**Mobile Application Development**

It consists of :

* Android Development
* IOS Development

It can be done through:

* Native Development
* Hybrid Development
* Cross Platform Development

**1)Native Development**

Native development is the process of building apps for a specific operating system like Android and IOS.

**i)Android Tech Stack**

* **Language:** Java / Kotlin
* **Java:** Most popular language
* **Kotlin:** Google’s most preferred language for android app development.

If we are starting , Kotlin means much less code to type, test and maintain.

* **Platform:**
* **Android Developer’s tool**

It consists of:

* Libraries
* APIs
* Debugger
* Handset Emulator
* Sample Projects with source code
* **Android Studio**

It consists of :

* Intelligent code editor
* Feature rich Emulator
* Built in Templates
* Plug Ins

**ii)IOS Tech Stack**

* **Language:** Swift/ Objective- C
* **Objective-C:** Objective-C is a superset of C language . Developers use it if they need to support an already existing project. Objective-C was designed for small talk messaging features.
* **Swift:** It was created as a replacement for C based languages. It can be considered as a faster and most efficient language.

**Swift is recommended**

* **Platform:**
* **IntelliJ Appcode**
* **Apple Xcode**

**2)Hybrid Development**

Blend of both native and web solutions. They function like websites and run from within a native app.

**i)Ionic**

* **Language:** HTML, CSS, JS
* **Framework:**  Angular, React, Veu, JS

**3)Cross Platform Development**

Creating applications that run on multiple mobile platforms.

**i)React Native**

Created by Facebook to develop natively renderd mobile app for IOS, Android, Web

* **Framework:**  JS
* It provides a faster and high performance development process
* Most efficient because of broader approval and mature presence in market

**ii)Xamarin**

Open source platform created by Microsoft to compile a native code to different mobile binaries.

* Xamarin is closest to native.
* With xamarin , we can share 96% of code across platforms.
* **Language:** C# and .NET

**iii)Flutter**

Created by Google which consists of fully customized sets of widgets that implement google and IOS design and provide full native performance.

App development with Flutter is speedy and robust.

**Comparative Analysis:**

**Native App and Hybrid App**

|  |  |
| --- | --- |
| **Native App** | **Hybrid App** |
| Native applications require installation. | These apps don't require installation. |
| They require high maintenance. | They require less maintenance. |
| A large amount of budget is required to develop a native application that will be compatible with multiple platforms (such as iOS and Android). | Hybrid apps are cheaper, or we can say cost-effective, as they require less development time. |
| They have multiple codebases. | They have a single codebase. |
| They provide the best user experience. | Hybrid apps don't have a good user experience. |
| The languages used in native apps are Java, Swift, Kotlin. | The languages used in Hybrid apps are JavaScript, HTML, CSS. |
| These applications are particularly developed for one platform. | Hybrid apps can work on various platforms. It means that they can operate on both iOS and Android. |
| The updation in the mobile native apps can be done from the play store or app store. | On the other hand, mobile web apps updations are centralized. |

**Java and Kotlin**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Java** | **Kotlin** |
| **Ease of Use** | Difficult as huge amount of code required | Less code required |
| **Performance** | Slower | High level |
| **Cost** | High | Low comparatively |
| **Scalability** | It is comparatively lower in Java developed apps | High in Kotlin developed apps |
| **Community Support** | Large community and support | Moderate level |
| **Cross-Platform support** | Support limited platforms | Support multiple platforms |
| **Documentation** | Good but easy to understand | Good but hard to understand |