# Software Requirement Specification (SRS) Document

## 1. Brief Description of Project

The **Food Nutrition Info System** is a console-based mini-project developed in C that allows users to search, view, and manage nutrition information of various food items. It simplifies the process of obtaining nutritional facts such as calories, proteins, fats, carbohydrates, and vitamins for different foods in a structured and user-friendly manner.

## 2. Purpose / Goal

The system aims to automate the process of looking up nutritional details of food items, assisting users in maintaining a healthy diet plan and making informed dietary choices. It provides quick access to essential nutrition data, eliminating the need to consult multiple resources.

## 3. Usefulness / Benefit

**→ For Users:**

- Quickly access nutrition details of various foods
- Make informed dietary decisions
- Maintain a balanced diet by comparing nutritional values

**→ For Health Enthusiasts / Diet Planners:**

- Aid in calorie and nutrient tracking
- Assist in creating personalized diet plans
- Help manage specific dietary requirements (low-fat, high-protein, etc.)

## 4. Hardware / Software Involved

**→ Hardware Requirements:**

- Standard PC/Laptop
- Minimum 1GB RAM
- 200MB disk space

**→ Software Requirements:**

- Programming Language: C
- Compiler: GCC / Turbo C / Code::Blocks

- Operating System: Windows/Linux/MacOS

## 5. Detailed Feature List

**→ Main Module:**

- Load Food Nutrition Database
- Display List of Available Food Items
- Search Food Item by Name
- View Nutrition Details (Calories, Protein, Carbs, Fats, Vitamins)
- Add New Food Item
- Update Existing Food Item
- Delete Food Item

**→ Additional Features:**

- Sort Food Items by Calories or Macronutrients
- Search by Category (e.g., Fruits, Vegetables, Dairy, Grains)
- Export Nutrition Data to a Text File

## 6. Test / Demonstration Plan

**→ Unit Testing:**

- Validate database loading, searching, and viewing details individually
- Test adding, updating, and deleting food items

**→ Integration Testing:**

- Ensure smooth workflow between file/database handling and user operations

**→ System Testing:**

- Confirm end-to-end functionality with real food nutrition data samples

**→ User Acceptance Testing:**

- Run with sample users to evaluate ease of use, accuracy, and system responsiveness

## 7. Expected Interaction Interface and Sample Use Cases

**→ Interaction Interface:**

- Console-based UI with menu-driven options

- Simple text-based input and output

**Sample Use Cases:**

**→ Use Case 1: User Views Nutrition Information**
User selects 'View Food List' → Enters food name → Views detailed nutrition data

**→ Use Case 2: User Adds a New Food Item**
User selects 'Add New Item' → Enters food details (Name, Calories, Protein, etc.) → Item added to database

**→ Use Case 3: User Searches by Category**
User selects 'Search by Category' → Enters category (e.g., 'Fruits') → Views list of matching items

**Individual Member Contribution:**

Student 1 :Yashas K C -PES2UG24CS669- Planning and Structuring of the code with implementation

Student 2 :Yashwith Y Acharya -PES2UG24EC153-Data handling and file operations in the programme

Student 3 :Vikas N Naik-PES2UG24CS583-Linked list Management and searching

Student 4 :Yashas V-PES2UG24AM188-User interaction and input handling