



Your intelligent personal fitness coach

**Team Member:**

Weicheng Zheng    Juekai Lin    Lixin Ma    Junhao Yang  
2154286              2253744              2153085              2154284

**Instructor: Ping Sun**



## **Content**

### **1. Description**

- 1.1. Project Goals
- 1.2. Project Scope
- 1.3. Target Readers and Suggestions
- 1.4. Glossary of Terms

### **2. Introduction**

- 2.1. Project Overview
- 2.2. Progress and Current Status
- 2.3. Changes in the System

### **3. Architecture Analysis**

- 3.1. System-Level Architecture Analysis and Designs
  - 3.1.1. Presentation Layer
  - 3.1.2. Application Layer
  - 3.1.3. External API Layer
  - 3.1.4. Business Interface Layer
  - 3.1.5. Business Implementation Layer
  - 3.1.6. General Service Layer
  - 3.1.7. DTO Component Library
  - 3.1.8. Business Entity Library
  - 3.1.9. Data Access Layer
  - 3.1.10. Common Service Library
- 3.2. Robustness Diagrams
  - 3.2.1. Login & Register System
  - 3.2.2. Fitness Tutorial Section
  - 3.2.3. Fitness Action Coaching Section
  - 3.2.4. Fitness Equipment Section
  - 3.2.5. Fitness Diet Section

### **4. Analysis Model**

- 4.1. Login & Register System
  - 4.1.1. Class Diagram
  - 4.1.2. Interaction Diagram
- 4.2. Fitness Tutorial Section
  - 4.2.1. Class Diagram
  - 4.2.2. Interaction Diagram
- 4.3. Fitness Action Coaching Section
  - 4.3.1. Class Diagram
  - 4.3.2. Interaction Diagram
- 4.4. Fitness Equipment Section
  - 4.4.1. Class Diagram
  - 4.4.2. Interaction Diagram
- 4.5. Fitness Diet Section
  - 4.5.1. Class Diagram
  - 4.5.2. Interaction Diagram

### **5. Update Requirements**

- 5.1 New Case: Fitness equipment product comparison platform
- 5.2 New Case: Dishes preparation tutorial
- 5.3 Updated Case: Login with Wechat
- 5.4 New Case: View the fitness progress and data

### **6. Updated Snapshot**

- 6.1 Fitness equipment product comparison platform
- 6.2 Diet Plan Page update
- 6.3 Login & Registration Interface Snapshot
- 6.4 Fitness Tutorial Section Update Snapshot

### **7. Correction of errors**

## [8. Open Questions](#)

[8.1 Challenges](#)

[8.2 Design Tasks](#)

## [9. The Annotated References](#)

## [10. Contributors](#)

# 1. Description

## 1.1. Project Goals

This project KeepFit aims to streamline the fitness journey with an integrated solution that combines diet plans, exercise routines, and equipment management. By providing personalized diet and workout programs, which dynamically adjust to each user's fitness schedule, it helps users pursue their goals more intelligently and efficiently. The equipment management feature guides users in selecting and optimizing fitness equipment usage. Through smart recommendations and real-time feedback, the project encourages healthy habits, supports the nationwide fitness movement, and fosters the adoption of a healthier lifestyle.

The purpose of this document is to illustrate the KEEP FIT's architectural analysis and analysis model. What's more, system function points are constantly being improved and snapshots of the user interface are constantly being updated. Therefore, we also include updated snapshots, use cases, and annotated references.

## 1.2. Project Scope

Keep-Fit is a Web-based smart fitness software, and the goal is to help users achieve their fitness goals smarter and more effectively. The main scenarios of the system include the selection of fitness tutorial, punching schedule, fitness posture guidance, food equipment recommendation and AI robot chat, etc. VIP members and ordinary users can browse information and participate in the activity after registering and logging in, and choose whether to become uploader. In addition, the platform administrators can maintain the platform order and keep the platform fitness content up to date.

Compared with the previous document, we added new content in each subsystem. The login part can be logged in using WeChat, QQ or TJ account; the progress tracking and data analysis and the tutorial part introduced the food tutorial; the action guidance system updated AI real-time chat; and the fitness equipment system imported into external shopping mall. This will be described in the later documentation.



In addition, Keep-Fit also has the following characteristics:

- The platform system operates in a network environment.
- The system has an accurate recommended fitness tutorial algorithm.
- The intelligent system allows users to chat and consult with the AI.
- Have a safe and reliable database to ensure information security.

## 1.3. Target Readers and Suggestions

---

This document is intended for KEEP FIT developers, users, and all involved stakeholders. For an overview, readers are encouraged to begin with **Part 1**, which provides a brief introduction to the system. **Part 2** details updates to project goals and highlights progress made since the previous version, outlining recent changes in the system.

For insights into architectural choices, **Part 3** delves into the high-level architecture and system-level analysis of KEEP FIT. Focusing on functionality, **Part 4** presents sequence diagrams for use cases and class diagrams to guide further development. As the project advances, system functionality is being refined, with related use cases detailed in **Part 5**.

To get a visual summary of recent changes, **Part 6** contains update snapshots. **Part 7** lists open questions, including challenges and design tasks for the next phase. Finally, **Part 8** (Annotated References) provides references for this document, while **Part 9** acknowledges team contributions.

## 1.4. Glossary of Terms

---

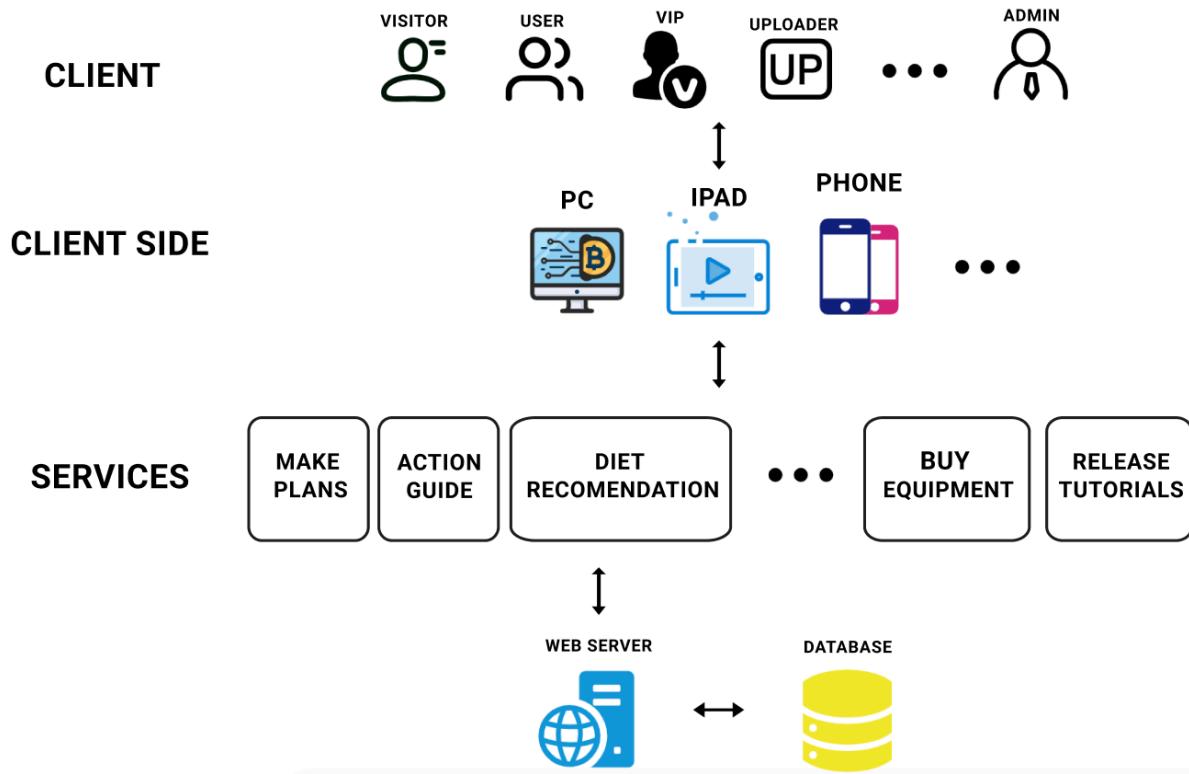
English terms	Terminology interpretation
<b>Keep-Fit</b>	The name of our project, the intelligent fitness platform, is designed to help users achieve their fitness goals smarter and more effectively.
<b>User</b>	The most basic users of the intelligent fitness platform can access the platform through registration, and use the most basic functions of the platform.
<b>VIP Member</b>	Ordinary users can become VIP members by recharging, and VIP members can use the richer functions in the platform, such as some paid tutorials, more intelligent AI chat systems and more accurate recommendation systems.
<b>Uploader</b>	Each kind of user(Users and VIP Members) can become a Uploaders by uploading exercise tutorials and sharing fitness experiences, and the Uploaders can also gain revenue and fans by selling paid fitness tutorials.
<b>Administrator</b>	The administrator needs to maintain the order of the platform and the stability of the environment, and can ban the illegal users. You can also make announcements, upload new fitness tutorials, diet combinations and fitness equipment, and so on.
<b>Fitness Tutorial</b>	Fitness tutorials are uploaded by uploaders or administrators, including paid and free, and cover a variety of fitness categories.
<b>Check-in</b>	After the user selects a fitness tutorial, the system automatically generates daily tasks and plans for the tutorial (the user can also plan the tutorial by himself/herself). Users need to complete the tasks every day to check in.

<b>English terms</b>	<b>Terminology interpretation</b>
<b>AI Motion Detection</b>	Users can upload photos of themselves during their workout process, and the system's AI will automatically check whether the exercise movements meet the specifications and give appropriate suggestions.
<b>Nutritional Composition</b>	After the user selects a fitness tutorial, the system will automatically generate the recommended diet package of the tutorial, and give the nutritional composition of the package. Users can also take photos and upload daily diet pictures, and the AI system will automatically identify and give the nutritional content list of the diet.
<b>Reward Mechanism</b>	Users need to complete the corresponding tasks every day according to the daily plan generated by the system(or tasks planning by himself/heself). After completing the tasks, they can get virtual rewards (such as level growth and platform transaction currency).
<b>Report</b>	Administrators regularly make announcements to various users of the platform, including some important notifications, new tutorials, and the latest feature points on the platform.
<b>Fitness Equipment</b>	Each fitness course is accompanied by a series of required fitness equipment, the system will give the corresponding purchase links and size size recommendations.
<b>Chat Platform</b>	In the chat platform, users can chat not only with their friends, but also with the system's AI robot, which can answer anything about fitness intelligently.
<b>Multi-device Access</b>	Users can access and use the functions of the platform through different devices (such as personal computers, tablets, smartphones, etc.). This flexible access approach can improve the user experience and engagement.
<b>Intelligent Recommendation</b>	The system automatically generates personalized suggestions based on the users' personal information, fitness goals, history, and preferences.

## 2. Introduction

### 2.1. Project Overview

The KeepFit system is a complete Web application containing the front-end Web end and back-end server parts. The server is responsible for processing user requests and interacting with the database to provide relevant data. Different user roles request different types of data from the server depending on their permissions. The server will return the corresponding data according to the role and permissions of the requester, such as tutorial management, device management, fitness records, and other functions. More detailed requirements and information will be further elaborated in the use case section, including the specific operational process and interaction details of each role.



## 2.2. Progress and Current Status

In the previous document, we provided readers with a comprehensive overview of the functionalities of KeepFit. This document builds upon that foundation by detailing the progress of our analysis model and architectural design.

This System Analysis Document presents the project's development through high-level architectural insights and the detailed architectural decisions made up to this stage. We have created a layered architecture diagram that outlines the structural hierarchy of the system and developed a package diagram to illustrate the components within each layer.

To enhance system reliability and streamline development, we also include an in-depth analysis model comprising sequence diagrams and class diagrams. In addition, during the analysis phase, we identified the need for system updates to enhance functionality and user experience. We have also refined the system snapshots to reflect these enhancements, making the interface more user-friendly and functionally robust.

## 2.3. Changes in the System

In the construction of the analysis model this time, we found that there were some operations can be optimized:

Only allow access to the data through given interface to avoid direct access to the database, which may cause errors in the data of the database. Separate all classes into three parts: boundary class, control class and entity class to make the system clearer and safer.

Add a some new functionalities to the existing system: Support for Wechat login in Login & Register System; Detailed tracking and data analysis in Fitness Tutorial Section; Dishes preparation tutorial in Fitness Diet Section; AI real-time chat in Fitness Action Coaching Section; Purchase in shopping malls in Fitness Equipment Section; Modify the use case diagram appropriately.

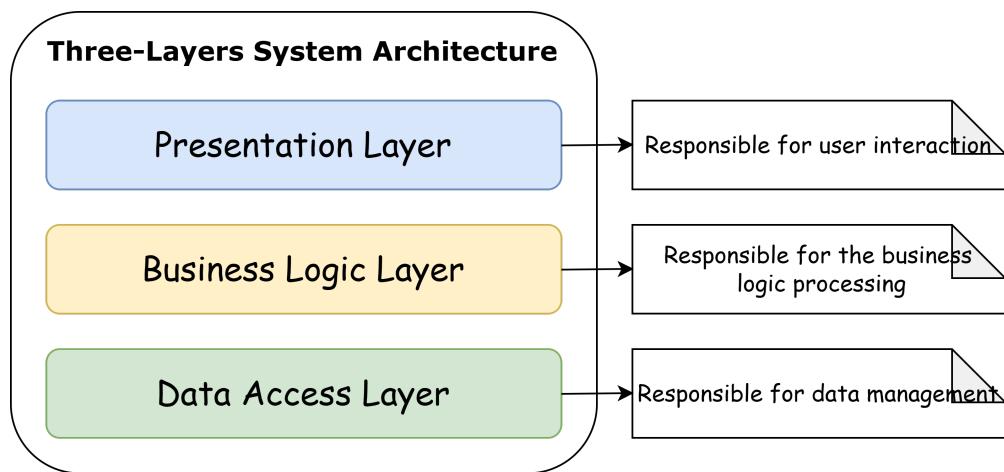
### 3. Architecture Analysis

KeepFit Is an intelligent fitness software for fitness and loved by the public. It needs to span multiple platforms (iOS, Android, Windows, Mac, Web) and provide services to users with different platforms and different personal information. Therefore, it is particularly important to design an architecture in line with the actual situation.

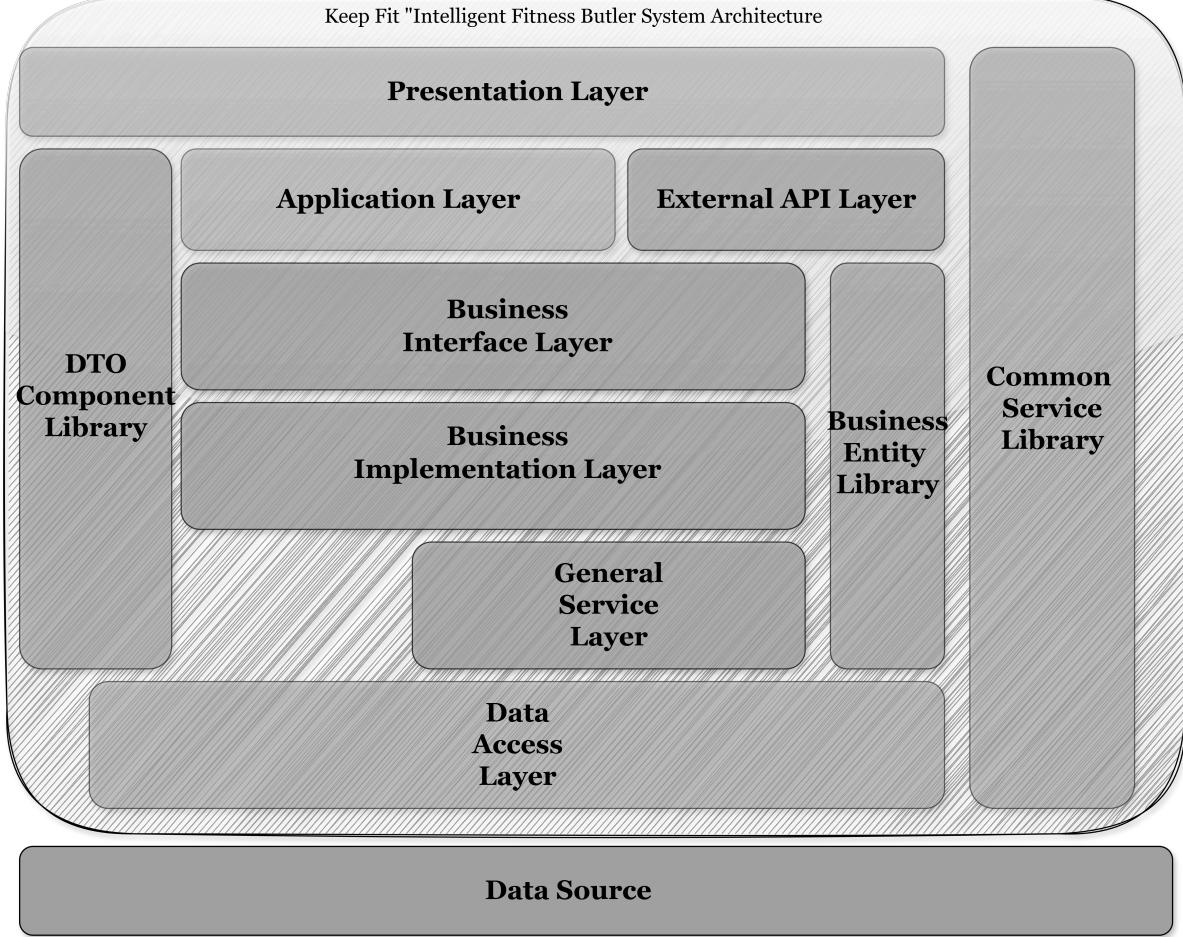
#### 3.1. System-Level Architecture Analysis and Designs

In order to find the appropriate architecture model, we first need to use previous system use cases, system activities, etc. The design, a more detailed analysis. After analysis, we found that KeepFit is composed of login registration subsystem, fitness course subsystem, healthy diet subsystem, action analysis subsystem and fitness equipment subsystem. Each system has strong independence and data with less overlapping data content. Therefore, we roughly divide our whole system into these five modules according to this classification.

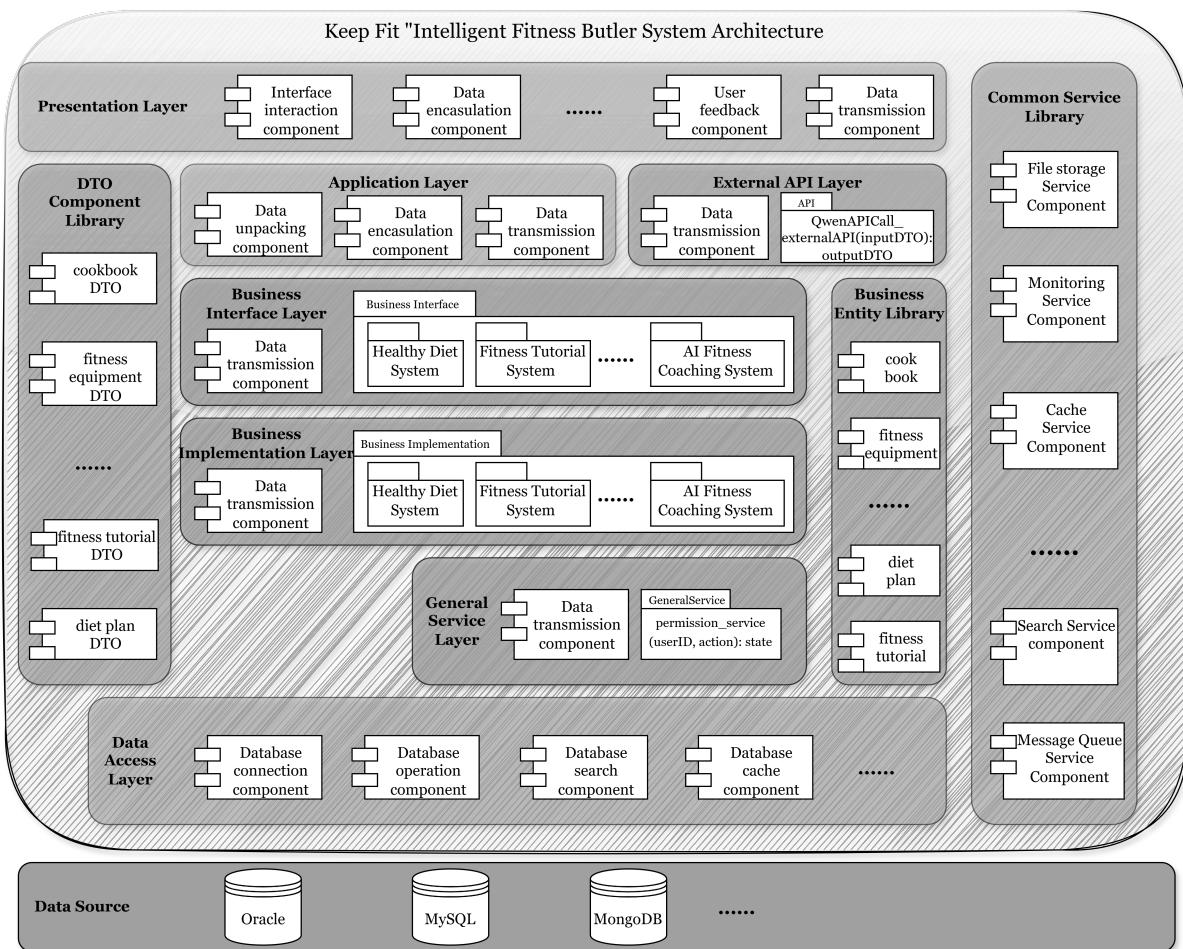
As we all know, the traditional three-story building design is as follows:



We have preliminarily completed the architecture analysis and design of the system through the subdivision of the traditional three-layer logical system architecture. The system-level architecture can be as shown below:

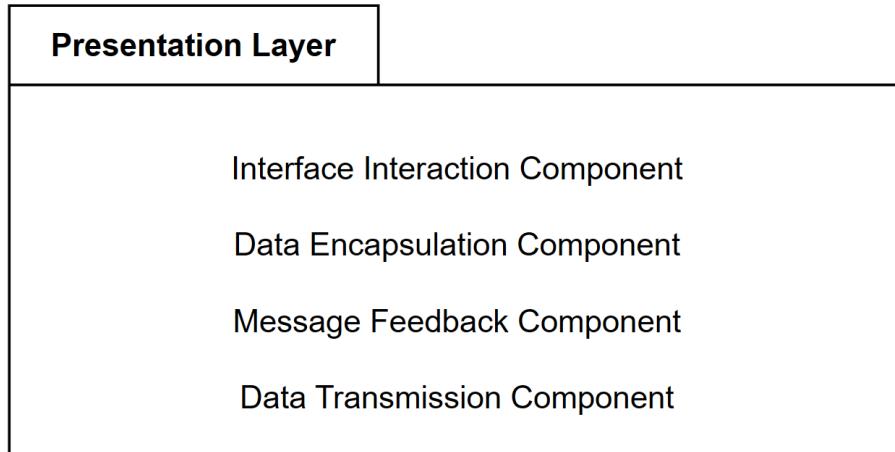


Combined with the functions of the system, we integrated the above architecture design into our intelligent fitness system KeepFit system through the package diagram and components:



### 3.1.1. Presentation Layer

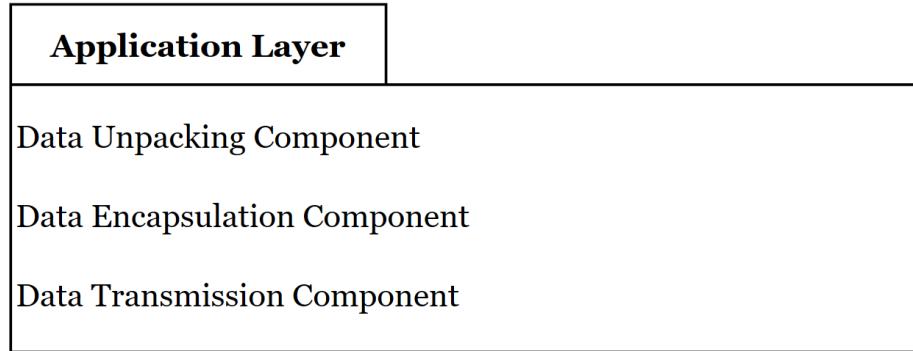
The presentation layer includes user interface information and components that facilitate user interaction. It handles user events through various components and encapsulates page data as a View Object (VO), allowing seamless transfer to application layer components for further business logic processing. Additionally, it can directly invoke external APIs to enable specific functionalities.



- **Interface Interaction Component:** This includes UI elements that directly interact with the user, such as buttons, input fields, and dropdown menus. These components are responsible for receiving user input and conveying the user's intentions to the system.
- **Data Encapsulation Component:** This component encapsulates the page's data state into a VO (view object) to facilitate transmission between layers. It converts complex page states into a format that's easier to transfer, providing a consistent data interface for application layer logic.
- **Message Feedback Component:** Responsible for providing real-time feedback to the user, such as pop-up messages, error alerts, and success notifications. This component helps users understand the current system state and the results of their actions, enhancing the user experience.
- **Data Transmission Component:** This component manages data transfer between the presentation layer and the application layer or external APIs. It can use network requests or other methods to transfer encapsulated data to other services, enabling dynamic system responses and data synchronization.

### 3.1.2. Application Layer

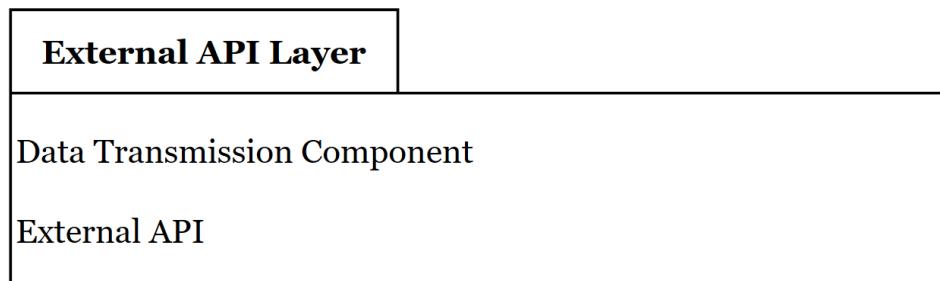
The application layer facilitates data exchange between the presentation and business logic layers. It retrieves information from the View Object (VO) passed by the presentation layer, converts it into a Data Transfer Object (DTO), and then routes it to components in the business interface layer for processing. Additionally, data received from the business interface layer is restructured as a VO before being sent back to the presentation layer for user interaction.



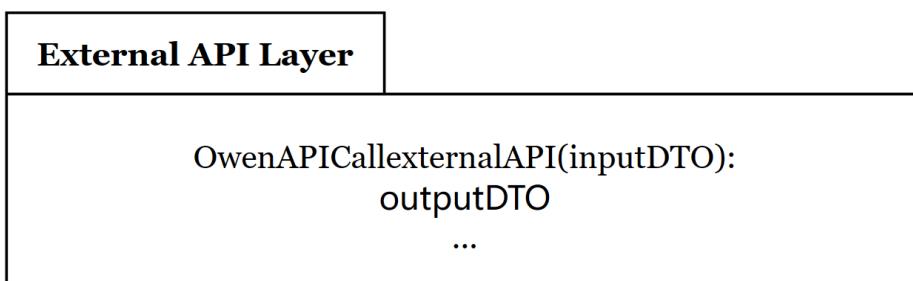
- **Data Unpacking Component:** Handles the extraction of data from incoming VOs, preparing it for further application layer tasks.
- **Data Encapsulation Component:** Packages data into DTOs, formatting it for streamlined transfer to the business logic layer.
- **Data Transmission Component:** Manages the movement of data between the application layer and both the presentation and business logic layers, ensuring smooth data flow throughout the system.

### 3.1.3. External API Layer

The external API layer handles data objects from specific layers or subsystems, invokes external interfaces for logical processing, and returns the processed data objects for use by the subsystem or presentation layer.

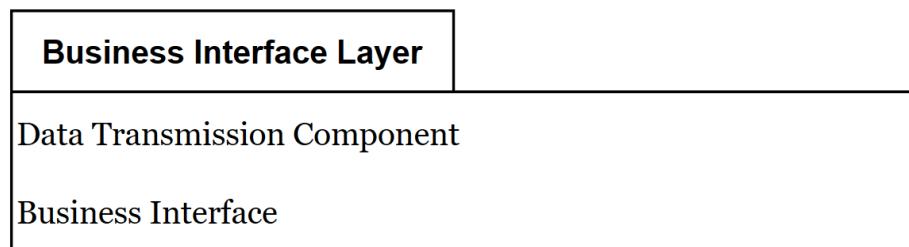


- **Data Transmission Component:** Manages the flow of data between the external API layer and other system layers, ensuring seamless interaction.
- **External API:** Connects with external interfaces to perform necessary logic operations, enabling the exchange and processing of data across systems.

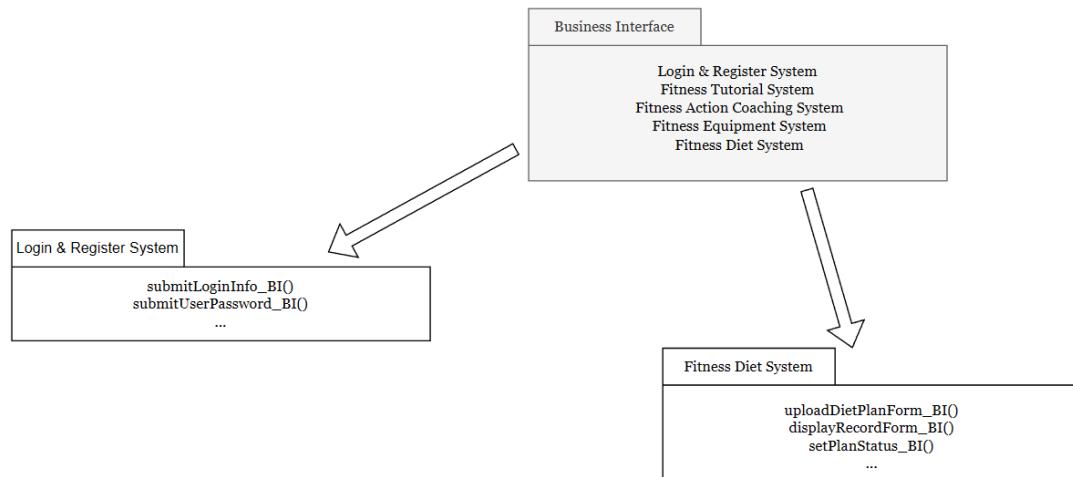


### 3.1.4. Business Interface Layer

The business interface layer represents the system's external functionality. It receives data from the controller layer, invokes the appropriate components in the business implementation layer to perform business logic processing, and then sends the processed information back through the business interface layer to the higher layers.



- **Data Transmission Component:** Facilitates data movement between the controller and business interface layers, ensuring efficient and reliable data flow.



- **Business Interface:** Acts as the access point for business logic, coordinating with implementation components to process requests and relay results back up the system.

### 3.1.5. Business Implementation Layer

The business implementation layer serves as the central component of the entire layered architecture, responsible for processing and executing the core business logic. This layer acts as the heart of the system, ensuring that all business rules, operations, and workflows are effectively handled, providing the necessary functionality for the application to meet its objectives.

## **Business Implementation Layer**

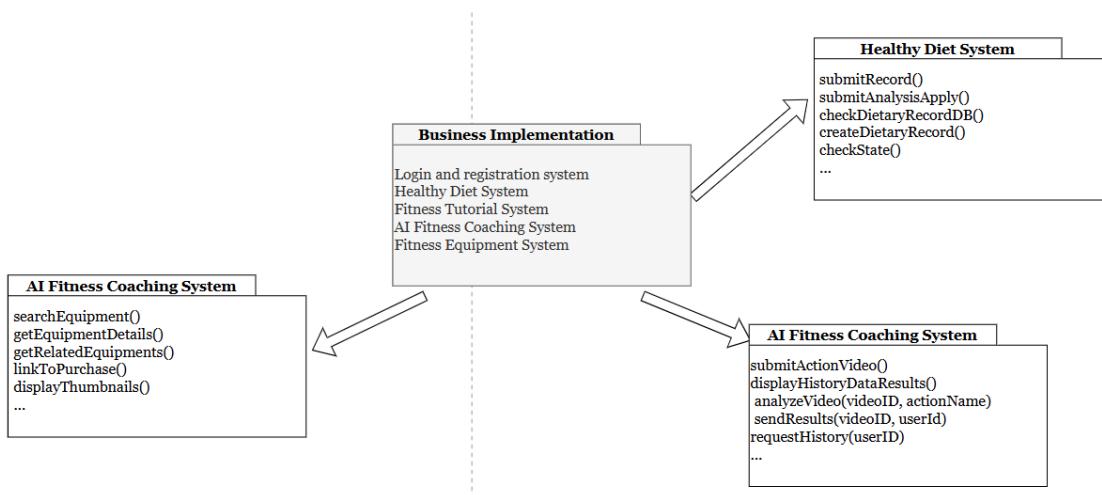
Data transmission component  
Business Implementation:  
Login and registration system  
Healthy Diet System  
Fitness Tutorial System  
AI Fitness Coaching System  
Fitness Equipment System

- **Data transmission component**

This layer takes in the data from the business interface layer and forwards it to the data access layer, ensuring it meets the specified business requirements.

- **Business Implementation**

The business logic at a finer level of granularity is organized within the business implementation package. To facilitate depth-first iterative development, this package is further divided into distinct functional areas, each handling specific tasks or processes.



### **3.1.6. General Service Layer**

The general service layer extracts the functional logic from the business implementation layer, aiming to enhance the reusability of business logic processing across the system.

## **General Service Layer**

General Service  
Data transmission component

- **Data transmission component**

This layer receives data from the business interface layer and forwards it to the data access layer, ensuring that the data is handled in accordance with the specified business requirements.

### General Service

```
register_service(userID, password, email): state  
logout_service(userID): state  
authentication_service(userID, token): state  
reset_password_service(userID, newPassword): state
```

- General Service

The general service package includes a set of reusable business methods that can be accessed and invoked by the business implementation layer. These methods are designed to handle common tasks, ensuring consistency and efficiency in the execution of business logic across the system.

### 3.1.7. DTO Component Library

The DTO (Data Transfer Object) layer comprises various classes tailored to meet specific business logic requirements. Utilizing DTOs helps filter out unnecessary information during data transfer, enhancing operational efficiency and ensuring data security.

#### DTO Component Library

```
dietPlanDTO(planID,title,description)  
dietRecordDTO(recordID,date,mealTime,nutrientComposition...)  
fitnessTutorialDTO(tutorialID,title,content,price,status)  
.....
```

### 3.1.8. Business Entity Library

The **Business Entity Library** is a collection of classes or components that represent the core business entities in a software system. These entities are typically modeled based on the main data structures and concepts within the business domain the software is intended to support.

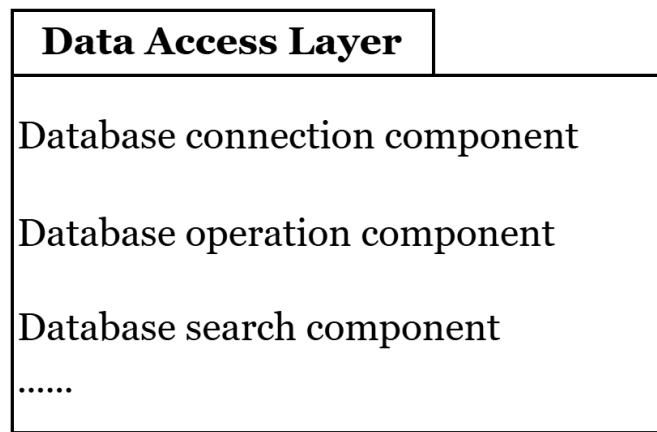
#### Business Entity Library

```
user(userID,username,password,accountBalance,gender...)  
cookbook(dishID,name,ingredients,instructions)  
equipment(equipmentID,name,description,price,thumbnail)  
.....
```

In this system, we have abstracted entities such as the *cookbook*, *fitness equipment* based on different key functional points, in order to reduce interactions with the data access layer and enhance the security of data operations.

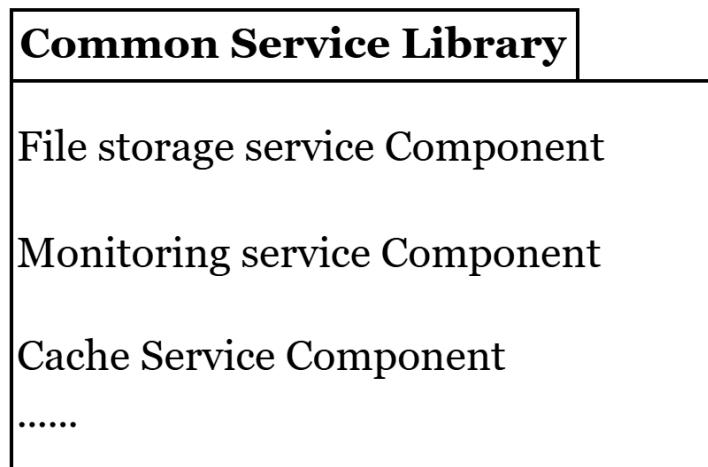
### **3.1.9. Data Access Layer**

The main purpose of the Data Access Layer (DAL) is to provide a unified interface for upper-layer business logic to read and manipulate data through components like the Database Connection Component, Data Cache component and so on, rather than directly interacting with the underlying database. This design enhances the system's maintainability, scalability, and security.



### **3.1.10. Common Service Library**

The common services library includes various functional services closely integrated with all layers of the system architecture, such as file storage service component, monitoring service component and so on, providing components at each level with accessible functionalities, ensuring comprehensive support across the system.

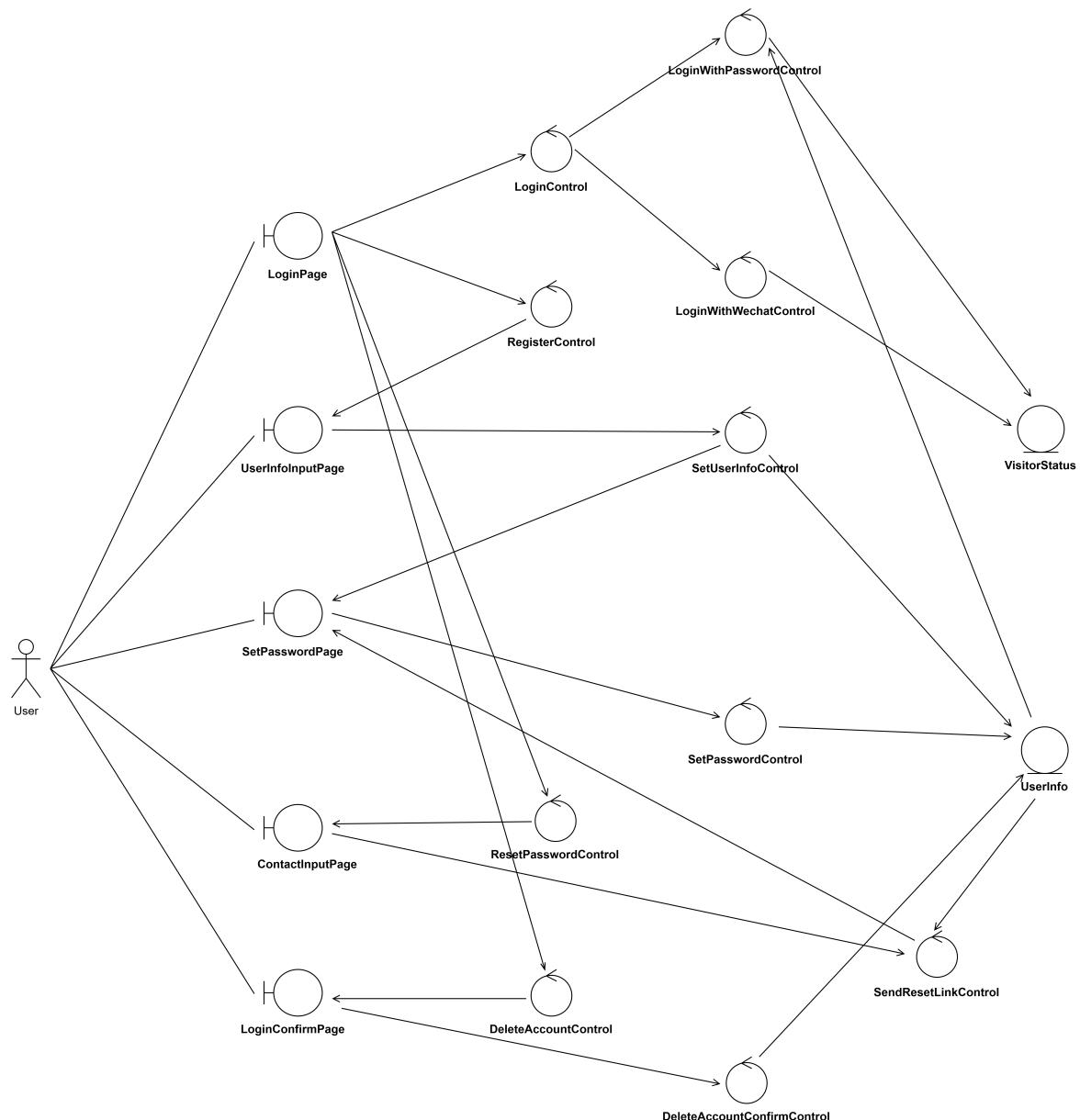


## 3.2. Robustness Diagrams

Robust diagram is a graphic tool to analyze the requirements and design the system structure. It is an important step between connecting the use case diagram and the detailed design. Robust graphs show the interactive relationships of objects in the system, and classify the boundary, control, and entity classes of the system to facilitate understanding and refining the functional requirements.

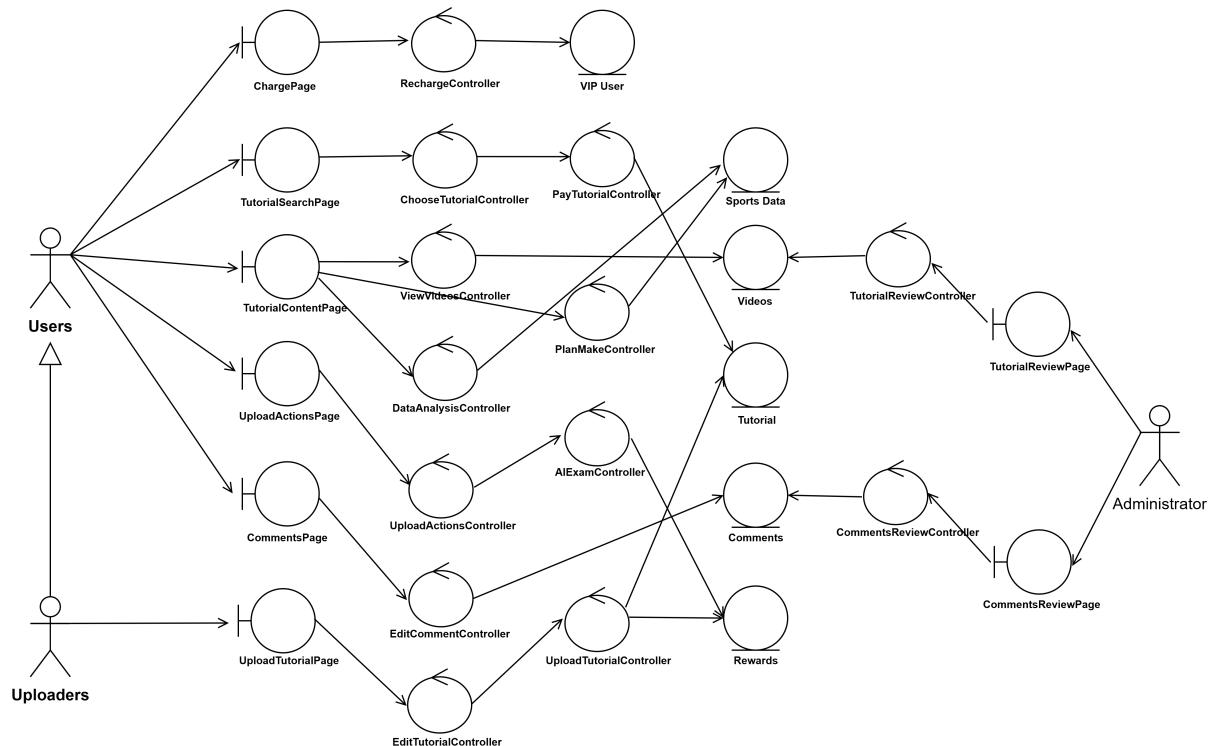
### 3.2.1. Login & Register System

This part of the robust map mainly shows the interactive process of user login and registration, explaining how the user can enter the system through the interface operation and create and manage the account in the system. This section can help understand how to safely and effectively process user authentication information and ensure that user data management meets application logic.



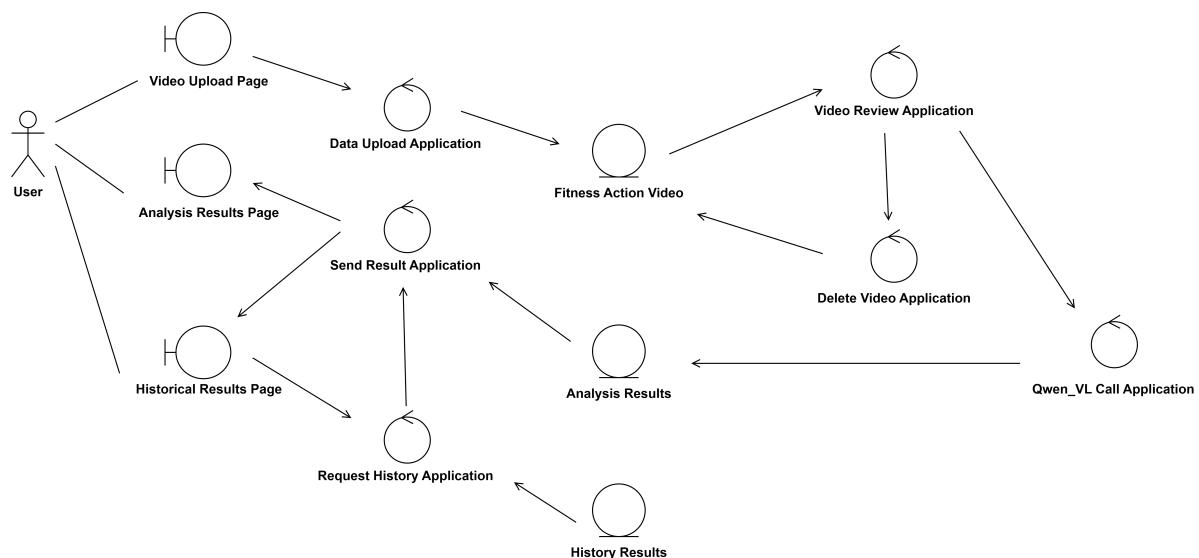
### 3.2.2. Fitness Tutorial Section

This section describes how users can find, buy, watch, and generate personalized fitness tutorial plans in the intelligent fitness software. It also includes how to track the user's fitness progress, analyze relevant data, and adjust their fitness plans according to their needs. The content shows the specific process of the user operation tutorial, and how these operations are handled and executed efficiently in the system. Through these functions, users can optimize according to their own situation, improve the fitness effect, and realize personalized fitness management.



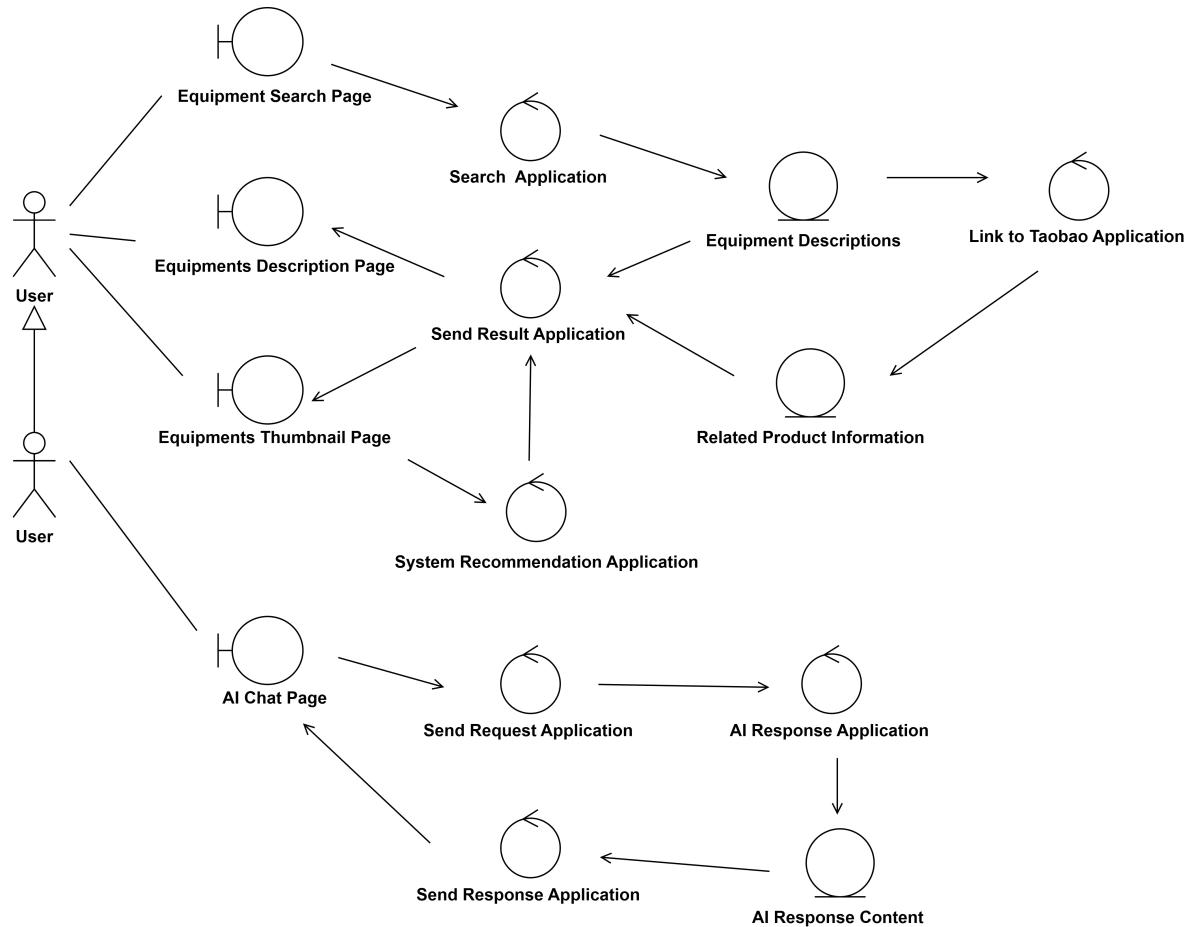
### 3.2.3. Fitness Action Coaching Section

This part of the robust map shows how users can upload fitness videos to get the AI to detect movements and get punch card rewards. This part is designed to support the user's fitness movement specification guidance, ensure that its movements meet the predetermined standards, and provide incentives after successful punching in.



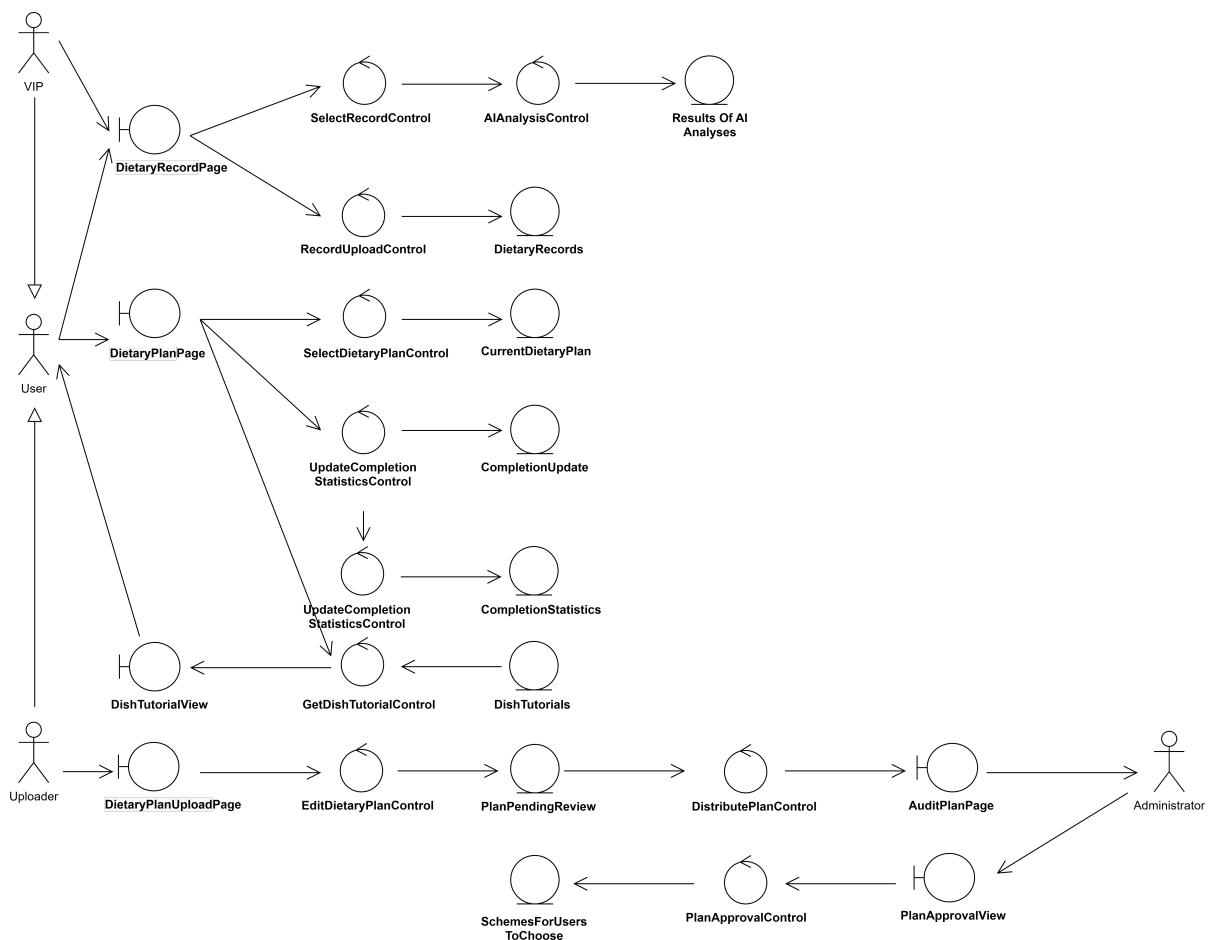
### 3.2.4. Fitness Equipment Section

This part of the robust map describes how users view, select and buy fitness equipment in the intelligent fitness software, how the system manages equipment information, and provides users with equipment recommendations. This part of the design to help users find the right fitness equipment and improve their fitness experience.



### 3.2.5. Fitness Diet Section

In this part of the robust map, the key steps and interactions of users' diet management and tracking in the application are mainly described. The graph allows users to view diet plans, edit personal needs, and analyze nutrients for comprehensive dietary guidance and progress feedback. The robust map also demonstrates how users can generate personalized diet plans and track their performance through interaction with the background AI system.

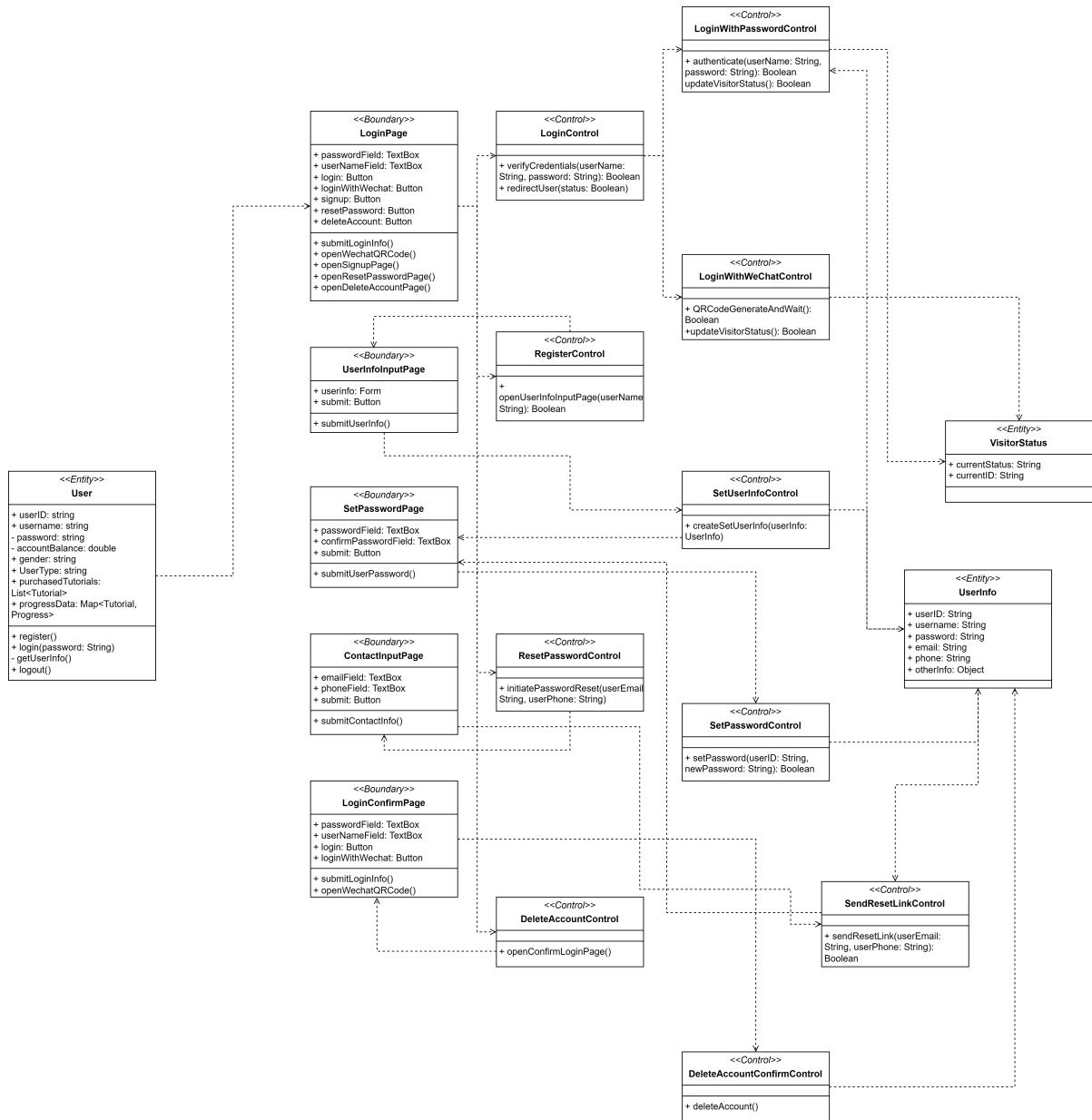


## 4. Analysis Model

### 4.1. Login & Register System

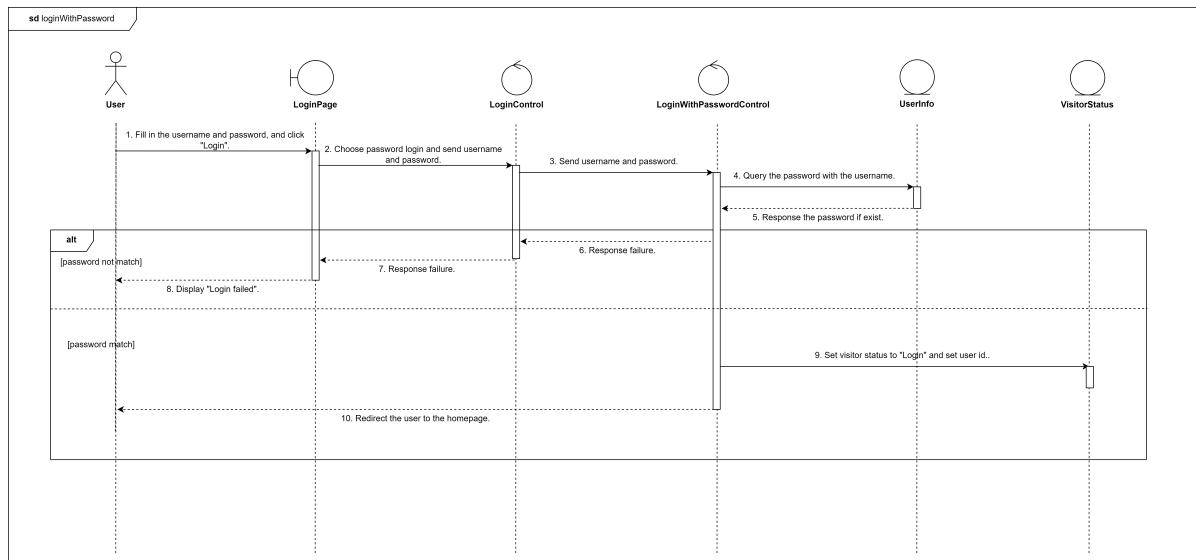
#### 4.1.1. Class Diagram

This is a class diagram of the login & register system in the KeepFit intelligent fitness platform. The user category includes basic user data management and session tracking, providing registration, login, password reset, and account deletion functions. The `User` and `UserInfo` classes handle core user information, including personal details, contact info, and login credentials, while `visitorStatus` manages the current session status for logged-in users. Control classes are responsible for core business logic, such as verifying login credentials, handling password-based and WeChat-based logins, processing password resets, and initiating account deletion. Boundary classes correspond to specific user interface pages, including login, registration, password setup, and account deletion confirmation pages. Users can log in, reset their passwords, and delete their accounts, while the system tracks and verifies user credentials and manages session status. The overall design enables a secure and streamlined user experience with functions for logging in, resetting passwords, deleting accounts, and managing session status.

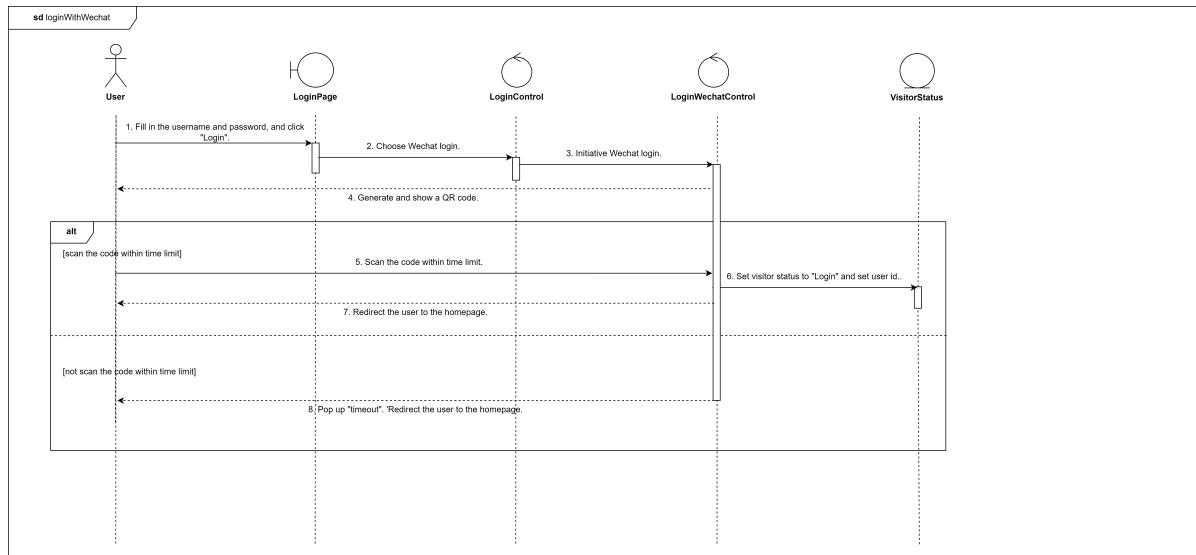


## 4.1.2. Interaction Diagram

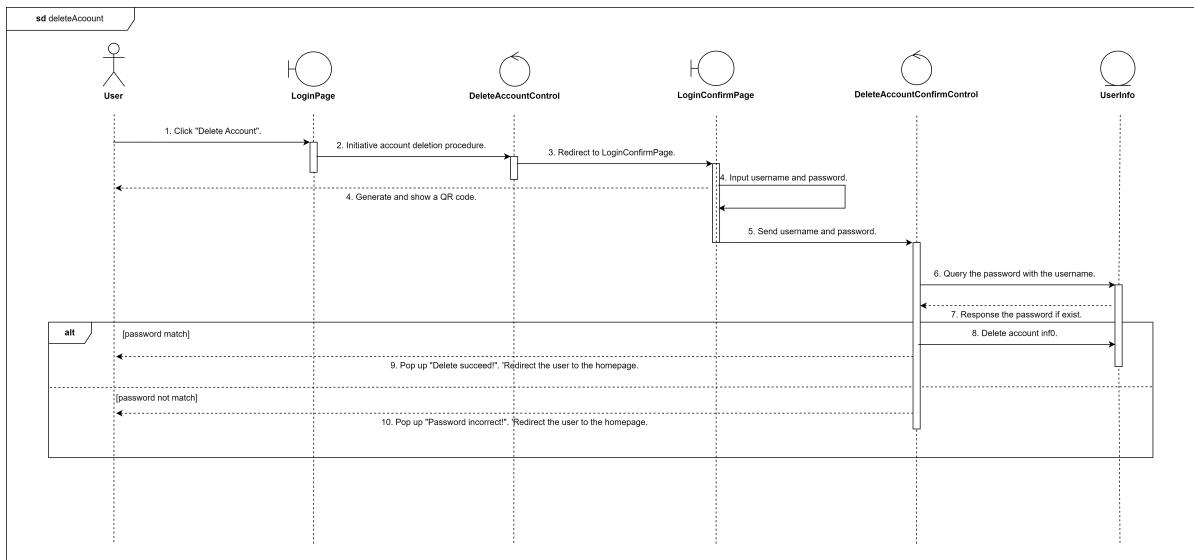
This diagram describes the process of logging in with a password. First, the user enters their username and password on the `LoginPage` and clicks "Login." The system sends these credentials through `LoginControl` to `LoginWithPasswordControl`, which verifies them by querying `UserInfo` to check if there is a match. If the username and password do not match, the system responds with a failure message, and `LoginPage` displays "Login failed" to inform the user. If the credentials are correct, `VisitorStatus` updates the user's status to "Logged in" and assigns a user ID. Finally, the system redirects the user to the homepage. This process shows the full sequence of user interaction for logging in with a password, including verification and feedback on success or failure.



This diagram illustrates the process of logging in via WeChat. The user initiates this process by selecting the WeChat login option on the login page. The system then forwards this request to `LoginWechatControl`, which generates a QR code displayed for the user to scan. If the user successfully scans the QR code within the given time, `visitorstatus` updates their status to "Logged in" and assigns a user ID, allowing them to access the homepage. If the QR code is not scanned within the time limit, the system times out, displays a timeout message, and redirects the user to the homepage. This interaction covers the complete WeChat login flow, from QR code generation to successful login or timeout handling.



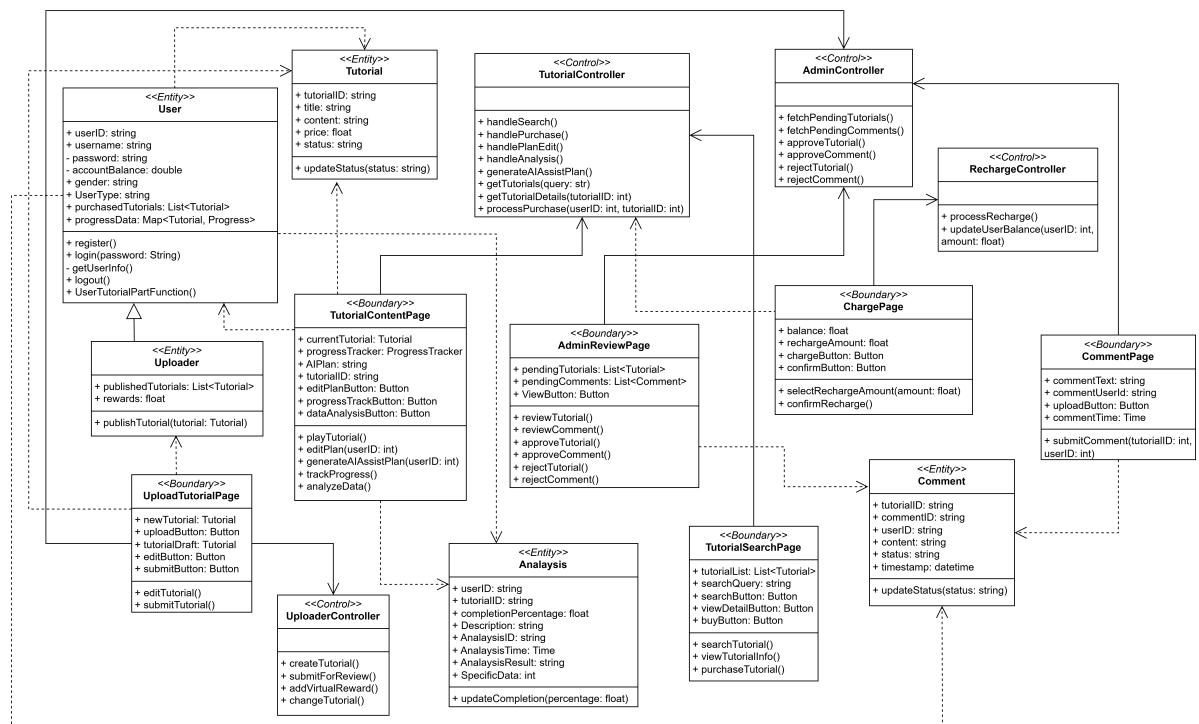
This diagram outlines the process of deleting an account. The user begins by clicking "Delete Account" on the login page. The system, through `DeleteAccountControl`, initiates the deletion process and redirects the user to a confirmation page where they must enter their username and password for verification. `DeleteAccountConfirmControl` checks these credentials against `userInfo`. If the credentials match, the account is deleted, and the system displays a "Delete successful" message before redirecting the user to the homepage. If the credentials do not match, a "Password incorrect" message is displayed, and the user is redirected to the homepage without any deletion occurring. This process ensures secure account deletion, requiring credential confirmation and providing feedback for both successful and failed attempts.



## 4.2. Fitness Tutorial Section

### 4.2.1. Class Diagram

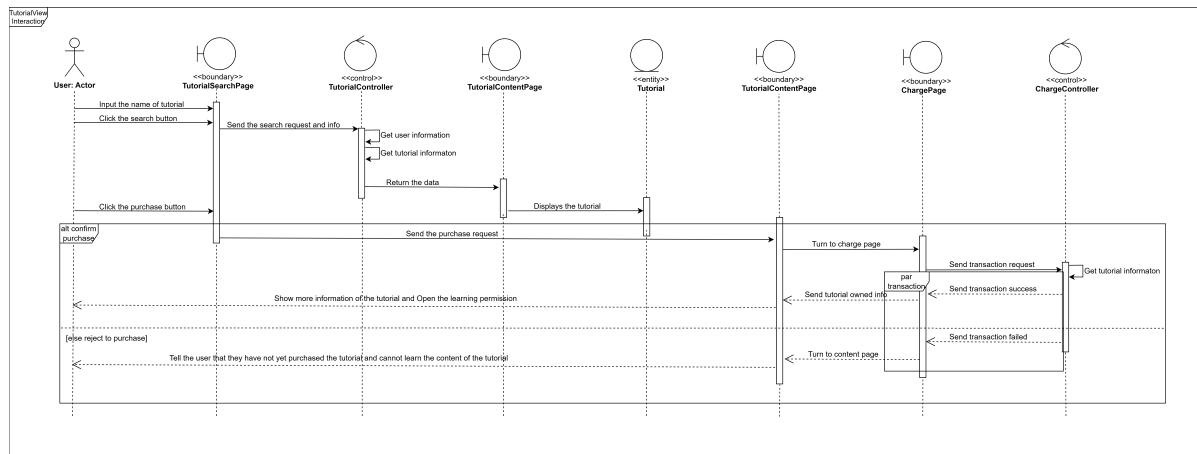
This is a class diagram of the fitness tutorial section in the KeepFit intelligent fitness platform. The user category includes ordinary users and uploaders, providing registration, login, purchase tutorial and other functions. Tutorial class management tutorial content, status, and comments. The control class is responsible for business logic, such as tutorial search and purchase, AI plan generation, and administrator review. Boundary classes correspond to specific pages, including tutorial content, recharge, comments, and tutorial search pages. Users can purchase and view tutorials, uploaders can post new tutorials, and administrators can review tutorials and comments. The overall design enables the search, purchase, watch, comment and management functions of the fitness tutorial.



## 4.2.2. Interaction Diagram

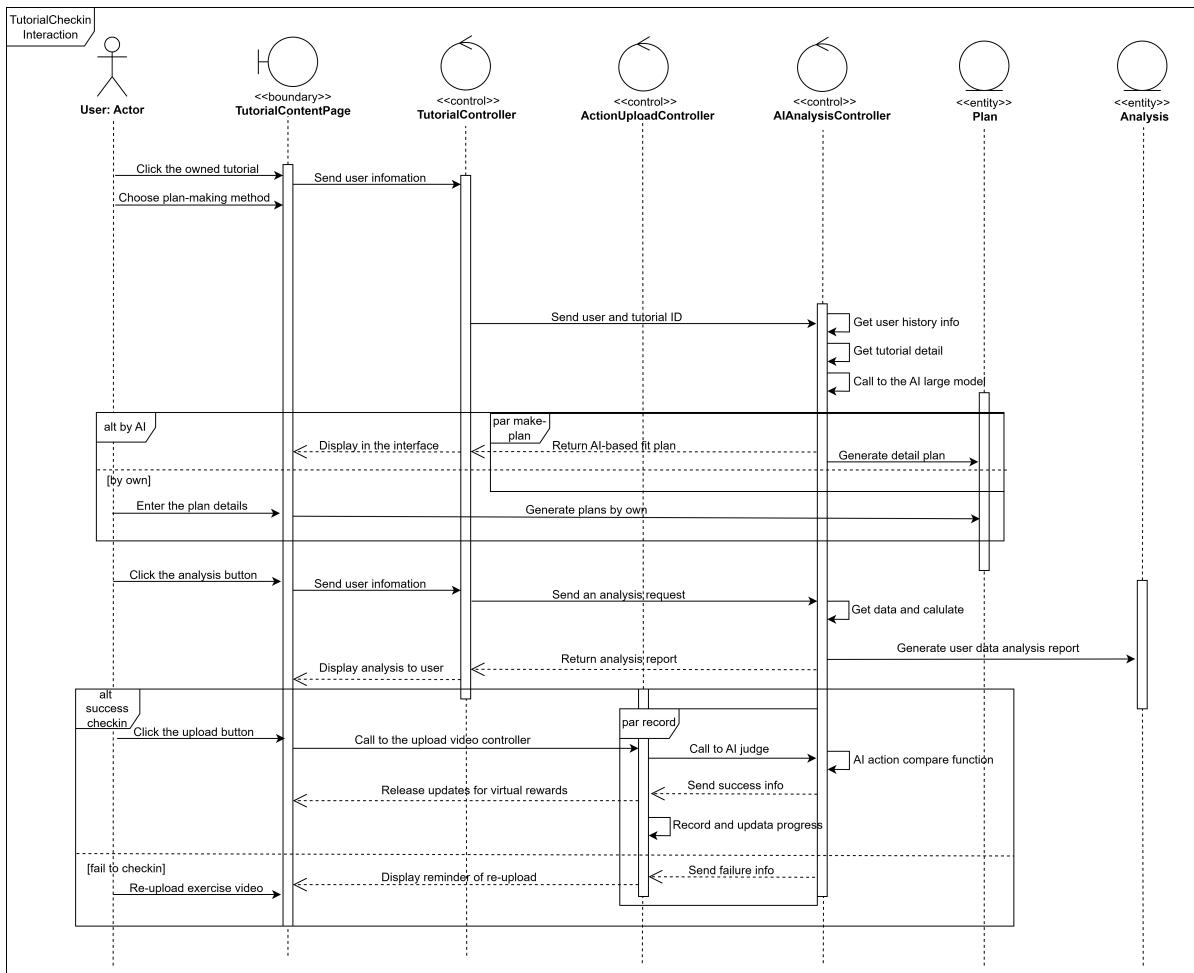
- Tutorial View Interaction

The interaction diagram describes the process of browsing the tutorial and purchasing it. First, the user enters the query in the tutorial search interface, the system retrieves the relevant tutorial through TutorialController and returns the tutorial list, and the user can view the detailed introduction. The user selects a specific tutorial to enter the details page, and the system checks whether the user has purchased the tutorial. If the purchase is not made, the system prompts the user to purchase, and the user confirms and then enters the payment page to enter the amount and confirm the purchase request. ChargeController Handle payment transactions and updates the user's account balances. After a successful purchase, the system allows users to access the tutorial details. The entire process shows the full user interaction of the tutorial browsing and buying.



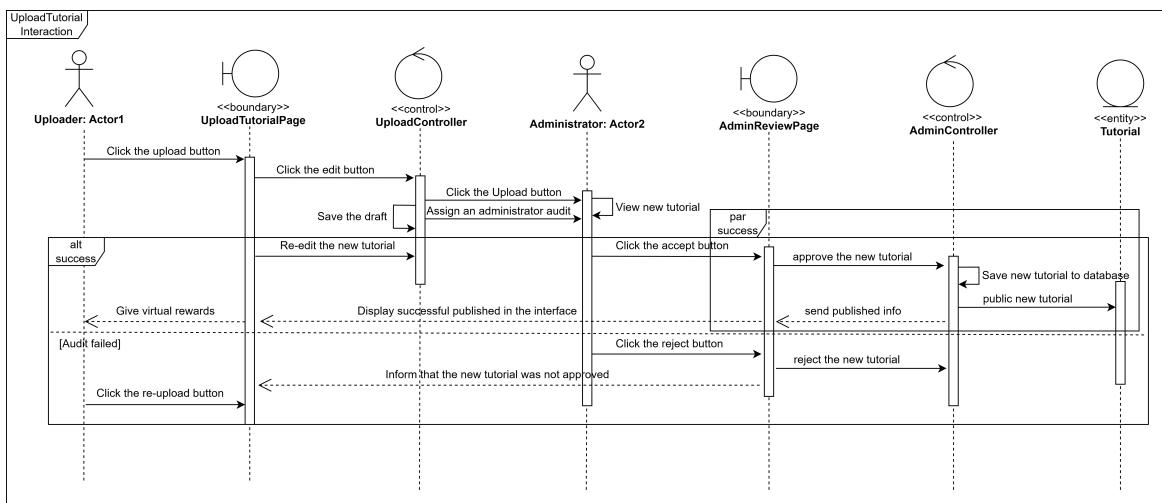
- Tutorial Checkin Interaction

The interactive diagram shows the user's punching in process in the tutorial. The user first starts to punch in, and the system calls the AI function through the TutorialController to detect the standardization of the user's action. If the test passes, the card is successful, and the system updates the user's punching record and rewards the corresponding points. If the test fails, the system prompts the user to try again. The user can also choose to generate the AI auxiliary plan, and the system generates a personalized plan according to the user's current training state. After that, the system records and analyzes the user's training data to help the user track the progress. This process fully reflects the interactive details of users' clocking, planning and training progress tracking in the intelligent fitness software.



### • Upload New Tutorial Interaction

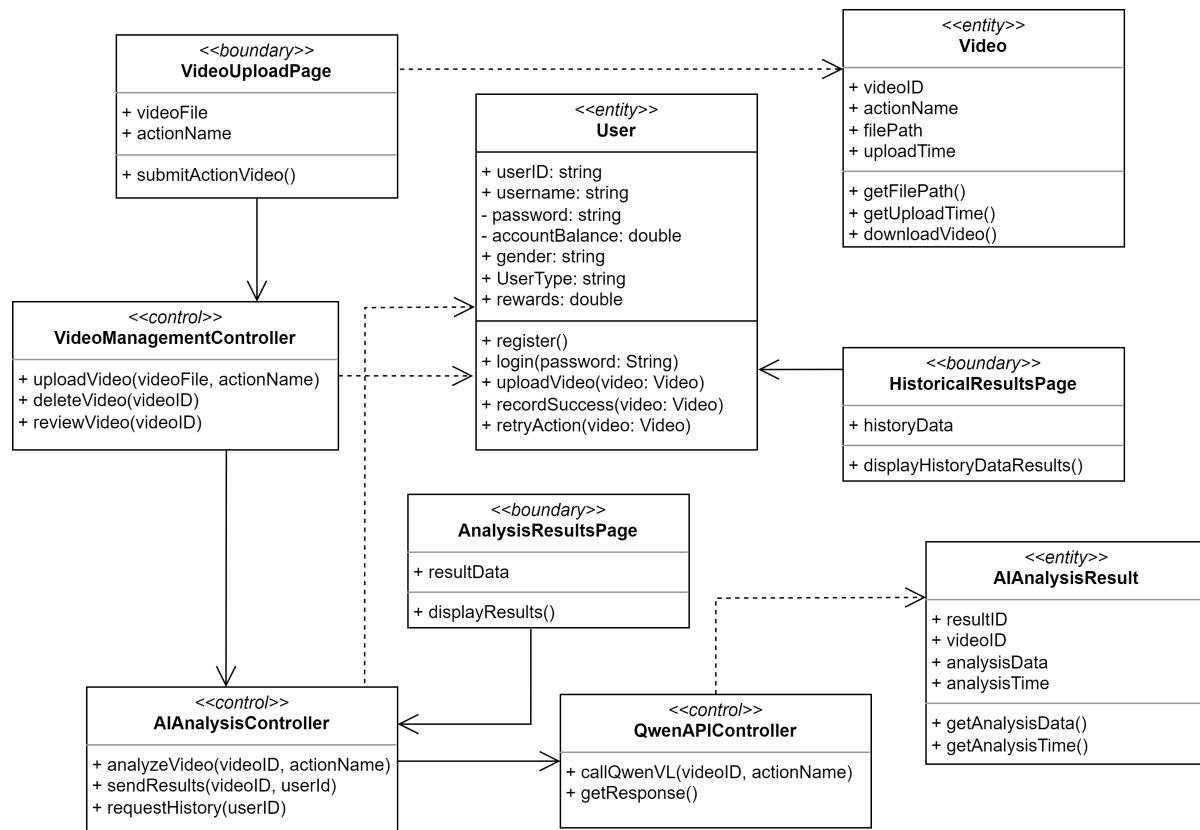
In the interactive diagram uploaded by the tutorial, the UP master first selects the new tutorial on the UploadTutorialPage page and fills in the relevant content. After filling in, the UP master clicks submit, and the page will save the tutorial draft through UploaderController. The UP master can select the preview and edit, and submit again after the modification is completed. When the tutorial content is confirmed, UploaderController will submit the tutorial to the administrator for review, and AdminController will check the content of the tutorial. After the approval, the status of the tutorial is updated to "released" and officially launched on the platform. If the audit fails to pass, it will be sent back to modify again. After the audit is completed, the UP master will get the corresponding virtual reward points to encourage them to continue to create quality content.



## 4.3. Fitness Action Coaching Section

### 4.3.1. Class Diagram

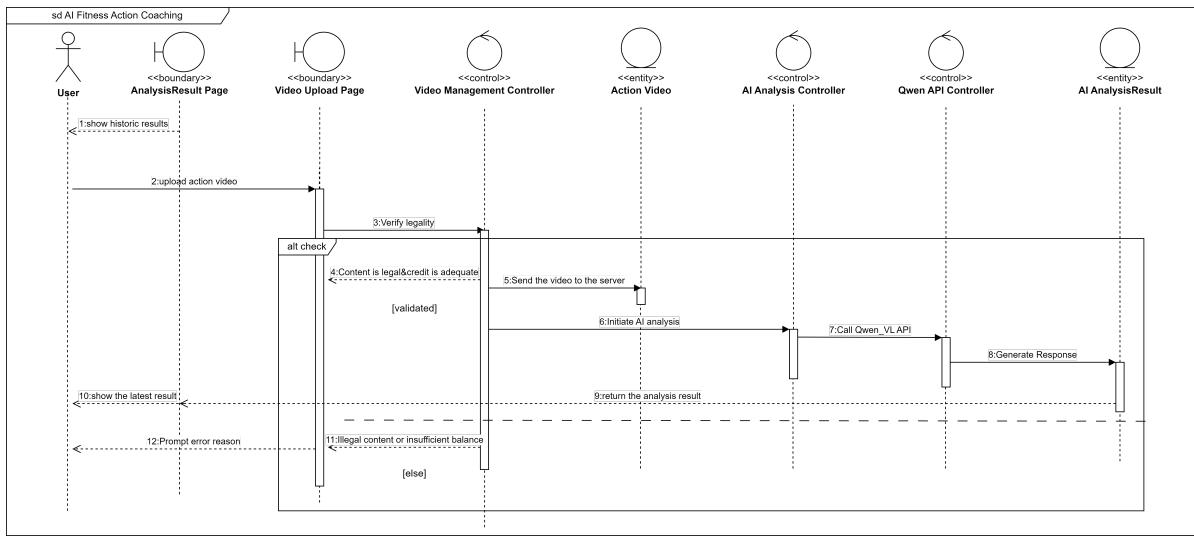
This class diagram of the AI fitness action analysis system illustrates the main classes and their relationships. The system includes the `User` class, which allows users to register, log in, upload videos, record successful actions, and retry actions. The `Video` class stores relevant information about the uploaded videos. Control classes such as `VideoManagementController` handle video uploading, deletion, and review, while the `AIAnalysisController` is responsible for video analysis, sending results, and requesting history, interacting with the `QwenAPIController` to call APIs and obtain analysis data. Boundary classes like `VideoUploadPage`, `AnalysisResultsPage`, and `HistoricalResultsPage` are used for users to upload videos, display analysis results, and show historical data, respectively. The `AIAnalysisResult` entity class stores analysis results and timestamps, showing how the system facilitates the entire process from video upload to analysis and historical queries.



### 4.3.2. Interaction Diagram

- AI Fitness Action Coaching

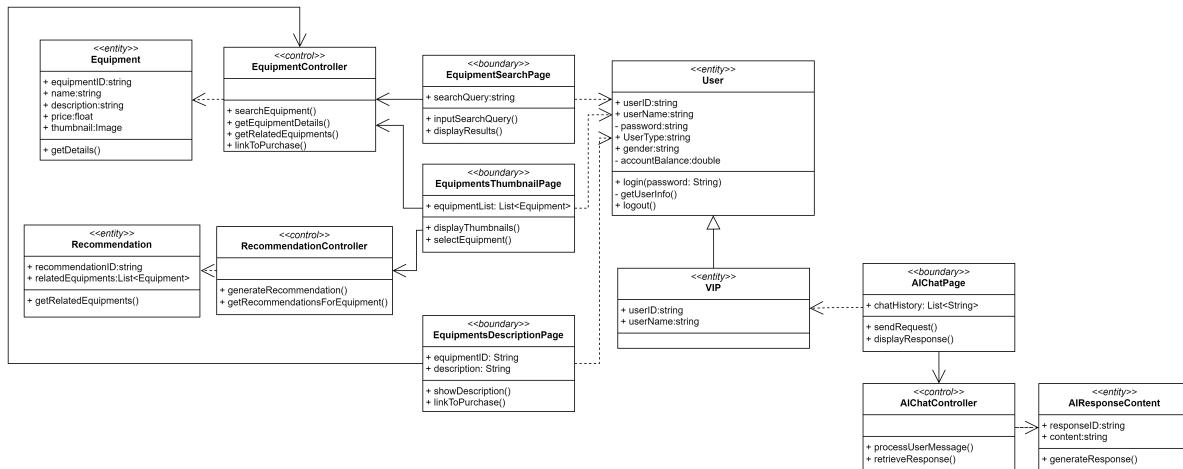
This sequence diagram describes the process of AI fitness action analysis. First, the user uploads a fitness action video, and the system verifies its content legality and checks if the user's daily usage quota is sufficient. If the content is legal and the quota is adequate, the video is sent to the server to initiate AI analysis. The system then calls the Qwen\_VL API for video analysis and generates the analysis result. Once the analysis result is returned, the system displays the latest analysis result. If the content is illegal or the user's quota is insufficient, the system will prompt an error message.



## 4.4. Fitness Equipment Section

### 4.4.1. Class Diagram

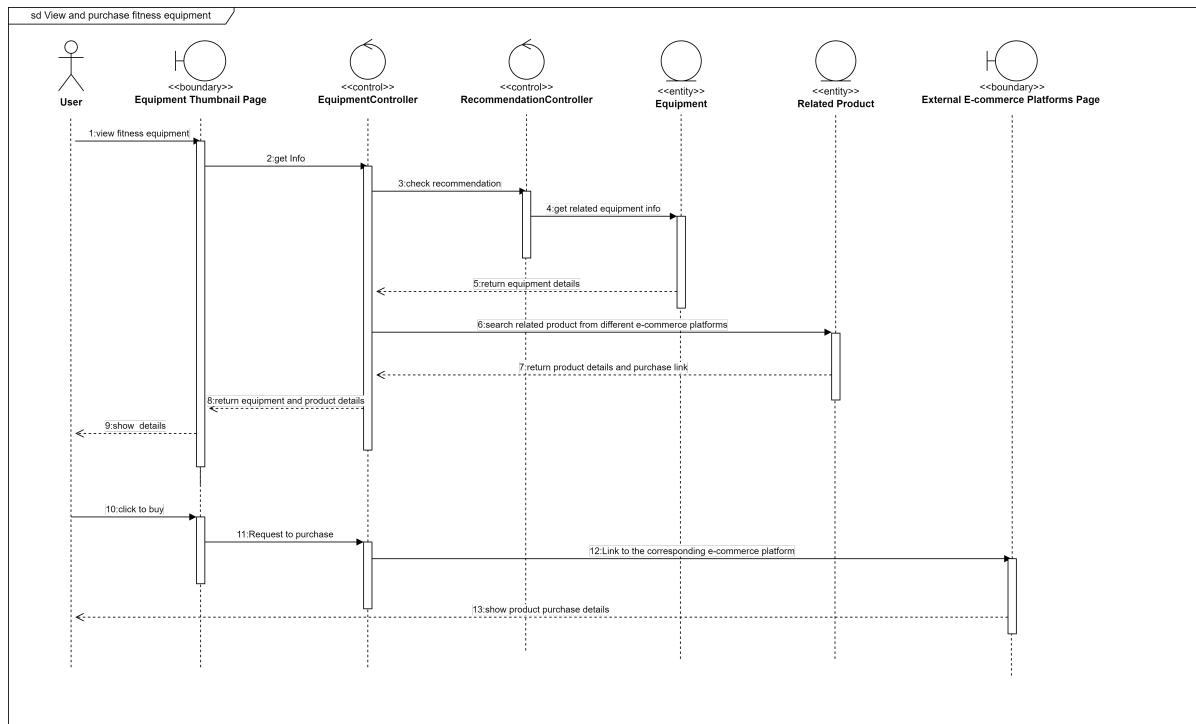
This class diagram illustrates the structure of a fitness equipment system, which includes multiple entity classes, control classes, and boundary classes. The main entity classes are `Equipment`, `User`, `VIP`, `Recommendation`, and `AIResponseContent`, representing equipment, user information, and recommendation content, respectively. The control classes include `EquipmentController`, `RecommendationController`, and `AIChatController`, which handle business logic such as equipment searching, recommendation generation, and user message processing. The boundary classes, `EquipmentSearchPage`, `EquipmentsThumbnailPage`, `EquipmentsDescriptionPage`, and `AIChatPage`, are used for user interaction, such as searching for and displaying equipment information and AI chat interface. These classes collaborate through composition and association relationships to enable the overall functioning of the system.



### 4.4.2. Interaction Diagram

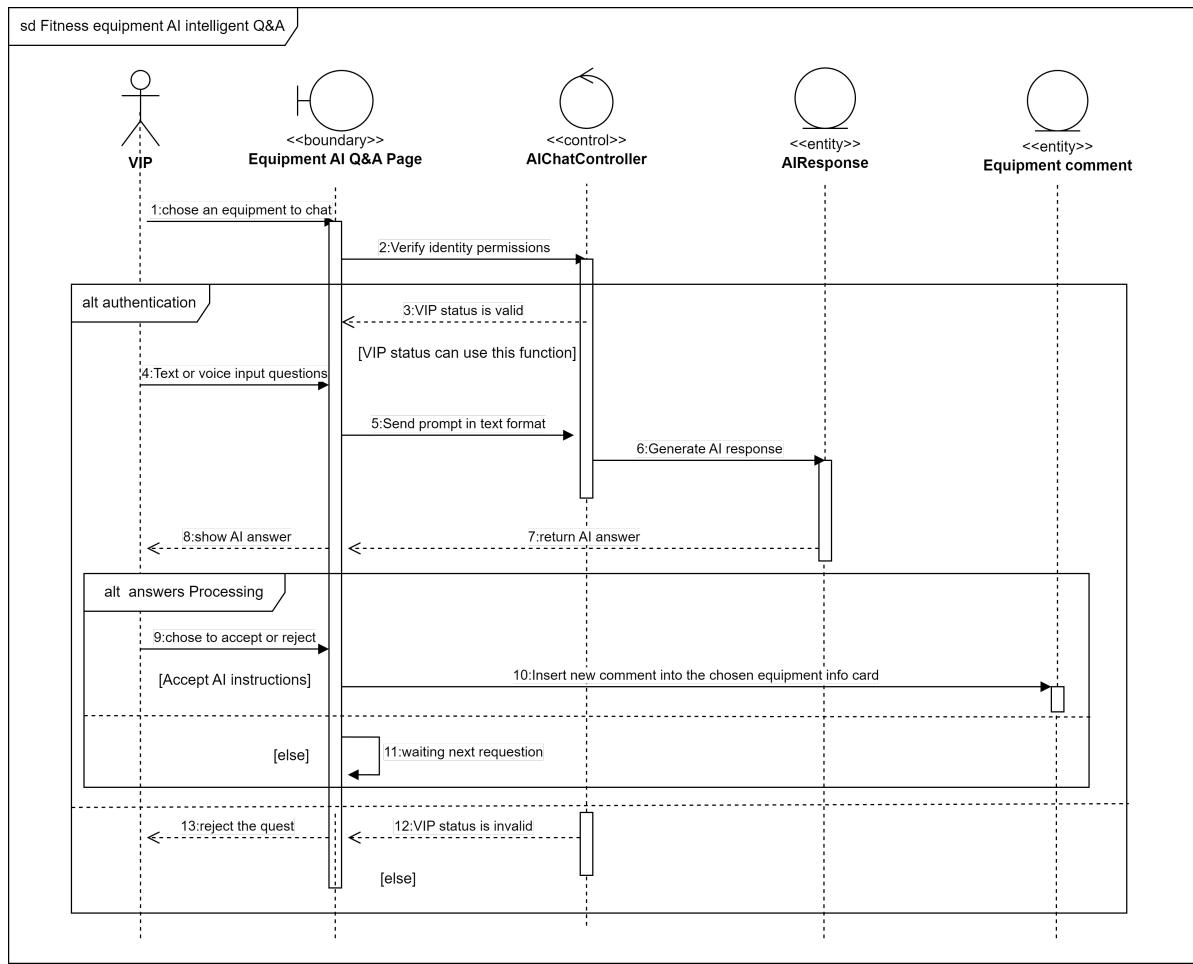
- View and Purchase Fitness Equipment

This sequence diagram describes the process of a user viewing and purchasing fitness equipment in the fitness equipment system. The user first browses the equipment thumbnail page and retrieves the equipment information. The system checks the user's personalized information to recommend content and returns relevant equipment details. Then, the system searches for related products from different e-commerce platforms and returns the product details and purchase link. After viewing the detailed information, the user can click to buy, and the system will redirect to the corresponding e-commerce platform for the purchase and display the purchase details.



- **Fitness Equipment AI Intelligent Q&A**

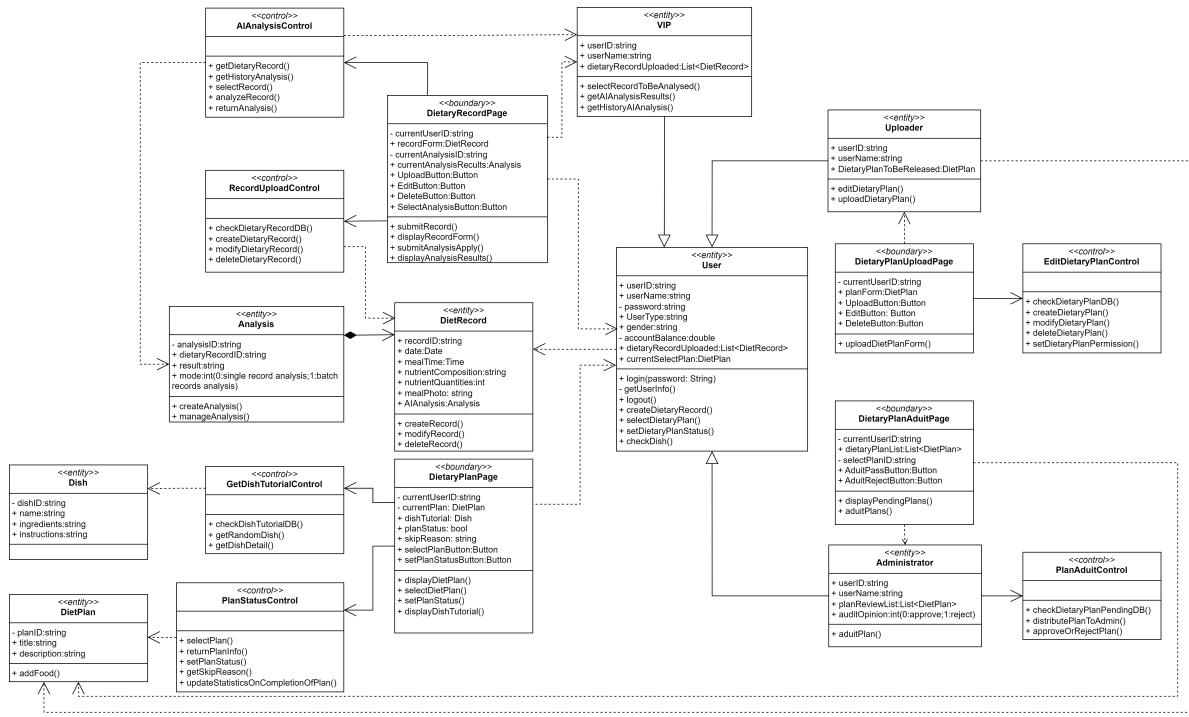
This sequence diagram describes the process where VIP users interact with a specific fitness equipment agent and update the equipment information card. The VIP user selects the equipment for interaction, and the system verifies their identity and permissions. Once the VIP status is confirmed, the user can input questions via text or voice. The system sends the question in text format to the AI for a response, which is then returned. The user reviews the response and can choose to accept or reject it. If accepted, the system inserts the new comment into the equipment information card. If rejected, the request ends. If the VIP status is invalid, the system will reject the request.



## 4.5. Fitness Diet Section

### 4.5.1. Class Diagram

The class diagram represents the architecture of the intelligent diet management system within the KeepFit platform. The entity classes manage the core data, which handle the details of diet plans, individual dishes, and user dietary records, respectively. The control classes are responsible for the business logic, which enable users to add food items to their diet plans, select diet plans, analyze their dietary records, and create new records. Boundary classes correspond to specific user interface pages within the system, which provide interfaces for users to interact with their diet plans and records. Users can view and manage their dietary plans, uploaders can contribute by adding new dishes to the system, and administrators have the ability to review and manage diet plans and records.

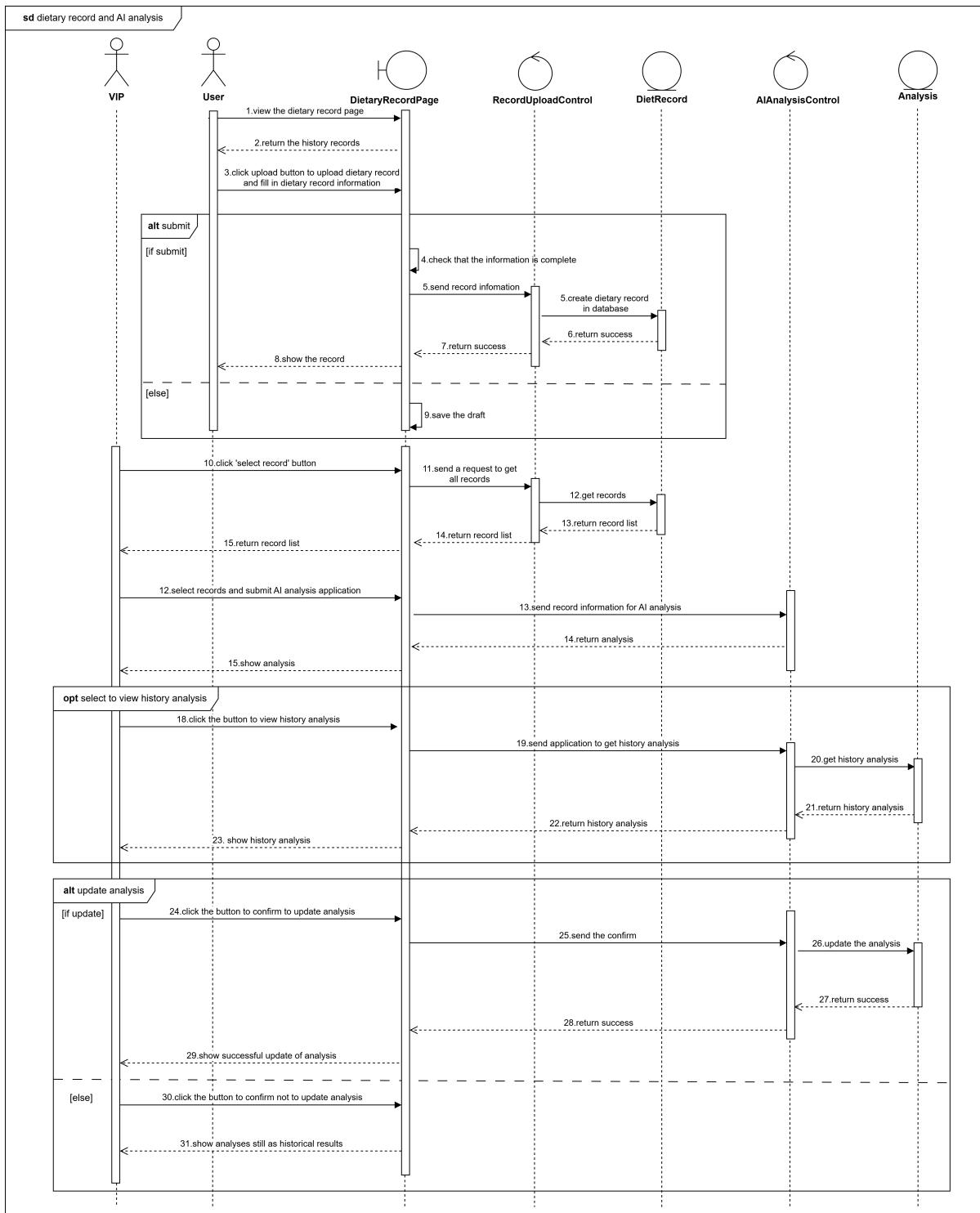


## 4.5.2. Interaction Diagram

- **Upload Dietary Record Interaction**

In this interactive diagram, users first navigate to the dietary record page to manage their own dietary records. They can view their dietary record page, return to check the history of their records, and click the upload button to fill in the required information and submit a new dietary record. After submission, RecordUploadControl checks whether the information is complete and sends the record details to the database. If successful, the system returns a confirmation message, and users can view the record. If not submitted, it is saved as a draft.

VIP users can also select records and submit applications for artificial intelligence analysis. AIAnalysisControl receives the AI analysis application, processes the dietary information, and returns the analysis results. Users can then view the analysis results. Additionally, users can choose to view historical analyses, and the system retrieves and returns the historical analysis, which is then displayed to the user. Users can opt to update the analysis and click the relevant button to confirm the update. The system sends a confirmation message and returns the successfully updated analysis results, displaying the updated outcomes to the user.



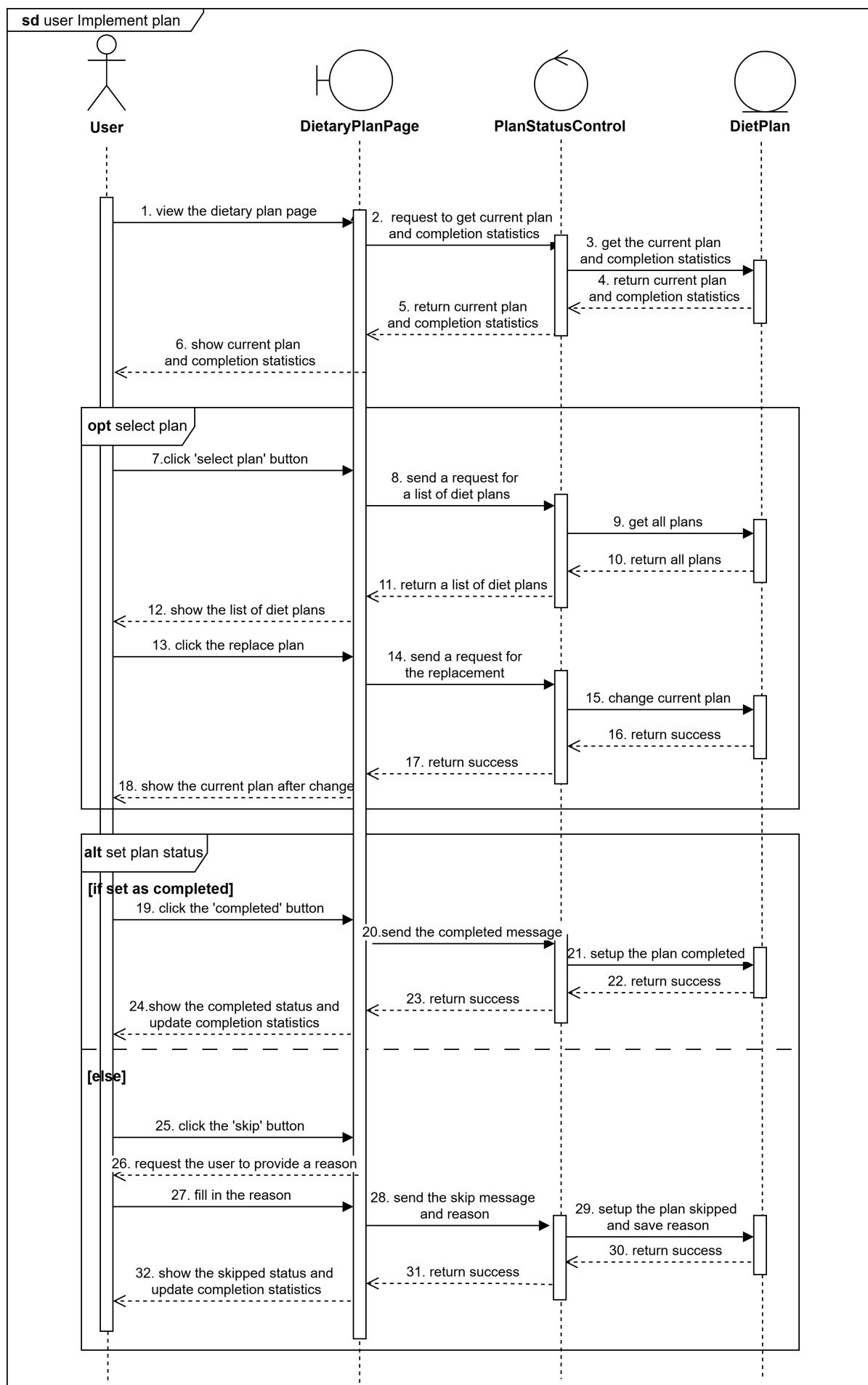
### • Implement Dietary Plan Interaction

In this interactive diagram, the user begins by accessing the DietaryPlanPage to manage their dietary plans and track their progress.

The user has the option to select a different plan. They click the 'select plan' button and the system retrieves all plans and returns a list, which is then displayed to the user, allowing them to choose a new plan. Upon selecting a plan, the user sends a request for the replacement, and the system changes the current plan, and updating the display to show the new current plan.

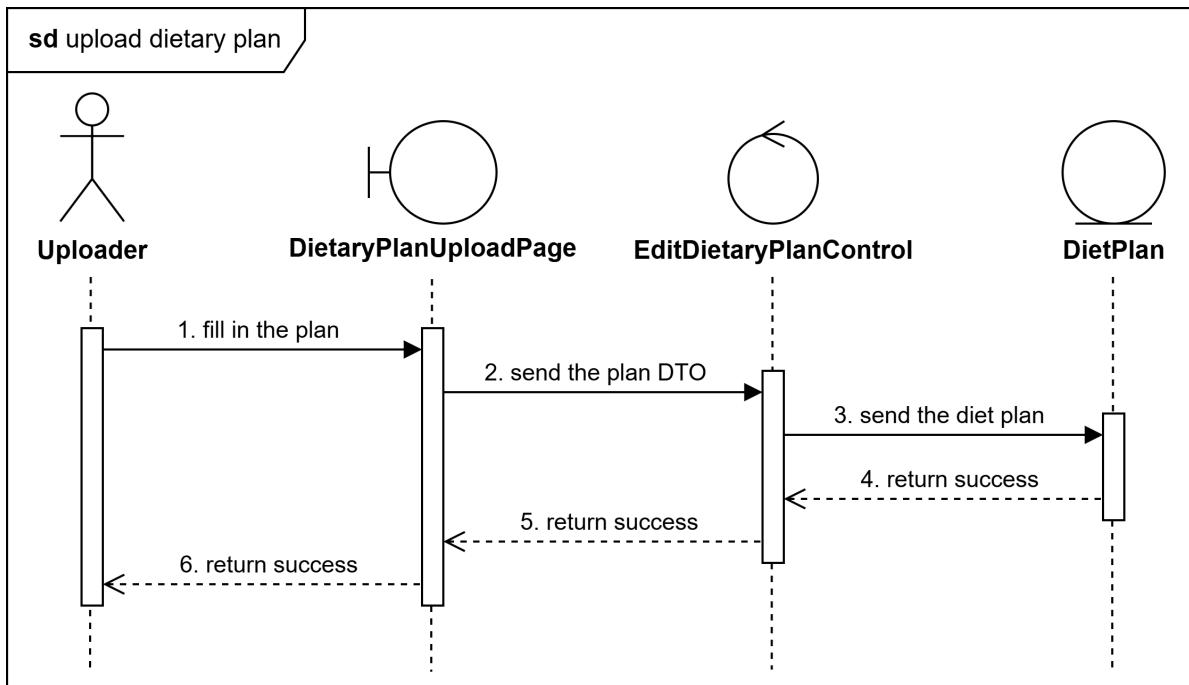
Alternatively, the user can set the plan status as completed. They click the 'completed' button and the system sets the plan as completed and returns a success message, updating the completion statistics and displaying the completed status to the user.

If the user decides to skip the plan, they click the 'skip' button and fill in the reason for skipping. The system sends the skip message, sets the plan as skipped, and saves the reason, returning a success message and updating the completion statistics.



- **Upload Dietary Plan Interaction**

In the interaction diagram, the Uploader fills in the necessary details of the plan and then the system sends the plan information to the controller. The control class then sends the diet plan to the backend for processing. The success message is returned after successful upload.



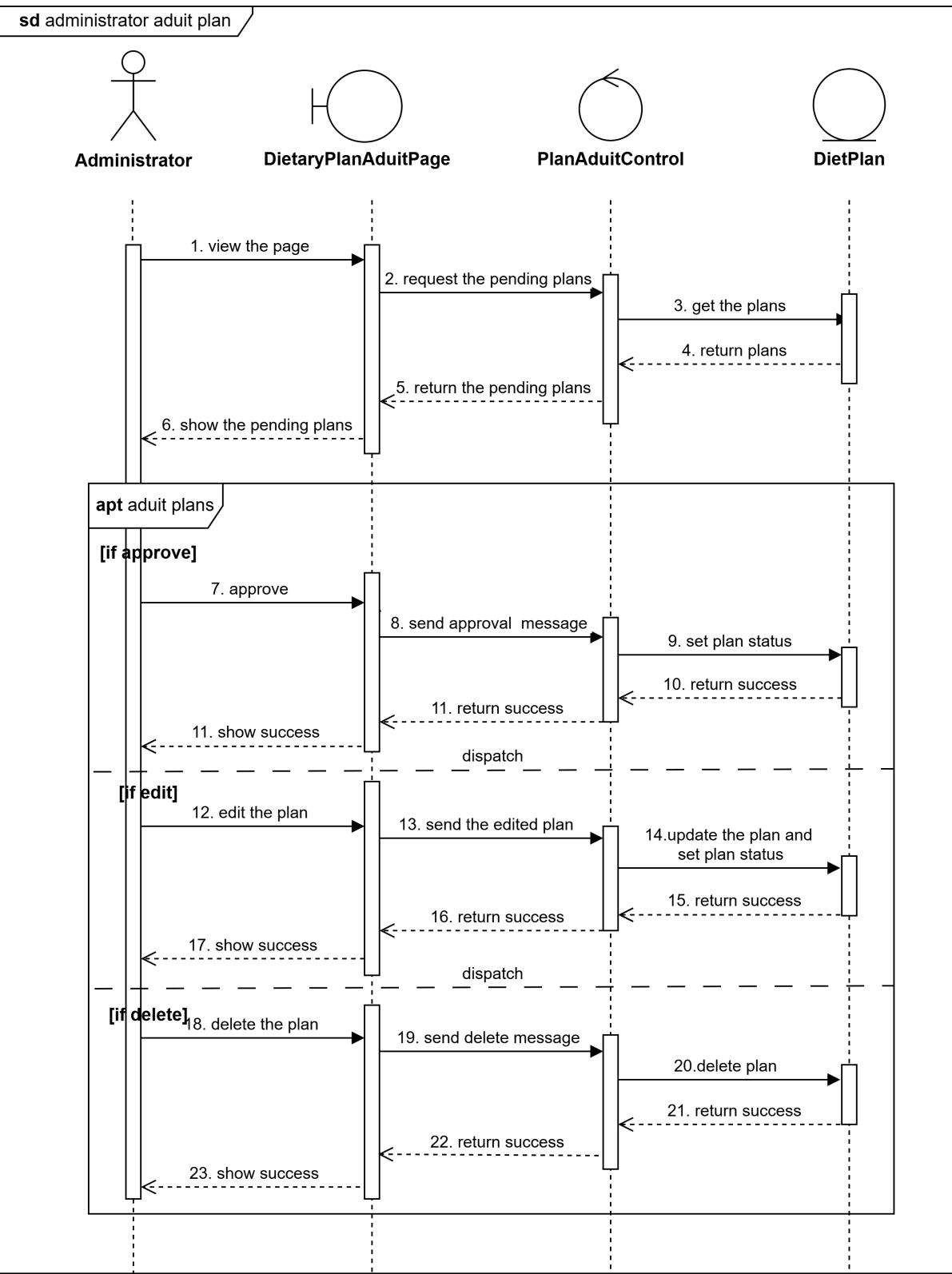
- **Aduit Dietary Plan Interaction**

In this interactive diagram, the Administrator starts by accessing the DietaryPlanAduitPage to manage and audit dietary plans. They can view the page and request a list of pending plans, which the system retrieves and returns, showing the Administrator the plans that are awaiting review.

If administrator choose to approve a plan, they send an approval message, and the PlanAduitControl sets the plan status to approved, returning a success message and updating the display to show the success of the operation.

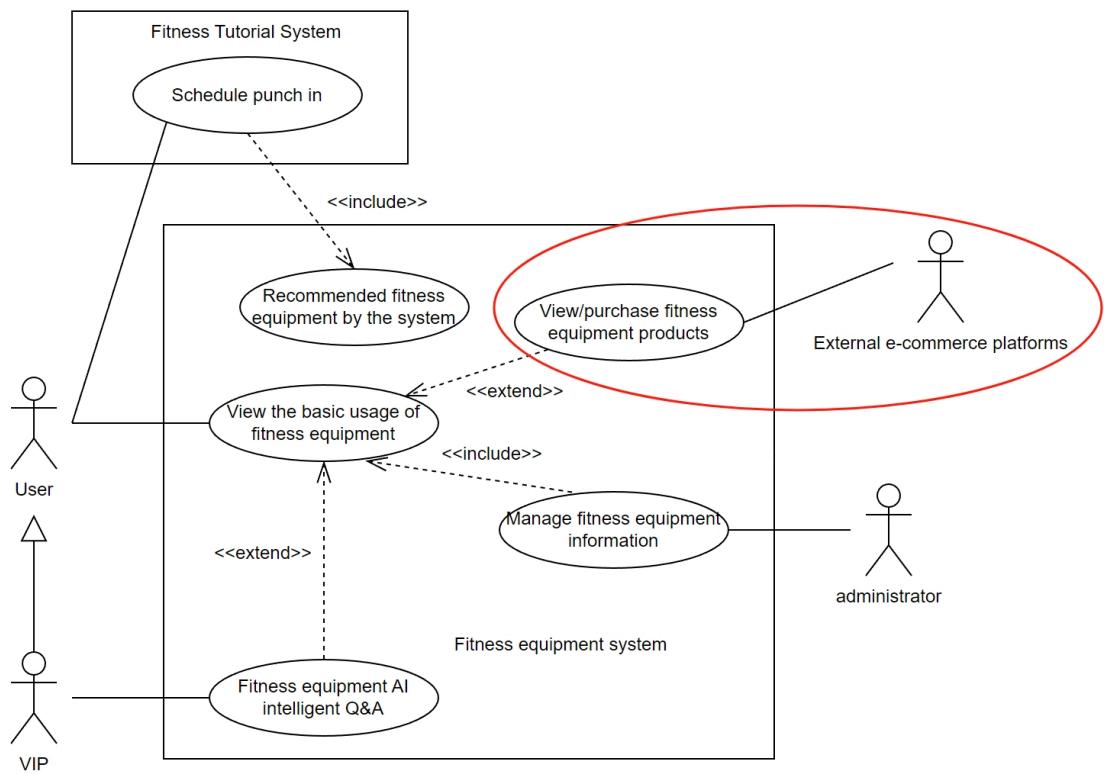
If the Administrator opts to edit a plan, they make the necessary changes and send the edited plan. The PlanAduitControl then updates the plan and sets its status accordingly, returning a success message and updating the display to reflect the successful edit.

In cases where a plan needs to be removed, the Administrator sends a delete message, and the PlanAduitControl deletes the plan from the system, returning a success message and updating the display to confirm the successful deletion.



# 5. Update Requirements

## 5.1 New Case: Fitness equipment product comparison platform

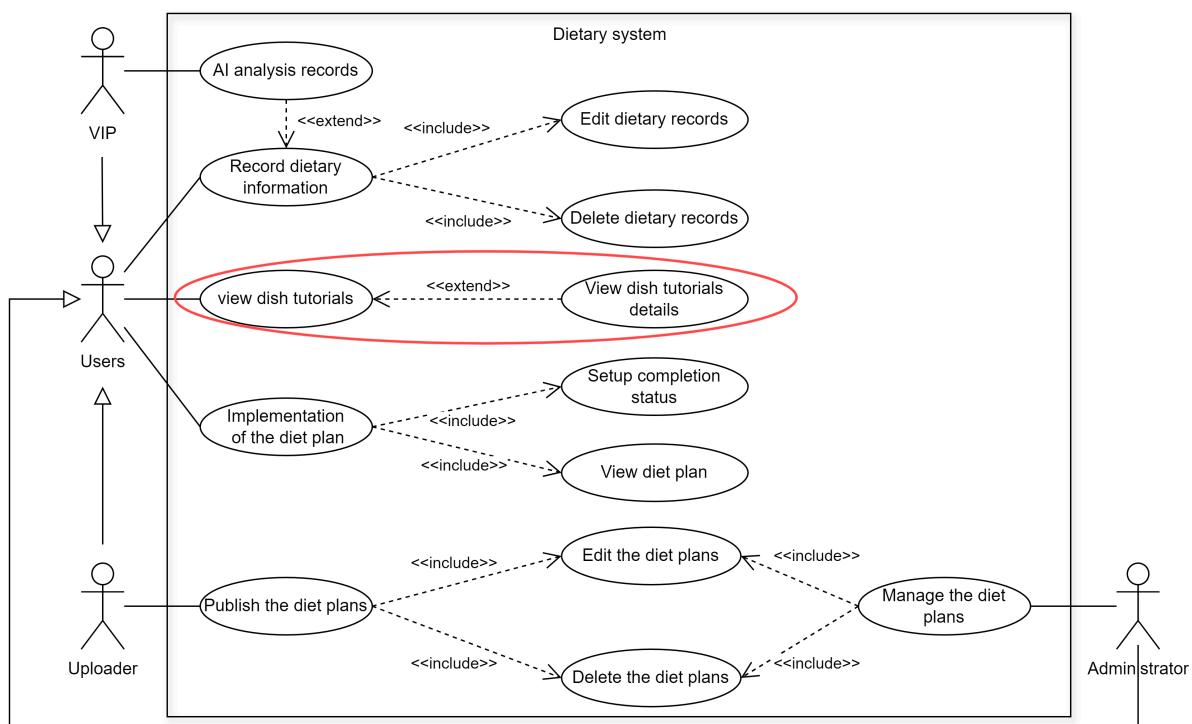


Use Case: Smart Price Comparison for Fitness Equipment

USE CASE	<b>Smart Price Comparison for Fitness Equipment</b>
ID	<b>UC01</b>
Specification	Enables users to compare prices, reviews, and details of fitness equipment across multiple e-commerce platforms through a pop-up window triggered by a shopping cart icon.
Actors	<b>User, E-commerce Platforms</b>
Pre-condition	The user is logged into the fitness app or website. The fitness equipment cards with the shopping cart icon are visible on the "Equipment" page.

USE CASE	Smart Price Comparison for Fitness Equipment
<b>Basic Path</b>	<ol style="list-style-type: none"> <li>1. The user clicks the shopping cart icon on a specific fitness equipment card (e.g., rowing machine).</li> <li>2. The system displays a pop-up window, showing price comparisons, product ratings, reviews, and other details for the selected equipment from various e-commerce platforms.</li> <li>3. The user examines the displayed prices and reviews from multiple platforms.</li> <li>4. The user selects a preferred platform by clicking the provided link within the pop-up window.</li> <li>5. The system redirects the user to the chosen e-commerce platform's product page in a new tab or window.</li> <li>6. The user views additional details and may proceed to complete the purchase on the external platform.</li> </ol>
<b>Alternative Path</b>	<ol style="list-style-type: none"> <li>1. After viewing the price comparison pop-up, the user decides not to proceed with any purchase and closes the pop-up window without selecting a platform link.</li> <li>2. If price or review data from an e-commerce platform is temporarily unavailable, the system displays a message indicating the missing data for that platform, allowing the user to view available options or try again later.</li> </ol>
<b>Post-condition</b>	<ol style="list-style-type: none"> <li>1. If the user completes a purchase on an external platform, the transaction is finalized on that platform, and the fitness app session remains active for continued browsing.</li> <li>2. If the user completes a purchase on an external platform, the transaction is finalized on that platform, and the fitness app session remains active for continued browsing.</li> </ol>

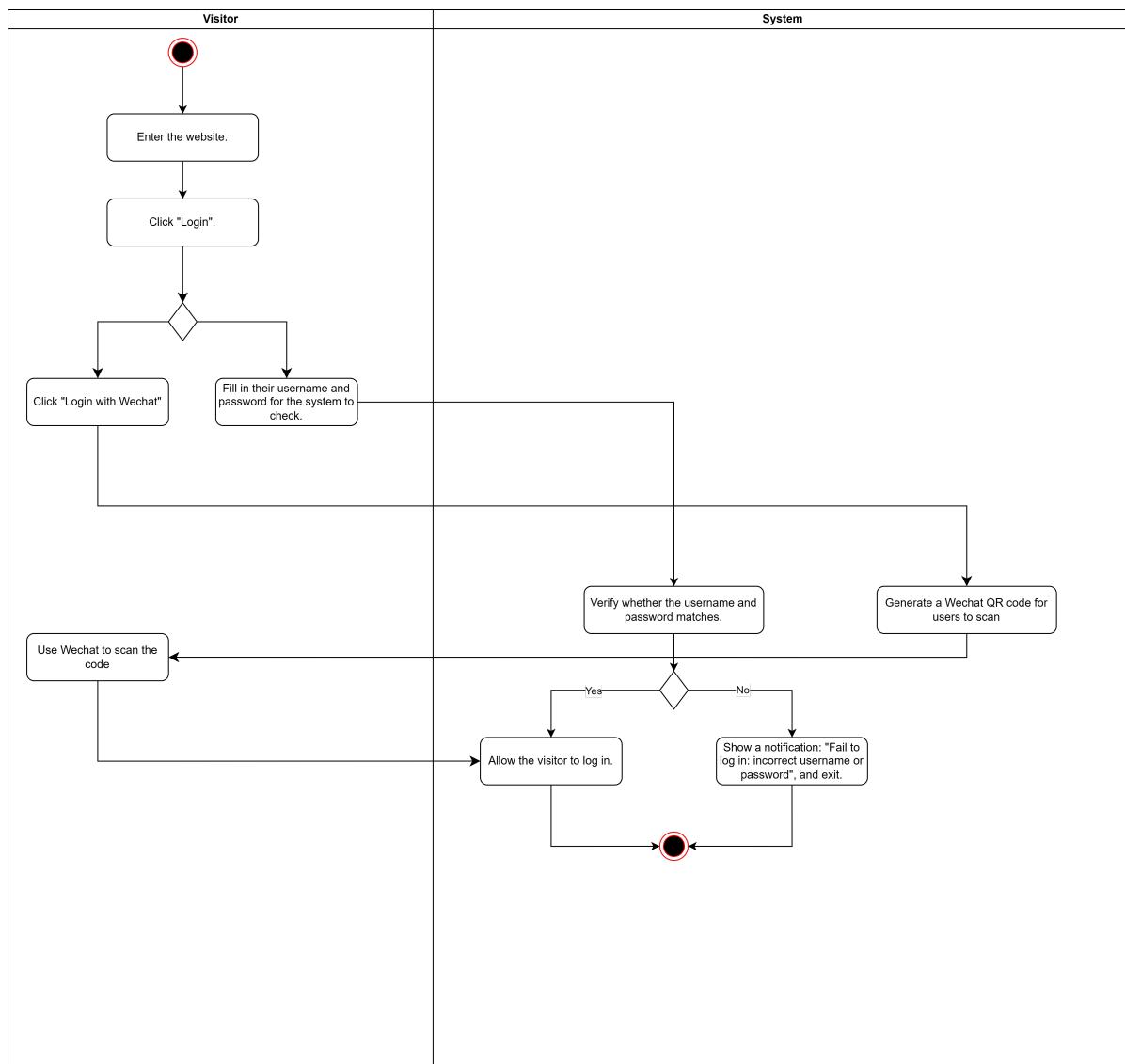
## 5.2 New Case:Dishes preparation tutorial



### Use Case: View dish tutorials

USE CASE	VIEW DISH TUTORIALS
ID	<b>UC02</b>
Specification	Users can browse the tutorials recommended by the system and click on the corresponding card to view the details of the tutorial.
Actors	User
Pre-condition	User successfully logged into the system and views the diet plan page.
Basic Path	<ol style="list-style-type: none"> <li>1. User views the diet plan page.</li> <li>2. System shows the recommended dish tutorials.</li> <li>3. User clicks on the card of the tutorial he is interested in.</li> <li>4. System shows the detail of that tutorial.</li> </ol>
Alternative Path	<ol style="list-style-type: none"> <li>1. If the selected tutorial cannot be displayed:             <ol style="list-style-type: none"> <li>1.1. System displays an error message and logs the issue.</li> <li>1.2. User may choose a different tutorial.</li> </ol> </li> </ol>
Post condition	User successfully views the details of the tutorial.

## 5.3 Updated Case: Login with Wechat



Updated Use Case: **Login**

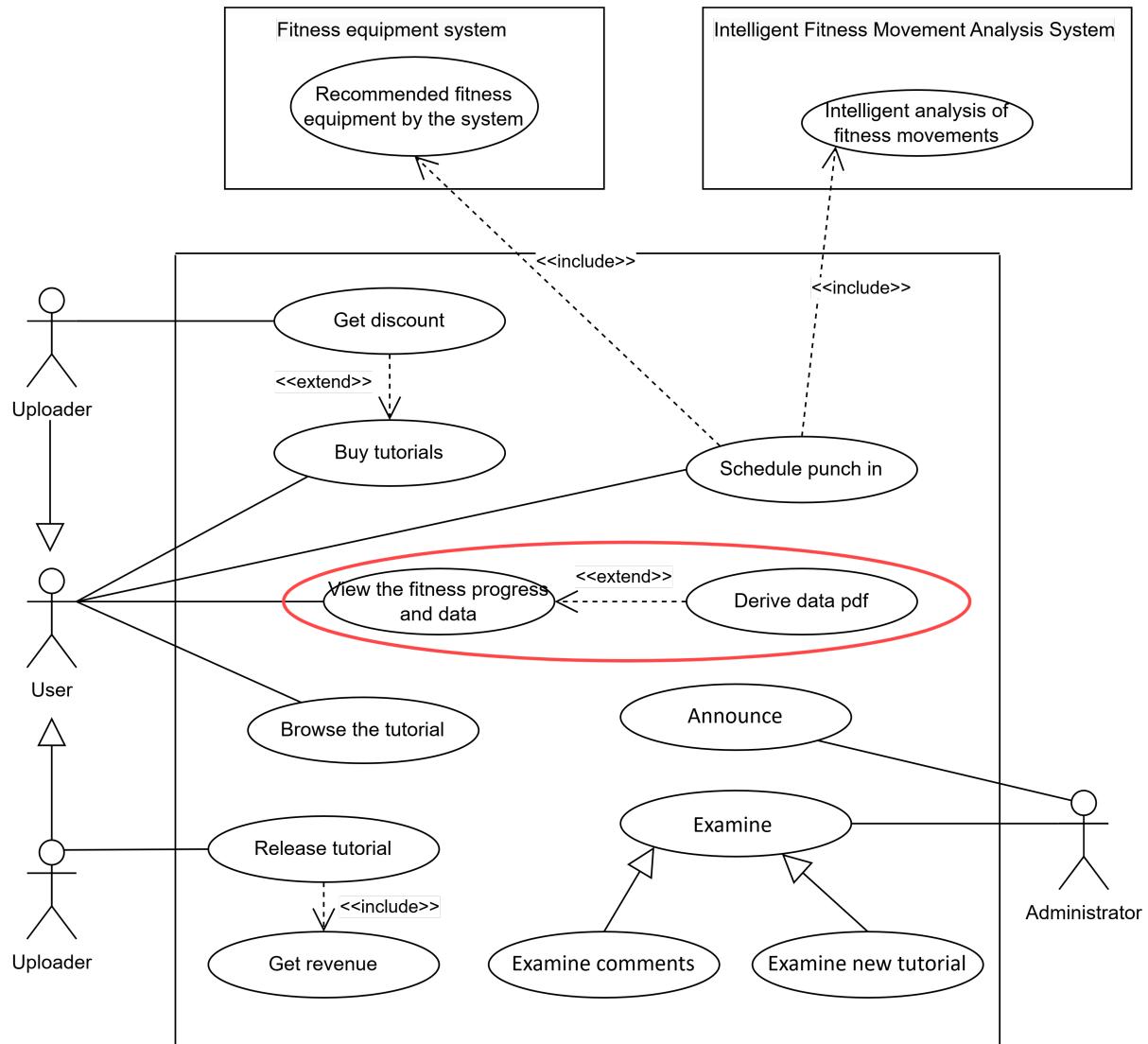
USE CASE	LOGIN
ID	<b>UC03</b>
Specification	Visitors type in their account ID and password to log in to the account.
Actors	Visitor
Pre-condition	Visitors have their own account and click "login" on the website.

USE CASE	LOGIN
<b>Basic Path</b>	<p>1. Visitors enter the website.</p> <p>2. Visitors click "login".</p> <p>3.1 If visitors fill in their username and password for the system to check and click "login".</p> <p>3.1.1 Visitors fill in their username and password.</p> <p>3.1.2 The system verifies the correctness of the username and password</p> <p>3.2 If visitors click "Continue with Wechat".</p> <p>3.2.1 The system pops up a Wechat QR code to scan.</p> <p>3.2.2 The visitor use their Wechat apps to scan the code and get authenticated</p> <p>4. Visitors successfully log in.</p>
<b>Alternative Path</b>	<p>1. Visitors input the username and password that do not match The system will show a notification: "Fail to log in: incorrect username or password", and exit.</p>
<b>Post condition</b>	Visitors enter the system, become users, and can access users' operations.

## 5.4 New Case: View the fitness progress and data

In this section, we have added a new use case diagram where users can **view the fitness progress and the function points of the data section**.

*At the same time, this part of the use case diagram was modified according to the teacher's opinion after the last defense.*



### Use Case: View the fitness progress and data

<b>USE CASE</b>	<b>View the fitness progress and data</b>
<b>ID</b>	<b>UC04</b>
<b>Specification</b>	This use case describes how the user can track their fitness progress and view data analysis in the tutorial section. Users can check their progress over different dates, see various physical indicators, and view/export detailed exercise reports.
<b>Actors</b>	<b>User, System</b>
<b>Pre-condition</b>	1. The user has logged into the system. 2. The user has accessed the tutorial content page.

USE CASE	View the fitness progress and data
Basic Path	<ol style="list-style-type: none"> <li>The user clicks on the data analysis button on the tutorial content page.</li> <li>The system opens a floating window for data analysis and progress tracking.</li> <li>The user selects a date to view specific fitness progress for that day.</li> <li>The system displays the user's fitness data, including relevant physical indicators like weight and body fat percentage, and recent exercise data (exercise time, aerobic, and anaerobic time).</li> <li>The user clicks on an option to export a PDF report for a more detailed analysis alternatively.</li> <li>The system generates and downloads the PDF report.</li> </ol>
Alternative Path	<p><b>No Data Available for Selected Date:</b> 1. If no fitness data is available for the selected date, the system displays a message indicating "No data available for this date."</p>
Post-condition	The user successfully views fitness progress, data analysis, and if the user needs, he/she can exports a detailed report.

## 6. Updated Snapshot

### 6.1 Fitness equipment product comparison platform

This latest UI update introduces a **smart price comparison feature** to help users make informed purchasing decisions across multiple e-commerce platforms. Each fitness equipment card now includes a **shopping cart icon**; clicking it opens a pop-up displaying real-time price comparisons, reviews, and key details from platforms like Taobao, Tmall, and Pinduoduo.

Users can quickly compare options at a glance, and if they want more information, they can click through to the product page on the selected platform to view details or make a purchase. This feature streamlines the shopping experience, making it easier to find the best value for fitness equipment.

The screenshot shows a mobile application interface for a fitness equipment comparison platform. At the top, there is a navigation bar with tabs: KEEP FIT, TUTORIALS, EQUIPMENT, DIET, COACHING, VIP, and ABOUT. A user profile icon is also present in the top right corner.

The main content area displays a grid of three fitness equipment cards:

- TREAD:** An image of a treadmill with a screen. Below it, text says "Used to simulate outdoor running or brisk walking..." and "Keep up the pace! Keep your gaze forward, don't lower your head". It has a 5-star rating and was posted by "Lily" on "2024-9-10".
- BIKES:** An image of a stationary bike with a screen. Below it, text says "Please make sure to adjust the seat height..." and "Safe and effective Practice cycling without leaving home". It has a 5-star rating and was posted by "Jack" on "2024-8-19".
- ROW:** An image of a rowing machine with a screen. Below it, text says "Can simulate the action of rowing on water..." and "Best Tool Essential for rowing practice!". It has a 5-star rating and was posted by "Selina" on "2024-7-18".

At the bottom of the grid, there are navigation buttons: "← Previous", "Next →", and a page number indicator "1 2 3 ... 67 68 Next →".

A floating window titled "Chat with Row" is open on the right side of the screen. It contains a message "Row usage instructions" with two points: "1. Check equipment: Before using the rowing machine, check if the equipment is in good condition..." and "2. Adjust the seat: Adjust the height of the seat according to individual height...". There are "Accept" and "Reject" buttons at the bottom of the message box. The window also has a close button "X" and a message input field "Message Coach Agent" with a send icon.



\$399

**MOKFITNESS M30**

折叠家用 水磁双阻 基础款...

Source Platform: Taobao

The MOKFITNESS M30 Folding Home Rowing Machine combines water and magnetic resistance for a versatile, full-body workout right in your home. Priced at just \$399, this entry-level model offers both smooth water resistance for a realistic rowing experience and magnetic resistance for adjustable intensity, catering to beginners and fitness enthusiasts alike !

...

## Related Product →

\$699

**MR-902-SYT**

The foldable design makes it space-efficient...

From Tmall •



\$1299

**MOKFITNESS-M30-ES**

Equipped with a digital display, you can...

From Taobao •



\$899

**Umay-R5L智能调阻**

Whether you're aiming for cardio endurance...

From Pinduoduo •

## 6.2 Diet Plan Page update

Figure 1:

The screenshot shows a diet planning interface. At the top, there's a navigation bar with links for 'KEEP FIT', 'TUTORIALS', 'EQUIPMENT', 'DIET', 'COACHING', 'VIP', and 'ABOUT'. A user profile icon is also present.

**Current diet plan:**  
**Total Body Building Programme**  
by Keep Fit VIP customisation

You've clocked in 16 days this month.  
You've skipped 1 days.

**Progress:** 51% [Change Plan](#)

**Select date**  
Mon, Aug 17

August 2023 [Select](#) [Previous](#) [Next](#)

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

**2024/10/17 Diet Plan**

**Breakfast(293 kcal)**

- Skimmed milk (1 box)
- Millet mixed grain porridge (1 bowl)
- Egg (1)

**Lunch(547 kcal)**

- Sweet potato rice(1 bowl)
- Scrambled egg with luffa (loofah)
- Stir-fried cabbage

**Dinner(421 kcal)**

- Rice (1 bowl)
- Stir-fried pork with cauliflower
- Scrambled eggs with tomato

**Spinach and Egg**  
Ingredients: eggs\*2, spinach\*1, bundle  
[view the steps](#)

**Beef with Tomato and Egg**  
Ingredients: tomato\*2, beef: 200g, egg\*2  
[view the steps](#)

**Shrimps in Soya Sauce with Eggs**  
Ingredients: eggs\*5, shrimps: some  
[view the steps](#)

← Previous 1 2 3 ... 67 68 Next →

Figure 2:

The screenshot shows a diet planning interface similar to Figure 1, but with more detailed meal plans and steps.

**Current diet plan:**  
**Total Body Building Programme**  
by Keep Fit VIP customisation

You've clocked in 16 days this month.  
You've skipped 1 days.

**Progress:** 51% [Change Plan](#)

**Hello, Lisa!**

**1/17 Diet Plan**

**Beef with Tomato and Egg**  
(293 kcal)  
Ingredients: tomato\*2, beef: 200g, egg\*2  
[view the steps](#)

**Steps:**

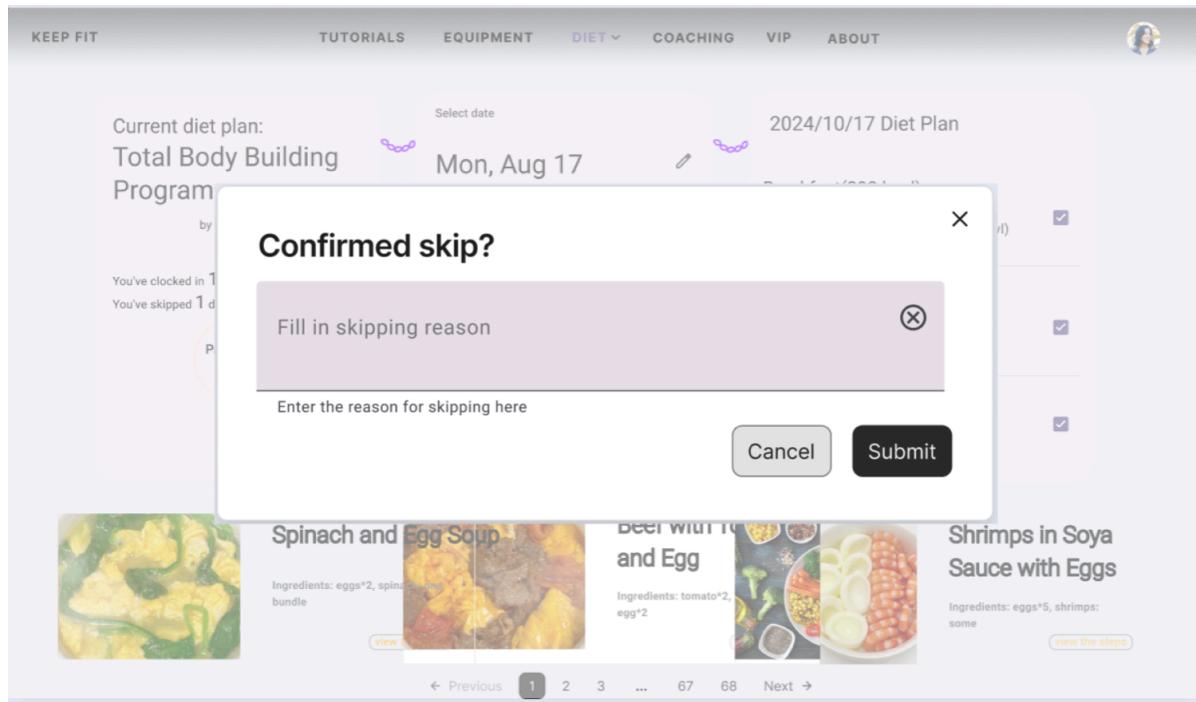
- 1 Cut the beef into pieces, a little oil, a spoonful of soy sauce, a spoonful of starch, marinate for about 10 minutes
- 2 Fry two eggs in a pan with little oil.
- 3 Stir fry the beef in less oil.
- 4 Stir fry tomatoes in a little oil until sandy
- 5 Pour in the beef and eggs and cook for two minutes, add a spoonful of soy sauce, moderate salt

**Spinach and Egg**  
Ingredients: eggs\*2, spinach\*1, bundle  
[view the steps](#)

**Shrimps in Soya Sauce with Eggs**  
Ingredients: eggs\*5, shrimps: some  
[view the steps](#)

← Previous 1 2 3 ... 67 68 Next →

Figure 3:



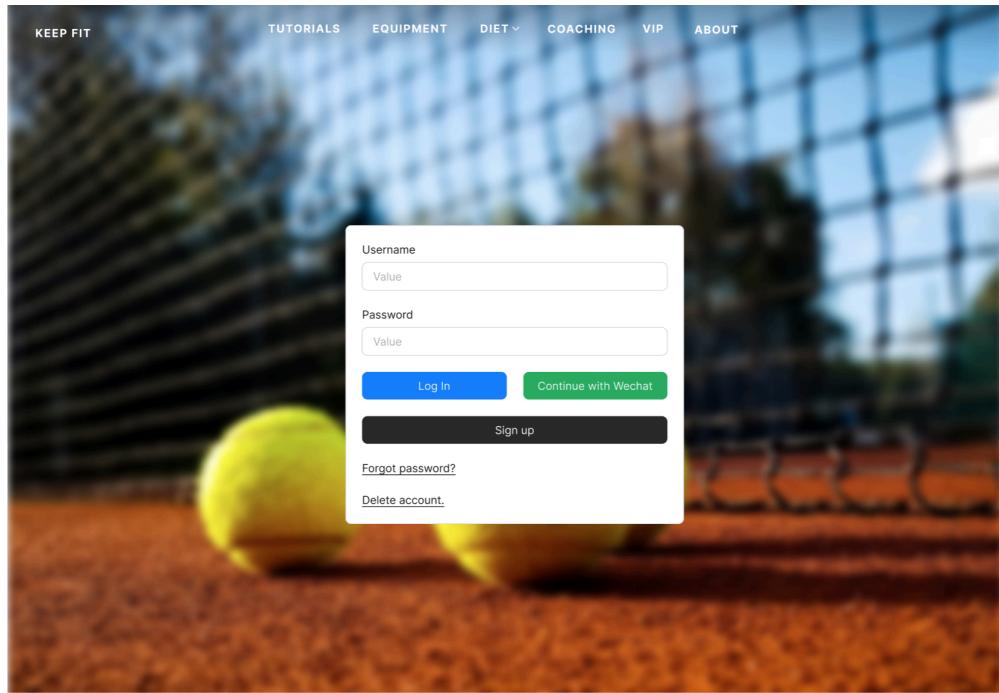
We modified the statistics function on the basis of the previous one to count the number of days of completed plans and the number of days of skipped plans in the month, and added a progress icon for the user to view intuitively. In addition, we also marked the calendar with a rose-red colour block to identify the 'Completed Plans', and a hollow block to identify the 'Skipped Plans' on the calendar to enhance the interactivity of the system (Figure 1).

The new diet plan page introduces the `dishes preparation tutorial` function, which provides users with tutorials on how to make some healthy and simple dishes. At the bottom of the diet plan page, the system will display the recipes that are intelligently recommended by the system for the user, and the user can click the button at the bottom right corner of the corresponding recipes to view the details, including the amount of ingredients, cooking steps, and picture instructions (Figure 2).

It was mentioned before that when the user chooses to skip the plan, the system has to request the reason for skipping from the user, and here we use a dialogue box to collect the information(Figure 3).

## 6.3 Login & Registration Interface Snapshot

In this latest UI interface, **Continued with Wechat** button is added so that users get to choose to **login with their Wechat account**. By clicking the button, they can see a QR code popping up, scanning which with their Wechat app can allow them get authenticated with their Wechat accounts without registering and typing their login information manually. This feature is provided due to the popularity of this communication app and the motivation of providing a easy-to-use platforms to all the users of this website.



## 6.4 Fitness Tutorial Section Update Snapshot

In this section, we have added new progress tracking and data analysis functions. A data analysis button is added in the original tutorial content page, the user can click a small floating window, in this small page, the user can see the fitness progress and task schedule of different days by selecting the date, and in the following data report, will simply present some relevant physical indicators, such as weight, body fat rate, etc., will also present the user's recent exercise data, including the exercise time and aerobic exercise time, anaerobic exercise time, more detailed data analysis report can the user view by exporting PDF report. The following are the button addition of the original tutorial content interface and the small window for progress tracking and data analysis functions.

**KEEP FIT**

**Hello, Lisa!**

Date: 08/17/2023

Pending    New    Completed

August 2023

Mean index

Month	Weight	Body fat rate
JAN	60kg	15%
FEB	58kg	20%
MAR	55kg	22%
APR	54kg	16%
JUN	53kg	19%
JUL	55kg	23%
AUG	58kg	20%
SEP	53kg	19%
OCT	53kg	19%

Recent exercise time

Date	Duration	Intensity
Aug 17, 2023	1.2h	moderate
Aug 18, 2023	1.5h	moderate
Aug 19, 2023	0.8h	moderate
Aug 20, 2023	1.1h	moderate
Aug 21, 2023	0.9h	moderate
Aug 22, 2023	1.3h	moderate
Aug 23, 2023	0.7h	moderate
Aug 24, 2023	1.0h	moderate
Aug 25, 2023	1.4h	moderate
Aug 26, 2023	0.9h	moderate
Aug 27, 2023	1.1h	moderate
Aug 28, 2023	0.8h	moderate
Aug 29, 2023	1.2h	moderate

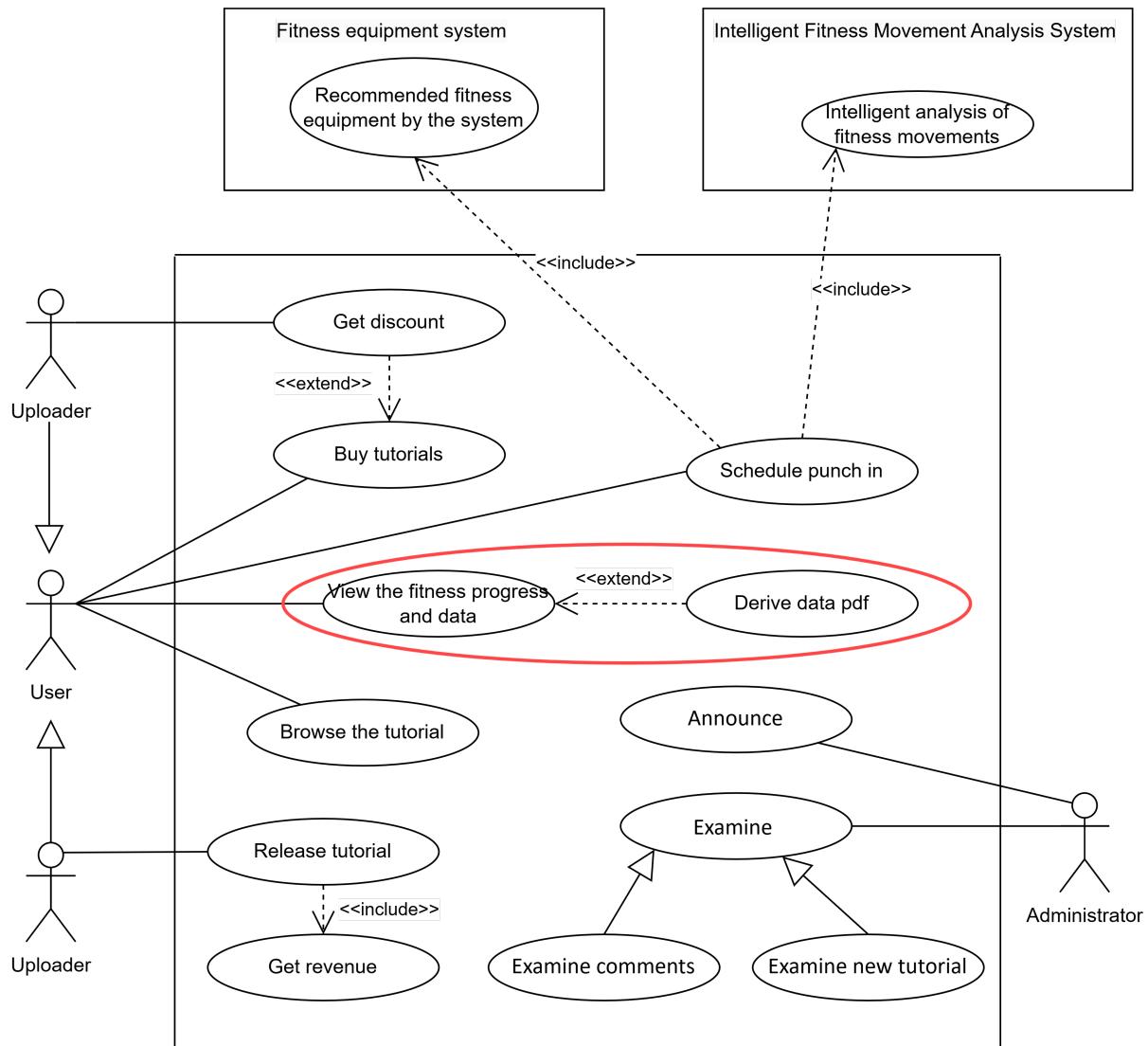
View details    Export as PDF

CHANGE PLANS

**DATA ANALYSIS**

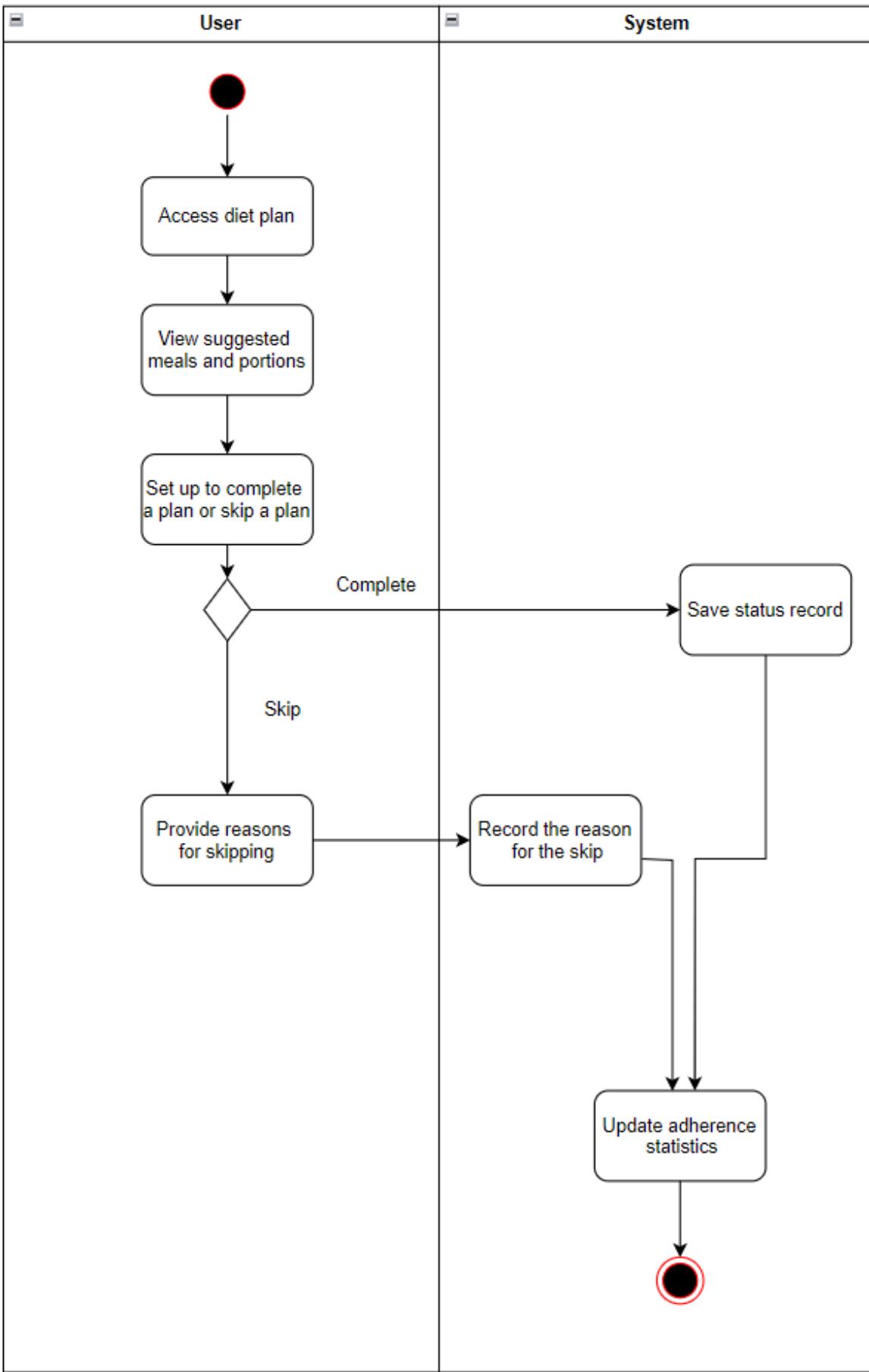
## 7. Correction of errors

The use case diagram of the tutorial part has been modified according to the teacher's opinion after the last defense. The detailed latest picture and introduction have been put in section 5.4.

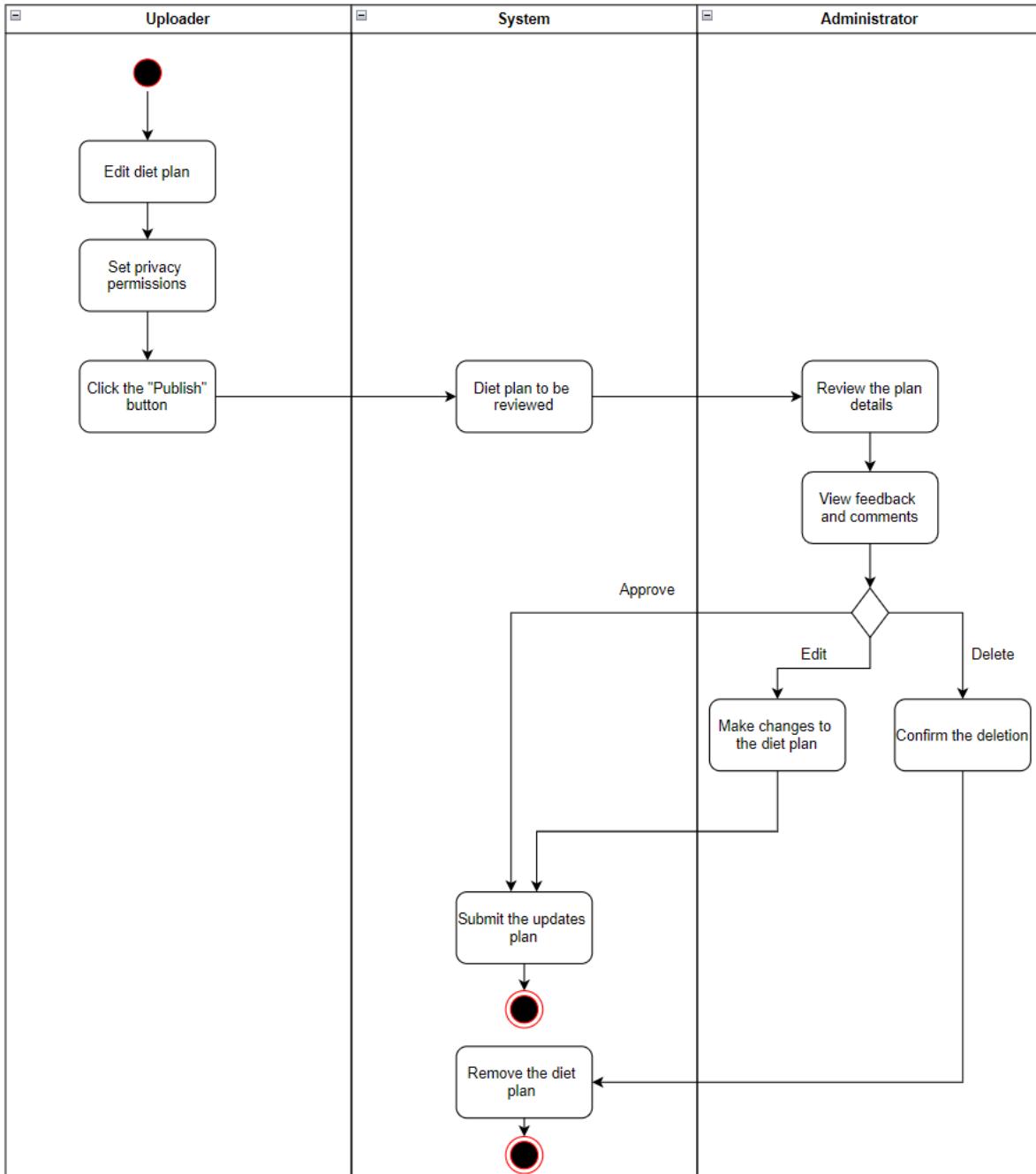


For the problems pointed out by the instructor in the midterm defence, we have corrected our previous work.

For the Diet module `Implement the diet plan activity diagram in Assignment 1`, we misused the synchronisation bar (black solid line) in the previous assignment. In fact, set `skip Plan` and set `complete Plan` are two parallel paths that do not need to reach the `update adherence statistics` at the same time, so we change the activity diagram to a parallel relationship as follows:



Similarly, in the `Publish diet plan` activity diagram, there is only a parallel relationship between `approve` and `edit`, not a synchronous relationship, and the updated activity diagram is as follows:



## 8. Open Questions

For the next phase of the intelligent fitness platform, here are some key challenges to explore and design tasks to tackle:

### 8.1 Challenges

- User Behavior Analysis and Personalized Recommendations**

Develop refined analytics to understand user fitness preferences and habits, enabling personalized recommendations. This requires optimized data collection, analysis models, and recommendation algorithms.

- Data Privacy and Security**

As the platform gathers more user data, ensuring data privacy compliance is essential, especially with sensitive health and body metrics. Implement secure storage, encrypted transmission, and strict access control measures.

- AI Movement Recognition and Feedback Accuracy**

Improve AI accuracy in recognizing and providing feedback on movement quality, ensuring consistency across different environments, whether indoors or outdoors.

- **Multi-Device Compatibility and Synchronization**

Enable platform access across various devices (e.g., smartphones, tablets, smartwatches) with real-time data synchronization, enhancing the user experience.

- **User Engagement and Retention**

Increase user retention through effective incentives and gamification, such as rewards, achievements, and social interaction features that encourage frequent engagement.

## 8.2 Design Tasks

---

- **Personalized Training Plan Optimization Module**

Design and implement a module that auto-generates and adjusts training plans based on users' history, fitness goals, and current health metrics, including adaptive load adjustments and suggested training times.

- **Advanced Data Analysis Report Features**

Expand report capabilities with more analytical dimensions (e.g., monthly trends, muscle growth rate), including intuitive charts and visualizations to help users understand their fitness progress.

- **Social Interaction and Challenges**

Introduce social interaction features such as friend challenges, group fitness plans, and a leaderboard to encourage user interaction and engagement.

- **Multilingual and Localization Support**

Design support for multiple languages and adapt the platform to fit local fitness habits and preferences, including interface translation and regional customization.

- **Expanded AI Assistant Functions:** Extend the AI assistant's capabilities to include dietary suggestions, sleep management, and other health aspects to enhance users' overall fitness and wellness.

- **Dynamic Load Adjustment and Real-Time Feedback**

Create a real-time load adjustment system based on metrics like heart rate and respiratory rate, providing dynamic intensity adjustments and guidance to help users avoid injuries and overtraining.

- **Social Media and Content Sharing Features**

Design features that allow users to share fitness data, achievements, and specific training content on social media, promoting fitness achievements and encouraging user sharing.

- **User Feedback and Continuous Improvement Loop**

Establish a feedback collection mechanism for users to report issues or suggest improvements, and periodically optimize the system based on this feedback to keep functionality up-to-date.

## 9. The Annotated References

---

- [https://www.tutorialspoint.com/software\\_architecture\\_design/index.htm](https://www.tutorialspoint.com/software_architecture_design/index.htm)

This comprehensive guide covers essential aspects of software architecture and design, providing insights into structuring applications for scalability, efficiency, and maintainability. For the KeepFit project, these principles are fundamental to creating a robust architecture capable of supporting features like personalized diet plans, workout routines, and equipment management. The guide explains architectural patterns (such as layered, client-server, and microservices), each with unique benefits for structuring modular applications. For KeepFit, a layered architecture could separate user interaction from data processing and storage, improving modularity and simplifying development and debugging.

Additionally, the guide emphasizes the importance of design quality attributes like performance, security, and scalability, which are critical in ensuring that KeepFit operates smoothly as user numbers grow. By adopting principles like separation of concerns and utilizing design patterns for efficient component interaction, we can create a system that handles complex operations, such as real-time data updates for workouts and diet recommendations, without sacrificing responsiveness.

- <https://zhuanlan.zhihu.com/p/109655171>

In the context of the KeepFit project, understanding UML class diagrams is essential for establishing a structured and maintainable architecture that defines relationships among key components like user profiles, workout routines, diet plans, and equipment management. This article's focus on class diagrams aligns directly with KeepFit's need to clearly model the static structures and interdependencies across its diverse features. By applying principles from this guide, we can accurately represent the class hierarchies and relationships within KeepFit, clarifying how components such as User, DietPlan, WorkoutRoutine, and Equipment interact, either through inheritance, associations, or dependencies. Notations covered in the article, such as "+" for public and "-" for private members, support the application's encapsulation needs, ensuring secure, controlled interactions across modules.

Moreover, understanding relationships like aggregation, composition, and dependency enables us to capture "has-a" and "uses" dynamics essential to KeepFit's core functionality—for example, connecting Equipment with WorkoutRoutine, where both components play distinct but interlinked roles. The article also provides a foundation for extensibility, ensuring that KeepFit's design can accommodate future enhancements, such as adding progress tracking or other fitness-related features, with minimal disruption to the core structure. Overall, incorporating UML class diagram principles enables the KeepFit team to plan, communicate, and manage complex interactions effectively, resulting in a scalable, user-centered fitness platform that remains adaptable and efficient as it evolves.

- [https://design-patterns.readthedocs.io/zh-cn/latest/read\\_uml.html](https://design-patterns.readthedocs.io/zh-cn/latest/read_uml.html)

This article is a tutorial blog on drawing UML class diagrams. Through the examples in the article, we gained a deeper understanding of the six types of relationships between classes and successfully applied them to our class design. For instance, in the dietary module class design, the `AIAnalysis` class serves as an attribute of the `DietRecord` class, and the relationship between them is a composition, represented by a solid diamond shape.

- <https://juejin.cn/post/6965684798174167076>

This article is a tutorial blog on drawing UML sequence diagrams. In the article, the author distinguishes between synchronous messages, asynchronous messages, return messages, and self-messages, and also introduces logical control structures such as alternative and choice combinations. Through this content, the article helps us gain a deeper understanding of the

structure and usage scenarios of sequence diagrams, which we have successfully applied to our own sequence diagrams.

- <https://mi.talkingdata.com/report-detail.html?id=544>

TalkingData The report provides an in-depth analysis of the portraits of Chinese sports and fitness people, covering the age, behavior habits and geographical distribution of users. Data show that the use rate of fitness APP is gradually increasing among young people, mainly preferring aerobic and strength training. The report also points out the time period that users exercise frequently, common functions, and participation differences between cities, and analyzes user motivation, common exercise environments, and the brand's marketing strategies among different fitness groups.

This report offers valuable guidance for designing new fitness platform features and effectively displaying exercise data. By analyzing user behaviors, preferences, and peak usage times, it suggests optimizing app interfaces with features that align with users' motivations and exercise routines, such as enhanced tracking for cardio and strength workouts. The report's insights on popular app functionalities and regional activity differences can also inform feature prioritization and data visualization design, helping to create an engaging, user-centered experience tailored to fitness app demographics.

## 10. Contributors

---

In the implementation process of the project, all members actively discuss and participate seriously, according to their respective interests and ability, to complete the task on time and with high quality. Harmonious team atmosphere, smooth communication and high efficiency. The division of labor and contribution of our group are like the following:

Members	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7	Part 8	Part 9	SCORE WEIGHT
2154286 Weicheng Zheng	√	√		√		√	√	√	√	25%
2253744 Juekai Lin	√		√	√	√	√		√	√	25%
2153085 Lixin Ma	√	√		√		√	√	√	√	25%
2154284 Junhao Yang	√		√	√		√		√	√	25%