

Mealer Application

Fall 2022

SEG2105: Intro to Software Engineering
University of Ottawa

Course Coordinator: Wassim El Ahmar

Students

Ash Bhattacharai - 300236157
Abd Ennour Souit- 300256956
Andrew Pham- 300226985
Wilt Moise- 300241883

Table of Contents

Introduction to Project

UML Class Diagram

Contribution Table

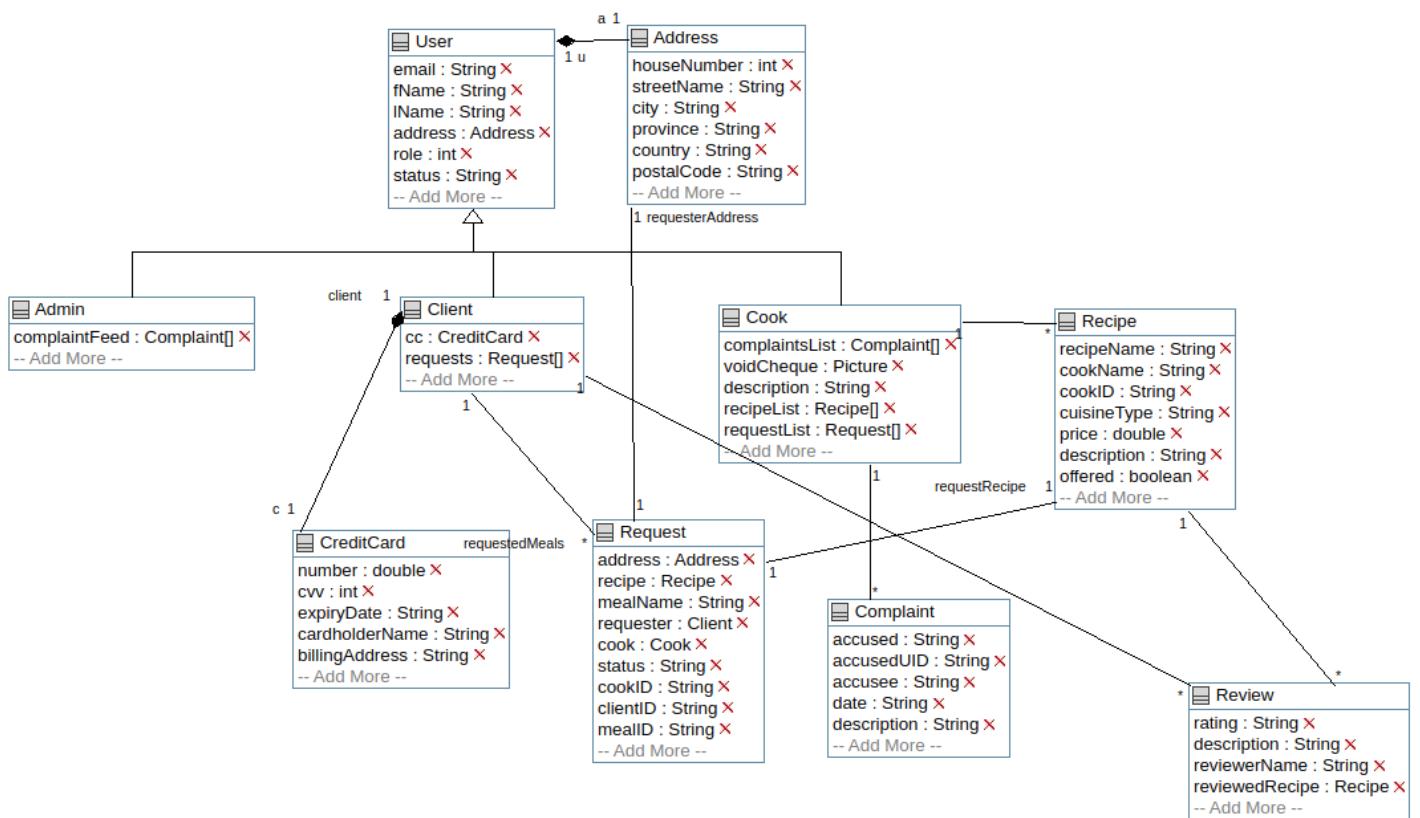
App Screenshots

Discussion/Lessons Learned

Introduction To Project

In this project, we created a Mealer application, which allows customers to purchase meals from home cooks. These home cooks prepare the meal at their own houses and list the meals for sale on the *Mealer* app. In our project, we implemented this concept using Android Studio to create the UI and used java for the backend work. Additionally, we used Firebase Authentication to implement the login process and utilized the Firestore Database to store the remainder of the data, including meals, reviews, permissions, meal requests, and different user permissions. In this report, I will provide the UML class diagram for our application, give a table specifying the contributions and work completed by each team member, display all screenshots of the app, and provide a short discussion section with lessons learned in the process of creating the app.

UML Class Diagram



Contributions

Below is a summarized list of contributions of each group member for each deliverable.

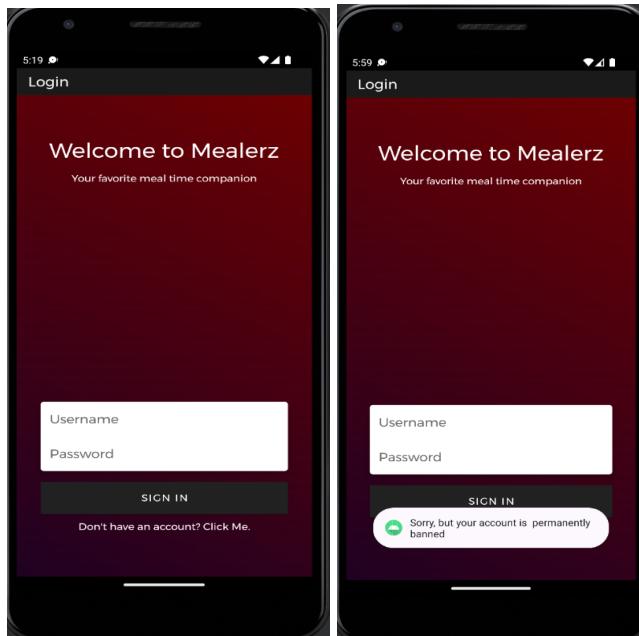
<u>Deliverable 1</u>	
Ash Bhattarai	-Created signup process and worked on firebase authentication -Worked on separating client login vs cook login -Created XML for signup activity
Andrew Pham	-Created client and cook classes in order to have user login be differentiated after login -Created additional java classes for storing information
Abd Ennour Souit	-Did the UI component and matched java methods to XML -Made existing XML designs more complete and matching with the colour scheme for login and signup
Wilt Moise	-Created login process and worked on firebase authentication -Created XML for login activity -Worked on storing user info in firestore database -Worked on separating client login vs cook login -Created additional java classes for storing information
<u>Deliverable 2</u>	
Ash Bhattarai	-Worked on differentiating ban scenarios of temporary vs permanent ban -Error checking for ban input from admin -Displaying of ban messages to cook -Created release v0.2 with APK
Andrew Pham	-Created Unit tests for testing functionality -Created UML class diagram -Updated remaining java classes to match functionality of admin role
Abd Ennour Souit	-Continued working on UI and implementing the required XMLs for each new functionality such as ban messages, complaints, etc. -Used the completed view adapters to display a list of complaints on the app
Wilt Moise	-Created view adapters to display various lists on the app -Worked and used firebase real-time database to store user complaints and admin ban features -Added functionality for the banning of users using Admin -Displaying of ban messages to cook in app

<u>Deliverable 3</u>	
Ash Bhattarai	<ul style="list-style-type: none"> -Worked on error checking of input from users to add/request meals -Worked on functionality to add meal to offered meals list -Created release v0.3 and APK
Andrew Pham	<ul style="list-style-type: none"> -Created Unit tests for testing functionality -Updated UML class diagram -Continued working on Admin functionality -Worked on functionality to add meal to offered meals list -Worked on ability to delete meals from menu and offered list
Abd Ennour Souit	<ul style="list-style-type: none"> -Created Navbar to traverse through app -Continued implementing UI for all the new functionality such as adding offered meals, menu, as well as the corresponding view for the client for each of the above.
Wilt Moise	<ul style="list-style-type: none"> -Added functionality to add meals to menu -Worked on functionality to add meal to offered meals list -Worked on ability to delete meals from menu -Added functionality to delete meals from offered meals list -Checking for suspension of cooks and displaying messages -Continued organizing view adapters
<u>Deliverable 4</u>	
Ash Bhattarai	<ul style="list-style-type: none"> -Wrote Final report -Created release v0.4 and APK -Worked on profile view of cook
Andrew Pham	<ul style="list-style-type: none"> -Created Unit tests for testing functionality -Updated UML class diagram -Worked on displaying of meal information -Worked on submit request and retrieval of these requests -Continued adding admin functionality such as
Abd Ennour Souit	<ul style="list-style-type: none"> -Updated Navbar to match new functionality -Added UI design for all new features such as viewing rating, searching, etc. -Worked on adding java methods into onClick functions on XML -Applied colour scheme and design to the UI to match the rest of the app
Wilt Moise	<ul style="list-style-type: none"> -Added functionality to search for meals -Worked on displaying of meal information -Worked on submit request and retrieval of these requests -Worked on review functionality and view -Worked on profile view of cook

Screenshots

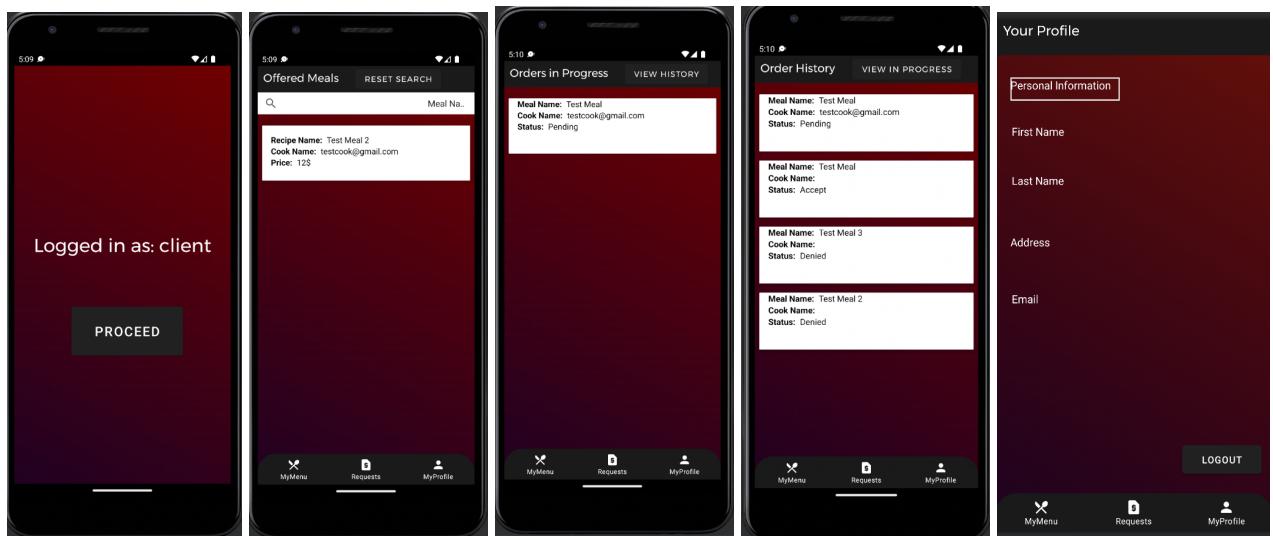
Login:

Below is the login screen for the application. The second picture is the login attempt of a banned cook.



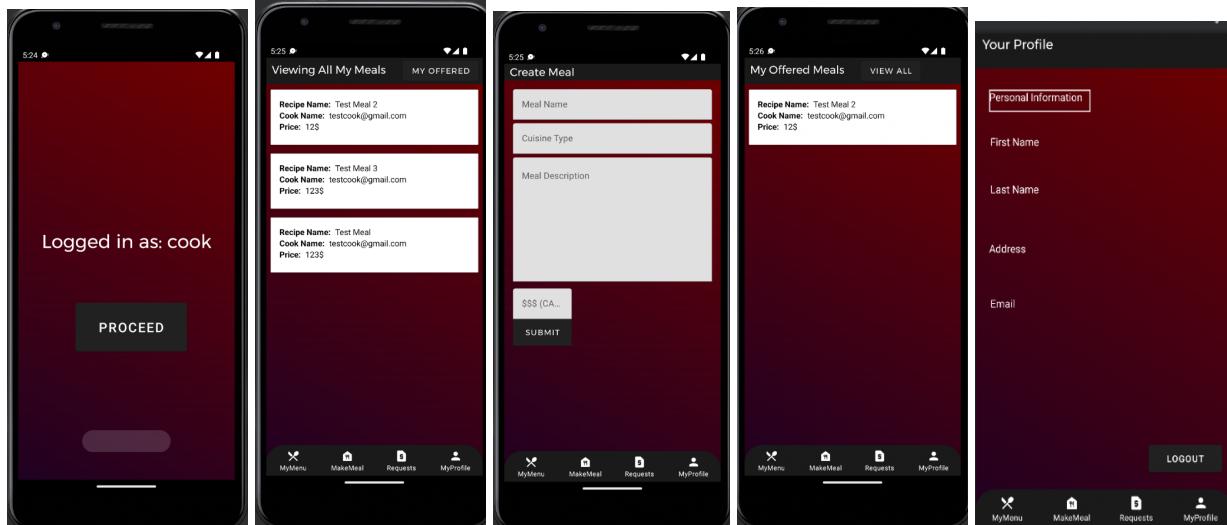
Client Side:

Below are the images of the login screen of a client, the homepage page with offered meals and search, the orders in progress, as well as the requested meals screen, respectively. Finally, logging out will direct you to the login screen as shown above.



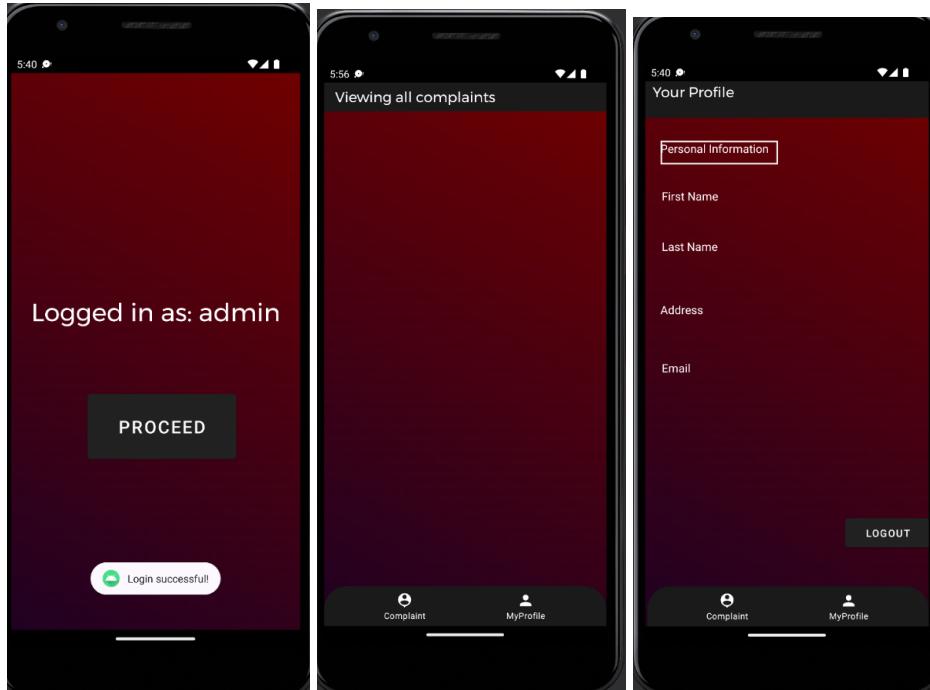
Cook Side:

Below are the images of the login screen of a cook, the homepage page with all meals, the offered meals page, as well as the create meals screen, respectively. Finally, logging out will direct you to the login screen as shown above in the first screenshot.



Admin:

Below are the login page for admin, the complaints page, and the profile page with logout. Logout will take you to the login page as shown in the first screenshot.



Discussion/Learned Lessons

While working on this project, we learned various critical programming principles that we also covered in class. While initially, this course looks entirely separated between the lectures and the labs/project, upon closer inspection, there are many links between the two. In addition, many issues we faced throughout the creation of the app were direct problems covered in class. In this discussion section, I will address those issues that we faced. In the following paragraphs, I will cover the programming issues we met, the lessons we learned about design, and the overall importance of modularity.

In terms of the actual programming aspect of the project, we learned many lessons for the future. For the first deliverable, everything mainly went smoothly. Aside from the research needed to use firebase authentication, the only issue we encountered was with git workflow etiquette, which was an important lesson. As we proceeded with the subsequent few deliverables, we began running into problems with the organization of java classes and proper id naming in XML. Upon looking back, we should have started using more abstract classes and interfaces that would reduce the number of classes. The lectures also taught this principle. Thus, we will proceed with these principles in the future.

Next, we also learned a great deal about the importance of design strategies. Unfortunately, design patterns and techniques were only taught in chapter 9 of the course, so we could not initially implement them. While our app is comparatively small, some of the design strategies covered in class would have been extremely useful. Specifically, a bottom-up design would have been the ideal strategy for us to use. Instead, we each individually did our parts of the project and had some difficulty merging the code. Therefore, we learned that a structured strategy is crucial for efficient programming.

This course's recurring concept of modularity was highly relevant to the project. We learned our lesson about modularity's importance and applied this concept to the project. We created more interfaces and abstract classes that created a foundation for everyone's code. These applications ensured minimal changes would be needed to each individual's code to run smoothly in the project.

In conclusion, the lessons we have learned in the process of designing this app will be extremely useful in designing other programs in the future. Furthermore, it will ensure that we will avoid repeating the same mistakes and make us better software engineers.