University of Ottawa

School of electrical Engineering and Computer Science (EECS)

SEG2105[B]
Fall 2022

Android Project: Mealer App

Group ID (24)
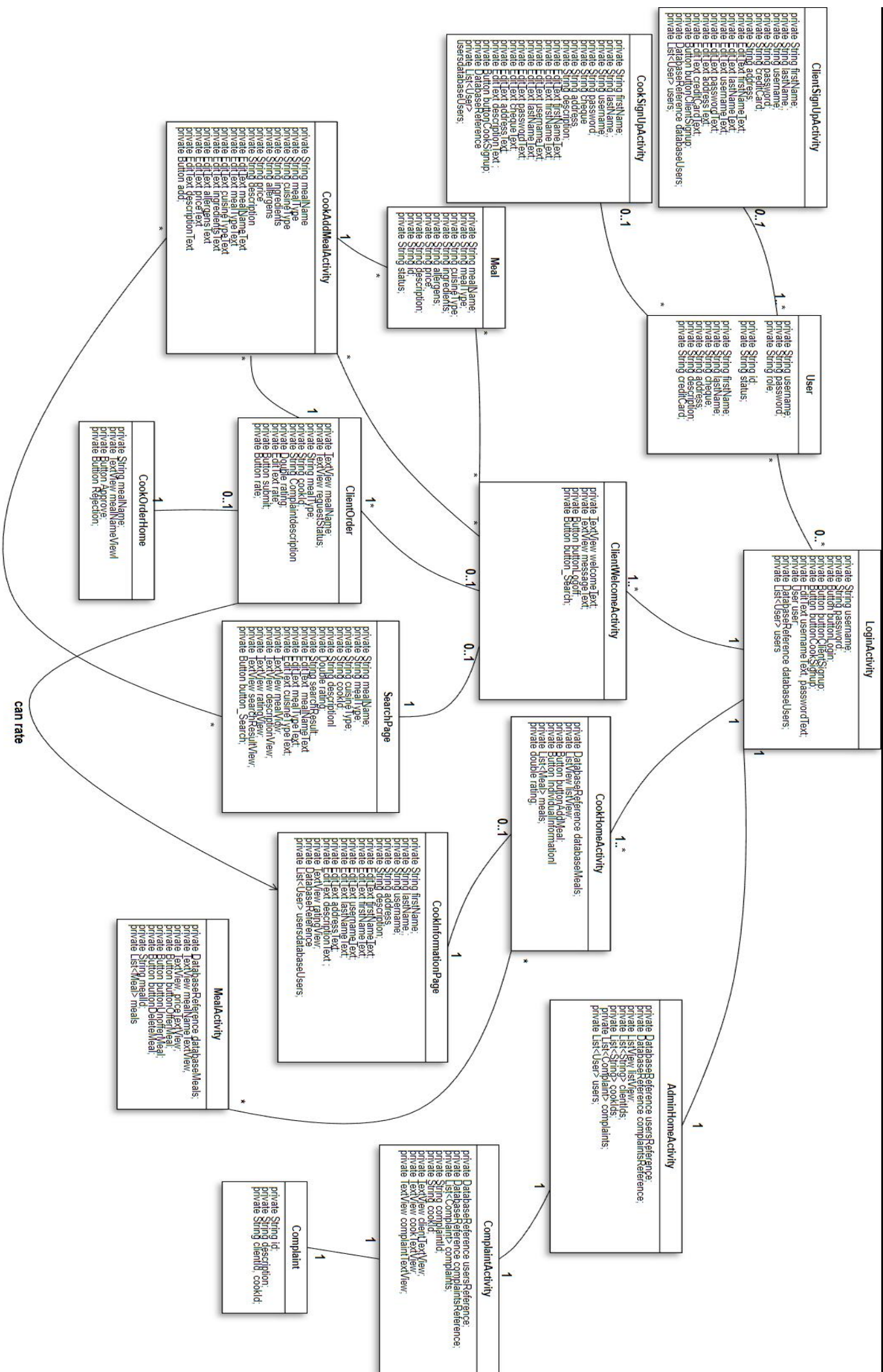
Zhiwei Fu

Haoran Jia

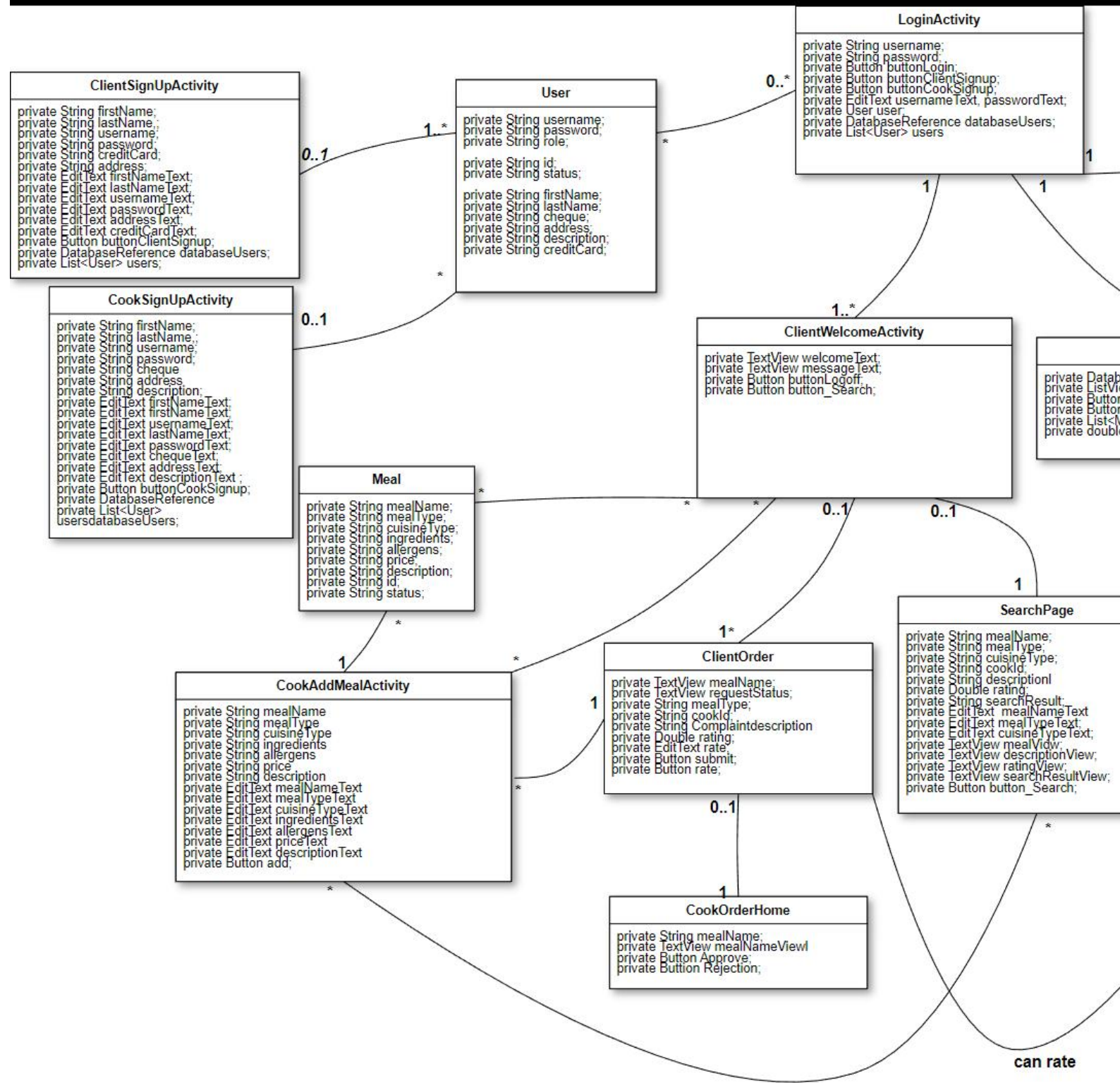Junbo Wang

Ruoyu Liu

Presented to:

Dr. Wassim El Ahmar

Introduction

Our project is about a meal app that allows cook and clients to interact. in our app, there are three types of users: cook, client and admin. we designed the login page and the user registration page. In the registration page, users can choose to register themselves as a chef or as a customer. By the user's account type, the app system will load a different page for the user. Chef account users will have the right to offer, delete dishes, or modify dish information. Customer account users can choose to purchase dishes, cancel dishes, and file complaints with the chef. admin account users will collect complaints from clients, and admin users will have the option to suspend the service of the chef being complained about. The system interconnects the different sections to link the chef, customer and admin. The whole system is logical and provides feedback based on reality.

UML diagram

UML Class Diagram

ClientSignUpActivity
- private String firstName;
- private String lastName;
- private String username;
- private String password;
- private String address;
- private String creditCard;
- private EditText firstName_Text;
- private EditText lastName_Text;
- private EditText username_Text;
- private EditText password_Text;
- private EditText address_Text;
- private EditText creditCardText;
- private Button button_ClientSignup;
- private DatabaseReference databaseUsers;
- private List<User> users;

CookSignUpActivity
- private String firstName;
- private String lastName;
- private String username;
- private String password;
- private String Cheque;
- private String address;
- private String description;
- private EditText firstName_Text;
- private EditText lastName_Text;
- private EditText username_Text;
- private EditText password_Text;
- private EditText address_Text;
- private EditText description_Text;
- private EditText cheque_Text;
- private Button button_CookSignup;
- private DatabaseReference
- private List<User> usersdatabaseUsers;

Meal
- private String mealName;
- private String mealType;
- private String cuisine_type;
- private String ingredients;
- private String allergens;
- private String price;
- private String id;
- private String description;
- private String status;

CookAddMealActivity
- private String mealName;
- private String mealType;
- private String cuisineType;
- private String ingredients;
- private String allergens;
- private String price;
- private String description;
- private EditText mealName_Text;
- private EditText mealType_Text;
- private EditText cuisineType_Text;
- private EditText ingredients_Text;
- private EditText allergens_Text;
- private EditText price_Text;
- private EditText description_Text;
- private Button add;

User
- private String username;
- private String password;
- private String id;
- private String firstName;
- private String lastName;
- private String cheque;
- private String address;
- private String description;
- private String creditCard;

LoginActivity
- private String username;
- private String password;
- private Button button_Login;
- private Button button_ClientSignup;
- private Button button_CookSignup;
- private EditText username_Text, passwordText;
- private User user;
- private DatabaseReference databaseUsers;
- private List<User> users;

ClientWelcomeActivity
- private TextView welcome_Text;
- private TextView message_Text;
- private Button button_Logoff;
- private Button button_Search;

ClientOrder
- private TextView mealName;
- private TextView requestStatus;
- private String mealType;
- private String cookId;
- private String ComplaintDescription;
- private Double rating;
- private EditText rate;
- private Button submit;
- private Button rate;

CookOrderHome
- private String mealName;
- private TextView mealNameView;
- private Button Approve;
- private Button Rejection;

SearchPage
- private String mealName;
- private String mealType;
- private String cuisineType;
- private String cookId;
- private String description;
- private Double rating;
- private EditText searchResult;
- private TextView mealNameView;
- private TextView cuisineType_text;
- private EditText mealType_text;
- private TextView description_view;
- private TextView rating_view;
- private Button button_searchResultView;
- private Button button_Search;

CookHomeActivity
- private DatabaseReference databaseMeals;
- private ListView listView;
- private Button button_AddMeal;
- private Button IndividualInformationI;
- private List<Meal> meals;
- private double rating;

CookInformationPage
- private String firstName;
- private String lastName;
- private String username;
- private String address;
- private String description;
- private EditText firstName_Text;
- private EditText lastName_Text;
- private EditText username_Text;
- private EditText address_Text;
- private EditText description_Text;
- private Button button;
- private DatabaseReference
- private List<User> usersdatabaseUsers;

MealActivity
- private DatabaseReference databaseMeals;
- private TextView mealName_TextView;
- private TextView price_TextView;
- private Button button_OrderMeal;
- private Button button_UpdateMeal;
- private Button button_DeleteMeal;
- private String mealId;
- private List<Meal> meals

AdminHomeActivity
- private DatabaseReference usersReference;
- private DatabaseReference complaintsReference;
- private ListView listView;
- private List<String> clients;
- private List<String> cookIds;
- private List<Complaint> complaints;
- private List<User> users;

ComplaintActivity
- private DatabaseReference usersReference;
- private DatabaseReference complaintsReference;
- private String complaintId;
- private TextView clientTextView;
- private TextView cook_textView;
- private TextView complaintTextView;

Complaint
- private String id;
- private String description;
- private String clientId, cookId;

Relationship labels: 0.1, 1..*, 1, *, can rate

Since the final UML diagram is too big, we'll separate it into two parts in the following two pages:

LoginActivity

private String username;
private String password;
private Button buttonLogin;
private Button buttonClientSignup;
private Button buttonCookSignup;
private EditText usernameText, passwordText;
private User user;
private DatabaseReference databaseUsers;
private List<User> users;

0..*

1

ClientSignUpActivity

private String firstName;
private String lastName.;
private String username;
private String password;
private String creditCard;
private String address;
private EditText firstNameText;
private EditText lastNameText;
private EditText usernameText;
private EditText passwordText;
private EditText addressText;
private EditText creditCardText;
private Button buttonClientSignup;
private DatabaseReference databaseUsers;
private List<User> users;

0..1   1..*

User

private String username;
private String password;
private String role;

private String id;
private String status;

private String firstName;
private String lastName;
private String cheque;
private String address;
private String description;
private String creditCard;

*

1   1

1..*

ClientWelcomeActivity

private TextView welcomeText;
private TextView messageText;
private Button buttonLogoff;
private Button button_Search;

private Datab
private ListVi
private Button
private Button
private List<M
private doubl

CookSignUpActivity

private String firstName;
private String lastName.;
private String username;
private String password;
private String cheque
private String address
private String description;
private EditText firstNameText;
private EditText firstNameText;
private EditText usernameText;
private EditText lastNameText;
private EditText passwordText;
private EditText chequeText;
private EditText addressText;
private EditText descriptionText ;
private Button buttonCookSignup;
private DatabaseReference
usersdatabaseUsers;
private List<User>

0..1

Meal

private String mealName;
private String mealType;
private String cuisineType;
private String ingredients;
private String allergens;
private String price;
private String description;
private String id;
private String status;

*   *   *   0..1   0..1

1

SearchPage

private String mealName;
private String mealType;
private String cuisineType;
private String cookId;
private String descriptionI
private Double rating;
private String searchResult;
private EditText mealNameText
private EditText mealTypeText
private EditText cuisineTypeText;
private TextView mealView;
private TextView descriptionView;
private TextView ratingView;
private TextView searchResultView;
private Button button_Search;

1   *

CookAddMealActivity

private String mealName
private String mealType
private String cuisineType
private String ingredients
private String allergens
private String price
private String description
private EditText mealNameText
private EditText mealTypeText
private EditText cuisineTypeText
private EditText ingredientsText
private EditText allergensText
private EditText priceText
private EditText descriptionText
private Button add;

1   *   1*

1

ClientOrder

private TextView mealName;
private TextView requestStatus;
private String mealType;
private String cookId;
private String Complaintdescription
private Double rating;
private EditText rate;
private Button submit;
private Button rate;

0..1

1

CookOrderHome

private String mealName;
private TextView mealNameViewI
private Button Approve;
private Button Rejection;

*

can rate

LoginActivity

username;
password;
buttonLogin;
buttonClientSignup;
buttonCookSignup;
xt usernameText, passwordText;
ser;
aseReference databaseUsers;
ser> users

1

1   **1**

Activity

1..*

CookHomeActivity

private DatabaseReference databaseMeals;
private ListView listView;
private Button buttonAddMeal;
private Button IndividualInformationl
private List<Meal> meals;
private double rating;

0..1

*****

AdminHomeActivity

private DatabaseReference usersReference;
private DatabaseReference complaintsReference;
private ListView listView;
private List<String> clientIds;
private List<String> cookIds;
private List<Complaint> complaints;
private List<User> users;

1

1

ComplaintActivity

private DatabaseReference usersReference;
private DatabaseReference complaintsReference;
private List<Complaint> complaints;
private String complaintId;
private String cookId;
private TextView clientTextView;
private TextView cookTextView;
private TextView complaintTextView;

0..1

1

SearchPage

private String mealName;
private String mealType;
private String cuisineType;
private String cookId;
private String descriptionl
private Double rating;
private String searchResult;
private EditText mealNameText
private EditText mealTypeText;
private EditText cuisineTypeText;
private TextView mealVidw;
private TextView descriptionView;
private TextView ratingView;
private TextView searchResultView;
private Button button_Search;

CookInformationPage

private String firstName;
private String lastName,;
private String username;
private String address,
private String description;
private EditText firstNameText;
private EditText firstNameText;
private EditText usernameText;
private EditText lastNameText;
private EditText addressText;
private EditText descriptionText ;
private TextView ratingView;
private DatabaseReference
private List<User> usersdatabaseUsers;

1

1

Complaint

private String id;
private String description;
private String clientId, cookId;

*****

MealActivity

private DatabaseReference databaseMeals;
private TextView mealNameTextView,
private TextView, priceTextView;
private Button buttonOfferMeal;
private Button buttonUnofferMeal;
private Button buttonDeleteMeal;
private String mealId;
private List<Meal> meals

*****

can rate

Table of contributions

Contributions / Name	Deliverable 1	Deliverable 2	Deliverable 3	Deliverable 4
Haoran Jia	The UML diagram	The UML diagram, Project test cases, Project testing	The UML diagram, Cook deleting meals function, Suspended message function	The UML diagram, Client rate function, Client complaint function
Junbo Wang	User register function, User log in/off function, Database use	Administrator function, Cook function, Project testing	Cook adding meals function, Cook deleting meals function, Remove meals from offered list function	Client search function, Client view setting, Client complaint function, Project test cases
Ruoyu Liu	User log in/off function	Cook function, Project test cases, Project testing	Cook deleting meals function, Cook remove meals from offered meals list function	Client search function, Client complaint function
Zhiwei Fu	User log in function, Welcome block function	Administrator function, Connection with DB(value stored)	Remove meals from offered list function,Suspended message function	Final report, Project test cases, Cook can view message from the clients

Screenshots

User login page
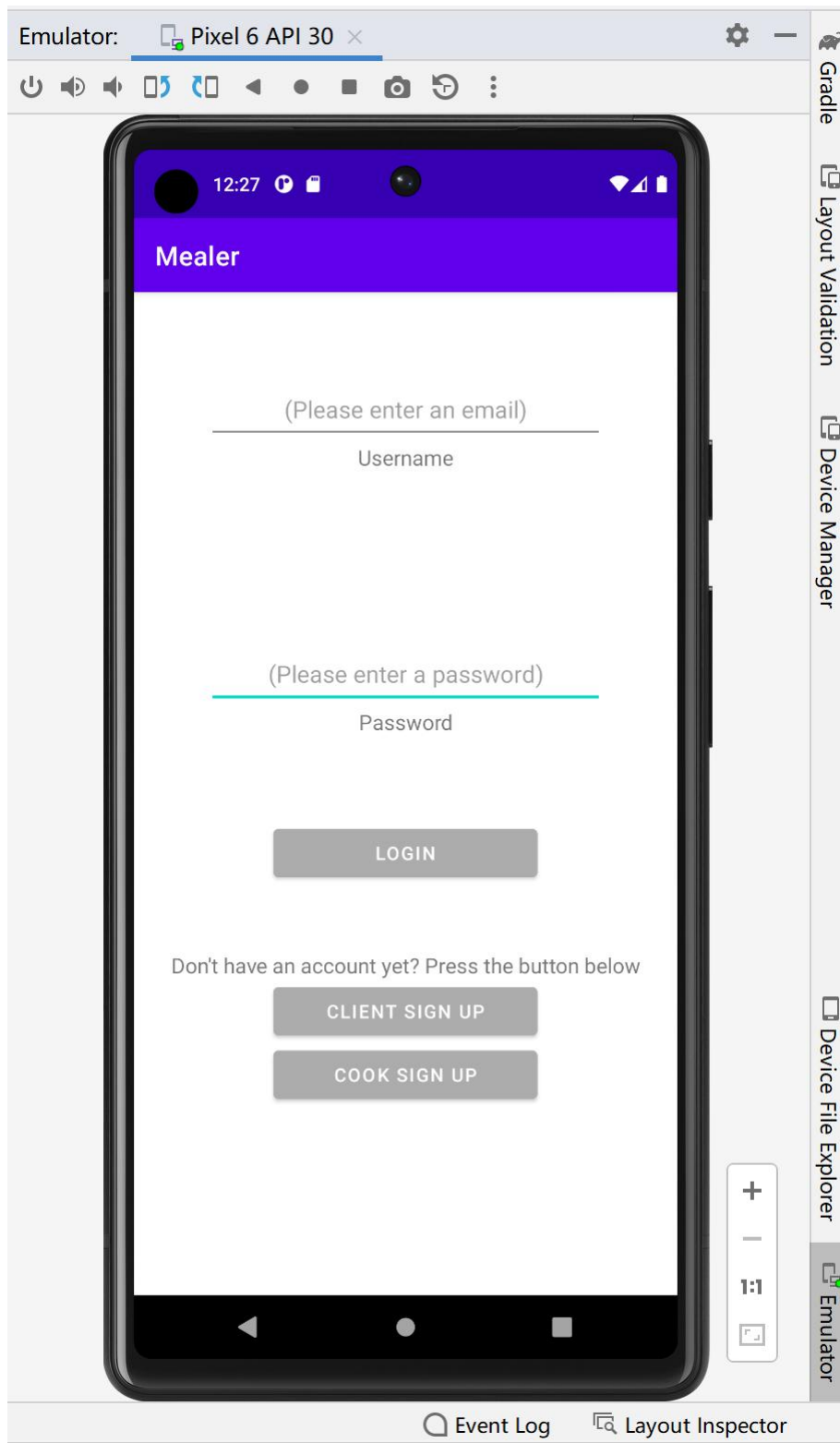
Admin login welcome block

Admin main page

Mealer

Complaints:

From:	Lee
To:	Cook1
From:	Lee
To:	Cook2
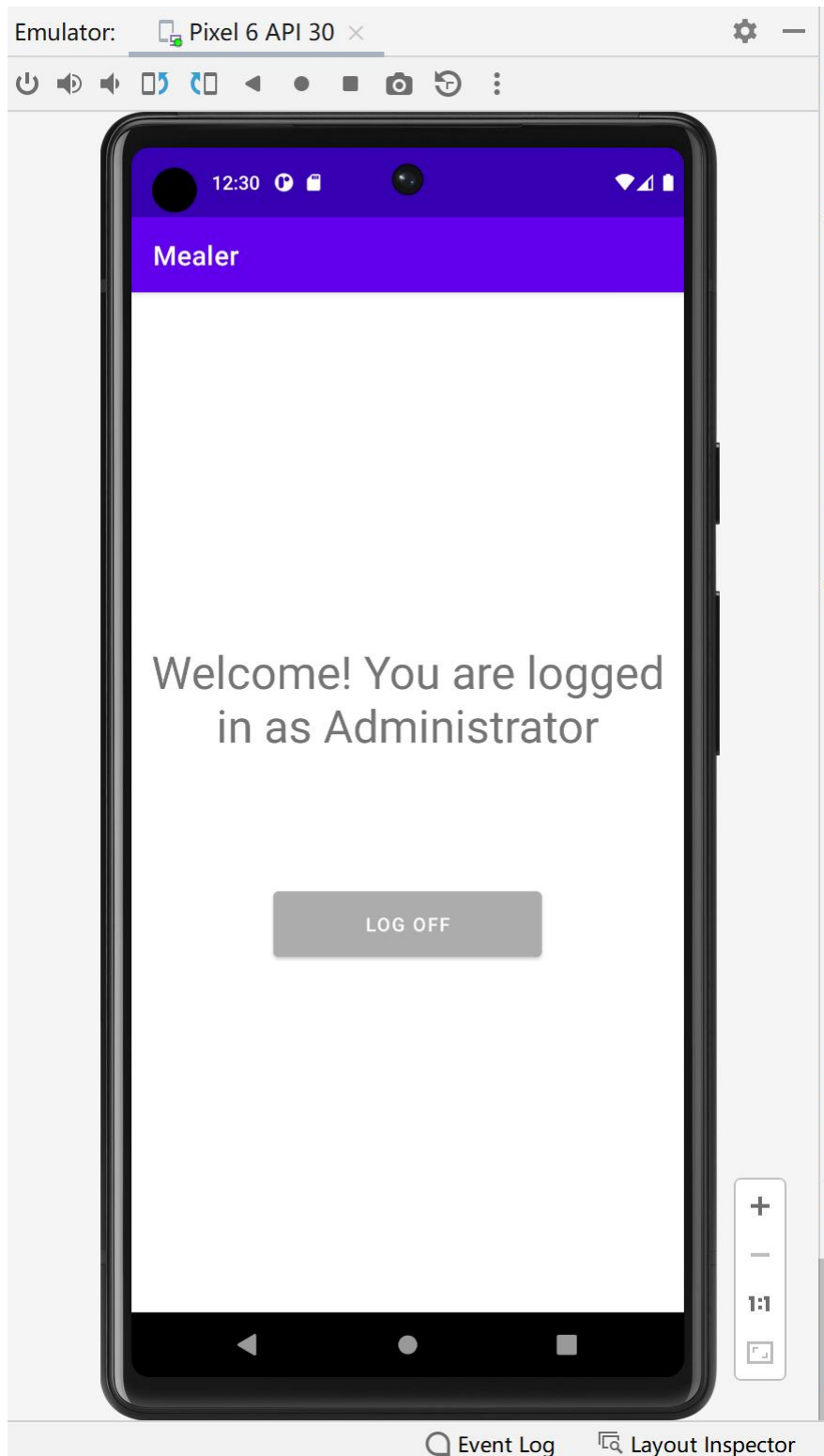
Complaints (view from admin account)

Cook sign up(Register) page
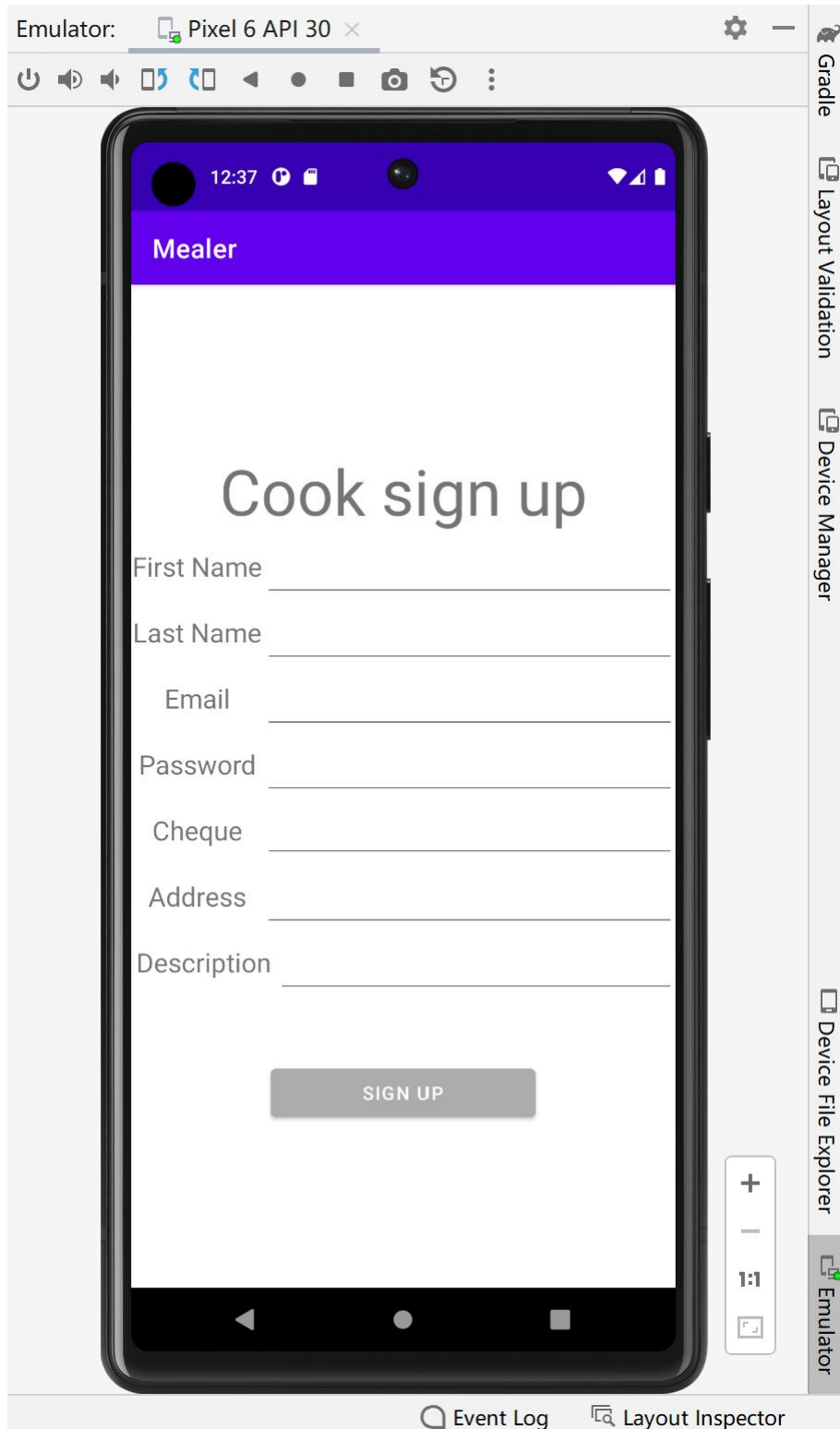
Cook main page
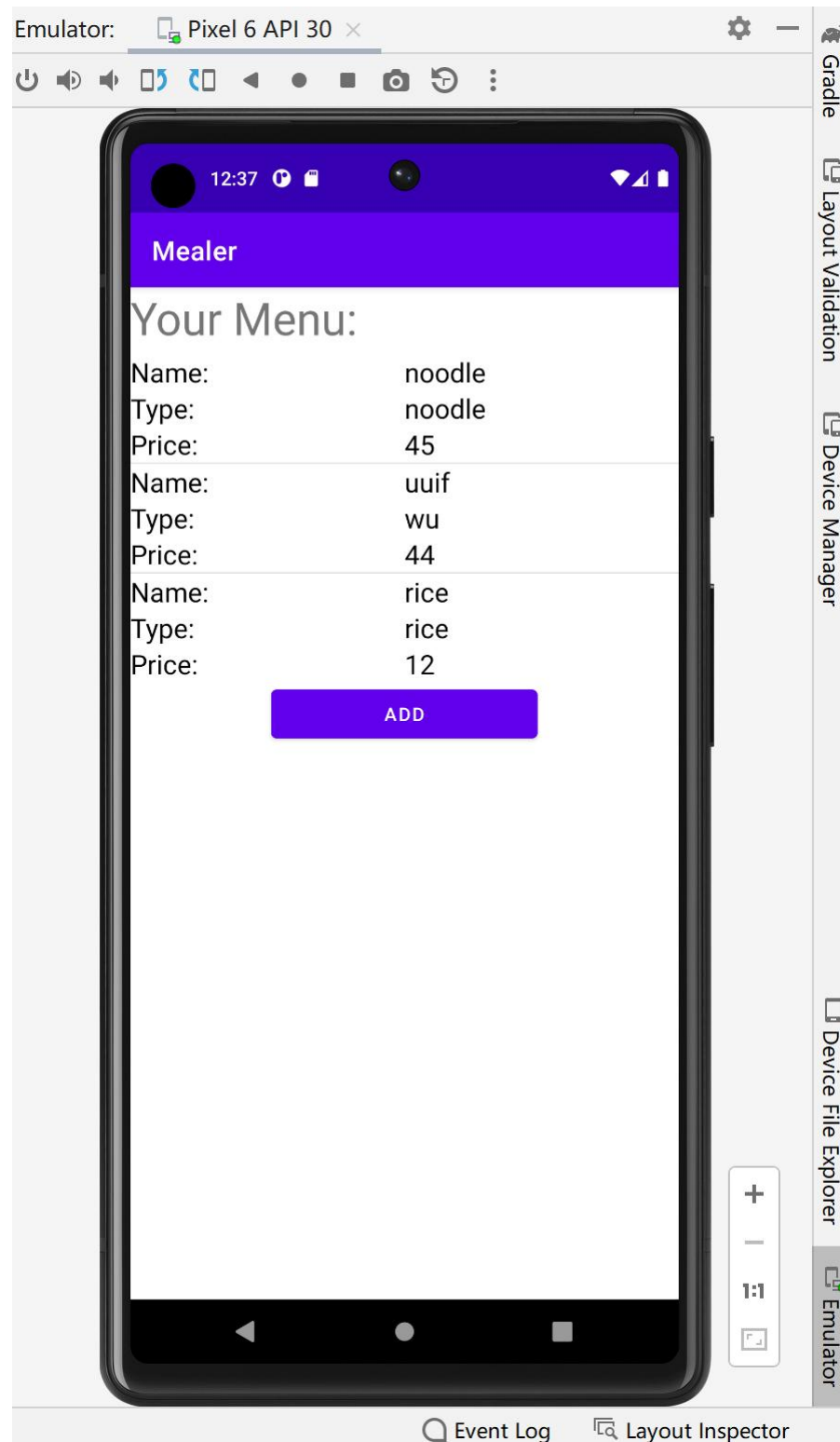
Cooks' meal management page
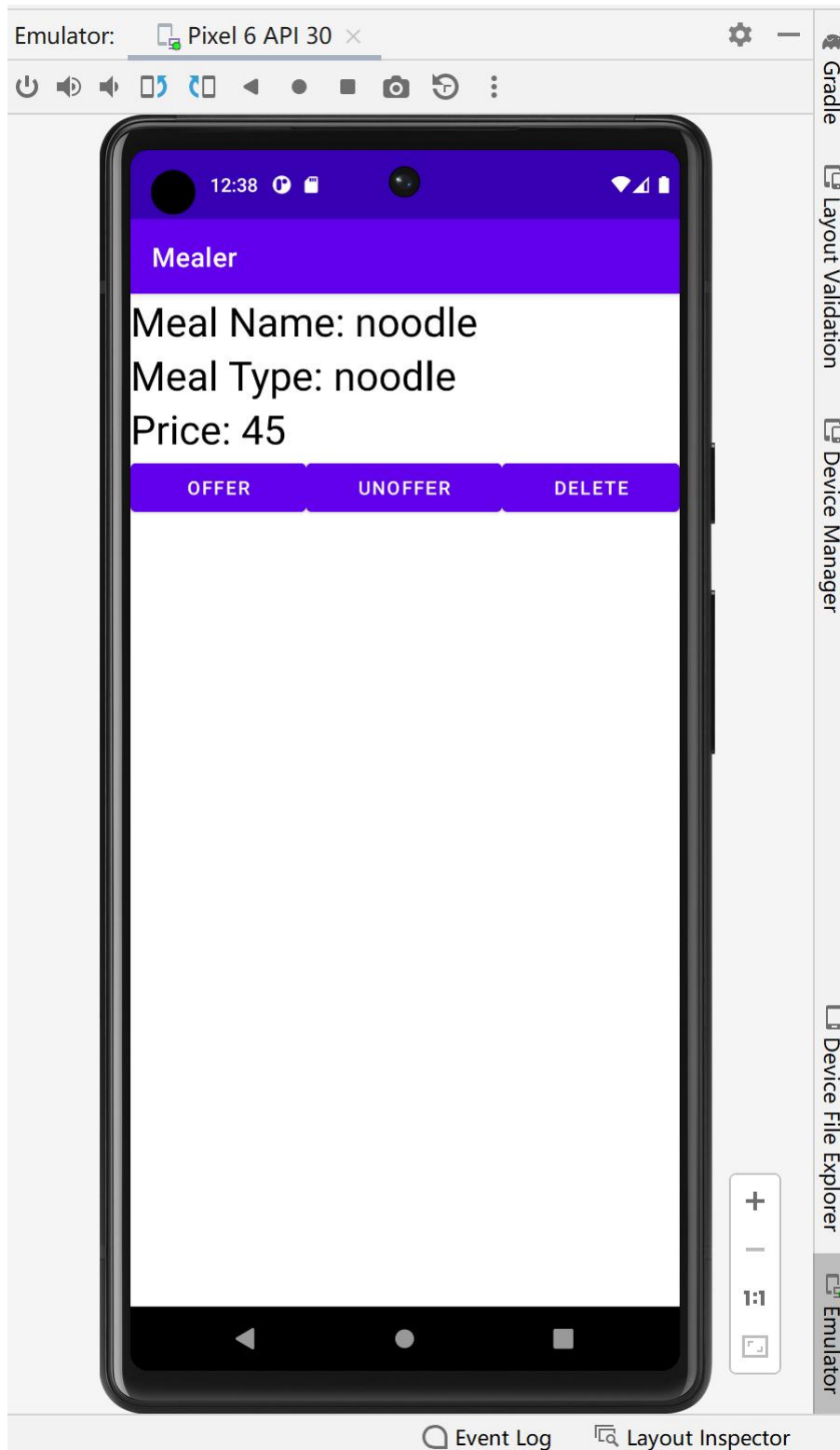
Suspended Cook login page

Mealer

Welcome! You are logged
in as Cook
Your account has been
suspended!

LOG OFF
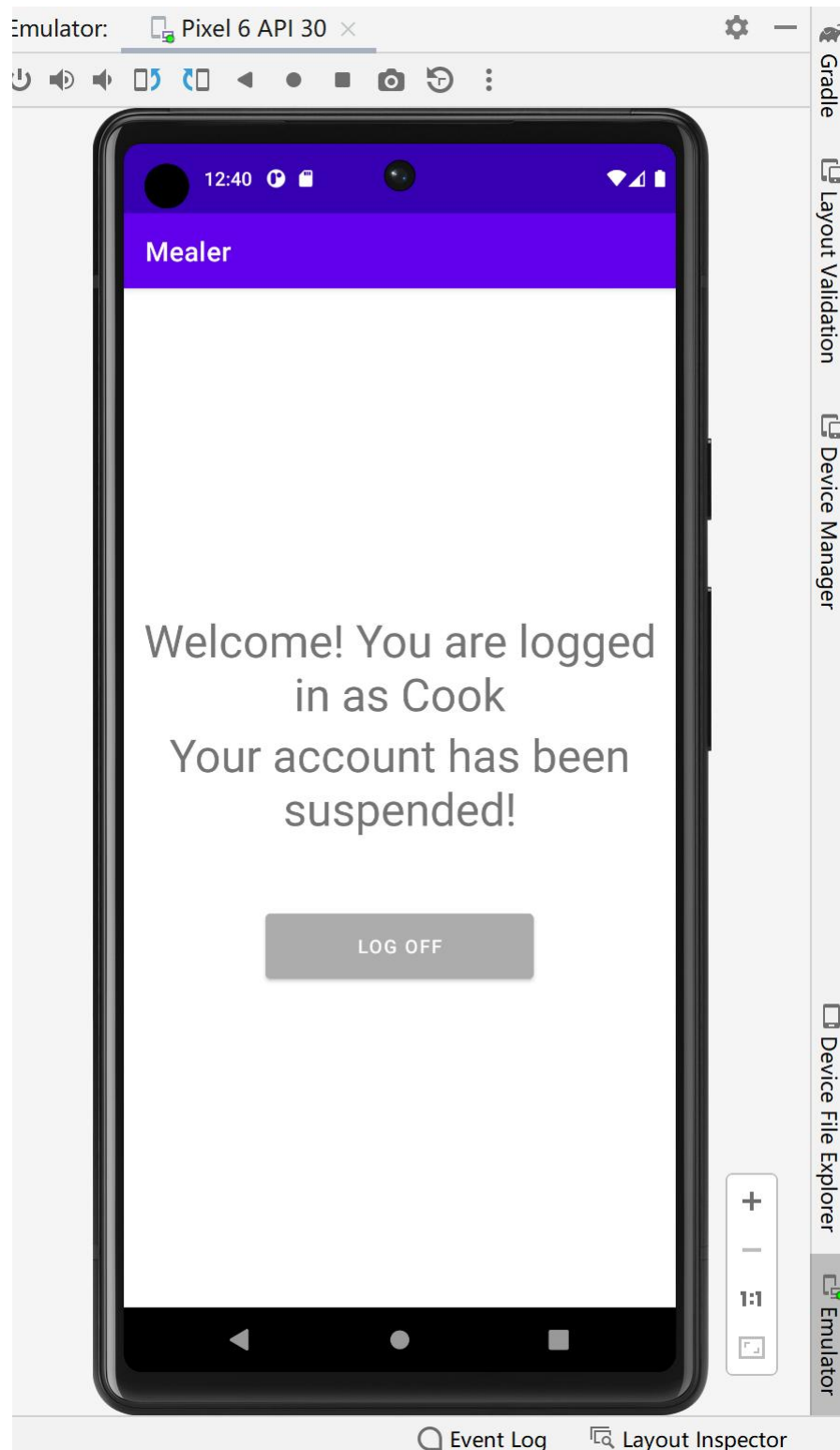
Client sign up(Register) page
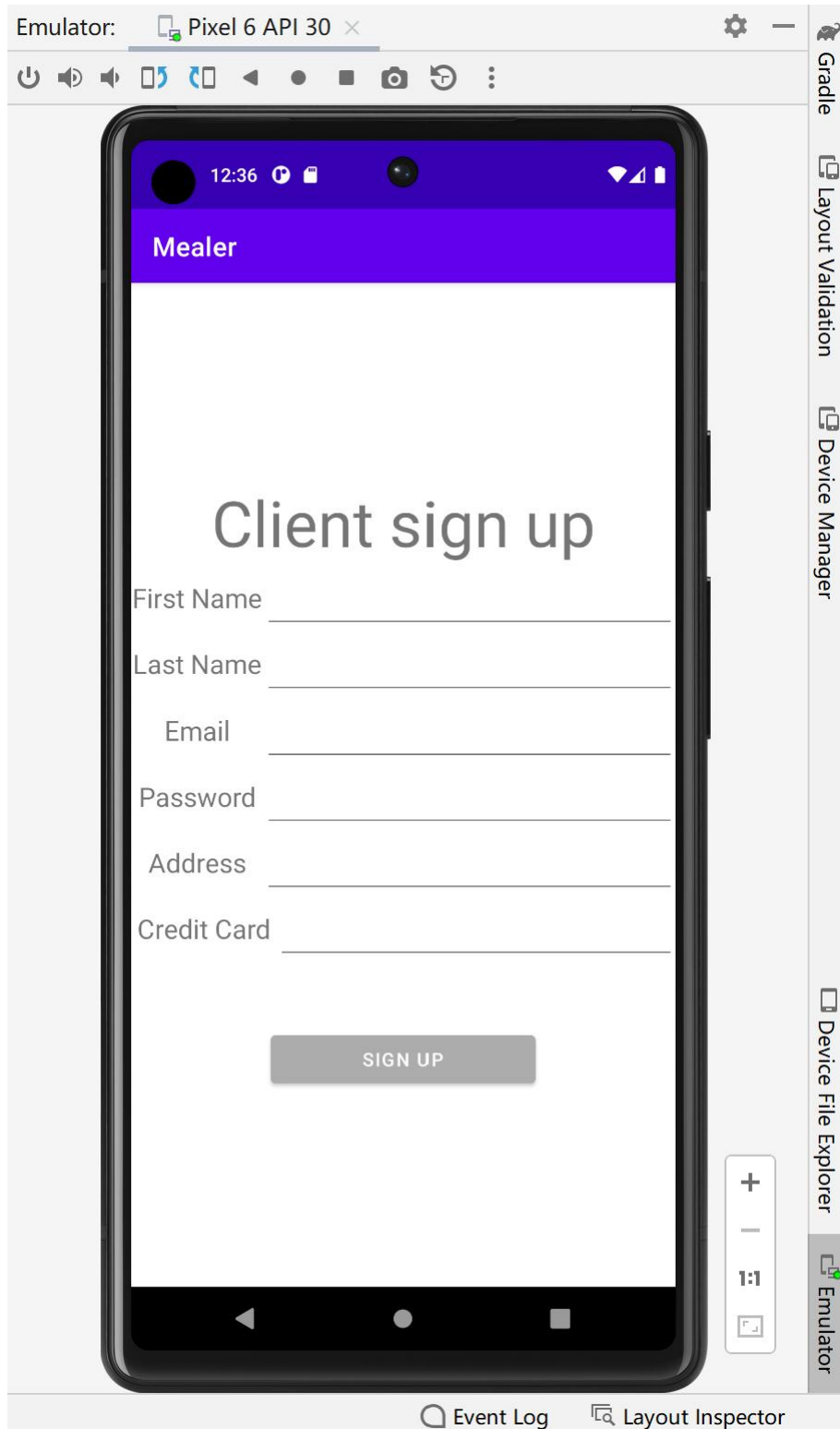
Client searching page
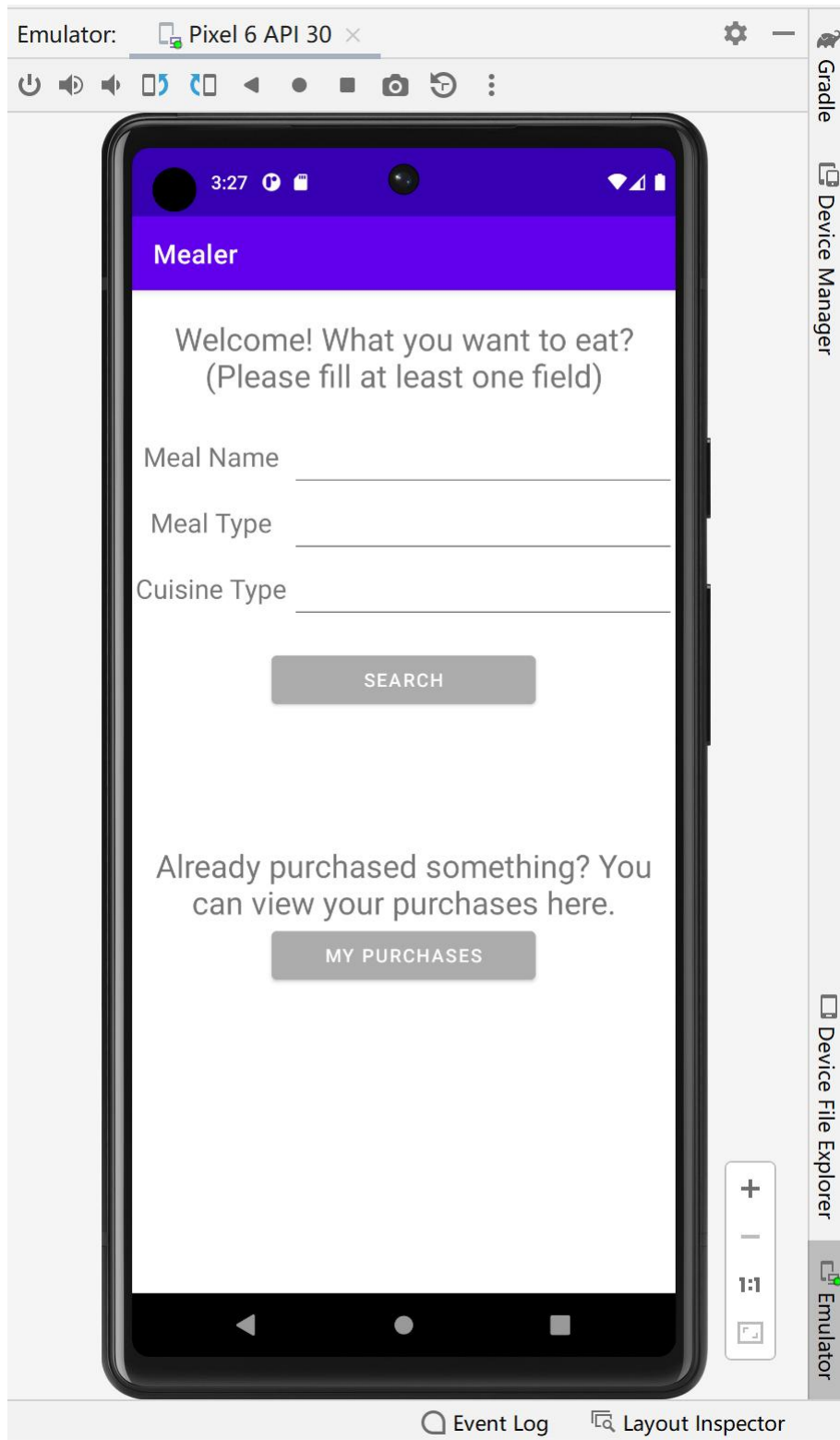
Client searching result page

Emulator: Pixel 6 API 30 ✕

Mealer

Your Result:

Name:	noodle	Type:	noodle
Cuisine:	noodle	Price:	45

Ingredients:
tomato
Allergens:
none
Description:
hot

Name:	ramen	Type:	noodle
Cuisine:	noodle	Price:	32

Ingredients:
fish
Allergens:
fish
Description:
good

Name:	soup	Type:	noodle
Cuisine:	noodle	Price:	44

Ingredients:
cabbage
Allergens:
err
Description:
vegetables

Name:	pork	Type:	noodle

Event Log   Layout Inspector

Client order page

Client view order page

Mealer

Your Purchases:

Cook Name:	Lee
Meal Name:	ramen
Address:	555 Bank Ave.
Pick-up Time:	6:30
Cook Name:	Lee
Meal Name:	soup
Address:	555 Bank Ave.
Pick-up Time:	5pms
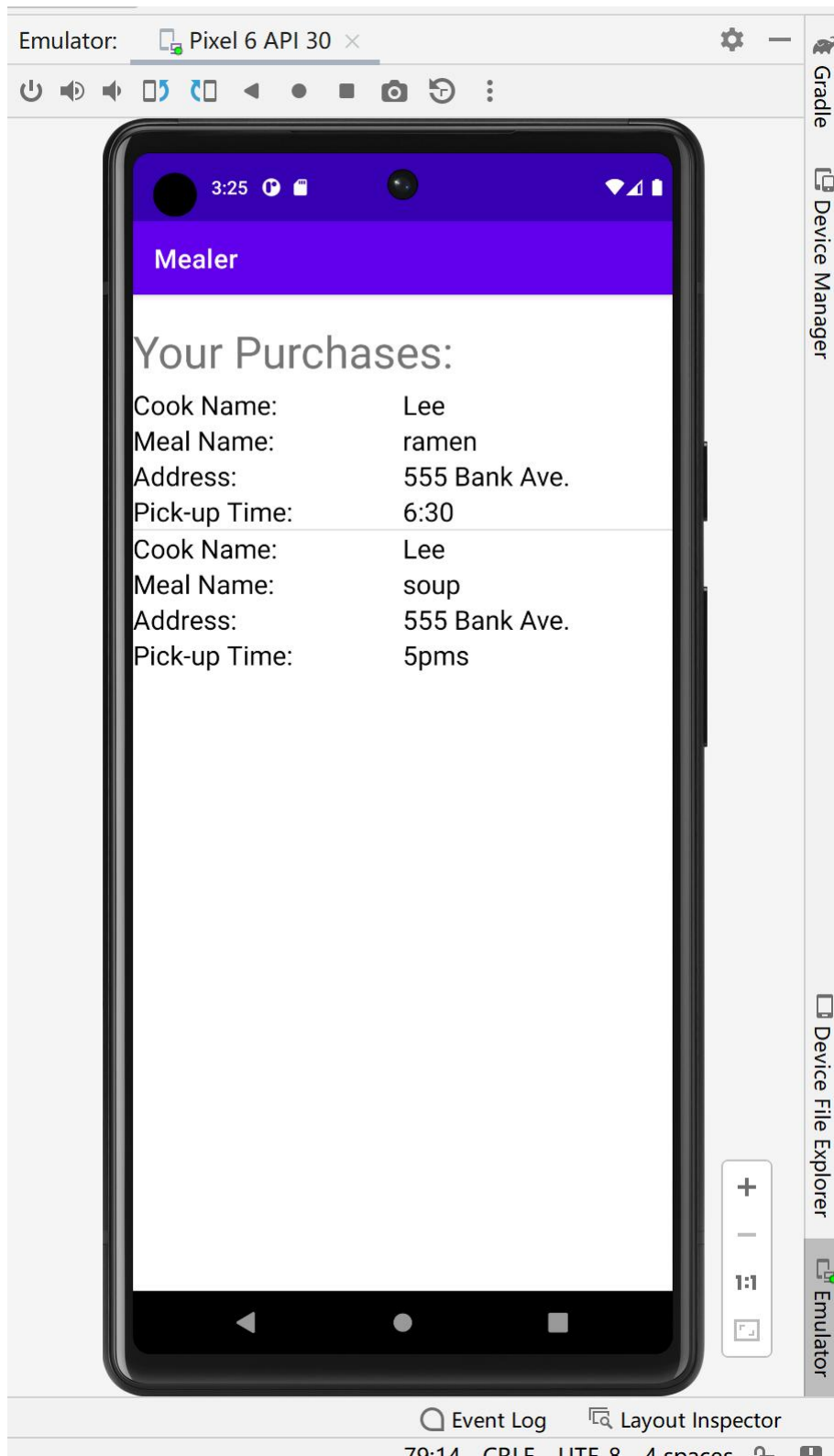
Client rating and complain page
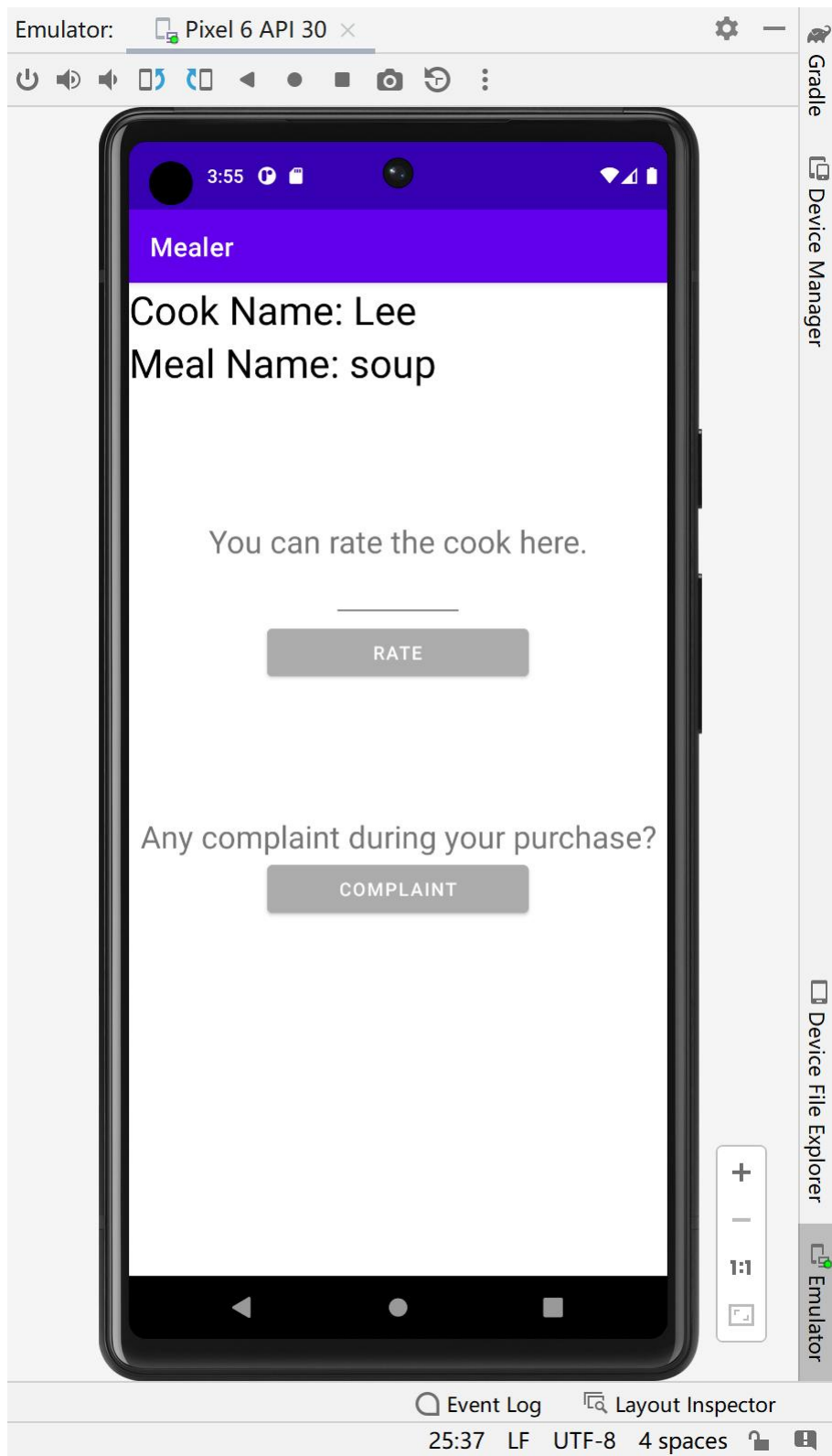
Lessons Learned

In this Android project, we designed a mealer app. we learned a lot of new knowledge by using Android Studio in combination with what the team members knew to complete the project. The team members had used Firebase as the database for the app system in deliverable 1. Firebase is an online database. by connecting with the offline system, the account information registered by all users in the registration page of the project will be transferred and stored in Firebase. We found that by doing this we could significantly reduce the application's need for local memory and share the database across multiple locations. One of the things about the use of Github. The panelists found that while we knew how to use git for code upload and download, the git plugin that comes with the computer programming software would be much easier and more time efficient. The team encountered difficulties in completing the cook feature, which has two functions: offer meals and reject offer meals. Since both features are associated with the client, the team members needed to ensure that the client could not see the cookie rejecting the offer, so that the client would not place an order for the cookie rejecting the offer. However, since this step was split into different members at the beginning of the project, it was not possible to connect the code effectively. Although the program runs successfully, the result fails to prioritize the removal of meals from the client's page. This was the result of team members using different call

methods. This led to a renewed focus on the division of labor among team members. After several bug-finding sessions and redistribution of project content, we managed to implement its functionality in Android Studio's virtual phone.