IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

Bridging The Gap between Android and iOS by Analysing their Incompatibility

Charu Anant Rajput[1], Sanjana Godiawala[2], Dr. Maheswari R[3]
Students, School of Computer Science Engineering, Vellore Institute of Technology, Chennai, India[1,2]
Professor, School of Computer Science Engineering, Vellore Institute of Technology, Chennai, India[3]

Abstract: *In the current generation there is widespread use of mobile devices. Most of them are based on the two widely known operating systems namely iOS and Android. There are few instances where some applications are compatible with Android while others are compatible with iOS, also there are few which work equally fine with both the operating systems. In this research paper we try to highlight the role of operating systems in application development and the reasons for their incompatibility. Along with this we have tried to find out solutions in order to establish a technological harmony between the two operating systems. In this paper we have also mentioned the comparison between the two operating systems and their specifications for application development.*

Keywords: Operating System, Android, iOS, Application, Incompatibility, Comparison, Analysis, Convertor, Swift, Kotlin, Xamarin, Android studio, XCode, Architecture

I. INTRODUCTION

In the current era where the mobile phones are one of the essential devices in our lifestyles. In today's era mobile phones have been developed in such a way so as to handle multiple functionalities thereby covering general as well as professional use cases. There are different operating systems in the market amongst which the two most common ones are Android and iOS.

While using these operating systems you must have encountered instances where in few applications are functioning on one operating system while not functioning on the other. There are multiple reasons for this incompatibility amongst the two operating systems which are further discussed in this paper.

There are multiple ways to develop an cross platform applications as they are more profitable since it works on both the operating systems, also it increases the market reach as the number of users would increase, this also saves time as it minimises the requirement of developing two independent applications but this restricts to the frameworks used as there are limited frameworks available for both the operating systems. Hence independent applications are more preferable, as they are specific to the operating systems as well as they are more reliable and can easily adapt to the software updates as the architecture and concept design is uniform. For creating applications independently they require it to be coded in different programming languages using different frameworks and structures.

In this research paper we aim to find these programing differences between the two programming languages namely java used for Android development and swift used for iOS development, as by understanding the differences the conversion process between the applications can be eased out as the developers only need to modify the structure of the code based on the language instead of developing from scratch. In this paper we have incorporated analysis of both the operating systems and the analysis challenges faced for porting the application as well as pros and cons of using cross platform development resources like Xamarin for application development. The paper ends with illustrating the different steps required for converting an application along with the structural differences present between the two programming languages in order to provide means for faster and easier conversion.

II. ANDROID OPERATING SYSTEM

Andy Rubin founded Android and then in 2005 Google acquired and developed it as a mobile application based on a linux kernel. This operating system was made primarily for touch screen devices, phones and tablets with the capability of gesture control actions like touching, pinching and swapping to name a few. Recently, Android has gone through a lot

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

of updates wherein it saw the inclusion of new features. Each of the version has been names after a dessert following the alphabetical order.

2.1 Architecture

The Android Operating System primarily has all the software components arranged in the form of a stack divided into five sessions and four layers. One of the layers namely the libraries has one more special section dedicated for android development namely the "Android Runtime". The architecture can be broadly classified and describes as follows

1. Linux Kernel: This is the lowest layer and forms the base of the architecture. It has direct access and control over the hardware drivers like camera, keypad and display as well as interaction with other peripheral devices.
2. Libraries: On top of the Linux Kernel sits the collection of open-source libraries.
3. Inside the library layer we have yet another section that has dedicated android libraries which majorly contributes in developing the core functionality of the apps.
4. The second layer is divided into two main parts; one was the library as discussed above and the other is the Android runtime which has Dalvik Virtual Environment as its key component which is similar to a JVM that significantly and critically contribute to the compilation of Android Codes.
5. The third layer is called the Application Framework which includes Activity Manager, Content Provider, Resource Manager, Notifications Manager and View Systems as the key services. Overall control over the application lifecycle is administered by the activity manager whereas the data sharing and publishing part is handled by the content provider. Resource manager is responsible for interface settings like colours, strings and layout settings. Notifications manager controls the highlighting of incoming alerts and Views System provides a vast array of views which can further be used for creating application user interfaces.
6. The final and the topmost layer is the Applications layer which is where the user/developer writes and deploys their custom build applications.

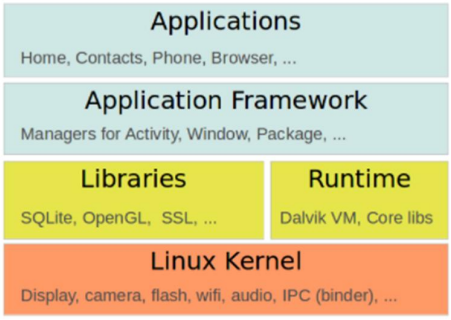

Figure 1. Android Architecture

2.2 Advantages

On holding a comparative study between the performance of IOS developed application and Android developed application it is observed that Android has better and faster server connectivity as compared to its counterpart. Also Android provides better user experience given to the enhanced ability of customization. According to [1] Android has been proven to support greater stability to apps as compared to IOS.

2.3 Drawbacks

Android is a little slow and laid back when it comes to displaying the UI onto the screen in comparison to Apple. Also in terms of privacy and user security Android is slightly insecure as Android is open source and used by many devices and therefore there is always a risk of data sharing to the respective companies, whereas in case of Apple the user data is encrypted and the OS is applicable only on native devices.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

III. IOS OPERATING SYSTEM

Apple iOS is the operating system originally developed for iPhones but now is used for iPads, iPod Touch. iOS uses a multi-touch interface in which simple finger gestures like, swapping, dragging, tapping, pinching are used to navigate and operate the device which makes it easier to use. Apple Inc. has been providing updates to it's operating systems throughout the years for its devices namely iPad, iPod, iPhone. iPhone OS 1 was amongst the earliest released OS corresponding to iOS and iOS is the most recently launched.

3.1 Architecture

The iOS operating system has layered architecture with broadly 4 layers. Since it is a layered structure there is no direct communication, instead communication takes place through intermediate layers. The lower layers are for basic services and the higher layers are for graphics and the interface related services. The architecture is as follows:-

1. Core OS Layer: - This layer provides all the features required for building all the iOS technologies.
2. Core Services: - It contains frameworks that help the operating system to cure itself and provide better functionality.
3. Media Layer: - This layer provides and enables all the graphics, videos and audio related technology and contains the required frameworks for the same.
4. Cocoa Touch: - This contributes as the primary interaction utility for users. Gesture control commands are interpreted and executed by this component.

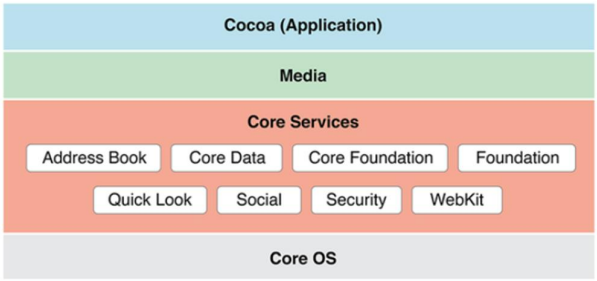

Figure 2: iOS Architecture

3.2 Advantages

The main advantage of iOS is that it is more secure than other operating systems, secondly it has an excellent user interface which is uniform, convenient and user friendly. iOS also has a larger number of peripherals to which it can be connected like iPad, AppleWatch, MacBook, Homekit, etc. It generates less heat as compared to android devices.

3.3 Disadvantages

The main disadvantage of iOS is the lack of flexible as it is compatible only with iOS devices and applications from App Store, even though it has a option to jailbreak and download applications from different sources but that is also restricted to a certain level. iOS is usually more costly than Android, it is restricted to Apple oriented devices and applications.

IV. SYSTEM DESIGN AND IMPLEMENTATION

4.1 Android

A. Programming Language

Kotlin and Java are the two popular languages used for android development. But apart from it languages like C#, Python can be employed for android development.

(i) Java -Java Micro Edition provides a compatible development environment for the building of apps aimed to be launched on mobile and embedded devices. Apps designed in Java are secure and also have the capability to take advantage of the native hardware device.

(ii) Kotlin - This is a relatively new language whose first edition was announced in 2016 and in 2017 it was declared as the first language for Android Apps by Google. Kotlin can be termed as a simpler and enhanced version of Java which is

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

specifically aimed at android development as it is more concise, reduces boilerplate code and is also able to handle errors such as null pointer exceptions. Apart from this it has multiple dedicated libraries for Android app development.

B. Development Environment

While there are many development environments available in the market we are going to describe the three most popular IDE'S here and they are as follows

(i) Android Studio: Has been announced as the official integrated development environment for all android applications, it was developed by Google in 2013 and primarily uses Eclipse Android Development Tools(ADT) as the primary IDE.

(ii) IntelliJ IDEA is a general-purpose IDE wherein the Android development can be started by making use of the Android Plugin.

C. System Specifications

Android apps are developed using partitions which break the application into fragments and activities. An activity is equivalent to an application screen while fragment is the part of the user interface that is used to navigate between different activities.

D. Hardware Requirements

A 64-bit environment is necessary for the working of the lowest version of Android that is GingerBread. Apart from this at least 250 GB of free disk space is required for debugging the code along with an extra 150 GB for app building.

E. Software Requirements

JDK is the first and foremost requirement for starting with the installation of development environments. For working with higher versions of Android like Oreo starting from Lollipop, installation of Dockerfile will be useful for importing and installing android library packages.

F. Development Complexity

In terms of development complexity, Android lacks behind iOS because of its fragmentation. In simpler terms, android apps have to be tested and deployed onto a vast array of devices with different device specifications and screen ratios and therefore the developer has to take care of all of this in order to get an even output across all the devices. However, the use of Kotlin as the development language has its own advantages as it's completely interoperable with Java and is way more concise than it. Secondly it offers safer codes owing to the reduced number of system failures encountered and also most of the error detection takes place at compile time itself. Furthermore, the development effort is about 30%-40% more as compared to iOS development. However the procedure of publishing the apps onto google play store is quite easy and smooth as the approval for app publication is usually received within hours.

4.2 IOS

A. Programming Language

There are 3 mainly used programming languages for iOS application development. Namely they are: - Swift, Objective C and C#. Apart from these other languages like HTML5, Java, React Native, and Flutter are also used.

(i) Swift: - this language was developed by Apple in 2014 as a successor to Objective C specially to work with frameworks like Cocoa, Cocoa Touch and extensive codebase written in Objective C. It has become the 17th most popular language to be used by developers worldwide in 2020, as it has good scalability, it is highly readable and easy to use, it can easily be interoperable with Objective C, it is stable and reliable. Compared to other languages it is quite young.

(ii) Objective C: -It can be assumed to be advancement of the C programming language with the inclusion of features concerning the principals of object-oriented programming. It is used as a general-purpose language for programming operating systems like OS X and iOS along with their APIs Cocoa and Cocoa Touch. It is the 20th most common language used by developers in 2020. It is stable and includes dynamic typing but on the other hand it is difficult to learn, provides

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

less security, has limited functionalities but it is highly used for creating apps for all the versions of iOS, since Swift doesn't support few older versions.

B. Development Environment

Xcode tool :- provided by apple is mainly used for developing iOS applications, as it supports swift as well as provides a variety of tools for debugging, and also offers support for a wide range of iOS devices with a user friendly interface. The only problem is that it can only be used in Mac OS and not on other desktop operating systems.

In the case of iOS app development, even if other environments like Xamrin can be used, they still require MAC OS to execute the code. For other operating systems it is preferred to use a virtual box with Mac OS and Xcode to develop iOS applications hassle free.

C. System Specifications

iOS application architecture relies on view controllers like page view, tab view, split view and many more. The view controllers basically control the entire screen or one of its parts, and are managed by writing them in code or organising the images in a storyboard and storing it using a XML file. Due to this iOS architecture is less error prone, manageable and doesn't require sizing for different screen specifications of iOS devices hence making it easier to develop applications.

D. Hardware Requirements

The minimum hardware requirement for the development of iOS applications is there should be a Intel i5 or greater CPU, 4 GB RAM, 128 GB or more Disk Storage and a MacOS 10.14.4 or later macOS version.

E. Software Requirements

The software requirements for iOS application development is having a stable version of Xcode running on your Mac. Xcode is Apple's official IDE(Integrated development environment) which includes the iOS SDK(software development kit), tools, compilers, and frameworks required for development.

F. Development Requirements

The software requirements for iOS application development is having a stable version of Xcode running on your Mac. Xcode is Apple's official IDE(Integrated development environment) which includes the iOS SDK(software development kit), tools, compilers, and frameworks required for development

V. COMPARISON

The birds eye view of the comparison between the two operating systems is mentioned in the following table:-

TABLE I: Comparison between Android and iOS

iOS	Android
Developed by Apple Inc. in 2007	Developed by Google in 2008
Mostly used in iPhones, iPads, Apple TV, iPods.	Mostly used in smartphones and tablets.
Designed specifically for Apple Devices	Is open for use irrespective of the smartphone companies
Kernel type is hybrid of OS X and Unix	Kernel type is Linux based
Programming language is mostly Swift and Objective C.	Programming language is mostly Kotlin and Java.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

UI rendering is better as compared to Android	Server connectivity is better compared to iOS
Highly secured	Less Secure than iOS
Can mainly download applications from AppleStore only.	Can download applications for open source resources and google play store.
Lesser development complexity due to apple native deployment.	Greater development complexity due to deployment on a vast array of devices.
iOS uses Siri(Voice Assistant), Safari(Web browser), iMessage (Messaging), iTunes, AppStore, iCloud,Mail as it's featured applications.	Android has google based components like google assistance (voice assistant), Chrome (web browser), Gmail, Play Store, as it's featured application also it provides options to use other applications.

VI. REASONS OF INCOMPATIBILITY

One of the main reasons for the incompatibility is the difference in the operating system and its architecture. Along with this the programming language used for the application development is different and hence it is difficult to compile given the different nature of runtime environments. Secondly, there can be a lack of resources for porting or has compatibility issues for porting. There are few features provided by one operating system while other does not, this requires adding extra features in order to make the application run appropriately. Switching the platform also requires use of different frameworks which causes issues while reusing the code as the developer needs to find the alternative supported by both the platforms, this also adds up to the cost for application development. Another major challenge is the software updates, as something few features are not supported by the operating system, for instance say there is a iOS update that updates/adds a new element, so the application needs to be updated accordingly, but also one needs to wait for Google to release a new update to avail the same functionality in both the operating systems. Secondly creating cross platform applications leads to sluggish code generation which in turn results in reducing the speed of the application.

VII. EXISTING SYSTEMS

There are multiple frameworks like Xamarin, Flutter, React Native, etc. which provide facilities for cross platform application development. But there are limitations to the design and user experience that is achieved using these frameworks. Taking Xamarin for instance it uses C# programming language for application development that is supported by both iOS and Android, but these are the few basic differences in the user experience found for apps developed in Xamarin. The use of Xamarin for development prompts the issue of large negative spaces and the absence of "Back" button for iOS apps while on the other hand it gives enhanced clarity and precision for the Android apps. In terms of application programming, it becomes for Android given the fact that it has to cater to a wide variety of screen sized and ratios. This in turn leads to fragmentation of Android versions, unlike iOS where the situation is convincingly easier given the availability of standard which in turn proves to be time saving for the developer's effort.

Looking at the advantages of cross-platform development, reusable code is one of the greatest advantages offered as it has many interlinked benefits both for the developers as well as the client. Reusable code reduces the development time to a great extent and is also proven to be cost effective as compared to the single platform development. These apps tend to have greater market reach and are also easy to update and edit as a standard code base is employed across all the platforms.

However, switching the platform proves to be a challenge as each platform uses its native JavaScript subsets that might lead to an unstable code. Besides, a lengthy integration process is required to make the app adapt to local settings and preferences. Major use of cross-compliance techniques during the development process leads to the formation of a sluggish code and thereby reducing the execution speed of the application.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

Alternative solution for portable application development is using an independent runtime environment for that application so that it becomes independent of the host operating system, basically it would be operating on its virtual machine.

VIII. DISCUSSION

From the above discussion, the main 4 challenges faced by the developers are the designing for different device sizes, stability with the OS versions and updates, different orientation of buttons and icons, and the programming language. These can be targeted for carrying out the conversion between the two operating systems are architecture, UI/UX design, implementation and testing. To begin with, the architecture of the new cross platform app is to be visualized for the sake of constructing a clear base about the future developments in functionality. The next step is to build a new UI/UX design while ensuring the configuration stability of the existing apps. The milestone step is implementation wherein the final porting ensures its adaptation with the existing hardware devices and local settings. As the last step we need, the code thus generated to go through the testing phase following the guidelines of beta testing, integration testing and validation testing.

In terms of the development environment used, the coding of iOS applications is done in XCode which creates the supporting files in JSON format. The programming language is Swift which is kind of similar to javascript, when it comes to basic programing and formatting the application. The UI design is created using view controllers which mainly are VStack, Hstack, Spacer, etc for formatting the structure and appearance of the application, due to this it is self adjusting to different device sizes. There are functionalities available to drag drop the required components and by changing the values using the GUI, which gets reflected in the code directly.

In the case of Android the software Android Studio has been used as the development environment for the Android application. The coding has been carried out using java which can be termed as the parent language of the android official language, Kotlin. Here the layout of the screen was written and stored in the form of an xml file while the functionality of the app was developed inside another java class. Android studio provides provision to modify the UI rendering by either modifying the xml file in “code” mode or “design” mode which in turn basically is GUI wherein the app components can be readily dragged and dropped into the workstation in order to render the basic structure and appearance of the application.

Several libraries need to be imported from android package and androidx package while in swift we can get all the libraries by just writing the “import SwiftUI” statement. The code in android starts with a class while it starts with a struct in case of swift which puts light on the fact that Android application development closely resembles Java object-oriented concept while in case of iOS the development coding closely resembles Objective C coding style.

Next difference is noted in the mode of variable declaration. In the case of Java we are using in-built classes like Button, EditText, TextView, etc available in the imported libraries in order to create objects like buttons, input text field, label display, etc that in turn create the components of our application, and the formatting of those components is done in the xml file. On the other hand, in the case of Swift we use syntax similar to javascript for creating and formatting the components.

For converting the user input to the required data types in case of Java is done by making use of wrapper class Integer and subsequently the conversion involved using the method functions “toBinaryString”,” toOctalString” and “toHexString” all of which required the input decimal entry to passed as the argument. In case of swift, the parsing of text input to integer has been done by explicitly specifying the data type and for the conversion, the corresponding number is converted to string using appropriate radix parameter argument.

In case of Swift for managing the UI of the application different view controllers are used like VStack, Spacer, etc which control the position and the layout of the components on the screen, while in case of java it is done using the XML file.

IX. PROPOSED MODEL AND FUTURE ENHANCEMENT

The applications developed in both the operating systems have predominantly two parts, one which deals with imparting functionality while the other imparts the design and user interface element. In case of Android it is stored in “.java code file” and “.xml design files” respectively while in case of iOS it is in the form of “.swift code file” and “.json supporting

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

file" respectively. Since the ".xml file" and ".json files" are automatically generated after dragging and dropping the app components in the designing workspace, therefore here we are feeding only the functionality imparting code files i.e ".java code files" and ".swift code file" into the code file processor. The code file processor with the help of file handling techniques will scan through both the files. Then we bring an algorithm that would make out the critical section of the code that is solely responsible for imparting the core application functionality.

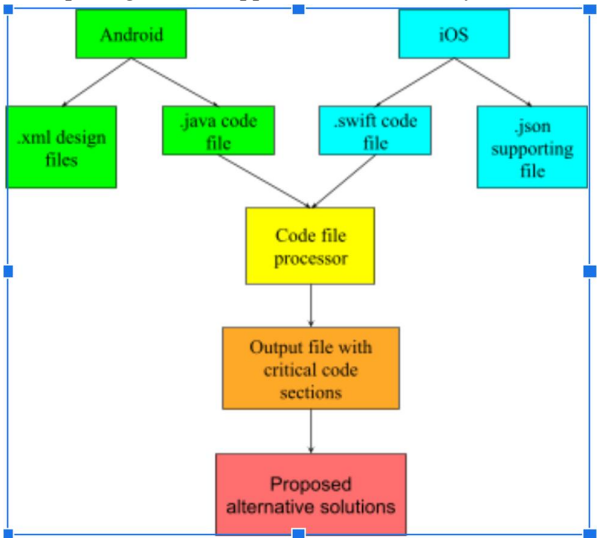

Figure 3: Flowchart of the proposed model

According to this proposed algorithm the code can be divided into three main parts :

(i) Library importing part

(ii) Instance declaration part

(iii) Event handling part

The proposed algorithm can basically work on the above mentioned parts sequentially and consequently identify the equivalent replacement in each of the above mentioned three parts. This will in turn lead to precise targeting and consequently accurate output production of the critical lines where the modification has to be made.

X. CONCLUSION

From the above discussions it can be concluded that there are multiple differences between the two mobile operating systems Android and iOS. The applications that are developed using Swift / Objective C in case of iOS which resembles C and JavaScript coding structure. While in the case of Android it uses Kotlin/Java which are inter-operable. One of the common approach present for creating portable applications is using cross platform sources like Xamarin which use programming language C# that is supported by both the operating systems but that too comes with it's own limitations. Hence the approach we are suggesting after studying the comparisons is to display to the developer specifically those lines of code which are the main contributor for functionality and UI design but differ syntactically and semantically in both the development environments. Once these points of differences are figured out, the cross-development speed would fasten. This can be achieved by using file handling as a tool. The code in both the environments can be converted into readable files conveniently of .txt format in our case and then after performing the read operation along with some comparison algorithms we would be able to display the target code section as the output. By doing this the developer will only have to modify the basic structure and syntax as the base syntax is almost the same. The main reason for proposing this kind of converter is to overcome the limitation of cross platform application developers by creating two separate independent applications which using this can be created in a shorter time span as the developer has the rough layout of the application and is not required to code from scratch. Also while integrating the application to function as a cross-platform application, the developer will already get the critical section in hand and will not be required to figure it out from the rest of the sluggish code.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

XI. FUTURE ENHANCEMENTS

The project can be further continued by creating the converter by finding all the syntactical and semantical differences and the corresponding alternatives for the simpler application development process. If not the entire converter we can develop a documentation which contains the required information about the alternatives in both the operating system, so it basically contains the java code as well as the swift code for performing the same task in both the programming languages. It would be great help for the developers if along with the documentation, the converter is also created as it would highlight the critical sections so that the other application can be developed in an easier and faster manner.

Alternatively, this project can also be used by the cross-platform applications development sources as they can also get aware about the limitations and the drawbacks of their system, and improve accordingly and overcome the mentioned limitations, as to provide a better development environment and application development that is more efficient, supportive and portable. Hence it can be used to analyse the Android and iOS operating systems and find means for creating applications that are suitable for both the operating systems as well as device methods through which the application development process can be improved and simplified.

XII. ACKNOWLEDGMENT

We would like to extend our sincere gratitude towards, Dr. Maheswari R, our co-author and guide for the development of this research work. We would also like to thank School of Computer Science Engineering (SCOPE), Vellore Institute of Technology, Chennai Campus for providing us with the opportunity and the necessary knowledge resources for carrying out the research work.

REFERENCES

[1]. Lazareska, Lazarela. "Analysis of the Advantages and Disadvantages of Android and IOS Systems and Converting Applications from Android to IOS Platform and Vice Versa." American Journal of Software Engineering and Applications 6, no. 5 (2017): 116. doi:10.11648/j.ajsea.20170605.11.

[2]. Gyorödi, Robert, Doina Zmaranda, Vlad Georgian, and Cornelia Gyorödi. "A Comparative Study between Applications Developed for Android and IOS." International Journal of Advanced Computer Science and Applications 8, no. 11 (2017). doi:10.14569/ijacsa.2017.081123.

[3]. Tracy, Kim W. "Mobile Application Development Experiences on Apples IOS and Android OS." IEEE Potentials 31, no. 4 (2012): 30-34. doi:10.1109/mpot.2011.2182571.

[4]. Khalid Lamhaddab, Mohamed Lachgar, Khalid Elbaamrani, "Porting Mobile Apps from iOS to Android: A Practical Experience", Mobile Information Systems, vol. 2019, Article ID 4324871, 29 pages, 2019. https://doi.org/10.1155/2019/4324871

[5]. "Operating Systems - Technical Implementation (software) - Higher Computing Science Revision - BBC Bitesize." BBC News. Accessed January 2021. https://www.bbc.co.uk/bitesize/guides/zprkd2p/revision/1.

[6]. "5 Major Differences Between IOS and Android App Development." EGO Creative Innovations – We Develop Mobile Apps for Businesses. https://www.ego-cms.com/post/5-major-differences-between-ios-and-android-app-development.

[7]. Chen, James. "Android Operating System: What You Need to Know." Investopedia. May 07, 2021. https://www.investopedia.com/terms/a/android-operating-system.asp.

[8]. Kenton, Will. "Apple IOS." Investopedia. September 17, 2020. https://www.investopedia.com/terms/a/apple-ios.asp#:~:text=Apple (AAPL) iOS is the seamless networking between Apple products.

[9]. Nations, Daniel. "What Is IOS?" Lifewire. November 09, 2019.. https://www.lifewire.com/what-is-ios-1994355.

[10]. "Requirements : Android Open Source Project." Android Open Source Project .https://source.android.com/setup/build/requirements.

[11]. Editor. "Top 20 Tools for Android Development." AltexSoft. February 28, 2020. https://www.altexsoft.com/blog/engineering/top-20-tools-for-android-development/.

[12]. Kaczorowski, Miłosz. "Picking The Best Language For IOS App Development In 2021." Agile Software

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, December 2022
Impact Factor: 6.252

Development Agency in Europe. https://www.ideamotive.co/blog/picking-the-best-language-for-ios-app-development.

[13]. "Architecture of IOS Operating System." GeeksforGeeks. March 15, 2021. https://www.geeksforgeeks.org/architecture-of-ios-operating-system/.

[14]. "Pros and Cons of Mobile App Development Using Cross-Platform Technologies: GoodWorkLabs: Big Data: AI: Outsourced Product Development Company." GoodWorkLabs. December 10, 2019. https://www.goodworklabs.com/pros-and-cons-of-mobile-app-development-using-cross-platform-technologies/#:~:text=One of the limitations of,same application on different platforms.

[15]. "Cross-Platform Mobile App Development: Challenges and Opportunities." VertexPlus Blog. February 18, 2020.. https://blog.vertexplus.com/cross-platform-mobile-app-development-challenges-and-opportunities/.

[16]. "Convert Android App to IOS: Android App to IOS App." JumpGrowth. May 04, 2021. Accessed https://jumpgrowth.com/convert-android-app-to-ios/?utm_source=&utm_medium=&utm_campaign=&utm_content=&utm_term=.

BIOGRAPHY

Charu Anant Rajput is a final year engineering students pursuing a **bachelor's degree** in **Computer Science and Engineering** at **Vellore Institute of Technology, Chennai**. She is interested in carrying out research concerning the domains of medical image processing, machine learning and deep learning. Apart from this her previous research works also involve the domain of cloud computing as well.

Sanjana Godiawala, final year engineering student pursuing **bachelor's degree** in **Computer Science and Engineering** at **Vellore Institute of Technology, Chennai.** She is interested in carrying out research in the domain of Artificial Intelligence and Machine Learning, and have worked with their applications in medical assistance. Along with this she has worked with application development and cloud computing.

Dr. Maheswari R M.E., Ph.D, Professor, Head of the Department, CSE- Cyber Physical Systems, SCOPE, Centre for Smart Grid and Technologies, Vellore Institute of Technology, Chennai. She has professional experience of **22+ years in Industry and in various prestigious institutions**. She has published **seven patent** and 50+ research works in various Book, Book Chapters, International Magazines and Journals in her research domains AI, ML, IoT. Has received many awards like **Outstanding FOSSEE Contributor Award-IIT Bombay & MHRD, Govt. of India**, Best Faculty Award, Best Achiever Award, Best Researcher Award, Best Alumni Chapter Award, Best paper and Excellent Paper Awards, Best Club Coordinator award. Acted as resource person, panel member, chief guest, guest of honor and given plenary talk in various industries, International and National Institutions as a part of training, seminars, workshops, conferences