

学号

XXX

武汉理工大学

实践课设计报告

课程名称	软件开发技术实训
题目	实验排课系统
学院	计算机与人工智能学院
专业	计算机科学与技术
班级	XXX
姓名	XXX
指导教师	XXX

2024 年 12 月 24 日

目录

1. 系统分析	4
1.1 问题说明	4
1.1.1 背景分析	4
1.1.2 现状分析	4
1.2 数据需求	5
1.2.1 数据字典	5
1.2.2 数据流图	6
1.3 功能需求	7
(1) 学生功能	7
(2) 教师功能	8
(3) 管理员管理	8
(4) 用户信息管理	9
2. 系统设计	10
2.1 总体设计	10
2.1.1 总体概述	10
2.1.2 后端架构设计	11
2.1.3 前端架构设计	12
2.2 详细设计	13
2.2.1 排课后端业务详细设计	13
2.2.2 选课后端业务详细设计	15
2.2.2 AI 对话后端业务详细设计	17
2.3 数据库表结构设计	17
2.3.1 概念结构设计	17
2.3.2 逻辑结构设计	19
3 系统实现与测试	20
3.1 技术框架与开发环境选型	20
3.1.1 后端技术框架	20
3.1.2 前端技术框架	20
3.1.4 数据库系统	20
3.1.5 部署环境	21
3.2 代码风格	21
3.2.1 代码结构	21
3.2.2 特定语言规范	21
3.2.5 语言规范检查	22
3.3 软件系统实现界面	22
3.3.1 主界面	22
3.3.2 登录界面	23
3.3.3 注册界面	24
3.3.4 选课界面	24
3.3.5 选课记录界面	25
3.3.6 课表界面	26
3.3.7 排课界面	27

3.3.8 助手问答界面	27
3.4 测试	28
3.5.2 选课功能的测试	29
4. 个人具体工作	30
5 实训总结	31
5.1 开发亮点	31
5.1.1 基于 CodeArts 的开发	32
5.1.2 异步技术	34
5.1.3 容器化 docker 技术	34
5.1.4 AI 分布式计算技术	35
5.2 实训总结	36
5.2.1 开发中的挑战与解决方案	36
5.2.2 收获与总结	39
本科生实践课程成绩评定表	41

1. 系统分析

1.1 问题说明

1.1.1 背景分析

实验课排课系统在现代教学管理中扮演着至关重要的角色，实验课程是培养学生实践能力、创新思维以及综合素质的重要环节。实验课排课系统应运而生，目标是通过高效、合理的排课安排，优化教学资源的配置，满足学生的选课需求，同时兼顾教师的教学安排，确保每一堂实验课都能顺利进行。

实验课排课系统是现代教育管理中不可或缺的一部分，系统能够帮助学校减少排课时间和人工成本，优化实验教学资源配置，确保实验课程能够顺利开展，并提高教学质量和学生满意度。

1.1.2 现状分析

目前选课与排课系统存在如图 1.1.1 的以下问题：

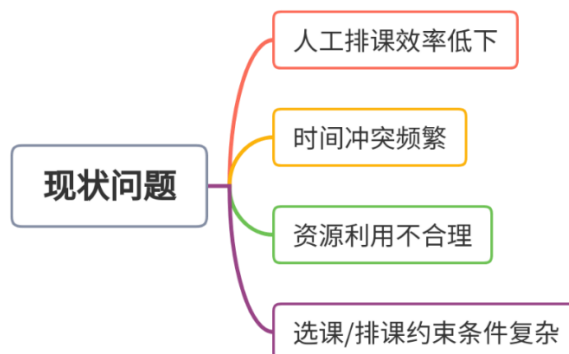


图 1.1.1 选课与排课系统现状问题

(1) 人工排课效率低下：目前，大多数高校仍采用人工方式进行课程安排，这种方法不仅耗时耗力，且容易出错。特别是对于多样化课程类型和资源限制。

(2) 资源利用不合理：

由于人工排课的局限性，教学资源往往无法得到充分利用。具体表现为资源安排出现空闲时间段，或资源过度闲置与分配不足。

(3) 时间冲突频繁：

教师和学生的时间安排难以协调，造成课程安排频繁冲突。这种冲突不仅影响教学效率，还降低了学生的学习体验和教师的教学安排。

(4) 约束条件复杂：

高校的实验排课系统面临许多复杂的约束条件。这些包括课程类型的多样性。课程时长、周次安排、时间段偏好等因素相互交织，增加了排课的复杂性。所有这些因素合理协调成为排课系统设计的难点。

1.2 数据需求

1.2.1 数据字典

如表 1.2.1 为部分关键数据项描述表，从数据项，含义说明，类型，长度，取值含义，与其他数据项的逻辑关系角度描述数据项。

表 1.2.1 数据项描述表

数据项	含义说明	类型	长度	取值含义	与其他数据项的逻辑关系
课程号	课程的唯一编号	字符型	/	课程唯一标识符	与课程名称、授课教师等数据项关联
课程名称	课程名称	字符型	最大 50	例如操作系统	每个课程号对应唯一的课程名称
学生号	学生编号	字符型	10	根据学生注册顺序生成唯一编号	学生号与学生姓名、选课信息等数据项一一对应
课程教室	上课教室	字符型	最大 10	例如“1 教室”	与课程号、课程时间等数据项关联
课程时间	上课时间	日期型	/	YY-MM-DD 表示	决定学生选课的顺序及是否存在冲突
教师号	教师编号	字符型	10	根据教师注册顺序生成唯一编号	与教师姓名、所教授的课程等数据项一一对应
课程容量	允许选修的最大学生人数	数值型	/	例如“50 人”等	与选课人数、空余名额等数据项关联，决定学生是否能成功选课
学生姓名	学生名字	字符型	10	学生名字	与学号数据项一一对应
学分	学分数	数值型	2	例如“3 学分”、“2 学分”等	与课程号和课程名称一一对应关系
教师姓名	教师名字	字符型	5	教师名字	与教师号等数据项一一对应

如表 1.2.2 为关键数据结构描述表，从名称，含义说明，组成角度描述数据结构。

表 1.2.2 数据结构描述表

数据结构	含义说明	组成
当前课程统计表	标识当前课程总体使用情况，包括课程状态、空余名额等	课程号、课程名称、教师号、课程时间、剩余名额
学生	学生的基本信息，用于选课及成绩管理	学生号、姓名、身份证号、选修课程列表（包括课程号、选课时间、成绩）
课程	学校提供的课程信息，包含课程的基本安排及信息	课程号、课程名称、课程学分、授课教师号、课程时间、教室号、最大人数、已选人数
教师	教师的基本信息及其所教授的课程	教师号、教师姓名、教授课程列表（包括课程号、课程名称）
教室	学校提供的课程上课教室的相关信息	教室号、教室名称、教室容量、是否有空闲时间（用于避免课程冲突）
排课记录	教师排课的记录	教师号，课程号，排课安排时间
选课记录	学生在特定学期选修的课程及相关信息	学生号、课程号、选课时间、选课状态（已选、未选）
成绩	记录学生在某门课程中的成绩	学生号、课程号、成绩

1.2.2 数据流程图

实验课排课系统系统数据流程图如图 1.2 所示：

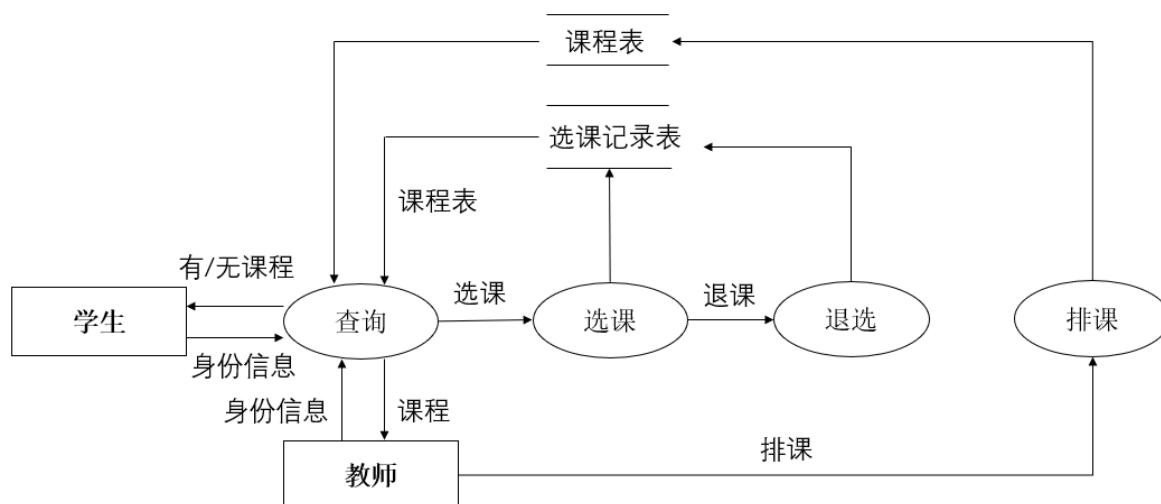


图 1.2.1 系统数据流程图

该数据流程图描述了一个选课管理系统的核心流程，包括学生、教师、课程信息以及相关操作之间的交互。首先，学生通过系统查询课程信息，根据返回的课程表进行选课操作。如果课程容量已满，系统提供扩容功能，通过增加课程容量满足更多学生的选课需求。学生完成选课后，选课记录表会更新相应信息。此外，学生也可以进行退课操作，

系统会同步更新选课记录表和课程信息。教师则提供课程信息，确保学生查询到准确的课程数据，并根据需要调整课程容量或管理学生的选课情况，同时教师可以进行排课。整体流程实现了排课、选课、退课、课程管理和扩容的闭环操作，确保系统高效运行并满足学生和教师的需求。

1.3 功能需求

本文设计的主要功能包括 4 个部分，分别是学生管理、教师管理、管理员管理、用户信息管理。顶层用例图如图 1.3.1 所示，各功能模块相互协作，支持用户信息的全面管理与操作，实现系统的高效运行和用户体验的优化。

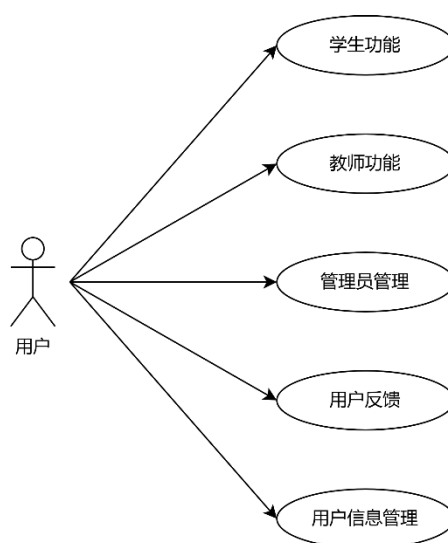


图 1.3.1 软件顶层用例图

(1) 学生功能

学生功能模块是系统的重要组成部分，主要功能包括学生选课、已选课程查看、课表查询和成绩查询。课表查询功能根据学生选课生成直观的每日或每周课表，方便规划学习时间，同时检测课程冲突以优化安排。成绩查询功能则为学生提供全部的学习成绩信息。该模块全面覆盖学生学习和课程管理需求，提升了学习安排的便捷性和效率，为学生提供了高效的课程与学习管理服务。学生用例图，如图 1.3.2 所示。

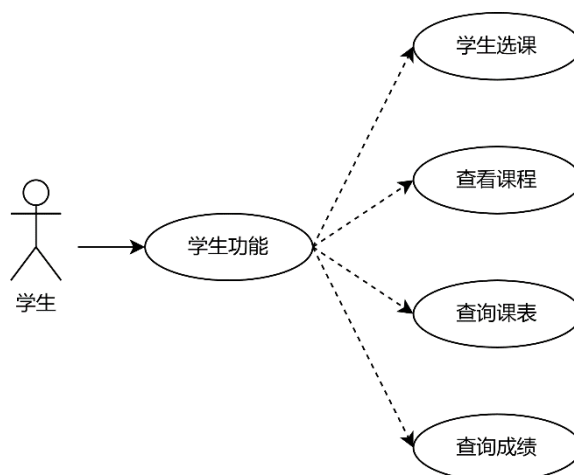


图 1.3.2 学生管理用例图

(2) 教师功能

教师功能模块是系统中针对教师用户的重要功能部分，主要涵盖实验课程安排、成绩录入以及课程查询等功能。通过实验课程安排功能，教师可以设置实验课的具体时间、地点和内容；成绩录入功能允许教师对所授课程的学生成绩进行管理，包括录入、修改和提交；课程查询功能包括课程名称、授课时间、授课地点等详细信息，帮助教师合理安排教学任务。通过这些功能，教师管理模块有效提高了教学管理的便捷性和效率，满足教师在教学组织和管理中的核心需求。教师用例图，如图 1.3.3 所示。

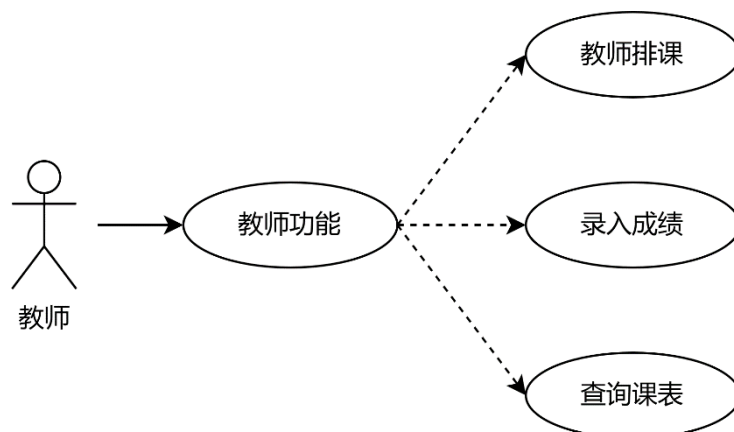


图 1.3.3 学生管理用例图

(3) 管理员管理

管理员管理模块是系统中用于维护和管理整体运行的重要功能模块，主要包含系统管理、课程管理和教师注册等核心功能。系统管理功能允许管理员设置关键时间节点，如选课时间、排课时间、成绩录入时间等，并支持查看系统延迟进度，以便灵活

调整系统运行计划。课程管理功能帮助管理员对课程进行统一管理，包括课程的创建、修改、删除以及调整，为课程安排提供支持。教师注册功能则用于管理员对教师账户的注册与审核，确保教师能够顺利加入系统并开展相关教学活动。通过这些功能，管理员管理模块能够有效保障系统的正常运行和高效运作，为整体教学管理提供强有力的支持。管理员用例图，如图 1.3.4 所示。

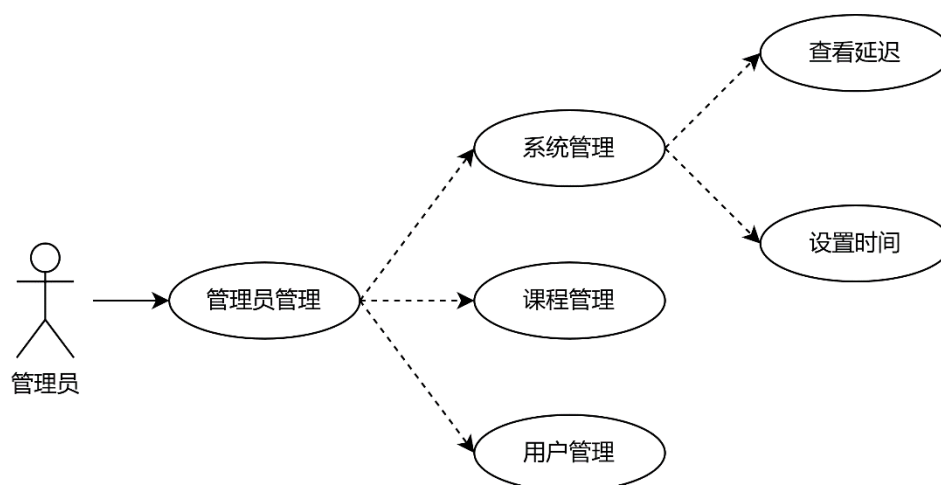


图 1.3.4 管理员管理用例图

(4) 用户信息管理

用户信息管理模块是系统中提供用户身份认证和账户管理的基础模块，主要包括用户登录和用户注册两大功能。用户登录功能支持不同角色用户的身份验证，包括学生登录、教师登录和管理员登录。用户注册功能提供学生注册和教师注册的入口，用于新用户录入系统，确保用户信息的完整性与准确性，同时实现角色的分类管理。此功能用例图如图 1.3.5 所示。

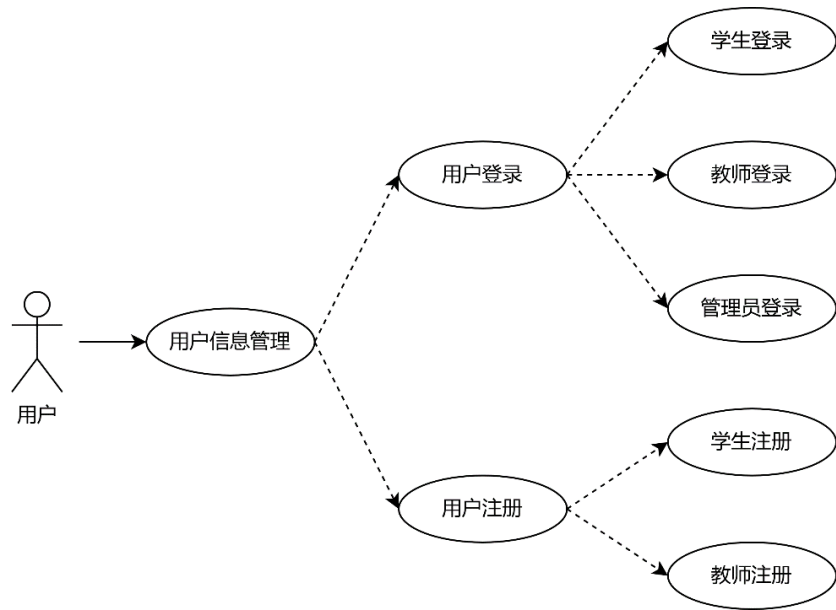


图 1.3.5 用户信息管理用例图

2.系统设计

2.1 总体设计

2.1.1 总体概述

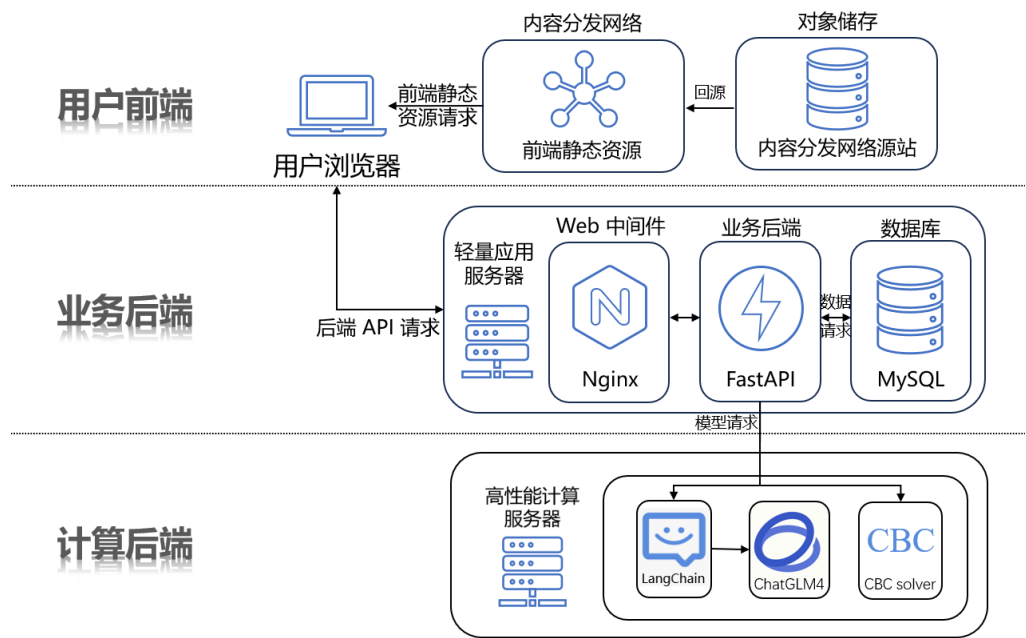


图 2.1.1 整体系统结构设计

如图 2.1.1 所示，用户通过浏览器访问前端界面，加载静态资源并与后端服务进行数据交互；业务后端作为系统的核心处理层，负责转发请求、运行业务逻辑并管理数

据存储；计算后端则提供模型运行、算法计算等高性能服务，支持复杂任务的快速执行。

用户前端是系统与用户交互的主要界面，负责为用户提供直观、友好的操作体验。用户通过浏览器发起静态资源请求，前端资源由内容分发网络（CDN）加速分发，确保快速响应并降低服务器负载。在加载静态资源后，用户通过浏览器发起后端 API 请求，实现动态交互。

业务后端是整个系统的核心逻辑处理层，连接用户前端与计算后端，负责接收、解析用户请求并执行具体业务逻辑。在该架构中，Nginx 承担了请求代理的作用，将用户请求转发至业务中间件处理模块。业务中间件采用 FastAPI 框架，凭借其高性能和灵活性，实现了各类业务逻辑的快速处理。

计算后端为系统提供了强大的模型运行和高性能计算能力，是处理复杂任务和算法优化的关键。LangChain 用于实现 RAG 检索结合 ChatGLM4，负责与用户对话；而 CBC Solver 则专注于排课数学优化和约束求解，为需要精确计算的场景提供支持。

2.1.2 后端架构设计

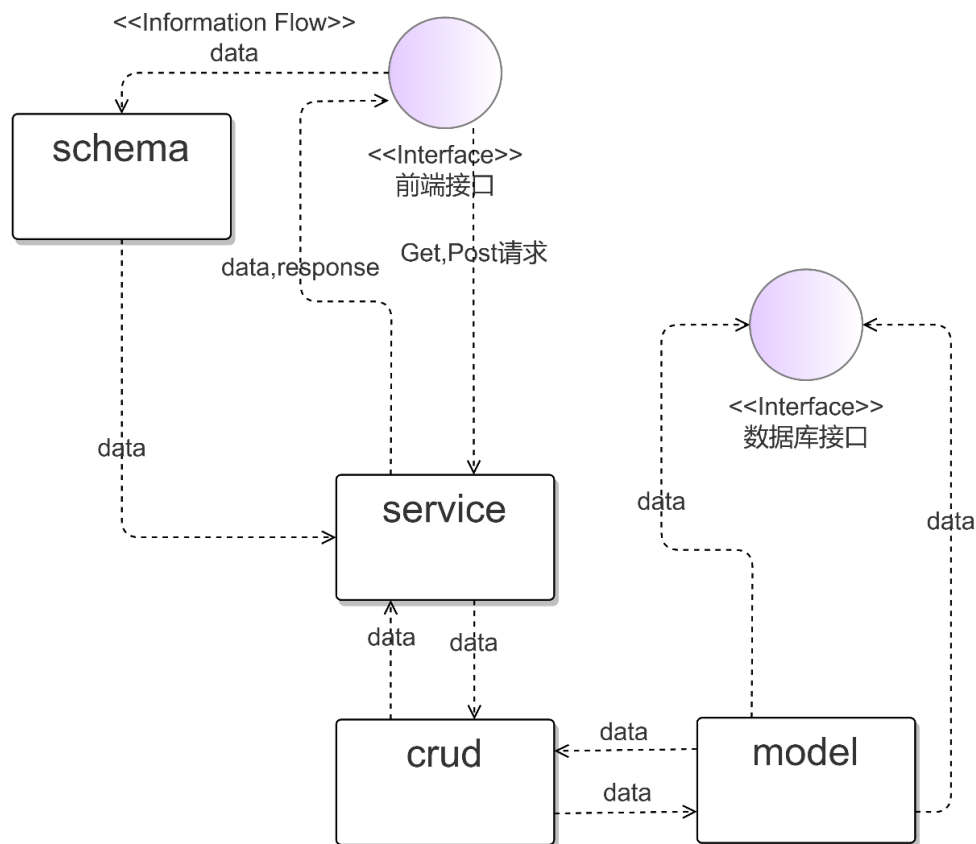


图 2.1.2 后端架构设计

后端设计大致分为四层，每一层都有明确的职责和功能，共同构成了一个高效、稳定、可扩展的系统架构。

首先是 **Service** 层，它是整个后端系统与前端交互的核心部分。**Schema** 层的主要作用是充当前端数据与后端逻辑之间的桥梁。接下来是 **CRUD** 层，它是后端与数据库交互的主要模块。。最后是 **Model** 层，它是后端与数据库表结构之间的映射层。

2.1.3 前端架构设计

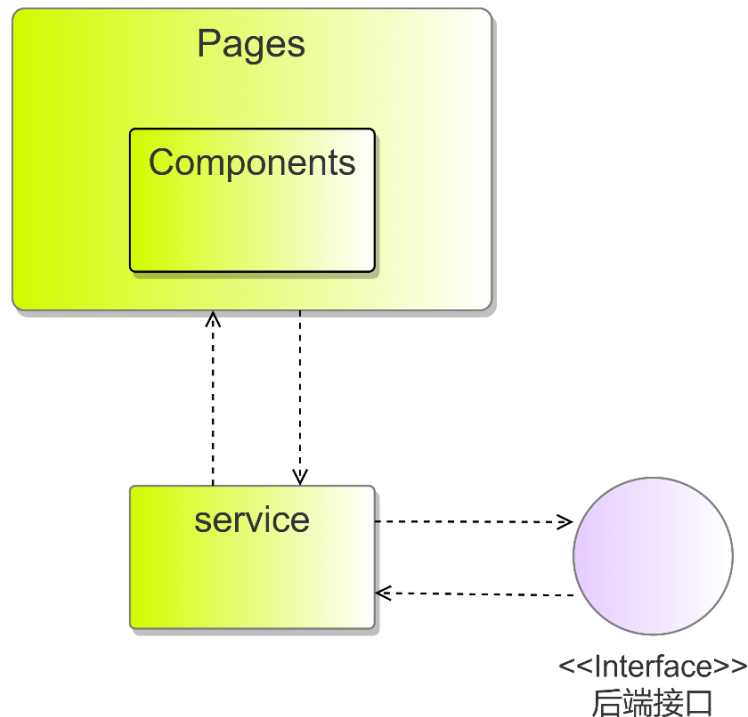


图 2.1.3 前端架构设计

如上述架构图 2.1.3 所示，展示了一个前端程序的设计，主要分为以下几个部分：

最上层是 **Pages** 模块，表示整个应用程序的页面。每个页面包含一个或多个 **Components**（组件）。在页面模块下方是 **Service** 模块，表示前端的业务逻辑层。后端接口，右侧显示的是后端接口（<>）。**Service** 模块通过接口与后端通信，发送请求和接收响应。

整个架构中，各模块之间通过箭头标注了交互逻辑：**Pages** 与 **Service** 之间通过虚线箭头连接，表示页面调用服务来获取数据或执行操作。

2.2 详细设计

2.2.1 排课后端业务详细设计

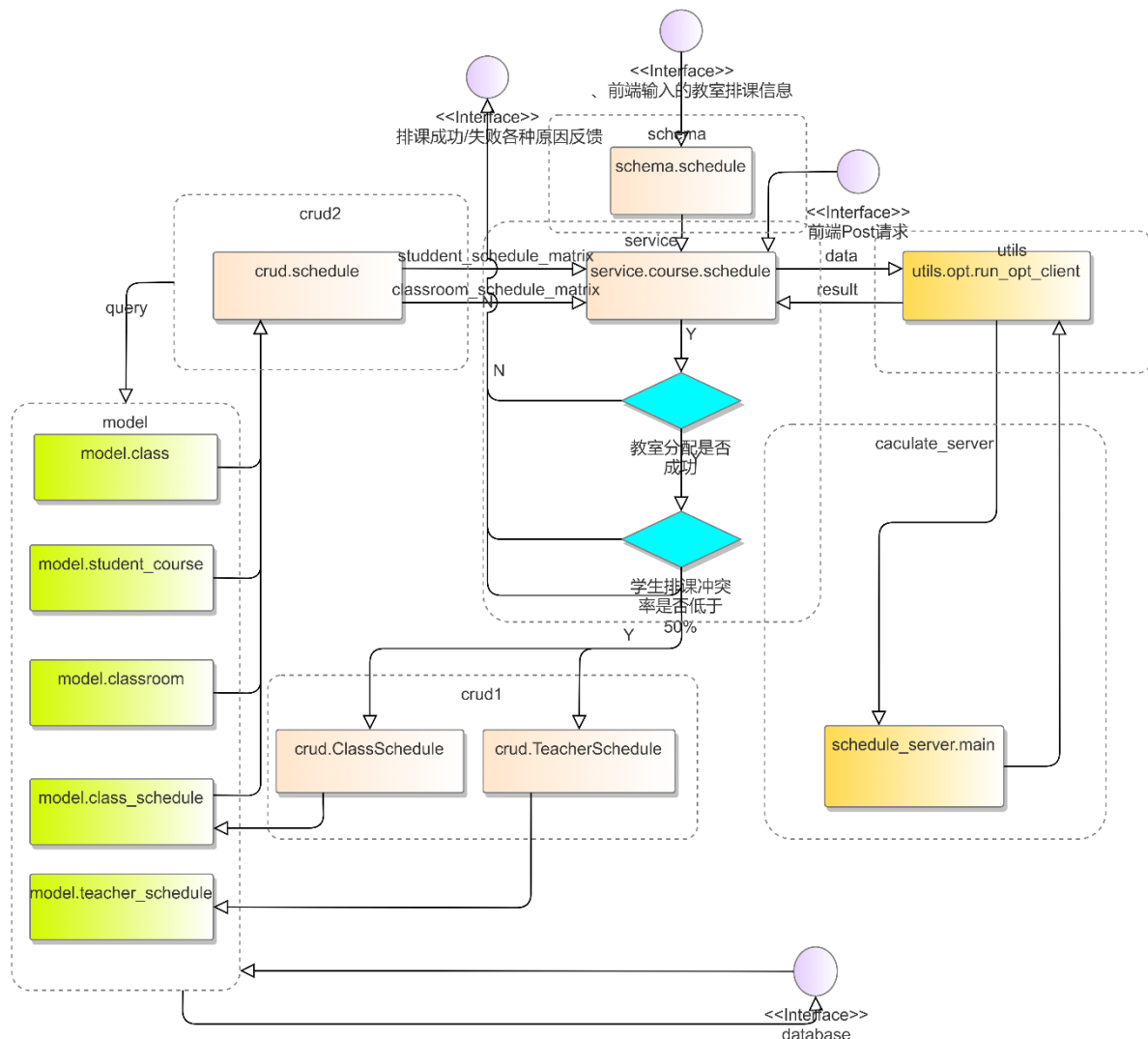


图 2.1.1 排课后端业务详细设计

排课流程描述

- (1) 前端用户输入筛选信息（偏好时间、偏好教室等），前端会将输入项中的信息通过 shcema 层全部送给 service.course_schedule，并发出 post 请求。
- (2) service.course_schedule 调用 crud.schedule 的两个 get 方法（get_student_schedule_matrix 和 get_classroom_schedule_matrix），从四个 model(model.class,model.classroom,model.class_schedule,model.student_course)模块中通过查询连接筛选得到选了该门课程的学生个人时间安排矩阵（5x5,哪一节课有课则为 1），以及学校实验室的时间安排矩阵。

- (3) service 层将数据整合后再发送给 utils 层中的 run_opt_client 函数，启动 schedule 计算的客户端 client。
- (4) 计算过程中，通过 crud.schedule 查询排课矩阵，利用 crud.ClassSchedule 和 crud.TeacherSchedule 获取相关排课数据。
- (5) 将信息再次发送到 schedule_server 所在的服务端口，通过矩阵运算求解出排课信息的结果，包括是否成功，学生冲突率，排课时间等，将这些信息返回给 client 端，再将数据传送给 service 层。
- (6) service 解析数据，逻辑判断若教室分配成功且排课冲突率低于 50%，将排课信息提交到数据库。数据库中更新 model.class_schedule 和 model.teacher_schedule，完成最终排课。如果失败则返回错误信息。
- (7) 最终结果返回到前端，包括排课成功或失败的原因及提示信息。

排课计算

对于排课，建立 0/1 整数规划数学模型，该模型的目标是通过优化课程变量安排变量 x 和教师安排变量 y ，满足特定约束条件的情况下，最大化学生和教室的偏好，同时分配课程。

(1) 决策变量： $x_{i,j}$ 是二进制变量。表示第 i 天的第 j 时段是否有课程被分配。 $y_{i,j}$ 是二进制变量。表示第 j 个教室是否被分配。 z_j 是连续变量。表示与教室分配相关的线性关系。

(2) 目标函数：目标函数由冲突率与偏好率两个部分组成。其中冲突率表示课程安排与学生课程安排的冲突比：

$$f_w = \frac{1}{I} \sum_{k=1}^I \sum_{i=1}^M \sum_{j=1}^N w_{i,k}[i,j] \cdot x_{i,j}$$

其中 I 是学生总数， M 表示天， N 表示一天的时间段。下面是课程偏好率：

$$f_{pref} = \frac{1}{M \cdot N} \sum_{i=1}^M \sum_{j=1}^N w_{pref} \cdot x_{i,j}$$

其中 M 表示天， N 表示一天的时间段。汇总目标函数得到：

$$\min f = f_w + f_{pref}$$

(3) 约束条件：对于教室分配约束，对于每个教室 j 有：

$$\left\{ \begin{array}{l} z_j \leq \sum_{i=1}^M \sum_{k=1}^N w \cdot w_j[i, k] \cdot x_{i,k} \\ z_j \leq y_j \cdot 10^6 \\ z_j \geq \sum_{i=1}^M \sum_{k=1}^N w \cdot w_j[i, k] \cdot x_{i,k} - (1 - y_j) \cdot 10^6 \end{array} \right.$$

对于教室数量进行约束：

$$\sum_{j=1}^J y_j = N_{classroom}$$

对于课程分配总数约束：

$$\sum_{i=1}^M \sum_{j=1}^N x_{i,j} = N_{class}$$

对于教室分配容量约束：

$$\sum_{j=1}^J z_j \leq 0$$

使用 CBC 求解器进行求解该优化模型进行排课技术。

2.2.2 选课后端业务详细设计

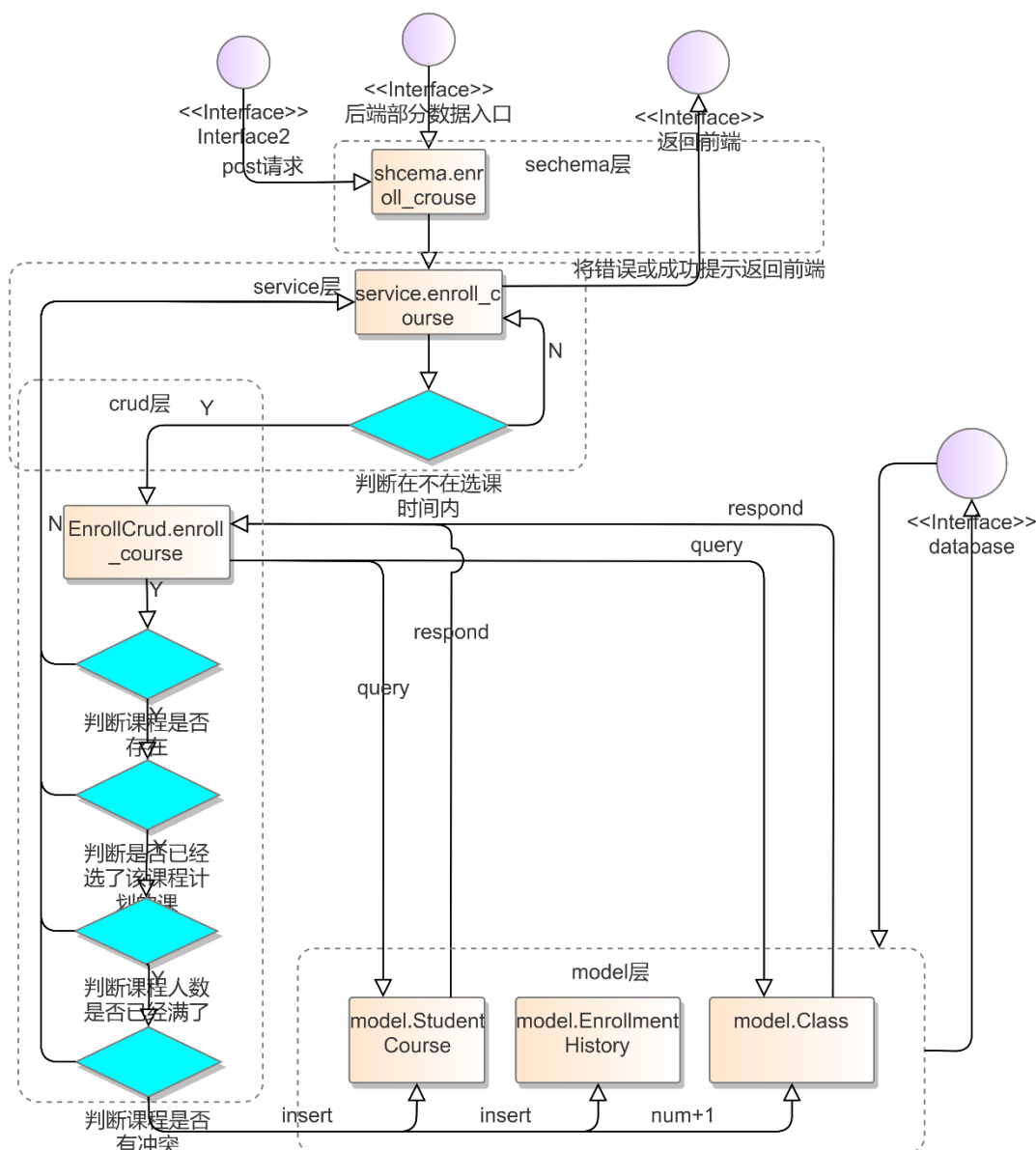


图 2.2.2 选课后端业务详细设计

选课流程描述:

- 整体上符合前面提到的后端架构设计，数据流从 schema 层，service 层，crud 层到 model 层，层层递进，shcema.enroll_course 负责将 class_id 从传送给 service.enroll_course, service.enroll_course 获得 user_id, class_id, 以及系统设置的选课时间段，如果判断此时在选课时间范围内，则将 user_id, class_id, 以及选课时间 time 输入 EnrollCrud.enroll_course 中。
- EnrollCrud.enroll_course 会从数据库的 student_course(sc 表)以及 class 表获得一系列信息，通过一些筛选和处理，来判断 class_id 对应的 class 是否存在，判断该学生是

不是以及选了相同 class_plan(课程计划)的 class,判断这门 class 是否还有容量可供选择,判断学生课表是否在该 class 的上课时间段由时间冲突。

- (3) 如果满足以上一系列要求,那么则将 sc 记录插入 student_course 表,将选课记录插入 enrollment_history 表,将 class 表中该门选课人数加一。最后统一提交数据库,并返回前端选课成功消息。如果不满足上述任何一点,都会将错误返回给 service 层,再由 service 层统一将错误返回给前端。

2.2.2 AI 对话后端业务详细设计

智能化的排课选课系统用户手册对话助手基于 LangChain 的 RAG(检索增强生成)技术构建,为用户提供高效、精准的支持服务。该助手的核心功能是将系统用户手册、常见问题解答和技术文档整合为一个语义检索和自然语言生成相结合的问答系统。

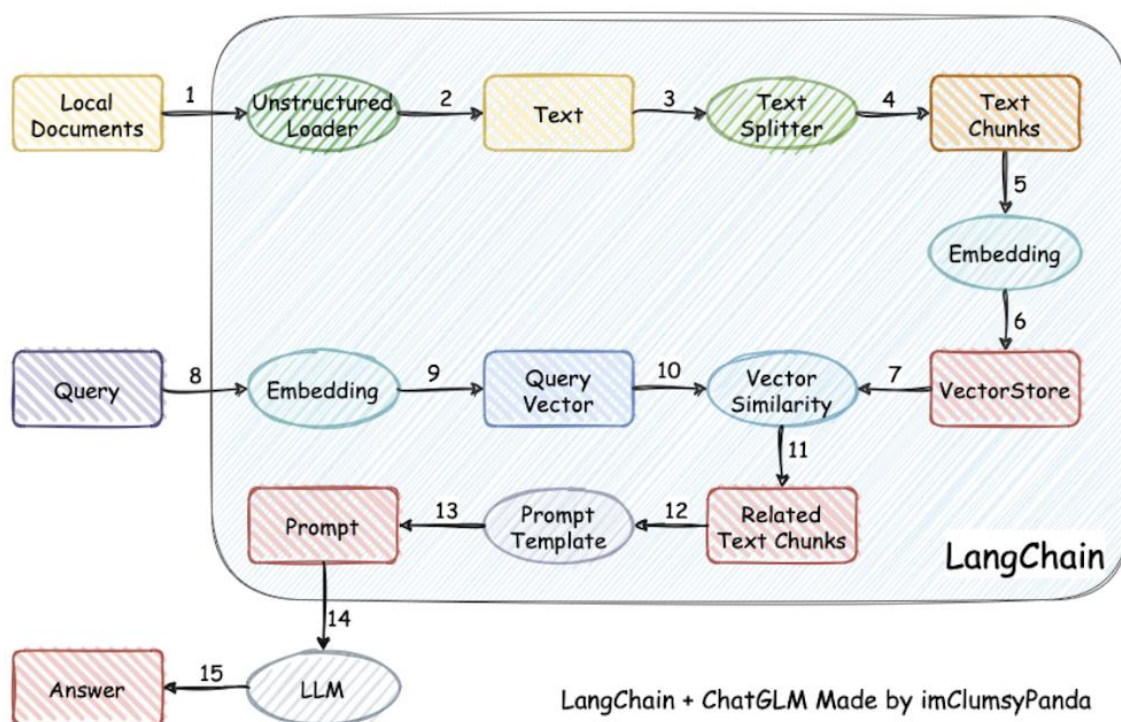


图 2.2.3 AI 对话后端详细设计

2.3 数据库表结构设计

2.3.1 概念结构设计

在本设计中，我们基于学生选课及课程安排系统的需求，抽象出了多个实体，如学生、教师、课程计划、课程班级等，同时定义了这些实体的属性和它们之间的联系。通过概念结构设计，我们能够清晰地描绘出数据库的整体框架，为后续的逻辑设计和物理设计提供坚实的基础。

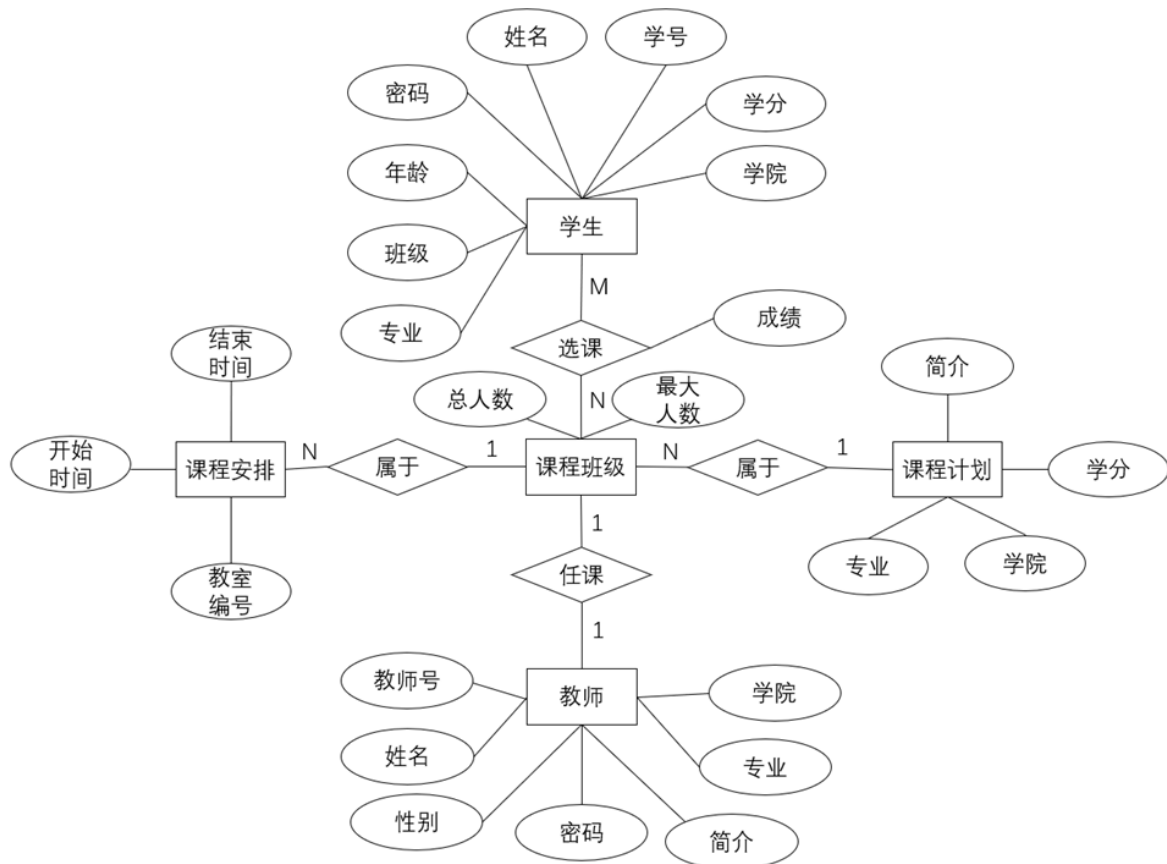


图 2.3.1 选课/排课系统 E-R 图

根据需求分析形成的数据字典和数据流图，抽象得到的实体有：

学生（学号，姓名，性别，密码，年龄，班级，专业，学院，身份证号，学分）

教师（教师号，姓名，密码，性别，学院，专业，简介）

课程计划（课程计划 id，课程名称，课程简介，学院，专业，学分）

课程班级（课程班级 id，任课老师，课程计划 id，总人数，最大选课人数）

课程安排（课程计划 id，开始时间，结束时间，教室编号）

成绩（学生 id，课程计划 id，成绩）

选课表（学生 id，课程班级 id）

实体之间的联系如下：

一个学生可以选多门课程，一个课程可以多个学生选。

一个老师可以教多门课程班级，一个课程班级只有一个主讲教师。

一个课程班级属于一个课程计划，一个课程计划有多个课程班级。

一个课程班级由多个课程安排构成，一个课程安排对应一次课程班级上课。

一个学生在一个课程只有一个成绩。

学生选课及课程安排系统 E-R 图如图 2.3.1 所示。

2.3.2 逻辑结构设计

逻辑设计的任务就是把概念设计阶段设计的 E-R 图转换为与选用 DBMS 产品所支持的数据模型相符合的逻辑结构。

(1) 实体转换为关系模式，生成相应的关系模式：

学生(Student):

学生(学号, 姓名, 密码, 年龄, 班级, 专业)

课程计划(CoursePlan):

课程计划(课程编号, 简介, 学分, 专业, 学院)

课程班级(CourseClass):

课程班级(课程班级编号, 最大人数, 总人数, 课程编号)

课程安排(CourseSchedule):

课程安排(课程安排编号, 开始时间, 结束时间, 教室编号, 课程班级编号)

教师(Teacher):

教师(教师编号, 姓名, 性别, 密码, 学院, 专业, 简介)

(2) 关系映射

选课(Student-CourseClass):

选课(学号, 课程班级编号, 成绩)

课程班级与课程计划(CourseClass-CoursePlan):

已包含在课程班级关系中，通过课程编号与课程计划关联。

课程安排与课程班级(CourseSchedule-CourseClass):

已包含在课程安排关系中，通过课程班级编号与课程班级关联。

任课(Teacher-CourseClass):

任课(教师编号, 课程班级编号)

(3) 用户子模式定义

根据具体的用户需求定义子模式。

学生子模式:

学生(学号, 姓名, 班级, 专业)

选课(学号, 课程班级编号, 成绩)

课程班级(课程班级编号, 课程编号)

课程计划(课程编号, 简介, 学分)

教师子模式:

教师(教师编号, 姓名, 学院, 专业)

任课(教师编号, 课程班级编号)

课程班级(课程班级编号, 最大人数, 总人数, 课程编号)

课程安排(课程安排编号, 开始时间, 结束时间, 教室编号)

3 系统实现与测试

3.1 技术框架与开发环境选型

3.1.1 后端技术框架

选用技术框架: FastAPI 0.115.6

应用场景: FastAPI 提供高效的 RESTful API 开发能力, 课程信息的增删改查 (CRUD) 操作可以通过 API 实现, 支持管理员实时更新课程信息。

3.1.2 前端技术框架

选用技术框架: React ^18.3.1

应用场景: 提供课程信息管理的前端界面, 包括选课列表、登录注册功能, 以及课程信息编辑表单。

前后端开发工具: Visual Studio Code

包管理工具: 后端使用 pip 和 virtualenv, 前端使用 npm

版本管理: Git、GitHub

3.1.4 数据库系统

选用数据库：MySQL v8.0

3.1.5 部署环境

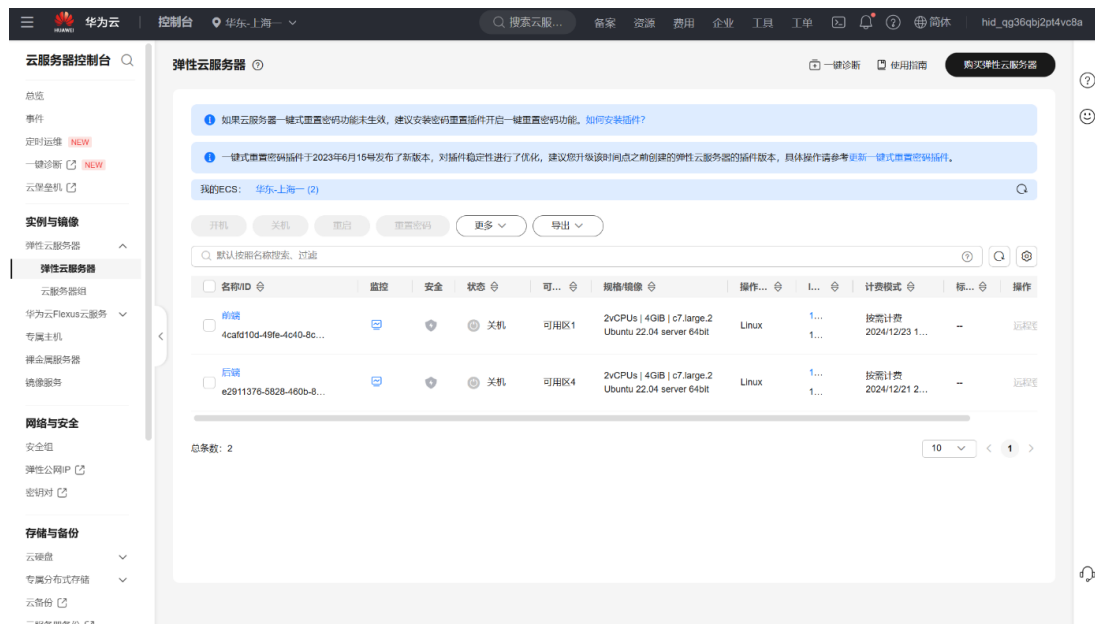


图 3.1.5 部署环境

软件依赖：Docker Compose

如图 3.1.5 所示，部署学生选课与实验排课系统，在项目根目录下执行 `docker-compose up -d` 命令，Docker Compose 会自动构建 Web 服务的镜像并启动 MySQL 和 Web 容器。

3.2 代码风格

3.2.1 代码结构

模块化设计：

- (1) 方法分工明确，分别处理创建记录（`create`）、按 ID 查询记录（`get_by_id`）、分页查询（`get_by_id_paginated`）等功能。
- (2) 代码逻辑分离清晰：如分页查询的偏移计算、记录统计和数据提取分开处理。
- (3) 每个文件封装了特定的功能逻辑（如 `ClassPlanCrud` 处理课程计划，`ClassScheduleCrud` 处理课程安排），实现了职责分离。
- (4) 从 `service` 到 `crud`，从 `crud` 到 `model`，前端与后端交互结构明确，后端和数据库交互结构明确。

3.2.2 特定语言规范

采用 Python PEP8 编码规范，编码规范链接：<http://legacy.python.org/dev/peps/pep-0008/>

3.2.5 语言规范检查

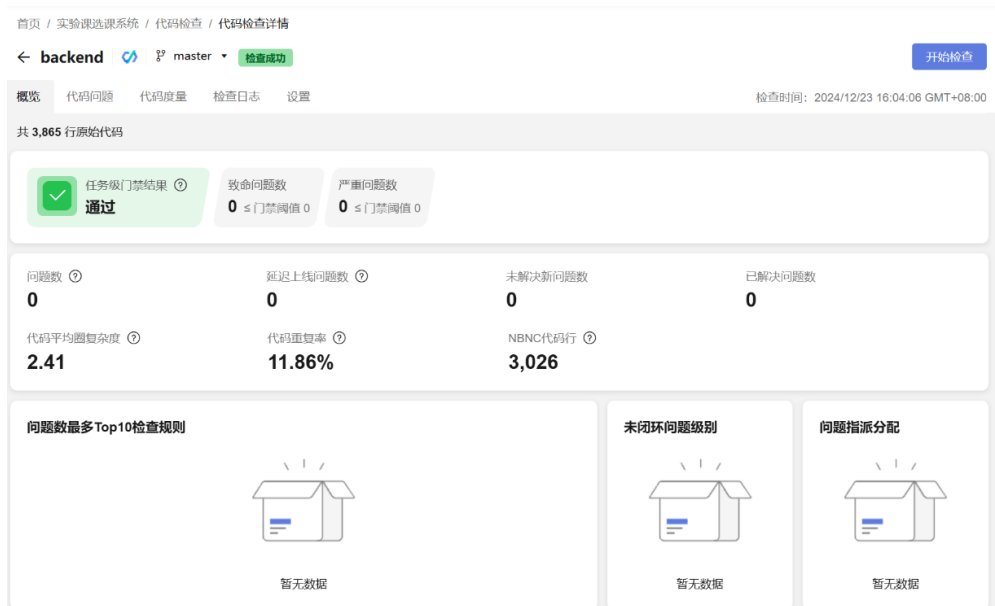


图 3.2.1 代码规范检查

借助华为云平台进行代码检查，可知当前代码共计 3,865 行，通过语言规范检查，未发现严重、一般或次要问题，整体质量较高。代码重复率为 11.86%，圈复杂度平均值为 2.41，建议针对重复代码部分进行优化，以提升代码的可维护性和质量。未决问题数为 0，符合语言规范要求，无需进一步调整。

3.3 软件系统实现界面

3.3.1 主界面



图 3.3.1 主界面截图

主界面功能描述:

(1) 教务系统介绍

页面标题“教务系统”直观地向用户传达了平台的用途，即与教学管理相关的服务，如课程管理、成绩查询、学籍管理等。

(2) 能助手或客服功能

右下角的机器人图标代表智能客服功能，用户可以通过它获取帮助，如解答常见问题或进行问题反馈。

3.3.2 登录界面



图 3.3.2 登录界面截图

登录界面功能描述:

(1) 标题和提示

页面顶部显示“登录”标题，并附有提示文字“登录以使用完整功能”，引导用户进行登录操作。

(2) 用户角色选择

在登录框的顶部有一个下拉框，默认为“学生登录”。用户可以通过此功能选择不同的登录角色（例如：教师或管理员）。

(3) 主要功能按钮

注册按钮：位于登录按钮左侧，用户可通过点击跳转到注册页面创建新账户。

登录按钮：蓝色按钮，用于提交用户的登录信息并进入系统。

3.3.3 注册界面



图 3.3.3 注册界面截图

(1) 标题和提示

界面顶部显示“注册”标题，配有提示文字“在此注册新账号”，明确该页面的功能是创建新账户。

(2) 用户角色选择

界面顶部提供一个下拉框，默认选项为“学生注册”。用户可以通过此下拉框选择角色类型（例如：教师、管理员等），以便注册相应类型的账户。

(3) 功能按钮

登录按钮：提供快速跳转回登录页面的功能，适用于已有账户的用户。

注册按钮：蓝色按钮，用于提交注册信息，完成新账户的创建。

3.3.4 选课界面



图 3.3.4 选课界面截图

(1) 侧边栏:

页面左侧有一个侧边栏，提供了“选课”、“选课记录”、“课表”和“成绩”四个功能模块的快速入口。

(2) 搜索和筛选:

页面顶部提供了搜索框和筛选选项，用户可以通过输入课程名称、选择专业、学院、学分和课程类型来筛选课程。

(3) 课程列表:

页面的主要部分是一个课程列表，展示了可选课程的详细信息

3.3.5 选课记录界面

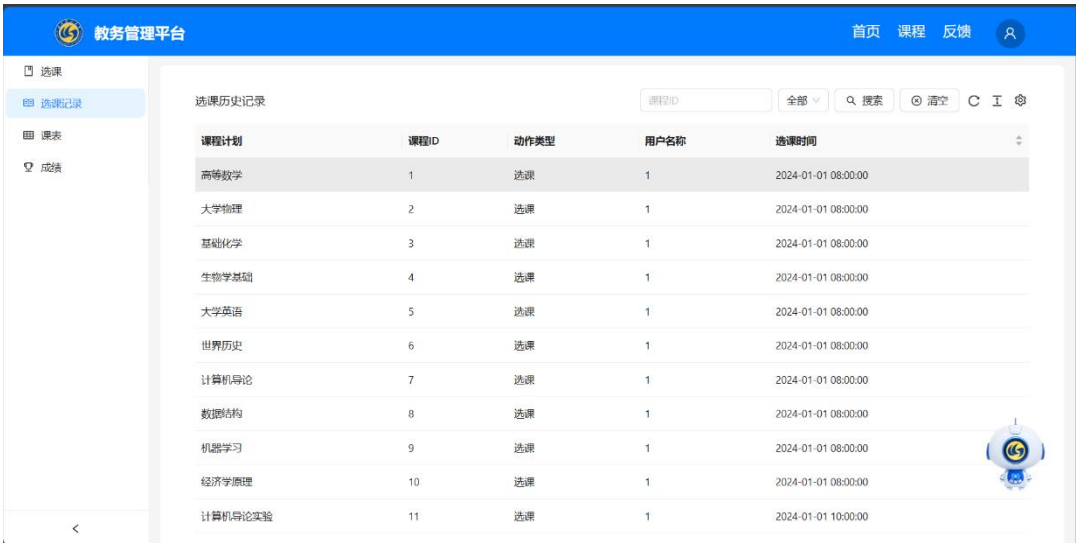


图 3.3.5 选课记录界面截图

(1) 选课历史记录表格：

页面的主要部分是一个表格，记录了用户选课的历史信息

(2) 操作按钮：页面顶部还有一些操作按钮，如“搜索”、“清空”，这些按钮提供了搜索的功能来管理选课记录。

3.3.6 课表界面

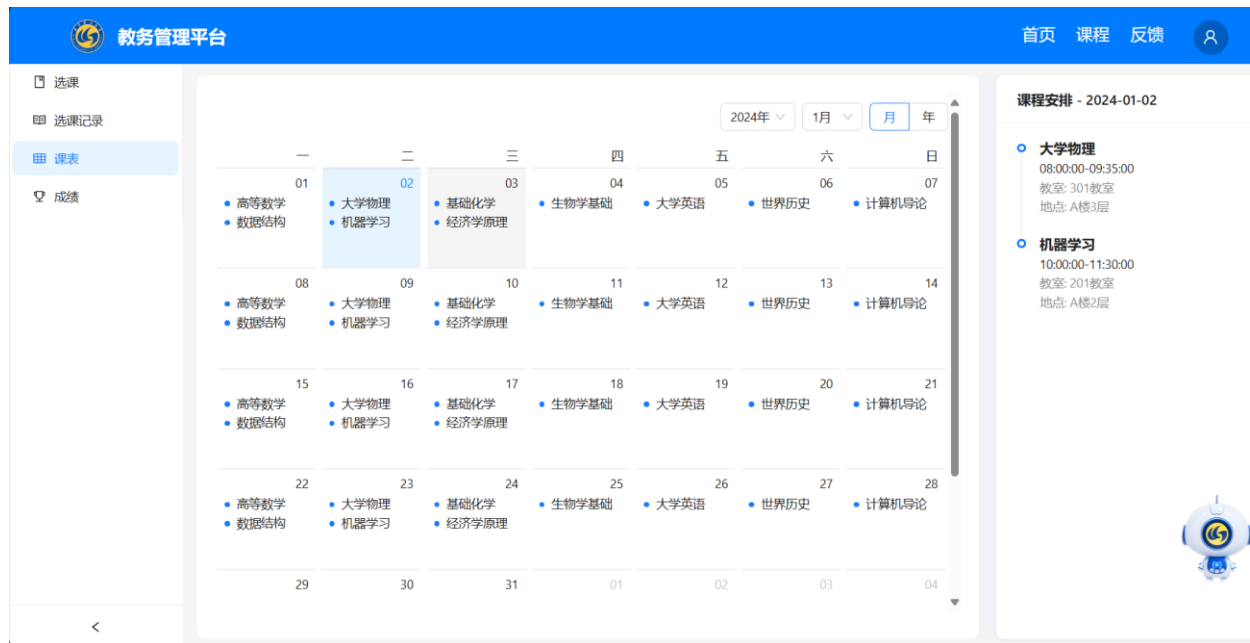


图 3.3.6 课表界面截图

(1) 日历视图：

页面的主要部分是一个日历视图，显示了用户在特定月份（这里是 2024 年 1 月）的课程安排。用户可以通过点击不同的日期来查看当天的课程。

(2) 课程安排：

在日历的每个日期上，用不同颜色的圆点表示当天的课程。点击圆点可以查看课程的详细信息。

(3) 课程详情：

在页面的右侧，有一个课程详情区域，显示了用户在选定日期（这里是 2024 年 1 月 20 日）的课程安排，包括课程名称、时间、地点和教室。

(4) 课程时间：

在课程详情区域，显示了课程的具体时间，例如“14:00:00-15:35:00”

(5) 教室信息：

课程详情区域还提供了课程的教室信息，例如“教室 201 教室”。

3.3.7 排课界面

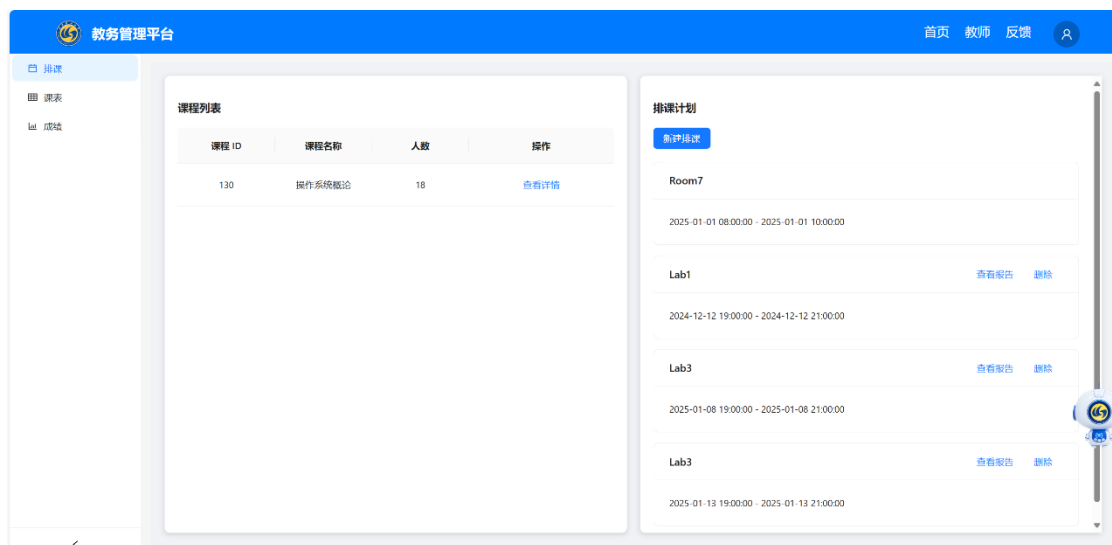


图 3.3.7 排课界面截图

(1) 课程列表展示：

界面左侧显示课程列表，包括课程 ID、课程名称、选课人数以及操作项，用户可以查看课程的详细信息，方便进行课程管理。

(2) 排课计划管理：

右侧区域展示已经创建的排课计划，包括排课的地点、时间范围等。用户可以直观查看当前排课的详细信息。

(3) 新建排课功能：

通过右侧的“新建排课”按钮，用户可以快速创建新的排课计划，为课程分配教室和时间。

(4) 查看报告与删除排课：

每个排课计划提供操作选项，包括“查看报告”用于查看排课相关的详细数据（如出勤率或反馈等），以及“删除”按钮用于移除指定排课计划。

3.3.8 助手问答界面

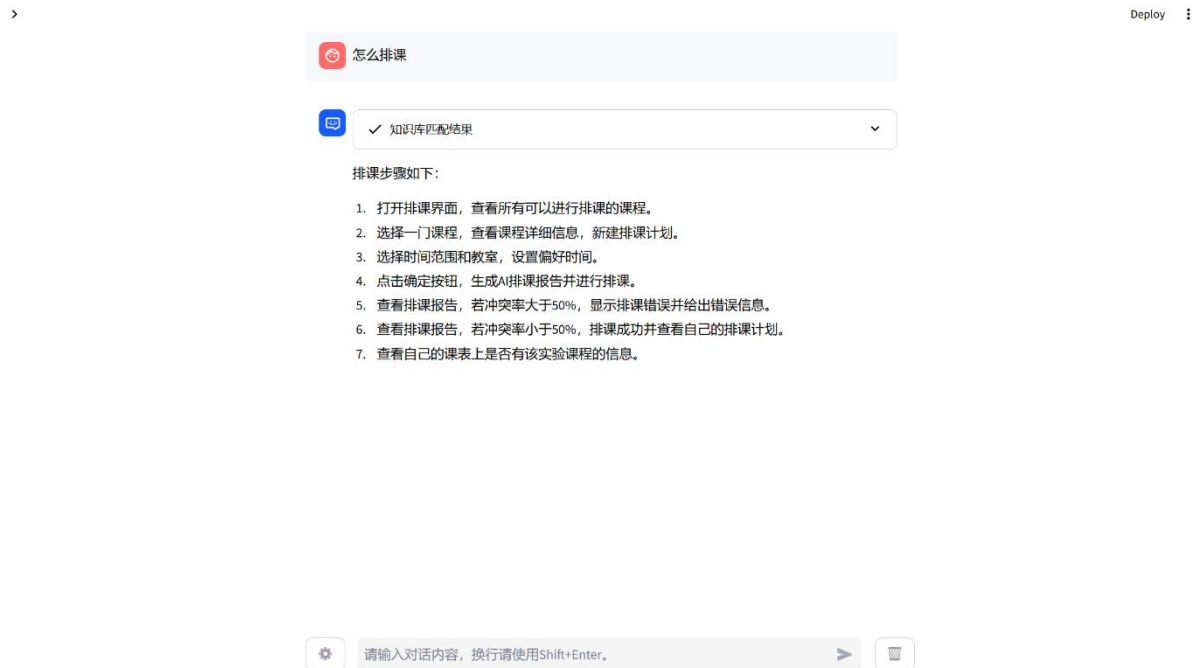


图 3.3.8 助手问题界面

该对话页面是一个智能化的问答界面,用户可以通过自然语言提问并获得精准的回答。例如,用户在输入框中提出“怎么排课”这一问题后,系统会根据用户的问题,通过语义检索和知识匹配,从知识库中提取相关的内容,并生成具体的回答。

页面的布局清晰,包含以下主要部分:

(1) 输入框:

位于页面底部,用户可以在输入框中自由输入问题。

(2) 提问与回答区域:

位于页面中央,显示用户的问题和系统生成的回答。用户的问题显示在左侧,系统的回答显示在右侧,回答的内容包括详细的步骤说明,确保清晰易懂。

(3) 知识库选择模块:

在回答中显示了“知识库匹配结果”标签,表明系统通过对话技术匹配到知识库中的内容,为回答提供了权威和可靠的来源。

3.4 测试

如表 3.4.1 所示为为登录注册功能的测试表,展示常见的测试用例与测试流程,通过对比预期测试结果与实际测试结果,说明本程序的正确性。

表 3.4.1 登录、注册功能测试表

测试用例名称	测试流程	输入数据	预期测试结果	实际测试结果	结论
登录学生账户	选择学生登录，输入学生邮箱密码点击登录按钮	学生邮箱、密码	如果学生邮箱密码不符合软件提示规范进行提示，密码错误提示登录失败原因，用户邮箱密码正确可以正常跳转首页，并在首页右上角显示用户信息	如果学生邮箱密码不符合软件提示规范进行提示，用户密码错误提示登录失败原因，用户名密码正确可以正常跳转首页，并在首页右上角显示用户信息	正确
登录教师账户	选择教师登录，输入教师邮箱密码点击登录按钮	教师邮箱、密码	如果教师邮箱密码不符合软件提示规范进行提示，密码错误提示登录失败原因，邮箱密码正确可以正常跳转首页，并在首页右上角显示用户信息	如果教师邮箱密码不符合软件提示规范进行提示，用户密码错误提示登录失败原因，邮箱密码正确可以正常跳转首页，并在首页右上角显示用户信息	正确
注册	点击注册按钮，在注册表单中输入信息点击注册后使用刚注册信息进行登录	用户名、邮箱、密码、确认密码	注册的新账户可以进行正常登录	注册的新账户可以进行正常登录	正确

3.5.2 选课功能的测试

如表 3.5.2 所示为为选课功能的测试表，展示常见的测试用例与测试流程，通过对比预期测试结果与实际测试结果，说明本程序的正确性。

表 3.4.2 选课功能测试表

名称	测试流程	输入	预期测试结果	实际测试结果	结论
选课功能验证	打开选课界面，查看所有可选课程列表。选择一门课程，点击“选课”按钮。检查系统是否提示该课程有无选课冲突（例如时间重叠）。如果没有冲突，提交选课请求。确认系统反馈是	无	用户能看到完整的可选课程列表。选课后，若无冲突，系统应提示“选课成功”。选课成功后，用户已选课程应立即更新。若用户再次选择已选课程，系统应提示“已选课程”或阻止重复选课。若存在选课冲	用户能看到完整可选课程列表。选课后，若无冲突，系统应提示“选课成功”。选课成功后，用户已选课程应立即更新。若用户再次选择已选课程，	正确

否显示“选课成功”消息。查看选课成功后，课程是否出现在用户已选课程列表中。	突（如时间重叠、学分超限等），系统应准确提示并阻止选课。	系统应提示“已选课程”或阻止重复选课。若存在选课冲突（如时间重叠、学分超限等），系统应准确提示并阻止选课。
打开排课界面，查看所有可排课程列表。选择一门课程，查看详细信息。新建排课计划，选择课程时间范围和教室，设置偏好时间。提交排课请求。确认系统反馈是否显示“排课成功”消息。查看排课成功后，课程是否出现在用户课程列表中。	教师可以看到完整的待排课课程列表。排课后，若无冲突，系统应提示“排课成功”。排课后，用户课程表应立即更新。排课后，可以查看对应的排课报告。若存在冲突，系统应提示“排课失败”并阻止排课。	教师可以看到完整的待排课课程列表。排课后，若无冲突，系统应提示“排课成功”。排课后，用户课程表应立即更新。排课后，可以查看对应的排课报告。若存在冲突，系统应提示“排课失败”并阻止排课。

4. 个人具体工作

如图 4.1 表示个人具体工作。

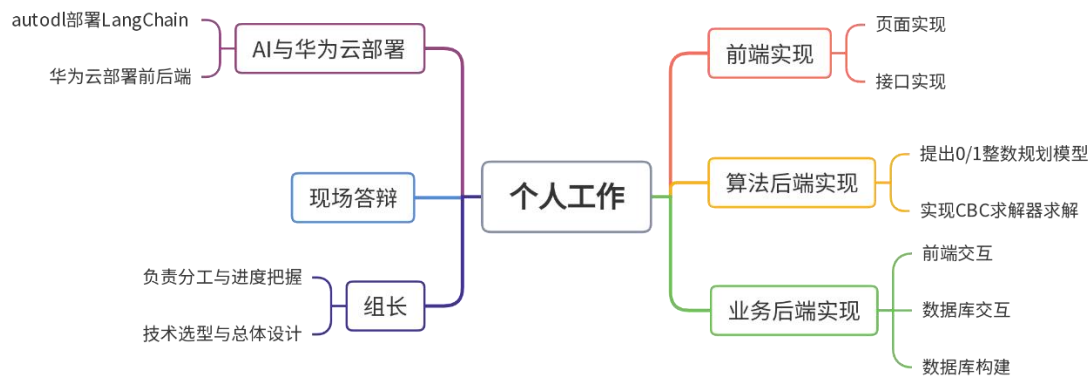


图 4.1 个人具体工作

- 组长：**决定项目的技术栈，包括前端框架、后端框架、数据库技术以及 AI 技术。设计系统的总体架构，包括前端、后端、数据库和 AI 助手的集成方案。制定代码规范和开发流程，确保团队遵循统一的开发标准。
- 答辩演讲：**在答辩过程中，回答了关于系统 AI 体现、排课算法实现和系统延迟检测等关键问题。

3. **算法后端实现：**建立 0/1 整数规划数学模型，优化课程变量安排和教师安排，满足特定约束条件，最大化学生和教室的偏好。负责复杂任务的处理，包括基于 CBC Solver 的数学优化算法，用于智能排课模型求解。通过分布式计算技术，提升了排课计算的性能和资源分配的效率。
4. **AI 助手部署和华为云部署：**引入基于 LangChain 技术的智能对话助手，实现智能用户对话功能，为用户提供关于选课和排课的实时支持与操作指导。利用华为云 CodeArts 平台进行高效开发与代码质量检查，优化了代码结构和规范性，同时确保系统在功能与性能上的稳定性。使用 Docker 技术实现服务的容器化部署，简化了部署流程，提高了系统维护效率和跨平台运行能力。
5. **前端实现：**使用 React 框架结合 Antd 组件库进行前端开发，构建用户友好的交互界面。开发了包括主界面、登录界面、注册界面、选课界面、选课记录界面、课表界面、排课界面、反馈界面和助手问答界面等多个页面。支持学生进行课程筛选、选课操作以及课表查询，同时提供教师排课界面和管理员管理界面。通过 API 与后端服务进行数据交互，实现动态内容的加载和用户操作的响应。
6. **业务后端实现：**使用 FastAPI 框架构建 RESTful API，处理核心业务逻辑。实现了学生管理、教师管理、管理员管理、用户信息管理和用户反馈等模块。负责转发请求、运行业务逻辑并管理数据存储，包括选课记录管理、课程冲突检测、排课结果处理等。通过 Sqlalchemy 实现 MySQL 数据库交互。

5 实训总结

5.1 开发亮点

如图 5.1.1 展示本项目的开发亮点，分为华为云 CodeArts 使用，异步技术使用，Docker 技术使用与 AI 分布式计算。

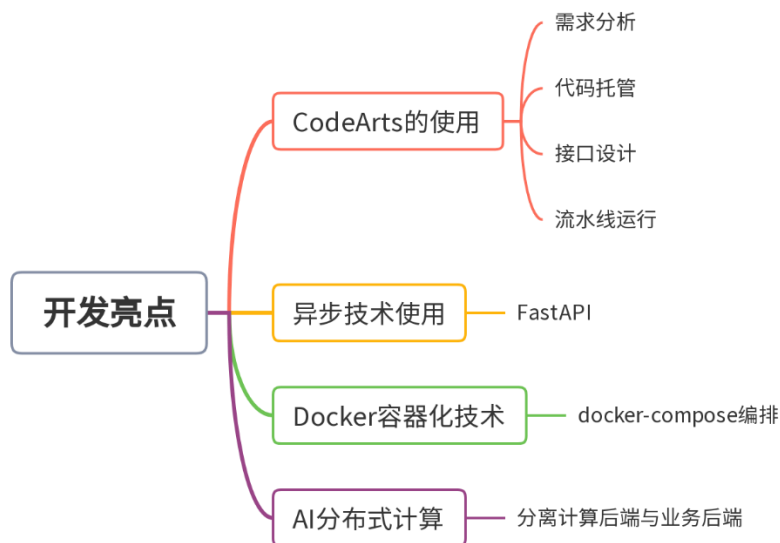


图 5.1.1 开发亮点

5.1.1 基于 CodeArts 的开发

借助华为云 CodeArts，搭建选课与排课项目，如图 5.1.2 所示：

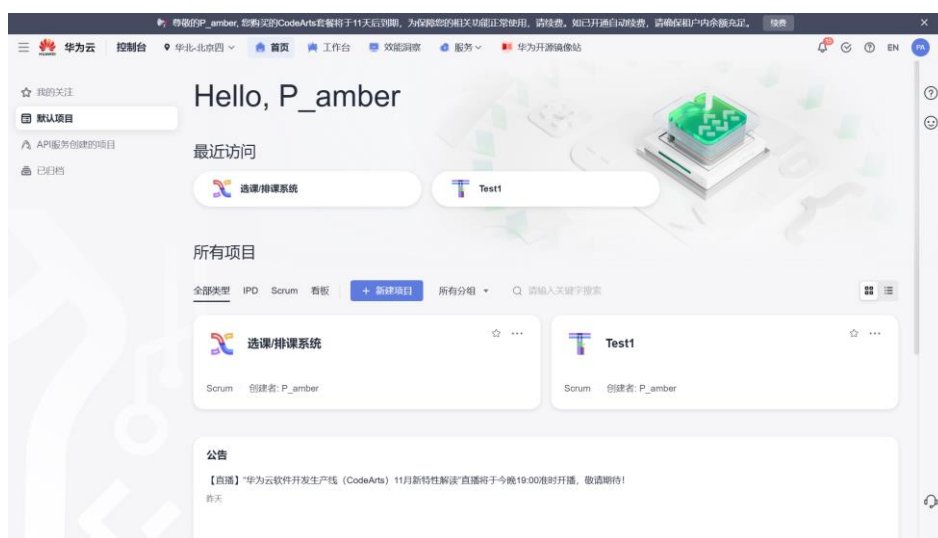


图 5.1.2 选课/排课系统

借助华为云的 CodeArts API,可以快速搭建 API,如图 5.1.3 展示本团队搭建的 API。这个界面是选课排课系统的 API 管理平台,设计了多个与课程管理、用户信息更新等相关的接口,支持前后端开发人员快速查看、测试和维护 API。该平台结构清晰、功能全面,适合高效管理复杂系统的 API 开发与协作。

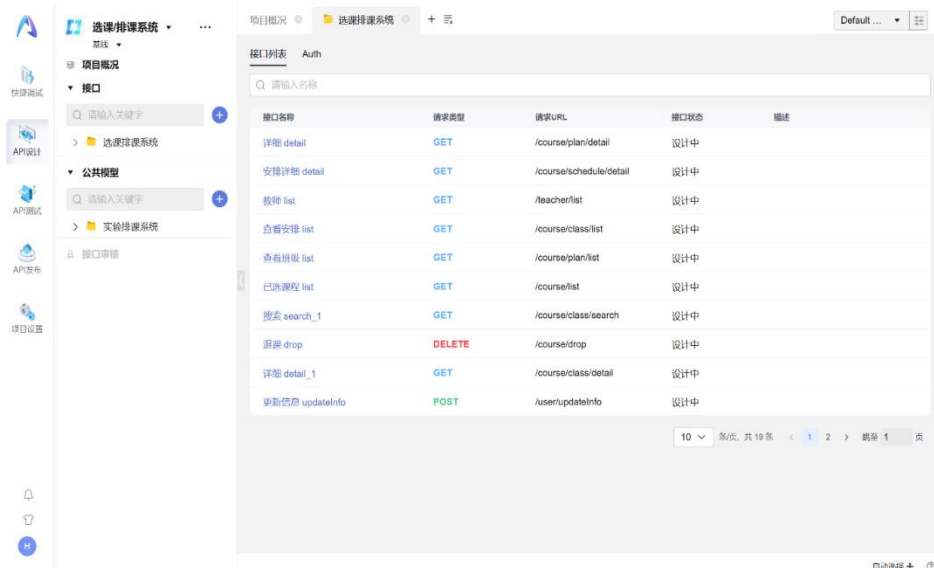


图 5.1.3 API 管理

同时将代码上传到华为云平台,借助华为云代码托管代码,图 5.1.4 展示了选课/排课系统后端代码的核心模块与整体组织方式,提供了高效的代码托管和协作环境,适合开发团队进行代码的集中管理与版本控制。

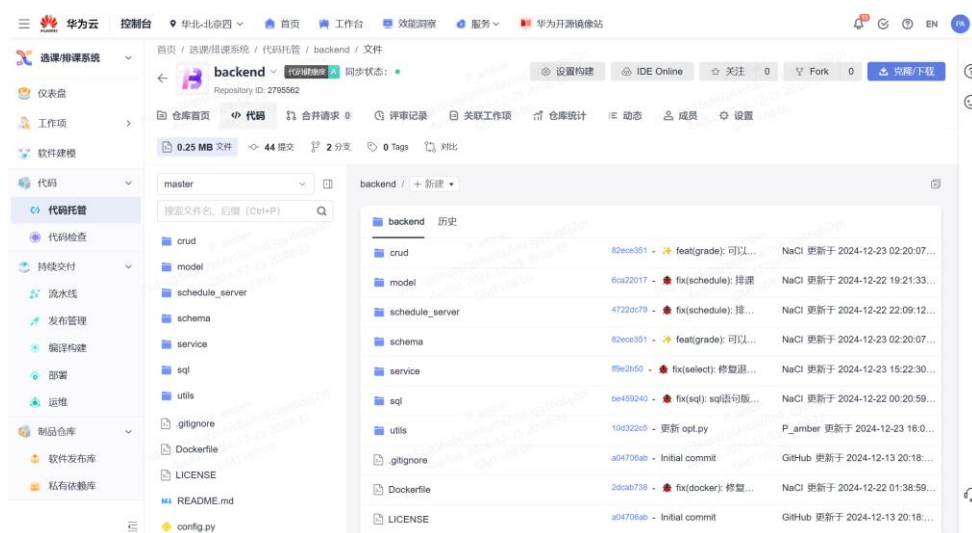


图 5.1.4 华为云代码托管

如图 5.1.5 所示为使用流水线进行运行应用。



图 5.1.5 流水线运行

5.1.2 异步技术

在本实验课排课系统中，异步技术被引入作为系统设计的重要组成部分，旨在提升系统的并发能力和响应速度。通过异步技术，系统可以在处理复杂的排课逻辑和数据操作时，充分利用服务器资源，减少阻塞等待时间，为用户提供更加流畅的使用体验。

应用示例：

(1) **用户选课请求：**学生提交选课请求后，系统立即返回请求已收到的确认信息，同时在后台异步处理冲突检测和课程分配。

(2) **排课进度查询：**管理员查看排课进度时，异步技术允许系统实时更新排课状态，而无需等待全部任务完成

(3) **大数据分析 & 优化：**系统通过异步任务队列对历史排课数据进行分析，以生成更优化的排课方案，同时不影响当前用户的操作。

技术优势总结：

通过异步技术的引入，实验课排课系统在性能、效率和用户体验上得到了显著提升，能够更好地满足高校复杂排课需求，为师生和管理者提供可靠的服务支持。

5.1.3 容器化 docker 技术

在本实验课排课系统的开发和部署过程中，Docker 技术被广泛应用，成为实现高效开发、快速部署和稳定运行的重要支撑工具。通过 Docker，我们能够以轻量级容器的形式打包应用程序及其依赖，确保系统在不同环境下的运行一致性和高效性。

应用示例：

(1) **开发环境容器化：**开发者通过拉取统一的 Docker 镜像快速搭建开发环境，镜像中预装了 Python、SQLAlchemy 和其他必要工具。

(2) **一键部署**：使用 Docker Compose 编排后端、前端和数据库服务，开发者只需运行 `docker-compose up`，即可启动整个系统。

技术优势总结：

通过引入 Docker 技术，实验课排课系统实现了开发、测试和部署的高效协作，并显著提高了系统的可扩展性和稳定性，为用户提供更优质的服务。

5.1.4 AI 分布式计算技术

由于 AI 计算耗时，在本智能 AI 排课系统的设计与实现中，分布式计算技术被广泛应用，成为提升系统性能、优化资源分配、增强可靠性的重要支撑架构。通过分布式计算，我们将排课计算与 AI 会话计算分离到不同的服务器上运行，充分发挥了分布式架构在并发处理和容错能力上的优势。

分布式计算是一种将复杂任务分解为多个子任务，并在不同的节点上同时执行的计算模式。各节点通过网络协作完成计算任务，具有高并行性和强扩展性。在我们的系统中，这种技术应用于核心模块，使排课计算与 AI 交互更高效、更稳定。

分布式计算在本系统中的应用：

(1) 计算任务分离：

1) **排课计算模块**：负责复杂规则的解析与排课逻辑的执行。配置专用服务器，搭载高性能硬件，支持密集型计算任务。

2) **AI 会话模块**：处理用户的实时交互请求。配置低延迟、高带宽的服务器，确保交互的流畅性和及时性。

通过任务分离，排课与 AI 会话计算可以独立运行，互不干扰，提升整体性能。

(2) 并发处理与高效资源利用：

1) **任务并行执行**：排课计算与 AI 会话计算分别运行在独立的节点上，在同一时间处理不同的任务，提升了系统并发处理能力。

2) **资源分配优化**：根据任务特点分配服务器资源，例如：排课模块优先使用多核 CPU 与大内存，AI 模块优化网络响应与计算效率。

(3) 容错与高可靠性：

1) **模块化设计**：两个模块独立运行，即使其中一个发生故障，另一个模块仍可正常提供服务。

2) **任务转移机制**: 当某一服务器出现问题时, 系统可自动将任务分配到其他节点, 确保服务不中断。

应用示例:

(1) **排课任务高效计算**: 排课服务器在后台执行复杂计算, 即使任务量大, 也能迅速生成优化的课表。

(2) **实时 AI 交互**: 用户与 AI 模块的对话响应迅速, 无需等待排课结果完成即可进行其他操作。

(3) **高并发访问**: 多个用户同时请求时, 分布式架构支持快速响应, 无性能瓶颈。

技术优势总结:

(1) **高性能**: 任务分离与并行处理显著提升了计算效率。

(2) **高可靠性**: 独立模块运行避免单点故障对系统的影响。

(3) **高扩展性**: 灵活的分布式架构支持动态扩容, 满足未来业务需求。

通过分布式计算技术的应用, 我们的智能 AI 排课系统实现了资源的最优配置与系统性能的显著提升, 为用户提供了高效、可靠的排课与交互服务。

5.2 实训总结

5.2.1 开发中的挑战与解决方案

开发过程中面临了许多技术和业务上的挑战:

(1) 如何选定技术:

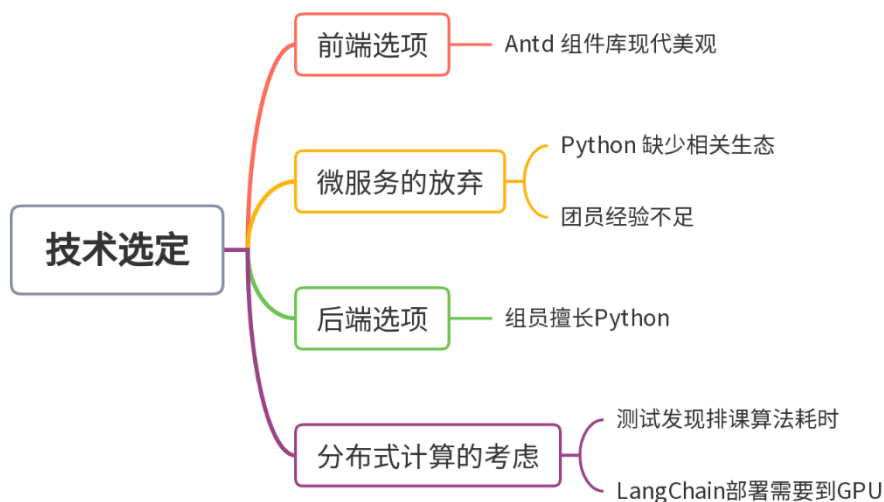


图 5.2.1 技术选定

如图 5.2.1 所示，在项目启动之初，对团队成员的技术水平和熟悉的技术栈进行了详细的调研，发现 Python 是团队成员普遍擅长的编程语言。考虑到项目的核心业务逻辑较为复杂，需要一个能够快速开发且易于维护的后端技术栈，Python 语言的简洁性使得 FastAPI 成为了一个理想的选择。FastAPI 以其高性能和异步特性脱颖而出，同时它的自动交互式文档也极大地方便了前后端的协作和 API 的测试。由于团队在数据库课程中已经接触过 MySQL，并且它是一个稳定且功能丰富的关系型数据库，因此选择 MySQL 作为数据存储解决方案。React 与 Ant Design：考虑到前端需要构建一个现代、响应式的用户界面，React 因其组件化和灵活性被选中。Ant Design 组件库提供了丰富的 UI 组件，能够加速开发进程，并且其现代的设计风格符合项目需求。虽然微服务架构能够提供更好的扩展性和灵活性，但考虑到 Python 在微服务生态中相对较弱，且团队对微服务架构的实践经验不足，最终决定不采用微服务架构。在开发测试阶段，发现排课 AI 计算非常耗时，影响了系统性能。AI 对话系统对计算资源要求较高，只有在高性能的 GPU 上才能勉强运行。为了解决性能瓶颈和硬件限制，决定采用分布式计算架构，将业务后端与计算后端分离，以提高系统的处理能力和响应速度。

（2）如何更好融合 AI:

- 1) **排课 AI 智能算法的实现：**开始提出了实现一个排课 AI 智能算法，旨在通过人工智能技术优化排课逻辑，提高排课效率和准确性。在提出构想后，进行了广泛的技术调研，最终建立 0/1 整数规划的排课模型，该模型能够处理复杂的排课约束，并找到最优的排课方案。
- 2) **AI Agent 应用的构想：**开始打造一个 AI Agent 应用，我希望通过这个应用提供更加个性化和智能化的服务。在探索 AI Agent 的概念时，发现虽然市场上对于此类应用的需求日益增长，但缺乏成熟的开源解决方案和参考案例。评估了当前的技术水平和团队的能力，考虑到开发一个 AI Agent 需要大量的数据支持和复杂的自然语言处理技术，这超出了项目的范围和预算。
- 3) **问答知识库 AI 的部署：**在权衡了技术可行性和项目需求后，决定利用 LangChain-chatchat 框架部署一个问答知识库 AI，以实现用户手册的自动问答功能。构建了一个包含系统操作手册和技术文档的知识库，使 AI 能够理解和回答用户关于系统使用的问题。通过这个问答系统，用户可以快速获得操作指导和问题解决方案，提升了用户体验和系统的易用性。

- 4) **系统的遗憾与未来展望:** 虽然问答知识库 AI 的部署是一个创新的尝试,但它并没有完全实现最初关于 AI Agent 的构想,这成为了系统的一个小遗憾。你计划在未来的项目中继续探索 AI Agent 的应用,随着技术的发展和开源资源的丰富,希望能够实现更加智能和个性化的 AI 服务。

(3) 如何完善团队分工:

- 1) **学习 DevOps:** 在项目初期,开始接触和学习 DevOps 的相关知识,理解到 DevOps 的核心在于促进开发(Dev)和运维(Ops)之间的沟通合作,以及实现软件开发生命周期的自动化。团队认同了 DevOps 能够提高软件交付的速度和质量,减少错误和提升系统的可靠性。
- 2) **寻找合适的工具:** 在理解了 DevOps 的理念后,团队开始寻找能够实现自动化集成、部署和测试的工具。在刘春老师的《软件工程》实验课程中,了解到华为云的 CodeArts 平台,它提供了一站式的 DevOps 服务,包括代码托管、持续集成、持续部署等。
- 3) **华为云 CodeArts 的应用:** 使用华为云 CodeArts 的代码托管服务,团队能够集中管理代码,实现多人协作开发,同时保证了代码的版本控制和追溯。通过 CodeArts 的持续集成/持续部署功能,团队实现了代码的自动化构建和部署,每次代码提交后自动运行测试,并部署到测试或生产环境。CodeArts 的流水线管理功能帮助团队可视化整个构建和部署流程,使得进度和状态一目了然,便于监控和管理。
- 4) **团队分工的优化:** 利用 CodeArts 平台的任务管理和协作功能,你更有效地分配任务给团队成员,并跟踪每个成员的工作进度。明确了开发人员、测试人员和运维人员的角色和职责,每个成员都能在 CodeArts 平台上找到适合自己的工作区域和工具。CodeArts 平台的沟通工具促进了团队成员之间的沟通 and 协作,特别是在代码审查和问题解决方面。

(4) 代码开发:

- 1) **复杂业务逻辑的处理:** 排课系统涉及的业务逻辑非常复杂,包括课程冲突检测、资源分配、用户权限管理等,这些都对代码的复杂性和可维护性提出了高要求。采用模块化设计,将复杂的业务逻辑拆分成独立的模块和函数,每个模块负责一部分逻辑,便于管理和维护。同时,编写详细的文档和注释,确保每个成员都能理解代码逻辑。

- 2) **代码质量和一致性：**团队成员的代码风格和质量参差不齐，导致代码一致性差，难以维护。制定统一的代码规范和审查流程，使用代码格式化工具和代码质量检查工具来自动化代码风格检查。定期进行代码审查会议，确保代码质量。
- 3) **代码后续维护：**为了实现代码后续维护，我将代码开源到 GitHub 社区，希望能为后续开发该类系统开发提供帮助。

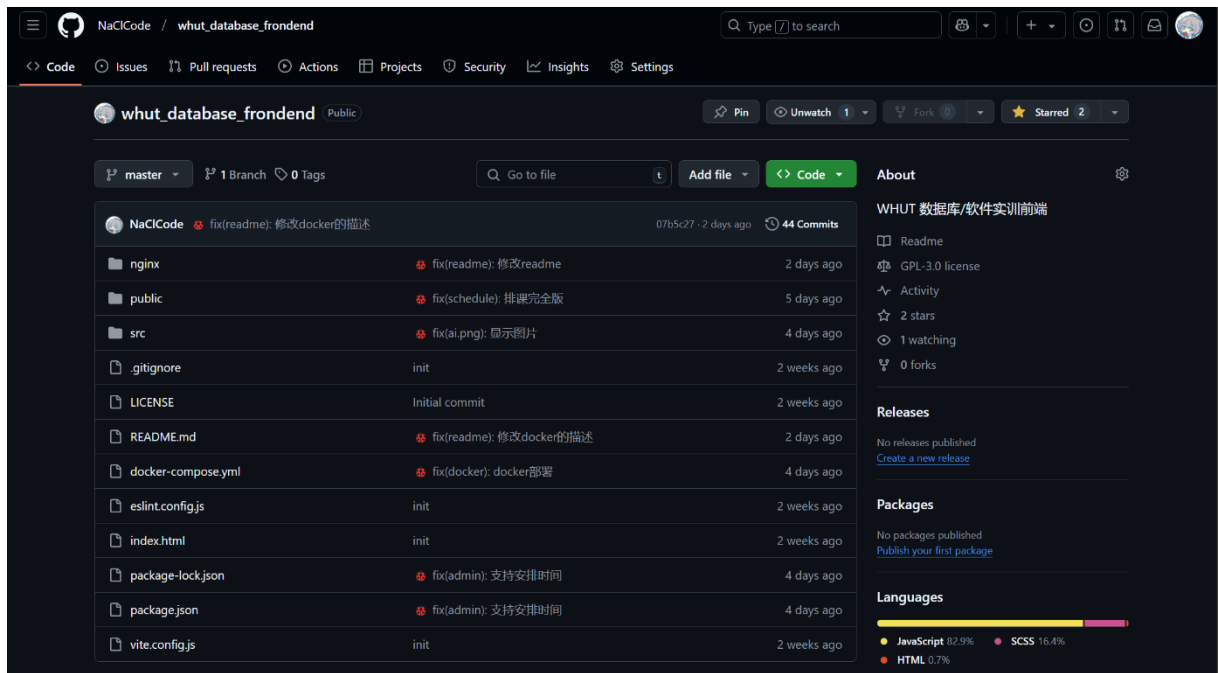


图 5.2.2 Github 社区开源

5.2.2 收获与总结

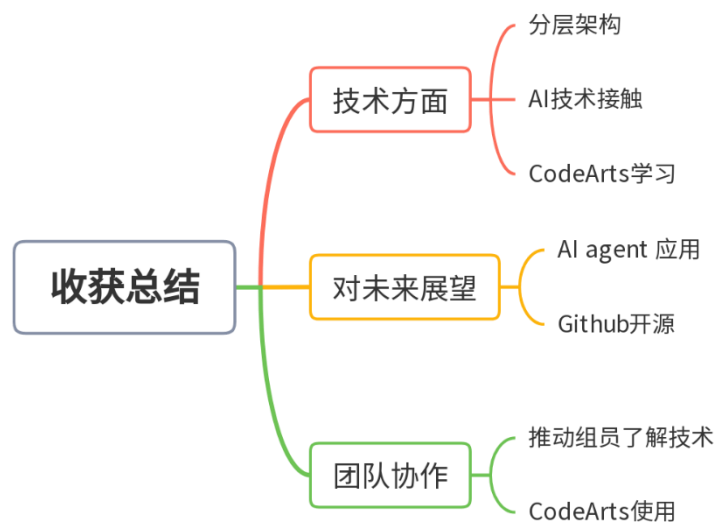


图 5.2.3 收获总结

- (1) **技术方面的收获：**本项目采用了分层架构设计，使得开发过程具有良好的逻辑性和扩展性。在后端设计中，充分利用了 FastAPI 框架的高性能和易用性，确保业务逻辑的快速响应与处理。前端采用 React 框架结合 Ant Design 组件库，不仅提升了开发效率，还实现了现代化的交互界面。利用 Docker 技术实现服务的容器化部署，进一步提高了系统的可移植性与部署效率。在排课算法设计中，采用 0/1 整数规划数学模型，结合 CBC Solver 优化算法，处理复杂的约束条件，实现智能化排课。将 AI 助手引入系统，通过 LangChain 技术实现问答知识库功能，为用户提供精准的操作指导与实时反馈。分布式计算架构的应用，使得排课计算与 AI 助手运行分离，优化了资源分配与系统性能。借助华为云 CodeArts 平台进行代码托管和质量检查，通过持续集成/持续部署功能，显著提高了开发效率和代码质量。
- (2) **团队协作与项目管理：**作为团队负责人，我深刻体会到合理的团队分工和协作对项目成功的重要性。通过制定开发流程、明确任务分配，并借助 CodeArts 平台的协作功能，团队成员能够高效完成各自的任务。在项目开发过程中，我学会了如何通过 DevOps 理念协调开发和运维的需求，优化了项目的交付流程。定期的团队会议和代码评审帮助我们及时发现问题并解决，确保了项目的进度和质量。在项目初期，团队成员对部分新技术（如 FastAPI、LangChain）不够熟悉。我通过推荐学习资料与分工学习，团队成员逐步掌握了这些技术，并成功应用于项目中。
- (3) **对未来的展望：**现在杨校长大力推进智能化教学，对于武理的一份子，我未来希望加入更多智能化功能，例如通过大数据分析用户选课偏好，进一步优化排课算法。深入学习 AI Agent 和微服务架构，将其融入系统开发中，以实现更高效的服务。我已将代码开源到 GitHub，期望能与更多开发者交流合作，为教育管理系统的技术进步贡献自己的力量。

本科生实践课程成绩评定表

姓 名	xxx	性 别	女
专业、班级	xxx		
课程设计题目：实验排课系统			
课程设计答辩或质疑记录：			
<p>1. 系统的 AI 怎么去体现的？</p> <p>系统通过建立 0/1 整数规划数学模型，优化课程变量安排和教师安排，以满足特定约束条件，最大化学生和教室的偏好，同时分配课程。这种模型能够处理复杂的排课逻辑，包括时间冲突检测、资源分配等。系统提供了一个基于 LangChain 的 RAG 技术的对话助手，它能够将用户手册和技术文档整合为一个语义检索和自然语言生成相结合的问答系统。用户可以通过自然语言提问，系统通过语义检索和知识匹配，从知识库中提取相关内容，并生成回答。</p> <p>2. 排课算法怎么实现的？</p> <p>设计目标函数，包括冲突率和偏好率两个部分，以最小化冲突并满足偏好。包括教室分配约束、教室数量约束、课程分配总数约束和教室分配容量约束。利用整数规划模型来求解排课问题，通过优化算法找到满足所有约束条件的最优解。排课计算在计算后端进行，使用 CBC Solver 进行数学优化和约束求解，以实现排课的自动化和智能化。</p> <p>3. 系统延迟检测怎么实现的？</p> <p>通过前端向后端发送 http 请求，计算发送与收到时间差得到延迟</p>			
成绩评定依据：			
报告评价指标	比例	得分	
报告正文内容完整，结构合理	50%		
答辩问题记录详实	20%		
格式规范	30%		
总分	100%		
最终评定成绩（以优、良、中、及格、不及格评定）			

指导教师签字：_____年 月 日