Travel Assistant For Blinds : HELEN

Main Project Report

Submitted by

Angel Mary Joly

Reg.No: FIT20MCA-2025

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications
Of
A P J Abdul Kalam Technological University

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®
ANGAMALY-683577, ERNAKULAM(DIST)

JUNE 2022

DECLARATION

I, **Angel Mary Joly** hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Applications is an authentic record of my original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date : **Signature:**

Place: Angamaly **Name:**

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY, ERNAKULAM-683577

DEPARTMENT OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project report titled **"Travel Assistant For Blinds : HELEN "** submitted by **Angel Mary Joly [Reg.No: FIT20MCA-2025]** towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by her during the year 2022.

Project Guide **Head of the Department**

ACKNOWLEDGEMENT

I am deeply grateful to **Dr.Manoj George** , Principal, FISAT, Angamaly for providing me with adequate facilities, way and means by which I was able to complete my mini project work and I express my sincere thanks to **Dr. C. Sheela** ,Vice Principal FISAT, Angamaly.

My sincere thanks to **Dr.Deepa Mary Mathwes**, Head of the Department of Computer Applications, FISAT, who had been a source of inspiration. I express heartiest thanks to **Dr.Shidha M V** ,my project guide for her encouragement and valuable suggestions . I express my heartiest gratitude to my scrum master **Dr.Deepa Mary Mathwes** and the faculty members in department for their constant encouragement and never ending support throughout the project.I would also like to express my sincere gratitude to the lab faculty members for their guidance

Finally I express my thanks to all my friends who gave me wealth of suggestion for successful completion of this project.

ABSTRACT

In today's advanced hi-tech world, the need of independent living is recognized in case of visually impaired people who are facing main problem of social restrictiveness. Visually impaired people face many problems in their day to day lives. Among them, travelling is one of the major concerns. In this application the blinds can travel through the bus independently without any assistance of people. The idea is implemented through an Android mobile app that focuses on voice assistant. There are two users in this application. The blind and the conductor. The blind requests help going to a certain destination. Server find out the condcutors going to that place and alerts conductor that a person at this location needs assistance. The conductor sets his/her destination at the beginning of trip. He/she can see the list of who all have requested help and their contact details. Also blind will get alert when he reach destination. This app also capable to assist using voice commands to find out the current location, to know the date and time, for making normal calls, Bluetooth connectivity, check network connection, to change the phone mode into vibrate/sound, voice chat bot.

Contents

Chapter 1

INTRODUCTION

VISUAL impairment is a major disability faced by the visually challenge people. A person who cannot see can never feel the emotion that a person feels who can see the world. This visibility problem is a black dot faced by billions of people around the entire world. Our aim is to remove this black dot with the help Artificial Intelligence and advanced technologies.

Statistics show that 87% of the visually challenged individuals live in developing countries, where the most viable mobility option are public bus transportation systems. An equipment called Bluetooth proximity sensor is currently used by developed countries to alert the visually impaired people of the arrival of buses. Public bus transport information system for visually impaired people suggests a more accurate and cost-effective idea to ensure safe journey of the visually challenged by utilizing the vast possibilities of smartphone applications.

1.1 Relevance Of The Topic

Visually impaired people face many problems in their day to day lives. Among them, travelling is one of the major concerns. In this application the blinds can travel through the bus independently without any assistance of people. Public bus transport information system for visually impaired people suggests a more accurate and cost-effective idea to ensure safe journey of the visually challenged by utilizing the vast possibilities of smartphone applications. The significance and purpose of this work is to provide a user-friendly and accurate travelling option for the visually challenged people within their budget. Compared to existing technologies, this project proposes a novel, user-friendly and cost-effective method for helping Visually challenged people.

1.2 Existing System

There are voice assistant for blinds to read text, identify and analyze images and detect objects. The existing system provides actual position and nearby places of the user with the help of GPS. It allow user to find nearby places from a specific category such as banks, health, shop, etc. None of them provide a chatbot to chat with or a voice assistant to interact with the application. It does not support speech-to-text function. There is only one type of user for this application. In existing systems there is always a human assistant is needed to handle the application. There are talk-back functionality but the user can't control the assistant by their own. Hence the system need to be updated to handle it easy by visually impaired people.

1.3 Proposed System

In the proposed system text-to-speech function is used. So the user can give any command as voice and the system will act according to the command. The main function of this application is take destination from the blind user as voice command and find out nearest conductor to the same destination as of the blind . There are two users for this application such as passenger and conductor. Both can register and login to the application for onetime. After that the app will run in background. The other functions in this application are finding current location, make calls, know the date and time, battery level information all these things can be done through the voice commands.

This system consists of three main parts - Passenger, Conductor and Server. Implementation of this idea would result in comfortable and safe journeys for the visually impaired without depending on co-passengers. It would also help in increasing the popularity of public bus transportation facilities which in turn helps in acquiring increased revenues.

Chapter 2

PROOF OF CONCEPT

This project is an Android application that is designed to help blinds to use the possibilities of smart phones to travel independently. In our world, all the people have the right to live independently but blinds want to seek help of others for doing their activities. To encourage them for coming out of their insecurity I am trying to boost their confident to travel independently. For a normal person it is easy to use a mobile, easy to travel alone. But for blinds or visually impaired people mobile phones, travelling etc are challenges.

There are some applications for blinds are available in Google play-store. They all aim to help blinds by providing object detection. Using the mobile camera object will be detected while walking. Another application is for recognizing images. None of them provide text-to-speech functionality. There is no voice assistant that help to travel along public bus transport.

The concept is to provide a voice assistant for blinds and visually impaired people. This can be implemented as a mobile application. The user can give destination through voice command and server will take the destination and connect with the nearest conductor. This voice assistant will also act as a guide to use mobile phones. Using the features of Android mobiles, blinds can access voice assistant by giving voice commands.

This is an innovative System for visually impaired people and acts as a voice assistant for them. This system is used to help the visually impaired to travel along bus transportation independently without the help of co-passengers and have access to the most important features of the phone enhancing the quality of the system making use of different custom layouts and using speech to text. The System has custom messaging feature with inbox and sent items, call dialer, notes and battery level checking and reminder. All actions performed by the user the system speaks out and helps the user to know his current position. The application mainly focus on the interaction between passenger and conductor for the safe and comfortable journey of blind passengers.

A person who has lost vision feels incomplete. A lot of challenges have to be faced when he/she has no vision. The system proposed handles the difficulties faced by the visually challenged people. Some of the problems faced by the visually challenged people are tried to be solved with the help of Artificial Intelligence and Machine learning. Natural and rich conversational experiences are building using the dialog-flow platform. A chat-bot is assisted, which helps the visually impaired to talk and get the desired response.

Visually-impaired people often feel handicapped and find it difficult to explore the world around them freely without any kind of help and assistance. This lack of independence restricts them from feeling confident about themselves. Furthermore, all virtual travel assistants currently existing in the market, are chatbots associated with textual interaction. While this may be useful more often than not, in certain situations, users may find textual interaction uncomfortable and prefer a conversational interface over text.

The major difference between a messenger chatbot and a voice bot is the way we interact with them. A textbased messenger chatbot exists on one or more messaging platforms, including both SMS and web-based messengers and users primarily interact through a screen via text or button presses. A user's interaction with a voice-enabled bot is different: they converse with such a bot via their voice in natural language. The voice bot then answers back using pre-recorded messages, text-tospeech responses or a mixture of both.

2.1 Objectives

- To help visually impaired people travel independently.
- To help visually impaired people capable of using mobile phone easily.

Chapter 3

IMPLEMENTATION

HELEN is an android application which act as a companion for visually impaired people. Here I am using Android as user friendly interface to make the entire front end. The project uses Android Studio to develop the android application.

3.1 Hardware Specification

The hard ware is the place where all the information and data are stored permanently. So hardware must be reliable and cost effective. The hardware must suit all the application development. It is fast enough to complete and do all the jobs and executions.
Processor: Intel Core i5 processor
Android device: ARMv7 Processor
Phone memory: 15MB
SD card: 500MB

3.2 Software Specification

The software specification means the operating system and all other application or tools used for the development of the proposed system. It includes the operating system, and the software which I am used.
Operating System: Windows 10
Front end: Android

3.3 Developing Tool

Android Studio is the official integrated (IDE) for Android platform development. It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0. Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, Mac OS X and Linux, and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

3.4 Developing Platform

WINDOWS 10

Windows 10 is a personal computer operating system developed and released by Microsoft as part of Windows NT family of operating systems. It was officially unveiled in September 2014 following a brief demo at Build 2014. The first version of the operating system entered a public beta testing process in October 2014, leading up to its consumer release on July 29,2015 Windows 10 in-

troduces what Microsoft described as “universal apps”; expanding on Metro-style apps, these apps can be designed to run across multiple Microsoft product families with nearly identical code including PCs, tablets, smartphones, embedded systems, Xbox ones, surface hub and Windows Holographic. The Windows user interface was revised to handle transitions between a mouse-oriented interface and a touch-screen optimized interface based on available input devices particularly on 2-in-1 PCs; both interfaces include an updated start menu which incorporates elements of Windows 7’s traditional start menu with the tiles of Windows 8 Microsoft described Windows 10 as an “Operating system as a service” that would receive or-going updates to its features are functionality, augmented with the ability for enterprise environments to receive non-critical updates, at a slower pace, or use long-term support milestones that will only receive critical updates, such as security patches, over their five-year lifespan of mainstream support Windows 10 received mostly positive reviews upon its original release in July 2015; critics praised Microsoft’s decision to downplay user-interface mechanics introduced by Windows 8 (including the full screen apps and start screen) in nor-touch environments to provide a desktop-oriented interface in line with previous version of Windows, although Windows 10’s touch-oriented user interface mode was panned for containing regressions upon the touch-oriented interface of Windows 8.

3.5 Front end Tool

ANDROID

Android is a mobile operating system that is based on a modified version of Linux. It was originally developed by a start-up of the same name, Android, Inc. In 2005, as a part of its strategy to enter the mobile space, Google purchased Android and took over its development work (as well as its development team). Google wanted Android to be open and free; hence, most of the Android code was released under the open source Apache License, which means that anyone who wants to use Android can do so by downloading the full Android source

code. This simple development model makes Android very attractive and has thus piqued the interest of many vendors. Tis has been especially true for companies affected by the phenomenon of Apple's iPhone, a hugely successful product that revolutionized the smartphone industry. Such companies include Motorola and Sony Ericsson, which for many years have been developing their own mobile operating systems. When the iPhone was launched, many of these manufactures hand to scramble to fired new ways of revitalizing their products. These manufactures see Android as a solution they will continue to design their own hardware and use Android as the operating system that powers it. The main advantage of adopting Android is that it offers a unified approach to application development. Developers need only develop for Android, and their applications should be able run on numerous different devices, as long as the devices are powered using Android. In the world of smartphones, applications are the most important part of the success chain.

3.6 Back End Tool

MongoDB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL) which is deemed non-free by several distributions. Fields in a MongoDB document can be indexed with primary and secondary indices or index. MongoDB provides high availability with replica sets. A replica set consists of two or more copies of the data. Each replica-set member may act in the role of primary or secondary replica at any time. All writes and reads are done on the primary replica by default. Secondary replicas maintain a copy of the data of the primary using built-in replication.

3.7 System Architecture

USE CASE DIAGRAM

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams.

It represents how an entity from the external environment can interact with a part of the system.It is essential to analyze the whole system before starting with drawing a use case diagram, and then the system's functionalities are found. And once every single functionality is identified, they are then transformed into the use cases to be used in the use case diagram.After that, we will enlist the actors that will interact with the system. The actors are the person or a thing that invokes the functionality of a system. It may be a system or a private entity, such that it requires an entity to be pertinent to the functionalities of the system to which it is going to interact. These are the purposes of a use case diagram :

- It gathers the system's needs.
- It depicts the external view of the system.
- It represents the interaction between the actors.

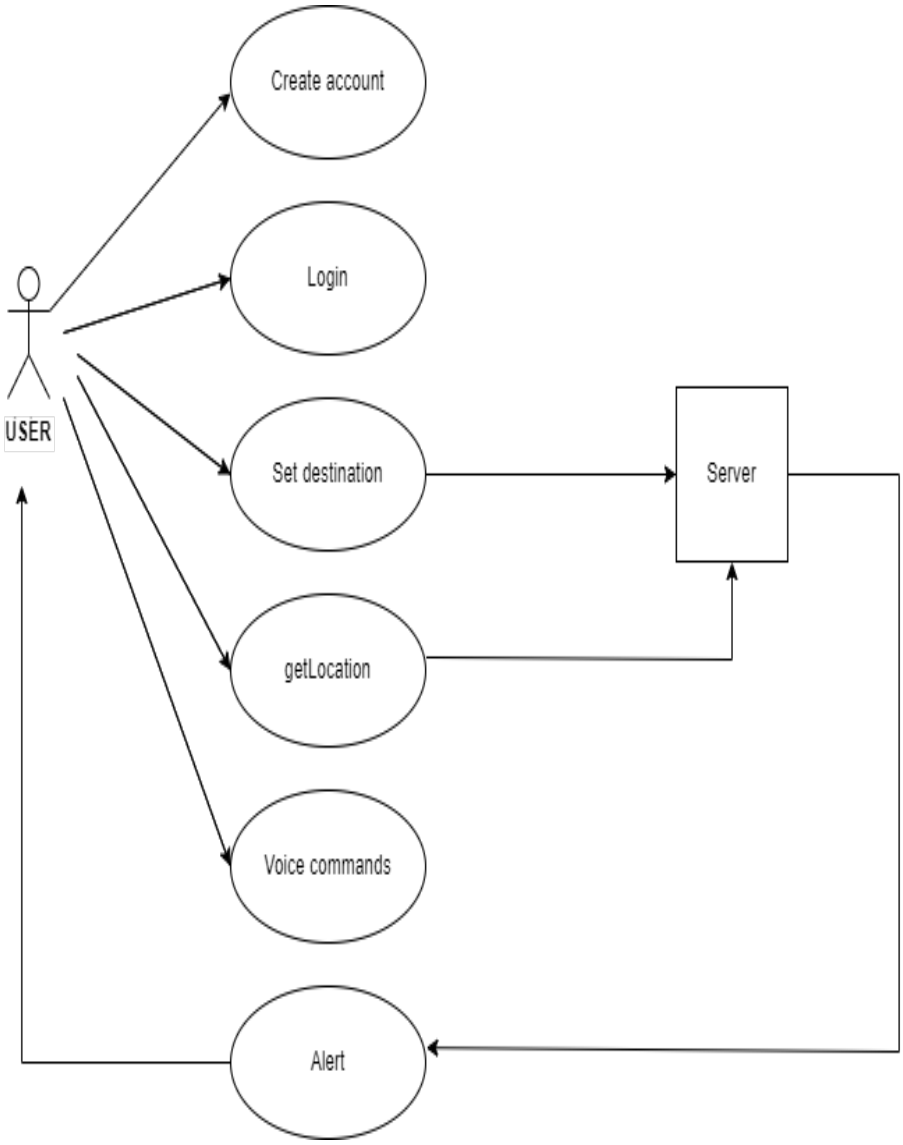

Figure 3.1: Use Case Diagram Of HELEN

3.8 Modules

3.8.1 GOOGLE SPEECH SERVICES

Using the function 'promptSpeechInput' imported the google speech services for converting the audio input to text format. Through this function voice commands are activated.

3.8.2 USER REGESTRATION

This module contains user login and registration. If it is a new user, then have to register into it. Registered users can use this application. Validations are there for user registration.

3.8.3 LOCATION IDENTIFICATION

The project takes location co-ordinates ie, the latitude and longitude, form the android module. This latitude and longitude values are converted to corresponding location.

3.8.4 SETTING DESTINATION

The destination of both users are set and send to the server. The destination and user details store in local storage.

3.8.5 CALL PERMISSION

To provide a call to any number a permission is needed by the default call application in the mobile. The permission can be granted through 'checkSelfPermission' function.

3.9 SYSTEM DESIGN

3.9.1 INPUT DESIGN

XML in Android

XML stands for Extensible Mark-up Language. XML is a markup language much like HTML used to describe data. XML tags are not predefined in XML. We must define our own Tags. Xml as itself is well readable both by human and machine.

In Android there are several xml files used for several different purposes:

1. Layout XML Files: Layout xml files are used to define the actual UI (User interface) of our application. It holds all the elements (views) or the tools that we want to use in our application. Like the TextView's, Button's and other UI elements. Created three layout XML files for splashActivity, LoginActivity and HomeActivity.

2. Manifest xml File (Mainfest.xml): This xml is used to define all the components of our application. It includes the names of our application packages, our Activities, receivers, services and the permissions that our application needs. For Example – Suppose we need to use the internet in our app then we need to define Internet permission in this file. In Manifest file given app icon, uses permission for bluetooth, call, location, network.

3. Strings xml File (strings.xml): This xml file is used to replace the Hard-coded strings with a single string. We define all the strings in this xml file and then access them in our app (Activity or in Layout XML files) from this file. This file enhances the reusability of the code.

4. Styles xml File (styles.xml): This xml is used to define different styles and looks for the UI (User Interface) of application. We define our custom themes and styles in this file.

5. Drawable xml Files: These are those xml files that are used to provide vari-

ous graphics to the elements or views of application. When we need to create a custom UI we use drawable xml files. Suppose if we need to define a gradient color in the background of Button or any custom shape for a view then we create a Drawable xml file and set it in the background of View. Here I have inserted the logo, wallpapers, the images used in this application.
6. Color xml File (colors.xml): This file is used to define the color codes that we used in our app. We simply define the colors in this file and used them in our app from this file. The color name teal_200 given to the support action bar.

3.9.2 OUTPUT DESIGN

User Interface Layout

The user interface for each component of your app is defined using a hierarchy of View and View Group objects. Each view group is an invisible container that organizes child views, while the child views may be input controls or other widgets that draw some part of the UI. This hierarchy tree can be as simple or complex as you need it to be (but simplicity is best for performance). To declare your layout, you can instantiate View objects in code and start building a tree, but the easiest and most effective way to define your layout is with an XML file. XML offers a human-readable structure for the layout, similar to HTML.

User Interface Components

You don't have to build all of your UI using View and View Group objects. Android provides several app components that offer a standard UI layout for which you simply need to define the content. These UI components each have a unique set of APIs that are described in their respective documents, such as Adding the App Bar, Dialogs, and Status Notifications.

Chapter 4

SYSTEM TESTING

In a software development project, errors can be injected at any stage during the development phase. For each phase I have discussed, there are different methods and techniques that are available for eliminating errors. However, no technique is perfect, and it is expected that some of the errors of the earlier phase will manifest themselves in the code. Hence, the code developed during the coding activities is likely to have some requirements errors and design errors, in addition to introduce during the coding activity.

4.1 White Box Testing

White box is a testing technique that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

4.2 Black Box Testing

Black box testing, also known as Behavioural testing, is a software testing method in which the internal structure design/ implementation if the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see.

4.3 Unit Testing

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyse and fix the defects.

4.4 Integration Testing

Integration testing, also known as integration and testing (IT), is a software development process which program units are combined and tested as groups in multiple ways. In this context, a unit is defined as the smallest testable part of an application. Integration testing can expose problems with the interfaces among program components before trouble occurs in real-world program execution. Integration testing is a component of Extreme Programing (XP), a pragmatic method of software development that takes meticulous approach to building a product by means of continual testing and revision.

4.5 Acceptance Testing

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it has met the required criteria for delivery to end users.

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

Artificial Intelligence is one of the most growing technology. These technologies are the playing vital role in the development of the IT sector. Here I have tried to use some features of Artificial Intelligence for the visually challenged people so that they can travel independently and also use mobile phones as normal people. Blind user can give destination on voice command and send it to the server. The developed application can track the location of blind passenger and the conductor and give alert to both of them when they are near. The location is updated in certain intervals. Voice button can be activated by clicking on the volume button.

The other additional features of this application are finding out current location of the user. The user can make calls. The features like date, time, battery level can be retrieved. All these features will be available by providing voice commands. Thus HELEN will really helpful to those blinds and visually impaired to travel independently in public bus transport and make use of the mobile features.

5.2 Future Scope

HELEN provides more and more updates, improvements and better user interaction with this application. Mobile phones are inevitable in our day to day life, so this application will be really helpful to the visually impaired people to move on with the technology.

- In future this proposed system will be able to use for travelling in all transport system.
- The calling facility can be enabled for contacting blind users.
- In addition to this, voice assistant will be able to detect objects in surrounding.

.

Chapter 6

APPENDIX

6.1 CODING

```
    package com.example.blindtravel;

import static android.util.Config.DEBUG;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;


import com.example.blindtravel.models.User;
import com.google.android.gms.location.LocationCallback;
```

```
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;
import java.text.DateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import java.util.Locale;
import java.util.Objects;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class home extends AppCompatActivity {

    private String BASE_URL = "http://10.100.16.76:5000";

    private SwipeRefreshLayout swipe_layout;
```

```
private RecyclerView clientList;
private EditText destination;
private Button profile;

private static final
int REQUEST_CODE_SPEECH_INPUT = 1;
private static final
int REQUEST_CODE_SPEECH_INPUT_VIVA = 4;
private static final
int REQUEST_CODE_LOCATION_SERVICE = 2;
private static final
int MY_CAMERA_PERMISSION_CODE = 100;
private static final
int MY_FINE_PERMISSION_CODE = 102;
private static final
int MY_COARSE_PERMISSION_CODE = 104;
int PERMISSION_ID = 44;

String id = null;
String role = "Client";
String email_glob = "";
Call<String> apiInstance;
ApiInterface apiInterface;
JSONArray fetched = null;
private int mBindFlag;

TextToSpeech prompt;
private TextToSpeech myTTS;

private View popupView;
```

```
int width;
int height;
private boolean focusable;
PopupWindow popupWindow;
private TextView chat_user;
private TextView chat_system;
private AudioManager audioManager;
FusedLocationProviderClient mFusedLocationClient;

@Override
protected void onCreate(Bundle savedInstanceState)  {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_home);
    getSupportActionBar().setTitle("Home Page");

    requestPermission();

    if(ContextCompat.checkSelfPermission
    (getApplicationContext(),
    Manifest.permission.ACCESS_FINE_LOCATION)
    != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(
                home.this,
                new String[]
         {Manifest.permission.ACCESS_FINE_LOCATION},
                2
        );
    }
    else {
        startLocationService();
```

```
}

Intent intent = getIntent();
id = intent.getStringExtra("id");
role = intent.getStringExtra("role");
email_glob = intent.getStringExtra("email");
Log.d("Home Page Intent Vals",id+" :
"+role+" : "+email_glob);
if(id==null) {
    Log.d("Test Shared Prefs Home",id+" :
    "+role+" : "+email_glob);
}

Intent service = new Intent(home.this,
VoiceCommandService.class);
service.putExtra("id",id);
service.putExtra("role",role);
service.putExtra("email",email_glob);
getApplicationContext().startService(service);
mBindFlag = Build.VERSION.SDK_INT <

Build.VERSION_CODES.ICE_CREAM_SANDWICH ?
0 : Context.BIND_ABOVE_CLIENT;



init();
onClickFunctions();

mFusedLocationClient = LocationServices.
```

```
        getFusedLocationProviderClient(this);
    }

    @Override
    public void onRequestPermissionsResult
    (int requestCode, @NonNull String[] permissions,
    @NonNull int[] grantResults) {
        super.onRequestPermissionsResult
        (requestCode, permissions, grantResults);

        if(requestCode==REQUEST_CODE_LOCATION_SERVICE
        && grantResults.length>0) {
            if(grantResults[0]==
            PackageManager.PERMISSION_GRANTED) {
                startLocationService();
            }
            else {
                Toast.makeText(home.this,
                "Permission Denied....",
                Toast.LENGTH_SHORT).show();
            }
        }

    @Override

        if(checkSelfPermission(Manifest.permission.
        ACCESS_COARSE_LOCATION) != PackageManager.
        PERMISSION_GRANTED){
            requestPermissions(new String[]{Manifest.permission.
            ACCESS_COARSE_LOCATION},
```

```
            MY_COARSE_PERMISSION_CODE);
        }
        if(checkSelfPermission(Manifest.permission.CAMERA)
        != PackageManager.PERMISSION_GRANTED){
            requestPermissions(new String[]
            {Manifest.permission.CAMERA},
            MY_CAMERA_PERMISSION_CODE);
        }
        //ActivityCompat.requestPermissions(this,
        new String[]{Manifest.permission.CAMERA},8);
       // ActivityCompat.requestPermissions(this,
       new String[]{Manifest.permission.BLUETOOTH},9);
    }

    public void initializeTextToSpeech() {
        myTTS = new TextToSpeech(this, i -> {
            if(myTTS.getEngines().size()==0){
                Toast.makeText(home.this,
                "Helen engine not on this device",
                Toast.LENGTH_LONG).show();
                finish();
            } else {
                myTTS.setLanguage(Locale.UK);

                Calendar c = Calendar.getInstance();
                int timeOfDay = c.get(Calendar.HOUR_OF_DAY);

                if(timeOfDay >= 0 && timeOfDay < 12){
                    speak("Good Morning.     ");
                    chat_system.setText("Good Morning");
```

```
                }else if(timeOfDay >= 12 && timeOfDay < 16){
                    speak("Good Afternoon.      ");
                    chat_system.setText("Good Afternoon");
                }else if(timeOfDay >= 16 && timeOfDay < 24){
                    speak("Good Evening.      ");
                    chat_system.setText("Good Evening");
                }
            }
        });
    }

    private boolean isLocationServiceRunning() {
        ActivityManager activityManager =
        (ActivityManager) getSystemService
        (Context.ACTIVITY_SERVICE);
        if(activityManager!=null) {
            for(ActivityManager.RunningServiceInfo
            service:activityManager

            .getRunningServices(Integer.MAX_VALUE)){
                if(LocationService.class.getName().
                equals(service.service.getClassName())) {
                    if(service.foreground) {
                        return true;
                    }
                }
            }
            return false;
        }
        return false;
```

```
    }

    private void startLocationService() {
        if(!isLocationServiceRunning()) {
            Intent intent = new Intent(getApplicationContext(),
            LocationService.class);
            intent.putExtra("role",role);
            intent.putExtra("email",email_glob);
            intent.putExtra("id",id);

            intent.setAction("startLocationService");
            startService(intent);

            Toast.makeText(this,"Location Service
            Started....",Toast.LENGTH_SHORT).show();
        }
    }

    private void stopLocationService() {
        if(isLocationServiceRunning()) {
            Intent intent = new Intent(getApplicationContext(),
            LocationService.class);
            intent.setAction("stopLocationService");
            intent.putExtra("id",id);
            intent.putExtra("role",role);
            startService(intent);
            Toast.makeText(this,"Location Service
            Stopped....",Toast.LENGTH_SHORT).show();
        }
    }
```

```
private void init() {
    clientList = findViewById(R.id.user_list_home);
    destination = findViewById(R.id.destination_entry_home);
    profile = findViewById(R.id.profile_home);
    swipe_layout = findViewById(R.id.home_layout);
}

private void initPopupWindow() {
    LayoutInflater inflater = (LayoutInflater)
    getSystemService(LAYOUT_INFLATER_SERVICE);
    popupView = inflater.inflate
    (R.layout.popup_window, null);
    width = LinearLayout.LayoutParams.MATCH_PARENT;
    height = LinearLayout.LayoutParams.WRAP_CONTENT;
    focusable = true;
    popupWindow = new PopupWindow(popupView,
    width, height, focusable);

    popupWindow.setFocusable(focusable);

    chat_system = popupView.findViewById(R.id.chat_system);
    chat_user = popupView.findViewById(R.id.chat_user);

    chat_system.setVisibility(View.VISIBLE);
    chat_user.setVisibility(View.VISIBLE);
    Toast.makeText(home.this,"Test toast",
    Toast.LENGTH_SHORT).show();
    //findViewById(R.id.imageView3).setVisibility(View.VISIBLE);
    //findViewById(R.id.imageView4).setVisibility(View.VISIBLE);
```

```
        Toast.makeText(home.this,"Test toast",
        Toast.LENGTH_SHORT).show();
    }

    private void show_popup(View view) {
        popupWindow.showAtLocation(view, Gravity.CENTER, 0, 0);
        popupView.setOnTouchListener(new View.OnTouchListener() {
            @Override
            public boolean onTouch(View view,

    MotionEvent motionEvent) {
                chat_system.setVisibility(View.GONE);
                chat_user.setVisibility(View.GONE);
                findViewById(R.id.imageView3).
                setVisibility(View.GONE);
                findViewById(R.id.imageView4).
                setVisibility(View.GONE);
                popupWindow.dismiss();
                return true;
            }
        });
    }

    private void setUserList() {
        UserListAdapter adaptor = new UserListAdapter
        (fetched,getApplicationContext(),role);
        RecyclerView.LayoutManager layoutManager =
        new LinearLayoutManager(home.this);
        clientList.setLayoutManager(layoutManager);
        clientList.setAdapter(adaptor);
```

```
    }

    private void speechToText() {

        Intent intent = new Intent(RecognizerIntent.
        ACTION_RECOGNIZE_SPEECH);

        prompt = new TextToSpeech(this, new
        TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int i) {
                if(i==TextToSpeech.SUCCESS) {
                    Toast.makeText(home.this,
                    "Speak your Destination ",
                    Toast.LENGTH_SHORT).show();
                    prompt.setLanguage(Locale.UK);
                    prompt.speak("Speak your destination :

                    ",TextToSpeech.QUEUE_ADD,null);
                }
                intent.putExtra(RecognizerIntent.
                EXTRA_LANGUAGE_MODEL, RecognizerIntent.
                LANGUAGE_MODEL_FREE_FORM);
                intent.putExtra(RecognizerIntent.
                EXTRA_LANGUAGE, Locale.getDefault());
                intent.putExtra(RecognizerIntent.
                EXTRA_PROMPT,
                "Speak your destination");
            }
        });
```

```
        try {
            startActivityForResult(intent,
            REQUEST_CODE_SPEECH_INPUT);
        }
        catch (Exception e) {
            Toast.makeText(home.this,e.getMessage(),
            Toast.LENGTH_SHORT).show();
        }
    }




            @Override
            public void onFailure(Call<String> call, Throwable t)
            {
                Toast.makeText(home.this,t.getMessage(),
                Toast.LENGTH_SHORT).show();
                Log.e("Api Error Test",t.getMessage());
            }
        });
    }
```

6.2 SCREEN SHOTS

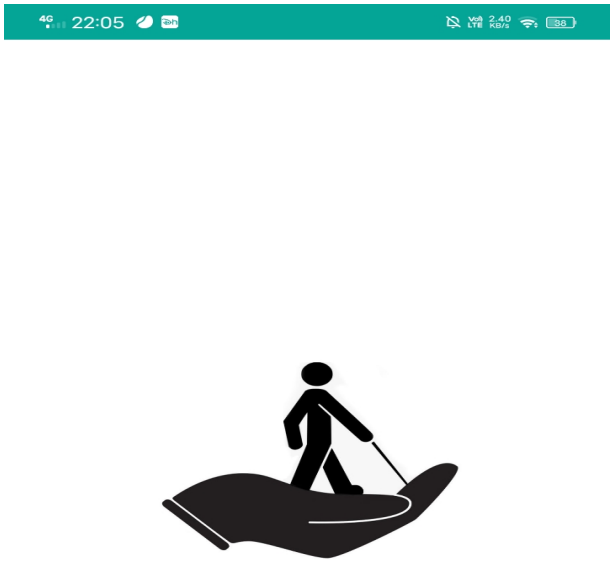

Figure 6.1: Splash Activity

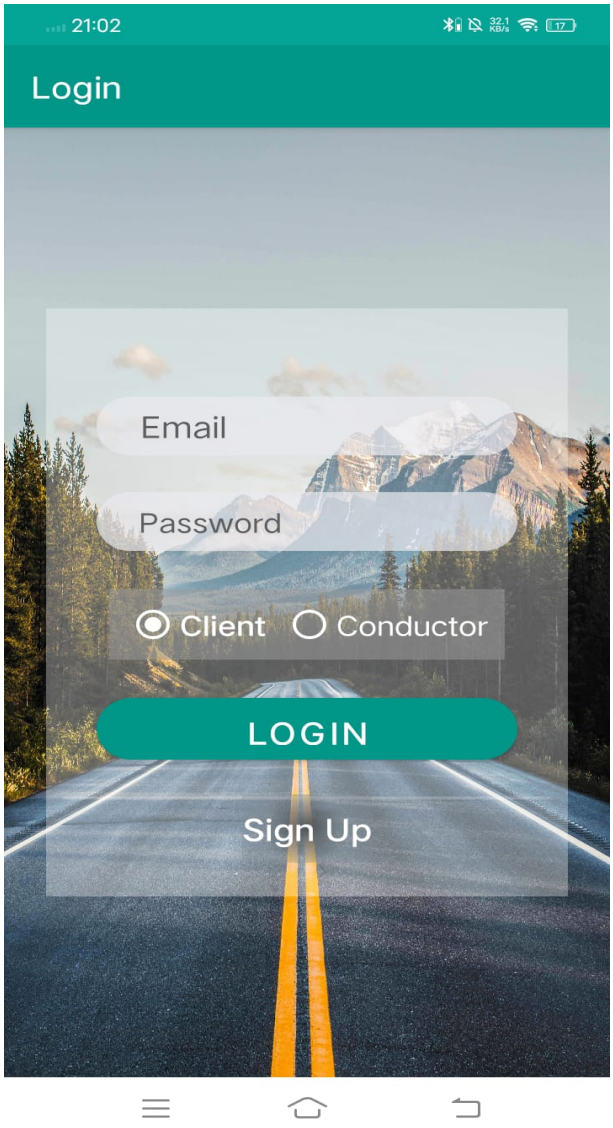

Figure 6.2: Login

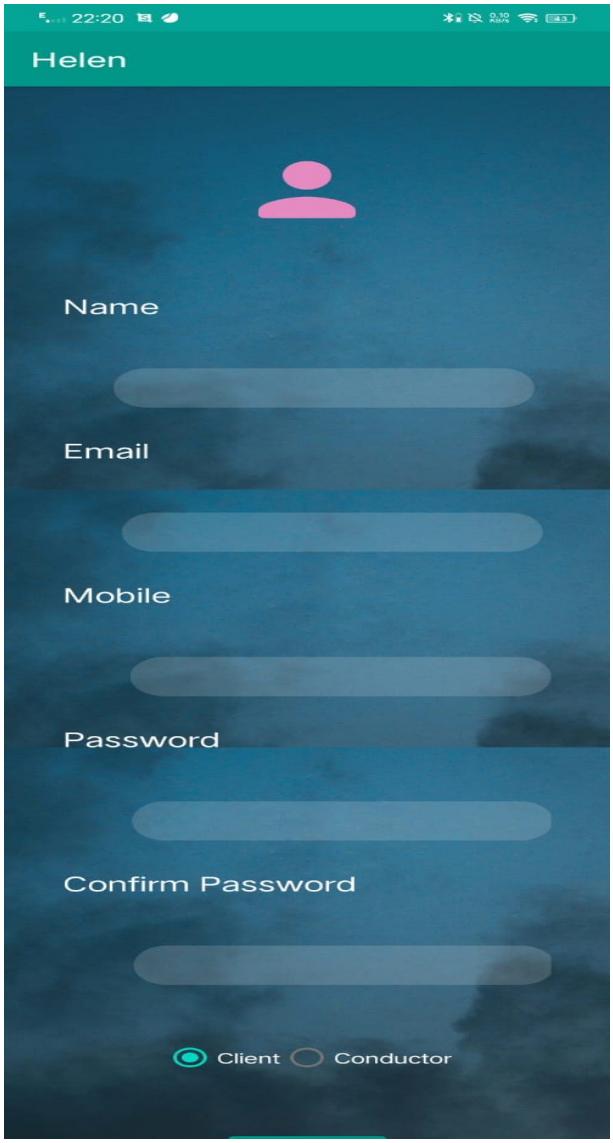

Figure 6.3: Sign Up

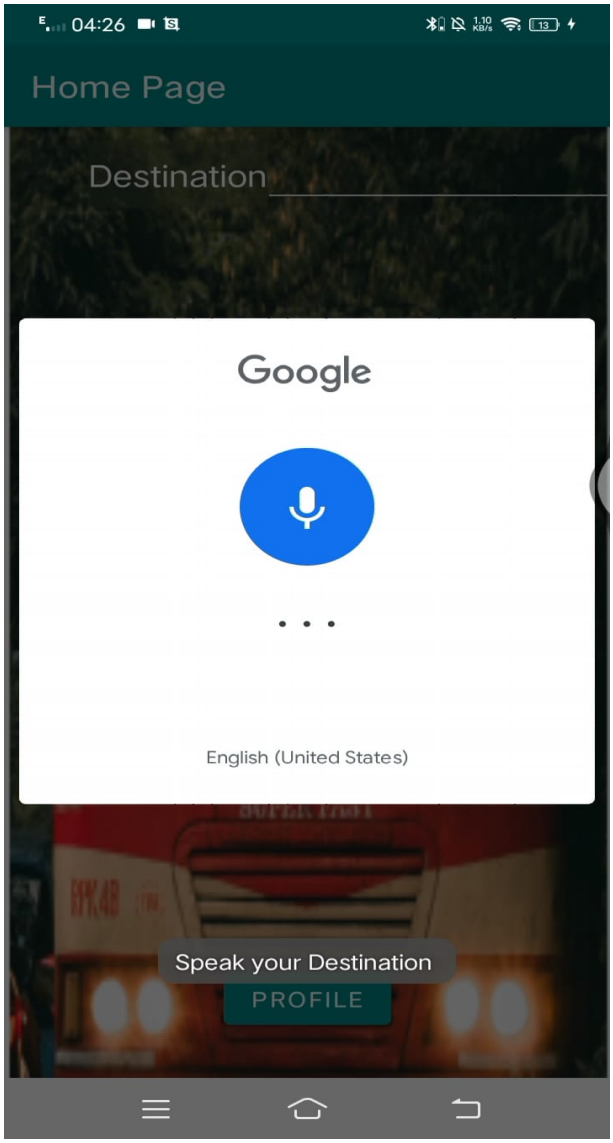

Figure 6.4: Client Home

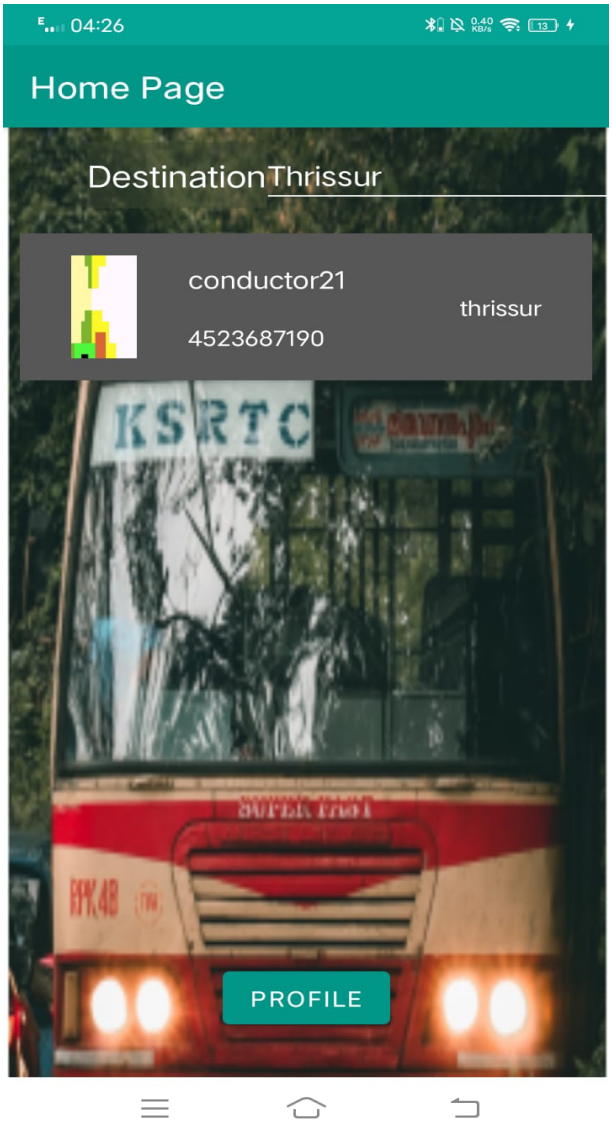

Figure 6.5: Conductor List

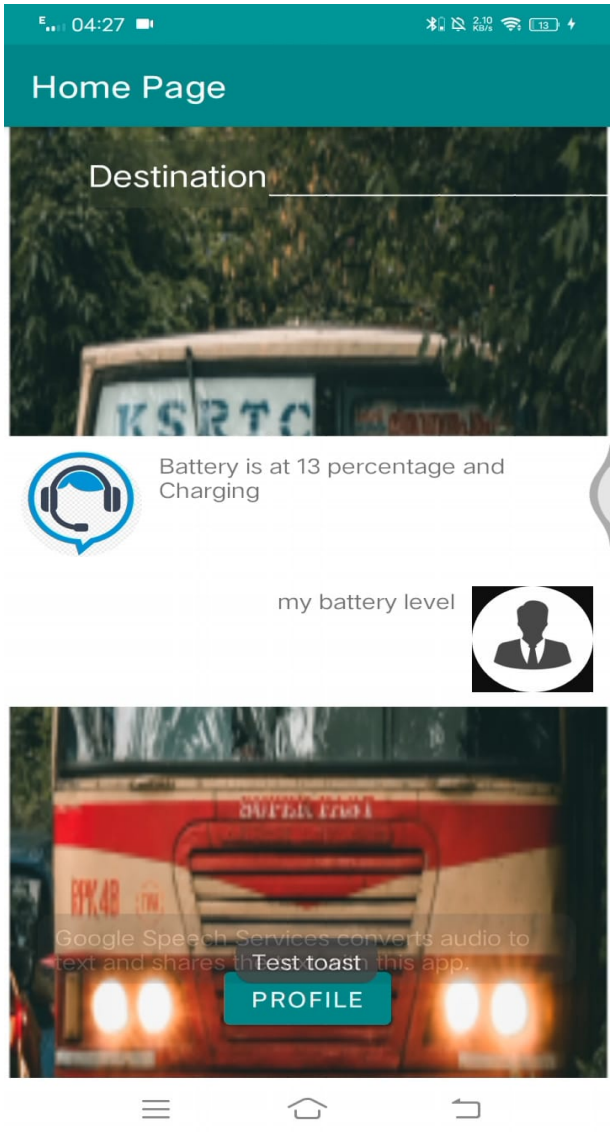

Figure 6.6: Battery

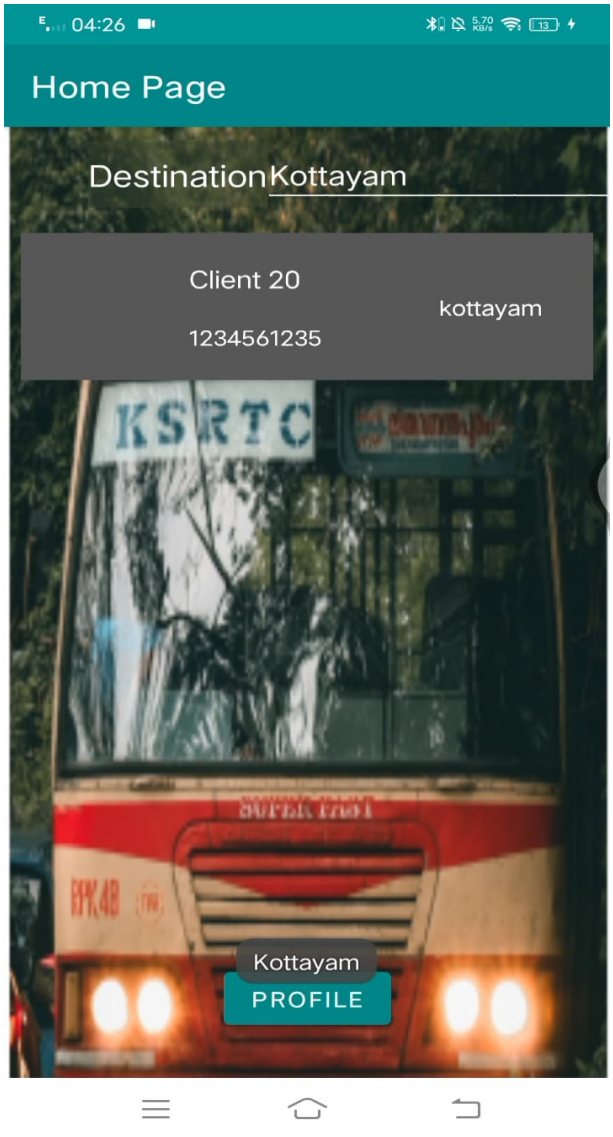

Figure 6.7: Conductor home

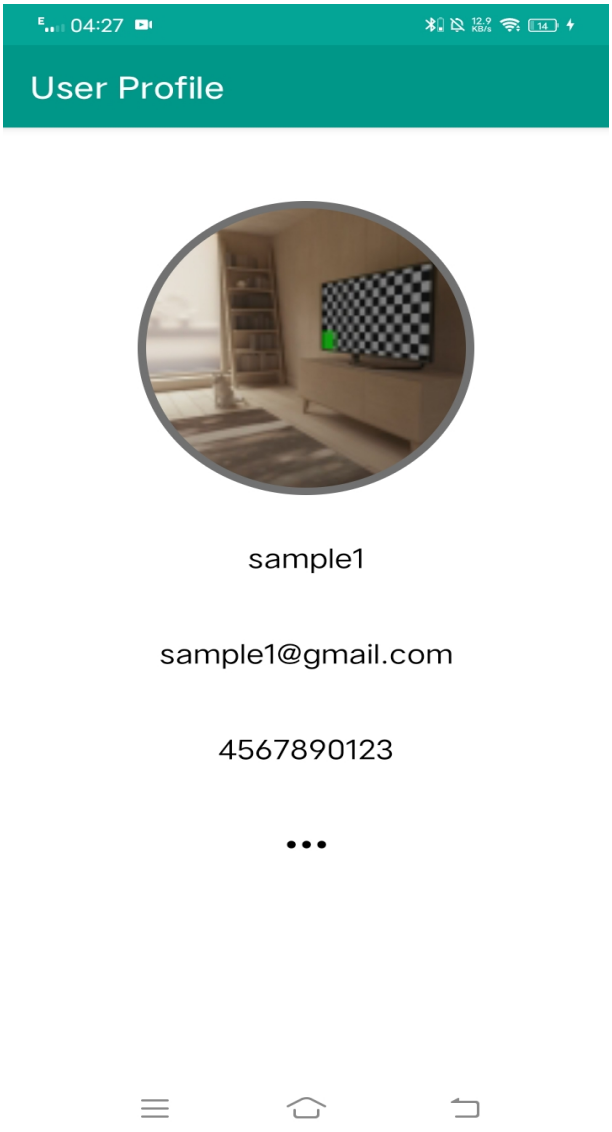

Figure 6.8: Profile

Chapter 7

REFERENCES

1. https://www.inettutor.com/source-code/voice-assistant-application-for-people-with-vision-impairment/

2. https://www.elprocus.com/gps-based-voice-alert-system-for-blind-people/

3. Wei-Meng Lee, Beginning Android 4 Application Development, Wiley publications, 2012.

4. N. Sripriya, S. Poornima, S. Mohanavalli, R. P. Bhaiya and V. Nikita, ”Speech-Based Virtual Travel Assistant For Visually Impaired,” 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP), 2020, pp. 1-7, doi: 10.1109/ICCCSP49186.2020.9315217.

5. S. P, S. N, P. D and U. M. R. N, ”BLIND ASSIST : A One Stop Mobile Application for the Visually Impaired,” 2021 IEEE Pune Section International Conference (PuneCon), 2021, pp. 1-4, doi: 10.1109/PuneCon52575.2021.9686476.