数据库课程设计报告

1、系统设计

1.1 需求分析

1.1.1 基本信息和功能需求

一个学校有多个系，每个系有多名教师，每个教师只能属于一个系，一个系可以有多个教师。每个系都有一个教师作为系主任，每个教师只能在一个系当系主任。每个系有多个班级，每个班级都有自己的专业名且只属于一个系。每个班级有多个学生，每个班级都有一个学生担任班长，一个学生只能在一个班级担任班长。有的学生担任指导员，一个指导员可以管理多个学生，每个学生只能有一个指导员。一个系可以有多门课程，每门课可以在多个系开课。一个教师可以讲多门课，一门课可以被多个教师讲。一个学生可以选择多门课，一门课可以被多个学生学。一个学生选定一门课时，就有一个确定的老师。一个学生选定的每一门课都有一个成绩。

针对上述基本信息，本系统设计目标是提高学生选课、学生学习和成绩管理工作的效率，并对学校的信息进行有效管理。主要功能包括以下三个方面：

（1）用户管理，包括对超级管理员、管理员、学生 3 种不同身份的识别及管理。

（2）管理员对学校信息管理，管理员根据用户名和密码登录系统进行包括对系、班级、课程、教师、学生、选课、成绩等数据的管理。

（3）学生对个人信息的查询及管理，学生根据用户名和密码登录系统进行修改密码、浏览课程信息、选课、查询个人信息等操作。

通过本系统管理员可以很方便的对学生个人信息、课程和教师信息、选课信息进行管理和维护。学生可以通过本系统进行个人信息查询、考试成绩查询、课表查询及选课操作。管理员拥有对数据库增删查改的权限，能够发布和查询学生信息、教师信息和课程信息。

1.1.2 数据元素表

针对上面的功能需求，结合实际得出本系统要处理的数据元素表如下：

表 1 数据元素表

数据项名	别名	数据类型	宽度	含义
系名	collegeName	varchar	50	系名
系编号	collegeNo	varchar	50	唯一标识系
班级名	className	varchar	50	班级名
班级编号	classNo	varchar	50	唯一标识班级
班级专业名	classProName	varchar	50	班级专业名
课程名	courseName	varchar	50	课程名
课程编号	courseNo	varchar	50	唯一标识课程
课程学时	coursePeriod	int	4	课程学时
课程学分	courseCredict	int	4	课程学分
教师名	teacherName	varchar	50	教师名
教师编号	teacherNo	varchar	50	唯一标识教师
研究领域	teacherResDir	varchar	50	教师研究领域
学生学号	studentNo	varchar	50	唯一标识学生
学生姓名	studentName	varchar	50	学生姓名
学生年龄	studentAge	varchar	50	学生年龄
学生性别	studentSex	varchar	50	学生性别
学生民族	studentNation	varchar	50	学生民族
学生生日	studentBirthday	varchar	50	学生生日
用户名	userName	varchar	50	唯一标识用户
用户密码	userPassword	varchar	50	用户密码
用户类型	userType	int	4	用户类型
管理员编号	managerNo	varchar	50	唯一标识管理员
管理员姓名	managerName	varchar	50	管理员姓名
管理员电话	managerPhone	varchar	50	管理员电话

1.1.3 数据流图

根据上述信息可以得出数据流图如下：

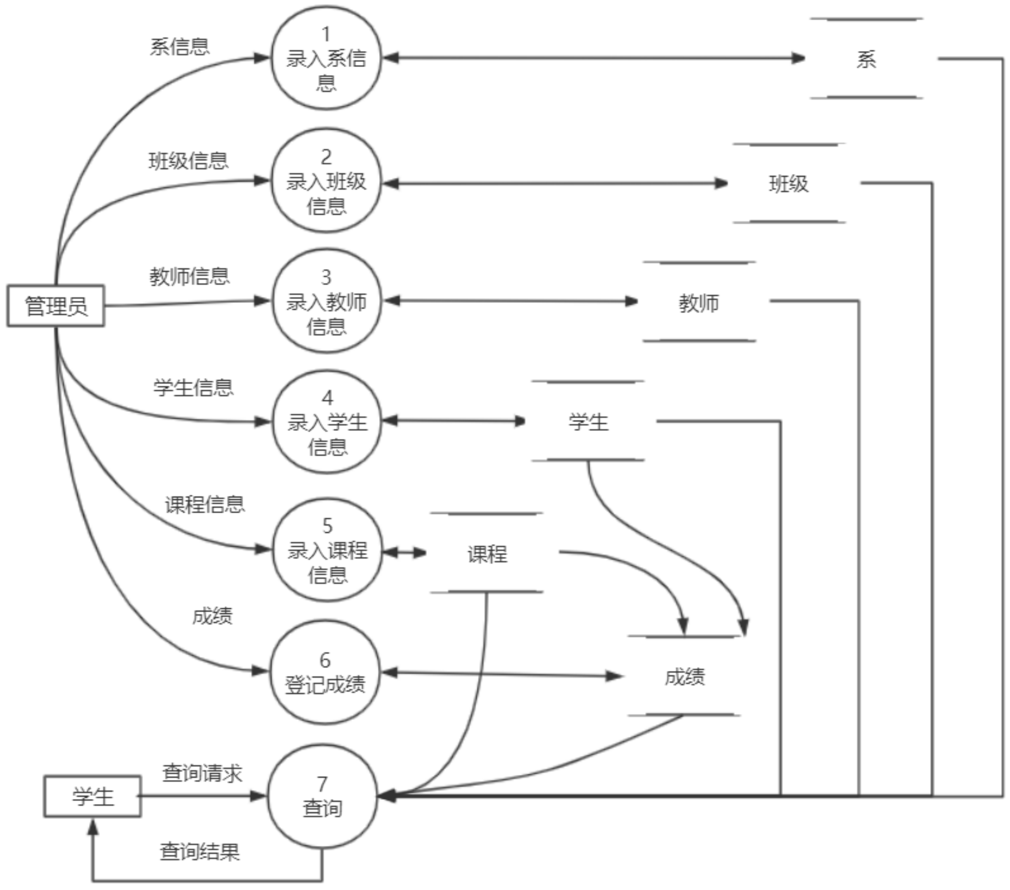
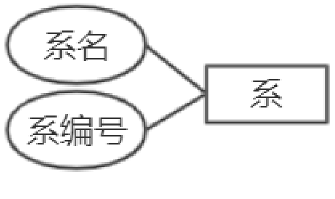


图 1 数据流图

1.2 数据库概念设计

本系统中包括五个实体：系实体、班级实体、课程实体、教师实体、学生实体。根据需求分析的结果以上五个实体所对应的 ER 属性图如下：



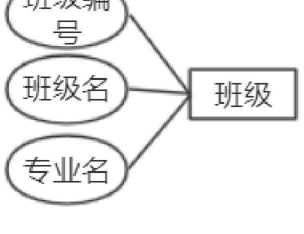
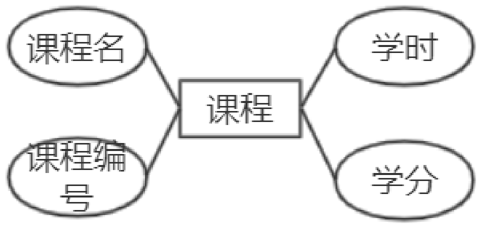


图 1 系实体属性图 **图 2 班级实体属性图**



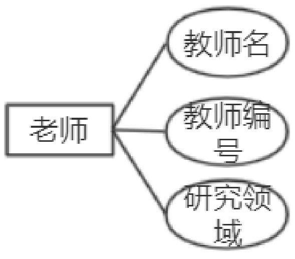


图 3 课程实体属性图 **图 4 老师实体属性图**

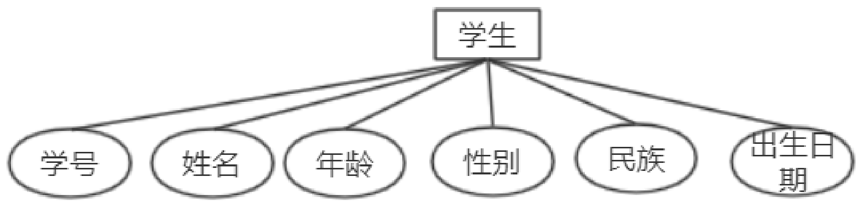


图 5 学生实体属性图

系统整体 ER 图如下：

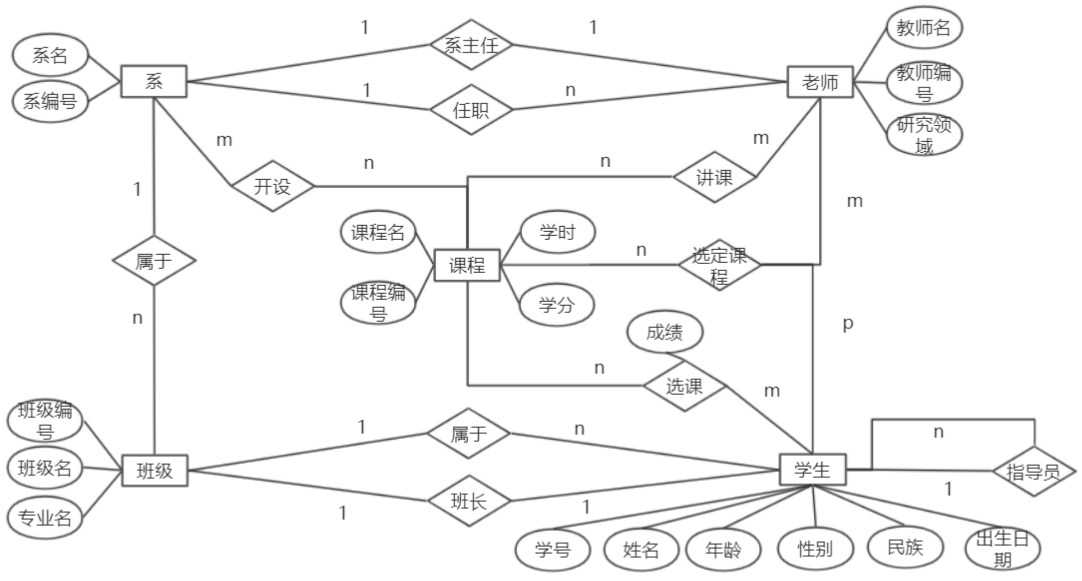


图 6 系统整体 ER 图

1.3 数据库逻辑设计

2.2.1 基本 ER 图向关系模型的转换

根据 ER 图向关系模型转换的原则，一个实体型转换为一个关系模式，关系的属性就是实体的属性，关系的码就是实体的码，因此按照上述的整体 ER 图，本系统的关系模型如下：

系（系编号，系名）

班级（班级编号，班级名，专业名，系编号）

课程（课程编号，课程名，学时，学分）

教师（教师编号，教师名，研究领域，系编号）
学生（学号，姓名，年龄，性别，民族，出生日期，指导员学号，班级编号）

系-系主任表（系编号，系主任教师编号）
班级-班长表（班级编号，班长学号）
学生-课程-教师-成绩表（学号，课程号，教师编号，成绩）
系-课程表（系编号，课程编号）
教师-课程表（教师编号，课程编号）

2.2.2 数据模型优化

（1）分析函数依赖
系表：系编号→系名
班级表：班级编号→班级名、班级编号→专业名、班级编号→系编号
课程表：课程编号→课程名、课程编号→学时、课程编号→学分
教师表：教师编号→教师名、教师编号→研究领域、教师编号→系编号
学生表：学号→姓名、学号→年龄、学号→性别、学号→民族、学号→出生日期、学号→指导员学号、学号→班级编号
系-系主任表：系编号→系主任教师编号
班级-班长表：班级编号→班长学号
学生-课程-教师-成绩表：（学号，课程编号）→教师编号、（学号，课程编号）→成绩
系-课程表：（系编号，课程编号）
教师-课程表：（教师编号，课程编号）

（2）分析传递函数依赖
　　经过分析每个关系的属性间的传递依赖性，可以发现每个关系的每一个非主属性既不部分依赖于码，也不传递依赖于码，因此该数据库所有关系至少拥有3NF。

2、系统实现

2.1 实现环境

开发工具：

前台开发语言为 Java，后台数据库为 SQL SERVER2019。

运行环境：

Java 10.0.2 版本

Windows10

2.2 系统功能结构图

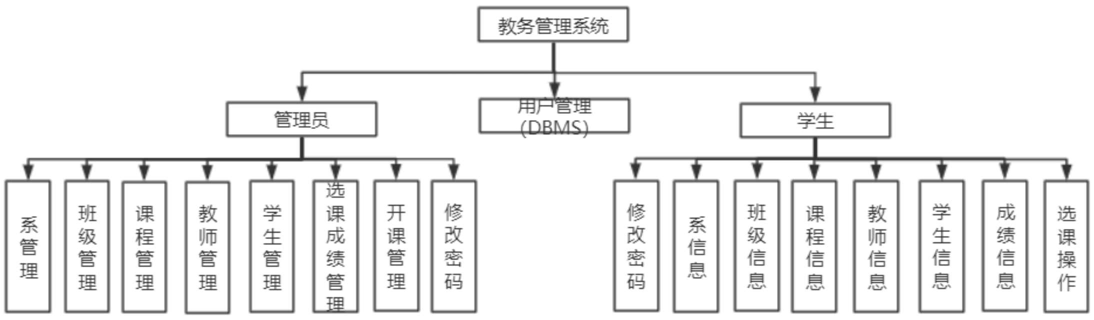


图 7 系统功能结构图

2.3 基本表定义

表 2 用户信息表（Users）

字段名	数据类型	宽度	字段描述
userName	varchar	50	用户登录名
userPassword	varchar	50	用户登录密码
userType	int	4	用户级别

表 3 管理员信息表（Manager）

字段名	数据类型	宽度	字段描述
managerName	varchar	50	管理员名
managerNo	varchar	50	管理员编号
managerPhone	varchar	50	管理员电话

表 4 系信息表（College）

字段名	数据类型	宽度	字段描述
collegeName	varchar	50	系名
collegeNo	varchar	50	系编号

表 5 班级信息表（Class）

字段名	数据类型	宽度	字段描述
className	varchar	50	班级名
classNo	varchar	50	班级编号
classProName	varchar	50	班级专业名
classColNo	varchar	50	班级系编号

表 6 课程信息表（Course）

字段名	数据类型	宽度	字段描述
courseName	varchar	50	课程名
courseNo	varchar	50	课程编号
courseCredict	int	4	学分
coursePeriod	int	4	学时

表 7 教师信息表（Teacher）

字段名	数据类型	宽度	字段描述
teacherNo	varchar	50	教师编号
teacherName	varchar	50	教师名
teacherResDir	varchar	50	教师研究领域
teacherColNo	varchar	50	教师系编号

表 8 学生信息表（Student）

字段名	数据类型	宽度	字段描述
studentNo	varchar	50	学生编号
studentName	varchar	50	学生名字
studentAge	varchar	50	学生年龄
studentSex	varchar	50	学生性别
studentNation	varchar	50	学生民族
studentBirthday	varchar	50	学生生日
studentInsNo	varchar	50	学生指导员学号
studentClassNo	varchar	50	学生班级编号

表 9 系-系主任表（CTTable）

字段名	数据类型	宽度	字段描述
collegeNo	varchar	50	系编号
teacherNo	varchar	50	系主任教师编号

表 10 班级-班长表（CSTable）

字段名	数据类型	宽度	字段描述
classNo	varchar	50	班级编号
studentNo	varchar	50	班长学号

表 11 学生-课程-教师-成绩表（SCTSTable）

字段名	数据类型	宽度	字段描述
studentNo	varchar	50	学生学号
courseNo	varchar	50	课程编号
teacherNo	varchar	50	教师编号
score	int	4	成绩

表 12 系-课程表（CCTable）

字段名	数据类型	宽度	字段描述
collegeNo	varchar	50	系编号
courseNo	varchar	50	课程编号

表 13 教师-课程表（TCTable）

字段名	数据类型	宽度	字段描述
teacherNo	varchar	50	教师编号
courseNo	varchar	50	课程编号

2.4 完整性设计

（1）实体完整性

用户登录名、管理员编号、系编号、班级编号、教师编号、学生编号、课程编号、系编号、班级编号、（学号，课程编号）、（系编号、课程编号）、（教师编号，课程编号）分别是用户表、管理员表、系表、班级表、教师表、学生表、课程表、系-系主任表、班级-班长表、学生-课程-教师-成绩表、系-课程表、教师-课程表的主码。

（2）参照完整性

班级表中的班级系编号是系表的主码、教师表中的教师系编号是系表中的主码、学生表中的指导员学号是学生表的主码、学生表中班级编号是班级表中的主码、系-系主任表中的教师编号是教师表的主码、班级-班长表中的班长学号是学生表的主码、学生-课程-教师-成绩表中的教师编号是教师表的主码。以上外码要么为空要么是参照表中的已有数据。

（3）用户定义完整性

用户登录名默认是管理员编号和学号，用户登录密码初始化为对应的管理员编号和学号。性别只能是男女。学分大于 0 且不超过 5。成绩不小于 0 且不超过 100。用户类型只能是 0、1、2：0 表示超级管理员、1 表示管理员、2 表示学生。

用户名不能为空。

2.5 安全性设计

（1）该系统需要用户进行账号的登录才能使用

（2）通过检测不同的用户种类来给予不同的权限和界面

（3）用户登录自己的账号只能进行属于自己的操作

（4）只有管理员才能对数据库进行增删改

2.6 存储过程代码说明

2.6.1 超级管理员

超级管理员仅用于用户管理。

（1）增加权限

1）增加用户

输入：userName 用户登录名，userPassword 用户登录密码，userType 用户类别

输出：无

效果：在 Users 表插入元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[superAddUser]
    @userName varchar(50),
    @userPassword varchar(50),
    @userType int
as
    insert into Users(userName, userPassword, userType) values (@userName, @userPassword,
@userType)
```

2）增加管理员

输入：managerName 管理员名，managerNo 管理员编号，managerPhone 管理员电话

输出：无

效果：在 Manager 表插入元组

```
USE [EducationManagement]
GO
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[superAddManager]
    @managerName varchar(50),
    @managerNo varchar(50),
    @managerPhone varchar(50)
as
    insert into Manager(managerName, managerNo, managerPhone) values (@managerName,
@managerNo, @managerPhone)
```

（2）删除权限

1）删除用户

输入：userName 用户登录名

输出：无

效果：删除 Users 表中主键为 userName 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[superDeleteUser]
    @userName varchar(50)
as
    delete from Users where userName = @userName
```

2）删除管理员

输入：managerNo 管理员编号

输出：无

效果：删除 Manager 表中主键为 managerNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[superDeleteManager]
    @managerNo varchar(50)
as
    delete from Manager where managerNo = @managerNo
```

（3）更新权限

1）更新用户信息

输入：userName 用户登录名，userPassword 用户登录密码，userType 用户类别

输出：无

效果：更新 Users 表中主键为 userName 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[superUpdateUser]
    @userName varchar(50),
    @userPassword varchar(50),
    @userType int
as
    update Users set userName = @userName, userPassword = @userPassword, userType = 
@userType where userName = @userName
```

2）更新管理员信息

输入：managerName 管理员名，managerNo 管理员编号，managerPhone 管理员电话

输出：无

效果：更新 Manager 表中主键为 managerNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[superUpdateManager]
    @managerName varchar(50),
    @managerNo varchar(50),
    @managerPhone int
as
    update Manager set managerName = @managerName, managerNo = @managerNo, managerPhone =
@managerPhone where managerNo = @managerNo
```

（4）查询权限

1）查询用户表

输入：无

输出：Users 表所有元组

效果：返回 Users 表

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[superSelectUsers]
as
    select * from Users
```

2）查询管理员表

输入：无

输出：Manager 表所有元组

效果：返回 Manager 表

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[superSelectManager]
as
    select * from Manager
```

2.6.2 管理员

管理员拥有增加、删除、修改和查看全部学校信息的权限

（1）增加权限：

1）增加系

输入：collegeNo 系编号、collegeName 系名

输出：无

效果：在 College 表中插入元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[addCollege]
    @collegeNo varchar(50),
    @collegeName varchar(50)
as
    insert into College(collegeNo, collegeName) values (@collegeNo, @collegeName)
```

2）增加班级

输入：classNo 班级编号、className 班级名、classProName 班级专业名、classColNo 班级系编号

输出：无

效果：在 Class 表中插入元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[addClass]
    @classNo varchar(50),
    @className varchar(50),
    @classProName varchar(50),
    @classColNo varchar(50)
as
    insert into Class(classNo, className, classProName, classColNo) values (@classNo,
@className, @classProName, @classColNo)
```

3）增加教师

输入：teacherNo 教师编号、teacherName 教师名、teacherResDir 教师研究领域、teacherColNo 教师系编号

输出：无

效果：在 Teacher 表中插入元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[addTeacher]
    @teacherNo varchar(50),
    @teacherName varchar(50),
    @teacherResDir varchar(50),
    @teacherColNo varchar(50)
as
    insert into Teacher(teacherNo, teacherName,teacherResDir, teacherColNo) values
(@teacherNo, @teacherName, @teacherResDir, @teacherColNo)
```

4）增加学生

输入：studentNo 学生编号、studentName 学生姓名、studentAge 学生年龄、studentSex 学生性别、studentNation 学生民族、studentBirthday 学生出生日期、studentInsNo 学生指导员学号、studentClassNo 学生班级编号

输出：无

效果：在 Student 表中插入元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[addStudent]
    @studentNo varchar(50),
    @studentName varchar(50),
    @studentAge varchar(50),
    @studentSex varchar(50),
    @studentNation varchar(50),
    @studentBirthday varchar(50),
    @studentInsNo varchar(50),
    @studentClassNo varchar(50)
as
    insert into Student(studentNo, studentName,studentAge, studentSex, studentNation,
studentBirthday, studentInsNo, studentClassNo) values (@studentNo,
@studentName,@studentAge, @studentSex, @studentNation, @studentBirthday, @studentInsNo,
@studentClassNo)
```

5）增加课程

输入：courseNo 课程编号、courseName 课程名、courseCredict 学分、coursePeriod 学时

输出：无

效果：在 Course 表中插入元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[addCourse]
    @courseNo varchar(50),
    @courseName varchar(50),
    @courseCredict int,
    @coursePeriod int
as
    insert into Course(courseNo, courseName, courseCredict, coursePeriod) values
(@courseNo, @courseName, @courseCredict, @coursePeriod)
```

6）增加系-系主任

输入：collegeNo 系编号、teacherNo 教师编号

输出：无

效果：在 CTTable 表中插入元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[addCollegeManager]
    @collegeNo varchar(50),
    @teacherNo varchar(50)
as
```

```
    insert into CTTable(collegeNo, teacherNo) values (@collegeNo, @teacherNo)
```

7）增加系-课程

输入：collegeNo 系编号、courseNo 课程编号

输出：无

效果：在 CCTable 表中插入元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[addCollegeCourse]
    @collegeNo varchar(50),
    @courseNo varchar(50)
as
    insert into CCTable(collegeNo, courseNo) values (@collegeNo, @courseNo)
```

8）增加班级-班长

输入：classNo 班级编号、studentNo 学生学号

输出：无

效果：在 CSTable 表中插入元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[addClassMonitor]
    @classNo varchar(50),
    @studentNo varchar(50)
as
    insert into CSTable(classNo, studentNo) values (@classNo, @studentNo)
```

9）增加教师-课程

输入：teacherNo 教师编号、courseNo 课程编号

输出：无

效果：在 TCTable 表中插入元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
```

```
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[addTeacherCourse]
    @teacherNo varchar(50),
    @courseNo varchar(50)
as
    insert into TCTable(teacherNo, courseNo) values (@teacherNo, @courseNo)
```

10）增加学生-课程-教师-成绩

输入：studentNo 学生学号、courseNo 课程编号、teacherNo 教师编号、score 成绩

输出：无

效果：在 SCTSTable 表中插入元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[addStuTeaCouSco]
    @studentNo varchar(50),
    @courseNo varchar(50),
    @teacherNo varchar(50),
    @score int
as
    insert into SCTSTable(studentNo, courseNo,teacherNo, score) values (@studentNo,
@courseNo, @teacherNo, @score)
```

（2）删除权限：

1）删除系

输入：collegeNo 系编号

输出：无

效果：删除 College 表中主键为 collegeNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
ALTER procedure [dbo].[deleteCollege]
    @collegeNo varchar(50)
as
    delete from  College where @collegeNo = collegeNo
```

2）删除班级

输入：classNo 班级编号

输出：无

效果：删除 Class 表中主键为 classNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[deleteClass]
    @classNo varchar(50)
as
    delete from Class where classNo = @classNo
```

3）删除教师

输入：teacherNo 教师编号

输出：无

效果：删除 Teacher 表中主键为 teacherNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[deleteTeacher]
    @teacherNo varchar(50)
as
    delete from Teacher where teacherNo = @teacherNo
```

4）删除学生

输入：studentNo 学生编号

输出：无

效果：删除 Student 表中主键为 studentNo 的元组

```
USE [EducationManagement]
GO
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[deleteStudent]
    @studentNo varchar(50)
as
    delete from Student where studentNo = @studentNo
```

5）删除课程

输入：courseNo 系编号

输出：无

效果：删除 Course 表中主键为 courseNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[deleteCourse]
    @courseNo varchar(50)
as
    delete from Course where courseNo = @courseNo
```

6）删除系-系主任

输入：collegeNo 系编号

输出：无

效果：删除 CTTable 表中主键为 collegeNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[deleteCollegeManager]
    @collegeNo varchar(50)
as
    delete from CTTable where collegeNo = @collegeNo
```

7）删除系-课程

输入：collegeNo 系编号、courseNo 课程编号

输出：无

效果：删除 CCTable 表中主键为（collegeNo，courseNo）的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[deleteCollegeCourse]
    @collegeNo varchar(50),
    @courseNo varchar(50)
as
    delete from CCTable where collegeNo = @collegeNo and courseNo = @courseNo
```

8）删除班级-班长

输入：classNo 班级编号、studentNo 学生学号

输出：无

效果：删除 CSTable 表中主键为 classNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[deleteClassMonitor]
    @classNo varchar(50),
    @studentNo varchar(50)
as
    delete from CSTable where classNo = @classNo and studentNo = @studentNo
```

9）删除教师-课程

输入：teacherNo 教师编号、courseNo 课程编号

输出：无

效果：删除 TCTable 表中主键为（teacherNo,courseNo）的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[deleteTeacherCourse]
    @teacherNo varchar(50),
    @courseNo varchar(50)
```

```
as
    delete from TCTable where teacherNo = @teacherNo and courseNo = @courseNo
```

10）删除学生-课程-教师-成绩

输入：studentNo 学生编号、courseNo 课程编号

输出：无

效果：删除 SCTSTable 表中主键为（studentNo，courseNo）的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[deleteStuCouTeaSco]
    @studentNo varchar(50),
    @courseNo varchar(50)
as
    delete from SCTSTable where studentNo = @studentNo and courseNo = @courseNo
```

（3）更新权限：

1）更新系信息

输入：collegeNo 系编号、collegeName

输出：无

效果：更新 College 表中主键为 collegeNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[updateCollege]
    @collegeNo varchar(50),
    @collegeName varchar(50)
as
    update  College set collegeNo = @collegeNo, collegeName = @collegeName where collegeNo
= @collegeNo
```

2）更新班级信息

输入：classNo 班级编号、className 班级名、classProName 班级专业名、classColNo 班级系编号

输出：无

效果：更新 Class 表中主键为 classNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[updateClass]
	@classNo varchar(50),
	@className varchar(50),
	@classProName varchar(50),
	@classColNo  varchar(50)
as
	update Class set classNo = @classNo , className = @className, classProName = 
@classProName, classColNo = @classColNo where classNo = @classNo
```

3）更新教师信息

输入：teacherNo 教师编号、teacherName 教师名字、teacherResDir 教师研究领域、teacherColNo 教师系编号

输出：无

效果：更新 Teacher 表中主键为 teacherNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[updateTeacher]
	@teacherNo varchar(50),
	@teacherName varchar(50),
	@teacherResDir varchar(50),
	@teacherColNo varchar(50)
as
	update Teacher set teacherNo = @teacherNo , teacherName = @teacherName, teacherResDir 
= @teacherResDir, teacherColNo = @teacherColNo where teacherNo = @teacherNo
```

4）更新学生信息

输入：studentNo 学生学号、studentName 学生姓名、studentAge 学生年龄、studentSex 学生性别、studentNation 学生民族、studentBirthday 学生出生日

期、studentInsNo 学生指导员学号、studentClassNo 学生班级编号

输出：无

效果：更新 Student 表中主键为 studentNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[updateStudent]
	@studentNo varchar(50),
	@studentName varchar(50),
	@studentAge varchar(50),
	@studentSex varchar(50),
	@studentNation	varchar(50),
	@studentBirthday varchar(50),
	@studentInsNo varchar(50),
	@studentClassNo varchar(50)
as
	update Student set studentNo = @studentNo , studentName = @studentName, studentAge = 
@studentAge, studentSex = @studentSex, studentNation = @studentNation, studentBirthday = 
@studentBirthday, studentInsNo = @studentInsNo, studentClassNo = @studentClassNo where 
studentNo = @studentNo
```

5）更新课程信息

输入：courseNo 课程编号、courseName 课程名、courseCredict 学分、coursePeriod 学时

输出：无

效果：更新 Course 表中主键为 courseNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[updateCourse]
	@courseNo varchar(50),
	@courseName varchar(50),
	@courseCredict int,
	@coursePeriod int
as
```

```
    update Course set courseNo = @courseNo , courseName = @courseName, courseCredict = @courseCredict, coursePeriod = @coursePeriod where courseNo = @courseNo
```

6）更新系-系主任

输入：collegeNo 系编号、teacherNo 教师编号

输出：无

效果：更新 CTTable 表中主键为 collegeNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[updateCollegeManager]
    @collegeNo varchar(50),
    @teacherNo varchar(50)
as
    update CTTable set collegeNo = @collegeNo , teacherNo = @teacherNo where collegeNo = @collegeNo
```

7）更新班级-班长

输入：classNo 班级编号、studentNo 学生编号

输出：无

效果：更新 CSTable 表中主键为 classNo 的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[updateClassMonitor]
    @classNo varchar(50),
    @studentNo varchar(50)
as
    update CSTable set classNo = @classNo , studentNo = @studentNo where classNo = @classNo
```

8）更新学生-课程-教师-成绩

输入：studentNo 学生编号、courseNo 课程编号、teacherNo 教师编号、score 成绩

输出：无

效果：更新 SCTSTable 表中主键为（studentNo,courseNo）的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[updateStuCouTeaSco]
    @studentNo varchar(50),
    @courseNo varchar(50),
    @teacherNo varchar(50),
    @score int
as
    update SCTSTable set studentNo = @studentNo, courseNo = @courseNo , teacherNo =
@teacherNo, score = @score where studentNo = @studentNo and courseNo = @courseNo
```

9）修改密码

输入：userName 用户登录名、userOldPassword 用户旧密码、userNewPassword 用户新密码

输出：无

效果：更新 Users 表中用户名和密码与输入的 userName、userOldPassword 相匹配的元组的密码为 userNewPassword

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[updatePassword]
    @userName varchar(50),
    @userOldPassword varchar(50),
    @userNewPassword varchar(50)
as
    update Users set userName = @userName, userPassword = @userNewPassword where userName
= @userName and userPassword = @userOldPassword
```

（4）查询权限：

输入：无

输出：College 表所有元组

效果：返回 College 表

1）查询系表

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[selectCollege]
as
    select * from College
```

2）查询班级表

输入：无

输出：Class 所有元组

效果：返回 Class 表

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[selectClass]
as
    select * from Class
```

3）查询教师表

输入：无

输出：Teacher 表所有元组

效果：返回 Teacher 表

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[selectTeacher]
```

```
as
    select * from Teacher
```

4）查询学生表

输入：无

输出：Student 表所有元组

效果：返回 Student 表

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[selectStudent]
as
    select * from Student
```

5）查询课程表

输入：无

输出：Course 表所有元组

效果：返回 Course 表

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[selectCourse]
as
    select * from Course
```

6）查询系-系主任表

输入：无

输出：CTTable 表所有元组

效果：返回 CTTable 表

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
ALTER procedure [dbo].[selectCollegeManager]
as
    select * from CTTable
```

7）查询系-课程表

输入：无

输出：CCTable 表所有元组

效果：返回 CCTable 表

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[selectCollegeCourse]
as
    select * from CCTable
```

8）查询班级-班长表

输入：无

输出：CSTable 表所有元组

效果：返回 CSTable 表

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[selectClassMonitor]
as
    select * from CSTable
```

9）查询教师-课程表

输入：无

输出：TCTable 表所有元组

效果：返回 TCTable 表

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
```

```
GO
ALTER procedure [dbo].[selectTeacherCourse]
as
    select * from TCTable
```

10）查询学生-课程-教师-成绩表

输入：无

输出：SCTSTable 表所有元组

效果：返回 SCTSTable 表

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[selectStuCouTeaSco]
as
    select * from SCTSTable
```

2.6.3 学生

学生拥有修改登录密码、选课、退课、查找个人相关信息的权限。

（1）选课

输入：studentNo 学生学号、courseNo 课程编号、teacherNo 教师编号、score 成绩

输出：无

效果：在 SCTSTable 中插入元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[studentAddSCTTable]
    @studentNo varchar(50),
    @courseNo varchar(50),
    @teacherNo varchar(50),
    @score int
as
   insert SCTTable(studentNo , courseNo , teacherNo , score) values (@studentNo , @courseNo ,
@teacherNo ,  @score)
```

（2）退课

输入：studentNo 学生学号、courseNo 课程编号

输出：无

效果：删除 SCTSTable 中插入主键为（studentNo,courseNo）的元组

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[studentDeleteSCTTable]
    @studentNo varchar(50),
    @courseNo varchar(50)
as
    delete from SCTTable where studentNo = @studentNo and courseNo = @courseNo
```

（3）修改登录密码

输入：userName 用户登录名、userOldPassword 用户旧密码、userNewPassword 用户新密码

输出：无

效果：更新 Users 表中用户名和密码与输入的 userName、userOldPassword 相匹配的元组的密码为 userNewPassword

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[updatePassword]
    @userName varchar(50),
    @userOldPassword varchar(50),
    @userNewPassword varchar(50)
as
```

```
    update Users set userName = @userName, userPassword = @userNewPassword where userName = @userName and userPassword = @userOldPassword
```

（4）查找个人相关信息

1）查找个人所在系信息

输入：studentNo 学生学号

输出：该学生所在系的有关信息

效果：返回该学生所在系的有关信息，包括系编号、系名、系主任教师编号、系主任名

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[studentSelectCollege]
    @studentNo varchar(50)
as
    declare @collegeNo varchar(50);
    set @collegeNo = (select classColNo from Class where classNo =
            (select studentClassNo from Student where studentNo = @studentNo)
                )
    select col.collegeNo, col.collegeName, coltea.teacherNo, tea.teacherName from
    (
        select * from College where collegeNo = @collegeNo
    )as col,
    (
        select * from CTTable where collegeNo = @collegeNo
    )as coltea,
    (
        select * from Teacher
    )as tea
    where coltea.teacherNo = tea.teacherNo
```

2）查找个人所在班级信息

输入：studentNo 学生学号

输出：该学生所在班级的有关信息

效果：返回该学生所在班级的有关信息，包括班级编号、班级名、班级专业名、班级所在系编号、班长学号、班长姓名

```
USE [EducationManagement]
GO
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[studentSelectClass]
    @studentNo varchar(50)
as
    declare @classNo varchar(50);
    set @classNo =   (select studentClassNo from Student where studentNo = @studentNo);

   select cla.classNo, cla.className, cla.classProName,cla.classColNo, clamon.studentNo ,
stu.studentName from
    (
        select * from Class where classNo = @classNo
    )as cla,
    (
        select * from CSTable where classNo = @classNo
    )as clamon,
    (
        select * from Student
    )as stu
    where stu.studentNo = clamon.studentNo
```

3）查找个人课表信息

输入：studentNo 学生学号

输出：该学生已选的所有课的信息

效果：返回该学生已选的所有课

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[studentSelectCourse]
    @studentNo varchar(50)
as
    select SCTSTable.courseNo, SCTSTable.teacherNo, Course.courseName,
Course.courseCredict,Course.coursePeriod from SCTSTable join Course on Course.courseNo =
SCTSTable.courseNo where studentNo = @studentNo
```

4）查找所有课程信息

输入：无

输出：所有课程信息

效果：返回所有课程信息

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[selectCourse]
as
    select * from Course
```

5）查找所有教师信息

输入：无

输出：所有教师信息

效果：返回所有教师信息

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[selectTeacher]
as
    select * from Teacher
```

6）查找个人信息

输入：无

输出：个人所有信息

效果：返回个人所有信息

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[studentSelectMessage]
    @studentNo varchar(50)
as
    select * from Student where studentNo = @studentNo
```

7）查找个人成绩信息

输入：无

输出：个人所有成绩信息

效果：返回个人所有成绩信息

```
USE [EducationManagement]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [dbo].[studentSelectScore]
    @studentNo varchar(50)
as
    select * from SCTSTable where studentNo = @studentNo
```

2.7 主要技术

2.7.1 java 程序连接数据库服务器

（1）下载 SQLServer 对应的 JDBC 驱动

（2）将 JDBC 驱动导入项目中，并通过 URL 连接数据库

```
public static final String DRIVER_NAME = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
public static final String DBURL = "jdbc:sqlserver://localhost;database=EducationManagement";
public static final String ROOT_NAME = "Idea";
public static final String PWD = "buaa1701";


try {
    Class.forName(ConstantUtils.DRIVER_NAME);
    System.out.println("数据库驱动加载成功");
    this.connection = DriverManager.getConnection(ConstantUtils.DBURL, ConstantUtils.ROOT_NAME, ConstantUtils.PWD);
    System.out.println("数据库连接成功");
} catch (Exception e) {
    e.printStackTrace();
    System.out.println("数据库连接失败");
}
```

2.7.2 java 程序调用数据库执行 sql 语句

先通过 java.sql 包的 connection 创建语句对象 stm，然后调用 stm 的方法 executeQuery 执行输入的 sql 语句，并通过 stm 的方法 getResultSet 等获取语句运行结果。

```
try {
    Statement stm = connection.createStatement();
    ResultSet resultSet = stm.executeQuery(sql);
    System.out.println("SqlServer : SQL Query 成功");
    return resultSet;
} catch (SQLException e) {
    e.printStackTrace();
    System.out.println("SqlServer : SQL Query 失败" + e.toString());
    return null;
}
```

2.7.3 java Swing 生成 GUI 界面

Swing 是一个为 Java 设计的 GUI 工具包，包含了图形用户界面器件如文本框，按钮，分隔窗格和表等。其基本组件和简介如下：

JFrame　–　java 的 GUI 程序的基本思路是以 JFrame 为基础，它是屏幕上 window 的对象，能够最大化、最小化、关闭。

JPanel　–　Java 图形用户界面(GUI)工具包 swing 中的面板容器类，包含在 javax.swing 包中，可以进行嵌套，功能是对窗体中具有相同逻辑功能的组件进行组合，是一种轻量级容器，可以加入到 JFrame 窗体中。

JLabel　–　JLabel 对象可以显示文本、图像或同时显示二者。可以通过设置垂直和水平对齐方式，指定标签显示区中标签内容在何处对齐。默认情况下，标签在其显示区内垂直居中对齐。默认情况下，只显示文本的标签是开始边对齐；而只显示图像的标签则水平居中对齐。

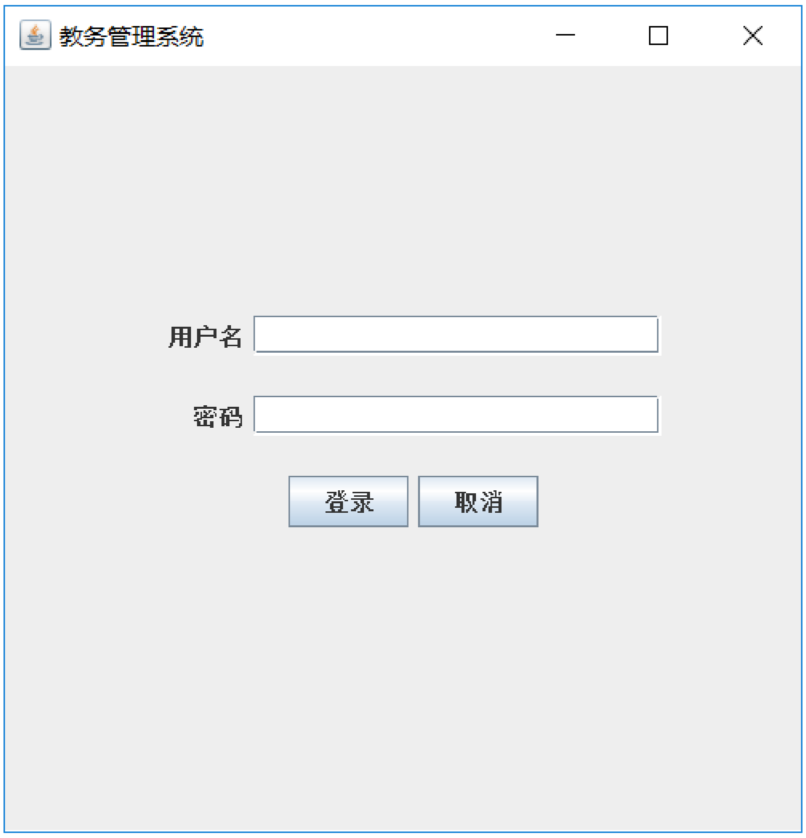
JTextField -一个轻量级组件，它允许编辑单行文本。

JPasswordField - 允许我们输入了一行字像输入框，但隐藏星号(*) 或点创建密码(密码)

JButton - JButton 类的实例。用于创建按钮类似实例中的 "Login"。

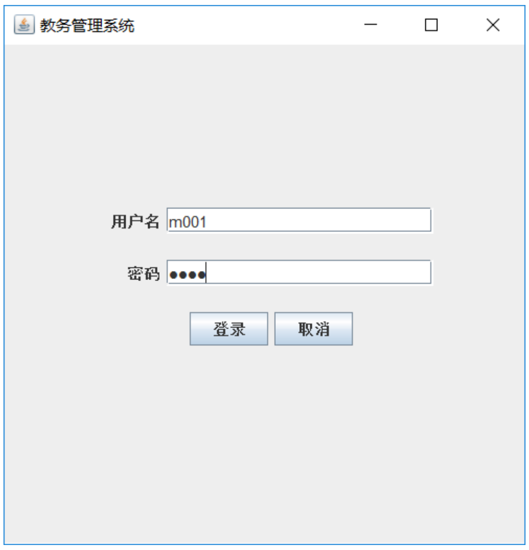
样例程序及运行结果如下：

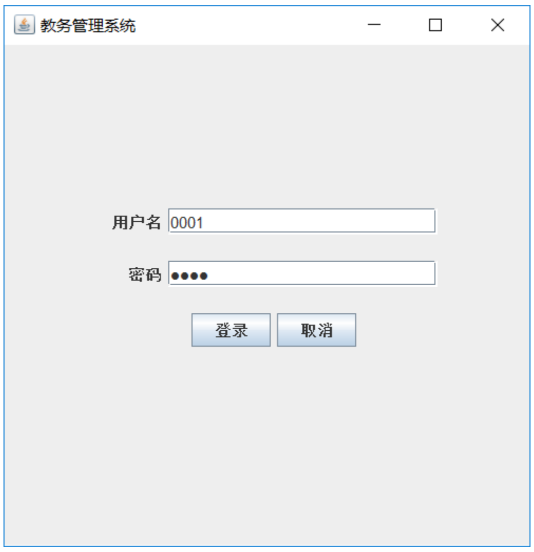
```
containerFrame = new MyFrame(ConstantUtils.LOGIN_X, ConstantUtils.LOGIN_Y, ConstantUtils.LOGIN_WIDTH, ConstantUtils.LOGIN_HEIGH - 200);
    userNamePanel = new JPanel();
    userPasswordPanel = new JPanel();
    loginPanel = new JPanel();

    userNameLabel = new JLabel("用户名");
    userPasswordLabel = new JLabel("    密码");

    userNameField = new JTextField(20);
    userPasswordField = new JPasswordField(20);//密码输入框

    loginButton = new JButton("登录");
    cancelButton = new JButton("取消");

    message = new JLabel();
    message.setForeground(Color.red);
    message.setText("登录失败");

    userNamePanel.add(userNameLabel);
    userNamePanel.add(userNameField);
    userPasswordPanel.add(userPasswordLabel);
    userPasswordPanel.add(userPasswordField);

    loginPanel.add(loginButton);
    loginPanel.add(cancelButton);

    containerFrame.add(userNamePanel);
    containerFrame.add(userPasswordPanel);
    containerFrame.add(loginPanel);
    containerFrame.add(message);
    JPanel jPanel = new JPanel();
    containerFrame.add(jPanel);


    userNamePanel.setBounds(20, 120, 370, 40);
    userPasswordPanel.setBounds(20, 160, 370, 40);
    loginPanel.setBounds(20, 200, 370, 40);
    message.setBounds(20, 240, 370, 40);

    message.setVisible(false);

    containerFrame.setVisible(true);
}
```


教务管理系统
用户名
密码
登录
取消

2.8 系统功能运行实例

2.8.1 登录功能

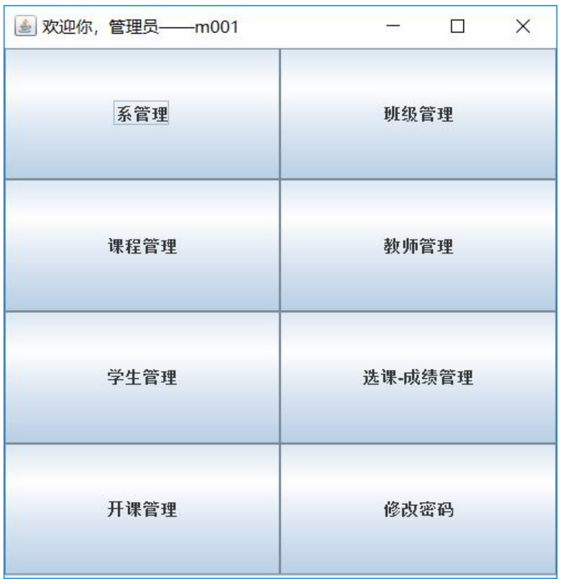




打开程序后会进入如图的登录界面，可以使用管理员账号或学生账号登陆，登录后会进入对应的管理员界面或学生界面。

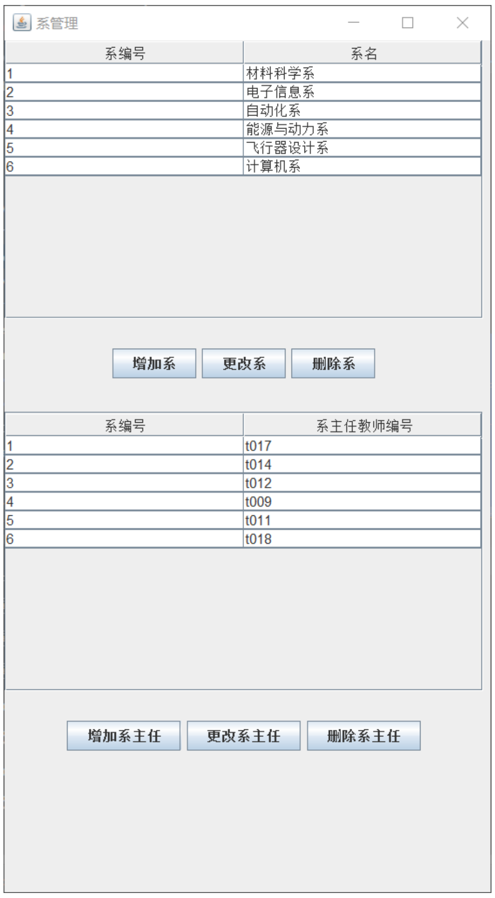
2.8.2 管理员功能

管理员登录后，会进入如下的管理员界面，管理员的功能有系管理、班级管理、课程管理、教师管理、学生管理、选课-成绩管理、开课管理和修改密码。



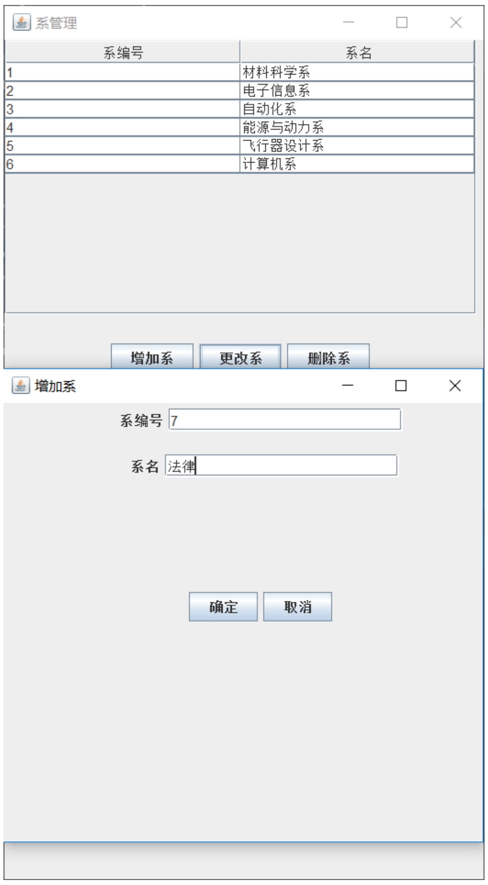
（1）系管理

按下“系管理”按钮后，会出现如下的系管理界面，管理员可以在这个界面进行系的增删改以及系-系主任的增删改操作。



系编号	系名
1	材料科学系
2	电子信息系
3	自动化系
4	能源与动力系
5	飞行器设计系
6	计算机系

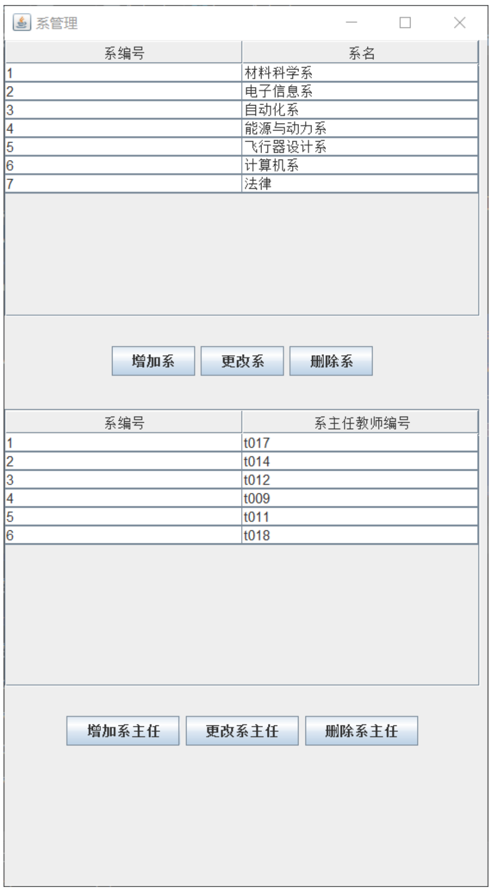
系编号	系主任教师编号
1	t017
2	t014
3	t012
4	t009
5	t011
6	t018



系编号	系名
1	材料科学系
2	电子信息系
3	自动化系
4	能源与动力系
5	飞行器设计系
6	计算机系

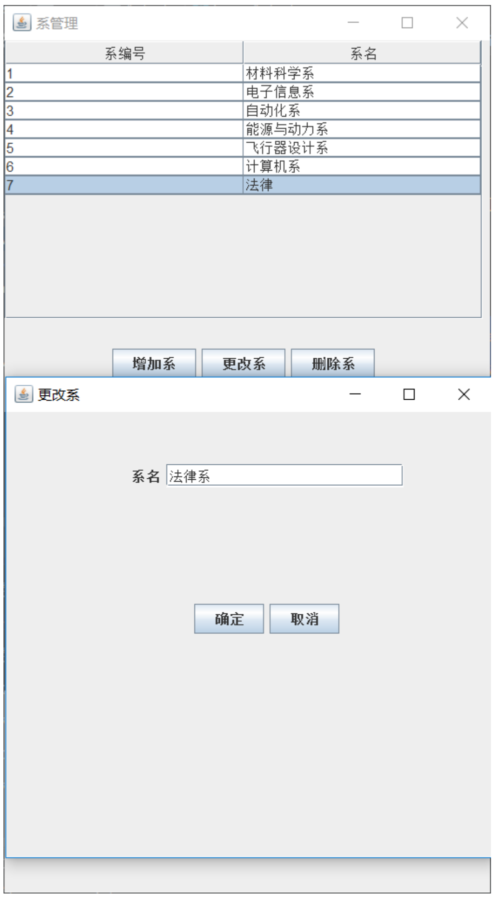
系编号 7

系名 法律



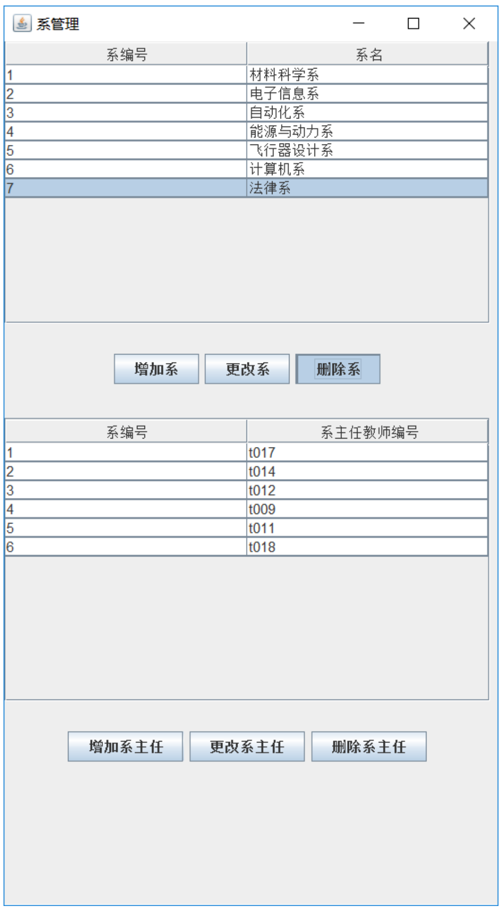
系编号	系名
1	材料科学系
2	电子信息系
3	自动化系
4	能源与动力系
5	飞行器设计系
6	计算机系
7	法律

系编号	系主任教师编号
1	t017
2	t014
3	t012
4	t009
5	t011
6	t018



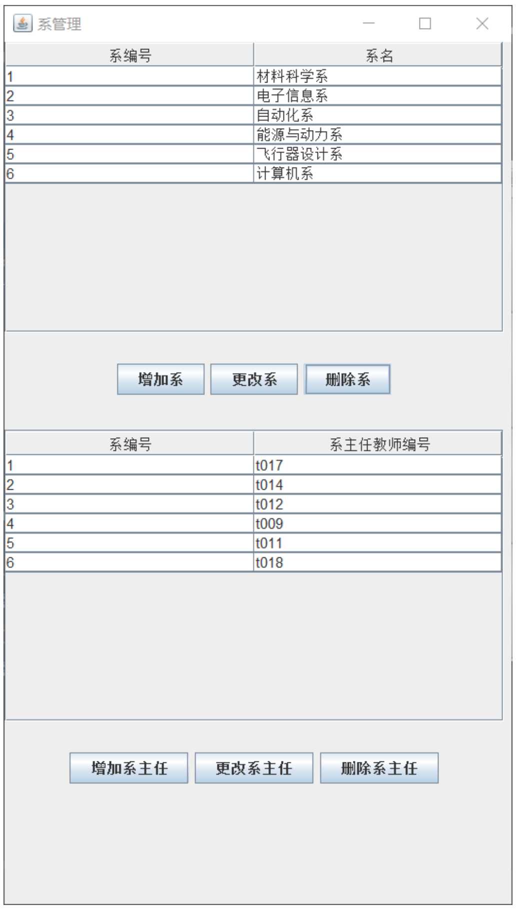
系编号	系名
1	材料科学系
2	电子信息系
3	自动化系
4	能源与动力系
5	飞行器设计系
6	计算机系
7	法律

系名 法律系



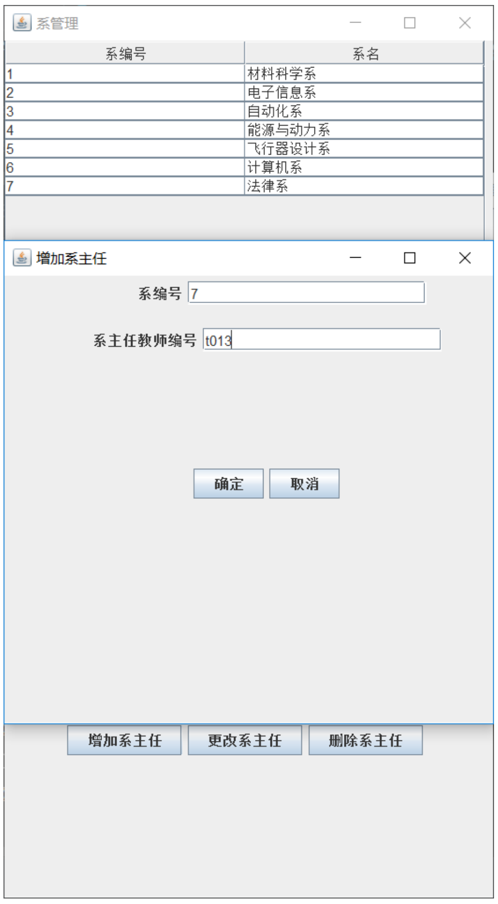
系编号	系名
1	材料科学系
2	电子信息系
3	自动化系
4	能源与动力系
5	飞行器设计系
6	计算机系
7	法律系

系编号	系主任教师编号
1	t017
2	t014
3	t012
4	t009
5	t011
6	t018



系编号	系名
1	材料科学系
2	电子信息系
3	自动化系
4	能源与动力系
5	飞行器设计系
6	计算机系

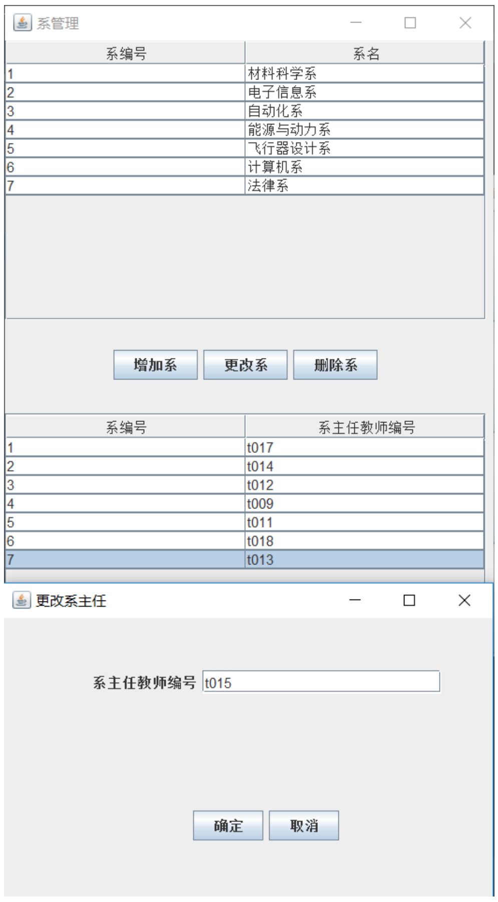
系编号	系主任教师编号
1	t017
2	t014
3	t012
4	t009
5	t011
6	t018



系编号	系名
1	材料科学系
2	电子信息系
3	自动化系
4	能源与动力系
5	飞行器设计系
6	计算机系
7	法律系

系编号 7

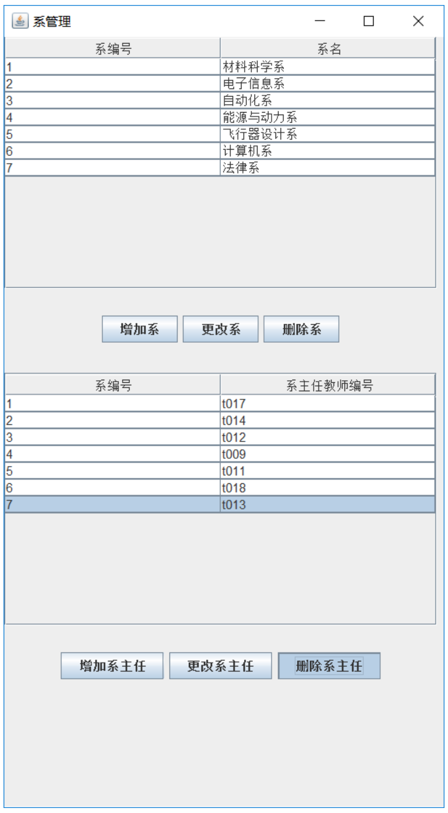
系主任教师编号 t013



系编号	系名
1	材料科学系
2	电子信息系
3	自动化系
4	能源与动力系
5	飞行器设计系
6	计算机系
7	法律系

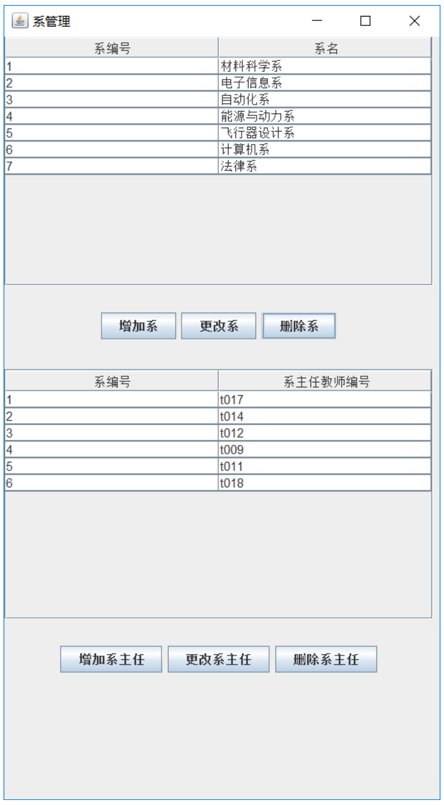
系编号	系主任教师编号
1	t017
2	t014
3	t012
4	t009
5	t011
6	t018
7	t013

系主任教师编号 t015



系编号	系名
1	材料科学系
2	电子信息系
3	自动化系
4	能源与动力系
5	飞行器设计系
6	计算机系
7	法律系

系编号	系主任教师编号
1	t017
2	t014
3	t012
4	t009
5	t011
6	t018
7	t013

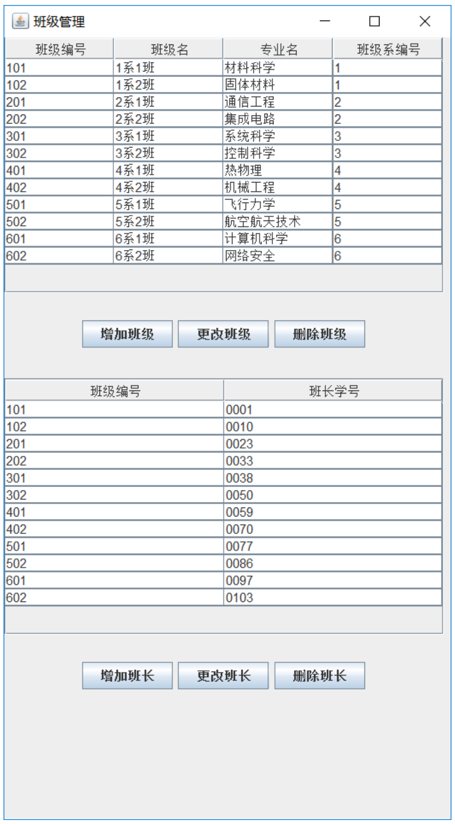


系编号	系名
1	材料科学系
2	电子信息系
3	自动化系
4	能源与动力系
5	飞行器设计系
6	计算机系
7	法律系

系编号	系主任教师编号
1	t017
2	t014
3	t012
4	t009
5	t011
6	t018

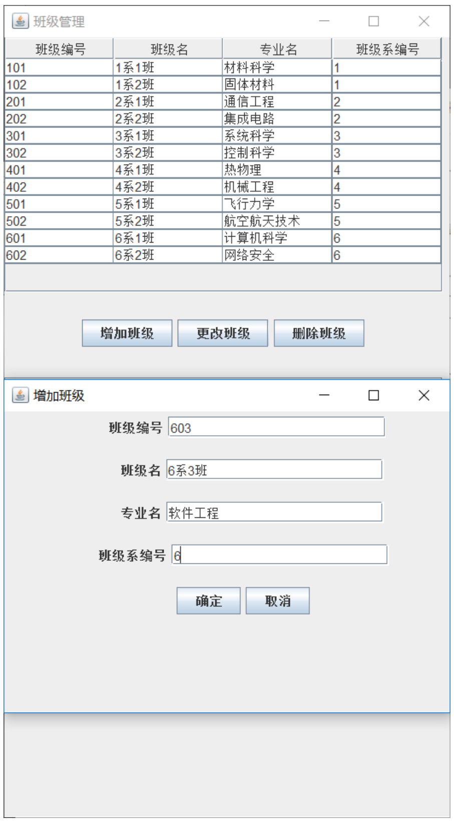
（2）班级管理

按下“班级管理”按钮后，会出现如下的班级管理界面，管理员可以在这个界面进行班级的增删改以及班级-班长的增删改操作。



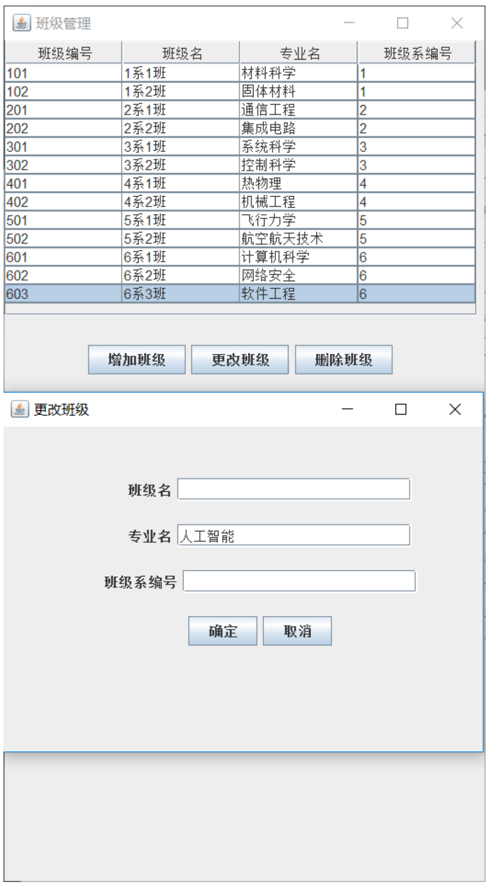
班级编号	班级名	专业名	班级系编号
101	1系1班	材料科学	1
102	1系2班	固体材料	1
201	2系1班	通信工程	2
202	2系2班	集成电路	2
301	3系1班	系统科学	3
302	3系2班	控制科学	3
401	4系1班	热物理	4
402	4系2班	机械工程	4
501	5系1班	飞行力学	5
502	5系2班	航空航天技术	5
601	6系1班	计算机科学	6
602	6系2班	网络安全	6

班级编号	班长学号
101	0001
102	0010
201	0023
202	0033
301	0038
302	0050
401	0059
402	0070
501	0077
502	0086
601	0097
602	0103



班级编号	班级名	专业名	班级系编号
101	1系1班	材料科学	1
102	1系2班	固体材料	1
201	2系1班	通信工程	2
202	2系2班	集成电路	2
301	3系1班	系统科学	3
302	3系2班	控制科学	3
401	4系1班	热物理	4
402	4系2班	机械工程	4
501	5系1班	飞行力学	5
502	5系2班	航空航天技术	5
601	6系1班	计算机科学	6
602	6系2班	网络安全	6

班级编号	603
班级名	6系3班
专业名	软件工程
班级系编号	6

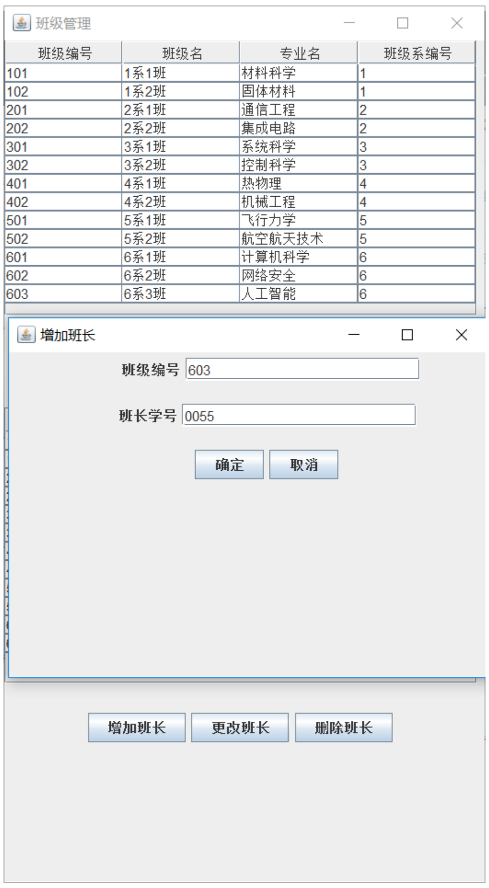


班级编号	班级名	专业名	班级系编号
101	1系1班	材料科学	1
102	1系2班	固体材料	1
201	2系1班	通信工程	2
202	2系2班	集成电路	2
301	3系1班	系统科学	3
302	3系2班	控制科学	3
401	4系1班	热物理	4
402	4系2班	机械工程	4
501	5系1班	飞行力学	5
502	5系2班	航空航天技术	5
601	6系1班	计算机科学	6
602	6系2班	网络安全	6
603	6系3班	软件工程	6

班级名

专业名 人工智能

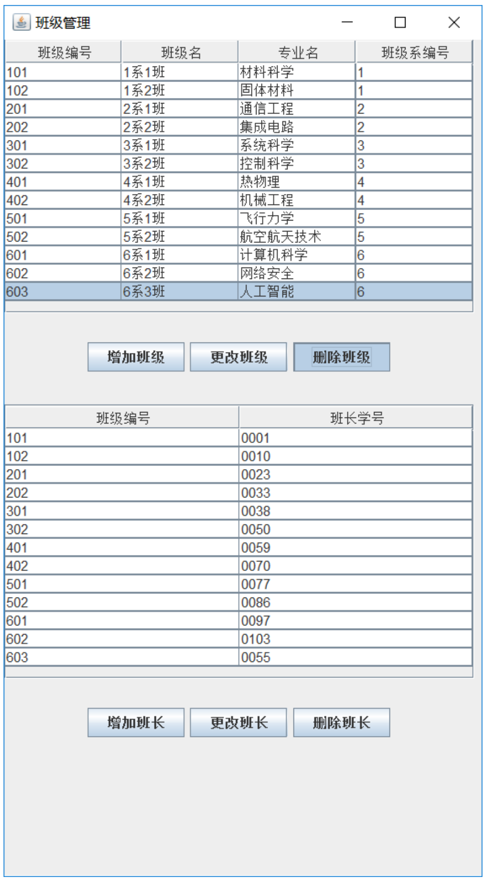
班级系编号



班级编号	班级名	专业名	班级系编号
101	1系1班	材料科学	1
102	1系2班	固体材料	1
201	2系1班	通信工程	2
202	2系2班	集成电路	2
301	3系1班	系统科学	3
302	3系2班	控制科学	3
401	4系1班	热物理	4
402	4系2班	机械工程	4
501	5系1班	飞行力学	5
502	5系2班	航空航天技术	5
601	6系1班	计算机科学	6
602	6系2班	网络安全	6
603	6系3班	人工智能	6

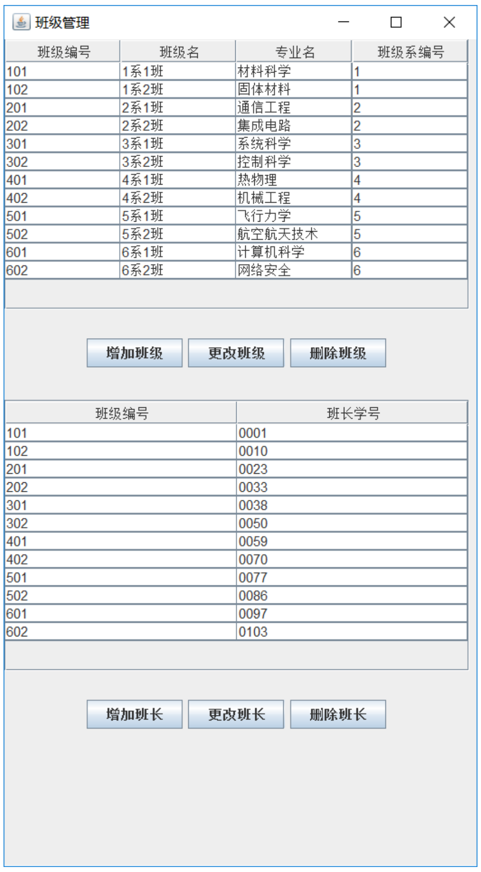
班级编号 603

班长学号 0055



班级编号	班级名	专业名	班级系编号
101	1系1班	材料科学	1
102	1系2班	固体材料	1
201	2系1班	通信工程	2
202	2系2班	集成电路	2
301	3系1班	系统科学	3
302	3系2班	控制科学	3
401	4系1班	热物理	4
402	4系2班	机械工程	4
501	5系1班	飞行力学	5
502	5系2班	航空航天技术	5
601	6系1班	计算机科学	6
602	6系2班	网络安全	6
603	6系3班	人工智能	6

班级编号	班长学号
101	0001
102	0010
201	0023
202	0033
301	0038
302	0050
401	0059
402	0070
501	0077
502	0086
601	0097
602	0103
603	0055

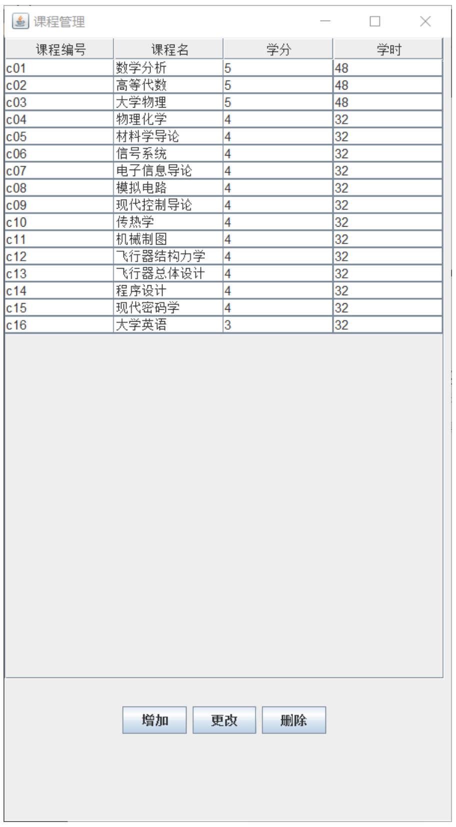


班级编号	班级名	专业名	班级系编号
101	1系1班	材料科学	1
102	1系2班	固体材料	1
201	2系1班	通信工程	2
202	2系2班	集成电路	2
301	3系1班	系统科学	3
302	3系2班	控制科学	3
401	4系1班	热物理	4
402	4系2班	机械工程	4
501	5系1班	飞行力学	5
502	5系2班	航空航天技术	5
601	6系1班	计算机科学	6
602	6系2班	网络安全	6

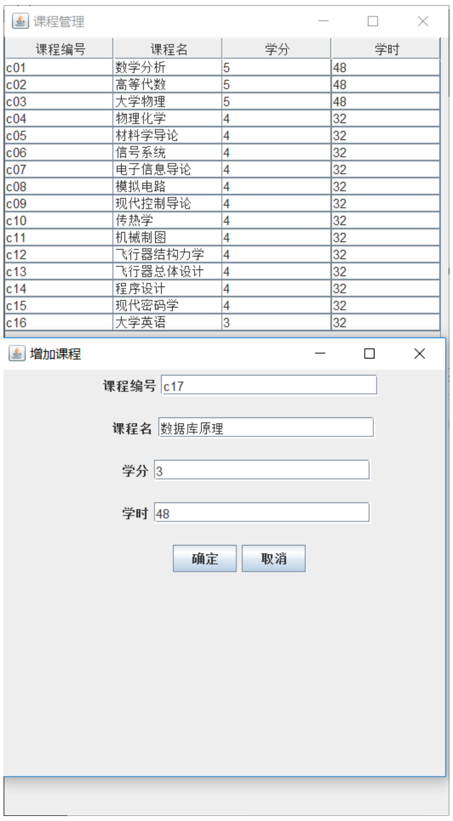
班级编号	班长学号
101	0001
102	0010
201	0023
202	0033
301	0038
302	0050
401	0059
402	0070
501	0077
502	0086
601	0097
602	0103

（3）课程管理

按下“课程管理”按钮后，会出现如下的课程管理界面，管理员可以在这个界面进行课程的增删改操作。

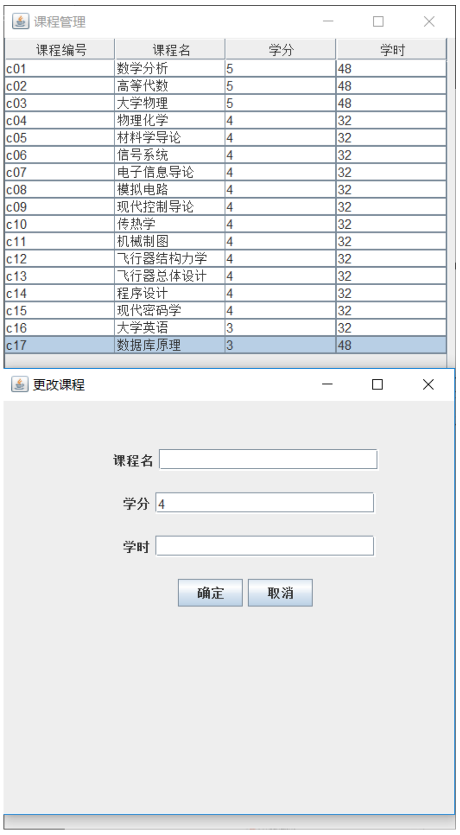


课程编号	课程名	学分	学时
c01	数学分析	5	48
c02	高等代数	5	48
c03	大学物理	5	48
c04	物理化学	4	32
c05	材料学导论	4	32
c06	信号系统	4	32
c07	电子信息导论	4	32
c08	模拟电路	4	32
c09	现代控制导论	4	32
c10	传热学	4	32
c11	机械制图	4	32
c12	飞行器结构力学	4	32
c13	飞行器总体设计	4	32
c14	程序设计	4	32
c15	现代密码学	4	32
c16	大学英语	3	32



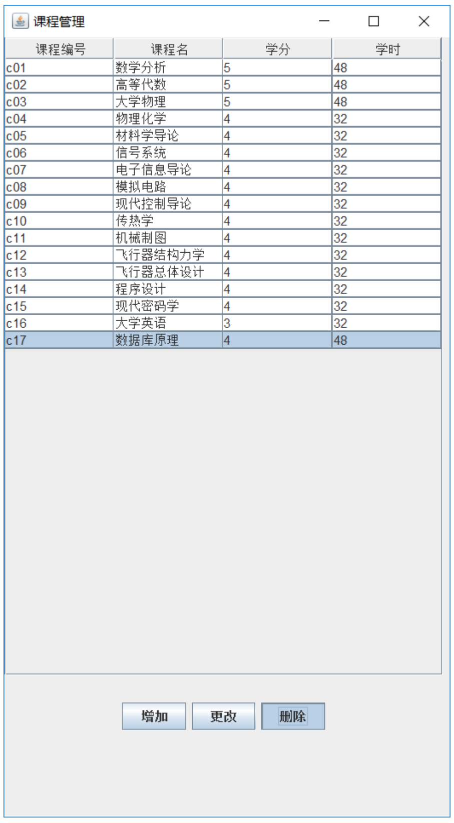
课程编号	课程名	学分	学时
c01	数学分析	5	48
c02	高等代数	5	48
c03	大学物理	5	48
c04	物理化学	4	32
c05	材料学导论	4	32
c06	信号系统	4	32
c07	电子信息导论	4	32
c08	模拟电路	4	32
c09	现代控制导论	4	32
c10	传热学	4	32
c11	机械制图	4	32
c12	飞行器结构力学	4	32
c13	飞行器总体设计	4	32
c14	程序设计	4	32
c15	现代密码学	4	32
c16	大学英语	3	32

课程编号 c17
课程名 数据库原理
学分 3
学时 48



课程编号	课程名	学分	学时
c01	数学分析	5	48
c02	高等代数	5	48
c03	大学物理	5	48
c04	物理化学	4	32
c05	材料学导论	4	32
c06	信号系统	4	32
c07	电子信息导论	4	32
c08	模拟电路	4	32
c09	现代控制导论	4	32
c10	传热学	4	32
c11	机械制图	4	32
c12	飞行器结构力学	4	32
c13	飞行器总体设计	4	32
c14	程序设计	4	32
c15	现代密码学	4	32
c16	大学英语	3	32
c17	数据库原理	3	48

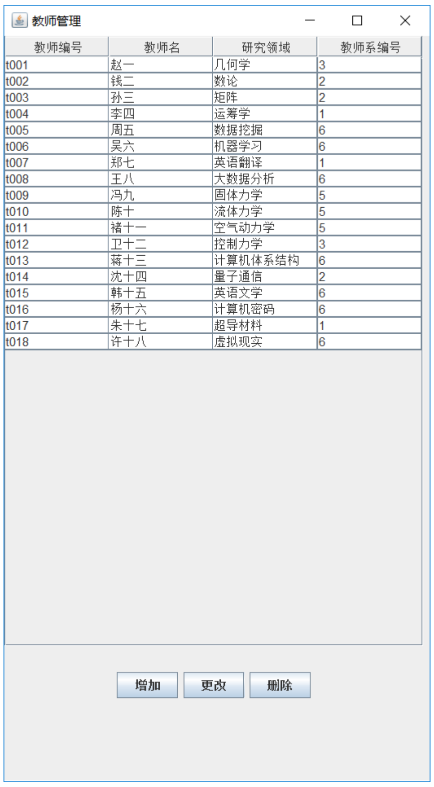
课程名
学分 4
学时



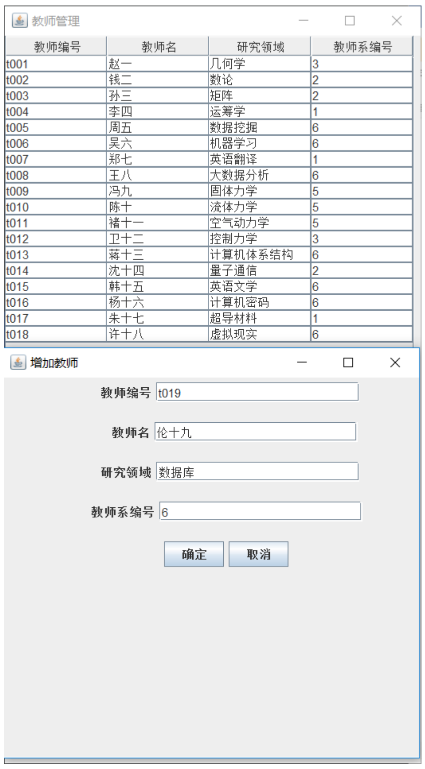
课程编号	课程名	学分	学时
c01	数学分析	5	48
c02	高等代数	5	48
c03	大学物理	5	48
c04	物理化学	4	32
c05	材料学导论	4	32
c06	信号系统	4	32
c07	电子信息导论	4	32
c08	模拟电路	4	32
c09	现代控制导论	4	32
c10	传热学	4	32
c11	机械制图	4	32
c12	飞行器结构力学	4	32
c13	飞行器总体设计	4	32
c14	程序设计	4	32
c15	现代密码学	4	32
c16	大学英语	3	32
c17	数据库原理	4	48

（4）教师管理

按下“教师管理”按钮后，会出现如下的教师管理界面，管理员可以在这个界面进行教师的增删改操作。

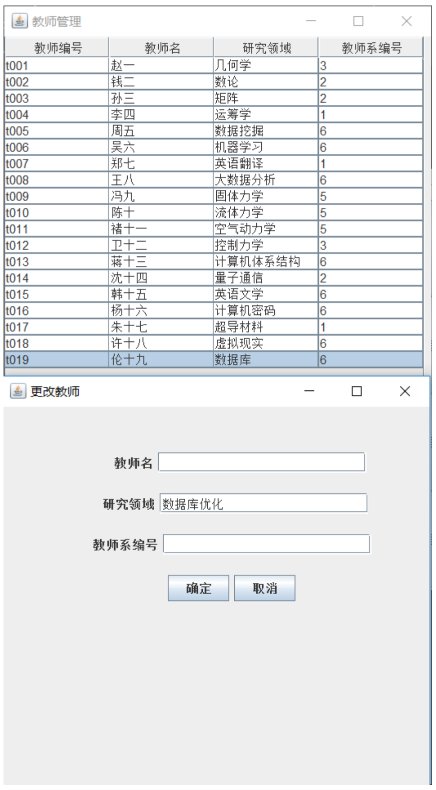


教师编号	教师名	研究领域	教师系编号
t001	赵一	几何学	3
t002	钱二	数论	2
t003	孙三	矩阵	2
t004	李四	运筹学	1
t005	周五	数据挖掘	6
t006	吴六	机器学习	6
t007	郑七	英语翻译	1
t008	王八	大数据分析	6
t009	冯九	固体力学	5
t010	陈十	流体力学	5
t011	褚十一	空气动力学	5
t012	卫十二	控制力学	3
t013	蒋十三	计算机体系结构	6
t014	沈十四	量子通信	2
t015	韩十五	英语文学	6
t016	杨十六	计算机密码	6
t017	朱十七	超导材料	1
t018	许十八	虚拟现实	6



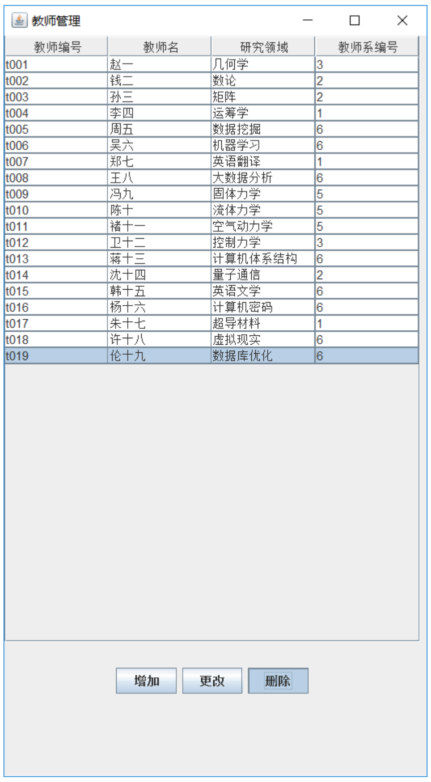
教师编号	教师名	研究领域	教师系编号
t001	赵一	几何学	3
t002	钱二	数论	2
t003	孙三	矩阵	2
t004	李四	运筹学	1
t005	周五	数据挖掘	6
t006	吴六	机器学习	6
t007	郑七	英语翻译	1
t008	王八	大数据分析	6
t009	冯九	固体力学	5
t010	陈十	流体力学	5
t011	褚十一	空气动力学	5
t012	卫十二	控制力学	3
t013	蒋十三	计算机体系结构	6
t014	沈十四	量子通信	2
t015	韩十五	英语文学	6
t016	杨十六	计算机密码	6
t017	朱十七	超导材料	1
t018	许十八	虚拟现实	6

教师编号	t019
教师名	伦十九
研究领域	数据库
教师系编号	6



教师编号	教师名	研究领域	教师系编号
t001	赵一	几何学	3
t002	钱二	数论	2
t003	孙三	矩阵	2
t004	李四	运筹学	1
t005	周五	数据挖掘	6
t006	吴六	机器学习	6
t007	郑七	英语翻译	1
t008	王八	大数据分析	6
t009	冯九	固体力学	5
t010	陈十	流体力学	5
t011	褚十一	空气动力学	5
t012	卫十二	控制力学	3
t013	蒋十三	计算机体系结构	6
t014	沈十四	量子通信	2
t015	韩十五	英语文学	6
t016	杨十六	计算机密码	6
t017	朱十七	超导材料	1
t018	许十八	虚拟现实	6
t019	伦十九	数据库	6

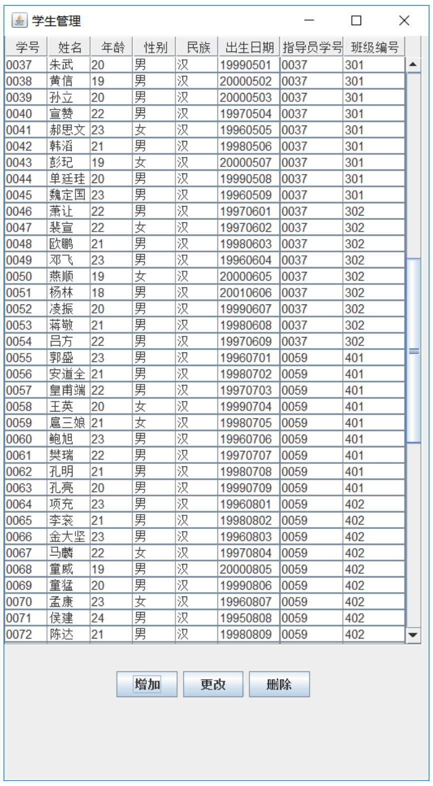
教师名	
研究领域	数据库优化
教师系编号	



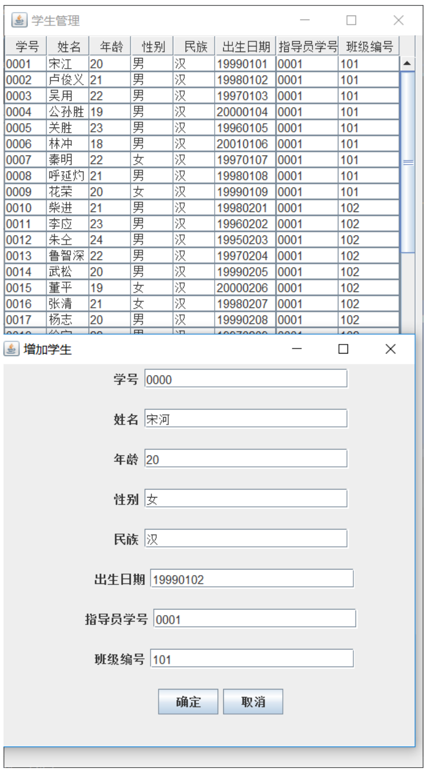
教师编号	教师名	研究领域	教师系编号
t001	赵一	几何学	3
t002	钱二	数论	2
t003	孙三	矩阵	2
t004	李四	运筹学	1
t005	周五	数据挖掘	6
t006	吴六	机器学习	6
t007	郑七	英语翻译	1
t008	王八	大数据分析	6
t009	冯九	固体力学	5
t010	陈十	流体力学	5
t011	褚十一	空气动力学	5
t012	卫十二	控制力学	3
t013	蒋十三	计算机体系结构	6
t014	沈十四	量子通信	2
t015	韩十五	英语文学	6
t016	杨十六	计算机密码	6
t017	朱十七	超导材料	1
t018	许十八	虚拟现实	6
t019	伦十九	数据库优化	6

（5）学生管理

按下“学生管理”按钮后，会出现如下的学生管理界面，管理员可以在这个界面进行学生的增删改操作。



学号	姓名	年龄	性别	民族	出生日期	指导员学号	班级编号
0037	朱武	20	男	汉	19990501	0037	301
0038	黄信	19	男	汉	20000502	0037	301
0039	孙立	20	男	汉	20000503	0037	301
0040	宣赞	22	男	汉	19970504	0037	301
0041	郝思文	23	女	汉	19960505	0037	301
0042	韩滔	21	男	汉	19980506	0037	301
0043	彭玘	19	女	汉	20000507	0037	301
0044	单廷珪	20	男	汉	19990508	0037	301
0045	魏定国	23	男	汉	19960509	0037	301
0046	萧让	22	男	汉	19970601	0037	302
0047	裴宣	22	女	汉	19970602	0037	302
0048	欧鹏	21	男	汉	19980603	0037	302
0049	邓飞	23	男	汉	19960604	0037	302
0050	燕顺	19	女	汉	20000605	0037	302
0051	杨林	18	男	汉	20010606	0037	302
0052	凌振	20	男	汉	19990607	0037	302
0053	蒋敬	21	男	汉	19980608	0037	302
0054	吕方	22	男	汉	19970609	0037	302
0055	郭盛	23	男	汉	19960701	0059	401
0056	安道全	21	男	汉	19980702	0059	401
0057	皇甫端	22	男	汉	19970703	0059	401
0058	王英	20	女	汉	19990704	0059	401
0059	扈三娘	21	女	汉	19980705	0059	401
0060	鲍旭	23	男	汉	19960706	0059	401
0061	樊瑞	22	男	汉	19970707	0059	401
0062	孔明	21	男	汉	19980708	0059	401
0063	孔亮	20	男	汉	19990709	0059	401
0064	项充	23	男	汉	19960801	0059	402
0065	李衮	21	男	汉	19980802	0059	402
0066	金大坚	23	男	汉	19960803	0059	402
0067	马麟	22	女	汉	19970804	0059	402
0068	童威	19	男	汉	20000805	0059	402
0069	童猛	20	男	汉	19990806	0059	402
0070	孟康	23	女	汉	19960807	0059	402
0071	侯建	24	男	汉	19950808	0059	402
0072	陈达	21	男	汉	19980809	0059	402



学号	姓名	年龄	性别	民族	出生日期	指导员学号	班级编号
0001	宋江	20	男	汉	19990101	0001	101
0002	卢俊义	21	男	汉	19980102	0001	101
0003	吴用	22	男	汉	19970103	0001	101
0004	公孙胜	19	男	汉	20000104	0001	101
0005	关胜	23	男	汉	19960105	0001	101
0006	林冲	18	男	汉	20010106	0001	101
0007	秦明	22	女	汉	19970107	0001	101
0008	呼延灼	21	男	汉	19980108	0001	101
0009	花荣	20	女	汉	19990109	0001	101
0010	柴进	21	男	汉	19980201	0001	102
0011	李应	23	男	汉	19960202	0001	102
0012	朱仝	24	男	汉	19950203	0001	102
0013	鲁智深	22	男	汉	19970204	0001	102
0014	武松	20	男	汉	19990205	0001	102
0015	董平	19	女	汉	20000206	0001	102
0016	张清	21	女	汉	19980207	0001	102
0017	杨志	20	男	汉	19990208	0001	102

学号 0000

姓名 宋河

年龄 20

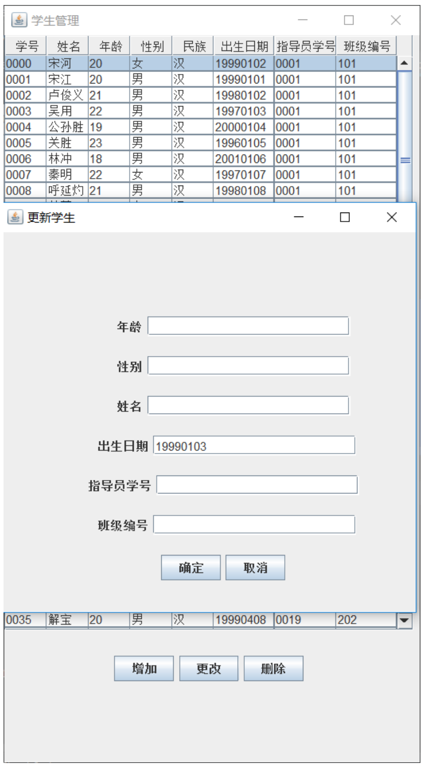
性别 女

民族 汉

出生日期 19990102

指导员学号 0001

班级编号 101



学号	姓名	年龄	性别	民族	出生日期	指导员学号	班级编号
0000	宋河	20	女	汉	19990102	0001	101
0001	宋江	20	男	汉	19990101	0001	101
0002	卢俊义	21	男	汉	19980102	0001	101
0003	吴用	22	男	汉	19970103	0001	101
0004	公孙胜	19	男	汉	20000104	0001	101
0005	关胜	23	男	汉	19960105	0001	101
0006	林冲	18	男	汉	20010106	0001	101
0007	秦明	22	女	汉	19970107	0001	101
0008	呼延灼	21	男	汉	19980108	0001	101
0035	解宝	20	男	汉	19990408	0019	202

年龄

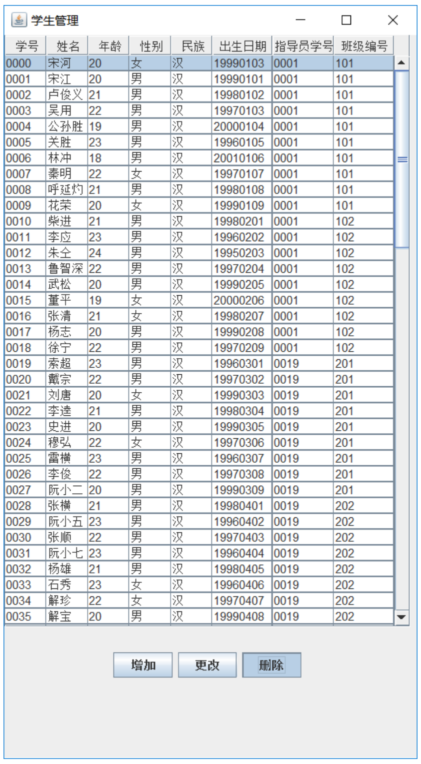
性别

姓名

出生日期 19990103

指导员学号

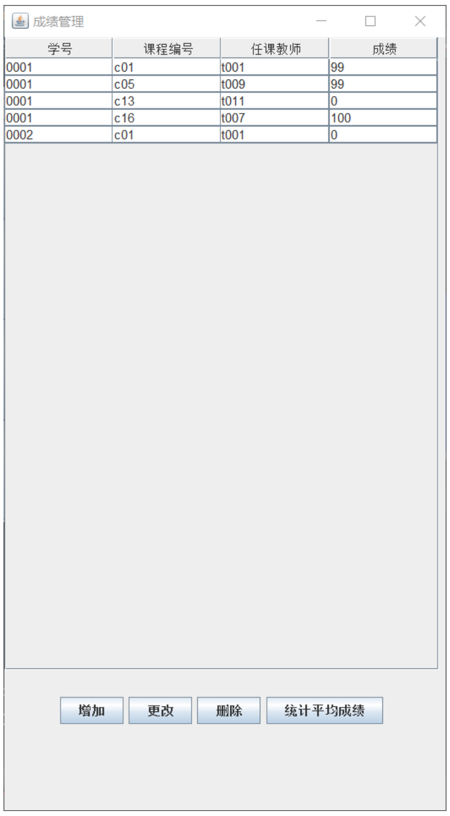
班级编号



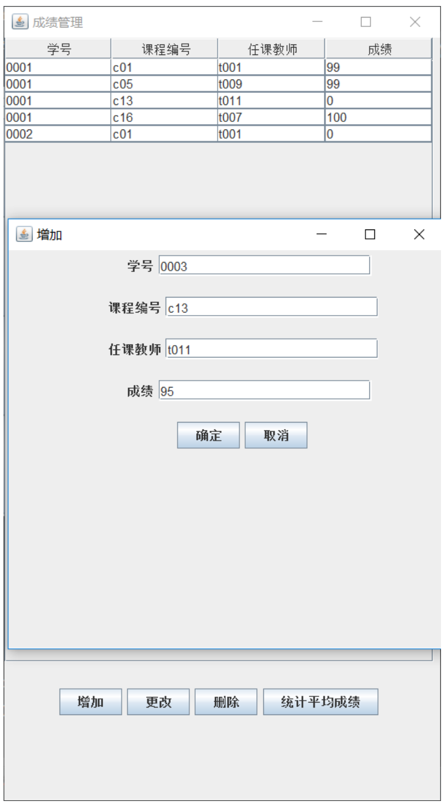
学号	姓名	年龄	性别	民族	出生日期	指导员学号	班级编号
0000	宋河	20	女	汉	19990103	0001	101
0001	宋江	20	男	汉	19990101	0001	101
0002	卢俊义	21	男	汉	19980102	0001	101
0003	吴用	22	男	汉	19970103	0001	101
0004	公孙胜	19	男	汉	20000104	0001	101
0005	关胜	23	男	汉	19960105	0001	101
0006	林冲	18	男	汉	20010106	0001	101
0007	秦明	22	女	汉	19970107	0001	101
0008	呼延灼	21	男	汉	19980108	0001	101
0009	花荣	20	女	汉	19990109	0001	101
0010	柴进	21	男	汉	19980201	0001	102
0011	李应	23	男	汉	19960202	0001	102
0012	朱仝	24	男	汉	19950203	0001	102
0013	鲁智深	22	男	汉	19970204	0001	102
0014	武松	20	男	汉	19990205	0001	102
0015	董平	19	女	汉	20000206	0001	102
0016	张清	21	女	汉	19980207	0001	102
0017	杨志	20	男	汉	19990208	0001	102
0018	徐宁	22	男	汉	19970209	0001	102
0019	索超	23	男	汉	19960301	0019	201
0020	戴宗	22	男	汉	19970302	0019	201
0021	刘唐	20	女	汉	19990303	0019	201
0022	李逵	21	男	汉	19980304	0019	201
0023	史进	20	男	汉	19990305	0019	201
0024	穆弘	22	女	汉	19970306	0019	201
0025	雷横	23	男	汉	19960307	0019	201
0026	李俊	22	男	汉	19970308	0019	201
0027	阮小二	20	男	汉	19990309	0019	201
0028	张横	21	男	汉	19980401	0019	202
0029	阮小五	23	男	汉	19960402	0019	202
0030	张顺	22	男	汉	19970403	0019	202
0031	阮小七	23	男	汉	19960404	0019	202
0032	杨雄	21	男	汉	19980405	0019	202
0033	石秀	23	女	汉	19960406	0019	202
0034	解珍	22	女	汉	19970407	0019	202
0035	解宝	20	男	汉	19990408	0019	202

（6）选课-成绩管理

按下“选课-成绩管理”按钮后，会出现如下的选课-成绩管理界面，管理员可以在这个界面进行学生选课和成绩的增删改操作，还可以统计课程的平均成绩。



学号	课程编号	任课教师	成绩
0001	c01	t001	99
0001	c05	t009	99
0001	c13	t011	0
0001	c16	t007	100
0002	c01	t001	0



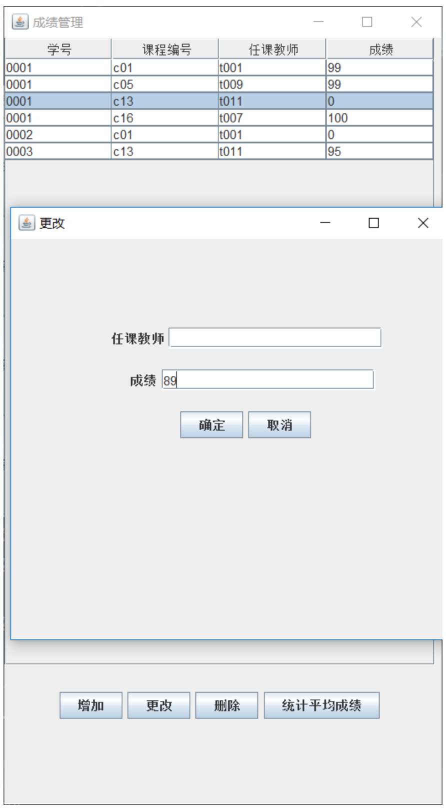
学号	课程编号	任课教师	成绩
0001	c01	t001	99
0001	c05	t009	99
0001	c13	t011	0
0001	c16	t007	100
0002	c01	t001	0

学号 0003

课程编号 c13

任课教师 t011

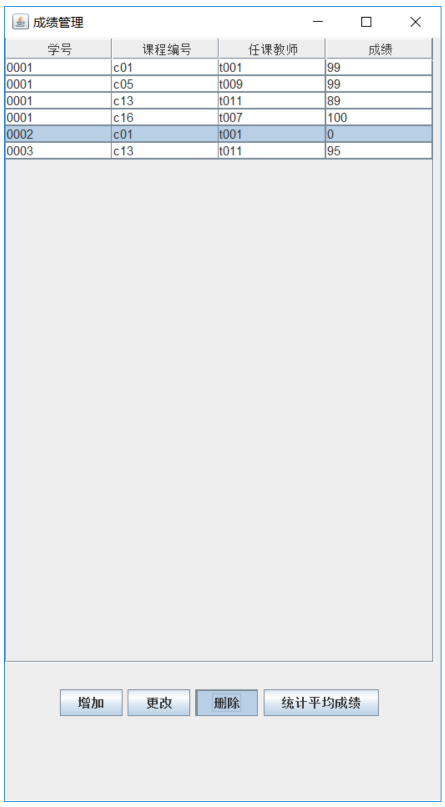
成绩 95



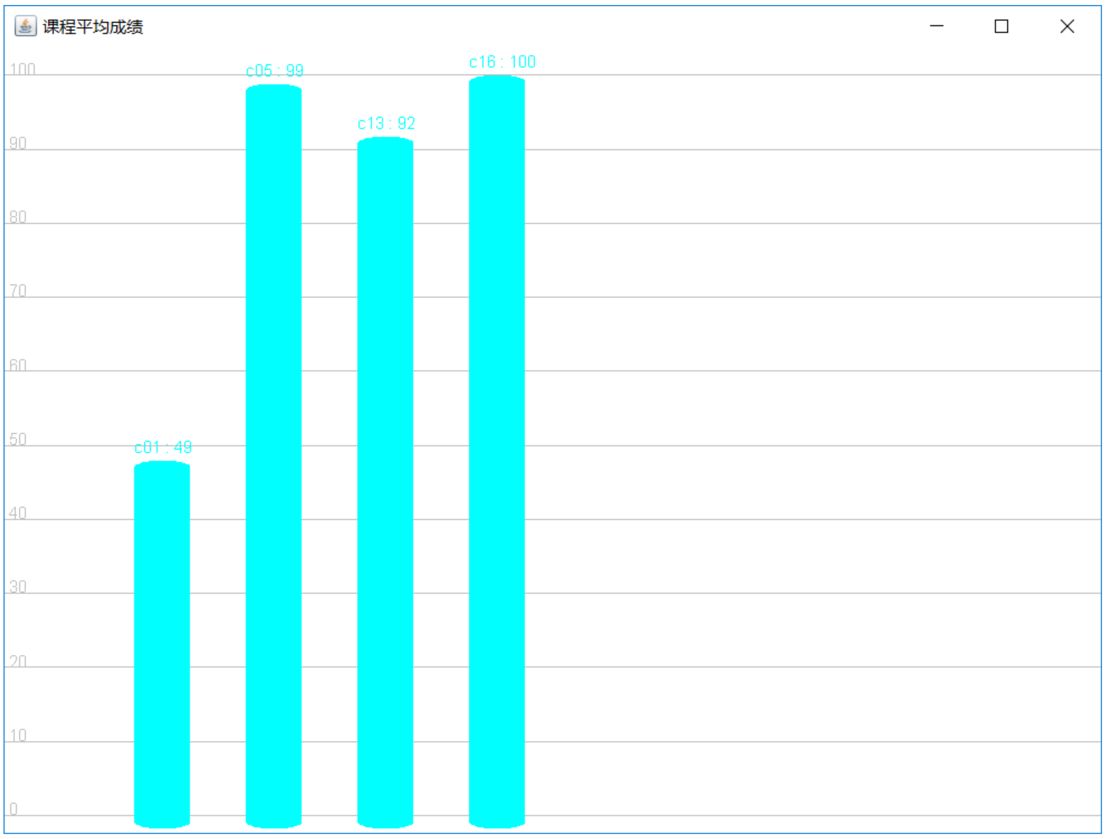
学号	课程编号	任课教师	成绩
0001	c01	t001	99
0001	c05	t009	99
0001	c13	t011	0
0001	c16	t007	100
0002	c01	t001	0
0003	c13	t011	95

任课教师

成绩 89

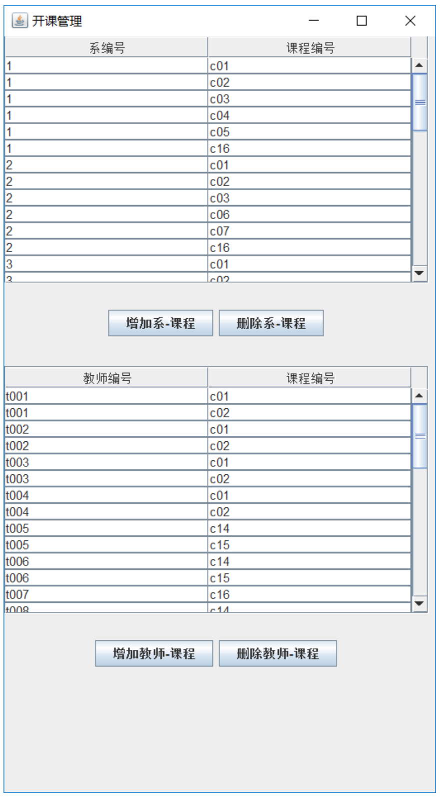


学号	课程编号	任课教师	成绩
0001	c01	t001	99
0001	c05	t009	99
0001	c13	t011	89
0001	c16	t007	100
0002	c01	t001	0
0003	c13	t011	95



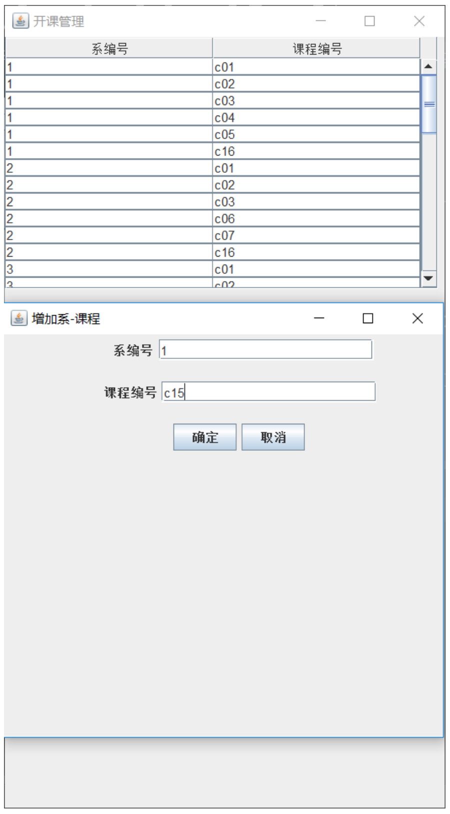
（7）开课管理

按下“开课管理”按钮后，会出现如下的开课管理界面，管理员可以在这个界面进行系-课程和教师-课程的增删操作。



系编号	课程编号
1	c01
1	c02
1	c03
1	c04
1	c05
1	c16
2	c01
2	c02
2	c03
2	c06
2	c07
2	c16
3	c01
3	c02

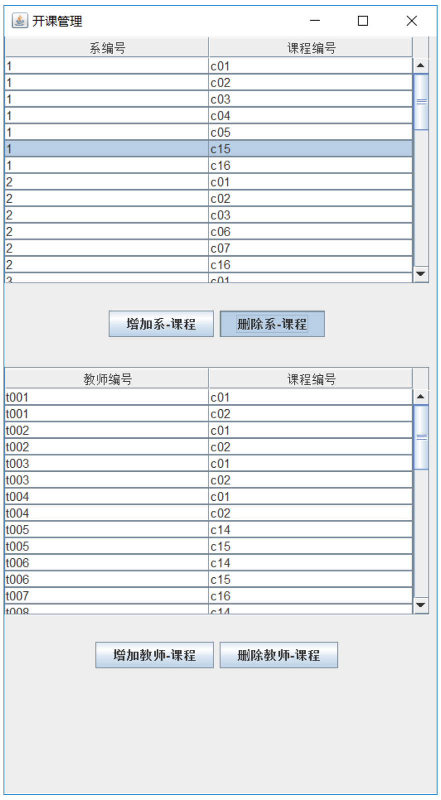
教师编号	课程编号
t001	c01
t001	c02
t002	c01
t002	c02
t003	c01
t003	c02
t004	c01
t004	c02
t005	c14
t005	c15
t006	c14
t006	c15
t007	c16
t008	c14



系编号	课程编号
1	c01
1	c02
1	c03
1	c04
1	c05
1	c16
2	c01
2	c02
2	c03
2	c06
2	c07
2	c16
3	c01
3	c02

系编号 1

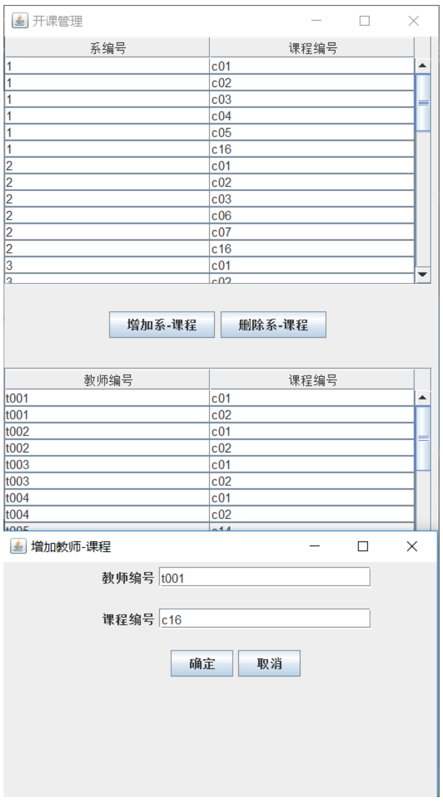
课程编号 c15



开课管理

系编号	课程编号
1	c01
1	c02
1	c03
1	c04
1	c05
1	c15
1	c16
2	c01
2	c02
2	c03
2	c06
2	c07
2	c16

教师编号	课程编号
t001	c01
t001	c02
t002	c01
t002	c02
t003	c01
t003	c02
t004	c01
t004	c02
t005	c14
t005	c15
t006	c14
t006	c15
t007	c16



开课管理

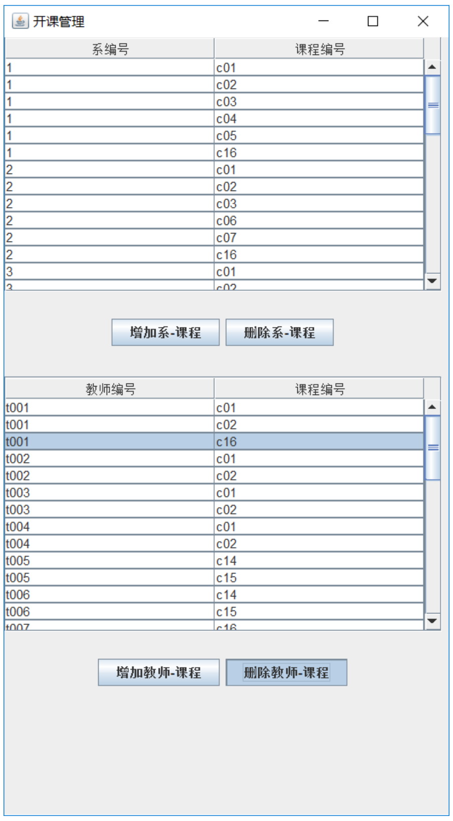
系编号	课程编号
1	c01
1	c02
1	c03
1	c04
1	c05
1	c16
2	c01
2	c02
2	c03
2	c06
2	c07
2	c16
3	c01

教师编号	课程编号
t001	c01
t001	c02
t002	c01
t002	c02
t003	c01
t003	c02
t004	c01
t004	c02

增加教师-课程

教师编号 t001

课程编号 c16



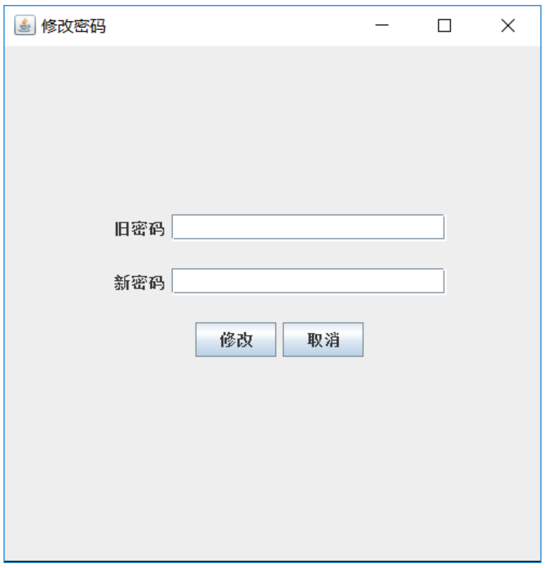
开课管理

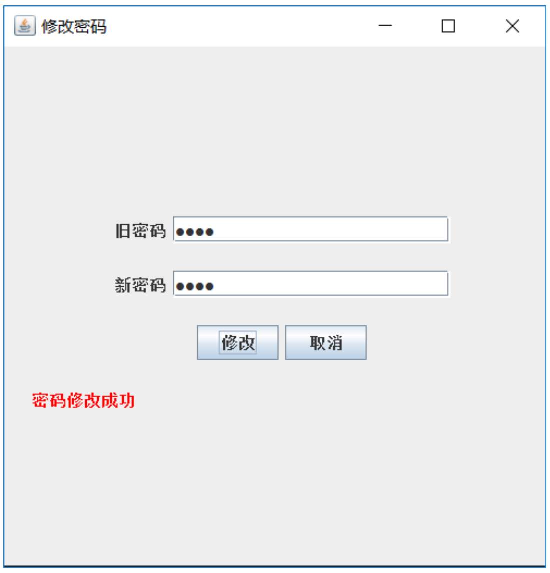
系编号	课程编号
1	c01
1	c02
1	c03
1	c04
1	c05
1	c16
2	c01
2	c02
2	c03
2	c06
2	c07
2	c16
3	c01

教师编号	课程编号
t001	c01
t001	c02
t001	c16
t002	c01
t002	c02
t003	c01
t003	c02
t004	c01
t004	c02
t005	c14
t005	c15
t006	c14
t006	c15

（8）修改密码

按下“修改密码”按钮后，会出现如下的修改密码界面，管理员可以在这个界面进行密码的修改。

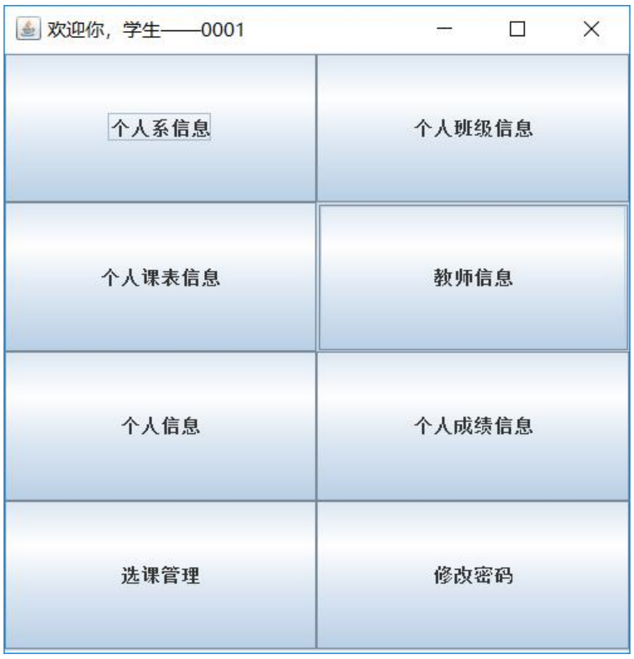




密码修改成功

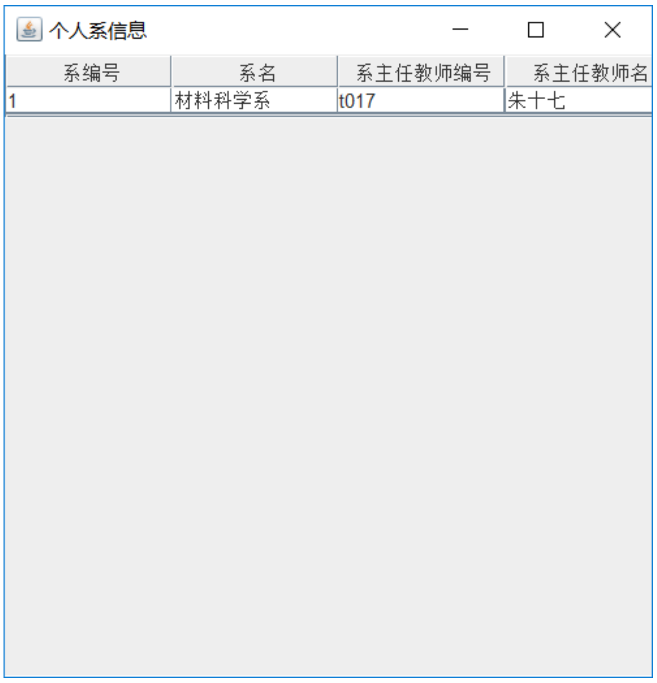
2.8.3 学生功能

学生登录后，会进入如下的学生界面，学生的功能有个人系信息、个人班级信息、个人课表信息、教师信息、个人信息、个人成绩信息、选课管理和修改密码。



（1）个人系信息

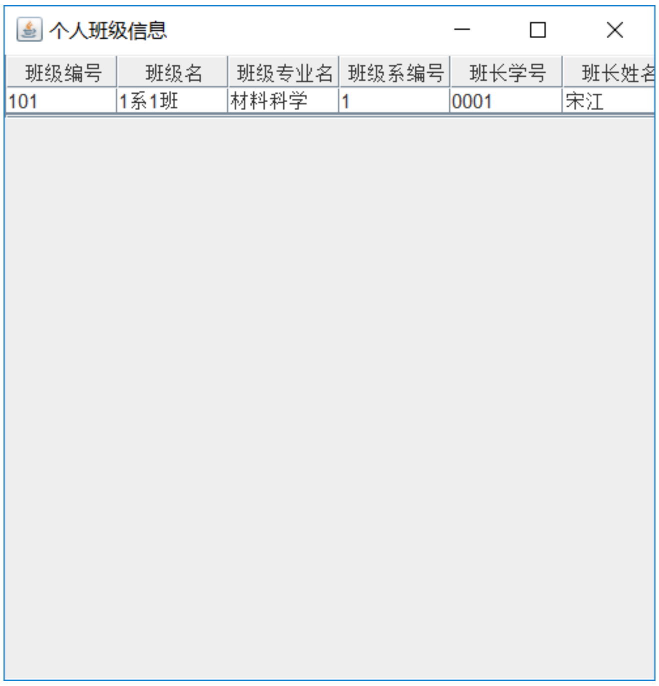
学生按下“个人系信息”按钮后，会出现如下学生个人所在系信息的界面。



系编号	系名	系主任教师编号	系主任教师名
1	材料科学系	t017	朱十七

（2）个人班级信息

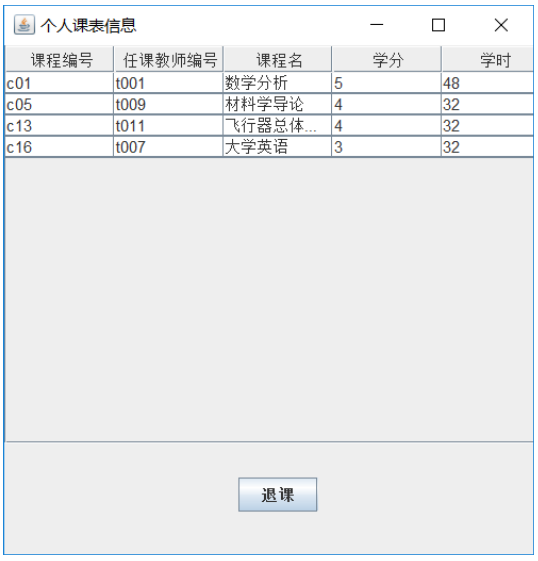
学生按下“个人班级信息”按钮后，会出现如下学生个人所在班级信息的界面。



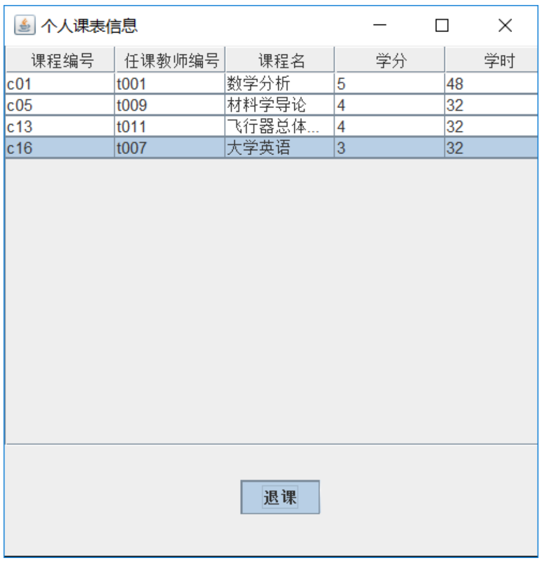
班级编号	班级名	班级专业名	班级系编号	班长学号	班长姓名
101	1系1班	材料科学	1	0001	宋江

（3）个人课表信息

学生按下“个人课表信息”按钮后，会出现如下学生个人课表信息的界面，在这个界面学生可以选定一门课进行退课操作。



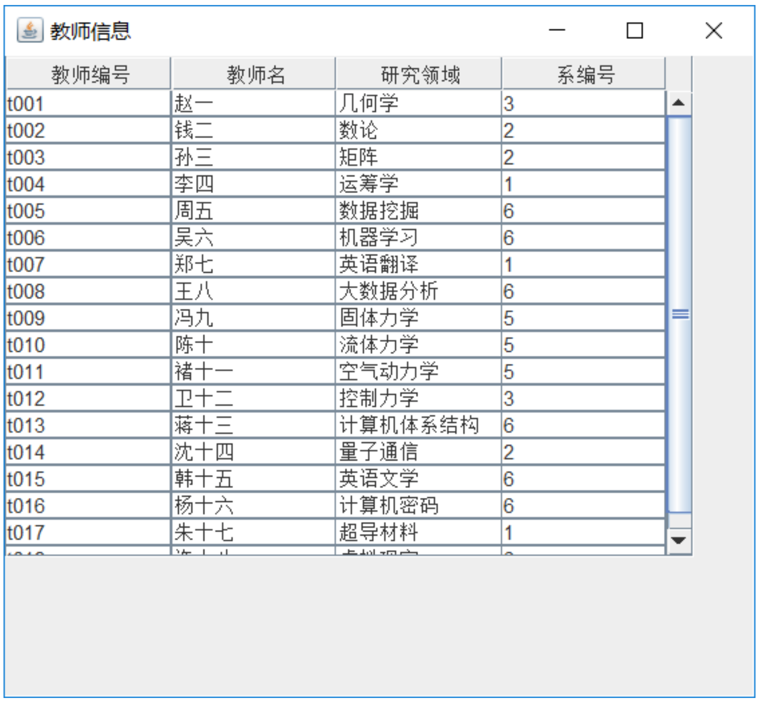
课程编号	任课教师编号	课程名	学分	学时
c01	t001	数学分析	5	48
c05	t009	材料学导论	4	32
c13	t011	飞行器总体...	4	32
c16	t007	大学英语	3	32



课程编号	任课教师编号	课程名	学分	学时
c01	t001	数学分析	5	48
c05	t009	材料学导论	4	32
c13	t011	飞行器总体...	4	32
c16	t007	大学英语	3	32

（4）教师信息

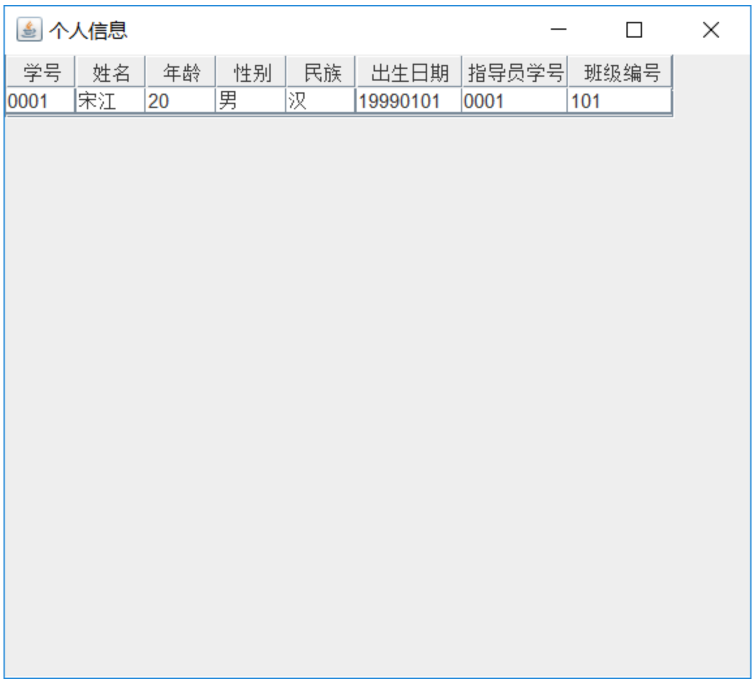
学生按下“教师信息”按钮后，会出现如下包含学校所有教师信息的界面，根据这些教师信息可以方便学生选课。



教师编号	教师名	研究领域	系编号
t001	赵一	几何学	3
t002	钱二	数论	2
t003	孙三	矩阵	2
t004	李四	运筹学	1
t005	周五	数据挖掘	6
t006	吴六	机器学习	6
t007	郑七	英语翻译	1
t008	王八	大数据分析	6
t009	冯九	固体力学	5
t010	陈十	流体力学	5
t011	褚十一	空气动力学	5
t012	卫十二	控制力学	3
t013	蒋十三	计算机体系结构	6
t014	沈十四	量子通信	2
t015	韩十五	英语文学	6
t016	杨十六	计算机密码	6
t017	朱十七	超导材料	1

（5）个人信息

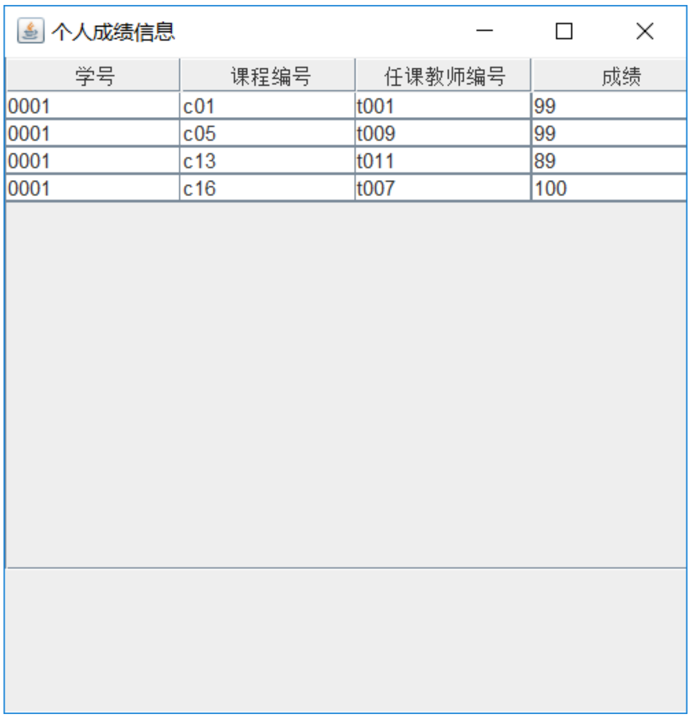
学生按下“个人信息”按钮后，会出现如下包含学生个人信息的界面。



学号	姓名	年龄	性别	民族	出生日期	指导员学号	班级编号
0001	宋江	20	男	汉	19990101	0001	101

（6）个人成绩信息

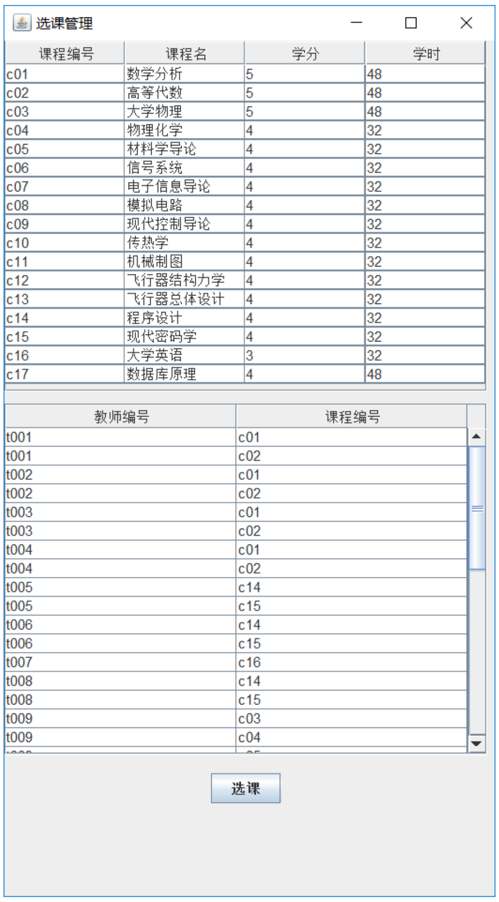
学生按下“个人成绩信息”按钮后，会出现如下包含学生个人成绩信息的界面。



学号	课程编号	任课教师编号	成绩
0001	c01	t001	99
0001	c05	t009	99
0001	c13	t011	89
0001	c16	t007	100

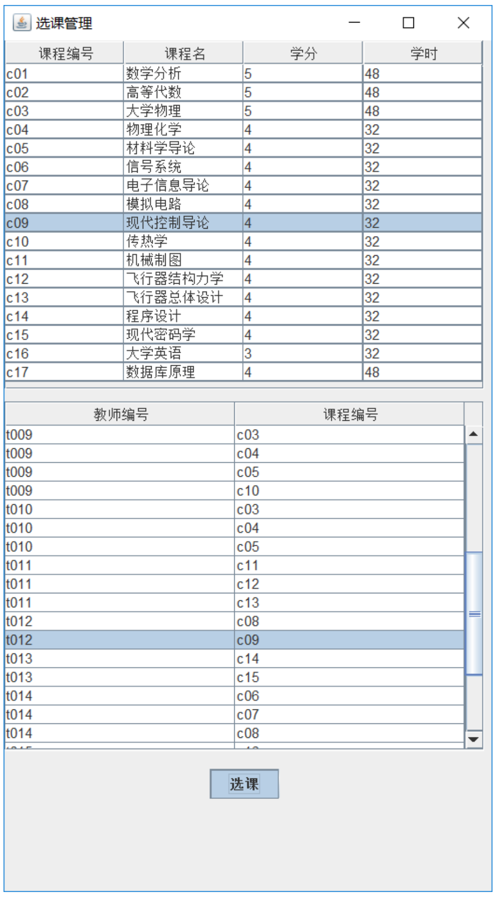
（7）选课管理

学生按下"选课管理"按钮后，会出现如下选课管理的界面，在这个界面学生可以根据现有课程表和教师授课表来选课程及授课教师。



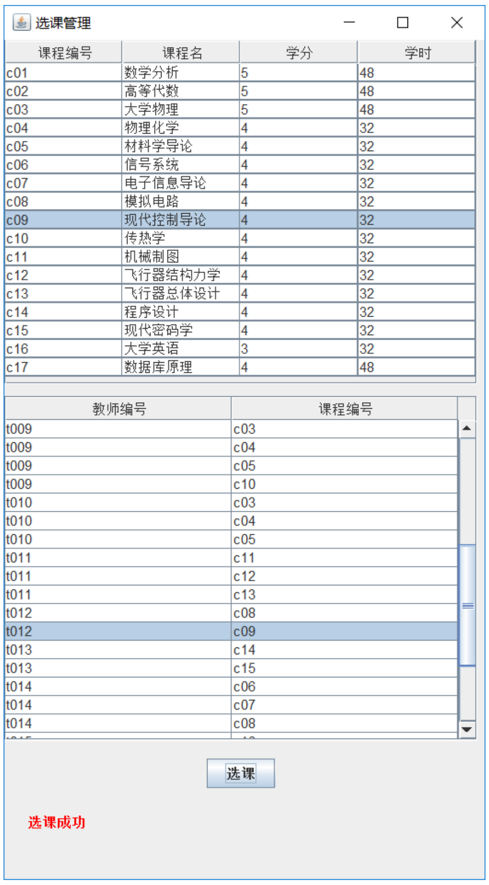
课程编号	课程名	学分	学时
c01	数学分析	5	48
c02	高等代数	5	48
c03	大学物理	5	48
c04	物理化学	4	32
c05	材料学导论	4	32
c06	信号系统	4	32
c07	电子信息导论	4	32
c08	模拟电路	4	32
c09	现代控制导论	4	32
c10	传热学	4	32
c11	机械制图	4	32
c12	飞行器结构力学	4	32
c13	飞行器总体设计	4	32
c14	程序设计	4	32
c15	现代密码学	4	32
c16	大学英语	3	32
c17	数据库原理	4	48

教师编号	课程编号
t001	c01
t001	c02
t002	c01
t002	c02
t003	c01
t003	c02
t004	c01
t004	c02
t005	c14
t005	c15
t006	c14
t006	c15
t007	c16
t008	c14
t008	c15
t009	c03
t009	c04



课程编号	课程名	学分	学时
c01	数学分析	5	48
c02	高等代数	5	48
c03	大学物理	5	48
c04	物理化学	4	32
c05	材料学导论	4	32
c06	信号系统	4	32
c07	电子信息导论	4	32
c08	模拟电路	4	32
c09	现代控制导论	4	32
c10	传热学	4	32
c11	机械制图	4	32
c12	飞行器结构力学	4	32
c13	飞行器总体设计	4	32
c14	程序设计	4	32
c15	现代密码学	4	32
c16	大学英语	3	32
c17	数据库原理	4	48

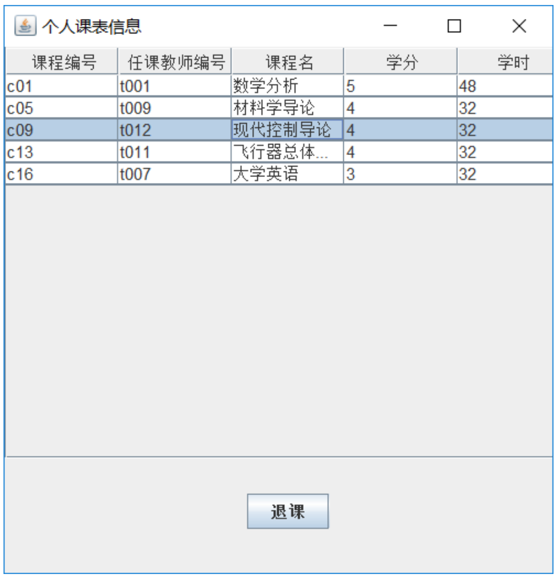
教师编号	课程编号
t009	c03
t009	c04
t009	c05
t009	c10
t010	c03
t010	c04
t010	c05
t011	c11
t011	c12
t011	c13
t012	c08
t012	c09
t013	c14
t013	c15
t014	c06
t014	c07
t014	c08



课程编号	课程名	学分	学时
c01	数学分析	5	48
c02	高等代数	5	48
c03	大学物理	5	48
c04	物理化学	4	32
c05	材料学导论	4	32
c06	信号系统	4	32
c07	电子信息导论	4	32
c08	模拟电路	4	32
c09	现代控制导论	4	32
c10	传热学	4	32
c11	机械制图	4	32
c12	飞行器结构力学	4	32
c13	飞行器总体设计	4	32
c14	程序设计	4	32
c15	现代密码学	4	32
c16	大学英语	3	32
c17	数据库原理	4	48

教师编号	课程编号
t009	c03
t009	c04
t009	c05
t009	c10
t010	c03
t010	c04
t010	c05
t011	c11
t011	c12
t011	c13
t012	c08
t012	c09
t013	c14
t013	c15
t014	c06
t014	c07
t014	c08

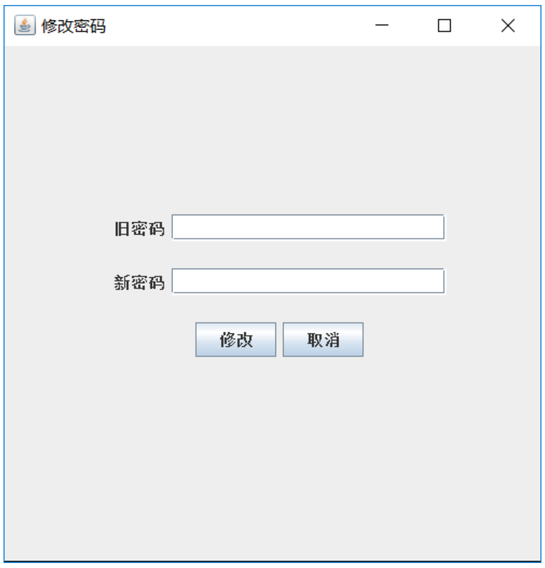
选课成功

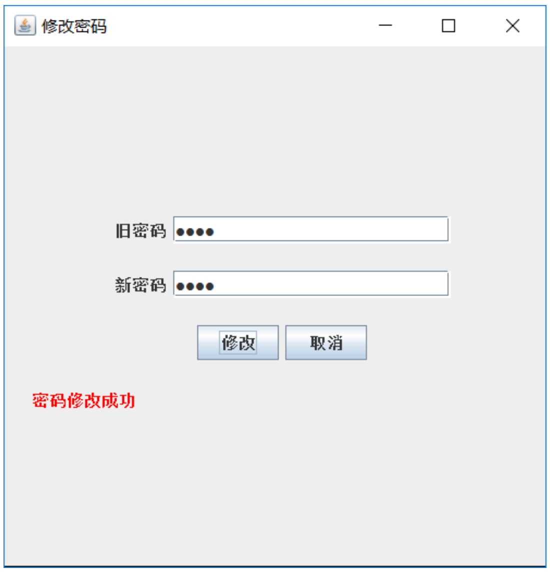


课程编号	任课教师编号	课程名	学分	学时
c01	t001	数学分析	5	48
c05	t009	材料学导论	4	32
c09	t012	现代控制导论	4	32
c13	t011	飞行器总体...	4	32
c16	t007	大学英语	3	32

（8）修改密码

按下“修改密码”按钮后，会出现如下的修改密码界面，学生可以在这个界面进行密码的修改。





旧密码 ●●●●

新密码 ●●●●

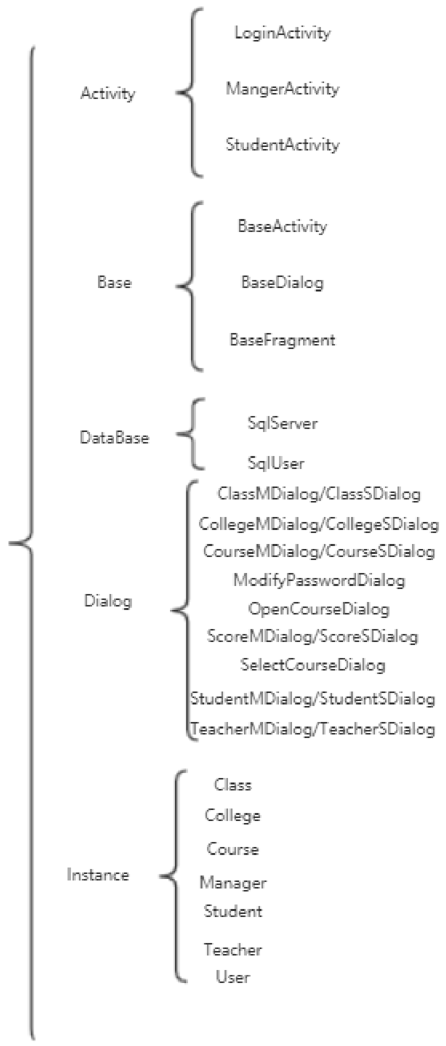
密码修改成功

2.9 源程序简要说明

2.9.1 主要架构

整个系统实现的源程序主要分为两大部分：一是和后端数据库服务器交互的部分，二是和前端用户交互、获取用户输入的信息和操作并调用函数完成后端操作的部分。

系统程序主要架构如下：



Activity 包下的文件主要用来构建用户操作界面，DataBase 包下的文件用来构建程序与数据库服务器的联系，Dialog 包用来产生用户对话框来实现与用户的交互，Instance 包用来产生对象来存储交互的中间信息。

2.9.2 主要代码

（1）前后端交互代码

```
public String modifyStr(String str) {
        return "'" + str + "'";
    }


    public boolean executeUpdateProcedure(String sql) {
        System.out.println(sql);
        try {
            Statement stm = connection.createStatement();
            stm.execute(sql);
            int result = stm.getUpdateCount();
            if (result > 0) {
                System.out.println("SqlServer : SQL EXEC 执行成功");
                return true;
            } else {
                stm.close();
                return false;
            }
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("SqlServer : SQL EXEC 失败" + e.toString());
            return false;
        }
    }


    public ResultSet executeQueryProcedure(String sql) {
        System.out.println(sql);
        try {
            Statement stm = connection.createStatement();
            stm.execute(sql);
            ResultSet result = stm.getResultSet();
            if (result != null) {
                System.out.println("SqlServer : SQL EXEC 执行成功");
                return result;
            } else {
                stm.close();
                return null;
            }
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("SqlServer : SQL EXEC 失败" + e.toString());
            return null;
        }
    }



    private boolean executeUpdate(String sql) {
        System.out.println(sql);
        try {
            Statement stm = connection.createStatement();
            int result = stm.executeUpdate(sql);
            if (result > 0) {
                System.out.println("SqlServer : SQL Update 执行成功");
                return true;
            } else {
                stm.close();
                return false;
            }
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("SqlServer : SQL Update 失败" + e.toString());
            return false;
        }
    }

    public ResultSet executeQuery(String sql) {
        System.out.println(sql);
        try {
            Statement stm = connection.createStatement();
            ResultSet resultSet = stm.executeQuery(sql);
            System.out.println("SqlServer : SQL Query 成功");
            return resultSet;
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("SqlServer : SQL Query 失败" + e.toString());
            return null;
        }
    }

    public boolean checkUser(String userName, String userPassword) {
        String sql = "select dbo.checkUser( " + modifyStr(userName) + " , " +
modifyStr(userPassword) + " )";
        ResultSet resultSet = executeQuery(sql);
        try {
            resultSet.next();
            int result = resultSet.getInt(0);
            return result == 1? true : false;
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("SqlServer : SQL 检验用户 失败" + e.toString());
            return false;
        }
    }

    /******************************/
    /**
     * 超级管理员
     **/

    public boolean rootInsertUser(User user) {
        String sql = "exec dbo.superAddUser " + modifyStr(user.getUserName()) + "," + modifyStr(user.getUserPassword())
                + "," + String.valueOf(user.getUserType());
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("插入用户成功");
            return true;
        } else {
            System.out.println("表中已有该用户信息，插入用户失败");
            return false;
        }
    }

    public boolean rootDeleteUser(String userName) {
        String sql = "exec dbo.superDeleteUser " + modifyStr(userName);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("删除用户成功");
            return true;
        } else {
            System.out.println("表中无该用户信息，删除用户失败");
            return false;
        }
    }

    public boolean rootInsertManager(Manager manager) {
        String sql = "exec dbo.superAddManager " + modifyStr(manager.getManagerName()) +
"," + modifyStr(manager.getManagerNo())
                + "," + modifyStr(manager.getManagerPhone());
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("插入管理员成功");
            User user = new User(manager.getManagerNo(), manager.getManagerNo(), 1);
            rootInsertUser(user);
            return true;
        } else {
            System.out.println("表中已有该管理员信息，插入管理员失败");
            return false;
        }
    }

    public boolean rootDeleteManager(String managerNo) {
        String sql = "exec dbo.superDeleteManager " + modifyStr(managerNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("删除管理员成功");
            rootDeleteUser(managerNo);
            return true;
        } else {
            System.out.println("表中无该管理员信息，删除管理员失败");
            return false;
        }
    }

    public boolean rootUpdateUser(User user) {
        String sql = SQL_SELECT.replace("sth", "*");
        sql = sql.replace("table", "Users");
        sql = sql + "where userName = " + modifyStr(user.getUserName());
        ResultSet resultSet = executeQuery(sql);
        if (resultSet == null) {
            System.out.println("表中无该用户信息，更新用户信息失败");
            return false;
        } else {
            try {
                resultSet.next();
                String nUpassword = resultSet.getString("userPassword");
                int nUtype = resultSet.getInt("userType");
                if (!user.getUserPassword().equals("")) {
                    nUpassword = user.getUserPassword();
                }
                if (user.getUserType() != -1) {
                    nUtype = user.getUserType();
                }
                String exesql = "exec dbo.superUpdateUser " + modifyStr(user.getUserName())
+ "," + modifyStr(nUpassword)
                        + "," + String.valueOf(nUtype);
                boolean result = executeUpdateProcedure(exesql);
                if (result) {
                    System.out.println("更新用户信息成功");
                    return true;
                } else {
```

```
                System.out.println("表中无该用户信息，更新用户信息失败");
                return false;
            }
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }
}

public boolean rootUpdateManager(Manager manager) {
    String sql = SQL_SELECT.replace("sth", "*");
    sql = sql.replace("table", "Manager");
    sql = sql + "where managerNo = " + modifyStr(manager.getManagerNo());
    ResultSet resultSet = executeQuery(sql);
    if (resultSet == null) {
        System.out.println("表中无该管理员信息，更新管理员信息失败");
        return false;
    } else {
        try {
            resultSet.next();
            String nMname = resultSet.getString("managerName");
            String nMphone = resultSet.getString("managerPhone");
            if (!manager.getManagerName().equals("")) {
                nMname = manager.getManagerName();
            }
            if (!manager.getManagerPhone().equals("")) {
                nMphone = manager.getManagerPhone();
            }
            String exesql = "exec dbo.superUpdateManager " + modifyStr(nMname) + ","
```

```
+ modifyStr(manager.getManagerNo())
                        + "," + modifyStr(nMphone);
                boolean result = executeUpdateProcedure(exesql);
                if (result) {
                    System.out.println("更新管理员信息成功");
                    return true;
                } else {
                    System.out.println("表中无该管理员信息，更新管理员信息失败");
                    return false;
                }
            } catch (SQLException e) {
                e.printStackTrace();
                return false;
            }
        }
    }

    public ArrayList<User> rootSelectUser() {
        String sql = "exec dbo.superSelectUsers ";
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<User> userList = new ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询用户表失败");
        } else {
            try {
                while (resultSet.next()) {
                    User user = new User(
                            resultSet.getString("userName"),
                            resultSet.getString("userPassword"),
                            resultSet.getInt("userType")
                    );
                    userList.add(user);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询用户表失败");
            }
        }
        return userList;
    }

    public ArrayList<Manager> rootSelectManager() {
        String sql = "exec dbo.superSelectManager ";
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Manager> managerList = new ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询管理员表失败");
        } else {
            try {
                while (resultSet.next()) {
                    Manager manager = new Manager(
                            resultSet.getString("managerName"),
                            resultSet.getString("managerNo"),
                            resultSet.getString("managerPhone")
                    );
                    managerList.add(manager);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询管理员表失败");
            }
        }
        return managerList;
    }

    /******************************/
    /**
     * 管理员
     **/
    public boolean managerInsertCollege(College college) {
        String sql = "exec dbo.addCollege " + modifyStr(college.getCollegeNo()) + "," +
modifyStr(college.getCollegeName());
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("插入系成功");
            return true;
        } else {
            System.out.println("表中已有该系信息，插入系失败");
            return false;
        }
    }

    public boolean managerDeleteCollege(String collegeNo) {
        String sql = "exec dbo.deleteCollege " + modifyStr(collegeNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("删除系成功");
            managerDeleteCollegeManager(collegeNo);
            String dsql = "delete from CCTable where collegeNo = " + modifyStr(collegeNo);
            executeUpdate(dsql);
            return true;
        } else {
            System.out.println("表中无该系信息，删除系失败");
            return false;
        }
    }

    public boolean managerUpdateCollege(College college) {
        String sql = SQL_SELECT.replace("sth", "*");
        sql = sql.replace("table", "College");
        sql = sql + "where collegeNo = " + modifyStr(college.getCollegeNo());
        ResultSet resultSet = executeQuery(sql);
        if (resultSet == null) {
            System.out.println("表中无该系信息，更新系信息失败");
            return false;
        } else {
            try {
                resultSet.next();
                String nCname = resultSet.getString("collegeName");
                if (!college.getCollegeName().equals("")) {
                    nCname = college.getCollegeName();
                }
                String exesql = "exec dbo.updateCollege " +
modifyStr(college.getCollegeNo()) + "," + modifyStr(nCname);
                boolean result = executeUpdateProcedure(exesql);
                if (result) {
                    System.out.println("更新系信息成功");
                    return true;
                } else {
                    System.out.println("表中无该系信息，更新系信息失败");
                    return false;
                }
            } catch (SQLException e) {
                e.printStackTrace();
                return false;
            }
        }
    }

    public boolean managerInsertClass(Class classObj) {
        String sql = "exec dbo.addClass " + modifyStr(classObj.getClassNo()) + "," +
modifyStr(classObj.getClassName())
                + "," + modifyStr(classObj.getClassProName()) + "," +
modifyStr(classObj.getClassCollegeNo());
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("插入班级成功");
            return true;
        } else {
            System.out.println("表中已有该班级信息，插入班级失败");
            return false;
        }
    }

    public boolean managerDeleteClass(String classNo) {
        String sql = "exec dbo.deleteClass " + modifyStr(classNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("删除班级成功");
            String dsql = "delete from CSTable where classNo = " + modifyStr(classNo);
```

```
            executeUpdate(dsql);
            return true;
        } else {
            System.out.println("表中无该班级信息，删除班级失败");
            return false;
        }
    }

    public boolean managerUpdateClass(Class classObj) {
        String sql = SQL_SELECT.replace("sth", "*");
        sql = sql.replace("table", "Class");
        sql = sql + "where classNo = " + modifyStr(classObj.getClassNo());
        ResultSet resultSet = executeQuery(sql);
        if (resultSet == null) {
            System.out.println("表中无该班级信息，更新班级信息失败");
            return false;
        } else {
            try {
                resultSet.next();
                String nCname = resultSet.getString("className");
                String nCproname = resultSet.getString("classProName");
                String nCcolno = resultSet.getString("classColNo");
                if (!classObj.getClassName().equals("")) {
                    nCname = classObj.getClassName();
                }
                if (!classObj.getClassProName().equals("")) {
                    nCproname = classObj.getClassProName();
                }
                if (!classObj.getClassCollegeNo().equals("")) {
                    nCcolno = classObj.getClassCollegeNo();
                }
                String exesql = "exec dbo.updateClass " + modifyStr(classObj.getClassNo())
+ "," + modifyStr(nCname)
                        + "," + modifyStr(nCproname) + "," + modifyStr(nCcolno);
                boolean result = executeUpdateProcedure(exesql);
                if (result) {
                    System.out.println("更新班级信息成功");
                    return true;
                } else {
                    System.out.println("表中无该班级信息，更新班级信息失败");
                    return false;
                }
            } catch (SQLException e) {
                e.printStackTrace();
                return false;
            }
        }
    }

    public boolean managerInsertTeacher(Teacher teacher) {
        String sql = "exec dbo.addTeacher " + modifyStr(teacher.getTeacherNo()) + "," +
modifyStr(teacher.getTeacherName())
                + "," + modifyStr(teacher.getTeacherResDir()) + "," +
modifyStr(teacher.getTeacherCollegeNo());
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("插入教师成功");
            return true;
        } else {
            System.out.println("表中已有该教师信息，插入教师失败");
            return false;
        }
    }

    public boolean managerDeleteTeacher(String teacherNo) {
        String sql = "exec dbo.deleteTeacher " + modifyStr(teacherNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("删除教师成功");
            String dsql = "delete from TCTable where teacherNo = " + modifyStr(teacherNo);
            executeUpdate(dsql);
            return true;
        } else {
            System.out.println("表中无该教师信息，删除教师失败");
            return false;
        }
    }

    public boolean managerUpdateTeacher(Teacher teacher) {
        String sql = SQL_SELECT.replace("sth", "*");
        sql = sql.replace("table", "Teacher");
        sql = sql + "where teacherNo = " + modifyStr(teacher.getTeacherNo());
        ResultSet resultSet = executeQuery(sql);
        if (resultSet == null) {
            System.out.println("表中无该教师信息，更新教师信息失败");
            return false;
        } else {
            try {
                resultSet.next();
                String nTname = resultSet.getString("teacherName");
                String nTresdir = resultSet.getString("teacherResDir");
                String nTcolno = resultSet.getString("teacherColNo");
                if (!teacher.getTeacherName().equals("")) {
                    nTname = teacher.getTeacherName();
                }
                if (!teacher.getTeacherResDir().equals("")) {
                    nTresdir = teacher.getTeacherResDir();
                }
                if (!teacher.getTeacherCollegeNo().equals("")) {
                    nTcolno = teacher.getTeacherCollegeNo();
                }
                String exesql = "exec dbo.updateTeacher " +
modifyStr(teacher.getTeacherNo()) + "," + modifyStr(nTname)
                        + "," + modifyStr(nTresdir) + "," + modifyStr(nTcolno);
                boolean result = executeUpdateProcedure(exesql);
                if (result) {
                    System.out.println("更新教师信息成功");
                    return true;
                } else {
                    System.out.println("表中无该教师信息，更新教师信息失败");
                    return false;
                }
            } catch (SQLException e) {
                e.printStackTrace();
                return false;
            }
        }
    }

    public boolean managerInsertStudent(Student student) {
        String sql = "exec dbo.addStudent " + modifyStr(student.getStudentNo()) + "," +
modifyStr(student.getStudentName())
                + "," + modifyStr(student.getStudentAge()) + "," +
modifyStr(student.getStudentSex())
                + "," + modifyStr(student.getStudentNation()) + "," +
modifyStr(student.getStudentBirthday())
                + "," + modifyStr(student.getStudentInsNo()) + "," +
modifyStr(student.getStudentClassNo());
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("插入学生成功");
            rootInsertUser(new User(student.getStudentNo(), student.getStudentNo(), 2));
            return true;
        } else {
            System.out.println("表中已有该学生信息，插入学生失败");
            return false;
        }
    }

    public boolean managerDeleteStudent(String studentNo) {
        String sql = "exec dbo.deleteStudent " + modifyStr(studentNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("删除学生成功");
            rootDeleteUser(studentNo);
            String dsql = "delete from SCTSTable where studentNo = " + modifyStr(studentNo);
            executeUpdate(dsql);
            return true;
        } else {
            System.out.println("表中无该学生信息，删除学生失败");
            return false;
        }
    }

    public boolean managerUpdateStudent(Student student) {
        String sql = SQL_SELECT.replace("sth", "*");
        sql = sql.replace("table", "Student");
        sql = sql + "where studentNo = " + modifyStr(student.getStudentNo());
        ResultSet resultSet = executeQuery(sql);
        if (resultSet == null) {
            System.out.println("表中无该学生信息，更新学生信息失败");
            return false;
        } else {
            try {
                resultSet.next();
                String nSname = resultSet.getString("studentName");
                String nSage = resultSet.getString("studentAge");
                String nSsex = resultSet.getString("studentSex");
                String nSnation = resultSet.getString("studentNation");
                String nSbirthday = resultSet.getString("studentBirthday");
                String nSinsno = resultSet.getString("studentInsNo");
                String nSclassno = resultSet.getString("studentClassNo");
                if (!student.getStudentName().equals("")) {
                    nSname = student.getStudentName();
                }
                if (!student.getStudentAge().equals("")) {
                    nSage = student.getStudentAge();
                }
                if (!student.getStudentSex().equals("")) {
                    nSsex = student.getStudentSex();
                }
                if (!student.getStudentNation().equals("")) {
                    nSnation = student.getStudentNation();
                }
                if (!student.getStudentBirthday().equals("")) {
                    nSbirthday = student.getStudentBirthday();
                }
                if (!student.getStudentInsNo().equals("")) {
                    nSinsno = student.getStudentInsNo();
                }
                if (!student.getStudentClassNo().equals("")) {
                    nSclassno = student.getStudentClassNo();
                }
                String      exesql      =      "exec      dbo.updateStudent      "      +
modifyStr(student.getStudentNo()) + "," + modifyStr(nSname)
                        + "," + modifyStr(nSage) + "," + modifyStr(nSsex) + "," +
modifyStr(nSnation) + "," + modifyStr(nSbirthday)
                        + "," + modifyStr(nSinsno) + "," + modifyStr(nSclassno);
                boolean result = executeUpdateProcedure(exesql);
                if (result) {
                    System.out.println("更新学生信息成功");
                    return true;
                } else {
                    System.out.println("表中无该学生信息，更新学生信息失败");
                    return false;
                }
            } catch (SQLException e) {
                e.printStackTrace();
                return false;
            }
```

```
        }
    }

    public boolean managerInsertCourse(Course course) {
        String sql = "exec dbo.addCourse " + modifyStr(course.getCourseNo()) + "," + 
modifyStr(course.getCourseName())
                + "," + String.valueOf(course.getCourseCredict()) + "," + 
String.valueOf(course.getCoursePeriod());
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("插入课程成功");
            return true;
        } else {
            System.out.println("表中已有该课程信息，插入课程失败");
            return false;
        }
    }

    public boolean managerDeleteCourse(String courseNo) {
        String sql = "exec dbo.deleteCourse " + modifyStr(courseNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("删除课程成功");
            String dsql1 = "delete from CCTable where courseNo = " + modifyStr(courseNo);
            executeUpdate(dsql1);
            String dsql2 = "delete from TCTable where courseNo = " + modifyStr(courseNo);
            executeUpdate(dsql2);
            String dsql3 = "delete from SCTSTable where courseNo = " + modifyStr(courseNo);
            executeUpdate(dsql3);
            return true;
        } else {
            System.out.println("表中无该课程信息，删除课程失败");
            return false;
        }
    }

    public boolean managerUpdateCourse(Course course) {
        String sql = SQL_SELECT.replace("sth", "*");
        sql = sql.replace("table", "Course");
        sql = sql + "where courseNo = " + modifyStr(course.getCourseNo());
        ResultSet resultSet = executeQuery(sql);
        if (resultSet == null) {
            System.out.println("表中无该班级信息，更新班级信息失败");
            return false;
        } else {
            try {
                resultSet.next();
                String nCname = resultSet.getString("courseName");
                int nCcredict = resultSet.getInt("courseCredict");
                int nCperiod = resultSet.getInt("coursePeriod");
                if (!course.getCourseName().equals("")) {
                    nCname = course.getCourseName();
                }
                if (course.getCourseCredict() != -1) {
                    nCcredict = course.getCourseCredict();
                }
                if (course.getCoursePeriod() != -1) {
                    nCperiod = course.getCoursePeriod();
                }
```

```
                String exesql = "exec dbo.updateCourse " + modifyStr(course.getCourseNo())
+ "," + modifyStr(nCname)
                        + "," + String.valueOf(nCcredict) + "," + String.valueOf(nCperiod);
                boolean result = executeUpdateProcedure(exesql);
                if (result) {
                    System.out.println("更新课程信息成功");
                    return true;
                } else {
                    System.out.println("表中无该课程信息，更新课程信息失败");
                    return false;
                }
            } catch (SQLException e) {
                e.printStackTrace();
                return false;
            }
        }
    }

    public boolean managerInsertCollegeManager(String collegeNo, String teacherNo) {
        String sql = "exec dbo.addCollegeManager " + modifyStr(collegeNo) + "," +
modifyStr(teacherNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("插入系-系主任成功");
            return true;
        } else {
            System.out.println("表中已有该系-系主任信息，插入系-系主任失败");
            return false;
        }
    }
```

```
public boolean managerDeleteCollegeManager(String collegeNo) {
    String sql = "exec dbo.deleteCollegeManager " + modifyStr(collegeNo);
    boolean result = executeUpdateProcedure(sql);
    if (result) {
        System.out.println("删除系-系主任成功");
        return true;
    } else {
        System.out.println("表中无该系-系主任信息，删除系-系主任失败");
        return false;
    }
}

public boolean managerUpdateCollegeManager(String collegeNo, String teacherNo) {
    String sql = SQL_SELECT.replace("sth", "*");
    sql = sql.replace("table", "CTTable");
    sql = sql + "where collegeNo = " + modifyStr(collegeNo);
    ResultSet resultSet = executeQuery(sql);
    if (resultSet == null) {
        System.out.println("表中无该系-系主任信息，更新系-系主任信息失败");
        return false;
    } else {
        try {
            resultSet.next();
            String nTno = resultSet.getString("teacherNo");
            if (!teacherNo.equals("")) {
                nTno = teacherNo;
            }

            String exesql = "exec dbo.updateCollegeManager " + modifyStr(collegeNo) +
"," + modifyStr(nTno);
                boolean result = executeUpdateProcedure(exesql);
                if (result) {
                    System.out.println("更新系-系主任信息成功");
                    return true;
                } else {
                    System.out.println("表中无该系-系主任信息，更新系-系主任信息失败");
                    return false;
                }
            } catch (SQLException e) {
                e.printStackTrace();
                return false;
            }
        }
    }

    public boolean managerInsertCollegeCourse(String collegeNo, String courseNo) {
        String sql = "exec dbo.addCollegeCourse " + modifyStr(collegeNo) + "," +
modifyStr(courseNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("插入系-课程成功");
            return true;
        } else {
            System.out.println("表中已有该系-课程信息，插入系-课程失败");
            return false;
        }
    }

    public boolean managerDeleteCollegeCourse(String collegeNo, String courseNo) {
        String sql = "exec dbo.deleteCollegeCourse " + modifyStr(collegeNo) + "," +
modifyStr(courseNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("删除系-课程成功");
            return true;
        } else {
            System.out.println("表中无该系-课程信息，删除系-课程失败");
            return false;
        }
    }

    public boolean managerInsertClassMonitor(String classNo, String studentNo) {
        String sql = "exec dbo.addClassMonitor " + modifyStr(classNo) + "," +
modifyStr(studentNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("插入班级-班长成功");
            return true;
        } else {
            System.out.println("表中已有该班级-班长信息，插入班级-班长失败");
            return false;
        }
    }

    public boolean managerDeleteClassMonitor(String classNo, String studentNo) {
        String sql = "exec dbo.deleteClassMonitor " + modifyStr(classNo) + "," +
modifyStr(studentNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("删除班级-班长成功");

            return true;

        } else {

            System.out.println("表中无该系信息，删除班级-班长失败");

            return false;

        }

    }


    public boolean managerUpdateClassMonitor(String classNo, String studentNo) {

        String sql = SQL_SELECT.replace("sth", "*");

        sql = sql.replace("table", "CSTable");

        sql = sql + "where classNo = " + modifyStr(classNo);

        ResultSet resultSet = executeQuery(sql);

        if (resultSet == null) {

            System.out.println("表中无该班级-班长信息，更新班级-班长信息失败");

            return false;

        } else {

            try {

                resultSet.next();

                String nSno = resultSet.getString("studentNo");

                if (!studentNo.equals("")) {

                    nSno = studentNo;

                }

                String exesql = "exec dbo.updateClassMonitor " + modifyStr(classNo) + ","
+ modifyStr(nSno);

                boolean result = executeUpdateProcedure(exesql);

                if (result) {

                    System.out.println("更新班级-班长信息成功");

                    return true;

                } else {
                    System.out.println("表中无该班级-班长信息，更新班级-班长信息失败");
                    return false;
                }
            } catch (SQLException e) {
                e.printStackTrace();
                return false;
            }
        }
    }

    public boolean managerInsertTeacherCourse(String teacherNo, String courseNo) {
        String  sql  =  "exec  dbo.addTeacherCourse  "  +  modifyStr(teacherNo)  +  ","  +
modifyStr(courseNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("插入教师-课程成功");
            return true;
        } else {
            System.out.println("表中已有该教师-课程信息，插入教师-课程失败");
            return false;
        }
    }

    public boolean managerDeleteTeacherCourse(String teacherNo, String courseNo) {
        String  sql  =  "exec  dbo.deleteTeacherCourse  "  +  modifyStr(teacherNo)  +  ","  +
modifyStr(courseNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("删除教师-课程成功");
            return true;
        } else {
            System.out.println("表中无该教师-课程信息，删除教师-课程失败");
            return false;
        }
    }

    public boolean managerInsertStuTeaCouSco(String studentNo, String courseNo, String
teacherNo, int score) {
        String sql = "exec dbo.addStuTeaCouSco " + modifyStr(studentNo) + "," +
modifyStr(courseNo)
                + "," + modifyStr(teacherNo) + "," + String.valueOf(score);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("插入学生-课程-教师-成绩成功");
            return true;
        } else {
            System.out.println("表中已有该学生-课程-教师-成绩信息，插入学生-课程-教师-
成绩失败");
            return false;
        }
    }

    public boolean managerDeleteStuCouTeaSco(String studentNo, String courseNo) {
        String sql = "exec dbo.deleteStuCouTeaSco " + modifyStr(studentNo) + "," +
modifyStr(courseNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("删除学生-课程-教师-成绩成功");
            return true;
        } else {
            System.out.println("表中无该学生-课程-教师-成绩信息，删除学生-课程-教师-成绩失败");
            return false;
        }
    }

    public boolean managerUpdateStuCouTeaSco(String studentNo, String courseNo, String teacherNo, int score) {
        String sql = SQL_SELECT.replace("sth", "*");
        sql = sql.replace("table", "SCTSTable");
        sql = sql + "where studentNo = " + modifyStr(studentNo) + " and " + "courseNo = " + modifyStr(courseNo);
        ResultSet resultSet = executeQuery(sql);
        if (resultSet == null) {
            System.out.println("表中无该学生-课程-教师-成绩信息，更新学生-课程-教师-成绩信息失败");
            return false;
        } else {
            try {
                resultSet.next();
                String nTno = resultSet.getString("teacherNo");
                int nScore = resultSet.getInt("score");
                if (!teacherNo.equals("")) {
                    nTno = teacherNo;
                }
                if (score != -1) {
                    nScore = score;
                }

                String exesql = "exec dbo.updateStuCouTeaSco " + modifyStr(studentNo) + ","
```

```
+ modifyStr(courseNo) + ","
                        + modifyStr(nTno) + "," + String.valueOf(nScore);
                boolean result = executeUpdateProcedure(exesql);
                if (result) {
                    System.out.println("更新学生-课程-教师-成绩信息成功");
                    return true;
                } else {
                    System.out.println("表中无该学生-课程-教师-成绩信息，更新学生-课程-
教师-成绩信息失败");
                    return false;
                }
            } catch (SQLException e) {
                e.printStackTrace();
                return false;
            }
        }
    }

    public boolean managerUpdatePassword(String userName, String userOldPassword, String
userNewPassword) {
        String exesql = "exec dbo.updatePassword " + modifyStr(userName) + "," +
modifyStr(userOldPassword) + ","
                + modifyStr(userNewPassword);
        boolean result = executeUpdateProcedure(exesql);
        if (result) {
            System.out.println("更新密码成功");
            return true;
        } else {
            System.out.println("表中无该用户信息或旧密码不正确，更新密码失败");
            return false;
        }
    }

    public ArrayList<College> managerSelectCollege() {
        String sql = "exec dbo.selectCollege ";
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<College> collegeList = new ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询系表失败");
        } else {
            try {
                while (resultSet.next()) {
                    College college = new College(
                            resultSet.getString("collegeName"),
                            resultSet.getString("collegeNo")
                    );
                    collegeList.add(college);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询系表失败");
            }
        }
        return collegeList;
    }

    public ArrayList<Class> managerSelectClass() {
        String sql = "exec dbo.selectClass ";
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Class> classList = new ArrayList<>();
```

```
        if (resultSet == null) {
            System.out.println("查询班级表失败");
        } else {
            try {
                while (resultSet.next()) {
                    Class classObj = new Class(
                            resultSet.getString("className"),
                            resultSet.getString("classNo"),
                            resultSet.getString("classProName"),
                            resultSet.getString("classColNo")
                    );
                    classList.add(classObj);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询班级表失败");
            }
        }
        return classList;
    }

    public ArrayList<Teacher> managerSelectTeacher() {
        String sql = "exec dbo.selectTeacher ";
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Teacher> teacherList = new ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询教师表失败");
        } else {
            try {
                while (resultSet.next()) {
```

```
                Teacher teacher = new Teacher(
                        resultSet.getString("teacherName"),
                        resultSet.getString("teacherNo"),
                        resultSet.getString("teacherResDir"),
                        resultSet.getString("teacherColNo")
                );
                teacherList.add(teacher);
            }
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("查询教师表失败");
        }
    }
    return teacherList;
}

public ArrayList<Student> managerSelectStudent() {
    String sql = "exec dbo.selectStudent ";
    ResultSet resultSet = executeQueryProcedure(sql);
    ArrayList<Student> studentList = new ArrayList<>();
    if (resultSet == null) {
        System.out.println("查询学生表失败");
    } else {
        try {
            while (resultSet.next()) {
                Student student = new Student(
                        resultSet.getString("studentName"),
                        resultSet.getString("studentNo"),
                        resultSet.getString("studentAge"),
                        resultSet.getString("studentSex"),
                        resultSet.getString("studentNation"),
                        resultSet.getString("studentBirthday"),
                        resultSet.getString("studentInsNo"),
                        resultSet.getString("studentClassNo")
                );
                studentList.add(student);
            }
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("查询学生表失败");
        }
    }
    return studentList;
}

public ArrayList<Course> managerSelectCourse() {
    String sql = "exec dbo.selectCourse ";
    ResultSet resultSet = executeQueryProcedure(sql);
    ArrayList<Course> courseList = new ArrayList<>();
    if (resultSet == null) {
        System.out.println("查询课程表失败");
    } else {
        try {
            while (resultSet.next()) {
                Course course = new Course(
                        resultSet.getString("courseName"),
                        resultSet.getString("courseNo"),
                        resultSet.getInt("courseCredict"),
                        resultSet.getInt("coursePeriod")
                );
```

```
                    courseList.add(course);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询课程表失败");
            }
        }
        return courseList;
    }

    public ArrayList<Pair<String, String>> managerSelectCollegeManager() {
        String sql = "exec dbo.selectCollegeManager ";
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Pair<String, String>> collegeManagerList = new ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询系-系主任表失败");
        } else {
            try {
                while (resultSet.next()) {
                    Pair<String, String> collegeManager = new Pair<>(
                            resultSet.getString("collegeNo"),
                            resultSet.getString("teacherNo")
                    );
                    collegeManagerList.add(collegeManager);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询系-系主任表失败");
            }
        }
        return collegeManagerList;
    }

    public ArrayList<Pair<String, String>> managerSelectCollegeCourse() {
        String sql = "exec dbo.selectCollegeCourse ";
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Pair<String, String>> collegeCourseList = new ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询系-课程表失败");
        } else {
            try {
                while (resultSet.next()) {
                    Pair<String, String> collegeCourse = new Pair<>(
                            resultSet.getString("collegeNo"),
                            resultSet.getString("courseNo")
                    );
                    collegeCourseList.add(collegeCourse);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询系-课程表失败");
            }
        }
        return collegeCourseList;
    }

    public ArrayList<Pair<String, String>> managerSelectClassMonitor() {
        String sql = "exec dbo.selectClassMonitor ";
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Pair<String, String>> classMonitorList = new ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询班级-班长表失败");
        } else {
            try {
                while (resultSet.next()) {
                    Pair<String, String> classMonitor = new Pair<>(
                            resultSet.getString("classNo"),
                            resultSet.getString("studentNo")
                    );
                    classMonitorList.add(classMonitor);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询班级-班长表失败");
            }
        }
        return classMonitorList;
    }

    public ArrayList<Pair<String, String>> managerSelectTeacherCourse() {
        String sql = "exec dbo.selectTeacherCourse ";
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Pair<String, String>> teacherCourseList = new ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询教师-课程表失败");
        } else {
            try {
                while (resultSet.next()) {
                    Pair<String, String> teacherCourse = new Pair<>(
                            resultSet.getString("teacherNo"),
                        resultSet.getString("courseNo")
                    );
                    teacherCourseList.add(teacherCourse);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询教师-课程表失败");
            }
        }
        return teacherCourseList;
    }

    public ArrayList<Quartet<String, String, String, Integer>> managerSelectStuCouTeaSco()
{
        String sql = "exec dbo.selectStuCouTeaSco ";
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Quartet<String, String, String, Integer>> stucouteascoList = new
ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询学生-课程-教师-成绩失败");
        } else {
            try {
                while (resultSet.next()) {
                    Quartet<String, String, String, Integer> stucouteasco = new Quartet<>(
                        resultSet.getString("studentNo"),
                        resultSet.getString("courseNo"),
                        resultSet.getString("teacherNo"),
                        resultSet.getInt("score")
                    );
                    stucouteascoList.add(stucouteasco);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询学生-课程-教师-成绩表失败");
            }
        }
        return stucouteascoList;
    }


    /******************************/

    /**
     * 学生
     **/
    public boolean studentInsertSCTSTable(String studentNo, String courseNo, String
teacherNo) {
        String sql = "exec dbo.studentAddSCTTable " + modifyStr(studentNo) + "," +
modifyStr(courseNo)
                + "," + modifyStr(teacherNo) + "," + String.valueOf(0);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("选课成功");
            return true;
        } else {
            System.out.println("表中已有该选课信息，选课失败");
            return false;
        }
    }
    public boolean studentDeleteSCTSTable(String studentNo, String courseNo) {
        String sql = "exec dbo.studentDeleteSCTTable " + modifyStr(studentNo) + "," +
modifyStr(courseNo);
        boolean result = executeUpdateProcedure(sql);
        if (result) {
            System.out.println("退课成功");
            return true;
        } else {
            System.out.println("表中无该选课信息，退课失败");
            return false;
        }
    }

    public boolean studentUpdatePassword(String userName, String userOldPassword, String
userNewPassword) {
        String exesql = "exec dbo.updatePassword " + modifyStr(userName) + "," +
modifyStr(userOldPassword) + ","
                + modifyStr(userNewPassword);
        boolean result = executeUpdateProcedure(exesql);
        if (result) {
            System.out.println("更新密码成功");
            return true;
        } else {
            System.out.println("表中无该用户信息或旧密码不正确，更新密码失败");
            return false;
        }
    }

    public ArrayList<Quartet<String, String, String, String>> studentSelectCollege(String
studentNo) {
        String sql = "exec dbo.studentSelectCollege " + modifyStr(studentNo);
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Quartet<String, String, String, String>> studentCollegeList = new
ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询学生所在系信息失败");
        } else {
            try {
                while (resultSet.next()) {
                    Quartet<String, String, String, String> studentCollege = new
Quartet<>(
                            resultSet.getString("collegeNo"),
                            resultSet.getString("collegeName"),
                            resultSet.getString("teacherNo"),
                            resultSet.getString("teacherName")
                    );
                    studentCollegeList.add(studentCollege);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询学生所在系信息失败");
            }
        }
        return studentCollegeList;
    }

    public ArrayList<Sextet<String, String, String, String, String, String>>
studentSelectClass(String studentNo) {
        String sql = "exec dbo.studentSelectClass " + modifyStr(studentNo);
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Sextet<String, String, String, String, String, String>> studentClassList
= new ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询学生所在班级信息失败");
        } else {
            try {
                while (resultSet.next()) {
                    Sextet<String, String, String, String, String, String> studentCollege
= new Sextet<>(
                            resultSet.getString("classNo"),
                            resultSet.getString("className"),
                            resultSet.getString("classProName"),
                            resultSet.getString("classColNo"),
                            resultSet.getString("studentNo"),
                            resultSet.getString("studentName")
                    );
                    studentClassList.add(studentCollege);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询学生所在班级信息失败");
            }
        }
        return studentClassList;
    }

    public   ArrayList<Quintet<String,   String,   String,   Integer,   Integer>>
studentSelectCourse(String studentNo) {
        String sql = "exec dbo.studentSelectCourse " + modifyStr(studentNo);
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Quintet<String, String, String, Integer, Integer>> studentCourseList =
new ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询学生选课信息失败");
        } else {
            try {
                while (resultSet.next()) {
                    Quintet<String, String, String, Integer, Integer> studentCourse = new
Quintet<>(
                            resultSet.getString("courseNo"),
                            resultSet.getString("teacherNo"),
                            resultSet.getString("courseName"),
                            resultSet.getInt("courseCredict"),
                            resultSet.getInt("coursePeriod")
                    );
                    studentCourseList.add(studentCourse);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询学生选课信息失败");
            }
        }
        return studentCourseList;
    }

    public ArrayList<Course> studentSelectCourse() {
        String sql = "exec dbo.selectCourse ";
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Course> courseList = new ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询课程表失败");
        } else {
            try {
                while (resultSet.next()) {
                    Course course = new Course(
                            resultSet.getString("courseName"),
                            resultSet.getString("courseNo"),
                            resultSet.getInt("courseCredict"),
                            resultSet.getInt("coursePeriod")
                    );
                    courseList.add(course);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询课程表失败");
            }
        }
        return courseList;
    }

    public ArrayList<Teacher> studentSelectTeacher() {
        String sql = "exec dbo.selectTeacher ";
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Teacher> teacherList = new ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询教师表失败");
        } else {
            try {
                while (resultSet.next()) {
                    Teacher teacher = new Teacher(
                            resultSet.getString("teacherName"),
                            resultSet.getString("teacherNo"),
                            resultSet.getString("teacherResDir"),
                            resultSet.getString("teacherColNo")
                    );
                    teacherList.add(teacher);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询教师表失败");
            }
        }
        return teacherList;
    }

    public Student studentSelectMessage(String studentNo) {
        String sql = "exec dbo.studentSelectMessage " + modifyStr(studentNo);
        ResultSet resultSet = executeQueryProcedure(sql);
        Student student = null;
        if (resultSet == null) {
            System.out.println("查询学生个人信息失败");
        } else {
            try {
                resultSet.next();
                student = new Student(
                        resultSet.getString("studentName"),
                        resultSet.getString("studentNo"),
                        resultSet.getString("studentAge"),
                        resultSet.getString("studentSex"),
                        resultSet.getString("studentNation"),
                        resultSet.getString("studentBirthday"),
                        resultSet.getString("studentInsNo"),
                        resultSet.getString("studentClassNo")
                );

            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询学生个人信息失败");
            }
        }
        return student;
    }

    public ArrayList<Quartet<String, String, String, Integer>> studentSelectScore(String
studentNo) {
        String sql = "exec dbo.studentSelectScore " + modifyStr(studentNo);
        ResultSet resultSet = executeQueryProcedure(sql);
        ArrayList<Quartet<String, String, String, Integer>> studentScoreList = new
ArrayList<>();
        if (resultSet == null) {
            System.out.println("查询学生个人成绩失败");
        } else {
            try {
                while (resultSet.next()) {
                    Quartet<String, String, String, Integer> studentScore = new Quartet<>(
                            resultSet.getString("studentNo"),
                            resultSet.getString("courseNo"),
                            resultSet.getString("teacherNo"),
                            resultSet.getInt("score")
                    );
                    studentScoreList.add(studentScore);
                }
            } catch (SQLException e) {
                e.printStackTrace();
                System.out.println("查询学生个人成绩失败");
            }
        }
        return studentScoreList;
    }
}
```

2.10 收获与体会

通过这次数据库课程设计，我在很过方面都有所提高。从各种文档的阅读到开始的需求分析、概念结构设计、逻辑结构设计、系统实现，我亲身体验了一回从后端到前端的系统开发流程。虽然很多东西书上写的很清楚，老师课上也讲的很清晰，但实际运用起来还得花费一番苦功夫，在设计表格的时候，我发现自己一开始设计的表格循环函数依赖了，导致 SQLServer 导入表格不成功，经过反复修改和设计才解决这个问题。在完成这次课程设计的过程中，我加深了对数据库的创建、修改、删除的方法的理解，通过导入表、删除表和修改表熟悉了表的操作，通过写 java 程序实现这一个小型应用，让我对前后端分离的软件工程设计有了初步认识，代码能力得到加强。

一开始我对数据库课设完全没有概念，感觉完成这个任务堪比登天。但是当我认真静下心来去查询资料、请教他人的时候，发现这个课程设计也不是那么难。所以我们无论做什么都要相信自己有能力做到，不惧困难，迎难而上，这才是数据库课程设计带给我们最好的东西。