© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed

Special Edition
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur,

Comparative Study of Cross Platform Development And Native Development

Abhay Ambule
Student, Department of Computer Engineering
St. Vincent Pallotti College of Engineering & Technology, Nagp\ur India

Aashish Omre
Student, Department of Computer Engineering
St. Vincent Pallotti College of Engineering & Technology, Nagpur India

Prof. Pallavi Wankhede
Professor , Department of Computer Engineering
St. Vincent Pallotti College of Engineering & Technology, Nagpur India

Abstract

With the rise of mobile devices and their widespread use, there has been a significant increase in the demand for mobile applications. Mobile applications are now essential for both consumers and organizations. However, it is getting more and more challenging for developers to make applications that work with all of the different mobile devices that are currently on the market. This has led to the emergence of two distinct types of mobile application development – native and cross-platform.
We present a comparison of native and cross-platform application development in this paper. We will discuss the advantages and disadvantages of each approach and provide insights into which approach would be better suited for specific types of mobile applications.
Keywords
native development, cross-platform development, React Native, Java, Kotlin, Flutter, Mobile applications.

I. INTRODUCTION

In the upcoming years, it is anticipated that the remarkable rate of growth in the mobile app market, which is already booming, will continue. Statistic estimates that by 2023, there will be 7.33 billion mobile phone users worldwide, with 3.8 billion of those using smartphones. The increasing demand for mobile applications has led to the development of two distinct types of mobile application development – native and cross-platform. Developing native applications entails making software tailored to a specific mobile operating system, such as iOS or Android. On the other hand, cross-platform application development involves creating applications that can run on multiple platforms.
The purpose of this paper is to present a comparative study of native and cross-platform application development. We will discuss the advantages and disadvantages of each approach and provide insights into which approach would be better suited for specific types of mobile applications.

II. LITERATURE SURVEY

Native Application Development

Native application development involves creating applications that are specific to a particular mobile platform, such as iOS or Android. A platform-specific language, such as Java or Kotlin for Android or Swift or Objective-C for iOS, is used to create native applications.[1]

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special Edition
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur,

Advantages of Native Application Development

Performance: Native applications are faster and more efficient than cross-platform applications. Native applications are compiled into machine code, which cross-platform applications.

User Experience: Native applications provide a better user experience than cross-platform applications. Native applications are designed specifically for a particular platform, which allows them to take advantage of the platform's unique features and provide a better user experience.

Integration: Native applications can be easily integrated with other native applications on the same platform. Native applications can take advantage of the platform's APIs and frameworks, which allows them to easily integrate with other native applications on the same platform. [2]

Disadvantages of Native Application Development

Cost: Native application development can be expensive. Developing native applications requires expertise in the specific language and tools used for the platform, which can be costly.

Time: Native application development can take longer than cross-platform application development. Developing native applications requires the development of separate applications for each platform, which can increase development time. It requires expertise in the specific language and tools used for the platform, which can increase development time.

Maintenance: Native applications require separate maintenance for each platform. Each platform has its own set of tools and requirements, which can make maintenance more difficult and time-consuming. This can also increase maintenance costs. [3]

Cross-Platform Application Development

On the other side, cross-platform application development is the process of creating mobile applications that can run on many platforms while using a single codebase. Cross-platform applications are built using frameworks such as React Native, Xamarin or flutter. [4]

Advantages of Cross-Platform Application Development

Cost: Cross-platform application development can be less expensive than native application development. Developing cross-platform applications allows developers to write code once and deploy it across multiple platforms, which can reduce development costs.

Time: Cross-platform application development can be faster than native application development. Developing cross-platform applications allows developers to write code once and deploy it across multiple platforms, which can reduce development time.

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special Edition
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur,

Maintenance: Cross-platform applications require less maintenance than native applications. Since cross-platform applications are developed using a single codebase, maintenance can be easier and less time-consuming. [5][6]

Disadvantages of Cross-Platform Application Development

Performance: Cross-platform applications can be slower and less efficient than native applications. Cross-platform applications are not compiled into machine code, which can make them slower and less efficient than native applications. Cross-platform applications also do not have access to the device's hardware, which can limit their capabilities and provide a less immersive user experience.

User Experience: Cross-platform applications can provide a less immersive user experience than native applications. Cross-platform applications are designed to work across multiple platforms, which can limit their ability to take advantage of the unique features of each platform. This can provide a less immersive user experience.

Integration: Cross-platform applications can be more the same platform. Cross-platform applications do not have access to the platform's APIs and frameworks, which can make integration with other native applications more difficult. [7][8]

Native Libraries and Frameworks

Java - Android SDK: The Android SDK provides a range of libraries and tools for developing native Android applications using Java. It includes a range of features such as UI controls, multimedia support, database access, and networking.

Kotlin - Kotlin Native: Kotlin Native is a framework that allows developers to write Kotlin code that can be compiled into native code for different platforms. It includes support for developing native applications for Android, iOS, macOS, and Linux.

Swift - Cocoa Touch: Cocoa Touch is a native framework for iOS development using Swift. It includes a range of pre-built components and libraries optimized for the iOS platform, such as user interface controls, media playback, networking, and graphics rendering.

Objective-C – Cocoa: Cocoa is a native framework for macOS development using Objective-C. It includes a range of pre-built components and libraries optimized for macOS, such as user interface controls, media playback, networking, and graphics rendering.

Java – JavaFX: JavaFX is a Java-based framework for developing native desktop applications. It includes a range of user interface controls, media playback, and graphics rendering components that allow developers to build rich and complex desktop applications. **[10]**

Cross-Platform Libraries and Frameworks

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special Edition
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur,

React Native: Developed by Facebook, React Native is an open-source framework for building mobile applications using JavaScript and React. It allows developers to write code once and deploy it on both iOS and Android platforms.

Xamarin: Developed by Microsoft, Xamarin is a difficult to integrate with other native applications on popular cross-platform framework that allows developers to write applications in C# and .NET. Xamarin allows developers to share code across multiple platforms and has native UI controls for each platform.

Flutter: Developed by Google, Flutter is an open-source mobile application development framework that allows developers to create native-looking applications for iOS and Android platforms using the Dart programming language.

Electron: Electron is a cross-platform desktop application development framework that uses web technologies such as HTML, CSS, and JavaScript. It allows developers to build native-like desktop applications that can run on Windows, Mac, and Linux operating systems.

Cordova: Cordova is a mobile application development framework that allows developers to create applications using web technologies such as HTML, CSS, and JavaScript. It uses a plugin architecture to provide access to native device functionality and can be used to build applications for iOS, Android, and other mobile platforms **[9]**

III. DISCUSSION

Project needs and goals determine cross-platform or native development. Both methods have pros and cons, and the best one depends on the app, audience, timeline, and budget. Native development requires creating apps for iOS, Android, and Windows. Each platform-optimized app provides a responsive and intuitive user experience. Native development provides access to all platform-specific features and functions, enabling high-performance apps.Cross-platform development uses a single codebase for multiple platforms. Eliminating platform-specific code makes this method faster and cheaper. Cross-platform development lets developers use many libraries and frameworks, speeding up development. Cross-platform development uses abstraction layers that may not be as efficient as platform-specific code, which may degrade limit access to platform-specific features and functions, making apps less feature-rich. Cross-platform or native development depends on performance, cost, and development time. Native development may be best for performance and user experience. Cross-platform development may be best for cost and time.
Before making a decision, it's important to carefully consider the project's needs and goals and work with experienced developers to make sure the chosen approach is the best fit.

IV. CONCLUSION

Both Native and Cross-platform application development have their strengths and weaknesses. Native development offers better performance and access to device features, but is more time-consuming and expensive. Cross-platform development is faster and more cost-effective, but may not perform as well as native apps and has limited access to device features. The choice of which platform to use depends on the app's requirements, budget, and target audience. Developers need to carefully consider these factors before deciding which platform to use.

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 14 Issue: 03 | April - June 2023
Paper is available at http://www.jrps.in | Email : info@jrps.in
Refereed & Peer Reviewed
Special Edition
NCASIT 2023, 29th April 2023
Department of Computer Engineering,
St. Vincent Pallotti College of Engineering & Technology, Nagpur,

REFERENCES

[1] Latif, M., Lakhrissi, Y., Nfaoui, E. H. & Es-Sbai, N. Cross platform approach for mobile application development: A survey in International Conference on Information Technology for Organizations Development (IEEE, 2016), 1–5.

[2] Palmieri, M., Singh, I. & Cicchetti, A. Comparison of cross-platform mobile development tools in International Conference on Intelligence in Next Generation Networks (IEEE, 2012), 179–186.

[3]. Javeed, A. Performance Optimization Techniques for ReactJS in Int. Conf. on Electrical, Computer and Communication Technologies (IEEE, 2019), 1–5.

[4] Serrano, N., Hernantes, J. & Gallardo, G. Mobile Web Apps. IEEE Software 30, 22–27 (2013).

[5] Lin, H. & Lee, G. Building a Secure Cross Platform Enterprise Service Mobile Apps Using HTML5 in International Conference on Network-Based Information Systems (IEEE, 2015), 162–166.

[6] Jia, X., Ebone, A. & Tan, Y. A Performance Evaluation of Cross-Platform Mobile Application Development Approaches in Int. Conf. on Mobile Software Engineering and Systems , 92–93.

[7] Dalmasso, I., Datta, S. K., Bonnet, C. & Nikaein, N. Survey, comparison and evaluation of cross platform mobile application development tools in Int. Wireless Communications and Mobile Computing Conference (IEEE, 2013), 323–328.

[8] Sommer, A. & Krusche, S. Evaluation of cross-platform frameworks for mobile applications in Software Engineering - Workshop band (eds Wagner, S. & Lichter, H.) (Gesellschaft fur Informatik e.V., 2013), 363–376. ¨

[9] https://medium.com/javarevisited/top-5-frameworks-to-create-cross-platform-android-and-ios-apps-in-2020-d02edf3d01f1

[10] https://technostacks.com/blog/mobile-app-development-frameworks/