Dishes

Topic and objectives of your Mobile Application

My App is based off the UBER model, but instead of anyone becoming a taxi driver, anyone can become a takeaway! People who are interested in becoming cooks, have to register whereas anyone looking to order food, can easily do so without any registration. For ordering, a user enters their address, selects how they want to find a dish, and enters what they're looking for. Once they've found it, and select it, they're order is placed and the applications shows the route of delivery.

Delivery is handled by the Cook. They can deliver themselves, or use a delivery partner (Deliveroo, someone they know, etc.). Payment is made at time of delivery for now. In the future, I plan to take payment through card or PayPal that will link to the cook's PayPal account. Dishes would then take a certain %(2-10) of each order before the transaction is complete. The cook bears this charge.

Target Audience

This app is targeted at users 16 - 70, and cooks 18+. Anyone that wants to order food can avail of this app but if the user wants to become a cook and start making an earning from home, they need to be 18+. The cook side of the app can appeal to all, but would probably appeal most to stay at home parents, retirees or people who might have more time for preparation(And know how to cook!).

Rationale behind Development Approach (hybrid or native) Technologies/Techniques used

I wanted to create a native application that was self contained.

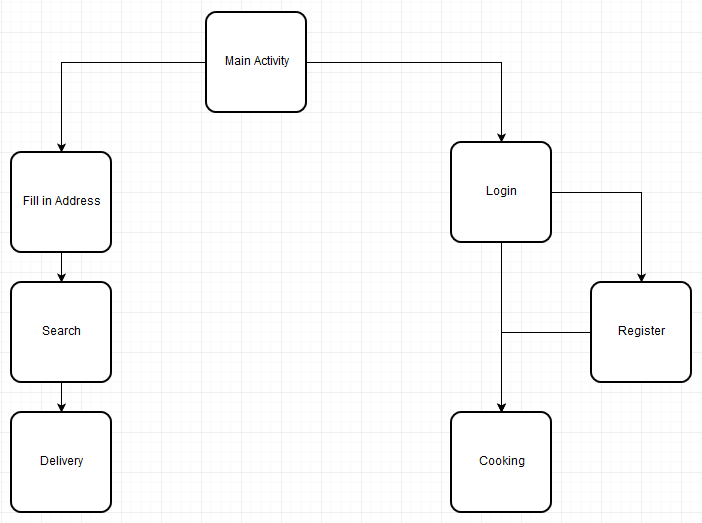
* I used SQLite to store cooks details
* I used JSON to read a file from my github account
* I used Google Maps API to show the origin and destination of delivery as markers on a map
* I used Shared Preferences to save and pass information from one activity to another, much later activity.

Back End

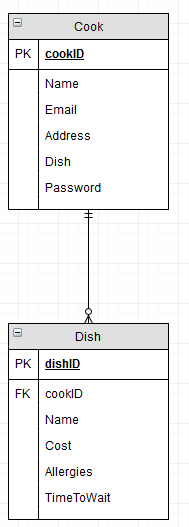
I use an internal SQLite database to store the applications data. I use this database to hold cooks information and query each record as well as authenticate cooks. This works as a temporary solution. I would like to migrate to a fully functional Firebase service in the future.

Currently, there are some hard coded cooks inserted into the database whenever the application is started. When a user registers, their details are saved in the SQLite database. These details are queried when this cook logs in and are also queried when a user wants to order food. Once the android emulator is closed, the newly registered data is destroyed.

Structural Hierarchy



Future ERD



Strengths

Read Json data

Query Database

Login + Built in Registration Authentication

Shared Preferences

Weakness

Internal SQLite Database

Error Handling + Validation

Google Maps functionality

Testing

Problems Encountered

Database - read + write + query

Lists - on item select

Google maps - import, place 2 markers using addresses, converting to lat lng

I could not get the Google maps API to import correctly. If you check my Delivery.java class, you will see that i have tried to create a map fragment using Google maps within my application using the two supplied addresses. I convert these address into Latitude and Longitude values and try to create map markers from them.

Future features

Google maps - route laid out between dest + orig

More structured db - dish table

Fully functional Firebase service

Error handling, validation checking

Link to delivery 'partners' for cooks use.

Add cook stats of earnings to entice more to join.

Upfront Payment Service

Hashing of cooks passwords

Rating system, so the good cooks naturally rise to the top.

Conclusion

Project Log

References/Bibliography