大连海事大学2024-2025（3）学期 序号：11

信息系统分析与设计课程设计

项目名称：手机课程安排

项目组名称：白金之星

项目组成员：

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 学号 | 任务 | 成绩 |
| 周俊龙 | 2220223561 | 需求分析，报告撰写，系统实现，界面设计 |  |
| 凌子杰 | 2220232253 | 设计绘图，报告撰写，数据流图设计，界面设计 |  |
| 孙世庆 | 2220222726 | 数据字典描述，报告撰写 |  |
| 魏国玉 | 2220223953 | 系统测试 |  |

项目组成员完成工作量：

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 学号 | 主要完成工作 | 占总工作量比例 |
| 周俊龙 | 2220223561 | 需求分析，报告撰写，系统实现，界面设计 | 30% | |
| 凌子杰 | 2220232253 | 设计绘图，报告撰写，数据流图设计，界面设计 | 35% | |
| 孙世庆 | 2220222726 | 数据字典描述，报告撰写 | 35% | |
| 魏国玉 | 2220223953 | 系统测试 | 5% | |

目录

[第1章 系统需求分析 1](#_Toc203038568)

[1.1 系统业务概述 1](#_Toc203038569)

[1.2 用户需求分析 2](#_Toc203038570)

[1.3 功能需求分析 2](#_Toc203038571)

[1.4 系统业务流图分析 3](#_Toc203038572)

[1.5 系统数据流图 3](#_Toc203038573)

[1.3.1 系统顶层数据流图 4](#_Toc203038574)

[1.3.2 系统第一层数据流图 4](#_Toc203038575)

[1.3.3 第二层数据流图 5](#_Toc203038576)

[1.6 系统数据字典描述 11](#_Toc203038577)

[1.7 系统用例图分析 17](#_Toc203038578)

[1.8 非功能需求分析 19](#_Toc203038579)

[1.9 需求分析小结 20](#_Toc203038580)

[第2章 系统设计 21](#_Toc203038581)

[2.1 系统功能设计 21](#_Toc203038582)

[2.1.1 用户管理模块 22](#_Toc203038583)

[2.1.2 学生中心 29](#_Toc203038584)

[2.1.3 管理员后台 35](#_Toc203038585)

[2.1.4 教师工作台 37](#_Toc203038586)

[2.2 输入输出设计 41](#_Toc203038587)

[2.2.1 用户层输入输出 41](#_Toc203038588)

[2.2.2 学生端输入输出 42](#_Toc203038589)

[2.2.3 教师端输入输出 42](#_Toc203038590)

[2.2.4 管理员后台输入输出 42](#_Toc203038591)

[2.3 数据库设计 43](#_Toc203038592)

[2.3.1 数据库概念模型设计 43](#_Toc203038593)

[2.3.2 数据库表结构设计（逻辑模型设计） 44](#_Toc203038594)

[2.4 信息编码设计 46](#_Toc203038595)

[2.4.1 A 类编码及规则 46](#_Toc203038596)

[2.4.2 B 类编码及规则 47](#_Toc203038597)

[2.4.3 C 类编码及其取值说明 47](#_Toc203038598)

[2.5 系统设计小结 47](#_Toc203038599)

[第3章 系统实现 48](#_Toc203038600)

[3.1 登录管理的实现 48](#_Toc203038601)

[3.2 主界面的实现 49](#_Toc203038602)

[3.3 个人中心的实现 50](#_Toc203038603)

[3.4 统计界面的实现 51](#_Toc203038604)

[3.5 用户管理界面的实现 52](#_Toc203038605)

[3.6 课程管理界面的实现 53](#_Toc203038606)

[3.7 学生选课界面的实现 55](#_Toc203038607)

[3.8 学生查看课表界面的实现 56](#_Toc203038608)

[第4章 系统测试 59](#_Toc203038609)

[4.1 系统测试环境 59](#_Toc203038610)

[4.2 系统功能测试 59](#_Toc203038611)

[结论与展望 65](#_Toc203038612)

# 系统需求分析

## 系统业务概述

手机课程安排系统面向高校教务场景，覆盖“资源维护—排课—选课—课表发布—统计反馈”全流程。系统将教务人员、教师与学生划分为三类核心角色，通过移动端与 Web 端统一入口完成信息收集、课程调度、冲突检测与数据呈现，实现资源高效利用与师生便捷交互。以下是简要的系统流程。

1. 教务管理员
2. 资源维护：登录后依次管理课程、教室、教师、教学班等基础数据。
3. 排课配置：在“课表配置”设置学期起始周、每日节次与时段。
4. 自动排课：选择教学班→设定每周节数→执行一键排课；系统返回时间-教室方案并同步冲突提示。
5. 冲突处理：查看冲突详情，手动调整或重新执行排课；确认无冲突后发布为正式课表。
6. 统计与通知：实时监控教室利用率、选课人数等指标，并向教师/学生推送变更通知。
7. 教师
8. 登录与身份验证：使用工号进入教师工作台，系统加载个人信息及授课教学班。
9. 教学班管理：查看任教课程列表，更新教学班容量限制，审核学生选课申请。
10. 课表查询：按周或学期查看自己的授课时间与教室，并可导出日历或分享。
11. 评价与反馈：课后查看学生评价，记录教学反思；接收管理员发布的排课调整通知。
12. 学生
13. 注册与登录：新生首次注册，填写学号、班级等信息后登录移动端主页。
14. 选课流程：浏览开课列表→系统实时提示容量与时间冲突→提交选课并即时获取结果。
15. 课表查看：自动生成个人课表，支持周视图/日视图切换及课程详情查看。
16. 通知与评价：接收教师/教务公告，课后对课程进行打分与评价。

上述三条流程在统一的冲突检测与权限控制框架下协同运行，确保数据一致与操作安全，为后续功能扩展提供清晰的业务脉络。

## 用户需求分析

1. 教务管理员需在手机上快速增删改课程、教室与教学班，界面应采用分组卡片+搜索，下拉两步即可完成资源维护。排课高峰期希望一键生成周课表并立刻得到冲突提示，系统应在结果页直观标注教室或教师冲突并提供调整按钮。首页仪表盘应实时展示统计图，并支持点击进入详情导出 Excel。
2. 教师需随时查看本学期个人课表及最近一周授课提醒，课表界面应支持周/日视图切换，点击课程可查看教室楼层与学生名单。在选课期希望实时掌握教学班容量变化，系统应自动推送“已满员”“低选课率”提示，并提供一键调整容量入口。
3. 学生选课时关注课程时间冲突、剩余名额和教师评价，系统应在课程卡片上直接显示冲突标记与剩余席位，并提供教师评分星级。希望课表自动同步到手机日历并支持自定义事项，课表页面应提供“一键同步”与“添加备注”功能。系统应通过消息中心送达变动信息，并允许学生一键确认已读。

## 功能需求分析

1. 身份认证与权限控制

提供注册、登录、注销及密码重置接口；采用角色-基于令牌的鉴权，精细化区分管理员、教师、学生访问范围。

1. 资源基础维护

课程管理——增删改查课程信息、学分与周课时；教室管理——维护教学楼、容量、设备等属性；教师/学生/教学班管理——批量导入、分页检索、状态启停。

1. 课表配置中心

学期起止日期与教学周自动生成；每日节次、统一课时长度和时段自定义；配置结果实时保存并供排课模块引用。

1. 排课管理

手动排课——拖拽式选择时间段与教室，实时冲突高亮；自动排课——按教学班一键生成指定周节数方案，输出冲突列表及调整建议；排课发布与回滚——管理员确认后批量生效，可回退历史版本。

1. 冲突检测与提示

教室、教师、教学班三维冲突实时校验；提供冲突详情 API（冲突类型、实体 ID、说明），支持前端弹窗展示；支持在排课、选课及课表查询场景下统一调用。

1. 选课服务

课程搜索与多条件筛选（时间段、教师、学分）；时间冲突与容量实时提示，选课提交返回结果与排队序号；支持退课与补选，管理员可设定选课批次与人数上限。

1. 个人课表与通知

教师、学生按周/日视图查看课表，支持导出 iCal 与离线缓存；系统消息中心将排课变动、选课成功、作业布置等推送至移动端，支持一键已读确认。

1. 数据统计与报表

教室利用率、课程满座率、教师工作量等实时图表；选课数据支持导出 CSV/Excel。

## 系统业务流图分析

本项目业务流程图以泳道方式将核心活动按“教务管理员、教师、学生”三条角色纵向展开，横向依时序排列“资源维护→排课→选课→课表发布→统计反馈”五大阶段。图中用实线箭头描述主干流程，例如管理员创建课程与教学班后进入排课子流程；用虚线箭头标注消息或数据同步，如排课发布后系统同时向教师与学生泳道推送变动通知。

手机课程安排业务流图如图1.1所示。

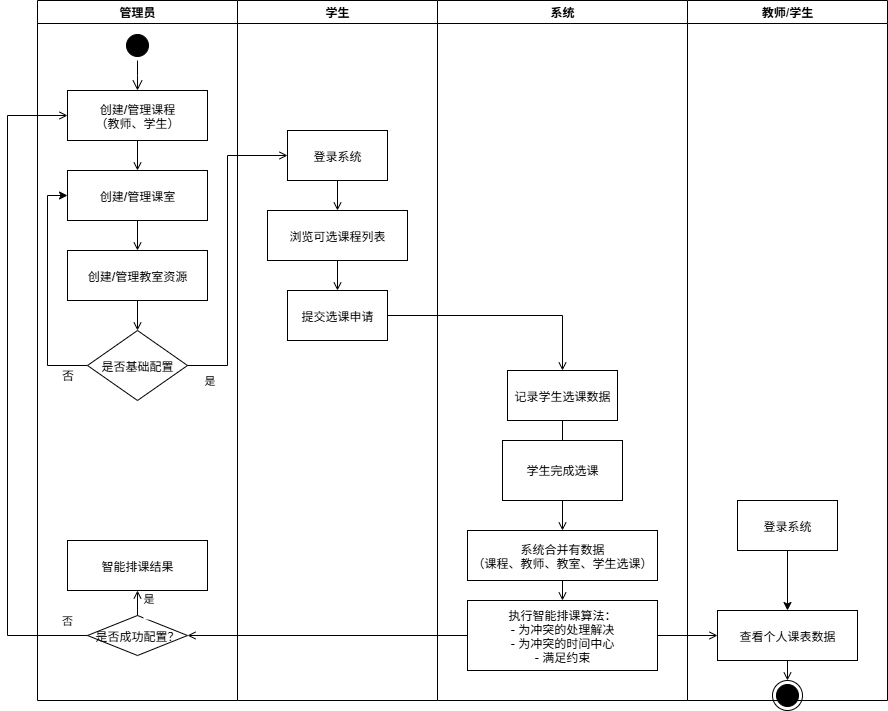


图 1.1 手机课程安排业务流图

## 系统数据流图

系统数据流图分析是对课程安排系统中数据流动和处理的详细描述。数据流图（Data Flow Diagram，DFD）是一种图形化工具，用于描述信息系统中数据的流动、处理和存储过程。对于手机课程安排系统，数据流图可以表示课程管理，用户管理，教学班管理，学生选课等各个环节之间的数据交互和流动情况。

### 系统顶层数据流图

系统顶层数据流图如图1.2所示。



图 1.2系统顶层数据流图

### 系统第一层数据流图

第一层数据流图将排课系统细化为三大角色七类子模块：登录、课程/时间/信息管理、教学班级、消息/通知、统计与校查。学生通过选退课、课表查询及个人服务完成学习规划；教师负责课程时间配置、教学班级及通知发布，保障教学进度；管理员聚焦课程安排、资源统筹与用户管理，支撑全局运行。各模块以身份、课程及通知数据流贯通，确保排课精准、高效与信息一致性。

第一层数据流图如图1.3。

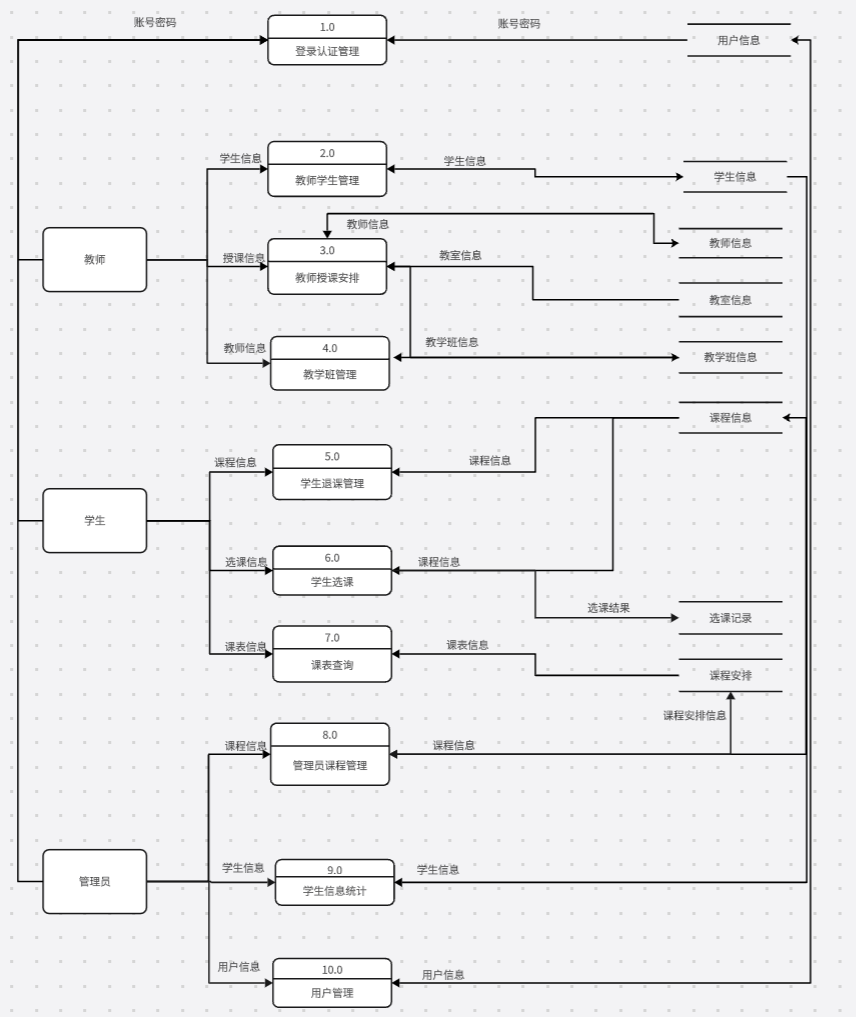


图 1.3第一层数据流图

### 第二层数据流图

**第二层数据流图-教师授课安排**

该图描述了教师如何查看和管理其授课安排，包括获取课程信息和更新教学计划。 教师授课安排数据流图如图1.4所示

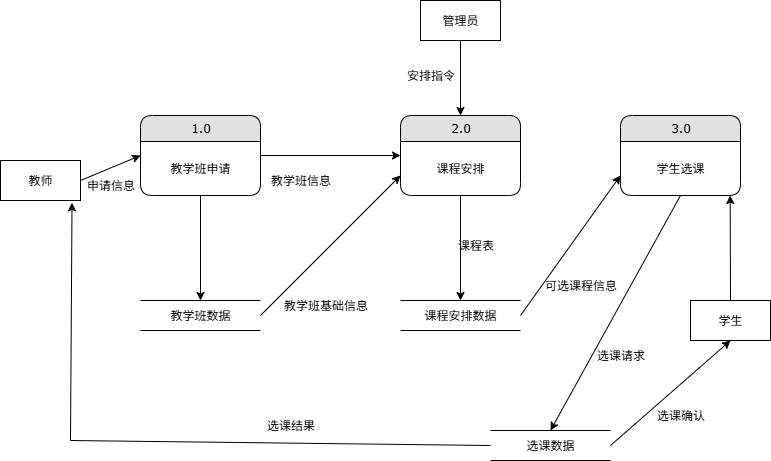


图 1.4教师授课安排数据流图

**第二层数据流图-管理员课程管理**:

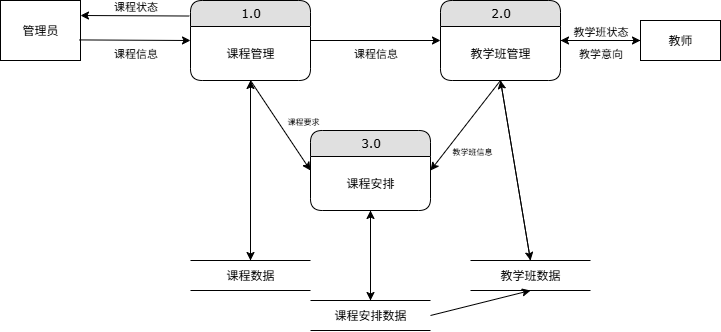
该图展示了管理员对系统中的课程进行全面管理的过程，包括添加、修改、删除和查询课程信息。 管理员课程管理数据流图如图1.5所示 。

图 1.5管理员课程管理数据流图

**第二层数据流图-教师 所教学生管理**:

该图描绘了教师管理其所教授班级内学生名单及相关信息的功能流程。

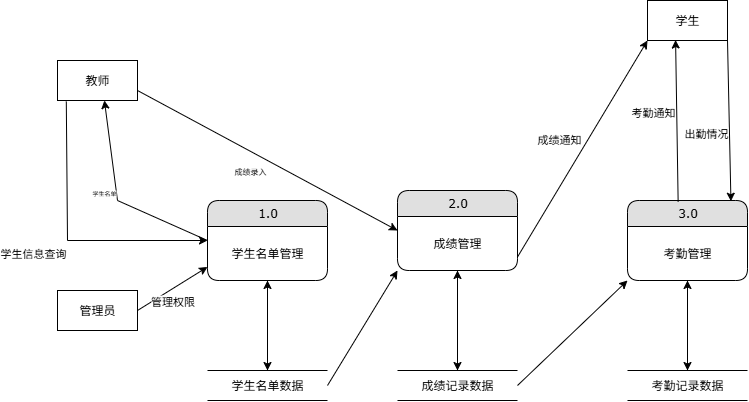
学生管理数据流图如图1.6所示。 

图 1.6学生管理数据流图

**第二层数据流图-登录认证管理**:

该图说明了登录系统时进行身份验证和授权，以访问个人信息和课程相关功能的处理过程。 登录认证数据流图如图1.7所示。

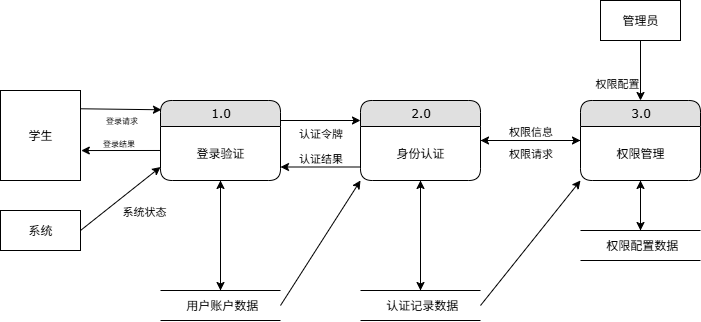


图 1.7登录认证数据流图

**第二层数据流图 - 学生退课管理**:

该图描述了学生在选课后申请退课，并由系统处理退课请求直至更新选课记录的流程。 学生退课数据流图如图1.8所示

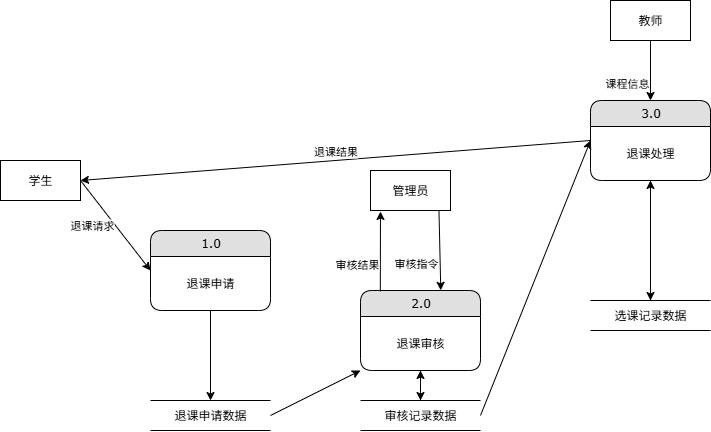


图 1.8学生退课数据流图

**第二层数据流图-管理员对学生信息统计**:

该图展示了管理员如何查询和生成关于学生信息的统计报告，例如学生数量和选课情况。 学生信息统计数据流图如图1.9所示。

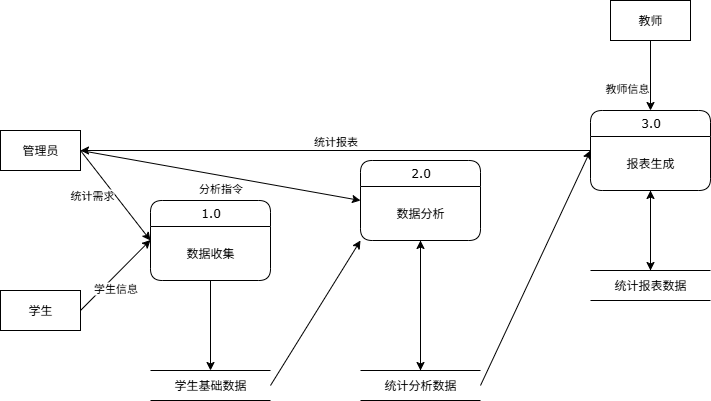


图 1.9学生信息统计数据流图

**第二层数据流图-管理员用户管理**:

该图描绘了管理员对系统中的所有用户账户（包括学生和教师）进行创建、查看、更新和删除等操作的管理流程。

管理员用户管理数据流图如图1.10所示。

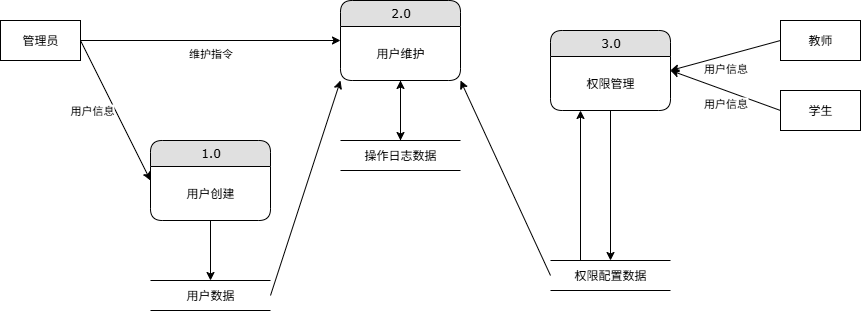


图 1.10管理员用户管理数据流图

**第二层数据流图-教师教学班管理**:

该图说明了教师创建、组织和管理其教学班级，并为班级分配课程和学生的流程。 教学班管理数据流图如图1.11所示。

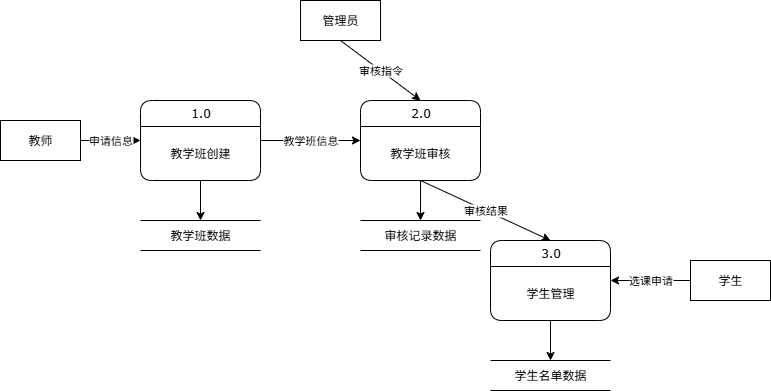


图 1.11教学班管理

**第二层数据流图-学生课表查询**:

该图展示了学生查询个人课表的完整流程，包括从系统获取已选课程信息、课程时间安排，并将这些信息整合成个性化的课程表展示给学生，使学生能够清晰地了解自己的每周课程安排。

学生课表查询数据流图如图1.12所示。

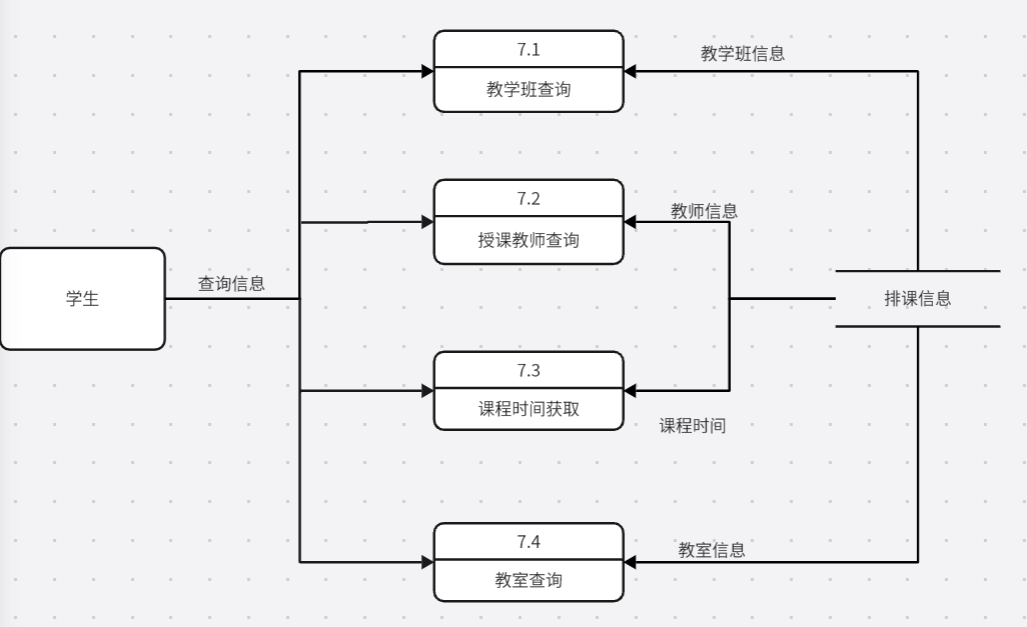


图 1.12学生课表查询数据流图

**第二层数据流图-学生选课**:

该图描述了学生进行选课操作的完整过程，从浏览可选课程列表开始，经过课程容量检查、时间冲突验证等环节，最终完成课程选择并更新选课记录，同时确保选课过程中的各项限制条件得到满足。学生选课&学生课表查询这两个数据流图补充了学生在系统中的核心操作流程，与之前的退课管理形成了完整的选课管理闭环，体现了系统在学生日常学习管理方面的完整功能。

学生选课数据流图如图1.13所示。

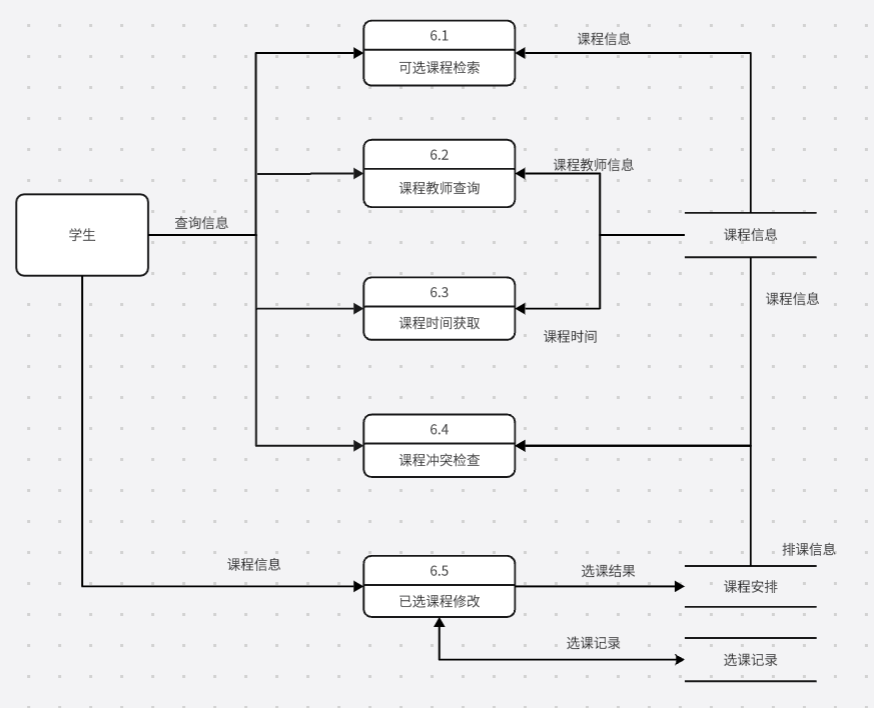


图1.13学生选课数据流图

## 系统数据字典描述

数据流图配以数据字典，就可以从图形和文字两个方面对系统的逻辑模型进行完整的描述。数据字典对数据流图的各种成分起注释、说明的作用，是对数据流图的重要补充。

数据字典中数据项定义如表1.1所示。

表 1.1数据项定义

| 数据项编号 | 数据项名称 | 别名 | 简述 | 类型及宽度 | 取值范围 |
| --- | --- | --- | --- | --- | --- |
| 101 | 用户ID | 用户标识符 | 标识用户的编号 | 整型 | 1~2147483647 |
| 102 | 用户名 | 登录用户名 | 用户登录系统的用户名 | 字符型，255 |  |
| 103 | 密码 | 登录密码 | 用户登录系统的密码 | 字符型，255 |  |
| 104 | 真实姓名 | 用户真实姓名 | 用户的真实姓名 | 字符型，255 |  |
| 105 | 用户角色 | 用户类型 | 用户在系统中的角色 | 字符型，50 | student、teacher、admin |
| 106 | 创建时间 | 用户创建时间 | 用户账户创建的时间 | 日期时间型 |  |
| 107 | 更新时间 | 用户更新时间 | 用户信息最后更新的时间 | 日期时间型 |  |
| 108 | 学生ID | 学生标识符 | 系统中唯一标识学生的编号 | 整型 | 1~2147483647 |
| 109 | 学生用户ID | 学生关联用户 | 学生关联的用户ID | 整型 | 1~2147483647 |
| 110 | 年级 | 学生年级 | 学生所属的年级 | 字符型，50 |  |
| 111 | 班级名称 | 学生班级 | 学生所属的班级名称 | 字符型，100 |  |
| 112 | 教师ID | 教师标识符 | 系统中唯一标识教师的编号 | 整型 | 1~2147483647 |
| 113 | 教师用户ID | 教师关联用户 | 教师关联的用户ID | 整型 | 1~2147483647 |
| 114 | 职称 | 教师职称 | 教师的职称信息 | 字符型，100 |  |
| 115 | 院系 | 所属院系 | 教师所属的院系 | 字符型，100 |  |
| 116 | 课程ID | 课程标识符 | 系统中唯一标识课程的编号 | 整型 | 1~2147483647 |
| 117 | 课程代码 | 课程编号 | 课程的唯一代码 | 字符型，50 |  |
| 118 | 课程名称 | 课程名 | 课程的名称 | 字符型，255 |  |
| 119 | 学分 | 课程学分 | 课程的学分数 | 双精度浮点型 | 0.5~10.0 |
| 120 | 课时 | 课程课时 | 课程的总课时数 | 整型 | 1~500 |
| 121 | 课程描述 | 课程简介 | 课程的详细描述 | 文本型 |  |
| 122 | 课程创建时间 | 课程创建日期 | 课程信息创建的时间 | 日期时间型 |  |
| 123 | 教学班ID | 教学班标识符 | 系统中唯一标识教学班的编号 | 整型 | 1~2147483647 |
| 124 | 教学班代码 | 教学班编号 | 教学班的唯一代码 | 字符型，50 |  |
| 125 | 最大学生数 | 教学班容量 | 教学班可容纳的最大学生数 | 整型 | 1~500 |
| 126 | 当前学生数 | 已选学生数 | 教学班当前已选课的学生数 | 整型 | 0~500 |
| 127 | 教学班课程ID | 关联课程 | 教学班关联的课程ID | 整型 | 1~2147483647 |
| 128 | 教学班教师ID | 任课教师 | 教学班的任课教师ID | 整型 | 1~2147483647 |
| 129 | 课程安排ID | 课程安排标识符 | 系统中唯一标识课程安排的编号 | 整型 | 1~2147483647 |
| 130 | 星期几 | 上课日期 | 课程安排在一周中的哪一天 | 整型 | 1~7 |
| 131 | 开始时间 | 上课开始时间 | 课程的开始时间 | 字符型，10 | HH:MM格式 |
| 132 | 结束时间 | 上课结束时间 | 课程的结束时间 | 字符型，10 | HH:MM格式 |
| 133 | 课程安排教室ID | 上课教室 | 课程安排使用的教室ID | 整型 | 1~2147483647 |
| 134 | 课程安排教学班ID | 关联教学班 | 课程安排关联的教学班ID | 整型 | 1~2147483647 |
| 135 | 选课记录ID | 选课记录标识符 | 系统中唯一标识选课记录的编号 | 整型 | 1~2147483647 |
| 136 | 选课学生ID | 选课学生 | 选课记录关联的学生ID | 整型 | 1~2147483647 |
| 137 | 选课教学班ID | 选择的教学班 | 选课记录关联的教学班ID | 整型 | 1~2147483647 |
| 138 | 选课时间 | 选课操作时间 | 学生进行选课操作的时间 | 日期时间型 |  |
| 139 | 教室ID | 教室标识符 | 系统中唯一标识教室的编号 | 整型 | 1~2147483647 |
| 140 | 建筑名称 | 教学楼名称 | 教室所在的建筑名称 | 字符型，100 |  |
| 141 | 教室名称 | 教室编号 | 教室的名称或编号 | 字符型，50 |  |
| 142 | 教室容量 | 教室可容纳人数 | 教室可容纳的最大人数 | 整型 | 1~1000 |

数据字典中数据流定义如表1.2所示。

表 1.2数据流定义

| 数据流编号 | 数据流名称 | 简述 | 数据流来源 | 数据流去向 | 组成数据项 |
| --- | --- | --- | --- | --- | --- |
| 201 | 用户登录信息 | 用户登录时提交的信息 | 用户界面 | 身份认证管理 | 用户名+密码 |
| 202 | 认证结果 | 用户身份认证的结果 | 身份认证管理 | 用户界面 | 用户角色+认证状态 |
| 203 | 课程信息 | 课程的基本信息 | 管理员 | 课程管理 | 课程代码+课程名称+学分+课时+课程描述 |
| 204 | 教学班信息 | 教学班的基本信息 | 教师 | 教学班管理 | 教学班代码+最大学生数+关联课程 |
| 205 | 选课信息 | 学生选课的信息 | 学生 | 选课管理 | 学生ID+教学班ID+选课时间 |
| 206 | 课程安排信息 | 课程的时间和地点安排 | 管理员/教师 | 课程安排管理 | 星期几+开始时间+结束时间+教室信息 |
| 207 | 用户信息 | 用户的基本信息 | 管理员 | 用户管理 | 用户名+真实姓名+用户角色 |
| 208 | 教室信息 | 教室的基本信息 | 管理员 | 教室管理 | 建筑名称+教室名称+教室容量 |
| 209 | 统计查询请求 | 查询统计信息的请求 | 管理员 | 统计管理 | 查询条件+统计类型 |
| 210 | 统计结果 | 统计查询的结果 | 统计管理 | 管理员 | 统计数据+图表信息 |

数据字典中数据存储定义如表1.3所示。

表1.3数据存储定义

| 数据存储编号 | 数据存储名称 | 简述 | 输入数据流 | 输出数据流 | 组成数据项 |
| --- | --- | --- | --- | --- | --- |
| 301 | 用户信息表 | 存储系统中所有用户的基本信息 | 用户信息 | 用户信息 | 用户ID+用户名+密码+真实姓名+用户角色+创建时间+更新时间 |
| 302 | 学生信息表 | 存储学生的详细信息 | 学生信息 | 学生信息 | 学生ID+学生用户ID+年级+班级名称 |
| 303 | 教师信息表 | 存储教师的详细信息 | 教师信息 | 教师信息 | 教师ID+教师用户ID+职称+院系 |
| 304 | 课程信息表 | 存储课程的基本信息 | 课程信息 | 课程信息 | 课程ID+课程代码+课程名称+学分+课时+课程描述+课程创建时间 |
| 305 | 教学班信息表 | 存储教学班的信息 | 教学班信息 | 教学班信息 | 教学班ID+教学班代码+最大学生数+当前学生数+教学班课程ID+教学班教师ID |
| 306 | 课程安排表 | 存储课程的时间和地点安排 | 课程安排信息 | 课程安排信息 | 课程安排ID+星期几+开始时间+结束时间+课程安排教室ID+课程安排教学班ID |
| 307 | 选课记录表 | 存储学生的选课记录 | 选课信息 | 选课信息 | 选课记录ID+选课学生ID+选课教学班ID+选课时间 |
| 308 | 教室信息表 | 存储教室的基本信息 | 教室信息 | 教室信息 | 教室ID+建筑名称+教室名称+教室容量 |

数据字典中数据处理的定义如图1.4所示。

表 1.4数据处理定义

| 处理编号 | 处理名称 | 简述 | 输入  数据流 | 输出  数据流 | 处理逻辑 |
| --- | --- | --- | --- | --- | --- |
| 401 | 用户身份认证 | 验证用户登录信息 | 用户登录信息 | 认证结果 | 检查用户名和密码是否匹配 |
| 402 | 课程信息管理 | 管理课程的增删改查 | 课程信息 | 课程信息 | 对课程信息进行增加、删除、修改、查询操作 |
| 403 | 教学班管理 | 管理教学班的增删改查 | 教学班信息 | 教学班信息 | 对教学班信息进行增加、删除、修改、查询操作 |
| 404 | 选课处理 | 处理学生的选课请求 | 选课信息 | 选课结果 | 检查选课条件，更新选课记录和教学班人数 |
| 405 | 课程安排管理 | 管理课程的时间和地点安排 | 课程安排信息 | 课程安排信息 | 对课程安排进行增加、删除、修改、查询操作 |
| 406 | 用户信息管理 | 管理用户的基本信息 | 用户信息 | 用户信息 | 对用户信息进行增加、删除、修改、查询操作 |
| 407 | 教室管理 | 管理教室信息 | 教室信息 | 教室信息 | 对教室信息进行增加、删除、修改、查询操作 |
| 408 | 统计分析 | 生成各种统计报表 | 统计查询请求 | 统计结果 | 根据查询条件生成相应的统计报表 |
| 409 | 退课处理 | 处理学生的退课请求 | 退课信息 | 退课结果 | 删除选课记录，更新教学班人数 |
| 410 | 智能排课 | 自动生成课程安排 | 排课参数 | 课程安排信息 | 根据约束条件自动生成合理的课程安排 |

数据字典中外部实体的定义如表1.5所示。

表 1.5外部实体定义

| 外部实体 编号 | 外部实体名称 | 简述 | 向网站输入的数据流 | 网站输出的数据流 |
| --- | --- | --- | --- | --- |
| S01 | 学生 | 使用系统进行选课、查看课程安排的学生用户 | 201,205,209 | 202,206,210 |
| S02 | 教师 | 使用系统管理教学班和课程安排的教师用户 | 201,204,206,209 | 202,204,206,210 |
| S03 | 管理员 | 管理系统用户、课程、教室等信息的管理员用户 | 201,203,206,207,208,209 | 202,203,206,207,208,210 |
| S04 | 系统管理员 | 负责系统维护和高级管理功能的超级管理员 | 201,203,207,208,209 | 202,203,207,208,210 |

数据字典中数据结构的定义如表1.6所示。

表 1.6数据结构定义

| 数据构 编号 | 数据结构名称 | 简述 | 数据结构组成 |
| --- | --- | --- | --- |
| DS01 | 用户信息 | 用户的基本信息包括用户名、密码、角色等 | 101+102+103+104+105+106+107 |
| DS02 | 学生信息 | 学生的详细信息包括年级、班级等 | 108+109+110+111 |
| DS03 | 教师信息 | 教师的详细信息包括职称、院系等 | 112+113+114+115 |
| DS04 | 课程信息 | 课程的基本信息包括名称、学分、课时等 | 116+117+118+119+120+121+122 |
| DS05 | 教学班信息 | 教学班的信息包括代码、容量、关联课程等 | 123+124+125+126+127+128 |
| DS06 | 课程安排信息 | 课程的时间和地点安排信息 | 129+130+131+132+133+134 |
| DS07 | 选课记录信息 | 学生选课的记录信息 | 135+136+137+138 |
| DS08 | 教室信息 | 教室的基本信息包括建筑、名称、容量等 | 139+140+141+142 |

## 系统用例图分析

总体用例图展示学生、教师、管理员等角色围绕选课、排课、信息管理、统计分析等核心用例的交互关系，为系统功能边界与用户需求提供宏观视图。

总体用例图如图1.14所示。

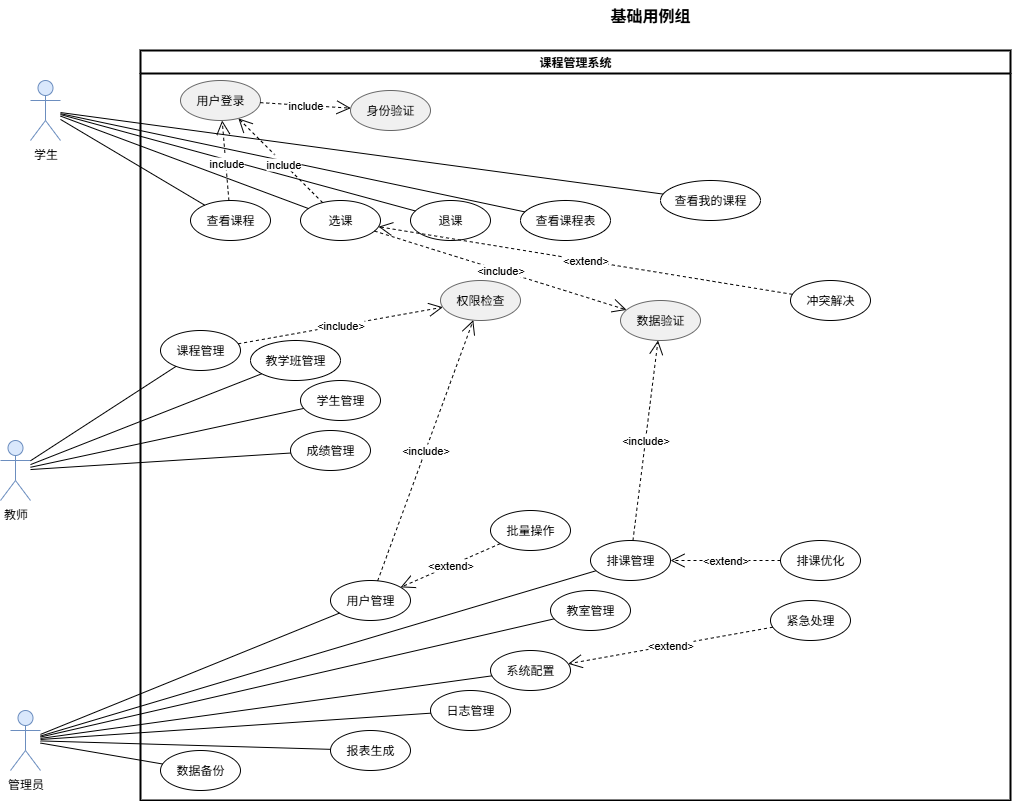


图 1.14总体用例图

管理员用例图展示管理员对系统的核心支撑职能：通过用户管理、课程安排管理、教学班级管理、学生信息统计、排课校查与通知管理等用例，完成资源配置、数据维护与全局监控，体现后台对平台运行的统筹控制。

管理员用例图如图1.15所示。

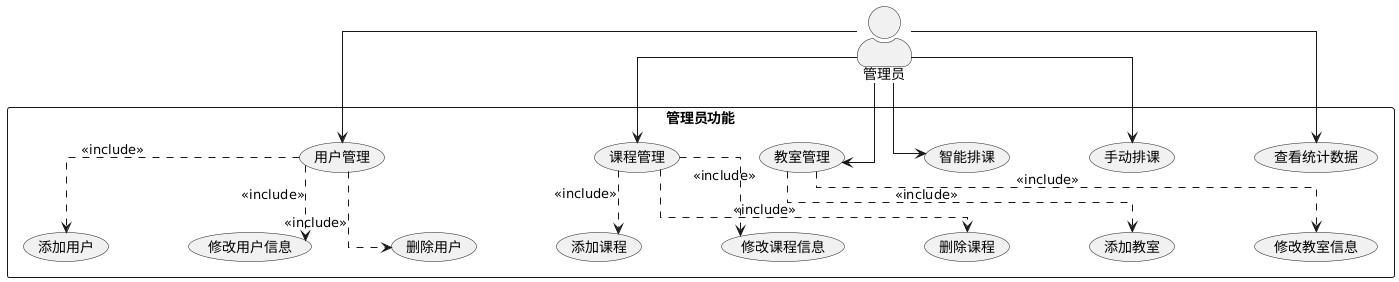


图 1.15管理员用例图

教师用例图刻画教师在教学执行层的职责链：围绕课程时间管理、课程信息管理、教学班级管理、所教学生管理与通知发布等用例开展日常教学，并通过学生选课校查及时掌握学习动态，实现教学过程闭环。教师用例图如图1.16所示。

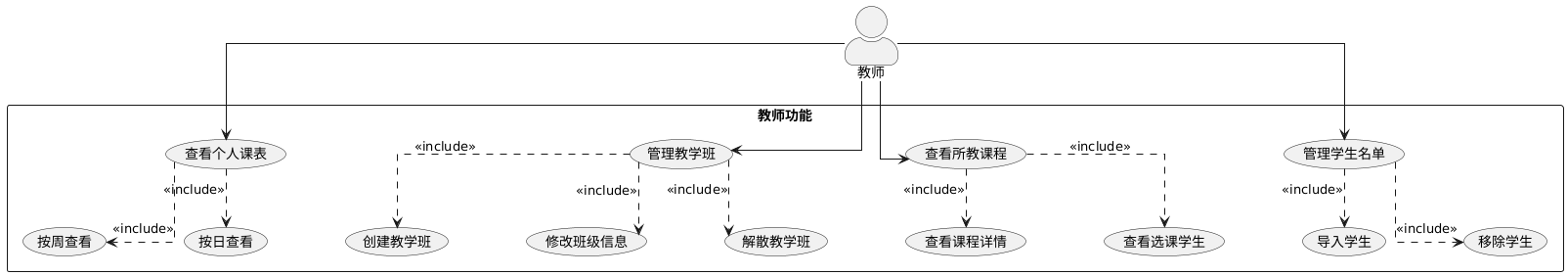


图 1.16教师用例图

学生用例图描绘学生以学习需求为中心的交互流程：通过选课管理、退课管理、课表查询、个人信息管理与个人消息管理等用例完成课程选择与个人事务处理，并借助学生认证管理保障账户安全。学生用例图如图1.17所示。

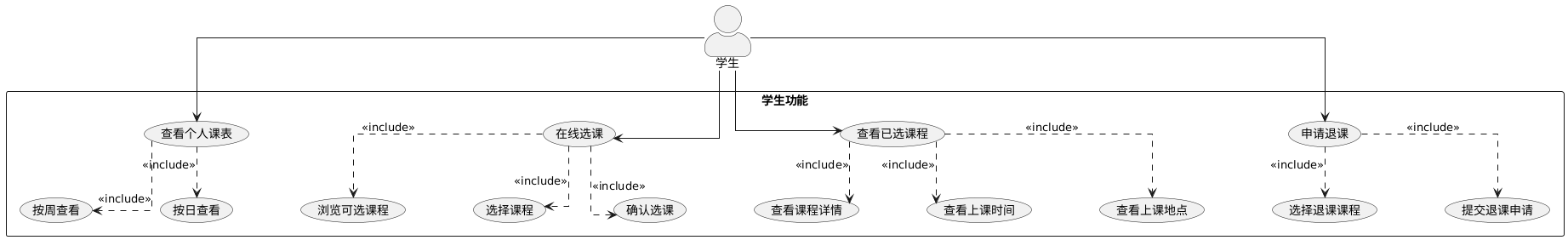


图1.17学生用例图

## 非功能需求分析

1. **高性能**系统应确保快速的响应时间，无论是学生浏览课程信息、进行选课操作，还是教师查看教学班、管理课程安排，以及管理员进行用户管理、课程管理和智能排课，都应尽可能减少等待时间。特别是在选课高峰期，系统需要高效地处理大量并发的选课请求，确保数据的准确性和实时性，避免出现超选、重选等问题，提升用户体验。
2. **安全性**系统应具备完善的身份验证和权限控制机制，只有经过注册验证的学生、教师和管理员账户才能登录系统。不同角色具有不同的功能权限：学生只能进行选课、退课和查看个人课表；教师可以管理自己的教学班和查看所教学生信息；管理员拥有最高权限，可以进行用户管理、课程管理和系统配置。这种分级权限控制有利于保护系统数据安全和正常运行。
3. **易用性**用户界面应简洁明了，易于操作，方便不同技术水平的用户快速找到所需功能。学生能够轻松浏览可选课程、查看课程详情和时间安排；教师能够便捷地管理教学班信息和查看学生选课情况；管理员能够高效地进行各项管理操作。系统还应提供清晰的操作反馈和错误提示，帮助用户正确完成各项操作。
4. **稳定性**系统应长时间稳定运行，减少故障和错误的发生，特别是在选课开放期间和学期关键时间节点。系统还应具备一定的容错能力，当遇到教室冲突、时间冲突或选课人数超限等异常情况时，能够及时检测并提供合理的处理方案，确保系统能够迅速恢复并继续提供服务。
5. **可维护性**采用模块化设计是提高课程调度系统可维护性的关键手段。通过将功能划分为用户管理模块、课程管理模块、选课管理模块、排课管理模块和统计分析模块等独立模块，每个模块都具有清晰的责任边界和接口定义。这样，当需要修改选课规则、增加新的排课算法或优化查询功能时，只需关注相关模块，而无需影响整个系统的稳定性。同时，模块化设计也便于进行单元测试和功能测试，降低系统维护和升级的复杂度。
6. **数据一致性**系统应确保在并发操作下数据的一致性和完整性，特别是在选课过程中，需要准确维护教学班的当前人数、避免超选情况；在排课过程中，需要确保教室和教师时间不冲突；在用户信息变更时，需要保证相关联数据的同步更新，确保整个系统数据的准确性和可靠性。

## 需求分析小结

本章先对高校课程调度管理系统的设计与实现进行系统概述，简要描述了系统流程，其次进行用户需求分析，接着从学生选课管理、教师教学管理、管理员系统管理、智能排课等几个方面进行系统功能需求分析，再依次分析系统业务流图、数据流图和用例图，最后分析非功能需求。

通过需求分析可知，课程调度管理系统主要服务于学生、教师和管理员三类用户群体。学生用户主要需求包括课程浏览、在线选课、退课操作和个人课表查看等功能；教师用户主要需求包括教学班管理、学生信息查看、课程安排查询等功能；管理员用户主要需求包括用户信息管理、课程信息管理、教室资源管理、课程安排管理以及智能排课等系统管理功能。

系统功能需求分析确定了四个核心功能模块：选课管理模块负责处理学生的选课、退课操作和课程冲突检测；教师管理模块负责教学班信息管理和学生信息查询；管理员管理模块负责系统用户、课程、教室等基础数据的维护；智能排课模块负责基于规则和AI算法的自动排课功能。

业务流图、数据流图和用例图的分析明确了系统的业务逻辑、数据流向和用户交互关系，为后续的系统设计和开发奠定了坚实的理论基础。非功能需求分析从高性能、安全性、易用性、稳定性、可维护性和数据一致性等方面提出了系统的质量要求，确保系统能够满足高校课程调度管理的实际需要。

# 系统设计

## 系统功能设计

根据上述系统需求分析，高校课程调度管理系统分为通用模块、管理员后台模块、教师工作台模块和学生中心模块，四大功能模块的设计。系统功能结构图如图4.1所示。其中通用模块为系统基础功能，包含用户登录注册、权限认证和个人信息管理；管理员后台模块、教师工作台模块和学生中心模块为课程调度管理系统的核心业务模块，是本文设计的主要内容。各模块具体功能如下：

**通用模块**包含用户登录注册功能，负责学生、教师、管理员的身份验证；权限认证功能，确保不同角色用户访问对应权限的功能；个人信息修改功能，允许用户维护和更新个人基本信息。

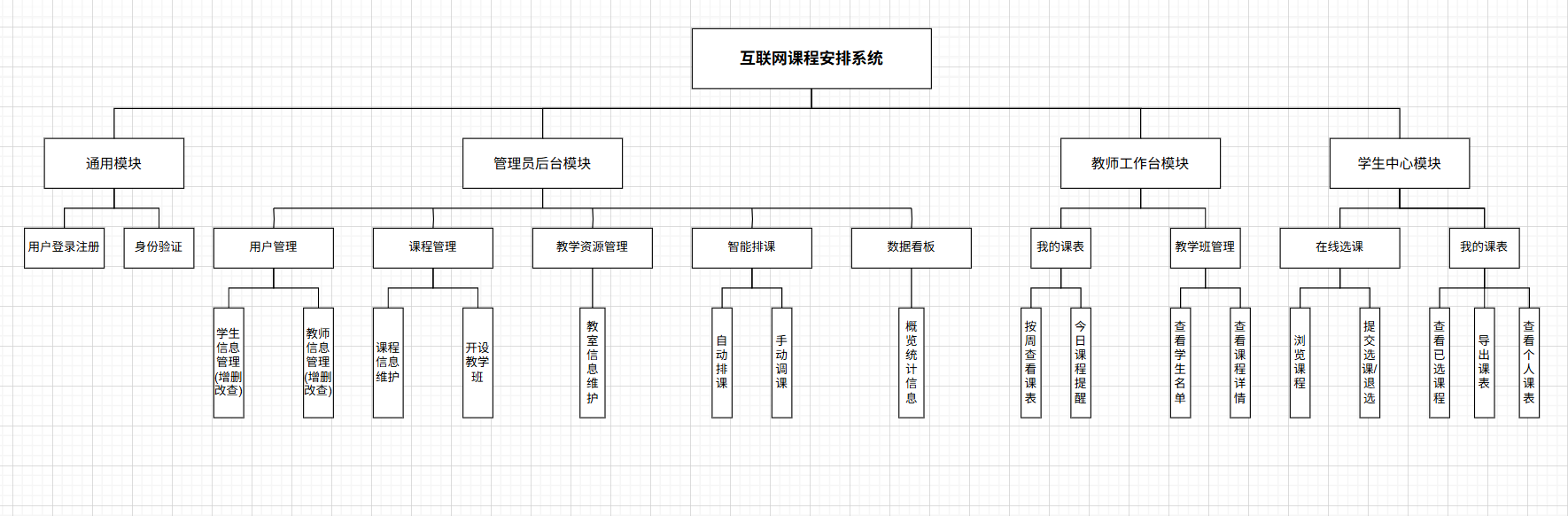
**管理员后台模块**包含用户管理功能，负责学生信息和教师信息的增删改查操作；课程管理功能，负责课程信息维护和教学班开设；教学资源管理功能，负责教室信息的维护管理；智能排课功能，提供自动排课、手动调整和课表发布功能；数据看板功能，提供系统运行的概览统计信息。

**教师工作台模块**包含我的课表功能，支持按周查看课表和今日课程提醒；教学班管理功能，允许教师查看所授课程的学生名单和课程详细信息。

**学生中心模块**包含在线选课功能，支持学生浏览全校课程、提交选课退选申请和查看已选课程状态；我的课表功能，提供个人课表查看和课表导出服务。

功能结构图将系统划分为通用模块、管理员后台、教师工作台、学生中心四层：通用负责登录认证与信息维护；后台涵盖用户、课程与资源管理、智能排课及数据统计；教师侧支持课表查看与教学班管理；学生侧实现在线选课、课表查询与导出。树状层级清晰呈现各功能及增删改查关系，体现系统从排课生成到教学服务的闭环。

功能结构图如图2.1所示。



图表 2.1 功能结构图

类图外框说明：

本章功能设计的UML类图，外框代表不同的架构层次或模块分组：

1. Controller层 - 包含AuthController，负责处理HTTP请求和响应
2. Service层 - 包含AuthService，负责业务逻辑处理
3. Repository层 - 包含UserRepository，负责数据访问操作
4. Entity层 - 包含User、Student、Teacher等实体类，代表数据模型
5. DTO层 - 包含AuthResponseDTO、LoginDTO等数据传输对象
6. Config层 - 包含SecurityConfig等配置类

这种分层架构设计遵循了经典的MVC模式和分层架构原则：

* 表现层(Controller)：处理用户请求
* 业务层(Service)：处理业务逻辑
* 数据访问层(Repository)：管理数据操作
* 实体层(Entity)：定义数据模型
* 配置层(Config)：系统配置

### 用户管理模块

#### 用户登录

用户登录功能允许已注册用户通过用户名和密码进入系统。系统将验证用户身份，并根据用户角色（管理员、教师、学生）跳转到相应的工作页面。

用户登录功能类图如图2.2所示。

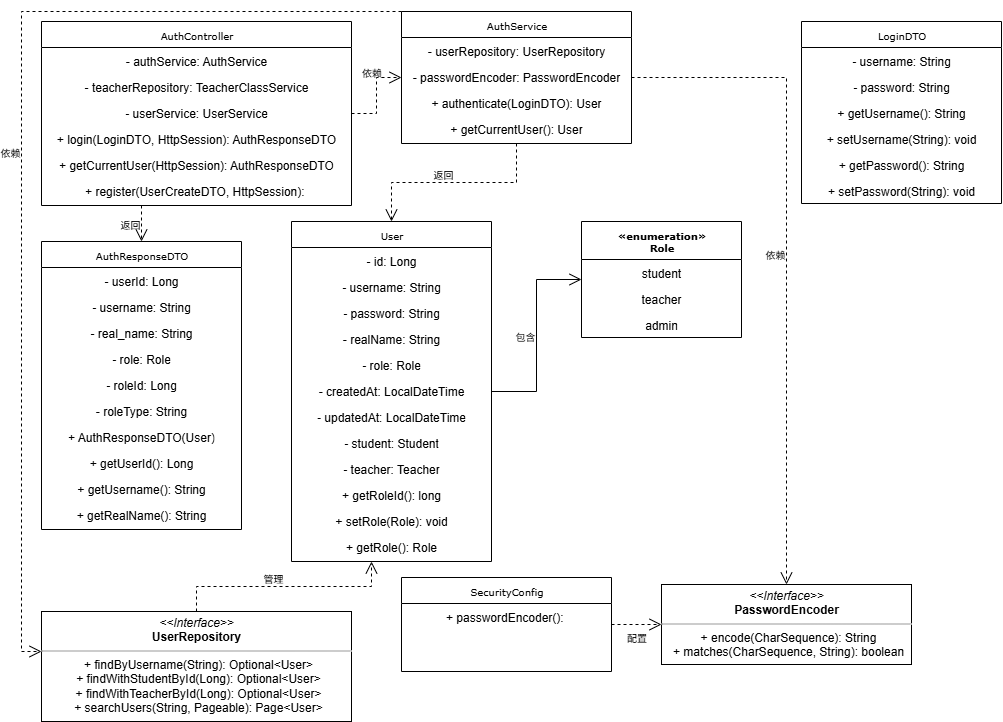
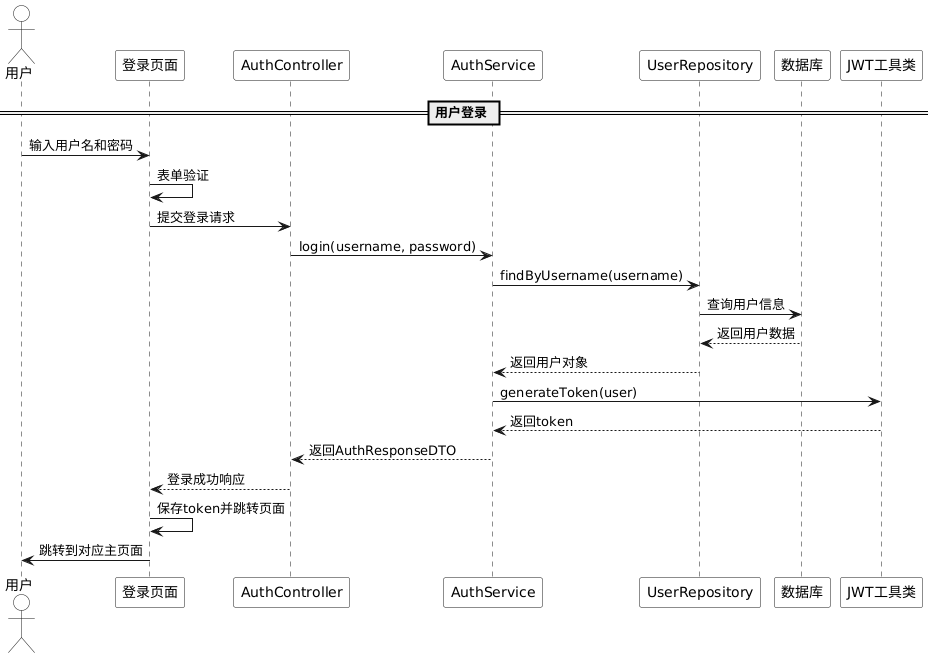


图2.2 用户登录功能类图



图表 2.3 用户登录时序图

#### 用户注册

用户注册功能允许新用户在系统中创建账户。用户需要填写基本信息和选择角色（学生或教师），系统会进行信息验证和重复性检查，成功注册后发送激活邮件进行账户验证。

用户注册时序图如图2.4所示。

用户注册功能类图如图2.5所示。

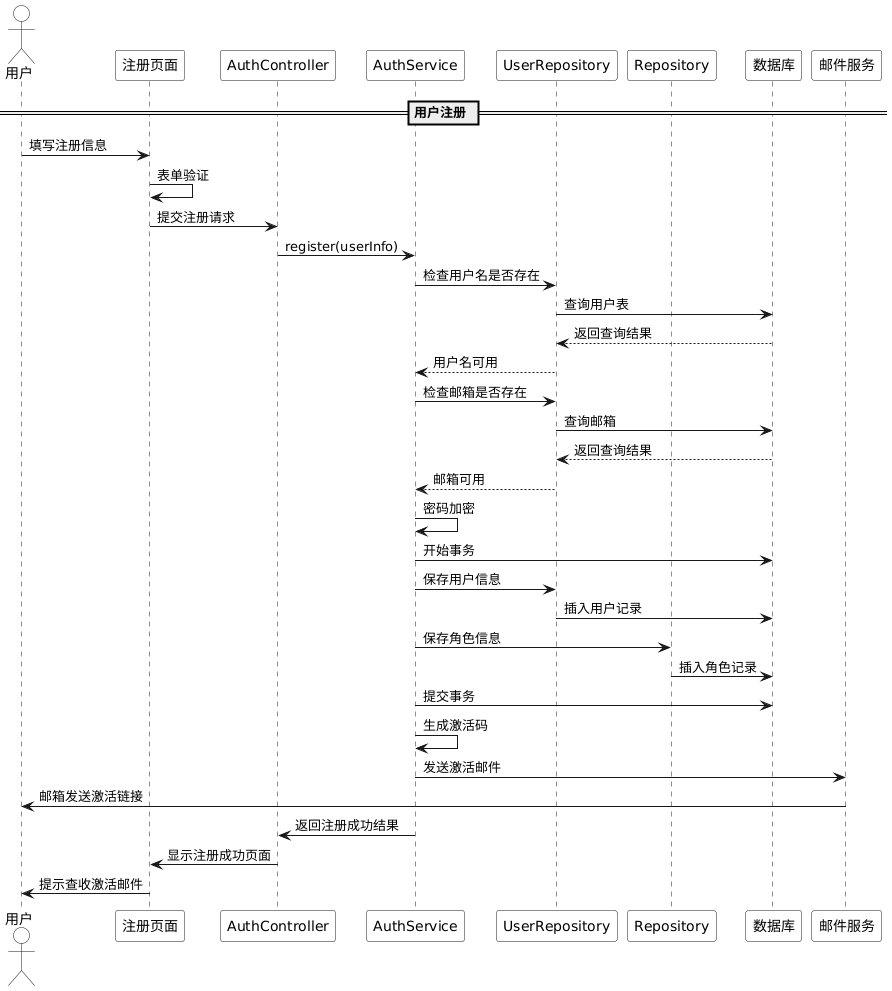


图 2.4用户注册时序图

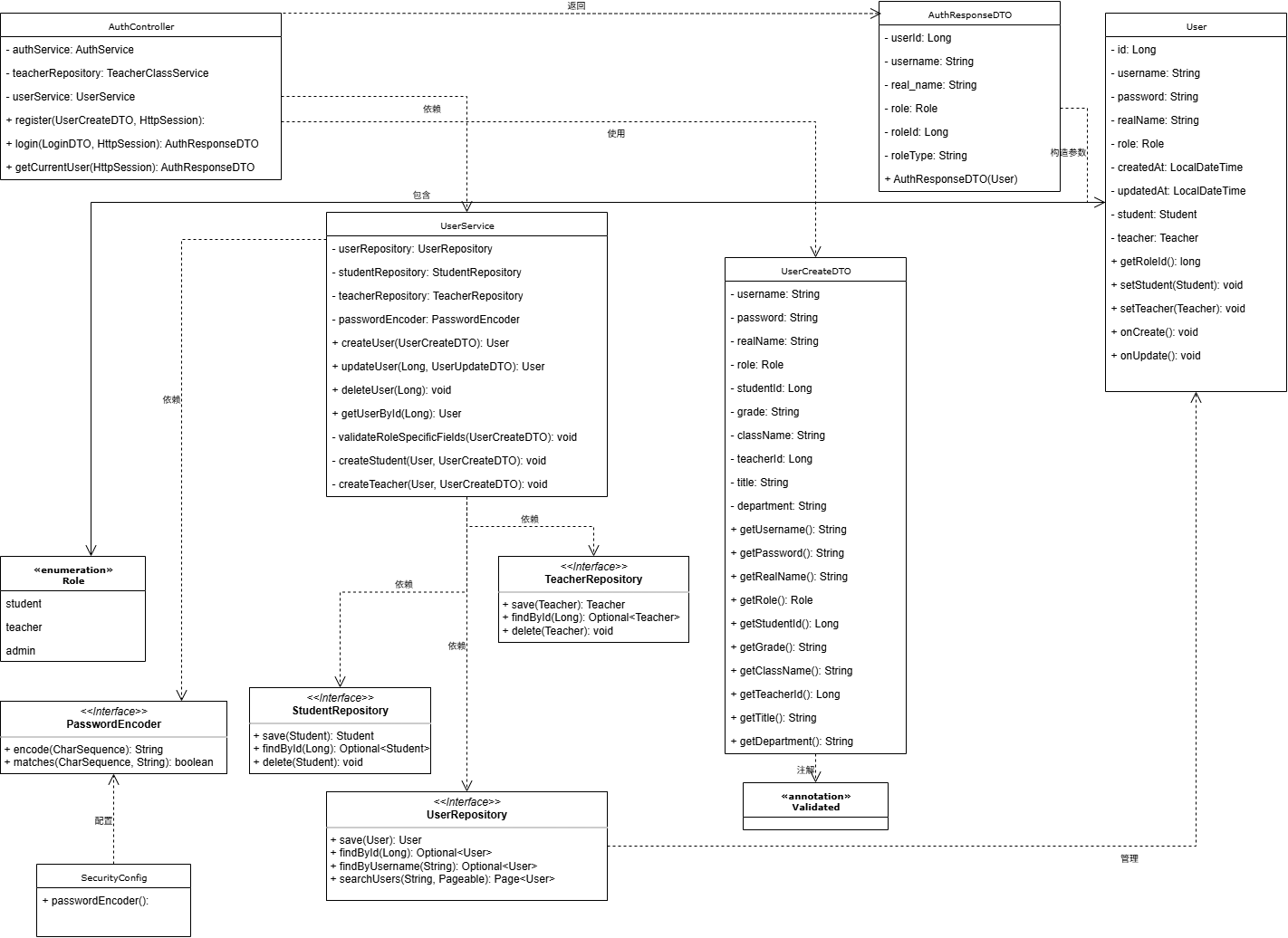


图 2.5用户注册功能类图

#### 权限认证

权限认证功能负责验证用户身份和访问权限，确保只有合法用户才能访问系统资源。通过JWT令牌机制实现无状态认证，支持细粒度的权限控制。

权限认证时序图如图2.6所示。

权限认证功能类图如图2.7所示。

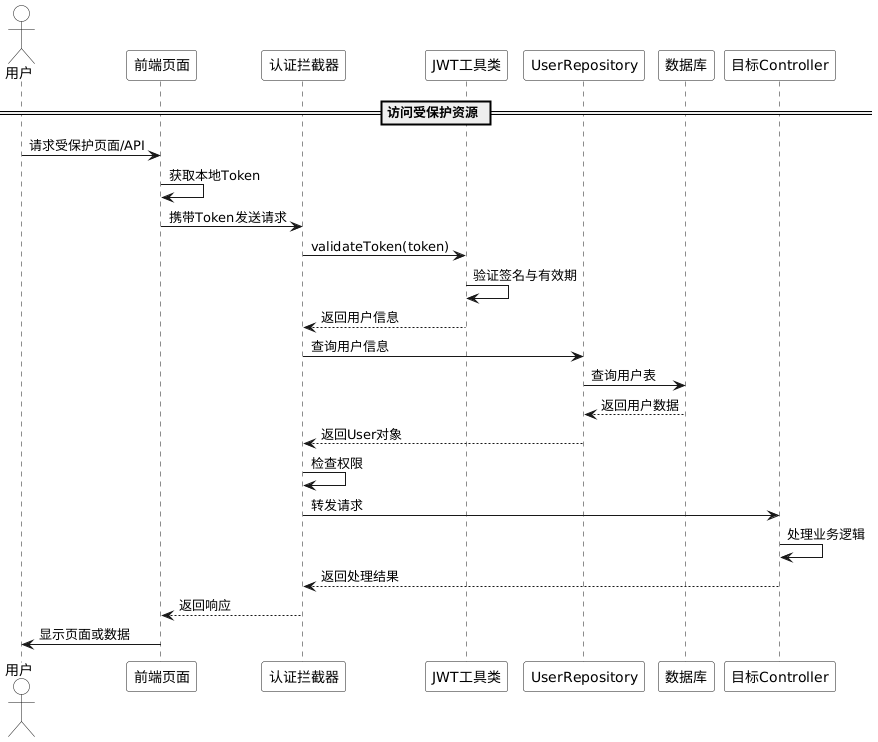


图 2.6权限认证时序图

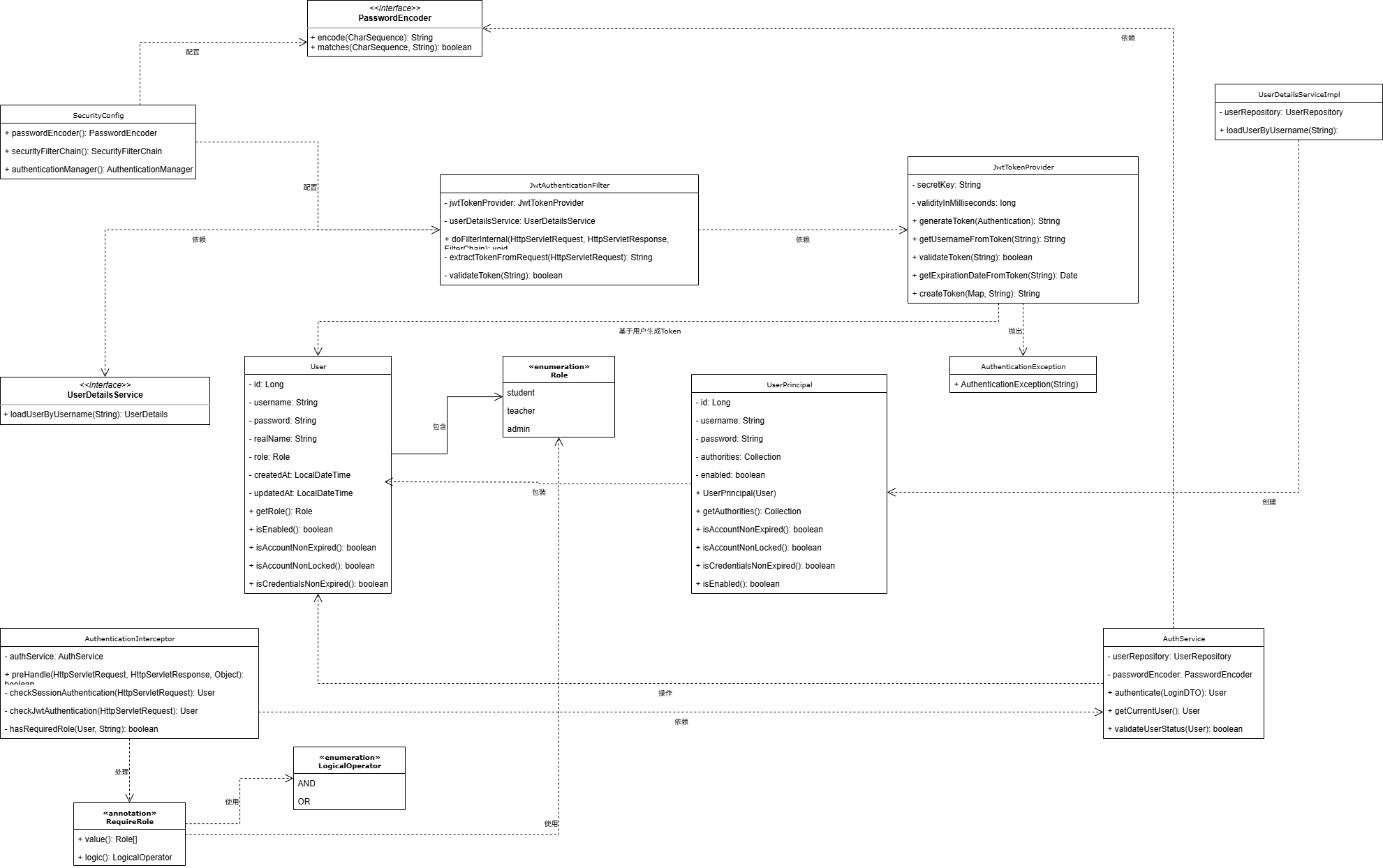


图 2.7权限认证功能类图

#### 个人信息修改

个人信息修改功能允许已登录用户查看和更新自己的个人资料。系统支持修改基本信息（如邮箱、手机号、真实姓名）和角色特定信息（如学生的专业班级、教师的部门职称等）。

个人信息修改时序图如图2.8所示。

个人信息修改功能类图如图2.9所示。

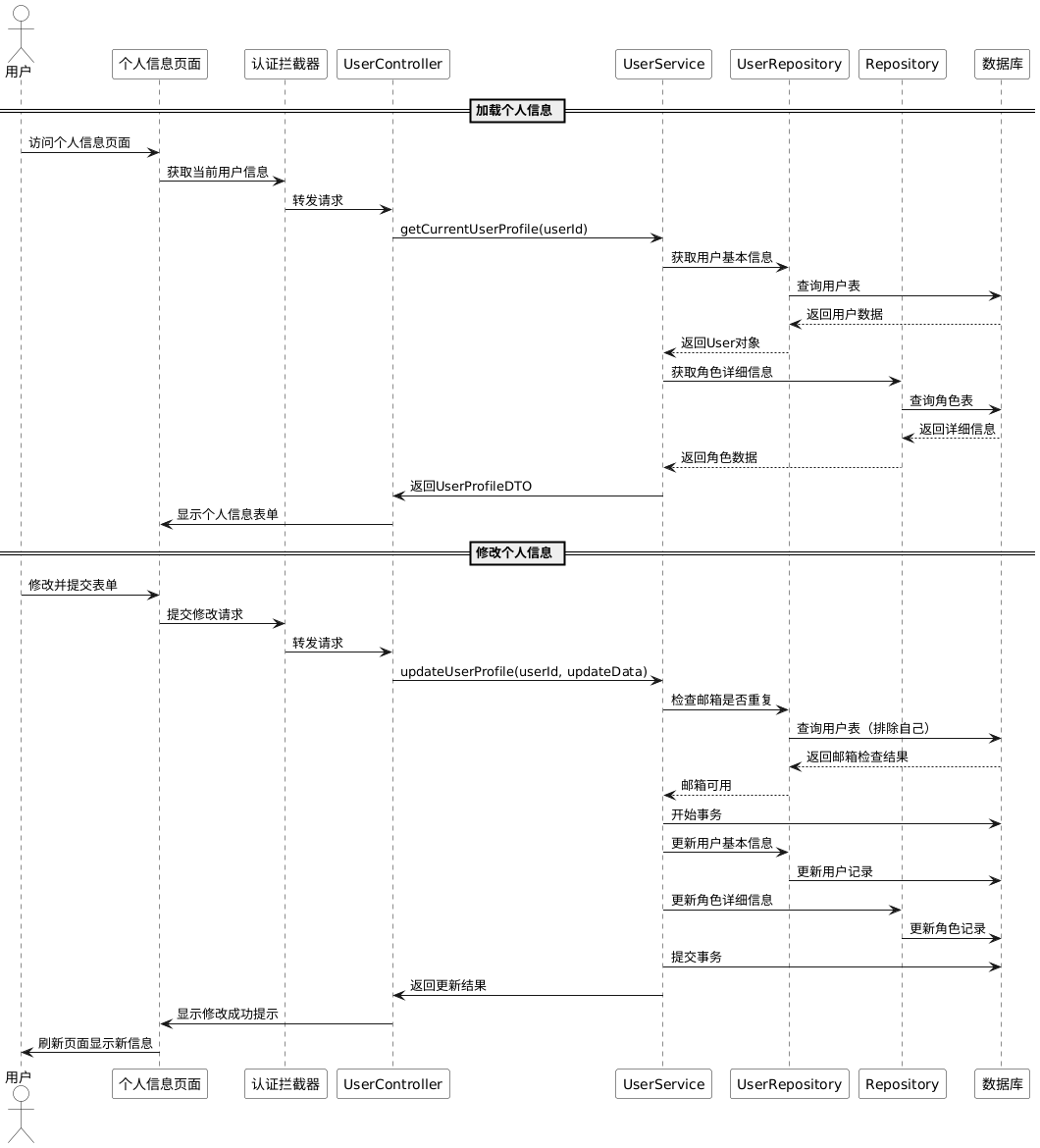


图 2.8个人信息修改时序图



图 2.9个人信息修改功能类图

### 学生中心

#### 学生选课：

系统在学生身份校验通过后，提供可选课程与对应教学班信息列表；学生发起选课请求时，后台会确认学生与教学班均存在、并校验教学班余量，若人数未满则写入选课记录并自动更新已选人数，失败场景会返回容量不足或记录不存在等提示。退课流程对选课记录进行删除并回滚人数，确保数据一致性与实时课余量展示。

学生选课时序图如图2.10所示。

选课与退课类图如图2.11所示。

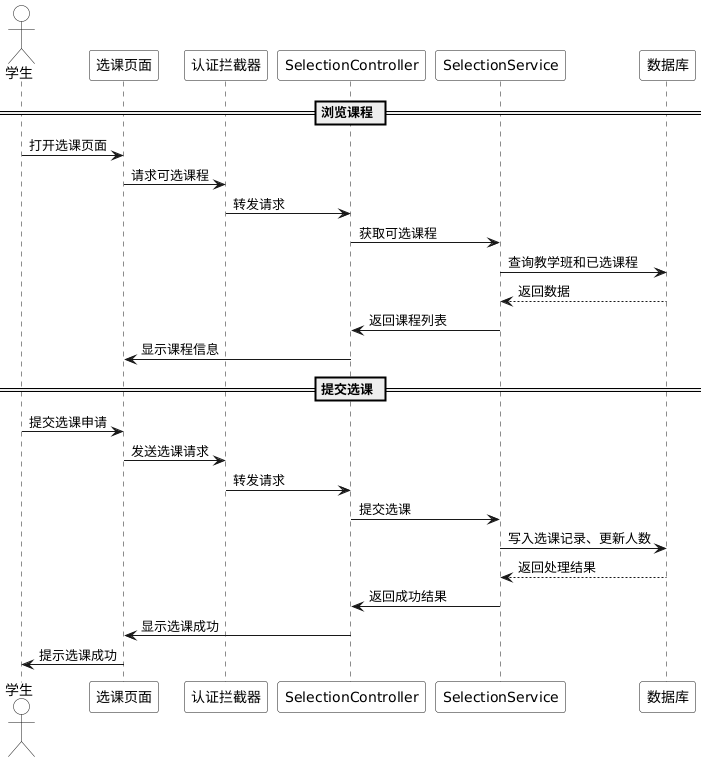


图 2.10学生选课时序图

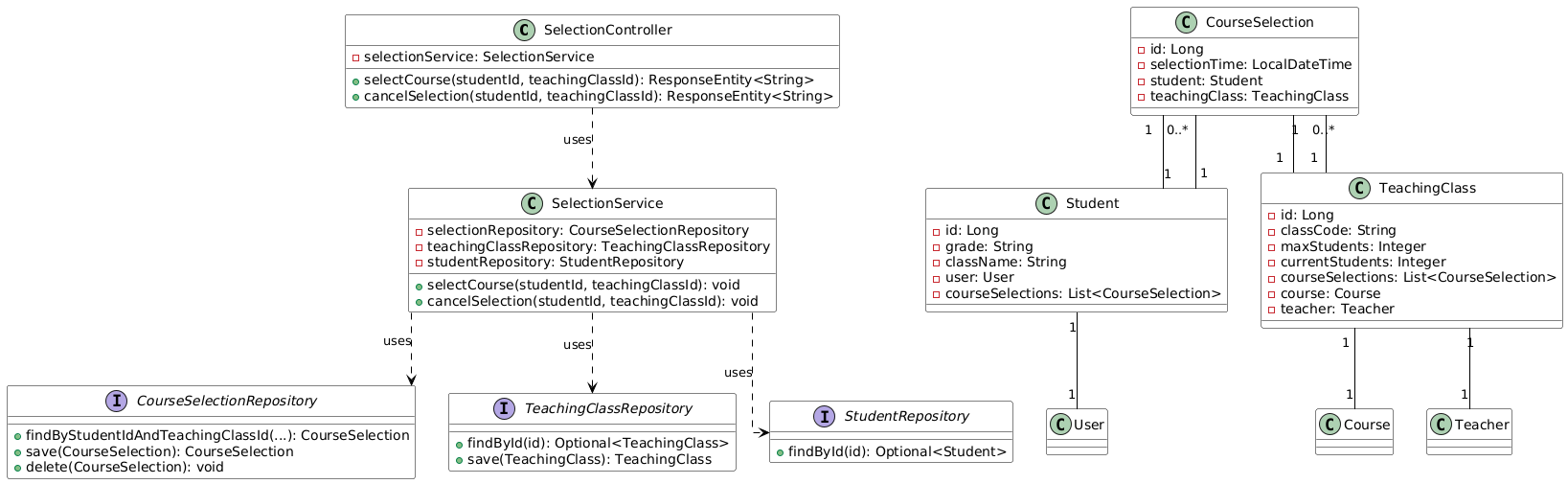


图 2.11选课与退课类图

#### 学生课表查看

学生选课功能基于身份认证入口，首先向学生展示按学院、课程属性分类的实时课表；系统内置容量限制、时间冲突与先修校验算法，点击即完成选课并生成个人课表；支持一键退课、实时更新余量，并通过消息中心推送选退结果与提醒，确保选课流程便捷、安全且可追溯。

学生查看课表时序图如图2.12所示。

查看个人课表类图如图2.13所示

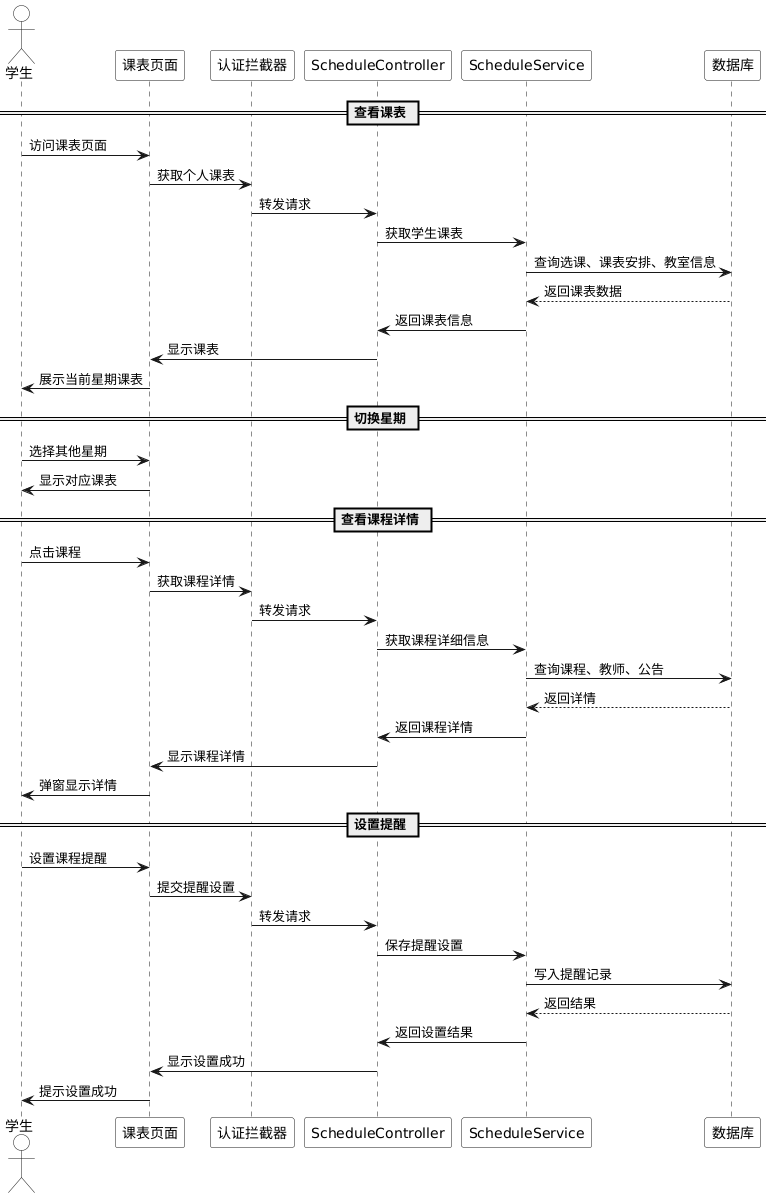


图 2.12查看个人课表时序图

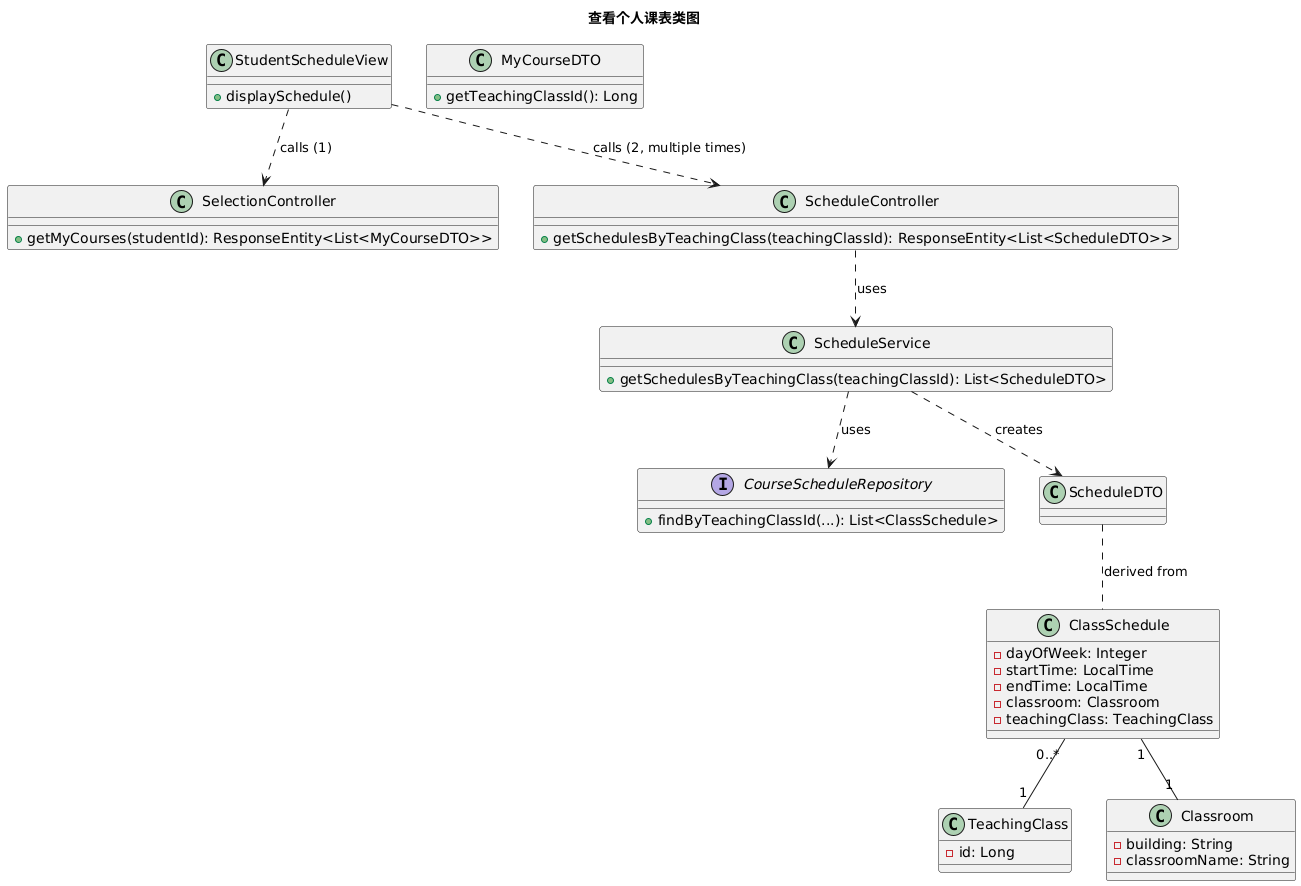


图 2.13查看个人课表类图

#### 学生查看我的课程

学生登录后即可获取个人已选课程清单，系统一次性汇总课程名称、授课教师、学分等关键信息并返回；同时可按需查询对应教学班的详细排课时间，实现课程列表与时间表的灵活切换，帮助学生全面掌握个人上课安排。

学生查看我的课程时序图如图2.14所示。

学生查看我的课程类图如图2.15所示。

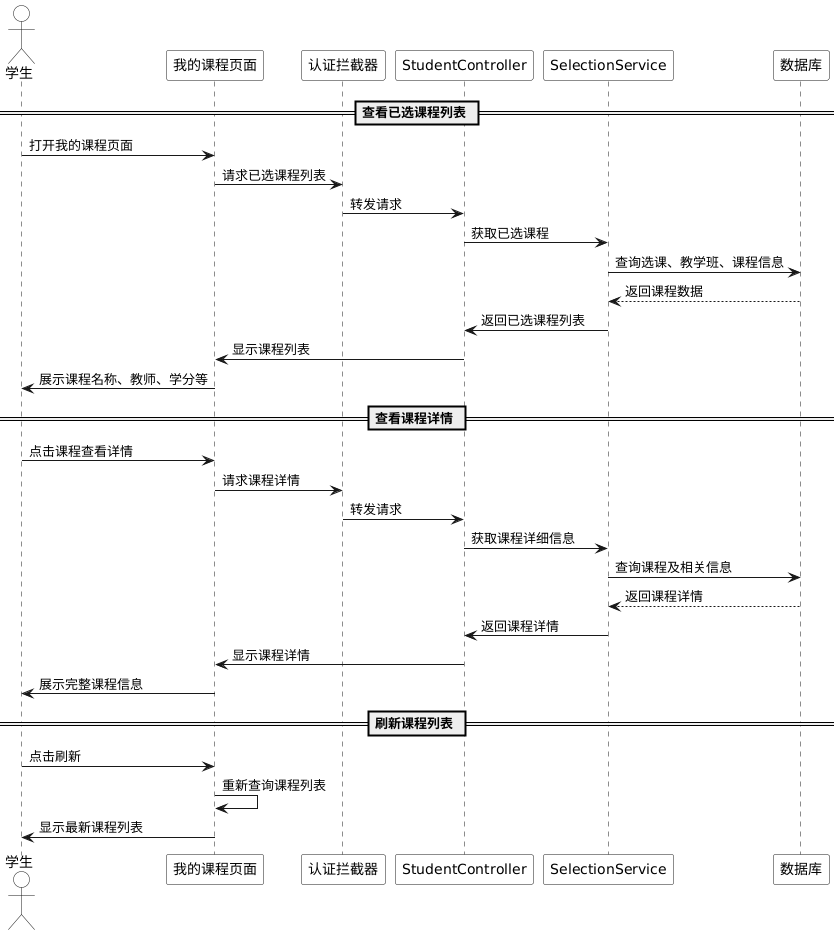


图 2.14查看我的课程时序图

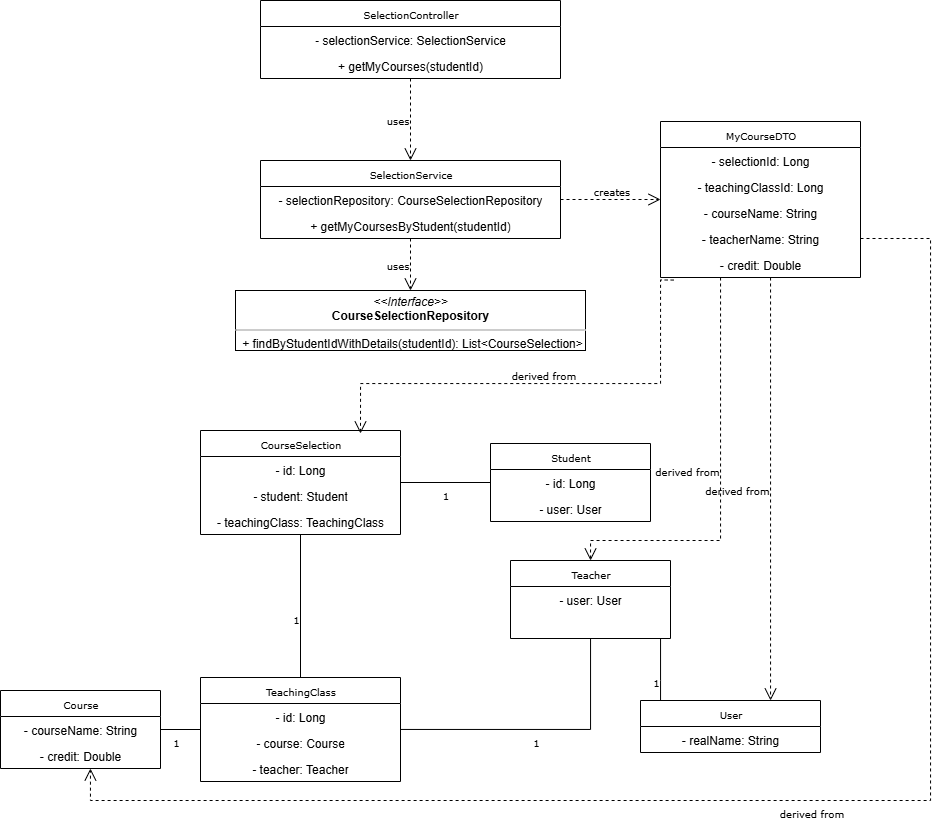


图 2.15查看我的课程类图

### 管理员后台

#### 学生信息管理（教师信息管理同理）

学生信息管理功能允许系统管理员对学生用户进行全生命周期管理，包括查看学生列表、添加新学生、修改学生信息、删除学生账户等操作。支持分页查询、条件搜索和批量操作。

学生信息管理时序图如图2.16所示。

学生信息管理类图如图2.17所示。



图 2.16学生信息管理时序图

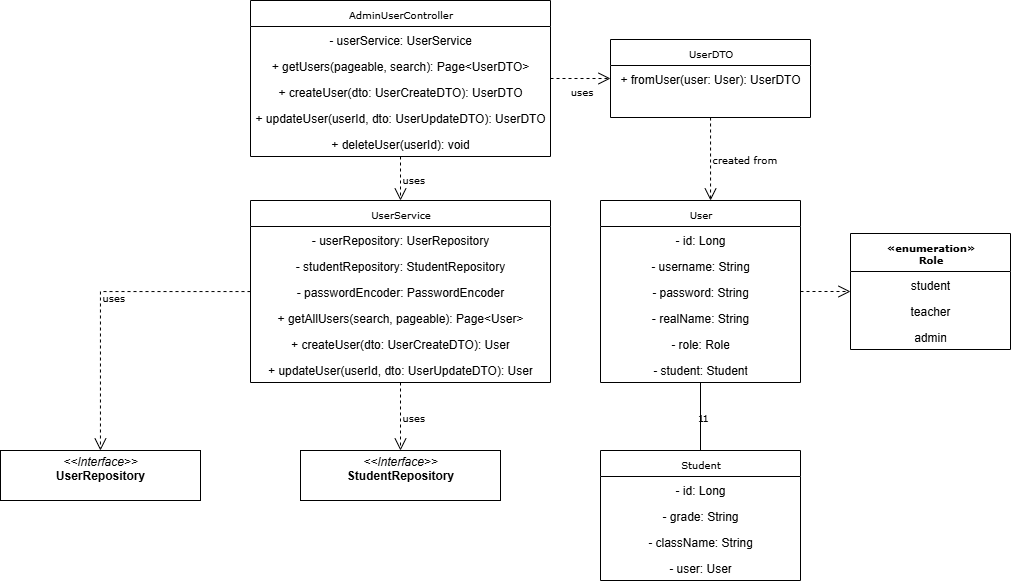


图 2.17学生信息管理类图

### 教师工作台

#### 教学班管理

功能描述：教师可以创建、编辑、删除个人的教学班，管理教学班的基本信息和学生容量。主要特性：

- 教学班信息管理：代码、名称、容量、当前人数

- 关联课程选择，支持从个人课程中选择

- 学生人数实时统计和容量控制

- 教学班代码唯一性验证

- 批量操作和搜索功能

- 数据完整性检查，防止误删有学生选课的教学班

教师教学班管理时序图如图2.18所示。

教师教学班管理类图如图2.19所示。

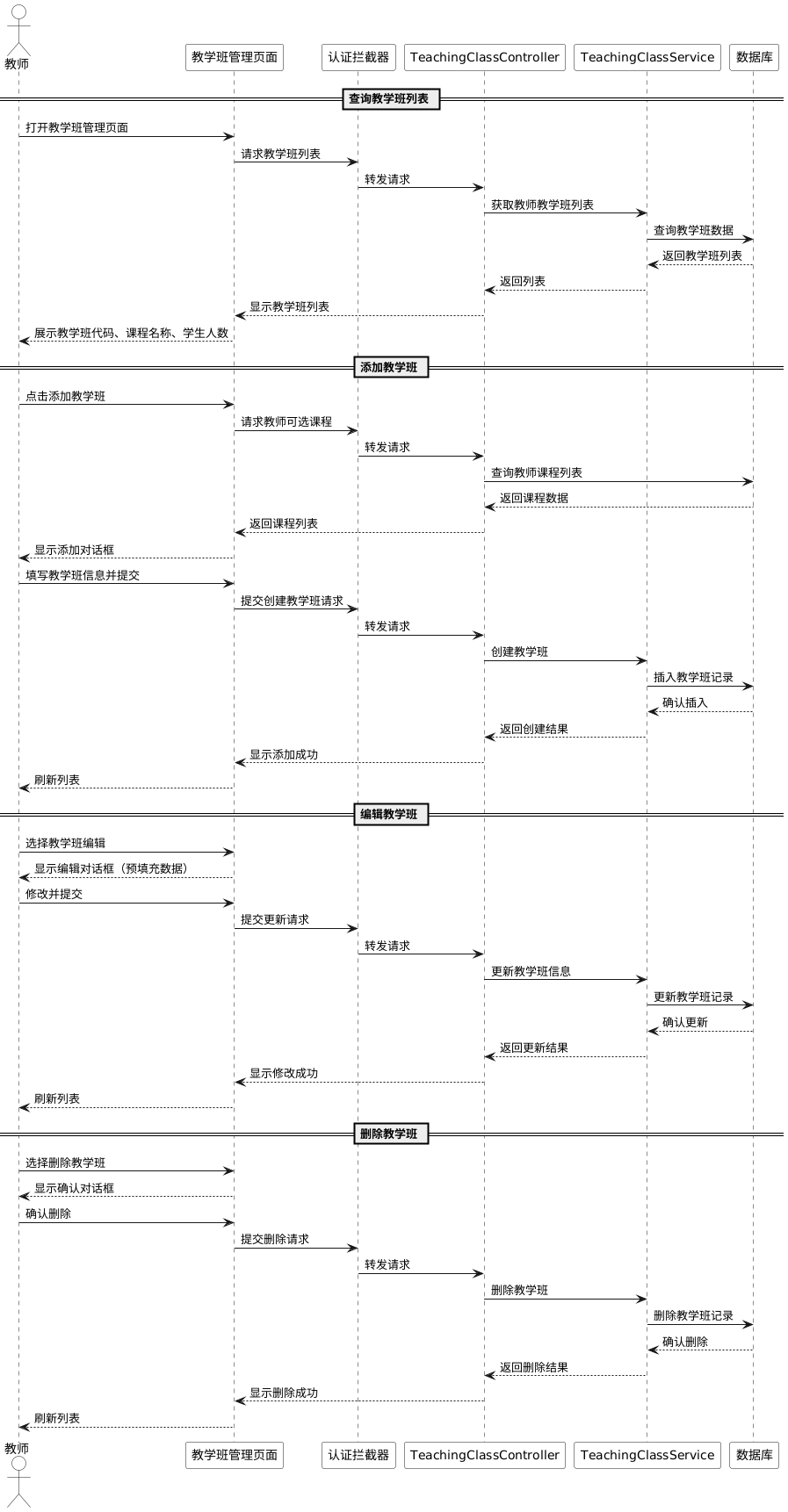


图 2.18教师教学班管理时序图

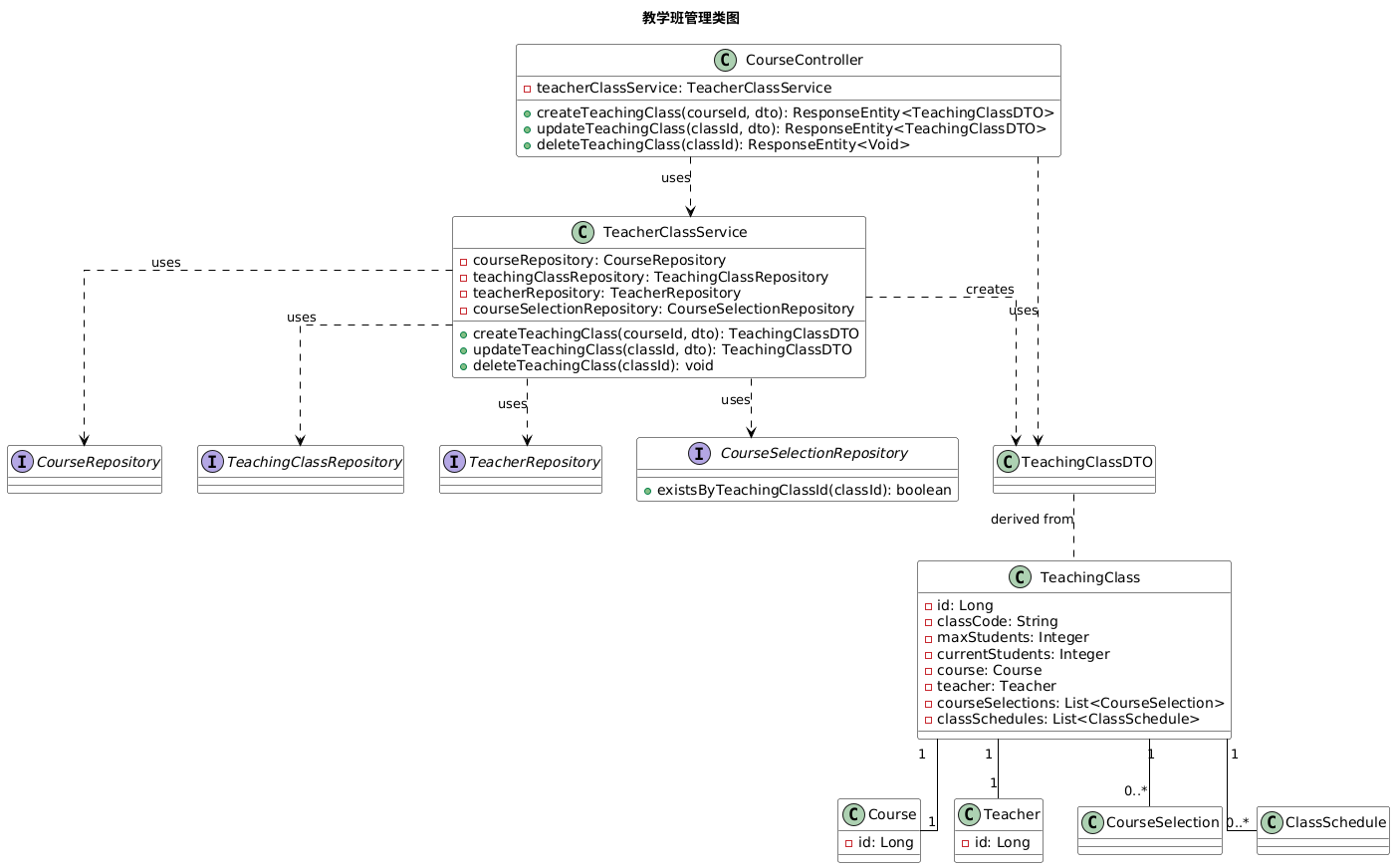


图 2.19教学班管理类图

#### 教师管理课程

功能描述：教师可以管理个人负责的课程信息，包括课程创建、编辑、删除等操作。主要特性：

- 课程基本信息管理：代码、名称、学分、课时、描述

- 课程代码唯一性验证

- 关联性检查，避免删除有教学班的课程

- 搜索和筛选功能

- 分页显示和批量操作

- 课程统计和分析功能

课程管理时序图如图2.20所示。

课程管理类图如图2.21所示。

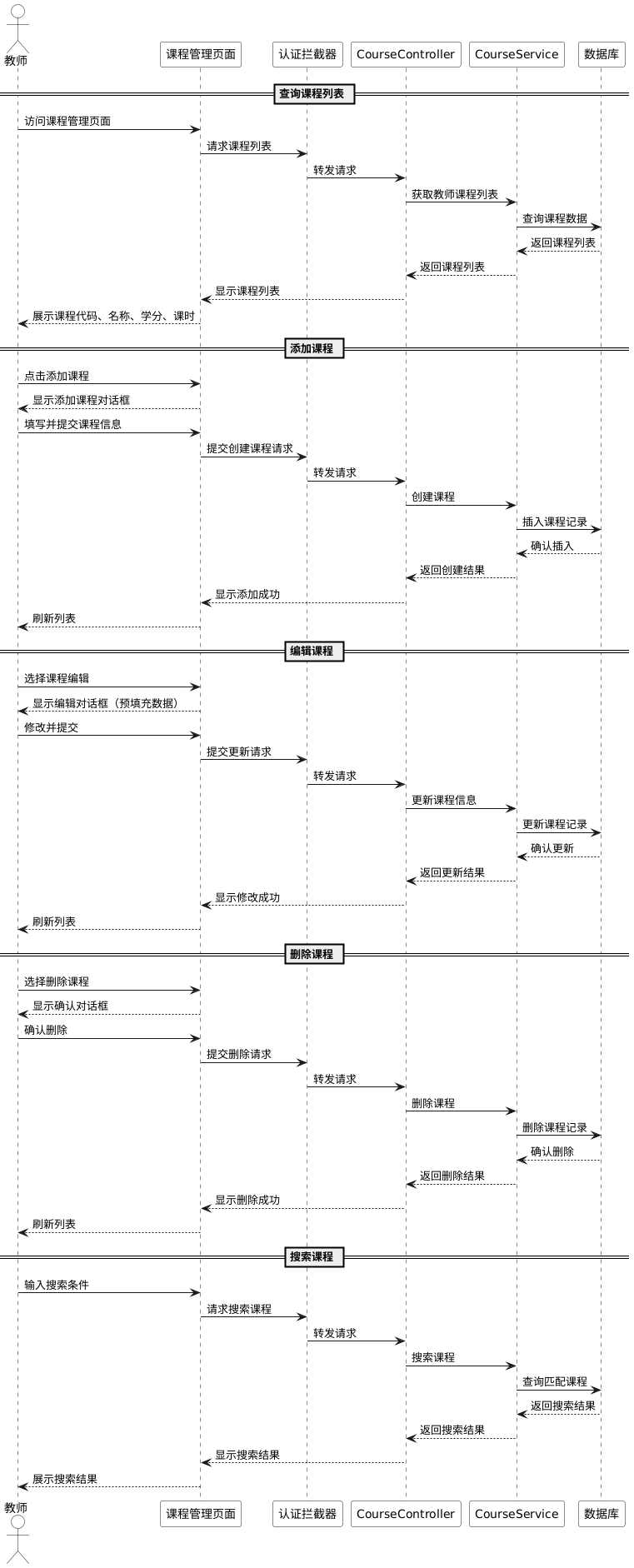


图 2.20课程管理时序图

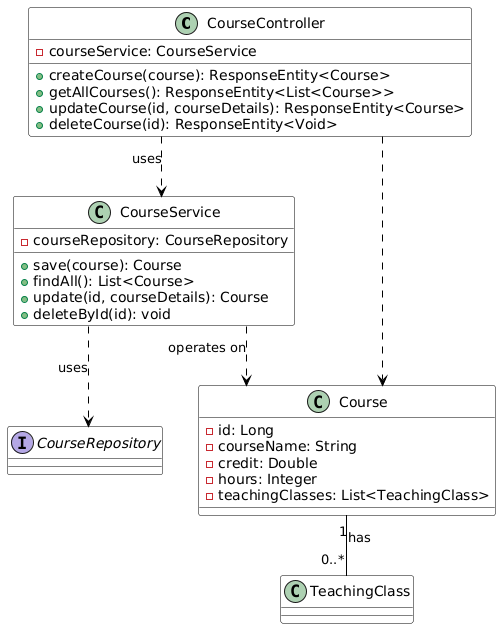


图 2.21课程管理类图

## 输入输出设计

本项目采取前端 vue + 后端 REST API 架构，输入数据以表单或文件方式经 HTTPS 传输为 JSON，对象字段统一使用 UTF-8 编码；输出数据一律返回 JSON，并在移动端或 Web 端以 View 组件渲染，同时提供 PDF／Excel 等可下载文件流以满足打印需求。下述分功能描述输入内容、格式与载体，以及对应输出与呈现方式。

### 用户层输入输出

1. 用户注册／登录

输入：用户名、密码（必填），角色下拉（学生／教师），JSON 格式随键盘输入提交；

输出：登录成功提示、会话令牌与用户基础信息 JSON；前端弹窗+页面跳转反馈。

1. 密码重置

输入：手机号或邮箱验证码＋新旧密码，键盘输入；

输出：重置结果 JSON 与 Toast 提示。

1. 个人信息修改

输入：头像文件（JPEG/PNG）、姓名、联系方式，文件上传＋表单键盘输入； 输出：修改后用户资料 JSON，界面实时刷新。

### 学生端输入输出

1. 选课

输入：教学班 ID（下拉点选）、确认按钮；

输出：选课成功／失败信息 JSON，界面余量数字实时变化。

1. 退课

输入：选课记录 ID，长按或按钮触发；

输出：退课结果 JSON 与 Toast，课程列表刷新。

1. 查看课表

输入：周次选择器（数字）、视图切换选项；

输出：课程时间表 JSON，经日历或列表组件显示，可导出 PDF／ICS 文件供打印或导入系统日历。

### 教师端输入输出

1. 教学班管理

输入：班号、容量、课程及教室选择，下拉＋键盘输入；

输出：教学班信息 JSON，表格组件展示，可导出 Excel。

1. 排课维护

输入：周几、起止时间、教室选择器，下拉＋时间选择器；

输出：排课冲突校验结果与成功信息 JSON，甘特图或表格视图展示。

### 管理员后台输入输出

1. 用户管理

输入：搜索关键词、分页参数，键盘输入；

输出：用户列表 JSON，表格分页展示，支持 CSV 导出。

1. 课程／教室管理

输入：课程代码、学分、教室容量等表单项；

输出：操作结果 JSON 与最新列表视图。

1. 统计报表

输入：日期范围选择器；

输出：用户、课程、教室数量等统计 JSON，经 ECharts 图表展示，并可一键导出 PDF 报告。

## 数据库设计

### 数据库概念模型设计

本系统的概念模型围绕“用户-课程-排课”三条主线，设计了用户、学生、教师、课程、教学班、选课记录、课程安排、教室八大实体。用户表负责统一的登录凭据与权限字段，并通过一对一外键延伸出学生与教师两个角色实体，满足双端业务。课程实体保存课程名称、学分等基础信息，教学班记录具体班号与容量并关联课程与教师；学生与教学班之间通过选课记录建立多对多联系，实现余量校验与退选。排课层面，以教学班产生多条课程安排，课程安排再与教室建立多对一映射，从而支持智能排课、教室占用检测与时间冲突校验。上述实体及其关联覆盖了用户管理、选课管理、排课管理等核心需求，也为统计分析、容量预警等后续扩展场景提供了坚实的数据基础。

系统的整体E-R图如图2.22所示。

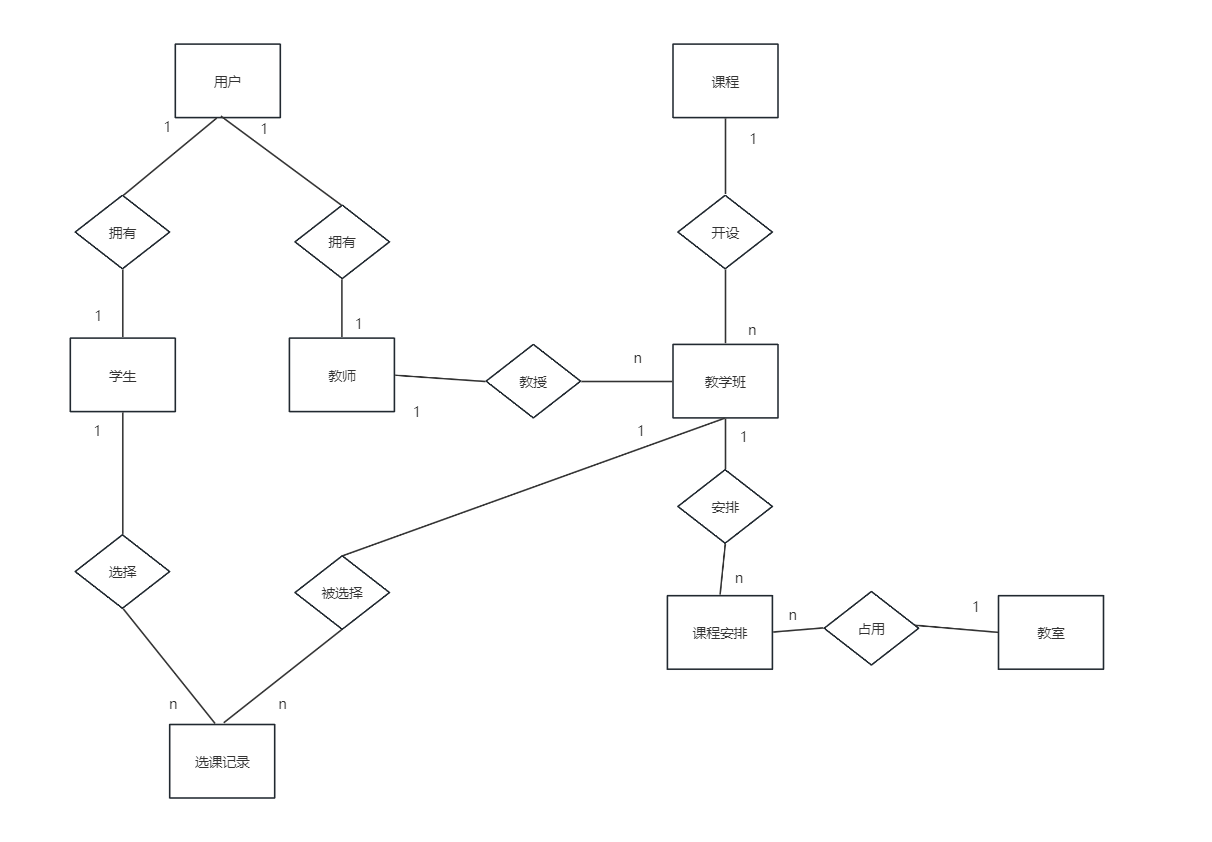


图 2.22系统整体E-R图

### 数据库表结构设计（逻辑模型设计）

根据上面对数据库概念模型的设计，下面将对数据库表的结构进行设计。

E-R图中user实体对应用户表详细设计如表2.1所示。

表 2.1用户表（user）

| 字段名 | 数据类型 | 长度 | 约束 | 默认值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| id | BIGINT | - | PK, AUTO\_INCREMENT | - | 用户唯一标识 |
| username | VARCHAR | 255 | UNIQUE, NOT NULL | - | 登录用户名 |
| password | VARCHAR | 255 | NOT NULL | - | 登录密码 |
| real\_name | VARCHAR | 255 | NOT NULL | - | 用户真实姓名 |
| role | ENUM | - | NOT NULL | - | 用户角色 |
| created\_at | DATETIME | - | - | CURRENT\_TIMESTAMP | 创建时间 |
| updated\_at | DATETIME | - | - | CURRENT\_TIMESTAMP | 更新时间 |

E-R图中student实体对应学生表详细设计如表2.2所示

表 2.2学生表（student）

| 字段名 | 数据类型 | 长度 | 约束 | 默认值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| id | BIGINT | - | PK | - | 学生ID |
| user\_id | BIGINT | - | FK, UNIQUE | - | 关联用户表ID |
| grade | VARCHAR | 50 | NOT NULL | - | 年级信息 |
| class\_name | VARCHAR | 100 | NOT NULL | - | 班级名称 |

E-R图中teacher实体对应教师表详细设计如表2.3所示

表2.3教师表（teacher）

| 字段名 | 数据类型 | 长度 | 约束 | 默认值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| id | BIGINT | - | PK | - | 教师ID |
| user\_id | BIGINT | - | FK, UNIQUE | - | 关联用户表ID |
| title | VARCHAR | 100 | NOT NULL | - | 职称 |
| department | VARCHAR | 200 | NOT NULL | - | 所属院系 |

E-R图中course实体对应课程表详细设计如表2.4所示

表 2.4课程表（course）

| 字段名 | 数据类型 | 长度 | 约束 | 默认值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| id | BIGINT | - | PK, AUTO\_INCREMENT | - | 课程唯一标识 |
| class\_code | VARCHAR | 50 | NOT NULL | - | 课程编码 |
| course\_name | VARCHAR | 200 | NOT NULL | - | 课程名称 |
| credit | DOUBLE | - | NOT NULL | - | 学分数 |
| hours | INTEGER | - | NOT NULL | - | 课时数 |
| description | TEXT | - | - | - | 课程描述 |
| created\_at | DATETIME | - | - | CURRENT\_TIMESTAMP | 创建时间 |

E-R图中classroom实体对应教室表详细设计如表2.5所示。

表 2.5教室表（classroom）

| 字段名 | 数据类型 | 长度 | 约束 | 默认值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| id | BIGINT | - | PK, AUTO\_INCREMENT | - | 教室唯一标识 |
| building | VARCHAR | 100 | NOT NULL | - | 教学楼名称 |
| classroom\_name | VARCHAR | 50 | NOT NULL | - | 教室号 |
| capacity | INTEGER | - | NOT NULL | - | 教室容量 |

E-R图中teachingclass实体对应教学班表详细设计如表2.6所示

表2.6教学班表（teachingclass）

| 字段名 | 数据类型 | 长度 | 约束 | 默认值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| id | BIGINT | - | PK, AUTO\_INCREMENT | - | 教学班唯一标识 |
| class\_code | VARCHAR | 100 | - | - | 教学班编码 |
| course\_id | BIGINT | - | FK, NOT NULL | - | 关联课程ID |
| teacher\_id | BIGINT | - | FK | - | 授课教师ID |
| max\_students | INTEGER | - | - | - | 最大学生数 |
| current\_students | INTEGER | - | - | - | 当前学生数 |

E-R图中course selection实体对应选课记录表如表2.7所示。

表 2.7选课记录表（course selection）

| 字段名 | 数据类型 | 长度 | 约束 | 默认值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| id | BIGINT | - | PK, AUTO\_INCREMENT | - | 选课记录唯一标识 |
| student\_id | BIGINT | - | FK, NOT NULL | - | 学生ID |
| teaching\_class\_id | BIGINT | - | FK, NOT NULL | - | 教学班ID |
| selection\_time | DATETIME | - | - | - | 选课时间 |

E-R图中class schedule实体对应课程安排表如表2.8所示。

表 2.8课程安排表（class schedule）

| 字段名 | 数据类型 | 长度 | 约束 | 默认值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| id | BIGINT | - | PK, AUTO\_INCREMENT | - | 课程安排唯一标识 |
| teaching\_class\_id | BIGINT | - | FK, NOT NULL | - | 教学班ID |
| classroom\_id | BIGINT | - | FK, NOT NULL | - | 教室ID |
| day\_of\_week | INTEGER | - | NOT NULL | - | 星期几 |
| start\_time | VARCHAR | 10 | NOT NULL | - | 开始时间 |
| end\_time | VARCHAR | 10 | NOT NULL | - | 结束时间 |

## 信息编码设计

### A 类编码及规则

1. 学号（student\_id）

结构：YYYYMMDDNNNN • YYYY 入学年份 • MM 院系代码 • DD 专业代码 • NNNN 当年顺序号。

特点：18 位数字，逻辑可解析，保证全校唯一。

1. 教师号（teacher\_id）

结构：T-XX-YYYY • T 常量 • XX 学院代码 • YYYY 行政顺序号。

特点：以 “T-” 开头，连字符分隔，便于与学号区分。

1. 课程号（course\_code）

结构：CYY-NNN • C 常量 • YY 学科门类代码 • NNN 课程流水号

1. 教学班号（class\_code）

结构：课程号 + -BXX（班次号），例如 C01-001-B02 表示第二教学班。

教室号（classroom\_code）

结构：BBB-RRR • BBB 楼栋编号 • RRR 房间号；同一校区保持唯一。

### B 类编码及规则

学科门类代码（subject\_code）

采用教育部《普通高等学校本科专业目录》两位门类代码；示例：08 表示工学。本系统仅在下拉框中引用上述代码及名称，无须建立完整明细表，更新时同步国家标准文件即可。

### C 类编码及其取值说明

1. gender：0 表示“女”，1 表示“男”，与 GB/T 2261.1-2003 性别代码保持一致。
2. user\_role：0 代表“管理员”，1 代表“学生”，2 代表“教师”，后端对应枚举 User.Role。
3. title\_level：1 为“讲师”，2 为“副教授”，3 为“教授”，用于区分教师职称层级。
4. weekday：取值 1 至 7，依次对应周一至周日，和 Java 的 DayOfWeek 保持一致。
5. course\_type：1 表示“必修”，2 表示“选修”，用于课程属性分类。

## 系统设计小结

本章系统设计从功能与数据两条主线展开。功能方面，平台以“通用认证、管理员后台、教师工作台、学生中心”四大模块构建：通用模块支撑登录验证与个人信息维护；后台聚焦用户、课程、教室与智能排课等管理职能；教师端提供课程信息维护、班级管理与排课查看；学生端实现在线选退课、课表查询及导出，并通过消息推送保持学习提醒。各模块通过 REST 接口解耦。数据库方面，概念模型设计了用户、学生、教师、课程、教学班、选课记录、课程安排、教室八大实体。用户与学生、教师一对一扩展，课程与教学班、教师与教学班均为一对多；学生与教学班通过选课记录形成多对多；教学班与课程安排一对多，课程安排与教室多对一。

# 系统实现

## 登录管理的实现

登录界面的主要流程如下：首先，通过 getText() 方法分别获取用户名输入框和密码输入框中的内容，再使用 toString() 方法将其转换为 String 类型，以便后续逻辑处理。接着，利用 if 语句判断用户名或密码是否为空：若任一字段为空，则调用 Toast.makeText() 提示用户对应信息不能为空；若两者均不为空，则调用 UserDao 中的 getUserById() 或 getUserByName() 方法检索数据库，判断该用户名是否存在。

若数据库中不存在该用户，则通过 Toast 提示“账号不存在，请先注册”；若用户存在，则进一步比较输入的密码与数据库中存储的密码是否一致。若密码不匹配，同样通过 Toast 提示“密码错误，请重新输入”；若密码匹配，则根据用户的 role 字段（如管理员、教师、学生）调用 Intent 跳转到对应的主页界面（例如 AdminDashboardActivity、TeacherDashboardActivity 或 StudentDashboardActivity）。同时，可在 SharedPreferences 中保存用户登录态信息，以支持自动登录或退出登录功能。整个登录成功流程结束后，通过 finish() 方法关闭当前 LoginActivity ，避免用户回退到登录界面。

用户登录界面如图3.1所示。用户登录流程图如图3.2所示。

|  |  |
| --- | --- |
| 图 3.1用户登录界面 | 图3.2用户登录流程图 |

## 主界面的实现

成功登录后，将进入系统主界面。该界面的设计直观且易于操作，主要分为顶部欢迎区域、中部核心功能模块和底部导航栏三大部分。界面的核心区域以卡片式（CardView）布局清晰地展示了三大主要功能入口：

1. 统计信息：此模块是系统数据的总览入口。通过为其视图对象设置onClickListener，当用户点击时，程序会创建一个新的Intent对象，启动用于展示统计图表信息的StatsActivity或加载对应的Fragment，从而跳转到数据显示页面。
2. 用户管理：该模块负责管理教师和学生账户。其实现方式与统计信息模块类似，点击后通过Intent跳转至UserManagementActivity，在此界面中，管理员可以对用户信息列表进行增、删、改、查等操作。数据的展示通常使用RecyclerView或ListView来高效地加载和显示用户列表。
3. 课程管理：用于维护系统中的所有课程信息。点击后，系统会跳转到CourseManagementActivity，管理员可以在此界面对课程进行全面的管理。

系统主界面如图3.3所示。主界面流程图如图3.4所示。

|  |  |
| --- | --- |
| 图 3.3系统主界面 | 图 3.4主界面流程图 |

## 个人中心的实现

个人中心是用户进行账户管理和查看个人信息的核心界面，可以通过主界面底部的“我的”导航标签进入。界面的实现逻辑如下：首先，在界面加载时，程序会从本地SharedPreferences或登录时传递的全局状态中读取当前用户的基本信息（如用户名、角色），并将其动态填充到顶部的用户信息卡片中，展示用户的头像和角色名称（例如“管理员”）。该界面主要包含以下功能模块：

1. 账户设置：这是一个列表项，通过为其视图对象注册onClickListener监听器，当用户点击时，程序会创建一个Intent，启动一个新的AccountSettingsActivity。在该活动中，用户可以修改密码、更新个人资料等。数据更新后，会通过调用UserDao中相应的方法将更改同步到数据库。
2. 消息通知：同样地，此选项也通过onClickListener进行响应。点击后，会跳转到NotificationActivity，用于展示系统公告、课程提醒等消息列表。
3. 退出登录：界面的底部设有一个“退出登录”按钮。它的实现方式为：首先为该按钮设置一个onClickListener。在点击事件的回调方法中，程序会执行一系列清理操作，包括清除保存在SharedPreferences中的用户登录凭证（如token和用户ID）。

|  |  |
| --- | --- |
| 图 3.5个人中心界面 | 图 3.6进入个人中心流程图 |

## 统计界面的实现

系统统计界面是管理员用于快速掌握系统整体运行状态的核心功能模块。该界面通过简洁明了的卡片式布局，直观地展示了各项关键业务数据的实时总量。该功能的实现流程如下：

1. 前端请求：当管理员从主界面点击“统计信息”进入此页面时，前端组件（如StatsMobile.vue）的生命周期钩子函数（例如mounted）会被触发。在此函数中，会调用一个预定义的API服务函数（例如定义在api/admin.js中的getStats()方法）。
2. 后端处理：该API服务函数会向后端的特定端点（例如/api/stats/summary）发起一个HTTP GET请求。后端的StatsController接收到请求后，会调用其处理方法。
3. 数据返回与渲染：StatsController将所有统计数据封装在一个数据传输对象（DTO）中，并以JSON格式返回给前端。

统计界面如图3.7所示。统计界面流程图如图3.8所示。

|  |  |
| --- | --- |
| 图 3.7统计界面 | 图 3.8统计界面流程图 |

## 用户管理界面的实现

用户管理是管理员的核心职能之一，该界面提供了对系统中所有用户（包括管理员、教师和学生）进行统一、高效管理的全部功能。其技术实现可以分为前端和后端两个部分；

**前端实现 (UserManagement.vue):**

1. 数据展示：当管理员进入此界面时，前端会立即调用API（例如admin.js中的getUsers方法），向后端请求第一页的用户数据。后端返回的数据是一个包含用户列表和分页信息的JSON对象。前端使用v-for指令将用户列表动态渲染成一个可滚动的列表（在移动端应用中，这通常由RecyclerView或ListView实现），每行显示用户的用户名和姓名。
2. 搜索功能：界面顶部的搜索框通过v-model与一个数据属性绑定。当用户输入关键词并点击搜索按钮时，会再次调用getUsersAPI，但这次会附带上搜索关键词作为查询参数。后端会根据关键词进行模糊匹配查询，并返回过滤后的结果，前端再重新渲染列表。
3. 添加用户：点击“+ 添加”按钮会弹出一个对话框（UserDialog.vue）。管理员在此对话框的表单中输入新用户的信息（用户名、初始密码、姓名、角色等），提交后，前端会将这些数据封装成一个对象，调用创建用户的API（createUser），发送一个POST请求到后端。成功后，会刷新当前用户列表。
4. 编辑与删除：点击列表中的任意一个用户，会打开同一个UserDialog.vue对话框，但此时对话框中已预先填充了该用户的现有信息。管理员可以修改信息并提交，触发更新用户的API调用（updateUser）。在该对话框中通常也包含一个“删除”按钮，点击后会调用删除用户的API（deleteUser），从系统中移除该用户。
5. 分页：界面底部的分页组件会显示总条目数和页码。点击不同的页码会触发getUsersAPI调用，并传入对应的页码参数，以获取和展示不同页的用户数据。

**后端实现 (AdminUserController.java & UserService.java):**

1. 获取用户列表：后端提供一个GET /api/admin/users接口，该接口支持分页（page, size）和搜索（keyword）参数。UserService会调用UserRepository，使用Spring Data JPA的Pageable对象和LIKE查询，从数据库中检索出匹配条件的用户数据，并封装成Page<User>对象返回。
2. 增删改查：后端相应地提供了POST /api/admin/users（创建）、PUT /api/admin/users/{id}（更新）和DELETE /api/admin/users/{id}（删除）等RESTful API接口，UserService中包含了对密码进行加密存储、检查用户名是否重复等业务逻辑。

用户管理界面如图3.9所示。用户管理流程图如图3.10所示。

|  |  |
| --- | --- |
| 图 3.9用户管理界面 | 图 3.10用户管理流程图 |

## 课程管理界面的实现

课程管理是系统教学资源配置的基础，该界面赋予管理员全面维护课程信息的能力，包括课程的创建、查询、修改和删除。其实现的技术细节如下；

**前端实现 (CourseManagement.vue):**

1. 数据获取与列表渲染：当管理员导航至此页面时，mounted生命周期钩子会触发，调用定义在api/admin.js中的getCourses方法，向后端发送GET请求以获取课程列表。获取到的数据（一个课程对象的数组）被存储在组件的data属性中，并通过v-for指令渲染成一个列表。每个列表项（ListView item）都清晰地展示了课程的名称和唯一的课程代码。
2. 搜索功能：界面上方的搜索框允许管理员通过输入课程名称或代码的关键字来快速筛选课程。搜索框的值通过v-model与一个数据属性双向绑定。点击搜索按钮或在输入时（通常配合防抖技术以优化性能），会附带搜索关键词再次调用getCoursesAPI。前端接收到筛选后的数据后，会立即更新视图，只显示符合条件的课程。
3. 添加课程：点击右上角的“+ 添加”按钮，系统会弹出一个表单模态框（Dialog）或跳转到一个新的页面。管理员需要在此表单中填写新课程的详细信息，如课程名称、课程代码、学分、学时等。提交表单后，前端会将这些信息封装为一个JSON对象，并通过createCourse API向后端发送一个POST请求。操作成功后，通常会自动刷新课程列表以显示新添加的课程。
4. 修改与删除课程：当管理员点击列表中的某一个课程时，程序会通过Intent或路由导航到课程的详情/编辑页面，并将该课程的ID作为参数传递。在详情页，课程的现有数据会被加载到表单中，管理员可以进行修改。提交修改会触发updateCourse API的调用（通常是PUT请求）。同时，此页面也会提供“删除”按钮，点击后会调用deleteCourse API（DELETE请求），将该课程从数据库中移除。在删除前，后端通常会进行校验，例如检查该课程是否已被教学班使用，以保证数据的完整性。

**后端实现 (CourseController.java & CourseService.java):**

1. 查询课程：后端的CourseController提供一个GET /api/courses的端点，用于处理课程的查询请求。该端点可以接收搜索关键词和分页参数。CourseService会调用CourseRepository的查询方法（如findByNameContainingIgnoreCaseOrCourseCodeContainingIgnoreCase），执行数据库查询，并返回课程数据列表。
2. 增删改操作：CourseController还定义了处理POST、PUT和DELETE请求的方法，分别对应课程的创建、更新和删除操作。CourseService中包含了核心的业务逻辑，例如在创建课程前检查课程代码的唯一性，或在删除课程前检查其关联的教学班是否存在。所有数据库操作最终都通过注入的CourseRepository实例来完成。

课程管理界面如图3.11所示。课程管理流程图如图3.12所示。

|  |  |
| --- | --- |
| 图 3.11课程管理界面 | 图 3.12课程管理流程图 |

## 学生选课界面的实现

学生选课界面是学生进行课程选择的核心功能页面。在此界面，系统会列出当前学期所有可供选择的教学班，学生可以根据自己的培养计划和兴趣进行筛选和提交选课申请。该功能的实现贯穿了前后端；

**前端实现 (SelectCourse.vue / SelectCourseMobile.vue):**

1. 数据加载与展示：页面初始化时，mounted钩子函数会执行，通过调用api/student.js中定义的getAvailableTeachingClasses方法，向后端请求所有可选的教学班列表。后端返回数据后，前端使用v-for指令将每个教学班渲染为一个列表项。每个列表项都清晰地展示了课程名称、授课教师、上课时间、地点和剩余容量等关键信息。
2. 搜索与筛选：为了方便学生快速定位课程，界面顶部通常会提供一个搜索框。通过v-model绑定一个数据属性，学生可以输入课程名称、教师姓名等关键词进行实时筛选，前端的计算属性（computed）或方法（method）会动态过滤列表，仅显示匹配的结果。
3. 选课操作：每个课程列表项的右侧都有一个“选择”按钮。该按钮绑定了一个点击事件，当学生点击时，会调用一个例如handleSelectCourse的方法，并将该教学班的ID作为参数传入。该方法会立即调用api/student.js中的submitSelection函数，向后端API（例如 POST /api/selections）发起一个请求，提交选课申请。
4. 实时反馈：后端处理选课请求后，会返回成功或失败的信息。如果选课成功，前端会通过Toast.makeText()等方式弹出“选课成功”的提示，并且可能会将该课程的“选择”按钮置灰或改变其状态，以防止重复提交。如果后端返回错误（如时间冲突、容量已满），前端会捕获该错误，并将具体的错误信息（如“上课时间冲突，选课失败”）提示给学生。

**后端实现 (SelectionController.java & SelectionService.java):**

1. 提供可选课程列表：后端的CourseController或TeachingClassController中有一个GET接口（如/api/teaching-classes/available），用于返回所有状态为可选的教学班。CourseService会从TeachingClassRepository中查询所有记录，并可能会过滤掉容量已满或不符合当前学期的课程。
2. 处理选课请求：SelectionController提供一个POST接口（如/api/selections），用于接收学生的选课请求。该接口会接收包含teachingClassId的请求体。
3. 核心业务逻辑：SelectionService中的createSelection方法是选课功能的核心。在将选课记录存入数据库之前，它会执行一系列严格的校验，并把这些操作包裹在一个数据库事务（@Transactional）中以确保数据一致性：

* 容量检查：查询该教学班的已选人数是否已达到容量上限。
* 时间冲突检测：查询该学生已选的所有课程，将其上课时间与待选课程的上课时间进行比对，判断是否存在时间重叠。
* 重复选课检查：判断学生是否已经选过同一门课程的同一个教学班。
* 如果所有检查都通过，系统才会在CourseSelection表中创建一条新的记录，并更新TeachingClass表中的已选人数。若任一检查失败，则会抛出一个自定义的ConflictException异常，并附带具体的失败原因。全局异常处理器（GlobalExceptionHandler）会捕获此异常，并向前端返回相应的HTTP错误状态码和错误信息。

学生选课界面如图3.13所示。选课流程图如图3.14所示

|  |  |
| --- | --- |
| 图 3.13选课界面 | 图3.14选课流程图 |

## 学生查看课表界面的实现

学生个人课表界面是学生日常学习的重要工具，它以直观的表格形式展示了学生已选课程在不同时间段的安排，方便学生按时上课和规划学习。该功能的实现涉及前端的动态渲染和后端的数据聚合；

**前端实现 (MySchedule.vue):**

1. 数据请求：当学生进入“我的课表”页面时，mounted生命周期钩子会立即被调用。在此钩子中，前端会通过调用api/schedule.js中定义的getMySchedule方法，向后端API端点（例如 GET /api/schedules/my）发送请求，以获取当前登录学生的个人课表数据。
2. 数据结构：后端返回的通常是一个结构化的JSON对象。为了便于前端渲染一个7天N节的表格，这个JSON对象可能会被组织成一个二维数组或者一个以“星期X-第Y节”为键的对象。例如，{ "1-1": { courseName: "Java", classroom: "A101" }, "1-3": { ... } }，其中1-1代表周一第一二节课。
3. 课表渲染：前端拿到数据后，会使用v-for指令来构建一个表格（<table>）或使用CSS Grid/Flexbox布局来模拟表格。外层循环遍历周一到周日，内层循环遍历每天的节次（如1-2节, 3-4节等）。在渲染每个单元格时，前端会根据当前的星期和节次，去查找后端返回的数据对象中是否存在对应的课程。如果存在，就在单元格中填充课程名称、教室等信息；如果不存在，则显示为空白。
4. 交互设计：为了提升用户体验，可以为有课的单元格添加点击事件。当用户点击某个课程格子时，可以弹出一个对话框（Dialog），显示该课程更详细的信息，如完整的上课周次、授课教师、学分等。

**后端实现 (ScheduleController.java & ScheduleService.java):**

1. 接口定义：ScheduleController中定义了一个GET请求的端点，如@GetMapping("/my")，专门用于处理获取个人课表的请求。该方法会从安全上下文（SecurityContextHolder）中获取当前登录用户的ID。
2. 数据查询：ScheduleService中的getMySchedule方法是核心。它首先会根据学生ID，从CourseSelectionRepository中查询出该学生所有已成功选择的课程记录（CourseSelection列表）。
3. 数据处理与聚合：接着，服务层会遍历这个选课列表。对于每一个选课记录，它会进一步通过关联关系，查询到对应的TeachingClass（教学班），然后是ClassSchedule（上课时间安排）。ClassSchedule实体中包含了星期几（dayOfWeek）、开始节次（startPeriod）、结束节次（endPeriod）以及教室（classroom）等关键信息。
4. 格式化输出：ScheduleService会将查询和处理后的课程安排聚合成一个前端友好的数据结构（如Map<String, ScheduleDTO>）。这个Map的键由“星期-节次”拼接而成，值是一个包含课程名称、教师、教室等信息的ScheduleDTO对象。最后，ScheduleController将这个Map序列化为JSON格式并返回给前端。

通过这种方式，系统将分散的选课数据和排课数据动态地整合成一个清晰、个性化的学生个人课表。

课表界面如图3.15所示。查看课表流程图如图3.16所示。

|  |  |
| --- | --- |
| 图 3.15课表界面 | 图 3.16查看课表流程图 |

# 系统测试

本阶段系统测试遵循“功能完备、数据准确、交互友好、权限安全”的原则，采用黑盒为主、白盒为辅的方法，覆盖学生、教师、管理员三端的核心业务流与异常场景。测试以功能模块为单元，将接口调用、界面操作与数据库验证相结合，确保注册、选课、排课等关键流程在不同角色和边界条件下均能得到正确响应与数据一致性。通过用例编号化管理与自动化脚本比对，形成可回溯的测试基线，支撑后续迭代的持续集成。

## 系统测试环境

测试环境如表4.1所示。

表 4.1环境配置

|  |  |
| --- | --- |
| 名称 | 配置 |
| 处理器 | Intel i5-11300H |
| 硬盘 | 1T |
| 内存 | 16G |
| 操作系统 | Windows11 64位 |
| IDEA版本 | IntelliJ IDEA 2024.1.2 |
| JDK版本 | 17 |
| 数据库 | MYSQL |

## 系统功能测试

在本章中，我们编制并执行表 4.2 至 表 4.13 的测试用例，以系统化地验证各功能模块与异常处理是否符合需求。测试过程中不仅针对正常流程输入操作，还刻意输入无效数据或执行越权动作，观察系统是否给出正确的错误提示并记录结果，再与预期值比对，及时定位问题并反馈修复。需由顾客或商家角色完成的 4.6 与 4.11 两组操作，为提高可重复性与效率，本章采用直接向数据库插入测试数据的方式模拟，确保覆盖所有关键分支与边界条件，从而为后续迭代提供可靠的质量保障。

表4.2 用户注册测试用例

| 用例编号 | 功能描述 | 输入/动作 | 预期结果 | 实际情况 |
| --- | --- | --- | --- | --- |
| 101 | 注册学生用户 | 输入空用户名，点击注册 | 提示“用户名不能为空” | 正常 |
| 102 | 注册学生用户 | 输入空密码，点击注册 | 提示“密码不能为空” | 正常 |
| 103 | 注册学生用户 | 输入重复用户名，点击注册 | 提示“用户名已存在” | 正常 |
| 104 | 注册学生用户 | 输入完整合法信息，点击注册 | 提示“注册成功”，跳转登录页面 | 正常 |

表4.3 用户登录测试用例

| 用例编号 | 功能描述 | 输入/动作 | 预期结果 | 实际情况 |
| --- | --- | --- | --- | --- |
| 201 | 登录系统 | 输入空账号，点击登录 | 提示“请输入用户名” | 正常 |
| 202 | 登录系统 | 输入错误账号或密码，点击登录 | 提示“用户名或密码错误” | 正常 |
| 203 | 登录系统 | 输入正确信息但选择错误角色 | 提示“角色不匹配” | 正常 |
| 204 | 登录系统 | 输入正确用户名密码和身份，点击登录 | 成功跳转至对应用户首页 | 正常 |

表4.4 密码重置测试用例

| 用例编号 | 功能描述 | 输入/动作 | 预期结果 | 实际情况 |
| --- | --- | --- | --- | --- |
| 301 | 修改密码 | 输入空密码并提交 | 提示“新密码不能为空” | 正常 |
| 302 | 修改密码 | 两次输入密码不一致，点击确认 | 提示“两次密码不一致” | 正常 |
| 303 | 修改密码 | 正确输入两次相同的新密码，点击确认 | 提示“修改成功”，返回登录界面 | 正常 |

表4.5 个人信息修改测试用例

| 用例编号 | 功能描述 | 输入/动作 | 预期结果 | 实际情况 |
| --- | --- | --- | --- | --- |
| 401 | 修改个人信息 | 留空姓名或其他必要字段，点击保存 | 提示“字段不能为空” | 正常 |
| 402 | 修改个人信息 | 输入合法信息，点击保存 | 提示“修改成功” | 正常 |

表4.6 选课功能测试用例

| 用例编号 | 功能描述 | 输入/动作 | 预期结果 | 实际情况 |
| --- | --- | --- | --- | --- |
| 501 | 进入选课界面 | 学生点击“前往选课”按钮 | 成功加载选课列表 | 正常 |
| 502 | 提交选课 | 选择冲突课程，点击选课按钮 | 提示“课程冲突，无法选课” | 正常 |
| 503 | 提交选课 | 选择已满课程，点击选课按钮 | 提示“课程已满，无法选课” | 正常 |
| 504 | 提交选课 | 选择正常课程，点击选课按钮 | 提示“选课成功”，课程添加至课表 | 正常 |

表4.7退课测试用例

| 用例编号 | 功能描述 | 输入/动作 | 预期结果 | 实际情况 |
| --- | --- | --- | --- | --- |
| 601 | 退选课程 | 点击已选课程的“退课”按钮，确认退课 | 提示“退课成功”，课表更新 | 正常 |
| 602 | 退选课程 | 尝试重复退选相同课程 | 提示“您尚未选择该课程” | 正常 |

表4.8 课表查看测试用例

| 用例编号 | 功能描述 | 输入/动作 | 预期结果 | 实际情况 |
| --- | --- | --- | --- | --- |
| 701 | 查看课表 | 登录后点击“我的课表”按钮 | 正确显示个人课表信息 | 正常 |
| 702 | 查看课表 | 无选课记录，点击“我的课表” | 显示“暂无课表信息” | 正常 |

表4.9 教师课程管理测试用例

| 用例编号 | 功能描述 | 输入/动作 | 预期结果 | 实际情况 |
| --- | --- | --- | --- | --- |
| 801 | 管理课程 | 教师点击“我的课程”，查看课程详情 | 显示课程基本信息与教学班信息 | 正常 |
| 802 | 教学班操作 | 教师点击编辑教学班并修改信息，点击保存 | 提示“修改成功”，教学班信息更新 | 正常 |
| 803 | 教学班操作 | 教师点击删除教学班并确认 | 提示“删除成功”，教学班消失 | 正常 |

表4.10 管理员用户管理测试用例

| 用例编号 | 功能描述 | 输入/动作 | 预期结果 | 实际情况 |
| --- | --- | --- | --- | --- |
| 901 | 添加用户 | 输入完整有效信息点击添加 | 提示“添加成功”，列表显示新用户 | 正常 |
| 902 | 添加用户 | 留空必填项点击添加 | 提示“请填写完整信息” | 正常 |
| 903 | 删除用户 | 点击删除按钮确认删除 | 用户被删除，不再显示 | 正常 |
| 904 | 编辑用户信息 | 修改用户信息点击保存 | 信息更新成功 | 正常 |

表4.11 管理员课程管理测试用例

| 用例编号 | 功能描述 | 输入/动作 | 预期结果 | 实际情况 |
| --- | --- | --- | --- | --- |
| 1001 | 添加课程 | 输入完整信息点击提交 | 新课程添加成功，显示在课程列表中 | 正常 |
| 1002 | 删除课程 | 选择课程点击删除 | 成功删除，不再显示 | 正常 |
| 1003 | 编辑课程信息 | 修改课程基本信息点击保存 | 信息修改成功 | 正常 |

表4.12 管理员课程管理测试用例

| 用例编号 | 功能描述 | 输入/动作 | 预期结果 | 实际情况 |
| --- | --- | --- | --- | --- |
| 1101 | 一键排课 | 课程资源完整，点击“一键排课”按钮 | 成功生成课表，并提示排课成功 | 正常 |
| 1102 | 冲突检测 | 存在教室/教师冲突时点击排课 | 系统提示冲突详情并阻止排课 | 正常 |
| 1103 | 发布课表 | 无冲突后点击“发布课表”按钮 | 提示发布成功，课表生效 | 正常 |

表4.13 管理员教室管理测试用例

| 用例编号 | 功能描述 | 输入/动作 | 预期结果 | 实际情况 |
| --- | --- | --- | --- | --- |
| 1201 | 添加教室 | 输入有效信息点击“添加”按钮 | 提示“添加成功”，教室显示在列表中 | 正常 |
| 1202 | 删除教室 | 点击“删除”按钮确认删除操作 | 教室成功删除 | 正常 |

# 结论与展望

本项目成功设计并实现了一个功能全面的现代化高校智能排课系统。系统基于主流的前后端分离架构，后端采用稳定可靠的 Java Spring Boot 框架，前端则使用灵活高效的 Vue.js 框架，实现了清晰的职责分离和良好的可维护性。项目完整地覆盖了排课系统的核心业务流程，成功为管理员、教师、学生三种关键角色提供了定制化的操作平台：

1. 管理员端实现了对用户、课程、教室等基础信息的全面管理，并提供了强大的一键式智能排课核心功能，极大地提高了排课效率与准确性，解决了传统手动排课中常见的冲突与资源浪费问题。
2. 教师端提供了便捷的教学班管理、个人课表查询等功能，使教师能够清晰地掌握自己的教学任务。
3. 学生端则提供了直观的课程查询、在线选课、个人课表查看等功能，优化了学生的选课体验。

系统设计规范，文档齐全（包含ER图、UML时序图、用例图等），展现了扎实的软件工程实践能力。同时，前端界面考虑了响应式设计，能够适配不同设备，为用户提供了流畅、友好的交互体验。总体而言，本项目不仅成功解决了高校排课的实际痛点，更构建了一个技术先进、功能完善、扩展性强的综合性教务管理解决方案。

尽管当前系统已具备高度的可用性，但仍可在以下方面进行深化与拓展，以适应未来更多样化的需求：

1. 核心算法优化：目前的智能排课算法可进一步引入更复杂的约束条件（如教师偏好、教室多功能性等），并探索如遗传算法、模拟退火等更前沿的优化算法，以应对更大规模、更复杂的排课场景，生成满意度更高的课表方案。
2. 消息通知系统：集成邮件或站内信功能，用于自动通知用户课表变更、选课结果、重要公告等，提升服务的即时性与人性化。
3. 考试安排模块：在现有排课逻辑基础上，拓展支持期中、期末考试的自动化编排功能。
4. 数据可视化与统计分析：为管理员提供更丰富的可视化图表，深度分析课程热门度、教室利用率、学生选课行为等，为教学管理决策提供数据支持。
5. 系统集成与生态建设：探索与校园内其他信息系统（如学生信息系统SIS、在线学习平台LMS）的接口打通，实现数据互联互通，消除信息孤岛，构建一体化的智慧校园生态。
6. 性能与体验提升：针对高并发场景（如抢课高峰），可引入缓存技术（如Redis），并对数据库查询进行持续优化，保障系统的稳定与快速响应。同时，持续收集用户反馈，对UI/UX进行迭代打磨。

我们相信，通过上述方向的持续演进，本系统将能更好地服务于现代化教育管理，成为智慧校园建设中不可或缺的一环。