

Software Design Document

Project Name: Nutritional Food - Assignment.

Group Number: 59

Team members

Student Number	Name
s5277614	Gemma Manns
s5384766	Joshua Wagner
s5280075	Juan Martinez

Table of Contents

- Software Design Document
 - Project Name: Nutritional Food - Assignment.
 - Group Number: 59
 - 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
 - 2.2 Software 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
 - 4. User Interface Design
 - 4.1 Structural Design
 - 4.2 Visual Design

1. System Vision

1.1 Problem Background

Problem Identification: What problem does this system solve?

The system addresses the challenge of accessing and understanding detailed nutritional information for a wide range of food items. Many individuals, healthcare professionals, and dietitians struggle to find reliable and comprehensive nutritional data in a user-friendly format. This tool solves the problem by providing a centralised platform where users can easily search for foods, analyse their nutritional content, and make informed dietary decisions. It facilitates personalised diet planning, supports health and medical research, and helps users manage specific dietary requirements or restrictions.

Dataset: What is the dataset used?

The dataset used is a CSV file named `Nutritional_Food_Database.csv`. This database contains detailed nutritional information for various food items commonly consumed around the world. It includes columns that describe food names and their nutritional content, such as caloric value, fat types, carbohydrates, proteins, vitamins, minerals, and more. The dataset is essential for supporting a wide range of dietary, health, and medical research applications.

Data Input/Output: What kind of data input and output is required?

Data Input:

- **Text Input:** Users will input text strings when searching for specific food items by name or when selecting nutrients to filter by range or level.
- **Numerical Input:** Users may enter numerical values to set minimum and maximum ranges for specific nutrients, such as calories, fat, or protein content.
- **Selection Input:** Users will use selection inputs, such as drop-down menus or checkboxes, to choose specific nutrients or categories of nutrients for analysis and filtering.

Data Output:

- **Textual Output:** The system will display lists of food items along with their corresponding nutritional information based on user searches and filters.
- **Graphical Output:** The tool will generate visual representations, such as pie charts and bar graphs, to illustrate the nutritional breakdown of selected food items. These visuals will help users easily interpret the data and make informed decisions.
- **Filtered Results:** The system will provide filtered lists of foods that meet the criteria set by the user, such as foods within a certain caloric range or with a specific nutrient level.

Target Users: Who will use the system, and why?

- **Sports Professionals:** Use the tool to optimise dietary plans for athletes by selecting foods that meet specific nutritional needs, such as high protein or low fat.
- **People with Dietary Requirements or Restrictions:** Individuals with allergies, medical conditions, or personal dietary preferences can use the system to find foods that align with their specific nutritional requirements. Doctors:

Utilise the tool to recommend and plan dietary interventions for patients with specific health conditions, such as diabetes, cardiovascular issues, or obesity.

- **Nutritionists:** Use the system to create balanced diet plans for clients, ensuring that all nutritional needs are met based on detailed food data.
- **Dietitians:** Provide expert advice and feedback on food choices, using the tool to support patient dietary management and to educate clients on healthy eating.
- **Health-Conscious Consumers:** Individuals looking to make informed decisions about their food choices can use the tool to understand the nutritional content of the foods they consume and to plan meals that support their health goals.

1.2 System capabilities/overview

Key system capabilities include:

Food Search:

Users can input text to search for specific food items by name. The system will return results in a table format, displaying the corresponding nutritional information for each matching food item. This search feature helps users quickly find the foods they are interested in and review their nutritional profiles.

Nutritional Breakdown:

For any selected food item, the system will display its nutritional breakdown in both numerical and graphical formats. Users can view pie charts and bar graphs that visually represent the nutrient composition, making it easier to understand the relative amounts of calories, fats, proteins, vitamins, and other nutrients.

Nutritional Range Filter:

This feature allows users to filter food items by setting minimum and maximum values for specific nutrients, such as calories, fats, or proteins. The system will display all food items that meet the specified nutrient ranges, providing a customised list of results based on the user's requirements.

Nutritional Level Filter:

Users can filter food items by predefined nutrient levels (low, medium, or high) compared to the maximum values in the database. This allows for a broader, qualitative search when precise numeric filtering is not necessary or desired. The system will return food items that fall into the selected nutrient level.

Weight Calculator:

Users can input a specific food item and a desired serving size (in grams). The system will then calculate the scaled nutritional values based on the input weight, providing personalised nutritional information for different portion sizes. This feature is useful for users aiming for portion control or tailored meal planning.

Visual and Textual Output:

The system supports both textual and graphical outputs. In addition to text-based tables listing the nutritional details, the graphical outputs such as pie charts and bar graphs provide a more engaging and intuitive way to analyse food data. This dual format caters to both detailed-oriented users and those who prefer visual summaries.

1.3 Benefit Analysis

The Nutritional Food Analysis System provides significant value across various user groups, contributing to better dietary decisions, improved health outcomes, and enhanced professional practice. The benefits of the system include:

Enhanced Dietary Planning:

- **Personalised Diets:** The system enables users to tailor their diets based on specific nutritional needs, whether for weight management, athletic performance, or medical conditions. By providing detailed nutritional information and filtering options, users can easily find foods that align with their dietary goals.
- **Accurate Nutritional Calculations:** The weight-based nutritional calculator allows users to scale the nutritional content of food items, ensuring precise portion control and more accurate dietary planning.

Improved Health Outcomes:

- **Informed Food Choices:** By offering detailed and reliable nutritional data, the system empowers users to make informed food choices that support their health objectives. This is particularly beneficial for individuals managing conditions like diabetes, cardiovascular diseases, or obesity.
- **Support for Health Professionals:** Nutritionists, dietitians, and doctors can use the system to develop and recommend scientifically-backed dietary plans, enhancing patient care and dietary interventions.

Educational Value:

- **Nutritional Awareness:** The system's visual representation tools, such as pie charts and bar graphs, make it easier for users to understand the nutritional content of foods. This helps in educating users about balanced diets and the impact of different nutrients on health.
- **Accessible Information:** The user-friendly interface and comprehensive database make nutritional information accessible to a wide audience, from health-conscious consumers to professionals in the healthcare industry.

Time Efficiency:

- **Quick Food Searches:** The advanced search and filtering capabilities allow users to quickly find and analyse food items, saving time compared to traditional methods of dietary research.
- **Streamlined Professional Use:** For professionals like nutritionists and dietitians, the system streamlines the process of dietary analysis and planning, allowing them to focus more on client interaction and less on manual data processing.

Comprehensive Data Access:

- **Centralised Nutritional Information:** The system acts as a central repository for detailed nutritional data, reducing the need to consult multiple sources and ensuring consistency in the information provided.

2. Requirements

2.1 User Requirements

A frequent user of the software is a **Nutritionist**. A Nutritionist is a person that advises their clients on their diet and how it impacts their health so it's critical they know the right foods to recommend. On behalf of their clients, Nutritionists need

to gather a collection of foods that they can recommend to their client so they will often use most or all features of the software during a singular session to gather the desired foods. The nutritionist has their own needs for each feature and how they present the results to their queries.

1. The **Food Search** must accept a text input and return all applicable foods and their nutritional values on a scrollable table that match or contain the search query.
2. The **Nutritional Breakdown** will give the nutritionist the option to input a food as text and the software visualises the nutrient composition of the selected food on a pie chart and a bar graph.
3. The **Nutritional Range** allows the nutritionist to select a nutrient type they want to filter and specify a minimum and maximum value (in g) that they want the results to fall within. When the search button is pressed, the nutritionist expects that the software will display all applicable foods and their nutritional details on a scrollable table.
4. The **Nutritional Level** allows the nutritionist to select a nutrient type they want to select and a specific nutrient level (Low, Mid or High) compared to the highest value on the database. When the search button is pressed, the nutritionist expects that the software will display all applicable foods and their nutritional details on a scrollable table.
5. The **Weight Calculator** Allows the nutritionist to input a food as text and a desired weight to scale to. Then the Nutritionist expects the system to visualise the scaled nutritional values on a table and bar graph.

2.2 Software Requirements

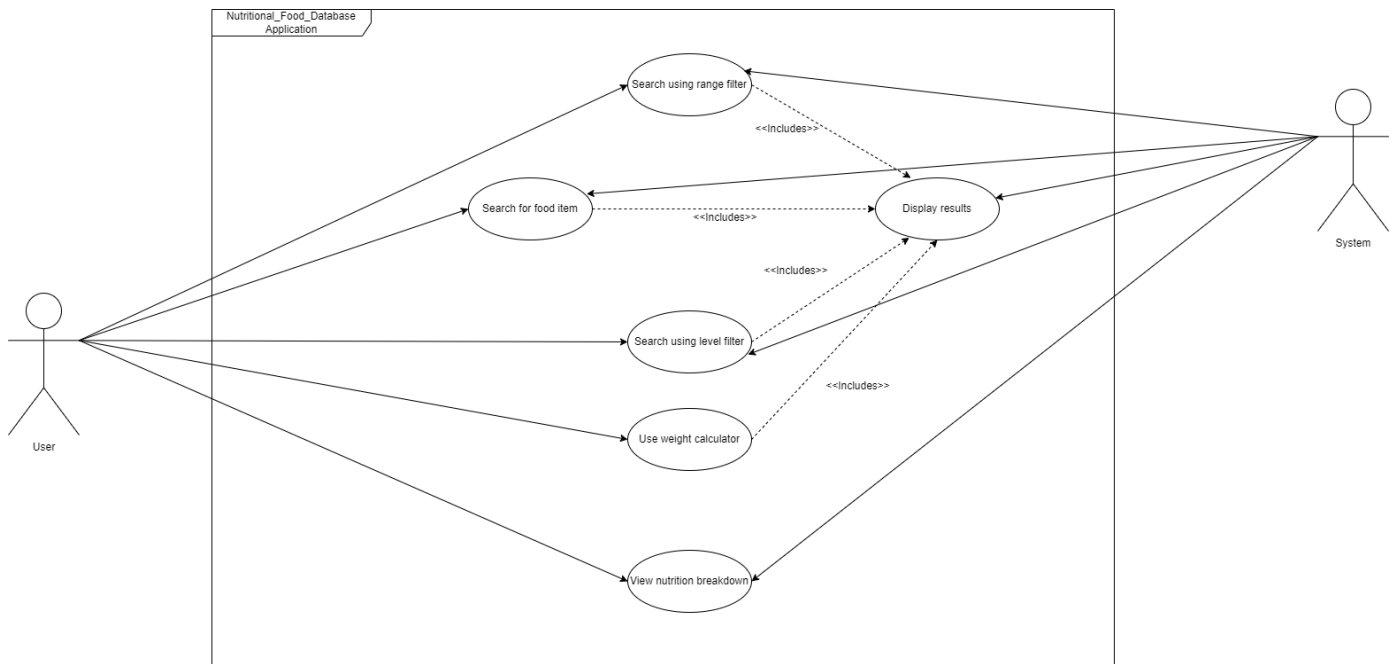
Functional requirements:

- R1.1 The program shall only accept all standard English letters, symbols and numerical characters as an input.
- R1.2 The program shall use the stored "Nutritional_Food_Database.csv" as its sole source of information and provide search results based on it.
- R2.1 The Food Search shall provide a list of all applicable results that contain the text contents of the user's input.
- R3.1 The Nutrition Breakdown function shall show pie charts and bar graphs displaying the nutritional breakdown of the selected food.
- R4.1 The Nutrition Range Filter shall accept a string value to determine the nutrition being range limited.
- R4.2 The Nutrition Range Filter shall accept floating point values to constrain the minimum and maximum value of a specific nutrition that show up in the results.
- R5.1 The Nutrition Level Filter shall provide a dropdown value to filter their search results based on nutritional density which includes "Low", "Mid" and "High" as selectable values or "NONE" as a placeholder if undecided.
- R6.1 The Weight calculator feature shall give the option for the user to input a food item in plaintext along with a numerical weight (in grams) in which they can scale the other nutritional values with.
- R6.2 The Weight calculator shall provide a list of scaled nutritional values based on the inputted food and weight.

Non-Functional requirements:

- R1.1 User Interface shall be clean and easy to navigate.
- R2.1 Widget layout shall be consistent both in size and position.
- R3.1 Widgets shall follow the same layout/sizing practices of other applications.
- R4.1 List of values shall be properly aligned with associated nutrition column names.
- R5.1 Buttons and navigational widgets shall transfer to the correct Frame/ function when interacted with.

2.3 Use Case Diagram



2.4 Use Cases

Use Case ID	UC-01
Use Case Name	Search for food item
Actors	User
Description	The user enters an item they want to search for in the textbox and presses search to retrieve all results that match the query.
Flow of Events	1. The user selects the search bar.
	2. The user writes a text input the matches/closely relates to an item they want to search for.
	3. The user presses search
	4. The system retrieves all matching rows.
	5. The system displays the results (or lack thereof).
Alternate Flow	1. If there are no matching results the user is notified through a message after pressing search and sent back to the search screen

Use Case ID	UC-02
Use Case Name	Search using range filter
Actors	User

Use Case ID	UC-02
Description	The user designates one nutritional value to filter by along with defining a minimum and maximum threshold for its corresponding value and returns food items that match the specified values
Flow of Events	1. User selects the nutritional filter feature.
	2. The user selects “nutritional range” option.
	3. The user inputs the nutrient they want to filter by.
	4. The user specifies a floating-point minimum and maximum nutritional value for the nutrient.
	5. The user presses the “confirm” button.
	6. The system retrieves all matching rows.
	7. The system displays the results (or lack thereof).
Alternate Flow	1. If there are no matching results the user is notified through a message after pressing search and sent back to the nutritional range screen.
	2. If the user presses “Filter again” it will return back to the filter screen to repeat the flow of events.

Use Case ID	UC-03
Use Case Name	Search using level filter
Actors	User
Description	The user designates one nutritional value to filter by along with defining if its nutritional levels is “LOW”, “MEDIUM” or “HIGH” compared to the maximum value and returns food items that matches the specified range
Flow of Events	1. User selects the nutritional filter feature.
	2. The user selects “nutritional level” option.
	3. The user inputs the nutrient they want to filter by.
	4. The user specifies if they want the nutritional level to be “LOW”, “MEDIUM” or “HIGH”.
	5. The user presses the “confirm” button.
	6. The system retrieves all matching rows.
	7. The system displays the results (or lack thereof).

Use Case ID	UC-03
Alternate Flow	1. If the user presses “Filter again” it will return back to the filter screen to repeat the flow of events.

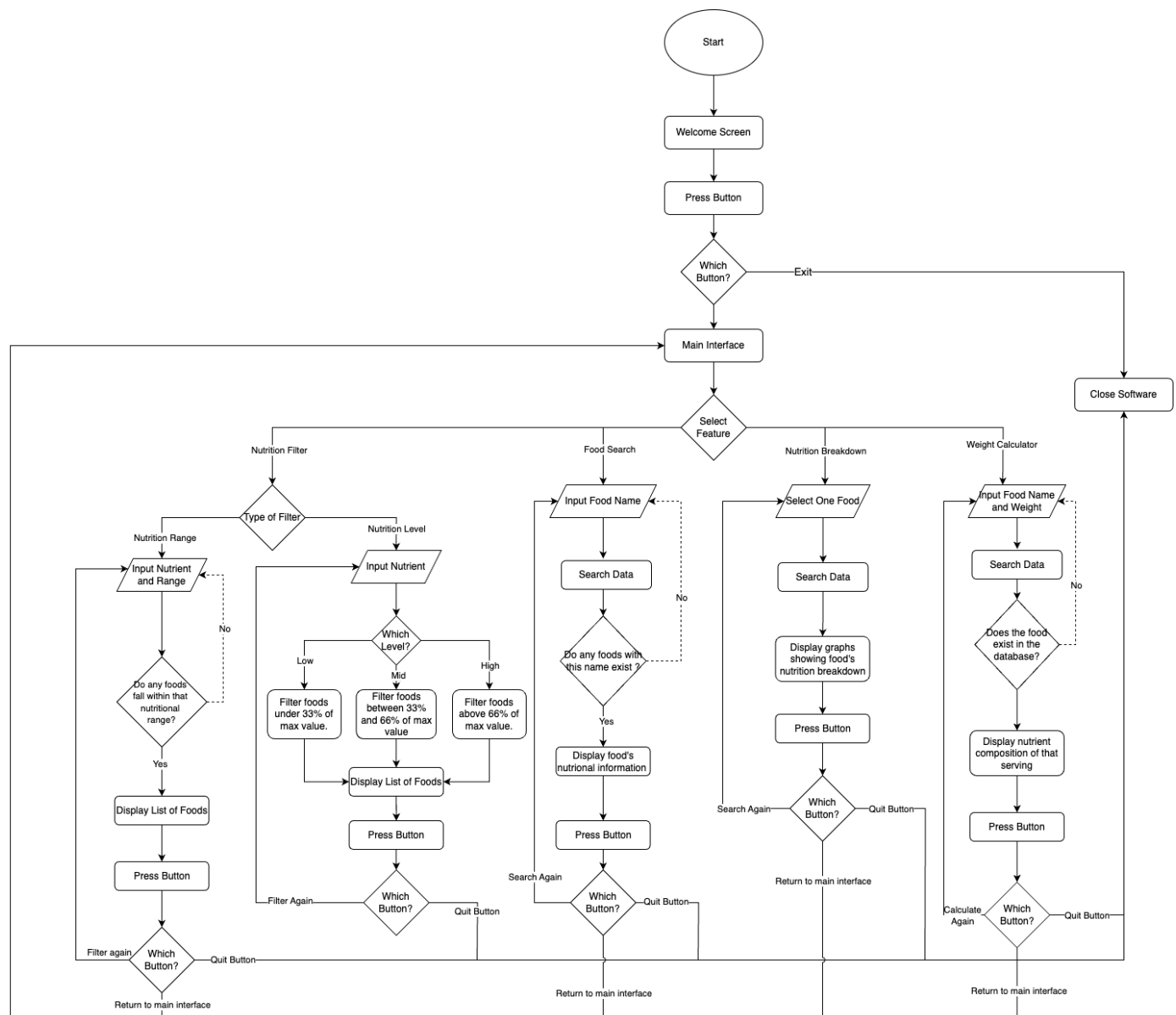
Use Case ID	UC-04
Use Case Name	View nutritional breakdown
Actors	User
Description	The user selects 1 food and the system displays its nutritional breakdown of the food in the form of pie charts and bar graphs.
Flow of Events	1. User selects a food item.
	2. The system searches the for the full data of the selected food item.
	3. The systems displays pie charts and bar graphs based on the food items nutritional breakdown.
	4. The user specifies if they want the nutritional level to be “LOW”, “MEDIUM” or “HIGH”.
	5. The user presses the “confirm” button.
	6. The system retrieves all matching rows.
	7. The system displays the results (or lack thereof).
Alternate Flow	1. The system gives the user an option to select another food item. If selected repeat the flow of events again.

Use Case ID	UC-05
Use Case Name	Weight Calculator
Actors	User
Description	The user inputs a food item and floating-point weight in grams that the system uses to calculate and display the appropriately scaled nutrient composition of the food item.
Flow of Events	1. User selects weight calculator feature.
	2. User inputs the food item’s name and weight they desire it to be scaled to.
	3. The system searches for the food item and calculates the results.
	4. The system displays the scaled nutrient composition for the specified weight.

Use Case ID	UC-05
Alternate Flow	1. If the food item doesn't exist in the database the user is prompted to input the food item and weight again.
	2. The system gives the user an option to select another food item to calculate. If selected repeat the flow of events again.

3. Software Design and System Components

3.1 Software Design



3.2 System Components

3.2.1 Functions

ReadLines

- Description: Read all lines of the Food_Nutrition_Dataset.csv and store it as an array.
- Input Parameters:
 - Food_Nutrition_Dataset.csv
- Return Value: CSV_Arr[][] --> A 2D array containing all the records of the Food_Nutrition_Dataset.csv.
- Side Effects: Data has to be re-initiated if the software is re-opened.

Search_All_Foods

- Description: Search all rows in the csv and retrieve the ones where the “food” element matches all or part of the user’s string input.
- Input Parameters:
 - User_Food_Input --> A string that stores the raw user input from the search bar.
- Return Value: Food_Arr[][] --> A 2D array containing all the records that match or contain the string in User_Food_Input
- Side Effects: All data structures are restored to their default values if the button “Calculate Again” is pressed, it’s re-opened by the main interface or the software is closed.

Search_Single_Food

- Description: The user selects a food in which the system retrieves its nutritional information of. Its used to create pie charts and bar graphs for the Nutrition_Breakdown feature and calculations for the Weight_Calculator.
- Input Parameters:
 - User_Food_Input --> A string that stores the raw user input.
- Return Value: SFood_Arr[] --> A 1D array containing all the entities that belong to the food specified User_Food_Input but this is displayed visually with pie charts and bar graphs.
- Side Effects: All data structures are restored to their default values if the button “Calculate Again” is pressed, it’s re-opened by the main interface or the software is closed.

Nutrition_Breakdown

- Description: The user selects a food in which the system calls the Retrieve_Single_Food to retrieve its details and create pie charts and bar graphs for the user to view.
- Input Parameters:
 - User_Food_Input --> A string that stores the raw user input from the text box. Used to generate SFood_Arr[] by calling Retrieve_Single_Food.
- Return Value: SFood_Arr[] displayed visually with pie charts and bar graphs.
- Side Effects: All data structures are restored to their default values if the button “Calculate Again” is pressed, it’s re-opened by the main interface or the software is closed.

Weight_Calculator

- Description: The user inputs the name of a food and weight (in g) of the desired portion for the system to calculate and display scaled nutritional information.
- Input Parameters:
- User_Food_Input --> A string that stores the raw user input from the text box. Used to generate SFood_Arr[] by calling Retrieve_Single_Food.
 - User_Weight_Input --> A floating point value that stores the user input for the weight.

- Return Value: Calc_Food_Arr[] --> A 1D array containing all the entities that belong to the food specified User_Food_Input and has been scaled based on User_Weight_Input.
- Side Effects: All data structures are restored to their default values if the button “Calculate Again” is pressed, it’s re-opened by the main interface or the software is closed.

Nutrition_Level_Filter

- Description: Filter all rows of the csv to only show rows that contain a specific weight range (density) of a nutrient that the user has defined.
- Input Parameters:
 - User_Nutrient_Input --> A string that stores the user’s selected nutrient.
 - User_Nutrient_Level --> A string that stores the specified nutrient level (Low, Mid or High).
- Return Value: Food_Arr[][] --> A 2D array containing all the records that meet the User_Nutrient_Level of the specified User_Nutrient_Input.
- Side Effects: All data structures are restored to their default values if the button “Calculate Again” is pressed, it’s re-opened by the main interface or the software is closed.

Nutrition_Range_Filter

- Description: Filter all rows of the csv to only show rows that contain a specific level of a nutrient that the user has defined.
- Input Parameters:
 - User_Nutrient_Input --> A string that stores the user’s selected nutrient.
 - User_Nutrient_Min --> A floating point value that stores the lower bound of the user’s specified range.
 - User_Nutrient_Max --> A floating point value that stores the upper bound of the user’s specified range.
- Return Value: Food_Arr[][] --> A 2D array containing all the records that are within the range of User_Nutrient_Min and User_Nutrient_Max of the specified User_Nutrient_Input.
- Side Effects: All data structures are restored to their default values if the button “Calculate Again” is pressed, it’s re-opened by the main interface or the software is closed.

3.2.2 Data Structures / Data Sources

Food_Nutrition_Dataset.csv

- Type: csv file
- Usage: Contains all the data in comma delimited format.
- Functions: All except for Weight_Calculator.

Line

- Type: String
- Usage: Stores the next line of csv data that needs to be read and stored in CSV_Arr[][].
- Functions: ReadLines

CSV_Arr[][]

- Type: 2D array
- Usage: A 2D array used to store the contents of the aforementioned Food_Nutrition_Dataset.csv.
- Functions: All except for Weight_Calculator.

“i”, “j”

- Type: int
- Usage: Ints used to represent positions in 1D and 2D arrays.
- Functions: All except for Weight_Calculator.

User_Food_Input

- Type: String
- Usage: Stores the raw user input of a food item from an editable textbox that is used to define the “food”(s) they want to retrieve and view from the csv.
- Functions: Search_All_Foods, Search_Single_Food and Weight_Calculator.

User_Nutrient_Input

- Type: String
- Usage: Functions identically to User_Food_Input but it's designed to store single nutrient type instead of a “food” that the user wants to filter their results by.
- Functions: Nutrition_Level_Filter and Nutrition_Range_Filter.

User_Weight_Input

- Type: Floating point
- Usage: Used to store the user defined weight that they want to scale the nutritional information by.
- Functions: Weight_Calculator

User_Nutrient_Level

- Type: String
- Usage: Stores the user's desired “level” of a specific nutrient as “Low”, “Mid” or “High”. It is used by the function to filters through all items in the csv to only show results that match the desired “level”.
- Functions: Nutrition_Level_Filter

User_Nutrient_Min

- Type: Floating point
- Usage: Stores the absolute minimum nutritional value that the user specifies for a specific nutrient. Is used to filter out any food from the results that has the defined nutrient's value below the min.
- Functions: Nutrition_Range_Filter

User_Nutrient_Max

- Type: Floating point
- Usage: Identical to User_Nutrient_Min but is used to define the maximum instead of the minimum and it's used to filter out any food from the results that has the defined nutrient's value above the max.
- Functions: Nutrition_Range_Filter

SFood_Arr[]

- Type: 1D array

- Usage: Same as Food_Arr[] but is designed to only store 1 row of data because it is used in functions that only work with one specific “food” rather than a list of them.
- Functions: Search_Single_Food and Weight_Calculator.

Calc_Food_Arr[]

- Type: 1D array
- Usage: Intends to build on Food_Arr[] by storing the results of the Weight_Calculator after the contents of Food_Arr[] have been scaled using User_Weight_Input.
- Functions: Weight_Calculator

Food_Arr[][]

- Type: 2D Array
- Usage: It stores the catalogue of results that match the user’s query that is listed on the user interface when the search has been completed.
- Functions: Search_All_Foods, Nutrition_Level_Filter and Nutrition_Range_Filter.

3.2.3 Detailed Design

ReadLines

START

Open Food_Nutrition_Dataset.csv for reading

Initialise array CSV_Arr[][]

For each Line in Food_Nutrition_Dataset.csv

 Line = Read line of “Food_Nutrition_Dataset.csv”

 Split Line by comma delimiter

 Insert line into one line of CSV_Arr[][]

End For

Close Food_Nutrition_Dataset.csv

Return CSV_Arr[][]

END

Search_All_Foods

START

Initialise array CSV_Arr[][] = Readlines(Food_Nutrition_Dataset.csv)

Initialise string User_Food_Input = system.readtextinput()

Initialise array Food_Arr[][]

Initialise int i = 0

For each index i from 0 to CSV_Arr[][].RowCount() – 1

 If the element at CSV_Arr[i][0] is equal to User_Food_Input

 Add the row CSV_Arr[][] at index i to Food_Arr[][]

 End If

```
    i += 1
End For

Return Food_Arr[]
END
```

Search_Single_Food

```
START
Initialise array CSV_Arr[][] = Readlines(Food_Nutrition_Dataset.csv)
initialise string User_Food_Input = system.readtextinput()
Initialise array SFood_Arr[]
Initialise int i = 0

For each index i from 0 to CSV_Arr[0].RowCount() - 1
    If the element at CSV_Arr[i][0] is equal to User_Food_Input
        Add the row CSV_Arr[i] at index i to SFood_Arr[]
    Return SFood_Arr[]
End If
    i += 1
End For

Return -1
END
```

Nutrition_Breakdown

```
START
initialise string User_Food_Input = system.readtextinput()
initialise array SFood_Arr[] = Search_Single_Food(User_Food_Input)
initialise int i = 0

For each index i from 0 to SFood_Arr[0].RowCount() - 1
    Add SFood_Arr[i] to visible table
    Add SFood_Arr[i] value to pie chat
    Add SFood_Arr[i] value to bar graph
End For

Return pie chart, bar graph
END
```

Weight_Calculator

```
START initialise string User_Food_Input = system.readtextinput()
initialise float User_Weight_Input = float(system.readtextinput())
initialise array SFood_Arr[] = Search_Single_Food(User_Food_Input)
```

```

initialise array Calc_Food_Arr[]
initialise int i = 0

For each index i from 0 to SFood_Arr[].RowCount() – 1
    Calc_Food_Arr[i] = SFood_Arr[i] * (User_Weight_Input /100)
End For

Return Calc_Food_Arr[]
END

```

Nutrition_Level_Filter

```

START Initialise CSV_Arr[][] = Readlines(Food_Nutrition_Dataset.csv)
initialise string User_Nutrient_Input = system.readtextinput()
initialise string User_Nutrient_Level = system.readtextinput()
initialise array Food_Arr[][]
initialise int i = 0
initialise int j = 0

While global.CSV_Arr[0][j] is not NULL
    If global.CSV_Arr[0][j] contains User_Nutrient_Input
        Break
    End If
    j += 1
End While

For each index i from 0 to CSV_Arr[i][j].RowCount() – 1
    If the element at CSV_Arr[i][j] meets level of User_Nutrient_Level
        Add the row CSV_Arr[i][j] at index i to Food_Arr[i][j]
    End If
    i += 1
End For

Return Food_Arr[i][j]
END

```

Nutrition_Range_Filter

```

START Initialise array CSV_Arr[][] = Readlines(Food_Nutrition_Dataset.csv)
initialise float User_Nutrient_Input = system.readtextinput()
initialise float User_Nutrient_Min = float(system.readtextinput())
initialise float User_Nutrient_Max = float(system.readtextinput())
initialise array Food_Arr[][]
initialise int i = 0
initialise int j = 0

```



```

While global.CSV_Arr[0][j] is not NULL
    If global.CSV_Arr[0][j] contains User_Nutrient_Input      Break
End If
j += 1
End While
For each index i from 0 to CSV_Arr[0].RowCount() - 1
    If the element at CSV_Arr[i][j] is greater than User_Nutrient_Min and less than User_Nutrient_Max
        Add the row CSV_Arr[i] at index i to Food_Arr[i]
    End If
    i += 1
End For

Return Food_Arr[i]
END

```

4. User Interface Design

4.1 Structural Design

Structure: How will the software be structured?

The software is structured in a modular fashion, with each primary feature represented as a distinct module linked from the Home Page. This modular approach ensures that each feature is self-contained, simplifying maintenance, updates, and scalability.

- **Home Page:** Serves as the central hub, directing users to the core features.
- **Food Search:** Allows users to search for food items and view detailed nutritional information.
- **Nutrition Breakdown:** Provides visual representations (charts/graphs) of the nutritional content of selected foods.
- **Nutrition Range Filter:** Enables users to filter foods by specific nutrient ranges.
- **Nutrition Level Filter:** Allows users to filter foods by predefined nutritional levels (low, mid, high).
- **Weight Calculator:** Allows users to search for a food and input a weight to see the nutrient breakdown for that serving size.

Information Grouping: How will information be organized?

Information is organised into logical groups based on functionality:

Search & Retrieval

- Food Search: Users can enter a food name to retrieve its detailed nutritional profile.

Visualisation:

- Nutrition Breakdown: Users can view visual breakdowns of nutrients for a selected food.

Filtering:

- Nutrition Range Filter: Users can set nutrient ranges to filter foods.

- Nutrition Level Filter: Users can filter foods based on predefined nutrient levels.

Nutrient Calculation:

- Weight Calculator: Users can input a food item and specify a weight in grams to view the nutrient breakdown for that serving size. Retrieval:

Navigation: How will users navigate through the software?

Home Page as Central Hub:

Users begin at the Home Page, which provides links to all primary modules. A consistent navigation bar at the top of the interface ensures that users can easily switch between modules or exit the application.

Navigation Flow:

Home Page → Food Search: Enter food names and view detailed profiles. Home Page → Nutrition Breakdown: Select a food to view its nutritional breakdown in various visual formats. Home Page → Nutrition Range and Level Filter: Set ranges for specific nutrients and view filtered results and select nutrient levels and view foods that match. Home Page → Weight Calculator: Search for a food, input a weight in grams, and view the nutrient breakdown for that specific serving size.

Design Choices: Explain why these design choices were made.

Modularity:

Justification: The modular structure simplifies the tool's design, making it easier to update or expand specific features without affecting the entire system. Each module is self-contained and focused, ensuring that users can complete tasks efficiently.

User-Centric Design:

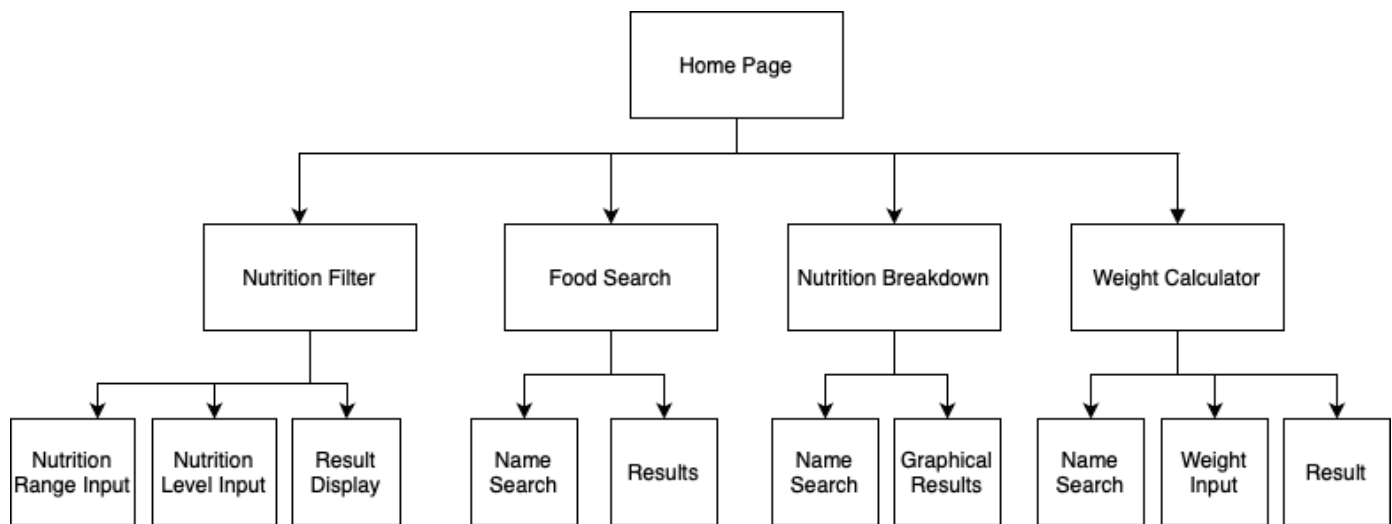
Justification: The structure is intuitive, with clear entry points for each function. This design choice prioritises ease of use, allowing users to accomplish their goals with minimal navigation or confusion.

Consistent Navigation:

Justification: A consistent navigation bar provides users with a sense of orientation within the tool, reducing cognitive load and preventing them from getting lost. Quick access to the Home Page or other modules ensures that users can move through the tool fluidly.

Visual Hierarchy:

Justification: The organisation of information into distinct groups based on functionality (Search & Retrieval, Visualisation, Filtering, Nutrient Calculation) allows users to quickly identify where to go for specific tasks. This reduces the time spent searching for features and improves overall usability.



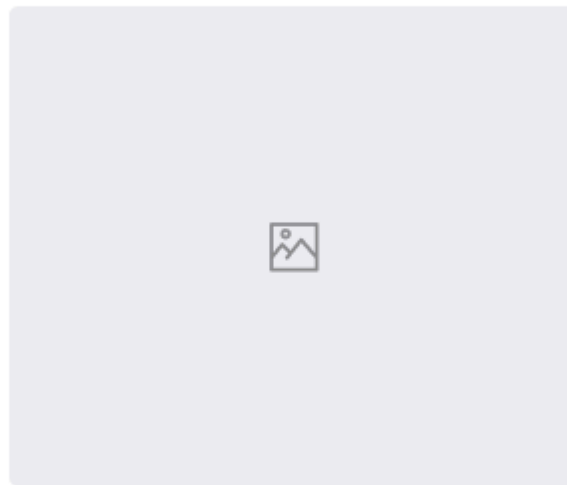
4.2 Visual Design

The wireframes correspond to different screens in the user journey as described in the flowchart. These include the welcome screen, nutrition filter, food search, nutrition breakdown, and weight calculator. Each wireframe represents a critical component of the application's interface.

4.2.1 Wireframe Breakdown and Justification

Wireframe 1: Welcome Screen

NUTRITIONAL FOOD APP



Exit

Continue

- **Visual Components:**
 - Two buttons: "Exit" and "Continue."
 - Centered image placeholder (likely for branding or a welcome illustration).
- **Functional Justification:**
 - **Exit button:** Allows users to quickly close the application, providing an easy and intuitive exit option.
 - **Continue button:** Prominently placed for immediate user interaction, guiding them toward the main interface.
 - **Image:** Placeholder may serve as a space for branding, adding visual engagement.
- **User Flow:** Simple decision-making process at the beginning, either to exit the app or proceed to the main interface.
- **Justification:** This screen minimizes decision fatigue by offering only two key choices, streamlining the user's first interaction with the application.

Wireframe 2: Nutrition Filter

LOGO

NUTRITION FILTER

FOOD SEARCH

NUTRITION BREAKDOWN

WEIGHT CALCULATOR

↩

NUTRITION FILTER

Nutrition Range

Nutrient

Proteins

Min Range

10

Max Range

20

Search

Nutrition Level

Nutrient

Proteins

Level

High

Search

Name	Fat	Carbohydrates	Sugars	Protein	Sodium
—	—	—	—	—	—
—	—	—	—	—	—
—	—	—	—	—	—

- **Visual Components:**
 - Two filtering sections: one for "Nutrition Range" and one for "Nutrition Level."
 - Data table displaying filtered results based on user input.
- **Functional Justification:**
 - **Nutrition Range filter:** Allows users to input a minimum and maximum range for a nutrient (e.g., proteins), giving precise control over the search results.
 - **Nutrition Level filter:** Offers predefined levels (e.g., low, mid, high), which simplifies the search for users who may not have specific numbers in mind.
 - **Search buttons:** Clear and prominent to trigger data retrieval after input.
 - **Data table:** Presents key nutritional metrics for the filtered foods, providing a clear overview.
- **User Flow:** Users can easily switch between filtering by range or by level and immediately see the results displayed below.
- **Justification:** This design supports users who want a more detailed or broader search of food data. It caters to different levels of expertise.

Wireframe 3: Food Search



FOOD SEARCH

Food

Name	Fat	Sugar	Sodium	Water	Actions
—	—	—	—	—	<input type="button" value="Breakdown"/> <input type="button" value="Caclulator"/>
—	—	—	—	—	<input type="button" value="Breakdown"/> <input type="button" value="Caclulator"/>
—	—	—	—	—	<input type="button" value="Breakdown"/> <input type="button" value="Caclulator"/>

- **Visual Components:**

- Input field for food name.
- Results table showing nutritional values.
- Actions buttons: "Breakdown" and "Calculator."

- **Functional Justification:**

- **Search field:** Users input the name of a food item (e.g., "Cheese"), making it a direct search functionality.
- **Results table:** Displays key metrics like fat, sugar, sodium, etc., allowing users to scan for the necessary data quickly.
- **Breakdown and Calculator buttons:** These offer further actions, leading users to more detailed analyses or weight calculations based on the selected food.

- **User Flow:** A simple search leads to clear and actionable next steps, allowing users to drill down into the data if needed.

- **Justification:** This wireframe provides a straightforward way for users to search for a specific food, see its data, and perform advanced actions like breakdowns and calculations, enhancing usability.

Wireframe 4: Nutrition Breakdown



NUTRITION BREAKDOWN

Food

Food

Cheese

▼

Search

- **Visual Components:**
 - Input field for food.
 - Graph for visualizing nutritional data.
- **Functional Justification:**
 - **Graphical representation:** Helps users understand data trends easily through visual breakdowns, enhancing comprehension.
 - **Food input:** Users can switch between different foods, offering flexibility in comparing different items.
- **User Flow:** After inputting a food item, users are presented with a graphical representation of the food's nutritional breakdown, making the data easier to interpret.
- **Justification:** Visual aids such as graphs are effective tools for data communication, especially in applications involving complex data like nutrition. This design choice helps simplify the information for non-expert users.

Wireframe 5: Weight Calculator



WEIGHT CALCULATOR

Food

Weight (gr)

Search

Head	Head	Head	Head
—	—	—	—
—	—	—	—
—	—	—	—

- **Visual Components:**
 - Input field for food and weight.
 - Results table with calculated values based on the input weight.
- **Functional Justification:**
 - **Weight input:** Allows users to calculate nutritional values based on a specific serving size, providing personalized data.
 - **Graph and results table:** These visually and numerically display the calculated values, ensuring users get both a detailed and high-level view.
- **User Flow:** Users input food and weight, then receive tailored nutritional data.
- **Justification:** This wireframe adds personalization by allowing users to adjust serving sizes and see the nutritional impact. It's especially useful for users tracking their intake or tailoring it to specific needs.

4.2.2 Consistency Across Wireframes

- **Navigation:** The top menu remains consistent, with navigation options like "Nutrition Filter", "Food Search", "Nutrition Breakdown", and "Weight Calculator". This ensures users can switch between features without confusion. The "Exit" button is consistently placed in the top-right corner of all screens except the Welcome Screen. This ensures that users have a clear and accessible option to exit the application at any time, contributing to a smooth user experience.
- **Visual Consistency:** The use of input fields, buttons, and tables follows a standardized format across wireframes, creating a good user experience.
- **Justification:** Maintaining consistency in user navigation and layout helps reduce the cognitive load on users and ensures that they can use the app efficiently across different features.