

**TUTORS MAP: AN ANDROID APP FOR PRIVATE TUTOR**

**BY**

**NAZIFA TABASSUM**

**ID: 142-15-3634**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of  
Bachelor of Science in Computer Science and Engineering

Supervised By

**MS. NAZMUN NESSA MOON**

Assistant professor

Department of Computer Science and Engineering

Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

**DHAKA, BANGLADESH**

**APRIL 2018**

## **APPROVAL**

This Project/internship titled “Tutors Map”, submitted by Nazifa Tabassum, ID No: 142-15-3634 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 7<sup>th</sup> May, 2018.

### **BOARD OF EXAMINERS**

---

**Dr. Syed Akhter Hossain**  
**Professor and Head**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University



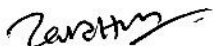
**Chairman**

---

**Dr. Sheak Rashed Haider Noori**

**Associate Professor and Associate Head**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University



**Internal Examiner**

---

**Md. Zahid Hasan**

**Assistant Professor**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University



**Internal Examiner**

---

**Dr. Mohammad Shorif Uddin**

**Professor**

Department of Computer Science and Engineering  
Jahangirnagar University

**External Examiner**

## DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Ms. Nazmun Nessa Moon, Assistant professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

**Supervised by:**



**Ms. Nazmun Nessa Moon**

**Assistant professor**

Department of CSE

Daffodil International University

**Submitted by:**

\_\_\_\_\_  
**Nazifa Tabassum**

ID: 142-15-3634

Department of CSE

Daffodil International University

## ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Ms. Nazmun Nessa Moon, Assistant professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “Android” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Syed Akhter Hossain, Professor & Head, Department of CSE**, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, 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 parents.

## Abstract

Education is the way to gain knowledge. The most common forms of education result from years of schooling that incorporates studies of a variety of subjects. Taking tuition is another way to gain some knowledge in many academic subjects. This is also a good way to earn some money. By tuition one individual can earn some money and another can acquire some specific knowledge. But in our country, there is business purpose everywhere. In tuition system the business is done by a system called media. This media system sometimes ruins the quality of education. This project presents an app that named **“Tutors Map: An Android app for Private Tutor”** can help anyone to find a tuition or a tutor. This app can prevent media business in tuition system. In this app anyone can create an account to add their desired subject for the purpose of getting tuition or giving tuition. In addition, the parents can be contacted in case of any need.

## Table of content

<b>Contents</b>	<b>Page</b>
Board of Examiners	i
Declaration	ii
Acknowledgement's	iii
Abstract	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii

## CHAPTER

<b>CHAPTER 1: INTRODUCTION</b>	<b>1-2</b>
1.1 Introduction	1
1.2 Motivation	1
1.3 Objective	1
1.4 Expected Outcome	2
1.4 Report Layout	2
<b>CHAPTER 2: BACKGROUND</b>	<b>3-7</b>
2.1 Introduction	3
2.2 Related Works	3
2.3 Comparative Studies	6
2.4 Scope of the Problem	6
2.5 Challenges	7
<b>CHAPTER 3: REQUIREMENT SPECIFICATION</b>	<b>8-19</b>
3.1 Business Process Modeling	8
3.2 Requirement Collection and Analysis	9

3.3	Use Case Modeling and Description	10
3.4	Logical Data Model	18
3.5	Design Requirements	19
<b>CHAPTER 4:</b>	<b>DESIGN SPECIFICATION</b>	<b>20-26</b>
4.1	Front-end Design	20
4.2	Back-end Design	23
4.3	Interaction Design and UX	26
4.4	Implementation Requirements	26
<b>CHAPTER 5:</b>	<b>IMPLEMENTATION AND TESTING</b>	<b>28-39</b>
5.1	Implementation of Database	28
5.2	Implementation of Front-end Design	29
5.3	Implementation of Interactions	37
5.4	Testing Implementation	37
5.5	Test Results and Reports	39
<b>CHAPTER 6:</b>	<b>CONCLUSION AND FUTURE SCOPE</b>	<b>40</b>
6.1	Discussion and Conclusion	40
6.2	Scope for Further Developments	40
<b>REFERENCES</b>		<b>41</b>
<b>APPENDIX</b>		
	Appendix A: Project Reflection	<b>42</b>
<b>Plagiarism</b>		
	Plagiarism Report	<b>43</b>

## LIST OF FIGURES

FIGURES	PAGE
---------	------

Figure 2.1: Tutor Finder	4
Figure 2.2: Tuition	5
Figure 3.1: BPM of Tutor Map	8
Figure 3.2: Use Case Diagram of Tutor	10
Figure 3.3: ER Diagram of Tutor Map	18
Figure 4.1 Tutor Map Starting Page Design in Android Studio	20
Figure 4.2: Tutor Map Starting Page XML Code in Android Studio	21
Figure 4.3: Tutor Map Starting Page Java Code in Android Studio	21
Figure 4.4: Tutor Map Registration Page Design in Android Studio	22
Figure 4.5: Tutor Map Registration Page XML Code in Android Studio	22
Figure 4.6: Tutor Map Registration Page Java Code in Android Studio	23
Figure 4.7: Firebase Authentication Page	24
Figure 4.8: Firebase Database Page 1	24
Figure 4.9: Firebase Database Page 2	25
Figure 4.10: Firebase Database Page 3	25
Figure 5.1: Firebase and Java	28
Figure 5.2: Tutor Map Starting Page	29
Figure 5.3: Registration Selection	30
Figure 5.4: Registration Page for Student and Teacher	31
Figure 5.5: Sign in Page	32
Figure 5.6: Home Page for Student and Teacher	33
Figure 5.7: List of Teacher and Student	34
Figure 5.8: Profile Edit Option	35
Figure 5.9: Searching Option	36



## List of Tables

<b>Tables</b>	<b>Page</b>
Table 5.1: Testing Implementation	37

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

These days the most important thing in our life is education. We all need some extra care in education. Or in other perspective people who are willing to be a tutor needs good tuition. So, for both of them one great system is needed. Now there are some Facebook group where you can find some tuition. But in the end, it's not so helpful. So, I propose to make this private tutor app, where tutor and parents can easily find what they are wishing for.

### 1.2 Motivation

Now a day everyone is focusing on earning some money. For student, it's really needed to earn some extra cash for their pocket money or for other necessary things or gaining knowledge. So, I wish to make an application for tutor and everyone who are willing to find a tutor. I believe that this application will be very helpful for our educational system. This application will also help to prevent some problems. Like, tutor will not have to pay the media for getting a tuition. So, this will save their hardly earned money.

### 1.3 Objective

- To find proper tutor for every subject
- To interact between parents and tutor
- To search student by location wise or other preferences
- To give every kind of learning option
- To stop paying 3rd party

- To settle salary

## **1.4 Expected Outcome**

We will get an app where anyone can easily find their demandable tutor and tutor will easily find tuition as well. There will be details about tutors. About their experience and education. Tutor will see the details about the students as well.

## **1.5 Report Layout**

### **Chapter 1: Introduction**

In this chapter we have discussed about the introduction, motivation, objectives and 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, comparison to other candidate systems, the scope of the problem and challenges of the project.

### **Chapter 3: Requirement Specification**

This chapter is all about the requirements like business process modeling, the requirement collection and analysis, the 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. Front-end design, back-end design, interaction design and UX and the implementation requirements

### **Chapter 5: Implementation and Testing**

This chapter contains the implementation of database, front-end designs, interactions, test implementation and the test results of the project.

### **Chapter 6: Conclusion and Future Scope**

We discussed about the conclusion and the scope for further developments which pretty much derive about the project

# CHAPTER 02

## BACKGROUND

### 2.1 Introduction

An android app is basically a software-based system and to develop the system it requires android studio IDE or ECLIPS ide. I have used android studio. In android studio one can code in JAVA, KOTLIN and C++ as the main coding language [6]. And for design purpose one can use XML language to design the GUI.

I have used JAVA as the main coding language to make the interaction working. And also used XML language to design a GUI for my app. There are several systems to use as database in android database like XAMPP, ORACLE, SQL-lite or firebase (online data base).

I have used firebase to use this app without any problem. Firebase is a googles product [4]. Which gives a running server for database.

### 2.2 Related Works

There are two apps related to my app. Which has almost the same motive as I had intended to do. When I planned to make the app, there was no app like this. But now there is two app. But there are some significant differences from their app.

Fist app is **Tutor Finder** [7], in figure 2.1.

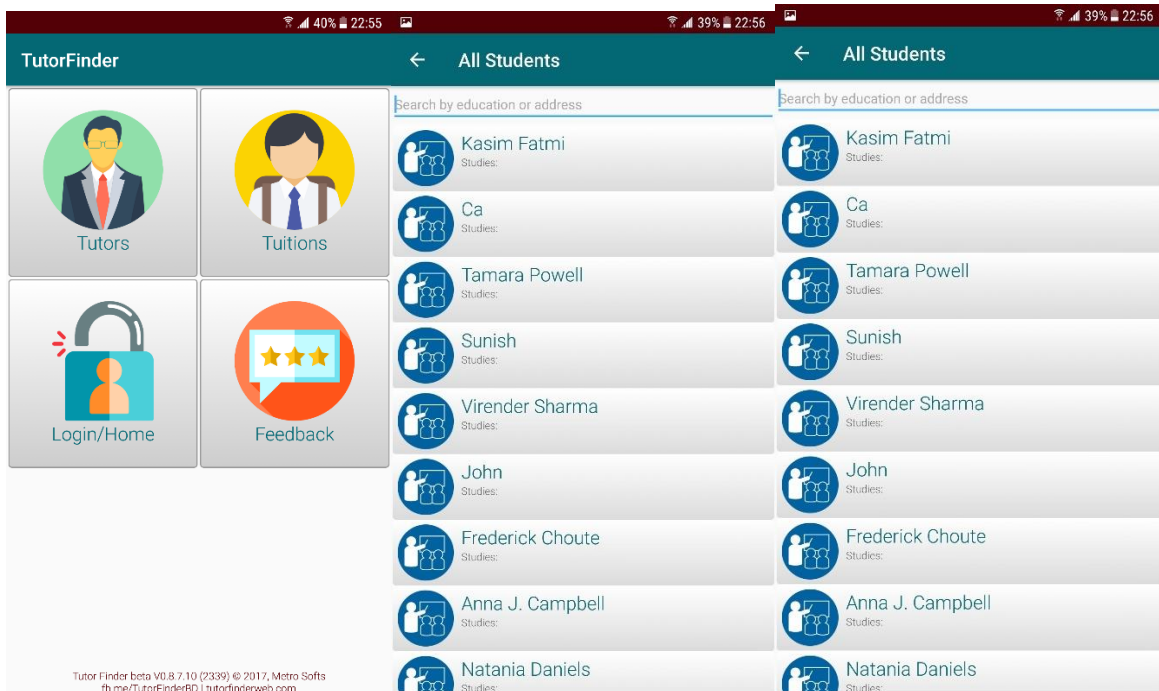


Figure 2.1: Tutor Finder

**Features:**

- Single account to find tuitions and tutors.
- Can search with location and subjects
- Don't need to login to find tutors or tuitions

**Problems:**

- Security issues
- No defined way to know if a user is a teacher or a student.
- No communications between parent and tutor
- No communication from the app
- Cannot use it in offline

2<sup>nd</sup> app is **Tuition** [8], in fig no 2.2:

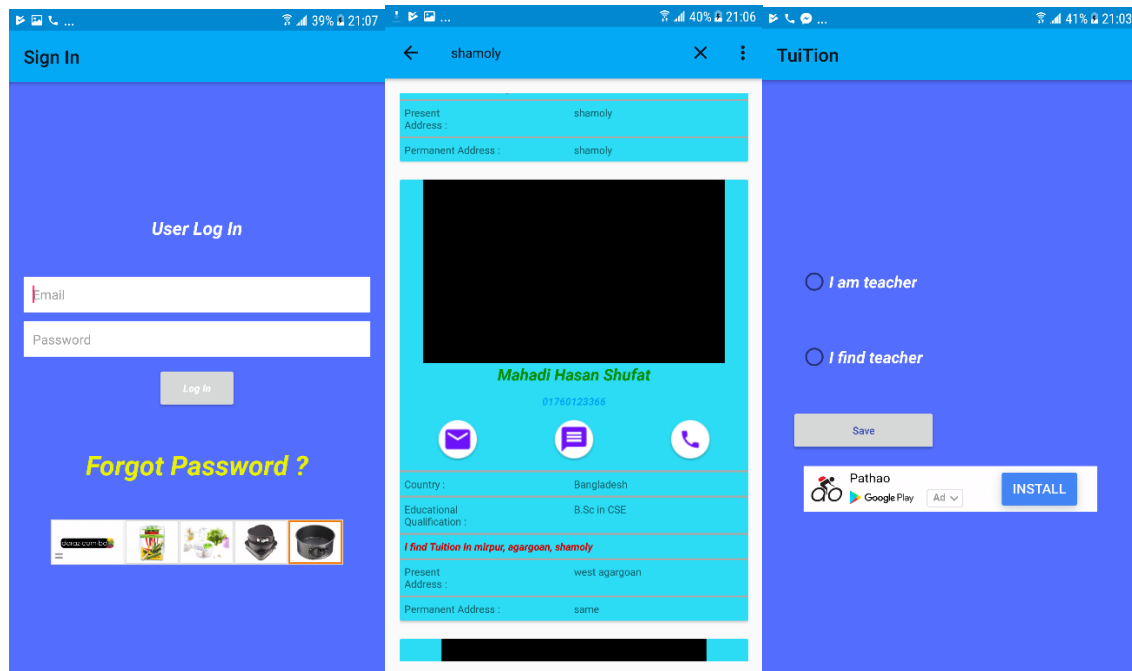


Figure 2.2: Tuition

### Features:

- Can define a user
- Can send message from the app using phone sim

### Problems:

- Security issues
- No communication with the parent
- Only area wise search option.
- Cannot use it under android version 6
- Cannot use it in offline

### **2.3 Comparative Studies**

There are many common things in our app. But there is a lot of significant differences also. And the problems of those two app can be solved by this app. Or in other word it can surpass the limitations of those two app. So, the differences from Tutor finder and Tuitions are provided down below,

- One user can use Tutors Map in offline mode. But those two app cannot be used in offline mode.
- One can run this app in any android version, where both of those app has limitations running on lower version of android.
- Tutor map has an option to contact with the parent of the students. Other apps don't have that kind of options.
- One can define that the user is a tutor or a student, where the Tutor Finder app fails to do that.
- Both app has security issues. Tutor map uses firebase authentication system which is made by google. So, it is almost impossible to break.

### **2.4 Scope of the Problem**

This application is based on android. Now android has become one of the mostly used operating system for a mobile system. So, there are many developers who are working for making android apps. And this media system was not only noticed by me. This has been noticed by many people. So, there are many people trying to solve this problem. But most of the application does not fulfill all the requirements of the needs. There will be always a limitation in the technology world.

Most of the current app is not user-friendly or have some limitations. That's why I planned to develop this tutor map app where a tutor and a student can interact for tuitions or educational purpose. The media system can be nullified by using this app.

## 2.5 Challenges

This application was not easy for me as I was a beginner in android developer world. So, there was challenges in every step of my work. Some of the challenges are like,

- I was not very good in programming. So, it took a lot of time to learn JAVA and start this project.
- I am using a new kind of database which is named “Firebase”. This system has a less amount of knowledge to everyone. So, it was a lot pain to understand the system.
- There was already two app with similar idea. So, I had to come up with ideas that those app didn't have.



## CHAPTER 3

### REQUIREMENT SPECIFICATION

#### 3.1 Business Process Modelling

BPM or Business Process Modeling is a process of constructing a structural view of a system or process. It includes some process, starts and symbol, condition as like a flow chart. In this process we can see the student can send request to a teacher to show that the student is interested to take tuition from that tutor. And again, the same system goes for the tutor, if the tutor likes a student then the tutor can send request to the student to show that the tutor is interested to teach that student.

BPM model show in figure 3.1.

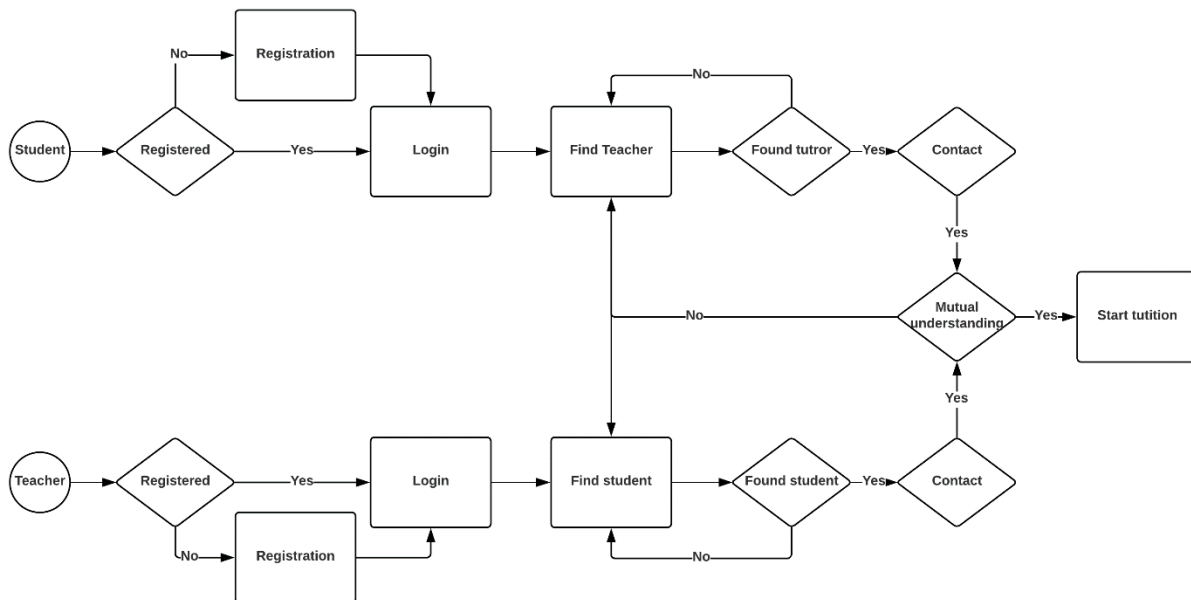


Figure 3.1: BPM of Tutor Map

### 3.2 Requirements Collection and Analysis

An android app requires many things to fulfil the desires of a system. Requirement could be software wise or hardware wise. My application required a lot of time and logics to finish.

So, the usual **requirements** that needed to full fill my applications were,

- Coding languages
  - JAVA
  - XML
  - Firebase system
- Software IDE
  - Android studio
- Hardware
  - Any android phone

#### Analysis

After a lot of analyses of the current system of tuition I wanted to make this app for the help of others. The current system is full of media business. When the education system is touched by business it ruins the quality of education. Sometimes people with no good knowledge can become a tutor. And people with needs may be not able to find any good tuition. This media system takes a first time of 100% or in-between 50%-100% of the first month payment from a tuition. They are like a broker in tuition system. After discovering this kind of problem, I decided to make this app for making it easy to find tuition or getting tutor.

### 3.3 Use Case Modelling and Description

A use case model is a graphic description of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify and organizing system requirements. Use case model show in figure 3.2.

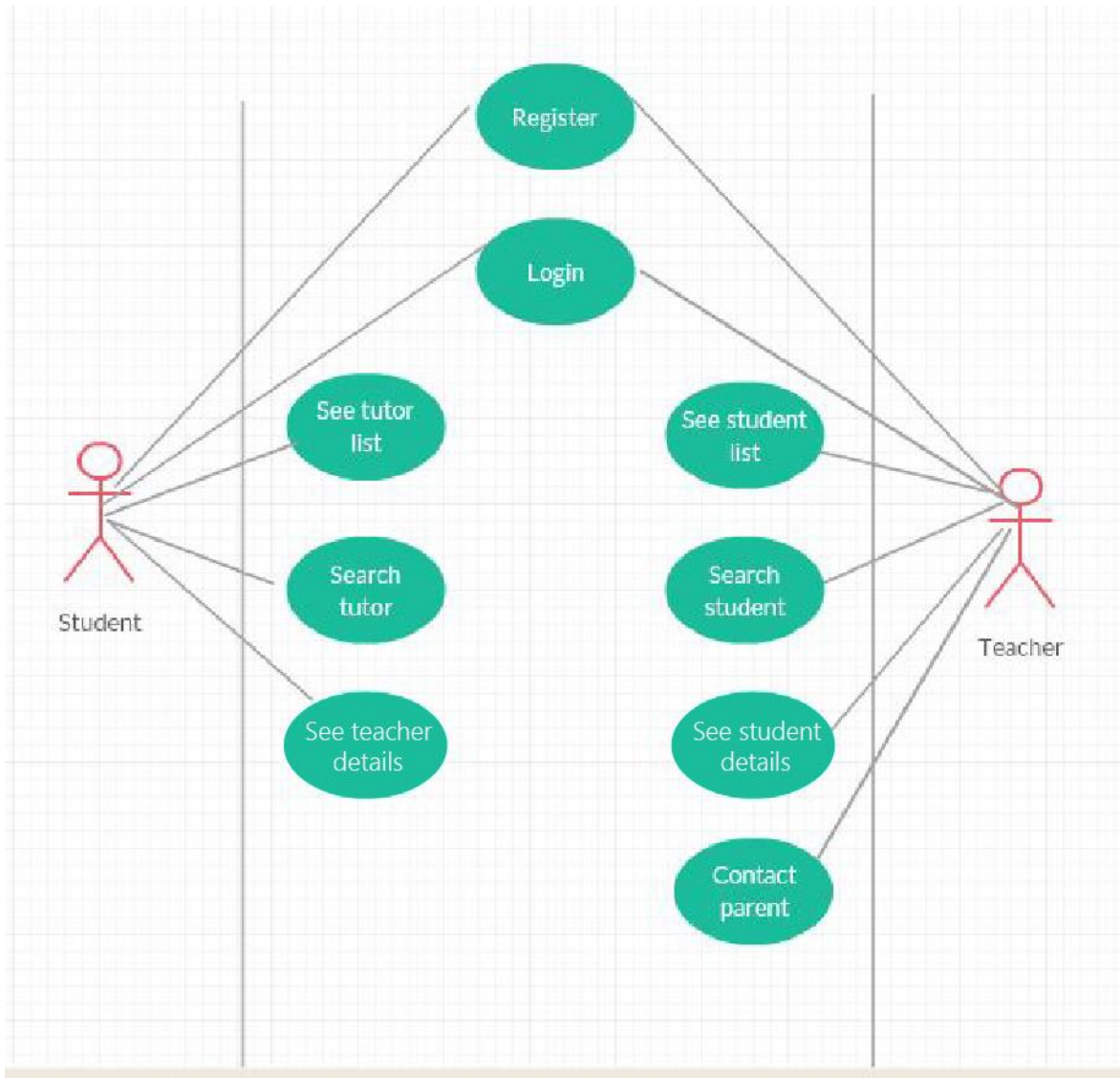


Figure 3.2: Use case diagram of Tutor

## **Use Case Description of Tutor Map**

**Use Case:** Student Register

**Actor:** Student

**Pre-condition:** NULL

### **Primary Path**

- Choose sign up as student
- Fill the data fields
- Click the registration button
- Take to the home page

### **Alternate Path**

- Didn't choose sign up as student
- Data field is empty
- Connection error

### **Post-condition**

- If registration successful
- If home page is loaded

**Use Case:** Student login

**Actor:** Student

**Pre-condition:** Registration

**Primary Path**

- Choose sign in
- Fill the data fields
- Click the log in button
- Take to the home page

**Alternate Path**

- Didn't choose sign in page
- Data field is empty
- Connection error

**Post-condition**

- If log in successful
- If home page is loaded

**Use Case:** See tutor list

**Actor:** Student

**Pre-condition:** Login

### **Primary Path**

- Choose list tab

### **Alternate Path**

- Didn't choose list tab
- Connection error

### **Post-condition**

- If list was loaded

**Use Case:** Search tutor

**Actor:** Student

**Pre-condition:** Login

### **Primary Path**

- Choose search button
- Type address or subject or name

### **Alternate Path**

- Didn't choose search button
- Didn't typed address or subject or name
- Typed wrong type of search
- Connection error

**Post-condition**

- If search is successful

**Use Case:** See teacher detail

**Actor:** Student

**Pre-condition:** Login

**Primary Path**

- Find tutor
- Click on the list

**Alternate Path**

- Didn't found tutor
- Didn't click on the list

**Use Case:** Tutor Register

**Actor:** Teacher

**Pre-condition:** NULL

### **Primary Path**

- Choose sign up as teacher
- Fill the data fields
- Click the registration button
- Take to the home page

### **Alternate Path**

- Didn't choose sign up as teacher
- Data field is empty
- Connection error

### **Post-condition**

- If registration successful
- If home page is loaded

**Use Case:** Teacher login

**Actor:** Teacher

**Pre-condition:** Registration

### **Primary Path**

- Choose sign in
- Fill the data fields
- Click the log in button
- Take to the home page



**Alternate Path**

- Didn't choose sign in page
- Data field is empty
- Connection error

**Post-condition**

- If log in successful
- If home page is loaded

**Use Case:** See student list

**Actor:** Teacher

**Pre-condition:** Login

**Primary Path**

- Choose list tab

**Alternate Path**

- Didn't choose list tab
- Connection error

**Post-condition**

- If list was loaded

**Use Case:** Search student

**Actor:** Teacher

**Pre-condition:** Login

**Primary Path**

- Choose search button
- Type address or subject or name

**Alternate Path**

- Didn't choose search button
- Didn't typed address or subject or name
- Typed wrong type of search
- Connection error

**Post-condition**

- If search is successful

**Use Case:** See student details

**Actor:** Teacher

**Pre-condition:** Login

**Primary Path**

- Find student
- Click on the list

## Alternate Path

- Didn't found student
- Didn't click on the list

### 3.4 Logical Data Model

Logical Data Modelling is a process used to define and analyse requirement needed to support the business processes within the scope of corresponding information systems in organizations. The Entity-Relationship model or Entity-Relationship diagram(ERD) is a logical data models, it includes the entity, attributes, table and relationships.

Logical data model is giving below in figure 3.3.

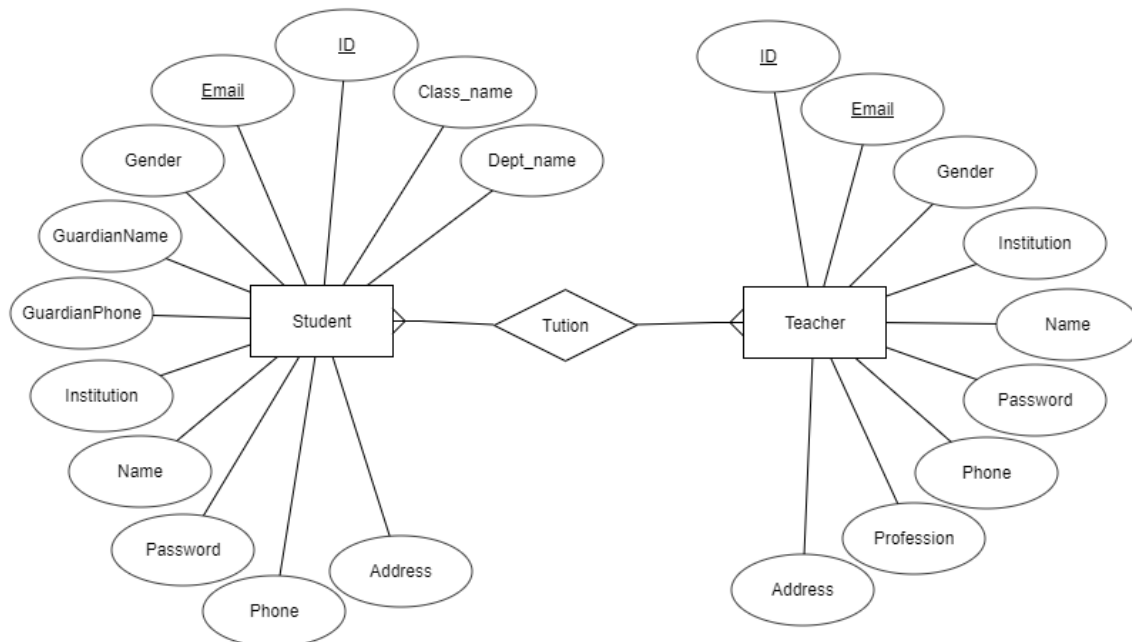


Figure 3.3: ER Diagram of Tutor Map

## **Description of Logical Data Model**

There are two users in this system. One is teacher and another one is student. Teacher can send request to the student to give tuition. And student can send request to get tuition. A teacher can give tuition to many students. And a student can take many tutors as well.

### **3.5 Design Requirements**

The following goals were kept in mind while designing the system:

#### **Make System Simple and Flexible for Users**

The system users are able to have a great amount of control over their purpose in achieving objectives.

#### **Make the System Compatible**

It should be fit in the total system, future maintenance and enhancement must less.

#### **Efficiency**

It is the most important. The system should run and work minimum hardware of software resource available without delaying.

# CHAPTER 4

## DESIGN SPECIFICATION

### 4.1 Front-End Design

In any application the front-end design is the visual part of an application. By which the user can interact with the app. In the sense of a user the front-end design is the main thing [1]. In the sense of the coder it means the languages put together to set the UX of an app.

For a convenient user experience, it is usually needed to be user-friendly and understandable by the users. There, while developing the app I tried to make it user-friendly so that the user can easily use it.

I have used XML (mark-up language) to make the user interface. Tutor map has many pages. So, there were a lot of designing thing for login page, registration page, and search, list and profile pages for both the student and the teacher. Screenshot given in figure 4.1.

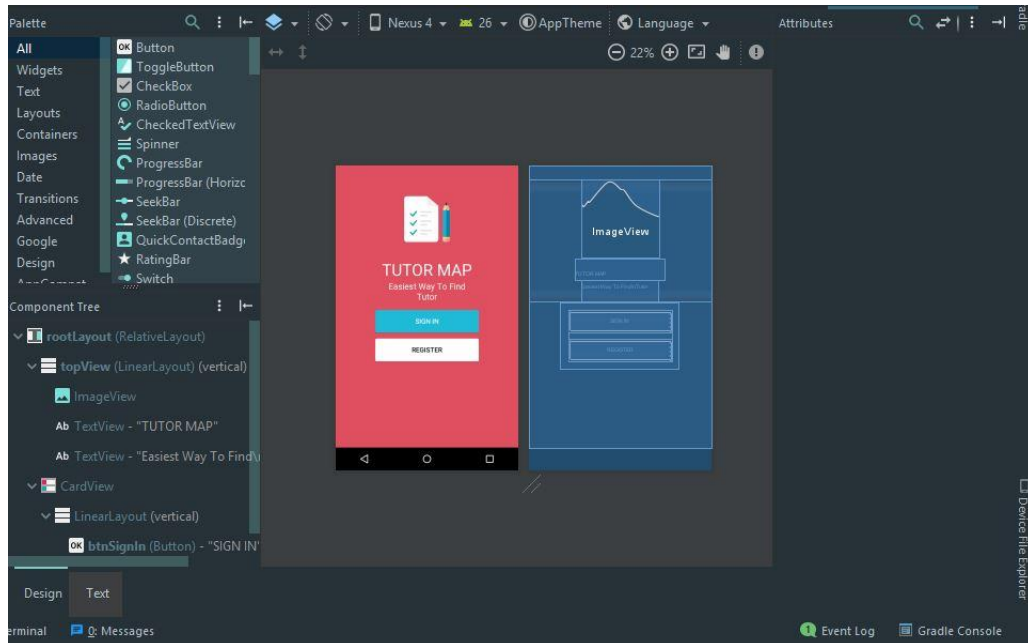
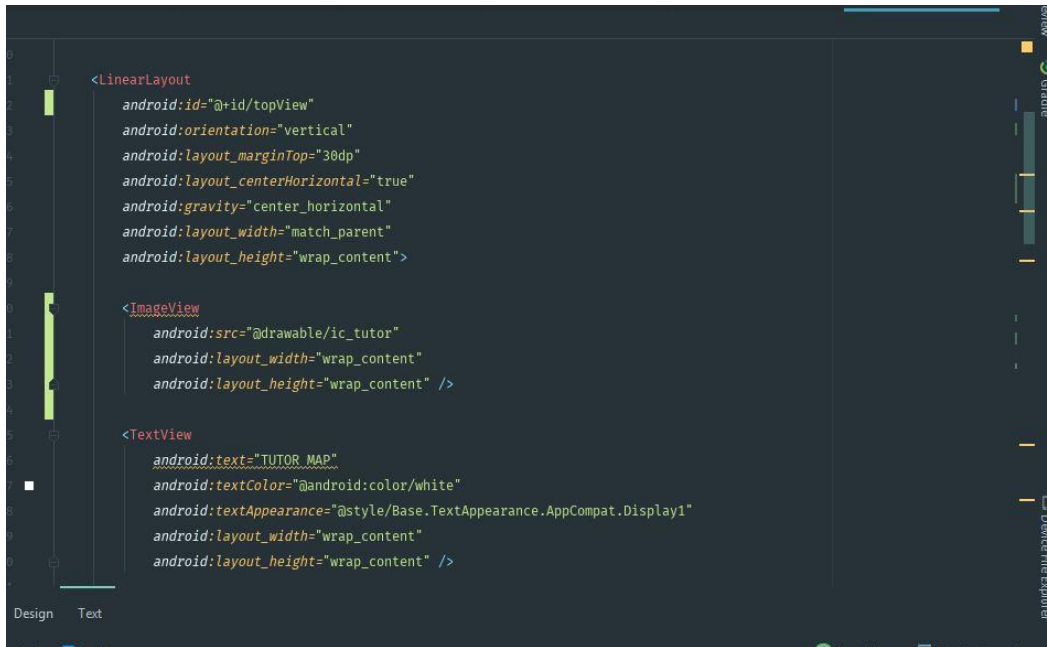


Figure 4.1 Tutor Map Starting Page Design in Android Studio

Here is the coding part that shows how I have used XML language to provide this user interface. Screenshot is given in figure 4.2.



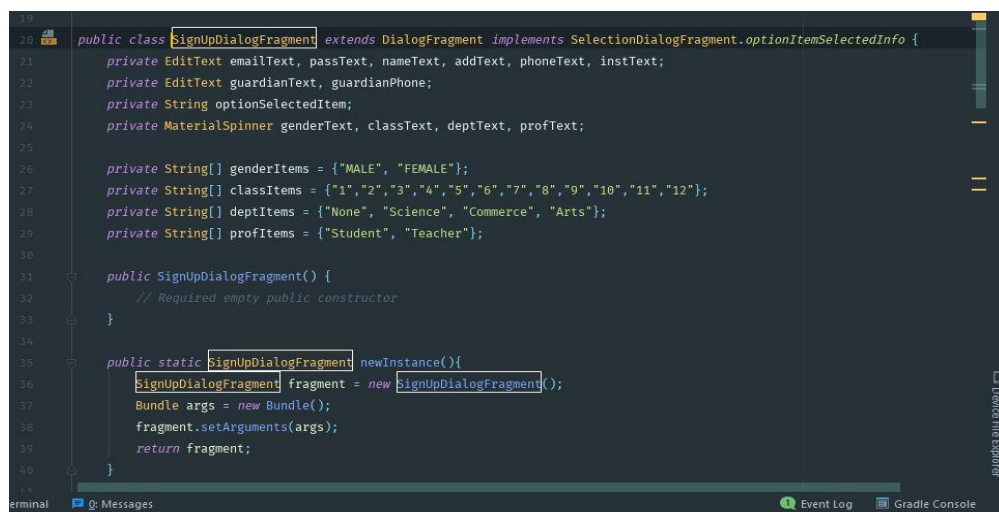
```
<LinearLayout
    android:id="@+id/topView"
    android:orientation="vertical"
    android:layout_marginTop="30dp"
    android:layout_centerHorizontal="true"
    android:gravity="center_horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <ImageView
        android:src="@drawable/ic_tutor"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:text="TUTOR MAP"
        android:textColor="@android:color/white"
        android:textAppearance="@style/Base.TextAppearance.AppCompat.Display1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
```

Figure 4.2: Tutor Map Starting Page XML Code in Android Studio

This portion cannot work alone. Because XML works like an empty body without soul. So, for getting this design work I needed to add JAVA code as its soul. Screenshot is given in figure 4.3.



```
public class SignUpDialogFragment extends DialogFragment implements SelectionDialogFragment.optionItemSelectedInfo {
    private EditText emailText, passText, nameText, addText, phoneText, instText;
    private EditText guardianText, guardianPhone;
    private String optionSelectedItem;
    private MaterialSpinner genderText, classText, deptText, profText;

    private String[] genderItems = {"MALE", "FEMALE"};
    private String[] classItems = {"1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"};
    private String[] deptItems = {"None", "Science", "Commerce", "Arts"};
    private String[] profItems = {"Student", "Teacher"};

    public SignUpDialogFragment() {
        // Required empty public constructor
    }

    public static SignUpDialogFragment newInstance(){
        SignUpDialogFragment fragment = new SignUpDialogFragment();
        Bundle args = new Bundle();
        fragment.setArguments(args);
        return fragment;
    }
}
```

Figure 4.3: Tutor Map Starting Page Java Code in Android Studio

Again, when a student wants to sign up for this app it will ask for some data. So, this page design is shown below, in figure 4.4.

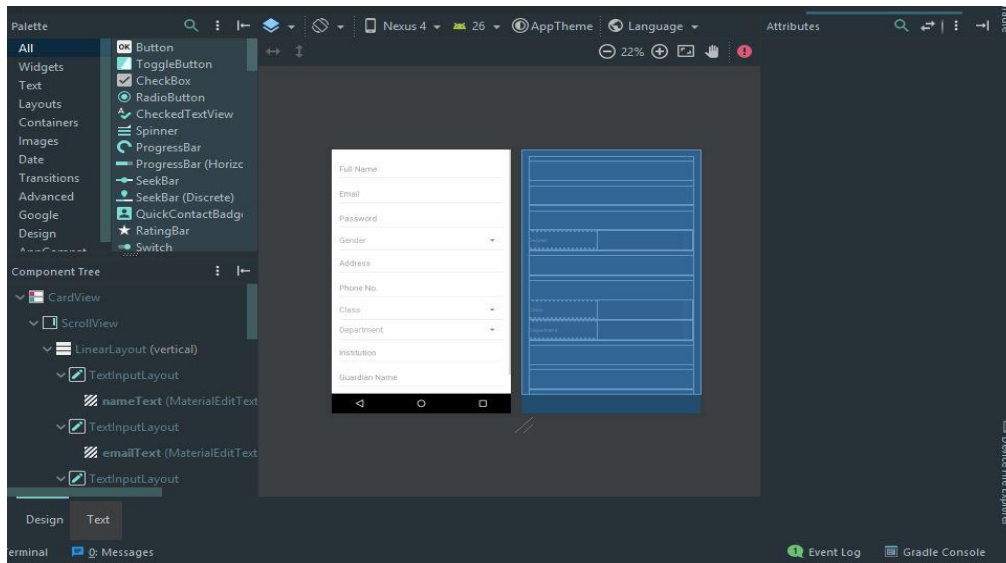


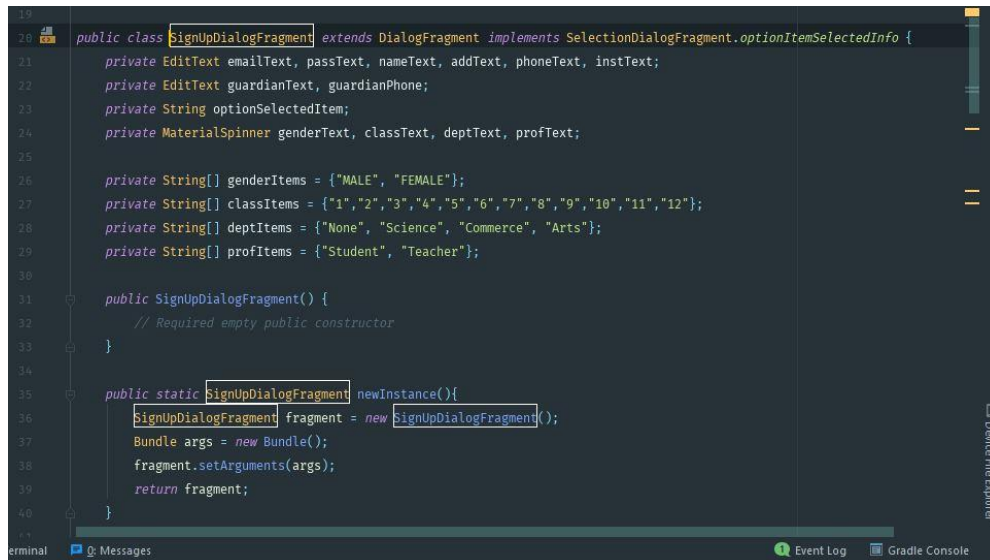
Figure 4.4: Tutor Map Registration Page Design in Android Studio

This part shows how the design was formed. And the next figure will show the XML part that formed the design shown below, in figure 4.5.



Figure 4.5: Tutor Map Registration Page XML Code in Android Studio

And this portion is not workable without its java portion. This design is meant to be taking the student data in a database and store it or crosschecking some data if its valid or not shown below, in figure 4.6.

A screenshot of the Android Studio IDE showing the Java code for a class named `SignUpDialogFragment`. The code is displayed in a dark-themed editor with line numbers on the left. The class extends `DialogFragment` and implements `SelectionDialogFragment.optionItemSelectedInfo`. It contains several private fields for text inputs and spinners, and arrays of items for dropdown menus. The code includes a constructor and a static `newInstance()` method that creates a new instance of the class and sets arguments.

```
20 public class SignUpDialogFragment extends DialogFragment implements SelectionDialogFragment.optionItemSelectedInfo {
21     private EditText emailText, passText, nameText, addText, phoneText, instText;
22     private EditText guardianText, guardianPhone;
23     private String optionSelectedItem;
24     private MaterialSpinner genderText, classText, deptText, profText;
25
26     private String[] genderItems = {"MALE", "FEMALE"};
27     private String[] classItems = {"1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"};
28     private String[] deptItems = {"None", "Science", "Commerce", "Arts"};
29     private String[] profItems = {"Student", "Teacher"};
30
31     public SignUpDialogFragment() {
32         // Required empty public constructor
33     }
34
35     public static SignUpDialogFragment newInstance(){
36         SignUpDialogFragment fragment = new SignUpDialogFragment();
37         Bundle args = new Bundle();
38         fragment.setArguments(args);
39         return fragment;
40     }
```

Figure 4.6: Tutor Map Registration Page Java Code in Android Studio

This is not the ending design of this project. But there is too many coding. So, I should not just put it in these pages.

## 4.2 Back-End Design

The back-end design is the part where the actual work happens in an app. The user cannot interact with this part or anything. The back-end part is the most crucial part of an app. Here, all the logics work behind the application. The logics of saving or displaying data from part to part is the work of back-end part.



We think the front-end part is the heart but in reality, without the back-end part we cannot do anything in an application. Most of the big companies work most of their time for designing back-end part. My project also took a lot of time for making the logic work.

To developing and maintaining the back-end section I have used Firebase database in my application [2]. Firebase is a Google product which gives support to authentication and real-time online server service for any database system shown, in figure 4.7.

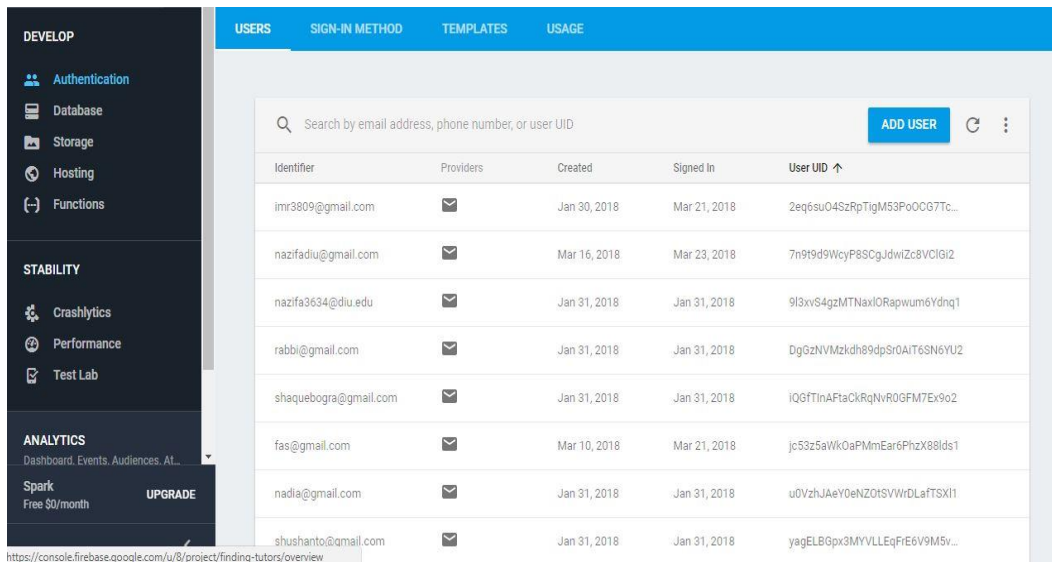


Figure 4.7: Firebase Authentication Page

This is authentication part. Where admin can add different method to get into the system. And from here admin can add user through add user option.

The main portion is database. Where I have created a simple database, which includes two entity student and teacher shown in figure, 4.8.



Figure 4.8: Firebase Database Page 1

This database system is a lot different from all other database system. It is a little easy to use and tougher to set up shown in figure 4.9 & 4.10.



Figure 4.9: Firebase Database Page 2

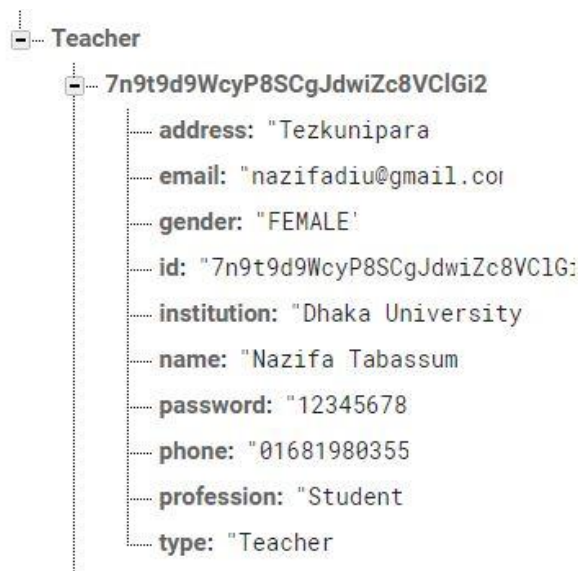


Figure 4.10: Firebase Database Page 3

### **4.3 Interaction Design And UX**

Interaction design represents the interaction of the application through the user. Interaction design means the system processing the issues that a user faces or want to proceed through the application. It mainly focuses the overall outcome of the application.

The UX stands for the user experience. User experience is important for any application. It means the overall features that is used conveniently by the user. For a better user experience there should be a better design and the application should be user friendly.

I have tried to put these two factors perfectly in my application as the user will have the satisfaction of using this app for their purpose of use. And the system shows the design as simple as it possible so that the user can have a great using experience.

### **4.4 Implementation Requirements**

My application is based on android system which can be made through the IDE android studio or ECLIPS IDE for android. These two IDE are freeware. Anyone can use it by installing it. Most android developer uses android studio for android development as it is a fast functioning software if we perfectly use a compatible hardware setting for our computer. This IDE helps us with code altering, troubleshooting, execution tooling, an adaptable form framework.

Here I will discuss about the implementation requirements that I need to complete this application.

#### **Android Studio**

Android studio IDE is a total package for developing android application. Which has everything to make an android application. I have used android studio version 3.0.1 which is the current latest version of android studio. It supports all the new plug-ins to develop an up to date application for any android version.

#### **Android SDK**

A software development kit that enables developers to create applications for the Android platform [3]. The Android SDK includes sample projects with source code, development tools, an emulator,

and required libraries to build Android applications. Applications are written using the Java programming language and run on Dalvik, a custom virtual machine designed for embedded use which runs on top of a Linux kernel. The current version of SDK is 26.1.1.

### **Firestore**

Firestore frees developers to focus crafting fantastic user experiences [4]. A user doesn't need to manage servers and don't need to write APIs. Firestore is a server, an API and a datastore, all written so generically that anyone can modify it to suit most needs. I have used it to handle the back-end of my application.

### **The Emulator**

An emulator is a special virtual operating system that can be created through the android studio to use different versions of android to debug and test an android application. So, before launching the application anyone can test the application on different versions of android [5]. I have used it to test my application on many versions of android to check it if it can be run on a physical device.

## CHAPTER 5

### IMPLEMENTATION AND TESTING

#### 5.1 Implementation of Database

Firestore real time database system has a lot of functions. It can be used as a database or a server. It can be used to authenticate the users. This is a real time database or server and it's not like the relational database system so this system can server millions and does not comprehend the quality of service.

The main database that I created to use in this project is named tutor finder. It has two parts, one is student and another one is teacher. And these parts can hold the necessary data to control the whole app.

First it takes input from the user and authenticate then it gives access to use the app shown in figure, 5.1.

```
private void onCreateUserDatabase(String as, Object user) {  
    assert auth.getUid()!=null;  
    users.child(as).child(auth.getUid())  
        .setValue(user)  
        .addOnSuccessListener((OnSuccessListener) (aVoid) → {  
            Snackbar.make(rootLayout, text: "Registration successfully done!", Snackbar.LENGTH_SHORT)  
                .show();  
            startActivity(new Intent( packageContext: MainActivity.this, DashActivity.class));  
            finish();  
        })  
        .addOnFailureListener((e) → {  
            Snackbar.make(rootLayout, text: "Registration failed due to "  
                + e.getMessage(), Snackbar.LENGTH_SHORT).show();  
        });  
}
```

Figure 5.1: Firebase and Java

This code above sends the data to firestore database to store the user.

Firestore console can be accessed from anywhere in this world. The admin can check the users at any time. The admin can delete or add new users from the console.

The system of this database or server is so easy to use that the data store is also easy in this system. All the data are stored in real time. The admin can also interfere here. The admin can delete any data that he intends to. Or add any users with all the data.

## 5.2 Implementation of Front-End Design

The usage of this application will depend on how convenient this application is. The success of an application depends on the function as well as in the design. If the user finds an interface easy to use it may attract him or her to use it.

So here I will discuss about the front-end implementations.

The starting page shows us that it requires to sign in or registration to get into the services of the application shown in figure 5.2.

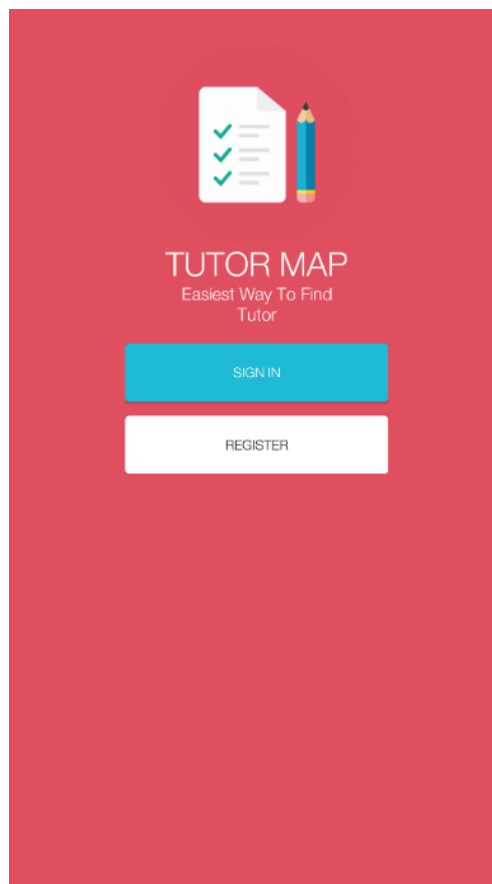


Figure 5.2: Tutor map starting page

If anyone is a new member he or she has to register to use the service. In the registration page it asks for if the user wants to register as a teacher or a student shown in figure 5.3.

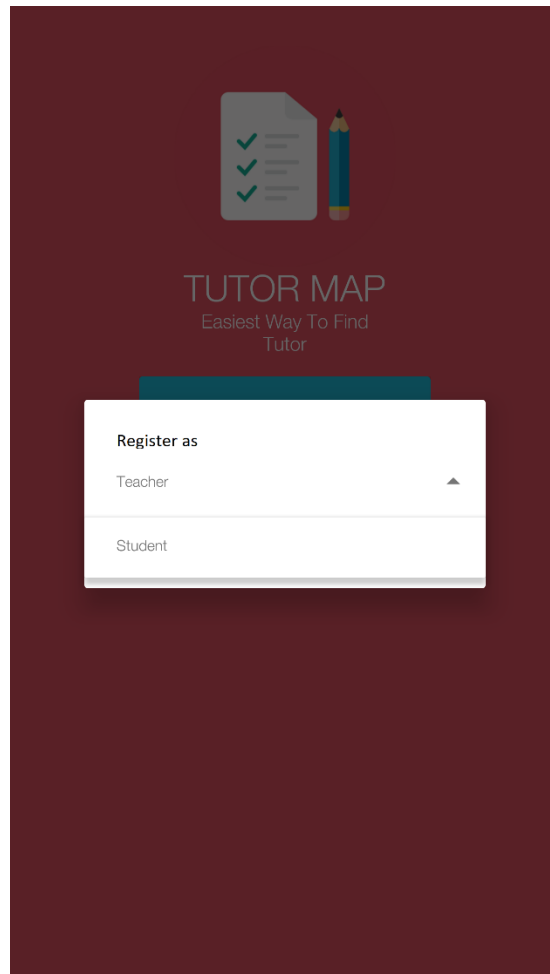


Figure 5.3: Registration Selection

If the user chooses to be a teacher then it takes in a different data field. And if the user chooses to be a student then it takes to another data field shown in figure 5.4

The image displays two registration forms side-by-side on a dark red background. The left form is titled 'Teacher REGISTER' and includes fields for Full Name, Email, Password, Gender (set to MALE), Address, Phone No., Select Your Profession (set to Student), and Institution. The right form is titled 'Student REGISTER' and includes fields for Full Name, Email, Password, Gender (set to MALE), Address, Phone No., Class (set to 1), Department (set to None), Institution, Guardian Name, and Guardian Phone No. Both forms have a 'CREATE ACCOUNT' button at the bottom right.

Figure 5.4: Registration Page for Student and Teacher



if the registration is complete it takes to their different home page. Now if a user is already registered the user can get into the app thorough the sign in option shown in figure 5.5.

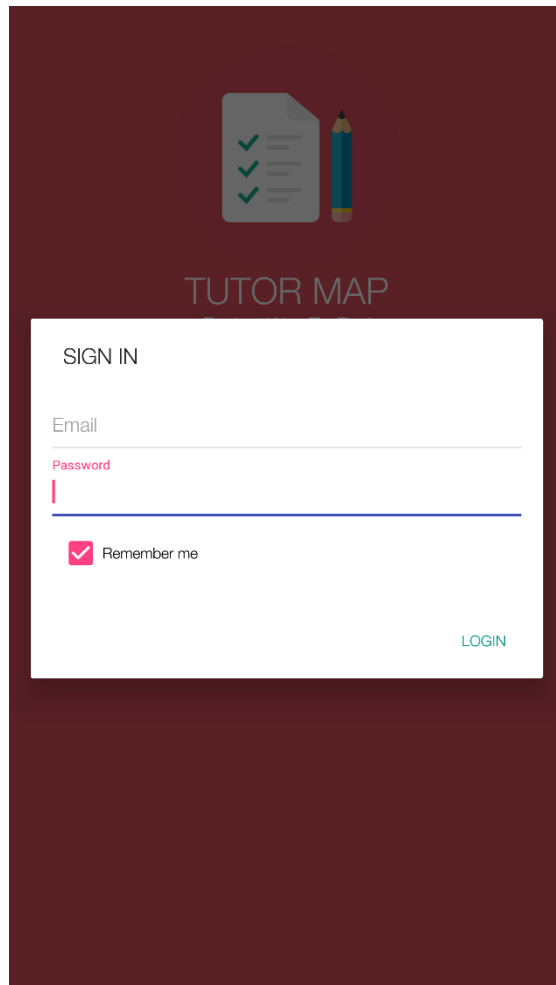


Figure 5.5: Sign in Page

After signing in it would automatically take the user to its defined page. If the user is a teacher then it would take the user to teacher home page or in other case to the student home page shown in figure 5.6.

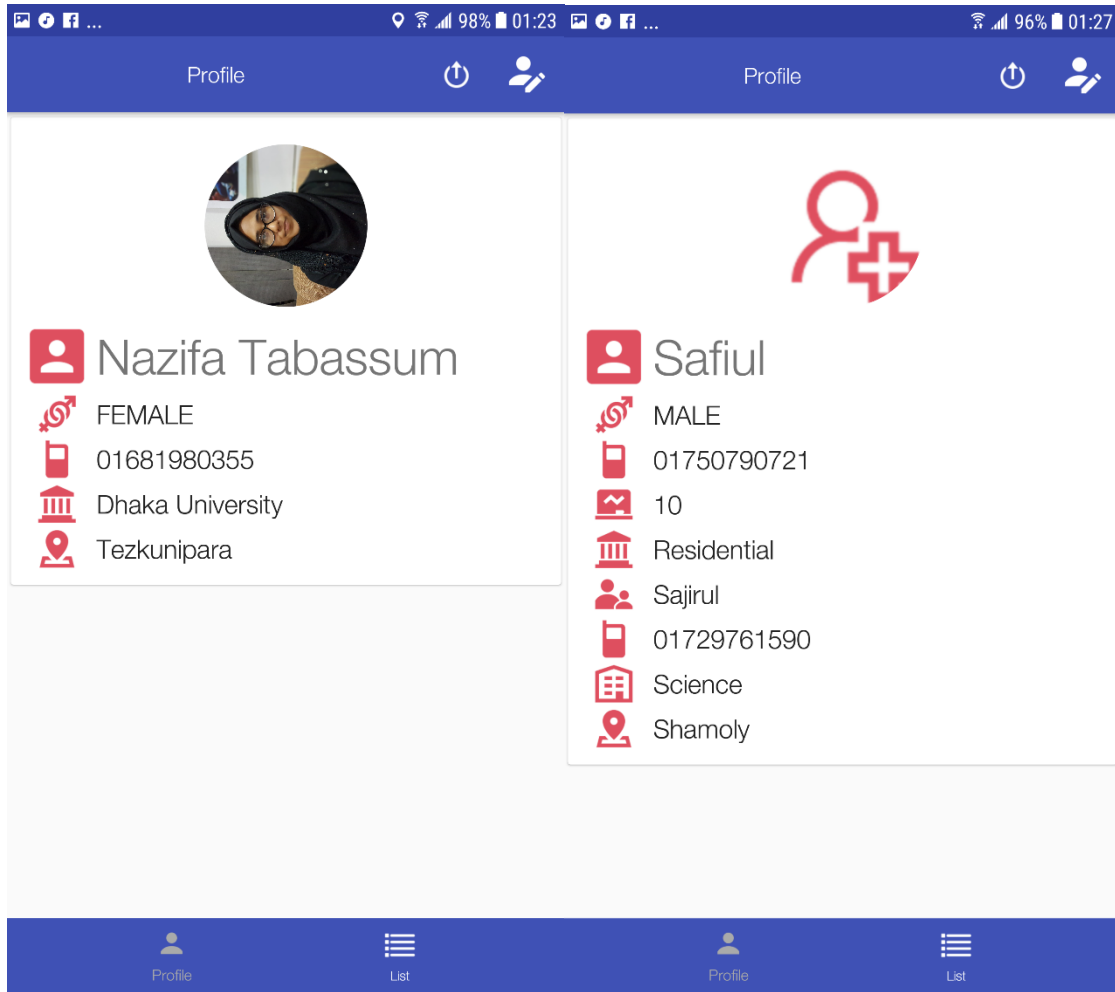


Figure 5.6: Home Page for Student and Teacher

The most of the things are almost the same as for teacher or for student. If the user wants to see the student list then he/she should click on the list tab to get into the list system. Same thing goes for the teacher list for students shown in figure 5.7.

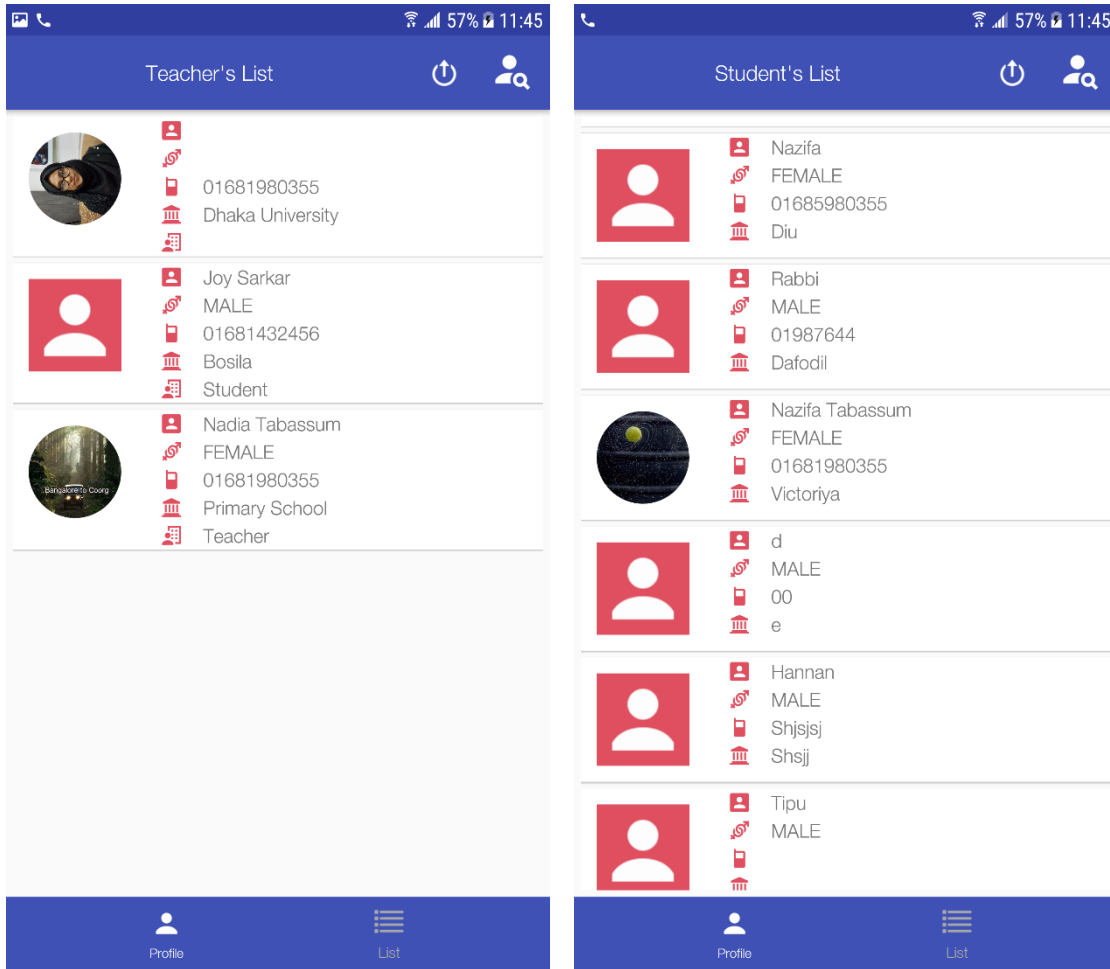


Figure 5.7: List of Teacher and Student

Now if the user wants to update the data that was provided in the registration process, the user just need to click on the edit button to do that. This process goes same for the teacher and the user shown in figure 5.8.

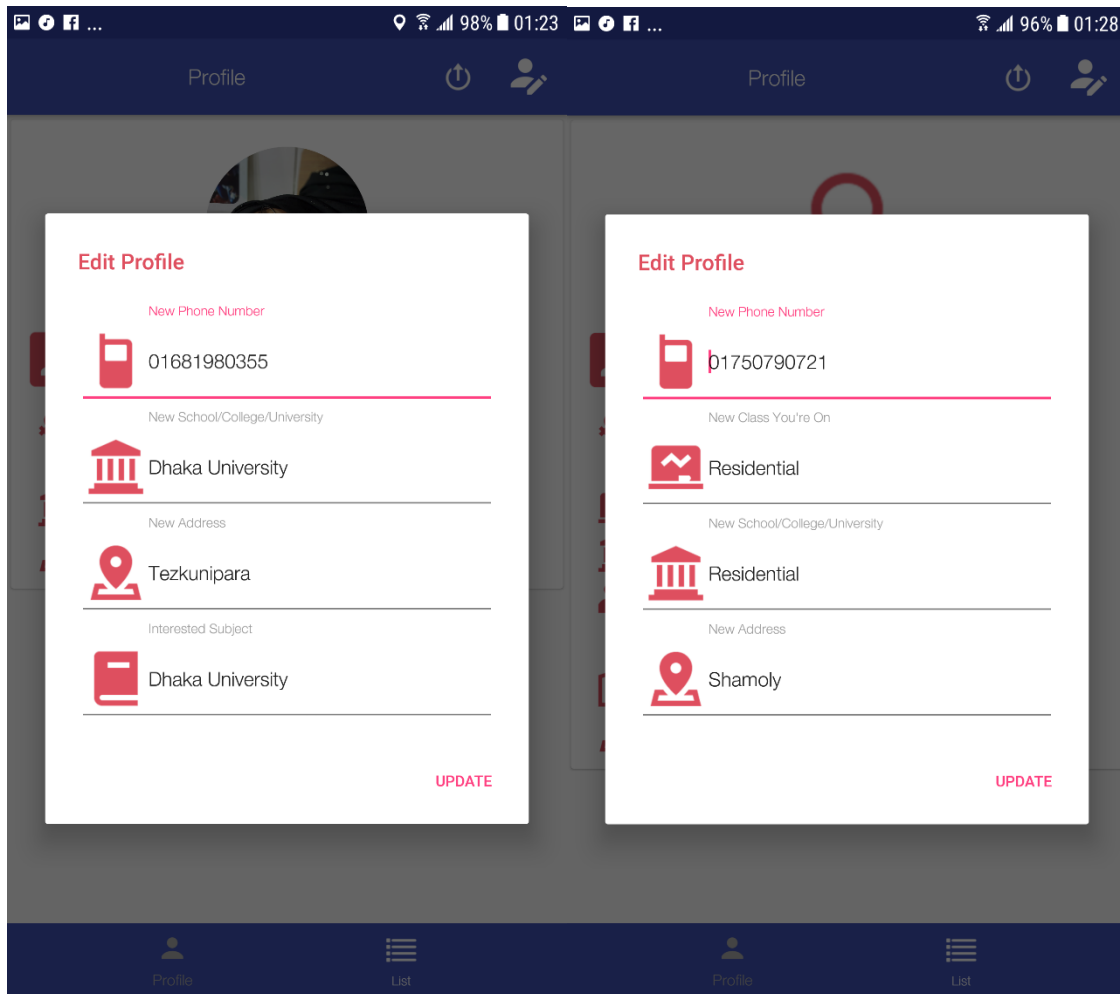


Figure 5.8: Profile Edit Option

Now for searching the user need to click on the search button, where the user can search for student or for teacher. Now the search can be completed by typing the standard of education or by the subject or by a place shown in figure 5.9.

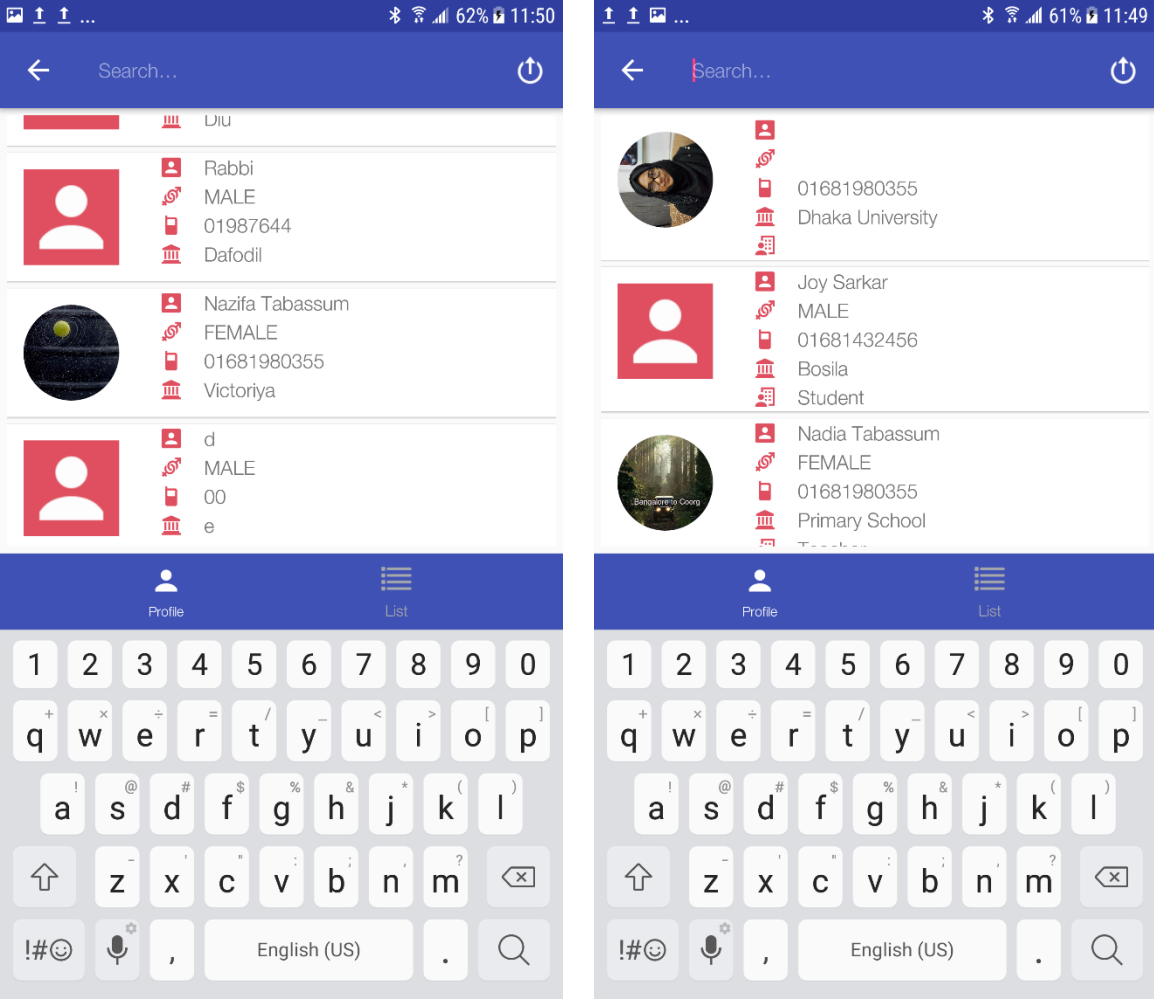


Figure 5.9: Searching Option

This is the end of the applications front-end design.

### 5.3 Implementation of Interactions

The more an application is interactive the more it is used. For making an application it needs some material design with user friendly system. For a convenient system the design should include good icon and beautiful text font with some good designed button.

I have tried to make my application more interactive to the users. So, I made the system responsive for a better user experience. For better interface I have used good icons with good text font and button.

### 5.4 Testing Implementation

Testing implementation is process of testing the implementation of a system, where tester or system architect will see various cases and specification, is it implementable or it has limitations.

Table 5.1: Testing Implementation

No	Tested case	Test input	Expected outcome	Actual outcome	Result	Tested on
1	Registration	Register as student	To successfully register	Registration complete	Passed	15.12.17
2	Registration	Register as teacher	To successfully register	Registration complete	Passed	15.12.17

3	Sign in	Enter email and password to enter the system	To enter the system	Entered the system	Passed	15.12.17
4	Sign in	Enter non-registered email and password	To not let in the system	Didn't enter the system	Passed	15.12.17
5	Searching for student	Enter place name to find student	To see the teachers of a place	Showed the list of teachers of a place	Passed	24.03.18
5	Searching for teachers	Enter place name to find teachers	To see the students of a place	Showed the list of students of a place	Passed	24.03.18

## **5.5 Test Results and Reports**

I have tested my application if it works or not as I wished it to. So, this test result will help me to understand the problem of my system and improve it. I tried to test every step of my project to make it successful tutors map.

I have tested my tutors map application many times with different android phones to see if it worked fluently. And yes, it worked perfectly on every device.



## CHAPTER 6

### CONCLUSION AND FUTURE SCOPE

#### 6.1 Discussion and Conclusion

Tutor map is an educational platform that allow students to find their ideal tutor in the most convenient way.

The app also proves to be a great platform for skilled and experience tutors to earn extra income by educating students in their leisure time. And students can choose their interested field and communicate directly with tutor through inbuilt messenger or call and schedule a class. The app also allows tutors to communicate with students through call or SMS.

#### 6.2 Scope for Further Developments

I would like to continue developing this app to add:

- Continuous communication with parent within the app
- Stability improvements.
- Giving more accurate tutor /tuition facility.
- Giving rating system, add them as a favorite and report them in the case of any misbehave or violence.

## REFERENCES

- [1] “What is the difference between front end and back end development”. [Online]. Available: <https://www.quora.com>. [Last accessed Sept. 22, 2017].
- [2] “Front End vs. Back End Mobile App Development: Learn what you need to know in 5 minutes”. [Online]. Available: <https://julyrapid.com>. [Last accessed Sept. 22, 2017].
- [3] “Android SDK”. [Online]. Available: <https://www.webopedia.com>. [Last accessed Dec. 16,2017].
- [4] “What is Firebase”. [Online]. Available: <https://howtofirebase.com>. [Last accessed Dec. 31, 2017].
- [5] “Run Apps on the Android Emulator”. [Online]. Available: <https://developer.android.com>. [Last accessed Jan. 18, 2018].
- [6] “Build Your First App”. [Online]. Available: <https://developer.android.com>. [Last accessed Feb. 11,2018].
- [7] “Tutor Finder”. [Online]. Available: <https://play.google.com>. [Last accessed Mar. 22, 2018].
- [8] “TuitionFinding”. [Online]. Available: <https://play.google.com>. [Last accessed Mar. 22, 2018].

## **APPENDIX**

### **Appendix A: Project Reflection**

The purpose of this appendix is about project reflection. From Fall 2017 semester I have started journey to make this application. The main feature of my application is finding tutor as well as student.

Also using this application one tutor can easily contact with a student or a guardian of a tutor. Firstly, I build a model for my project then I implemented my idea step by step. After many hard work and spending a lot of time finally I was able to reach my goal.

The project "Tutors Map" will be very helpful for not only student but also for a teacher. Education system can be out of media system. So, I believe that my application "Tutors Map" will be a positive application for users.

# Plagiarism Report

Checked by: <https://my.plagramme.com>

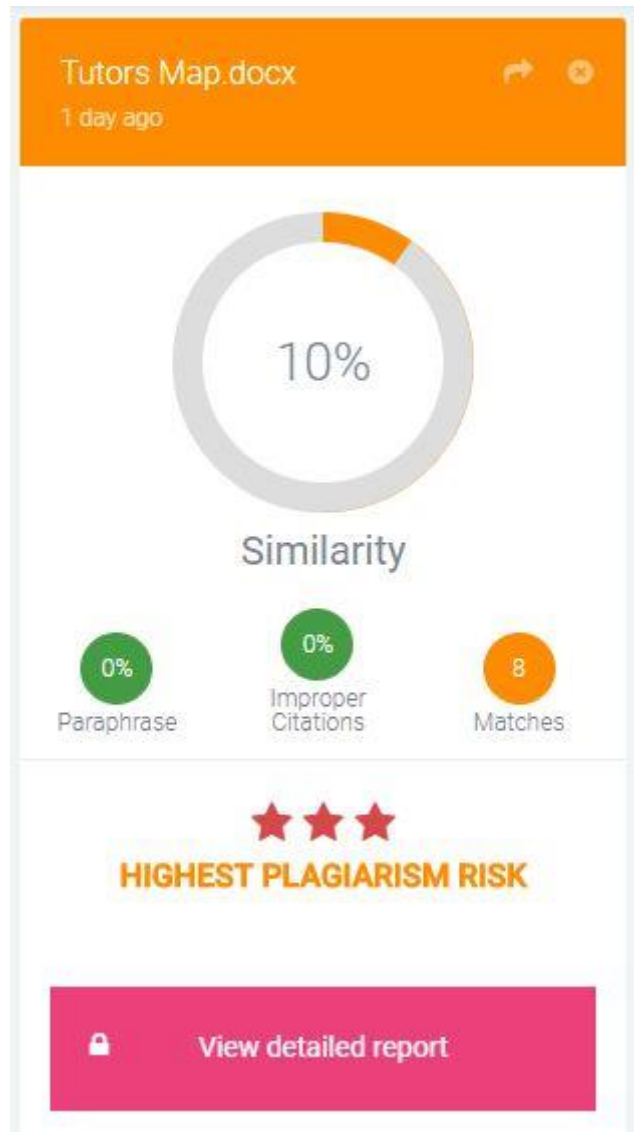


Fig: Plagiarism Report