Tampere University

Muhammad Ali Gulzar

USING MOBILE APPLICATION FRAME-WORK FOR SOLVING THE PROBLEM OF UNAFFORDABLE TUTION FEES

Faculty of Engineering and
Natural Sciences
05-2019

ABSTRACT

Muhammad Ali Gulzar : Technology For Innovation
Tampere University
Bachelor in Science and Engineering
05-2019

The problem that is discussed in this thesis is the students who could not afford education fees and are bound to leave the educational institution.

This problem could be solved using different technologies that are easily accessible on the Internet. However, after valid arguments and research, the use of making a mobile application that could minimize the number of children being forced out of education institution could be minimized. In this modern era, most people have a smart phone which makes the mobile application easily accessible to the users.

It is easy to prove why mobile applications would be a suitable and optimum choice in the short term. However, there are different technologies that could be used in making a mobile application. These technologies include React Native, Flutter, Ionic and many more. These technologies are called framework of mobile application development. Each framework has its own advantages and disadvantages. However, React Native [8], developed by Facebook, being stable, popular and widely used by the community, is the favorite choice of making a mobile application to solve the current problem in a short period.

React Native is an open source framework which means anyone can contribute to the architecture and functionalities of the framework. As React Native is quite popular among the developing community, there are large number of contributors as well as developers making third party packages that can be used in React Native. Similarly, third party packages are widely used in the development of this application such as Firebase which is developed by Google. Finally, the application is deployed in the store for users to download and hopefully have the impact on the problem that is being targeted.

PREFACE

I would like to thank my supervisor, Marko Helenius, for his excellent guidance and support during this process of writing the thesis. I would also like to appreciate Sarina Lewis for guidance in the structure and English grammar.

Tampere, 20 December 2018

Muhammad Ali Gulzar

CONTENTS

1. INTRODUCTION

This thesis covers the aspect of using mobile architectures to make a simple mobile application. One of the main problems that is targeted to solve and discuss in thesis is about the expensive education and how due to this unaffordable education, children from low income families cannot afford it. Most of the university students drop out of their institution mainly because they cannot pay the extensive fees. They need a source of income so that they can supplement their education and meet their daily life needs. The other simultaneous event happening in third world countries is that parents hire professional teachers to teach their very small children even after school timing. Professional teachers teach only basic subject concepts with a great amount of finance. Keeping this in mind, parents could instead hire university students that are not able to afford their education who can also teach the smaller children the same basic concepts that a professional teacher is going to teach. There are benefits from both sides of the party. University students would be able to finance their education and the burden on them would be lightened and parents would be able to hire less expensive teachers for the smaller children that would give them the same amount of benefits in terms of teaching their children.

The aim of this thesis was to utilize Facebook's React Native and create a mobile application that would try to solve this problem. University students who are not able to finance their studies are able to make a profile for free on the mobile application giving additional information such as contact number, what are they currently studying and much more. Then there is a list of students who want to teach smaller children who can be accessed by parents and they could decide which student to hire based on the information provided in the profile. React Native makes mobile application available in both iOS and Android and give near native performance. Moreover, comparison of different mobile application frameworks would be analyzed to see similarities and differences and what other frameworks lack.

The thesis opens with Chapter 2 which introduces research methods implemented to collect data and information about the react native framework and technologies. Chapter 3 will give insights about different mobile operating systems and mobile applications. Chapter 4 will discuss the comparison between React Native and other frameworks that could be used to implement the mobile application and why React Native was chosen. Brief examples of different popular mobile application examples will be given that are using React Native. Then, in Chapter 5, React Native framework is studied in detail and how the application would be constructed. Finally, in Chapter 6, conclusions about different frameworks and preference of mobile developers is analyzed.

2. RESEARCH METHODS AND TECHNIQUES

This chapter covers the methods that were used to do the research and writing this thesis. Scientific information was collected using Andor [38], a search engine provided by Tampere University of Technology for published documents and articles. Information was extracted from official documents provided by the developers of the technology stack.

There were many obstacles and pitfalls when searching for good quality and authentic information. The first problem that occurred frequently was with the word *React* because in the dictionary it is a verb which means reacting to something. Similarly, Dart would refer to the game of throwing arrows in a cardboard. Thus, the terms provided some issues as the results did not refer to the actual concepts and definitions that was aimed to be found. Specifying multiple search parameters also provided irrelevant results such as *"React" AND "Mobile"* showed results of mobile that is made by a company and is available on the market for purchase and other research how the consumers react to mobile marketing.

Due to the technologies discussed in this thesis are highly popular and thus the information provided is of good quality on its own. Hence, there is very little published scientific research on the technologies discussed in this thesis which made it difficult to find scientifically proven information.

Despite all the obstacles, the information was retrieved from different sources and used in many different parts of thesis. The research was based on design science which is not only the designed framework but also raise the possibility of comparing the results obtained from different brands of design-orientated research [43].

There is a similar scientific research done. It has done similar kind of comparisons between different mobile application frameworks and determine which framework is suitable for developers. The research is done in University de Cádiz [44]. The author describes the situation that with time, the mobile usage has increased which led to an increase in the mobile application frameworks. It becomes difficult in the life of a mobile developer to choose a framework to accomplish a task. Hence comparison of different mobile application frameworks is discussed which leads to the conclusion of which framework most developers should prefer to complete a specific task.

3. MOBILE APPLICATIONS

3.1 Introduction to Mobile Applications

Mobile applications are services that are designed to operate on a mobile device. People in the early times preferred using a computer and accessing the web applications but people have started using mobile applications in the recent years due to the easy access of smart phones and the increase usage of the Internet. As of December 2017, there are around 4,000,000,000 internet users around the globe according to a survey conducted by VPN Mentor [1]. The percentage share of mobile internet traffic compared to global online traffic in 2018 was 51.2 % [11]. As we can see from the statistics of the increased usage of mobile and internet, there is more time and technology invested for the advancement of mobile applications. As of today, there are three types of mobile applications, that we will be discussing in the next section.

3.2 Types of Mobile Applications

Before understanding the types of mobile applications, we need to understand what is a mobile operating system and how many of them exists in the present day.

There are three major mobile operating systems: Android, iOS and Windows Mobile. All of them are built by different techniques, methods and programming languages. Furthermore, they have a different user interface and provide different features which makes them unique among each other. For example, iOS uses Objective-C [2] as its primary programming language whereas Android uses Java [3]. As of the market share between the mobile operating system, in the second quarter of 2018, 88 percent of mobile devices have Android as an operating system where as 11.9 percent have iOS. The remaining 0.01 percent of mobile devices run on different mobile operating system according to a survey conducted by Statista [12]. Due to the differences between mobile operating system, there are three types of mobile applications; **Native**, **Hybrid** and **Web Applications**. We will be discussing each of them in detail in the following paragraphs.

Native applications mean that the applications will be made specifically for the operating system it is intended to perform on, using operating system specific programming language and other specifications [4]. If a simple mobile application is made using Java programming language, the application could only function in mobile devices which have Android operating system. There are many advantages of building a Native mobile application. The major advantage is that Native applications are quite fast as they are using the built-in functions of the mobile operating system. Native applications also provide an elegant and user-friendly feeling when using them. There are different sizes of mobile devices in each operating system, hence, when making native applications, the aspect ratio is easier to handle so that the view of the application is the same across all mobile devices of different sizes. Despite having a lot of advantages of making a Native application, there are numerous disadvantages. As there are different types of mobile operating system, a developer should prepare mobile applications for each mobile operating system which concludes that the developer must have knowledge of making applications in each platform which makes it time and energy consuming. The cost of making applications are proportional to the time and energy consumed in making them, thus, this makes Native applications quite expensive.

The concept behind hybrid applications is that they are simple applications made using web technologies such as HTML5, CSS and JavaScript and wrapped inside a mobile device so that they can look like a native application. These types of application are neither pure native nor pure web applications based but lies in the middle. So, the first advantage that could be highlighted is that there is only one code base which will be used to make a mobile application and it would perform in all operating systems. This leads to less energy and time consumption and as discussed before that cost of developing an application is proportional to energy and time, the development cost would also be relatively cheaper to development cost of native applications. As there is only one code base, it is easier to update an application for a developer. A single change in the code would be reflected in all the mobile operating system it is published in. Although, there seems to many advantages of making a hybrid application rather than native, the disadvantages of Hybrid applications have its own weightage. As it's architecture is not as the same as Native applications, the user interface and usage capabilities are not the same; hence they are quite slow. A Hybrid application would require an internet connection to work which is a huge drawback because the point of having a mobile device is that a user can access information whenever they want and wherever there are. Overall, a Hybrid application leads to a poor user experience. As of today, there are many frameworks that support the development of Hybrid applications. Some of the most famous are Ionic and Cordova [6]. There are famous applications that are Hybrid and are used all around the world including MarketWatch and ChefSteps [7].

The last one that will be discussed is Web applications. As the name suggests, there are simple website that run on the browsers. Some web applications provide a functionality of installing them, which means a user can create a bookmark of the web page and save it like an application icon on a mobile device. This is the simplest form of web application and have the same features of hybrid except it is not installed from a platform specific store where all the real mobile applications are published.

As the previous arguments demonstrate, there are different advantages and disadvantages in making mobile applications using different techniques. The developer must choose which technique he/she should choose to make the desired mobile application. As problems arise, so does the solution and hence on March 2015, Facebook announced a framework called React Native [8] that would revolutionize the development of mobile application. Let's discuss what is React Native and how it has solved problems in the modern era of mobile development.

3.3 React Native Framework

React Native is a framework in JavaScript [9] programming language. It gives the power to developers to prepare a mobile application that is real and natively rendering for both iOS and Android. It allows the developer to use one programming language; JavaScript to develop cross platform applications in iOS and Android which are the two most used operating systems in the world right now. Before diving into the architecture of React Native, let's study a brief history of how React Native emerged and established itself as one of the leading frameworks in the mobile development.

It started in the summer of 2013 at Facebook's internal hackathon project [13]. After two years, it was announced on January 2015, that React Native was available for public review and was available as Open Source on GitHub on March 2015 [13]. Since its announcement, it has become one of the most popular framework for mobile development. It has not shown any signs of any decline and is currently the 4th most starred repository on GitHub [14]. Repository is a placeholder

for the code and as React Native is concerned, its repository is public and anyone can see and recommend any changes to it. It has a large developer community and it's keep on growing. The community releases new updates almost every two weeks that would further enhance the performance and user experience of making and using mobile applications in React Native. It is no surprise that people search the term "React Native" on Google more than terms "android development" and "iOS development" [15] combined. Let us discuss now what React Native offers that it has convinced a large community of developers to switch from native development to use a framework called React Native.

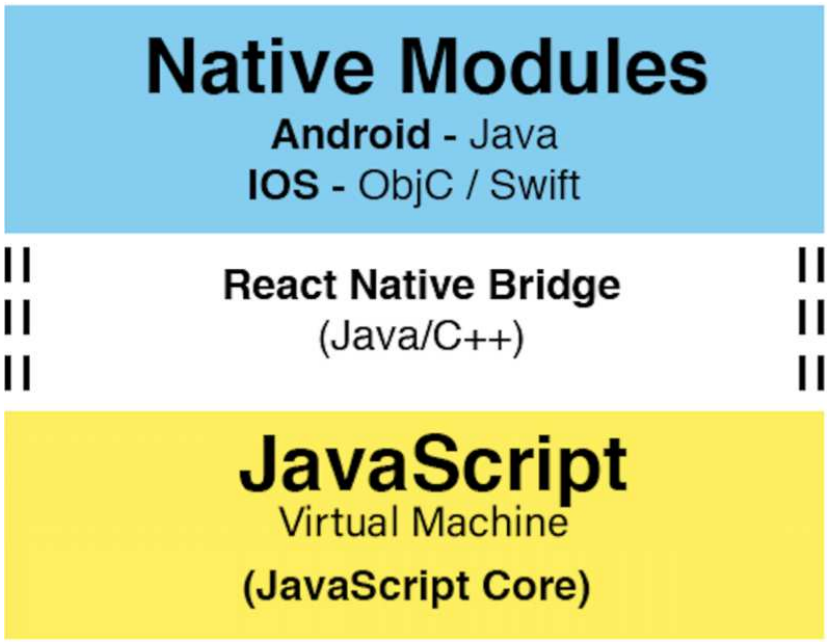

Figure 1. React Native architecture diagram [44]

React Native combines the advantages of developing in native style while giving the benefit of developer to produce in cross-platform using one single code base without sacrificing flexibility in platform specific requirements. React Native renders real user interface element which are always preferred by the users to have a good user experience. It makes applications look and behave like native iOS and Android applications. This is contrary to Hybrid and Web Applications which are not able to provide a Native look and feel which leads to bad user experience. Besides for having the advantage of having a Native interface and experience, it provides the developers to have one code base to develop for cross platforms (iOS and Android). The developer does not have to learn or even get familiar with more than one programming language. It could only learn and use JavaScript which is the fundamental programming language to make mobile application in React Native. The developer has to learn only once and it can implement a function that would be performed both in iOS and Android. Hence, it is no longer required for a developer to learn different programming languages and development concepts for different mobile operating systems. Again, for the development side, React Native provides the ability for the developer to make changes in its code and it would be immediately reflected on a simulator that would be running on your computer which is quite fast. This saves development time and allows the developer to focus more on the producing a high quality mobile application. With all these advantages combined, it leads to one more advantage which is saved time. Furthermore, the cost is also directly portioned to the time it takes to make a mobile application. In a short summary, using React Native, it allows a developer to make a working mobile application in very short period that would work in cross platforms, providing native performance and features which would provide a good

user experience and decrease development costs compared to developing application either in Native, Hybrid or Web.

According to Rodriguez-Sanchez Guerra; Manuel [44], React Native Bridge is responsible of communicating the JavaScript virtual machine with the native code of the platform. When rendering the components, React Native, instead of rendering them over the browser DOM, calls the corresponding API's of the OS to render the components. Figure 1 shows the React Native's architecture.

4. COMPARISON OF DIFFERENT FRAMEWORKS

Previously we have described only one framework called React Native. In this section of the thesis, different mobile frameworks will be discussed and compare it with React Native and at the end draw upon a conclusion on which frameworks most mobile developers would prefer. Let's introduce two frameworks we are going to examine. The first framework to inspect is called Ionic [16]. Let's go into the details later but it is a framework for making Hybrid Applications. The second framework is the direct competitor of React Native called Flutter [17]. As it is the direct competitor, it adopts the similar concepts of making applications as in React Native. Let's examine Ionic and Flutter in detail and conclude which will help us determine and convince that React Native is the framework to go to.

4.1 Ionic

Ionic is a hybrid mobile application development framework released in 2013 by Max Lynch, Ben Sperry and Adam Bradley [18]. It uses web technologies like HTML5, CSS3 and JavaScript to make mobile applications. The framework is perceived as focusing on the look, feel and user interface of the application so that the user is convinced that it is using a Native application rather than a Hybrid application. As it a Hybrid application framework, there is only one code base and its primary programming language is JavaScript which is the same as used in React Native and TypeScript [20] which is the superset of JavaScript programming language. It provides cross platform application production which means the developer have the privilege of making applications both in Android, iOS and Universal Windows Platform using just one programming language and one code base. Iconic uses Cordova [19] which is a bridge between the framework and mobile device Native components. We will not go into details of Cordova as this is not the main topic of this thesis. Ionic comes with pre-developed and pre-defined components which makes the development easier and faster for the user interface of the applications. Until now we have introduced the basic concepts of Ionic. Now we will discuss about the features that Ionic offers to the mobile application developer.

First let's dive into the technology stack used in Ionic. If we consider Ionic, it embraces technologies of web development which are HTML, CSS and JavaScript to build multi-platform applications with one code base. Ionic provides regular updates which enables the development and performance of the mobile application much faster and easier. The design and implementation in the ionic framework is straight forward. However, it does not allow various customization according to the needs of the mobile application developer.

4.2 Flutter

Flutter [23] [24] is the most closet and familiar framework to React Native when it comes to mobile development. By closet, it means that the features provided by Flutter and React Native are very much alike. For example, Flutter also provide mobile applications that give true native experience while developing from one code base. Before going into the key features of Flutter, we would discuss more about Flutter's history.

Flutter was release on May 2017 by Google. The first version of the framework was called Sky but was later changed to Flutter and previously only supported development of Android mobile applications. As the time progressed, it now supports the development of both Android and iOS mobile application. It uses Dart programming language [37] which is also developed by Google. Now let us look on the key features Flutter provides to the mobile application developer community. Figure 2 shows the Flutter architecture [44].

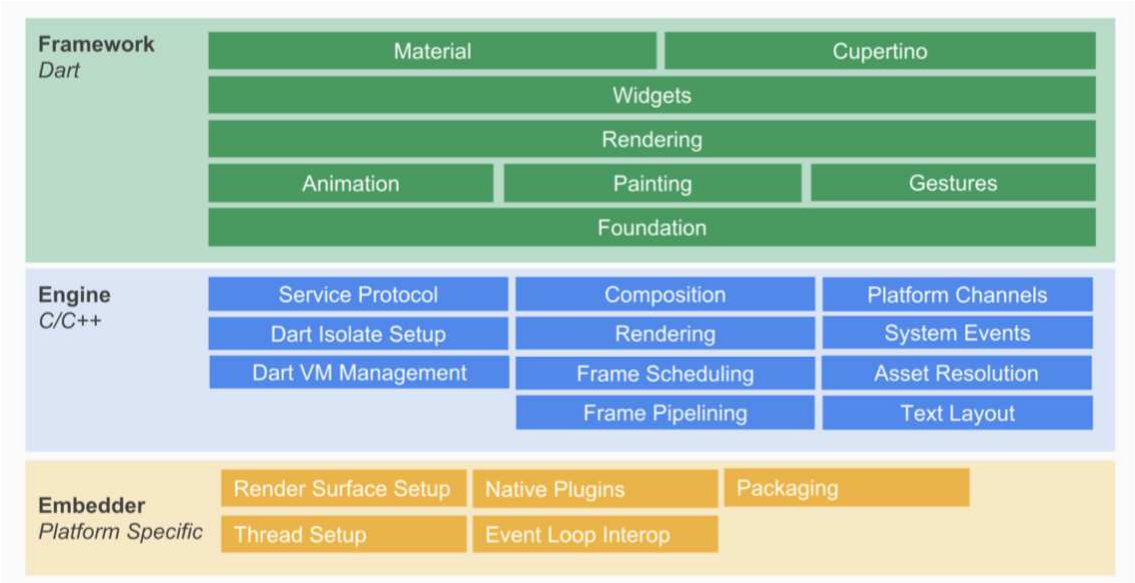

Figure 2. Flutter architecture diagram [44]

Flutter provides a large variety of built in widgets that can be customized according to the needs of the mobile application developer. It can range from animations to the build in button types with a wide range of customization availability. Flutter includes a new feature called Hot Reload which gives an access to a wide range of widget set along with working on the dynamic interface with ease. One of the most eye catching and useful feature for a good user experience is that it provides a responsive view that will dynamically change according to the size of the mobile application. Development in Flutter can be done in various IDEs (Integrated Development Environment) [36] including Android Studio and Xcode

4.3 Conclusion

We have discussed two different frameworks and their key features that set them apart from other mobile development frameworks. Table 1 will directly compares React Native, Ionic and Flutter which can help to determine which framework is the most favorable among the mobile application developer community onto this date.

	React Native	**Ionic**	**Flutter**
Language	React and JavaScript	Web Technologies – HTML, CSS, JavaScript, Angular JS and TypeScript	Dart
Nature of Apps	Cross-Platform	Hybrid Application	Cross-Platform
Developer	Facebook	Drifty.co	Google
Popular for	Native-like and elegant user interfaces across platforms using one single code base	Using single code base, you can develop an application for iOS, Android and Windows	Fast development of prototypes
Community and support	Strong and stable	Strong and stable	Emerging
GitHub Stars	70,256	35,696	42,027
GitHub Contributors	1768	272	285
Performance	Closer native look and comparatively faster as it does not use Web View	Slower than React Native due to Web View	Closer native but slower than React Native
User Experience	Extremely fast and responsive user experience	Moderate performance compared to React Native framework	Fast but less dynamic user interface compared to React Native
Hardware Accessibility	Has enough capability to run without external dependency	Needs the help of tools such as Cordova to operate	Uses a compiled programming language
Production Time	Takes more time than Ionic	Takes less time than React Native	Takes less time to build a prototype
Testing	It requires an emulator on a computer or real device to test on	It can be tested on any browser in the computer but would not give the true interface as it would be in the real mobile device.	It requires an emulator on a computer or real device to test on

Table 1. Comparison between React, Ionic and Flutter

React Native is used by companies such as Facebook, Instagram, Skype and many more which shows its popularity as well as usage among the community. React Native has a clear upper hand in the number of contributors maintaining and improving React Native when compared to Ionic and Flutter. This means React Native have more support and more probability of detecting an error and resolving it. Hence, React Native has established itself as favorites among the developer's community.

Now we will consider different frameworks performance related activities: Battery consumption, CPU usage and memory usage [22].

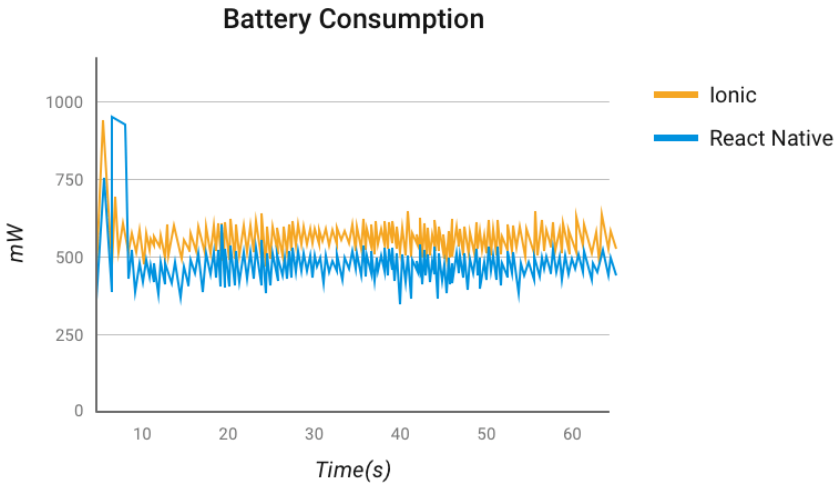

Figure 3. Battery Consumption evaluation between Ionic and React Native [22]

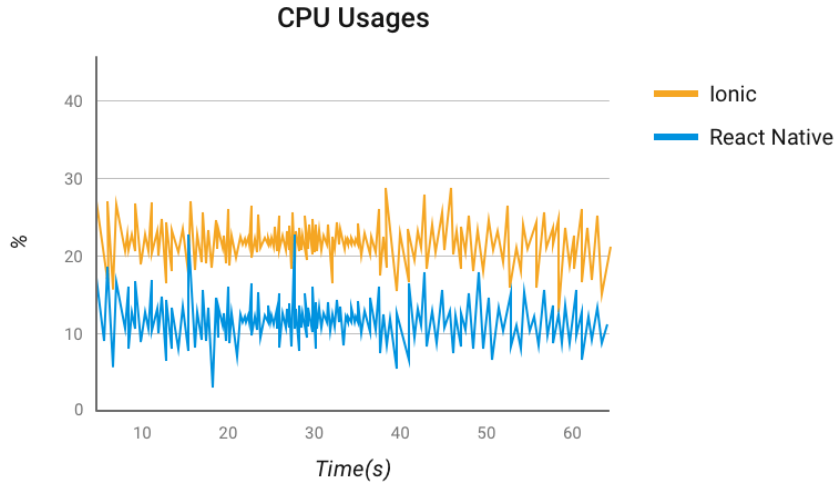

Figure 4. CPU usage evaluation between Ionic and React Native [22]

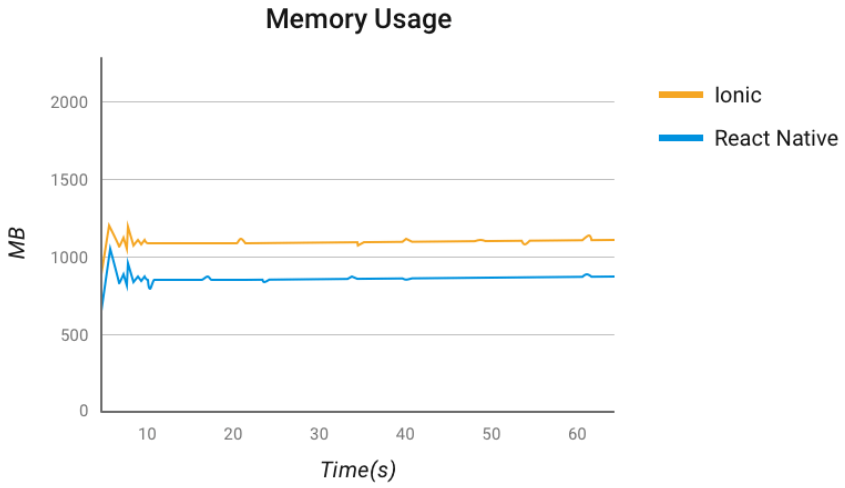

Figure 5. Memory usage evaluation between Ionic and React Native [22]

It is deduced by conducting the experiment that React Native is always better in all three aspects which makes it a more productive and agile framework to use. In the next section, we will discuss how to use React Native to produce a working mobile application to solve the problem that we discussed in Chapter 1.

5. DEVELOPING THE MOBILE APPLICATION

In this section of the thesis, we will discuss how the mobile application will solve the problem that was discussed in the introduction part of this thesis. This mobile application would be made in React Native as it concluded in the previous sections that it is the most efficient and favourable way of developing cross-platform application. Before diving into the implementation and coding, we will look at what features would be available in the mobile application that would be meaningful and help us solve the problem we are trying to solve.

5.1 Key Features

There are two essential features that would make us fulfil our objective. The first one would be to allow the users to register themselves to the service. By providing certain personal information such as name and email, a user can make a personal account which they can later edit and even upload a picture of themselves for other individuals to view. Now we have talked about other people viewing students who have registered for the account. The second feature would be a list view where people can view the information of users who have created an account. To access this feature, the users do not have to create an account because there are looking and analysing the information provided by the registered user. By this reasoning, we can conclude we have two users. One who registers them for the service and one who wants to view the information and contact the registered users. We have discussed two fundamental features that would make a functioning and effective mobile application. However, there would be two more features that would be added to enhance user experience.

One extra feature would be a feedback view in the application. This will allow both users to provide feedback about the application or any sort of complaint about registered users. This will allow to increase the user experience by providing solutions demanded by the users of the mobile application. One last feature that would be included in this mobile application would by a “Top Student of the Week” section. This feature is mainly for attracting users to register for the service and allow non-registered users to identify the most qualified registered user to contact.

Now we have discussed various features that would make up one functioning mobile application. In the next section, we will talk about the two main parts of a mobile application and how these features fit into these two.

5.2 Back – end and Front – end

A mobile application consists of two parts known as back-end and front-end [25]. As the name suggests back-end works in the anonymous area of the application and hence the process in the back-end cannot be viewed or modified by the user of mobile application. Front-end is the viewable and interactive part of the application. Front-end of the application enables users to view what functionalities and information are available to get benefit from. First we will discuss the back-end of the mobile application we are developing followed by the front-end.

The first feature that we discussed was about registering. A user will fill in the form providing certain personal information and an account will be created. A certain system of authentication would be required in the back-end to store information about users and their information so it can be retrieved and used. Hence, for the back-end of this application, Firebase [26] would be used. We will discuss about Firebase in the next section but it is an essential part for the back-end of the application. It would store the registered user's information including text and images. It will also provide an authentication system that would filter out any fraudulent activity when a user registers for an account. The second feature that our application provides is the list of all registered users in the application. With the help of back-end service, the data/information in the list will be provided when it is fetched from Firebase. All in all, Firebase is essential for a back-end for storing and retrieving data within the application.

Front-end plays a key role when it comes to mobile application development. Front-end consists of the views and interactivity of the application which is the main concern for the user. If the front-end is appealing, the user would have good user experience and would be encouraged to be use the application more often. However, without understanding the user needs and implementing features that the user does not want would lead to a bad user experience that would be reflected in the front-end of the application. For the application, every part of view that would be visible would be included in the front-end. The forms that would be used for registering and providing feedback, list of all registered users and every other visible and interactive part of the application would be the front-end.

Now we have discussed key features and the division of these key features. Now we will dive into how to implement these key features in our application in the next section.

5.3 Libraries

In this section, we will discuss about third party libraries or modules that will help us accomplish the functions we are going to implement in the mobile application. Some of the modules that would be discussing are:

1. Firebase [26] [27]
2. React-native-animatable [28]
3. React-native-image-picker [29]
4. React-native-vector-icons [30]
5. React-navigation [31]
6. Tcomb-form-native [32]
7. React-native-indicators [33]
8. React-native-app-intro-slider [34]

These are main libraries that we would be using to make a functioning mobile application. All libraries have certain number of contributors and constantly reviewed by professional and hence improved overtime. One of the GitHub account [35] compares different similar types of packages including contributors and developers using it. Now we would discuss each of them in detail and get insights on how to integrate into our own mobile application.

5.3.1 Firebase

Firebase [26] is a key element in building the back-end of the application. Firebase is a library developed by James Tamplin and Andrew Lee in 2012 and later was acquired by Google in 2014. Firebase offers a lot of power to the developer including:

1. Authentication
2. Real time Database
3. Cloud Storage
4. Cloud Functions
5. Machine Learning Kit
6. Hosting
7. Cloud Fire Store

There is a lot we can do with firebase module but we will only use authentication, real time database and cloud storage functionalities in our application. Firebase authentication allows us to register users using email or third party applications such as Facebook or Twitter. However, in our application, we would be using simple email login and registration. The line of code that will be used to register the user would be:

```
1.  firebase.auth().createUserWithEmailAndPassword(email, password)
```

Program 1 *– Simple line of code to register a user*

The email and password would be provided by the user. If a user already exists or the email address is not valid, the registration would fail. After a successful registration, the user will automatically login and can see it personal profile information. Normally, it will take a lot of time to self-construct an authentication system and even after developing one, there should be one dedicated team to always monitor and maintain the system. However, with firebase, with few lines of code, we can achieve the same task with less time and effort.

Real time database and cloud storage are also providing the privilege of saving time and effort by allowing the developers to use the ready-made services instead of developing one and further maintaining it. Real time database allows us to save user's personal information and provide it when demanded. Cloud storage facility saves the picture of each individual user if they upload one. When the user registers, all the data is saved it into the real-time database along with the authentication data. When the user has successfully registered, it will upload his/her personal photograph and it will be saved it into the cloud storage with a unique identification number that would be saved in the user's personal information. Both real-time database and cloud storage provides built-in security features of who can read and write in the database. Hence, when a registered user is editing his/her profile, his/her has the privilege to read and write his/her own personal data whereas when a non-registering user is using the application, he/she has the privilege of reading all the data in the database but could not edit, remove or add new data. Consequently, Firebase is becoming a clear choice as the back-end service which provide various features including security.

5.3.2 React-native-animatable

React-native-animatable library [28] provides the privilege of animating different views in the mobile application in various ways. Animations in the application enhance user experience which leads to increased usage of the application. React-native-animatable library, at the time of writing this thesis, have 19 contributors and 5,841 stars which reflects its popularity and usage among the developers. It is famous for easy usage as well as the different techniques that can be utilised to animate a view in the mobile application. Let us look up into an example of how a simple view would be constructed and how that simple view can be changed into animatable view.

```
<Text style={styles.heading}>
        Registration
<Text>
```

Program 2 - *Simple Program*

```
import * as Animatable from 'react-native-animatable';

<Animatable.Text animation="pulse" easing="ease-out" iterationCount="infi-
nite" style={styles.heading}>
        Registration
</Animatable.Text>
```

Program 3 - *Animated Program*

5.3.3 React-native-image-picker

React-native-image-picker [29] is one of the crucial module that is utilized for the information that will be provided by the user. A registered user will like to upload his or her own personal picture so that non-registered or even registered users can identify him or her. This is where the role of react-native-image-picker comes in. The library allows the user to pick an image from the gallery or take a picture from a camera. When the image is successfully fetched, it is saved in the Firebase cloud storage facility with a unique id. The partial code to implement this function would look like this:

```
uploadPic(){
        ImagePicker.launchImageLibrary(imageOptions, (response) => {

            if (response.didCancel) {
                console.log('User cancelled image picker');
            }
            else if (response.error) {
                console.log('ImagePicker Error: ', response.error);
            }
            else {
                this.storage = firebase.storage().ref(this.props.userId)
                let uri = RNFetchBlob.wrap(response.path)
                Blob.build(uri, {type: response.type})
                  .then((blob) => {
```

```
                    this.storage
                      .child('profilePic.jpg')
                      .put(blob, {contentType: response.type})
                        .then( () => {
                            this.storage.child('profilePic.jpg').getDown-
loadURL().then( (url) => {
                                this.setState({profilePic:url})
                                firebase.database().ref('us-
ers/'+this.props.userId).update({
                                    profilePic: url
                                })
                            })
                        })
                })

            }
        });
    }
```

Program 4 – *Code use to upload pictures and do error handling*

5.3.4 React-native-vector-icons

React-native-vector-icons provides various icons that would be adjusted to the size of the mobile screen. Currently, react-native-vector-icons have 63 contributors and 9,777 stars in GitHub which reflects its popularity among developers [30]. Icons are essential part of the user experience and react-native-vector-icons module provide a vast range of them. This module simply allows the icon to render on the mobile screen by calling out its name. The size and the colour of the icons can also be modified by the developer's own choice. One simple example to render an icon with a size 30 and colour blue can be done as in Program 5.

```
import FontAwesome from 'react-native-vector-icons/FontAwesome';

<FontAwesome
          name={'home'}
          size={30}
          style={{ color: 'blue' }}
/>
```

Program 5 – *Rendering an icon using react-native-vector-icons package*

5.3.5 React-navigation

Navigation around the application is an essential part as it describes into how many parts the application is divided into. React-navigation [31] provides us the facility to easily integrate navigation bars into the mobile application. It provides with a wide range of options including top navigation bar and drawer navigation bar. However, in this application, bottom navigation bar is used. There will be four navigation tabs in the mobile application. The first one will be the list of the registered students; the second will have the top student of the week and the third navigation tab will allow the user to register, sign up and login in the service. The fourth tab will allow the user to give feedback about the application and a form would be rendered to allow this function. A simple extract of implementing of navigation system is shown in Program 6.

```
1.  import { createBottomTabNavigator } from 'react-navigation';
2.
3.  const Navigation = createBottomTabNavigator({
4.      STList: { screen: STList },
5.      TopST: { screen: TopST },
6.      Home: { screen: Home },
7.      MainProfile: { screen: MainProfile },
8.      Feedback: { screen: Feedback }
9.  },{
10.     initialRouteName: 'Home',
11.     tabBarOptions:{
12.         activeTintColor: '#F29400',
13.         style: { backgroundColor: 'black' },
14.         showLabel: false,
15.     }
16. })
```

Program 6 – *Creating multiple screens using react navigation package*

6. CONCLUSION

Mobile applications have been used massively in the last decade and it will be no surprise that it will gain superiority over web applications in the future. This leads to the thinking that users should be provided with such an application that will provide them an exceptional user experience.

Three frameworks, React Native, Flutter and Ionic, for mobile application development were discussed and compared. We discussed some similarities and key differences among these frameworks that distinguish themselves and help us evaluate which framework is the most popular among the mobile application development community.

The aim of the thesis was to make a mobile application in React Native and prove why React Native is the preferred choice of making mobile applications. We also discussed different technologies and methods used along with React Native to achieve a production ready mobile application. You can find the whole code for the development of this application on GitHub [39]. The development of this mobile application concludes the thesis. Hopefully, insights for readers have been provided about mobile applications and its frameworks and would possibly serve future references and research in the field.

The main aim of the thesis was to solve the problem that was discussed in the introduction part of this thesis. That is to help the students, who are not financially stable to afford their education, get a teaching job that will support and aid them in their daily life. After creating the mobile application in React Native, online survey was conducted to know that impact students had in their life with the introduction of this mobile application. The survey was conducted according to the article of “Good practice in conducting and reporting of survey research” [40]. First it was studied what kind of survey is to be conducted such as face to face, questionnaire or observation. As this mobile application was targeted at Pakistani student, face to face and observation was not feasible. Hence, an online questionnaire was constructed. The online survey form was made using Google Forms [41] and was distributed using the email student used to create an account in the Student Teach application. After reading research papers how to conduct a survey and creating quality questions, three feedbacks were selected. The survey answers are uploaded in the google drive for the readers to analyze [42]. Student were asked if this application helped them in financial terms. One of the response was “*Somehow, some people contacted me but was not able to negotiate due to travel constraints. I think with some more marketing, this application can help students like us make a better side income to support our education*.”

REFERENCES

[1] Vpn mentor – 06.04.2019 - Trends in mobile and internet usage. Available: https://www.vpnmentor.com/blog/vital-internet-trends/ (Latest Access 14.04.2019)

[2] Davd Tilson; Carsten Sorensen; Kalle Lyytinen – 09.02.2012 - iOS Operating System introduction: Available: https://ieeexplore.ieee.org/document/6148683 (Latest Access 14.04.2019)

[3] Technopedia – 06.04.2019 - Android Operating System. Available: http://www.techopedia.com/definition/4219/android-platform (Latest Access 14.04.2019)

[4] Constantin, Novac Ovidiu; Robert-Gyula, Marczin; Cornelia, Novac Mihaela – 10.2016 - Hybrid and Native Applications comparison. Available: https://search-proquest-com.libproxy.tut.fi/docview/1857759973?pq-origsite=summon (Latest Access 14.04.2019)

[5] Google – 06.04.2019 - Ionic framework specification. Available: https://ionicframework.com (Latest Access 14.04.2019)

[6] Stefan Bosnic; Istvan Papp; Sebastian Novak – 19.01.2017- Cordova specifications. Available: https://ieeexplore.ieee.org/document/7818919 (Latest Access 14.04.2019)

[7] Dev Team – 25.04.2018 - Ionic mobile applications examples. Available: https://csform.com/top-10-apps-built-with-ionic-framework/ (Latest Access 14.04.2019)

[8] Facebook – 06.04.2019 - React Native official website. Available: https://facebook.github.io/react-native/ (Latest Access 14.04.2019)

[9] Pluralsight – 06.04.2019 - JavaScript official website. Available: https://www.javascript.com (Latest Access 14.04.2019)

[10] Statista – 06.04.2019 - Share of mobile operating systems in the world. Available: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/ (Latest Access 14.04.2019)

[11] Statista – 06.04.2019 - Share of internet used in mobile. Available: https://www.statista.com/topics/779/mobile-internet/ (Latest Access 14.04.2019)

[12] Statista – 06.04.2019 - Share of mobile operating system. Available: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/ (Latest Access 14.04.2019)

[13] Facebook – 26.03.2015 - React Native release announcements. Available: https://code.facebook.com/posts/754869551286944/f8-big-technology-bets-and-open-source-announcements/ (Latest Access 14.04.2019)

[14] Github – 06.03.2019 - React Native stars no GitHub. Available: https://github.com/search?p=2&q=stars%3A%3E1&s=stars&type=Repositories (Latest Access 14.04.2019)

[15] Google – 06.03.2019 - Google trend on React Native. Available: https://trends.google.com/trends/explore?q=ios%20development,android%20development,react%20native (Latest Access 14.04.2019)

[16] Google – 06.03.2019 - Flutter official website. Available: https://flutter.io (Latest Access 14.04.2019)

[17] Google – 06.03.2019 - Ionic official website. Available: https://ionicframework.com (Latest Access 14.04.2019)

[18] Hugo Brito; Anabela Gomes; Alvaro Santos e Jorge Bernardino – 28.06.2018 - Ionic history. Available: https://ieeexplore.ieee.org/document/8399283 (Latest Access 14.04.2019)

[19] Stefan Bosnic; Istvan Papp; Sebastian Novak - 19.01.2017 - Phone Gap and Cordova. Available: https://ieeexplore.ieee.org/document/7818919 (Latest Access 14.04.2019)

[20] Microsoft - 06.04.2019 - Typescript official website. Available: https://www.typescript-lang.org/ (Latest Access 14.04.2019)

[21] Rakshit Soral – 28.02.2018 - Performance of ionic and React Native. Available: https://www.simform.com/react-native-vs-ionic/ (Latest Access 14.04.2019)

[22] Rushi Trivedi – 23.04.2018 - Ionic and React Native differentiation: Available: https://www.websoptimization.com/blog/ionic-vs-react-native/ (Latest Access 14.04.2019)

[23] Google – 06.04.2019 - Flutter official website. Available: https://flutter.io (Latest Access 14.04.2019)

[24] Krill, Paul; San Mateo – 01.03.2018 - Information about Flutter framework. Available: https://search-proquest-com.libproxy.tut.fi/docview/2010964253/fulltext/3646D9F92DB744D3PQ/1?accountid=27303 (-Latest Access 14.04.2019)

[25] Karlos Artto; Tuomas Ahola; Valtteri Vartiainen – 02.2016 - Back-end and Front-end Information: Available: https://www-sciencedirect-com.libproxy.tut.fi/science/article/pii/S0263786315000897 (Latest Access 14.04.2019)

[26] Google – 06.04.2019 - Firebase official website: Available: https://firebase.google.com/ (Latest Access 14.04.2019)

[27] Google – 06.04.2019 - Firebase npm package. Available: https://www.npmjs.com/package/firebase (Latest Access 14.04.2019)

[28] Oblador – 06.04.2019 - React-native-animatable GitHub repository. Available: https://github.com/oblador/react-native-animatable (Latest Access 14.04.2019)

[29] React-native-community – 06.04.2019 - React-native-image-picker GitHub repository. Available: https://github.com/react-community/react-native-image-picker (Latest Access 14.04.2019)

[30] Oblador – 06.04.2019 - React-native-vector-icon GitHub repository. Available: https://github.com/oblador/react-native-vector-icons (Latest Access 14.04.2019)

[31] React-navigation – 06.04.2019 - React-navigation official website. Available: https://react-navigation.org/ (Latest Access 14.04.2019)

[32] Giulio Canti – 06.04.2019 - Tcomb-fom-native GitHub repository. Available: https://github.com/gcanti/tcomb-form-native (Latest Access 14.04.2019)

[33] Alexander Nazarov – 06.04.2019 - React-native-indicators GitHub repository. Available: https://github.com/n4kz/react-native-indicators (Latest Access 14.04.2019)

[34] Jacob Lauritzen – 06.04.2019 - React-native-app-intro-sliders GitHub repository. Available: https://github.com/Jacse/react-native-app-intro-slider (Latest Access 14.04.2019)

[35] Dotan J.Nahum – 06.04.2019 - React-native-packages-comparison GitHub repository. Available: https://github.com/jondot/awesome-react-native (Latest Access 14.04.2019)

[36] LinkedIn – 06.04.2019 - Integrated Development Environment. Available: https://www.lynda.com/Development-Tools-tutorials/What-IDE/486760/505357-4.html (Latest Access 14.04.2019)

[37] Google – 06.04.2019 - Dart programming language. Available: https://www.dartlang.org/ (Latest Access 14.04.2019)

[38] Tampere Univerity – 06.04.2019 - Andor. Available: https://tut.summon.serialssolutions.com/ (Latest Access 14.04.2019)

[39] Muhammad Ali Gulzar – 06.04.2019 - Application code on GitHub. Available: https://github.com/ali-gulzar/StudentTeacher (Latest Access 14.04.2019)

[40] Kate Kelley; Belinda Clark; Vivienne Brown; John Sitzia – 03.05.2003 - Survey Methodology. Available: https://academic.oup.com/intqhc/article/15/3/261/1856193 (Latest Access 14.04.2019)

[41] Google – 06.04.2019 - Google Forms. Available: https://www.google.com/forms/about/ (Latest Access 14.04.2019)

[42] Muhammad Ali Gulzar – 06.04.2019 - Survey Results. Available : https://docs.google.com/spreadsheets/d/1Y9xNe8z6w7Qj4ji32dMaPDKP2ElGlrQo4GyB_LsXogA/edit?usp=sharing (Latest Access 14.04.2019)

[43] Kalle A.Piirainen; Rafael A.Gonzalez – 11.06.2013 - Design Science and Constructive Research. Available: https://link.springer.com/chapter/10.1007/978-3-642-38827-9_5 (Latest Access 14.04.2019)

[44] Rodriguez-Sanchez Guerra; Manuel – 01.10.2018 - Scientific Research on comparison of different mobile applications frameworks. Available: https://rodin.uca.es/xmlui/handle/10498/20951 (Latest Access 14.04.2019)