**“SWEETCHAT”**

**– An Android Based Mobile Application**

**PROJECT By**

**MD. MAHFUJAR RAHMAN**

**ID: 1737620108**

**Department of Computer Science and Engineering,**

**PART III ODD, Session: 2016-2017**

**Rangpur Engineering College, Rangpur**

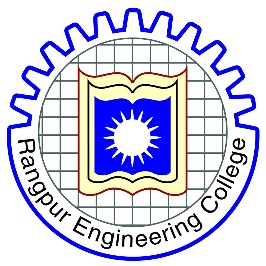
**SUPERVISED BY**

**MD. OMAR FARUK KHAN**

**Lecturer**

**Department of Computer Science and Engineering,**

**Rangpur Engineering College, Rangpur**

****

**RANGPUR ENGINEERING COLLEGE, RANGPUR**

(Affiliated by University of Rajshahi)

September, 2019

**ACKNOWLEDGEMENT**

First, we express our heartiest thanks and gratefulness to almighty Allah for His divine blessing makes us possible to complete this project successfully.

We feel, grateful to and wish our profound our indebtedness to **MD. OMAR FARUK KHAN**, Lecturer, Department of Computer Science and Engineering (CSE), Rangpur Engineering College, Rangpur. Deep Knowledge & keen interest of our supervisor in the field of “**SWEETCHAT**” -an android based mobile application to carry out this project. His endless encouragement, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, especially friendly learning environment and reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to thank our entire course mate at Rangpur Engineering College, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our beloved parents.

**Abstract**

This project provides the online chatting service named SWEETCHAT. This chat service provides the best option to users according to their flexibility. SWEETCHAT is a user friendly application which is easily accessible from any android based smartphone or Tab. SWEETCHAT gives the user flexibility to send text messages or other files according to their needs. It will be time efficiency because of instant messaging. Where one can see the received message from another user. There is no static cost service.

**REPORT LAYOUT**

**Chapter 1: Introduction**

In this chapter we have discussed about the introduction, motivation, projects objective, scope of the project and the expected outcome of the project. Later followed by the report layout.

**Chapter 2: Background**

We discuss about the background circumstances of our project. We also talk about the related works, comparative studies and challenges of the project.

**Chapter 3: Requirement Specification**

The chapter is all about the requirements like business process modeling and use case model of the project and their description, the logical data model and the design requirements.

**Chapter 4: Design Specification**

In this chapter, all the designs of the project are described. Front-end design for example- login page, sign up page and back-end design.

**Chapter 5: Implementation and Testing**

This chapter contains the implementation of database, Implementation of front-end design, Implementation of Java and XML code, Test Implementation and the test results of the projects.

**Chapter 6: Conclusion and Future Scope**

In the conclusion part we discussed about limitation of our application and the scope for further developments which pretty much derive about the project.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CHAPTER****1: INTRODUCTION**   * 1. Introduction   2. Project Objective   **CHAPTER 2: BACKGROUND**  2.1 Introduction  2.2 Related works  2.2.1 WHATSAPP Application  2.2.2 VIBER Application  2.2.3 WECHAT Application  2.2.4 TELEGRAM Application  2.3 Challenges  **CHAPPTER 3: REQUIREMENT SPECIFICATION**  3.1 Business Process Modeling  3.2 Use Case Modeling and Description  3.2.1 Use Case description  3.3 Logical Date Model  3.4 Design Requirement  **CHAPTER 4: DESIGN SPECIFICATION**  4.1 Front-end Design  4.2 Back-end Design  **CHAPTER 5: IMPLEMENTING AND TESTING**  5.1 Implementation of Database  5.2 Implementation of Front-end Design  5.3 Implementation of Java and XML code  5.4 Test Implementation  5.5 Test Results and Reports  **CHAPTER 6: CONCLUSION AND FUTURE SCOPE**  6.1 Discussion and Conclusion  6.2 Limitations of our application  6.3 Scope for Future Development | **01-01**  01  01  **02-10**  04  04  05  06  07  08  09  10  **11-19**  11  13  14  17  19  **20-27**  20  27  **28-32**  28  28  29  30  32  **33-34**  33  33  33 |

**APPENDIX 35**

**REFERENCS 36**

**LIST OF FIGURES**

|  |  |
| --- | --- |
| **FIGURES** | **PAGE NO** |
| Figure 2.1: A Screenshot of WHATSAPP Application | 05 |
| Figure 2.2: A Screenshot of VIBER Application  Figure 2.3: A Screenshot of WECHAT Mama Application  Figure 2.4: A Screenshot of TELEGRAM Rides Application  Figure 3.1: Data Flow Diagram of the Propose System  Figure 3.2: Use Case Modeling of SWEETCHAT  Figure 3.3: Logical Data Model  Figure 4.1: A Screenshot of Front-end Design of SWEETCHAT  Figure 4.2: A Screenshot of Sign Up page of SWEETCHAT  Figure 4.3: A Screenshot of Sign in page of SWEETCHAT  **LIST OF-TABLE** | 06  07  08  12  13  18  20  21  21 |
| **TABLE**  Table 3.1: Use case description of registration  Table 3.2: Use case description of Login  Table 3.3: Use case description of profile setting  Table 3.4: Use case description of find friends  Table 3.5: Use case description private messages  Table 3.6: Use case description get message request  Table 3.7: Use case description accept request | **PAGE NO**  09  14  14  15  15  16  16 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Introduction**

SWEETCHAT is an online-based android mobile application. This is an online chatting app by which one user can messages to another. Here user can send text messages as well as images files and other files. Other messaging options like online audio calling or online video calling system ca be implemented in future.

Users must have to create an account firstly. Users can be able to send friend request or receive friend request. If two user are online friend, then they will be able to send message or files.

* 1. **Project Objective**
* People can easily register/login using their email or phone number.
* People can easily send message or other files instantly.
* With this mobile application, people will be able to use this app with internet connection.
* It is a user-friendly Android application for users.
* Saving the valuable time and easily getting the messages.
* Users can save money.
* People will easily understand all the features of this android application.

**CHAPTER 2**

**BACKGROUND**

**2.1 Introduction**

We designed interactive app which can be operated in many devices based on Android OS through internet. In our application users connectivity will enriched. They can collaborate with each other. For the people, Our application represents as a new medium to collaborate.

**2.2 Related work**

Now a day, there are many Application those are working on the field of online messaging. The aim is helps to the peoples in an efficient messaging service. Here we have listed a few high indexed smart phone Application those we have found after searching <https://play.google.com/store/apps>.

Top 4 online messaging service related to our application in Bangladesh are:

1. WHATSAPP Application
2. VIBER Application
3. WECHAT Application
4. TELEGRAM Application

There are many more also.

There are many mobile applications also in Google Play Store which are helpful for messaging service with users.

**2.2.1 WHATSAPP Application**

**WhatsApp** is a freeware, cross-platform messaging and Voice over IP (VoIP) service owned by Facebook. It allows users to send text messages and voice messages, make voice and video calls, and share images, documents, user locations, and other media.

WhatsApp was founded in 2009 by **Brian Acton** and **Jan Koum**, former employees of Yahoo!.. Koum named the app WhatsApp to sound like "what's up". On February 19, 2014, months after a venture capital financing round at a $1.5 billion valuation. Facebook announced it was acquiring WhatsApp for US$19 billion, its largest acquisition to date. Figure 2.1 shows the WHATSAPP application home page.

.

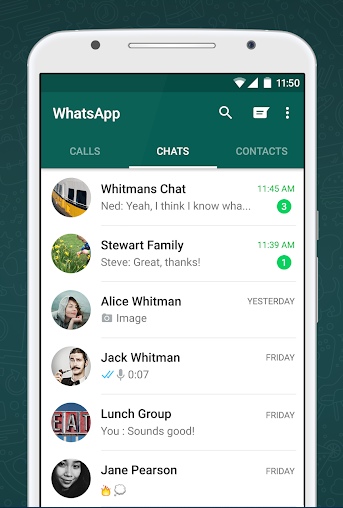


Figure 2.1: A Screenshot of WHATSAPP Application [2]

**2.2.2 VIBER Application**

**Viber** is a cross-platform voice over IP (VoIP) and instant messaging (IM) software application operated by Japanese multinational company Rakuten, provided as a freeware for the Android, iOS, Microsoft Windows, macOS and Linux platforms. In addition to instant messaging it allows users to exchange media such as images and video records. Viber Media might have been founded in **Tel Aviv**, Israel, in 2010 by **Talmon Marco** and **Igor Magazinnik**, who are friends from the Israel Defense Forces where they were chief information officers. On February 13, 2014, Rakuten announced they had acquired Viber Media for $900 million.

Figure 2.2 shows the VIBER application home page.

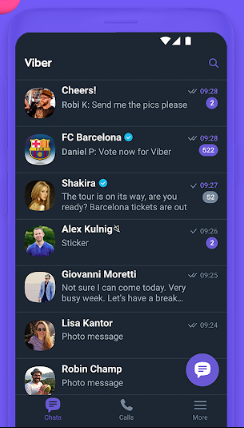


Figure 2.2: A Screenshot of VIBER Application [3].

**2.2.3 WECHAT Application**

**WeChat** is a Chinese multi-purpose messaging social media and mobile payment app developed by Tencent. It was first released in 2011, and become one of the world’s largest standalone mobile apps in 2018, with over 1 billion monthly active users. WeChat has been described as China's "app for everything" and a "super app" because of its wide range of functions. **WeChat** began as a project at Tencent Guangzhou Research and Project center in October 2010. The original version of the app was created by **Zhang Xiaolong** and named "Weixin" by Ma Huateng, CEO of Tencentand launched in 2011.

Figure 2.2 shows the WECHAT application home page.

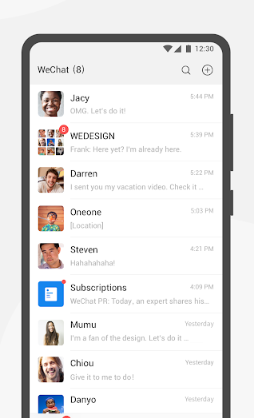


Figure 2.3: A Screenshot of WECHATApplication [4].

**2.2.4 TELEGRAM Application**

**Telegram** is a cloud-based instant messaging and voice over IP service. Telegram client apps are available for Android, iOS, Windows Phone, Windows NT, macOS and Linux. Users can send messages and exchange photos, videos, stickers, audio and files of any type.

Telegram's client-side code is open-source software but the source code for recent versions is not always immediately published, whereas its server-side code is closed-source and proprietary.The service also provides APIs to independent developers. In March 2018, Telegram stated that it had 200 million monthly active users.

Figure 2.4 shows the TELEGRAM application home page.

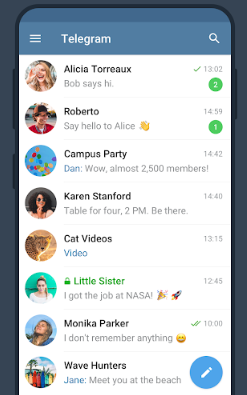


Figure 2.4: A Screenshot of SHOHOZ Rides Application [5].

**2.4 Challenges**

When any developer wants to make anything different type of thinking, then the developer has face to some challenges. As like this situation, our project has some different types of challenges.

There are some challenges are-

* Connecting Google Firebase API
* Handling the Data.

We believe that one day it will be place as favorite application of the user because of user friendly and helpful work in the society. Now this is our main goal. Now for completing our mission, all those things might be challenging for us.

**CHAPTER 3**

**REQUIREMENT SPECIFICATION**

**3.1 Business Process Modeling**

Business process modeling (BPM) is a modern process and methodology. Which the represented the activity of an enterprise of a system engineering to improve or analysis the current process. In this process one can easily represent their workflow of a system. The main characteristic of the methodology is based on diagram as ‘Flow Diagram’. Here we are trying to describe our project’s business model using data flow diagram. Data flow diagram describes how data is processed through a system or project [8].

Data flow diagram is one of the most useable Diagram to show the work flow of a system. It’s easy and understand to any workflow. We used level 1data flow diagram for our work. Figure 3.1 shows the data flow diagram of the propose system.

Already Logged In?

(Check Session)

Yes

No

Registration

Login

No

By Phone

By Email

Is Successful?

Yes

No

Is Successful?

User Account

Private Chat

Group Chat

Yes

Contacts

Requests

Update Profile

Find friends

Create Group

Settings

Log Out

Figure 3.1: Data Flow Diagram of the Propose System

**3.2 Use Case Modeling and Description**

A use case is a list of actions or event steps typically defining the interactions between a role and a system to achieve a goal. Figure 3.2 shows the use case modeling of SWEETCHAT application.

**User**

Figure 3.2: Use Case Modeling of SWEETCHAT

**3.2.1 Use case Description**

Table 3.1: Use case description of registration

|  |  |
| --- | --- |
| Use Case | Registration |
| Primary Actor | User |
| Secondary Actor | Null |
| Pre-condition | Null |
| Scenario | Enter Email address  Enter password minimum of 8 character |
| Enter valid phone number (+880174\*\*\*\*\*\*\*)  Enter Valid Confirmation Code |
| Post-condition | Registration successfully or failed  Edit profile  Update profile(user-name and user-status) |

Table 3.2: Use case description of login

|  |  |
| --- | --- |
| Use Case | Login |
| Primary Actor | User, |
| Secondary Actor | Null |
| Pre-condition | Registration |
| Scenario | Enter valid Email address/ Phone Number  Enter password / Phone confirmation code |
| Post-condition | Login successfully or failed  Edit profile  Update profile |

Table 3.3: Use case description of profile setting

|  |  |
| --- | --- |
| Use Case | Profile setting |
| Primary Actor | User |
| Secondary Actor | Null |
| Pre-condition | Login |
| Scenario | Update profile photo  Update basic information |
| Post-condition | Update successfully or failed  View profile |

Table 3.4: Use case description of find friends

|  |  |
| --- | --- |
| Use Case | Find Friends |
| Primary Actor | User |
| Secondary Actor | Null |
| Pre-condition | Login |
| Scenario | Search your friends  Send message request |
| Post-condition | Show result successfully or failed  View information  Get Friends |

Table 3.5: Use case description private message

|  |  |
| --- | --- |
| Use Case | Private Message |
| Primary Actor | User |
| Secondary Actor | Null |
| Pre-condition | Login |
| Scenario | Select Friend  Type Message  Send Message |
| Post-condition | Show Message successfully send or not  View Message |

Table 3.6: Use case description get message request

|  |  |
| --- | --- |
| Use Case | Get Message request |
| Primary Actor | User |
| Secondary Actor | Null |
| Pre-condition | Login |
| Scenario | View Message Request  Request pending  Select request |
| Post-condition | Accept or Reject |

Table 3.7: Use case description accept request

|  |  |
| --- | --- |
| Use Case | Accept request |
| Primary Actor | User |
| Secondary Actor | Null |
| Pre-condition | Login |
| Scenario | Show message request  View profile  Add Contact  Accept request |
| Post-condition | Contact add  Show contact successfully added or failed |

**3.3 Logical Data Model**

Logical data model mainly consisted of few elements like, data entities, attributes and key and relationship between the entities. By which the organization data and business rule is defined and govern the relationship between them. Implementation of the conceptual data model is considered by logical data model. Figure 3.3 shows the logical data model [9].

User Information

User

has



Message Request

Request Type

Figure 3.3: Logical Data Model

**3.4 Design Requirement**

* In our system, we have an authentication for our user. Every users have to complete registration process by using email or phone. As only registered users can send messages each other.
* We used Google Firebase authentication system for email and phone authentication.
* Every registered users must have a profile, after registration user must have to complete user profile info (username, user-status, etc.)
* We used Google Firebase Database to store the user’s information as well as their messages.
* Users need to send message request or accept message request to contact each other using our system.
* Overall, the Google Firebase API makes our work easier.

We are trying to build our as complex free. We think about user when we design our application. Because of this reason, our android application is so user friendly.

**CHAPTER 4**

**DESIGN SPECIFICATION**

Design specification is a statement of how a design is developed. In the section of design specification, we try to show the front-end and back-end design of our smartphone application. We also discussed about many tools and platforms, which we use to develop our application.

**4.1 Front-end Design**

The front-end is everything involved with what the users sees, including design and some languages. The front-end design is the interface users see when he/she opens the application. That means to keep the users interested in the application, the most important part of a project is frond-end designing. Usually most of the users expect a simple user interface from the developer. If the front-end design is so complex, the application is fails to attract of the user. Figure 4.1 shows a front-end design of SWEERTCHAT application.

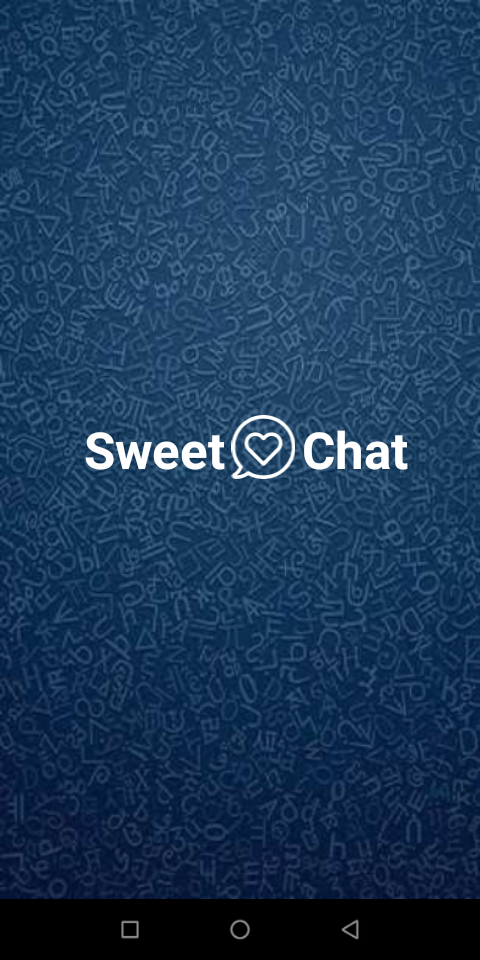
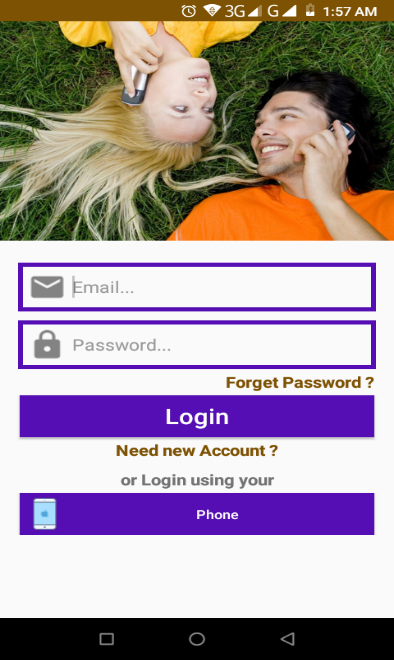


Figure 4.1: A Screenshot of Front-end Design of SWEETCHAT

In this figure shows the front page of the user interface. Users will see it every time they open the application. We tried to make the graphical user interface easily accessible to the user. But it is really tough to keep the interface simple with this much of facilities. Yet we tried our best to give the users the best experience. Hope the users will find it easily assessable and get benefitted from this service. Figure 4.2 shows the Signup page of SWEETCHAT application. First time user can Sign up your account and update profile information. Now they can sign in using their email and password. Figure 4.3 is a snapshot of SWEETCHAT application Sign in page. User can also continue sign in with their phone numbers. Figure 4.4 is a snapshot of SWEETCHAT application Phone Log in page.



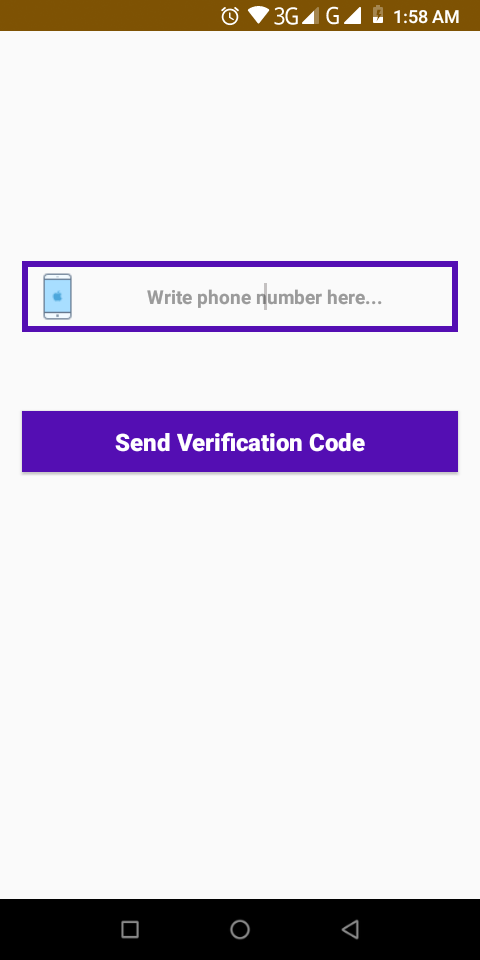
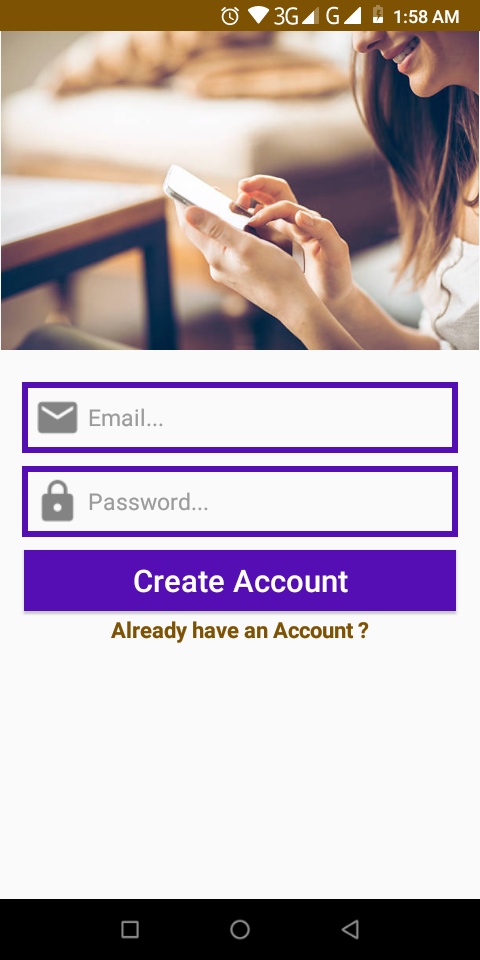


Figure 4.3: Email Sign in page

Figure 4.4: Phone log in page

Figure 4.2: Sign Up Page

After a successful login, the users must update their profile (username and user-status is compulsory) and can click the update button to save or update the profile. If user set his/her username, it cannot be possible to change the username twice. User can also update his/her profile photo as they wish.

Figure 4.5 is a snapshot of SWEETCHAT application Profile Update Page.

Figure 4.6 is a snapshot of SWEETCHAT application Profile Picture Update Page.

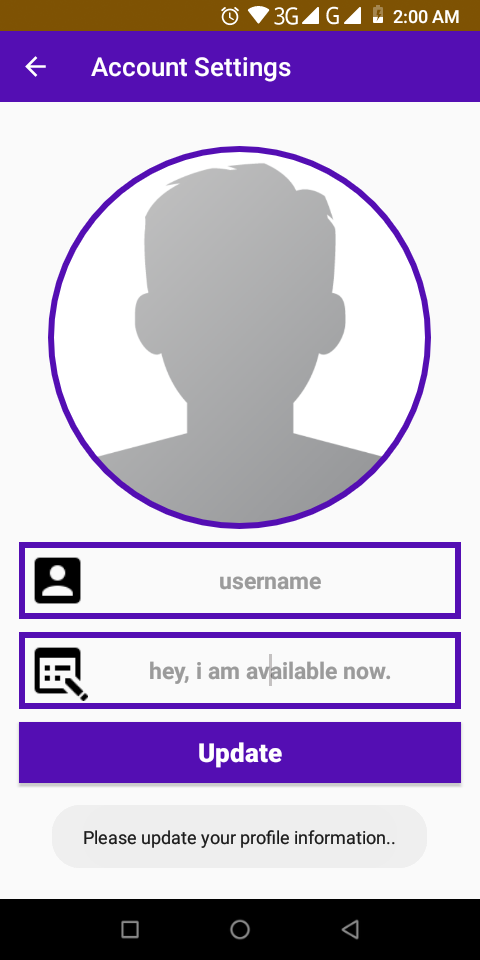


Figure 4.6: A Screenshot of Updating Profile picture

Figure 4.5: A Screenshot of Account Settings

Figure 4.6 is the rider profile information. Rider can easily profile create and updated. And finally click the save button.

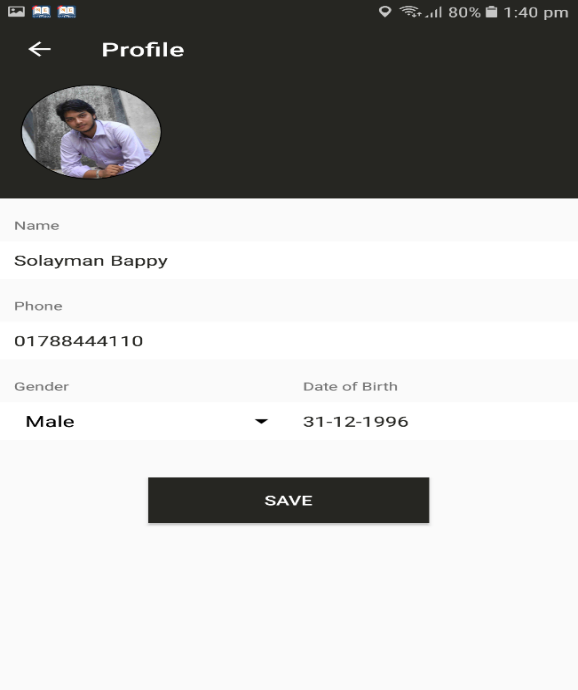


Figure 4.6: A Screenshot of Rider profile of EMERGENCY ride

In Figure 4.7 it is shown how a user will search for a specific place or destination. Figure 4.8 it is shown how a user will find a driver. First time user can select your riding option and click the find driver button.

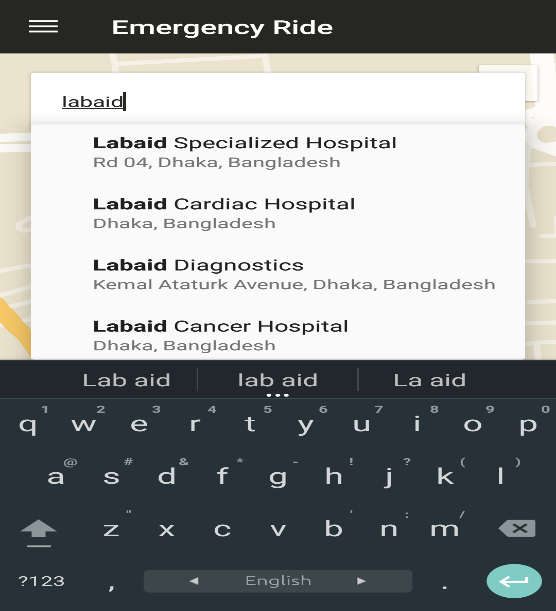
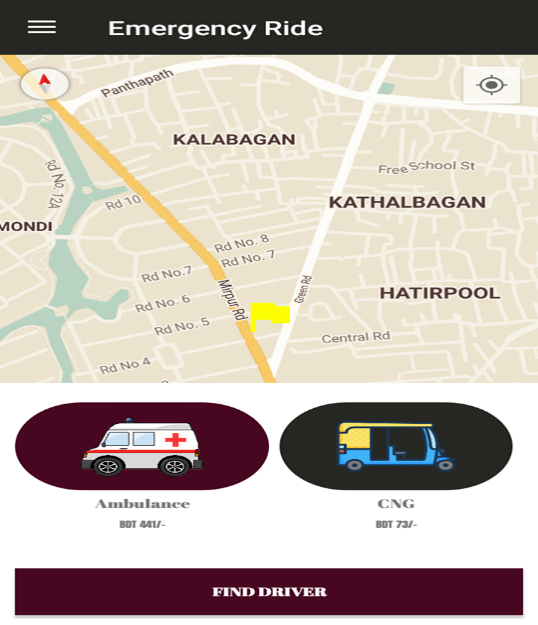
 

Figure 4.7: A Screenshot of Search option of Emergency ride Figure 4.8: A Screenshot of Find Driver of Emergency ride

Rider enter your destination place and select your emergency vehicle for example- CNG or Ambulance. Now rider click the find driver button. In this time user can show the pickup your driver information. Figure 4.9 shown the diver information.

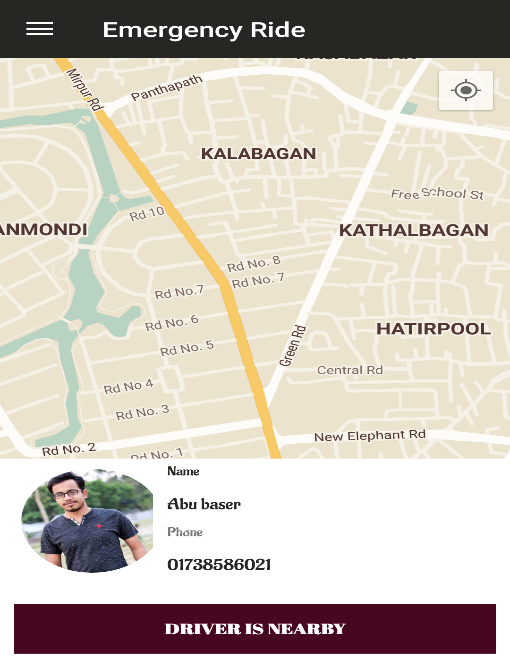


Figure 4.9: A Screenshot of driver Information of Emergency ride

In figure 4.10 shown the driver location on using google map. In this time driver is ready for riding request. After a successful login, the user can see this profile. If the device already has a google account connected to it. It will login using it automatically. Figure 4.11 shown the driver menu options of EMERGENCY ride application.

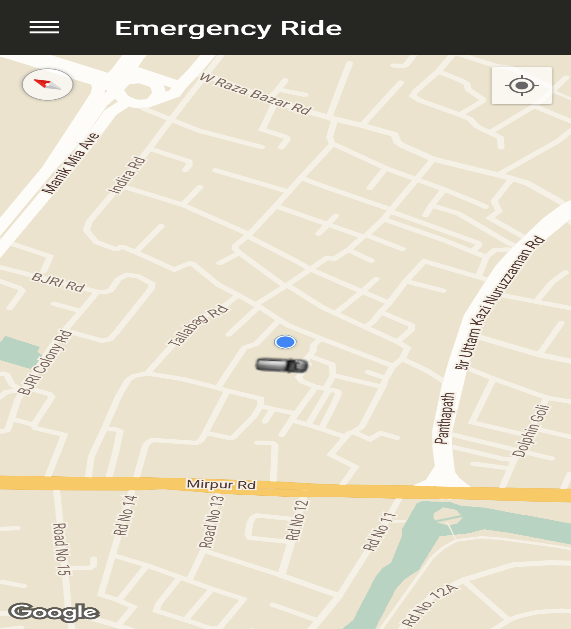
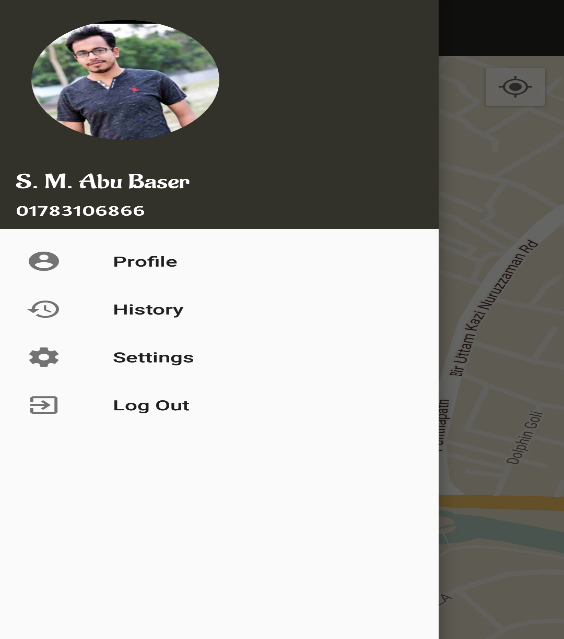
 

Fig 4.11: Google map view of Emergency ride Fig 4.12: Driver Menu option of Emergency ride

In figure 4.13 shown the driver profile information. Driver profile update or editing any time. If driver can drive all vehicle in different time. Figure 4.14 shown the editing driver profile.

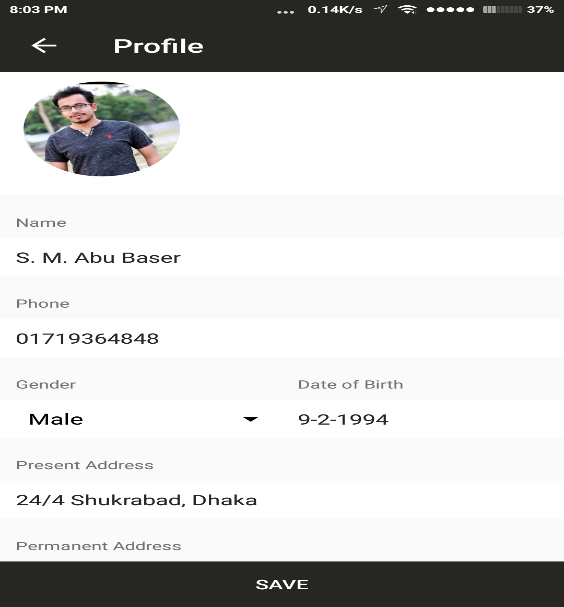
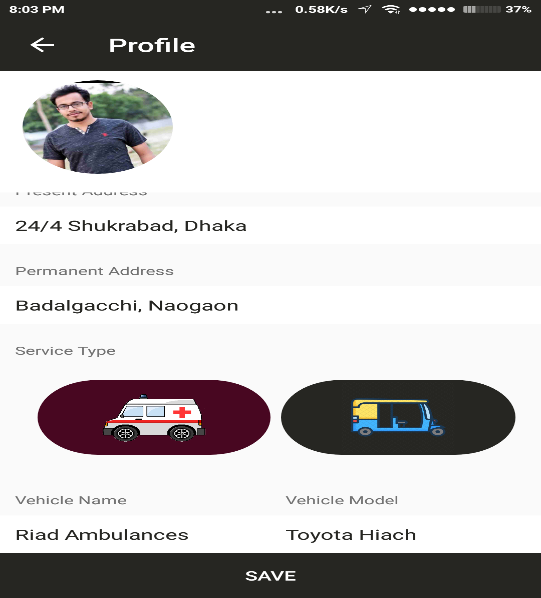
 

Fig 4.13: A Screenshot of driver profile Fig 4.14: A Screenshot of Editing driver profile

History is the most important part of the rider and driver. All rides information stored in history. Figure 4.15 shown the driver history of EMERGENCY ride. Some apps information available in about option. Figure 4.16 is about of EMERGENCY ride application.

Fig 4.15: A Screenshot of driver history of Emergency ride Fig 4.16: A Screenshot of About of Emergency ride

Rider select your destination and choice your vehicle option. Finally, rider find your driver. In this time driver receive the request. Figure 4.17 shown the receive request of EMERGENCY ride application. Driver receive rider request and click the accepted button then driver is ready for calling. Now click the calling button and contact to rider. Figure 4.18 shown the accepted and calling request of EMERGENCY ride.

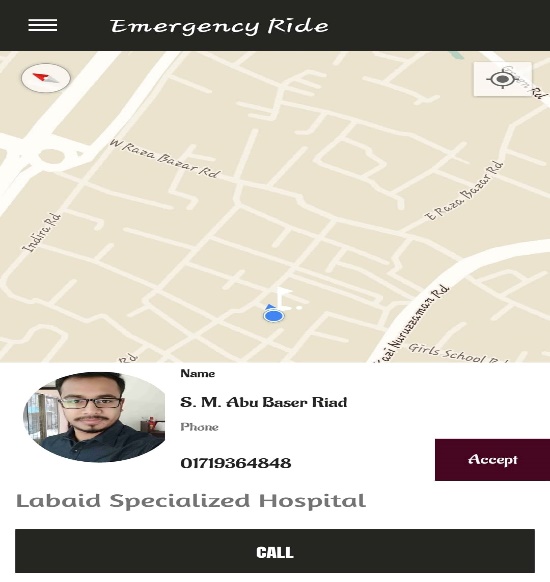
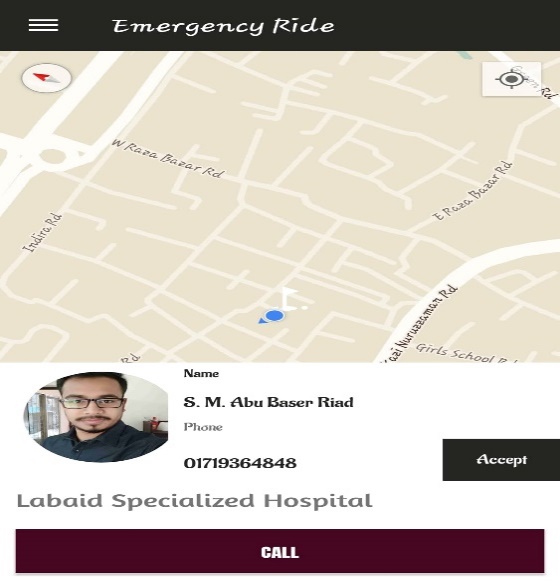
 

Figure 4.17: A request for acceptance of Emergency ride Figure 4.18: Accept and calling request of Emergency ride

Driver contact to rider and driver going to rider location. Now pick your client and click the pick your client button. Figure 4.19 shown the pick your client of EMERGENCY ride. Now driver pick your client and safely reach rider destination. Driver click the ride completed button and complete your ride. Figure 4.20 shown the ride completed of EMERGENCY ride.

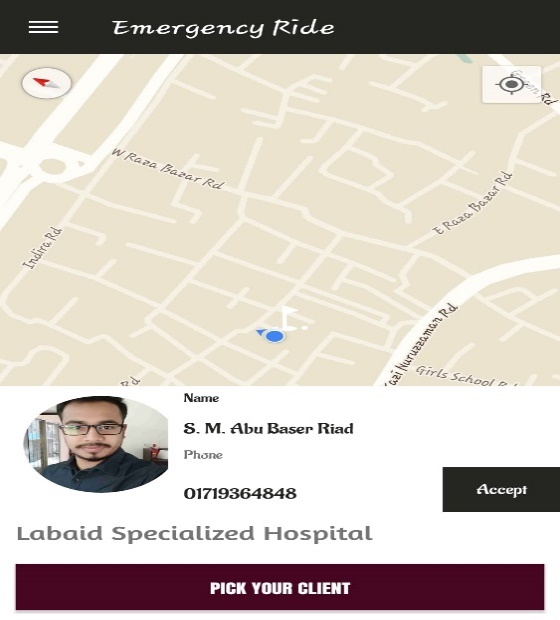
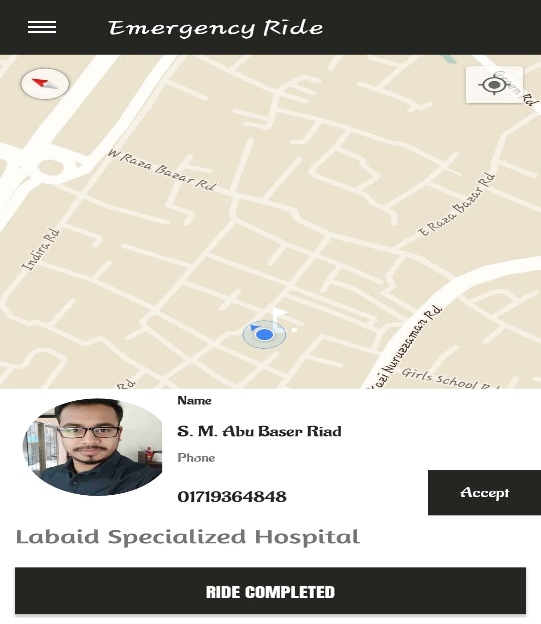
 

Fig 4.19: A Screenshot of Pick your client f Emergency ride Fig 4.20: A Screenshot of Ride completed of Emergency ride

Now your drive has been successfully completed and shown your total cost. Now driver click done button and ready for riding request again. Figure 4.21 shown the total fair.

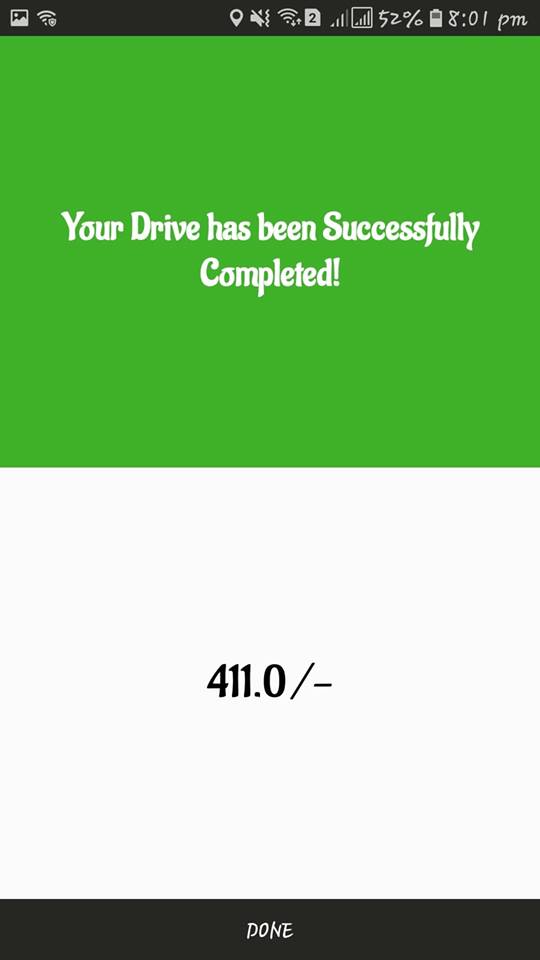


Figure 4.21: A Screenshot of Total fair of Emergency ride

**4.2 Back-end Design**

The back-end design is also called the server side. It controls the behavior of the site upon and action is taken. The back-end is the core portion of an application form where everything is being controlled whereas the front-end is the visual representation of the back-end. Back-end is the most important part. So, the security, structure, and content of it is very much important. We used Firebase Real-time Database this application.

We used some of googles API distribution packages for our mobile application back-end designing such as map API. We also used firebase for email authentication for the platforms.

The users email address is authenticated using firebase, the data user seeks or inputs is beings carried to the server for further processing is being done using java.

**CHAPTER 5**

**IMPLEMENTATION AND TESTING**

**5.1 Implementation of Database**

We have used firebase server as data manager to store our android application data. The Real-time Database provides a flexible, expression-based rules language, called Firebase Real-time Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it [10].

The Real-time Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Real-time Database API is designed to only allow operations that can be executed quickly. This enables to build a great Real-time experience that can serve millions of users without compromising on responsiveness.

**5.2 Implementation of Front-end Design**

It is really a good challenge to develop a gorgeous front-end design which will be attracted to user. Because, for developing a design for android devices, all the time developer has to consider the display dimension of android device, it is very tough work to balance the design with android display size. Sometimes it can’t fix with the display for many reasons. So, a developer needs to check several times of his/her android application by building or running within an android device. For interactive design we always try to think as a simple and easier in user interface design for creating user attraction to our android application. We also try some materials and tools design for making and creating attraction to the user. On the other hand, the user can enjoy to using a very simple and easier interface.

Front-end design is the first impression of a user. We know that people are not easily forget the first impression. So, we have tried our best to make the front-end design simple, attractive and user friendly. So, it was really a good challenge to us when we are designed our android application’s user interface.

There are some factors of implementing the front-end design are given below

* There will be two types of users like rider and driver.
* Every types of user must be registered by filling up the required information fields.
* User can login using their registered email and password.

**5.3 Implementation of Java and XML Code**

We use android studio for our application user interface design with xml file and for java code to connect with the xml file, Firebase Real-time Database server and apache server.

We also android studio for our application design and connection with internet. Because, in android studio, is supported many types of languages. In java code, we use some class, method and process for connection view. We use JSON parsing method for parsing data from online to user device. That means major work was done in java code. In xml code, we take some text view, button and image view for our application. By using in xml coding, we design our application.

**5.4 Test Implementation**

Table 5.1: Test case for EMERGENCY ride

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Input** | **Expected Outcome** | **Obtained Outcome** | **Result** | **Tested on** |
| 1.  Install application | Tested on various android version   * Jelly bean   (4.1-4.3.1)   * KitKat   (4.4-4.4.4)   * Lollipop   (5.5-5.0.2)   * Marshmallow (6.0) * Nougat   (7.0-7.1) | Successfully install all those various | Install successfully | Passed | 02-11-2018 |
| 2.  Login | Login via any smart phone device and tablet | Successfully login | Successfully login | Passed | 02-11-2018 |
| 3.  Registration or sign up | Registration any smart phone device and tablet | Successfully registration | Successfully registration | Passed | 02-11-2018 |
| 4.  Email | Blank or incorrect email | To warn that correct email must be entered | Showed the warning | Passed | 02-11-2018 |
| 5.  Password | Blank or incorrect password | To warn that correct password must be entered. | Showed the warning | Passed | 02-11-2018 |
| 6.  Picture Update | Input profile picture | To add profile picture in user profile | Update profile picture successfully | Passed | 02-11-2018 |
| 7.  Profile Update for rider | Input full name, phone number, gender and birthdate | To update rider profile information | Update rider profile information successfully | Passed | 02-11-2018 |
| 8.  Profile update for driver | Input full name, phone number, gender, present address, permanent address, vehicle type, vehicle model and vehicle number. | To updated driver profile information | Update driver profile information successfully | Passed | 02-11-2018 |
| 9.  Search destination | Input destination name | To show the similar other destination name and selected one | Showed the destination name | Passed | 02-11-2018 |
| 10.  History | Click on the history button | Show the all riding information | Showed the all riding information | Passed | 02-11-2018 |
| 11.  Setting | Click on the setting button | Show the profile information | Show profile | Passed | 02-11-2018 |
| 12.  Requested rider | Rider get requested to any smart phone devices | Show your location, search destination name and enter search driver | Accepted requested | Passed | 02-11-2018 |
| 13.  Accepted requested | Driver accepted requested to any smart phone devices or tablet | Show the rider requested | Successfully accepted | Passed | 02-11-2018 |
| 14.  Logout | Click on the logout button | To logout from that account | Logged out successfully | Passed | 02-11-2018 |

**5.5 Test Results and Reports**

Test report is needed to reflect the result of testing the application in a formal way, which gives an opportunity to estimate the result of testing quickly. It is a document that records data obtained from a determine, experiment in an organization manner, describe the environmental or operating systems conditions and shows the comparison of test results with objectives, which are so important for any types of application.

In table 5.4, we show the test case, test input, expected outcome, obtained outcome and finally we find our expected results for our application. The test result was quite successful. The user satisfies to using our application. Our expectation will be that user can easily use and understand our application as a better user interface.

**CHAPTER 6**

**CONCLUSION AND FUTURE SCOPE**

**6.1 Discussion and Conclusion**

Our android based mobile application has been successfully implemented. For implementing our application, we tested the application with many types of smart phones, we saw that our application worked properly and it also gave the required data from database server, map navigation was successful.

The design is very user friendly. The user interface is so simple and not looking like as a complex view. User can easily handle our application. We tried our best to complete all requirements of our application. We hope that people will use our application and will get proper service.

**6.2 Limitations of our Application**

As like every application, our application has also some limitations. We will overcome those limitations in future. Here, we want to mention that some of the main limitations of our application are given bellow:

* Proper not uses in Google map API
* Not uses valid email address
* Payment type is only hand cash

**6.3 Scope for Future Development**

We try our best for developing our application in present time. If we find any scope for developing in future, we want to develop and change something of our application. Our future developments scopes are given below:

* In future, we will try to develop this application another vehicle adding. For, example- bicycle.
* We will try to develop this application for all Types payment option, for example bKash, DBBL, credit card and debit card.

**APPENDIX**

**Java Code:**

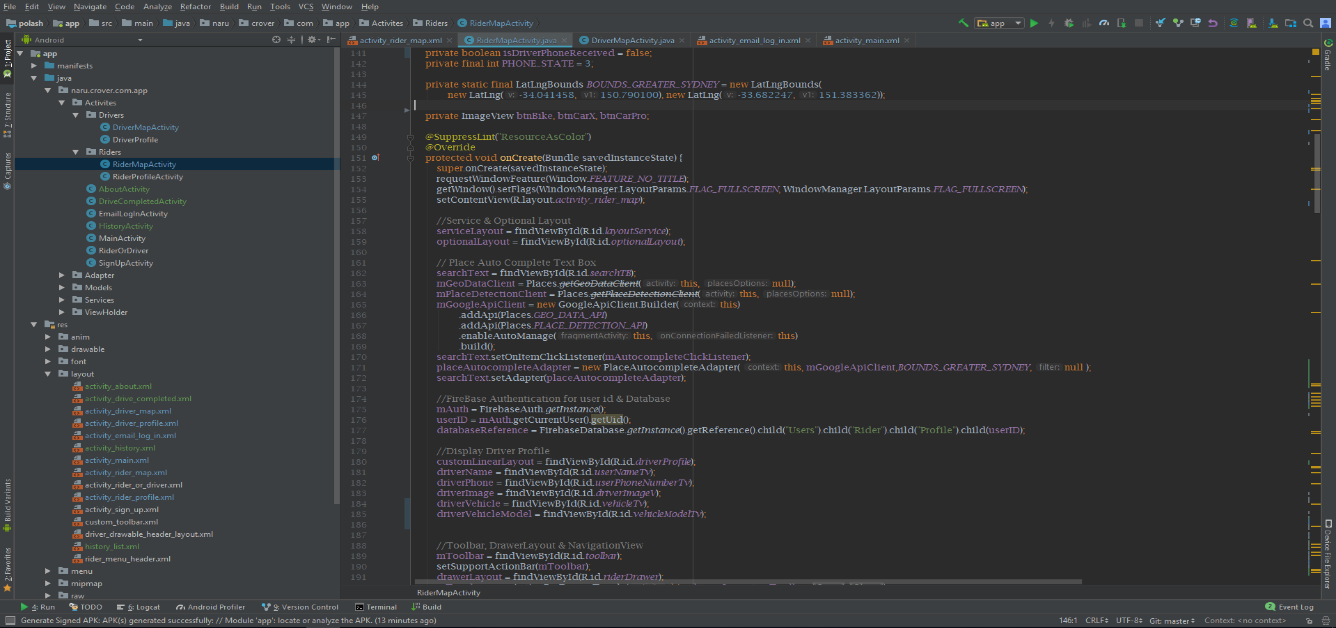


Figure 6.1: A Screenshot of Java Code of EMERGENCY Ride

**XML Code:**

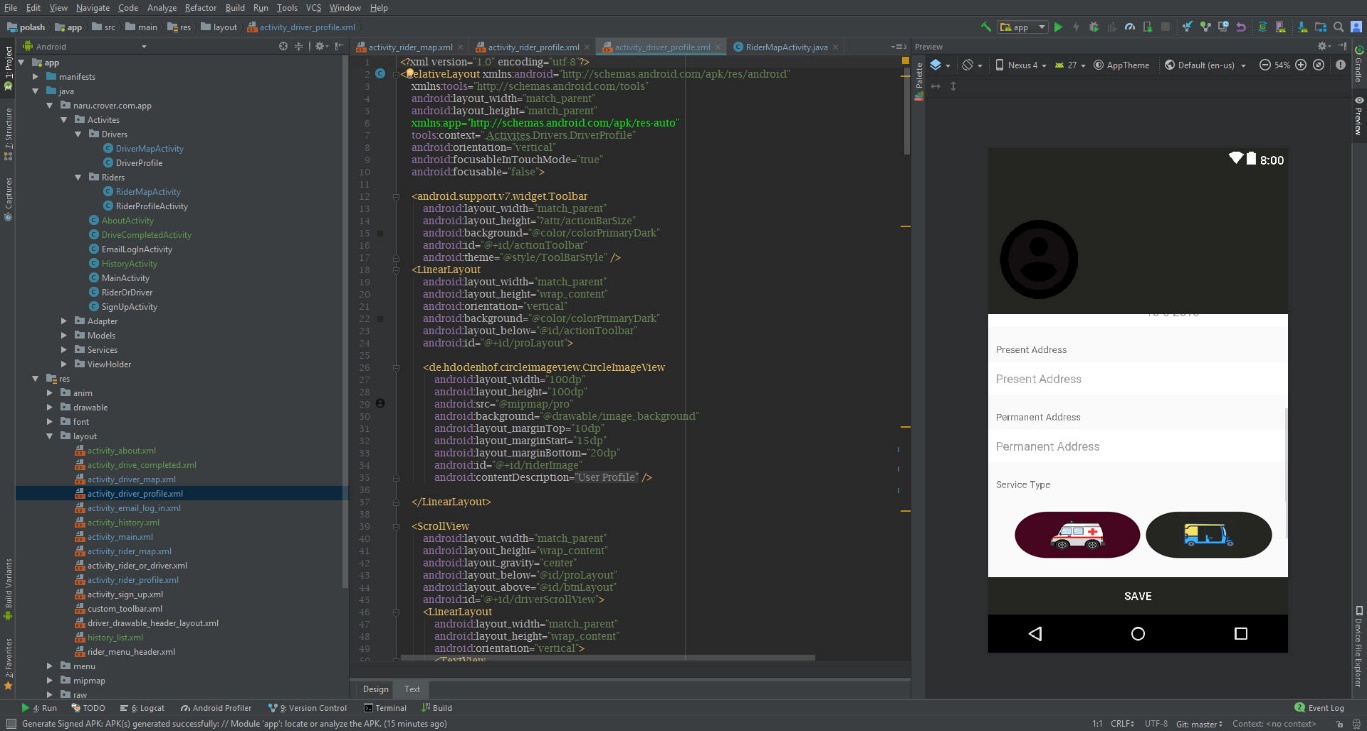


Figure 6.2: A Screenshot of XML Code of EMERGENCY Ride

**REFERENCE**

[1] “History of Uber application” <https://en.wikipedia.org/wiki/Uber> [accessed May 24, 2018].

[2] “Uber application” <https://play.google.com/store/apps/details?id=com.ubercab&hl=en> [accessed May 24, 2018].

[3] “PATHAO application” <https://play.google.com/store/apps/details?id=com.pathao.user&hl=en> [accessed May 24, 2018].

[4] “TAXIWALA Mama Application” <https://www.google.com/search?q=taxiwala+mama+application&source=lnms&tbm=isch&sa=X&ved=0ahUKEwivsKCBx8fdAhVVWH0KHYIVBHoQ_AUIDigB&biw=1366&bih=626#imgrc=mTlcRl8tnu4FvM>: [accessed May 24, 2018].

[5] “SHOHOJ Rides Application” <https://www.google.com/search?q=shohoj+ride+app&tbm=isch&source=lnms&sa=X&ved=0ahUKEwjX6oDcx8fdAhURdysKHa5MD88Q_AUIDCgD&biw=1366&bih=626&dpr=1> [accessed May 24, 2018].

[6] “Comparative studies of PATHAO Rides”

<http://www.macrothink.org/journal/index.php/csbm/article/download/11906/9526> [accessed June 1, 2018].

[7] “Comparative pricing chart for Uber and PATHAO Rides” <https://www.researchgate.net/figure/Comparative-pricing-chart-for-Pathao-Rides-amount-in-BDT-Initial-Fee-Cost-per-minute_tbl3_320465767> [accessed July 12, 2018]

[8] “Business process modelling” <https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation> [accessed July 12, 2018]

[9] “Logical data model” <https://en.wikipedia.org/wiki/Logical_data_model> [accessed July 13, 2018]

[10] “Implementation of database” <https://firebase.google.com/docs/database/> [accessed July 13, 2018]

**Plagiarism Report:**