

**计算机与信息技术学院**

**课程设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 课程名称 | 软件系统设计 |  |
|  | 设计名称 | 健身智囊助手 |  |
|  | 专业班级 | 计科2104 |  |
|  | 学 号 | 21851053 |  |
|  | 姓 名 | 刘佳骏 |  |
|  | 任课教师 | 田晋坤 |  |
|  | 日 期 | 2024-06-27 |  |

**《软件系统设计》课程考核说明**

**一、考试对象**

2021级计科2101-2106班

**二、考核方式**

课程设计报告 + 答辩。

**三、考试内容**

基于大语言模型（LLM）完成一个智能化系统，使用的技术包括但不限于Python、Langchain等。其中，大语言模型可以采用现有的第三方服务商提供的模型（如：通义千问、千帆等），也可以使用开源模型。

**四、技术要求**

1、要求使用Python、Langchain框架。

2、项目文件命名：学生姓名的拼音首字母-网站英文名称，名称与多个英文单词之间用“-”连接，如李明的手机销售网站命名为：lm-phone-seller

3、单文件组件命名要符合Pascal命名规范，变量、文件及文件夹命名要符合规范，尽量做到见名知意。

**五、提交物**

项目源码、设计说明书电子版、设计说明书纸质打印版、答辩纪录。

**六、期末成绩构成**

由项目成绩和答辩成绩组成，其中项目成绩60分，答辩成绩40分

项目成绩评分标准如下：

|  |  |
| --- | --- |
| **综合/单项类评分** | **分值** |
| **1 提交物** | **10** |
| 1.1 内容完整性 | 5 |
| 1.2 目录结构 | 5 |
| **2 代码** | **20** |
| 2.1 项目、代码命名规范 | 5 |
| 2.2 代码编写规范性 | 5 |
| 2.3 代码可读性 | 5 |
| 2.4 技术全面 | 5 |
| **3 程序演示** | **30** |
| 3.1 结果正确，功能完整 | 15 |
| 3.2 页面设计美观 | 15 |

**《软件系统设计》课程考核成绩**

|  |  |
| --- | --- |
| 项目成绩（60分） |  |
| 答辩成绩（40分） |  |
| 期末总成绩 |  |
| **答辩评语：**  提交物内容（完整、较完整、基本完整、不完整），目录结构（正确、较正确、基本正确、不正确）；项目、代码命名（符合规范、较符合规范、基本符合规范、不规范），代码注释（完整、较完整、基本完整、不完整），可读性（好、较好、尚可、差）；技术应用方面（全面、较全面、基本合格、有欠缺）；程序功能（完整、较完整、基本完整、不完整），页面（美观、较美观、设计尚可、不美观），演示过程（顺畅、较顺畅、基本顺畅、不顺畅），问题回答（正确、较正确、基本正确、错误较多），思路（清晰、较清晰、基本清晰、不清晰），概念（清楚、较清楚、基本清楚、不清楚）。 | |

**目 录**

**[第1章 产品需求 2](#_Toc18183)**

[1.1 背景和目的 2](#_Toc26687)

[1.2 用户需求和系统功能 4](#_Toc22206)

[1.3 产品设计和页面 6](#_Toc7484)

[1.4 项目资源和进度安排 11](#_Toc14979)

**[第2章 概要设计](#_Toc23640)** [14](#_Toc23640)

[2.1 架构设计 14](#_Toc26126)

[2.2 功能设计（自己完成部分） 16](#_Toc9919)

[2.3 数据库设计 19](#_Toc2272)

[2.4 接口设计 25](#_Toc31949)

**[第3章 系统测试](#_Toc32068)** [29](#_Toc32068)

[3.2 测试用例（自己完成部分） 29](#_Toc9626)

[第4章 总结 35](#_Toc13339)

# 第1章 产品需求

## 1.1 背景和目的

### 1.1.1 产品背景

随着人们对健康和健身的关注日益增加，健身已经成为现代人生活中不可或缺的一部分。特别是在疫情后，越来越多的人开始重视身体健康，追求科学的健身方式。然而，传统健身方式面临许多问题，如缺乏专业指导、健身效果难以量化等。这些问题导致许多健身爱好者难以坚持健身计划，甚至可能因错误的健身方法造成身体损伤。

传统健身房提供的服务虽然能够解决部分问题，但其高昂的费用和固定的时间安排限制了很多人的健身计划。特别是在快节奏的都市生活中，很多人难以抽出固定的时间去健身房。此外，健身房的指导员无法24小时提供个性化指导，健身效果难以保障。

随着人工智能和大数据技术的发展，智能健身设备和应用逐渐进入人们的视野。这些设备和应用可以通过智能算法和传感器技术，实时监测用户的健身数据，并提供个性化的健身建议。然而，目前市场上的大多数智能健身应用仍然存在一些不足，如数据监测不全面、个性化指导不够精准等。

基于此背景，本项目旨在开发一款基于OpenAI和计算机视觉技术的智能健身管理系统。该系统能够实时监测用户的健身动作，通过人工智能分析用户的健身数据，提供个性化的健身建议和指导。系统通过摄像头实时捕捉用户的健身动作，结合计算机视觉技术，识别用户的关键运动姿态，并通过大数据分析和AI模型，评估用户的运动效果，给出改进建议。

我们的目标是通过先进的技术手段，提供一个科学、便捷、个性化的健身解决方案，帮助用户科学健身，避免运动损伤，提高健身效果。同时，系统还可以记录和分析用户的健身数据，生成详细的健身报告，帮助用户直观了解自己的健身进展，制定科学的健身计划。

通过本项目的实施，我们希望能够解决传统健身方式存在的问题，提高用户的健身体验，让更多的人能够享受到科技带来的便利，实现健康生活的目标。

### 1.1.2 产品目标

本项目的主要目标是开发一款基于OpenAI和计算机视觉（CV）技术的智能健身管理系统，旨在为用户提供科学、便捷、个性化的健身解决方案。具体目标如下：

**实时健身动作监测与分析**：利用计算机视觉技术，通过摄像头实时捕捉用户的健身动作，精确识别并记录关键运动姿态。通过人工智能分析这些数据，评估用户的运动质量，检测错误动作，并提供即时的反馈和纠正建议。

**个性化健身指导与建议**：结合用户的健身目标、身体数据和历史训练数据，系统利用OpenAI提供的算法模型，生成个性化的健身计划和指导建议。根据用户的实际情况动态调整训练计划，提高健身效果。

**健身数据的记录与可视化**：系统能够全面记录用户的健身数据，包括动作类型、次数、持续时间等。通过数据分析和可视化工具，生成详细的健身报告，帮助用户直观了解自己的健身进展和效果。

**用户友好界面与交互体验**：设计简洁、直观的用户界面，确保用户能够轻松使用各项功能。提供良好的用户交互体验，使用户能够在使用过程中感受到系统的智能和便利。

**多平台支持与数据同步**：系统支持多平台使用，包括Android和iOS设备，确保用户可以随时随地进行健身训练。通过云端同步功能，用户的健身数据能够在不同设备间无缝切换，提供一致的用户体验。

社会化分享与互动：增加社交功能，允许用户分享自己的健身成果和训练经验，与其他用户互动交流，形成健身社区，提升用户的健身动力和参与感。

## 1.2 用户需求和系统功能

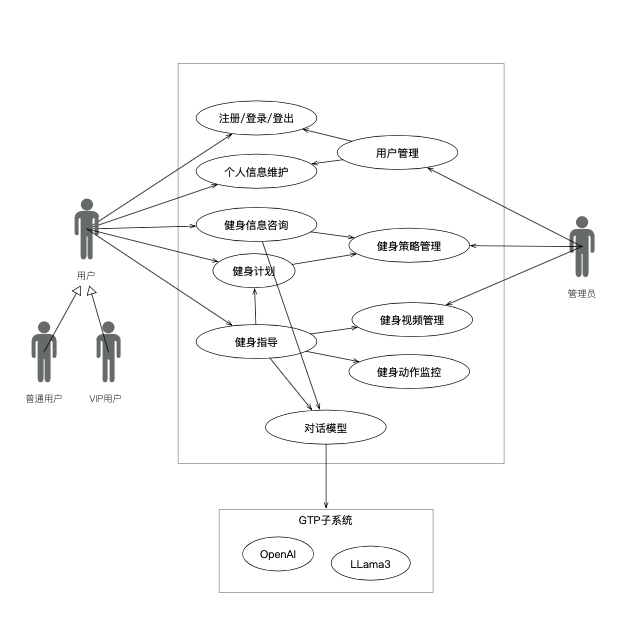
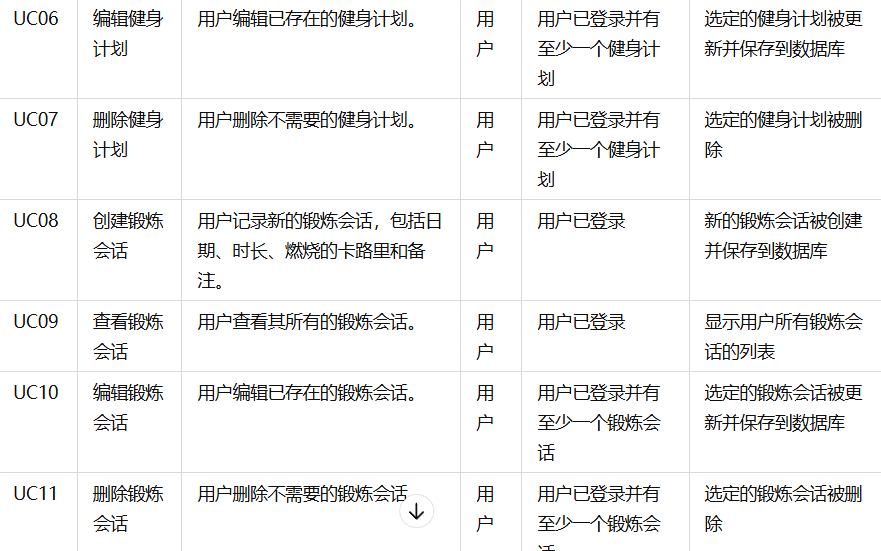


图1-1 系统用例图

表1-1 用户用例说明表

用户可以进行注册、登录、登出、动作监控、健身计划、健身会话和动作管理和数据分析。每个用例的详细说明如表1-2所示。





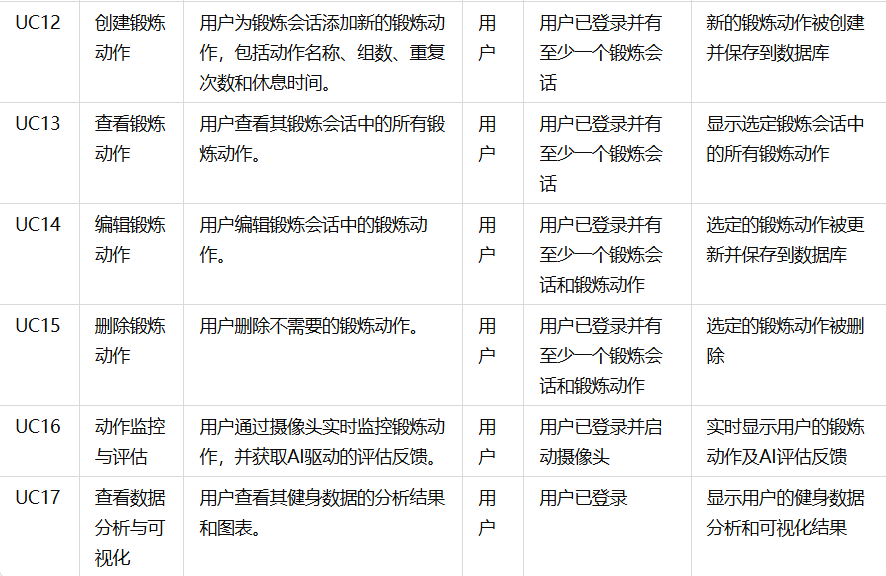


表1-2 用户用例说明表

## 1.3 产品设计和页面

### 1.3.1 登录页面

· **功能**：允许用户输入用户名和密码进行登录。

· **描述**：用户在此页面输入其注册时使用的用户名和密码进行身份验证。如果验证成功，用户将被重定向到仪表盘页面。如果登录失败，将显示错误消息并提示用户重新输入信息。展示如图1-2所示。



图1-2 登录界面展示图

### 1.3.2 注册页面

* **功能**：允许新用户创建一个账户。
* **描述**：用户在此页面输入用户名、密码和电子邮件地址进行注册。系统将验证输入的信息，确保用户名唯一，密码符合安全标准。如果注册成功，用户将被重定向到登录页面。展示如图1-3所示。



图1-3 注册界面展示图

### 1.3.3 健身计划页面

* **功能**：管理用户的健身计划。
* **描述**：用户可以在此页面创建、查看、编辑和删除健身计划。每个计划包括计划名称、描述和创建时间。用户可以查看所有已创建的健身计划，并在需要时进行修改或删除。展示如图1-4所示。



图1-4 健身计划界面展示图

### 1.3.4 锻炼会话页面

* **功能**：记录和管理用户的锻炼会话。
* **描述**：用户可以在此页面记录新的锻炼会话，包括锻炼日期、时长、燃烧的卡路里和备注。用户还可以查看和管理过去的锻炼会话，查看详细信息和进度。展示如图1-5所示。



图1-5 锻炼会话界面展示图

### 1.3.5 锻炼动作页面

* **功能**：管理具体的锻炼动作。
* **描述**：用户可以在此页面添加、查看和编辑锻炼动作。每个锻炼动作包括动作名称、组数、重复次数和休息时间。用户可以为每个锻炼会话分配不同的锻炼动作，并查看动作的详细信息。展示如图1-6所示。

图1-6 锻炼动作界面展示图

### 1.3.6 动作监控页面

* **功能**：实时监控用户的锻炼动作，并提供评估反馈。
* **描述**：通过摄像头实时监控用户的锻炼动作，使用集成的机器学习模型评估用户的动作规范性。页面显示实时视频和评估反馈，帮助用户纠正动作，提高锻炼效果。展示如图1-7所示。

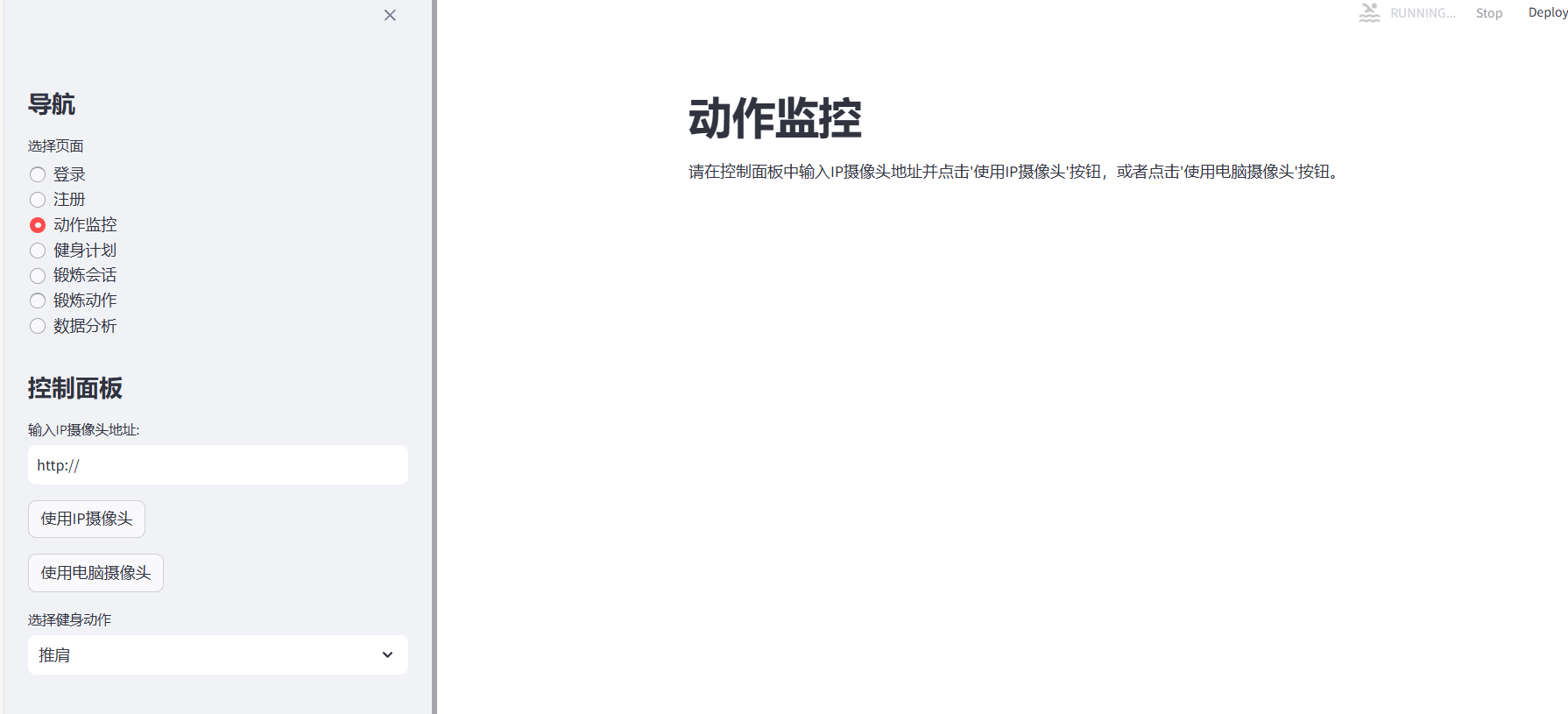


图1-7 动作监控界面展示图

### 1.3.7 数据分析页面

* **功能**：分析和可视化用户的健身数据。
* **描述**：用户可以在此页面查看其健身数据的分析结果，包括锻炼时长、卡路里消耗、动作规范性等。数据以图表形式展示，帮助用户直观了解其健身进展和效果。展示如图1-8所示。



图1-8 数据分析界面展示图

## 1.4 项目资源和进度安排

本项目的主要任务包括：产品设计、系统设计、模块开发、测试、上线等阶段。项目任务

**项目需求分析**（2024年6月2日 - 2024年6月6日）

* 1. 任务描述：确定项目的需求，收集和分析用户需求，形成需求文档。
  2. 参与人员：刘佳骏, 侯姝华

**资源规划**（2024年6月6日 - 2024年6月9日）

* 1. 任务描述：确定项目所需的资源，包括人员、硬件、软件等。
  2. 参与人员：何雪莉, 黄亚楠

**时间表制定**（2024年6月9日 - 2024年6月11日）

* 1. 任务描述：制定项目的详细时间表，确定各阶段的完成时间。
  2. 参与人员：李畅

**系统架构设计**（2024年6月11日 - 2024年6月16日）

* 1. 任务描述：设计系统的总体架构，确定技术方案。
  2. 参与人员：王旭

**数据库设计**（2024年6月16日 - 2024年6月20日）

* 1. 任务描述：设计数据库的结构，确保数据存储的合理性和效率。
  2. 参与人员：张颖

**UI/UX设计**（2024年6月20日 - 2024年6月25日）

* 1. 任务描述：设计用户界面和用户体验，确保系统的易用性和美观性。
  2. 参与人员：侯姝华

**用户管理模块开发**（2024年6月25日 - 2024年7月10日）

* 1. 任务描述：开发用户管理模块，实现用户的注册、登录、权限管理等功能。
  2. 参与人员：刘佳骏, 何雪莉

**健身信息咨询模块开发**（2024年7月10日 - 2024年7月20日）

* 1. 任务描述：开发健身信息咨询模块，提供健身知识和资讯查询功能。
  2. 参与人员：黄亚楠, 李畅

**健身计划模块开发**（2024年7月20日 - 2024年7月30日）

* 1. 任务描述：开发健身计划模块，用户可以制定和管理自己的健身计划。
  2. 参与人员：王旭, 张颖

**健身指导模块开发**（2024年7月30日 - 2024年8月10日）

* 1. 任务描述：开发健身指导模块，通过计算机视觉技术提供实时的健身指导。
  2. 参与人员：侯姝华, 刘佳骏

**动作监控模块开发**（2024年8月10日 - 2024年8月20日）

* 1. 任务描述：开发动作监控模块，实时监控用户的健身动作并提供反馈。
  2. 参与人员：何雪莉, 黄亚楠

**单元测试**（2024年8月20日 - 2024年8月30日）

* 1. 任务描述：对各个模块进行单元测试，确保各模块功能正常。
  2. 参与人员：李畅, 王旭

**集成测试**（2024年8月30日 - 2024年9月10日）

* 1. 任务描述：对系统进行集成测试，确保各模块之间的协调工作。
  2. 参与人员：张颖, 侯姝华

**系统测试**（2024年9月10日 - 2024年9月20日）

* 1. 任务描述：对整个系统进行全面的功能和性能测试。
  2. 参与人员：刘佳骏, 何雪莉

**性能测试**（2024年9月20日 - 2024年9月30日）

* 1. 任务描述：对系统进行性能测试，确保系统在高负载下的稳定性和响应速度。
  2. 参与人员：黄亚楠, 李畅

**回归测试**（2024年9月30日 - 2024年10月5日）

* 1. 任务描述：进行回归测试，确保修改后的系统没有引入新的问题。
  2. 参与人员：王旭, 张颖

**系统验收**（2024年10月5日 - 2024年10月10日）

* 1. 任务描述：由用户对系统进行验收测试，确保系统满足所有需求。
  2. 参与人员：侯姝华, 刘佳骏

**上线准备**（2024年10月10日 - 2024年10月15日）

* 1. 任务描述：进行上线前的准备工作，包括数据迁移、系统配置等。
  2. 参与人员：何雪莉, 黄亚楠

**正式发布**（2024年10月15日）

* 1. 任务描述：系统正式上线，用户可以开始使用。
  2. 参与人员：全体项目成员

**持续监控**（2024年10月15日 - 2024年11月15日）

* 1. 任务描述：对系统进行持续监控，确保系统的稳定运行，及时处理出现的问题。
  2. 参与人员：李畅, 王旭

**问题修复**（2024年10月15日 - 2024年11月15日）

* 1. 任务描述：根据用户反馈和监控结果，修复系统中的问题。
  2. 参与人员：张颖, 侯姝华

**功能优化**（2024年10月15日 - 2024年11月15日）

* 1. 任务描述：对系统进行功能优化，提高系统性能和用户体验。
  2. 参与人员：刘佳骏, 何雪莉

**用户反馈处理**（2024年10月15日 - 2024年11月15日）

* 1. 任务描述：收集和处理用户反馈，进一步完善系统。
  2. 参与人员：黄亚楠, 李畅

参与本项目的人员包括：

· 刘佳骏

· 侯姝华

· 何雪莉

· 黄亚楠

· 李畅

· 王旭

· 张颖

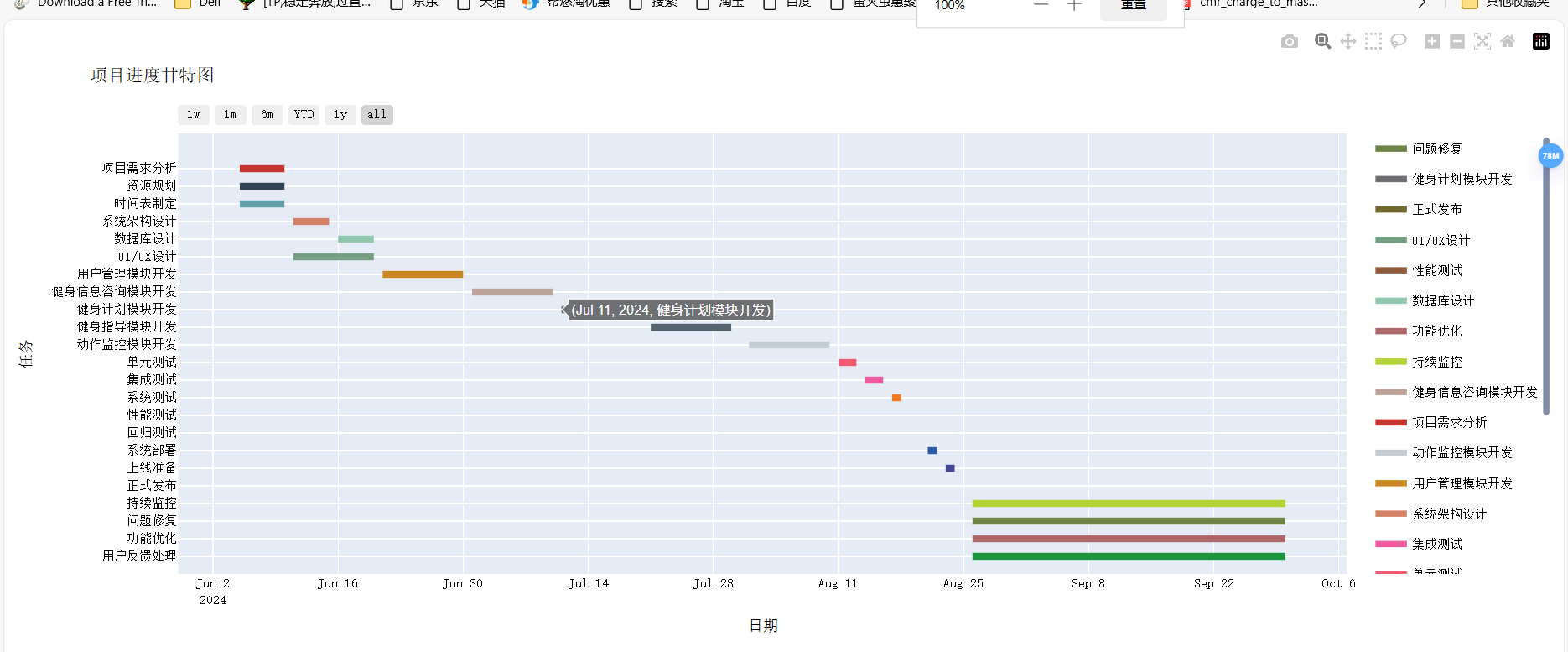


图1-9 资源和进度安排图

# 第2章 概要设计

## 2.1 架构设计

### 2.1.1 功能架构

在本系统中，将系统的功能划分为以下几个模块：注册/登录/登出、个人信息维护、健身信息咨询、健身计划、健身会话、健身动作管理、健身动作监控、数据分析。每个模块下都有其要完成的功能。通过功能架构图，可以清晰地展现出系统需要完成的任务。功能架构图具体设计如图2-1所示。

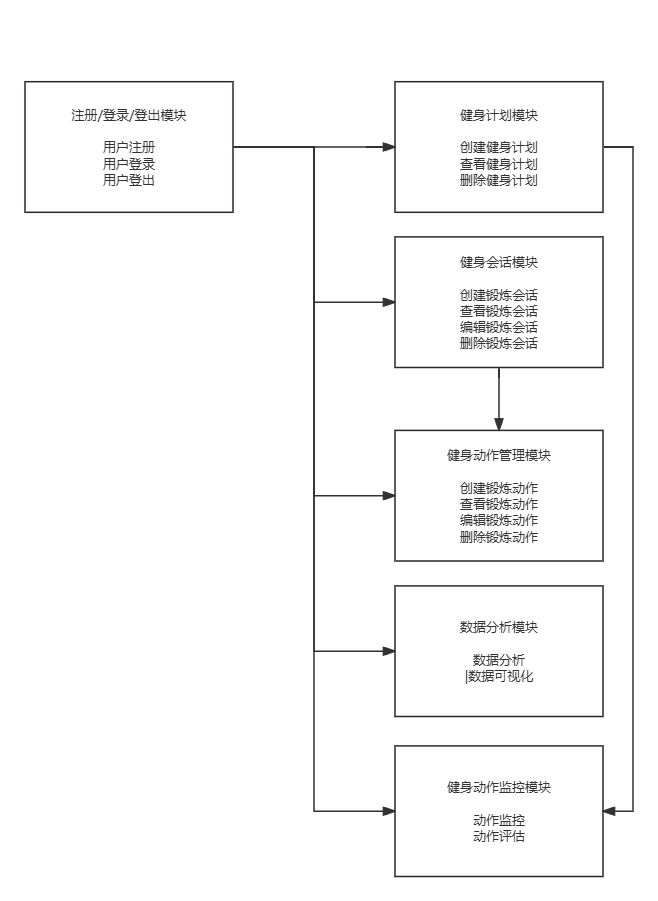


图2-1 功能架构设计图

### 2.1.2 技术架构

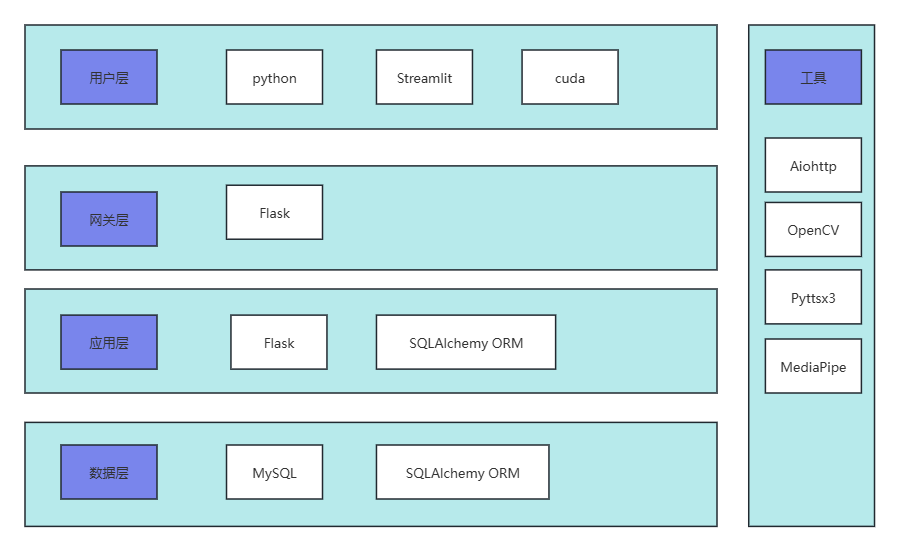


图2-2 技术架构设计图

## 2.2 功能设计（自己完成部分）

### 2.2.1健身动作

健身动作功能设计主要包括创建、查看、编辑和删除健身动作，以及实时监控和评估用户的动作。以下是详细的功能设计和技术细节。

#### 1.功能描述

创建健身动作：用户可以为锻炼会话添加新的健身动作，包括动作名称、组数、重复次数和休息时间。

查看健身动作：用户可以查看其锻炼会话中的所有健身动作。

编辑健身动作：用户可以编辑锻炼会话中的健身动作，修改动作名称、组数、重复次数和休息时间。

删除健身动作：用户可以删除不需要的健身动作。

动作监控与评估：通过摄像头实时监控用户的锻炼动作，使用机器学习模型评估动作的规范性，并提供反馈。

#### 2.数据库设计

锻炼动作表 (Exercise)

exercise\_id (PK): 锻炼动作的唯一标识符。

session\_id (FK): 关联锻炼会话的唯一标识符，引用锻炼会话表的 session\_id。

exercise\_name: 锻炼动作的名称。

sets: 动作的组数。

reps: 每组的重复次数。

rest\_time: 组间休息时间（秒）。

API 设计

#### 3.创建健身动作

URL: /create\_exercise

方法: POST

{

"session\_id": 1,

"exercise\_name": "Bench Press",

"sets": 4,

"reps": 12,

"rest\_time": 60}

{

"message": "Exercise created successfully",

"exercise\_id": 1}

#### 4.查看健身动作

URL: /get\_exercises/<session\_id>

方法: GET

响应:

[

{

"exercise\_id": 1,

"exercise\_name": "Bench Press",

"sets": 4,

"reps": 12,

"rest\_time": 60

},

...]

#### 5.编辑健身动作

URL: /edit\_exercise/<exercise\_id>

方法: PUT

请求体:

{

"exercise\_name": "Updated Bench Press",

"sets": 3,

"reps": 10,

"rest\_time": 90}

{

"message": "Exercise updated successfully"}

#### 6.删除健身动作

URL: /delete\_exercise/<exercise\_id>

方法: DELETE

响应:

{

"message": "Exercise deleted successfully"}

#### 7.技术细节

1. 创建健身动作

当用户提交创建健身动作的表单时，系统接收并处理请求，验证数据的有效性，然后将新动作保存到数据库中。

python

复制代码

from flask import request, jsonifyfrom models import db, Exercise

@app.route('/create\_exercise', methods=['POST'])def create\_exercise():

data = request.get\_json()

new\_exercise = Exercise(

session\_id=data['session\_id'],

exercise\_name=data['exercise\_name'],

sets=data['sets'],

reps=data['reps'],

rest\_time=data['rest\_time']

)

db.session.add(new\_exercise)

db.session.commit()

return jsonify({"message": "Exercise created successfully", "exercise\_id": new\_exercise.ex

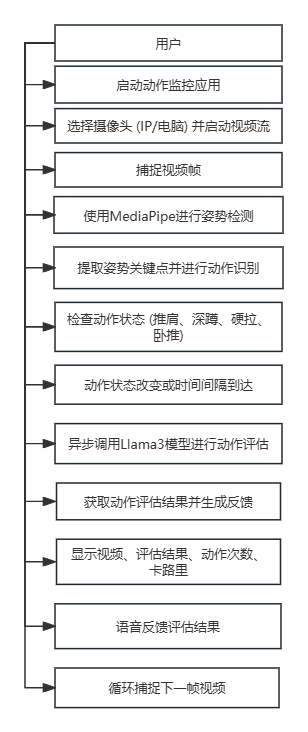


图2-3 动作监控活动图

## 2.3 数据库设计

### 2.3.1 概念模型（整体）

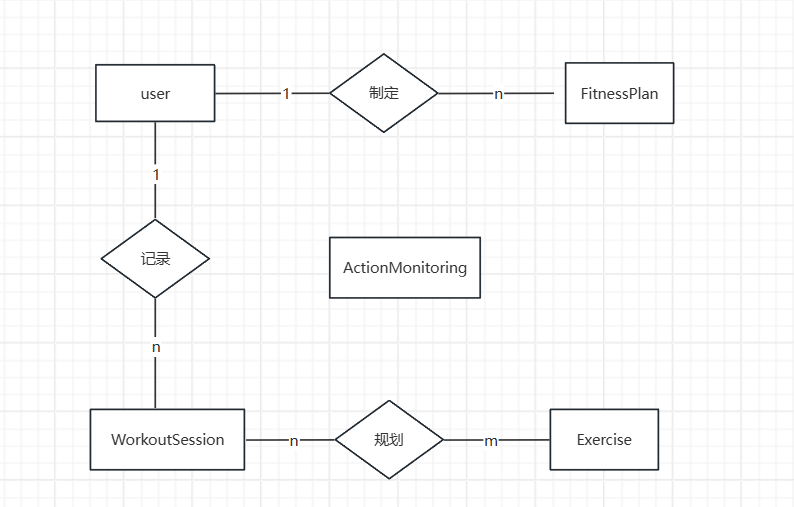
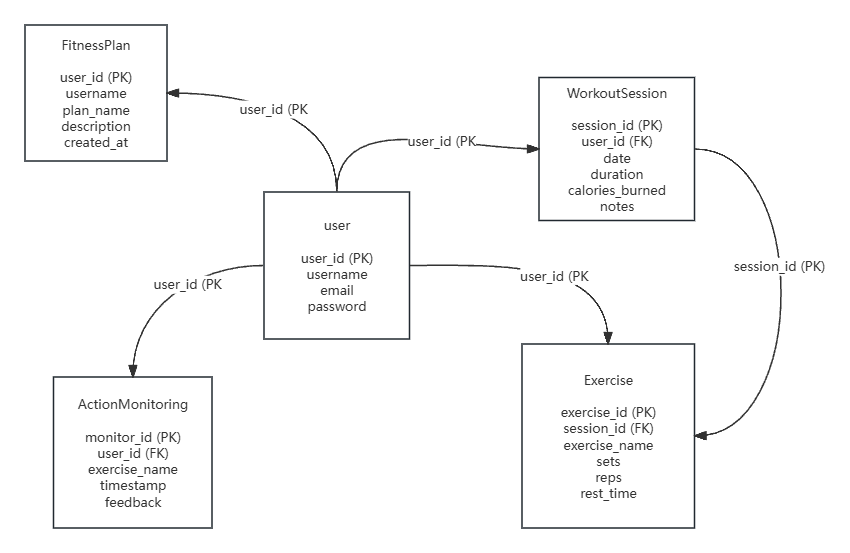


图2-4 E-R图

用户表存储所有用户的基本信息。每个用户都有一个唯一的 user\_id 作为主键。健身计划表存储用户创建的健身计划。每个健身计划都关联一个用户。

锻炼会话表存储用户的锻炼会话信息。每个锻炼会话都关联一个用户。锻炼动作表存储用户在每个锻炼会话中进行的锻炼动作。每个锻炼动作都关联一个锻炼会话。

动作监控表存储系统对用户锻炼动作的实时监控记录和反馈信息。每条记录都关联一个用户，包含动作名称、时间戳和系统的评估反馈。



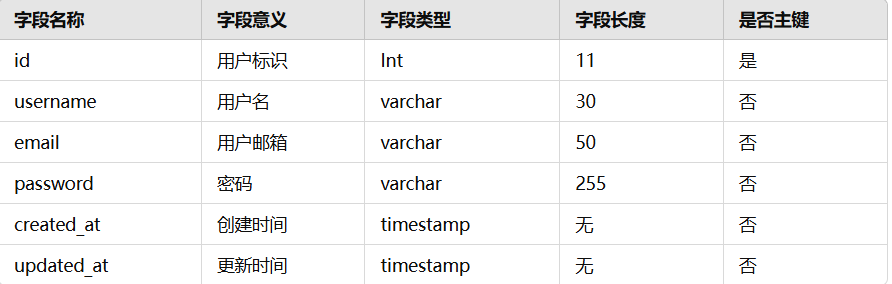
2.3.2 逻辑模型（整体）

数据库逻辑结构的设计通过数据库表来体现，系统中数据库表的设计决定了系统的拓展性、维护性等。该健身管理系统的逻辑模型如下图所示：

图2-5 逻辑模型图

（1）用户信息表的主要字段包括用户ID、用户名、用户邮箱等字段

表2-1 用户信息表

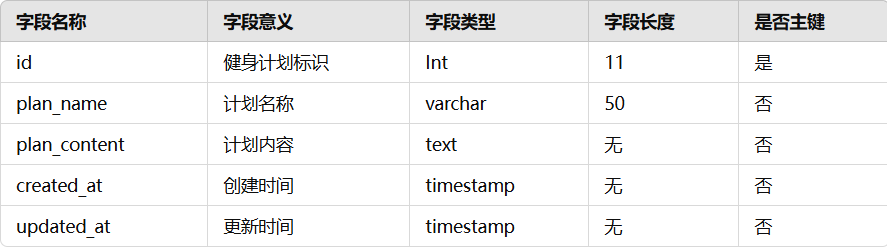


1. 角色信息表主要字段包括角色ID、角色名称等字段

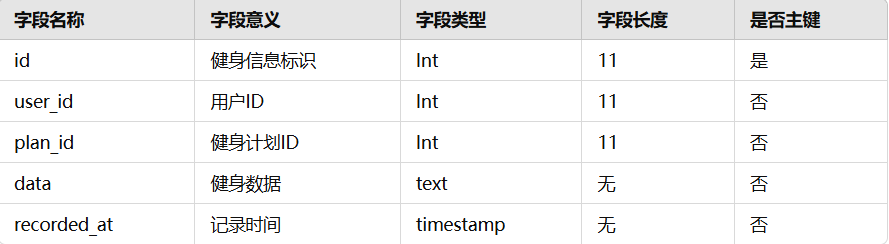
表2-2 角色信息表



1. 健身计划表主要字段包括健身计划ID、计划名称、计划内容等字段

表2-3健身计划表

1. 健身信息表主要字段包括健身信息ID、用户ID、健身计划ID、健身数据等字段

表2-4 健身信息表

### 2.3.3 物理模型（自己完成部分）

1.以下健身动作的SQL语句：

|  |
| --- |
| CREATE TABLE Exercise (  exercise\_id INT AUTO\_INCREMENT PRIMARY KEY,  session\_id INT,  exercise\_name VARCHAR(100),  sets INT,  reps INT,  rest\_time INT,  FOREIGN KEY (session\_id) REFERENCES WorkoutSession(session\_id)  ); |

## 2.4 接口设计

### 2.3.1 整体接口设计（整体）

表2-2健身管理接口表



表2-3用户管理接口表



表2-4 健身信息接口表



表2-5 角色管理接口表

### 2.3.2 接口详细设计（自己完成功能）

#### 1. 健身动作

1）获取用户信息

表2-7接口描述表

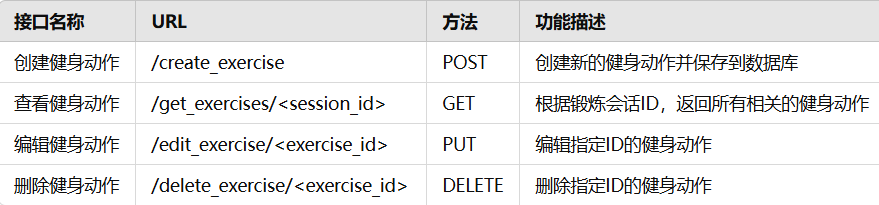


表2-8参数表



表2-9返回数据表

# 系统测试

3.1 测试范围（整体）

测试范围涵盖了整个健身房管理系统的各个模块，以确保系统功能的全面性、稳定性和可靠性。测试将包括以下主要模块：

1. **用户管理**：包括用户信息的增删改查功能及用户权限的维护和管理。验证用户的创建、修改、删除和检索功能，以及用户权限设置和控制。
2. **角色管理**：包括角色的创建、修改、删除及权限设置和验证。确保角色管理功能的有效性。
3. **健身计划管理**：包括健身计划的创建、修改、删除及分配给用户。验证健身计划的管理功能及其与用户的关联关系。
4. **健身信息管理**：包括健身记录的录入、修改和查看功能。确保健身信息的准确记录和有效管理。

主要采用的测试方法包括：

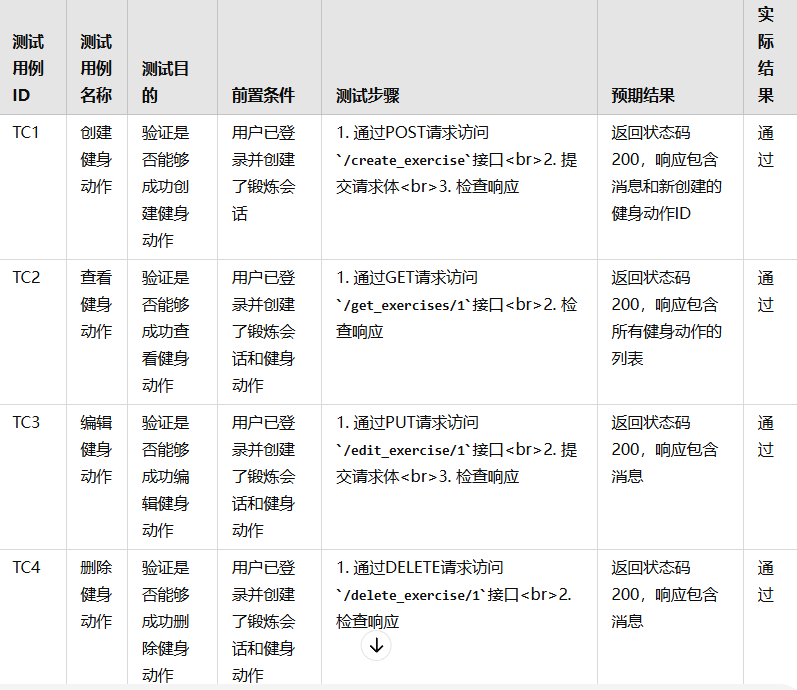
1. **功能测试**：验证每个功能模块的基本功能是否按预期工作。
2. **边界测试**：测试系统在各种边界条件下的表现，如最大输入、最小输入和异常输入。

## 3.2 测试用例（自己完成部分）

### 3.2.1健身动作测试

为了确保智能健身应用程序中健身动作功能的正确性和可靠性，需要进行详细的测试。以下是针对创建、查看、编辑和删除健身动作功能的测试用例描述。这些测试用例覆盖了基本的CRUD操作，验证了系统在不同输入情况下的行为。健身动作测试用例表如表3-1所示。

表3-1 用户管理测试用例表



# 第4章 总结

智能健身应用程序是一个利用现代技术提升用户健身体验的综合性平台。项目采用了Flask和Streamlit作为后端和前端框架，结合MediaPipe、OpenCV、Pyttsx3和异步HTTP请求等技术，实现了用户注册、登录、个人信息维护、健身计划管理、锻炼会话记录、实时动作监控和数据分析等一系列功能。用户可以通过注册和登录功能创建账户并管理个人信息，系统通过加密技术确保用户数据的安全性。在个人信息维护模块中，用户可以查看和编辑自己的基本信息，如用户名、电子邮件和密码。健身信息咨询模块提供了用户查看和管理健身计划、锻炼会话和动作监控数据的接口，用户可以根据个人需要创建、编辑和删除健身计划，详细记录每次锻炼会话的日期、时长、卡路里消耗和备注等信息。

项目的核心功能之一是实时动作监控，通过摄像头捕捉用户的锻炼动作，并利用MediaPipe进行姿势检测，系统能够实时获取用户的关键点数据，识别用户进行的动作（如推肩、深蹲、硬拉和卧推），并评估其动作的规范性。评估结果通过集成的Llama3模型生成，系统会根据关键点数据提供直接且简明的反馈，帮助用户优化锻炼姿势。此外，系统还具备语音反馈功能，利用Pyttsx3将评估结果实时播报给用户，进一步提升用户体验。在动作监控的同时，系统会记录用户完成的动作次数和消耗的卡路里，并在界面上实时展示这些数据。

为了实现数据的持久化和管理，项目采用了MySQL数据库，并使用SQLAlchemy进行ORM映射。用户、健身计划、锻炼会话、锻炼动作和动作监控数据分别存储在各自的表中，通过外键关系将不同表的数据关联起来。例如，用户表中的每个用户可以关联多个健身计划和锻炼会话，锻炼会话表中的每个会话可以包含多个锻炼动作，而动作监控表则记录了系统对用户锻炼动作的实时监控数据和反馈信息。通过清晰的数据库设计，项目能够高效地管理和查询用户的健身数据。

在技术架构方面，项目分为用户层、网关层、应用层、数据层和工具层。用户层通过Streamlit提供用户友好的界面，网关层使用Flask处理客户端请求并将其路由到相应的后端服务，应用层负责核心业务逻辑的实现，包括用户管理、健身计划管理、锻炼会话管理、动作监控和数据分析等功能，数据层负责数据的持久化存储和访问，而工具层提供了语音合成、姿势检测和异步HTTP请求等辅助功能。各层之间的协同工作确保了系统的高效性和可扩展性。

项目的设计和实现过程中，充分考虑了用户体验和系统性能。通过流畅的用户界面和实时反馈机制，用户可以方便地管理健身计划和锻炼数据，并在锻炼过程中获得及时的动作评估和改进建议。系统的架构设计则确保了各模块的高效运行和数据的安全性，为未来的功能扩展和性能优化提供了良好的基础。

总结而言，智能健身应用程序通过现代技术手段，为用户提供了一个全面的健身管理平台。它不仅涵盖了基本的用户管理和健身数据记录功能，还通过实时动作监控和智能评估提升了用户的锻炼体验。项目的成功实施展示了技术在健康领域的巨大潜力，未来可以进一步扩展和优化，为用户提供更加个性化和智能化的健身指导服务。

附件：答辩纪录

**《软件系统设计》课程答辩记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业班级 | 计科2104 | 姓名 | 刘佳骏 | 学号 | 21851053 |
| 答辩时间 | 2024-06-25 9：00-10：50 | | 答辩地点 | 思源楼334 | |
| 答辩题目 | 健身智囊助手 | | | | |
| 答辩老师 | 田晋坤 | | | | |
| 提问及答辩记录：  **问题1：如何实现创建健身动作的功能？请详细描述实现过程？**  前端部分：  用户在前端页面填写健身动作的相关信息，包括动作名称、组数、重复次数和休息时间。  前端页面通过一个表单将这些信息发送到后端服务器。  后端部分：  后端服务器接收前端发送的POST请求，并解析请求体中的JSON数据。  验证请求体中的数据是否完整且符合要求，例如动作名称不能为空，组数和重复次数必须为正整数等。  使用SQLAlchemy ORM创建一个新的Exercise对象，并将解析后的数据填充到这个对象中。  将新创建的Exercise对象添加到数据库会话中，并提交会话，以将数据保存到数据库中。  返回一个JSON响应，包含成功消息和新创建的健身动作的ID。  **问题2：如何确保用户在编辑健身动作时，数据的正确性和安全性？**  数据验证：  后端在接收到编辑请求时，首先要验证请求体中的数据。确保提供的动作名称、组数、重复次数和休息时间都是有效的。  动作名称应为非空字符串，组数和重复次数应为正整数，休息时间应为非负整数。  身份验证和授权：  确保用户已登录，并且有权限编辑该健身动作。这可以通过会话管理或JWT（JSON Web Token）来实现。  验证用户身份后，确保用户只能编辑属于自己的健身动作，而不能修改其他用户的记录。  防止SQL注入攻击：  使用ORM（如SQLAlchemy）进行数据库操作，避免直接使用SQL查询，以防止SQL注入攻击。  事务管理：  在进行数据库更新操作时，使用事务管理（Transaction Management）确保数据一致性。如果在更新过程中出现任何错误，可以回滚事务，以确保数据库状态的一致性和完整性。 | | | | | |