e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

FLUTTER AND FIREBASE MAKING CROSS-PLATFORM APPLICATION DEVELOPMENT HASSLE-FREE

Jashandeep Singh*[1], Swapnil Srivastva*[2], Dipanshu Raj*[3],
Shubhampreet Singh*[4], Mir Junaid Rasool*[5]
*[1,2,3,4]Student Department Of Computer Science And Engineering LPU Phagwara, India.
*[5]Assistant Professor Department Of Computer Science And Engineering LPU Phagwara, India.

ABSTRACT

The thesis focuses on the modern-day problem of cross platform mobile development, that is met with compromises. The developers either need to choose between creating multiple code bases of the same app various operating systems used or they are forced to accept a less optimal solution that handicaps the application by limiting its compile time, accuracy and UI modifications. This is where Flutter steps in, it is a Google's UI toolkit that is used to develop beautiful applications that are natively compiled. Besides the features of development technologies, choosing the correct database is also vital. Firebase is a sturdy BaaS which handles the problem of storing huge unstructured data and is also relatively faster when compared to conventional RDMBS. This paper tries to demonstrate the advantages of using these newer technologies and selection of best framework for making a learning management application for students of Lovely Professional University.

I. INTRODUCTION

We all are working on an app called 'Discipulus' which is an app to connect all the students of the University. We are interested to make an app which can connect all the Yertos and let them share all the knowledge that they have. Student can also download all the notes related to any Course Code. It can connect to any Vertos registered with app and can be used to match students with same field of Interest. It uses gps, location to find someone nearby which can Increase the idea of Collaboration.

1.1 ABOUT THE APPLICATION

It can help students with high interest in competitive programming to learn together. Furthermore, the app will provide an interface for getting the study material. This interface will be having various modules that will be providing peers with access to viewing and downloading course material which will comprise of notes, presentations. Another feature that is added to this interface will be videos relates to topics that will describe the topics in an elaborative way and links will be provided in various languages so that the viewers can grasp every necessary point. Lastly, users will be given an option to share their own notes if some course needed better explanation and provide better video assets which will be then stored in the database and displayed in the app. Another module added is Event registration. In this module, students and faculties will be shown a calendar for the current academic session, in which various technical and non-technical activities that are going to take place, would be available. These are segregated based on event type, for example. coding event, singing competition, etcetera. We will also build a website for this project. This platform in form of website would be solving this problem for students by providing them with a well sorted list of companies visiting the university offering either full time jobs or internships. All the openings for jobs would be presented in a way that anyone looking for a job/internship in their field of specialization would get all the information regarding the venue for the interview, cutoff, package offered, where to apply, etc. in a well sorted way which would not be requiring them to read the whole announcement. This website would also be displaying the previous students along with their social media links who got placed in the companies visiting the university. This would allow the current applying students to get in touch with their seniors and get benefits from their experience of the interview and in the company itself. There also would be a section where all the news specific to the placements and internships would be showing with their follow up links. With the help of all these features students can connect and share knowledge and grow together.

1.2 FLUTTER FRAMEWORK

The main purpose of building this app is to make education easy for all the students by connecting them together easily and letting them share their study materials easily. We, are going build this app using flutter. Flutter is the newest framework to make a splash in the world of mobile app development. Here, we delve more into what the Flutter framework is, its benefits and drawbacks, as well as the different ways to test a Flutter application. The Flutter framework consists of both a software development kit (SDK) and their widget-based UI library. This library consists of various reusable UI elements, such as sliders, buttons, and text inputs. Developers building mobile applications with the Flutter framework will do so using a programming language called Dart. With a syntax like JavaScript, Dart is a typed object programming language that focuses on front-end development. High performance and productivity in Flutter are achieved by using several techniques: Unlike many other popular mobile platforms, Flutter doesn't use JavaScript in any way. Dart is the programming language. It compiles to binary code, and that's why it runs with the native performance of Objective-C, Swift, Java or Kotlin. Flutter doesn't use native UI components. That may sound awkward at first. However, because components are implemented in Flutter itself, there is no communication layer between.

1.3 FIREBASE FRAMEWORK

The Firebase Framework which is used in this application is helpful for building such a database where every admin with access to the Google account where firebase it set up, to update the data and these changes are reflected instantly to every user. Firebase service handles most of the server-side work and there are numerous other elements that makes firebase a fitting choice as a free BaaS for applications. These services include a real-time database which is hosted on cloud where the data is stored as JSON files, it also enables the developers to authenticate the user using emails and passwords, also firebase provides storage for all the

data that is generated by the user, which in our application is the about section and all the information entered during the registration of events.

II. REVIEW OF LITERATURE

This literature review examines the factors that makes the use of flutter, favourable in cross-platform app development and the use of Firebase as BaaS for application development.

Almost everyone today has a smartphone in their pocket, so it is crucial to build mobile applications which are not only user friendly but also available across multiple platforms. According to a study, mobile phones have contributed to 48% traffic compared to PCs (47%). There are two major operating systems prevalent in the market namely android and iOS, two vastly different systems. There was a need to bind these two by developing a platform that would build apps compatible with both the systems. That's where flutter steps in. Flutter is basically an environment that provides you to build a reliable and efficient application across platforms. Cross-platform frameworks that show resemblance to React Native and flutter, are discussed and implemented by various companies numerous times formerly[1].

Flutter was released in in 2016 by Google. Apart from running on android and iOS , it works on Fuchsia as well. Flutter is exceptional because it is dependent on the device's OEM widgets rather than consuming web views[2]. A high efficiency rendering engine is used that build the apps as high performance as the native programs. Programming language proficiency is a hurdle for many beginners in the app development. Flutter is based on Dart, a language that was created to replace JavaScript. It is used extensively at Google and apps like Google AdWords have been created by using dart.

Approach of flutter is radical for the cross-platform development. Interface objects, render and an engine to build animation, graphics and other libraries is provided. Inspired by React-native, Flutter also follows uni-directional data flow philosophy where widgets (equivalents of React's components) only rely on data provided by the parent widget and optionally their own state[3]A flutter project is written in Dart programming language and AOT (Ahead-of-time) compiled to the native platform architecture, hence achieving uncompromised speed. At the top level, flutter provides widgets that are composed of many widgets to make the most common interface objects that we're used to on iOS and Android platforms. Because flutter follows an open and layered architecture, developers can make their own widgets compositing other widgets at any level of the layered architecture. In fact, that is how the Flutter team made the existing high-level widgets and there is no barrier from the framework for developers to do the same. This customization flexibility is unmatched by UI toolkits from either iOS or Android with hierarchical implementations and limited access levels.

Usually, when the app is developed using native instruments, the resulting application communicates with the OS and requests it to draw the OEM widgets (buttons, text fields etc. provided by the operating system). With the cross-platform technologies, the situation is different. Because JavaScript is not capable to contact OEM widgets directly, in the frameworks like React Native, developers write JS code which is then interpreted by the framework and the framework requests the OEM widgets from the OS[4]. This 20 creates a JavaScript bridge which is a significant overhead affecting the performance in an extremely negative way. Flutter is different. It does not use the OEM widgets at all. In other words, whenever a developer creates a button in Flutter, the framework does not ask the operating system to draw it.

Instead, it uses its own renderer, and draws the button itself, pixel-by-pixel. The high-performance renderer in Flutter utilizes Skia Graphics Engine which is also a product of Google. It is used for rendering in such famous products as Google Chrome, Chrome OS, Mozilla Firefox, Android, LibreOffice and others. In sum, the previously mentioned facts mean that Flutter actually acts more like a game engine rather than a traditional cross-platform development solution. Unity or Unreal Engine, for instance, can be compared to Flutter to some extent[5]. Flutter draws user interfaces instead of sprites and objects in games. The apps made with Flutter can look exactly the same way as native apps, but in fact they don't have that much in common with native, because instead of requesting the OS to provide an OEM widget, Flutter draws everything itself. This means that if a developer wants, he or she can make the user interface look exactly the same on all iOS and Android versions supported by Flutter. Or, as another example, Flutter can bring iOS 14 look and feel to iOS 10.

In this paper, we look to create a application based on previous works and determine how useful Firebase is, in the case of development of messaging applications. Firebase was established by Andrew Lee and James Tamplin back in 2011 yet was launched formally in April 2012[6]. Initially, the framework was designed to be used solely as a real-time database giving its APIs, enabling users to store and synchronize data and information across various users. However, Firebase was taken over by Google in 2014 and today, the service has various functionalities that offer development tools to various enthusiasts and entrepreneurs alike.

Firebase is a framework which is useful for building portable and web applications for businesses which require real-time database which implies when one user updates a record in the database, the update should be conveyed to every single user instantly. It gives a basic and unified platform to many applications along with a host of other Google features packed-in with the service. Firebase handles most of the server-side work when it comes to the development of applications[7]. There are numerous elements that make Firebase such an essential tool in development from a developer's point of view. In this way, it helps maintain a state of harmony between the developer and the client by causing minimal delay of work

Many developers are currently developing messaging applications with online solutions similar to Firebase, which provide real-time database integration facilities. Various open-source platforms such as Parse Server or Horizon offer similar services such as Firebase and offer developers to migrate from one vendor to another, but they also come with their own problems[8]. Developers are also trying to develop methods to optimize file transfer through such applications and to integrate more technologically advanced features into their applications.

III. RESEARCH METHODOLOGY AND DATA ANALYSIS

The approach for research is based upon analysing the popularity of flutter and react-native among developers and also comparing the two technologies based on their features and ease of use. By looking into the popularity matrix of various building platforms and determining what technology is widely used by aspiring developers will provide closure to choose what framework best for our application.

3.1 INTEREST OF DEVELOPERS

Figure 1 shows the outcome for the programming language they favour based on the programmers' interests on Stack Over Flow a forum site where software engineers exchange ideas, discuss, and ask questions.

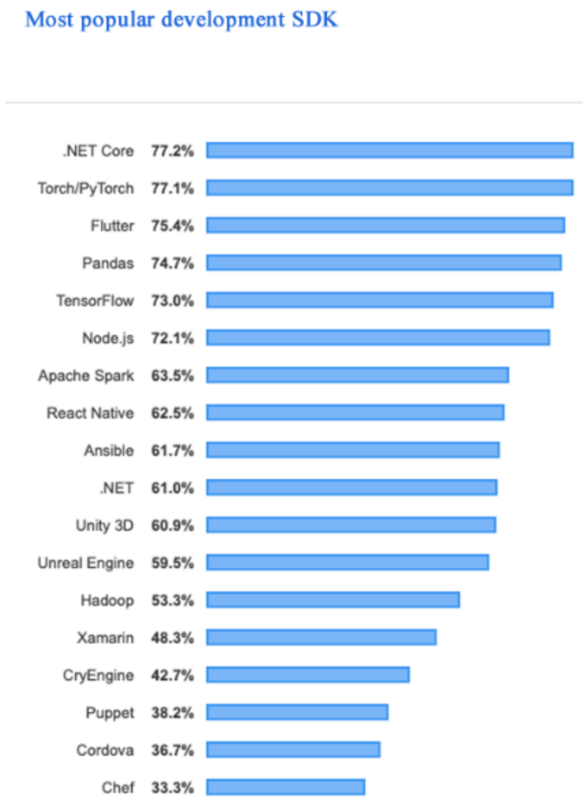


Figure 1: Ranking of various development technologies on Stack Over Flow

Flutter, React Native and Xamarin are the cross-platform frameworks that have entered the list in figure 1. While Flutter rated first among cross-platform structures with 75.4 percent, React Native came in second with 62.5 percent, making a significant difference above Xamarin, which came in third with 48.3 percent. Furthermore, these two platforms, among other frameworks, were in the top ten on this list.

Figure 2 depicts the graph showing the trend in search for React Native and Flutter on Google.com in the past 12 months.

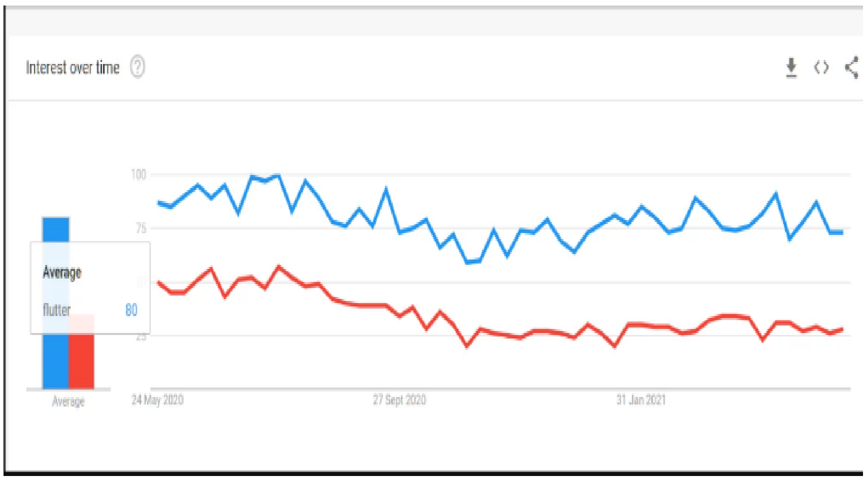


Figure 2: Flutter vs React-native search comparison

Figure 2 depicts the interest in Flutter and React Native over the past year. According to the data, global search interest has increased in the recent year compared to the highest point on the graph. A score of 100 indicates that the term is the most popular. A score of 50 indicates that a word is half as popular as another. A value of 0 indicates that there is insufficient information for this topic. While Flutter and React Native's search frequencies have been similar since September 2020, Flutter's search frequency has gained more in the last nine months.

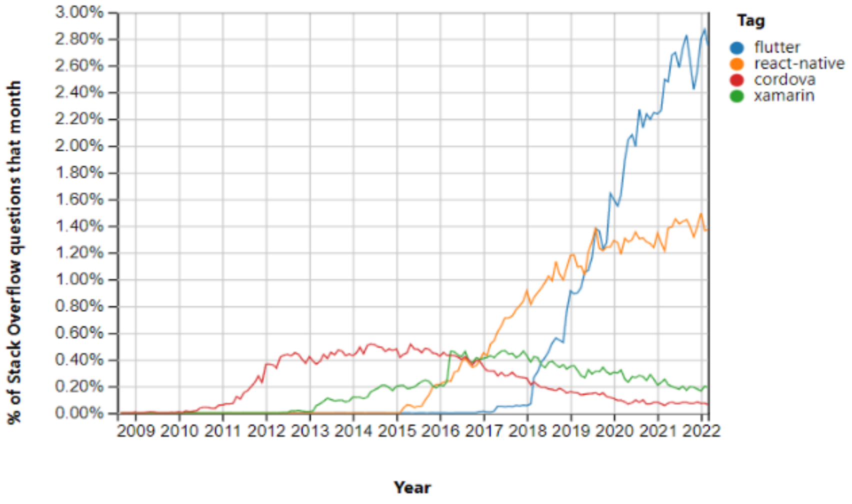


Figure 3: Tag benchmark trends on Stack Over Flow

Figure 3 shows the results of comparing the tags on the Stack Over Flow website for Cordova, React Native, Flutter, and Xamarin. Flutter and its closest competitor, React Native, differ by an average of 1500 questions. Despite the fact that Flutter was introduced two years after React Native, it swiftly closed the gap with a rise in user numbers and entered the race.

3.2 STRUCTURE AND CONCEPTS OF FLUTTER

Material is a mobile and web visual design language that is widely used. Material widgets are plentiful in Flutter. A uses-material-design: true entry in the flutter section of your pubspec.yaml file is a nice idea. This will allow you to take advantage of more Material features, such as the pre-defined Icons. Models are collections of data that are typically generated from servers, users, or external APIs, and are combined with widgets to complete the app's user experience. The folder by type pattern, by far the most popular among Flutter developers, advises organizing files according to their functionality/type. All screens, for example, would be placed in a folder called screens or something similar, data models would be placed in a folder called models or something similar, and so on. This pattern is simple to follow and suitable for beginners. This technique works fine for smaller projects, but once you have more than 10–15 files of each category, managing and maintaining them becomes exceedingly tough. Because the majority of a developer's effort is spent looking for the correct file, this design does not scale well for large projects

Using the weather app as an example, a model or a set of data may be the name of the place, as well as the temperature in both Celsius and Fahrenheit. If we examine a social media app that displays a user's profile page, it may display the username, age, profile picture, description, and so on.

The app extends Stateless Widget, making it a widget in and of itself. Nearly everything in Flutter is a widget, including alignment, padding, and layout. A default app bar and a body property that holds the widget tree for the home screen are provided by the Scaffold widget from the Material library. The widget subtree might be rather difficult to navigate.

3.4 FLUTTER FRAMEWORK ARCHITECTURE

Flutter is built as a layered, expandable architecture. It exists as a collection of separate libraries, each of which is dependent on the underlying layer. Every aspect of the framework level is designed to be optional and changeable, and no layer has escalated privileges to the layer below it. Flutter applications are packed in the same way as any other native application by the underlying operating system. An entry point is provided by a platform-specific embedder, which coordinates with the underlying operating system for access to services such as rendering surfaces, accessibility, and input, as well as managing the message event loop.

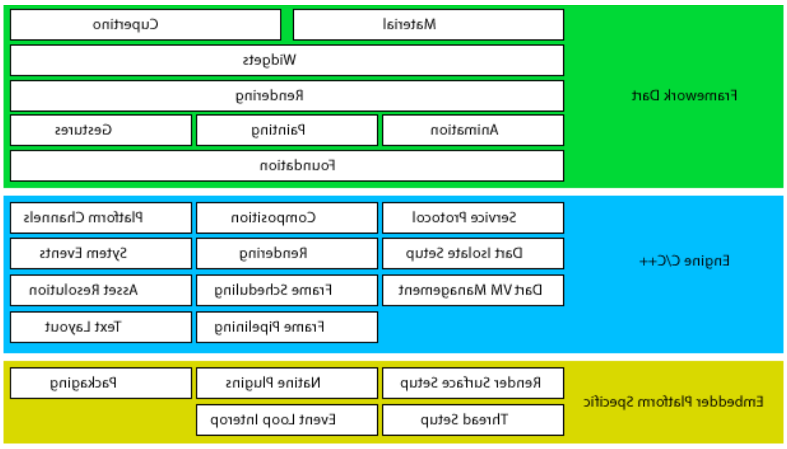


Figure 4: Flutter Architecture

The embedder is written in a platform-specific language, such as Java and C++ for Android, Objective-C/Objective-C++ for iOS and macOS, and C++ for Windows and Linux. Flutter code can be embedded into an existing application as a module or as the complete application's content using the embedder. Flutter comes with a number of embedders for popular target platforms, however there are many alternative embedders available.

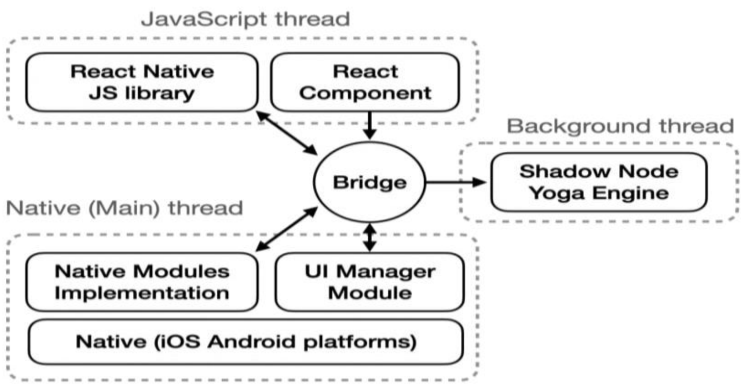
Dart:ui, which encapsulates the underlying C++ code in Dart classes, exposes the engine to the Flutter framework. The lowest-level primitives, such as classes for driving input, graphics, and text rendering subsystems, are exposed by this library.

3.5 STRUCTURE AND CONCEPTS IN REACT NATIVE

Structures known as components are utilized while designing React Native applications. These structures can be represented by components such as Text, View, Button, Image, and TextInput. These components can be changed just as they were when the web interface was created. React Native code block with View, text, picture, scrollview, and textinput components. The structure of component generation and placement is identical to that of HTML. In Html, the View component, for example, has a structure that is comparable to "div" tags. These components, like Html, can be changed. These customizable component libraries can be utilized in a variety of projects.

3.6 REACT NATIVE ARCHITECTURE

The JavaScript runtime environment is used by React. It's a web browser for the web. There is only one JavaScript thread, which makes advantage of natively integrated web APIs in the browser.



When using asynchronous functions, it's critical to understand how callbacks interact between the main JS thread and Web APIs. This interplay between the JavaScript engine and native APIs is also crucial for understanding how React Native interacts with its surroundings. In the React Native execution environment, there are three major threads: the JavaScript thread, the Native main thread, and the background thread that handles Shadow Node. In the current architecture, communication between these threads is handled by the "bridge" library.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

In the React Native execution environment, there are three major threads: the JavaScript thread, the Native main thread, and the background thread that handles Shadow Node. In the current architecture, communication between these threads is handled by the "bridge" library.

3.7 SUPPORT AND COMMUNITIES

Since 2018, Flutter has seen an increase in popularity on Google and Github over a two-year span. It is free to use and distribute because it is open source.

Google is the company that created Android.

1. Stack Over Flow, a help resource on GitHub, with tens of thousands of questions and answers. Between developers and beginners, questions are shared and various methods are discussed.

2. On Udemy, there are over 980,000 students and 140 Flutter courses.

3. Flutter has over 93,200 stars on Github, indicating its popularity among developers in less than two years.

4. Flutter provides support in its own libraries and in its official website.

3.8 PROS OF FLUTTER

From the perspective of an application developer the advantages of using flutter are as follows:

1. Flutter provides Fast Development, which saves a hell lot of your time for any individual or organization as flutter allows you to use the same code base for building apps for different platforms like iOS, Android, Web and Windows. So, it means you will only need a team of flutter developers to get started with the development of your product without being worried about the platform's development as a single code base will develop the app for all. It also saves you a lot of your time and resources.
2. Flutter's "hot reload" will help you to see the result of code changes immediately so you will not need to rebuild and compile your whole code to see the effect of a small code change.
3. Fast rendering and amazing customization that flutter offers as a result of its layered architecture. It gives you full control over each pixel of your screen so that you can convert your imagination into a real product without being worried about any limitations.
4. Separate UI from native controls cancel out all the possibilities of mistakes that any developer can think of because of smartphone manufacturers. Separate UI will help in achieving a unified view of all system versions. So, you can keep up your worry-free coding.
5. Flutter's Performance is also similar to native app performance so that you can code your app exactly the same as you will be coding it with native programming languages like Java or Kotlin and will not need to be worried about its performance compared to native app development.

3.9 CONS OF REACT-NATIVE

React Native has its own advantages and challenges but apart from this it also comes with many cons which are discussed below.

1. Development with React-native comes with one it's the worst disadvantage that is it is difficult to debug usually the process of debugging your code for any possible error is not that hard in native app development but in development, with React-native it is not that easy.
2. If you are developing an app with multiple screen transitions, animations and interactions then you should not consider React Native for your app development because it will not be the right choice for you. So, any app with complex gestures should not build using React Native.
3. Even though React Native is widely used but still the framework is immature in many aspects. Any new release or update brings so many significant changes that end up taking a toll on the developers. Even the update comes at a faster pace result of which developer did not get an adequate amount of time to adjust the changes.
4. React Native is based on JavaScript which is a flexible and powerful language for programming but it is also loosely typed. There is no type safety which makes it hard to scale any React Native apps which results in hard to integrate with third-party integrations tools like Flow and TypeScript.

5. React Native at its core is a cross-platform app development technology and to use it developers should have concrete knowledge of both native and web technologies. They should be aware of working with JavaScript, CI, UX guidelines, project configuration and many others.

IV. CONCLUSION

As a consequence of this research, it has been found that both systems are valuable in different ways, and that neither platform has a clear edge over the other. The number of people using these two open-source software is steadily expanding. In the areas of development environment preparation, online application development, editor diversity, the ability to use the platform without learning a new language for those who know JavaScript, diversity for ready-to-use components, and accessible resources, React Native's development environment is seen to be more advantageous than Flutter's. Flutter's three-stage test system is clearly more useful in terms of the program size taking up less space, being faster at the first launching of the app, and enhancing the developers' usage patterns. Despite their structural differences, these two platforms can be used to construct mobile applications for Android and IOS. Using the "Hot Reload" capability while developing applications on both sides saves time and adds functionality for the developer.

Both utilities are compatible with all current operating systems. On both platforms, the dimensions of the applications built using native technologies are larger. However, given the possibility to simultaneously develop IOS and Android applications on both platforms, the size of the applications generated can be overlooked. Regarding the difficulties or problems encountered while learning a language or constructing an application, given the prevalence of JavaScript in today's society. In this way, React Native outperforms Flutter in terms of responding to questions posted on software forums. The Dart programming language used in Flutter is fairly simple to learn (for someone who understands at least one object-oriented programming language such as C #, C ++, or Java). React Native makes use of the JavaScript programming language, which is well-known and widely utilized. Third-party software is heavily reliant on React Native. When using native modules (Bluetooth, Wi-Fi, SMS, and so on), developers may need to employ third-party libraries. On the Flutter side, widgets (device API access, navigation, etc.) are less reliant on third-party software[9].

Flutter has been supported by Google since 2018 and, like React Native, is still evolving. In terms of institutional support, they have no significant advantages or disadvantages over one another. As a result, no major superiority between these two cross-platform programming software can be claimed[10]. Programmers may prefer these two development environments, which are updated on a daily basis and have a growing number of users. It is feasible to create applications for several platforms using these two development environments while saving time.

V. GITHUB LINK OF APPLICATION CREATED

https://github.com/DiscipulusAcademy/discipulusapp

VI. REFERENCE

[1] A. Kumar, “Cross Platform Development using Flutter,” 2019. [Online]. Available: http://ijesc.org/

[2] A. Tashildar, N. Shah, R. Gala, T. Giri, and P. Chavhan, “APPLICATION DEVELOPMENT USING FLUTTER,” 2018. [Online]. Available: www.irjmets.com

[3] B. J. Crha and R. V. Rusnak, “Comparison of Technologies for Multiplatform Mobile Applications Development,” 2020.

[4] M. Olsson, “A Comparison of Performance and Looks Between Flutter and Native Applications When to prefer Flutter over native in mobile application development,” 2020. [Online]. Available: www.bth.se/dipt

[5] S. Stender and H. Akensson, “Cross-platform Framework Comparison,” May 2021.

[6] D. Sharma and H. Dand, “Firebase as BaaS for College Android Application,” 2019.

[7] C. Khawas and P. Shah, “Application of Firebase in Android App Development-A Study,” 2018. [Online]. Available: https://www.firebase.com/login/

[8] N. Chatterjee, S. Chakraborty, A. Decosta, and A. Nath, “Real-time Communication Application Based on Android Using Google Firebase,” International Journal of Advance Research in Computer Science and Management Studies, vol. 6, no. 4, 2018, [Online]. Available: www.ijarcsms.com

[9] Dagne, “Lukas Dagne Flutter for cross-platform App and SDK development Author Title Number of Pages Date,” 2019.

[10] S. Dmitrii, “STATE MANAGEMENT APPROACHES IN FLUTTER,” 2020.

[11] M. J. Rasool, “Machine Learning on Healthcare Big Data Using Apache Spark”, doi: 10.5281/zenodo.6374971.

[12] M. J. Rasool, “Spark based Health Status Prediction on Real Time Streaming Data using Machine Learning,” International Journal for Research in Applied Science and Engineering Technology, vol. 8, no. 11, pp. 471–478, Nov. 2020, doi: 10.22214/ijraset.2020.32197.

[13] M. J. Rasool, A. Singh Brar, and H. S. Kang, “Risk Prediction Of Breast Cancer From Real Time Streaming Health Data Using Machine Learning”, doi: 10.5281/zenodo.4284315.