Mixbook

Mixology Application

Sprint 1 Planning Document

Team 3:

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

John Tyler Preston (prestoj@purdue.edu), Nicholas Zenobi (nzenobi@purdue.edu)

Sprint Overview: For the first sprint we will be working on the review and recipe features and UI design. By the end of the sprint users will be able to edit a review and edit a recipe. He or she will be able to experience a fully set up infrastructure that is scalable and secure. Users will also have their own profiles. Users will be able to view others profiles and their badges. Users will also be able to gain badges for various achievements such as creating 10 recipes. Users will also be able to reset their password if forgotten.

Scrum Master: John Tyler Preston

Meeting Schedule: Tue 4:30-5:30

Risks/Challenges: Some team members are not familiar with various frameworks, languages, or technologies that we will be utilizing, such as MySQL, JavaScript, and React Native. The bulk of this sprint will be revolving around laying the groundwork for future development as well as catching up and teaching those who are not familiar with various technologies said technologies while improving existing features and adding some more. Another difficulty that we will face will be initially working out our team dynamic and working together, as not every team member has worked together before. Overall, the first sprint will be more about figuring out the ropes than anything, especially ramping up those not familiar with the codebase.

User Story 1: As a user, I would like to edit my own recipes

#	Task Description	Estimated Time	Owner
1	Create edit recipe page for the front end	5 hours	Matt
2	Implement handling editing a in the back end	3 hours	Tyler
3	Unit testing and usability testing	4 hours	Matt
4	Set up backend infrastructure	10 hours	Tyler
5	Set-up development environment and consistent ecosystem	12 hours (3 hours per team member)	Alex, Matt, Nick, and Tyler

Acceptance Criteria:

- ❏ Given a recipe exists, when the user opens a recipe, then an edit recipe option should be easily visible and intuitively accessible from the recipe page.
- ❏ Given a recipe exists, when the user edits a recipe, then the server should process this request and save the necessary recipe change of information in the database.
- ❏ Given a recipe exists, when the user edits a recipe with invalid information, then the front-end should reject the edit request.
- ❏ Given a recipe exists, when the user edits a recipe with invalid information, then the back-end should reject the edit request.
- ❏ Given a user successfully edits a recipe, when the user goes back to the recipe, then the new information should appear.

The front end for this is mostly creating a form that allows you to edit your recipe information. The backend mostly deals with validating, processing, and saving this information into the database as an updated recipe.

User Story 2: As a user, I would like to be able to view other user’s profiles

#	Task Description	Estimated Time	Owner
1	Create user profile page for the front end	5 hours	Matt
2	Implement handling user profiles in the back end	3 hours	Tyler
3	Unit testing and usability testing	4 hours	Matt
4	Optimize some general user related processes on backend	2 hours	Tyler

Acceptance Criteria:

- ❏ Given a user exists, when the user opens a user’s profile, then the user interface displaying that user’s profile should be user friendly.
- ❏ Given a user exists, when the user opens a user’s profile, then the server should process this request and load the user profile information in the database in less than a second.
- ❏ Given a user exists, when the user opens their own profile, then the user interface should include additional options regarding their own profile.
- ❏ Given a user exists, when the user opens a user’s profile, then the information should be loaded in a bug-free manner and be displayed properly.
- ❏ Given a user successfully loads a user’s profile, when the user interacts with the user’s profile, then the app should allow interactivity.

The front end for this is mostly creating a form that allows you to view a users profile. The backend mostly deals with validating, processing, and saving this information into the database as a user profile.

User Story 3: As a user, I would like to be able to have my own user profile

#	Task Description	Estimated Time	Owner
1	Create user profile page options for interacting with own user profile for the front end	3 hours	Nick
2	Implement handling user profile interactions for own user profile in the back end	3 hours	Tyler
3	Unit testing and usability testing	2 hours	Nick
4	Ensure security and privileges are enforced on backend for user profile interactions	2 hours	Tyler

Acceptance Criteria:

- ❏ Given a user exists, when the user opens a user's own profile, then the user interface displaying that user's profile should be user friendly.
- ❏ Given a user exists, when the user opens a user's own profile, then the server should process this request and load the user profile information in the database in less than a second including additional information not available to other users.
- ❏ Given a user exists, when the user opens their own profile, then the user interface should include additional options regarding their own profile.
- ❏ Given a user exists, when the user opens a user's own profile, then the information should be loaded in a bug-free manner and be displayed properly.

- ❑ Given a user successfully loads a user’s own profile, when the user interacts with the user’s profile, then the app should allow interactivity.

The front end for this is mostly creating a form that allows you to interact with a user’s own profile. The backend mostly deals with validating, processing, and saving this information/interaction into the database as a user profile.

User Story 4: As a developer, I would like for users to be able to reset a password if forgotten.

#	Task Description	Estimated Time	Owner
1	Create an option on the front end to allow the user to reset their password	4 hours	Matt
3	Create a page on the app that allows the user to change their password	5 hours	Matt
3	Implement password reset handling on the back end	3 hours	Tyler
4	Unit testing and usability testing	2 hours	Alex
5	Ensure the process of resetting the password is secure	2 hours	Alex

Acceptance Criteria:

- ❑ Given a user loads login page, when they can’t login, then there should be a reset password button in an easily accessible location.

- ❏ Given that a user exists, when the user clicks the button to reset their password, then they should be presented with a screen indicating an email has been sent.
- ❏ Given that a user has submitted a password reset request, when the user checks their email, then the server should have sent an email to the user with a link to reset their password.
- ❏ Given that a reset link is active, when the link in the email is clicked, then the app should then open to a page that allows the user to reset their password.
- ❏ Given that a user has filled out the reset password form correctly, when the user submits their new password information, then they should be able to login immediately with their new password.

The front end of this story is creating a button that will trigger the sending of an email to reset the user's password. It will also incorporate the creation of a new page to change the user's password. The backend will deal with validating and changing the user's password. It also will make sure the information being transmitted it secure.

User Story 5: As a user, I would like to sort recipes based on ratings.

#	Task Description	Estimated Time	Owner
1	Create an option on the front end to allow the user to sort recipes based on their rankins	2 hours	Nick
2	Sort the recipes by ratings on the front end	5 hours	Nick
3	Unit testing and usability testing	3 hours	Nick
4	Have the server return the ratings	3 hours	Tyler

	of each recipe as well as the names when the front end requests the list of recipes		

Acceptance Criteria:

- ❑ Given that the user is on the recipe list page, when they requests a list of recipes from the server, then the server will return the names of the recipes as well as the ratings.
- ❑ Given some recipes exist that have been rated, when the user selects the button to sort by ratings, then the list of ratings will become sorted according to ratings.
- ❑ Given that the user is engaged with another form of filtering, when they attempt to sort by ratings, then the list will become sorted according to ratings.
- ❑ Given some recipes exist that have been rated, when a user attempts to sort the recipes based on rating, then the sorting will occur seamlessly without freezing the frontend.
- ❑ Given some recipes exist that have been rated, when a user attempts to sort the recipes based on rating, then the sorting will be able to occur in either ascending or descending order in terms of rating.

The front end of this story is creating a button that will trigger the list of recipes to be sorted by rankings ascending or descending. The back end will provide the front end with the rankings of each recipe when the front end requests a list of recipes.

User Story 6: As a developer, I would like to remove the bugs from recipe reviews.

#	Task Description	Estimated Time	Owner
1	Prevent crashing when a user clicks button to add	5 hours	Nick

	recipe		
2	Prevent crashing when user is finished with their review and attempts to submit it	4 hours	Nick
3	Prevent a user from adding a review to their own recipe	3 hours	Nick, Alex
4	Ensure the back end is saving and retrieving recipe reviews properly	3 hours	Tyler
5	Unit testing and usability testing	3 hours	Nick

Acceptance Criteria:

- ❏ Given that the user is logged in, when the user attempts to add a review to a recipe, then the application should not crash.
- ❏ Given that the user is logged in, when the user attempts to review his/her own recipe, then the app should show an alert telling them this is not allowed.
- ❏ Given that the user has satisfied the requirements for a review, when the user has completed the review and submitted it, then the review should then allow other users to see that review.
- ❏ Given that the user is logged in, when the user attempts to review a recipe, then they should be prompted with a method of inputting a review.
- ❏ Given that the user has satisfied the requirements for a review, when the user submits the review, then they should receive some feedback letting them know their review has been submitted.

The front end of this story is ensuring that users cannot review their own recipes and that the correct information is getting sent to the server to prevent crashes. The backend of this story is ensuring the serving is accepting review data correctly and saving it to the database properly.

User Story 7: As a user, I would like to be able to go back and edit my reviews.

#	Task Description	Estimated Time	Owner
1	Add a button that allows users to edit their own review on a recipe	2 hours	Alex
2	Implement an edit mode on reviews that allows the user to edit and submit an update to their review	4 hours	Alex
3	Implement the updating of a review on the back end	4 hours	Tyler
4	Unit testing and usability testing	3 hours	Alex

Acceptance Criteria:

- ❑ Given that a user is logged in, when the user looks at one of their past reviews, then they should see a button allowing them to edit their review.
- ❑ Given that a user is logged in, when the user clicks the edit review button, then the review should become editable.
- ❑ Given that the user has satisfied the criteria for a review, when the user submits the edited review, then it should automatically update and become visible to the current user.
- ❑ Given that a user is logged in, when the user finished editing a review, then the changes should propagate and become visible to other user within one second.
- ❑ Given that a user is logged in, when a user tries to edit someone else's review, then they will be denied permission to edit someone else's review.

The front end of this story involves adding an edit button and implementing an edit mode for the users review. The back end involves adding functionality to update a review in the database with the new version.

User Story 8: As a user, I would like to receive profile badges/awards

#	Task Description	Estimated Time	Owner
1	Come up with various accomplishments that will be rewarded with badges	4 hours (1 hour per team member)	Alex, Matt, Nick, and Tyler
2	Implement functionality on back end to save badges and awards to a user's profile	5 hours	Tyler
3	Implement functionality to track a user's progress towards these badges/awards	5 hours	Alex, Tyler
4	Unit testing and usability testing	3 hours	Alex

Acceptance Criteria:

- ❑ Given that the user has achieved the necessary points, when the user reaches a predetermined goal, then they should receive a badge/award.
- ❑ Given that the user is logged in, when the user receives a badge/reward, then they should be notified.

- ❏ Given that a user currently has a award/badge, when the user falls beneath the threshold for that badge/award, then they will no longer have that badge/award.
- ❏ Given that the user has achieved the necessary requirements for the badge, when the user checks their badges, then they should immediately see the new earned badge.
- ❏ Given that the user drops below the requirements for a badge, when the user checks their badges, then they should immediately see that they no longer have the badge.

The front end for this user story will involve creating the icons for the badges/awards. The back end work will involve keeping track of a user's progress towards the various goals.

User Story 9: As a user, I would like to view badges/awards that I have received.

#	Task Description	Estimated Time	Owner
1	Implement a new page for users to view their awards	5 hours	Matt
2	Allow the user to see their progress towards the badges/awards	5 hours	Matt
3	Unit testing and usability testing	3 hours	Alex

Acceptance Criteria:

- ❏ Given that the user is logged in, when the user looks at their profile, then they should be able to see what awards they have earned.
- ❏ Given that the user is logged in, when the user looks at their profile, then they should see their progress towards badges/awards they have not earned yet.

- ❏ Given that the user is logged in, when the user looks at their profile, then they should be able to see the requirements for all of the awards.
- ❏ Given that the user is logged in, when the user looks at their profile, then they should be able to load and view their badges in less than a second.
- ❏ Given that the user is logged in, when the user looks at their profile, then the requirements should load in a nominal amount of time towards badges.

The front end work will involve creating a page on the app that will allow users to view what awards/badges they have earned. It will also show users the requirements for other badges/awards and their progress towards them. The back end will involve creating calls that will allow the front end to retrieve information about the badges/awards for a user.

Remaining Product Backlog:

Functional

1. As a user, I would like to choose from pre-populated recipes as well as user-added recipes.
2. **As a user, I would like to receive push notifications for a variety of events (if time allows).**
3. As a user, I would like to be able to recommend recipes to other users.
4. As a developer, I would like for users to be able to lock their accounts.
5. As a user, I would like to have pagination of results.
6. As a user, I would like to have access to even more ingredients from which to choose.
7. **As a user, I would like to see images of cocktails (if time allows).**
8. **As a user, I would like to add and remove images of cocktails (if time allows).**
9. As a developer, I would like to overhaul the current error handling platform.
10. As a developer, I would like to overhaul the interchange format of data to a more standardized version of JSON.
11. As a developer, I would like to remove the use of deprecated functions as much as possible.
12. As a developer, I would like overhaul the request handling to improve and make request handling more standardized.
13. As a developer, I would like redesign the add recipe page to be more user-friendly.
14. As a user, I would like to say if reviews are helpful or not.
15. As a user, I would like to filter the reviews based on rating.
16. As a user, I would like to have a profile rating that reflected the ratings of the recipes I have created.
17. As a user, I would like to be able to find and sort users by their profile ratings.
18. As a user, I would like to see recipes that I have made.

Non-Functional

1. Should be able to use this application on an Android device
2. Should use some sort of near-real-time or real-time solution for updates, utilizing AWS services where possible
3. Should be scalable to be user demand, utilizing AWS services where possible
4. Should meet standard security practices, utilizing AWS services where possible

5. Should have an interface that is simple enough so that the average user would be able to use the application without struggling
6. Should be able to have negligible downtime while allowing for continuous development, utilizing AWS services where possible
7. Should be able to be high performing while maintaining proper security practices, utilizing AWS services where possible
8. Should have the ability to receive feedback in some from
9. Should design database schema independent of application so that it follows proper database schema design and can be ported for other uses
10. Should incorporate a three-tier architecture, utilizing AWS services where possible
11. Should provide fault-tolerant infrastructure to minimize downtime, utilizing AWS services where possible
12. Should implement AWS best practices for all applicable aspects of infrastructure that utilize AWS services
13. Should thoroughly document all code base in order to enhance maintainability
14. Should design backend in such a way to provide for easy testing at all phases of development
15. Should utilize AWS services as much as possible to avoid custom configuration of individual services that only serves to accomplish what another AWS service already provides