

Team Reflection - WeTube

Jimmy Zhang, Tiange Zhan, Miri Choi, Dhroova Aiylam

December 7, 2015

Peer Review

- **Jimmy's peer review of**

- **Tiange:** Tiange was very hard-working and wrote a lot of the code in our app. She added a ton of spotify features, and worked until 4am fixing bugs. It would have been nice if she had given more tasks to the others to do, so it's less a burden on herself and others learn more.
- **Miri:** Miri isn't too familiar yet with some of the technical concepts, but she learned more throughout the semester and made the UI look a lot nicer. Miri also seemed very busy during the semester, so she wasn't able to come to or stay for the whole time for some meetings.
- **Dhroova:** Dhroova isn't primarily course 6, so he initially wasn't too familiar with some of the concepts. He did take initiative and learned a lot during the semester working together with us. In the end, he helped out with a lot of the features in UI, javascript, and backend.

- **Tiange's peer review of**

- **Jimmy:** Jimmy continued to be a sort of leader for the group. He continued to be the one scheduling most of the meetings and encouraging us to start working early. In addition, Jimmy is very strong technically. For example, he taught me how to work with promises, and he was the main person in implementing the profile image upload, which proved rather difficult. He could have been less involved with pushing us, because we started to depend on him, making us less likely to take initiative ourselves.
- **Miri:** Miri continued be very helpful with team dynamics. She is very talented with UI design, and she helped a lot in that regard. Although she is not too strong with some technical aspects, she's a very hard worker. She did seem busy for a lot of the time, which made it a bit difficult for some meetings. She continued to

be very positive about everything, which was helpful for team spirit.

- **Dhroova:** Dhroova started out saying he wasn't too comfortable some of the technicals, but he was able to pick up skills very quickly. He continued to work very hard and contributed a lot to the code. In addition, he also helped with team dynamics, encouraging us and helping everyone get along. He was always willing to take initiative with the tasks at hand.

- **Miri's peer review of**

- **Jimmy:** Jimmy stayed to be a great leader until the end of the semester and kept a very good team dynamics overall. He managed to keep an effective deadline and sent us an reminder email for each of our meeting. He wrote lots of quality codes as well.
- **Tiange:** Alice is a talented coder, and she helped a lot with implementing with Spotify API into our web application. She kept a positive attitude throughout the semester and helped the member, including me, with questions on the code when we needed help with concepts and debugging.
- **Dhroova:** Dhroova did a great job on managing javascript codes, and had himself more available to the team meetings. He also kept a positive attitude and put lots of efforts into the team project. It was great to work with him as a group.

- **Dhroova's peer review of**

- **Jimmy:** Jimmy did a great job organizing the team and keeping us on track. Although it was difficult to coordinate people's schedules, he managed to set effective internal deadlines and we were able 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 a lot of the code and was instrumental in helping us get a finished product done on time. Since she is quite a strong coder, she was able not only to finish her assigned features, but also have time to help others with theirs. She could have done a better job of communicating what she was doing with the rest of the group, however.

- **Miri:** Miri did a nice job of designing the UI and was easy to work with. She helped with the UI implementation as well. In the future, maybe she could be available and on time for more of the team meetings and try to keep the code that she writes compatible with already-existing, dependent code.

Evaluation

- What went well
 - Final product was great!
 - We managed time / deadlines better than for the MVP
 - We improved communication and teamwork after MVP reflection
 - We became better web developers
 - We added a lot of cool features, like uploading profile images and importing playlists, and integrating with existing api's and libraries to achieve these
 - We had fun making the site and adding the features.
 - We were better about dividing tasks and integrating work
- What could have been improved
 - Could have tried to finish the final commits/deploy with a little more time to spare
 - Did not ultimately implement some of the features we planned to (e.g. boos)
 - Could have started writing tests earlier. (Especially since we were unfamiliar with the testing framework with promises).
 - Could have practiced the presentation more.
- What we would do differently in the future
 - Try to make sure everyone can be present for the entirety of the team meetings; this is hard, given that people have different schedules, but would help everyone stay on the same page
 - See Lessons Learned section for more details

Lessons Learned

- UI can make a big difference in how the project comes off. Even though we had cool features, the website didn't seem very legitimate until we worked through a lot of UI fixes and made it more professional.

- Don't be afraid to try code that might break the website. You can always go back with git, and we should be free to play around as we want with fun new things.
- Don't reinvent the wheel - there are a lot of node libraries that do useful things for you (e.g. hashing, automatically sending emails) and it's easier and more reliable to use these instead of trying to write them yourself.
- Commit regularly - People generally worked on self-contained sections of the code (one person for a certain feature or fix, etc.), but sometimes sections are dependent on other sections. Regular commits would help everyone have an idea of at least roughly what others are working on, so that their own code can depend on it properly, and can prevent issues where one feature or fix breaks another.
- Work on it early - Working on it earlier would have allowed us more time to discuss the various features and their different components. It would also have given us more time to merge our code and make sure that the UI, javascript, server, and database worked together properly. If we had finished the features earlier, it would also have given us more time to do more comprehensive error handling and testing.
- 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 ensures that we are all on the same page about expected behavior of various components, and we can help each other out if needed. Better communication can also facilitate the different sections of code interfacing correctly.
- Test the website very thoroughly. There are all sorts of crazy things users can do that we didn't catch until later on. For example, starting a gathering with no songs, or joining a gathering when you're already in one, or clicking text boxes with no input. It's important to consider all these cases and make sure they are handled properly.