# GEOGRAPHIC TUTION CENTER FINDING APPLICATION FOR STUDENTS

*Project report submitted*
*in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology**
**in**
**Computer Science & Engineering**
**By**

| | |
|---|---|
| **B S MANEESH** | **(17UEAG0009)** |
| **S B VIJAYA BHASKAR** | **(17UECS0805)** |
| **L MAHESH KUMAR** | **(17UECS0415)** |

*Under the guidance of*
*Ms.Almas Begum.,M.E.*
*Assistant Professor*

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
**SCHOOL OF COMPUTING**
**VEL TECH RANGARAJAN Dr.SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY**
**(Deemed to be University Estd u/s 3 of UGC Act, 1956)**
**CHENNAI 600 062, TAMILNADU, INDIA**
**June, 2021**

# CERTIFICATE

It is certified that the work contained in the project report titled ”GEOGRAPHIC TUTION CENTER FINDING APPLICATION FOR STUDENTS” by the team ”B S MANEESH (17UEAG0009), S B VIJAYA BHASKAR (17UECS0805), L MAHESH KUMAR (17UECS0415)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

**Signature of Supervisor**
**Ms. ALMAS BEGUM**
**Assistant Professor**
**Computer Science & Engineering**
**School of Computing**
**Vel Tech Rangarajan Dr.Sagunthala R&D**
**Institute of Science and Technology**
**June, 2021**

**Signature of Head of the Department**
**Dr.V.Srinivasa Rao**
**Professor & Head**
**Computer Science & Engineering**
**School of Computing**
**Vel Tech Rangarajan Dr.Sagunthala R&D**
**Institute of Science and Technology**
**June, 2021**

# DECLARATION

We declare that this written submission represents my ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
(B S MANEESH)
Date: / /

(Signature)
(S B VIJAYA BHASKAR)
Date: / /

(Signature)
(L MAHESH KUMAR)
Date: / /

# APPROVAL SHEET

This project report entitled GEOGRAPHIC TUTION CENTER FINDING APPLICATION FOR STUDENTS by B S MANEESH (17UEAG0009), S B VIJAYA BHASKAR (17UECS0805), L MAHESH KUMAR (17UECS0415) is approved for the degree of B.Tech in Computer Science & Engineering.

**Examiners** **Supervisor**

Ms. ALMAS BEGUM,M.E.

**Date: / /**

**place:**

# ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO). DSc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN** for providing us with an environment to complete our project successfully.

We obligated to our beloved **Registrar Prof. Dr.E.KANNAN, Ph.D** for providing immense support in all our endeavors.

We thankful to our esteemed **Director of Academics Dr. ANNE KOTESWARA RAO, Ph.D.,** providing a wonderful environment to complete our project successfully.

We record indebtedness to our **Dean/Head of the Department Dr.V.SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

A special thanks to our **Project Coordinators Mr.V.ASHOK KUMAR, M.Tech., Ms.S.FLORENCE, M.Tech., Ms.C.SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We also take this opportunity to express a deep sense of gratitude to Our Internal Supervisor **Ms. ALMAS BEGUM,M.E.,** for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

| | |
|---|---|
| **B S MANEESH** | **(17UEAG0009)** |
| **S B VIJAYA BHASKAR** | **(17UECS0805)** |
| **L MAHESH KUMAR** | **(17UECS0415)** |

# ABSTRACT

In this current world, we infinitely are dependent on Mobile applications that we are doing everything from mobile apps, from our morning emails to finding people, getting food to all our reservations are dependent on our smartphones. Over the past few years, new Mobile app developers and Billion dollar software companies have come with innovative new ideas to make our life simpler. This project proposes such similar idea that might help the students and teachers living in any same area to self-identify themselves and connect through an encrypted and secure student-mentor relationship. The application acts as a bridge for both parties where any two kinds of different user types can able to communicate with one another, identify the users with reviews, all via self-secured privacy and encrypted data. The application has the validity to make an impact in society, schools, and even universities android mobile phone users.

Keyword:**Application,Mobile,Tuition center,Android,Database**

# LIST OF FIGURES

# LIST OF ACRONYM AND ABBREVIATION

| | |
|---|---|
| AD | Advertisement |
| API | Application Programming Interface |
| DB | Data Base |
| GPS | Global Positioning System |
| IOS | iPhone Operating System |
| NOSQL | Non Structured Query Language |
| SQL | Structured Query Language |
| SDK | Software Development Kit |
| UI | User Interface |
| XAML | Extensible Application Mark-up Language |

# TABLE OF CONTENTS

**Page.No**

# Chapter 1

# INTRODUCTION

## 1.1 Introduction

ANDROID BASED COLLEGE APPLICATION FOR STUDENTS[1] people using and utilizing most of the mobile applications both Android and IOS, Android mobile phone users stand out in numbers higher than IOS users. So the application is specifically created for Android app users due to the larger market and greater number of download possibilities. There are many kinds of possible technologies implemented to make user's life much simpler every day. Among many kinds of apps, Social media and education directly provide a huge help to our daily life since it reduces the possibility of physical meeting and makes connections and data passing faster than ever before in history.

This project is a Social media, Educational app that helps users of different types example Student-Tutor to connect via a mobile app interface with the No SQL i.e. Fire base DB. Since the connection of people for various purposes seems possible in our era, this application might have some validity out in the market. There are many students and tutors out there who might find this app very useful either to promote their identity or to find the required one. There is always a demand for these kinds of applications but society is not well aware to acknowledge the existence of such kinds of applications. But there is a point to assure that there are needs for people to find tutors and students on both sides of user parties. It is just that they lack information about the existence of such kinds of Mobile application

## 1.2 Aim of the project

The aim of this project is that of creating an Android application that allows any student who uses the app living in any area around the world to find small, large-time tuition centers that share the same place as the tuition centers. Allowing them to connect via end-to-end encrypted personal chat and google location-based services for both parties with guaranteed security for data privacy. Secured review system that does not allows third parties to review over the tutor just like that, only the students who are under the learning circle of the tuition can able to review feedback over the place.

The choice to change location will be a convenient option for both students and tutors if they wanted to change their locations from one to another any number of times the filter based on location change gets implemented for users around the particular user type.

## 1.3 Project Domain

The domain of the project falls under Android mobile application building and development and social media network, thus it both is applicable for this project, A mobile application that acts as a platform for social people i.e. teacher and student to connect via a com-men social media platform.

## 1.4 Scope of the Project

The scope of the project is to help students connect with tutors who are living within the same area. To maintain a secured Teacher-student relationship between the users with increased privacy. Manage chat between users encrypted in the database. Increase the number of users on both types so users get better options of choices while selecting the respective needs. To make it cost free and available for all is a key aspect of this application

## 1.5 Methodology

Firstly to create our profile in that app either as student nor teacher on done we can search for tuition centers if we created profile as student. The app will show nearest tuition centre, fees, faculty name etc According to the address we provided. If create teacher type account the user can upload details of our tuition centre like timings, fees, location etc in that app so that students can view our details . Once the users are convenient with tuition timing location and fee we can contact the specific tuition faculty and join the classes. The application is desgined for many type of tuition classes like yoga, karate, coding, school, spoken class, music, arts etc

# Chapter 2

# LITERATURE REVIEW

Biswajeet Sethi [1] et al, proposed that Some of the applications of web service 2.0 need big data handling. This requires a relational database to scale vertically to achieve demand for better performance, An application that requires a higher scale of user data. These are some important considerations for designers to come up with a new set of databases, known as NoSQL. With the growing demand for cloud computing and its development, this paper deals with features and data models of NoSQL databases used in cloud computing environments.

Hana R [2] proposed tools to develop the mobile application. In this paper SDK Emulator was used to implement some features of Android devices. such as( send messages, connect to the internet, launch google Browser, view the basic information of the device(Mobile memory space, and system security)).

Prachi Sasankar [3] et al introduced the Android platform and its features give a detailed description of the Android application view of mobile android application developers. A simple music player is an instance to represent the basic working processes of the application itself. This paper guides understanding the operation mechanism of Android applications.

1Iqbaldeep Kaur [4] et al introduced that, with the development of the software industry, the use of Object-Oriented Software Engineering (OOSE) has increased the demand in the real world. The origin of the OOSE design of the software has expanded much and is now considered as one of the smooth software integration processes. The OOSE is the combination of Object-Oriented Programming (OOP) which provides a powerful way for development.

M. Bishop [5] projected systems programming languages. This paper looks at what makes it special. The projects exercise Java to the full along with its features and API. The first is a Web Computing Skeleton for long remote execution. The second provides open query mechanisms to a better and spatial database. The third expands a distributed algorithm visualization system. .

Niharika Dedhia1 [6] et al proposed important of application in education sector. Nowadays Smartphones have become an important part of everyone's life. These smartphones have changed human life for the better. "Android Based Campus Solution" is a college management application aimed at managing most of the activities of the college. The main objective of this mobile app is to make advancements in institutional activities. This mobile application helps in adding automation in managing institutional information. The existing system uses the website for publishing notices, or peon helps in circulating notices.

Sind [7] et al exposed the importance of tuition centers in the perception of students. It tries to find out the role of tuition centers and their owners to support the student to learn in a better way and to bring good grades exams.

Rashad Yazdanifard [8] proposed a fact that, Online marketing has increased nowadays. To expand their business, using mobile applications to do online business is the best way to promote their business. As today people have the habit to use smartphones irresistibly in their daily lives, they could download the application from any online store. It will be much more convenient to start a business with a low-cost budget.

RajatPorwal [9] et al exposed the importance of a mobile application is to use and access information and this requisite encouraged the need for a college that can be used by all the students and faculties on the campus. The features applied in the college application, the techniques used on them are described in this paper. Summarizes all the major segments such as sharing class notes, and for many other purposes.

# Chapter 3

# PROJECT DESCRIPTION

## 3.1 Existing System

As per the existing system, there is much application that is available to find tuition centers, but not any one of them is honest in reviews and price, Most of the app available out there only is not free and many of them are private profitable coaching organization that promotes the users only to support their organizations. This takes away the opportunity that needs to be given to third-party small-time private canters to self-identify themselves.

**Disadvantages**

- Not free
- Only large corporation
- honest review system
- No free services.

## 3.2 Proposed System

The advantage of this application is that it is free completely, no cost is applied for any users for any service provided within the application. The next thing is that not everyone can review tutors just like that, where only those students under the faculties couching are allowed to review the respective faculties. And this app supports all third party users to be part of the search list. Free for users.

**Advantages**

- User friendly
- Safe proof review system
- Available for all third parties

By comparing the proposed model with different ML and exploitary data analysis techniques,we can conclude with the best algorithm with the highest accuracy score possible.

## 3.3 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key consideration in the feasibility analysis are:
1.Economic Feasibility.

2.Technical Feasibility.

3.Social Feasibility.

### 3.3.1 Economic Feasibility

As of now the application available in the market contains in-app punches and making the users pay for the service provider makes only the affordable and knowledgeable to access services. And it is an understandable thing to do since there are bills that needed to pay for database services, but the project application we have been working on user monetization for the app promotions and income allowing both types of users to access all services provided within the app

### 3.3.2 Technical Feasibility

Basically this project is a Global tuition student find an app that allows bothering the type of users to find the opposite types of user, this application is constructed in android studio. For designing, the front end is built with XML and backed with java and Firebase, NoSQL database. Instead of developing a different user interface for a different type of users, the user interface is designed in such a way that both the type of users can use the same page but it differs based on their user type.3.3.3 Social Feasibility.

### 3.3.3 Social Feasibility.

The interaction between users is following certain conditions of connections. Where not any use of the same type can interact or connect. The teacher and student as different users can connect. The reach-out start-up connections are only allowed for students to connect with the teacher and not vice versa. The chat is encoded so not even the admin view it from the server side database.

## 3.4 System Specification

### 3.4.1 Hardware Specification

- Android Mobile
- 2GB RAM
- 5MB memory space
- Sim card

### 3.4.2 Software Specification

- Android OS Greater than 4.0
- Internet connection Greater than 2G

### 3.4.3 Standards and Policies

The Mobile application team is committed to protecting. Your personal information and your right to privacy. If you have any questions or concerns about this privacy notice, or our practices With regards to your personal information, please contact us at maneeshbhallae@gmail.com When you use our mobile application, as the case may be (the ”TueOn’) and more generally, use any to our services (the Services, which include the App), we appreciate that you are trusting us with your personal information. Vet take your privacy very seriously. In this privacy notice, we seek to explain to you in the clearest way possible what information we collect, how we use it and what rights you have in relation to it. We hope you take some time to read through it carefully, as it is important. If there are any terms in this privacy notice that you do not agree with, please discontinue use of our Services immediately. This privacy notice applies t0 all information collected through our Services (which, as described above, includes Our App), as well as, any related Services, sales, marketing or events.

# Chapter 4

# MODULE DESCRIPTION

## 4.1 General Architecture

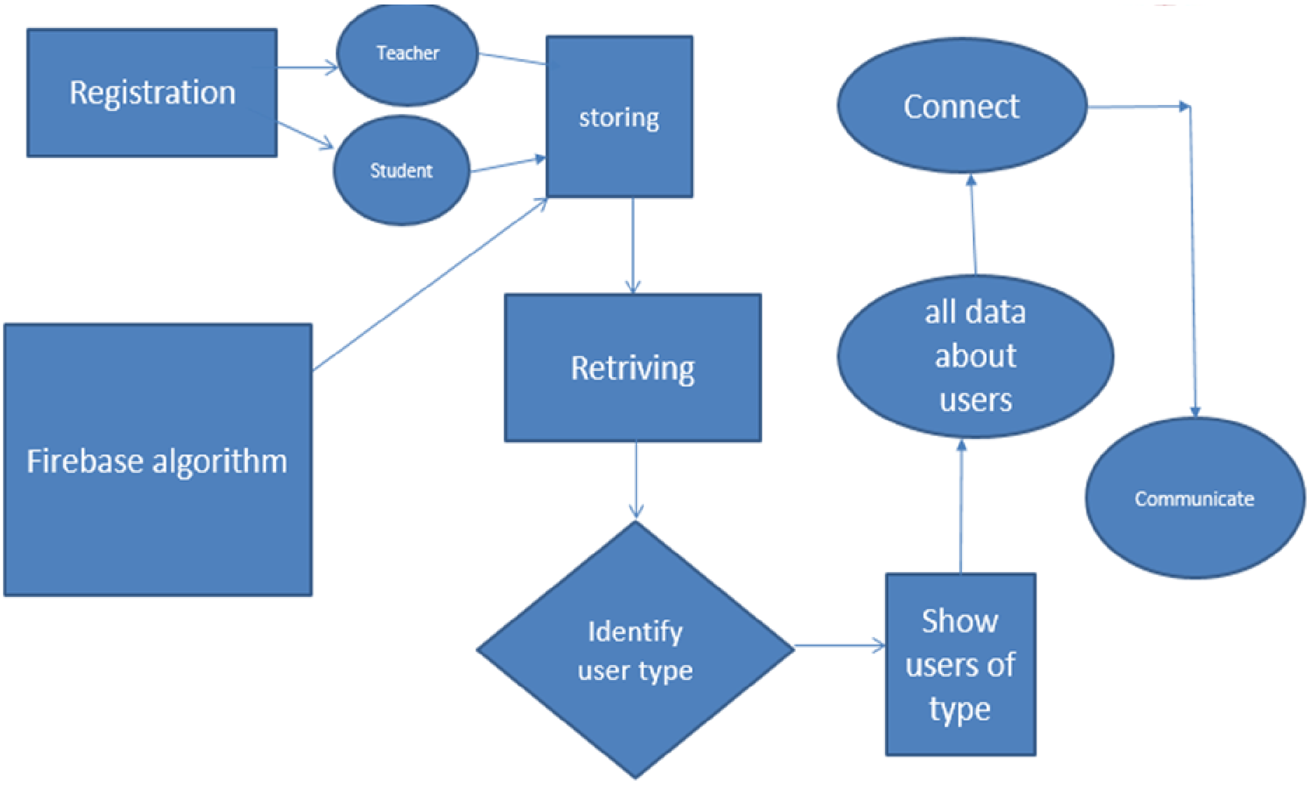


Figure 4.1: **General Architecture Diagram**

### 4.1.1 General Architecture Description

The architecture shows that the register is categorized into two types one is the user and the teacher are asked to enter the information in the application input platforms and the entered data gets stored in the database. There is a retrieving algorithm that runs in the app based on location and needed types of search results, after the filter and the data of the needed users and displayed to the user. The possibility of connections is possible between teacher and student.

## 4.2 Design Phase

### 4.2.1 Data Flow Diagram

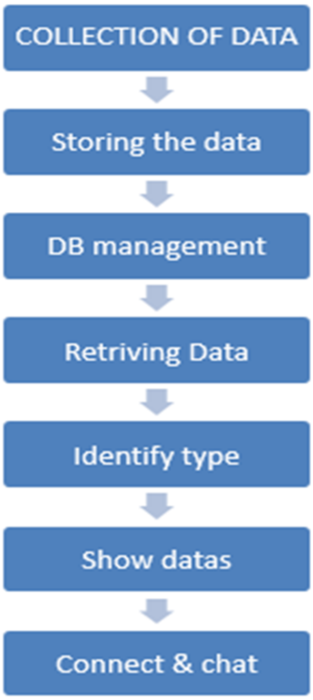


Figure 4.2: **Data Flow Diagram**

### 4.2.2 Data Flow Description

The data flow shows the flow of data through the application TueOn, firstly the collections of data from the users are stored in the database and retrieved based on search filters, as the data gets displayed to the users chat connection and call can be made available.

### 4.2.3 ER Diagram

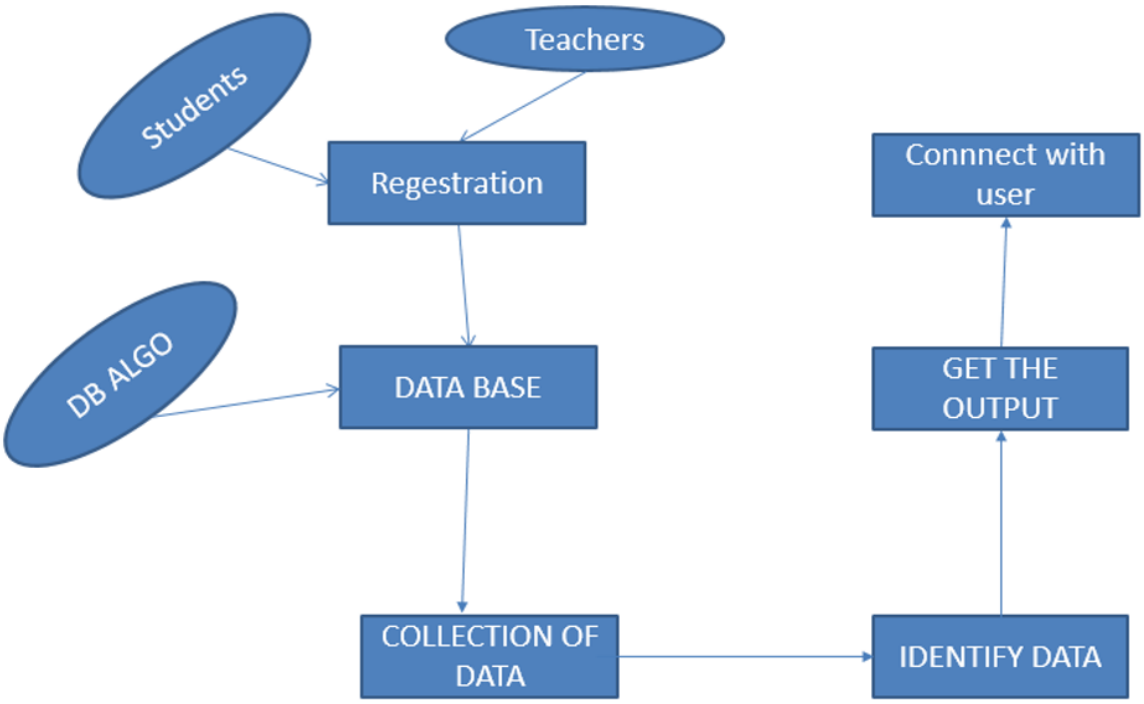


Figure 4.3: **ER Diagram**

### 4.2.4 Description of ER Diagram

The two different users i.e. Teacher and Student type users are allowed to login through the same interface and the algorithm identifies the type of users and changes the activity as per the users. The entered data are allowed to store in the database can be altered whenever we want.

# Chapter 5

# IMPLEMENTATION TESTING

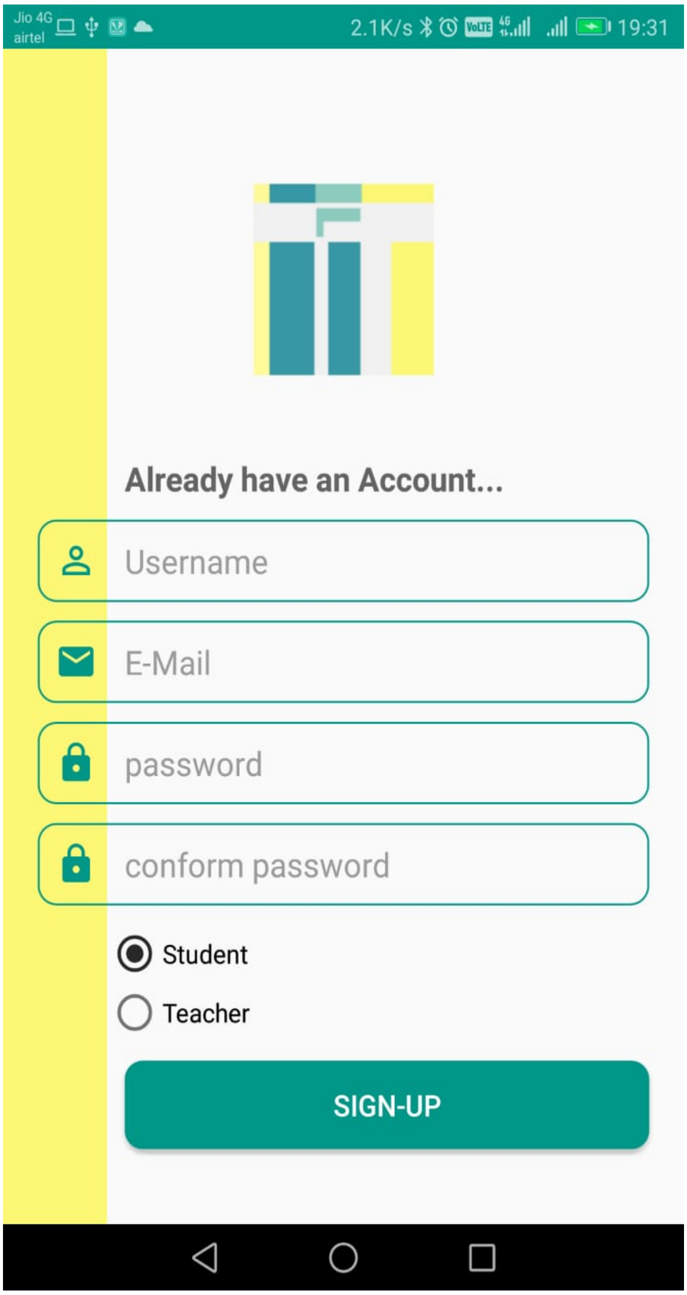
In this phase, the implementation of individual modules as explained in the previous chapters and are trying to combine all into one single module. With these ,it is may be sited as one of the intuitive methods in solving and satisfying a complete new analytical architecture.This ensures that all other modules are integrated and works as expected.The expected output from the combination of all the modules can be thus be converted into a fully working prototype to be presented to the client.The implementation process merges technological different modules to become as a whole part and thus can be expected to produce the desired output.

## 5.1 Input Output

### 5.1.1 Input Design

The input designs consist of several steps of input providing field. Many activities with a variety of details needed to be filled before any set of processes to be undergoing. Some of their deals with personal information and other social-friendly details about the individual student or the teacher. All the set of information provided in the settings activity makes up and categorizes the users to upcoming extents.

### 5.1.2 Input Screenshots



Already have an Account...

● Student

○ Teacher

Figure 5.1: **Registration Activity**

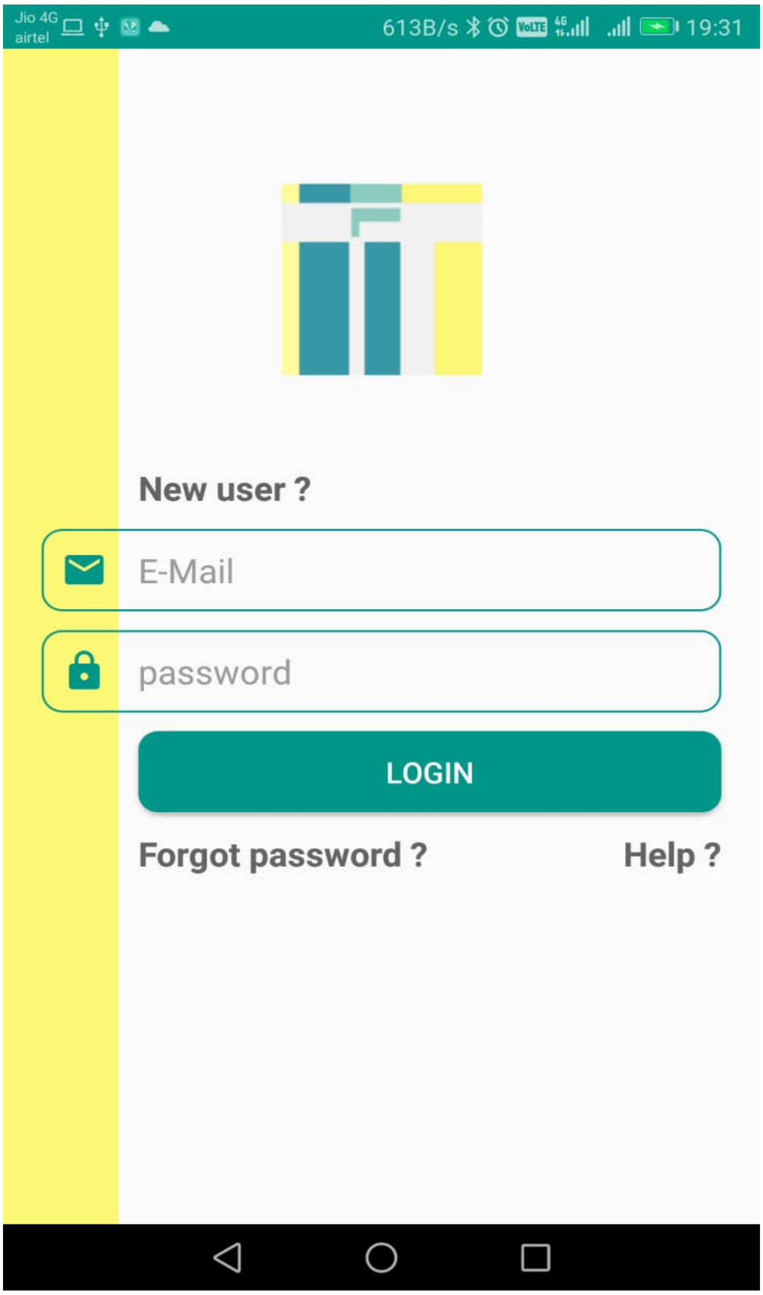
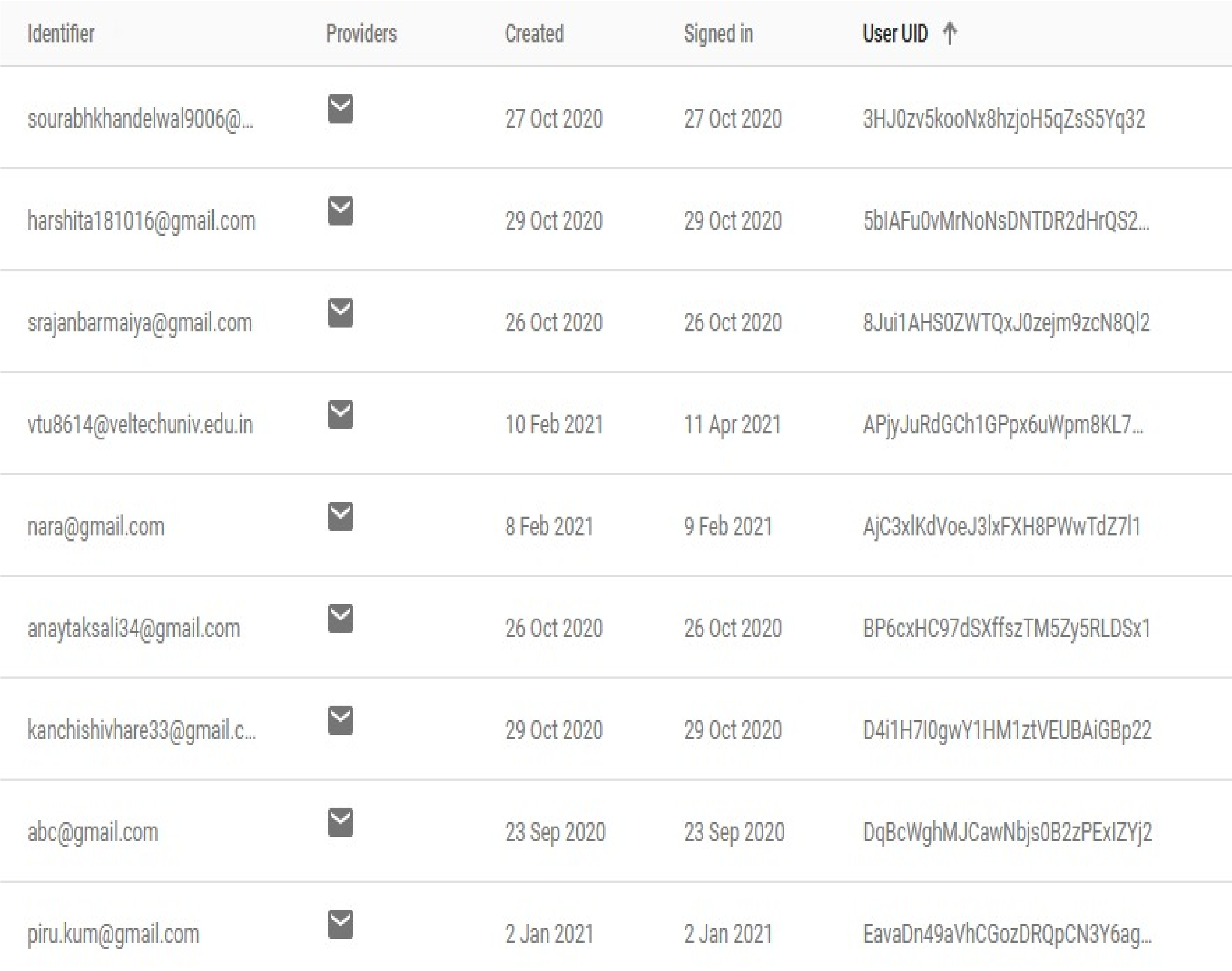


Figure 5.2: **Login Activity**

### 5.1.3 Output Design

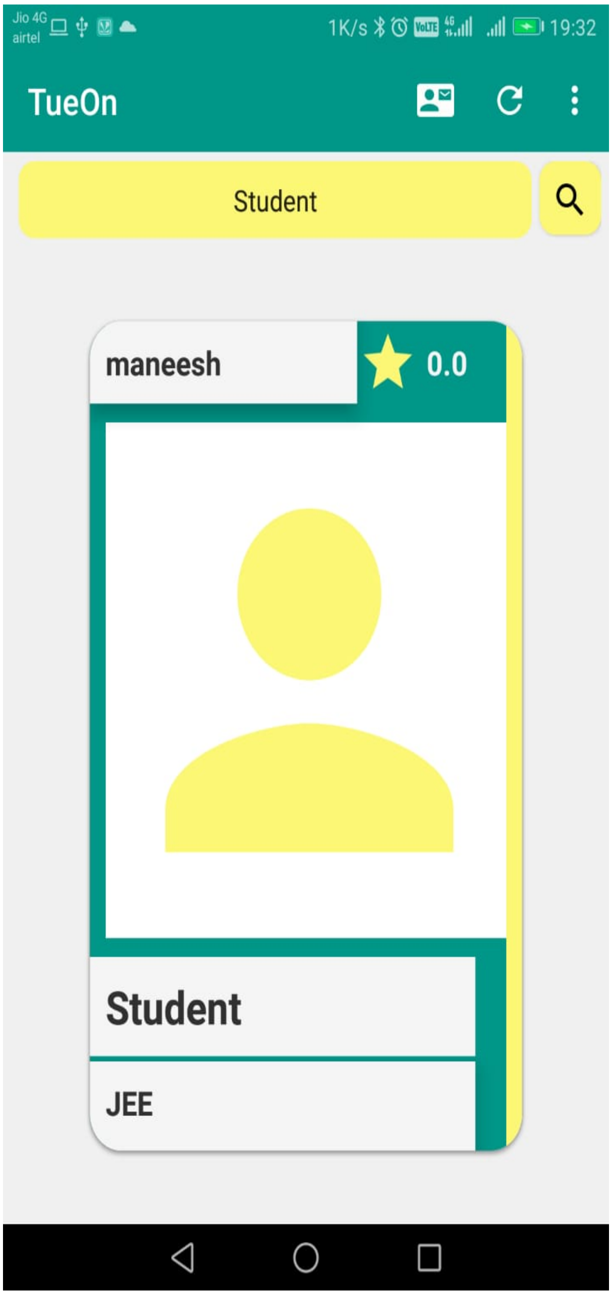
The output of the users are filtered based on the location they have provided in the settings activity and the possibility of connecting with the Teacher for students via chat gets enabled if the proper selecting process is done, The retrieved data are displayed in a car view system which is swappable either left or right. Only the opposite type of users are visible for the current type of user

### 5.1.4 OutputScreenshots



| Identifier | Providers | Created | Signed in | User UID ↑ |
|---|---|---|---|---|
| sourabhkhandelwal9006@... | ✉ | 27 Oct 2020 | 27 Oct 2020 | 3HJ0zv5kooNx8hzjoH5qZsS5Yq32 |
| harshita181016@gmail.com | ✉ | 29 Oct 2020 | 29 Oct 2020 | 5bIAFu0vMrNoNsDNTDR2dHrQS2... |
| srajanbarmaiya@gmail.com | ✉ | 26 Oct 2020 | 26 Oct 2020 | 8Jui1AHS0ZWTQxJ0zejm9zcN8Ql2 |
| vtu8614@veltechuniv.edu.in | ✉ | 10 Feb 2021 | 11 Apr 2021 | APjyJuRdGCh1GPpx6uWpm8KL7... |
| nara@gmail.com | ✉ | 8 Feb 2021 | 9 Feb 2021 | AjC3xlKdVoeJ3lxFXH8PWwTdZ7l1 |
| anaytaksali34@gmail.com | ✉ | 26 Oct 2020 | 26 Oct 2020 | BP6cxHC97dSXffszTM5Zy5RLDSx1 |
| kanchishivhare33@gmail.c... | ✉ | 29 Oct 2020 | 29 Oct 2020 | D4i1H7l0gwY1HM1ztVEUBAiGBp22 |
| abc@gmail.com | ✉ | 23 Sep 2020 | 23 Sep 2020 | DqBcWghMJCawNbjs0B2zPExIZYj2 |
| piru.kum@gmail.com | ✉ | 2 Jan 2021 | 2 Jan 2021 | EavaDn49aVhCGozDRQpCN3Y6ag... |

Figure 5.3: **Authentication**



maneesh

0.0

Student

JEE

Figure 5.4: **Home page activity**

## 5.2 Testing

Testing is the process of finding a bug in the model. There are many types of testing depending upon the project but we here in this project only rely on unit testing, white box, and black-box testing. We check the connectivity of mobile applications with the firebase database by reading data hosted on the server cloud.

## 5.3 Types of Testing

### 5.3.1 Unit Testing

**Input**

```
public class LoginActivity extends AppCompatActivity {
    private Button login;
    private EditText mail;
    private EditText password,mail2;
    private TextView jumpToR,forgotPass,LoginHelpBtn;
    private ProgressBar LogPg;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        login=findViewById(R.id.loginbtn);
        login.setEnabled(true);
        mail=findViewById(R.id.mailText);
        password=findViewById(R.id.passwordText);
        jumpToR=findViewById(R.id.jumpLogin);
        forgotPass=findViewById(R.id.forgotpass);
        LogPg=findViewById(R.id.LogPg);
        LoginHelpBtn=findViewById(R.id.LoginHelpBtn);
        LoginHelpBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent jjj=new Intent(LoginActivity.this,HelpPageActivity.class);
                startActivity(jjj);
            }
        });
        jumpToR.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(LoginActivity.this,RegisterActivity.class);
                startActivity(intent);
                finish();
            }
        });
        forgotPass.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                LayoutInflater layoutInflater = null;
                final AlertDialog.Builder builder= new AlertDialog.Builder(LoginActivity.this);
                builder.setTitle(”Forgot Password”);
                View view= LayoutInflater.from(LoginActivity.this). inflate(R.layout.dialog_forgotpass,null);
                mail2=view.findViewById(R.id.dialogpass);
                builder.setView(view);
                builder.setPositiveButton(”Reset”, new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int whichButton) {
                        String m=mail2.getText().toString();
                        if(m!=null && android.util.Patterns.EMAIL_ADDRESS.matcher(m).matches()){
                            forgotPassword(m);
                        }else{
                            Toast.makeText(LoginActivity.this,”Please verify the Mail”,Toast.LENGTH_SHORT).show();

                        }
                    }
                })
                        .setNegativeButton(”Cancel”, new DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface dialog, int
                                whichButton) {
                                dialog.cancel();
                            }
                        });
                builder.show();
            }
        });
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String tempMail=mail.getText().toString();
                String tempPass=password.getText().toString();

                if(!TextUtils.isEmpty(tempMail)  &&  !TextUtils.isEmpty(tempPass
                    )){
                    LogPg.setVisibility(View.VISIBLE);
                    login.setEnabled(false);
                    FirebaseAuth.getInstance().signInWithEmailAndPassword(
                        tempMail,tempPass).addOnCompleteListener(new
                        OnCompleteListener<AuthResult>() {
                        @Override
                        public void onComplete(@NonNull Task<AuthResult> task) {
                            if(task.isSuccessful()){
                                LogPg.setVisibility(View.INVISIBLE);
                                login.setEnabled(true);
                                Intent intent=new Intent(LoginActivity.this,
                                    MainActivity.class);
                                startActivity(intent);
                                finish();
                            }
                            else{
                                LogPg.setVisibility(View.INVISIBLE);
                                login.setEnabled(true);        Toast.makeText(
                                    LoginActivity.this,”Error”+task.getException
                                    ().getMessage(),Toast.LENGTH_SHORT).show();
                            }
                        }
                    });
                }else{
                    login.setEnabled(true);
                    Toast.makeText(LoginActivity.this,”Please fill all the fields”,Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
    private void forgotPassword(String m) {
        FirebaseAuth.getInstance().sendPasswordResetEmail(m)
                .addOnCompleteListener(new OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {
                        if (task.isSuccessful()) {
                            Toast.makeText(LoginActivity.this,”Verification mail sent”,Toast.LENGTH_SHORT).show();
                        }
                        else
                        {
                            Toast.makeText(LoginActivity.this,”Error”+task.getException(),Toast.LENGTH_SHORT).show();
                        }
                    }
                });
    }
    @Override
    protected void onStart() {
        super.onStart();
        FirebaseUser firebaseUser= FirebaseAuth.getInstance().getCurrentUser();

        if(firebaseUser!=null){
            Intent intent=new Intent(LoginActivity.this,MainActivity.class);
            startActivity(intent);
            finish();
        }
    }
}
```

### Test result

The login of any two kind of users are working perfectly as they can login with with their mailed and password to login.

## 5.3.2 Integration Testing

### Input

```
public class MainActivity extends AppCompatActivity {
private cards cards_data[];
    private arrayAdapter arrayAdapter;
    private int i;
    Toolbar tb;
    private Spinner dropdown;
    private Button search;
    private String searchedProfessioni;
    private DatabaseReference databaseReference;
    private String currentUserType2;
    private String otherUserType2;
    private TextView sorryText;
    String local=””;
    private String fistSearch ;
    ArrayAdapter<CharSequence> adapter;
    List<cards>al;
    private boolean isConnected(){
        ConnectivityManager connectivityManager=(ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo=connectivityManager.getActiveNetworkInfo();
        return  networkInfo!=null && networkInfo.isConnected();
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        dropdown = findViewById(R.id.spinner1);
        adapter = ArrayAdapter.createFromResource(MainActivity.this,
                R.array.load_test, android.R.layout.simple_spinner_item);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        dropdown.setAdapter(adapter);
        MobileAds.initialize(this, new OnInitializationCompleteListener() {
            @Override
            public void onInitializationComplete(InitializationStatus initializationStatus) {
            }
        });
        tb=(Toolbar) findViewById(R.id.main_toolbar);
        sorryText=findViewById(R.id.sorryText);

        setSupportActionBar(tb);
        getSupportActionBar().setTitle(”TueOn”);
        databaseReference = FirebaseDatabase.getInstance().getReference().child(”User”);
        al = new ArrayList<cards>();
        int location = 0;
        if(FirebaseAuth.getInstance().getCurrentUser()==null){
            Intent intent=new Intent(MainActivity.this,LoginActivity.class);
        startActivity(intent);
        finish();}
        if(FirebaseAuth.getInstance().getCurrentUser()!=null)
        FirebaseDatabase.getInstance().getReference().child(”User”).child(FirebaseAuth.getInstance().getCurrentUser().getUid())
                .addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                        if (dataSnapshot.exists()) {
                            if (dataSnapshot.child(”userType”) != null) {
                                currentUserType2 = dataSnapshot.child(”userType”).getValue().toString();
                                MatchActivity.getUserType(currentUserType2);
                                switch (currentUserType2) {
                                    case ”Student”:
                                        otherUserType2 = ”Teacher”;
                                        break;
                                    case ”Teacher”:
```

```
                            otherUserType2 = ”Student”;
                            break;
                    }
                }
            }
            if (otherUserType2.equals(”Teacher”)) {

                adapter = ArrayAdapter.createFromResource(
                    MainActivity.this,
                        R.array.planets_array, android.R.layout.
                            simple_spinner_item);
                adapter.setDropDownViewResource(android.R.layout.
                    simple_spinner_dropdown_item);
                dropdown.setAdapter(adapter);
                fistSearch = ”Up to 9th”;
                if (FirebaseAuth.getInstance().getCurrentUser() !=
                    null) {
                    getCostomerType(fistSearch.toLowerCase());
                }

            } else {

                adapter = ArrayAdapter.createFromResource(
                    MainActivity.this,
                        R.array.planets_array2, android.R.layout.
                            simple_spinner_item);
                adapter.setDropDownViewResource(android.R.layout.
                    simple_spinner_dropdown_item);
                dropdown.setAdapter(adapter);
                fistSearch = ”student”;
                if (FirebaseAuth.getInstance().getCurrentUser() !=
                    null) {
                    getCostomerType(fistSearch.toLowerCase());
                }
            }
        }

        @Override
```

```
                    public void onCancelled(@NonNull DatabaseError databaseError) {
                    }
                });
        arrayAdapter = new arrayAdapter(this, R.layout.item, al);
        search = findViewById(R.id.search);

        SwipeFlingAdapterView flingContainer = (SwipeFlingAdapterView) findViewById(R.id.frame);
        dropdown.setPrompt(”Search for?”);
        dropdown.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
                local = (String) parent.getItemAtPosition(position);
            }
            @Override
            public void onNothingSelected(AdapterView<?> parent) {

            }
        });
        flingContainer.setAdapter(arrayAdapter);
        flingContainer.setFlingListener(new SwipeFlingAdapterView.onFlingListener() {
            @Override
            public void removeFirstObjectInAdapter() {
                 (/AdapterView)
                Log.d(”LIST”, ”removed object!”);
                al.remove(0);
                arrayAdapter.notifyDataSetChanged();
            }
            @Override
            public void onLeftCardExit(Object dataObject) {
                if(!isConnected()){
                    new AlertDialog.Builder(MainActivity.this)
                            .setIcon(android.R.drawable.ic_dialog_alert)
                            .setTitle(”Internet Connection Error!!!”)
                            .setMessage(”Please Check you internet Connection”)
                            .setPositiveButton(”Close”, new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog, int which) {
                                    finish();
                                }
                            }).show();
                }
                {
                    if (currentUserType.equals(”Student”)) {

                      Toast.makeText(MainActivity.this, ”Not needed now”, Toast.LENGTH_SHORT).show();
                    }
                }
            }

            @Override
            public void onRightCardExit(Object dataObject) {
                if(!isConnected()){
                    new AlertDialog.Builder(MainActivity.this)
                            .setIcon(android.R.drawable.ic_dialog_alert)
                            .setTitle(”Internet Connection Error!!!”)
                            .setMessage(”Please Check you internet Connection”)
                            .setPositiveButton(”Close”, new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog, int which) {
                                    finish();
                                }
                            }).show();
                }else {
                }
            }

            @Override
            public void onAdapterAboutToEmpty(int itemsInAdapter) {
```

```
                if(itemsInAdapter <=0){
                    sorryText.setVisibility(View.VISIBLE);
                }else{
             sorryText.setVisibility(View.INVISIBLE);
                }
            }

            @Override
            public void onScroll(float scrollProgressPercent) {
            }
        });
        flingContainer.setOnItemClickListener(new SwipeFlingAdapterView.OnItemClickListener() {
            @Override
            public void onItemClicked(int itemPosition, Object dataObject) {

                if(otherUserType2.equals(”Teacher”)){

                    Intent it=new Intent(MainActivity.this,AboutInfo.class);

                    cards c = (cards) dataObject;
                    String uid = c.getUid();

                    it.putExtra(”card-uid”,uid);
                    it.putExtra(”Card-Info-name”,al.get(itemPosition).name);
                    it.putExtra(”Card-Info-job”,al.get(itemPosition).job);
                    it.putExtra(”Card-Info-locality”,al.get(itemPosition).locality);
                    it.putExtra(”Card-Info-cardImage”,al.get(itemPosition).cardImage);
                    it.putExtra(”Card-Info-teachings”,al.get(itemPosition).teachings);
                    it.putExtra(”Card-Info-timing”,al.get(itemPosition).timing);
                    it.putExtra(”Card-Info-uid”,al.get(itemPosition).uid);
                    it.putExtra(”Card-Info-number”,al.get(itemPosition).number);
                    it.putExtra(”Card-Info-seatCount”,al.get(itemPosition).seat_count);
                    it.putExtra(”Card-Info-about”,al.get(itemPosition).about);
                    it.putExtra("Card-Info-locationType",al.get(itemPosition).locationType);

                    startActivity(it);

                }else{
                    Toast.makeText(MainActivity.this,"Access restricted",Toast.LENGTH_SHORT).show();

                }


            }
        });

        search.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (!isConnected()) {
                    new AlertDialog.Builder(MainActivity.this)
                            .setIcon(android.R.drawable.ic_dialog_alert)
                            .setTitle("Internet Connection Error!!!")
                            .setMessage("Please Check you internet Connection")
                            .setPositiveButton("Close", new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog, int which) {
                                    finish();
                                }
                            }).show();

                } else {
                    al.clear();
                    arrayAdapter.notifyDataSetChanged();
                    getCostomerType(local.toLowerCase());


```

```java
                    dropdown.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
                        @Override
                        public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
                            local = (String) parent.getItemAtPosition(position);


                        }

                        @Override
                        public void onNothingSelected(AdapterView<?> parent) {

                        }
                    });


                }
            }
        });
    }

    static void makeToast(Context ctx, String s) {
        Toast.makeText(ctx, s, Toast.LENGTH_SHORT).show();
    }
    @Override
    protected void onStart() {
        super.onStart();
        FirebaseUser firebaseUser = FirebaseAuth.getInstance().getCurrentUser();

        if (firebaseUser == null) {
            Intent intent = new Intent(MainActivity.this, LoginActivity.class);
            startActivity(intent);
            finish();
        }
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater=getMenuInflater();
        inflater.inflate(R.menu.menu,menu);
        return true;

    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case R.id.account_settings_btn:
                Intent intent=new Intent(MainActivity.this,SettingsActivity.class);
                startActivity(intent);
                return true;
            case R.id.btnContact:
                Intent i2=new Intent(MainActivity.this,MatchActivity.class);
                startActivity(i2);
                return true;
            case R.id.btn_refresh:
                finish();
                startActivity(getIntent());
                return true;
            case R.id.logOut_btn:
                FirebaseAuth.getInstance().signOut();
                Intent i=new Intent(MainActivity.this,LoginActivity.class);
                startActivity(i);
                finish();
                return true;
            case R.id.helpBtn:
                Intent hi=new Intent(MainActivity.this,HelpPageActivity.class);
                startActivity(hi);
            default:
                return false;
        }
    }

    private String currentUserType;
    private String otherUserType;
    private String toBeSearchedPlace;

    public void getCostomerType(String temp){
        searchedProfessioni=temp;


        DatabaseReference databaseReference= FirebaseDatabase.getInstance().
            getReference().child(”User”)
                .child(FirebaseAuth.getInstance().getCurrentUser().getUid());
        databaseReference.addListenerForSingleValueEvent(new ValueEventListener
            () {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                    if(dataSnapshot.exists()){
                        if(dataSnapshot.child(”userType”)!=null){
                            currentUserType=dataSnapshot.child(”userType”).
                                getValue().toString();
                            AboutInfo.getUserType(currentUserType);
                            switch (currentUserType){
                                case”Student”:
                                    otherUserType=”Teacher”;
                                    break;
                                case”Teacher”:
                                    otherUserType=”Student”;
                                    break;
                            }
                        }toBeSearchedPlace=dataSnapshot.child(”number”).getValue
                            ().toString()
                                .toLowerCase().trim();
                        generatepeople(searchedProfessioni);
                    }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    public void generatepeople(String temp){
```

```
final String tempLast=temp;
FirebaseDatabase.getInstance().getReference().child(”User”).
    addChildEventListener(new ChildEventListener() {
@Override
public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable
    String s) {
    if(dataSnapshot.child(”userType”).getValue()!=null) {
        if (dataSnapshot.exists()
              && dataSnapshot.child(”userType”).getValue().toString().
                  equals(otherUserType)
                && dataSnapshot.child(”number”).getValue().toString().
                    toLowerCase().equals(toBeSearchedPlace)
                && dataSnapshot.child(”job”).getValue().toString().
                    toLowerCase().trim().equals(tempLast)
        ){
            cards item = new cards(
                    dataSnapshot.child(”name”).getValue().toString(),
                    dataSnapshot.getKey(),
                    dataSnapshot.child(”imgUri”).getValue().toString(),
                    dataSnapshot.child(”job”).getValue().toString(),
                    dataSnapshot.child(”number”).getValue().toString(),
                    dataSnapshot.child(”about”).getValue().toString(),
                    dataSnapshot.child(”teachings”).getValue().toString
                        (),
                    dataSnapshot.child(”locality”).getValue().toString()
                        ,
                    dataSnapshot.child(”locationType”).getValue().
                        toString(),
                    dataSnapshot.child(”timing”).getValue().toString(),
                    dataSnapshot.child(”seat_count”).getValue().toString
                        ()
                    );
            al.add(item);
            arrayAdapter.notifyDataSetChanged();
        }
    }
}
@Override
```

```
        public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

        }

        @Override
        public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

        }

        @Override
        public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
    }
}
```

**Test result**

The main activity gets imported with proper card view layout as a proper array adapter functions. And the filters works perfectly.

### 5.3.3 Functional testing

**Input**

```
public class ChatActivity extends AppCompatActivity {

    public static String userType;
    private RecyclerView rv;
    String matchId, chatId;

    EditText message;
    ImageView sendBtn;
    Toolbar chat_toolbar;
    List<Address> addresses;
    FloatingActionButton locationBtn;
    RatingBar ratingbar;
    String nameOfPerson;
    static String usernameForChat;
    public static final int REQUEST_CODE_LOCATION_PERMISSION = 1;
    private RecyclerView.Adapter mChatAdapter;
    private RecyclerView.LayoutManager mChatLayoutManger;
    DatabaseReference mDatabaseRef, mDatabaseChat;
    private AdView mAdView_3;
    public static void setUserTpe(String s) {
        userType = s;
        Log.i(”type--”, s);
    }
    public boolean isConnected() {
        ConnectivityManager connectivityManager = (ConnectivityManager)
            getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = connectivityManager.getActiveNetworkInfo();
        return networkInfo != null && networkInfo.isConnected();
    }

    @SuppressLint(”RestrictedApi”)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_chat);
        chat_toolbar = findViewById(R.id.chat_toolbar);
        locationBtn = findViewById(R.id.locationBtn);

        if (userType.equals(”Student”)) {
            locationBtn.setImageResource(R.drawable.ic_call_black_24dp);
        }else{
            locationBtn.setImageResource(R.drawable.my_location1);
        }
        mAdView_3 = findViewById(R.id.adView_chat);
        AdRequest adRequest = new AdRequest.Builder().build();
        mAdView_3.loadAd(adRequest);
        matchId = getIntent().getExtras().getString(”matchId”);
        FirebaseDatabase.getInstance().getReference()
                .child(”User”).child(matchId).addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                if (dataSnapshot.exists()) {
                    nameOfPerson=dataSnapshot.child(”name”).getValue().toString();
                    usernameForChat=nameOfPerson;
                    chat_toolbar.setTitle(nameOfPerson);
                    if(true){
                        chat_toolbar.setSubtitle(”Online”);
                    }else{
                        chat_toolbar.setSubtitle(”Offline”);
                    }
                    chat_toolbar.setTitleTextColor(Color.WHITE);
                    chat_toolbar.setSubtitleTextColor(Color.WHITE);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
        mDatabaseRef = FirebaseDatabase.getInstance().getReference()
                .child(”User”)
                .child(FirebaseAuth.getInstance().getCurrentUser().getUid())
                .child(”connections”)
                .child(”match”)
                .child(matchId)
                .child(”Chat_Id”);
        mDatabaseChat = FirebaseDatabase.getInstance().getReference()
                .child(”Chat”);
```

```java

        getChatId();

        rv = findViewById(R.id.chatRecyclerView);

        rv.setNestedScrollingEnabled(false);
        rv.setHasFixedSize(false);
        mChatLayoutManger = new LinearLayoutManager(ChatActivity.this);
        rv.setLayoutManager(mChatLayoutManger);
        mChatAdapter = new ChatAdapter(getDataSetChats(), ChatActivity.this);
        rv.setAdapter(mChatAdapter);




        message = findViewById(R.id.chatText);
        sendBtn = findViewById(R.id.sendBtn);

        sendBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (!isConnected()) {
                    new AlertDialog.Builder(ChatActivity.this)
                            .setIcon(android.R.drawable.ic_dialog_alert)
                            .setTitle("Internet Connection Error!!!")
                            .setMessage("Please Check you internet Connection")
                            .setPositiveButton("Close", new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog, int which) {
                                    finish();
                                }
                            }).show();

                } else {
                    if (!TextUtils.isEmpty(message.getText().toString())) {
                        sendMessage();
                    }
                }
            }
        });

        findViewById(R.id.locationBtn).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (!isConnected()) {
                    new AlertDialog.Builder(ChatActivity.this)
                            .setIcon(android.R.drawable.ic_dialog_alert)
                            .setTitle(”Internet Connection Error!!!”)
                            .setMessage(”Please Check you internet Connection”)
                            .setPositiveButton(”Close”, new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog, int which) {
                                    finish();
                                }
                            }).show();

                } else {
                    if (ContextCompat.checkSelfPermission(
                            getApplicationContext(), Manifest.permission.ACCESS_FINE_LOCATION)
                            != PackageManager.PERMISSION_GRANTED) {
                        ActivityCompat.requestPermissions(
                                ChatActivity.this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                                REQUEST_CODE_LOCATION_PERMISSION
                        );
                    } else {
                        if(userType.equals(”Student”)){
                            callTheUser();
                        }
                        else{getCurrentLcoation();}

                    }
```

```java
                }

            }
        });


    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            if(userType.equals("Student")){
                callTheUser();
            }
            else{getCurrentLcoation();}
        } else {
            if(userType.equals("Student")){
                callTheUser();
            }
            else{getCurrentLcoation();}
        }
    }

    private void callTheUser() {
        String phone = "+34666777888";
        Intent intent = new Intent(Intent.ACTION_DIAL, Uri.fromParts("tel", phone, null));
        startActivity(intent);
    }

    public void getCurrentLcoation() {


        final LocationRequest locationRequest = new LocationRequest();
        locationRequest.setInterval(10000);
        locationRequest.setFastestInterval(3000);
        locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {

            return;
        }
        LocationServices.getFusedLocationProviderClient(ChatActivity.this)

                .requestLocationUpdates(locationRequest, new LocationCallback() {
                    @Override
                    public void onLocationResult(LocationResult locationResult) {
                        super.onLocationResult(locationResult);
                        LocationServices.getFusedLocationProviderClient(ChatActivity.this)
                                .removeLocationUpdates(this);
                        if (locationResult != null && locationResult.getLocations().size() > 0) {
                            int latestLocationIndex = locationResult.getLocations().size() - 1;
                            double lattitide =
                                    locationResult.getLocations().get(latestLocationIndex).getLatitude();
                            double lontitude =
                                    locationResult.getLocations().get(latestLocationIndex).getLongitude();
                            Geocoder geocoder = new Geocoder(ChatActivity.this, Locale.getDefault());
                            try {
                                addresses = geocoder.getFromLocation(lattitide, lontitude, 1);
                            } catch (IOException e) {
                                e.printStackTrace();
                            }
                            String address = addresses.get(0).getAddressLine(0);
                                getMaxAddressLineIndex()
                            String city = addresses.get(0).getLocality();
                            String state = addresses.get(0).getAdminArea();

                            message.setText(””);
                            message.setText(”http://www.google.com/maps/place/”
                               + address + ”,” + city + ”,” + state);


                        }
                    }
                }, Looper.getMainLooper());


    }

    private void sendMessage() {
        DatabaseReference newMsgDb = mDatabaseChat.push();
        Map newMessage = new HashMap();
        newMessage.put(”createdByUser”, FirebaseAuth.getInstance().
            getCurrentUser().getUid());
        newMessage.put(”text”, message.getText().toString());
        newMessage.put(”timeStamp”, ServerValue.TIMESTAMP);
        newMsgDb.setValue(newMessage).addOnSuccessListener(new OnSuccessListener
            <Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                message.setText(””);


            }
        });


    }

    private void getChatId() {
        mDatabaseRef.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                if (dataSnapshot.exists()) {
                    chatId = dataSnapshot.getValue().toString();
                    mDatabaseChat = mDatabaseChat.child(chatId);
                    getChatMsg();
                }


            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {


            }
        });
    }

    private void getChatMsg() {
        mDatabaseChat.addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

                if (dataSnapshot.exists()) {

                    String msg = null;
                    String createdBy = null;
                    if (dataSnapshot.child(”text”).getValue() != null) {
                        msg = dataSnapshot.child(”text”).getValue().toString();
                    }
                    if (dataSnapshot.child(”createdByUser”).getValue() != null) {
                        createdBy = dataSnapshot.child(”createdByUser”).getValue().toString();

                    }

                    Boolean istThisCurrentUser = false;
                    if (createdBy.equals(FirebaseAuth.getInstance().
                        getCurrentUser().getUid())) {
                        istThisCurrentUser = true;
                    }
                    ChatObject newMsg = new ChatObject(msg, istThisCurrentUser);
                    Chatlist.add(newMsg);
                    mChatAdapter.notifyDataSetChanged();
                }

            }

            @Override
            public void onChildChanged(@NonNull DataSnapshot dataSnapshot,
                @Nullable String s) {

            }

            @Override
            public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

            }

            @Override
            public void onChildMoved(@NonNull DataSnapshot dataSnapshot,
                @Nullable String s) {

            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private ArrayList<ChatObject> Chatlist = new ArrayList<ChatObject>();

    private List<ChatObject> getDataSetChats() {
        return Chatlist;
    }

    public void ShowReviewDialog(View view) {
        LayoutInflater layoutInflater = null;
        final AlertDialog.Builder builder= new AlertDialog.Builder(ChatActivity.this);
        View v2= LayoutInflater.from(ChatActivity.this). inflate(R.layout.rating_tutor_dialog,null);
        builder.setTitle(”Rate ”+nameOfPerson);
        builder.setView(v2);

        ratingbar=(RatingBar)v2.findViewById(R.id.ratingBar1);




        builder.setPositiveButton(”Submit”, new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int whichButton) {


                String rating=String.valueOf(ratingbar.getRating());
                Toast.makeText(getApplicationContext(), rating, Toast.LENGTH_LONG).show();
            }
        })
                .setNegativeButton(”Cancel”, new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int whichButton) {
                        dialog.cancel();
                    }
                });

        builder.show();
    }
}
```

**Test Result**

The GPS module from google api and Phone caller module works perfectly to share the current location and make phone call accordingly to the respective user types.

### 5.3.4 White Box Testing

**Inuput**

```java
public class MatchActivity extends AppCompatActivity {
    private RecyclerView rv;
    private RecyclerView.Adapter mMatchesAdapter;
    private RecyclerView.LayoutManager mMatchesLayoutManger;
    private AdView mAdView_2;
    public TextView NoConText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_match);

        mAdView_2 = findViewById(R.id.adView_match);
        AdRequest adRequest = new AdRequest.Builder().build();
        mAdView_2.loadAd(adRequest);
        rv=findViewById(R.id.matchRecyclerView);
        rv.setNestedScrollingEnabled(false);
        rv.setHasFixedSize(true);
        mMatchesLayoutManger=new LinearLayoutManager(MatchActivity.this);
        rv.setLayoutManager(mMatchesLayoutManger);
        mMatchesAdapter=new MatchAdapter(getDataSetMatches(),MatchActivity.this);
        rv.setAdapter(mMatchesAdapter);
        NoConText=findViewById(R.id.NoConText);
        getUserMatch();
    }
    static void getUserType(String s){
        ChatActivity.setUserTpe(s);
    }

    private void getUserMatch() {
        DatabaseReference usermatchRef= FirebaseDatabase.getInstance().getReference()
                .child(”User”)
                .child(FirebaseAuth.getInstance().getCurrentUser().getUid())
                .child(”connections”)
                .child(”match”);
        usermatchRef.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                if(dataSnapshot.exists()){
                    for(DataSnapshot match:dataSnapshot.getChildren()){
                        fetchMatchInfo(match.getKey());
                    }
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private void fetchMatchInfo(String key) {
        DatabaseReference matchdetails= FirebaseDatabase.getInstance().getReference()
                .child(”User”)
                .child(key);
        matchdetails.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                if(dataSnapshot.exists()){
                    String userId=dataSnapshot.getKey();
                    String userName=””;
                    String userImage=””;
                    String userJob=””;
                    String number=””;
            String teachings=””;
            String locality=””;

            if(dataSnapshot.child(”name”).getValue()!=null){
                userName=dataSnapshot.child(”name”).getValue().toString();
            }
            if(dataSnapshot.child(”imgUri”).getValue()!=null){
                userImage=dataSnapshot.child(”imgUri”).getValue().toString();
            }
      if(dataSnapshot.child(”job”).getValue()!=null){
userJob=dataSnapshot.child(”job”).getValue().toString();
            }
     if(dataSnapshot.child(”number”).getValue()!=null){
                number=dataSnapshot.child(”number”).getValue().toString();
            }
            if(dataSnapshot.child(”teachings”).getValue()!=null){
                teachings=dataSnapshot.child(”teachings”).getValue().toString();
            }
            if(dataSnapshot.child(”locality”).getValue()!=null){
                locality=dataSnapshot.child(”locality”).getValue().toString();
            }

            MatchObjects obj=new MatchObjects(userId,userName,userImage,userJob,number,teachings,locality);
            list.add(obj);
            if(list.size()<1){NoConText.setVisibility(View.VISIBLE);}else{
                NoConText.setVisibility(View.INVISIBLE);
            }
            mMatchesAdapter.notifyDataSetChanged();
        }
    }

    @Override
```

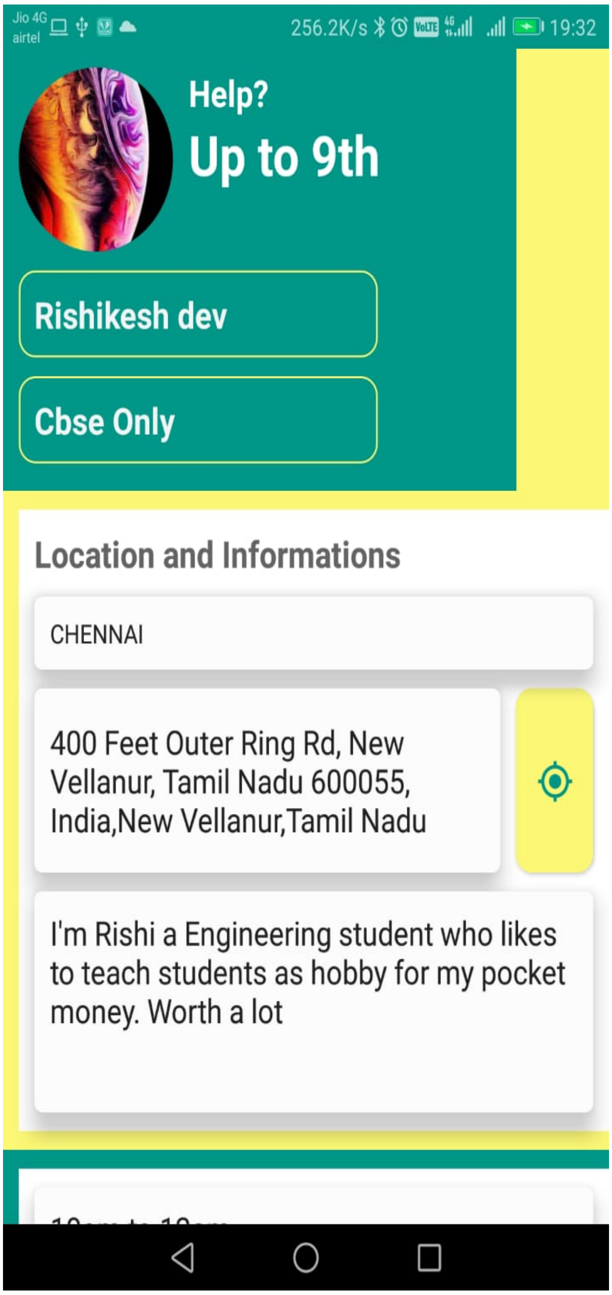
```java
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }

    private ArrayList<MatchObjects>list=new ArrayList<MatchObjects >();
    private List<MatchObjects> getDataSetMatches() {
        return list;
    }
}
```

### 5.3.5 Black Box Testing

#### Input

```java
package com.tO.sociohub;

public class cards {
    public String name;
    public String uid;
    public String cardImage;
    public String job;
    public String number;
    public String about;
    public String teachings,locality,locationType,timing,seat_count;


    public cards(String name, String uid, String cardImage, String job, String number, String about,
    String teachings, String locality, String locationType, String timing, String seat_count) {
        this.name = name;
        this.uid = uid;
        this.cardImage = cardImage;
        this.job=job;
        this.number=number;
        this.about=about;
        this.teachings = teachings;
        this.locality = locality;
        this.locationType = locationType;
        this.timing = timing;
        this.seat_count = seat_count;
    }

    public String getJob() {
        return job;
    }

    public void setJob(String job) {
        this.job = job;
    }

    public String getNumber() {
        return number;
    }

    public void setNumber(String number) {
        this.number = number;
    }

    public String getAbout() {
        return about;
    }

    public void setAbout(String about) {
        this.about = about;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getUid() {
        return uid;
    }

    public void setUid(String uid) {
        this.uid = uid;
    }
    public String getCardImage() {
        return cardImage;
    }

    public void setCardImage(String cardImage) {
        this.cardImage = cardImage;
    }
    public String getTeachings() {
        return teachings;
    }

    public void setTeachings(String teachings) {
        this.teachings = teachings;
    }

    public String getLocality() {
        return locality;
    }

    public void setLocality(String locality) {
        this.locality = locality;
    }

    public String getLocationType() {
        return locationType;
    }

    public void setLocationType(String locationType) {
        this.locationType = locationType;
    }

    public String getTiming() {
        return timing;
    }

    public void setTiming(String timing) {
        this.timing = timing;
    }

    public String getSeat_count() {
        return seat_count;
    }

    public void setSeat_count(String seat_count) {
        this.seat_count = seat_count;
    }
}


Test Output:connected and able to retrieve data
Expected Output:proper connection and able to manipulate/view data
```

### 5.3.6 Test Result



Help?

Up to 9th

Rishikesh dev

Cbse Only

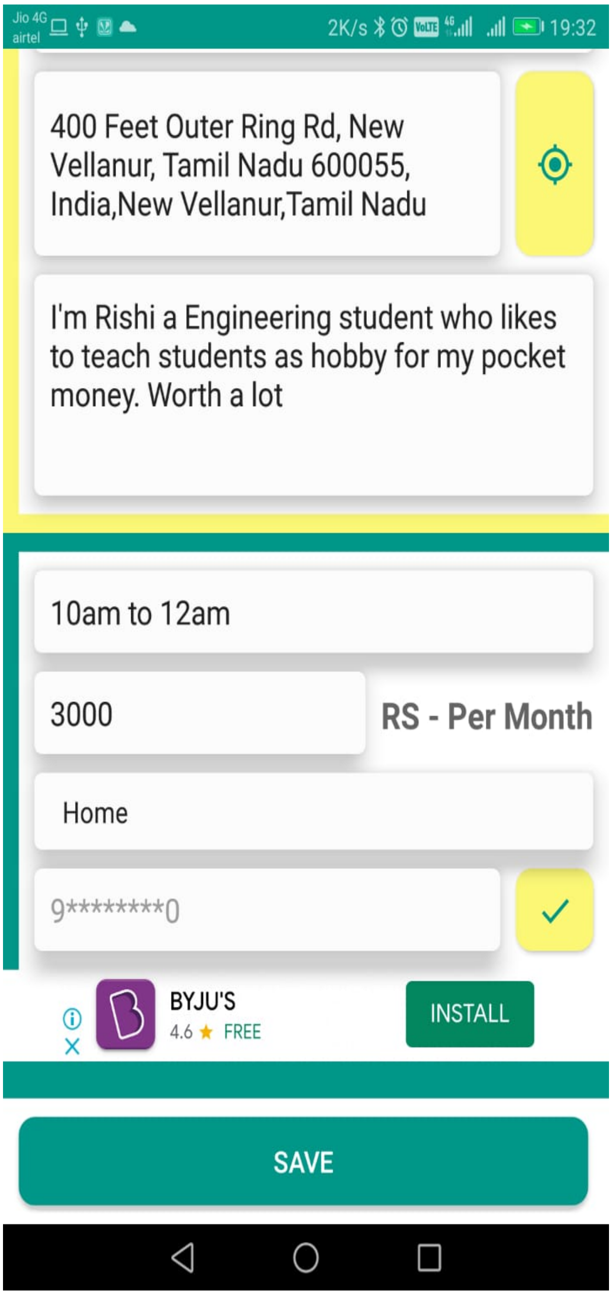
Location and Informations

CHENNAI

400 Feet Outer Ring Rd, New Vellanur, Tamil Nadu 600055, India,New Vellanur,Tamil Nadu

I'm Rishi a Engineering student who likes to teach students as hobby for my pocket money. Worth a lot

Figure 5.5: **User settings activity-1**



400 Feet Outer Ring Rd, New Vellanur, Tamil Nadu 600055, India,New Vellanur,Tamil Nadu

I'm Rishi a Engineering student who likes to teach students as hobby for my pocket money. Worth a lot

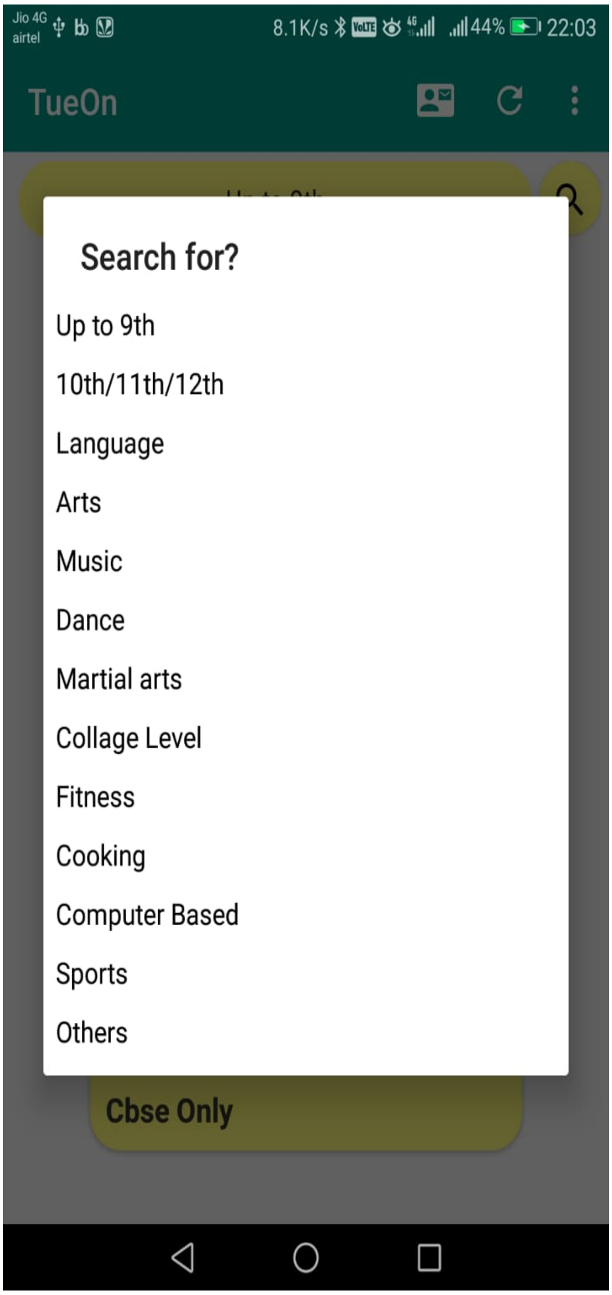
10am to 12am

3000 RS - Per Month

Home

9********0 ✓

Figure 5.6: **User settings activity-2**



Jio 4G airtel

8.1K/s 44% 22:03

TueOn

Search for?

Up to 9th

10th/11th/12th

Language

Arts

Music

Dance

Martial arts

Collage Level

Fitness

Cooking

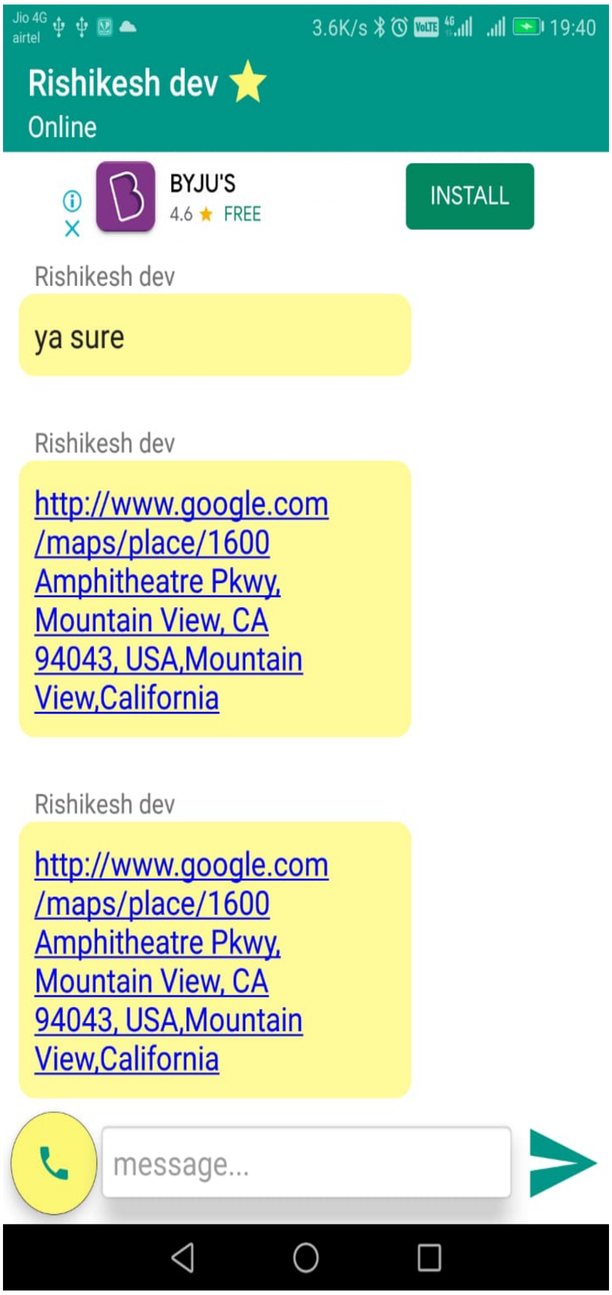
Computer Based

Sports

Others

Cbse Only

Figure 5.7: **Type Filter**



Rishikesh dev

Online

Rishikesh dev

ya sure

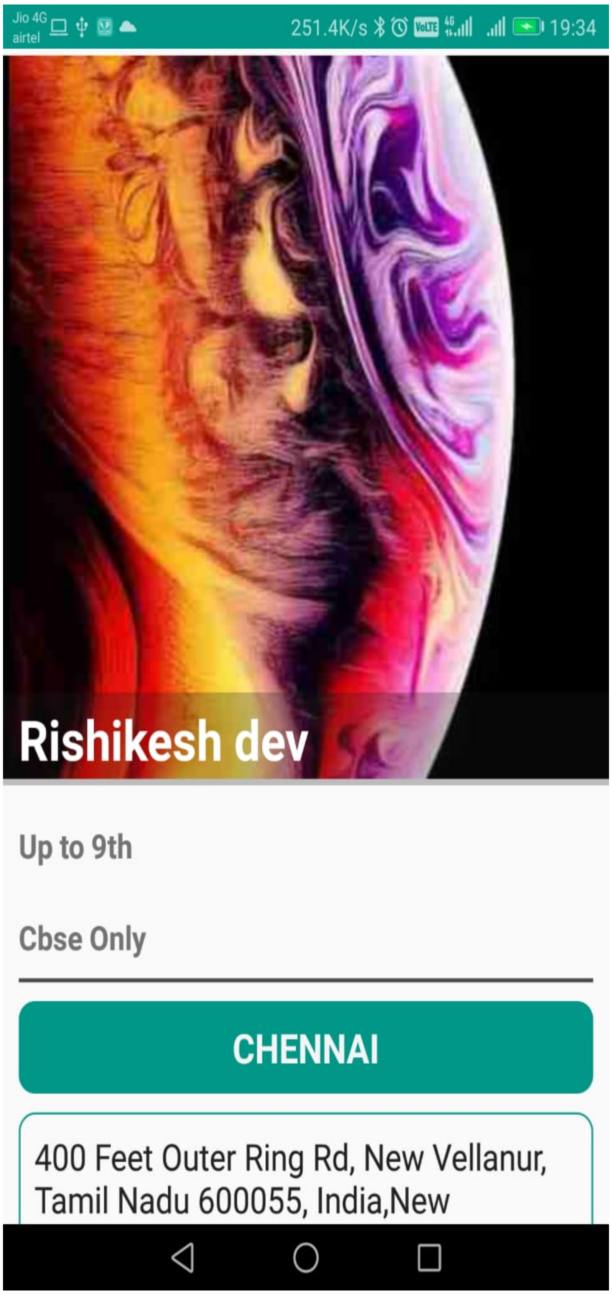
Rishikesh dev

http://www.google.com/maps/place/1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA,Mountain View,California

Rishikesh dev

http://www.google.com/maps/place/1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA,Mountain View,California

Figure 5.8: **Location sharing**

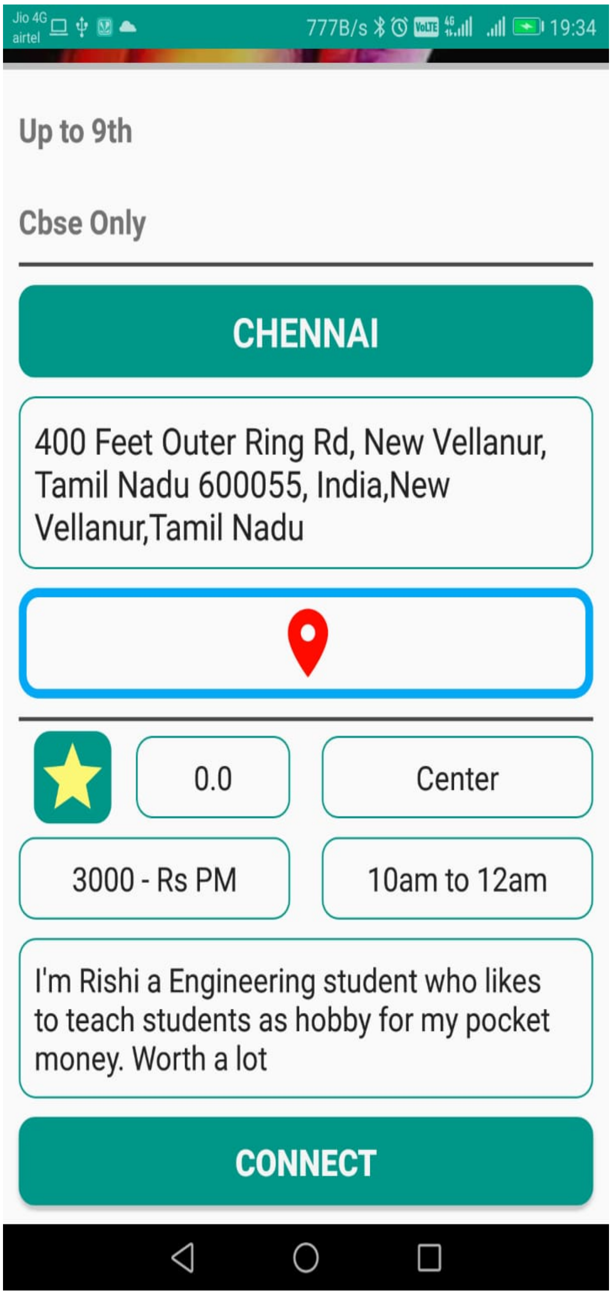


Up to 9th

Cbse Only

400 Feet Outer Ring Rd, New Vellanur, Tamil Nadu 600055, India,New

Figure 5.9: **Tuition profile 1**



Up to 9th

Cbse Only

400 Feet Outer Ring Rd, New Vellanur, Tamil Nadu 600055, India,New Vellanur,Tamil Nadu

0.0

Center

3000 - Rs PM

10am to 12am

I'm Rishi a Engineering student who likes to teach students as hobby for my pocket money. Worth a lot

Figure 5.10: **Tuition profile 2**

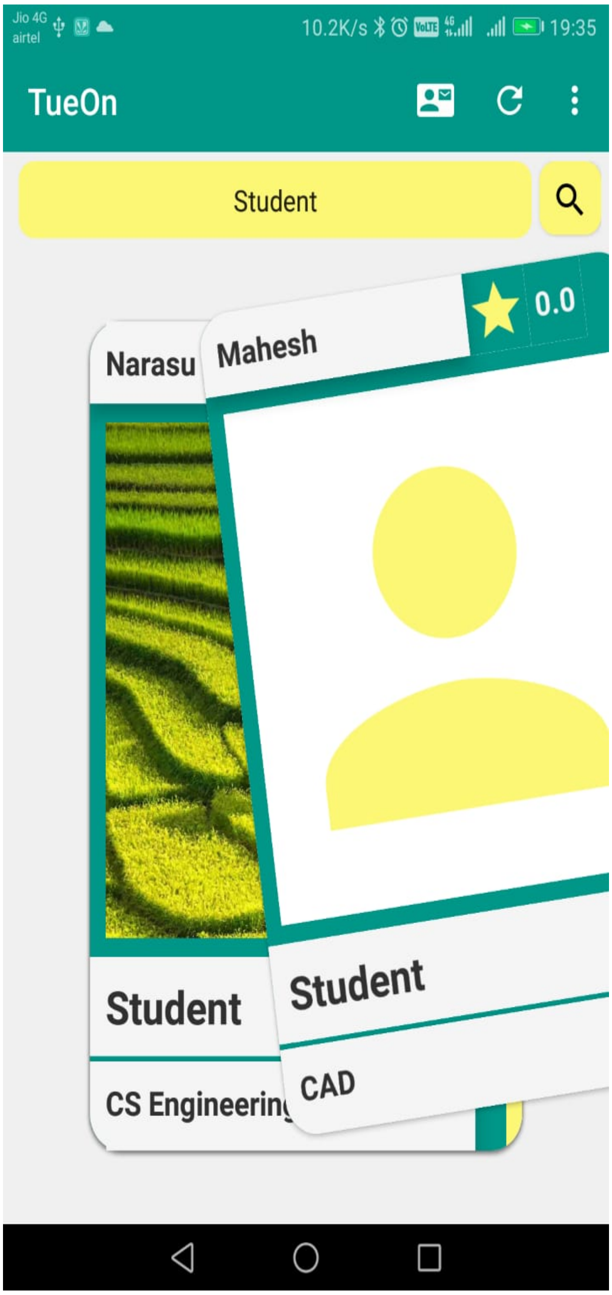
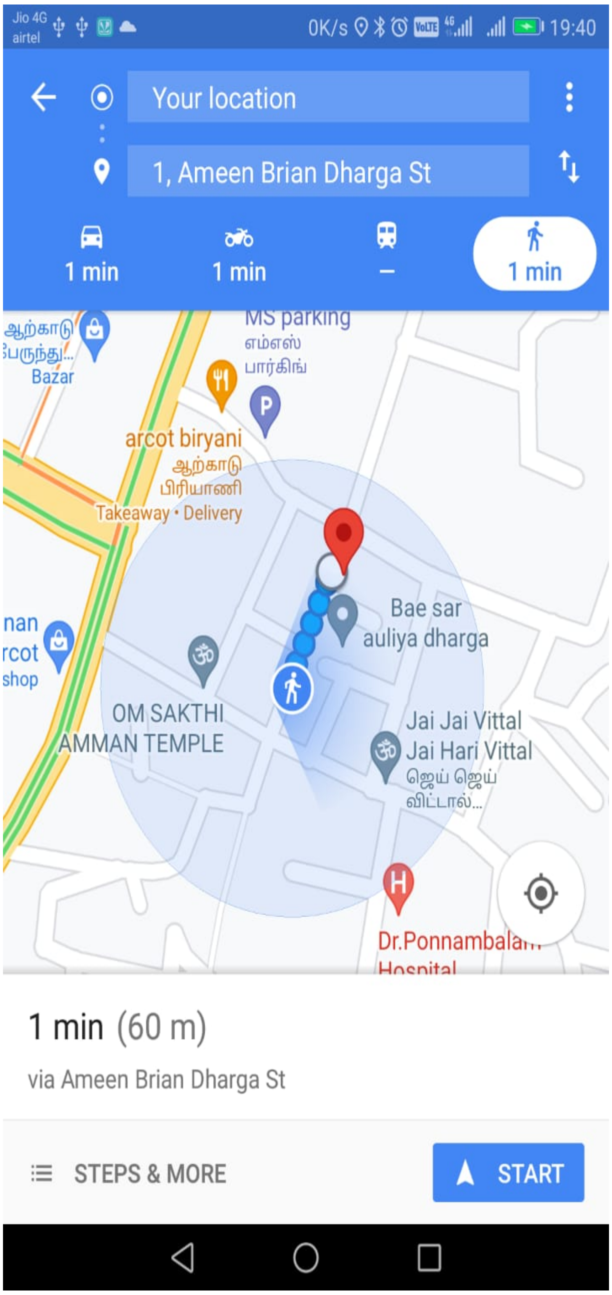


Figure 5.11: **Filtered Card View**



Your location

1, Ameen Brian Dharga St

1 min (60 m)

via Ameen Brian Dharga St

Figure 5.12: **Google maps Direction**

## 5.4 Testing Strategy

The strategy for this mobile application is a well-planned step that resulted in the successful construction of the project. The testing strategy must co-operate with test planning including the resultant data collection and evaluation. A strategy for software testing must accommodate low-level tests needed to verify that a small code segment has been implemented correctly. Thus, a series of testing is performed for the proposed system before its ready for roll out.

# Chapter 6

# RESULTS AND DISCUSSIONS

## 6.1 Efficiency of the Proposed System

A multi-layered system provides knowledge about Android technology connecting the modern application and traditional communication area of architecture. They have adopted the In-app punches system concept and embedded within the app to design the monetize the profit for the application. The architecture of the application is feasible in all possible states depending upon the users. The use of GPS is well associated with Google Map API. Latitude and latitude sensors are used for location sensing. Phone call only for the users are allowed in more information were to be sleeked.

## 6.2 Comparison of Existing and Proposed System.

The main difference comes between the cost expense becomes comply null in the proposed system, where in the existing system in-app punches play a major role in the source of incomes for the application. All third parties users can open an account free of any cost wherein existing system only big organization can stand a chance in the market. The review system is limited for certain users and cannot be misused to any possible extent but where in the existing system there is a chance for it to go wrong.

## 6.3 Advantages of the Proposed System

- Free for users.
- User friendly
- Safe proof review system
- Available for all third parties

## 6.4 Sample Code

```
2package com.tO.sociohub;

import android.content.Context;
import android.net.Uri;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;

import java.util.List;

public class MatchAdapter extends RecyclerView.Adapter<MatchViewHolder> {

    private List<MatchObjects>list;
    private Context context;

    public MatchAdapter(List<MatchObjects>list,Context context){
        this.list=list;
        this.context=context;
```

```java
    }

    @NonNull
    @Override
    public MatchViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View layoutView= LayoutInflater.from(parent.getContext()).inflate(R.layout.item_match,null,false);
        RecyclerView.LayoutParams lp=new RecyclerView.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT);
        layoutView.setLayoutParams(lp);
        MatchViewHolder rcv=new MatchViewHolder((layoutView));
        return rcv;
    }

    @Override
    public void onBindViewHolder(@NonNull MatchViewHolder holder, int position) {
        holder.userId.setText(list.get(position).getUid());
        holder.userName.setText(list.get(position).getName());
        holder.userCity.setText(list.get(position).getNumber());
        holder.userJob.setText(list.get(position).getUserJob());
        holder.local.setText(list.get(position).getLocality());
        holder.matchlearnin.setText(list.get(position).getTeachings());
        Glide.with(context).load(Uri.parse(list.get(position).getImgUri()))
                .placeholder(R.drawable.ic_person_2)
                .into(holder.MatchImage);


    }

    @Override
    public int getItemCount() {
        return list.size();
    }
}
```

# Chapter 7

# CONCLUSION AND FUTURE ENHANCEMENTS

## 7.1 Conclusion

There are several apps for finding tuition centers and doing online courses with their pros and cons, but most of them are privatized. This application is free and can be used by any individual with a mobile number and e-mail id. But unlike other apps, this app's database is managed by a Firebase algorithm. Considering the validity of this project might have great market usage.

## 7.2 Future Enhancements

The number of active users is limited for the database free trials usage in the future it is expected to expand the database so the number of users can increase long with the data storage capacity. The usage of this app is also limited to only android users so in future we create a replica of this application. The limitation of distracts are attached with the nation India but in the future, the application might have the potential to fall under multinational usage. .

# References

[1] Biswajeet Sethi, Samaresh Mishra, Prasant ku. Patnaik (2018) ,A Study of NoSQL Database, School of Computer Engineering, KIIT University Bhubaneswar, India.

[2] Hana R. Esmaeel (2016) ,Apply Android Studio (SDK) Tools,Department of Inform. Comm. Engg., Al-Nahrain University, Iraq.

[3] Mrs. Prachi Sasankar1 . Mrs. Usha Kosarkar. 2 1 ( Prachi.sasankar@raisoni.net, BCA, Sadabai Raisoni Women's College,Nagpur SNDT Women's University,Mumbai , India.) 2 (Usha.kosarkar@raisoni.net , BCA,(2015) , Research on Development of Android ApplicationsG.H.R.I.I.T.Nagpur, R.T.M.Nagpur University,Nagpur.,India).

[4] Iqbaldeep Kaur, 2Navneet Kaur, 3Amandeep Ummat, 4Jaspreet Kaur, 5 Navjot Kaur (2012) ,Research Paper on Object Oriented Software Engineering, Dept. of CSE, Chandigarh Engineering College, Landran, Punjab, India.

[5] M. Bishop University of Pretoria Computer Science Department(2003) ,Java as a systems programming language: three case studies, Pretoria 0002, South

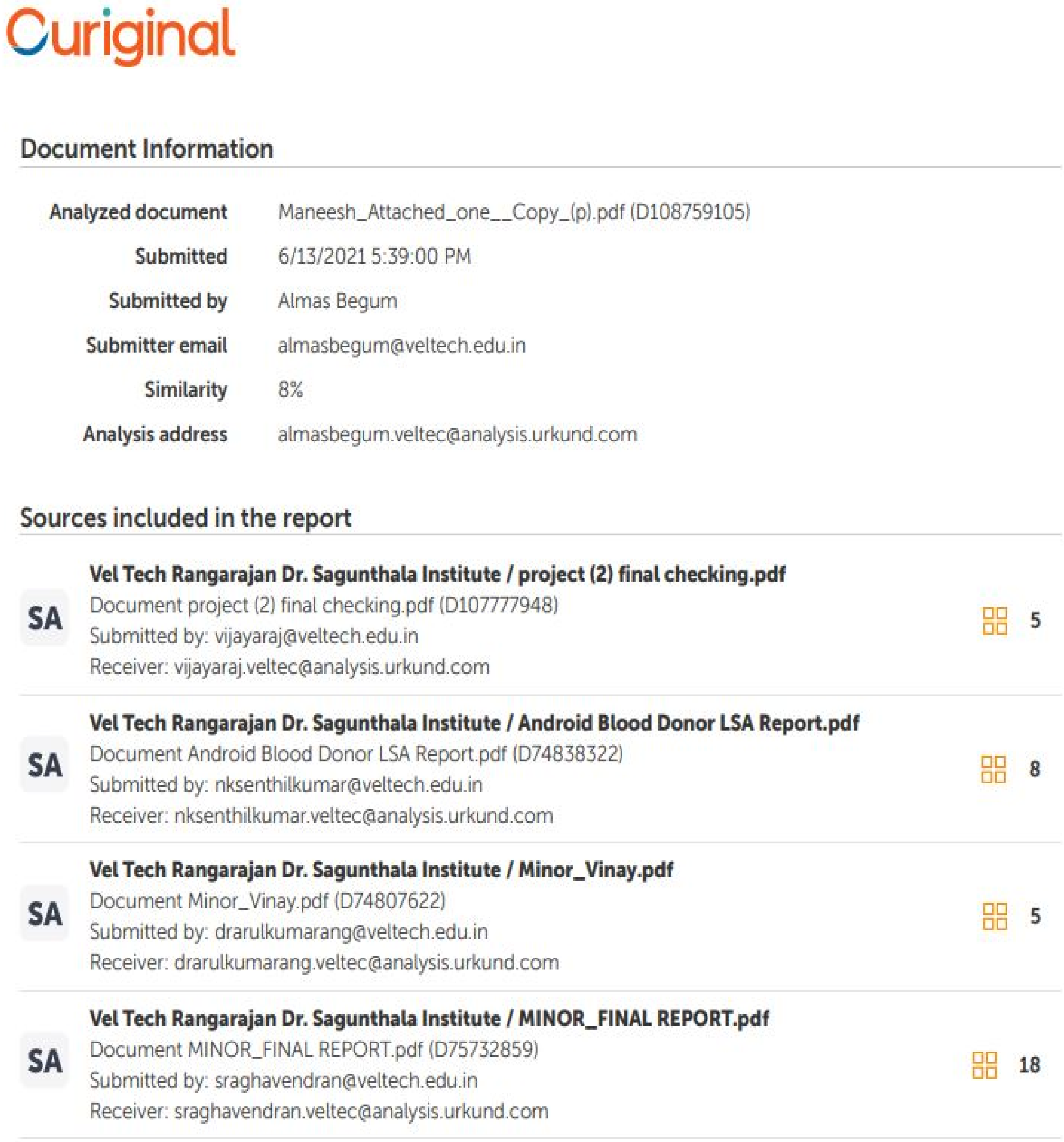
Africa.

[6] Ms. Niharika Dedhia1 , Dr. V. C. Kotak2 (2018) . ANDROID BASED CAMPUS SOLUTION FOR COLLEGE MANAGEMENT SYSTEM Dept. of Information Technology, Shah and Anchor Kutchhi Engineering College, Mumbai, India

[7] Sindh, Pakistan Zafarullah Sahito University of Eastern Finland Sukkur IBA University (2019), Role of Tuition Centers in the Performance and Achievement of Students· School of Applied Educational Science and Teacher Education Department of Education Postdoc.;

[8] . Prof. Dr. Rashad Yazdanifard Malaysia University of Science and Technology (2015) ·Online Marketing; the Impact of Mobile Application on Online Business Assc. School of Business Ph.D (Management).

[9] Associate Professor, Department of Information Science Engineering, Rajat Porwal, Srajan Singhal, Srijan ,Vaishnavi M UG Student (2014),ANDROID BASED COLLEGE APPLICATION FOR STUDENTS Department of Information Science Engineering, Acharya Institute of Technology, Bengaluru



Ouriginal

## Document Information

| | |
|---|---|
| Analyzed document | Maneesh_Attached_one__Copy_(p).pdf (D108759105) |
| Submitted | 6/13/2021 5:39:00 PM |
| Submitted by | Almas Begum |
| Submitter email | almasbegum@veltech.edu.in |
| Similarity | 8% |
| Analysis address | almasbegum.veltec@analysis.urkund.com |

## Sources included in the report

| | | |
|---|---|---|
| SA | **Vel Tech Rangarajan Dr. Sagunthala Institute / project (2) final checking.pdf**<br>Document project (2) final checking.pdf (D107777948)<br>Submitted by: vijayaraj@veltech.edu.in<br>Receiver: vijayaraj.veltec@analysis.urkund.com | 5 |
| SA | **Vel Tech Rangarajan Dr. Sagunthala Institute / Android Blood Donor LSA Report.pdf**<br>Document Android Blood Donor LSA Report.pdf (D74838322)<br>Submitted by: nksenthilkumar@veltech.edu.in<br>Receiver: nksenthilkumar.veltec@analysis.urkund.com | 8 |
| SA | **Vel Tech Rangarajan Dr. Sagunthala Institute / Minor_Vinay.pdf**<br>Document Minor_Vinay.pdf (D74807622)<br>Submitted by: drarulkumarang@veltech.edu.in<br>Receiver: drarulkumarang.veltec@analysis.urkund.com | 5 |
| SA | **Vel Tech Rangarajan Dr. Sagunthala Institute / MINOR_FINAL REPORT.pdf**<br>Document MINOR_FINAL REPORT.pdf (D75732859)<br>Submitted by: sraghavendran@veltech.edu.in<br>Receiver: sraghavendran.veltec@analysis.urkund.com | 18 |