Food Pants

Use Cases

ID UC 1.1 Manage Pantry
Scope Pantry system
Level User goal
Stakeholders and interests

- **User:** Person interested in managing a digital pantry. Wants the ability to view food items currently in the pantry. Wants the ability to add food items to their pantry. Wants the ability to modify food items in their pantry. Wants the ability to remove food items from their pantry.
- **System maintainer:** Person responsible for application execution. Wants to satisfy customer interests.

Precondition: User has gone through setup process.
Postcondition: Pantry is updated with any user modifications.

Main success scenario:

1. User wants to manage pantry.
2. User navigates to the "Pantry" page.
3. System displays a list of all food items in the pantry.
4. User presses the "Add Item" button.
5. User enters the name of the food type to add to pantry in a search box.
6. System displays food type search results in a drop-down menu.
7. User selects the food type to add.
8. User enters the quantity/amount of the food item to add to pantry.
9. System adds food item to pantry.

User repeats steps 4-9 until satisfied with the pantry

Extensions:

a.* Anytime the user decides to stop managing the pantry
    1. *User presses the back button*
        a. *User has unsaved changes*
            i. *System prompts user to save their changes*
            ii. *User presses "OK"*
            iii. *System redirects user to the a pantry overview page where all food items in the pantry are displayed*

4.a User wants to completely remove a food item from the pantry
    1. *User selects trash icon next to food item*
    2. *System prompts the user for confirmation*
    3. *a. User presses the "Yes" button*
        i. *System deletes the food item from the pantry*
        b. User presses the "No" button

4.b User wants to change the quantity/amount of an existing food item

1. *User selects the pencil button next to a food item to edit it*
2. *User selects the quantity/amount text box*
3. *User enters a new quantity/amount*

4.c User wants to consume an item in the pantry

1. *User selects the "Consume" option next to a food item*
2. *extend <consumePantryItem>*

4.d User wants to search for an item

1. *extend <searchPantry>*

6.a No results meaning food type is not registered in the food type database

1. *User selects the "Add food type" option from the end of the drop-down menu*
2. *extend <addFoodType>*

9.a Food item already exists in pantry

1. *System adds quantity of food item user entered to quantity of food item already in the pantry*

ID UC 1.2 Search Pantry
Scope Pantry system
Level User goal
Stakeholders and interests

- **User:** Person interested in searching the pantry. Wants the ability to find a food item in the pantry.
- **System maintainer:** Person responsible for application execution. Wants to satisfy customer interests.

Precondition: User has gone through setup process.
Postcondition: Food items in the pantry are displayed based on user search criteria.

Main success scenario:

1. User wants to search the pantry.
2. User navigates to the "Pantry" page to view the pantry.
3. User navigates to the search bar along the top.
4. User types in the search bar.
5. System displays food items in the pantry in a list with names that contain the user's entered search keyword.

User repeats steps 4-5 until done searching the pantry

Extensions:

a.* Anytime the user decides to stop searching the pantry
    1. *User presses the back button*
    2. *System returns full list of food items in the pantry*

5.a Pantry is empty
    1. *System tells users that there are no food items in the pantry*
    2. *System prompts user if they would like to add a food item to the pantry*
    3. *User clicks "Add Item" button*
    4. *extend <managePantry>*

5.b No food items in pantry contain user search criteria in name
    1. *System tells the user no results were found*
    2. *System prompts user if they would like to add a food item to the pantry*
    3. *User clicks "Add Item" button*
    4. *extend <managePantry>*

ID UC 1.3 Consume Pantry Item
Scope Pantry system
Level User goal
Stakeholders and interests

- **User:** Person interested in consuming an item from the pantry. Wants the ability to decrement the quantity of a food item in the pantry.
- **System maintainer:** Person responsible for application execution. Wants to satisfy customer interests.

Precondition: Food types to purchase are registered in the food type database.
Postcondition: Shopping list modifications are saved.

Main success scenario:

1. User wants to consume a pantry item.
2. User navigates to the “Pantry” page to view the pantry.
3. User navigates to food item they want to consume.
4. User clicks “Consume” option.
5. System decrements quantity of food item on pantry.

User repeats steps 3-5 until consumed all food items they wish to

Extensions:

a.* Anytime the user decides to stop consuming items from the pantry
    1. *User presses the back button*
    2. *System returns full list of food items in the pantry*

5.a Food item is decremented to zero
    1. *System removes food item from the pantry*

ID UC 2.1 Manage Nutritional log
Scope Nutrition Log system
Level User goal
Stakeholders and Interests

- **User:** A person utilizing the Nutrition Log system within the FoodPants application to view, add, remove, or edit an item.
- **System maintainer:** Person responsible for application execution.

Precondition: FoodPants is installed as an executable on the user's machine and the user is currently in their nutrition log.
Postcondition: An item within the pantry has been viewed, added, edited, or removed.

Main success scenario:

1. User wants to manage their nutrition log
2. User wants to view the details of a specific item
3. User scrolls through the months view until the month containing the desired item is displayed
4. User clicks on the month containing the desired item and the system displays the month broken up into weeks within the week view
5. User scrolls through the week view until the week containing the desired item is displayed
6. User clicks on the week containing the desired item and the system displays the week broken up into days within the day view
7. User scrolls through the day view until the day containing the desired item is displayed
8. User clicks on the day containing the desired item and the system displays the day broken up into hours within the hour view
9. User scrolls through the hour view until the hour containing the desired item is displayed
10. User clicks on the item they want to view

User repeats steps 2-10 until satisfied with the management of their nutrition log

Extensions:

a.* Anytime the user wants to quit the process

1. *User can close the current menu by hitting the X button*

User repeats this step until they have gotten all the way out of the specified process

2. a If the user wants to manually add an item to the nutrition log

1. *User presses the "+" Add Item button*
2. *User enters the name of the Item to add in a search box.*
3. *System displays search results in a drop-down menu.*
4. *a. User selects the item they want to add.*
   b. If the food type is not registered in the food type database

i. User selects the "Add food type" option from the end of the drop-down menu

ii. extend <addFoodType>

5. *User enters the quantity they consumed.*
   a. The user can change the timestamp for consumption if it is not the current time.
6. *User selects create item*

10. a If the User wants to modify the existing item in the pantry

1. *User clicks on the 'Edit Item' button within the Item Details interface*
2. *System provides a form with the current Item values which can be changed*
3. *User alters the desired values within the form which include the timestamp for consumption and nutritional information (calories, carbs, protein, fat, cholesterol, sodium)*
4. *a. User hits the 'Confirm' button*
   b. User hits an 'X' which exits without saving changes
5. *System updates Item information*

10. b If the User wants to remove the existing item from the nutrition log

1. *User clicks on the 'Delete Item' button within the Item Details interface*
2. *System asks the user to confirm that they want to delete the item*
3. *User clicks 'Confirm'*
4. *System removes Item from the nutrition log*

ID UC 2.2 Manage Nutrition Goals
Scope Nutrition Log system
Level User goal
Stakeholders and Interests

- **User:** A person utilizing the Nutrition Log system within the FoodPants application to set and manage nutritional goals.
- **System maintainer:** Person responsible for application execution.

Precondition: FoodPants is installed as an executable on the user's machine and the user is currently in their nutrition log.
Postcondition: A nutrition goal has been added, edited, or removed.

Main success scenario:

1. User wants to manage their nutrition goals
2. User selects the nutrition goals button to navigate to the nutrition goals menu
3. User wants to set a nutrition goal
4. User selects the "+" button and the set nutrition goal menu is displayed
5. User selects the nutrition goal field (calories, carbs, fat, protein, item, cholesterol, sodium)
6. User enters the desired goal value
7. User selects whether the goal is a minimum or a maximum
8. User checks whether the want an alert when nearing their goal
9. User selects time period for time based alerts on goal progress
10. User selects create goal and confirms the creation at the systems prompting

User repeats steps 3-10 until satisfied with the management of their nutrition goals

Extensions:

a.* Anytime the user wants to quit the process

1. *User can close the current menu by hitting the X button*

User repeats this step until they have gotten all the way out of the specified process

3. a If the User wants to modify a pre-existing nutrition goal

1. *User clicks on the edit goal button within the nutritional goals interface*
2. *System displays form with the current goal values which can be changed*
3. *User alters the desired values within the form*
   a. User hits the 'Confirm' button
   b. User hits an 'X' which exits without saving changes
4. *System updates goal information*

3. b If the User wants to remove a pre-existing goal from the nutrition log

1. *User clicks on the edit goal button within the nutritional goals interface*
2. *System displays form with the current goal values*

3. *User clicks on the delete goal button within the edit goal interface*
4. *System asks the user to confirm that they want to delete the goal*
5. *User clicks 'Confirm'*
6. *System removes goal from the nutrition goals menu*

ID UC 2.3 Manage Nutrition Report
Scope Nutrition Log system
Level User goal
Stakeholders and Interests

- **User:** A person utilizing the Nutrition Log system within the FoodPants application to view, create, modify, or remove a nutrition report.
- **System maintainer:** Person responsible for application execution.

Precondition: FoodPants is installed as an executable on the user's machine and the user is currently in their nutrition log.
Postcondition: A nutrition report has been viewed, added, modified, or removed

Main success scenario:

1. User wants to manage nutrition reports
2. User selects the nutrition reports button to navigate to the nutrition reports menu
3. User wants to create a nutrition report
4. User selects the "+" button and the create nutrition report menu is displayed
5. User selects the pie chart option
6. User selects the total nutrition option
7. User selects the desired time period that the chart should show
8. User clicks the confirm report changes button and confirms when prompted by the system

User repeats steps 3-10 until satisfied with the management of their nutrition goals

Extensions:

a.* Anytime the user wants to quit the process

1. *User can close the current menu by hitting the X button*

User repeats this step until they have gotten all the way out of the specified process

3. a If the User wants to modify a pre-existing nutrition report

1. *User clicks on the edit report button within the specified report*
2. *System displays form with the current report specifications which can be changed*
3. *User alters the desired values within the form*
   a. User hits the 'Confirm' button
   b. User hits an 'X' which exits without saving changes
4. *System updates report information*

3. b If the User wants to remove a pre-existing report from the nutrition report menu

1. *User clicks on the edit report button within the specified report*
2. *System displays form with current report specifications*
3. *User clicks on the delete report button within the edit report interface*
4. *System asks the user to confirm that they want to delete the report*

5. *User clicks 'Confirm'*
6. *System removes report from the nutrition reports menu*

3. c If the User wants to view previously generated reports
    1. *The user can should avoid the remaining steps and view the currently generated reports*

5. a If the user wants the bar chart option instead
    1. *User selects the bar chart option*
    2. *User selects the field to be on the y axis*
    3. *User selects the time period to be on the x axis*
    4. *User clicks the confirm report changes button and confirms when prompted by the system*

5. b If the user wants the table option instead
    1. *User selects the table option*
    2. *User selects the time period which the data should come from*
    3. *a User selects field they want in the table*
       b If the user wants to remove an added field they can select it to be removed
    4. *User clicks the confirm report changes button and confirms when prompted by the system*

User repeats steps 5. b-3 until the desired number of fields have been added

6. a If the user wants the chart to display data about a specific field instead
    1. *User selects the field they want the chart to describe*
    2. *User adds the items they want to be in the chart to it*

User repeats steps 6. a-2 until the desired number of items have been added

ID UC 3.1
Scope Recipe creation system.
Level User goal
Stakeholders and interests

- **Project Leader:** Person responsible for overseeing system goals.
- **Project Member:** Person responsible for implementing and maintaining the system.
- **System:** Person responsible for recipe creation execution.

Precondition: The user can successfully execute the application.
Postcondition: Recipe successfully created.

Main success scenario:

1. User clicks on “Recipes” from sidebar menu
2. System displays recipe page with recipes list and recommended recipes
3. User wants to create a recipe manually
4. System provides a recipe creation form
5. User enters recipe name into form
6. User enters list of ingredients into form
7. User enters recipe servings into form
8. User submits recipe form
9. System verifies form has required information
10. New recipe is added into recipes list

Extensions:

a.a User exits form abruptly
    1. *recipe creation will be canceled.*

a.b User exits page abruptly
    1. *Recipe creation will be canceled*

a.c User exits application abruptly
    1. *Recipe creation will be canceled*

b.* System encounters an unexpected error
    1. *Recipe creation will be canceled*
    2. *Page will be restarted*

1.a If there are no recipes yet created
    1. *System will not display a recipes list*

9.a User does not enter a recipe name into form
    1. *System will ask user for recipe name*

9.b If user does not enter at least one ingredient into form
    1. *System will ask user for at least one ingredient*

9.c If user does not enter number of servings into form

1. *System will ask user for number of servings*

ID UC 3.2
Scope View recipe system
Level User goal
Stakeholders and interests

- **Project Leader:** Person responsible for overseeing system goals.
- **Project Member:** Person responsible for implementing and maintaining the system.
- **System:** Person responsible for recipe viewing execution.

Precondition: User can successfully execute the application. User has created at least one recipe.
Postcondition: Recipes successfully viewed.

Main success scenario:

1. User selects “Recipes” from sidebar menu
2. System displays list of recipes to user
3. User wants to search for specific recipe
4. User can search for recipe by recipe name
5. System displays recipe search results
6. User views results from search
7. User clicks on a specific recipe
8. System displays information from recipe

Extensions:

a.a User exits page abruptly
    1. *Recipes are no longer displayed to the user*

a.b System encounters an unexpected error
    1. *System will exit recipe page*
    2. *System will store the error internally*

2.a User has no recipes
    1. *System will only display recommended recipes in list*

3.a If no matching recipes from search
    1. *System will display a blank page to the user*

4.a User searches for recipe based on ingredients
    1. *System displays recipes based on matching ingredients*

8.a User wants to modify the information in the recipe
    1. *System allows input to modify the recipe*

ID UC 3.3
Scope Recipe to shopping list system
Level User goal
Stakeholders and interests

- **Project Leader:** Person responsible for overseeing system goals.
- **Project Member:** Person responsible for implementing and maintaining the system.
- **System:** Person responsible for recipe viewing execution.

Precondition: A user can successfully view recipes list.
Postcondition: Recipe successfully added to shopping list

Main success scenario:

1. User clicks on “Recipes” from sidebar menu
2. System displays stored recipes to user
3. User views recipes
4. User searches for specific recipe
5. User wants to add recipe to shopping list
6. System adds recipe ingredients to shopping list

Extensions:

a.a User exits page abruptly
    1. *Recipes are no longer displayed to the user*

a.b System encounters an unexpected error
    1. *System will exit recipe page*
    2. *System will store the error internally*

2.a If user has no recipes
    1. *System will only display recommended recipes in recipe list*
    2. *System will ask user to create a recipe or choose from recommended*

4.a If no recipes are found in search
    1. *System will display a blank page*

6.a Recipe ingredients already exist in shopping list
    1. *System will add the ingredients again*

ID UC 3.4 Modify Recipe
Scope Recipe System
Level User goal
Stakeholders and Interests

- **User:** A person utilizing the Recipe System within the FoodPants application to modify existing recipes.
- **System maintainer:** Person responsible for application execution.

Precondition: Recipes exist within the Recipe System so there is something to modify.
Postcondition: A recipe has been edited.

Main success scenario:

1. User wants to modify a recipe.
2. User navigates to the recipe page.
3. User selects recipe to be modified.
4. User selects the "Edit Recipe" button and the recipe specifications are displayed.
5. User edits food type and amount fields and/or the recipe instruction field.
6. User selects the "Save Changes" button to save edited fields.
7. The system replaces the old recipe with the user-edited recipe.
8. The user is returned to the recipe view page with the newly modified recipe listed.

Extensions:

3.a User has no recipes
    1. *User selected the "+" button.*
    2. *Recipe creation page is displayed to the user.*

5.a The user wants to add a new food type
    1. *User selects the "+" button in the food type field.*
    2. *New food type page is displayed*
    3. *User fills required fields.*
    4. *User selects the "Create Item" button.*
    5. *Recipe modification page is displayed with newly created food type added to the food type field.*

6.a The user wants to exit the recipe modification page without saving modifications
    1. *The user selects the "Discard Changes" button.*
    2. *The user is returned to the recipe view page.*

6.b The user wants to delete the recipe.
    1. *The user selects the "Delete Recipe" button.*
    2. *The user is returned to the recipe view page and the recipe is no longer listed.*

ID UC 3.5 Recommend Recipes
Scope Recipe System and Digital Pantry
Level User goal
Stakeholders and Interests

- **User:** A person utilizing the Recipe System within the FoodPants application to view recommended recipes.
- **System maintainer:** Person responsible for application execution.

Precondition: Recipes exist within the Recipe System and the user has items in their pantry.
Postcondition: Recommended recipes are displayed to the user for further action to be taken.

Main success scenario:

1. User wants to view recommended recipes based on what they already have in their pantry.
2. User navigates to the recipe page.
3. User selects the "View Recommendations" button.
4. Recommended recipes retrieved from the Recipe System are displayed
5. The user selects a recipe to view

Extensions:

2.a User exits the recipe viewing page
    1. *User selects the "Back" button.*
    2. *Home page is displayed to the user.*

3.a The user wants to stop viewing recommended recipes
    1. *User selects the "View Recommendations" button, removing its highlight*
    2. *Default recipe page is displayed*

ID UC 3.6 Cook Recipe
Scope Recipe system
Level User goal
Stakeholders and Interests

- **User:** Person who has stored recipes and items within the FoodPants digital pantry, wants to use stored items to make a recipe and log the results to their nutrition log.
- **System maintainer:** Person responsible for application execution. Wants to satisfy customer interests.

Precondition: The user has created the recipe within the recipe system, and has decided how much of the recipe they will eat (they can perform this step after cooking if necessary)
Postcondition: Used items will have been consumed from the user's pantry, and the recipe and nutritional info will be stored in the user's nutrition log.

Main success scenario:

1. The user navigates to the recipe page.
2. User selects desired recipe to produce.
3. User selects to cook the recipe.
4. The system confirms the user's consumption of the recipe.
5. The system removes items from the digital pantry that were used to make the recipe.
6. The User is prompted for how much of the recipe they consumed.
7. The amount of the recipe consumed is added to the nutrition log.
8. include (manageNutritionLog)
9. The User is prompted to enter how much of the recipe is leftover.
10. The system stores leftovers as part of the User's digital pantry.
11. include (managePantry)
12. The user is returned to the recipe view menu.

Extensions:

a. * The desired recipe does not exist
    1. *The System will not allow the user to cook the recipe.*
    2. *The user can add the recipe to the system.*

4. a The user does not have all the required items to cook the recipe.
    1. *The system prompts the user if they want to cook the recipe anyways.*
    2. *a. User confirms to cook the recipe anyways.*
        i. *User selects to consume only existing items from the pantry.*
        ii. *User selects to consume no items from the pantry.*
    b. User requests desired items to be added to the shopping list.
        i. *System adds items to the shopping list (include addRecipeItemsToShoppingList).*

ii. *User is redirected to the shopping list menu, preparing the recipe is aborted.*

6. a The user has not set up serving sizes for this recipe
    1. *The system prompts the user to dictate how many servings the recipe makes*
    2. *The user enters how many servings were consumed.*

9. a The user has no food remaining or does not wish to record leftovers
    1. *The user can press an option to skip this step.*

Special Requirements

- Users can utilize this function even if they are missing ingredients.
- This will interface nicely with the shopping list feature so that users can automatically fill in missing items.

ID UC 4.1 Manage Shopping List
Scope Shopping List system
Level User goal
Stakeholders and interests

- **User:** Person interested in managing a shopping list. Wants the ability to modify their shopping list. Wants the ability to add items to their shopping list based on a recipe. Wants an easily visible display of their shopping list.
- **System maintainer:** Person responsible for application execution. Wants to satisfy customer interests.

Precondition: Food types to purchase are registered in the food type database.
Postcondition: Shopping list modifications are saved.

Main success scenario:

1. User wants to manage a shopping list.
2. User navigates to the "Shopping List" page to view the list.
3. User navigates to the "Modify Shopping List" option.
4. User presses the "Add Item" button.
5. User enters the name of the food type to purchase in a search box.
6. System displays food type search results in a drop-down menu.
7. User selects the food type to purchase.
8. User enters the quantity/amount of the food item to purchase.

User repeats steps 4-8 until satisfied with the shopping list

9. User presses the "Save" button.

Extensions:

a.* Anytime the user decides to stop managing the shopping list
    1. *User presses the back button*
        a. *User has unsaved changes*
            i. *System prompts user to save their changes*
            ii. *User presses "OK"*
            iii. *System redirects user to the "Modify Shopping List" page*

3.a User wants to add items to their list from a recipe instead of manually
    1. *User navigates to the "Add Items From Recipe" option*
    2. *extend <addRecipeItemsToShoppingList>*

4.a User presses the "Delete Item" button to delete an existing item
    1. *System prompts the user for confirmation*
    2. *a. User presses the "Yes" button*
        i. *System deletes the item*
        b. User presses the "No" button

4.b User wants to change the quantity/amount to purchase for an existing item

1. *User selects the quantity/amount text box*
2. *User enters a new quantity/amount*

7.a Food type is not registered in the food type database

1. *User selects the "Add food type" option from the end of the drop-down menu*
2. *extend <addFoodType>*

9.a User wants to discard their changes

1. *User presses the "Cancel" button*
2. *User leaves "Modify" mode and is redirected to the main "Shopping List" page*

9.b User wants to export the shopping list

1. *extend <exportShoppingList>*

ID UC 4.2 Export Shopping List
Scope Shopping List system
Level User goal
Stakeholders and Interests

- **User:** Person that is interested in creating a grocery list of their items and exporting it from the application.
- **System maintainer:** Person responsible for application execution.

Precondition: Shopping list is created, desired items have been added
Postcondition: A shopping list is added to shopping lists and can be viewed from that menu.

Main success scenario:

1. The user navigates to their shopping list.
2. The user wants to export their shopping list.
3. The user selects to export the shopping list.
4. The system prompts the user on how they would like to export the list.
5. The user selects that they would like to print the list.
6. The List is shown as a print preview.
7. The user can print the list.
8. The System saves the list for view in the list menu.

Extensions:

a. * If the list is empty
    1. *The system will deny the user request, telling them to add items*

3. a If the user changes their mind,
    1. *The system will return the user to the main shopping list window.*

5. a The user wants to save the list as a PDF.
    1. *The system will export the list to a PDF.*
    2. *The user can download the file.*

5. b The user wants to save the list as a text file.
    1. *The system will export the list to a text file.*
    2. *The user can download the file.*

5. c User does not want to export the list
    1. *The system will proceed to save the list for view in the list section.*

ID UC 4.3 Mark Purchased Items
Scope Shopping List system
Level User goal
Stakeholders and interests

- **User:** Person interested in shopping list. Wants the ability to cross out purchased items from their shopping list (one by one or all items at once).
- **System maintainer:** Person responsible for application execution. Wants to satisfy customer interests.

Precondition: User had previously added items to the shopping list. User purchased all items from their shopping list or a subset of the items from their shopping list.
Postcondition: Items marked as purchased are removed from the shopping list. Items marked as purchased are added to the pantry.

Main success scenario:

1. User wants to mark items from their shopping list as purchased.
2. User navigates to the "Shopping List" page to view the list.
3. User presses the "Mark All as Purchased" button.
4. System prompts user for confirmation.
5. User presses the "Yes" button.
6. System displays an info message.

Extensions:

3.a User wants to mark an individual item as complete
    1. *User presses the "Mark Item as Purchased" button for the given item*
    2. *System displays an info message with an "Undo" option*
        a. User presses the "Undo" button to unmark the item as purchased

5.a User no longer wishes to mark all items as purchased
    1. *User presses* the "No" button

ID UC 5.1 Startup
Scope Startup
Level Summary
Stakeholders and Interests

- **New User:** New person utilizing the FoodPants application for the first time, wants to become familiarized with the application.
- **System maintainer:** Person responsible for application execution.

Precondition: FoodPants is installed as an executable on the user's machine and running.
Postcondition: New User has a profile created, and has been familiarized with the system.

Main success scenario:

1. The system greets the new user
2. The system prompts the new user for their details.
3. The user enters details such as name, gender, height, and weight for calorie info.
4. The system saves the user data and configures the new profile.
5. The user is shown the digital pantry.
6. The user is prompted to enter items they may already have in their pantry.
7. include (addFoodToPantry)

Continue step 7 until the user decides they are satisfied with their pantry.

8. The user is shown the recipe menu and suggested a recipe to add
9. include (getNewReccomendedRecipies)
10. The user is shown the shopping list and prompted to add an item of their choice
11. include (manageShoppingList)
12. The system thanks the user for using the walkthrough
13. The user is directed to documentation for further questions.
14. The tutorial is finished and the user is redirected to the home window.

Extensions:

a. * The user does not wish to continue the tutorial
    1. *The user can click a button to exit the tutorial*
    2. *The system will resume processes from the home menu*

3. a If the user does not wish to enter gender, height, or weight.
    1. *These fields are optional, some nutrition features will be disabled.*

6. b User does not wish to set up the digital pantry
    1. *The user can press an option to skip this step for now.*

8. a The user does not want to add the suggested recipe
    1. *The user can press an option to skip this step.*

10. a The user does not want to add an item to their list
    1. *The user can press an option to skip this step.*
13. a The user does not want to view the documentation
    1. *The user can deny this option to finish the tutorial.*

Special Requirements

- Provide startup as an embedded walkthrough within the application.

ID UC 6.1 Manage Food Types
Scope Food Type system
Level User goal
Stakeholders and interests

- **User:** Person interested in managing the food types. Wants to customize their available food types for tracking within their pantry, shopping list, recipes, and nutrition log.
- **System maintainer:** Person responsible for application execution. Wants to satisfy customer interests.

Precondition: Basic food types are predefined (but still modifiable).
Postcondition: Food type modifications are saved.

Main success scenario:

1. User wants to manage their food types.
2. User navigates to the "Food Types" page to view the food types.
3. User navigates to the "Modify Food Types" option.
4. User presses the "Add Food Type" button.
5. extend <addFoodType>

User repeats steps 4-5 until satisfied with the list of food types

6. User presses the "Save" button.

Extensions:

a.* Anytime the user decides to stop managing the list of food types
    1. *User presses the back button*
        a. User has unsaved changes
            i. *System prompts user to save their changes*
            ii. *User presses "OK"*
            iii. *System redirects user to the "Modify Shopping List" page*

4.a User presses the "Delete Food Type" button to delete an existing food type
    1. *System prompts the user for confirmation*
    2. *a. User presses the "Yes" button*
        i. *System deletes items in pantry of the given food type*
        ii. *System deletes items in shopping list of the given food type*
        iii. *System removes the given food type from the list of food types displayed*
        b. User presses the "No" button

4.b User wants to edit an existing food type name
    1. *User selects the food type name text box*
    2. *User enters a new name*
        b. Name is blank
            i. *System prevents the user from clicking out of the text box*

ii. System prompts the user to use a non-empty name

iii. Return to step 2

c. Name is already used by another food type

i. System prevents the user from clicking out of the text box

ii. System prompts the user to use a name that is not taken

iii. Return to step 2

4.b User wants to edit an existing food type's details

3. User selects the food type name text box

4. User enters new details

9.a User wants to discard their changes

1. User presses the "Cancel" button

2. System exits "Modify" mode and redirects user to the main "Food Types" page

ID UC 6.2 Create New Food Types
Scope Food Database and Digital Pantry
Level User goal
Stakeholders and Interests

- **User:** A person utilizing the Recipe System within the FoodPants application to create a new food type to store in their digital pantry.
- **System maintainer:** Person responsible for application execution.

Precondition: User wishes to create a new food type.
Postcondition: New food type is created and is selectable by the user.

Main success scenario:

1. User wants to manage their food types.
2. User navigates to the "Food Types" page to view the food types.
3. User navigates to the "Modify Food Types" option.
4. User presses the "Add Food Type" button.
5. User is directed to the new food type form.
6. User enters required information to complete a new food type.
7. User selects the "Create Food Type" button.
8. The Food Types page is displayed to the user.

User repeats steps 4-5 until satisfied with the list of food types

9. User presses the "Save" button.

Extensions:

a.* The user no longer wants to manage food types.
    1. *User presses the "Back" button*
        a. *User has unsaved changes*
            i. *System prompts user to save their changes*
            ii. *User presses "OK"*
            iii. *System redirects user to the "Modify Shopping List" page*

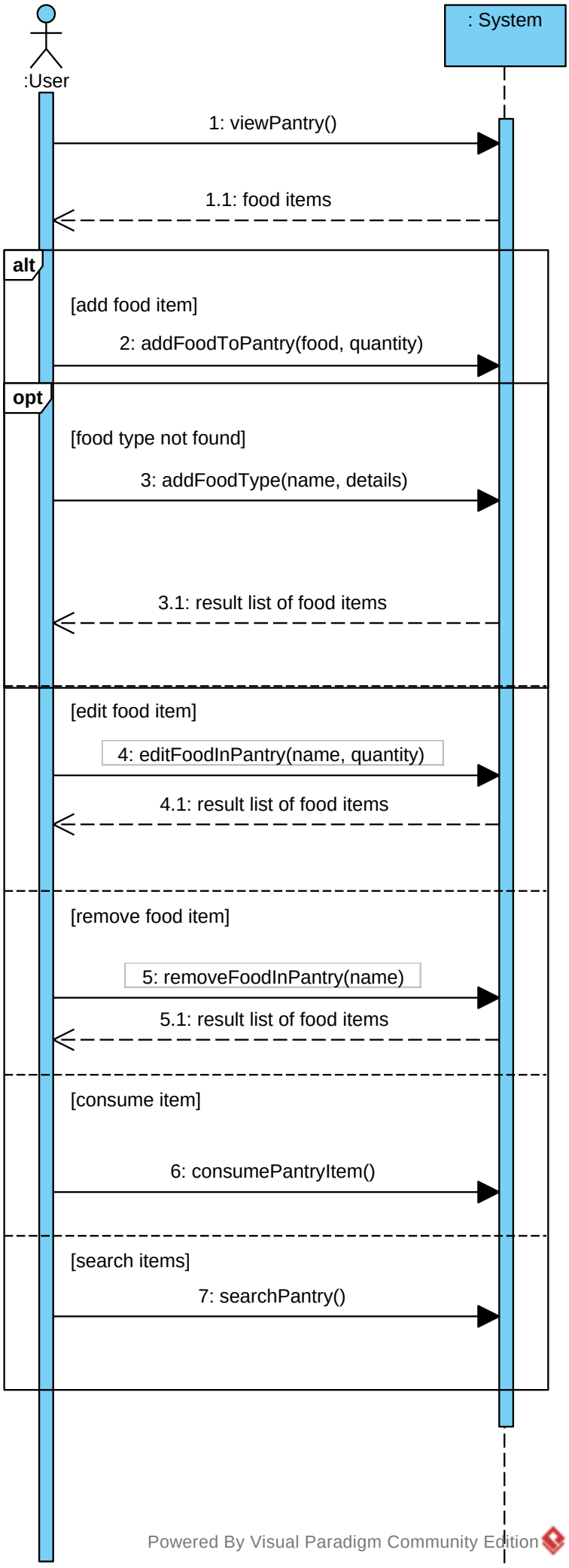
3.a User enters an existing name in the "Food Name" text box
    1. *The system prompts the user to change the name and prevents the user from exiting the text box.*
    2. *Return to step 3 of the main success scenario.*

3.b User leaves a required text box empty
    1. *User prompts user to fill out required text fields.*
    2. *Return to step 7 of main success scenario.*

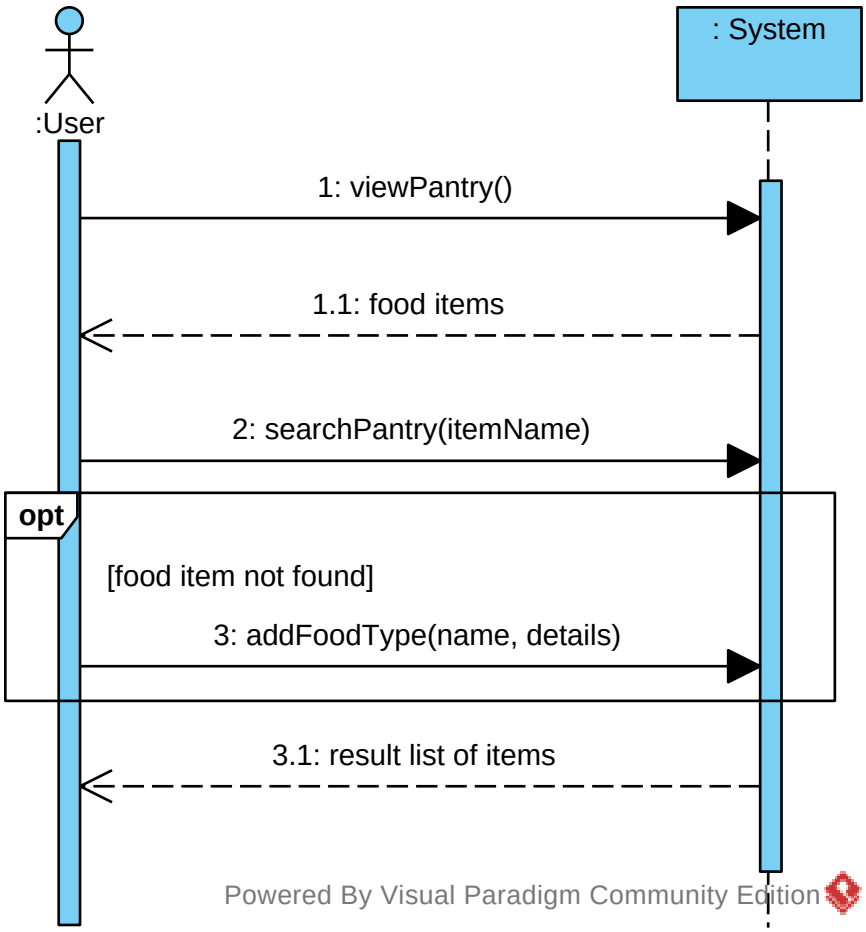
Traceability Matrix

Use Case Name	Use Case ID	REQ 1	REQ 2	REQ 3	REQ 4	REQ 5	REQ 6	REQ 7	REQ 8	REQ 9	REQ 10
Manage Pantry	1.1	X						X	X		X
Search Pantry	1.2	X							X		X
Consume Pantry Item	1.3	X							X		X
Manage Nutrition Log	2.1		X					X	X		X
Nutrition Goals	2.2		X						X	X	X
Nutritional Report	2.3		X						X		X
Create Recipe	3.1			X			X	X	X	X	
View Recipes	3.2			X				X	X		
Add Recipe Items to Shopping List	3.3			X			X		X		X
Modify Recipe	3.4			X					X	X	
Recommend Recipe	3.5			X				X	X		X
Cook Recipe	3.6	X		X					X		X
Manage Shopping List	4.1				X		X	X	X		X
Export Shopping List	4.2				X				X		X
Mark Purchased Items	4.3				X				X		X
Startup Tutorial	5.1	X	X	X	X	X	X	X	X	X	X
Manage Food Types	6.1						X	X	X	X	
Create New Food Type	6.2						X			X	

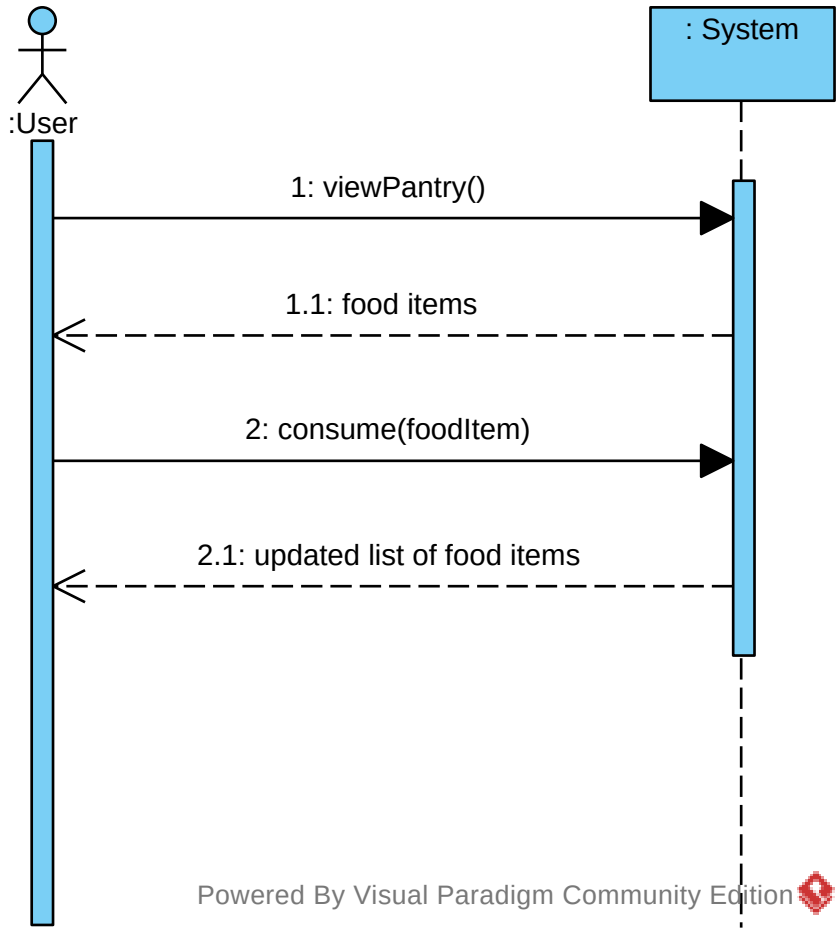
1.1 Manage Pantry



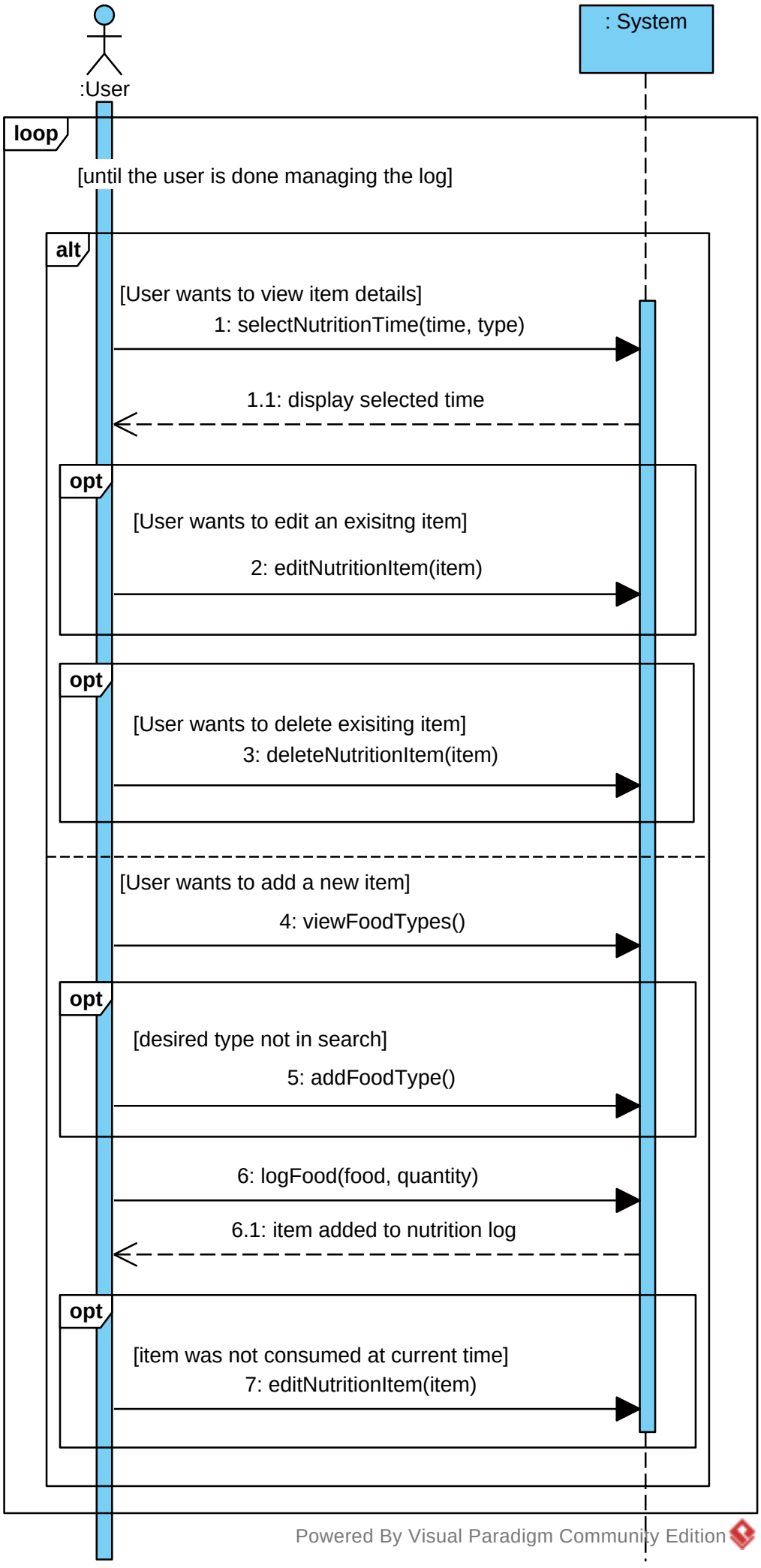
1.2 Search Pantry



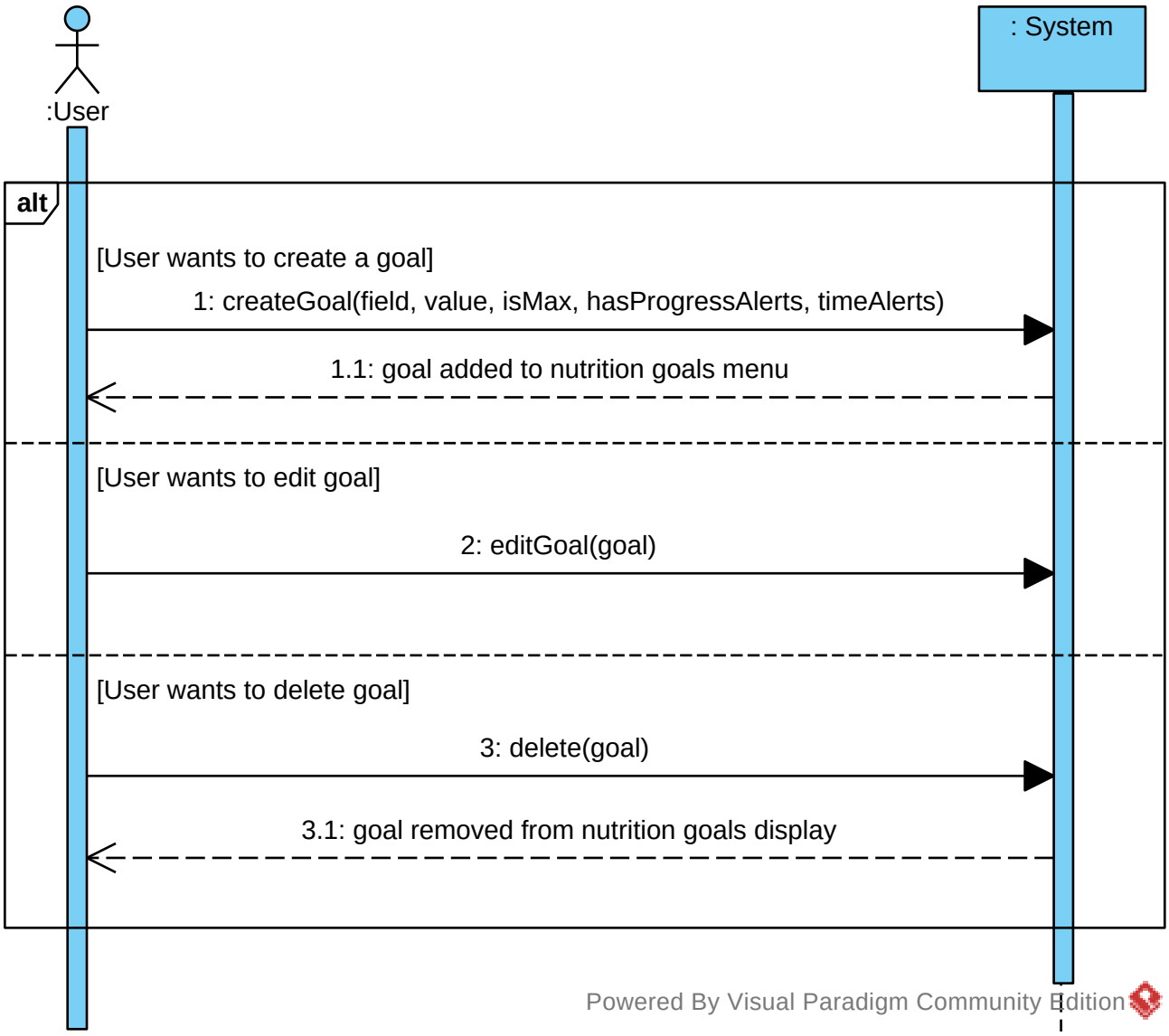
1.3 Consume Pantry Item



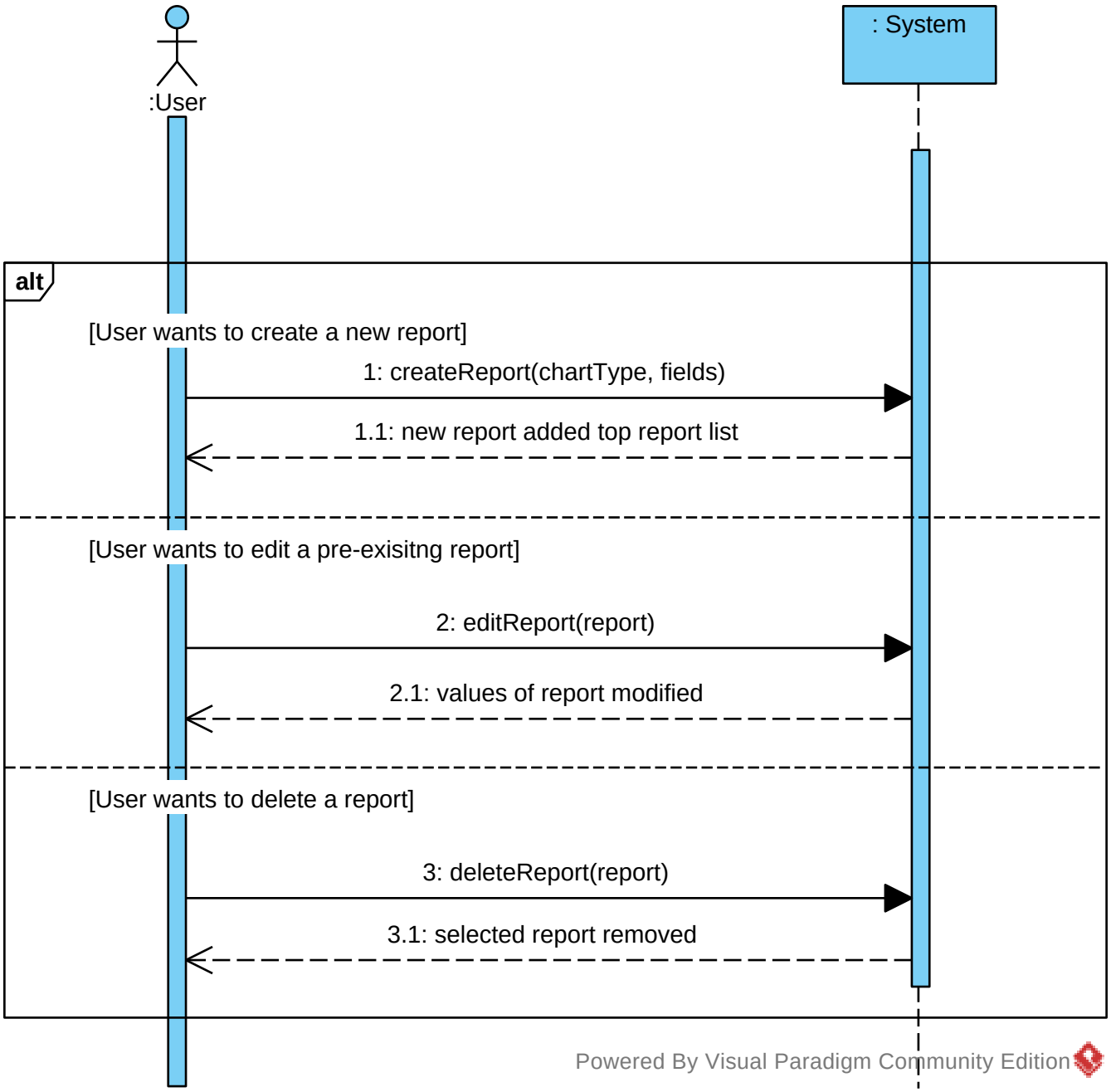
2.1 Manage Nutrition Log



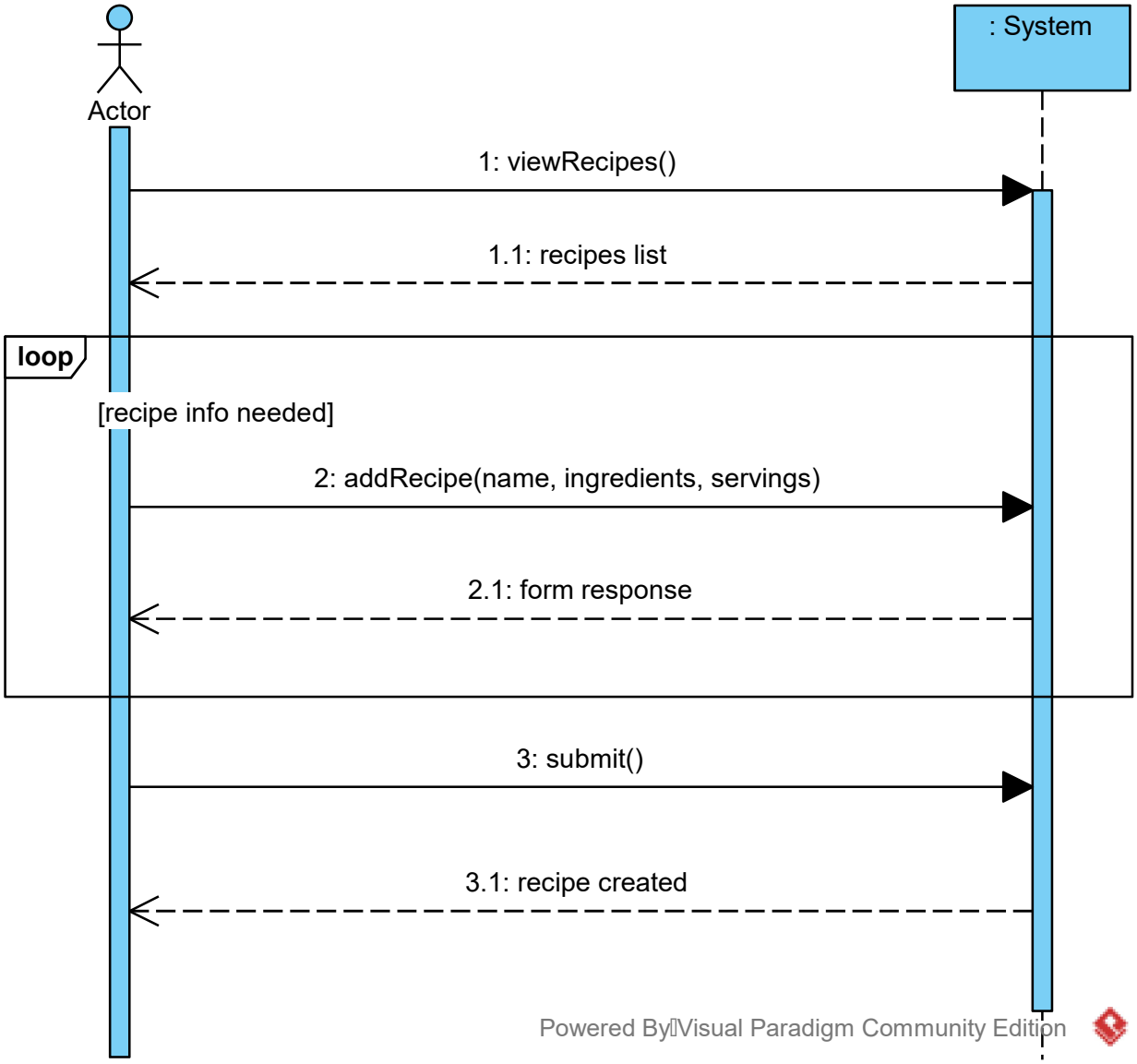
2.2 Nutrition Goals



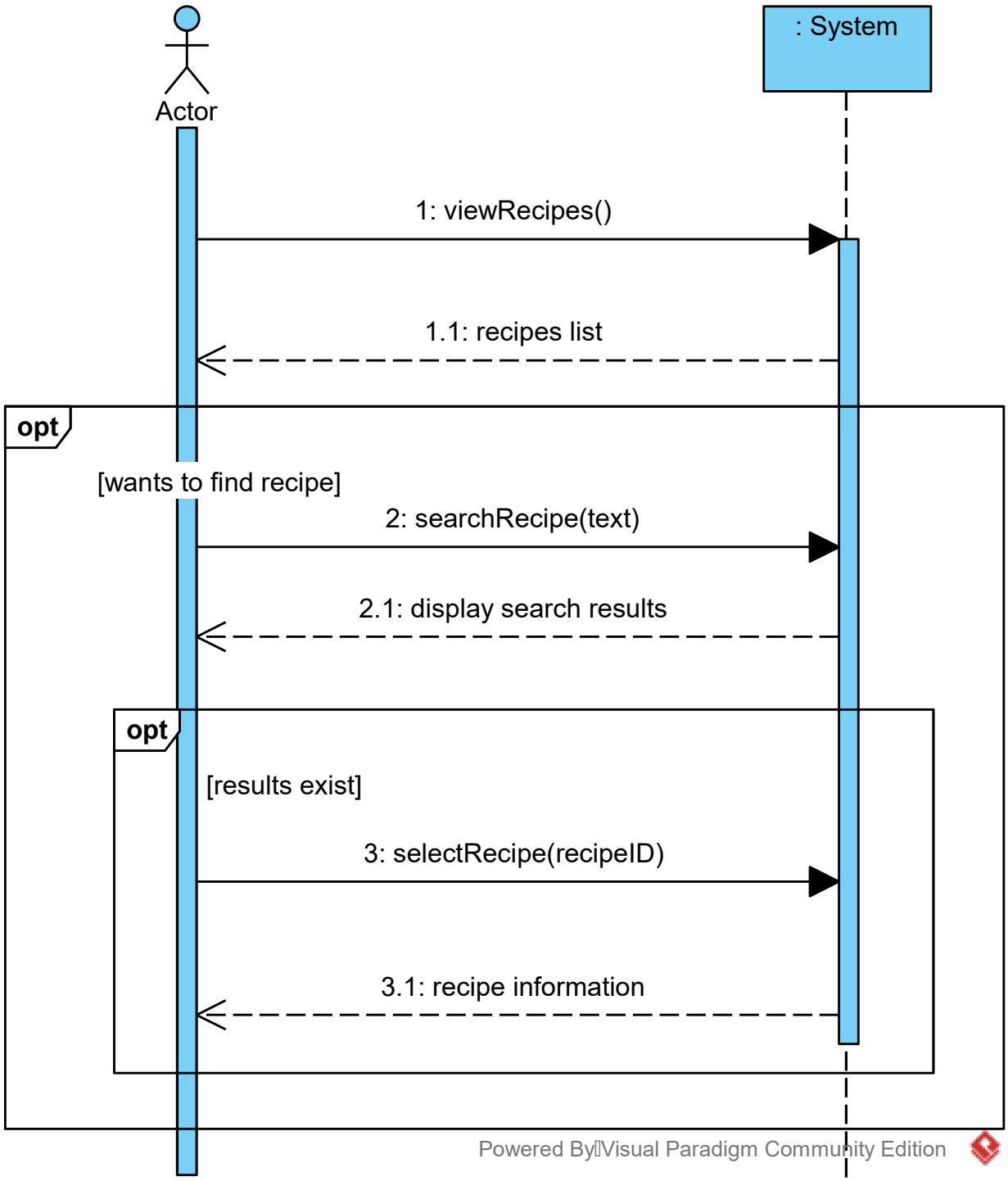
2.3 Nutritional Report



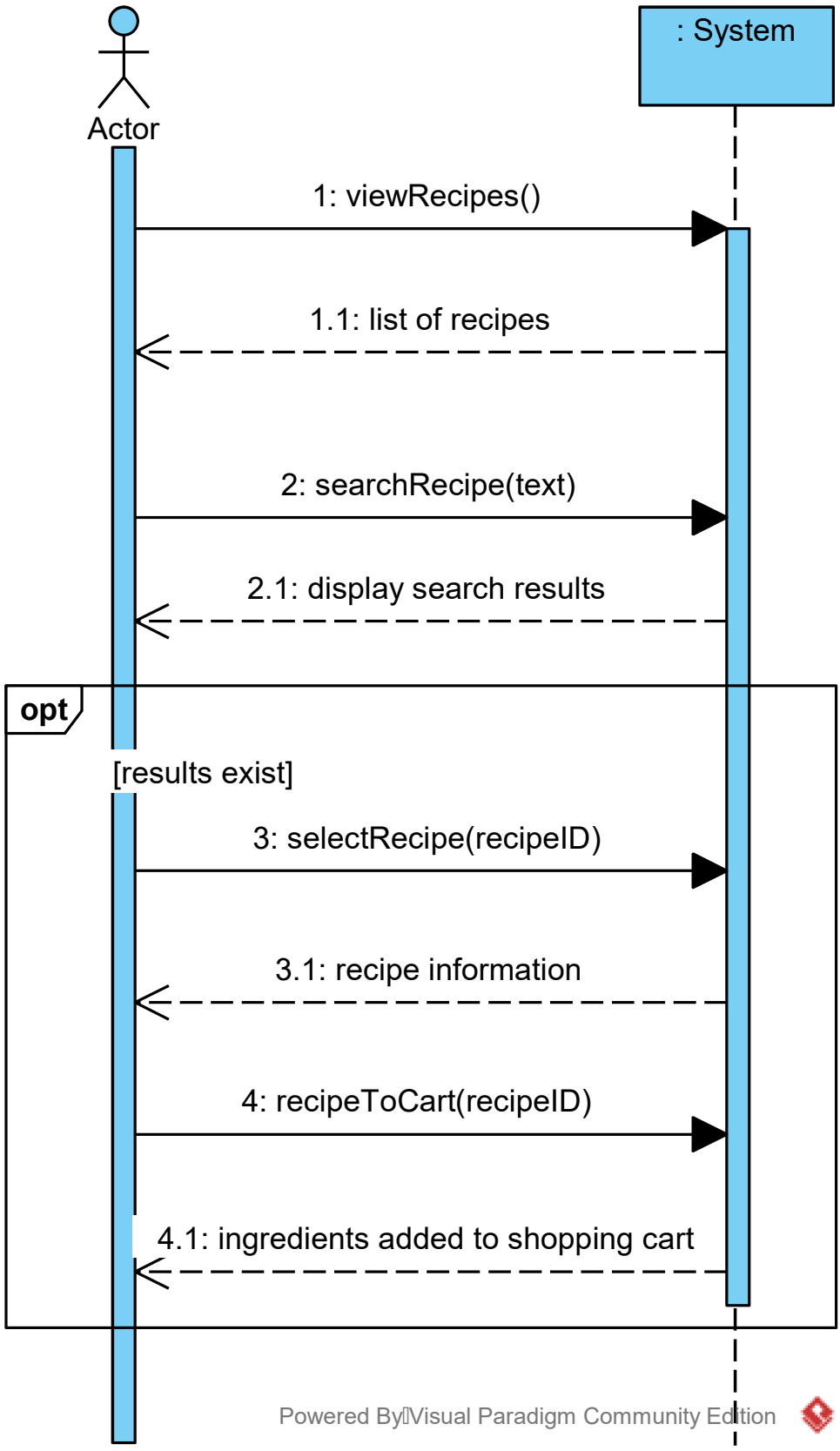
3.1 Create Recipes



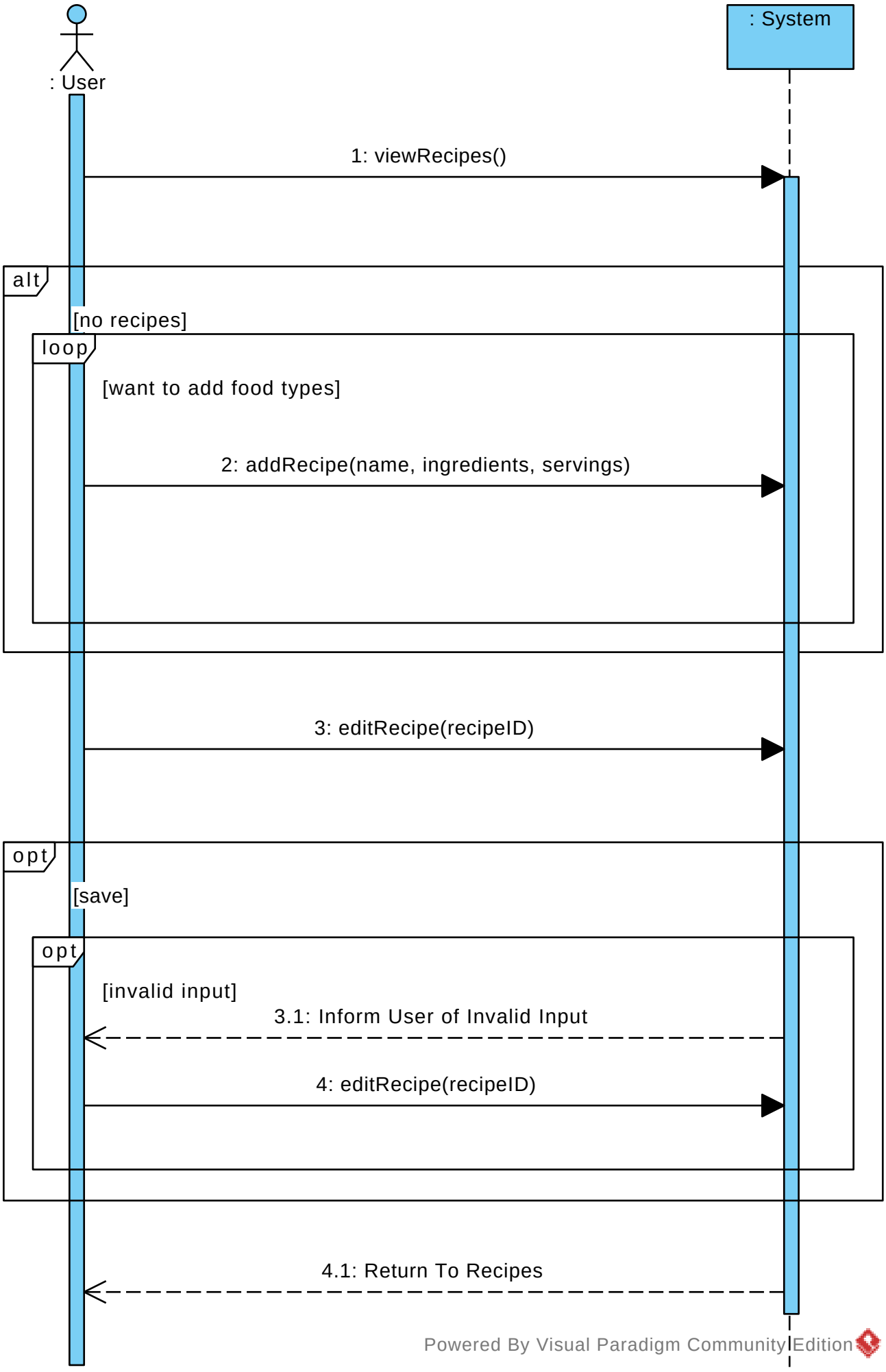
3.2 View Recipes



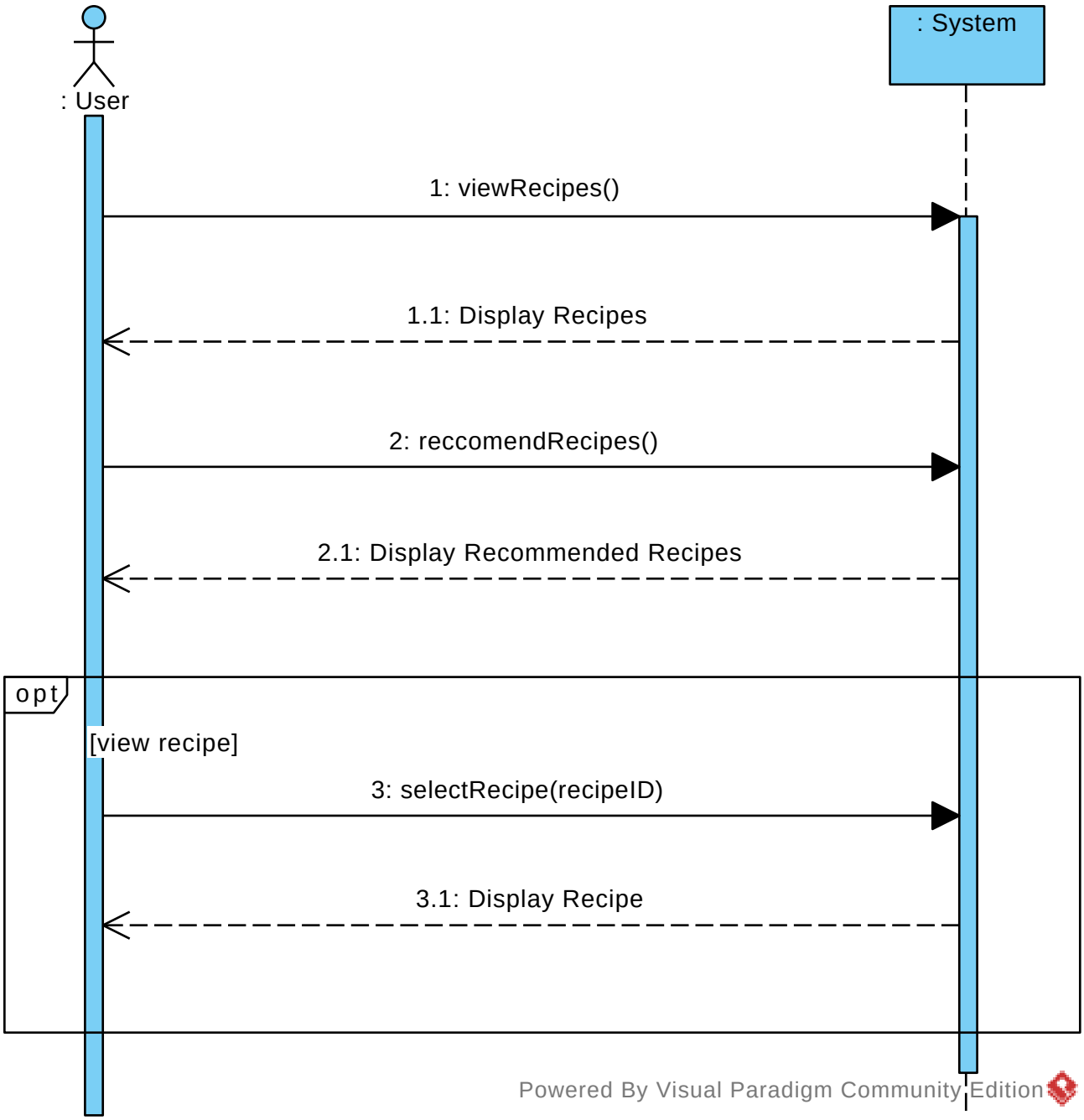
3.3 Add Recipe to Shopping List



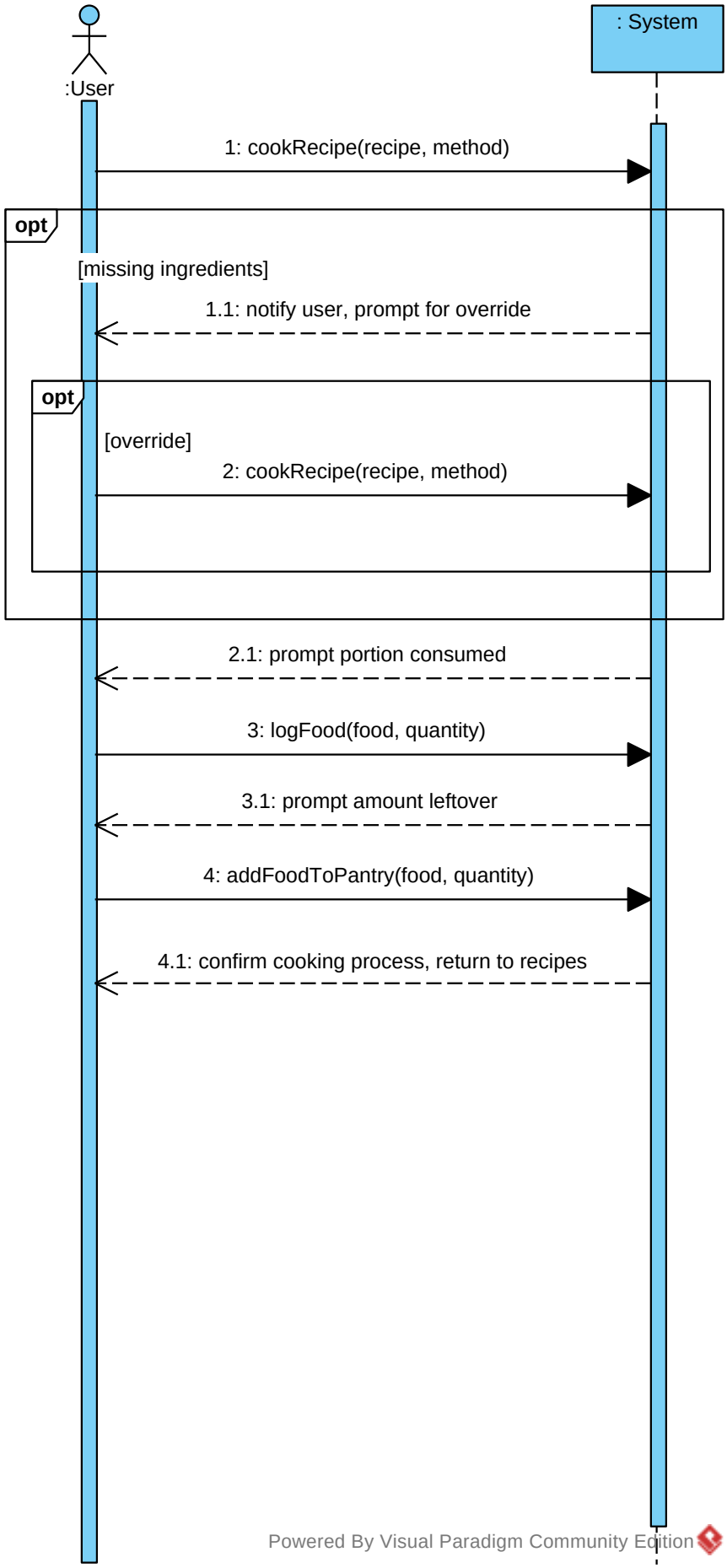
3.4 Modify Recipe



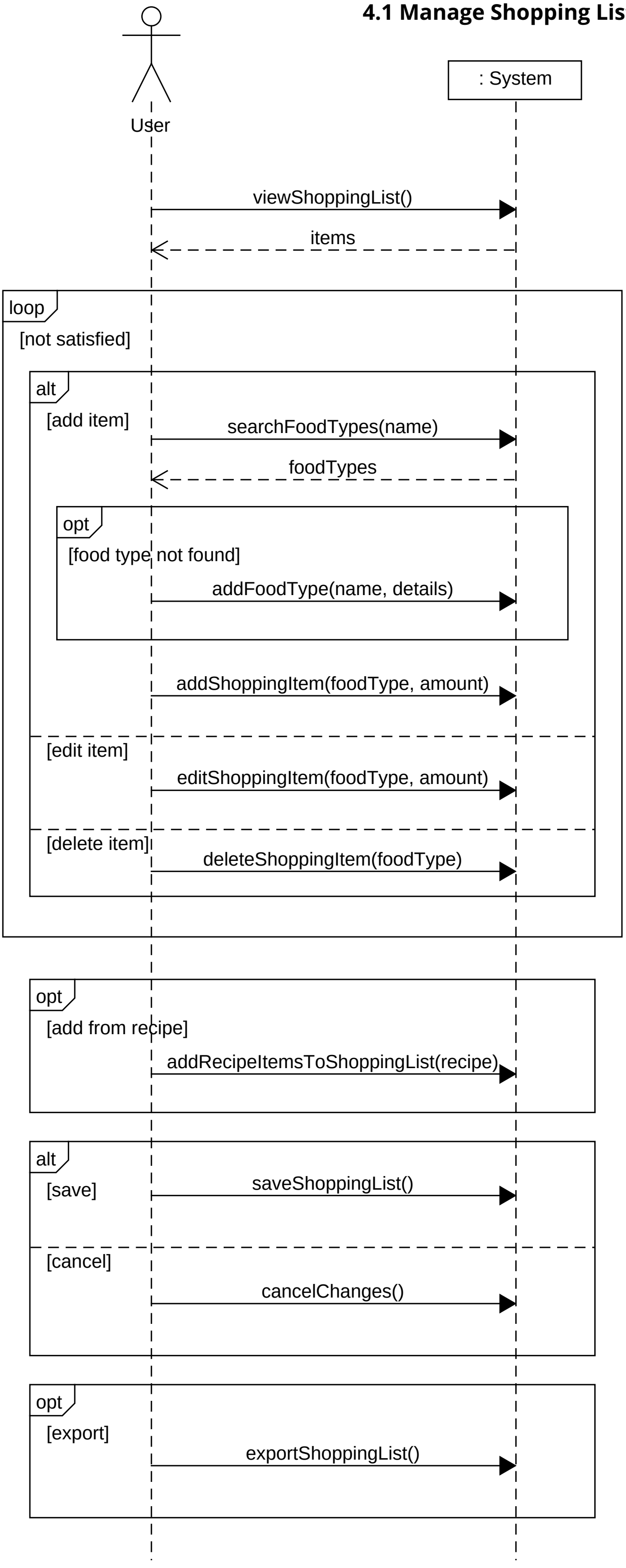
3.5 Recommend Recipe



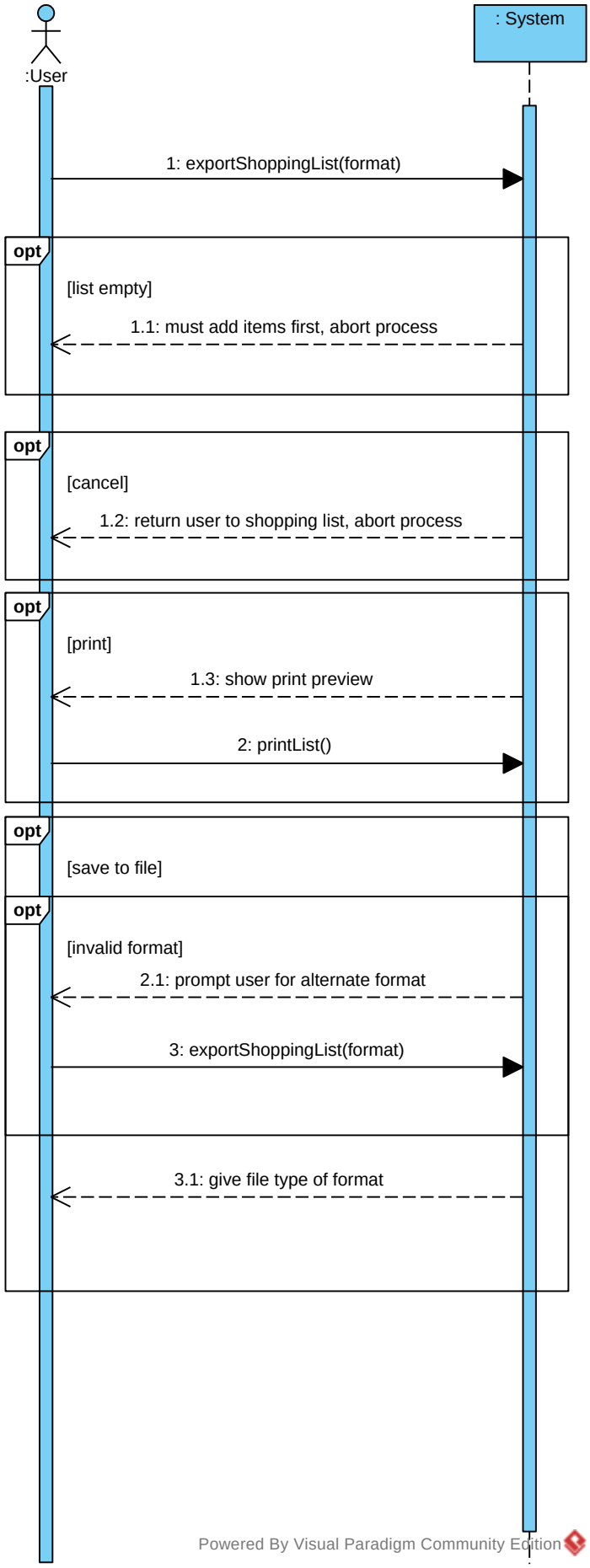
3.6 Cook Recipe



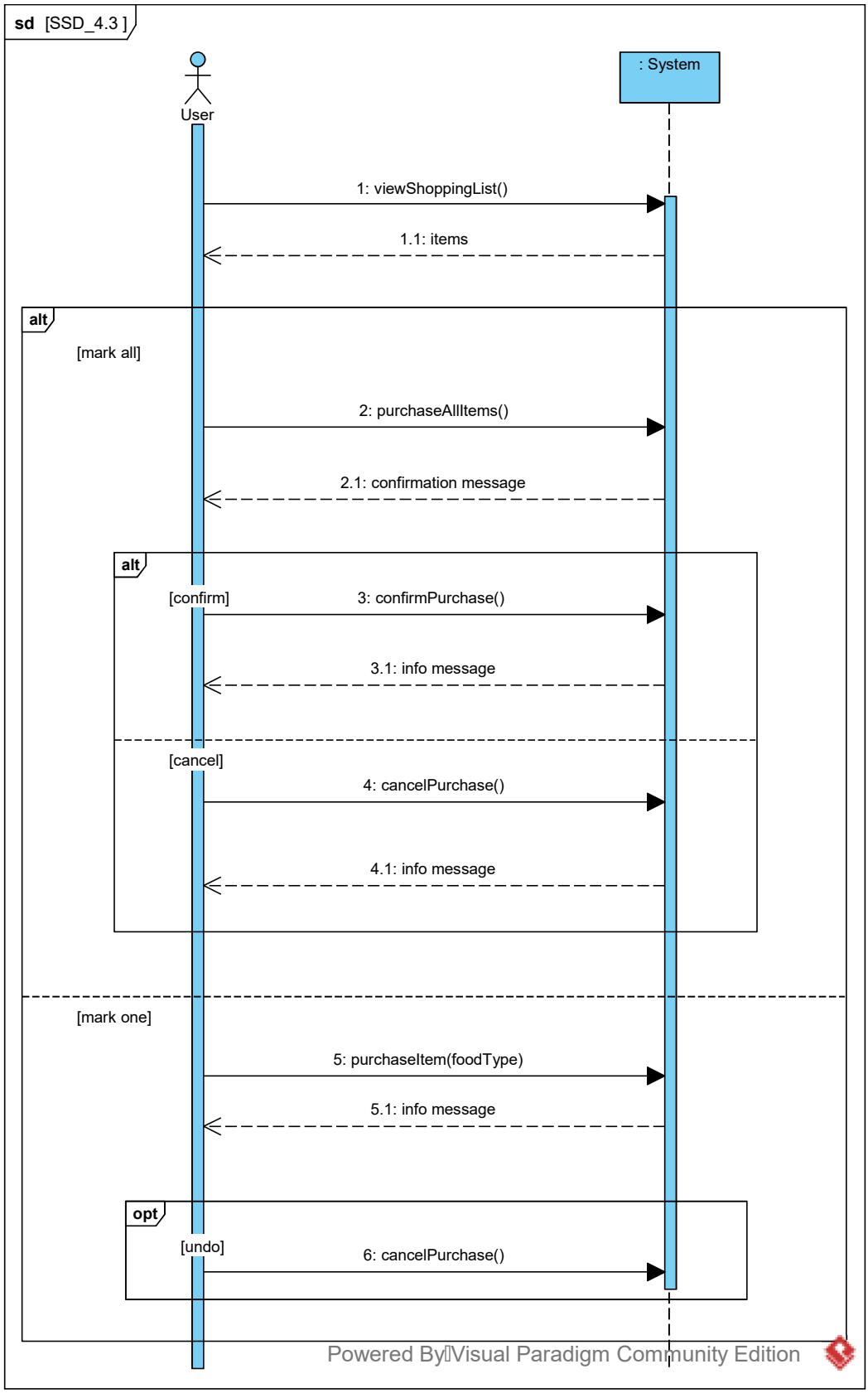
4.1 Manage Shopping List



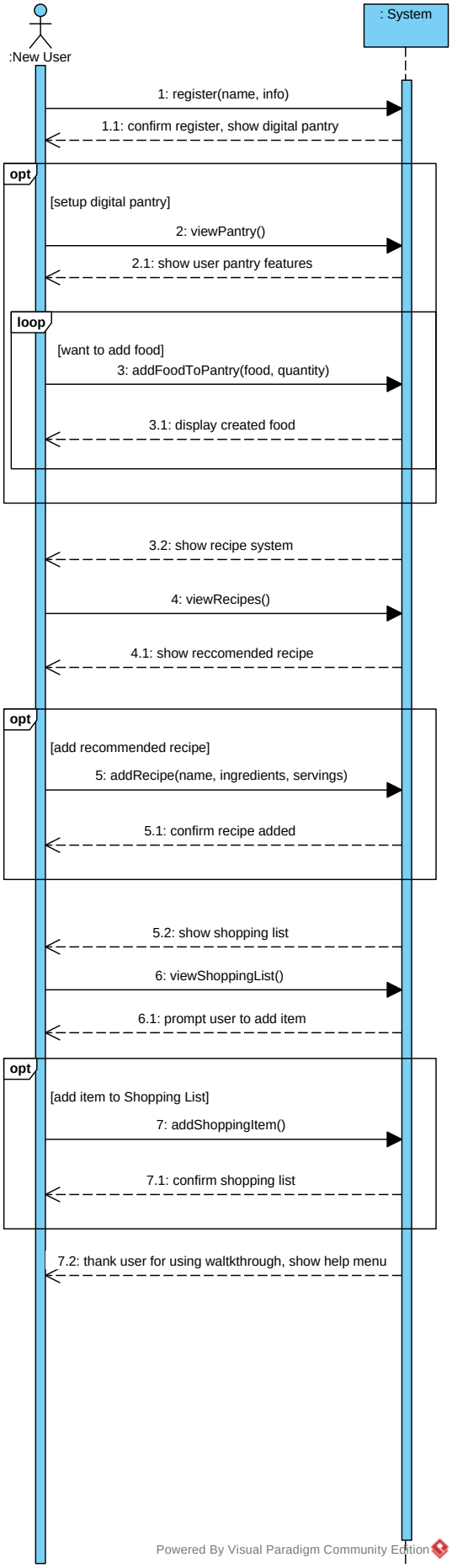
4.2 Export Shopping List



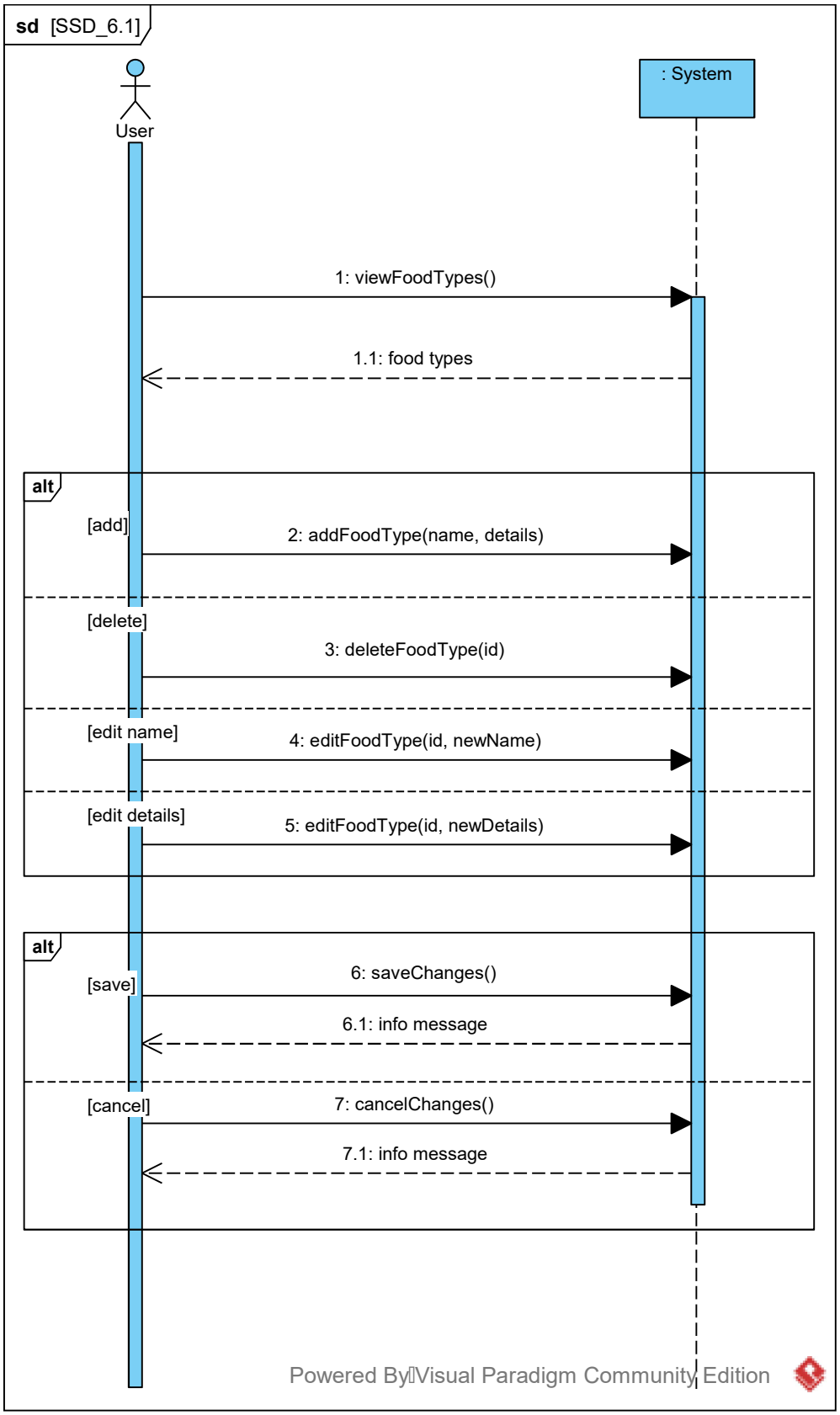
4.3 Mark Purchased Items



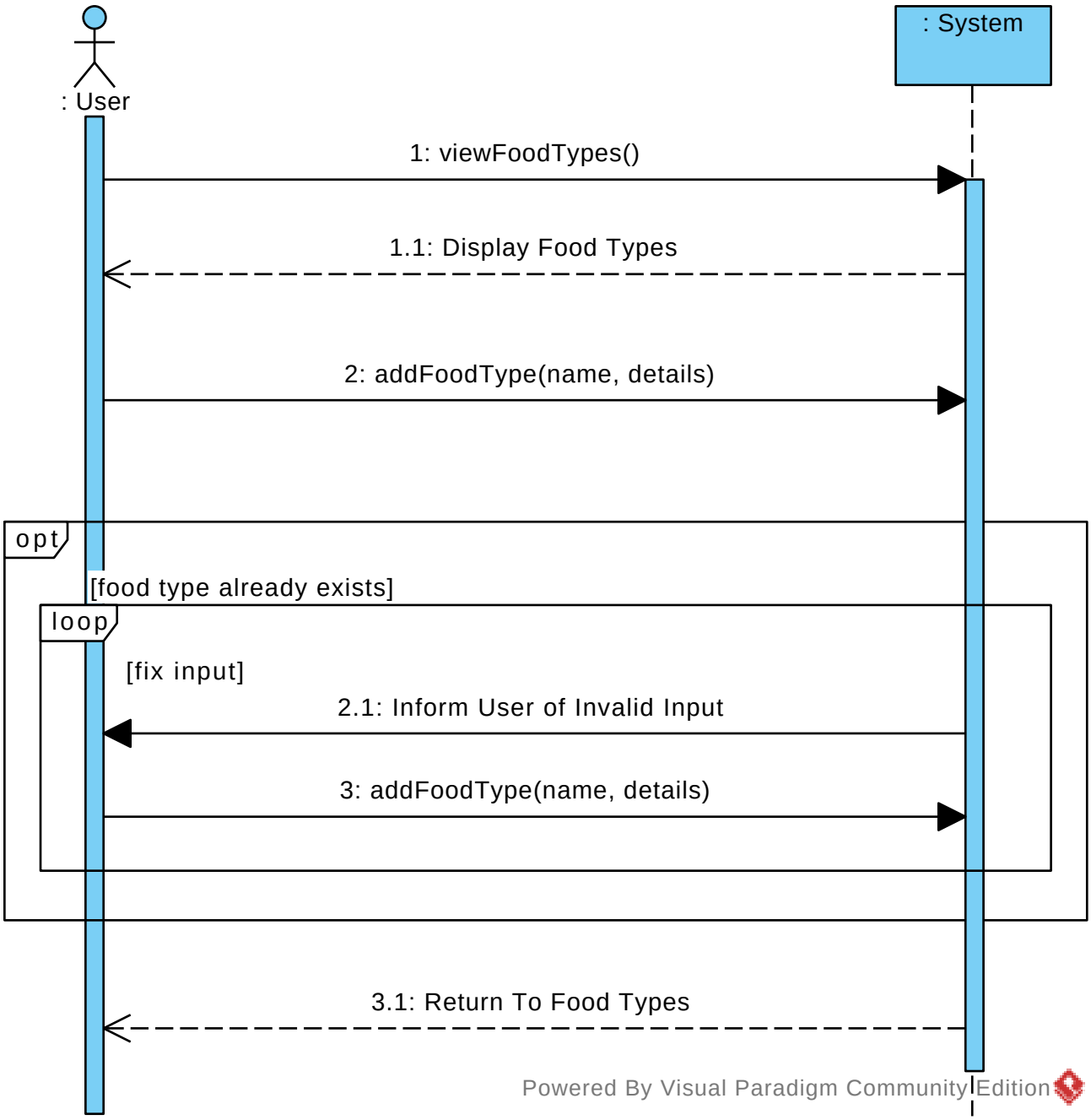
5.1 Startup Tutorial



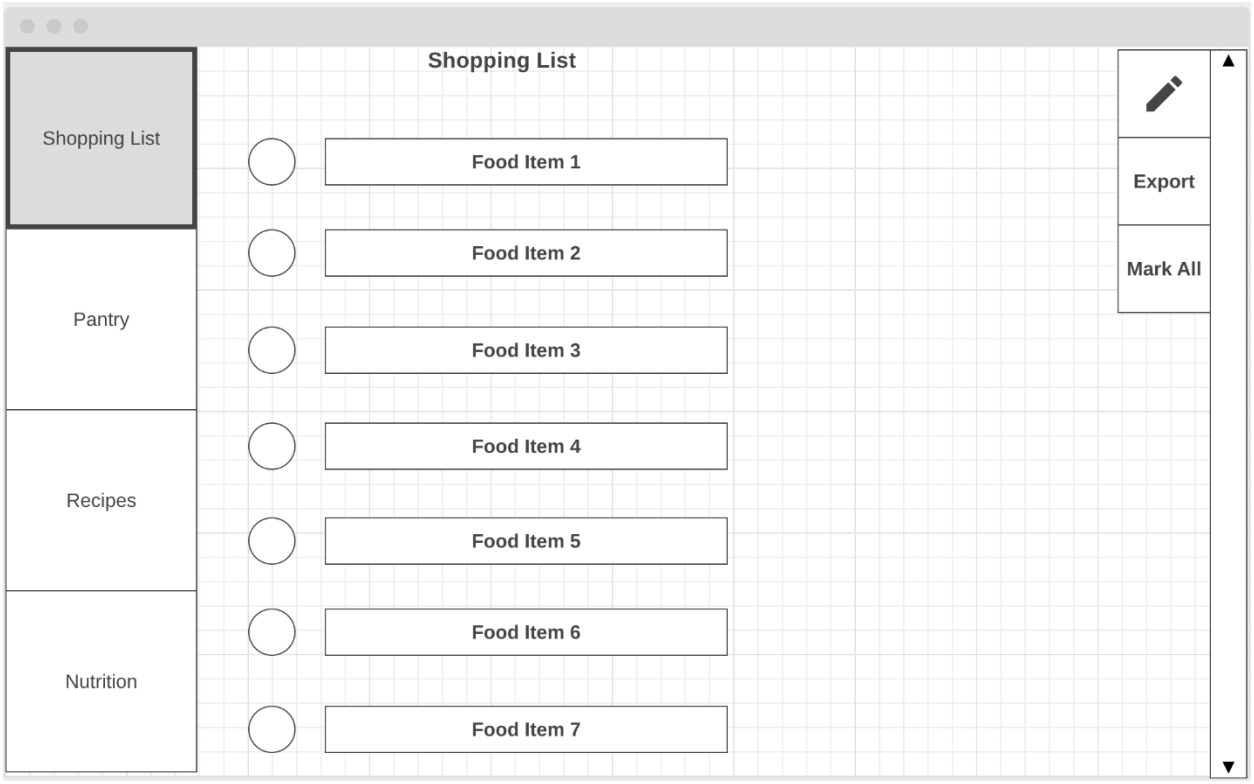
6.1 Manage Food Types

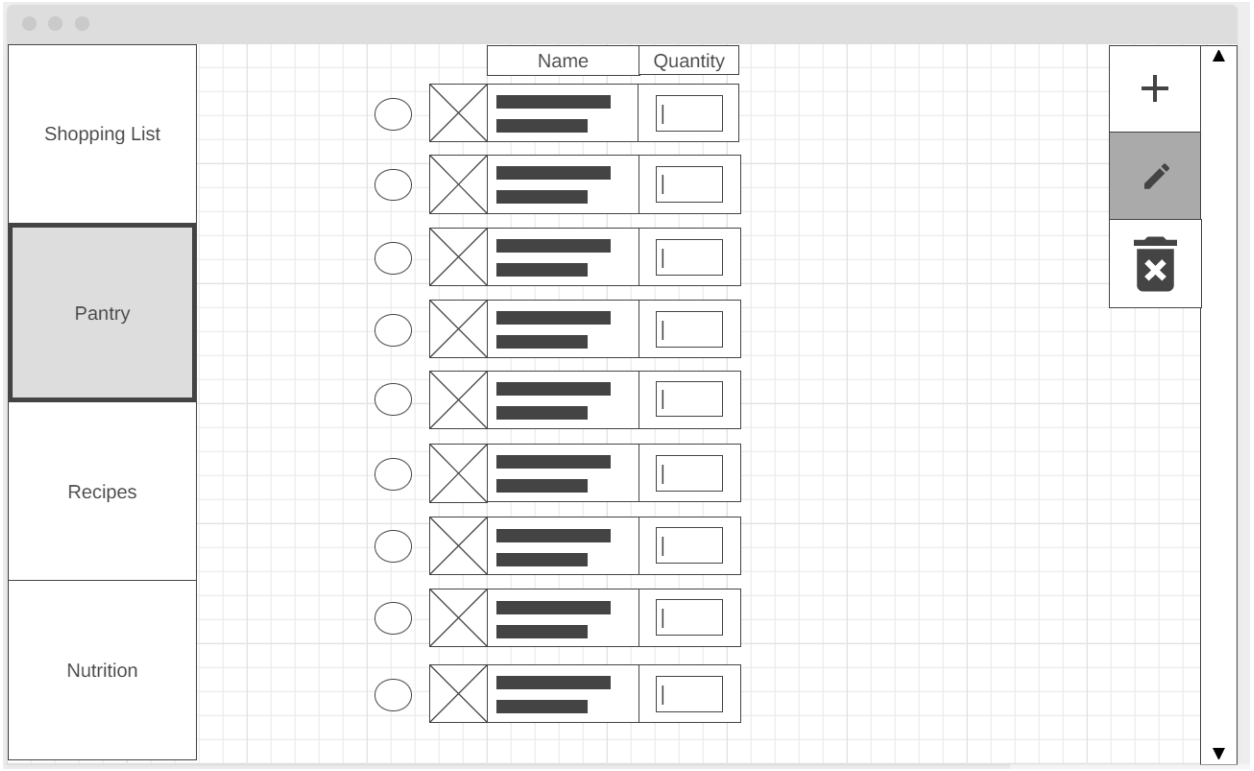


6.2 Create New Food Type

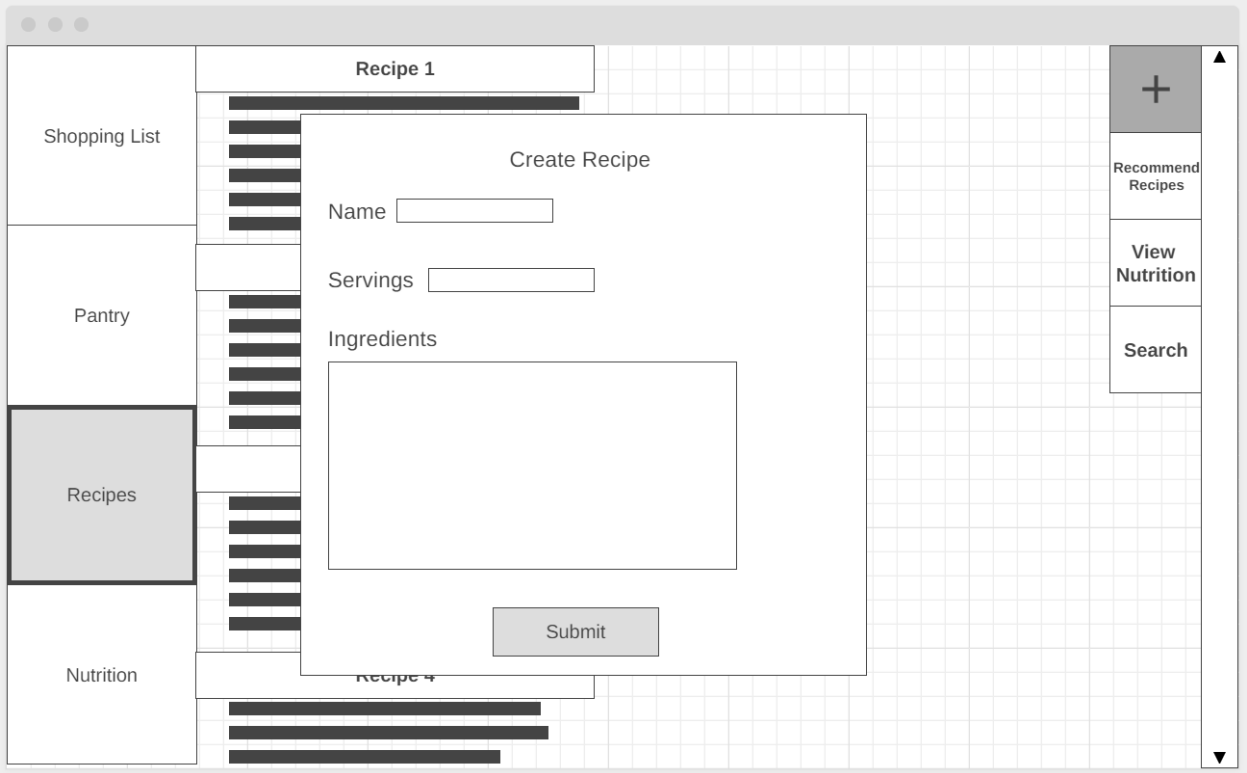


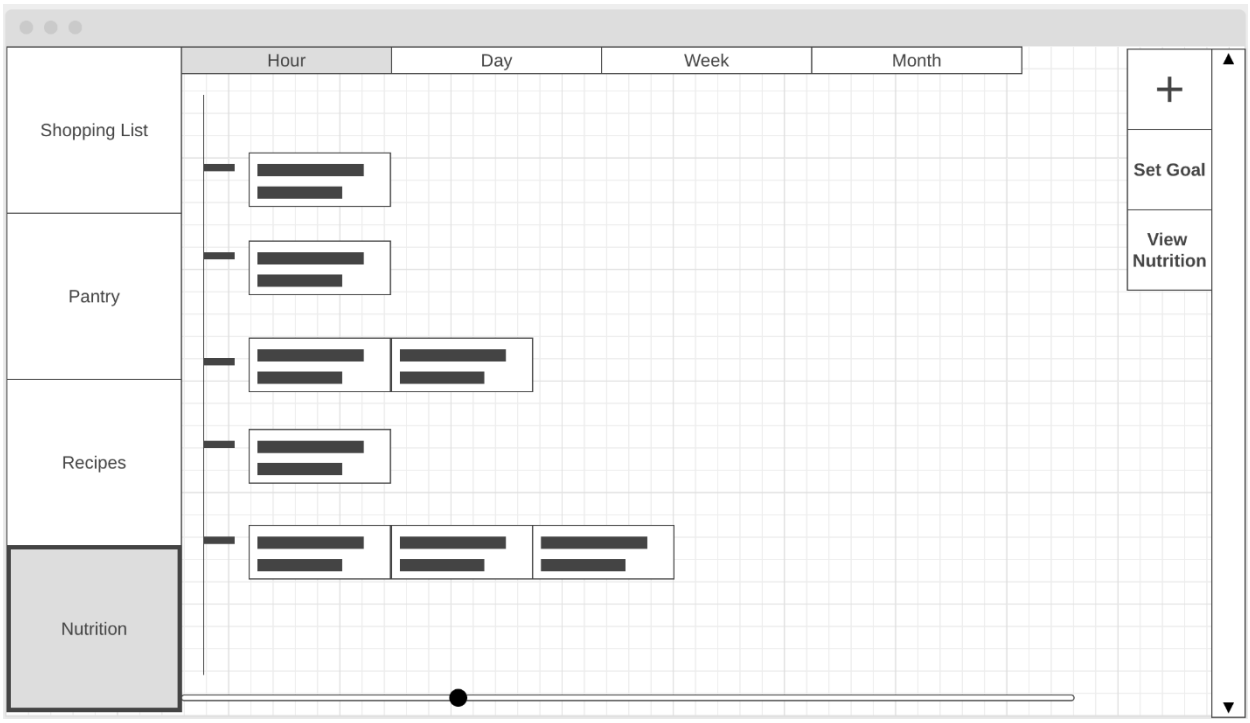
Wire Frames

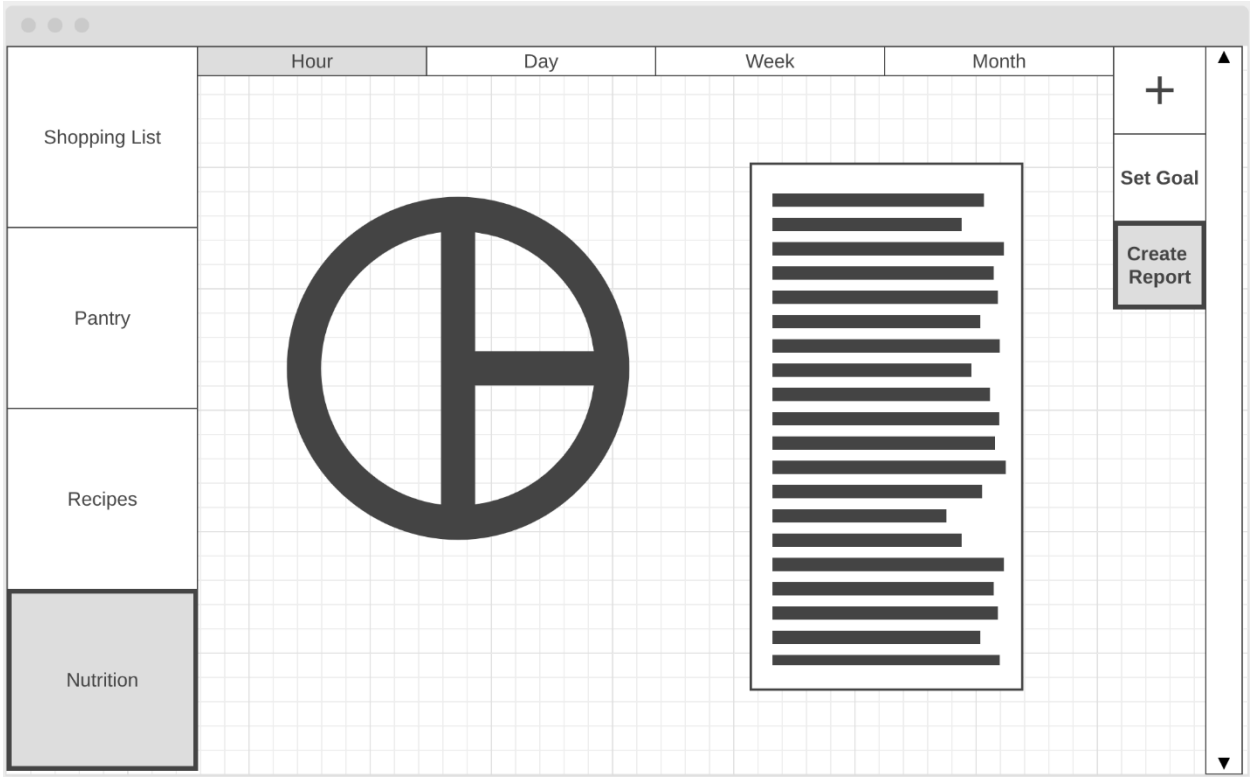

Shopping List
Shopping List
Pantry
Recipes
Nutrition
Food Item 1
Food Item 2
Food Item 3
Food Item 4
Food Item 5
Food Item 6
Food Item 7
Export
Mark All


Shopping List
Pantry
Recipes
Nutrition
Name
Quantity

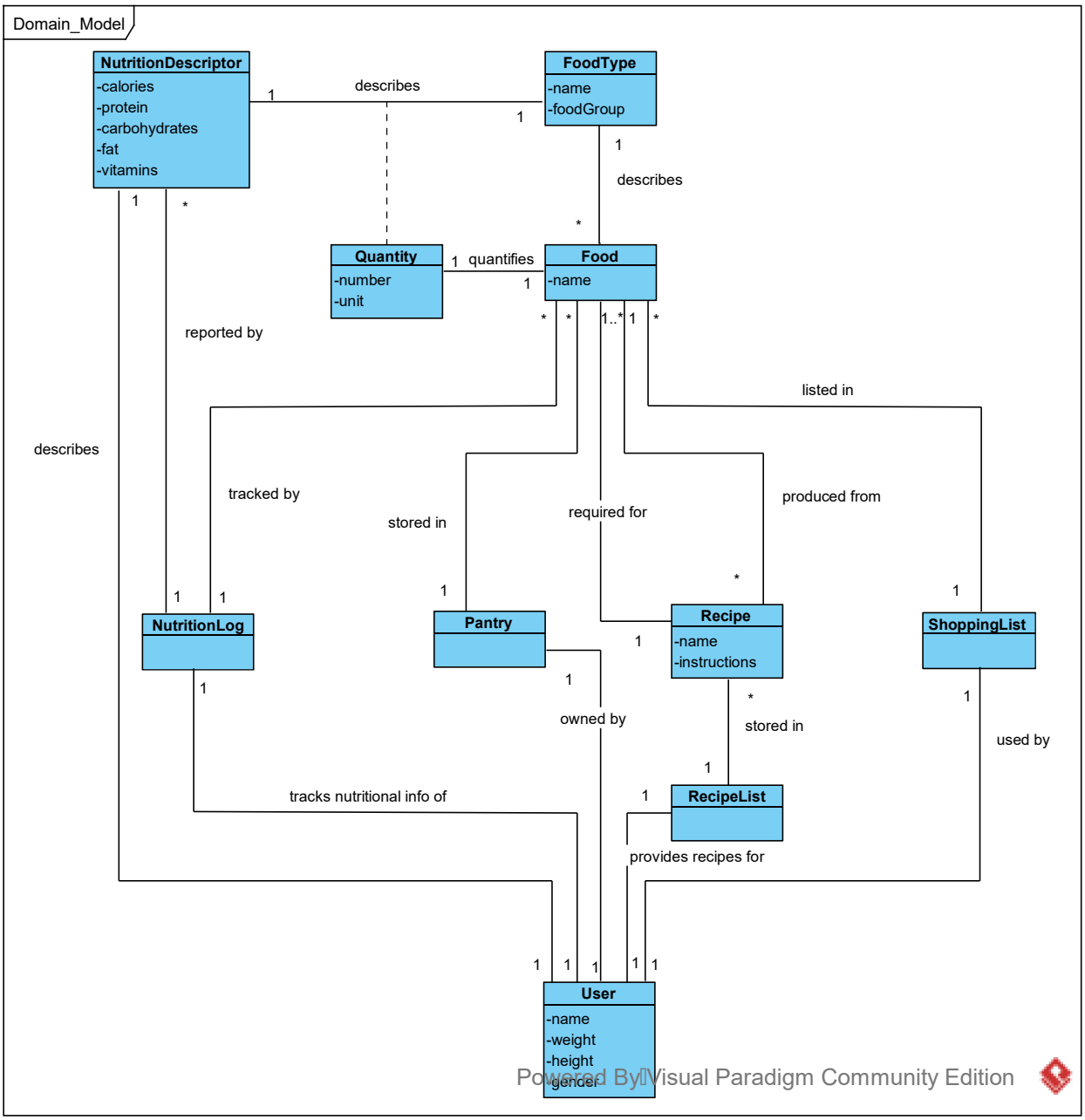

Shopping List
Pantry
Recipes
Nutrition
Recipe 1
Recipe 2
Recipe 3
Recipe 4
+
Recommend Recipes
View Nutrition
Search


Shopping List
Pantry
Recipes
Nutrition
Recipe 1
Create Recipe
Name
Servings
Ingredients
Submit
+
Recommend Recipes
View Nutrition
Search


Hour
Day
Week
Month
Shopping List
Pantry
Recipes
Nutrition
+
Set Goal
View Nutrition


Hour
Day
Week
Month
Shopping List
Pantry
Recipes
Nutrition
+
Set Goal
Create Report

Domain Model



Operation Contracts

STARTUP CONTRACTS

Contract CO1: register

Operation:	register(name, info)
Cross References:	Use Cases: 5.1 Startup
Preconditions:	none
Postconditions:	● A user instance has been instantiated ● User attributes have been initialized (name, opt. weight, gender)

PANTRY CONTRACTS

Contract CO2: viewPantry

Operation:	viewPantry()
Cross References:	Use Cases: 5.1 Startup, 1.1 Manage Pantry, 1.2 Search Pantry, 1.3 Consume Pantry Item
Preconditions:	User has been registered within the system and initialized
Postconditions:	● System provides pantry data to user

Contract CO3: addFoodToPantry

Operation:	addFoodToPantry(food, quantity)
Cross References:	Use Cases: 5.1 Startup, 1.1 Manage Pantry
Preconditions:	User is viewing their pantry interface.
Postconditions:	● Added food exists within the food database ● Food exists within the user's pantry ● Pantry quantity of food item has been incremented to quantity + old quantity

Contract CO4: editFoodInPantry

Operation:	editFoodInPantry(name, quantity)
Cross References:	Use Cases: 1.1 Manage Pantry
Preconditions:	User is viewing their pantry interface. User has selected an existing food item.
Postconditions:	● Food item in pantry with given name is updated with new quantity ● Pantry view is updated.

Contract CO5: removeFoodInPantry

Operation:	removeFoodInPantry(name)
Cross References:	Use Cases: 5.1 Startup, 1.1 Manage Pantry
Preconditions:	User is viewing their pantry interface. User has selected an existing food item.
Postconditions:	● Food item with given name has its instance removed from the pantry. ● Pantry view is updated.

Contract CO6: searchPantry

Operation:	searchPantry(itemName)
Cross References:	Use Cases: 1.2 Search Pantry
Preconditions:	User is viewing their pantry interface.
Postconditions:	● List of food items containing itemName are returned or an empty list if no pantry items match the itemName

Contract CO7: consume

Operation:	consume(foodItem)
Cross References:	Use Cases: 1.3 Consume Pantry Item
Preconditions:	User is viewing their pantry interface. FoodItem passed in exists in the pantry.
Postconditions:	● Food in pantry with type foodItem has its quantity decremented ● If new quantity is 0, foodItem is removed from pantry

SHOPPING LIST CONTRACTS

Contract CO8: viewShoppingList

Operation:	viewShoppingList()
Cross References:	Use Cases: 4.1 Manage Shopping List, 4.3 Mark Purchased Items
Preconditions:	User has been registered within the system and initialized
Postconditions:	● System provides user with their shopping list

Contract CO9: addShoppingItem

Operation:	addShoppingItem(foodType, quantity)
Cross References:	Use Cases: 4.1 Manage Shopping List
Preconditions:	● User is viewing the shopping list interface ● System is in "Modify Shopping List" mode ● The food type exists within the list of registered food types ● The quantity is valid (non-negative, etc)
Postconditions:	● Food type and quantity added to the shopping list

Contract CO10: editShoppingItem

Operation:	editShoppingItem(foodType, quantity)
Cross References:	Use Cases: 4.1 Manage Shopping List
Preconditions:	● User is viewing the shopping list interface ● System is in "Modify Shopping List" mode ● The food type exists within the shopping list ● The quantity is valid (non-negative, etc)
Postconditions:	● Quantity of given food type updated within the shopping list

Contract CO11: deleteShoppingItem

Operation:	deleteShoppingItem(foodType)
Cross References:	Use Cases: 4.1 Manage Shopping List
Preconditions:	● User is viewing the shopping list interface ● System is in "Modify Shopping List" mode ● The food type exists within the shopping list ● The quantity is valid (non-negative, etc)
Postconditions:	● Food type removed from shopping list

Contract CO12: exportShoppingList

Operation:	exportShoppingList(format)
Cross References:	Use Cases: 4.1 Manage Shopping List, 4.2 Export Shopping List
Preconditions:	● User has selected that they wish to export shopping list ● Shopping list instance exists
Postconditions:	● Shopping list has been exported to desired format

Contract CO13: printList

Operation:	printList()
Cross References:	Use Cases: 4.2 Export Shopping List
Preconditions:	● User has selected that they wish to export shopping list ● Shopping list instance exists ● Format is selected as print
Postconditions:	● Shopping list has been opened in print preview ● User can confirm printing at their printer

Contract CO14: purchaseAllItems

Operation:	purchaseAllItems()
Cross References:	Use Cases: 4.3 Mark Purchased Items
Preconditions:	● User is viewing the shopping list interface ● Shopping list food types are registered in the list of food types ● User purchased all items on the shopping list

Postconditions:	● System moves all items from shopping list to pantry

Contract CO15: purchaseItem

Operation:	purchaseItem(foodType)
Cross References:	Use Cases: 4.3 Mark Purchased Items
Preconditions:	● User is viewing the shopping list interface ● Given food type is registered in the shopping list ● Given food type is registered in the list of food types ● User purchased the given item
Postconditions:	● System moves item of given food type from shopping list to pantry

RECIPE CONTRACTS

Contract CO16: viewRecipes

Operation:	viewRecipes()
Cross References:	Use Cases: 3.1 Create Recipe, 3.2 View Recipe, 3.3 Add Recipe to Shopping List, 3.4 Modify Recipe, 3.5 Recommend Recipes
Preconditions:	The user has been initialized and registered within the system.
Postconditions:	● System provides the user's list of recipes ● System fetches and displays a few recommended recipes

Contract CO17: addRecipe

Operation:	addRecipe(name, ingredients, servings)
Cross References:	Use Cases: 5.1 Startup, 3.4 Modify Recipe
Preconditions:	The user has been initialized and registered within the system.
Postconditions:	● Ingredients have been stored has food instances within the database ● The recipe represents a new recipe instance within the recipe system ● Recipe appears within the recipe list

Contract CO18: cookRecipe

Operation:	cookRecipe(recipe, method)
Cross References:	Use Cases: 3.6 Cook Recipe
Preconditions:	● Recipe exists within the recipe system ● Ingredients exist within the food type database ● Method specifies how to cook the recipe (use all pantry items, use no pantry items, use only on hand).
Postconditions:	● User is notified of any errors preventing cooking ● If errors have been resolved, user can specify serving amounts and leftover amounts ● Recipe is logged to food log ● Pantry is updated to contain leftovers

Contract CO19: editRecipe

Operation:	editRecipe(recipeID)
Cross References:	Use Cases: 3.4 Modify Recipe
Preconditions:	● Method specifies what fields of recipe are to be modified
Postconditions:	● User is notified of any errors preventing modification ● If errors have been resolved, user can save the modified recipe ● Recipe is saved to Recipe List ● Food Database is updated with any new Food Types

Contract CO20: selectRecipe

Operation:	selectRecipe(recipeID)
Cross References:	Use Cases: 3.5 Recommend Recipes
Preconditions:	● Method specifies what recipe is to be displayed ● Recipe exists within the recipe system
Postconditions:	● Recipe is displayed to the user

Contract CO21: recommendRecipes

Operation:	recommendRecipes()
Cross References:	Use Cases: 3.5 Recommend Recipe
Preconditions:	● Digital Pantry is non-empty ● Food Database is non-empty ● Recipe List is non-empty
Postconditions:	● The user is presented with all recommended recipes from the Recipe List based off of items in their pantry

Contract CO22: searchRecipe

Operation:	searchRecipe(text)
Cross References:	Use Cases: 3.2 View Recipes
Preconditions:	● User is registered within the system ● User has created at least one recipe
Postconditions:	● List of matching recipes are displayed to the user

Contract CO23: recipeToCart

Operation:	recipeToCart(recipeID)
Cross References:	Use Cases: 3.3 Add Recipe to Shopping Cart
Preconditions:	● User is registered within the system ● User has created at least one recipe
Postconditions:	● Recipe ingredients added to shopping cart

NUTRITION LOG CONTRACTS

Contract CO24: logFood

Operation:	logFood(food, quantity)
Cross References:	Use Cases: 3.6 Cook Recipe, 2.1 Manage Nutrition Log
Preconditions:	● Recipe exists within the recipe system ● Ingredients exist within the food type database with nutrition info ● Recipe cook has been completed
Postconditions:	● New food instance is created from the cooked recipe ● Nutrition log is updated with new food instance

Contract CO25: selectNutritionTime

Operation:	selectNutritionTime(time, type)
Cross References:	Use Cases: 2.1 Manage Nutrition Log
Preconditions:	● The user is in the nutrition log ● At least one item exists within the log
Postconditions:	● The selected item has been displayed within the timeline

Contract CO26: editNutritionItem

Operation:	editNutritionItem(item)
Cross References:	Use Cases: 2.1 Manage Nutrition Log
Preconditions:	● The user is in the nutrition log ● At least one item exists within the log which the user wants to edit
Postconditions:	● The selected item has had its values changed as per user request

Contract CO27: deleteNutritionItem

Operation:	deleteNutritionItem()
Cross References:	Use Cases: 2.1 Manage Nutrition Log
Preconditions:	● The user is in the nutrition log ● At least one item exists within the log which the user wants to delete
Postconditions:	● The selected item has been removed from the log

Contract CO28: createGoal

Operation:	createGoal(field, value, isMax, hasProgressAlerts, timeAlerts)
Cross References:	Use Cases: 2.2 Nutrition Goals
Preconditions:	● The user is in the nutrition goals menu ● The input values are valid and logical
Postconditions:	● A nutrition goal has been created using the given fields

Contract CO29: editGoal

Operation:	editGoal(goal)
Cross References:	Use Cases: 2.2 Nutrition Goals
Preconditions:	● The user is in the nutrition goals menu ● At least one goal exists within the menu which the user wants to edit
Postconditions:	● The selected goal has had its values changed as per user request

Contract CO30: deleteGoal

Operation:	deleteGoal(goal)
Cross References:	Use Cases: 2.2 Nutrition Goals
Preconditions:	● The user is in the nutrition goal menu ● At least one goal exists within the menu which the user wants to delete
Postconditions:	● The selected item has been removed from the log

Contract CO31: createReport

Operation:	createReport(chartType, fields)
Cross References:	Use Cases: 2.3 Manage Nutrition Report
Preconditions:	● The user is in the nutrition reports menu ● The input values are valid and logical
Postconditions:	● A nutrition report has been created using the given fields

Contract CO32: editReport

Operation:	editReport(report)
Cross References:	Use Cases: 2.3 Manage Nutrition Report
Preconditions:	● The user is in the nutrition reports menu ● At least one report exists within the menu which the user wants to edit
Postconditions:	● The selected report has had its values changed as per user request

Contract CO33: deleteReport

Operation:	deleteReport(report)
Cross References:	Use Cases: 2.3 Manage Nutrition Report
Preconditions:	● The user is in the nutrition report menu ● At least one report exists within the menu which the user wants to delete
Postconditions:	● The selected report has been removed from the report menu

FOODTYPE CONTRACTS

Contract CO34: viewFoodTypes

Operation:	viewFoodTypes()
Cross References:	Use Cases: 6.1 Manage Food Types, 6.2 Create New Food Types, 2.1 Manage Nutrition Log
Preconditions:	● User has been registered within the system and initialized
Postconditions:	● System provides user with the list of food types

Contract CO35: addFoodType

Operation:	addFoodType(name, details), 6.2 Create New Food Types
Cross References:	Use Cases: 6.1 Manage Food Types, 4.1 Manage Shopping List, 1.1 Manage Pantry, 1.2 Search Pantry, 2.1 Manage Nutrition Log
Preconditions:	● User is viewing the food types interface ● System is in "Modify Food Types" mode ● The name is not already in use by another food type ● The name is not empty ● The details are valid (non-negative weight, etc)
Postconditions:	● System adds food type to list of food types

Contract CO36: editFoodType

Operation:	editFoodType(id, newData)
Cross References:	Use Cases: 6.1 Manage Food Types
Preconditions:	● User is viewing the food types interface ● System is in "Modify Food Types" mode ● The new data is valid ● The id refers to an already registered food type
Postconditions:	● System updates food type in list of food types to have a new name and/or details

CO37: deleteFoodType

Operation:	deleteFoodType(id)
Cross References:	Use Cases: 6.1 Manage Food Types
Preconditions:	● User is viewing the food types interface ● System is in “Modify Food Types” mode ● The id refers to an already registered food type
Postconditions:	● System removes given food type from the list of food types

FoodPants-Gantt

Feb 23, 2022

http://

Boothverse

Project manager	Austin Huizinga
Project dates	Jan 18, 2022 - Apr 28, 2022
Completion	0%
Tasks	28
Resources	4

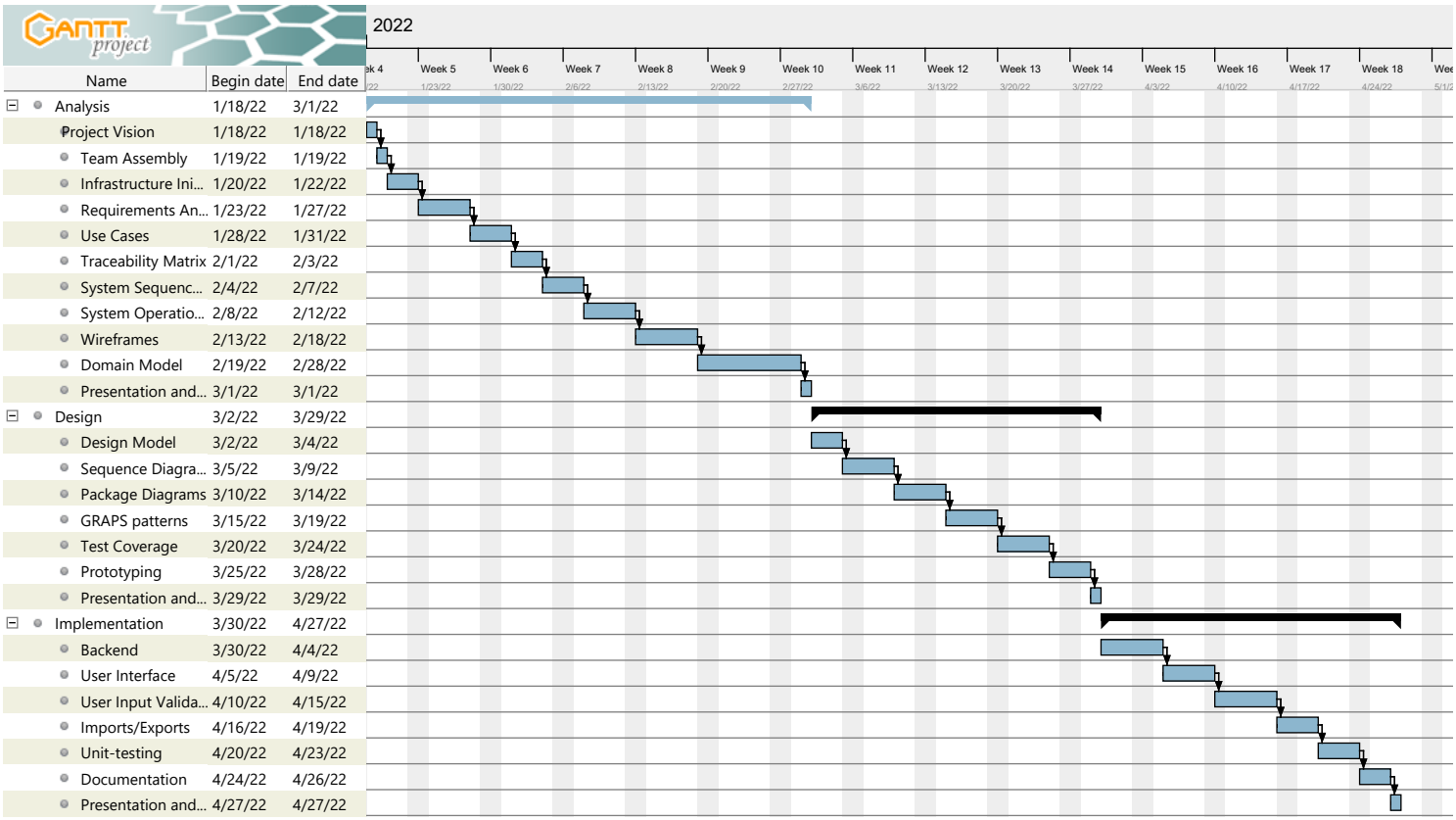
Tasks

Name	Begin date	End date
Analysis	1/18/22	3/1/22
Project Vision	1/18/22	1/18/22
Team Assembly	1/19/22	1/19/22
Infrastructure Initialization	1/20/22	1/22/22
Requirements Analysis	1/23/22	1/27/22
Use Cases	1/28/22	1/31/22
Traceability Matrix	2/1/22	2/3/22
System Sequence Diagrams	2/4/22	2/7/22
System Operations	2/8/22	2/12/22
Wireframes	2/13/22	2/18/22
Domain Model	2/19/22	2/28/22
Presentation and Reporting	3/1/22	3/1/22
Design	3/2/22	3/29/22
Design Model	3/2/22	3/4/22
Sequence Diagrams	3/5/22	3/9/22
Package Diagrams	3/10/22	3/14/22
GRAPS patterns	3/15/22	3/19/22
Test Coverage	3/20/22	3/24/22
Prototyping	3/25/22	3/28/22
Presentation and Reporting	3/29/22	3/29/22
Implementation	3/30/22	4/27/22
Backend	3/30/22	4/4/22
User Interface	4/5/22	4/9/22
User Input Validation	4/10/22	4/15/22
Imports/Exports	4/16/22	4/19/22
Unit-testing	4/20/22	4/23/22
Documentation	4/24/22	4/26/22
Presentation and Reporting	4/27/22	4/27/22

Resources

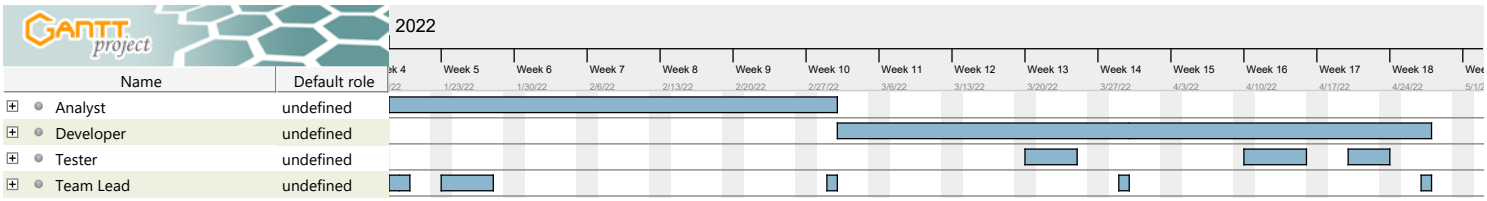
Name	Default role
Analyst	undefined
Developer	undefined
Tester	undefined
Team Lead	undefined

Gantt Chart



Name	Begin date	End date
Analysis	1/18/22	3/1/22
Project Vision	1/18/22	1/18/22
Team Assembly	1/19/22	1/19/22
Infrastructure Ini...	1/20/22	1/22/22
Requirements An...	1/23/22	1/27/22
Use Cases	1/28/22	1/31/22
Traceability Matrix	2/1/22	2/3/22
System Sequenc...	2/4/22	2/7/22
System Operatio...	2/8/22	2/12/22
Wireframes	2/13/22	2/18/22
Domain Model	2/19/22	2/28/22
Presentation and...	3/1/22	3/1/22
Design	3/2/22	3/29/22
Design Model	3/2/22	3/4/22
Sequence Diagra...	3/5/22	3/9/22
Package Diagrams	3/10/22	3/14/22
GRAPS patterns	3/15/22	3/19/22
Test Coverage	3/20/22	3/24/22
Prototyping	3/25/22	3/28/22
Presentation and...	3/29/22	3/29/22
Implementation	3/30/22	4/27/22
Backend	3/30/22	4/4/22
User Interface	4/5/22	4/9/22
User Input Valida...	4/10/22	4/15/22
Imports/Exports	4/16/22	4/19/22
Unit-testing	4/20/22	4/23/22
Documentation	4/24/22	4/26/22
Presentation and...	4/27/22	4/27/22

Resources Chart



Name	Default role
Analyst	undefined
Developer	undefined
Tester	undefined
Team Lead	undefined

External Links:

Github:
Github

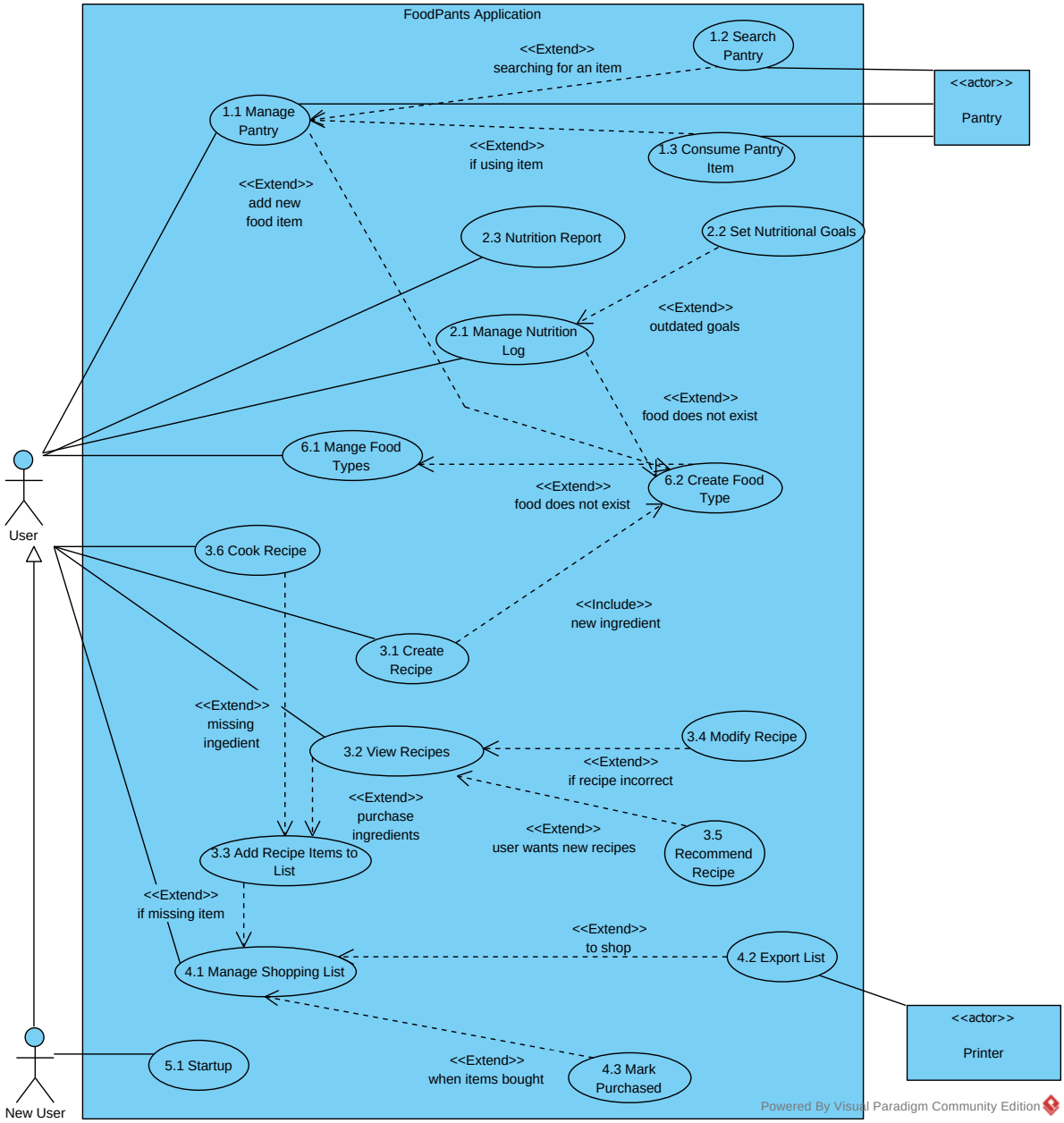
Requirements:
Product Backlog (Trello Requirements)

Progress:
Progress and Hours Report

Live Google Drive Files:
Analysis Files

Issue Tracking:
Issue Tracking

Website:
Website


FoodPants Application
1.2 Search Pantry
<<Extend>>
searching for an item
<<actor>>
Pantry
1.1 Manage Pantry
<<Extend>>
if using item
1.3 Consume Pantry Item
<<Extend>>
add new food item
2.3 Nutrition Report
2.2 Set Nutritional Goals
<<Extend>>
outdated goals
2.1 Manage Nutrition Log
<<Extend>>
food does not exist
6.1 Mange Food Types
6.2 Create Food Type
<<Extend>>
food does not exist
User
3.6 Cook Recipe
<<Include>>
new ingredient
3.1 Create Recipe
<<Extend>>
missing ingedient
3.4 Modify Recipe
3.2 View Recipes
<<Extend>>
if recipe incorrect
<<Extend>>
purchase ingredients
<<Extend>>
user wants new recipes
3.5 Recommend Recipe
3.3 Add Recipe Items to List
<<Extend>>
if missing item
<<Extend>>
to shop
4.2 Export List
4.1 Manage Shopping List
<<actor>>
Printer
5.1 Startup
<<Extend>>
when items bought
4.3 Mark Purchased
New User
Powered By Visual Paradigm Community Edition