

**ASSIGNMENT: 01**

NAME: OSAMA

CLASS: BSE-6

SUBJECT: MAD

REG NO: SP20-BSE-011

SUBMITTED TO: SIR KAMRAN

DATE: 22-10-2022

**QUESTION:**

Explore the different frameworks/Tech Stacks available for cross platform mobile application development. Prepare a report that include following:

1. A comparison of Native and Cross Platform mobile app development.
2. Different scenarios where each native and cross platform mobile app development is preferred.
3. List of frameworks/Tech Stack for cross platform mobile Application development.

**ANSWER:**

Native app development means creating a mobile application that is tailored and dedicated to a specified platform like [iOS](https://mdevelopers.com/ios), or [Android](https://mdevelopers.com/android).

The term ***Native App Development*** refers to building a mobile app. The app is built with programming languages and tools that are specific to a single platform. For example, you can develop a native Android app with Java or Kotlin and choose Swift and Objective-C for iOS apps.

***Cross-platform development*** points to the process of creating an app that works on several platforms. This is done by using tools like React Native, Xamarin, and Flutter, where the apps created can be deployed in both android and IOS.

While cross-platform development saves time and cost.

**PRIORTIES:**

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’ll need only one to create a cross-platform app. Therefore, you’ll 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 can’t 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. Also, native apps generally have a better UX that is synonymous with the platform.   
‍

**Frameworks/Tech Stack for cross platform mobile Application development:**

**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.

**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 both on Android and iOS.

**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.

**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.

**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.

**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.

**appellatory: Appellatory** 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.

THE END