DONATE FOR SMILE

BY

MST. BIPASHA AKTER
ID:172-15-9953

MOST. DILRUBA YASMIN DOLON
ID:172-15-9955

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

Supervised By

Ms. AFSARA TASNEEM MISHA
Lecturer
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

OCTOBER 2020

APPROVAL

This Project/internship "**DONATE FOR SMILE**", submitted by **ID: 172- 15-9953, Most. Dilruba Yasmin Dolon, ID: 172-15-9955** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on

.

BOARD OF EXAMINERS

Dr. Syed AkhterHossain **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Sadik

Md. Sadekur Rahman **Internal Examiner**
AssistantProfessor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Subhenur Latif **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Saddam

Dr. Md. Saddam Hossain **External Examiner**
Assistant Professor
Department of Computer Science and Engineering
United International University

DECLARATION

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

Supervised by:

Ms. AfsaraTasneemMisha
Lecturer
Department of CSE
Daffodil International University

Submitted by:

Bipasha

Mst. BipashaAkter
ID: -172-15-9953
Department of CSE
Daffodil International University

MOST. Dilruba Yasmin Dolon

Most.DilrubaYasminDolon
ID: -172-15-9955
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

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

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

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

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of
our parents.

ABSTRACT

The purpose behind this project is to present an idea of reduce food wastage and help people. Food wastage is a big problem and we need to solve this problem. Every day there are huge amount of food goes to waste. In Bangladesh there are so many poor people suffer from starvation. Every day there are millions of children and adults who do not get food they need to help. We work to get contending food from restaurants, party, manufactures and retailers to people in when they need. At the same time we seek to help the people. This app is for social work. This apps feature is login, registration, donated food receiving. This app is very helpful for poor and helpless people and easy to use. You can donate easily through this app.

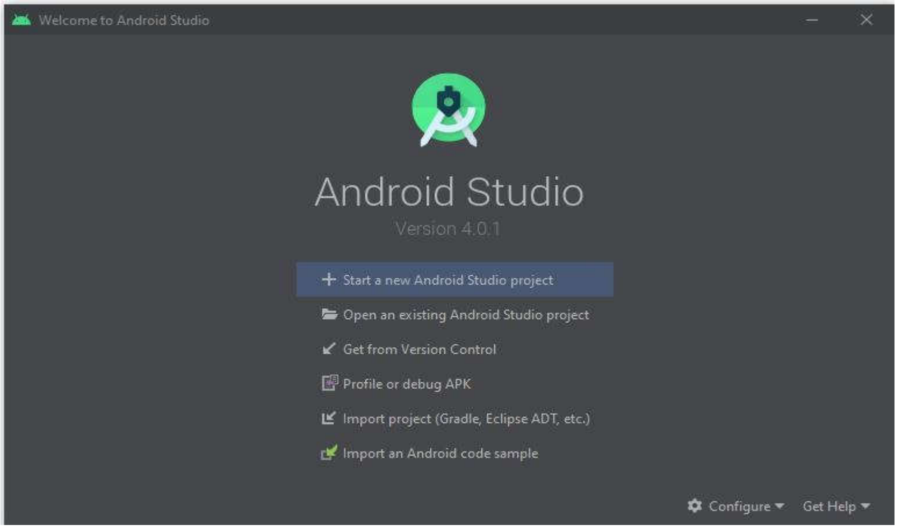
CHAPTER

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Introduction

Bangladesh is a small country. There are huge amount of food is thrown away as waste in daily life. The waste of food is approximates 5.5% of the amount of food. Food waste is a big disturbing issue in a poor country like Bangladesh. Canteens, Restaurants, Marriages, Family and social and programs expel out so much food.



Welcome to Android Studio
Android Studio
Version 4.0.1
+ Start a new Android Studio project
Open an existing Android Studio project
Get from Version Control
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Import an Android code sample
Configure
Get Help

Motivation

Now day a huge amount of meals waste in the restaurants, hotels and houses. For all of those problems we make an android app where donor can easily donate foo, money blood and also clothes. And receiver can easily receive the food.

By using this app as a donor the restaurants and programs can be informed as a volunteer can be get information that donor want to donate for helping.

Objectives:

- Easy touse.
- People can easily donate necessary things to helplesspeople.
- Helpless and poor people are getting meals, money clothes and othersthings.
- Easy to supplyfood.
- By using this app we can reduce foodwastage.

ExpectedOutcome

After all hard work we will get the app which is very easy and useful for all kind of people. It will help those who need things most. By this app we can provide help poor women, children and old people and we will reduce wastagefood.

ProjectQuestions

- What kind of data we will be using to complete ourapp?
- How to prepare the applayout?
- Finding out the algorithm behind the analysis of thesystem?

- How to compare the method with other methods in the samefields?
- Who will be the user for theproject?
- How can we become Fortunate to use theapp?

CHAPTER 2
BACKGROUND

Introduction

Bangladesh is a country that growing as a digital country. Many people are now digital. Our networking system every people use android phone and internet. People want too many things in a short time. They want everything smartly so they use social side. Nowadays every people use smart phone and internet. So this app is helpful for every type of people and they use is easily. This app is easy to use and helps to save your time and food wastage.

Relatedwork

Amp Your Good (United States)

There are many charity android apps but our app is people helping and social working app.

Caritas (Vatican City)

There is lots of donation app in Bangladesh so we are trying our best to reach with people and help them in an easy way. So we can help needy and interested people.

We made this app for all kind of people who really need help. It is a food donation app and also other thins donating app. We connect with helpless people through this app. So they can easily eat food. We can reduce number of poor and helpless people. We can make them happy and satisfy by using this androidapp.

Comparativesstudies

There are many social working apps we have. Some apps work for donating people or some apps works for helpless people. We use many apps but there areno this kind of apps feature that can helps people in a very easy way. So this app is very useful for people who want to donate and also who needs to food and necessarythings.

Scope theproblem

The concept of the app is very satisfied. We tried to make better concept app. When we make this app we faced lots of problems. The apps gives us many errors. We always try to make this app unique concept and unique feature. First time when we run this app not worked properly. Then it takes much time to fixerrors.

Challenges

Challenges that we face

- We faced many problems to complete this project and we try ourbest to complete this appsuccessfully.
- We learn how to fix errors and complete an app and how toovercome formproblems.
- Now tow kind of people can use this app. One who wants to donateand another who needfood.

CHAPTER 3

REQUIREMENTS SPECIFICATIONS

Introduction

The project contains the model for “DONATE FOR SIMILE” including all the models. This model helps to the viewer to understand the project all the models. This model helps to the viewer to understand the project easily all the models.

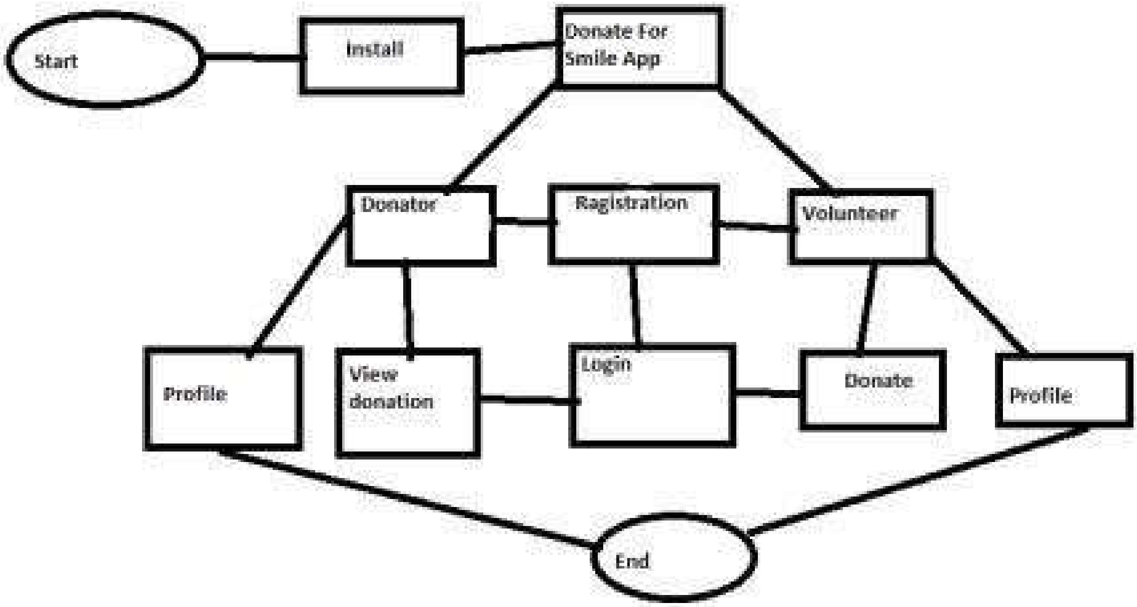


Fig 3.1: Donate For Smile App Model.

This business process model shows apps features like download to install process. It helps to know about the app easily. After seeing tis model we can see how to use this app and how it works.

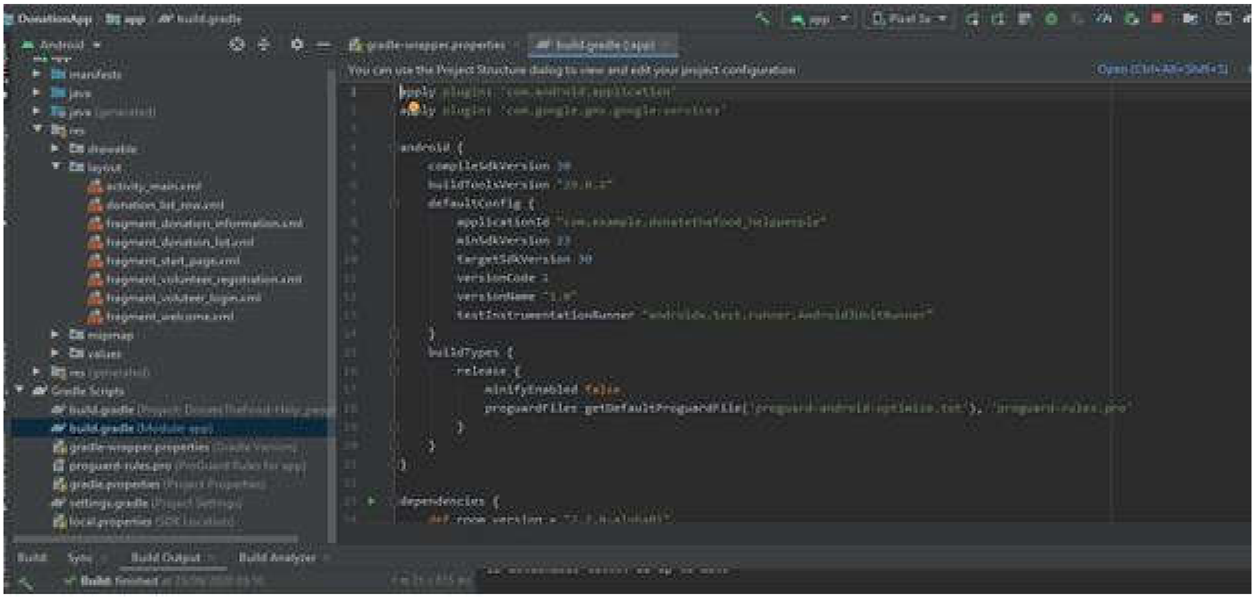
Requirementcollection

This android studio project has much kind of requirements. This is important to identify project model design and structure so that we can fulfil apps requirements. We worked step by step and make sure there is no problem in the app. We collect lots of information from internet and work our project base of that information. After that we can able to provide a better performance.

.

Build Settings

After the player Settings, we need to control the settings where we are building this application. This setting is also known as Build Settings.



DonationApp
app
build.gradle
Android
manifests
java
java (generated)
res
drawable
layout
activity_main.xml
donation_list_row.xml
fragment_donation_information.xml
fragment_donation_list.xml
fragment_start_page.xml
fragment_volunteer_registration.xml
fragment_volunteer_login.xml
fragment_welcome.xml
mipmap
values
res (generated)
Gradle Scripts
build.gradle (Project: DonateTheFood-Help_people)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
gradle-wrapper.properties
build.gradle (:app)
app
Pixel 3a
Build
Sync
Build Output
Build Analyzer
Build: finished

You can use the Project Structure dialog to view and edit your project configuration

```
apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'

android {
    compileSdkVersion 30
    buildToolsVersion "29.0.2"
    defaultConfig {
        applicationId "com.example.donatethefood_helppeople"
        minSdkVersion 23
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    def room_version = "2.3.0-alpha01"
```

Fig 3.2.1: Build settings of the project.

The following figure represents that we have built the app in Android. As we can see there we use build gradle version 30.

.

Use case modelling

We know that we worked our project step by step. In software engineering all we know to describe a project we use case diagram in our use case diagram we can see the process of out app. First of all donor have to registration for donate then donor can login and post about what he wants to donate. After that he can see what he has donated. Also Volunteer can see what donor has donated and distributethen poorpeople.

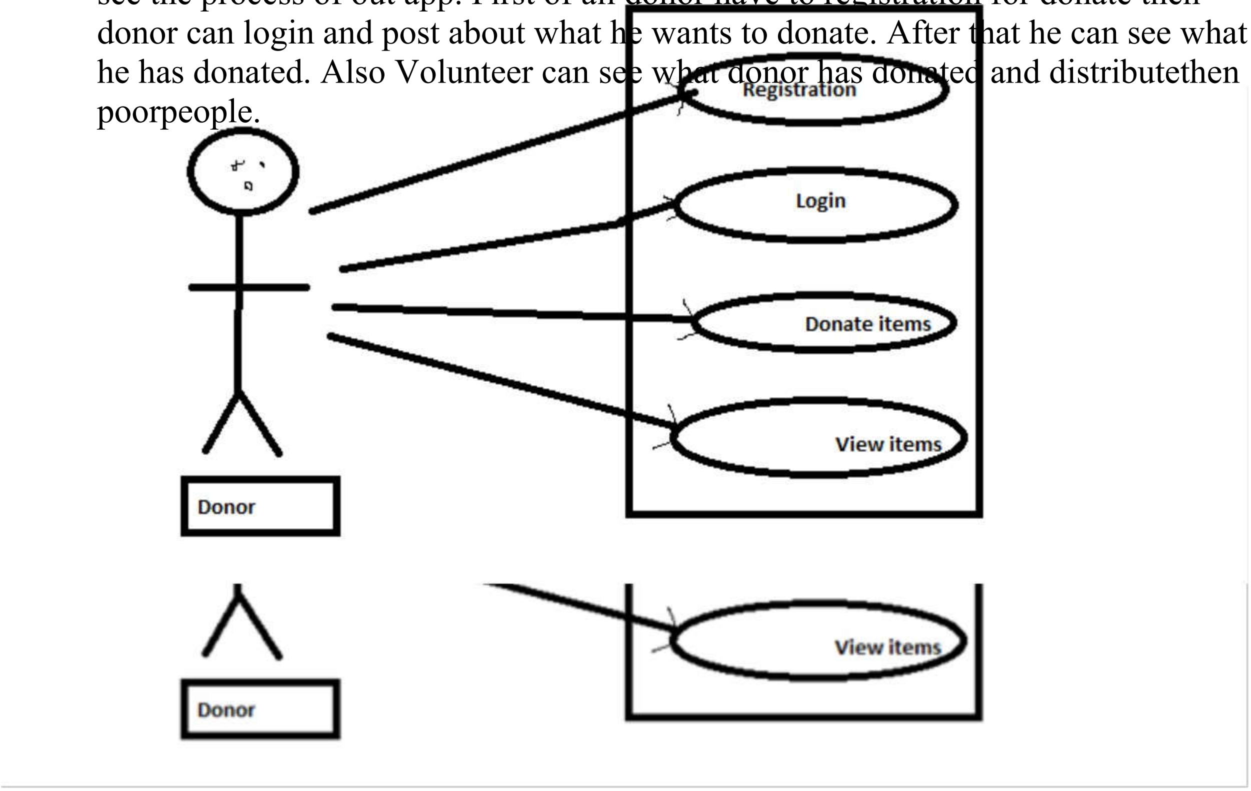


Fig 3.2.2: Use case diagram of donor

Volunteer

In this use case diagram where we use our apps process step by step. firstly volunteer needs to login the app. If volunteer is not complete registration then he/she needs to complete-

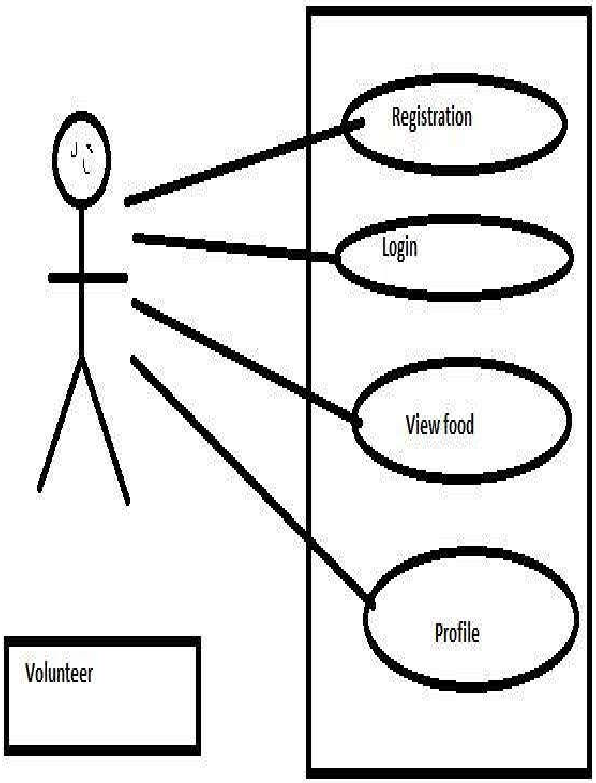


Fig 3.2.3: Use case diagra volunteer

Registration and then he can login thepage.
In this diagram we can see app feature are user friendly. Users all information stored into database. After register into app donors information will store in database also receivers information will store into database. Name, address, phone number, email etc. using this app users information will keep secure andsafe.

Design requirement

Donator

Volunteer

Donation posting

View post

View donation

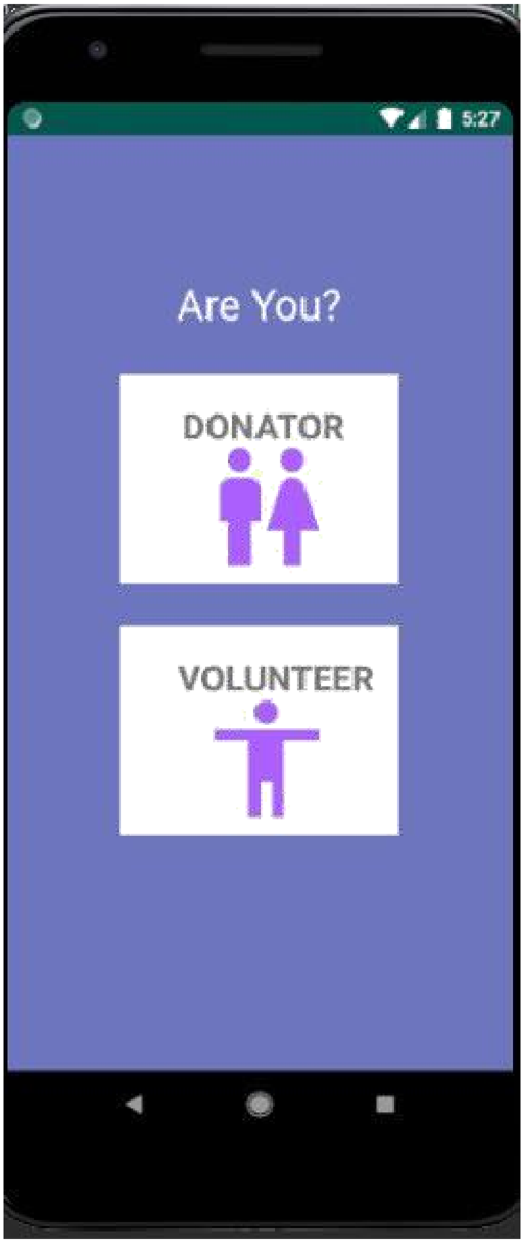
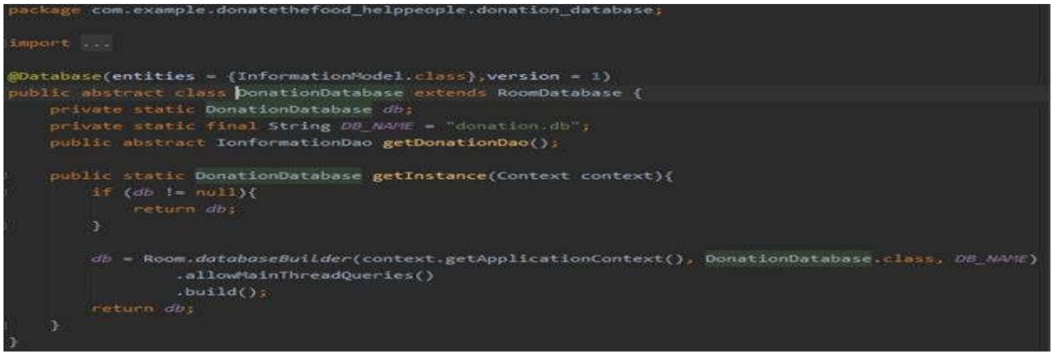


Fig 3.2.3: DonatorVolunteer

Database Implementation

Here we can see our apps database implementation code. The version we used in this app is ‘1’. We can see donation database extends room database.



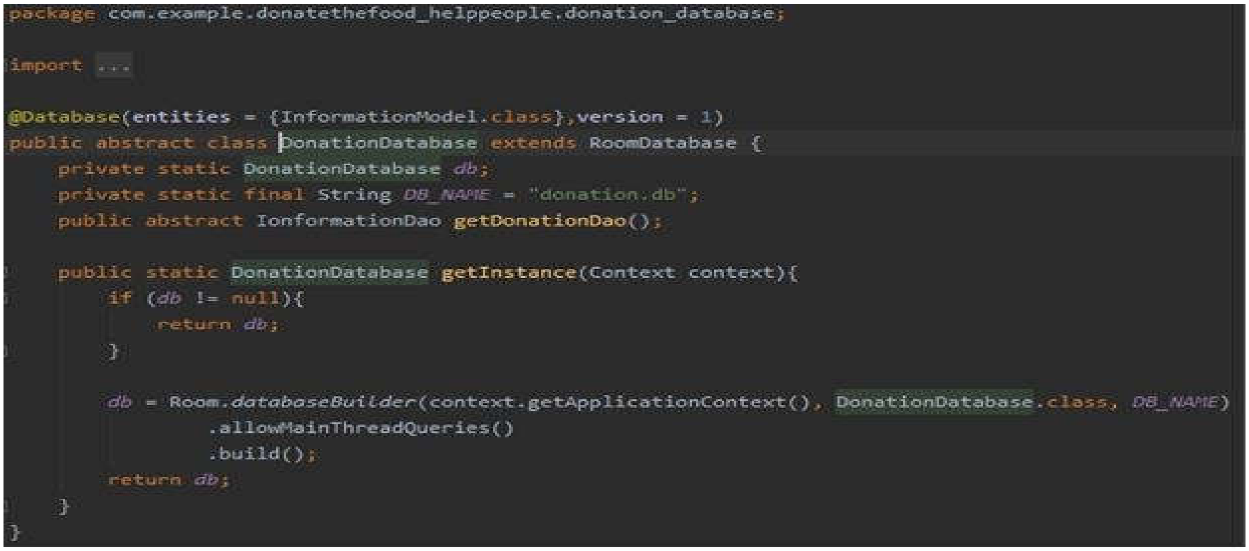
```
package com.example.donatethefood_helppeople.donation_database;

import ...

@Database(entities = {InformationModel.class},version = 1)
public abstract class DonationDatabase extends RoomDatabase {
    private static DonationDatabase db;
    private static final String DB_NAME = "donation.db";
    public abstract IonformationDao getDonationDao();

    public static DonationDatabase getInstance(Context context){
        if (db != null){
            return db;
        }

        db = Room.databaseBuilder(context.getApplicationContext(), DonationDatabase.class, DB_NAME)
                .allowMainThreadQueries()
                .build();
        return db;
    }
}
```

Fig 3.2.4: Codes for database of objects.

Here a function db has been called in the code. With this function users of this app will be able to process the objects.

Database object

As we are working with an object so it is very important that object can fix where we need to store our data and how to secure it.



```
package com.example.donatethefood_helppeople.donation_database;

import ...

@Database(entities = {InformationModel.class},version = 1)
public abstract class DonationDatabase extends RoomDatabase {
    private static DonationDatabase db;
    private static final String DB_NAME = "donation.db";
    public abstract IonformationDao getDonationDao();

    public static DonationDatabase getInstance(Context context){
        if (db != null){
            return db;
        }

        db = Room.databaseBuilder(context.getApplicationContext(), DonationDatabase.class, DB_NAME)
                .allowMainThreadQueries()
                .build();
        return db;
    }
}
```

Fig 3.2.5: Code for Database object.

Database Platform

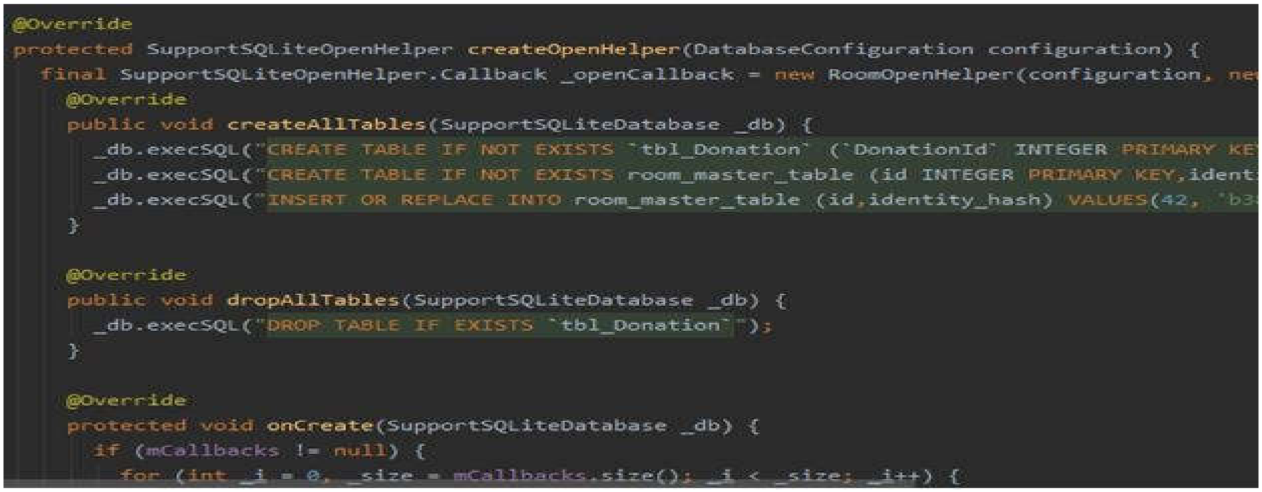
Firebase Database

Firebase platform is developed by Google for helping and creating mobile application.

forconfig our app we use google-service.json. we use the individual instructions for each module: FirebaseUIAuth, Firebase Fire store, FirebaseUI database, FirebaseUIStroge.

Database Interface Information

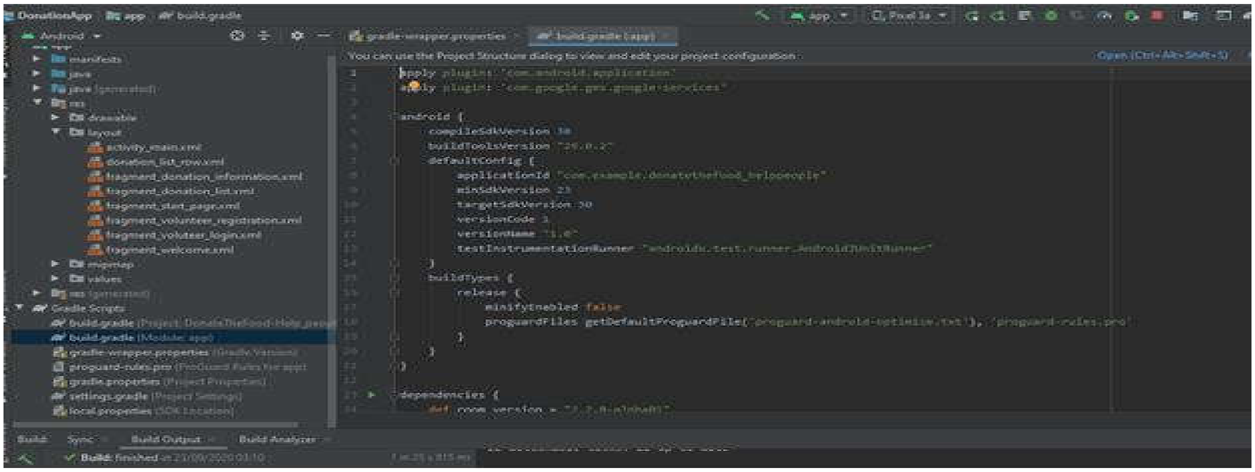
Here we can see code of database interface information code. Each FirebaseUIlivrary has a transitive dependency on the appropriate Firebase SDK.



```
@Override
protected SupportSQLiteOpenHelper createOpenHelper(DatabaseConfiguration configuration) {
  final SupportSQLiteOpenHelper.Callback _openCallback = new RoomOpenHelper(configuration, ne
    @Override
    public void createAllTables(SupportSQLiteDatabase _db) {
      _db.execSQL("CREATE TABLE IF NOT EXISTS `tbl_Donation` (`DonationId` INTEGER PRIMARY KE
      _db.execSQL("CREATE TABLE IF NOT EXISTS room_master_table (id INTEGER PRIMARY KEY,ident
      _db.execSQL("INSERT OR REPLACE INTO room_master_table (id,identity_hash) VALUES(42, 'b3
    }

    @Override
    public void dropAllTables(SupportSQLiteDatabase _db) {
      _db.execSQL("DROP TABLE IF EXISTS `tbl_Donation`");
    }

    @Override
    protected void onCreate(SupportSQLiteDatabase _db) {
      if (mCallbacks != null) {
        for (int _i = 0, _size = mCallbacks.size(); _i < _size; _i++) {
```

Fig 3.2.6: Database Interface Information code

The code declared db-execSQL. For implementation we use ‘com.firebase-ui-database: 6.3.0’ ,firestore : 6.3.0


DonationApp
app
build.gradle
Android
manifests
java
java (generated)
res
drawable
layout
activity_main.xml
donation_list_row.xml
fragment_donation_information.xml
fragment_donation_list.xml
fragment_start_page.xml
fragment_volunteer_registration.xml
fragment_volutear_login.xml
fragment_welcome.xml
mipmap
values
res (generated)
Gradle Scripts
build.gradle (Project: DonateTheFood-Help_peopl
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
gradle-wrapper.properties
build.gradle (app)
app
Pixel 3a
Build
Sync
Build Output
Build Analyzer
Build: finished at 21/09/2020 03:10
Open (Ctrl+Alt+Shift+S)

You can use the Project Structure dialog to view and edit your project configuration

```
apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'

android {
    compileSdkVersion 30
    buildToolsVersion "29.0.2"
    defaultConfig {
        applicationId "com.example.donatethefood_helppeople"
        minSdkVersion 23
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    def room_version = "2.3.0-alpha01"
```

Fig 3.2.7: Implementation of database.

Here is the following figure we Written a code for Implementation of database.

Button Design

Here you can see our apps login registration button design layout. For design we use many activity like activity main, donation list, fragment donation, fragment start page etc.

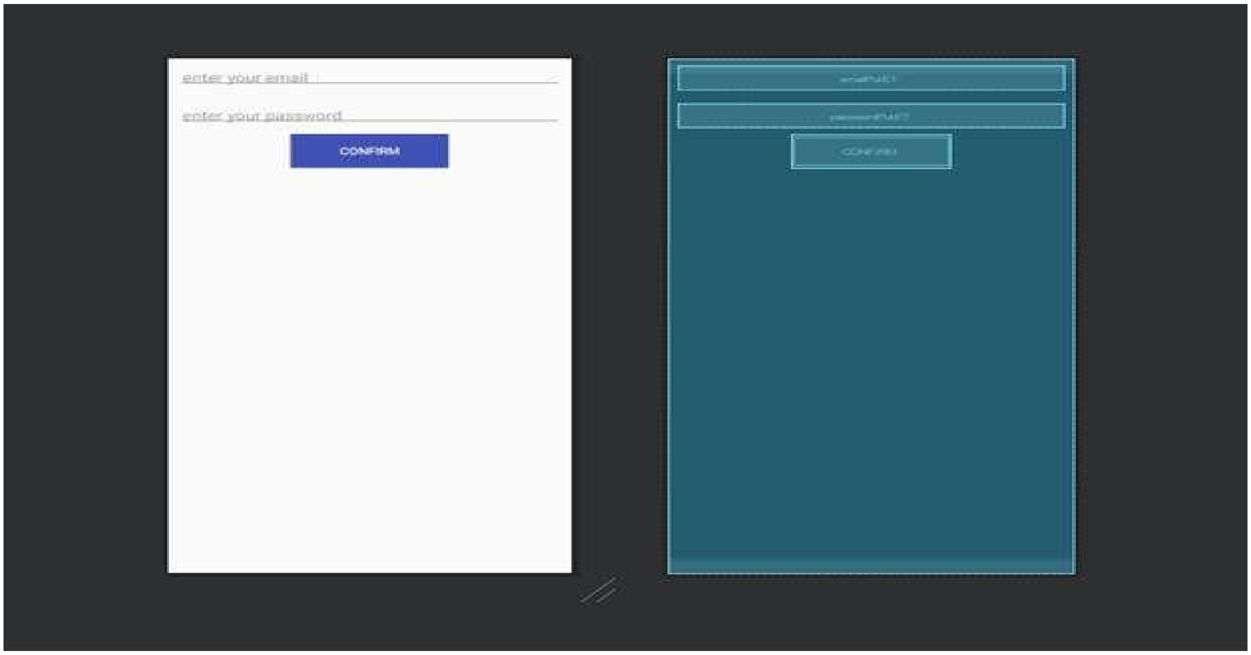
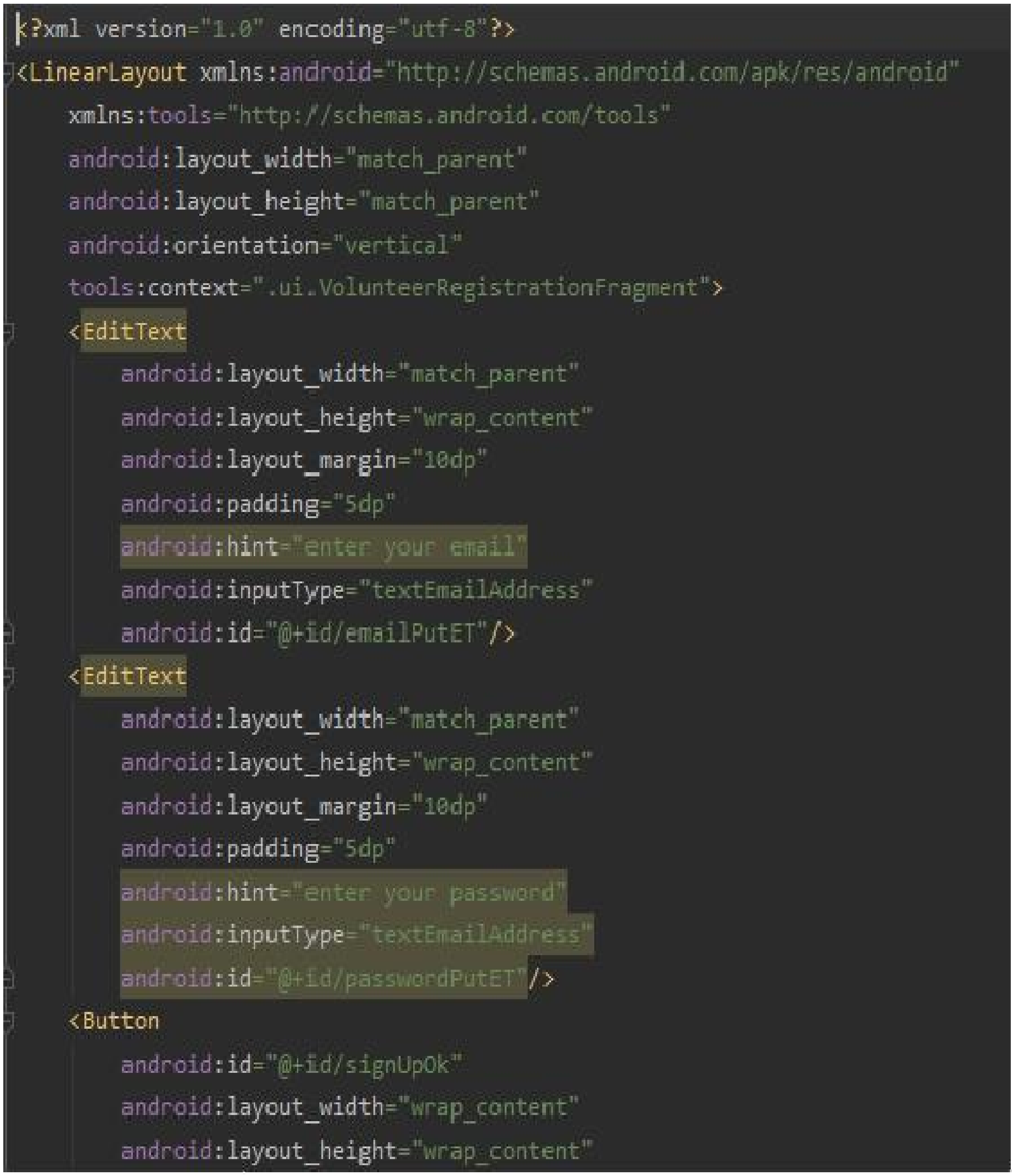


Fig 3.2.8: Button design.

Here is the main page we use for button. We design it very easily that is the advantage.

We have done our design first. Next we see our design code. It shows how to create a design perfectly. Match parent contains the whole design. We use for layout is linear layout, relative layout, image view text view etc.



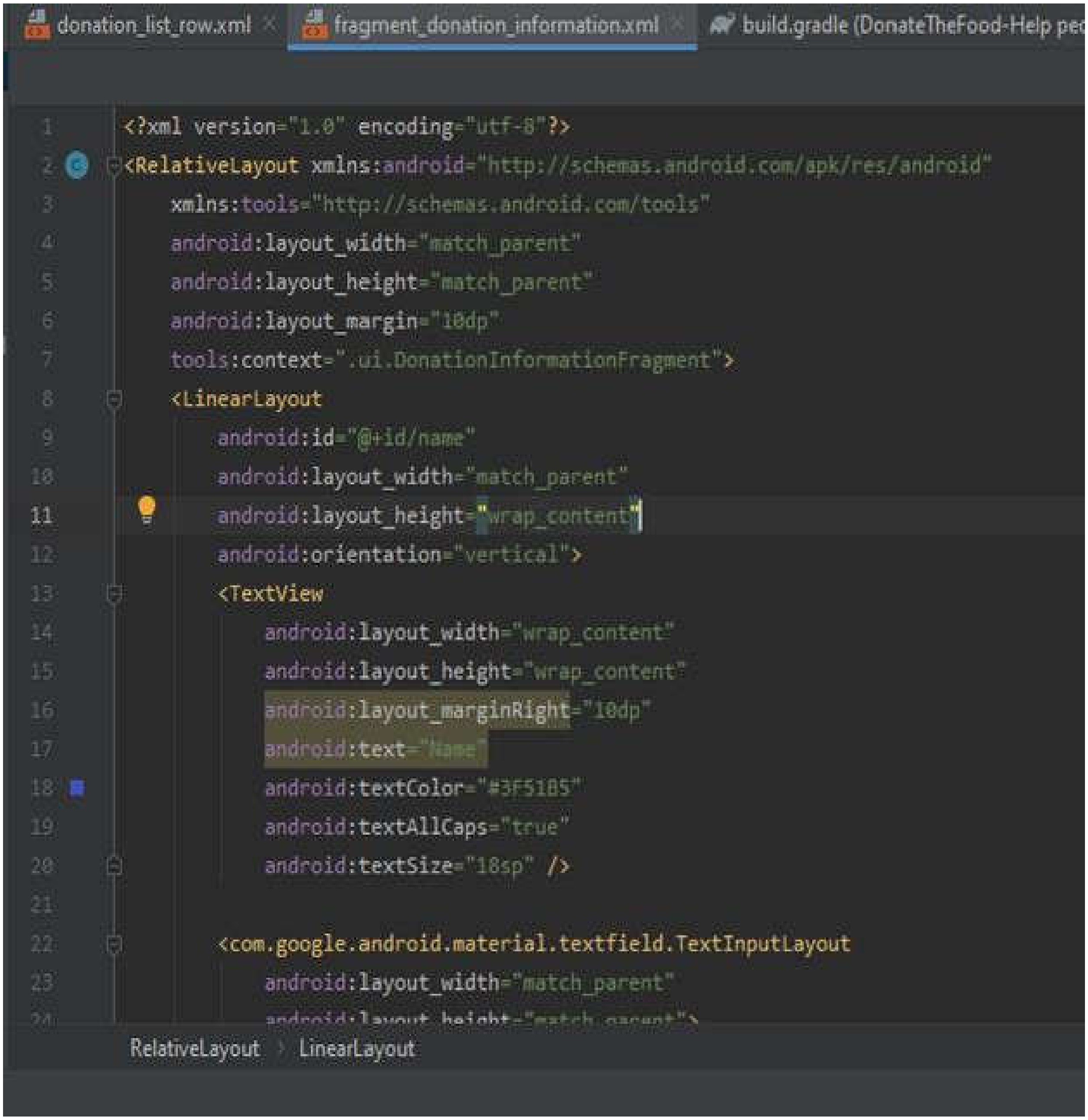
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".ui.VolunteerRegistrationFragment">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:padding="5dp"
        android:hint="enter your email"
        android:inputType="textEmailAddress"
        android:id="@+id/emailPutET"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:padding="5dp"
        android:hint="enter your password"
        android:inputType="textEmailAddress"
        android:id="@+id/passwordPutET"/>
    <Button
        android:id="@+id/signUpOk"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

Fig 3.2.9: Button code.

We can see in our figure that how we fix our apps width and height by using wrap content. Layout margin fix 10dp.

Fragment donation

We can see fragment donation code xml version is 1.0 and utf is 8. Android id in linear layout is @+id/name and orientation

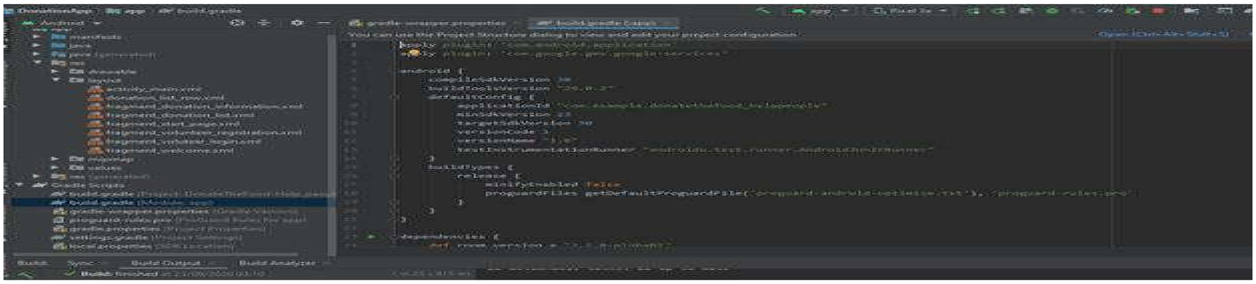

donation_list_row.xml
fragment_donation_information.xml
build.gradle (DonateTheFood-Help pe
RelativeLayout
LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    tools:context=".ui.DonationInformationFragment">
    <LinearLayout
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginRight="10dp"
            android:text="Name"
            android:textColor="#3F51B5"
            android:textAllCaps="true"
            android:textSize="18sp" />

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">
```

Fig 3.2.110: Code for fragment donation.

Build Gradle Version

There are some common materials are the project for almost every code. our apps build gradle version is 30.


DonationApp
app
build.gradle
Android
manifests
java
java (generated)
res
drawable
layout
activity_main.xml
donation_list_row.xml
fragment_donation_information.xml
fragment_donation_list.xml
fragment_start_page.xml
fragment_volunteer_registration.xml
fragment_volunteer_login.xml
fragment_welcome.xml
mipmap
values
res (generated)
Gradle Scripts
build.gradle (Project: DonateTheFood-Help...)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
gradle-wrapper.properties
build.gradle (:app)
Build
Sync
Build Output
Build Analyzer
Build: finished

You can use the Project Structure dialog to view and edit your project configuration

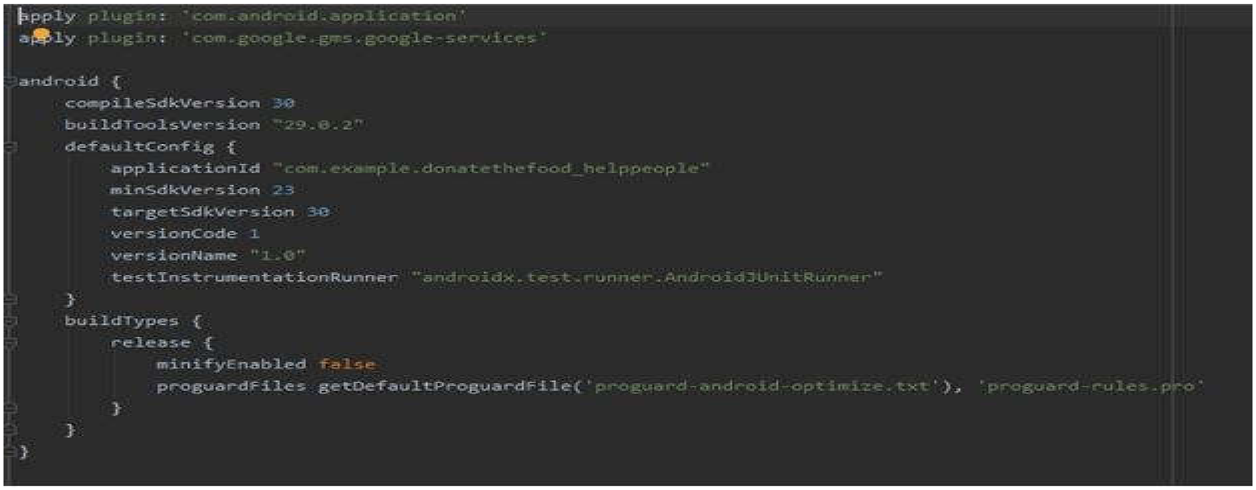
```
apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'

android {
    compileSdkVersion 30
    buildToolsVersion "29.0.2"
    defaultConfig {
        applicationId "com.example.donatethefood_helppeople"
        minSdkVersion 23
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    def room_version = "[illegible]"
```

Fig 3.2.11: Build Gradle Version code.

Here the following figure, there we can see app gradle versions. We can control latest version by editing version number. We use 2 dependencies here one is com.android.tools.build:gradle:4.0.1 and another is com.google.gms:google-services:4.3.2

Build Module App

As we can see we fix our module version and that is. Build tool version is 29.0.2, version code is 1and SDK version is 23.



```
apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'

android {
    compileSdkVersion 30
    buildToolsVersion "29.0.2"
    defaultConfig {
        applicationId "com.example.donatethefood_helppeople"
        minSdkVersion 23
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}
```

Fig 3.2.12: Build Module App.

Without version control we cannot build and run project. So it is important to control versions.

Donation List Activity

We design our donation list activity to see what donator is donated. Also we can see the picture of donated food, donated item name, quantity, date and address.

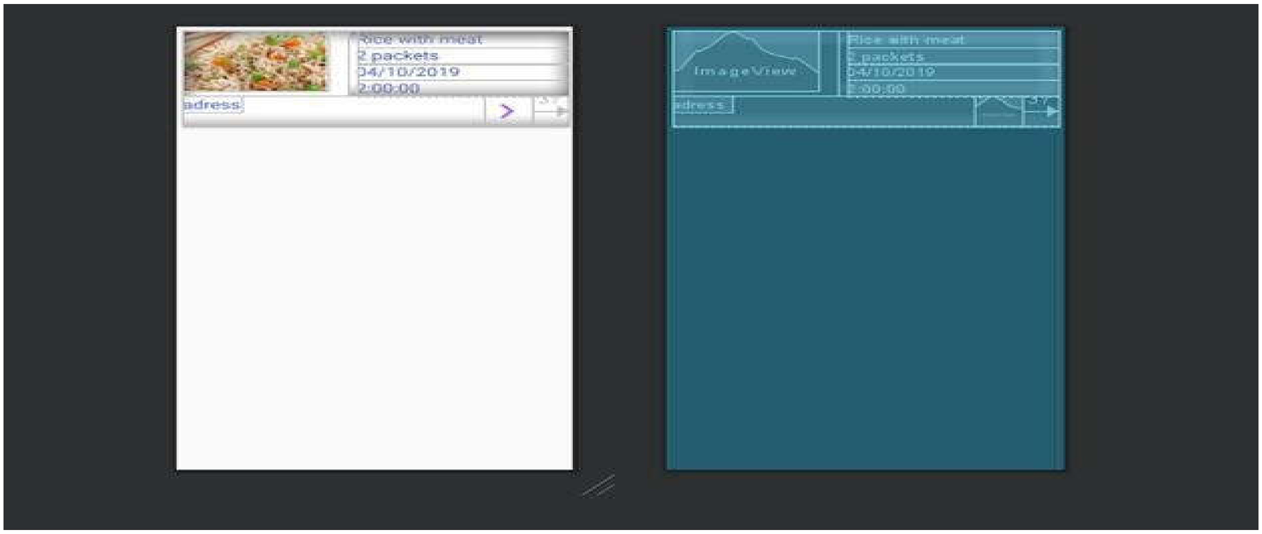
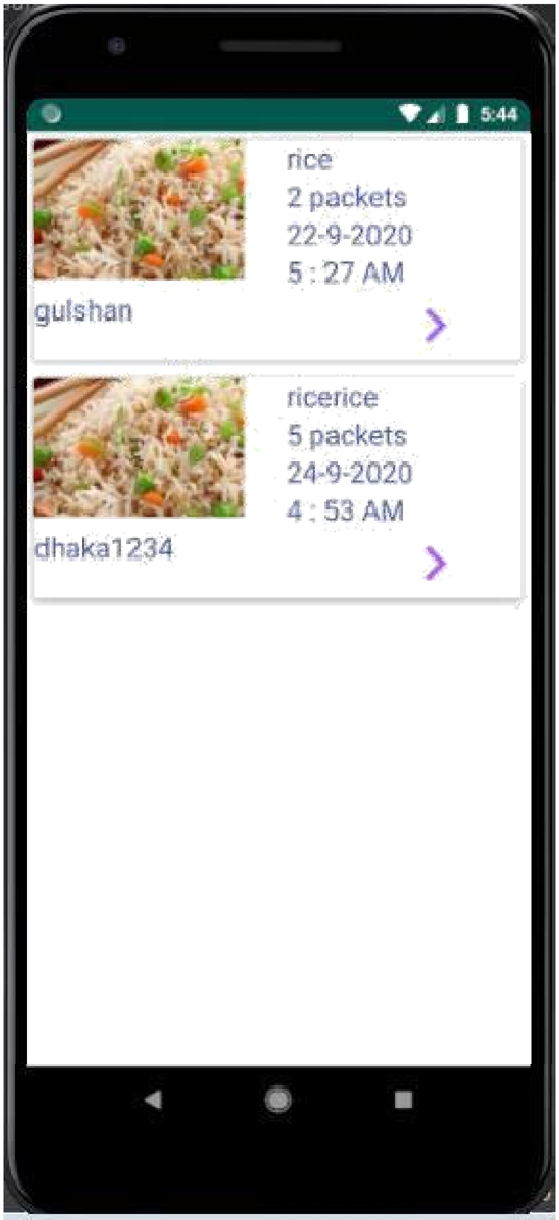


Fig 3.2.13: Donation List Activity Design.

After seeing this volunteer collect that item and distribute into poor and helpless people. We design it for reason of volunteer that he can easily see the list.



5:44

rice
2 packets
22-9-2020
5 : 27 AM
gulshan

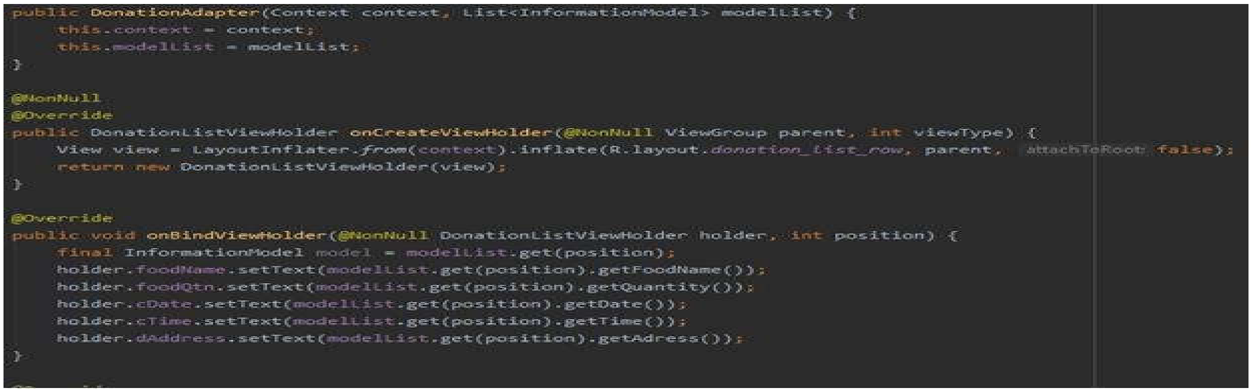
ricerice
5 packets
24-9-2020
4 : 53 AM
dhaka1234

Fig 3.2.14: List of donated food.

Whenever donator donates foods it recorded. After login app volunteer can see the list of donated items and distribute them quickly.

Donation adapter

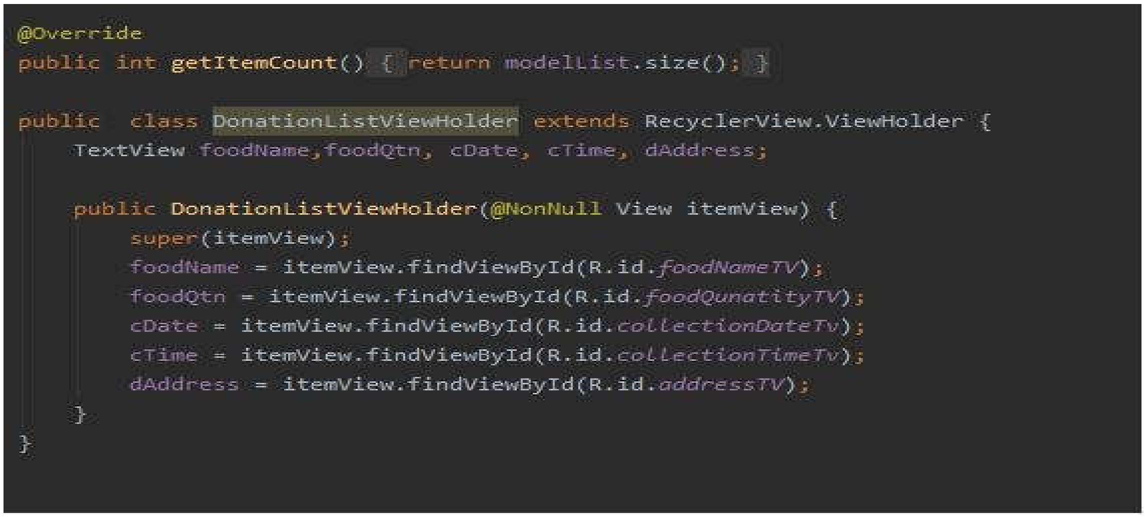
We use adapter in our code to see donated items. Usually adapter is use for only this reason. Here we can see donation adapter extends recycle view.



```
public DonationAdapter(Context context, List<InformationModel> modelList) {
    this.context = context;
    this.modelList = modelList;
}

@NonNull
@Override
public DonationListViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(context).inflate(R.layout.donation_list_row, parent,  attachToRoot: false);
    return new DonationListViewHolder(view);
}

@Override
public void onBindViewHolder(@NonNull DonationListViewHolder holder, int position) {
    final InformationModel model = modelList.get(position);
    holder.foodName.setText(modelList.get(position).getFoodName());
    holder.foodQtn.setText(modelList.get(position).getQuantity());
    holder.cDate.setText(modelList.get(position).getDate());
    holder.cTime.setText(modelList.get(position).getTime());
    holder.dAddress.setText(modelList.get(position).getAdress());
}
```

Fig 3.2.15: Donation adapter code.

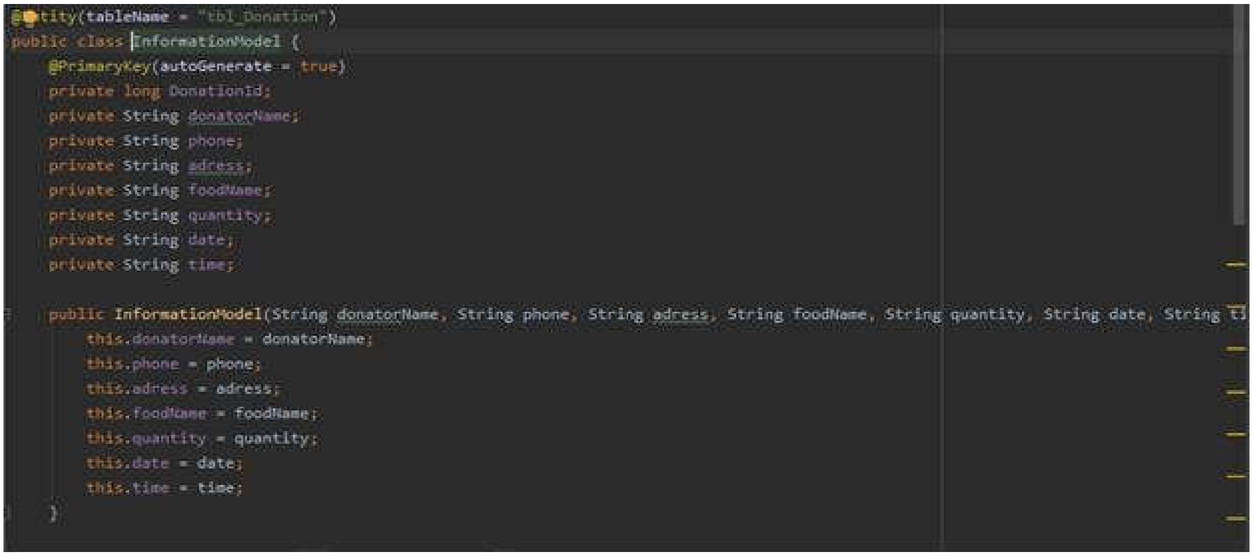
Here we can see donation list view code where item views are food name, foodqtn, cdate, ctime and caddress. Donation list holder extends recycleview.



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

public  class DonationListViewHolder extends RecyclerView.ViewHolder {
    TextView foodName,foodQtn, cDate, cTime, dAddress;

    public DonationListViewHolder(@NonNull View itemView) {
        super(itemView);
        foodName = itemView.findViewById(R.id.foodNameTV);
        foodQtn = itemView.findViewById(R.id.foodQunatityTV);
        cDate = itemView.findViewById(R.id.collectionDateTv);
        cTime = itemView.findViewById(R.id.collectionTimeTv);
        dAddress = itemView.findViewById(R.id.addressTV);
    }
}
```

Fig 3.2.16: Donation List View Holder code.

To see following figure we can understand which reason why we use list view and item view.



```
@tity(tableName = "tbl_Donation")
public class InformationModel {
    @PrimaryKey(autoGenerate = true)
    private long DonationId;
    private String donatorName;
    private String phone;
    private String adress;
    private String foodName;
    private String quantity;
    private String date;
    private String time;

    public InformationModel(String donatorName, String phone, String adress, String foodName, String quantity, String date, String ti
        this.donatorName = donatorName;
        this.phone = phone;
        this.adress = adress;
        this.foodName = foodName;
        this.quantity = quantity;
        this.date = date;
        this.time = time;
    }
```

Fig 3.2.17: Information model Code

Challenges

We face huge challenges in this project. We face problem in coding part. We faced a lot of problem like-

- It takes lots of time to find out where is theerror.
- As we no android studio gradle version updates very quickly,for gradle version control we couldn’t sync our projecteasily.
- Android studio software takes huge space like RAM. For this reasonsome time it takes much time to load and build theproject.
- It is very hard to find out problem and error and fixit.
- There is huge library in android studio for that reason weface someproblem.

CHAPTER 4
DESIGN SPECIFICATIONS

Introduction

In this chapter we will discuss about our design and final outcomes. We can see here our Front end design and Back end design. To create this project we use tow language. One is XML and another is Java. XML is for app designing.

Front-endDesign

Here we discuss about Front end design of our app. We can see figure that there are two user options. One is Donator and another is Volunteer. Those who wants to donate something will choose donator part and who will distribute the food will chooseVolunteer.

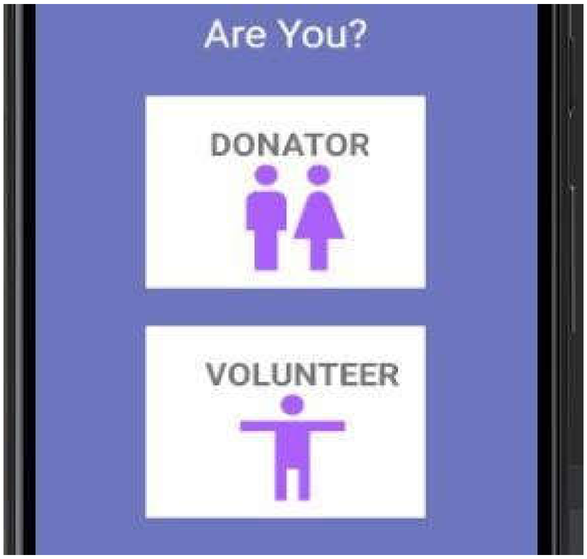
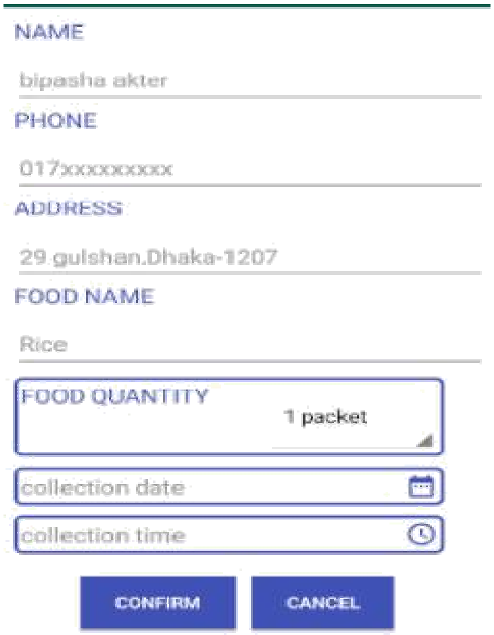


Fig 4.2.1: Donator and Volunteer page



NAME

bipasha akter

PHONE

017xxxxxxxx

ADDRESS

29 gulshan.Dhaka-1207

FOOD NAME

Rice

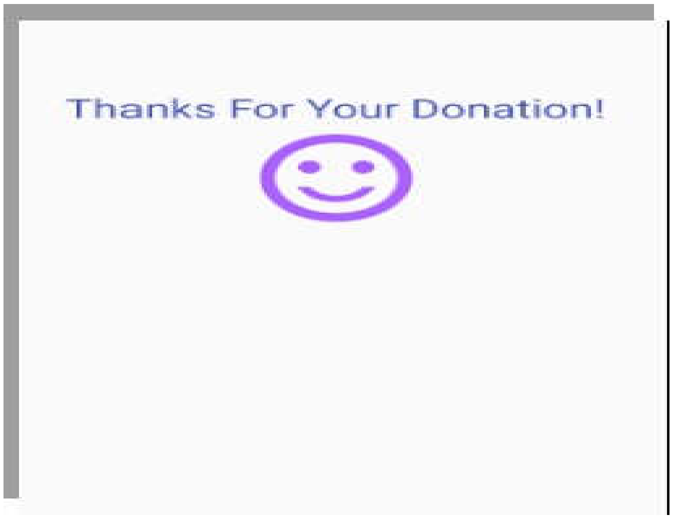
FOOD QUANTITY 1 packet

collection date

collection time

Fig 4.2.2: Donation process.

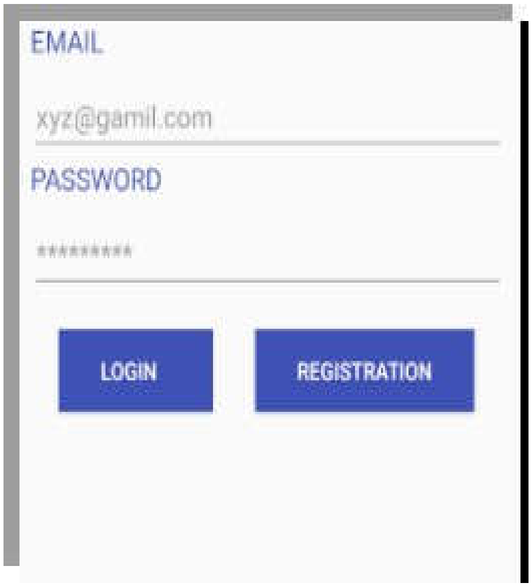
When user choose donation process he/she will see the page like this and he needs to provide name, address, phone number, food name, quantity, date and time.after that user needs to confirm donation.



Thanks For Your Donation!

Fig 4.2.3: Notification page.

After donation food user will get a notification. The notification is like ‘Thanks for your donation’



EMAIL

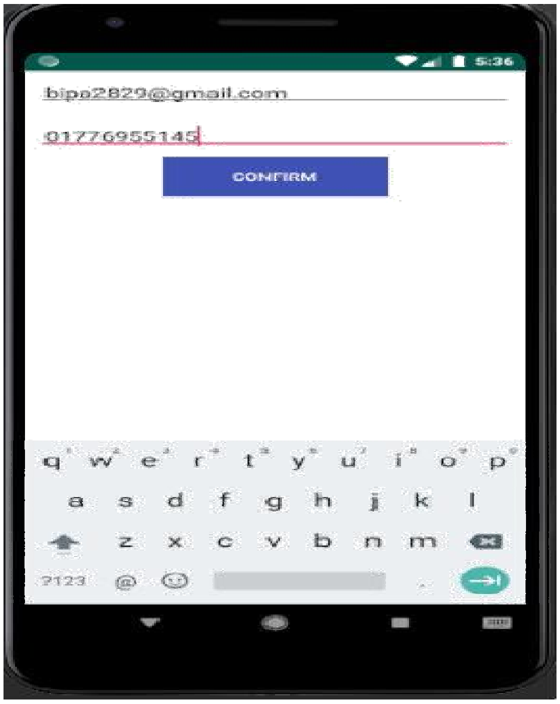
xyz@gamil.com

PASSWORD

****

Fig 4.2.3: Login Page.

To see what donator is donated volunteer needs to login first. After login volunteer can see number of donation and list of items.

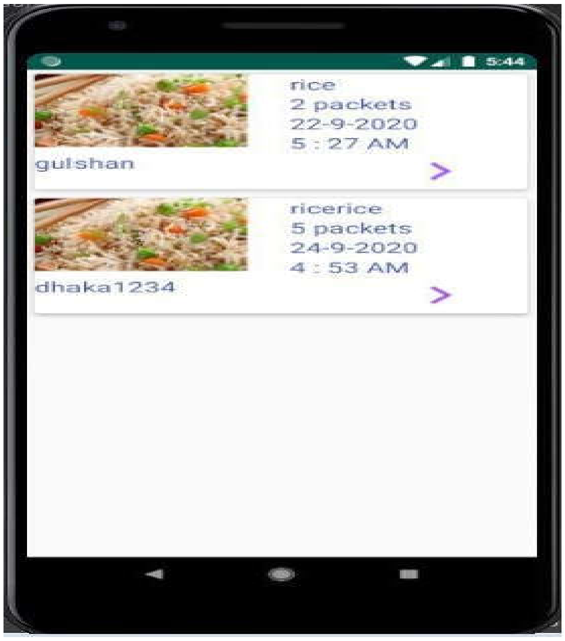


bipa2829@gmail.com

01776955145

Fig 4.2.4: Registration page

If user cannot go to login then user need to complete registration first after that user can login the page. To registration user need to provide user email and user password.



rice
2 packets
22-9-2020
5 : 27 AM
gulshan

ricerice
5 packets
24-9-2020
4 : 53 AM
dhaka1234

Fig 4.2.5: View Post Page

After login user van see how many people donate food and what the items are. Volunteer will see donation details and pick them and distribute them to needy helpless people.

Back-end Design

Back end is not visible for user. In the app.The back end service for the user. It works in the background of the app. Give service to the user. By using android studio we can build easily back end design.

Discussion

First we need to develop the Front end design for the app. We need some tools like android studio and java development kit, Programming language, Firebase database. In future we add more flexible features for user. We want to add cloth donate option money donate option and much more.

CHAPTER 5

CONCLUSION AND FURURE APP

CONCLUSION

We tried to our best to finish this app successfully. First we only started it by login page and then we add more features. After that now we can view post, we can login, we can donate. We added flexible feature so that user can use it very easily. This app is very useful now. Now we can donate things easily by using this app.

Future scope

An android app can be developing more and more day by day. So we have no limitation for develop the app. As a developer we will make this app more flexible and user friendly. We will add more features like blood donation, cloth donation, money donation and other things. We shall try to add these features in this app.

Reference

Varsha Jain (2016) “An Automated Food Wastage Tracking System for Dormitory Student's Mess”, International Conference on Internet of Things and Applications (IOTA)#

Swapnil R. Rajput, MohdSohelDeshmukh, Karbhari V. Kale, “Cross-platform Smartphone Emergency Reporting Application in Urban Areas using GIS Location based and Google Web Services”, International Journal of Computer Applications, Volume 130 – No.12, November 2015.

Hitesh V. Raut, Swapnil R. Rajput, Dhananjay B. Nalawade, Karbhari V. Kale,‘‘ SMARTPHONE BASED WASTE FOOD SUPPLY CHAIN FOR AURANGABAD CITY USING GIS LOCATION BASED AND GOOGLE WEB SERVICES”,International Journal of Research in Engineering and Technology, Volume: 05 Issue: 04 | Apr-2016.

Dels Aliments Bank Foundation.(2011). FoodDonation |.Retrieved July 09, 2014.

Food Banks Canada.(2013). Food Banking in Canada. Retrieved from http://www.foodbankscanada.ca/Learn-About-Hunger/Food-Banking-in-Canada.aspx

www.rescuingleftovercuisine.org/

PLAGIARISM REPORT

ORIGINALITY REPORT

11%	2%	0%	10%
SIMILARITYINDEX	INTERNETSOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	9
2	Submitted to University of South Florida Student Paper	1
3	Billy Grados, Hector Bedon. "Chapter 12 Software Components of an IoT Monitoring Platform in Google Cloud Platform: A Descriptive Research and an Architectural Proposal", Springer Science and Business Media LLC, 2020 Publication	<1