6.170 Reflection

What went well

- We roughly met all of our deadlines and minimally deviated from our originally planned timeline and features for each part of the project.
- We worked together efficiently and effectively.
- We met in person a few times to work together and ask each other questions.
- Online communication was also prompt and effective.
- Front end: The application looks really nice on the front end and was generally well received to people we showed it to.
- Backend:

What could have been improved

- As always, we could have generally procrastinated less. As college students tend to do, we left tasks until the last minute and sometimes did not have enough time before the deadline to work the last few bugs out.
- Frontend: There are still some usability kinks that can be sorted out even though the overall feel of the application is good. There are some user flows that could be improved and features that can be added to help the user. One example is the flow to post an announcement -- in order to keep the application simple, we made it so that admins and teams leads posted to that specific group, meaning they would have to navigate to that group or team page in order to post an announcement. However, this flow could be made more intuitive.
- Backend: While we implemented middlewares, I feel that there is still room to further refactor the code using more middlewares so as to prevent repetitive code. We pushed most refactoring to the end after we had a working solution but maybe a better option would have been to refactor as we moved along.

What we could do different in the future

- In the future, we could have set deadlines further back in time, therefore avoiding the grind right before the deadline to finish the tasks. By setting deadlines for us to finish the specs a few days before the deadline, we can also use those few days to polish the app and test thoroughly and fix bugs before submitting the implementation.

Evaluation of the project from team planning perspective

- From a team planning perspective, we did a good job organizing the tasks in the project plan to split up. When there were tasks that could not be as easily split, we worked together to implement or solve them. When one member had questions about something another team member did, the question would often be answered promptly. Overall, we worked together well as a team.

Key lessons learned

- Group

  - It would be better to do more tasks earlier so that we have more time to work out bugs last minute instead of scrambling right before the deadline to finish things.
  - It is inevitable that some people will complete their tasks before others, and in those cases it is useful to have the person who finished first document everything they did thoroughly so that whoever works after will be able to easily pick up from where they left off instead of having to comb through a lot of code trying to figure out inputs and outputs. In general for projects, it is important to have good documentation.
- Yolanda
  - Since I mostly worked on front end, it was important to communicate with my team a lot about what routes I needed to obtain and submit data, and what format that data needed to be in. For the first part, I clarified a lot of these values over chat, and by the final section, I documented all the todo items as well as what information I would need or would return directly in the code so that my teammates would know without having to ask me directly.
  - Another UI lesson I learned is the importance of thoroughly planning out the user flows before starting to implement the application. By carefully planning out the user flows, it makes creating the application and figuring out which parts need to be modularized much easier.
- Victor
  - I worked mostly on the backend, which included routes and model implementation, as well as the stable matching algorithm. The biggest take away is that communication is key when working in a team environment. We had some trouble communicating and assigning roles, and clearing this up would have sped up and made the developing process easier.
  - Some other challenges I faced included connecting the back and front end together. Since the front/back end was worked on usually by separate people, it could be hard at times knowing how something worked and how it should fit in. Thus this ties back to good communication.
- Lesian
  - I worked on the data models, schema design and implementing routes. I would say one of the biggest lessons I have learned is that good design saves a lot of work when it comes to actual implementation. I think I could have designed the schema better and that would have made the queries written by my teammate more efficient and straightforward. From this, I have learned to spend more time designing the model and schema because they can have adverse effects on implementation.
  - On routes implementation, I feel I and Yolanda who was working on front end could have communicated well in advance on what data she needs and in which format. Agreeing this beforehand would have eliminated most of the challenges we faced when hooking up the front end to the back end. Overall good communication could have saved us some time and resources.
- Faaya

- I worked mostly on the models in the backend, and tried to get as much of the work done ahead of time. However, my inexperience with SQL did slow down the team at the last minute after the routes were set up, since some of the queries I had written were not returning values I expected. It was a good experience learning to debug what went wrong with my teammates, and I was ultimately able to figure out what the correct ways of fetching what I wanted from the database were. I have learned that it is a good idea to make unit tests regardless of the extra effort it might take, since it will save the several hours spent debugging a large codebase where it is hard to identify where the errors are coming from.