Functional & Non-Functional Requirements

1. Functional Requirements

1.1. Registration

1. The user must be able to register for an account on the system by clicking the ‘New user’ button in the ‘Landing’ page.
    1.1. The system must display text fields for the user to enter his information in the “Register” page.
        1.1.1. The text fields must consist of username.
        1.1.2. The text fields must consist of email addresses.
        1.1.3. The text fields must consist of passwords.
    1.2. The user must fill in all the fields before clicking the ‘Register’ button.
        1.2.1. If the fields are not filled in and the “Register” button is clicked, an error message, “Please fill in this field.” must be displayed.
    1.3. The system must verify the fields filled in by the user before creating an account.
        1.3.1. The username entered must be unique and never been registered in the system.
            1.3.1.1. If the username entered has been registered in the system, an error message, “Registration failed. Username or email already exists” must be displayed.
        1.3.2. The password entered must contain at least 8 characters and at least 1 uppercase character and 1 special character.
            1.3.2.1. If the password entered does not meet the criterias, an error message, “Your password must contain between 8 to 16 characters, at least 1 uppercase character and at least 1 special character.” must be displayed.
    1.4. The system must create an account for the user upon verification.
2. The user must be able to create a profile after registering an account.
    2.1. The system must display text fields for the user to enter his information in the “Register” page.
        2.1.1. The text fields must consist of the user's height, at least 1 cm.
        2.1.2. The text fields must consist of the user's weight, at least 1 kg.
        2.1.3. The text fields must consist of the user's age, at least 1 years old.
    2.2. The system must display choice fields allowing the user to select from predefined options.
        2.2.1. The choice fields must include gender categories (Male, Female).

2.2.2. The choice fields must include activity level categories (Sedentary, Lightly active, Moderately active, Very active, Super active)
2.2.3. The choice fields must include diet type categories (Vegetarian, Vegan, Gluten free).
2.2.4. The choice fields must include allergy categories (dairy, egg, gluten, grain, peanut, seafood, sesame, shellfish, soy, sulfite, tree nut, wheat).

2.3. The user must fill in all the fields before clicking the 'Next' button.
2.3.1. If the fields are not filled in and the "Next" button is clicked, an error message, "Please fill in the blank fields." must be displayed.

2.4. When clicking the 'Next' button, the system must display suggested daily estimated nutrition intake for the user.
2.4.1. The system must calculate the user's basal metabolic rate (BMR).
2.4.2. The system must calculate the user's total daily energy expenditure (TDEE) which is the estimated daily calories intake, displayed in kcal/day and rounded off to the nearest whole number .
2.4.3. The system must calculate the user's estimated daily protein intake and display it in grams/day, rounded off to the nearest whole number.
2.4.4. The system must calculate the user's estimated daily fats intake and display in grams/day, rounded off to the nearest whole number.
2.4.5. The system must calculate the user's estimated daily carbohydrates intake and display it in grams/day, rounded off to the nearest whole number.

2.5. When the "Next" button is clicked, the system must display the "My Plan" page.

1.2. Login

1. The user must be able to log into the system by clicking the 'Existing user' button in the 'Landing' page.
1.1. The system must display text fields for the user to enter his information in the "Login" page.
1.1.1. The text fields must consist of username.
1.1.2. The text fields must consist of password.

1.2. The user must input the login credentials (username and password) in the corresponding text fields before clicking "Login" button

1.3. The system must verify the login credentials input by the user.
1.3.1. The system must find the username in the database of the system.
1.3.2. The system must ensure that the password entered matches the password associated with the specified username.

1.4. If the username entered is found in the database of the system and the password entered matches the password associated with the specified username, the login credentials are verified.
1.4.1. The system must log the user into the "My Plan" page of the system.

1.5. If the username entered is not found in the database of the system, the login credentials are not verified.
1.5.1. The system must display the message "Login failed: Invalid username or password."

1.6. If the username entered is found in the database of the system but the password entered does not match the password associated with the username, the login credentials are not verified.
1.6.1. The system must display the message “Login failed: Invalid username or password.”
1.7. The user must be allowed to return to the ‘Landing’ page.

1.3. My Plan

1.3.1 Calendar

1. The system displays a personalised greeting (e.g., "Hello [Username]") at the top of the page.
2. The system will provide an interactive weekly calendar for users to select dates easily.
2.1. The user must be able to toggle between weeks by clicking on the left and right arrows.
2.2. The user must be able to switch between days within the selected week by clicking on the date.
2.2.1. The system must display the 4 recipes planned for each course (breakfast, lunch, dinner and snack).
2.2.1.1. Each planned recipe must display basic information of the recipe.
2.2.1.1.1. The meal card must display the image of the meal.
2.2.1.1.2. The meal card must display the name of the meal.
2.2.1.1.3. The meal card must display the cost of the meal.
2.2.1.1.4. The meal card must display the cuisine of the meal.
2.2.1.2. The system must display a tick icon.
2.2.1.2.1. When the user completes the meal, the user can click the tick icon to represent completion.
2.2.1.3. The system must display a cross icon.
2.2.1.3.1. When the cross icon is clicked, the system will remove the recipe from that course of that particular day.
2.2.1.3.2. When the cross icon is clicked, the system will remove the ingredients needed for the recipe from the grocery list
2.2.1.4. If no recipes are planned for a course, the message, ‘No course planned’ and an ‘Add recipe’ button will be displayed.
2.2.1.5. When the ‘Add recipe’ button is clicked, the system will direct the user to the ‘Discover’ page.

1.3.2. Statistics

1. The system must display the overall nutrition statistics of a particular day.
1.1. The system must display the total accumulated values of each nutrient from all the meals planned throughout the day.
1.1.1. The system must calculate the total accumulated calories intake from all the meals of that day and display in kcal, rounded to the nearest whole number.
1.1.2. The system must calculate the total accumulated protein intake from all the meals of that day and display in grams, rounded to the nearest whole number.

1.1.3. The system must calculate the total accumulated fats intake from all the meals of that day and display in grams, rounded to the nearest whole number.
1.1.4. The system must calculate the total accumulated carbohydrate intake from all the meals of that day and display in grams, rounded to the nearest whole number.

1.2. The system must display pie charts for each nutrient, showing the accumulated daily values and the progress toward the recommended intake. The percentage of the pie chart filled represents this progress.
1.2.1. The pie chart must display the progress of reaching the suggested intake for each nutrition in percentages to the nearest whole number.
1.2.2. The pie chart must have different designated colours for each nutrition to visually distinguish them.

1.4. Discover

1. The system must display a list of recipes that matches the user's profile.
1.1. The system calls the recipeAPI and obtains the list of recipes.
1.2. Each recipe in the "Discover" page must display basic information about the recipe.
1.2.1. The meal card should display the image of the meal.
1.2.2. The meal card should display the name of the meal.
1.2.3. The meal card should display the cost of the meal.
1.2.4. The meal card should display the estimated preparation time of the meal.
1.2.5. The meal card should display the cuisine of the meal.
2. The system must display a list of filters.
2.1. The list of filters shall consist of the cost filter.
2.1.1. The system displays a text field for the user to enter the minimum and maximum price, in dollars.
2.1.2. Recipes that fall within that specified price range will be displayed.
2.2. The list of filters shall consist of the cuisine filter.
2.2.1. The system displays a choice field of cuisines available.
2.2.2. Recipes matching the cuisines selected will be displayed.
2.3. The list of filters shall consist of the estimated preparation time filter.
2.3.1. The system displays a text field for the user to enter the minimum and maximum estimated preparation time, in minutes.
2.3.2. Recipes that fall within that specified estimated preparation time range will be displayed.
2.4. The list of filters shall consist of the course filter.
2.4.1. The system displays a choice field of courses.
2.4.2. Recipes matching the courses selected will be displayed.

1.5. Recipe

1. When clicking on the recipe's image, the system will display the recipe's page.
1.1. The recipe's page must display detailed information about the recipe.
1.1.1. The page must display the image of the recipe
1.1.2. The page must display the name of the recipe
1.1.3. The page must display the cost of the recipe

1.1.4. The page must display the course of the recipe
1.1.5. The page must display the estimated preparation time.
1.1.6. The page must display the ingredients needed for the recipe
1.1.7. The page must display the nutrition of the recipe
1.1.8. The page must display the link to a website with instructions.

1.2. The recipe page must have a “Add to Meal Plan” button to add the recipe to the user’s meal plan.
1.2.1. When the button is clicked, the system will display a popup.
1.2.1.1. The popup consists of a choice field that prompts users to input the date the meal will be prepared.
1.2.1.2. The popup consist of a choice field that prompts users to input the course (breakfast, lunch, dinner, snack)
1.2.1.3. When users click on the ‘Confirm’ button, the recipe will be added to the meal plan for the specific date and course selected.
1.2.1.4. If the date and course selected already has a meal planned, the system will display the message, “Meal Plan for this date and course exists!”
1.2.1.5. If the date is not selected, the system will display the message, “Please select a date”
1.2.1.6. If the course is not selected, the system will display the message, “Please select a course.”

1.6. Grocery List

1. The system must display the grocery list of a particular week, displaying all ingredients needed to make all the meals for that particular week.
1.1. The system must display the name of the ingredient.
1.2. The system must display the quantity required for each ingredient.
1.3. The system must display the prices of each ingredient.
1.3.1. The system calls recipe API to fetch the price of each ingredient.
1.3.2. The price of each ingredient in the grocery list must be displayed in SGD, to 2 decimal places of accuracy.
1.4. The system must display the total cost of the grocery list for a particular week.
1.4.1. The system calculates the total price by summing up all the prices of the ingredients.
1.4.2. The total price of the grocery list for a particular week must be displayed in SGD, to 2 decimal places of accuracy.
1.5. The user can strike through an ingredient from the grocery list by clicking on the text of the ingredient to mark them as purchased.
1.5.1. The system recalculates the total cost of all ingredients.
1.5.2. The system displays the new total cost.

1.7. History

1. The user must be able to view a list of past recipes.
1.1. The system must display the list of recipes the user has previously used.
1.1.1. The meal card must display the recipe name.
1.1.2. The meal card must display the dates the recipes were used.
1.1.3. The meal card must display the cuisine of the recipe.

1.1.4. The meal card must display the images of the recipe.
1.1.5. The meal card must display the estimated prep time of the recipe.
1.1.6. The meal card must display the cost of the recipes, in SGD, displayed to 2 decimal places of accuracy.

1.2. The user must be able to filter past recipes by selecting a specific date.
1.2.1. The system displays past recipes used on the specific date

1.3. The user must be able to search for a specific recipe.
1.3.1. The system must provide a text field search bar for users to enter keywords related to the recipe.
1.3.2. The search criteria text must be minimally 1 character long.
1.3.3. The search criteria text must be maximally 50 characters long.
1.3.4. The search criteria can contain any ASCII recognised characters.
1.3.5. The system must return recipes that match the search criteria.

1.4. When the user clicks on the image of the recipe, the system will display the recipe's page.

2. Non-Functional Requirements

2.1. Security Requirements

- Emails given for user accounts are not shared externally.
- Each user account will have a unique username.
- Personal data should not be shared with third parties.
- SQL injection preventive measures must be in place.

2.2. Software Quality Attributes

2.2.1. Performance

- The system should be able to handle a minimum of 100 concurrent users at a time allowing them to use any functionalities of the application
- The application should be able to respond to any user's request within 10 seconds
- The system should be able to scale up to 1000 users concurrently
- The system should be available 24/7 with no more than 1 day of downtime per month

2.2.2. Reliability

- The system shall perform a complete backup procedure every 48 hours, ensuring all data can be recovered in the event of system failure
- Maximum allowed time for data recovery shall not exceed 30 minutes
- Automated system health checks will automatically detect and address software issues

2.2.3. Usability

- The system UI shall be designed such that a minimum of 90% of first time users can navigate from the home page to the intended feature within 5 minutes of first visiting the website.

2.2.4. Supportability

- The system should log errors together with performance metrics and user activity to allow future analysis and troubleshooting.

- System design and implementation should be modular to allow individual components to be serviced without affecting the entire system