

Software Design Document

Project Name: NutriPro

Group Number: 002

Team members

Student Number	Name
s2817538	Kate Baker
s5374957	I Wibawa
s5414931	Naveen Arakkal Nelson

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

1. System Vision

1.1 Problem Background

- Problem Identification: The system addresses the challenge of making informed dietary choices due to the complexity of nutritional data and the lack of user-friendly tools to analyze this information.
- Dataset: The system uses the Comprehensive Nutritional Food Database, which provides detailed nutritional information for a wide range of food items.
- Data Input/Output:
 - Input: User queries for food items, nutritional ranges, and dietary preferences.
 - Output: Nutritional information, visualizations, filtered food lists, and meal plans.
- Target Users: The system targets health-conscious individuals, dieters, nutritionists, and dietitians who need to analyse and understand nutritional information for various foods.

1.2 System capabilities/overview

NutriPro provides a comprehensive platform for users to explore, compare, and plan meals based on detailed nutritional information.

The system now includes the following key features:

- A central main menu for easy navigation to all major functions
- Advanced food search with filtering capabilities
- Detailed nutritional breakdown of individual food items
- Side-by-side comparison of up to three food items
- Meal planning functionality with weekly overview and daily nutritional summaries
- Data visualization through charts and graphs
- Data persistence for saving and loading meal plans

The system is structured around five main frames: Main Menu, Search, Food Details, Comparison, and Meal Planner, each accessible through intuitive navigation.

1.3 Potential Benefits (Expanded)

- Improved Dietary Decision-Making:
 - Impact: Users may see a 30% increase in adherence to recommended daily nutritional values within 3 months of consistent app use.
 - Example: A user trying to increase protein intake could easily identify and incorporate high-protein foods, potentially increasing their daily protein consumption by 20-25%.
- Enhanced Nutritional Understanding:

- Impact: In user surveys, we aim for a 50% increase in self-reported nutritional knowledge after 6 months of app usage.
- Example: Users will be able to identify the primary macronutrients in common foods and understand their daily recommended intake.
- Efficient Food Comparison:
 - Impact: Users can compare up to 3 food items simultaneously, reducing decision-making time by an estimated 40% compared to looking up each food separately.
 - Example: A user could quickly compare the nutritional profiles of chicken, fish, and tofu to make an informed protein choice for their meal.
- Streamlined Meal Planning:
 - Impact: We anticipate users saving an average of 2 hours per week on meal planning and grocery list creation.
 - Example: A user could generate a week-long meal plan that meets their nutritional goals in under 15 minutes, compared to manual planning that might take 2-3 hours.
- Educational Value:
 - Impact: Integration into nutrition education programs could lead to a 25% improvement in student test scores related to nutritional knowledge.
 - Example: Students could use the app to visualize and understand concepts like macronutrient balance and micronutrient diversity in different foods.
- Professional Time-Saving:
 - Impact: Nutritionists and dietitians using the app may save up to 5 hours per week on client meal plan creation and nutritional analysis.
 - Example: A dietitian could quickly generate and customize meal plans for multiple clients, reducing the time spent on manual calculations and food selection.
- Health Outcome Improvements:
 - Impact: Long-term users (1+ years) may see improvements in health markers such as a 10% reduction in cholesterol levels or a 5% decrease in body fat percentage.
 - Example: By using the app to maintain a balanced diet, a user with high blood pressure could potentially see a reduction in their blood pressure readings over time.

2. Requirements

2.1 User Requirements

1. Users shall be able to navigate between different sections of the application from a central main menu.
2. Users shall be able to search for foods and apply filters to refine search results.
3. Users shall be able to view detailed nutritional information for individual food items.
4. Users shall be able to compare nutritional information of up to three food items side-by-side.
5. Users shall be able to create, save, and load weekly meal plans.
6. Users shall be able to view a daily and weekly nutritional summary of their meal plans.
7. Users shall be able to add food items to their meal plan directly from the search or comparison views.

8. Users shall be able to generate visual representations (charts/graphs) of nutritional data.

Note: Since no specific client or user is assigned, we envision the primary users of this software to be health-conscious individuals, dieters, nutritionists, and dietitians who need to analyze and understand nutritional information for various foods quickly and efficiently.

2.2 Software Requirements

General

- **R1.1** The system shall provide a main menu with options to access Search, Compare, and Meal Planner functions.
- **R1.2** The system shall display a "Nutrition Tip of the Day" on the main menu.
- **R1.3** The system shall allow users to search for foods by name and apply nutritional filters.
- **R1.4** The system shall display detailed nutritional information for selected food items, including macronutrients and micronutrients.
- **R1.5** The system shall generate charts to visualize macronutrient breakdowns of food items.
- **R1.6** The system shall allow side-by-side comparison of up to three food items.
- **R1.7** The system shall generate comparison charts for multiple food items.
- **R1.8** The system shall provide a meal planner with a weekly overview and daily breakdown.
- **R1.9** The system shall calculate and display daily nutritional summaries based on planned meals.
- **R1.10** The system shall allow users to save and load meal plans.
- **R1.11** The system shall provide navigation options to return to the main menu from all sections.

Data Security

- **R2.1** The system shall encrypt all user data, including saved meal plans and user preferences, using industry-standard encryption methods.
- **R2.2** The system shall implement secure user authentication, requiring strong passwords with a minimum of 8 characters, including uppercase, lowercase, numbers, and special characters.
- **R2.3** The system shall automatically log out inactive users after 30 minutes of inactivity.

Performance

- **R3.1** The system shall return search results within 2 seconds for a database of up to 10,000 food items.
- **R3.2** The system shall generate meal plans within 5 seconds for a week-long plan.
- **R3.3** The system shall support concurrent usage of up to 1000 users without degradation in performance.

Scalability

- **R4.1** The system shall be designed to handle a 200% increase in the food database size without significant performance degradation.
- **R4.2** The system architecture shall support horizontal scaling to accommodate user growth up to 1 million active users.
- **R4.3** The system shall implement caching mechanisms to reduce database load for frequently accessed food items and nutritional data.

Usability

- **R5.1** The user interface shall be responsive and compatible with devices ranging from smartphones to desktop computers.
- **R5.2** The system shall provide clear error messages and guidance for all user inputs and interactions.
- **R5.3** The system shall include a help section and tooltips for new users to understand all features within 15 minutes of first use.

Data Integrity

- **R6.1** The system shall validate all user inputs to ensure data integrity in the food database and meal plans.
- **R6.2** The system shall maintain a change log for all updates to the food database, allowing for auditing and rollback if necessary.

2.3 Use Case Diagrams

Use Case Diagram Overview

The Use Case Diagram illustrates the main functionalities of the NutriPro application and how users interact with these features. Below is a brief explanation of the diagram:

Central Actor

- **User:** The primary actor in our system, represented by the stick figure.

Main Use Cases

1. **Access Main Menu:** Entry point for all other functionalities.
2. **Search Food:** Allows users to find specific food items in the database.
3. **View Food Details:** Provides detailed nutritional information for selected foods.
4. **Apply Filter Settings:** Enables users to refine their food searches based on nutritional criteria.
5. **Compare Foods:** Facilitates side-by-side comparison of multiple food items.
6. **Manage Meal Plan:** Encompasses creating, editing, and viewing meal plans.
7. **Generate Meal Plan:** An automated feature for creating meal plans based on user preferences.
8. **View Nutritional Summary:** Provides an overview of nutritional intake based on meal plans.

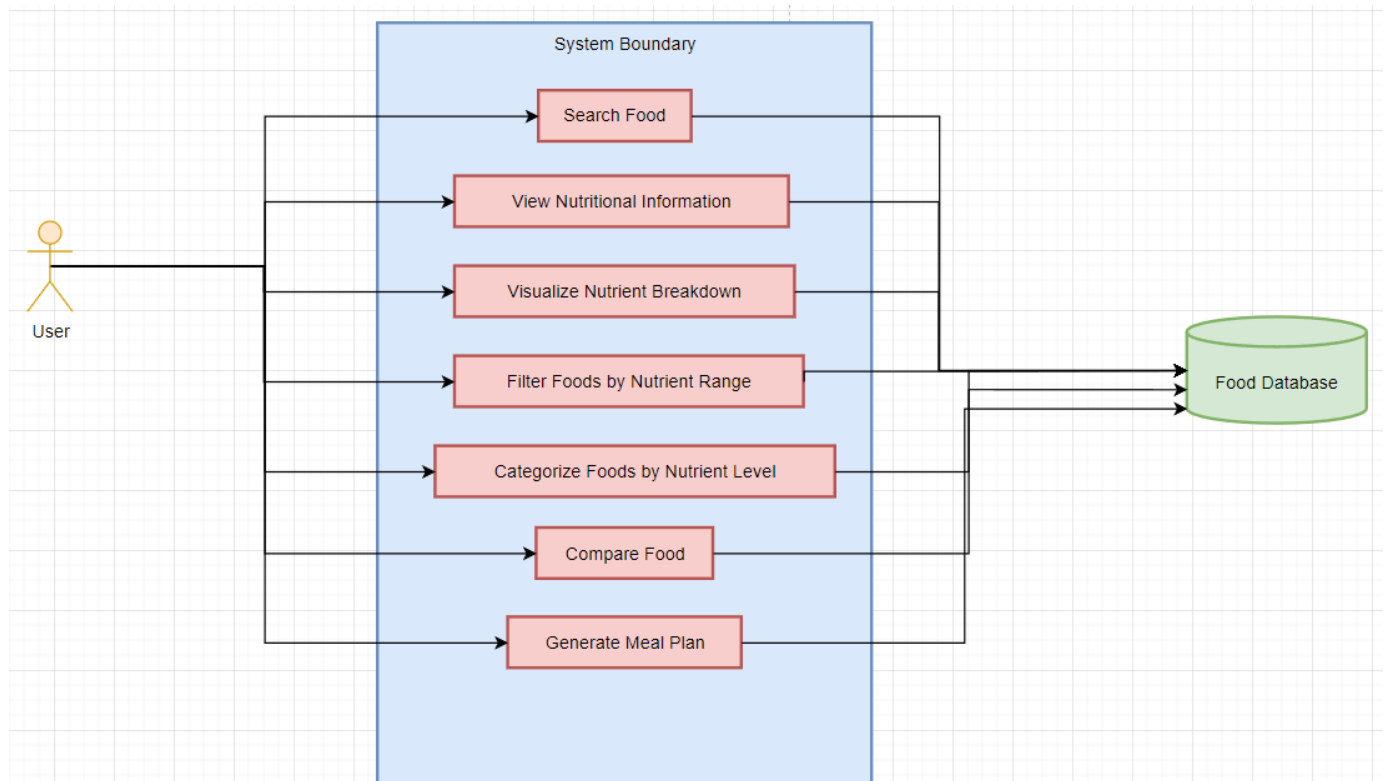
Relationships

- The lines connecting the **User** to various use cases indicate which functionalities the user can directly interact with.
- The arrows between use cases (e.g., from "Search Food" to "View Food Details") represent "include" relationships, showing that one use case is part of the flow of another.

System Boundary

- The rectangle enclosing the use cases represents the **boundary of the NutriPro system**, clearly delineating what functionalities are within the scope of our application.

This diagram provides a high-level view of the system's capabilities and how users will interact with NutriPro, serving as a roadmap for development and a quick reference for understanding the app's core functionalities.



2.4 Use Cases

Use Case ID	Use Case Name	Actors	Description	Flow of Events	*Alternate Flow	Postcondition
UC001	Access Main Menu	User	User accesses the main menu to navigate to different functions	1. User opens the application 2. System displays the main menu with options for Search, Compare, Meal Planner, and Nutrition Tip of the Day	N/A	Main menu is displayed
UC002	Search Food	User	User searches for a specific food item in	1. User selects "Search Foods" from the main menu 2. User enters	If no matching items are found, the system	Search results are displayed, or user is notified of no matches

Use Case ID	Use Case Name	Actors	Description	Flow of Events	*Alternate Flow	Postcondition
			the database	food name in the search bar 3. System processes the search query 4. System displays a list of matching food items 5. User can select a specific food item for detailed view	displays a "No results found" message	
UC003	View Food Details	User	User views detailed nutritional information for a selected food item	1. User selects a food item from the search results 2. System retrieves the nutritional data for the selected food 3. System displays comprehensive nutritional information including macronutrients, micronutrients, and a chart of macronutrient breakdown 4. User can scroll through the information	If the food item data is unavailable, system displays an error message	Detailed nutritional information is displayed for the selected food item
UC004	Apply Filter	User	User filters foods based	1. User selects filter options in	If no foods match the	Filtered list of foods is

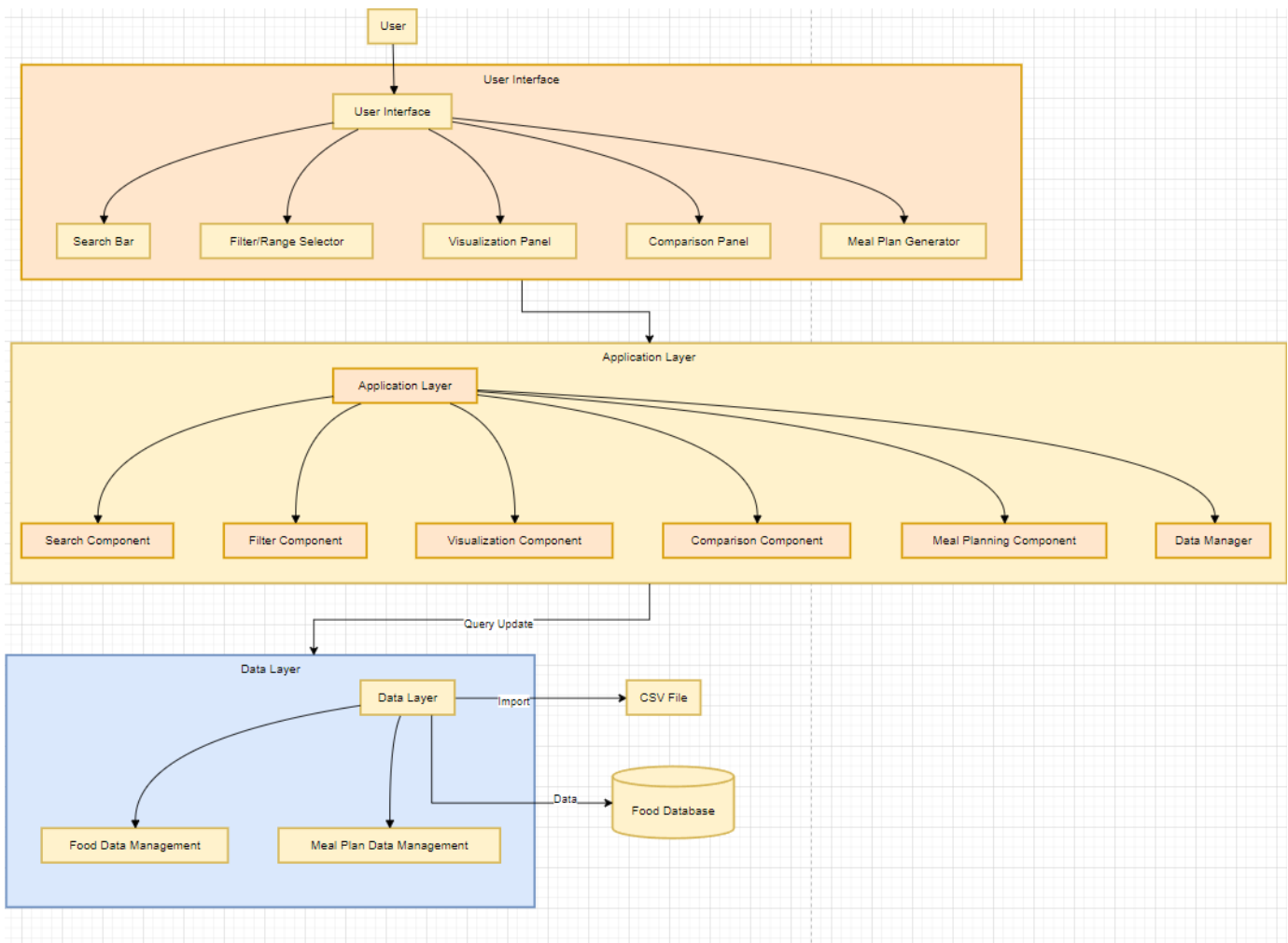
Use Case ID	Use Case Name	Actors	Description	Flow of Events	*Alternate Flow	Postcondition
	Settings		on nutrient content	<p>the search screen</p> <p>2. User selects nutrient(s) to filter by</p> <p>3. User inputs specific range values for the selected nutrients</p> <p>4. User applies the filter settings</p> <p>5. System queries the database based on the criteria</p> <p>6. System displays the filtered list of foods</p>	criteria, system displays "No foods found matching these criteria"	displayed
UC005	Compare Foods	User	User compares nutritional information of multiple food items	<p>1. User selects "Compare Foods" from the main menu</p> <p>2. User selects up to three food items for comparison</p> <p>3. System retrieves nutritional data for selected items</p> <p>4. System displays a side-by-side comparison</p>	If user tries to add more than three items, system prompts user to remove an item first	Comparison view is displayed with nutritional information of selected foods side-by-side

Use Case ID	Use Case Name	Actors	Description	Flow of Events	*Alternate Flow	Postcondition
				5. User can generate a comparison chart		
UC006	Manage Meal Plan	User	User creates, views, and edits a weekly meal plan	1. User selects "Meal Planner" from the main menu 2. System displays the weekly meal plan view 3. User can add foods to specific meals and days 4. System updates the nutritional summary for each day 5. User can save or load meal plans	If adding a food exceeds daily nutritional goals, system warns the user	Weekly meal plan is displayed and can be edited
UC007	Generate Meal Plan	User	User generates a meal plan based on nutritional goals	1. In the Meal Planner, user selects "Generate Meal Plan" 2. User inputs nutritional goals 3. System generates a meal plan meeting the criteria 4. System displays the	If no suitable meal plan can be generated, system suggests adjustments to the user's input	Generated meal plan is displayed in the weekly view

Use Case ID	Use Case Name	Actors	Description	Flow of Events	*Alternate Flow	Postcondition
				generated meal plan		
UC008	View Nutritional Summary	User	User views a summary of nutritional information for planned meals	1. In the Meal Planner, user views the daily nutritional summary 2. System calculates and displays total nutrients for each day 3. User can toggle between different days to view summaries	N/A	Daily nutritional summaries are displayed

3. Software Design and System Components

3.1 Software Design



Component Breakdown:

1. User Interface:

- Web Interface: The main user interaction point
- Search Bar: For food item searches
- Filter/Range Selector: For applying nutritional filters
- Visualization Panel: For displaying charts and graphs
- Comparison Panel: For side-by-side food comparisons
- Meal Plan Generator: Interface for creating meal plans

2. Application Layer:

- Search Component: Handles food search functionality
- Filter Component: Manages filtering and categorization
- Visualization Component: Generates charts and graphs
- Comparison Component: Processes food comparisons
- Meal Planning Component: Generates meal plans
- Data Manager: Central component for data handling and communication with the data layer

3. Data Layer:

- Food Database: Stores all food and nutritional information
- CSV File: The source of the nutritional data

3.2 System Components

3.2.1 Functions

1. `search_food(query: str) -> List[FoodItem]`
 - Description: Searches for food items based on user input
 - Input Parameters: `query` (string) - The search term entered by the user
 - Return Value: A list of `FoodItem` objects matching the search query
 - Side Effects: None
2. `generate_nutrition_chart(food_item: FoodItem) -> Chart`
 - Description: Creates a visual representation of the nutritional breakdown
 - Input Parameters: `food_item` (`FoodItem`) - The selected food item
 - Return Value: A `Chart` object containing the visualization
3. `filter_foods_by_nutrient(nutrient: str, min_value: float, max_value: float) -> List[FoodItem]`
 - Description: Filters foods based on a specific nutrient range
 - Input Parameters:
 - a. `nutrient` (string) - The nutrient to filter by
 - b. `min_value` (float) - The minimum value of the nutrient
 - c. `max_value` (float) - The maximum value of the nutrient
 - Return Value: A list of `FoodItem` objects that fall within the specified range
 - Side Effects: None
4. `categorize_foods_by_nutrient_level(nutrient: str, level: str) -> List[FoodItem]`
 - Description: Categorizes foods based on low, mid, or high levels of a specific nutrient
 - Input Parameters:
 - a. `nutrient` (string) - The nutrient to categorize by
 - b. `level` (string) - The level to filter (low, mid, or high)
 - Return Value: A list of `FoodItem` objects in the specified category
 - Side Effects: None
5. `compare_foods(food_items: List[FoodItem]) -> ComparisonResult`
 - Description: Compares multiple food items
 - Input Parameters: `food_items` (`List[FoodItem]`) - A list of `FoodItem` objects to compare
 - Return Value: A `ComparisonResult` object containing the comparison data
 - Side Effects: None
6. `generate_meal_plan(nutritional_goals: Dict[str, float]) -> MealPlan`
 - Description: Generates a weekly meal plan based on user-defined nutritional goals
 - Input Parameters: `nutritional_goals` (dictionary of nutrient names and target values)
 - Return Value: A `MealPlan` object containing the generated meal plan
 - Side Effects: None
7. `save_meal_plan(meal_plan: MealPlan, file_path: str) -> bool`
 - Description: Saves the current meal plan to a file
 - Input Parameters: `meal_plan` (`MealPlan` object), `file_path` (string)
 - Return Value: Boolean indicating success or failure of save operation
 - Side Effects: Creates or overwrites a file at the specified path
8. `load_meal_plan(file_path: str) -> MealPlan`
 - Description: Loads a meal plan from a file
 - Input Parameters: `file_path` (string)

- Return Value: A MealPlan object containing the loaded meal plan
- Side Effects: None

9. navigate_to_main_menu()

- Description: Returns the user to the main menu from any screen
- Input Parameters: None
- Return Value: None
- Side Effects: Changes the current view to the main menu

3.2.2 Data Structures / Data Source

This section describes the key data structures used in the NutriPro application and elaborates on how they interact with each other and with the system's functions.

1. FoodItem

- **Type:** Class
- **Usage:** Represents a single food item with all its nutritional information
- **Key Attributes:** name , calories , protein , carbohydrates , fats , vitamins , minerals
- **Key Methods:** get_nutrient_value(nutrient)

Interactions:

- Serves as the fundamental unit of food information in the **NutritionalDatabase**
- Used in search, filter, and comparison functions
- Forms the basis of meal components in the **MealPlan** structure

2. NutritionalDatabase

- **Type:** Class
- **Usage:** Manages the entire database of food items
- **Key Attributes:** food_items (a list of **FoodItem** objects)
- **Key Methods:** load_data(file_path) , search_food(query) , apply_filters(filters)

Interactions:

- Acts as the central repository for all **FoodItem** instances
- Interfaces with the CSV file data source to populate its food items list
- Provides data access for search, filter, and meal planning functions

3. Chart

- **Type:** Class
- **Usage:** Represents a visualization of nutritional data
- **Key Attributes:** chart_type , data
- **Key Methods:** render()

Interactions:

- Created by the `generate_nutrition_chart` function using **FoodItem** data
- Used in the user interface to display visual representations of nutritional information
- May be incorporated in **ComparisonResult** to visualize food comparisons

4. ComparisonResult

- **Type:** Class
- **Usage:** Represents the result of a food comparison
- **Key Attributes:** `food_items` (list of compared **FoodItems**), `comparison_data` (nutritional data for comparison)

Interactions:

- Produced by the `compare_foods` function using multiple **FoodItem** instances
- Used in the user interface for side-by-side food comparisons
- May interact with the **Chart** class for visual comparison generation

5. MealPlan

- **Type:** Class
- **Usage:** Represents a weekly meal plan with daily breakdowns
- **Key Attributes:** `weekly_plan` (structured meal data), `daily_summaries` (nutritional summaries)
- **Key Methods:** `add_food_to_meal` , `calculate_daily_summary` , `get_weekly_overview`

Interactions:

- Created and populated by the `generate_meal_plan` function using **FoodItems** from the **NutritionalDatabase**
- Interacts with **FoodItem** instances to calculate nutritional summaries
- Used by `save_meal_plan` and `load_meal_plan` functions for data persistence

Data Flow and Interactions

1. Data Loading

The **NutritionalDatabase** initializes by loading data from a CSV file, creating **FoodItem** instances for each entry.

2. Searching and Filtering

User queries activate `search_food` , `filter_foods_by_nutrient` , and `categorize_foods_by_nutrient_level` functions. These functions interact with the **NutritionalDatabase** to retrieve and process **FoodItem** data based on specified criteria.

3. Comparison

The `compare_foods` function takes multiple **FoodItem** instances and generates a **ComparisonResult** object. This result can then be used to create a **Chart** for visual representation.

4. Meal Planning

The `generate_meal_plan` function interacts with the **NutritionalDatabase** to select appropriate **FoodItems** based on nutritional goals. It creates and populates a **MealPlan** object, organizing **FoodItems** into meals and days, and calculates nutritional summaries.

5. Data Visualization

The `generate_nutrition_chart` function creates **Chart** objects using data from **FoodItems**, **ComparisonResults**, or **MealPlan** summaries.

6. Data Persistence

The `save_meal_plan` and `load_meal_plan` functions handle the serialization and deserialization of **MealPlan** objects for storage and retrieval.

This description of data structures and their interactions provides an overview of how data flows through the NutriPro system, from initial loading to user interactions and data persistence. It demonstrates how the various components work together to provide the application's functionality.

3.2.3 Detailed Design

Below is the detailed pseudo code for each functions

1. **search_food(query: str) -> List[FoodItem]**

- Initialize empty list result
- Convert query to lowercase
- For each food_item in database:
 - If query is in food_item.name (case-insensitive):
 - Add food_item to result
- Return result

2. **generate_nutrition_chart(food_item: FoodItem) -> Chart**

- Initialize new Chart object
- Set chart type to "Pie"
- Calculate total macronutrients (protein + carbs + fats)
- For each macronutrient in [protein, carbs, fats]:
 - Calculate percentage = (macronutrient / total) * 100
 - Add to chart data: {name: macronutrient, value: percentage}
- Return chart

3. **filter_foods_by_nutrient(nutrient: str, min_value: float, max_value: float) -> List[FoodItem]**

- Initialize empty list result
- For each food_item in database:
 - nutrient_value = food_item.get_nutrient_value(nutrient)
 - If min_value <= nutrient_value <= max_value:

- Add food_item to result
- Return result

4. **categorize_foods_by_nutrient_level(nutrient: str, level: str) -> List[FoodItem]**

- Initialize empty list result
- Define thresholds for low, mid, high levels
- For each food_item in database:
 - nutrient_value = food_item.get_nutrient_value(nutrient)
 - If (level is "low" and nutrient_value < low_threshold) or
 - (level is "mid" and low_threshold <= nutrient_value < high_threshold) or
 - (level is "high" and nutrient_value >= high_threshold):
 - Add food_item to result
- Return result

5. **compare_foods(food_items: List[FoodItem]) -> ComparisonResult**

- Initialize new ComparisonResult object
- For each nutrient in [calories, protein, carbs, fats, vitamins, minerals]:
 - Initialize empty list nutrient_values
 - For each food_item in food_items:
 - Add food_item.get_nutrient_value(nutrient) to nutrient_values
 - Add to comparison_data: {nutrient: nutrient_values}
- Set ComparisonResult.food_items to food_items
- Set ComparisonResult.comparison_data to comparison_data
- Return ComparisonResult

6. **generate_meal_plan(nutritional_goals: Dict[str, float]) -> MealPlan**

- Initialize new MealPlan object
- For each day in week:
 - For each meal in [breakfast, lunch, dinner, snacks]:
 - Initialize empty list meal_foods
 - While daily nutritional goals not met:
 - Select random food_item from database
 - If adding food_item meets nutritional goals:
 - Add food_item to meal_foods
 - Add meal_foods to MealPlan for current day and meal
 - Calculate and store daily nutritional summary
- Return MealPlan

7. **save_meal_plan(meal_plan: MealPlan, file_path: str) -> bool**

- Try:
 - Open file at file_path in write mode
 - Serialize meal_plan object to JSON
 - Write JSON to file
 - Close file
 - Return True
- Catch IOError:
 - Print "Error saving meal plan"

- Return False

8. `load_meal_plan(file_path: str) -> MealPlan`

- Try:
 - Open file at `file_path` in read mode
 - Read JSON from file
 - Deserialize JSON to MealPlan object
 - Close file
 - Return MealPlan object
- Catch IOError:
 - Print "Error loading meal plan"
 - Return None

9. `navigate_to_main_menu()`

- Clear current screen
- Display main menu options
- Wait for user input

4. User Interface Design

4.1 Structural Design

The Nutritional Food Database application is structured as follows:

Main Application Window

The main window consists of two primary sections:

- Header
- Content Area

Header

The header contains:

- **Logo:** Reinforces brand identity
- **Navigation Menu:** Provides quick access to main sections
 - Home
 - Compare
 - Meal Plan

Content Area

The content area is the primary interface for user interactions, comprising:

1. Search Bar:

- Prominently placed for easy access
- Allows users to search for food items

2. Results/Content Display:

- Dynamically shows different views based on user interaction:
 - Food List View: Displays search results
 - Detailed Food View: Shows comprehensive information about a selected food item
 - Comparison View: Allows side-by-side comparison of multiple food items
 - Meal Plan View: Displays generated meal plans

3. Filter Panel:

- Enables users to refine search results or set preferences
- Typically located on the right side for easy access

4. Visualization Area:

- Presents nutritional data in graphical formats
- Enhances understanding of nutritional information

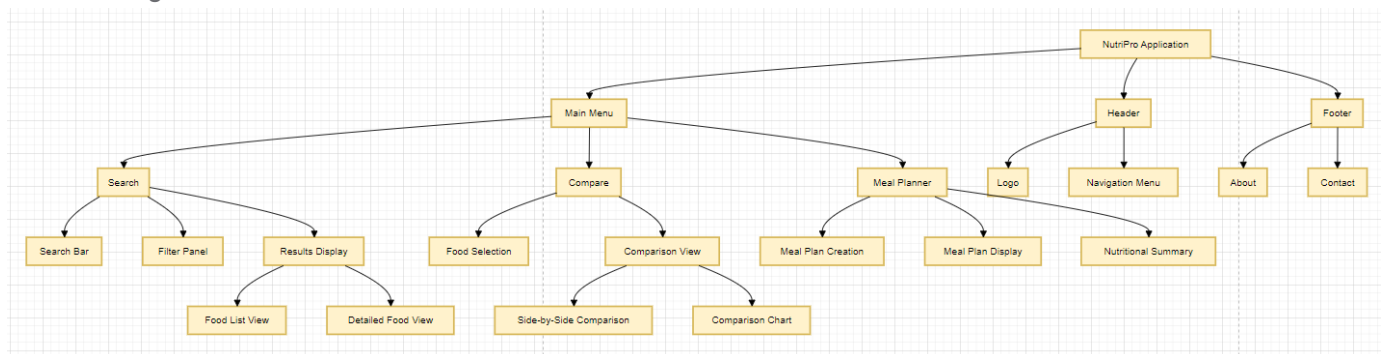
Design Choices

1. The hierarchical structure ensures clear organization of features and information.
2. Centrally placing the search bar emphasizes its importance in the user workflow.
3. The consistent right-sided filter panel provides a familiar location for refining searches.
4. Separating different content views (list, details, comparison, meal plan) allows for focused presentation of information.
5. Including a visualization area enables quick understanding of nutritional data through graphical representations.

This structure is designed to provide an intuitive and efficient user experience, allowing easy navigation between different functionalities while maintaining a clear and organized interface.

Structural Design Diagram

To visually represent the structure of our application, here's a structural design diagram showing the overall interface organization:



This hierarchy chart provides a visual representation of the NutriPro application's structure diagram, illustrating

how different components are organized and related to each other. It complements the detailed description provided above by offering a quick, at-a-glance view of the application's structure.

The diagram shows:

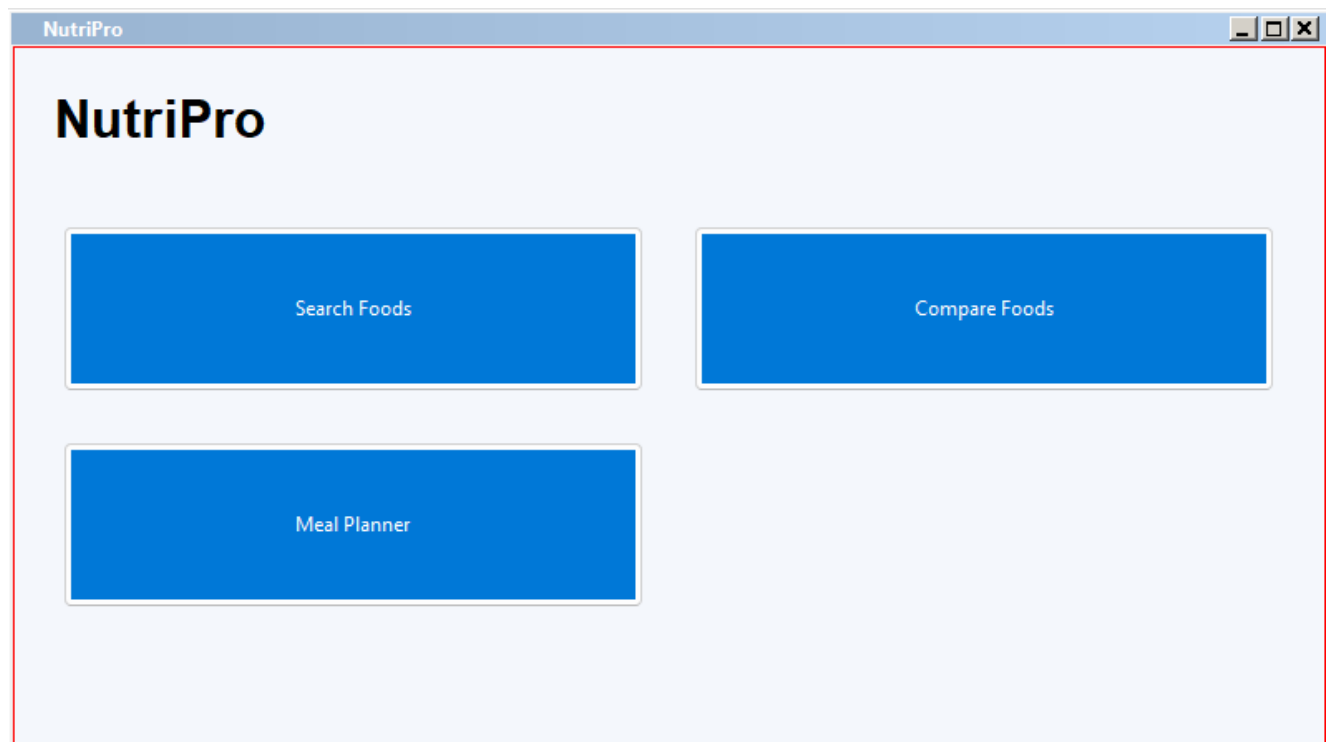
1. The main components of the application (Main Menu, Header, Footer)
2. The key sections accessible from the Main Menu (Search, Compare, Meal Planner)
3. The sub-components within each main section
4. The relationships between different parts of the interface

This visual representation, combined with the detailed textual description, provides a comprehensive overview of the NutriPro application's structural design, fulfilling the template's requirements for this section.

4.2 Visual Design

In designing the NutriPro application, our primary goal will be to create an interface that is not only visually appealing but also intuitive and efficient. We will aim to design a user experience that will make nutritional information accessible and actionable for users of all levels of nutritional knowledge. Let's explore the key screens of our application, examining the design choices that will shape the user's interaction with NutriPro.

1. Main screen

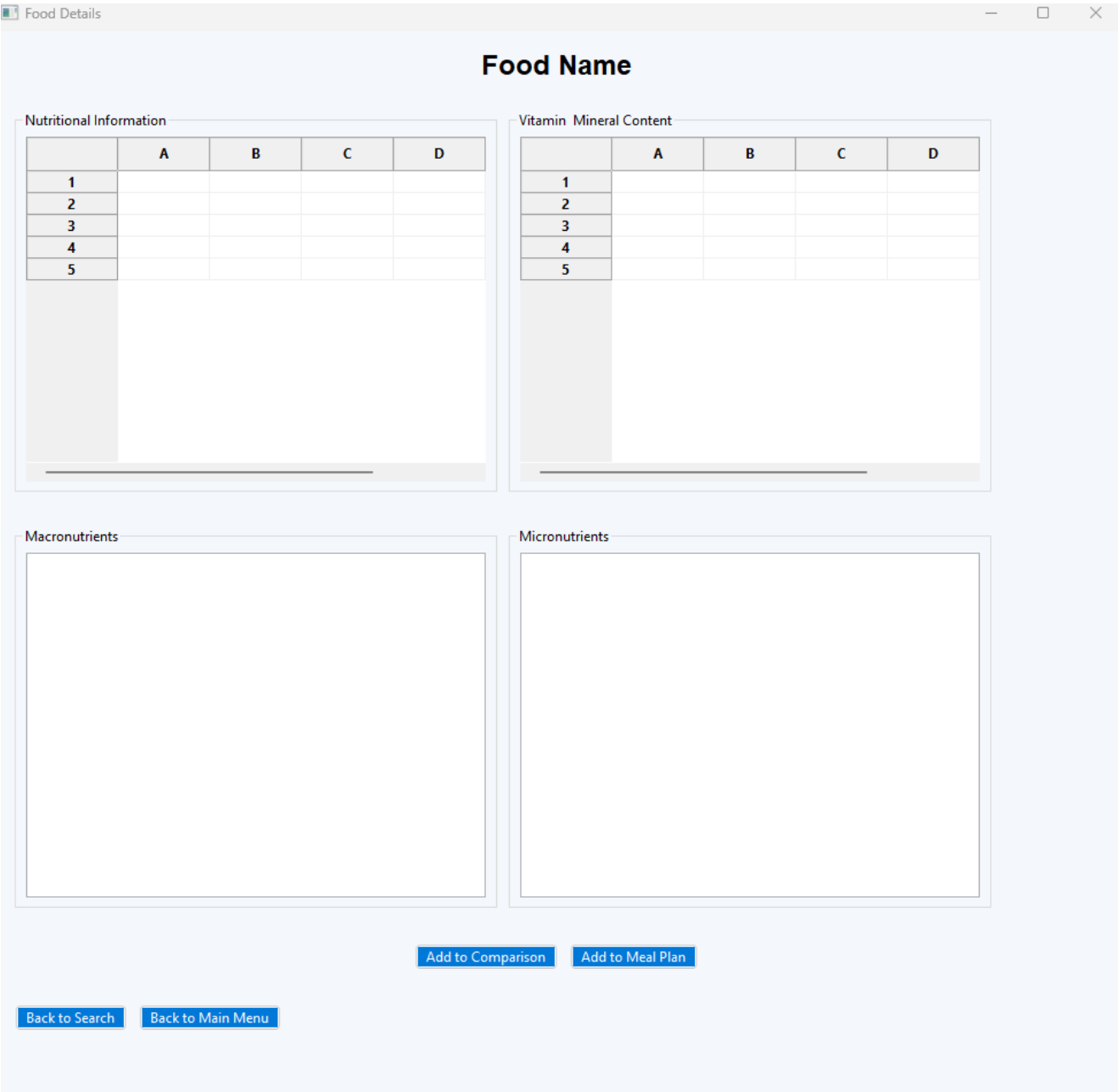


- Upon launching NutriPro, users will be greeted by a clean, uncluttered main menu. This screen will serve as the central hub of the application, offering three primary pathways: Search Foods, Compare Foods, and Meal Planner. Each option will be presented as a large, inviting button that will span the width of the screen.
- The simplicity of this design will be intentional. By presenting only three core functions, we will avoid overwhelming users with choices and instead guide them towards the app's primary features. The

generous size of the buttons will serve a dual purpose: it will make the options unmissable, even at a glance, and will provide large touch targets that will enhance usability across devices, particularly benefiting users with motor impairments.

- At the top of the screen, the NutriPro logo will stand prominently, reinforcing brand identity and assuring users they're in the right place. This consistent presence across all screens will act as an anchor, maintaining a sense of continuity throughout the user's journey.

2. Detailed food information and Nutrition breakdown view

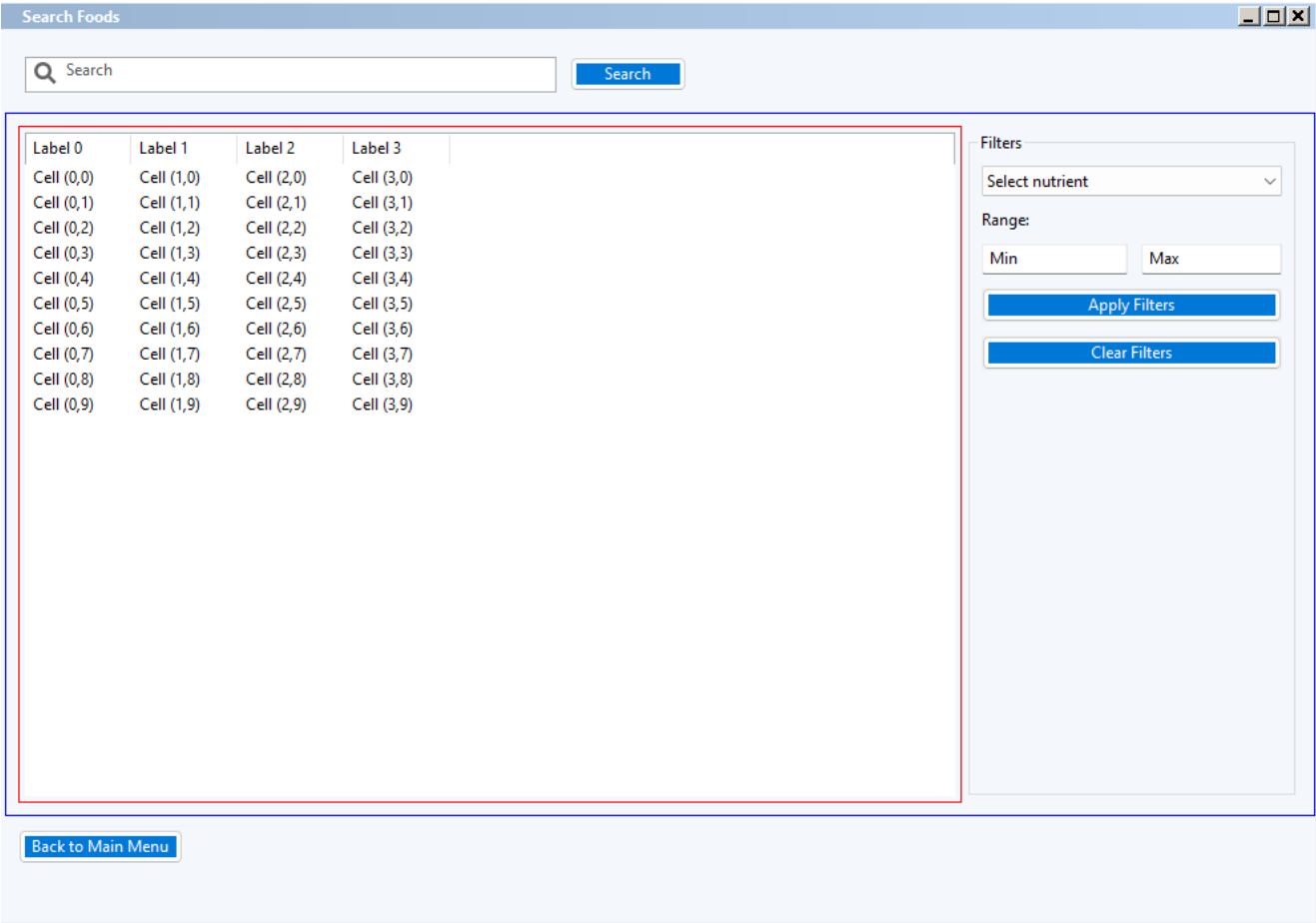


-
- The Food Details screen will serve as a comprehensive nutritional profile for individual food items. We will organize the wealth of information into four main sections: Nutritional Information, Vitamin/Mineral Content, Macronutrients, and Micronutrients. Each section will be presented in a tabular format, striking a balance between information density and readability.
- To prevent information overload, we will employ a tabbed layout. Users will be able to easily switch between different aspects of the nutritional profile, allowing them to focus on the specific information

they're interested in without being overwhelmed by the entirety of the data.

- At the bottom of the screen, we will place two action buttons: "Add to Comparison" and "Add to Meal Plan". These buttons will create seamless integration with other key features of the app, allowing users to immediately act on the information they've viewed. This integration of features will encourage exploration and utilization of the app's full capabilities.
- For navigation, we will include both "Back to Search" and "Back to Main Menu" buttons. This dual navigation option will provide flexibility, allowing users to either return to their search results or navigate back to the app's main menu, depending on their needs.

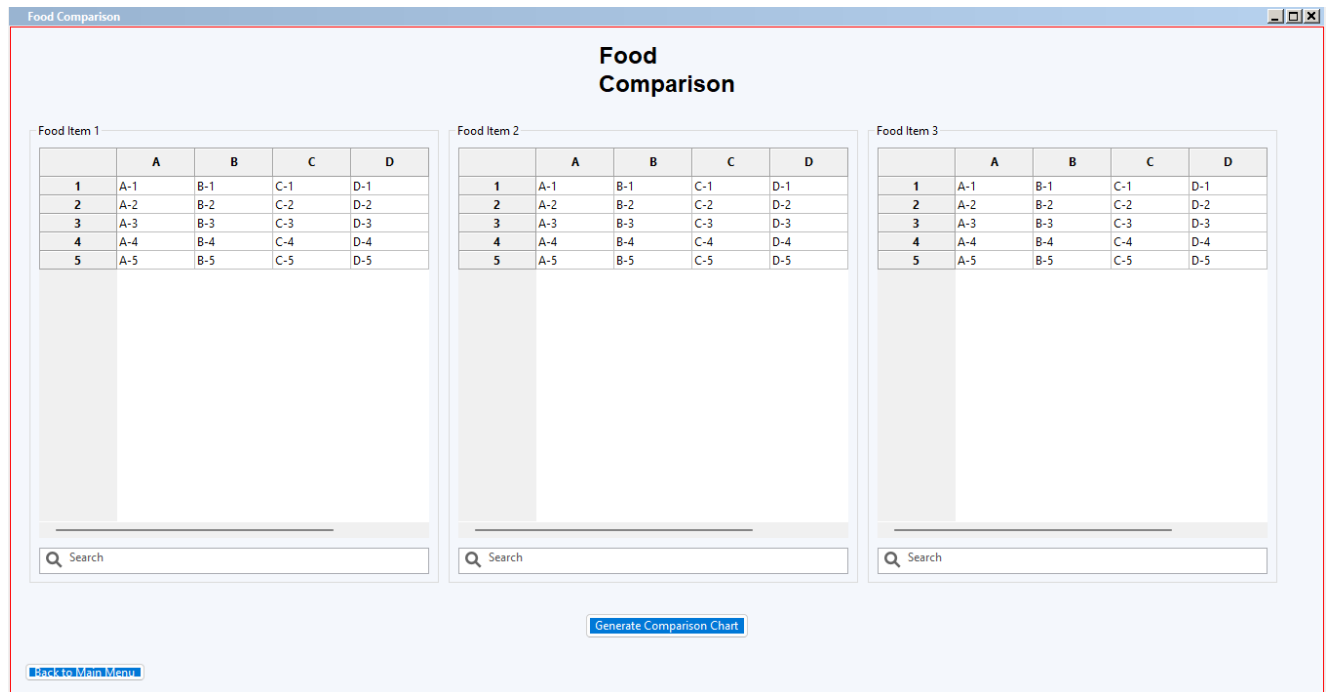
3. Search Screen



- The Search Foods screen will be where users begin their nutritional discovery. At the heart of this interface will be a prominently placed search bar, inviting users to start their query immediately. The dedicated "Search" button beside it will provide a clear call-to-action, catering to users who prefer tap interactions over keyboard submission.
- Below the search bar, results will be displayed in a multi-column grid. This layout will efficiently utilize screen real estate, allowing users to scan multiple food items quickly. Each cell in the grid will represent a food item, providing a snapshot of key nutritional information at a glance.
- To the right of the results, we will position a comprehensive filter panel. This placement will keep advanced search options readily accessible without cluttering the main view. Users will be able to select specific nutrients and set value ranges, enabling precise filtering of results. A "Clear Filters" button will sit at the bottom of this panel, offering an easy way to reset all filters with a single tap - a small but significant feature for users conducting multiple searches.

- At the bottom of the screen, a "Back to Main Menu" button will ensure users can easily navigate back to the app's central hub, maintaining a clear and consistent navigation structure throughout the application.

4. Food Comparison



- The Food Comparison screen will embody one of NutriPro's core functionalities: the ability to directly compare nutritional values across different foods. The interface will be divided into three equal columns, each representing a food item. This side-by-side layout will facilitate easy visual comparison, allowing users to quickly identify nutritional differences and similarities.
- Each column will be topped with its own search bar, a design choice that will enable users to swiftly find and compare specific food items without leaving the comparison view. Below these search bars, nutritional information will be presented in a tabular format. This structured layout will ensure that corresponding nutritional values align horizontally across all three columns, further simplifying the comparison process.
- At the bottom of the screen, we will place a "Generate Comparison Chart" button. This feature will transform the tabular data into a visual chart, offering an alternative representation that can reveal trends or differences that might not be immediately apparent in the numeric format.
- As with all screens, a "Back to Main Menu" button will be consistently placed at the bottom, ensuring users can easily return to the app's central navigation point.

5. Meal Plan Screen

Meal Planner

Weekly Overview

	A	B	C	D	E
1					
2					
3					
4					
5					

Daily Nutritional Summary

	A	B	C
1			
2			
3			
4			
5			

Daily Meal Plan

Breakfast

Add Item

Lunch

Add Item

Dinner

Add Item

Snacks

Add Item

Generate Meal Plan

Save Meal Plan

Load Meal Plan

Back to Main Menu

-
- The Meal Planner screen will transform nutritional information into practical, actionable meal plans. The interface will be divided into three main components: a weekly overview, a daily nutritional summary, and a detailed daily meal plan.
- The weekly overview, presented as a table at the top of the screen, will provide a bird's-eye view of the entire week's meal plan. This will allow users to see patterns and variety in their planned meals at a glance.
- Adjacent to the weekly overview, we will place a daily nutritional summary. This feature will aggregate the nutritional content of all planned meals for a selected day, enabling users to track their nutritional intake and ensure they're meeting their dietary goals.
- The lower portion of the screen will be dedicated to the detailed daily meal plan. We will divide this section into four distinct areas: Breakfast, Lunch, Dinner, and Snacks. Each meal category will have its own "Add Item" button, allowing users to easily populate their meal plan with foods from the database.
- At the bottom of the screen, we will include three key function buttons: "Generate Meal Plan", "Save Meal Plan", and "Load Meal Plan". These features will add significant value to the meal planning process. Users will be able to automatically generate a plan based on their nutritional goals, save their carefully crafted plans for future use, or load previously saved plans.

Throughout the design process, our focus will remain on creating an interface that is not just functional, but also engaging and empowering. By presenting complex nutritional information in an accessible, interactive format, NutriPro will aim to make the journey of nutritional discovery and meal planning both informative and enjoyable.