CSUN Pantry Pal

Use Cases

Group 6

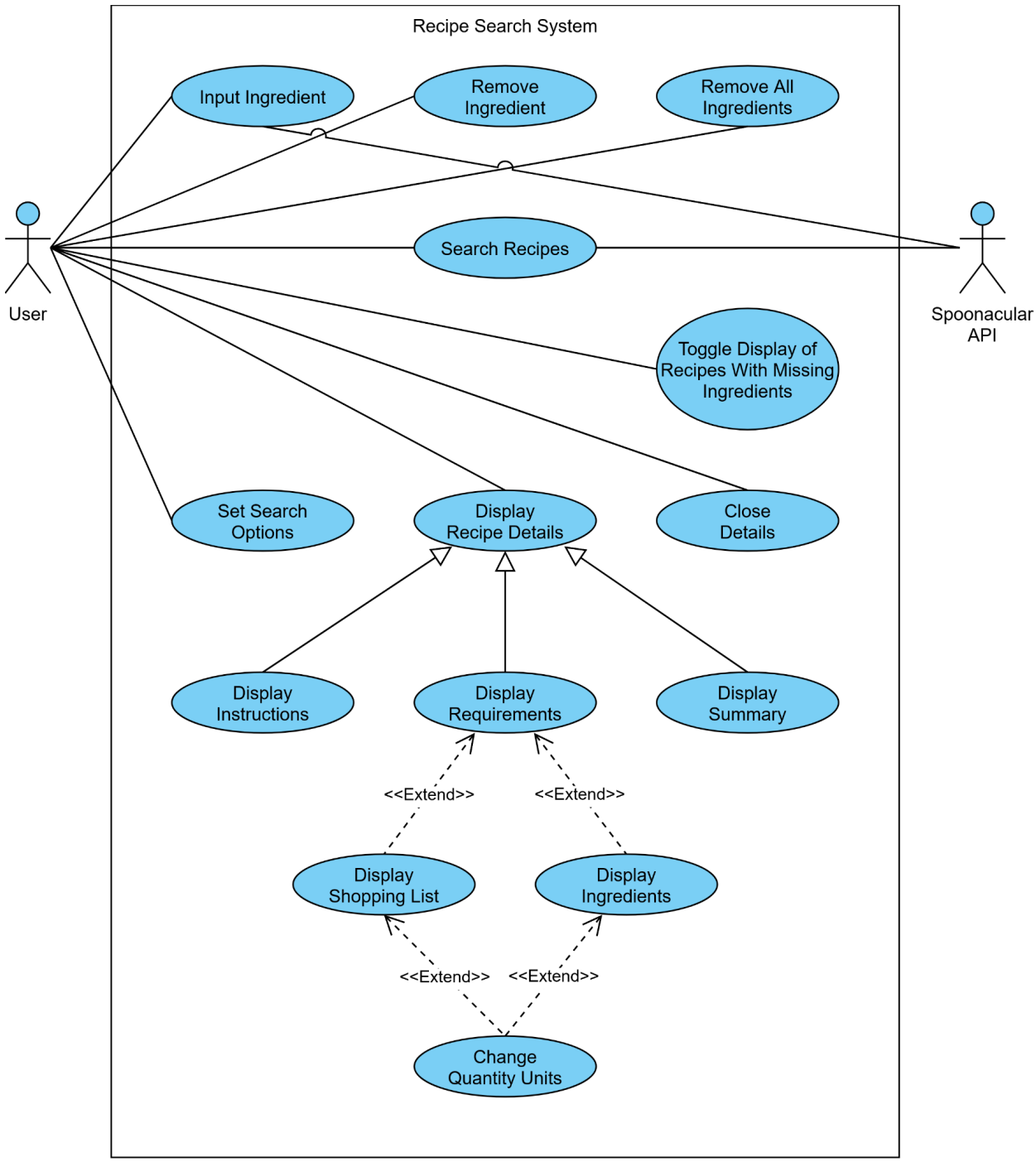
Ziaur Chowdhury
Mark Kozlov
Esteban Maciel
Brian Melgar
Sheran Morais

CSUN | CALIFORNIA STATE UNIVERSITY NORTHRIDGE

Date: 2021-07-09

Table of Contents

1. Use Case Diagram



2. Use Case Descriptions

2.1 Input Ingredient

Description:
A user wants to specify an ingredient they have. By providing owned ingredients, the system will be able to narrow down recipe search results to ones that contain ingredients the user has.

Actors:

- User
- Spoonacular

Triggers
The user starts typing in the ingredient input form.

Pre-conditions
None.

Post-conditions
The submitted ingredient's name will be displayed beside the names of all other submitted ingredients, and the ingredient will be included in subsequent recipe searches.

Main Flow

1. The user types into the text box.
2. The frontend makes a request for autocomplete data from the backed API using the user's input.
3. The backend API validates the request.
4. The backend API forwards the request to the Spoonacular API along with an API authentication token.
5. Spoonacular responds with data.
6. The backend API responds to the frontend's request with Spooonacular's response data.
7. The frontend displays an autocomplete menu with suggestions for the ingredient the user wanted to add.
8. The user selects an ingredient from the menu.
9. The frontend replaces the user's input in the text box with the name of the selected ingredient.
10. The frontend closes the autocomplete menu and restores focus to the text box.

11. The user submits the form by pressing enter on their keyboard or by pressing the submit button.
12. The frontend clears the input text box.
13. The frontend displays the submitted ingredient's name with a remove button.

Alternate Flows

8.1 The user keeps typing into the text box.
    1. Go to step 2 and continue from there.

8.2 There are no autocomplete results to select.
    1. The user deletes some or all of their input.
    2. Go back to step 2 and continue from there.

11.1 The user keeps typing into the text box.
    1. Go to step 2 and continue from there.

Exceptions

3.1 The request is invalid.
    1. The backend API responds to the frontend's request with the appropriate HTTP status code and error message.
    2. The frontend displays an error message.

5.1 Spoonacular responds with an error.
    1. The backend API forwards the error response as a response to the frontend's request.
    2. The frontend displays an error message.

8.1 The user submits the form.
    1. The frontend displays an error message which states that an ingredient from the autocomplete menu must be selected.

12.1 The user submits an ingredient that is already added.
    1. The frontend displays an error message which includes the name of the ingredient and states that it has already been added.

2.2 Remove Ingredient

Description:
A user wants to remove an ingredient they have previously submitted. This may be due to mistakenly entering that ingredient i.e. selected the wrong option or realised they don't actually have that ingredient. Alternatively, they may have decided that they simply aren't interested in recipes with those ingredients. Removing a single ingredient is a convenient way to correct a mistake, compared to resetting the entire form or refreshing the web page.

Actors:

- User

Triggers
The user presses the remove button for an ingredient.

Pre-conditions
The ingredient to be removed is currently added and being displayed (see 2.1).

Post-conditions
The ingredient is removed from display and will no longer be included in recipe searches.

Main Flow

1. The user presses the remove button for an ingredient.
2. The frontend removes the element for the ingredient so that it is no longer displayed.

Alternate Flows
None.

Exceptions
None.

2.3 Remove All Ingredients

Description:
A user wants to remove all submitted ingredients. As with 2.2, this may be due to a mistake, but for many ingredients. However, it is also a convenient way to start a completely new search with different inputs.

Actors:

- User

Triggers
The user presses the reset button.

Pre-conditions
None.

Post-conditions
All previously added ingredients are removed from display and will no longer be included in recipe searches.

Main Flow

1. The user presses the remove button for an ingredient.
2. The frontend removes the elements of all ingredients so that they're no longer displayed.
3. The frontend clears the ingredient input text box.
4. The frontend hides the ingredient autocomplete menu if it is currently being displayed.

Alternate Flows
None.

Exceptions
None.

2.4 Set Search Options

Description:
A user wants to customise the recipe search (2.5). They want to change the way results are sorted or how they're filtered. This helps the user hone in on the recipes they're interested in. It gives them the power to increase the likelihood that they will be shown a recipe that they will want to actually prepare.

Actors:

- User

Triggers
The user presses the button to open the search options menu.

Pre-conditions
None.

Post-conditions
The set values for search options will be used when performing subsequent recipe searches.

Main Flow

1. The user presses the button to open the search options menu.
2. The frontend displays a dropdown menu which contains the following:
    a. A dropdown menu with sorting options, which allows a single selection
    b. A dropdown menu with meal types, which allows a single selection
    c. A dropdown menu with cuisines, which allows multiple selections
    d. A positive integer input box for the maximum preparation time in minutes
3. The user sets any or all values.
4. The user closes the search options menu by clicking outside it.

Alternate Flows
None.

Exceptions
None.

2.5 Search for Recipes

Description:
A user has added ingredients and wants to search for recipes containing those ingredients. This is the main feature of the software. The user can make sense of the various ingredients they have and find recipes they can make from them.

Actors:

- User
- Spoonacular

Triggers
The user presses the button to open the search options menu.

Pre-conditions
At least one ingredient is currently added (see 2.1).

Post-conditions
The search results are displayed.

Main Flow

1. The user presses the search button.
2. The frontend makes a request for recipes from the backend API using the ingredients the user added (2.1) and the search options they set (2.4).
3. The backend API validates the request.
4. The backend API forwards the request to the Spoonacular API along with an API authentication token.
5. Spoonacular responds with data.
6. The backend API responds to the frontend's request with Spooonacular's response data.
7. The frontend clears previous search results and displays the following for each recipe search results:
    a. The recipe's image
    b. The recipe's price per serving
    c. The recipe's preparation time in minutes
    d. The recipe's healthiness score
    e. Upon hovering the image, the recipe's name

Alternate Flows

7.1 Hiding recipes with missing ingredients is toggled on (see 2.6).

1. The frontend skips displaying results which use ingredients the user did not add.
2. The frontend displays the rest of the results as describe in step 7.

7.2 The recipe is missing an image.

1. The frontend displays a placeholder image.
2. The frontend displays the rest of the recipe as describe in step 7.

Exceptions

3.1 The request is invalid.

1. The backend API responds to the frontend's request with the appropriate HTTP status code and error message.
2. The frontend displays an error message.

5.1 Spoonacular responds with an error.

1. The backend API forwards the error response as a response to the frontend's request.
2. The frontend displays an error message.

2.6 Toggle Display of Recipes With Missing Ingredients

Description:
A user wants to show or hide recipe search results (2.5) which use ingredients the user did not add. When toggled on, it hides them and helps them narrow down results to recipes they may be able to make right then and there. When toggled off, it shows all results and gives the user a wider range of choices, albeit requiring them to obtain the missing ingredients somehow.

Actors:

- User

Triggers
The user presses the associated toggle button.

Pre-conditions
None.

Post-conditions
If toggled on, recipe search results with missing ingredients are hidden. If toggled off, all results are displayed. This will persist and apply to the results of any subsequent searches.

Main Flow

1. The user presses the associated toggle button.
2. The frontend hides recipe search results which use missing ingredients, if any.

Alternate Flows

2.1 The button is toggled off.

1. The frontend shows all recipe search results, if any.

Exceptions
None.

2.7 Display Recipe Details

Description:
A user wants to see more information about a recipe in the search results. They may be interested in preparing the recipe and the additional information is critical in helping them achieve that. The system shall open a modal with different pages for the recipe's summary, its requirements, and its instructions.

Actors:

- User

Triggers
The user presses on the recipe's image in the search results.

Pre-conditions
The recipe for which to show details is currently being displayed as a search result (see 2.5).

Post-conditions
A modal is displayed with pages for the recipe's summary, its requirements, and its instructions.

Main Flow

1. The user presses on the recipe's image in the search results.
2. The frontend displays a modal, which contains the following:
    a. The recipe's title in the header
    b. A navigation menu in the footer, which has buttons for the following pages:
        i. The recipe's summary (see 2.8)
        ii. The recipe's requirements (see 2.9)
        iii. The recipe's instructions (see 2.10)
3. The frontend makes the summary page active i.e. displays it and not the other pages.

Alternate Flows

3.1 The recipe pressed is the same as the last recipe that was pressed.
    1. The frontend displays the page that was left active when the modal was most recently closed.

Exceptions
None.

2.8 Display Recipe Summary

Description:
A user wants to see a synopsis of the recipe. This will provide some basic facts about the recipe, including nutrition information, serving size, popularity, and similar recipes. The summary helps the user make a more informed decision on whether they want to prepare the recipe they are viewing.

Actors:

- User

Triggers
The user presses on the page navigation button corresponding to the summary page, or the frontend automatically makes this page active when it opens the modal (see 2.7).

Pre-conditions
The recipe details modal is open (see 2.7).

Post-conditions
The recipe details modal displays the summary page and does not display the other pages in the modal.

Main Flow

1. The user presses on the page navigation button corresponding to the summary page.
2. The frontend hides the previously active page.
3. The frontend shows the summary page, which contains the following:
    a. The recipe's image
    b. A summary for the recipe

Alternate Flows

1.1 The frontend displays the page automatically after the modal is opened.
    1. Proceed to step 2.

3.1 The recipe has no image.
    1. The frontend displays a placeholder image.
    2. The frontend displays the rest of the page as described in step 3.

Exceptions

1.1 The page is already being displayed.
    1. Do nothing.

2.9 Display Recipe Requirements

Description:
A user wants to see what's required to prepare the recipe. The system shall display the required ingredients (see 2.13) and equipment. This is vital information for the user to know in order to be able to successfully prepare the recipe. The ingredient quantities may be displayed in different units (see 2.12), and a shopping list may be displayed instead of all required ingredients (see 2.11).

Actors:

- User

Triggers
The user presses on the page navigation button corresponding to the requirements page, or the frontend automatically makes this page active when it opens the modal (see 2.7).

Pre-conditions
The recipe details modal is open (see 2.7).

Post-conditions
The recipe details modal displays the requirements page and does not display the other pages in the modal.

Main Flow

1. The user presses on the page navigation button corresponding to the requirements page.
2. The frontend hides the previously active page.
3. The frontend shows the summary page, which contains the following:
    a. The recipe's required ingredients (see 2.13)
    b. The recipe's required equipment with the following:
        i. Equipment's image
        ii. Equipment's name

Alternate Flows

1.1 The frontend displays the page automatically after the modal is opened.
    1. Do step 2 as usual.
    2. Do step 3, but if the shopping list (see 2.11) was being shown when the modal was most recently closed, the frontend displays the shopping list instead of the required ingredients.

3.1 The equipment has no image.

1. The frontend displays a placeholder image.
2. The frontend displays the rest of the equipment as described in step 3.

Exceptions

1.1 The page is already being displayed.

1. Do nothing.

2.10 Display Recipe Instructions

Description:
A user wants to see the preparation instructions for the recipe. This is vital information for the user to know in order to be able to successfully prepare the recipe. Each step shall be numbered and shown in sequential order.

Actors:

- User

Triggers
The user presses the page navigation button corresponding to the instructions page, or the frontend automatically makes this page active when it opens the modal (see 2.7).

Pre-conditions
The recipe details modal is open (see 2.7).

Post-conditions
The recipe details modal displays the instructions page and does not display the other pages in the modal.

Main Flow

1. The user presses the page navigation button corresponding to the instructions page.
2. The frontend hides the previously active page.
3. The frontend shows the instructions page, which contains a numbered list of steps for preparing the recipe.

Alternate Flows

1.1 The frontend displays the list automatically after the modal is opened.
    1. Proceed to step 2.

Exceptions

1.1 The page is already being displayed.
    1. Do nothing.

2.11 Display Recipe Shopping List

Description:
A user wants to see a list of ingredients they're missing for this recipe. They are missing an ingredient if they did not add it before they searched for recipes. The system shall provide a succinct list of ingredient names and quantities which the user would have to purchase or otherwise obtain to prepare the recipe.

Actors:

- User

Triggers
The user presses on the shopping list tab button, or the frontend automatically triggers this when it displays the requirements page (see 2.9).

Pre-conditions
The recipe requirements page is being displayed in the modal (see 2.9).

Post-conditions
The requirements page displays an ingredient shopping list instead of all required ingredients.

Main Flow

1. The user presses on the shopping list tab button.
2. The frontend hides all ingredients.
3. The frontend shows a shopping list, which contains a numbered list of ingredient names and quantities (in US units).

Alternate Flows

1.1 The frontend automatically triggers this when it displays the requirements page.
    1. Proceed to step 2.

3.1 The units are set to metric.
    1. The frontend displays the ingredient quantities in metric (see 2.12).

Exceptions

1.1 The shopping list is already being displayed.
    1. Do nothing.

2.12 Change Ingredient Quantity Units

Description:
A user wants to switch their units of measurement between metric and US.

Actors:

- User

Triggers
The user presses the button to switch to metric or US.

Pre-conditions
The recipe requirements page is being displayed in the modal (see 2.9).

Post-conditions
The ingredient quantities are displayed in the chosen units.

Main Flow

1. The user presses the US button.
2. The frontend displays the ingredient quantities in US units.

Alternate Flows

1.1 The user presses the metric button.

1. The frontend displays the ingredient quantities in metric.

Exceptions

1.1 Desired units are already displayed.

1. Do nothing.

2.13 Display Required Ingredients

Description:
A user wants to see all ingredients required to prepare a recipe. The system shall display the ingredients' name, quantity, and image. This is critical information for the user to be able to prepare the recipe.

Actors:

- User

Triggers
The user presses the "all" tab button, or the frontend automatically triggers this when it displays the requirements page (see 2.9).

Pre-conditions
The recipe requirements page is being displayed in the modal (see 2.9).

Post-conditions
The requirements page displays all required ingredients instead of an ingredient shopping list.

Main Flow

1. The user presses the "all" tab button.
2. The frontend hides the shopping list.
3. The frontend displays the following for each required ingredient:
    a. Ingredient's image
    b. Ingredient's quantity in US units
    c. Ingredient's name

Alternate Flows

1.1 The frontend automatically triggers this when it displays the requirements page.
    2. Proceed to step 2.

3.1 The units are set to metric.
    1. The frontend displays the ingredient quantities in metric (see 2.12).
    2. The frontend displays the rest of the requirements as described in step 3.

3.2 The ingredient or equipment has no image.
    1. The frontend displays a placeholder image.
    2. The frontend displays the rest of the ingredient or equipment as described in step 3.

Exceptions

1.1 All required ingredients are already being displayed.

2. Do nothing.

2.14 Close Recipe Details

Description:
A user wants to close the recipe details. The system shall remove the modal from display and allow the user to interact with the main page with the search results. This lets the user start a new search or view the details of a different recipe.

Actors:

- User

Triggers
The user presses the close button in the modal.

Pre-conditions
The recipe details modal is open (see 2.7).

Post-conditions
The modal is closed and the user can interact with the main page.

Main Flow

1. The user presses the close button in the modal.
2. The frontend closes the modal.

Alternate Flows

1.1 The user clicks outside the modal.

1. The frontend closes the model.

Exceptions
None.