

Software Design Document

Project Name: Development of a Nutrition Tracking and Analysis Application called NutriMate

Group Number: 044

Team members

Student Number	Name
s5302345	Hartono Susanto
s5270370	Lindsay Soldevilla
s5327772	Hao Tsuneyama

Table of Contents

- Table of Contents
 - 1. System Vision
 - 1.1 Problem Background
 - 1.2 System capabilities/overview
 - 1.3 Potential Benefits
 - 2. Requirements
 - 2.1 User Requirements
 - 2.2 Software Requirements
 - 2.3 Use Case Diagrams
 - 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

- The nutritional base of Better Life is overly simplistic and challenging for customers to navigate, leading many to abandon it in favor of other tools. This base database is intended to provide users with nutritional information about various products, but its current format does not facilitate easy access and understanding. As a result customer may struggle to find the information they need, which diminishes their overall experience and satisfaction with this service. A more user-friendly data analysis and visualization tool could significantly enhance the usability of this nutritional database, making it to engage with the platform more consistently. However, due to poor design and functionalities, many of this target user are not effectively served by current database.
- **Dataset:** The dataset contains nutritional information for various products, likely including details such as calories value, vitamins, and minerals.
- **Data Input:** Users can only input basic search queries to product names.
- **Data Output** The system likely provides plain text or simple tabular data of nutritional information, with no visual representation.
- **Target Users:** The intended users for NutriMate are user who are seeking for nutritional information about products , research, educators, and Healthcare Professionals. These may include health-conscious individuals and people with dietary restrictions.

1.2 System capabilities/overview

The business will benefit from this new capabilities increasing efficiency, convenience and improve sales.

1. User Authentication and Personalization
 - Secure login
 - Personalised dashboard to each user
 - Customisable daily food intake
2. Food database and search function
 - Extensive database of food with nutritional information
3. Data Visualization
 - Interactive pie chart and bar graphs for food intake such as
 - Visual representation for macronutrient
4. Detailed Nutritional Analysis
 - Comprehensive breakdown of macronutrients (proteins, carbohydrates, fats) and micronutrients
5. Daily Nutrition Calculator:
 - Customizable daily food intake
 - Calculate total nutritional consumption
 - Visual tracking with chart and graphs for day.

1.3 Benefit Analysis

- We anticipate a 30% reduction in user drop-off within the first six months by introducing an intuitive user interface and enhanced search features
- Improve customer satisfaction: The addition of visual nutritional breakdowns (e.g., pie charts for macronutrients) will provide users with easy-to-understand information, potentially increasing user satisfaction scores from 2 to 4.5 out of 5.
- The improved database can integrate with grocery store inventory systems, allowing users to find nutritional information for products available in their local stores, opening up partnership opportunities with chains like Woolworths or Coles.
- With enhanced features and user retention, we project a 20% increase in premium subscriptions within the first year, translating to an estimated \$500,000 additional annual revenue.
- By providing accurate and comprehensive nutritional information, NutriMate aims to become the go-to resource for dietary information, potentially increasing our app store rating from 3.0 to 4.5 stars.
- The unique Nutrition Level Filter feature will set NutriMate apart from competitors, allowing users to easily find foods that match their specific nutritional needs, a feature currently not offered by major competitors.
- The improved system will collect user behavior data, allowing NutriMate to identify trends and potentially develop new products or features based on user preferences.

2. Requirements

2.1 User Requirements

Catalina, a 32-year-old marketing executive, leads a busy life balancing her career and personal health goals. She's health-conscious but struggles to maintain a consistent diet due to her hectic schedule. Recently diagnosed with mild lactose intolerance, she's determined to make better food choices while managing her condition.

Catalina opens the NutriMate app on her laptop during her lunch break. The personalized dashboard greets her with "Welcome back, Catalina!" and displays her daily nutritional calculator. She quickly searches for "steak sandwich" and adds it to her lunch entry. The app instantly updates her daily totals, showing her remaining calorie allowance and nutrient intake.

Noticing her protein intake is low, Catalina decides to add a lactose-free Greek yogurt from one of her favorite brands to her afternoon snack. She uses the search feature to find it and selects it. The system then recalculates her nutritional intake, showing that she has now met her protein goal for the day.

Before closing the app, Catalina checks the weekly overview, which shows she's been consistent with her calorie goals but could improve her fiber intake. She makes a mental note to include more vegetables in her dinner.

User Needs:

- Quick and easy food search functionality to find nutritional information.
- Personalized dashboard with daily nutritional goals and progress tracking.
- Real-time updating of nutritional intake as foods are added to the calculator.

- Visual representations of macronutrient balance (e.g., pie charts, progress bars).
- Weekly overview of nutritional trends to monitor progress over time.
- Suggestions for improving nutrient intake based on personal goals.
- Ability to plan future meals and snacks easily.
- Customizable alerts for exceeding or not meeting specific nutritional targets.
- A recipe suggestion feature that recommends meals based on the user's nutritional goals.

2.2 Software Requirements

Define the functionality the software will provide. This section should list requirements formally, often using the word "shall" to describe functionalities.

R1. User Interface and Navigation

- R1.1 The application shall display a personalized dashboard upon user login.
- R1.2 The application shall provide a search function for users to find foods.

R2. Nutritional Information

- R2.1 The system shall maintain a database of foods and their nutritional information.
- R2.2 The system shall display detailed nutritional information for each food item, including calories, macronutrients, and micronutrients.

R3. Calorie Calculator

- R3.1 The application shall allow users to set personalized nutritional calories' intake.
- R3.2 The application shall allow users to add food items to their daily calculator.
- R3.3 The system shall automatically calculate and update the user's daily nutritional totals when food items are added or removed from the log.
- R3.4 The system shall track the user's progress towards their nutritional goals in real-time.

R4. Visualization

- R4.1 The application shall provide visual representations (e.g., charts, graphs) of the user's nutritional intake.
- R4.2 The system shall display a weekly overview of the user's nutritional trends.
- R4.3 The system shall show detailed nutritional information for each food item when selected by the user.

R5. Advanced Nutritional Filtering

- R5.1 The system shall provide a Nutrition Range Filter feature.
 - R5.1.1 The system shall allow users to select a specific nutrient.
 - R5.1.2 The system shall enable users to input minimum and maximum values for the selected nutrient.
 - R5.1.3 The system shall display a list of foods that fall within the specified nutritional range.
- R5.2 The system shall provide a Nutrition Level Filter feature.

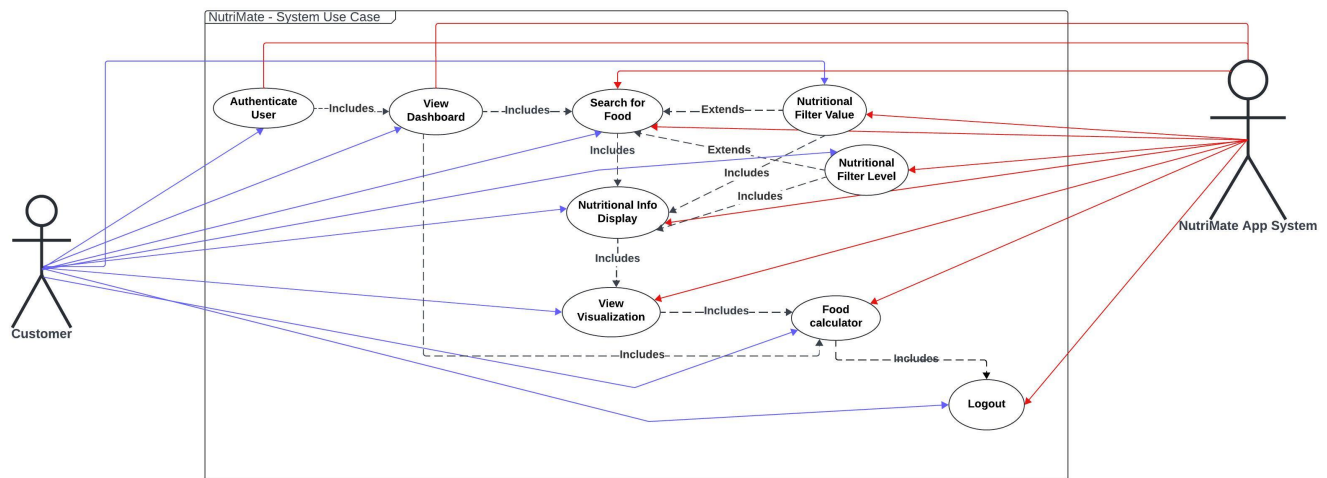
- R5.2.1 The system shall allow users to filter foods by nutritional content levels (low, mid, and high) for specific nutrients including fat, protein, carbohydrates, sugar, and nutritional density.
- R5.2.2 The system shall define the levels as follows: Low: Less than 33% of the highest value in the database for that nutrient. Mid: Between 33% and 66% of the highest value in the database for that nutrient. High: Greater than 66% of the highest value in the database for that nutrient.
- R5.2.3 The system shall display foods that match the selected nutritional level criteria.

R6. User Data Management

- R6.1 The system shall securely store user profile information and nutritional data.

2.3 Use Case Diagram

The use case diagram visually represents the interactions between the user and the NutriMate system. It includes various functionalities that users can access, demonstrating the application's capabilities.



2.4 Use Cases

Include at least 5 use cases, each corresponding to a specific function.

Use Case ID	UC-01
Use Case Name	Authenticate User
Actors	Customer
Description	The customer opens the app in the laptop and enter their password to authenticate their identity and access their account.
Flow of Events	1. Customer open the app in their laptop.
	2. System prompts for password.

Use Case ID	UC-01
	3. Customer enters password.
	4. System verifies password with the NutriMate system.
	5. System grants access to the customer.
Alternate Flow	If the password is incorrect, the system prompts the customer to re-enter the password.
Post-condition	The user is logged into their account

Use Case ID	UC-03
Use Case Name	Search for food
Actors	Customer
Description	The user search for a specific food item and retrieves the detail nutritional information
Preconditions	The system must have a database of food and their nutritional information
	The user is on the main screen and has access to the search bar
Trigger	The user types a food item into the search bar and press search
Flow of Events	The user opens the program and sees the search bar
	The user types a name of the food item 'vegetable fruit juice' into the search bar
	The user press the search button
	The system queries the database and retrieves the nutritional information for the food item
	The system display nutritional details in pie charts & bar graphs showing
Alternate Flow	If the food is not found, the system display the following message "No results found" and prompts the user to try again
Post-condition	The user successfully retrieved the data.

Use Case ID	UC-04
Use Case Name	Filter by Nutrition Range
Actors	User

Use Case ID	UC-04
Description	The user filters foods by selecting a nutrient and specifying a range of values. The system displays foods that match the criteria.
Precondition	The system must allow users to select nutrients and specify ranges.
	The system must be able to retrieve food items that fall within the selected range.
Trigger	The user selects a nutrient and specifies a range, then presses the "Apply Filter" button.
Flow of Events	The user navigates to the "Filter by Nutrient Range" option.
	The user selects a nutrient (e.g., "protein").
	The user inputs a minimum value (e.g., "5g") and a maximum value (e.g., "20g").
	The user presses the "Apply Filter" button.
	The system queries the database and retrieves a list of foods that meet the nutrient range criteria.
	The system displays the filtered list of foods to the user.
Alternate Flow	If no foods match the criteria, the system notifies the user and prompts the user to adjusting the range.
Post-condition	The user successfully filters and views foods that meet their specified nutrient range.

Use Case ID	UC- 05
Use Case Name	Filter by Nutritional Level
Actors	User
Description	The user filters foods based on nutrient levels Low for a selected nutrient Sodium.
Precondition	The system categorizes nutrient levels (low, medium, high) based on predefined thresholds.
	The user has access to the filtering option in the interface.
Trigger	The user selects a nutrient and chooses a level (Low, Mid, or High), then presses the "Apply Filter" button.
Flow of Events	The user navigates to the "Filter by Nutritional Level" section.
	The user selects a nutrient (e.g., "Vitamin A").

Use Case ID	UC- 05
	The user selects one of the levels.
	The user presses the "Apply Filter" button.
	The system queries the database and retrieves a list of foods that meet the criteria.
Alternate Flow	If no foods match the criteria, the system notifies the user and prompts the user to adjust the criteria
Post-condition	The user successfully filters foods based on the nutrient levels they selected

Use Case ID	UC-06
Use Case Name	Daily nutrition calculator
Actors	User
Description	The user inputs foods they have consumed throughout the day and calculates their total daily nutrient intake.
Trigger	The user selects the foods they have consumed during the day and presses the "Add to my day" button to determine their daily nutritional intake.
Precondition	The system must store food items and allow users to input their consumption data.
	The system must calculate and display total nutrient intake.
	The user need to be inside the food information to add
Flow of Events	The user presses "Add to my day" button.
	The system sums the nutritional content (calories, protein, fats, carbohydrates, etc.) for the entire day.
	The system displays the total daily nutrient intake
Alternate Flow	If the user exceeds or falls short of certain nutrient, the system can notify them.
Post-condition	The user successfully calculates their total daily nutrient intake and compares it to recommended values.

Use Case ID	UC-07
Use Case Name	Logout
Actors	User

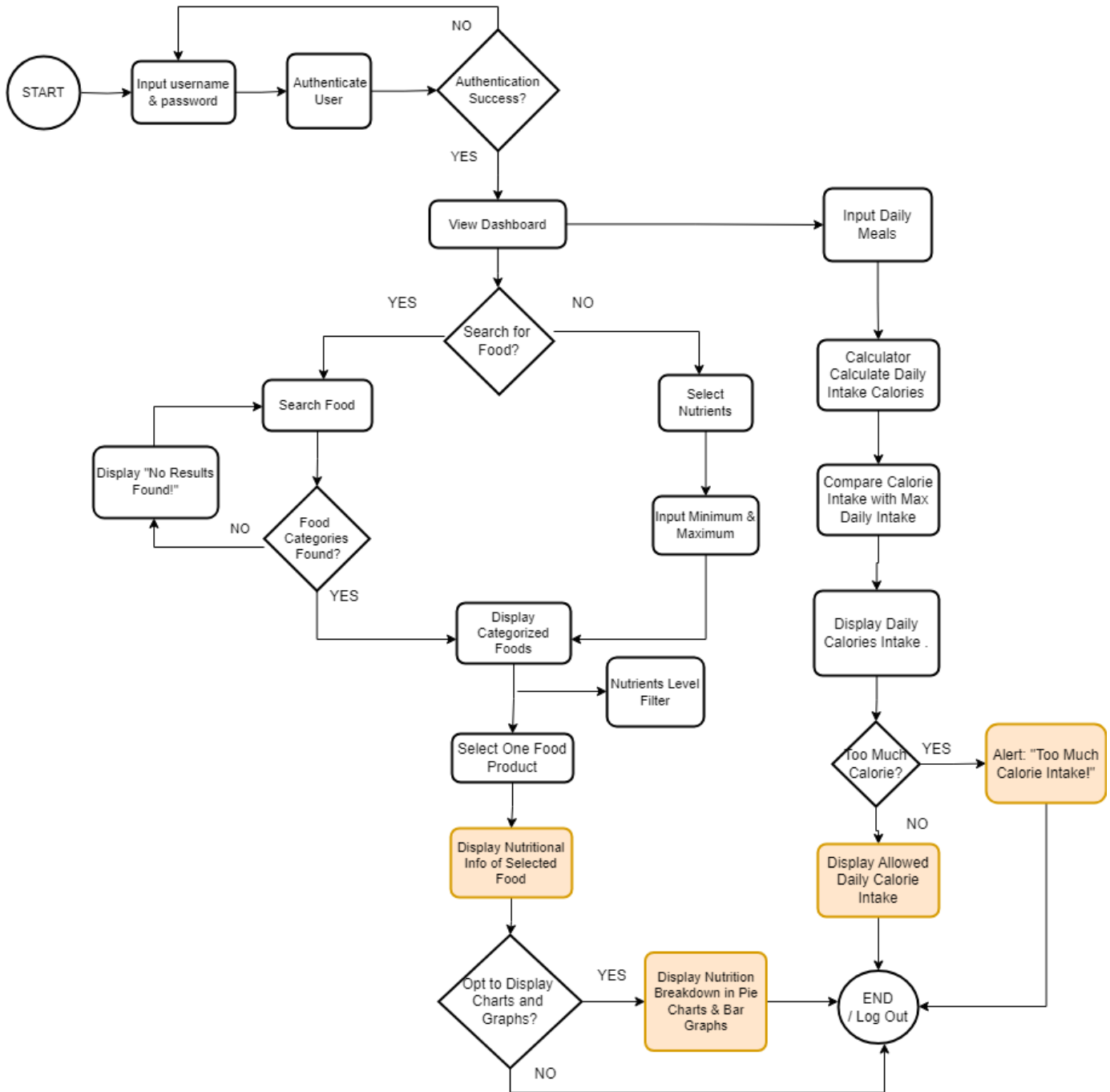
Use Case ID	UC-07
Description	The user logs out of the app
Trigger	The customer selects "Logout."
Precondition	The customer is authenticated
Flow of Events	The customer selects "Logout"
	The system ends the session
Post-condition	The user is logged out

3. Software Design and System Components

3.1 Software Design

This section explains how the NutriMate App is set up and works. A flowchart shows the main parts and how they connect, making it clear how data flows through the system.

NutriMate App Flowchart:



3.2 System Components

3.2.1 Functions

This section lists the main functions used in the NutriApp software. Each function's purpose is explained, along with the inputs it takes, what it returns, and any side effects. These functions are key to making the app work, like searching for foods, filtering nutrition data, calculating users' calorie and nutrient intake and needs, and showing nutrient breakdowns visually.

3.2.1.1 find_food(name)

- **Description:** Looks up the food database for an item by name and returns matching results.

- **Input Parameters:**
 - name (str): The food name to search for.
 - food_data_list (list): A list of dictionaries that hold food items and their nutritional details.
- **Return Value:** Returns a list of dictionaries with the food items that match the name.
- **Side Effects:** None. This function doesn't change any global variables or the data it's given.

3.2.1.2 get_nutrition(name)

- **Description:** Gets the nutritional details for a specific food item.
- **Input Parameters:**
 - name (str): The name of the food item to get details for.
 - food_data_list (list): A list of dictionaries that hold food items and their nutritional details.
- **Return Value:** Returns a dictionary with the nutrition details for the selected food item.
- **Side Effects:** None. This function doesn't change any global variables or the data it's given.

3.2.1.3 filter_by_range(nutrient, min_val, max_val)

- **Description:** Filters the food database to find items within a specific range of a nutrient.
- **Input Parameters:**
 - nutrient (str): The nutrient to filter by (e.g., 'calories', 'protein').
 - min_val (float): The minimum value for the nutrient range.
 - max_val (float): The maximum value for the nutrient range.
 - food_data_list (list): A list of dictionaries that hold food items and their nutritional details.
- **Return Value:** Returns a list of dictionaries with food items that fit within the given nutrient range.
- **Side Effects:** None. This function doesn't change any global variables or the data it's given.

3.2.1.4 filter_by_level(level)

- **Description:** Filters food items based on their nutrient levels (low, mid, high).
- **Input Parameters:**
 - level (str): The nutrient level to filter by (e.g., 'low', 'mid', 'high').
 - nutrient_levels (dict): A dictionary with nutrient levels for various foods.
 - food_data_list (list): A list of dictionaries that hold food items and their nutritional details.
- **Return Value:** Returns a list of dictionaries with food items that match the chosen nutrient level.
- **Side Effects:** None. This function doesn't change any global variables or the data it's given.

3.2.1.5 calc_calories(user_data)

- **Description:** Calculates the daily calorie needs for a user based on their personal info.
- **Input Parameters:**
 - user_data (dict): A dictionary with user info like weight, height, age, and activity level.
- **Return Value:** Returns a float value that represents the user's daily calorie requirement.
- **Side Effects:** None. This function only uses the input data and doesn't affect anything else.

3.2.1.6 calc_nutrients(user_data)

- **Description:** Calculates the daily recommended nutrient intake for a user.

- **Input Parameters:**
 - `user_data` (dict): A dictionary with user info like weight, height, age, and dietary goals.
- **Return Value:** Returns a dictionary with the recommended daily intake of nutrients (e.g., protein, fat, carbs).
- **Side Effects:** None. This function only uses the input data and doesn't affect anything else.

3.2.1.7 `load_food_csv(source)`

- **Description:** Loads food data from a CSV file into the app's internal data structures.
- **Input Parameters:**
 - `source` (str): The file path or location of the CSV file with food and nutrient data.
- **Return Value:** Returns a list of dictionaries where each represents a food item with its nutritional details.
- **Side Effects:** This function fills the internal food data list with info from the CSV file.

3.2.1.8 `load_nutrient_csv(source)`

- **Description:** Loads nutrient descriptions from a CSV file into the app.
- **Input Parameters:**
 - `source` (str): The file path or location of the CSV file with nutrient descriptions.
- **Return Value:** Returns a dictionary where each key is a nutrient name and the value is its description.
- **Side Effects:** This function fills the internal nutrient description data with info from the CSV file.

3.2.1.9 `update_food_csv(new_data)`

- **Description:** Updates the current food data with new or changed food items.
- **Input Parameters:**
 - `new_data` (list): A list of dictionaries where each one represents a new or updated food item.
- **Return Value:** Returns an updated list of food items that includes the new data.
- **Side Effects:** This function updates the internal food data list by adding or modifying items.

3.2.1.10 `update_nutrient_csv(new_data)`

- **Description:** Updates the current nutrient descriptions with new or changed info.
- **Input Parameters:**
 - `new_data` (dict): A dictionary where each key is a nutrient and the value is the updated description.
- **Return Value:** Returns an updated dictionary of nutrient descriptions.
- **Side Effects:** This function updates the internal nutrient description data with the new info.

3.2.1.11 `prep_visual_data(food_item)`

- **Description:** Prepares the nutritional info of a selected food item for visualization.
- **Input Parameters:**
 - `food_item` (dict): A dictionary that represents the food item with its nutritional details.
- **Return Value:** Returns a dictionary or array set up for use in visualizations like charts.
- **Side Effects:** None. This function only prepares data for visualization and doesn't change anything else.

3.2.1.12 `make_pie(data)`

- **Description:** Creates a pie chart from the provided nutritional data.
- **Input Parameters:**
 - data (dict or array): The structured data ready for visualization, representing nutrient distribution.
- **Return Value:** Returns a pie chart or directly displays the chart.
- **Side Effects:** None. This function only visualizes the data without changing other parts of the app.

3.2.1.13 make_bar(data)

- **Description:** Creates a bar chart to show the nutrient breakdown of a food item.
- **Input Parameters:**
 - data (dict or array): The structured data ready for visualization, representing nutrient amounts.
- **Return Value:** Returns a bar chart or directly displays the chart.
- **Side Effects:** None. This function only visualizes the data without changing other parts of the app.

3.2.2 Data Structures / Data Sources

This section explains the data structures and external sources used in NutriApp. It covers what each data structure is, how it's used in the app, and which functions work with it. These data structures are important for handling and processing the nutrition info, user data, and visual outputs that the app relies on.

3.2.2.1 Food Data List

- Type: List of dictionaries
- Usage:
 - Stores food items, each with details like calories, protein, and fat in a dictionary format.
 - Used in the Food Search feature to find foods by name and show their nutritional info.
- Functions:
 - find_food(name): Finds and returns foods that match the name given.
 - get_nutrition(name): Gets the nutrition details for the selected food.
 - filter_by_range(nutrient, min_val, max_val): Filters foods based on a range of nutrient values.
 - filter_by_level(level): Filters foods by their nutrient levels like low, mid, or high.

3.2.2.2 Nutrient Range Filter

- Type: Dictionary
- Usage:
 - Keeps the min and max values for each nutrient that the user sets.
 - Used in the Nutrition Range Filter feature to filter foods by these nutrient values.
- Functions:
 - set_range(nutrient, min_val, max_val): Sets the range for a nutrient.
 - apply_range_filter(): Filters the foods using the current range settings.

3.2.2.3 User Input Data

- Type: Dictionary
- Usage:
 - Stores user data, like weight and activity level, for doing different calculations.

- Used in the Calculator feature to give personalized nutrition info.
- Functions:
 - `calc_calories(user_data)`: Calculates how many calories the user needs daily.
 - `calc_nutrients(user_data)`: Calculates how much of each nutrient the user needs.

3.2.2.4 Food Nutrition Levels

- Type: Set
- Usage:
 - Stores preset levels (low, mid, high) for nutrients based on percentage ranges.
 - Used in the Nutrition Level Filter feature to group foods by their nutrient levels.
- Functions:
 - `get_level(food_item, nutrient)`: Figures out if a food's nutrient level is low, mid, or high.
 - `filter_by_level(level)`: Lists foods that match the chosen nutrient level.

3.2.2.5 Food Database

- Type: External Data Source (CSV files)
The main source of data for the app, consisting of two CSV files. One file contains food names and their nutrients. The other file provides descriptions for each nutrient.
- Usage:
 - Used to fill the Food Data List with food items and their nutritional values and to add descriptions for each nutrient.
- Functions:
 - `load_food_csv(source)`: Loads food and nutrient data from the CSV file into the app.
 - `load_nutrient_csv(source)`: Loads nutrient descriptions from the CSV file.
 - `update_food_csv(new_data)`: Updates the existing food data with new or changed items.
 - `update_nutrient_csv(new_data)`: Updates the nutrient descriptions with new or changed details.

3.2.2.6 Visualization Data

- **Type:** List of dictionaries
- **Usage:**
 - Holds data that's ready to be shown visually (e.g., bar charts, pie charts).
 - Used in the Nutrition Breakdown feature to show nutrient distributions visually.
- **Functions:**
 - `prep_visual_data(food_item)`: Prepares the nutrition info of a food item for visualization.
 - `make_pie(data)`: Creates a pie chart from the data.
 - `make_bar(data)`: Creates a bar chart to show the nutrient breakdown.

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.

3.2.3 Detailed Design

This section presents the detailed design of the key functions described in Section 3.2.1. For each function that interacts with the data structures outlined in Section 3.2.2, pseudocode is provided to clarify the logical flow and operations.

3.2.3.1 Pseudocode for find_food(name)

Purpose: Searches for food items in the database by name.

```
FUNCTION find_food(name, food_data_list)
    CREATE an empty list to store matching food items
    FOR each food item in the list
        IF the name matches the food item
            ADD the food item to the list
    RETURN the list of matching food items
```

3.2.3.2 Pseudocode for get_nutrition(name)

Purpose: Retrieves nutritional details of a specific food item.

Pseudocode:

```
FUNCTION get_nutrition(name, food_data_list)
    FOR each food item in the list
        IF the name matches the food item
            RETURN the nutritional details of the food item
    RETURN nothing if no match is found
```

3.2.3.3 Pseudocode for filter_by_range(nutrient, min_val, max_val)

Purpose: Filters food items by a specified nutrient range.

Pseudocode:

```
FUNCTION filter_by_range(nutrient, min_val, max_val, food_data_list)
    CREATE an empty list to store filtered food items
    FOR each food item in the list
        IF the nutrient value is within the specified range
            ADD the food item to the list
    RETURN the list of filtered food items
```

3.2.3.4 Pseudocode for filter_by_level(level)

Purpose: Filters food items based on nutrient levels (low, mid, high).

Pseudocode:

```
FUNCTION filter_by_level(level, nutrient_levels, food_data_list)
    CREATE an empty list to store filtered food items
```



```
FOR each food item in the list
  IF the nutrient level matches the specified level
    ADD the food item to the list
RETURN the list of filtered food items
```

3.2.3.5 Pseudocode for calc_calories(user_data)

Purpose: Calculates the daily calorie needs for a user.

Pseudocode:

```
FUNCTION calc_calories(user_data)
  GET the user's personal data (weight, height, age, activity level)
  CALCULATE the daily calorie needs using the user's data
  RETURN the calculated calorie needs
```

3.2.3.6 Pseudocode for calc_nutrients(user_data)

Purpose: Calculates the recommended daily intake of nutrients for a user.

Pseudocode:

```
FUNCTION calc_nutrients(user_data)
  GET the user's personal data (weight, dietary goals)
  CALCULATE the recommended intake for protein, fat, and carbohydrates
  RETURN the calculated nutrient needs
```

3.2.3.7 Pseudocode for load_food_csv(source)

Purpose: Loads food data from a CSV file.

Pseudocode:

```
FUNCTION load_food_csv(source)
  READ the CSV file
  STORE the data in a list of dictionaries
  RETURN the list of food items
```

3.2.3.8 Pseudocode for load_nutrient_csv(source)

Purpose: Loads nutrient descriptions from a CSV file.

Pseudocode:

```
FUNCTION load_nutrient_csv(source)
  READ the CSV file
```

```
STORE the descriptions in a dictionary
RETURN the nutrient descriptions
```

3.2.3.9 Pseudocode for update_food_csv(new_data)

Purpose: Updates the existing food data with new or modified items.

Pseudocode:

```
FUNCTION update_food_csv(new_data, food_data_list)
  FOR each new or updated food item in new_data
    ADD or UPDATE the food item in the food data list
  RETURN the updated food data list
```

3.2.3.10 Pseudocode for update_nutrient_csv(new_data)

Purpose: Updates the existing nutrient descriptions.

Pseudocode:

```
FUNCTION update_nutrient_csv(new_data, nutrient_descriptions)
  FOR each new or updated nutrient in new_data
    ADD or UPDATE the nutrient in the nutrient descriptions
  RETURN the updated nutrient descriptions
```

3.2.3.11 Pseudocode for prep_visual_data(food_item)

Purpose: Prepares data for visualization.

Pseudocode:

```
FUNCTION prep_visual_data(food_item)
  EXTRACT relevant nutritional information from the food item
  ORGANIZE the data for visualization
  RETURN the prepared data
```

3.2.3.12 Pseudocode for make_pie(data)

Purpose: Creates a pie chart from nutritional data.

Pseudocode:

```
FUNCTION make_pie(data)
  USE the data to create a pie chart
  RETURN the pie chart
```

3.2.3.13 Pseudocode for make_bar(data)

Purpose: Creates a bar chart to show nutrient breakdown.

Pseudocode:

```
FUNCTION make_bar(data)
    USE the data to create a bar chart
    RETURN the bar chart
```

4. User Interface Design

4.1 Structural Design

Present a structural design, a hierarchy chart, showing the overall interface's structure. Address:

- Structure:

The software will follow a hierarchical structure with a centralized Dashboard as the main control center, from which users can access key features such as User Profile, Food Logging, Summary, Charts & Analytics, Settings, and Network Settings. Each main section branches into more specific functionalities. For instance, User Profile contains personal information, daily goals, and health stats, while Summary provides details on calorie intake, nutrient breakdowns, and exercise information. Overall, this structure allows for easy scalability and maintainability, where new features or updates can be added as sub-sections under the relevant categories.

- Information

User Profile: Focuses on user-specific information such as personal details, health stats, and goals.

Food Logging: Allows users to input and track their meals, and view the nutrient content of their diet.

Summary: Displays aggregate information such as calorie intake, nutrient breakdown and giving users a comprehensive view of their progress.

Charts & Analytics: Provides visual insights like trends and nutrient analysis to help users better understand their dietary habits.

Settings and Network Settings: Manage preferences like notifications, themes, and network configurations.

- Navigation:

Users will navigate the app primarily through the Dashboard, which serves as the central hub. From the Dashboard, users can directly access all primary functions via buttons or tabs. Each primary function (e.g., User Profile, Food Logging, Summary) leads to more detailed sub-sections. Also, a back navigation button will always be present to allow users to return to the Dashboard or the previous section.

- Design Choices

The decision to have a Dashboard as the central control hub ensures users can easily access the most important features without feeling overwhelmed by too many options at once. This is especially helpful in a nutrition app, where users want to quickly log meals or view progress.

Grouping related information together, such as all calorie and nutrition tracking under Summary, helps users intuitively understand where to go for specific functions. It also aligns with how users think about health and nutrition—by logging their data, tracking it over time, and reviewing their progress. The clean, hierarchical structure makes it easy to add new features without disrupting the user experience. As the app grows, more sub-sections, like new types of data visualization, can be easily integrated without complicating the core navigation flow.

The design ensures users can quickly jump between different sections with minimal steps, making the app convenient for daily use. Keeping the navigation one-click away from the Dashboard is crucial for ensuring the app remains efficient for quick actions like logging meals.

Structural Design:



4.2 Visual Design

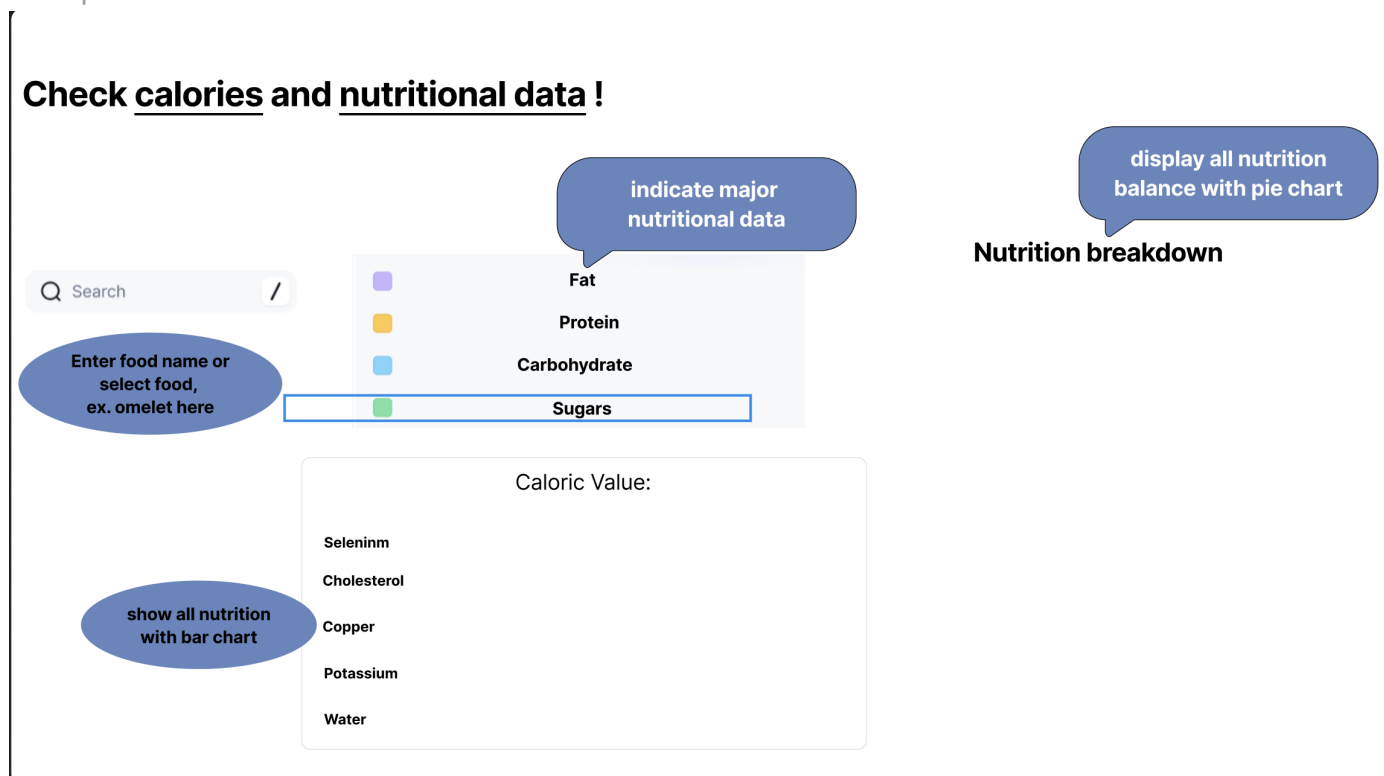
Include all wireframes or mock-ups of the interface. Provide a discussion, explanation, and justification for your design choices. Hand-drawn wireframes are acceptable.

The top section, "Check Calories and Nutritional Data," is designed to allow users to search for specific foods and view detailed nutritional information. It features a search bar where users can type the name of a food, such as "omelet," to fetch its nutritional data. The nutritional breakdown is visually represented using a pie chart, showing the distribution of key nutrients such as fats, proteins, carbohydrates, and sugars. Additionally, a bar graph is used to display detailed micronutrient information, including elements like Selenium, Cholesterol, and Potassium, enabling users to understand the concentrations of these nutrients in the food.

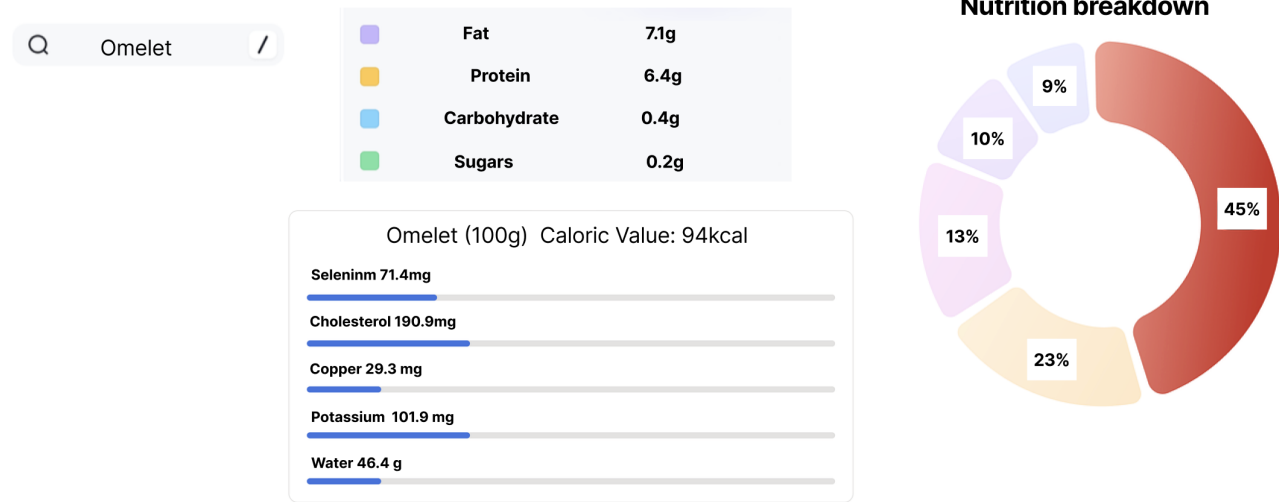
The middle section, titled "Find Food that Suits You," allows users to filter foods based on their nutritional needs, either by setting specific nutrient ranges or by filtering according to nutritional content levels. For the range filter, users can specify a nutrient, such as protein, and define minimum and maximum values, after which foods within that range are displayed. The level filter, on the other hand, enables users to categorize foods by "low," "mid," or "high" levels for various nutrients like fat, protein, carbohydrates, sugars, and nutritional density. Based on the user's selections, matching foods are displayed with accompanying images and brief descriptions, such as "low fat & sugars" or "high protein."

The bottom section, labeled "Summary," provides a recap of the user's health and dietary activity. This section shows the user's daily calorie intake using a progress circle (e.g., 1500 calories consumed). Additionally, it offers a nutritional breakdown of the user's food intake, showing consumed nutrients like carbohydrates, proteins, and fats, alongside suggested daily intake values for these nutrients.

Example:



Check calories and nutritional data !



Find food that suits you !

Nutrition Range Filter
Enter nutrition name

Q Search /

max: mg

min: mg

Nutrition Level Filter

Fat

Protein

Sugars

Carbohydrates

Nutritional density

Suggestion from NutriMate

Enter Nutrition name or input high mid, low

Recipe comes up here

Find food that suits you !

Nutrition Range Filter
Enter nutrition name


max: mg

min: mg

Nutrition Level Filter


Fat	low
Protein	high
Sugars	low
Carbohydrates	low
Nutritional density	high

Suggestion from NutriMate




Sunny Egg & Toast Avocado

low fat & sugars




Bowl of noodle with beef

high protein & sugars



Easy homemade beef burger

high protein



Half boiled egg sandwich

low fat & sugars

Summary

Enter your exercise data and what you had today!

Enter meal data and
app calculate

Tell me what you had !

When	What
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Summary

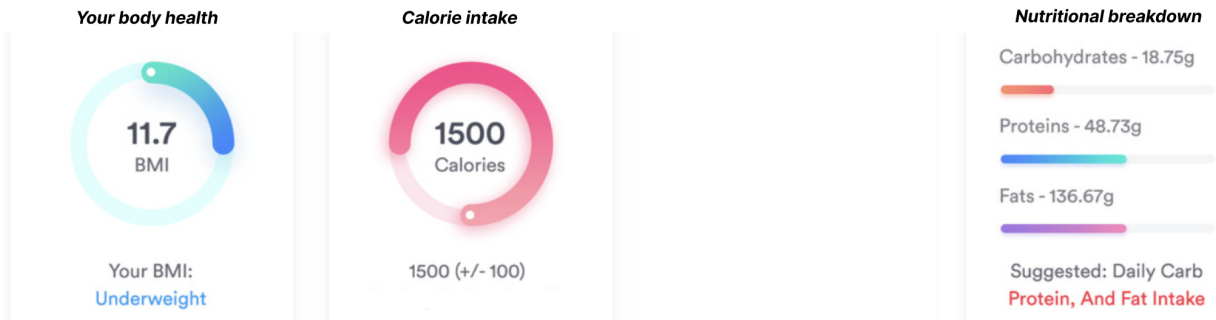
Enter your exercise data and what you had today!

Tell me what you had !

When	What
Breakfast ▾	Avo Toast ▾
Lunch ▾	Tomato Pasta ▾
Dinner ▾	Japanese Pancake ▾
+ Add another	

Summary

CHECK Calorie intake, Nutritional breakdown, and Calories burned!



message from NutriMate: Great job! You're almost at your goal!