

Sprint 2 Report

Product: Recipe Retriever

Team Name: RexFetch

Members: Kim Ho, Vyankatesh Nandapurkar, Jackson Kohls,
Kevin Yosifov, Prithvi Arunshankar

Date: 05/12/2023

Note: The “team” or “group members” here refers to all team members, not just the development team.

Actions to Stop Doing:

1. Stop waiting until the last minute to do our work. We should be starting early so that if we come across any unexpected problems or obstacles, we can figure them out and continue with our work to create the features optimally.
2. The team members should stop waiting for people to assign them tasks and rather take up tasks for themselves that they know they are capable of accomplishing within the timeframe of the Sprint. We should be taking initiative and if we do have any problems, talk to the Scrum Master or Project Owner to ask for help or advice.

Actions to Start Doing:

1. Our team should expand on working with tests in order to make sure our code is fully functional and keep using tests to make sure that new code doesn't cause any issues with the existing code. This is especially true for the frontend development.
2. We should start talking about what tasks we are doing and keep each other accountable to do the tasks we self-assign ourselves to. We should communicate whenever we have issues and see if any other team members can help us figure them out so we can continue our work.
3. If team members are unable to complete their work, they should tell the Scrum Master or Product Owner why they are unable to finish their tasks. Also this can help as the Scrum Master/Product Owner can assign somebody else to the task to make sure it gets done if it is integral to the project or integral to the next Sprint.

Actions to Keep Doing:

1. We should keep track of our tasks with Trello and assign ourselves work from there. It has been very helpful toward the organization of our team.
2. We should keep up our technical communications. For example, we've been talking in discord about different components we worked with individually to see how we can integrate them together.

Work Completed/Not Completed:

User Story 1: As a user, I want to look up a recipe based on the ingredients in my kitchen. (MVP)

1. Set up recipe API calls and parsing the recipe data into the outgoing JSON format that is described in the communication_docs.md or backend_doc.md [Incomplete]
2. Make frontend fragment that displays the received backend recipe data [Incomplete]
3. Database communication for recipes (testing) [Done]
4. Communication documentation (describes the json objects being sent, request format, etc.) [Done]
5. Backend to frontend communication (implements http requests from server to app) [Incomplete]
6. Frontend to backend communication (implements http requests from app to server) [Incomplete]

User Story 2: As a user, I want my login information to be stored in a profile, so I can save some information about myself (favorite recipe, my fridge content, etc.)

1. Set up user database and integrate with server [Incomplete]
2. Front end changes [Incomplete]
 - a. Grocery Page [Done]
 - b. Dynamically Create Recipe Cards [Incomplete]
 - c. ~~Spinner~~ Recipe Randomizer [Done]
 - d. Shared View Model for Data Sharing [Removed]
 - e. Settings - (Substitutions & allergens) [Done]
 - f. Error Messages [Done]
3. Synchronize data with live front-end (authentication) [Done]
4. Make a new user functionality versus returning user authentication. [Incomplete]

Work Completion Rate:

- Total number of user stories completed: 0 of 2
- Estimated ideal work hours spent:

User Story 1: As a user, I want to look up a recipe based on the ingredients in my kitchen. (MVP)

- 35% Complete

User Story 2: As a user, I want my login information to be stored in a profile, so I can save some information about myself (favorite recipe, my fridge content, etc.)

- 60% Complete