Team Reflection - WeTube

Jimmy Zhang, Tiange Zhan, Miri Choi, Dhroova Aiylam

November 24, 2015

Peer Review

- **Jimmy's peer review of**
  - **Tiange:** Tiange was very hard-working and wrote a lot of the code in our app. When people were unable to figure things out, she wrote the corresponding code. It would be better to let others work on more, so they can learn as well. She also didn't know how to use Promises before, but she does now.
  - **Miri:** Miri isn't too familiar yet with some of the technical concepts, but she's been working hard during the semester trying to learn more about the technicals. Miri also seemed very busy during the semester, so she wasn't able to make some of the meetings. Better communication and scheduling can probably resolve this issue.
  - **Dhroova:** Dhroova isn't primarily course 6, so he wasn't too familiar with some of the technical concepts. Dhroova did, however, take initiative into learning many of the concepts and starting early. He didn't know EJS, but took the initiative to learn it for our project. Coding together as a team more would help all of us learn better.
- **Tiange's peer review of**
  - **Jimmy:** Jimmy was good with knowing all the deadlines and keeping us on track. He would often be the one organizing us to meet, and setting internal deadlines for the team. It would be great if he had set earlier deadlines for us, but obviously that's all of ours responsibilities, too. Jimmy is also very technically strong and wrote very clean, good code.
  - **Miri:** Miri is great with helping team dynamics; she's very good at making sure we're all not too stressed and maintaining a positive outlook on the project when we might be feeling very worried about things. She might not be too comfortable yet with some of the technical aspects, but she is super hard working and great with design. We would love to see her at more of the meetings, as she was busy during a couple of them.

    - **Dhroova:** Dhroova worked really hard and began coding earlier than any of us. He's very easy-going and helps the team get along well. He also might not be too comfortable with certain portions of the technicals yet, but he was very good about making it a priority to learn and willing to put in a lot of effort for the team.
- **Miri's peer review of**
    - **Jimmy:** Jimmy did a good job of setting the deadline for all the little works and acted as the leader of our team. He communicated effectively with the team and wrote a quality of code. He seemed to be getting too stressed, but it is also our responsibility to share the stress together and try to resolve the issue.
    - **Tiange:** Alice is a very hard worker, and she put lots of time into figuring out Spotify API for us. She is very enthusiastic and helped us debugging a lot, as she is a very skillful web developer. She supported our team mentally and technically. However, she could also have the team update with the current progress more often.
    - **Dhroova:** Dhroova is a hard worker. As a student who is not course 6, he puts a lot of time into figuring out the code and to learn. He is a great team member to be around and a good communicator with the team. However, he could come to the meeting on time.
- **Dhroova's peer review of**
    - **Jimmy:** Jimmy did a good job organizing the group and keeping us on track. Aside from perhaps the MVP, he set effective internal deadlines and we managed to meet most of them. He wrote quality code as well, and communicated effectively to the team what his code did and what it expected of other code.
    - **Tiange:** Tiange wrote, directly or indirectly, a lot of the code and was instrumental in helping us get a finished MVP done on time. Since she is a strong coder, she was able to finish her own portions of the code and have time to help others with theirs as well. Perhaps she could have kept us better updated as to what she was doing, though.
    - **Miri:** Miri did a good job of the UI design and was easy to work with. She helped a lot with the UI implementation as well. In the future, maybe she could try to be available for more of the team

meetings and try to keep her new code compatible with already-existing, dependent code.

Evaluation

- What went well
    - We all put in a lot of effort for the project
    - We all learned a lot and feel more comfortable with these web technologies and coding paradigms.
    - We had fun! :)
    - Seeing the product come together was cool! :)
- What could have been improved
    - Ran out of time and had a hard time merging
    - Last-minute fixing of bugs
    - We weren’t able to implement as much as we had wanted to in the MVP, as many of the basic tasks (such as logging in) took us much longer than we expected
    - We met on Monday to put our individual parts together. This was more challenging than expected. We really should have finished our individual parts by Saturday, and used both Sunday and Monday to connect all the pieces.
    - We tried to use promises instead of callbacks, which none of us were very familiar with. The learning curve associated with using and debugging that also set us back with our schedule.
- What we would do differently in the future
    - For the final implementation, maybe we could split tasks based on feature and have one person do everything (view, frontend js, backend js, database) related to it. That way we minimize issues where one person’s code is dependent on another’s, and variable names, etc, end up not matching.
    - See Lessons Learned section for more details

Lessons Learned

- Commit regularly - People worked on self-contained sections of the code (ejs templates, server routes, etc.), but sections are dependent on other sections. When a section is written, assuming a certain behavior from the other sections, and another section is written, assuming a different behavior from the other code, the first section may need to be rewritten altogether. Regular commits would help everyone have an idea of at least roughly what the dependent code looks like.

- Work on it early - The design document took us a while to hash out, and we weren’t able to start working until that was finalized. We all started working a bit later than we should have, giving us very little time to merge in the end. Joining together each section and making them work together is a lot more time consuming than we anticipated, even though we anticipated that it would be very time consuming.

- More communication - Maybe work more with everyone in the same place, because it got difficult to know what others were working on all the time. This also helps keep the expected behavior of a given piece of code consistent, and can help people get un-stuck. Better communication can also facilitate the different sections of code interfacing correctly.

- Test individual parts before trying to merge. At the time of merge, we had a lot of bugs at various levels of the code, and it was challenging trying to debug everything together, as we had to pinpoint at which level(s) the error was occurring.