



软件工程课程设计报告

**教务查询系统**

|  |  |  |
| --- | --- | --- |
| 学 | 院 | **计算机学院** |
| 专 | 业 | **软件工程** |

年级班级 **22软件工程卓越班**

学 号 **3222004465**

学生姓名 **陈婉瑜**

指导教师 **欧毓毅**

2025 年 1 月

**目录**

[摘要 2](#_Toc187456844)

[一、可行性研究 3](#_Toc187456846)

[二、需求分析 4](#_Toc187456847)

[三、数据库设计 15](#_Toc187456848)

[四、概要设计 18](#_Toc187456849)

[4.1模块功能描述： 18](#_Toc187456850)

[4.2系统结构设计 19](#_Toc187456851)

[4.3数据结构设计： 19](#_Toc187456852)

[五、详细设计 21](#_Toc187456853)

[5.1项目结构 21](#_Toc187456854)

[5.2数据结构具体实现 22](#_Toc187456855)

[5.3模块功能实现 24](#_Toc187456856)

[六、测试与实现 33](#_Toc187456857)

[测试方案 33](#_Toc187456858)

[测试报告 35](#_Toc187456859)

[核心功能截图 36](#_Toc187456860)

[七、总结 40](#_Toc187456861)

[八、参考文献 41](#_Toc187456862)

## 摘要

教育信息化是国家教育现代化的重要内容，在当今数字化时代，高校教务管理系统作为教育信息化的关键支撑，对于提升教学管理效率、优化资源配置以及保障教学质量起着举足轻重的作用。随着学校规模的扩大，人员的不断增加，学校中教务是一项繁琐的事。每学期的考试成果都须要分析，纵、横向比较反馈信息，还要对老师的教学成果考核供应数据。以前简洁的用Excel 来打印几张报表的人机作坊再也无法适应当今学校的教学管理了， 使用Exce1初期虽然能简洁管理学生的学籍、成果等，但时间一长,将产生大量的文件和数据,这对于查找、更新和维护都带来了不少困难，并且在操作过程中没有用户权限，只要能打开计算机的人就能打开数据进行随意的修改,大大的降低了信息的安全性和保密性。

传统的教务管理模式已难以满足日益增长的管理需求，教务信息查询系统应运而生。我将基于目前对高校教务管理系统的研究现状以及实际需求，设计一个简单的高校教务查询系统。

系统主要用户：教师、学生、管理员

* 管理员：对课程、学生信息、教师信息等进行统一管理，重置系统用户登录密码，修改个人密码
* 教师：查看自己教授的课程，查询选修该课程的学生，对选修该课程的学生进行打分，修改个人密码
* 学生：可以进行选课，查看已修课程，查看已选课程，退选课程，修改个人密码

该系统能够整合各类教务信息，为管理员、教师和学生提供便捷、高效的信息查询与管理服务，极大地减轻了教务人员的工作负担，提高了管理的精准性和及时性以及数据的保密性和安全性。

教务查询系统

## **一、可行性研究**

目前高校使用的教务系统有web应用系统、桌面应用系统、小程序等类型。大多教务管理系统存在功能局限、兼容性较差等问题。部分老系统只能在特定的操作系统环境下运行，难以适应如今多元化的终端设备访问需求。而且，旧系统的技术架构可能相对陈旧，后续拓展新功能和进行维护升级成本较高、难度较大。

**开发方案：**

（1）Web应用系统：采用浏览器访问的方式，可以跨平台操作，便于维护和更新，适合大规模用户同时操作，符合高校日益增长的管理需求。

（2）桌面应用系统：适合单一设备操作，数据安全性高，但跨平台性较差，维护成本较高。

（3）云端解决方案：通过云计算平台进行数据存储和管理，能够实现高效的数据处理和安全性保障。但需要依赖稳定的互联网连接，且云端服务费用可能较高。

综合考虑,选择第一个方案，开发一个基于BS架构的教务查询系统。经济上，开发基于Web的高校教务查询系统所需的投入主要是初期开发成本，包括开发团队人员费用、开发工具、硬件设备的采购以及系统上线投入使用后的维护更新成本。但从长期来看，系统投入使用后可大幅度节省管理数据的人工成本、系统的权限控制和数据备份功能可以减少因数据丢失或泄露带来的经济损失并且Web系统易于维护和扩展，不需要频繁的更新硬件。社会上，随着高校规模的扩大和信息化建设的推进，教师、学生和管理员对教务信息管理的需求日益增加。教务查询系统能够整合各种教学资源和管理数据，满足各类用户的需求，提升教学管理质量和效率。推动高校信息化管理的发展，为学校其他业务系统的数字化和智能化提供基础支持。

综合分析技术、经济和社会可行性后，可以得出结论：开发一个基于Web的高校教务查询系统是完全可行的。该系统能够解决当前高校教务管理中存在的问题，提高工作效率，确保数据安全，且符合教育信息化的长远发展需求。

## 二、需求分析

**1.性能需求：**

1.1 响应时间：

用户登录系统的响应时间应不超过2秒。用户增删改查等操作响应时间不超过5秒。

1.2系统稳定性：

系统在发生硬件或软件故障后，应能在1小时内恢复正常运行（包括自动重启和手动干预）。

系统定期自动备份数据，备份数据应能在必要时快速恢复，恢复时间不超过2小时。

1.3并发处理能力：

系统应能支持至少1000个并发用户同时在线操作，包括但不限于查询、选课、提交成绩等。

1.4安全性：

系统应对敏感数据（如用户密码、成绩等）进行加密存储和传输，确保数据安全。

系统应实施严格的访问控制策略，确保只有授权用户才能访问和修改数据。

1.5用户界面与体验：

系统界面应简洁明了，易于用户理解和操作。提供清晰、具体的错误提示信息，帮助用户快速定位并解决问题。

**2.功能需求：**

**2.1 管理员功能需求分析**

教师管理：

1.添加、修改、删除教师信息：管理员可以手动输入教师的基本信息（如用户ID、用户姓名、教师性别、出生日期、职称/职位、入职日期、院系ID/所属院系），对于已存在的教师信息，管理员可以进行修改或删除信息，以便系统能够准确记录并管理教师资源。

2.查询教师信息：管理员可以根据姓名快速检索到对应的教师信息

学生管理：

3.添加、修改、删除学生名单：管理员可以手动输入学生的基本信息（如用户ID、用户姓名、学生性别、出生日期、入学时间、院系ID/所属院系等），以便系统能够准确记录并管理学生资源。

4.查询学生名单：管理员可以根据姓名快速检索到对应的学生信息

用户管理：

5.重置密码：当教师或学生忘记自己的登录密码时，管理员可以为其重置密码。并且可以修改个人密码

课程管理：

6.管理课程：增删改查课程信息

**2.2学生功能需求分析**

1.查询信息：学生可以登录系统查看已修课程，查看已选课程，包括课程名称、授课老师、上课时间、地点、学分、成绩等信息

2. 选课退课：查看所有课程信息，并进行选课，退课操作

**2.3教师功能需求分析**

1．查询信息：查看自己教授的课程，查询选修该课程的学生名单

2．提交成绩：教师可以在系统中提交自己所教课程的学生成绩

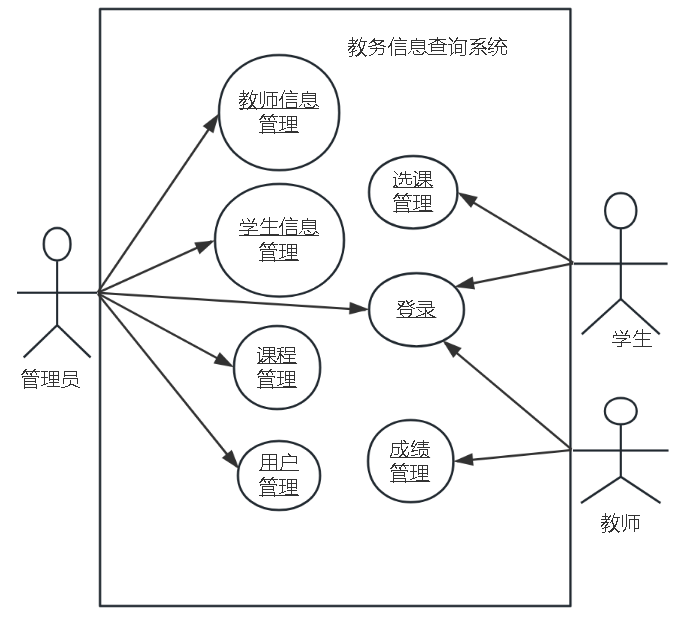


图1：用例图

**3．用例描述**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 1 | 名称 | **教师信息管理** | 优先级 | 高 |
| 参与者 | 管理员，目标是对学校所有教师信息进行统一管理，可以增加新入职教师信息，删除离职教师信息，修改教师信息 | | | | |
| 触发条件 | 有新教师入职或教师信息需要更新/删除。 | | | | |
| 前置条件 | 管理员登录系统成功，进入教师管理界面。 | | | | |
| 后置条件 | 系统根据管理员操作，相应地更新教师信息存储，确保教师信息目录准确反映最新情况，包括教师编号、姓名、性别、职称 / 职位、入职日期、院系 ID / 所属院系等信息准确无误。 | | | | |
| 正常流程 | 1. 管理员点击 “教师管理” 功能入口，打开教师信息操作界面。 2. 系统展示当前已有的教师信息列表 3. 若要添加教师信息，管理员点击 “添加教师” 按钮。 4. 系统弹出添加教师信息的输入框，提示管理员依次输入教师的基本信息（包括教师编号、姓名、教师性别、出生日期、职称 / 职位、入职日期、所属院系）。 5. 管理员按要求输入完成后点击 “保存” 按钮。 6. 系统存储新教师信息并更新教师信息目录，再次显示更新后的教师信息列表。 7. 若要查询教师信息，管理员在查询框中输入教师姓名，点击 “搜索” 按钮。 8. 系统根据输入的姓名检索并显示对应的教师信息并显示 9. 若要修改教师信息，管理员在教师信息列表中选择要修改的教师条目，点击 “修改” 按钮。 10. 系统弹出该教师信息的编辑框，管理员修改相应信息后点击 “保存” 按钮。 11. 系统存储修改后的信息并更新教师信息目录，重新显示更新后的教师信息列表。 12. 若要删除教师信息，管理员在教师信息列表中选择要删除的教师条目，点击 “删除” 按钮。   13.系统弹出确认删除提示框，管理员确认后，将该教师信息从教师信息列表及存储中删除，并刷新显示剩余教师信息列表。 | | | | |
| 扩展流程 | 3a. 管理员在输入教师基本信息过程中，若输入已存在的教师编号，再点击下一个信息输入框时系统提示信息“教师编号已存在，请重新输入”，并阻止信息提交，直到输入符合要求为止。 | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 2 | 名称 | **学生信息管理** | 优先级 | 高 |
| 参与者 | 管理员，目标是准确管理学生资源，对学生的各类信息进行添加、修改、删除以及查询操作，保障系统中学生信息的准确性和完整性。 | | | | |
| 触发条件 | 学校有新生入学、学生信息变更或需查询特定学生信息时。 | | | | |
| 前置条件 | 管理员登录系统成功，进入学生管理界面。 | | | | |
| 后置条件 | 系统根据管理员操作，相应地更新学生信息存储，确保学生信息目录准确反映最新情况，包含学生编号、姓名、性别、出生日期、入学时间、院系 ID / 所属院系等信息准确无误。 | | | | |
| 正常流程 | 1.管理员点击 “学生管理” 功能入口，打开学生信息操作界面。  2.系统展示当前已有的学生信息列表  3.若要添加学生信息，管理员点击 “添加学生” 按钮。  4.系统弹出添加学生信息的输入框，提示管理员依次输入学生的基本信息  5.管理员按要求输入完成后点击 “保存” 按钮。  6.系统存储新学生信息并更新学生信息目录，，并再次显示更新后的学生信息列表。  7.若要查询学生信息，管理员在查询框中输入学生姓名，点击 “查询” 按钮。  8.系统根据输入的姓名检索并显示对应的学生信息。  9.若要修改学生信息，管理员在学生信息列表中选择要修改的学生条目，点击 “修改” 按钮。  10.系统弹出该学生信息的编辑框，管理员修改相应信息后点击 “确认修改” 按钮。  11.系统存储修改后的信息并更新学生信息目录，提示修改成功，重新显示更新后的学生信息列表。  12.若要删除学生信息，管理员在学生信息列表中选择要删除的学生条目，点击 “删除” 按钮。  13系统弹出确认删除提示框，管理员确认后，将该学生信息从学生信息列表及存储中删除，提示删除成功，并刷新显示剩余学生信息列表。 | | | | |
| 扩展流程 | 3a. 管理员在输入教师基本信息过程中，若输入已存在的学生编号，再点击下一个信息输入框时系统提示信息“学生编号已存在，请重新输入”，并阻止信息提交，直到输入符合要求为止。 | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 3 | 名称 | **课程管理** | 优先级 | 高 |
| 参与者 | 管理员，目标是对学校开设的课程信息进行全面管理，包括课程的添加、修改、删除以及查询操作，确保课程信息的准确性和及时性，以满足教学安排需求。 | | | | |
| 触发条件 | 学校有新课程开设、课程信息变更或需查询特定课程信息时。 | | | | |
| 前置条件 | 管理员登录系统成功，进入课程管理相关界面。 | | | | |
| 后置条件 | 系统根据管理员操作，相应地更新课程信息存储，确保课程信息目录准确反映最新情况，包含课程的各项关键信息（如课程名称、授课老师、上课时间、地点、学分等）准确无误。 | | | | |
| 正常流程 | 1．管理员点击 “课程管理” 功能入口，打开课程信息操作界面。  2.系统展示当前已有的课程信息列表，包含课程名称、授课老师、上课时间、地点、学分等关键信息。  3.若要添加课程信息，管理员点击 “添加课程” 按钮。  4.系统弹出添加课程信息的输入框，提示管理员依次输入课程的基本信息（如课程名称、授课老师、上课时间、地点、学分等）。  5.管理员按要求输入完成后点击 “保存” 按钮。  6.系统存储新课程信息并更新课程信息目录，并再次显示更新后的课程信息列表。  7.若要查询课程信息，管理员在查询框中输入课程相关关键词（如课程名称、授课老师姓名等），点击 “查询” 按钮。  8.系统根据输入的关键词检索并显示对应的课程信息。  9.若要修改课程信息，管理员在课程信息列表中选择要修改的课程条目，点击 “修改” 按钮。  10.系统弹出该课程信息的编辑框，管理员修改相应信息后点击 “保存” 按钮。  11.系统存储修改后的信息并更新课程信息目录，重新显示更新后的课程信息列表。  12.若要删除课程信息，管理员在课程信息列表中选择要删除的课程条目，点击 “删除” 按钮。  13.系统弹出确认删除提示框，管理员确认后，将该课程信息从课程信息列表及存储中删除，提示删除成功，并刷新显示剩余课程信息列表。 | | | | |
| 扩展流程 | 3a. 管理员在输入课程信息过程中，若输入已存在的课程编号，再点击下一个信息输入框时系统提示信息“课程编号已存在，请重新输入”，并阻止信息提交，直到输入符合要求为止。 | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 4 | 名称 | **用户管理** | 优先级 | 中 |
| 参与者 | 管理员，目标是帮助忘记登录密码的教师或学生重置密码，以及管理员可自行修改个人密码，保障用户能正常登录系统。 | | | | |
| 触发条件 | 教师或学生反馈忘记密码，或管理员自身需要修改密码时。 | | | | |
| 前置条件 | 管理员登录系统成功，进入用户管理相关密码操作界面。 | | | | |
| 后置条件 | 教师或学生的登录密码被成功重置或修改，相应的密码信息在系统中更新存储，下次登录时需使用新密码。 | | | | |
| 正常流程 | * 1. 管理员选择 “账号密码重置 / 个人密码修改” 选项，进入相应界面。   2. 若为教师或学生重置密码，管理员输入需重置密码的用户账号，并输入重置的新密码，点击“提交”按钮   3. 系统将新密码更新到对应用户的密码存储字段。   4. 若管理员修改自身密码，先输入原密码，再输入新密码及确认新密码，点击 “提交” 按钮。   5. 系统验证原密码正确且新密码两次输入一致后，更新管理员的密码存储信息，返回登录页面 | | | | |
| 扩展流程 | 2a. 管理员在输入重置密码账户过程中，若输入的是管理员账户，当输入完毕点击“提交”按钮时，系统重置密码失败，提示“该账户为管理员账户，不可修改” | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 5 | 名称 | **登录** | 优先级 | 高 |
| 参与者 | 学生、教师、管理员，目标是依据不同身份登录系统显示用户对应板块，对应功能 | | | | |
| 触发条件 | 用户需要进行相关操作 | | | | |
| 前置条件 | 浏览器成功打开登录界面 | | | | |
| 后置条件 | 登陆成功，显示用户对应界面 | | | | |
| 正常流程 | 1. 输入账号和密码并点击登录 2. 系统登陆成功，跳转到用户对应界面 | | | | |
| 扩展流程 | 1a输入错误信息，账号密码不匹配，系统登录失败，提示“请重新输入” | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 6 | 名称 | **成绩管理** | 优先级 | 高 |
| 参与者 | 教师，目标是通过系统查看教授课程的学生信息、提交成绩等，管理教学任务。 | | | | |
| 触发条件 | 教师需要查询自己教授的课程、选修该课程的学生名单或提交学生成绩。 | | | | |
| 前置条件 | 教师登录系统成功。 | | | | |
| 后置条件 | 课程信息、学生名单被正确显示，学生成绩被正确提交。 | | | | |
| 正常流程 | 1. 教师点击课程信息。 2. 系统显示教师所教授的所有课程信息 3. 教师选定需要查看成绩的课程，并点击“成绩”按钮 4. 系统显示选修该课程的所有学生信息 5. 教师点击需要打分学生的“打分”按钮 6. 系统显示打分页面 7. 教师输入分数并点击提交 8. 系统保存成绩并跳转到该门课程学生页面并显示成绩 | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | 7 | 名称 | **选课管理** | 优先级 | 高 |
| 参与者 | 学生，目标是根据学校开设的课程信息，自主进行选课和退课操作，合理安排自己的学习计划。 | | | | |
| 触发条件 | 学生有选课需求，或学生因课程调整等原因需要退课。 | | | | |
| 前置条件 | 学生使用正确的账号和密码登录系统成功，进入选课退课相关界面 | | | | |
| 后置条件 | 系统根据学生的选课或退课操作，更新学生的选课记录，确保选课和退课结果准确反映在系统中。 | | | | |
| 正常流程 | * 1. 学生在课程列表中选择感兴趣的课程，点击 “选课” 按钮。   2. 系统将该课程添加到学生已选课程记录中，跳转到已选课程页面并显示”；   3. 若要退课，学生点击进入已选课程列表中选择要退课的课程，点击 “退课” 按钮。   系统将该课程从学生已选课程记录中删除 | | | | |
| 扩展流程 | 2a若学生已选修过该课程，系统提示“该门课程你已经选了，不能再选”，选课失败 | | | | |

**4.数据流图：**

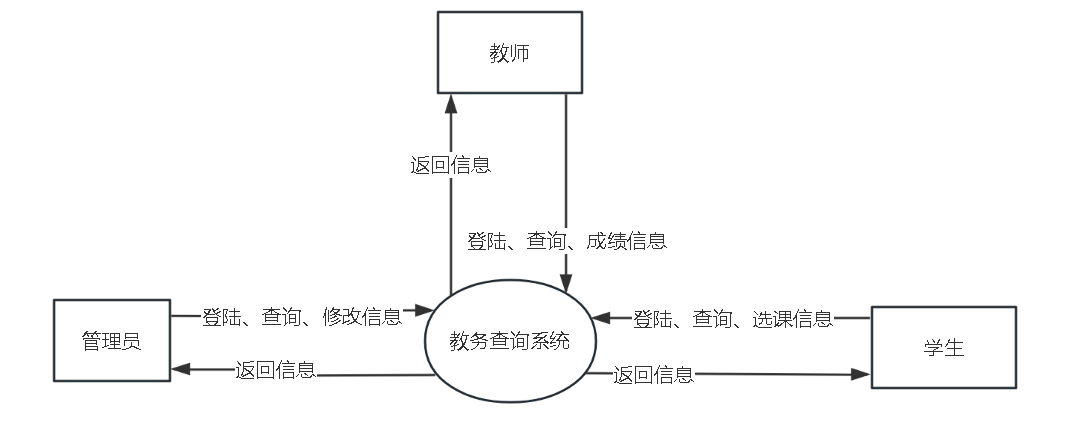


图2: 顶层数据流图

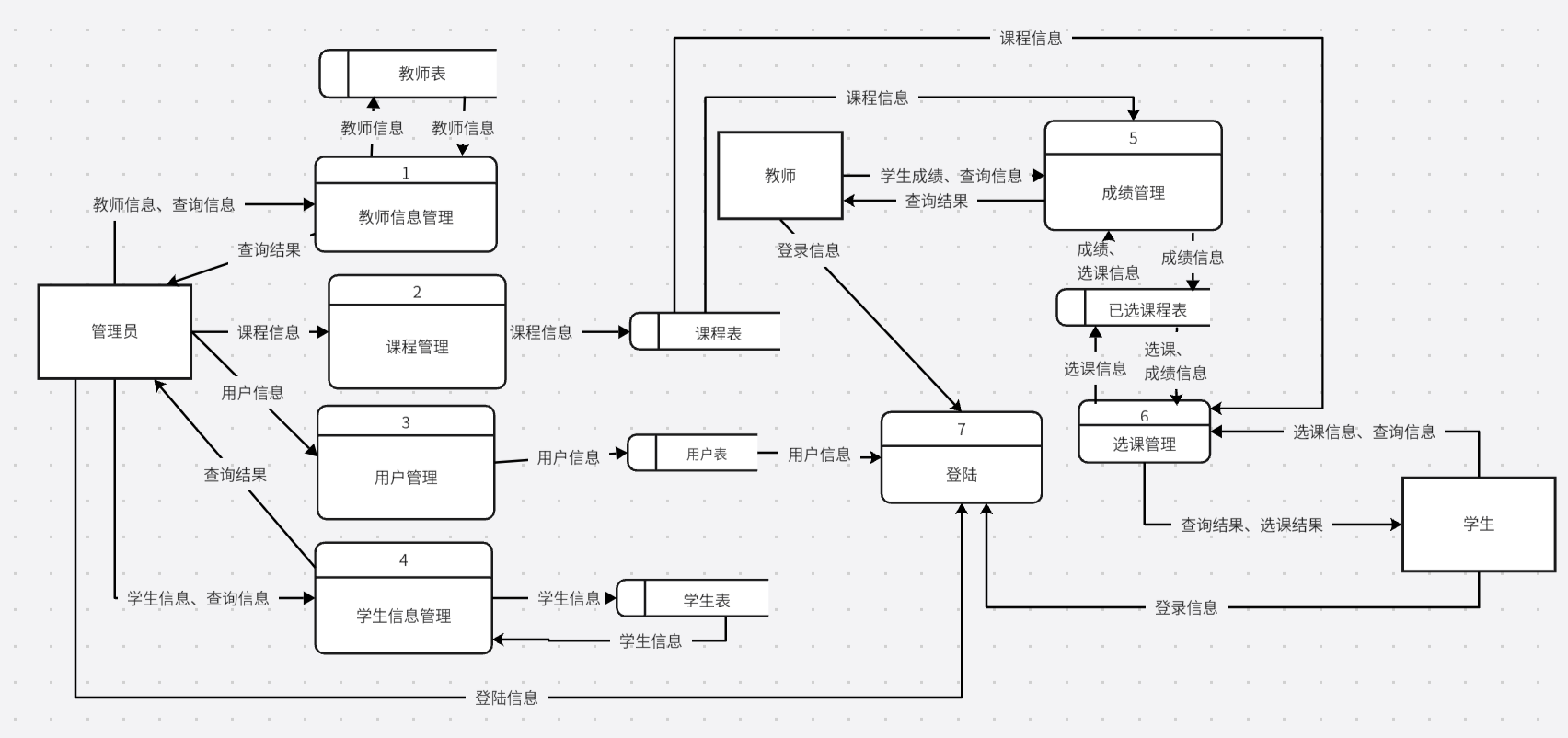


图3: 1层数据流图

**5.数据字典**

**5.1数据项**

教师 ID、姓名、性别、出生日期、职称、入职日期、学生 ID、出生日期、入学时间、院系 ID、院系名、课程 ID、课程名称、上课时间、上课地点、学分、课程 ID、成绩分数、用户 ID、密码、权限ID

**5.2数据流**

教师信息 = 教师 ID、姓名、性别、出生日期、职称、入职日期、院系 ID

学生信息 = 学生 ID、姓名、性别、出生日期、入学时间、院系 ID

课程信息 = 课程 ID、课程名称、教师 ID、上课时间、上课地点、学分

选课信息 = 课程 ID、学生 ID、课程 ID

成绩信息 = 学生 ID、课程 ID、成绩分数

登录信息 = 用户 ID、密码

用户信息 = 用户 ID、用户类型

**5.3数据流描述：**

ID: 1.1

数据流名: 教师信息

说明: 管理员对教师信息进行管理时涉及的教师详细信息

数据流来源: 管理员（外部实体）

数据流去向: 教师信息管理（过程）

组成: 教师 ID、姓名、性别、出生日期、职称、入职日期、院系 ID

ID: 1.2

数据流名: 学生信息

说明: 管理员对学生信息进行管理时涉及的学生详细信息

数据流来源: 管理员（外部实体）

数据流去向: 学生信息管理（过程）

组成: 学生 ID、姓名、性别、出生日期、入学时间、院系 ID

ID: 1.3

数据流名: 课程信息

说明: 管理员对课程信息进行管理时涉及的课程详细信息

数据流来源: 管理员（外部实体）

数据流去向: 课程管理（过程）

组成: 课程 ID、课程名称、授课教师 ID、上课时间、上课地点、学分

ID: 1.4

数据流名: 选课信息

说明: 学生选课操作产生的已选课程相关信息

数据流来源: 学生（外部实体）

数据流去向: 选课管理（过程）

组成:课程 ID、学生 ID、课程 ID

ID: 1.5

数据流名: 成绩信息

说明: 教师录入学生成绩时产生的成绩相关信息

数据流来源: 教师（外部实体）

数据流去向: 成绩管理（过程）

组成: 学生 ID、课程 ID、成绩分数

ID: 1.6

数据流名: 登录信息

说明: 用户登录系统时输入的账号和密码信息

数据流来源: 教师 / 学生 / 管理员（外部实体）

数据流去向: 登录（过程）

组成:用户 ID、密码

ID: 1.7

数据流名: 用户信息

说明: 管理员对用户信息进行管理时涉及的用户基本信息

数据流来源: 管理员（外部实体）

数据流去向: 用户管理（过程）

组成: 用户 ID、用户类型、密码

**5.4数据存储描述**

ID: 2.1

数据存储名: 教师表

说明: 存储教师的详细信息

流入数据流: 管理员输入的教师信息、教师信息修改后的信息

流出数据流: 教师信息查询结果、课程管理中授课教师相关信息

组成: 教师 ID、姓名、性别、出生日期、职称、入职日期、院系 ID

ID: 2.2

数据存储名: 学生表

说明: 存储学生的详细信息

流入数据流: 管理员输入的学生信息、学生信息修改后的信息

流出数据流: 学生信息查询结果、选课管理中学生相关信息

组成: 学生 ID、姓名、性别、出生日期、入学时间、院系 ID

ID: 2.3

数据存储名: 课程表

说明: 存储课程的详细信息

流入数据流: 管理员输入的课程信息、课程信息修改后的信息

流出数据流: 课程查询结果、选课管理中课程相关信息、教师查询教授课程相关信息

组成: 课程 ID、课程名称、教师 ID、上课时间、上课地点、学分

ID: 2.4

数据存储名: 已选课程表

说明: 存储学生已选课程的信息以及成绩

流入数据流: 学生选课操作产生的已选课程信息、选课退课操作后的更新信息、教师录入的成绩信息

流出数据流: 学生查询已选课程结果和成绩结果、教师查询成绩相关信息

组成:学生 ID、课程 ID、成绩分数

ID: 2.5

数据存储名: 用户表

说明: 后台存储的用户登录系统的账号密码和权限信息

数据流来源: : 管理员输入的用户信息、用户信息修改后的信息

流出数据流: 用户信息查询结果、登录验证相关信息

组成:用户 ID、密码、权限ID

**5.5过程描述：**

(1)过程名: 教师信息管理

简要说明: 管理员对教师信息进行添加、修改、删除和查询操作

输入（数据流）: 管理员输入的教师信息、教师信息查询条件

输出（数据流）: 更新后的教师信息、教师信息查询结果

功能: 实现对教师信息的全面管理，确保教师信息的准确性和完整性

处理逻辑：

a.添加信息：系统接收管理员输入的教师信息，首先检查教师 ID 是否已存在于教师表中，若存在则提示管理员 “教师 ID 重复，请重新输入”，并拒绝添加操作；否则将教师信息插入教师表中，并更新教师信息列表

b.修改信息：系统根据教师 ID 在教师表中查找对应记录，并更新表中数据为修改后的信息。

c.删除信息：系统根据教师 ID在教师表中查找对应的记录并删除，同时更新教师信息查询结果，确保查询时不再显示已删除的教师信息。

d.查询信息：

系统根据查询条件在教师表中进行筛选，将符合条件的教师信息查询结果返回并显示

(2)过程名: 学生信息管理

简要说明: 管理员对学生信息进行添加、修改、删除和查询操作

输入（数据流）: 管理员输入的学生信息、学生信息查询条件

输出（数据流）: 更新后的学生信息、学生信息查询结果

功能: 实现对学生信息的全面管理，确保学生信息的准确性和完整性

处理逻辑：

a.添加信息：系统接收管理员输入的学生信息，首先检查教师 ID 是否已存在于学生表中，若存在则提示管理员 “学生 ID 重复，请重新输入”，并拒绝添加操作；否则将学生信息插入教师表中，并更新教师信息列表

b.修改信息：系统根据学生 ID 在学生表中查找对应记录，并更新表中数据为修改后的信息。

c.删除信息：系统根据学生 ID在学生表中查找对应的记录并删除，同时更新学生信息查询结果，确保查询时不再显示已删除的教师信息。

d.查询信息：

系统根据查询条件在学生表中进行筛选，将符合条件的信息查询结果返回并显示

(3)过程名: 课程管理

简要说明: 管理员对课程信息进行添加、修改、删除和查询操作

输入（数据流）: 管理员输入的课程信息、课程信息查询条件

输出（数据流）: 更新后的课程信息、课程信息查询结果

功能: 实现对课程信息的全面管理，确保课程信息的准确性和及时性，满足教学安排需求

处理逻辑：

a.添加信息：系统接收管理员输入的课程信息,检查课程 ID 是否重复，若重复提示 “课程 ID 已存在，请重新输入”，否则将课程信息插入课程表，更新课程信息查询结果并打印显示。

b.修改信息：系统根据课程 ID 在课程表中查找记录，并更新为修改后的数据，刷新课程信息查询结果并显示。

c.删除信息：系统根据课程 ID 在选课表中查找记录，若该课程有学生已选，提示管理员 “该课程有学生已选，是否继续删除？”，若管理员确认删除则再查询课程表找到对应记录执行删除操作，并在已选课程表中删除与该课程相关的学生选课记录。否则不删除，返回原始数据

d.查询信息：统根据查询条件在课程表中进行筛选，将符合条件的信息查询结果返回并显示

(4)过程名: 选课管理

简要说明: 学生进行选课、退选课程操作，系统处理选课相关逻辑并更新数据

输入（数据流）: 学生的选课操作信息、课程信息、学生已选课程信息

输出（数据流）: 选课结果、更新后的已选课程信息

功能: 处理学生的选课和退课请求，更新学生已选课程记录和课程的选课人数等信息，确保选课和退课结果准确反映在系统中

处理逻辑：

a.选课操作：系统接收学生的选课请求，在已选课程表中插入一条学生 ID 和课程 ID 对应的记录，并更新显示学生的已选课程信息查询结果。

b.退课操作：

系统接收学生的退课操作请求，在已选课程表中根据学生 ID 和课程 ID查找对应的记录并删除，更新学生的已选课程信息查询结果并显示。

(5)过程名: 成绩管理

简要说明: 教师录入学生成绩，系统存储并管理成绩信息

输入（数据流）: 教师录入的成绩信息、学生信息、课程信息

输出（数据流）: 成绩录入结果

功能: 为教师提供成绩录入平台，存储成绩信息，方便学生和教师查询成绩，确保成绩数据的准确性和安全性

处理逻辑：

a成绩录入：教师输入成绩并提交，系统接收到打分请求，先通过学生ID和课程ID在选课表中查找对应记录，在该记录成绩字段添加接收到的分数，同时更新成绩查询结果并显示

(6)过程名: 用户管理

简要说明: 管理员对用户信息进行管理，重置账号密码

输入（数据流）: 管理员输入的用户信息

输出（数据流）: 更新后的用户信息

功能: 保障用户信息的准确性和完整性，确保只有授权人员能够进行用户管理操作，帮助忘记密码的用户重置账号

处理逻辑：管理员输入要修改的用户 ID 及新的用户信息。系统依据用户 ID 在用户表中查找记录并更新用户表中对应信息

(7)过程名: 登录

简要说明: 用户输入登录信息，系统进行验证并允许用户进入系统

输入（数据流）: 用户输入的登录信息（用户 ID、密码）

输出（数据流）: 登录成功或失败的结果、用户角色信息（教师 / 学生 / 管理员）

功能: 验证用户身份，根据用户类型引导用户进入相应的系统界面，保障系统的安全性和访问权限控制

处理逻辑：系统接收用户输入的登录信息，包括用户 ID 和密码。首先在用户表中查找该用户 ID 对应的记录，若不存在则提示 “用户 ID 不存在，请核实后重新输入”。若用户 ID 存在，则检查输入的密码与用户表中存储的密码是否一致，若不一致则提示 “密码错误，请重新输入”，若一致则判断用户类型（教师 / 学生 / 管理员），根据用户类型跳转到相应的系统界面。

## 三、数据库设计

数据库类型：mysql

数据库服务器地址：192.168.56.1

数据库名称：EducationSystem

数据库用户名和密码：root 123456

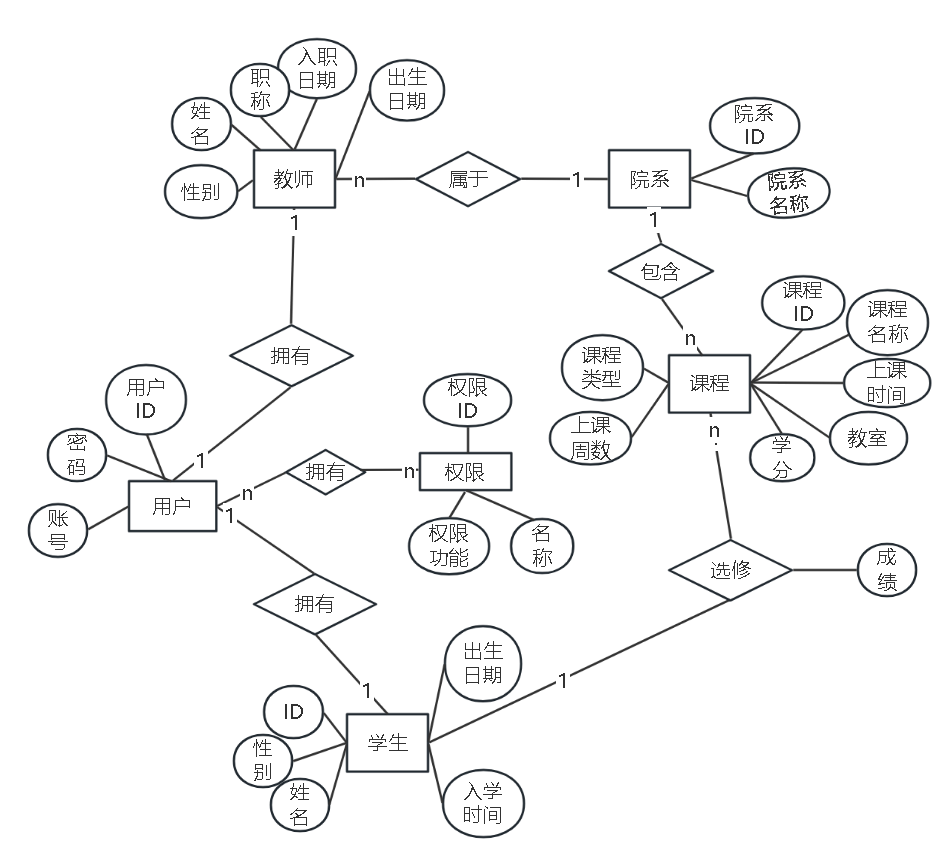


图4：实体联系ER图

**3.1根据上述章节系统功能分析和实体联系图可设计出如下关系模型：**

1、院系表（college）：院系ID、院系名称

2、课程表（couese）：课程ID、课程名称、教师ID、上课时间、上课教室、上课周数、课程类型、所属院系ID、学分

3、权限表（role）：权限ID、权限名称、权限

4、所选课程表（selectedcourse）：课程表ID、学生ID、成绩

5、学生表（student）：用户ID、姓名、性别、出生日期、入学时间、所属院系

6、教师表（teacher）：用户ID、姓名、性别、出生日期、学历、职称/职位、入职日期、所属院系

7、用户表(userlogin)：用户ID、用户账号、用户密码、权限系统功能

**3.2优化分析：**

1.设置外键关联，避免在多个表中重复存储相同的信息。例如：院系信息只在院系表中详细存储，课程表通过所属院系 ID 引用，减少了院系名称等信息在课程相关表中的冗余存储。

2.教师和学生的基本用户信息统一通过用户 ID 关联，避免在不同功能相关表中重复存储姓名等通用信息。

3.外键约束保证了关联数据的一致性，课程表中的教师 ID 必须对应教师表中已存在的教师，防止出现无效的教师关联情况；所选课程表中的学生 ID 和课程表 ID 分别关联学生表和课程表，保证选课记录的合法性和数据一致性。权限表通过权限 ID 关联用户登录表，统一管理用户权限，当权限规则变更时，只需在权限表中修改，就能保证所有关联用户权限的一致性。

4. 在插入课程信息时，由于所属院系 ID 有外键约束，必须先存在对应的院系记录，避免了无对应院系的孤立课程记录插入，保证数据完整性

**3.3优化分析后设计的数据库中实际关系数据模型：**

**1. college（院系表）**

|  |  |  |
| --- | --- | --- |
| **字段** | **数据类型** | **说明** |
| collegeID | int(11) | 主键，非空、院系ID |
| collegeName | varchar(200) | 非空、院系名称 |

**2. course（课程表）**

|  |  |  |
| --- | --- | --- |
| **字段** | **数据类型** | **说明** |
| courseID | int(11) | 主键，非空、课程ID |
| courseName | varchar(200) | 非空、课程名称 |
| teacherID | int(11) | 非空、外键，教师ID（teacher.userID） |
| courseTime | varchar(200) | 开课时间 |
| classRoom | varchar(200) | 开课地点 |
| courseWeek | int(200) | 学时 |
| courseType | varchar(20) | 课程类型 |
| collegeID | int(11) | 外键，院系ID（college.collegeID） |
| score | int(11) | 非空、学分 |

**外键关系：**collegeID -> college.collegeID：表示课程所属院系。

teacherID -> teacher.userID：表示课程的授课教师。

**3. role（角色表）**

|  |  |  |
| --- | --- | --- |
| **字段** | **数据类型** | **说明** |
| roleID | int(11) | 主键，非空、角色ID |
| roleName | varchar(20) | 非空、角色名称（如：admin, teacher, student） |
| permissions | varchar(255) | 权限描述 |

**4. selectedcourse（选课表）**

|  |  |  |
| --- | --- | --- |
| **字段** | **数据类型** | **说明** |
| courseID | int(11) | 外键，非空、课程ID（course.courseID） |
| studentID | int(11) | 非空、外键，学生ID（student.userID） |
| mark | int(11) | 成绩（可为空） |

**外键关系：**

* courseID -> course.courseID：表示学生所选课程。
* studentID -> student.userID：表示选课的学生。

**5. student（学生表）**

|  |  |  |
| --- | --- | --- |
| **字段** | **数据类型** | **说明** |
| userID | int(11) | 非空、主键，自增、学生ID |
| userName | varchar(200) | 非空、学生姓名 |
| sex | varchar(20) | 性别 |
| birthYear | date | 出生日期 |
| grade | date | 入学时间 |
| collegeID | int(11) | 外键，院系ID（college.collegeID） |

**外键关系：**

* collegeID -> college.collegeID：表示学生所属院系。

**6. teacher（教师表）**

|  |  |  |
| --- | --- | --- |
| **字段** | **数据类型** | **说明** |
| userID | int(11) | 非空、主键，自增、教师ID |
| userName | varchar(200) | 非空、教师姓名 |
| sex | varchar(20) | 性别 |
| birthYear | date | 出生日期 |
| degree | varchar(20) | 学历 |
| title | varchar(255) | 职称 |
| grade | date | 入职时间 |
| collegeID | int(11) | 外键，非空、院系ID（college.collegeID） |

**外键关系：**

* collegeID -> college.collegeID：表示教师所属院系。

**7. userlogin（用户登录表）**

|  |  |  |
| --- | --- | --- |
| **字段** | **数据类型** | **说明** |
| userID | int(11) | 主键，自增、非空、用户ID（student.userID 或 teacher.userID） |
| userName | varchar(200) | 非空、用户名 |
| password | varchar(200) | 非空、密码 |
| role | int(11) | 外键，非空、默认2、角色ID（role.roleID） |

**外键关系：**

* role -> role.roleID：表示用户角色（admin、teacher、student）。

关系说明：

college表与course表通过collegeID建立一对多关系，一个院系可以有多门课程。

course表与teacher表通过teacherID建立多对一关系，多门课程可以对应一个教师。

course表与selectedcourse表通过courseID建立一对多关系，一门课程可以被多个学生选修。

student表与selectedcourse表通过studentID建立一对多关系，一个学生可以选修多门课程。

userlogin表与student表、teacher表通过userID建立一对一关系，每个用户（学生或教师）在userlogin表中有一条对应的登录信息记录。

userlogin表与role表通过role建立多对一关系，多个用户可以属于同一个角色。

## 四、概要设计

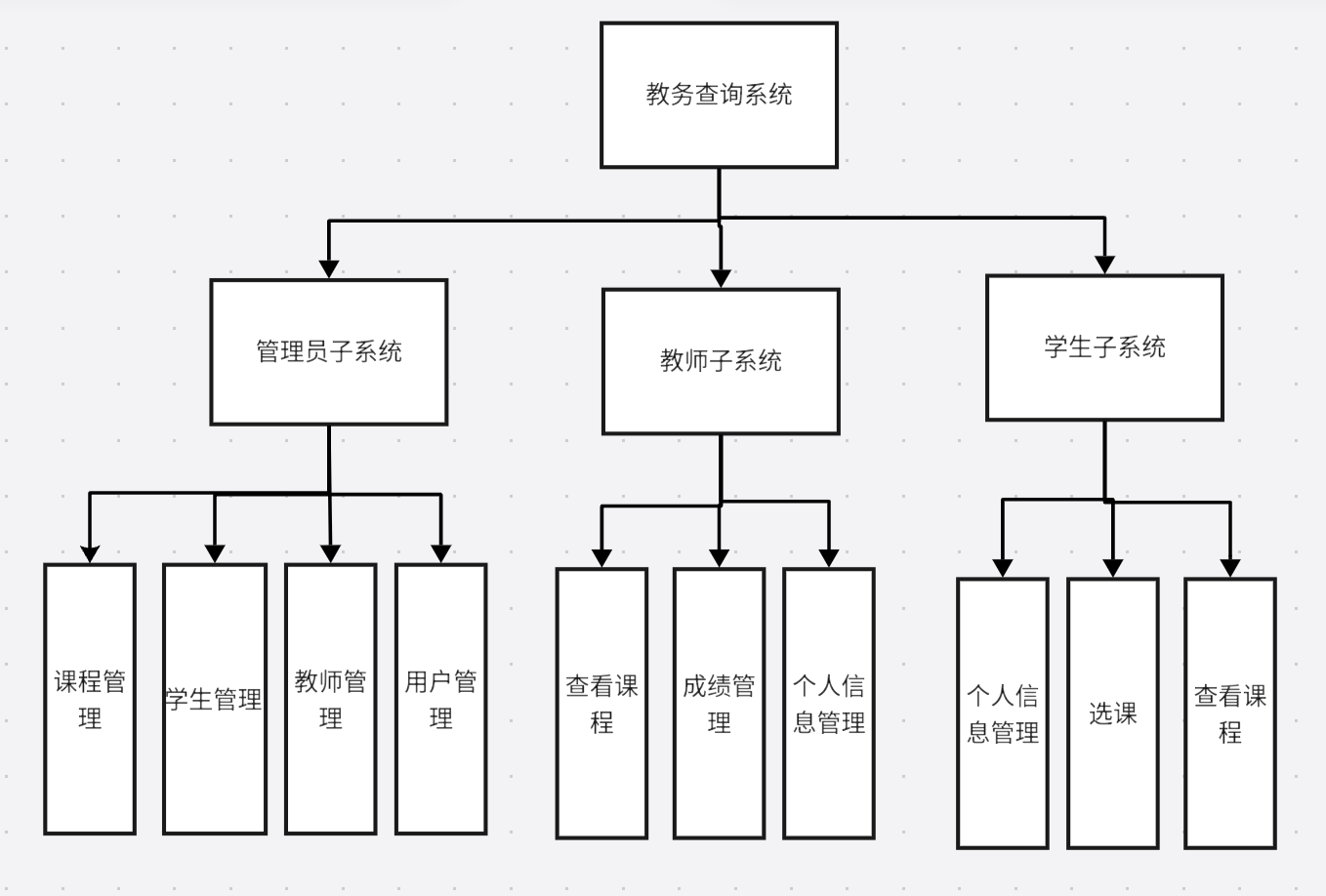


图5：系统结构图

### 4.1模块功能描述：

• 管理员模块

• 课程管理：增删改查课程。

• 学生管理：增删改查学生信息

• 教师管理：增删改查教师信息

• 用户管理：管理用户角色和权限，重置用户账号密码

• 教师模块

• 课程管理：查看自己教授的课程信息。

• 成绩管理：查看选修自己课程的学生，给学生打分。

• 个人信息管理：修改个人信息和密码。

• 学生模块

• 选课：学生可以对课程进行选课或者退课

• 课程查看：查看已选课程信息、已修课程信息和成绩。

• 个人信息管理：修改个人信息和密码。

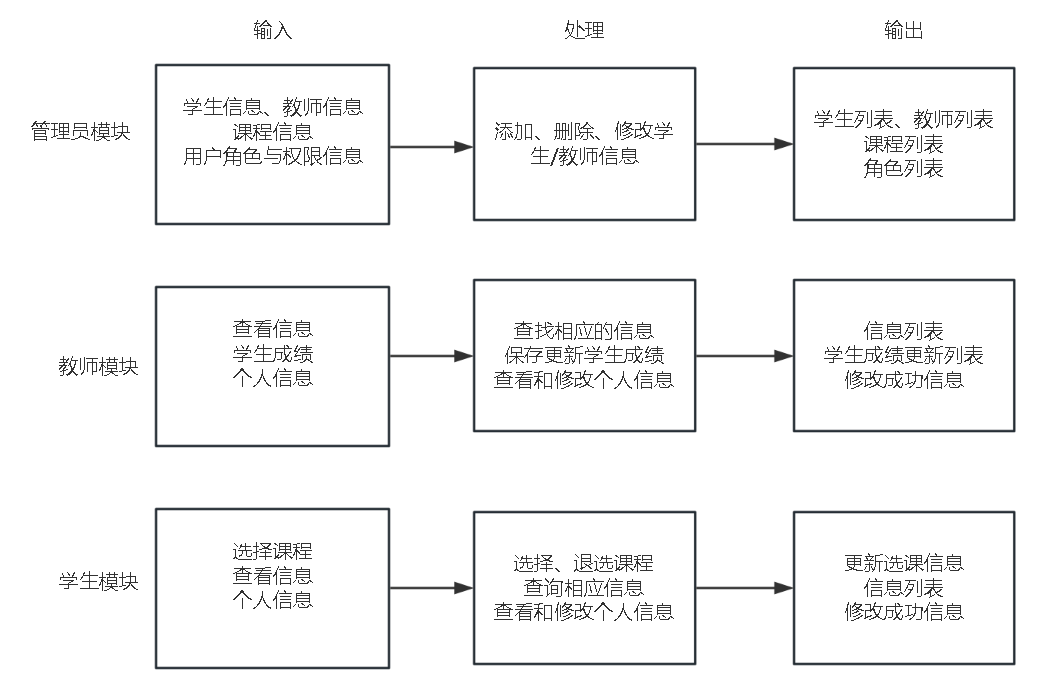


图6：模块IPO图

### 4.2系统结构设计

**三层架构**

**表示层（Presentation Layer）**：主要由前端页面组成，webapp目录下的JSP index.jsp、JSP login.jsp以及css、fonts、images、js等资源文件。该层负责与用户进行交互，展示数据和接收用户输入，将用户请求发送到业务逻辑层，并将业务逻辑层返回的结果展示给用户。例如，学生在前端页面进行选课、查询课程信息等操作，这些操作请求会被传递到后端进行处理。

**业务逻辑层（Business Logic Layer）**：对应java目录下的service包以及部分controller包的功能。该层处理系统的核心业务逻辑，它接收表示层传递的请求，调用数据访问层的方法获取或操作数据，并进行业务规则的校验和处理。

**数据访问层（Data Access Layer）**：主要包括mapper包以及与数据库交互相关的部分。该层负责与数据库进行通信，执行具体的数据库操作，如查询、插入、更新、删除等。它将业务逻辑层的请求转化为对数据库的操作，并将数据库返回的结果传递给业务逻辑层。

表示层通过 HTTP 请求与业务逻辑层的controller进行交互，controller接收请求后，调用service层的业务方法进行处理。

### 4.3数据结构设计：

实体类

College：对应college表，包含collegeID（院系 ID）和collegeName（院系名称）属性，以及相应的 getter 和 setter 方法。

Course：对应course表，包含courseID（课程 ID）、courseName（课程名称）、teacherID（教师 ID）、courseTime（开课时间）、classRoom（开课地点）、courseWeek（学时）、courseType（课程类型）、collegeID（院系 ID）、score（学分）等属性，以及相应的 getter 和 setter 方法。

Role：对应role表，包含roleID（角色 ID）、roleName（角色名称）、permissions（权限描述）等属性，以及相应的 getter 和 setter 方法。

SelectedCourse：对应selectedcourse表，包含courseID（课程 ID）、studentID（学生 ID）、mark（成绩）等属性，以及相应的 getter 和 setter 方法。

Student：对应student表，包含userID（学生 ID）、userName（学生姓名）、sex（性别）、birthYear（出生日期）、grade（入学时间）、collegeID（院系 ID）等属性，以及相应的 getter 和 setter 方法。

Teacher：对应teacher表，包含userID（教师 ID）、userName（教师姓名）、sex（性别）、birthYear（出生日期）、degree（学历）、title（职称）、grade（入职时间）、collegeID（院系 ID）等属性，以及相应的 getter 和 setter 方法。

UserLogin：对应userlogin表，包含userID（用户 ID）、userName（用户名）、password（密码）、role（角色 ID）等属性，以及相应的 getter 和 setter 方法。

## 五、详细设计

### 5.1项目结构：

po:存放的是封装的实体类，用于表示数据库表对应的 Java 实体类。

Controller：控制层，包含处理HTTP请求的控制器类，它会接收来自前端的请求，并根据请求的类型和路径将其转发到相应的服务层（service）方法。在服务层处理完业务逻辑后，控制器再将结果返回给前端页面，起到了前后端交互的桥梁作用。

Service：存放的是业务逻辑层的服务类，也是一些关于数据库处理的操作，但不是直接操作数据库，这些类处理来自控制器层的请求，调用数据访问层（mapper）进行数据操作，并处理业务逻辑

Mapper：数据访问层接口包，定义数据访问接口，将数据库操作与 Java 对象进行映射

Resources文件下的mapper里的xml存放操作数据库的sql语句

Exception：放的是全局异常处理器和系统自定义异常类，用于处理在应用程序执行过程中可能出现的各种异常情况

Realm：存放登录验证信息

Webapp: web前端页面数据

Application.Properties：Spring Boot 的配置文件，包括服务器端口号、数据库配置信息

表示层通过 HTTP 请求与业务逻辑层的controller进行交互，controller接收请求后，调用service层的业务方法进行处理。service层在处理业务逻辑时，根据需要调用mapper层的方法与数据库进行交互，获取或更新数据。mapper层通过 MyBatis框架与数据库进行映射和操作，将 Java 对象持久化到数据库中。

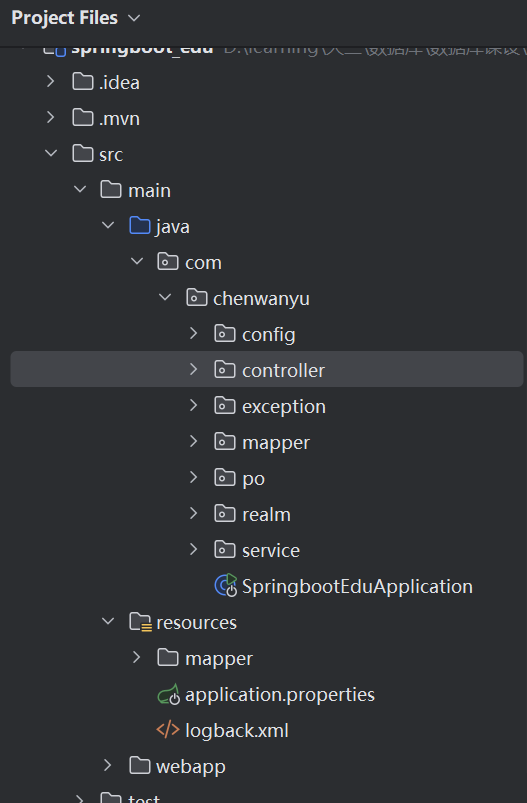
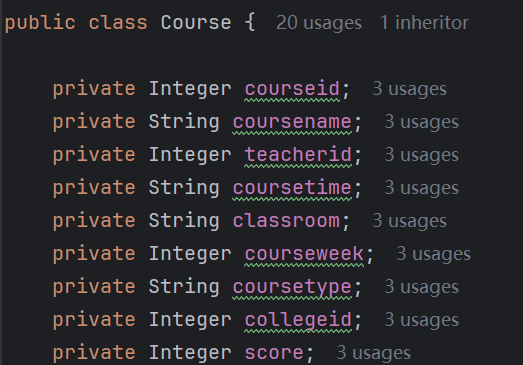
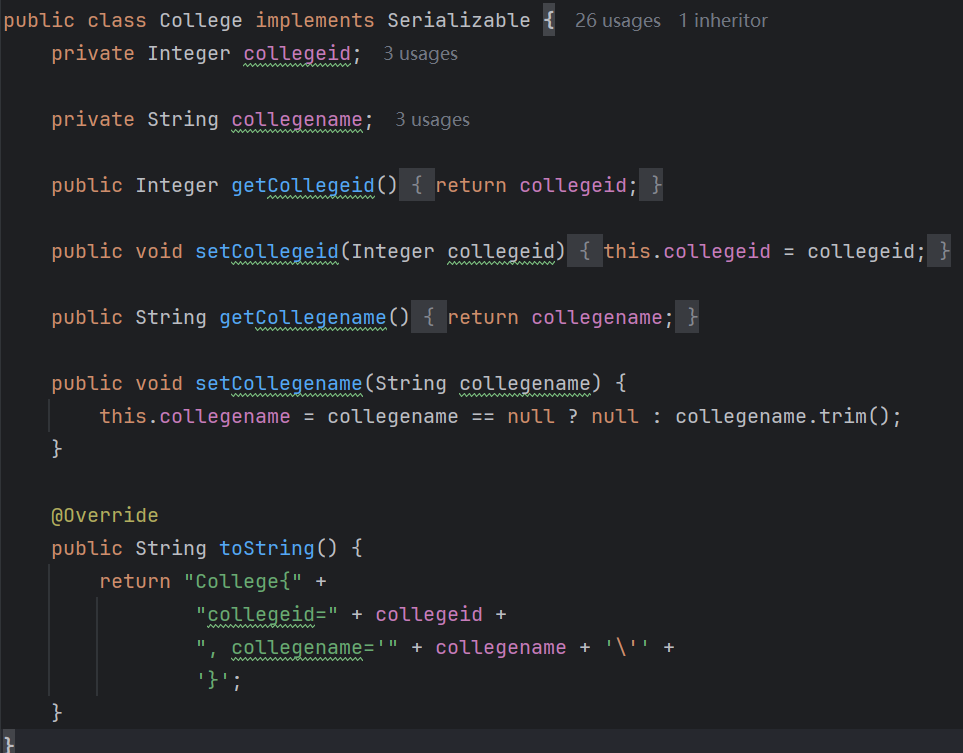
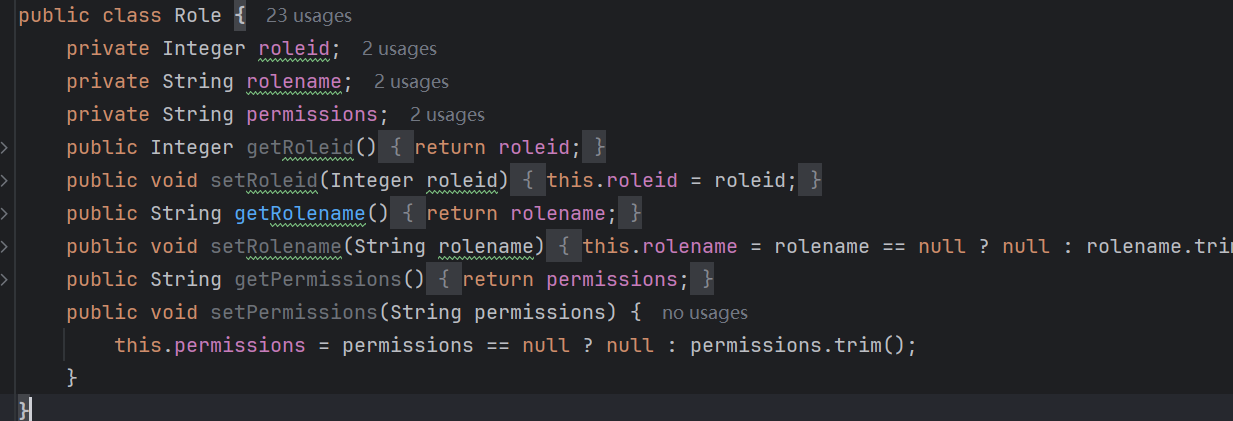


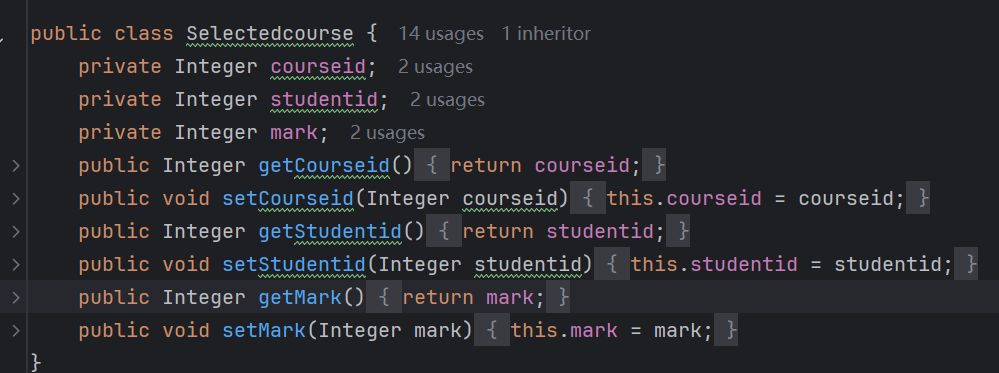
图6：教务查询系统项目文件结构

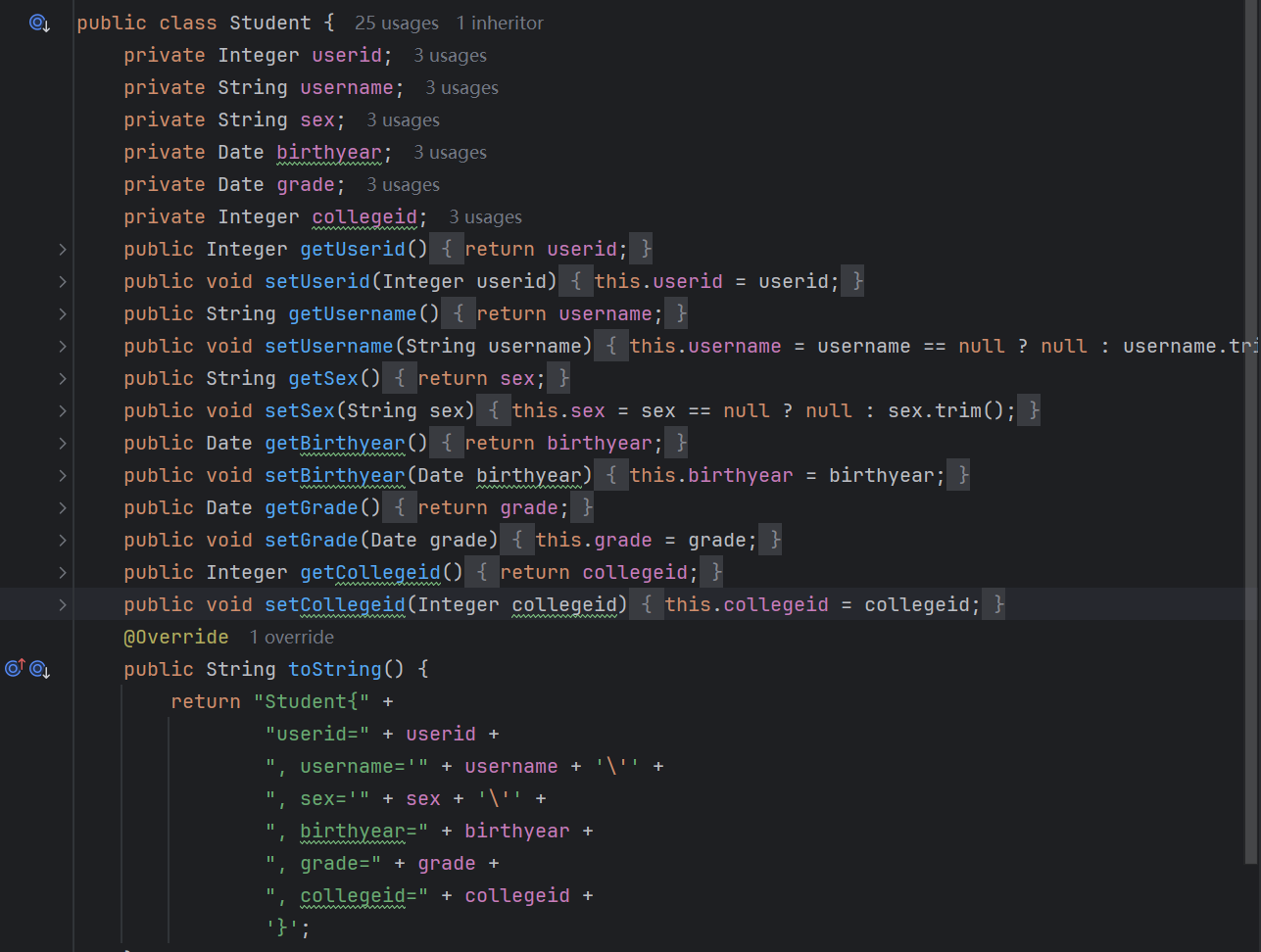
### 5.2数据结构具体实现

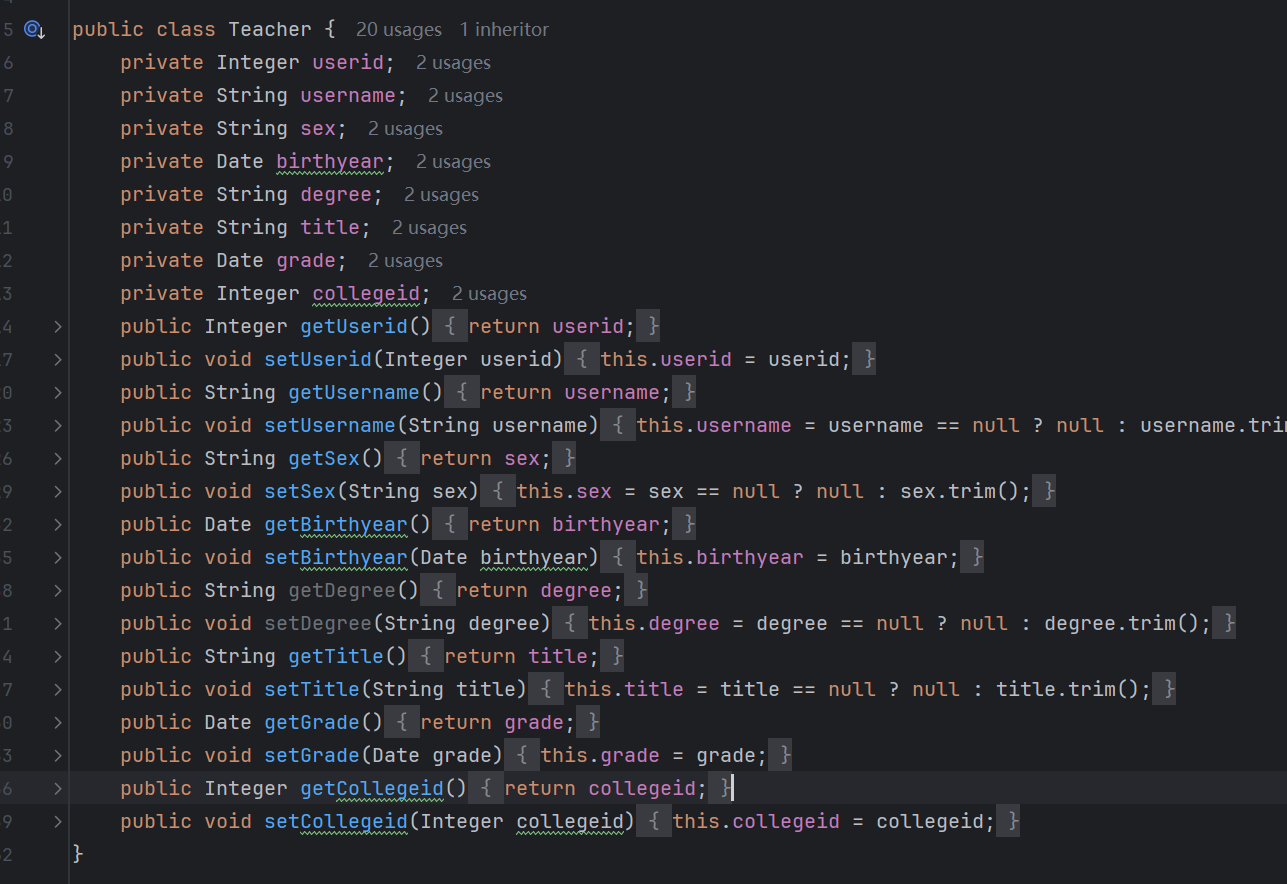
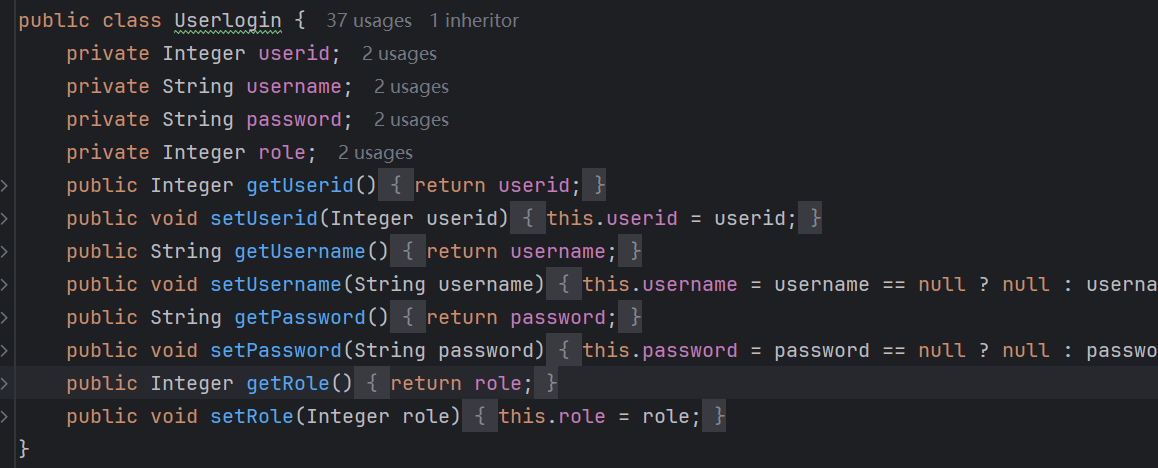








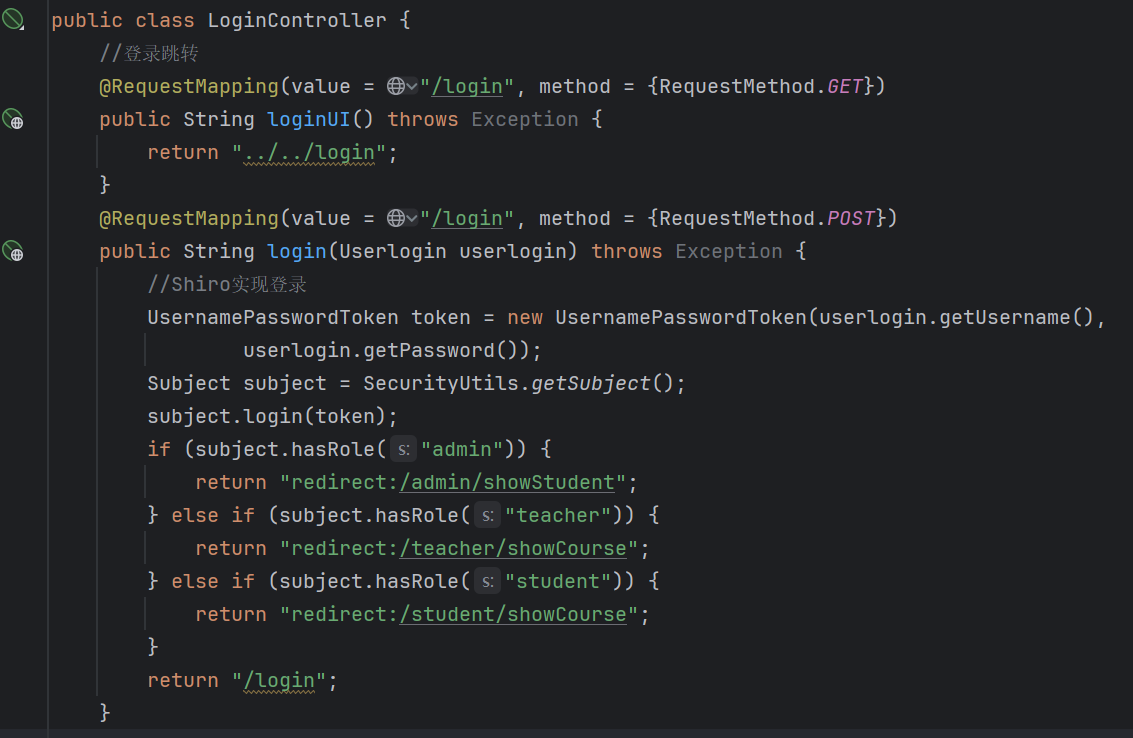




### 5.3模块功能实现

**登录模块：**

主要代码：



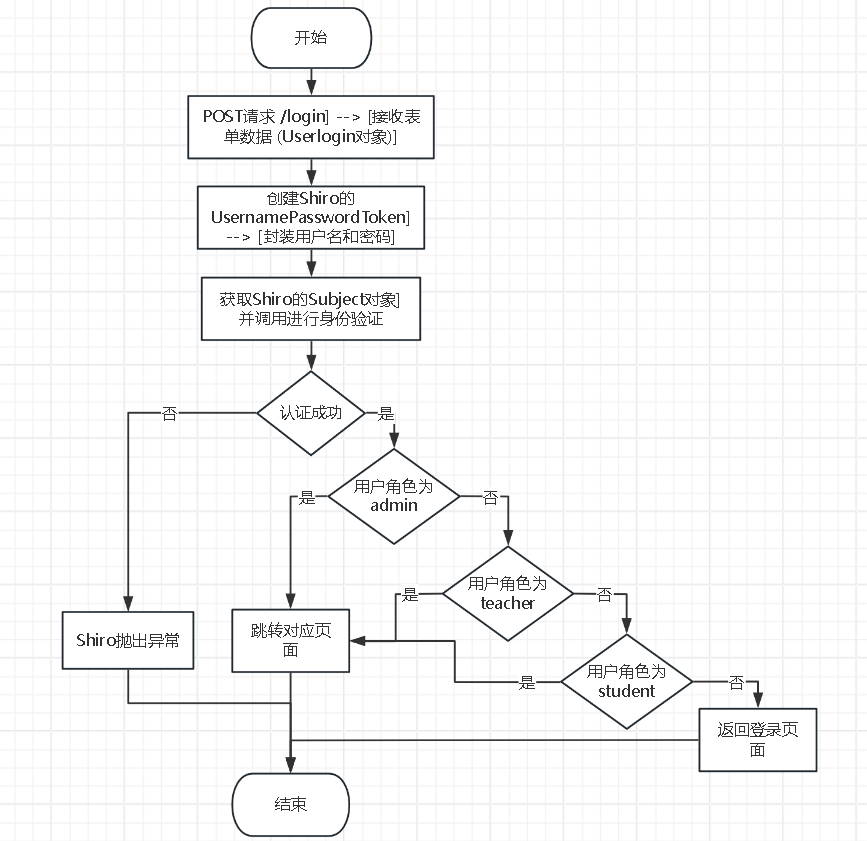
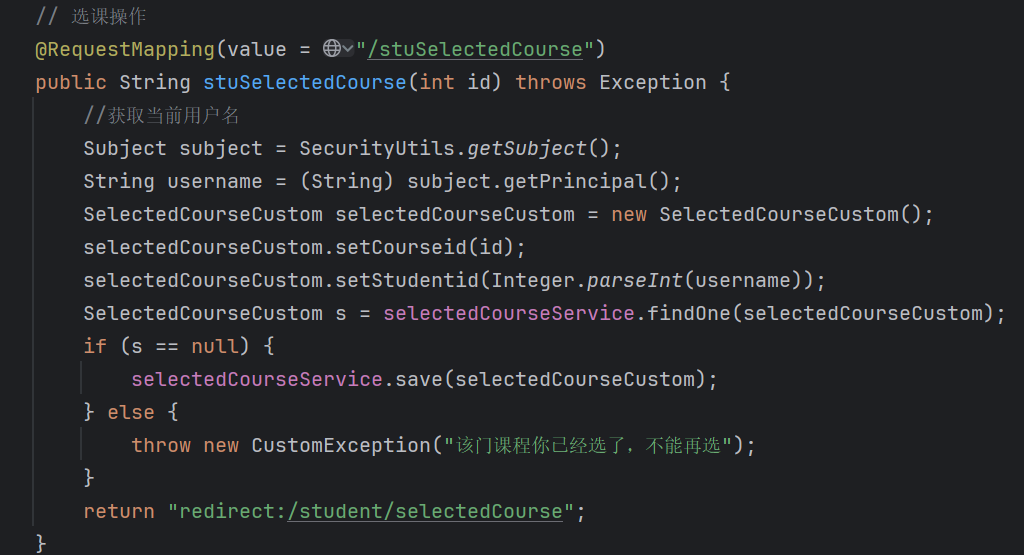


图7：登陆算法流程图

**学生模块：**

选课功能主要代码：



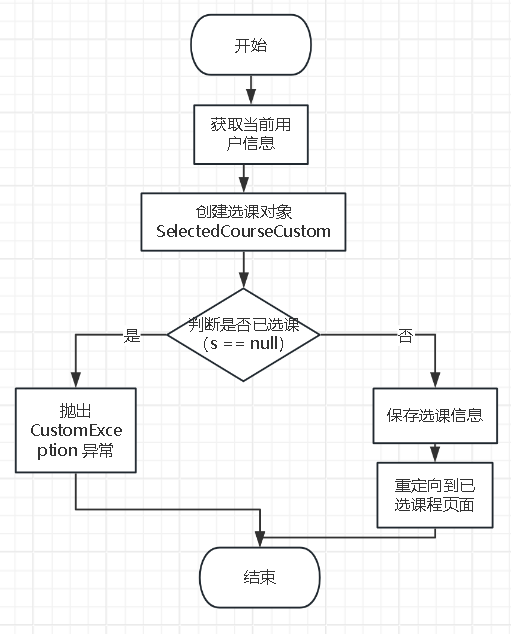
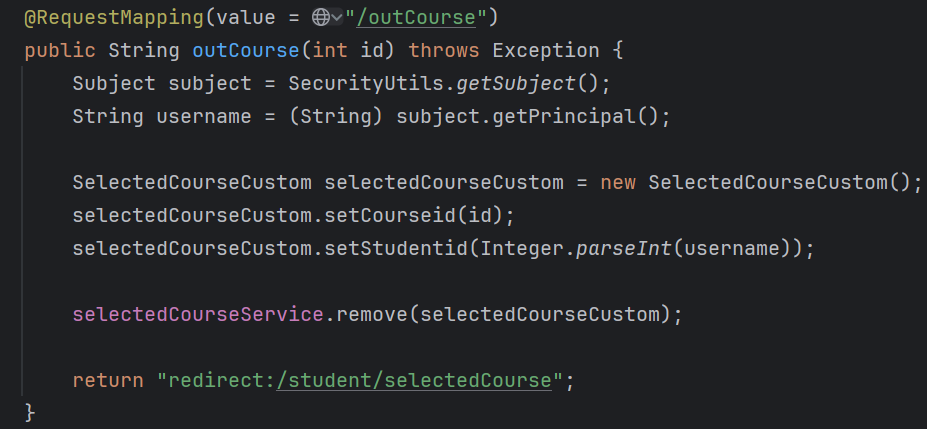


图8：选课算法流程图

退课功能主要代码：



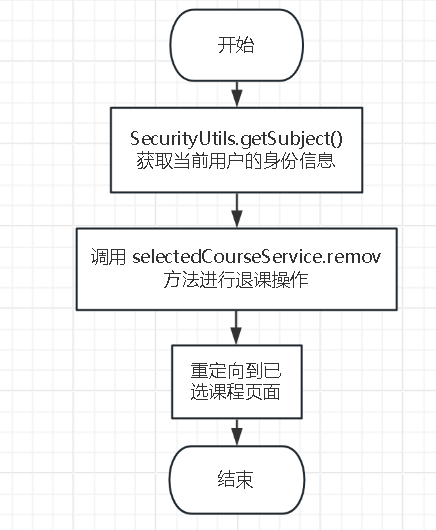
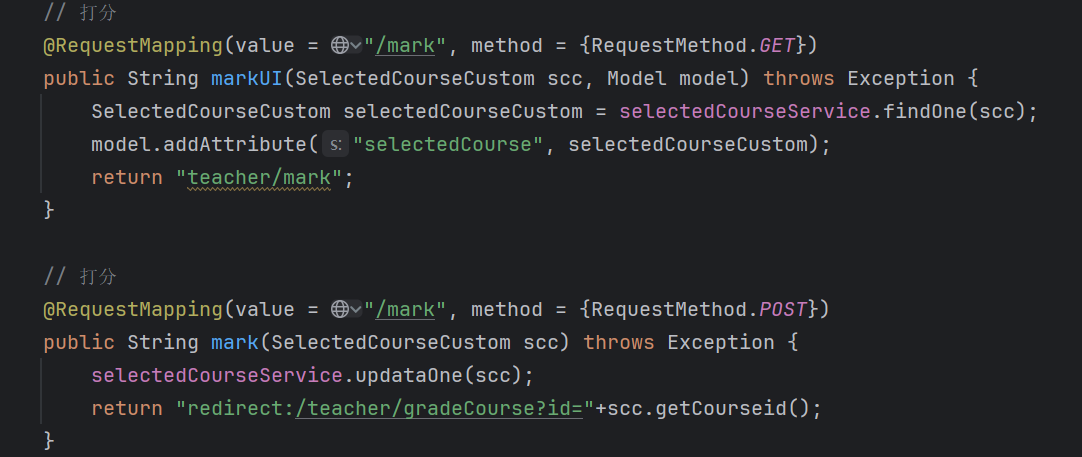


图9：选课算法流程图

**教师模块：**

提交成绩主要代码



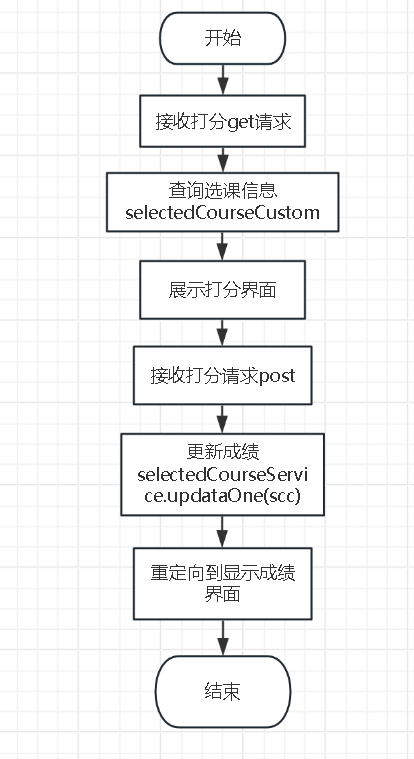


图10：提交成绩算法流程图

查看所授课程功能主要代码



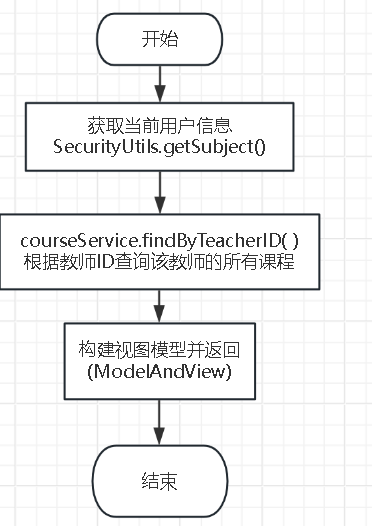
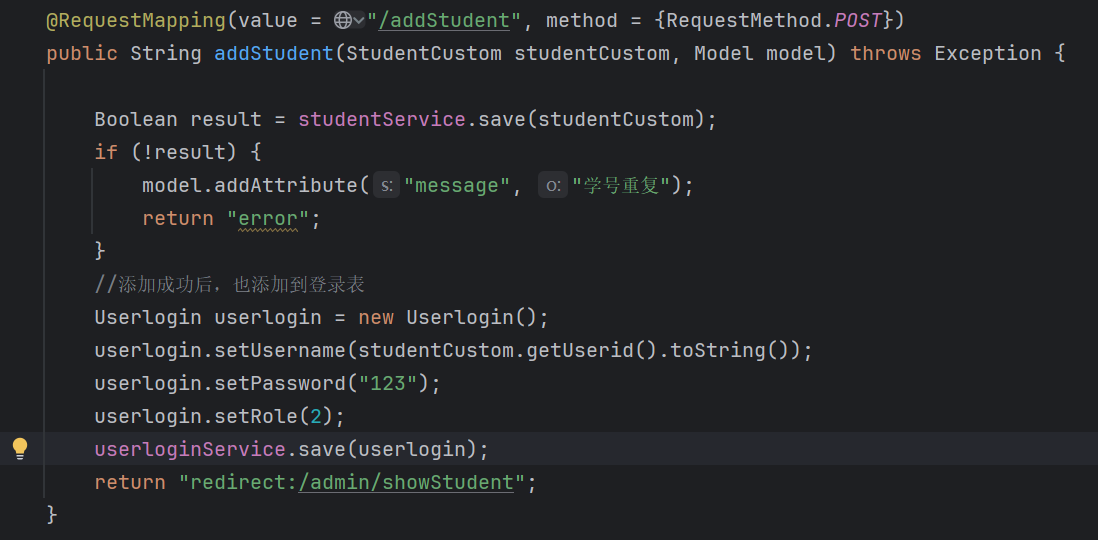


图11：查看授课信息算法流程图

**管理员模块**

添加学生信息主要代码：



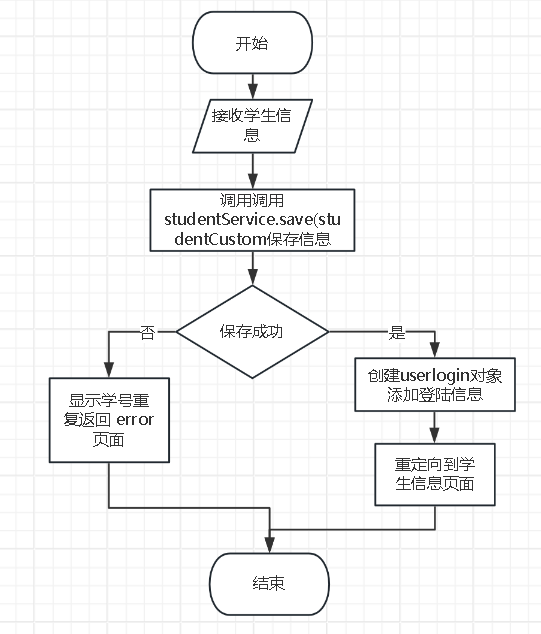
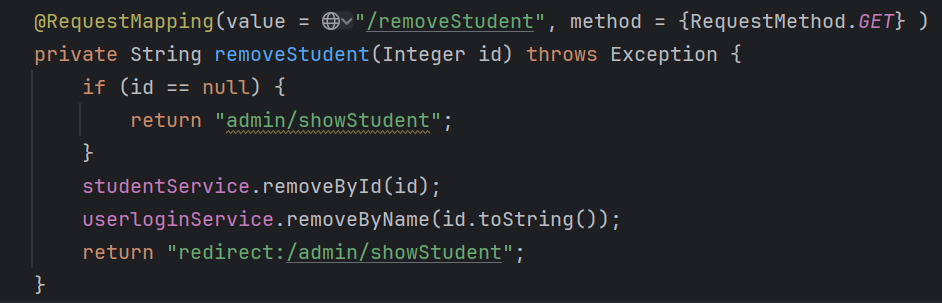


图12：添加学生信息算法流程图

删除学生信息主要代码：



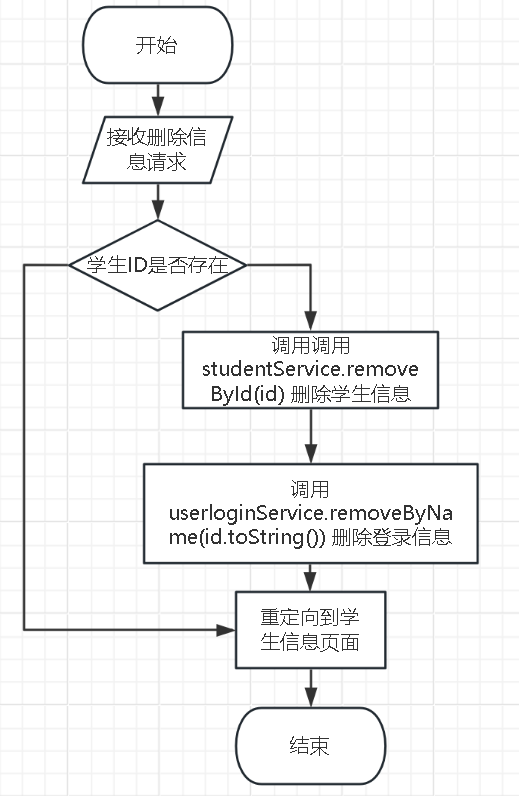
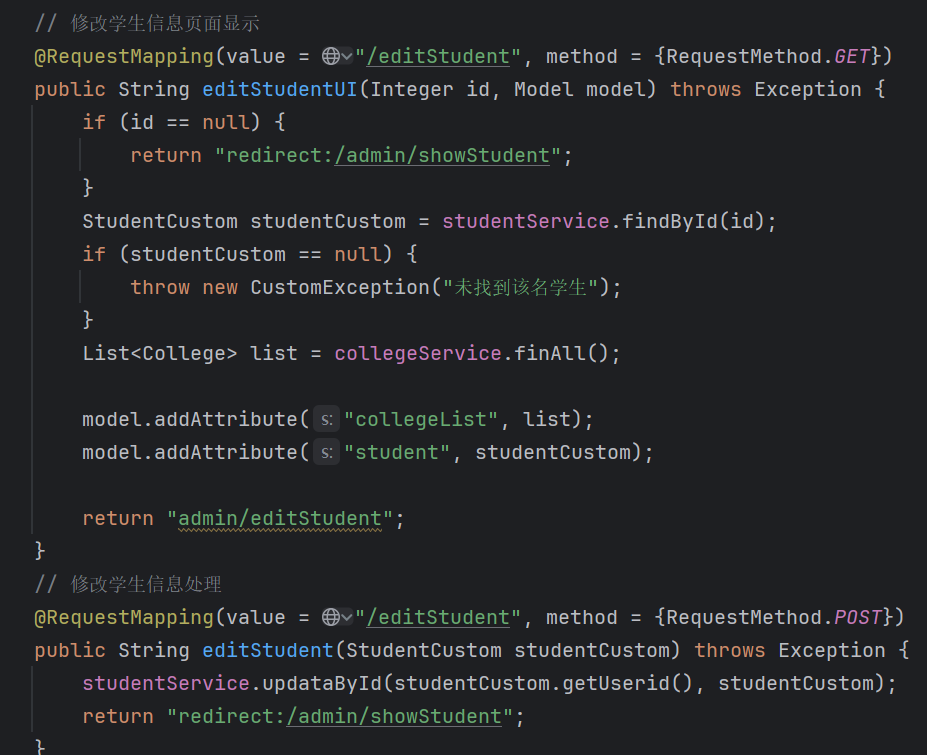


图13：删除学生信息算法流程图

修改学生信息主要代码：



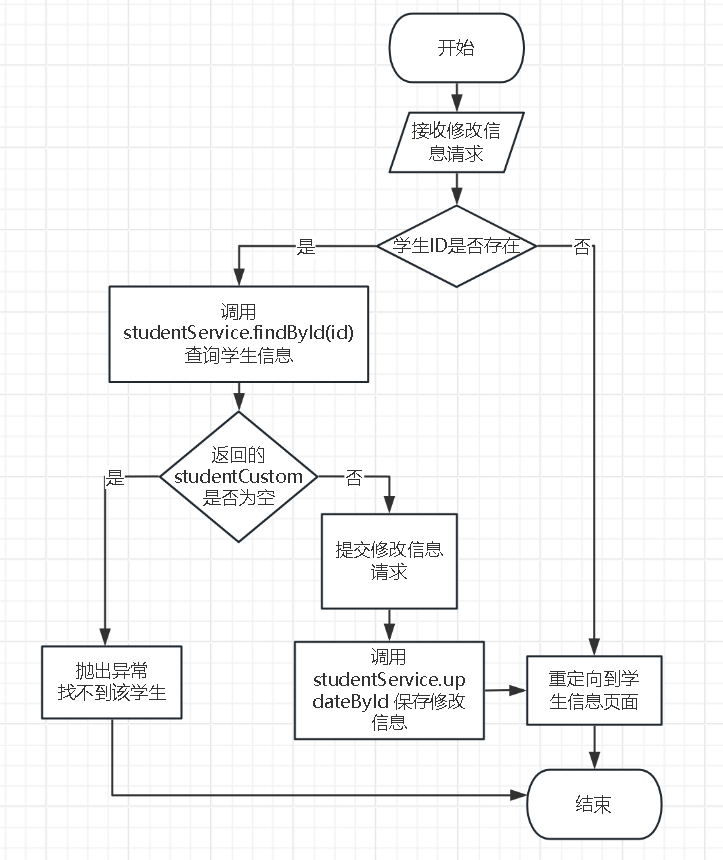
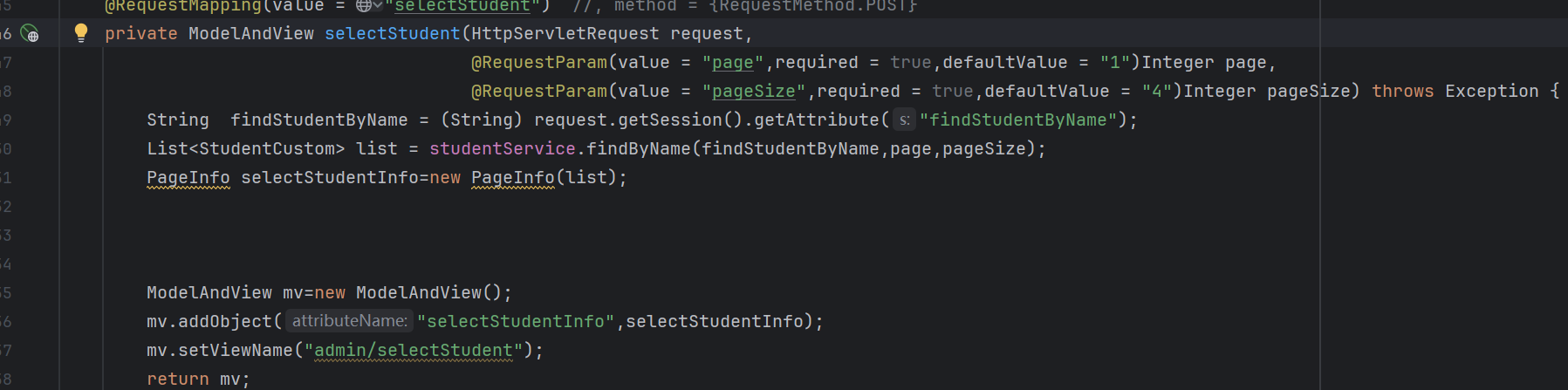


图14：修改信息算法流程图

搜索信息功能主要代码：



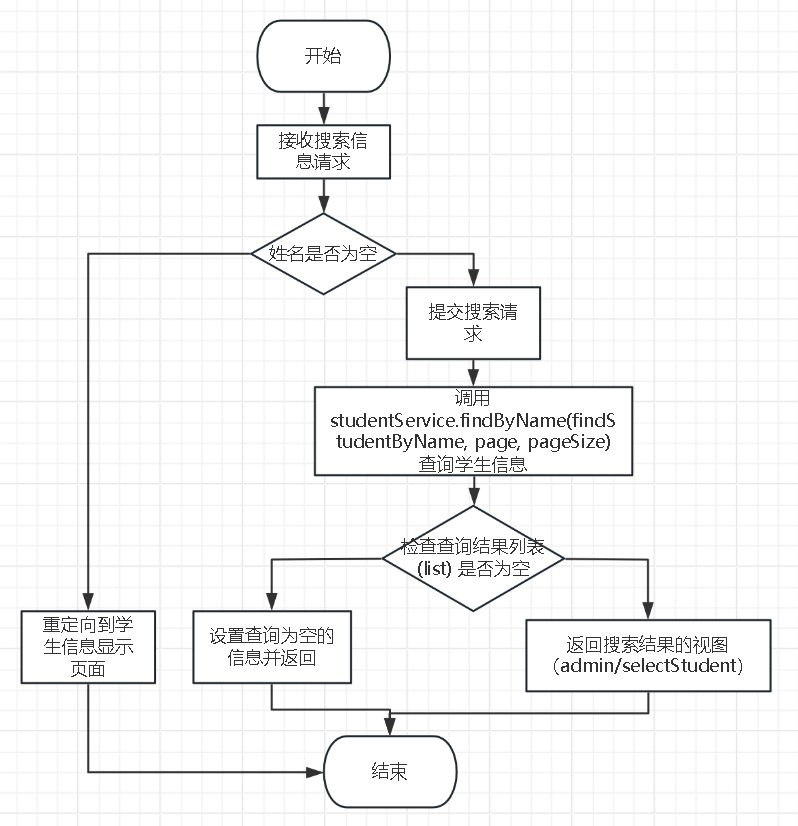


图15：搜索功能算法流程图

## 六、测试与实现

### 测试方案

（一）测试目的

对高校教务查询系统进行全面测试，确保系统功能的完整性、准确性和可靠性，满足教师、学生和管理员三类用户的操作需求。

（二）测试范围

包括管理员、教师和学生的所有功能模块，如信息管理、课程操作、成绩管理、选课操作和密码修改等。

（三）测试策略

a.功能测试

对系统每个功能点进行测试，确保其正常工作。通过场景法设计测试用例，确保测试用例覆盖所有功能模块的操作场景，包括正常操作和异常操作。

b.界面测试

检查系统界面的布局、风格是否符合用户习惯。

验证界面元素（如按钮、文本框等）的显示和操作是否正常。

c.数据完整性测试

检查数据库中数据的准确性和一致性。

验证在进行增删改查操作后，数据是否正确存储和更新。

（四）测试用例设计

1. 管理员功能测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | **测试用例名称** | **测试步骤** | **预期结果** | **实际结果** |
| ADM - 001 | 添加教师信息 | 1. 以管理员身份登录系统 | 教师信息在教师表中正确存储 |  |
| 2. 进入教师管理模块 |
| 3. 输入教师的完整信息，点击添加按钮 |
| ADM - 002 | 修改教师信息 | 1. 以管理员身份登录系统 | 教师表中的对应信息更新 |  |
| 2. 在教师管理模块中查询到要修改的教师 |
| 3. 修改教师信息后点击保存按钮 |
| ADM - 003 | 删除教师信息 | 1. 以管理员身份登录系统 | 教师表中对应记录删除 |  |
| 2. 在教师管理模块中查询到要删除的教师 |
| 3. 点击删除按钮 |
| ADM - 004 | 查询教师信息 | 1. 以管理员身份登录系统 | 系统返回符合条件的教师信息列表 |  |
| 2. 进入教师管理模块 |
| 3. 输入查询条件（如姓名、ID），点击查询按钮 |
| ADM - 005 | 添加学生信息 | 1. 以管理员身份登录系统 | 学生信息在学生表中正确存储 |  |
| 2. 进入学生管理模块 |
| 3. 输入学生的完整信息，点击添加按钮 |
| ADM - 006 | 修改学生信息 | 1. 以管理员身份登录系统 | 学生表中的对应信息更新 |  |
| 2. 在学生管理模块中查询到要修改的学生 |
| 3. 修改学生信息后点击保存按钮 |
| ADM - 007 | 查询学生信息 | 1. 以管理员身份登录系统 | 系统返回符合条件的学生信息列表 |  |
| 2. 进入学生管理模块 |
| 3. 输入查询条件（如姓名、ID），点击查询按钮 |
| ADM - 008 | 重置教师密码 | 1. 以管理员身份登录系统 | 数据库登录表中对应的密码更新 |  |
| 2. 在教师管理模块中找到要重置密码的教师 |
| 3. 点击重置密码按钮，输入新密码 |
| ADM - 009 | 重置学生密码 | 1. 以管理员身份登录系统 | 数据库登录表中对应的密码更新 |  |
| 2. 在学生管理模块中找到要重置密码的学生 |  |
| 3. 点击重置密码按钮，输入新密码 |  |

2.教师功能测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | **测试用例名称** | **测试步骤** | **预期结果** | **实际结果** |
| TCH - 001 | 查询教授课程 | 1. 以教师身份登录系统 | 系统显示该教师所教授的所有课程信息 |  |
| 2. 进入课程查询模块 |
| TCH - 002 | 查询选修学生 | 1. 以教师身份登录系统 | 系统显示选修该课程的学生名单 |  |
| 2. 在课程查询模块中选择一门课程，点击查看选修学生按钮 |
| TCH - 003 | 学生成绩打分 | 1. 以教师身份登录系统 | 所选课程表中的成绩字段更新 |  |
| 2. 在课程查询模块中选择一门课程，点击成绩管理按钮 |
| 3. 为选修该课程的学生输入成绩后点击保存 |
| TCH - 004 | 修改个人密码 | 1. 以教师身份登录系统 | 数据库登录表中对应的密码更新 |  |
| 2. 进入个人密码修改模块 |  |
| 3. 输入旧密码和新密码，点击确认修改按钮 |  |

3.学生功能测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例编号** | **测试用例名称** | **测试步骤** | **预期结果** | **实际结果** |
| STD -001 | 选课操作 | 1. 以学生身份登录系统 | 所选课程表中增加一条选课记录 |  |
| 2. 进入选课模块 |
| 3. 从选课列表中选择课程，点击选课按钮 |
| STD -002 | 查看已修课程 | 1. 以学生身份登录系统 | 系统显示学生已修课程信息 |  |
| 2. 进入课程查看模块，点击查看已修课程按钮 |
| STD -003 | 查看已选课程 | 1. 以学生身份登录系统 | 系统显示学生已选课程信息 |  |
| 2. 进入课程查看模块，点击查看已选课程按钮 |
| STD -004 | 退选课程 | 1. 以学生身份登录系统 | 所选课程表中对应的选课记录删除 |  |
| 2. 在已选课程列表中选择要退选的课程，点击退选按钮 |
| STD -005 | 修改个人密码 | 1. 以学生身份登录系统 | 学生登录表中的密码更新 |  |
| 2. 进入个人密码修改模块 |
| 3. 输入旧密码和新密码，点击确认修改按钮 |

（五）测试执行

按照测试用例的顺序依次执行测试。在测试过程中，记录每个测试用例的实际结果

对于测试中发现的问题，及时记录问题的详细情况，包括出现问题的步骤、错误提示信息、问题的严重程度等。

### 测试报告

（一）测试概述

本次测试对高校教务查询系统进行了全面的功能、界面、数据完整性和安全性测试，共设计了 18 个测试用例，涵盖了管理员、教师和学生的所有功能模块。

（二）测试结果汇总

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试模块** | **测试用例数量** | **通过数量** | **失败数量** | **缺陷率** |
| 管理员功能 | 9 | 9 | 0 | 0 |
| 教师功能 | 4 | 4 | 0 | 0 |
| 学生功能 | 5 | 5 | 0 | 0 |
| 总计 | 18 | 18 | 0 | 0 |

（三）测试结论

最新一期已提交版本项目测试通过。第一轮测试中发现了系统存在的一些缺陷，部分功能未能达到预期要求。经过多次修复并进行回归测试，系统目前所有发现的bug已修复完成

### 核心功能截图：

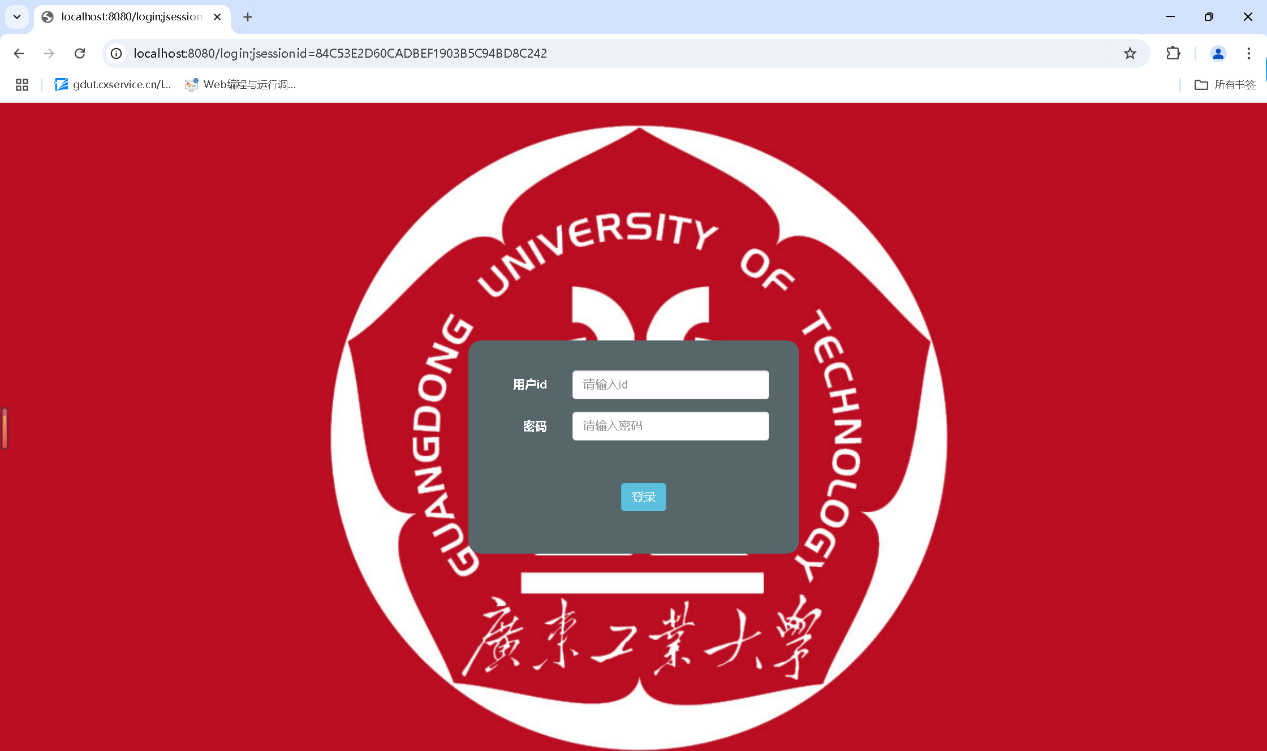


图16：用户登陆界面



图17：登陆成功进入管理员子系统-课程管理界面



图18：课程查询（英语）搜索结果



图19：添加课程信息



图20：修改课程信息



图21：删除课程信息



图22：学生子系统-选课

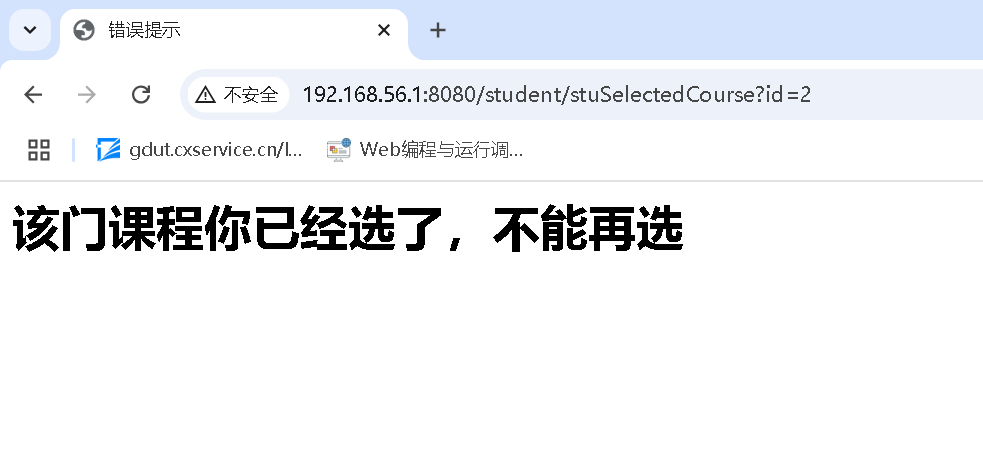


图23：学生选课-选课失败反馈



图24：学生子系统-已修课程（成绩查看）



图25：教师子系统-显示选课学生成绩名单



图26：教师子系统-成绩提交页面

## 七、总结

在完成整个课程设计的过程，我受益匪浅。之前只是接触到书本上的知识比较多，但是没怎么亲自动手实践，有点眼高手低的感觉。但是经过这次实践，可以说对自己有了一个很好的磨炼。从题目的确立，系统的设计，系统的实现，到系统的完成，经历了一整个软件系统开发的过程，让自己的见识有了提高，同时学会了很多的处理问题和解决问题的方法，为将来走入到工作岗位上打好了基础。在需求分析阶段，学会了如何分析问题准确地获取用户需求，并将其转化为可实现的功能需求和非功能需求；在系统设计阶段，掌握了架构设计、数据库设计和详细设计的方法和技巧，能够设计出合理、高效的系统架构和数据库结构；在编码实现阶段，进一步熟悉了程序员对需求进行编码实现的全过程，提高了代码的编写能力和调试能力；在测试阶段，学会了如何制定测试计划、设计测试用例、执行测试以及分析测试结果，提高了软件测试的能力和质量意识。

本次软件工程课程设计是一次宝贵的实践经历，让我在理论与实践的结合中得到了全面的锻炼和提升。通过这次课程设计，我不仅完成了一个具有实际应用价值的系统，还收获了丰富的专业知识和实践经验，提高了我的问题解决能力和综合素质。在今后的学习和工作中，我将继续巩固和深化在课程设计中学到的知识和技能，不断提高自己的软件工程水平。同时，我也将以更加严谨的态度和科学的方法对待软件开发工作，注重软件的质量和用户体验，为推动软件行业的发展贡献自己的力量。

## 八、参考文献

[1] 王珊, 萨师煊. 数据库系统概论.第5版.北京：高等教育出版社.2014

[2] Zhang, S., & Li, J. (2017). "Design of Educational Management Information System Based on Web." Journal of Educational Technology Development and Exchange

[3] 张海藩.软件工程导论.第 6 版.清华大学出版社.2013.