

Mixbook

Mixology Application

Sprint 1 Retrospective

<https://github.com/alexcoll/mixbook>

Team 3:

Alexander Coll (acoll@purdue.edu), Matthew Fouts

(foutsm@purdue.edu),

John Tyler Preston (presto@purdue.edu), Nicholas

Zenobi (nzenobi@purdue.edu)

What went well?

Overall we were able to complete all of the user stories as well as set up our infrastructure and development environment.

User stories:

- As a user, I would like to edit my own recipes
 - Created edit recipe UI
 - Implemented backend handling of editing a recipe
 - Tested and ensured robustness of editing recipes, checking common corner cases and the likes
- As a user, I would like to be able to view other user's profiles
 - Created a profile view UI
 - Backend loads information in less than a second
 - Options related to another person's profile vs. the user's profile differ correctly
 - The UI is interactive for the user's profile
 - Tested and ensured robustness of viewing other users' profiles, checking common corner cases and the likes
- As a user, I would like to be able to have my own user profile
 - Created a UI to view own user profile
 - The backend loads any additional information that is not available to other users
 - The backend handles all operations regarding a user's profile
 - The backend loads all relevant information in less than a second
 - Security and privileges regarding a user's profile are enforced on the backend
 - The UI is interactive for the user's own profile
- As a developer, I would like for users to be able to reset a password if forgotten
 - Created a UI on website to request to reset password
 - Created a UI on website to handle changing of the user's password after clicking on the link in the email
 - Backend handles creating password reset tokens and persisting them as well as sending the email with the reset link
 - Backend handles cron job related cleanup of expired tokens
 - Backend handles saving the new password
 - Backend validates that a token isn't expired as well as ensures it belongs to the requesting user

- Backend and frontend validate that the password meets the minimum requirements
 - Unit testing and usability testing ensured ease of use and robustness of password recovery
- As a user, I would like to sort recipes based on ratings
 - Backend return a list of recipes with their ranking
 - Frontend provides ability to sort in either ascending or descending order based on ratings without the UI freezing
 - Unit and usability testing of sorting to ensure seamless sorting
- As a developer, I would like to remove the bugs from recipe reviews
 - Frontend no longer crashes when clicking on the button to add a recipe review
 - Backend accommodates data required to create recipe reviews in an efficient and robust manner
 - Frontend no longer crashes when a user submits their review of a recipe
 - Unit and usability testing of recipe reviews ensured ease of use and robustness
 - Ensured the backend is properly handling all review operations
 - Backend ensures a user does not leave a review on their own recipe and the frontend prompts the user that they cannot review their own recipe
- As a user, I would like to be able to go back and edit my reviews
 - Frontend now provides a button to move to a screen to edit their review
 - Frontend now has a view to edit the review
 - Backend handles editing reviews
 - Unit and usability testing ensure that the process is seamless and robust
 - Privileges are enforced in that a user can only edit their own review
 - Review edits propagate in less than a second
- As a user, I would like to receive profile badges/awards
 - We came up with a variety of potential badges/awards that users can achieve
 - Backend tracks these badges and all associated data
 - Backend tracks progress toward badges
 - Unit and usability testing ensure that the badges are easy to understand and that the process is robust when awarding badges as well as tracking progress

- As a user, I would like to view badges/awards that I have received
 - Frontend displays progress toward badges on user's profile
 - Frontend displays earned badges and backend loads the badges for a user on a user's profile
 - Unit and usability testing ensure that process is robust when loading badges as well as badge progress
 - Front end loads badges and badge progress for other users as well so that one can see a friend's badges/badge progress
 - Badges and requirements for badges both load in less than a second

What did not go well?

- Build fails
 - One team member had difficulties building the react native application
- Silly errors that were easily preventable by simple code reviews
 - An occasional error would paralyze the app all due to some trivial error (i.e. a typo) and would take hours to diagnose, whereas a simple code review would have found the issue much faster
- Had to slightly modify user stories as new issues presented themselves
 - Password recovery happens completely outside the app, outside of if a user wants to click on the link in the app to go to website page to request new password, though could just have easily typed in the URL themselves
 - Certain design decisions made some acceptance criteria need to be changed to a more nuanced view
- Steep learning curve/ramp up time for new group member
 - Since we added on a new member to the group that was not part of the original development team when we developed the application in CS 408, the ramp up time was significantly longer than expected
 - The new member also faced a steep learning curve for how react native works outside of his own use cases

How can you improve?

- **More Standup Meetings/Consistency of Member Attendance**
 - Follow the standup meeting style as described in class
 - Communicate our troubles
 - Communicate the task we are working on

- Ensure all members can make it and have members communicate well in advance if they cannot make it to the meeting
- **Running the App**
 - Ensure that the app can be built for every team member with consistent results
- **Communication of issues**
 - Make sure team members actively communicate any issues they are having to allow speedy fixes of issues
- **Buddy coding/coding sessions**
 - This would help brainstorm solutions to issues on the fly
 - Any technical troubles would be resolved far quicker
 - It would improve communication regarding progress on different features