Foodie

Product Backlog

Team members

Guocheng Wei, Jiayi Kou, Yuchen Zhang, Zhaoya Sun

Problem Statement

Have you ever found yourself looking into the fridge but had no idea what can be done with the ingredients? With Foodie, you will be able to explore a variety of creative recipes designed just for the ingredients found in your fridge! Foodie also creates a platform for you to customize and collect personalized recipes with your chosen ingredients. Most of the recipe apps out there are recipe-based and the users have to prepare the ingredients if they want to follow a recipe. Foodie thinks otherwise. This ingredient-based recipe app allows the users to customize their ingredients and find the right recipes for them.

Background Information

While many apps and websites aim at providing users with numerous recipes about how to cook certain dish, none of them provides the recipes based on what the users have. For sure, those apps and webs will give users an ocean of receipts and instructions to make good meals, but they never consider how to make the most out of already existent ingredients. Our goal is to fix this deficiency. Edamam API will be used to help users find as many recipes as possible and make cooking more convenient and diverse. We believe our web app will be much more user-friendly and considerate in providing recipes, especially for the users who are not willing to spend too much time on shopping, and therefore, bringing users a more delicate cooking journey.

Environment

Foodie will be a JavaScript web application built using Meteor.js frameworks. HTML5, CSS3, React, and Materialize UI will be used for the frontend implementation and design. The backend will collect recipe data from Edamam API as our recipe library. We will be using MongoDB as our database for storing user profile and data. User data will be backed up in the form of JSON documents. The web application will be hosted and managed by a cloud service platform.

Requirements

Functional

ID	Function Requirements	Hours	Status
1	As a user, I would like to create an account in the application	8	Planed for sprint 1
2	As a user, I would like to log in to my account	5	Planed for sprint 1
3	As a user, I would like to log out my account	2	Planed for sprint 1
4	As a user, I would like to have email as the unique username	5	Planed for sprint 1
5	As a user, I would like to change my account password	8	Planed for sprint 1
6	As a user, I would like to view my own personal profile	20	Planed for sprint 1
7	As a user, I would like to save my favorite recipes	10	Planed for sprint 2
8	As a user, I would like to enter the ingredients I have into the app	20	Planed for sprint 1
9	As a user, I would like to see recommended recipes based on my ingredients	20	Planed for sprint 1
10	As a user, I would like to see the detailed information of each recipe	15	Planed for sprint 1
11	As a user, I would like to have a back to top button	5	Planed for sprint 1
12	As a user, I would like to receive email notifications if my account is created or my password is changed	15	Planed for sprint 2
13	As a user, I would like to comment on other people's recipes	15	Planed for sprint 2
14	As a user, I would like to reply to other people's comments on the recipe page	10	Planed for sprint 2
15	As a user, I would like to change my avatar picture.	10	Planed for sprint 1
16	As a user, I would like to "like" other people's recipe	10	Planed for sprint 2
17	As a user, I would like to see how many views each recipe has	10	Planed for sprint 2
18	As a user, I would like to know what ingredients I don't have in stock from each recipe page	20	Planed for sprint 2
19	As a user, I would like to delete my account	5	Planed for sprint 1
20	As a user, I would like to edit my account information	10	Planed for sprint 1
21	As a user, I would like to sort/filter my recipe results based on likes	10	Planed for sprint 2
22	As a user, I would like to sort/filter my recipe results based on views	10	Planed for sprint 2

23	As a user, I would like to search recipes with keywords	15	Planed for sprint 2
	Total	258	

Non-Functional

ID	Non-Functional Requirements	Hours	Status
24	As a user, I would like to have an interactive and responsive user interface	10	Planed for sprint 1
25	As a user, I would like to store my information and recipe data securely	10	Planed for sprint 2
26	As a user, I would like to have helpful error messages	5	Planed for sprint 1
27	As a developer, I would like to improve the response time	10	Planed for sprint 2
28	As a developer, I would like the application to be functional across multiple browsers	5	Planed for sprint 2
	Total	40	

Usability

Our web application will support most browsers and be accessible to the average users. It will be user-friendly and intuitive for users. There will not be countless complicated steps about setup and usage for users to follow. In addition, the integration of the Edamam API will be bugless and the API will provide users with abundant resources. We will also ensure our web application fit in any size of screens and devices.

Security

Security will be an important part for web application since the users' information should not be shared to others, except for the username. Since Meteor.js will be our backend framework, different security protections will be used based on its library functionalities. On the frontend, cookie theft will be prevented by setting the http-only flag on the cookies. Users' inputs are always filtered and sanitized to prevent the command injection. Last part of the security check will be transmitted between frontend and backend.

Scalability

As more and more users register on the website, the web database will be able to increase its total output under an increased load when resources (typically hardware) are added. The system should process to handle a growing amount of work, or have the potential ability to be enlarged in order to accommodate that data growth. We will strive for a server response time of under 200 milliseconds. We would utilize multi-threads to run independent processes concurrently to improve response time further if time allows.

User cases

Case 1: Create an account

Action:	System Response:
1. User fills out all related information and click "Sign Up" button	2. Application gives a success message and create an account for the user. The user will receive an email notification about the new account.

Case 2: Log in as a user

Action:	System Response:
1. User fills out username and password and clicks on "Login" button	2. Application is redirected to user profile page on success; otherwise, am error message will show up

Case 3: Log out as a user

Action:	System Response:
1. User clicks on "Logout" button	2. Application logs out the user and redirected to home page

Case 4: Email as unique username

Action:	System Response:
1. User enters an email address as username	2. Application will verify if the email address is valid upon clicking "Sign Up" button

Case 5: Change password

Action:	System Response:
1. User clicks on "Forget your password" link	2. Application redirects the user to the page to enter his/her email address (a.k.a. username)
3. User receives an email notification. After clicking on the link in the email, user enters another webpage and enter the new password.	4. Upon clicking, application stores the new password into the database and user can log in with new password

Case 6: Personal Profile

Action:	System Response:
1. User clicks on the profile button	2. Application displays the user profile page with user info, favorite recipes and ingredients, etc.

Case 7: Edit my account information

Action:	System Response:
1. User edits his/her personal account information	2. Application saves the updated information into the database

Case 8: Enter ingredients in profile

Action:	System Response:
1. User enters the ingredients and click on the "submit" button	2. Application saves the ingredients into the database

Case 9: View recommended recipes based on ingredients

Action:	System Response:
1. After entering the ingredients, the user clicks on the recommended recipe tab	2. Application displays the recommended recipe page

Case 10: Detailed recipe page

Action:	System Response:
1. User clicks on a recipe	2. Application displays a detailed page of the particular recipe

Case 11: Back to top button

Action:	System Response:
1. Click "back to top" button	2. The web page slides back to top

Case 12: Email notifications about account setup and password change

Action:	System Response:
1. User sign up for an account or change the password of his/her account	2. Application will notify the user by sending an email notification

Case 13: Comment on recipes

Action:	System Response:
1. User writes a comment on a particular recipe	2. Application will save the comment and display it

Case 14: Reply to other people's comments

Action:	System Response:
1. User writes a reply to other people's comments	2. Application will save the replies and displa

Case 15: Change the avatar picture

Action:	System Response:
1. User click on the original avatar picture	2. Application pops out the local machine file system
3. User chooses the new avatar picture	4. Application uploads the picture to the filesystem on the server
5. User can view his/her new avatar picture	

Case 16: "Like" other people's recipe

Action:
1. User likes other people's recipe

System Response:
2. Application updates the "likes" number of the recipe in the database
3. Updates the number in the client view

Case 17: See how many views each recipe has

Action:
1. User can see how many views each recipe has

System Response:
2. Application show how many views each recipe has in the recipe page

Case 18: Know what ingredients I don't have in stock from each recipe page

Action:
1. User click a recipe link
3. User check what ingredients he or she doesn't have in stock

System Response:
2. Application go to the specific recipe page
4. Application highlights the lacking ingredients

Case 19: Delete the account

Action:
1. User clicks delete on his/her account profile

System Response:
2. Application deletes the user account from the database

Case 20: Edit my account information

Action:
1. User edits his/her personal account information

System Response:
2. Application updates the newly entered information into the database

Case 21: Filter/Sort recipes results based on likes

Action:
1. User click filter/sort button based on the number of likes

System Response:
2. Application gives a dropdown filter option to filter the result based on the number of likes
3. Application would sort the recipes based on the number of likes

Case 22: Filter recipes results based on views

Action:
1. User click filter/sort button based on the number of views

System Response:
2. Application gives a dropdown filter option to filter the result based on the number of views
3. Application would sort the recipes based on the number of views

Case 23: Search recipes with keywords

Action:	System Response:
1. User inputs keyword into search box and hit submit/search	1. Application gives a list of the recipes that match the keyword