

**(北京)**

# CHINA UNIVERSITY OF PETROLEUM

# 数据库课程设计报告

**高校教务管理与学生服务系统**

**数据库设计报告**

院系名称： 信息科学与工程学院

专业名称： 计算机科学与技术

组 长： 赵新龙

小组成员： 佟迪 焦汇珅 周宇航 赵翊 丁嘉豪

完成日期 2021 年 12 月 30 日

**目录**

[0引言 4](#_Toc91791668)

[编写说明 4](#_Toc91791669)

[背景 4](#_Toc91791670)

[标识符和状态 4](#_Toc91791671)

[数据对象命名约定 4](#_Toc91791672)

[（1）数据表中表（Table）的命名约定 4](#_Toc91791673)

[（2）数据库表中字段（表的列Column）命名约定 4](#_Toc91791674)

[（3）主键（primary key）命名约定 4](#_Toc91791675)

[（4）外键（foreign key）命名约定 5](#_Toc91791676)

[（5）数据库中存储过程（Store Procedure）命名约定 5](#_Toc91791677)

[（6）数据库中触发器（Trigger）命名约定 5](#_Toc91791678)

[（10）数据库中创建的视图（View）命名约定 5](#_Toc91791679)

[（11）数据库中创建的函数（FUNCTION）的命名约定 5](#_Toc91791680)

[1、系统概述 6](#_Toc91791681)

[2、系统需求分析 6](#_Toc91791682)

[​​2.1 功能需求 6](#_Toc91791683)

[​2.2 数据需求 7](#_Toc91791684)

[2.2.1 数据流图 7](#_Toc91791685)

[2.2.2数据字典 8](#_Toc91791686)

[3、系统设计 12](#_Toc91791687)

[​3.1 系统软件结构设计 12](#_Toc91791688)

[​3.2 数据库设计 13](#_Toc91791689)

[3.2.1 概念设计 13](#_Toc91791690)

[3.2.2 逻辑结构设计 14](#_Toc91791691)

[3.2.2.1表结构 14](#_Toc91791692)

[3.2.2.2 视图设计 15](#_Toc91791693)

[3.2.2.5 函数设计 17](#_Toc91791694)

[3.2.2.6 角色设计 17](#_Toc91791695)

[3.2.3 物理结构设计 18](#_Toc91791696)

[3.2.3.1设计数据的存取路径 18](#_Toc91791697)

[3.2.3.2设计数据的存放位置 18](#_Toc91791698)

[3.3 其它设计 18](#_Toc91791699)

[3.3.1 数据库完整性设计 18](#_Toc91791700)

[3.3.2 数据库安全性设计 19](#_Toc91791701)

[3.3.3 故障诊断与恢复 19](#_Toc91791702)

[3.3.4 扩展功能：语音识别模块 19](#_Toc91791703)

[4、系统实现 19](#_Toc91791704)

[4.1 用户登录 20](#_Toc91791705)

[4.2 教师端 21](#_Toc91791706)

[4.3 学生端 25](#_Toc91791707)

[4.4 教务处管理端 30](#_Toc91791708)

[4.5 数据库存储过程 40](#_Toc91791709)

[5、结论 41](#_Toc91791710)

[参考资料 41](#_Toc91791711)

**高校教务管理与学生服务系统设计与实现说明书**

# 0引言

# 编写说明

编写《数据库设计说明书》的目的是为系统设计、系统开发提供参考，是对于设计的系统中各个业务模块数据库的所有标识，逻辑结构和物理结构做出具体的设计规定，其预期读者为，此系统的系统分析人员、开发人员、具体业务用户。

# 背景

系统名称： 高校教务管理与学生服务系统

版本号：1.0

提出者： 数据库课程设计小组

需求调研、设计者：数据库课程设计小组

用户：学生、教师、教务处、辅导员等

## 标识符和状态

数据库名称: 高校教务管理与学生服务数据库 标识：1.0

使用数据库的应用程序名称：Eclipse IDEA 版本号：2018

开发工具程序名称：Eclipse IDEA 版本号：2018

SQL Server 版本号：2019

## 数据对象命名约定

# （1）数据表中表（Table）的命名约定

数据库中的基本表，以实际业务名称的英文缩写进行命名。

**（2）数据库表中字段（表的列Column）命名约定**

表中字段的命名以整理后数据项名称的英文代码缩写为主。

**（3）主键（primary key）命名约定**

（主健英文缩写）PK\_+表名，有时直接以关键字PRIMARY KEY附加的方式声明。

**（4）外键（foreign key）命名约定**

（外健英文缩写）FK\_+表名（多个时，表明后面跟阿拉伯数字1、2、3。。。），有时直接以子句形式声明。

**（5）数据库中存储过程（Store Procedure）命名约定**

存储过程一般以涉及操作的中文名称命名。

**（6）数据库中触发器（Trigger）命名约定**

主要以触发字段+\_trigger的形式命名，有时为了区分不同表上建立的相同字段名的触发器，在上述命名后加上表名。

**（10）数据库中创建的视图（View）命名约定**

使用视图所依赖的基本表名+\_view的方式进行命名。

**（11）数据库中创建的函数（FUNCTION）的命名约定**

在数据库中创建的函数（FUNCTION），之后加最能表现函数（FUNCTION）内容的英文缩写命名。

# 1、系统概述

设计的高校教务管理与学生服务系统主要分为三大模块层面，一是：学生使用层面，二是：教师使用层面，包括教师、导师、辅导员和班主任，三是：教务处管理层面。不同层面根据不同用户的需求所实现的功能不同，这样能够更人性化地贴合个体的使用，最大程度地提升系统的使用及运行效率。所以系统设有多种不同的登录选择，用户根据实际情况自行登录，修改、查询、管理信息。

# 2、系统需求分析

# ​​2.1 功能需求

该系统由三大功能模块构成，分别为教务信息管理模块、学生模块、教师模块。

###### 2.1.1 教务处管理模块

a) 学生基本信息录入：教务信息管理员通过此模块录入学生学籍信息。

b) 学生基本信息查询：教务信息管理员通过此模块查询学生学籍信息。

c) 学生基本信息更新：教务信息管理员通过此模块修改学生学籍信息。

d) 处理学生的待办事项：教务处管理员处理学生通过系统提交的待办事项。

e) 荣誉、处分处理：教务处录入、删改学生的奖惩信息。

f) 学期报道管理：教务处管理员查询学生的学期报道。

g) 查询学生老师关系：查询每个学生对应的老师，如导师、辅导员等。

###### 2.1.2 学生端

a) 个人信息查询模块：学生通过此功能查询自己的个人信息，主要包括：学号、姓名、性别、入学年份、生源地、所在学院、系、班级等。

b) 选课：学生通过此功能进行选课，支持选课的提交与撤销。

c) 获奖情况查询：学生可以查询自己的获奖情况。

d) 选课结果查询：选课完成后，学生可以查询个人选课情况。

e) 待办事项提交：学生通过教务学籍管理系统学生端提交待办事项，后续由教务管理员处理。

f) 班级信息查询：学生可以查询自己的班级信息和导师信息。

g) 学生缴费管理

h) 志愿服务管理：相关负责人可以通过此功能模块登录，查询学生的志愿时长，提交待办事项请求修改志愿时长。

i) 党务管理：相关负责人可以通过此功能登录系统，查询学生政治面貌等信息，管理党员、团员信息，提交待办事项，如更新学生政治面貌为团员。

###### 2.1.3 教师端

a) 成绩录入：教师可以通过此模块录入学生成绩，便于学生成绩的管理。

b) 查询成绩：教师可以通过此模块进行学生成绩的查询。

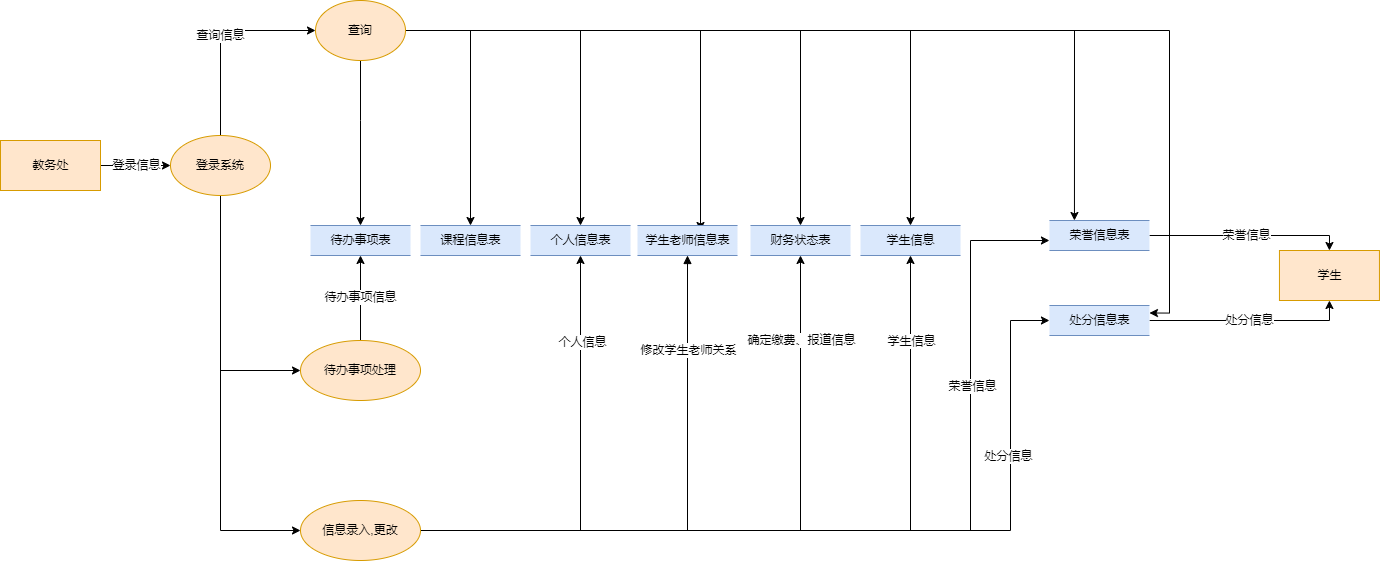
c) 查询选课学生信息：教师可以通过此模块查询选课学生的信息。

d) 查询所授课程信息：教师可以通过此模块查询所教课程信息。

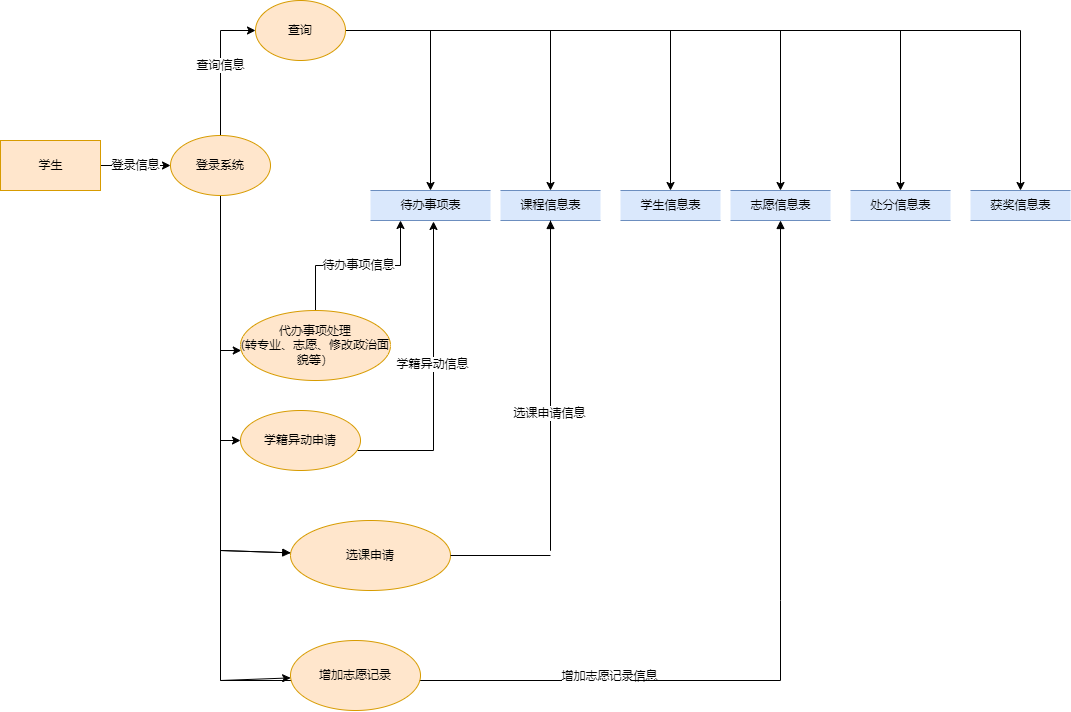
# ​2.2 数据需求

# 2.2.1 数据流图

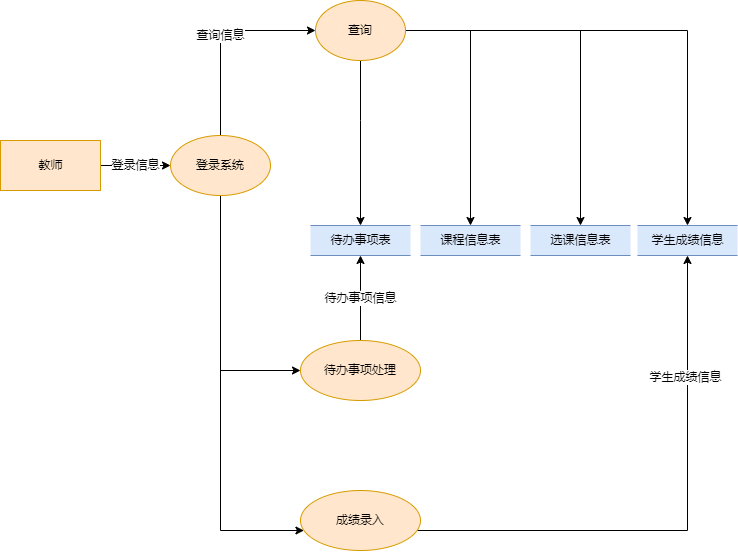
**（1） 教务处管理数据流图**

****

**（2） 学生使用数据流图**

****

**（3） 教师使用数据流图**

****

# 2.2.2数据字典

数据字典由两部分组成：数据表信息、数据项信息。

数据类型列表

|  |  |
| --- | --- |
| **数据类型** | **说明** |
| 实体 | 表示复合元素，可由元素或实体组成 |
| 字符型 | 自由文本，表明对字段的内容没有限制 |
| 整型 | 通过整数的形式表达值的类型 |
| 浮点型 | 通过浮点数的形式表达值的类型 |
| 日期型 | 通过YYYYMMDD的形式表达的值的类型 |
| 位型 | 只有两个表明条件的值，如：是或否 |

（1）学生信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 | 备注 |
| 1 | Stunum | char(12) | 学号 | Key, not null |
| 2 | Stuname | char(10) | 姓名 | Not null |
| 3 | Stupro | char(10) | 专业 | Not null |
| 4 | Stuage | int | 年龄 |  |
| 5 | Iden | char(10) | 政治面貌 |  |
| 6 | Teanum | char(10) | 导师 | Foreign key(teacher) |
| 7 | Clanum | char(10) | 班级号 | Foreign key(class) |
| 8 | College | char(10) | 所在学院 |  |
| 9 | Grade | char(10) | 年级 | Not null |
| 10 | condition | char(10) | 学生状态 | //休学、退学 |
| 11 | Location | char(10) | 籍贯 |  |
| 12 | Stuvol | double | 志愿时长 |  |
| 13 | Tuition | int | 学费 |  |
| 14 | Contui | bool | 是否缴费 |  |

（2）教师信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 | 备注 |
| 1 | TeaID | char(10) | 职工号 | Primary key, not null |
| 2 | Teaname | char(20) | 姓名 | Not null |
| 3 | Teapro | char(30) | 专业 | Not null |
| 4 | Teage | int | 年龄 | 0<年龄<140 |
| 5 | Iden | char(10) | 政治面貌 | 群众/团员/积极分子/党员/预备党员 |
| 6 | Teacol | char(30) | 所在学院 |  |
| 7 | Cournum | char(10) | 课程号 | Foreign key(课程信息表) |
| 8 | Workyear | int | 工作年限 |  |
| 9 | Title | char(10) | 职称 | 讲师/助理/副教授/教授/院士 |

（3）课程信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 | 备注 |
| 1 | CourseID | Char(10) | 课程号 | Primary key, not null |
| 2 | CourseName | Char(30) | 课程名 | Not Null |
| 3 | TeacherID | Char(10) | 任课教师编号 | Foreign key(教师信息表) |
| 4 | Counts | int | 选修人数 |  |
| 5 | Credit | int | 学分 | Not Null |
| 6 | Classroom | Char(3) | 上课教室 |  |

（4）选课信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 | 备份 |
| 1 | StudentID | Char(10) | 学号 | Key, Not Null |
| 2 | CourseID | Char(10) | 课程号 | Key, Not Null |
| 3 | Grade | Int | 成绩 | 0~100 |
| 4 | Credit | int | 学分 | Not Null |
| 5 | gpa | float |  |  |

（5）班级信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 | 备注 |
| 1 | Clanum | char(10) | 班级号 | Key |
| 2 | Teapro | char(30) | 专业 |  |
| 3 | Teacol | char(30) | 所在学院 |  |
| 4 | Account | int | 学生数量 |  |

（6）操作记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 | 备注 |
| 1 | Aplnum | char(10) | 申请编号 | Key |
| 2 | Persontype | char(10) | 申请人类型 | Not null |
| 3 | Aplpro | char(30) | 申请项目 | //类似退课、选课等 |
| 4 | Apltable | char(30) | 需要变更的表 |  |
| 5 | Num | char(10) | 申请人（学）职工号 |  |
| 6 | Date | date | 申请发起日期 |  |
| 7 | Limits | Bit | 是否允许 |  |
| 8 | Accomplish | Bit | 是否完成 |  |
| 9 | Result | char(30) | 申请结果 |  |
| 10 | Aplcon | Char(30) |  |  |

（7）财务状态表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 | 备注 |
| 1 | Budnum | char(10) | 记录编号 | Key, not null |
| 2 | StuID | char(10) | 学号 | Foreign key(学生信息表), not null |
| 3 | Stuname | Char(20) | 姓名 | Not null |
| 4 | Stupro | char(30) | 专业 | Not null |
| 5 | Grade | char(10) | 年级 |  |
| 6 | condition | char(10) | 学生状态 | 休学、退学 |
| 7 | Contui | bit | 缴费状态 |  |
| 8 | Tuition | int | 学费 | >=0 |
| 9 | Tuidate | date | 缴费时间 |  |

（8）志愿信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 | 备注 |
| 1 | VolID | char(10) | 志愿编号 | Key, not null |
| 2 | StuID | char(10) | 学号 | Foreign key(学生信息表) |
| 3 | Stuname | char(10) | 姓名 | Not null |
| 4 | Stupro | char(10) | 专业 | Not null |
| 5 | Stuvol | float | 志愿时长 | >=0 |
| 6 | Volpro | char(10) | 志愿项目 |  |
| 7 | Voldate | date | 志愿时间 |  |

（9）获奖记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 | 备注 |
| 1 | Awanum | char(30) | 记录号 | Key, not null |
| 2 | StuID | char(10) | 学号 | Foreign key(student) |
| 3 | Stuname | char(10) | 姓名 | Not null |
| 4 | Stupro | char(10) | 专业 | Not null |
| 5 | Awardate | date | 获奖时间 |  |
| 6 | Awarea | char(30) | 获奖项目 |  |
| 7 | Awagra | char(10) | 获奖等级 |  |

（10）处分记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 | 备注 |
| 1 | Punum | char(10) | 记录号 | Key, not null |
| 2 | StuID | char(10) | 学号 | Foreign key(student) |
| 3 | Stuname | char(10) | 姓名 | Not null |
| 4 | Stupro | char(30) | 专业 | Not null |
| 5 | Pundate | Date | 处分时间 |  |
| 6 | Punarea | char(30) | 处分项目 |  |
| 7 | Pungra | char(10) | 处分种类 |  |

（11）学生老师信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 | 备注 |
| 1 | StuID | char(10) | 学号 | Key, not null |
| 2 | Stuname | char(20) | 学生姓名 | Not null |
| 3 | Assistname | Char(20) | 辅导员姓名 |  |
| 4 | Turtorname | char(20) | 导师姓名 |  |
| 5 | ClassTeaname | char(20) | 班主任姓名 |  |
| 6 | Clanum | char(10) | 班级 | Not null |
| 7 | College | Char(30) | 学院 | Not null |

（12）登录信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段类型 | 说明 | 备注 |
| 1 | userid | char(10) | 用户编号 | Key, not null |
| 2 | password | Char(20) | 密码 | Not null, 长度6~20 |
| 3 | Role | Char(10) | 角色 |  |

# 3、系统设计

# ​3.1 系统软件结构设计

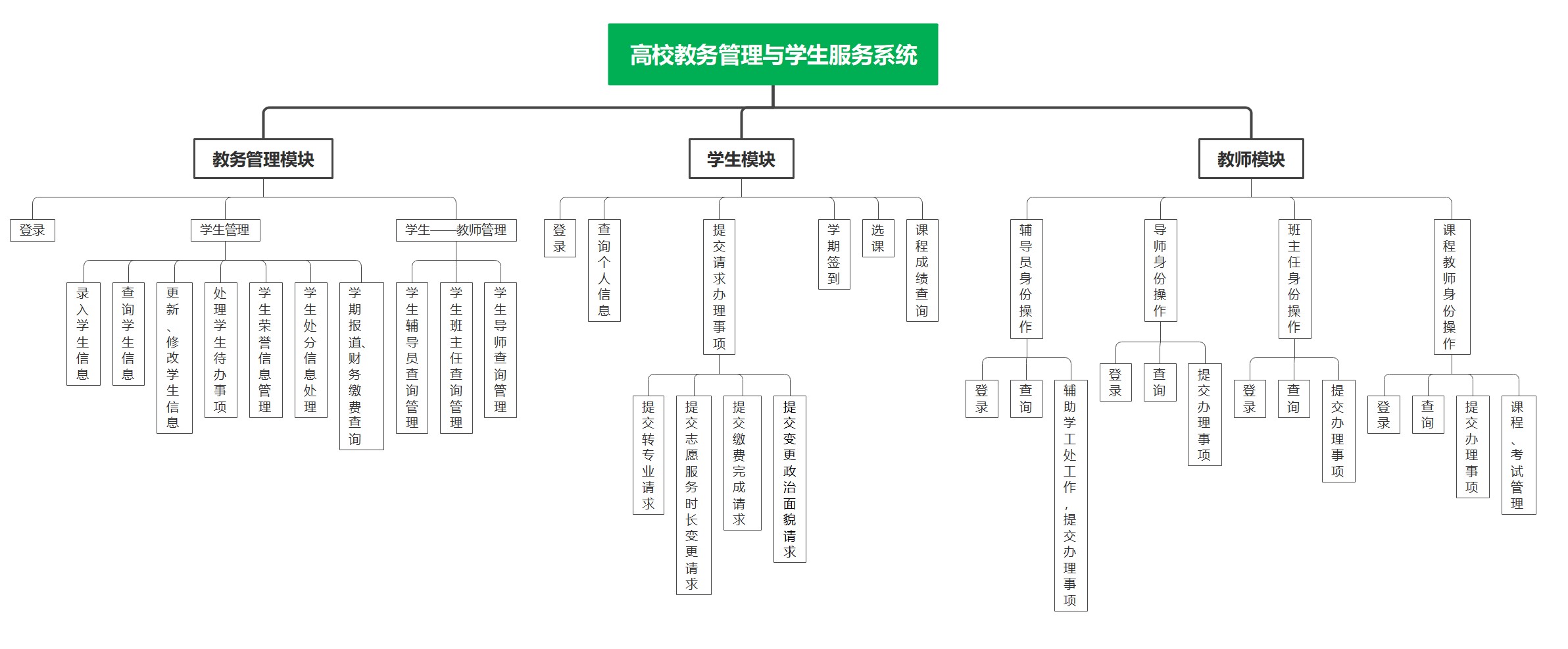
高校教务管理与学生服务系统主要由后台SQL Server数据库和前端Java GUI图形用户界面组成。后台SQL数据库主要负责基本表、视图、存储过程、函数、触发器、角色的定义、创建和维护，为应用系统提供数据库支撑；前台Java程序通过图形化界面与用户进行交互，采用AWT和SWING工具库设计开发各类组件，通过事件响应调用存储过程、函数等实现具体功能。同时，提供了视图和角色机制保障数据库系统的安全性，通过各约束和触发器实现数据表关联数据的合法性，增强了系统的健壮性。

（1）软件系统架构图

图示

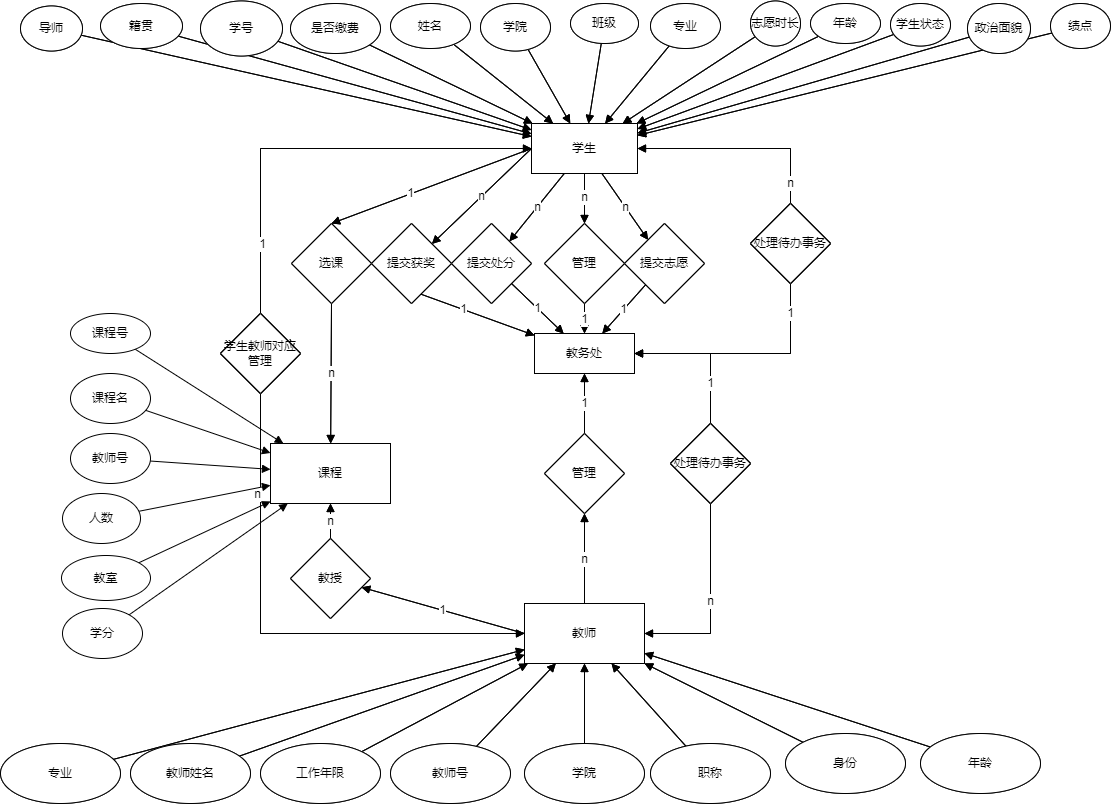
已生成极高可信度的说明

（2）软件功能结构图



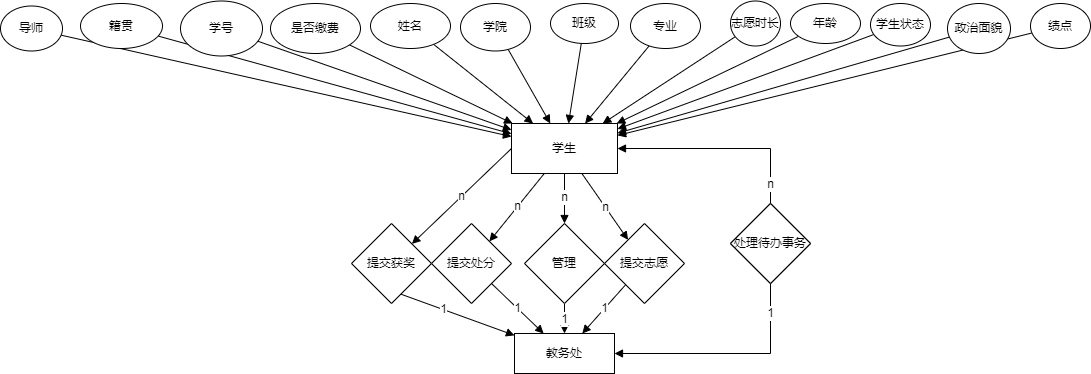
# ​3.2 数据库设计

# 3.2.1 概念设计

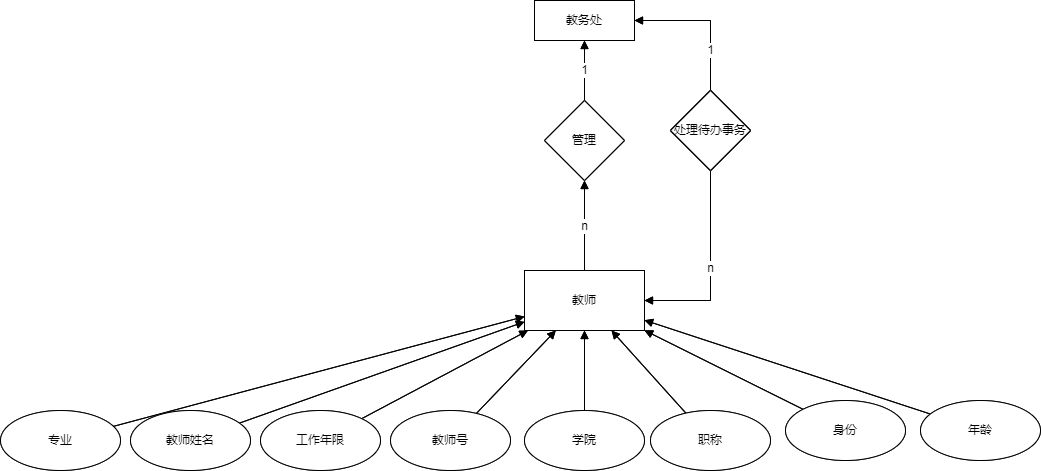


分E-R图：

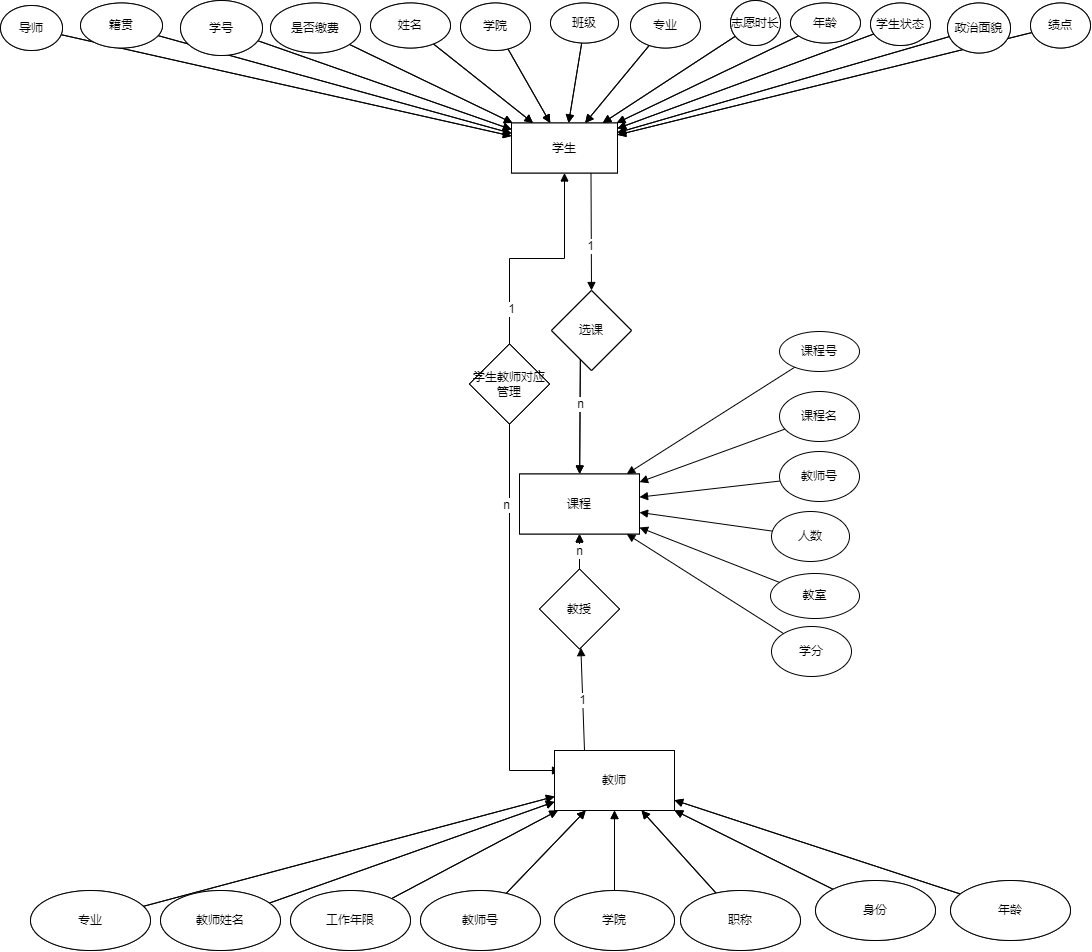
（1）学生-教务处



（2）教师-教务处



（3）教师-学生-课程



# 3.2.2 逻辑结构设计

**3.2.2.1表结构**

见数据字典

**3.2.2.2 视图设计**

表3 视图设计

| **序号** | **视图名称** | **功能描述** | **基表** |
| --- | --- | --- | --- |
| 1 | 教师信息表\_view | 教师信息表的视图，方便授权给指定用户 | 教师信息表 |
| 2 | 处分记录表\_view | 处分记录表的视图，方便授权给指定用户 | 处分记录表 |
| 3 | 学生信息表\_view | 学生信息表的视图，方便授权给指定用户 | 学生信息表 |
| 4 | 志愿信息表\_view | 志愿信息表的视图，方便授权给指定用户 | 志愿信息表 |
| 5 | 操作记录表\_view | 操作记录表的视图，方便授权给指定用户 | 操作记录表 |
| 6 | 班级信息表\_view | 班级信息表的视图，方便授权给指定用户 | 班级信息表 |
| 7 | 获奖记录表\_view | 获奖记录表的视图，方便授权给指定用户 | 获奖记录表 |
| 8 | 课程信息表\_view | 课程信息表的视图，方便授权给指定用户 | 课程信息表 |
| 9 | 财务状态表\_view | 财务状态表的视图，方便授权给指定用户 | 财务状态表 |
| 10 | 选课信息表\_view | 选课信息表的视图，方便授权给指定用户 | 选课信息表 |
| 11 | 志愿时长统计\_view | 为用户查询学生志愿时长提供统一的权限控制，方便授权给指定用户 | 志愿时长统计 |
| 12 | 查询操作记录表 | 对操作记录表的申请编号、申请项目、申请日期、是否允许、完成情况、申请结果提供统一的权限控制，方便授权给指定用户 | 操作记录表 |

#### 3.2.2.3 存储过程设计

表4 存储过程设计

| **序号** | **名称** | **参数** | **功能描述** |
| --- | --- | --- | --- |
| **1** | 发起申请 | 申请日期date，申请人类型iden，申请人编号id，申请项目Aplpro，需要变更的表Apltable | 将用户在程序界面相应组件输入的申请信息插入到操作记录表对应字段中 |
| **2** | 接受申请 | 申请项目编号alpn | 对操作记录表中的申请信息进行处理，主要包括操作结果设置、退学/休学/缴费/毕业操作，对应更新相应基本表 |
| **3** | 拒绝申请 | 申请项目编号alpn | 将需要拒绝的事项置为“拒绝” |
| **4** | 获奖插入 | 编号id，姓名name，专业pro，获奖日期date，获奖项目area，获奖等级gra | 在获奖记录表中插入学生获奖信息 |
| **5** | 处分插入 | 编号id，姓名name，专业pro，处分日期date，处分项目area，处分等级gra | 在处分记录表中插入学生处分信息 |
| **6** | 获奖删除 | 项目编号alpn | 在获奖记录表中删除学生获奖信息 |
| **7** | 处分删除 | 项目编号alpn | 在处分记录表中删除学生处分信息 |
| **8** | 获奖更新 | 项目编号alpn，编号id，姓名name，专业pro，获奖日期date，获奖项目area，获奖等级gra | 更新获奖记录表中的学生获奖信息 |
| **9** | 处分更新 | @apln，@id，@name，@pro，@date，@area，@gra | 更新学生处分信息 |
| **10** | 学生插入 | @id，@name,@pro,@age,@iden,@teaname,@class,@college,@grade,@cond,@loc,@tui,@contui | 向学生信息表、学生老师信息表和登录信息表插入学生信息 |

#### 3.2.2.4 触发器设计

表5 触发器设计

| **序号** | **触发器名称** | **数据表** | **触发方式** | **功能描述** |
| --- | --- | --- | --- | --- |
| **1** | stucon\_trigger | 学生信息表  财务状态表 | After Update | 将学生信息表修改的学生状态信息更新至财务状态表 |
| **2** | grade\_trigger | 学生信息表  财务状态表 | AFTER UPDATE | 将学生信息表修改的学生年级更新至财务状态表 |
| **3** | stupro\_trigger | 学生信息表  财务状态表 | AFTER UPDATE | 将学生信息表修改的学生专业更新至财务状态表 |
| **4** | stupro\_trigger\_学生\_志愿 | 学生信息表  志愿信息表 | AFTER UPDATE | 将学生信息表修改的学生专业更新至志愿信息表 |
| **5** | stupro\_trigger\_学生\_获奖 | 学生信息表  获奖信息表 | AFTER UPDATE | 将学生信息表修改的学生专业更新至获奖信息表 |
| **6** | stupro\_trigger\_学生\_处分 | 学生信息表  处分信息表 | AFTER UPDATE | 将学生信息表修改的学生专业更新至获奖信息表 |
| **7** | 志愿时长同步 | 学生信息表  志愿信息表 | AFTER UPDATE | 修改志愿信息表的志愿时长时，更新至学生信息表 |
| **8** | 成绩\_trigger\_insert | 学生信息表  选课信息表 | after insert | 在选课信息表插入学生成绩后，将信息更新至学生信息表 |
| **9** | 成绩\_trigger\_update | 学生信息表  选课信息表 | After update | 在选课信息表更新学生成绩后，将信息更新至学生信息表 |
| **10** | 选课人数限制\_trigger | 选课信息表  课程信息表 | After insert | 插入学生选课记录前，检查该门课程的选课人数是否超过课程容量（假设100人/课堂） |
| **11** | 学生信息表\_学生老师信息表 | 学生信息表  学生老师信息表 | After insert | 在学生信息表中插入元组时，在学生老师信息表中插入相应的记录 |

**3.2.2.5 函数设计**

表6 函数设计

| **序号** | **名称** | **参数** | **功能描述** |
| --- | --- | --- | --- |
| 1 | 课程绩点计算 | 课程成绩Grade | 若成绩大于50，则该课程绩点为grade/10-5；否则，该课程绩点为0 |
| 2 | 学生总绩点计算 | 学生学号stuid | 在选课信息表中查询指定学生的课程绩点和学分，总绩点为各门课程绩点与学分乘积之和 |

**3.2.2.6 角色设计**

表7 角色设计

| **序号** | **名称** | **说明** | **权限** |
| --- | --- | --- | --- |
| 1 | Student | 学生 | 学生信息表、选课信息表、志愿信息表、志愿时长统计视图、获奖记录表、处分记录表、课程信息表的查询权限，选课信息表的更新权限，操作记录表的更新、插入权限，操作记录表的查询权限 |
| 2 | Teacher | 老师 | 教师信息表、课程信息表的查询、更新、插入权限，选课信息表的查询权限，操作记录表的查询、更新和插入权限 |
| 3 | Manager | 教务处 | 教务处拥有对所有基本表的全部权限 |

# 3.2.3 物理结构设计

**3.2.3.1设计数据的存取路径**

数据库在物理设备上的存储结构与存取方法称为数据库的物理结构，它依赖于选定的数据库管理系统。为了提升数据库系统时间效率和空间性能，需要对数据库的物理结构进行设计，指定关系数据库的存取方法和存储结构，主要包括索引、聚簇索引、存储分区等内容。数据库管理系统一般提供索引和聚簇以实现数据的快速存取。为了加快某个属性或属性组的查询速度，把相应属性上具有相同值的元组集中存放在连续的物理块中称为聚簇。采用SQL Server默认模式，将主键设为聚簇索引。在相同结果集情况下，尽可能减少逻辑I/O。

**3.2.3.2设计数据的存放位置**

确定数据的存放位置和存储结构要综合考虑存取时间、存储空间利用率和维护代价三方面的因素。将数据的经常存取部分和存取频率较低部分分开存放，以提高系统性能。

# 3.3 其它设计

# 3.3.1 数据库完整性设计

综合考虑数据库各数据表之间的关联关系，借助外键、参照完整性约束、触发器机制实现数据表关联信息的一致性。同时，根据字段的实际含义，使用CHECK子句、CONSTRAINT约束等手段，保证插入、修改数据的合理性，增强数据库系统的健壮性。

# 3.3.2 数据库安全性设计

一个健壮、安全、高可用性的数据库是保证企业应用和数据的前提条件。保护措施是数据库管理系统的一个重要技术指标。采用基于角色分配、权限授予等技术，对学生、老师、教务管理人员授予不同权限，并在基本表的基础上创建多张视图，较好地保障了数据库系统的安全性。

# 3.3.3 故障诊断与恢复

在Java界面中设置错误诊断与提示功能，对数据库返回的数据异常进行捕获和处理，便于与用户进行交互，有效提升了系统的可用性。

# 3.3.4 扩展功能：语音识别模块

采用科大讯飞语音识别API接口，将用户输入的语音生成文件，对文件进行语音识别，进而完成成绩录入，方便用户使用。

核心代码如下：

//语音录入成绩

JButton btnSound=new JButton("语音录入成绩");

pane.add(btnSound);

btnSound.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ee) {

Sound f = new Sound();

try

{

f.save("E:\\code\\java\_sqlserver\\record\\test.pcm");

}

catch (Exception e)

{

e.printStackTrace();

}

}

});

# 4、系统实现

采用Java JDBC技术，建立应用界面程序与后台数据库的连接。为界面设计各类组件，用户通过组件输入信息，对按钮点击实施事件响应，系统将信息作为参数传递给后台数据库的存储过程、函数等予以调用和处理，实现前端界面与后台数据库的交互。

各主要功能模块的数据对象结构见前述章节图表。

为了实现对数据库的访问，必须首先使用Class.forName方法载入数据库驱动程序，并通过DriverManager提供的getConnection方法建立连接。

Java连接数据库的核心代码如下（以教师端查询课程信息为例）：

String connectionUrl = "jdbc:sqlserver://localhost:1433;DatabaseName=dbkeshe";

try {

// Establish the connection.

//com.microsoft.jdbc.Sqlserver.SQLServerDriver

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

con = DriverManager.getConnection(connectionUrl, "sa", "159357");

String SQL = "select CourseID,Credit from 课程信息表\_view where TeacherID = " + teacherId;

stmt = con.createStatement();

rs = stmt.executeQuery(SQL);

while (rs.next()) {

courseId = rs.getString("CourseID");

credit = rs.getString("Credit");

}

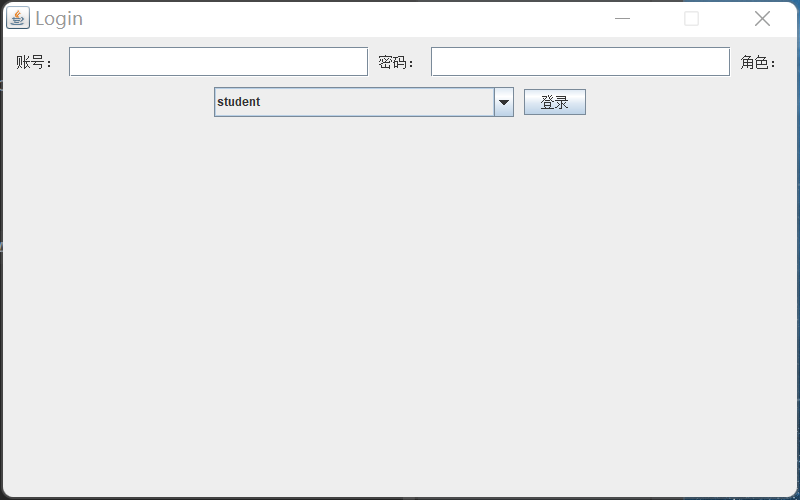
}catch (Exception ex) {

ex.printStackTrace();

}

# 4.1 用户登录

主界面实现登录功能，采用流式布局管理界面组件，用户通过文本框输入账号和密码，选择用户角色，点击登录按钮跳转至功能模块对应的界面。



为了实现用户角色的判断，在“登录”事件监听中，查询后端数据库中的登录信息表，判断输入的用户名和密码是否合法：对合法的登录请求予以跳转相应的界面；对非法用户弹窗提示“用户身份不匹配”或“账号或密码错误”，以增强交互，并保障系统的安全性。

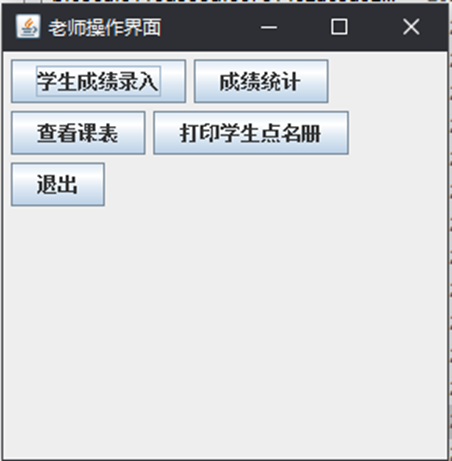
核心代码如下：

文本

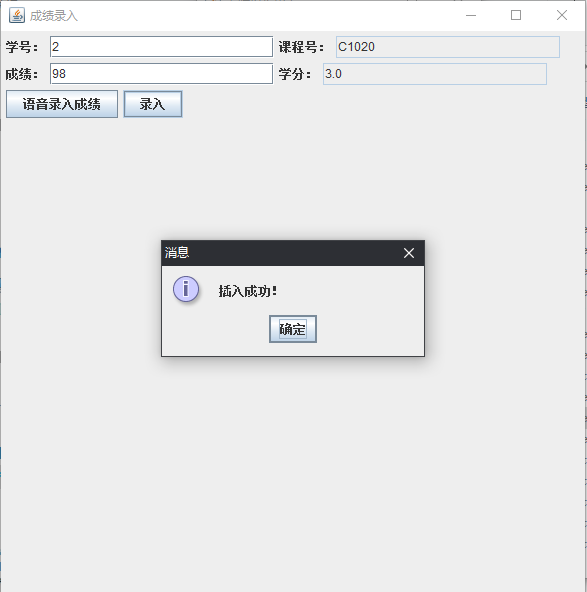
已生成极高可信度的说明

# 4.2 教师端

教师端的主要包括学生成绩录入、成绩统计、查看课表、打印学生点名册和退出五大功能。采用主界面+分界面的设计思想，教师登录后，首先获取教师编号。在GUI界面上，采用流式布局，设置五大功能对应的操作按钮。对每个按钮实现点击事件监听器，跳转对应子界面。



老师可以点击学生成绩录入按钮，通过教师id号，查出所授课程id和此课程的学分，传入课程id号和对应学分，新的页面只需要输入学生id号和学生成绩，课程id和课程学分自动填入。在这个输入页面和我们组另一个同学的语音输入对接，点击语音输入成绩按钮，教师可以语音输入学生的id号和成绩，说出“xx成绩xx”，对应数字会自动填入学生id框和学生成绩框，然后点击录入，对应数据就会插入选课信息表，然后教师可以继续点击按钮语音输入成绩。所有返回结果的表格都配有一个查询按钮，点击查询会弹出另一个窗口，显示对应的数据库最新数据。

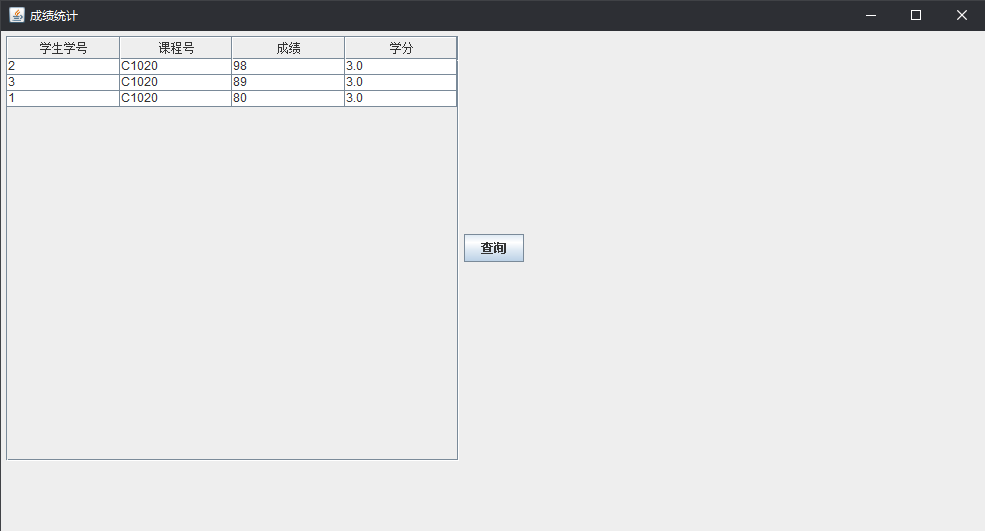


核心代码如下：

//查询语句

String SQL = "insert into 选课信息表\_view(StuID, CourseID, Grade, Credit)" +"values('"+txtSno.getText()+"','"+txtCno.getText()+"','"+txtScores.getText()+"','"+txtCredit.getText()+"')";

教师可以点击成绩统计，系统通过教师id号查询到所授课程id号，通过课程id号查询查询选课信息表，返回选择这门课的学生的id和成绩，显示在表格里。



核心代码如下：

String SQL = "select \* from 选课信息表\_view where CourseID =" + "'"+courseId +"'"+ " order by Grade desc";

System.out.println(SQL);

stmt=con.createStatement();

rs =stmt.executeQuery(SQL);

while(rs.next()) {

String[] s = new String[] {rs.getString("StuID"),rs.getString("CourseID"),rs.getString("Grade"),rs.getString("Credit")};

list.add(s);

}

教师可以点击查询自己所授课程，系统会通过已获得的教师id号，查询课程信息表的视图，返回表中所有与教师有关的课程和相关信息，并显示在页面表格中。



核心代码如下：

// Create and execute an SQL statement that returns some data.

String SQL = "select \* from 课程信息表\_view where TeacherID =" + teacherId;

stmt=con.createStatement();

rs =stmt.executeQuery(SQL);

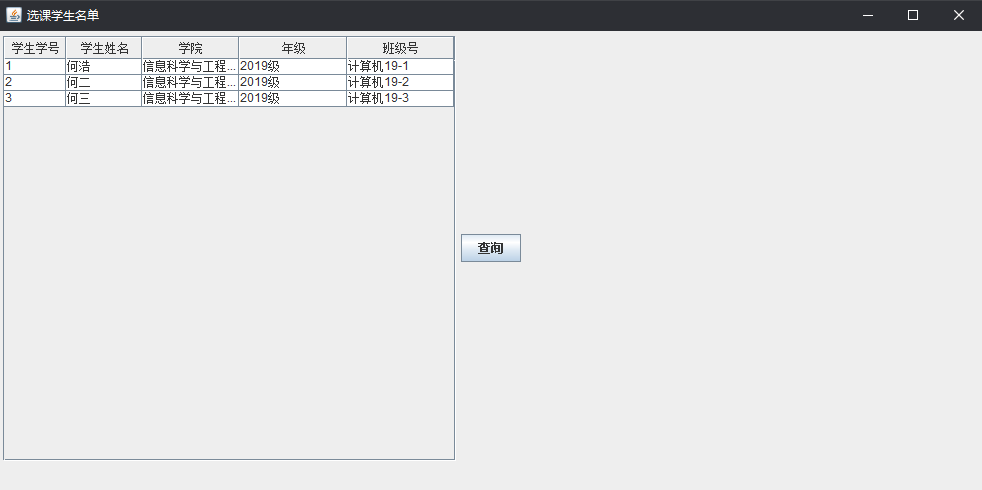
while(rs.next()) {

String[] s = new String[] {rs.getString("CourseID"),rs.getString("CourseName"),rs.getString("TeacherID"),rs.getString("Counts"),rs.getString("Credit"),rs.getString("Classroom")};

list.add(s);

}

教师还可以点击打印学生点名册，系统会通过教师id号查询到所授课程id号，通过课程id号查询选课信息表的视图得到学生id号，然后通过学生id号查询学生信息表返回上课学生信息。



核心代码如下：

// Create and execute an SQL statement that returns some data.

String SQL = "select 学生信息表\_view.StuID sno,学生信息表\_view.StuName sname,学生信息表\_view.College coll,学生信息表\_view.Grade gra,学生信息表\_view.Clanum cla from 选课信息表\_view,学生信息表\_view where 选课信息表\_view.StuID = 学生信息表\_view.StuID and CourseID = " + "'"+courseId +"'";

stmt=con.createStatement();

rs =stmt.executeQuery(SQL);

String[] s = new String[]{rs.getString("sno"),rs.getString("sname"),rs.getString("coll"),rs.getString("gra"),rs.getString("cla")};

# 4.3 学生端

学生端主要包括选课容量查询、提交选课申请、修改志愿信息和党务相关功能。

在student.java中，实现学生端主界面，界面整体采用流式布局进行组件管理，左上部分使用JLabel组件显示学生信息表中的学生个人基本信息，数据来源于select查询学生信息表所得结果；右侧和下方的操作按钮进行实践监听并响应，为了方便用户操作，采用下拉菜单的形式进行信息输入。

学生申请包括选课、转专业、退学、休学和交学费五项功能。

党务，志愿，转专业，休学，都要交给操作记录表，然后由管理员审批。



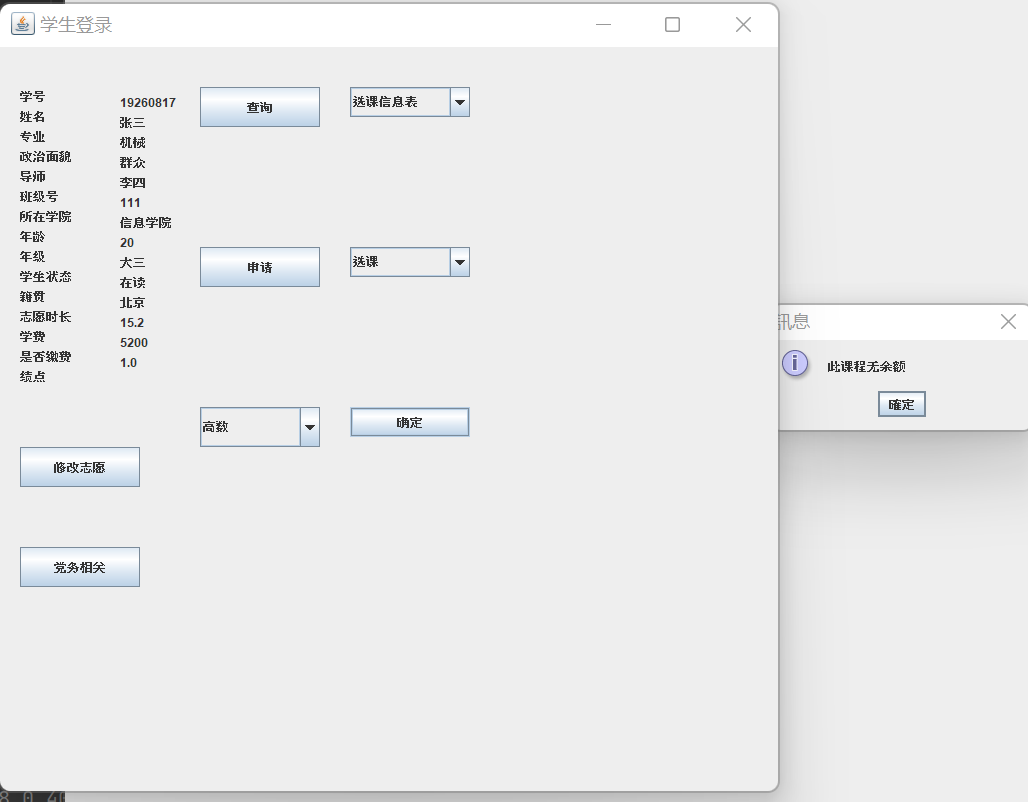
学生通过系统可以进行转专业操作，通过下拉菜单选择转入专业名称，点击确定按钮，可以进行事件响应，修改学生信息表中的学生专业信息。



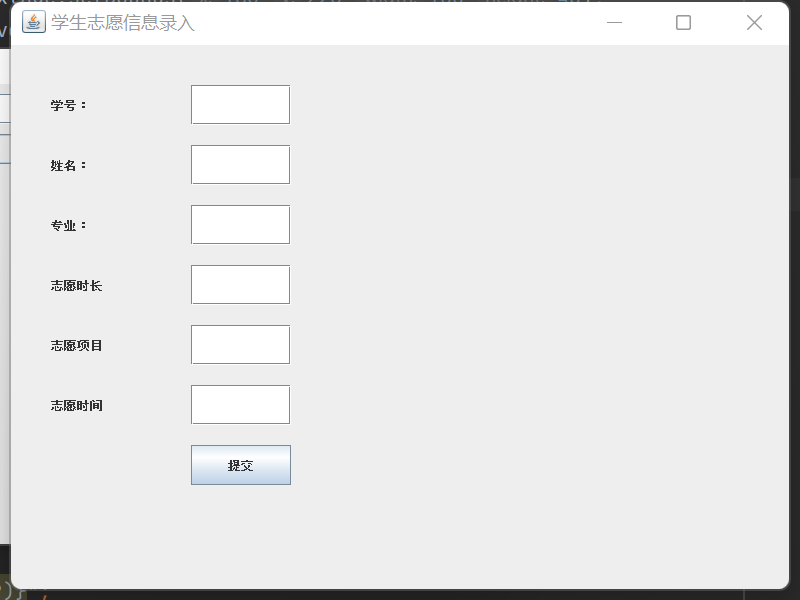
可以查询所在班级、专业、学院以及学生数量信息，以表格形式显示查询结果。

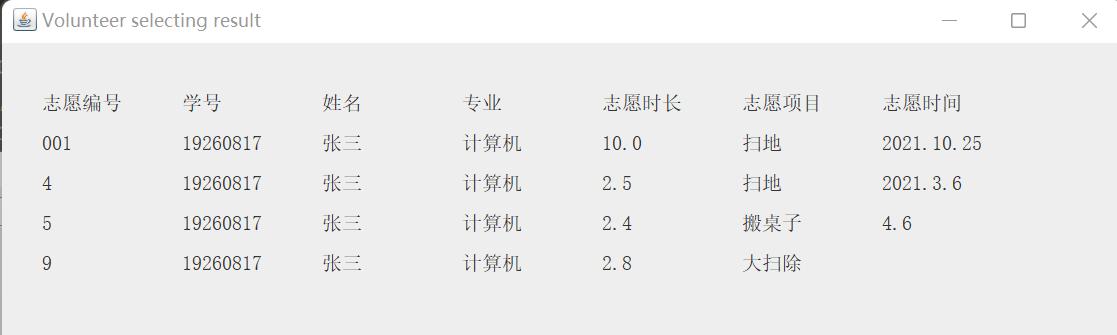


当学生拟选课程无课余量时，弹窗提示“此课程无余额”。



用户可以进行志愿信息的查询和办理，与前述流程类似，连接数据库调用SQL语句进行查询、录入等操作，将数据库操作结果赋给界面组件，以表格形式显示。

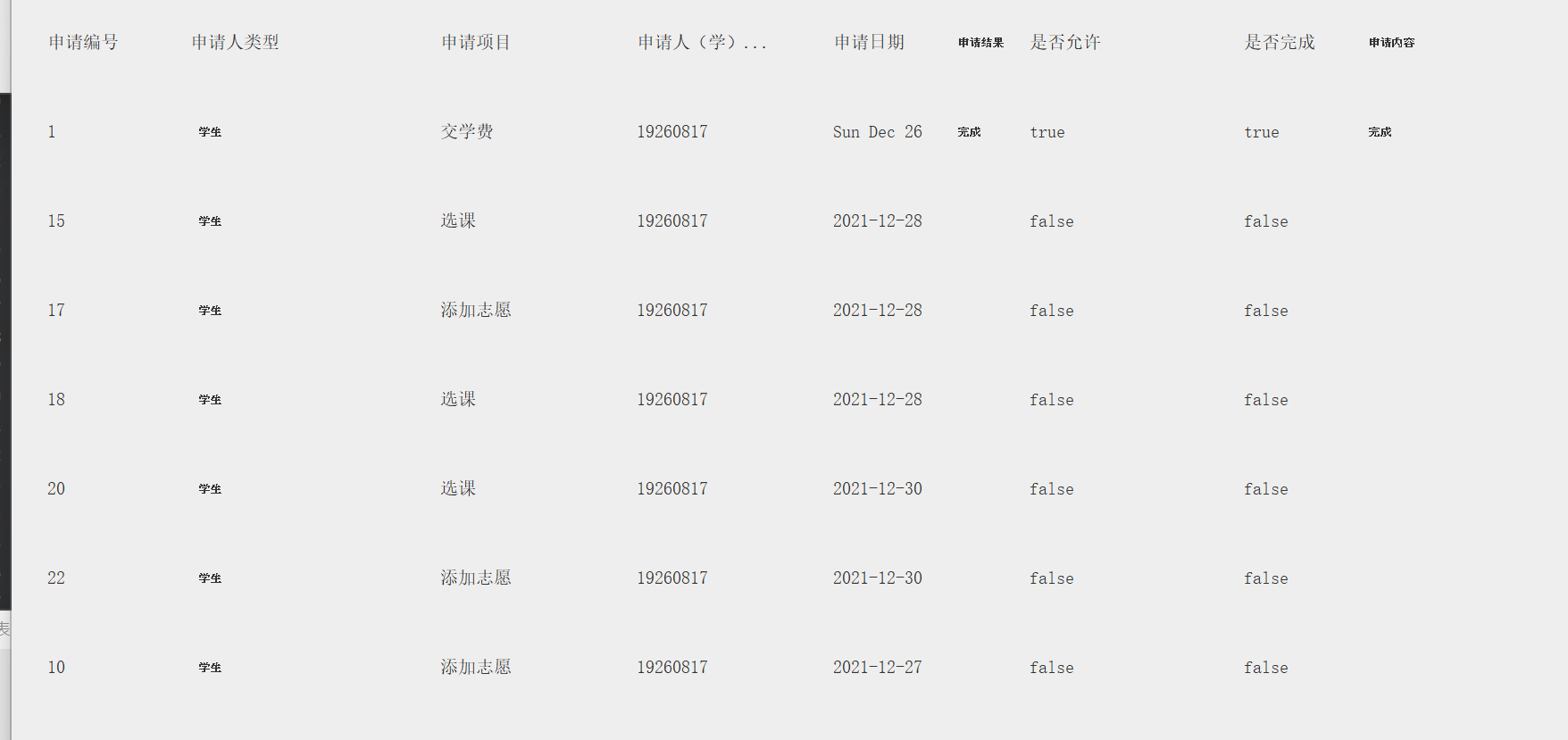




用户可以查询学生处分信息，显示查询结果。



对操作记录表进行查询，显示申请人提交的申请项目信息。可以查看申请事项完成情况。



学生端还实现了党务管理功能，党务事项的按钮是提交操作记录表管理，由管理员负责审核。



核心代码如下：

//选课按钮响应

if(option.equals("选课")) {

veri.setVisible(true);opt.setVisible(true);

exeapply xuanke = new exeapply();

xuanke.init(opt,id,option);

veri.addActionListener(xuanke);

}

//获奖记录表

文本

已生成高可信度的说明

//选课申请（调用存储过程）

文本, 信件

已生成极高可信度的说明

//志愿信息插入

文本

已生成极高可信度的说明

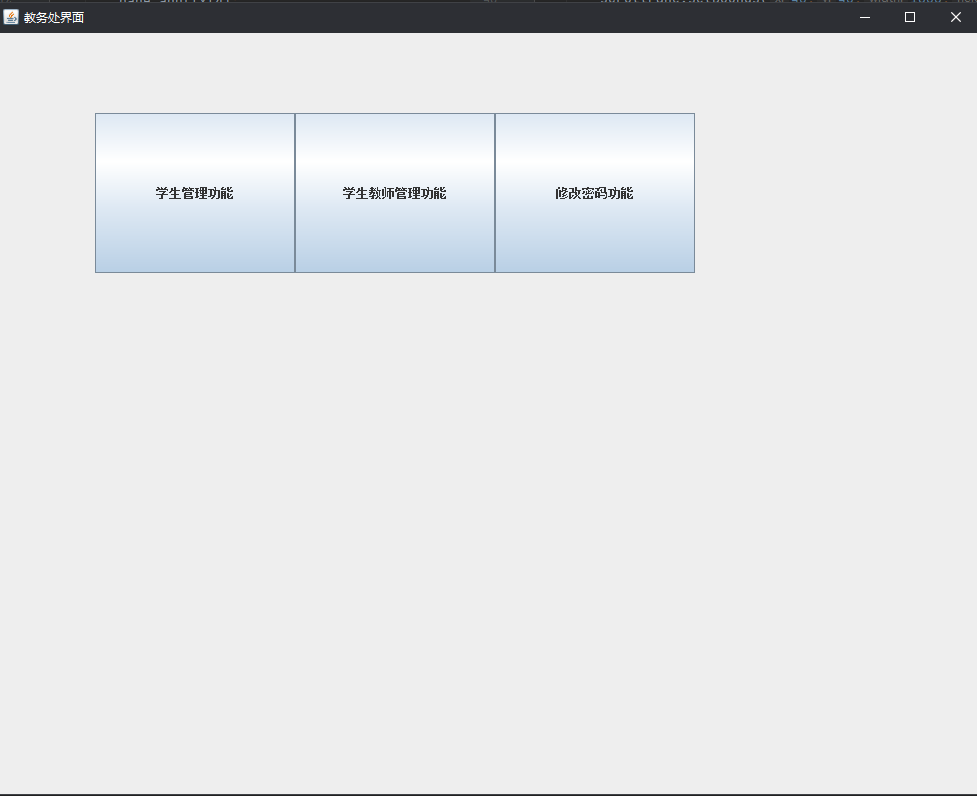
//党务相关申请

文本, 信件

已生成极高可信度的说明

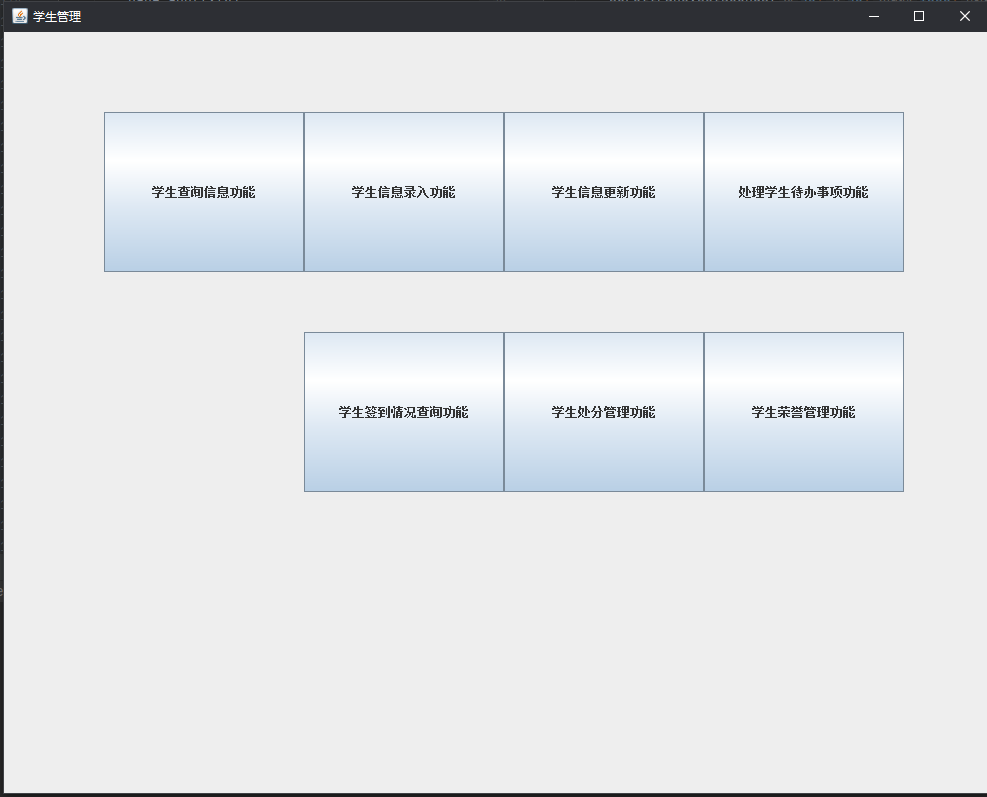
# 4.4 教务处管理端

教务处管理端主要实现学生管理、学生教师管理、修改密码等功能模块。调用SQL语句，将前端输入的数据传递给后台服务器存储过程等，前端调用数据库SQL语句即可完成对数据库的相应操作。

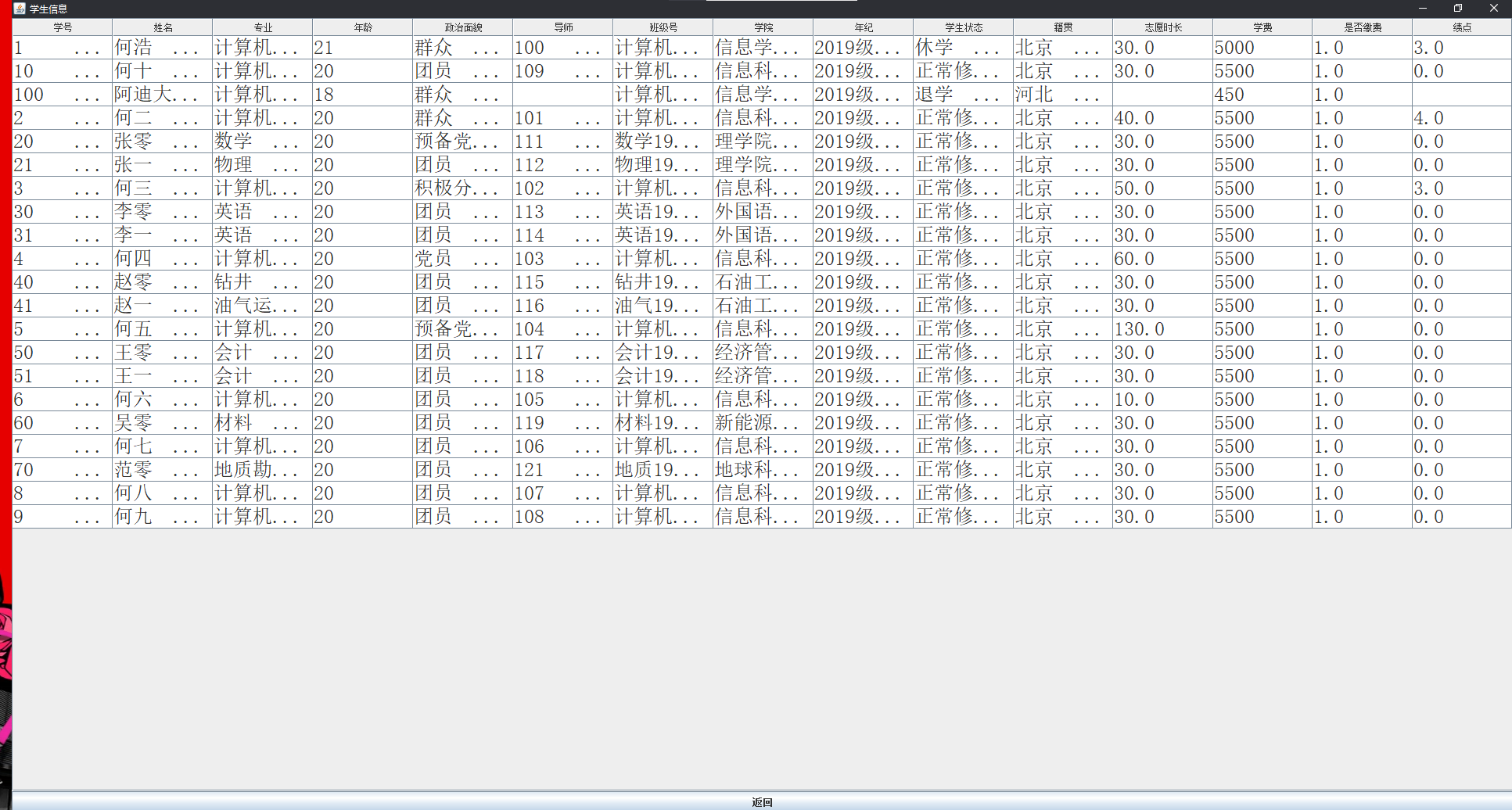


（1）学生管理模块

教务处管理员可以对学生查询信息、信息录入、变更，处理学生待办事项，学生奖惩管理，学生签到情况进行增删改查，点击按钮可跳转至对应界面。



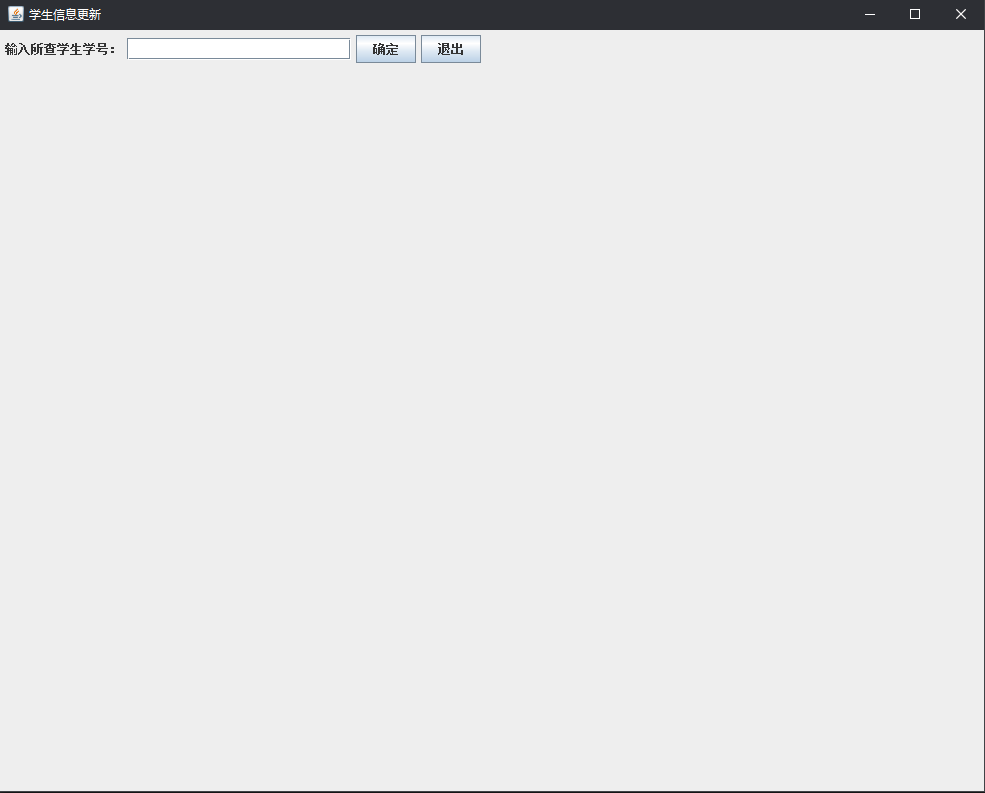
点击“学生查询信息功能”，调用SQL语句，将查询结果以表格方式显示。执行SQL语句后，将得到ResultSet类型的结果集。通过遍历结果集记录的方法可以将每条结果解析为字符串数组元素，便于组件使用和显示。



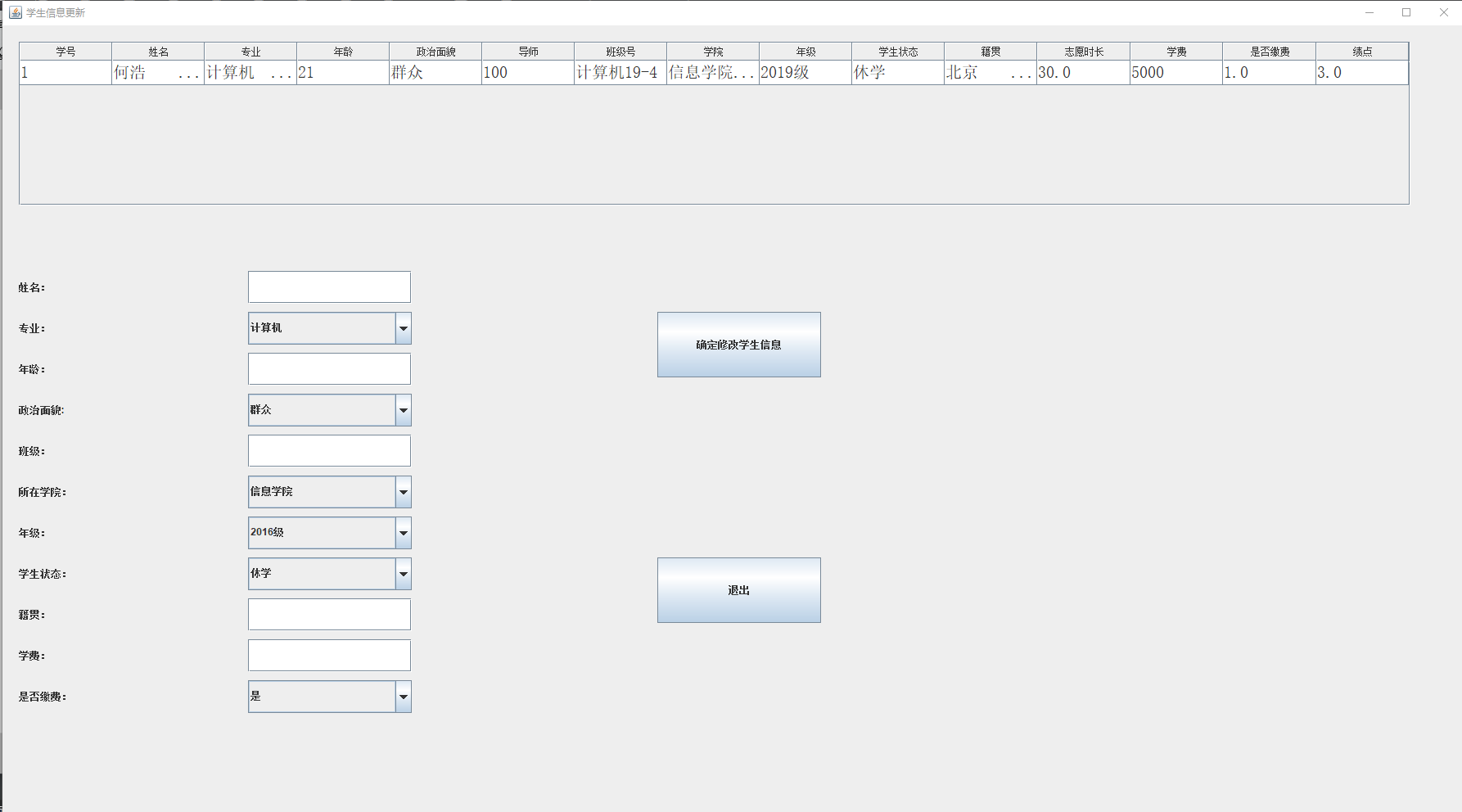
点击“学生信息录入”，调用SQL存储过程，将Java GUI组件获取到的数据写入数据表。注意需要对学生老师信息表添加学生信息，对学生信息表新增记录。



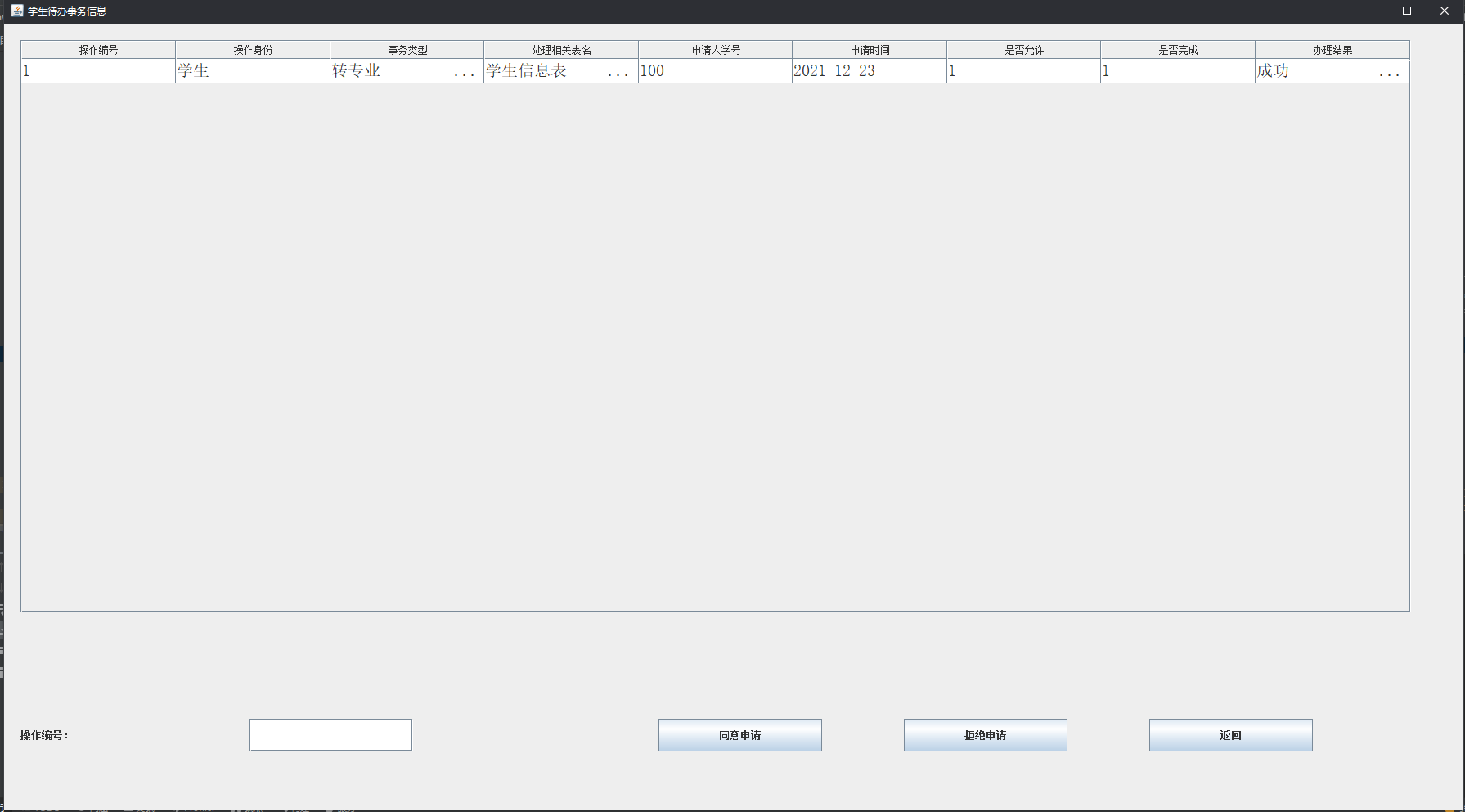
对学生信息可以进行修改操作，首先在界面的文本组件中获取输入的学生学号，点击确定按钮，从学生信息表中查找学生的学号、姓名、专业等基本信息并显示在JTable组件中。



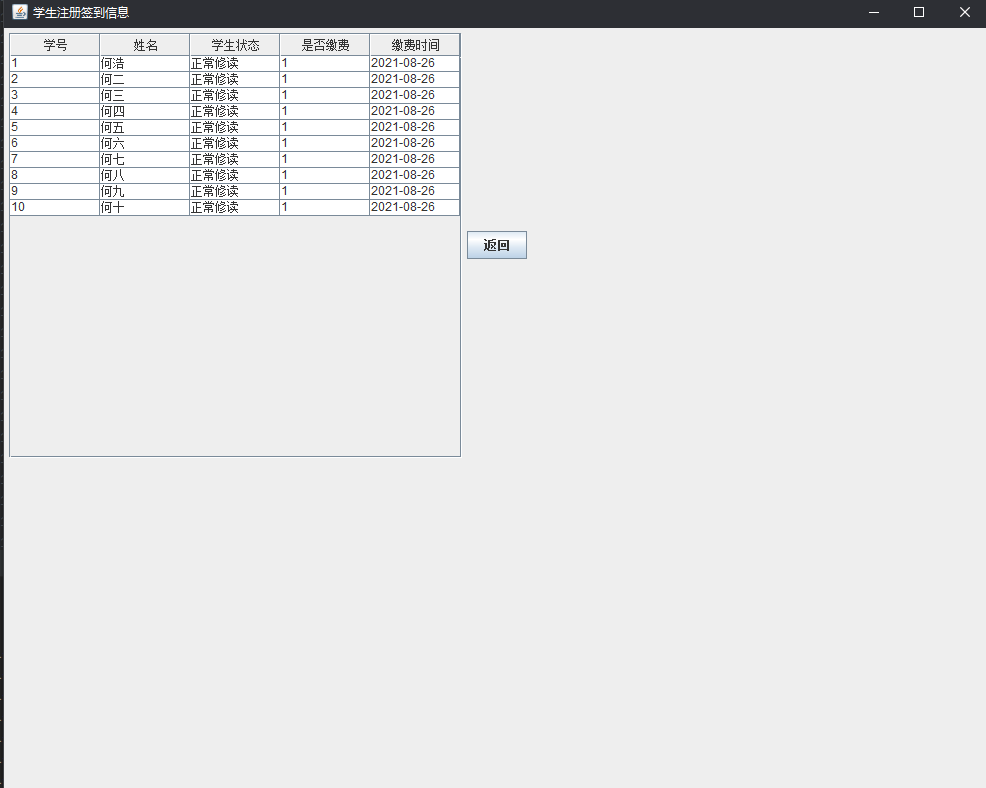
在学生信息修改界面中，可以通过文本输入和下拉菜单选择两种方式输入修改的学生信息项。具体修改通过操作学生信息表实现，同时考虑了不同基本表信息同一属性修改的同步问题。



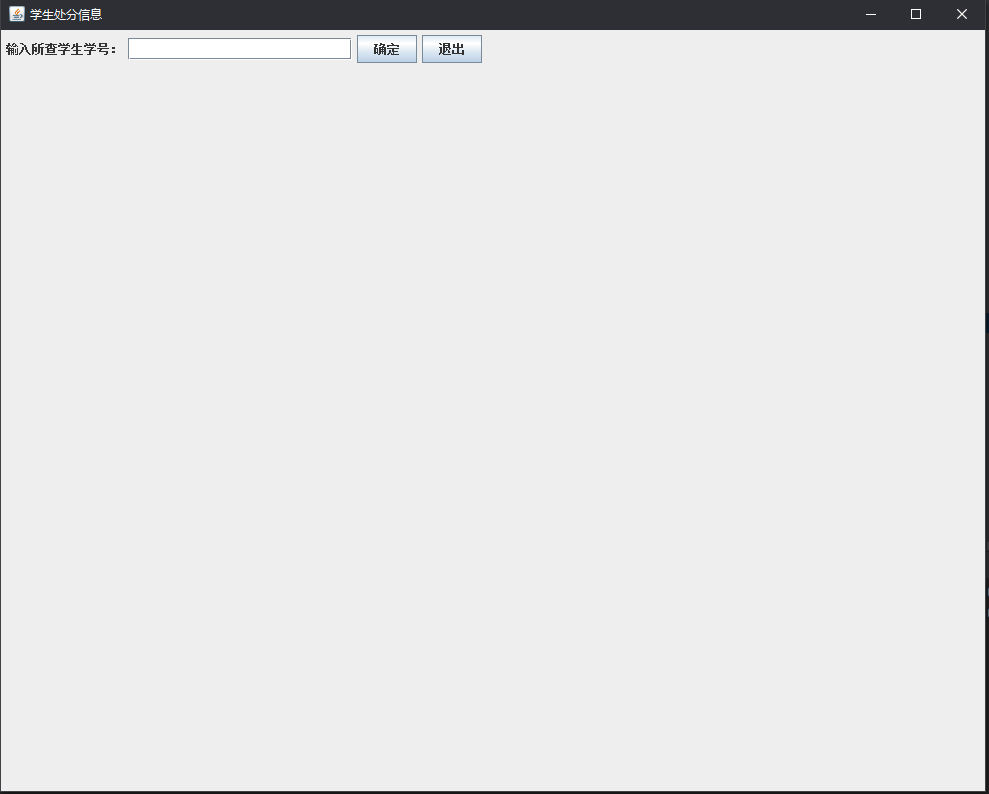
教务处管理人员可以输入需要处理的操作记录编号，对操作记录表视图中的待办事项进行查询，以表格形式列出查询结果。对操作记录表中学生或教师已提交的待办事项予以同意或拒绝。具体实现基于操作记录表，通过在按钮点击事件监听函数体内调用后台数据库的存储过程，将界面输入的数据信息作为存储过程的参数传入后台数据库服务器，由存储过程执行对应的功能。

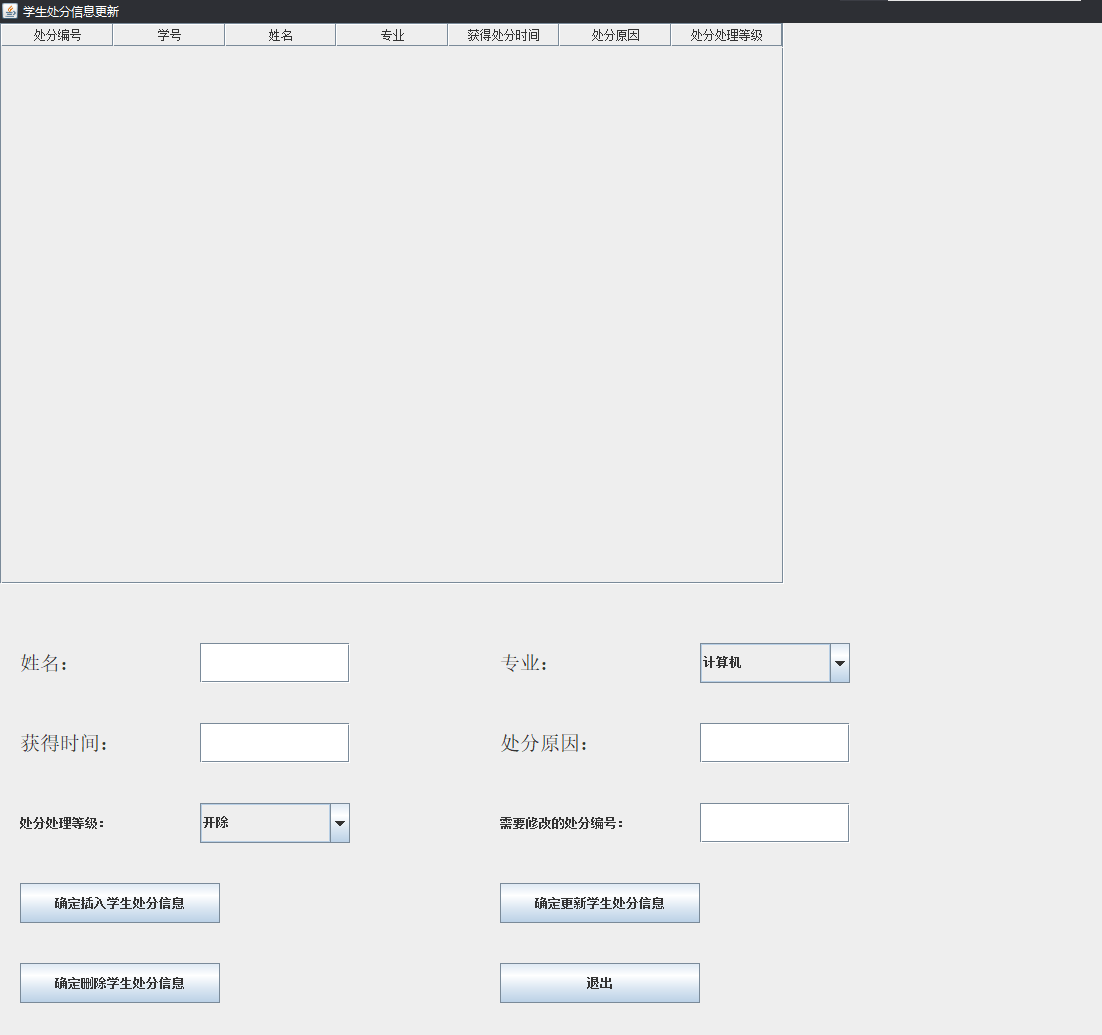


通过教务端，还可以查询所有学生的修读状态、缴费情况和缴费时间。具体实现是通过在财务信息表中调用SQL查询语句，将返回的查询结果集记录项解析，得到字符数组并显示在表格组件上。

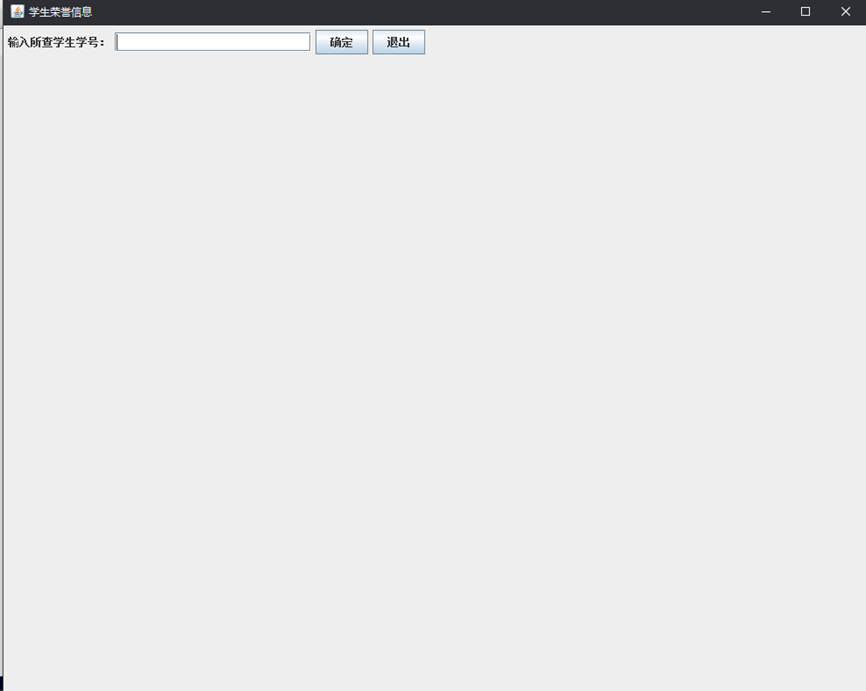


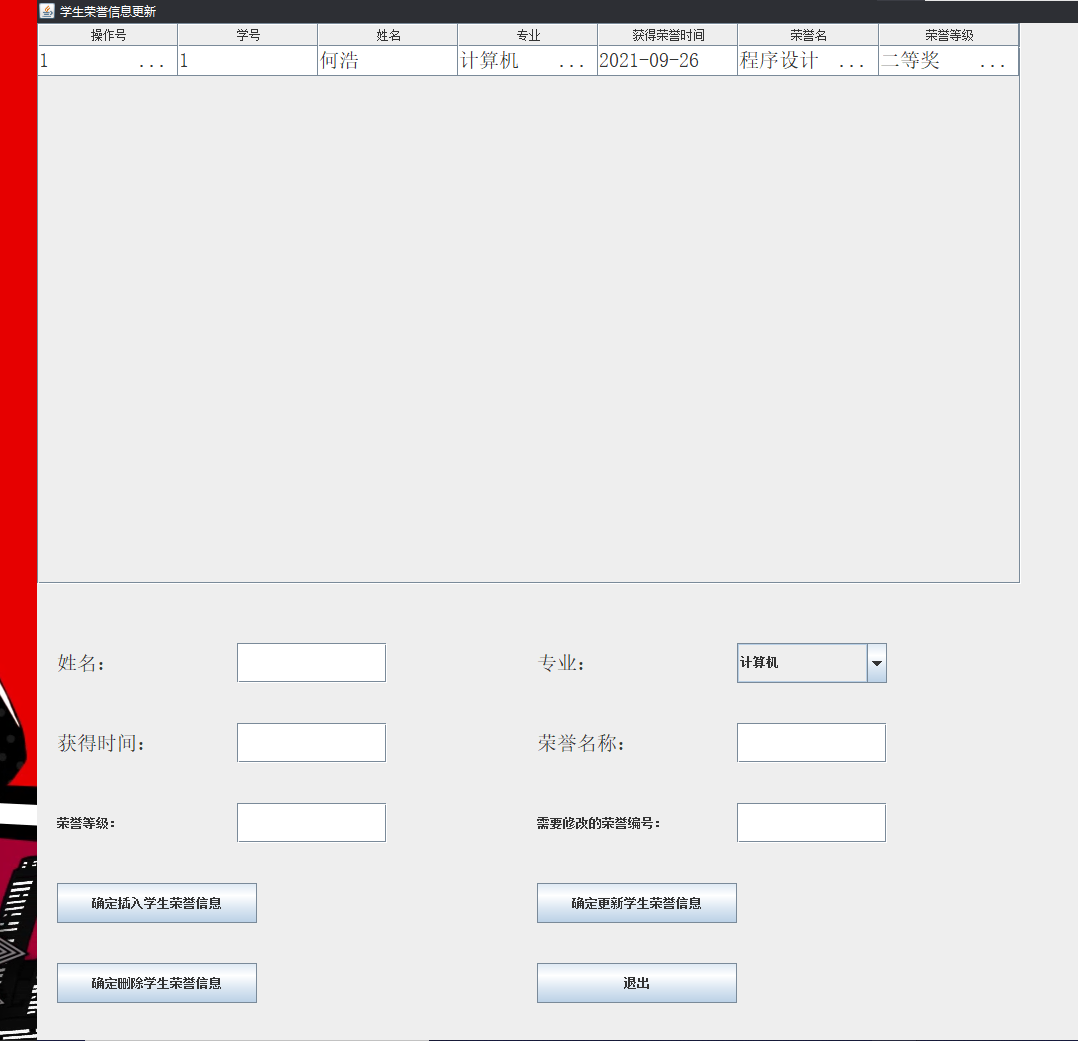
对于学生处分信息的增删改查，其实现原理与前述功能类似，此处仅给出运行截图。



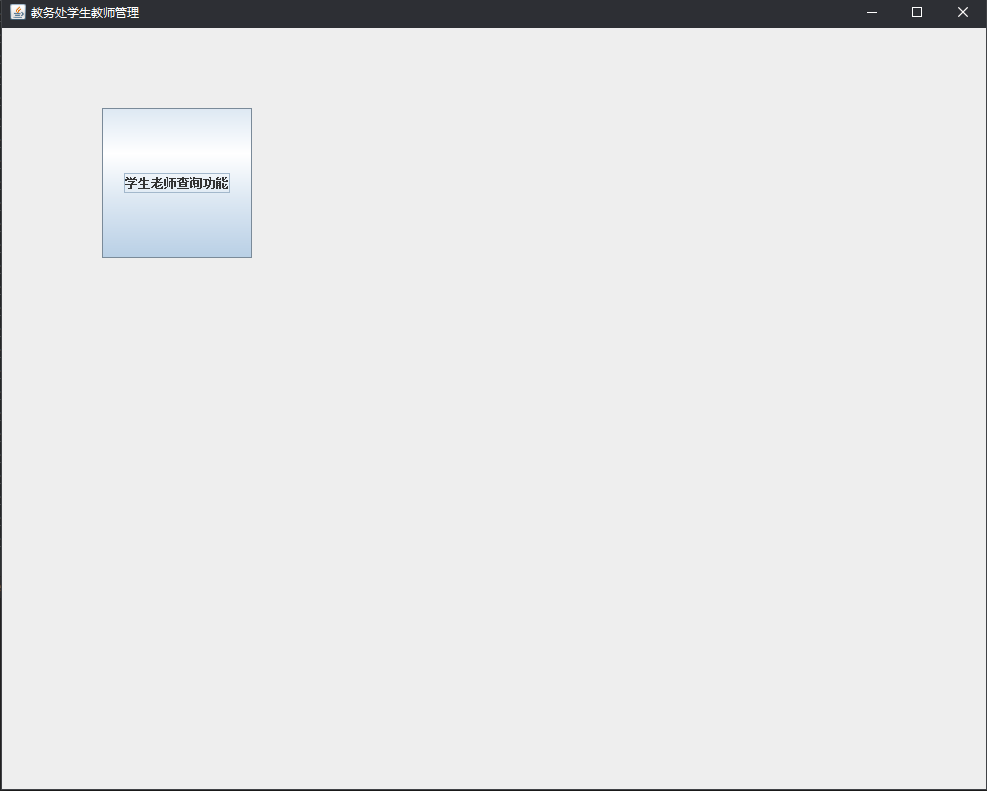


对于学生获奖信息的增删改查，其实现原理与前述功能类似，此处仅给出运行截图。

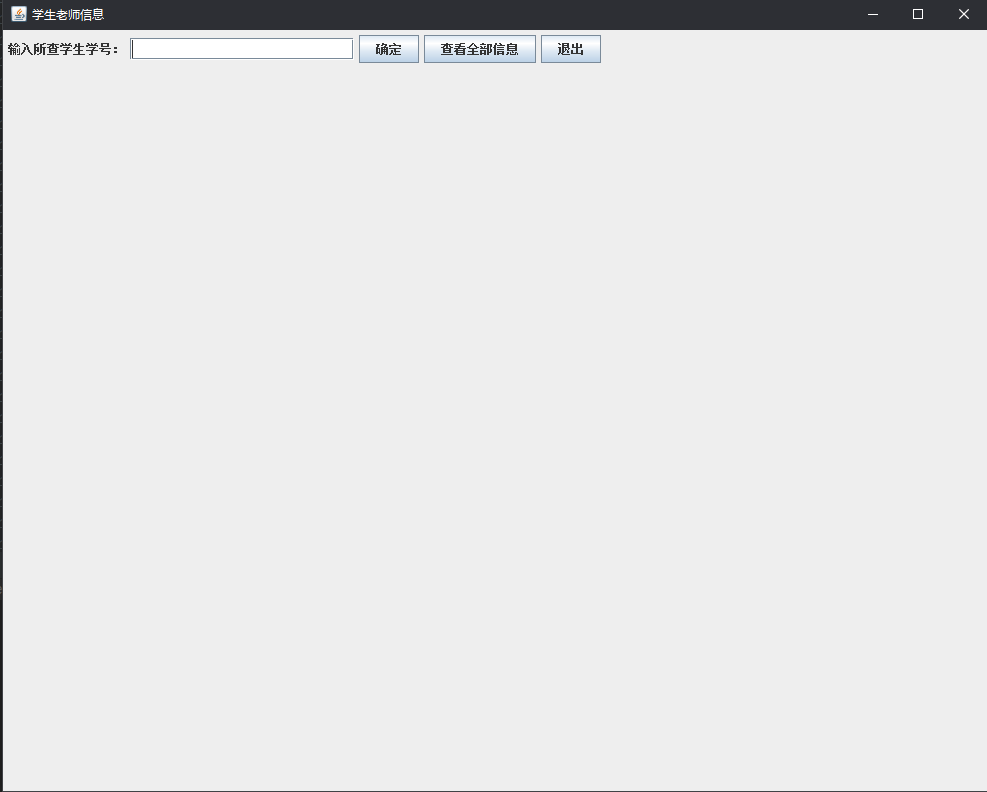




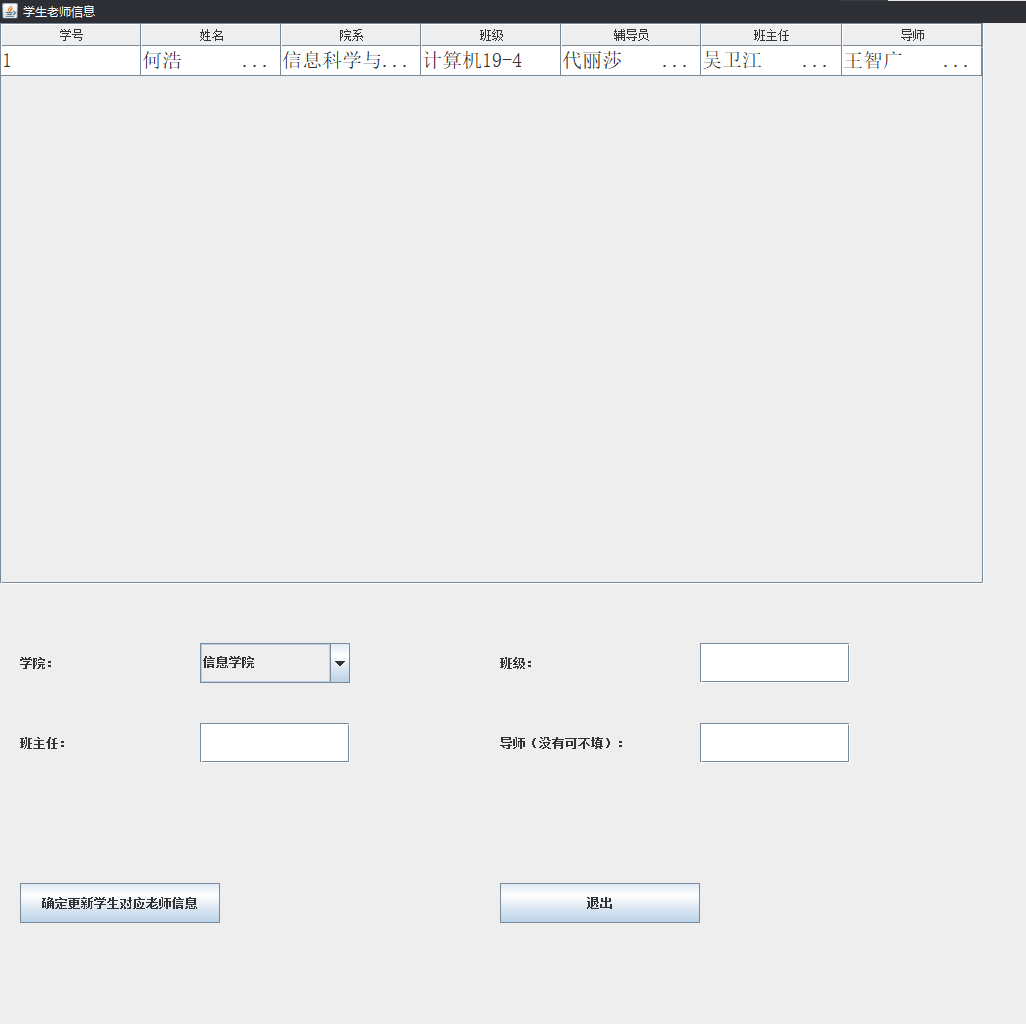
学生老师查询功能主要实现学生导师、辅导员和班主任信息的查询和修改。主要通过SQL语句对学生老师信息表进行操作来实现。



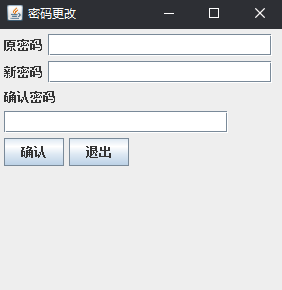
查询界面支持两种管理模式，一是用户输入学生学号，点击确定按钮，即可对单个学生的教师信息进行查询；二是点击“查看所有信息”按钮，在JTable组件中显示学生老师信息表中的全部记录。



教务处管理人员通过文本输入或下拉菜单组件选择表项，Java程序使用对应的方法将获取到的数据传给被调存储过程，由后台数据库存储过程执行对基本表的操作。



另外，教务端还提供了密码更改功能。用户输入原始密码和新密码，并确认密码，即可修改登录信息表中的账号密码。



核心代码如下：

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

con = DriverManager.getConnection(connectionUrl, "sa", "159357");

String SQLs = "select 登录信息表.password from 登录信息表 where userid=" + id;

stmt = con.createStatement();

ResultSet qry = stmt.executeQuery(SQLs);

while(qry.next())

ps=qry.getString(1);

if(ps!=null){

if((txtoldpw.getText()).equals(ps.trim())){

if(txtnewpw.getText().equals(txtcheckpw.getText())){

if(txtnewpw.getText().length()<20 && txtnewpw.getText().length()>=6) {

String SQL = "update 登录信息表 set 登录信息表.password="+txtnewpw.getText()+" where userid="+id;

int result = stmt.executeUpdate(SQL);

if (result == 1)

JOptionPane.showMessageDialog(frame, "修改成功！");

frame.setVisible(false);

}

else

JOptionPane.showMessageDialog(frame, "密码过长或过短");

}

else{

JOptionPane.showMessageDialog(frame, "确认密码与新密码不同");

}

}

else{

JOptionPane.showMessageDialog(frame, "原密码错误");

}

}

else{

JOptionPane.showMessageDialog(frame, "用户密码有误");

}

教务端提供了学生导师、辅导员、班主任的查询功能，其依赖的数据表为学生老师信息表。



插入学生信息的核心代码如下：

文本

已生成极高可信度的说明

# 4.5 数据库存储过程

（1）处分更新

create proc 处分更新(@apln char(10),@id char(10),@name char(10),@pro char(30),@date date,@area char(30),@gra char(30))

as begin

delete from 处分记录表 where Punum=@apln;

insert into 处分记录表 values (@apln,@id,@name,@pro,@date,@area,@gra);

end

（2）学生插入

create proc 插入学生(@id char(10),@name char(20),@pro char(30),@age int,@iden char(10),@teaname char(10),@class char(10),@college char(30),@grade char(10),@cond char(10),@loc char(20),@tui int,@contui real)

as begin

DECLARE @teanum char(10)

set @teanum = (select TeaID from 教师信息表 where Teaname = @teaname)

--insert into 学生信息表 values (@id,@name,@pro,@age,@iden,@teanum,@class,@college,@grade,@cond,@loc,@vol,@tui,@contui,NULL);

insert into 学生信息表 values (@id,@name,@pro,@age,@iden,@teanum,@class,@college,@grade,@cond,@loc,NULL,@tui,@contui,NULL);

insert into 学生老师信息表 values (@id,@name,NULL,@teaname,NULL,@class,@college);

insert into 登录信息表 values (@id,@id+'123456','student');

end

go

# 5、结论

本小组数据库课程设计课题是高校教务管理与学生服务系统，采用前端Java GUI+后端SQL数据库的系统架构，实现了学生、教师、导师、班主任、辅导员、教务处等实体的多种操作，并采用科大讯飞语音识别API实现了成绩语音录入功能。经过测试，教务处管理数据库应用系统界面设计简洁清晰，数据库系统具有较好的安全性和健壮性，能够满足实际的教务管理需要。

# 参考资料

[1]林聃. 呼和浩特乒羽中心综合管理系统设计与实现[D]. 大连理工大学.