

Recipe Shop Test Plan

Change Log			
Version	Change Date	By	Description
1.0.0	2024/10/04	All members	Initial test plan

1.Introduction

1.1 Scope

The scope of testing for sprint 1:

- 1. Search Feature
 - Get a list of recipes based on keyword(required), cooking time(optional) and meal type(optional)
 - View individual recipes in the list
- 2. User Account Management
 - Create account
 - Sign in to account

1.2 Roles and Responsibilities

Name	GitHub Username	Role
Meixuan Chen	njzfjiang	Front End Developer, QA Analyst
Densson Zhu	zhude2	Back End Developer
Troy Thomas	TroyT21	Back End Developer, QA Analyst
Ifeanyi Ochiagha	Francis518	Back End Developer

Role Details:

1. Front End Developer: Developer focusing on creating UI and components.
2. Back End Developer: Developer focusing on creating the logic layer and connecting to APIs and Database
3. QA Analyst: Responsible for creating tests and maintaining the quality of the software when there is an update in the project.

2. Testing Methodology

2.1 Test Levels

Core Feature: Search Feature

User story: As a user, be able to search for recipes based on my preference.

Unit Tests:

1. Make sure the search button is visible on the homepage.
2. Make sure the user can click search to go to the search page.
3. Make sure the user can enter keywords on the textbox input.
4. Make sure the user can enter min cooking time and max cooking time.
5. Make sure the user can select the meal type.
6. Make sure the user cannot submit the query without a keyword specified.
7. Make sure the user can click to view individual recipes.
8. Make sure individual recipes display relevant information.
9. Make sure the server responds with search results for a search request.
10. Make sure the server responds with recipe info for a recipe request.
11. Make sure the server responds with error if searching for non-existent recipes.

Integration Tests:

1. Make sure that clicking the search button sends a GET request to the API.
2. Make sure after a successful GET request(keyword is provided, state is 200), a list of related recipes are shown on the screen.
3. Make sure after an unsuccessful GET request, an error message is shown on the screen.
4. Make sure a loading message shows when the recipes are loading or no query has been submitted.

Acceptance Tests:

1. (Manual) As a user, I am able to search for recipes based on my preference.

Steps:

1. Given I am the role of a guest user(not logged in).
2. Open the Recipe Shop Home page.
3. Click on the "Search now" button.
4. Select "Breakfast" for the meal type.
5. Enter 0 in the left input field for cooking time.
6. Enter 10 in the right input field for cooking time.
7. Enter "Eggs" for the keyword textbox.
8. Click "Search"
9. A list of recipes related to "Eggs" with their respective images show up on the screen, indicating a successful search.

Core Feature: User Account Management

User story: As a user, be able to create account with valid username and password

Unit Tests:

1. Make sure the user can input the username.
2. Make sure the user can input the password.
3. Make sure the Register button can be clicked.
4. Make sure the Sign in button can be clicked.
5. Make sure the form does not submit with empty fields.

Integration Tests:

1. Make sure the user can register with a unique username and correct, matching passwords.
2. Make sure an error message shows if the user is unable to register(error occurred in API request).
3. Make sure if the username is not unique, the registration will fail.

Acceptance Tests:

1. (Manual) As a user, I am able to register with valid username and password

Steps:

1. Given I am a user and I have no account registered for Recipe shop.
2. Open the Recipe Shop Home page.
3. Click on the Register now link.
4. Enter a unique username
5. Enter a valid password(only strings allowed)
6. Enter the same password again in the repeat password field
7. Click Register

8. The Application redirects the user to the login page, signifying a successful register.

User story: As a user, be able to sign in to my account with correct username and password

Unit Tests:

1. Make sure the user can input the username.
2. Make sure the user can input the password.
3. Make sure the Register button can be clicked.
4. Make sure the Sign in button can be clicked.
5. Make sure the user cannot sign in with an empty username or an empty password.

Integration Tests:

1. Make sure user can sign in with correct, matching username and password
2. Make sure the user cannot sign in with a username and password that does not match.

Acceptance Tests:

2. (Manual) As a user, I am able to sign in to my account with correct username and password

Steps:

1. Given I am a user with a registered account for Recipe Shop
2. Open the Recipe Shop homepage.
3. Click the Sign in button on the top left of the navbar
4. Enter the correct username and password
5. Click the Sign in button under the textboxes
6. The application redirects the user to the home page, signifying a successful register.

Regression Tests: All unit tests and integration tests will run before a push or pull request to the main branch.

2.2 Test Completeness

Tests will be completed when:

1. 80% class coverage
2. All automated and manual test cases passed

3 Resource & Environment Needs

3.1 Testing tools

General Tools:

GitHub Actions

GitHub issues for bug tracking

Testing Library:

- Jest
- supertest

Front End:

Developer Tools (Chrome)

Libraries:

- React testing library
- React test renderer
- Axios mock adapter
- Jest

3.2 Testing Environment

Minimum Hardware:

- Windows 10 or Mac OS

Required Software

- Node js 20.+
- Chrome Browser

4 Terms & Acronyms

Term/Acronyms	Definition
API	Application Programming Interface
QA	Quality Assurance
UI	User Interface