BoilerScout Sprint 2 Retrospective

Team 15

Project Coordinator: Mohammad Haseeb

Baris Dingil, Hardy Montoya, Jacob Miecznikowski, Selin Ovali, Terry Lam

(a) Include general retrospective review for this sprint.
(b) Ensure to list ALL unsuccessful user stories and tasks with detailed discussions. (They should be in
line with the Sprint Planning Document for the next sprint.)

1 What went well? (0.5 point)

Sprint 2 and its demo went far more smoothly than Sprint 1. We improved greatly in time estimation of tasks and were more comfortable with the technologies we were using for our project. All but one member finished all the user stories and tasks assigned to them, compared to Sprint 1 when we mostly failed to do so. Furthermore, we were more prepared to demo now that we knew what to expect in a demo.

In addition to that, our team communication has drastically improved. Ever since we made the switch to Slack, the team has been able to share progress and results a lot easier across channels. It has been really easy to keep each other accountable and check up on each other's work. Communication has been the key reason why our frontend and backend were able to integrate so much more successfully than the first sprint. Everybody has been doing a great job of asking each other for help when needed, and doing their best to help one another in order to get tasks done. By asking questions and communicating our need for help, every member has learned important knowledge that will help us complete tasks faster in this upcoming sprint.

Following are the tasks that were completed during Sprint 2:

- Ability for user to login and stay logged by saving a token and the user id
- Ability for user to sign up with their Purdue email
- Sending a verification email after signing up
- Ability to request the verification email to be resent if needed
- Ability for user to reset password if they forget their password and receive a new one via email
- A "Scout" page where users can search for other members by querying for a name, skill, or course.
- Ability to view other members' profiles and own profile
- Ability for users to edit their own profile information, including their personal biography, personal skills, or enrolled courses.

2 What did not go well? (0.5 point)
From a backend perspective, we did not have many unit tests that supported our new features and API that we developed. This came back to affect us when the frontend was attempting to integrate and test our API endpoints. There were a few edge cases that the backend team did not catch with manual testing, and it left us trying to administer quick fixes because we did not have time to sit down and properly address the issues.

Additionally, two user stories involving messaging were not complete.
Following are the tasks that were not completed during Sprint 2:

- <User Story 9> Private messaging (backend)
- <User Story 10> Inbox (backend)

The main cause of our unfinished user stories was poor time management. Too much time was spent on other things outside of class and some tasks were not able to be completed. As the deadline approached for sprint 2 reviews, we scrambled to work on these tasks. But, as happens frequently in software engineering, unforeseen errors arose that slowed progress down so much that we are unable to finish by the due date.

3 How should you improve? (1.0 point)
While the overall sprint was successful, there are still major portions of our project that need to be focused on. One aspect of our project that needs to be improved is the user experience and UI design, which the frontend plans to focus on during Sprint 3. We plan to have a few meetings throughout the sprint where we sit down as a team and go through our app while giving feedback on the UX/UI to our frontend developers.

Additionally, we will have be having in depth discussions between our frontend and backend developers to sync up and ensure that that are no discrepancies when integrating new API endpoints. For example, we will make sure that we discuss exactly how we need to structure request bodies and response objects when outlining intended input/output. This will ensure that time is not wasted writing code that does not line up with our functional requirements. Additionally, for new features and pull requests, members will properly review each other's code to give feedback. These types of discussions will allow us to tie up this sprint as smoothly as possible.

For the unfinished tasks, we will starting working and finish those as soon as possible so that we may move on and complete other tasks. During sprint 3, these tasks will be followed by everyone and frontend will join to backend and together there will be a finished User stories. While coding we should manage our time better and use each

other's knowledge as much as possible to not have problems and lack of knowledge. We should start coding earlier to have time to fix the code in case of unusual errors.

Lastly, we will also be focusing more on testing this sprint. More specifically, writing unit tests on the backend. In the previous two sprints, we have not done a good job of writing a robust test suite for our API, as we had only relied on manual testing. Moving forward however, we want to create unit tests for not only new features, but to also circle back to our previously written API and write tests for those too. This will allow us to further the scope on our testing and catch more cases than we would by just manually testing.