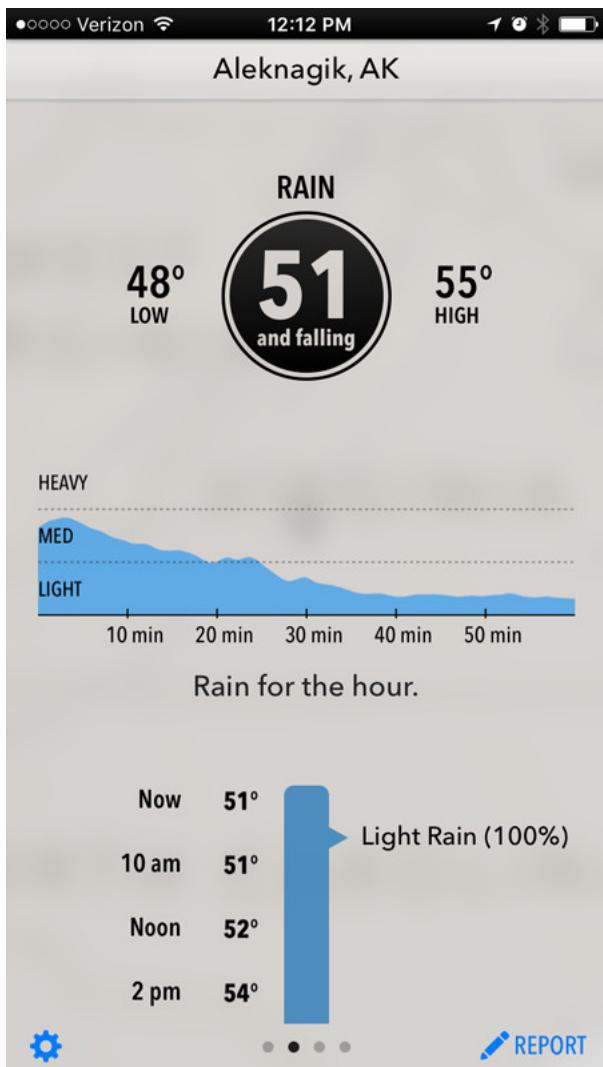


Photo credit: [Dark Sky](#)

- **Give each micro-interaction a human voice.** Text should read [like people talk](#), not like a machine.
- **Pay particular attention to copy.** Any words or language you use must match the tone of the moment. Use only text that is needed and keep it simple, conversational and useful.
- **Add a fun pivot with animation, but don't go crazy.** Consider the bouncing icon in the dock of your MacBook as a program loads. It lets you know the program is responding with a simple animation that doesn't obstruct your current tasks.

- **Create visual harmony with other elements.** If your app has a blue color scheme, micro-interactions should use the same hues so that the visual connection to the parent design or app is there for users. Contrast is your friend, but use it carefully.
- **Don't overthink it.** Overdesigning a micro-interaction can be lethal. Let's go back to the simplicity of the text message notification. Just a simple, singular display on the screen with enough information to be effective – who the message is from, what the message contains and a way to respond.
- **Consider each detail with care.** Because micro-interactions are so small, every element of the design matters. Ensure that every detail, down to the last pixel, is perfected before launch.
- **Think about further adaptations or how subsequent micro-interactions will work.** Does the exact same thing happen every time for every user? Or are there changes to the micro-interaction over time? (Consider that the alarm in the example above gets louder each time after the snooze button is hit.) Smart details set the best micro-interactions apart from everything else.

Ways to Incorporate Micro-interactions into the Design

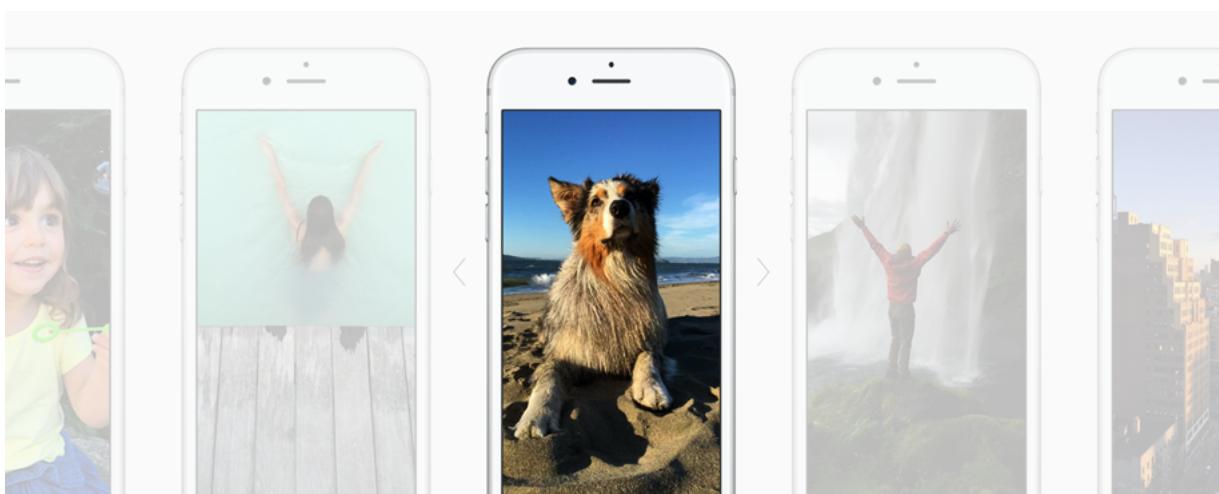
There are plenty of ways to think about and incorporate micro-interactions into the design. Chances are you have actually been doing it for some time now without giving it a lot of thought.

The backbone of every micro-interaction is formed of smartly designed animations. (You can learn more about designing animation from the [free e-book, *Interaction Design and Animations*.](#))



Photo credit: Apple

Let's start with an example of one of the best microinteractions in recent years.



Live Photos add a new element of wonder to your still photos with motion and sound. Relive moments of joy, surprise, and excitement simply by pressing a Live Photo.

Photo credit: Apple

The new camera and [Photos App](#) in Apple's iOS 9 includes a "Live" feature. When a photo is "live," it is recorded almost like a GIF. The

trick – and fun micro-interaction – is how the app retains a hint of that movement.

Say you snap a photo of your puppy and he sticks his tongue out mid photo. In the gallery, the photo is completely static. But when you press on the photo, the screen blurs for a moment as it triggers the “Live” photo, showing you a few seconds of the dog’s goofiness (and even sounds). GIFs and videos are nothing new, but the hybrid execution here is quite unique. In fact, if you swipe between photos, parts of the images will also animate in the transition.

By creating a micro-interaction that’s simpler than a video (just hold to play a loop) yet more immersive than a GIF, Live Photos brings the user back in time to relive a complete moment. The gallery is more than just a carousel, it becomes a moving recollection of one’s memories and feelings. Just by restructuring the standard photo gallery microinteraction, Apple actually triggers quite a transformative experience.

Of course, there are plenty of other things you can do to incorporate micro-interactions into your apps. Two of the most common ways are through the use of notifications and animated buttons or links. Each of these tools can improve engagement and usability.

You need to think about the user first when determining how to design micro-interactions. The key concepts behind micro-interaction usability are:

- One touch = one action (such as a share, or triggering a photo animation in our previous example).
- Anticipate what the user will want next. In the Live Photo example, the animated photo transition continues the micro-interaction as users scroll through photos. Apple knows that people will continue browsing after triggering the first Live Photo, so it maintains consistency in the experience.

Once you have that figured out, you have to make the micro-interactions practically invisible. We can't reiterate this enough: micro-interactions must feel invisible. Like a good magic trick, users should not see the cycle of trigger, rules, feedback and loops. Users should only know that the app does what they want when they want it. If you can accomplish that feeling of "[passive magic](#)," then you stand a better chance of users repeatedly engaging the microinteraction.

What's Next?

We are just at the beginning of designing micro-interactions and they will only grow in importance and scope. While many of the best examples are from mobile applications, micro-interaction usage will be a prominent design element for smaller apps such as wearables and robotic apps used in machines in the home.

Not only are you connecting with micro-interactions on a phone, but the growing number of wearable devices is making micro-interactions

just as important on smaller screens. Arguably micro-interactions are even more important on wearables because the screen is so small that every interaction is a single-step process.

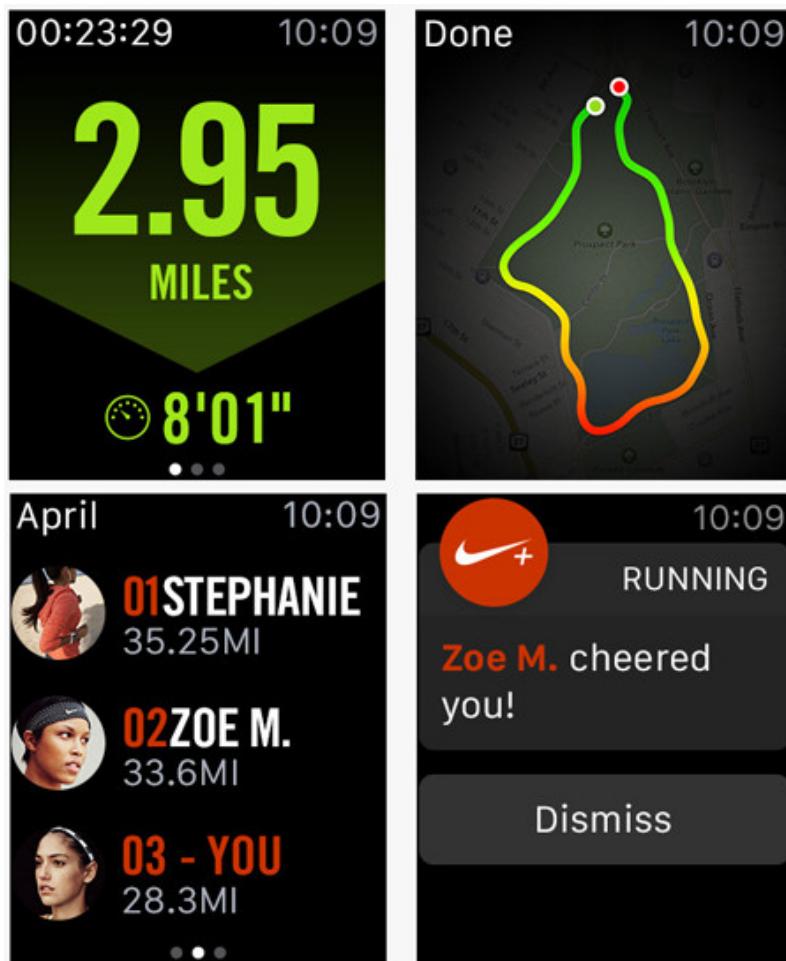


Photo credit: Nike Running

Consider the [Nike Running](#) app. Not only does it cross-integrate with other apps on your phone (such as [Health](#) and [Spotify](#)), it provides a full micro-interaction experience on the Apple Watch as well. From live tracking mileage to getting cheers from friends, micro-interactions make the app usable on the wearable platform.

Robots will rule with integrated micro-interactions. From home security to comfort control to cleaning, interactive robots and connected

apps will create an experience that is unlike almost anything we've seen to date. And it will all "work" based on micro-interactions.

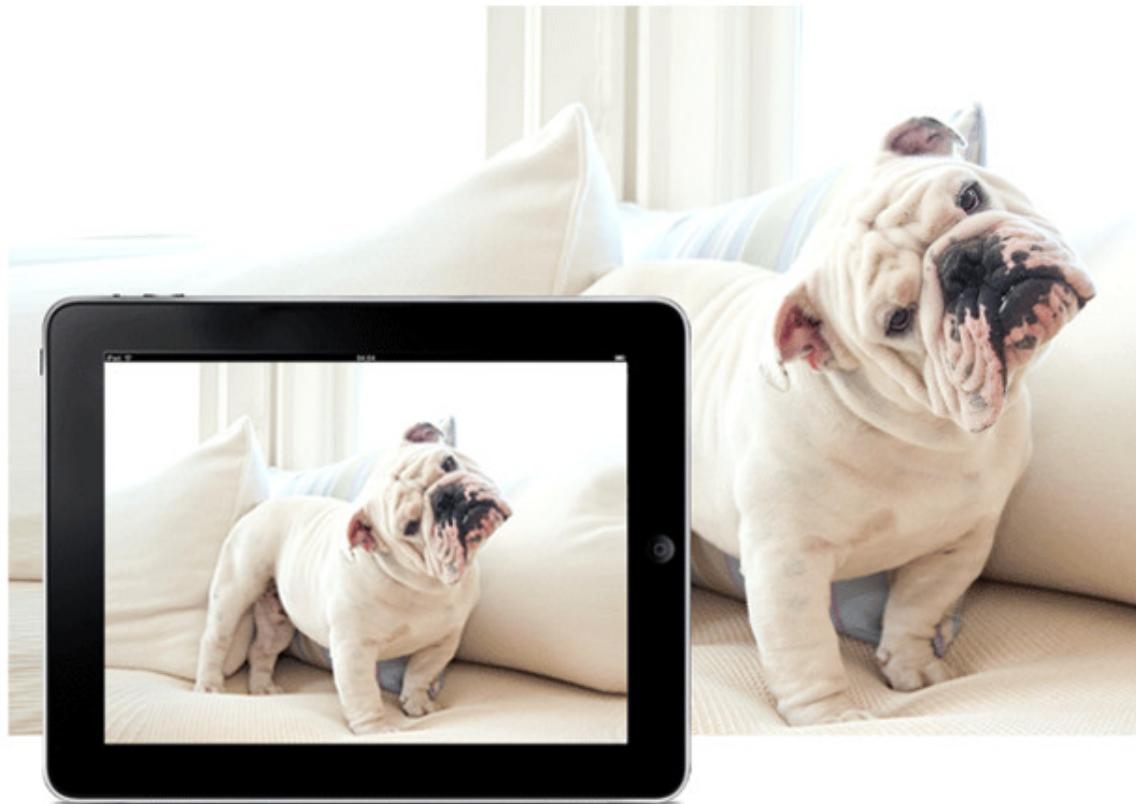


Photo credit: Cox Homelife

Home security and cable television provider [Cox](#) is one of the many companies testing these systems in homes now. With an integrated robot-style home, you can turn off the lights and see video from inside your home. (The system can tell you if your pet is moving around the house.)

The [Nest](#) thermostat also uses micro-interactions as a tool that "learns" your preferred temperature settings to make your home as comfortable as possible. Each of these robotic-style tools connects with an app on your phone to provide feedback and trigger better interaction the more you use it.

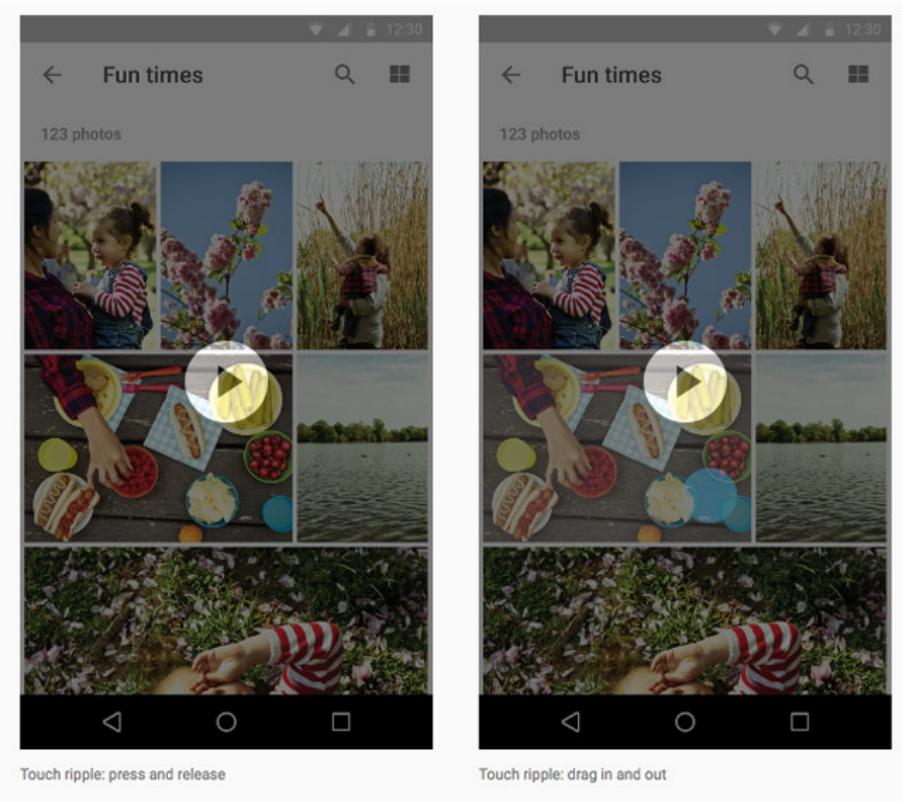


Photo credit: Google Material Design

Micro-interactions will also continue to evolve in the way they look and interact. Some of the best practical advice on micro-interactions can be found in the documentation for [Google's Material Design](#). The concept goes beyond function, but also explains the element of touch, gesturing and even aesthetics.

“Responsive interaction encourages deeper exploration of an app by creating timely, logical, and delightful screen reactions to user input. Each interaction is thoughtful, perhaps whimsical, but never distracting. What will happen if I touch this screen? And then this icon?”

Micro-interactions will also improve further as apps better connect with the physical world. It will become even more common for micro-interactions to trigger when users come in contact with [beacons](#).

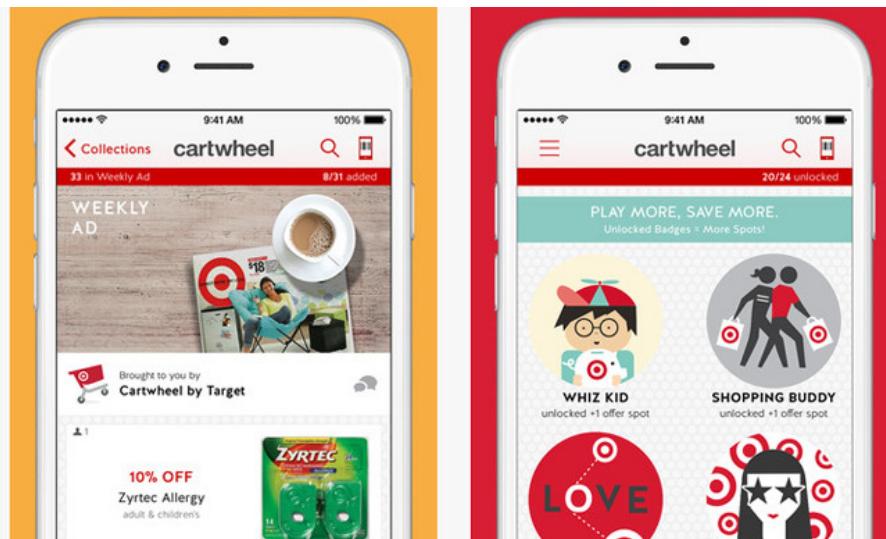


Photo credit: Target Cartwheel

Think about the 2002 movie “Minority Report:” As Tom Cruise’s character walks through a shopping mall, billboards talk directly to him based on his wants. This can happen in apps as well – and some places such as [Target](#) are testing it out with geolocation tools – with deals, coupons and specials delivered via notification to your phone as you arrive at a particular location.

Takeaways

Micro-interactions are the key component to an app design that people want to use. They help create engagement, contribute to function and delight users. Pull it all together and you need to create tiny moments that users don’t see, but need or want so that your app is a vital part of their daily lives.

[Design mobile interactions in UXPin \(free trial\)](#)

10 Resources and Tools

1. Video: “[The Power of Thinking Small](#)” by Smart Design
2. Case Study: “[/r/thebutton as a Microinteraction](#)”
3. [UXPin Animations Editor](#)
4. [Google Material Design: Authentic Motion](#)
5. “[Microinteractions: Designing with Details](#)” by Dan Saffer
6. “[Why Human-Centered Design Matters](#)” by Wired Magazine
7. [Tuts+: Signals and Microinteractions for Smartwatches](#)
8. “[The Ultimate Guide to Web Animation](#)” by Webdesigner Depot
9. [How to Create Animated Buttons with CSS](#)
10. Free e-book: *[Interaction Design Best Practices](#)*

Bite-Sized Cards

2015 is the Year of the Card when it comes to app design.. Screen-size cards are everywhere, from websites to native apps. and are designed to look like their physical counterparts. It's an easy way for you to shuffle through a series of digital containers with the flick of a thumb.

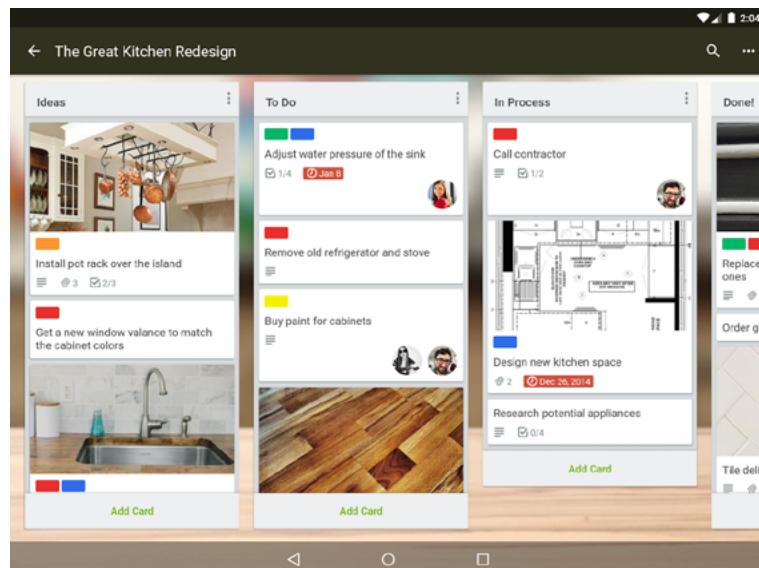


Photo credit: [Trello](#)

Regardless of how you feel about the concept, cards are here to stay. Frankly, cards are a style that seems just made for apps. There are so many apps using cards nowadays, that you almost don't even think about it.

Let's do a little experiment. Pick up your phone. Open the first 10 apps you see. How many feature cards?

Now that we've established how cards dominate mobile design, let's dive into how you actually use them.

Card-Style Interfaces 101

Card layouts put information – images, text, buttons, links, etc. – into a series of rectangular containers. These blocks can be layered or moved and tend to adjust to the size of the screen, stacking and falling into columns if you turn your phone on its side.

Cards are neat little containers for information. And as pointed out in the e-book [*Web UI Trends Present & Future: Card Design Patterns*](#), it's best to think of each card as a singular thought, or primary action.



Photo credit: Apple AirDrop

Consider the common [AirDrop](#) function on Apple devices. When you have incoming data, a card pops up with a notification to accept or decline – a single action no matter the choice you make. The action works in the same way whether you are on a phone, tablet or computer, which means that the user easily understands what is supposed to happen and how to use the design.

While the proliferation of cards is fairly recent, the design is not all that new. [Pinterest](#) really gave first life to cards as a dominant design technique. But plenty of other companies soon followed.



Photo credit: [Tinder](#)

One of the reasons cards became a popular design choice is their [compatibility with mobile screens](#). The digestible chunks are a perfect match for most mobile user scenarios while the rectangular aesthetic also works well for the UI design.

Just think about the design of a card: it's almost the exact shape and size of a mobile phone screen. (Not exact dimensions because of all the different models available, but a good representation based on aspect ratio.)

Card-style design lies at the intersection of design for desktops and mobile devices, bridging bridges the gap between interaction and usability. As we first described in *Mobile UI Patterns*, cards create a consistent experience regardless of device. This is even more true when thinking about responsive design since cards act as “content containers” that easily scale up or down (like restacking boxes).

Two Distinct Uses for Card-Style Design

When it comes to cards, they are most commonly being used in one of two ways: as the interface or as a distraction within the interface (often in the form of advertising).

1. Cards as the Interface

Sometimes you don't even see the cards in the design since they're perfectly sized to the screen. But you can still identify the techniques if you pay close attention.

Card-style interfaces are often one-touch elements. Tap or swipe anywhere on the screen for an action. Cards are also used as a reward in gaming interfaces.

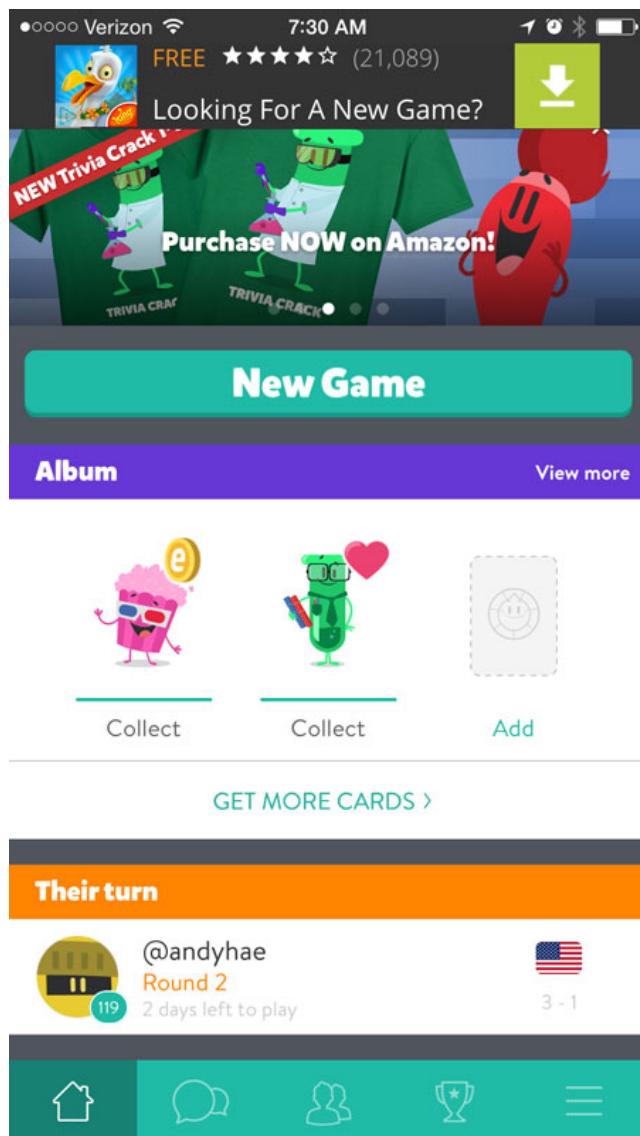


Photo credit: Trivia Crack

Take the popular [Trivia Crack](#), for example, the home screen is a series of cards that organizes each game and opponent. It also embeds an advertisement (cleverly disguised as a card) and in-app card collection as another level to the game. The card stacking makes the game easy to use and understand.

2. Cards as the Distraction

Cards are also a very common style of mobile and in-app advertising, with elements that drop down to cover all or most of the

screen. Unlike the interface-style card, these cards contain two links – one very large and one very small. The large link gets you to the product being advertised. The tiny link (which is often difficult to activate) takes you back to the original interface.

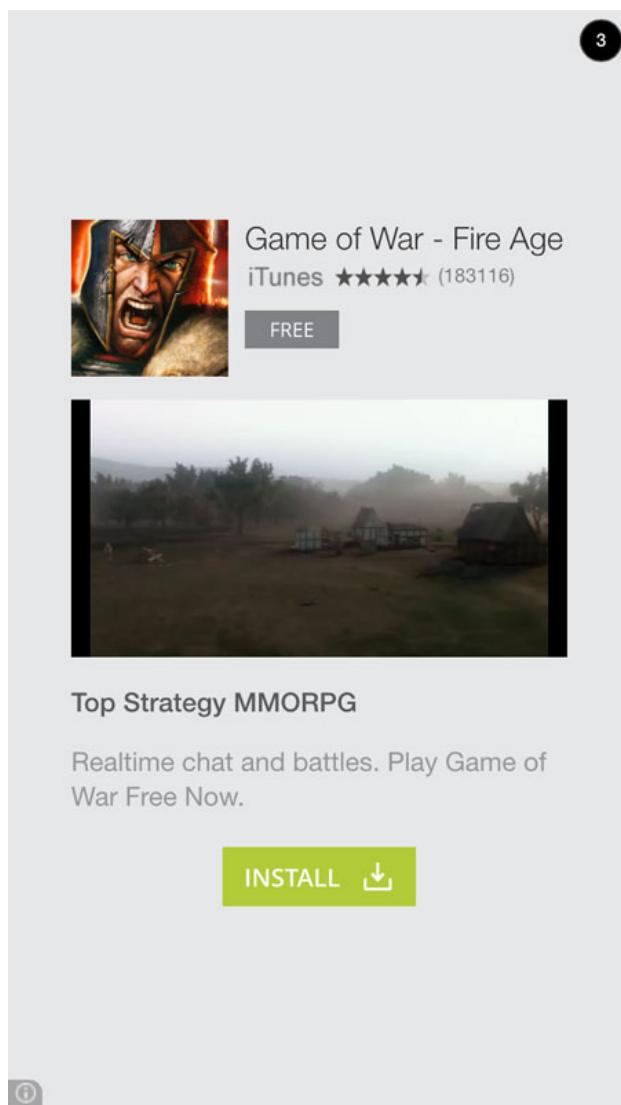


Photo credit: Trivia Crack in-app advertising

Let's keep thinking about [Trivia Crack](#). After your turn is complete, an ad fills the screen. It too is a card. These advertising containers can include a static image, sound, video, and a variety of other bits of information to help persuade you to engage with the paid card content.

From a UX standpoint, the advertising feels more integrated within the overall interface. Even though the ad might occupy the screen, the transition is less jarring since you're just layering cards on top of each other.

Cards help users browse information quickly and provide direct business value with visually consistent advertising.

Cards Work Well With Content

Cards are design “containers” that can hold almost anything. Because cards can work with various types of content, they are perfect for content-heavy sites and apps. Nothing gets left out when creating this type of universal framework.

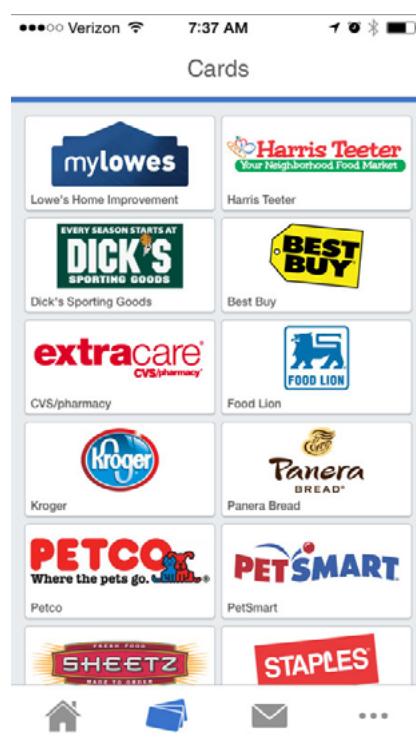


Photo credit: Card Star

Think of all the different elements cards can contain:

- Photos
- Text
- Video
- Coupons
- Music
- Payment information
- Signups or forms
- Game data
- Social media streams or sharing
- Rewards information
- Links
- Combinations of these elements

Placing content in cards makes it digestible for users, who can quickly scan and swipe. When used in a typical layout, like the [Card Star](#) example above, cards are equal players on the screen, neither one dominating the other. They provide variety without clutter, allowing users the power to engage in any way they want.

Cards and Behavior Patterns

Cards are made for thumbs. It sounds very primitive, right? But it is a key part of the popularity and usability of cards when it comes to mobile app design.

What makes cards easy for users is that this digital interface mirrors reality. Think about how you would handle a deck of physical playing cards. You can stack them, expand them, turn them over, spread them out, fold them and put them away in another container. Digital cards behave in the same ways, making a comfortable experience for users. Users don't have to think about how to make things work.

The design can then take this physical idea of cards to the next level when thinking about apps in the digital sphere. The metaphor transfers over seamlessly.

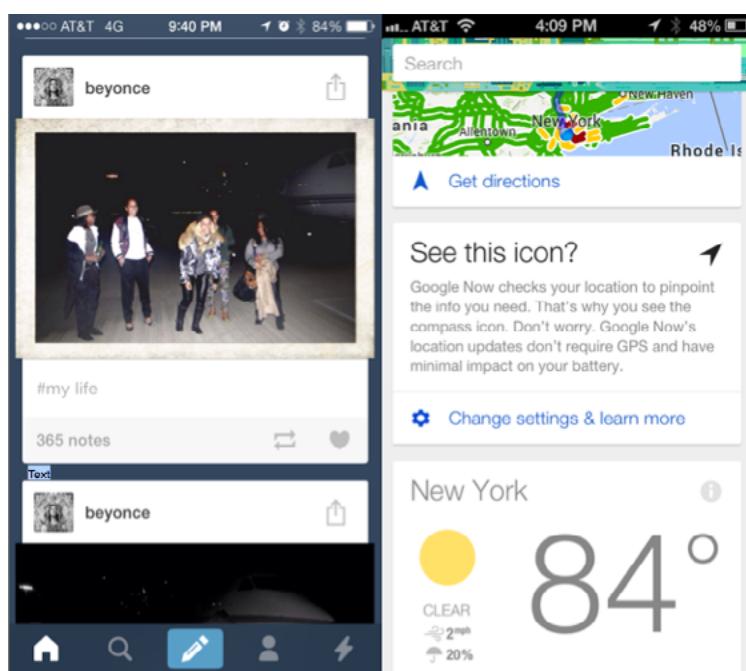


Photo credit: [Tumblr](#) (left), [Google Now](#) (right)

When users interact with cards they fall into a handful of behavior patterns, according to [a great talk on cards](#) by UI designer Chris Tse. Cards tend to do one of three things: record information, tease users with information or alert users to information.

That further breaks down to different types of containers for each of these card elements:

- **Narrative:** Cards appear in a stream, creating a natural timeline of events. Think about how [Medium](#) uses cards to present a quick overview, then deliver the details of that story in a linear flow.

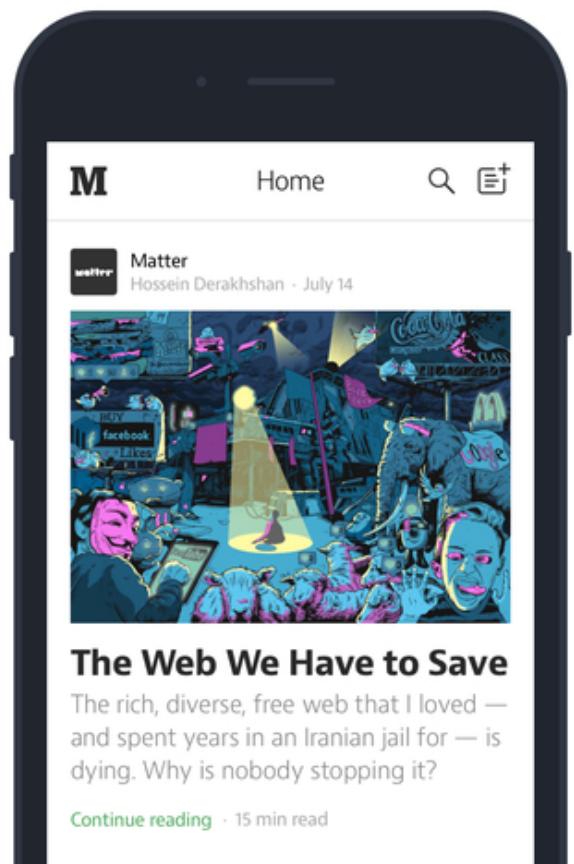


Photo credit: Medium

- **Discovery:** Cards allow relevant content to naturally reveal itself. When presented in a grid or stream layout with fade-in effects, the

cards feel fun and inviting. Take a look at Spotify's card pattern below: as you swipe left or right, you reveal songs that match your tastes.

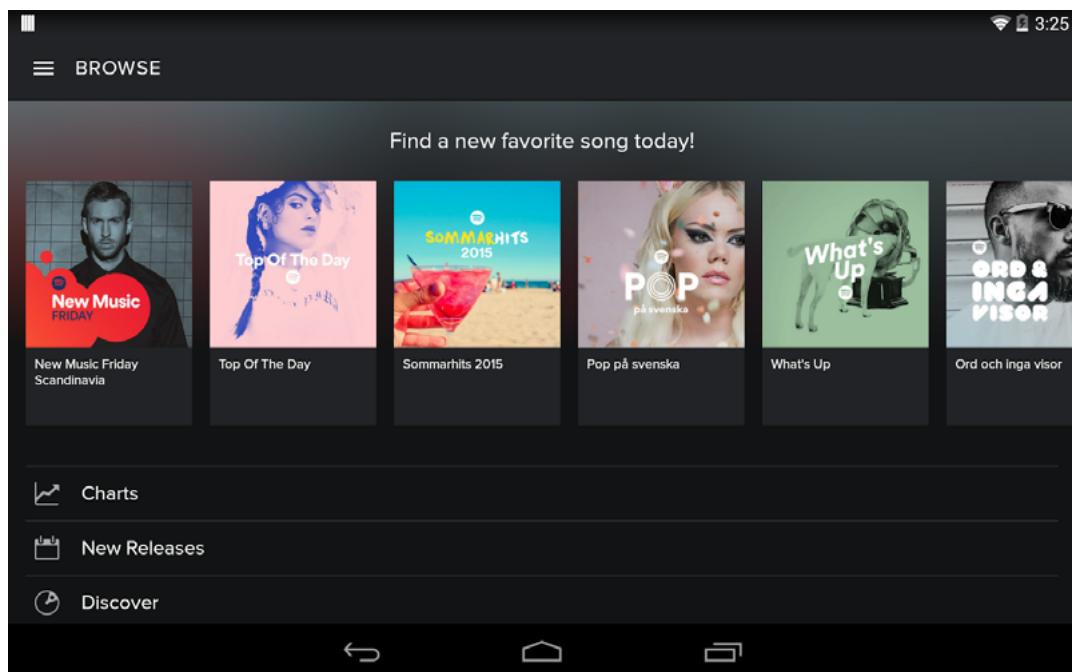


Photo credit: [Spotify Android App](#)

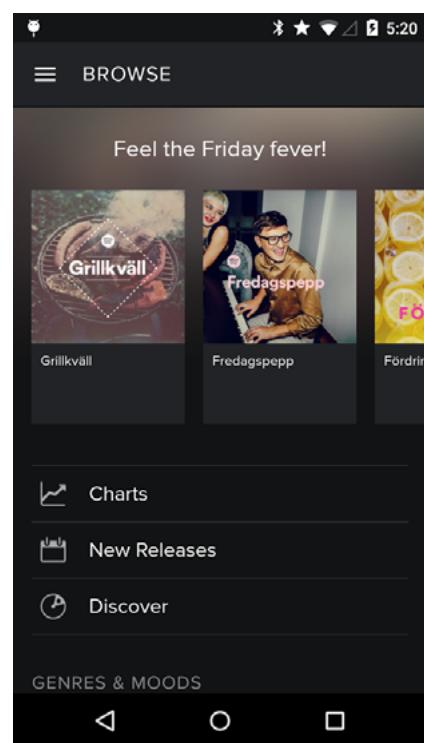


Photo credit: [Spotify Android App](#)

- **Conversation:** Since cards are self-contained, they're perfect for representing ongoing conversations. In the below example from [WeChat](#), notice how the interface uses the [Law of Proximity](#) to create the card with close spacing between profile picture and text. The two pieces aren't physically connected, but our mind helps close the gap.

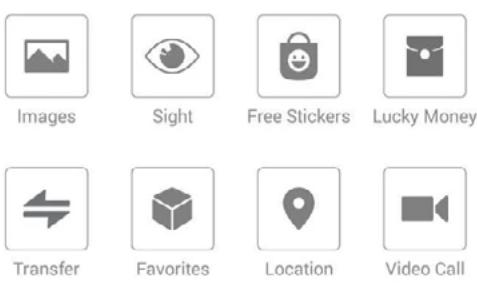
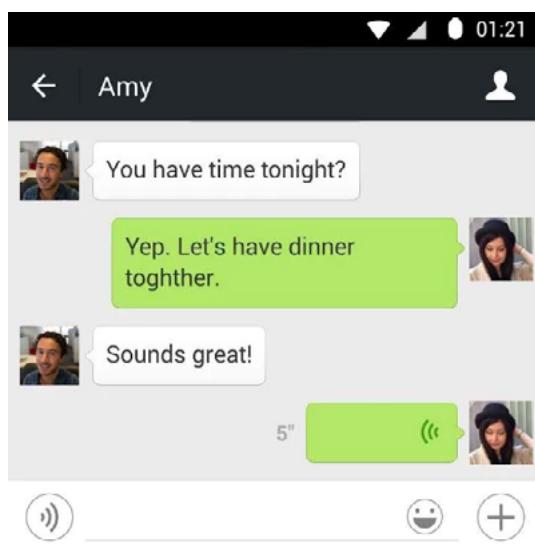


Photo credit: [WeChat](#)

- **Workflow:** Cards are easily categorized for a quick to-do list of tasks. With [Evernote](#), for example, you can create cards that each represent a note or to-do item. As you erase them, the remaining cards rearrange themselves naturally back into position.

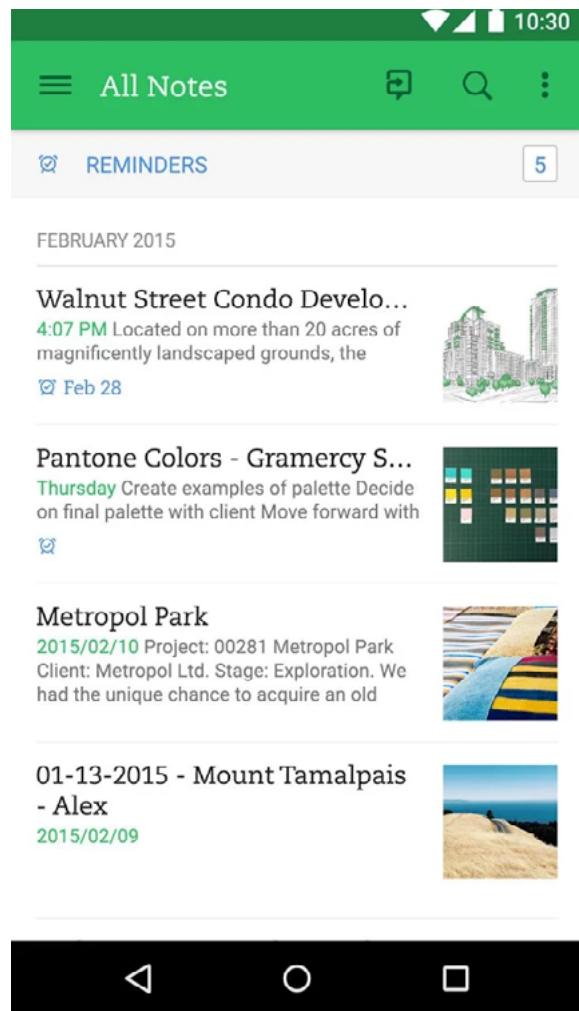


Photo credit: Evernote

Now let's think about cards from a multi-device standpoint.. While cards often live within the container of a particular app, they can also move between apps, devices and users. (Think back to the Air-Drop example, where one user transmits a card of information to another user.)

Cards Are Easier to Organize

Like their physical counterpart, cards are easy to organize for both designers and users.. When it comes to the actual design of cards, you'll need to make a few important decisions.

What size will the card be? And what type of aesthetic are you trying to create?

When it comes to container size, the most popular options are: small, summary-style cards; a medium-sized option that fills part of the screen or allows for multiple cards; a full-screen card; or a card overlay that pops up on top of other elements.

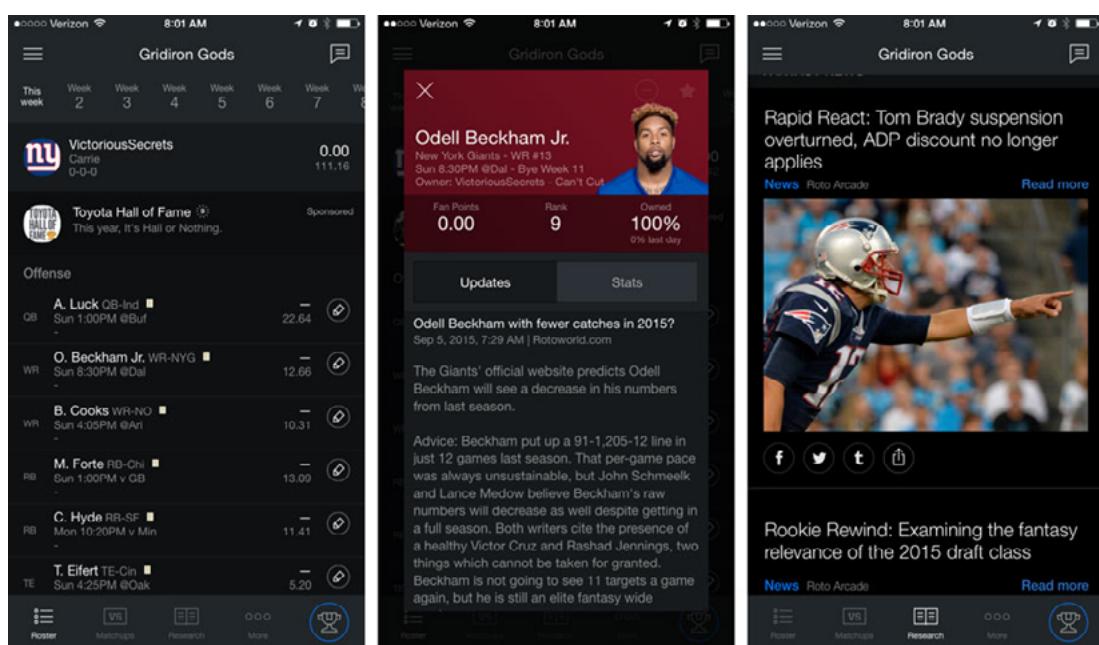


Photo credit: Yahoo! Fantasy

The [Yahoo! Fantasy](#) app actually delivers cards in each of these ways. The first screen shows a player roster, where each player name is a card. Tap a name and a new card pops up with detailed player in-

formation in an almost full-screen layered style. Switch to the news stream for full screen stacking cards that take you to recent headlines, pertaining to your league or team.

Each of these different cards makes it easy for the user to know what content they're digesting and where they're in the app. Layered cards, for example, are understood as extra information on the screen (such as the detailed player card above).

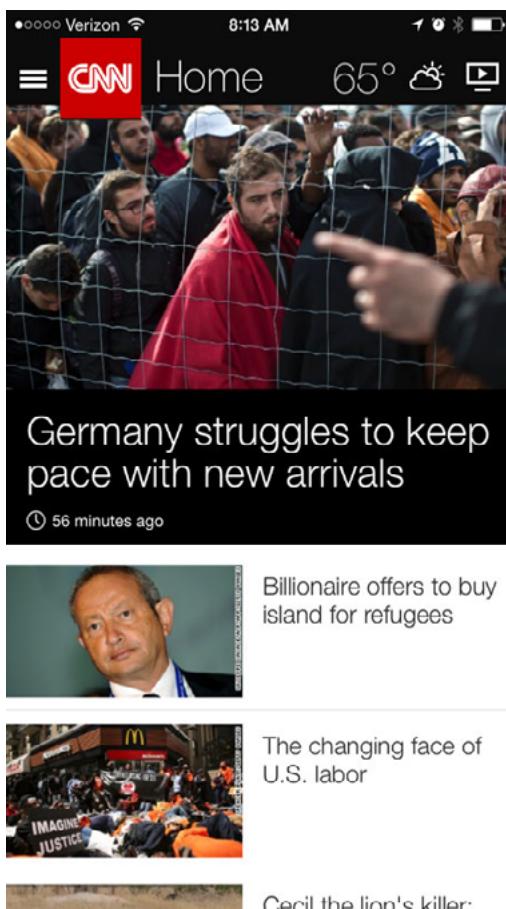


Photo credit: CNN

Cards have evolved aesthetically as well. While the overall trend is moving to less “gimmicky” card visuals, it is important to understand the evolution of the design (and why some of these patterns continue to live on).

- Pins were made popular by Pinterest and are still quite popular although they often result in designs with similar aesthetics.

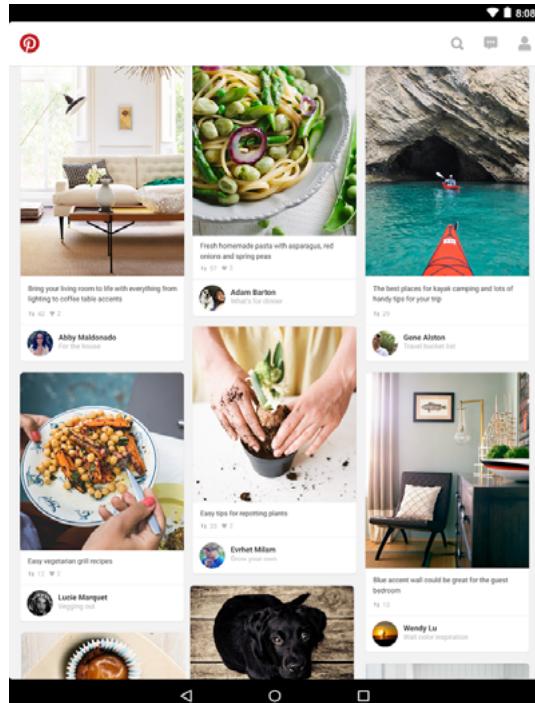


Photo credit: Pinterest

- Metro (from Microsoft) and flat-style cards were some of the earliest uses of cards specifically for apps and mobile devices. While Metro-style cards aren't used very much anymore, flat in particular continues to grow and evolve as a popular card-style option.

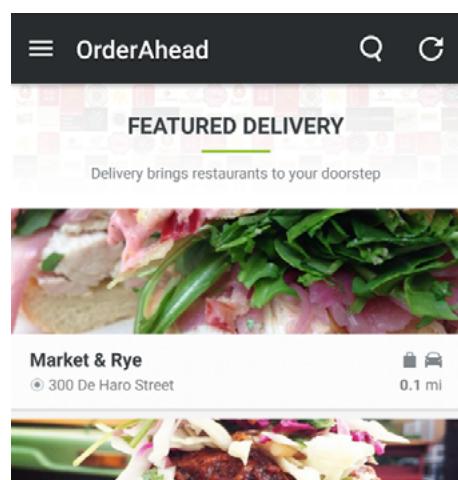


Photo credit: Order Ahead

- Grid or masonry styles are one of those classics that just seem to work. Cards in a neatly packaged container allows flexibility.

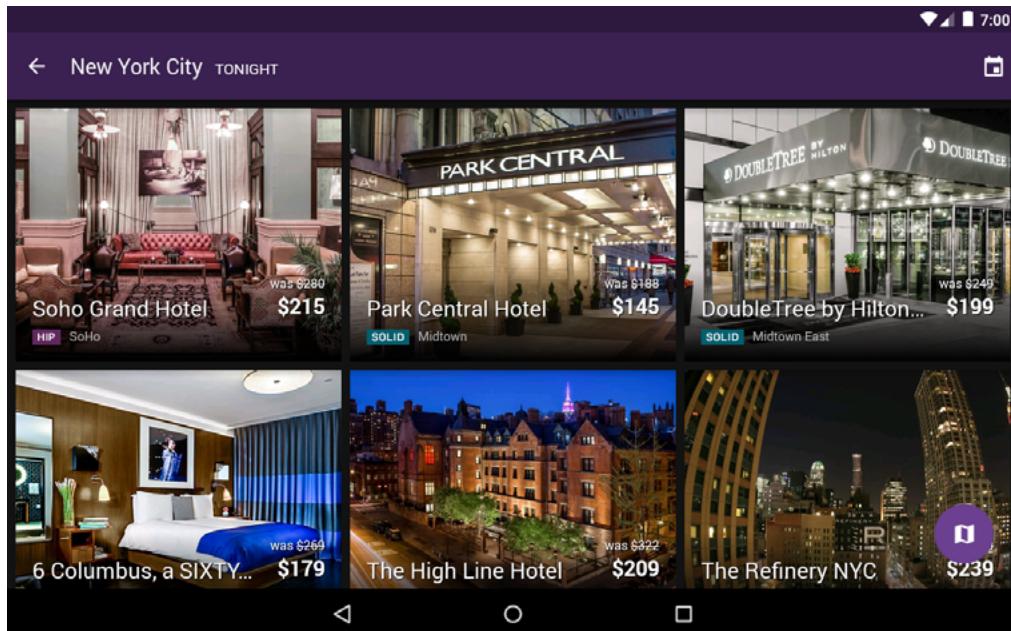


Photo credit: Hotel Tonight

- Magazine-style cards are starting to emerge even more in apps, especially those for news sites or where there is a lot of text to display. [Flipboard](#), [CNN](#) and [Newsify](#) all use this style.

See everything you like in one place.

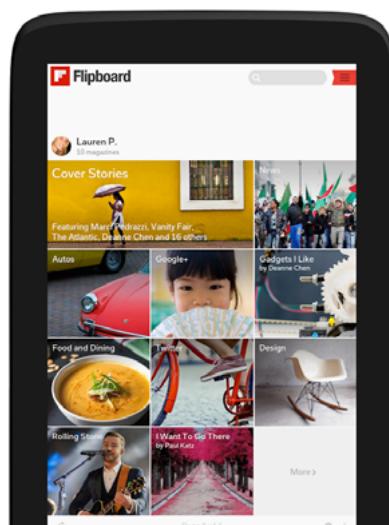


Photo credit: Flipboard

Thinking Aesthetically

Finally, what makes cards work is good design and great usability. Because of all the content options available with card-style design, you need to be an expert in everything from typography to color, image use, and cropping. Design theory is your best friend when planning and executing this type of framework.

UX designer Erik D. Kennedy wrote a fantastic two-part article ([part one](#) and [part two](#)), that breaks down the basics for in a practical manner. Here are his rules that we think apply most to card pattern design:

- **Understand the physical properties of lighting:** Consider carefully how you use shadows and gradients to make elements feel “real.” This is especially important when it comes to card design. If the shadows are cast at all corners and sides, then the illusion of it being a physical element is ruined.
- **Ensure the UI works in black & white:** Design without color first. This will allow you to focus on what’s important – usability and content. According to Kennedy, you should add color last and only with purpose.
- **Apply white space generously:** Give your cards some space to inhabit, then slowly scale back. As we described in the [Zen of White Space for Web Design](#), negative space is your ally in organizing and separating elements.

- **Master the art of layering text:** This can be tricky. Be sure to use a clear, crisp image for the background. To ensure text looks good, you can use a dark overlay, put the text in a box, or blur the background image.
- **Know how to create contrast with typography:** Draw users' attention with the use of either big, bold text or smaller text with less weight. Simply typography often works best when it comes to cards. A sans-serif typeface with medium weight for the card headlines and normal weight for body copy.

By giving cards a bit of aesthetic polish, your card design feels familiar yet still creative. Elements like shadows go a long way in making users relate to the cards as they would with their physical counterparts.

What's Next for Mobile Card Design?

You are probably getting a good feel for why card-style design is increasing in popularity. And it's a trend that won't die out any time soon.. That means more card-style apps and interfaces, including designs that use more layered cards, cards that look less card-like, a rebirth of flat cards and strong use of cards for sites with a lot of content.

Layered cards using [Material Design](#) characteristics are going to hit us in a major way. Layers will emerge in two ways:

- As subtle container elements, such as shading or shadows that set cards apart from the container element.
- As stacked card elements that you can swipe through on the screen, rather than scroll.

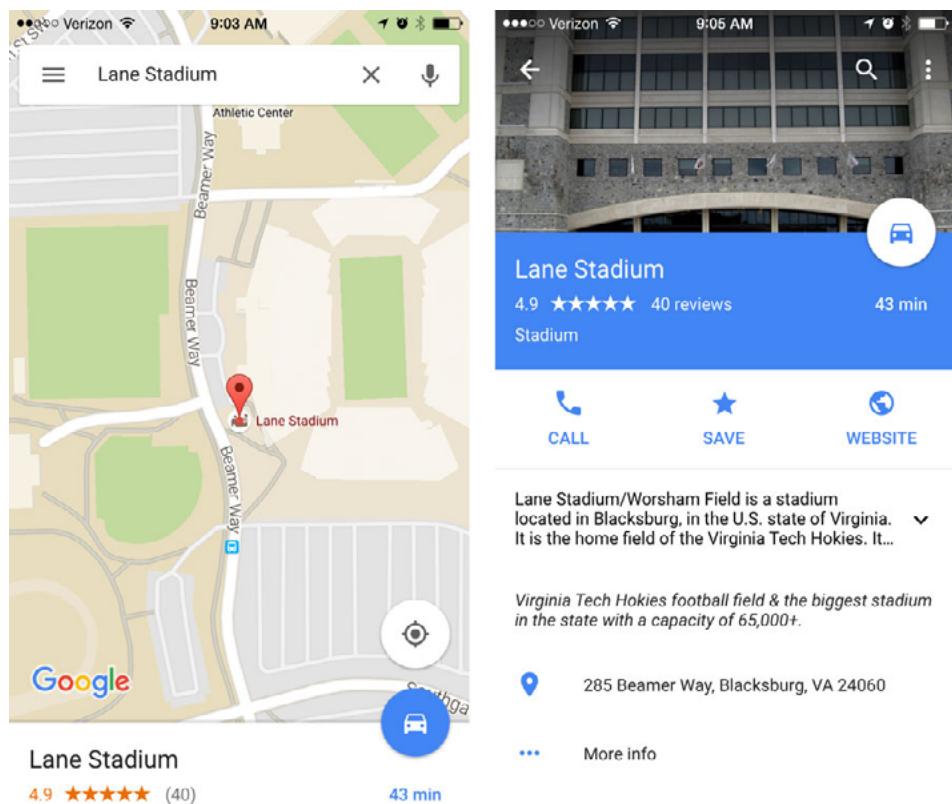


Photo credit: [Google Maps](#)

[Google Maps](#) uses layered cards to help users navigate. Look at the example above, which contains a base map of a location and two distinct cards, one at the top and one at the bottom. The top card is the identifier and helps you map the location, while the bottom card expands for site-specific details that can be swiped over the initial map.

Designers will eventually look for new ways to make cards less... well, card-like. The common aesthetic pattern is a image or video,

then headline, followed by the main text in a stack so that none of the elements really touch each other.

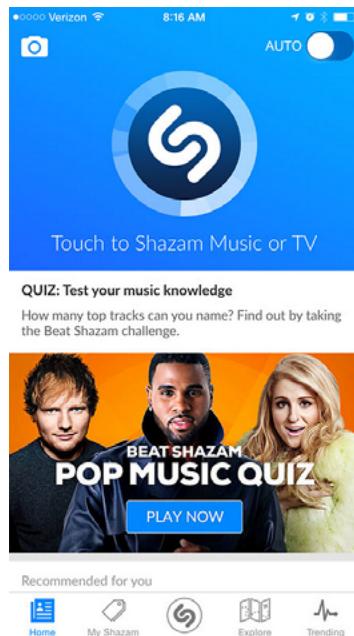


Photo credit: Shazam

Card design will likely take on characteristics of other design trends, such as hero headers or adding buttons on images.

While the outline and framing of cards will be maintained, the design will keep changing, especially for larger card styles. Some apps already do this. Take [Shazam](#) (see above), which stacks cards along with other elements, such as text and buttons, all within a single container. The design is sleek and highly usable.

While flat design never really went out of style, its evolution will continue to inspire cards. Cards in this style will include plenty of color, streamlined typography and subtle design tricks to help users navigate content.

The [7-Minute Workout](#) app features this type of flat style with rows of colored cards. Each expands to another card with instruction for the exercise that includes embedded content such as images, text and video.

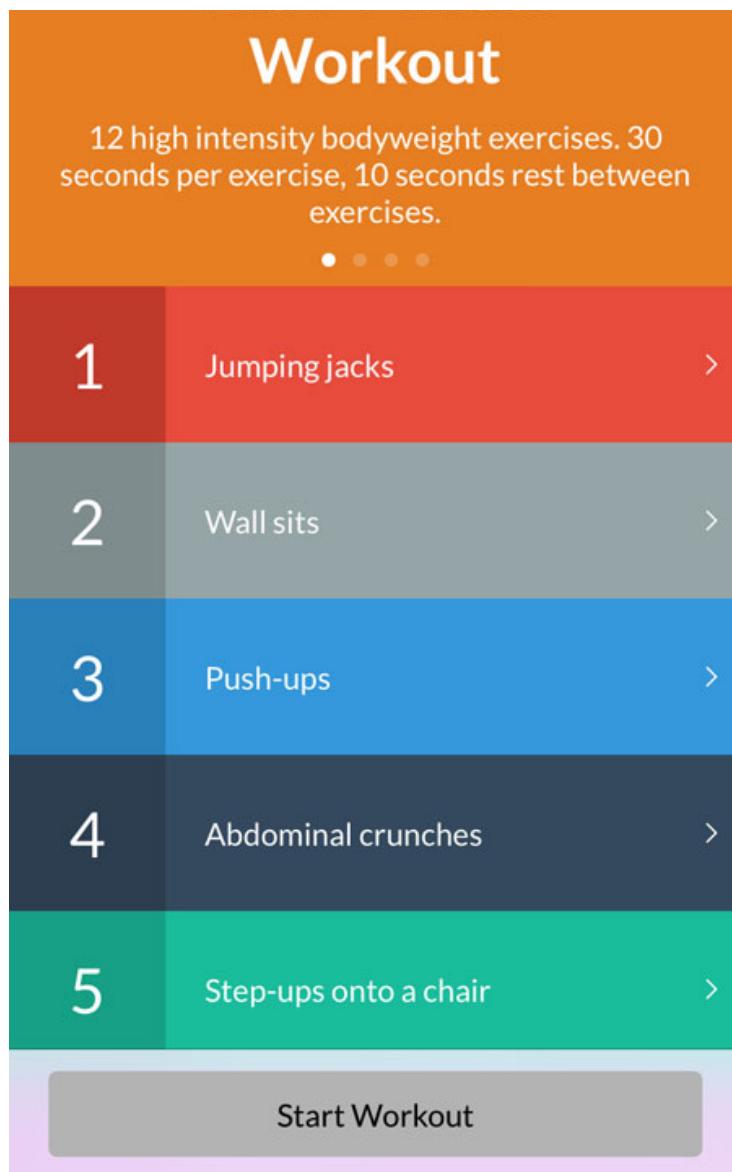


Photo credit: [The 7-Minute Workout](#)

More content-heavy sites will likely shift to card-style interfaces. Right now, this is the best method for organizing a lot of content in a manageable way. The packaging is almost newspaper-esque in that every piece of content comes in a box ready for users to digest.

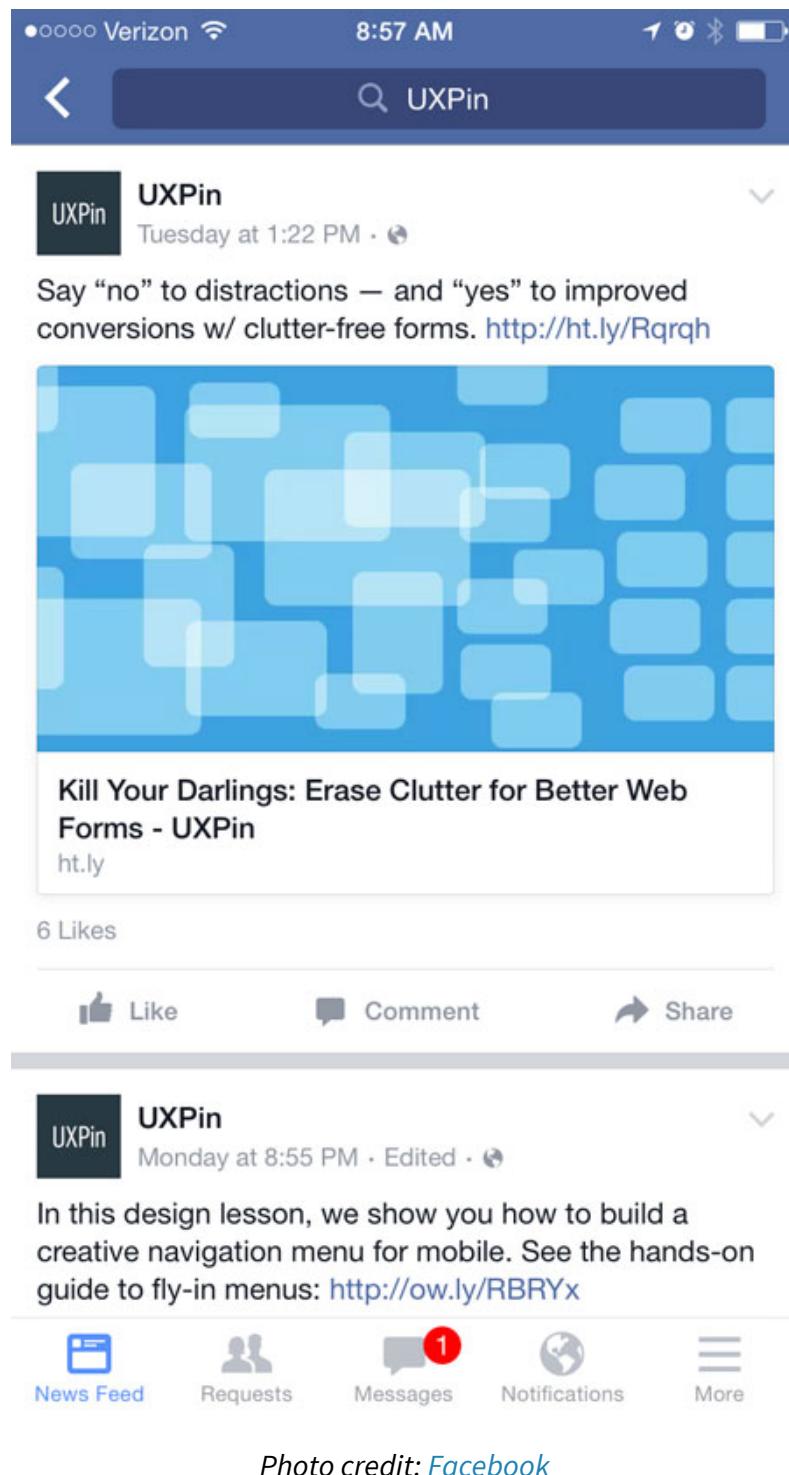


Photo credit: Facebook

Facebook continues to be a great example of content-driven card design, and will continue to lead the way with more news- and blogging-based features. As one of the dominant places where people find a lot of content, it is not surprising that their card style interface and patterns will only encourage others to follow.

In the end, cards are here to stay and will continue to be a fixture in mobile app design. The trick is to now build upon the common design language created with cards to find more interesting ways of incorporating them into our apps.

Create mobile prototypes faster in UXPin (free trial)

10 Resources and Tools

1. [CardStack Open Source Embeddable Card Runtime](#)
2. [“The Complete Guide to an Effective Card-Style Interface Design”](#)
by Design Shack
3. [Google Material Design: Cards](#)
4. [“Card Architecture and Card Design”](#) by Taylor Davidson
5. [Card User Interfaces Gallery](#) by Khoi Vinh on Pinterest
6. [“7 Rules for Creating Gorgeous UI”](#) by Erik D. Kennedy
7. [Case Study: How The Guardian Uses the Container Model and Cards](#)
8. [Mobile Interface Design Patterns](#)
9. [“How Cards Are Taking Over Web Design”](#) by Jerry Cao for The Next Web
10. [Boring Cards Free UI Kit](#)

Everything you ever wanted in a **UX Design Platform**

- ✓ Complete prototyping framework for web and mobile
- ✓ Collaboration and feedback for any team size
- ✓ Lo-fi to hi-fi design in a single tool
- ✓ Integration with Photoshop and Sketch

Start using it now!