### **Software Requirements Specification**

#### **for**

## **FitLife**

**Version**: 1.0

#### **Prepared by Team Macro**

Mert Tömekçe  
 Halil Alan  
 Muhammet Taha Yıldırım  
 Ramazan Bağış  
 Orkun Yiğit Tekin

#### **For Client**

Çankaya University

**Date**: 25.04.2025

**Table of Contents**

[**1. Introduction 2**](#_fvnp1ws6zg47)

[1.1 Purpose 2](#_s0aez59zh27g)

[1.2 Scope 2](#_jusfdpn5xuck)

[1.4 References 3](#_f8lyjuy94i47)

[1.5 Overview 4](#_cwts6bb1n0rb)

[**2. GENERAL DESCRIPTION 4**](#_pa3tv3h7iscn)

[2.1 Product Perspective 4](#_64dvskgxzz2b)

[2.2 Product Functions 4](#_77gj24g42uk4)

[2.3 Users and Characteristics 5](#_fv83ly14mdxn)

[2.4 General Constraints 5](#_1d4oeh8a3h5d)

[2.5 Assumptions and Dependencies 5](#_wa8d9dgvp48y)

[**3. Specific Requirements 6**](#_p2hjo9s5ywnv)

[3.1 External Interface Requirements 6](#_n0msxzthbhkd)

[3.1.1 User Interfaces 6](#_cbcid02j0r61)

[3.1.2 Hardware Interfaces 11](#_vi9ab8m19mrb)

[3.1.3 Software Interfaces 11](#_ve46rm6lsea3)

[3.1.4 Communication Interfaces 12](#_gxcu6o43f2ut)

[3.2 Functional Requirements 12](#_g7duj2grz4zv)

[**3.2.1 Calorie Calculation 13**](#_ae5fxgcvl0m9)

[3.2.1.1 Description and Priority 13](#_xsrf2fygjv84)

[3.2.1.2 Stimulus/Response Sequences 14](#_hm92frscmy8e)

[3.2.1.3 Functional Requirements for Calorie Calculation 15](#_4do6vflcv1ji)

[**3.2.2 Food Search 16**](#_66wh2mcmxgyk)

[3.2.2.1 Description and Priority 16](#_ckbcnjr776of)

[3.2.2.2 Stimulus/Response Sequences 17](#_a7hvslwj7orv)

[3.2.2.3 Functional Requirements for Food Search 18](#_fwa26ag30oqh)

[**3.2.3 Goal Tracking 19**](#_ckvfq0h0j12i)

[3.2.3.1 Description and Priority 19](#_p6awzbdeimqr)

[3.2.3.2 Stimulus/Response Sequence 19](#_d99zl4s47mk8)

[3.2.3.3 Functional Requirements for Goal Tracking 20](#_xy03clzimid)

[**4. Other Non-Functional Requirements** 21](#_3yq4s3sl76m0)

[4.1 Performance Requirements 21](#_bbi3pzouuiio)

[4.2 Security Requirements 22](#_jvlgmdximwrp)

[4.3 Software Quality Attributes 23](#_ogijmsc56xlz)

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| FitLife SRS | 25.04.2025 | First Version created | 1.0 |
|  |  |  |  |
|  |  |  |  |

## **1. Introduction**

### **1.1 Purpose**

The purpose of this Software Requirements Specification (SRS) document is to comprehensively define the functional and non-functional requirements of the *FitLife* application. *FitLife* is a mobile fitness and wellness software platform that aims to support individuals in managing their nutritional intake and achieving their weight-related goals through generalized meal planning, dietary tracking, and progress monitoring.

### **1.2 Scope**

The *FitLife* application is designed to provide users with a platform to track their daily calorie intake, set weight goals (e.g., weight loss or gain), and view meal plans. Users can register an account, log in securely, search for various food items, and view detailed nutritional information including calories, carbohydrates, proteins, and fats.

The system will calculate personalized daily calorie allowances based on user attributes such as age, weight, height, gender, and activity level; by using Mifflin Algorithm. Based on this data and the user’s selected goal, the user can make adjustments on their diets. Users can monitor their progress over time through graphical weight tracking and summary reports.

The first version of the system will support:

* User registration and authentication
* Food database search and nutrient display
* Calorie calculation and goal setting
* Daily meal planning and tracking
* Visualization of progress reports

The initial release will focus on a single-user model with standalone functionality.

**1.3 Definitions, Acronyms, and Abbreviations**

| **Term** | **Definition** |
| --- | --- |
| **FitLife** | The name of the proposed fitness tracking application. |
| **API** | A set of protocols, tools, and definitions that allow different software systems to communicate and interact with each other. |
| **BMR** | Basal Metabolic Rate – the minimum number of calories required to keep the body functioning at rest. |
| **BMI** | Body Mass Index – an index derived from a person's weight and height, used to classify underweight, normal weight, overweight, and obesity. |
| **Calorie Allowance** | The number of daily calories a user should consume to meet their selected goal. |
| **Calorie Intake** | The total number of calories consumed through eating and drinking within a specific period, typically measured daily. |
| **Meal Plan** | A structured daily guide recommending what a user should eat, based on their dietary requirements and fitness goals. |
| **MB** | A megabyte (MB) is a unit of digital information storage that represents 1,024 kilobytes (KB) or 1,048,576 bytes. |
| **GB** | A gigabyte (GB) is a unit of digital information storage that represents 1,024 megabytes (MB) or 1,073,741,824 bytes. |
| **Nutrient Data** | Information about a food item’s composition including calories, carbohydrates, proteins, and fats. |
| **SRS** | Software Requirements Specification – a document that describes what the software will do and how it will be expected to perform. |
|  |  |
|  |  |

### **1.4 References**

* IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirements Specifications.
* Dietary Guidelines for Americans, 2020–2025. U.S. Department of Agriculture (USDA) and U.S. Department of Health and Human Services (HHS).
* Mifflin, M. D., St Jeor, S. T., Hill, L. A., Scott, B. J., Daugherty, S. A., & Koh, Y. O. (1990)/*The American Journal of Clinical Nutrition, 51*(2), 241–247.
* Open Food Facts API Documentation. Available at: <https://world.openfoodfacts.org/data>
* Özkurt, İ. (2025). *Turkish Food Recipes JSON Dataset*. GitHub Repository. Available at:<https://github.com/ilyasozkurt/turkish-food-recipes>

### **1.5 Overview**

This document is structured to provide a complete and systematic specification of the *FitLife* system:

* **Section 2: General Description** — provides a high-level view of the product’s context, primary functions, user characteristics, system constraints, and assumptions.
* **Section 3: Specific Requirements** — outlines detailed functional and interface requirements, including diagrams such as a use-case model, activity diagrams, and a data model to guide implementation.
* **Section 4: Non-Functional Requirements** — specifies performance benchmarks, security considerations, quality attributes, and other essential system-level requirements that ensure the reliability and usability of the product.

The SRS is intended to remain a living document throughout the development process and may be updated as needed based on feedback, testing results, or changes in project scope.

## **2. GENERAL DESCRIPTION**

### **2.1 Product Perspective**

FitLife is a mobile application designed to help users manage their daily nutrition by tracking calorie intake and offering generalized meal recommendations. It is a standalone product and will be accessible via smartphones. The app does not integrate with fitness trackers or health devices in its initial release. FitLife uses internal databases and APIs for nutritional data, and all user data will be securely stored and synchronized to cloud services for backup and multi-device access.

### **2.2 Product Functions**

-The product will include the following functions:

-Calculate daily calorie needs based on age, weight, height,gender and activity level.

-Provide a searchable food database with nutritional values (calories, protein, carbs, fats).

-Track daily food intake and weight progress over time.

-Allow users to customize personal fitness goals (weight loss, muscle gain, maintenance).

-Show meal suggestions.

-Enable dieticians design generalized meal plans for users.

-Display nutritional summaries and progress charts.

-Allow calorie input using common units like grams, ml, servings, or household measures.

### **2.3 Users and Characteristics**

FitLife will be used by a wide range of users including gym members, athletes, personal trainers, nutritionists, and individuals who are sedentary but seeking healthier lifestyles. While some users may have limited technical experience, the app will provide a clean and user-friendly interface, along with high-contrast themes to support for especially for elderly users.

### **2.4 General Constraints**

a. FitLife will be developed as a cross-platform mobile application (Android and iOS).  
b. All user data will be encrypted and secured.  
c. The food database must include both global and local cuisines, including Turkish dishes.  
d. Offline access will be available for core tracking features; food database will sync when online.  
e. App will be available only in English.  
f. The app must comply with relevant data protection regulations, including GDPR.

**Minimum Device Requirements:**

* Android 8.0 (Oreo) or iOS 13.0
* 2 GB RAM
* 100 MB of free storage space
* Internet connection (Wi-Fi or mobile data) for food database access and syncing (optional offline mode available for core features)

### **2.5 Assumptions and Dependencies**

Internet connection is required for syncing data and updating food database.

Users are expected to provide accurate input (weight, food portions) for effective tracking.

Cloud infrastructure will be available for data backup and restore features.

No third-party ad networks or in-app purchase systems will be integrated.

Nutritional data APIs (such as USDA FoodData Central or Nutritionix) will remain available and reliable.

App will be distributed through official app stores (Google Play Store and Apple App Store).

Users will need to agree to the terms of service and privacy policy before using the app

## **3. Specific Requirements**

### **3.1 External Interface Requirements**

#### **3.1.1 User Interfaces**

This section describes how users will interact with the system. The user interface will be menu-driven. Users can navigate through options like **Calorie Calculation**, **Food Search**, and **Goal Tracking** using touch controls. The mobile application will include the following user interfaces:

* **FitLife Main Menu**:  
   The central hub of the application, providing quick access to the main features such as Calorie Calculator, Goal Tracking, Food Search, and Notifications.

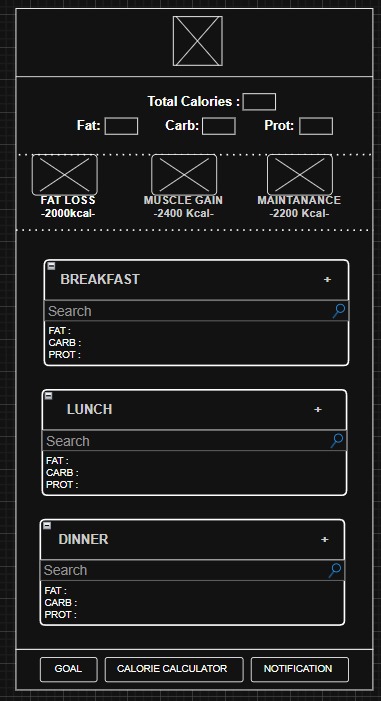


Image 1: FitLife Main Menu

* **Calorie Calculator Menu**:  
   Allows users to calculate the total calorie allowance based on the selected values.

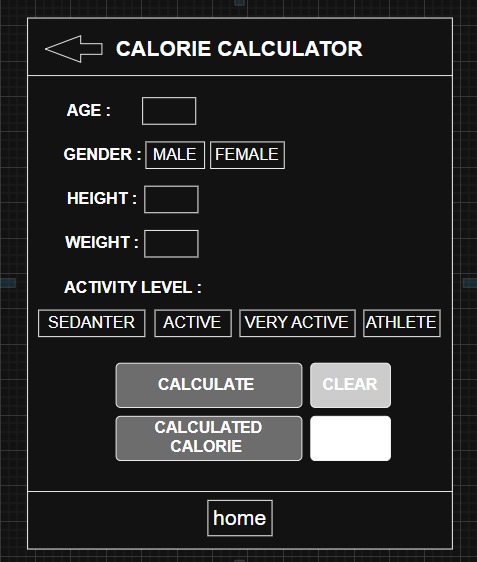


Image 2: FitLife Calorie Calculator Menu

* **Goals Menu**:  
   Displays the user’s progress toward their daily or weekly nutritional goals, helping users stay on track with their targets.

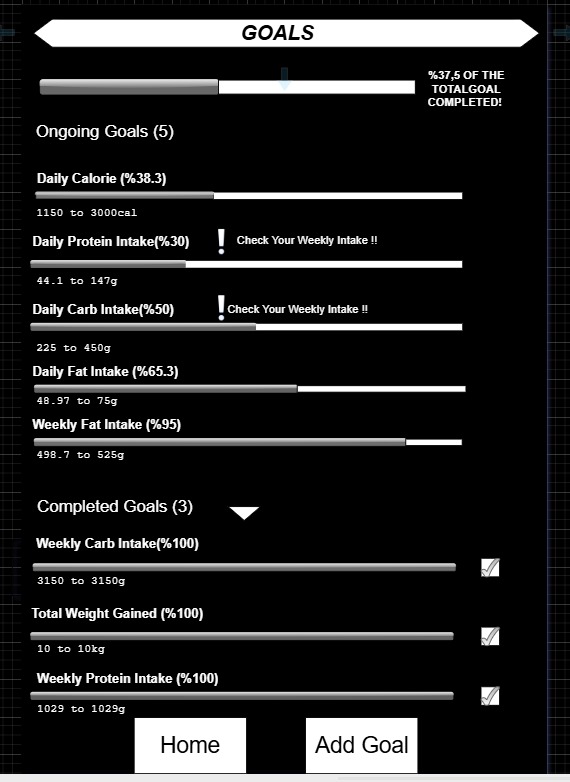


Image 3: FitLife Goals Menu

* **Add Goal Menu**:  
   Allows users to enter goal info such as start-end dates, goal name, goal type and calorie-weight values in order to create a new goal.

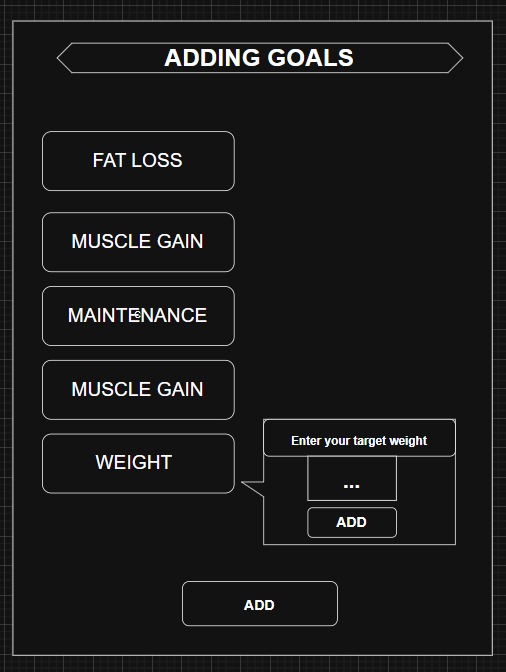


Image 4: FitLife Add Goal Menu

* **Notifications Menu**:  
   Lists reminders and updates related to meal logging, goal achievements, or other user alerts generated by the app.



Image 5: FitLife Notifications Menu

* **Search Menu**:  
   Enables users to search for foods in the database using keywords, with options to view detailed nutritional profiles.

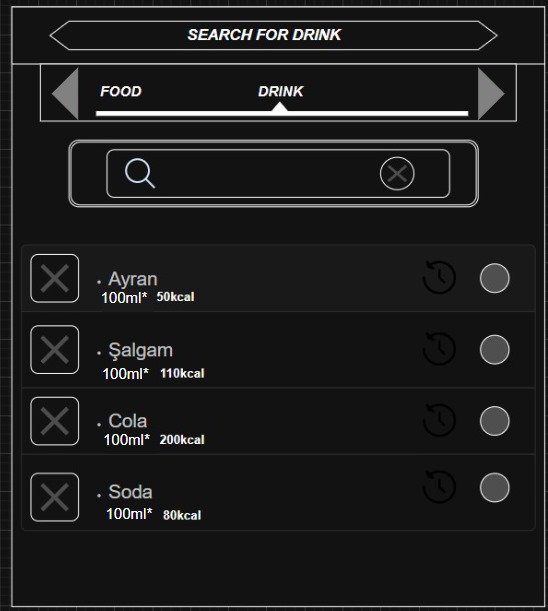
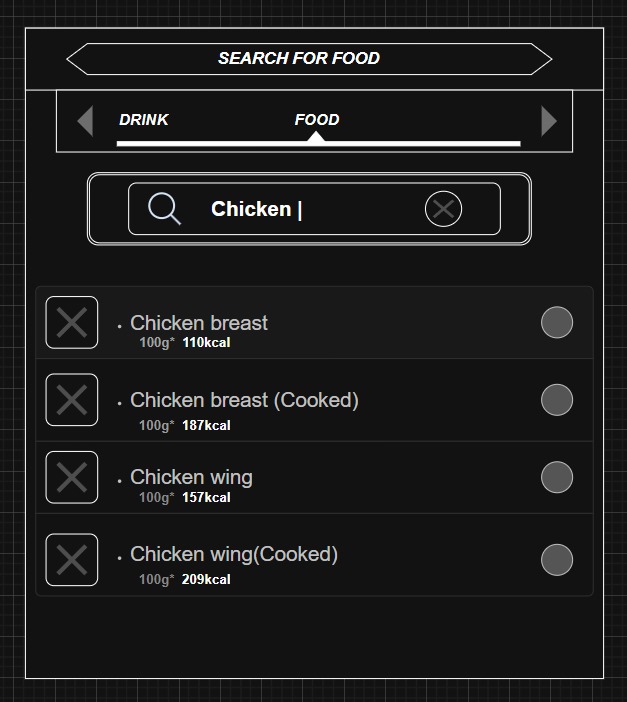


Image 6: FitLife Search Food Menu Image 7: FitLife Search Drink Menu

* **Food Nutrients Menu**:  
   Shows detailed nutritional information for a selected food item, including calories, macronutrients, and serving size options.

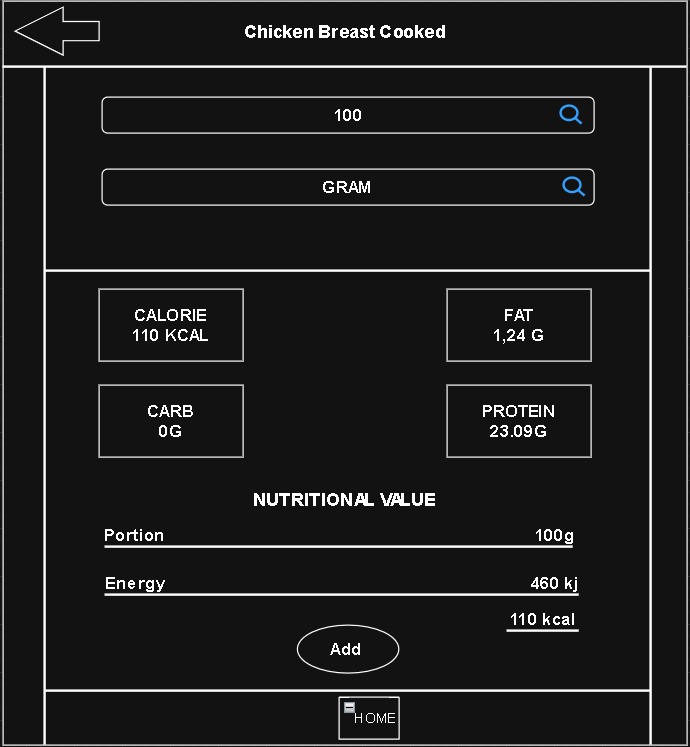


Image 8: FitLife Food Nutrients Menu

Here are some example app designs we have for reference:

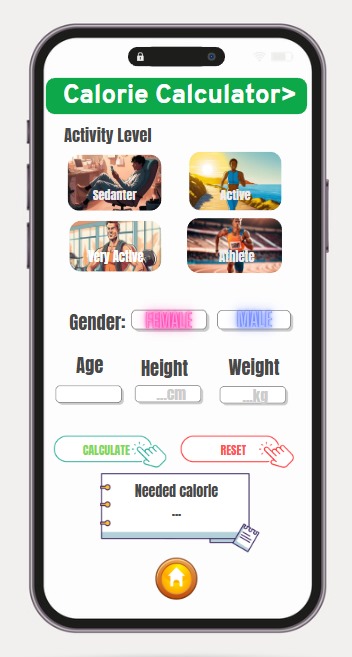
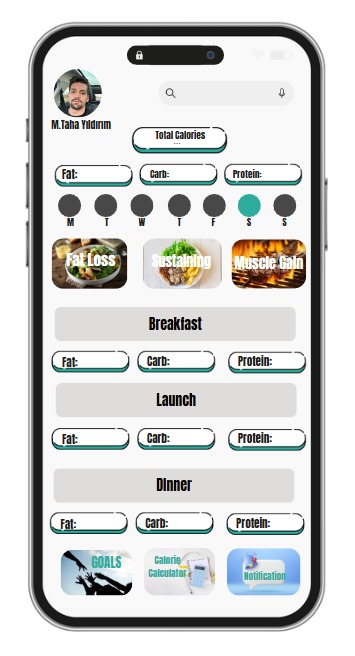


Image 9: FitLife Main Menu Mockup Image 10: FitLife Calorie Calculator Menu Mockup

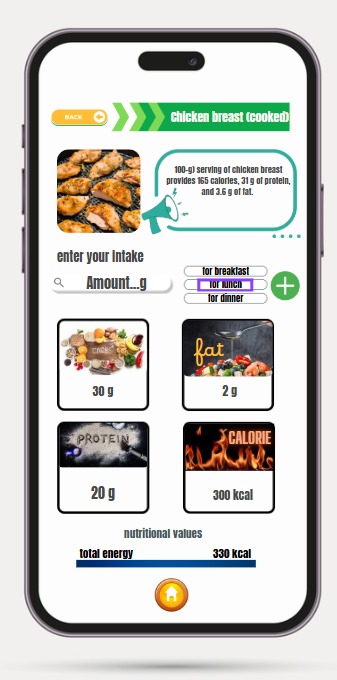
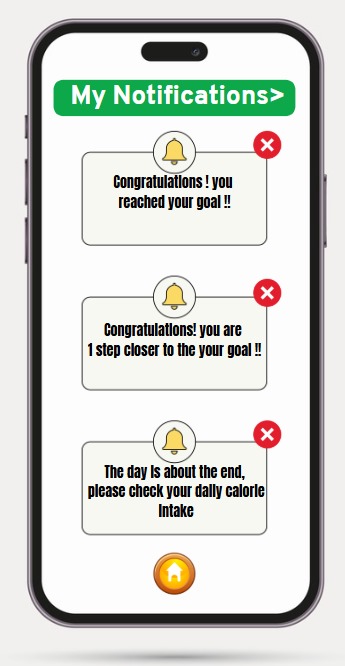


Image 11: FitLife Notifications Menu Mockup Image 12: FitLife Nutritional Value Menu Mockup

#### **3.1.2 Hardware Interfaces**

The application is developed for mobile devices and interacts with the following hardware components:

* **Touchscreen**: The primary input method. All interactions are designed for tap, swipe, and scroll gestures.
* **Device Storage**: Used for persisting user preferences, meal history, and offline cache.

#### **3.1.3 Software Interfaces**

The application interacts with the following software components:

* **Operating Systems Supported**:
  + Android (via React Native/Flutter)
  + iOS (via React Native/Flutter)
* **Local Storage**:
  + SQLite (for storing user progress and meal logs)
* **Food Database Integration**:
  + The app will connect to a custom backend food database to meet the requirements of including special Turkish Cuisine.
* **Analytics and Error Logging**:
  + Firebase Analytics will be integrated to track usage and crashes.

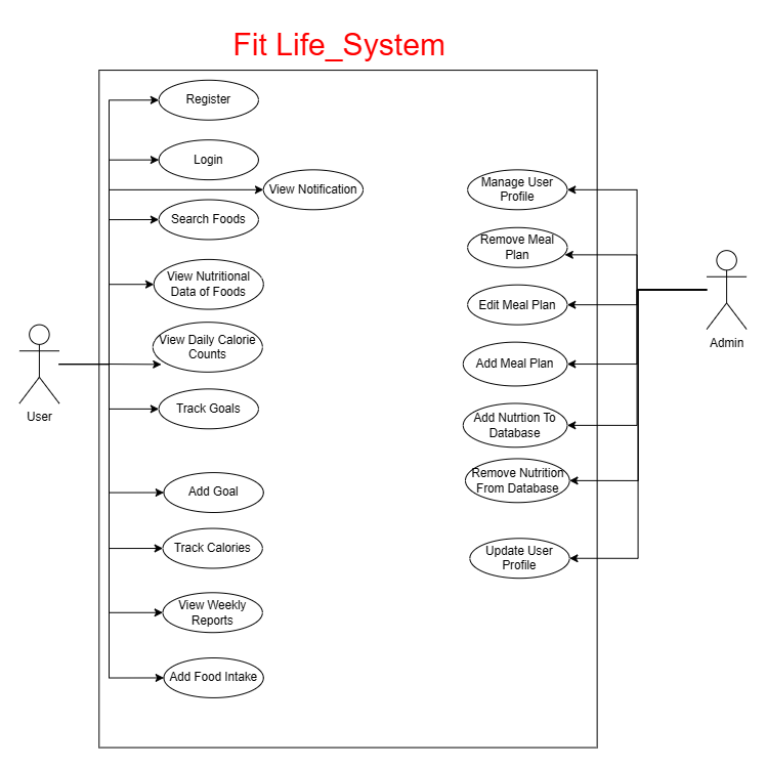
#### **3.1.4 Communication Interfaces**

The application will communicate with remote servers over secure HTTPs protocols.  
Key aspects:

* **API Requests**: Sending user queries (e.g., food search) to the backend.
* **Data Synchronization**: Fetching updates for food databases and syncing user progress.
* **Notification Delivery**: Receiving push notifications for reminders, if the feature is active.

All communication will use industry-standard encryption (TLS 1.2 or higher).

### **3.2 Functional Requirements**

The **FitLife** app is designed to assist users in managing their daily nutrition and achieving fitness goals. It provides features to calculate daily calorie intake, search for various foods and their nutritional content, and track progress toward fitness-related objectives. These functionalities aim to support users in making informed dietary choices, optimizing their fitness routines, and maintaining a balanced lifestyle.

#### 

#### 

#### 

#### 

#### 

Image 13: FitLife Use-Case Diagram

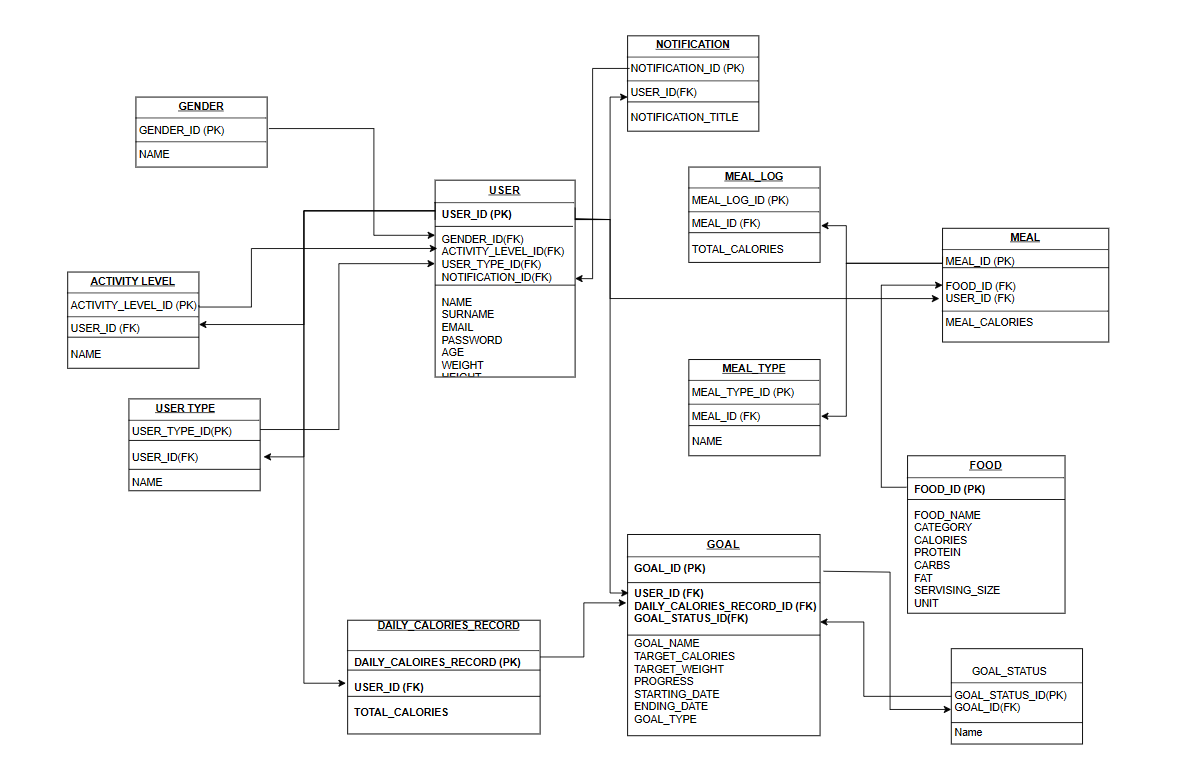


Image 14: FitLife Data Model

## **3.2.1 Calorie Calculation**

### **3.2.1.1 Description and Priority**

The **Calorie Calculation** feature enables users to calculate the total calories, proteins, carbohydrates, and fats of their daily food intake by selecting food items from the database.  
 This is a **high priority** feature because it forms the core functionality of FitLife.

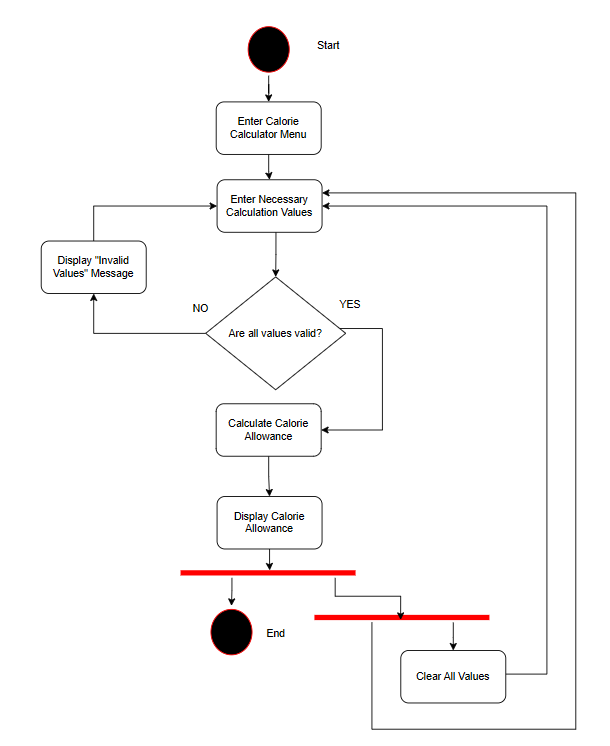


Image 15: FitLife Calorie Calculator Activity Diagram

### **3.2.1.2 Stimulus/Response Sequences**

The expected user interactions and system responses are as follows:

* **User**: Opens the **Calorie Calculator Menu** from the FitLife Main Menu.
* **User**: Searches for a food item or browses categories (e.g., fruits, vegetables, meats).
* **System**: Displays search results with food name, portion size, and nutritional values (calories, protein, carbs, fat).
* **User**: Selects a food item and enters the consumed quantity.
* **System**: Calculates the calorie and nutrient values based on the entered quantity.
* **User**: Adds the food item to their respective list.
* **System**: Updates and displays the running total of calories and nutrients for the day.
* **User**: Can remove or edit added food entries if necessary.
* **System**: Recalculates and updates the totals immediately after changes.

### **3.2.1.3 Functional Requirements for Calorie Calculation**

The following functional requirements are associated with the Calorie Calculation feature:

| **Code** | **Requirement** |
| --- | --- |
| REQ-CC-1 | System shall allow users to search or browse for food items. |
| REQ-CC-2 | System shall display food name, portion size, calories, protein, carbohydrate, and fat content. |
| REQ-CC-3 | System shall allow users to input the quantity consumed for each food item. |
| REQ-CC-4 | System shall calculate total calories and nutrients based on the quantity entered. |
| REQ-CC-5 | System shall allow users to add multiple food items to their respective intake list. |
| REQ-CC-6 | System shall update the daily total whenever a food item is added, edited, or deleted. |
|  |  |
| REQ-CC-7 | System shall display the current day's total calories, protein, carbohydrates, and fats at all times while in the Main Menu. |
|  |  |

## **3.2.2 Food Search**

### **3.2.2.1 Description and Priority**

The **Food Search** feature allows users to quickly find food items by name or category in the database.  
 This is a **high priority** feature because it supports the Calorie Calculation and overall usability of FitLife.

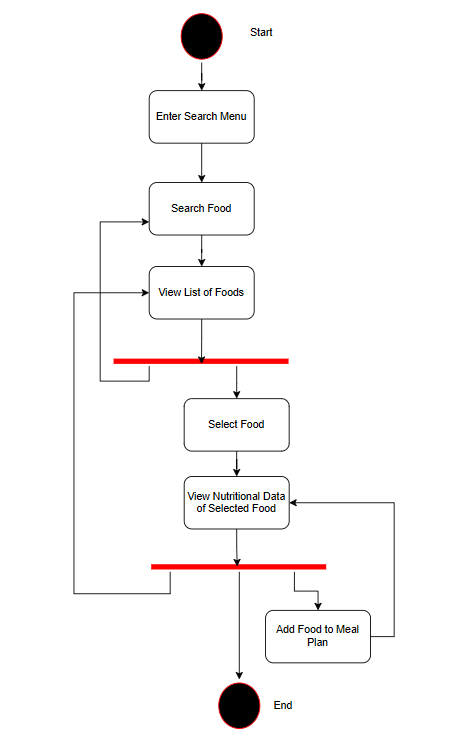


Image 12: FitLife Search Menu Activity Diagram

### **3.2.2.2 Stimulus/Response Sequences**

The expected user actions and system responses are listed below:

* **User**: Opens the **Search Menu** from the FitLife Main Menu.
* **User**: Types a food name or category keyword.
* **System**: Suggests auto-complete options based on entered characters (if available).
* **User**: Selects a suggested option or continues typing a full query.
* **System**: Displays a list of matching food items with basic nutritional info (calories per portion).
* **User**: Taps on a food item to view its detailed nutritional breakdown (calories, protein, carbs, fats) in the **Food Nutrients Menu**.

### **3.2.2.3 Functional Requirements for Food Search**

The following functional requirements are associated with the Food Search feature:

| **Code** | **Requirement** |
| --- | --- |
| REQ-FS-1 | System shall allow users to search food items by name or category, |
| REQ-FS-2 | System shall provide auto-complete suggestions during text input. |
| REQ-FS-3 | System shall display a list of matching food items with their basic nutritional info. |
| REQ-FS-4 | System shall allow users to select a food item to view detailed nutrition facts. |
| REQ-FS-5 | System shall navigate to the **Food Nutrients Menu** upon selection of a food item. |
| REQ-FS-6 | System shall handle cases where no search results are found by displaying a message saying “There is no such food in the database”. |
|  |  |

## **3.2.3 Goal Tracking**

### **3.2.3.1 Description and Priority**

The **Goal Tracking** feature enables users to set personalized daily or weekly calorie and macronutrient goals, and monitor their progress over time.  
This is a **high priority** feature because it directly supports the users' fitness, and diet management objectives.

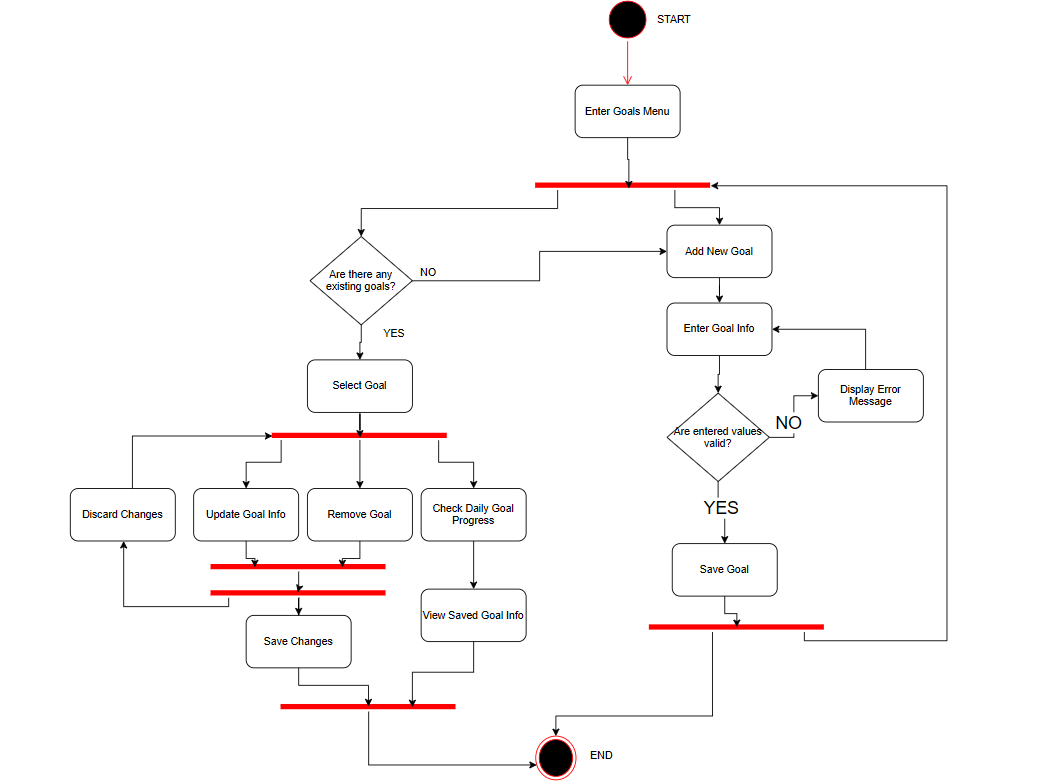


Image 13: FitLife Goal Tracking Activity Diagram

### **3.2.3.2 Stimulus/Response Sequence**

The expected user actions and system responses are listed below:

* **User**: Opens the **Track Goals Menu** from the FitLife Main Menu.
* **User**: Enters or updates daily/weekly goals for calories, protein, carbs, and fats.
* **System**: Saves the goals and initializes daily/weekly tracking.
* **System**: Displays a progress dashboard showing the user's current intake compared to their goals.
* **User**: Records daily intake through the Track Goals Menu.
* **System**: Automatically updates progress based on the newly added foods.
* **System**: Notifies users when goals are reached or when nearing thresholds.

### **3.2.3.3 Functional Requirements for Goal Tracking**

The following functional requirements are associated with the Goal Tracking feature:

| **Code** | **Requirement** |
| --- | --- |
| REQ-GT-1 | System shall allow users to set daily calorie and macronutrient (protein, carbohydrate, fat) goals. |
| REQ-GT-2 | System shall allow users to optionally set weekly goals. |
| REQ-GT-3 | System shall save user goals and retrieve them automatically upon app restart. |
| REQ-GT-4 | System shall display current progress toward goals using a visual progress dashboard. |
| REQ-GT-5 | System shall update the user's progress automatically when a food item is logged. |
| REQ-GT-6 | System shall send a notification or display visual alert when a user meets or exceeds their goals. |
| REQ-GT-7 | System shall allow users to delete goals at any time. |
|  |  |

### **4. Other Non-Functional Requirements**

### **4.1 Performance Requirements**

1. Response Time for User Queries  
   * The system will respond to user queries (e.g., food searches, calorie calculations) within 5 seconds.
   * The response time for 95% of user queries must not exceed 5 seconds, and for 99% of queries, it must not exceed 7 seconds, measured using system performance logs. This response time includes the time taken to process the request and return results to the user.
2. App Size and Optimization  
   * The app will be at most 100 MB and optimized to run smoothly even on low-end mobile devices with at least 2 GB of RAM.
   * The app will be optimized to maintain at least 30 FPS during common interactions (e.g., scrolling, data input, switching screens) on devices with the minimum specified hardware.
3. Efficient Food Search  
   * Despite having thousands of food items in the database, efficient filtering and indexing will ensure fast search results.
   * The system must be capable of returning food search results in less than 5 seconds even when the food database contains over 50,000 items. The search algorithm must support keyword-based searches, category filtering, and nutritional value-based filtering without noticeable delays.
4. System Uptime  
   * The system will maintain at least 95% uptime during daily operations such as food logging and progress tracking.
   * Downtime must not exceed 1.8 days per month. The system should be resilient to common failures such as network interruptions and temporary server outages, and any downtime should be minimized.

### **4.2 Security Requirements**

1. User Data Encryption  
   * User data (e.g., weight history, goals, login credentials) will be stored using encryption.
   * All sensitive user data, such as weight history, calorie goals, and login credentials, shall be securely stored using encryption methods that adhere to industry standards, such as AES-256 for data at rest and RSA-2048 for key management.
2. Login Security and Password Policies  
   * Login will be protected with strong password policies (e.g., minimum length, special character requirements).
   * Passwords must include a minimum of 8 characters, at least one uppercase letter, one number, and one special character. The system shall enforce the password policy during user registration and password change. Any password not meeting the specified requirements shall be rejected.
3. User Control Over Data and Account Deletion  
   * Users will have full control to delete their data or accounts permanently with confirmation prompts.
   * The system shall provide an intuitive and accessible process for data deletion within the settings menu. The system must permanently delete user data from both local storage and cloud servers upon confirmation. A deletion log will be maintained for audit purposes.
4. GDPR Compliance  
   * The app will comply with GDPR principles: clear user consent, data minimization, and transparent data handling.
   * The system must include clear and accessible consent forms that users must sign during registration and whenever there are significant changes in data usage policies. The system must allow users to request access to their personal data and, upon request, allow them to delete their data or request a data export.
5. Cloud Synchronization Security  
   Cloud synchronization will use encrypted connections (HTTPS) for data transmission.  
   * All data transmitted to and from the cloud shall use encrypted communication channels. The system shall use HTTPS (SSL/TLS) for all data transmission to ensure secure cloud synchronization. The app shall be periodically tested for vulnerabilities to ensure that secure connections are maintained.

### **4.3 Software Quality Attributes**

The FitLife app is built with a focus on **usability and accessibility**, offering a clean and intuitive interface suitable for all users, including the elderly. High-contrast themes, readable fonts, and simple navigation ensure ease of use and inclusivity.

For **reliability and correctness**, the system ensures accurate calorie calculations using validated formulas like Mifflin-St Jeor. All user data is stored and retrieved reliably, and the app prevents conflicting or unsafe goal settings.

The app is optimized for **efficiency and scalability**, enabling fast searches within a large food database and smooth operation even on low-end devices. Its architecture allows for performance consistency as user and data volume grows.

**Maintainability and portability** are ensured through a modular design that supports easy updates and feature expansion. The app runs seamlessly on both Android and iOS devices, adapting to various screen sizes.

In terms of **security**, all user data is encrypted, and strong password policies are applied. The system complies with GDPR standards, offering secure cloud backups and full user control over account and data privacy.