

Development of Native Mobile Application

Jatinder Singh Bal

Guru Kashi University, Talwandi Sabo

Abstract

The popularity of mobile applications has skyrocketed in the last few decades. Mobile device adoption is now outpacing that of laptops and desktop computers. It's because mobile devices have a wide variety of applications that allow the user to complete tasks from any location using a small, portable device. There are some drawbacks to this, such as the need for cross-platform application compatibility. Today's developers are primarily concerned with creating applications that can be used by a large number of people without limiting their features or performance capabilities.

In order to run on mobile devices like smart phones and tablets, mobile applications (apps) are developed specifically for those platforms. Many people rely on mobile apps to keep track of their day-to-day activities and commitments, as well as their informational requirements. Developing mobile apps has its own set of challenges and characteristics. When faced with dwindling resources, developers must use their ingenuity to meet the ever-increasing demands of their users.

Keywords: Native, Applications, Development of Native Mobile, Cross Platform.

1. Introduction

For most people, Android phones are their primary method of communication on a daily basis. There are a lot of apps that allow people to have endless fun, and the Android system has become a popular choice for smart phones. These software programmes, also referred to as mobile applications or simply apps, are made to run on portable electronic devices, such as smart phones and tablets. These applications serve a variety of functions, ranging from practical ones like maps and contact lists to lighter ones like games and social media. Developing an app for a specific operating system requires a mobile application developer. Application development is referred to as "android app" for Android apps and "iOS app" for iOS apps. Both the Android and iOS operating systems are supported by a small number of applications. Cross-platform applications are what they're called. Native applications, on the other hand, are those developed for a single operating system. In this article, we'll explain in detail the difference between native and cross-platform apps.

It is difficult for a hobbyist programmer to learn all the different programming languages and port their application to all of them. Because of the initial Hybrid App Platforms, it became significantly simpler to develop mobile applications for a variety of operating systems (Paulo R, Adriano B, 2015). HTML5, CSS, and JavaScript can all be used to create an application that can be ported to any operating system (Jeff Whatcott, 2011).

OS	Programming Language	Development Environment	Application Store
Google's Android	JAVA, Kotlin	Android Studio, Android SDK	Play Store
Apple's iOS	Objective-C/Swift	XCode	Appel-iTunes

Microsoft Windows phone	Visual C#, C++	Visual Studio	Window Phone Market
RIM BlackBerry OS	JAVA	BlackBerry Plug-in for Eclipse	BlackBerry Apps World

1.1. Cross Platform Application:

These apps can be downloaded and installed on a user's device just like a native app because they have the same look and feel. Despite the fact that they appear to be native apps, they are actually just a collection of web pages housed in a container. HTML5, JavaScript, and CSS are commonly used to build them. As a result, cross-platform applications can appear and function as if they were developed for a single platform. In fact, cross-platform apps like Twitter, Uber, and Instagram are some of the most popular mobile apps. The development environment for cross-platform applications differs from that for native applications. With the help of various third-party toolkits, the applications look and feels like native applications. When it comes to building and deploying cross-platform applications, the web-based component is very similar to that of standard web applications. In terms of both advantages and disadvantages, both cross-platform and native mobile applications are viable options.

1.2. Native Mobile:

Native mobile applications have only recently appeared on the scene. The emergence of mobile devices ushered in a new era of application development on a dynamic new platform. Both web-based and desktop applications can be found in the mobile apps.

2. Native Application

Unlike cross-platform apps, native mobile apps are optimised for a single platform. A closer connection to the platform is made by them. Windows, iOS, and Android are all supported. In order to create these apps, you must use the OS's native programming language, in this case Java for Android and Swift or Objective-C on iOS only. Mobile application developers who work exclusively on native platforms have as their goal the complete integration of the app with the mobile device. APIs provided by mobile operating system vendors are utilized in the creation of these applications. With the help of a variety of IDEs, programmers can build efficient applications that also provide a pleasant user experience. Many of the device's built-in features can also be used in conjunction with the app. Native mobile applications have a better access to features like GPS and Maps. They can be more costly and time consuming in the development stage but native mobile applications turn out to be more efficient and leave better user experience.

2.1. Native Application Development:

An Integrated Development Environment (IDE) is a term used to describe a software development process where the developer employs the primary language, tools, and framework for the platform being targeted (IDE).

A native app is one that is built using the platform-specific development tools and languages (XCode and ObjectiveC for iOS apps, Eclipse for Android, Java for Android, Visual Studio for Windows; and C# for Windows) that are supported by the respective platforms. To take advantage of the platform's operating system features and other software installed on the platform, native apps must be written specifically for that platform.

2.2. Feature of Native Application Development:

1. The most enjoyable overall. Multi-touch, faster graphics APIs, fluid animation, built-in components, and ease of use are some of the typical processes native applications would typically perform (Paulo R, Adriano B, 2015).

2. User interface (User Interface) gestures are made possible thanks to the native application multi-touch features. Users could, for example, double-tap to zoom in. Advanced gestures such as pinch-spread (Bernard K and Joseph M, 2015.).

3. As a result of the different device characteristics, native applications offer a fast graphics API. When providing a gaming experience on a mobile device, animation is a must. It's also required for complex computational algorithms and highly interactive reporting (Paulo R, Adriano B, 2015).

2.2. Limitations in Native Application Development:

1. For businesses, native applications can be a boon in terms of graphics, app distribution, and device compatibility. However, their lack of portability presents significant challenges.

2. When developing a native app, developers must deal with the risks of an unstable mobile-platform landscape and limited app control. Developing a native app is the most time- and money-consuming approach, but the costs vary depending on the complexity of the app. Most native apps, according to Forrester Research, require at least six months of full-time work and cost between \$20,000 and \$150,000, depending on the complexity of the app.

3. The owner of the app store retains control of a native app once it is placed in the store (like Apple or Google). As a result, companies using the app-store model are at the mercy of a third-party vendor for their apps.

4. Because native apps operate in a silo-based model, there may be an increase in maintenance costs. In order to make sure that each application works with the device, updates must be performed multiple times because each operating system is unique.

5. Last but not least, organisations will have to go through multiple processes to ensure the successful deployment of their application on each device because each platform has a specific process for app approval.

3. Mobile App Development Issues

3.1. Platforms:

Apple's iOS, Google's Android, Microsoft's Windows Phone, and Blackberry round out the top three mobile app development platforms. While iOS and Windows Phone are both proprietary, Android is the only open source platform.

- i) **Apple iOS:** Apple Inc.'s iOS is a proprietary operating system that is only available on Apple devices. A computer running OS X is required to develop an iOS app. Developers can use Apple's Xcode IDE to create the app. In order to complete the project, you'll need the iOS SDK. In this case, objective-C is being used as the coding language.
- ii) **Google Android:** Android apps can be developed on Windows or Linux computers. Android OS 5.0 Lollipop is the most recent release. Android Studio is Google's proprietary integrated development environment (IDE) for Java mobile applications. However, the SDK provided allows for the creation of apps from the command line.
- iii) **Microsoft Windows Phone:** Microsoft's own mobile operating system code-named Windows Phone. Windows Phone 8 is the most recent iteration. A visual programming language such as C# or Visual Basic.Net is used to create the apps themselves. Windows Phone SDK and a powerful IDE, Microsoft Visual Studio, are also available.

Table 1 gives the different OS platforms and their development tools for mobile applications development.

OS Platform	Development Tools			
	Current OS version	Programming language	IDE	Others
1. Apple iOS		Objective-C, Ruby	Xcode	iOS SDK
2. Google Android	Android 5.0 Lollipop	Java	Eclipse, IntelliJ, Android Studio	Android APIs, SDK manager
3. Microsoft Windows Phone	Windows Phone 8	Visual C#, Visual Basic .Net	Microsoft Visual Studio 2012	Windows Phone SDK

Table 1: The different OS platforms and their Development tools

4. Conclusion

Mobile app development has its own unique set of characteristics and challenges, and it places a heavy burden on the developers' creativity to meet the ever-increasing demands of their users while working with a finite amount of resources. In order to ensure the best possible experience for customers, developers must be able to strike a delicate balance between app functionality and system capabilities.

Because of the many advantages that mobile devices offer, customers are more likely to use and prefer mobile applications. They have a wide variety of applications, ranging from simple functions like calculating to more complex ones like storing and transferring files. With the wide range of potential customers that mobile applications have, they must be developed in a way that reaches the widest possible audience. Different mobile operating systems may be used by different people. It's difficult to design applications that don't impose any restrictions on the user.

Cross-platform approaches are a viable alternative to native apps because of their better compatibility with different mobile operating systems and developer resources. With tight budgets, small development teams, and a short time frame in which to develop mobile apps, a cross-platform approach is necessary.

5. References

[1]. Ribeiro and A. R. D. Silva, "Survey on Cross-Platforms and Languages for Mobile Apps," *Eighth International Conference on the Quality of Information and Communications Technology*, 2012.

[2]. Pohares, V. C. Kulloli, T. Bhattacharyya, and S. Bhure, "Cross Platform Mobile Application Development," *International Journal of Computer Trends and Technology*, vol. 4, no. 5, pp. 1095-1100, 2013.

- [3]. S. S. Jagtap and D. B. Hanchate, "Development of Android Based Mobile App for PrestaShop eCommerce Shopping Cart (ALC) ," *International Research Journal of Engineering and Technology (IRJET)* , vol. 4, no. 7, pp. 2248–2254 , Jul. 2017.
- [4]. M. Q. Huynh and P. Ghimire, "Browser App Approach: Can It Be an Answer to the Challenges in CrossPlatform App Development?," *Journal of Information Technology Education: Innovations in Practice*, vol. 16, pp. 047–068, 2017.
- [5] B. König-Ries, "Challenges in Mobile Application Development," *it - Inf. Technol.*, vol. 51, no. 2, pp. 69–71, 2009.
- [6] Neha Verma , Sarita Kansal and Huned Malvi, "Development of Native Mobile Application Using Android Studio for Cabs and Some Glimpse of Cross Platform Apps", in *International Journal of Applied Engineering Research*, ISSN 0973-4562, Volume 13, Number 16, 2018, pp. 12527-12528.
- [7] Chaitanya Kaul and Saurav Verma, "A Review Paper on Cross Platform Mobile Application Development IDE", in *IOSR Journal of Computer Engineering (IOSR-JCE)*, e-ISSN 2278-0661,p-ISSN 2278-8727, Volume 17, Issue 1, Ver. VI (Jan –Feb. 2015), pp. 30-33.
- [8] Statista, "Number of apps available in leading app stores as of may 2015," 2015. [Online]. Available: www.statista.com/statistics/276623/number-of-apps-avaiable-in-leading-apps-stores. [Accessed: 05-Jun-2015].
- [9] Apple Inc., "Start Developing iOS Apps Today," 2015. [Online]. Available: <https://developer.apple.com/library/GettingStarted/RoadmapiOS>. [Accessed: 16-Jun-2015].
- [10] Microsoft Developer Network, "How to create your first app for Windows Phone 8," 2015.
- [11] E. Gamma, H. Richard, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Publishing, 1995.
- [12] R. Jorg, "Patterns of Mobile Interaction," *Pers. Ubiquitous Comput.*, vol. 6, no. 4, pp. 282–289, 2002.
- [13] A. Tranfici, "5 Mobile Design Patterns for a Successful App," 2014. [Online]. Available:
- [14] A. Tranfici, "Examples of Mobile Design Patterns Part 1 and 2." 2013.
- [15] Apple Inc., "iOS Human Interface Guidelines: Human Interface Principles." 2013.