

The Complete Beginner's Guide to Mobile Application Development





Mobile Application Development An Overview

As of the third quarter of 2019, there were 2.47 million apps on Android's app store and 1.8 million on Apple's app store. Another data suggests that there are more than 12 million registered mobile app developers globally.

According to Statista, global mobile data traffic is projected to grow seven times between 2017 and 2022. More and more people are using the Internet through their smartphones and prefer mobile applications to mobile websites. This is because mobile applications are easily accessible, easier to use and fully customizable to individual preferences.

Step-by-Step Guide

Whether you are a business looking to develop a mobile app to engage with your audience, mobile app developer building an app for a client or amateur dabbling with mobile app development out of curiosity, here is a step by step guide to walk you through the mobile application development process.

Have an App Idea?

Businesses of all sizes, from single person company to a conglomerate, can benefit from developing apps that focus on different products and services. Whether you are in retail business or banking and finance or education industry, all you need to build mobile application is an idea to get you started.

You must brainstorm with your team or mentors for multiple app ideas. Each app idea must be researched for feasibility, viability and suitability in the target market.

Choose Device Platforms

As mentioned earlier, mobile applications have two major platforms – Android and iOS. The platform that you use should depend upon your business goals and what your intended customers are using.

Android platform is more popular than iOS because Google makes it available to companies developing or launching their own mobile phones. In contrast, iOS is available only on Apple's own devices.

iOS platform provides better security and is perceived premium, so many companies prefer to launch their app first on the apple Appstore. iOS apps have higher entry barrier in terms of both entry fees and the expensive Mac devices you need to develop the applications.



Develop for both Android and iOS

It is definitely possible to develop for both Android and iOS. If your customers use both Android and iOS phones, you will need to develop application for both platforms. When developing for both the platform you need to strategize and take an approach that allows you to reuse code, saving time and resources.

Select a programming language

After you have chosen the platforms for which you want to develop the app, you need to select a programming language. The most important criteria in selecting the programming language is the type of app you will develop – native, hybrid, cross platform or progressive web apps.

Here are some popular languages for mobile app development:

- Swift and Objective-C for iOS native apps
- Java and Kotlin for Android native apps
- Xamarin, Flutter, React Native, etc. for cross-platform apps

Besides the type of app, you also need to take stock of existing skills and what can be upgraded quickly. It does not make sense to invest time and money in learning a new language from scratch when there are so many options available.

Depending upon the features of the app to be developed, one language might be more suitable than others. Look into list of in-built and third-party libraries available to judge what suits you best.

Each of the language has its own pros and cons, and you must consider them before finalizing a language for app development. For example, if you want to avoid the verbose Java code, you can choose Kotlin for a leaner and cleaner code. If you want more control over coding of your cross-platform app, TypeScript with its static typing might suit you better than JavaScript.

Don't be in a hurry to zero-in on a language. Weigh the pros and cons of each option before deciding.



Alternatives for Building Mobile Applications

Before the actual development process, you need to take quite a few decisions about the mobile app. The first one usually is the platform to be used to develop the application. After that you can choose from any of the four different approaches to building mobile apps, all of which are discussed here briefly.

Native Mobile Applications

Native mobile applications use the operating system and frameworks provided by the app platform. Apps built using native language boast of:

- Faster development times
- Ability to build more robust applications
- Access to platform specific features
- Better development experience

Cross-Platform Native Mobile Applications

You can also choose to build applications for any one language and then convert the same codebase for other platforms. Such applications are developed using standard frameworks that would work on both Android and iOS platforms. Cross platform mobile applications are compiled into a native application that runs on the operating system of the mobile device.

Hybrid Mobile Applications

Like cross platform mobile applications, hybrid mobile applications are developed using frameworks that would work equally well on both the platforms. They are not compiled into native applications but installed as bundles that run-in web containers when launched by the user.

Progressive Web Applications

As the name suggests, progressive web applications are not developed as mobile applications. They are web applications that give a feeling of mobile application when accessed through smartphones. Progressive web applications are not launched and made available to the people through any of the app stores.



Comparing Native Apps vs. Hybrid Mobile Applications

You have an app idea, but you are confused whether you should build native apps for both platforms separately or a hybrid one. Understanding your intended audience's expectations from the app is key to deciding between native and hybrid mobile application development.

Consider the differences between them, which are listed in this table.

Native app	Hybrid app
Best runtime performance	Unable to access native APIs, reducing runtime performance
Best runtime performance	Accesses device hardware plugins and native modules, which restricts the performance
Separate codebases for each platform need to be maintained.	Single codebase that runs on each platform.
Uses platform frameworks	Uses web frameworks
Higher app building cost	Less cost as compared to native application development

Remember that if you prefer native apps, you can focus on one of the platforms and then convert the code to the other platform. You will need to redesign the interface because the two platforms are quite different when it comes to rendering the app on users' screen. Otherwise, if you choose your language and framework judiciously, you should be able to reuse most of the code.

How do Hybrid/Cross-platform Frameworks work?

As you know by now, non-native applications cannot access device hardware or software directly. Hybrid or cross platform applications are written in universal languages like JavaScript, CSS or HTML, which would run on all platforms. When the application is launched on the user's system, it runs in a container that enables these web applications to behave like native applications. If you already have a web app, most of the code can be reused in mobile application development.

There are two approaches to developing cross platform applications:

- Use platforms like ReactJS and NativeJS that provide native modules
- Use platforms like Flutter and Xamarin that compile the code into native code

Hybrid apps are best suited for smaller applications that do not need much resources.



Mobile Application Development Lifecycle

Mobile application life cycle is not much different from software or web application life cycle. It always starts with an idea that you must research to find its feasibility in the current market scenario. Once you are sure that your app will solve some user problem, you can go ahead with the actual development.

Wireframes and Prototyping

Before the actual coding begins, a lot of thought must go into what the app would bring to your users. Wireframes are a good way of understanding what your app would do and how. There are no design elements included in wireframe. These are some of the information that must be included in a wireframe so that all the stakeholders understand what they are signing up for:

- How the application will help the users
- User journey through the application
- Identify possible pitfalls
- Positioning of important menus, navigation & content areas, and other elements

Wireframe is a two-dimensional depiction of your app. After creating a wireframe, you should include the design elements into it and develop a clickable prototype for the app. Prototyping allows you to get a look and feel of how the application would actually be used by the audience. It should include buttons that you click to go to other screens, as a real user would do for the app.

Defining the Frontend and Backend

Once you have the prototype ready, you need to define the front end and the back end for the app. The initial mobile applications were self-contained units that functioned on the user's smartphones without any connection to the external world. However, as the availability of internet data increased, some of the data and functions could be off loaded to the back end on the cloud.

This need arose primarily because smartphones have less resources as compared to web apps. The mobile apps were shortened versions of web apps. But, if some of the features could be offloaded to cloud, a fully featured mobile app could be a reality. And this is what happened with increased availability of internet and cloud storage space. To improve their own efficiency and performance, developers started defining tasks that could be relegated to the cloud as backend.



Mobile Application Front-End

Mobile application front end is typically the user interface. The front end focuses on interface design and user experiences through awesome visual and interactive elements. When a user downloads the app, it is the front end that gets downloaded. It resides on the user's mobile and can be accessed by clicking on an icon on the screen.

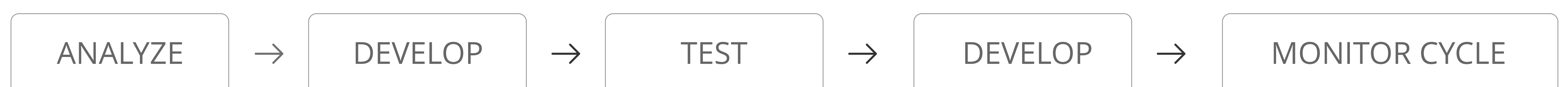
What a Front-end Development Workflow Looks Like?



The number of people involved in this workflow depends upon the size of the application. In case of a very small application all the stages could be handled by just one or two developers. But for enterprise-wide apps, dedicated teams could be handling each stage of the work flow.

Mobile Application Back-End

The back end of a mobile application could be maintained by the app owners or a third-party cloud service provider. With organizations adopting lean and agile methodologies, developers prefer outsourcing backend to third party service providers who are responsible for providing the infrastructure and maintaining it.



How Mobile Apps Integrate with the Backend?

The front end typically uses programming interface is called APIs to connect with the backend. These APIs could be provided by the back-end service provider or developed in-house. Developing in house is becoming rarer as fully customizable software development kits (SDKs) are easily available for use. There are wide range of APIs available and even more variety of ways in which they are accessed.

How to Interact with the Backend Data?

Back end data typically stores user information on cloud rather than the device itself. This makes the app lightweight but increases the time lag between request and response. Developers need to program the APIs in such a way that users do not feel the lag due to the data source being away from the device itself.



Test Locally, Test Globally

As mobile apps are no longer confined to the users' device, developers need to test the app both locally and globally. Local testing ensures that local servers are working properly, and users are getting the awesome experience that the app was designed for. It also checks the APIs responsible for integration of front end and Global testing is required for applications that offload their backend to cloud, the most common scenario in the current mobile app development ecosystem.

Publish Your App to The App Store

To publish your app to the app store, you must join the App store's developer program. Once you join the program you get access to native APIs, beta releases and proprietary frameworks. Follow the store guidelines and benchmarks to develop quality applications.

Android developer program can be easily joined if you already have a google account. Google play store is not very strict about app distribution. You can distribute the apps outside the store and let the users directly download and use them.

TechAhead

TechAhead is a **global leader** in Mobile Platforms that is trusted by **Fortune 500** companies to Startups. Founded in 2009, we are a 150 people company having a decade long **proven track record** in engineering innovative, robust, and scalable apps at speed.

Email us at **sales@techaheadcorp.com** or give us a call at **(818) 318-0727** (8:00 am to 5:00 pm pacific time). We usually respond within one business day.