Design Use Cases

Pantry Raider

Supreme Meals for Heros (SMFH)

Project Manager: Steve Tran
Business Analyst: Erxi Li (Lex)
Senior System Analyst: Xiaocheng Huang (Frank)
Software Architect: Mingqi Zhu
Software Development Lead: William Ma
Algorithm Specialist: Randall Lu
Database Specialist: Ajay Gopinath
Quality Assurance Lead: Cho-Yuan Su (Alex)
User Interface Specialist: Collin Kawahara
User Interface Specialist: Ruoxin Huang (Angelo)

Table of Contents

Priority Key:
1 - High

2 - Medium
3 - Low
4 - Extra, if time permits

Status/TestStatus:
Planned, In-Progress, Completed, Cancelled

1. Accounts

1.1 [A1] User Login

Description	The user will be able to log in.
Desired Outcome	The user will successfully log in.
User Goal	The user wants to log in to use the application.
Dependent Use Cases	[A3] User Registration
Priority	1
Status	Completed
Requirements	The system shall verify account login credentials.
Pre-Conditions	1. The user has a registered account. 2. The user knows their credentials.
Post-Conditions	The user can access the application.
Trigger	The user wants to use the application
Workflow	1. The frontend shall render the login screen. 2. The user shall type their email and password. 3. The frontend shall determine the inputs are valid and login with the these inputs. 4. The backend should verify that the user's credentials are valid and login the user. 4. The frontend shall redirect the user to the home page.
Alternate Workflow	3. The backend shall check that the email and password are invalid. 4. The frontend shall notify the user that the credentials are invalid. 5. Return to step 2 of the workflow.

1.2 [A2] User Logout

Description	The user will be able to log out.
Desired Outcome	The user will successfully logout, making their information and data no longer accessible until log in.
User Goal	The user wants to logout of the application.
Dependent Use Cases	[A1] User Login [A3] User Registration
Priority	1
Status	Completed
Requirements	The system shall allow the user to logout, securing their information and data.
Pre-Conditions	1. The user is logged into their account.
Post-Conditions	The user is logged out of their account and their data can no longer be accessed.
Trigger	The user desires to be logged out of the application.
Workflow	1. The user shall navigate to the settings page. 2. The user shall click the “SIGN OUT” button. 2. The backend shall nullify the authentication token generated by Firebase. 3. The frontend shall render the login page.
Alternate Workflow	N/A

1.3 [A3] User Registration

Description	The user will be able to register their account.
Desired Outcome	The user will successfully register and have their account and password in order to login.
User Goal	The user wants to register an account for the application.
Dependent Use Cases	None
Priority	1
Status	Completed
Requirements	The system shall allow the user to register and store their account and password.
Pre-Conditions	1. The user does not have an account to log in.
Post-Conditions	The system shall store the user's account and password.
Trigger	The user wants to have an account to use the application.
Workflow	1. The frontend shall render the registration screen. 2. The user shall type in their name, email, password, and confirmation password. 3. The frontend shall determine that the password and confirm password are correct. 4.The backend shall determine the email has not been registered. 5.The frontend shall redirect the user to the login page.
Alternate Workflow	3.The backend shall determine the email has been registered. 4.The frontend shall notify the user that the email has been registered, and suggest user to go to "Reset Password" page. 5. Return to step 2 of the workflow.

1.4 [A4] Reset Password

Description	The user will be able to reset their password.
Desired Outcome	The user will have successfully reset their password.
User Goal	The user wants to reset their password.
Dependent Use Cases	[A3] User Registration
Priority	3
Status	Completed
Requirements	The system shall allow the user to reset their password.
Pre-Conditions	1. The user has an account.
Post-Conditions	The user changes their password.
Trigger	The user desires to reset password.
Workflow	1. The frontend shall render the account login screen. 2. The user shall click the “Reset Password” button. 3. The frontend shall prompt the user for their email. 4. The user shall enter their email and press the “Reset Password” button. 5. The system shall verify the email is stored in the backend. 6. The system shall email the user with a verification code. 7. The user shall access their email for the verification code. 8. The frontend shall render one field for the user to enter the verification code. 9. The user shall enter the verification code correctly and presses the “Next” button. 10. The frontend shall render two fields for the user to enter their new password twice. 11. The system shall check if the two passwords are the same. 12. The user shall click the confirm button. 13. The backend shall replace the user’s old password with their new password.
Alternate Workflow (Invalid Email)	5. The system determines the email is not stored in the backend. 6. The frontend shall display an “invalid email” notification. 7. Return to step 4 of the workflow.
Alternate Workflow (Invalid Verification Code)	9. The user shall enter the verification code incorrectly and presses the “Next” button.

	10. The frontend shall display an “invalid verification code” notification. 11. Return to step 9 of the workflow.
Alternate Workflow (Non-matching Passwords)	11. The user enters two passwords that are not the same. 12. The frontend shall display a “passwords don’t match” notification.’ 13. Return to step 10 of the workflow.

1.5 [A5] Change Password

Description	The user will be able to change their password.
Desired Outcome	The user successfully changes their password.
User Goal	The user wants to change their password.
Dependent Use Cases	[A1] User Login
Priority	3
Status	Completed
Requirements	The system shall allow a user to change their password.
Pre-Conditions	1. The user has an account. 2. The user is logged in.
Post-Conditions	The user has changed their password.
Trigger	The user desires to change their password.
Workflow	1. The user shall navigate to the settings page. 2. The frontend shall render the settings screen. 3. The user shall click the change password button. 4. The frontend shall render three fields for the user to enter their old password once and their new password twice. 5. The user shall fill the password fields. 6. The backend shall verify the old password is correct. 7. The frontend shall verify the two new passwords entered are the same. 8. The user shall click the “Change Password” button. 9. The frontend shall send the new password to the backend. 10. The backend shall update the password for the specific user. 11. The backend shall log the user out of the app. 12. The frontend shall render the login page.
Alternate Workflow (Invalid Password)	6. The backend shall verify the old password is incorrect. 7. The frontend shall display an “incorrect old password” notification. 8. Return to step 5 of the workflow.
Alternate Workflow (Non-Matching Passwords)	5. The user enters two passwords that are not the same. 6. The frontend shall display a “passwords don’t match” Notification. 7. The frontend shall clear the new password and confirm password fields.

	8. Return to step 5 of the workflow.

1.6 [A6] Set dietary preferences

Description	The user will be able to set their dietary preferences.
Desired Outcome	The user successfully changed their dietary preferences.
User Goal	The user wants to receive recipes that match their dietary preferences.
Dependent Use Cases	[A1] User Login
Priority	1
Status	Completed
Requirements	The system shall allow user to set up dietary preferences.
Pre-Conditions	1. The user has an account and has logged in.
Post-Conditions	The user has set their dietary preferences.
Trigger	The user desires to set their dietary preferences.
Workflow	1. The user goes to the settings section. 2. The user clicks on the “Set Dietary Preferences” button. 3. The frontend shall render the dietary preference screen. 4. The user shall update their dietary preferences. 5. The user clicks the “update” button. 6. The system shall update the backend with the data of the preferences the user selected. 7. The backend shall update the user’s dietary preferences. 8. The frontend displays a “Preferences successfully updated!” notification.
Alternate Workflow	N/A

1.7 [A7] Change themes

Description	The user will be able to change the theme of the app
Desired Outcome	The colors of the background and buttons will changed

	based on the theme that the user choose
User Goal	The user wants to use different color combinations for the app interface
Dependent Use Cases	[A1] User Login
Priority	3
Status	Cancelled
Requirements	The system shall allow user to change the theme of the interface.
Pre-Conditions	1. The user is logged in
Post-Conditions	The colors of the background and buttons are changed with the user's desired theme
Trigger	The user wants to try out different theme for the app interface
Workflow	1. The user navigates to the setting section 2. The user clicks on "change theme" button 3. The frontend shall redirect the user to the screen where different themes are available for user to choose 4. The user shall click on the desired theme 5. The frontend shall send a POST request to the backend with the data of the theme the user selected 6. The backend receives the request and shall return back with the corresponding theme data in JSON format 7. The frontend receives the return data and shall change the appearance of the interface accordingly 8. The user clicks the "save" button 9. The frontend shall send a POST request to the backend with the theme setting that the user just selected 10. The backend shall receive the data and shall update this user's setting profile
Alternate Workflow	8. The user clicks on other theme 9. Loop back to 5

2. Pantry

2.1 [P1] Search ingredients

Description	The user will be able to search for ingredients from the database of the app so they can add the searched ingredients to their pantry.
Desired Outcome	The user successfully finds the ingredient they are searching for.
User Goal	The user wants to find ingredients to they own so they can add them to the pantry in the app.
Dependent Use Cases	[A1] User Login
Priority	1
Status	Completed
Requirements	The system shall list the ingredients based on the keywords the user put in the search box.
Pre-Conditions	1. The user has an account and logged into their account.
Post-Conditions	The user will have found the ingredient they are searching for or the system shall display “no such ingredient found.”
Trigger	The user wants to search for ingredients they own in real life so they can add them to their pantry.
Workflow	1. The user shall navigate to the My Ingredients screen from the pantry page. 2. The user shall click the cart with a plus button. 3. The frontend shall render a dialog with a search field. 4. The user shall enter a search query. 5. The frontend shall autocomplete the search query and show a list of matches. 6. The user shall click ingredient name. 7. The backend shall add the ingredient id and name to the personal FireBase account child. 8. The My Ingredients screen shall update the screen with a current list of ingredients.
Alternate Workflow	5. The frontend shall display no results if no ingredients are returned. 6. Return to step 4 of the workflow.

2.2 [P2] View ingredients

Description	Display the list of all ingredients that the user has.
Desired Outcome	A list of all available ingredients with pictures will be shown on the screen.
User Goal	The user wants to see their current owned ingredients.
Dependent Use Cases	[A1] User Login [P3] Add ingredients
Priority	2
Status	Completed
Requirements	The system shall display all ingredients that the user has.
Pre-Conditions	1. The user has an account and logged in successfully.
Post-Conditions	A list of all available ingredients with pictures will be shown on the screen.
Trigger	The user wants to see their current owned ingredients.
Workflow	1. The user shall navigate to pantry. 2. The user shall hit the view ingredients button. 3. The frontend shall connect to backend and ask for the pantry data of this user. 3. The backend shall return data of all saved ingredients by this user. 4. The frontend shall display all the ingredients currently the user owns.
Alternate Workflow	N/A

2.3 [P3] Add ingredients

Description	The user will be able to add any ingredients they successfully searched to their pantry
Desired Outcome	The ingredient the user added is now in the user`s pantry.
User Goal	The user wants to add ingredients to their pantry so that they can search for recipes based on their pantry
Dependent Use Cases	[A1] User Login [P1] Search for ingredients
Priority	1
Status	Completed
Requirements	The system shall allow a user to add ingredients to their pantry.
Pre-Conditions	1. The user has an account and logged into their account. 2. The user successfully searched for an ingredient that they wish to add to their pantry.
Post-Conditions	The ingredient will be added to the pantry.
Trigger	The user wants to add an ingredient they own in real life to their pantry.
Workflow	1. The user successfully searched for an ingredient. 2. The user clicks “add to pantry” button. 3. The frontend shall send the information to the backend. 4. The backend shall update the user’s pantry profile with the ingredient selected and return with success to the frontend. 5. The frontend shall display “Added successfully” text next to the button.
Alternate Workflow	N/A

2.4 [P4] Delete ingredients

Description	The user will be able to remove any ingredients from their pantry.

Desired Outcome	The user has removed the ingredient from their pantry.
User Goal	The user want to delete the ingredients that are not available from the pantry.
Dependent Use Cases	[P1] Search ingredients [P2] View ingredients [P3] Add ingredients
Priority	1
Status	Completed
Requirements	The system shall allow the user to delete any ingredients they have added in the past.
Pre-Conditions	1. The user has added ingredients to their pantry.
Post-Conditions	The user will have deleted specified ingredients.
Trigger	The user wants to delete certain ingredients in their possession, affecting their “Search By Pantry” recipe search.
Workflow	1. The user shall navigate to their pantry using the bottom navigation bar. 2. The frontend shall send a request to the backend for the user’s stored ingredients. 3. The backend should return with the user’s stored ingredients in JSON format. 3. The frontend shall display the ingredients retrieved from the backend. 4. The user shall click on an ingredient. 5. The frontend shall render a screen with the ingredient name, a “cancel” button, and “delete” button. 6. The user shall click the “delete” button. 7. The frontend shall send a request to the backend with the ingredient the user just selected. 8. The backend shall update the user’s pantry profile based on the data and return with success to to the frontend. 9. The frontend shall display the modified list of ingredients.
Alternate Workflow	6. The user shall click the “cancel” button. 7. The frontend removes the dialog and returns to the My Ingredients screen.

3. Recipe

3.1 [R1] Explore recipes

Description	The user will be able to see some random recipes
Desired Outcome	The user can see random recipes
User Goal	The user wants to see some recipes
Dependent Use Cases	[A1] User Login
Priority	1
Status	Completed
Requirements	The system shall list recipes based on current pantry.
Pre-Conditions	1. The user has an account and has logged in.
Post-Conditions	The user will see a list of random recipes
Trigger	The user desires a list of recipes that are based on their existing pantry.
Workflow	1. The frontend shall render the home page. 2. The fronted shall make a POST request to the backend at api/explore 3. The backend shall return a list of randomly selected recipes 2. The frontend shall render random recipes on the home page
Alternate Workflow	N/A

3.2 [R2] Search recipes by pantry

Description	The user will be able to search recipes based on the items in the pantry.
Desired Outcome	The user sees the recipes based on the items in the pantry.
User Goal	User wants to see the recipes that involves only what they currently have.
Dependent Use Cases	[A1] User Login [P3] Add ingredients
Priority	1
Status	Completed
Requirements	The system shall list all the recipes based on the pantry items.
Pre-Conditions	1. The user has an account and has logged in. 2. The user has added ingredients to their pantry.
Post-Conditions	The user will receive a list of recipes based on the pantry items..
Trigger	The users desires a list of recipes of what they can make with current pantry ingredients.
Workflow	1. The frontend shall render the home page. 2. The user shall go to the search tab. 3. The frontend shall render the search page. 4. The user shall click the “ADD FILTER” button and select “Search By Pantry”. 5. The user shall click the “DONE” button. 6 The user shall click the “SEARCH” button. 7. The frontend shall send the POST request to the backend at api/search_recipes. 8. The backend shall send requests to Food API to get results, and returns them to frontend. 9. The frontend shall render the results to the user.
Alternate Workflow	N/A

3.3 [R3] Search recipes by name

Description	The user will be able to search recipes by the name.
Desired Outcome	The user receives recipes that match the name.
User Goal	The user desires to search recipes by name.
Dependent Use Cases	[A1] User Login
Priority	1
Status	Completed
Requirements	The system shall list the recipes based on the keywords the user put in the search box.
Pre-Conditions	1. The user has an account and has logged in.
Post-Conditions	The user will receive a list of recipes.
Trigger	The user desires a list of recipes that match the name.
Workflow	1. The user shall navigate to the search page. 2. The frontend shall render the search page. 3. The user shall enter their desired search query. 4. The frontend shall send the query to the backend at api/search_recipes. 5. The backend shall send a request to Food API to get a list of matching recipes and returns them to the frontend. 6. The frontend shall render the results to the user.
Alternate Workflow	N/A

3.4 [R4] Filter recipes

Description	The user shall be able to set filters to receive specific recipes.
Desired Outcome	The user receives a list of filtered recipes.
User Goal	The user desires to receive filtered recipes.
Dependent Use Cases	[A1] User Login [R3] Search Recipes by name
Priority	2
Status	Completed
Requirements	The system shall display a list of recipes based on the search key and filters chosen.
Pre-Conditions	1. The user has an account and is logged in. 2. The user shall be in the search screen and have entered keywords in the search bar.
Post-Conditions	The list of recipes received fit the filters and the search keywords.
Trigger	The user wants receive recipes that fit their specifications.
Workflow	1. The user shall login. 2. The frontend shall render the home screen. 3. The user shall navigate to the search screen. 4. The frontend shall render the search screen. 5. The user shall type in the desired search keywords. 6. The user shall click the “ADD FILTERS” button. 7. The frontend shall render a pop-up screen that has a list of filters. 8. The user shall select their desired filters. 9. The user shall click the “DONE” button. 10. The frontend shall go back to the search screen. 11. The user shall click the “SEARCH” button. 12. The frontend shall send the search keywords and filters in JSON format to the backend at api/search_recipes. 13. The backend shall make a request to the food API with the search keywords and filters. 14. The backend shall receive a list of recipes that fits the criteria. 15. The backend shall send a list of recipes to the frontend. 16. The frontend shall render the list of recipes.
Alternate Workflow	N/A

3.5 [R5] View selected recipe

Description	The user will be able to view the details of a recipe including used ingredients, directions, nutrition value by clicking on one of the recipes in the search result.
Desired Outcome	The user can view the details of the recipe selected
User Goal	The user wants to view details of a particular recipe from

	their search results.
Dependent Use Cases	[A1] User Login [R3] Search Recipes by name
Priority	1
Status	Completed
Requirements	N/A
Pre-Conditions	1. The user has an account and is logged into their account. 2. The user has performed a search on recipes.
Post-Conditions	The user is shown a page containing all relevant information about the recipe.
Trigger	The user wants to view details of a recipe from the search results.
Workflow	1. The user shall select the recipe they want to view from the search result list. 2. The frontend shall request the recipe information from the backend at api/recipe_detail. 3. The backend shall send a request for the detail of the recipe to the Food API. 4. The backend shall return the result to the frontend. 5. The frontend receives the information and renders it on screen in a new page containing all information about the recipe.
Alternate Workflow	N/A

3.6 [R6] Save recipes

Description	The user will be able to save their favourite recipes.
Desired Outcome	The user successfully saves the recipe, and the recipe will be added to user' personal list.
User Goal	The user wants to save the recipes they liked, so they can view after.
Dependent Use Cases	[A1] User Login [R6] View Selected Recipe

Priority	2
Status	Completed
Requirements	The system shall allow a user to save their favourite recipe in order to use the same recipe later.
Pre-Conditions	1. The user has an account and logged into their accounts. 2. The user has selected a recipe to view.
Post-Conditions	The user will have this recipe in their saved recipe list.
Trigger	The user wants to save the recipes in order to use the recipe again.
Workflow	1. The user shall click on the save button on the recipe page. 2. The frontend shall send the information to the backend. 3. The backend shall add the reference to recipe to the user's saved recipes profile. 4. The backend shall send a success message to the frontend. 5. The frontend shall display a "Recipe saved!" message.
Alternate Workflow	N/A

3.7 [R7] View saved recipes

Description	The user will be able to see saved recipes.
Desired Outcome	The user will be displayed all the recipes they have saved..
User Goal	The user wants to see the recipes they saved.
Dependent Use Cases	[A1] User Login [R6] View Selected Recipe [R7] Save Recipes
Priority	2
Status	Completed
Requirements	The system shall allow the user to see saved recipes.
Pre-Conditions	1. The user has logged in and saved recipes.

Post-Conditions	The saved recipes will be displayed.
Trigger	The user wants to see recipes they've saved.
Workflow	1. The user shall navigate to the "Pantry" page. 2. The frontend shall render the Pantry page. 3. The user shall click the "SAVED RECIPES" button. 4. The frontend shall send a POST request to the backend at api/get_recipes_bulk. 5. The backend shall return a list of recipes that the user saved. 6. The frontend shall render the list of recipes to the user.
Alternate Workflow	N/A

3.8 [R8] Remove saved recipes

Description	The user will be able to remove a recipe that currently exists in their list of personal recipes
Desired Outcome	The user successfully removes the recipe from the user's list of personal recipes, and the list no longer displays the removed recipe
User Goal	The user wants to remove the recipes they no longer need from their list of personal recipes
Dependent Use Cases	[A1] User Login [R6] View selected recipe [R7] Save recipes
Priority	2
Status	Completed
Requirements	The system shall allow a user to remove their saved recipes.
Pre-Conditions	1. The user has an account and logged into their accounts. 2. The user has a saved recipe in their list of saved recipe that they wish to remove.

Post-Conditions	The user will no longer have this recipe in their saved recipe list.
Trigger	The user wants to remove the recipes that they no longer need from their list of saved recipes
Workflow	1. The user shall navigate to the Pantry page. 2. The frontend shall render the Pantry page. 3. The user shall click the “SAVED RECIPES” button. 4. The backend shall retrieve the list of saved recipes from the database. 5. The frontend shall render the Saved Recipes page with a list of previously saved recipes. 6. The user shall click on the recipe they wish to delete. 7. The backend shall retrieve the data from the recipe. 8. The frontend shall render the recipe data. 9. The user shall click the “UNSAVE RECIPE” button. 10. The backend shall send a request to remove the recipe from the user’s database.
Alternate Workflow	N/A