

Midterm Personal Project Writeup

For the Whereerbottle app, we all created specific use cases in order to maximize our collective effort in creating an app with a significantly large scope that would remain scalable. For this project I learned about UML design and the importance of creating a design for specific use cases. As the scope of our project increased at the beginning it was very easy to forget about some aspects of our project amongst everything we'd all worked on. Without the UML design and specific use cases with detailed descriptions which were very simple in their definition of steps involved, it was very easy to see where holes would develop in our app if we didn't use these elements as a guideline for development. I also learned more about version control and using a style similar to git flow; in which we develop in our own branches before pushing to production. Fortunately I was the primary coder working on the backend development of the app, so merge conflicts weren't a significant concern within the code that makes up our server, database and API calls. As far as technologies are concerned, I learned about the following things: Javascript and asynchronous execution of code within a javascript application, Mongodb, the non relational database which stores information in the form of an array of JSON strings, how to create a server that listens for specific http requests, Ec2 the amazon instance in which a linux machine runs our server constantly, and finally, Pm2 which is a node manager that allows us to keep our production server running constantly and monitor requests coming into our server.

Our group decided that Whereerbottle would be a cross platform app that runs standalone with a backend that would store information in a database and send and

receive information from clients(users using the app on their mobile devices). This required us to design, organize, and maintain a database, create a server that would accept requests from the user, create or use existing API's, and connect our app to the database by using some request protocol. My primary contribution to this project was rooted in the development of the back end of Whereterbottle. Firstly, we decided that we were going to utilize xamarin forms in Visual Studio in order to create an app with cross platform capabilities. Based on this we looked for easy ways to populate objects in classes, with information from a database query. Mongodb, a nonrelational database, which stores objects in the form of json strings which are called documents. A query to this type of database returns one of two things, an array with one json string which can contain a whole or partial document stored in a collection, or an array of multiple documents which match query parameters specified in client request body. Based on the classes that were developed for use cases, I designed the JSON documents to be stored in the collections within the database we would maintain. Seen below:

Bottle: [{"id":ObjectId(""),"size":"","total_refills":"","last_refill_day":"","day_refills":"","x_coord":"","y_coord":""}]

Fountain: [{"id":ObjectId(""),"x_coord":"","y_coord":"","filter_staus":"","rating":"","num_ratings":"","coldness":""}]

User:[{"_id":ObjectId(""),"first_name":"Forrest","last_name":"Joy","bottle_id":"","last_fill":"","favorites":[""],"friends":[""],"email":"","address":""}]

Then, having inserted those documents into collections, I then designed and created the API calls which insert, edit, remove, and retrieve documents in the database. I developed a restful API using Node js and a mongodb driver available for javascript. Using a mongodb makes the documents in the database very portable and scalable as it is easy to add fields to documents as necessary. The API developed

Forrest Joy
CS443

allows our app to communicate with the database regardless of what device it's run on.

If we developed a web app eventually, the API could easily be implemented on that as well. Therefore, in keeping our mobile app supported by a database, not required to store much information on the mobile devices, My contribution helped make wherterbottle scalable, and lightweight and cross platform.

Server code:

```
const url = "mongodb://localhost:27017/";

app.listen(3000, () =>{
  console.log("Server running on port 3000");
  ///////////////////////////////////
});
```

GetuserbyId route:

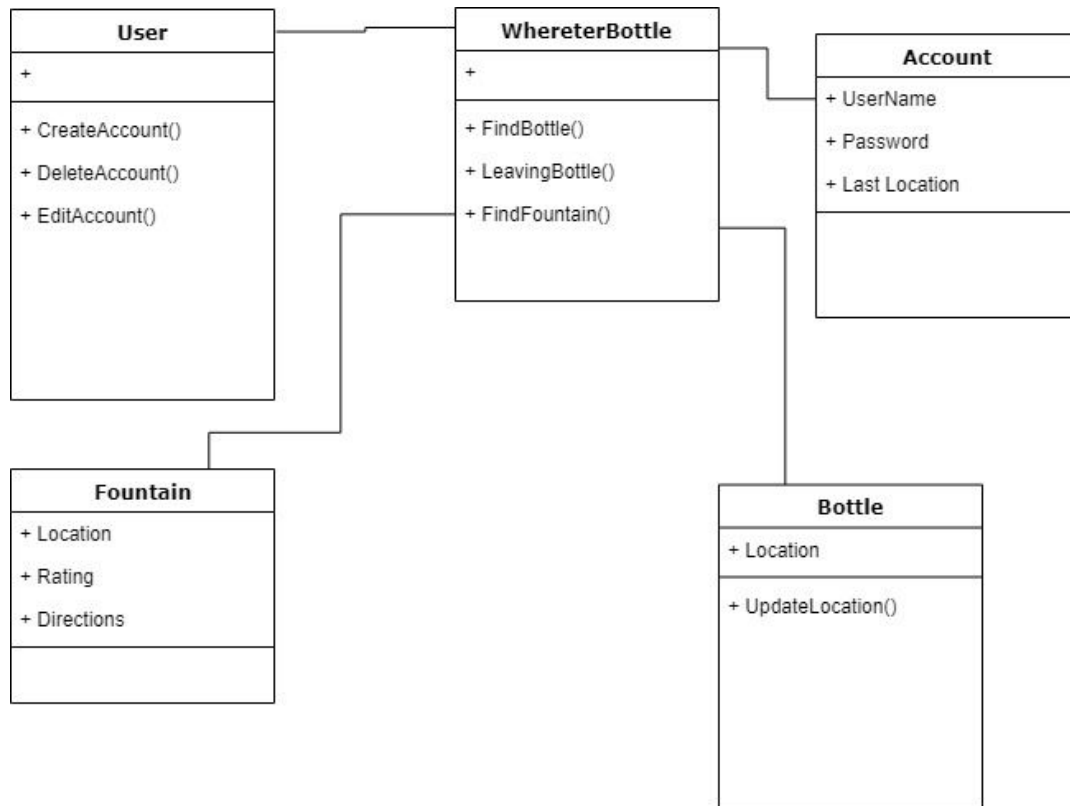
```
app.post("/getuserbyid", (req, res, next) => {
  clientIp = requestIp.getClientIp(req);
  console.log('body: ', req.body);
  console.log('IP: ' + clientIp);
  payload=req.body; //search by all parameters given in
  connectionHelper.getUserbyid(payload, res);
}); //???
```

As a group, we created more use cases than we intended to finish, so that we would not have a struggle expanding our scope if we'd been able to finish the project before the due date. The use cases that I created and helped refine include, add friend, delete friend, and delete fountain. However, all use cases that require any communication with the database I contributed to. Moving to the UML design, I helped contribute to the first UML design with the use cases, and after the first sprint, I began to focus primarily on the development of the database and API.

Forrest Joy

CS443

UMLv1:



Coding was my primary focus after the first sprint of our project. I developed the helper class and database queries, as well as the files that establish the connection of the server to our database and maintains and manages all open connections. These Classes were developed in Javascript, seperate from the c# files that were completed in xamarin. My primary focus for the rest of the semester will be cleaning up code, and refactoring to make my code more efficient by eliminating unnecessary objects and code within the API layer of whereterbottle.