

University of Ottawa

Android Studio Project - Mealer

SEG 2105[B] - Intro to Software Engineering

Ryan Frost-Garant - 300114543

Alona Petrova - 300074852

Kevin Luong - 300232125

Kevin Yu - 300230560

Instructor: Wassim El Ahmar

Date Submitted: December 7th, 2022

Introduction

Mealer is an Ottawa-based meal sharing application that allows local cooks to sell meals to clients from their home. The application supports three types of users; client, cook, and administrator. The app allows cooks to create meals and add them to the offered meals menu. The clients are able to purchase meals from the offered meal list and then rate the cook and/or leave a complaint. The admin is then able to action this complaint by either temporarily suspending the cook, permanently suspending the cook, or dismissing the complaint.

Objectives

The Mealer project provided a good revision of the basics of object oriented java coding. It also ensured that all students were able to get familiar with material taught in the lectures, tutorials, and labs such as Android Studio, databases, Circle CI, unit testing, GitHub, and general communication capabilities for working in a group environment.

Platform

The application was designed using Android Studio with Java used as the language to program. The Firebase Realtime Database was selected to create and maintain the database, and Firebase Cloud Storage was used to store void cheque images provided by the cooks.

Major Project Steps

The project was completed in four major steps. The first step consisted of brainstorming the application details such as the way the user should be able to navigate the app. In addition, major classes necessary for the project were drafted using a UML diagram. The team focused on implementing the account management components where the users, Client and Cook, could register and log in to the welcome page. The second step of the application development focused on the implementation of administrative features where the Administrator could see complaints sent by the Client and suspend or dismiss them depending on the complaint nature. In this phase, the complaints were stored in Firebase Realtime Database to allow the testing since the main Client features were implemented later on in the project. It is important to note, that starting from the second step, the team has consistently implemented unit tests to ensure proper operation. The third major step of the project involved implementation of the cook features such as the addition of a meal, and deletion of the meal from the menu or offered list. Finally, the last step included implementation of the remaining features mainly related to the Client functionality. Upon completion of all steps, the app was tested for proper performance.

Benefit of The Project

The development of Mealer was an important learning experience for students and served as an example of some of the tasks that a software engineer may encounter during their career. This project also helped us put the material taught in the lab, tutorial, and lecture, to use in a group environment where we have to collaborate and coordinate who will work on what specific aspects of the project. This taught us to communicate with each other and to manage our time.

UML

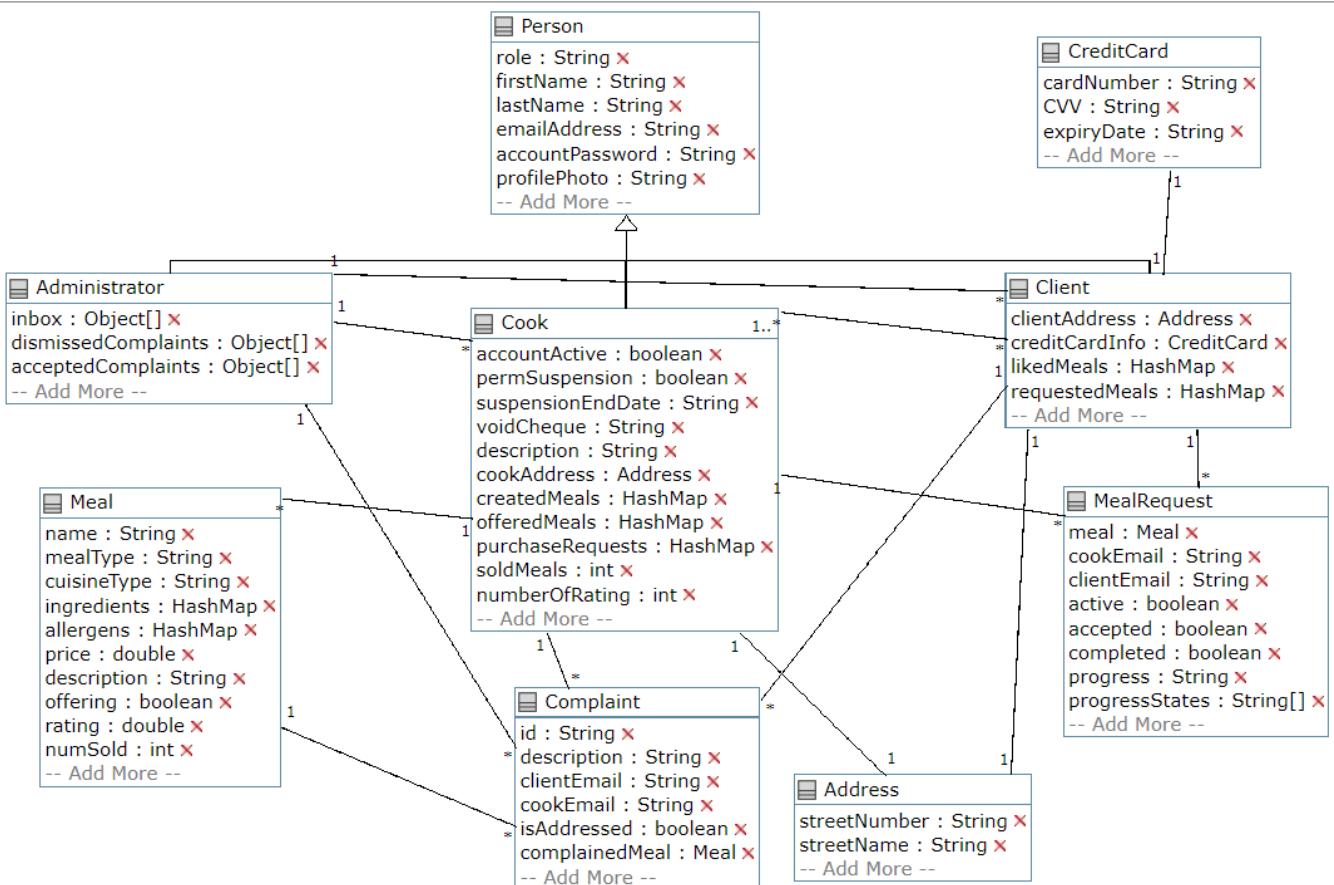


Figure 1. The Updated UML diagram for Group #5's Mealer project

Contributions For Each Deliverable

Deliverable 1	
Features Implemented	Completed By:
<ul style="list-style-type: none"> ❖ Set up necessary activities, fragment, connection between main (login page) and fragments to select a role and registration form. ❖ Void cheque image upload. ❖ Welcome message for users based on the data entered in the database. 	Alona Petrova
<ul style="list-style-type: none"> ❖ Created classes for Cook, Credit Card, and Address. ❖ Implemented input verification for the cook and client registration. ❖ Final UML Diagram. 	Ryan Frost-Garant
<ul style="list-style-type: none"> ❖ Connect and retrieve from Firebase Realtime Database (BONUS). ❖ Upload void cheque image to Firebase Cloud Storage. ❖ Input verification. 	Kevin Luong
<ul style="list-style-type: none"> ❖ Main user interface. ❖ Navigation buttons. ❖ User welcome page. ❖ Helped with void cheque image upload. 	Kevin Yu
<ul style="list-style-type: none"> ❖ Testing implemented features. 	Everyone
<ul style="list-style-type: none"> ❖ Creation of UML. 	Everyone

Deliverable 2	
Features Implemented	Completed By:
<ul style="list-style-type: none"> ❖ Complaint class. ❖ Created and added 5 complaints to the firebase. ❖ AdminComplaints created where Admin can view the list. ❖ Created action_dialog.xml to prompt the administrator to act upon a complaint. ❖ Implemented a feature to Dismiss complaints. ❖ Implemented a Back button. 	Alona Petrova
<ul style="list-style-type: none"> ❖ Updated UML Diagram. ❖ Implemented Unit Tests. 	Ryan Frost-Garant
<ul style="list-style-type: none"> ❖ Helped with Cook suspension. ❖ Set up a Cook suspension message. ❖ Helped with updating Firebase from complaint and cook suspension. 	Kevin Luong
<ul style="list-style-type: none"> ❖ Address, Credit Card, and void cheque field. 	Kevin Yu

<ul style="list-style-type: none"> ❖ Implemented action for Administrator Complaints (temporary/permanent suspension). 	
<ul style="list-style-type: none"> ❖ Testing implemented features. ❖ Input verification. 	Everyone

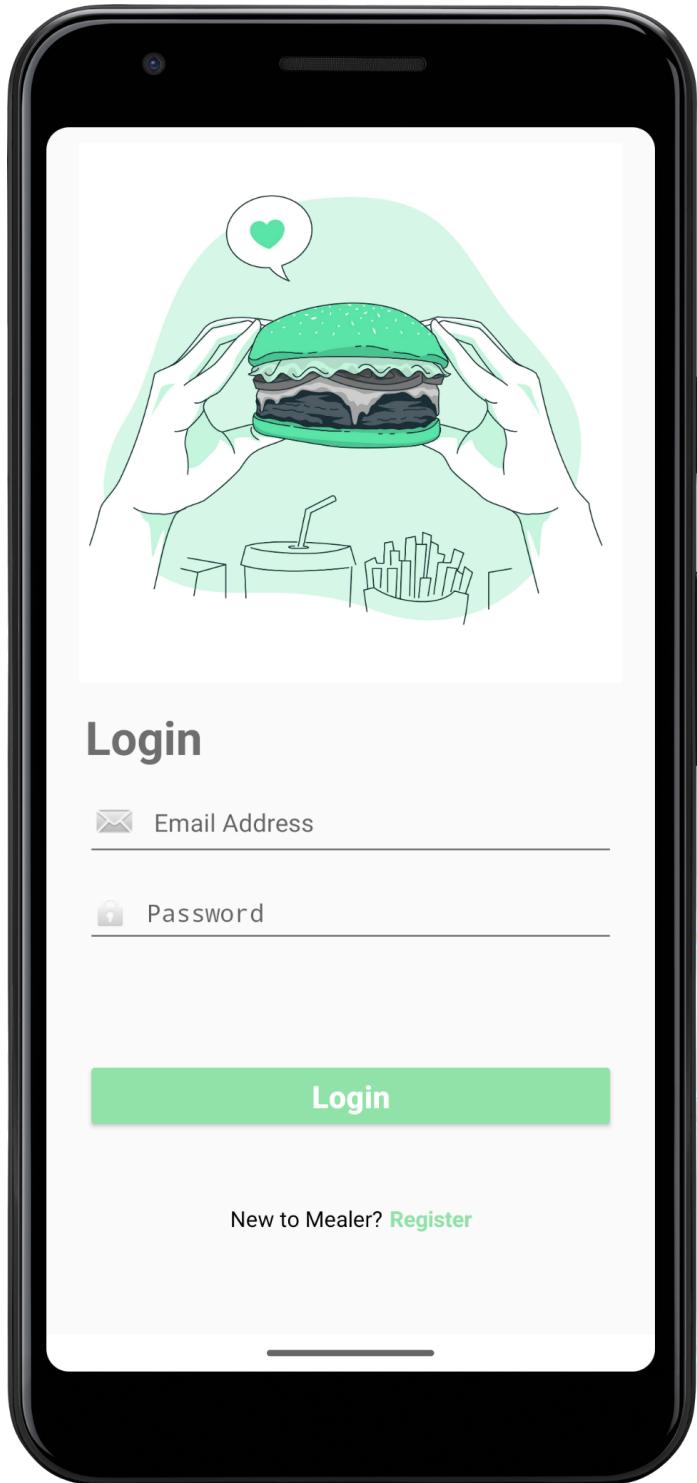
Deliverable 3	
Features Implemented	Completed By
<ul style="list-style-type: none"> ❖ MealActivity added where the information about the meal can be input. ❖ New meals are created and added to the database under their respective cooks (under tab meals). 	Alona Petrova
<ul style="list-style-type: none"> ❖ Input Verification for creating a meal. ❖ Implemented Unit Tests. ❖ Updated UML Diagram. 	Ryan Frost-Garant
<ul style="list-style-type: none"> ❖ Suspension message (from Deliverable 2). ❖ Meal actions (offer, delete, remove meal) from within the profile page. ❖ Created a profile page for the logged in user (includes all meal information and lists for ingredients and allergens). ❖ Created a page to display a meal's information. ❖ Added a bottom navigation bar. 	Kevin Luong
<ul style="list-style-type: none"> ❖ Created a date picker for suspension of cook (Deliverable 2). 	Kevin Yu
<ul style="list-style-type: none"> ❖ Testing implemented features. 	Everyone

Deliverable 4: <u>BONUS NOTIFICATION COMPLETED</u>	
Features Implemented	Completed By
<ul style="list-style-type: none"> ❖ Modified toolbar_profile to have an option complaint. ❖ Created an action dialog that prompts the user to leave a complaint. ❖ Set up the complaint creation and upload to the database. ❖ Verified the admin can action the complaints. 	Alona Petrova
<ul style="list-style-type: none"> ❖ Creation of meal requests by clients. ❖ Allow cooks to accept or reject meal requests. ❖ Updates made to cook, client for meal requests. ❖ Allow cooks to complete an order. ❖ Implemented Unit Tests. ❖ Updated Final UML Diagram. 	Ryan Frost-Garant
<ul style="list-style-type: none"> ❖ Implemented notifications for when an order was sent, approved, and completed (BONUS). 	Kevin Luong

<ul style="list-style-type: none"> ❖ Helped with submitting complaints. ❖ Creation of cook profile with statistics. ❖ Implemented a search function that is able to search by meal name, cuisine, or meal type. ❖ Implemented a meal list that displays all of the meals from a non-suspended cook on their welcome page. ❖ Helped with allowing users to place an order on meals. ❖ Helped with displaying a list of a client's orders (and allow them to see the status of their order). ❖ Implemented toolbars. ❖ Helped with implementing cook and meal rating. 	
<ul style="list-style-type: none"> ❖ Created and implemented rating bar for cook and meal. ❖ Edited input for date picker ❖ Allow clients to rate cooks that they purchased meals from 	Kevin Yu
<ul style="list-style-type: none"> ❖ Testing implemented features. ❖ Input verification. ❖ SEG2105 mealer final report. 	Everyone

Screenshots

LOGIN AND REGISTRATION



Select User Role

Cook Registration Form

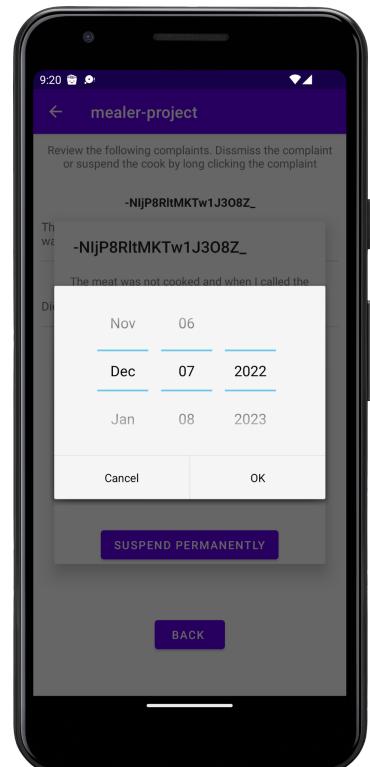
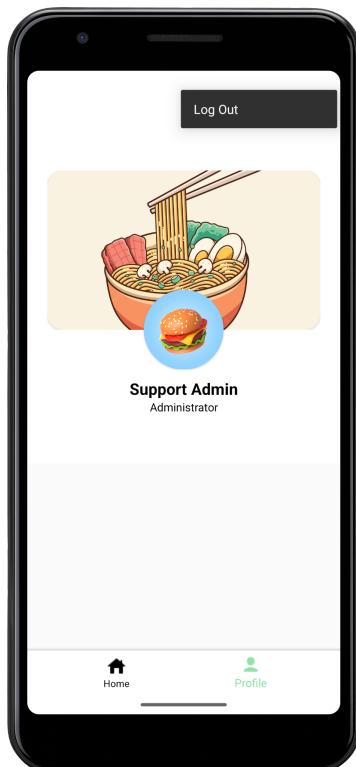
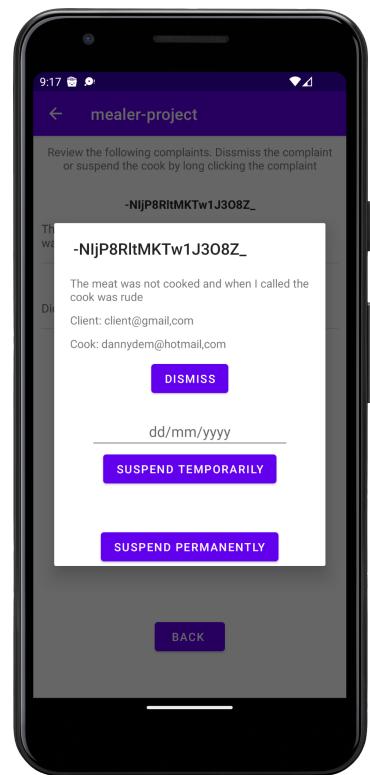
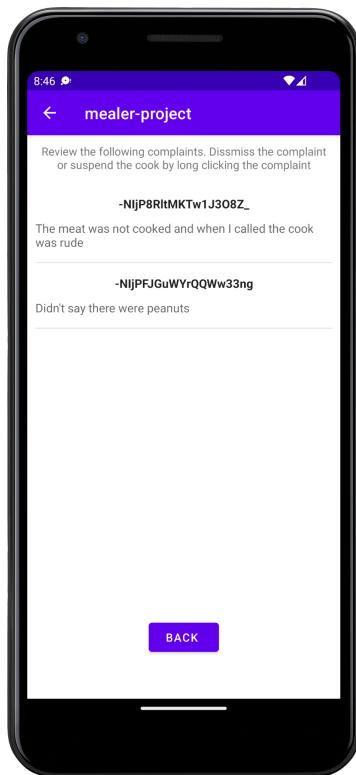
First Name _____
Last Name _____
Email Address _____
Pick-Up Street Number _____
Pick-Up Street Name _____
Description (Meals/Info) _____

Enter your password _____

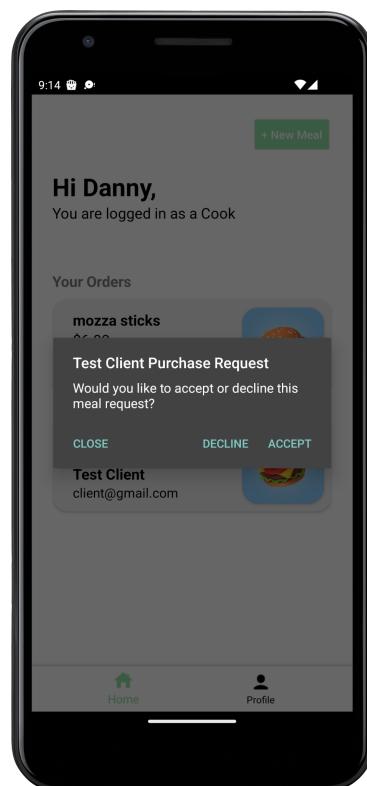
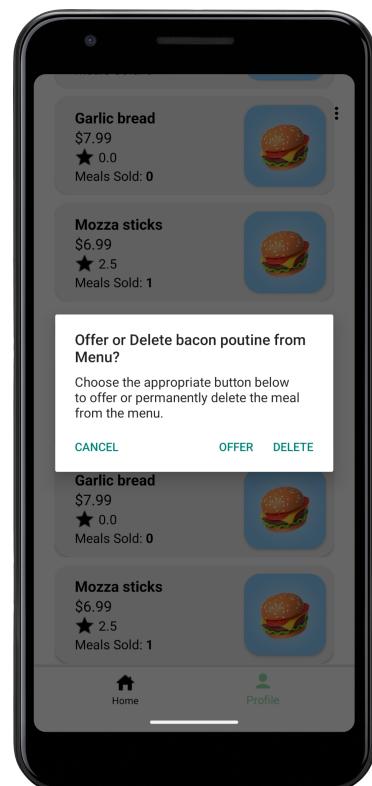
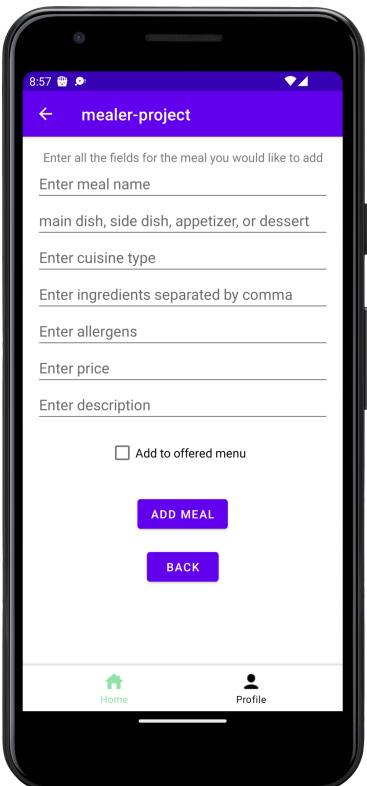
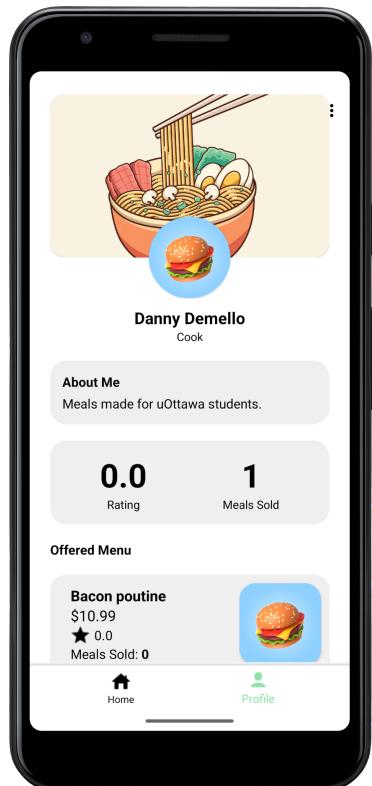
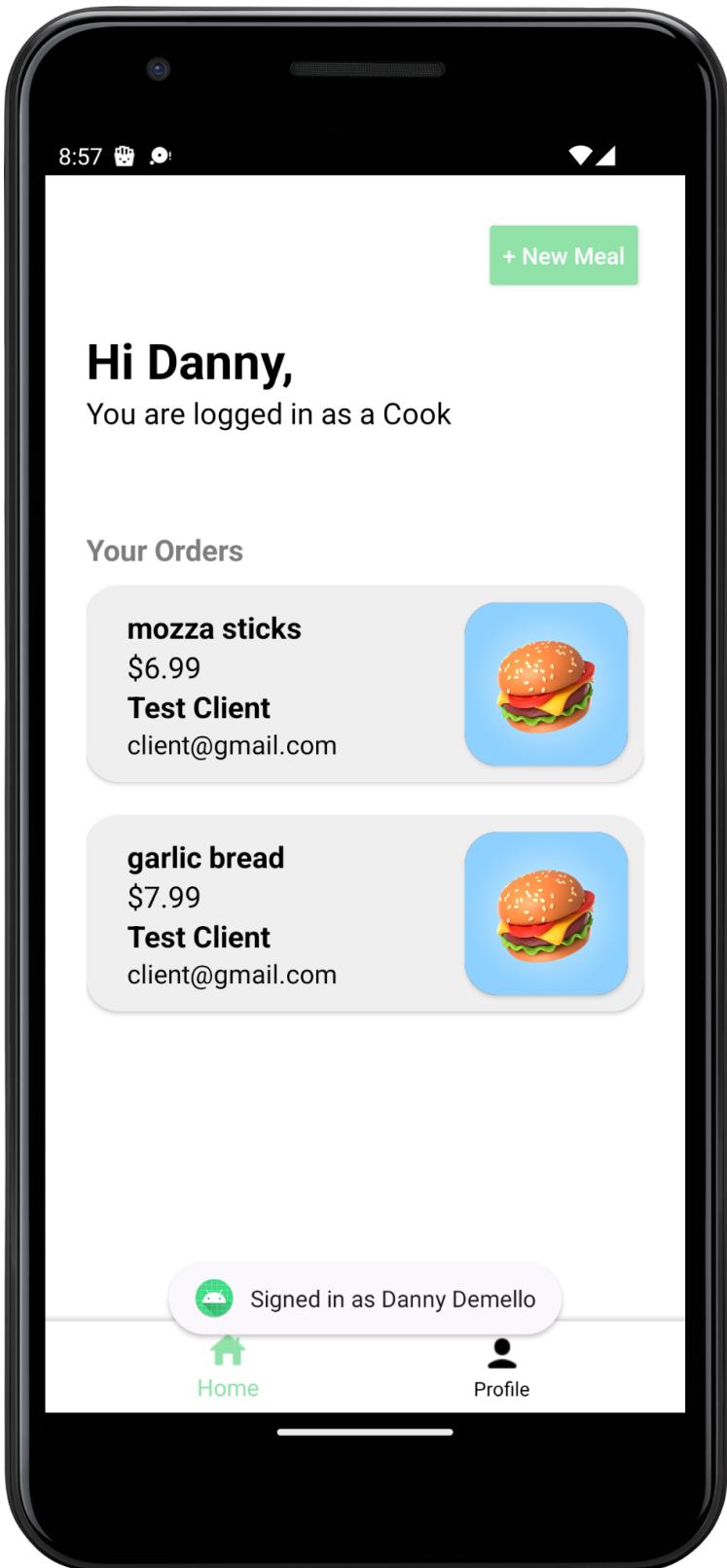
Client Registration Form

First Name _____
Last Name _____
Email Address _____
Your Street Number _____
Your Street Name _____
Credit Card Number _____
Credit Card CVV _____
Card Expiry Date (mm/yy) _____
Password _____

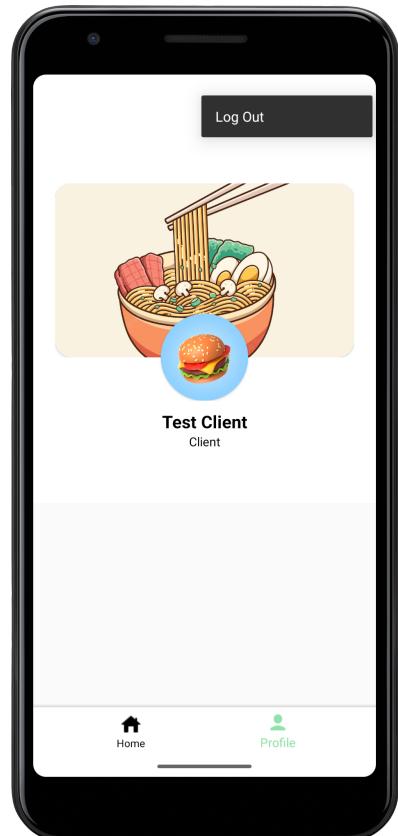
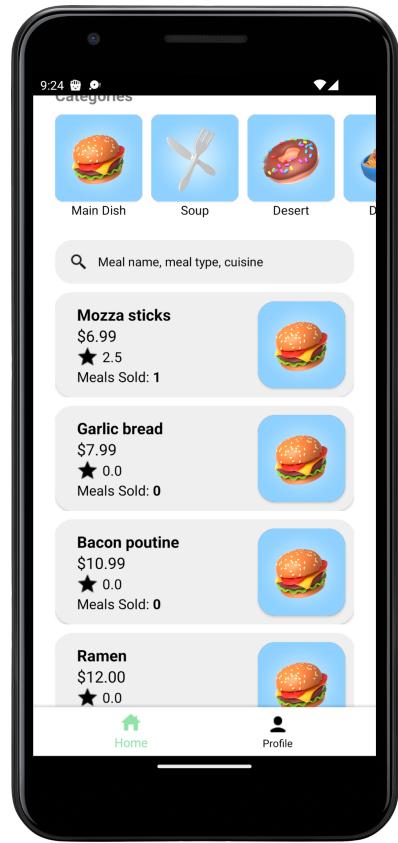
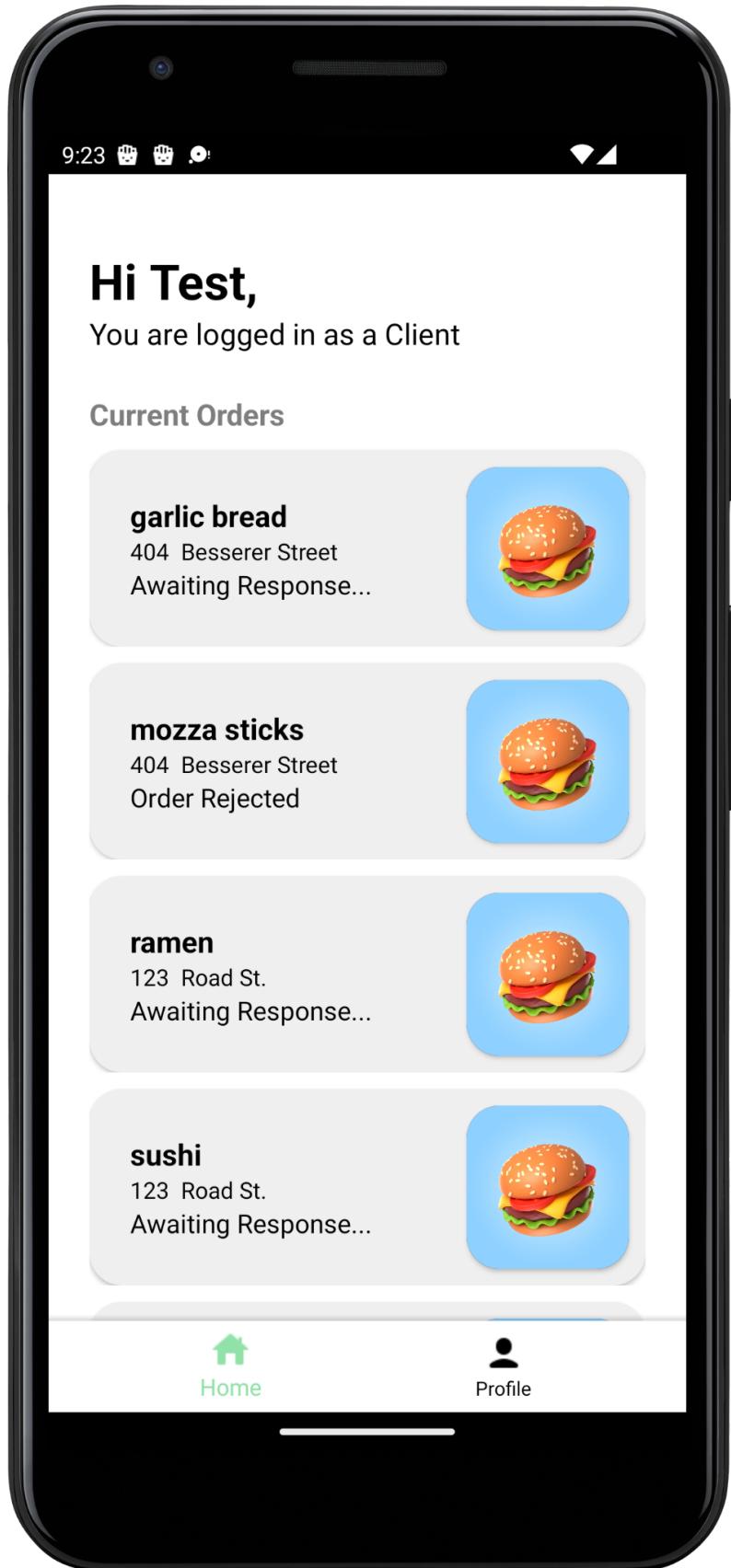
ADMINISTRATOR

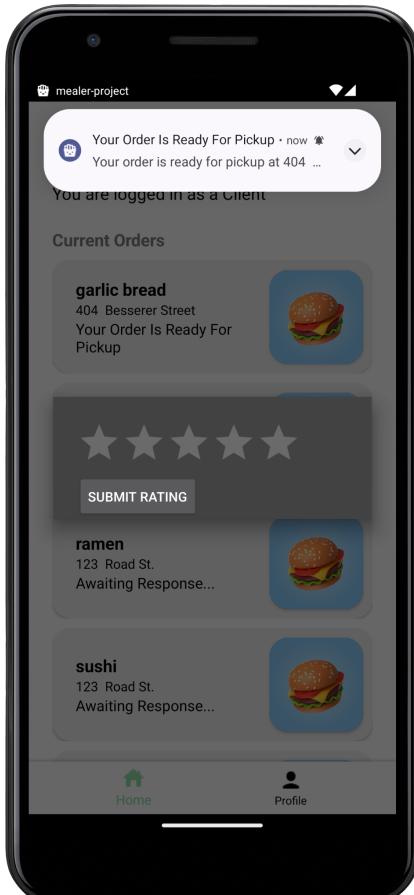
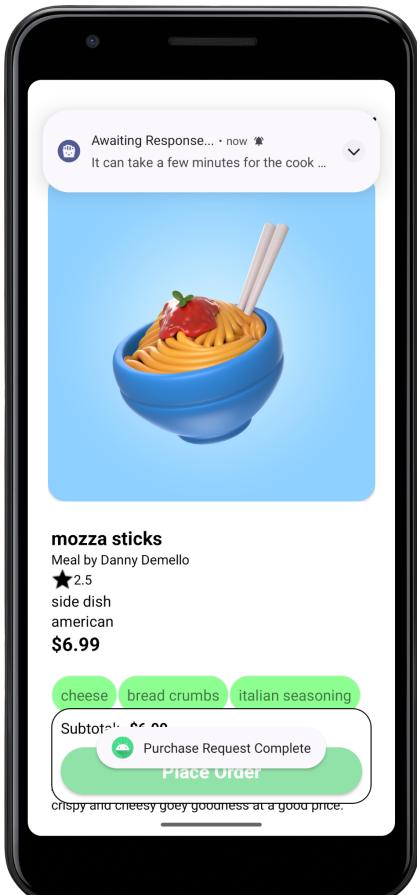
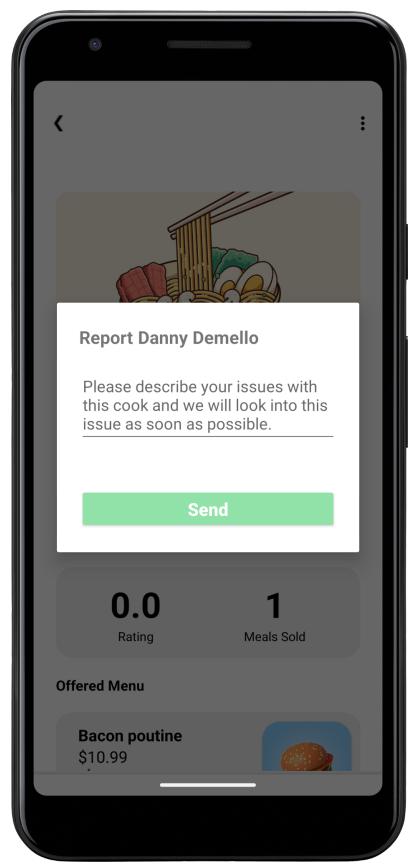
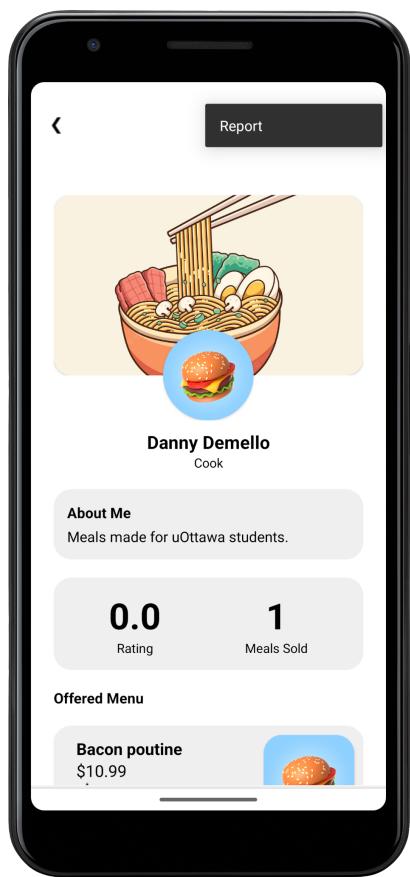
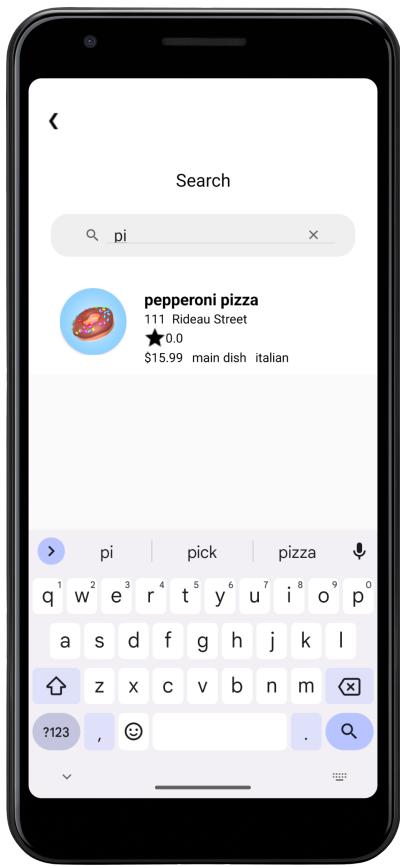


COOK



CLIENT





Lessons learned

The team successfully developed an application according to provided instructions. The key lessons and skills acquired from this project are

- ❖ Working with Firebase and connecting app to a database
- ❖ Adding activities and fragments in Android Studio;
- ❖ Getting familiar with Github;
- ❖ Setting up UnitTests;
- ❖ Working in a collaborative environment and ensuring proper organization and prioritization of the tasks;
- ❖ Learning to put functionality ahead of UI when initially creating an application (UI can be updated and changed later).

Challenges faced and resolved

Some of the main challenges faced during this project was learning how to use specific tools inside of Android Studio, an example of this is using RecyclerView to display information about a specific meal to a client, or about an order to a cook. Issues with displaying items inside of a RecyclerView were solved by researching and following tutorials that were available online. We also had to learn how to set up and use GitHub within Android Studio. This was a relatively simple task, but required some problem solving skills to figure out how to link your GitHub account to Android Studio. This was done using a temporary token created on GitHub which was then entered into Android Studio. There were also issues setting up Firebase to be used by all group members. One group member had to add a dependency in order for the Firebase Realtime Database to work properly that other group members did not have to add, the reason why is still unknown, but the problem was solved.