Personal Project Writeup

I learned a lot from my teammates. I learned about new software, programming languages, and about the different roles that are involved in a software engineering team. Some of the new software I've needed to learn include Trello, Slack, and GitBash. Trello is a project management website; it also has an app. We used this to keep track of our sprints. Slack is a communications app; we didn't use this a whole lot. GitBash is an application that helps connect my computer to our GitHub. For all three of these, I learned about them and how to use them through my teammates who have used them in their jobs. They ran through some of the commands and did a walkthrough of how each website or software worked. If I needed to figure out something on my own, I just did a google search. This turned out to be confusing at times, so it was easier to ask my team. My teammates also showed me Amazon Web Services (AWS) and Postman which are used for mock servers, I think.

I also learned some C# and about XAML. I watched the video "C# WPF UI Tutorials: 01 – The Basics" (https://youtu.be/Vjldip84CXQ) to learn about how C# is used to add functionality to a GUI. This was similar to what my team's programmers had to do with the app. This video "Xamarin Forms Tutorial: Build Native Mobile Apps with C# | Mosh" (https://youtu.be/93ZU6j59wL4) which used the same technologies that my team used for the app. I did try to code the fountain class; it just had getters and setters. My code didn't end up getting used. To learn more about these languages, I asked Dan who was working on the UI code to explain his process and how the code works. I used Visual Studio to actually look at our code. Some of my teammates also taught me how to emulate an android device. They also taught me about how creating a database works as well as how it connects to and populates our app.

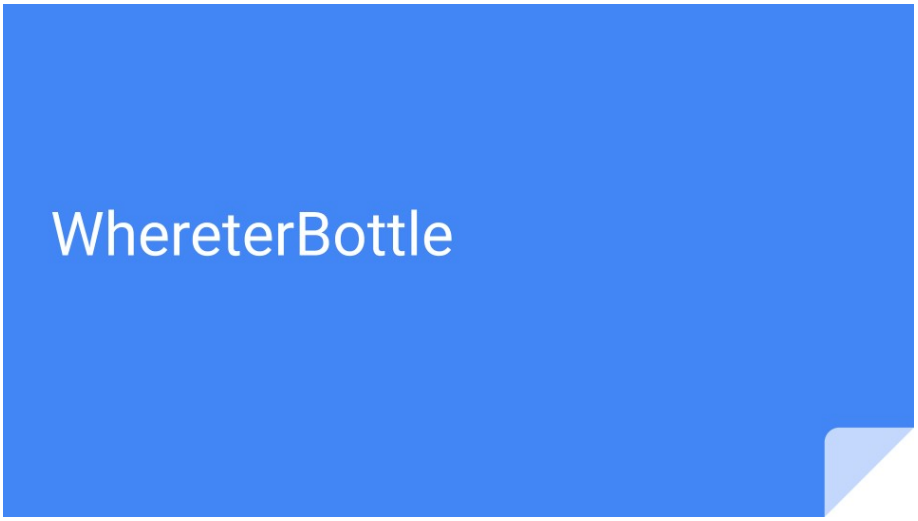
With the roles that are involved with software engineering, we learned about some of them in lecture. My team also supplemented my understanding. When there wasn't a lot of for me to do, I observed my teammates and how they communicate. I also took up some roles myself. For one class, I acted as project manager for our SCRUM and documented our progress and ensured that everyone was on track. That being said, I was kind of a "floater" on the team. Initially, when we were separating roles, I was assigned to code, but I haven't contributed much to our current code. I've assisted in design, project management, and documentation. Our design team drew the UML, but I contributed to the discussion of the apps design. Jorge and I set up our SRS document which lists our Use Cases. Together, we also created the fountain-related Use Cases. I've also been keeping track of what gets done during each sprint which assisted our project managers. So, I would say my main role was in documentation. At the end of this, I've attached documents that I've worked on.

My current project is to get our presentation ready to go. I also plan on making sure that all of our documents are in order for our final submission. There are many use cases that still need to be written up and added to our SRS document. During this process, I have learned that it would be useful to update our documentation at the end of each sprint rather than track what we accomplished and what we need to do next. I also want to work on developing test cases.

Project Requirements that I worked on

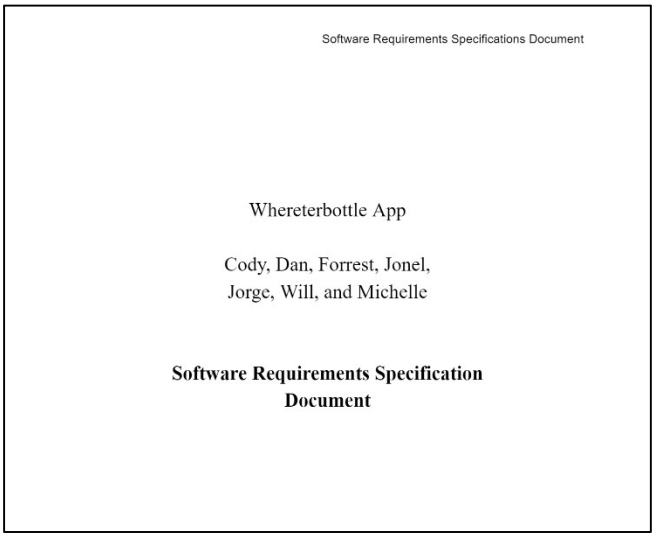
Whereterbottle Presentation

I've only displayed the first slide.

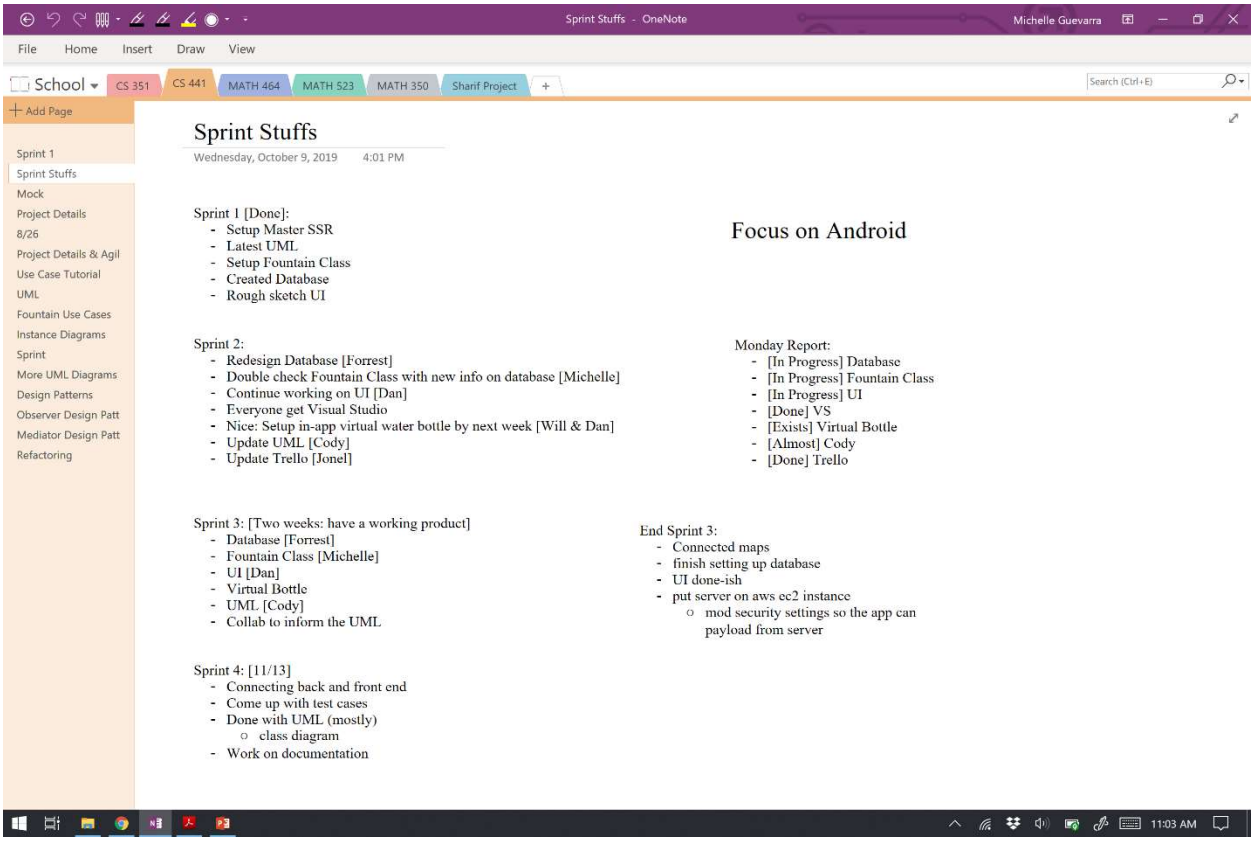


SRS Document

I've only displayed a screenshot of the first page rather than attaching everything.



SCRUM Notes



Sprint Stuffs - OneNote

Michelle Guevarra

File Home Insert Draw View

School | CS 351 | CS 441 | MATH 464 | MATH 523 | MATH 350 | Sharif Project

Add Page

Sprint 1
Sprint Stuffs
Mock
Project Details
8/26
Project Details & Agil
Use Case Tutorial
UML
Fountain Use Cases
Instance Diagrams
Sprint
More UML Diagrams
Design Patterns
Observer Design Patt
Mediator Design Patt
Refactoring

Sprint Stuffs

Wednesday, October 9, 2019 4:01 PM

Sprint 1 [Done]:
- Setup Master SSR
- Latest UML
- Setup Fountain Class
- Created Database
- Rough sketch UI

Focus on Android

Sprint 2:
- Redesign Database [Forrest]
- Double check Fountain Class with new info on database [Michelle]
- Continue working on UI [Dan]
- Everyone get Visual Studio
- Nice: Setup in-app virtual water bottle by next week [Will & Dan]
- Update UML [Cody]
- Update Trello [Jonel]

Monday Report:
- [In Progress] Database
- [In Progress] Fountain Class
- [In Progress] UI
- [Done] VS
- [Exists] Virtual Bottle
- [Almost] Cody
- [Done] Trello

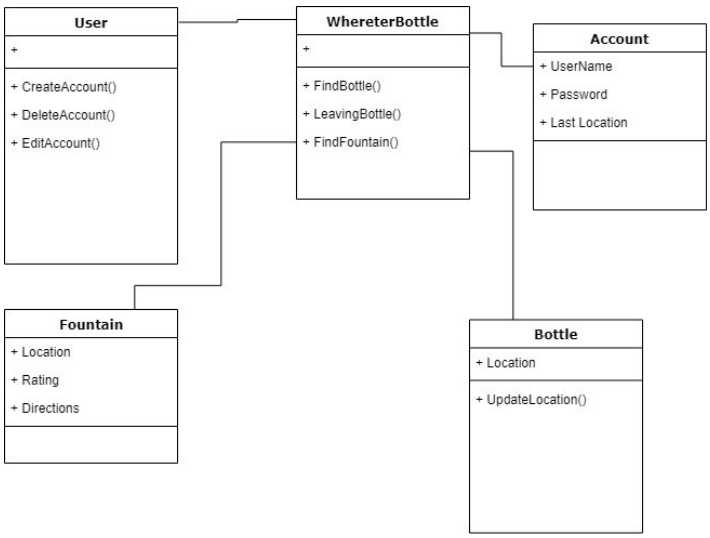
Sprint 3: [Two weeks: have a working product]
- Database [Forrest]
- Fountain Class [Michelle]
- UI [Dan]
- Virtual Bottle
- UML [Cody]
- Collab to inform the UML

End Sprint 3:
- Connected maps
- finish setting up database
- UI done-ish
- put server on aws ec2 instance
  - mod security settings so the app can payload from server

Sprint 4: [11/13]
- Connecting back and front end
- Come up with test cases
- Done with UML (mostly)
  - class diagram
- Work on documentation

11:03 AM

Initial UML Diagram



Use Case

Name: Add Fountain

Description: A use case that allows an existing user to add a new fountain to the whereterbottle app map.

Goal: Add a new fountain.

Preconditions:
1. User is at a fountain that doesn't yet exist in the whereterbottle database.

Basic Course:
1. User wants to add a fountain
2. System prompts user to add the fountain
3. User takes a picture of the fountain
4. System gets location data (coordinates)
5. System adds it to the whereterbottle database

Alternate course A:
Condition:

Post Conditions:
1. Success - The fountain is added to the whereterbottle database and can now be seen on the in-app map.

Actors:
- User
- System

Notes:

Use Case (cont.)

Name: Review Fountain

Description: A use case that allows an existing user to rate or review an existing fountain.

Goal: Rate or review a particular fountain.

Preconditions:
1. User is in-app and visits an existing fountain.

Basic Course:
1. User is near/visits an existing fountain.
2. System prompts user to review the fountain.
3. User writes a review or rates the fountain.
4. System saves the review with the fountain data.

Alternate Course A: User doesn't want to review the fountain.
Condition: User aborts review creation.
1. At any point, the user can choose to cancel the review creation.
2. The process is terminated.

Alternate Course B: Fountain doesn't exist.
Condition: User visits a fountain not in the database and wants to review it.
1. See "Use Case: Add Fountain."

Post Conditions:
1. Success - The new review is added to the fountain information.
2. The fountain information remains the same.

Actors:
- User
- System

Notes:

Use Case (cont.)

Name: Locate Fountains

Description: A use case that allows an existing user to see nearby fountains.

Goal: Display nearby fountains on a map.

Preconditions:
1. Launch in-app map

Basic Course:
1. User wants to see nearby fountains.
2. Systems accesses fountain database. (coordinates)
3. System shows a map of recorded fountains. (Coordinates to API)

Alternate Course A: App doesn't display map.
Condition: Map doesn't load.
1. System checks for wifi or data connection.
2. System prompts user to restart app or notifies the user that there's no connection.

Alternate Course B: Fountain doesn't exist.
Condition: User is near a fountain that isn't on the map.
1. User chooses to add the fountain.
2. See "Use Case: Add Fountain."

Post Conditions:
1. The app displays the map.

Actors:
- User
- System

Notes: