Software Requirements Specification

for

TrackMate

Version 1.0

Prepared by: Earth's Mightiest Dozen (Minus 2)

Abubakar Siddique 240043 abubakars24@iitk.ac.in
Aditi 240048 aditij@iitk.ac.in
Aman Gupta 240100 amanu24@iitk.ac.in
Ankit Kumar Jha 240137 ankitkj24@iitk.ac.in
Ayush Mangal 240242 ayushn24@iitk.ac.in
Chandradip Karmakar 240296 chandradip24@iitk.ac.in
Krishna Agrawal 240563 krishnaa24@iitk.ac.in
Nideesh H 240690 nideesh24@iitk.ac.in
Rushali Vijayakumar Myageri 240894 rushali24@iitk.ac.in
Sahil Agarwal 240902 sahilag24@iitk.ac.in

Course: CS253
Mentor TA: George TL
Submission Date: ***January 24, 2026***

Contents

1. Introduction

1.1 Purpose

The purpose of this document is to describe the functional and non-functional requirements for the AI-Integrated Fitness & Workout Tracker application named **TrackMate**. This document is intended for the project team, stakeholders, and developers to ensure a shared understanding of the system's scope and objectives.

1.2 Product Scope

The product aims to assist gym-goers and fitness enthusiasts by providing a structured approach to their workout regimes. Currently, individuals often lack data and historical records of their workouts and resulting metrics (e.g., calorie loss/gain).

This product provides a one-stop solution for fitness needs, featuring:

- **Data Tracking:** Recording workout history and results.
- **Motivation:** Social comparison with friends and leaderboard features.
- **Professional Guidance:** Advice and monitoring from certified trainers.
- **Dietary Management:** Tracking daily calorie intake and burn to evaluate diet effectiveness.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS:** Software Requirements Specification
- **LLM:** Large Language Model (AI used for generating summaries)
- **API:** Application Programming Interface
- **UI/UX:** User Interface / User Experience
- **IEEE:** Institute of Electrical and Electronics Engineering

1.4 Document Conventions

- This document follows the IEEE standards for Software Requirements Specification (SRS).
- The document is written with the Arial font of size 11 with single spacing and 1-inch margins.
- Bold font is used to highlight important keywords, actors, and specific requirements, while *italics* are used for comments and emphasis.
- The document is prepared using Google Docs to ensure consistency in formatting, ease of maintenance, and scalability for future updates.
- Section titles are hierarchical and numbered, with clear and concise descriptions under each heading.

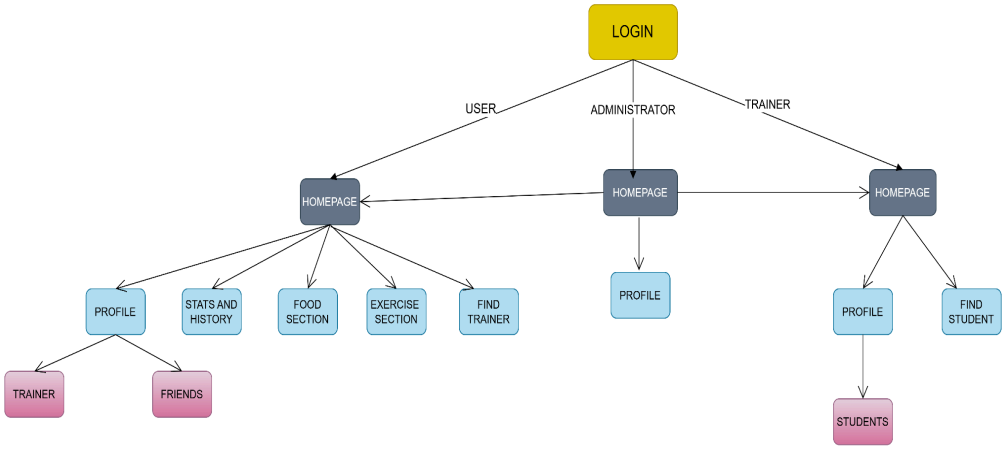
1.5 References and Acknowledgments

- We extend our gratitude to our TA, George TL, and our course instructor, Prof. Indranil Saha, for their invaluable guidance throughout the creation of this document and for providing us with a template for the Software Requirements Specification.
- References include initial project discussions and relevant course materials.

2. Overall Description

2.1 Product Perspective

This is a new, self-contained mobile application. At its core, the system utilizes mobile sensors to detect physical activity and integrates with cloud services for data processing. Uniquely, it generates weekly summaries using a pre-trained **Large Language Model (LLM)** to provide actionable insights for future workouts.



2.2 Product Functions

The major functions of the system are outlined below:

1. **User Roles and Authentication:** Management of user access and privileges.
2. **Nutrition and Calorie Tracking:** specific focus on Indian food items and custom entries.
3. **Fitness and Workout Management:** Logging exercises, duration, sets, and reps.
4. **Step Counter and Gamification:** Sensor-based tracking with reward systems.
5. **AI and LLM Integration:** Weekly progress analysis and smart suggestions.
6. **Social and Community Features:** Forums and social interaction.
7. **Trainer–Student Interaction:** Monitoring tools for trainers.
8. **Utilities and Reminders:** Health-tracking alerts (water, sleep, etc.).

2.3 Design and Implementation Constraints

- **Platform and Device Constraints:** The system must operate on mobile platforms with limitations related to screen size, device performance, and available hardware sensors.
- **Technology Stack Constraints:** The application must be developed using predefined technologies such as cloud-based databases, RESTful APIs, and LLM-based services.
- **Third-Party Integration Constraints:** System functionality depends on the availability and limitations of external services such as fitness sensors, food databases, and notification APIs.
- **Security and Privacy Constraints:** All user data shall be transmitted and stored using secure communication and encryption mechanisms; access control and privacy settings shall restrict data visibility in compliance with standard data protection practices.
- **Performance and Storage Constraints:** The system shall manage computational resources and storage efficiently; historical user data may require archiving or summarization to prevent degradation in application performance.
- **Architectural Design Constraint:** The system must follow a modular, layered architecture separating presentation, business logic, and data management to ensure maintainability and scalability.

2.4 Assumptions and Dependencies

1. **Device Capability:** Users possess smartphones with functioning accelerometers, gyroscopes, and supported OS versions (Android/iOS).
2. **Internet Connectivity:** The system requires periodic internet access for data synchronization, AI analysis, and trainer interactions.
3. **Third-Party Services:** Continued availability of external services for social auth (Google/Apple), push notifications, and LLM APIs.
4. **User Data Accuracy:** Analytics rely on the user providing accurate manual inputs for nutrition and specific workout details.
5. **Backend Availability:** Cloud infrastructure availability is required for data processing.

3. Specific Requirements

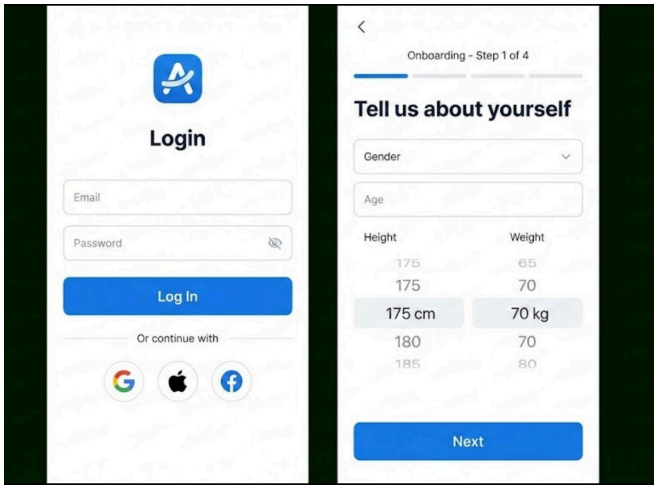
3.1. External Interface Requirements

3.1.1 User Interfaces

The system provides two distinct, role-based interfaces optimized for mobile touch interaction in portrait orientation: **Fitness User** and **Trainer**. A third specially designed interface will be the **Administrator** who shall be able to toggle the interfaces between the two. Since the interface of an administrator is a combination of the two, we omit the redundancy and focus on the **User** and **Trainer**.

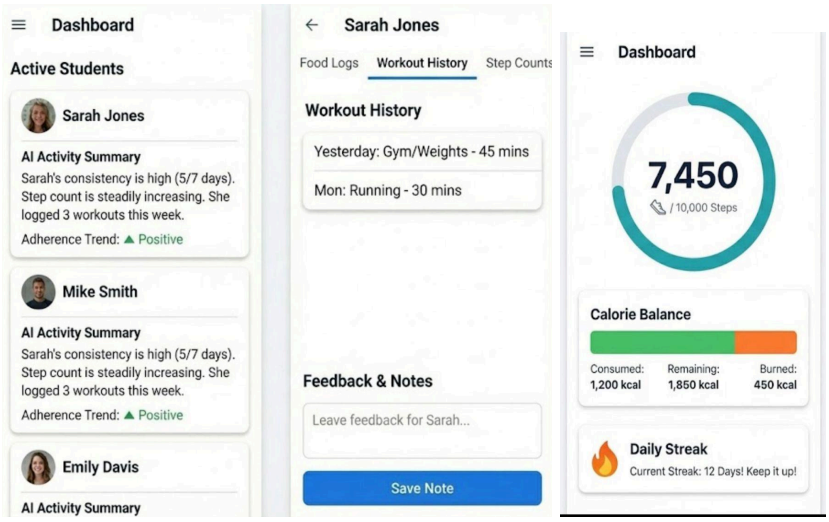
A. Fitness User Interface

- **Login & Onboarding Interface**
    - **Visual Description:** A clean welcome screen featuring fields for Email and Password, alongside "Login with Google/Apple" buttons. The subsequent onboarding screens display sliders and input fields for height, weight, age, and fitness goals.
    - **Interaction:** Users enter credentials or tap a social provider to authenticate. During onboarding, users swipe through screens and input biometric data to initialize their profile.

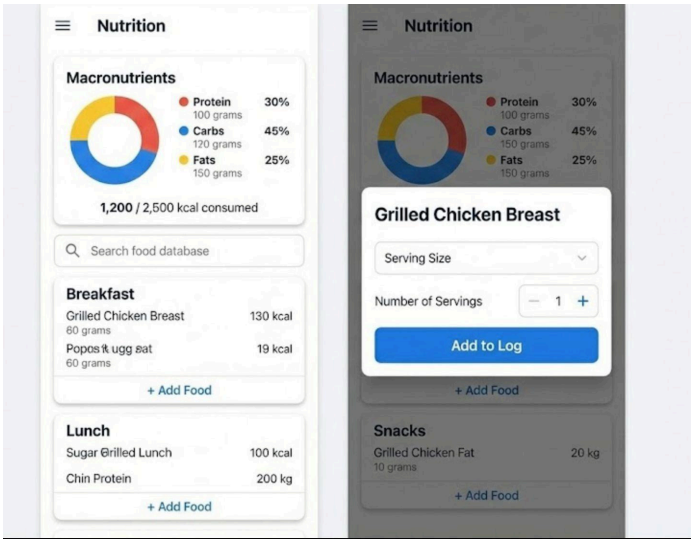


- **Dashboard (Home Screen)**
    - **Visual Description:** Features a central circular progress indicator for daily steps and a summary card for Calorie Balance (Eaten vs. Burned). A "Daily Streak" flame icon appears at the top.

- **Interaction:** Users view real-time progress at a glance. Tapping the step circle or calorie card navigates to the detailed Statistics view.

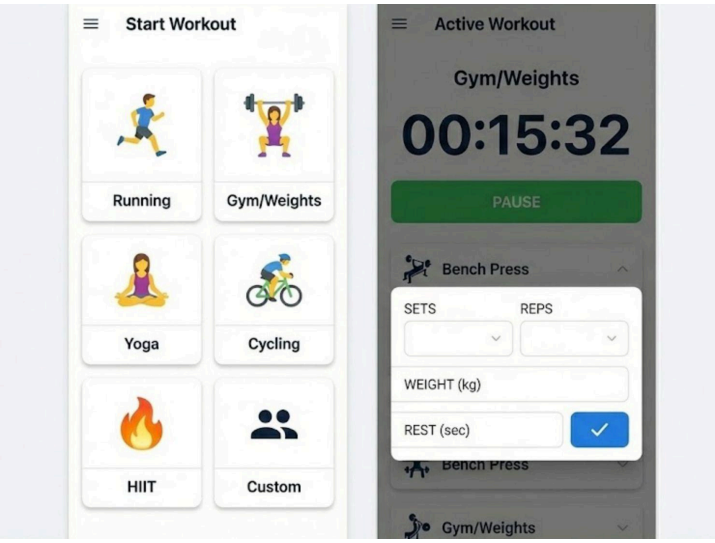


- **Food Logging Interface**
  - **Visual Description:** Displays a search bar at the top with a list of recently used foods below. A stacked bar chart visualizes the day's macronutrient split (Protein/Carbs/Fats) against targets.
  - **Interaction:** Users type to search the external database, select an item, use a stepper control to adjust portion size, and tap "Log" to save.

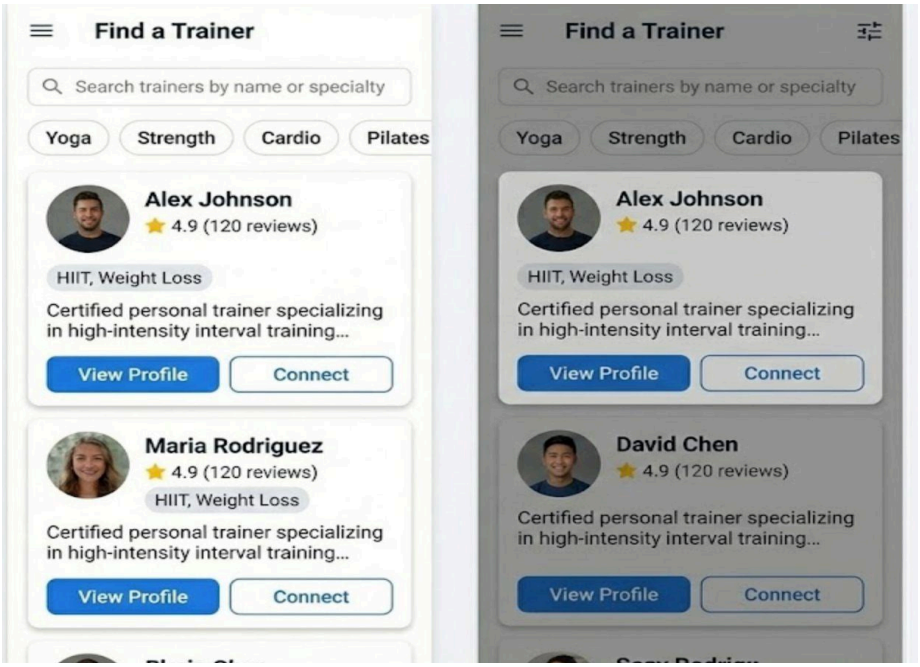


- **Exercise & Timer Interface**

- **Visual Description:** A grid menu displays workout types (Running, Gym, Yoga). Active sessions show a high-contrast timer. Gym modes include rows for inputting Sets, Reps, and Weight.
- **Interaction:** Tapping a workout type starts the session. Users interact with Start/Stop buttons for the timer and use the numeric keypad to log weight data between sets.

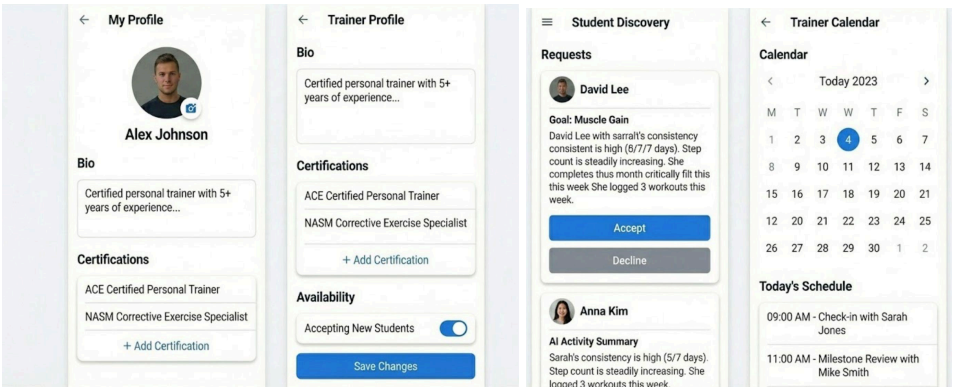


- **Statistics, History & Calendar Interface**
  - **Visual Description:** Interactive line and bar graphs show trends for steps and weight. A calendar view below utilizes color-coded indicators: green dots for completed past activities and blue bell icons for upcoming scheduled alarms.
  - **Interaction:** Users toggle between Weekly/Monthly tabs to view graphs. Users tap future dates on the calendar to open a "Set Reminder" modal, where they can configure specific times and workout types for push notification alarms.
- **Social & Trainer Finder Interface**
  - **Visual Description:** A scrollable leaderboard ranks friends by step count. The marketplace displays trainer profile cards with photos, ratings, and specializations.
  - **Interaction:** Users tap a friend’s name to view their public summary or tap a trainer's card to send a connection request.



B. Trainer User Interface

- **Trainer Dashboard**
  - **Visual Description:** A clean list view of active students, displaying their profile photo and name. An AI-generated text summary box highlights recent student trends and adherence alerts.
  - **Interaction:** Trainers scroll to monitor their roster and tap a specific student row to open their detailed profile.



- **Student Profile & Feedback**
  - **Visual Description:** Read-only tabs mirror the student's Logs, Workouts, and History. A prominent text input field or "Add Note" button is available for coaching.

    - **Interaction:** Trainers navigate tabs to review data and type feedback notes, which are pushed to the student's timeline.

- **Requests & Calendar**
    - **Visual Description:** A list of pending coaching requests with "Accept/Decline" actions, alongside a calendar marking upcoming student check-ins.
    - **Interaction:** Trainers tap action buttons to manage their roster and tap calendar dates to view scheduled milestones.

3.1.2 Hardware Interfaces

The software interacts directly with the mobile device hardware to enable automation and feedback.

- **Inertial Measurement Unit (IMU):** The system accesses the accelerometer and gyroscope sensors to detect motion patterns and automatically increment step counts without user intervention.
- **Touchscreen Display:** Serves as the primary input for navigation and data entry, and the output for all visual dashboards and graphs.
- **Haptic Engine:** The system triggers the device's vibration motor to provide tactile feedback when a workout timer completes or an interval alert is triggered.

3.1.3 Software Interfaces

The system relies on secure API connections to external services for core functionality.

- **Cloud Backend & Database:** Stores all user profiles, biometric data, logs, and social connections. The app communicates via RESTful APIs over HTTPS to ensure data persistence across devices.
- **Food & Nutrition Database API:** The system queries a third-party nutrition API to retrieve real-time calorie and macronutrient data for user food searches.
- **LLM Model API:** The system sends anonymized weekly activity logs to a Large Language Model API to generate personalized health summaries and beginner tips for the user.

3.2 Functional Requirements

3.2.1 User-Centric Functional Requirements

F1 – Authentication
The system shall allow users to register, authenticate, log out, and recover passwords securely.

F2 – Step Tracking
The system shall continuously track steps using mobile sensors and update the dashboard in near real time.

F3 – Meal Management
The system shall allow users to search for food items, log meals with custom portion sizes, and calculate daily calorie and macronutrient totals.

F4 – Workout Execution
The system shall provide workout timers that continue running in the background and notify the user upon completion.

F5 – Progress Visualization
The system shall display historical trends for fitness metrics through graphs and calendar views. The calendars can also be used for setting reminders for medications and hydration.

F6 – Social Connectivity
The system shall allow users to search for other users, send and accept friend requests, and view shared activity statistics.

F7 – Trainer Connection
The system shall allow users to browse trainer profiles and initiate connection requests. Upon acceptance, trainers shall receive read-only access to user data.

F8 – AI Insights
The system shall generate a weekly health summary using AI by analyzing user activity, nutrition, and workout consistency.

3.2.2 Trainer-Centric Functional Requirements

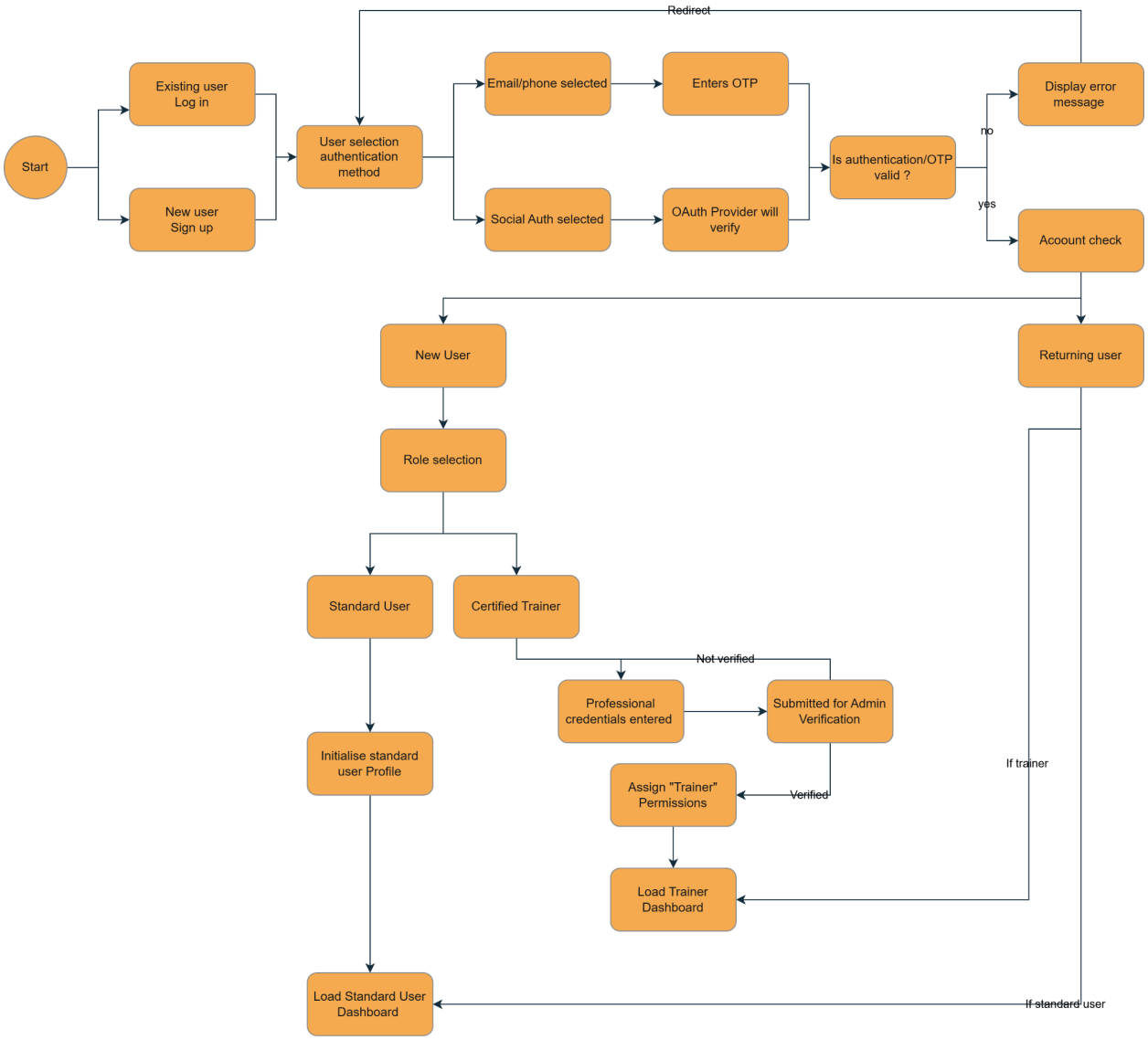
F9 – Student Monitoring
The system shall allow trainers to view and filter student activity data by date using a calendar-based interface.

F10 – Request Management
The system shall allow trainers to accept or reject incoming student coaching requests.

F11 – Feedback and Guidance
The system shall allow trainers to provide feedback, recommendations, and notes directly on student profiles.

3.3 Use Case Model

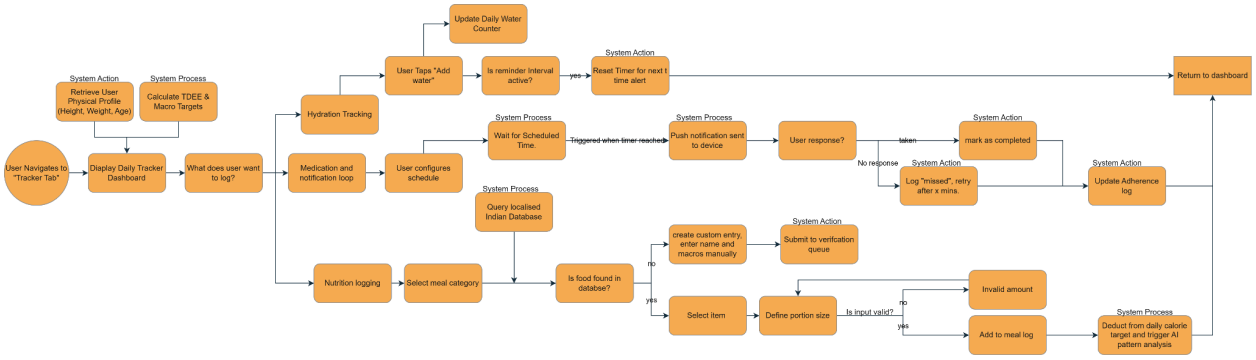
3.3.1 Use Case #1 User Authentication & Role Setup (U1)



- **Purpose:** To allow users to securely access the system and establish their identity as either a Standard User or a Certified Trainer.

- **Requirements Traceability:** F1, 4.2.1 (Auth Access), 4.2.2 (Privacy), 4.2.7 (Brute Force)
- **Priority:** High
- **Preconditions:** The application is installed and the device has an active internet connection.
- **Post conditions:** The user is authenticated, assigned a role, and granted access to the specific dashboard associated with that role.
- **Actors:** New User, Returning User, OAuth Provider (Google/Apple).
- **Exceptions:** Incorrect OTP entered; Social Auth login fails; User attempts to access Trainer data without being a Trainer.
- **Includes:** N/A
- **Notes/Issues:** Need to define the verification process for "Certified Trainers" to ensure they are actually professionals.

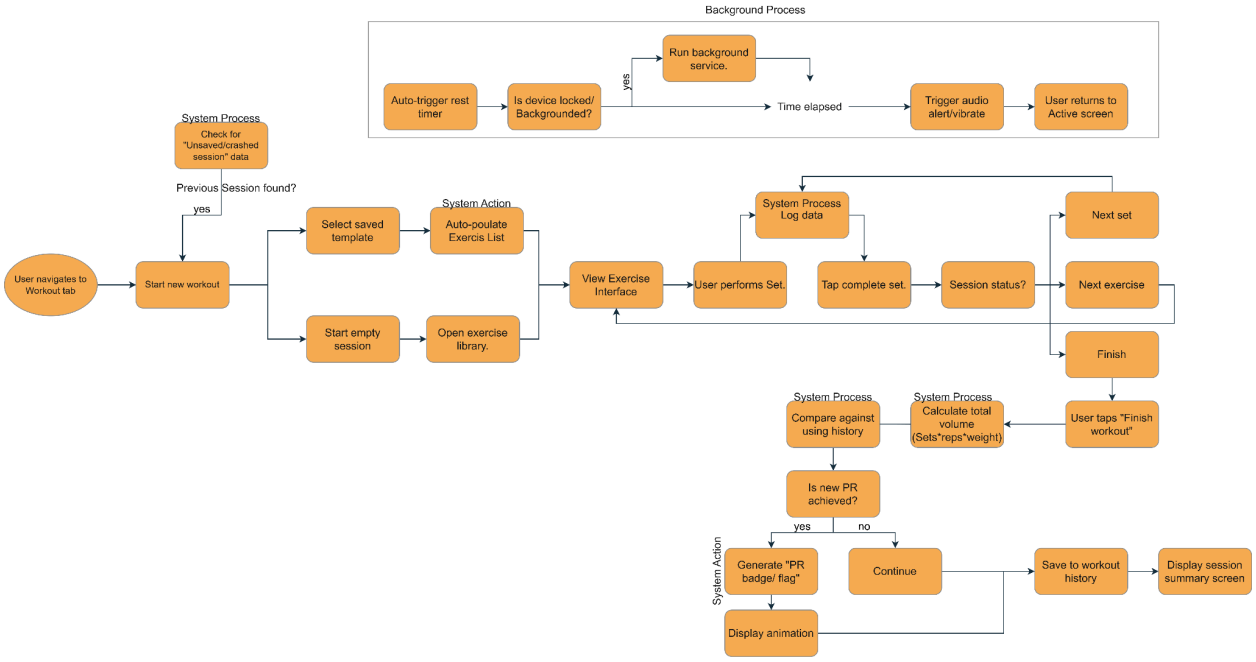
3.3.2 Use Case #2 Nutrition, Hydration & Medication Tracking (U2)



- **Purpose:** To enable users to calculate their TDEE and log daily food intake against a specific calorie/macro target, while assisting users in maintaining auxiliary health habits like water intake and timely consumption of supplements or medicine.
- **Requirements Traceability:** F3, F5, 4.1.2 (Simple Latency <500ms) , 4.3.1 (Usability/2-taps)
- **Priority:** High
- **Preconditions:** User is logged in, has completed their physical profile (height, weight, age), and notification permissions must be enabled on the user's device.

- **Post conditions:** Daily remaining calories are updated; the meal is saved to the history log; reminder logs are updated; user receives push notifications at set intervals.
- **Actors:** Standard User, Food Database API, System Notification Service.
- **Exceptions:** Food item not found in database; User enters invalid portion size; User ignores notifications; device "Do Not Disturb" mode suppresses alerts.
- **Includes:** U7 (AI Integration for pattern analysis).
- **Notes/Issues:** Indian food's database needs to handle "Katori" or "Plate" measurements accurately; users should be able to snooze reminders if they are busy.

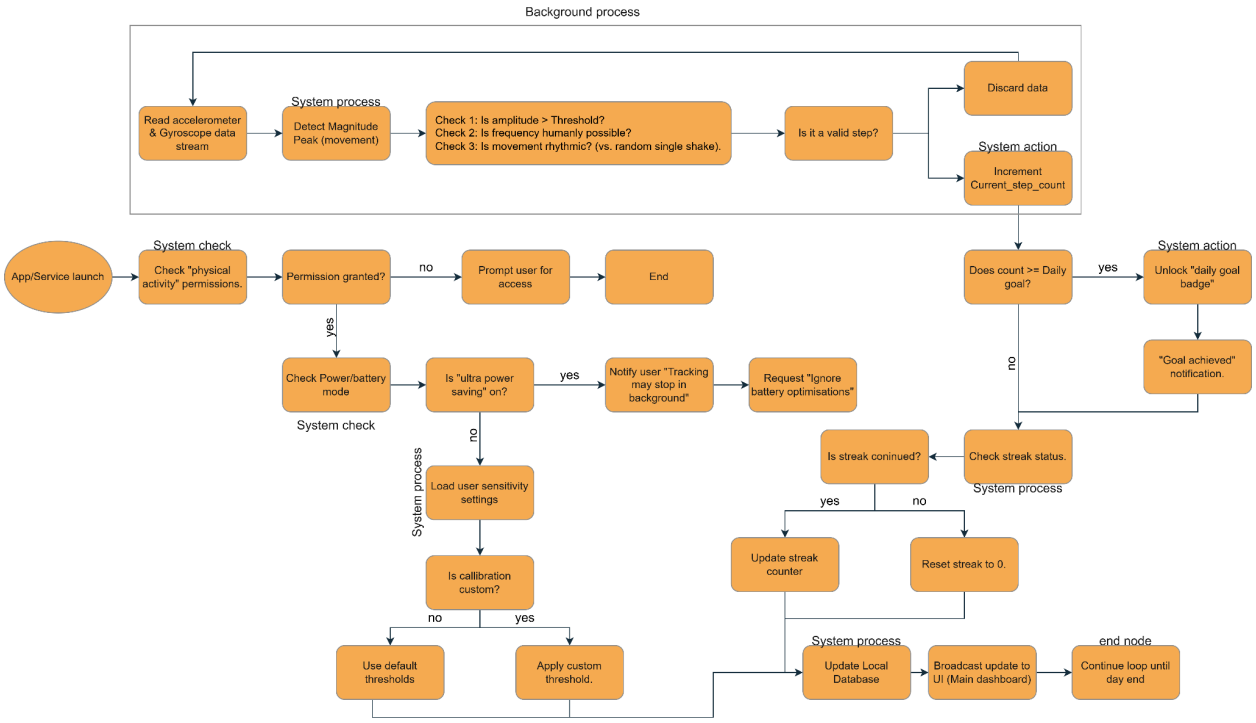
3.3.3 Use Case #3: Workout Session Management (U3)



- **Purpose:** To provide tools for users to execute, time, and record their physical exercise routines.
- **Requirements Traceability:** F4, F5, 4.1.5 (Timer Accuracy), 4.1.6 (Sync Delay)
- **Priority:** High
- **Preconditions:** User has selected a routine template or started a blank session.
- **Post conditions:** Workout volume is calculated; any new PRs are flagged in the user profile.

- **Actors:** Standard User.
- **Exceptions:** App crashes during live session (session recovery needed); Stopwatch fails to run in background.
- **Includes:** N/A
- **Notes/Issues:** Ensure the "Rest Timer" provides an audible notification even if the phone is locked.

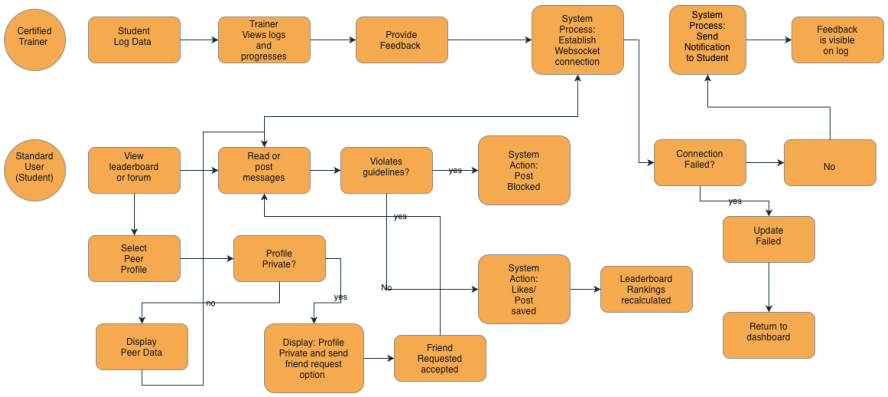
3.3.4 Use Case #4: Passive Step Tracking & Gamification (U4)



- **Purpose:** To monitor the user's movement throughout the day using hardware sensors and provide rewards for consistency.
- **Requirements Traceability:** F2, 4.1.4 (Sensor Latency), 4.2.5 (Sensor Safety)
- **Priority:** Low

- **Preconditions:** System has been granted permission to access "Physical Activity" or "Motion" sensors.
- **Post conditions:** Step count is updated in the database; "Streak" status is updated.
- **Actors:** Standard User, Mobile Device Accelerometer/Gyroscope.
- **Exceptions:** User disables sensor permissions; Device is in "Ultra Power Saving" mode which kills background processes.
- **Includes:** N/A
- **Notes/Issues:** Filtering logic must be robust to prevent users from "shaking" the phone to cheat.

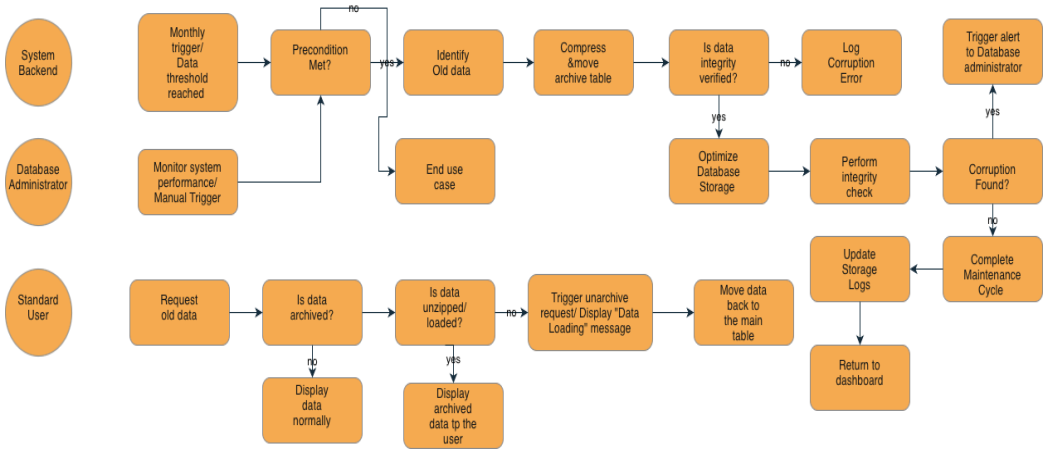
3.3.5 Use Case #5: Trainer-Student Feedback & Social Interaction (U5)



- **Purpose:** To allow trainers to monitor student progress and provide professional guidance directly through the app, while fostering user engagement and retention through peer support, competitive leaderboards, and discussion forums.
- **Requirements Traceability:** F6, F7, F9, F10, F11, 4.1.3 (Concurrent Users), 4.2.1 (Role Access)
- **Priority:** Medium
- **Preconditions:** A "Standard User" has accepted a link request from a "Certified Trainer," created a profile, and established a network connection.
- **Post conditions:** Trainer feedback is visible on the student's log; student receives a notification; user interactions (likes/posts) are saved; leaderboard rankings are recalculated based on new data.

- **Actors:** Certified Trainer, Standard User (Student).
- **Exceptions:** Student revokes privacy permission; WebSocket connection fails; user attempts to view data of a friend who has set their profile to "Private"; forum post violates community guidelines.
- **Includes:** N/A
- **Notes/Issues:** Real-time updates via WebSockets are critical for the "Live" feel of the dashboard.

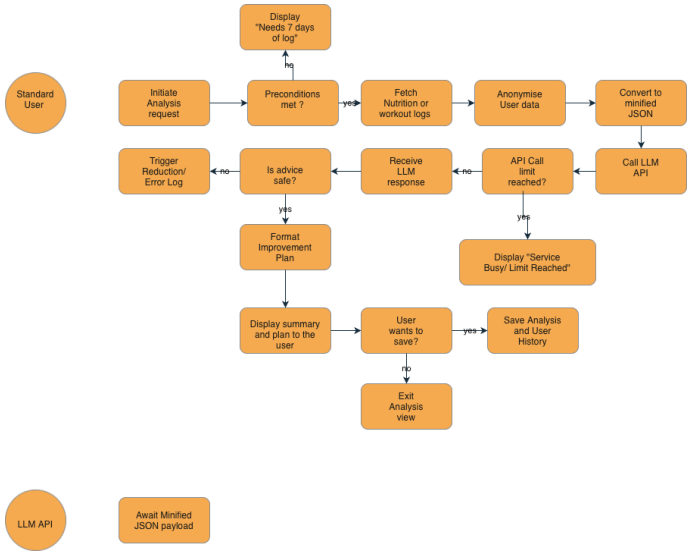
3.3.6 Use Case #6: Data Archiving & System Maintenance (U6)



- **Purpose:** To maintain system performance by managing the volume of historical data and ensuring database integrity.
- **Requirements Traceability:** 4.2.4 (Data Loss/Backup), 4.2.2 (Data Protection)
- **Priority:** Medium
- **Preconditions:** System background worker is triggered (e.g., once a month or when data exceeds 1 year).
- **Post conditions:** Old data is compressed or moved to an archive table; system storage is optimized.
- **Actors:** System Backend, Database Administrator.
- **Exceptions:** Data corruption during archiving; user attempts to access archived data that hasn't been "unzipped" yet.
- **Includes:** N/A

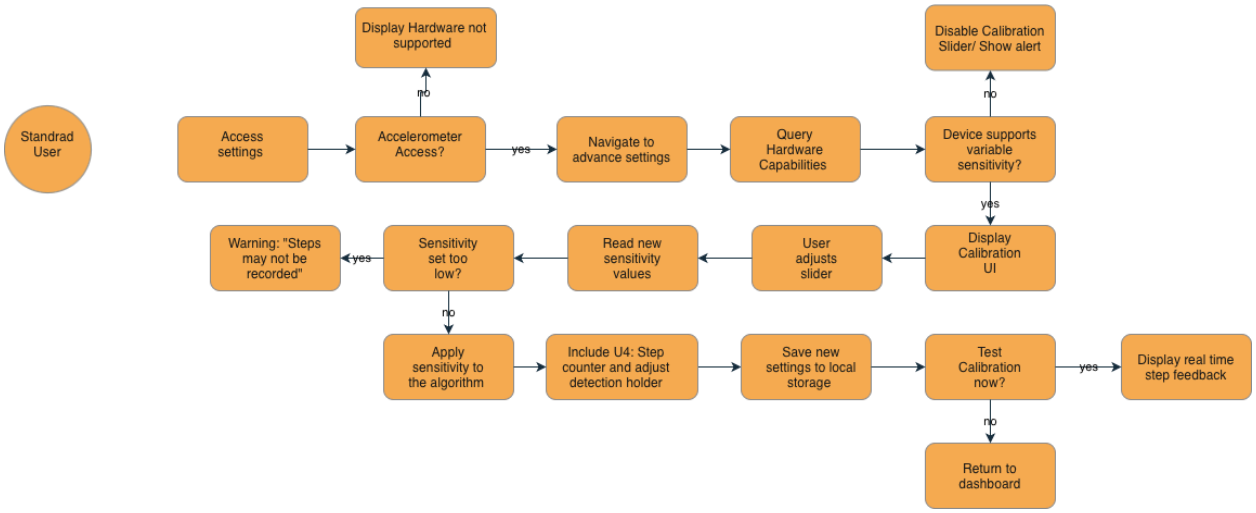
- **Notes/Issues:** Need to decide if "Archived" data remains viewable in the app or requires a special request to load.

3.3.7 Use Case #7: AI-Powered Performance Analysis (U7)



- **Purpose:** To leverage LLMs to analyze user data and provide personalized health insights and summaries.
- **Requirements Traceability:** F8, 4.1.7 (Token Opt), 4.1.8 (LLM Latency), 4.2.8 (Data Exposure)
- **Priority:** Low
- **Preconditions:** User has at least 7 days of nutrition and workout logs.
- **Post conditions:** A text-based summary and improvement plan are generated and displayed to the user.
- **Actors:** Standard User, LLM API (GPT/Llama).
- **Exceptions:** API call limit reached; LLM returns nonsensical or unsafe health advice.
- **Includes:** N/A
- **Notes/Issues:** Must ensure data sent to LLM is anonymized and follows "minified JSON" to save tokens.

3.3.8 Use Case #8: Sensor Calibration (U8)



- **Purpose:** To allow users to adjust the sensitivity of the step counter to match their specific hardware or walking style.
- **Requirements Traceability:** F2, 4.3.1 (Usability), 4.3.7 (Configurability)
- **Priority:** Low
- **Preconditions:** Access to device accelerometer.
- **Post conditions:** The step-counting algorithm's threshold is adjusted based on the calibration slider value.
- **Actors:** Standard User.
- **Exceptions:** Hardware does not support variable sensitivity; user sets sensitivity so low that no steps are recorded.
- **Includes:** N/A
- **Notes/Issues:** This is a technical setting that should likely be in "Advanced Settings" to avoid confusing casual users

4. Non-functional requirements

4.1 Performance Requirements

4.1.1 Response Time for user interactions

- **Requirement:**

  The system shall ensure that the average response time for user actions does not exceed 2 seconds.

- **Rationale:**

  Users expect quick processing for complex tasks. Delays beyond 2 seconds may lead to unwillingness to continue using the application.

4.1.2 Response Time for simple interactions (database queries, fetching user-specific data)

- **Requirement:**
  The system shall process simple actions such as fetching user-specific data, querying the calorie database, and logging user inputs (meals, workouts, water intake) within **500 milliseconds**.

- **Rationale:**
  Timely system feedback is critical for user engagement. Delays in simple fetch or logging operations beyond 500 ms may negatively impact usability and make the system appear unresponsive.

4.1.3 Concurrent User Handling

- **Requirement:**
  The system shall support concurrent access by multiple users, including read-only operations such as users viewing statistics, history, and friend data, and write operations such as trainers updating feedback or users logging meals and workouts. Concurrent read and write requests shall be processed without data inconsistency or system failure.

- **Rationale:**
  The application is expected to be used simultaneously by many users viewing their data while trainers and active users perform updates. The system must support high traffic periods without performance degradation.

4.1.4 Step Counter and Sensor Processing Latency

- **Requirement:**
  The system shall process accelerometer and gyroscope data and update the step count with a delay not exceeding 2 seconds during active usage.

- **Rationale:**
  Users expect near-real-time feedback for physical activity tracking. Excessive delay in step updates reduces trust in sensor accuracy and app reliability.

4.1.5 Workout Timer Accuracy and Drift

- **Requirement:**
  Workout timers shall maintain an accuracy of ±0.1 seconds per minute of operation during a session.

- **Rationale:**
  Accurate timers are critical for interval training, rest periods, and structured workouts. Timer drift can negatively affect workout quality and user confidence.

4.1.6 Data Synchronization Delay

- **Requirement:**
  User-generated data (meals, workouts, steps) shall be reflected in statistics, history, and dashboards within 10 minutes of submission.

- **Rationale:**
  Timely synchronization ensures users see meaningful progress updates and prevents confusion caused by inconsistent data.

4.1.7 LLM API Token Usage Optimization

- **Requirement:**
  The system shall minimize token usage for Large Language Model (LLM) API interactions by using concise prompts, controlled response lengths, and context trimming techniques.
- **Rationale:**
  LLM API costs and response latency are directly proportional to token usage. Reducing unnecessary tokens improves performance, lowers operational cost, and ensures predictable system behavior.
- **Metrics (example):**

  ➔ Maximum token budget per LLM request

  ➔ Enforced response length limits

➔ Prompt context limited to relevant user data only

4.1.8 LLM Response Latency

- **Requirement:**
  The system shall ensure that LLM-based responses are returned within an acceptable time threshold (e.g., ≤ 5 seconds for standard requests).
- **Rationale:**
  Excessive latency in AI-powered features degrades user experience and reduces trust in intelligent recommendations.

4.2 Safety and Security Requirements

4.2.1 User Authentication and Access Control

- **Requirement:**
  The system shall require users and trainers to authenticate using secure login credentials before accessing any personal or restricted features. Role-based access control shall be enforced to differentiate between regular users and trainers.
- **Rationale:**
  Authentication and role-based access prevent unauthorized access to sensitive personal, fitness, and trainer-related data.
- **Safeguards:**
  1. Role-based access control for user and trainer functionalities.
  2. Secure password-based authentication.
  3. OTP verification
- **CIA Triad:**
  1. **Confidentiality:** Prevents unauthorized access to user data.
  2. **Integrity:** Ensures only authorized users can modify data.

4.2.2 Data Protection and Privacy

- **Requirement:**
  Sensitive user data, including personal details, health-related data, and activity logs, shall be securely stored and protected from unauthorized access or disclosure.

- **Rationale:**
  Protecting personal and health data is critical to user trust and compliance with privacy expectations.
- **Safeguards:**
  1. Encryption of sensitive data during storage and transmission.
  2. Access controls restricting data access to authorized users only.
- **CIA Triad:**
  1. **Confidentiality:** Protects sensitive user information.
  2. **Integrity:** Ensures stored data remains accurate and unaltered.

4.2.3 Secure Communication

- **Requirement:**
  All communication between the mobile application, backend services, and third-party APIs (e.g., food databases, LLM services) shall be conducted over secure communication channels.
- **Rationale:**
  Secure communication prevents data interception and man-in-the-middle attacks.
- **Safeguards:**
  1. Use of encrypted communication protocols (e.g., HTTPS).
- **CIA Triad:**
  1. **Confidentiality:** Prevents data leakage during transmission.
  2. **Integrity:** Ensures transmitted data is not altered.

4.2.4 Safeguards Against Data Loss

- **Requirement:**
  The system shall implement regular data backup mechanisms to prevent loss of critical user and application data.

- **Rationale:**
  Data backups ensure recovery in case of system failure, crashes, or accidental data deletion.
- **Safeguards:**
  1. Periodic backups of user and application data.
  2. Recovery mechanisms to restore data when required.
- **CIA Triad:**
  1. **Availability:** Ensures data remains accessible after failures.

4.2.5 Sensor Data Safety

- **Requirement:**
  Sensor data collected from device accelerometers and gyroscopes shall be used strictly for step counting and activity tracking purposes.
- **Rationale:**
  Limiting sensor data usage prevents misuse and protects user privacy.
- **Safeguards:**
  1. Sensor access restricted to required functionalities only.
- **CIA Triad:**
  1. **Confidentiality:** Prevents misuse of sensor-derived data.

4.2.6 API Rate Limiting and Request Throttling

- **Requirement:**
  The system shall detect and prevent abusive behaviors such as repeated API calls, automated scraping, or denial-of-service attempts by enforcing rate limits and request validation.
- **Rationale:**
  Preventing API abuse protects backend services from overload and ensures fair usage for all users.
- **Safeguards:**
  1. API rate limiting per IP/user ID

2. Temporary IP blocking after repeated violations
3. Request validation and authentication checks

- **CIA Triad:**
  1. **Availability:** Ensures the system remains accessible under high traffic.
  2. **Integrity:** Prevents unauthorized or malicious manipulation of requests, ensuring reliable performance for legitimate users and protecting against abuse by automated scripts or bots.

4.2.7 Account Lockout and Brute-Force Protection

- **Requirement:**
  The system shall temporarily lock user accounts or delay responses after a predefined number of failed login attempts.
- **Rationale:**
  This reduces the risk of brute-force attacks on user credentials.
- **Safeguards:**
  1. Progressive login delays
  2. Temporary account lockout after multiple failures
- **CIA Triad:**
  1. **Confidentiality:** Protects user credentials
  2. **Integrity:** Prevents unauthorized access

4.2.8 LLM Data Exposure Control

- **Requirement:**
  The system shall ensure that only minimal, non-sensitive, and anonymized user data is included in LLM prompts.
- **Rationale:**
  Sending excessive or sensitive data to third-party LLM services increases privacy and security risks.
- **Safeguards:**
  1. Removal of personally identifiable information (PII) from prompts

2. Prompt sanitization and validation

- **CIA Triad:**

1. **Confidentiality:** Prevents leakage of personal data

4.3 Software Quality Attributes

4.3.1 Usability

The user interface shall be simple and intuitive, with core actions such as logging meals, workouts, and water intake, and viewing statistics accessible within two taps. The design shall be refined using user feedback to ensure ease of use for beginners as well as experienced fitness enthusiasts.

4.3.2 Maintainability

The system shall be implemented using a modular and well-documented codebase to facilitate bug fixes, feature enhancements, and long-term maintenance. Version control mechanisms shall be used to manage changes and support collaborative development.

4.3.3 Interoperability

The application shall support both Android and iOS platforms and maintain consistent behavior across supported devices. The system shall also expose compatible interfaces to integrate with external services such as online food databases and LLM-based APIs.

4.3.4 Robustness

The system shall handle common failure scenarios, including network interruptions, sensor inaccuracies, and invalid user inputs, without crashing. User-friendly error messages and fallback mechanisms shall ensure continued usability under such conditions.

4.3.5 Flexibility

The system architecture shall be designed to accommodate future enhancements such as additional health metrics, new workout types, advanced analytics, and improved AI-based recommendations. Modular design principles shall enable feature expansion without impacting existing functionality.

4.3.6 Explainability

The system should provide users with clear explanations when AI-generated recommendations or insights are presented, indicating that the content is AI-assisted. AI-generated

recommendations are advisory only and shall not be presented as medical or professional advice.

4.3.7 Configurability

The system should allow administrators to configure thresholds such as API rate limits, token budgets, and AI feature availability without redeploying the application.

5 Other Requirements

5.1 Legal Requirements and Copyright

User data collected should be stored in the country's regional data center. This is to comply with data protection and privacy rules that may be enforced by the Government of the nation. Copyright of the source code and the documents must be retained by the developers of the System.

5.2 Authentication

Permission will be required for utilising OTPs for the authentication process as this will require sending automated emails to the user's email id.

A. Data Dictionary

A.1 User Database

Variable	Description	Requirements
UserID	Unique ID of the user	Unique UUID (128-bit)
Name	Full name of the user	String
Email	Email address used for login	Valid Email Format
Hashed Password	User's login password	One-way hashed password (bcrypt)
Role	Type of user account	Enum {User, Trainer, Admin}
Biometrics	JSON object containing Height, Weight, Age	JSON / Numeric values
DailyStepGoal	The user's target step count	Integer (Default: 10,000)
TDEE	Total Daily Energy Expenditure	Numeric (Calculated)

A.2 Trainer Database

Variable	Description	Requirements
TrainerID	Unique ID of the trainer	Unique UUID
UserID	Links to the User Database	Foreign Key
Specialization	Area of expertise (e.g., Yoga, HIIT)	String / List
Rating	Average rating from students	Float (0.0 - 5.0)
Certifications	List of verified credentials	String / File Path
StudentList	List of currently enrolled students	Array of UserIDs
Availability	Calendar slots available for coaching	Date/Time Object

A.3 Workout & Activity Database

Variable	Description	Requirements
SessionID	Unique ID of the workout session	Unique UUID
UserID	ID of the user performing the workout	Foreign Key
Type	Category of exercise (Run, Gym, Swim)	String
Duration	Total time spent exercising	Time (MM:SS)
CaloriesBurned	Estimated energy expenditure	Numeric
Steps	Steps recorded during the session	Integer
LogData	Detailed sets, reps, and weights	JSON Object

A.4 Nutrition Database

Variable	Description	Requirements
FoodID	Unique ID of the food item	Unique UUID
Name	Name of the dish or ingredient	String
Calories	Energy content per serving	Numeric
Macros	Protein, Carbs, and Fats content	JSON {P, C, F}
ServingSize	Standard measurement unit	String (e.g., "100g", "1 cup")
Source	Origin of data (User-added or API)	String

B Group Log

Meeting notes and activities will be maintained here

S. No	Date	Timings	Venue	Agenda
1.	07/01/2026	10:00 AM to 11:00 AM	In front of RM	Discussed various project ideas and narrowed down to 2-3 ideas.
2.	09/01/2026	7:00 PM to 8:00 PM	RM 602(With the instructor)	Asked for help in choosing the project according to the feasibility and scope of the syllabus.
3.	16/01/2026	4:00 PM to 5:00 PM	In front of the Library.	Divided portions and planned the structure and content of the SRS.
4.	20/01/2026	2:00 PM to 3:00 PM	KD 314(With the tutor)	Pitched our structure and asked for suggestions so as to not miss anything important.
5.	22/01/2026	2:00 PM to 3:00 PM	Library	Did Final Checks and worked on the format, spacing and the final product of the SRS document.