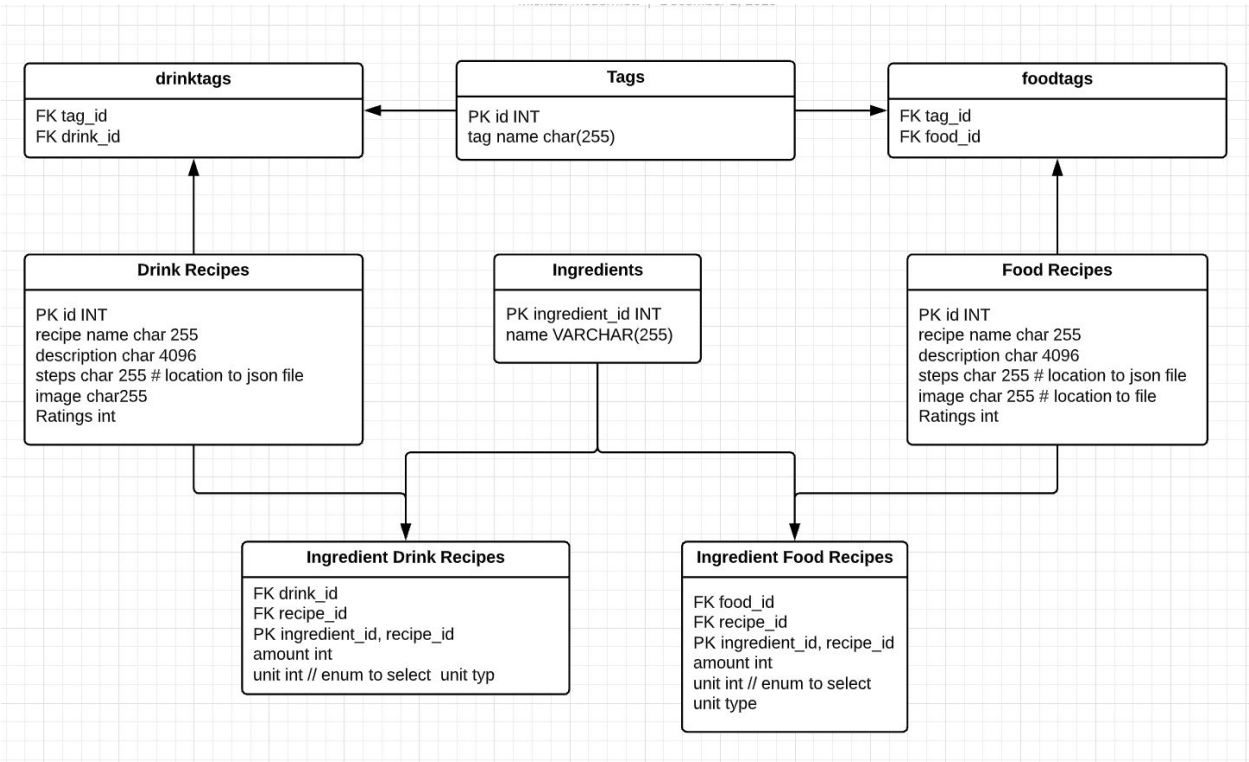
Personal Project Write-up

In this project, I have contributed mostly on the backend part of the project. This includes the use-cases with the backend, the design of the backend, and the implementation of the backend. The use-cases that were thought of were primarily about accessing data and the procedure of sending information back to the user. The design of the backend was modeled through an ERD Database diagram using Lucidchart, then implemented through CloudSQL using MySQL as the database language. For serving the database resources to the client, we went with a REST-like architecture implemented in Flask, using Python.

The use cases I helped developed mostly dealt with the interaction between server resources and the client. In this part of the project, I wanted to ensure that the interactions between the resources and the client were secure. The user should be able to access the data in the database, but not be given direct access. This provided a layer of protection were the information requested would be validated before accessing the database. To solve this problem of validating a response from the client, we decided to create a REST-like service. This service will serve the content of the database to the client and be able to validate each and every response sent to it.

After deciding on how the user interacts with the database, I started working with Micheal on how the database should be represented. We decided on a relational database architecture, since most of our data had a relationship between each other. Another factor in deciding on the architecture was the availability of knowledge. We then proceeded to develop

an Entity Relationship Diagram (ERD) for the database.

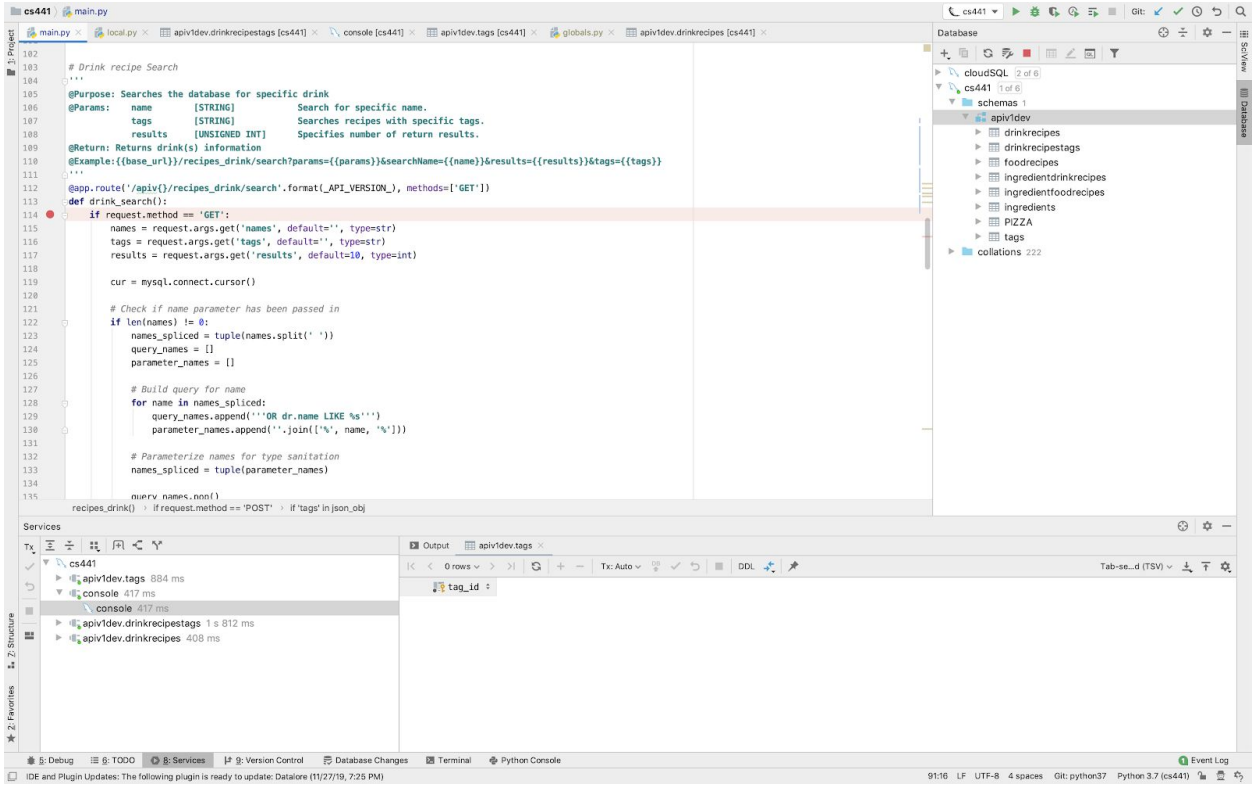


Once the ERD was created, I started working on the REST-like service using Flask. In this service, I was focused on interpreting the request from the client and giving an expected result. Most of my code was focused on the GET method of the drink recipes tag. This included how to sanitize the input from the user to prevent SQL injection, how to present the data to the user, and creating documentation/testing the route.

Most of the project time was spent on learning how to use Flask, connecting a MySQL database, and learning about REST services. Most of the learning material I consumed was direct documentation from Flask framework, W3School's information of SQL, and also StackOverFlow questions on security and optimizations in Python.

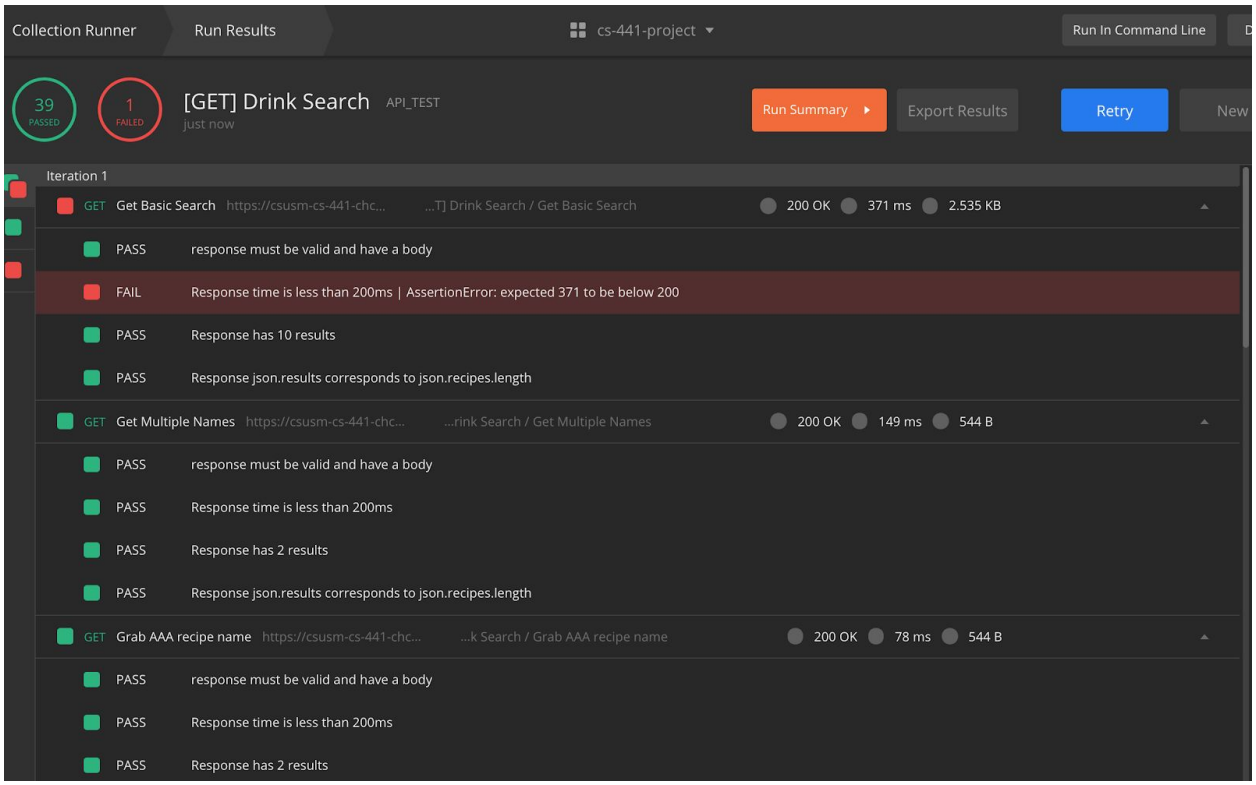
For developing this service, I used PyCharm Professional to help me develop this portion of the project. It allowed me to connect, view, and submit changes to the database hosted on the Google Cloud Platform (GCP). It also gave me the option to deploy to the app services of

GCP.



In testing our routes created in Flask, we used Postman to help automate the test. This ensured the information requested by the user ended with a correct result with correctly formatted data. We also designed our service using Postman, to help us see what the client

should have received when interacting with our services



After this, I still plan to refine my routes and work on POST new data to the database.