

Sprint 5: Sprint 5 & Overall Look Back

Introduction

Welcome, here are the lessons learnt throughout the process of our project that were either positive or negative. The team went through 4 previous sprints and accomplished many tasks and surpassed many hurdles.

Version Control: GitHub for collaborative development and version control.

IDE: Visual Studio Code (VS Co our Sprint 5 & Overall retrospective. Sprint 5 was dedicated to finalizing all unfinished tasks that were left in the backlog. It is also time to produce all relative final documentation related to this sprint. Furthermore, this is the opportunity to reflect on the overall project and lode) for code development.

Diagram Creation: Draw.io for creating diagrams.

Prototyping: Figma for designing and prototyping user interfaces.

Testing: Cypress for end-to-end testing, along with Angular's spec tests for unit tests using the Jasmine/Karma testing framework.

Continuous Integration/Continuous Deployment (CI/CD): Utilized GitHub Actions for automated CI/CD pipelines to ensure a smooth and efficient transition to the customer.

Architecture: Client-Server with Angular and Firebase.

Sprint 5

What went wrong

1 - Changes in format when using Capacitor

In Sprint 5, we had planned to transform our project into a mobile app using Capacitor. However, there were slight changes that needed to be made to the CSS. They weren't necessary since they did not affect the format grandly, but they are nice to have. Any extra tasks in the last sprint feel much harder to complete because of finals and deadline pressure.

2 - Report complexities

Composing the final report was difficult since the requirements weren't really clear. There was no sample document for us to base ourselves on. It was hard to decide what belongs in the final report without it seeming repetitive and redundant.

What went right

1 - Ease of use using Capacitor

Even if there were small format problems using Capacitor, it was an easy tool to use. After following a few directives, the app could be used on a mobile phone. This was a success since this process could have taken much much longer with a different technique.

2 - Finishing all Front/Back-End

Since there was a lot of documentation and finalizing deployment, having finished all programming tasks besides testing helped the workload for this sprint.

Overall

Negative Lessons

1 - Underestimating Documentation

There were many documents required during the development. Even documents that we did not realize needed to be formally produced such as the Code Management document. I believe that the separation of work could have been managed differently. Producing all these documents and diagrams takes time and effort, while some are crucial to the development. Although all these documents were successfully produced, I believe that in the next group project, the diagrams should be separated among every team member as much as possible. For example, if a team member is assigned to a feature, they should produce the activity diagram for that feature.

2 - Security measures when testing plays with database

We used spec tests and cypress tests throughout the project. This can be a risky process since it interacts with the database which was Firebase. At one point, we lost a huge chunk of the database when running

tests. This was a big lesson since it was very time-consuming to fix. Therefore, moving forward, we will adopt more careful habits when creating and running tests that interact with the database.

3 - Technological Adaptations

Learning and integrating new tools and languages presented challenges for some team members. To address these issues, we focused on building a supportive learning environment. We organized comprehensive training sessions that covered both the theoretical aspects and practical applications of the new tools. Additionally, experienced team members took on mentoring roles, offering one-on-one help and creating detailed documentation to ease the learning process. This experience demonstrated the critical importance of patience and structured support when introducing new technologies in a complex project.

Positive Lessons

1 - Planning in advance

The fact that we had prepared most of the tickets during sprint 1 greatly helped us stay on track during development. It allowed us to assess how much work needed to be done and organize how many issues should be done in every sprint.

2 - Getting Ahead

I believe that planning to do more during the first 2 sprints was very helpful. As shown in the release plans, our story points were much higher in the first and second sprints compared to the third and fourth sprints. This allows for some float when issues occasionally go into the backlog while still staying relatively ahead.