FOOD SHARER MOBILE APPLICATION

IT7611-CREATIVE AND INNOVATIVE PROJECT

A PROJECT REPORT

Submitted by

LOURDUMARGARATE A(2017506553)

PARTHIBAN S(2017506571)

KALAIYARASI K(2017506537)

NITHYA S(2017506568)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY, CHENNAI-600 044.

APRIL-2020

FOOD SHARER MOBILE APPLICATION

IT7611-CREATIVE AND INNOVATIVE PROJECT

A PROJECT REPORT

Submitted by

LOURDUMARGARATE A(2017506553)

PARTHIBAN S(2017506571)

KALAIYARASI K(2017506537)

NITHYA S(2017506568)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY, CHENNAI-600 044.

APRIL-2020

DEPARTMENT OF INFORMATION TECHNOLOGY

ANNA UNIVERSITY, CHENNAI 600044

BONAFIDE CERTIFICATE

Certified that this project Report titled "**Food Sharer Mobile Application"** is the bonafide work of **Lourdumargarate A**(RegNo.:**2017506553), Parthiban S(**RegNo.:**2017506571),Kalaiyarasi K(**RegNo.:**2017506537),Nithya S(**RegNo.:**2017506568)** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part or full of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this to any other candidate.

SIGNATURE	SIGNATURE
DR.DHANANJAY KUMAR,	DR.M.R.SUMALATHA
HEAD OF THE DEAPARTMENT,	SUPERVISOR,
DEPT OF INFORMATION TECHNOLOGY,	DEPT OF INFORMATION TECHNOLOGY,
MADRAS INSTITUTE OF TECHNOLOGY,	MADRAS INSTITUTE OF TECHNOLOGY,
ANNA UNIVERSITY,	ANNA UNIVERSITY,
CHENNAI-600 044.	CHENNAI-600 044.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of our task would be incomplete without mentioning the names of people who made it possible, whose constant guidance and encouragement crowns all efforts with success. We express our gratitude and sincere thanks to our respected Dean, DR.T.THIYAGARAJAN, Anna University, MIT Campus for providing the excellent computing facilities to do the project. Our heartfelt thanks to DR.DHANANJAY KUMAR, Head, Department of Information Technology, Madras Institute of Technology Campus, Anna University, for the prompt and limitless help in providing the excellent computing facilities to do the project.

We express our profound sense of gratitude to our project panel Member DR.M.R.SUMALATHA, Department of Information Technology for their invaluable support, guidance and encouragement for the successful completion of this project.

ABSTRACT

Food is one of the most vital and basic thing for a human survival. We are living in a world where there are lot of inequalities.For example there are families who has so much money that they can feed many of their generations on the other hand there are families who are not even able to make the ends meet for their basic survival. This problem is not specific to developing countries but developed countries are too affected by this problem. From the land to the table, foods gets wasted in many parts of its processing cycle. Hunger deaths are the worst thing that could happen to humanity. This types of deaths occur not due to the failure of the man but due to the failure of the entire society in which we live. We are living in a society where everyone is busy with their own work but when it comes to hunger no one can compromise the fact that food is the basic necessity for life. So we are much obliged to provide a solution for this societal evil of food waste by developing an mobile application using which any people who is having excess of food can share it to the needy at free of cost.This mobile app will come with a Google map facility where people can connect with one another for sharing food. This mobile app will connect the volunteers with the distributor who will process the excess food and distribute it, so the needy is benefited. The people will be able to post the excess food details in our app and it will be displayed to all users and the distributors will come forward to distribute it to the beneficiaries. We are happy to devote our work to one of the most important and essential cause which was not given enough attention before.This project act as a bridge between the food donors and acceptors who need the help from our society.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Food waste is huge crisis arising in today's world. Recently, the issue of surplus food management has attracted much attention from academics and practitioners, since it is believed to have a huge possibility to reduce food insecurity, i.e. the condition when the food intake of some household members is reduced and normal eating patterns are disrupted at times due to limited resources .Food insecurity and food losses together generate a paradoxical reality; a total of 842 million people worldwide were estimated to be suffering from regularly not getting enough food, while approximately 1.3 billion tons per year food is wasted globally. Food insecurity is a relevant issue even in high-income countries. Today, in Europe, nearly 43.6 million people are estimated to be food insecure. At the same time, European countries are reported to generate 179 kg per capita of food waste every year. In other words, 89 million tons per year food is wasted in Europe, approximately 58% of what is produced by firms in manufacturing, wholesale, retail and food service stages of the food supply chain. Besides the economic loss for the firms producing food waste, social and environmental losses are also clear especially when the food is edible but for various reasons is not sold to or consumed to the intended customer and eventually becomes waste.

1.2. EXISTING APPROACHES

No Food Waste is a mission to end food waste and hunger to make the "World Hunger Free". We recover surplus food from weddings, parties and functions and donate it to needy and hungry people.This app process the surplus food app using mobile calls which is a tedious process. We process the surplus food using GUI.

This main apporoaches:

1. Donator make a entry in the app and post their wasted/extra food. Once donar make a entry,doesn't needed again.
2. Acceptor make a call to donator and get the food based on distance.and already make a entry in the app.

The app main goal is "DON'T WASTE FOOD; GIVE TO OTHERS".

1.3 PROBLEM STATEMENT

In literature, a generally accepted definition of food waste and surplus food does not exist, since it is subject to different interpretations within different perspectives. In fact, going back in time, one of the oldest papers found about food waste published in 1979 brings about the debate on the issue of what the food waste is and what food waste is not. Most of the studies try to explain the food waste by dividing it into different categories or directly quantifying the value of waste along the food supply chain. A recently developed conceptual model called "ASRW" gives a clear definition for "Food share app" and distinguishes it from the food waste by introducing the concepts of Availability, Surplus, Recoverability and Waste. According to the model, Food Availability is defined as "all food produced throughout the food supply chain" and it consists of three components: food availability for "human consumption", "surplus food" and "food scraps". Food availability for human consumption includes "the edible food that is delivered through traditional market channels and consumed by people to satisfy their food needs". Surplus food is defined as "the edible food that is produced, manufactured, retailed or served but for various reasons is not sold to or consumed by the intended customer" and food scraps are "inedible food that is no longer suitable for human consumption"

CHAPTER 2

LITERATURE SURVEY

Narvanen, Mesiranta, Mattila and Heikkinen present a much-needed framework for managing food waste, including food surplus, food loss and food waste. The framework discusses the characteristics of food waste as unstructured, cross-cutting and relentless, that is, as a wicked problem. The chapter provides a concise review of recent food waste studies, particularly from the perspective of finding solutions. From agriculture and fishing stage to final household consumption food is wasted throughout the whole supply chain, together with water, cropland and fertilizers used to produce losses. Therefore, reducing the food losses along the food supply chain is a critical and high priority component of any sustainability strategy with a high potential to improve.

2.1 WICKED PROBLEM OF FOOD WASTE:

There is an increasing political and scientific consensus about the need to reduce global food waste. In 2015, the United Nation's Sustainable Development goal 12.3 set the target of "By 2030,halve per capita global food waste at the retail and consumer levels and reduce food losses along production and supply chains, including post-harvest losses"(United Nations 2015).This target stems from a broad understanding of food losses and waste, including the waste of land, water and energy, while causing unnecessary emissions of greenhouse gases. The Food and Agriculture Organization of the United Nations (FAO) has estimated that food losses and waste account for more than 10% of global energy consumption(FAO 2017). Hence food waste is a major contributor to climate change. Furthermore wasting food is a moral concern, since it impacts global food security and increases the gap between affluent and income people. Food produced for human consumption is wasted at the same time as a large part of the global population suffers from hunger and malnutrition. From an economic point of view, inefficiencies accrue from food losses and waste for both organizations such as farms, food manufacturers, retailers and restaurants and households.

Food waste can be characterized as a "wicked problem" (Narvanen et al. 2019), which are defined as unstructured, cross-cutting and relentless(Weber and Khademian 2008; also Rittel) Firstly, food waste is an unstructured problem, because its precise causes and effects are difficulty to identify, and there is no shared problem definition.

Many of the suggested solutions for reducing food waste in the literature have focussed on changing the attitudes and behaviour of individuals (see van Geffen et al., Chapter 2), for example through awareness-raising informational campaigns(for a review, see Aschemann-Witzel et al. 2017; Queued et al. 2013; see Sutinen, Chapter 9). However in developing solutions for food waste reduction, it must be acknowledged that even though food distributors and households may produce the greatest amount of (quantifiable) food waste, they should not be held exclusively responsible for its emergence (Evans 2011). Instead food waste occurs at the intersection of several influences across the food system. These include the myriad ways in which food is, for instance, produced, transported, processed, packaged and stored on the supermarket shelves and at home. For instance, at the consumer level, many routines and contexts influence the emergence of food waste, not only those directly related to the disposal of food (Evans 2014).

2.2 FOOD RECOVERY HIERARCHY PROCESS:

According to Food Recovery Hierarchy Process, the strategies suggested are source reduction, feeding human, feeding animals, industrial usage, composting and disposal respectively. Besides feeding the human and feeding the animal, another way is the reusing, since the food waste residues containing a number of valuable compounds derivable have a great potential to be reused into other production systems. Otherwise, food waste can be used for fuel conversation and for digestion to recover energy. On the other hand, many of the food processing companies use composting as a means of waste management. Composting is the natural aerobic biochemical process in which microorganisms transform organic materials such as waste from vegetables, fruits, fish and meat into stable soil-like product. In addition to economic value of the product, the process is generally perceived as being environmentally friendly. Obviously, sending the valuable materials to landfilling, incineration and disposal is not the ideal situation for the society and economy, and consequently appears at the bottom of the Food Recovery Hierarchy Process (EPA 2013).

2.3 SURPLUS FOOD REDISTRIBUTION TO DISFAVORED POPULATION:

Although there are some debates on the issues, food assistance through surplus food is considered as a mean to fight against food poverty by many researchers and practitioners (Tarasuk & Eakin ,2003; Aleksandar & Smaje, 2008; Foresight, 2011; Hawkes & Webster, 2014). First of all, from the sustainability perspective, surplus food redistribution has the most socially prioritized strategy since nothing is wasted and food is used to tackle food security, i.e the highest priority (Schneider, 2013). Moreover, in most countries, tex deductions are granted based on the social return from the donated food, helping to save cost for companies (Aiello, Enea, & Muriana, 2013). Furthermore and specifically, it has been demonstrated that food donation programs save large companies more than $100000 in annual shipping and landfilling cost(Thang, 2009). Food redistribution strategy is analysed also from the Corporate Social Reasonability (CSR) perspective, which is very crucial leverage for global companies receiving great attention from their stakeholders. Dimensions of CSR in the food supply chain can be classified as animal welfare, biotechnology, health and safety, environment, labor and human rights, community , fair trade, and procurement (Maloni & Brown, 2006). Another interesting stream of the literature focuses on the issue of prosocial behaviour , which can be divided as the “voluntary behaviour intended to benefit another”, such as helping, sharing, donating, co-operating, and volunteering (Brief & Motowidlo, 1886).Studies show that there is a positive correlation between prosocial activities and employee satisfaction and team performance (Anik, et al., 2009; Aknin et al., 2011). This appears to be a promising avenue to motivate surplus food redistribution, but there is no study directly addressing the relationship between the prosocial behaviour of the donor company employees and the efficiency and effectiveness of surplus food redistribution to needy. Although surplus food redistribution by the food supply chain players has many advantages for companies like improving corporate image, eliminating disposal costs and tipping fees, there are some legal and "red tape" requirements that the donator has to meet, which can be considered as a barrier for companies. The attempts by the governments like the Bill Emerson Good Samaritan Food Donation are seen as good practices to facilitate the donation of food and grocery products to social welfare organizations (Schneider, 2013).

2.4 FOOD WASTE IN RECENT YEARS:

In recent years food waste has received growing interest from local, national and European policymakers, international organizations, NGOs as well as academics from various disciplinary fields. Increasing concerns about food security and environmental impacts, such as resource depletion and greenhouse gas emissions attributed to food waste, have intensified as key actors in food waste generation. However, the evidence on why food waste occurs remains scattered. This maps the still small but expanding academic territory of consumer food waste by systematically reviewing empirical studies on food waste practices as well as distilling factors that foster and impede the generation of food waste on the household level. The analysis reveals food waste as a complex and multi-faceted issue that cannot be attributed to single variables; this also calls for a stronger integration of different disciplinary perspectives. Mapping the determinants of waste generation deepens the understanding of household practices and helps design food waste prevention strategies.

CHAPTER 3

PROPOSED WORK

3.1 MODULES

Module 1:

HOME PAGE PHASE:

At first we started by developing   a home page for User Registration. In this page we have picture for our app and one simple button to enter into an app. While entering into the app we will have two buttons which is login and register. If a user is new to this app, first they have to register in this app by giving their details and after successful registration only they can enter into the app. If a user is already a user to this app, then they can directly enter into the app by just logging into this app. This two things are the basic things to work in this app and these two buttons will lead us to all other pages in this app. In this both login and register page, we will option for us to choose either we are food provider or we are food needer.

Module 2:

FORMS PHASE:

In this phase we are concentrating in creating the various forms in different ways to work into this app and also help us to move into other various pages. At first we will create the form for new user, from where we get various information about the user such as their Email, Password, Mobile no and so on. Next we will create the form for already user from where they can log in by providing their registered mail id and their password. Simultaneously while we are getting the information from the users we also store that information in databases by providing the database connectivity to this app. Whatever the actions performing in this app we are maintaining the database and the information can be viewed at any time.

Module 3:

FOOD PROVIDER PHASE:

Nowadays, the No.of.deaths due to hungry is the most. In order to avoid that we are developing this app to help whatever we can do for them. In this app the food providers are those who have excess food in their houses, parties, hotels and so on.

The food providers can register at any time whenever they can provide the food. Once registered into this app they need not to register again and also they just need to login alone. The food providers also have to fill the form like how much food they have, when they prepared that food, how much distance that food is available and so on. They will post all this information by using their user id.

Module 4:

FOOD NEEDER PHASE:

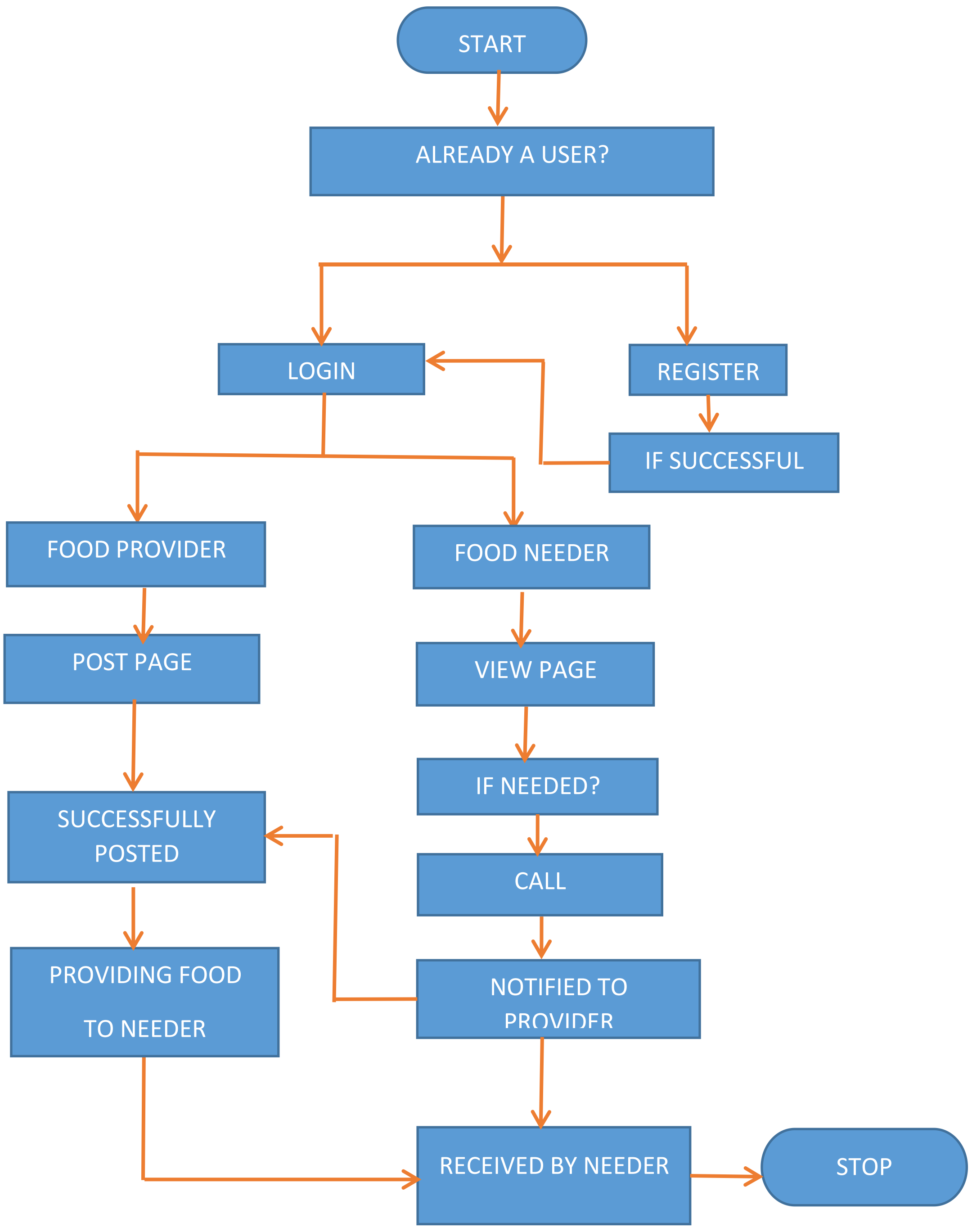
In this phase, we will get the information about the food needer by providing the form. The food needer persons are those really need the food which will fade away their hungry. After log in into this app the food needers can view the post page which is posted by the food provider. By viewing the information the food needers can directly contact the food provider by making the calls. If the needer is very near from the provider they can easily get the food by walk distance otherwise the food needer is far away from the provider, the food provider will trace the food needer where they are by using the Google maps to locate their location easily and the food providers will provide the food to the needer.

Module 5:

DATABASE CONNECTIVITY PHASE:

In this phase we will maintain the database for all the food providers and food needers. Only the app admins can see the database details. For every food providers we will give the unique id for their easy use. In this course of time we concentrated on the posting the form and the food provider details table that will be displayed to the food needers. The food providers will be provide with a unique OTP which they will share with the food needers and the food needers uses the OTP to remove the post from the page so singularity is ensured.

3.2ARCHITECTURE



CHAPTER 4

IMPLEMENTATION AND ANALYSIS

4.1 PROJECT DESCRIPTION:

OVERALL AIM:

To develop a useful interface that helps in communication between the donors and the acceptors of food.

SUB-TOPICS:

HOME PAGE:

A page that allows the user to navigate through different parts of the project.

SING IN/SIGN UP PAGE:

A page that helps the users to choose between sign in and sign up.

POST YOUR SURPLUS PAGE:

This page allows the user to capture the picture of the food they are going to donate to others.They have to submit atleast two pictures of the item,then only it allows to post the food item.

DONORS DETAILS PAGE:

A page that displays information like donor name, phone number of donors, address, veg or non-veg, quantity. These fields will be displayed only to acceptors who will simultaneously uses the phone number in the table to contact and collect food from donors.

MANAGEMENT PAGE:

A page that displays the youtube videos that decribe about how to prevent food wastage,how we can conserve food etc.

ACCEPTOR PAGE:

When the acceptor selects the food they want then this page allows the acceptor to make call to the donor.After making call they delete those item from the item list.

4.2 PERFORMANCES:

Our Graphical User Interface is designed in such a way that even a beginner can handle the system with ease. We have rigorously tested the system for various inputs and our system have responded to it with correct outputs and exception.We have divided the system layout into various parts so that the user would find it easy to navigate across various pages in the system. Since our system is a software oriented system it will not get deteriorated over time so we need not care about maintenance.

Our system enables the acceptor of the food to make call to get the food from the donor.This app allows the donor to capture the picture of the food to post and make available to the acceptor.

4.3 SCREEN SHOTS:

Home page:

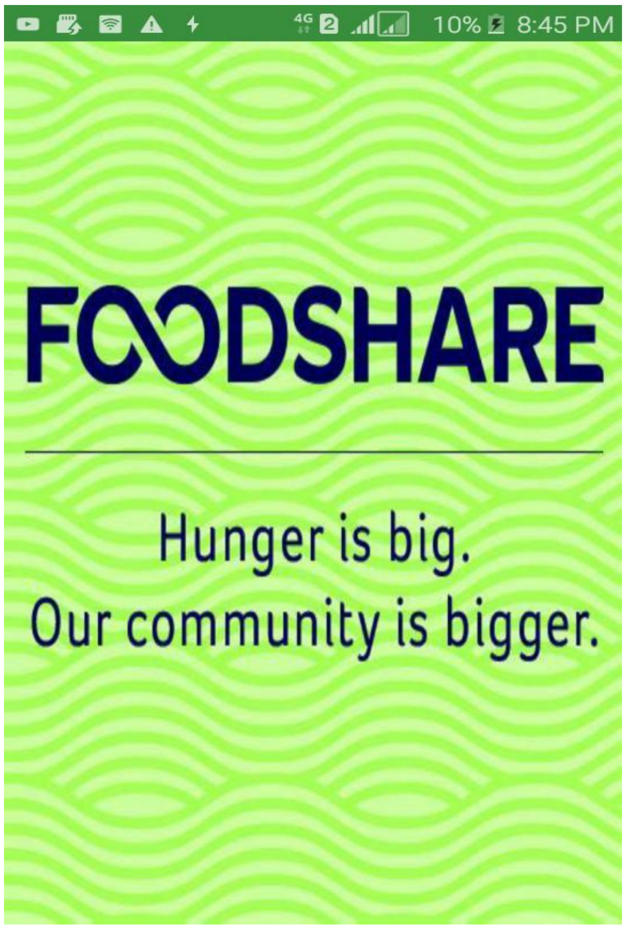
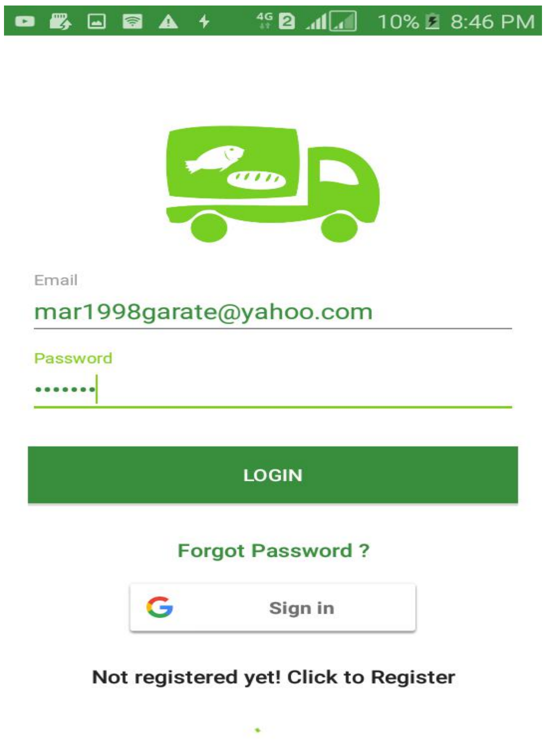


Fig 1.1 Homepage

LOGIN PAGE:



10% 8:46 PM

Email

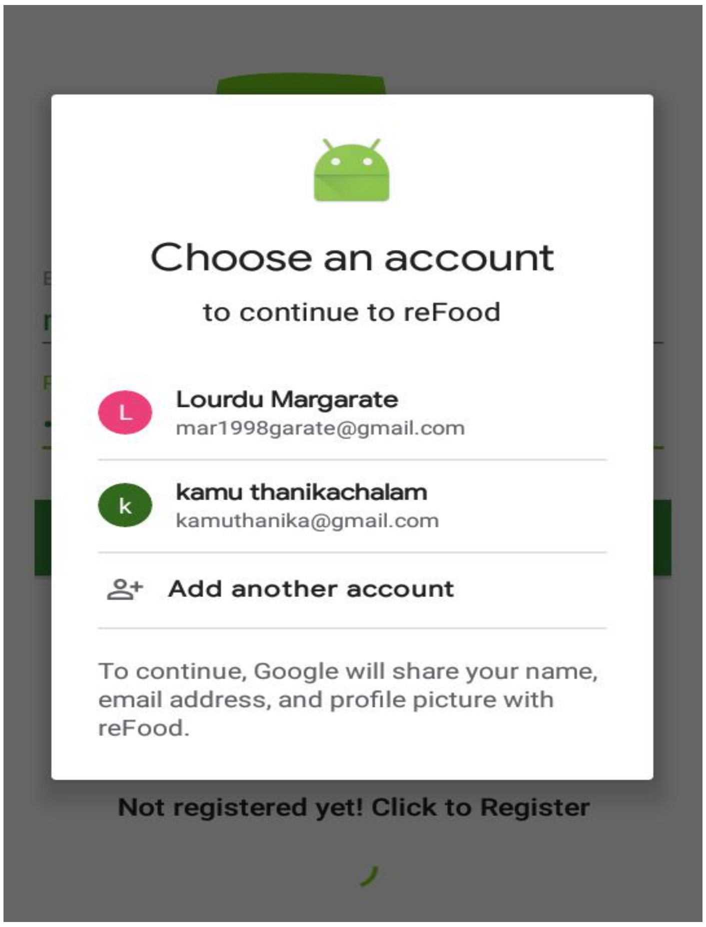
mar1998garate@yahoo.com

Password

•••••••

Fig 1.2 Login page

SIGNIN FROM GMAIL:


Choose an account

to continue to reFood

Lourdu Margarate
mar1998garate@gmail.com

kamu thanikachalam
kamuthanika@gmail.com

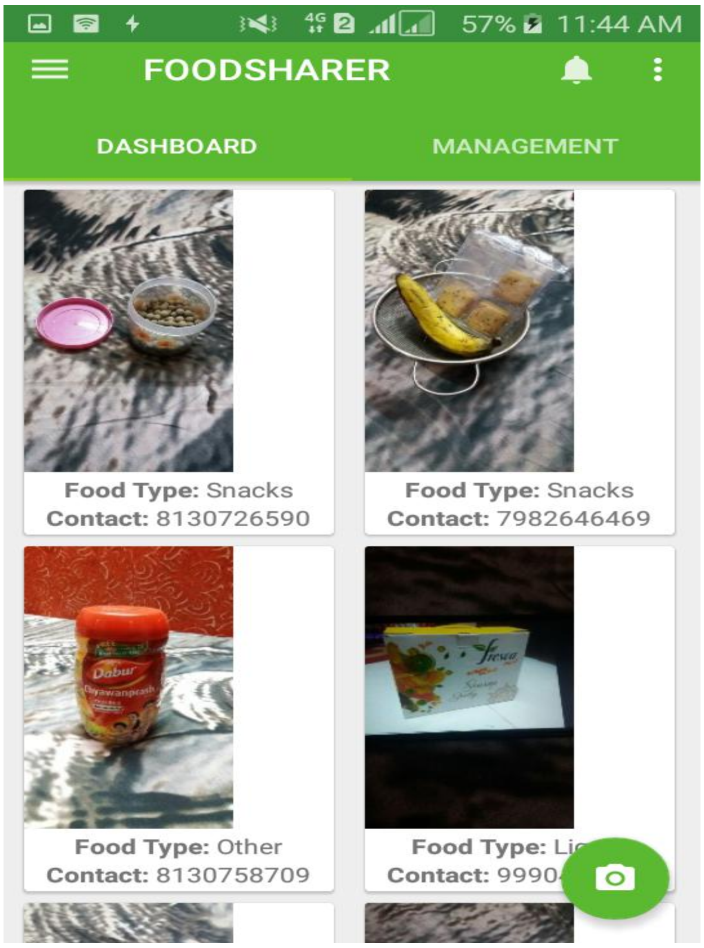
Add another account

To continue, Google will share your name, email address, and profile picture with reFood.

Not registered yet! Click to Register

Fig 1.3 Signin page

DASHBOARD:



Food Type: Snacks
Contact: 8130726590

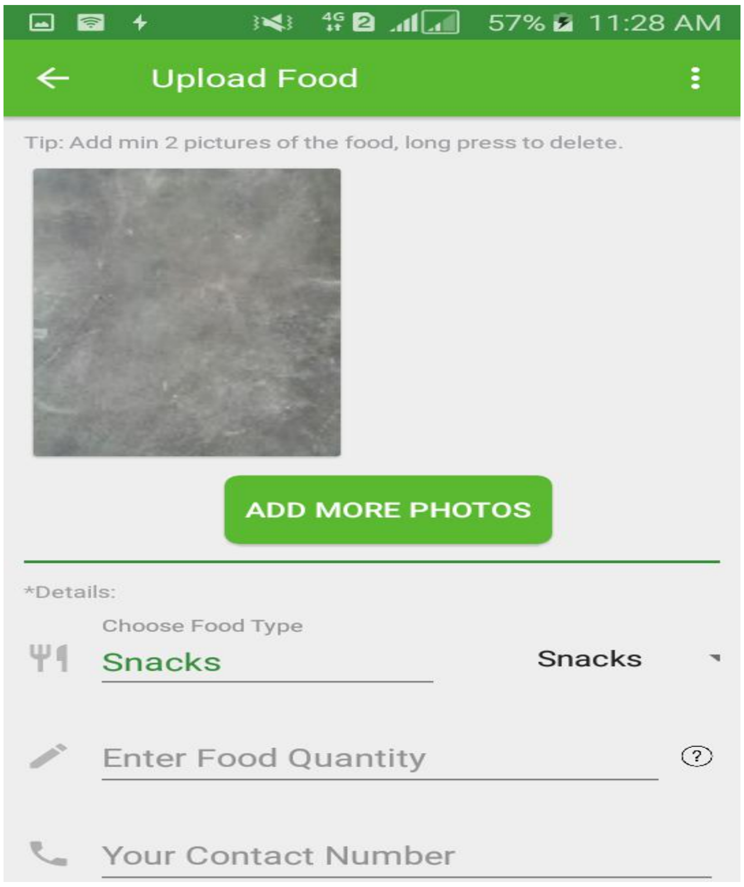
Food Type: Snacks
Contact: 7982646469

Food Type: Other
Contact: 8130758709

Food Type: Li
Contact: 9990

Fig 1.4 DashboardPage

UPLOAD PAGE:


57% 11:28 AM

Upload Food

Tip: Add min 2 pictures of the food, long press to delete.

ADD MORE PHOTOS

*Details:

Choose Food Type

Snacks

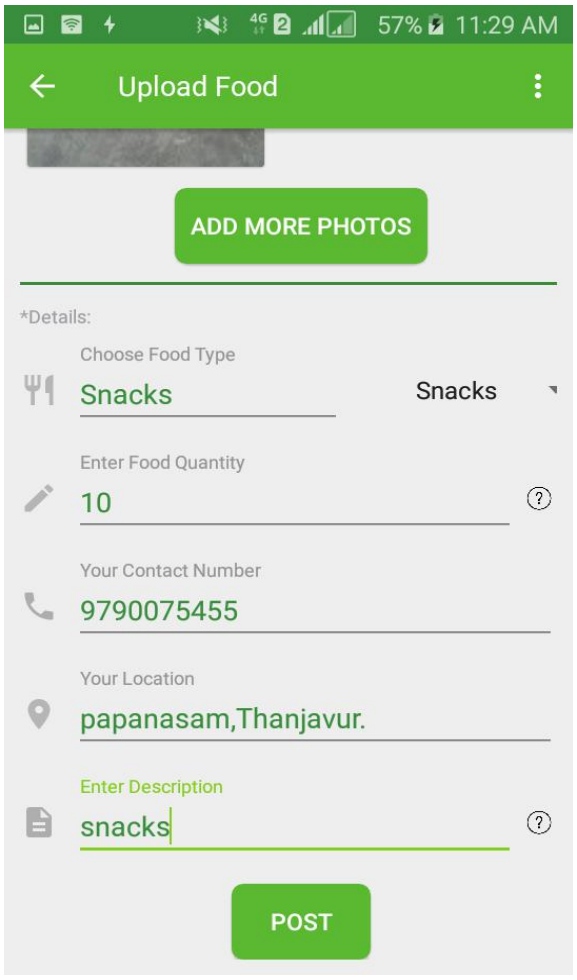
Snacks

Enter Food Quantity

Your Contact Number

1.5 Upload Page

POSTING FOOD DETAILS:



*Details:

Choose Food Type: Snacks

Snacks

Enter Food Quantity: 10

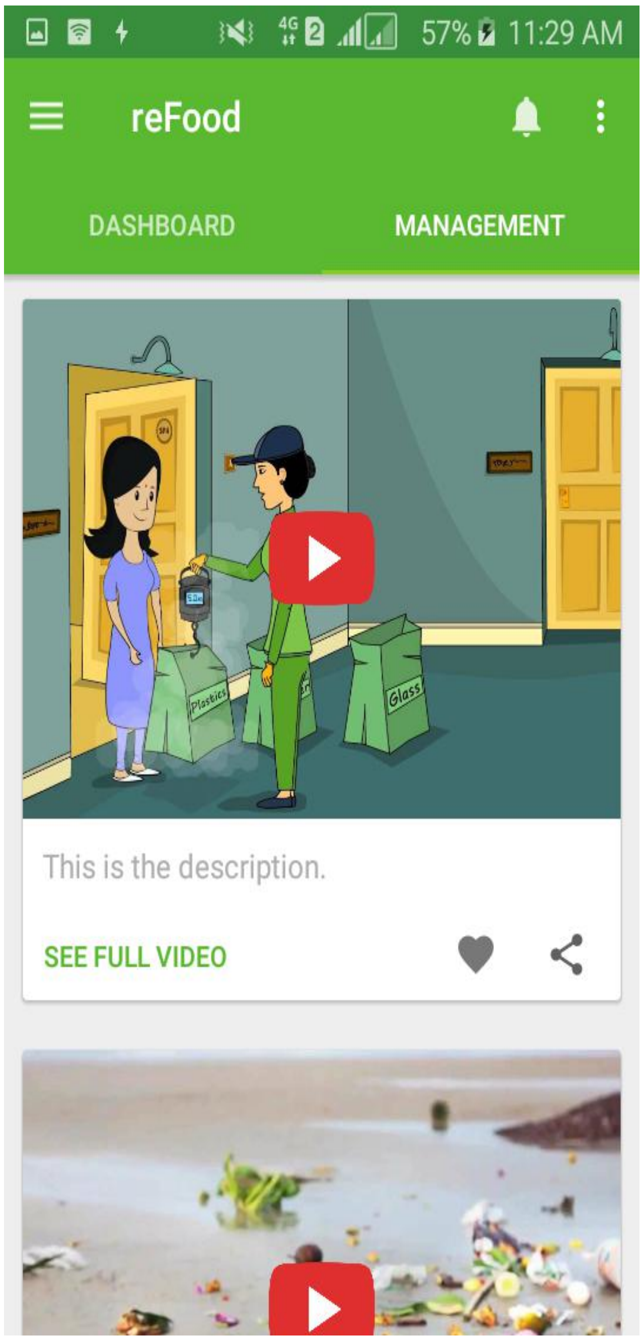
Your Contact Number: 9790075455

Your Location: papanasam,Thanjavur.

Enter Description: snacks

1.6 Food Posting Page

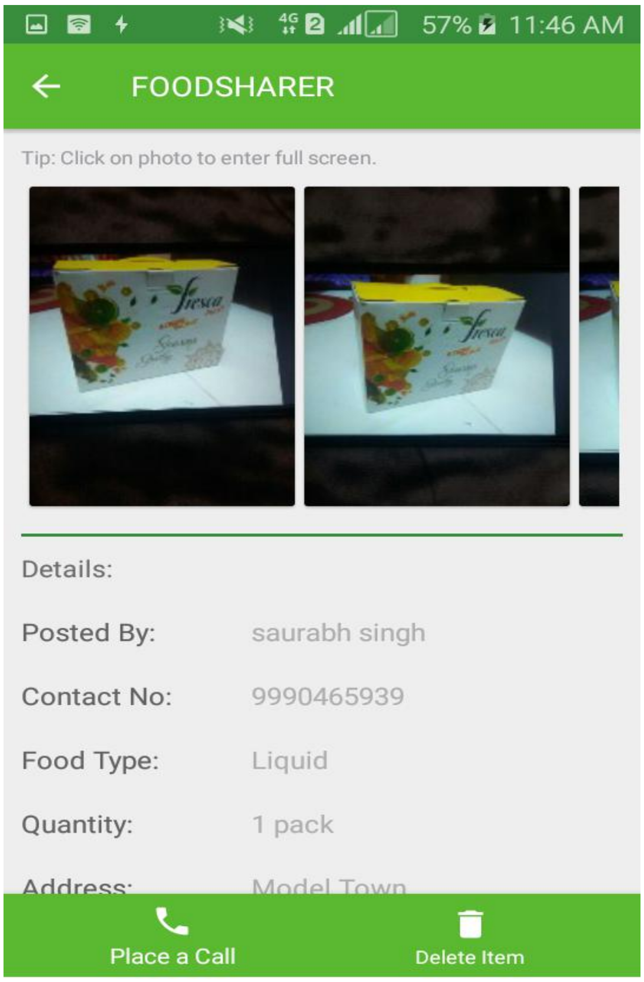
VIDEOS FOR FOOD MANAGEMENT:



This is the description.

1.7 FoodManagent VideoPage

POSTED FOOD:


11:46 AM

FOODSHARER

Tip: Click on photo to enter full screen.

Details:

Posted By: saurabh singh

Contact No: 9990465939

Food Type: Liquid

Quantity: 1 pack

Address: Model Town

1.8 Viewing page

CODE:

FullImageActivity:

```
package com.appsomniac.refood.activity;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Base64;
import android.widget.ImageView;

import com.appsomniac.refood.R;

public class FullImageActivity extends AppCompatActivity {
    ImageView imgFullImage;

//    @Override
//    protected void onCreate(Bundle savedInstanceState) {
//        super.onCreate(savedInstanceState);
//        setContentView(R.layout.activity_full_image);
//
//        ///findViewBYID
//        imgFullImage = (ImageView) findViewById(R.id.full_imageview);
//
//        // Bundle bundle = getIntent().getExtras();
//        //String image = bundle.getString("image");
//        //String image = ModelBase64.base64Image;
//        //Bitmap bitmap = decodeImage(image);
//        imgFullImage.setImageBitmap(bitmap);
//    }
//
//    private Bitmap decodeImage(String data) {
//        byte[] b = Base64.decode(data, Base64.DEFAULT);
//        Bitmap bmp = BitmapFactory.decodeByteArray(b, 0, b.length);
//        return bmp;
//    }
}
```

LoginActivity:

```
package com.appsomniac.refood.activity;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.appsomniac.refood.R;
import com.appsomniac.refood.base.MainActivity;
import com.appsomniac.refood.model.User;
import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.auth.api.signin.GoogleSignInResult;
import com.google.android.gms.common.ConnectionResult;
```

```java
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthCredential;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.GoogleAuthProvider;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

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

public class LoginActivity extends AppCompatActivity implements
GoogleApiClient.OnConnectionFailedListener,
        View.OnClickListener {

    private static final String TAG = "LoginActivity";
    private static final int RC_SIGN_IN = 9001;
    private SignInButton mSignInButton;
    private GoogleApiClient mGoogleApiClient;

    public static int firstLogin = 0;

    //add Firebase Database stuff
    private FirebaseDatabase mFirebaseDatabase;
    private FirebaseAuth mFirebaseAuth;
    private FirebaseAuth.AuthStateListener mAuthListener;
    private DatabaseReference myRef;
    private String userID;

    private EditText inputEmail, inputPassword;
    private ProgressBar progressBar;
    private Button btnSignup, btnLogin, btnReset;


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

        inputEmail = (EditText) findViewById(R.id.email);
        inputPassword = (EditText) findViewById(R.id.password);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);
        btnSignup = (Button) findViewById(R.id.btn_signup);
        btnLogin = (Button) findViewById(R.id.btn_login);
        btnReset = (Button) findViewById(R.id.btn_reset_password);

        btnSignup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(LoginActivity.this,
SignUpActivity.class));
            }
        });

        btnReset.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(LoginActivity.this,
ResetPasswordActivity.class));
            }
        });
        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email = inputEmail.getText().toString();
                final String password = inputPassword.getText().toString();

                if (TextUtils.isEmpty(email)) {
                    Toast.makeText(getApplicationContext(), "Enter email
address!", Toast.LENGTH_SHORT).show();
                    return;
                }

                if (TextUtils.isEmpty(password)) {
                    Toast.makeText(getApplicationContext(), "Enter password!",
Toast.LENGTH_SHORT).show();
                    return;
                }

                progressBar.setVisibility(View.VISIBLE);

                //authenticate user
                mFirebaseAuth.signInWithEmailAndPassword(email, password)
                        .addOnCompleteListener(LoginActivity.this, new
OnCompleteListener<AuthResult>() {
                            @Override
                            public void onComplete(@NonNull Task<AuthResult>
task) {
                                // If sign in fails, display a message to the
user. If sign in succeeds
                                // the auth state listener will be notified and
logic to handle the
                                // signed in user can be handled in the listener.
                                progressBar.setVisibility(View.GONE);
                                if (!task.isSuccessful()) {
                                    // there was an error
                                    if (password.length() < 6) {

inputPassword.setError(getString(R.string.minimum_password));
                                    } else {
                                        Toast.makeText(LoginActivity.this,
getString(R.string.auth_failed), Toast.LENGTH_LONG).show();
                                    }
                                } else {
                                    Intent intent = new
Intent(LoginActivity.this, MainActivity.class);
                                    startActivity(intent);
                                    finish();
                                }
                            }
                        });
            }
        });


        // Assign fields
        mSignInButton = (SignInButton)
findViewById(R.id.google_sign_in_button);

        // Set click listeners
        mSignInButton.setOnClickListener(this);
//
        GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestIdToken(getString(R.string.default_web_client_id))
                .requestEmail()
                .build();
        mGoogleApiClient = new GoogleApiClient.Builder(this)
```

```
                .enableAutoManage(this /* FragmentActivity */, this /*
OnConnectionFailedListener */)
                .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
                .build();

        // Initialize FirebaseAuth
        mFirebaseAuth = FirebaseAuth.getInstance();
    }

    private void handleFirebaseAuthResult(AuthResult authResult) {
        if (authResult != null) {
            // Welcome the user
            FirebaseUser user = authResult.getUser();
            Toast.makeText(this, "Welcome " + user.getDisplayName(),
Toast.LENGTH_SHORT).show();

            SharedPreferences.Editor editor = getSharedPreferences("user_data",
MODE_PRIVATE).edit();
            editor.putString("name", user.getDisplayName());
            editor.putString("email", user.getEmail());
            editor.putString("contact", user.getPhoneNumber());
            editor.putString("photo_url", String.valueOf(user.getPhotoUrl()));
            editor.apply();

        }
    }

    @Override
    public void onClick(View v) {

        switch (v.getId()) {
            case R.id.google_sign_in_button:
                signIn();
                break;
            default:
                return;
        }
    }

    private void signIn() {
        Intent signInIntent =
Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);

        if (mGoogleApiClient != null && mGoogleApiClient.isConnected()) {
            mGoogleApiClient.clearDefaultAccountAndReconnect();
        }

        progressBar.setVisibility(View.VISIBLE);

        startActivityForResult(signInIntent, RC_SIGN_IN);
    }

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

        // Result returned from launching the Intent from
GoogleSignInApi.getSignInIntent(...);
        if (requestCode == RC_SIGN_IN) {
            GoogleSignInResult result =
Auth.GoogleSignInApi.getSignInResultFromIntent(data);
            if (result.isSuccess()) {
                // Google Sign In was successful, authenticate with Firebase
                GoogleSignInAccount account = result.getSignInAccount();
                firebaseAuthWithGoogle(account);
            } else {
                // Google Sign In failed
                Log.e(TAG, "Google Sign In failed.");
            }
        }
    }

    private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {
        Log.d(TAG, "firebaseAuthWithGoogle:" + acct.getId());
        AuthCredential credential =
GoogleAuthProvider.getCredential(acct.getIdToken(), null);
        mFirebaseAuth.signInWithCredential(credential)
                .addOnCompleteListener(this, new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {

                        Log.e(TAG, "signInWithCredential:onComplete:" +
task.isSuccessful());

                        // If sign in fails, display a message to the user. If
sign in succeeds
                        // the auth state listener will be notified and logic
to handle the
                        // signed in user can be handled in the listener.
                        if (!task.isSuccessful()) {
                            Log.e(TAG, "signInWithCredential",
task.getException());
                            Toast.makeText(LoginActivity.this,
"Authentication failed.",
                                    Toast.LENGTH_SHORT).show();
                        } else {

                            progressBar.setVisibility(View.GONE);
                            checkUniqueUser(task.getResult());

                        }
                    }
                });
    }

    @Override
    public void onConnectionFailed(@NonNull ConnectionResult connectionResult)
{
        // An unresolvable error has occurred and Google APIs (including Sign-In)
will not
        // be available.
        Log.d(TAG, "onConnectionFailed:" + connectionResult);
        Toast.makeText(this, "Google Play Services error.",
Toast.LENGTH_SHORT).show();
    }

    public void checkUniqueUser(final AuthResult authResult){

        mFirebaseAuth = FirebaseAuth.getInstance();
        mFirebaseDatabase = FirebaseDatabase.getInstance();
        myRef = mFirebaseDatabase.getReference("users");
        FirebaseUser user = mFirebaseAuth.getCurrentUser();
        userID = user.getUid();

        myRef.child(userID).addListenerForSingleValueEvent(new
ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                if(!dataSnapshot.exists()){

                    addUserToDatabase();
                    handleFirebaseAuthResult(authResult);
                    firstLogin = 1;
                    Log.e("Unique USER:. ", String.valueOf(firstLogin));
                    startActivity(new Intent(getApplicationContext(),
MainActivity.class));
                    finish();

                }else{
                    Log.e("Unique USER:. ", String.valueOf(firstLogin));
                    handleFirebaseAuthResult(authResult);
                    startActivity(new Intent(getApplicationContext(),
MainActivity.class));
                    finish();
                }

            }
            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });

    }

    public void addUserToDatabase(){

        mFirebaseAuth = FirebaseAuth.getInstance();
        mFirebaseDatabase = FirebaseDatabase.getInstance();
        myRef = mFirebaseDatabase.getReference();
        FirebaseUser user = mFirebaseAuth.getCurrentUser();
        userID = user.getUid();

        User userInformation = new
User(user.getDisplayName(),user.getEmail(),user.getPhoneNumber(),
user.getPhotoUrl().toString());
        myRef.child("users").child(userID).setValue(userInformation);

    }
}
```

ResetPasswordActivity:

```
package com.appsomniac.refood.activity;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.appsomniac.refood.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;

public class ResetPasswordActivity extends AppCompatActivity {

    private EditText inputEmail;
    private Button btnReset, btnBack;
    private FirebaseAuth auth;
    private ProgressBar progressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_reset_password);
        inputEmail = (EditText) findViewById(R.id.email);
        btnReset = (Button) findViewById(R.id.btn_reset_password);
        btnBack = (Button) findViewById(R.id.btn_back);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);

        auth = FirebaseAuth.getInstance();

        btnBack.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

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

                String email = inputEmail.getText().toString().trim();

                if (TextUtils.isEmpty(email)) {
                    Toast.makeText(getApplication(), "Enter your registered
email id", Toast.LENGTH_SHORT).show();
                    return;
                }

                progressBar.setVisibility(View.VISIBLE);
                auth.sendPasswordResetEmail(email)
                        .addOnCompleteListener(new OnCompleteListener<Void>()
{
                            @Override
                            public void onComplete(@NonNull Task<Void> task) {
                                if (task.isSuccessful()) {

Toast.makeText(ResetPasswordActivity.this, "We have sent you instructions to
reset your password!", Toast.LENGTH_SHORT).show();
                                } else {

Toast.makeText(ResetPasswordActivity.this, "Failed to send reset email!",
Toast.LENGTH_SHORT).show();
                                }

                                progressBar.setVisibility(View.GONE);
                            }
                        });
            }
        });
    }

}
```

SignupActivity:

```
package com.appsomniac.refood.activity;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
```

```java
import android.widget.Toast;

import com.appsomniac.refood.R;
import com.appsomniac.refood.base.MainActivity;
import com.appsomniac.refood.model.User;
import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.auth.api.signin.GoogleSignInResult;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthCredential;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.GoogleAuthProvider;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class SignUpActivity extends AppCompatActivity implements
GoogleApiClient.OnConnectionFailedListener, View.OnClickListener {

    private EditText inputEmail, inputPassword;
    private Button btnSignIn, btnSignUp, btnResetPassword;
    private ProgressBar progressBar;
    private FirebaseAuth auth;

    public static int firstLogin = 0;
    public static int email_signUp_flag = 0;

    //add Firebase Database stuff
    private FirebaseDatabase mFirebaseDatabase;
    private FirebaseAuth.AuthStateListener mAuthListener;
    private DatabaseReference myRef;
    private String userID;


    private static final String TAG = "SignUpnActivity";
    private static final int RC_SIGN_IN = 9001;
    private SignInButton mSignInButton;
    private GoogleApiClient mGoogleApiClient;
    private FirebaseAuth mFirebaseAuth;


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

        // Initialize FirebaseAuth
        mFirebaseAuth = FirebaseAuth.getInstance();

        //Get Firebase auth instance
        auth = FirebaseAuth.getInstance();

        btnSignIn = (Button) findViewById(R.id.sign_in_button);
        btnSignUp = (Button) findViewById(R.id.sign_up_button);
        inputEmail = (EditText) findViewById(R.id.email);
        inputPassword = (EditText) findViewById(R.id.password);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);

        btnSignIn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

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

                String email = inputEmail.getText().toString().trim();
                String password = inputPassword.getText().toString().trim();

                if (TextUtils.isEmpty(email)) {
                    Toast.makeText(getApplicationContext(), "Enter email
address!", Toast.LENGTH_SHORT).show();
                    return;
                }

                if (TextUtils.isEmpty(password)) {
                    Toast.makeText(getApplicationContext(), "Enter password!",
Toast.LENGTH_SHORT).show();
                    return;
                }

                if (password.length() < 6) {
                    Toast.makeText(getApplicationContext(), "Password too
short, enter minimum 6 characters!", Toast.LENGTH_SHORT).show();
                    return;
                }

                progressBar.setVisibility(View.VISIBLE);
                //create user
                auth.createUserWithEmailAndPassword(email, password)
                        .addOnCompleteListener(SignUpActivity.this, new
OnCompleteListener<AuthResult>() {
                            @Override
                            public void onComplete(@NonNull Task<AuthResult>
task) {
                                Toast.makeText(SignUpActivity.this,
"Registered Successfully:", Toast.LENGTH_SHORT).show();
                                progressBar.setVisibility(View.GONE);
                                // If sign in fails, display a message to the
user. If sign in succeeds
                                // the auth state listener will be notified and
logic to handle the
                                // signed in user can be handled in the listener.
                                if (!task.isSuccessful()) {
                                    Toast.makeText(SignUpActivity.this, "User
already exists.",
                                            Toast.LENGTH_SHORT).show();
                                } else {

                                    email_signUp_flag=1;
                                    checkUniqueUser(task.getResult());
                                    startActivity(new
Intent(SignUpActivity.this, MainActivity.class));
                                    finish();
                                }
                            }
                        });
            }
        });

        // Assign fields
        mSignInButton = (SignInButton)
findViewById(R.id.google_sign_in_button);
```

```
        // Set click listeners
        mSignInButton.setOnClickListener(this);
//
        GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestIdToken(getString(R.string.default_web_client_id))
                .requestEmail()
                .build();
        mGoogleApiClient = new GoogleApiClient.Builder(this)
                .enableAutoManage(this /* FragmentActivity */, this /*
OnConnectionFailedListener */)
                .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
                .build();

        // Initialize FirebaseAuth
        mFirebaseAuth = FirebaseAuth.getInstance();
    }

    private void handleFirebaseAuthResult(AuthResult authResult) {
        if (authResult != null) {
            // Welcome the user
            FirebaseUser user = authResult.getUser();
            Toast.makeText(this, "Welcome " + user.getDisplayName(),
Toast.LENGTH_SHORT).show();

            SharedPreferences.Editor editor = getSharedPreferences("user_data",
MODE_PRIVATE).edit();
            editor.putString("name", user.getDisplayName());
            editor.putString("email", user.getEmail());
            editor.putString("contact", user.getPhoneNumber());
            editor.putString("photo_url", String.valueOf(user.getPhotoUrl()));
            editor.apply();
            // Go back to the main activity
            startActivity(new Intent(this, MainActivity.class));
            finish();
        }
    }


    @Override
    public void onClick(View v) {

        switch (v.getId()) {
            case R.id.google_sign_in_button:
                signIn();
                break;
            default:
                return;
        }
    }

    private void signIn() {
        Intent signInIntent =
Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);

        if (mGoogleApiClient != null && mGoogleApiClient.isConnected()) {
            mGoogleApiClient.clearDefaultAccountAndReconnect();
        }
        startActivityForResult(signInIntent, RC_SIGN_IN);
    }

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

        // Result returned from launching the Intent from
GoogleSignInApi.getSignInIntent(...);
```

```
        if (requestCode == RC_SIGN_IN) {
            GoogleSignInResult result =
Auth.GoogleSignInApi.getSignInResultFromIntent(data);
            if (result.isSuccess()) {
                // Google Sign In was successful, authenticate with Firebase
                GoogleSignInAccount account = result.getSignInAccount();
                firebaseAuthWithGoogle(account);
            } else {
                // Google Sign In failed
                Log.e(TAG, "Google Sign In failed.");
            }
        }
    }


    private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {
        Log.d(TAG, "firebaseAuthWithGoogle:" + acct.getId());
        AuthCredential credential =
GoogleAuthProvider.getCredential(acct.getIdToken(), null);
        mFirebaseAuth.signInWithCredential(credential)
                .addOnCompleteListener(this, new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {

                        Log.e(TAG, "signInWithCredential:onComplete:" +
task.isSuccessful());

                        // If sign in fails, display a message to the user. If
sign in succeeds
                        // the auth state listener will be notified and logic
to handle the
                        // signed in user can be handled in the listener.
                        if (!task.isSuccessful()) {
                            Log.e(TAG, "signInWithCredential",
task.getException());

                            Toast.makeText(SignUpActivity.this,
"Authentication failed.",
                                    Toast.LENGTH_SHORT).show();
                        } else {

                            checkUniqueUser(task.getResult());

                        }
                    }
                });
    }

    @Override
    public void onConnectionFailed(@NonNull ConnectionResult connectionResult)
{
        // An unresolvable error has occurred and Google APIs (including Sign-In)
will not
        // be available.
        Log.d(TAG, "onConnectionFailed:" + connectionResult);
        Toast.makeText(this, "Google Play Services error.",
Toast.LENGTH_SHORT).show();
    }


    @Override
    protected void onResume() {
        super.onResume();
        progressBar.setVisibility(View.GONE);
    }

    public void checkUniqueUser(final AuthResult authResult){
        mFirebaseAuth = FirebaseAuth.getInstance();
        mFirebaseDatabase = FirebaseDatabase.getInstance();
        myRef = mFirebaseDatabase.getReference("users");
        FirebaseUser user = mFirebaseAuth.getCurrentUser();
        userID = user.getUid();

        myRef.child(userID).addListenerForSingleValueEvent(new
ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                if(!dataSnapshot.exists()){

                    if(email_signUp_flag == 1) {

                        addUserToDatabaseViaEmail();
                    }else {

                        addUserToDatabase();
                    }

                    handleFirebaseAuthResult(authResult);
                    firstLogin = 1;
                    Log.e("Unique USER:. ", String.valueOf(firstLogin));
                    startActivity(new Intent(getApplicationContext(),
MainActivity.class));
                    finish();


                }else{
                    Log.e("Unique USER:. ", String.valueOf(firstLogin));
                    handleFirebaseAuthResult(authResult);
                    startActivity(new Intent(getApplicationContext(),
MainActivity.class));
                    finish();
                }

            }
            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });

    }

    public void addUserToDatabaseViaEmail(){

        mFirebaseAuth = FirebaseAuth.getInstance();
        mFirebaseDatabase = FirebaseDatabase.getInstance();
        myRef = mFirebaseDatabase.getReference();
        FirebaseUser user = mFirebaseAuth.getCurrentUser();
        userID = user.getUid();

        User userInformation = new
User(user.getDisplayName(),user.getEmail(),user.getPhoneNumber(), "");
        myRef.child("users").child(userID).setValue(userInformation);

    }

    public void addUserToDatabase(){

        mFirebaseAuth = FirebaseAuth.getInstance();
        mFirebaseDatabase = FirebaseDatabase.getInstance();
        myRef = mFirebaseDatabase.getReference();
        FirebaseUser user = mFirebaseAuth.getCurrentUser();
        userID = user.getUid();
        User userInformation = new
User(user.getDisplayName(),user.getEmail(),user.getPhoneNumber(),
user.getPhotoUrl().toString());
        myRef.child("users").child(userID).setValue(userInformation);

    }
}
```

SingleItemActivity:

```
package com.appsomniac.refood.activity;

import android.content.Intent;
import android.net.Uri;
import android.support.annotation.NonNull;
import android.support.design.widget.BottomNavigationView;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.view.MenuItem;
import android.widget.TextView;
import android.widget.Toast;
import com.appsomniac.refood.R;
import com.appsomniac.refood.adapter.uploadActivity.Radapter;
import com.appsomniac.refood.base.MainActivity;
import com.appsomniac.refood.classFragments.DashboardFragment;
import com.appsomniac.refood.model.FoodPost;
import com.google.android.gms.maps.model.Dash;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class SingleItemActivity extends AppCompatActivity {

    private FoodPost foodPost;
    private int position;
    Radapter adapter;
    RecyclerView my_recycler_view;

    //add Firebase Database stuff
    private FirebaseDatabase mFirebaseDatabase;
    private FirebaseAuth mFirebaseAuth;
    private FirebaseAuth.AuthStateListener mAuthListener;
    private DatabaseReference myRef;
    private String userID;

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

        // Adding Toolbar to Main screen
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        position = getIntent().getIntExtra("position", 0);
        setBottomNavView();

        my_recycler_view = (RecyclerView) findViewById(R.id.my_recycler_view);
        my_recycler_view.setHasFixedSize(true);
        my_recycler_view.setNestedScrollingEnabled(true);

        adapter = new Radapter(this,
DashboardFragment.all_posts.get(position).getAl_imageEncoded(), "singleview");
        my_recycler_view.setLayoutManager(new
LinearLayoutManager(getApplicationContext(), LinearLayoutManager.HORIZONTAL,
false));
        my_recycler_view.setAdapter(adapter);

        setItemValues();

    }

    public void setItemValues(){

        TextView postedBy = (TextView) findViewById(R.id.postedBy_value);

postedBy.setText(DashboardFragment.all_posts.get(position).getFoodPostedByNa
me());

        TextView contact = (TextView) findViewById(R.id.contact_value);

contact.setText(DashboardFragment.all_posts.get(position).getContact());

        TextView foodType = (TextView) findViewById(R.id.foodtype_value);

foodType.setText(DashboardFragment.all_posts.get(position).getFoodType());

        TextView quantity = (TextView) findViewById(R.id.quantity_value);

quantity.setText(DashboardFragment.all_posts.get(position).getFoodQunatity())
;

        TextView description = (TextView) findViewById(R.id.description_value);

description.setText(DashboardFragment.all_posts.get(position).getFoodDescrip
tion());

        TextView address = (TextView) findViewById(R.id.address_value);

address.setText(DashboardFragment.all_posts.get(position).getFoodLocation());
    }

    public void setBottomNavView(){

        BottomNavigationView bottomBar = (BottomNavigationView)
findViewById(R.id.bottom_navigation);
        bottomBar.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {

                switch (item.getItemId()){

                    case R.id.call:

                        Intent dialIntent = new Intent(Intent.ACTION_DIAL);
                        dialIntent.setData(Uri.parse("tel:" +
DashboardFragment.all_posts.get(position).getContact()));
                        startActivity(dialIntent);

                        break;
                    case R.id.delete:

                        checkUserAndDeleteFood();
                        break;
                }
                return true;
            }
        });
    }
    public void checkUserAndDeleteFood(){

        mFirebaseAuth = FirebaseAuth.getInstance();
        mFirebaseDatabase = FirebaseDatabase.getInstance();
        myRef = mFirebaseDatabase.getReference();
        FirebaseUser user = mFirebaseAuth.getCurrentUser();
        userID = user.getUid();


if(userID.equals(DashboardFragment.all_posts.get(position).getFoodPostedByUs
erId())){

            DatabaseReference reference =
FirebaseDatabase.getInstance().getReference().child("all_posts").child(Dashb
oardFragment.all_posts.get(position).getRefKey());
            reference.removeValue();

            startActivity(new Intent(this, MainActivity.class));

        }else{
            Toast.makeText(getApplicationContext(), "You can't delete this
item", Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    public void onBackPressed() {

        finish();
        super.onBackPressed();
    }


    @Override
    public boolean onSupportNavigateUp(){

        finish();
        return true;
    }
}
```

UploadActivity:

```
package com.appsomniac.refood.activity;

import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.media.MediaScannerConnection;
import android.net.Uri;
import android.os.Environment;
import android.provider.MediaStore;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.util.Base64;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.Toast;

import com.appsomniac.refood.R;
import com.appsomniac.refood.base.MainActivity;
import com.appsomniac.refood.adapter.uploadActivity.Radapter;
import com.appsomniac.refood.model.FoodPost;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;

public class UploadActivity extends AppCompatActivity {

    ArrayList<String> image_uris;
    ArrayList<String> al_image_encoded;
    private RecyclerView rv;
    public Button add_more, post_btn;

    //add Firebase Database stuff
    private FirebaseDatabase mFirebaseDatabase;
    private FirebaseAuth mFirebaseAuth;
    private FirebaseAuth.AuthStateListener mAuthListener;
    private DatabaseReference myRef;
    private String userID;

    EditText editFoodType, editFoodQuantity, editUserLocation, editUserContact,
editFoodDescription;
    Spinner foodTypeSpinner;
    Radapter adapter;
    RecyclerView my_recycler_view;
    private static final String IMAGE_DIRECTORY = "/reFood";
    private int GALLERY = 1, CAMERA = 2;
    String[] permissions = new String[]{
            android.Manifest.permission.CAMERA,
            android.Manifest.permission.WRITE_EXTERNAL_STORAGE,
            android.Manifest.permission.READ_EXTERNAL_STORAGE,
    };

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

        initializeViews();

        image_uris = getIntent().getStringArrayListExtra("image_uris");
        al_image_encoded =
getIntent().getStringArrayListExtra("al_image_encoded");

        if(image_uris.size()<=3) {

            add_more.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    checkPermissions();
                    showPictureDialog();
                }
            });
        }else
            if(image_uris.size()>=4){
                add_more.setEnabled(false);
            }


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

                if(editFoodQuantity.getText().length()>=0 &&
editFoodType.getText().length()>=1 && editUserLocation.getText().length()>=4
                        && editUserContact.getText().length()>=7 &&
image_uris.size()>=2){

                    postToDatabase();
                    startActivity(new Intent(getApplicationContext(),
MainActivity.class));
                    finish();

                }else {

                    if(image_uris.size()<2){
                        Toast.makeText(getApplicationContext(), "Add atleast 2
photos of the food.", Toast.LENGTH_SHORT).show();
                    }else {
                        Toast.makeText(getApplicationContext(), "Complete
Details first!", Toast.LENGTH_SHORT).show();
                    }

                }
            }
        });

        ImageView foodQuantityHint = (ImageView)
findViewById(R.id.foodQuantityHint);
        foodQuantityHint.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Toast.makeText(getApplicationContext(), "Eg: 1 bottle, 2
plates", Toast.LENGTH_SHORT).show();

            }
        });

        ImageView foodDescriptionHint = (ImageView)
findViewById(R.id.foodDescriptionHint);
        foodDescriptionHint.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Toast.makeText(getApplicationContext(), "Eg: 1 bottle frooti
available.", Toast.LENGTH_SHORT).show();

            }
        });

        my_recycler_view = (RecyclerView) findViewById(R.id.my_recycler_view);
        my_recycler_view.setHasFixedSize(true);
        my_recycler_view.setNestedScrollingEnabled(true);
        adapter = new Radapter(this, image_uris, "upload");
        my_recycler_view.setLayoutManager(new
LinearLayoutManager(getApplicationContext(), LinearLayoutManager.HORIZONTAL,
false));
        my_recycler_view.setAdapter(adapter);

    }

    public void postToDatabase(){

        mFirebaseAuth = FirebaseAuth.getInstance();
        mFirebaseDatabase = FirebaseDatabase.getInstance();
        myRef = mFirebaseDatabase.getReference();
        FirebaseUser user = mFirebaseAuth.getCurrentUser();
        userID = user.getUid();

        //no need to add the same the same posts to user structre as it can be
related by all_posts using user_id.

//myRef.child("users").child(userID).child("posts").push().setValue(post);

        //to get the same push() key, call push() only once;
        DatabaseReference newRef = myRef.child("all_posts").push();

        String refKey = newRef.getKey();

        SharedPreferences prefs = getSharedPreferences("user_data",
MODE_PRIVATE);
        String user_name = prefs.getString("name", "User");

        FoodPost post = new FoodPost(editFoodType.getText().toString(),
editFoodQuantity.getText().toString(), user_name
                , userID, editUserLocation.getText().toString(),
editUserContact.getText().toString(), editFoodDescription.getText().toString()
                , al_image_encoded, refKey);

        newRef.setValue(post);
    }

    public void initializeViews(){

        // Adding Toolbar to Main screen
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        add_more = (Button) findViewById(R.id.button_add_more);
        post_btn = (Button) findViewById(R.id.button_post);
        editFoodType = (EditText) findViewById(R.id.foodTypeText);
        editFoodQuantity = (EditText) findViewById(R.id.foodQuantity);
        editUserContact = (EditText) findViewById(R.id.senderContact);
        editUserLocation = (EditText) findViewById(R.id.senderLocation);
        editFoodDescription = (EditText) findViewById(R.id.foodDescription);
        foodTypeSpinner = (Spinner) findViewById(R.id.foodTypeSpinner);

        addListenerOnSpinnerItemSelection();
    }

    public void addListenerOnSpinnerItemSelection() {

        foodTypeSpinner.setOnItemSelectedListener(new
CustomOnItemSelectedListener());
    }

    public class CustomOnItemSelectedListener implements
AdapterView.OnItemSelectedListener {

        public void onItemSelected(AdapterView<?> parent, View view, int pos,
long id) {

editFoodType.setText(foodTypeSpinner.getSelectedItem().toString());
        }

        @Override
        public void onNothingSelected(AdapterView<?> arg0) {
            // TODO Auto-generated method stub
        }
    }


    private boolean checkPermissions() {
        int result;
        List<String> listPermissionsNeeded = new ArrayList<>();
        for (String p : permissions) {
            result = ContextCompat.checkSelfPermission(this, p);
            if (result != PackageManager.PERMISSION_GRANTED) {
                listPermissionsNeeded.add(p);
            }
        }
        if (!listPermissionsNeeded.isEmpty()) {
            ActivityCompat.requestPermissions(this,
listPermissionsNeeded.toArray(new String[listPermissionsNeeded.size()]), 100);
            return false;
        }
        return true;
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String
permissions[], int[] grantResults) {
        if (requestCode == 100) {
            if (grantResults.length > 0
                    && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

            }
            return;
        }
    }
    private void showPictureDialog(){
        AlertDialog.Builder pictureDialog = new AlertDialog.Builder(this);
        pictureDialog.setTitle("Select Action");
//        String[] pictureDialogItems = {
//                "Select photo from gallery",
//                "Capture photo from camera" };

        String[] pictureDialogItems = {
                "Capture photo from camera" };
        pictureDialog.setItems(pictureDialogItems,
                new DialogInterface.OnClickListener(){
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        switch (which) {
                            case 0:
                                takePhotoFromCamera();
                                break;
                        }
                    }
                });
        pictureDialog.show();
    }

    public void choosePhotoFromGallary() {
        Intent galleryIntent = new Intent(Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        startActivityForResult(galleryIntent, GALLERY);
    }

    private void takePhotoFromCamera() {

        Intent intent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(intent, CAMERA);
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data)
{

        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode == this.RESULT_CANCELED) {
            return;
        }
        if (requestCode == GALLERY) {
            if (data != null) {
                Uri contentURI = data.getData();
                try {
                    Bitmap bitmap =
MediaStore.Images.Media.getBitmap(this.getContentResolver(), contentURI);
                    Log.e("Bitmap: ", String.valueOf(bitmap));

                    String path = saveImage(bitmap);
                    Toast.makeText(UploadActivity.this, "Image Saved!",
Toast.LENGTH_SHORT).show();
                } catch (IOException e) {
                    e.printStackTrace();
                    Toast.makeText(UploadActivity.this, "Failed!",
Toast.LENGTH_SHORT).show();
                }
            }
        } else if (requestCode == CAMERA) {
            Bitmap thumbnail = (Bitmap) data.getExtras().get("data");
            Log.e("Bitmap: ", String.valueOf(thumbnail));
            saveImage(thumbnail);
            Toast.makeText(UploadActivity.this, "Image Saved!",
Toast.LENGTH_SHORT).show();
        }
    }

    public String saveImage(Bitmap myBitmap) {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        myBitmap.compress(Bitmap.CompressFormat.JPEG, 90, baos);

        String imageEncoded = Base64.encodeToString(baos.toByteArray(),
Base64.DEFAULT);
        al_image_encoded.add(imageEncoded);

        File wallpaperDirectory = new File(
                Environment.getExternalStorageDirectory() + IMAGE_DIRECTORY);
        // have the object build the directory structure, if needed.
        if (!wallpaperDirectory.exists()) {
            wallpaperDirectory.mkdirs();
        }

        try {
            File f = new File(wallpaperDirectory, Calendar.getInstance()
                    .getTimeInMillis() + ".jpg");
            f.createNewFile();
            FileOutputStream fo = new FileOutputStream(f);
            fo.write(baos.toByteArray());
            MediaScannerConnection.scanFile(this,
                    new String[]{f.getPath()},
                    new String[]{"image/jpeg"}, null);
            fo.close();
//            Log.d("TAG", "File Saved::--->" + f.getAbsolutePath());

            image_uris.add(f.getAbsolutePath());
            if(image_uris.size()>=4){
                add_more.setEnabled(false);
            }
            adapter.notifyDataSetChanged();
            return f.getAbsolutePath();
        } catch (IOException e1) {
            e1.printStackTrace();
        }
        return "";
    }

    @Override
    public void onBackPressed() {
        image_uris.clear();
        adapter.notifyDataSetChanged();
        finish();
        super.onBackPressed();
        //finish();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_upload, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.cancel:
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }

    @Override
    public boolean onSupportNavigateUp(){

        finish();
        return true;
    }
}
```

MainActivity:

```
package com.appsomniac.refood.base;

import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.util.Log;
import android.view.View;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.appsomniac.refood.R;
import com.appsomniac.refood.activity.LoginActivity;
import com.appsomniac.refood.fragments.HomeFragment;
import com.appsomniac.refood.fragments.ProfileFragment;
import com.appsomniac.refood.service.FirebaseNotificationService;
import com.bumptech.glide.Glide;
import com.bumptech.glide.request.RequestOptions;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.FirebaseDatabase;

public class MainActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener,
HomeFragment.OnFragmentInteractionListener,
        ProfileFragment.OnFragmentInteractionListener{

    // Firebase instance variables
    private FirebaseAuth mFirebaseAuth;
    private FirebaseUser mFirebaseUser;
    private String mUsername;

    FloatingActionButton fab_camera;
    NavigationView nav;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize Firebase Auth
        mFirebaseAuth = FirebaseAuth.getInstance();
        mFirebaseUser = mFirebaseAuth.getCurrentUser();


        if (mFirebaseUser == null) {
            // Not signed in, launch the Sign In activity
            startActivity(new Intent(this, LoginActivity.class));
            finish();
            return;
        } else {
            mUsername = mFirebaseUser.getDisplayName();
            //startService(new Intent(getBaseContext(),
FirebaseNotificationService.class));
        }

        // initialize the views
        initializeViews();

        //set the avatar in navigationView
        setUserProfileInNavigationView();

        //Initially land to HomeFragment
        Fragment fragment = new HomeFragment();
        //replacing the fragment
        if (fragment != null) {
            FragmentTransaction ft =
getSupportFragmentManager().beginTransaction();
            ft.replace(R.id.content_frame, fragment);
            ft.commit();
        }
    }


    public void setUserProfileInNavigationView(){

        nav = ( NavigationView ) findViewById( R.id.nav_view );
        if( nav != null ){
            LinearLayout mParent = ( LinearLayout ) nav.getHeaderView( 0 );

            if( mParent != null ){
                // Set your values to the image and text view by declaring and
setting as you need to here.

                SharedPreferences prefs = getSharedPreferences("user_data",
MODE_PRIVATE);

                String photoUrl = prefs.getString("photo_url", null);
                String user_name = prefs.getString("name", "User");

                if(photoUrl!=null) {
                    Log.e("Photo Url: ", photoUrl);

                    TextView userName = (TextView)
mParent.findViewById(R.id.user_name);
                    userName.setText(user_name);

                    ImageView user_imageView = (ImageView)
mParent.findViewById(R.id.user_avatar);

                    RequestOptions requestOptions = new RequestOptions();

requestOptions.placeholder(R.drawable.ic_person_black_24dp);
                    requestOptions.error(R.drawable.ic_person_black_24dp);

                    Glide.with(this).load(photoUrl)
                            .apply(requestOptions).thumbnail(0.5f).into(user
_imageView);
                }
            }
        }
    }


    public void initializeViews(){

        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        fab_camera = (FloatingActionButton) findViewById(R.id.fab);

        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        drawer.setDrawerListener(toggle);
        toggle.syncState();

        nav = (NavigationView) findViewById(R.id.nav_view);
        nav.setNavigationItemSelectedListener(this);

    }

    @Override
    public void onBackPressed() {
        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START)) {
            drawer.closeDrawer(GravityCompat.START);
        } else {
            finish();
            super.onBackPressed();
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.sign_out_menu:
                mFirebaseAuth.signOut();
                //Auth.GoogleSignInApi.signOut(mGoogleApiClient);
                mUsername = "ANONMOUS";
                startActivity(new Intent(MainActivity.this,
LoginActivity.class));
                finish();
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }

    @SuppressWarnings("StatementWithEmptyBody")
    @Override
    public boolean onNavigationItemSelected(MenuItem item) {

        //creating fragment object
        Fragment fragment = null;

        // Handle navigation view item clicks here.
        int id = item.getItemId();

        if (id == R.id.nav_dashboard) {

            fragment = new HomeFragment();
            fab_camera.show();

        } else if (id == R.id.nav_profile) {
            fragment = new ProfileFragment();
            fab_camera.setVisibility(View.GONE);

        } else if (id == R.id.nav_settings) {

            fab_camera.setVisibility(View.GONE);

        } else if (id == R.id.nav_coins) {

            fab_camera.setVisibility(View.GONE);

        } else if (id == R.id.nav_refer_and_earn) {

            fab_camera.setVisibility(View.GONE);

        }

        //replacing the fragment
        if (fragment != null) {
            FragmentTransaction ft =
getSupportFragmentManager().beginTransaction();
            ft.replace(R.id.content_frame, fragment);
            ft.commit();
        }
        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        drawer.closeDrawer(GravityCompat.START);
        return true;
    }

    public void setActionBarTitle(String title) {
        getSupportActionBar().setTitle(title);
    }

    @Override
    public void onFragmentInteraction(Uri uri) {

    }
}
```

RefoodApplication:

```
package com.appsomniac.refood.base;

import android.app.Application;
import com.google.firebase.database.FirebaseDatabase;
public class ReFoodApplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();
        /* Enable disk persistence  */
        FirebaseDatabase.getInstance().setPersistenceEnabled(true);
    }
}
```

SplashActivity:

```
package com.appsomniac.refood.base;


import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

import com.appsomniac.refood.R;
import com.appsomniac.refood.model.FoodPost;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class SplashActivity extends Activity {
    // Splash screen timer
    private static int SPLASH_TIME_OUT = 2000;

    //add Firebase Database stuff
    private FirebaseDatabase mFirebaseDatabase;
    private FirebaseAuth.AuthStateListener mAuthListener;
    private DatabaseReference myRef;
    private String userID;

    // Firebase instance variables
    private FirebaseAuth mFirebaseAuth;
    private FirebaseUser mFirebaseUser;
    private String mUsername;

    public static ArrayList<FoodPost> all_posts;

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

        all_posts = new ArrayList<FoodPost>();
        new Handler().postDelayed(new Runnable() {

            /*
             * Showing splash screen with a timer. This will be useful when you
             * want to show case your app logo / company
             */

            @Override
            public void run() {

                Intent i = new Intent(SplashActivity.this, MainActivity.class);
                startActivity(i);
                finish();
            }
        }, SPLASH_TIME_OUT);
    }


    public void getAllPostsFromDatabase(){

        mFirebaseDatabase = FirebaseDatabase.getInstance();
        myRef = mFirebaseDatabase.getReference("all_posts");

        myRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                for(DataSnapshot dataSnapshot1: dataSnapshot.getChildren()){

                    FoodPost posts = dataSnapshot1.getValue(FoodPost.class);

                    all_posts.add(posts);
                }
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });

    }

}
```

DashboardFragment:

```
package com.appsomniac.refood.classFragments;

import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.media.MediaScannerConnection;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
```

```java
import android.support.design.widget.FloatingActionButton;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.Fragment;
import android.support.v4.content.ContextCompat;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Base64;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.LayoutAnimationController;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.RelativeLayout;
import android.widget.Toast;

import com.appsomniac.refood.R;
import com.appsomniac.refood.activity.UploadActivity;
import com.appsomniac.refood.adapter.dashboard.DashboardRadapter;
import com.appsomniac.refood.model.FoodPost;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;

public class DashboardFragment extends Fragment {

    View dashboard_fragment;
    FloatingActionButton fab_camera;
    DashboardRadapter adapter;
    RecyclerView my_recycler_view;
    private Button btn;
    private ImageView imageview;
    ProgressBar progressBar;

    LayoutAnimationController controller;

    //add Firebase Database stuff
    private FirebaseDatabase mFirebaseDatabase;
    private FirebaseAuth mFirebaseAuth;
    private FirebaseAuth.AuthStateListener mAuthListener;
    private DatabaseReference myRef;
    private String userID;

    public static ArrayList<FoodPost> all_posts;

    private ArrayList<String> image_uris;
    private ArrayList<String> al_image_encoded;

    private static final String IMAGE_DIRECTORY = "/reFood";
    private int GALLERY = 1, CAMERA = 2;
    private RecyclerView rv;
    String[] permissions = new String[]{
            android.Manifest.permission.CAMERA,
            android.Manifest.permission.WRITE_EXTERNAL_STORAGE,
            android.Manifest.permission.READ_EXTERNAL_STORAGE,
    };

    public DashboardFragment() {
        //empty constructor
    }

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

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {

        dashboard_fragment = inflater.inflate(R.layout.dashboard_layout,
container, false);

        checkPermissions();

        fab_camera = getActivity().findViewById(R.id.fab);
        image_uris=new ArrayList<String>();
        al_image_encoded = new ArrayList<>();
        progressBar =
dashboard_fragment.findViewById(R.id.dashboard_progressBar);
        progressBar.setVisibility(View.VISIBLE);

        getAllPostsFromDatabase();
        setListenersOnFab();

        return dashboard_fragment;
    }

    public void getAllPostsFromDatabase(){

        mFirebaseDatabase = FirebaseDatabase.getInstance();
        myRef = mFirebaseDatabase.getReference("all_posts");

        myRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                all_posts = new ArrayList<FoodPost>();
                for(DataSnapshot dataSnapshot1: dataSnapshot.getChildren()){

                    FoodPost posts = dataSnapshot1.getValue(FoodPost.class);

                    all_posts.add(posts);
                }

                if(all_posts.size()!=0) {

                    my_recycler_view = (RecyclerView)
dashboard_fragment.findViewById(R.id.dashboard_frag_recycler_view);
                    my_recycler_view.setHasFixedSize(true);
                    my_recycler_view.setNestedScrollingEnabled(true);

                    //need to get the arrayList of the post POJOs.
                    adapter = new DashboardRadapter(getContext(), all_posts,
my_recycler_view);
                    my_recycler_view.setLayoutManager(new
GridLayoutManager(getContext(), 2));

                    my_recycler_view.setAdapter(adapter);
                    progressBar.setVisibility(View.GONE);

                    //This methos is for FADEin FADEout animation of each card
                    setAnimationAndAdapter();
```

```java
                }else
                    if(all_posts.size()==0){

                        RelativeLayout placeholder =
dashboard_fragment.findViewById(R.id.empty_placeholder);
                        progressBar.setVisibility(View.GONE);
                        placeholder.setVisibility(View.VISIBLE);
                    }
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });

    }

    public void setAnimationAndAdapter(){

        //Animation to recyclerView
//        AnimationSet set = new AnimationSet(true);
//        Animation animation = new AlphaAnimation(0.0f, 1.0f);
//        animation.setDuration(500);
//        set.addAnimation(animation);
//
//        animation = new TranslateAnimation(
//              Animation.RELATIVE_TO_SELF, 0.0f, Animation.RELATIVE_TO_SELF,
0.0f,
//             Animation.RELATIVE_TO_SELF, -1.0f, Animation.RELATIVE_TO_SELF,
0.0f
//        );
//        animation.setDuration(100);
//        set.addAnimation(animation);

//        controller = new LayoutAnimationController(set, 0.5f);
//        my_recycler_view.setAdapter(adapter);
//        my_recycler_view.setLayoutAnimation(controller);
//        progressBar.setVisibility(View.GONE);

    }

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

                checkPermissions();
                showPictureDialog();
            }
        });
    }

    private boolean checkPermissions() {
        int result;
        List<String> listPermissionsNeeded = new ArrayList<>();
        for (String p : permissions) {
            result = ContextCompat.checkSelfPermission(getContext(), p);
            if (result != PackageManager.PERMISSION_GRANTED) {
                listPermissionsNeeded.add(p);
            }
        }
        if (!listPermissionsNeeded.isEmpty()) {
            ActivityCompat.requestPermissions(getActivity(),
listPermissionsNeeded.toArray(new String[listPermissionsNeeded.size()]), 100);
            return false;
        }
        return true;
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String
permissions[], int[] grantResults) {
        if (requestCode == 100) {
            if (grantResults.length > 0
                    && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

            }
            return;
        }
    }
    private void showPictureDialog(){
        takePhotoFromCamera();
    }

    public void choosePhotoFromGallary() {
        Intent galleryIntent = new Intent(Intent.ACTION_PICK,

android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);

        startActivityForResult(galleryIntent, GALLERY);
    }

    private void takePhotoFromCamera() {

        Intent intent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(intent, CAMERA);
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data)
{

        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode == getActivity().RESULT_CANCELED) {
            return;
        }
        if (requestCode == GALLERY) {
            if (data != null) {
                Uri contentURI = data.getData();
                try {
                    Bitmap bitmap =
MediaStore.Images.Media.getBitmap(getContext().getContentResolver(),
contentURI);
                    String path = saveImage(bitmap);
                    Toast.makeText(getContext(), "Image Saved!",
Toast.LENGTH_SHORT).show();

                    Intent i = new Intent(getContext(), UploadActivity.class);
                    i.putStringArrayListExtra("image_uris", image_uris);
                    i.putStringArrayListExtra("al_image_encoded",
al_image_encoded);
                    startActivity(i);

                } catch (IOException e) {
                    e.printStackTrace();
                    Toast.makeText(getContext(), "Failed!",
Toast.LENGTH_SHORT).show();
                }
            }

        } else if (requestCode == CAMERA) {
            Bitmap thumbnail = (Bitmap) data.getExtras().get("data");
            saveImage(thumbnail);
            Toast.makeText(getContext(), "Image Saved!",
Toast.LENGTH_SHORT).show();

            Intent i = new Intent(getContext(), UploadActivity.class);
            i.putStringArrayListExtra("image_uris", image_uris);
            i.putStringArrayListExtra("al_image_encoded", al_image_encoded);
            startActivity(i);

            getActivity().finish();

        }
    }

    public String saveImage(Bitmap myBitmap) {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        myBitmap.compress(Bitmap.CompressFormat.JPEG, 90, baos);

        String imageEncoded = Base64.encodeToString(baos.toByteArray(),
Base64.DEFAULT);
        al_image_encoded.add(imageEncoded);

        File wallpaperDirectory = new File(
                Environment.getExternalStorageDirectory() + IMAGE_DIRECTORY);
        // have the object build the directory structure, if needed.
        if (!wallpaperDirectory.exists()) {
            wallpaperDirectory.mkdirs();
        }

        try {
            File f = new File(wallpaperDirectory, Calendar.getInstance()
                    .getTimeInMillis() + ".jpg");
            f.createNewFile();
            FileOutputStream fo = new FileOutputStream(f);
            fo.write(baos.toByteArray());
            MediaScannerConnection.scanFile(getContext(),
                    new String[]{f.getPath()},
                    new String[]{"image/jpeg"}, null);
            fo.close();
            Log.d("TAG", "File Saved::--->" + f.getAbsolutePath());
            image_uris.add(f.getAbsolutePath());
            //Toast.makeText(getContext(), "Absolute Path: "+
f.getAbsolutePath(), Toast.LENGTH_SHORT).show();

            // rd.notifyDataSetChanged();
            return f.getAbsolutePath();
        } catch (IOException e1) {
            e1.printStackTrace();
        }
        return "";
    }

}
```

CHAPTER 5

CONCLUSION AND FUTURE WORK

We have tried our best to implement this project so that nobody sleeps with empty stomach. We are proposing a future enhancement for this project. Those who are interested in this project are free to develop this project in other platforms .The ultimate aim of those who are future enhancing this project is to provide wide range services at free of cost to the users. At last one thing to keep in mind about this project is that this project is clearly service oriented and not money oriented. Let us all join hands for this common cause.

CHAPTER 6

REFERENCES:

1. Parfitt, J., Barthel, M., & Macnaughton, S. (2010). Food Waste within Food Supply Chains: Quantification and Potential for Change to 2050. Philosophical Transactions of the Royal Society, 2065-3081.

2. Ahumada, O., & Villalobos, J. (2009). Application of Planning Models in the Agri-food Supply Chain: A Review. European Journal of Operational Research, 1-20.

3. Anik, L., Aknin, L., Norton, M., & Dunn, E. (2009). Feeling Good About Giving: The Benefits (and Costs) of SelfInterested Charitable Behaviour. Harvard Business School.

4. Beretta, C., Stoessel, F., Baier, U., & Hellweg, S. (2013). Quantifying Food Losses and The Potential for Reduction. Waste Management , 764-773.

5. Darlington, R., & Rahimifard, S. (2006). A Responsive Demand Management Framework for the Minimization of Waste in Convenience Food Manufacture. International Journal of Computer Integrated Manufacturing, 1-22.

6. FAO. (2013). The State of Food Insecurity in the World. Rome, Italy: Food and Agriculture Organization of the United Nations.