

# Food Delivery App

**A Project Report for Industrial Internship/Major Project/Minor Project**

***Submitted by***

Indronil Chatterjee, Debanjan Roy, Sayan Dirghangi,  
Dipankar Guha, Avishek Chakraborty

***In the partial fulfillment for the award of the degree of***

**B.Tech**

*In the*

CSE

**Sur Institute of Technology**



at

**Ardent Computech Pvt. Ltd.**



## **Ardent® Computech Pvt. Ltd.**

Module – 132, Ground Floor, SDF Building,  
Sector V, Salt Lake, Kolkata – 700091.

+91 33 40073507

✉ [training@ardentcollaborations.com](mailto:training@ardentcollaborations.com)

### **CERTIFICATE FROM SUPERVISOR**

This is to certify that **Debanjan Roy** have successfully completed the project titled **Food Delivery App** under my supervision during the period from **17<sup>th</sup> May 2021** to **24<sup>th</sup> July 2021** which is in partial fulfilment of requirements for the award of the **B.Tech** degree and submitted to the Department of **CSE** of **Sur Institute of Technology**.

\_\_\_\_\_  
Signature of the Supervisor

**Date:**

Name of the Project Supervisor: **PALLABI SAH**



## **Acknowledgement**

The achievement that is associated with the successful completion of any task would be incomplete without mentioning the names of those people whose endless cooperation made it possible. Their constant guidance and encouragement made all our efforts successful.

We take this opportunity to express our deep gratitude towards our project mentor, *Ms. PALLABI SAHA* for giving such valuable suggestions, guidance and encouragement during the development of this project work.

Last but not the least we are grateful to all the faculty members of Ardent Computech Pvt. Ltd. for their support.

## Chapter 1. INTRODUCTION

Amidst the 21st century, Android is quite possibly the most moving theme in the present innovative industry. For being an open-source and free stage, Android partakes in the trust of a considerable client base whose number is expanding step by step. Android PDAs clients like the similarity, standard free application dispatches, and refreshes, which makes it the best desire to challenge the iPhone's invasion, which now orders an enormous base of utilizations.

The account of Android traces back to 2003 when some American software engineers Andy. Rubin, Rich Miner, Nick Sears, and Chris White helped establish a beginning up Android Inc. in Palo Alto, California. The organization was subsequently confronted with the inadequacy of assets, bringing the global tech goliath Google into the image. Google could detect the potential the item conveyed inside and did what needs to be done worth \$50 Million to gain Android in 2005. Before long, every one of the four Co-authors moved to the Googleplex to keep on fostering the OS further under their new proprietors. The primary public Android Beta Version 1.0 was at last distributed on fifth November 2007.

Android is an open-source working framework dependent on an adjusted variant of the Linux bit and another open-source programming, planned basically for touchscreen cell phones, for example, cell phones and tablets. Its source code is Android Open-Source Project (AOSP), which is essentially authorized under the Apache License. Android has been the top-of-the-line OS worldwide on cell phones since 2011 and tablets since 2013. As of May 2017, it has more than two billion month-to-month dynamic clients, the enormous introduced base of any working framework. As of January 2021, the Google Play Store highlights more than 3 million applications. The main business form, Android 1.0(API level 1), was delivered on September 23, 2008, and the current stable adaptation is Android 11(API level 30), born on September 8, 2020 [1].

Since Android is open source and unreservedly accessible to makers for customization, there are no fixed equipment or programming setups. Furthermore, since the day it was dispatched, Android has been accessibly liberated from cost, and Google clarified that it would be accessible in the future. The OS grabbed the eye of producers worldwide, and numerous at first embraced it for minimal expense advanced mobile phones. Android upholds numerous sound/video/still media designs like MP3, MIDI, WebM, H.263, AAC, MPEG-4. Most Android gadgets incorporate microSD card openings and peruse microSD cards arranged with

the FAT32, Ext3, or Ext4 document frameworks. Android upholds various dialects. Performing multiple uses, with the special treatment of memory allotment, is additionally accessible in Android.

Android has consistently been on the highest point of the hustle when engineers ponder versatile application improvement. An enormous number of Android App Development Companies expect to get high advantages to the extent of \$6.3 trillion going before 2021. This shows Android Mobile Apps will be the trendiest applications later on. Fortuitously all the Android Mobile Apps will have the best UI and better exercises.

## **1.1 OBJECTIVE**

The primary goal of the Online Food Ordering System is to deal with the subtleties of Item Category, Food, Delivery Address, Order, Shopping Cart. It deals with all the data about Item Category, Customer, Shopping Cart, Item Category. The undertaking is completely worked at the regulatory end, and subsequently, just the director has ensured the entrance. The motivation behind the project is to construct an application program to decrease the manual work for dealing with the Item Category, Food, Customer, Delivery Address. It tracks every one of the insights concerning the Delivery Address, Order, Shopping Cart.

## **1.2 SCOPE**

The platform and current demand for food allow many fresh apps to the food market, also apps that are available hits success. Food is a most required thing, this on-demand food delivery app provides delicious foods people ordered, without making them skip a meal.

## **Chapter 2. Study**

### **2.1 Feasibility Study**

At the present moment, the system is entirely functional, save the few minor bugs which are bound to present themselves during more extensive testing. A user is currently able to register and log in to the website and place an order. That order is then displayed, correctly and completely, and all details are stored in google firebase database.

## 2.4 Functional requirements

- **User registration:** User must be able to register for the application through a email id. On installing the application, user must be first register their email and set password.
- **Ordering Food:** User can easily select any food item from the search option and order it after login to the system.
- **User Profile:** User Can see their details by clicking profile option and change their profile picture if they want to.

## 2.6 Hardware and Software Requirement

### 2.6.1 From Developers Perspective

To develop the application the developers require some essential software and hardware, they are listed below

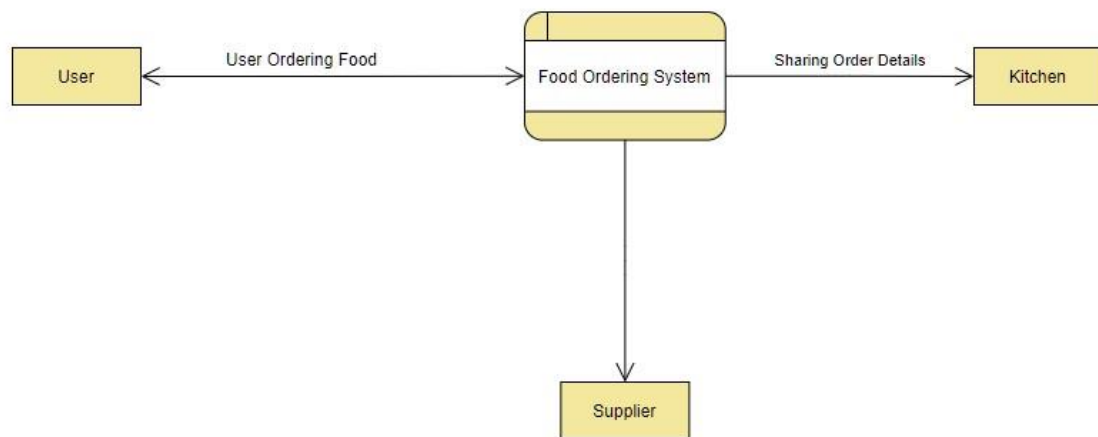
- i. A working laptop or desktop (with minimum 4GB RAM, recommended 8GB or above)
- ii. An Android phone.
- iii. JDK version 8 or above
- iv. Android Studio Software
- v. A Google account

### 2.6.2 From Users Perspective

For using this application the users need an android phone with API version 29 or above and internet connection. By having these the users can register in the application and use it.

## Chapter 3. System Design

### 3.1 Data Flow Diagram (Level 0)



## Chapter 4. Coding and Implementation

### 4.1 Codes

#### 4.1.1 MainActivity/login.java

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

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

import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.material.textfield.TextInputEditText;
import com.google.android.material.textfield.TextInputLayout;
import com.google.firebase.auth.FirebaseAuth;

public class MainActivity extends AppCompatActivity {

    TextInputEditText email_id,password_id;
    TextInputLayout el,pl;
    Button loginbtn;
    ProgressBar p1;
    FirebaseAuth auth;
    String email,pass;
    public static final String EMAIL =
"com.dr.test_project.MainActivity.EMAIL";

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

        email_id = findViewById(R.id.input_email);
        password_id = findViewById(R.id.editTextNumberPassword);
        loginbtn = findViewById(R.id.login);
        el= findViewById(R.id.email_lay);
        pl = findViewById(R.id.passlay);
        p1 = findViewById(R.id.progressBar2);

        loginbtn.setOnClickListener(v -> {
            email = String.valueOf(email_id.getText()).trim();
            pass = String.valueOf(password_id.getText()).trim();
            pl.setError(null);
            el.setError(null);

            if (email.isEmpty()) {
                el.setError("Email Cannot Be Empty");
                return;
            }
            if (pass.isEmpty()) {
                pl.setError("Password Cannot Be Empty");
                return;
            }
            if (pass.length() < 5) {
```



```

        pl.setError("Password Must Be 5 Characters");
        return;
    }

    pl.setVisibility(View.VISIBLE);
    login(email, pass);
    });
}

private void login(String email, String pass) {
    auth = FirebaseAuth.getInstance();

    auth.signInWithEmailAndPassword(email, pass).addOnSuccessListener(authResult
-> {
        Show_msg("Logged in");
        startActivity(new
Intent(MainActivity.this, fooddelivery.class));
        finish();
    }).addOnFailureListener(e -> {
        Log.d("Error:", e.toString());
        Toast.makeText(MainActivity.this, "Error:User Not Found!!!",
Toast.LENGTH_SHORT).show();
        pl.setVisibility(View.INVISIBLE);
    });
}

public void Sign_up (View view)
{
    Show_msg("Please Wait");
    Intent in = new Intent(MainActivity.this, registration.class);
    startActivity(in);
    this.finish();
}

protected void Show_msg (String msg)
{
    Toast.makeText(MainActivity.this, msg, Toast.LENGTH_LONG).show();
}
}

```

#### 4.1.2 MainActivity/Login.java

```

?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity"
android:background="@drawable/new_theme"
android:padding="10dp">

<ImageView
    android:id="@+id/imageView"
    android:layout_width="276dp"
    android:layout_height="219dp"
    android:contentDescription="@string/Login_img"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.496"
    app:layout_constraintStart_toStartOf="parent"

```

```

        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.167"
        app:srcCompat="@drawable/login_1"/>

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="390dp"
    android:layout_height="80dp"
    android:layout_marginTop="36dp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="@+id/imageView"
    app:layout_constraintHorizontal_bias="0.478"
    app:layout_constraintStart_toStartOf="@+id/imageView"
    app:layout_constraintTop_toBottomOf="@+id/imageView">

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/email_lay"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/input_email"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Email ID"
            android:inputType="textEmailAddress"/>
    </com.google.android.material.textfield.TextInputLayout>
</LinearLayout>

<LinearLayout
    android:id="@+id/linearLayout2"
    android:layout_width="390dp"
    android:layout_height="80dp"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="@+id/imageView"
    app:layout_constraintHorizontal_bias="0.478"
    app:layout_constraintStart_toStartOf="@+id/imageView"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout"
    app:layout_constraintVertical_bias="0.089">

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/passlay"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:passwordToggleEnabled="true">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/editTextNumberPassword"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Password"

            android:inputType="numberPassword"/>
    </com.google.android.material.textfield.TextInputLayout>
</LinearLayout>

<Button
    android:id="@+id/login"
    android:layout_width="95dp"
    android:layout_height="48dp"
    android:text="@string/login"
    android:textColor="@color/black"

```

```

        android:background="@drawable/button"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/linearLayout2"
        app:layout_constraintVertical_bias="0.16" />

<LinearLayout
    android:id="@+id/linearLayout3"
    android:layout_width="395dp"
    android:layout_height="39dp"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.444"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/login"
    app:layout_constraintVertical_bias="0.0">

    <TextView
        android:id="@+id/Sign_up"
        android:layout_width="wrap_content"
        android:layout_height="31dp"
        android:layout_marginStart="10dp"
        android:layout_weight="1"
        android:clickable="true"
        android:focusable="true"
        android:fontFamily="monospace"
        android:onClick="Sign_up"
        android:text="@string/to_create_a_new_account"
        android:textColor="@color/blue" />

</LinearLayout>

<ProgressBar
    android:id="@+id/progressBar2"
    style="?android:attr/progressBarStyle"
    android:layout_width="58dp"
    android:layout_height="51dp"
    android:visibility="invisible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.528"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout3"
    app:layout_constraintVertical_bias="0.0" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

### 4.1.3 Registration.java

```

package com.dr.test_project;

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

import android.app.Activity;
import android.content.Context;
import android.content.Intent;

```

```

import android.nfc.Tag;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.textfield.TextInputEditText;
import com.google.android.material.textfield.TextInputLayout;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.storage.FirebaseStorage;

import org.jetbrains.annotations.NotNull;
import org.w3c.dom.Text;

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

public class registration extends AppCompatActivity {

    TextInputLayout email_id,password,confirm_pass,name_lay;
    TextInputEditText email_input,pass_input,confirm_pass_input,name_inp;
    Button signup;
    FirebaseAuth auth;
    FirebaseFirestore data_fire;
    String email,pass_in,con_pass,uid,name;

    protected void showing(String msg)
    {
        Toast.makeText(registration.this, msg, Toast.LENGTH_SHORT).show();
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);
        email_id = findViewById(R.id.new_email);
        password = findViewById(R.id.new_password);
        confirm_pass = findViewById(R.id.confirm);
        email_input = findViewById(R.id.new_email_input);
        pass_input = findViewById(R.id.new_password_input);
        confirm_pass_input = findViewById(R.id.confirm_input);
        signup = findViewById(R.id.signup_btn);
        name_inp= findViewById(R.id.name_layout_input);
        name_lay = findViewById(R.id.name_layout);

        signup.setOnClickListener(v -> {

            email = String.valueOf(email_input.getText()).trim();
            pass_in = String.valueOf(pass_input.getText()).trim();
            con_pass = String.valueOf(confirm_pass_input.getText()).trim();
            name = String.valueOf(name_inp.getText()).trim();
            email_input.setError(null);
            pass_input.setError(null);
            name_lay.setError(null);
            confirm_pass_input.setError(null);
            if (name.isEmpty()) {
                name_lay.setError("Name cannot be empty");
            }
        });
    }
}

```

```

        return;
    }
    if (email.isEmpty()) {
        email_input.setError("Email Cannot Be Empty");
        return;
    }
    if (pass_in.isEmpty()) {
        pass_input.setError("Password Cannot Be Empty");
        return;
    }
    if (pass_in.length() < 5) {
        pass_input.setError("Password Must Be 5 Characters");
        return;
    }
    if (con_pass.equals(pass_in) == false) {
        confirm_pass_input.setError("Not Equal To Your New
Password");
        return;
    }

    auth = FirebaseAuth.getInstance(); //getting the uid
    data_fire = FirebaseFirestore.getInstance();
    //stored the email and pass in my fire database
    auth.createUserWithEmailAndPassword(email,
pass_in).addOnCompleteListener(task -> {
        if(task.isSuccessful())
        {
            Toast.makeText(registration.this, "Account
Created", Toast.LENGTH_SHORT).show();
            uid = auth.getCurrentUser().getUid();
            DocumentReference documentReference =
data_fire.collection("New_user").document(uid);
            Map<String, Object> user = new HashMap<>(); // adding
database to the fire store
            user.put("Name", name);
            user.put("E-mail", email);

            documentReference.set(user).addOnCompleteListener(task1 -> {
                if(task1.isSuccessful())
                {
                    Log.d("TAG", "User profile successfully
created"+ uid);
                }
            }).addOnFailureListener(e -> {
                Log.d("TAG", "Failure!!! Error: "+e.toString());
            });
            startActivity(new
Intent(getApplicationContext(), MainActivity.class));
        }
        else
        {
            Toast.makeText(registration.this, "Error:" +
task.getException().getLocalizedMessage(), Toast.LENGTH_SHORT).show();
        }
    });
}

public void login_page(View view)
{
    Intent in = new Intent(registration.this, MainActivity.class);
    startActivity(in);
    this.finish();
}

```

```
}
```

#### 4.1.4 Registration.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".registration"
android:background="@drawable/theme_darkmode">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="20dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.157"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0">

        <TextView
            android:id="@+id/welcome"
            android:layout_width="129dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="44dp"
            android:fontFamily="@font/amita_bold"
            android:text="@string/welcome"
            android:textColor="@color/black"
            android:textSize="25sp" />

        <TextView
            android:id="@+id/info"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginVertical="10dp"
            android:text="@string/for_registration_please_fill_up_the_following_details"
            android:textColor="@color/black"
            android:textSize="15sp" />

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:id="@+id/name_layout"
            android:layout_height="match_parent"
            android:layout_marginVertical="50dp"
            >

            <com.google.android.material.textfield.TextInputEditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/name_layout_input"
                android:hint="Name"
                />

            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/new_email"
                android:layout_width="match_parent"
```

```

        android:layout_height="match_parent"
        android:layout_marginTop="25dp"
        app:startIconDrawable="@drawable/ic_baseline_email_24">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/new_email_input"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/email_id_2"
            android:inputType="textEmailAddress" />

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/new_password"
            android:layout_width="match_parent"
            android:layout_height="349dp"
            android:layout_marginVertical="5dp"
            android:paddingVertical="25dp"
            app:passwordToggleEnabled="true"

app:startIconDrawable="@drawable/ic_baseline_vpn_lock_24">

<com.google.android.material.textfield.TextInputEditText
            android:id="@+id/new_password_input"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/new_password"
            android:inputType="numberPassword" />

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/confirm"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginVertical="5dp"
            android:paddingVertical="25dp"

app:startIconDrawable="@drawable/ic_baseline_vpn_lock_24">

<com.google.android.material.textfield.TextInputEditText
            android:id="@+id/confirm_input"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/confirm_password"
            android:inputType="numberPassword" />

        <Button
            android:id="@+id/signup_btn"
            android:layout_marginTop="10dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:background="@drawable/button"
            android:text="@string/create"
            android:textColor="@color/black" />

        <TextView
            android:id="@+id/back_login"
            android:layout_width="match_parent"
            android:layout_height="35dp"
            android:layout_gravity="end"
            android:layout_marginVertical="10dp"
            android:clickable="true"

```

```

        android:focusable="true"
        android:fontFamily="sans-serif"
        android:gravity="left"
        android:onClick="login_page"
        android:text="Already have an account"
        android:textSize="15sp"
        android:textColor="@color/black" />

</com.google.android.material.textfield.TextInputLayout>
    </com.google.android.material.textfield.TextInputLayout>

    </com.google.android.material.textfield.TextInputLayout>
</com.google.android.material.textfield.TextInputLayout>

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

```

#### 4.1.5 Fooddelivery.java

```

package com.dr.test_project;

import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.view.Menu;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.material.snackbar.Snackbar;
import com.google.android.material.navigation.NavigationView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AlertDialog;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.appcompat.app.AppCompatActivity;

import com.dr.test_project.databinding.ActivityFooddeliveryBinding;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.EventListener;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.FirebaseFirestoreException;

public class fooddelivery extends AppCompatActivity {

    private AppBarConfiguration mAppBarConfiguration;
    private ActivityFooddeliveryBinding binding;
    FirebaseAuth auth;
    FirebaseFirestore db;
    String uid;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityFooddeliveryBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
    }
}

```



```

        setSupportActionBar(binding.appBarFooddelivery.toolbar1);
        binding.appBarFooddelivery.toolbar1.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
    }
});
DrawerLayout drawer = binding.drawerLayout;
NavigationView navigationView = binding.navView;
// Passing each menu ID as a set of Ids because each
// menu should be considered as top level destinations.
mAppBarConfiguration = new AppBarConfiguration.Builder(
    R.id.nav_home, R.id.nav_Search,
R.id.nav_Order, R.id.nav_Payment, R.id.nav_offers)
    .setDrawerLayout(drawer)
    .build();
NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment_content_fooddelivery);
NavigationUI.setupActionBarWithNavController(this, navController,
mAppBarConfiguration);
NavigationUI.setupWithNavController(navigationView, navController);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is
present.
    getMenuInflater().inflate(R.menu.fooddelivery, menu);
    NavigationView navigationView = (NavigationView)
findViewById(R.id.nav_view);
    View headerView = navigationView.getHeaderView(0);
    TextView navUsername = (TextView)
headerView.findViewById(R.id.useremail_add);
    TextView navUser = (TextView)
headerView.findViewById(R.id.name_dis);
    auth = FirebaseAuth.getInstance();
    db = FirebaseFirestore.getInstance();
    uid = auth.getCurrentUser().getUid();
    DocumentReference documentReference =
db.collection("New_user").document(uid);
    documentReference.addSnapshotListener(this, new
EventListener<DocumentSnapshot>() {
        @Override
        public void onEvent(@Nullable
@org.jetbrains.annotations.Nullable DocumentSnapshot value, @Nullable
@org.jetbrains.annotations.Nullable FirebaseFirestoreException error) {
            navUsername.setText(value.getString("E-mail"));
            navUser.setText(value.getString("Name"));
        }
    });
}

return true;
}

```

```

@Override
public boolean onSupportNavigateUp() {
    NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment_content_fooddelivery);
    return NavigationUI.navigateUp(navController, mAppBarConfiguration)
        || super.onSupportNavigateUp();
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    if(item.getItemId() == R.id.action_logout){
        callLogoutDialog();
    }
    if(item.getItemId()==R.id.u_profile)
    {
        startActivity(new
Intent(this.getApplicationContext(),user_prof.class));
    }
    return super.onOptionsItemSelected(item);
}

private void callLogoutDialog() {
    AlertDialog.Builder alertBuilder = new
AlertDialog.Builder(fooddelivery.this);
    // alertBuilder.setTitle("What");
    alertBuilder.setTitle("What!!!! Are You Sure Want To Logout");
    alertBuilder.setCancelable(false);
    alertBuilder.setPositiveButton("Yes", ((dialog, which) -> {
        this.finish();
        Toast.makeText(fooddelivery.this,"Please Be Back
Again",Toast.LENGTH_SHORT).show();
        auth.signOut();
        finishAffinity();
    }));
    alertBuilder.setNegativeButton("No", (dialog, which) ->
dialog.dismiss());
    alertBuilder.show();
}
}

```

#### 4.1.6 app\_barFooddelivery.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".fooddelivery">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/Theme.AppCompat">
        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar1"
            android:layout_width="match_parent"
            android:layout_height="55dp"
            android:background="@color/silver"
            app:popupTheme="@style/Theme.Test_project.PopupOverlay"

```

```

        android:padding="10dp"/>

    </com.google.android.material.appbar.AppBarLayout>

    <include layout="@layout/content_fooddelivery" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

#### 4.1.7 contentFooddelivery.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/app_bar_fooddelivery">

    <fragment
        android:id="@+id/nav_host_fragment_content_fooddelivery"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation2" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

#### 4.1.8 Nav\_headerFooddelivery.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="@dimen/nav_header_height"
    android:background="@drawable/theme_darkmode"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/name_dis"
        android:text="User-Name"/>

    <TextView
        android:id="@+id/useremail_add"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```
        android:text="@string/abcd_gmail_com"/>
</LinearLayout>
```

#### 4.1.9 Fooddelivery.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/u_profile"
        android:orderInCategory="100"
        android:title="Profile"
        app:showAsAction="never"/>
    <item
        android:id="@+id/action_logout"
        android:orderInCategory="100"
        android:title="Logout"
        app:showAsAction="never" />

</menu>
```

#### 4.1.9 mobile\_navigation.xml

```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:app="http://schemas.android.com/apk/res-auto"
            xmlns:tools="http://schemas.android.com/tools"
            android:id="@+id/mobile_navigation2"
            app:startDestination="@+id/nav_home">

    <fragment
        android:id="@+id/nav_home"
        android:name="com.dr.test_project.ui.home.HomeFragment"
        android:label="@string/menu_home"
        tools:layout="@layout/fragment_home" />

    <fragment
        android:id="@+id/nav_offers"
        android:name="com.dr.test_project.ui.Offers.offer"
        android:label="Address"
        tools:layout="@layout/order_fragment" >
        <action
            android:id="@+id/action_nav_offers_to_nav_Payment"
            app:destination="@id/nav_Payment" />
    </fragment>

    <fragment
        android:id="@+id/nav_Order"
        android:name="com.dr.test_project.ui.order.order"
        android:label="Order"
        tools:layout="@layout/order_fragment" >
        <action
            android:id="@+id/action_nav_Order_to_nav_offers"
            app:destination="@id/nav_offers" />
    </fragment>
```

```

<fragment
    android:id="@+id/nav_Payment"
    android:name="com.dr.test_project.ui.payment.payment"
    android:label="Payment"
    tools:layout="@layout/payment_fragment" />
<fragment
    android:id="@+id/nav_Search"
    android:name="com.dr.test_project.ui.Search.Search"
    android:label="Search"
    tools:layout="@layout/search_fragment" />

</navigation>

```

#### 4.1.10 Home.java

```

package com.dr.test_project.ui.home;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProvider;

import com.dr.test_project.R;
import com.dr.test_project.databinding.FragmentHomeBinding;

public class HomeFragment extends Fragment {

    private HomeViewModel homeViewModel;
    private FragmentHomeBinding binding;

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle
savedInstanceState) {
        homeViewModel =
            new ViewModelProvider(this).get(HomeViewModel.class);

        binding = FragmentHomeBinding.inflate(inflater, container, false);
        View root = binding.getRoot();

        final TextView textView = binding.welcome;
        homeViewModel.getText().observe(getViewLifecycleOwner(), new
Observer<String>() {
            @Override
            public void onChanged(@Nullable String s) {
                textView.setText("Welcome");
            }
        });
        return root;
    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}

```

```
}  
}
```

#### 4.1.10 fragment\_Home.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:app="http://schemas.android.com/apk/res-auto"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
tools:context=".ui.home.HomeFragment"  
android:background="@drawable/tajine">  
  
    <TextView  
        android:id="@+id/wels"  
        android:layout_width="292dp"  
        android:layout_height="114dp"  
        android:text="@string/welcome"  
        android:textColor="@color/black"  
        android:textSize="60sp"  
        android:textStyle="bold"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintHorizontal_bias="0.63"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent"  
        app:layout_constraintVertical_bias="0.299" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>
```

#### 4.1.11 HomeViewModel.java

```
package com.dr.test_project.ui.home;  
  
import androidx.lifecycle.LiveData;  
import androidx.lifecycle.MutableLiveData;  
import androidx.lifecycle.ViewModel;  
  
public class HomeViewModel extends ViewModel {  
  
    private MutableLiveData<String> mText;  
  
    public HomeViewModel() {  
        mText = new MutableLiveData<>();  
        mText.setValue("I Dont know");  
    }  
  
    public LiveData<String> getText() {  
        return mText;  
    }  
  
}
```

#### 4.1.12 Search.java

```
package com.dr.test_project.ui.Search;  
  
import androidx.lifecycle.ViewModelProvider;
```

```

import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import com.dr.test_project.FoodModel;
import com.dr.test_project.adapter.home_adapter;
import com.dr.test_project.databinding.SearchFragmentBinding;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QuerySnapshot;

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

public class Search extends Fragment {

    private SearchViewModel mViewModel;
    private SearchFragmentBinding binding;
    RecyclerView rView;
    LinearLayoutManager layoutManager;
    FirebaseFirestore firestore;
    List<FoodModel> foodModels = new ArrayList<>();
    public static Search newInstance() {
        return new Search();
    }

    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
ViewGroup container,
                           @Nullable Bundle savedInstanceState) {
        mViewModel =
            new ViewModelProvider(this).get(SearchViewModel.class);
        binding = SearchFragmentBinding.inflate(inflater, container,
false);
        View root = binding.getRoot();
        rView = binding.foodList;
        layoutManager = new LinearLayoutManager(getContext());
        rView.setLayoutManager(layoutManager); // setting the layout

        fetchdata();

        return root;
    }

    private void fetchdata() {
        firestore = FirebaseFirestore.getInstance();
        firestore.collection("Food_item")
            .get()
            .addOnCompleteListener(task -> {
                if(task.isSuccessful())
                {
                    QuerySnapshot snapshots = task.getResult();
                    for (DocumentSnapshot doc:

```

```

        snapshots.getDocuments()) {
            FoodModel model = new FoodModel(
                // doc.getId(),
                doc.getString("name"),
                "₹"+doc.get("price"),
                doc.getString("Category"),
                "EDT:"+doc.getString("Est")+"min",
                doc.getString("description"),
                doc.getString("pic")
            );
            foodModels.add(model);
        }
        //now use this list to get the data
        home_adapter adapter = new
home_adapter(foodModels,requireActivity());
        rView.setAdapter(adapter); //to display items
    }
}
.addOnFailureListener(e -> {
    Snackbar.make(requireContext(),binding.getRoot(),"Failed:
"+e.getMessage(),Snackbar.LENGTH_SHORT).show();
});
}

@Override
public void onDestroyView() {
    super.onDestroyView();
    binding = null;
}
}
}

```

#### 4.1.13 fragment\_Search.xml

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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"
    tools:context=".ui.Search.Search"
    android:background="@drawable/kip">

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/food_list"
        android:layout_margin="5dp"/>

</FrameLayout>

```

#### 4.1.14 foodmodel.java

```

package com.dr.test_project;

public class FoodModel {
    // private String id;
    private final String name;
    private final String price;
}

```



```

private final String category;
private final String EST;
private final String description;
private String imgurl;

public FoodModel( String name, String price, String category, String
EST, String description, String imgurl){
    //    this.id=id;
    this.name=name;
    this.category=category;
    this.price=price;
    this.EST=EST;
    this.description=description;
    this.imgurl=imgurl;
}

//    public String getId() {
//        return id;
//    }

public String getImgurl() {
    return imgurl;
}

public String getPrice() {
    return price;
}

public String getCategory() {
    return category;
}

public String getEST() {
    return EST;
}

public String getName() {
    return name;
}

public String getDescription() {
    return description;
}
}

```

#### 4.1.16 Food\_details.java

```

package com.dr.test_project;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import com.dr.test_project.adapter.Constants;
import com.dr.test_project.databinding.ActivityFoodDetailsBinding;
import com.dr.test_project.ui.Search.Search;
import com.dr.test_project.ui.order.order;

```

```

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;
import com.squareup.picasso.Picasso;

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

public class Food_details extends AppCompatActivity {
    ActivityFoodDetailsBinding food_detail_binding;
    ImageView img;
    TextView ti,des,pr,est;
    Button or;
    String name,de,price,etd,im;
    FirebaseAuth auth;
    FirebaseFirestore Fstore;
    String id;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_food_details);
        food_detail_binding =
ActivityFoodDetailsBinding.inflate(getLayoutInflater());
        setContentView(food_detail_binding.getRoot());
        img = food_detail_binding.imageView2;
        ti = food_detail_binding.textTitle;
        des = food_detail_binding.textDes;
        pr = food_detail_binding.textPrice;
        est = food_detail_binding.textEst;
        or = food_detail_binding.button;

        Bundle bundle = getIntent().getBundleExtra("data");

        name = bundle.getString(Constants.Food_title);
        de = bundle.getString(Constants.Food_des);
        price = bundle.getString(Constants.Food_price);
        etd = bundle.getString(Constants.Food_est);
        im = bundle.getString(Constants.Food_img);
        // id = bundle.getString(Constants.Food_id);

        // setting the data get from the database

        ti.setText(name);
        des.setText("Description : \n"+de);
        pr.setText("Price : "+price);
        est.setText(etd);
        Picasso.get().load(im).into(img);

        or.setOnClickListener(v -> {
            //
getSupportFragmentManager().beginTransaction().add(R.id.cont,new
order()).commit();
            auth = FirebaseAuth.getInstance();
            Fstore = FirebaseFirestore.getInstance();
            id = auth.getCurrentUser().getUid();
            DocumentReference documentReference =
Fstore.collection("Order").document(id);
            Map<String,Object> user = new HashMap<>(); // adding database
to the fire store
            user.put("Name",name);
            user.put("Price",price);

```

```

        user.put("img", im);
        user.put("id", id);
        documentReference.set(user).addOnCompleteListener(task1 -> {
            if(task1.isSuccessful())
            {
                Log.d("TAG", "Order Placed Successfully"+ id);
            }
        }).addOnFailureListener(e -> {
            Log.d("TAG", "Failure!!! Error: "+e.toString());
        });
        startActivity(new
Intent(getApplicationContext(), fooddelivery.class));

    });
}
}

```

#### 4.1.17 Food\_details.xml

```

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

    android:background="@drawable/kip2">
    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/cont">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent">
            <androidx.cardview.widget.CardView
                android:layout_width="match_parent"
                android:layout_height="290dp"
                android:layout_margin="12dp"
                app:cardCornerRadius="25dp"

                android:layout_marginBottom="20dp">

            <ImageView
                android:id="@+id/imageView2"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                app:srcCompat="@drawable/order"
                android:cropToPadding="true" />
            </androidx.cardview.widget.CardView>

            <TextView

```

```

        android:id="@+id/text_title"
        android:layout_width="401dp"
        android:layout_height="45dp"
        android:layout_margin="5dp"
        android:padding="5dp"
        android:textSize="20dp"
        android:textColor="@color/white"
        android:text="tile" />

<TextView
    android:id="@+id/text_des"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Des"
    android:textSize="17dp"
    android:layout_margin="5dp"
    android:textColor="@color/white"
    android:padding="5dp"/>

<TextView
    android:id="@+id/text_est"
    android:layout_width="match_parent"
    android:layout_height="30dp"
    android:padding="5dp"
    android:textSize="15dp"
    android:layout_margin="5dp"
    android:text="etd"
    android:textColor="@color/white"/>

<TextView
    android:id="@+id/text_price"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:layout_margin="5dp"
    android:padding="5dp"
    android:text="rs 90"
    android:textColor="@color/white"
    android:textSize="18dp" />

<Button
    android:id="@+id/button"
    android:layout_width="110dp"
    android:layout_height="56dp"
    android:layout_gravity="center horizontal"
    android:layout_marginTop="60dp"
    android:padding="10dp"
    android:text="ADD"
    android:backgroundTint="@color/transparent"
    android:textColor="@color/white" />
</LinearLayout>

</FrameLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

#### 4.1.18 order.java

```

package com.dr.test_project.ui.order;

import androidx.cardview.widget.CardView;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;
import androidx.lifecycle.ViewModelProvider;

```

```

import android.content.Intent;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.navigation.Navigation;

import android.renderscript.Sampler;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import com.dr.test_project.Food_details;
import com.dr.test_project.R;
import com.dr.test_project.adapter.Constants;
import com.dr.test_project.adapter.Order_details;
import com.dr.test_project.databinding.ActivityFoodDetailsBinding;
import com.dr.test_project.databinding.OrderFragmentBinding;
import com.dr.test_project.databinding.SearchFragmentBinding;
import com.dr.test_project.ui.Search.SearchViewModel;
import com.dr.test_project.ui.payment.payment;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.EventListener;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.FirebaseFirestoreException;
import com.squareup.picasso.Picasso;

import org.jetbrains.annotations.NotNull;

import java.util.concurrent.Executor;

public class order extends Fragment {

    private OrderViewModel mViewModel;
    OrderFragmentBinding binding;
    ImageView img;
    TextView ti,pr,tot;
    FirebaseFirestore db;
    FirebaseAuth auth;
    String id;
    Button confirm;
    CardView c;

    public static order newInstance() {
        return new order();
    }

    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
    ViewGroup container,
                                @Nullable Bundle savedInstanceState) {
        mViewModel =
            new ViewModelProvider(this).get(OrderViewModel.class);

```

```

        binding = OrderFragmentBinding.inflate(inflater, container, false);
        View root = binding.getRoot();
        img = binding.imageOrder;
        ti = binding.orderName;
        pr = binding.orderPrice;
        tot = binding.orderTotal;
        confirm = binding.confirmOrder;
        c= binding.cardorder;

        db = FirebaseFirestore.getInstance();
        auth = FirebaseAuth.getInstance();
        id = auth.getUid();
        if(!id.isEmpty())
        {
            Log.d("Id",id);
        }
        DatabaseReference documentReference =
db.collection("Order").document(id);
        db.collection("Order").document(id).addSnapshotListener(new
EventListener<DocumentSnapshot>() {
            @Override
            public void onEvent(@Nullable
@org.jetbrains.annotations.Nullable DocumentSnapshot value, @Nullable
@org.jetbrains.annotations.Nullable FirebaseFirestoreException error) {

                ti.setText(value.getString("Name"));
                pr.setText(value.getString("Price"));
                Picasso.get().load(value.getString("img")).into(img);
                tot.setText("Total: "+value.getString("Price"));

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

Navigation.findNavController(v).navigate(R.id.action_nav_Order_to_nav_offers);

            }
        });

        return root;
    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
}4.1.18 order.xml

```

#### 4.1.19 fragment\_order.xml

```

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

```

```

        android:layout_height="match_parent"
        tools:context=".ui.order.order"
        android:padding="10dp"
        android:background="@drawable/kip2">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="10dp"
            android:orientation="vertical"
            android:background="@color/transparent">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Your Order"
                android:textSize="25dp"
                android:layout_gravity="center"
                android:textStyle="bold"
                android:textColor="@color/light_blue"
                android:layout_margin="20dp">

            </TextView>

            <androidx.cardview.widget.CardView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                app:cardCornerRadius="15dp"
                android:id="@+id/cardorder">

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:orientation="horizontal"
                    android:padding="10dp"
                    android:background="@color/transparent">

                    <ImageView
                        android:id="@+id/image_order"
                        android:layout_width="142dp"
                        android:layout_height="100dp"
                        android:cropToPadding="true"
                        android:src="@drawable/food" />

                    <TextView
                        android:id="@+id/order_name"
                        android:layout_width="144dp"
                        android:layout_height="match_parent"
                        android:text="name"
                        android:textSize="18sp"
                        android:textColor="@color/white"/>

                    <TextView
                        android:id="@+id/order_price"
                        android:layout_width="66dp"
                        android:textColor="@color/white"
                        android:textSize="16sp"
                        android:layout_height="match_parent"
                        android:text="price"

                        android:layout_gravity="right"/>

                </LinearLayout>

```

```

</androidx.cardview.widget.CardView>

<TextView
    android:id="@+id/order_total"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end"
    android:textSize="16sp"
    android:textColor="@color/white"
    android:gravity="end"
    android:padding="15dp"
    android:text="Total" />

<Button
    android:id="@+id/confirm_order"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Pay"
    android:layout_gravity="center"
    android:layout_marginTop="50dp"
    android:textColor="@color/white"/>

</LinearLayout>

</FrameLayout>

```

#### 4.1.20 Address.java

```

package com.dr.test_project.ui.Offers;

import androidx.lifecycle.ViewModelProvider;

import android.content.Intent;
import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.navigation.Navigation;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;

import com.dr.test_project.R;
import com.dr.test_project.databinding.OfferFragmentBinding;
import com.dr.test_project.databinding.OrderFragmentBinding;
import com.dr.test_project.ui.order.OrderViewModel;
import com.dr.test_project.ui.payment.payment;
import com.dr.test_project.user_prof;
import com.google.android.material.snackbar.Snackbar;
import com.google.android.material.textfield.TextInputEditText;
import com.google.android.material.textfield.TextInputLayout;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.FirebaseFirestore;

public class offer extends Fragment {

    private OfferViewModel mViewModel;

```



```

OfferFragmentBinding binding;
FirebaseAuth auth;
FirebaseFirestore database;
TextInputLayout add, pin, lmark, pnun;
TextInputEditText addinp, pininp, lmarkinp, pnuninp;
String sa, sp, sl, sn, uid;
Button b;

public static offer newInstance() {
    return new offer();
}

@Override
public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
ViewGroup container,
                        @Nullable Bundle savedInstanceState) {
    mViewModel = new ViewModelProvider(this).get(OfferViewModel.class);
    mViewModel =
        new ViewModelProvider(this).get(OfferViewModel.class);
    binding = OfferFragmentBinding.inflate(inflater, container, false);
    View root = binding.getRoot();
    b = binding.confirmadd;
    add = binding.addlay;
    pin = binding.pinlay;
    lmark = binding.marklay;
    pnun = binding.pnunlay;
    addinp = binding.addinp;
    pininp = binding.pininp;
    lmarkinp = binding.markinp;
    pnuninp = binding.pnuninp;
    auth = FirebaseAuth.getInstance();
    uid = auth.getCurrentUser().getUid();
    database = FirebaseFirestore.getInstance();
    b.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            sa = String.valueOf(addinp.getText());
            sp = String.valueOf(pininp.getText());
            sl = String.valueOf(lmarkinp.getText());
            sn = String.valueOf(pnuninp.getText());
            add.setError(null);
            pin.setError(null);
            lmark.setError(null);
            pnun.setError(null);
            if (sa.isEmpty())
            {
                add.setError("Address cannot be empty");
            }
            else if (sp.isEmpty()) {
                pin.setError("Pin cannot be empty");
            }
            else if (sl.isEmpty()) {
                lmark.setError("Cannot be empty");
            }
            else if (sn.isEmpty())
            {
                pnun.setError("Phone Number cannot be empty");
            }
            else {
                update_db(sa);
            }
        }
    });
}

```

```

Navigation.findNavController(v).navigate(R.id.action_nav_offers_to_nav_Paym
ent);

        }

    });

    return root;
}

private void update_db(String sa) {
    database.collection("New_user").document(uid)
        .update("Current_Address", sa)
        .addOnCompleteListener(task -> {
            if (task.isSuccessful())
            {
                Log.d("Address:", "Address uploaded successfully");
            }
        });
}

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

```

#### 4.1.21 address.xml

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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"
    tools:context=".ui.Offers.offer">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="20dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Your Address"
            android:textSize="30sp"
            android:padding="10dp"

            android:textStyle="bold" />

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:id="@+id/addlay"
            android:layout_margin="5dp">

            <com.google.android.material.textfield.TextInputEditText
                android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:hint="Address"
        android:id="@+id/addingp"
        android:inputType="text"
    />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:id="@+id/pinlay"
    android:layout_margin="5dp">

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Pin Code"
        android:id="@+id/pinout"
        android:inputType="textPostalAddress"/>

</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:id="@+id/marklay"
    android:layout_margin="5dp">

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Nearest Land Mark"
        android:id="@+id/markinp"
        android:inputType="text"/>
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:id="@+id/phnumlay"
    android:layout_margin="5dp">

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Phone Number"
        android:inputType="number"
        android:id="@+id/phnuminp"/>
</com.google.android.material.textfield.TextInputLayout>

<Button
    android:id="@+id/confirmadd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="50dp"
    android:layout_weight="1"
    android:layout_gravity="center"
    android:text="Confirm" />

</LinearLayout>
</FrameLayout>

```

#### 4.1.22 userprofile.java

```
package com.dr.test_project;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Binder;
import android.os.Bundle;
import android.provider.MediaStore;
import android.renderscript.ScriptGroup;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;

import com.dr.test_project.databinding.ActivityCamMenuBinding;
import com.dr.test_project.databinding.ActivityFoodDetailsBinding;
import com.dr.test_project.databinding.ActivityUserProfBinding;
import com.google.android.material.bottomsheet.BottomSheetDialog;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.EventListener;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.FirebaseFirestoreException;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.squareup.picasso.Picasso;

public class user_prof extends AppCompatActivity {
    ActivityUserProfBinding binding;
    TextView user_name,user_email;
    FirebaseStorage storage;
    ImageView img;
    FloatingActionButton flt;
    FirebaseFirestore db;
    FirebaseAuth auth;
    String uid;
    ImageButton cam,gal,del;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_user_prof);
        binding = ActivityUserProfBinding.inflate(getLayoutInflater());
```

```

        setContentView(binding.getRoot());
        user_name = binding.userName1;
        user_email = binding.userEmail1;
        img = binding.imgUserProf;
        flt = binding.floatingActionButton;
        auth = FirebaseAuth.getInstance();
        db = FirebaseFirestore.getInstance();
        flt.setOnClickListener(v -> modalDialogue());
        uid = auth.getCurrentUser().getUid();
        DocumentReference documentReference =
db.collection("New_user").document(uid);
        documentReference.addSnapshotListener(this, new
EventListener<DocumentSnapshot>() {
            @Override
            public void onEvent(@Nullable
@org.jetbrains.annotations.Nullable DocumentSnapshot value, @Nullable
@org.jetbrains.annotations.Nullable FirebaseFirestoreException error) {
                user_email.setText("Email: "+value.getString("E-mail"));
                user_name.setText("Name:
"+value.getString("Name").toUpperCase());
            }
        });

    }

    private void modalDialogue() {
        //open dropdown sheet
        BottomSheetDialog sheetDialog = new
BottomSheetDialog(user_prof.this);
        View view =
LayoutInflater.from(getApplicationContext()).inflate(R.layout.activity_cam
menu,null);
        ActivityCamMenuBinding camBinding =
ActivityCamMenuBinding.bind(view);
        sheetDialog.setContentView(view);
        cam=camBinding.cam;
        gal=camBinding.gal;
        del=camBinding.del;
        sheetDialog.show();
        cam.setOnClickListener(v -> {
            Intent camintent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(camintent,1000);
            sheetDialog.dismiss();
        });
        gal.setOnClickListener(v -> {
            Intent openintent = new Intent(Intent.ACTION_GET_CONTENT);
            openintent.setType("image/*");
            Intent selint = Intent.createChooser(openintent,"Images");
            startActivityForResult(selint,2000);
            sheetDialog.dismiss();
        });
        del.setOnClickListener(v -> {
            Picasso.get().load(R.drawable.login_1).into(img);

            sheetDialog.dismiss();
        });
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode,

```

```

@Nullable @org.jetbrains.annotations.Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode==1000 && resultCode == Activity.RESULT_OK){
        Bitmap bit = (Bitmap) data.getExtras().get("data");
        img.setImageBitmap(bit);

    }
    else if(requestCode==2000 && resultCode == Activity.RESULT_OK){
        Uri galuri = data.getData();
        img.setImageURI(galuri);
        uploadgal(galuri);
    }
}

private void uploadgal(Uri galuri) {
    storage = FirebaseStorage.getInstance();
    StorageReference ref = storage.getReference().child("New_profile");
    ref.child(uid+".jpg")
        .putFile(galuri)
        .addOnCompleteListener(task -> {
            if(task.isSuccessful())
            {
                ref.child(uid+".jpg").getDownloadUrl()
                    .addOnSuccessListener(uri -> {
                        String uploaduri = galuri.toString();
                        Updatedb(uploaduri);
                    });
            }
            }).addOnFailureListener(e -> {
                Log.d("Error:",e.toString());
            });
});

private void Updatedb(String uploaduri) {

    db.collection("New_user").document(uid)
        .update("New_profile",uploaduri)
        .addOnCompleteListener(task -> {
            if(task.isSuccessful())
            {

Snackbar.make(user_prof.this, binding.getRoot(), "Image
uploaded", Snackbar.LENGTH_SHORT).show();
            }
        });

    }

private void upload(Bitmap bit) {
}
}

```

#### 4.1.23 user\_profile.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".user_prof"
android:background="@drawable/new_theme">

```

```

<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".user_profile_act"
    android:background="@color/transparent">
    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:cardCornerRadius="35dp"
        android:layout_margin="10dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:background="@drawable/green"
            >
            <androidx.cardview.widget.CardView
                android:layout_width="300dp"
                android:layout_height="300dp"
                app:cardCornerRadius="35dp"
                android:layout_gravity="center"
                android:layout_margin="30dp"
                >
                <ImageView
                    android:id="@+id/img_user_prof"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:src="@drawable/login_1"
                    android:cropToPadding="true"/>

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/floatingActionButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:clickable="true"
    android:layout_gravity="bottom"
    app:srcCompat="@drawable/ic_menu_gallery" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="60dp"
    app:cardCornerRadius="35dp"
    android:layout_margin="15dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/transparent1"
        android:orientation="vertical"
        android:padding="10dp">

        <TextView
            android:id="@+id/user_name1"
            android:layout_width="match_parent"
            android:layout_height="41dp"
            android:layout_marginTop="20dp"
            android:text="name"
            android:textColor="@color/black"
            android:textSize="20sp" />

```

```

        <TextView
            android:id="@+id/user_email1"
            android:layout_width="match_parent"
            android:layout_height="41dp"
            android:layout_marginTop="20dp"
            android:text="@string/abcd_gmail_com"
            android:textColor="@color/black"
            android:textSize="20sp" />

    </LinearLayout>
</androidx.cardview.widget.CardView>

</LinearLayout>
</androidx.cardview.widget.CardView>

</FrameLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

#### 4.1.24 constatnt.java

```

package com.dr.test_project.adapter;

public class Constants {
    public static String Food_price="price";
    public static String Food_des="description";
    public static String id="id";
    public static String Food_img="pic";
    public static String Food_est="Est";

    public static String Food_id="Est";
    public static String Food_title="name";
}

```

#### 4.1.25 home\_adapter.java

```

package com.dr.test_project.adapter;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.provider.SyncStateContract;
import android.renderscript.ScriptGroup;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import androidx.constraintlayout.solver.state.State;
import androidx.recyclerview.widget.RecyclerView;

import com.dr.test_project.FoodModel;
import com.dr.test_project.Food_details;
import com.dr.test_project.R;
import com.dr.test_project.databinding.FoodListBinding;
import com.squareup.picasso.Picasso;

import org.jetbrains.annotations.NotNull;

```



```

import java.util.List;

public class home_adapter extends
RecyclerView.Adapter<home_adapter.InnerClass> {

    List<FoodModel> foodModelList;
    Context ctx;
    int id = 0;

    public home_adapter(List<FoodModel> foodModelList, Context ctx) {
        this.foodModelList = foodModelList;
        this.ctx = ctx;
    }

    @NonNull
    @NotNull
    @Override
    public InnerClass onCreateViewHolder(@NonNull @NotNull ViewGroup
parent, int viewType) {
        LayoutInflater inflater =LayoutInflater.from(ctx);
        View design = inflater.inflate(R.layout.food_list,parent,false);
        InnerClass ob = new InnerClass(design);
        return ob;
    }

    @Override
    public void onBindViewHolder(@NonNull @NotNull InnerClass holder, int
position) {
        //display the value for food_list view which was created early to
        //overlap with the search_fragment.xml
        FoodModel food = foodModelList.get(position);
        holder.name.setText(food.getName());
        holder.price.setText(food.getPrice());
        // holder.des.setText(food.getDescription());
        // holder.time.setText(food.getEST());
        Picasso.get().load(food.getImgurl()).into(holder.image);

        holder.cardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FoodModel food = foodModelList.get(position);
                Bundle bundle = new Bundle();
                // bundle.putString(Constants.id, food.getId());
                bundle.putString(Constants.Food_title,food.getName());
                bundle.putString(Constants.Food_des,food.getDescription());
                bundle.putString(Constants.Food_est, food.getEST());
                bundle.putString(Constants.Food_price, food.getPrice());
                bundle.putString(Constants.Food_img, food.getImgurl());
                bundle.putString(Constants.Food_id, String.valueOf(id));
                id++;
                Intent i = new Intent(ctx,Food_details.class);
                i.putExtra("data",bundle);
                ctx.startActivity(i); //fetch to a new screen to get the
full details
            }
        });
    }

    @Override
    public int getItemCount() {

```

```

        return foodModelList.size();
    }
    // this class will help to define the objects used in the layout
    public class InnerClass extends RecyclerView.ViewHolder {
        FoodListBinding itemBinding;
        CardView cardView;
        ImageView image;
        TextView name;
        TextView des;
        TextView time;
        TextView price;
        public InnerClass(@NonNull @NotNull View itemView) {
            super(itemView);
            itemBinding = FoodListBinding.bind(itemView);
            cardView = itemBinding.cardViewFood;
            image = itemBinding.foodPic;
            name = itemBinding.itemTitle;
            // des = itemBinding.des;
            // time= itemBinding.time;
            price = itemBinding.price;
        }
    }
}

```

#### 4.1.26 Oder\_details.java

```

package com.dr.test_project.adapter;

public class Order_details {

    public static String Food_img;
    public static String Food_price;
    public static String Food_title;

}

```

#### 4.1.27 payment.java

```

package com.dr.test_project.ui.payment;

import androidx.lifecycle.ViewModelProvider;

import android.os.Bundle;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

import android.text.Layout;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import com.dr.test_project.R;
import com.dr.test_project.databinding.OrderFragmentBinding;
import com.dr.test_project.databinding.PaymentFragmentBinding;
import com.dr.test_project.registration;

```

```

import com.dr.test_project.ui.order.OrderViewModel;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.EventListener;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.FirebaseFirestoreException;
import com.google.firebase.storage.FirebaseStorage;

public class payment extends Fragment {

    private PaymentViewModel mViewModel;
    PaymentFragmentBinding binding;
    TextView amt;
    EditText Card, cvv, dt, nm, ph;
    Button confirm;
    LinearLayout layout;
    FirebaseAuth auth;
    FirebaseFirestore store;
    String uid;

    public static payment newInstance() {
        return new payment();
    }

    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
ViewGroup container,
                                @Nullable Bundle savedInstanceState) {
        mViewModel =
            new ViewModelProvider(this).get(PaymentViewModel.class);
        binding = PaymentFragmentBinding.inflate(inflater, container,
false);
        View root = binding.getRoot();
        amt = binding.amt;
        Card = binding.cardno;
        cvv = binding.cvv;
        dt = binding.editTextDate;
        nm = binding.editname;
        ph = binding.number;
        confirm = binding.pay;
        layout = binding.layli;
        auth = FirebaseAuth.getInstance();
        store = FirebaseFirestore.getInstance();
        uid = auth.getCurrentUser().getUid();
        DocumentReference documentReference =
store.collection("Order").document(uid);
        documentReference.addSnapshotListener(requireActivity(), new
EventListener<DocumentSnapshot>() {
            @Override
            public void onEvent(@Nullable
@org.jetbrains.annotations.Nullable DocumentSnapshot value, @Nullable
@org.jetbrains.annotations.Nullable FirebaseFirestoreException error) {
                amt.setText("Payable Amount: "+value.getString("Price"));
            }
        });

        confirm.setOnClickListener(v -> {
            String card = String.valueOf(Card.getText()).trim();
            String cv = String.valueOf(cvv.getText()).trim();
            String n = String.valueOf(nm.getText()).trim();
            String d = String.valueOf(dt.getText()).trim();
            String num = String.valueOf(ph.getText()).trim();

```

```

        if(card.length()<16)
        {
            Card.setError("Cannot Be Empty/Less than 16 digits");
        }
        else if(cv.length()>3 )
        {
            cvv.setError("Cannot Be Empty/More than 3 digits");
        }
        else if(n.isEmpty())
        {
            nm.setError("Cannot Be Empty");
        }
        else if(d.isEmpty())
        {
            dt.setError("Cannot Be Empty");
        }
        else if(num.isEmpty())
        {
            ph.setError("Cannot Be Empty");
        }
        else
        {
            Toast.makeText(requireContext(), "Order Confirmed \n Thank
You For Ordering", Toast.LENGTH_SHORT).show();
            layout.setVisibility(View.INVISIBLE);
            DocumentReference reference =
store.collection("Order").document(uid);
            reference.delete();
        }
    });

    return root;
}

@Override
public void onDestroyView() {
    super.onDestroyView();
    binding = null;
}
}

```

#### 4.1.28 fragment\_payment.xml

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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"
    tools:context=".ui.payment.payment">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="20dp"
        android:layout_margin="20dp"
        android:id="@+id/layli">

        <EditText

```

```

        android:id="@+id/cardno"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Card Number"
        android:inputType="number" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <EditText
        android:id="@+id/editTextDate"
        android:layout_width="183dp"
        android:layout_height="match_parent"
        android:hint="Expire Date"
        android:ems="10"
        android:inputType="date" />

    <EditText
        android:id="@+id/cvv"
        android:layout_width="79dp"
        android:layout_marginLeft="70dp"
        android:layout_height="match_parent"
        android:hint="CVV"
        android:ems="10"
        android:inputType="numberPassword" />
</LinearLayout>

<EditText
    android:id="@+id/editname"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Name on Card"
    android:inputType="text" />

<EditText
    android:id="@+id/number"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Phone Number"
    android:inputType="number" />

<TextView
    android:id="@+id/amt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Amount"
    android:textColor="@color/black"
    android:textSize="20sp" />

<Button
    android:id="@+id/pay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    android:layout_gravity="center"
    android:text="Confirm" />
</LinearLayout>
</FrameLayout>

```

#### 4.1.29 activity\_drawer\_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:showIn="navigation_view">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/nav_home"
            android:icon="@drawable/home"
            android:title="@string/menu_home" />
        <item
            android:id="@+id/nav_Search"
            android:icon="@drawable/ic_baseline_search_24"
            android:title="Search" />

    </group>
    <item android:title="Order">
        <menu>
            <item
                android:id="@+id/nav_Order"
                android:icon="@drawable/order2"
                android:title="Cart" />
            <item
                android:id="@+id/nav_Payment"
                android:title="Payment"
                android:icon="@drawable/ic_baseline_attach_money_24"/>
        </menu>

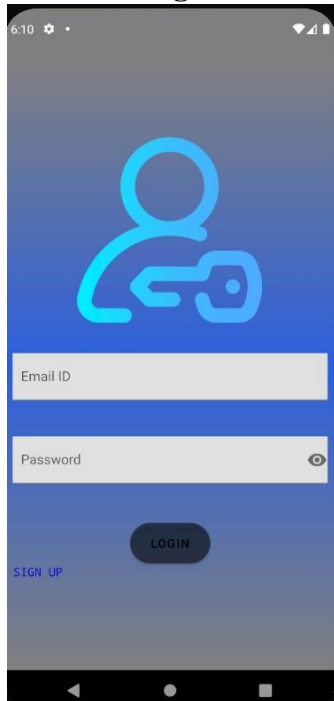
    </item>
    <item android:title="">
        <menu>
            <item
                android:id="@+id/nav_offers"
                android:title=""
            />
        </menu>

    </item>

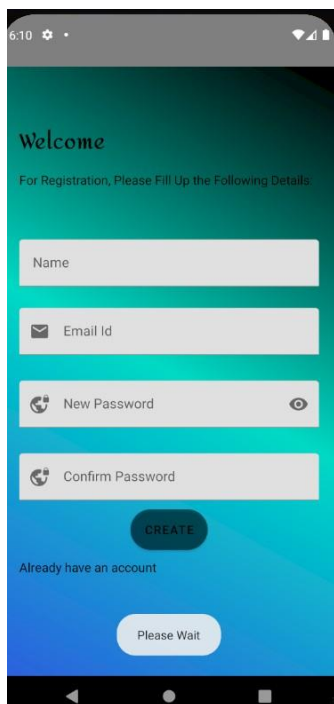
</menu>
```

## 4.2 Screenshots as Outputs

### 4.2.1 Login Screen



### 4.2.2 SignUp Screen



### 4.2.3 Home Screen

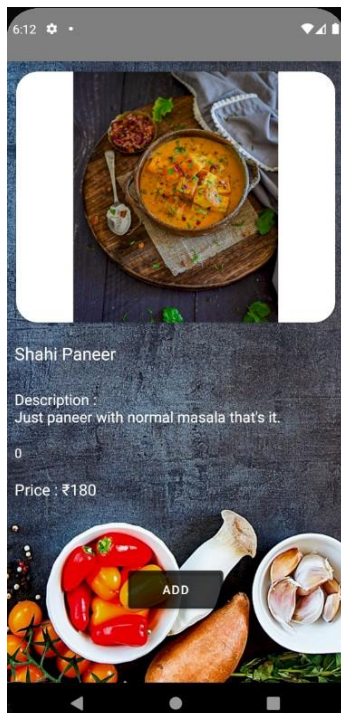


### 4.2.4 Search Screen

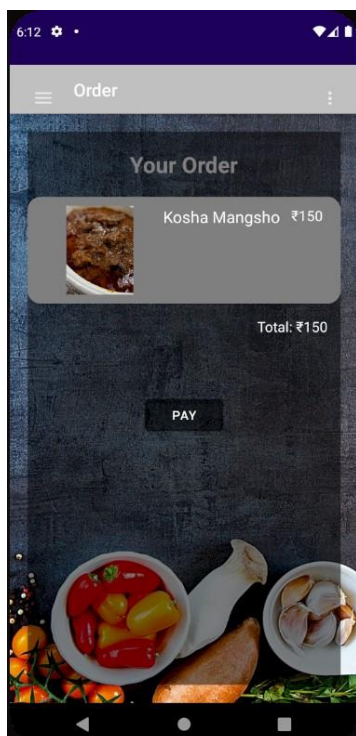




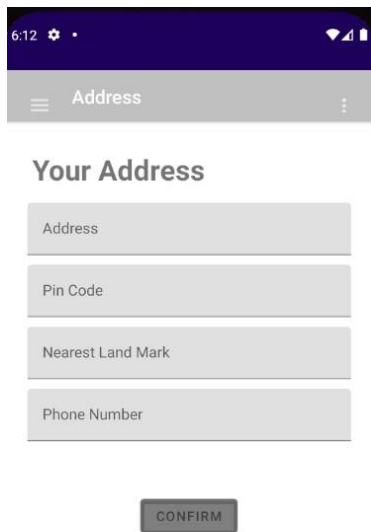
#### 4.2.5 Food Details Screen



#### 4.2.6 Order Screen

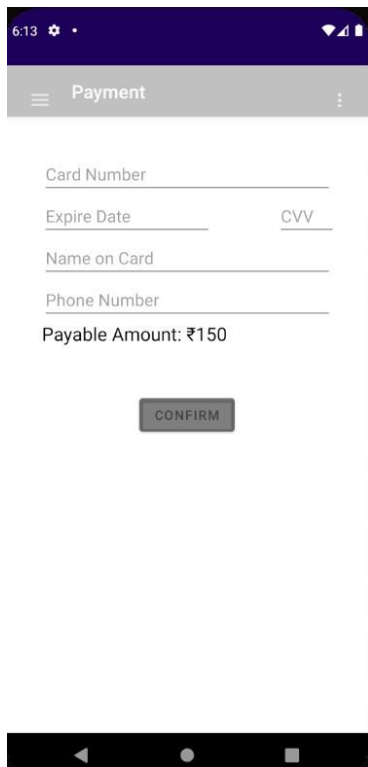


### 4.2.7 Address screen



The screenshot shows a mobile application interface for the 'Address' screen. At the top, there is a dark blue header bar with the time '6:12', a settings icon, and a back arrow. Below the header is a grey navigation bar with a hamburger menu icon on the left, the word 'Address' in the center, and a vertical ellipsis icon on the right. The main content area is white and titled 'Your Address' in bold. Below the title are four stacked input fields with light grey borders and placeholder text: 'Address', 'Pin Code', 'Nearest Land Mark', and 'Phone Number'. At the bottom of the screen is a grey button with the text 'CONFIRM' in white. The bottom of the screen shows a black Android navigation bar with back, home, and recent apps icons.

### 4.2.8 Payment screen



The screenshot shows a mobile application interface for the 'Payment' screen. At the top, there is a dark blue header bar with the time '6:13', a settings icon, and a back arrow. Below the header is a grey navigation bar with a hamburger menu icon on the left, the word 'Payment' in the center, and a vertical ellipsis icon on the right. The main content area is white. It contains several input fields with light grey borders and placeholder text: 'Card Number', 'Expire Date', 'CVV', 'Name on Card', and 'Phone Number'. Below these fields, the text 'Payable Amount: ₹150' is displayed. At the bottom of the screen is a grey button with the text 'CONFIRM' in white. The bottom of the screen shows a black Android navigation bar with back, home, and recent apps icons.

## Chapter 5. Testing and Maintenance

### 5.1 Objective of testing

Software Testing has different goals and objectives. The major objectives of Software testing are:

- i. Finding defects which may get created by the programmer while developing the software.
- ii. Gaining confidence in and providing information about the level of quality.
- iii. To prevent defects.
- iv. To make sure that the end result meets the business and user requirements.
- v. To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- vi. To gain the confidence of the customers by providing them a quality product.

### 5.2 Test cases

#### 5.2.1

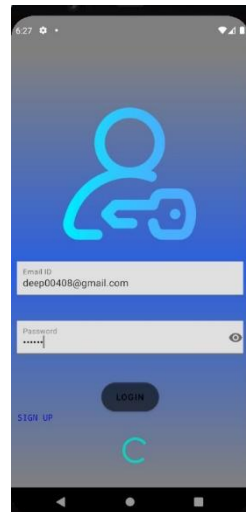
Test Case Number	1
Test Case Name	Log In Page
Test Case Description	For login user has to register first if they havent registered they will not get the access to the home screen.
Test performed by	Debanjan Roy

*Items to be tested:*

Sucessful login

Input	Expected output
Tap on the Log In button	1. Login error message 2. Sucessful login 3. Directed to proper page

*Output received:* Sucessful LogIn, Login error, Directed to proper page



### 5.2.2

Test Case Number	2
Test Case Name	Blank credential in Log In screen
Test Case Description	For log in, user has to provide registered phone number.
Test performed by	Debanjan Roy

*Items to be tested:*

Blank Log In details

Input	Expected output
-------	-----------------

Tap on the Sign In button	1.Error message
---------------------------	-----------------

*Output received:* Error message



### 5.2.3

Test Case Number	3
Test Case Name	Registration
Test Case Description	If user already registered/registering using already registered email id
Test performed by	Debanjan Roy

*Items to be tested:*

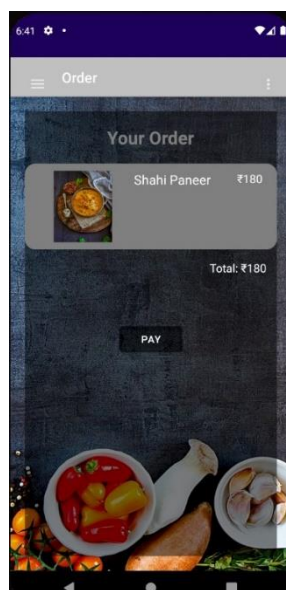
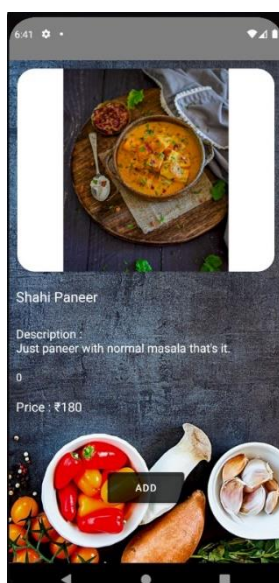
Blank Log In details

Input	Expected output
Tap on the Create button	1.Error message



#### 5.2.4

Test Case Number	4
Test Case Name	Order
Test Case Description	If selected item placed in order screen
Test performed by	Debanjan Roy



## **Chapter 6. Security Measures**

### **6.1 Database Security measures**

In this project we have used Google Firebase as our database. In addition, several Firebase services also encrypt their data at rest, like Cloud Firestore, Cloud Functions for Firebase, Cloud Storage for Firebase, Firebase Crashlytics, Firebase Authentication, Firebase Cloud Messaging, Firebase Realtime Database, Firebase Test Lab. Firebase services encrypt data in transit using HTTPS and logically isolate customer data. To keep personal data safe, Firebase employs extensive security measures to minimize access, Firebase restricts access to a select employees who have a business purpose to access personal data. Firebase logs employee access to systems that contain personal data. Firebase only permits access to personal data by employees who sign in with Google Sign-In and 2-factor authentication.

## **Chapter 7. Limitation**

### **7.1 Limitation**

- i. This application cannot run without internet.
- ii. Order status is not updated and user cannot see their already ordered items
- iii. Only one order can be placed at a time.

## **Chapter 8. Conclusion**

Nowadays one of the most important principles of designing a software is to keep the interface simple yet effective and absolutely user-friendly. This has been the guiding force behind making our app. The interface that we provide is no exception. Against the backdrop of this ideology we design a simple yet very effective interface. Testing is also necessary for the failures and problems that arise during the operation of an application. Our testing and maintenance activity included a post implementation of the application features to ensure that the newly implemented app meets the objective. Bug fixing has been done to implement the app effectively.



## Chapter 9. References

- [1] PraveenRuhil (2021). Introduction to Android Development from <https://www.geeksforgeeks.org/introduction-to-android-development/>
- [2] Jackson, W. (2011). Android Apps for Absolute Beginners. New York, NY. Apress.
- [3] Meier, R. (2010). Professional Android 2 Application Development. Indianapolis, IN. Wiley Publishing, Inc.
- [4] Bishop, T. (2012, March 6). “Google Play replaces Android Market, consolidates Google’s media marketplaces”. Retrieved from <http://www.geekwire.com/2012/google-playreplaces-android-market-consolidatesgoogles-media-marketplaces>
- [5] What is Android? (2012). Retrieved March 10, 2012 from <http://developer.android.com/guide/basics/what-is-android.html>
- [6] Felker, D. (2011). Android Application Development for Dummies. Hoboken, NJ. Wiley Publishing, Inc.