# Software Requirements Specification

**for**

# FITLIFE

**Version 1.0 approved**

**Prepared by Zohaib Hassan 22i0800 and Fahad Faisal 22i0817**

**23rd March 2025**

# Table of Contents

# 1. Introduction

## 1.1 Purpose

This Software Requirements Specification (SRS) document specifies the requirements for the FitLife Personal Fitness & Workout Tracker web application, version 1.0. This document covers the complete system including user registration, profile management, workout tracking, calorie monitoring, AI-based recommendations, trainer interactions, progress analytics, and social features.

## 1.2 Document Conventions

This document follows these conventions:

- **Bold text** indicates feature names, important concepts, or system components
- *Italic text* indicates emphasis on critical requirements
- Code formatting represents UI elements or technical specifications
- Priority levels for requirements:
  - HIGH: Must be implemented in the initial release
  - MEDIUM: Should be implemented if time and resources permit
  - LOW: Desirable but can be deferred to future releases
- Requirements are categorized using a hierarchical numbering system (1.0, 1.1, 1.1.1)
- User stories follow the format: "As a [role], I want [goal] so that [benefit]"

## 1.3 Intended Audience and Reading Suggestions

This document is intended for:

- **Development Team**: Focus on sections 3 (System Features) and 4 (External Interface Requirements)
- **Project Managers**: Focus on sections 1 (Introduction), 2 (Overall Description), and 5 (Schedule & Milestones)
- **Quality Assurance Team**: Focus on sections 3 (System Features) and 6 (Testing Requirements)
- **Stakeholders & Product Owners**: Focus on sections 1 (Introduction) and 2 (Overall Description)
- **UI/UX Designers**: Focus on sections 2.1 (Product Perspective), 3 (System Features), and 4.1 (User Interfaces)

For a comprehensive understanding of the document, it is recommended to read the sections in sequential order. However, technical team members may wish to start with the System Features section after reviewing the Introduction and Overall Description.

## 1.4 Product Scope

FitLife is a comprehensive web-based fitness tracking system designed to help individuals maintain consistent fitness routines through personalized guidance, workout tracking, and diet management. The system aims to solve the problems of:

- Lack of proper guidance in fitness journeys
- Difficulty in tracking workouts consistently
- Challenges in managing diet and nutrition
- Absence of personalized recommendations based on fitness levels
- Limited access to professional training expertise

The primary objectives of FitLife are to:

- Enable users to set and track fitness goals based on their body type and preferences
- Provide an intuitive interface for logging and monitoring workout progress
- Track daily calorie intake and provide AI-based diet suggestions
- Generate personalized exercise recommendations based on user fitness levels
- Facilitate connections between users and fitness trainers for professional guidance

FitLife aligns with the organizational goal of promoting healthier lifestyles through accessible technology solutions and building a community-driven fitness ecosystem.

## 1.5 References


- FitLife Project Outline Document (i220800_i220817_A1.docx)
- GitHub Repository: https://github.com/zohaybhassan/FitLife
- Agile Development Methodology Guide, Version 2.0, 2023
- UI/UX Design Guidelines for Web Applications, 2024
- OWASP Security Standards for Web Applications, 2024
- Web Content Accessibility Guidelines (WCAG) 2.1
- General Data Protection Regulation (GDPR) Compliance Documentation
- RESTful API Documentation Standards, 2023
- IEEE Standard 830-1998: Recommended Practice for Software Requirements Specifications

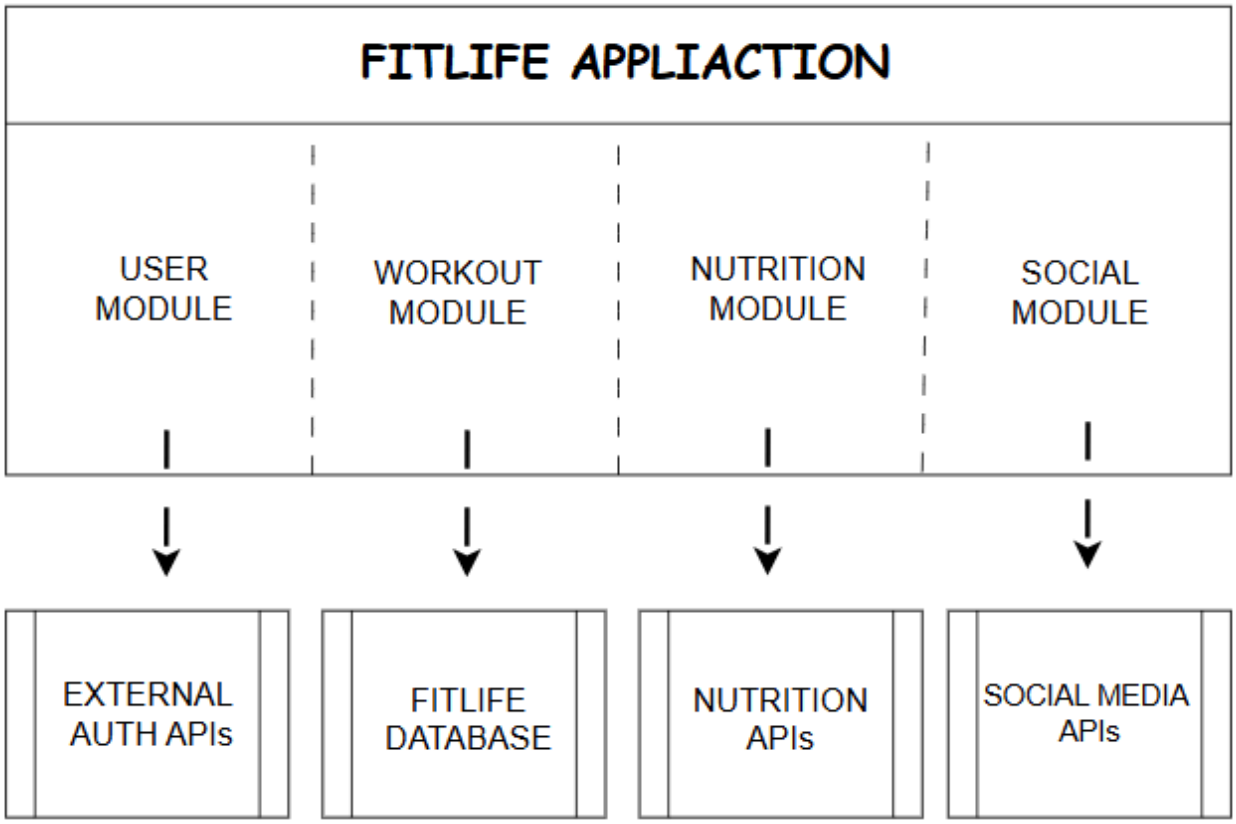

# 2. Overall Description

## 2.1 Product Perspective

FitLife is a new, self-contained web-based fitness tracking application that addresses the need for an integrated solution combining workout tracking, diet management, and personalized recommendations. While FitLife exists in a market alongside other fitness applications, it distinguishes itself through its comprehensive approach and user-friendly interface.

The system operates as a standalone web application with a client-server architecture. The client-side component runs in modern web browsers, while the server-side component handles data processing, storage, and AI-based recommendations. The application will interface with third-party authentication providers (Google/Facebook) for simplified user access and potentially with nutrition databases for food information.

The following diagram illustrates the high-level system architecture of FitLife:




## 2.2 Product Functions

FitLife provides the following major functions:

- **User Account Management**

- Registration and authentication (including third-party login options)
- Profile creation and management
- Personal fitness goal setting

- **Workout Management**

- Exercise logging and tracking

- Workout history visualization
- Exercise library with tutorials and descriptions
- Personalized workout recommendations

- **Nutrition Tracking**

- Food and meal logging
- Calorie intake tracking
- Diet plan recommendations based on fitness goals

- **Progress Monitoring**

- Statistical analysis of workout performance
- Weight and body measurement tracking
- Visual representation of progress through charts and reports

- **Professional Guidance**

- Connection with fitness trainers
- Booking and scheduling of coaching sessions
- Secure messaging system

- **Community Engagement**

- Participation in fitness challenges
- Connection with other users
- Achievement sharing on social media

- **Notification System**

- Workout and meal reminders
- Progress alerts and milestone achievements
- Personal trainer communications

## 2.3 User Classes and Characteristics

FitLife will serve several user classes, each with specific needs and usage patterns:

1. **Standard Users** (Primary)

- Characteristics: Individuals seeking to improve fitness and track progress
- Technical expertise: Basic to intermediate computer skills
- Usage frequency: Daily to several times per week
- Primary features used: Workout logging, calorie tracking, progress monitoring

- Priority: High

2. **Fitness Beginners** (Primary)

- Characteristics: New to fitness regimens, requiring extra guidance
- Technical expertise: Basic computer skills
- Usage frequency: Daily to several times per week
- Primary features used: Exercise tutorials, beginner workouts, basic tracking
- Priority: High

3. **Advanced Fitness Enthusiasts** (Secondary)

- Characteristics: Experienced in fitness, looking for detailed tracking and analysis
- Technical expertise: Intermediate computer skills
- Usage frequency: Daily
- Primary features used: Advanced workout tracking, detailed analytics, community challenges
- Priority: Medium

4. **Fitness Trainers/Coaches** (Secondary)

- Characteristics: Professional trainers who guide users
- Technical expertise: Intermediate computer skills
- Usage frequency: Daily
- Primary features used: Client management, program creation, communication tools
- Priority: Medium

5. **System Administrators** (Support)

- Characteristics: Technical staff maintaining the system
- Technical expertise: Advanced computer skills
- Usage frequency: As needed
- Primary features used: User management, system configuration, data management
- Priority: Low (but essential for system operation)

## 2.4 Operating Environment

FitLife will operate in the following environment:

- **Client-Side**:

- Web Browsers: Google Chrome (version 90+), Mozilla Firefox (version 85+), Safari (version 14+), Microsoft Edge (version 90+)
- Mobile Browsers: Safari iOS 14+, Chrome for Android 90+
- Screen Resolutions: Responsive design supporting 320px to 2560px widths
- Internet Connection: Minimum 1 Mbps recommended

- **Server-Side**:

- Operating System: Linux (Ubuntu 20.04 LTS or higher)
- Web Server: Nginx 1.18+ or Apache 2.4+
- Database: MySQL 8.0+ or PostgreSQL 12+
- Runtime Environment: Node.js 16+ or Python 3.8+ with appropriate frameworks

- **External Systems**:

- OAuth providers (Google, Facebook) for authentication
- Cloud storage for user data
- Email service for notifications

## 2.5 Design and Implementation Constraints

The following constraints will affect the design and implementation of FitLife:

- **Security Requirements**:

- Must comply with data protection regulations (GDPR, CCPA)
- Secure storage of personal and health information
- Implementation of proper authentication and authorization mechanisms

- **Technical Constraints**:

- Development using JavaScript/TypeScript for frontend (React framework)
- Backend implementation using Node.js or Python
- RESTful API architecture
- Responsive design for cross-platform compatibility

- **Hardware Limitations**:

- Server resources based on initial user projections with scalability considerations
- Optimization for mobile devices with limited processing power and bandwidth

- **Development Standards**:

- Code must follow established team coding standards
- Version control through Git with branch management strategy
- Comprehensive test coverage (unit tests, integration tests)

- **External Dependencies**:

- Reliance on third-party authentication systems
- Potential dependency on nutrition databases for food information
- Integration with social media platforms for sharing functionality

## 2.6 User Documentation

The following user documentation will be provided with FitLife:

- **User Manual**:
  - Comprehensive guide covering all system features
  - Available as downloadable PDF and searchable online documentation
- **Quick Start Guide**:
  - Brief introduction to essential features
  - Step-by-step instructions for first-time users
- **Video Tutorials**:
  - Short instructional videos for key functionalities
  - Available within the application and on a dedicated YouTube channel
- **In-App Help**:
  - Context-sensitive help tooltips
  - Searchable FAQ section
  - Interactive walkthrough for new users
- **Exercise Library Documentation**:
  - Detailed descriptions and instructions for exercises
  - Categorized by muscle groups and difficulty levels

All documentation will be available in English initially, with plans to support additional languages based on user demographics.

## 2.7 Assumptions and Dependencies

**Assumptions**

- Users have basic knowledge of fitness concepts and terminology
- Users have consistent internet access for regular application usage
- Modern web browsers are used that support HTML5, CSS3, and ES6+ JavaScript
- Mobile users have sufficient screen space to interact with the application effectively
- Users are willing to input personal fitness data to receive personalized recommendations

**Dependencies**

- **External Services**:
  - Availability and continued support of OAuth providers (Google, Facebook)
  - Access to reliable nutrition databases for food information
  - Email service providers for notification delivery

- **Technical Dependencies**:
  - Backend frameworks and libraries (Express.js, Django, etc.)
  - Frontend frameworks (React, Redux, etc.)
  - Data visualization libraries (D3.js, Chart.js, etc.)
  - Database management systems
- **Development Dependencies**:
  - Team skills and availability
  - Access to testing environments that mirror production
  - Code review and quality assurance processes
- **Business Dependencies**:
  - Timely stakeholder feedback on prototypes and releases
  - Resource allocation for hosting and ongoing maintenance

# 3. External Interface Requirements

## 3.1 User Interfaces

FitLife will provide a responsive, intuitive user interface with these key characteristics:

- **Design Consistency**: Standardized design system with consistent typography, colors, spacing, and components across all screens.
- **Responsive Design**: Interface adapts to screen sizes from mobile (320px) to desktop (2560px).
- **Navigation**: Primary navigation bar with main sections (Dashboard, Workouts, Nutrition, Progress, Community).
- **Standard Elements**: All screens include header with logo and user profile access, consistent footer, toast notifications for errors, and loading indicators.
- **Accessibility**: WCAG 2.1 AA compliance with proper color contrast, keyboard navigation, and screen reader support.

**Key Screens**:

- **Dashboard**: Summary widgets, upcoming activities, progress highlights, quick actions
- **Workout Tracking**: Exercise selection, log entry forms, timer functionality, historical comparison
- **Nutrition Tracking**: Food search with autocomplete, meal categorization, nutrient visualization
- **Progress Analytics**: Interactive charts, measurement tracking, goal achievement visualization

Detailed UI specifications will be documented separately in the FitLife UI/UX Design Guide.

## 3.2 Hardware Interfaces

As a web-based application, FitLife will interact with these hardware components:

- **Client Devices**:
  - Desktop computers: Support for standard resolutions and input devices
  - Mobile devices: Touch interface optimization, portrait/landscape support
  - Tablets: Optimized layouts for medium-sized screens
- **Fitness Wearables** (Future Implementation):
  - Design prepared for integration with heart rate monitors, step counters, and smart watches through their APIs

All device interactions will occur through standard web browser interfaces.

## 3.3 Software Interfaces

FitLife will interface with these software systems:

- **Authentication Services**:
  - Google Sign-In API (v2.0+) and Facebook Login API (v12.0+)
  - Purpose: User authentication and basic profile retrieval
  - Implementation: OAuth 2.0 protocol
- **Database Management System**:
  - PostgreSQL (v12.0+) or MongoDB (v4.4+)
  - Purpose: Storage of user profiles, workout data, nutrition logs
- **External Data Sources**:
  - Nutrition Database API (Edamam or similar)
  - Exercise Library Database (custom or licensed)
  - Purpose: Provide food nutrition data and exercise information
- **Analytics & Notifications**:
  - Google Analytics for usage metrics
  - SendGrid API (v3) for email notifications
  - Firebase Cloud Messaging for push notifications
- **Recommendation Engine**:
  - Custom ML model or third-party service
  - Purpose: Generate personalized workout and nutrition recommendations

## 3.4 Communications Interfaces

FitLife will use these communication methods:

- **Network Protocols**:

    - HTTPS (TLS 1.2+) for all client-server communications
    - RESTful API design pattern with JSON data format
    - WebSockets for real-time features (trainer chat, workout timers)
- **API Standards**:
    - Standard HTTP methods with resource-oriented endpoints
    - JWT authentication
    - API versioning in URL path (/api/v1/)
- **Data Transfer Requirements**:
    - Image optimization and progressive loading
    - Client-side caching of static resources
    - Minimum connection speed: 1 Mbps
- **Security Requirements**:
    - HTTPS for all communications
    - JWT for API authentication
    - CORS policy implementation
    - Protection against common web vulnerabilities

# 4. System Features

This section details the primary functional capabilities of the FitLife platform, describing each major service the application will provide to users seeking to improve their fitness journey through tracking, personalization, and community engagement.

## 4.1 User Registration and Profile Management

### 4.1.1 Description and Priority

The User Registration and Profile Management feature allows users to create accounts, customize their profiles with personal fitness information, and manage their account settings. This feature serves as the foundation for personalized experiences throughout the application. **Priority: High** (Benefit: 9, Penalty: 8, Cost: 6, Risk: 4)

### 4.1.2 Stimulus/Response Sequences

- **Registration Flow:**
    - User navigates to registration page and inputs required information
    - System validates input and creates user account
    - System prompts user to complete profile with fitness-related information
    - System generates initial dashboard based on profile information

- **Profile Management Flow:**
    - User selects profile from dashboard
    - System displays editable profile fields
    - User modifies information (weight, height, goals, etc.)
    - System validates and saves changes
    - System updates personalized recommendations based on new profile data

### 4.1.3 Functional Requirements

REQ-1: The system shall allow new users to register using email/password or social media authentication (Google/Facebook).
REQ-2: The system shall collect and store user fitness profile data including height, weight, age, fitness goals, and experience level.
REQ-3: The system shall allow users to update their profile information at any time.
REQ-4: The system shall implement secure password reset functionality for users who forget their login credentials.
REQ-5: The system shall support profile image uploads with size restrictions of 5MB and acceptable formats (JPEG, PNG).
REQ-6: The system shall store user body measurements and provide historical tracking of these metrics.

## 4.2 Workout Logging and Tracking

### 4.2.1 Description and Priority

The Workout Logging and Tracking feature enables users to record their exercise activities, including type, duration, intensity, and details like sets and repetitions. The system records this data to track progress over time. **Priority: High** (Benefit: 9, Penalty: 7, Cost: 5, Risk: 3)

### 4.2.2 Stimulus/Response Sequences

- **Workout Creation Flow:**
    - User selects "Log Workout" from dashboard
    - System presents workout template options
    - User selects workout type and adds exercises
    - User inputs exercise details (sets, reps, weight, duration)
    - System saves and confirms workout entry
- **Workout History Flow:**
    - User navigates to workout history
    - System displays calendar view of past workouts
    - User selects specific workout date

- System displays detailed workout information

### 4.2.3 Functional Requirements

REQ-1: The system shall provide a searchable exercise library with at least 500 exercises across different categories.
REQ-2: The system shall allow users to log workouts with details including exercise type, sets, repetitions, weight, and duration.
REQ-3: The system shall enable users to categorize workouts (strength, cardio, flexibility, etc.).
REQ-4: The system shall provide workout templates for users of different fitness levels.
REQ-5: The system shall store workout history and allow users to view, filter, and analyze past workouts.
REQ-6: The system shall calculate and display workout statistics (total volume, progression over time).
REQ-7: The system shall allow users to duplicate previous workouts for easy logging.

## 4.3 Calorie and Nutrition Tracking

### 4.3.1 Description and Priority

The Calorie and Nutrition Tracking feature allows users to record their daily food intake, track caloric consumption, and monitor nutritional balance. The system calculates daily caloric needs based on user data and goals. **Priority: High** (Benefit: 8, Penalty: 7, Cost: 7, Risk: 5)

### 4.3.2 Stimulus/Response Sequences

- **Meal Logging Flow:**
  - User selects "Log Meal" from dashboard
  - System presents meal categories (breakfast, lunch, dinner, snack)
  - User searches for and selects food items from database
  - User inputs portion sizes
  - System calculates and displays nutritional information
  - System updates daily nutrition totals
- **Nutrition Summary Flow:**
  - User navigates to nutrition dashboard
  - System displays daily, weekly, and monthly nutrition summaries
  - User can filter by specific nutrients or time periods

### 4.3.3 Functional Requirements

REQ-1: The system shall provide a searchable food database with nutritional information for at least 10,000 common food items.
REQ-2: The system shall calculate daily caloric needs based on user profile data, activity level, and fitness goals.
REQ-3: The system shall track macronutrient (protein, carbohydrates, fats) and micronutrient intake.
REQ-4: The system shall allow users to create and save custom meals and recipes.
REQ-5: The system shall generate visual reports of nutritional intake over time.

REQ-6: The system shall provide warnings when daily caloric or nutritional targets are exceeded.
REQ-7: The system shall allow barcode scanning for quick food logging.

## 4.4 AI-Based Personalized Recommendations

### 4.4.1 Description and Priority

The AI-Based Personalized Recommendations feature analyzes user data to provide customized workout plans, nutritional suggestions, and training adjustments. This feature leverages machine learning to optimize user progress toward their fitness goals. **Priority: Medium** (Benefit: 9, Penalty: 5, Cost: 8, Risk: 7)

### 4.4.2 Stimulus/Response Sequences

- **Initial Recommendation Flow:**
  - User completes profile setup
  - System analyzes user data
  - System generates personalized workout and nutrition plans
  - User reviews and accepts or modifies recommendations
- **Ongoing Adjustment Flow:**
  - System analyzes user progress data weekly
  - System identifies plateaus or areas for improvement
  - System suggests workout or nutrition adjustments
  - User reviews and implements recommendations

### 4.4.3 Functional Requirements

REQ-1: The system shall generate personalized workout plans based on user's fitness goals, experience level, and available equipment.
REQ-2: The system shall adjust workout recommendations based on user progress and feedback.
REQ-3: The system shall provide alternative exercise suggestions when users report injuries or limitations.
REQ-4: The system shall recommend meal plans aligned with user's caloric needs and nutritional requirements.
REQ-5: The system shall incorporate machine learning algorithms to improve recommendation accuracy over time.
REQ-6: The system shall provide difficulty progression to continually challenge users as they improve.

## 4.5 Trainer and Coach Connection

### 4.5.1 Description and Priority

The Trainer and Coach Connection feature enables users to connect with fitness professionals for personalized guidance, feedback, and motivation. This feature bridges the gap between self-guided fitness tracking and professional coaching. **Priority: Medium** (Benefit: 7, Penalty: 4, Cost: 7, Risk: 6)

### 4.5.2 Stimulus/Response Sequences

- **Trainer Search Flow:**
  - User navigates to trainer marketplace
  - System displays available trainers with filters (specialization, price, rating)
  - User reviews trainer profiles and selects preferred trainer
  - System facilitates connection and payment processing
- **Coaching Interaction Flow:**
  - User shares workout/nutrition data with trainer
  - Trainer reviews data and provides feedback
  - User implements recommendations
  - System tracks progress and shares updates with trainer

### 4.5.3 Functional Requirements

REQ-1: The system shall provide a searchable directory of verified fitness trainers and nutrition coaches.
REQ-2: The system shall enable secure messaging between users and trainers.
REQ-3: The system shall allow trainers to create and assign custom workout plans to their clients.
REQ-4: The system shall support video consultations between users and trainers.
REQ-5: The system shall implement a rating and review system for trainers.
REQ-6: The system shall facilitate scheduling and reminders for coaching sessions.
REQ-7: The system shall provide trainers with a dashboard to monitor their clients' progress.

## 4.6 Progress Analytics and Reporting

- **4.6.1 Description and Priority**

The Progress Analytics and Reporting feature transforms user workout and nutrition data into meaningful insights, visualizations, and progress reports. This feature helps users understand their progression and identify patterns. **Priority: Medium** (Benefit: 8, Penalty: 5, Cost: 6, Risk: 4)

### 4.6.2 Stimulus/Response Sequences

- **Analytics Dashboard Flow:**
  - User navigates to analytics section
  - System processes historical data
  - System generates visual representations of key metrics
  - User selects time periods or specific metrics for detailed analysis
- **Progress Report Flow:**
  - User requests progress report
  - System compiles data from selected time period
  - System generates comprehensive report with highlights
  - User reviews report and can share or download it

### 4.6.3 Functional Requirements

REQ-1: The system shall track and display key fitness metrics over time (weight, body measurements, strength progression).

REQ-2: The system shall generate visual representations of workout and nutrition data through charts and graphs.
REQ-3: The system shall calculate and display trend analysis for key metrics.
REQ-4: The system shall provide weekly and monthly progress reports highlighting achievements and areas for improvement.
REQ-5: The system shall enable comparison of current performance against historical benchmarks.
REQ-6: The system shall allow users to export progress data in common formats (PDF, CSV).
REQ-7: The system shall highlight correlations between nutrition habits and workout performance.

## 4.7 Community and Challenges

### 4.7.1 Description and Priority

The Community and Challenges feature creates a social environment where users can participate in fitness challenges, interact with other members, and build accountability networks. This feature fosters motivation through friendly competition and community support. **Priority: Low** (Benefit: 7, Penalty: 3, Cost: 7, Risk: 5)

### 4.7.2 Stimulus/Response Sequences

- **Challenge Participation Flow:**
  - User browses available challenges
  - System displays challenge details (duration, requirements, rewards)
  - User joins selected challenge
  - System tracks user's relevant activities toward challenge goals
  - System updates leaderboard and notifies user of progress
- **Community Interaction Flow:**
  - User navigates to community section
  - System displays activity feed with recent posts
  - User creates post or interacts with others' content
  - System notifies relevant users of interactions

### 4.7.3 Functional Requirements

REQ-1: The system shall enable users to create and join fitness challenges with defined goals and durations.
REQ-2: The system shall maintain leaderboards for active challenges with real-time updates.
REQ-3: The system shall allow users to follow others and view a customized activity feed.
REQ-4: The system shall support formation of fitness groups based on common goals or interests.
REQ-5: The system shall enable users to share accomplishments with community members.
REQ-6: The system shall provide achievement badges for reaching fitness milestones.
REQ-7: The system shall facilitate accountability partnerships between users.

## 4.8 Reminders and Notifications

### 4.8.1 Description and Priority

The Reminders and Notifications feature keeps users engaged through timely alerts about scheduled workouts, meal times, achievements, and community activities. This feature supports habit formation and consistent app usage. **Priority: Medium** (Benefit: 7, Penalty: 6, Cost: 4, Risk: 3)

#### 4.8.2 Stimulus/Response Sequences

- **Reminder Setup Flow:**
  - User navigates to reminder settings
  - System displays configurable reminder categories
  - User selects desired reminders and frequency
  - System confirms settings and activates reminders
- **Notification Delivery Flow:**
  - System identifies scheduled reminder
  - System sends notification through preferred channel
  - User receives and acknowledges notification
  - System logs user response for pattern analysis

#### 4.8.3 Functional Requirements

REQ-1: The system shall allow users to set customizable reminders for workouts, meals, and water intake.
REQ-2: The system shall support multiple notification channels (push notifications, email, SMS).
REQ-3: The system shall send achievement notifications when users reach fitness milestones.
REQ-4: The system shall provide smart reminders based on user behavior patterns.
REQ-5: The system shall notify users of community challenge updates and deadlines.
REQ-6: The system shall allow users to customize notification frequency and quiet hours.
REQ-7: The system shall send weekly summary notifications highlighting progress and upcoming activities.
.

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

FitLife shall meet the following performance requirements to ensure a responsive and efficient user experience:

#### 5.1.1 Response Time

- Page load time shall not exceed 2 seconds for initial load and 1 second for subsequent navigation under normal network conditions (4G/WiFi).
- API response time shall be under 300ms for 95% of requests and shall not exceed 1 second for complex data operations.

- Search functionality shall return results within 500ms after input completion.

### 5.1.2 Throughput

- The system shall support at least 1,000 concurrent users during peak hours without performance degradation.
- The system shall process at least 50 transactions per second during normal operation.

### 5.1.3 Capacity

- User database shall accommodate at least 500,000 user profiles with associated workout and nutrition data.
- The system shall store at least 3 years of historical workout and nutrition data per user.
- Exercise library shall support at least 1,000 exercise entries with associated images and videos.

### 5.1.4 Scalability

- System architecture shall allow horizontal scaling to accommodate user growth of up to 20% per month.
- Database performance shall maintain response times with data volume increases of up to 50GB per month.

### 5.1.5 Resource Utilization

- Mobile application shall consume no more than 150MB of device memory during active use.
- Client-side processing shall be optimized to minimize battery consumption on mobile devices.
- Data transfer shall be minimized through efficient caching to reduce bandwidth usage.

## 5.2 Safety Requirements

While FitLife is a software application with limited physical safety risks, the following safety requirements shall be implemented:

### 5.2.1 Exercise Guidance

- Exercise tutorials shall include proper form guidance and safety warnings for all activities.
- Beginner workout plans shall include appropriate warm-up and cool-down activities.
- Exercise recommendations shall consider user-reported health conditions and limitations.

### 5.2.2 Health Advisories

- System shall display appropriate health disclaimers before users begin any workout program.
- Users shall be prompted to consult healthcare professionals before starting new fitness regimens, especially for users who indicate pre-existing conditions.

- Warning messages shall be displayed when users input potentially unsafe workout parameters (excessive weights, durations).

#### 5.2.3 Overexertion Prevention

- System shall monitor for unusual patterns that may indicate overtraining and provide appropriate notifications.
- Rest day recommendations shall be incorporated into workout scheduling algorithms.
- Progressive overload recommendations shall follow established fitness industry safety guidelines.

## 5.3 Security Requirements

FitLife shall implement comprehensive security measures to protect user data and system integrity:

#### 5.3.1 Authentication and Authorization

User authentication shall use industry-standard OAuth 2.0 protocol with additional two-factor authentication option.

- Password requirements shall enforce minimum length of 8 characters, with complexity requirements (uppercase, lowercase, numbers, special characters).
- Session tokens shall expire after 24 hours of inactivity.
- Role-based access control shall restrict user access to appropriate functionality and data.

#### 5.3.2 Data Protection

- All personal user data shall be encrypted at rest using AES-256 encryption.
- All data transmissions shall use TLS 1.2 or higher encryption.
- Health-related data shall be stored in compliance with relevant regulations (HIPAA principles, GDPR).
- User data shall be logically separated to prevent unauthorized cross-user access.

#### 5.3.3 Privacy Controls

- Users shall have granular control over what personal information is shared with trainers and community.
- Data collection and usage policies shall be clearly communicated during onboarding.
- Users shall have the ability to download or delete their personal data in accordance with data protection regulations.
- Third-party data sharing shall be explicitly opt-in only.

#### 5.3.4 System Security

- Regular security audits and penetration testing shall be conducted quarterly.
- Web application shall be protected against common vulnerabilities (OWASP Top 10).

- API endpoints shall implement rate limiting to prevent abuse.
- Input validation shall be performed server-side to prevent injection attacks.

## 5.4 Software Quality Attributes

The FitLife application shall exhibit the following quality attributes:

### 5.4.1 Usability

- The user interface shall follow intuitive design patterns requiring no specialized training.
- Common tasks shall be completable in 3 or fewer steps.
- System shall maintain consistent navigation and interaction patterns throughout.
- Help documentation shall be context-sensitive and easily accessible.
- User interface shall accommodate users with varying technical expertise.

### 5.4.2 Reliability

- System uptime shall exceed 99.9% (excluding scheduled maintenance).
- Mean time between failures (MTBF) shall exceed 720 hours.
- Data backup procedures shall ensure no more than 1 hour of data loss in case of system failure.
- Automatic recovery mechanisms shall restore service after failure without manual intervention.

### 5.4.3 Maintainability

- Code shall follow documented coding standards and design patterns.
- System architecture shall be modular to allow component-level updates.
- Comprehensive test coverage (minimum 80%) shall be maintained for all code.
- System documentation shall be updated with each release.
- Dependency management shall identify and resolve security vulnerabilities.

### 5.4.4 Portability

- Web application shall function consistently across major browsers (Chrome, Firefox, Safari, Edge).
- Mobile interface shall support both iOS (13+) and Android (10+) platforms.
- System architecture shall support migration between cloud service providers if necessary.

### 5.4.5 Interoperability

- System shall implement standard RESTful APIs for potential future integrations.
- Data export functionality shall support common formats (CSV, JSON).
- Authentication system shall support integration with enterprise SSO systems.

## 5.5 Business Rules

The following business rules shall govern the operation of the FitLife platform:

### 5.5.1 User Account Management

- Basic user accounts shall be created free of charge.
- Premium features shall require subscription payment.
- Account inactivity exceeding 12 months may result in account archival with prior notification.
- Users under 18 years of age shall require parental consent during registration.

### 5.5.2 Trainer Verification

- Fitness trainers must provide certification documentation before being listed in the trainer directory.
- Trainer certifications must be from recognized fitness institutions.
- Trainers must maintain a minimum user satisfaction rating to remain active on the platform.

### 5.5.3 Content Moderation

- User-generated content (profile pictures, comments) shall be subject to community guidelines.
- Inappropriate content shall be removed after review by moderators.
- Repeated guideline violations may result in account restrictions.

### 5.5.4 Payment Processing

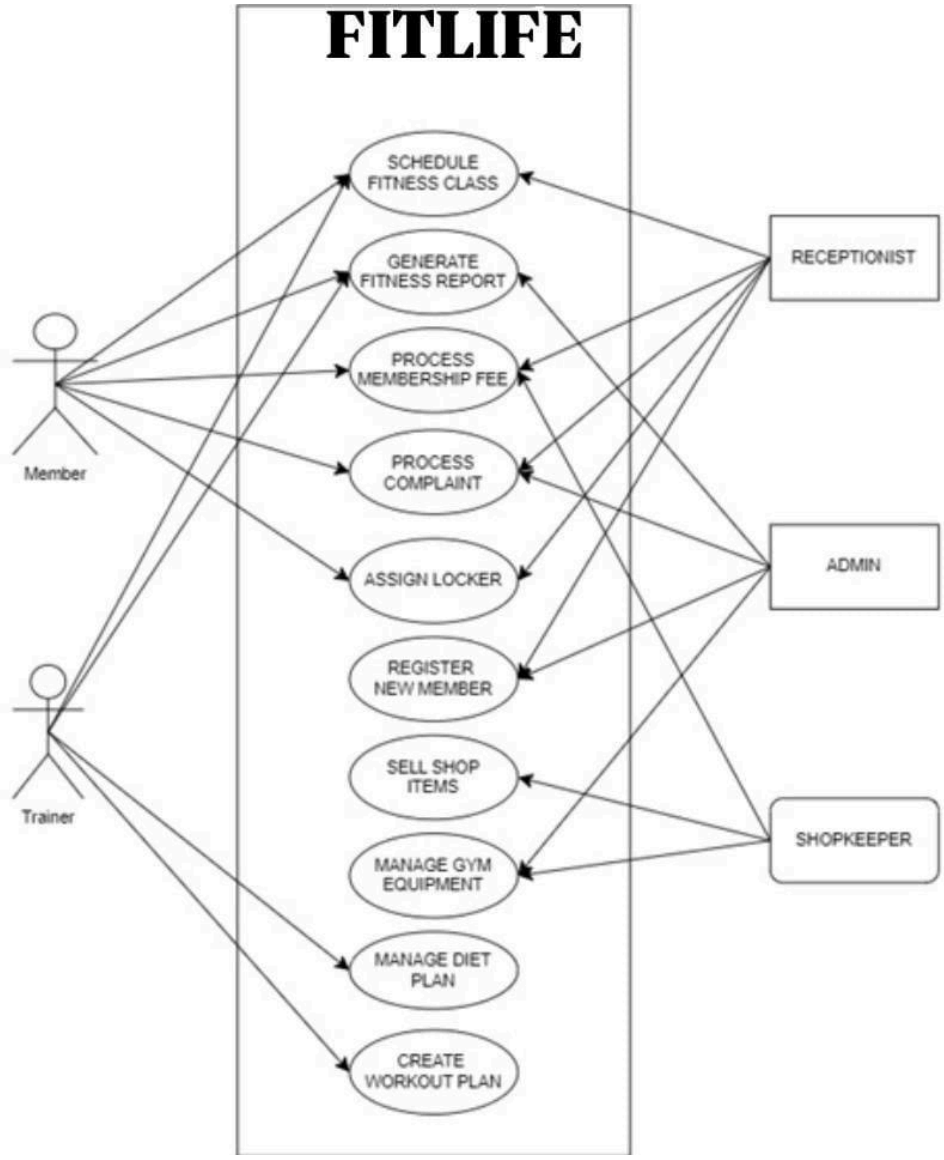
- All financial transactions shall be processed through PCI-DSS compliant payment processors.
- Subscription billing shall occur on a monthly or annual basis.
- Users shall receive notification prior to automatic subscription renewal.
- Refund requests shall be processed according to the published refund policy.

### 5.5.5 Data Retention

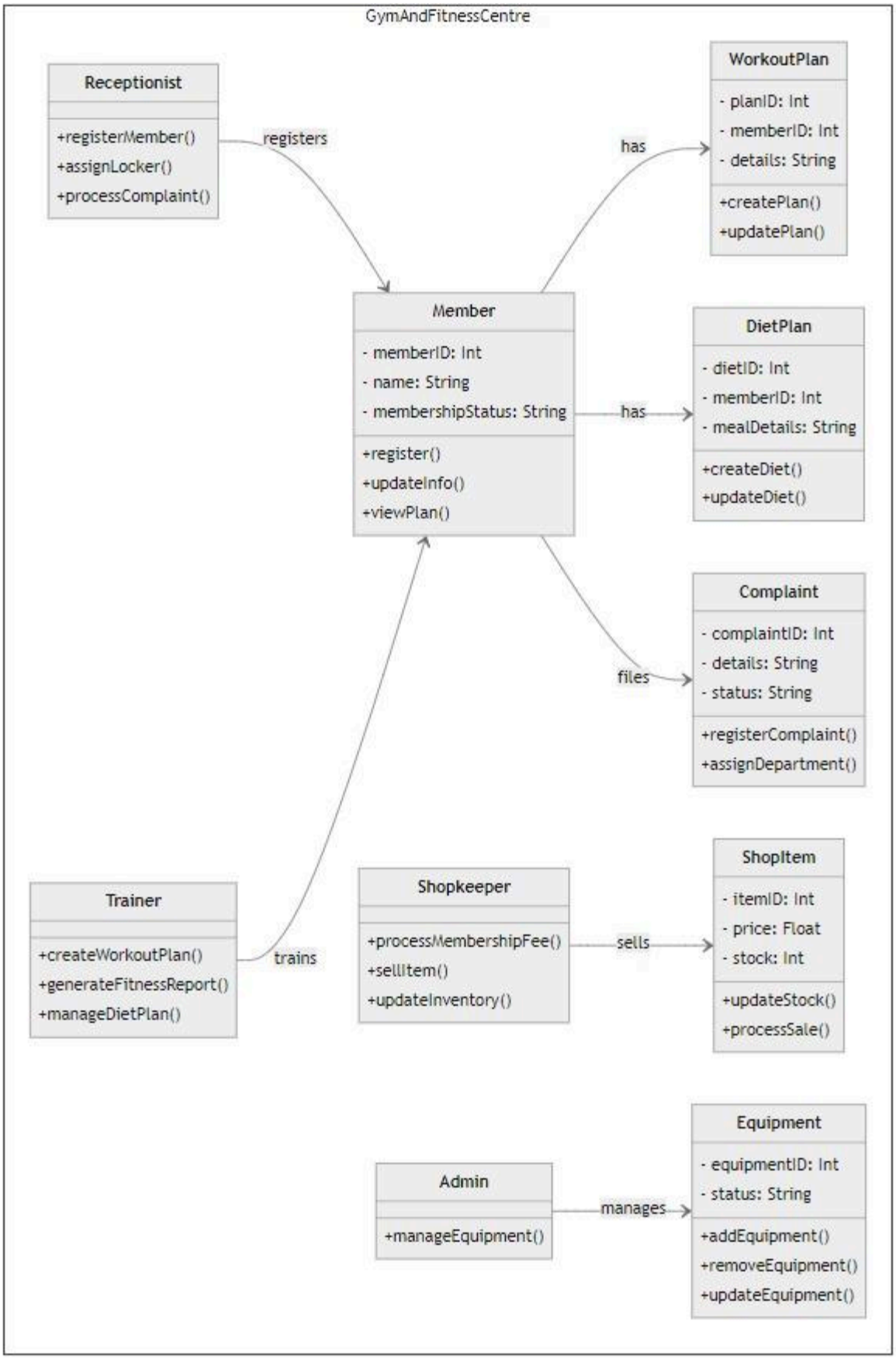
- User workout and nutrition history shall be retained for the duration of account activity plus 12 months.
- Anonymized fitness data may be used for platform improvement and research purposes.
- Personal identification information shall be removed from accounts closed more than 30 days.

# 6. Diagram

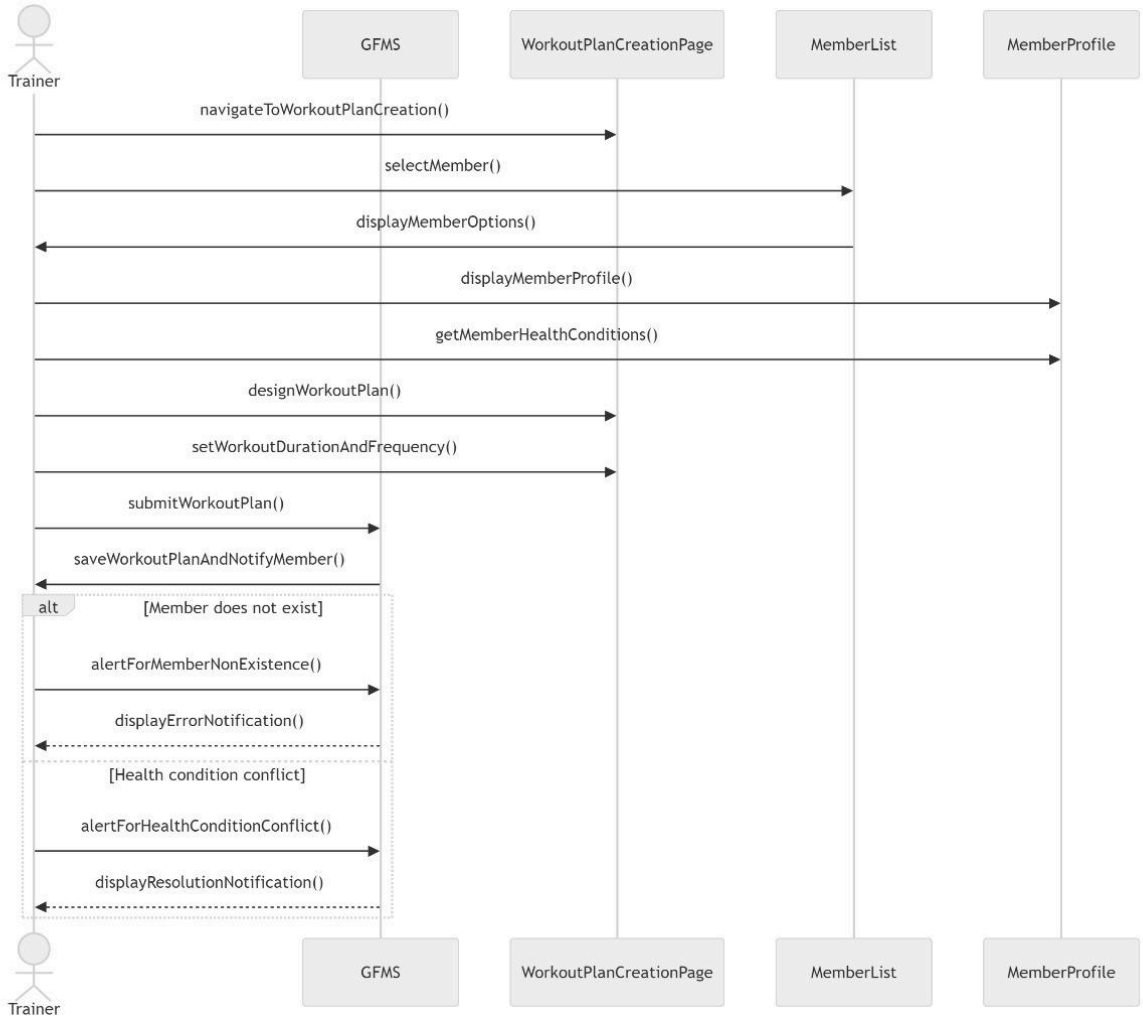
## 6.1 Use Case Diagram

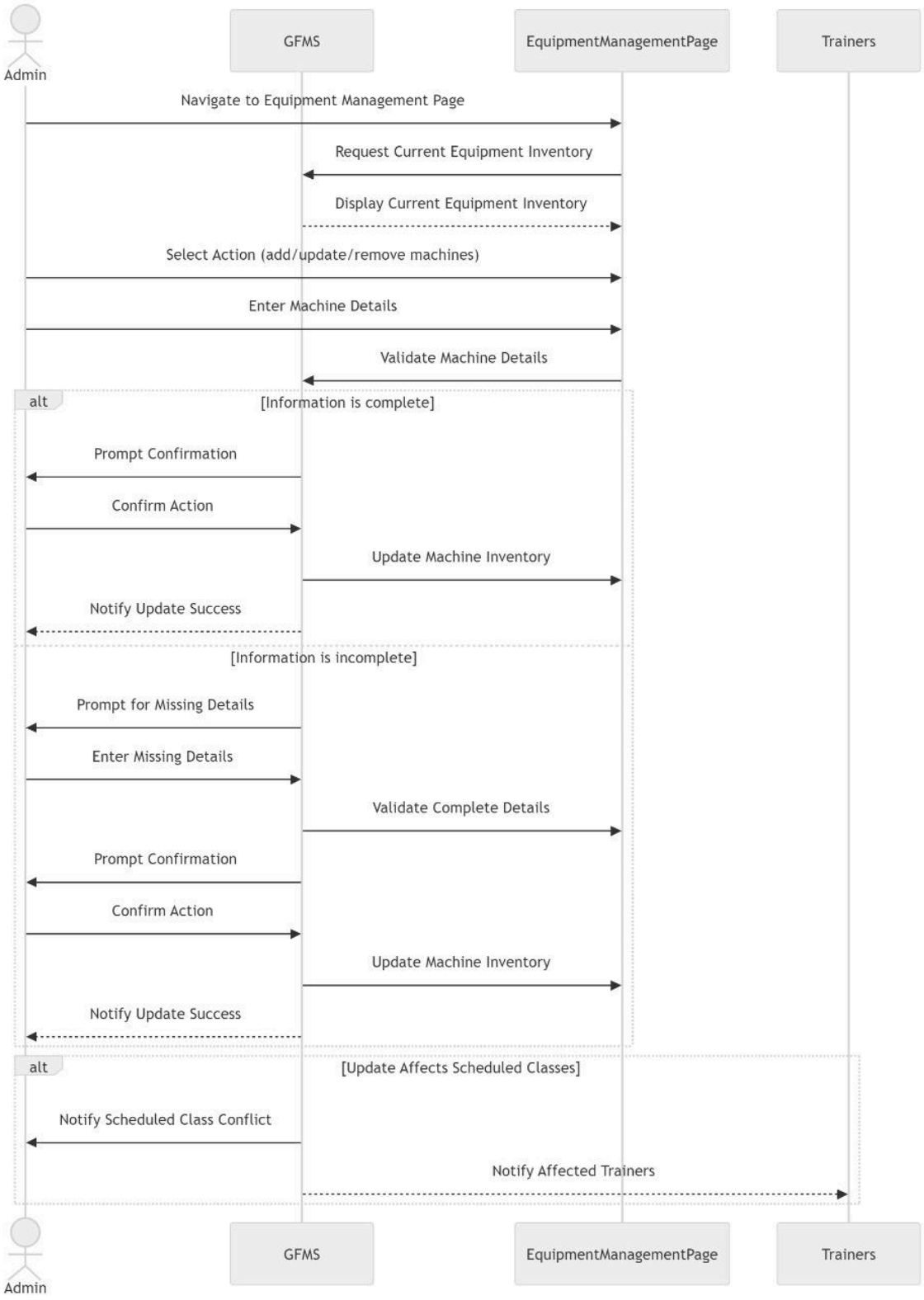



## 6.2 Class Diagram

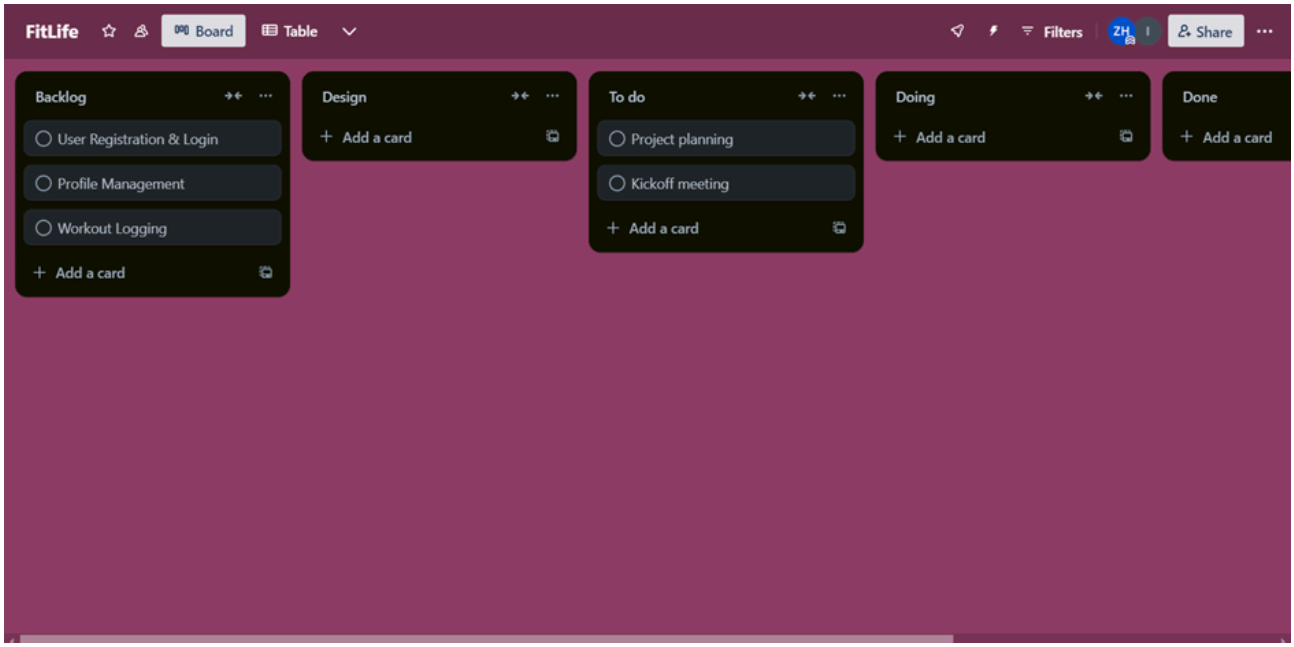

GymAndFitnessCentre
Receptionist
+registerMember()
+assignLocker()
+processComplaint()
registers
WorkoutPlan
- planID: Int
- memberID: Int
- details: String
+createPlan()
+updatePlan()
has
Member
- memberID: Int
- name: String
- membershipStatus: String
+register()
+updateInfo()
+viewPlan()
has
DietPlan
- dietID: Int
- memberID: Int
- mealDetails: String
+createDiet()
+updateDiet()
files
Complaint
- complaintID: Int
- details: String
- status: String
+registerComplaint()
+assignDepartment()
Trainer
+createWorkoutPlan()
+generateFitnessReport()
+manageDietPlan()
trains
Shopkeeper
+processMembershipFee()
+sellItem()
+updateInventory()
sells
ShopItem
- itemID: Int
- price: Float
- stock: Int
+updateStock()
+processSale()
Admin
+manageEquipment()
manages
Equipment
- equipmentID: Int
- status: String
+addEquipment()
+removeEquipment()
+updateEquipment()

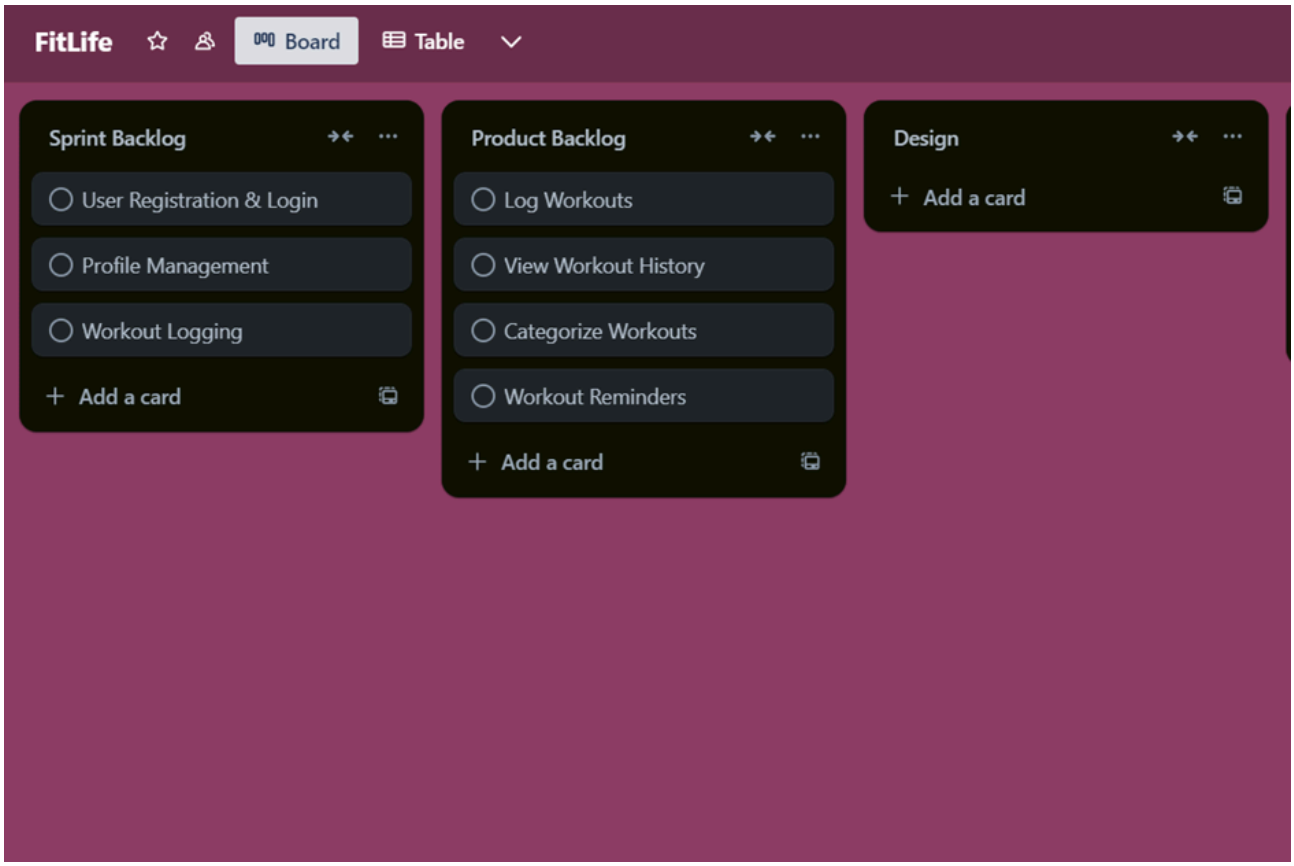
# 6.3 Sequence Diagram




Admin
GFMS
EquipmentManagementPage
Trainers
Navigate to Equipment Management Page
Request Current Equipment Inventory
Display Current Equipment Inventory
Select Action (add/update/remove machines)
Enter Machine Details
Validate Machine Details
alt
[Information is complete]
Prompt Confirmation
Confirm Action
Update Machine Inventory
Notify Update Success
[Information is incomplete]
Prompt for Missing Details
Enter Missing Details
Validate Complete Details
Prompt Confirmation
Confirm Action
Update Machine Inventory
Notify Update Success
alt
[Update Affects Scheduled Classes]
Notify Scheduled Class Conflict
Notify Affected Trainers
Admin
GFMS
EquipmentManagementPage
Trainers

# 6.4 Trello Screenshots



# 6.5 GitHub Screenshots