

Software Design Document

Project Name: Nutrition Information Data Analysis Tool

Group Number: 027

Team members

| Student Number | Name |
|----------------|-----------------|
| s5113175 | Peter Symoniw |
| s5094725 | Sean O'Sullivan |
| s5291335 | Sipa Sunuwar |

Table of Contents

- Software Design Document
 - Project Name: Nutrition Information Data Analysis Tool
 - Group Number: 027
 - Team members
- Table of Contents
 - 1. System Vision
 - 1.1 Problem Background
 - 1.2 System capabilities/overview
 - 1.3 Benefit Analysis
 - 2. Requirements
 - 2.1 User Requirements
 - Core User Requirements
 - Fictional User Profile:
 - 2.2 Software Requirements
 - Functional Requirements
 - Non-Functional Requirements
 - 2.3 Use Case Diagram
 - 2.4 Use Cases
 - 3. Software Design and System Components
 - 3.1 Software Design
 - 3.2 System Components
 - 3.2.1 Functions
 - 3.2.2 Data Structures / Data Sources
 - 3.2.3 Detailed Design
 - 3.2.4 Test Entry
 - 4. User Interface Design
 - 4.1 Structural Design
 - 4.2 Visual Design
 - 4.3 General UI Considerations

1. System Vision

1.1 Problem Background

- Problem Identification: What problem does this system solve?
- Dataset: What is the dataset used?
- Data Input/Output: What kind of data input and output is required?
- Target Users: Who will use the system, and why?

The nutrition visualiser software aims to enhance end users health and eating habits by allowing them to see and visualise the nutrients of foods they consume. It will give them all the important information they need to know in a easily digestible manner.

The dataset being utilised for the application is about food nutrition in a comma separated value (CSV) format. The values present are different items of food, and the macronutrients and micronutrients of the respective food items.

The user will be able to interact with the application through its GUI. They will be able to enter strings into the nutrition visualiser when searching for specific food items and nutrient values. The software will deliver the searched items nutrition composition in a readable format with models of the data being presented to help understanding.

There is a range of target users for the nutrition visualiser software, depending on the users nutritional goals. They can be infrequent users that use the application to look up specific food items occasionally to get a quick breakdown of its nutritional value. Additionally, they could be frequent users that enter all the food items they want to consume into the nutrition visualiser application. The software will allow them to easily keep track of their total nutrition consumption.

1.2 System capabilities/overview

- System Functionality: What will the system do?
- Features and Functionalities: Describe the key features and functionalities of the system.

The nutrition visualiser application will allow for quick and orderly retrieval of information regarding foods nutritional composition. The food nutrition dataset csv will be used as the database that the user will be able to interact with as part of the software.

The user will be able to search for specific food items and their corresponding nutritional values by typing into a search bar. It will present the nutritional information in written text as well as through various models of important values, depending on the user's selection. Additionally, the user will be able to filter through foods depending on their nutritional composition. All features and functions will be accessible through the integrated GUI.

1.3 Benefit Analysis

How will this system provide value or benefit?

This system will be practical and provide great benefits to the user, by helping them make more informed decisions related to their eating habits. Correspondingly, this will aid users health and make it easier for them to improve, due to the quantifiable nature of the application. For example a user could be trying to lose weight, and they want to figure out the composition of their diet. They can search the database and get concrete numbers on their current situation. Then by using the search and filtering functions the user will be able to find food items that provide better macro and micronutrients, and take actionable steps to improving their diet.

2. Requirements

2.1 User Requirements

Users of the system are likely to be individuals who are health-conscious and interested in managing their diet by making informed food choices based on nutritional information. The system must be user-friendly, offering simple navigation and clear presentation of nutritional data.

Core User Requirements

1. Food search

- Users enter a food item into search bar and system will display the results to match the search query. Each result will display their nutritional information like caloric value, fat content, etc.
- System should be able to present search results using general or broad terms.
- Example:
 - User searches cheese.
 - System displays all food items with 'cheese' like goat cheese, cheddar cheese, cream cheese.
 - User can select the specific food item.
 - Nutritional information of selected item is displayed.

2. Nutrition breakdown

- System should display detailed nutritional information for food items.
- Information must be presented in a clear way using visual elements like pie charts and bar graphs to facilitate better understanding.

3. Nutrition range filter

- Users should be able to select a nutrient (e.g. fat, sugar) and input minimum and maximum values (e.g. 5g to 15g).
- When filters are applied, the system should display a list of foods that fall within the specified range.

4. Nutrition level filter

- Users should be able to select one of several nutrients and choose a level (low, mid, high) based on predefined thresholds.
- The levels are defined as:
 - Low: Less than 33% of the highest value.
 - Mid: Between 33% and 66% of the highest value.
 - High: Greater than 66% of the highest value.

5. Recipe Builder

- Users should be able to select multiple food items and create a simple recipe.
- System should automatically add up their nutritional values and display them for users.
- Users should be able to save their created recipes for future reference.

Fictional User Profile:

Name: Alex Li

Occupation: Student

Age: 22

Comments: Fitness enthusiast, tech-savvy

Goals: Find healthy food options, keep track of his nutritional intake to maintain a balanced diet. Looking for low sugar, high protein diet.

Expected flow of events:

1. Alex opens the application.
2. Food Search:
 - The system displays a list of food items, but Alex is looking for a specific dietary criteria.
3. Apply nutrition range filters:
 - Alex chooses 'sugar' as the nutrient to filter.
 - Alex sets the maximum 'sugar' to be '5g'.
 - The system updates the list to show food items that fall within this low sugar range.
4. Apply nutrition level filter:
 - Alex selects "protein" as the nutrient to filter.
 - The system then updates the list again to show food items that meet both criteria:
 - low in sugar (up to 5g)
 - high in protein
 - This allows Alex to make informed decisions about his diet.

2.2 Software Requirements

This section provides a detailed overview of the capabilities and features of the software. It defines what the system should do and how it should perform. It covers the functional requirements, which refer to specific

tasks and functions the software must perform, and non-functional requirements, which refer to the performance standards to perform efficiently.

Functional Requirements

1. Food Search Functionality

- R1.1 The system shall allow users to enter a food item into a search bar.
- R1.2 The system shall display a list of food items matching the search query.
- R1.3 Each search result shall include nutritional information such as caloric value, fat content, etc.
- R1.4 The system shall allow users to select a food item from the search results to view detailed nutritional information.

2. Nutrition Breakdown

- R2.1 The system shall display a detailed view of nutritional information upon
- R2.2 The detailed view shall use pie charts and bar graphs to represent different nutrients such as fats, carbohydrates, and proteins.

3. Nutrition Range Filter

- R3.1 The system shall provide a feature to select a nutrient (e.g. fat, sugar).
- R3.2 The system shall allow users to input a desired range (e.g. 5g to 15g) for the selected nutrient.
- R3.3 The system shall display a list of food items that fall within the specified nutrient range.

4. Nutrition Level Filter

- R4.1 The system shall allow users to select a nutrient (e.g. protein) and choose a level (low, mid, high).
- R4.2 The system shall filter the database based on predefined thresholds for each nutritional level.
- R4.3 The system shall display a list of foods categorized into the selected nutrient levels (low, mid, high).

5. Recipe Builder

- R5.1 The system shall allow users to select multiple food items.
- R5.2 The system shall calculate the total nutritional values (e.g., calories, proteins, fats, etc.) for the selected food items.
- R5.3 The system shall display the combined nutritional values of the selected food items in a clear format.

Non-Functional Requirements

1. Performance

- N1.1 The system shall respond to search queries within 2 seconds.
- N1.2 The system shall display detailed nutritional information within 3 seconds of selecting a food item.

2. Usability

- N2.1 The system shall provide a user-friendly interface with intuitive navigation.

- N2.2 The system shall include clear labels and a logical flow to facilitate easy use.

3. Scalability

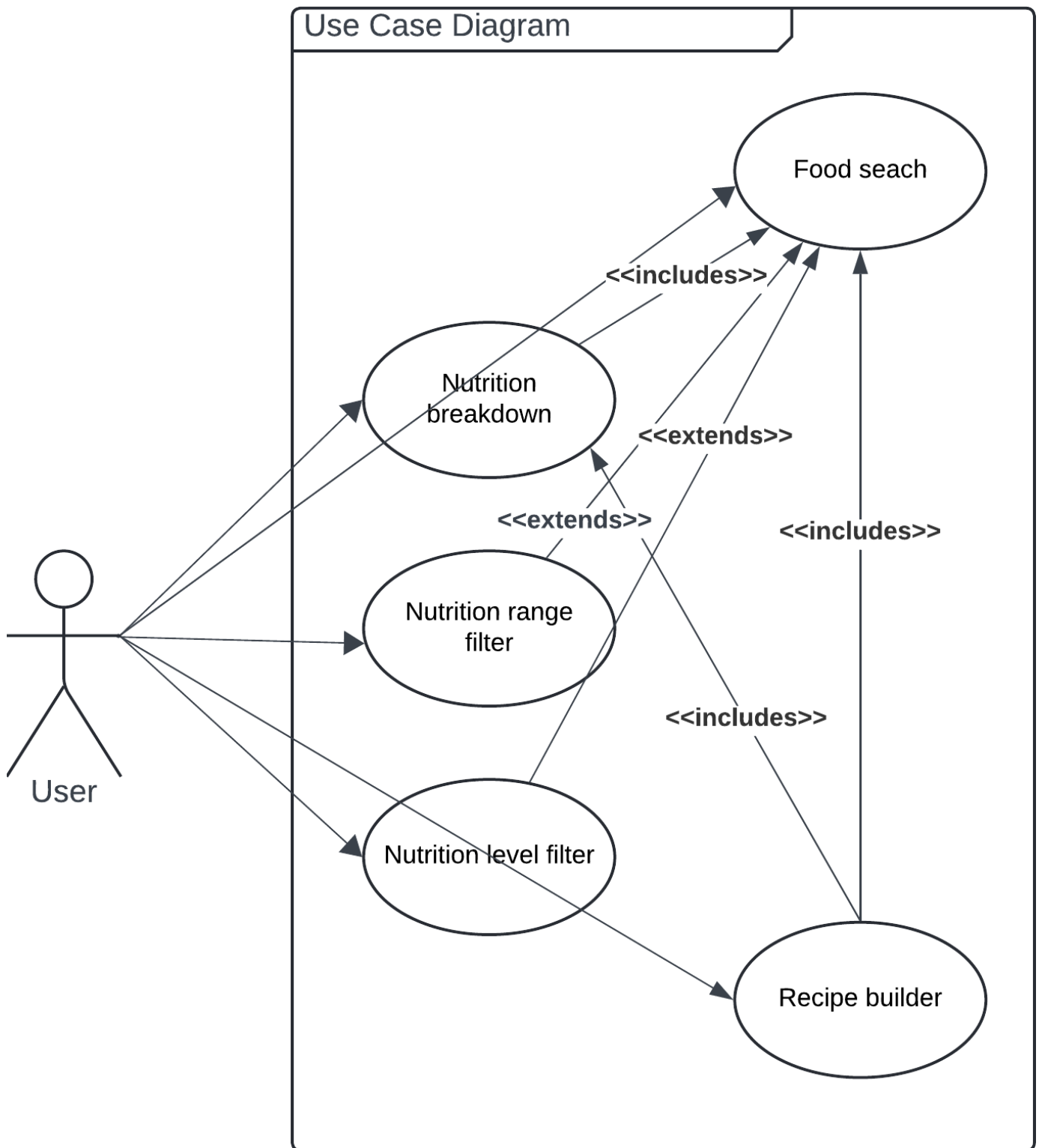
- N3.1 The system shall handle up to 10,000 food items without performance degradation.
- N3.2 The system shall allow for future expansion of nutritional data without requiring significant modifications.
- N3.3 The system shall handle up to 5,000 simultaneous user sessions during peak hours.

4. Security

- N4.1 The system shall protect user data from unauthorized access.
- N4.2 The system shall implement secure data storage and transmission protocols.

2.3 Use Case Diagram

Below is a system-level Use Case Diagram illustrating all required features:



2.4 Use Cases

| | |
|---------------|-------------|
| User Case ID | 1 |
| Use Case Name | Food Search |
| Actors | User |

| | |
|---------------------|--|
| User Case ID | 1 |
| Description | User can search for food items by entering a keyword into the search bar. |
| Flow of Events | 1. User enters a search term into the search bar. |
| | 2. System processes the search request. |
| | 3. System displays a list of food items that match the search. |
| | 4. User selects a food item from the results to view the nutritional information about it. |
| Alternate Flow | 1. If no matching items are found, system displays message to inform users that no results were found. |
| | 2. User can modify their search term and try again. |

| | |
|---------------------|---|
| User Case ID | 2 |
| Use Case Name | Nutritional Breakdown |
| Actors | User |
| Description | User can view nutritional information for a selected food item. |
| Flow of Events | 1. User selects a food item from the search results. |
| | 2. System retrieves the nutritional information for selected food item. |
| | 3. System displays detailed nutritional information. |
| Alternate Flow | 1. If nutritional information is not available, the system will display a message indicating the details are not available. |
| | 2. Users can return to search results to select another item. |

| | |
|---------------------|--|
| User Case ID | 3 |
| Use Case Name | Nutrition Range Filter |
| Actors | User |
| Description | Users can filter food items based on their desired range for any one of the nutrients. |
| Flow of Events | 1. User sets a nutritional range filter (e.g. 10g fat). |
| | 2. System applies the filter to the list of food items. |
| | 3. System displays the filtered list of food items that meet the range set by user. |

| | |
|---------------------|---|
| User Case ID | 3 |
| | 4. User reviews the filtered list. |
| Alternate Flow | 1. If no food items meet the filter set by user, the system displays a message indicating that no results were found. |
| | 2. User can adjust the filter and try again. |

| | |
|---------------------|---|
| User Case ID | 4 |
| Use Case Name | Nutrition Level Filter |
| Actors | User |
| Description | Users can filter food items based on the predefined nutritional content levels (low, mid, and high) for one of the following nutrient: fat , protein , carbohydrates , sugar , or nutritional density . |
| Flow of Events | 1. User selects a nutritional level filter option. |
| | 2. System applies the nutritional level filter to the list of food items. |
| | 3. System displays the filtered list of food items that meet the nutritional level set by user. |
| | 4. User reviews the list. |
| Alternate Flow | 1. If no food items meet the filters set by the user, the system displays a message indicating that no results were found. |
| | 2. Users can then readjust the filter. |

| | |
|---------------------|---|
| User Case ID | 5 |
| Use Case Name | Recipe builder |
| Actors | User |
| Description | User can select multiple foods and create a simple recipe by summing up their nutritional values. |
| Flow of Events | 1. User searches for food items. |

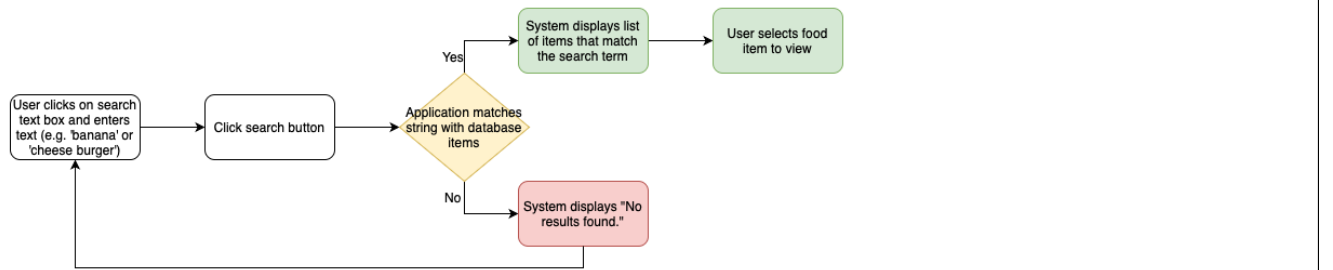
| | |
|----------------|---|
| User Case ID | 5 |
| | 2. User adds selected food items to the recipe. |
| | 3. System calculates and displays the total nutritional values of selected items. |
| | 4. User names the recipe and saves it for future reference. |
| Alternate Flow | 1. If system cannot retrieve nutritional information for a selected item, it notifies the user and excludes that item from the total calculation. |

3. Software Design and System Components

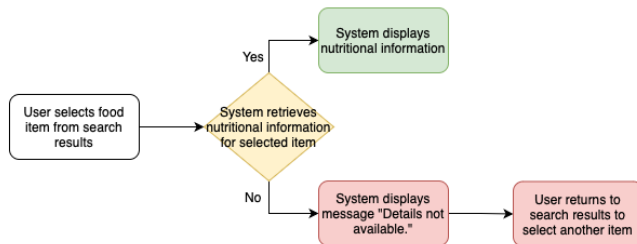
3.1 Software Design

Flowchart that illustrates how the software will operate.

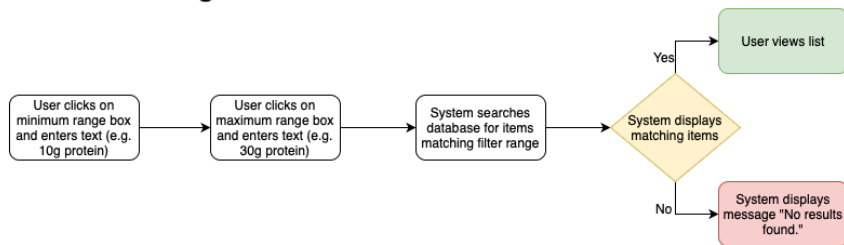
Food Search



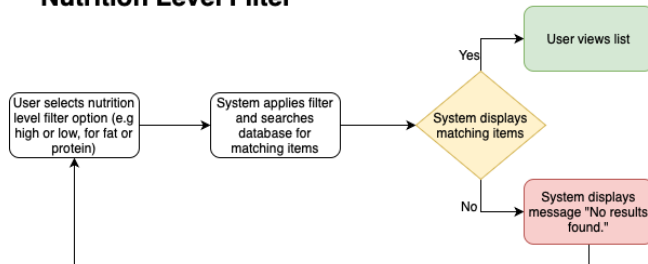
Nutritional Breakdown



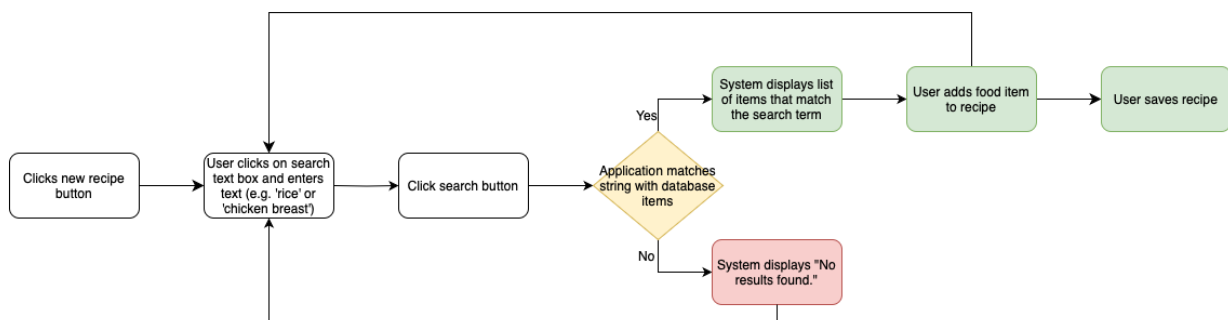
Nutrition Range Filter



Nutrition Level Filter



Recipe Builder



3.2 System Components

3.2.1 Functions

List all key functions within the software. For each function, provide:

- Description: Brief explanation of the function's purpose.
- Input Parameters: List parameters, their data types, and their use.
- Return Value: Describe what the function returns.
- Side Effects: Note any side effects, such as changes to global variables or data passed by reference.

Food Search Function

- Description: The purpose of this function is to match a search term, entered by the user, with food item(s) in the database.
- Input Parameters: A string entered by the user, and the 'search item' button being clicked will trigger the processing of the users search term against database items.
- Return Value: Matching item(s) in the database will be returned as a list in a panel.
- Side Effects: There is a global variable used for the search result items.

Nutritional Breakdown Function

- Description: This function allows the user to see the macronutrient and micronutrient values of a selected food item with models that help user understanding.
- Input Parameters: Clicking the respective button of a food item triggers the function to search and retrieve all nutrient values of that item in the database.
- Return Value: The selected food item's corresponding macronutrient and micronutrient values alongside models are returned.
- Side Effects: Once the food item is selected a global variable will be set to the name of that item.

Nutritional Range Filter Function

- Description: This function allows the user to apply a filter based on a selected nutrient and input minimum and maximum values of that nutrient. The application will then display a list of foods that are in that range.
- Input Parameters: Nutrient type will be input as a string in a textbox, the minimum and maximum vales will be input as integers in their respective textboxes. A button titled 'filter' will be the clicked to initiate the function.
- Return Value: The function returns a list of food items that fall within the minimum and maximum range of the specified nutrient.
- Side Effects: A global variable with an array datatype will be used to group food items within the specified nutrient range.

Nutritional Level Filter Function

- Description: The purpose of this function is to allow the user to filter foods by their nutritional content. There are three varying levels low, mid, and high.

- Input Parameters: The user clicks one of three nutritional level filter buttons, title low, mid, or high.
- Return Value: The function returns a list of food items that are within the filtered level.
- Side Effects: A global variable with the highest nutritional value will be allocated to that particular food item.

Recipe Builder Function

- Description: This function allows the user to select multiple food items and add them to recipe. This will add up the nutrients from the various components and give the total nutritional content of the recipe.
- Input Parameters: The user clicks on the 'new recipe' button, this initiates the function. Then the user will be able to add searched items to the recipe which will use the array datatype.
- Return Value: The function returns a recipe that contains total nutritional composition of the combined foods.
- Side Effects: A global variable with a dictionary datatype will be used to store the recipes, to allow later access.

3.2.2 Data Structures / Data Sources

List all data structures or sources used in the software. For each, provide:

- Type: Type of data structure (e.g., list, set, dictionary).
- Usage: Describe where and how it is used.
- Functions: List functions that utilize this structure.
- Type: Lists
- Usage: Used for storing and displaying search results, and nutritional breakdown values of specific food items. This will allow search results and models of nutritional values to be displayed and formatted.
- Functions: Food search function, nutritional breakdown function.
- Type: Arrays
- Usage: Used for storing larger amounts of data, to allow for quicker loading times and operation of the application. Specifically, when storing values for filter functions where there will possibly large amounts of data that needs to be retrieved. The array data structure will also be used for the recipe builder function to allow quick access to the various recipes and their respective nutritional values.
- Functions: Nutritional range filter function, nutritional level filter function, recipe builder function.
- Type: Modules
- Usage: Modules will be used to interface various components of the application. The various functions central to the software wouldn't be able to work without cross functionality from modules.
- Functions: Food search function, nutritional breakdown function, nutritional range filter function, nutritional level filter function, recipe builder function.
- Type: Packages
- Usage: Packages will be used for integrating the GUI, models and functionality for the application. The ones being utilised are wx and matplotlib.
- Functions: Food search function, nutritional breakdown function, nutritional range filter function, nutritional level filter function, recipe builder function.

3.2.3 Detailed Design

Provide pseudocode or flowcharts for all functions listed in Section 3.2.1 that operate on data structures. For instance, include pseudocode or a flowchart for a custom searching function.

Food Search Function

```
1. Start
2. Get searchValue
3. For i in foodArray
4.     if searchValue == i
5.         searchResult.append(i)
6.         returnResult = searchResult
7.     else
8.         returnResult = "No results found."
9. displayResult = returnResult
10. Panel.set(displayResult)
```

Nutritional Breakdown Function

```
1. Start
2. selectedItem = returnResult(0)
3. for i in foodArray(selectedItem[1, ])
4.     nutritionalValues.append(i)
5.     macroCategory = [nutrientValues(2, 10)]
6.     microCategory = [nutrientValues(10, -1)]
7.     macroColors = [ "red", "blue", "green", (more colours will be featured for all macronutrient values)]
8.     microColors = [ "gold", "aqua", "lilac", (more colours will be featured for all micronutrient values)]
9.     macroTotal = 0
10.    microTotal = 0
11.    for i in range(macroCategory)
12.        macroTotal += i
13.    for i in range(microCategory)
14.        microTotal += i
16.    macroSizes = [for i in range(macroCategory(i))]
17.    microSizes = [for i in range(microCategory(i))]
18.    fig, axes = plot.subplots
19.    ax1, ax2 = axes
20.    ax1.pie(macroSizes)
21.    ax2.pie(microSizes)
22. Layout()
```

Nutritional Range Filter Function

```
1. Start
2. Get filterNutrient, nutrientMin, nutrientMax
3. nutrientIndex = 0
4. for i in foodArray(0[,])
5.     if filterNutrient == i
6.         nutrientIndex = i.index
7. filteredItems = []
8. for i in foodArray([nutrientIndex])
9.     if i > nutrientMin & < nutrientMax
10.         filteredItems.append([0])
11.     else
12.         pass
13. if filteredItems is empty
14.     returnedItems = "No results found."
15. else
16.     returnedItems = filteredItems
17. Panel.set(returnedItems)
```

Nutrition Level Filter Function

```
1. Start
2. Get nutrientLevelFilter
3. nutrientFilterMax, nutrientFilterMin = 0
4. if nutrientLevelFilter = "low"
5.     nutrientFilterMax = highestNutrientFood * 0.33
6. elif nutrientLevelFilter = "mid"
7.     nutrientFilterMax = highestNutrientFood * 0.66
8.     nutrientFilterMin = highestNutrientFood * 0.33
9. elif nutrientLevelFilter = "high"
10.    nutrientFilterMax = highestNutrientFood
11.    nutrientFilterMin = highestNutrientFood * 0.66
12. filteredItems = []
13. for i in foodArray([1,])
14.     if i > nutrientFilterMin & < nutrientFilterMax
15.         filterItems.append(foodArray([0])
16.     else
17.         pass
18. if filteredItems is empty
19.     returnedItems = "No results found."
20. else
21.     returnedItems = filterItems
22. Panel.set(returnedItems)
```

Recipe Builder Function

```
1. Start
2. selectedItem = returnResult(0)
3. for i in foodArray(selectedItem[1, ])
4.     nutritionalValues.append(i)
5. macroCategory = [nutrientValues(2, 10)]
6. microCategory = [nutrientValues(10, -1)]
```

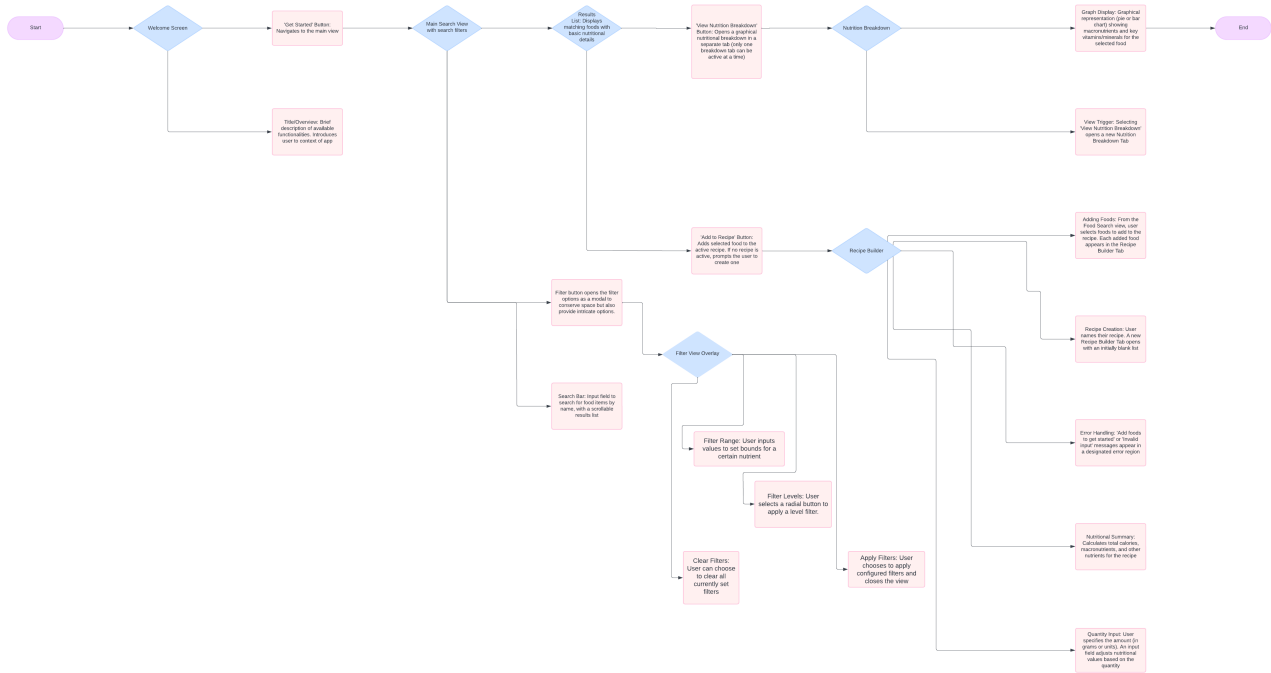


```
6. microCategory = [nutrientValues(10, -1)]
7. macroComponents = macroCategory.split
8. microComponents = microCategory.split
9. foodItemName = selectedItem
10. recipeComponentDictionary = {foodItemName: [macroComponents, microComponents]}
11. recipeName =
12. if recipeSave is clicked
13. recipeName = input("enter recipe name: ")
```

User Interface Design

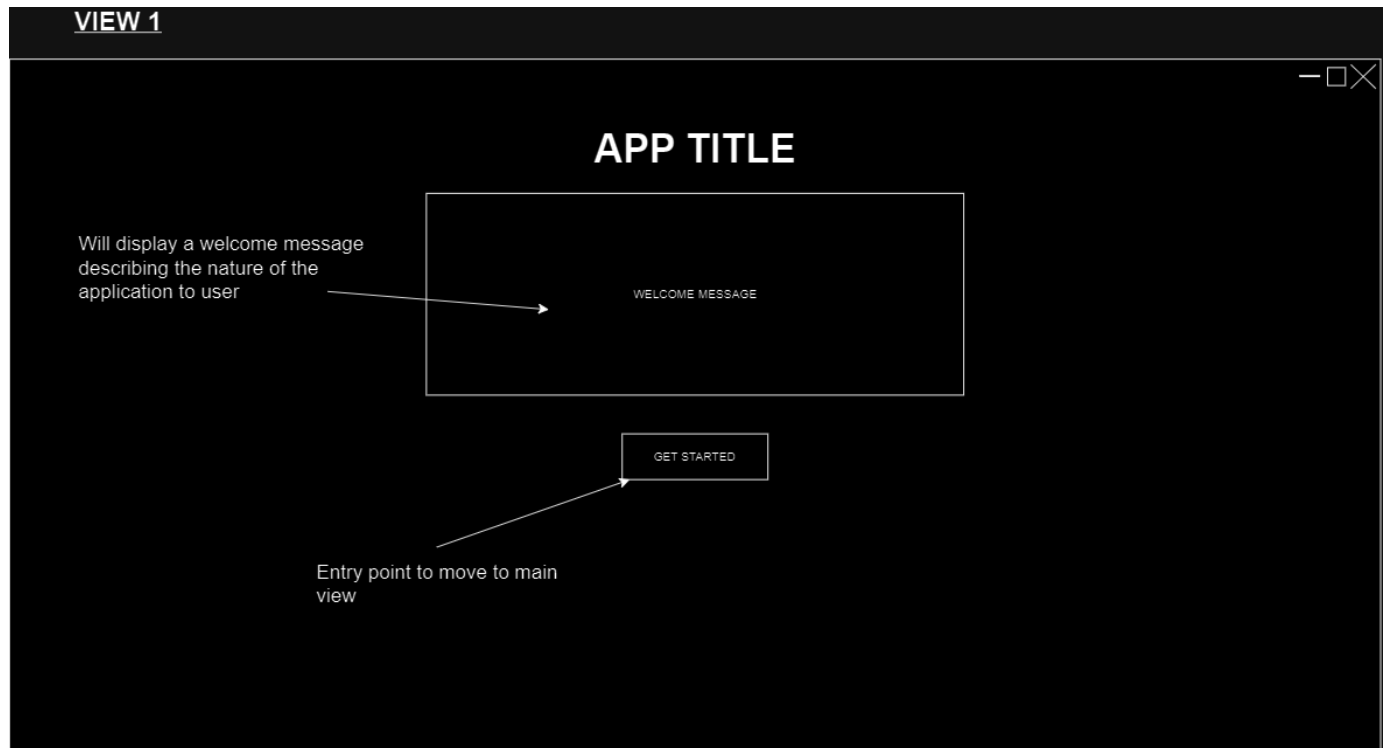
4.1 Structural Design

Hierarchy Chart:



4.2 Visual Design

Welcome View:



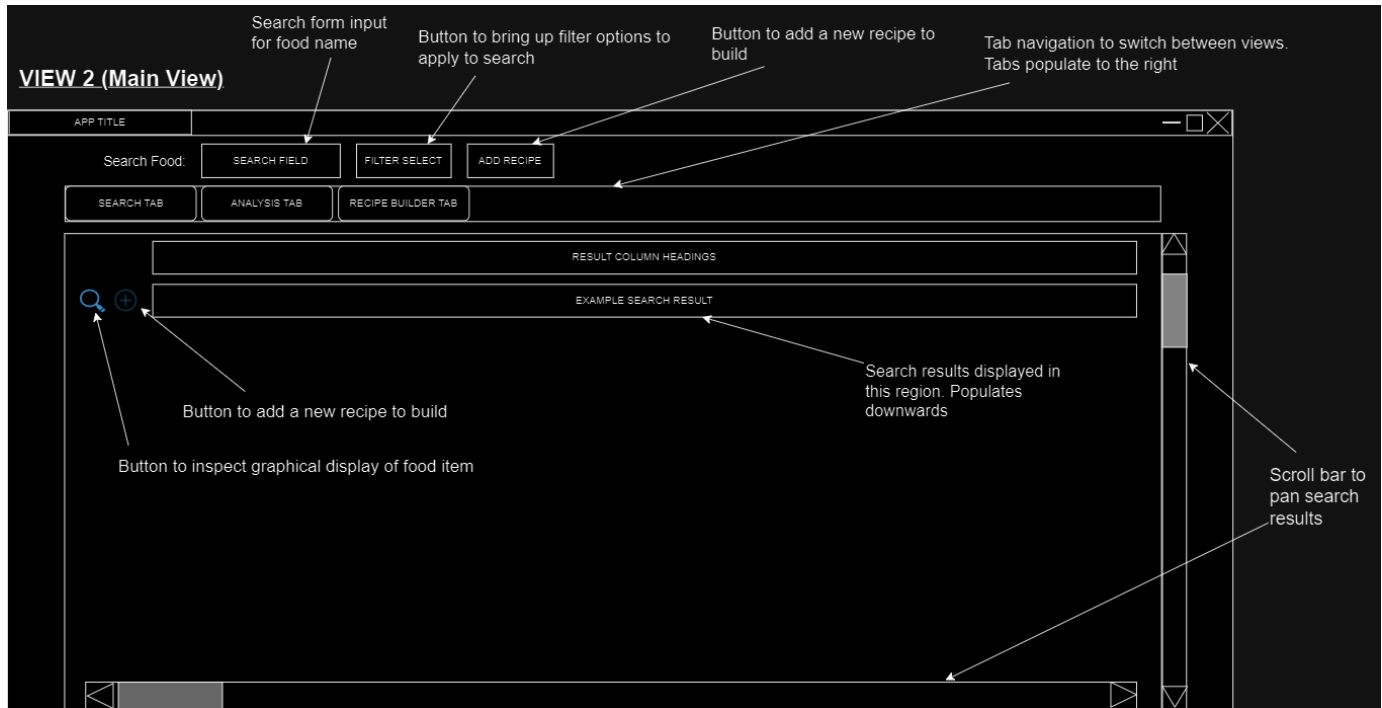
Purpose:

Serves as the main entry point to the application and provides the user with context about the tool.

Components:

- **Title** - The title of the application, large and centered.
- **Welcome Message** - A brief message informing the user about the capabilities of the application readable and centered to ensure they know what they can get out of the application.
- **Get Started** - Takes the user to the main UI of the application when they're ready clearly visible when they're ready to proceed.

Main View:



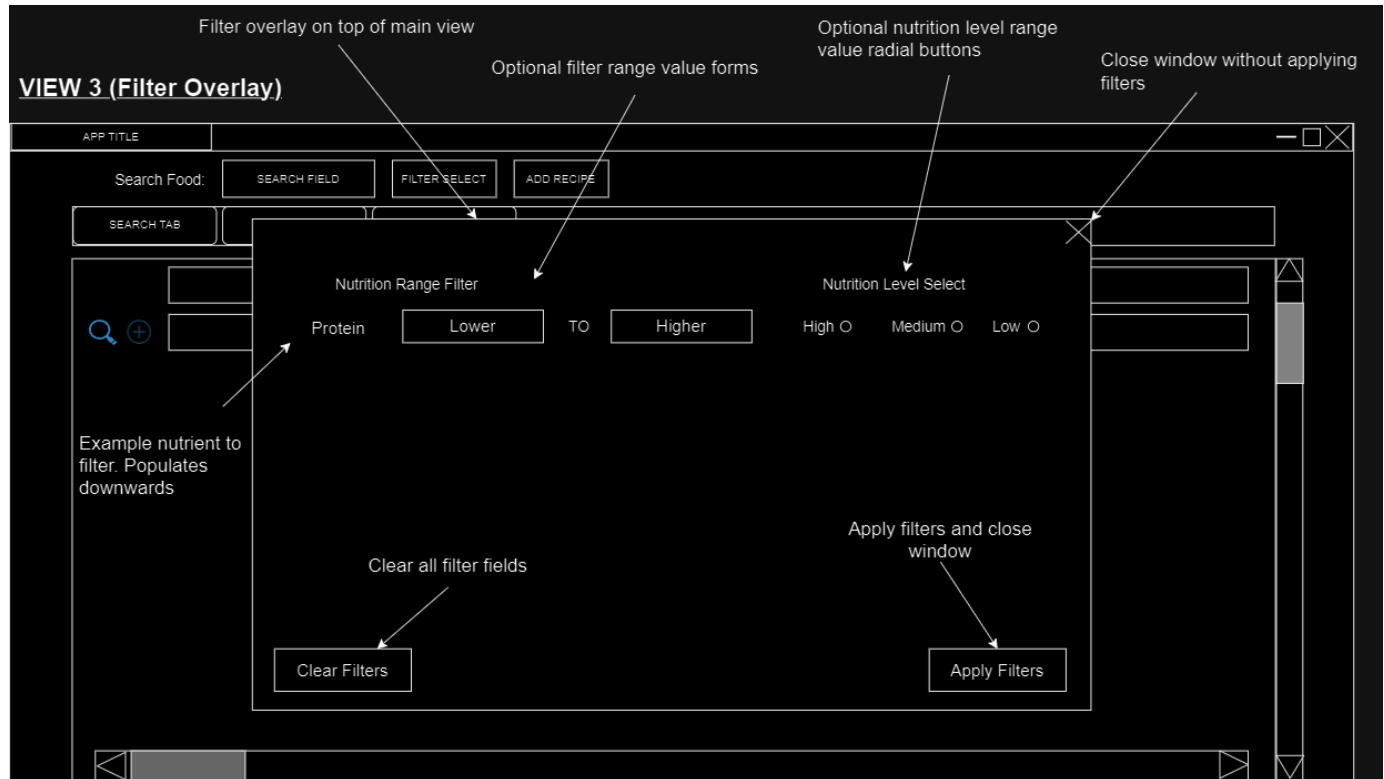
Purpose:

The Main Search View serves as the starting point for users to query the database for food items, apply desired filters and view their search results. It's the central hub between all the application's functions that allows the user to interact with data.

Components:

- **Search Bar** - Text input field positioned at the top for immediate access allowing users to identify the most essential operation. It allows users to query the dataset for a certain food.
- **Search Button** - Starts the search for the desired item positioned immediately next to the field for intuitive display.
- **Filter Button** - Opens a new view overlay that offers advanced filter options that the user can select for their search. Compiled into an overlay as it is unessential once specified so the user does not need to see it at all times. This is positioned next to search to denote its association with the feature.
- **Search Results Area** - Displays a list of the fetched search results from the dataset and grows downwards sorted into columns for each attribute. The area is large and can be scrolled to maximise space as it is the element the user will be parsing.
- **Interactive Buttons** - The plus symbol and magnifying glass depict buttons that will allow the user to add the item to a pending recipe or inspect the food item for a more graphical display. They're positioned such that the user knows which item they are associated with and the symbols are intuitive the depict th action to be taken.
- **Tab Navigation Area** - The different views of the app can be navigated to by clicking the tabs at the top. Again they are placed in an intuitive and visible location whilst not impeding any display.

Apply Filter View:



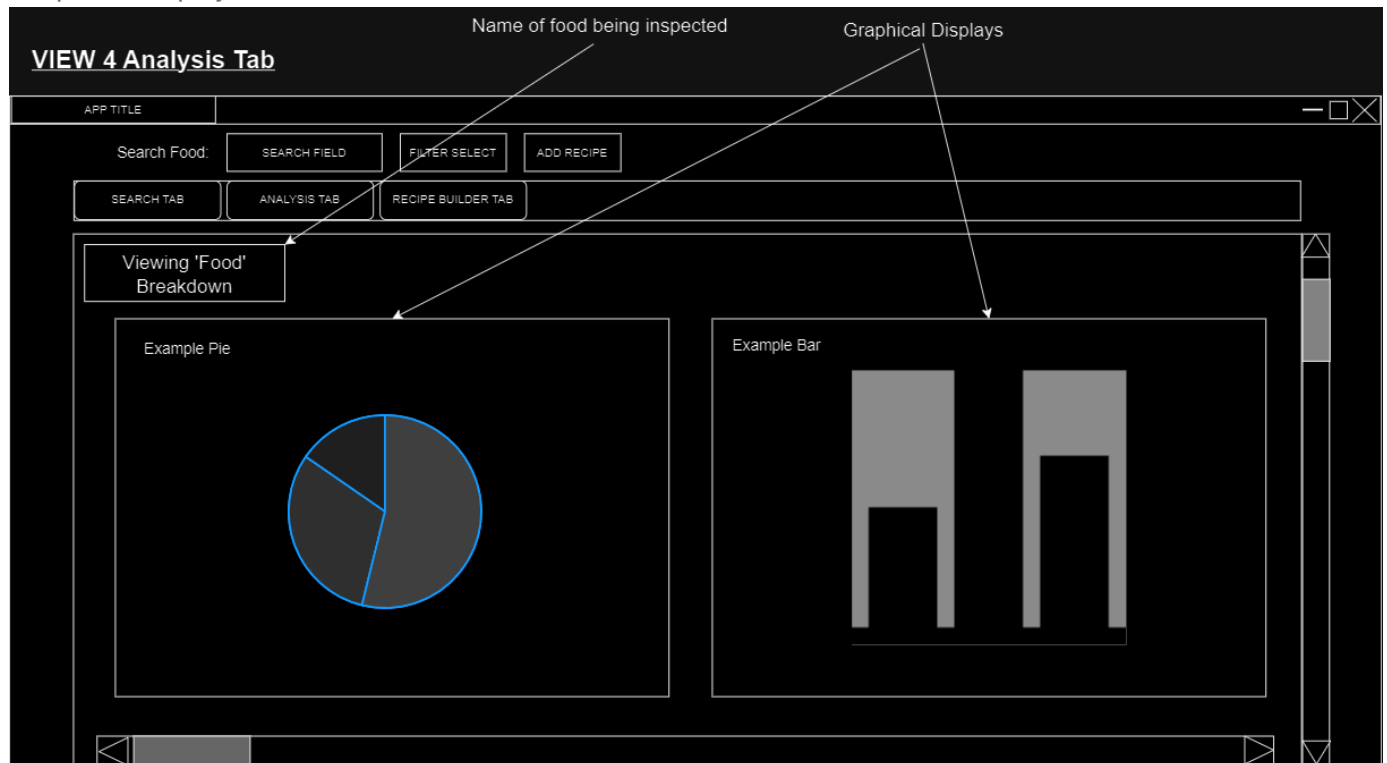
Purpose:

This view allows users to apply selected filters to their search queries. It is presented to the user as a modal or overlay to conserve space on the main screen whilst still providing comprehensive functionality.

Components:

- **Optional Filter Range Values** - Allows the user to search for nutrient values within the specified range for the associated nutrient. It will list the nutrient components of foods and can be scrolled if necessary.
- **Optional Nutrient Level Values** - Radial buttons that allows a user to specify the various levels of a particular nutrient they want to have in their results. Only one button will be selected at a time and double tapping the same button should de-select it.
- **Close Button** - Intuitive symbol and positioning. Closes the filters without saving.
- **Clear Filters** - Clears the search filters if any are set. Positioned in a clear location but separated from the apply filters button to reduce the likelihood of accidentally clearing.
- **Apply Filters** - Applies the search filters to be used and closes the overlay window.

Graphical Display View:



Purpose:

This view presents the user with the graphical analysis made by the application.

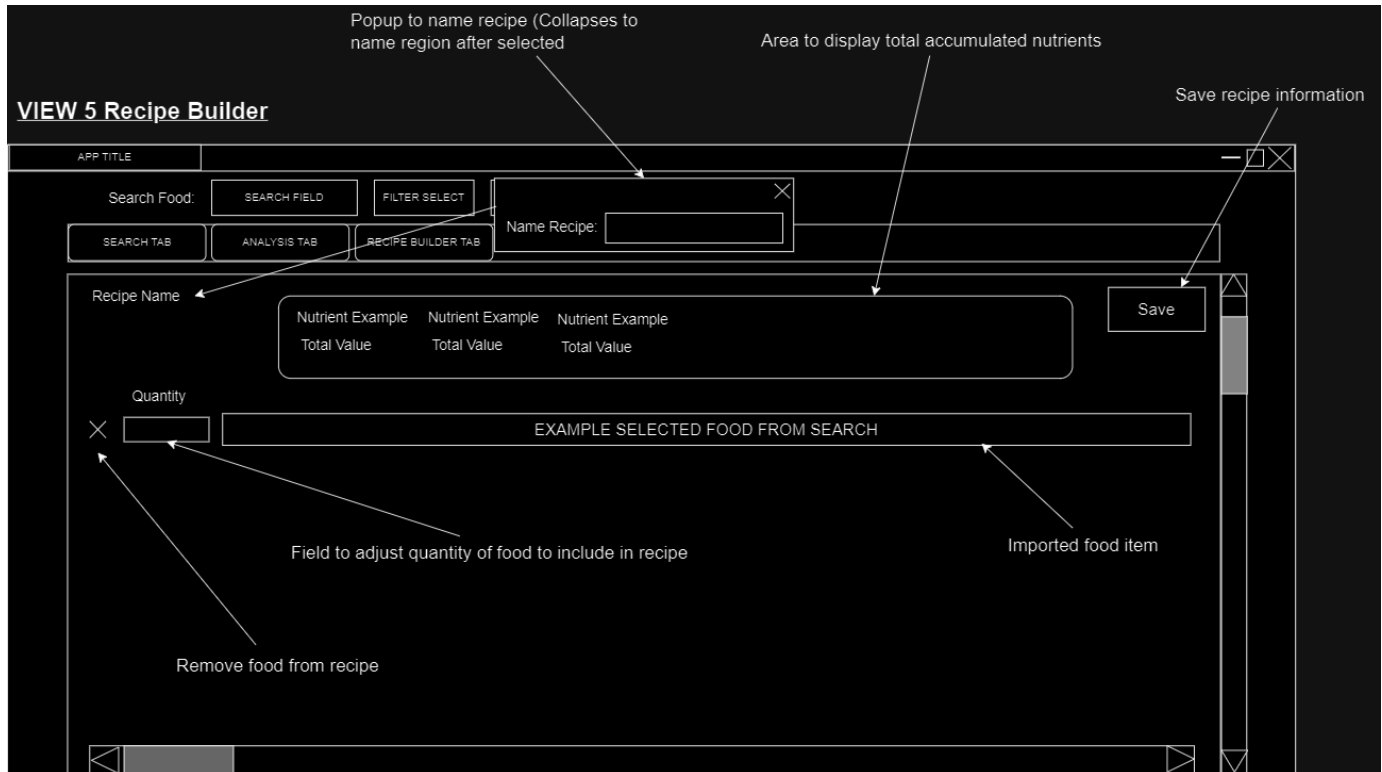
Components:

- **Food Name** - Displays the currently inspected food item. Positioned within the main viewing area as a clear indicator of what information is being displayed.
- **Pie Chart Region** - Displays the calculated pie chart visual representation for the associated food. Takes up half the view space as it is an important element.
- **Bar Chart Region** - Displays the calculated bar graph for the associated food. Takes up half the space as it is an important element.

Additional considerations:

Depending how the visualisations are rendered an option may be to switch to full view width displays and placing the graphs on top of each other with a scrollable view space.

Recipe Builder View:



Purpose:

This view allows the user to compile a list of various food items into a 'recipe' and sum its total nutritional value.

Components:

- **Name Popup** - When initialised the recipe must be named so the popup ensures that the user names their recipe. It is displayed as a prominent screen element and requires confirmation to proceed to operating the view. Once named it closes and moves out of the way populating a region of the screen to denote clear association with the recipe being constructed.
- **Total Accumulated Nutrients** - A region that displays the total accumulated nutritional value of the recipe. Positioned at the top so the user always has a clear indication of their recipe.
- **Save** - Allows the user to save their recipe. Positioned at the top with other important information but far to the right as it is not always applicable to the user.
- **Remove Button** - Removes an item from the recipe. Uses an X symbol for intuitive understanding. Positioned next to the item it should remove.
- **Quantity** - Allows the user to specify the amount of the associated food to calculate the nutrients for as the dataset lists quantities per 100g so amounts need to be adjusted for more realistic recipes.
- **Imported Food Item** - Displays the nutrient information for the associated food as a general reference to the user. Presented in a recognisable format from the previous search view.

4.3 General UI Considerations

Tab Limitations:

- Users can have up to two additional tabs open at any time:
- One Recipe Builder Tab (to manage an active recipe).
- One Nutrition Breakdown Tab (for graphical information on the last viewed food item).
- If the user requests a new graphical breakdown, it will replace the current Nutrition Breakdown Tab, ensuring only one graphical summary is viewable at a time.

Handling Recipe Creation:

- If a user attempts to add an item to a recipe and no recipe tab is open, the system will prompt the user to create a new recipe (similar to manually starting a new recipe).
- The user will be asked to name their recipe before the Recipe Builder Tab opens, keeping the flow intuitive.

Feedback and Error Display:

- Errors (e.g., "invalid input" or "no foods added") and user actions (e.g., "food added to recipe") will be handled in a clear, consistent region of the UI. Errors should be visually distinct (e.g., red for errors, green for success).