

Evaluation

What could have gone better?

As our project comes to a close, we realize there are great deal of things we could have done to be more efficient, and to have produced a better final project.

To start, from the beginning it would have been useful to flush out ideas earlier on, and make sure they were rock solid before doing any actual coding or other work for that idea. We needed to be sure that these ideas were feasible, and had a high probability of being used in our final submission.

Another thing that we found is that we all had knowledge of really useful packages or helpful libraries, but that knowledge may not have been shared by everybody in the group. This led to problems among other members when they had to touch code used by someone else who had extensive knowledge of a package or library. Given this problem, it would have been useful for us to share our knowledge with each other of any external sources we may have been using, to at least bring the other team members up to speed a bit more.

We also found that we were often not working physically together, and when we did, we were all much more productive (especially in the final hours of coding). We found that when working together, it was much easier to bounce ideas off each other and prevent and fix bugs earlier on. We attempted to make sure this happened by setting weekly team meetings, but each of our weekly schedules were so different that this was very difficult to schedule. Essentially, life events for each team member would get in the way of our team being able to meet as often as we wanted to, so there was not much that we could have done about this.

Finally, we should not have put off as much work as we did for close to the deadline of submission. We had a Trello board that we used to assign tasks to every team member with a due date for that task, but there were times where that board was not as up to date as it should have been. Therefore, we could have and should have paid much more attention to our Trello board, and assigned people more concrete tasks, and always made sure it was as up to date as possible.

What went well?

Although there was a great deal that we know we could have improved on, there were some aspects of our project that we know went well. One thing that definitely went well was

our final sprint. Our final sprint was by far the most productive period our team had for our project. Every team member knew what tasks had to be done, and were fully committed to completing the tasks they had assigned to them before the deadline. Moreover, everybody was committed to finishing the project, so if someone found bug or a task that was not assigned to them, nobody had any problems with tackling whatever was left just because they had “finished the work they were assigned”.

We also had a great deal of solid communication throughout the duration of the project. Although we were not able to meet up often, we were able to bounce ideas off each other remotely, and also ask questions if we needed to. People were very good about speaking about any code they were about to change and why they were about to change it, and we had some great discussions about ideas.

In an attempt to make sure that we were able to meet up in some way, every week before our mentoring meetings we had a quick meeting amongst ourselves to flush out the progress report and agenda. However what was more useful was that during this time, we would also often come up with our best ideas for the app, as we were able to quickly build off of each others ideas and have a productive, intelligent discussion.

Finally, one particular team meeting went very well. During this meeting, we hashed out which ideas were worth keeping and which were worth throwing away. Essentially, we decided on the ideas that would be most useful for our app, and this set us on a path to a better final product.

Lessons Learned

Theme 1: Design Saves Time

We started this project with a completely different set of ideas in mind. Originally we had a sort of simplified LinkedIn in mind, and originally started building a product aimed at that. In hindsight, this was a mistake, but at the time we had a difficult time shooting down ideas, which is extremely important when you want to remove tangents that will take away from the core functionality, and waste time. Sometimes it takes some concrete implementation to see that, but we didn't really hash out a useful, new product until we sat down in front of a whiteboard and drew out our ideas.

That whiteboard contained more erasure than actual content because we figured out how to *focus*. At the bottom there were two Purposes, and above that was a few views, but we took a final look at that whiteboard and realized that it all finally just worked. We also realized that all the work we had done up until that point was worthless.

Mistakes during the design process require a wave of the eraser. Mistakes during the implementation process involve a fury of git commands, a potential nightmare.

Theme 2: Large Projects Require Greater Foresight

I find that my definition for a large project depends on what portion of the code I can comprehend in a single moment. Thinking about Project 4 versus the other projects requires a paradigm shift because you can no longer comprehend the entire codebase in a single moment. If you cannot picture the structure in your mind, then you need to come up with a plan to traverse parts of it that are in the dark.

Our plan for preventing a convoluted project was to have a directory structure that made sense, a standard format for documenting APIs, and following standard coding procedures that make it readable.

Our issue is that, aside from directory structure, we focused on micro solutions, and were naive to the macro problems that were arising. These were issues like having a huge list of modules that not everyone understands, not documenting unusual implementations like adding 'deepPopulate', and having a login system that was defined in a different way than any normal routes. We had micro documentation down, but we learned that system for *macro* documentation could have really saved us a ton of time and confusion by allowing the rest of the team catch up.

Theme 3: There's a Module for That

We definitely found that great programmers learn how to do more with less effort. This is especially true for Node.js, where NPM has a module for just about everything. There were endless instances where I was trying to get around some issue with JavaScript callbacks or something, and using the module, `async`, would save me tons of time.

It's not only for modules where looking for someone else's solution is a good idea. It applies everywhere, so we learned that looking for general solutions to your specific problem saves a lot of time and prevents errors as well.

Peer Review

Danny Sanchez

Danny killed the UI for our team. I would have like to have seen him at a few more of our group meetings, but he was just always working while we were just in the peace and quiet of his own place. He has a lot of experience with front end design and I definitely learned quite a bit interacting with the code that he wrote. It would have been nice to have been educated on some of the stuff he was using to generate the views that he created, but it worked out really well anyways. - **Grant Gunnison**

Danny was a great member to have for our application's UI. He had a lot of knowledge and experience with front end web development, and was able to bring cool and important features into our application (like drag&drop and search drop down). However, in order to create some of these features Danny needed to use a technology that no other members of the group were familiar with (AngularJS). This became difficult as I wanted to update some of his work, but was unfamiliar with Angular.

I wish that Danny was able to work with the team more often. Because he often worked remotely, we had some communication issues which led to him doing extra work.

Lastly, I think Danny could of pushed earlier and more often, even if his work wasn't fully functional. This would of helped everyone in the group understand the state of the project and design better. - **Sabrina Drammis**

It was sweet working with Danny because you would be sitting there, and something would end up broken somehow, and then you would shoot off a Facebook message to him. After a while he'd respond with "it's fixed now", but not only was it fixed, it was usually better than before it broke. The only issue is that it usually took longer to explain over Facebook chat than it did in person, so it would have been nice to see more of him in person. - **Sam Edson**

Sabrina Drammis

She owned this project. She by far put the most time into our project. Definitely is very knowledgeable across all aspects. She was helpful in not only implementation but in suggesting and having knowledge about packages and tools we could use to make many of the processes easier for us. She also has an awesome meticulous eye for detail. Every little

thing mattered to her, which sometimes led to her burning time on aspects of project that were fairly unimportant at the time, but in the end it was better that she paid such close attention to detail and its very apparent in the final product. She was very enjoyable to work with and I had a lot of fun building the app with her. She helped me out a lot and I learned quite a bit from her. - **Grant Gunnison**

Sabrina was definitely the most dedicated to the project out of all of us. She dedicated a great deal of time to the project, and not just to the app coding wise, but to the logistics as well. She was one of the main managers of the Trello board, assigning tasks in a fair and meaningful way, and brought the greatest node package knowledge to the group, which made some coding aspects later on much easier. In addition, she also had a very particular eye for detail, being able to point out very small things about the app that could be improved. At times, this was useful as we wanted to make our app as good as it could be, but it might have been more productive to keep track of all the small things and save them for the end, so that everybody can stay focused on larger tasks. Sabrina was always willing to lend a hand whenever she could, and I really appreciated that from her end. - **Danny Sanchez**

It is an understatement saying it was nice having Sabrina on my team. She lead the implementation portion of the project as she was definitely the most knowledgeable and had the most foresight on *how* we were going to implement our ideas. This was a bonus, but had the drawback that we often had to keep up with her. It would have helped us a ton if she had taken a bit more time to explain what she had done, or what the modules she included did. I'm not saying she is responsible for this because none of us did a great job with this, but it would have benefitted us most if she had. - **Sam Edson**

Grant Gunnison

I enjoyed working with Grant. He was an important teammate to have during the design stages. In the first stage he brought in great ideas, when the rest of the team was struggling. He was especially helpful in helping us narrow down the focus of our application and come up with a solid and direct purpose.

Despite having the least experience in web development, Grant quickly learned and was able to implement the technologies we were using. He also brought passport authentication in to our application and helped me understand it.

There was a few days close to the deadline that Grant went m.i.a. However, he pulled through and contributed what he said he would, despite giving me a scare. - **Sabrina Drammis**

Grant was a great contributor when it came to generating ideas during the brainstorming stages. He was a very fast learner, he picked up new technologies that we were using very quickly. I was a bit concerned for a few days when we were not able to get in contact with him, but things ended up working themselves out and he was generally dependable throughout the rest of the duration of the project. - **Danny Sanchez**

Grant is a good guy at getting things done. It is always super important to have one of these guys on the team that will let everyone know when the discussion has digressed and the topic at hand needs to be handled. It was tough for a while to get ahold of him, and I think we agree that it would have been good if he just let us know he was going to be incapacitated by two heavy tests, but he got back on board afterwards and ended up killing it with the rest of us on our final sprint. - **Sam Edson**

Sam Edson

Sam is a fun team member to have. He is a little bit last minute, but when he does work, he works efficiently. He is knowledgeable and very talented at picking up new technologies. When there were random issues on the project, I felt confident in asking Sam to try and fix them even if it was an area of the code he had never worked on. He was also the only member of the team that tried to keep our Trello board up to date. It would of been nice if he worked more consistently and focused on writing cleaner code. - **Sabrina Drammis**

Sam is awesome, definitely a joy to work with. He was sort of hard to get together with because of schedule conflicts but whenever we got together we were very productive, had a lot of fun, and always got a lot done. He knows what he is doing and built in the core search functionality of the site. He has great ability to pick up a lot of stuff and kicked ass fixing bugs and implementing a lot of stuff. - **Grant Gunnison**

In general, Sam was a very solid team member to work with on this project. He came up with some of the best ideas for the project while we were still in the design and brainstorming phase, and had some very creative overarching app and feature ideas. When it came to actual coding and getting work done, he was dependable. (**Danny Sanchez**)