**COMSATS University Islamabad Attock Campus**



**Assignment # 01**

**Student Name: M.Usman Baig**

**Roll#: SP20/BSE-040/ATK**

**Submitted To: Sir M. Kamran**

**Date: 03/10/2022**

**A comparison of Native and Cross Platform mobile app development.**

**Native mobile app development:**

Native mobile apps are applications developed to target either Android or iOS. Depending on which operating system you are developing for, your applications are usually coded in a specific programming language.

You write native Android applications using Java or Kotlin programming languages. Java was the initial language used to create Android apps. Google introduced support for Kotlin on Android in 2017. Kotlin supports [object-oriented and functional programming](https://circleci.com/blog/functional-vs-object-oriented-programming/), while Java is limited to object-oriented programming.

If you are developing native iOS applications, you use Objective-C or Swift programming languages. Objective-C is a superset of the C programming language. It is the initial language used to write software running on iOS. Apple introduced Swift in 2014 during their World Wide Developer Conference. It is a powerful general-purpose, high-level programming language for the Apple ecosystem. Apple states that Swift is 2.6 times faster than Objective-C, and the syntax is easier to learn.

Some well-known examples of native mobile applications include:

Google Maps

Pinterest

Spotify

WhatsApp

**Cross Platform mobile app development.**

You create cross-platform mobile applications from a single codebase. The goal of cross-platform app development is to target different operating systems with one project. You create these apps using cross-platform frameworks, which use platform-specific SDKs (Android SDKs and iOS SDKs) from a unified API. This enables you to easily access the different platform SDKs and libraries.

Private companies create these frameworks. Examples of popular cross-platform frameworks include:

React Native by Meta. It uses JavaScript as the programming language.

Flutter by Google. It uses Dart as the programming language.

Xamarin by Microsoft (which is being migrated to MAUI). It uses C# and XAML as the programming language.

Cross-platform mobile apps are usually compiled to use native UI elements that make the app feel native. As mentioned earlier, they provide an abstraction to the underlying platform SDKs. The exposed sensors include access to GPS, battery level, camera, and microphone.

Well-known examples of cross-platform mobile applications include:

Instagram, Skype, Walmart, and Airbnb (React Native)

Google Ads, My BMW App, eBay Motors, and the New York Times (Flutter)

The World Bank, Fox Sports, Alaska Airlines, and BBC Good Food (Xamarin)

**Different scenarios where each native and cross platform mobile app development is preferred.**

The work on **Cross~~-~~platform** apps entails creating the same code base wherein the resulting application is intended to function on both Android and iOS. Instead of having two teams of developers, you will need only one to create a cross-platform app. Therefore, you will save on the [development cost](https://www.uptech.team/blog/software-development-costs). Only a single cycle of development is needed to create an app that runs on multiple platforms.**‍** as the app is created with a single cross-platform development tool, only one code base is created. Cross-platform apps are unable to take advantage of native UX components. Therefore, it cannot deliver the same UX experience that is accustomed to the platform.

This is different from **native app** development wherein different teams work on a version of an app for every different platform in parallel development processes. **‍**A native app is easier to be published and usually ranked higher on the platform’s app store because it delivers better performance and speed. Apps built for the native environment also tend to be more scalable, thanks to the flexibility in resources management and the array of tools available. The direct interaction between the code and the underlying resources results in high performance. In addition, native apps generally have a better UX that is synonymous with the platform.

**List of frameworks/Tech Stack for cross platform mobile Application development.**

* **Node.js**: It is an incredible framework for developing cross-platform apps. It is an open-source environment that supports the development of server-side and networking apps. Node.js cross-platform apps are inherently highly efficient and responsive. The framework is capable of handling several concurrent connections together. It also comes loaded with a rich library of numerous JavaScript modules that help in simplifying the [development of web applications](https://appinventiv.com/web-application-development/).
* **Xamarin**: it is a streamlined framework used for developing apps for Android, Windows, and iOS with the help of C# and .Net, instead of JS libraries and HTML. It allows the developers to use 90% of the code for building an app for three distinct platforms. [Xamarin](https://dotnet.microsoft.com/apps/xamarin) delivers applications with aesthetics like a native app with the help of its amazing APIs, something which makes the decision making a lot harder [between Xamarin vs React Native](https://appinventiv.com/blog/react-native-vs-xamarin/).
* **Native Script**: renders beautiful, accessible, and platform-native UI and that too without the Web Views. Developers are only required to define once and let the Native Script adapt to everywhere. They can even customize the UI to specific devices and screens. Native Script also offers all native APIs, rendering an ability to the developers to reuse existing plugins straight from NPM into the projects.
* **Ionic:** Ionic is one of the most remarkable and popular cross-platform app frameworks, based on [AngularJS](https://camrojud.com/benefits-of-web-development-using-angularjs/). It allows developers to use a combination of [top programming languages](https://appinventiv.com/blog/top-programming-languages/) i.e., HTML5, JavaScript, and CSS and Cordova wrapper to access native platform controllers.
* **Appcelerator**: Appcelerator offers various tools for rapid application development. This indicates that a prototype can be created with much less time and effort to evaluate user interaction with UI. It is a great way to create cross-platform apps with just a single code base. Its primary focus is on streamlining the [app development process](https://appinventiv.com/blog/mobile-app-development-process/) with the help of native components present in JavaScript code. It has Arrow DB- a schema-less data store that allows developers to deploy data models with no additional efforts for setup.
* **React Native:** It is a framework built on JavaScript and is used to write real code and give the native-like feel to mobile applications that work on both Android and iOS. Due to its remarkable features, it is not only a preferred choice of developers but businesses also trust [React Native as the right platform for their apps](https://appinventiv.com/blog/is-react-native-the-right-platform-for-your-next-app/).
* **Flutter**: It is a software development kit designed to assist in the expeditious Android and development. Flutter promotes portable GPU, which renders UI power, allowing it to work on the latest interfaces. Flutter offers apps that easily and effectively run on multiple platforms with uniformity and dynamicity. Here are some of the remarkable features that make [Flutter an ideal cross-platform framework](https://appinventiv.com/blog/flutter-cross-platform-mobile-app-development/) among developers.