

MOBILE NAVIGATION: PATTERNS AND EXAMPLES



Navigation is known to be the backbone of any app - but how can we make it intuitive and seamless? What patterns are most popular? Read on to find out!

Mobile navigation design is all about getting users where they want to go, with the least amount of friction possible. There's a wide selection of tools that designers can implement in order to create an entire navigation system in their mobile app – but the best one to use, of course, is an app prototyping tool.

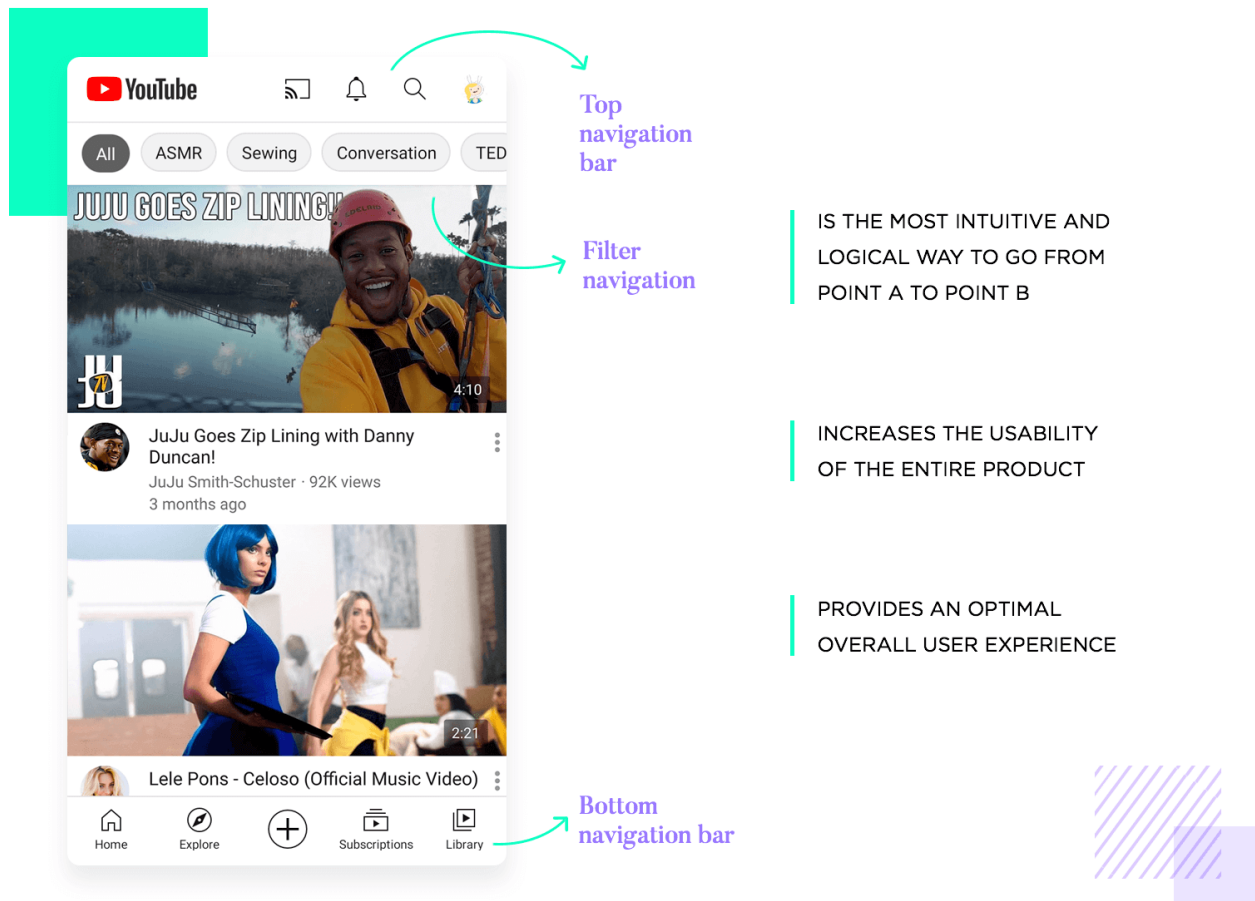
We put together a list of the most common UI components and patterns that designers all over the world use when prototyping their navigation. Read on to find why they work and the things each one does best! As a bonus, we also included a list where these components can be seen in action in a section of serious UI design inspiration.

What is mobile navigation?

There are so many ways to describe navigation in mobile apps and how important it is. The navigation is the way your users will get from point A to point B. It's how they'll discover the design and interact with the product. Designers and writers have described navigation as the road system of an app, because they are the highways that users need to enjoy the product.

The navigation design of any product is crucial. While it's tempting to think of the goal of the navigation as “getting from A to B in the least amount of time possible”, that would be a mistake. Your navigation isn't about getting users there quickly – it's about making the way there logical and easy.

What is mobile navigation?



Good mobile navigation will boost the usability of the entire product, helping users to enjoy all the features offered. Bad navigation will make it difficult to find things, making it less likely that users will ever experience the product the way the design team had envisioned. This is the kind of thing that can make or break any mobile app – a life and death kind of situation. That is how important the navigation design is.

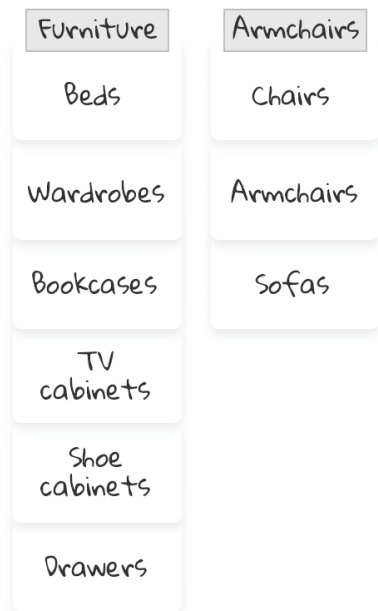
Best practices for mobile navigation

1. Navigation needs to be evident and intuitive

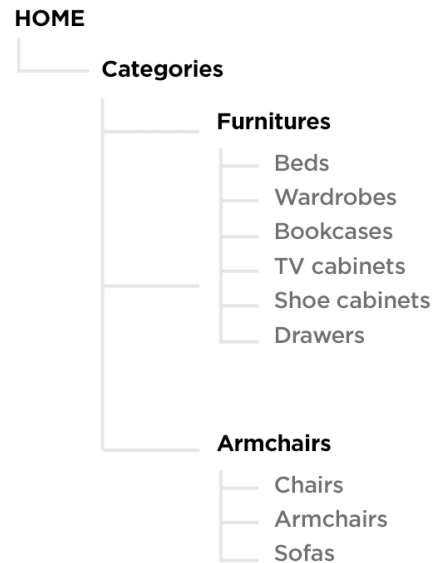
This is true for all sorts of navigation, it's true. However, everything is made more noticeable in mobile devices because of the reduced space and increased interaction cost. That means that any mistake made in the navigation of an app can have a bigger impact on the experience than a web-based app.



Card sorting



Tree testing



Users don't want to wonder what buttons do or where links will go. From a usability point-of-view, it's crucial that the navigation is intuitive to your target users. This means rigorous testing and lots of research. You want to understand users' mental models, and carry out tree tests and card sorting tests. This is the time to pull out all stops and create a mobile navigation system that works with users instead of against them.

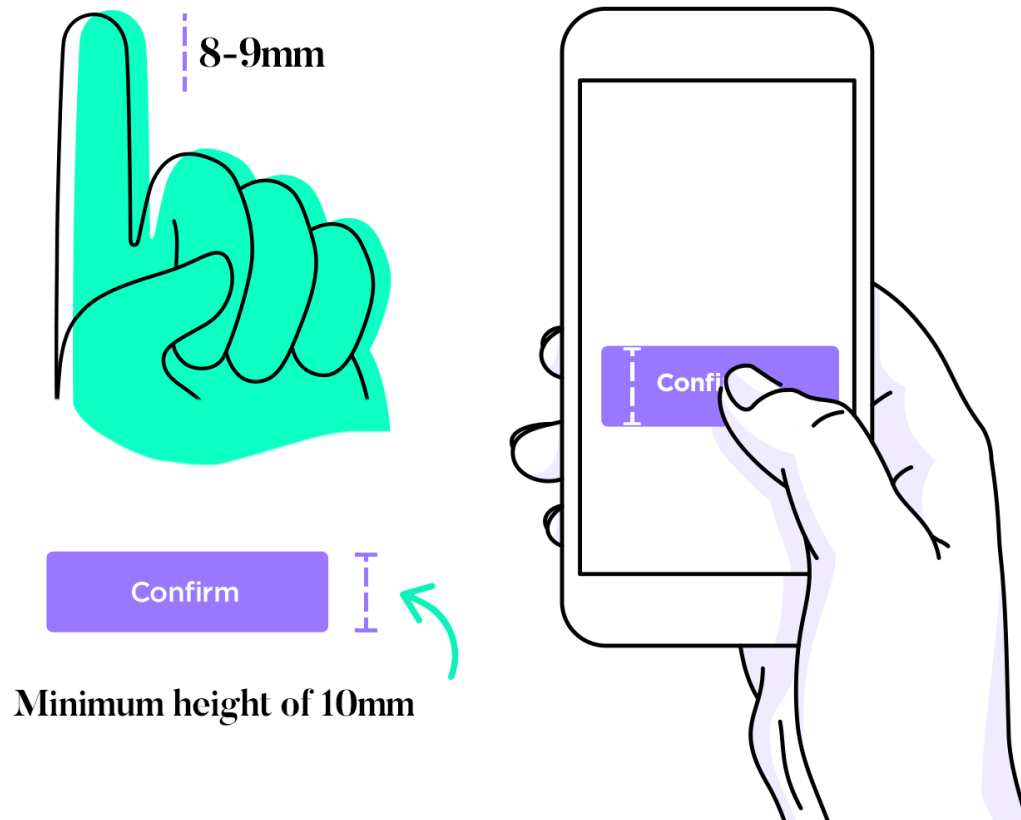
Even after the bare bones of the navigation design are put together, you'll still want to double-check that it truly is intuitive. From low to high-fidelity prototypes, this type of validation is standard procedure.

The name of the game here is understanding what users expect and removing any friction in the navigation patterns.

2. Account for finger and hand positioning

This is of vital importance for mobile apps. Users' fingers aren't always slim but everyone has to be able to use the app comfortably. Nobody wants to waste time by trying to tap on an icon repeatedly and not being able to get anywhere. It's frustrating, distractive and can ruin the carefully planned experience the design team had worked so hard to create.

This means that your links and buttons need to be big enough for most people to be able to tap them successfully on the first try. The general advice is to install a minimum 10mm size to all buttons. To know more about creating wonderful buttons, check out our post on [button design](#).



Another thing to consider is the spacing between elements that support the navigation. You want to make sure not just that everything is big enough so users won't stress themselves out when they try to tap them. You want available space so they don't tap on random elements that are nearby.

3. Content needs to be legible

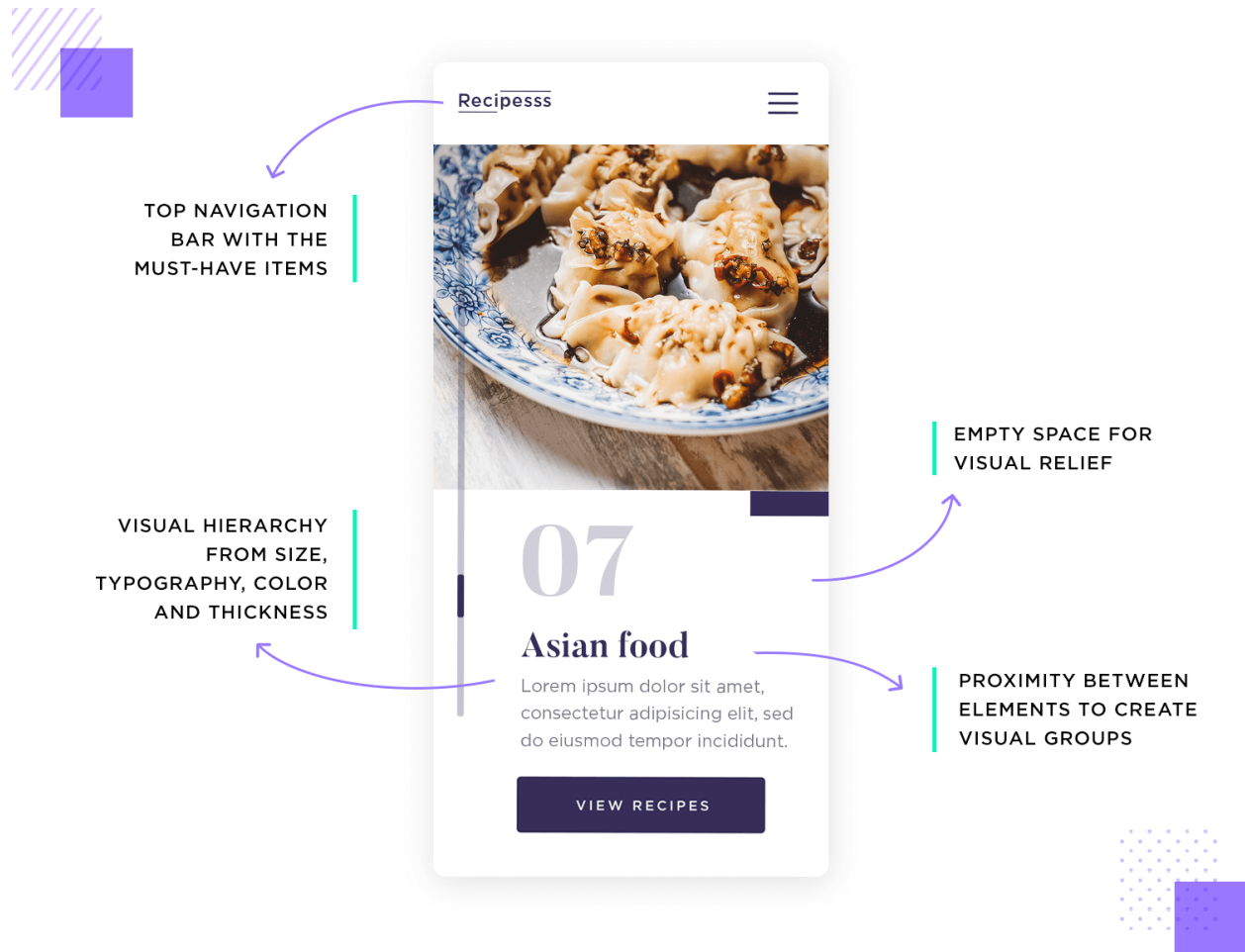
Mobile screens are much smaller than computers – which creates the issue of readability for the content at play. The navigation in most apps is text-based, even if they rely on icons like the hamburger menu icon for some of it.

You want users to not have to struggle to read anything, from the actual content to the links and button texts. When it comes to mobile navigation, size matters! It's worth taking the time while you're still prototyping the navigation to test it out and see how users interact with it. This is a solid reason to invest in a [prototyping tool](#) that allows for realistic navigation simulation and can work with your favorite user testing platforms.

4. Avoid clutter, respect visual hierarchy

Small screens mean that it's much easier to fall into the clutter trap. Mobile app design requires designers to forsake anything that isn't absolutely crucial for the user experience, trimming any

extra fat the design might have. Minimalism may be a specific style of UI design, but all designers dabble with it when it comes to creating mobile apps.



Even with a small number of elements, users can still get a feeling of chaos if the elements don't have a visual balance. It's very important that the UI layout, the content and even the navigation work together to guide the user's eyes around the screen. This includes having the content maintain a certain hierarchy, analyzing the proximity between elements and allowing plenty of empty space for visual relief.

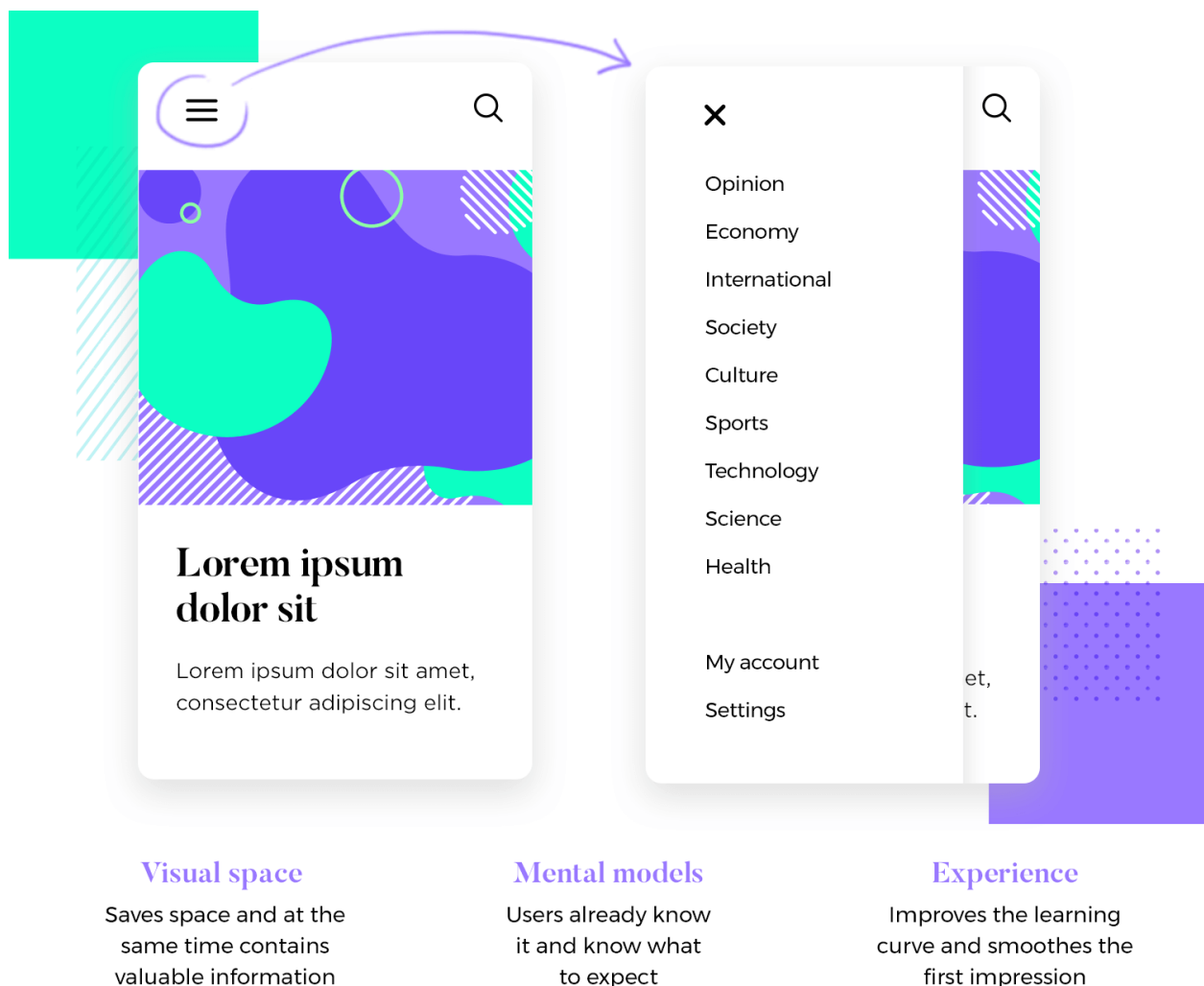
8 UI COMPONENTS FOR MOBILE NAVIGATION APPS

1. The hamburger menu

The infamous hamburger menu. Some designers love it, others simply tolerate it (and a small number want to eradicate it from existence).

Whatever your thoughts are on the humble hamburger menu, those three little horizontal lines which sit comfortably in the corner of a mobile app can be extremely useful. With such a limited amount of real estate to play with, the hamburger menu is a method of hiding more elaborate navigation so users can enjoy more screen space.

Like so many other UI components that designers have come to rely on, the hamburger menu brings another great advantage to the design. Most users are already familiar with it. They know what it is, how it works and what to expect from the icon.



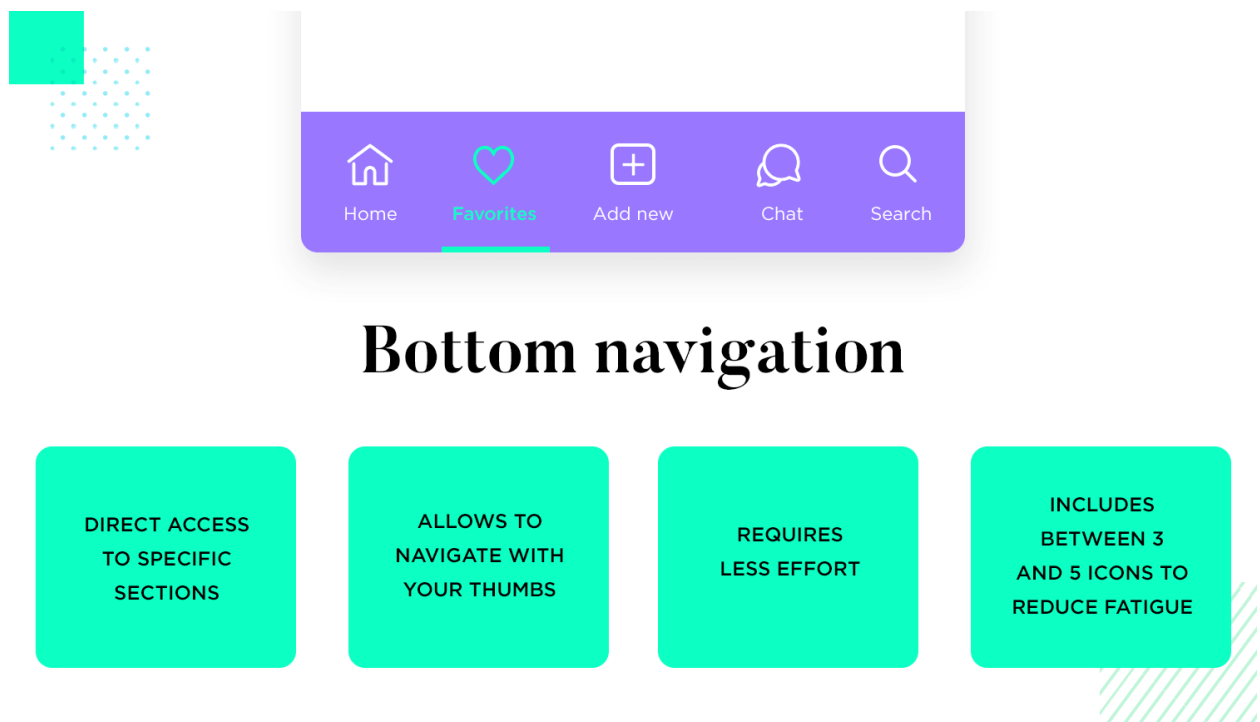
This may seem like a rather small detail, but having a familiar face in a brand new product can make a big difference in the user experience. This is especially true for those vital first moments, where users must learn their way around.

Of course, everything has its downsides. With the hamburger menu, some designers say that in hiding away the navigation options we may hinder the user experience instead of improving it. Some arguments against it include that users may not notice it unless they actively look for it in the UI. Others focus on the fact that in hiding away crucial links and buttons, we may make it more difficult for the user to discover the product upon first encounter.

While there's still a debate surrounding the hamburger menu, most in the design community have surrendered to its charms. The hamburger icon has become so popular, most designers will be at ease simply using a good color scheme that ensures a contrast between icon and background.

2. Bottom navigation

Bottom navigation is exactly that – a navigation bar that gathers the primary or secondary navigation links. With a simple tap, users can intuitively explore and negotiate between top-level views.



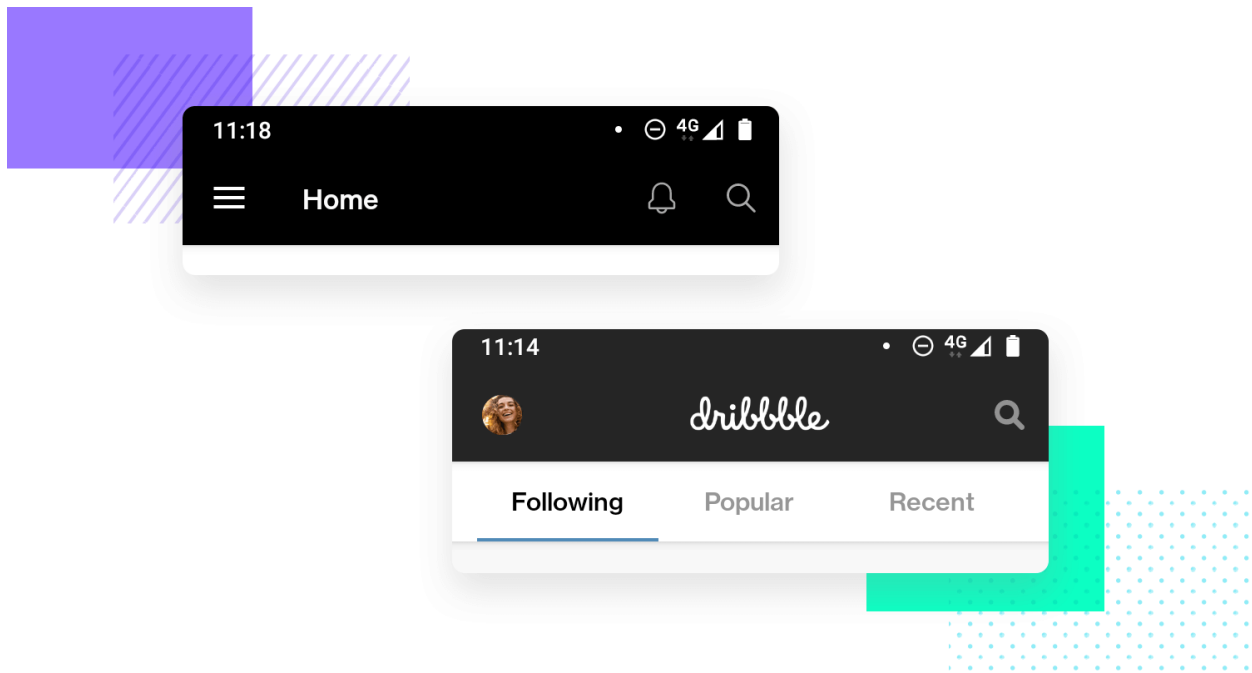
The reason why bottom navigation is so popular for mobile apps is that most users will be able to navigate with their thumbs comfortably while holding their device. It requires less strain and reduces the need to change the method of grasping the device, improving the usability of the entire product.

To reduce fatigue, most bottom navigation bars have between three and five destinations for quick access. Another argument in favor of bottom navigation is that it takes less effort to reach certain destinations within the design, with only a single tap being needed.

Discover some [presentation techniques](#) to present your mobile navigation to stakeholders in the best possible light.

3. Top navigation

The other side of the same coin as bottom navigation. This also consists of a navigation bar, only located at the top of the screen. It still offers most of the benefits from bottom navigation, minus the ease of use while holding mobile devices. With larger phones, many users will have to use both hands or change their grasp in order to reach all the links.

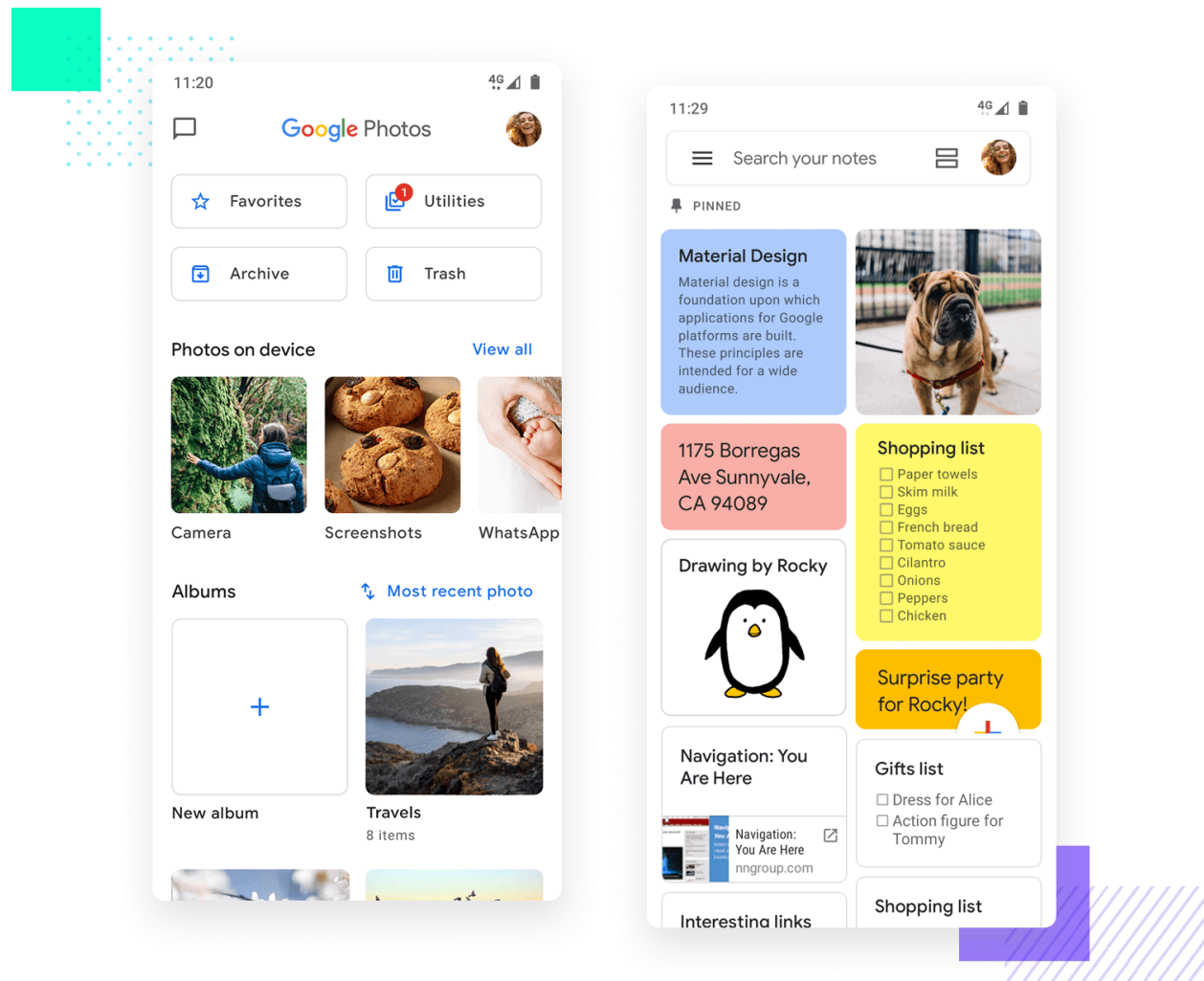


Like many components on this list, top navigation is often used in combination with other ways of navigation. A silly but practical example might be when mobile apps use both top navigation (for primary links) and secondary navigation (for secondary links). The top bar can convey to users that the links it contains are very important, especially if they remain visible as the user scrolls down.

4. Cards

[Card UI design](#) is a brilliant design pattern and can really make your mobile UI pop as a result of their highly visual and customizable nature. They come in all shapes and sizes and are a great way to showcase various elements such as text, a link, or a photo in one place and have become very popular in mobile app navigation.

As content has become more compartmentalized and personalized over the last few years, cards have been proven to be a great way to aggregate individual pieces of information in one place.



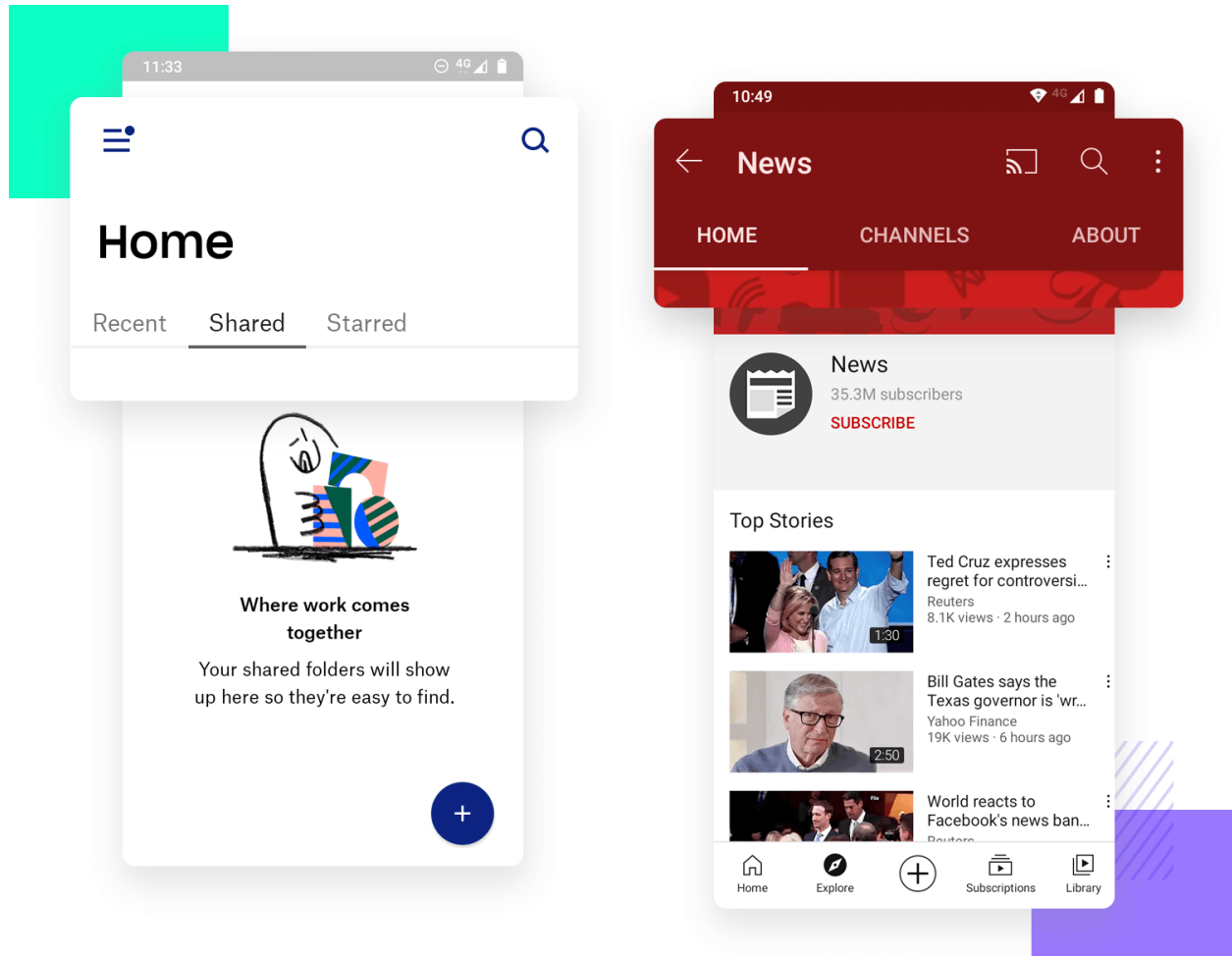
In an effort to improve your UX design, cards can be personalized to display different content. Another upside is that cards can help users notice immediately when the cards can be scrolled horizontally, improving general usability. Yet another argument in favor of cards is that they can be easily adapted to different screen sizes, making them a great option for responsive apps.

5. Tabs

Tabs are not too different from navigation bars when it comes to UI patterns. They share the same layout, that is to say, a tab will be a row of multiple options leading to different screens. Where they differ, however, is a navigation bar will contain options that are unrelated to each other. Tabs, on the other hand, tend to have an overlying theme to them.

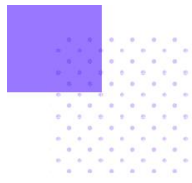
For example, in a navigation bar, you might have a home button. Next to the home button may be a search function or even a favorites button which will take you to those respective views. While unrelated, these buttons represent key features of the app.

Essentially, tabs are used to switch between alternative views within the same context. To illustrate, consider a tab system for an email platform. The first tab shows “primary” senders who are in your contact list, while the second shows other senders who you may not know. Two different sides of the same coin that deal with incoming emails.



6. Gesture-Based Navigation

Gesture-based navigation enables users to quickly swipe in their desired direction to navigate through an app or perform a particular action. As a UI navigation pattern, it has been around for decades but gestural navigation really gained traction among mobile app users with the advent of popular dating app Tinder. Swipe right, anyone?



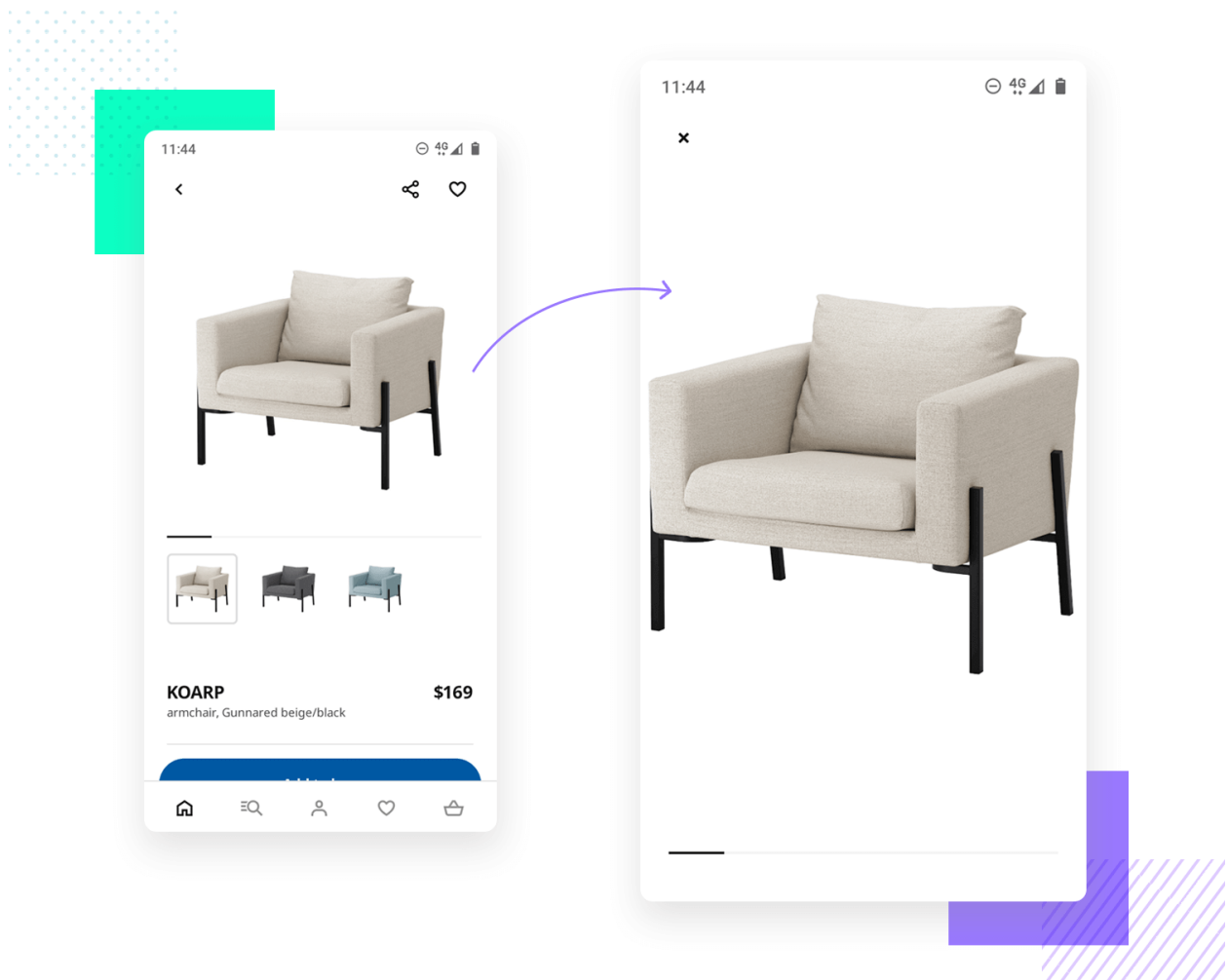
Using gestural navigation helps to create a journey between different scenes within an app and gestures include touching and dragging, both horizontally and vertically, as well as zooming in and out. It's highly immersive and interactive, creating a dynamic experience.

The great thing about this UI design pattern is that it's relatively simple to grasp for even the most inexperienced user as the gestures are often intuitive and only require a little experimentation to get right.

7. Full-screen navigation

Many of the navigation patterns on this list are all about minimizing the use of space. This is a completely opposite approach, dedicating the majority of the screen exclusively to mobile navigation. Sometimes referred to as a “navigation hub”, this approach works well to funnel users from broad sections of the product to very specific ones.

It's true that sometimes, so much navigation can be a little overwhelming. Some designers are adventurous enough to dedicate an entire screen to the app's navigation, using careful visual hierarchy to avoid overwhelming users. It's a way to offer a lot of the navigation options in a coherent way, immediately helping users understand the product's features all at once.



8. 3D touch

And last but not least, we have 3D touch. It was first brought to users by Apple, who found a way to offer direct options from the homescreen of the iPhone. It's a way to create a navigation shortcut, showing the key actions for the selected app.

Another common use for this mobile navigation is the preview of content. When dealing with a list of options for content, like an email inbox or a list of articles, this can be a good way to give the user a preview.

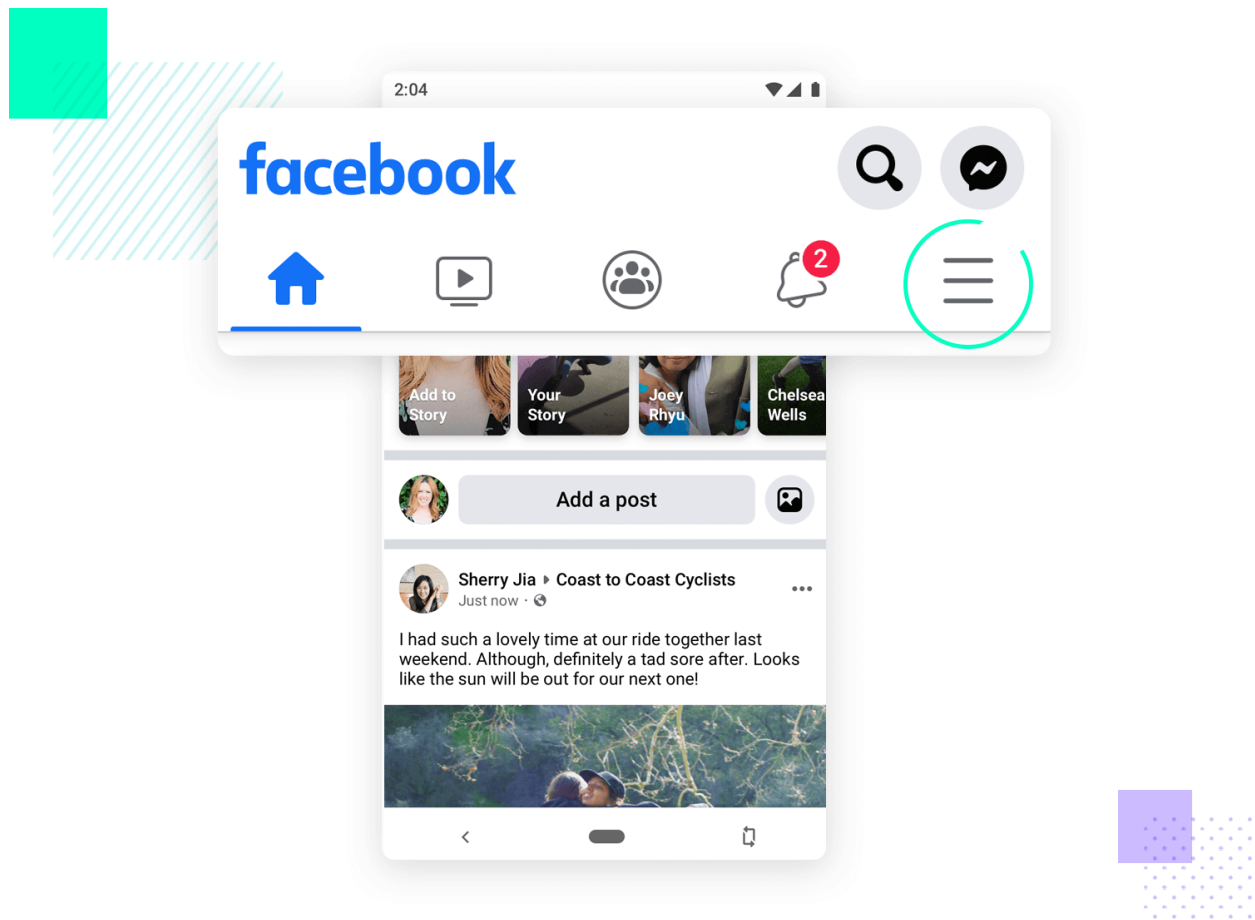


It's worth remembering that central features shouldn't be accessible only via 3D touch. Mobile navigation should still provide a clear path so users can find the primary features without having to discover the 3D touch option.

10 great mobile navigation examples

1. Facebook

Facebook is known for many things – one of which is its mobile navigation. This great example shows the power of combining different mobile navigation patterns. Facebook uses a hamburger menu, a top navigation bar as well as a bottom navigation bar.



It's true that, in theory, this all sounds like a bit too much – but Facebook makes it work. By now, most users know exactly what goes where in the three navigation options, without anyone getting overwhelmed. With this approach, the social media giant manages to clearly separate the primary things from the secondary.

The platform uses 5 destinations in its navigation bar below. These buttons are persistent, which means they don't change and are always present, creating consistency across the app.

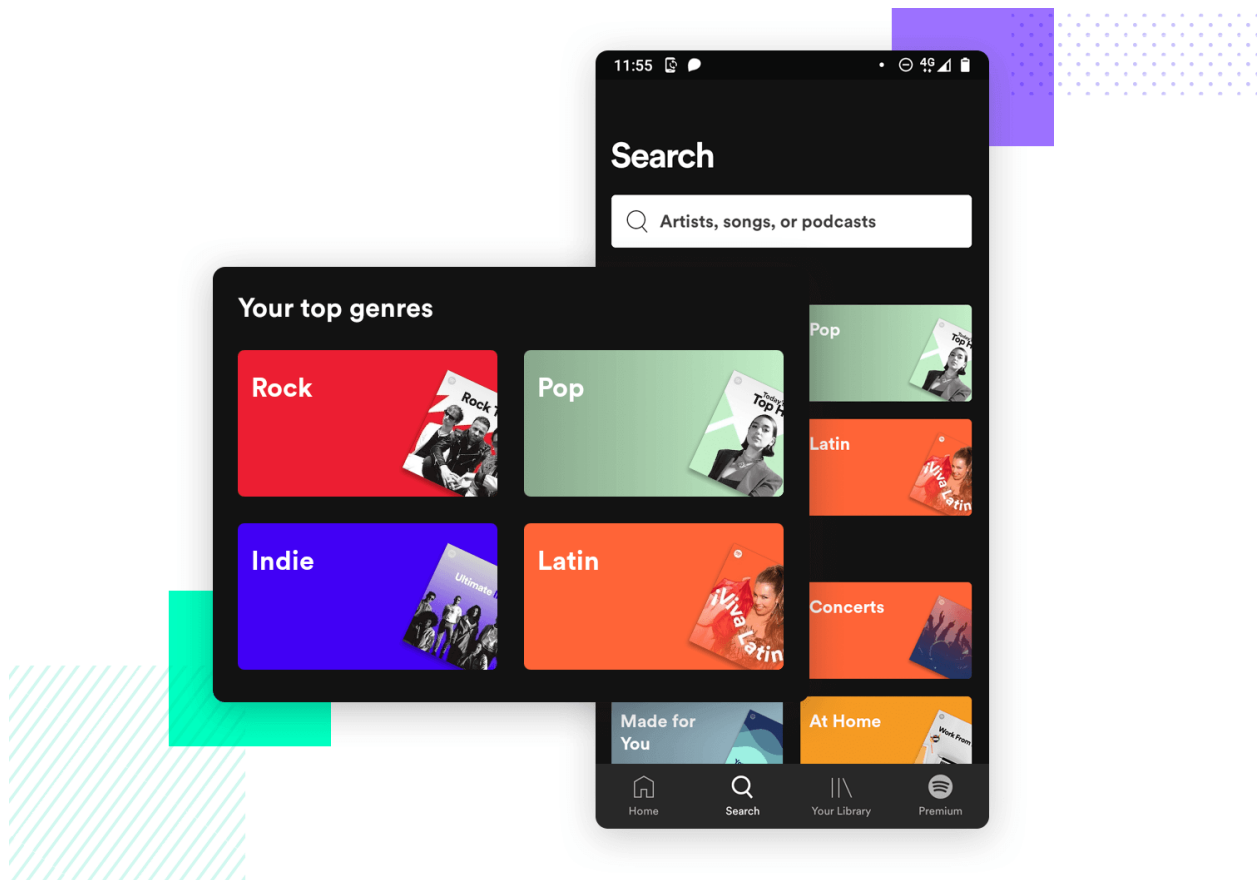
2. Apple's 3D touch

What better mobile navigation example than the folks who first invented 3D touch? Apple created a whole new type of shortcut for mobile phones, managing to offer great usability at the same time. It's true that it comes with some limitations but it's still an impressive way to offer navigational options to users.



3. Spotify

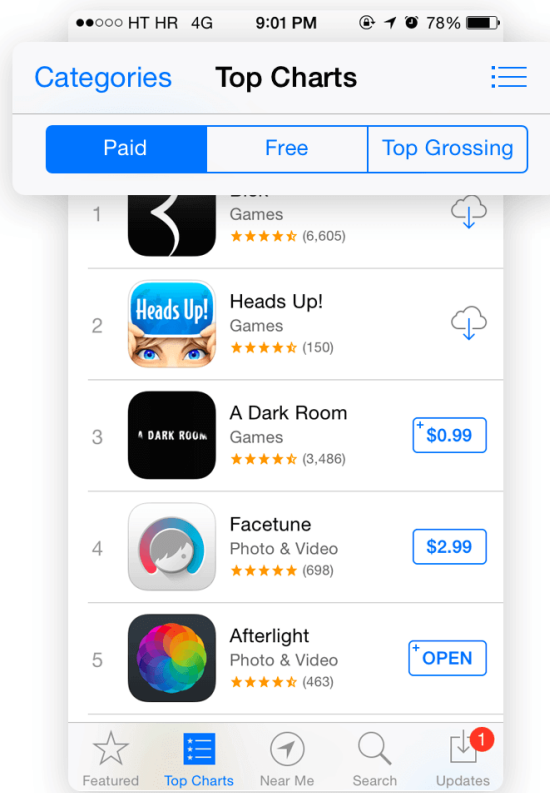
Spotify has grown to be a giant in the music streaming business, and their UI design reflects it. This mobile navigation example is all about cards in the homepage, using them as a dynamic way for users to discover content. Spotify is known for encouraging users to discover new musical horizons, which improves the entire user experience.



The cards are done in a highly visual way, highlighting the key idea behind each card. As a bonus, we also have a bottom navigation bar that does the heavy lifting when it comes to the main navigation of the app. Overall, it's easy to understand and explore even for new users.

4. The App Store's tabs

The App Store is a wonderful example of using tabs for mobile navigation. We can see that it reflects the theory behind navigation design perfectly, with the tabs touching on different aspects of the same content. Within the Top Charts section of the App, a user has the ability to explore Paid, Free and Top Grossing tabs – creating coherency and awareness of alternative views.



The bottom navigation bar covers the more primary features of the entire store, creating a highway that gets users to where they need without an overall theme to the buttons. Classic mobile navigation art!

5. Telegram

This is a great example of a humble hamburger menu at work. Telegram has gotten a lot of attention lately both for its treatment of user's personal data and its easy UI design. The app can be used by anyone, with the hamburger menu offering all the key navigation options a user may need.



This is a perfect example of the strength of the hamburger menu. Everyone recognizes it immediately, with the expanding menu being extremely easy to find, use and understand. All of this, with a simple icon that barely takes up precious space in the interface.

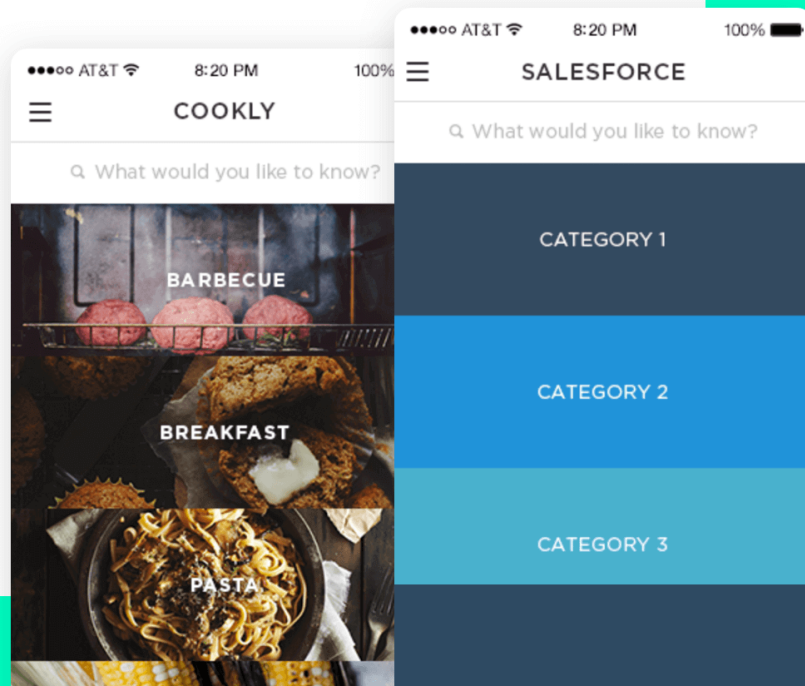
6. Tinder

Oh, Tinder. We all know of it, with the app making a huge impact on how people think of online dating. Gesture-based navigation wasn't invented by Tinder, but the app definitely brought it to the masses. The classic swipe left or right pattern is all about keeping things dynamic, easy and entertaining. No matter how you feel about the app, there's definitely merit to its mobile navigation design!



7. Cookly

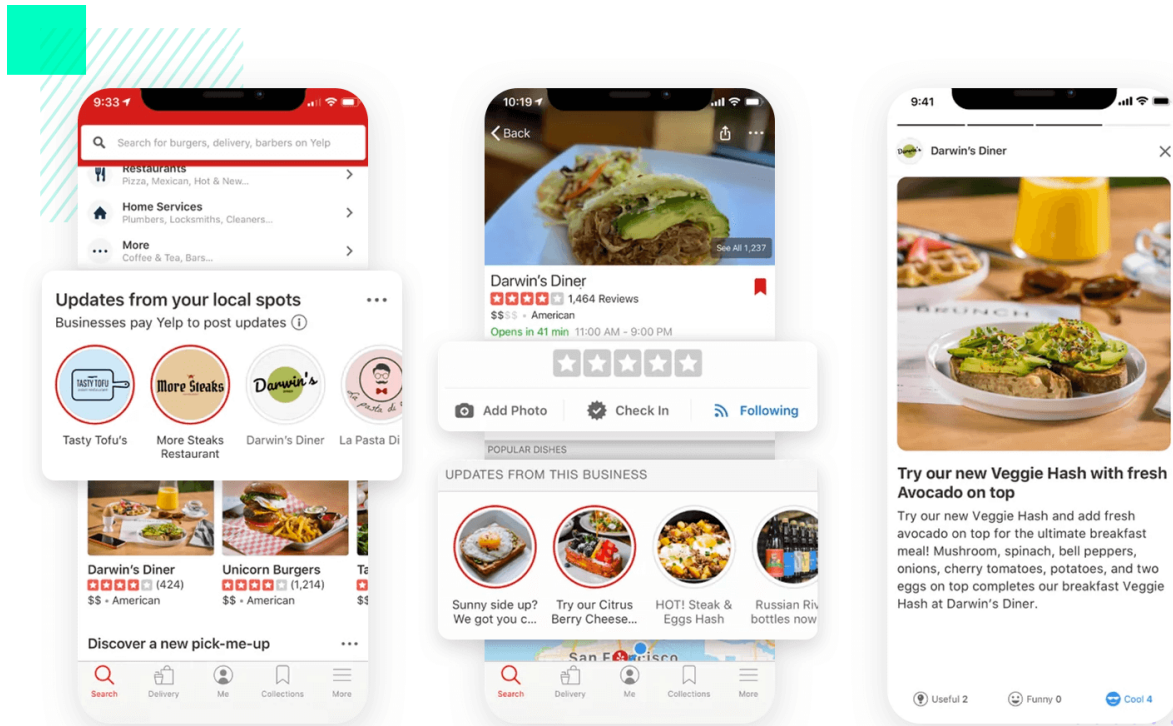
Brought to you by [Patrick Wong](#), this is a great example of full-screen navigation at work. In fact, we can appreciate that the “navigation hub” pattern can be adapted to suit different needs. Any way you want to go about it, Patrick’s work goes to show that this approach can render a wonderful user experience when done right.



For us here at Justinmind, a good navigation hub is all about spacing, good visual hierarchy and usability. These three aspects can be thought of as pillars of the hub, making it possible for users to process and interact with the screen – opening up the entire product in a single swing.

8. Yelp

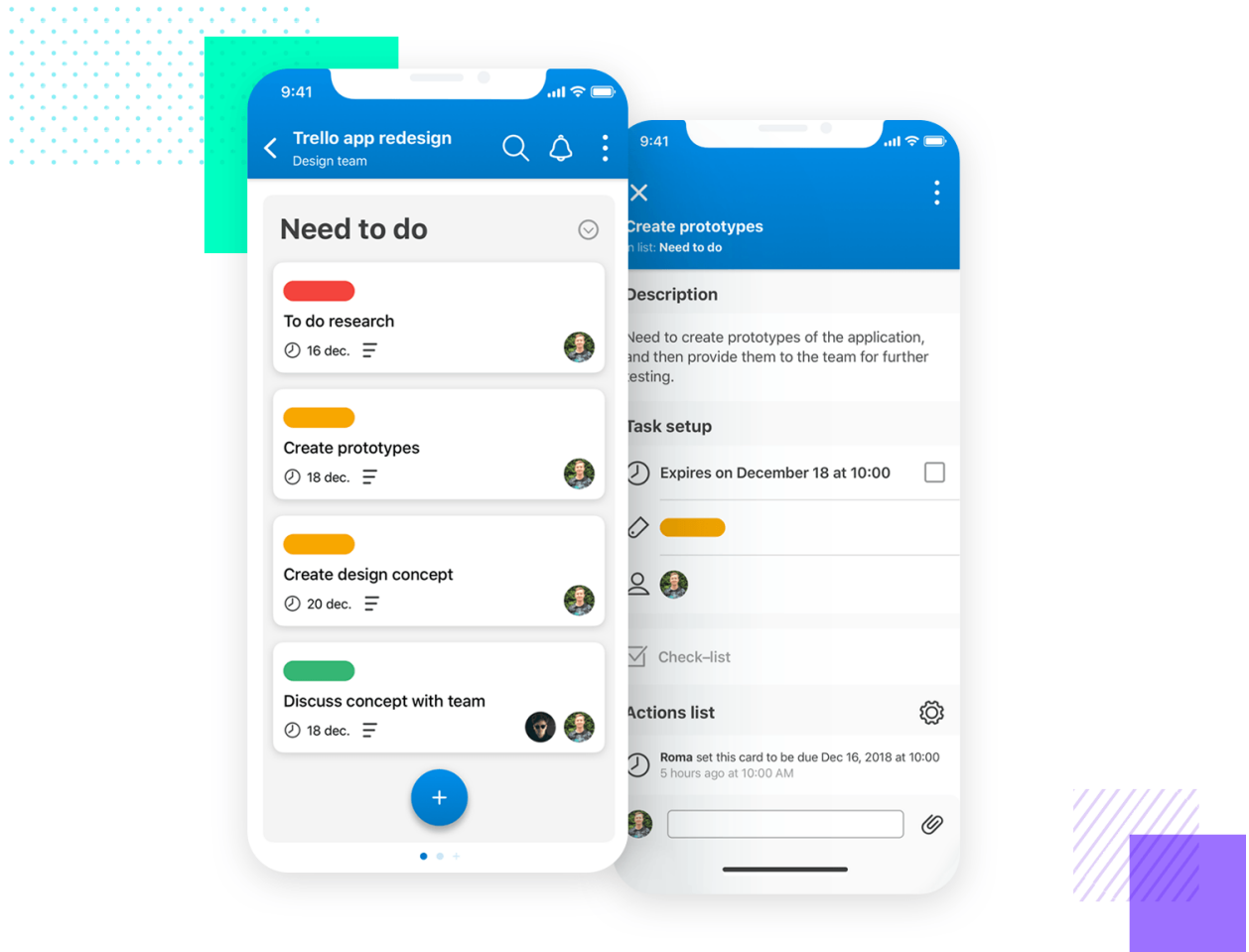
Yelp takes a slightly different approach when creating a full-screen navigation experience. Dividing the screen into two different areas, Yelp doesn't actually dedicate the entire screen to the navigation options, using the top part for navigation options and leaving more negative space at the bottom.



We like that the platform uses the navigation options to funnel users into the category they're truly interested in, making the road dynamic and easy. For example, for users looking for restaurants, there's no real need to take the time to type the "restaurants" category.

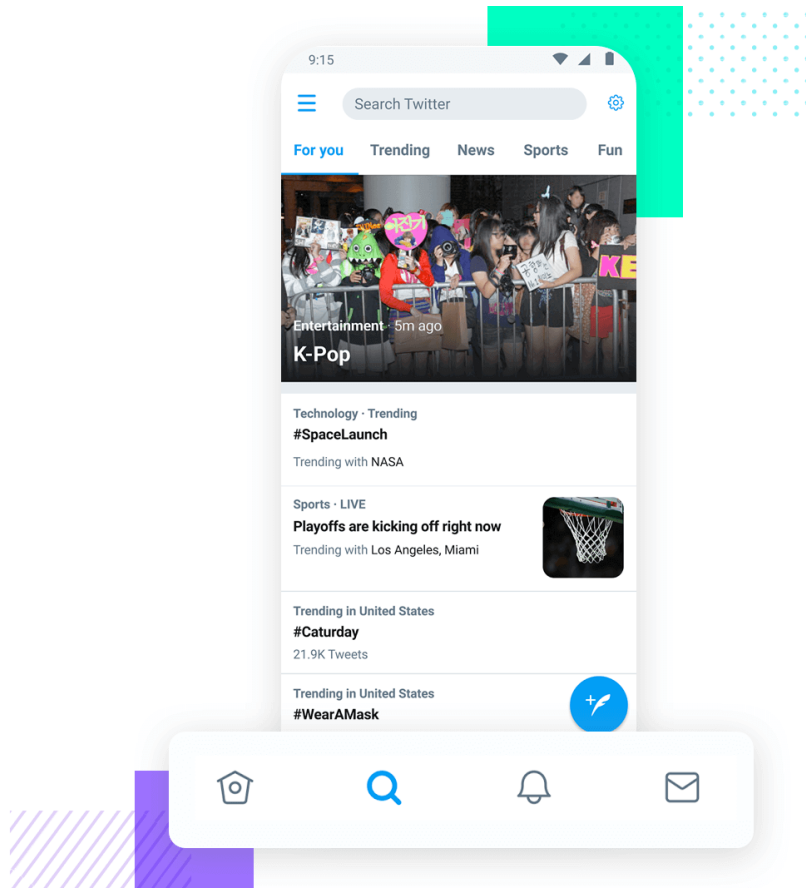
9. Trello

This Trello concept design by Roman Yurchenko is a great example of card-based mobile navigation. The cards aren't squares but rather elongated rectangles that fill up the screen space without overwhelming users. The card system is a good representative of Trello's ability to simplify complex workflows, keeping everything tidy and easy for the people involved.



10. Twitter

Twitter's mobile app is another good example of more than a single navigation pattern at work. The app focuses its navigation on the bottom bar, covering the four main aspects of the entire platform. It's broad and covers all the relevant ground.



Mobile App Development Lifecycle

In contrast to the 7 steps of the software development life cycle, the app development lifecycle is much shorter. There are generally five primary parts to the procedure, as there are with those:

- i. ***Inception*** - Every software begins with a concept. Typically, such a concept is evolved into a strong foundation for an application.
- ii. ***Design*** - The design phase include establishing the app's User Experience (UX), such as the overall layout, how it operates, and so on, as well as translating that UX into a good User Interface (UI) design, which is generally done with the assistance of a graphic designer.
- iii. ***Development*** — The actual creation of the application is usually the most resource-intensive step.

- iv. ***Stabilization*** — Once development has progressed far enough, QA generally begins testing the program and fixing issues. Frequently, an application will enter a restricted beta phase in which a larger user population is given the opportunity to try it out and provide feedback and suggestions for improvements.
- v. ***Deployment.***

Many of these parts often overlap; for example, programming may take place while the UI is being finished, and it may even inform the UI design. Additionally, a program may be in the process of stabilizing while new features are being added to a new version.

These stages may also be utilized in a variety of SDLC techniques, including Agile, Spiral, and Waterfall.

The next sections will go through each of these phases in further depth.

1. Inception

Because of the widespread use of mobile devices and the high degree of engagement they generate, practically everyone has an idea for a mobile app. Mobile devices provide a completely new method of interacting with computers, the internet, and even business infrastructure.

The ideation stage focuses on identifying and improving an app's concept. It's critical to consider certain fundamental questions before developing a successful app.

Before submitting an app to one of the public App Stores, keep the following points in mind:

- Are there any similar apps presently on the market? If that's the case, how does this software differ from others?

In the case of apps that will be distributed within an organization:

- ***Infrastructure Integration*** — How will it work with or enhance existing infrastructure?

Apps should also be assessed in the context of the mobile form factor:

- What value does this app provide to its users? What will they do with it?

- **Form/Mobility** — How will this app function on a mobile device? How can I use mobile technologies like location awareness, the camera, and others to offer value?

Defining Actors and Use Cases might be helpful when creating the functionality of an app. Actors are users who play a part in an application. Actions or intentions are commonly used as use cases.

A task tracking application, for example, may include two Actors: User and Friend. A user may make a task and then share it with a friend. In this scenario, generating a task and sharing a task are two unique use cases that will dictate what screens you'll need to construct, as well as what business entities and logic you'll need to develop, in conjunction with the Actors.

It's considerably easier to start building an application once you've gathered a sufficient number of use cases and actors. The development team may then concentrate on how to build the app rather than what it is or should do.

2. Design

After determining the app's features and functioning, the following stage is to try to address the User Experience or UX.

2.1.Designing a user experience

Wireframes or mockups, created with one of the various design toolkits, are typically used to create UX. UX mockups help you to develop the user experience without having to worry about the UI design:

Wireframes or mockups, such as those created with Balsamiq, are commonly used in UX.

It's critical to consider the interface rules for the various platforms that the app will target while generating UX mockups. On each platform, the software should "feel at home." The following are the official design guidelines for each platform:

- ✓ Human Interface Guidelines for Apple
- ✓ Design Guidelines for Android
- ✓ UWP - UWP Design Fundamentals

For instance, each program has its own metaphor for toggling between parts. iOS utilizes a bottom-of-the-screen tab bar, Android uses a top-of-the-screen tab bar, and UWP uses the Pivot or tab view.

In addition, UX decisions are influenced by the hardware. Because iOS devices lack a real back button, the Navigation Controller metaphor is used instead:

In addition, the form factor has an impact on UX considerations. A tablet has a lot larger screen real estate, thus it can show a lot more information. What requires many screens on a phone is sometimes squeezed into a single screen on a tablet:

Due to the variety of form factors available, you may also wish to target mid-size form factors (those that fall midway between a phone and a tablet).

2.2.Designing a user interface (UI)

The next stage is to construct the UI design after the UX has been chosen. While most UX mockups are black and white, the UI Design phase is where colors, graphics, and other elements are added and finished. It is critical to devote attention to decent UI design, and most popular applications have a professional appearance.

It's crucial to remember, as with UX, that each platform has its own design language thus even a well-designed app may seem different on each platform:

3. Development

Typically, the development period begins quite early. In reality, once an idea has matured in the conceptual/inspiration phase, a functioning prototype is typically created to check functionality, assumptions, and to assist define the scope of the project.

4. Stabilization

The process of sorting out the issues in your software is known as stabilization. Not only in terms of functionality, such as "it crashes when I click this button," but also in terms of usability and performance. Stabilization should begin early in the development process to allow for course adjustments before they become expensive. Prototype, Alpha, Beta, and Release Candidate are

common phases for apps. Different individuals have different definitions for these, but they all follow the same pattern:

- **Prototype** — The app is still in the proof-of-concept stage, with just fundamental functionality or select elements of the app functional. There are a lot of bugs.
- **Alpha** — The majority of the core functionality is code-complete (built, but not fully tested). Major issues still exist, and outlying functionality may not be available.
- **Beta** — The majority of the functionality is now complete, and it has undergone some light testing and problem fixes. Major flaws that have been identified may still exist.
- **Release Candidate**: All functionality has been tested and is complete. The software is ready to be released into the world, barring any additional issues.

It's never too early to start testing a new app. If a big issue is discovered during the prototype stage, for example, the app's UX can still be updated to suit it. If a performance issue is discovered at the alpha stage, it is still possible to change the design before a large amount of code has been written based on incorrect assumptions.

As an application progresses through its lifespan, more individuals are invited to check it out, test it, offer comments, and so on. Prototype apps, for example, may be exhibited or delivered solely to key stakeholders, whereas release candidate applications may be released to clients who sign up for early access.

Typically, delivering directly from a development computer is adequate for early testing and distribution to a small number of devices. This, however, might rapidly become inconvenient as the audience grows. As a result, there are a variety of test deployment alternatives available that make this process much easier by allowing you to invite individuals to a testing pool, distribute builds over the web, and give feedback tools.

App Center may be used to continually develop, test, release, and monitor apps for testing and deployment.

5. Distribution

It's time to release the app into the wild after it's been stabilized. Depending on the platform, there are a variety of distribution alternatives.

Apps for iOS and Objective-C are deployed in the same way:

- i. ***Apple App Store*** - The App Store, which is integrated into Mac OS X via iTunes, is a worldwide accessible online program repository. It's by far the most common form of app distribution, allowing developers to sell and distribute their programs online with minimal effort.
- ii. ***In-House Deployment*** — This method is used for the internal distribution of company programs that aren't published on the App Store.
- iii. ***Ad-Hoc Deployment*** — Ad-hoc deployment is used for development and testing, and it allows you to deploy to a small number of fully supplied devices. Ad-hoc deployment is what happens when you use Xcode or Visual Studio for Mac to deploy to a device.

❖ **Android**

Before being deployed, all Android applications must be signed. Developers use their own certificate, which is protected by a private key, to sign their apps. This certificate can establish a chain of trust between an application developer and the apps he or she has created and distributed. While an Android development certificate may be signed by a recognized certificate authority, most developers choose to self-sign their certificates rather than use these services. Certificates are mostly used to distinguish between different developers and apps. This information is used by Android to aid in the enforcement of permission delegation between programs and components running on the Android OS.

Unlike other successful mobile platforms, Android promotes app distribution in a more open manner. Devices are not restricted to a particular app store. Instead, anybody may open an app shop, and most Android phones support software installation from third-party stores.

This provides developers with a potentially bigger, but more complicated distribution route for their apps. Although Google Play is the official app store of Google, there are numerous others. Among the most well-known are:

- ✓ Handango
- ✓ AppBrain

- ✓ Amazon
- ✓ App Store for Android

❖ UWP

Users can get GetJar UWP apps from the Microsoft Store. Developers submit their apps for review, and once approved, they appear in the Store. See the UWP Publish page for additional information on publishing Windows applications.