THE MEALER APP

SEG2105- Software Engineering- Fall 2022

School of Electrical Engineering and Computer Science
University of Ottawa

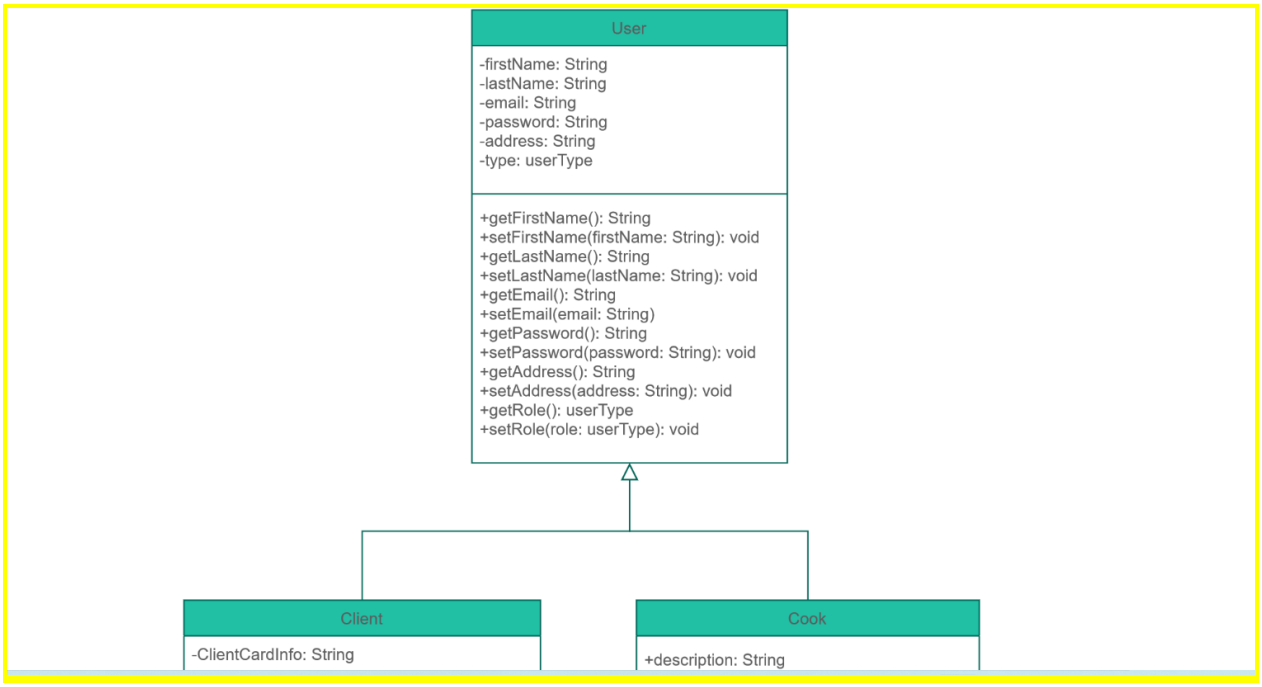
Course Coordinator: Hussein Al Osman
Project TA: Dharmin Sodwadia
Group#: 13

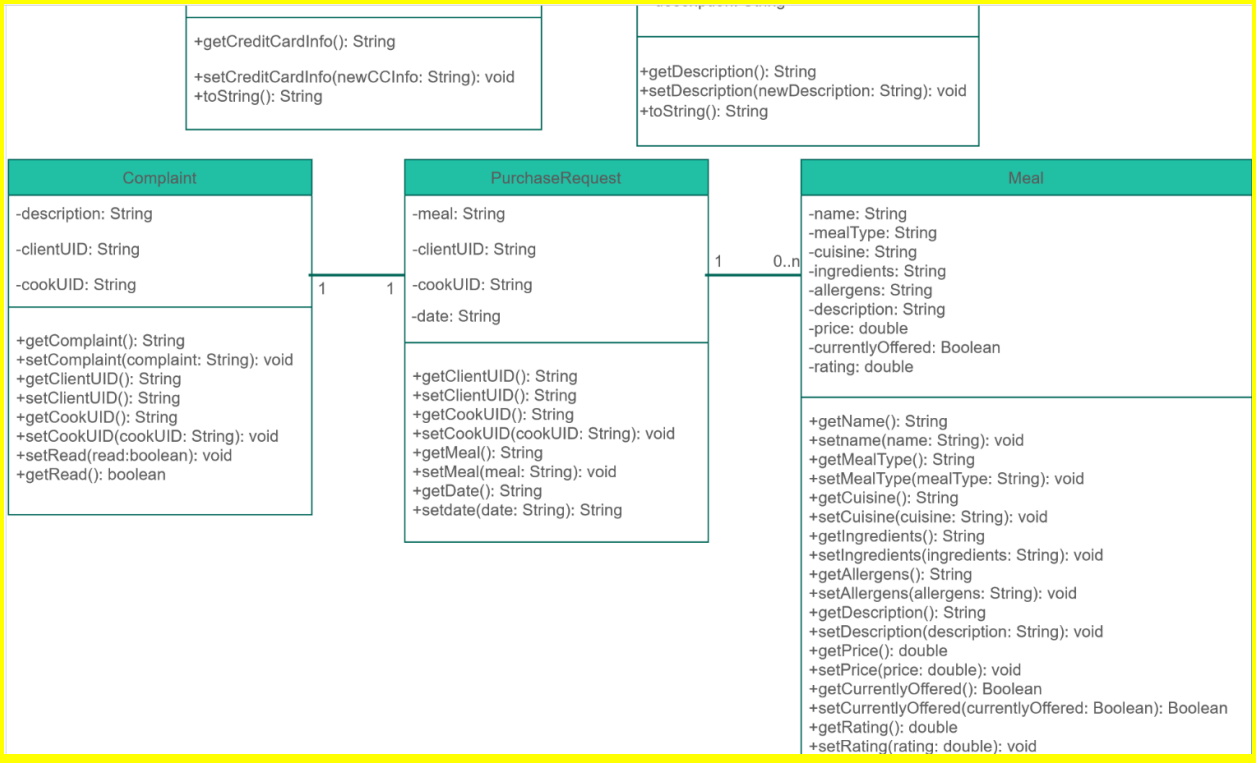
Joanna Reginald #300245131
Kyle Mendes #300253978
Lauren Gu # 300320106
Maisha Habib #300263062
Rami-Slimane Kadi #300237431
Algo Woolf # 300267107
Submission Date: 2022-12-07

Introduction:

The Mealer App is an advanced technology app that allows individuals to order meals online at home easily. Registering through our app gives one access to thousands of freelance chefs who are ready to cook delicious meals. Chefs are employees registered in this app and are monitored by an administrator. The app has a dedicated page for sending messages to the administrator about their experience on a particular issue, so any criticism or general complaints regarding food quality or quantity will be taken into account. Overall, the functioning process of this app is very safe and reliable as multiple people cannot log into one account with the same email address. With many more features, this App will simultaneously work with all the coded programs and get one a home-cooked meal in a flash!

UML Class Diagram:





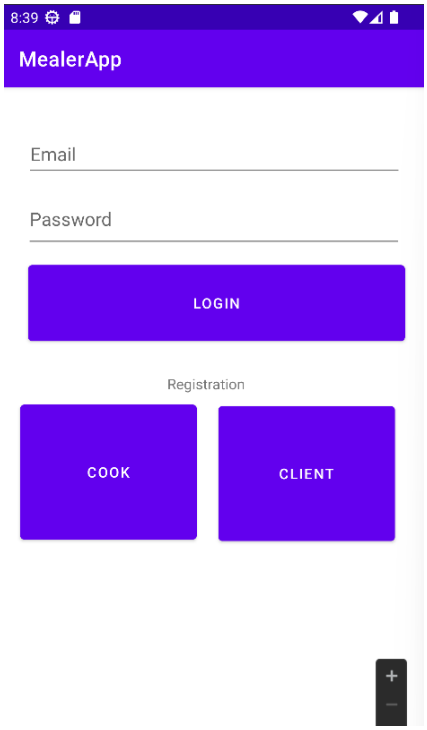
Contributions Of Team Members:

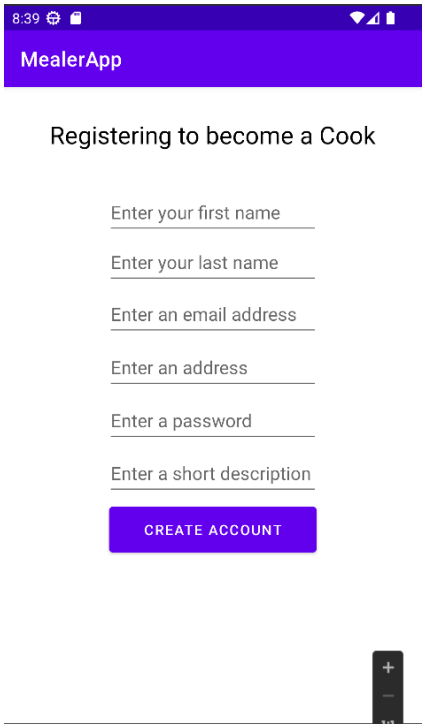
Group members	Deliverable 1	Deliverable 2	Deliverable 3	Deliverable 4
Joanna Reginald	-Coded Log-in page for the Mealer App	-Coded Complaints Class - Coded Complaints Database	-Local Test: Coded Junit test for the meal - Coded an Extra client instrumented test case.	-Final Report - Helped Code: the Client can search for a meal - Helped Code: The client can see Search results for meals offered by non-suspended Cooks
Kyle Mendes	-Coded cook class	-Coded adapter to	-Coded data validation	-Coded Make complaint

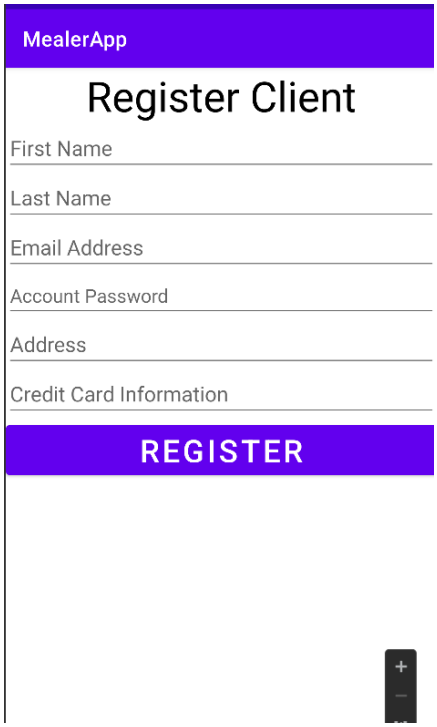
	-Coded cook registration -Helped with welcome page	view the complaints from the database -Coded view complaints page	-Coded a unit test	page -Coded data validation -Coded view information -Took screenshots of app
Lauren Gu	-Coded the database class -Coded user database class -Helped code welcome page -Coded User Class	-Coded the test cases -Helped code complaints database -Coded Suspension Page -Coded handle Complaint Page -Made UML -Coded suspended cook page	-Coded menu page -Coded add a meal page -Coded edit meal page -Made UML -Coded Meal Class -Coded Menu class -Coded Menu Database Class	-Coded: the Client can search for a meal -Coded:The client can see Search results for meals offered by non-suspended Cooks -Coded client can view the meal's information for each meal -Coded Client can submit at purchase request -Coded cook can receive the purchase request from a client -Coded the client can view the status of their purchase -Helped code

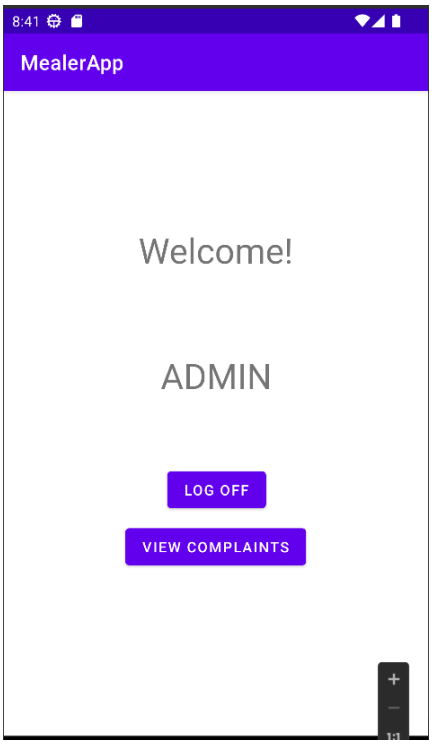
				the cook can approve and deny meal requests
Maisha Habib	-Coded client class -Coded client registration	-Helped code suspension page - Edited deleted database(Administrator)	-Added code to edit meal page -added code to menu database	-Added code to request database -added code to Purchase pending class - coded a view request xml page(not used) - coded a reject/deny page +class (not used)
Rami Slimane-Kadi	-Made UML -Added methods to the User class	-Local Test: Coded Junit tests to test code	-Local Test: Coded 2 Junit tests to test code	- Coded 4 Junit test to test code
Algo Woolf	-Welcome Page	-Edited database	-UML	-UML

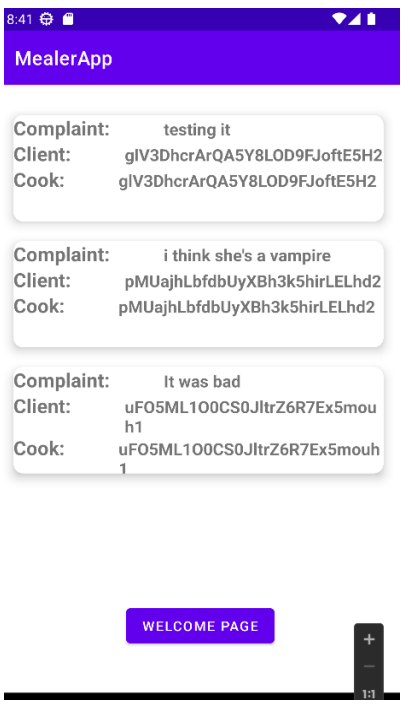
All the screenshots of Mealer App:

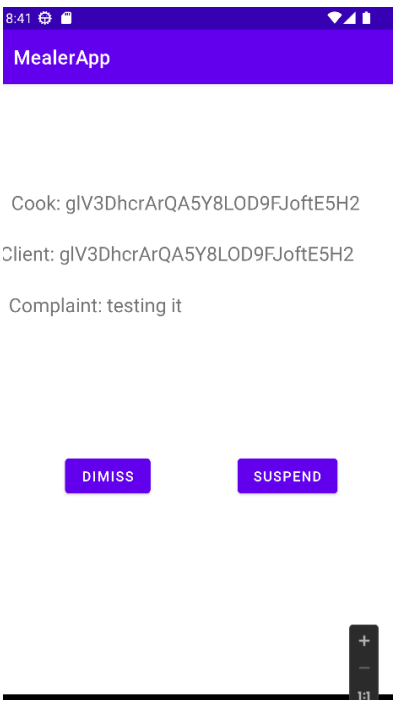

8:39
MealerApp
Email
Password
LOGIN
Registration
COOK
CLIENT

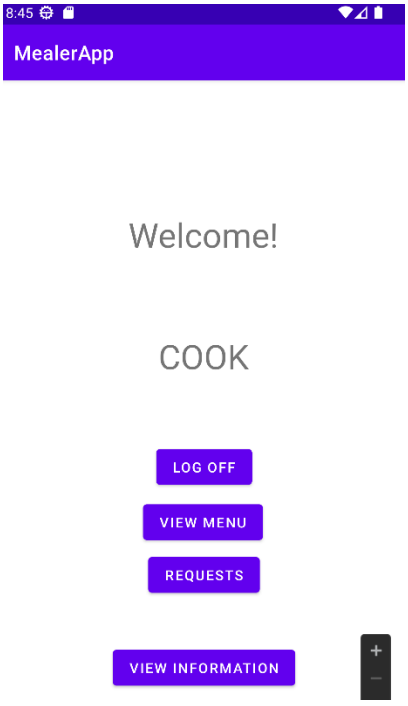

8:39
MealerApp
Registering to become a Cook
Enter your first name
Enter your last name
Enter an email address
Enter an address
Enter a password
Enter a short description
CREATE ACCOUNT

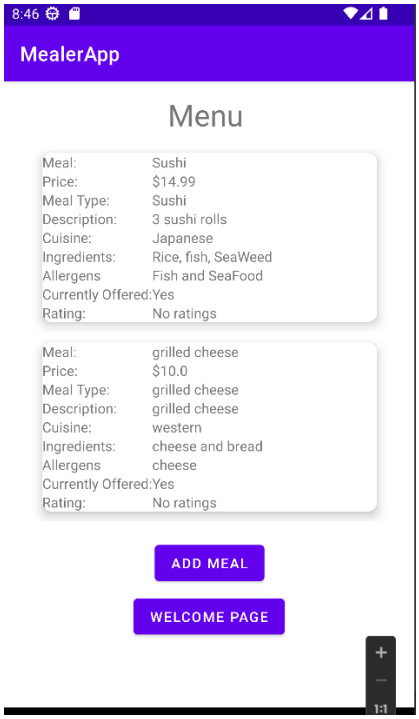

MealerApp
Register Client
First Name
Last Name
Email Address
Account Password
Address
Credit Card Information
REGISTER

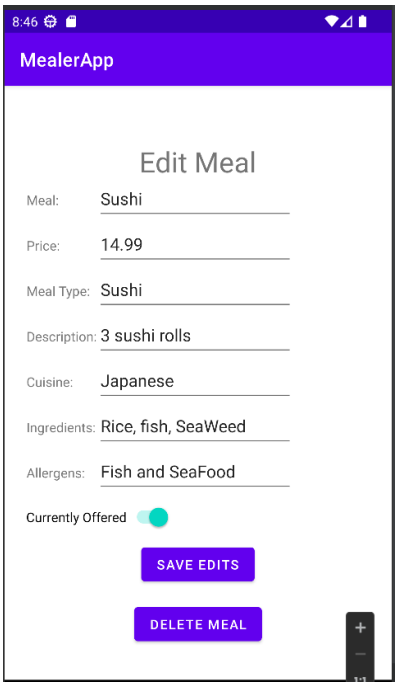

8:41
MealerApp
Welcome!
ADMIN
LOG OFF
VIEW COMPLAINTS

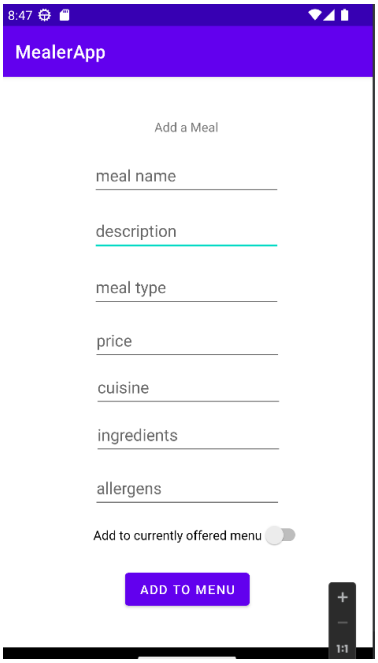

8:41
MealerApp
Complaint: testing it
Client: glV3DhcrArQA5Y8LOD9FJoftE5H2
Cook: glV3DhcrArQA5Y8LOD9FJoftE5H2
Complaint: i think she's a vampire
Client: pMUajhLbfdbUyXBh3k5hirLELhd2
Cook: pMUajhLbfdbUyXBh3k5hirLELhd2
Complaint: It was bad
Client: uFO5ML1O0CS0JltrZ6R7Ex5mouh1
Cook: uFO5ML1O0CS0JltrZ6R7Ex5mouh1
WELCOME PAGE

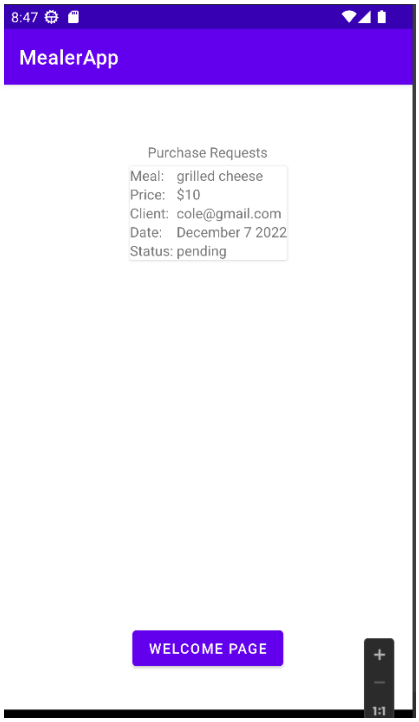

8:41
MealerApp
Cook: glV3DhcrArQA5Y8LOD9FJoftE5H2
Client: glV3DhcrArQA5Y8LOD9FJoftE5H2
Complaint: testing it
DIMISS
SUSPEND

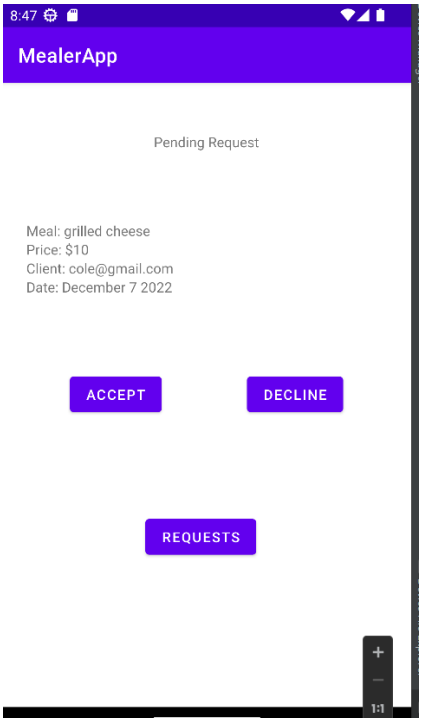

8:45
MealerApp
Welcome!
COOK
LOG OFF
VIEW MENU
REQUESTS
VIEW INFORMATION

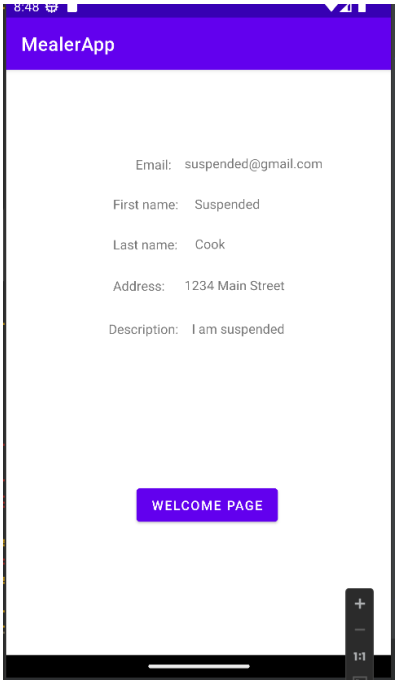

8:46
MealerApp
Menu
Meal: Sushi
Price: $14.99
Meal Type: Sushi
Description: 3 sushi rolls
Cuisine: Japanese
Ingredients: Rice, fish, SeaWeed
Allergens Fish and SeaFood
Currently Offered: Yes
Rating: No ratings
Meal: grilled cheese
Price: $10.0
Meal Type: grilled cheese
Description: grilled cheese
Cuisine: western
Ingredients: cheese and bread
Allergens cheese
Currently Offered: Yes
Rating: No ratings
ADD MEAL
WELCOME PAGE

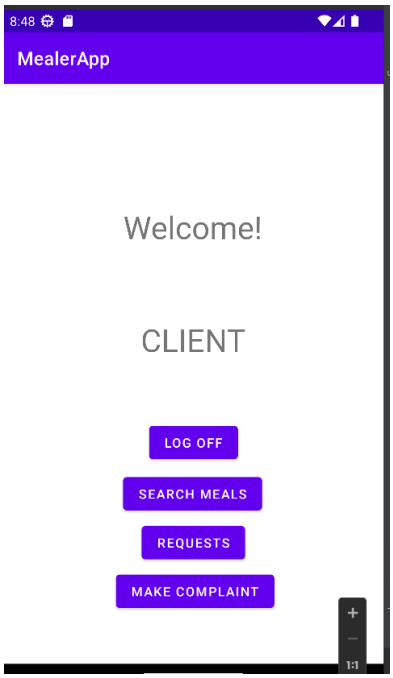

8:46
MealerApp
Edit Meal
Meal: Sushi
Price: 14.99
Meal Type: Sushi
Description: 3 sushi rolls
Cuisine: Japanese
Ingredients: Rice, fish, SeaWeed
Allergens: Fish and SeaFood
Currently Offered
SAVE EDITS
DELETE MEAL

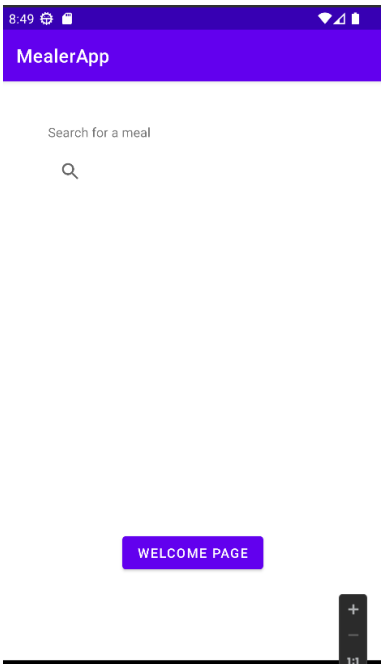

8:47
MealerApp
Add a Meal
meal name
description
meal type
price
cuisine
ingredients
allergens
Add to currently offered menu
ADD TO MENU


8:47
MealerApp
Purchase Requests
Meal: grilled cheese
Price: $10
Client: cole@gmail.com
Date: December 7 2022
Status: pending
WELCOME PAGE

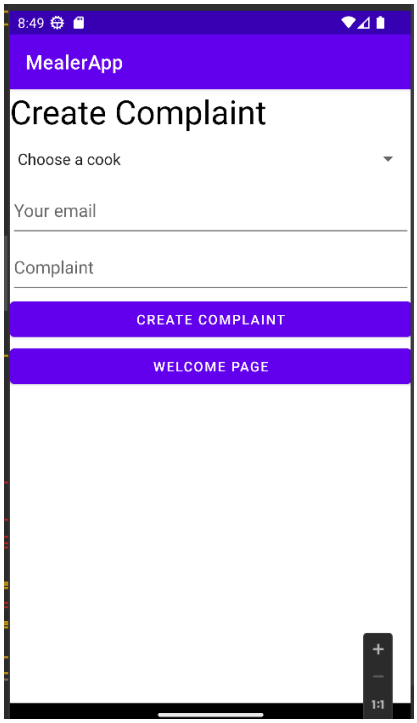

8:47
MealerApp
Pending Request
Meal: grilled cheese
Price: $10
Client: cole@gmail.com
Date: December 7 2022
ACCEPT
DECLINE
REQUESTS


MealerApp
Email: suspended@gmail.com
First name: Suspended
Last name: Cook
Address: 1234 Main Street
Description: I am suspended
WELCOME PAGE


8:48
MealerApp
Welcome!
CLIENT
LOG OFF
SEARCH MEALS
REQUESTS
MAKE COMPLAINT


8:49
MealerApp
Search for a meal
WELCOME PAGE


8:49
MealerApp
Purchase Requests
Meal: grilled cheese
Price: $10
Cook: suspended@gmail.com
Date: December 7 2022
Status: pending
WELCOME PAGE


8:49
MealerApp
Create Complaint
Choose a cook
Your email
Complaint
CREATE COMPLAINT
WELCOME PAGE

AddMealPage.java

```java
package com.example.mealerapp;

import androidx.appcompat.app.AppCompatActivity;

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

public class AddMealPage extends AppCompatActivity {

    private EditText name,description,ingredients,allergerns, cuisine, mealType, price;
    private Switch currentlyOffered;
    private Button addButton;

    private String mealName,descriptionString,ingredientsString,allergensString, cuisineString, mealTypeString, UID;
    private boolean isOffered;
    private double priceDouble;


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

        Bundle bundle = getIntent().getExtras();
        UID = bundle.getString("UID");


        name = (EditText) findViewById(R.id.mealNameEditText);
        description = (EditText) findViewById(R.id.descriptionEditText);
        ingredients = (EditText) findViewById(R.id.ingredientsEditText);
        allergerns = (EditText) findViewById(R.id.allergensEditText);
```

```
        cuisine = (EditText)findViewById(R.id.cuisineEditText);
        mealType = (EditText)findViewById(R.id.mealNameEditText);
        price = (EditText)findViewById(R.id.priceNumberEdit);
        addButton = (Button)findViewById(R.id.addButton);
        currentlyOffered = (Switch)findViewById(R.id.currentlyOfferedSwitch);
        currentlyOffered.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                isOffered = isChecked;
            }
        });

        addButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mealName = name.getText().toString();
                descriptionString = description.getText().toString();
                allergensString = allergerns.getText().toString();
                ingredientsString = ingredients.getText().toString();
                cuisineString = cuisine.getText().toString();
                mealTypeString = mealType.getText().toString();
                priceDouble = Double.parseDouble(price.getText().toString());
                addNewMeal(v);
                returnToMenu(v);
            }
        });
    }

    public void addNewMeal(View view){
        Meal newMeal = new Meal(mealName,mealTypeString,cuisineString,ingredientsString,allergensString,descriptionString,priceDouble,isOffered);
        Menu menu = new Menu();
        menu.addMeal(UID,newMeal);
    }

    public void returnToMenu(View view){
        Intent returnToMenu = new Intent(getApplicationContext(),MenuPage.class);

        returnToMenu.putExtra("UID",UID);
        startActivity(returnToMenu);
    }
}
```

Client.java

```java
package com.example.mealerapp;

public class Client extends User {

    private String creditCardInfo;

    public Client( String firstName, String lastName, String emailAddress, String accountPassword, String address, String creditCardInfo) {
        super(firstName, lastName, emailAddress, accountPassword, userType.CLIENT, address);
        this.creditCardInfo = creditCardInfo;
    }

    public void registerClient(){

       UserDatabase dtb = new UserDatabase();
        dtb.registerUser(this);
    }

    public String getCreditCardInfo() {
        return creditCardInfo;
    }

    public void setCreditCardInfo(String newCCInfo) {
        creditCardInfo = newCCInfo;
    }

    public String toString() {
        return "\nAccount Information  \nFirst name: " + firstName + "\n" + "Last name: " + lastName + "\n" + "Email: " + email + "\n" + "Password: " + password + "\n" + "Credit Card Informa
    }

    public PurchaseRequest createRequest(String clientUID, String cookUID, String meal, String date){
        return new PurchaseRequest(cookUID,clientUID,meal);
    }
```

ClientRegistration.java

```java
package com.example.mealerapp;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;


import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.content.Intent;

public class ClientRegistration extends AppCompatActivity {

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


        //Cook newCook = new Cook(firstname.getText().toString(), lastname.getText().toString(), email.getText().toString(), password.getText().toString(), description.getText().to


    }


    public void createClient(View view) {
        //setContentView(R.layout.activity_client_registration);
        EditText firstName= (EditText)  findViewById(R.id.firstName);
        EditText lastName= (EditText)  findViewById(R.id.lastName);
        EditText emailAddress= (EditText)  findViewById(R.id.emailAddress);
        EditText accountPassword= (EditText)  findViewById(R.id.accountPassword);
        EditText address= (EditText)  findViewById(R.id.address);
        EditText creditCardInfo= (EditText)  findViewById(R.id.creditCardInfo);

        Button Register= (Button) findViewById(R.id.Register);

         Register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String firstNameOne = firstName.getText().toString();
                String lastNameOne = lastName.getText().toString();
                String emailAddressOne = emailAddress.getText().toString();
                String accountPasswordOne= accountPassword.getText().toString();
                String addressOne = address.getText().toString();
                String creditCardInfoOne = creditCardInfo.getText().toString();

                if(checkFields() && checkEmailFormat(emailAddressOne)) {
                    Client newClient = new Client(firstNameOne, lastNameOne, emailAddressOne, accountPasswordOne, addressOne, creditCardInfoOne);
                    newClient.registerClient();

                    Intent intent = new Intent(getApplicationContext(), WelcomePage.class);
                    intent.putExtra("role", "client");
                    startActivity(intent);
                }
            }
        });
```

```
    }

    private boolean checkFields(){
        boolean status = true;

        EditText firstname = (EditText) findViewById(R.id.firstName);
        EditText lastname = (EditText) findViewById(R.id.lastName);
        EditText email = (EditText) findViewById(R.id.emailAddress);
        EditText password = (EditText) findViewById(R.id.accountPassword);
        EditText address = (EditText) findViewById(R.id.address);
        EditText description = (EditText) findViewById(R.id.creditCardInfo);

        if(firstname.length() == 0){
            firstname.setError("This field is required");
            status = false;
        }
        if(lastname.length() == 0){
            lastname.setError("This field is required");
            status = false;
        }
        if(email.length() == 0 || !email.getText().toString().contains("@") || !email.getText().toString().contains(".com")){
            email.setError("This field is required and required a valid email with an @ symbol and .com");
            status = false;
        }
        if(password.length() == 0){
            password.setError("This field is required");
            status = false;
        }
        if(address.length() == 0){
            address.setError("This field is required");
            status = false;
        }
        if(description.length() == 0){
            description.setError("This field is required");
            status = false;
        }


         return status;
     }

     public static boolean checkEmailFormat(String email){
         if(email.length() == 0 || !email.contains("@") || !email.contains(".com")){
             return false;
         }
         return true;
     }
```

Complaints.java

```java
package com.example.mealerapp;

public class Complaints {
    String cookUID;
    String complaint;
    String clientUID;
    Boolean read;

    public Complaints(String complaint, String clientUID,String cookUID){
        this.clientUID = clientUID;
        this.complaint = complaint;
        this.cookUID = cookUID;
        this.read = false;
    }

    public Complaints(String complaint, String clientUID,String cookUID,Boolean read){
        this.clientUID = clientUID;
        this.complaint = complaint;
        this.cookUID = cookUID;
        this.read = read;
    }

    public String getComplaint() {
        return complaint;
    }

    public void setComplaint(String complaint) {
        this.complaint = complaint;
    }

    public String getClientUID() {
        return clientUID;

     public void setClientUID(String clientUID) {
         this.clientUID = clientUID;
     }

     public String getCookUID() {
         return cookUID;
     }

     public void setCookUID(String cookUID) {
         this.cookUID = cookUID;
     }

     public void setRead(Boolean read){this.read = read;}

     public Boolean getRead(){return read;}

}
```

ComplaintsDatabase.java

```
package com.example.mealerapp;
import android.util.Log;

import androidx.annotation.NonNull;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.lang.reflect.Array;
import java.util.ArrayList;

public class ComplaintsDataBase extends Database implements Database.retrieveListener{

    FirebaseDatabase database;
    DatabaseReference complaintsRef;

    public ComplaintsDataBase(){
        database = FirebaseDatabase.getInstance();
        complaintsRef = database.getReference("COMPLAINTS");
    }

   public void addComplaint(String cookUID, Complaints complaint){
       setInformation(complaintsRef.child(cookUID), complaint);
   }
   public void deleteComplaint(String cookUID){
       deleteInformation(complaintsRef.child(cookUID));
   }

   public void setRead(String cookUID){
       complaintsRef.child(cookUID).child("read").setValue(true);
   }

   @Override
   public void onDataReceived(Object data) {
   }

   @Override
   public void onError() {
   }
}
```

Cook.java

```java
package com.example.mealerapp;

import java.io.Serializable;

public class Cook extends User implements Serializable {

    private String description, suspensionDate;
    private boolean isSuspended;
    private Menu menu;
    private int requestsFulfilled;

    public Cook(String firstName, String lastName, String emailAddress, String accountPassword, String address, String description) {
        super(firstName, lastName, emailAddress, accountPassword, userType.COOK, address);
        this.description = description;
        suspensionDate = "N/A";
        isSuspended = false;
        menu = new Menu();
        this.requestsFulfilled = 0;
    }

    public Cook(String firstName, String lastName, String emailAddress, String accountPassword, String address, String description, Boolean isSuspended, String suspensionDate) {
        super(firstName, lastName, emailAddress, accountPassword, userType.COOK, address);
        this.description = description;
        this.suspensionDate = suspensionDate;
        this.isSuspended = isSuspended;
        menu = new Menu();

    }

    public String getSuspensionDate() {
        return suspensionDate;
    }

    public void setSuspensionDate(String suspensionDate) {
        this.suspensionDate = suspensionDate;
    }

   public boolean isSuspended() {
       return isSuspended;
   }

   public void setSuspended(boolean suspended) {
       isSuspended = suspended;
   }

   public void registerCook(){

       UserDatabase dtb = new UserDatabase();
       dtb.registerUser(this);
   }

   public String getDescription() {
       return description;
   }

   public void setDescription(String newDescription) {
       description = newDescription;
   }

    public String toString() {
    return "\nAccount Information \n************************ \nFirst name: " + firstName + "\n" + "Last name: " + lastName + "\n" + "Email: " + email + "\n" + "Password: " + passw
    }

   public int getRequestsFulfilled() {
       return requestsFulfilled;
   }

   public void setRequestsFulfilled(int requestsFulfilled) {
       this.requestsFulfilled = requestsFulfilled;
   }
}
```

CookRegistration.java

```
package com.example.mealerapp;

import androidx.appcompat.app.AppCompatActivity;

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

public class CookRegistration extends AppCompatActivity {

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


        //Cook newCook = new Cook(firstname.getText().toString(), lastname.getText().toString(), email.getText().toString(), password.getText().toString(), description.getText().to


    }

    public void createAccount(View view){
        EditText firstname = (EditText) findViewById(R.id.firstname);
        EditText lastname = (EditText) findViewById(R.id.lastname);
        EditText email = (EditText) findViewById(R.id.email);
        EditText password = (EditText) findViewById(R.id.password);
        EditText address = (EditText) findViewById(R.id.address);
        EditText description = (EditText) findViewById(R.id.description);
        //EditText p = (EditText) findViewById(R.id.firstname);

        Button createaccount = (Button) findViewById(R.id.createaccount);


            createaccount.setOnClickListener(new View.OnClickListener(){
                public void onClick(View A){
                    String firstnameone = firstname.getText().toString();
                    String lastnameone = lastname.getText().toString();
                    String emailone = email.getText().toString();
                    String passwordone = password.getText().toString();
                    String addressone = address.getText().toString();
                    String descriptionone = description.getText().toString();

                    if(checkFields() && checkEmailFormat(emailone)) {
                        Cook newCook = new Cook(firstnameone, lastnameone, emailone, passwordone, addressone, descriptionone);
                        newCook.registerCook();
                        //Database dbt = new Databse();
                        //dtb.registerUser(newCook);

                        // dont forget to change main act to page

                        Intent intent = new Intent(getApplicationContext(), WelcomePage.class);

                        //Intent intent = new Intent(CookRegistration.this, WelcomePage.class);
                        intent.putExtra("role", "COOK");

                        startActivity(intent);
                    }
                }
            });
```

```
}

private boolean checkFields(){
    boolean status = true;

    EditText firstname = (EditText) findViewById(R.id.firstname);
    EditText lastname = (EditText) findViewById(R.id.lastname);
    EditText email = (EditText) findViewById(R.id.email);
    EditText password = (EditText) findViewById(R.id.password);
    EditText address = (EditText) findViewById(R.id.address);
    EditText description = (EditText) findViewById(R.id.description);

    if(firstname.length() == 0){
        firstname.setError("This field is required");
        status = false;
    }
    if(lastname.length() == 0){
        lastname.setError("This field is required");
        status = false;
    }
    if(email.length() == 0 || !email.getText().toString().contains("@") || !email.getText().toString().contains(".com")){
        email.setError("This field is required and required a valid email with an @ symbol and a .com");
        status = false;
    }
    if(password.length() == 0){
        password.setError("This field is required");
        status = false;
    }
    if(address.length() == 0){
        address.setError("This field is required");
        status = false;
    }
    if(description.length() == 0){
        description.setError("This field is required");
        status = false;
    }


        return status;
     }

     public static boolean checkEmailFormat(String email){
         if(email.length() == 0 || !email.contains("@") || !email.contains(".com")){
             return false;
         }
         return true;
     }


}
```

CreateComplaint.java

```
package com.example.mealerapp;

import static com.example.mealerapp.UserDatabase.getUID;

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

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

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

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

import android.widget.AdapterView;
import android.widget.TextView;
import android.widget.Toast;

public class CreateComplaint extends AppCompatActivity {
```

```
Spinner spinner;
//Spinner spinner2;
DatabaseReference databaseReference;
List<String> cookNames;
//bject item;
List<String> cookUIDs;
//List<String> clientNames;

private Button welcomeButton;



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

    spinner = findViewById(R.id.spinner);
    cookNames = new ArrayList<>();
    cookUIDs = new ArrayList<>();

    //spinner2 = findViewById(R.id.spinner2);
    //clientNames = new ArrayList<>();

    welcomeButton = (Button)findViewById(R.id.welcomePageButton);
    welcomeButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            welcome(v);
        }
    });

    databaseReference = FirebaseDatabase.getInstance().getReference();
    databaseReference.child("USERS").addValueEventListener((new ValueEventListener() {
```

```
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {

        cookNames.add("Choose a cook");
        //clientNames.add("Choose your email");

        for(DataSnapshot childSnapshot:snapshot.getChildren()) {
            String spinnerName = childSnapshot.child("email").getValue(String.class);
            //String cookUID = childSnapshot.getValue(String.class);


            if (childSnapshot.child("role").getValue(String.class).equals("COOK")) {
                cookNames.add(spinnerName);
            }


            //cookNames.add(spinnerName);

        }

        ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(CreateComplaint.this, android.R.layout.simple_spinner_item, cookNames);
        arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_item);
        spinner.setAdapter(arrayAdapter);

    }

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

    }
}));
```

```
}

public void createComplaint(View view){

    Spinner mySpinner = (Spinner) findViewById(R.id.spinner);
    String text = mySpinner.getSelectedItem().toString();



    EditText complaint = (EditText) findViewById(R.id.Complaint);
    EditText clientEmail = (EditText) findViewById(R.id.clientEmail);

    Button create = (Button) findViewById(R.id.CreateComplaint);

    create.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View view) {
            String complaintText = complaint.getText().toString();
            String clientEmailOne = clientEmail.getText().toString();

            if(checkComplaint()){


  /**

  int count = 0;
  int count2 = 0;

  for(int i = 0; i < cookNames.size(); i++){
      if(cookNames.equals(text)){
          count = i;
      }
      if(cookNames.equals(clientEmailOne)){
          count2 = i;
      }

  }
  */

  //String cookID = text.getUID().getText().toString();
  //String clientID = cookUIDs.get(count2 -1);
```

```
                ComplaintsDataBase dtb = new ComplaintsDataBase();

                String clientUID = getUID();

                Database.retrieveListener cookListener = new Database.retrieveListener() {
                    @Override
                    public void onDataReceived(Object data) {
                        String cookUID = (String)data;
                        Complaints newComplaint = new Complaints(complaintText, clientUID , cookUID);
                        dtb.addComplaint(cookUID, newComplaint);

                        Intent intent = new Intent(getApplicationContext(), WelcomePage.class);
                        intent.putExtra("role", "CLIENT");
                        Toast.makeText(getApplicationContext(),"Complaint Submitted",Toast.LENGTH_LONG).show();
                        startActivity(intent);
                    }

                    @Override
                    public void onError() {

                    }
                };

                getEmailUID(text, cookListener);
            }



        }

    });
}


private void getEmailUID(String emailText, Database.retrieveListener listener){

    String emailUID;
```

```
    /**
    database = FirebaseDatabase.getInstance().getReference();
    database.child("USERS").addValueEventListener((new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {

            for(DataSnapshot childSnapshot:snapshot.getChildren()) {
                String spinnerName = childSnapshot.child("email").getValue(String.class);
                //String cookUID = childSnapshot.getValue(String.class);

                if(spinnerName.equals(emailText)){
                    emailUID = childSnapshot.getValue(String.class);
                }

                //String userType = snapshot.child("role").getValue(String.class);
                /**
                 if (userType == "COOK") {
                 cookNames.add(spinnerName);
                 }
                 */



            // }
        // }


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

        //}
    //}));

    FirebaseDatabase.getInstance().getReference().child("USERS")
            .addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {
                    for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
                        String user = snapshot.child("email").getValue().toString();
                        //System.out.println(user.email);

                        if(user.equals(emailText)){
                           listener.onDataReceived(snapshot.getKey());
                           break;
                        }


                    }


                }
                @Override
                public void onCancelled(DatabaseError databaseError) {
                }
            });


}

public void welcome(View view){
    Intent returnToWelcome = new Intent(getApplicationContext(),WelcomePage.class);
    returnToWelcome.putExtra("role","CLIENT");
    startActivity(returnToWelcome);
}

//data validation
private boolean checkComplaint(){
    boolean status = true;
```

```
        Spinner spin = (Spinner) findViewById(R.id.spinner);
        String text = spin.getSelectedItem().toString();

        EditText complaint = (EditText) findViewById(R.id.Complaint);
        EditText email = (EditText) findViewById(R.id.clientEmail);


        if(complaint.length() == 0){
            complaint.setError("This field is required");
            status = false;
        }

        if(email.length() == 0 || !email.getText().toString().contains("@") || !email.getText().toString().contains(".com")){
            email.setError("This field is required and required a valid email with an @ symbol and .com");
            status = false;
        }

        if(text.equals("Choose a cook")){
            //spin.setError("Required");
            ((TextView)spin.getSelectedView()).setError("Error message");
        }

        return status;
    }


}
```

Database.java

```
package com.example.mealerapp;

import android.util.Log;

import androidx.annotation.NonNull;


import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.EventListener;
import java.util.Locale;

abstract public class Database {

    private FirebaseDatabase database;

    public Database(){
        database = FirebaseDatabase.getInstance();
    }

    public void setInformation(DatabaseReference reference, Object information){

        reference.setValue(information);

    }
```

```
    public void deleteInformation(DatabaseReference reference){

        reference.removeValue();
    }

    public void getInformation(DatabaseReference reference, final Database.retrieveListener listener){
        reference.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                String data = snapshot.getValue().toString();
                listener.onDataReceived(data);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                listener.onError();
            }
        });
    }

    public interface retrieveListener{

        void onDataReceived(Object data);
        void onError();
    }


}
```

EditMealPage.java

```java
package com.example.mealerapp;

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.CompoundButton;
import android.widget.EditText;
import android.widget.Switch;
import android.widget.Toast;

public class EditMealPage extends AppCompatActivity {

    private String name, mealType, description,cuisine,allergens,ingredients, UID;
    private boolean currentlyOffered;
    private Double price, rating;
    private Menu menu;

    private EditText nameEditText,mealTypeEditText,EditText,cuisineEditText,allergensEditText,ingredientsEditText,priceEditText, descriptionEditText;
    private Switch currentlyOfferedSwitch;
    private Button saveEditButton;
    private Button deleteMealButton;

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

        menu = new Menu();

        Bundle bundle = getIntent().getExtras();
        name = bundle.getString("name");
        mealType = bundle.getString("mealType");
```

```
description = bundle.getString("description");
cuisine = bundle.getString("cuisine");
allergens = bundle.getString("allergens");
ingredients = bundle.getString("ingredients");
currentlyOffered = Boolean.valueOf(bundle.getString("currentlyOffered"));
price = Double.parseDouble(bundle.getString("price"));
UID = bundle.getString("UID");
rating = Double.parseDouble(bundle.getString("rating"));


nameEditText = (EditText)findViewById(R.id.nameEditText2);
mealTypeEditText = (EditText)findViewById(R.id.mealTypeEditText2);
descriptionEditText = (EditText)findViewById(R.id.descriptionEditText2);
cuisineEditText = (EditText)findViewById(R.id.cuisineEditText2);
allergensEditText = (EditText)findViewById(R.id.allergensEditText2);
ingredientsEditText = (EditText)findViewById(R.id.ingredientsEditText2);
priceEditText = (EditText)findViewById(R.id.priceNumberEdit2);
currentlyOfferedSwitch = (Switch)findViewById(R.id.currentlyOfferedSwitch2);

nameEditText.setText(name);
mealTypeEditText.setText(mealType);
descriptionEditText.setText(description);
cuisineEditText.setText(cuisine);
allergensEditText.setText(allergens);
ingredientsEditText.setText(ingredients);
priceEditText.setText(String.valueOf(price));
currentlyOfferedSwitch.setChecked(currentlyOffered);
currentlyOfferedSwitch.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        currentlyOffered = isChecked;
    }
});
saveEditButton = (Button)findViewById(R.id.saveEditsButton);
saveEditButton.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
            public void onClick(View v) {
                editMeal(v);
                returnToMenuPage(v);
            }
        });
        deleteMealButton = (Button)findViewById(R.id.deleteMealButton);
        deleteMealButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(currentlyOffered==false){
                    menu.deleteMeal(UID,name);
                    returnToMenuPage(v);

                }
                else{
                    Toast.makeText(getApplicationContext(), "cannot delete meal since it's currently offered", Toast.LENGTH_LONG).show();

                }

            }
        });


    }

    private void editMeal(View view){
        menu.deleteMeal(UID, name);
        String mealName = nameEditText.getText().toString();
        String descriptionString = descriptionEditText.getText().toString();
        String allergensString = allergensEditText.getText().toString();
        String ingredientsString = ingredientsEditText.getText().toString();
        String cuisineString = cuisineEditText.getText().toString();
        String mealTypeString = mealTypeEditText.getText().toString();
        Double priceDouble = Double.parseDouble(priceEditText.getText().toString());

        Meal editedMeal = new Meal(mealName,mealTypeString,cuisineString,ingredientsString,allergensString
                ,descriptionString,priceDouble,currentlyOffered,rating);
        menu.addMeal(UID, editedMeal);

    }
    private void returnToMenuPage(View view){
        Intent returnToMenu = new Intent(getApplicationContext(),MenuPage.class);
        returnToMenu.putExtra("UID",UID);
        startActivity(returnToMenu);
    }
}
```

HandleComplaintPage.java

```java
package com.example.mealerapp;

import androidx.appcompat.app.AppCompatActivity;

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

public class HandleComplaintPage extends AppCompatActivity {

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

        Bundle bundle = getIntent().getExtras();
        String cookUID = bundle.getString("cookUID");
        String clientUID = bundle.getString("clientUID");
        String complaint = bundle.getString("complaint");

        TextView cookTextView = (TextView) findViewById(R.id.cookUIDTextView);
        TextView clientTextView = (TextView) findViewById(R.id.clientUIDTextView);
        TextView complaintTextView = (TextView) findViewById(R.id.complaintTextView);

        cookTextView.setText("Cook: "+cookUID);
        clientTextView.setText("Client: "+clientUID);
        complaintTextView.setText("Complaint: "+complaint);

        Button dismissButton = (Button) findViewById(R.id.dismissButton);
        Button suspendButton = (Button) findViewById(R.id.suspendButton);

        dismissButton.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
            public void onClick(View v) {
                ComplaintsDataBase dtb = new ComplaintsDataBase();
                dtb.setRead(cookUID);
                Toast.makeText(getApplicationContext(), "Complaint Dismissed", Toast.LENGTH_LONG).show();
                Intent returnToComplaints = new Intent(getApplicationContext(),SuspensionPage.class);
                startActivity(returnToComplaints);
            }
        });

        suspendButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent suspendCook = new Intent(getApplicationContext(),SuspensionPage.class);
                suspendCook.putExtra("cookUID",cookUID);
                startActivity(suspendCook);
            }
        });

    }
}
```

MainActivity.java

```java
package com.example.mealerapp;

import androidx.appcompat.app.AppCompatActivity;

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

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.FirebaseDatabase;

import java.text.SimpleDateFormat;
import java.time.LocalDate;

public class MainActivity extends AppCompatActivity implements Database.retrieveListener{
    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
    Date today = new Date();


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

        Button login = (Button) findViewById(R.id.login2);
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                loginpage(v);
            }
        });
```

```
}

public void loginpage(View view){

    EditText emailEditText = (EditText)findViewById(R.id.username2);
    EditText passwordEditText = (EditText)findViewById(R.id.password2);


    String email = emailEditText.getText().toString();
    String password = passwordEditText.getText().toString();

    UserDatabase dtb = new UserDatabase();
    dtb.login(email,password);
    Intent intent = new Intent(getApplicationContext(), WelcomePage.class);

    Database.retrieveListener roleListener = new Database.retrieveListener() {
        @Override
        public void onDataReceived(Object data) {
            String dataString = data.toString();
            if(dataString.equals("COOK")){
                Database.retrieveListener suspendedListener = new Database.retrieveListener() {
                    @Override
                    public void onDataReceived(Object data) {
                        Boolean isSuspended = Boolean.valueOf(data.toString());
                        if(isSuspended){
                            Database.retrieveListener dateListener = new Database.retrieveListener() {
                                @Override
                                public void onDataReceived(Object data) {
                                    if(isDatePassed(data.toString())){
                                        dtb.liftSuspension();
                                        Intent intent = new Intent(getApplicationContext(), WelcomePage.class);
                                        intent.putExtra("role",dataString);
                                        startActivity(intent);
                                    }
                                    else {
                                        Intent intent = new Intent(getApplicationContext(), SuspendedCookPage.class);
```

```
                                intent.putExtra("date", data.toString());
                                startActivity(intent);
                            }
                        }

                        @Override
                        public void onError() {

                        }
                    };
                    dtb.retrieveInfo(UserDatabase.dataField.SUSPENSIONDATE, dateListener);
                }
            }

            @Override
            public void onError() {

            }
        };
        dtb.retrieveInfo(UserDatabase.dataField.ISSUSPENDED,suspendedListener);
    }
    Log.d("Here","Here");

    intent.putExtra("role", dataString);
    startActivity(intent);
}

@Override
public void onError() {
    //Toast.makeText(getApplicationContext(), "Invalid Login Credentials", Toast.LENGTH_SHORT);

}
};


dtb.retrieveInfo(UserDatabase.dataField.ROLE,roleListener);
```

```
    }

    /**
     * Helper function that compares dates
     * @param date
     * @return true is the parameter date is before or equal to today and false otherwise
     */
    private boolean isDatePassed(String date){

       Date suspension;

        try{
            suspension = formatter.parse(date);
            if(suspension.compareTo(today)<=0){
                return true;
            }
        }
        catch(Exception e){

        }
        return false;
    }

    public void cookpage(View view){
        Intent intent = new Intent(getApplicationContext(), CookRegistration.class);
        startActivity(intent);
    }

    public void clientpage(View view){
        Intent intent = new Intent(getApplicationContext(), ClientRegistration.class);
        startActivity(intent);
    }

    @Override
    public void onDataReceived(Object data) {
    }

    @Override
    public void onError() {

    }
}
```

Meal.java

```
package com.example.mealerapp;

import java.io.Serializable;

public class Meal implements Serializable {

    private String name, mealType, cuisine, ingredients, allergens, description;
    private double price;
    private Boolean currentlyOffered;
    private double rating;

    public Meal(String name, String mealType, String cuisine,
                    String ingredients, String allergens, String description, double price, boolean currentlyOffered){

        this.name = name;
        this.mealType = mealType;
        this.cuisine = cuisine;
        this.ingredients = ingredients;
        this.allergens = allergens;
        this.description = description;
        this.price = price;
        this.currentlyOffered = currentlyOffered;
        this.rating = -1;

    }
    public Meal(String name, String mealType, String cuisine,
                String ingredients, String allergens, String description, double price, boolean currentlyOffered, double rating){

        this.name = name;
        this.mealType = mealType;
        this.cuisine = cuisine;
        this.ingredients = ingredients;
        this.allergens = allergens;
        this.description = description;
        this.price = price;
        this.currentlyOffered = currentlyOffered;
```

```
        this.rating = rating;
    }

    public String getName() {
        return name;
    }

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

    public String getMealType() {
        return mealType;
    }

    public void setMealType(String mealType) {
        this.mealType = mealType;
    }

    public String getCuisine() {
        return cuisine;
    }

    public void setCuisine(String cuisine) {
        this.cuisine = cuisine;
    }

    public String getIngredients() {
        return ingredients;
    }

    public void setIngredients(String ingredients) {
        this.ingredients = ingredients;
    }

    public String getAllergens() {
```

```
        return allergens;
    }

    public void setAllergens(String allergens) {
        this.allergens = allergens;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public Boolean getCurrentlyOffered() {
        return currentlyOffered;
    }

    public void setCurrentlyOffered(Boolean currentlyOffered) {
        this.currentlyOffered = currentlyOffered;
    }

    public double getRating() {
        return rating;
    }

      public void setRating(double rating) {
          this.rating = rating;
      }
}
```

Menu.java

```java
package com.example.mealerapp;

import com.google.firebase.database.DatabaseReference;

import java.util.LinkedList;
import java.util.List;

public class Menu {
    private MenuDatabase database;


    public Menu(){
        database = new MenuDatabase();
    }

    public void addMeal(String cookUID, Meal meal){
        database.addMeal(cookUID,meal);
    }

    public void deleteMeal(String cookUID, String meal){database.deleteMeal(cookUID, meal);}
}
```

MenuAdapter.java

```java
package com.example.mealerapp;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import org.w3c.dom.Text;

import java.util.ArrayList;

public class MenuAdapter extends RecyclerView.Adapter<MenuAdapter.MenuViewHolder> {

    ArrayList<Meal> list;
    Context context;
    MenuAdapter.RecyclerViewInterface recyclerViewInterface;

    public MenuAdapter (Context context, ArrayList<Meal> list, MenuAdapter.RecyclerViewInterface recyclerViewInterface){
        this.list = list;
        this.context = context;
        this.recyclerViewInterface = recyclerViewInterface;
    }

    @NonNull
    @Override
    public MenuViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(context).inflate(R.layout.meal_item, parent, false);
        return new MenuViewHolder(v,recyclerViewInterface);
    }
```

```java
@Override
public void onBindViewHolder(@NonNull MenuViewHolder holder, int position) {
    Meal meal = list.get(position);
    holder.meal.setText(meal.getName());
    holder.ingredients.setText(meal.getIngredients());
    holder.allergens.setText(meal.getAllergens());
    holder.description.setText(meal.getDescription());
    holder.mealType.setText(meal.getMealType());
    holder.cuisine.setText(meal.getCuisine());
    holder.price.setText("$"+meal.getPrice());
    String currentlyOfferedString = "No";
    if(meal.getCurrentlyOffered()){
        currentlyOfferedString = "Yes";
    }

    holder.currentlyOffered.setText(currentlyOfferedString);

    String ratingString = String.valueOf(meal.getRating());
    if(ratingString.equals("-1.0")){
        ratingString = "No ratings";
    }
    holder.rating.setText(ratingString);
}

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

public static class MenuViewHolder extends RecyclerView.ViewHolder{

    TextView meal, ingredients, allergens, description, cuisine, currentlyOffered, mealType, price, rating;

    public MenuViewHolder(@NonNull View itemView, RecyclerViewInterface recyclerViewInterface) {
        super(itemView);
        meal = itemView.findViewById(R.id.mealNameTextView);
```

```
ingredients = itemView.findViewById(R.id.ingredientsEditTextView);
allergens = itemView.findViewById(R.id.allergensEditTextView);
description = itemView.findViewById(R.id.descriptionEditTextView);
cuisine = itemView.findViewById(R.id.cuisineEditTextView);
currentlyOffered = itemView.findViewById(R.id.currentlyOfferedEditTextView);
mealType = itemView.findViewById(R.id.mealTypeEditTextView);
price = itemView.findViewById(R.id.priceEditTextView);
rating = itemView.findViewById(R.id.ratingEditTextView);
itemView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(recyclerViewInterface!=null) {
            int position = getAdapterPosition();
            String mealString = meal.getText().toString();
            String descriptionString = description.getText().toString();
            String allergensString = allergens.getText().toString();
            String ingredientsString = ingredients.getText().toString();
            String mealTypeString = mealType.getText().toString();
            String cuisineString = cuisine.getText().toString();
            Boolean currentlyOfferedBool = false;
            if(currentlyOffered.getText().toString().equals("Yes")){
                currentlyOfferedBool = true;
            }
            Double priceDouble = Double.parseDouble(price.getText().toString().substring(1));
            double ratingDouble;
            if(rating.getText().toString().equals("No ratings")){
                ratingDouble = -1;
            }
            else{
                ratingDouble = Double.parseDouble(rating.getText().toString());
            }
            if(position!=RecyclerView.NO_POSITION){
                recyclerViewInterface.onItemClick(position,mealString, descriptionString,allergensString,ingredientsString,cuisineString
                ,mealTypeString,currentlyOfferedBool,priceDouble,ratingDouble);
            }
        }

               }
           });
        }

    }

    public interface RecyclerViewInterface{
        public void onItemClick(int position, String meal, String description, String allergens,
                                String ingredients, String cuisine, String mealType, boolean currentlyOffered, Double price, double rating);
    }
}
```

MenuDatabase.java

```java
package com.example.mealerapp;

import android.util.Log;

import androidx.annotation.NonNull;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class MenuDatabase extends Database implements Database.retrieveListener{

    FirebaseDatabase database;
    DatabaseReference reference;

    public MenuDatabase(){
        database = FirebaseDatabase.getInstance();
        reference = database.getReference("USERS");
    }

    public void addMeal(String cookUID, Meal meal){
        reference.child(cookUID).child("MENU").child(meal.getName()).setValue(meal);
    }
    public void deleteMeal(String cookUID, String meal){
        reference.child(cookUID).child("MENU").child(meal).removeValue();
    }

    public void setCurrentlyOffered(String cookUID, Meal meal, boolean isOffered){
        reference.child(cookUID).child("MENU").child(meal.getName()).child("currentlyOffered").setValue(isOffered);
    }
```

```
    public void checkDeleted(String cookUID, String meal, retrieveListener listener){
        reference.child(cookUID).child("MENU").addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                for(DataSnapshot dataSnapshot : snapshot.getChildren()){
                    if(dataSnapshot.getValue().toString().equals(meal)){
                        listener.onDataReceived(false);
                    }
                    else{
                        listener.onDataReceived(true);
                    }
                }
            }

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

            }
        });
    }


    @Override
    public void onDataReceived(Object data) {

    }

    @Override
    public void onError() {

    }
}
```

MenuPage.java

```java
package com.example.mealerapp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

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

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class MenuPage extends AppCompatActivity implements MenuAdapter.RecyclerViewInterface{

    private String UID;
    private RecyclerView recyclerView;

    private MenuAdapter adapter;
    private ArrayList<Meal> list;
    private DatabaseReference ref, ratingRef;

    private Button welcomeButton, addMealButton;
    private TextView emptyMenu;
    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menu_page);

    Bundle bundle = getIntent().getExtras();
    UID = bundle.getString("UID");
    Log.d("UID MENU",UID);

    ref = FirebaseDatabase.getInstance().getReference("USERS").child(UID).child("MENU");
    ratingRef = FirebaseDatabase.getInstance().getReference("USERS").child(UID).child("RATING");

    addMealButton = (Button)findViewById(R.id.addMealButton);
    addMealButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            addMeal(v);
        }
    });

    emptyMenu = (TextView) findViewById(R.id.emptyMenuTextView);
    emptyMenu.setVisibility(View.GONE);

    welcomeButton = (Button)findViewById(R.id.returnWelcomeButton);
    welcomeButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            returnToWelcome(v);
        }
    });

    recyclerView = (RecyclerView) findViewById(R.id.menuRecyclerView);
    recyclerView.setHasFixedSize(true);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
```

```
list = new ArrayList<>();
adapter = new MenuAdapter(this, list,this);

recyclerView.setAdapter(adapter);

ref.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        for(DataSnapshot dataSnapshot: snapshot.getChildren()){
            String[] mealFields = new String[8];
            int counter = 0;
            for(DataSnapshot fields :dataSnapshot.getChildren()){
                if (counter==8){break;}
                mealFields[counter] = fields.getValue().toString();
                counter++;
            }
            String allergens = mealFields[0];
            String cuisine = mealFields[1];
            boolean currentlyOffered = Boolean.parseBoolean(mealFields[2]);
            String description = mealFields[3];
            String ingredients = mealFields[4];
            String mealType = mealFields[5];
            String name = mealFields[6];
            double price = Double.parseDouble(mealFields[7]);

            ratingRef.child(name).addValueEventListener(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    double total = 0;
                    int numRatings = 0;
                    for(DataSnapshot rating : snapshot.getChildren()){
                        total += Double.valueOf(rating.getValue().toString());
                        numRatings++;
                    }
```

```
                        double rating;
                        if(numRatings == 0){
                            rating = -1;
                        }
                        else{
                            rating = Math.round(total/numRatings);
                        }

                        Meal meal = new Meal(name,mealType,cuisine,ingredients,allergens,description,price, currentlyOffered, rating);
                        list.add(meal);
                        adapter.notifyDataSetChanged();

                        if(list.isEmpty()){
                            emptyMenu.setVisibility(View.VISIBLE);
                        }
                    }

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

                    }
                });

            }
        }

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

        }
    });
}


@Override
public void onItemClick(int position, String meal, String description, String allergens,
                        String ingredients, String cuisine, String mealType, boolean currentlyOffered, Double price, double rating) {
    Intent editMealPage = new Intent(getApplicationContext(),EditMealPage.class);
    editMealPage.putExtra("name",meal);
    editMealPage.putExtra("description",description);
    editMealPage.putExtra("allergens",allergens);
    editMealPage.putExtra("ingredients",ingredients);
    editMealPage.putExtra("cuisine",cuisine);
    editMealPage.putExtra("mealType",mealType);
    editMealPage.putExtra("currentlyOffered",String.valueOf(currentlyOffered));
    editMealPage.putExtra("price",String.valueOf(price));
    editMealPage.putExtra("UID",UID);
    editMealPage.putExtra("rating",String.valueOf(rating));
    startActivity(editMealPage);

}

@Override
public void onPointerCaptureChanged(boolean hasCapture) {
    super.onPointerCaptureChanged(hasCapture);
}

private void returnToWelcome(View view){
    Intent returnToWelcome = new Intent(getApplicationContext(),WelcomePage.class);
    returnToWelcome.putExtra("role","COOK");
    startActivity(returnToWelcome);
}
```

```
    private void addMeal(View view){
        Intent addMealPage = new Intent(getApplicationContext(),AddMealPage.class);
        addMealPage.putExtra("UID",UID);
        startActivity(addMealPage);
    }
}
```

Model.java

```
package com.example.mealerapp;

public class Model {

    String complaint, client, cook;

    public Model(String complaint, String cook, String client){
        this.complaint = complaint;
        this.cook = cook;
        this.client = client;
    }

    public String getComplaint() {
        return complaint;
    }

    public String getClient() {
        return client;
    }

    public String getCook() {
        return cook;
    }

}
```

MyAdapter.java

```java
package com.example.mealerapp;

import android.content.Context;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

public class MyAdapter extends RecyclerView.Adapter<MyAdapter.MyViewHolder> {

    ArrayList<Model> mList;
    Context context;
    private final RecyclerViewInterface recyclerViewInterface;


    public MyAdapter(Context context, ArrayList<Model> mList, RecyclerViewInterface recyclerViewInterface) {
        this.mList = mList;
        this.context = context;
        this.recyclerViewInterface = recyclerViewInterface;
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        //return null;
        View v = LayoutInflater.from(context).inflate(R.layout.item, parent, false);
        return new MyViewHolder(v, recyclerViewInterface);

    }
```

```
    }

    @Override
    public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {
        Model model = mList.get(position);
        holder.complaint.setText(model.getComplaint());
        holder.cook.setText(model.getCook());
        holder.client.setText(model.getCook());
    }

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

    public static class MyViewHolder extends RecyclerView.ViewHolder {

        TextView complaint, client, cook;

        public MyViewHolder(@NonNull View itemView, RecyclerViewInterface recyclerViewInterface) {
            super(itemView);

            complaint = itemView.findViewById(R.id.complainttext);
            cook = itemView.findViewById(R.id.cooktext);
            client = itemView.findViewById(R.id.clienttext);

            itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    if (recyclerViewInterface != null) {
                        int position = getAdapterPosition();
                        String cookUID = cook.getText().toString();
                        String clientUID = client.getText().toString();
                        String complaintTxt = complaint.getText().toString();
                        if (position != RecyclerView.NO_POSITION) {
                            recyclerViewInterface.onItemClick(position, cookUID, clientUID, complaintTxt);

                        }
                    }
                });
            }
        }

        public interface RecyclerViewInterface {
            public void onItemClick(int position, String cookUID, String clientUID, String complaint);
        }
    }
```

PurchaseRequest.java

```java
package com.example.mealerapp;

import java.util.Date;
import java.util.Locale;

public class PurchaseRequest {

    public enum STATUS{APPROVED, DENIED, PENDING};

    private String cookUID, clientUID, meal;
    private STATUS status;
    private String date;

    public PurchaseRequest(String cookUID, String clientUID, String meal) {
        this.cookUID = cookUID;
        this.clientUID = clientUID;
        this.meal = meal;
        this.status = STATUS.PENDING;
        Date date = new Date();
        this.date = date.toString();
    }

    public PurchaseRequest(String cookUID, String clientUID, String meal, STATUS status, String date) {
        this.cookUID = cookUID;
        this.clientUID = clientUID;
        this.meal = meal;
        this.status = status;
        this.date = date;
    }

    public String getCookUID() {
        return cookUID;
    }

    public void setCookUID(String cookUID) {
        this.cookUID = cookUID;
```

```
public String getClientUID() {
    return clientUID;
}

public void setClientUID(String clientUID) {
    this.clientUID = clientUID;
}

public String getMeal() {
    return meal;
}

public void setMeal(String meal) {
    this.meal = meal;
}

public STATUS getStatus() {
    return status;
}

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

public static STATUS stringToStatus(String statusString){
    String status = statusString.toLowerCase();
    switch (status){
        case "approved": return STATUS.APPROVED;
        case "pending": return STATUS.PENDING;
        case "denied": return STATUS.DENIED;
    }

    return STATUS.PENDING;
}

public String getDate() {
    return date;
```

```
    }

    public void setDate(String date) {
        this.date = date;
    }
}
```

PurchaseRequestAdapter.java

```
package com.example.mealerapp;

import android.content.Context;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import org.w3c.dom.Text;

import java.util.ArrayList;
import java.util.Locale;

public class PurchaseRequestAdapter extends RecyclerView.Adapter<PurchaseRequestAdapter.PurchaseRequestViewHolder> {

    String roleString;
   ArrayList<PurchaseRequest> list;
   Context context;
   PurchaseRequestAdapter.RecyclerViewInterface recyclerViewInterface;

    public PurchaseRequestAdapter(ArrayList<PurchaseRequest> list, Context context, PurchaseRequestAdapter.RecyclerViewInterface recyclerViewInterface, String role) {
        this.list = list;
        this.context = context;
        this.recyclerViewInterface = recyclerViewInterface;
        this.roleString = role;
    }
```

```
@NonNull
@Override
public PurchaseRequestViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View v = LayoutInflater.from(context).inflate(R.layout.purchase_request_item,parent,false);
    return new PurchaseRequestViewHolder(v,recyclerViewInterface);

}

@Override
public void onBindViewHolder(@NonNull PurchaseRequestViewHolder holder, int position) {
    PurchaseRequest request = list.get(position);
    holder.meal.setText(request.getMeal());
    holder.status.setText(request.getStatus().toString().toLowerCase());
    holder.date.setText(request.getDate());
    DatabaseReference priceRef = FirebaseDatabase.getInstance().getReference("USERS").child(request.getCookUID()).child("MENU")
                    .child(request.getMeal()).child("price");
    if(roleString.equals("COOK")){
        holder.roleTitle.setText("Client: ");
    }
    else{
        holder.roleTitle.setText("Cook: ");
    }

    priceRef.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            String priceString = snapshot.getValue().toString();
            holder.price.setText("$"+priceString);

            DatabaseReference ref = FirebaseDatabase.getInstance().getReference("USERS").child(request.getCookUID()).child("email");

            if(roleString.equals("COOK")){
                ref = FirebaseDatabase.getInstance().getReference("USERS").child(request.getClientUID()).child("email");
            }
```

```
            ref.addValueEventListener(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    holder.role.setText(snapshot.getValue().toString());
                }

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

                }
            });
        }

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

        }
    });



}

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

public static class PurchaseRequestViewHolder extends RecyclerView.ViewHolder{

    TextView meal,price,role,status,date;
    TextView roleTitle;

    public PurchaseRequestViewHolder(@NonNull View itemView, RecyclerViewInterface recyclerViewInterface) {
        super(itemView);
```

```
            meal = itemView.findViewById(R.id.mealEditTextview);
            price = itemView.findViewById(R.id.priceEditTextview);
            role = itemView.findViewById(R.id.roleEditTextView);
            status = itemView.findViewById(R.id.statusEditTextView);
            date = itemView.findViewById(R.id.dateEditTextView);

            roleTitle = itemView.findViewById(R.id.roleTextView);

            itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    if(roleTitle.getText().toString().equals("Client: ") && status.getText().toString().equals("pending")){
                        if(recyclerViewInterface!=null){
                            int position = getAdapterPosition();

                            String mealString = meal.getText().toString();
                            String priceString =price.getText().toString();
                            String clientString = role.getText().toString();
                            String dateString = date.getText().toString();

                            if(position!=RecyclerView.NO_POSITION){
                                recyclerViewInterface.onItemClick(position,mealString,priceString, clientString, dateString);
                            }

                        }
                    }

                }
            });
        }
    }

    public interface RecyclerViewInterface{
        public void onItemClick(int position,String meal, String priceString, String clientString, String dateString);
    }
}
```

RequestDatabase.java

```java
package com.example.mealerapp;

import android.net.wifi.aware.PublishConfig;
import android.util.Log;

import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.Date;

public class RequestDatabase extends Database{

    private FirebaseDatabase database;
    private DatabaseReference reference;
    private DatabaseReference statusRef;
    private boolean stop = false;

    public RequestDatabase(){
        database = FirebaseDatabase.getInstance();
        reference = FirebaseDatabase.getInstance().getReference("USERS");
        statusRef = FirebaseDatabase.getInstance().getReference("STATUS");


    }

    public void addRequest(PurchaseRequest request){
        if(!stop) {
            String requestID = request.getClientUID() + request.getDate();
            reference.child(request.getCookUID()).child("REQUESTS").child(requestID).setValue(request);
            reference.child(request.getClientUID()).child("REQUESTS").child(requestID).setValue(request);
            stop = true;
        }
    }

    public void deleteRequest(PurchaseRequest request) {

    }

    public void setAccepted(PurchaseRequest request){

        Log.d("setAccepted","called");
        String requestID = request.getClientUID() + request.getDate();
        reference.child(request.getCookUID()).child("REQUESTS").child(requestID).child("status").setValue("APPROVED");
        reference.child(request.getClientUID()).child("REQUESTS").child(requestID).child("status").setValue("APPROVED");

    }
    public void setRejected(PurchaseRequest request){
        String requestID = request.getClientUID() + request.getDate();
        reference.child(request.getCookUID()).child("REQUESTS").child(requestID).child("status").setValue("DENIED");
        reference.child(request.getClientUID()).child("REQUESTS").child(requestID).child("status").setValue("DENIED");

    }
}
```

SearchMeals.java

```java
package com.example.mealerapp;

import androidx.activity.result.contract.ActivityResultContracts;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

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

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class SearchMeals extends AppCompatActivity implements SearchMealsAdapter.RecyclerViewInterface{

    private SearchMealsAdapter adapter;
    private ArrayList<SearchableMeal> list, searchResults;
    private RecyclerView recyclerView;
    private SearchView searchView;
    private Button welcome;

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

```
DatabaseReference userRef = FirebaseDatabase.getInstance().getReference("USERS");

welcome = (Button)findViewById(R.id.welcomeButton);
welcome.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        returnToWelcome(v);
    }
});

list = new ArrayList<>();
searchResults = new ArrayList<>();
adapter = new SearchMealsAdapter(this, searchResults,this);
recyclerView = (RecyclerView) findViewById(R.id.searchMealsRecyclerView);
recyclerView.setHasFixedSize(true);
recyclerView.setLayoutManager(new LinearLayoutManager(this));
recyclerView.setAdapter(adapter);

userRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        for(DataSnapshot userSnapshot: snapshot.getChildren()){
            String role = userSnapshot.child("role").getValue().toString();
            if(role.equals("COOK")){
                boolean isSuspended = Boolean.valueOf(userSnapshot.child("suspended").getValue().toString());
                if(!isSuspended){
                    DatabaseReference menuRef = userSnapshot.child("MENU").getRef();
                    menuRef.addValueEventListener(new ValueEventListener() {
                        @Override
                        public void onDataChange(@NonNull DataSnapshot snapshot) {
                            for(DataSnapshot mealSnapshot : snapshot.getChildren()){
                                String[] mealFields = new String[8];
                                int counter = 0;
                                for(DataSnapshot mealInfoSnapshot: mealSnapshot.getChildren()){
                                    if (counter==8){break;}
                                    mealFields[counter] = mealInfoSnapshot.getValue().toString();
                                    counter++;
```

```
    counter++;

}

String allergens = mealFields[0];
String cuisine = mealFields[1];
boolean currentlyOffered = Boolean.parseBoolean(mealFields[2]);
String description = mealFields[3];
String ingredients = mealFields[4];
String mealType = mealFields[5];
String name = mealFields[6];
double price = Double.parseDouble(mealFields[7]);

DatabaseReference ratingRef = userSnapshot.getRef().child("RATING");

ratingRef.child(name).addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        double total = 0;
        int numRatings = 0;
        for(DataSnapshot rating : snapshot.getChildren()){
            total += Double.valueOf(rating.getValue().toString());
            numRatings++;
        }

        double rating;
        if(numRatings == 0){
            rating = -1;
        }
        else{
            rating = Math.round(total/numRatings);
        }

        SearchableMeal meal = new SearchableMeal(name,mealType,cuisine,ingredients,allergens,description,price, currentlyOffered, rating, us
        list.add(meal);
```

```
                                    if(list.isEmpty()){
                                        //Set a textview to visible that says no search results
                                    }


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

                                }
                            });

                        }

                    }

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

                    }
                });
            }
        }
    }

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

    }
});

searchView = (SearchView)findViewById(R.id.mealSearchView);
searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
    @Override
    public boolean onQueryTextSubmit(String query) {

      if(!searchResults.isEmpty()){
          for(SearchableMeal searchResult: searchResults){
              searchResults.remove(searchResult);
          }
      }
      for(SearchableMeal mealItem:list){
          if(mealItem.getName().contains(query)){
              searchResults.add(mealItem);
              adapter.notifyDataSetChanged();
          }
      }

      return false;
    }

    @Override
    public boolean onQueryTextChange(String newText) {
        return false;
    }
});
}

@Override
public void onItemClick(int position, String meal, String description, String allergens, String ingredients, String cuisine, String mealType, boolean currentlyOffered, Double p
    Intent purchaseRequest = new Intent(getApplicationContext(),ViewMealInfo.class);
    Meal purchaseMeal = new SearchableMeal(meal,mealType,cuisine,ingredients,allergens,description,price,true,cook);
    purchaseRequest.putExtra("meal",purchaseMeal);
    startActivity(purchaseRequest);
}

private void returnToWelcome(View view){
    Intent returnToWelcome = new Intent(getApplicationContext(),WelcomePage.class);
    returnToWelcome.putExtra("role","CLIENT");
    startActivity(returnToWelcome);
}
```

SearchMealsAdapter.java

```
package com.example.mealerapp;

import android.content.Context;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class SearchMealsAdapter extends RecyclerView.Adapter<SearchMealsAdapter.SearchMealViewHolder>{
    ArrayList<SearchableMeal> list;
    Context context;
    SearchMealsAdapter.RecyclerViewInterface recyclerViewInterface;

    public SearchMealsAdapter (Context context, ArrayList<SearchableMeal> list, SearchMealsAdapter.RecyclerViewInterface recyclerViewInterface){
        this.list = list;
        this.context = context;
        this.recyclerViewInterface = recyclerViewInterface;
    }

    @NonNull
    @Override
    public SearchMealsAdapter.SearchMealViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(context).inflate(R.layout.search_meal_item, parent, false);
```

```
    return new SearchMealsAdapter.SearchMealViewHolder(v,recyclerViewInterface);
}

@Override
public void onBindViewHolder(@NonNull SearchMealsAdapter.SearchMealViewHolder holder, int position) {
    SearchableMeal meal = list.get(position);
    holder.meal.setText(meal.getName());
    holder.ingredients.setText(meal.getIngredients());
    holder.allergens.setText(meal.getAllergens());
    holder.description.setText(meal.getDescription());
    holder.mealType.setText(meal.getMealType());
    holder.cuisine.setText(meal.getCuisine());
    holder.price.setText("$"+meal.getPrice());
    String currentlyOfferedString = "No";
    if(meal.getCurrentlyOffered()){
        currentlyOfferedString = "Yes";
    }

    holder.currentlyOffered.setText(currentlyOfferedString);

    String ratingString = String.valueOf(meal.getRating());
    if(ratingString.equals("-1.0")){
        ratingString = "No ratings";
    }
    holder.rating.setText(ratingString);

   holder.cook.setText(meal.getCook());
}

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

public static class SearchMealViewHolder extends RecyclerView.ViewHolder{
```

```
TextView meal, ingredients, allergens, description, cuisine, currentlyOffered, mealType, price, rating,cook;

public SearchMealViewHolder(@NonNull View itemView, SearchMealsAdapter.RecyclerViewInterface recyclerViewInterface) {
    super(itemView);
    meal = itemView.findViewById(R.id.mealNameTextView);
    ingredients = itemView.findViewById(R.id.ingredientsEditTextView);
    allergens = itemView.findViewById(R.id.allergensEditTextView);
    description = itemView.findViewById(R.id.descriptionEditTextView);
    cuisine = itemView.findViewById(R.id.cuisineEditTextView);
    currentlyOffered = itemView.findViewById(R.id.currentlyOfferedEditTextView);
    mealType = itemView.findViewById(R.id.mealTypeEditTextView);
    price = itemView.findViewById(R.id.priceEditTextView);
    rating = itemView.findViewById(R.id.ratingEditTextView);
    cook = itemView.findViewById(R.id.cookNameEditTextView);
    itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(recyclerViewInterface!=null) {
                int position = getAdapterPosition();
                String mealString = meal.getText().toString();
                String descriptionString = description.getText().toString();
                String allergensString = allergens.getText().toString();
                String ingredientsString = ingredients.getText().toString();
                String mealTypeString = mealType.getText().toString();
                String cuisineString = cuisine.getText().toString();
                String cookString = cook.getText().toString();
                Boolean currentlyOfferedBool = false;
                if(currentlyOffered.getText().toString().equals("Yes")){
                    currentlyOfferedBool = true;
                }
                Double priceDouble = Double.parseDouble(price.getText().toString().substring(1));
                double ratingDouble;
                if(rating.getText().toString().equals("No ratings")){
                    ratingDouble = -1;
                }
                else{
                    ratingDouble = Double.parseDouble(rating.getText().toString());

                    }
                    if(position!=RecyclerView.NO_POSITION){
                        recyclerViewInterface.onItemClick(position,mealString, descriptionString,allergensString,ingredientsString,cuisineString
                                ,mealTypeString,currentlyOfferedBool,priceDouble,ratingDouble,cookString);
                    }
                }
            }
        });
    }

}

public interface RecyclerViewInterface{
    public void onItemClick(int position, String meal, String description, String allergens,
                            String ingredients, String cuisine, String mealType, boolean currentlyOffered, Double price, double rating, String cook);
}
}
```

SearchableMeal.java

```java
package com.example.mealerapp;

public class SearchableMeal extends Meal{
    private String cook;

    public SearchableMeal(String name, String mealType, String cuisine, String ingredients, String allergens, String description, double price, boolean currentlyOffered,String cook
        super(name, mealType, cuisine, ingredients, allergens, description, price, currentlyOffered);
        this.cook = cook;
    }

    public SearchableMeal(String name, String mealType, String cuisine, String ingredients, String allergens, String description, double price, boolean currentlyOffered, double rat
        super(name, mealType, cuisine, ingredients, allergens, description, price, currentlyOffered, rating);
        this.cook = cook;
    }

    public String getCook() {
        return cook;
    }

    public void setCook(String cook) {
        this.cook = cook;
    }
`
```

SuspendedCookPage.java

```java
package com.example.mealerapp;

import androidx.appcompat.app.AppCompatActivity;

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

public class SuspendedCookPage extends AppCompatActivity {

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

        Bundle bundle = getIntent().getExtras();
        String suspensionDate = bundle.getString("date");

        TextView suspensionText = (TextView) findViewById(R.id.suspendedTextView);
        suspensionText.setText("You are suspended until "+suspensionDate);

        Button logoff = (Button) findViewById(R.id.logoffButton);

        logoff.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), MainActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

SuspensionPage.java

```java
package com.example.mealerapp;

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.EditText;
import android.widget.Toast;


public class SuspensionPage extends AppCompatActivity implements View.OnClickListener {

    private String cookUID;
    private String date;
    private EditText suspensionDate;

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

        Bundle bundle = getIntent().getExtras();
        cookUID = bundle.getString("cookUID");

        suspensionDate = (EditText) findViewById(R.id.suspensionDateEditText);

        Button permanent = findViewById(R.id.permanentButton);
        permanent.setOnClickListener(this);
        Button temporary = findViewById(R.id.temporaryButton);
        temporary.setOnClickListener(this);
```

```
}

@Override
public void onClick(View view) {
    UserDatabase dtb = new UserDatabase();
    ComplaintsDataBase database = new ComplaintsDataBase();
    Intent returnToComplaints = new Intent(getApplicationContext(), ComplaintsPage.class);
    switch (view.getId()) {
        case R.id.temporaryButton:
            date = suspensionDate.getText().toString();
            boolean valid = verifyDate(date);
            if (!valid) {
                Toast.makeText(getApplicationContext(), "Invalid date", Toast.LENGTH_SHORT).show();
            } else {
                dtb.suspendCook(cookUID, date);
                database.setRead(cookUID);
                startActivity(returnToComplaints);

           }
            break;

    case R.id.permanentButton:
    dtb.suspendCook(cookUID, "Indefinite");
    database.setRead(cookUID);
    startActivity(returnToComplaints);
    break;

}
}
```

```
    //This method verfies that the date input is valid but it is incomplete
    //Needs to check the the date makes sense (ex: cannot put febuary 31)
    //Need to check that the date is not in the past
    //Add functionality that checks the current date to make sure the date input is in the future
    private boolean verifyDate(String date){
        boolean isValid = false;
        int month, day, year;

        if(date.length()<10){
            return isValid;
        }

        try {
            month = Integer.parseInt(date.substring(0, 2));
            day = Integer.parseInt(date.substring(3, 5));
            year = Integer.parseInt(date.substring(6));

        }
        catch(Exception e){
            Log.d("HIT",date);
            return isValid;
        }

        if(month < 13 && month> 0){
            if(day < 32 && day > 0){
                if(year >2021){
                    isValid = true;
                }
            }
        }

        return isValid;

    }
}
```

User.java

```java
package com.example.mealerapp;

import android.net.Uri;

import com.google.firebase.database.Exclude;

public class User {

    protected String firstName, lastName, email, password, address;
    public enum userType {COOK, CLIENT, ADMIN};
    private userType type;

    public User(String fName, String lName, String mail, String pswrd, userType role, String adrs){
        firstName=fName;
        lastName=lName;
        email = mail;
        password = pswrd;
        type = role;
        address = adrs;
    }
    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
```

```
    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public userType getRole(){return type;}

    public void setRole(userType role){type = role;}


}
```

UserDatabase.java

```java
package com.example.mealerapp;

import android.util.Log;

import androidx.annotation.NonNull;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;

import java.util.ArrayList;

public class UserDatabase extends Database{

    private DatabaseReference userReference;
    private FirebaseDatabase database;
    private FirebaseAuth auth;
    public enum dataField{FIRSTNAME,LASTNAME,EMAIL,PASSWORD,ADDRESS,CREDITCARDINFO, DESCRIPTION, ROLE, ISSUSPENDED, SUSPENSIONDATE};

    public UserDatabase(){
        database = FirebaseDatabase.getInstance();
        userReference = database.getReference().child("USERS");
        auth = FirebaseAuth.getInstance();
    }

    public void registerUser(User user){
```

```
    //Create an account using email and password
    auth.createUserWithEmailAndPassword(user.getEmail(),user.getPassword()).addOnSuccessListener(new OnSuccessListener<AuthResult>() {
        @Override

        //If sign up with email and password is successful, stores user information into realtime database
        public void onSuccess(AuthResult authResult) {

            setInformation(userReference.child(auth.getCurrentUser().getUid()), user);

        }
    });

}

public void deleteUser(User user){

    //Delete user information
    deleteInformation(userReference.child(FirebaseAuth.getInstance().getCurrentUser().getUid()));

    //Delete user authentication information
    auth.getCurrentUser().delete();

}

public void retrieveInfo(UserDatabase.dataField field, final UserDatabase.retrieveListener listener){

    DatabaseReference fieldReference = userReference.child(FirebaseAuth.getInstance().getCurrentUser().getUid()).child(fieldToString(field));

    getInformation(fieldReference, listener);

}

//Login user using email and password
public void login(String email, String password){

    auth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
```

```
        @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if(task.isSuccessful()){
                    Log.d("Login Status: ", "Success");
                }
                else{
                    Log.d("Login Status: ", "Fail");
                }
            }
        });

    if(auth.getCurrentUser() == null){
        Log.d("Login Status", "Failed");
    }

        auth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if(task.isSuccessful()){
                    Log.d("Login Status: ", "Success");
                }
                else{
                    Log.d("Login Status: ", "Fail");
                }
            }
        });
}

//Log out current user
public void logoff(){

    if(auth.getCurrentUser()!=null) {
        auth.signOut();
    }

}
```

```
    public void suspendCook(String cookUID, String suspensionDate){
        DatabaseReference cookRef = database.getReference("USERS").child(cookUID);
        cookRef.child("suspended").setValue(true);
        cookRef.child("suspensionDate").setValue(suspensionDate);

    }

    public void liftSuspension(){
        DatabaseReference cookRef = database.getReference("USERS").child(getUID());
        cookRef.child("suspended").setValue(false);
        cookRef.child("suspensionDate").setValue("N/A");
    }

    public static String getUID(){
        return FirebaseAuth.getInstance().getUid().toString();
    }

    //Helper function converts field enum into a properly formatted string
    private String fieldToString(UserDatabase.dataField field){

        String fieldString;

        if(field== UserDatabase.dataField.FIRSTNAME){
            fieldString = "lastName";

        }
        else if (field== UserDatabase.dataField.LASTNAME){
            fieldString = "firstName";
        }
        else if(field== UserDatabase.dataField.CREDITCARDINFO){
            fieldString = "creditCardInfo";
        }
        else if(field == dataField.ISSUSPENDED){
            fieldString = "suspended";
        }
        else if(field == dataField.SUSPENSIONDATE){
            fieldString = "suspensionDate";
          }
          else{
              fieldString = field.toString().toLowerCase();
          }

          return fieldString;

       }
    }
```

ViewCookInfoPage.java

```java
package com.example.mealerapp;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class ViewCookInfoPage extends AppCompatActivity {

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

ViewMealInfo.java

```java
package com.example.mealerapp;

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

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

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.io.Serializable;
import java.util.Date;

public class ViewMealInfo extends AppCompatActivity implements Serializable {

    private String cookUID, clientUID;
    private SearchableMeal meal;
    private TextView name,type,ingredients,allergens,description,price,cuisine, cookTextView;
    private Button requestMeal, cookInfo;
    boolean stop = false;

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

        meal = (SearchableMeal) getIntent().getSerializableExtra("meal");
```

```
        Log.d("pastMeal", "here");
        name = (TextView) findViewById(R.id.NameTextView);
        type = (TextView) findViewById(R.id.MealTypeTextView);
        ingredients = (TextView) findViewById(R.id.IngredientsTextView);
        allergens = (TextView) findViewById(R.id.AllergensTextView);
        description = (TextView) findViewById(R.id.DescriptionTextView);
        price = (TextView) findViewById(R.id.PriceTextView);
        cuisine = (TextView) findViewById(R.id.MealCuisineTextView);
        cookTextView = (TextView)findViewById(R.id.cookNameTextView);

        requestMeal = (Button)findViewById(R.id.requestMealButton);
        requestMeal.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                requestMeal(v);
            }
        });
        cookInfo = (Button)findViewById(R.id.cookInfoButton);
        cookInfo.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                cookInfo(v);
            }
        });

        name.setText("Meal Name: "+meal.getName());
        type.setText("Meal Type: "+meal.getMealType());
        ingredients.setText("Ingredients: "+meal.getIngredients());
        allergens.setText("Allergens: "+meal.getAllergens());
        description.setText("Description: "+meal.getDescription());
        price.setText("Price: $"+String.valueOf(meal.getPrice()));
        cuisine.setText("Cuisine: "+meal.getCuisine());
        cookTextView.setText("Cook email: "+meal.getCook());

    }
```

```
    public void requestMeal(View view){


        clientUID = UserDatabase.getUID();
        DatabaseReference cookRef = FirebaseDatabase.getInstance().getReference("USERS");
        cookRef.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                for(DataSnapshot dataSnapshot: snapshot.getChildren()){
                    if(dataSnapshot.child("email").getValue().toString().equals(meal.getCook())){
                        cookUID = dataSnapshot.getKey();

                    }
                }
                if(!stop){
                    PurchaseRequest request = new PurchaseRequest(cookUID,clientUID,meal.getName());
                    RequestDatabase dtb = new RequestDatabase();
                    dtb.addRequest(request);
                    stop  = true;
                }

            }

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

            }
        });

        Intent returnToSearchMeals = new Intent(getApplicationContext(),SearchMeals.class);
        startActivity(returnToSearchMeals);

    }

    public void cookInfo(View view){
        Intent viewCookInfo = new Intent(getApplicationContext(),ViewCookInfoPage.class);

         viewCookInfo.putExtra("cook",meal.getCook());
         startActivity(viewCookInfo);
     }
}
```

ViewPendingRequest.java

```java
package com.example.mealerapp;

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

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

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.io.Serializable;

public class ViewPendingRequest extends AppCompatActivity{

    private RequestDatabase database;

    private String meal, price, date, client;
    private PurchaseRequest request;
    private Button accept,decline,requests;
    private TextView mealTextView, priceTextView, clientTextView, dateTextView;
```

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


    database = new RequestDatabase();

    Bundle bundle = getIntent().getExtras();
    meal = bundle.getString("meal");
    price = bundle.getString("price");
    date = bundle.getString("date");
    client = bundle.getString("client");


    mealTextView = (TextView) findViewById(R.id.mealTextView);
    priceTextView = (TextView) findViewById(R.id.priceTextView);
    clientTextView = (TextView) findViewById(R.id.clientTextView);
    dateTextView = (TextView) findViewById(R.id.dateTextView);

    mealTextView.setText("Meal: "+meal);
    priceTextView.setText("Price: " + price);
    dateTextView.setText("Date: "+date);
    clientTextView.setText("Client: "+client);

    accept = (Button)findViewById(R.id.acceptButton);
    decline = (Button)findViewById(R.id.declineButton);
    requests = (Button)findViewById(R.id.requestPageButton);

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

```
    decline.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            declineRequest(v);

        }
    });

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


}
public void acceptRequest(View v){


    DatabaseReference cookUID = FirebaseDatabase.getInstance().getReference("USERS");
    cookUID.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            for(DataSnapshot dataSnapshot : snapshot.getChildren()){
                if(dataSnapshot.child("email").getValue().toString().equals(client)){
                    request = new PurchaseRequest(UserDatabase.getUID(),dataSnapshot.getKey(),meal, PurchaseRequest.STATUS.PENDING,date);
                    database.setAccepted(request);
                }
            }
        }

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

        }
```

```
        });


        returnToRequestPage(v);

    }

    public void declineRequest(View v){
        DatabaseReference cookUID = FirebaseDatabase.getInstance().getReference("USERS");
        cookUID.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                for(DataSnapshot dataSnapshot : snapshot.getChildren()){
                    if(dataSnapshot.child("email").getValue().toString().equals(client)){
                        request = new PurchaseRequest(UserDatabase.getUID(),dataSnapshot.getKey(),meal, PurchaseRequest.STATUS.PENDING,date);
                        database.setRejected(request);
                    }
                }
            }

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

            }
        });

        returnToRequestPage(v);

    }

    public void returnToRequestPage(View v){
        Intent returnToRequestPage = new Intent(getApplicationContext(),ViewPurchaseRequests.class);
        returnToRequestPage.putExtra("role","COOK");
        startActivity(returnToRequestPage);
    }

}
```

ViewProfile.java

```java
package com.example.mealerapp;

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

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

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

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

public class ViewProfile extends AppCompatActivity {


    String email = "";
    String fn = "";
    String ln = "";
    String ad = "";
    String des = "";
    String menu = "";
    DatabaseReference databaseReference, dtb;
    Button welcomeButton;
```

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


    welcomeButton = (Button)findViewById(R.id.welcomePage);
    welcomeButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            welcome(v);
        }
    });

    String uid = UserDatabase.getUID();

    databaseReference = FirebaseDatabase.getInstance().getReference();
    databaseReference.child("USERS").child(uid).addValueEventListener((new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            //DataSnapshot childSnapshot = snapshot.getChildren();

            email = snapshot.child("email").getValue(String.class);
            TextView myTextView = findViewById(R.id.email3);
            myTextView.setText(email);


            fn = snapshot.child("firstName").getValue(String.class);
            TextView fnview = findViewById(R.id.firstname3);
            fnview.setText(fn);

            ln = snapshot.child("lastName").getValue(String.class);
            TextView lnview = findViewById(R.id.lastname3);
            lnview.setText(ln);

            ad = snapshot.child("address").getValue(String.class);
            TextView adview = findViewById(R.id.address3);
            adview.setText(ad);

            des = snapshot.child("description").getValue(String.class);
            TextView desview = findViewById(R.id.des3);
            desview.setText(des);


        }

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

        }
    }));
```

```
    }


    public void welcome(View view){
        Intent returnToWelcome = new Intent(getApplicationContext(),WelcomePage.class);
        returnToWelcome.putExtra("role","COOK");
        startActivity(returnToWelcome);
    }



}
```

ViewPurchaseRequests.java

```
package com.example.mealerapp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.io.Serializable;
import java.util.ArrayList;

public class ViewPurchaseRequests extends AppCompatActivity implements PurchaseRequestAdapter.RecyclerViewInterface{

    private RecyclerView recyclerView;
    private TextView noRequests, roleTextView;
    private Button welcomeButton;
    private ArrayList<PurchaseRequest> list;
    private DatabaseReference ref;
    private String UID,role;
    private PurchaseRequestAdapter adapter;
```

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

    Bundle bundle = getIntent().getExtras();
    role = bundle.getString("role");

    UID = UserDatabase.getUID();

    ref = FirebaseDatabase.getInstance().getReference("USERS").child(UID).child("REQUESTS");

    list = new ArrayList<>();
    adapter = new PurchaseRequestAdapter(list,this,this, role);

    recyclerView = (RecyclerView) findViewById(R.id.purchaseRequestsRecycleView);
    recyclerView.setHasFixedSize(true);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
    recyclerView.setAdapter(adapter);

    noRequests = (TextView) findViewById(R.id.noRequestsTextView);
    noRequests.setVisibility(View.GONE);


    welcomeButton = (Button)findViewById(R.id.welcomeButton);
    welcomeButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            returnToWelcome(v);
        }
    });

    ref.addValueEventListener(new ValueEventListener() {
```

```
ref.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        String clientUID, cookUID, mealString, status, date;
        for(DataSnapshot dataSnapshot : snapshot.getChildren()){
            String[] requestFields = new String[5];
            int counter = 0;
            for(DataSnapshot field : dataSnapshot.getChildren()){
                requestFields[counter] = field.getValue().toString();
                counter++;
            }

            clientUID = requestFields[0];
            cookUID = requestFields[1];
            date = requestFields[2];
            mealString = requestFields[3];
            status = requestFields[4];


            PurchaseRequest purchaseRequest = new PurchaseRequest(cookUID,clientUID,mealString,PurchaseRequest.stringToStatus(status),date);
            list.add(purchaseRequest);
            adapter.notifyDataSetChanged();

            if(list.isEmpty()){
                noRequests.setVisibility(View.VISIBLE);
            }
        }

    }

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

    }
});
```

```
    }
    @Override
    public void onPointerCaptureChanged(boolean hasCapture) {
        super.onPointerCaptureChanged(hasCapture);
    }

    public void returnToWelcome(View view){
        Intent returnToWelcome = new Intent(getApplicationContext(),WelcomePage.class);
        returnToWelcome.putExtra("role",role);
        startActivity(returnToWelcome);
    }

    @Override
    public void onItemClick(int position, String meal,String price, String clientString, String dateString) {
       if(list.get(position).getStatus().equals(PurchaseRequest.STATUS.PENDING)){
           Intent viewPendingRequest = new Intent(getApplicationContext(),ViewPendingRequest.class);
           viewPendingRequest.putExtra("client",clientString);
           viewPendingRequest.putExtra("meal",meal);
           viewPendingRequest.putExtra("price",price);
           viewPendingRequest.putExtra("date",dateString);
           startActivity(viewPendingRequest);
       }
       else{
          Toast.makeText(getApplicationContext(),"Cannot edit a non-pending request",Toast.LENGTH_LONG).show();
       }
    }
```

WelcomePage.java

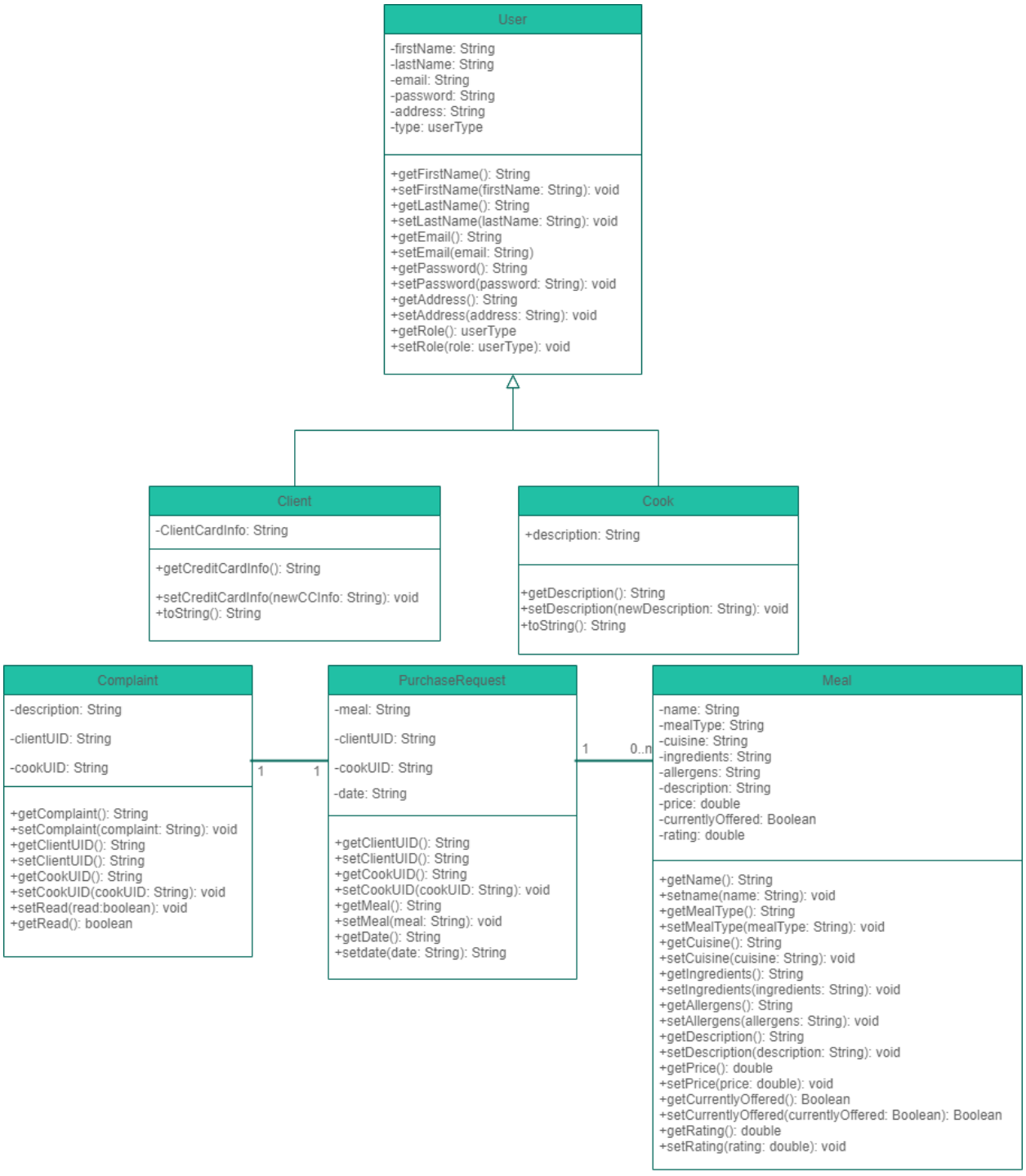
```java
package com.example.mealerapp;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
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.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class WelcomePage extends AppCompatActivity{

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

        Bundle bundle = getIntent().getExtras();
        String role = bundle.getString("role");


        TextView userProfile = (TextView) findViewById(R.id.UserProfile);
        userProfile.setText(role);


        Button logoff = (Button) findViewById(R.id.LogOff);
        logoff.setOnClickListener(new View.OnClickListener() {
```

```
        public void onClick(View v) {
            LogInPage(v);
        }
    });

    Button makeComplaint = (Button) findViewById(R.id.MakeComplaint);
    makeComplaint.setVisibility(View.GONE);
    makeComplaint.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {CreateComplaints(v); }
    });

    Button viewPro = (Button) findViewById(R.id.viewProfile);
    viewPro.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {ViewProfile(v); }
    });

    Button nextActivity = (Button) findViewById(R.id.nextActivityButton);
    Button requests = (Button)findViewById(R.id.requestsButton);
    requests.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            viewRequests(v,role);
        }
    });

    if(role.equals("CLIENT")){
        nextActivity.setText("Search Meals");
        nextActivity.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                searchMeals(v);
            }
        });
        makeComplaint.setVisibility(View.VISIBLE);
        makeComplaint.setVisibility(View.VISIBLE);
```

```
    }

    else if(role.equals("ADMIN")){
        nextActivity.setText("View Complaints");
        requests.setVisibility(View.GONE);
        nextActivity.setOnClickListener(new View.OnClickListener(){
            public void onClick(View v){
                ComplaintsPage(v);
            }
        });
        //makeComplaint.setVisibility(View.GONE);

    }
    else if(role.equals("COOK")){
        nextActivity.setText("View Menu");
        nextActivity.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                menuPage(v);
            }
        });
        //makeComplaint.setVisibility(View.GONE);
        viewPro.setVisibility(View.VISIBLE);
    }


}

public void LogInPage(View view){
    Intent intent = new Intent(getApplicationContext(), MainActivity.class);
    startActivity(intent);


}
```

```
}

public void viewRequests(View view,String role){
    Intent viewRequestsPage = new Intent(getApplicationContext(), ViewPurchaseRequests.class);
    viewRequestsPage.putExtra("UID",UserDatabase.getUID());
    viewRequestsPage.putExtra("role",role);
    startActivity(viewRequestsPage);
}

public void CreateComplaints(View view){
    Intent intent = new Intent(getApplicationContext(), CreateComplaint.class);
    startActivity(intent);
}

public void ComplaintsPage(View view){
    Intent intent = new Intent(getApplication(), ComplaintsPage.class);
    startActivity(intent);
}

public void ViewProfile(View view){
    Intent intent = new Intent(getApplication(), ViewProfile.class);
    startActivity(intent);
}

public void menuPage(View view){
    Intent menuPage = new Intent(getApplicationContext(), MenuPage.class);
    UserDatabase dtb = new UserDatabase();
    String uid = dtb.getUID();
    menuPage.putExtra("UID", uid);
        startActivity(menuPage);

}


   public void searchMeals(View view){
       Intent searchMealsPage = new Intent(getApplicationContext(),SearchMeals.class);
       startActivity(searchMealsPage);
   }


}
```

UML:


User
-firstName: String
-lastName: String
-email: String
-password: String
-address: String
-type: userType
+getFirstName(): String
+setFirstName(firstName: String): void
+getLastName(): String
+setLastName(lastName: String): void
+getEmail(): String
+setEmail(email: String)
+getPassword(): String
+setPassword(password: String): void
+getAddress(): String
+setAddress(address: String): void
+getRole(): userType
+setRole(role: userType): void
Client
-ClientCardInfo: String
+getCreditCardInfo(): String
+setCreditCardInfo(newCCInfo: String): void
+toString(): String
Cook
+description: String
+getDescription(): String
+setDescription(newDescription: String): void
+toString(): String
Complaint
-description: String
-clientUID: String
-cookUID: String
+getComplaint(): String
+setComplaint(complaint: String): void
+getClientUID(): String
+setClientUID(): String
+getCookUID(): String
+setCookUID(cookUID: String): void
+setRead(read:boolean): void
+getRead(): boolean
1
1
PurchaseRequest
-meal: String
-clientUID: String
-cookUID: String
-date: String
+getClientUID(): String
+setClientUID(): String
+getCookUID(): String
+setCookUID(cookUID: String): void
+getMeal(): String
+setMeal(meal: String): void
+getDate(): String
+setdate(date: String): String
1
0..n
Meal
-name: String
-mealType: String
-cuisine: String
-ingredients: String
-allergens: String
-description: String
-price: double
-currentlyOffered: Boolean
-rating: double
+getName(): String
+setname(name: String): void
+getMealType(): String
+setMealType(mealType: String): void
+getCuisine(): String
+setCuisine(cuisine: String): void
+getIngredients(): String
+setIngredients(ingredients: String): void
+getAllergens(): String
+setAllergens(allergens: String): void
+getDescription(): String
+setDescription(description: String): void
+getPrice(): double
+setPrice(price: double): void
+getCurrentlyOffered(): Boolean
+setCurrentlyOffered(currentlyOffered: Boolean): Boolean
+getRating(): double
+setRating(rating: double): void

Conclusion:

In this project, we learned how to use Android Studio to create an app using Firebase and the Java language. We learned how to implement the code in a superclass called User, redirect it to subclasses called Client, Cook, and Complaint, and additionally add a database. We've also implemented several methods and features that make the app work with a set of rules provided by the client and cook class. We also created a food class and a menu class with functions and a set of rules that also act on the app. Later, we also created a UML diagram to better understand the polymorphism and inheritance that occur in the code. This helped a lot in building our understanding. After that, we learned the concept of "firebase," which allows multiple people to view the output or input of particular data. Also, we have learned to work as a team using GitHub and Gitbash and have concluded this project with many successful running programs. This project has been very helpful to us, future engineers, as it has given us insight into our future work environment and a better picture of what to expect there.