IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

REVIEW ON: CROSS PLATFORM MOBILE APPLICATION DEVELOPMENT

Pranjali Hiwale[1], Anand Kalsait[2], Kishori Choukade[3], Aishwarya Puri[4], Prof. Priyanka Vikrant Shirbhate[5]

[1,2,3,4,5]Department of Computer Engineering
[1,2,3,4]Government College of Engineering Yavatmal, Maharashtra, India.
[5]Assistant Professor, Government College Of Engineering Yavatmal, Maharashtra, India
[1,2,3,4]Dr. Babasaheb Ambedkar Technological University,
Lonere

Abstract: Cross-Platform Application Development is the development of mobile-based applications so that the development of these types of applications can be made platform-independent. A review has been made in this field while considering Integrated Development Environment is being proposed which will help a software developer to code an application in a single code base and deploy that single code base to multiple operating systems meaning if a developer codes in Java for Android, the developers can deploy the same code from JAVA to iOS subsystem. First the development of an operating system compatibility architecture which helped to run unmodified iOS binaries on Android operating system and second phase of proposed solution also helped to understand the cross-platform application development tools which are currently available in the market. The final step is to implement such cross-platform development tools in more detail and finally a mathematical model-based application.

Keywords: Cross-Platform Mobile application development, IDE, Android development, iOS development, Native

I. INTRODUCTION

Mobile applications are becoming increasingly common today, especially in the business world. Mobile App Software Development refers to the process of developing software for mobile devices such as smartphones and the personal assistant. Through the use of mobile applications the user is provided with various features and services that will enable him to fulfill all his needs and much more than that. Apps must interact with users. Apps can be downloaded from various platforms such as Google Play Store and iOS App Store. There are free apps and paid apps. Some apps can be used for free for a while before you sign up for a premium membership. For priced apps, about 20% -30% goes to distribution provider (for example iTunes) and the rest to the app manufacturer. With the development of applications, the limitations and features of mobile devices must be considered. Mobile devices have low processing power, are battery-operated but have many features such as location detection. The wide range of screen sizes and hardware specifications also need to be considered while developing the applications.

For developing applications, specialized integrated development sites such as Android Studio or Eclipse or any Application Development Environmet are required. The app is first tested using devices called emulators that simulate real hardware device software and then perform a field test [1]. Mobile User interface (UI) design is another important component in application development. The UI includes configuration considerations, screen and user input and output flow. The user cheats the system by installing and then the expected results are displayed exit. There are limits to the mobile UI as the screen size is limited. The mobile UI is considered as the front and rear of the background to support access to business systems. Background services such as data routing, security and authorization and alerts are provided with middleware components such as Mobile Backend as a service (MBAaS). The paper begins by examining the challenges of mobile application development.

II. COMPARISION BETWEEN NATIVE AND CROSS PLATFORM APPLICATION DEVELOPMENT

Performance: Undoubtedly, native applications work better and are more responsive than cross-platform applications. This is great because native code interacts directly with device resources. Eg. in the most demanding computer applications, the environment is 10 times faster or faster than the RN application.

Development time: However, as a trade, traditional applications take a lot of time to develop as different teams are required to develop different forums. Cross-platform applications shine here as the code can be easily replicated in every few platforms.

Cost of development: The same concept applies here. More development time and staff means more development costs. The ability to develop different platforms with one team will save a lot of resources.

Product life: It is not the same if you plan to make a mobile app that will be used for the next 8 years (mobile bank, Telco, etc.) or a first instance. Indigenous development is safer in long-term projects because the tools and framework are more stable and there are no changes overnight.

Access to Hardware Services: If your app requires full access to all target device resources and services, then native upgrades are the way to go. Although cross-platform applications are trying to integrate the plugin, interaction issues are still common, and the application often ends up taking up a lot of space.

UI / UX Design: Finally, the development of the opposite field also lags behind the knowledge of UI / UX. Because upgrades to the native area provide unlimited access to parts of the UI / UX, native applications often bring excellent visuals and outstanding UX.

Background processes: To allow the app to perform tasks while not pre-existing, it must be native.

App notification: Managing app notification is clever on the cross-platform, the native ones do it much better.

OS updates: It takes a while (a few weeks) until the forum receives the latest OS updates.

Debugging: Cross-platform does not provide debugging capabilities for the same application. Eg. Crashlitics does not work on RN & Flutter.

III. CHALLENGES CONCERNING APPLICATION DEVELOPMENT

When setting up a mobile app, you should focus more on “what not to do” than on “what to do”. Engineers face a number of challenges while working on a new application. A few of them are listed below.

A. App visibility

Because of the increase in applications being developed, there is also a similar increase in competition in the mobile apps market and there is an ongoing pressure to create recognized applications [2]. All that you want is people to download your product. App Store Optimization (ASO) is an important step in releasing a promising app in the world. Also search engine optimization, ASO involves using keywords and phrases to guide the user to the product. The main goal is to get your product visible while searching for a name using search engines like Google. You can look at competing applications in the term you want and look for items in a level program that appear while searching. Signs and colors are another important factor to be noticed. The first thing that draws the attention of any user is the icon used. Try enriching the look and functions make it more attractive to tempt the user to click on your application.

B. Screen Variation

Users have different hardware devices of which screen sizes vary distinctively. Only designing the latest platform is a bad idea. It is necessary to create an application that works on as many different devices as possible [3]. In addition, you need to worry about OS and- Windows, Android and iOS in particular. Each has its own design patterns and UI. For example, Nokia phone may have a small screen with a QWERTY virtual keyboard, such as the Blackberry Curve or it may have a larger touch screen and a virtual keyboard like the Blackberry Storm. The phone requires large buttons for touch screen interaction while the curve will need smaller navigation elements to fit on a smaller screen. It is very important to have a responsive design. It will provide more flexibility and so can be customized to different formats and different screen sizes. To handle technical OS problems, having a comprehensive test section helps. Uninstalling a beta version of the app can also help deal with the problem.

C. Performance vs. Battery Life

App performance and battery usage are also important aspect for considerations after design. The biggest challenge is to design a app that is unobtrusive and uses the least amount of battery possible. This is important as mobile devices have limited processing and power as they use the energy available on batteries. If your app uses a lot of power, chances are the user will stop using the app and try the same system to save battery power. Performance is addressed in the final phase of the design phase of development. In addition, different people view application performance differently. If there are too many images, there may be a repository of storage and it will interfere with operation. The functionality varies due to thousands of different devices all with different computing power [4]. The successful implementation of the app on the target device is a great achievement in itself. Involving the user in the app is a difficult task and any slowness of the app is one of the biggest risks to user involvement. It is believed that the reason people quit apps due to performance issues is almost 52% and is second only to 58% of user interactions.

IV. SUGGESTED SOLUTION

Good idea Start by finding a problem at work and then solving it in a practical way first. This is an important difference between a good problem-solving app and a good problem-solving app in the first mobile way. Generally, the objective is the amount of data collected rather than its availability on a mobile device. This will result in a data-based application but not a user-focused application. After that, the features of your app should be determined. The more it offers features and benefits, the more likely it is that users will be shocked by the app and use it more often. Do not create a mobile app that just pleases the End-user's eye. It has to do something else and will forget about keeping some of the key features needed by –

● Touch Support

Due to the ease of use and simplicity provided by touch screens, it is important that the mobile app should support the touch screen rather than the traditional keyboard / mouse usage. The mobile app should be accurate and made such that it requires minimal attention from the user.

● Working Offline

Good apps should be able to work offline. An IMP feature of mobile is that wireless data services are not available everywhere. That’s why the application must be able to provide certain features in offline mode as well.

● Responsive Design

Responsive design greatly improves user experience. The ability to change the layout according to the screen size is an important factor. The clever design is for the app to provide the user interface with the screen size and generate advanced info from a singular code base so we don't have to worry about details in specific.

● Updates

The app should be provided with bug fix updates and add additional features and services to the app. In this rapidly changing world, the latest problems and needs arise daily and your application must be tailored to the needs of the user in order to achieve their satisfaction.

● Single Source

It would not be a high-quality and expensive thing to send an application to each target area for indigenous stadium development. You will need to have the iOS version, android and one for web browsers. The best way is to have a single development area that focuses on different platforms and different screen sizes.

V. BENEFITS OF ANDROID APP DEVELOPMENT

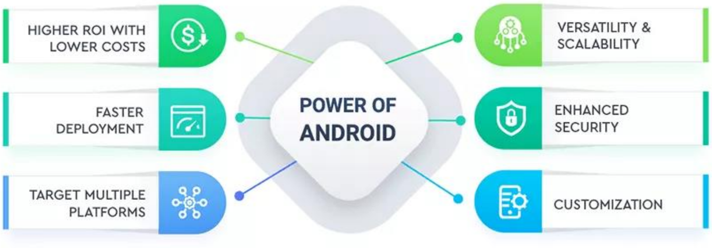

HIGHER ROI WITH LOWER COSTS
FASTER DEPLOYMENT
TARGET MULTIPLE PLATFORMS
POWER OF ANDROID
VERSATILITY & SCALABILITY
ENHANCED SECURITY
CUSTOMIZATION

Figure 1: Advantages of Android

● High ROI with low cost

The most important benefit of app development is the easy availability of the SDK. Project Development can use the key design from these SDKs to build interactive applications. However, teams of engineers / developers are required to pay a registration and distribution fee.

● Quick Shipping

Business Apps have a fast development cycle that lasts for a short period of time. It offers a competitive edge to companies that wish to go to the market quickly with their new idea Reduced Time to Market (TTM), that’s why, is one of the best benefits of Android development.

● Identify More Fields

The use of JAVA language as a coding language makes it easy to install applications on different operating systems such as Ubuntu and various Distros. Businesses can direct multiple forums with the development of the Android app. It is one of the many reasons why organizations are opting for Android upgrades.

● Diversity and Stability

With the advantage of Android Studio, the OS has improved with flexibility. It integrates with the whole Android ecosystem including smartphones, tablets, smartphones, and also Android TV. To make Android Apps compatible with emerging technologies like IoT, AR, and VR there should be other variants of the applications too. Therefore, it is also one of the most important benefit of the android app.

- **Improved Security**

Android P has introduced a few additional and built-in security features. It will help protect you from malware and viruses. Therefore, security and reliability are unique benefits of the development of the android app.

- **Customization**

Android is an open source platform and offers many customizable features for development. That is why android apps are a popular choice.

VI. WHAT IS ARCHITECTURE OF APPLICATION DEVELOPMENT?

Application structure states the patterns and techniques and ways used to design and build an app. The properties give you a road map and best practices to follow when building an app.

Software design architectures help developers build an app. The pattern describes a recurring solution to a problem.

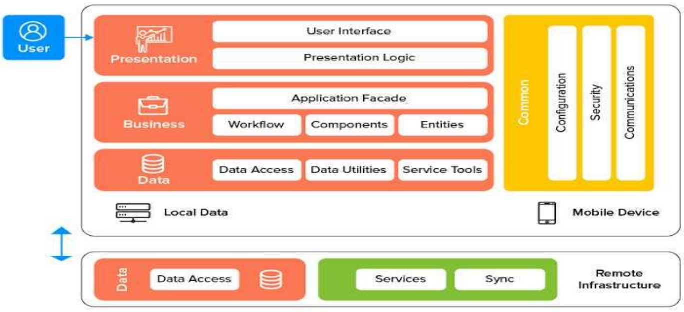


Figure 2: Architecture of Application Development

Patterns can be linked together to create custom application layouts. Developers can use existing design patterns, which also ensures that things will work the way they are intended to. As an important part of the app architecture, there will be both end-to-end and back-end services. Front-end enhancement is related to in-app end-user experience which matters the most, while back-end enhancement focuses on providing access to data, services, and other existing systems that make the app work flawlessly. Architecture is a start or road map for building an app, but you will need to make application decisions that are not taken up in architecture. The initial step is to choose the preferred programming language in which you will develop the application.

There are many programming languages used for software development. Certain languages may be used to build certain types of applications, such as Swift for mobile applications or JavaScript front end enhancements. JavaScript used with HTML and CSS is currently the 1st language of the most popular web application development programs. Other popular editing languages include Ruby, Python, Swift, TypeScript, Java, PHP, and SQL, among others. The language used in the development of the application will depend on the type of application, the development resources available, and the requirements.

Historically, applications were written as a single unit of code, in which all components shared the same resources and memory location. This architectural style is called monolith. Modern application architectures are often integrated seamlessly, using microservices and application programming interfaces (APIs) to connect resources, which provide the basis for cloud-based applications. Indigenous cloud computing is a way to accelerate the way you build new applications, upgrade existing ones, and provide consistent and authentic improvements for automated management across all private, public, and multicultural clouds.

VII. TECHNOLOGY

Frontend upgrade:

The front of the web- or mobile app is the part the user interacts directly with. It is usually called the "client side" of the application. The front area contains everything the user sees when interacting with a website or app, such as text colors and styles, images, graphs and tables, buttons, colors, navigation menu, and much more. Frontend engineers provide the layout, look, behavior, and content of everything from the browser that is displayed when websites, online applications, or mobile applications are opened. Key points to focus on improving frontend response and performance. The previous developer should ensure that the site is responsive, which means that it works well on devices of all sizes. Application performance must be stable at all times, regardless of the device used to access the app.

Front languages: The main languages used for frontend development are HTML, CSS, and JavaScript. They work together to make a website or app work better while looking good at the same time. Apart from HTML, CSS and JavaScript, there are many different languages that can be used for frontend development depending on the framework, such as Dart in Flutter, JavaScript on React, Python on Django, and so on.

Hypertext Markup Language (HTML): HTML stands for Markup Hypertext Language. The language of the language used to design the front of the web pages. HTML is a simple language that combines hypertext with punctuation. The word "hypertext" means a link between web pages. The language of the text is used to construct the architecture of the text under the mark. In this way, the content on the page is structured in the form of headings, headings, subtitles, basic text, images, etc.

Cascading Style Sheets (CSS): CSS is a simple language designed to simplify the process of changing the content of a web page. A piece of CSS code can be easily linked to HTML code, which automatically renders certain styles and is formatted on web pages. CSS allows you to do so without having to worry about the HTML code that forms each web page. Although there is a way to assign styles directly to HTML code, having a separate code on the CSS writing side ensures you do not accidentally waste existing HTML code. Additionally, you can make a format change on a single CSS style sheet, and it will automatically apply to all areas where that particular style is applied across the site. This makes it very easy to maintain a consistent style across all your web pages.

JavaScript: JavaScript is a well-known programming language used to integrate interactive features into websites to make the interface more attractive to users. It is used to improve website performance and run games and web-based software.

Front panels: Frames or libraries in application development provide developers with the right tools to assist them in the development process. The most popular libraries in frontend development are ReactJS, AngularJS, and jQuery.

ReactJS: ReactJS is a JavaScript library for creating fast, easy-to-use and flexible user links. ReactJS is an open source, front-end library based solely responsible for application overview. Facebook manages to keep it up to date.

AngularJS1: AngularJS is a front-end JavaScript framework that is widely used to build single-page web applications (SPAs). It is an ever-changing framework that provides better ways to improve online applications. Stable HTML is replaced by dynamic HTML. An open source project that can be downloaded for free. Add Directions to HTML attributes and uses HTML to compile data

JQuery: jQuery is a free JavaScript framework that makes it easy to manage HTML / CSS documents. JQuery makes it easy to manipulate HTML texts

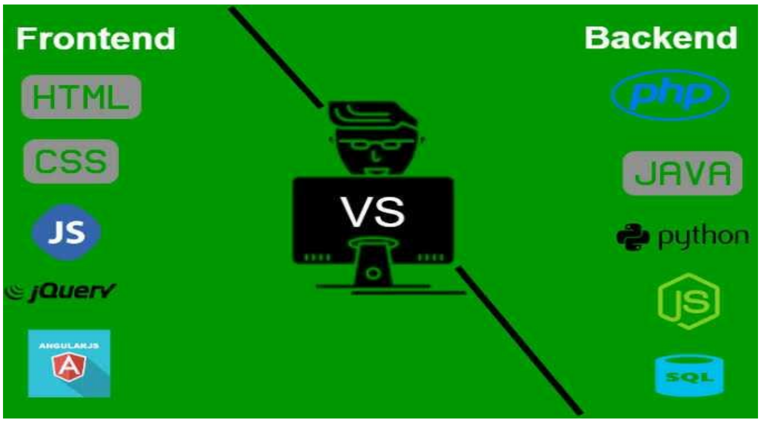
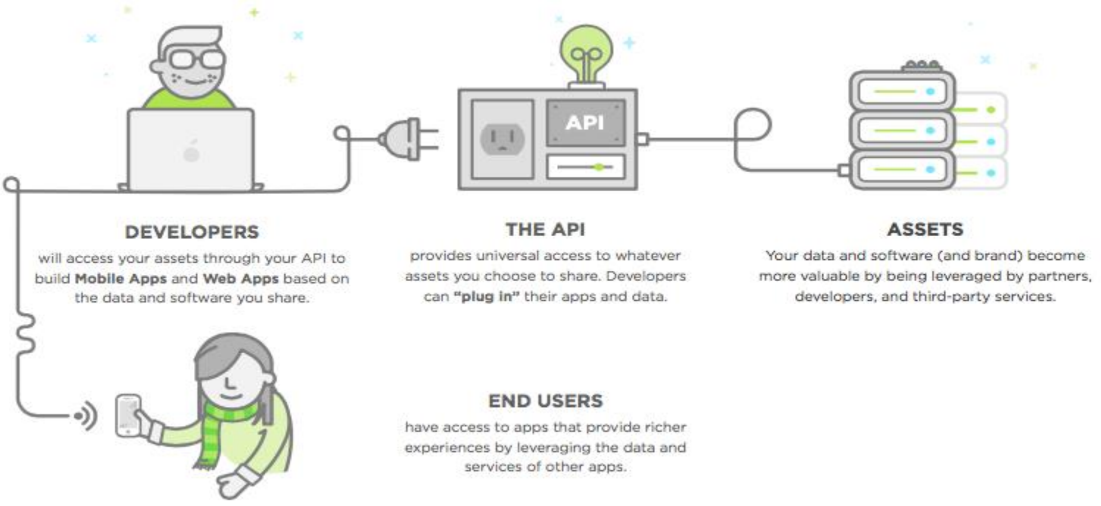


Figure 3: Frontend and Backend

Backend Development:

Backend is part of the website server. Maintains and organizes data, and ensures that everything around the website client works properly. It is part of a website that you can see and communicate with. It is part of the software that does not communicate directly with users. Features and features developed by backend designers are indirectly accessed by users through a pre-app. Tasks, such as writing APIs, creating libraries, and working with system components without user interaction or science program systems, are also included in the background.



DEVELOPERS

will access your assets through your API to build **Mobile Apps** and **Web Apps** based on the data and software you share.

THE API

provides universal access to whatever assets you choose to share. Developers can **"plug in"** their apps and data.

ASSETS

Your data and software (and brand) become more valuable by being leveraged by partners, developers, and third-party services.

END USERS

have access to apps that provide richer experiences by leveraging the data and services of other apps.

Figure 4: Backend Development

Background languages: The back section is built using the other languages discussed below:

PHP: PHP is a server-side scripting language specifically designed for web development. Since PHP code is generated on the server side, it is also called server-side scripting language.

C ++: A common programming language that is common and widely used today in competitive systems. It is also used as a background language.

Java: Java is one of the most popular and widely used editing languages and platforms. It sounds great. Parts of Java are easily accessible.

Python: Python is a programming language that allows you to work quickly and integrate applications successfully.

JavaScript: JavaScript can be used in both editing languages (front and back).

Node.js: Node.js is an open source and cross-platform for using JavaScript code without a browser. You need to remember that NodeJS is not a framework, nor is it a programming language. Most people are confused and understand that it is a framework or language of planning. We often use Node.js to build background apps such as APIs such as the Web App or the Mobile App. It is used in the production of big companies like PayPal, Uber, Netflix, Walmart, and so on.

Back End Frameworks: List of backgrounds are: Express, Django, Rails, Laravel, Spring, etc. Other background / writing languages are C #, Ruby, REST, GO, etc.

IDE: Integrated development area is a combination of all the tools you need to write and test software. Engineers use many tools throughout their development cycle including text editors, code libraries, moderators, and test platforms. Now imagine that you have to select, extract, compile, and manage all of these tools separately. It's crowded, time-consuming, and frustrating. This is where these IDEs come in and bring many development tools to a single framework, application, or service - making coding, finding and removing bugs or errors easy. IDE can be open source or commercially available at a tag price. In addition, it can serve as a standalone application or as part of a comprehensive package.

Android Studio: Android Studio is the official IDE for Android. It is a software suite developed by Google and has all the built-in tools to build a high quality Android app. Android Studio is best known for its ability to speed up the development process while not losing any quality. If you are new to Android development, Android Studio has a lot of apps for building apps. You will easily capture android mobile conversations and business development.

Eclipse: Eclipse is one of the most widely used Java IDE applications that provides customizable working environment and an expandable plug-in system. Written in Java, this Integrated Development site is very popular among Java development companies. The Eclipse Software Development Kit (SDK) includes a wide range of Java development tools to help Java developers build solid applications.

Xcode: Xcode is Apple's integrated development platform (IDE). It is used to build applications for Apple products, such as the iPad, iPhone, Apple Watch, and Mac. It is a powerful tool that allows you to control the flow of your development work from start to finish; that is to create your app, test it, configure it, and send it to the Apple App Store.

VIII. CONCLUSION

Mobile applications have become an integral part of our daily lives because of the variety of functions they offer. Building a successful, trouble-free and easy-to-use application is essential due to the rapid increase in the number of applications. The developer should consider the challenges he faces and try to overcome them by following the appropriate steps. Also, it is important for the developer to be open-minded and well-informed about current technologies, needs and events in the field of mobile application. In order to build success, all guidelines must be properly considered and properly followed to avoid the risk of losing users due to crashing / crashing applications [12]. Developing new and innovative applications will lead to greater profit potential. There are also a few research studies conducted that could serve as a platform for future research sites.

In its face, Android seems to be the most successful platform. An operating system that enables 85 percent of new smartphones shipped, and, along with iOS, completely eliminates competition. Android is also resisting the problems that Google.Inc does not seem to be solving. Developers must develop software that will work reliably on many different devices from various manufacturers. Sure, developers can focus on devices from major companies, but that defeat the purpose of having a platform in the first place. Then there is the problem with the update. Google has an annual release plan for Android, but it takes a new version for a good part of the year to break the 10 percent usage mark. And it takes about four years for the release to fully enter the ecosystem. While Google is quick to discover new releases on its Nexus Hardware, it can take months for owners of Android devices from other manufacturers to see the update. And many do not see the revision. Without buying a Nexus device, the easiest way to get your hands on a new Android release is to buy a new device that is fresh in the market.

IX. REFERENCES

[1] Mobile App, https://en.wikipedia.org/wiki/Mobile_app

[2] Get App Noticed, https://www.forbes.com/sites/karstenstrauss/2014/09/23/how-to-get-your-app-noticed-wisdom-from-app-annie/#316e98cd3675

[3] Ian Darwin, "Android Cookbook -Problems and Solutions for Android Developers", O"Reilly Media, May 2017

[4] Performance of App, https://www.safaribooksonline.com/library/view/high-performance-android/9781491913994/ch03.html

[5] Mobile App Design, https://www.noupe.com/imho/the-importance-creative-design-for-mobile-app-user-experience.html

[6] Ryan Cohen, Tao Wang," GUI Design for Android Apps", apress.open

[7] UI and UX design, http://colure.co/the-importance-of-ux-and-ui-in-mobile-app-design/

[8] Venkata N Inukollu, Divya D keshamoni, Taeghyun Kang and Manikanta Inukollu, "Factors Influencing Quality Of Mobile Apps: Role Of Mobile App Development Life Cycle", International Journal of Software Engineering & Applications(IJSEA), Vol.5, No.5, September 2014

[9] Prasant Kumar Pattnaik, Rajib Mall, "Fundamentals of Mobile Computing", PHI Learning Pvt. Ltd, New Delhi-2012

[10] Prof. K.D. Tamhane, Mr. Wsim T. Khan, Mr. Sagar R. Tribhuvan, Mr. Akshay P. Burke, Mr. Sachin B. Take," Mobile Learning Application", International Journal of Scientific and Research Publications, Volume 5, Issue 3 ,March 2015

[11] Ryan Cohen, Tao Wang, "Android Application Development for the Intel Platform", apress.open

[12] Zigurd Mednieks, Laird Dornin, G. Blake Meike and Masumi Nakamura, "Programming Android" ,O"Reilly