FitTrack Software development instructions

1.Introduction

1.1 Purpose of writing

The function of this project is to realize the separation of the front and rear end architecture to improve the development flexibility and the scalability of the system. For our team, we aim to improve the programming technology and Java ability, master the front and rear end development skills, and lay a solid foundation for the future work efficiency。

1.2 Project background

Software name: FitTrack

Project task proposer: Teacher Qi-feng Lin

Project developer: No Fantasy

1.3 References

Teacher Registration System Requirements Specification
Buy together Required Specifications

2.General description

2.1 Goals

2.1.1 Development intention

Our goal is to improve the fitness effect of users through technological innovation, and to make health management simpler and more efficient.

2.1.2 Application goals and scope

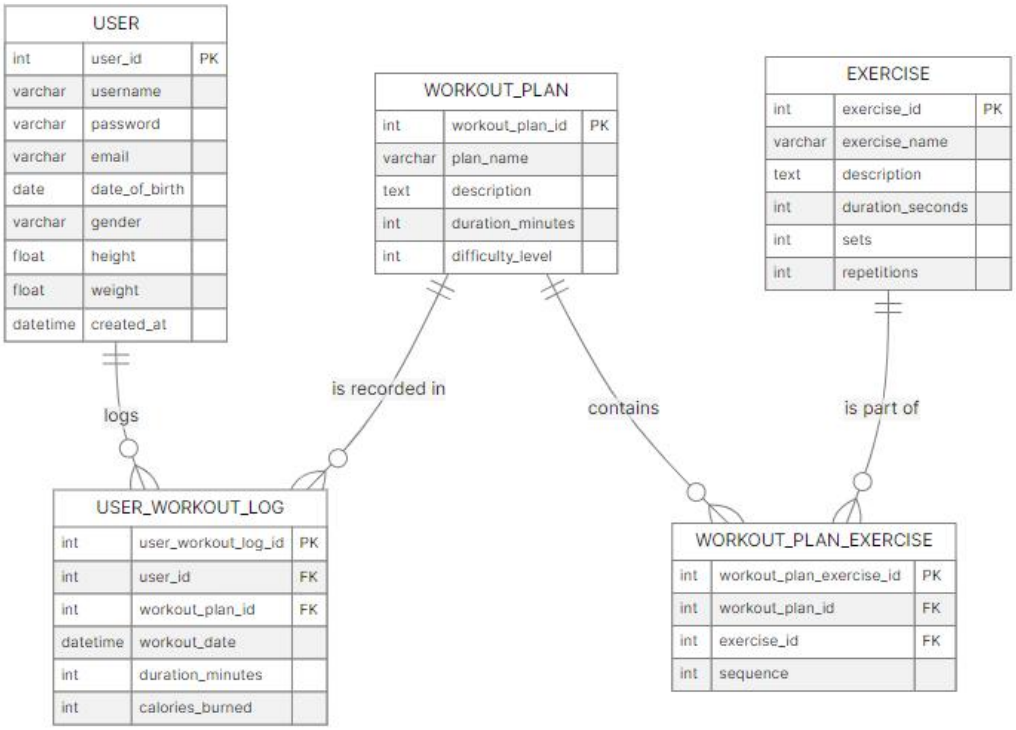
Including all kinds of fitness enthusiasts, from beginners to experienced fitness practitioners. People who focus on health and want to manage their weight or improve their lifestyle. Sports teams, fitness clubs, and group event organizers, etc.

2.1.3 Product prospects

FitTrack It aims to become the preferred platform for users' fitness and health management, providing comprehensive and personalized solutions to help users to easily track exercise progress, set goals, and obtain professional health advice. FitTrack It has the potential to become a leading platform in the fitness and health management fields. Achieving these objectives requires continuous investment and flexible strategy adjustments to accommodate the rapidly changing market environment.

3. Specific requirements

3.1 Class diagram



3.2.1 Availability

(1) System stability

Architecture design: The front and rear end separation architecture is adopted to ensure that the front end and back end can be developed and deployed independently, reducing the risk of overall system failure.

Load balancing: User requests can be dispersed through the load equalizer to ensure the stability of the system at high concurrency conditions.

Monitoring and alarm: implement the monitoring system, track the system performance and health status in real time, and find and handle abnormal situations in time.

(2) Failure recovery ability

Data backup: back up the user data regularly to prevent data loss.

Redundant design: Use redundant servers and databases to switch quickly when a component fails to ensure continuous service availability.

Disaster recovery solution: realize multi-place deployment to prevent comprehensive service interruption caused by single point of failure.

(3) User access ability

High availability architecture: containerization and automatic expansion using Docker and Kubernetes to ensure that the system can dynamically adjust resources to traffic requirements.

Quick response time: Optimize API design and use caching mechanisms such as Redis to improve data access speed and reduce user waiting time.

Global Access: Increase access to users in different regions through the Content Distribution Network (CDN) to ensure smooth access to users around the world.

(4) Maintenance and management

Continuous Integration and Continuous Deployment (CI / CD): Ensure the stability of new releases by automating testing and deployment processes, reducing decreased availability due to updates.

User feedback mechanism: establish user feedback channels, timely collect users' problems and suggestions in the process of use, and quickly fix the bugs that affect the usability.

Documentation and support: Provide detailed user manual and technical support to help users quickly solve problems in use.

3.2.1 Security

OAuth 2.0 is adopted for user authentication, and data transmission security is guaranteed through TLS encryption to ensure user privacy and data security.

4 interface prototype

It’s not implemented yet, it will be added later.

5.Function description and acceptance verification standards

5.1 Specific function description

(1) User registration and login

Registration: Users can register through their email or social media account and fill in the basic information (user name, password, height, weight, etc.).

Login: It provides a secure authentication mechanism, including OAuth 2.0 support, and users can log in using a social media account or email.

(2) User's personal data management

Personal information: Users can view and edit their personal data, including height, weight, fitness goals, etc.

Health information record: Users can record health indicators (such as body fat rate, blood pressure, etc.) and view historical data.

(3) Fitness target setting and tracking

Objective creation: Users can set fitness goals (such as weight loss, muscle gain, endurance training, etc.) according to their personal needs.

Progress tracking: the system automatically tracks the user's exercise progress, and provides visual data (such as charts and progress bars) to show the completion of the target.

(4) Exercise record management

Exercise log: Users can record the type, duration, calories consumed and other information of each exercise.

Exercise plan: Users can create and manage the exercise plan according to their own needs, and the system will remind users to execute it according to the plan.

(5) Nutrition and diet management

Food database: provides a wide range of food information, users can search and choose food, view the nutritional composition.

Diet log: Users can record their daily diet, systematically calculate calories and nutrition intake, and help users keep a balanced diet.

(6) Data analysis and reporting

Health report: The system generates periodic health reports, analyze users' exercise and diet conditions, and provide suggestions for improvement.

Trend analysis: Show users' health trends (such as weight change, exercise frequency, etc.) through charts and statistics.

(7) Third-party service integration

Strava API: Automatically import users' movement data, integrate external device data, and improve user experience.

Wearable device support: Connect to smartwatches and other fitness trackers to synchronize data in real time.

(8). Community and social functions

User community: Build a user community to support discussion, experience sharing, and mutual motivation.

Achievement sharing: Users can share their fitness achievements to social media and increase the interactivity of the platform.

(9) Security and privacy protection

Data encryption: Use TLS to encrypt user data to ensure the security of the information during transmission.

Privacy Settings: Users can manage the visibility of their personal information and select which data to share.

(10) System setting and help support

Application Settings: Users can customize their application for notifications, interface topics, and other preferences.

Help center: provide user guide, FAQ and technical support.

5.2 Input and output format

FitTrack The input and output format adopts JSON structure, which is simple and easy to understand to facilitate data interaction between the front and back ends.

5.3 Interface acceptance criteria

Interface acceptance criteria are used to ensure expected interactions between the modules of FitTrack software and external services and ensure the stability and reliability of the system. The following are the acceptance criteria for the main interface:

1. User registration and login interface

functionality:

Can successfully create a new user, return the user ID.

Ability to verify the user identity and return the authentication token.

function:

The registration request should be responded to within 2 seconds.

The login request should respond within 1 second.

safety:

User passwords shall be stored in an encrypted manner.

The interface shall support the HTTPS.

error handling:

Invalid email format, password does not meet the requirements and other input errors should return the specific error information.

2. User personal data management interface

functionality:

Be able to successfully update the user's personal information (such as height, weight).

Be able to successfully access the user's personal data.

function:

The update request should be completed within 2 seconds.

The acquisition request should be completed within 1 second.

safety:

Only let the user himself or an authorized person are allowed to access their personal information.

error handling:

Invalid user ID or missing information should return a specific error message.

3. Fitness target setting and tracking interface

functionality:

Ability to successfully create, update, and delete fitness targets.

Can successfully obtain all the target information of the user.

function:

The target creation and update request should be completed within 2 seconds.

safety:

The interface shall ensure the user of the management for their target information.

error handling:

Input data format is incorrect or expired destination should return specific error messages.

4. Exercise the record management interface

functionality:

Be able to successfully record the exercise logs.

Can successfully obtain the user's exercise history.

function:

The exercise recording request should be completed within 2 seconds.

safety:

Users are only allowed to access their exercise records.

error handling:

Exercise records The missing necessary fields should return specific error messages.

5. Interface between nutrition and diet management

functionality:

Ability to successfully record dietary information.

Be able to successfully obtain the user's food records.

function:

The request for diet records should be completed within 2 seconds.

safety:

The users are only allowed to access their diet records.

error handling:

The food information entered should not return the corresponding prompt in the database.

6. Data analysis and reporting interface

functionality:

Be able to successfully generate user health reports.

Can provide the results of the trend analysis.

function:

The report generation request should be completed within 3 seconds.

safety:

Only only only only have access to their reported data.

error handling:

The request invalid reporting period should return a specific error message.

5.4 Functional acceptance criteria

The functional acceptance criteria are used to assess whether the functions of the FitTrack software meet the design requirements and user requirements. The following are the acceptance criteria for the main functions:

1. User registration and log in

functional completeness:

Users are able to successfully register a new account and receive a confirmation notification.

Users can log in successfully with the correct credentials.

When the login fails, the appropriate error message should be returned.

User Experience UE:

The registration and login interface shall be simple and easy to use, and support the password visible / hidden function.

Provide the password reset function, users can retrieve the password through the mailbox.

2. User's personal data management

functional completeness:

Users are able to view, edit, and save personal information (such as height, weight).

Users are able to add or modify health indicators (such as body fat ratio).

data validity:

The input height, weight and other data shall be within the reasonable range (such as height between 50250 cm)。

User Experience UE:

After updating the information, display the successful prompt and return the latest information.

3. Fitness target setting and tracking

functional completeness:

Users are able to create, update, and delete fitness targets.

The system shall automatically track and update the target progress.

data validity:

The target value and end date shall be logical (if the end date cannot be earlier than the current date).

User Experience UE:

Target status should be clearly displayed, supporting chart or progress bar display.

4. Exercise record management

functional completeness:

Users were able to record information about each exercise, including type, duration, and calories consumed.

Users are able to view and filter the historical workout records.

data validity:

Exercise duration and calories consumed should be positive.

User Experience UE:

Exercise records should support fast entry and editing functions.

5. Nutrition and diet management

functional completeness:

Users are able to search for and record food intake and calculate calories and nutrients.

The system shall provide a food database to support the addition of new foods.

data validity:

Food information and the calories and nutritional ingredients should be accurate。

User Experience UE:

Users can easily view the daily diet summaries.

6. Data analysis and reporting

functional completeness:

The system shall regularly generate user health reports and analyze exercise and diet data.

Provide data trend analysis (e. g. weight change, exercise frequency, etc.).

data validity:

The report should accurately reflect the health status based on the users' real data。

User Experience UE:

The report interface shall be intuitive and support chart and data visualization.

7. Community and social functions

functional completeness:

Users are able to participate in community discussions and share experiences and outcomes.

The system shall support user interaction (e. g., likes, comments).

User Experience UE:

The community interface should be friendly, easy to navigate and interactive.