**SOFTWARE REQUIREMENTS SPECIFICATION (SRS)**

Version 1.0

Date : 02/04/2025

# **DIET AND NUTRITION TRACKER**

# 

**Sof**

**1.Introduction**

**1.1 Purpose**

This document serves as a guide to outline the requirements for Diet and Nutrition Tracker software. Our goal is to create a clear understanding between everyone involved developers, project managers, and users

so that we can build a product that truly meets people's needs and supports their health journey.

**1.2 Scope**

The Diet and Nutrition Tracker is designed to empower users by providing them with the tools they need to log their food intake, track their nutritional values, and manage their dietary preferences. The **Diet and Nutrition Tracker** is a Python-based application that enables users to track their daily food intake, monitor calories and macronutrients, and set health goals.

**1.3 Intended Audience**

This document is for:

* **Developers:** Who will bring this vision to life.
* **Project Managers:** Who will oversee the project to ensure we stay on track.
* **QA Engineers:** Who will test every aspect of the software for quality.
* **End Users:** The individuals using the tracker to monitor their diet (Fitness enthusiasts, dietitians, health-conscious individuals)
* **Nutritionists/Dietitians:** Professionals who will guide users in making healthy choices.
* **System Administrators:** Those who will maintain the system's integrity

**1.4 References**

* IEEE 830-1998: A handy guideline for writing this document.

Industry Standards for Health & Nutrition Applications

**2. Overall Description**

**2.1 Product Perspective**

The Diet and Nutrition Tracker will be a standalone application, seamlessly integrating with existing health and fitness platforms. It will support cloud-based storage for user logs while also allowing offline functionality for basic tracking.

**2.2 Product Functions**

* Easy user sign-up and secure login.
* **Food logging** (add, update, delete meals with nutritional values) Smart data storage to keep everything organized.
* **Calorie & macronutrient tracking** (real-time calculations and tracking)
* **Goal setting** (customizable targets for weight loss, muscle gain, etc.)
* **Calorie & macronutrient tracking** (real-time calculations and tracking)
* **System monitoring and alerts** (reminders for meal logging, notifications for exceeding goals)

**2.3 User Characteristics**

We envision a wide range of users:

* Novice Users: Require an intuitive and easy-to-use interface for food logging.
* Advanced Users: Need detailed nutritional breakdowns, goal customization, and analytics.
* Nutritionists/Dietitians: Friends in the health space who need tools to support their clients effectively.

**2.4 Constraints**

We must adhere to health data privacy laws (like HIPAA and GDPR) to protect our users. Additionally, we need to ensure that the system is both secure and performs well under varying loads.

Compliance with **industry regulations** for health data privacy (GDPR, HIPAA if applicable).

Performance efficiency ensuring **low response times (<1s for core operations)**.

Compatibility with **multiple platforms (Web, Mobile, Desktop CLI)**.

**2.5 Assumptions and Dependencies**

* Users will need internet access for all online features.
* The application should work well on a variety of devices and web browsers, ensuring wide accessibility.

**3. External Interface Requirements**

**3.1 User Interfaces**

* A **responsive web application** that looks great on desktops and tablets.
* **A user-friendly** mobile app for iOS and Android, perfect for tracking on the go.
* **Command-line interface (CLI)** for administrative and advanced user tasks.

**3.2 Hardware Interfaces**

* Standard input/output devices (keyboard, mouse, touchscreen).
* Database servers for storage and retrieval of user data.

**3.3 Software Interfaces**

* **RESTful API** integration for retrieving food and nutritional data.
* **Database management system** for user data storage and historical tracking.

**3.4 Communication Interfaces**

* HTTPS to ensure all data transmitted remains secure.
* WebSockets for real-time interactions(updates and notifications)

**4. System Features**

**4.1  User Account Management**

* **Role-based access control** (User, Admin, Moderator).
* **Profile settings** (user preferences, dietary restrictions, goal tracking).

**4.2 Food Logging**

* **CRUD operations** on meal entries (Create, Read, Update, Delete).
* **Auto-fetch nutritional values** from an external API.
* **Offline functionality** for manual data entry and later synchronization.

**4.3 Reporting & Analytics**

* **Real-time data visualization** (daily calorie intake, weekly trends, goal tracking).
* **Generate and export reports**
* **Insights & recommendations** based on eating habits

**4.4 User Interaction Tracking**

* Identifies repeated spamming behavior.
* Notifies moderators of flagged content.

**5. Non-Functional Requirements**

**5.1 Performance Requirements**

* We aim for a response time of less than 2 seconds when users log food or retrieve their data.
* The system should handle at least 5,000 concurrent

**5.2 Security Requirements**

* **Data encryption** using **AES-256** for sensitive information.
* **Multi-factor authentication (MFA)** for enhanced security.
* **Secure API communication** to prevent unauthorized access.

**5.3 Software Quality Attributes**

* **Maintainability:** The software should be easy to update and improve over time.
* **Reliability:**  To keep the system running smoothly 99.5% of the time.
* **Usability: Intuitive UI/UX design for all user levels.**

**6. Other Requirements**

* Compliance with **GDPR, HIPAA** for health data privacy (if applicable).
* Regular software updates and **security patches** for data protection.
* Backup and **disaster recovery** strategies for user data.

**7. Appendix**

* **Glossary of Terms**
  + **CRUD:** Create, Read, Update, Delete - the basic operations we’ll be performing on user data.
  + **MFA:** Multi-Factor Authentication - an extra layer of security for user accounts.
  + **API:** Application Programming Interface - a bridge for different software to talk to each other.
  + GDPR (General Data Protection Regulation)
  + HIPAA (Health Insurance Portability and Accountability Act)
  + **Command-line interface (CLI)**