

Comparative Analysis between Flutter and React Native

Sharjeel Moqrab Khan¹, Aftab ul Nabi Shahani², Tahir Hussain Bhanbhro³

¹Department of Computer Science Benazir Bhutto Shaheed University Lyari, Karachi, Sindh, Pakistan

²School of Information Communication Engineering, South China University of Technology (SCUT) China.

³Department of Computer Science, Shah Abdul Latif University, Khairpur Mirs, Sindh, Pakistan-66020.

E-mail: sharjeelmoqrabkhan@gmail.com, aftabshahani644@yahoo.com, tahir.bhanbhro@gmail.com

Corresponding Author: Sharjeel Moqrab Khan, sharjeelmoqrabkhan@gmail.com

Received: 03-07-2022; Accepted: 23-08-2022 Published: 29-08-2022

Abstract: Today's well know technology is mobile applications and mobile application development depends on demand skills after the invention of smartphones. The current mobile market shares are divided into two powerful technologies giant Android and iPhone. The Native Android mobile applications development Java-Kotlin are used, and on the other hand, for iPhone Apps are used Swift and Objective-C native IOS development. Those applications of Android and iPhone increased the development process and cost the development cost. Recently, cross-platform proved that developers quickly develop Android and iPhone applications with the same code known as a cross-platform mobile application, reducing company and customer effort and cost. In this research paper, the performance is compared between React Native Vs Flutter and efficiency, effectiveness, compatibility, community growth, documentation, Architecture, developer productivity, and Testing Automation Support CI/CD Support.

Index Terms: Flutter, React Native, Performance, Programming, Cross Platforms

1. Introduction

Therefore, cell phone usage has increased, while desktop usage has decreased. There are numerous methods to build an application, particularly for smartphones related to a native application, web application, and hybrid application. Moreover, a massive modification has been identified in developing native, hybrid, and web applications [1]. For native mobile applications development, developers use Java/Kotlin for native android development, and for iPhone, the language is used, Swift. However, Objective-c native IOS development, however, has been changed, which increased the development process and reduced the development cost. Now, developers have to option that they can develop Android and iPhone applications by using the same code known as a cross-platform mobile application, which reduces the effort and cost. In developing such hybrid applications developers have multiple tools and frameworks like JavaScript frameworks ionic, Vue, React Native, AngularJS, and so on. Google also wants to be in the business, not only in the business. However, dominate all markets for that purpose. Google introduced the newest cross-platform framework, which gets popularity Google's Flutter, which is also very effective and efficient for developing a cross-platform application [11]. Currently, React Native (JavaScript Framework) becomes the most useful framework for cross-platform mobile applications. However, Google has some advantages over React Native, which have been discussed in this paper.

The React Native story was fascinating and impressive: what impeccable beginning as Facebook's internal project Hackathon was the most admirable framework, introduced in 2013 summer. The initial review was published in 2015 January at React.js Con. In 2015 March, Facebook launched at F8, revealing that React Native has opened and offered on GitHub [15, 16]. Later, in the same year, it was recognized that React Native was not revealing a single sign of getting down. React Native was initially developed for the iOS application [20]. However, recently it also supports the Android operating system [2]. React Native has a strong community the mobile applications because it is open-source and supported by the JavaScript community. In March 2018, it featured that React Native has become the third most instant sensation on GitHub. The popularity was on a spike. Big tech companies such as Samsung and Microsoft contributed along with Facebook and individual developers they all were playing an essential role in promoting React native [3]. The GitHub source data was inspiring to see such numbers as 124 releases in 45 branches along with 1002

Comparative Analysis of Flutter between React Native

sponsored committed 7971. It became the 14th most featured source on GitHub. Moreover, it regularly updated the React Native has been following the two weeks' train publication. On the other hand, the Release Candidate Branch is offered every two weeks [2].

The interactive websites based on web applications give the same feeling likewise mobile apps. Web applications processed numerous browsers like Chrome and Safari, also mentioned in HTML5 and JavaScript. If a person's startup is based on the budget and has not required critical functionalities or allows operating system attributes, developing a web application could be the lowest inexpensive option [14]. For Android developer uses JAVA/Kotlin, and for IOS, they use objective C or Swift. A native app does not require a browser; it can be downloaded through the particular App store or Google store. Local apps could also connect with the operating system device features like the Contact list, Camera, Microphone, etc. However, local advancement needs a huge budget [13]. In 2015 at Dart Developers Summit, about four years back, the very first Flutter had been announced. In the beginning, it was decided to name "SKY" the Flutter (V-0.06) alpha released in May 2017. Later on, Google introduced the second review of Flutter lately in 2018 September. The makers around the globe began to feel the temptation. Although, before the announcement of Flutter 1.0, numerous Flutter applications were allowed in the shops. Big applications such as Google Ads, Alibaba, and Tencent Reflect proved that Flutter was astonishing [3].

Hybrid applications were developed for both webs and on a specific platform. Hybrid Applications use the same base code for a cross-platform, which means we just need to type once it can be run for Android and iPhone, making development fast. Developers have multiple tools and frameworks for a hybrid application like React Native and Flutter. In this paper, we chose Hybrid Application Development for the comparison of

React Native Vs Flutter. The rest of the paper is organized as: In sections 2 and 3 are based on the JSX and virtual dom. Section 4 presents details of Flutter, and section 5 is based on Redux and Principle. Section 6 is so many based on the comparison of Flutter vs. React Native. Section 7 presents details of users of Flutter in Production, and the section is based on open research issues. Finally, in section 8 we conclude our research work.

2. JavaScript XML (JSX)

React Native framework uses the JSX (JavaScript XML), which is the syntax of JavaScript that allows writing code just like HTML syntax, and it is used basically for how the User Interface shows.

1. It is faster than standard JavaScript as it performs optimization while translating to regular JavaScript.
2. It makes easier for us to create templates.
3. Instead of separating the markup and logic in separated files, react uses *components* for this purpose. We will learn about components in detail in further articles [4].

3. Virtual DOM

Knowing that React Native is a cross-platform framework, we can proceed with the applications of React Native at multiple platforms by utilizing DOM virtually [5]. The application to influence a lighter DOM tree, which is plotted along with an authentic DOM tree, boosts performance. The workflow of Virtual DOM is given in Figure 1.

Comparative Analysis of Flutter between React Native

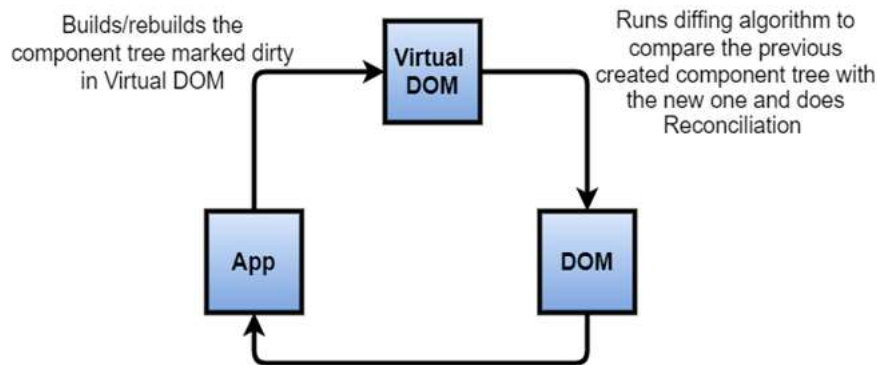


Figure 1. The workflow of Virtual DOM [4]

3.1 Props and State

In the React, the two data models named as Props and State. They have been utilizing for numerous reasons and have discrete construction. The Props model refers to external property, whereas the state is used in the components. The state is a global variable of a component, State stores components data and behavior, and after the changed store components, data, and behavior components will be re-render.

4. Flutter

Flutter is a cross-platform framework introduced by Google in 2017. The aim was to develop a high-performance mobile application. Flutter's application is not run on IOS and Android devices, and on Google Fuchsia, Google's next-generation operating system. Flutter would be the primary method to build an application for Google Fuchsia OS. Flutter is exclusive. Instead of using website views or depending on OEM device's widgets, the Flutter suppliers model utilizes its optimum performance varying engine [17]. This environment allows the opportunity to develop optimum performance applications according to the local application. As per architecture, the engine's code C/C++ has combined with NDK Android and iOS LLVM, respectively, or to some extent code of Dart is AOT-combined inside the local code while in progress. The Stateful Flutter supports hot-reload when developing that is taken as the primary attribute to spike the cycle of development [18, 19]. The Stateful hot-reload is importantly applied by inserting updated resource code into the ongoing Dart VM with no amendments. The app is an inside foundation, so all conversions and movements of the app would be conserved after the hot-reloading.

4.1. Dart Programming Language

The programming language of Google has introduced by Dart, and the reason behind the beginning was direct rivals to JavaScript. The initial adoption of Dart on Google is massive with AdSense, and AdWords. The Fiber team of Google to share their web app related to customer-facing.

4.2. Why Flutter Uses Dart

They were developing for user interface There are many object-oriented functions available in Dart, which are essential for thoughtful development compared to JavaScript holds. On the other hand, for starting an authentic programming language, JavaScript does not up, and Dart will fill the evidence gap. Dart is combined (AOT) "ahead of time into the local code for different platforms. The Flutter to interrelate with that platform devoid of a JavaScript pull, which does change in framework.

4.3. Widget

Widgets are considered crucial to an application's interface and views. They have to appear decent and natural, regardless of screen size. They have to perform rapidly, must be extensible moreover, and customizable. Flutter allows it by giving its widgets. Dart also provides the shaking tree compiler, which only contains code, required in your application. You freely access and utilize many widgets libraries; nevertheless, you require only one or might be two

Comparative Analysis of Flutter between React Native

from them. Flutter is a full package of widgets that works as a productivity promoter. The "Single thing is Widget" tactic of the Flutter created it simple to unite traditional UIs by a well-known set of creating blocks giving the framework. Due to Flutter working for both the devices Android and iOS, this is simpler to invest our time in building attractive models rather than UI porting [7]. Table 1 shows the two main kinds of widgets available if Flutter that is Stateful and Stateless Widgets. The below table 1 has described the main difference between the two widgets.

Table 1. Comparison between widgets

| | Dynamic Composition | Itself immutable | Sub State object mutable |
|------------------|---------------------|------------------|--------------------------|
| Stateless Widget | False | True | false |
| Stateful Widget | True | True | True |

4.4. MVC VS FLUX

The Model-View-Controller (MVC) is an app model design included in the correlated three parts. They contain the design model (data), the Controller (input handling process), and the View (interface of the user). The design or pattern of MVC is naturally utilized for creating a fresh interface of the user. It gives crucial pieces to design the programs for mobile and desktop, also for a web app. It runs excellent with object-oriented programming since the unique models, controllers, and views are cured as entities and reutilized inside an app.

4.4.1. Model

The model is the data utilized by a particular program. This might be the file, database, or maybe an ordinary object like a character or an icon presented in the video game.

4.4.2. View

When an application displays an object that considers a view, the examples are contained exhibiting text, buttons, or a window within a window. It contains everything which a user would be able to see.

4.4.3. Controller

The Controller has updated views and models. It receives input and does update corresponding. The characters can be modified and updated in a video game, just by changing the Controller's features. This might change the View by showing the most recent character in a game.

In Figure 2, it is seen that MVC parts are interconnected with each other. The related model, specifically for the user, has been displayed by the View. The organizer accordingly receives the input of the user and upgrades the views and the Model. On the other hand, there is no requirement for MVC in designing an application; multiple programming languages also IDEs hold the architecture of MVC, creating an ordinary selection for developers [8].

Flux is a design for controlling the flow of data by the React app. Although, it has been seen that a prior way of working with the components of React is by sharing data to its child components with the parent component. The design of Flux creates this prototype as the default way of managing the data. The three diverse characters available to handle the data by the Flux methodology:

- Dispatcher
- Stores
- Views (our components)

Flux's purpose is that there is each cause of belief (the stores), only can have upgraded through triggering activities. The movements are answerable for asking the sender, which the shops could promise for amendments and upgrade their data. When a sender has been activated and updated by the store, it will exclude altering results, from which the opinion could be again rendered. It might have identified unimportant difficulties that the Model creates it is amazingly laidback to reason regarding the form of the data. The reason for its variations, how it brings variance and

Comparative Analysis of Flutter between React Native

allows us to record the particular user flows, etc. the vital purpose behind the Flux is: inside the stores, the data moves singly by saving ultimately [8]

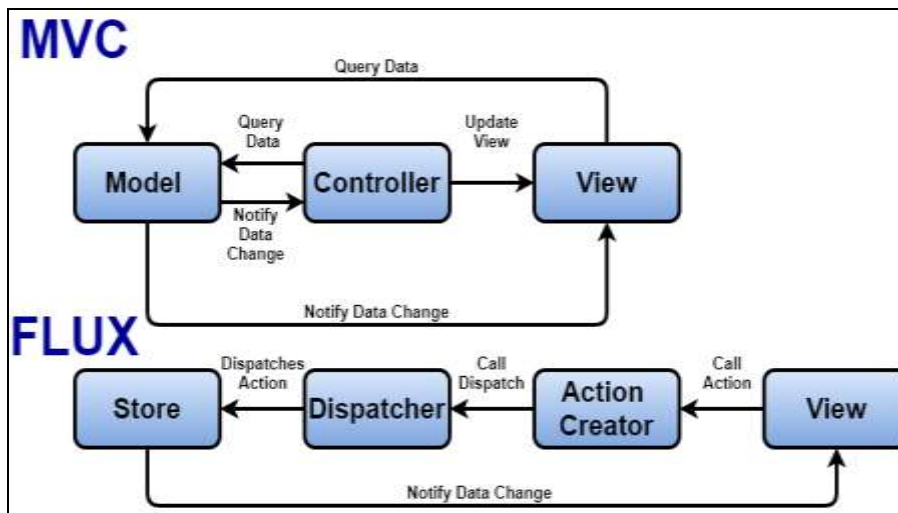


Figure 2. Interconnected Parts of MVC

5. Redux and its Principle

The main transformation between Flux and Redux is the idea of reducer and reliable role release the app's past state and the favorable action, moreover outcome the next state app. The reducers could be defined as the following equality.

$$f(state,action) = newstate$$

Influencing or modifying the remaining state, Reducers develop a fresh state related to the input act, which offers two main advantages:

1. All the components from corner to corner are universally developed by absolute state (solo resource of truth).
2. The reducer is useful and irrelevant context, which means for a similar output, there will always be similar input.

6. Comparison of Flutter and React Native

A comparison of Flutter and React Native from development support is given in Table 3.

Table 2: Comparison of Parameters Of Flutter And React Native

| Parameters | Flutter | React Native |
|-----------------------------|---|---|
| Programming Language | Flutter enables Dart's programming language was launched via Google in 2011 and is hardly accessed by creators. | JavaScript is accessed by React Native to develop across platform smartphone applications. In that case, the JS app is quite a famous language in the website community at this moment. |

Comparative Analysis of Flutter between React Native

| | | |
|-------------------------------|---|--|
| Installation | Flutter could be downloaded and installed both for particular meaning by GitHub. In this scenario of MacOS, we need to install the zip file of Flutter and make it a tracking variable. | React Native Model could be downloaded by accessing the package of Node NPM Manager. Makers that have back-end installation of a JavaScript of React Native are simple compared to other creators will require to study the Node manager package. |
| Documentation | In the beginning support, Flutter has extensive knowledge in the IDE procedure for Android and IOS. You may see the needed procedure facts on installing Flutter here for MacOS. On the other hand, Flutter application has a CLI feature named Flutter Doctor that can guide the maker step-by-step. | The initial instructions in the project of React Native imagine that the creators previously have entire needed stages for developing Android and IOS. Some knowledge is there regarding XCode's command-line tools, but it won't be sufficient to keep going. |
| Architecture | The Dart model utilizes a Skia C++ that entire protocols, channels, and also composition. On GitHub, the makers of Flutter have given detailed information there. Cutting it short, the Flutter has all the desirable features that can be demanded by the Engine of Flutter itself [21]. | The architecture of React Native comprehensively depends on the runtime environment of JS, and it is also famous as JavaScript pull. At the runtime, the code of React Native complied with the code of JavaScript. Through Facebook Flux architecture utilized by React Native. |
| Features and API | The flutter model is gathered with UI versions mechanism, AOI access device, testing, navigation, Stateful management also numerous libraries. The exclusive set of features excludes the requirement to utilize the library with a third party [10]. | The primary Model of React Native enables only APIs device access and UI rendering. To get access to more Native frameworks, it needs to depend on the libraries provided by a third party [10]. |
| Developer Productivity | According to the application's growing complexities, the creators will require to study and accept the fresh Flutter ideas. Moreover, it is an identified absence of support in Dart's features like text Editor and IDEs, and also it is not considered an ordinary programming language. | During the testing of modified UI, React Native has a remarkable tool that avoids the developers spending extra time. While in support of IDE, the creators are allowed to use it according to their selection or use IDE's text editor. |
| Community Support | The Flutter success's popularity started when Google promoted it in 2017 at I/O's Google conference. However, this app was launched way back before but did not gain popularity. Many things are happening in Flutter, like meetings and conferences happening in the Flutter community. | In 2015 React Native was introduced and got popularity ever before. Since then, the React Native Community of developers has been active and organizing conferences and meetups in different places. |
| Testing | It is recognized that when it comes to fantastic documentation, then Flutter is highly recommended. You may read the blog related Never code to explore more about Flutter and its testing. | Where we talk about the integrating testing of UI, not any authorized support is for React native. There are tools available from the third party for testing, such as Detox and Appium applications, but authentically do not support them. |
| Automation Support | Inline interfacing Flutter has the ability in it. We may make a binary for the application by utilizing the CLI feature and studying the rules and regulations through Flutter documentation for IOS and Android applications. | Authorized documentation of React Native has nothing programmed stages of installing the iOS application into the application store. Though, this gives a non-automatic procedure by XCode for installing the app. |

Comparative Analysis of Flutter between React Native

| | | |
|-----------------------------|---|--|
| <p>CI/CD Support</p> | <p>The external sources link has been added by ongoing testing and integration through one of Flutter's sections [22]. Though, Flutter's high command Interface line enables the users to set the CI/CD easily.</p> | <p>The official documentation of React native has not anything setting the CI/CD. Although, more or fewer articles for React Native are available that define CI/CD.</p> |
|-----------------------------|---|--|

Flutter is a fast-growing community for developers and the latest study shows that over 3 million developers used to flutter in past 3 years. This study was extracted from the Google Trends where figure 3 shows the comparison between the Flutter and React Native. So, the developers show the 97% of the interest for using Flutter and React Native 59% interest till September 2020. The collected results based on 12 months of record about Flutter and React Native. This graph shows week wise data.

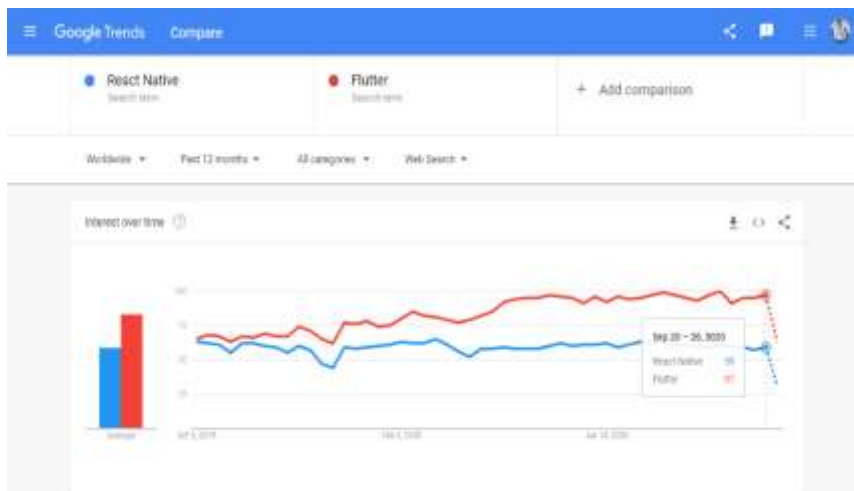


Figure 3. The people's interest level in Flutter and React Native [27]

After that, the interest level country-wise has been shown in Figure 4. In this graph, it has been noticed that flutter is growing too fast. So, the 73 countries' developers mostly doing work on flutter.

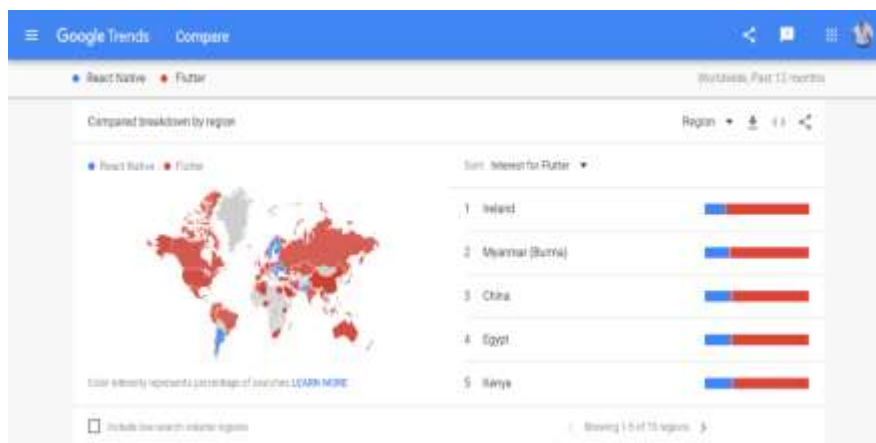


Figure 4. Country Level interest in Flutter and React Native [27]

7. Who Is Using Flutter in Production?

Flutter becomes the global trend for developing hybrid mobile applications. The software industry, including the major tech giants like Google and the biggest e-commerce company Ali Baba, adopted Flutter. The number is increasing rapidly as the flutter community grows so here is a list below, which has taken form flutter official site flutter showcase [9, 12]. The Flutter official showcase is given in Table 4.

Comparative Analysis of Flutter between React Native

Table 3: The Official Flutter Site Flutter Showcase

| | |
|---------------------------|---|
| <u>Reflect</u> | An impeccable and award-winning application manufactures with Flutter. That was included in the App Store of Apple as 'app of the day. |
| <u>Google Ads</u> | The feature of Google Ads allows users to control, optimize, and track their ad campaigns' progress in even a single mobile device. |
| <u>Insight Timer</u> | The application called Insight timer, which allows to access the most essential free library on the earth by guided meditation. The recorded that more than 9M people used this app. |
| <u>STA</u> | The Stadia app of Google was created to utilize the Flutter for either Android or iOS. |
| <u>Grab</u> | Flutter assisted Grab to develop the trading application for rapid growth in the business of food delivery. |
| <u>Alibaba Group</u> | Flutter assisted in carrying a fresh application for the most significant world's online platform to life. |
| <u>Baidu</u> | The Baidu Chinese company has hosted the Chinese biggest communication platform called Baidu Tieba. |
| <u>Abbey Road Studios</u> | The topmost application is Abbey Road Studio has continued the innovation old studio. |
| <u>Groupon</u> | To track the campaign's performance and analyze customer satisfaction. One app allows all the features to more than thousands of merchants under the name of Groupon officially by Flutter. |
| <u>Hamilton</u> | Precise popular Broadway application serves music on exclusive news, daily lotteries, a trivia merchandise game store, and many more. |
| <u>ByteDance</u> | Hello, Top buzz and Toutiao, Xigua Video, and more additional apps applied by users hundreds. Millions of times have been established with Flutter. |
| <u>Dream11</u> | India's biggest fascinating sports business, with over 50M users addicted, it's a fascinating sports application that is Flutter. |
| <u>NUBank</u> | Flutter also enables outdated bank problems to windup and manage the financial risk back to its user's hand through Nubank. |
| <u>Philips Hue</u> | Flutter carried innate reins to Philips Hue apps, letting customers establish the attitude and connect their keen lights to their broadcasting rapidly and effortlessly. |
| <u>EMAAR</u> | Worldwide stuff makers, Emaar create a fresh application considered for smartphones-initial communication with inimitable features of blockchain. |
| <u>GreenTea</u> | The inside View of the application based on customer management utilized by broadly Google revenue enables high imagination for stalking the sales targets |
| <u>Ebay</u> | The eBay-based application is so influential for surfing and selling automobiles straight by the consumer's mobile phones. |

8. Open Research Issue

Future mobile application development will be based on user feedback, such as quality of experience and user experience [23]. Applications also support measuring the quality of service (QoS) parameters of user devices and networks [24]. The application will track information on the user's usage pattern to easily recommend the contents of their choice [25, 26]. The quality must be considered in the development of the mobile application, which will meet the standards and requirements of organizations and users [27].

9. Application Experiment

9.1 Flutter

It was very straightforward to install Flutter. First of all, you need to download an SDK containing the latest version of Flutter. Then, if you are on Windows, you need to add Flutter to your PATH environment variable. Access flutter commands in the terminal to get access to them. Then in the terminal, you run "Flutter Doctor" to check if there are any missing dependencies. There are the dependencies:

- Newest Flutter version
- Android Studio
- Connected Devices
- Toolchain

After the command is run, if there are any, it will then give information on how to install these.

9.2 React Native

There were plenty of guides to install React Native, but the one that followed was the official react-native guide to get started. Which showed guidance for common operating systems such as Mac OS, Windows, etc. Linux, and. For this thesis application, the chosen operating system was Linux (Ubuntu 18) and the React-Native installation went very smoothly. The only complicated installation is to install the correct android studio packages and configure the emulator to run.

Even though Windows had Flutter installed. This will have no impact on the performance tests that we will perform because the app will run on the same mobile device and React Native has been installed on Linux. The installation continued with Android Studio. The customized installation to guarantee the Android SDK, Android SDK Platform, and Android Virtual Device was included. The initial phase of installing React-Native where to get the Java Development Kit (JDK). Android Pie (9) and three additional Android Pie (9) Android SDK Platform 28 packages will be installed in the second stage. Intel s86 Atom 64 System Image, or Intel x86 Atom System Image from Google APIs. The third stage was to add the following environment variables lines:

```
SHOMAbash_profile or SHOMEI.bashrc config file:  
export ANDROID_HOME=SHOME/Android/Sdk  
export PATH=$PATH:$ANDROID_HOME/emulator  
export PATH=$PATH:$ANDROID_HOME/tools  
export PATH=$PATH:$ANDROID_HOME/tools/bin  
export PATH=$PATH:$ANDROID_HOME/platform-tools
```

The last stage of the installation was to create a project through react-native init project name > tips and run react-native run-android rips that installed the emulators react-native application and explained that the emulator also starts with a running local server in the getting started guide. There was a bug in react-native 0.62 where, after a successful installation, the local server host closed. You had to write a react-native start for rips to solve this and then you could write react-native run-android rips when the service started. This ensured that you were running the local server and that you could develop the application.

9.3 Packages

Comparative Analysis of Flutter between React Native

Flutter

The packages that were used in the application are mentioned below:

- Geolocation
- device info
- HTTP
- Work manager

React Native

React-native packages are used to develop the application functionality.

- react-native push-notifications
- react-native background-timer
- react-native device-info
- react-native background-fetch
- react-native geolocation

Flutter Package Installation

To install a package, you must first add the package name under "dependencies" to the pubspec yaml file, then run "flutter pub get" in the terminal to update your project with the new package. If it is running the application in Android Studio. The developer will handle it. Just click on "update dependencies" at the top of the screen to detect if a new one is available. The package will be automatically added.

React Native Package Installation

The installed react-native version was 0.62, which resulted in a simpler package installation process because package links are automatically supported by this version. "The only thing you need to write when installing a package is" npm install "package-name" —save.

Flutter

Three screens were used to make the application. In order to see if the user is already stored in the database, or if the user needs to be added, SmartScreen will simply do some background checks. The Home Screen will be the main "hub" of the application where the user can create new activities or monitor activities that have already been saved. A selected activity can also be removed by the user. The Stopwatch Screen is where the user can start the activity by using the buttons shown on the figure 6 screen. The user can choose either to save the activity or cancel the activity. Figure 5 shows the starting screen, and the tab screen.

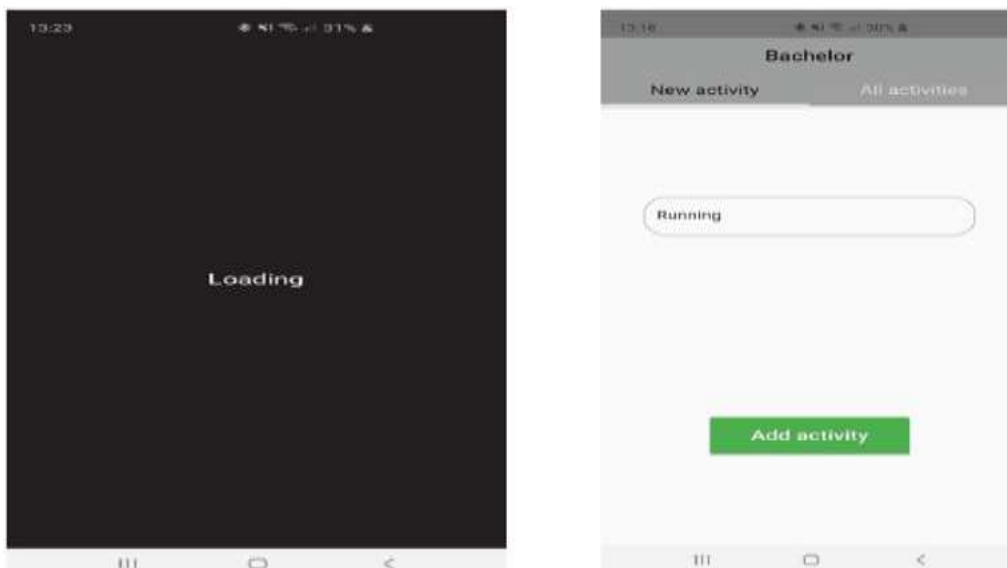


Figure 5. Left side shows the Start Screen where the right side shows the Tab 1 Screen

Comparative Analysis of Flutter between React Native

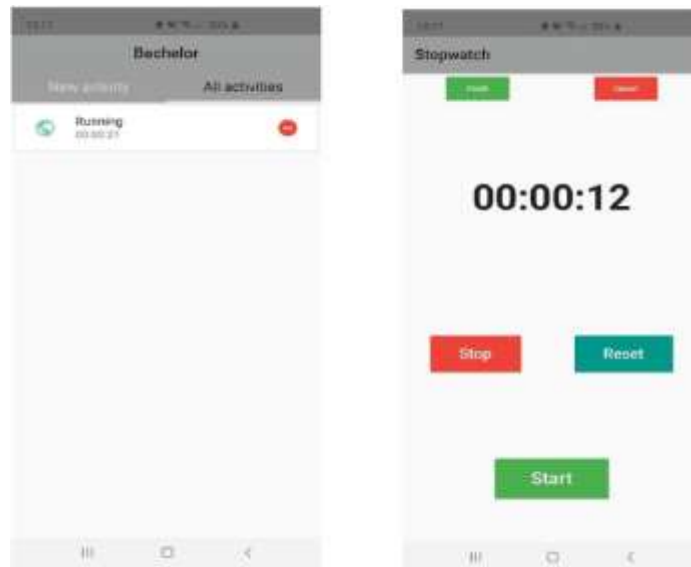


Figure 6. Left side shows the Tab 2 Screen where the right side shows the home screen

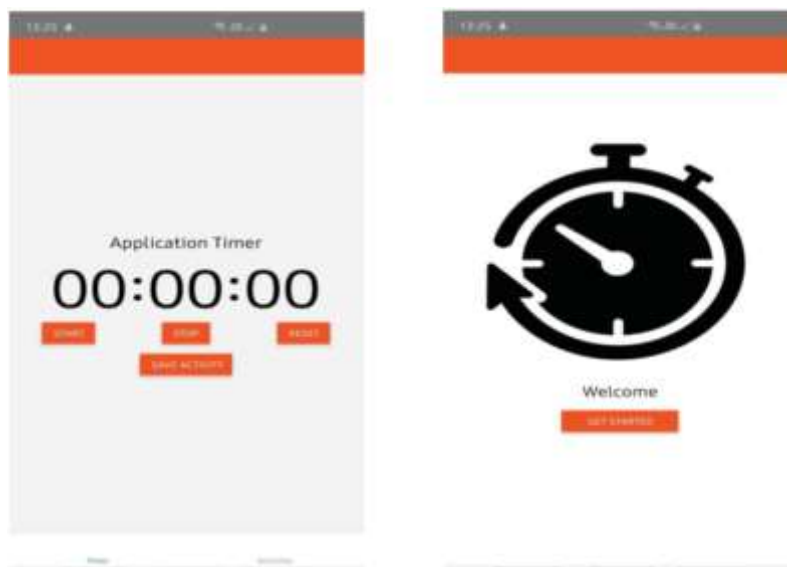


Figure 7. Left side shows the Home "Tab View Timer" and the right side shows the Authentication

React Native

Two screens make up the react-native application. Figure 9 shows the authentication feature. An authentication screen is only created to ensure that the user is registered in the system and the home screen is the main functionality panel that includes a tab view with two view options such as new timer and new activity. Figure 7 the Tab View Timer. In the vision of the timer "Stop" you can begin "With reset" Save the activity, and Figure 8 shows the activity view displays the activities that were saved.

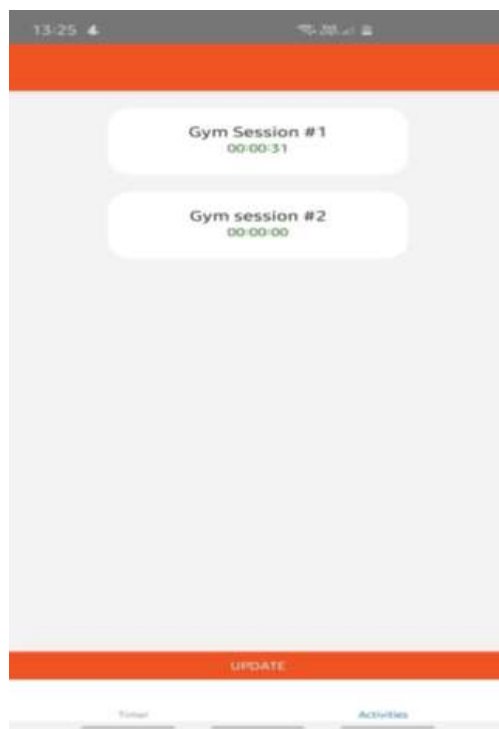


Figure 8. Home "Tab View Activities"

Functionality

We initially wanted to include GPS functionality in the application, but we discovered during the development that it is necessary to choose a payment method to toggle Maps API on Flutter. That is why in our experiment we did not include this.

Flutter

Start Screens:

first. It will go through the 'Start Screen' when the application is run. This screen will only show the user a word saying-loading 'while it will check in the background if the current user is already in the database or if it needs to be saved as a new use. Then, to see if the user current has any activities that have been started, it will. The application will immediately direct the user to the stopwatch page if an activity has started.

Home Screen:

The Home Screen page is where the user can either start a new activity by entering the name of the activity that will be stopped, or by navigating through a tab at the top of the same, the user can show all finished activities. On the tab for the activity. The user can see all the latest activities and delete a selected activity as well. A name for this activity has to be given if the user wants to create a new activity. Once the user has finished, the application navigates to the Screen for Stopwatch.

Stopwatch Screen:

There will be 5 buttons to control the stopwatch on the Stopwatch Screen. Each button has its own functionality to control the stopwatch. The included buttons are the Start button, the Stop button, and the reset button. Button Cancel and Button Finish. The first time the start button is pressed, it will create an activity and post it to the database. Then a background task process will start to notify the user every 15 minutes. And the stopwatch will start ticking at last. What happens if the user presses the stop button is that the current activity updates the database and indicates that it has been stopped. The stopwatch will then stop. The reset button updates the current activity to indicate that rest will remove the current activity from the database, then reset the stopwatch. Resetting the

Comparative Analysis of Flutter between React Native

stopwatch removes the background task and finally navigates the user to the home screen and finally stops the stopwatch button, saves the activity from the stopwatch time database, removes the background task, and then navigates to the user on the home screen.

React Native

In the get started, save the activity, and update activity button, the react native fetch API feature is implemented. The get started button makes a regular request to the API to check if the user is registered on the system and if the user has not added a POST request and the updated button updates the activity list by a GET request. The user is stored in the database via the unique ID of the user device, and the application informs the user about specific information in the local push notification. Every 15 minutes, when the app is killed as a periodic running background and notification when the timer activity is started, a notification will inform the user not to forget to track the activity.

10. Conclusion

In this paper, we discussed the current hot comparison between the Flutter app models does commendable work by enabling an extraordinary model to create mobile phone applications truly through the independent method. Flutter allows quickly building procedures, outstanding performance in giving smartphones app outcomes, exclusive and powerful user interface for both platforms such as iOS and Android. The Flutter model allows many fresh developers to make outstanding performance and mobile apps with full features soon. Also, Flutter may be using Google Fuchsia, Google's operating system for desktops and phones. So there is much future for Flutter Developers.

References

- [1] Heitkötter, Henning, Sebastian Hanschke, and Tim A. Majchrzak. "Evaluating cross-platform development approaches for mobile applications." In *Web Information Systems and Technologies: 8th International Conference, WEBIST 2012, Porto, Portugal, April 18-21, 2012, Revised Selected Papers 8*, pp. 120-138. Springer Berlin Heidelberg, 2013.
- [2] Kaushik, Vipul, Kamali Gupta, and Deepali Gupta. "React native application development." *International Journal of Advanced Studies of Scientific Research* 4, no. 1 (2019).
- [3] Novac, Ovidiu Constantin, Cornelia Mihaela Novac, Bogdan Ciora, Cornelia Emilia Gordan, Mircea Ioan Gordan, and Gyöngyi Bujdosó. "The rise of mobile development: a comparison between Ionic and Flutter." In *2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1-10. IEEE, 2022.
- [4] Wu, Wenhao. "React Native vs Flutter, Cross-platforms mobile application frameworks." (2018).
- [5] Stender, Simon, and Hampus Åkesson. "Cross-platform Framework Comparison: Flutter & React Native." (2020).
- [6] Japikse, Philip, Kevin Grossnicklaus, Ben Dewey, Philip Japikse, Kevin Grossnicklaus, and Ben Dewey. "Build the Web Application with ASP. NET Core, Part 1." *Building Web Applications with .NET Core 2.1 and JavaScript: Leveraging Modern JavaScript Frameworks* (2020): 271-315.
- [7] Campit, Scott, and Sriram Chandrasekaran. "Inferring metabolic flux from time-course metabolomics." *Metabolic Flux Analysis in Eukaryotic Cells: Methods and Protocols* (2020): 299-313.
- [8] Pop, Dragos-Paul, and Adam Altar. "Designing an MVC model for rapid web application development." *Procedia Engineering* 69 (2014): 1172-1179.
- [9] Wu, Wenhao. "React Native vs Flutter, Cross-platforms mobile application frameworks." (2018).
- [10] Sharma, Yogesh, and Sachin Gupta. "A study of flutter and react native for mobile app development." *Our Heritage Journal* 68, no. 27 (2020): 692-8.
- [11] Kuzmin, Nikita, Konstantin Ignatiev, and Denis Grafov. "Experience of developing a mobile application using flutter." In *Information Science and Applications: ICISA 2019*, pp. 571-575. Springer Singapore, 2020.
- [12] Khan, Sharjeel Moqrab, Aftab ul Nabi, and Tahir Hussain Bhanbhro. "Comparative Analysis between Flutter and React Native." *International Journal of Artificial Intelligence & Mathematical Sciences* 1, no. 1 (2022): 15-28.

Comparative Analysis of Flutter between React Native

- [13] Ju, Ran, Xingchen Zhou, Bo Xu, Weiqing Liang, Wanyi Yang, Yuan Cao, Eryan Zhang et al. "DUES-Adapt: Exploring Distributed User Experience With Neural UI Adaptation." In Proceedings of the 25th International Conference on Intelligent User Interfaces Companion, pp. 91-92. 2020.
- [14] Höchst, Jonas, Lars Baumgärtner, Franz Kuntke, Alvar Penning, Artur Sterz, and Bernd Freisleben. "Lora-based device-to-device smartphone communication for crisis scenarios." In Proceedings of the 17th International Conference on Information Systems for Crisis Response and Management (ISCRAM). 2020.
- [15] Armagan, Ferdi. "Vergleich von nativer App-und Cross-Platform-Entwicklung (Facebook React Native und Google Flutter)." PhD diss., Hochschule Hannover, 2020.
- [16] Hamzah, Nur Asyiqin Bt Amir, Muhammad Radzi B. Abu Saad, Wan Zakiah Bt Wan Ismail, Thangavel Bhunaeswari, and Noor Ziela Bt Abd Rahman. "No. 7 Development of A Prototype of An IoT Based Smart Home with Security System Flutter Mobile." Journal of Engineering Technology and Applied Physics 1, no. 2 (2019): 34-41.
- [17] Payne, Rap. Beginning app development with flutter: Create cross-platform mobile apps. Apress, 2019.
- [18] Zammetti, Frank, and Frank Zammetti. "Flutter: A Gentle Introduction." Practical Flutter: Improve your Mobile Development with Google's Latest Open-Source SDK (2019): 1-36.
- [19] Fayzullaev, Jakhongir. "Native-like cross-platform mobile development: Multi-os engine & kotlin native vs flutter." (2018).
- [20] Profita, Halley, Nicholas Farrow, and Nikolaus Correll. "Flutter: An exploration of an assistive garment using distributed sensing, computation and actuation." In Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction, pp. 359-362. 2015.
- [21] Kamaludin, Aa Bayu, and Dian Dharmayanti. "APPLICATION MONITORING THE COURIER ON PESHOP UTILIZING WEBSOCKET AND FLUTTER."
- [22] Laghari, Asif Ali, Hui He, Asiya Khan, Neetesh Kumar, and Rupak Kharel. "Quality of experience framework for cloud computing (QoC)." IEEE Access 6 (2018): 64876-64890.
- [23] Laghari, Asif Ali, Hui He, Muhammad Shafiq, and Asiya Khan. "Impact of storage of mobile on quality of experience (QoE) at user level accessing cloud." In 2017 IEEE 9th international conference on communication software and networks (ICCSN), pp. 1402-1409. IEEE, 2017.
- [24] Balapour, Ali, Hamid Reza Nikkha, and Rajiv Sabherwal. "Mobile application security: Role of perceived privacy as the predictor of security perceptions." International Journal of Information Management 52 (2020): 102063.
- [25] Laghari, Asif Ali, Hui He, Muhammad Shafiq, and Asiya Khan. "Application of quality of experience in networked services: Review, trend & perspectives." Systemic Practice and Action Research 32 (2019): 501-519.
- [26] Shaikh, Zaffar Ahmed, Asif Ali Wagan, Asif Ali Laghari, Karar Ali, Mashooq Ahmed Memon, and Anwar Ali Sathio. "The role of software configuration management and capability maturity model in system quality." IJCSNS 19, no. 11 (2019): 114.
- [27] Allain, Hugo. "Improving productivity and reducing costs of mobile app development with Flutter and Backend-as-a-Service." (2020).