

# **Let's Cook!**

## **Amanda Hoelting, Ellison Largent, Lily Logan, Will Marceau**

### **Project Plan**

#### **1. Management Plan**

Our team decided to organize our group by assigning roles: project manager, frontend manager, backend manager, and documentation manager. Each role is in charge of that area but team members are more than welcome to float between regions as needed. We assigned our team as the following:

- Project Manager - Lily Logan
- Frontend Manager - Will Marceau
- Backend Manager - Amanda Hoelting
- Documentation Manager - Ellison Largent

Our team will make decisions by having discussions on important issues. The “manager” of said area will make the final call.

WALE will meet on Tuesdays each week for a longer scheduled meeting. The two times we all have available are 10 am-11:30 am and 5:30 pm-7 pm. The Monday before, we will all decide to meet at one of the two time slots depending on the team's weekly availability. We will also meet at least two other times a week for shorter meetings to go over current issues. Every day, each member will update the rest of the group by sending a text to our group chat containing what they did that day and what they will do tomorrow. Meetings will be scheduled as needed.

Communication will be through a text group chat.

#### **2. Work Breakdown Schedule (import from Jira)**

[Jira timeline](#)



### 3. Monitoring and Reporting

Like previous stated, team members will update the group with daily progress checks. To track individual tasks, we will be using Jira to mark the time when a task is completed.

### 4. Building Plan

The following plan consists of a tentative plan which will allow us to build the system for this application. This plan is not concrete and may change slightly to complete our application within the timeframe. We have broken down the system into the following features: login, main page (contains recipe deck), profile, filter tab, your recipes, liked recipes, grocery list, meal plan, recipe creation.

#### 4.1. Week 2: Development of Initial Project

Team:

- 4.1.1. Create project topic
- 4.1.2. Write initial SDS and SRS

#### 4.2. Week 3: Database and UI Structure

Backend:

- 4.2.1. Import recipe database.
- 4.2.2. Clean database such that it can be usable for the deck feature.
- 4.2.3. Initialize user database.

Frontend:

- 4.2.4. Set up basic UI structure (login page and recipe cards)

#### 4.3. Week 4: Classes, Deck, and Profiles

Backend:

- 4.3.1. Recipe class which contains recipe name, rating, prep time, etc.
- 4.3.2. User class which contains user's name, contact info, etc.
- 4.3.3. Define relationships between recipes and users.

Frontend:

- 4.3.4. Layout for recipe deck (swiping cards, user interactions such as likes/dislikes)
- 4.3.5. Skeleton for profile page.

#### **4.4. Week 5: Login and Swiping**

Backend:

- 4.4.1. Implement login functionality.
- 4.4.2. Implement filtering and swiping features.
- 4.4.3. Implement user management to keep track of user's saved, liked, and disliked recipes.

Frontend:

- 4.4.4. Design login page.
- 4.4.5. Build recipe deck and filtering features.

#### **4.5. Week 6: Meal Plan and Grocery List**

Backend:

- 4.5.1. Implement meal plan feature.
- 4.5.2. Implement grocery list which was curated from the meal plan.
- 4.5.3. Implement functionality for users to create their own recipes.

Frontend:

- 4.5.4. Complete swiping feature for deck
- 4.5.5. Design meal plan and shopping list

#### **4.6. Week 7: Recipe Creation**

Backend:

- 4.6.1. Support frontend

Frontend:

- 4.6.2. Complete meal plan and grocery list.
- 4.6.3. Integrate user-created recipes feature.

#### **4.7. Week 8: Testing, Debugging, and Feature Refinement**

Team:

- 4.7.1. Conduct tests to reach edge cases and faulty code.
- 4.7.2. Debug any features flagged by tests.

#### **4.8. Week 9: Testing, Debugging, and Feature Refinement**

Team:

4.8.1. Conduct tests to reach edge cases and faulty code.

4.8.2. Debug any features flagged by tests.

#### **4.9. Week 10: Completed and Present!**

## **5. Rationale for the Plan**

We have divided the system into distinct components to ensure a structured and efficient development process. By establishing solid backend functionality early on, the frontend team can seamlessly integrate user-facing features without being delayed by pending backend tasks. This parallel workflow enables faster and smoother development. The risk with this plan is timing. The frontend team is relying on the backend group to complete certain features in order to maintain this timeline. This might come in to play and be used against us at first, but with communication within the team and flexibility, this application will be completed by Week 10.