

# **MEALER Project**

SEG2105A

Course Coordinator: Dr. Hussein Al Osman

Group number 7

Lab section B

Chujian Xiao: 300256072

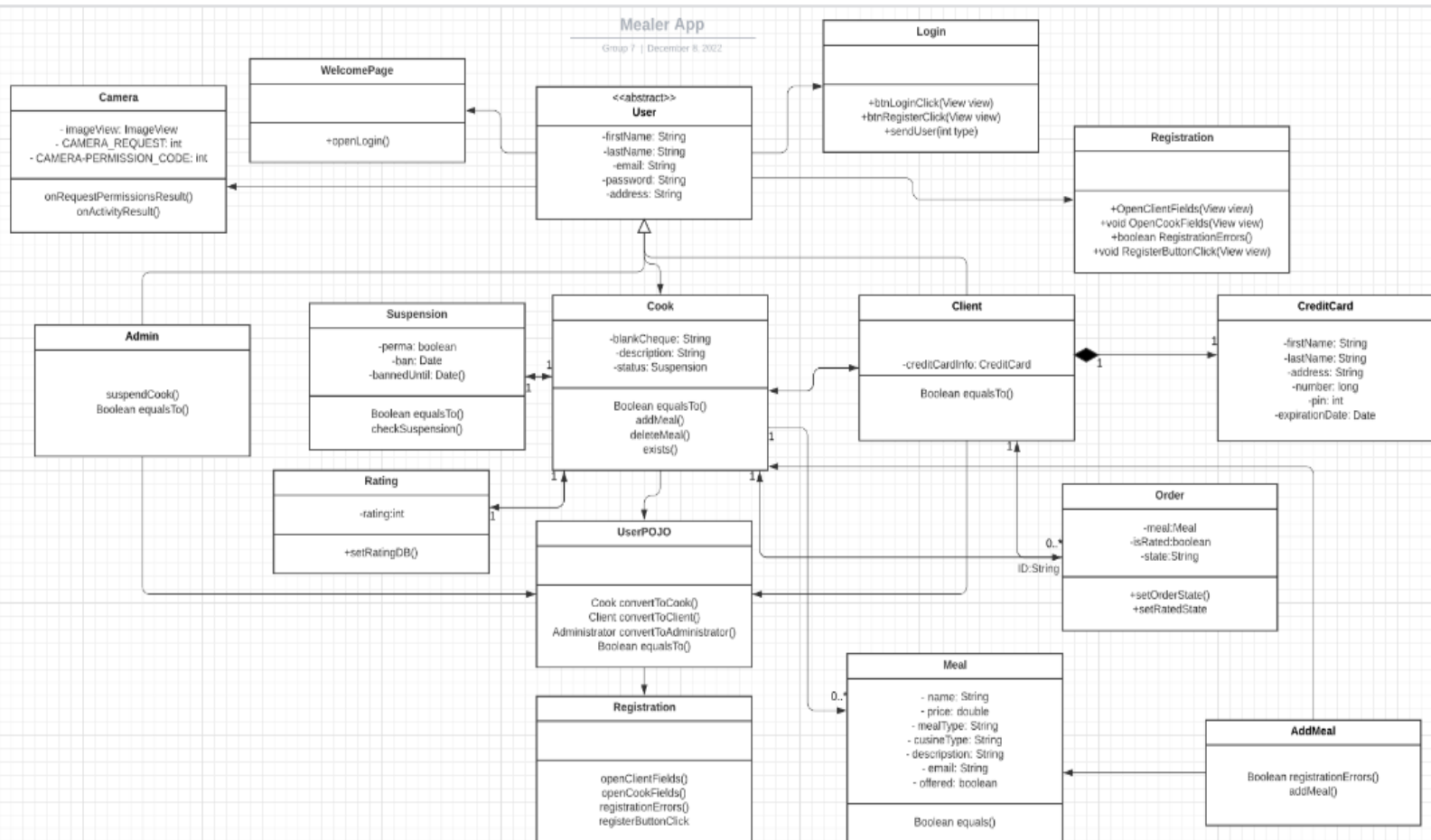
Shezaad Jiffry: 300129568

Karthikan Suntharesan: 300240065

Renojan Kannan: 300240070

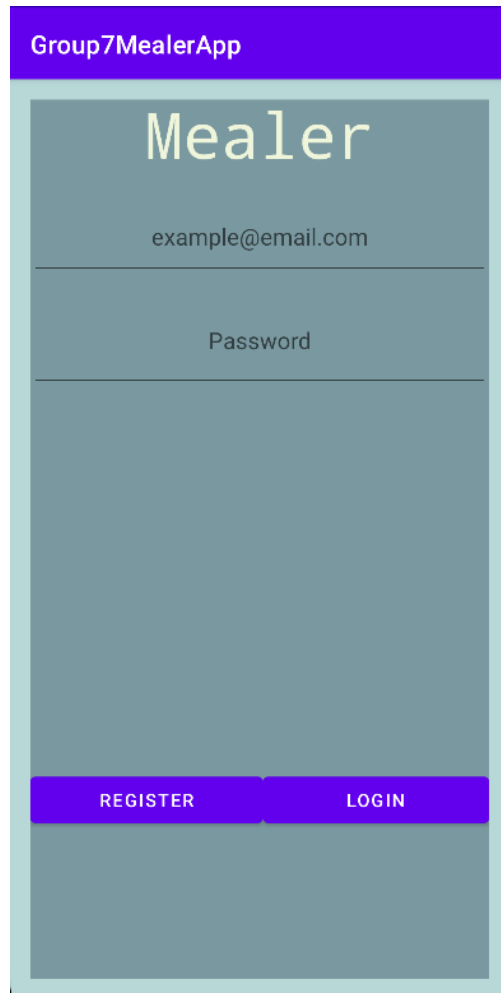
Zakaia Hiri: 300175288

The app designed for this project is called “Mealer”, an Ottawa-based meal-sharing app on Android developed by our team that allows cooks to sell meals they prepared to clients directly from their homes. There are 3 types of users on the platform: the admin, the cooks and, the users. The admin resolves complaints from the users and can choose to ban, suspend or ignore complaints. Cooks can make meals to be sold and clients can buy said meals, rate the cook and complain about the food as well. All this functionality has been programmed except for debiting the card or having a system that delivers the food to the customer directly. The database for the project is implemented using Firebase, where information about the user is passed and stored as Java Objects, which is then retrieved when a user logs in.

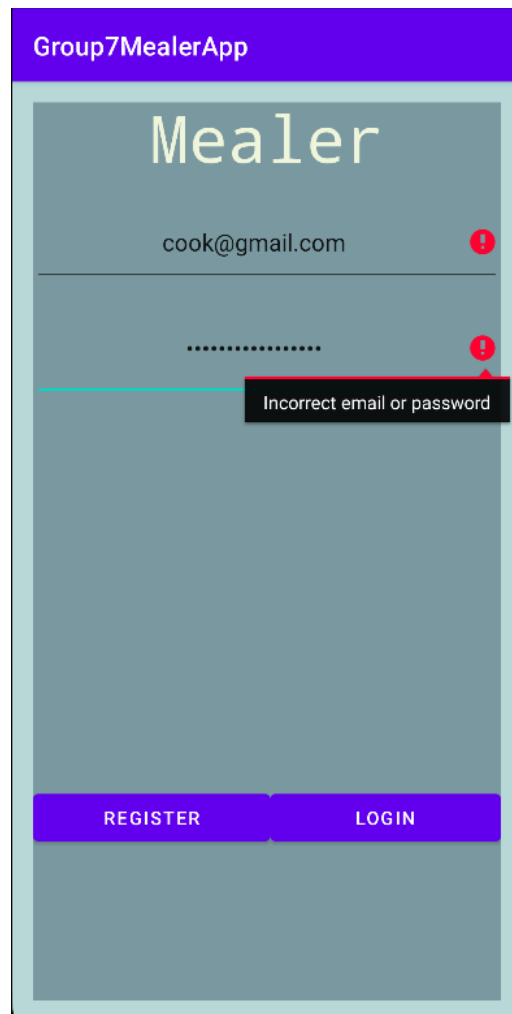


## App screen shots

The following screenshots are of all the pages implemented for the Mealer app captioned with what they exactly do.



Login page of Mealer. A user can enter their password and email to log in, or they can tap the register button to register a new account



If a user tries to log in with an incorrect email address or password, the login fails.

ACCOUNT REGISTRATION

First Name

Enter First Name

Last Name

Enter Last Name

Enter Address

Email

Enter email address

Password

Enter password

Confirm Password

Re-Enter Password

Account Role

☐ Client

☐ Cook

registering as cook will require a blank cheque picture, please allow camera access and take a picture

REGISTER

After tapping the register button, the user is taken to the registration page. A user can choose to register as a cook or a client

ACCOUNT REGISTRATION

First Name

Wilson

Last Name

Higgsbury

The Constant

Email

wilson@mail.com

Password

12345

Confirm Password

12345

Account Role

☒ Client

Wilson

Higgsbury

1234123412341234

01/23

123

☐ Cook

registering as cook will require a blank cheque picture, please allow camera access and take a picture

REGISTER

Sample info of registration as client

ACCOUNT REGISTRATION

First Name

Wilson

Last Name

Enter Last Name

The Constant

Email

wilson@mail.com

Password

12345

Confirm Password

123456

Account Type

☒ Client

☐ Cook

Wilson

Higgsbury

123412341234

01/23

123

☐ Cook

registering as cook will require a blank cheque picture, please allow camera access and take a picture

REGISTER

If the info is in incorrect format, the user cannot register and is shown hints based on the incorrect information they provided

## ACCOUNT REGISTRATION

First Name

William

Last Name

Carter

The Constant

Email

maxwell@mail.com

Password

maxwellgaming

Confirm Password

maxwellgaming

Account Role

☐ Client

☒ Cook

say pal, you dont look so good

registering as cook will require a blank cheque picture, please allow camera access and take a picture

REGISTER



ACCOUNT REGISTRATION

First Name

Enter First Name

Last Name

Carter

The Constant

Email

maxwell@mail.com

Password

maxwellgaming

Confirm Password

waxwellgaming

Account Role

☐ Client

☒ Cook

Enter Description

registering as cook will require a bio, a profile picture, allow camera access and take a picture of your kitchen

Field cannot be empty

REGISTER

A similar process when registering as a cook.



A photo of a blank cheque must be taken as a cook so the client can pay the cook.

Group7MealerApp


Mealer

example@email.com

Password

REGISTER

LOGIN

 Registration Successful. Log in now

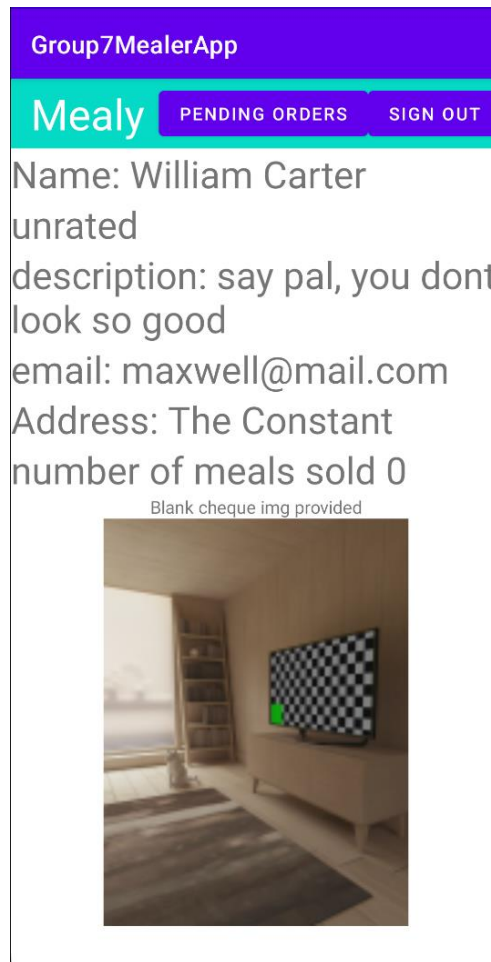
If the registration is successful, the user is taken back to the log-in page to log in using their newly registered credentials.

welcome,William Carter, you are a cook.

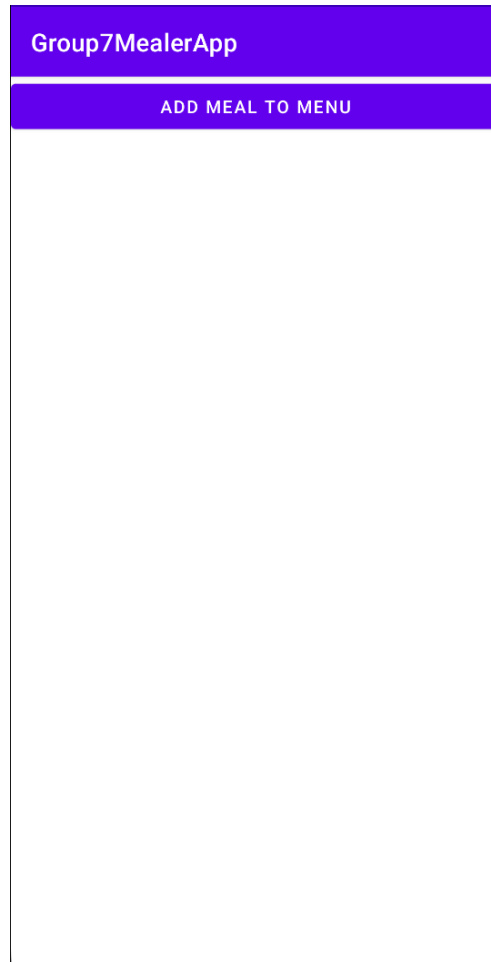
VIEW PROFILE

EDIT  
MENU

Home page of a cook.



After tapping the “View Profile” button, the cook entered their profile. They currently have 0 meals sold and don’t have any pending orders since no meals are offered yet.



After tapping the edit menu button on the home page, the cook is taken to the menu page. Currently, the cook doesn't have any meals to offer.

Group7MealerApp

Add Meal

Name

Monster Lasagna

Price

10

Meal Type

☐ Breakfast

☐ Lunch

☒ Dinner

☐ N/A

Cuisine Type

Don't Starve

Ingredients

2 monster meat, 2 filler

Allergens

monster meat

Description

owers your sanity. Can fill your stomach though.

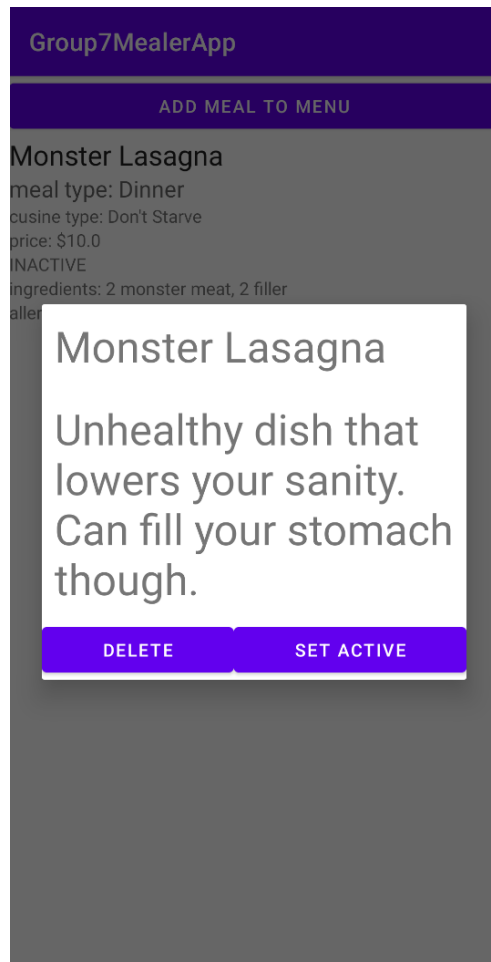
Do you want to offer this meal to clients immediately?

☐ Yes

☒ No

ADD MEAL

A cook can choose to add a meal to their own menu.



After the meal is added, the cook can see it in their menu and tap on it to view the detailed information. They can then choose to set the meal to active (offer it to clients), or they can delete it. A meal cannot be deleted if it is currently offered.



Group7MealerApp

Mealy

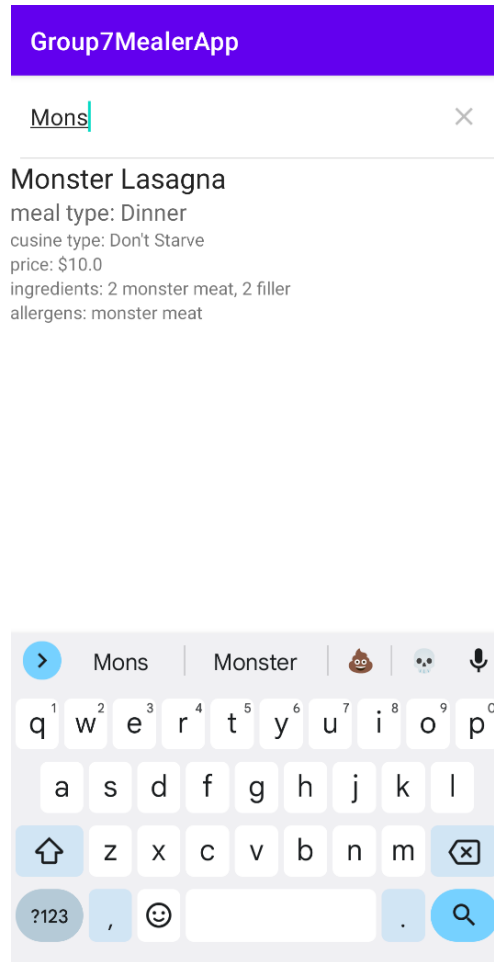
SIGN OUT

welcome,Wilson Higgsbury, you are a client.

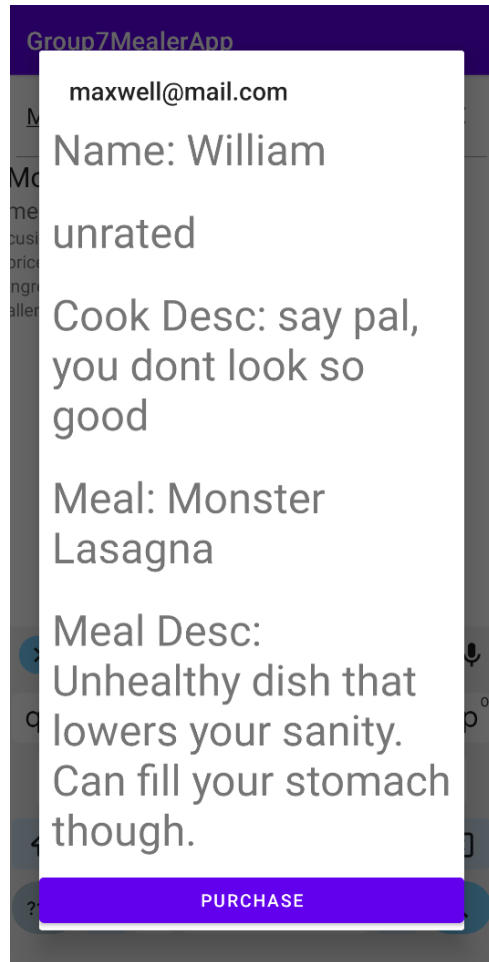
VIEW ORDER HISTORY

SEARCH FOR  
MEALS

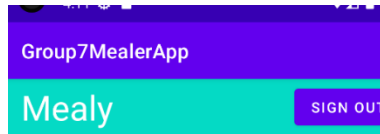
Now we log in as the client.



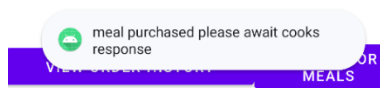
After clicking on the “Search for Meals” button, the client enters the search page and can search for meals that cooks have set as active.



A description of the meal is given, and the client can choose to purchase it.



welcome,Wilson Higgsbury, you are a client.



Now that the meal is purchased, the client waits for the cook to finish preparing the meal.

Group7MealerApp
Monster Lasagna Dinner Don't Starve 10.0 2 monster meat, 2 filler monster meat true maxwell@mail.com Unhealthy dish that lowers your sanity. Can fill your stomach though. maxwell@mail.com PENDING

If we tap on the “View Order History” button, the client can see the meals that they have ordered.

## Group7MealerApp

Monster Lasagna

\$10.0

wilson@mail.com

---

Now we log in again as cook and go to the “Pending Orders” menu, located as a button in the profile page of the cook. We can then see the pending order from the client.

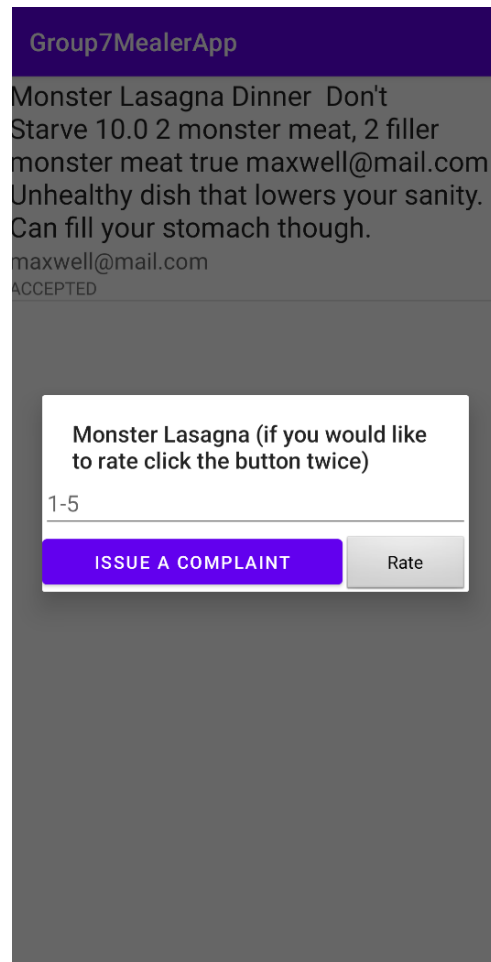
Monster Lasagna

Accept or Deny This Meal Request?

ACCEPT

DENY

The cook can choose to accept or deny this order.



After the cook has accepted the transaction, we log in as the client again. Now in the order menu, the order status changed from “Pending” to “Accepted”. The client can then choose to rate the meal or issue a complaint.



Group7MealerApp

Mealy

PENDING ORDERS

SIGN OUT

Name: William Carter

Cook Rating: 5.0


description: say pal, you dont look so good

email: maxwell@mail.com

Address: The Constant

number of meals sold 1

Blank cheque img provided



Logging in as the cook, we can see that the average rating has changed, and the number of meals sold has increased by 1.

We're sorry for your negative experience

Monster Lasagna made me lose 20 health, which wasn't included in the description. It only mentioned losing sanity. I would like to speak to your manager ASAP.

SUBMIT COMPLAINT

The customer can choose to submit a complaint about the meal as well.

**Group7MealerApp**

**maxwell@mail.com**

Monster Lasagna made me lose 20 health, which wasn't included in the description. It only mentioned losing sanity. I would like to speak to your manager ASAP.

The cook can see the complaint after clicking on their “complaints” button on the home page. Unfortunately, the cook cannot reply to the complaint to tell the client they are actually self-employed.

Group7MealerApp

Mealer

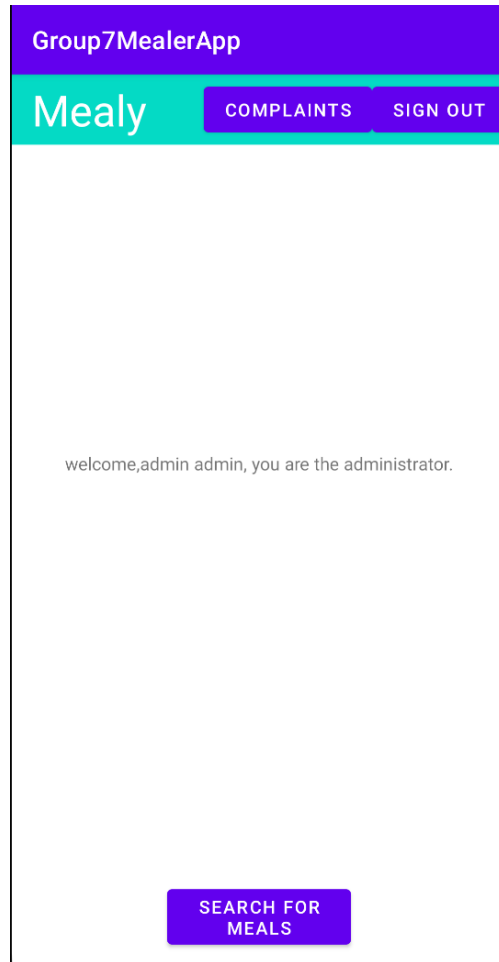
admin@gmail.com

.....

REGISTER

LOGIN

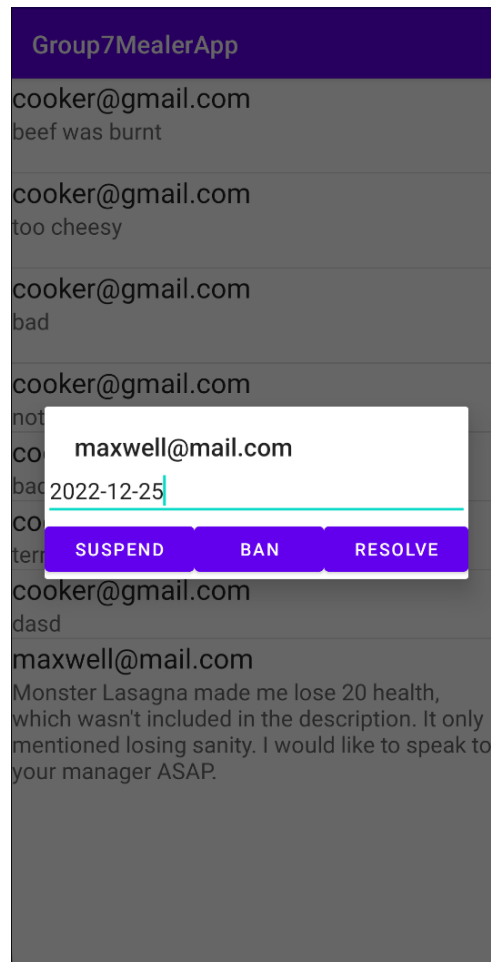
An admin logs in using a special email and password. They deal with complaints from clients.



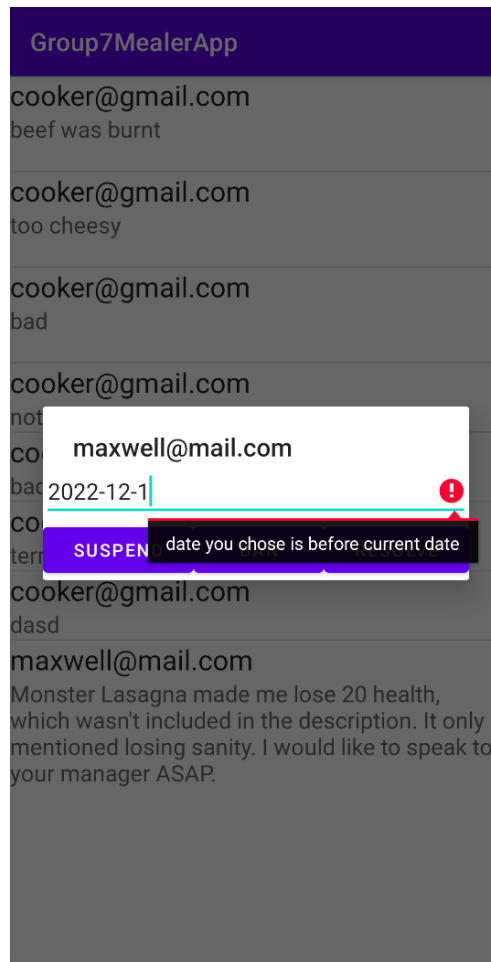
Home page of the admin.

Group7MealerApp	
cooker@gmail.com	beef was burnt
cooker@gmail.com	too cheesy
cooker@gmail.com	bad
cooker@gmail.com	not good
cooker@gmail.com	bad
cooker@gmail.com	terrible
cooker@gmail.com	dasd
maxwell@mail.com	Monster Lasagna made me lose 20 health, which wasn't included in the description. It only mentioned losing sanity. I would like to speak to your manager ASAP.

In their complaints page, the admin can see a list of complaints that every client has issued to a certain cook.



They can then choose to suspend the cook for a certain period, ban them permanently or resolve the issue.



The admin cannot set the date of the ban to be lifted to be before the current date.



Group7MealerApp

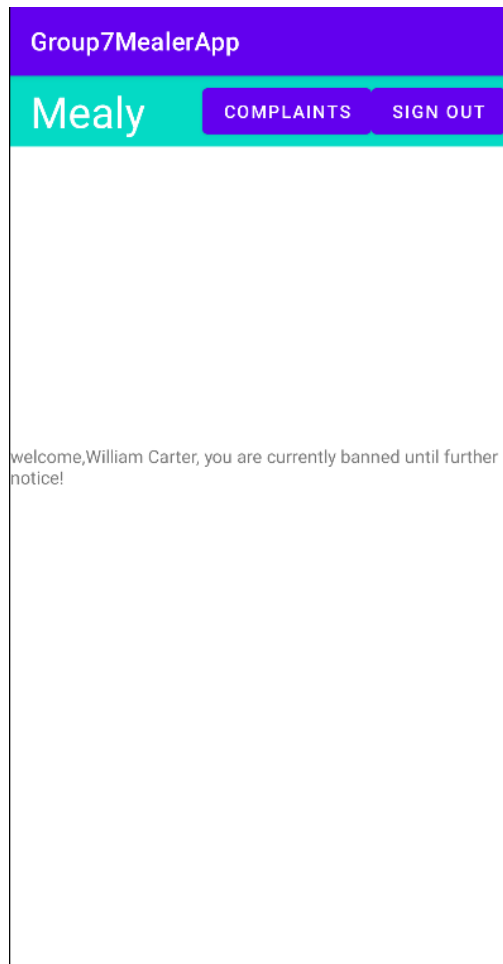
Mealy

COMPLAINTS

SIGN OUT

welcome,William Carter, you are currently suspended until Sun  
Dec 25 00:00:00 EST 2022

The cook can now see that they are suspended until the specified date.



If the “ban” button is tapped, the cook sees that they are banned until further notice,

If the resolve button is tapped, the complaint is deleted.

## Contributions to Each Deliverable

	Deliverable 1	Deliverable 2	Deliverable 3	Deliverable 4
Shezaad	firebase implementation, bug fixing, making sure pages navigate to the right place with proper information in tow	hooked up ban and suspension buttons, added error catching and hooked up everything to firebase (dealt with anything firebase related)	Helped around everywhere as well as did future work for deliverable 4, anything firebase related was dealt with as well as bug fixing	implemented search helped make the cook profile with reno made it so that meals could be purchased, dealt with anything related to firebase i.e purchase, ratings, orders etc
Zakaria	Created log out functionality that brings the user back to log in page and adjusted the welcome page so that it displays the users type and name.	Worked on frontend to make the complaint functionality accessible to only cooks and clients.	Developed unit tests for all the new functionalities we added & updated the UML diagram.	Worked on frontend for the client to see the status of their purchase request.
Renojan	Frontend and backend of the registration page. Includes error handling as well as appending registrants' info to firebase.	Backend of allowing the admin to suspend a cook.	Worked on backend for offered meals list as well as associating a menu to a cook	Worked on the frontend and backend to create cook profile as well as view pending client purchases and accepting and rejecting pending purchases (with Shezaad).
Karthikan	created the login page and iterated through firebase to validate the credentials	created a new java class called suspension and modified the cook class to make it	created the java class called meals and worked on some of the backend for	worked on the frontend and backend of purchase history and the

	entered and decide whether they are a user or not	use the suspension class	dealing with offered meals	functionality of rating and complaints of a cook based off meals the client had received
Chujian	build Java classes that represent the three types of users using abstraction, make UML diagram	developed unit tests for the previous deliverables, update the UML diagram so it includes all new classes	front end for adding and removing meals	developed unit tests, updated UML diagram, made skeleton of report

## Discussion & Conclusions (lessons learned)

For this project, we successfully completed all tasks for all four deliverables and our application is fully functional. During this project, we learned how to use Android Studio and Junit when we were conducting tests on the application. We learned a lot about software development and had team meetings to discuss and organize our ideas to implement them in our code. We also learned to collaborate as a team to fulfill the requirements of each deliverable by distributing parts evenly and helping each other when needed. A very crucial lesson was learning to properly comment on all our code and make it as clear as possible. Many times, over we would have to explain to each other how certain code snippets work but this can be totally avoided with clean code and proper commenting.

Specific Difficulties Encountered:

- An outdated version of Date was used at first for the credit card information. It had to be reimplemented using DateFormat, since many functionalities for Date were not supported anymore. We learned that it is important to reconsider using deprecated classes, since they might need to be replaced anyway.
- Creating tests individually for different classes was tedious but necessary, since we found out how creating new functionalities caused conflicts using tests. Automating Junit testing might make the process much easier, although in this project it was not implemented.
- Sometimes when merging between branches the merge conflicts would be resolved but problems would then occur on the branch you merged into. This

caused some of us to create a secondary branch but with this secondary branch merging was not an issue.

#### New Skills Learned:

- Interacting with and storing information in a database specifically firebase
- Learned how to use the tools necessary to create a basic program in Android Studio
- Using junit to test classes
- Using github to work on a project with a team