

Group 27: Home Food Delivery App Summary

Christian Gutierrez, Jordan Nguyen, Natalie Reyes, Wasay Ahmed

Home Food Delivery app provides users with the ability to order home cooked meals directly to your home. To make this possible there are three users that are crucial to this process, the cook, the customer, and the driver. The cook will be preparing the ordered meals within their own kitchen that the customer has placed. The driver will then receive an order on their end so they can deliver the meal to the proper destination.

We have all user and order information entered into the cloud database so it can be retrieved and used when logging in and retrieving orders. Android studio provides its own way of testing each activity in the developers documentation under test your activities which may differ from how traditional tests are run and created.

The UML diagram that was created represents the classes that were used within the project which are the Person, Cook, Customer, Driver, Food, and Order class. Below each class name is a description of what information is needed to create them. A symbol that is used within the diagram that represents composition and dependency is the solid diamond symbol for the Cook, Food, and Order class. The food class cannot exist without the cook class and the same applies to the Order class, an order cannot exist without the Food class therefore a diamond symbol is placed.

The application has been released in two parts, release 1 and release 2. The first release focused on the functionality of the Login, Cook user, and a basic mapping of the app. The mapping included all three profiles, buttons that can go to and from each XML activity page, and order pages we developed. The second release focused on the functionality of the Customer, driver, and Database. The Customer has the ability to view their order details and view nearby cooks on the CustomerMain class. After viewing available cooks the customer is able to see the Cook's menu using the CustomerViewCookActivity and the CustomerViewMenuActivity. The DriverMain activity worked in real time, functionality of drivers maps in certain locations worked, and start/end order was functioning correctly. Figure 1 shows the activity diagram for the app.

In comparison to the original project design the commonalities were the ability to utilize Google Maps API, rating system, login API, database updating user and order information. Differences include not having real time messaging capabilities, no cooked food recipes provided by homecook, did not include food calories, and we did not have a detailed ingredient list.

For our testing, we focused on the functionality of the app and tested the activities the user would interact with. By testing what would be interacted with, the backend methods would also be tested. We tested the main activity of all three user types and the login activity. In order to accomplish this, we tested all interactive features the user could use using the library espresso. In CookMainActivity, buttons such as startOrder and orderDetails were tested by comparing results from the database after the button was pressed. This same logic was applied to all other activities. We found no bugs or errors while testing all the functionality.

For our inspection, we choose the complex parts of our code to be inspected including Login Activity, Register Activity, Cook Main Activity, and Driver Main and Map Activity. To have a good inspection, we followed a guide by Queen's Computing. Common problems seen across all inspection reviews were lack of documented code, bad variable names, and lack of encapsulation/helper functions leading to lengthy code which is hard to decipher. The improvements are straightforward, we should add comments to the code and choose better variable names that are more specific. To improve encapsulation and readability, we should create helper functions of repeated code so the logic is easier to understand.

The main open issue for this project are the legal questions for the cooks and the overall business. The Home Food Delivery App business must research the Cottage Food Laws of the specific states that it wants to operate in and help cooks meet the regulations. The app requires a payment system for all three user types to properly function. To lower operating costs for the app, developers need to lower the amount of database reads and writes in the app once the number of users scales up. The appearance of the app and the UI experience should be remade and must look and feel nice for all users. The Home Food Delivery App must have an iOS version that works together with the Android version without any issues.

Working on Home Food Delivery App made us realize the importance of planning software development. Even if we don't know exactly what the client or in this case ourselves want. There should always be a far more comprehensive plan of a software project that every member understands for a better time developing and to save time. The plan should include a design pattern that the developers agree upon and must follow throughout development. For applications, most of the UI should be made first to make it much easier for the developers to fill in the functionality. UI changes should be minimal during the development unless required.

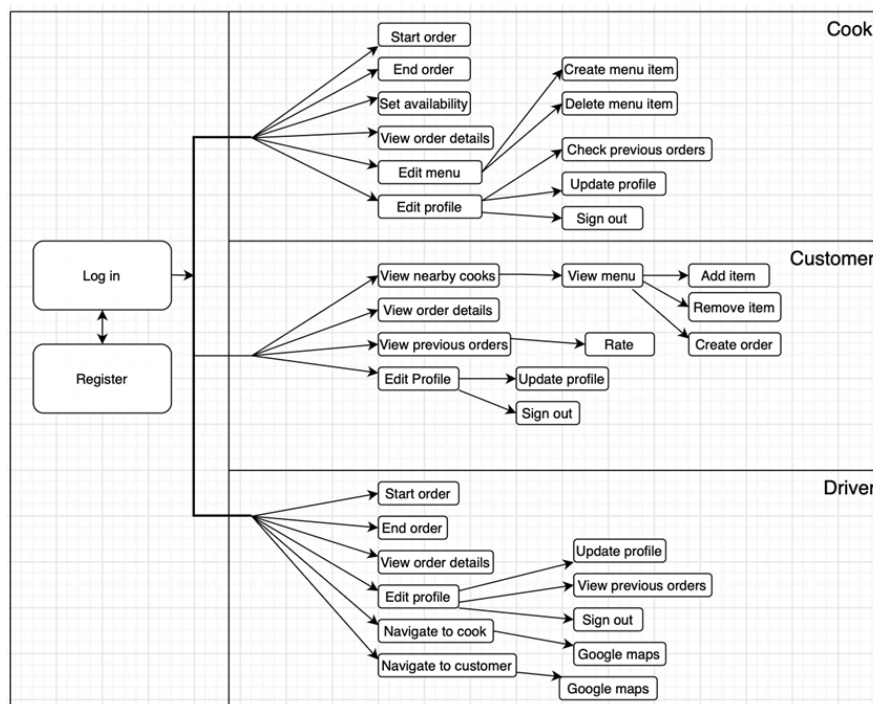


Figure 1. Activity Diagram for Home Food Delivery App.