Project 3 Group Retrospective

Spotifind

Sehun Joo
Akash Gajendra
Jordan Hassmann
Nathan Wilke
Joshua Mo

Executive Summary

At our last retrospective meeting we gave our final thoughts on our work and reflected on all that we have learned throughout the sprints. We discussed the final product, and how all of the user data and UI function work well, but we thought that we needed more work on our API connections. The spotify connection works well but we had trouble deploying other API connections. We also discussed the general skills we learned through the project including gaining experience with various front-end and back-end technologies.

One of the lessons we learned from this project was that it's more important to get working connections with APIs early on in a project than getting a clean looking UI as without the API connections, the UI is mostly meaningless. In a way, the UI has to be balanced around the API connections. We also learned more about how poor most public documentation is and how this can roadblock development. One thing we wished we had changed was the date at which we set up the website hosting as hosting introduced some new errors in the website that we could have resolved if we had hosted earlier.

Overall, we were happy with our finished product, and think that we learned a lot of valuable technologies and lessons that we can apply at future internships or jobs. The experience of working in a group environment was invaluable.

Product Analysis

The map, front-end design, front-end user interaction related features, and user data storage/retrieval all work very well. They function on most edge cases and don't lack in speed or efficiency. The implemented APIs require redirects to separate pages and don't provide persistent data, which is something that could be improved upon. This could potentially be accomplished by storing the user's 3-party music streaming data in our database upon connection to the outside service, however this would have required another revamp of our user schema and design. This feature however could be implemented in a version 2.0.

Many new skills were developed through this project, including front-end and backend skills regarding React, Redux, JavaScript, how the browser works and its tools, Firebase noSQL database, and API communication. Additionally, the concept of storing data locally in Redux as hash maps proved to be very efficient and clean, and is absolutely something worth being applied to future projects.

Work Analysis

During the retrospective, our team performed an in-depth analysis of our collaborative development process and had an honest conversation about it. The largest challenges that our team faced revolved around time management and the learning curve associated with adapting to new technologies. The presence of conflicting project deadlines during sprints made time management difficult. Additionally, some members felt that having to adapt to new technologies came with its own learning curve and often led to roadblocks. If we were to redo it, we would have tried to have more assisted learning sessions among members who have specific domain knowledge to aid the learning curve. It would have also been a better idea to deploy our project earlier to have longer to debug any possible issues.

In terms of communication, our team was successful at having constant communication and feedback set up. Our decision to use discord for this proved to be successful but it did come with its own set of challenges. It was a communal feeling that scheduling meetings became an inconvenience as all of us have different schedules/deadlines that ultimately forced compromise into less than ideal circumstances. In retrospect, having predetermined SCRUM meetings and sticking by them would have been a better option as opposed to scheduling them one day ahead of the actual meeting. Additionally, the team members were active and made it a priority to notify each other of github merges which made the process much smoother.

Having said that, our team did manage to balance the workload and execute on the project vision. It was a good decision to use JIRA to track our user stories, the built-in features of the platform provided us with greater transparency surrounding the progress of certain tasks. Ideally, we should set aside a fixed time to work on the tasks together. At the end of the day, the meeting was insightful and each of us came away with useful nuggets that we will carry forward in other projects.

Going Forward

As a team, the primary lesson we learned is that selection of technologies during the planning phase can be considered one of the most crucial components of the Agile workflow. While many other areas of the development process can be altered and modified, as per the nature of the Agile development process, what technology you choose to leverage and implement is not something easily changeable once development has begun. In our case, the APIs we chose to use were troublesome once we began attempting to implement them.

Issues we encountered included very poor documentation, closed application for developer accounts, not React friendly implementation, and more. Had we done more research into them prior to beginning their implementation, we might have avoided some of the resulting roadblocks. Additionally, we also learned that we should spread out larger, potentially important tasks throughout each sprint. For example, we encountered some issues with other features once we hosted our website due to unforeseen internet protocol issues, cors issues, and https issues. Had we begun hosting earlier on, we could have spotted the issues sooner before we got too deep in implementing them.

These lessons are important because they have the possibility to prevent more dangerous or detrimental issues in future projects. If we can learn how and what to prevent now, we can save a lot more blood sweat and tears in the long run for future projects.

Appendix 1: Retrospective Meeting Agenda and Minutes

Agenda:

7:30 PM 12/8/2021, held online

Attendance:

Akash Gajendra
Joshua Mo
Nathan Wilke
Sehun Joo
Jordan Hassmann

Discussion topics

Technology issues
Review of all sections of website
What went well/not well
Communication on Discord vs. Real Life
Agile as a process

Minutes:

Asked a series of question:

- Most challenging problems you faced
  - **Nathan** - Having to learn React, Redux stores, and API's all in a short timespan
  - **Jordan** - Determining which tasks needed to get done first
  - **Sehun** - Trying to understand how to use the Firebase Firestore
  - **Akash** - Interfacing with Redux stores was a challenge
  - **Joshua** - The most challenging problem I faced was making it on time to meetings and working on things progressively throughout a week, instead of just last minute
- What went well
  - **Nathan** - Good communication, solid final product
  - **Jordan** - Technologies chosen worked fairly well
  - **Sehun** - Communication with each other
  - **Akash** - Good call with regards to the stack and constant communication
  - **Joshua** - I think that the UI design went the best. The flow of the website works really well and is very visually appealing.
- Thoughts on Discord vs IRL Meetings
  - **Nathan** - Discord is a lot more convenient, but less effective overall
  - **Jordan** - Discord is good communication for small scale meetings, but bad for large scale ones

  - **Sehun** - Discord meetings are comfortable, but IRL meetings would probably be more productive
  - **Akash** - Discord does well, but code reviews might be more productive IRL
  - **Joshua** - I think that online meetings make it harder to keep people accountable for their work but are more flexible with people's chaotic schedules.
- Regarding waterfall
  - **Nathan** - Agile is a lot better than waterfall. It allowed us to be more flexible in our overall goal and solution. It was also nice being able to spread the coding section throughout the entire development period as opposed to the final 2 weeks like our waterfall project
  - **Jordan** - Much preferred over Waterfall as it was more adaptable to issues we encountered due to the use of new and unfamiliar technologies.
  - **Sehun** -
  - **Akash** - I like the sprint based deliverable mechanism of Agile, helps adapt to any curveballs within the process more than Waterfall
  - **Joshua** - I like Agile much more than Waterfall. It is simply easier to get a picture of what's working and what needs to be done as you build and expand upon a working product every sprint.
- Favorite part of the project
  - **Nathan** - Seeing the working final product! I am super proud of the work we got done and like being able to show it off. I did find it super interesting working on the API's too.
  - **Jordan** - Implementing Redux stores
  - **Sehun** -
  - **Akash** - I'm happy I was able to gain experience with both front-end and back-end. Learning more about React.js and Redux was a highlight.
  - **Joshua** - My favorite part of the project was working with the other team members and learning how to connect APIs to display information on a website.

Assignments:

Collectively write the retrospective as needed. Fill out the survey

Appendix 2: Retrospective Survey

Survey Questions

1. Was the project fun and interesting?
2. Did it provide opportunities to innovate?
3. What went well? What didn't go well?
4. What lessons did you learn that you would share with your team?
5. What one topic do you want to make sure we address in the retrospective meeting?

Survey Responses

6. **Was the project fun and interesting?**
Jordan: I thought the project was very fun and interesting. The open endedness of the project provided a very engaging development process because we could really think through what would work best rather than stick to a set path.
Sehun: I thought the project was interesting because it was very close to a real life project that we would work on in the professional world.
Akash: I preferred this project over Project 2 because it's open-endedness allowed our team to explore whatever niches/features we wanted to integrate into the application. It enabled us to lead the way from product conception to final prototype delivery. Definitely fun and interesting!
Joshua: The project was interesting, and it gave the opportunity to learn a lot of new technologies. It also felt closer to the typical flow of code development in the industry.
Nathan: I found this project to be super interesting, and I had a lot of fun working on it. I personally prefer web development to a lot of other things in computer science, so I was a lot more passionate about this project than some others.

7. **Did it provide opportunities to innovate?**
Jordan: The project provided ample room to innovate. There was potential for innovation in the schema of our database, in our Redux store, areas to improve speed and efficiency, and in design.
Sehun: We had many opportunities to innovate, especially when it came to the technologies we were allowed to use, such as choosing our Web framework or database.
Akash: The volatile nature of the process enabled us to explore whatever routes we wanted to take the final product. No restrictions on technology encourage us to explore innovative solutions despite the roadblock.

Joshua: The project had lots of room for innovation. There weren't many limits on what we were and weren't allowed to do, so we were basically only limited by what we thought we could fit into the timespan of the project.
Nathan: Definitely. We started off with a simple idea of a musically based social media, and from this was able to add more and more features and innovations as we decided. This resulted in cool things I have never worked with before, including getting user location data and accessing external API's

8. **What went well? What didn't go well?**
Jordan: The majority of the technologies we chose were very easy to use and provided a good basket of tools. Waiting to host the site until the end, however, did cause some issues to arise.
Sehun: Our planning for the site went very well , and we had a good plan of what to execute. However, hosting the website had issues due to one of our apis not being compatible with firebase.
Akash: All three sprint weeks went well, however the final deployment led to some production issues that showed up right by the delivery deadline.
Joshua: Our initial connections with Spotify worked very well. However our connection to deezer and attempts to use some other APIs did not go as smoothly.
Nathan: Our overall scheduling and technology usage went mostly smoothly. The React App and Redux stores had minimal issues. Our API's didn't work as well as desired once moved to online hosting, so we probably should've worked on that section earlier.

9. **What lessons did you learn that you would share with your team?**
Jordan: I learned that more research into specific technologies as well as how they interact with each other is of great importance.
Sehun: I learned that we should test more early on live hosting to ensure that our page is working.
Akash: I learnt the importance of testing/validation web applications as deployment may lead to nullification of certain code segments and patches have to be created.
Joshua: I learned that when working on a project with API connections, I should focus on their functionality at the start of the project. This way I have more time to troubleshoot as the UI goes up around it.
Nathan: I learned that error debugging with API's can be quite difficult, and that using tools such as Postman can help greatly with this. These errors plus using localstorage can cause major issues.

10. **What one topic do you want to make sure we address in the retrospective meeting?**
Jordan: It would be good to address our plan going forward.
Sehun: Discuss how we can do better in future classes such as 431.
Akash: Open discussion about what went well and what didnt go so well?

Joshua: I would like to discuss what everyone's favorite part of the project was and if it affected what kind of work they would like to do (frontend, backend, etc).
Nathan: I think it would be good to go over our thoughts on this project verses previous ones