Part 1: Self-Evaluation

- Alex Diviney
- Chief Architect, Microservices Design, Microservices implementation
- Assign yourself the grade: A+
- What percentage of the work did you complete?

  30%

- Did you complete your tasks? If you did not complete your assigned tasks explain why.

  There were only a few major tasks for me to complete this sprint. I had to deal with Dockerization, a couple of login features, some documentation, and then a variety of bug fixes and manual (yes, manual) testing. I completed all of them in a satisfactory manner.

- How would you rate your participation in this group? (choose one of the following)
  - I was incredibly involved.

    I was able to take the foot off the gas a little bit this sprint, and mostly dealt with bugs that manifested themselves (mainly due to not unit testing my code as I wrote it) towards the end of the project. I attended every group meeting, and was in constant communication with all of my team members, and did all necessary work to polish the final product and make it release (if not production) ready.

Part 2 - Team Member Evaluation

For each other team member complete the following:

- State their name: Show Pratt:
- Assign this person a grade: A+
- What percentage of the work did this person complete? (Total team percentage must add up to 100)

  30%

- How would you rate their participation in the group? (select one of the following)
  - This person was incredibly involved

  Show is the team member that other teams dream about having, I couldn't build a better teammate if I was given a laboratory and a billion dollars of R&D funding.

For each other team member complete the following:

- State their name: Michael Collier:

- Assign this person a grade: A+
- What percentage of the work did this person complete? (Total team percentage must add up to 100)

  30%

- How would you rate their participation in the group? (select one of the following)
  - This person was incredibly involved

    Michael did an excellent job. Bear in mind that Michael programmed both in JavaScript and Java for this project, and had to deal with a variety of eccentric libraries, and yet was still able to pull off a nearly professional grade interface with working backend connections.

For each other team member complete the following:

- State their name: David Hellwig:
- Assign this person a grade: A+
- What percentage of the work did this person complete? (Total team percentage must add up to 100)

  9%

- How would you rate their participation in the group? (select one of the following)
  - This person was incredibly involved

    Michael did good work. He was the tester, and I do wish that he followed my work closely to keep me in check, but really I should have just done my own unit tests and not relied on him having to parse esoteric network pipeline code. He was pretty busy this sprint but did a fantastic job.

For each other team member complete the following:

- State their name: Ekow Barlow:
- Assign this person a grade: C-
- What percentage of the work did this person complete? (Total team percentage must add up to 100)

  1%

- How would you rate their participation in the group? (select one of the following)
- This person did not contribute

  I do not like writing failing scores, so I won't fail Ekow again. Not because he was a useful team member, but because he showed up to all of our discord team meetings (didn't talk though). We could have used an additional tester that was very proactive,

but he wasn't proactive and it is not my job to poke and prod a team member to earn participation points when I can ensure a good product without them. That is all I have to say on this matter.

Part 3 - Team Use of Scrum and Essence

- Briefly describe how your team has been utilizing Scrum and the Essence concepts

  Agile development has in my opinion been a wonderful tool. One of the biggest pitfalls that CS kids fall into in my opinion is procrastination. Typically Computer Science students were gifted students in high school, meaning that for the most part they can afford to procrastinate (read: Submission time of this evaluation). However, coding requires an inordinate amount of time. Not only does a programmer have to deal with the typical mentally intensive tasks like design, implementation, and especially bug fixing, but they also have to constantly think about abstraction and how to generalize.

  Agile development basically takes procrastination off the table by setting concrete expectations for biweekly deliverables, further enhanced by constant concise meetings that pressure members to work through the structure and of a daily scrum. Agile development flattens the slope of development intensity over time (see our GitHub insights page) which is a godsend.

- Describe any issues the team is having following these approaches

  I have found no main issues, in a worse team, communication issues could have become very bad, but crossed wires in our team were always fixed quickly.

- Describe any methods your team has adopted to improve

  Our team often did weekend meetings. We also set up a github push notification bot in our Discord server, which was immensely helpful. I thought diagramming our system use cases and architecture (informally) was useful, but I personally am a very visual thinker so not everyone may have found that as useful as me.

Also: Just in case you don't have our GitHub (We submitted it during one of the earlier sprints) here it is. https://github.com/Team-Over-Zero/UML-Diagrammer