FEED THE NEED

A CAPSTONE PROJECT REPORT

Submitted in partial fulfillment of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

by

Nithin Sai Maguluri (19BCD7099)
Saranya Koganti (19BCN7187)
Bolagani Leela Naga Manumitha (19BCE7555)

Under the Guidance of

DR. Venkata Rami Reddy Chirra

SCHOOL OF COMPUTER SCIENCE ENGINEERING
VIT-AP UNIVERSITY
AMARAVATI- 522237

DECEMBER 2023

CERTIFICATE

This is to certify that the Capstone Project work titled "**FEED THE NEED**" that is being submitted by **Nithin Sai Maguluri (19BCD7099), Saranya Koganti (19BCN787), Bolagani Leela Naga Manumitha (19BCE7555)** is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Dr. Venkata Rami Reddy Chirra
Guide

The thesis is satisfactory / unsatisfactory

Internal Examiner **External Examiner**

Approved by

PROGRAM CHAIR **DEAN**
B. Tech. CSE School Of Computer Science Engineering

ACKNOWLEDGEMENTS

We would especially want to extend our gratitude to Dr. Venkata Ramireddy Chirra, our project guide, for his time and efforts during the project's completion. Your useful advice and suggestions were really helpful to us during the project's completion. We thank you for solving all our queries and being understanding throughout the whole time and finally we feel grateful to have you as our project guide.

Nithin Sai Maguluri
Saranya Koganti
Bolagani Leela Naga Manumitha

Computer Science and Engineering, VIT-AP

ABSTRACT

Every year, almost 1.3 billion tons or about one-thirdof the food produced for human consumption worldwide is lost or squandered. According to the United Nations, 25000 people every day die from hunger-related illnesses, including more than 10000 children. Not only is there wastage in this situation, but there is also poor waste management. In the realm of smartphones, the Feed the Need app is incredibly helpful for addressing these issues. it serves as a conduit between those who want to donate food and those NGOs that want to receive it.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Every year, almost 1.3 billion tons or about one-thirdof the food produced for human consumption worldwide is lost or squandered. According to the United Nations, 25000 people every day die from hunger-related illnesses, including more than 10000 children. Not only is there wastage in this situation, but there is also poor waste management. In the realm of smartphones, the Feed the Need app is incredibly helpful for addressing these issues. Along with another special feature where any regular person can volunteer to fight world hunger by delivering the food requested by the NGO, it serves as a conduit between those who want to donate food and those NGOs that want to receive it.

1.1 Objectives

The following are the objectives of this project:

- reducing food waste and satisfying hunger.
- To develop a food donation application that will serve as a conduit for users looking for a way to donate extra food to NGOs without wasting it.
- To provide a sustainable atmosphere where leftover food from events or restaurants can be fed to the hungry and undernourished.

1.2 Background and Literature Survey

- Up to 40% of the food produced in India each day is wasted, claims a study. There is enough evidence in our landfills, streets, and trash cans.
- Indians squander as much food as the entire country of the United Kingdom does, according to a statistic that may not actually reflect our love of excess as much as it does the size of our population.
- In India, over 21 million tonnes of wheat are lost to trash, and 50% of all grain produced worldwide suffers from the same fate and is never distributed to the poor.

- In fact, the agricultural ministry estimates that the nation wastes food worth INR 50,000 crores each year.

1.3 Organization of the Report

The remaining chapters of the project report are described as follows:

- Chapter 2 contains the proposed system, methodology, hardware and software details.
- Chapter 3 gives the cost involved in the implementation of the project.
- Chapter 4 discusses the results obtained after the project was implemented.
- Chapter 5 concludes the report.
- Chapter 6 consists of codes.
- Chapter 7 gives references.

CHAPTER 2

FEED THE NEED

This Chapter describes the proposed system, working methodology, software and hardware details.

2.1 Proposed System

- The two types of end users in this system are donors and NGOs
- If there is any leftover food, the restaurants or events (donors) will submit a request through the Feed the Need application.
- The food will be visible to any nearby NGOs, who can then use the app to request it.

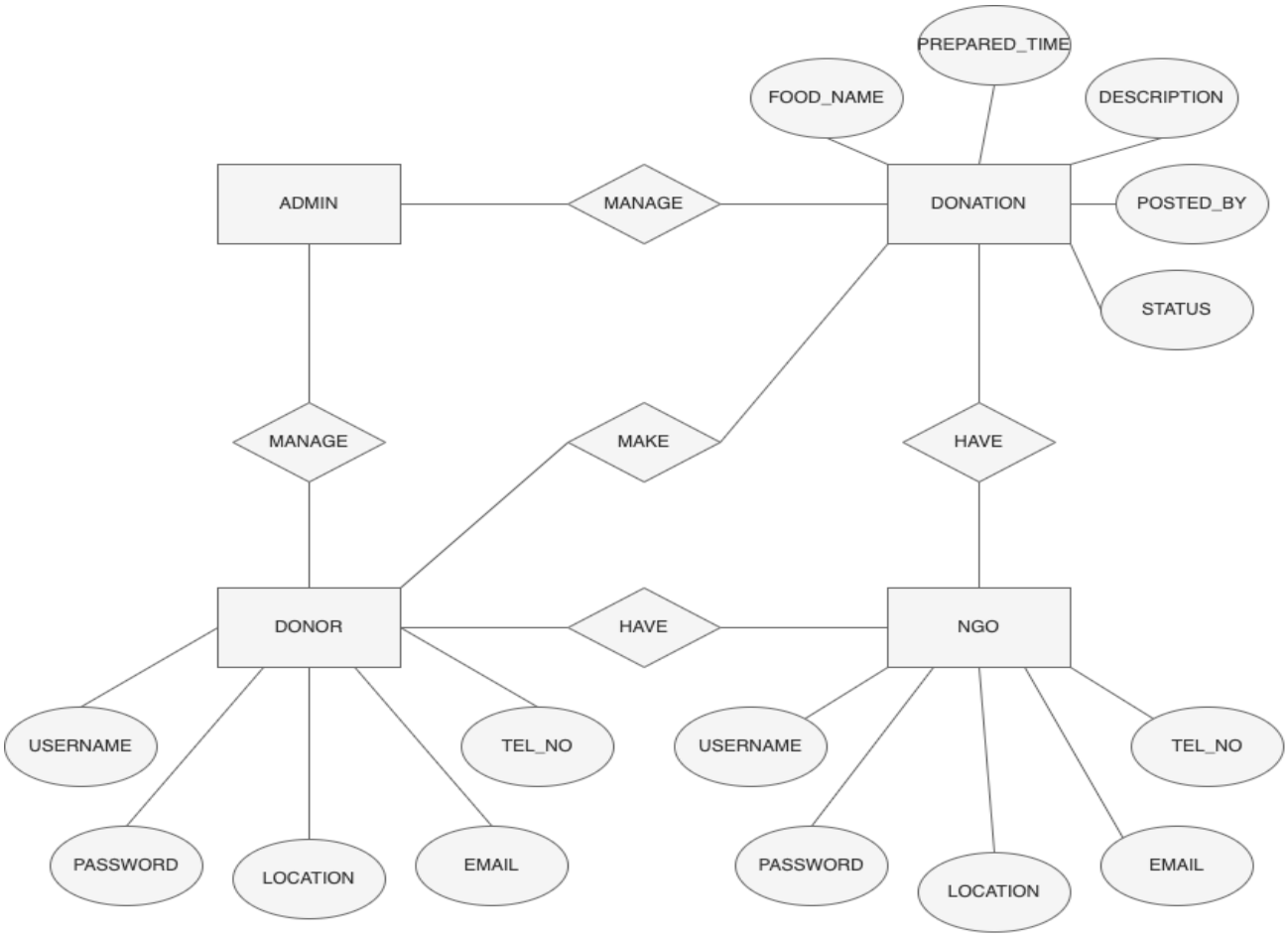
2.2 Working Methodology

- The user is presented with two options as soon as he launches the Feed The Need app.
- The register button can be used to enter information for a new NGO or DONOR, and the login button allows those users to log in with the email address and password they provided during registration.

- After logging in, the donor can add food details, upload a photo of the food he wants to contribute, and the app will take his location automatically.
- The NGO logs in to view these donor -uploaded details and asks the food using the request option; the app also automatically takes note of the NGO's location.

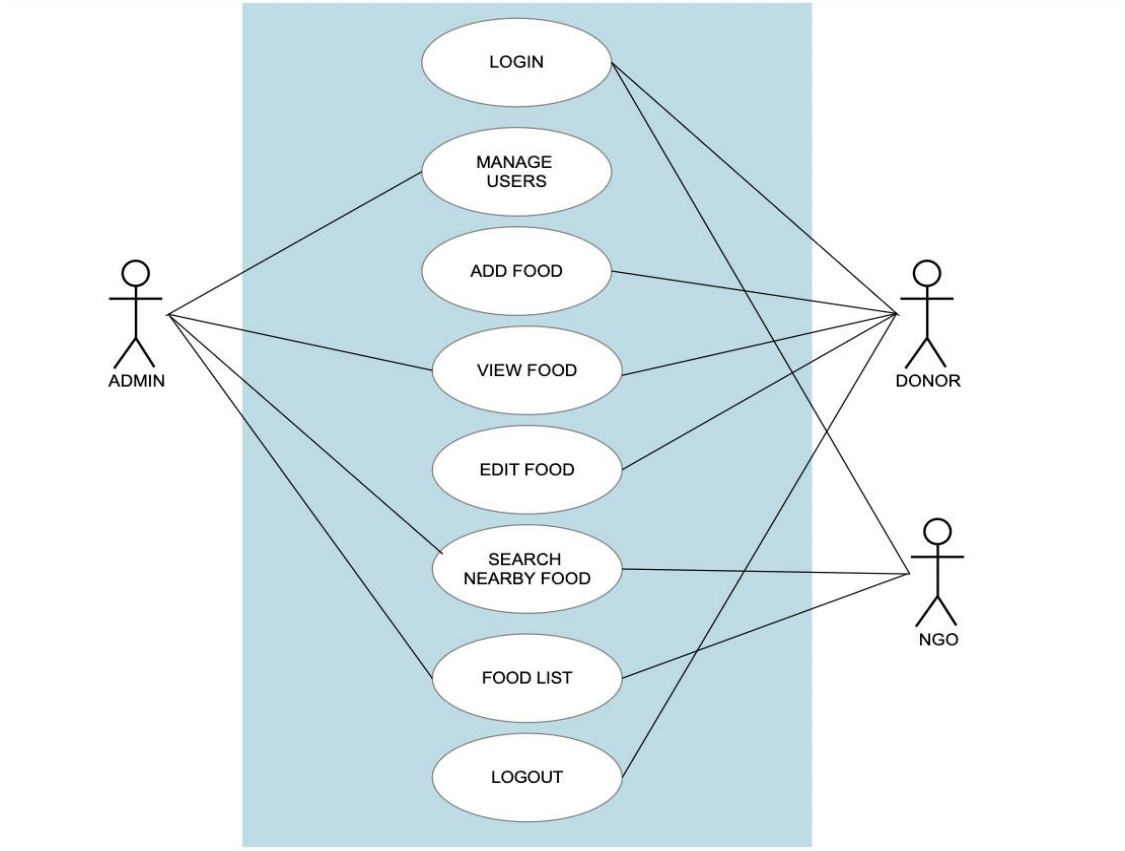
2.3 Design

2.3.1 Entity relationship diagram

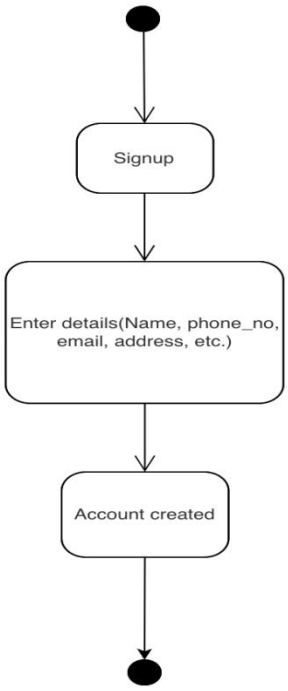


2.3.2 UML Diagrams

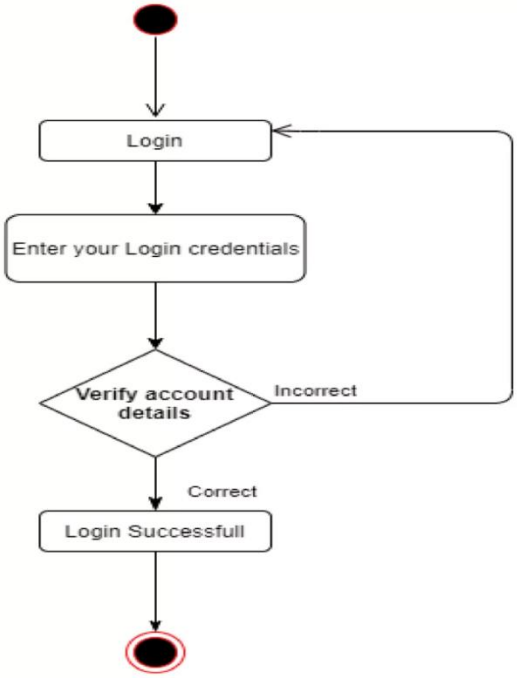
Use case diagram



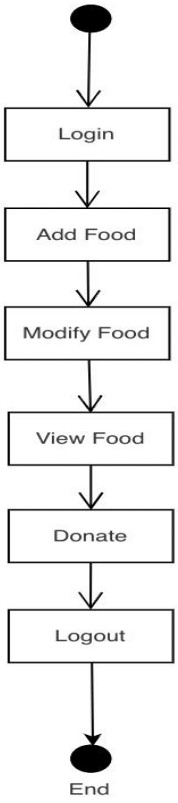
Activity diagram for registration



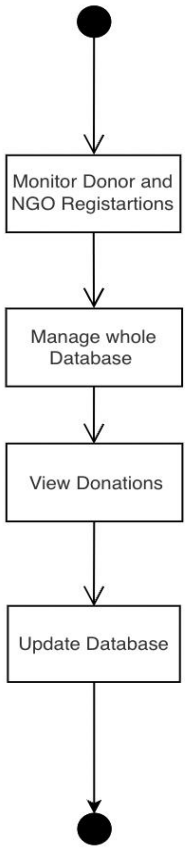
Activity diagram for login



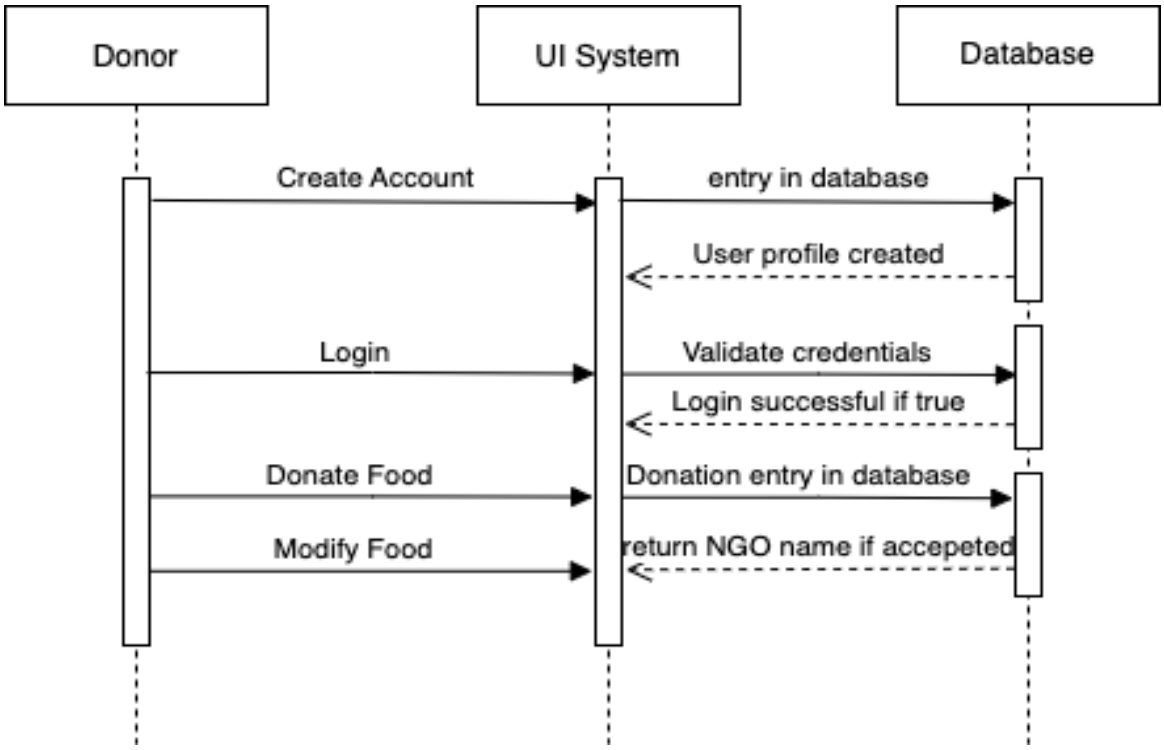
Activity diagram for donor



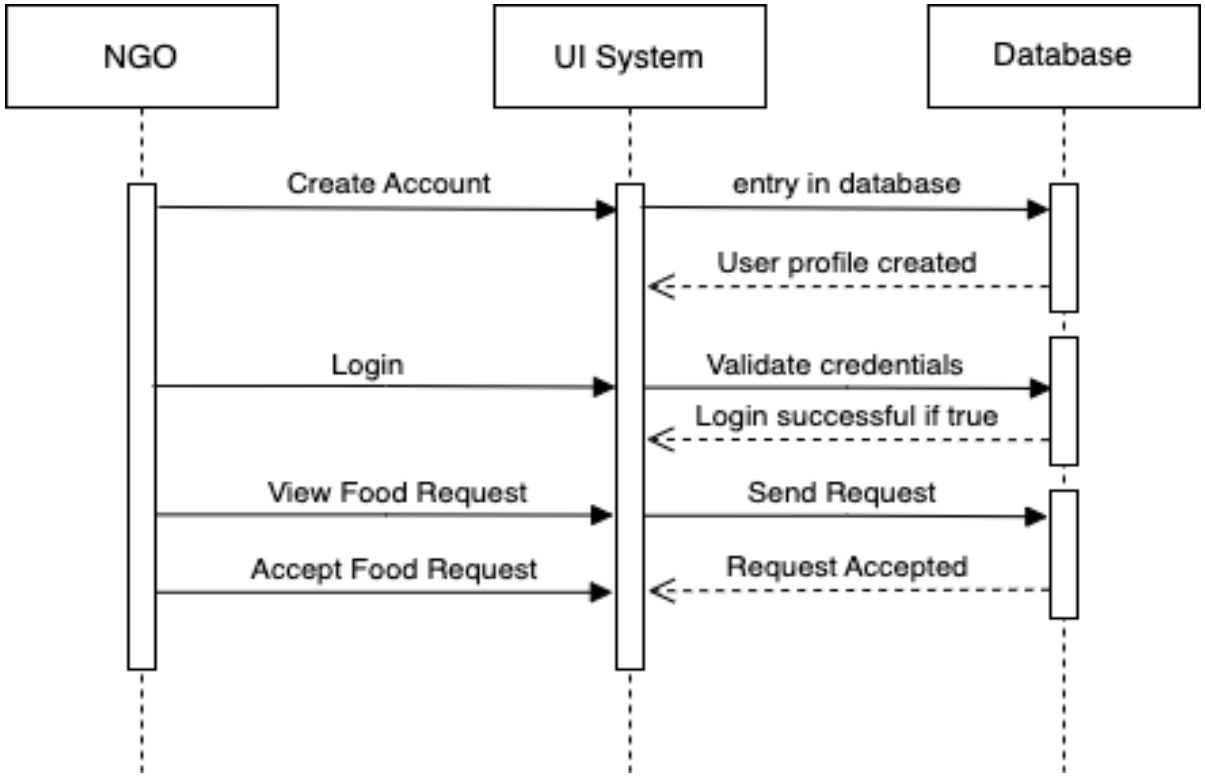
Activity diagram for NGO



Sequence diagram for donor



Sequence diagram for NGO



2.4 System Details

This section describes the software and hardware details of the system:

2.4.1 Software Details

This project is implemented in android studio with the programming language being java and the backend as firebase.

2.4.2 Hardware Details

There are no hardware components needed in this project, only laptop is sufficient for code execution and result.

CHAPTER 3

COST ANALYSIS

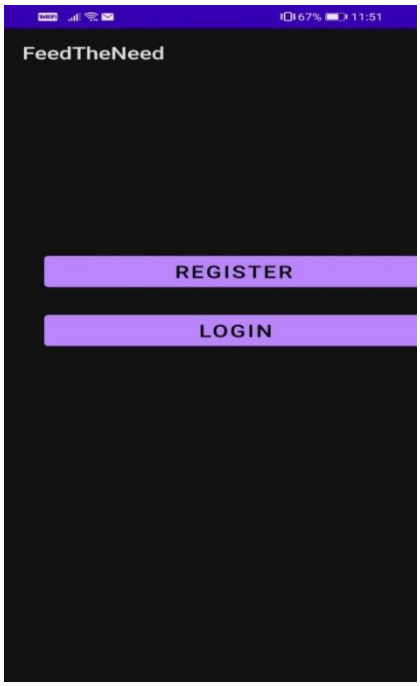
3.1 List of components and their cost

There are no hardware components needed for this project. everything is done in Android Studio software present in the laptop.

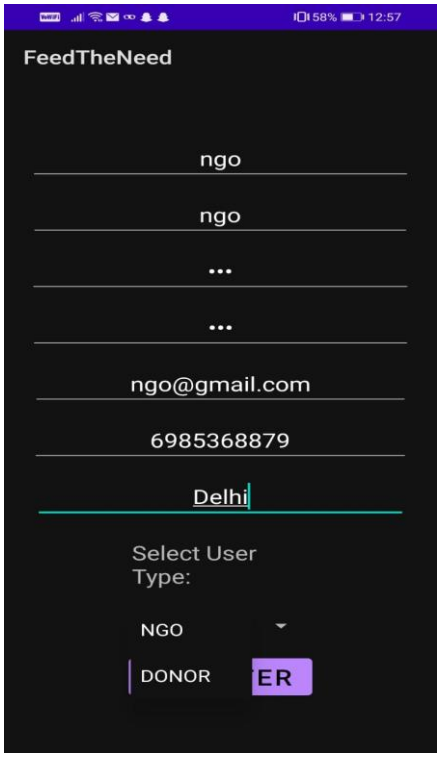
CHAPTER 4

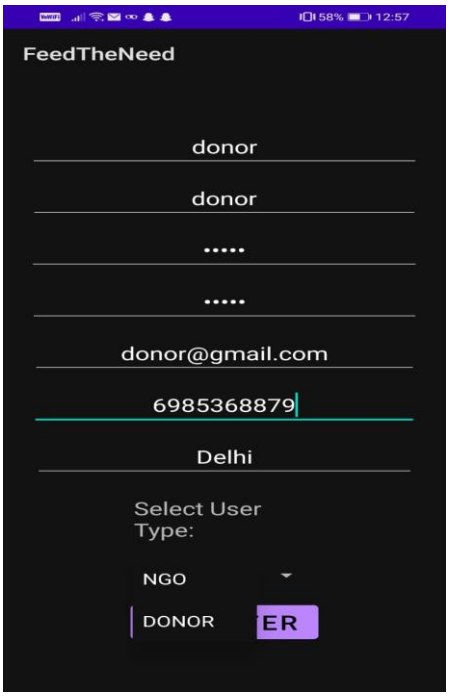
RESULTS AND DISCUSSIONS

Home Page of Feed The Need:

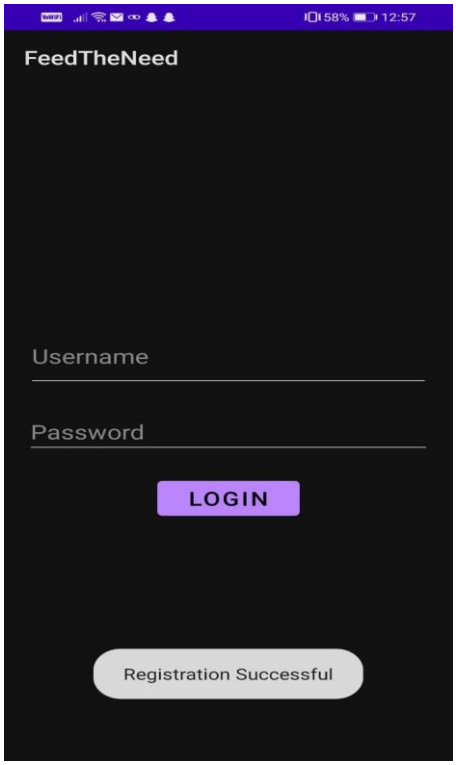


Registration Page for NGO and Donor:



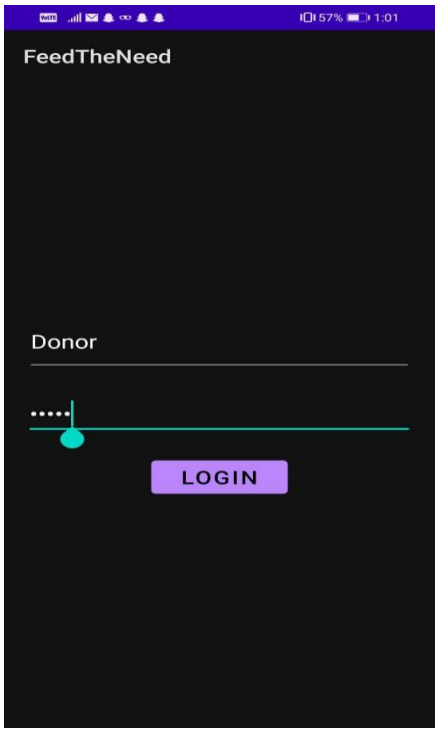


NGO and Donor Login Page:



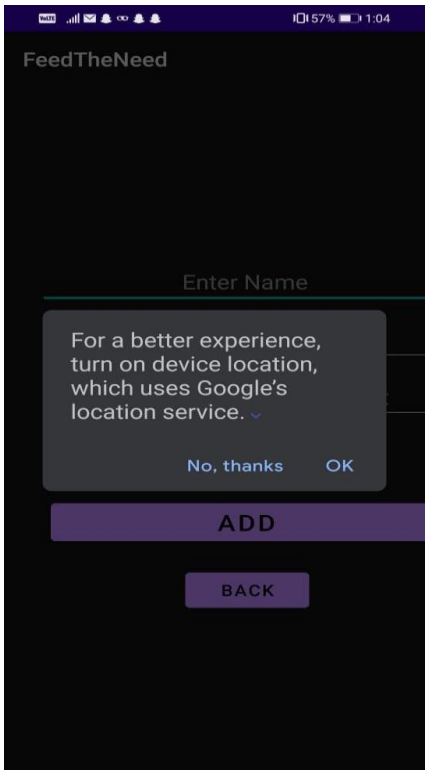
Feed The Need App Flow:

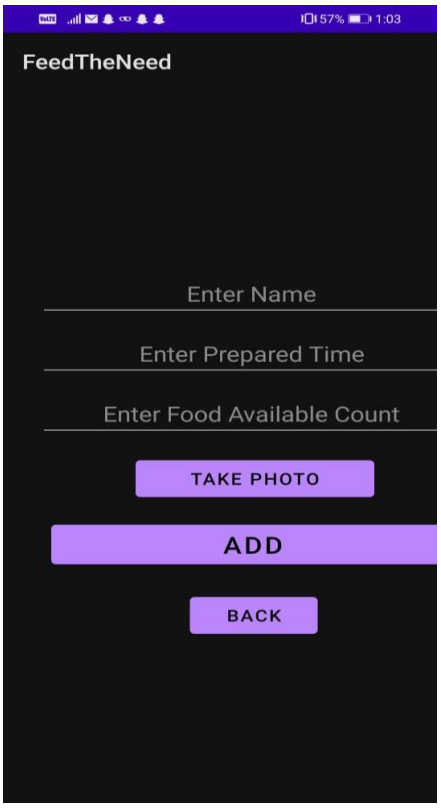
1. Donor logins with their credentials:



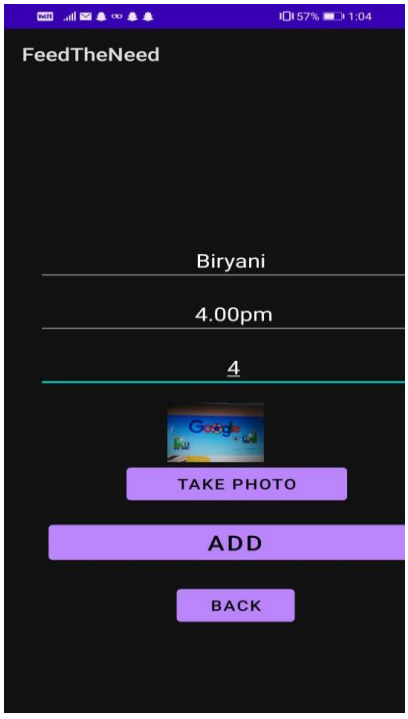
2. Donor Home Page:

3. If Donor clicks on add food and his location permission is accessed and no manual address entry is needed:

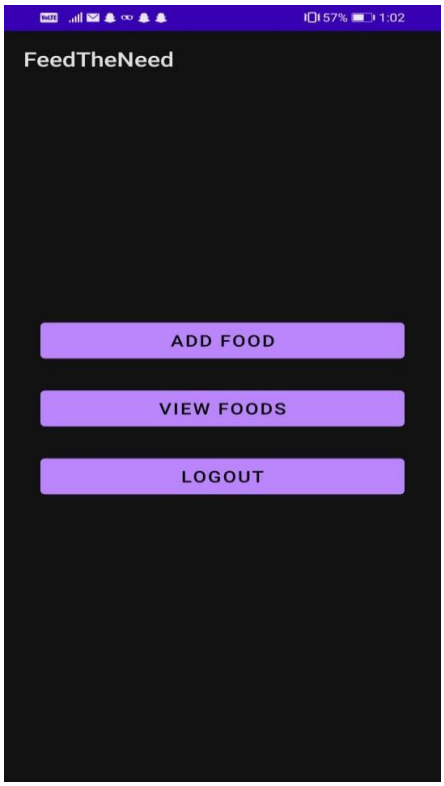


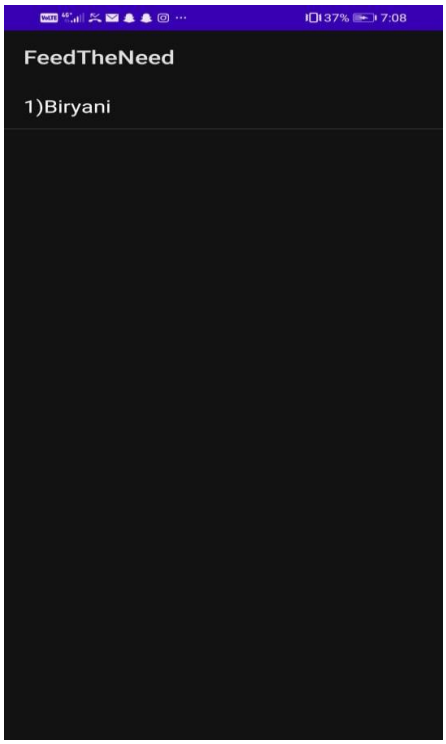


4. Food details are entered by the donor along with the photo.



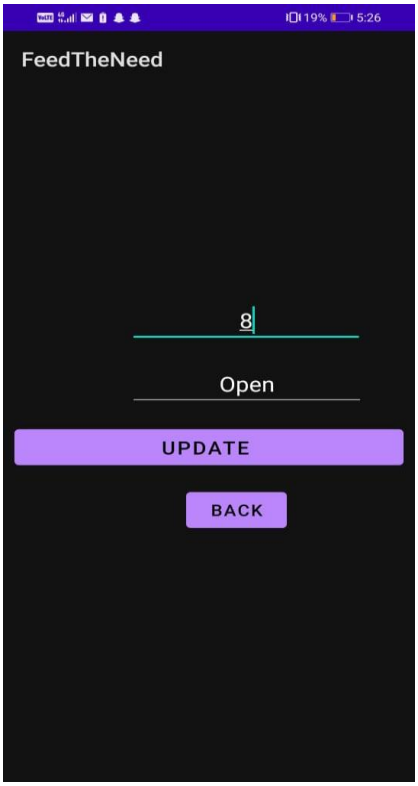
5. Donor added food can be viewed and updated by clicking on the view foods button:



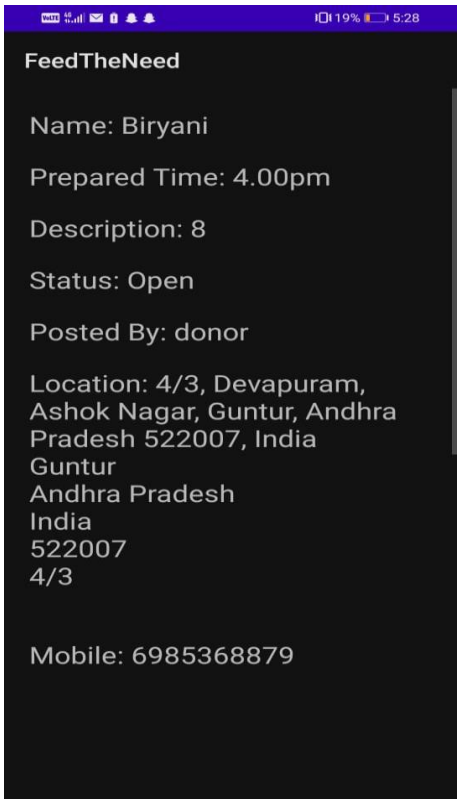


6. Once the donor selects the food he donated, all the details pertaining to it will be shown.

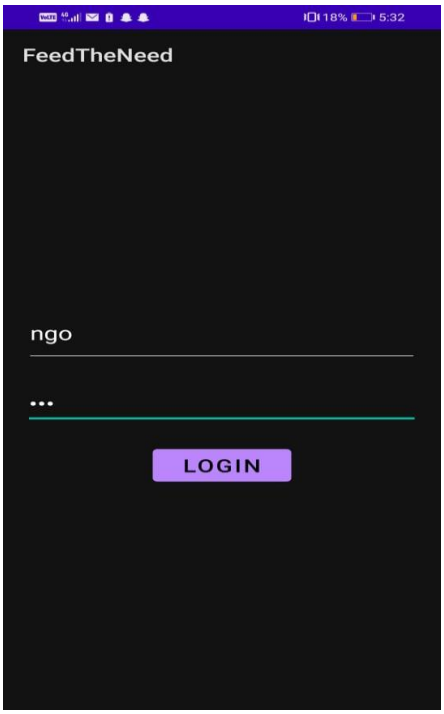
7. Update food can be clicked to update the quantity of food left and delete food will permanently delete the item.



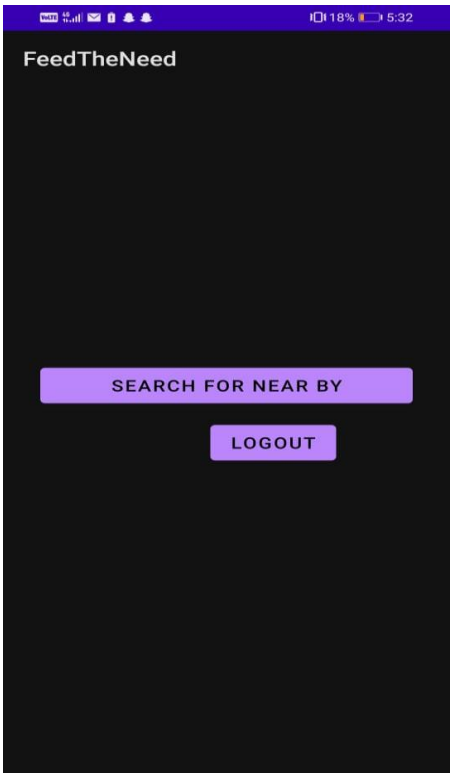
8. The updated food quantity will be shown in the app.



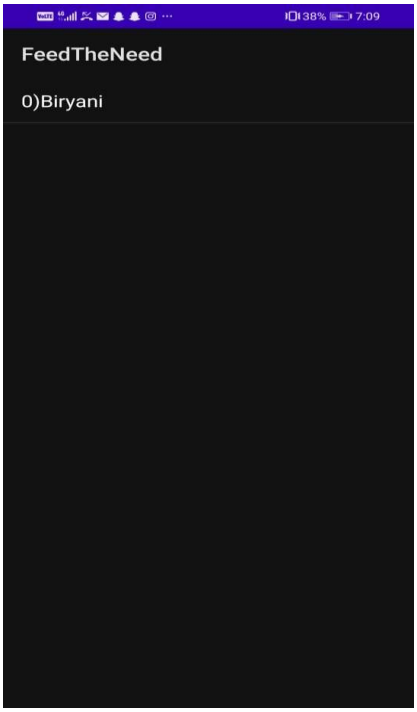
9. NGO will login with their credentials.



10. NGO Home Page

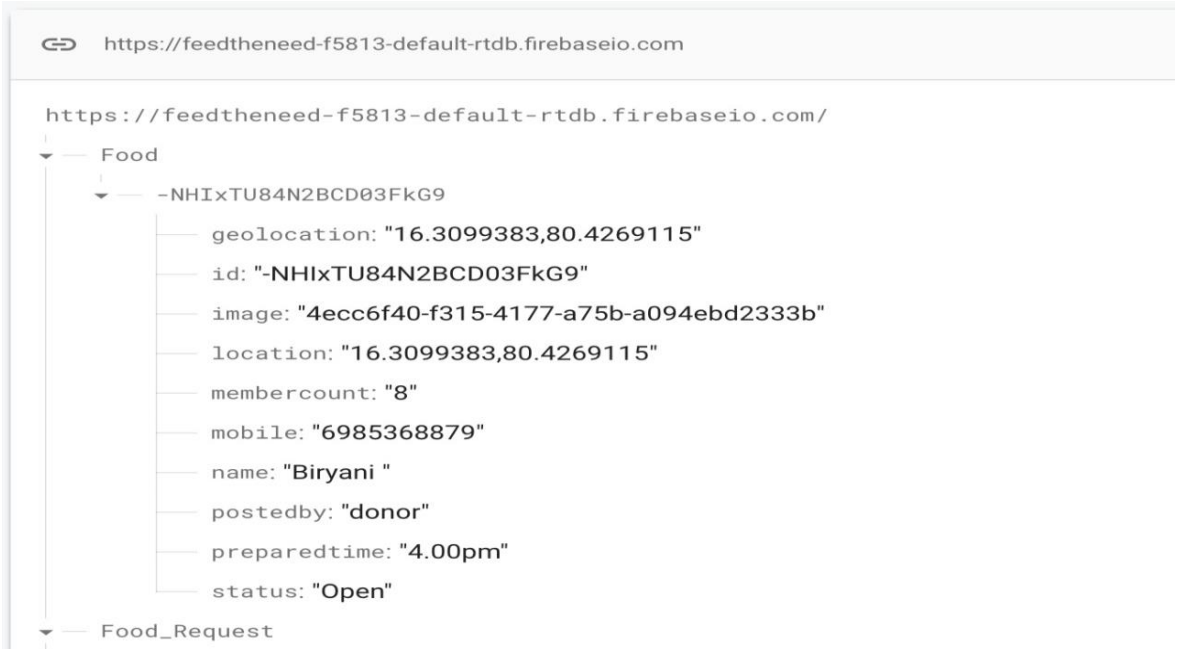


11. The search for a nearby button will show the nearby donated foods.



12. All the details pertaining to the food are shown along with a request option

Backend Firebase Database:



CHAPTER 5

CONCLUSION AND FUTURE WORK

Feed The Need has achieved all of the predetermined goals for building a food donation interface. The struggle against food hunger in India would drastically change if this application were to be used in real life. The project's future objectives include the addition of integrated maps for user convenience and search by location tools. We can add a special feature in the future where regular individuals may help by delivering the food.

CHAPTER 6

APPENDIX

activity_add_food.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<ScrollViewxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".view.AddFood">

<LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginLeft="20dp"
android:orientation="vertical"
android:paddingTop="175dp">

<EditText
android:id="@+id/addFoodName"
android:layout_width="352dp"
android:layout_height="wrap_content"
android:layout_marginLeft="10dp"
android:layout_marginBottom="10dp"
android:gravity="center"
android:hint="Enter Name" />

<EditText
android:id="@+id/addFoodPrepared"
android:layout_width="353dp"
android:layout_height="wrap_content"
android:layout_marginLeft="10dp"
android:layout_marginBottom="10dp"
android:gravity="center"
android:hint="Enter Prepared Time" />

<EditText
android:id="@+id/addFoodMemberCount"
android:layout_width="352dp"
android:layout_height="wrap_content"
android:layout_marginLeft="10dp"
android:layout_marginBottom="15dp"
android:gravity="center"
android:hint="Enter Food Available Count"
```

```
android:inputType="text" />

<ImageView
android:id="@+id/postFoodImgView"
android:layout_width="match_parent"
android:layout_height="match_parent" />

<Button
android:id="@+id/chooseimagebutton"
android:layout_width="198dp"
android:layout_height="wrap_content"
android:layout_marginLeft="90dp"
android:layout_marginBottom="15dp"
android:text="Take Photo" />

<Button
android:id="@+id/addFoodSubmit"
android:layout_width="335dp"
android:layout_height="wrap_content"
android:layout_marginLeft="20dp"
android:layout_marginBottom="20dp"
android:text="Add"
android:textSize="20sp" />

<Button
android:id="@+id/addFoodCancel"
android:layout_width="106dp"
android:layout_height="wrap_content"
android:layout_marginLeft="135dp"
android:text="Back" />

</LinearLayout>

</ScrollView>
```

activity_donor_home.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<ScrollViewxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:orientation="vertical"
android:layout_margin="20dp"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".view.DonorHome">

<LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:paddingTop="200dp">

<Button
android:id="@+id/donaraddfood"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="10dp"
android:text="Add Food" />

<Button
android:id="@+id/donarviewfood"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="10dp"
android:text="View Foods" />

<Button
android:id="@+id/donarlogout"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="10dp"
android:text="Logout" />
```

```
</LinearLayout>

</ScrollView>
```

activity_list_food_requests.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".view.ListFoodRequests">

<ListView
android:layout_width="match_parent"
android:layout_height="match_parent"
android:id="@+id/FoodRequestList"></ListView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_list_near_by.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".view.ListNearBy">

<ListView
android:layout_width="match_parent"
android:layout_height="match_parent"
android:id="@+id/FoodsNearbyList"></ListView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_login.xml:

```
</LinearLayout><?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:orientation="vertical"
android:layout_height="match_parent"
tools:context=".view.LoginActivity"
android:layout_margin="10dp"
android:gravity="center">

<EditText
android:id="@+id/loginPhone"
android:layout_width="323dp"
android:layout_height="67dp"
android:layout_marginBottom="20dp"
android:hint="Username" />

<EditText
android:id="@+id/loginPass"
android:layout_width="325dp"
android:layout_height="wrap_content"
android:layout_marginBottom="20dp"
android:hint="Password"
android:inputType="textPassword" />

<Button
android:id="@+id/loginConfirm"
android:layout_width="114dp"
android:layout_height="wrap_content"
android:text="Login"
```

```
android:textSize="20dp" />
```

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:paddingTop="200dp"
tools:context=".MainActivity">


<Button
android:id="@+id/registerButton"
android:layout_width="331dp"
android:layout_height="wrap_content"
android:layout_marginLeft="35dp"
android:text="Register"
android:textSize="20dp" />

<Button
android:id="@+id/loginButton"
android:layout_width="334dp"
android:layout_height="wrap_content"
android:layout_marginLeft="35dp"
android:layout_marginTop="20dp"
android:text="Login"
android:textSize="20dp" />



</LinearLayout>
```

activity_register.xml:

```
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:orientation="vertical"
android:layout_height="match_parent"
tools:context=".view.RegisterActivity"
android:layout_margin="10dp"
android:gravity="center">

<EditText
android:id="@+id/registerName"
android:layout_width="318dp"
android:layout_height="wrap_content"
android:layout_marginBottom="10dp"
android:gravity="center"
android:hint="Enter Name"
android:inputType="text" />

<EditText
android:id="@+id/registerUserName"
android:layout_width="317dp"
android:layout_height="wrap_content"
android:layout_marginBottom="10dp"
android:gravity="center"
android:hint="Enter User Name"
android:inputType="text" />

<EditText
android:id="@+id/registerPassword"
android:layout_width="319dp"
android:layout_height="wrap_content"
android:layout_marginBottom="10dp"
```

```
android:gravity="center"
android:hint="Enter password"
android:inputType="textPassword" />

<EditText
android:id="@+id/registerConPass"
android:layout_width="318dp"
android:layout_height="wrap_content"
android:layout_marginBottom="10dp"
android:gravity="center"
android:hint="Retype Password"
android:inputType="textPassword" />

<EditText
android:id="@+id/registerEmail"
android:layout_width="314dp"
android:layout_height="wrap_content"
android:layout_marginBottom="10dp"
android:gravity="center"
android:hint="Enter Email"
android:inputType="textEmailAddress" />

<EditText
android:id="@+id/registerMobile"
android:layout_width="315dp"
android:layout_height="wrap_content"
android:layout_marginBottom="10dp"
android:gravity="center"
android:hint="Enter Mobile"
android:inputType="phone" />


<EditText
android:id="@+id/registerAddress"
android:layout_width="310dp"
android:layout_height="wrap_content"
android:layout_marginBottom="10dp"
android:gravity="center"
android:hint="Enter Address" />

<TextView
android:layout_width="147dp"
android:layout_height="wrap_content"
android:layout_margin="10dp"
android:text="Select User Type:"
android:textSize="18sp" />

<Spinner
android:id="@+id/spinner"
android:layout_width="149dp"
android:layout_height="40dp"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.502"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.498" />

<Button
android:id="@+id/registerButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Register"
android:textSize="20sp" />

</LinearLayout>
```

activity_update_food.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<ScrollViewxmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:orientation="vertical"
android:layout_height="match_parent"
tools:context=".view.UpdateFood"
android:layout_margin="10dp"
android:gravity="center">

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
android:paddingTop="200dp">

<EditText
android:id="@+id/updateFoodMemberCount"
android:layout_width="205dp"
android:layout_height="wrap_content"
android:layout_marginLeft="100dp"
android:layout_marginBottom="15dp"
android:gravity="center"
android:hint="Enter Member Count" />

<EditText
android:id="@+id/updateFoodStatus"
android:layout_width="206dp"
android:layout_height="wrap_content"
android:layout_marginLeft="100dp"
android:layout_marginBottom="15dp"
android:gravity="center"
android:hint="Enter Status" />

<Button
android:id="@+id/updateFoodSubmit"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginBottom="15dp"
android:text="Update "
android:textSize="16sp" />

<Button
android:id="@+id/updateFoodCancel"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="150dp"
android:text="Back" />

</LinearLayout>

</ScrollView>
```

activity_user_home.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<ScrollViewxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:orientation="vertical"
android:layout_margin="20dp"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".view.NGOHome">

<LinearLayout
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
android:orientation="vertical"
android:paddingTop="250dp">

<Button
android:id="@+id/userviewnearbybutton"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="10dp"
android:text="Search for Near By" />

<Button
android:id="@+id/userlogout"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="147dp"
android:text="Logout" />

</LinearLayout>

</ScrollView>
```

activity_view_food.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollViewxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent">

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:orientation="vertical"
android:layout_margin="10dp"
android:gravity="center"
android:layout_height="match_parent"
tools:context=".view.ViewFood">

<TextView
android:textSize="23dp"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/foodviewname"/>

<TextView
android:textSize="23dp"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/foodviewpreparedtime"/>

<TextView
android:textSize="23dp"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/foodviewmembercount"/>

<TextView
android:textSize="23dp"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/foodviewstatus"/>

<TextView
android:textSize="23dp"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/foodviewpostedby"/>
```

```
<TextView
android:textSize="23dp"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/foodviewlocation"/>

<TextView
android:textSize="23dp"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/foodviewmobile"/>

<ImageView
android:id="@+id/foodviewimage"
android:layout_width="fill_parent"
android:layout_height="500dp"/>

<Button
android:id="@+id/updateFood"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Update"/>

<Button
android:id="@+id/sendfoodrequest"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Request Food"/>

<Button
android:id="@+id/viewFoodBack"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Cancel"/>

<Button
android:id="@+id/menuDeleteFood"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Delete Food"/>

</LinearLayout>

</ScrollView>
```

activity_view_food_request.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollViewxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent">

<LinearLayout
android:layout_width="match_parent"
android:orientation="vertical"
android:layout_margin="10dp"
android:gravity="center"
android:layout_height="match_parent"
tools:context=".view.ViewFoodRequest">

<TextView
android:textSize="23dp"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/foodrequestsource"/>
```

```
<TextView
android:textSize="23dp"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/foodrequestdestination"/>

<TextView
android:textSize="23dp"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/foodrequestsourcerequestedby"/>

<TextView
android:textSize="23dp"
android:layout_margin="10dp"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/foodrequestsourcedonatedby"/>

<Button
android:id="@+id/viewFoodrequestBack"
android:layout_width="118dp"
android:layout_height="wrap_content"
android:layout_margin="10dp"
android:text="Cancel" />

</LinearLayout>

</ScrollView>
```

MainActivity.java:

```
package com.example.FeedTheNeed;

import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.Manifest;
import android.content.DialogInterface;
import android.net.Uri;
import android.os.Build;
import android.provider.Settings;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.example.FeedTheNeed.view.LoginActivity;
import com.example.FeedTheNeed.view.RegisterActivity;

import java.util.ArrayList;
import java.util.List;

public class MainActivityextends AppCompatActivity {

    Button b1,b2,b3,b4;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
```

```
b1 = (Button) findViewById(R.id.loginButton);
b1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
if (permissionAlreadyGranted()) {
                    Intent i = new Intent(getApplicationContext(), LoginActivity.class);
startActivity(i);
}
requestPermission();
}
        });
b2 = (Button) findViewById(R.id.registerButton);
b2.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
                Intent i = new Intent(getApplicationContext(), RegisterActivity.class);
startActivity(i);
}
        });


}

private booleanpermissionAlreadyGranted() {

int result1 = ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA);
        int result2 = ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION);
        int result3 = ContextCompat.checkSelfPermission(this, Manifest.permission.INTERNET);
//int result4 = ContextCompat.checkSelfPermission(this, Manifest.permission.SYSTEM_ALERT_WINDOW);
int result5 = ContextCompat.checkSelfPermission(this, Manifest.permission.WRITE_EXTERNAL_STORAGE);
        int result6 = ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE);
        int result7 = ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_PHONE_STATE);
        int result8 = ContextCompat.checkSelfPermission(this, Manifest.permission.SEND_SMS);
        int result9 = ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION);

List l=new ArrayList();
l.add(result1);
l.add(result2);
l.add(result3);
//l.add(result4);
l.add(result5);
l.add(result6);
l.add(result7);
l.add(result8);
l.add(result9);

Log.v("results :",l.toString());
Log.v("Permission Granted :",PackageManager.PERMISSION_GRANTED+"");

        if (result1 == PackageManager.PERMISSION_GRANTED&&
result2==PackageManager.PERMISSION_GRANTED&& result3==PackageManager.PERMISSION_GRANTED&&
result5==PackageManager.PERMISSION_GRANTED&&result6==PackageManager.PERMISSION_GRANTED&&
result7==PackageManager.PERMISSION_GRANTED&& result8==PackageManager.PERMISSION_GRANTED&&
result9==PackageManager.PERMISSION_GRANTED) {
return true;
}
return false;
}

private void requestPermission() {

ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.CAMERA,Manifest.permission.ACCESS_FINE_LOCATION,Manifest.permission.I
NTERNET,Manifest.permission.SYSTEM_ALERT_WINDOW,Manifest.permission.WRITE_EXTERNAL_STORAGE,Manifes
t.permission.READ_EXTERNAL_STORAGE,Manifest.permission.READ_PHONE_STATE,Manifest.permission.SEND_S
MS,Manifest.permission.ACCESS_COARSE_LOCATION},1);
}

@RequiresApi(api = Build.VERSION_CODES.M)
```

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

Log.v("request code:",requestCode+"");

        for(String s : permissions)
        {
Log.v("permission:",s);
}

for(int i : grantResults)
        {
Log.v("result:",i+"");
}

if (requestCode == 1) {

if (grantResults.length>0 &&grantResults[0] == PackageManager.PERMISSION_GRANTED) {
Toast.makeText(this, "Permission granted successfully", Toast.LENGTH_SHORT).show();
} else {
Toast.makeText(this, "Permission is denied!", Toast.LENGTH_SHORT).show();
booleanshowRationale = shouldShowRequestPermissionRationale( Manifest.permission.CAMERA);
                if (! showRationale) {
openSettingsDialog();
}
            }
        }
    }

private void openSettingsDialog() {

AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
builder.setTitle("Required Permissions");
builder.setMessage("This app requires permission to automatically take your location. Kindly grant
them in app settings.");
builder.setPositiveButton("Take Me To SETTINGS", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
dialog.cancel();
Intent intent = new Intent(Settings.ACTION_APPLICATION_DETAILS_SETTINGS);
Uri uri = Uri.fromParts("package", getPackageName(), null);
intent.setData(uri);
startActivityForResult(intent, 101);
}
        });
builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
dialog.cancel();
}
        });
builder.show();
}
}
```

AddFood.java:

```
package com.example.FeedTheNeed.view;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.provider.MediaStore;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;
```

```
import android.Manifest;
import android.annotation.SuppressLint;
import android.content.pm.PackageManager;
import android.location.Location;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import com.example.FeedTheNeed.R;
import com.example.FeedTheNeed.dao.DAO;
import com.example.FeedTheNeed.form.Food;
import com.example.FeedTheNeed.form.NGO;
import com.example.FeedTheNeed.util.Constants;
import com.example.FeedTheNeed.util.Session;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.OnProgressListener;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import com.google.android.gms.location.FusedLocationProviderClient;
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 java.io.ByteArrayOutputStream;
import java.util.UUID;

public class AddFoodextends AppCompatActivity {

EditTextaddFoodName;
EditTextaddFoodPrepared;
EditTextaddMemeberCount;

    private ImageViewimageView;

Button addFoodSubmit;
Button addFoodCancel;
Button btnChoose;

    private static final int Image_Capture_Code= 1;
    private Uri imageUri;

    private FusedLocationProviderClientmFusedLocationClient;

    private double wayLatitude= 0.0, wayLongitude= 0.0;
    private LocationRequestlocationRequest;
    private LocationCallbacklocationCallback;
    private String txtLocation;
    private String lat;
    private String lang;

    private booleanisGPS= false;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

setContentView(R.layout.activity_add_food);

mFusedLocationClient= LocationServices.getFusedLocationProviderClient(this);

locationRequest= LocationRequest.create();
locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
locationRequest.setInterval(10 * 1000); // 10 seconds
locationRequest.setFastestInterval(5 * 1000); // 5 seconds

new GpsUtils(this).turnGPSOn(new GpsUtils.onGpsListener() {
@Override
```

```
public void gpsStatus(booleanisGPSEnable) {
// turn on GPS
isGPS= isGPSEnable;
}
        });

locationCallback= new LocationCallback() {
@Override
public void onLocationResult(LocationResultlocationResult) {
if (locationResult == null) {
return;
}
for (Location location : locationResult.getLocations()) {
if (location != null) {
wayLatitude= location.getLatitude();
wayLongitude= location.getLongitude();

txtLocation= wayLatitude+ "," + wayLongitude;

                        if (mFusedLocationClient!= null) {
mFusedLocationClient.removeLocationUpdates(locationCallback);
}
                    }
                }
            }
        };

        final Session s = new Session(getApplicationContext());

addFoodName= (EditText) findViewById(R.id.addFoodName);
addFoodPrepared= (EditText) findViewById(R.id.addFoodPrepared);
addMemeberCount= (EditText) findViewById(R.id.addFoodMemberCount);

btnChoose= (Button) findViewById(R.id.chooseimagebutton);
addFoodSubmit= (Button) findViewById(R.id.addFoodSubmit);
addFoodCancel= (Button) findViewById(R.id.addFoodCancel);

imageView= (ImageView) findViewById(R.id.postFoodImgView);

btnChoose.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {

                Intent cInt = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
startActivityForResult(cInt, Image_Capture_Code);
}
        });

addFoodSubmit.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {

                DAO d=new DAO();
d.getDBReference(Constants.USER_DB).child(s.getusename()).addListenerForSingleValueEvent(new
ValueEventListener() {
@Override
public void onDataChange(@NonNull DataSnapshot snapshot) {

                        NGO user=snapshot.getValue(NGO.class);

                        if(user!=null)
                        {
Log.v("food ", "1");
                            if (!isGPS) {
Toast.makeText(getApplicationContext(), "Please turn on GPS", Toast.LENGTH_SHORT).show();
                                return;
}

Log.v("food ", "2");

getLocation();

                            final Session s = new Session(getApplicationContext());
```

```
Log.v("food ", txtLocation+ "");

Log.v("food ", "3");

                            if (txtLocation!= null &&lat!=null &lang!=null) {
Log.v("food ", "4");

String name = addFoodName.getText().toString();
String prepared = addFoodPrepared.getText().toString();
String memberCount = addMemeberCount.getText().toString();

                                if (name == null || prepared == null || memberCount == null) {
Toast.makeText(getApplicationContext(), "Please Enter Valid Data", Toast.LENGTH_SHORT).show();
} else {

Log.v("food ", "5");

String imageName = UUID.randomUUID().toString();

DAO dao = new DAO();

Food food = new Food();
food.setId(dao.getUnicKey(Constants.FOOD_DB));
food.setName(name);
food.setPreparedtime(prepared);
food.setMembercount(memberCount);
food.setPostedby(s.getusename());
food.setStatus("open");
food.setGeolocation(lat+","+lang);
food.setImage(imageName);
food.setLocation(txtLocation);
food.setMobile(user.getMobile());

                                    try {

dao.addObject(Constants.FOOD_DB, food, food.getId());
uploadImage(imageName);

Intent i = new Intent(getApplicationContext(), DonorHome.class);
startActivity(i);
Toast.makeText(getApplicationContext(), "Food Posted Successfully", Toast.LENGTH_SHORT).show();
} catch (Exception ex) {
Toast.makeText(getApplicationContext(), "Food Failed", Toast.LENGTH_SHORT).show();
ex.printStackTrace();
}
                                }
                            } else {
Log.v("food ", "8");
}
                        }
                    }

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

                    }
                });
}
        });

addFoodCancel.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {

                Intent i = new Intent(getApplicationContext(), DonorHome.class);
startActivity(i);
}
        });
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
super.onActivityResult(requestCode, resultCode, data);
```

```
Log.v("onemeal", "in activity results return");

        if (requestCode == Image_Capture_Code) {
if (resultCode == RESULT_OK) {
Log.v("onemeal", "result ok");
Bitmap bp = (Bitmap) data.getExtras().get("data");
imageView.setImageBitmap(bp);
imageUri= getImageUri(getApplicationContext(), bp);
} else if (resultCode == RESULT_CANCELED) {
Log.v("onemeal", "result cancelled");
Toast.makeText(this, "Cancelled", Toast.LENGTH_LONG).show();
}
        }
Log.v("onemeal", "at last");

        if (resultCode == Activity.RESULT_OK) {
if (requestCode == AppConstants.GPS_REQUEST) {
isGPS= true; // flag maintain before get location
}
        }
    }

private void uploadImage(String fileName) {

final ProgressDialogprogressDialog = new ProgressDialog(this);
progressDialog.setTitle("Uploading...");
progressDialog.show();

StorageReference ref = DAO.getStorageReference().child("images/" + fileName);

ref.putFile(imageUri)
                .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
@Override
public void onSuccess(UploadTask.TaskSnapshottaskSnapshot) {
progressDialog.dismiss();
}
                })
                .addOnFailureListener(new OnFailureListener() {
@Override
public void onFailure(@NonNull Exception e) {
progressDialog.dismiss();
Toast.makeText(getApplicationContext(), "Failed " + e.getMessage(), Toast.LENGTH_SHORT).show();
}
                })
                .addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {
@Override
public void onProgress(UploadTask.TaskSnapshottaskSnapshot) {
double progress = (100.0 * taskSnapshot.getBytesTransferred() / taskSnapshot
                                .getTotalByteCount());
progressDialog.setMessage("Uploaded " + (int) progress + "%");
}
                });
}

public Uri getImageUri(Context inContext, Bitmap inImage) {
ByteArrayOutputStream bytes = new ByteArrayOutputStream();
inImage.compress(Bitmap.CompressFormat.JPEG, 100, bytes);
String path = MediaStore.Images.Media.insertImage(inContext.getContentResolver(), inImage,
"Title", null);
        return Uri.parse(path);
}

private void getLocation() {
if (ActivityCompat.checkSelfPermission(AddFood.this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED
&&ActivityCompat.checkSelfPermission(AddFood.this, Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
ActivityCompat.requestPermissions(AddFood.this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION, Manifest.permission.ACCESS_COARSE_LOCATION},
AppConstants.LOCATION_REQUEST);
} else {

mFusedLocationClient.getLastLocation().addOnSuccessListener(AddFood.this, new
OnSuccessListener<Location>() {
```

```
@Override
public void onSuccess(Location location) {

if (location != null) {
wayLatitude= location.getLatitude();
wayLongitude= location.getLongitude();
lat=wayLatitude+"";
lang=wayLongitude+"";
txtLocation= wayLatitude+ "," + wayLongitude;
} else {
if (ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED&&ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
// TODO: Consider calling
//    ActivityCompat#requestPermissions
// here to request the missing permissions, and then overriding
                            //   public void onRequestPermissionsResult(int requestCode, String[]
permissions,
                            //                                          int[] grantResults)
                            // to handle the case where the user grants the permission. See the
documentation
                            // for ActivityCompat#requestPermissions for more details.
return;
}
mFusedLocationClient.requestLocationUpdates(locationRequest, locationCallback, null);
}
                }
            });
}
    }

@SuppressLint("MissingPermission")
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull
int[] grantResults) {
super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        switch (requestCode) {
case 1000: {
// If request is cancelled, the result arrays are empty.
if (grantResults.length>0
&&grantResults[0] == PackageManager.PERMISSION_GRANTED) {

mFusedLocationClient.getLastLocation().addOnSuccessListener(AddFood.this, new
OnSuccessListener<Location>() {
@Override
public void onSuccess(Location location) {
if (location != null) {
wayLatitude= location.getLatitude();
wayLongitude= location.getLongitude();
txtLocation=wayLatitude+","+wayLongitude;
} else {
mFusedLocationClient.requestLocationUpdates(locationRequest, locationCallback, null);
}
                        }
                    });

} else {
Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show();
}
break;
}
        }
    }
}
```

AppConstants.java:

```
package com.example.FeedTheNeed.view;

public class AppConstants {

public static final int LOCATION_REQUEST = 1000;
    public static final int GPS_REQUEST = 1001;
}
```

DonorHome.java:

```
package com.example.FeedTheNeed.view;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

import com.example.FeedTheNeed.MainActivity;
import com.example.FeedTheNeed.R;
import com.example.FeedTheNeed.util.Session;

public class DonorHomeextends AppCompatActivity {

    Button addfood;
Button donarLogout;
Button viewfood;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_donor_home);

addfood=(Button) findViewById(R.id.donaraddfood);
viewfood=(Button) findViewById(R.id.donarviewfood);
donarLogout=(Button) findViewById(R.id.donarlogout);

        final Session s = new Session(getApplicationContext());

addfood.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {

                Intent i = new Intent(getApplicationContext(),AddFood.class);
startActivity(i);
}
        });

viewfood.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
                Intent i = new Intent(getApplicationContext(),ListFood.class);
startActivity(i);
}
        });

donarLogout.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
s.loggingOut();
Intent i = new Intent(getApplicationContext(), MainActivity.class);
startActivity(i);
}
        });
}
```

GpsUtils.java:

```
package com.example.FeedTheNeed.view;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Context;
import android.content.IntentSender;
import android.location.LocationManager;
import android.util.Log;
```

```
import android.widget.Toast;

import androidx.annotation.NonNull;

import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.common.api.ResolvableApiException;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.location.LocationSettingsRequest;
import com.google.android.gms.location.LocationSettingsResponse;
import com.google.android.gms.location.LocationSettingsStatusCodes;
import com.google.android.gms.location.SettingsClient;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;

import static android.content.ContentValues.TAG;

public class GpsUtils {

private Context context;
    private SettingsClientmSettingsClient;
    private LocationSettingsRequestmLocationSettingsRequest;
    private LocationManagerlocationManager;
    private LocationRequestlocationRequest;

    public GpsUtils(Context context) {
this.context= context;
locationManager= (LocationManager) context.getSystemService(Context.LOCATION_SERVICE);
mSettingsClient= LocationServices.getSettingsClient(context);

locationRequest= LocationRequest.create();
locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
locationRequest.setInterval(10 * 1000);
locationRequest.setFastestInterval(2 * 1000);
LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder()
                .addLocationRequest(locationRequest);
mLocationSettingsRequest= builder.build();

//**************************
builder.setAlwaysShow(true); //this is the key ingredient
        //**************************
}

// method for turn on GPS
public void turnGPSOn(final onGpsListeneronGpsListener) {

if (locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
if (onGpsListener != null) {
onGpsListener.gpsStatus(true);
}
        } else {
mSettingsClient
.checkLocationSettings(mLocationSettingsRequest)
                    .addOnSuccessListener((Activity) context, new
OnSuccessListener<LocationSettingsResponse>() {
@SuppressLint("MissingPermission")
@Override
public void onSuccess(LocationSettingsResponselocationSettingsResponse) {

//  GPS is already enable, callback GPS status through listener
if (onGpsListener!= null) {
onGpsListener.gpsStatus(true);
}
                        }
                    })
                    .addOnFailureListener((Activity) context, new OnFailureListener() {
@Override
public void onFailure(@NonNull Exception e) {
int statusCode = ((ApiException) e).getStatusCode();
                            switch (statusCode) {
case LocationSettingsStatusCodes.RESOLUTION_REQUIRED:

try {
// Show the dialog by calling startResolutionForResult(), and check the
```

```
                                        // result in onActivityResult().
ResolvableApiExceptionrae = (ResolvableApiException) e;
rae.startResolutionForResult((Activity) context, AppConstants.GPS_REQUEST);
} catch (IntentSender.SendIntentExceptionsie) {
Log.i(TAG, "PendingIntent unable to execute request.");
}
break;
                                case LocationSettingsStatusCodes.SETTINGS_CHANGE_UNAVAILABLE:
                                    String errorMessage = "Location settings are inadequate, and
cannot be " +
"fixed here. Fix in Settings.";
Log.e(TAG, errorMessage);

Toast.makeText((Activity) context, errorMessage, Toast.LENGTH_LONG).show();
}
                        }
                    });
}
    }

public interface onGpsListener {
void gpsStatus(booleanisGPSEnable);
}
}
```

ListFood.java:

```
package com.example.FeedTheNeed.view;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;

import com.example.FeedTheNeed.R;
import com.example.FeedTheNeed.dao.DAO;
import com.example.FeedTheNeed.form.Food;
import com.example.FeedTheNeed.util.Constants;
import com.example.FeedTheNeed.util.MapUtil;
import com.example.FeedTheNeed.util.Session;

public class ListFoodextends AppCompatActivity {

ListViewlistView;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_list_food);

listView=(ListView) findViewById(R.id.FoodList);
        final Session s=new Session(getApplicationContext());

        final DAO dao=new DAO();

dao.setDataToAdapterList(listView, Food.class, Constants.FOOD_DB,"employee");

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

                String item = listView.getItemAtPosition(position).toString();
item= MapUtil.stringToMap(s.getViewMap()).get(item);

Intent intent=new Intent(getApplicationContext(),ViewFood.class);
intent.putExtra("foodid",item);
startActivity(intent);
}
        });
}
}
```

ListFoodRequests.java:

```
package com.example.FeedTheNeed.view;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;

import com.example.FeedTheNeed.R;
import com.example.FeedTheNeed.dao.DAO;
import com.example.FeedTheNeed.form.FoodRequest;
import com.example.FeedTheNeed.util.Constants;
import com.example.FeedTheNeed.util.MapUtil;
import com.example.FeedTheNeed.util.Session;

public class ListFoodRequestsextends AppCompatActivity {

ListViewlistView;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_list_food_requests);

listView=(ListView) findViewById(R.id.FoodRequestList);
        final Session s=new Session(getApplicationContext());

        final DAO dao=new DAO();

dao.setDataToAdapterList(listView, FoodRequest.class, Constants.FOOD_REQUESTS_DB,"food request");

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

                String item = listView.getItemAtPosition(position).toString();
item= MapUtil.stringToMap(s.getViewMap()).get(item);

Intent intent=new Intent(getApplicationContext(),ViewFoodRequest.class);
intent.putExtra("foodrequestid",item);
startActivity(intent);
}
        });
}
}
```

ListNearBy.java:

```
package com.example.FeedTheNeed.view;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.FeedTheNeed.R;
import com.example.FeedTheNeed.dao.DAO;
import com.example.FeedTheNeed.form.Food;
import com.example.FeedTheNeed.util.Constants;
import com.example.FeedTheNeed.util.MapUtil;
import com.example.FeedTheNeed.util.Session;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
```

```
import java.util.HashMap;
import java.util.Map;

public class ListNearByextends AppCompatActivity {

ListViewlistView;

@Override
protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_list_near_by);

listView=(ListView) findViewById(R.id.FoodsNearbyList);
        final Session s=new Session(getApplicationContext());

Intent i=getIntent();
savedInstanceState=i.getExtras();

        final String[] userLatLongs=savedInstanceState.getString("latlang").split(",");

        final DAO dao=new DAO();

//dao.setDataToAdapterList(listView, Food.class, Constants.COMPLAINTS_DB,"user");

DAO d=new DAO();
d.getDBReference(Constants.FOOD_DB).addValueEventListener(new ValueEventListener() {
@Override
public void onDataChange(DataSnapshotdataSnapshot) {

final Map<String,Object> map=new HashMap<String,Object>();
                final Map<String,String>viewMap=new HashMap<String,String>();

                int i=0;

                for (DataSnapshotsnapshotNode: dataSnapshot.getChildren()) {

                    Food food=(Food)snapshotNode.getValue(Food.class);

                    if(food!=null)
                    {
                        String[] latLongs=food.getGeolocation().split(",");

                        float distance = getDistanceFromCurrentPosition(new
Double(userLatLongs[0]), new Double(userLatLongs[1]), new Double(latLongs[0]), new
Double(latLongs[1]));

                        if(distance<10000)
                        {
viewMap.put(i + ")"+food.getName(),food.getId());
i++;

                            if(i==10)
                            {
break;
}
                        }
                    }
                }

ArrayList<String> al=new ArrayList<String>(viewMap.keySet());
ArrayAdapter<String> adapter = new ArrayAdapter<String>(listView.getContext(),
android.R.layout.simple_list_item_1, (al.toArray(new String[al.size()])));

listView.setAdapter(adapter);
s.setViewMap(MapUtil.mapToString(viewMap));
}

@Override
public void onCancelled(DatabaseErrordatabaseError) {

            }
        });
```

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?>adapterView, View view, int i, long l) {

                String item = listView.getItemAtPosition(i).toString();
item= MapUtil.stringToMap(s.getViewMap()).get(item);

Intent intent=new Intent(getApplicationContext(),ViewFood.class);
intent.putExtra("foodid",item);
startActivity(intent);
}
        });
}

public static float getDistanceFromCurrentPosition(double lat1,double lng1, double lat2, double
lng2)
    {
double earthRadius = 3958.75;

        double dLat = Math.toRadians(lat2 - lat1);

        double dLng = Math.toRadians(lng2 - lng1);

        double a = Math.sin(dLat / 2) * Math.sin(dLat / 2)
                + Math.cos(Math.toRadians(lat1))
                * Math.cos(Math.toRadians(lat2)) * Math.sin(dLng / 2)
                * Math.sin(dLng / 2);

        double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));

        double dist = earthRadius * c;

        int meterConversion = 1609;

        return new Float(dist * meterConversion).floatValue();

}
}
```

LoginActivity.java:

```
package com.example.FeedTheNeed.view;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.FeedTheNeed.R;
import com.example.FeedTheNeed.dao.DAO;
import com.example.FeedTheNeed.form.NGO;
import com.example.FeedTheNeed.util.Constants;
import com.example.FeedTheNeed.util.Session;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.ValueEventListener;

public class
LoginActivityextends AppCompatActivity {

private Session session;
EditTexte1,e2;
Button b1;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
session = new Session(getApplicationContext());

setContentView(R.layout.activity_login);

e1=(EditText)findViewById(R.id.loginPhone);
```

```
e2=(EditText)findViewById(R.id.loginPass);
b1=(Button)findViewById(R.id.loginConfirm);

b1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {

final String username=e1.getText().toString();
                final String password=e2.getText().toString();

                if(username==null|| password==null || username.length()<=0|| password.length()<=0)
                {
Toast.makeText(getApplicationContext(),"Please Enter Username and
Password",Toast.LENGTH_SHORT).show();
}
else {

                    DAO d = new DAO();
d.getDBReference(Constants.USER_DB).child(username).addListenerForSingleValueEvent(new
ValueEventListener() {

@Override
public void onDataChange(DataSnapshotdataSnapshot) {

                            NGO user = (NGO) dataSnapshot.getValue(NGO.class);

                            if (user == null) {
Toast.makeText(getApplicationContext(), "Invalid Username ", Toast.LENGTH_SHORT).show();
} else if (user != null &&user.getPassword().equals(password)) {

session.setusename(user.getUsername());
session.setRole(user.getType());

Intent i = null;

                                if (user.getType().equals("DONOR")){
i = new Intent(getApplicationContext(), DonorHome.class);
}else if (user.getType().equals("NGO")){
i = new Intent(getApplicationContext(), NGOHome.class);
}

startActivity(i);

} else {
Toast.makeText(getApplicationContext(), "Invalid Password", Toast.LENGTH_SHORT).show();
}
                        }

@Override
public void onCancelled(DatabaseErrordatabaseError) {

                        }
                    });
}
            }
        });
}
}
```

NGOHome.java:

```
package com.example.FeedTheNeed.view;

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
```

```
import androidx.core.app.ActivityCompat;

import com.example.FeedTheNeed.MainActivity;
import com.example.FeedTheNeed.R;
import com.example.FeedTheNeed.util.Session;
import com.google.android.gms.location.FusedLocationProviderClient;
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.OnSuccessListener;

public class NGOHomeextends AppCompatActivity {

    Button userlogout;
Button nearby;

    private FusedLocationProviderClientmFusedLocationClient;

    private double wayLatitude= 0.0, wayLongitude= 0.0;
    private LocationRequestlocationRequest;
    private LocationCallbacklocationCallback;
    private String txtLocation;
    private String lat;
    private String lang;

    private booleanisGPS= false;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_user_home);

userlogout=(Button) findViewById(R.id.userlogout);
nearby = (Button) findViewById(R.id.userviewnearbybutton);

mFusedLocationClient= LocationServices.getFusedLocationProviderClient(this);

locationRequest= LocationRequest.create();
locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
locationRequest.setInterval(10 * 1000); // 10 seconds
locationRequest.setFastestInterval(5 * 1000); // 5 seconds

new GpsUtils(this).turnGPSOn(new GpsUtils.onGpsListener() {
@Override
public void gpsStatus(booleanisGPSEnable) {
// turn on GPS
isGPS= isGPSEnable;
}
        });

locationCallback= new LocationCallback() {
@Override
public void onLocationResult(LocationResultlocationResult) {
if (locationResult == null) {
return;
}
for (Location location : locationResult.getLocations()) {
if (location != null) {
wayLatitude= location.getLatitude();
wayLongitude= location.getLongitude();

txtLocation= wayLatitude+ "," + wayLongitude;

                        if (mFusedLocationClient!= null) {
mFusedLocationClient.removeLocationUpdates(locationCallback);
}
                    }
                }
            }
        };

        final Session s = new Session(getApplicationContext());
```

```
nearby.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
getLocation();
                if (txtLocation!= null &&lat!=null &lang!=null) {
                    Intent i = new Intent(getApplicationContext(), ListNearBy.class);
i.putExtra("latlang",lat+","+lang);
startActivity(i);
}
            }
        });

userlogout.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {

s.loggingOut();
Intent i = new Intent(getApplicationContext(), MainActivity.class);
startActivity(i);
}
        });
}

private void getLocation() {
if (ActivityCompat.checkSelfPermission(NGOHome.this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED
&&ActivityCompat.checkSelfPermission(NGOHome.this, Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
ActivityCompat.requestPermissions(NGOHome.this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION, Manifest.permission.ACCESS_COARSE_LOCATION},
AppConstants.LOCATION_REQUEST);
} else {

mFusedLocationClient.getLastLocation().addOnSuccessListener(NGOHome.this, new
OnSuccessListener<Location>() {
@Override
public void onSuccess(Location location) {

if (location != null) {
wayLatitude= location.getLatitude();
wayLongitude= location.getLongitude();
lat=wayLatitude+"";
lang=wayLongitude+"";
txtLocation= wayLatitude+ "," + wayLongitude;
} else {
if (ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED&&ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
// TODO: Consider calling
//    ActivityCompat#requestPermissions
// here to request the missing permissions, and then overriding
                            //   public void onRequestPermissionsResult(int requestCode, String[]
permissions,
                            //                                          int[] grantResults)
                            // to handle the case where the user grants the permission. See the
documentation
                            // for ActivityCompat#requestPermissions for more details.
return;
}
mFusedLocationClient.requestLocationUpdates(locationRequest, locationCallback, null);
}
                }
            });
}
    }

@SuppressLint("MissingPermission")
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull
int[] grantResults) {
super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        switch (requestCode) {
case 1000: {
```

```
// If request is cancelled, the result arrays are empty.
if (grantResults.length>0
&&grantResults[0] == PackageManager.PERMISSION_GRANTED) {

mFusedLocationClient.getLastLocation().addOnSuccessListener(NGOHome.this, new
OnSuccessListener<Location>() {
@Override
public void onSuccess(Location location) {
if (location != null) {
wayLatitude= location.getLatitude();
wayLongitude= location.getLongitude();
txtLocation=wayLatitude+","+wayLongitude;
} else {
mFusedLocationClient.requestLocationUpdates(locationRequest, locationCallback, null);
}
                        }
                    });

} else {
Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show();
}
break;
}
        }
    }
}
```

RegisterActivity.java:

```
package com.example.FeedTheNeed.view;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.FeedTheNeed.R;
import com.example.FeedTheNeed.dao.DAO;
import com.example.FeedTheNeed.form.NGO;
import com.example.FeedTheNeed.util.Constants;

import java.util.regex.Pattern;

public class RegisterActivityextends AppCompatActivityimplements
AdapterView.OnItemSelectedListener{

EditTexte1,e2,e3,e4,e5,e6,e7;
Button b1;

String[] userTypes={"NGO","DONOR"};
String type;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

setContentView(R.layout.activity_register);

e1=(EditText)findViewById(R.id.registerUserName);
e2=(EditText)findViewById(R.id.registerPassword);
e3=(EditText)findViewById(R.id.registerConPass);
e4=(EditText)findViewById(R.id.registerEmail);
e5=(EditText)findViewById(R.id.registerMobile);
e6=(EditText)findViewById(R.id.registerName);
e7=(EditText)findViewById(R.id.registerAddress);
```

```
Spinner spin = (Spinner) findViewById(R.id.spinner);
spin.setOnItemSelectedListener(this);

//Creating the ArrayAdapter instance having the country list
ArrayAdapter aa = new
ArrayAdapter(getApplicationContext(),android.R.layout.simple_spinner_item,userTypes);
aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
//Setting the ArrayAdapter data on the Spinner
spin.setAdapter(aa);

b1=(Button)findViewById(R.id.registerButton);

b1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {

                String username=e1.getText().toString();
String password=e2.getText().toString();
String conformPassword=e3.getText().toString();
String email=e4.getText().toString();
String mobile=e5.getText().toString();
String name=e6.getText().toString();
String address=e7.getText().toString();

                if(username==null|| password==null|| conformPassword==null|| email==null||
mobile==null|| name==null|| address==null)
                {
Toast.makeText(getApplicationContext(),"Please Enter Valid Data",Toast.LENGTH_SHORT).show();
}
else if(mobile.length()<10|| mobile.length()>12)
                {
Toast.makeText(getApplicationContext(),"Invalid Mobile",Toast.LENGTH_SHORT).show();
}
else if(!password.equals(conformPassword))
                {
Toast.makeText(getApplicationContext(),"Password Mismatch",Toast.LENGTH_SHORT).show();
}
else if(!isValid(email))
                {
Toast.makeText(getApplicationContext(),"Invalid Email",Toast.LENGTH_SHORT).show();
}
else
{
                    NGO user=new NGO();

user.setUsername(username);
user.setPassword(password);
user.setEmail(email);
user.setMobile(mobile);
user.setName(name);
user.setAddress(address);
user.setType(type);

DAO dao=new DAO();

                    try
{
dao.addObject(Constants.USER_DB,user,user.getUsername());
Toast.makeText(getApplicationContext(),"Registration Successful",Toast.LENGTH_SHORT).show();
Intent i=new Intent(getApplicationContext(),LoginActivity.class);
startActivity(i);
}
catch (Exception ex)
                    {
Toast.makeText(getApplicationContext(),"Registration Error",Toast.LENGTH_SHORT).show();
Log.v("User Registration Ex", ex.toString());
ex.printStackTrace();
}
                }
            }
        });
}
```

```
//Performing action onItemSelected and onNothing selected
@Override
public void onItemSelected(AdapterView<?> arg0, View arg1, int position, long id) {
type=userTypes[position];
}
@Override
public void onNothingSelected(AdapterView<?> arg0) {
// TODO Auto-generated method stub
}

public static booleanisValid(String email)
    {
        String emailRegex = "^[a-zA-Z0-9_+&*-]+(?:\\."+
"[a-zA-Z0-9_+&*-]+)*@" +
"(?:[a-zA-Z0-9-]+\\.)+[a-z" +
"A-Z]{2,7}$";

Pattern pat = Pattern.compile(emailRegex);
        if (email == null)
return false;
        return pat.matcher(email).matches();
}
}
```

UpdateFood.java:

```
package com.example.FeedTheNeed.view;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.FeedTheNeed.R;
import com.example.FeedTheNeed.dao.DAO;
import com.example.FeedTheNeed.form.Food;
import com.example.FeedTheNeed.util.Constants;
import com.example.FeedTheNeed.util.Session;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.ValueEventListener;

public class UpdateFoodextends AppCompatActivity {

EditTextupdateFoodStatus;
EditTextupdateFoodCount;
Button updateFoodSubmit;
Button updateFoodCancel;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

setContentView(R.layout.activity_update_food);

        final Session s = new Session(getApplicationContext());

updateFoodCount= (EditText) findViewById(R.id.updateFoodMemberCount);
updateFoodStatus= (EditText) findViewById(R.id.updateFoodStatus);
updateFoodSubmit= (Button) findViewById(R.id.updateFoodSubmit);
updateFoodCancel= (Button) findViewById(R.id.updateFoodCancel);

Intent i = getIntent();
savedInstanceState = i.getExtras();

        final String foodid = savedInstanceState.getString("foodid");

updateFoodSubmit.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
```

```
                String status = updateFoodStatus.getText().toString();
String count = updateFoodCount.getText().toString();

                if (status == null || count ==null) {
Toast.makeText(getApplicationContext(), "Please Enter Food Status or Count",
Toast.LENGTH_SHORT).show();
} else {

                    DAO dao = new DAO();
dao.getDBReference(Constants.FOOD_DB).child(foodid).addListenerForSingleValueEvent(new
ValueEventListener() {

@Override
public void onDataChange(DataSnapshotdataSnapshot) {

                            Food food=dataSnapshot.getValue(Food.class);

                            if(food!=null)
                            {
food.setStatus(status);
food.setMembercount(count);
dao.addObject(Constants.FOOD_DB,food,foodid);

Intent i = new Intent(getApplicationContext(), DonorHome.class);
startActivity(i);
}
                        }

@Override
public void onCancelled(DatabaseErrordatabaseError) {

                        }
                    });
}
            }
        });

updateFoodCancel.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {

                Intent i = new Intent(getApplicationContext(), DonorHome.class);
startActivity(i);
}
        });
}
}
```

ViewFood.java:

```
package com.example.FeedTheNeed.view;
import android.Manifest;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import com.example.FeedTheNeed.R;
import com.example.FeedTheNeed.dao.DAO;
import com.example.FeedTheNeed.form.Food;
```

```
import com.example.FeedTheNeed.form.FoodRequest;
import com.example.FeedTheNeed.form.NGO;
import com.example.FeedTheNeed.util.Constants;
import com.example.FeedTheNeed.util.Session;
import com.google.android.gms.location.FusedLocationProviderClient;
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.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.StorageReference;

import java.util.List;
import java.util.Locale;

public class ViewFoodextends AppCompatActivity {

    Button menuDeleteFood;
Button viewFoodBack;
Button updatefood;
Button sendfoodrequest;
ImageViewimageView;

String foodAddress="";
String donormobile="";

    private FusedLocationProviderClientmFusedLocationClient;

    private double wayLatitude= 0.0, wayLongitude= 0.0;
    private LocationRequestlocationRequest;
    private LocationCallbacklocationCallback;
    private String txtLocation;
    private String lat;
    private String lang;

    private booleanisGPS= false;

TextViewt1,t2,t3,t4,t5,t6,t7;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_view_food);

mFusedLocationClient= LocationServices.getFusedLocationProviderClient(this);

locationRequest= LocationRequest.create();
locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
locationRequest.setInterval(10 * 1000); // 10 seconds
locationRequest.setFastestInterval(5 * 1000); // 5 seconds

new GpsUtils(this).turnGPSOn(new GpsUtils.onGpsListener() {
@Override
public void gpsStatus(booleanisGPSEnable) {
// turn on GPS
isGPS= isGPSEnable;
}
        });

locationCallback= new LocationCallback() {
@Override
public void onLocationResult(LocationResultlocationResult) {
if (locationResult == null) {
return;
}
for (Location location : locationResult.getLocations()) {
if (location != null) {
wayLatitude= location.getLatitude();
wayLongitude= location.getLongitude();
```

```
txtLocation= wayLatitude+ "," + wayLongitude;

                        if (mFusedLocationClient!= null) {
mFusedLocationClient.removeLocationUpdates(locationCallback);
}
                    }
                }
            }
        };

menuDeleteFood=(Button) findViewById(R.id.menuDeleteFood);
viewFoodBack=(Button) findViewById(R.id.viewFoodBack);
updatefood=(Button) findViewById(R.id.updateFood);
sendfoodrequest=(Button) findViewById(R.id.sendfoodrequest);

t1=(TextView) findViewById(R.id.foodviewname);
t2=(TextView)findViewById(R.id.foodviewpreparedtime);
t3=(TextView)findViewById(R.id.foodviewmembercount);
t4=(TextView)findViewById(R.id.foodviewstatus);
t5=(TextView)findViewById(R.id.foodviewpostedby);
t6=(TextView)findViewById(R.id.foodviewlocation);
t7=(TextView)findViewById(R.id.foodviewmobile);

imageView= (ImageView) findViewById(R.id.foodviewimage);

        final Session s = new Session(getApplicationContext());

Intent i = getIntent();
savedInstanceState = i.getExtras();
        final String foodid = savedInstanceState.getString("foodid");


        if(s.getRole().equals("DONOR"))
        {
sendfoodrequest.setEnabled(false);
}

        DAO d=new DAO();
d.getDBReference(Constants.FOOD_DB).child(foodid).addListenerForSingleValueEvent(new
ValueEventListener() {

@Override
public void onDataChange(DataSnapshotdataSnapshot) {

                Food food=dataSnapshot.getValue(Food.class);

                if(!food.getPostedby().equals(s.getusename()))
                {
menuDeleteFood.setEnabled(false);
updatefood.setEnabled(false);
}

if(food!=null)
                {
                    String[] foodLocation=food.getLocation().split(",");

Geocoder geocoder;
List<Address> addresses;
geocoder = new Geocoder(getApplicationContext(), Locale.getDefault());

                    try {

                        addresses = geocoder.getFromLocation(new Double(foodLocation[0]),new
Double(foodLocation[1]), 1); // Here 1 represent max location result to returned, by documents it
recommended 1 to 5

String address = addresses.get(0).getAddressLine(0); // If any additional address line present
than only, check with max available address lines by getMaxAddressLineIndex()
String city = addresses.get(0).getLocality();
String state = addresses.get(0).getAdminArea();
String country = addresses.get(0).getCountryName();
String postalCode = addresses.get(0).getPostalCode();
String knownName = addresses.get(0).getFeatureName(); // Only if available else return NULL
```

```
if(address!=null)
                        {
foodAddress=foodAddress+address+"\n";
}

if(city!=null)
                        {
foodAddress=foodAddress+city+"\n";
}

if(state!=null)
                        {
foodAddress=foodAddress+state+"\n";
}

if(country!=null)
                        {
foodAddress=foodAddress+country+"\n";
}

if(postalCode!=null)
                        {
foodAddress=foodAddress+postalCode+"\n";
}

if(knownName!=null)
                        {
foodAddress=foodAddress+knownName+"\n";
}
                    }
catch(Exception e)
                    {
Log.v("onemeal ","in on success ");
}

Log.v("onemeal Address ",foodAddress);

t1.setText("Name: "+food.getName());
t2.setText("Prepared Time: "+food.getPreparedtime());
t3.setText("Description: "+food.getMembercount());
t4.setText("Status: "+food.getStatus());
t5.setText("Posted By: "+food.getPostedby());
t6.setText("Location: "+foodAddress);
t7.setText("Mobile: "+food.getMobile());

donormobile=food.getMobile();
foodAddress=foodAddress;

StorageReference ref = DAO.getStorageReference().child("images/" + food.getImage());
                    long ONE_MEGABYTE = 1024 * 1024 *5;
ref.getBytes(ONE_MEGABYTE)
                            .addOnSuccessListener(new OnSuccessListener<byte[]>() {
@Override
public void onSuccess(byte[] bytes) {

                                    Bitmap bm = BitmapFactory.decodeByteArray(bytes, 0,
bytes.length);

                                    if(bm!=null)
                                    {
imageView.setImageBitmap(bm);
}
else
{
Log.v("onemeal ","bm null");
}
                                }
                            }).addOnFailureListener(new OnFailureListener() {
@Override
public void onFailure(@NonNull Exception exception) {
// Handle any errors
Log.v("onemeal ","image reading failure");
}
                    });
```

```
}
            }

@Override
public void onCancelled(DatabaseErrordatabaseError) {

            }
        });

menuDeleteFood.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {

final DAO dao=new DAO();
dao.deleteObject(Constants.FOOD_DB,foodid);

Intent i= new Intent(getApplicationContext(), DonorHome.class);
startActivity(i);
}
        });

updatefood.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
                Intent intent=new Intent(getApplicationContext(),UpdateFood.class);
intent.putExtra("foodid",foodid);
startActivity(intent);
}
        });

viewFoodBack.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {

                Intent i=null;

String role=s.getRole();

                if(role.equals("DONOR"))
                {
i= new Intent(getApplicationContext(), DonorHome.class);
}else if(role.equals("NGO"))
                {
i= new Intent(getApplicationContext(), NGOHome.class);
}
startActivity(i);
}
        });

sendfoodrequest.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {

                DAO d = new DAO();
d.getDBReference(Constants.USER_DB).child(s.getusename()).addListenerForSingleValueEvent(new
ValueEventListener() {

@Override
public void onDataChange(DataSnapshotdataSnapshot) {

                        NGO user = (NGO) dataSnapshot.getValue(NGO.class);

                        if (user == null) {
Toast.makeText(getApplicationContext(), "Invalid UserName ", Toast.LENGTH_SHORT).show();
} else if (user != null) {

if (!isGPS) {
Toast.makeText(getApplicationContext(), "Please turn on GPS", Toast.LENGTH_SHORT).show();
                                return;
}

getLocation();

                            final Session s = new Session(getApplicationContext());
```

```
                            if (txtLocation!= null &&lat!=null &lang!=null) {

Toast.makeText(getApplicationContext(), "LatLangs:"+lat+"\t"+lang, Toast.LENGTH_SHORT).show();

String[] requestLocation=txtLocation.split(",");

Geocoder geocoder;
List<Address> addresses;
geocoder = new Geocoder(getApplicationContext(), Locale.getDefault());

String requestAddress="";

                                try {

                                    addresses = geocoder.getFromLocation(new Double(lat),new
Double(lang), 1); // Here 1 represent max location result to returned, by documents it recommended
1 to 5

String address = addresses.get(0).getAddressLine(0); // If any additional address line present
than only, check with max available address lines by getMaxAddressLineIndex()
String city = addresses.get(0).getLocality();
String state = addresses.get(0).getAdminArea();
String country = addresses.get(0).getCountryName();
String postalCode = addresses.get(0).getPostalCode();
String knownName = addresses.get(0).getFeatureName(); // Only if available else return NULL

if(address!=null)
                                    {
requestAddress=requestAddress+address+"\n";
}

if(city!=null)
                                    {
requestAddress=requestAddress+city+"\n";
}

if(state!=null)
                                    {
requestAddress=requestAddress+state+"\n";
}

if(country!=null)
                                    {
requestAddress=requestAddress+country+"\n";
}

if(postalCode!=null)
                                    {
requestAddress=requestAddress+postalCode+"\n";
}

if(knownName!=null)
                                    {
requestAddress=requestAddress+knownName+"\n";
}
                                }
catch(Exception e)
                                {
Log.v("onemeal ","in on success ");
}

                                DAO dao = new DAO();

FoodRequestfoodRequest = new FoodRequest();
foodRequest.setFoodrequestid(dao.getUnicKey(Constants.FOOD_REQUESTS_DB));
foodRequest.setSourcelocation(foodAddress);
foodRequest.setDestinationlocation(requestAddress);
foodRequest.setRequestdby(user.getMobile());
foodRequest.setDonatedby(donormobile);

                                try {

dao.addObject(Constants.FOOD_REQUESTS_DB, foodRequest, foodRequest.getFoodrequestid());
```

```
Intent i = new Intent(getApplicationContext(), NGOHome.class);
startActivity(i);
Toast.makeText(getApplicationContext(), "Food Request Sent Successfully",
Toast.LENGTH_SHORT).show();
} catch (Exception ex) {
Toast.makeText(getApplicationContext(), "Food Request Failed", Toast.LENGTH_SHORT).show();
ex.printStackTrace();
}
                            }
else
{
Toast.makeText(getApplicationContext(), "Unable to get Location", Toast.LENGTH_SHORT).show();
}

                        } else {
Toast.makeText(getApplicationContext(), "Invalid User", Toast.LENGTH_SHORT).show();
}
                    }

@Override
public void onCancelled(DatabaseErrordatabaseError) {

                    }
                });
}
        });
}

private void getLocation() {
if (ActivityCompat.checkSelfPermission(ViewFood.this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED
&&ActivityCompat.checkSelfPermission(ViewFood.this, Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
ActivityCompat.requestPermissions(ViewFood.this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION, Manifest.permission.ACCESS_COARSE_LOCATION},
AppConstants.LOCATION_REQUEST);
} else {

mFusedLocationClient.getLastLocation().addOnSuccessListener(ViewFood.this, new
OnSuccessListener<Location>() {
@Override
public void onSuccess(Location location) {

if (location != null) {
wayLatitude= location.getLatitude();
wayLongitude= location.getLongitude();
lat=wayLatitude+"";
lang=wayLongitude+"";
txtLocation= wayLatitude+ "," + wayLongitude;
} else {
if (ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS FINE LOCATION) !=
PackageManager.PERMISSION_GRANTED&&ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
// TODO: Consider calling
//    ActivityCompat#requestPermissions
// here to request the missing permissions, and then overriding
                            //   public void onRequestPermissionsResult(int requestCode, String[]
permissions,
                            //                                          int[] grantResults)
                            // to handle the case where the user grants the permission. See the
documentation
                            // for ActivityCompat#requestPermissions for more details.
return;
}
mFusedLocationClient.requestLocationUpdates(locationRequest, locationCallback, null);
}
                }
            });
}
    }

@SuppressLint("MissingPermission")
```

```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull
int[] grantResults) {
super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        switch (requestCode) {
case 1000: {
// If request is cancelled, the result arrays are empty.
if (grantResults.length>0
&&grantResults[0] == PackageManager.PERMISSION_GRANTED) {

mFusedLocationClient.getLastLocation().addOnSuccessListener(ViewFood.this, new
OnSuccessListener<Location>() {
@Override
public void onSuccess(Location location) {
if (location != null) {
wayLatitude= location.getLatitude();
wayLongitude= location.getLongitude();
txtLocation=wayLatitude+","+wayLongitude;
} else {
mFusedLocationClient.requestLocationUpdates(locationRequest, locationCallback, null);
}
                        }
                    });

} else {
Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show();
}
break;
}
        }
    }
}
```

ViewFoodRequests.java:

```
package com.example.FeedTheNeed.view;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.FeedTheNeed.R;
import com.example.FeedTheNeed.dao.DAO;
import com.example.FeedTheNeed.form.FoodRequest;
import com.example.FeedTheNeed.util.Constants;
import com.example.FeedTheNeed.util.Session;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.ValueEventListener;

public class ViewFoodRequestextends AppCompatActivity {

    Button viewFoodRequestBack;
TextViewt1,t2,t3,t4;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_view_food_request);

viewFoodRequestBack=(Button) findViewById(R.id.viewFoodrequestBack);

t1=(TextView) findViewById(R.id.foodrequestsource);
t2=(TextView)findViewById(R.id.foodrequestdestination);
t3=(TextView)findViewById(R.id.foodrequestsourcerequestedby);
t4=(TextView)findViewById(R.id.foodrequestsourcedonatedby);

        final Session s = new Session(getApplicationContext());

Intent i = getIntent();
savedInstanceState = i.getExtras();
        final String foodrequestid = savedInstanceState.getString("foodrequestid");
```

```
DAO d=new DAO();
d.getDBReference(Constants.FOOD_REQUESTS_DB).child(foodrequestid).addListenerForSingleValueEvent(n
ew ValueEventListener() {

@Override
public void onDataChange(DataSnapshotdataSnapshot) {

FoodRequestfoodRequest=dataSnapshot.getValue(FoodRequest.class);

                if(foodRequest!=null)
                {
t1.setText("Source Location: "+foodRequest.getSourcelocation());
t2.setText("Destination Location: "+foodRequest.getDestinationlocation());
t3.setText("Source Mobile: "+foodRequest.getRequestdby());
t4.setText("Destination Mobile: "+foodRequest.getDonatedby());
}
            }

@Override
public void onCancelled(DatabaseErrordatabaseError) {

            }
        });


}
}
```

Food.java:

```
package com.example.FeedTheNeed.form;

public class Food {

private String id;
    private String name;
    private String image;
    private String preparedtime;
    private String geolocation;
    private String membercount;
    private String status;
    private String postedby;
    private String location;
    private String mobile;

    public String getMobile() {
return mobile;
}

public void setMobile(String mobile) {
this.mobile= mobile;
}

public String getLocation() {
return location;
}
public void setLocation(String location) {
this.location= location;
}
public String getMembercount() {
return membercount;
}
public void setMembercount(String membercount) {
this.membercount= membercount;
}
public String getPostedby() {
return postedby;
}

public void setPostedby(String postedby) {
this.postedby= postedby;
}
public String getId() {
return id;
}
```

```
public void setId(String id) {
this.id = id;
}

public String getName() { return name;  }

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

public String getImage() {
return image;
}

public void setImage(String image) {
this.image= image;
}

public String getPreparedtime() {
return preparedtime;
}

public void setPreparedtime(String preparedtime) {
this.preparedtime= preparedtime;
}

public String getGeolocation() {
return geolocation;
}

public void setGeolocation(String geolocation) {
this.geolocation= geolocation;
}

public String getStatus() {
return status;
}

public void setStatus(String status) {
this.status= status;
}

}
```

FoodRequest.java:

```
package com.example.FeedTheNeed.form;

public class FoodRequest {

private String foodrequestid;
    private String sourcelocation;
    private String destinationlocation;
    private String requestdby;
    private String donatedby;

    public String getFoodrequestid() {
return foodrequestid;
}

public void setFoodrequestid(String foodrequestid) {
this.foodrequestid= foodrequestid;
}

public String getSourcelocation() {
return sourcelocation;
}

public void setSourcelocation(String sourcelocation) {
this.sourcelocation= sourcelocation;
}

public String getDestinationlocation() {
return destinationlocation;
```

```
}

public void setDestinationlocation(String destinationlocation) {
this.destinationlocation= destinationlocation;
}

public String getRequestdby() {
return requestdby;
}

public void setRequestdby(String requestdby) {
this.requestdby= requestdby;
}

public String getDonatedby() {
return donatedby;
}

public void setDonatedby(String donatedby) {
this.donatedby= donatedby;
}
}
```

NGO.java:

```
package com.example.FeedTheNeed.form;

public class FoodRequest {

private String foodrequestid;
    private String sourcelocation;
    private String destinationlocation;
    private String requestdby;
    private String donatedby;

    public String getFoodrequestid() {
return foodrequestid;
}

public void setFoodrequestid(String foodrequestid) {
this.foodrequestid= foodrequestid;
}

public String getSourcelocation() {
return sourcelocation;
}

public void setSourcelocation(String sourcelocation) {
this.sourcelocation= sourcelocation;
}

public String getDestinationlocation() {
return destinationlocation;
}

public void setDestinationlocation(String destinationlocation) {
this.destinationlocation= destinationlocation;
}

public String getRequestdby() {
return requestdby;
}

public void setRequestdby(String requestdby) {
this.requestdby= requestdby;
}

public String getDonatedby() {
return donatedby;
}

public void setDonatedby(String donatedby) {
this.donatedby= donatedby;
```

```
}
}
```

DAO.java:

```
package com.example.FeedTheNeed.form;

public class FoodRequest {

private String foodrequestid;
    private String sourcelocation;
    private String destinationlocation;
    private String requestdby;
    private String donatedby;

    public String getFoodrequestid() {
return foodrequestid;
}

public void setFoodrequestid(String foodrequestid) {
this.foodrequestid= foodrequestid;
}

public String getSourcelocation() {
return sourcelocation;
}

public void setSourcelocation(String sourcelocation) {
this.sourcelocation= sourcelocation;
}

public String getDestinationlocation() {
return destinationlocation;
}

public void setDestinationlocation(String destinationlocation) {
this.destinationlocation= destinationlocation;
}

public String getRequestdby() {
return requestdby;
}

public void setRequestdby(String requestdby) {
this.requestdby= requestdby;
}

public String getDonatedby() {
return donatedby;
}

public void setDonatedby(String donatedby) {
this.donatedby= donatedby;
}
}
package com.example.FeedTheNeed.form;

public class FoodRequest {

private String foodrequestid;
    private String sourcelocation;
    private String destinationlocation;
    private String requestdby;
    private String donatedby;

    public String getFoodrequestid() {
return foodrequestid;
}

public void setFoodrequestid(String foodrequestid) {
this.foodrequestid= foodrequestid;
}

public String getSourcelocation() {
return sourcelocation;
```

```
}

public void setSourcelocation(String sourcelocation) {
this.sourcelocation= sourcelocation;
}

public String getDestinationlocation() {
return destinationlocation;
}

public void setDestinationlocation(String destinationlocation) {
this.destinationlocation= destinationlocation;
}

public String getRequestdby() {
return requestdby;
}

public void setRequestdby(String requestdby) {
this.requestdby= requestdby;
}

public String getDonatedby() {
return donatedby;
}

public void setDonatedby(String donatedby) {
this.donatedby= donatedby;
}
}
```

GetFireBaseConnection.java:

```
package com.example.FeedTheNeed.form;

public class FoodRequest {

private String foodrequestid;
    private String sourcelocation;
    private String destinationlocation;
    private String requestdby;
    private String donatedby;

    public String getFoodrequestid() {
return foodrequestid;
}

public void setFoodrequestid(String foodrequestid) {
this.foodrequestid= foodrequestid;
}

public String getSourcelocation() {
return sourcelocation;
}

public void setSourcelocation(String sourcelocation) {
this.sourcelocation= sourcelocation;
}

public String getDestinationlocation() {
return destinationlocation;
}

public void setDestinationlocation(String destinationlocation) {
this.destinationlocation= destinationlocation;
}

public String getRequestdby() {
return requestdby;
}

public void setRequestdby(String requestdby) {
this.requestdby= requestdby;
}
```

```
public String getDonatedby() {
return donatedby;
}

public void setDonatedby(String donatedby) {
this.donatedby= donatedby;
}
}
```

REFERENCES

B. Kabilan, C. Karthickraja, A. Karthik (2021) Food Conservation Application - Mobile App Connecting Provider and Consumer

P. Vijay Bhaskar Reddy, V. Revanth Kumar (2020) Food Wastage Management using Android Application

M.S.Elavarasan, MR.C.Daniel Nesakumar (2019) Food Wastage Reduction Mobile Application

Suraya Masrom, Abdullah Sani Abd. Rahman, Farah Norliyana Azahar, Nasiroh Omar (2018) Food for You (F4U) Mobile Charity Application.