CS 407 Fall 2024 | Team 4

Course Clash | Sprint 2 Retrospective

Team Members

Thomas Carsello <tcarsell@purdue.edu>
Jason Bodzy <jbodzy@purdue.edu>
Zachary Heskett <zheskett@purdue.edu>
Sooha Park <park1343@purdue.edu>
Celia Patricio <cpatrici@purdue.edu>
Sergio Alvarez <alvar166@purdue.edu>

What went well?

One of the things that went pretty well during this sprint was the separation of duties we had as a team. Sooha did a lot of work with frontend UI components, as he has shown that he has a strong ability to create user-friendly and aesthetically pleasing web interfaces. Tom spent a lot of time working on backend API endpoints as he was particularly skilled with complicated PostgreSQL queries, and making those queries helped to simplify the process of using the API endpoints in the frontend.

Another strong aspect of this sprint was our ability to plan out the technical details of different subsystems before implementing them. A lot of the things we did this sprint were quite complicated, and interrelated. It was important that we built things in a way such that it would be simple to integrate future user stories into the structure. The way we planned out our database schema, and the API call protocols was super helpful in creating a long-term game plan that came together smoothly.

What did not go well?

One thing that did not go so well was the scheduling of which user stories to complete in what order. Many of our user stories were dependent on each other, and they often required other stories to be implemented before they could be started. There were three large groups of user stories that could be considered distinct from each other, and it would have been more efficient to have different members of the team work on those disjoint areas in a parallel way. Instead, we tackled each functional area one at a time, and it created a situation where only one person could work on things at a time, since all the possible parallel items were already done. We did not benefit from the size of our team as much as we could have if we optimized our scheduling better for concurrent implementation.

How should we improve?

The way we plan to improve for sprint 3 is to create a topological ordering of our user stories, and use that to inform the way in which we schedule them. That way, we can better make use of the number of people on our team, and get more work done early. We will have to identify which

user stories depend on each other in order to construct a DAG and run a topological sort on it. This can help us identify which user stories should be done in what order, and when there is an opportunity to develop things in parallel/