

MVP Reflection

Evaluation

At the moment it seems that we are on track to finish the features planned for the Final version of our project by the deadline. We laid out in detail all of the different tasks that we wanted to get done for the MVP in our Teamwork Plan and we completed all of the tasks as well as made some progress on other tasks that were part of the final. We implemented the basic CRUD parts of our application for the MVP (as planned), and will be implementing more of the interesting functionality in the final such as email reminders and notifications, budget allocation and warnings, multiple planners for a given event, etc. Considering how well the initial MVP went, we are sure that we can make great progress on the Final and finish by the deadline.

Lessons Learned

What went well

Having a clearly specced out design saved us a lot of time since whenever we didn't know what else to work on, we could simply look at the tasks that we detailed in our teamwork plan, and then would pick one that no one was working on. Putting in that extra bit of effort when creating the design doc and teamwork plan saved us a lot of wasted time in the long run, so that definitely went well.

Another thing that went well was having clearly defined boundaries between the work that different people would do. Doing this ensured that we were stepping on each other's toes as little as possible, and so therefore we were wasting as little time as possible.

What went poorly

It is exceedingly useful to have regular check-ins and status reports on the different parts of the app. We noticed that when we had meetings (check-ins or just getting together to work), the work got completed much more quickly and effectively than when we were working separately. We can definitely always communicate better, so that's something for us to work towards while working on the final.

Also, we occasionally had arguments about how to spec a particular feature or method in the app. When these arguments happened, we would usually discuss as a group until we came to a consensus as to what should be done, and in those cases we worked very well together. However, there were one or two times where we didn't have a discussion (effective or

otherwise), and it definitely showed. This is basically another reason for us to communicate better.

Peer Review

Amanda Liu (ajliu)

I appreciated the fact that Amanda took the time to create pull requests and work in a different branch so that we can review her code before it goes directly into master. I think that she still needs to work on getting correct estimates on the amount of time a particular task will take and reaching out in advance if she anticipates needing help. - erosales

I was impressed by how much Amanda contributed to the project - I think I hadn't realized how much work went into UI and frontend design. I think one thing Amanda could work on is delegating, maybe? I felt bad that she had to do so much work, and I think that if she had given me tasks I probably could have done them, lightening her workload. (I don't think I had as much work as her.) - erosolar

I feel that Amanda took the most laborious role for the MVP. UI and front end involves a lot of decisions and attention to details, meaning this role takes deceptively large amount of time. Amanda should definitely reach out to us more often for help on later UI work. I also need to keep in mind that the division of work in Teamwork Plan is flexible, so I should help out over-burdened teammates. - kimsin98

Everardo Rosales (erosales)

I'm thankful that Ever put himself down as an overall helper. At first I was worried that he wouldn't do as much work as Amanda, Sin and I (because he wasn't the lead for a particular part of the project), but he turned out to be very useful as sort of a project manager, keeping us on track and helping out wherever needed (for example, I know he helped Sin a bit with the user authentication). I think one thing Ever could work on is identifying possibilities to split up work differently. This is more in regards to his 'role' as project manager, where it would be useful to have a person to identify when one person could do more work and one person could do less and then to re-arrange work between them. - erosolar

I like how Ever became the effective manager of the group. Even though he wasn't assigned to be in charge of a specific part of the project (he was "aux" for everything), he played an essential role of making sure we all coordinated well as a team, reminding us when work was unfinished, and delegating tasks. I also appreciate how he created a Readme doc covering the routes he worked on, because this was a helpful reference to me when I was working on the front-end. In general, Ever has been good at communicating with the group, but I think he can

still work on communicating code-specific issues, as I think it could save him valuable debugging time. - ajliu

Ever meant to be the “coordinator” between different roles, and he has done that and much more. He seems to be most knowledgeable about various web programming topics, so he helped solve a lot of issues we encountered. As someone who works on all aspects of the project, he also made decisions on things that need to be agreed upon by different layers (e.g. planning out all the routes). I agree with Sam that he should delegate work more evenly. He actually wrote up basic code for all of the routes after mapping out the routes. That left me with little work for MVP outside proofreading and revising some of his code. - kimsin98

Samantha Briasco-Steward (erosolar)

Since Sam was working on the Databases and Models, I appreciate that she finished this as quickly as possible since the rest of the project relied on it. I think that she can still work on being working on more parts of the codebase that are not just related to the databases that she was charged with. I feel that for a project like this, there is a lot to be learned by working on the different parts of the project and as such, she can still learn a lot about different aspects of web development that would be very useful at large. - erosales

I'm super impressed with the quality of Sam's code. It's well-organized and well-tested, which was really helpful to me when I started working on front-end and needed to debug things. Like erosales, I appreciate how she finished the bulk of the database functionality as soon as possible. - ajliu

We used Sam's Fritter project as our base skeletal code, and I am really glad we did because the way she organizes her code is very clean and conducive to collaborative coding. She also finished the database/model code very quickly, so that gave other layers a nice foundation to stand on, especially for testing. - kimsin98

Sin Kim (kimsin98)

I think that Sin needs to work on being more open about the work that he is doing and the progress he has made. I feel that at times he was at a loss and either didn't know what to do or how to do something. I was unable to tell which of the two this was and it made it very difficult to help him. Moreover, the lack of communication resulted in not having a clear understanding of what was expected or needed to be worked on. - erosales

I was glad that Sin contributed meaningful code to the app - Amanda and Ever and I had planned to be in a group ahead of time, and Sin was only put into our group last minute. Given that, I'm impressed that he was able to become part of our group - I know I probably wouldn't have been able to do the same, were I in his place. One thing that I think Sin could work on is

being more in contact with the rest of us, so that redundant work isn't being done, and so that we all know where everyone else is with regards to the project as a whole. - erosolar

I think Sin has contributed a lot of helpful feedback on the design of the app. I'd appreciate it if he were more responsive to group messages, as sometimes it's difficult to get a hold of him when tasks need to be done or when we want to organize a group meeting. - ajliu