Sprint 4: A Look Back

Introduction

Welcome to our Sprint 4 retrospective. Sprint 4 was dedicated to finalizing all programming-related tasks, setting the stage for Sprint 5 to shift gears towards deployment preparations and potential overflow tasks. Despite facing a relatively light workload, external constraints such as other classes may have limited our available time. However, as a team, routines have been set for easier flow of operations. Now, as we move forward, it's crucial to leverage the groundwork laid in the previous sprint and focus on deploying our completed work.The following is an overview of the important components of our project.

Version Control: GitHub for collaborative development and version control.
IDE: Visual Studio Code (VS Code) 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.

What went wrong

1 - Longer development for spec tests

Although this sprint has fewer issues which should shorten development time, some components have longer specs tests than others depending on how much logic there is in the ts file. These tests can take up a considerable amount of time to develop, especially when we are near the end of the project's development and there exist well over 300 tests. This means that running these tests to check if the new ones in question pass or not can be timely.

2 - Code Refactoring due to duplications

After completing acceptance tests for individual condo pages and pushing them to GitHub, we encountered a setback when we had more than 60% code duplications. Thus, it was necessary to refactor the code which requires more time and resources as duplicates could lead to maintainability, scalability and performance issues.

3 - Data loss in Firebase

An unfortunate problem had happened during this phase. An important part of our database was lost. This most likely happened during the development of our tests, either spec test or cypress test. These tests interact with our database, so it is very important to be careful when running tests before we are sure they are safe. This fix took a lot of time.

What went right

1 - Lighter workload

This sprint felt like a lighter workload compared to other sprints, which made it easier to complete most of our tasks. Most teammates had 1 back-end implementation and 1 cypress test to accomplish. I believe we did the right thing and purposefully tried to make this sprint a lighter sprint because these 3 weeks were heavy with other classes as well.

Conclusion

As we conclude our retrospective on Sprint 4, we look back for important insights. Sprint 4 was an important milestone in our project as it finalized the development of our project. We changed our tasks distribution system in sprint 2 and it worked better in terms of productivity. We continued to be patient with each other's mistakes in this sprint and held up with communication. In all, Sprint 4 was a success for many reasons such as good communication, good productivity, good coding practices, and good feature outcomes. We look forward to building upon the work done during Sprint 4 and hope that the last sprint allows us time to properly spend time on completing the numerous documents that are demanded at the end of the project.