

Introduction to Mobile Applications

Mobile Operating System

A mobile operating system, also called a **mobile OS**, is an operating system that is specifically designed to run on mobile devices such as mobile phones, smartphones, PDAs, tablet computers and other handheld devices.

Popular Mobile OS

Popular Mobile OSs along with market share in 2019 are as follows:

- Android: **75.27%**
- iOS: **22.74%**
- KaiOS: **0.75%**
- Windows: **0.24%**
- Samsung: **0.22%**

Ref: <http://gs.statcounter.com/os-market-share/mobile/worldwide>

Types of Mobile Applications

1. Native Applications
2. Hybrid Applications
3. Progressive Web Applications (PWAs)

1. Native Applications

These are applications developed to be used on a particular platform or operating system such as Android, iOS etc. Native apps are usually written in languages that the platform accepts.

The principal advantage of native apps is that they optimize the user experience. By being designed and developed specifically for that platform, they look and perform better.

1. Native Application Languages

Some languages used to develop native applications:

- Swift or Objective C for **iOS** applications
- Java, Kotlin for **Android** applications
- C# or VB.NET for **Windows** applications

1. Native Applications - Cons

- They are considerably more expensive to develop compared to cross-platform and web applications.
- They require more time to develop as one application has to be written in different languages for different platforms.
- They have a higher cost of maintenance and pushing out updates, due to multiple source code bases.

1. Native Applications - Pros

- They are very fast and responsive because they are built for that specific platform.
- They have the best performance.
- They are more interactive, intuitive and run much smoother in terms of user input and output.

2. Hybrid Applications

These are applications developed to be used across multiple platforms. This type of applications are developed in one language and later deployed to multiple platforms (iOS, Android,... etc.).

Hybrid mobile applications are built in a similar manner as websites. Both use a combination of technologies like HTML, CSS, and JavaScript. However, instead of targeting a mobile browser, hybrid applications target a WebView hosted inside a native container. This enables them to do things like access hardware capabilities of the mobile device.

2. Hybrid Applications - Frameworks

Some popular frameworks for building Hybrid applications include:

- Ionic Framework
- PhoneGap
- Sencha Touch etc.

2. Hybrid Applications - Pros

- Adaptable to multiple platforms, as the same code can be re-used for Android, iOS, and Windows.
- Unified and less expensive development, as the app only has to be developed once using one code base.
- Faster development time when compared to native apps as only one development process is involved.

2. Hybrid Applications - Cons

- Slower app performance when compared to Native Apps, because the hybrid framework acts as a bridge to communicate with the phone's native features.
- Apps with heavy animations and sound effects aren't as seamless as their native counterparts.

3. Progressive Web Applications (PWAs)

A Progressive Web App (PWA) is a web app that uses modern web capabilities to deliver an app-like experience to users without requiring them to install an app from the AppStore/PlayStore. They are usually accessible by a web URL which can always be pinned or saved on your phone's home screen. PWAs are usually built using HTML, CSS, JavaScript too.

3. PWAs - Pros

- Easy to build.
- Easy to maintain.
- Works offline and performs well on low-quality networks.
- Build one app for all platforms — iOS, Android, etc. as long as it can run a browser.

3. PWAs - Cons

- Needs a browser to run. Users have to take another step to type in the URL of the app which amounts to a poor user experience.
- Web apps are less interactive and intuitive than native apps.