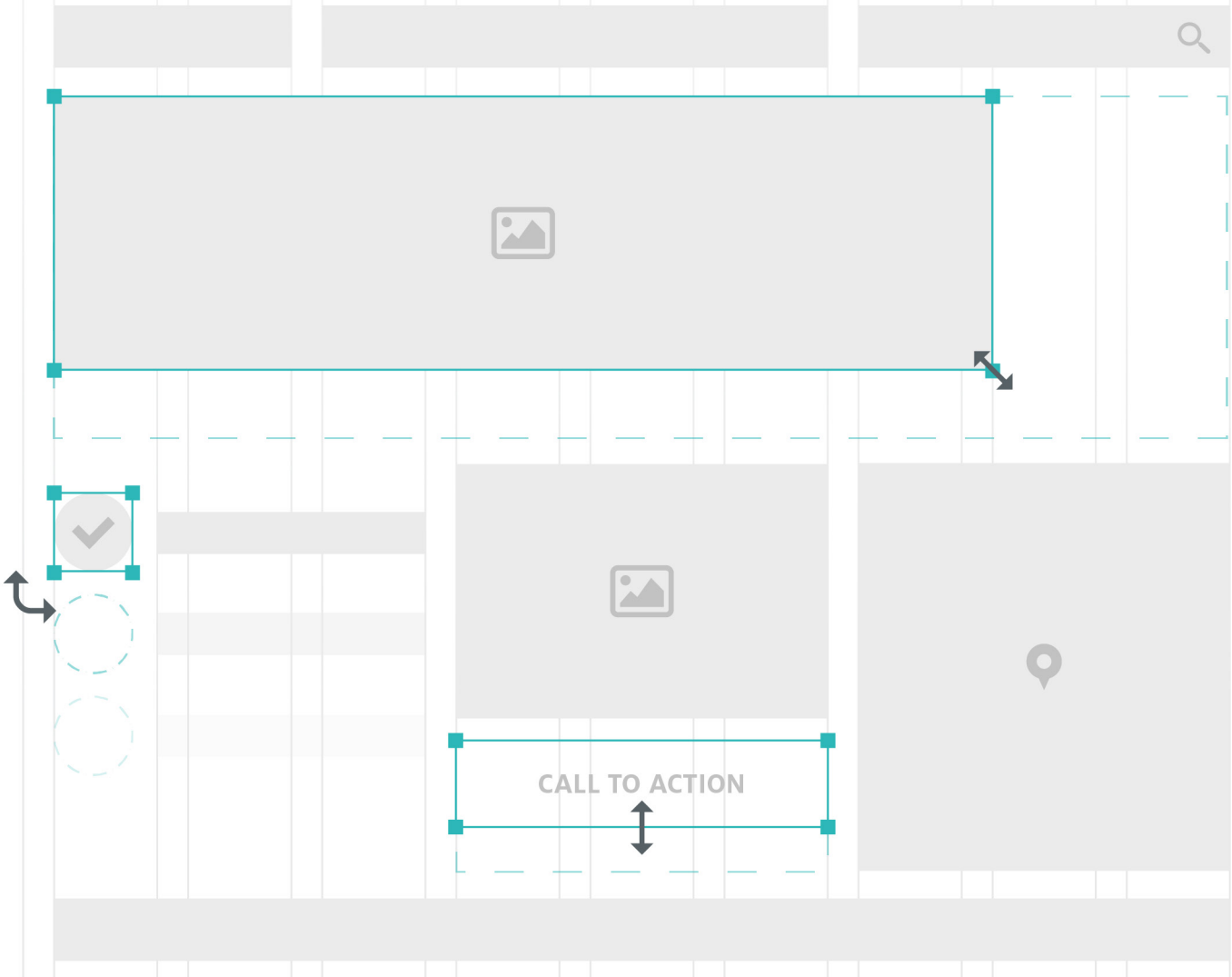
UXPin

THE GUIDE TO WIREFRAMING

For Designers, PMs, Engineers and Anyone Who Touches Product



INDEX

1

INTRODUCTION

In this book, I share a wide breadth of examples, comparisons, frameworks, references and insights to help you better utilize wireframes in your product design and development process.

I talk about basic concepts like who uses wireframes, what's their purpose and how you can use them. I also detail the many types of wireframes you can create, and the types of tools used to create them. For more experienced readers, I've also laid out the top web and mobile wireframe example and pattern sources, and highlight some of the best web and mobile UI design patterns that have cropped up in the past few years. Finally, I cover design principles that are hopefully helpful as you wireframe and, ultimately, build products.

Hopefully, this is the start of a discussion. These are some of the many questions that my team at UXPin and I are trying to answer every day. And we need to answer them so we can build one of the best wireframing and prototyping tools that you or people like you will use - and love.

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

For the love of products,
Chris Bank

2

A PRACTICAL LOOK AT USING WIREFRAMES

Who uses wireframes, what's their purpose & how do they work together

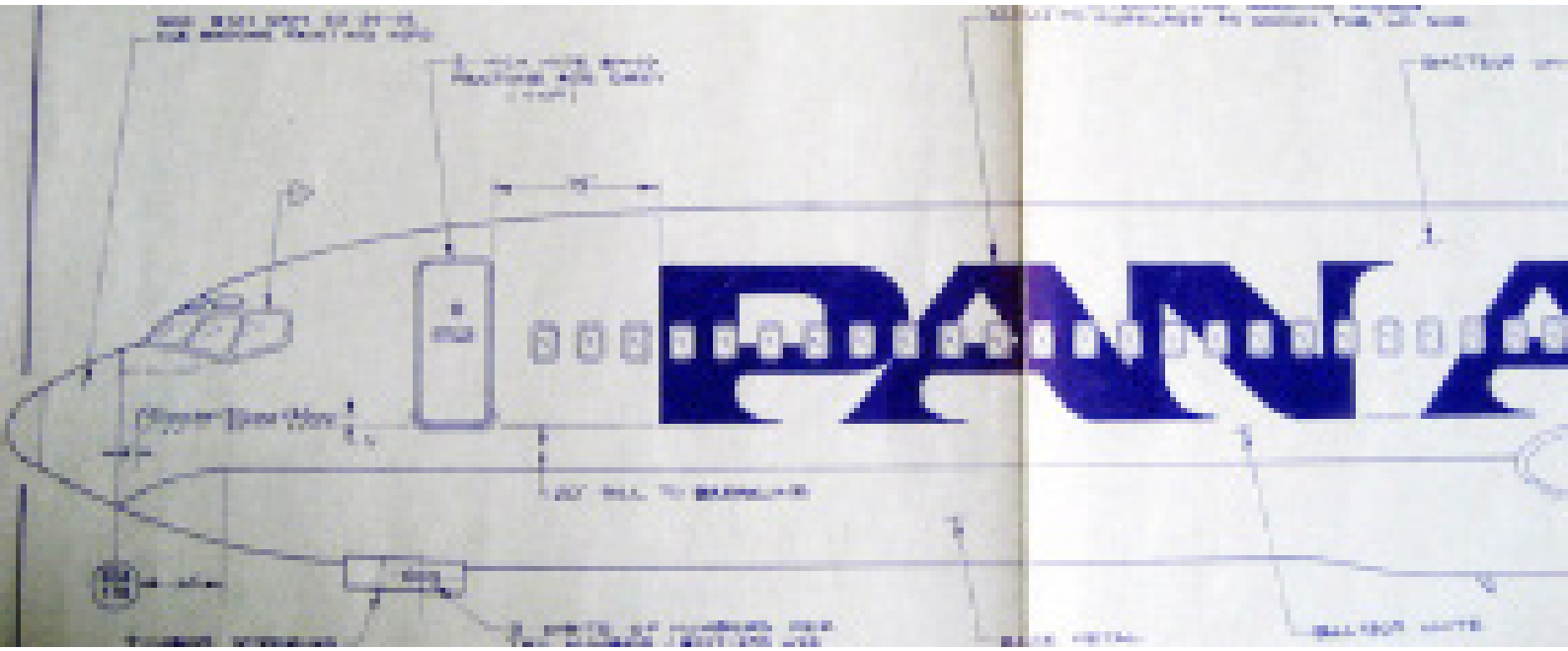


image: Todd Lappin

Wireframing is an important tool for product design and development.

Whether you're building the next hot startup or a solid website or mobile application, wireframes are invaluable in keeping everyone on the same page - not just product managers, designers, and engineers. And they can be changed really quickly to accommodate the collaborative and iterative nature of product design and development, especially in agile startups and enterprises.

For this reason, wireframes are typically created in the product design and development process in one way or another, even if it's a quick sketch on scratch paper or a grid notepad.

Who Uses Wireframes?

In short, anyone involved in the product - in any capacity.

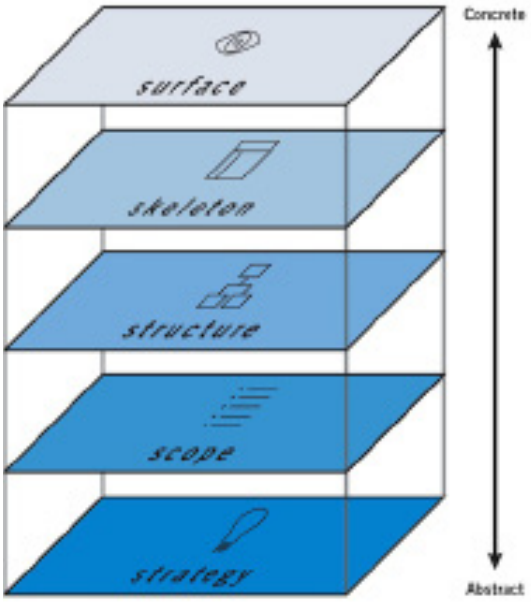
Although designers, developers, and product managers typically create and use wireframes the most in their daily work, many people benefit from wireframes. These may include business analysts, information architects, interaction designers, user experience designers, graphic designers, programmers, and product managers.

In a later series, I'll talk more about how these team members should think about working together, including their use of wireframes and other design tools.

Why Should Anyone Use Wireframes?

Wireframes are the "blueprint for design."

They're supposed to connect the underlying conceptual structure (or information architecture) to the surface (or visual design) of a website or mobile app. More specifically, they're visual representations of an interface, used to communicate the following details to get everyone on the same page:



- **Structure** - How will the pieces of this site be put together?
- **Content** - What will be displayed on the site?
- **Informational hierarchy** - How is this information organized and displayed?
- **Functionality** - How will this interface work?
- **Behavior** - How does it interact with the user? And how does it behave?

image: Jesse James Garret, The Elements of User Experience

Wireframes are not supposed to represent the visual design, contact graphic elements, or convey the brand or identity.

Why Should I Use Wireframes?

The purpose of any particular wireframe varies slightly by the people creating or using them in any company. Whether sketched, “grey boxed”, wireframed in a graphic editor or in a dedicated wireframing and prototyping tool like UXPin, the same information can be conveyed. Therefore, it’s less about the tool and more about how fast you can convey the information you need to others.

Interaction & UX Designers and Information Architects use wireframes to show user flows between views or pages. Typically, a combination of flowcharting, storyboarding and wireframing are used to achieve this.

In the following examples, you mostly see detail about product structure, functionality, and behavior, with limited details about content or information hierarchy - we’ll leave that for the designers.

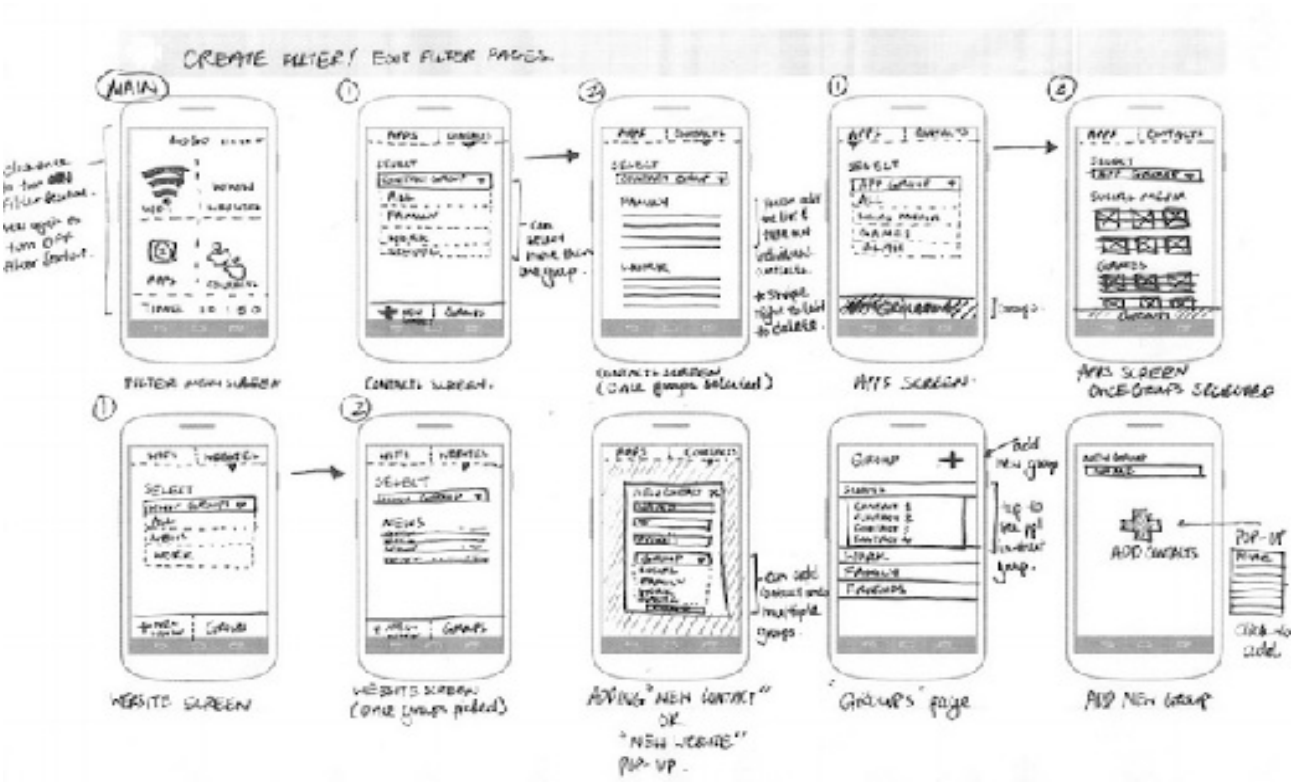


image: Jiani Lu

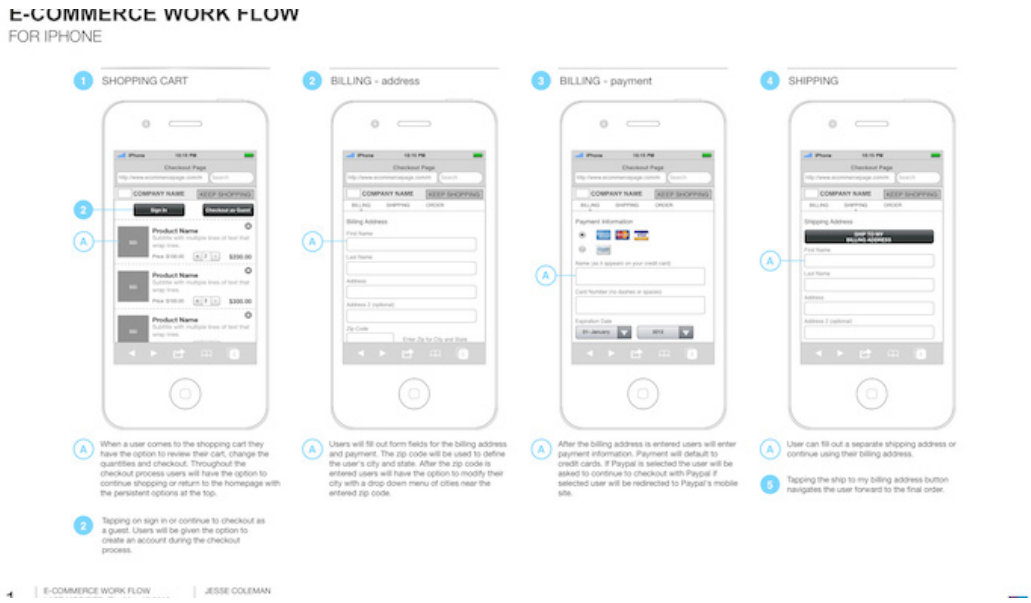


image: *Jesse Bob*

image: *Andrew Brennan*

Graphic Designers use wireframes to push the user interface (UI) development process. It can inspire the designer, resulting in a more fluid creative process. And is ultimately used to create graphic mockups, interactive prototypes, and the final design. Typically, a combination of sketching, storyboarding and wireframing in low or high-fidelity are used to achieve this.

In the following examples, you mostly see detail about product content and information hierarchy, and limited detail about structure, functionality, and behavior - that was done by the UX designers and information architects.

Developers use wireframes to get a more tangible grasp of the site's functionality. It gives the developer a clear picture of the elements that they will need to code.

For back-end development, wireframes can be low-fidelity the way a ux designer or information architect might produce them - they care more about product structure, functionality and behavior. For front-end development, it's more helpful to have high-fidelity wireframes a designer would produce - they care about content and information hierarchy as much as the structure, functionality and behavior.

image: *Mike Rhode*

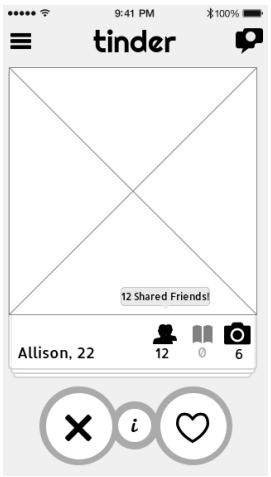


image: *UX Porn*

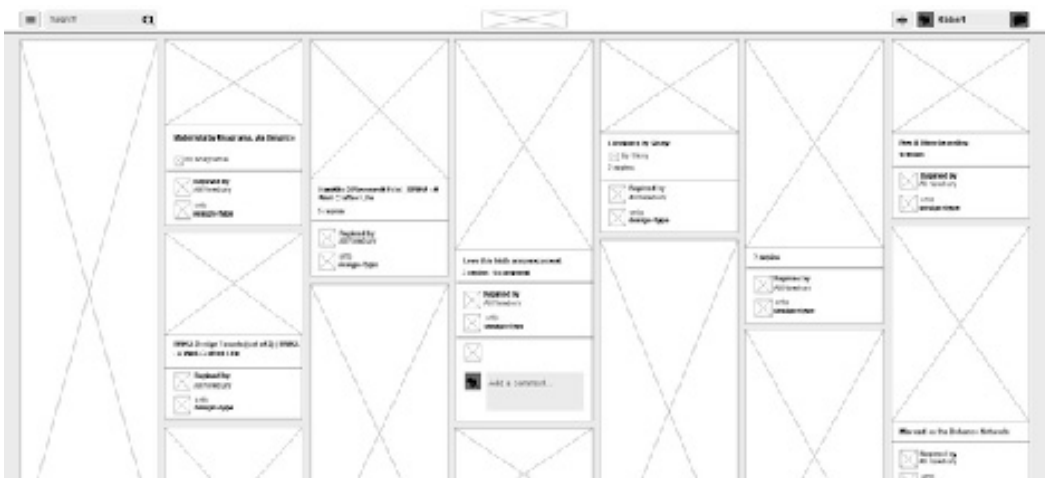


image: *UX Porn*

Business Analysts use wireframes to visually support the business rules and interaction requirements for a screen.

Depending on the business, industry or the analyst's specific role, they will care more about only one or two of the following - structure, content, information hierarchy, functionality and behavior. For example, an advertising or content analyst probably cares more about content, functionality and behavior whereas a regulatory analyst may care more about information hierarchy or structure. However, it varies too widely to say.

Internal Business Stakeholders (ex: Product Managers, Project Managers and Executives) review wireframes to ensure that requirements and objectives are met through the design. This fits into their overall product strategy and scope of a specific project or set of projects. It gives managers an early, close-up view of the site design (or re-design).

Barring some rough sketches, they care about every type of wireframe the team is looking at because they're all part of the "blueprint of design." Any wireframe that conveys the requirements and objectives of the product is the Holy Grail to a product and project management - other executives typically care about various parts of the design process, especially the end result.

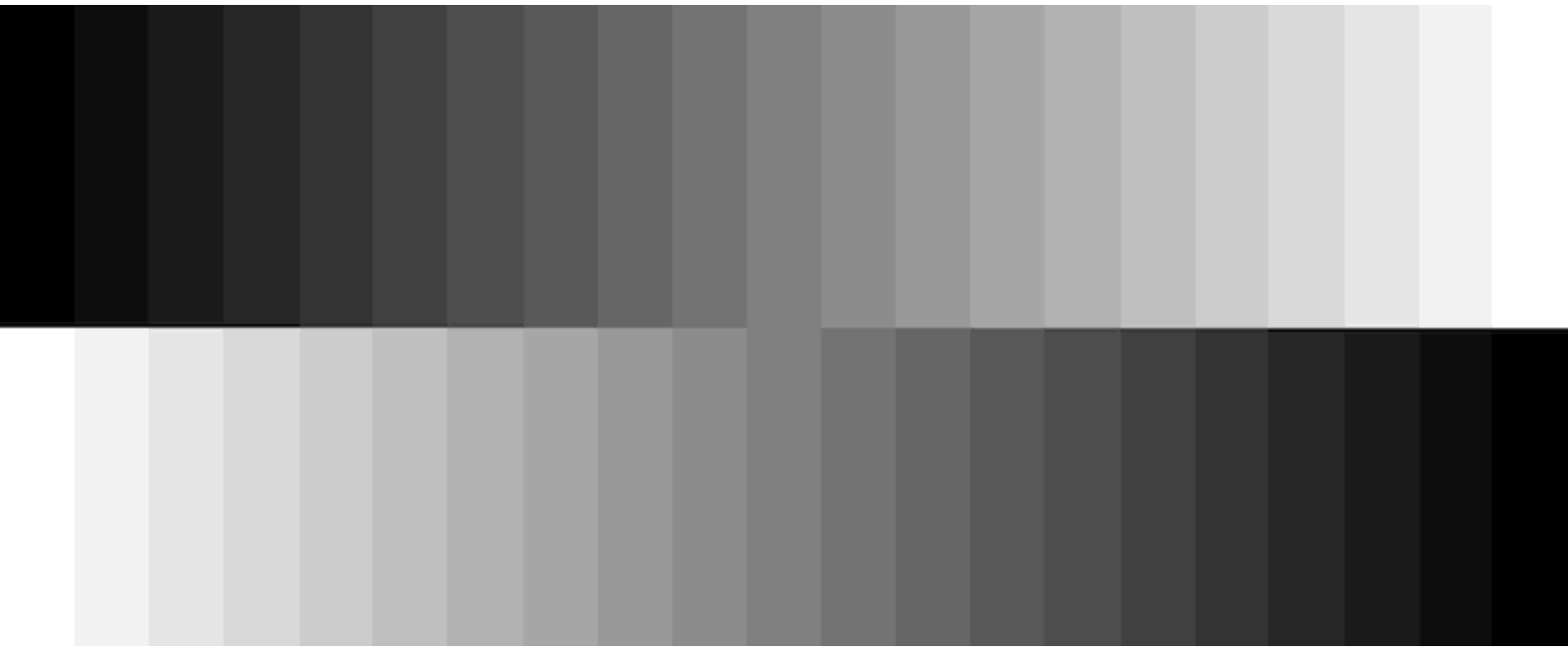
External Business Stakeholders (ex: Partners and Clients) also review wireframes to ensure that requirements and objectives are met through the design. It gives managers an early, close-up view of the site design (or re-design).

However, they typically care more about annotated, polished (if not high-fidelity) wireframes than many of the types of wireframes internal business stakeholders favor for speed in collaboration and execution. Remember, they're busy running their own businesses. And because they typically don't work as closely with you, they want something clean and readable to quickly get up to speed on what they've missed while you're toiling away on design.

3

50 SHADES OF WIREFRAMES

The many styles that make wireframing a sexy part of product design



Now that you have a better understanding of who uses wireframes, what's their purpose and how teams work together to use them, it's time to cover the many shades of wireframing. Specifically, I cover the ways you can make wireframes, what levels of detail you can provide, and how to tie all product details together in the finishing stages.

But first, let's revisit the purpose of wireframing so the irresistible pictures below make sense. Wireframes are visual representations of an interface, used to communicate the following details to get everyone on the same page:

- **Structure** - How will the pieces of this site be put together?
- **Content** - What will be displayed on the site?

- **Informational hierarchy** - How is this information organized and displayed?
- **Functionality** - How will this interface work?
- **Behavior** - How does it interact with the user? And how does it behave?

Wireframing & Prototyping Convergence

There's a lot of chatter about high-fidelity, functional prototypes, and MVPs these days - and for good reason: you know exactly what you're getting. While that sounds sexy in theory, product teams can't just go from 0 to 100mph in most cases - instead of making a Minimum Viable Product (MVP), you'll make a Sucky User Experience (SUX).

Even elite designers, product managers, and engineers who are exceptional at conceptualizing high- and low-level details about a product on-the-fly need to wireframe in some way, shape, or form. And most product teams - small and lean, or large and bulky - still rely heavily on wireframing, using digital tools like UXPin or analogue ones like pencil and paper.

What I'm seeing, however, is a convergence of tools for wireframing and prototyping, a blurring of traditionally distinct stages in the product development cycle to create a broadened spectrum of wireframe styles that allow teams to communicate at the level of detail and speed required.

Choosing Your Wireframe Style

Okay, so there's probably only a few kinds of wireframes you'll ever use. But if you count variations of medium and level of detail, you could probably get close to 50 - and it matters!

I covered this topic to some extent in The Aesthetics of Wireframes and the Importance of Context, but wanted to go deeper. Whether you're aware of it or not, every shade, line, text, graphic, interaction that you add to a wireframes impacts

how people interpret and understand the product concepts. Like English (or French if you're looking for a more romantic language), wireframing is a language - for product design and development.

And as people across the World still debate semantics of their respective languages, product teams do the same - especially with wireframing. Over the years, I've heard the terms "sketches", "mockups", and even "prototypes" used synonymously with "wireframes." To overcome much of this confusion, traditional wireframing tools have remained incredibly austere and carved out their own niche.

But the UXPin team sees this line blurring as product iteration cycles shorten, team composition and process varies more widely, and team members have more influence over communication and style preferences. The bottom line is that every team develops its own internal language, a slight variation from the norm, and wireframing is flexible enough to accommodate many styles to get the job done.

Tools & Medium

Many pundits may disagree whether or not some of these tools and medium are relevant to wireframing, but if it looks like a wireframe and smells like a wireframe, I'm calling it a wireframe. No matter what anyone tells you, you can achieve the majority of your objectives in wireframing through any of these. I'm not here to hold you back with semantics - just get the wireframes done!

Sketching. You can sketch with anything that makes pigment on a surface. Because there are so many options, we gave you double the examples below so you could see the true breadth of tools and medium you could use. So get out your pens, pencils, markers, crayons, Paper App - even chalk or whiteboard markers - and start sketching!

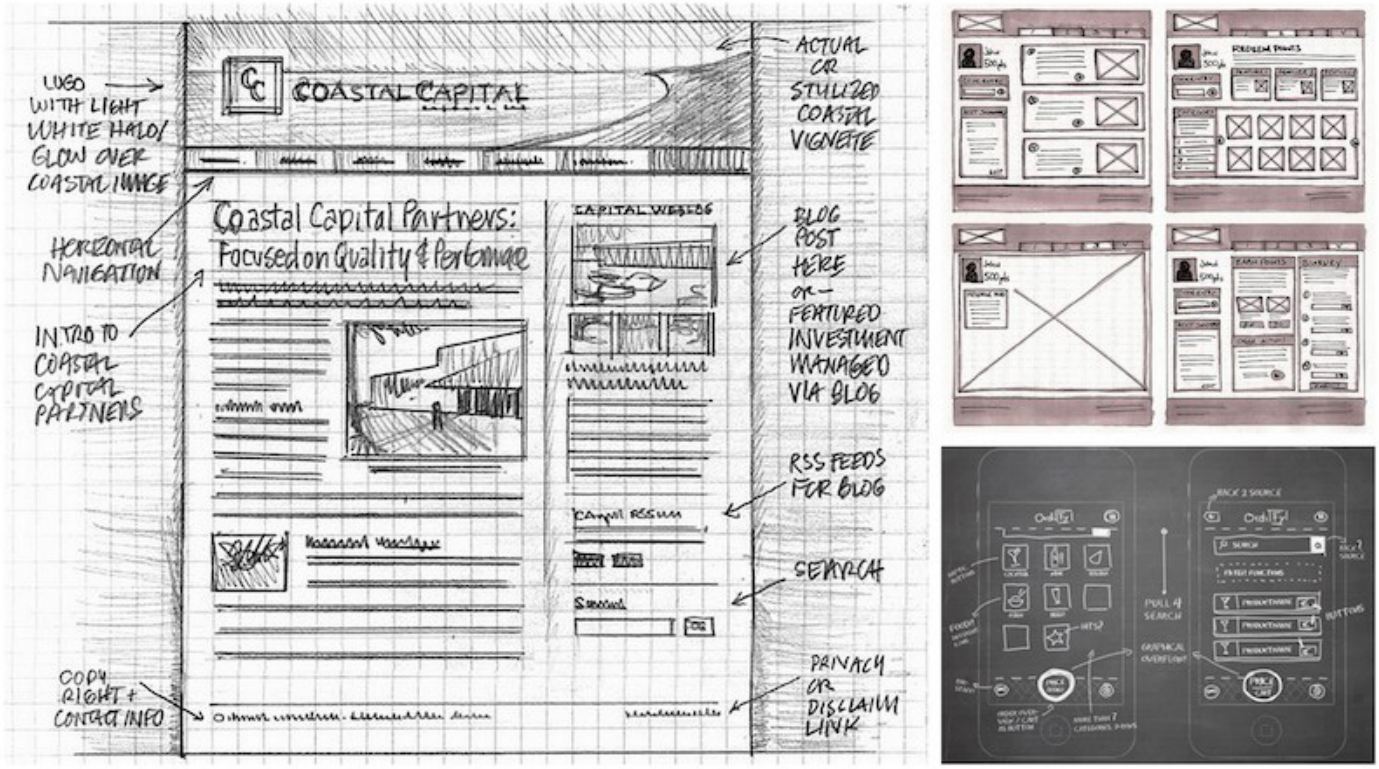


image: *Mike Rhode, Chris Stevens, Christos Chiotis*

Paper Cutouts. You can use standard paper wireframing kits (which was the precursor to the UXPin wireframing and prototyping software), or simply cut out the sketches you've made on paper or another medium - you can always cut down edges later for "re-sizing". Although this isn't a wireframe, Common Craft's famous Dropbox explainer video illustrates how fun this type of wireframing can be.

image: *Caryn Vainio, Winnie Chang, Adrian Kosmaczewski*

Stenciling. There are many templates you can print out, stenciling kits to make your own, and full paper wireframing kits that come with standard paper elements like the UXPin paper product I mentioned above.

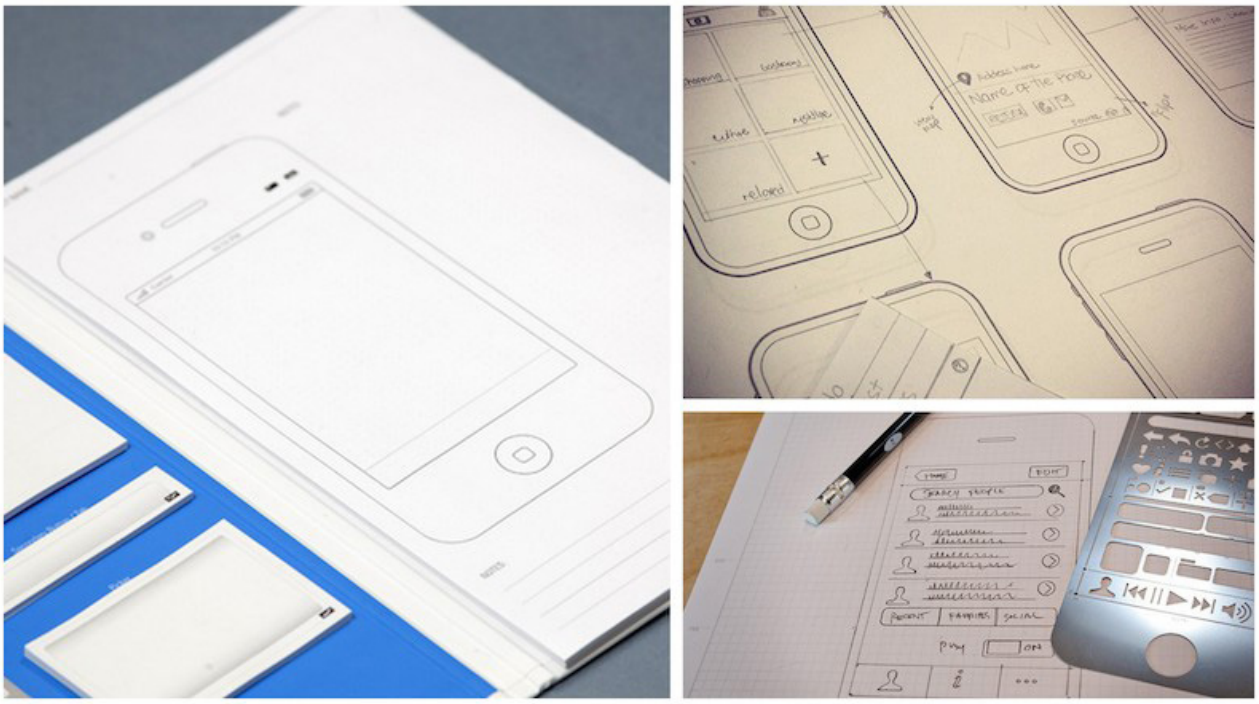


image: UXPin, Natalia Sourdis, iphonized

Wireframing Software. There are many digital wireframing software solutions that you can use, but only a few serious apps that are truly worth checking out, in our opinion - the rest tend to be limited, or have stopped innovating. To name a few products, UXPin, Balsamiq, Axure, and Proto.io are great.

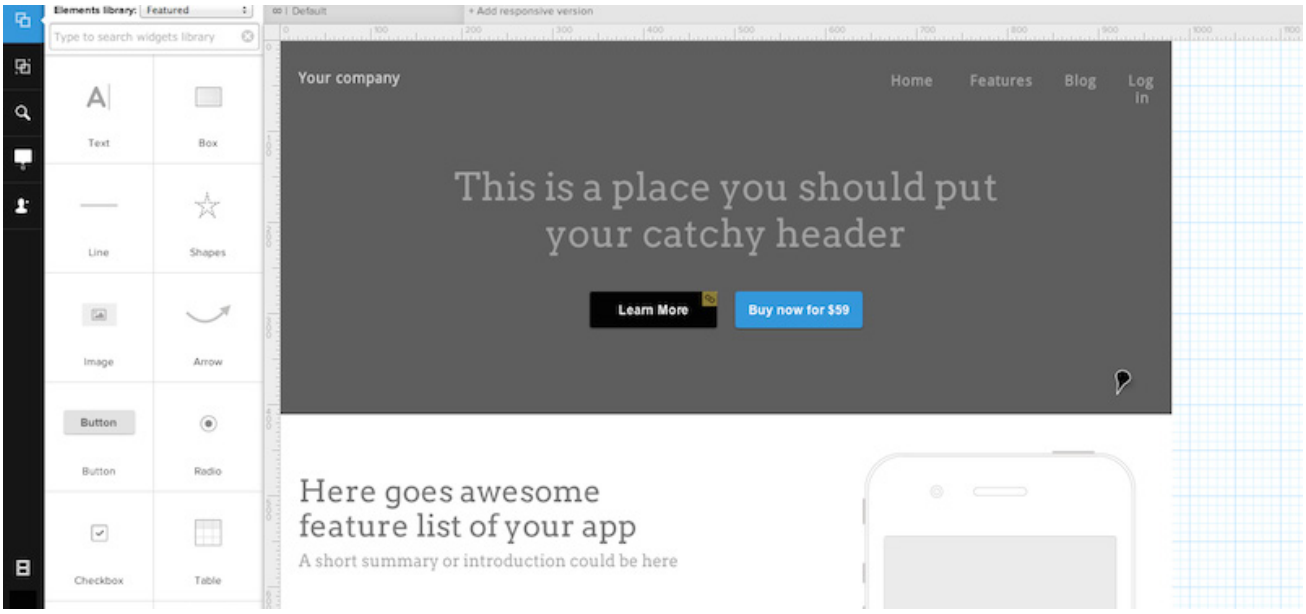


image: UXPin

Graphic Design Software. There are many digital graphics software (free and paid) that could be used but the vast majority of designers and product managers use Adobe Illustrator, Photoshop, or Sketch for graphic design already - so those are clear winners if you choose to use them for wireframing too.

For many product teams, however, the better option is wireframing software because they're focused solutions for wireframing, and a central medium for collaborating on wireframes and integrating with other tools that are important in the entire product design lifecycle. I will cover this in greater detail in the next part of this series.

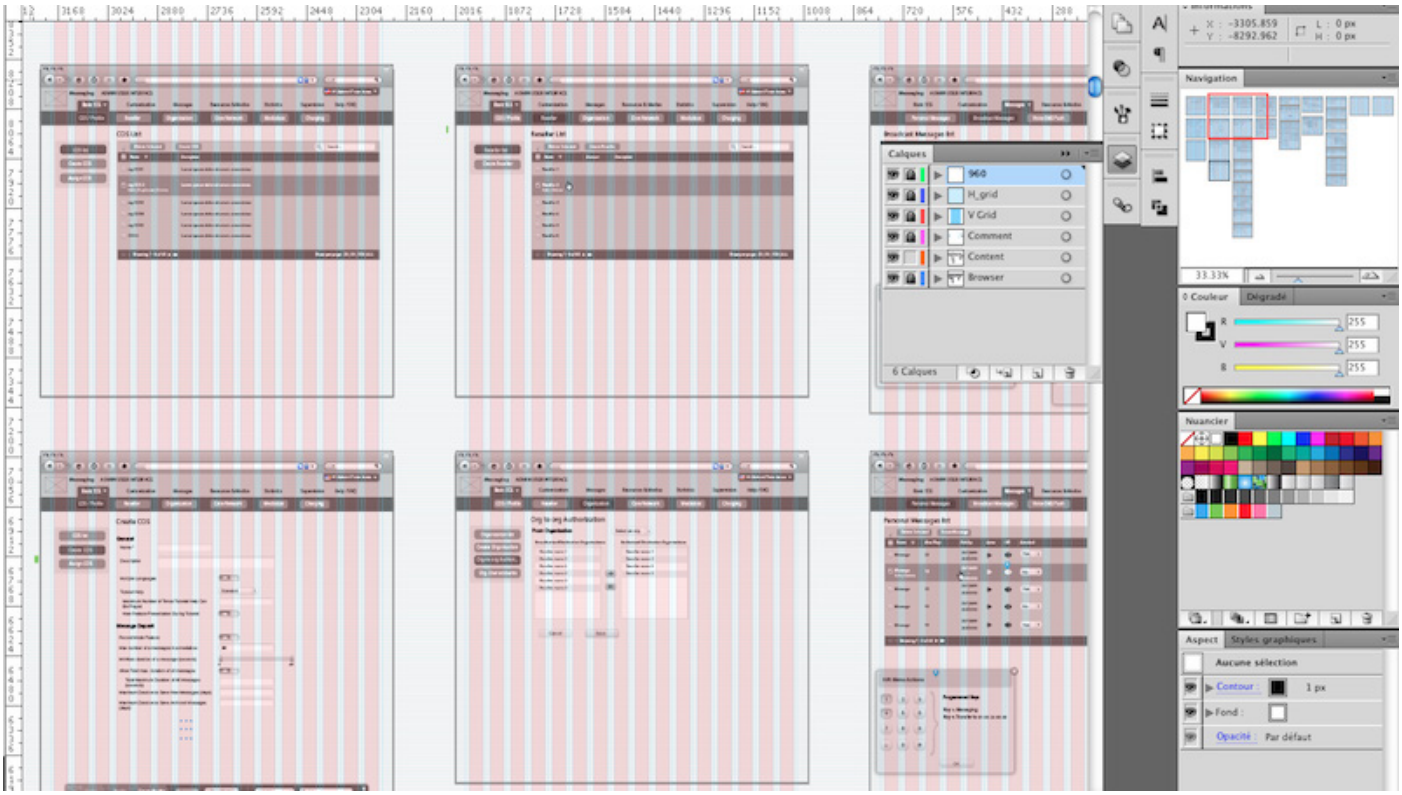
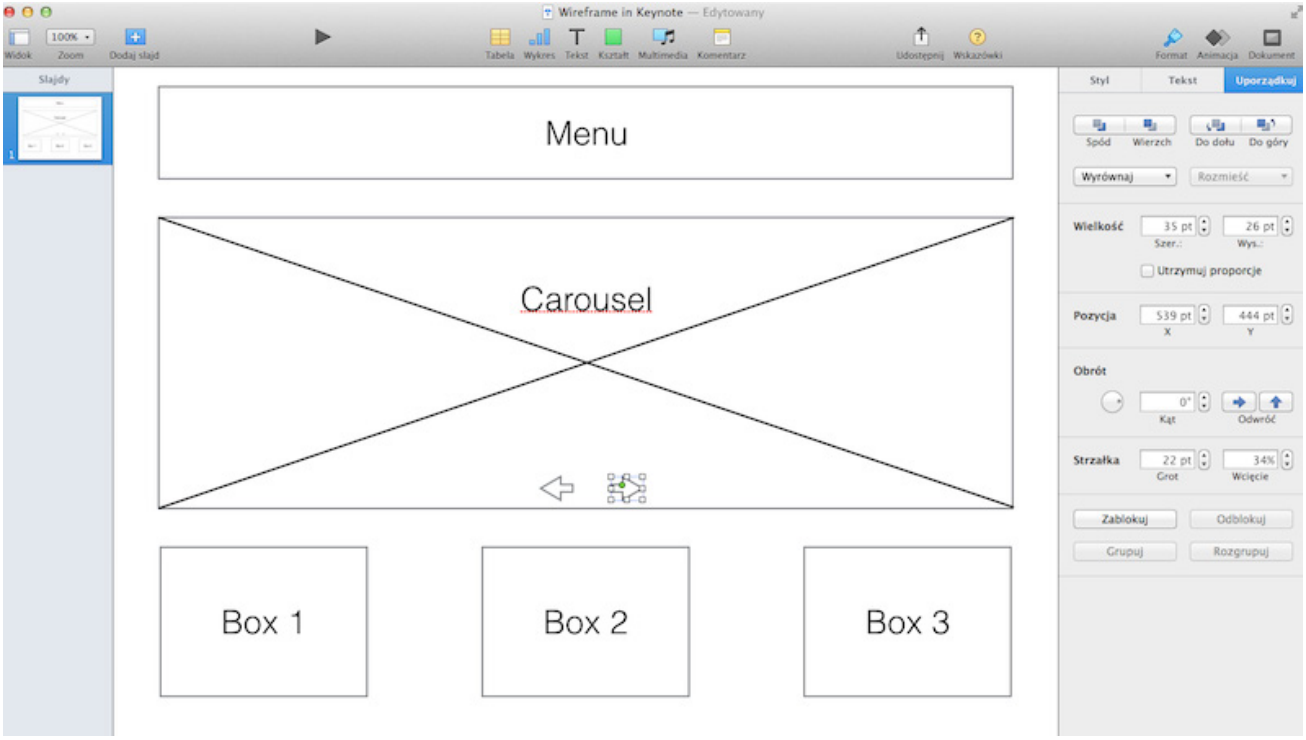


image: *Marc Decerle*

Presentation Software. Most people have used presentation software in their life - it seems almost mandatory in basic US education - and it's even more common among business people. The broad familiarity of this tool makes it another great basic option for creating and presenting wireframes to teams. And you basically have two choices - PowerPoint or Keynote - and templates like Keynotopia exist to get you started.



Fidelity Levels

Barring limitations on the medium and tools you use, you should there are varying levels of detail you can dive into based on the stage the product design process and overall objective of the particular wireframe.

Block Diagrams provide the most basic but essential information about a wireframe: the layout and types of content or functionality you would like to present to your visitors. You can also set the information hierarchy and typography in this stage by adding text and varying the sizes to show emphasis or hierarchy. Just make sure to set your grid ahead of time to stay structured and help you visually tie together the elements in the beginning stage.

You can get creative in this stage since you're at the beginning stages of setting constraints but keep in mind existing design patterns, overall objectives of the product, and any other constraints or user flows at this stage - because it's easy to get off track if you're working with an existing product or haven't built an intuition around good and bad page layouts. And make sure that you that you don't lose sight of the overall layout when adding text - it can bite you later on.

Unsurprisingly, you can create this level of wireframe with every tool and medium I mentioned above.

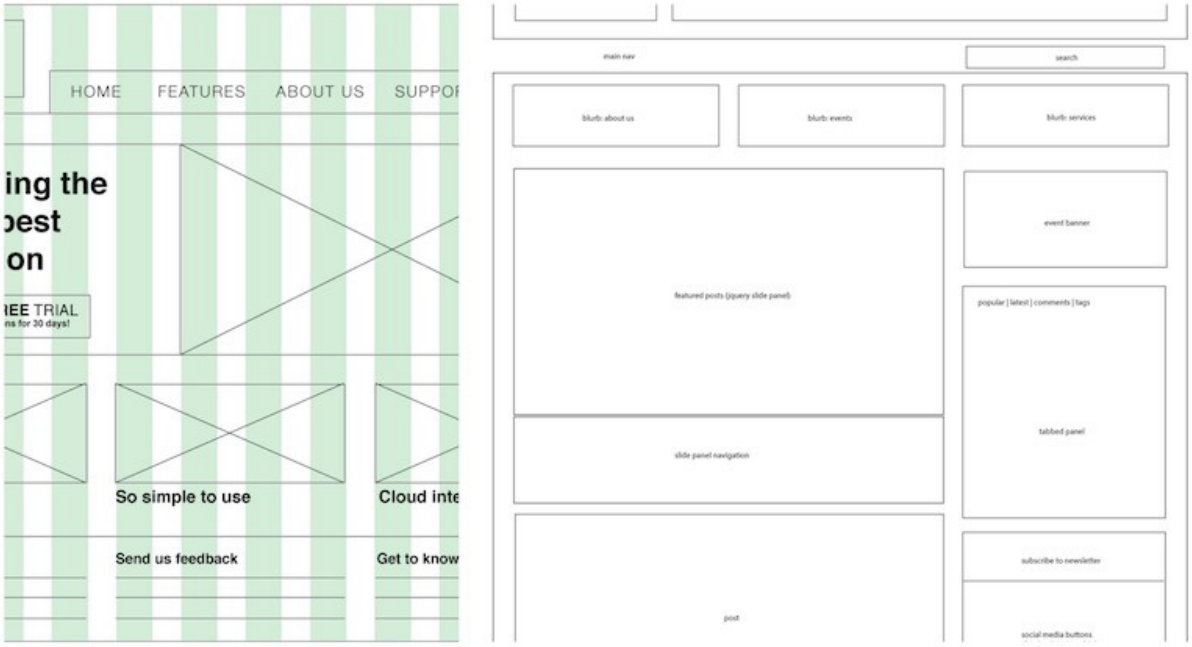


image: Winnie Lim

"Grey Boxes" allow you use the full grayscale spectrum to emphasize the layout and particular elements without taking the time to choose a color palette or cluttering your wireframe with "lorem ipsum" text, both which may end up distracting you in the initial design steps. This may help you later on with the graphical design as well. It's especially great for testing user flows and organizing graphical content, but can also be a personal preference over block diagrams or blocks with text.

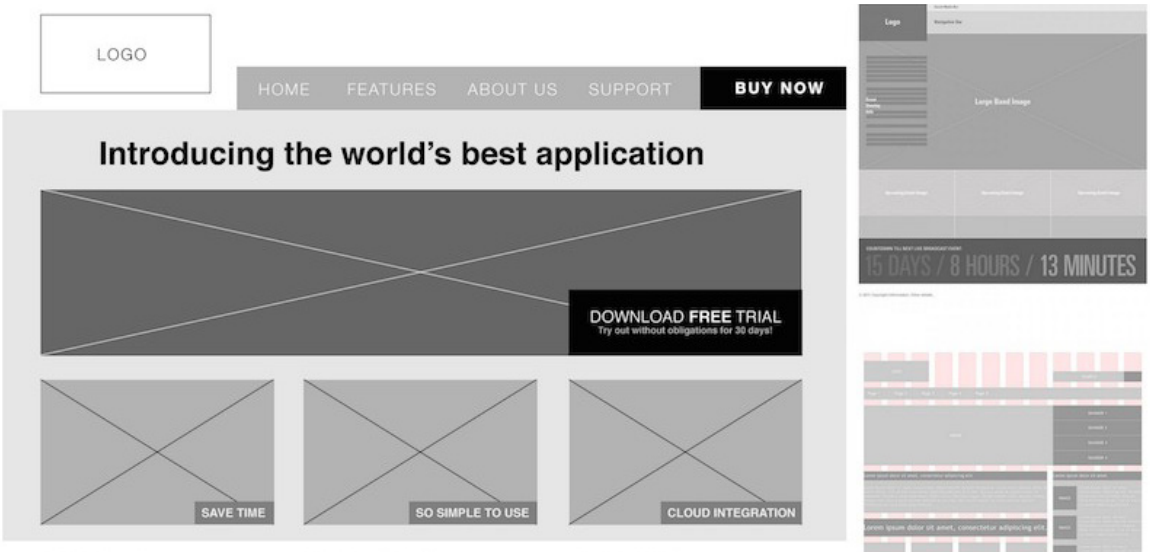


image: Winie Lim, Paul Armstrong, Eu Diaz

High-Fidelity Text is one of several ways you can make a wireframe more realistic without going too granular into the graphical details. This may mean filling in the actual copy, the copy length, ideal font, sizing, weights, and so on. As with the other variations of high-fidelity wireframes, this level of detail about the product should be understood with little or no explanation - so you can move onto the next wireframe to keep the product design and development process going.

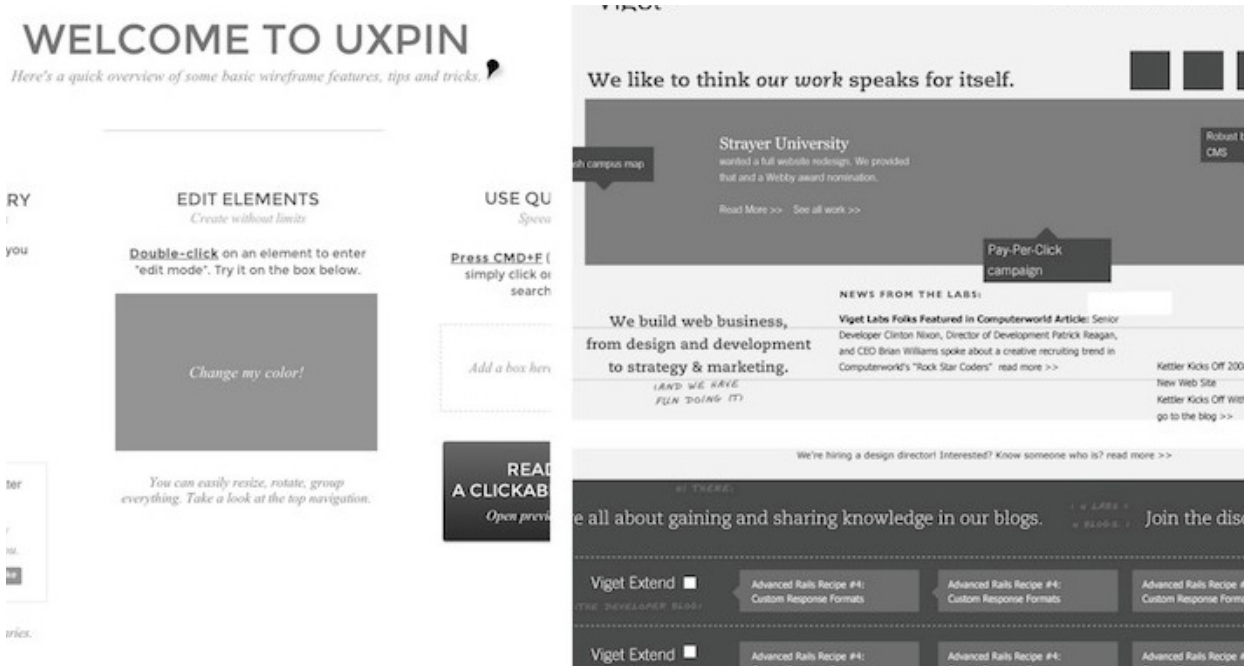


image: UXPin, Viget Labs

High-Fidelity Color is another way you can make a wireframe more realistic without going too granular into the graphical details. This may mean filling in very specific background, button, text, or other colors to emphasize desired actions or user flows beyond what grayscale can offer. Be careful not to go crazy with the color or you risk both reducing the effectiveness of color "highlighting", wasting a lot of time on making a pseudo-mockup that your graphic designer will ultimately replace, or making it harder to communicate other details of the wireframe to stakeholders because they're distracted by the visuals - they may even perceive the wireframe as a bad mockup instead of a slightly more detailed wireframe.

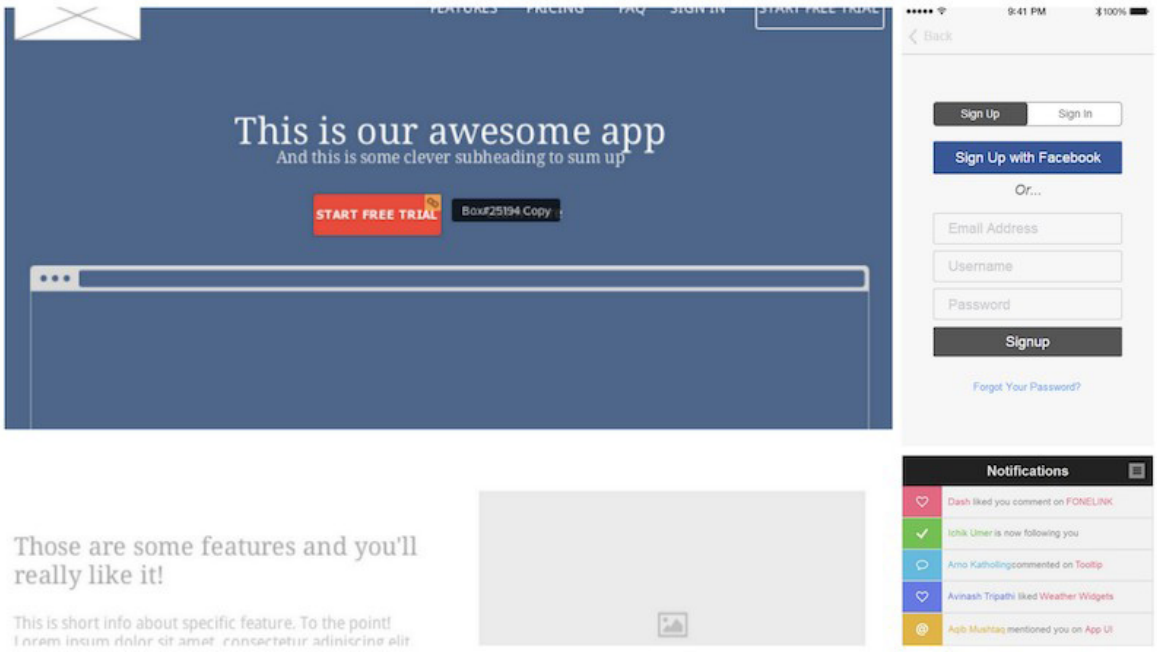


image: UX Porn

High-Fidelity Media is another way you can make a wireframe more realistic without going too granular into the graphical details. This may mean filling in a picture, video, thumbnail, avatar, background image, or the like. This is becoming increasingly popular with the proliferation of content-centric products - one of the trends highlighted in UXPin's Design Book of Trends 2013-2014 - since the type of content can have a significant impact on the overall structure and information hierarchy.

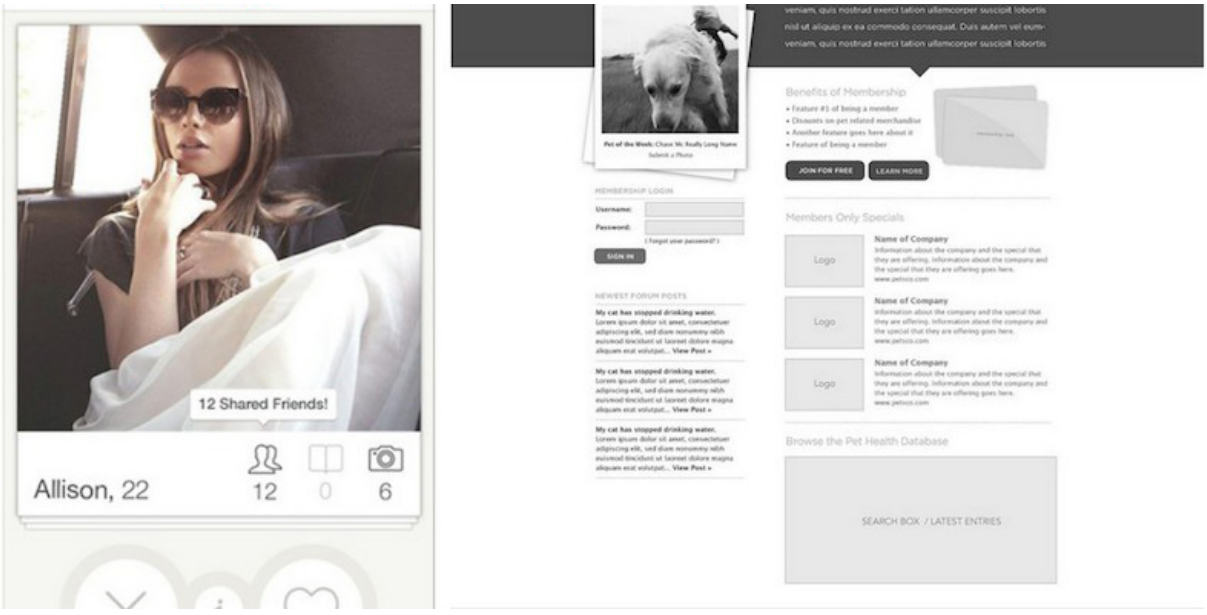


image: UX Porn, Jesse Bennett-Chamberlain

High-Fidelity Interactions is another way you can make a wireframe more realistic without going too granular into the graphical details. While I will cover adding interactions and click-through links to your wireframe later, this type of wireframe is more about visually showing interactions on a static wireframe. For example, pop--up windows, expanding or zooming in on content, and broadly showing swiping, scrolling, tapping and clicking interactions (see live example).

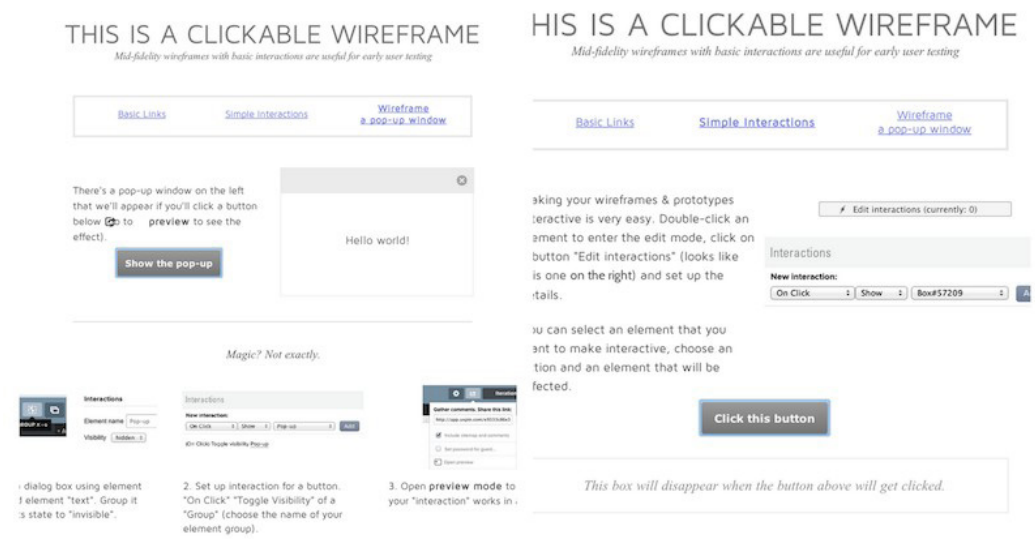


image: UXPin

Additional Details

Now that you've chosen the proper medium and tools, and have drawn out a wireframe or wireframes at the right level of detail, you need to add the finishing touches. Some wireframe details may need explanation, multiple wireframes may need to be tied together to show the user flows, while others may need to show interactions.

Basic Wireframes alone are only as good as your ability to produce the right level of detail needed to build or improve the product. Typically this isn't sufficient alone, but can be in several scenarios. For example:

- You're the one wireframing and prototyping or programming the product - you don't need to cover every detail!

- Your graphic designer is more experienced and is given the freedom to interpret the wireframe with fewer constraints.
- You're making very specific changes to the site that are well-understood and/or documented elsewhere, perhaps in a product requirements doc or wiki like Google Docs, or product management software like JIRA.

You've seen enough basic wireframes above so I'll hold off on sharing more.

Annotated Wireframes are great for adding a broad range of details to any basic wireframe without cluttering it or risking scope creep - or missing the point completely. For example, although wireframes are the quickest and easiest way to specifying interaction, annotations are often necessary to communicates what happens when certain actions are triggered. Some examples of detail you can include:

- **Functionality:** clicks, taps, swipes, zoom, pop-ups, data inputs and outputs, etc.
- **Content:** text, fonts, layout, sizing, linking, graphics, multimedia, dimensions, resolution, etc.
- **Behavior:** animation styles, speed, and positioning, interactions, link destinations, etc.
- **Key constraints:** hardware, software, browser, data, etc.

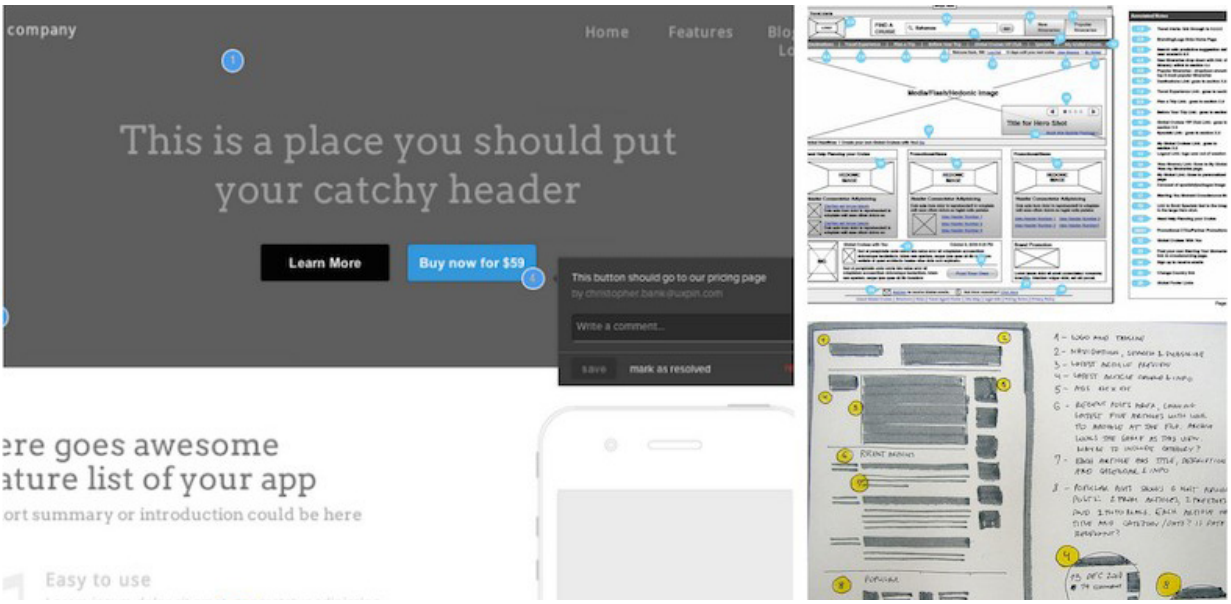


image: UXPin, Will Evans, Janko Jovanovich

Wireframes With User Flows are increasingly common as flowcharting and wireframing converge at the earlier stages of product design. Sometimes words alone cannot communicate a behavior, especially with user flows. They can look like traditional storyboards or more precise interfaces as wireframing tools evolve. In some instances, these wireframes are simply static views of a fully interactive wireframe. In other cases, it could be a slideshow or a collection of wireframes set side-by-side to show a primary user flow or set of user flows without detailing every link and flow.

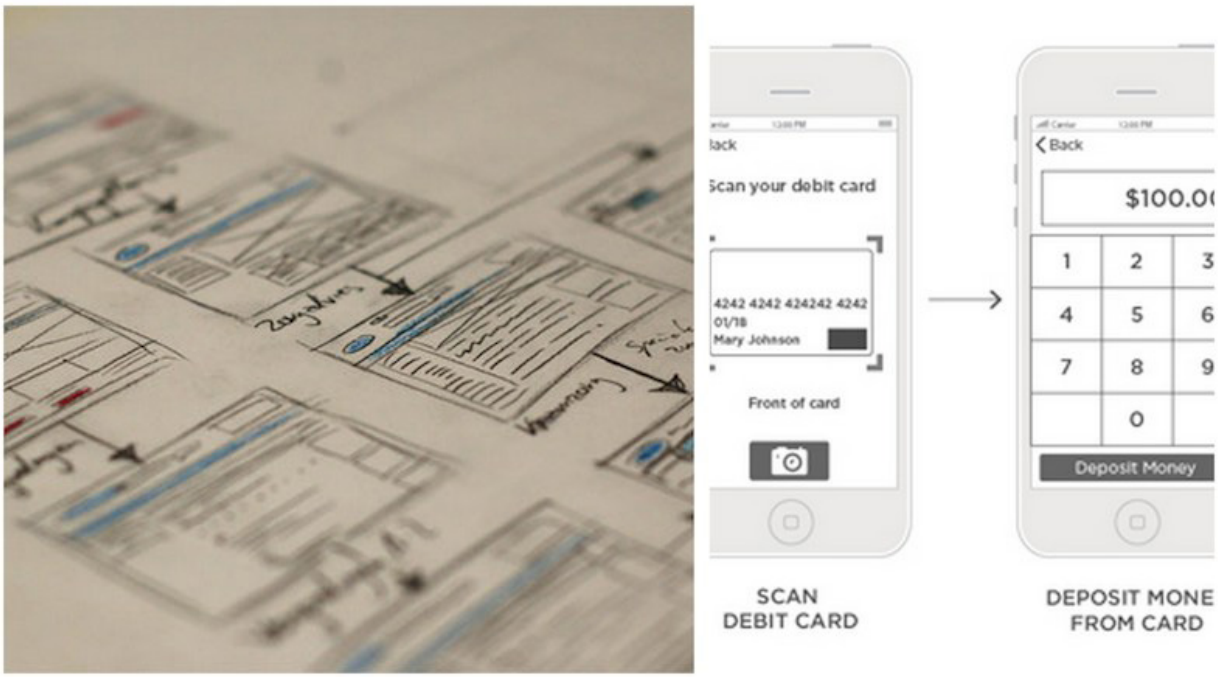


image: Henk Wijnholds, Victoria Ngo

Interactive & "Click-through" Wireframes are as close to a prototype as you can get before final product - it's like animating a storyboard. They allow you to actually experience interactions (ex: taps, clicks, and swipes) within or between individual wireframes instead of imagining the user experience through static wireframes and flows. Adding interactions before going to full mockups or even live prototyping can save significant designer and developer hours. This is a powerful variation of wireframes that can only be done in presentation and graphic design software, or wireframing and prototyping software like UXPin, Balsamiq, Axure, and Proto.io.

Unfortunately, I can't visually represent this but you can always play around with some interactive wireframes in UXPin.

Practice Makes Perfect

Now that you've done your homework on how to think about creating wireframes, it's time to put pen to paper and pixels to screens. I gave you a brief overview of the tools and medium you can use to create wireframes so you understand the breadth of your options.

In the next two chapters - 4 Non-Digital Wireframing Weapons and 4 Digital Wireframing Weapons - I'll cover each of their strengths, weaknesses, and next best alternatives so you know where to start.

4

4 NON-DIGITAL WIREFRAMING WEAPONS

A comparison of the different tools for wireframing - choose wisely

THE PEN IS MIGHTIER THAN THE SWORD

In this chapter, I going in-depth on the analog tools and medium used to create wireframes, their strengths and weaknesses, and how they relate to wireframing and prototyping software. I'll detail wireframing and prototyping software in the next chapter, 4 Digital Wireframing Weapons.

Beyond Wireframing Software

Over the years, I've heard the terms "sketches", "mockups", and even "prototypes" used synonymously with "wireframes." Initially, wireframing tools were paper-only with what many now consider basic (not comprehensive) layouts. And the initial

versions of digital wireframing and prototyping tools were either incredibly austere and covered the wireframing niche of product design (i.e. Balsamiq circa 2008), or exceptionally complex and bridged the product design stages, namely wireframing and prototyping (i.e. Axure circa 2003). While digital tools proliferate, the old paradigms of paper wireframing and prototyping remain, leading designers, product managers, and engineers alike to further conflate the terms used daily.

Fifteen years since the first wireframing and prototyping software applications were built, the UXPin team sees lines between product stages - and corresponding terminology - blurring as product iteration cycles shorten, team composition and process varies more widely, and team members have more influence over communication and style preferences, and development is much more customer-centric which varies by product and industry. The bottom line is that every team develops its own unique product design process to meet their goals with the resources they have, and wireframing categorically has the best balance of structure and flexibility to accommodate myriad ways of getting the job done.

Below, I've laid out the most common types of analog tools and medium used for wireframing.You can also read more about paper tools in UXPin's article Better Use of Paper in UX Design. Many pundits may disagree whether or not some of these are relevant to wireframing, but if it looks like a wireframe and smells like a wireframe, I'm calling it a wireframe. No matter what anyone tells you, you can achieve the majority of your wireframing objectives - defining product structure, content, information hierarchy, functionality, and behavior - through any of these. What were once concretely "prototypes", "sketches", or "mockups" may now be variations of "wireframes."

I'm not here to hold you back with semantics - just get the wireframes done!

Sketching

You can sketch with anything that makes pigment on a surface - there are many beautiful options. So get out your pens, pencils, markers, crayons, or Paper App and start sketching!

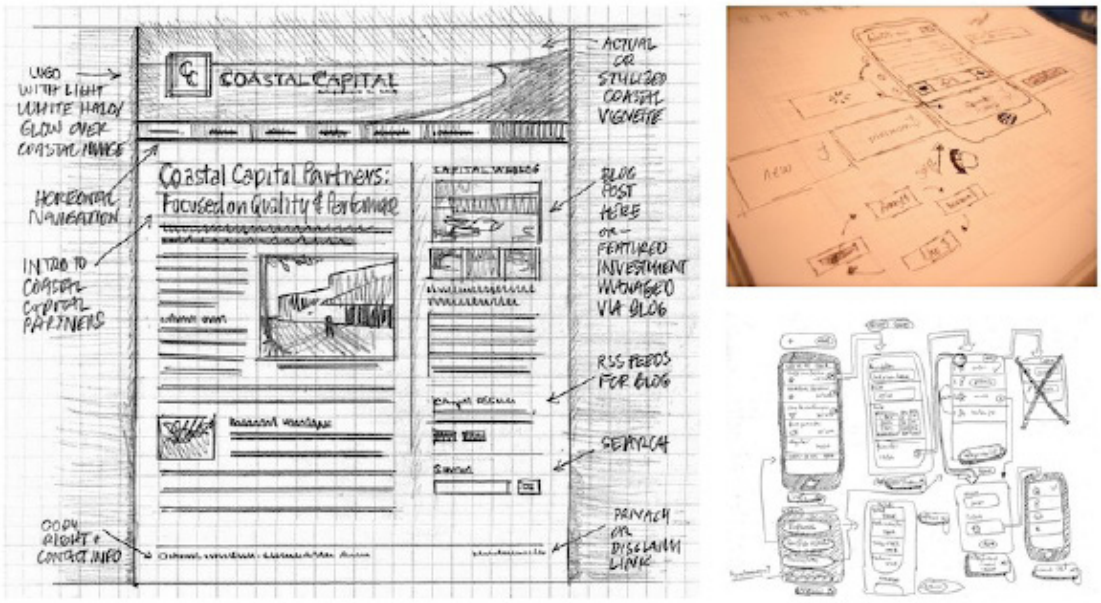


image: *Mike Rhode, Benoit Meunier, Fernando Guillen*

Pros

- **Speed** - It's really quick to pull out a piece of paper or open a blank sketching screen, and start drawing out product design ideas at any level as they come to you. You don't have to search through element libraries, switch stylus or cursor tool, memorize keyboard shortcuts, or rifle through files, which all interrupt your thought process. As documentation of the product matures (ex: existing wireframes, maser templates, and saved element libraries), however, wireframing and prototyping software can become faster even at this stage.

- **Flexibility** - You can easily switch from high to low-level product concepts or details, and even include them on the same page. Wireframing and prototyping apps are inherently more structured or focused - and thus constrained - which may not accommodate your full thought process especially at the earlier product design stages. While traditional wireframing apps like Balsamiq intentionally

constrain the user, newer wireframing software like UXPin is blurring the lines between sketching, wireframing, flowcharting, and prototyping to offer more flexibility across the full design process.

- **Limited Distractions** - You can really focus on the product structure, content, information hierarchy, functionality and behavior - instead of visual design - by having few if any options for fonts, colors, or general styling. Some wireframing tools do this intentionally for the same reason.

Cons

- **Duplicate Work -** You don't have master templates or elements to re-use or apply changes across different views. You also can't neatly edit your sketches, especially as you go into more detail - sure you can erase or undo lines, but it quickly becomes tedious to make multiple changes. Finally, once you've got a finalized sketch, you can't export that work to another application to pick up where you left off. All of your hard design work has to be transcribed somewhere else, whether a graphic editing, presentation, wireframing, or other tool. Unfortunately, there's still some level of duplication between wireframing and graphic editing for a host of reasons, but products like UXPin, Axure, Proto.io, and Just In Mind provide significant overlap between those two stages to reduce duplicate work.

- **Low-Fidelity -** You can't include media or fonts and font formatting, and it's far more difficult to visually show interactions or animations than the digital tools I described below. Depending on the tools you're using for sketching, you may also find limitations in providing color fidelity. I covered these types of high-fidelity wireframes in the second section of UXPin's 50 Shades of Wireframing.

- **Non-Interactive -** Beyond limitations of visually showing interactions, you also can't link sketches together. And forget about animations or advanced interactions - there are far better ways to convey this information. Amongst wireframing and protoyping software, UXPin, Balsamiq, Axure, and Proto.io probably provide the best solutions for this.

- **No Version Control -** This is an inherent issue with all analog tools since they're not timestamped, you have no idea who made changes, you can't make copies, and you can't easily flip through versions and compare changes made. UXPin and Balsamiq both have revision history to address this problem.

- **Limited Collaboration -** Because there's usually only a single, non-centralized version of a sketch, collaboration around them mainly happens if and when the person who drew them decides to show share them, possibly in a formalized meeting. Before or after that point, few if anyone has access to them. Consider this in stark contrast to UXPin that automatically alerts teammates when edits or comments are made. InVision is also a great solution but you have to upload your sketches first since it's more of a prototyping and collaboration tool than a wireframing tool.

- **No Standardization -** Elements can be inconsistent if you're not careful, and proportions could be off, or sizing completely off which could impact a lot of information conveyed in sketches. Stencils, wireframing and prototyping apps, and wireframing templates for graphics editing and presentation software provide a better solution for this problem.

- **Ugly -** For the majority of you who haven't gone to art or design school, your sketches could be so bad that they're rendered useless. Consider that before spending too much time on sketches.

Next Best Alternative

Whiteboarding (or Blackboarding) due to the vast similarities between the two.

Black & Whiteboarding

Similar to sketching with a few minor but important differences, including that they're vertical and typically bigger in size. While blackboards are being replaced by whiteboards across the World, there are tons still laying around - so I wanted to

mention them. As for whiteboarding, you can either pay a lot of money for official whiteboards, try Whiteyboard, make them cheaply out of glass or tile board, or use a whiteboarding app like Conojo.

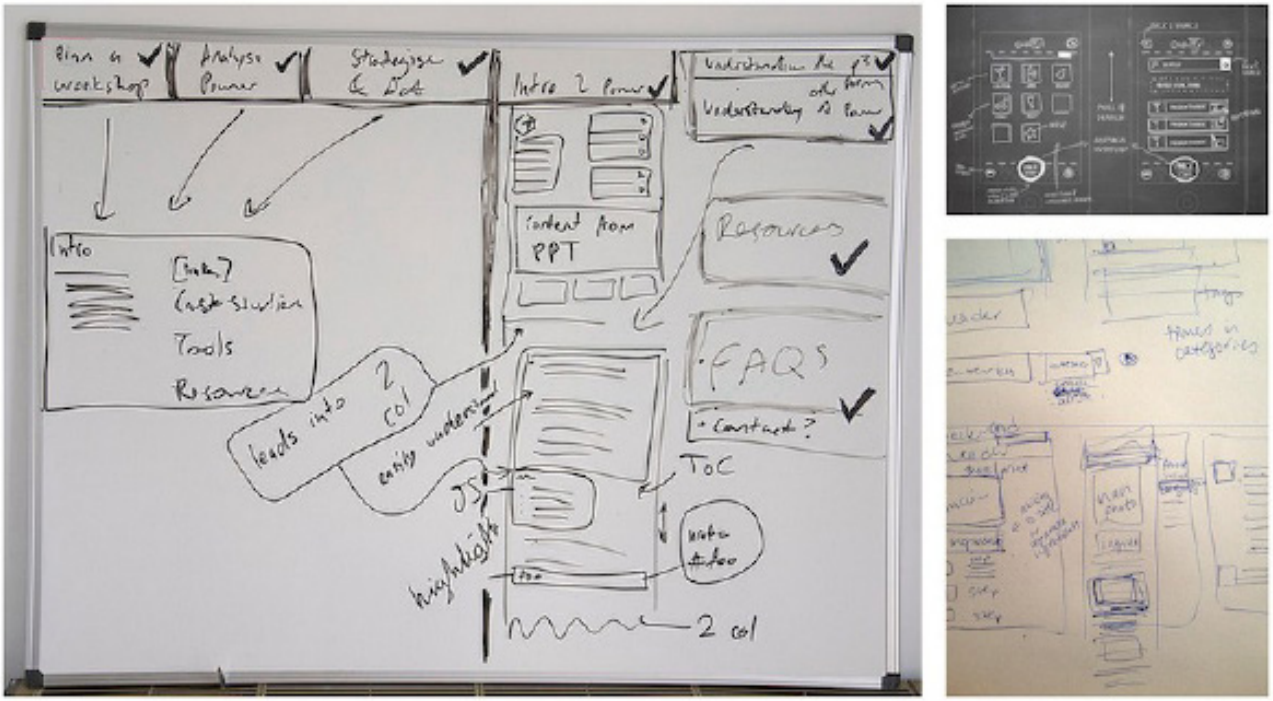


image: Yandle, Christos Chiotis, Chirstine Rondeau

More wireframing and prototyping software are also starting to provide whiteboarding functionality or something similar to it. For example, InVision now offers a whiteboarding options to better incorporate this into the full product design process, while UXPin provides real-time collaboration so different team members can work on the same wireframes together, no matter how simple or complex.

Pros

- **Similar to sketching (see above)**

- **Scale -** The larger size of this wireframing medium really allows you to start at a higher-level in the product design process, and visually tie different aspects of the product design together, low- and high-level. With the capability to zoom-in/out, most digital wireframing tools also provide a relatively large workspace although the view is ultimately limited by your screen size.

- **Prominence -** Most offices I've seen are decked out with whiteboards or glass

walls in the halls and in every room, with notes and diagrams scattered everywhere. This is great for product design transparency as anyone who passes by these whiteboards can understand what's going on and possibly even contribute - this is a stark contrast to sketches. Due to the collaborative nature of their job and the abundance of meetings they typically have, whiteboards are an extremely popular wireframing tool for product managers.

Cons

- **Similar to sketching (see above)**

- **Fixed Orientation -** Being mounted to a wall can make it more difficult to sketch, and may be a dealbreaker for some. It's surprisingly a real skill to whiteboard well.

- **Limited Details -** The level of detail is usually limited unless you're really good with blackboards and whiteboards - or have really sharp chalk or markers. Either way, it's probably more time than it's worth to go into too much detail on a whiteboard because graphic editing, presentation, and wireframing tools are better at this.

Next Best Alternative

Sketching due to the vast similarities between the two.

Paper Cutouts (i.e. Paper Prototyping)

Similar to digital or sketch wireframing depending on how they're produced. You can use standard paper wireframing kits (which was the precursor to the UXPin wireframing and prototyping software), or simply cut out paper sketches or digital wireframe printouts - you can always cut down edges later for "re-sizing." Although Common Craft's famous Dropbox explainer video isn't about wireframing, it illustrates how fun this type of wireframing can be.

image: UXPin, Flickr, Winnie Chang

Pros

- **Fidelity -** Because you can use digital wireframing tools or sketching to create the initial elements, you get to choose the level of fidelity you want.

- **Standardization** - Because you're either creating a bunch of the same cutouts or re-arranging the same cutout, your wireframe elements are inherently standardized.

- **Interactivity -** You can pin pin them on the wall, wrap them around your mobile phone, tape them on your computer screen, move and rearrange elements. Adding some basic interactivity to your sketched or digital wireframes can be really effective in understanding the user experience.

Pros

- **Similar to sketching or digital wireframing -** Depending on how you create the initial interfaces you're cutting out.
- **Speed -** They're more time-consuming than sketches at first but can save time if you expect to have to rearrange a lot of wireframe elements. And this is also

an extra step if you're making digital wireframes in applications like UXPin which have basic and advanced interactions already.

Next Best Alternative

Wireframing Software because it offers similar fidelity flexibility, standardization, and interactivity without as many of the drawbacks.

Stenciling

Similar to sketching with a few minor but important differences, including standardization and structure. There are many templates you can print out, stenciling kits to make your own, and full paper wireframing kits that come with standard paper elements like the UXPin paper product I mentioned above.

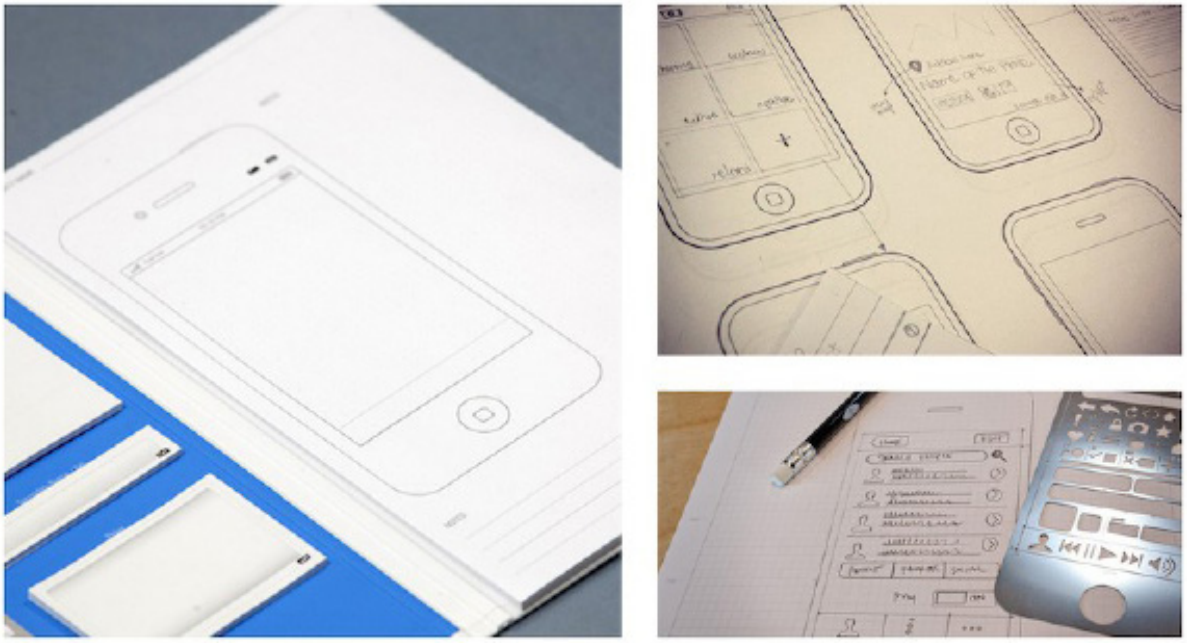


image: UXPin, Natalia Sourdis, iphonized

Pros

- **Similar to Sketching (see above)**

- **Speed** - Stencils act as a master templates for standard elements you may use many times, while allowing you to sketch the rest of the details. Depending on

your skills and preferences, it can be as fast if not faster than sketching wireframes.

- **Standardization** - By standardizing details (such as hardware and product elements) across your sketches, it becomes easier to see the differences between sketches and place more deliberate emphasis on certain details like layout, key features or content, flows and so forth.

- **Structure** - Cleaner and better-aligned lines that more closely resemble digital tools also lend more structure to your sketches so you can get a better understanding of how product content and features fit together.

- **Cleaner** - Unlike the other analog wireframing methods above, stenciling is typically a cleaner presentation for anyone who isn't a trained designer or has a shaky hand.

Cons

- **Similar to Sketching (see above)**

- **Element Constraints -** Some of the flexibility of sketching is lost when you start stenciling because you're limited to what the stencil offers. If you want to go beyond these standard elements, you're purely sketching again.

- **Outdated Elements -** Unlike wireframing software, you don't get updates on your element libraries. So each stencil becomes increasingly useless with each new release of the frameworks with which you're building or the platforms on which you're building.

- **Focus on a Tool -** One of the major reasons people love sketching and black- or whiteboarding over digital wireframing tools is that they focus less on the tool and more on the substance. This benefit starts to get lost with a constrained tool like a stencil.

- **Multiple Styles -** This is one of the few wireframing techniques that uses multiple styles in it's composition. The merging of free-hand sketching and clean, ruled lines can be distracting.

Next Best Alternative

Wireframing Software because it offers similar standardization, structure, and cleanliness without as many of the drawbacks.

Use Whatever Works

In 4 Digital Wireframing Weapons, I'll dive into digital tools for creating wireframes.

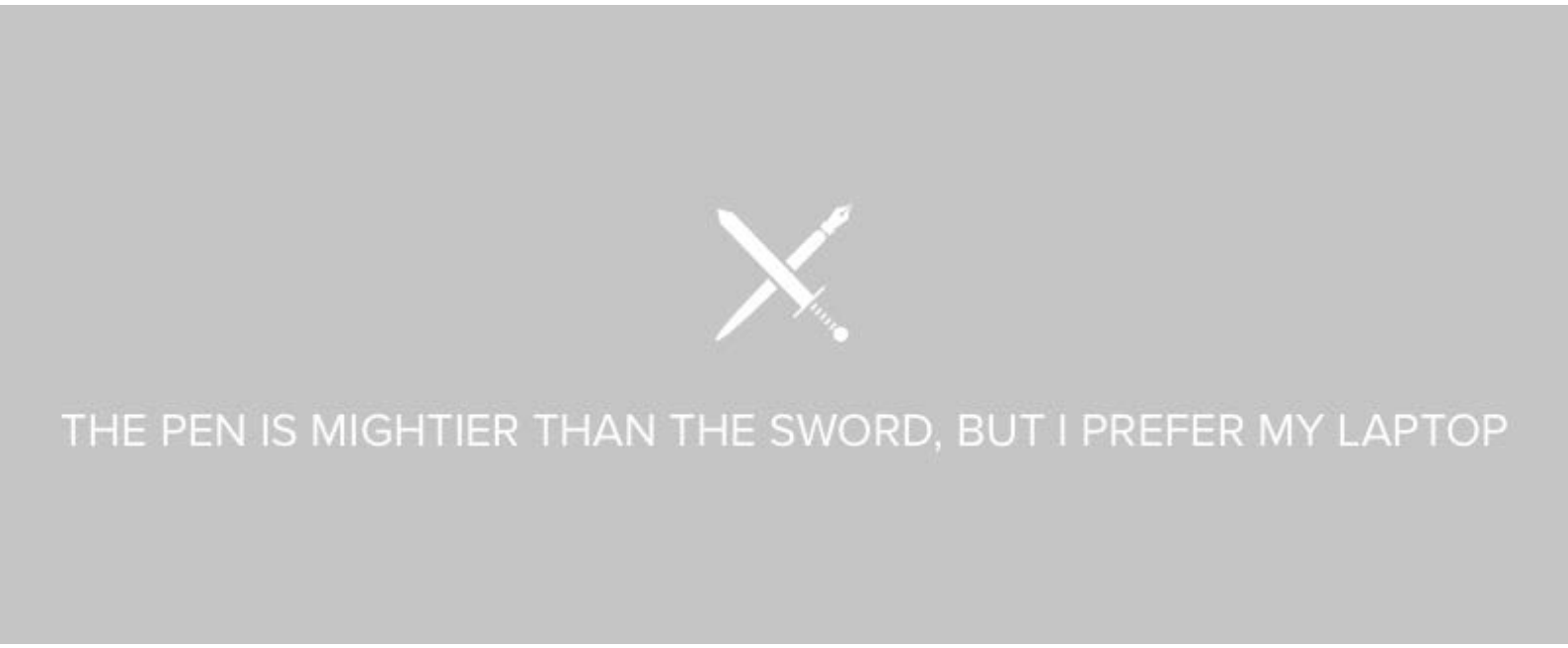
If your free-hand sketching skills simply suck, and you think (or know) you'll spend more time erasing, scribbling-out, scrapping, and overwriting your sketches instead of producing a useful wireframe, digital wireframing tools are for you. As you'll see, there are many other reasons digital wireframing tools ought to be the primary method for creating and maintaining wireframes. I personally know a lot of amazing designers and product managers who go straight to the software and never sketch, use paper cutouts, or stencil.

As you think more about how you want to create wireframes, remember two things - it's okay to use more than one tool in your wireframing process, and do whatever it takes to get the right wireframes done. Don't let semantics hold you back!

5

4 DIGITAL WIREFRAMING WEAPONS

A comparison of the different tools for wireframing - choose wisely



In the previous chapter, I covered analog tools and medium to create wireframes. Below, I'm going in-depth on the digital tools used to create wireframes, their strengths and weaknesses, and how tools like UXPin are evolving to address the evolution and diversification of product design.

Word Processing Software

Everyone (yes, everyone) has used some form of word processing software in their lives - Google Docs, Microsoft Word, Apple Pages, and so on. Because this is a much

less common tool, it was very difficult to find examples online. But here are some instructions if you want to take this route.

Pros

- **Familiarity -** We've all used them so we know what we can do with them. And since most word processors offer the very basic elements of wireframes - shapes, colors and text - you can theoretically use them for this purpose. It's definitely an option if you're absolutely terrible at using everything else.

Cons

- **Similar to Presentation Software (see below)**

- **Basic Elements -** This is even more limited than presentation software, although you can always import images and adjust the arrangement and set them above or below text to achieve this goal. It will be far more time-consuming than the other tools mentioned, however.

- **Difficult to Layout -** Changing the arrangement of text and pictures will easily get you started, especially the "bring in front of text" command so you can move elements around freely. However, I've often run into issues when re-arranging elements that are higher up on a page than others. Because word processors are designed to adjust surrounding content as you make changes to simplify the document creation process, it actually works against you when trying to rearrange elements independently of one another. Essentially, you can't set or lock elements into place like you can with almost all other software.

Next Best Alternative

Presentation Software is the next best alternative because you're probably familiar with a program, and it offers far more functionality to let you create beautiful basic wireframes. While Adobe Illustrator is the most similar to word processing while giving you more control over shapes, colors, and text, you'll quickly realize how simi-

lar the limitations are between the two. And, of course, if you're not stuck in your ways with tools you're already familiar with, I'd suggest going straight to dedicated wireframing and prototyping software - they exist solely for that reason.

Graphics Editing Software

There are many digital graphics applications (free and paid) that could be used but the vast majority of designers and product managers use Adobe Illustrator, Photoshop, or Sketch for graphic design already - so those are clear winners if you choose to use them for wireframing too. But you could also try out GIMP, Pixlr, or another popular free image editors.

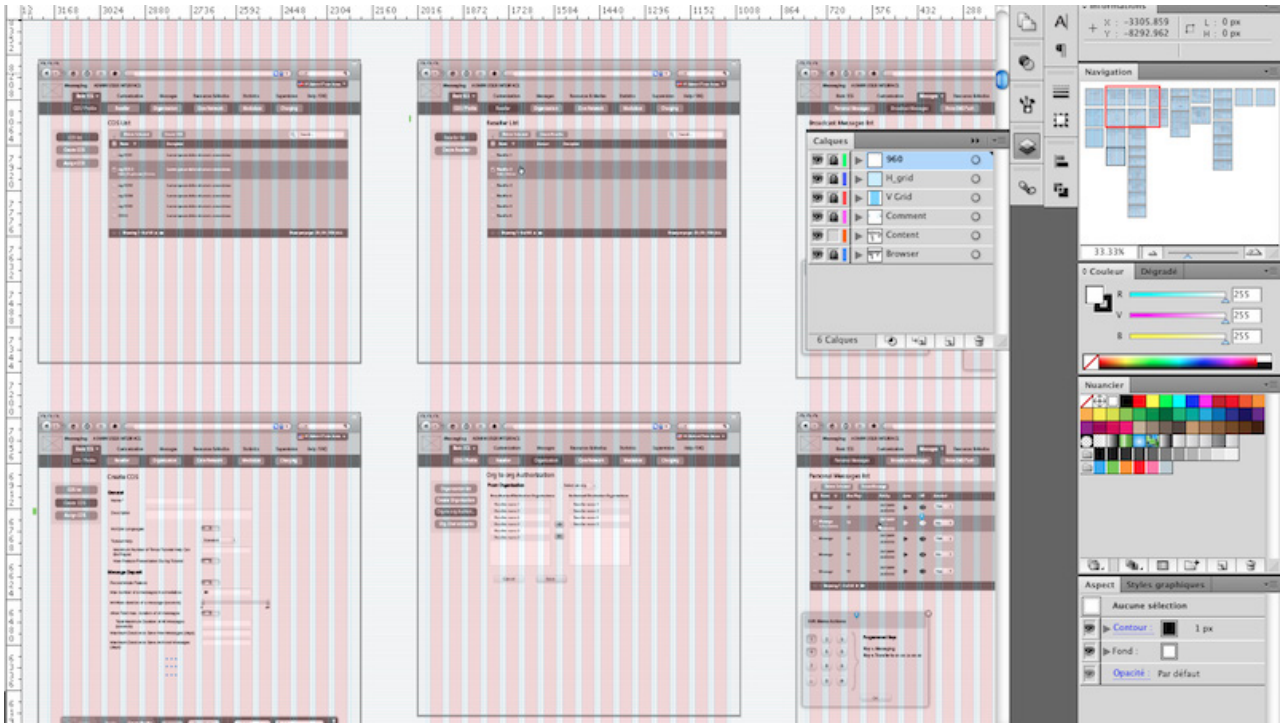


image: Marc Decerle

Pros

- **Familiarity -** Marshall McLuhan once said, "We become what we behold. We shape our tools, and thereafter our tools shape us." Since the vast majority of designers, engineers and product managers have used or regularly use Photoshop or Illustrator, they've adapted their workflow around those tools over time. If you are a power user of this software, the speed gap compared to other wirefra-

ming tools closes - even dedicated wireframing and prototyping tools like UXPin. To wit, advanced usage of master templates, smart elements, existing graphics work, workspace organization, and/or keyboard shortcuts could potentially make you as effective in Photoshop or Illustrator as UXPin for basic types of wireframes - but not for advanced interactions, animations, user flows, and many of the other features provided by dedicated wireframing and prototyping tools. And if I had a timer and gave you a few days to learn the app, you'd probably be faster in UXPin.

- **Wireframing Element Libraries -** First and foremost, there are many wireframing templates and toolkits which turn this app into a pseudo-wireframing tool - a very basic wireframe is mainly shapes, colors, and text after all. As you work more in this software, you can leverage master templates, smart elements and wireframes or wireframe elements you've already created if you're smart about it - this takes a lot of discipline, though, and I've heard more people mess this up than do it well.

- **Fidelity -** With the ability to use almost any shape, text, or color your heart desires, you can literally create almost any wireframe you can imagine, not constrained by wireframing element or icon libraries. You can add any level of fidelity (i.e. media, color, and text) in here with the exception of interactions and animations. Many wireframing tools limit your ability to do this intentionally. However, that's gradually changing thanks to products like UXPin, which has been launching many high-fidelity features lately with image editing, gradient editing, one of the most beautiful color pickers on the market, and many more to come.

Cons

- **Feature Bloat -** For the same reason these tools are so flexible, there's also a steep learning curve and a lot of ongoing distraction from feature bloat. This is mainly due to these products not originally being intended for wireframing. Moreover, they've evolved to accommodate more types of graphic work - in a sense, trying to become "all things to all people." As a result, these tools are typically not as quick or effective as wireframing and prototyping applications, or analog

methods for anyone who isn't an advanced user. With that said, Sketch, a new and popular graphic editor in town, is attempting to solve this feature bloat problem through a focus on the essential features (a la the 80-20 rule), less obtrusive layers, drawing tabs within a project. However, it's limited compatibility with Illustrator and Photoshop and poor .svg support have been significant drawbacks to using this as an option at all, especially in larger organizations where everyone would have to switch.

- **Non-Stock Element Libraries -** Although you can find wireframing element libraries to import, it can be cumbersome to find the right ones instead of using the stock or community-generated libraries provided by dedicated wireframing and prototyping software. And, if you're not aware of these libraries, you may run the risk of spending time making your own from scratch or not providing the right level of detail because you don't know what typical wireframes should look like.

- **No Collaboration -** Most graphic editing applications don't offer any collaboration within the application - it's one of the main reasons Flinto and InVision entered the market. For example, you can't edit the same document together in real-time or invite a colleague to leave comments or annotations - which is important when working on content and information hierarchy. Moreover, there aren't dedicated areas or functionality in the software to document additional details about the product structure, functionality or behavior. While presentation applications like PowerPoint or Keynote provides additional areas for annotations and better illustrates user flows, wireframing and prototyping apps provide the total package, especially applications like UXPin or Axure which allow you to produce higher-fidelity wireframes that come very close to final mockups you'd produce in graphic editing software. Oddly enough, Adobe Creative Suite once offered reviewing services then discontinued them - hopefully this will change, for their sake.

- **No Presentation -** You can export to PDF, email a file, or share your screen. That's about it.

- **No Flowcharting or User Flows -** It simply doesn't exist. You can't export ima-

ges to a PowerPoint, or visualize them in series (i.e. a user flow). The best you can do is export them to UXPin, Axure, InVision or Flinto individually look at them side-by-side or create links between them to visualize more granular user flows.

- **Non-Interactive -** It simply doesn't exist. You can't link wireframes or mockups together, show animations, create clickable buttons, menus, or other elements in these tools. Prototyping tools like Invision (for web and mobile) and Flinto (for mobile) entered the market to address this issue. Additionally, wireframing and prototyping tools like UXPin are addressing this interactivity issue both at the wireframing and at the prototyping level so product teams can build high or low--fidelity prototypes (or interactive wireframes) - so they can better test usability and user experience at any stage in the product design cycle.

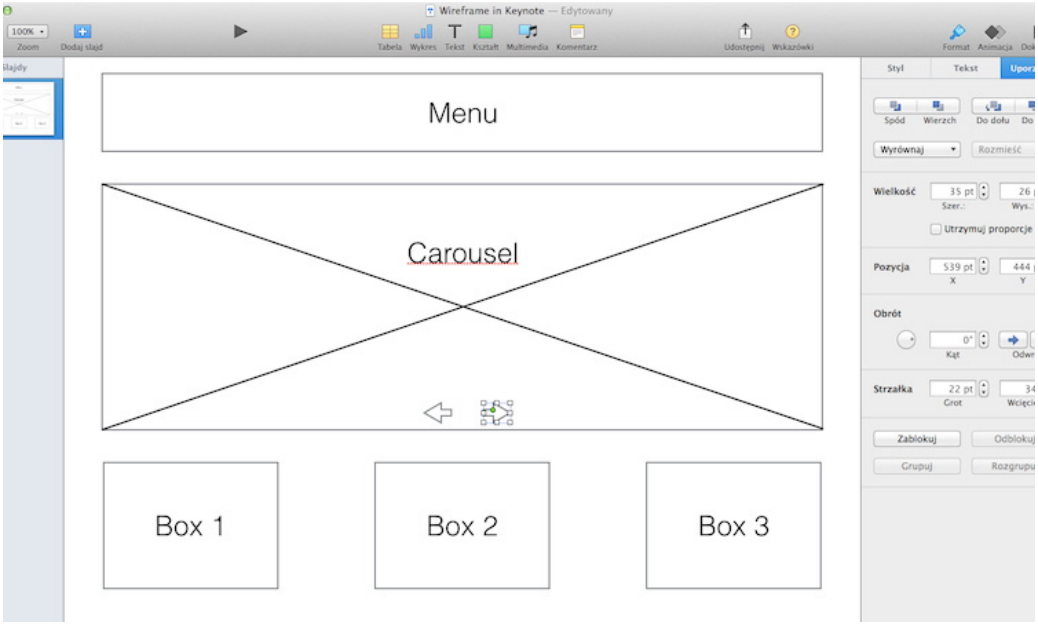
- **Fidelity Creep -** Whether or not you import and use third-party wireframing element libraries, you run the risk of not providing the right level of detail or fidelity desired by the team. There are so many "adjustments" you could make - even if you're not supposed to.

Next Best Alternative

Wireframing and Prototyping Software because they not only cover the basic wireframing functionality offered in graphic editors (shapes, text, and colors), but they offer interactions, animations, user flows, and much more. Moreover, some modern tools like UXPin are starting to integrate with and overlap with the basic functionality of graphic editing applications to make the decision easier. For many product teams, the better option is wireframing software because they're focused solutions for wireframing, and a central medium for collaborating on wireframes and integrating with other tools that are important in the entire product design lifecycle.

Presentation Software

As with word processing software, most people have used presentation software at some point in their lives, and almost every businessperson has - it's probably mandatory in US education by now. The broad familiarity of this tool makes it another great basic option for creating and presenting wireframes to teams. Although the predominant choices are PowerPoint and Keynote, you can also use Google Presentations, Prezi, or another popular free presentation tool.



Pros

- **Familiarity -** As with word processing software, you know the basics. And it's really easy to learn more advanced features like animations, slide transitions, linking slides for interactions and so forth.

- **Wireframing Element Libraries -** Because of wireframing libraries like Keynotopia, this app can quickly become a pseudo-wireframing tool with the basic shapes, colors, text, animations, hyperlinks, and focus on UI narratives and user flows you need. As you work more in this software, you can also leverage master templates, and reuse slides or parts of slides when making changes to your wireframes. It's far easier to be disciplined about this in presentation software compared graphic design apps.

- **User Flows -** By nature, the application takes you through a linear flow. This forcing function is great at making you think about the actual product experience instead of separately, pretty wireframes or mockups. For more advanced users, you can actually link presentation slides in more complex ways to showcase many user flows that don't necessarily follow the linear progression of a presentation. Unfortunately, most wireframing and prototyping software is clunky when it comes to visualizing user flows but UXPin, Flinto and InVision do a great job.

- **Interactivity -** Whether you're showcasing a full-bleed wireframe or a wireframe with annotations, you can add hyperlinks between slides and triggered animations within slides to visualize desired interactions. I've personally done this quite a bit and love the experience, it can be extremely time-consuming and not give you the desired results. Also, once you start playing around with advanced user flows and interactions, however, it becomes more difficult to incorporate slides with supplemental notes or other contextual information about the product design. Wireframing tools like UXPin, Axure and Proto.io are better for this.

Cons

- **Non-Stock Element Libraries -** Similar to the issues in graphic design software, it can be cumbersome to find the right element libraries, if they exist at all. Unlike dedicated wireframing and prototyping apps and their communities, presentation libraries like these aren't updated as frequently and the library quality almost always suffers. Alternatively, you may run the risk of spending time making your own from scratch or not providing the right level of detail because you don't know what typical wireframes should look like.

- **Limited Collaboration -** Most presentation software doesn't offer any collaboration within the application, with the exception of Google Presentation where multiple people can work on slides at the same time and even leave comments. However, software outside of Keynote and PowerPoint seriously lack in interactivity, manipulation of graphics, and shape, text, and color options that even make this a tool worth considering. If you care about collaboration without compromising other benefits, you should stick with UXPin, Balsamiq, Axure, Proto.io or

other wireframing apps - or check out InVision or Flinto for prototyping software.

- **Limited Flow Charting & User Flows -** As I mentioned before, you can actually communicate pretty advanced user flows. Keynote and PowerPoint actually have flow diagram images for high-level flows, and you can link slides together to show many user flows that don't necessarily follow the linear progression of a presentation. However, it's not easy to do and the flow charts (or sitemaps) aren't linked to the wireframes themselves in ways that product like Axure or UXPin do - the UXPin team is even aware of these limitations, and working on an even tighter integration between these two features.

- **Limited Interactivity -** Clever users can actually get pretty far if they use all of the features in Keynote or PowerPoint. And that may be all you need. Once you start to think about the ease of adding basic interactions within wireframing and prototyping software, and the breadth of options when you start to think about all the combinations of elements, content, views, and animations, you quickly come to realize why people use wireframing and prototyping software to communicate the functionality and behavior of the product. UXPin, Axure, InVision, and Flinto are pushing the envelope here.

Next Best Alternative

Wireframing and Prototyping Software because they not only cover the basic wireframing functionality offered in presentation software (shapes, text, colors, user flows, and interactions), but they offer more advanced interactions and user flows, among other things. Additionally, an increasing number of wireframing and prototyping applications allow you to export to presentations so you get the best of both worlds here.

Wireframing & Prototyping Software

There are many wireframing and prototyping software you can use, but only a few serious apps that are truly worth checking out, in our opinion. To name a few products, UXPin, Balsamiq, Axure, and Proto.io are great - these are not to be confused with prototyping-only tools like InVision or Flinto which do not offer most of the wireframing functionality I've been discussing. There are also myriad free, but very limited wireframing applications.

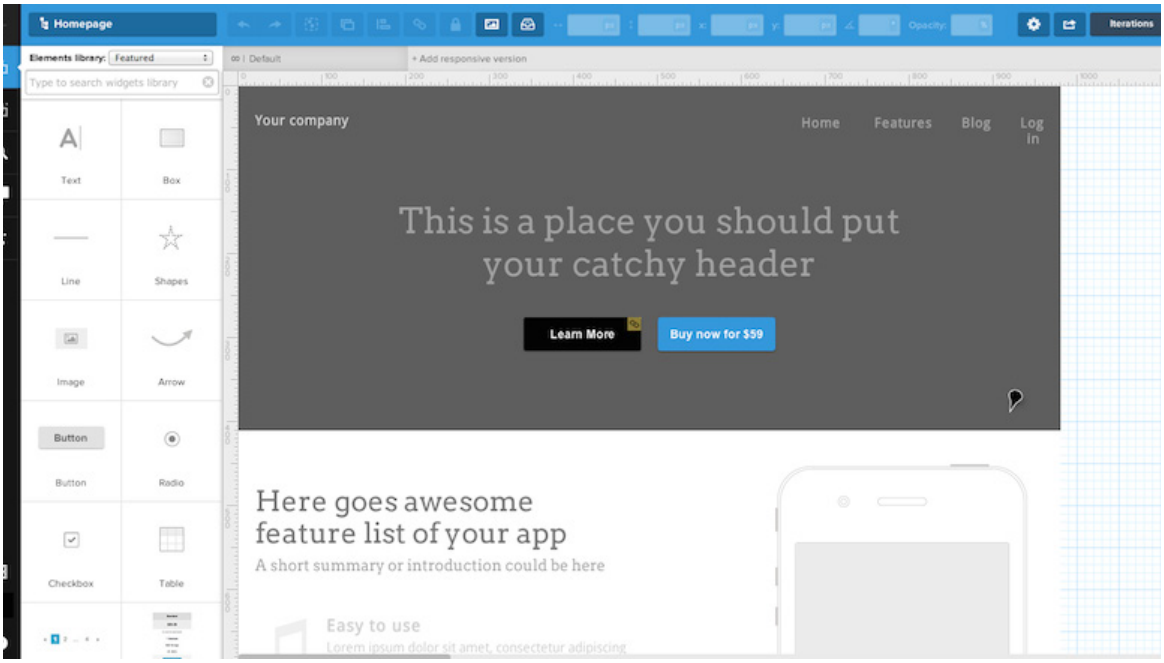


image: UXPin

Pros

- **Built for Wireframing -** This is self-explanatory, especially for more advanced wireframing tools that also provide prototyping functionality like UXPin and Axure.

- **Speed -** Because they're designed specifically for wireframing, they should be the fastest tools for creation, collaboration, and presentation of wireframes. For example, power users can actually be faster than stenciling or sketching because of how quickly they can create, copy, or adjust elements that would otherwise be drawn out. But we know there are many ways to get from point A to B in the world of wireframing, and often multiple tools are used to accommodate the cre-

ative thought process.

- **Element Libraries -** As I mentioned above, many of the other tools are mainly comparable to wireframing and prototyping software because of element libraries. However, they often have limited and lower-quality libraries if any - and they don't come stock. In contrast, UXPin has 900+ web and mobile elements, and 150+ UX patterns that combine many of these elements both for inspiration and to skip starting from scratch. And these libraries are updated regularly to keep up with the ever-evolving product standards - alternatives don't come close!

- **Advanced Flowcharting & User Flows -** Most wireframing and prototyping software offer at least a flow chart element comparable to what you'd find in presentation and possibly even word processing software. However, many of them also generate sitemaps as you create new wireframes or let you visualize wireframes side-by-side so you can navigate through them.

- **Advanced Interactions -** Product interactions and animations change so rapidly that only dedicated wireframing and prototyping tools can really offer you the best option for this. Sure, you can reference other products in your documentation and say, "Copy that," and you can get pretty far with presentation software for basic interactions, but there's better ways to achieve that objective. Simply put, use a tool like UXPin, Axure, Just In Mind, InVision, or Flinto for advanced interactions and see how far you can get before trying something else.

- **Collaboration -** When thinking about collaboration, you should be looking for a few key features. The first aspect of collaboration is commenting and resolving comments as a pseudo-task management system. Secondly, you must have the ability to allow multiple people to login to the same wireframe or prototype and edit them simultaneously. Additionally, sharing links to projects also facilitates collaboration because you can pull collaborators into wireframes and prototypes instead of throwing it over the wall so-to-speak. Finally, revision history and cloud storage both make it easier to sort through the archives and find the right work. Many wireframing and prototyping tools provide some portion of these features. While UXPin and InVision are probably the most robust, Balsamiq, Hot-

gloo, Just In Mind, and a few others have some level of collaboration as well.

- **Presentation -** This can either mean exporting to PDF, or presentation software, having a built-in presentation mode, or exporting to a web or mobile app so you can see a real-world presentation (a prototype) of what the app should look and feel like. Most software allows you to export to PDF while only a few like Mockflow export to presentation apps, and only web-based versions have links to projects that you can share. And many wireframing and prototyping tools like UXPin, Hotgloo, Moqups and InVision have a presentation mode and/or let you export to a web or mobile app.

Cons

- **Lack of Familiarity -** Simply put, this may require a few days or weeks of training, but there are a handful of tools like UXPin that are known for usability and ease of learning.

- **Limited Functionality -** If you're not happy with the element libraries, interactions, animations, colors, or level of collaboration and presentation, you may have to switch to a new wireframing tool that takes time to learn again. For this reason, Balsamiq has become one of the standards in the industry - until now, it has provided just enough to keep the broadest set of users happy. However, it is far from a complete solution today. Products like UXPin and InVision are quickly addressing the more of the expanding wireframing and prototyping landscape.

- **Limited Fidelity -** With the exception of tools like UXPin and Axure, most wireframing tools are low-fidelity. In fact, UXPin and Axure are the only tools to my knowledge that allow you to go from low-to high-fidelity wireframing within the same app. And it won't be long before some advanced wireframing tools deeply integrate into graphic design tools like Photoshop and Illustrator to make this process more seamless. In most wireframing apps, however, if you want higher--fidelity, you have to create mockups in a graphic design tool then export them to prototyping products like InVision or Flinto.

Next Best Alternative

None! Kidding. I'd probably say presentation software for many of the reasons I mentioned above. This is personally how I got started wireframing - but I was in business school at the time, so that was a natural fit.

Sharpen Your Skills

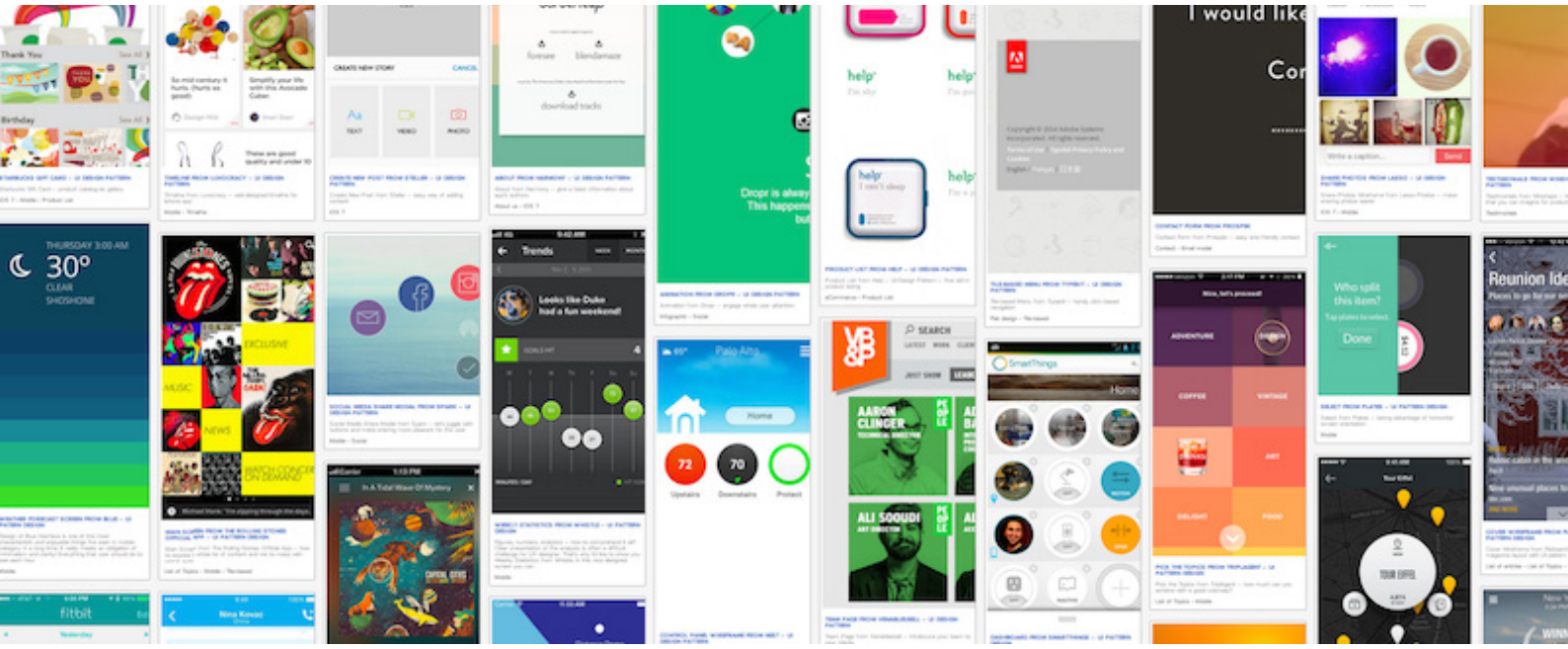
As you think more about how you want to create wireframes, remember two things - it's okay to use more than one tool in your wireframing process, and do whatever it takes to get the right wireframes done. Don't let semantics hold you back!

Whether you've been designing and managing products for years or just starting out, it will take you a lifetime of collaboration to stay on top of your game - although you'll never be perfect. In the next chapter, The Design Pattern & Wireframe Libraries Guide, I detail the very best sites on the web to give you inspiration, or to kickstart your actual work with examples that import directly into wireframing and prototyping tools like UXPin.

6

THE DESIGN PATTERN & WIREFRAME LIBRARIES GUIDE

The Very Best From All Over The Web



According to Ralph Waldo Emerson, "(Design) is a journey, not a destination." While he was originally referencing "life", broadly defined, many of you might agree with me that design, broadly defined, is synonymous with life in many contexts.

Recently, my team at UXPin scoured the web for the top web and mobile wireframe example and pattern sources, and came up with a pretty exhaustive list I wanted to share. It's my hope that they will help you immerse yourself in design more throughout life - to live design - and breathe life into your product designs - developing your purpose, planning or intent for the design of your next product.

Whether you're just starting to wireframe or prototype or you've been doing it for years, you'll hopefully find this useful for staying on top of mobile and web product design trends. Use them freely for finding inspiration, or kickstarting your actual work with examples that import directly into popular wireframing and prototyping tools like UXPin and Balsamiq.

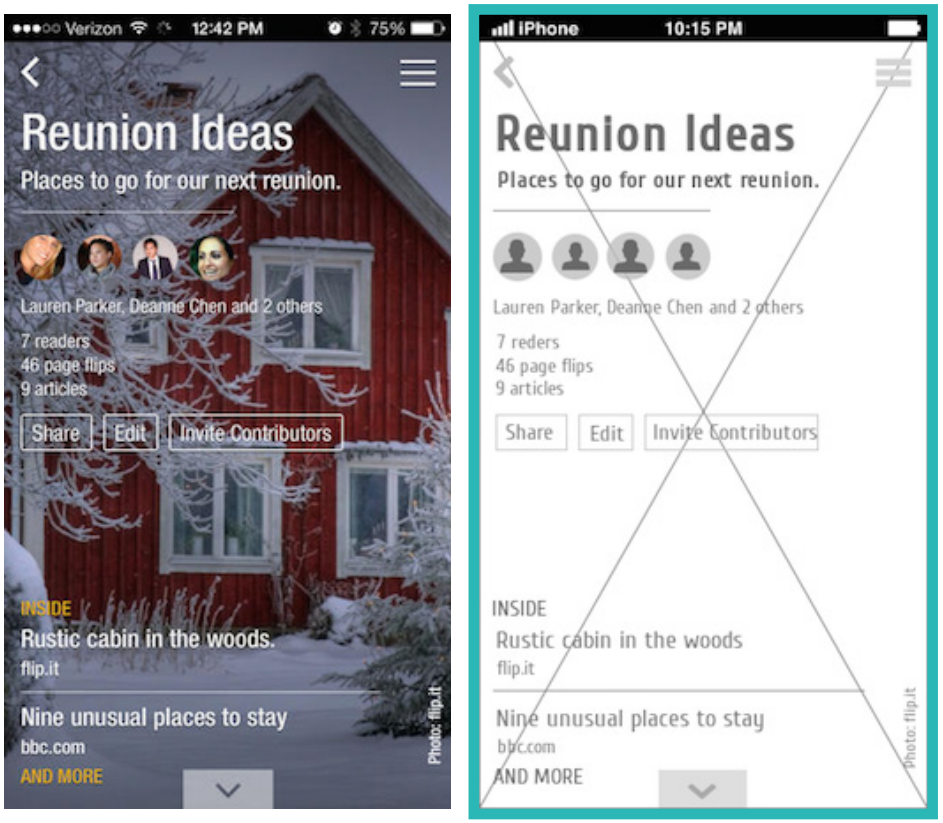
Make sure you also read up on UXPin's free Web Design Trends and Mobile Design Trends e-books, which cover some of the product design patterns you'll find in the libraries below.

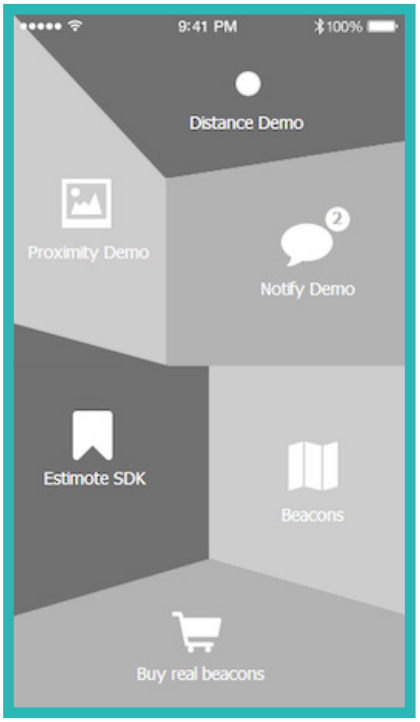
Importable Libraries

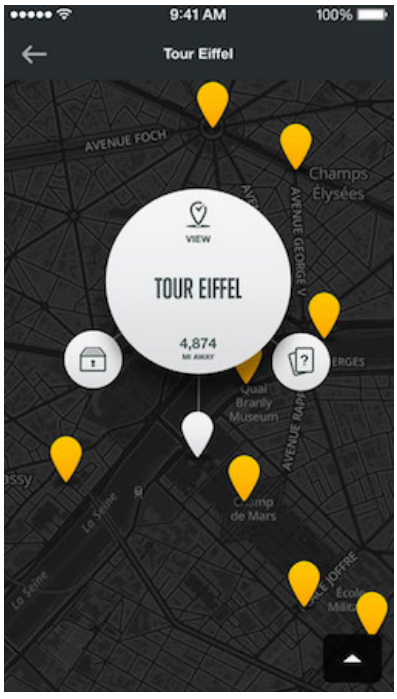
UXPorn by UXPin

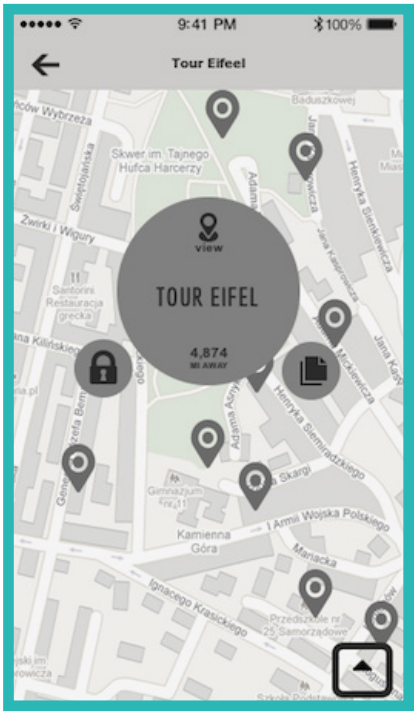
uxporn.uxpin.com

~150 UI design patterns and wireframe examples - and growing. It's a UX wireframe and patterns library that can be imported directly into the popular wireframing and prototyping tool, UXPin. For an increasing number of the UX patterns, there's a matching wireframe example to get you started in UXPin or use it for inspiration elsewhere - your choice.







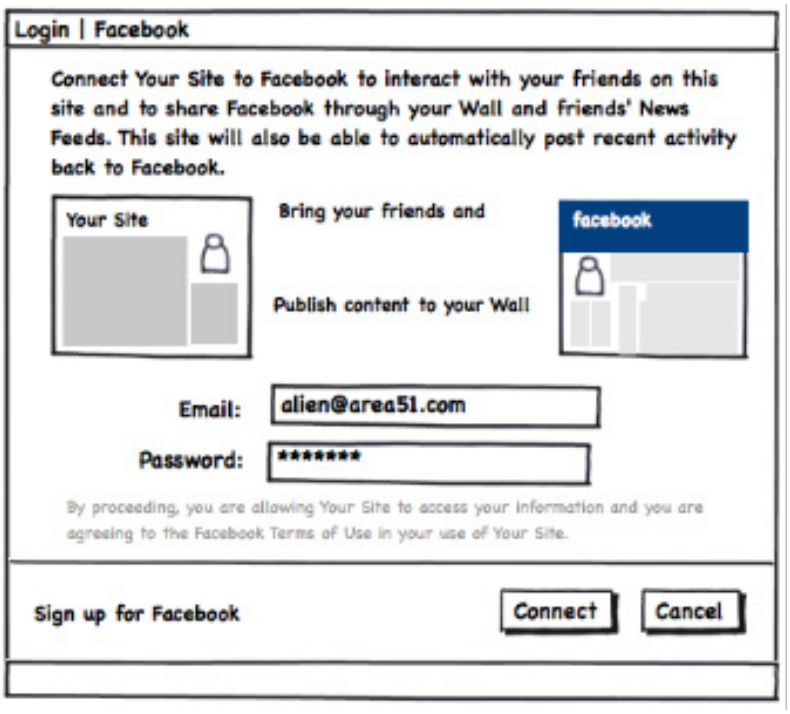


Click to use this wireframe in UXPin

Mockups To Go

mockupstogo.net

~100 wireframe examples, but no UI design patterns and longer growing and relatively outdated. It's a user-contributed collection of UI components and wireframe examples built using Balsamiq Mockups. Some of the mockups are full web pages. These are downloadable mockups that you can use as a starting point for your own designs in Balsamiq Mockups.



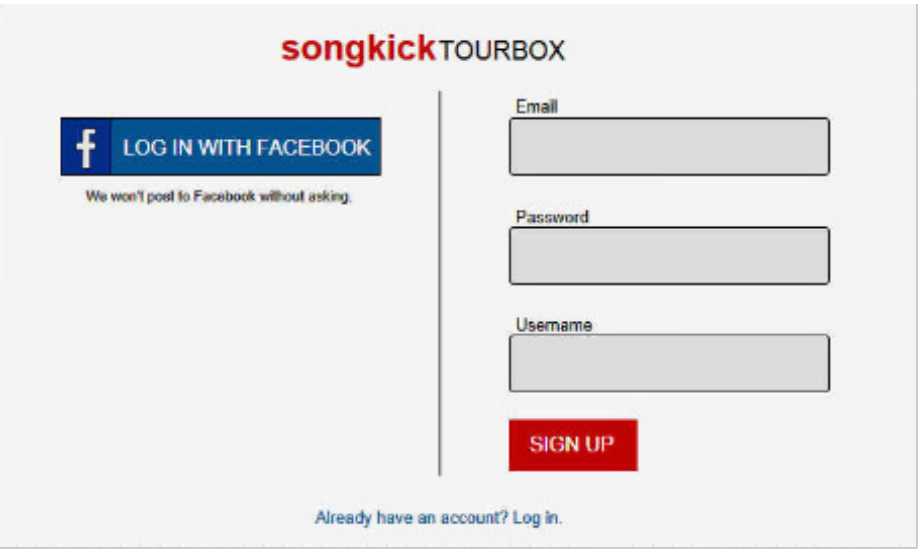
Standalone Libraries

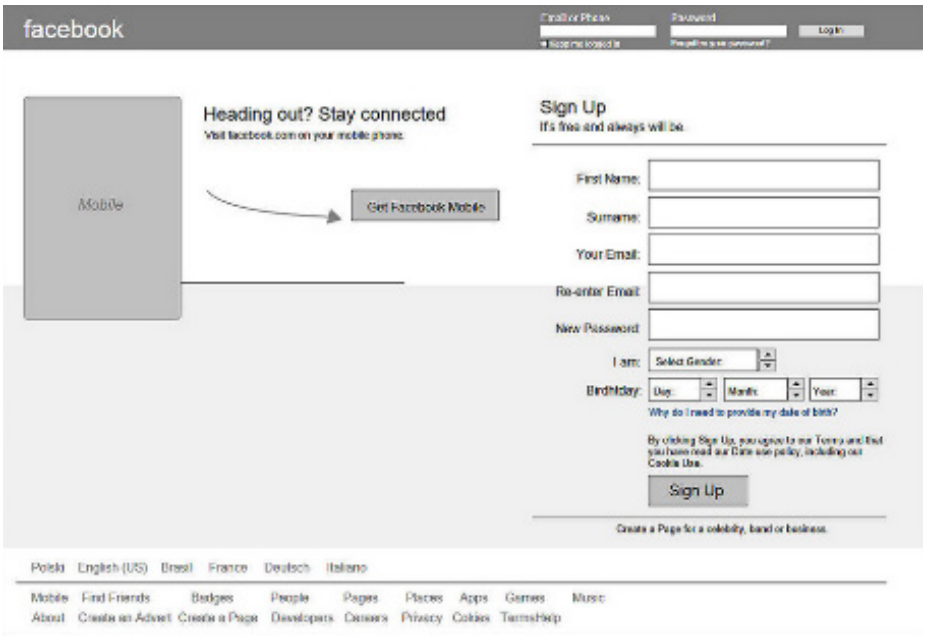
Web & Mobile

I Love Wireframes

flickr.com/groups/ilovewireframes

~1200 wireframe examples but no UI design patterns. It's a flickr group for everyone who enjoys creating wireframes.

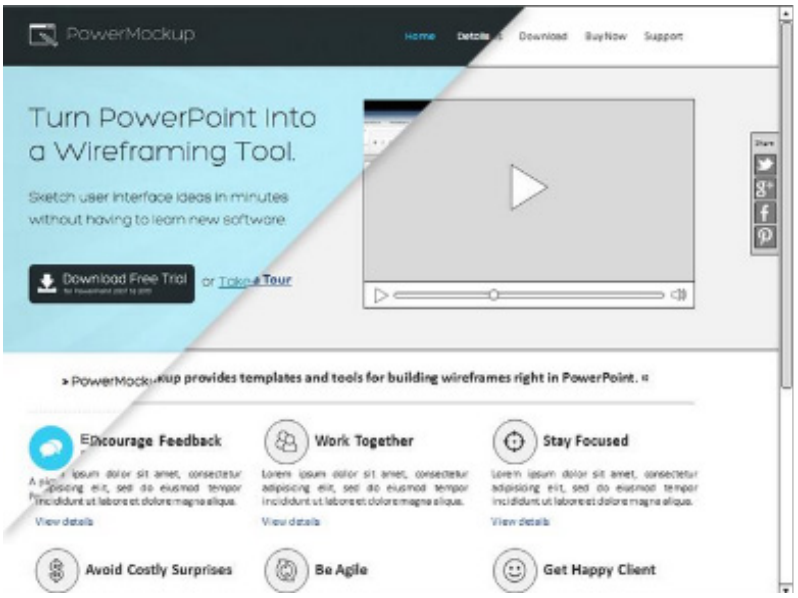




Wireframe Showcase

wireframeshowcase.com

Countless wireframe examples but no UI design patterns and no easy way to search archives - still very helpful if you have the time. It's a user-submitted blog of wireframes, juxtaposed with final executed site designs, with commentary. It contains wireframe examples in all their variety, from photos of paper prototypes in the workplace, to annotated screenshots describing the design process.



Web Without Words

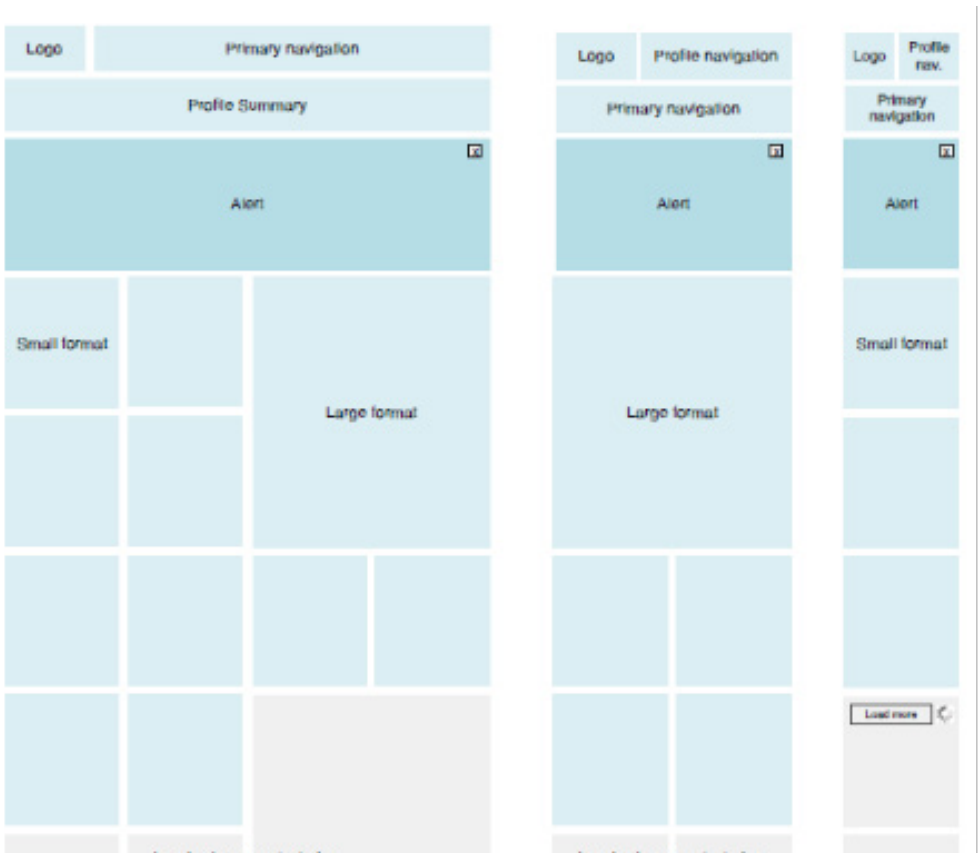
webwithoutwords.com

It's a gallery of popular websites, deconstructed by removing all words and images, abstracted into primitive wireframes. Pretty funky, no?

Wireframes Magazine

wireframes.linowski.ca/tag/wireframe

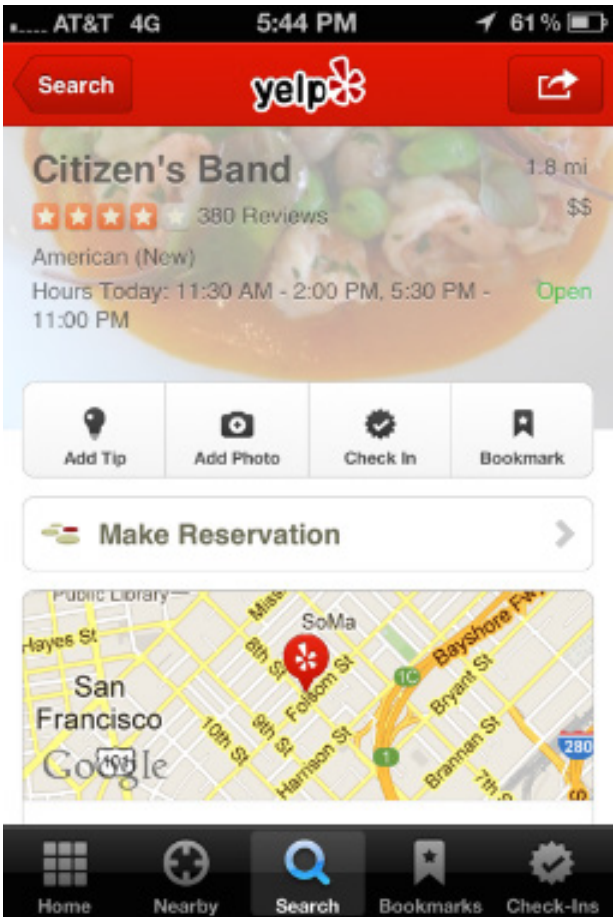
Countless wireframe examples but no UI design patterns and no easy way to search archives. It's not a design pattern or wireframe example library, but does have a great amount of examples. Wireframes Magazine is run by Jakub Linowski, an interaction designer from Toronto.

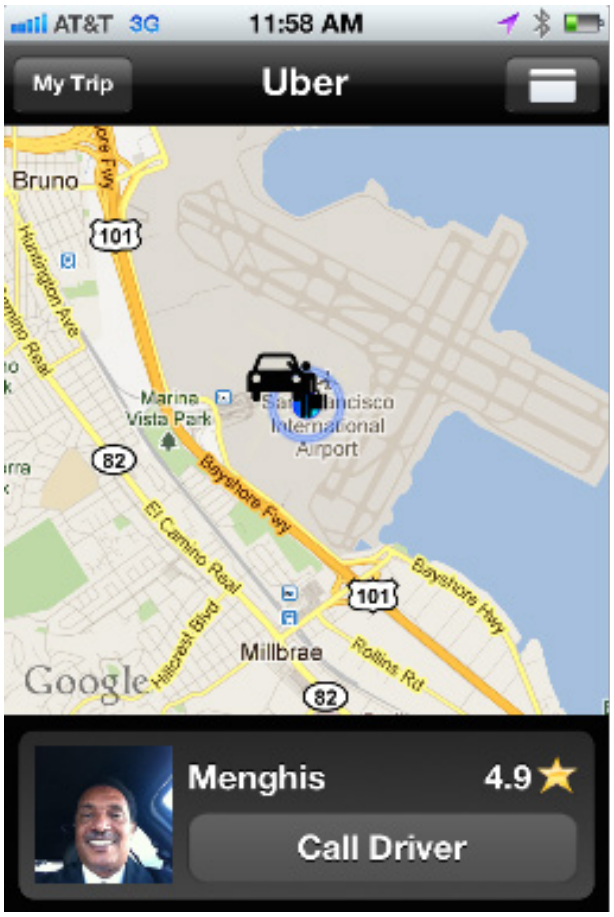


Patterntap by Zurb

patterntap.com

~5500 UI design patterns but no wireframe examples. It's a collection of specific designs and patterns of user interaction, a living classroom where designers learn what is working well on the Web and why. It allows designers to find inspiration and influence from the world around them since they don't design in a vacuum. Here, they can collect other designs besides their own - ones they love and others they hate - so they pick them apart to understand the techniques behind them, learning what works and doesn't work.

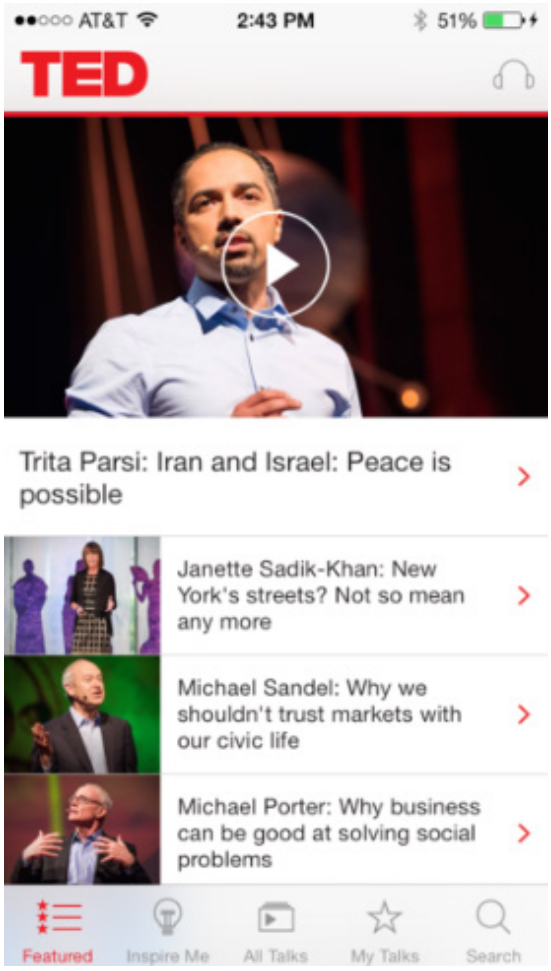


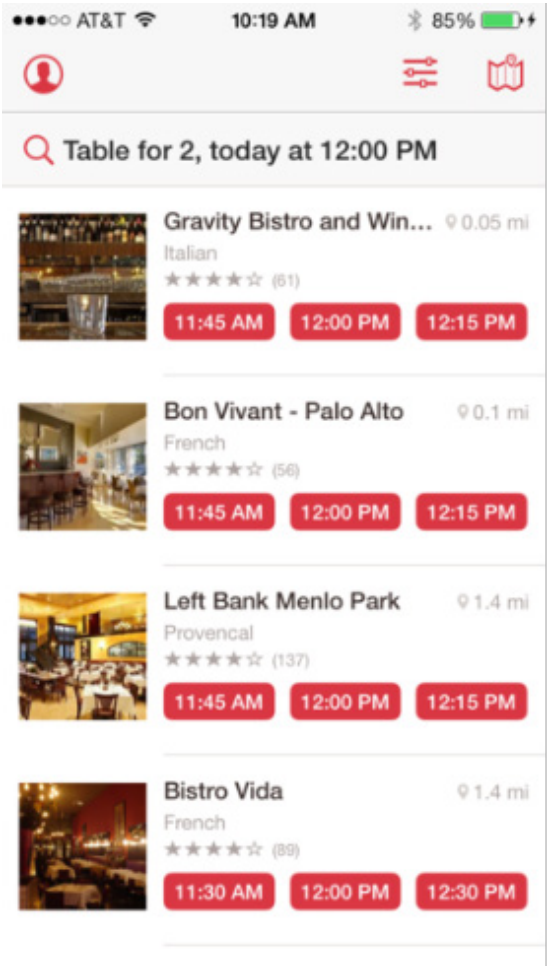


UX Archive

uxarchive.com

~2000 UI design patterns with a focus on flows, but no wireframes. It aims to help people create the best user experiences possible for their apps. In the creative process of shaping such an experience, designers need to explore what others have done to define what works, what doesn't and to observe patterns that you should or shouldn't integrate in your app. UX Archive helps designers in this process by laying out the most interesting user flows so you can compare them, build your point of view and be inspired.

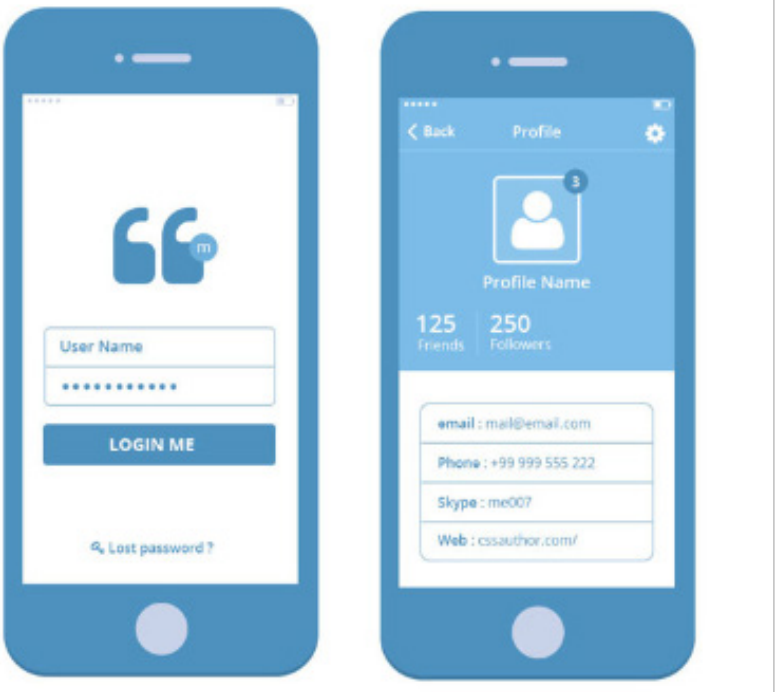


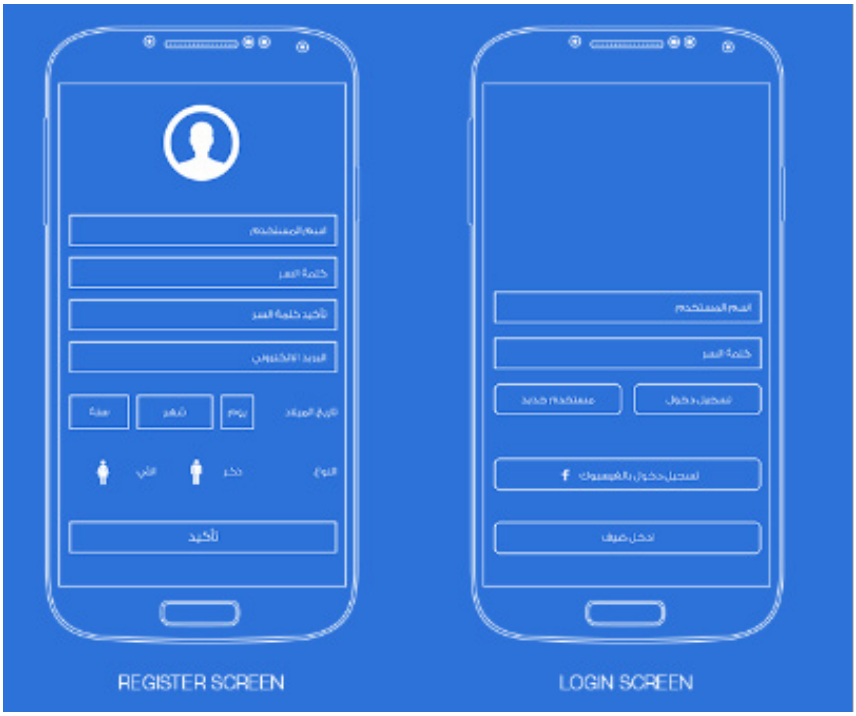


Behance

behance.net/search?search=wireframe

Countless searchable UI design patterns and wireframe examples, but not a focused library like some of the ones mentioned above. It's a leading online platform to showcase & discover creative work so you get a wide array of design work on it, mainly visual design. The creative world updates their work in one place to broadcast it widely and efficiently. Companies explore the work and access talent on a global scale.

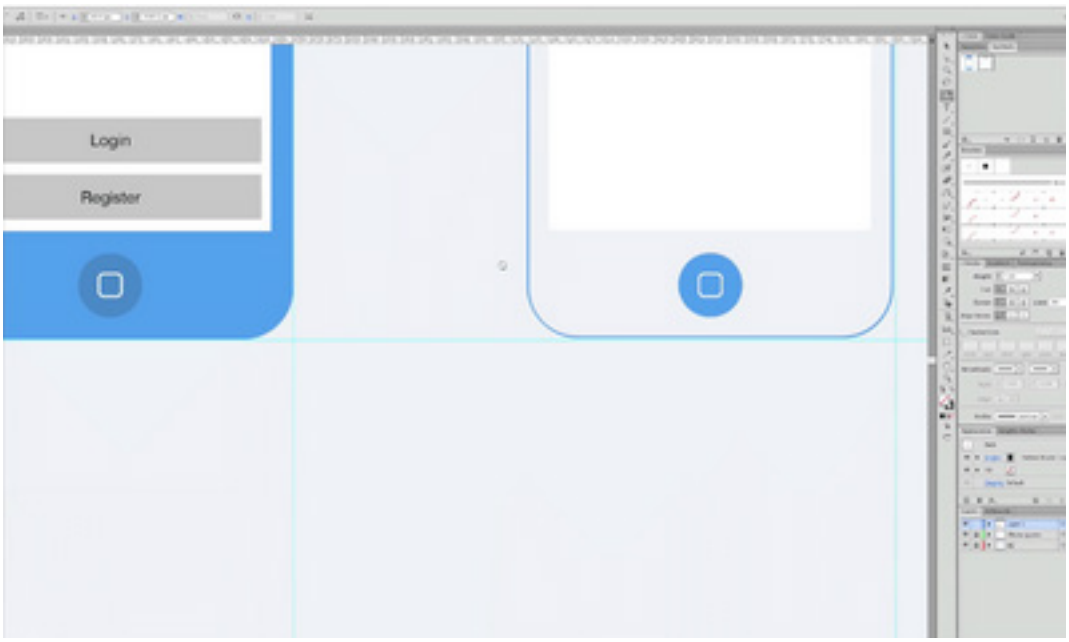




Dribbble

dribbble.com/search?q=wireframe

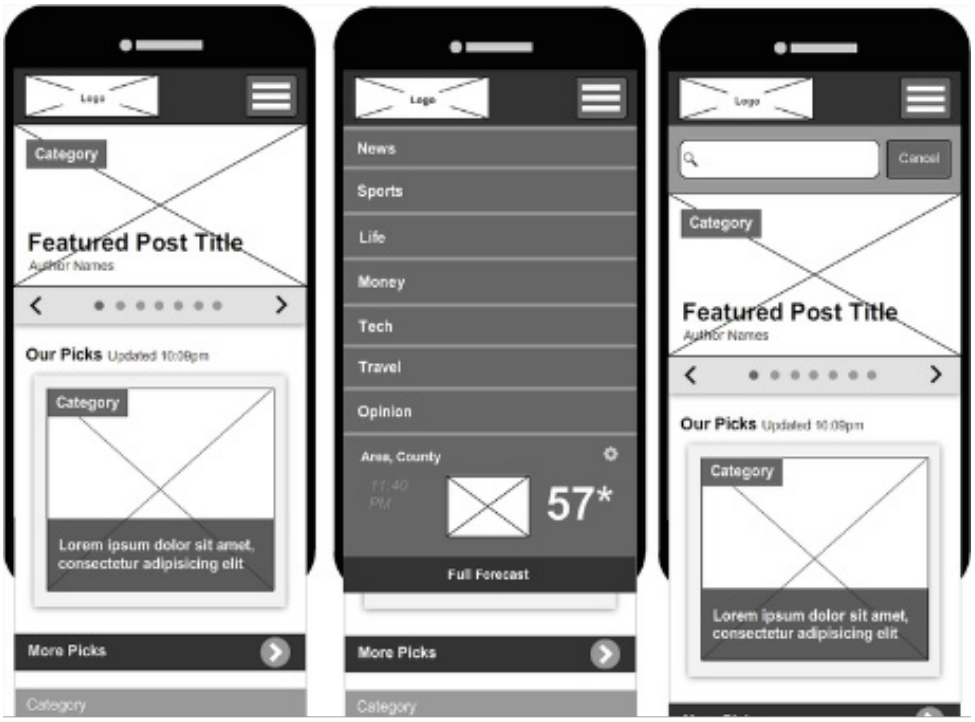
Countless searchable UI design patterns and wireframe examples, but not a focused library like some of the ones mentioned above. What are you working on? It's a community of designers answering that question each day. Web designers, graphic designers, illustrators, icon artists, typographers, logo designers, and other creative types share small screenshots (shots) that show their work, process, and current projects. Dribbble is a place to show and tell, promote, discover, and explore design.

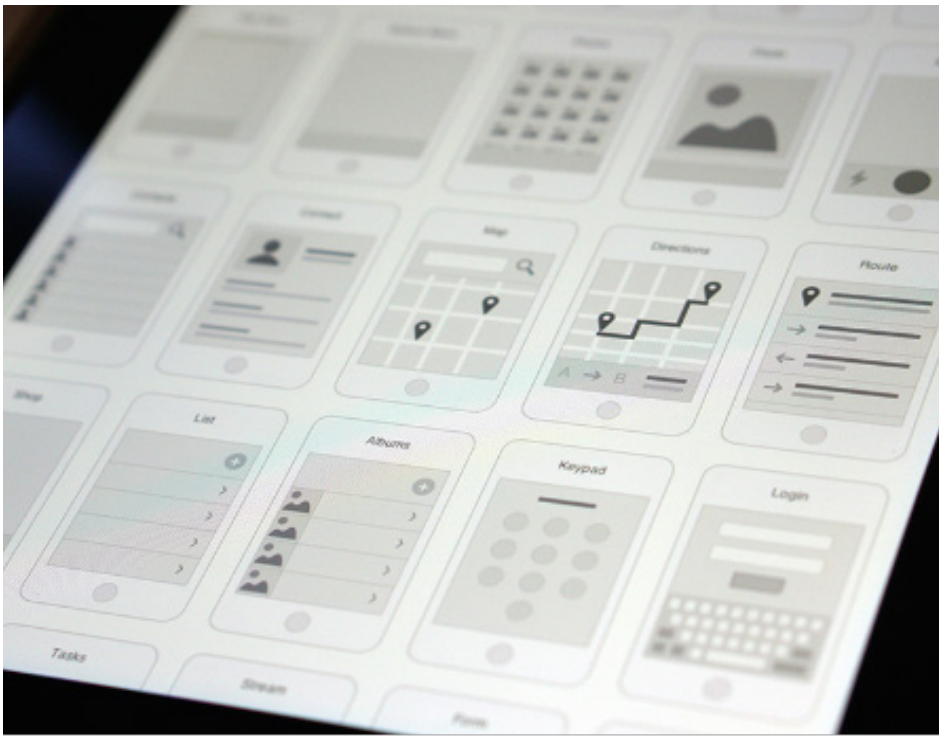


Forrst by Zurb

forrst.com/posts?search=wireframe

Countless searchable UI design patterns and wireframe examples, but not a focused library like some of the ones mentioned above. It's a community where designers and developers can share their work and get the feedback they need so you get a wide array of design work on it, mainly visual design. It's maintained by ZURB, a product design company that helps companies design better web sites, services and online products.

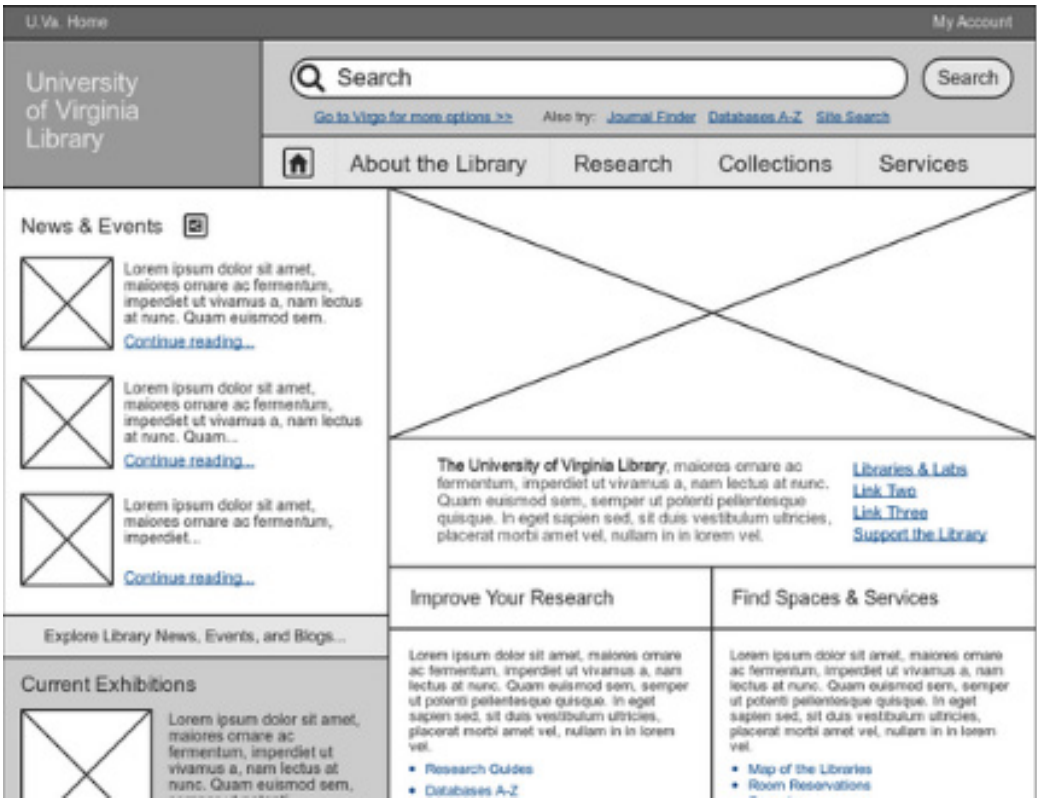




Google Image Search

google.com/#q=wireframes

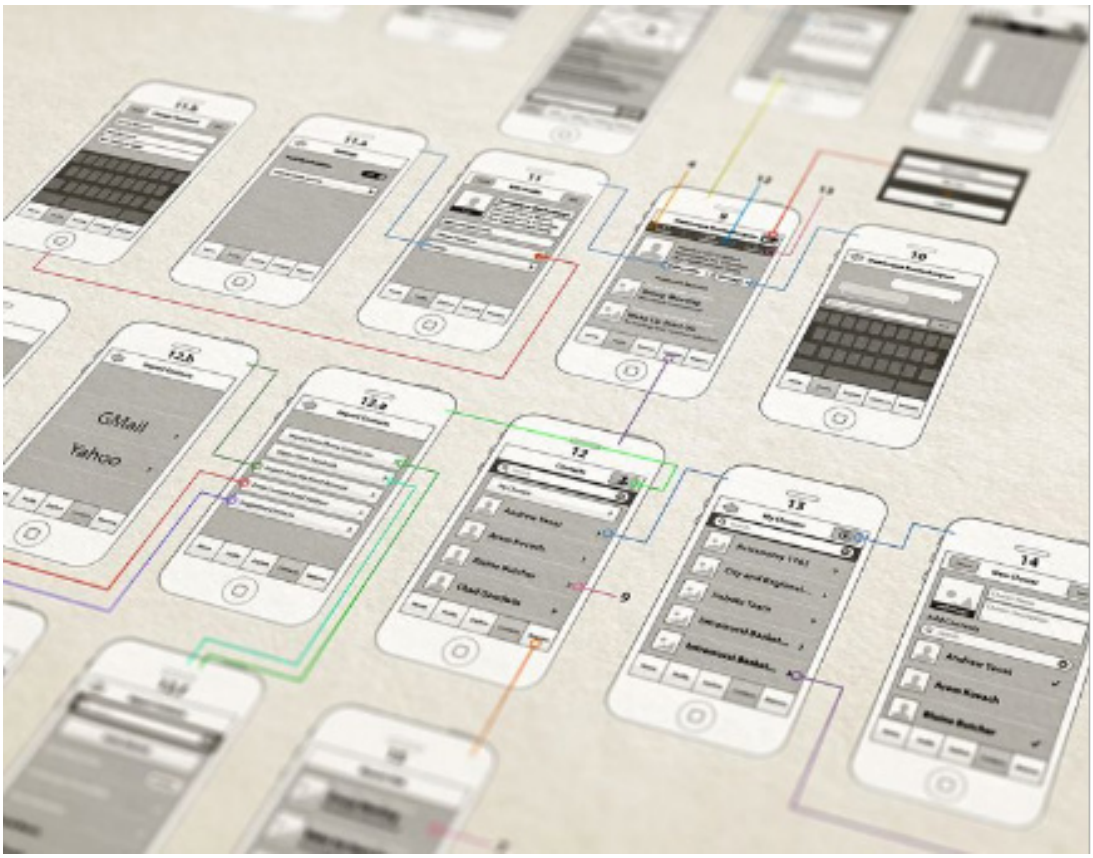
Countless wireframe examples and UI design patterns, although patterns are harder to search for.

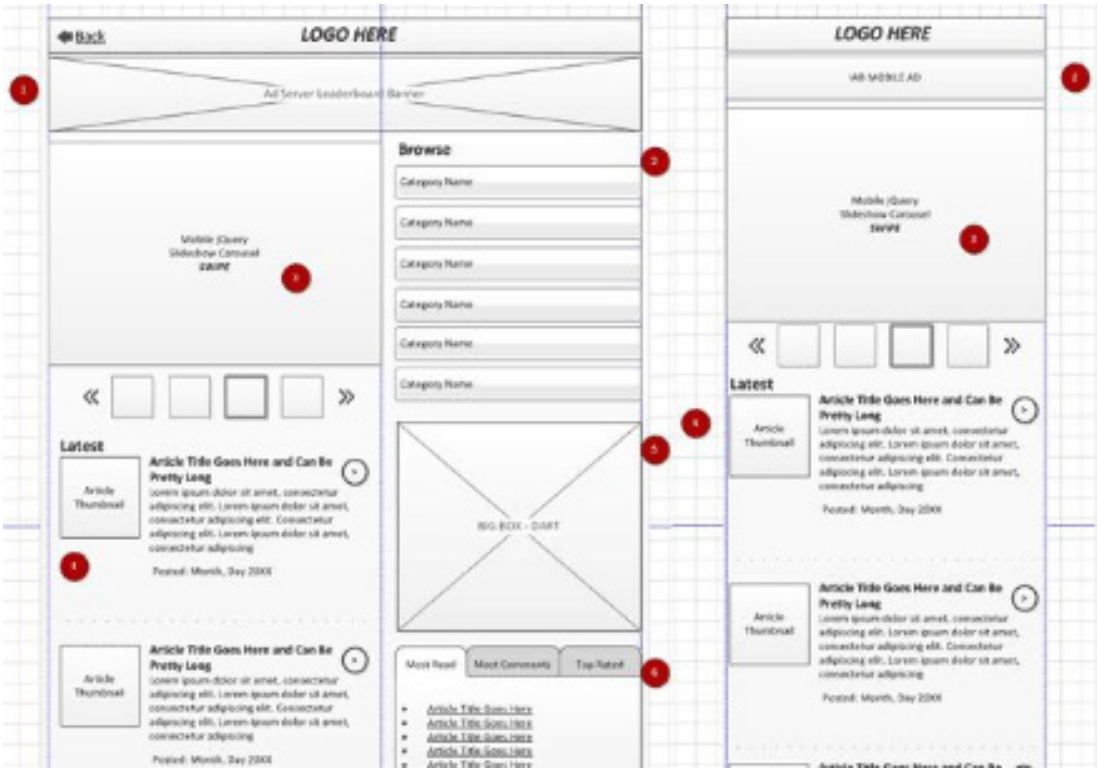


Pinterest

pinterest.com/search/pins/?q=wireframes

Countless wireframe examples and UI design patterns, although patterns are harder to search for. As expected, this is a more pleasant experience than Googling the same things.



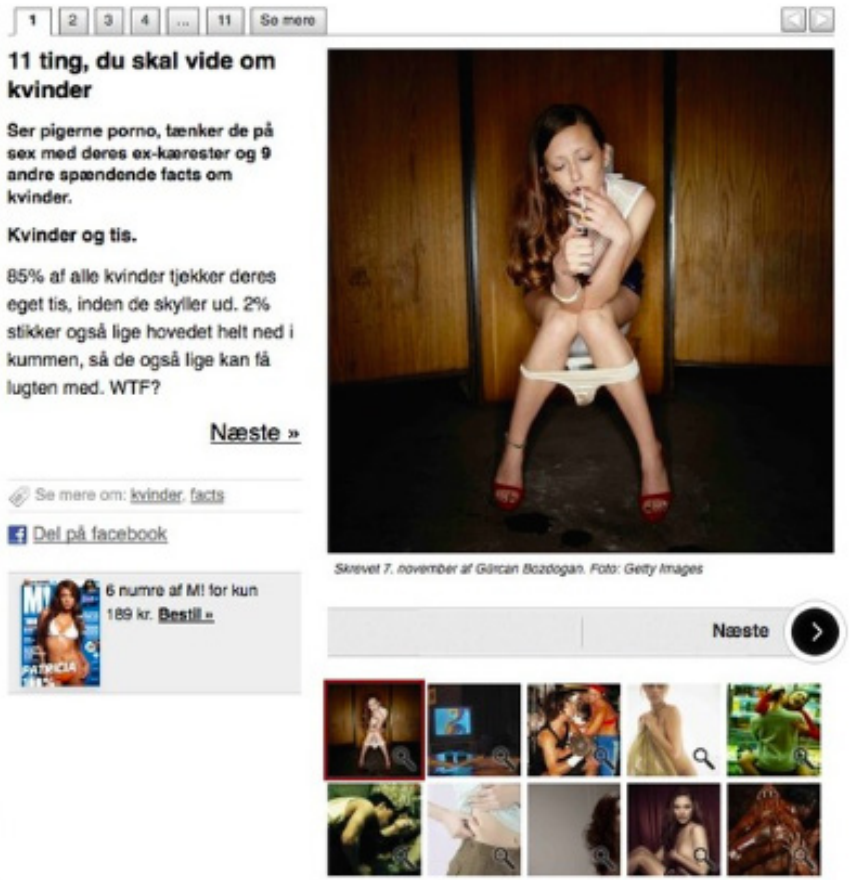


Web Only

UI Patterns

ui-patterns.com/patterns

Many UI design patterns but no wireframes, and hasn't been updated in a year or so. Unlike other design pattern libraries, it actually has descriptions of patterns, when they're used, what problems they're solving, and tips when designing these patterns. See this example of a gallery.

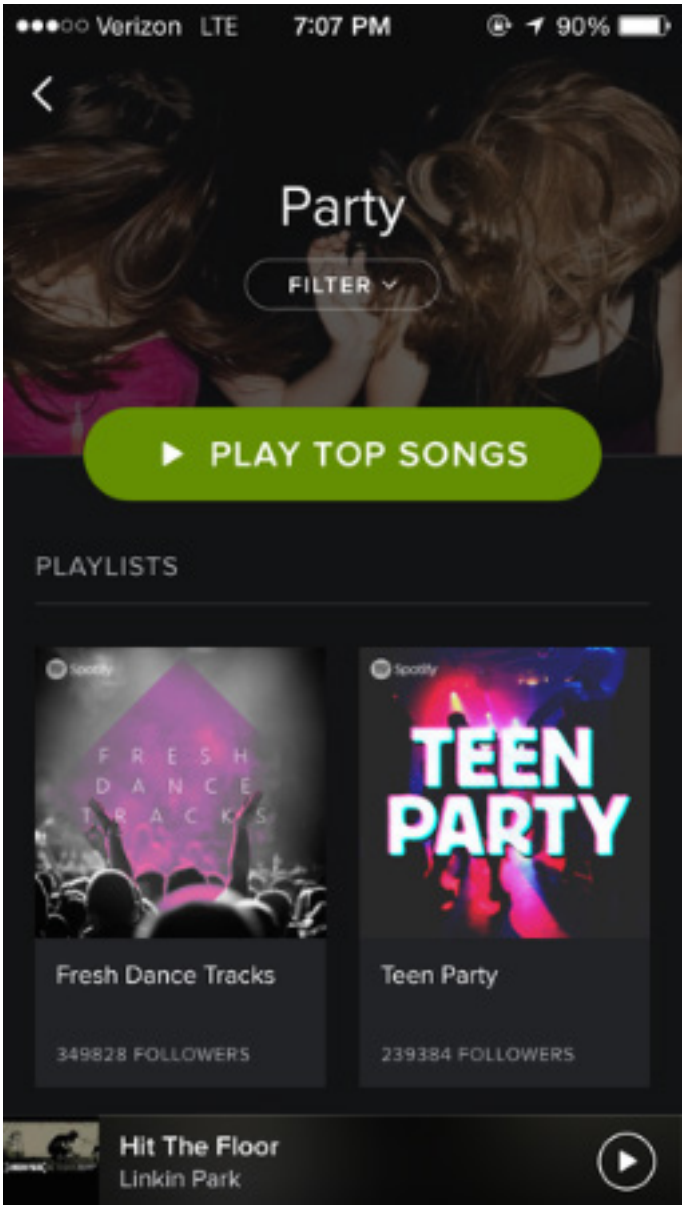


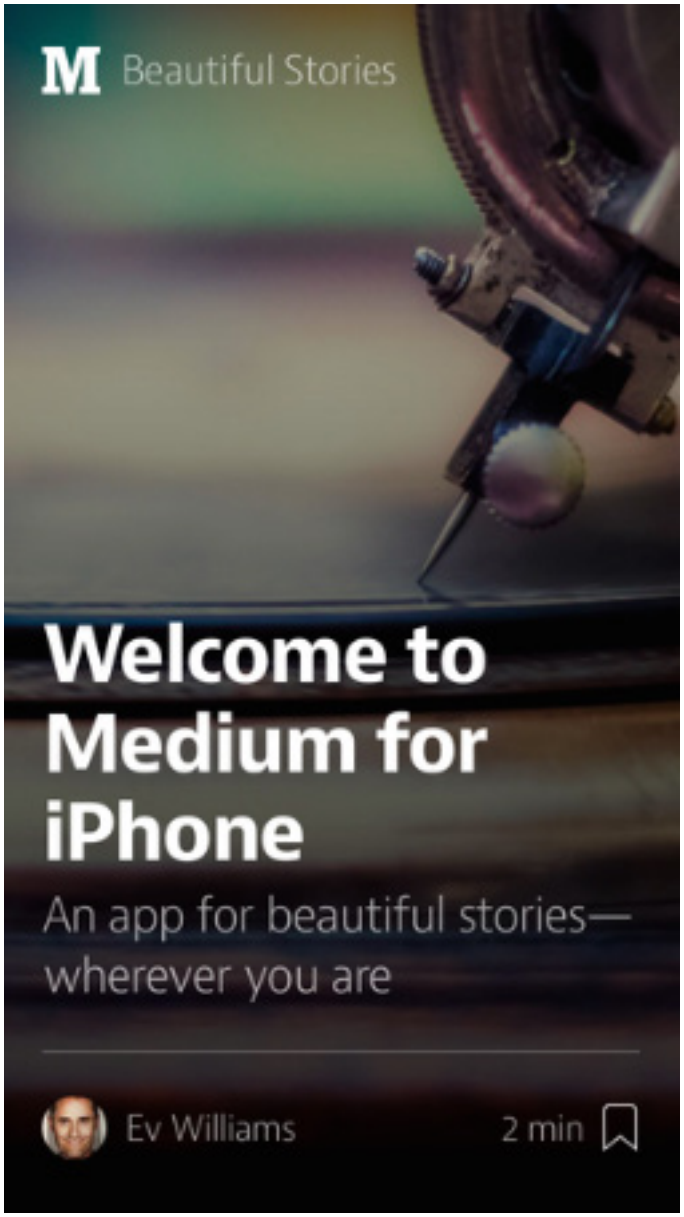
Mobile Only

Pttrns

pttrns.com

~2500 UI design patterns but no wireframe examples. It's a curated library of iPhone and iPad user interface patterns. All patterns has been carefully chosen before they were added to maintain excellent content quality. The site is currently doing almost 2 millions page views per month.

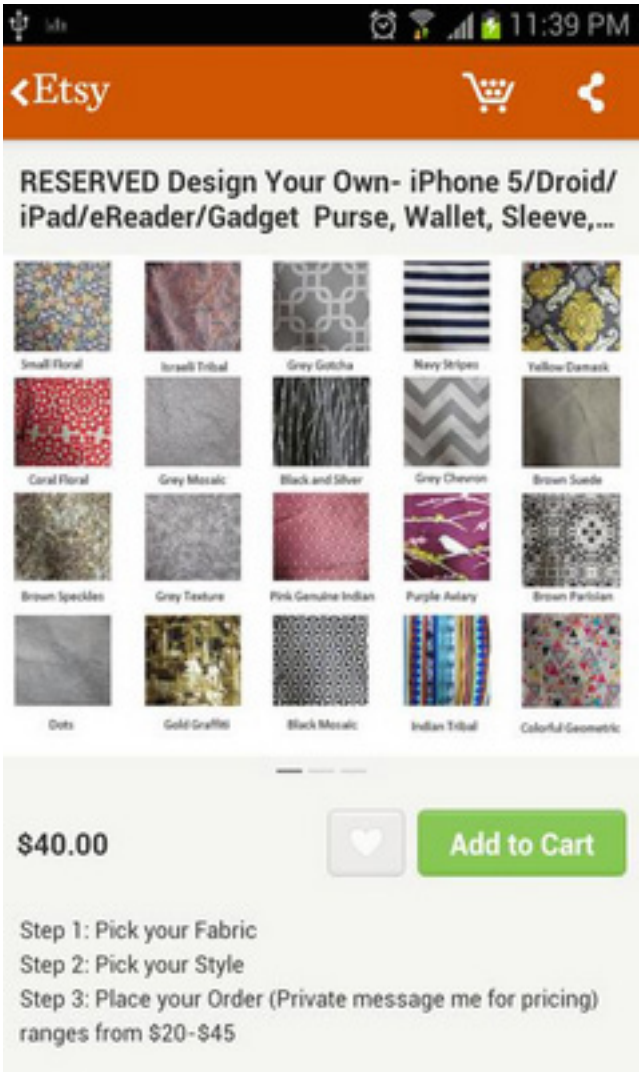


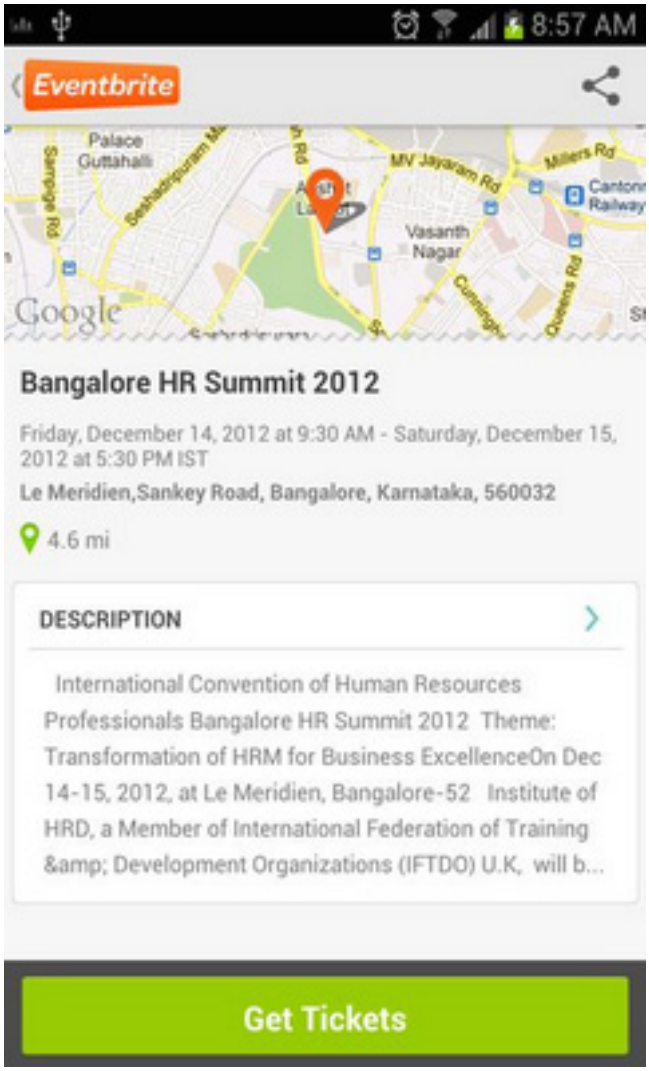


Android App Patterns

android-app-patterns.com

~1550 UI design patterns but no wireframes. Simply put, it's design inspiration for your Android app.

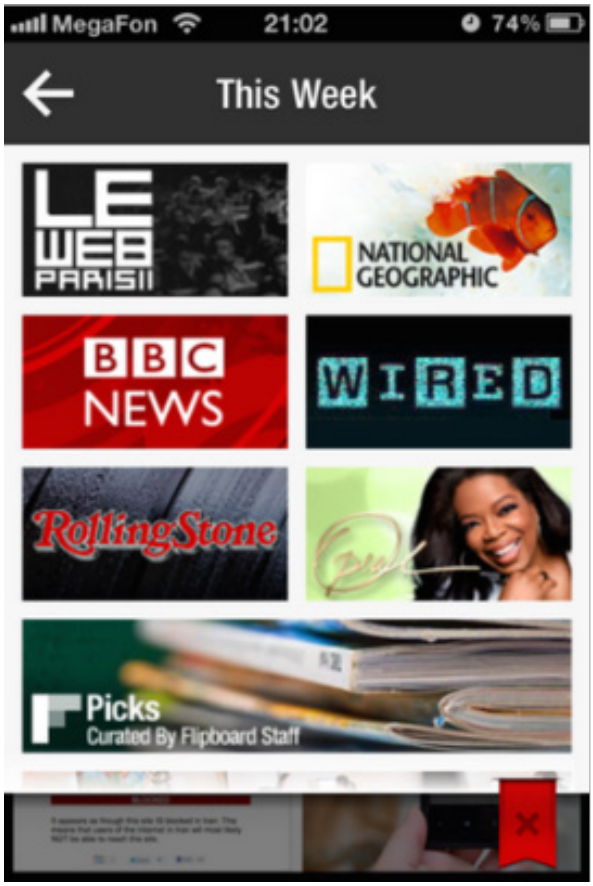


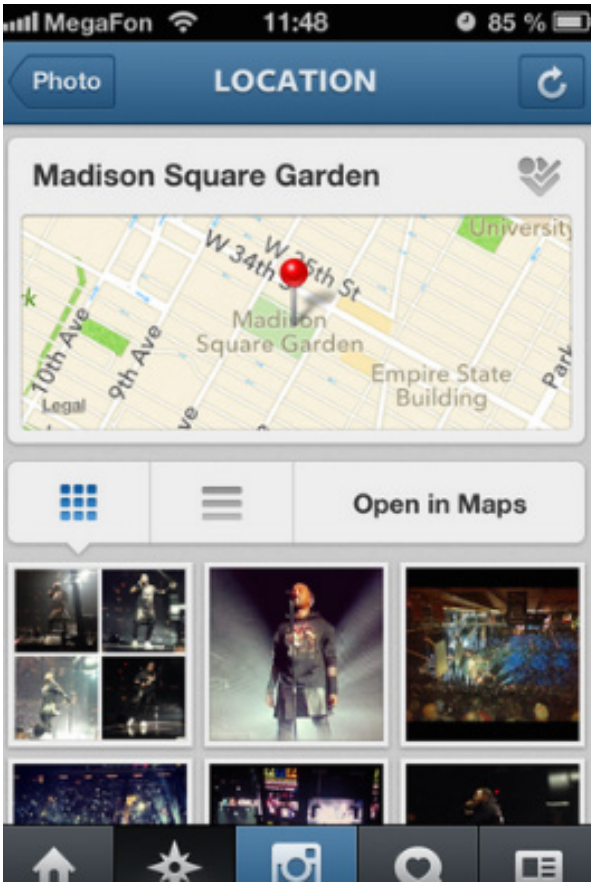


Inspired UI

inspired-ui.com

Countless UI design patterns but no wireframe examples for Android, iPad, and iPhone.

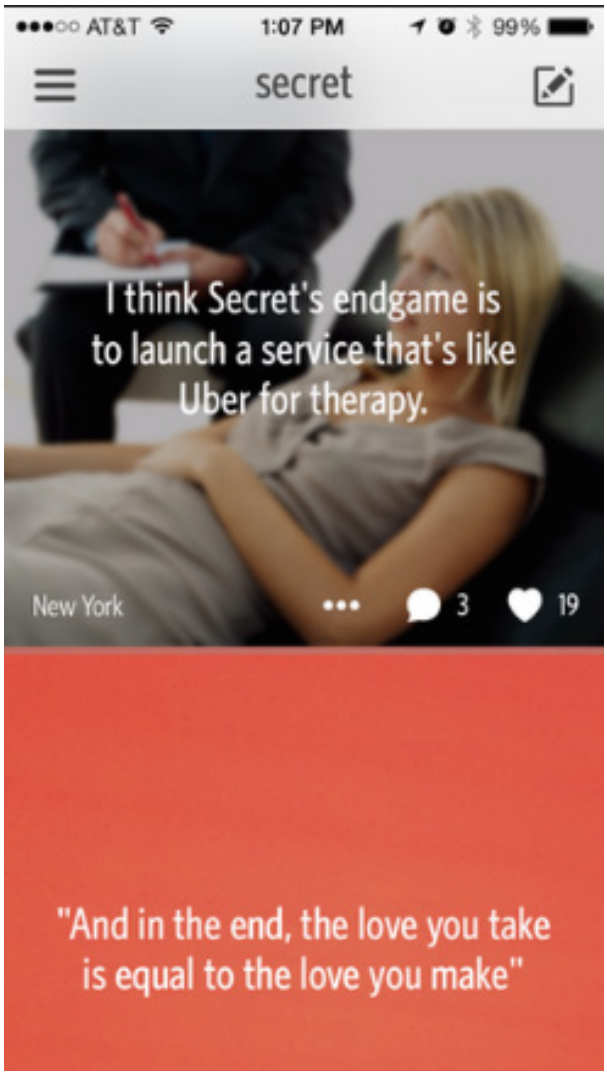


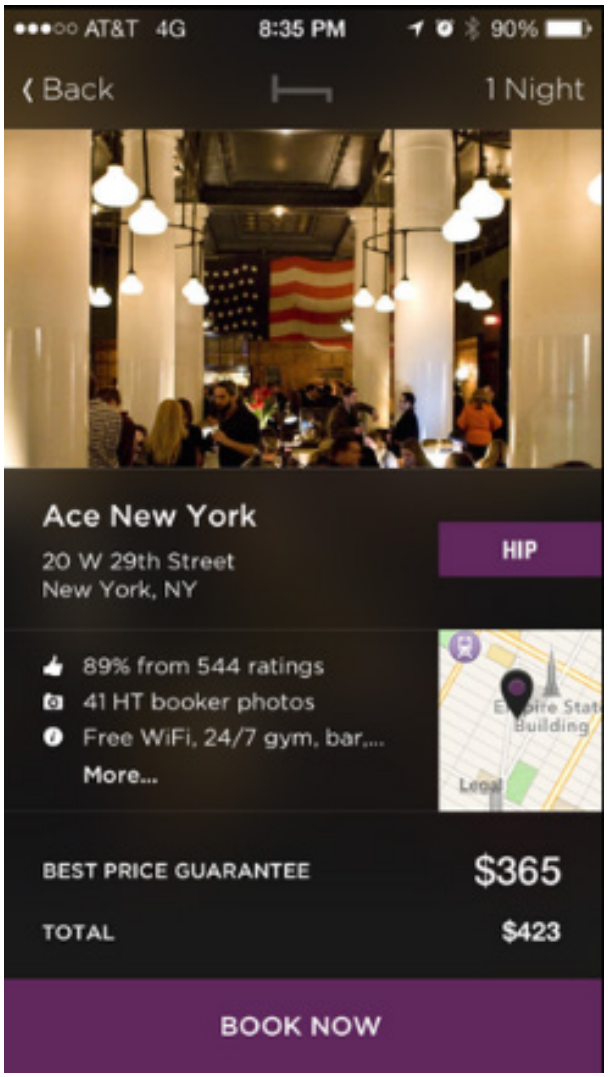


Mobile Patterns

mobile-patterns.com

Countless UI design patterns but no wireframes for iPhone and Android.

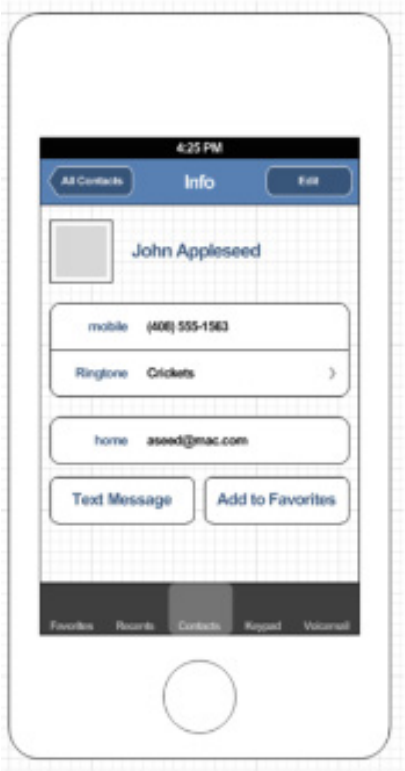


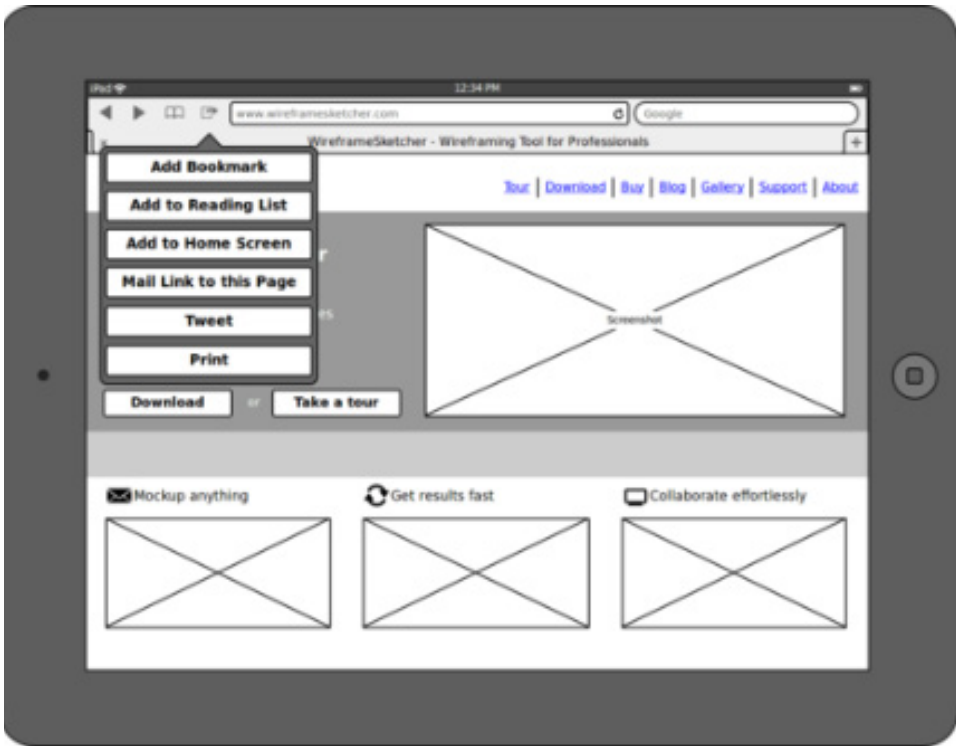


MOObileFrames

moobileframes.tumblr.com

A showcase of wireframe examples for mobile apps.





You've Got Options!

Take what you need and leave the rest.

If you value being able to take your wireframe examples and design pattern inspirations from top companies and directly adapting them to your needs, UXPorn by UXPin is probably your best option since you can import them directly. If you mainly need a broad array of inspirations, you've obviously got a lot of other options - I personally love Pttrns and UX Archive.

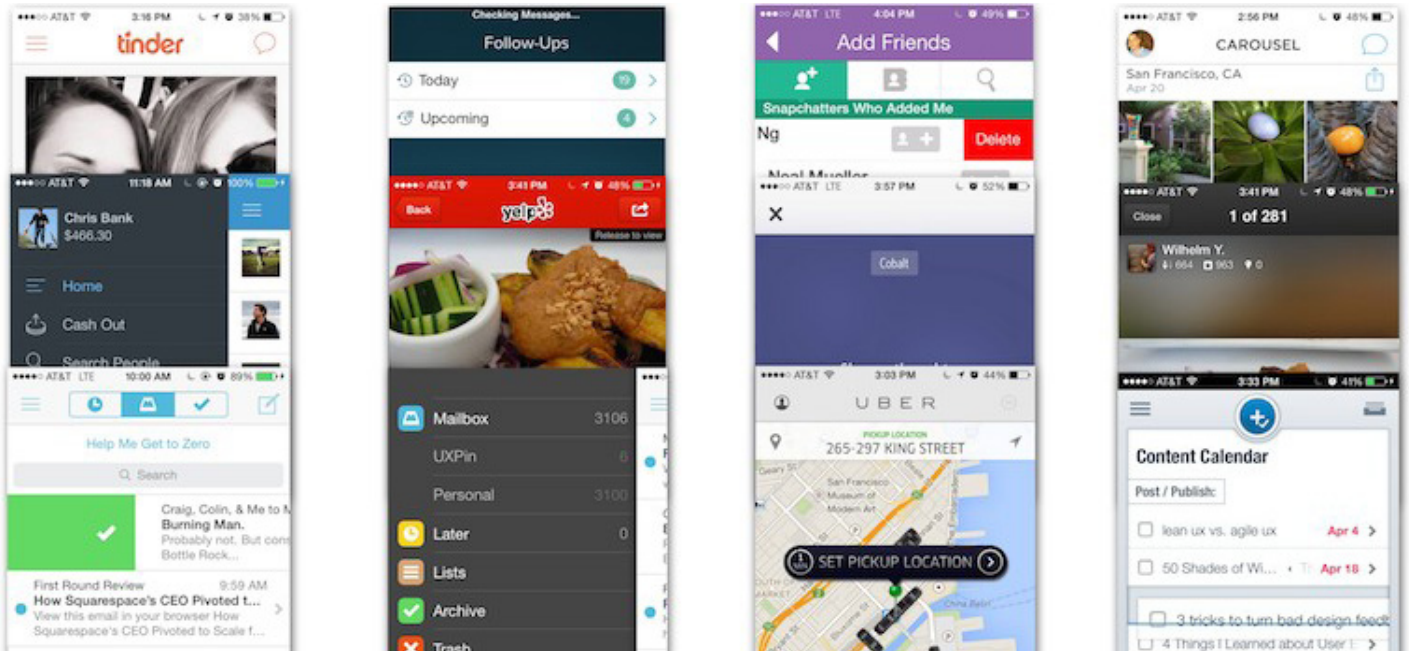
Regardless of which resources you save for later, keep in mind that seeing is not synonymous with understanding. You can browse these galleries all day, and still never take anything valuable away from your perusal. In the next chapter, Fresh UI

Design Patterns, I detail many of the best UI design patterns that have cropped up in modern web sites and mobile apps recently, and examples of how they're used in popular web and mobile products. Some of these patterns are also covered in UXPin's free Web Design Trends and Mobile Design Trends e-books.

7

FRESH UI DESIGN PATTERNS

Web and Mobile Product Design Trends I've Noticed In 2013 and 2014



Picasso once said (and Steve Jobs echoed), „Good artists borrow, great artists steal." This may be one of the most misunderstood and misused creative phrases of all time, but also the most important in a time when product design and development trends have exploded.

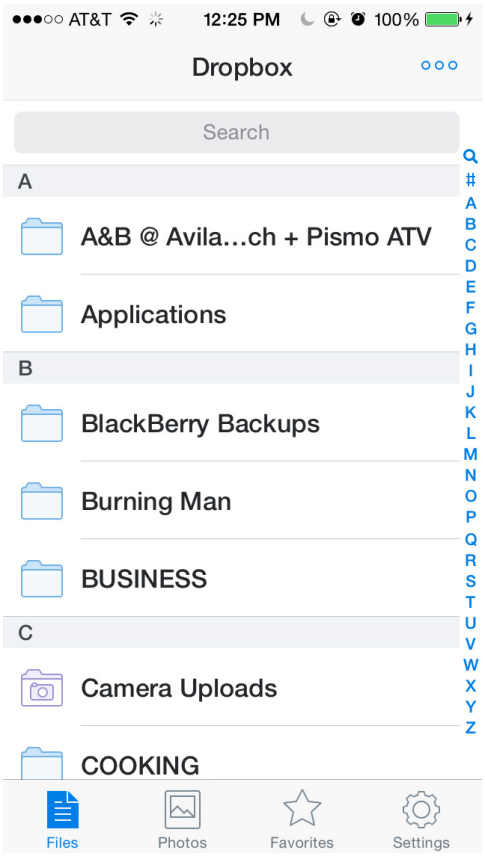
The controversy of this quote comes from its simplicity and, therefore, openness to interpretation. Out of context, it fails to clarify the difference between copying and internalizing; originality and innovativeness; aping and assimilating. It's not an excuse to be lazy. Rather, encouragement to learn from the work of others, build upon it, and own the products you've designed and built to solve real, focused problems of people - those for whom you ultimately design.

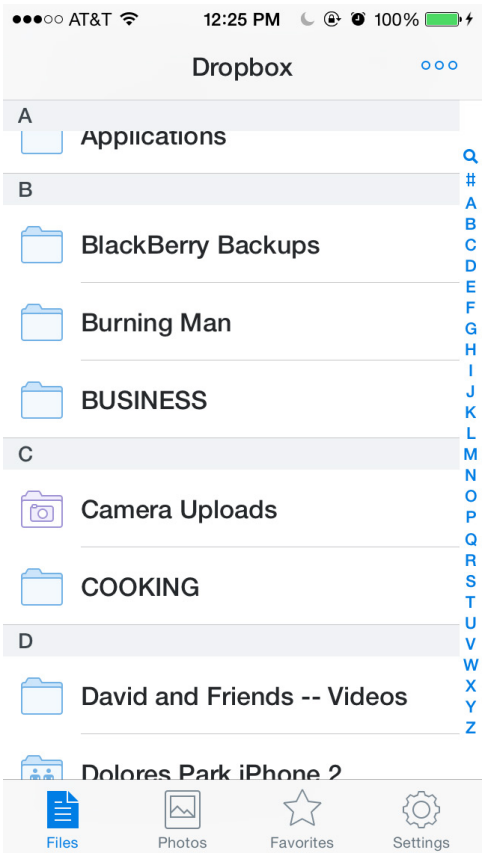
The team at UXPin recently dug through their favorite UI design pattern sources to identify the recurring solutions that companies were coming up with to address common design problems as web and mobile applications continue to evolve at a rapid pace. I've seen a beautiful assortment of ways that products use swipe, click, hover, and hold triggers in web and mobile apps to bring users to the most relevant secondary and tertiary information or next step in a major user flow without cluttering the interface, or even cleaning it up.

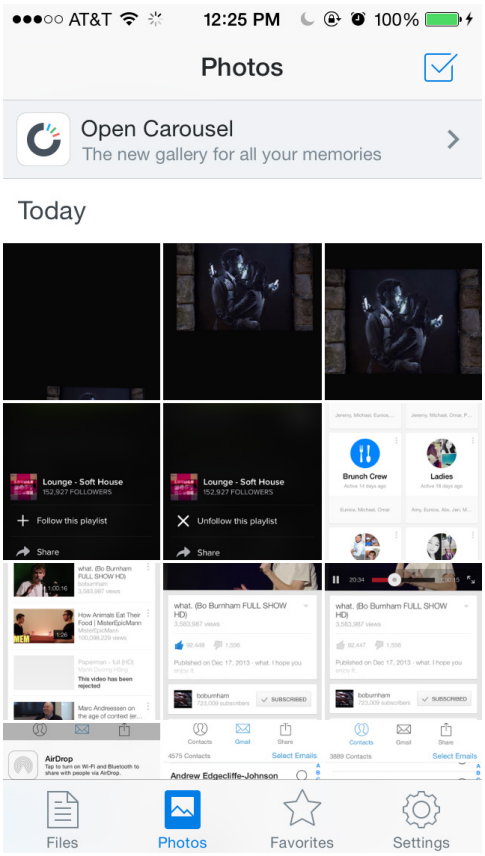
Below, I've shared a few of these standard reference points with you to help you as you brainstorm, sketch, wireframe, and prototype. I've even included wireframe examples to help you visualize these patterns at different stages in the design process. While most of these UI design patterns specifically address mobile problems, many apply to web as well.

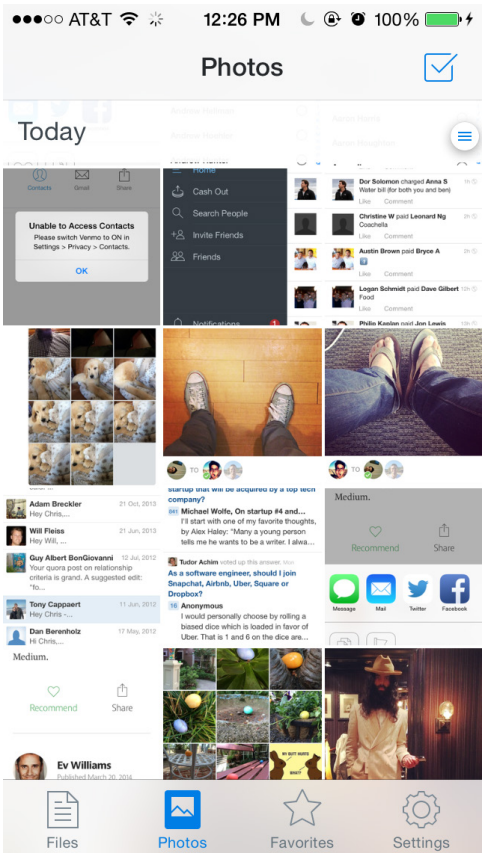
Design Patterns & Wireframe Examples

Dropbox and Carousel and almost every other application have **"sticky" fixed navigation**, a universal mobile UI design pattern. As a user, you don't have to scroll all the way back up to find your footing. Many applications also have allow you to scroll all the way up to the top by tapping the top of your screen on mobile - there's usually no visual indicator that this functionality exists though (unlike Pinterest's "scroll to top" tab). And as a website developer, you don't have to throw in links all over the place to make sure people have all the relevant links they need. Likewise, now many of the traditionally fixed links can now be added directly to the content and sections of any page or view, not at the top-level navigation. I've included a wireframe example of this design pattern using UXPin.


AT&T 12:25 PM 100%
Dropbox
Search
A
A&B @ Avila...ch + Pismo ATV
Applications
B
BlackBerry Backups
Burning Man
BUSINESS
C
Camera Uploads
COOKING
Files
Photos
Favorites
Settings

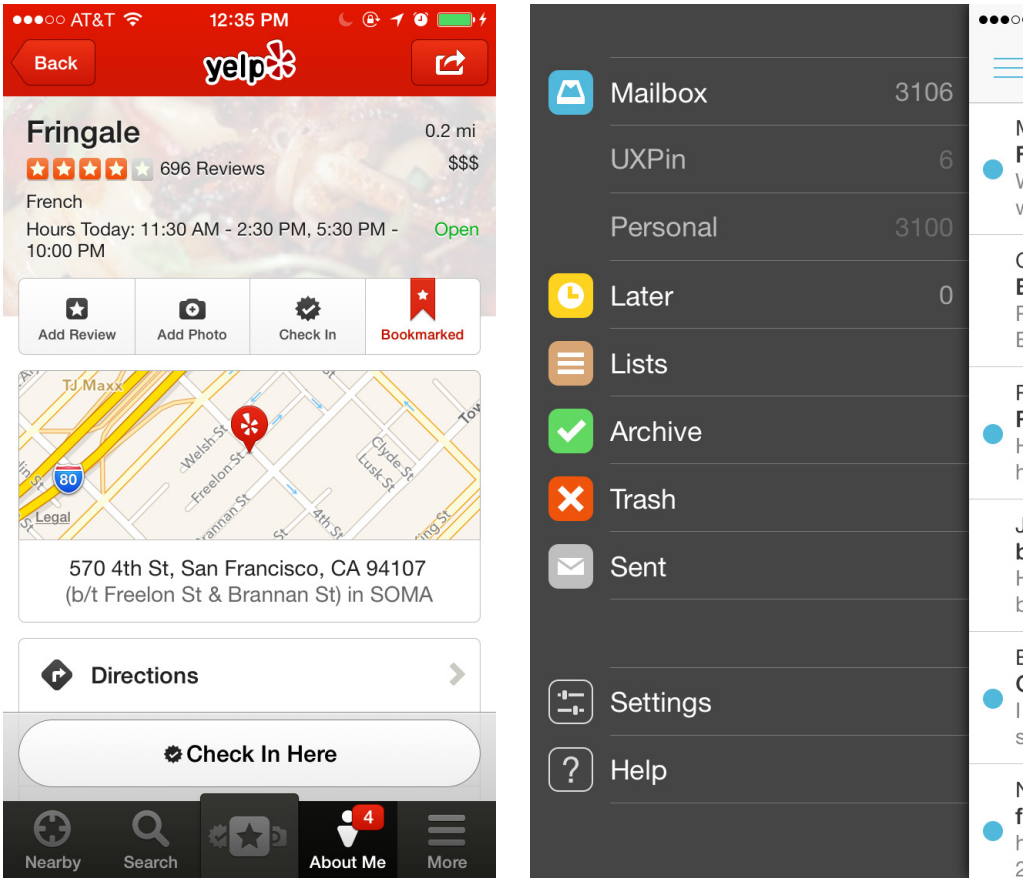

AT&T 12:25 PM 100%
Dropbox
A
Applications
B
BlackBerry Backups
Burning Man
BUSINESS
C
Camera Uploads
COOKING
D
David and Friends -- Videos
Files
Photos
Favorites
Settings


AT&T 12:25 PM 100%
Photos
Open Carousel
The new gallery for all your memories
Today
Files
Photos
Favorites
Settings

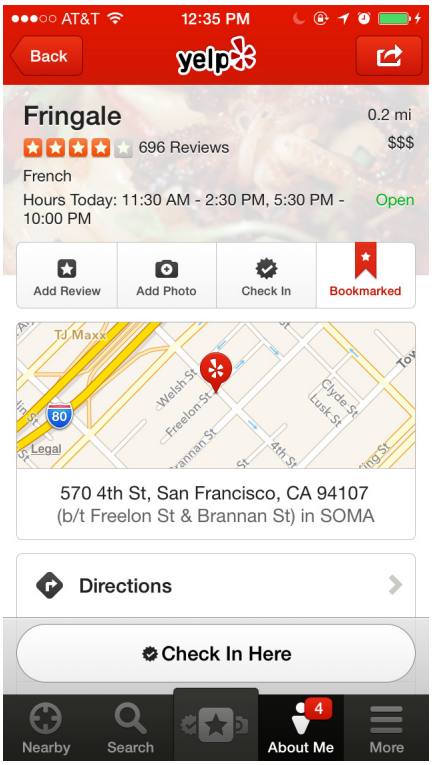

AT&T 12:26 PM 100%
Photos
Today
Files
Photos
Favorites
Settings

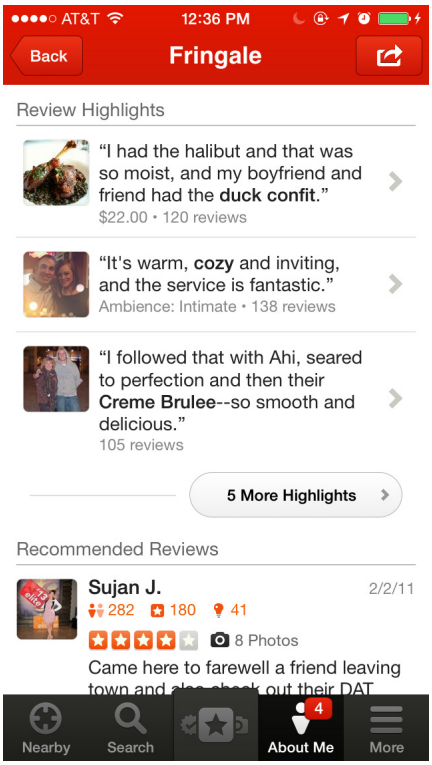
Facebook and Mailbox and many more apps have **vertical (non-standard) navigation** on both web and mobile. This will help you find sections with an application in addition to finding all the content and data you're looking for about the product, company, etc. on a single page through a fluid navigation that jumps you to different parts of the site. This is also a great example of a **drawer,** which is a very popular UI design pattern on mobile.

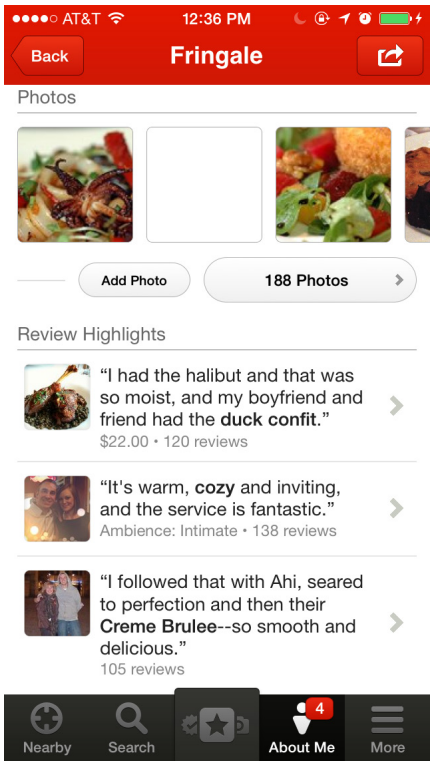
Similar to toggling, drawers are life-saving UI design pattern trends in light of mobile screen constraints and user's need for speed. Since each drawer and slide-out is a separate "layer" in an application, it's not as constrained and, therefore, I've seen even greater variety in implementations, including many terrible ones - but I won't show you those ones.

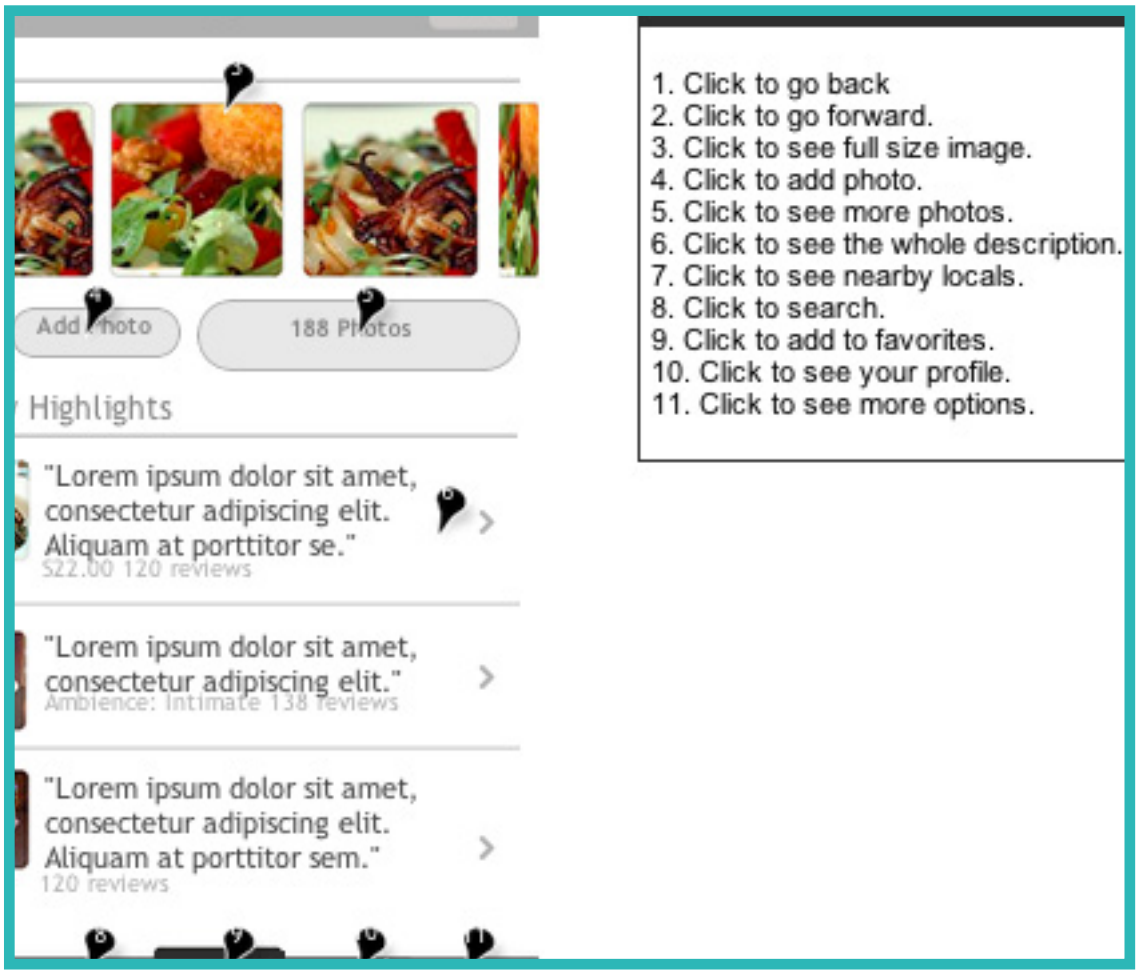


Yelp is a phenomenal example of providing **links to everything** in their application. While they provide a more explicit UI design pattern, more content-heavy applications like Flipboard can actually get confusing with the endless ways you can swipe, tap, x-out, undo, and go-back as you navigate through it's digital magazine. We've included a wireframe example of this design pattern below using UXPin.



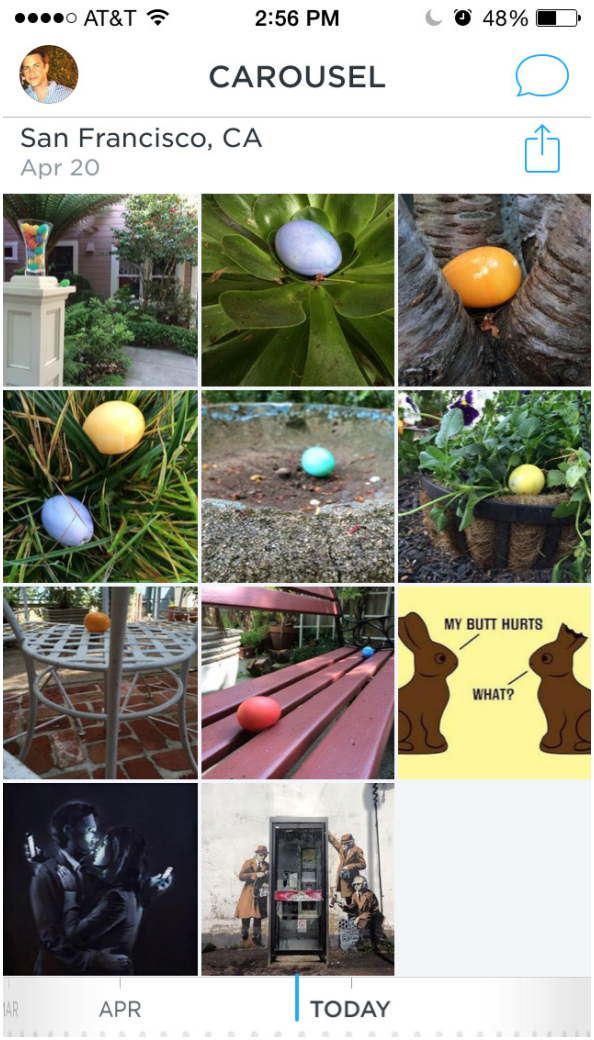




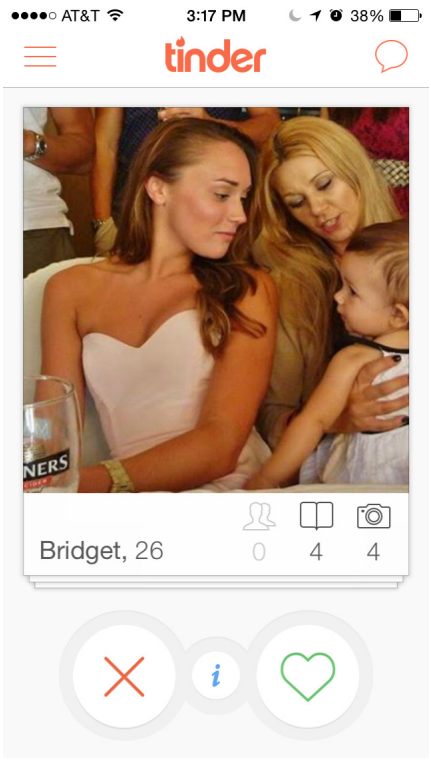


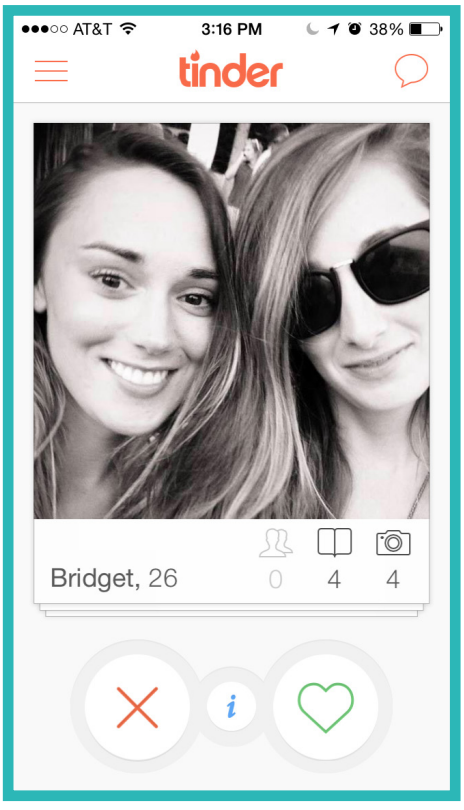
Click to use this wireframe in UXPin

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



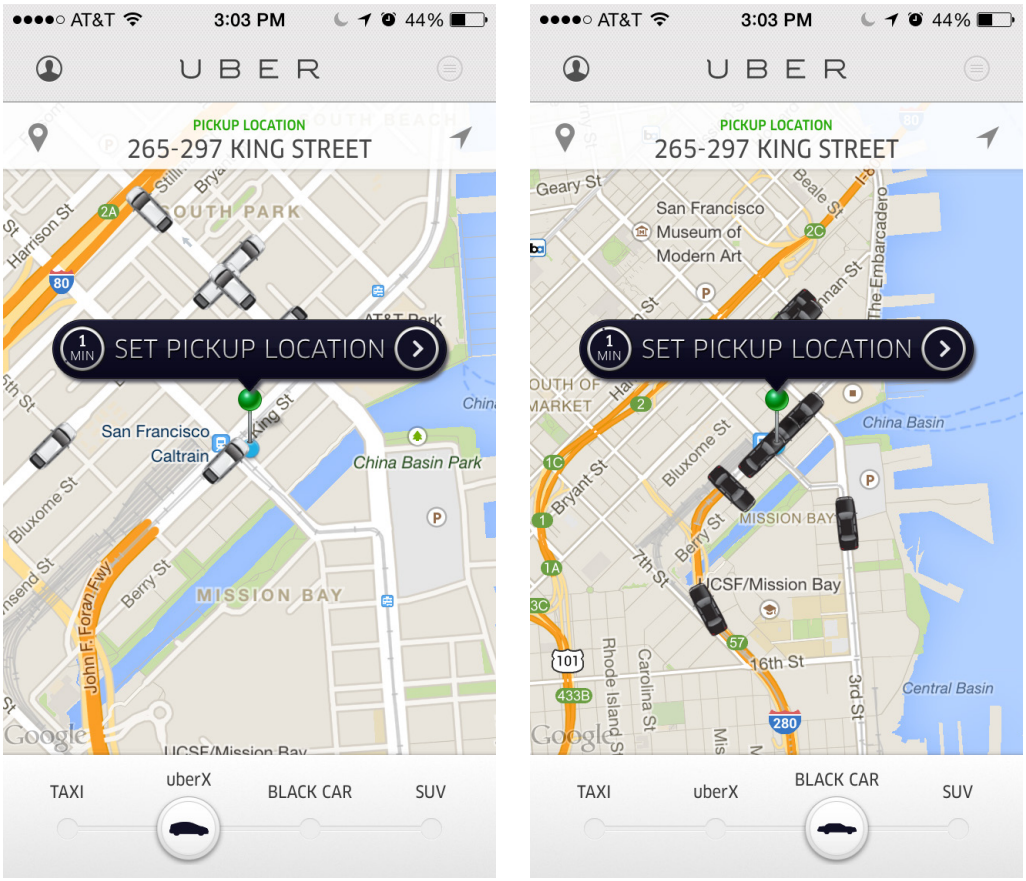
Tinder has made it's **content seamlessly responsive.** This UI design pattern lets you **toggle** between 2 states of a user's profile simply by clicking on the main picture in each view. But they go one step further. If you swipe through pictures in the detailed view of a user profile then click on the picture to go back to to the basic view, it stays on the picture you clicked on. This creates an extremely fluid and intuitive user experience and flow. In all fairness, OKCupid lets you do this too.



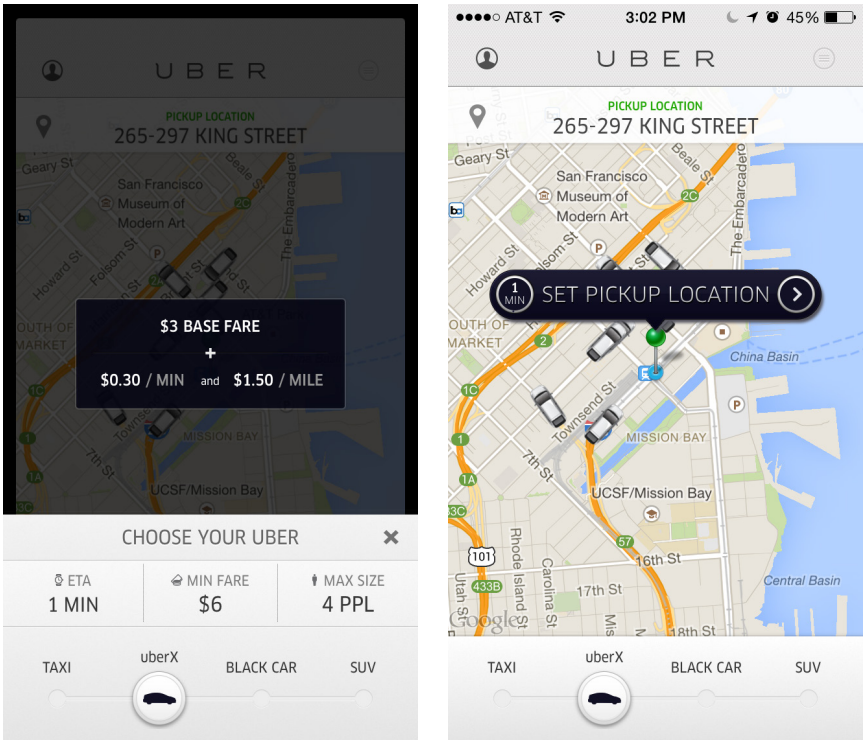


Click to use this wireframe in UXPin

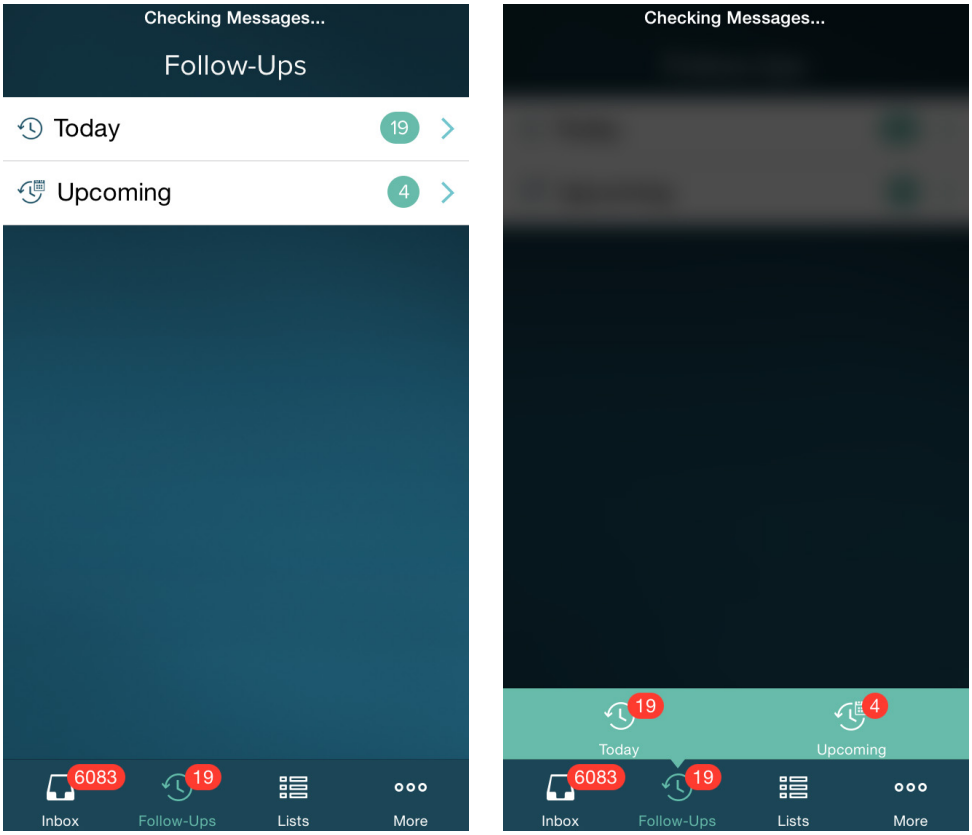
Uber lets you toggle between four types of ride services seamlessly by **dragging a slider side-to-side**. In this UI design pattern, they even zoom in and out to give you a similar level of density of cars nearby so you can see an acceptable number of options automatically.



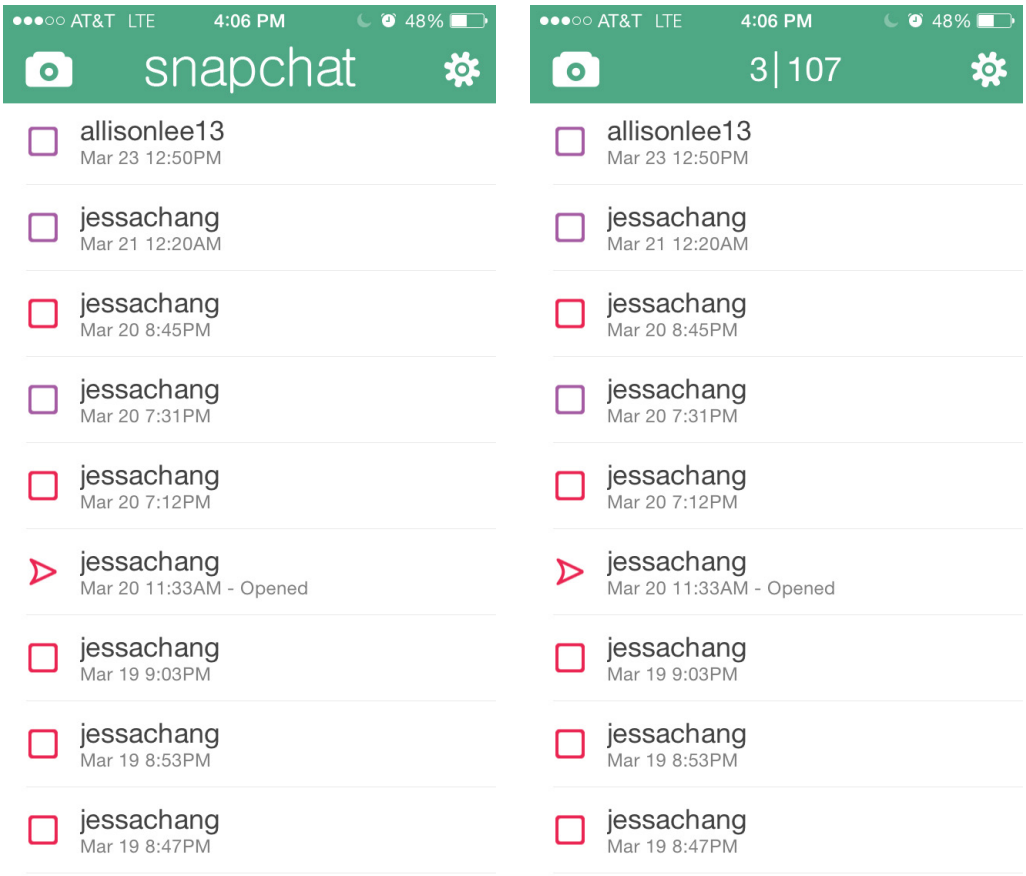
Uber also let's you toggle between booking a ride and seeing the fare estimation by **tapping the slider button** once you've chosen which ride type you want. This is a simple yet important UI design pattern that makes me smile every time I'm doing five things while trying to get a ride somewhere, but want to make sure Uber isn't ripping me off with surge pricing.



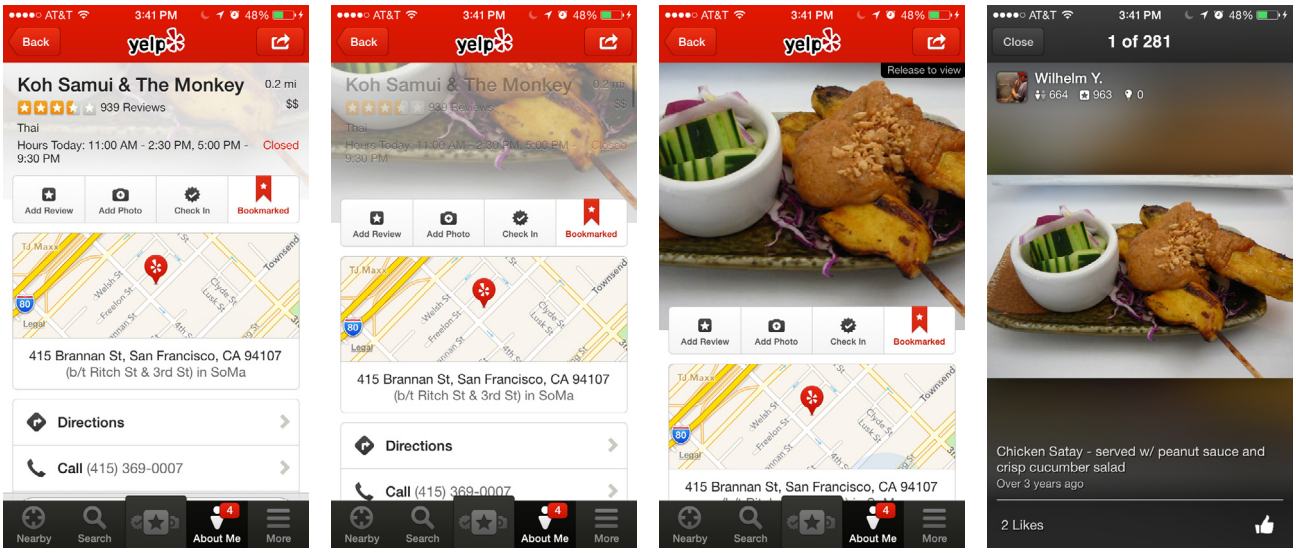
RelateIQ let's you **hold down main menu items** to see submenus for faster navigation to views. It's one of the most complex enterprise mobile apps on the market today so they're going to be pulling out a lot of new and existing mobile UI design patterns to keep it a quick and clean experience without sacrificing the power of their web product.



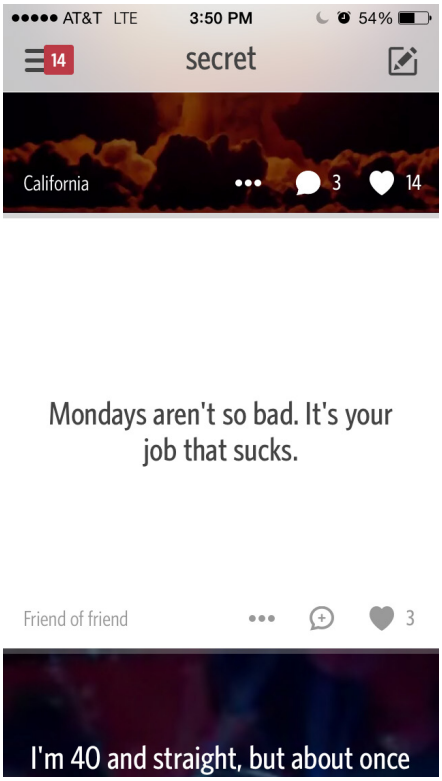
Snapchat let's you see **hidden information** - the number of messages received and unread - by clicking on the snapchat header. Simple UI design pattern, right?

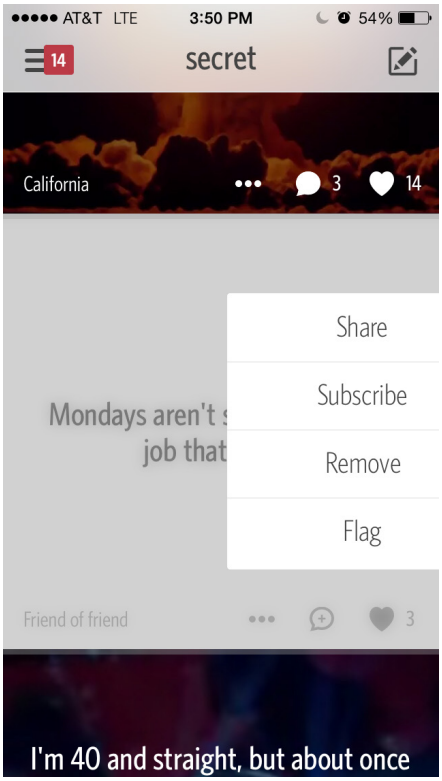


Yelp let's you go between listing details and the photo gallery when you drag downward to further expose the photo hidden behind a semi-transparent listing header. The use of **semi-transparency and responsive content** creates a wonderful experience here. It's probably one of the more advanced but least known UI design patterns out there. Hope everyone uses it more now!

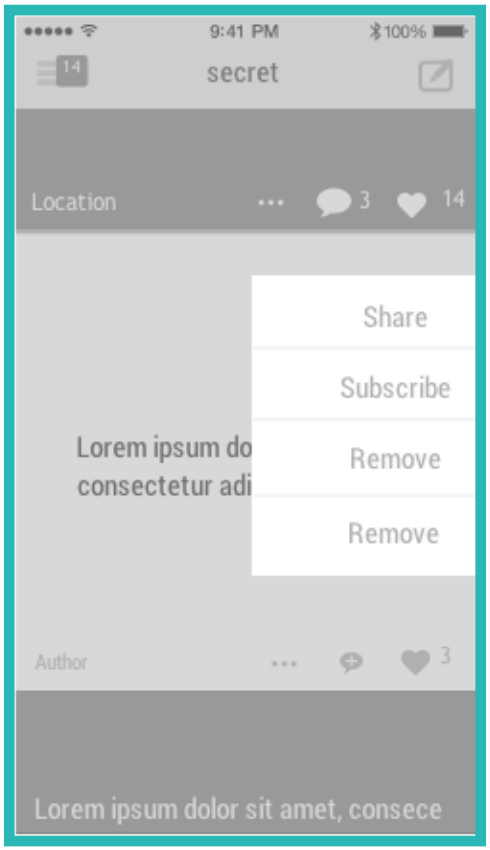


Secret only lets you take certain actions once you **discover how to access the menu**. Swipe right and you'll expose an action menu. This UI design pattern takes minimalism using **drawers and slideouts** to a new level. I've included a wireframe example of this design pattern below using UXPin.

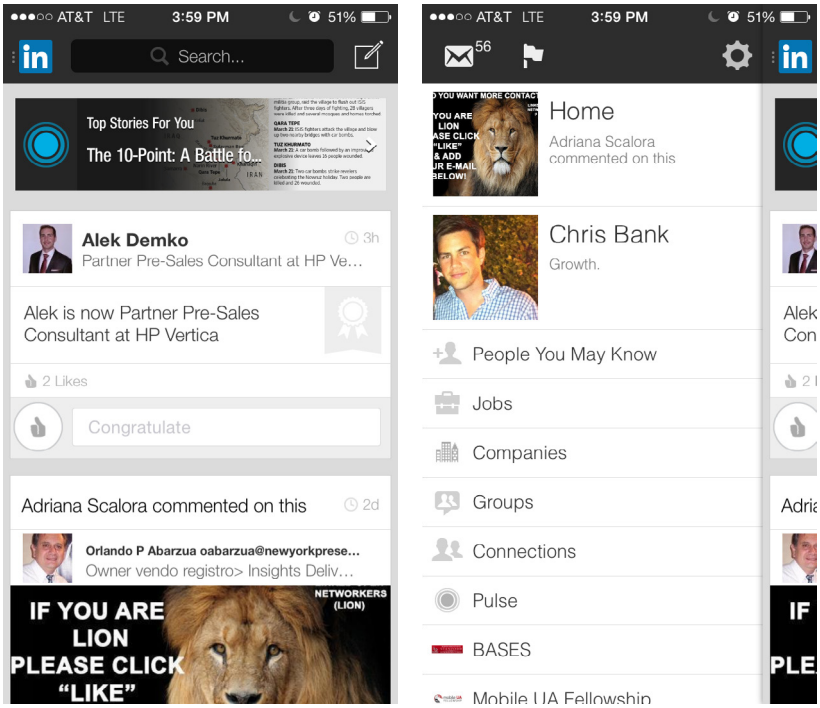


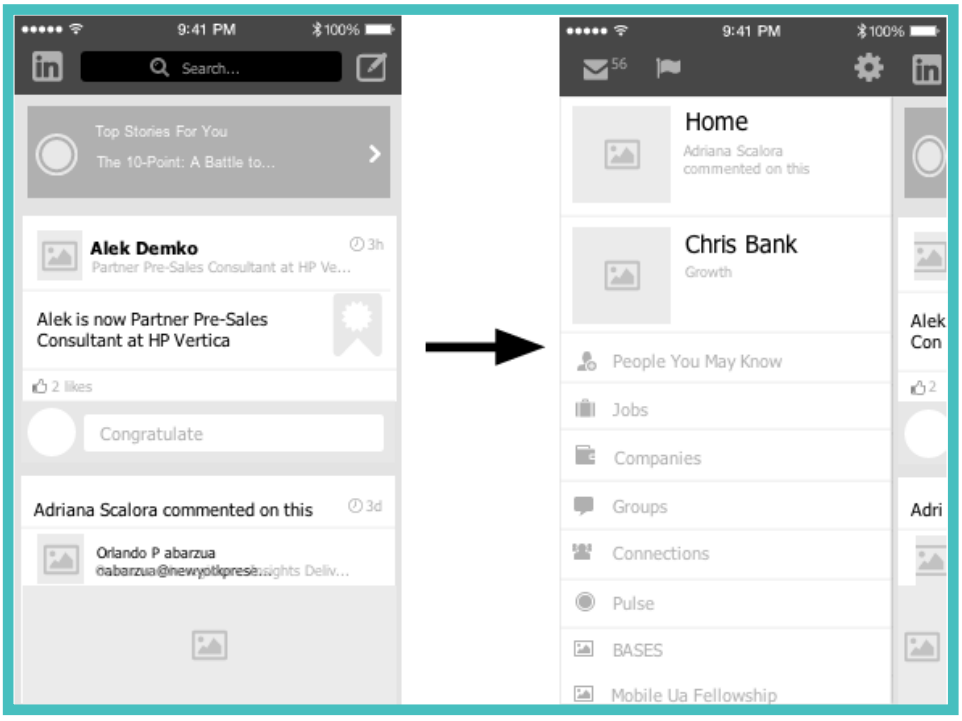


Click to use this wireframe in UXPin



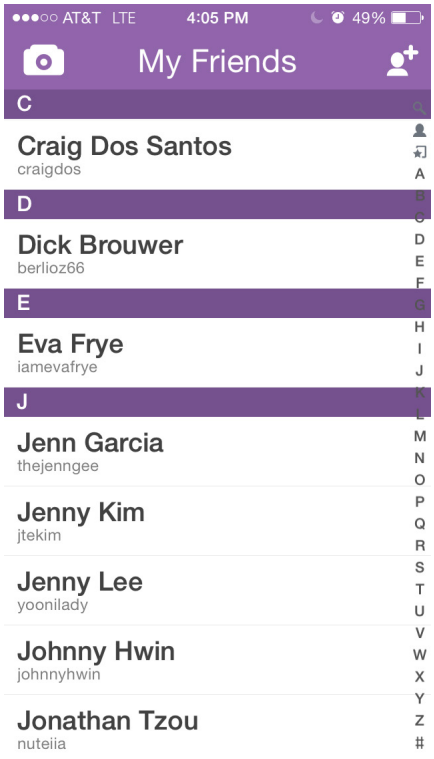
LinkedIn let's you get to reach the main menu from anywhere by tapping the logo (**commonly a 3-line "hamburger" menu icon**) in the top left corner to access this drawer. This UI design pattern was first introduced to the masses by Facebook in their mobile app, and then adopted by many companies like Path, Fancy, and so many others. It's an easy way to hide all the less important things in a "side drawer" and not worry about how a mobile application should distill the most important information. Instead, you only have to focus on how to distill the most important information in each view that's accessible from the side drawer. I've included a wire-frame example of this design pattern below using UXPin.

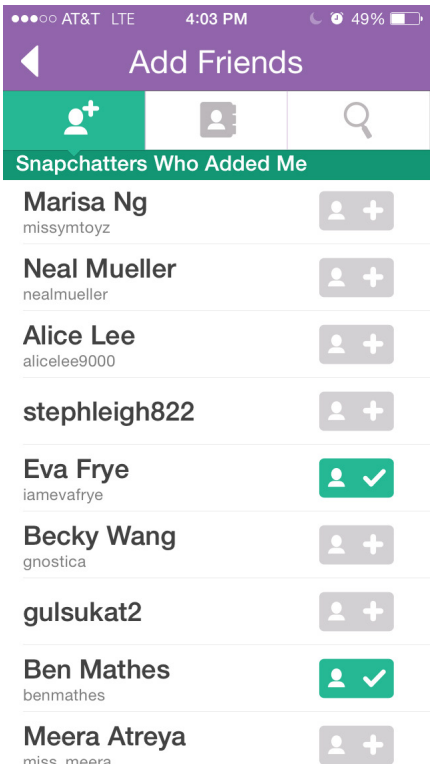


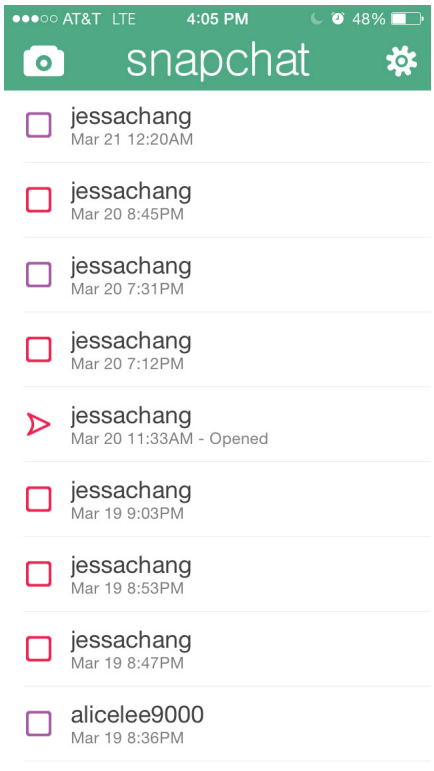


Click to use this wireframe in UXPin

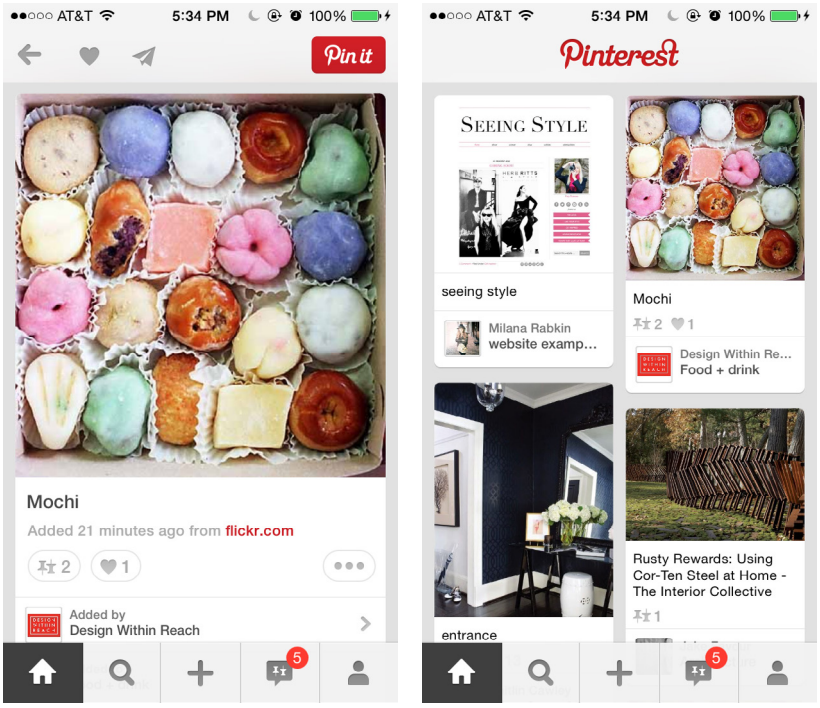
Snapchat is an exemplar of **minimal navigation** for an immersive experience. Instead of showing users 4 menu buttons, they show you the 1 or 2 more important buttons and change these primary buttons depending on what view you're in. To get between views, you can either click one of these primary buttons or swipe left-or--right. This is a unique implementation of such a UI design pattern - I haven't seen such pure execution anywhere else.



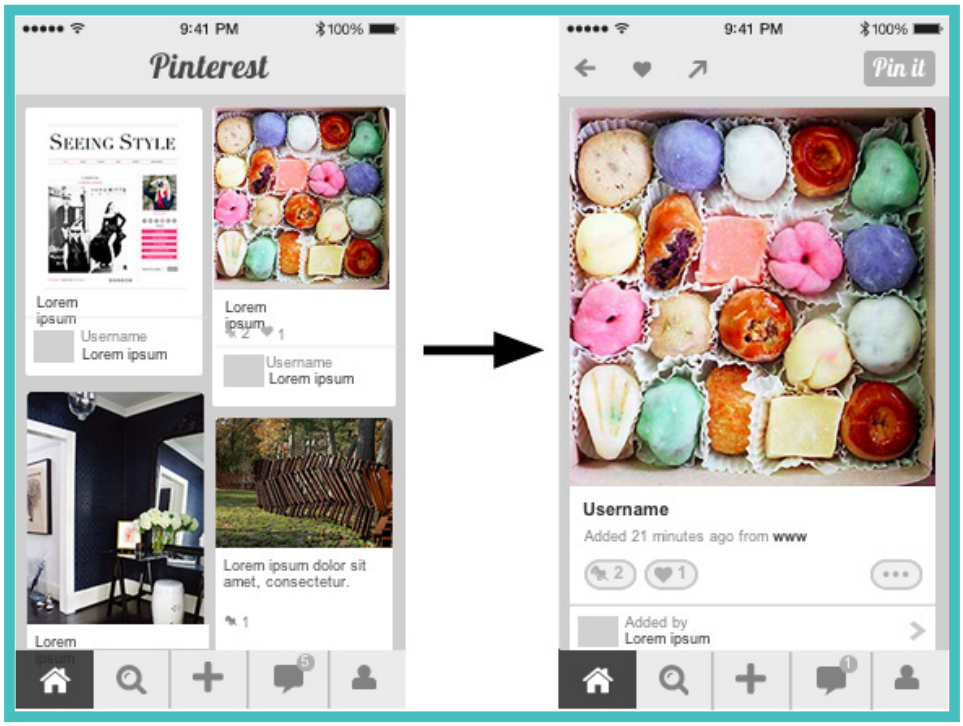




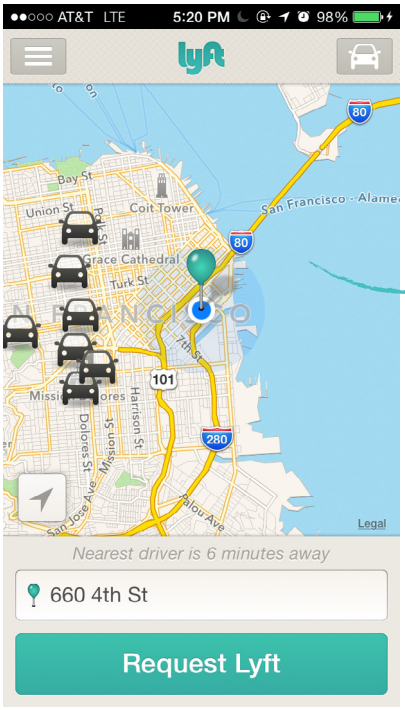
Pinterest popularized **"cards"**, a way to allow users to browse and discover all kinds of content in a more engaging way while accommodating responsive design trends as well as social feed patterns extremely gracefully given it's staggered and randomly sized tiles. It feels like puzzle pieces that could fit anywhere. Sadly, however, there are many terrible implementations of this, especially if you look at Pinterest's competitors who have broadly done anything but truly understand why Pinterest's UI design pattern is better - out of curiosity, I actually compared them in-depth in my spare time in late 2013. I've included a wireframe example of this design pattern below using UXPin.

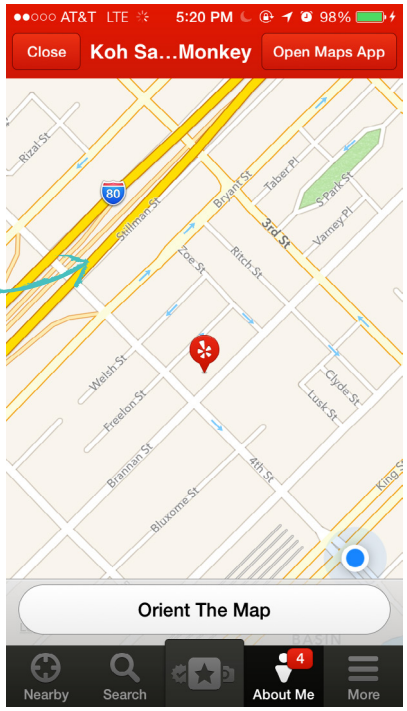


Click to use this wireframe in UXPin

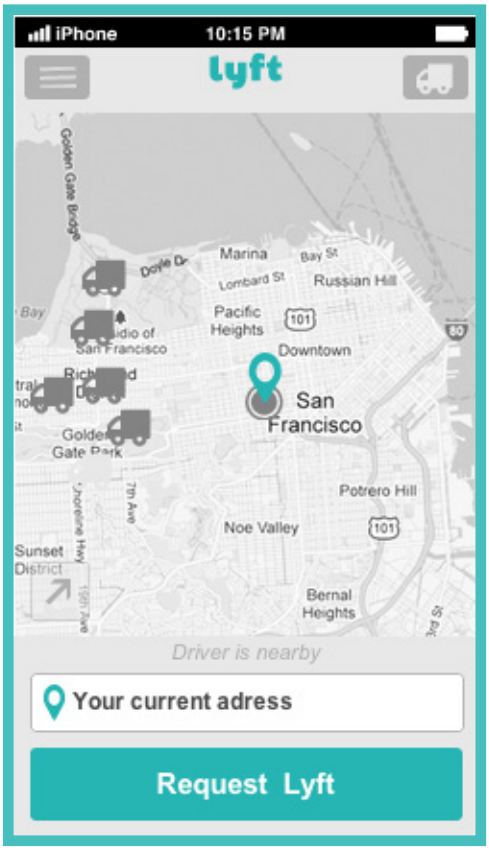


Lyft and **Yelp** provide **maps as backgrounds**, which makes sense given their local nature. This will become an increasing trend as local applications become more prominent and more information can be layered onto the map view, making maps a full experience not just for one-off directions on web or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

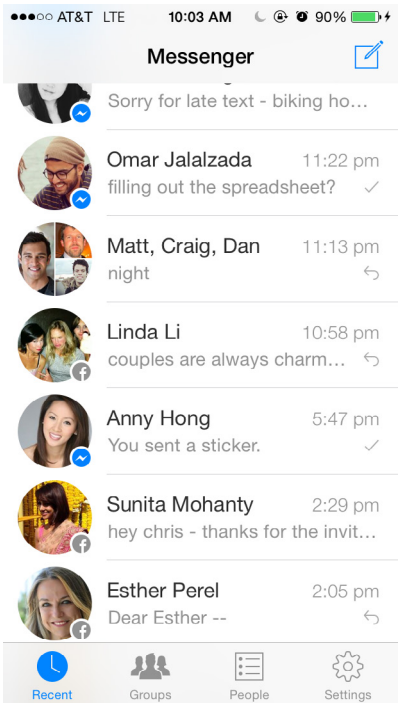


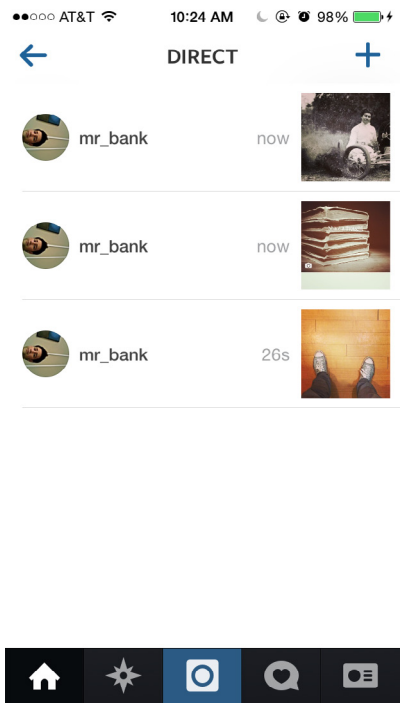


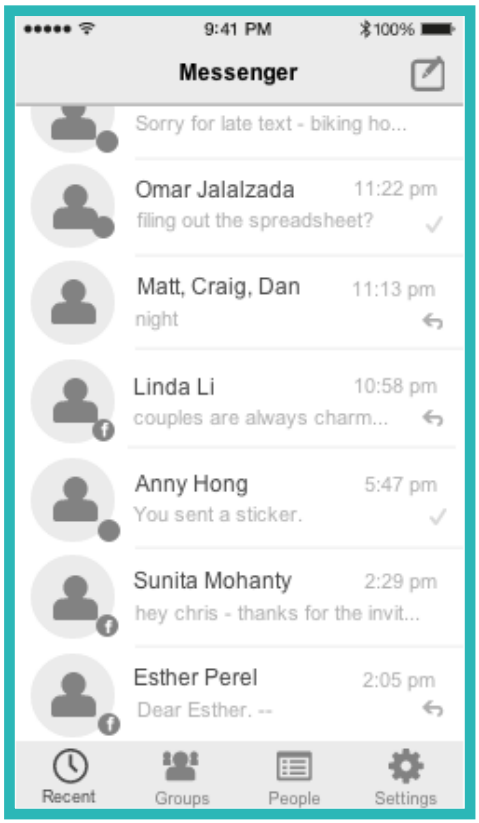
Click to use this wireframe in UXPin

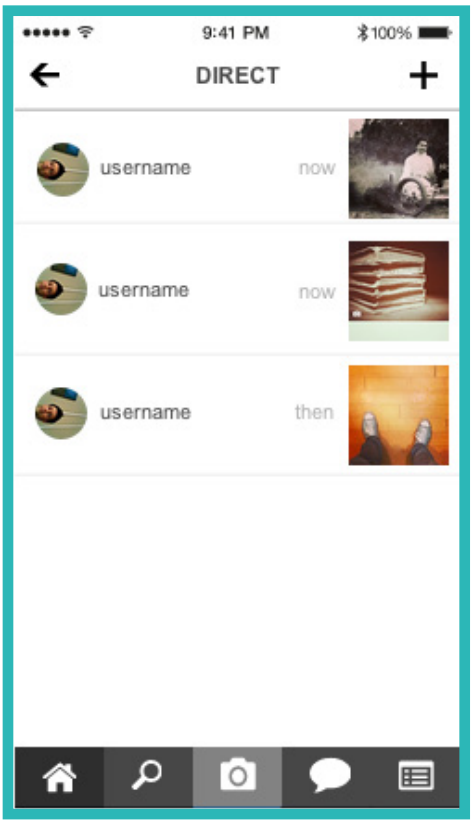


Facebook Messenger and **Instagram** shows all user thumbnails in **circles**. Popularized by Google+ and improved by Path in some respects, this UI design pattern is gaining popularity although it's benefit over the traditional square thumbnail is not clear other than adding variety, the unequivocal "spice of life." We've included wireframe examples of this design pattern below using UXPin.



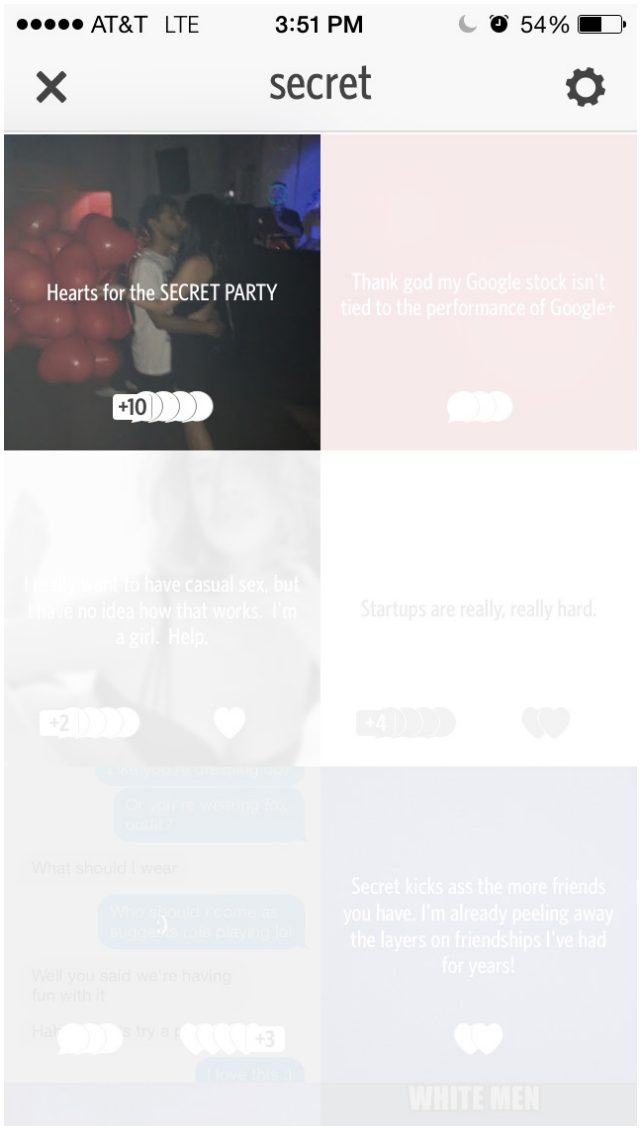


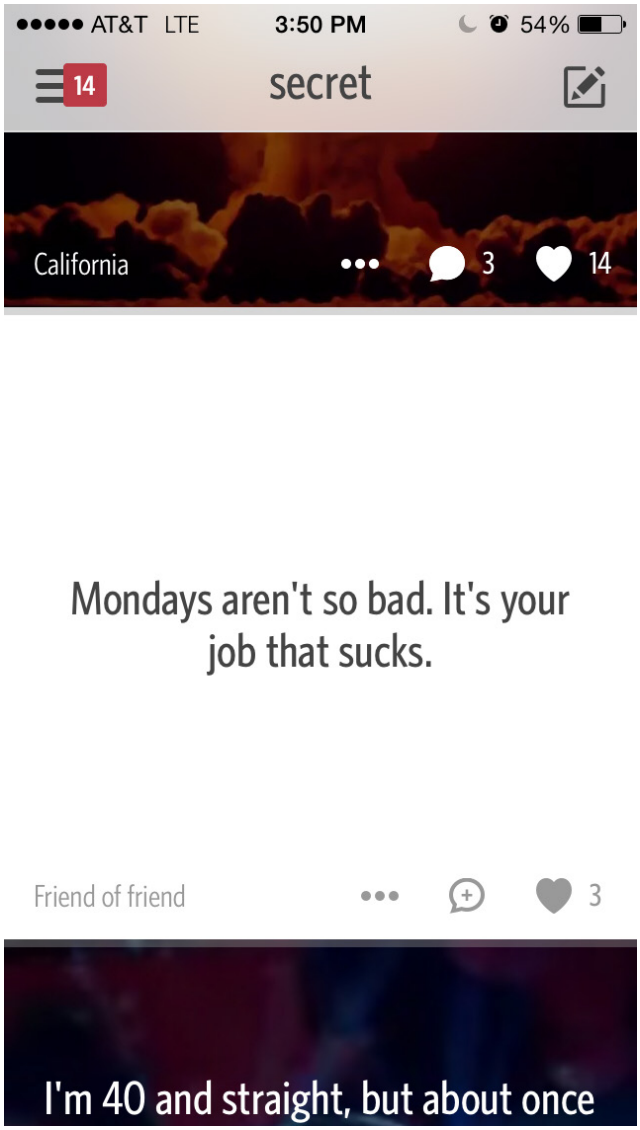




Click to use this wireframe in UXPin

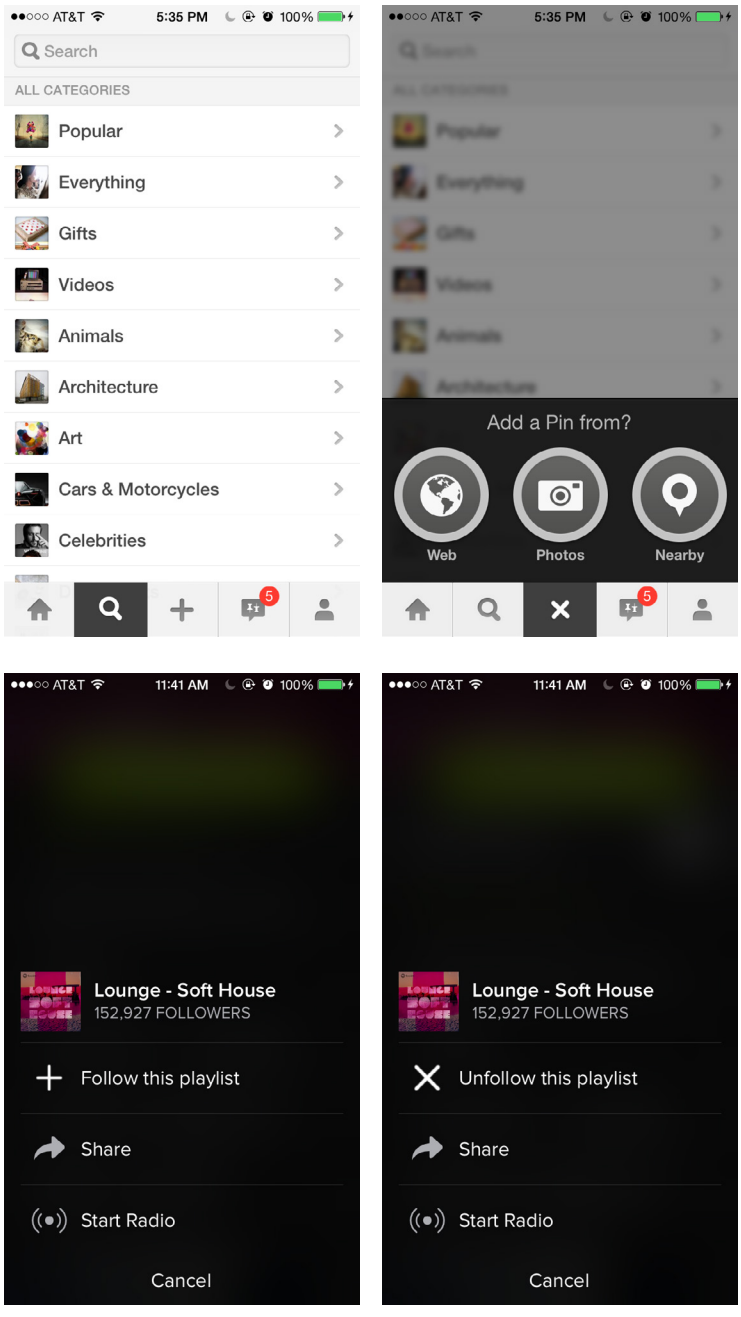
Secret takes no whitespace for granted by stacking **full-bleed images** on top and next to each other while layering some important information on top of them. In a way, these images act as **images as backgrounds**. In another, this UI design pattern kwwpa the user extremely engaged with even less white space and distracting design details than Pinterest.



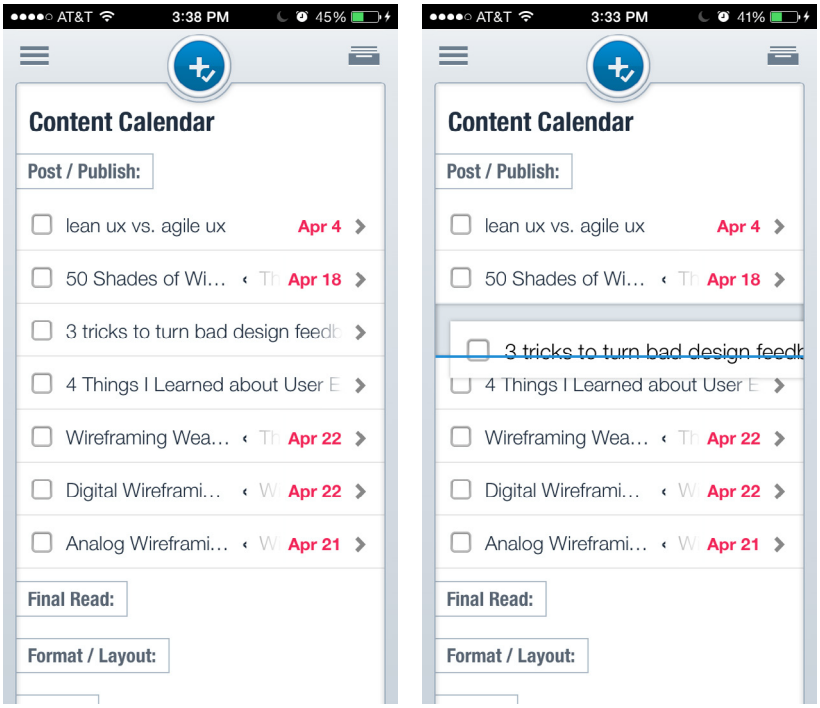


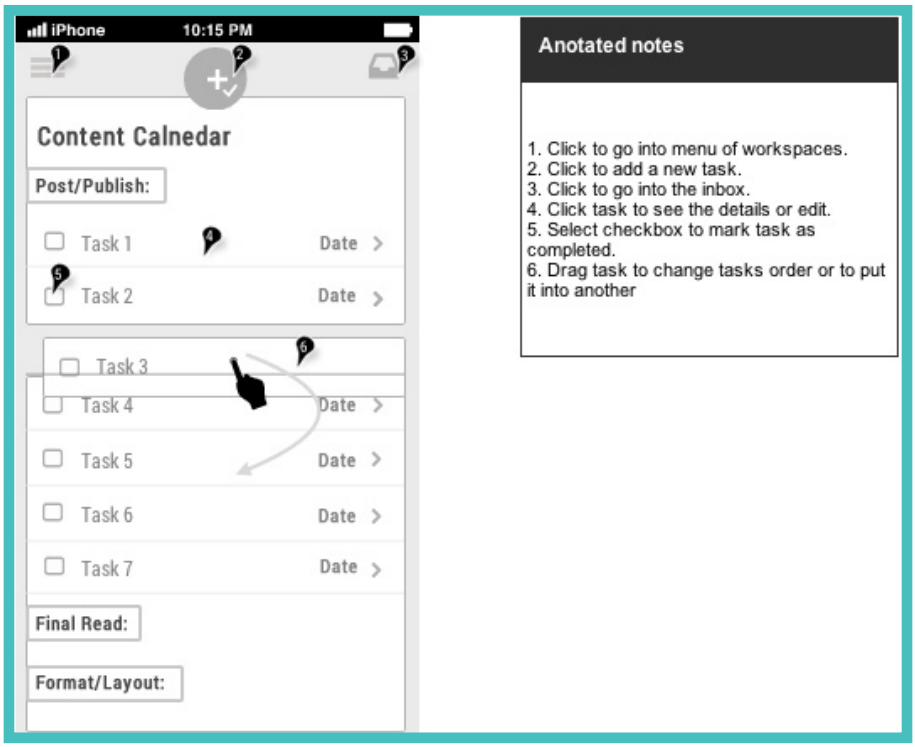
Pinterest and **Spotify** let you know you can cancel adding a pin or following an album, respectively, by **transforming the "+" into an "x" button**. This UI design pattern saves real estate, makes undoing any action quick and clean, and is an overall playful solution.

Transformations and animations are particularly important in mobile applications. You could 1) entirely replace an element with another that has slightly different functionality, e.g. "do" and "undo", 2) visually connect elements, e.g. zooming in on a photo and dissolving surrounding elements when you click on it, 3) or give visual feedback about what is happening, e.g. a transparent shadow under a draggable object on the phone.

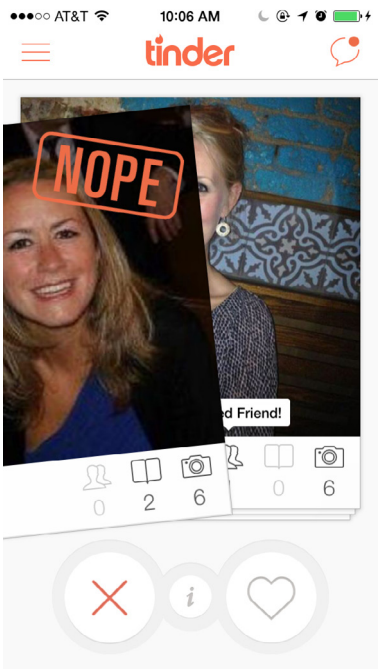


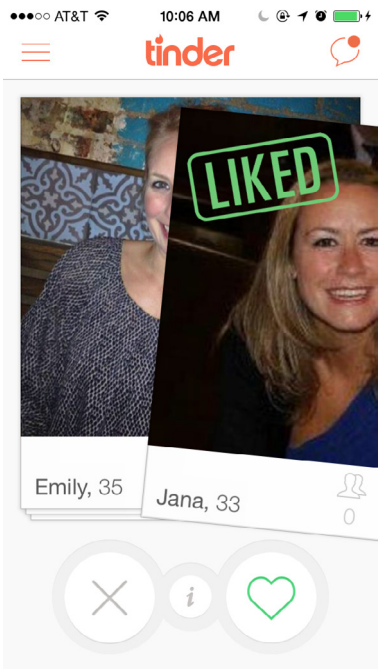
Asana let's you **manipulate content directly**, such as moving tasks around by clicking-and-holding or keyboard shortcutting (on web) or pressing-and-holding (on mobile) then dragging-and-dropping them wherever you want - you obviously don't need to do this for keyboard shortcuts. If you have a lot of tasks, another UI design pattern may be more useful but, for most use cases, this is a great solution if you need to re-arrange list items. I've included a wireframe example of this design pattern below using UXPin.

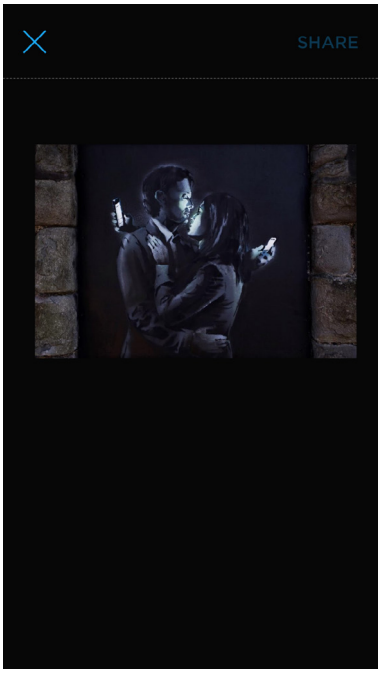




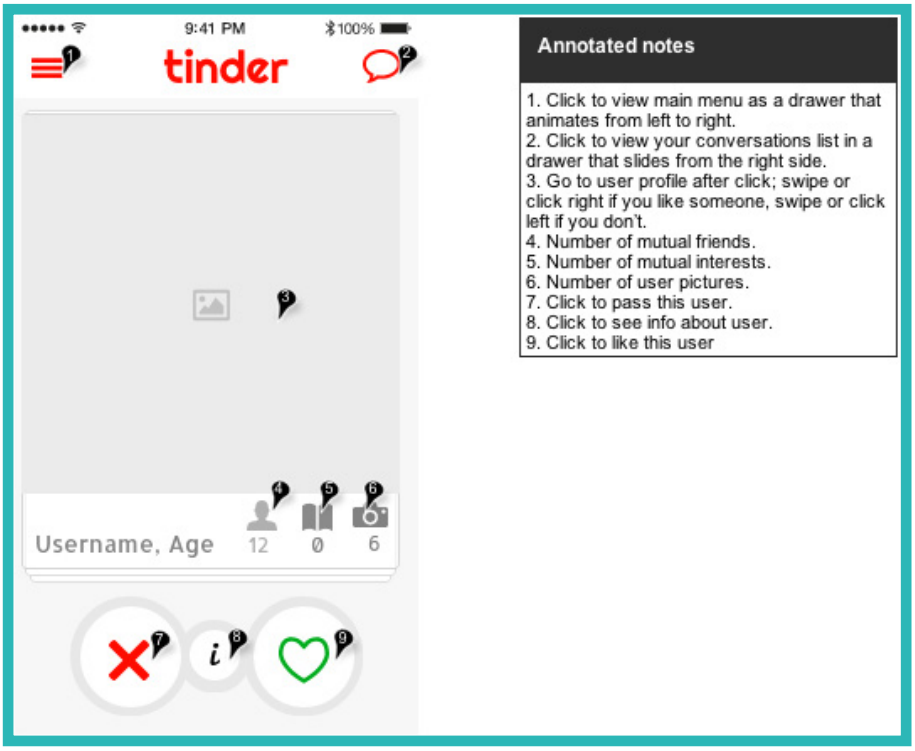
Tinder and **Carousel** let you vote on people or share / hide photos, respectively, with **draggable images**. Tinder also gives you **huge buttons** so you know exactly what to do and can do it quickly wherever you are and whatever you're doing. For Tinder, swipe or click right if you like someone, swipe or click left if you don't. For Carousel, swipe up to share your photos and swipe down if you want to hide them. Tinder is one of the most well-known examples of this UI design pattern because it's the core of this mobile application - it's "Hot or Not" on-the-go. I've included a wire-frame example of this design pattern below using UXPin.



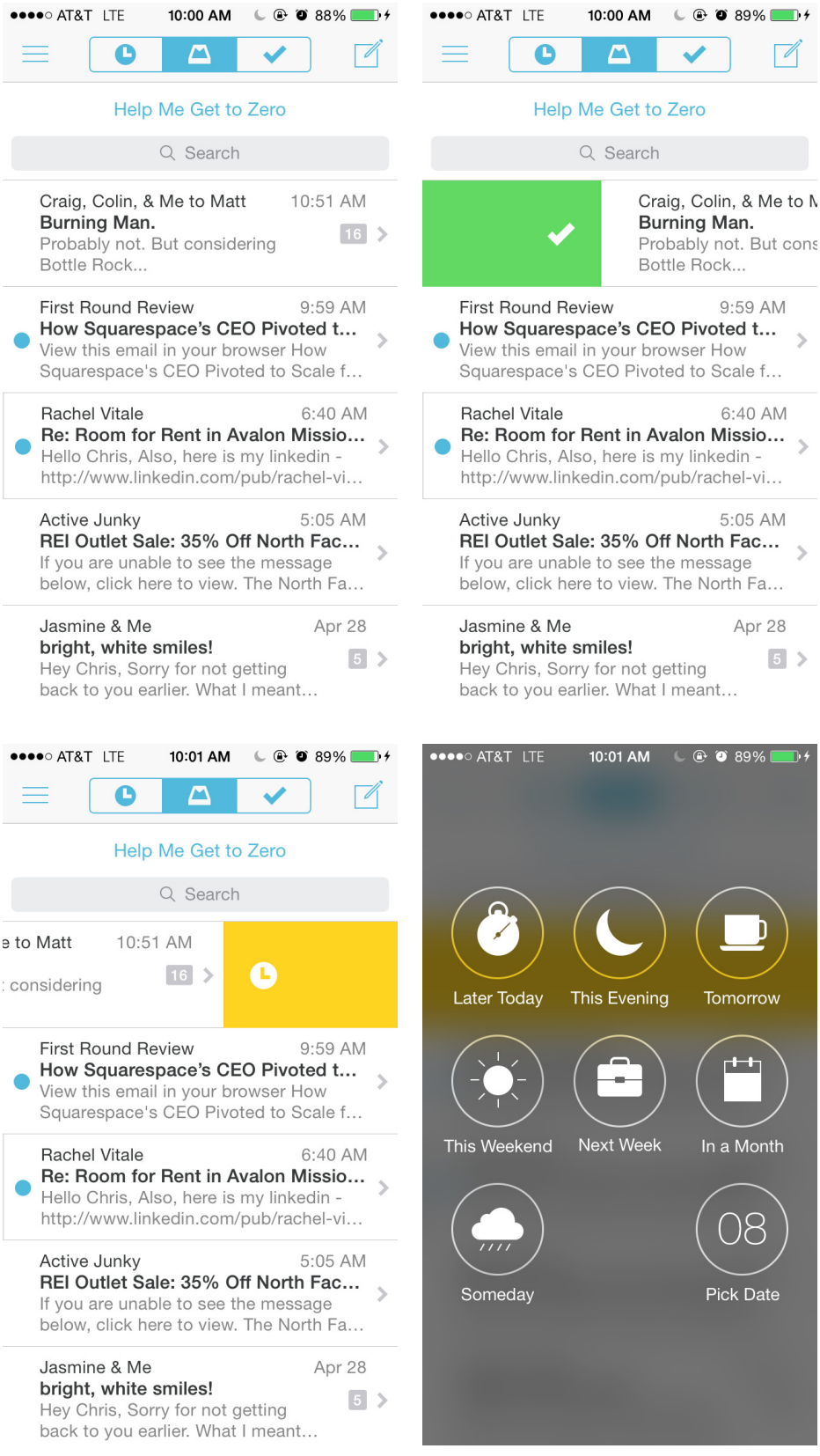




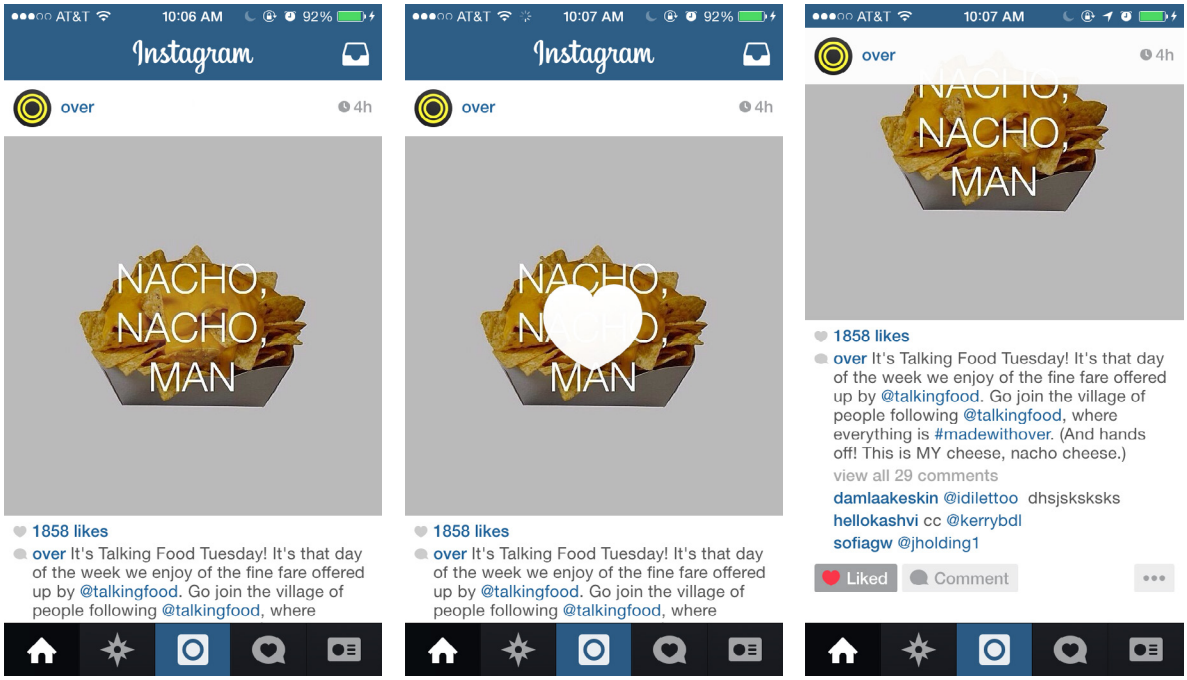
Click to use this wireframe in UXPin

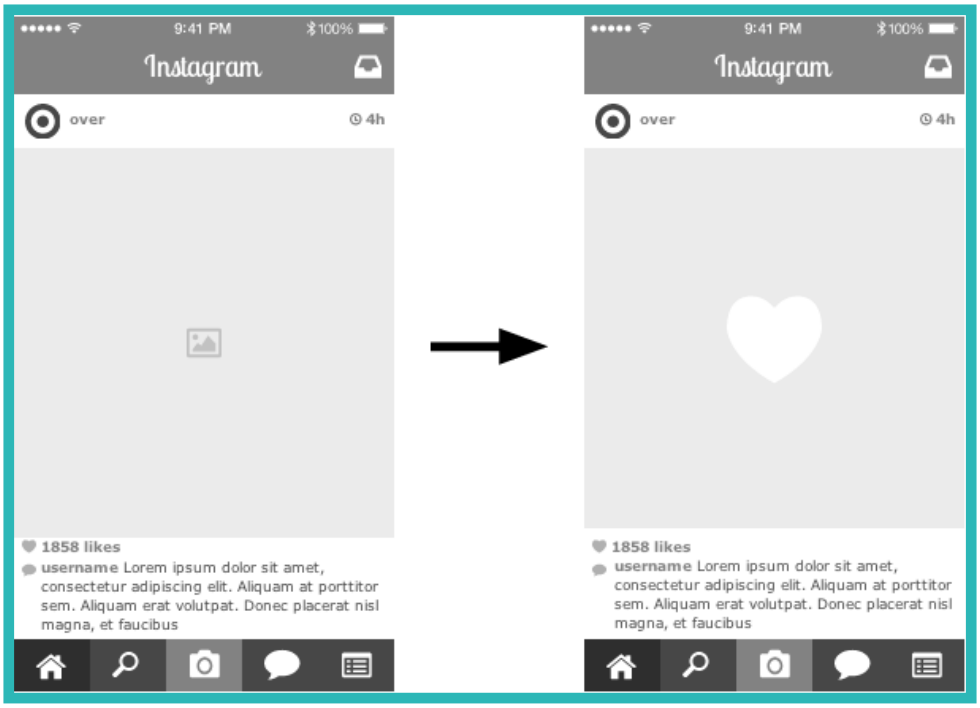


Mailbox popularized the side-to-side **swiping actions** for email clients, allowing you to mark emails as read and schedule them for follow-up by swiping right or left, respectively. This UI design pattern is such an enjoyable and efficient experience that it's no wonder the company was acquired for $100M after being live for 1 month - their viral marketing launch campaign didn't hurt either.



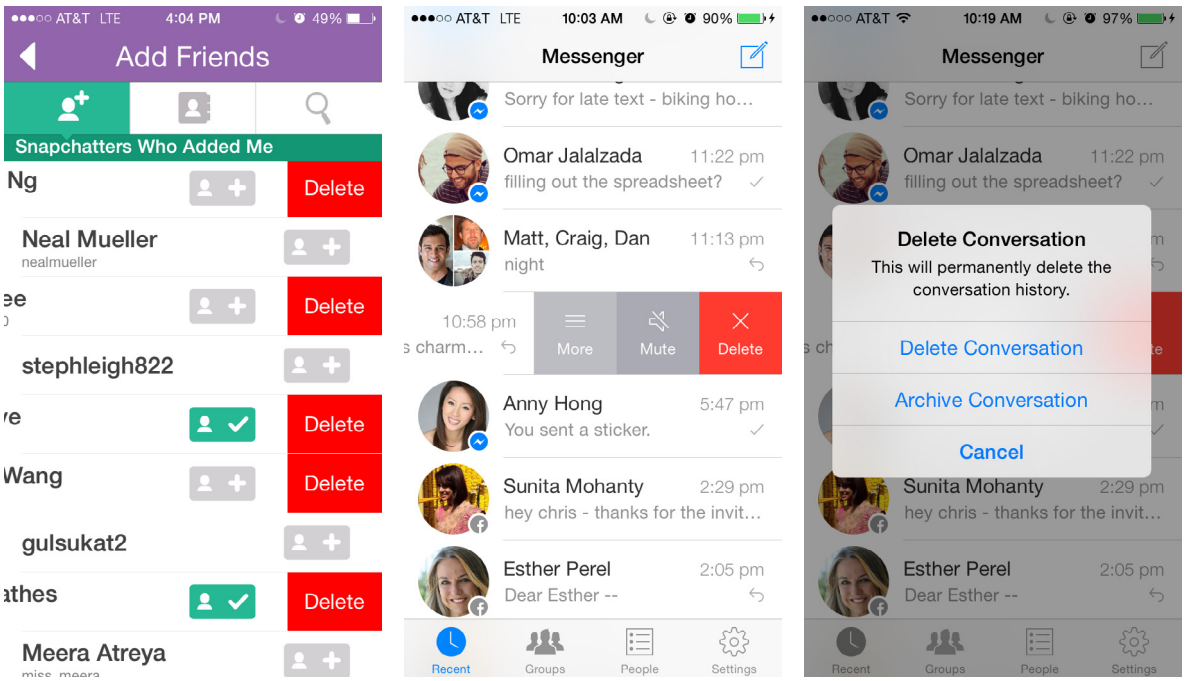
Instagram let's you **discover faster actions** such as liking a photo by double-tapping the photo instead of scrolling down to click the "like" button. I'm personally not a huge fan of UI design patterns that don't let you undo an action, but it's the only example I've seen where you can tap content to categorize it - I'm sure many people have accidentally liked ugly Instagram photos due to this pattern. We've included a wireframe example of this design pattern below using UXPin.

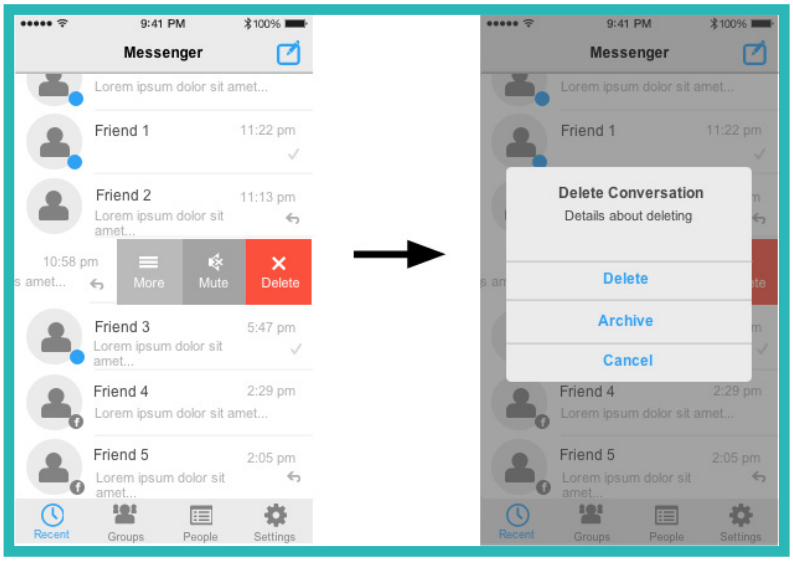




Click to use this wireframe in UXPin

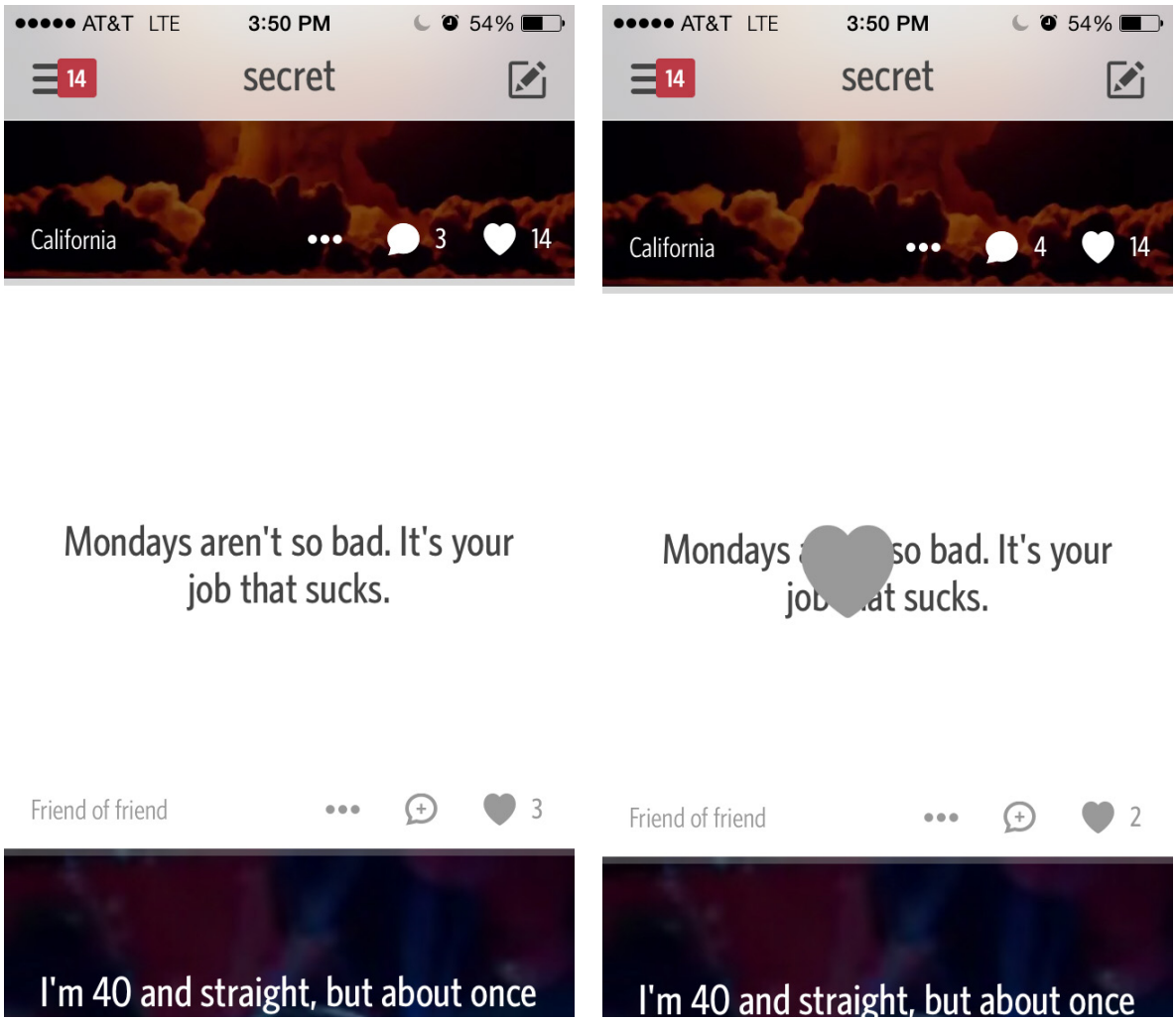
Snapchat and **Facebook Messenger** let you access **features when you need them** by swiping any friend left. For Snapchat, now you can delete as many friends as you want all at once - I call it the “disappearing friend” act, but you can call it whatever you want. For Facebook Messenger, you access even more discoverable actions, including a sub-menu called “more”. Interestingly enough the “delete” button actually gives you options to archive or delete the conversation. In most UI design patterns, you aren’t given that option since there’s an “archive” button in the “more” sub-menu. I’ve included a wireframe example of this design pattern below using UXPin.



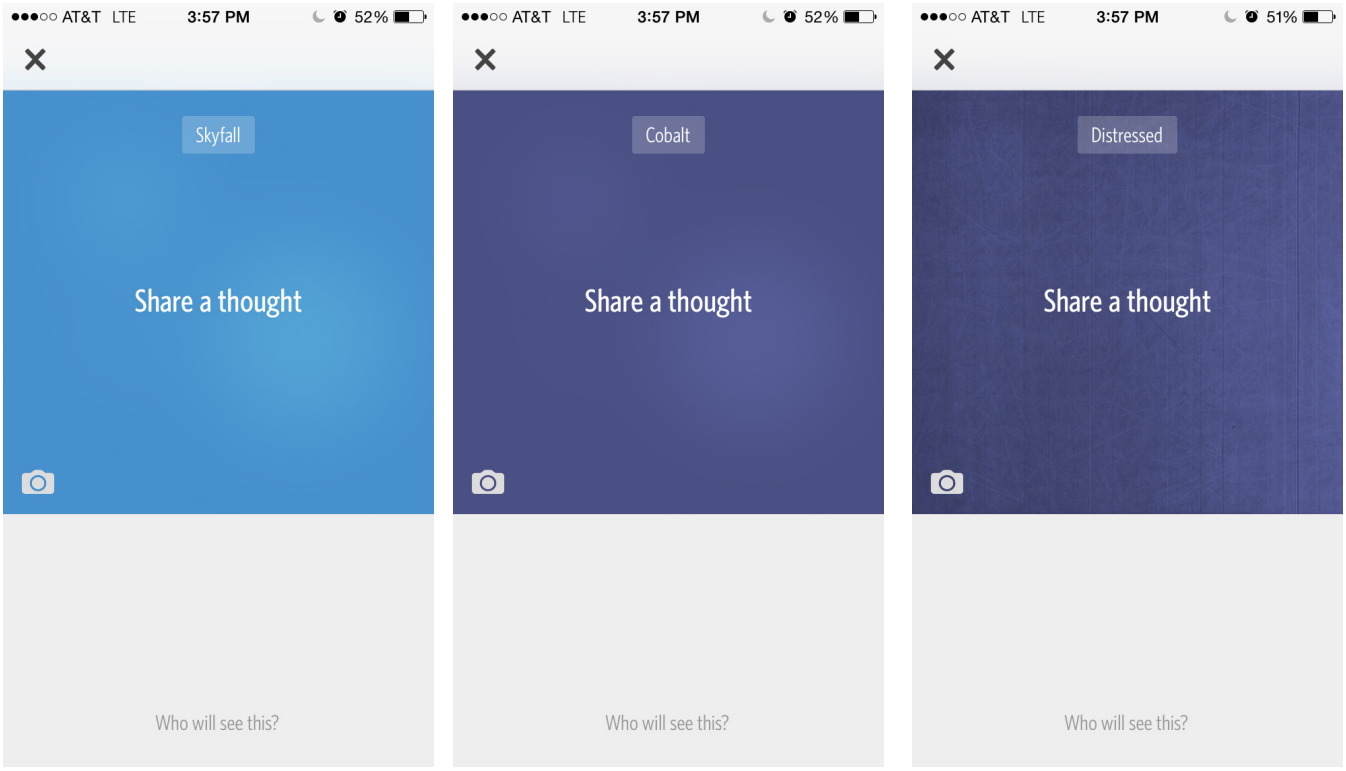


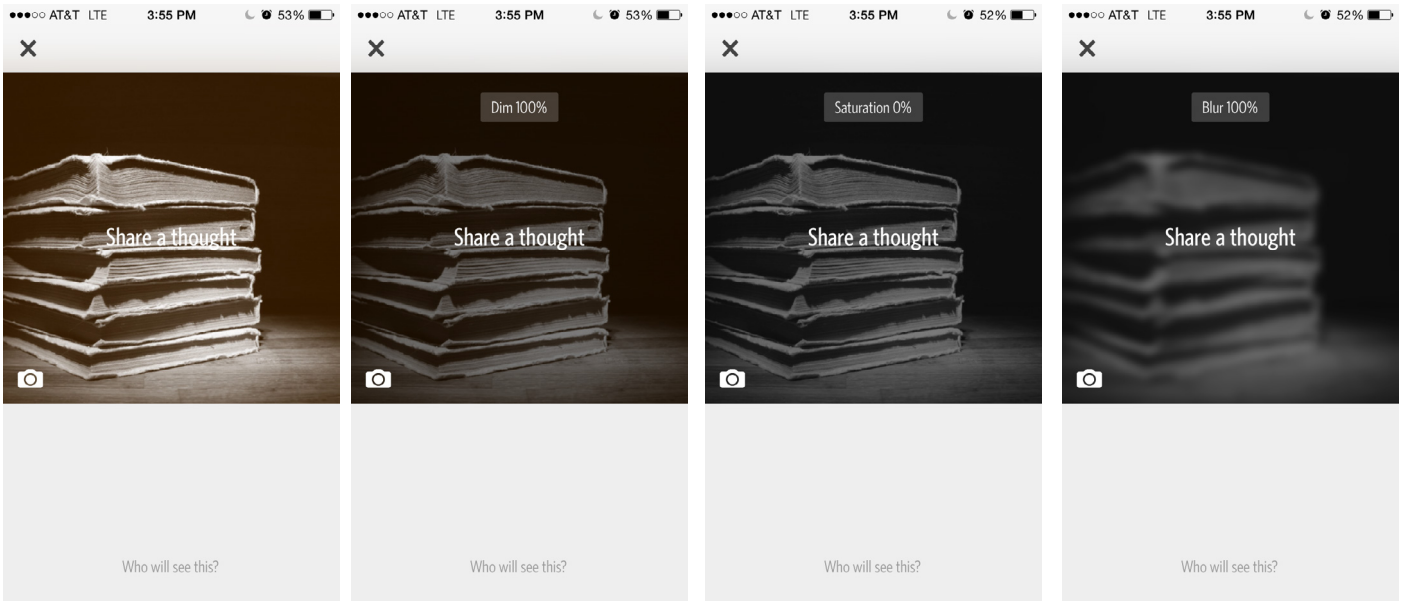
Click to use this wireframe in UXPin

Secret let's you **discover new actions** the way it let's you discover new menus. Swipe left on a secret and you like it. Unlike Tinder, swiping right doesn't let you unlike it, though - that gives you the hidden menu we mentioned above. I'm a huge fan of how they've implemented this UI design pattern although it requires some brain power to remember to remember that swiping right brings up menus instead of "disliking" or "hiding" a secret the way that many similar design patterns have been implemented.

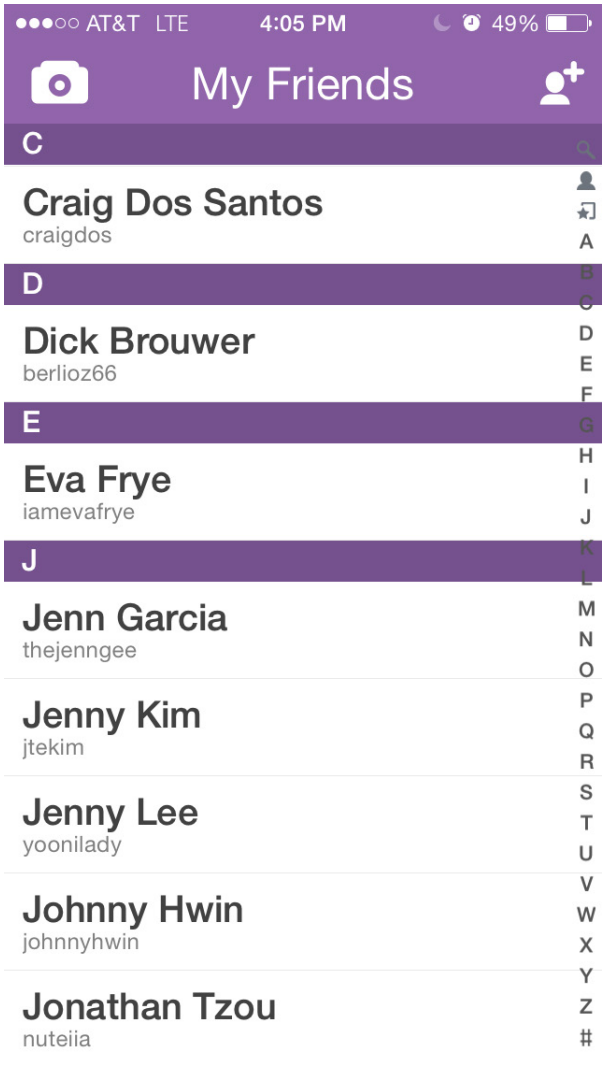


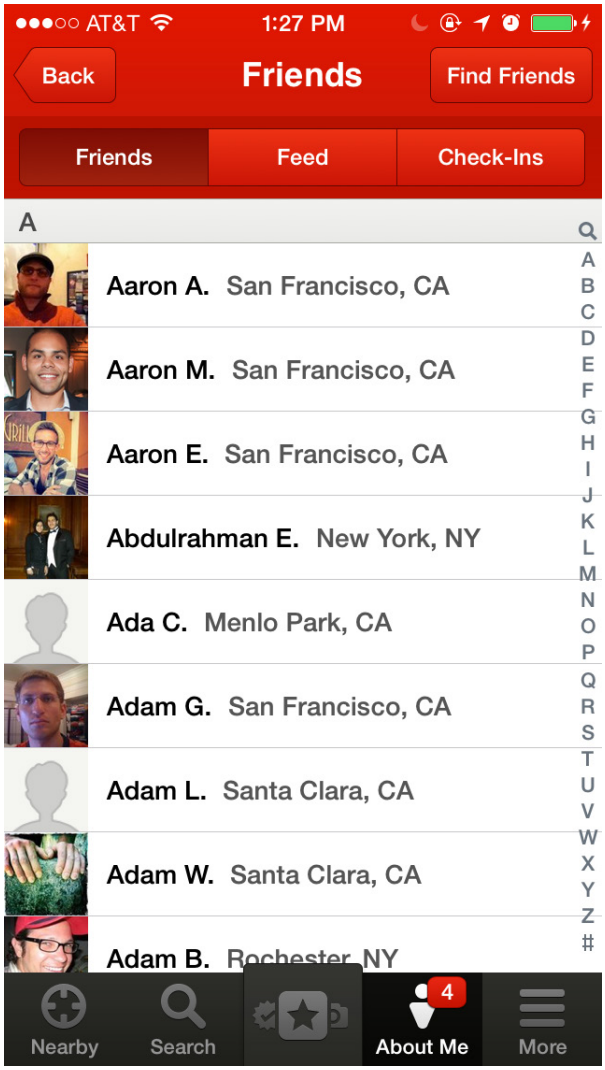
Secret also has **“discoverable” tools** on the content creation side. If you don’t upload a picture, swiping left-or-right changes the color of the background while swiping up-or-down changes the pattern. If you do upload a picture, the actions are even more impressive. Swiping up-or-down on the right side changes dimming while doing so on the left side changes saturation. Swiping left-or-right changes blur of the picture. There are no other controls that let you do this - nor should there be. This UI design pattern is so intuitive and clean that you’re bound to see a lot more of this.



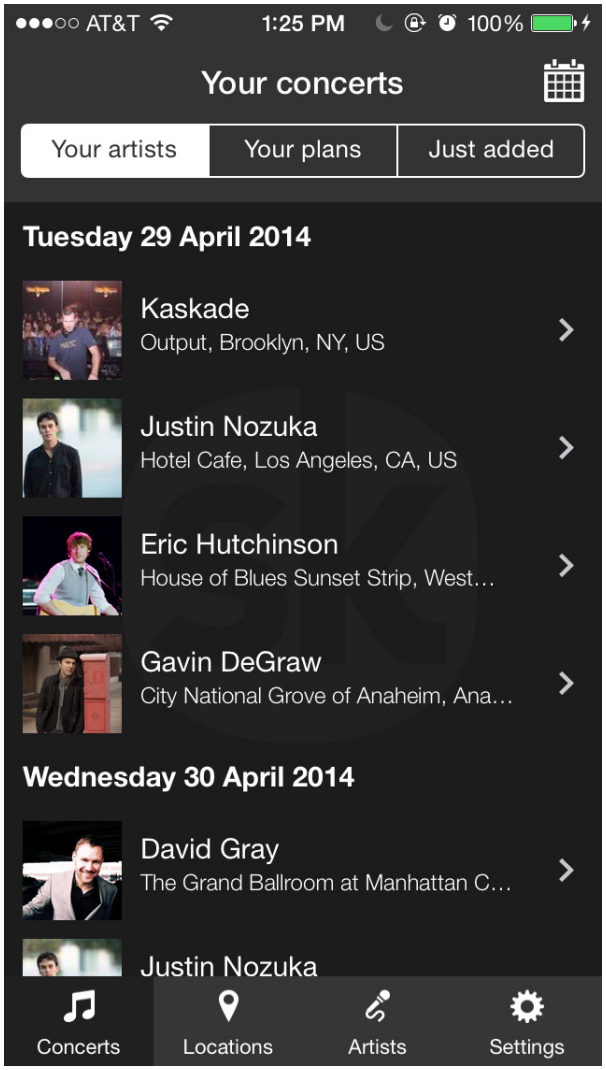


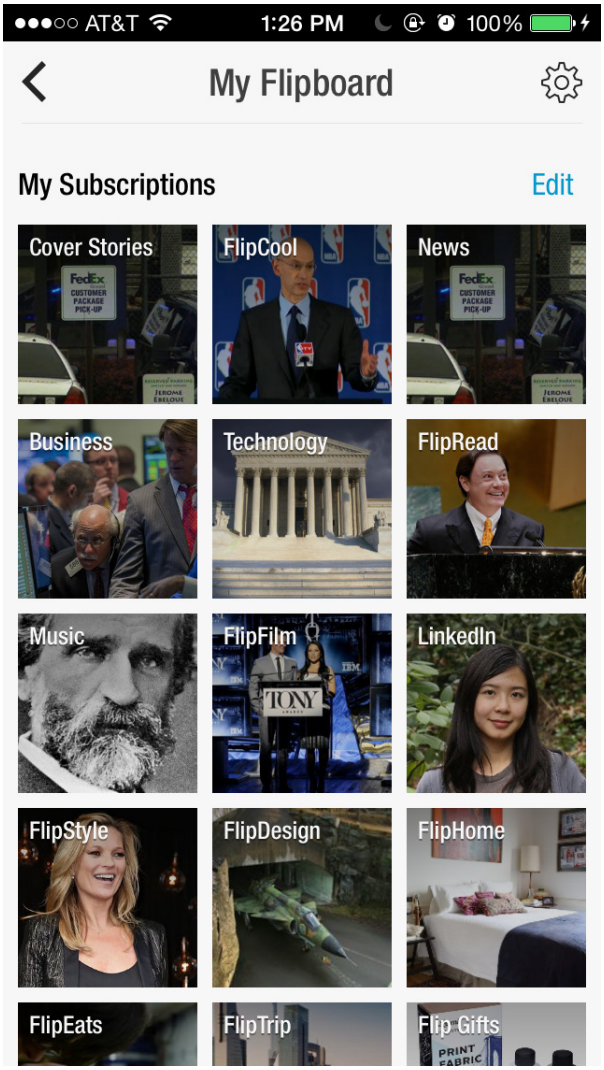
Snapchat and **Yelp** are part of the growing number of apps that give you **friend lists**. Whether it's one-on-one communication or keeping track of someone's tastes and preferences, the way users explore their blossoming friend groups will become increasingly contextual, requiring friends to become a more integral part of the web and mobile experience. I firmly believe social UI design patterns will follow a similar trajectory that content UI design patterns have taken as the average web and mobile user goes from hundreds of friends and followers to thousands if not millions.



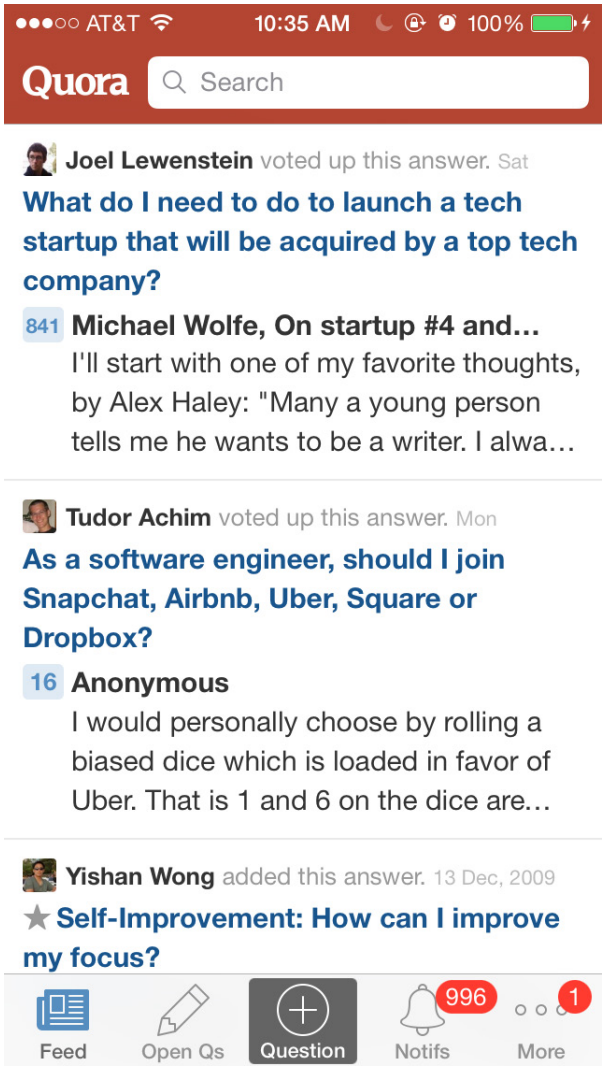


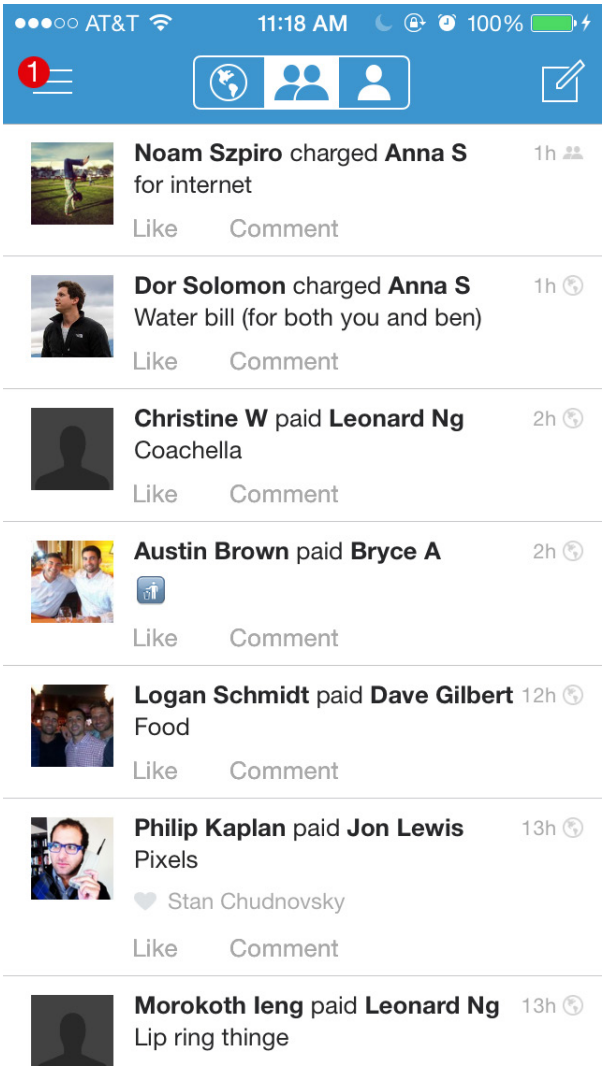
Songkick and **Flipboard** are great examples of products that let you **follow** something, anything! Whether you have friends or not, there's endless user-generated content to keep you busy. For the same reason friend lists will become an increasingly important UI design pattern, so will following.



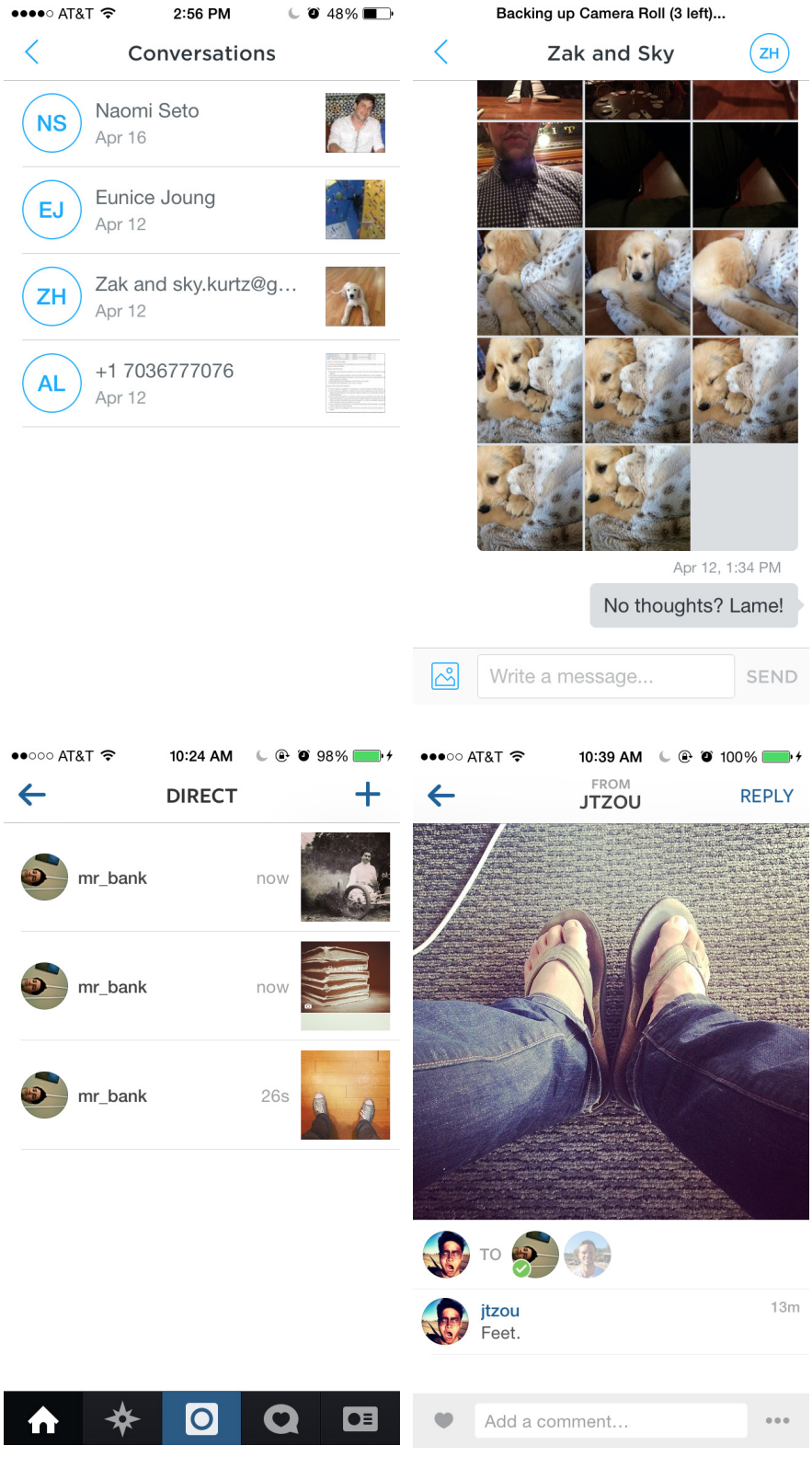


Quora and **Venmo** are two of my favorite **activity feeds** because "learning" and "earning" are two of the primary things people do in life. It's fascinating to passively see people I know provide meaningful answers about self-improvement while others are spending their hard-earned money on pixels and lip rings. Thanks to this UI design pattern, I have as much variety on most of my web and mobile applications as I do on cable TV!

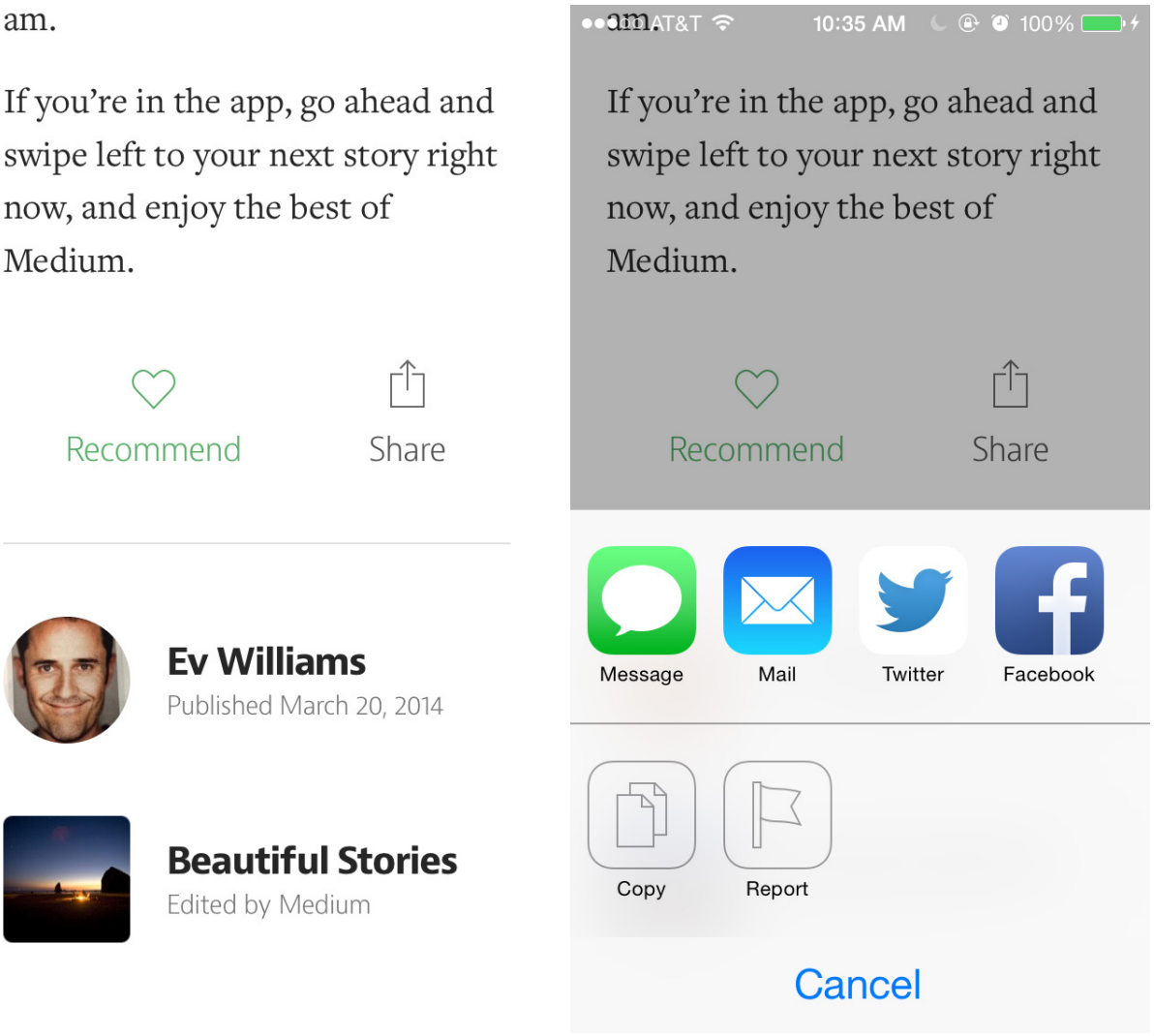




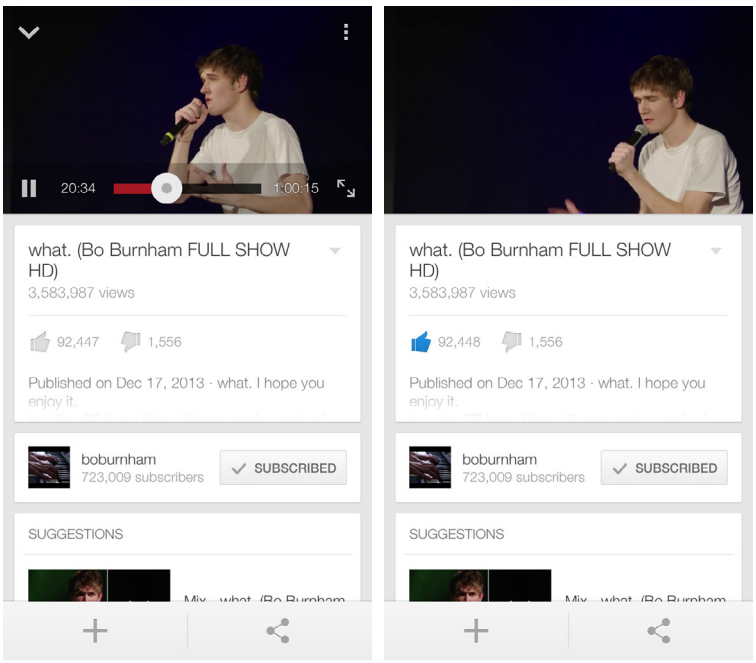
Carousel and **Instagram** and many other apps offer **chat or direct messaging** as an integral part of their experience. Private chat UI design patterns will continue to blossom across many apps, not just traditional “social networks” now that users are finally comfortable sharing more private things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

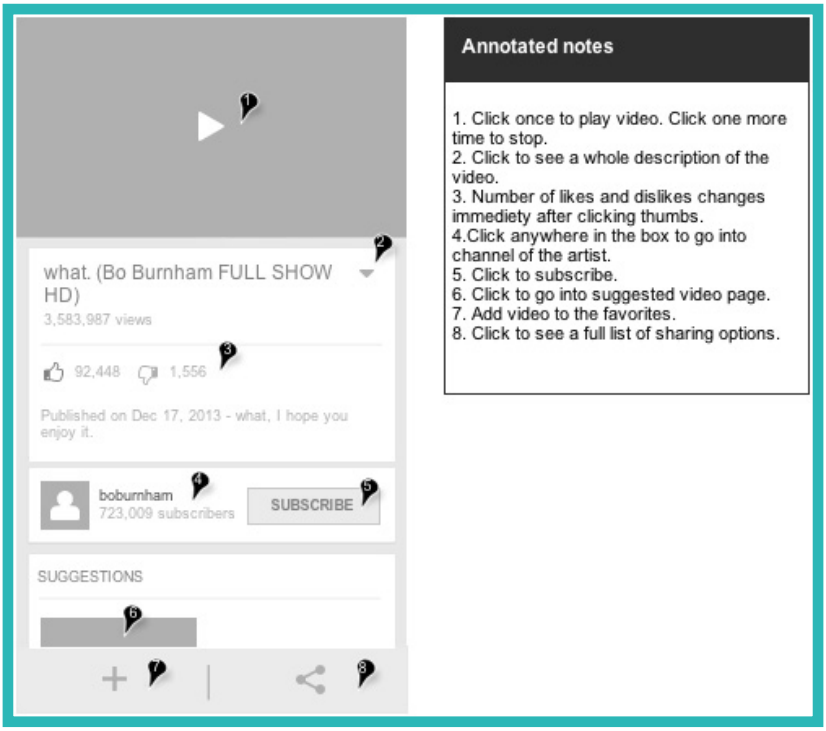


Medium, like many other apps, has consolidated the ugly "share" widgets with a **single share button** to give you a beautiful experience as well as a clear action to share the content, regardless of where you want to share it. Sounds like a win-win UI design pattern to me.



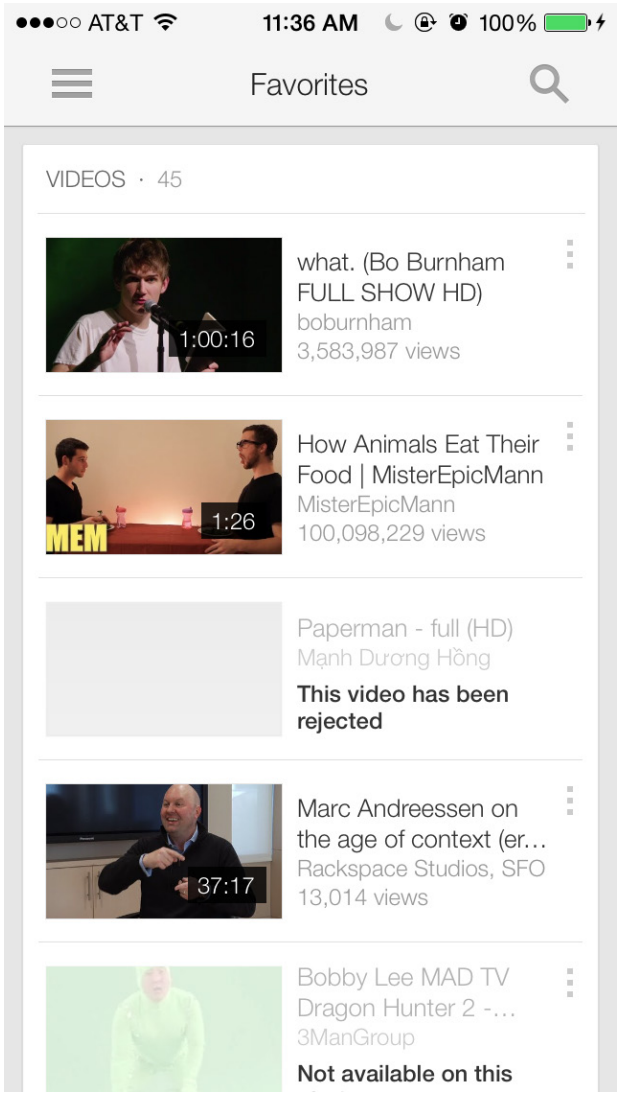
YouTube and almost every application lets you **like (or dislike) everything in a binary way** instead of using stars or other ratings. Although ride sharing apps like Uber and Lyft currently use rating systems, it will eventually make sense for them to have a more binary rating system - the driving was either acceptable or it wasn't. Since this UI design pattern is one of the most elegant ways of organizing content on web and mobile applications, you will see this for a long time - I'm personally hoping that the combination of content tags will allow users to see their favorite comedy shows and music videos separately without any additional work. I've included a wireframe example of this design pattern below using UXPin.

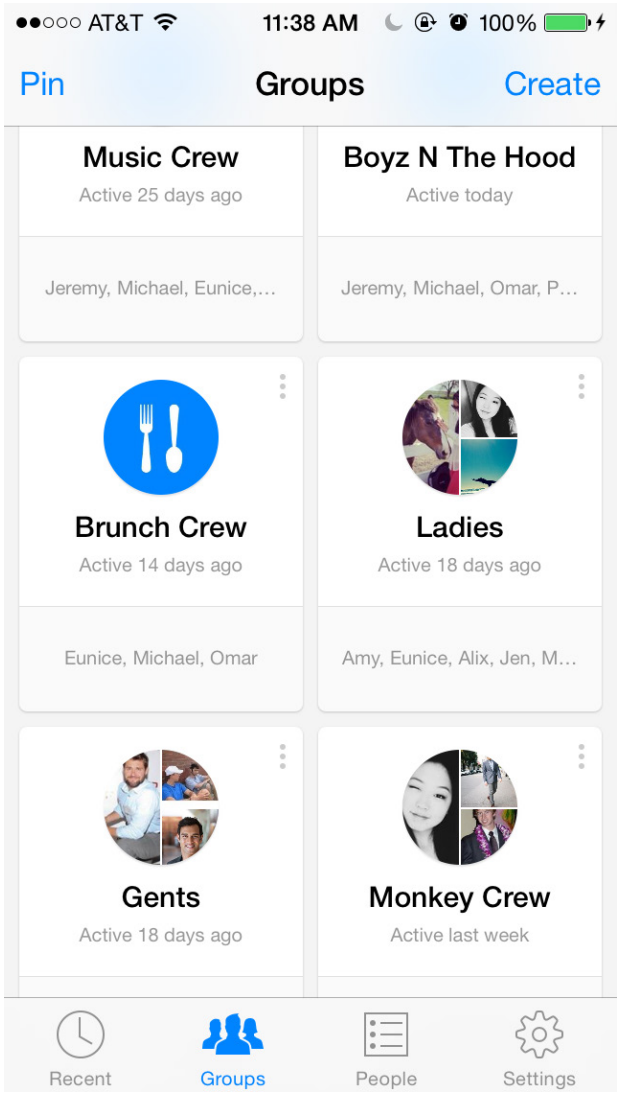




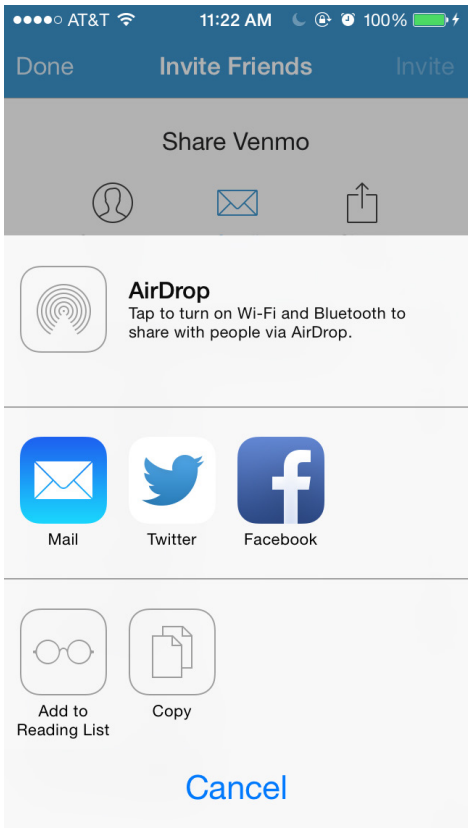
Click to use this wireframe in UXPin

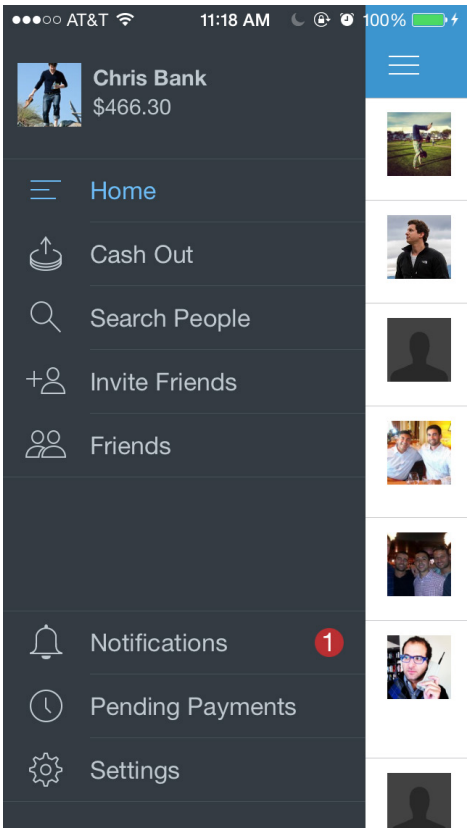
Youtube and **Facebook Messenger** allow you to **group your friends and content** alike. As content of all forms - including friend profiles - continues to proliferate, the ability for users to curate and organize everything. I'm very curious how these separate applications will eventually more deeply integrate with one another as new UI design patterns emerge.

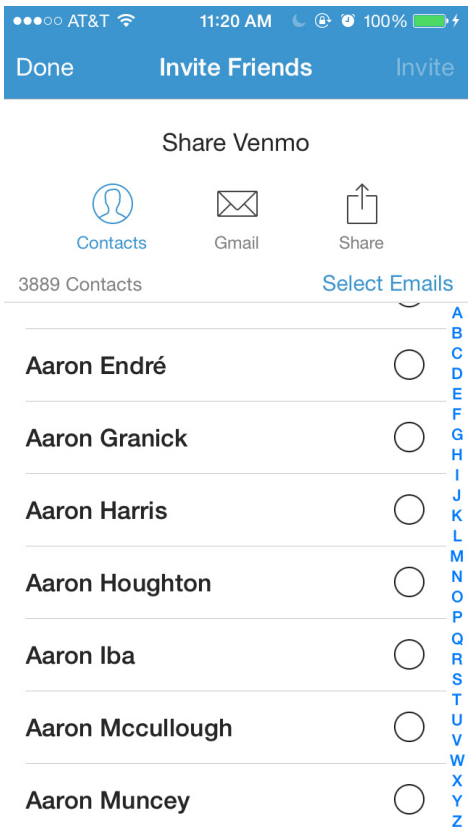


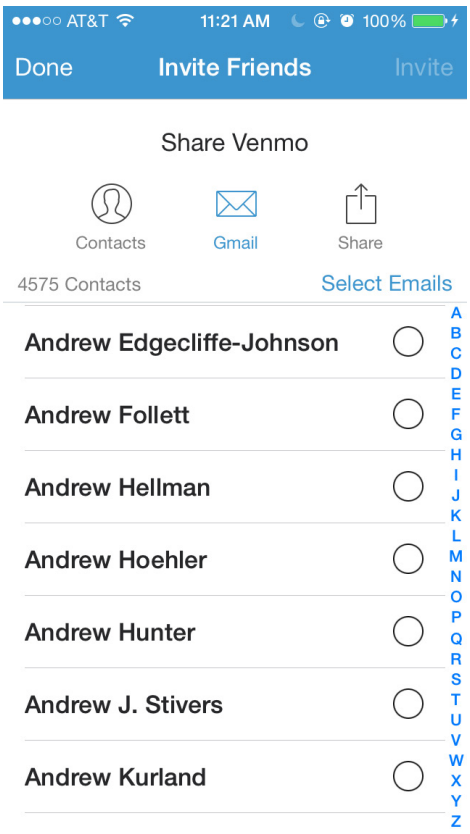


Venmo makes it really easy to **invite others** through social, mobile contacts, and email integrations. Since word-of-mouth and referrals are a huge driver of growth especially in consumer applications, you'll see this UI design pattern proliferate and evolve even more.









Beg, Borrow & Steal... When Appropriate

Don't be afraid to beg, borrow and steal these designs - but make them your own, and solve real problems for your users so they like the product designs you produce.

Because product design trends are moving faster than ever before, and the variety of devices from which your web and mobile applications can be accessed is splintering, you'll have to think about your product more comprehensively than ever before. Luckily, many people are in the same boat so there are many examples out there - including what I've laid out above.

For a full list of UI design patterns, you'll have to wait for our next web design trends e-book. In the meantime, UXPin's current free e-books - Web Design Trends and Mobile Design Trends has a bunch of examples.

To round off this book, I saved the best for last - design principles. In the next chapter, Design Principles for Wireframing, I detail some of the principles that will help you design better wireframes and, ultimately, better products.

8

DESIGN PRINCIPLES FOR WIREFRAMING

A few things to remember while you're iterating on product concepts

image: Jiani Lu

This chapter is an attempt to cut through the noise from the designers, developers, art directors, usability experts, Venture Capitalist and family members to help you design better wireframes and, ultimately, products.

So what does it take to design a successful digital product or service? Is it the brand, the chosen platform, the functionality, the choice of colors, or some viral features? All of the above have some importance but no single element is why company X is a success. We're often attached to the idea that if you just get the right idea or if your design is cool or uses a certain technology, we will be successful - there is no recipe for success.

Take the design process serious, but don't get too attached with one particular part of it and don't rely on any one particular discipline to give you the right answers. Get to the point where you have real users or customers as quickly as possible. It's these users that will provide you with the information that will get you in a position to make better design decisions.

With the help of our friend, Des Traynor of Intercom.io, the following principles should help you get in a position.

Plan a little - do the rest

An important part of building great products consistently has to do with planning. A few questions that you may want to answer before jumping too far into the details:

- Who are the core users?
- What are the user needs and goals?
- What are the business needs and goals?
- What existing product or design patterns work for your users and business?
- What are the gaps in what's currently out there?
- What are some product requirements you have given your users needs and goals (as well as those of your business?
- What are your constraints (i.e. time, resources, money, skills)?

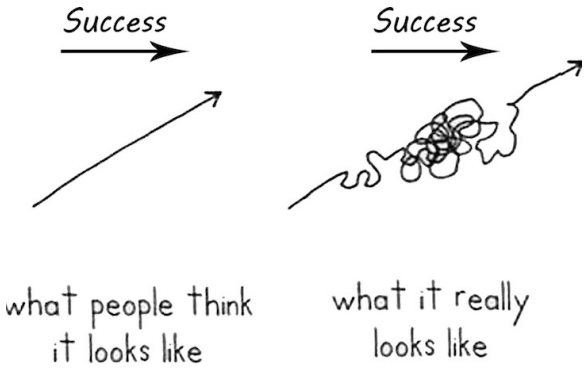


image: weddinggirl.ca

Don't spend too much time on this upfront because sketching, wireframing and prototyping will help you refine your understanding of the answers to these questions. But they're important to have in the back of your mind throughout the process.

Set expectations - not just goals

Have a clear goal of what you're trying to achieve and how you'll measure success.

Intel and Google popularized the term Objective Key Results (OKRs) -- which was presumably derived from S.M.A.R.T goals -- to give product teams this direction and many teams now use this to stay extremely focused in setting their goals. Within this system, each individual outlines their major objectives and the quantifiable actions (i.e. key results) it'll take to achieve them. It's a technique designed for accountability and enforced with scores. In other companies, it serves as a layer of communication that holds the company together and elevates its game at the same time.

Beyond OKRs and SMART goals, it's valuable to set expectations about the design process and each step involved. For example, what are the steps in the design process, who is involved, what are the deadlines for each step, what level of fidelity is necessary to properly communicate, and so forth. Think about what you expect of others and what is expected of them, but try to keep it simple so you don't get bogged down.

image: Incedo

Start with users - and their needs

“Who is going to use my product and what will he or she do first?”

So many people never answer this question or answer it too late in the design process. They start to add boxes, text, colors, and images in a way that makes sense to them. This is great if you’re the actual user. But you often aren’t. Chamath Palihapitiya shares this sentiment when talking about building a product that grows - talk to your users and know your users.

The difference between art and design is that design solves a problem. The design process must start with identifying and thinking about real user needs. You should design around those individual needs and user flows to satisfy those needs. In order to do so, you must understand those needs thoroughly - interrogating data, not just making assumptions - and remember that what users ask for is not always what they need.

The average person - your users - has an okay life. It’s not as amazing or glamorous as it seems in the movies, TV or social media. They’re not getting made, paid or laid as much as they want. And they don’t care about anything you’ve built unless it makes their life better. Don’t forget that while you’re getting lost in boxes, text, colors, and images.

Think how, not what

Your product isn't just a bunch of features - so stop focusing on them.

What matters is not what functionality your product has, but how it works. A sign-up process is not just a sign-up process, a checkout process is not just a checkout process, a button is not just a button, a rating system is not just a rating system. Think about how you can stand out by introducing something that everyone else might have but in a unique way - what you are selling at it's core. Here are a few questions to answer:

- What is needed for your product to function well?
- How much can you take away from it without sacrificing the core product?
- Why will people be excited about it?

Glasses have one primary purpose, to help you see. Everything on top of that - such as colors, shapes, sizes and logos - is a feature. Understand when you are working on your core product and when you are working on adding features. The benefits of thinking like this, is that it will help you establish a very clear and precise picture of what makes your product your product. This will help you understand the minimal work necessary to get a valuable product out the door, and why you are adding features when you are.

You will be surprised how much the "how" can help improving your product - and getting an MVP out the door.

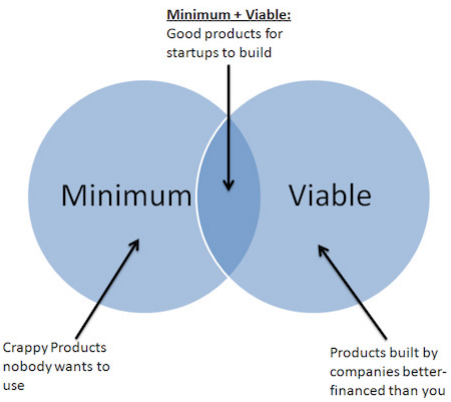


image: Paul Kortman

Start simple, stay simple

Wireframe only what you and, more importantly, your users need.

Constrain yourself. A good product has limitations. It doesn't just succumb to every temptation that comes along. Only add features if you get clear signs that it is needed. It cannot be said enough. Less is more – much more, and for very good reasons:

- First, if you do less you can measure more. And if you can measure more, you can better experiment with what works. Most products are simple, based on simple insights - and the product teams behind them have made many small variations of those products in the process.

- Second, most users will have to learn your product anyway so don't try to impress them with features before they understand what your product is all about. Don't add new features just because you think that it will help, it wont, not yet. If your product becomes a success it's not because of how many features it has.

- Finally, you'll save more money by focusing resources where they'll do the most good.

Dropbox has much to be desired, Mailbox is pretty basic, Spotify still has a way to go, Pinterest could be so much more robust, and Uber could have way more bells and whistles. But they're all rock-solid products with focused features for focused user needs and use cases.

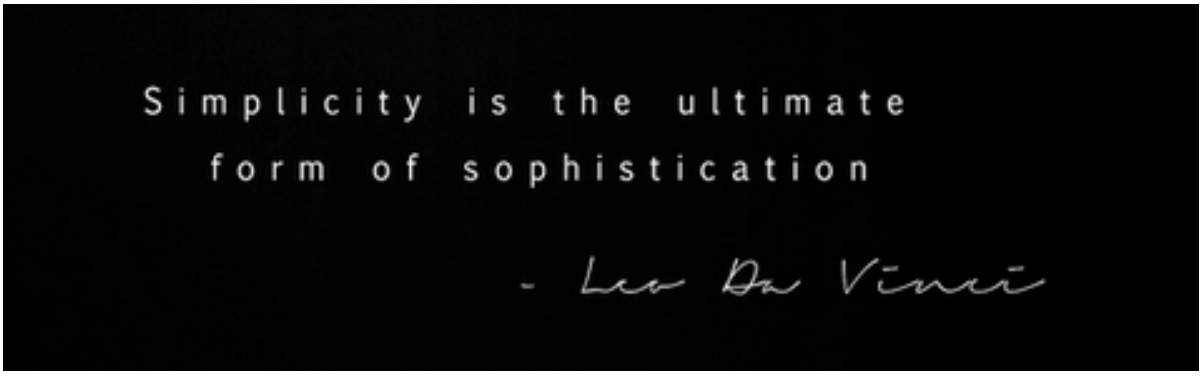


image: ***boldchat***

Don’t confuse change with improvement

One of the biggest challenges faced by designers, artists, and generally creative people when producing new work is fatigue. They get this from reviewing their work and their inspirations - every minute details - over and over and over. Top tech companies and startups have similar problems - largely due to the intensity and time spent immersed in them.

It’s very tempting after a couple of months of looking at the same design to want to change it, and think you are improving your product. Make sure you stay focused on things that really improve the product and make your users even more happy, acquire more users, convert them and so on - whatever your product objectives are. If you end up making changes that don’t improve - or, more commonly, hurt - the product, you’ll have to spend resources on changing things after you launch.

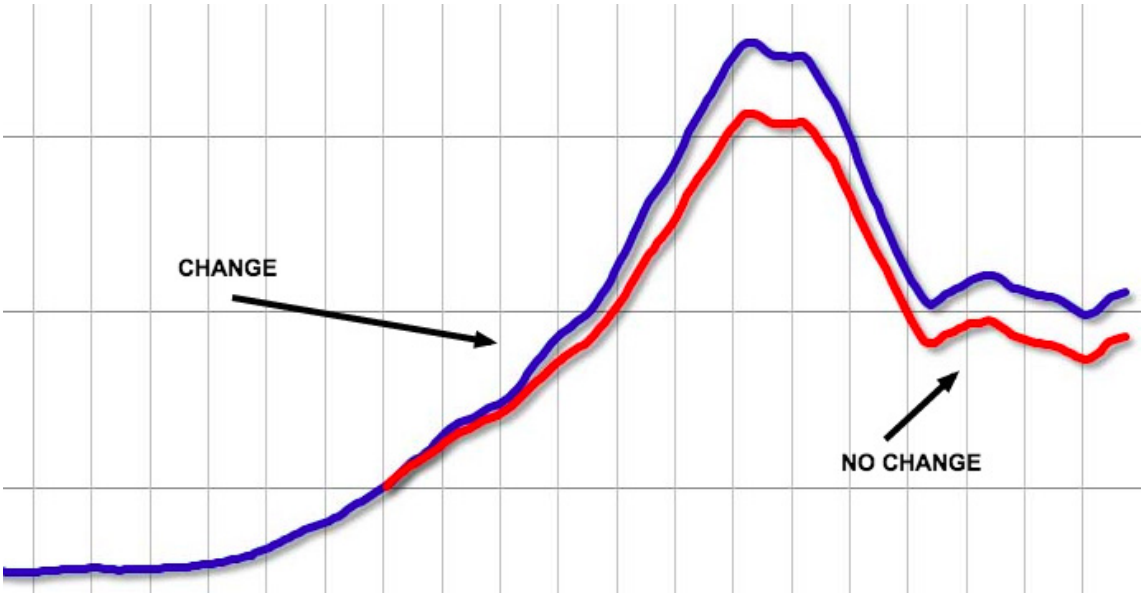


image: 4.bp

Everything has a meaning

Your product sketches, wireframes, and prototypes are methods of communicating just like any language in the world. Every piece of text, color, gradient, shadow, shape, and image you put down has a meaning, just like there are definitions for every word.

A frequent issue I see with wireframes and prototypes is the variety of colors or grayscale shades, line weights, font types, and element sizing or layout - all without much thought, if any. This adds confusion as you don't know whether these slight variations should translate to variations in the end-product and, if so, what they will communicate. It's as if someone did graphic design without a style guide, or someone spoke the English language without ever reading a page of the Oxford English Dictionary.

image: Grammarly

Be consistent, not uniform

Wherever possible, use the same language and design patterns to help familiarize users with your product. But, when this isn't possible, you should at least make sure your underlying approach is consistent so your users can reasonably guess what they're supposed to do.

Presumably, you hope they use the product very often - hourly, daily, or weekly - so

you're helping them get comfortable doing so. You're also helping them develop habits that will pay dividends down the road - all while going about their daily lives.

Granted, you can't imagine every scenario and write rules for it like a design style guide. Every circumstance is different and should be addressed on its own terms. What unites things, therefore, should be a consistent approach - one that users will hopefully come to understand and trust.

As this pertains to sketching, wireframing and prototyping, it's a valuable discipline to limit the palette of shapes, colors, text, and graphics initially. This will help you be consistent without being totally uniform (and arguably boring) or having too much going on, which could be wildly confusing. As a rule of thumb, I always suggest limiting each attribute (color, grayscale shades, text, elements, etc.) to 2-3 variations so you're comparing apples to apples instead of wondering which variation is an apple, and which one is something else - could be an orange or a lemon. This will result in a less "sexy" wireframe, but that's not the purpose of a wireframe - leave that for mockups and high-fidelity prototypes.

What's nice about sketching is that you only have one color and font everywhere until you physically decide to switch sketching tools (i.e. a thicker marker or a colored pencil). Similarly, wireframing tools also limit your options for line weight, color and font types. While wireframing and prototyping tools like UXPin give you more flexibility, other tools like Balsamiq are extremely limiting for this precise reason. Compared to graphic editing and presentation software, however, most wireframing software will help you more with staying consistent throughout the design process.

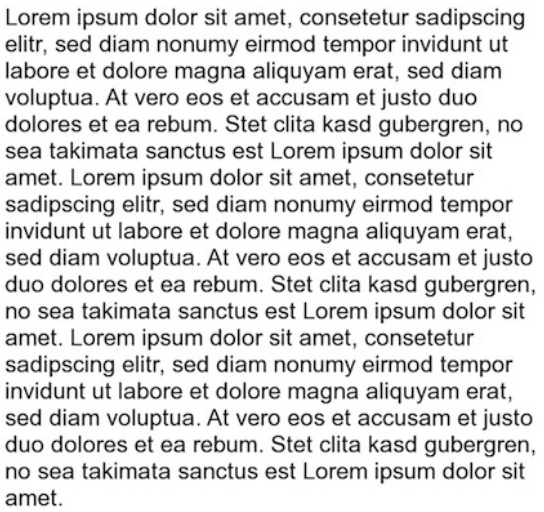
Low-fidelity doesn't mean unrealistic

When possible, add the details you know you'll need, even if they're rough approximations.

It's better than putting in arbitrary fillers you may change dramatically - which will ultimately change your design. One of the most common requests I get from designers when working together is content - real content. In 50 Shades of Wireframing, I emphasized the value of having higher-fidelity for text, colors, media, and interactions depending on which details will have the greatest impact on the overall product design.

At my last company, I had to actually think about whether my product summary paragraphs were going to be 3 or 6 lines long because the bulk of those paragraphs may change the overall look, feel, and emphasis on the page given the existing layout - and, therefore, the layout would have to be re-considered for the product design. If I'm including product photos, I have to see the photos - or at least an approximation - before I can decide whether to include it, make it a primary feature or fight against it. The truth is that headings wrap, videos can be distracting or even creepy, photo dimensions and orientations can be inconvenient, and decimals don't creep out of their input boxes.

So scrap the "Lorem Ipsum" and get realistic about what you're trying to convey to your users as early as possible.



Experiment & collaborate quickly

The best way to build effective products is to start small and iterate wildly. Iteration reduces risk. It makes big failures unlikely and turns small failures into lessons. This avoids the 200 page spec document which can turn into a bottleneck.

Low-fidelity design helps you and your team to explore many potential solutions quickly before focusing in on one solution and polishing and refining it into the final product. Initially, it may appear there are many solutions to a design challenge. However, you can only decide which will work best after exploring a few of them and laying them out in front of you - to see, touch and feel. Inexperienced designers will over-commit to their initial ideas and fall in love with what is likely the wrong solution - good designers know better.

Ultimately, you're working in the wrong fidelity if you can't product concepts quickly. And you're wasting precious time if your wireframes are just grayscale versions of a design you already have your heart set on. Use wireframing as a means to an end, not the end itself.

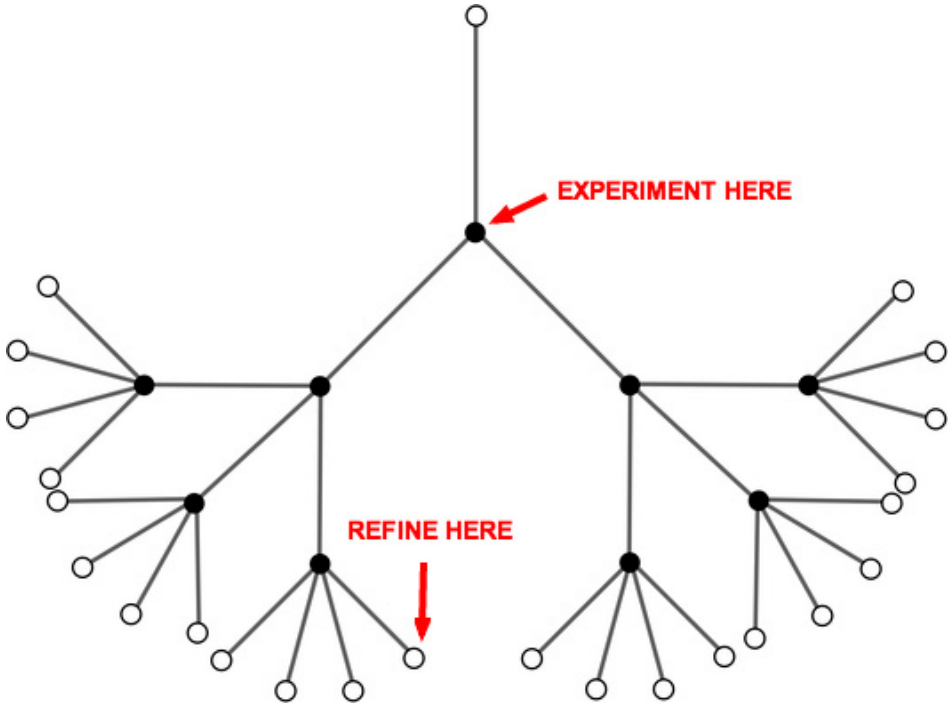


image: *Rosalind*

Your designs will get built

A great design can be a terrible solution.

Remember that every animation, box, button, menu, modal window, photo gallery, and map view needs to be programmed - and you can draw a lot faster than anyone in the world can code. If you know HTML/CSS/JS, and you've seen what it takes to test a page across every web browser, you'll try to find an easier way to solve your users' problems when designing wireframes. Every project has budgetary, time, and resource constraints, and you should internalize that with everything you add to your wireframes, even little components - there are no small changes, and there's a trade-off for every decision you make.

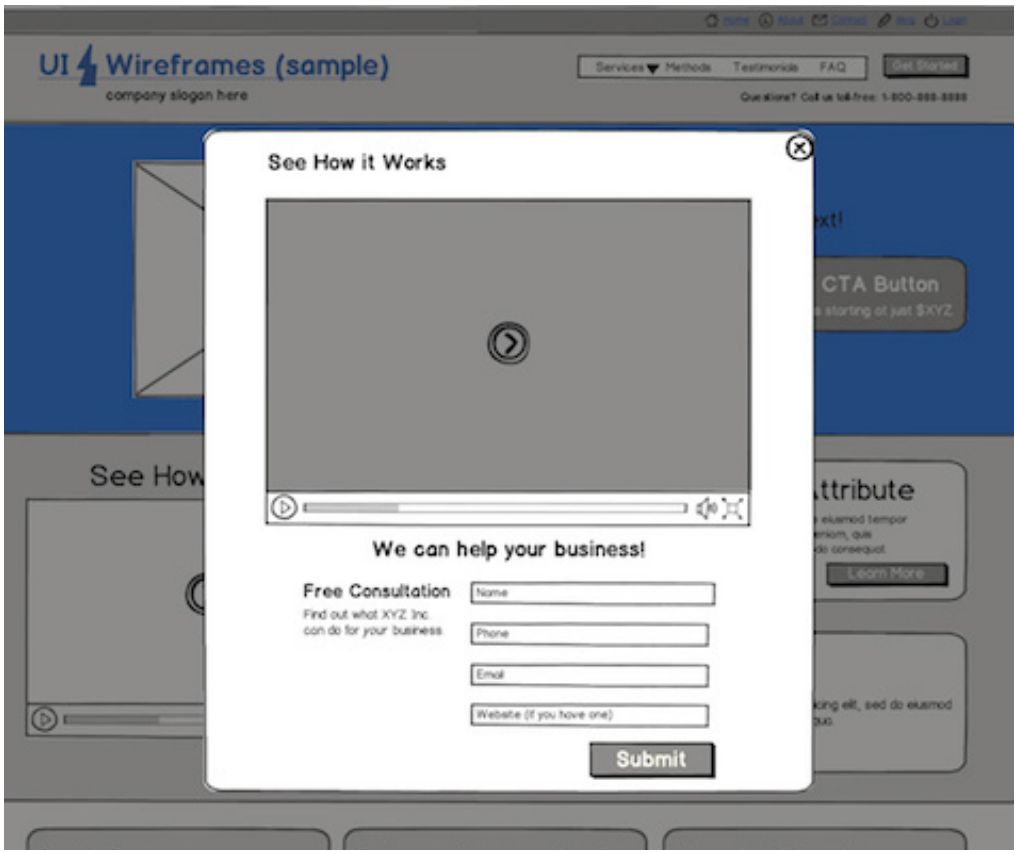


image: UIwireframes

Shipped is better than perfect

The goal of sketching, wireframing and prototyping is delivering great product concepts, not great deliverables.

Nobody cares if your deliverables are great if they're not finished. And few will care how perfect they are even if they are finished. The only people who may marvel at the deliverables are the UX designers, but they're hopefully too busy to care about internal documentation. On the other hand, everyone wants or absolutely needs you to communicate the right level of details about what they need to do to make a great and feasible solution so they can ship better products, faster.

If you've sketched something on scrap paper that you're confident is a solid solution that everyone can act on immediately, there's no value in re-creating it in a wire-frame or anywhere else. Just take a picture of it and share it where appropriate. In some cases, you may have to quickly replicate it for organizational purposes, but don't make deliverables for the sake of it - you have better things to do.

In 4 Non-Digital Wireframing Weapons and 4 Digital Wireframing Weapons, I cover a wide variety of tools you can use to get the job done.

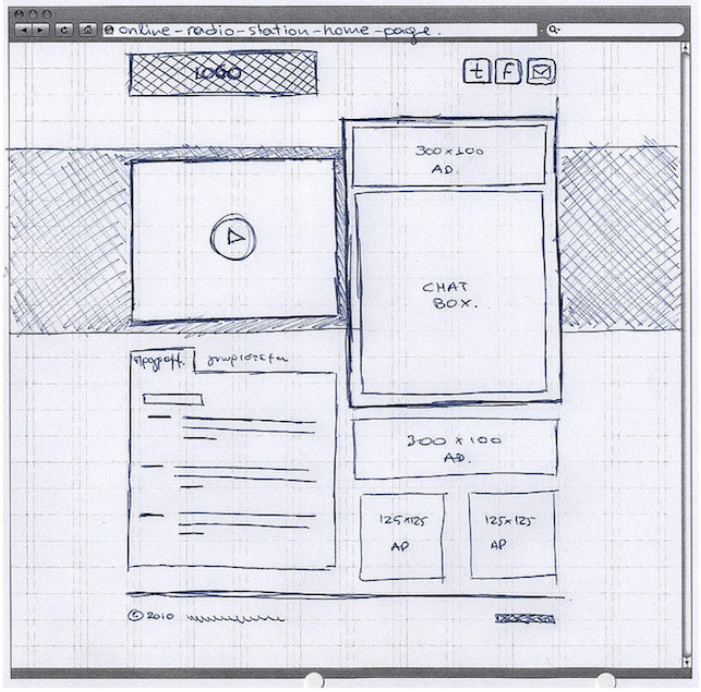
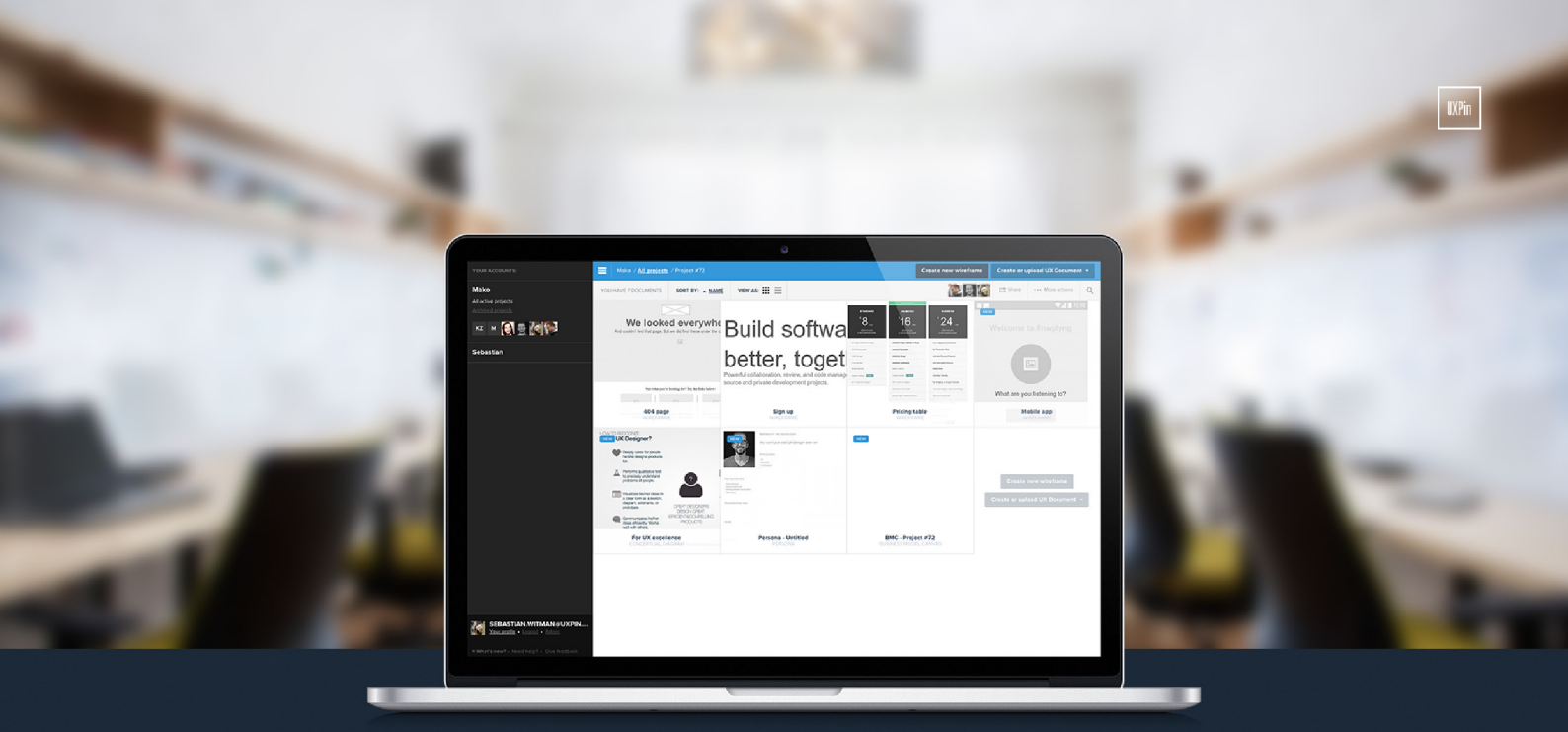


image: *Christos Chiotis*

Ready, set, wireframe!

I've done enough talking. And you need to get back to your product. So grab your favorite tool, and keep some of these in mind as you ready for your next release.



- ✔ The UX Design Platform
- ✔ Geared up for all wireframe styles and all fidelity levels
- ✔ Complete Prototyping Framework for Web, Mobile and Wearable
- ✔ Built for Teams

UXPin

www.uxpin.com