

# Final Reflection

## Evaluation

We completed pretty much what we set out to do. Our app runs well, without errors, and we implemented almost all of the features we set out to implement (and some that we didn't!). There were some additional things that we initially wanted to have in our final, but we opted for focusing on other aspects of the app so these got pushed down the priorities list. We also adjusted our focus for the different features and who would be working on what, learning from the MVP.

We also started to assign more vertical features as things for a person to work on as opposed to what we did on the MVP where we had different people working on horizontal parts of the stack. Since the app was more or less fleshed out, this was a lot easier to do since there was already a lot of code that was in place that we could always work off of.

## Lessons Learned

### What went well

The clearly specced design helped us a lot in the final, just as it had for the MVP. We added to it a spreadsheet checklist of tasks to complete and bugs to fix. The combination of these two things enabled us to pick up extra tasks easily and in general to keep track of the state of the project, to make sure everything was being completed on schedule.

Using the spreadsheet checklist also allowed us to assign tasks to different members of the group, and ensured that we weren't wasting time by working on overlapping tasks. The checklist also helped us keep track of and address any bugs that came up.

We also made efforts to communicate better as a group, and this showed in that we weren't wasting time stuck or doing useless work. As an example, Sam and Amanda decided to switch parts of their tasks every so often, a task which was made much easier because we had the spreadsheet and were communicating better.

Another thing we started to do that went really well was having features be built in separate branches and then opening pull requests for the different branches so that we can have a peer code review process. This prevented a lot of potentially broken code from being checked into master and subsequently breaking our site.

## What went poorly

One thing that is constant about working in groups is that communication could always be improved. We did pretty well in this regard, but there is always room for improvement!

Something we could have done better with was making sure that everyone was familiar with all of the code in the codebase. Often times it was apparent we weren't familiar with everyone else's code when we needed to extend functionality that someone had previously partially implemented. This often resulted in rewriting the same bits of code in multiple places and in some cases rewriting the same routes. This of course lead to some very interesting behavior until we found the additional route. The way we could have solved this is having better documentation in the README.md and more commented code. We also could have communicated better on what code we were working on - that way, if someone had already implemented that feature/route, duplicate routes could have been avoided.

## Peer Review

### Amanda Liu (ajliu)

Looking back, I think it makes sense that Amanda had done so much work for the MVP, because she seemed a lot busier during the implementation of the Final. Despite this, she was responsive to messages, and able to answer questions quickly and efficiently, and the work she did was nicely done. I'm also grateful that she had pretty clean UI code that I could use as an example while doing my UI tasks. ~erosolar

Amanda did a great job at getting a lot of different features done. Once her focus shifted from more of the UI and frontend specific code, she started to add additional functionality that really rounded out the app. I think that she can still work on estimating the amount of work that she's has but she has done a lot better in this regard. - erosales

Amanda laid out much of the UI for the MVP and her code was well-organized, so it was easy to proofread and add new features on top of the MVP. For the final version, she added several additional features that fleshed out the app, such as the map for event location input. Her communication was quick, so we were always updated on her status and could help her without delay. -kimsin98

### Everardo Rosales (erosales)

Ever mostly continued in his role as pseudo-"project manager" for the second half of the project. It was very helpful - I personally find it hard to lay out the tasks needed to get something done,

and because Ever was good at figuring out what specific things needed to get done, it was a lot easier for me to get work done. In addition, Ever was really helpful in debugging code and going over code reviews to proofread our work. He also implemented some tasks as well - he wasn't just behind the scenes. I think that being able to do all of this was really impressive. ~erosolar

Ever continued to serve as the overall project director and did an even better job at it. He was the one who suggested we enforce the branches/pull requests rule and reviewed most of the pull requests to ensure that our master branch remains clean and bug-free (making edits as needed). He also added a very nice-looking UI framework that not only made the app look better but also added various features like error flash messages and input fields that warns the users when they receive improper values. -kimsin98

Ever had an important role as a kind of team manager. I appreciated that he put a lot of work into the project, especially during the final stage. He also initiated the system of putting each feature on a branch, which ended up being very helpful for keeping things organized. He needs to improve on staying positive for the group during times of high stress. Because of his managerial role, it was kind of understandable that he had individual conversations with team members. However, sometimes this made me feel left out when feature decisions were made off the group chat without my opinions being fully considered. - ajliu

## Samantha Briasco-Stewart (erosolar)

I appreciate that Sam did get all of the work that she committed to doing as early as possible to prevent anyone from being blocked. I appreciate how she was able to contribute more to the full stack of the app for the final. I think that she can still work on being able to break down higher level tasks so that they can be easily built, but this in no way hurt her ability to contribute to this project. - erosales

For the final version, Sam worked on all layers of the app to ensure that the app's core features are functional. As with the MVP, her work was prompt, so whenever anyone needed new database methods, that person did not have to wait for long. She also made sure all her code ran as expected by running thorough tests, so whenever something went wrong, I knew I messed something up, rather than having any doubts about her code. -kimsin98

I appreciated that Sam and I coordinated our strengths pretty well by switching off between tasks when the other person was more familiar with the existing code. I also like how she devoted time to writing thorough tests. There were a few times when I felt that Sam could have communicated with me more - Ever would reach out to me telling me something Sam said/needed, and it seemed inefficient for Ever to be the middleman of communications. - ajliu

## Sin Kim (kimsin98)

I'm sure Sin meant well. He had good ideas, usually, although he was I think the worst at communicating of the people in our group, because he wasn't used to using Hangouts, the communication the rest of the team all preferred. His code was written well, but he took so long to check in sometimes that we would just give up and write code for him, because we couldn't be sure he would submit it on time. It was because of this that Sin didn't commit as much code to our project as everyone else, I think. ~erosolar

I appreciate that Sin was able to contribute in a meaningful way to the project with his validation checking code and his email sending code. Most of my difficulties of working with Sin stemmed from having a hard time reaching him and not really knowing what his progress was on the tasks he was working on. I think that if he was more clear with what he was working on and made this transparent by regularly checking in code to a different branch, even if it is incomplete would have cleared a lot of this up. -erosales

Sin did well by contributing much more code during this stage of the project than before. I appreciate how he took into account the rest of team's evaluations of it being hard to reach him during the MVP stage, and tried to be more responsive during this stage. I wish he had communicated more with the group on what he was working on. This was especially difficult when I was waiting for him to finish some backend for a feature that I was working on the UI on, and didn't know if he was working on it or how progress was. - ajliu