

**Name:** Gautam Singh

**Roll No. :** 20BCM023

**Practical** – 1

**Subject :** Mobiles Operating System & Application

## Summary Of Mobile OS

### Mobile OS

An operating system that supports running other application software on mobile devices is known as a mobile operating system. Although it is the same type of software as well-known computer operating systems like Linux and Windows, they are now somewhat lighter and simpler. Smartphones run a variety of operating systems, including Window OS, iPhone OS, Android OS.

#### Famous OS for Mobile:

1. Android OS
2. iPhone OS
3. Windows OS

All the three mobile OS platforms keep on introducing new facilities to make it easier for the developers come out with more user-friendly, interactive and visually appealing apps. And this is where the real competition gets alive between the three stalwarts.

## Android OS

The Android operating system is a mobile operating system that was developed by Google (GOOGL) to be primarily used for touchscreen devices, cell phones, and tablets. Its design lets users manipulate the mobile devices intuitively, with finger movements that mirror common motions, such as pinching, swiping, and tapping. Google also employs Android software in televisions, cars, and wristwatches—each of which is fitted with a unique user interface.

- The Android operating system was developed by Google (GOOGLE) for use in all of its touchscreen devices, tablets, and cell phones.
- This operating system was first developed by Android Inc., a software company located in Silicon Valley before it was acquired by Google in 2005.
- While the Android source code is released in an open-source format to help advance open standards across mobile devices, it is still packaged with proprietary software when sold on handset devices.

## iPhone OS

Apple (AAPL) iOS is the operating system for iPhone, iPad, and other Apple mobile devices. Based on Mac OS, the operating system which runs Apple's line of Mac desktop and laptop computers, Apple iOS is designed for easy, seamless networking between a range of Apple products.

- Apple iOS is the proprietary operating system used on Apple mobile devices such as the iPhone and iPad.
- iOS ranks as the second-most used mobile device operating platform in the world, behind Android.
- iOS features an intuitive user-centered design and the ability for app developers to create apps, distributed through the iOS app store.

## Windows OS

Windows Phone is a proprietary mobile operating system developed by Microsoft. Windows Phone introduced a new design language, previously called Metro UI, but later renamed to simply Modern.

Microsoft licenses the software to third-party hardware manufacturers, but keeps a stringent list of minimum requirements for the hardware it runs on to ensure the best user experience.

## Difference and Similarities

	Android	iOS	Windows Mobile
Language	Java (part of code can be in C/C++)	Objective-C	C#, Visual Basic, C or C++
Development Tool	Eclipse or IntelliJ IDEA	Xcode (only on Mac)	Visual Studio 2010+ (only on Windows)
Application	apk	ipa	cab/XAP/APPX
Development cost	Free	Tools are free but need publisher account	Visual Studio cost
Publisher account needed for Development	No	Yes	No
App Publisher	Google Play, Amazon Store, Samsung Store, SlideMe, F-Droid, AppsLib, etc	Apple Store	Windows Store
Publisher account cost	25\$ one time payment for Google Play	\$99 per year	\$19 per year
Verification process	When app goes to trending	Every time app is published or updated	Every time app is published or updated
Time taken for the app to be visible in app store (approx) after publishing	2 hours	2 weeks	2 weeks

	Android	iOS	Windows
<b>Memory Management</b>			
Memory usage	High	Low	High
Memory used for App handling	RAM	RAM	RAM + VM
Process running in background	Not Efficiently	Efficiently	Not Efficiently
Use of Garbage Collector	Yes	No	Yes
Background Processes	Do not freeze	Freeze	Suspend
To increase process speed	Uses internal memory	Don't use internal memory	Uses internal or virtual memory
Interface	User Friendly	User Friendly	Not User Friendly
Increase in Memory demand	Lag in app handling	No lag in app handling	Lag in app handling
Shortage of Memory	May kill some processes	Freeze background processes	Uses Virtual Memory
Capable of loading large number of apps	No	No	Yes
<b>Security</b>			
Arrival of new process	May kill existing process	Freeze some processes	No other processes will be affected
Utilities used	Own and third party	Own	Third Party Mostly
Issue Occurrence	Use patches	Use patches	Deliver updates
Rooting	Allowed	Not allowed	Not allowed

## Native App Development

The term native app development refers to building a mobile app exclusively for a single platform. 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.



Native apps are known to deliver exceptional user experience as they are generally high performance. User experience is also enhanced as the visuals are tailored to the platform UX.

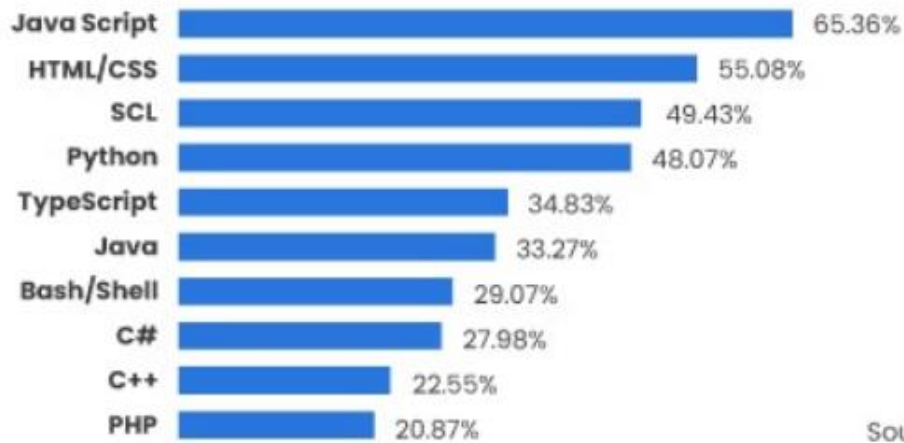
Native mobile apps, unlike websites and web applications, do not operate in the browser. They must be downloaded from platform-specific app stores like Apple's App Store and Google Play. Each program may be accessed by tapping its icon on your device's screen after installation.

Native applications are more complicated to create than mobile websites. There's no need to be concerned about browser compatibility or behavior. You may use the native capabilities of mobile operating systems to create a richer user experience and implement app features.

## Tools and Technologies for Mobile OS

### Technologies

#### Top 10 Most Used Programming Languages Worldwide in 2022



Source: Statista

### Tools

#### Android

- ✓ *Android Studio*
- ✓ *Java*
- ✓ *Kotlin*

#### IOS

- ✓ *Swift*
- ✓ *Objective-c*
- ✓ *X-code*

## Frameworks

*A mobile development framework is a software library or framework that provide an underlying structure to enable the development of mobile application for a specific environment.*

*In general, there are three broad frameworks:*

### 1. Native Apps

Native mobile app frameworks enable platform-specific development. A native framework is designed to build applications for a specific operating system, such as iOS, Android, or Windows. For example, Java and Kotlin are used for developing native Android apps, while Objective-C and Swift are used for developing native iOS apps. Native applications take advantage of the device-specific hardware and software to offer highly optimized performance and results.

### 2. Web-Native Mobile Apps

This framework is an approach to promote cross-platform mobile development. Most cross-platform frameworks support developers to build mobile apps with the web languages they already know, reducing the learning curve and development cycle. Frameworks like Ionic, Xamarin, React Native, and Flutter enable organizations to build truly native mobile applications for iOS and Android as well as mobile-optimized Progressive Web Apps (PWAs).

### 3. Hybrid Apps

Hybrid frameworks combine the features of both native and cross-platform frameworks. This framework offers plugins that allow developers to embed the code written using standard web technologies (HTML5, CSS3, and JavaScript) into a native application. For example, Apache Cordova offers plugins to connect its framework with native APIs like camera, GPS, notifications, and much more to deliver a native mobile experience. Ionic framework's Capacitor plugin helps create web native apps which also run natively on iOS and Android mobile devices.

## Benefits of Native App Development

### Better Performance

First and foremost, native apps have better performance than cross-platform or hybrid solutions. Primarily, because the application interacts directly with native APIs and doesn't depend on middleware.

Thus, by getting full hardware and operating system support, native apps end up being more responsive and highly efficient. Plus, the devices they're loaded on actually store all the key data and help content load faster.

### Advanced Customization

Since native applications make the most of the features within the operating system, they can be highly customized. With this approach, developers can access the GPS, camera, microphone, and other hardware to build a final product that is truly unique.

In short, thanks to developers not needing to combine the features of two operating systems, they have fewer limitations and can focus on creating a one-of-a-kind solution.

### Enhanced User Experience

Every platform has its own UI/UX guidelines that developers ought to stick to. Hence, when creating a native mobile app, these standards are well adhered to, resulting in a look and feel that is consistent with the operating system.

The consistency of native mobile apps also brings a much more intuitive and interactive user experience because people are familiar with the layouts typical for their operating system. So, whenever they use your application, they feel like it's something they are used to.



## Greater Security

Another reason to opt for native app development is the higher security level that it offers. With data protection concerns rising, software solutions across all industries should make users feel safe when sharing digital information.

You see, cross-platform and hybrid solutions may carry with them the vulnerabilities of each platform, the browser they work on, and the universal languages they employ. Of course, this isn't always the case, just a possibility. However, definitely one to consider before starting your project.

## Fewer Bugs

Since building native apps incorporates using specific and different for each platform tools, it has fewer tech dependencies and decreases the possibility for bugs to arise. In the end, keeping two apps in two independent codebases is less difficult and error-prone than implementing two apps in the same codebase.

## Improved Scalability

The architecture of native applications allows for acquiring more user traffic and supporting a higher load. Thus, there are more opportunities for scalability and your product's future growth.

It's easier for native apps to scale as you need since they effectively use existing hardware, provide a better performance, and reduce the risk of the app crashing during the increased traffic.

## Easier Launch to the Market

Another advantage of native applications is the relatively easy process of launching the app to the app stores. Both App Store and Google Play have rules and standards to follow when you publish your solution to the market.

For cross-platform apps, it can entail more bottlenecks to cope with. On the other hand, when delivering native products, mobile developers use native tech

stack and align the code from the first lines to the regulations and guidelines provided by each store.

### Instant Updating

The truth is, with cross-platform applications, developers aren't immediately able to implement new features. Given the frequency with which iOS and Android release updates, this can sometimes be a problem as the delays degrade the user experience.

With native apps, everything is much simpler. When Android and iOS roll out software upgrades, IT teams can immediately implement the latest features since they've got quick and easy access to new SDKs that help modify the application.

### Challenge in Native App Development

#### Higher Development Costs

Since you will have to build separate apps for different platforms, the development will get costly. For each platform, you will need to hire different development teams. Further, app maintenance also has to be done separately for each platform. This is a major challenge of native app development.

#### More Development Time

Unlike hybrid apps, native apps cannot use a single codebase for multiple platforms. Separate codes need to be written for different platforms. That doubles up the time of app development.

### **Need for Skilled Developers**

The development of native apps is somewhat complex, and a skilled team of developers are needed to create a successful native app. The programming languages used are more advanced, and it isn't easy to find developers proficient in those languages.

### **Require Constant Updates**

Whenever there is a new update or a bug fix, the users have to update the app to the latest version from the respective app store. Otherwise, they may experience glitches while operating the app.

### **Lengthy Downloading Process**

Compared to web app development, a major disadvantage of native app development is that native apps have to be downloaded. The app download is a multi-step process including going to the app store, finding an app, complying with its terms, downloading, and installing. This requires users to spend considerable time and effort and may result in user attrition.