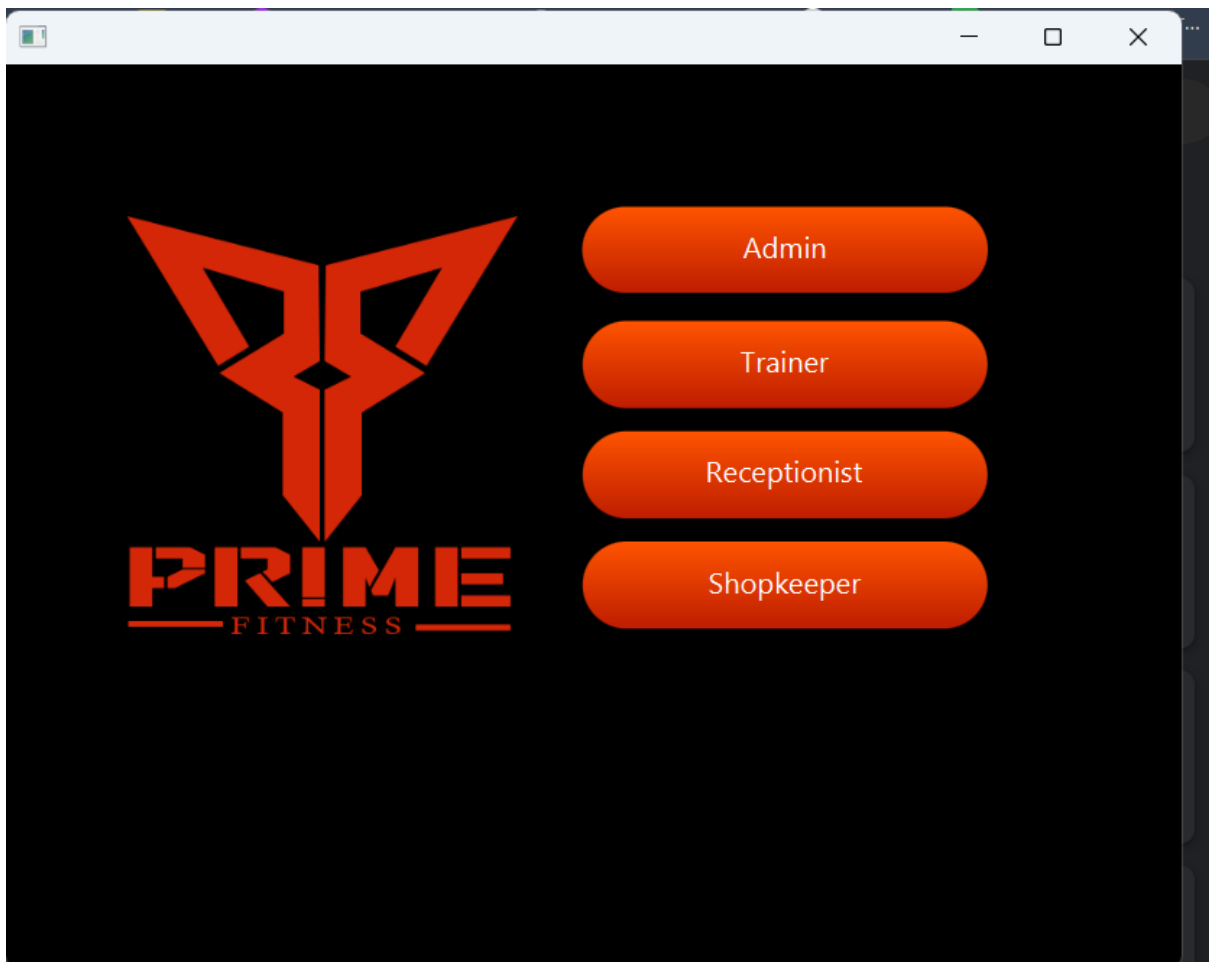


---

# **PRIME FITNESS**

Prepared by  
Zohaib Hassan 22i0800  
Fahad Faisal 22i0817



# Contents

**Table of Contents** .....Error! Bookmark not defined.

**Introduction**..... 3

**1. Functional Requirements** ..... 3

1.1 User Registration and Profile Management .....3

1.2 Workout Logging and Tracking.....5

1.3 Calorie and Nutrition Tracking .....6

1.4 AI-Based Personalized Recommendations .....7

1.5 Trainer and Coach Connection .....7

1.6 Progress Analytics and Reporting .....8

1.7 Community and Challenges .....9

1.8 Reminders and Notifications .....9

**2. Nonfunctional Requirements**..... 10

2.1 Performance Requirements.....10

2.2 Safety Requirements .....11

2.3 Security Requirements .....12

2.4 Software Quality Attributes.....13

2.5 Business Rules .....14

**Overview** ..... 16

**Tasks**..... 16

**2. User Login (Secure)**..... 16

**3. Create Workout Plan**..... 16

**4. Manage Gym Equipment**..... 17

**5. Assign Locker** ..... 18

**6. Profile Management** ..... 18

**7. Admin Verification**..... 18

**8. Basic User Interface** ..... 18

**9. Service Selection** ..... 19

**10. Database Foundation**..... 19

**Overview** ..... 19

**Tasks**..... 19

**2. Diet Plan Management**..... 19

**3. Progress Analytics** ..... 19

**4. Exercise Library** ..... 20

**5. Community & Challenges**..... 20

**6. Notifications System** ..... 21

**7. Fitness Goal Setting**..... 21

**8. Complaint Management** ..... 21

<b>9. Basic Social Features.....</b>	<b>21</b>
<b>10. Mobile Responsiveness.....</b>	<b>21</b>
<b>3. Diagram.....</b>	<b>37</b>
3.1 Use Case Diagram.....	37
3.3 Sequence Diagram .....	39

## Introduction

Prime Fitness 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 Prime Fitness 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

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

## 1. Functional Requirements

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

### 1.1 User Registration and Profile Management

#### 1.1.1 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)

#### 1.1.2 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

### 1.1.3 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.

## 1.2 Workout Logging and Tracking

### 1.2.1 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)

### 1.2.2 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

### 1.2.3 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.

## 1.3 Calorie and Nutrition Tracking

### 1.3.1 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)

### 1.3.2 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

### 1.3.3 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.

## 1.4 AI-Based Personalized Recommendations

### 1.4.1 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)

### 1.4.2 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

### 1.4.3 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.

## 1.5 Trainer and Coach Connection

### 1.5.1 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)

### 1.5.2 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

### 1.5.3 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.

## 1.6 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.

## 1.7 Community and Challenges

### 1.7.1 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)

### 1.7.2 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

### 1.7.3 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.

## 1.8 Reminders and Notifications

### 1.8.1 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)

### 1.8.2 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

### 1.8.3 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.

.

## 2. Nonfunctional Requirements

### 2.1 Performance Requirements

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

#### 2.1.1 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.

### **2.1.2 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.

### **2.1.3 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.

### **2.1.4 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.

### **2.1.5 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.

## **2.2 Safety Requirements**

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

### **2.2.1 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.

### **2.2.2 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).

### **2.2.3 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.

## **2.3 Security Requirements**

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

### **2.3.1 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.

### **2.3.2 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.

### **2.3.3 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.

### **2.3.4 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.

## **2.4 Software Quality Attributes**

The Prime Fitness application shall exhibit the following quality attributes:

### **2.4.1 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.

### **2.4.2 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.

### **2.4.3 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.

### **2.4.4 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.

### **2.4.5 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.

## **2.5 Business Rules**

The following business rules shall govern the operation of the Prime Fitness platform:

### **2.5.1 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.

### **2.5.2 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.

### **2.5.3 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.

### **2.5.4 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.

### **2.5.5 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.

# User Stories

## User Registration & Profile

1. As a new user, I want to create an account so that I can access my fitness dashboard.
2. As a user, I want to log in using Google/Facebook so that I can sign in quickly.
3. As a user, I want to edit my profile so that I can update my fitness information.
4. As a user, I want to input my weight, height, and fitness goals so that I get personalized recommendations.

## Workout Tracking

5. As a user, I want to log my workouts so that I can track my progress.
6. As a user, I want to view my previous workout history so that I can analyze my performance.
7. As a user, I want to categorize my workouts (e.g., cardio, strength) so that I can organize my training.
8. As a user, I want to get reminders for my scheduled workouts so that I stay consistent.

## Calorie Tracking & Diet Management

9. As a user, I want to log my meals so that I can track my calorie intake.
10. As a user, I want the system to suggest diet plans based on my fitness goals so that I eat healthy.

## AI-Based Recommendations

11. As a user, I want to receive AI-generated workout plans so that I get the best exercises for my goals.
12. As a beginner, I want the system to suggest easy workouts so that I don't get overwhelmed.

## Trainer/Coach Interaction

13. As a user, I want to chat with a fitness trainer so that I can get professional guidance.
14. As a user, I want to book online coaching sessions so that I can improve my training.

## Progress & Analytics

15. As a user, I want to see my weight progress chart so that I can track my improvements.
16. As a user, I want to get weekly progress reports so that I stay motivated.

## Community & Challenges

17. As a user, I want to join fitness challenges so that I stay motivated.
18. As a user, I want to compete with friends so that I can stay accountable.

## Social Features

19. As a user, I want to share my fitness achievements on social media so that I can inspire others.

20. As a user, I want to follow other users so that I can stay connected with my fitness community.

### **Reminders & Notifications**

21. As a user, I want to receive reminders for workouts and meals so that I don't forget them.

### **Security & Settings**

22. As a user, I want to change my password so that I can keep my account secure.

23. As an admin, I want to manage user accounts so that I can remove inactive users.

## **Sprint 1 Backlog**

### **Overview**

Sprint 1 focuses on implementing the core user management and basic fitness tracking functionalities required for the Prime Fitness application. The goal is to establish a foundation for user authentication, member registration, workout planning, and primary gym operations.

### **Tasks**

#### **1. Register New Member**

- Implement member registration page
- Create personal details collection form
- Build information validation system
- Implement membership plan selection
- Develop account creation and member ID generation
- Create welcome email functionality

#### **2. User Login (Secure)**

- Implement authentication system with password hashing
- Enable session management
- Create secure login interface
- Build password reset functionality
- Implement role-based access control

#### **3. Create Workout Plan**

- Implement workout plan creation interface
- Create member selection functionality
- Build fitness profile display
- Develop workout design tools



- Implement plan submission and notification system

#### **4. Manage Gym Equipment**

- Create equipment ineVENTORY management interface

- Implement equipment addition, update, and removal functionality
- Build information validation system
- Develop inventory update mechanism
- Create maintenance tracking

## **5. Assign Locker**

- Implement locker assignment interface
- Create available locker display
- Build locker selection functionality
- Develop locker status updating
- Implement member notification system

## **6. Profile Management**

- Create profile editing interface
- Implement fitness metrics tracking
- Build profile picture upload functionality
- Develop trainer view for member profiles
- Create fitness goal management

## **7. Admin Verification**

- Implement admin dashboard
- Create trainer verification system
- Build status field management
- Develop notification system for verification status
- Implement admin user management

## **8. Basic User Interface**

- Create responsive dashboard layout
- Implement navigation system
- Build user interface components
- Develop mobile-friendly design
- Create consistent styling across the application

## **9. Service Selection**

- Implement service category display
- Create service details view
- Build service selection functionality
- Develop service scheduling options
- Implement trainer specialization filtering

## **10. Database Foundation**

- Design and implement user data schema
- Create equipment inventory database
- Build workout plans storage
- Develop locker management data structure
- Implement secure data access patterns

# **Sprint 2 Backlog**

## **Overview**

Sprint 2 focuses on expanding the Prime Fitness application by implementing diet management, progress tracking, and community features. Building on the user authentication and workout management foundation established in Sprint 1, this sprint aims to enhance the user experience with nutritional guidance and social engagement functionalities.

## **Tasks**

### **1. Calorie Tracking System**

- Implement daily food intake logging functionality
- Create calorie calculation based on food items
- Develop meal categorization (breakfast, lunch, dinner, snacks)
- Build visual representation of daily calorie consumption

### **2. Diet Plan Management**

- Create diet plan interface for trainers
- Implement nutritional guidelines and meal recommendations
- Build meal scheduling system
- Develop diet plan assignment to members

### **3. Progress Analytics**

- Implement weight progress chart

- Create workout performance metrics
- Build weekly/monthly progress reports
- Develop goal achievement tracking

#### 4. **Exercise Library**

- Create comprehensive exercise database
- Implement categorization by muscle groups and difficulty
- Add exercise descriptions and instructional content
- Develop search and filter functionality

#### 5. **Community & Challenges**

- Implement fitness challenge creation system
- Build challenge participation and tracking
- Create leaderboards for challenges
- Develop achievement badges and rewards

## **6. Notifications System**

- Implement workout reminders
- Create meal schedule notifications
- Build progress milestone alerts
- Develop system for trainer communication notifications

## **7. Fitness Goal Setting**

- Create interface for setting short and long-term goals
- Implement goal tracking mechanisms
- Build goal adjustment functionality
- Develop goal achievement celebration notifications

## **8. Complaint Management**

- Implement complaint submission system
- Create complaint categorization and priority assignment
- Build complaint tracking and resolution workflow
- Develop feedback collection after resolution

## **9. Basic Social Features**

- Implement user following functionality
- Create activity feed for connected users
- Build basic achievement sharing
- Develop privacy controls for shared content

## **10. Mobile Responsiveness**

- Optimize all interfaces for mobile devices
- Implement responsive design for key features
- Build mobile-specific workout logging interface
- Ensure seamless experience across device types

# **Software Project Plan**

## **1.0 Planning Phase**

This phase focuses on establishing the foundation for the Prime Fitness project, spanning the first two weeks of March 2025.

### **1.1 Requirements Gathering**

**Description:** Comprehensive collection and documentation of all functional and non-functional requirements for the Prime Fitness application.

**Deliverables:**

- Complete requirements documentation for all Prime Fitness modules: user account management, workout tracking, nutrition monitoring, progress visualization, professional guidance, community features, and notification system
- Acceptance criteria for each major feature

**Key Activities:**

- Define user stories for all primary functions
- Document technical constraints and performance requirements
- Establish project scope boundaries

### **1.2 Architecture Design**

**Description:** Development of the technical blueprint for the Prime Fitness application, focusing on the client-server architecture.

**Deliverables:**

- System architecture documentation
- API specifications
- Security architecture plan

**Key Activities:**

- Design the client-server communication protocols
- Define database architecture and relationships
- Plan authentication and authorization mechanisms

### **1.3 Database Design**

**Description:** Detailed design of the database structure to support all Prime Fitness application features.

**Deliverables:**

- Entity-relationship diagrams
- Database schema documentation
- Data dictionary
- Query optimization strategy

**Key Activities:**

- Define data entities and relationships
- Design tables for users, workouts, nutrition, progress tracking, etc.

- Plan indexing strategy for performance optimization
- Design data security measures

#### **1.4 UI Design**

**Description:** Creation of the visual design and user experience for all Prime Fitness interfaces.

**Deliverables:**

- Complete UI mockups for all application screens
- Design system with color scheme, typography, and component library
- Interactive prototypes for key user flows
- Responsive design specifications for mobile and desktop

**Key Activities:**

- Create wireframes for all major screens
- Design cohesive visual identity
- Develop interactive prototypes for user testing
- Establish UI component library for consistent implementation

#### **1.5 Development Environment Setup**

**Description:** Configuration of all development tools and environments needed for the project.

**Deliverables:**

- Configured development environments for all team members
- Version control repository setup
- Continuous integration/deployment pipeline
- Development standards documentation

**Key Activities:**

- Set up development, testing, and staging environments
- Configure version control system with branching strategy
- Establish code review process
- Set up automated build and testing pipelines

### **2.0 Backend Development**

This phase covers the development of all server-side components of the Prime Fitness application.

#### **2.1 Authentication System**

**Description:** Implementation of secure user authentication including third-party integration.

**Deliverables:**

- User registration and authentication API
- Third-party authentication integration (Google/Facebook)
- Password management functionality
- Session handling system

**Key Activities:**

- Implement secure password storage using industry best practices
- Develop JWT-based authentication system
- Create account verification workflows
- Integrate OAuth for third-party authentication services

**2.2 Database Setup**

**Description:** Implementation of the database schema and initial data population.

**Deliverables:**

- Implemented database schema
- Database migration scripts
- Initial data seeds for testing
- Database backup and recovery procedures

**Key Activities:**

- Create database tables according to schema design
- Implement database constraints and relationships
- Set up database users and access permissions
- Configure database replication if needed

**2.3 User Management API**

**Description:** Development of APIs for user profile management and personal settings.

**Deliverables:**

- User profile CRUD endpoints
- User preferences API
- Profile image handling
- Personal fitness goal management API

**Key Activities:**

- Implement user profile creation and update functionality
- Develop API for managing user preferences
- Create endpoints for fitness goal setting and tracking
- Implement secure data access controls

**2.4 Workout Management API**

**Description:** Implementation of APIs for tracking and managing workout activities.

**Deliverables:**

- Workout logging endpoints
- Workout history and tracking API

**Key Activities:**



- Implement workout logging and tracking functionality
- Create workout history retrieval endpoints

## **2.8 Notification System**

**Description:** Implementation of the system for delivering notifications to users.

**Deliverables:**

- Push notification service
- Email notification system
- In-app notification center
- Notification preference management API

**Key Activities:**

- Implement real-time notification delivery system
- Develop email notification templates and delivery
- Create in-app notification storage and display system
- Build user notification preferences management

## **2.9 Backend Testing**

**Description:** Comprehensive testing of all backend components and APIs.

**Deliverables:**

- Performance test results
- Security audit report

**Key Activities:**

- Develop comprehensive unit tests for all backend components
- Create integration tests for API workflows

## **3.0 Frontend Development**

This phase covers the development of all client-side components of the Prime Fitness application.

### **3.1 Login UI**

**Description:** Implementation of the user authentication interfaces.

**Deliverables:**

- Login screen implementation
- Registration interface
- Password reset UI
- Social login integration

**Key Activities:**

- Create responsive login and registration screens
- Implement client-side validation for user inputs
- Develop password reset workflow interface

- Integrate social login buttons and authentication flows

### **3.2 Profile UI**

**Description:** Development of user profile management interfaces.

**Deliverables:**

- User profile view and edit screens
- User settings interface
- Goal setting UI
- Profile image management

**Key Activities:**

- Implement profile information display and editing
- Create user settings management screens
- Develop fitness goal setting interface
- Build profile image upload and management functionality

### **3.3 Dashboard UI**

**Description:** Implementation of the main application dashboard.

**Deliverables:**

- Main dashboard layout
- Activity summary widgets
- Progress snapshot components
- Notification display area

**Key Activities:**

- Create responsive dashboard layout
- Implement activity summary visualization components
- Develop progress tracking widgets
- Build notification display component

### **3.4 Workout UI**

**Description:** Development of interfaces for workout management.

**Deliverables:**

- Workout history visualization
- Workout recommendation UI

**Key Activities:**

- Develop workout history visualization components
- Build recommendation display and selection interface

### **3.5 Diet UI**

**Description:** Implementation of interfaces for nutrition tracking.

**Deliverables:**

- Food logging interface
- Meal planning screens

**Key Activities:**

- Create food search and logging functionality
- Implement meal planning and scheduling interface

**3.8 Frontend Testing**

**Description:** Comprehensive testing of all frontend components and interfaces.

**Deliverables:**

- Unit test suite for frontend components
- End-to-end test suite for user workflows
- Cross-browser compatibility report
- Responsive design verification report

**Key Activities:**

- Develop component unit tests
- Create end-to-end tests for critical user journeys
- Test application across multiple browsers and devices
- Verify responsive design at various screen sizes

**4.0 Testing Phase**

This phase focuses on integration testing and quality assurance for the entire application.

**4.1 Integration Testing**

**Description:** Verification of frontend and backend component integration.

**Deliverables:**

- Integration test plan
- API connection verification report
- End-to-end workflow test results
- Integration issue log

**Key Activities:**

- Test frontend-backend API connections
- Verify data flow across system components
- Identify and document integration issues
- Coordinate with development teams for issue resolution
- 

**4.2 System Testing**

**Description:** Comprehensive testing of the full application system.

**Deliverables:**

- System test plan
- Functional test results
- Performance test report
- Security assessment

**Key Activities:**

- Execute comprehensive functional testing
- Perform load and stress testing
- Conduct security and vulnerability assessment
- Test data integrity across the system
- 

**4.4 User Acceptance Testing**

**Description:** Validation of the application with representative users.

**Deliverables:**

- User feedback documentation
- Issue prioritization list
- Final acceptance report

**Key Activities:**

- Collect and analyze user feedback
- Identify critical issues requiring resolution before launch
- Obtain stakeholder sign-off on acceptance criteria

**5.0 Deployment Phase**

This phase covers the deployment of the application to production.

**5.1 Staging Environment Setup**

**Description:** Preparation of the pre-production environment for final testing.

**Deliverables:**

- Configured staging environment
- Deployed application to staging
- Staging test results
- Pre-production readiness report

**Key Activities:**

- Configure staging servers to match production specifications
- Deploy the full application stack to staging environment
- Conduct comprehensive testing in staging
- Verify all integrations and dependencies

## 5.2 Production Setup

**Description:** Preparation of the production environment for application launch.

**Deliverables:**

- Configured production servers
- Load balancing implementation
- Security measures deployment
- Monitoring system setup

**Key Activities:**

- Set up production server infrastructure
- Configure load balancing for scalability
- Implement security measures and hardening
- Set up monitoring and alerting systems

## 5.3 Data Migration

**Description:** Transfer of any necessary data to the production environment.

**Deliverables:**

- Data migration plan execution
- Data validation report
- Migration issue resolution documentation
- Production data readiness report

**Key Activities:**

- Execute data migration scripts
- Validate migrated data integrity
- Resolve any data inconsistencies or issues
- Perform final data readiness checks

## 5.4 Launch

**Description:** Final deployment and public release of the Prime Fitness application.

**Deliverables:**

- Launch plan execution
- Production deployment
- Launch monitoring report
- Post-launch issue resolution log

**Key Activities:**

- Execute production deployment according to launch plan
- Implement DNS changes and public access
- Monitor system performance during initial usage

# SYSTEM ARCHITECTURE

## **Package Diagram**

### **1. Views Package**

- Contains UI components: Admin, Trainer, and Login
- This package handles the user interface that clients interact with

### **2. Application Package**

- Contains Main and Controller components
- This is likely the core application logic that coordinates between views and data

### **3. Models Package**

- Contains data models: Customer, Trainer, Receptionist, Shopkeeper, Complaint, Lockers, and Machines
- These represent the business entities in your fitness tracking system

### **4. DB Package**

- Contains DBHandler component
- Handles database operations and persistence

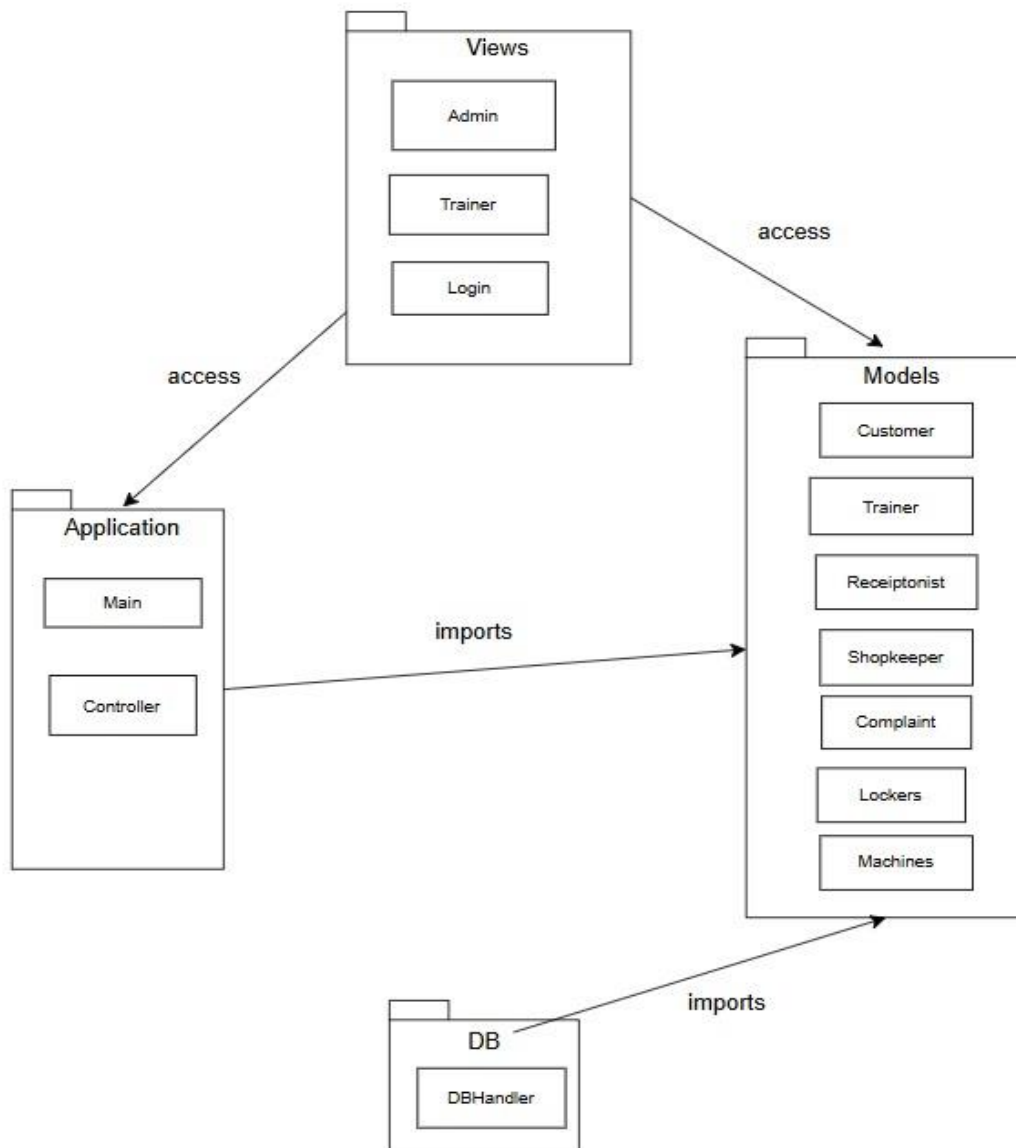
The relationships between these packages are:

- Views package has "access" to the Application package
- Views package has "access" to the Models package
- Application package "imports" the Models package
- DB package "imports" the Models package

This structure aligns well with your Prime Fitness application requirements. The separation of concerns is appropriate for a web-based fitness tracking system:

- The Views package supports your user interfaces for different user types (admin, trainer, and general login)
- The Models package contains the entities needed for your workout management, nutrition tracking, and user accounts
- The Application package provides the controller logic to implement your product functions
- The DB package handles data persistence for tracking progress, workout history, and user information

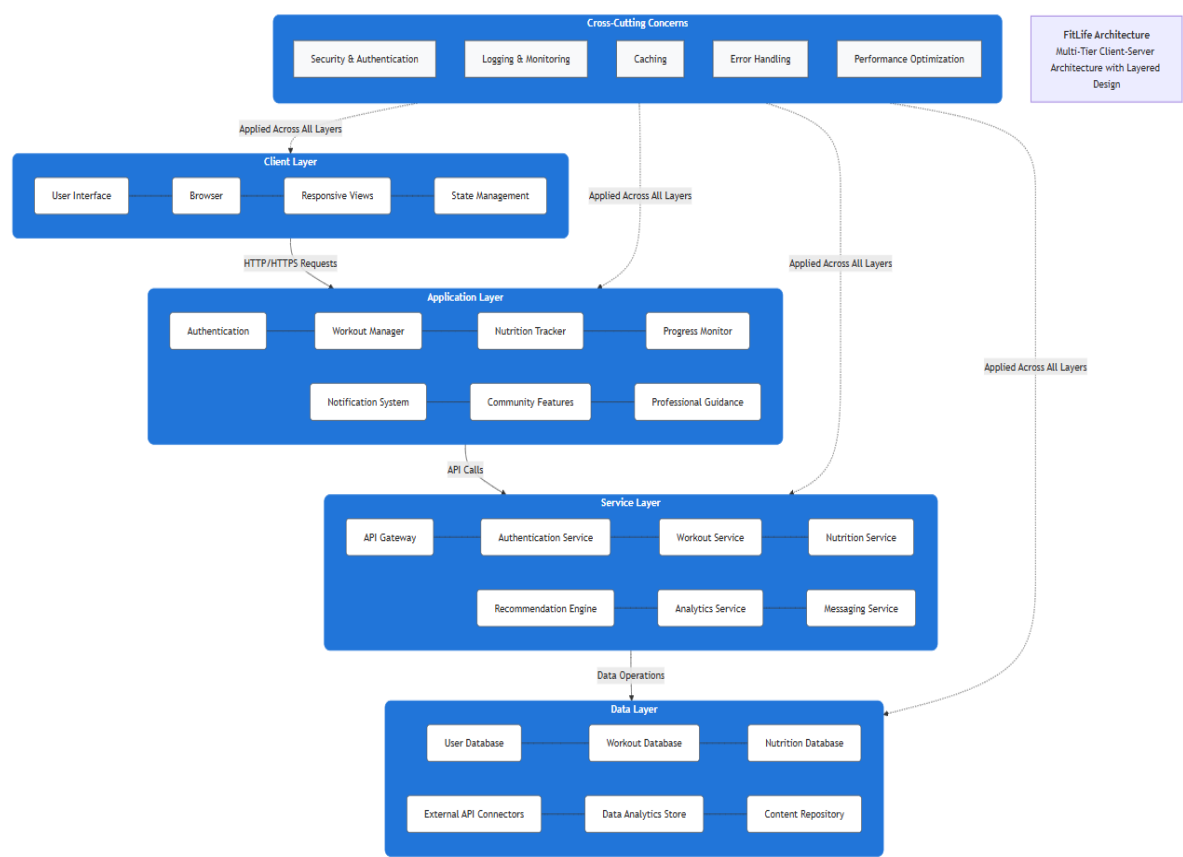
This architecture should facilitate the implementation of key features like user account management, workout tracking, nutrition logging, progress monitoring, professional guidance, community engagement, and notifications as described in your requirements.



## Architecture Styles

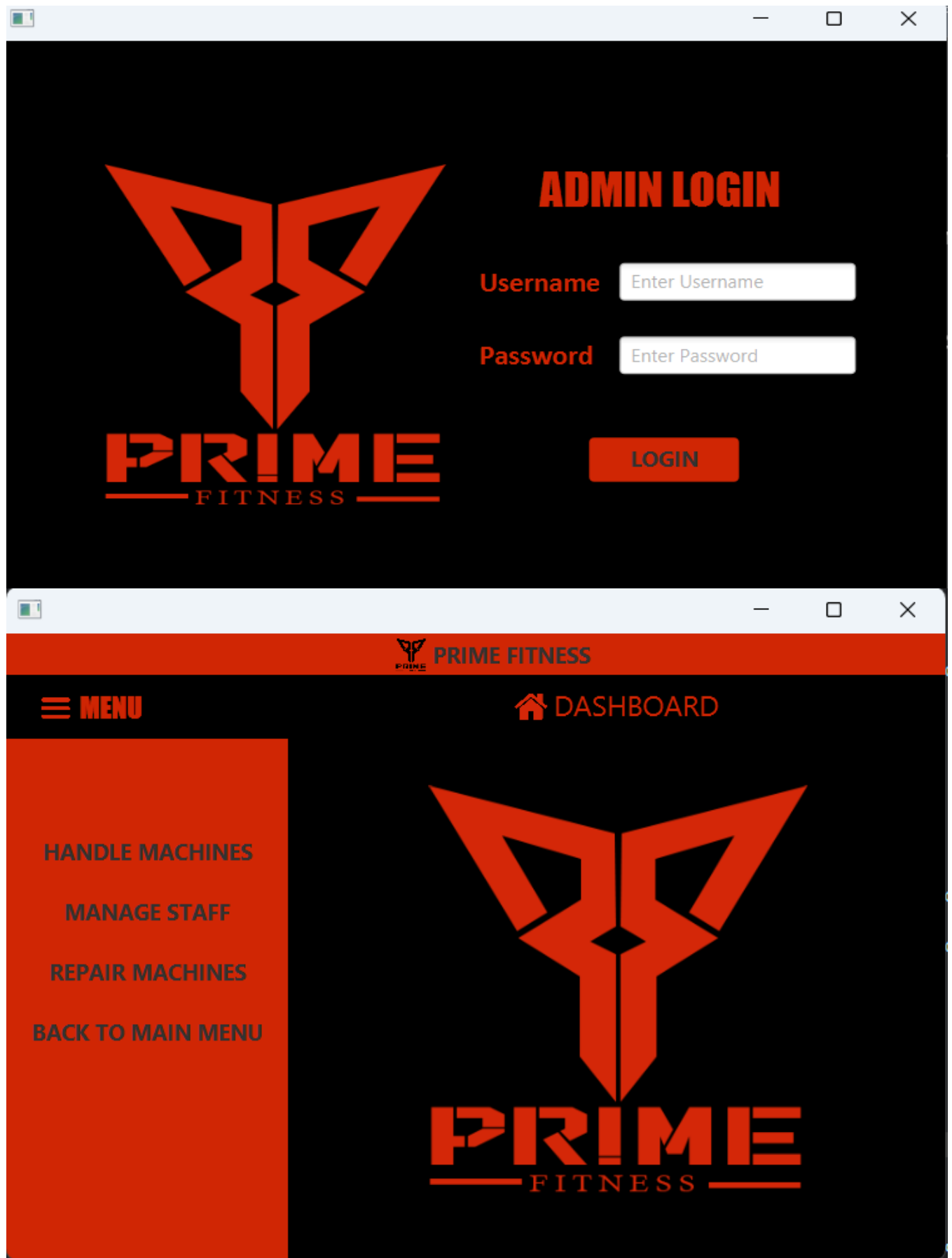
1. **Client Layer:** Handles the user-facing interface, including browser-based UI components, responsive views, and state management.
2. **Application Layer:** Contains the core business logic components like authentication, workout management, nutrition tracking, progress monitoring, notification systems, professional guidance, and community features.
3. **Service Layer:** Provides specialized services that support the application components, including API gateway, authentication, workout services, nutrition services, recommendation engine, analytics, and messaging.
4. **Data Layer:** Manages data storage and access through various databases (user, workout, nutrition) along with external API connectors, analytics store, and content repository.

This architecture separates concerns while allowing components to communicate across layers, enabling scalability, maintainability, and flexibility. The design allows FitLife to handle diverse fitness tracking functions while maintaining a clean separation between presentation, business logic, services, and data management.





## Screenshots



MENU

HANDLE MACHINES

MANAGE STAFF

REPAIR MACHINE

DASHBOARD

STAFF DETAILS

First Name

faisal

Last Name

hassan

CNIC No.

425678593234

DOB

29/04/2025

Salary

50000

Password

123456

Phone No.

03325232864

Type

☒ RECEPTIONIST
 ☐ SHOPKEEPER
 ☐ TRAINER

Comission

ADD

Please fill required field

## Blackbox Testing

Test Case ID	Description	Input	Expected Output
TC-AL-01	Valid admin login	Username: adminPassword: admin123	Redirect to admin dashboard
TC-AL-02	Empty username	Username: Password: admin123	Show "Username required" error
TC-AL-03	Empty password	Username: admin Password:	Show "Password required" error
TC-AL-04	Invalid credentials	Username: adminPassword: wrongpass	Show "Invalid credentials" message
TC-AL-05	SQL injection input	Username: ' OR 1=1 --Password: anything	Show "Invalid credentials" or block request

Test Case ID	Description	Input	Expected Output
TC-AS-01	Valid staff entry	All fields valid	Staff added successfully
TC-AS-02	Missing required fields	Name field empty	Show "Name is required" error
TC-AS-03	Invalid phone number	Phone: 123abc!	Show "Invalid phone number" error
TC-AS-04	Duplicate username/email	Username already exists	Show "Username already exists" message

Test Case ID	Description	Input	Expected Output
TC-US-01	Valid update	Change name or contact details	Show "Updated successfully" message
TC-US-02	Invalid ID	ID does not exist	Show "Staff not found" message

Test Case ID	Description	Input	Expected Output
TC-DS-01	Valid deletion	Existing staff ID	Staff deleted successfully
TC-DS-02	Non-existent ID	ID: 9999	Show "Staff not found" error

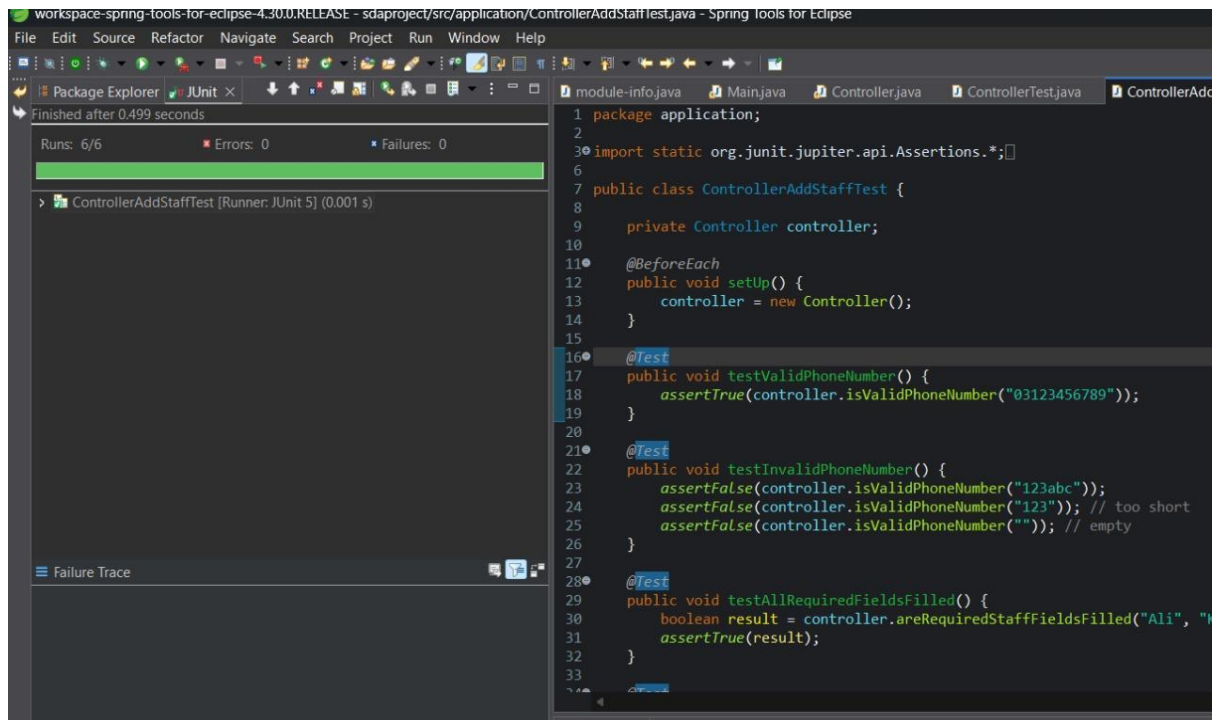
Test Case ID	Description	Input	Expected Output
TC-VS-01	View all staff	None	List of all staff is displayed
TC-VS-02	Filter by role	Role: Trainer	Only trainers displayed

## Whitebox Testing

```

File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer JUnit x
Finished after 0.781 seconds
Runs: 3/3 Errors: 0 Failures: 0
> ControllerTest (Runner: JUnit 5) (0.658 s)
Failure Trace
1 package application;
2
3 import org.junit.jupiter.api.Test;
4 import static org.junit.jupiter.api.Assertions.*;
5
6 public class ControllerTest {
7
8     Controller controller = new Controller();
9
10    // Test valid admin credentials
11
12
13    // Test invalid credentials
14    @Test
15    public void testInvalidAdminLogin() {
16        boolean result = controller.checkAdminCredentials("wronguser", "wrongpass");
17        assertFalse(result, "Invalid credentials should return false");
18    }
19
20    // Test empty username and password
21    @Test
22    public void testEmptyLogin() {
23        boolean result = controller.checkAdminCredentials("", "");
24        assertFalse(result, "Empty username and password should return false");
25    }
26
27    // Test SQL injection input
28    @Test
29    public void testSqlInjectionAttempt() {
30        boolean result = controller.checkAdminCredentials("'" OR '1'='1", "anything");
31        assertFalse(result, "SQL injection-like input should return false");
32    }
33
34

```

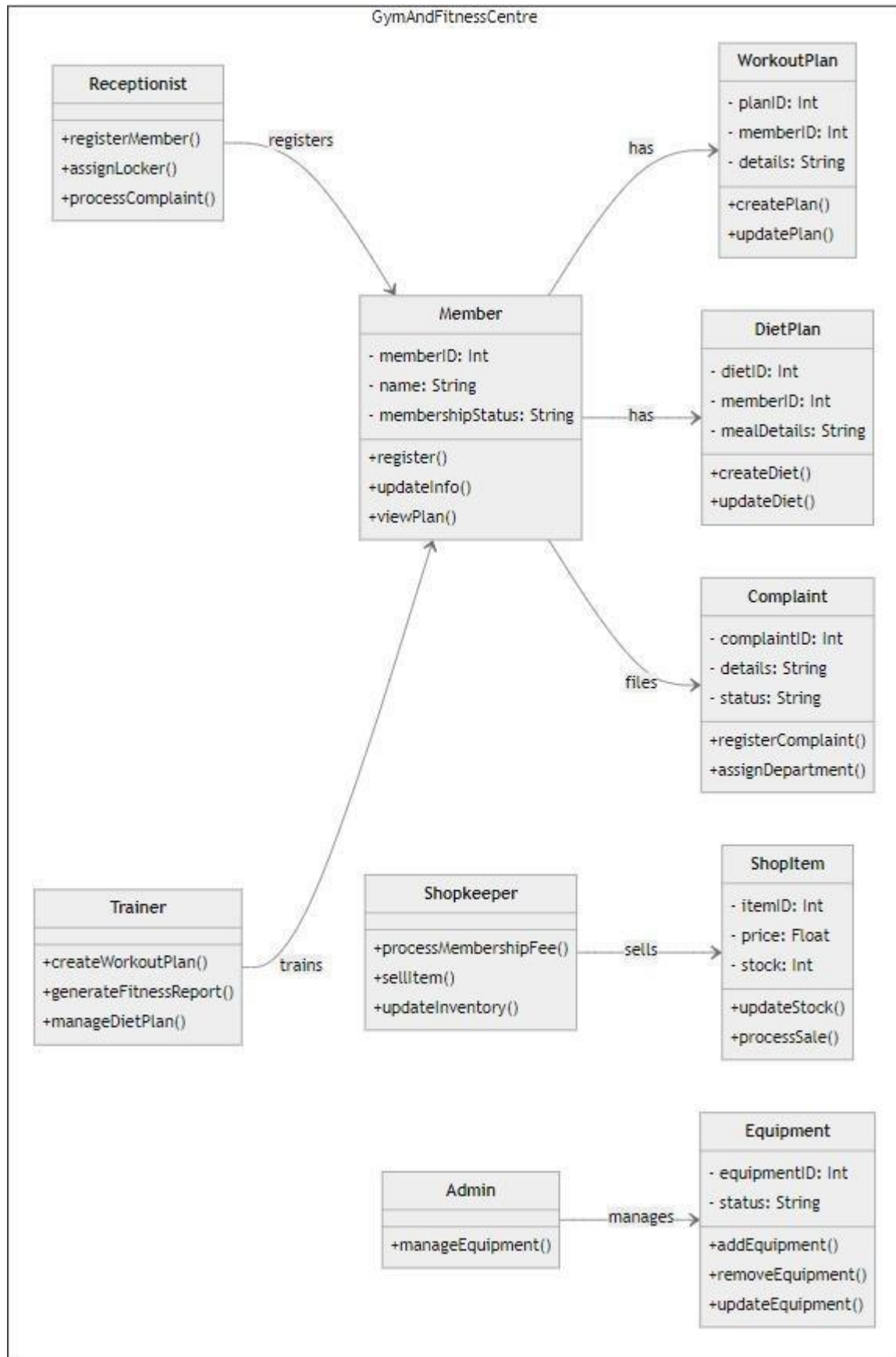


### 3. Diagram

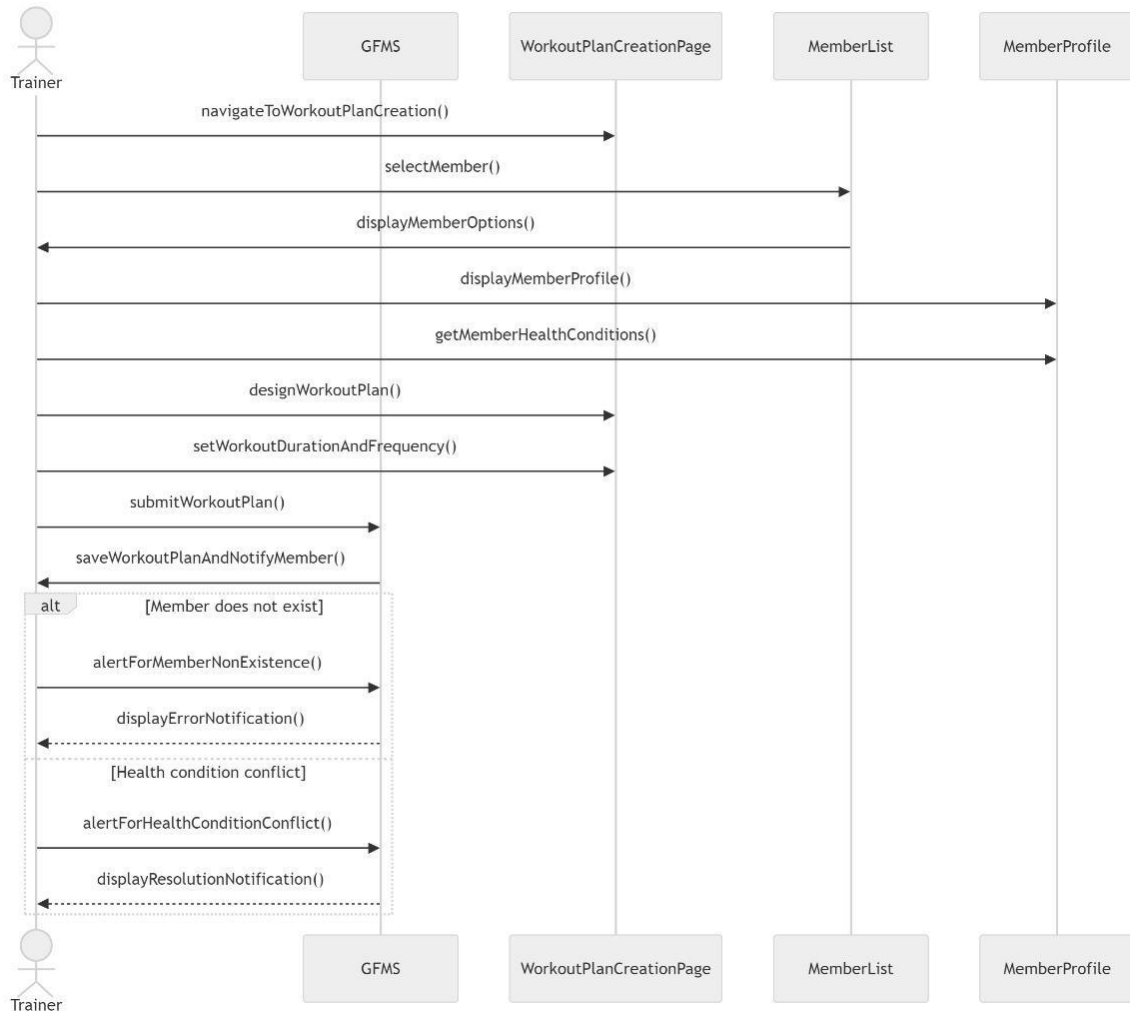
#### 3.1 Use Case Diagram

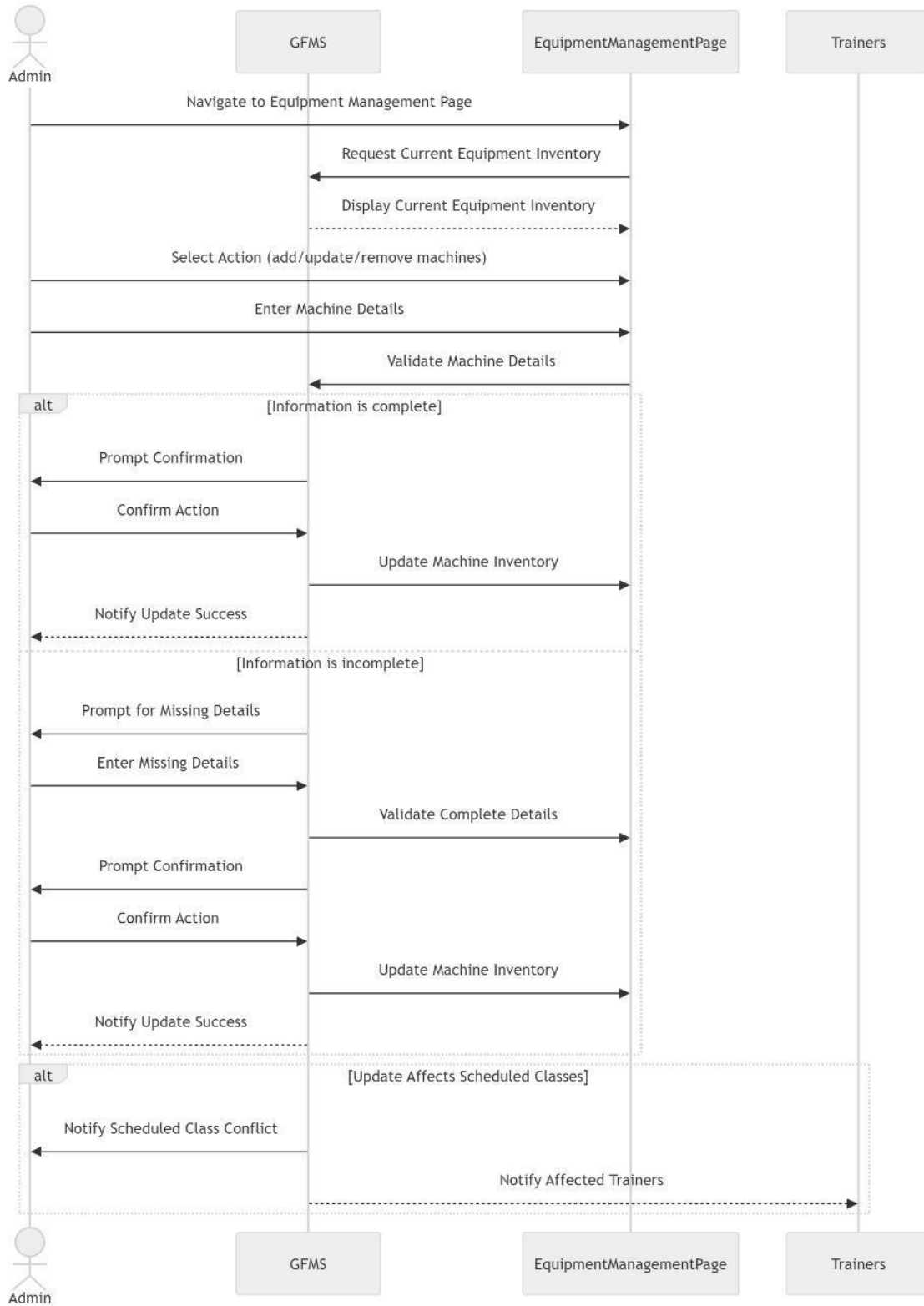


#### 3.2 Class Diagram



### 3.3 Sequence Diagram







### 3.4 Trello Screenshots

