****

《软件工程》课程考核报告

**题目：学籍管理软件系统文档**

组长 黄阳 学号 2017219009 专业班级 物联网17-2

组员 文华 学号 2017218007 专业班级 物联网17-2

组员 刘振 学号 2017218178 专业班级 物联网17-2

组员 刘城浩 学号 2017217995 专业班级 物联网17-2

指导教师 谢昭

**20220** 年 6 月 21 日

**摘要：**

近年来信息技术在多个领域的发展迅速，信息技术在管理上已经广泛应用并且已经日渐成熟，日后还会不断发展，信息管理系统可以提高信息管理的效率。我们小组使用IntelliJ IDEA 和MySQL为开发工具，以Windows10为开发平台设计出一 个学籍管理系统，对学生的基本信息、课程、成绩进行统一管理。主要实现的功能模块有学生管理模块、课程管理模块、成绩管理模块。

目前，系统功能基本完善，可以对学生信息进行有效地管理。

**关键字：**

学籍管理；MySQL

**Abstract：**

In recent years, information technology has developed rapidly in many fields. Information technology has been widely used in management and has become increasingly mature. It will continue to develop in the future.Using IntelliJ IDEA and MySQL as development tools, our team designed a school roll management system using Windows10 as development platform to manage students' basic information, courses and grades. The main functional modules are student management module, course management module and grade management module.At present, the function of the system is basically perfect, which can effectively manage student information.

**Key words:**

student status management; MySQL

1 引言

学籍管理师学校教务管理的重要工作之一，学籍信息记录了学生的基本信息和在校的真实表现，对学生和学校极为重要，因此档案管理必须严肃、认真，这两点便以为这工作量巨大。

学校学生数量庞大，人工书写数据基本无法处理这庞大的数据，计算机技术和数据库技术的发展可以对学校的学籍管理工作进行高效的处理。因此，设计一个学校学籍管理系统对学校学生的学籍进行管理，可以在短时间内高效的管理学校学生的学籍。

由于以上的原因，收集相关资料和数据，查阅有关文献及技术参数，对高校学生学籍管理的实际需求等方面进行了调研，发现目前所采用的手工记录的方法进行管理存在以下几个方面的弊端:对象范围广，数据存储不易;填写的数据多，每个学生所填写的内容相当复杂，且大量数据重复;不易存档，成千上万的信息和堆积如山的单据对管理人员来说是一个负担，需要大量的文档资料。鉴于以上种种原因，高校学生学籍管理的管理急需一种软件来代替手工操作来管理数据资料。

计算机技术在现代管理中的应用，使计算机成为领导者和管理人员应用现代技术的重要工具”。计算机辅助学校管理活动，可以极大地增强管理者采集、处理信息的能力，从而有利于管理者及时决策。计算机系统能根据管理过程的变化情况，将原始数据、资料等进行加工、保存，管理人员可以在解决具体问题需要信息资料时，随时进行检索查询，了解整个学校学生学籍管理系统的动态情况，进行动态管理，从而有效的处理学校学生学籍管理的管理工作，实现对学校学生学籍管理信息管理的自动化，提高效率。

高校学生学籍管理系统的开发目的是使高校的高校学生学籍管理模式从手工记录转变成信息管理，为高校学生学籍管理人员提供方便条件。对高校学生的实际情况进行调研之后，进行详细的需求分析，对现有的管理模式进行改进，开发出一套新型的管理系统，从中领悟系统开发的思想，掌握系统开发的流程和方法。随着高校信息系统的不断完善，高校的管理将越来越依赖于信息化的管理。所以系统开发将围绕高校学生学籍管理工作的实际情况，使之能迅速适应高校的需要。

高校学生学籍管理系统的实施的现实意义:减少管理高校学生学籍管理的工作人员;管理人员可以随时浏览查看，而且更加直观;改变了以前手工记录的方式，电脑录入更加方便快捷;实现了高校学生学籍管理的计算机化。

2 需求分析

负责人：刘城浩

2.1 功能需求分析

这个是一个学生管理系统，我们的目标是提高学籍管理工作的效率，具有对学生个人信息，课程信息，所选课程信息进行管理及维护的功能。学生可以通过此系统进行个人信息，考试成绩查询。系统开发的总体任务是实现学生信息管理的系统化，规范化和自动化。

总之，本学籍管理系统的功能需求即主要是管理员对功能的需求：

管理员权限最大，可以对学生，课程，老师的管理，管理员对功能的需求细分如下：

1、有关学生信息的浏览，包括学生的姓名，入学日期，所在班级，学制，家庭住址，联系电话。

2、学生的添加，本系统由管理员对学生进行添加。

3、学生信息的修改，删除。

4、教师信息的添加，本系统管理员对教师的基本信息进行添加。

5、学校基本课程的浏览，包括课程名，学时，学分，任课老师和课程具体描述。

6、学校基本课程的添加。

7、学校基本课程的修改，删除。

2.2 性能需求分析：

系统应具有良好的可操作性，界面操作应方便简单、使用户能够快速的找到所需要的信息并进行操作。

系统应具有较强的安全保密性。系统应提供安全可靠的数据操作权限，系统数据被篡改可能会造成重大的事故，所以系统的安全保密性一定要有所保证。

系统应具有高可靠性系统在功能完整的同时，应具有很好的可靠性。作为支撑学生学籍管理工作的基础环境，系统出现故障将直接影响学校正常的工作进程，所以系统必须具有高可靠性。

2.3 出错处理需求分析：

系统在面对一些错误数据或非法操作时要有一定的自动识别行性，在发现错误之后进行提示报错，数据错误或操作非法，停止对数据的操作或停止非法操作。若是访问数据时出错，则提示报错。

2.4 接口需求分析：

本操作系统的目标是提高管理员对学生学籍等基本信息管理工作的效率，所以各个界面必须简洁易懂，以求能狗哦最大程度的提高工作效率。

用户登录及进行操作的界面都必须简洁易懂，由于我们是仅供管理员使用的管理系统，所以在登陆及操作界面就无需对老师或学生的需求进行考虑，这进一步简化了界面。

登陆界面只需包含基本的信息，如账号、密码的输入框；注册、登录、退出的按钮；操作界面则包含对信息的登记、查询、修改等基本功能。

2.5可扩展性需求分析：

目前我们所做的是仅供管理员使用的操作系统，未来由于使用需要可能需要加入学生端与教师端，这就需要保证特定的群体只能对特定的信息进行特定的操作，这就要求我们建立更加完善的数据库系统以及多元化的操作界面，由于时间及选题原因，这些问题在本报告中目前暂不予考虑。

3 系统总体设计

负责人：黄阳（状态转换图由刘振完成）

3.1 系统总体模块

我们小组设计的学籍管理系统主要包括学生管理模块、课程管理模块、成绩管理模块

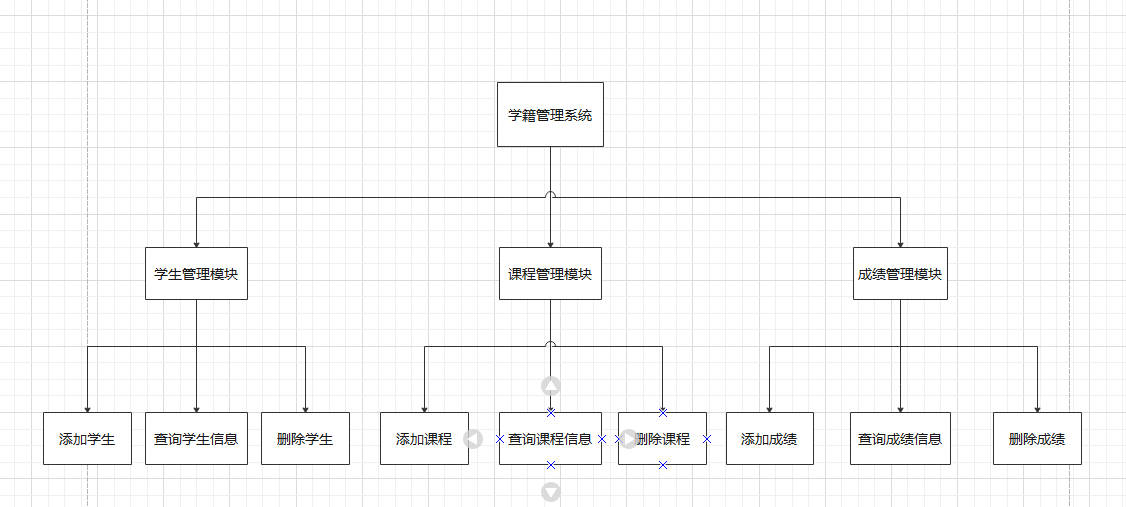


图3.1 系统总体模块图

3.11 学生管理模块

该模块主要负责管理所有在校学生的个人信息。管理者通过这个模块，可以添加、更新、删除、查询学生的学籍信息。

3.12 课程管理模块

该模块主要的功能是管理全校的课程信息。管理员可以添加，删除，查询课程信息，也可以对课程信息进行维护的权限（增加、删除、更新）。

3.13 成绩管理模块

管理员可以对成绩进行添加、删除、查询处理。

3.2系统用例

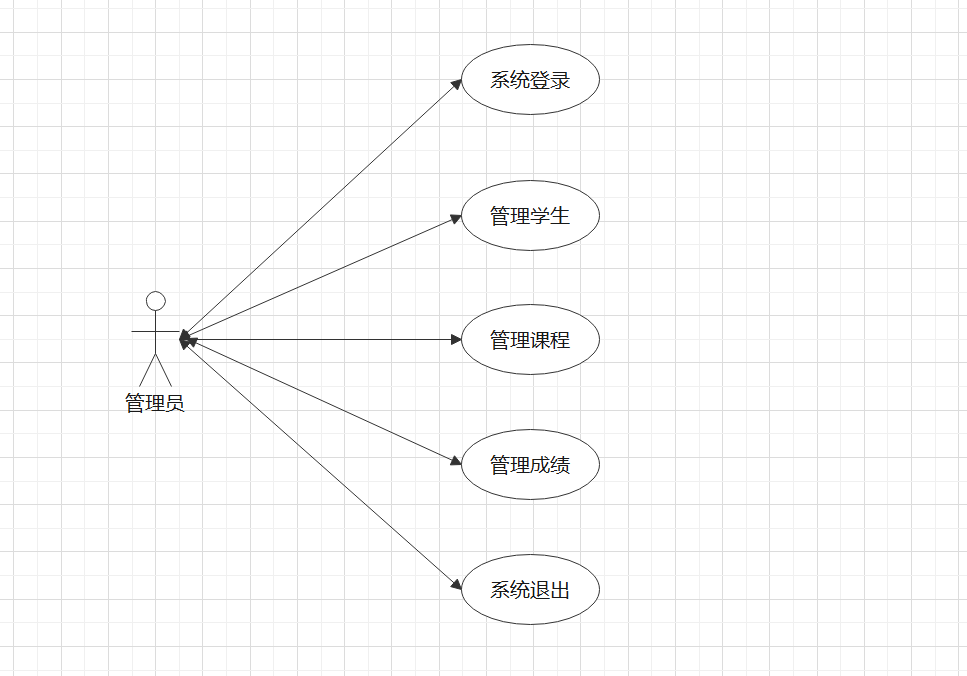


图3.2 管理员例图

管理员例图，管理员（admin）可以实现的功能主要有：系统的登陆、对学生的管理（增加、删除、更新、查看）、对学生的管理（增加、删除、更新、查看）、对课程的管理（增加、删除、更新、查看）以及对新闻的管理（增加、删除、更新、查看）。

3.3业务流程图

系统流程图是描绘物理系统的传统工具。它的基本思想是用图形符号以黑盒子形式描述系统里面的每个部件(程序,文件,数据库,表格,人工过程等等)。系统流程图表达的是信息在系统各部件之间流动的情况,而不是对信息进行加工处理的控制过程,因此尽管系统流程图使用的某些符号和程序流程图中使用的符号相同,但是它确是物理流程图而不是程序流程图。

3.31 系统登录流程图

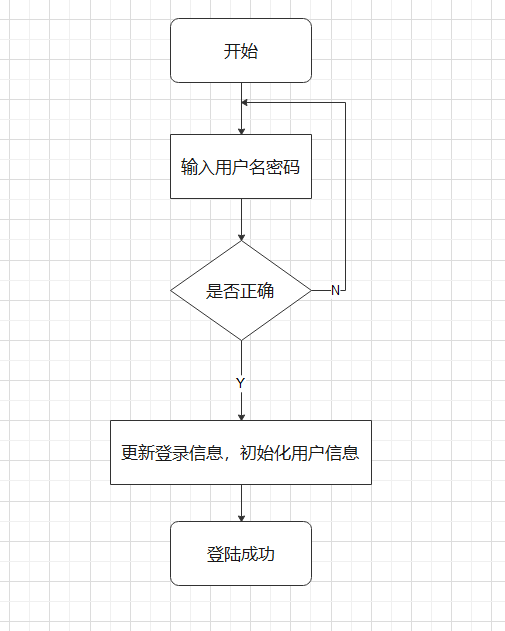


图3.3 登录流程图

3.32 信息管理流程图

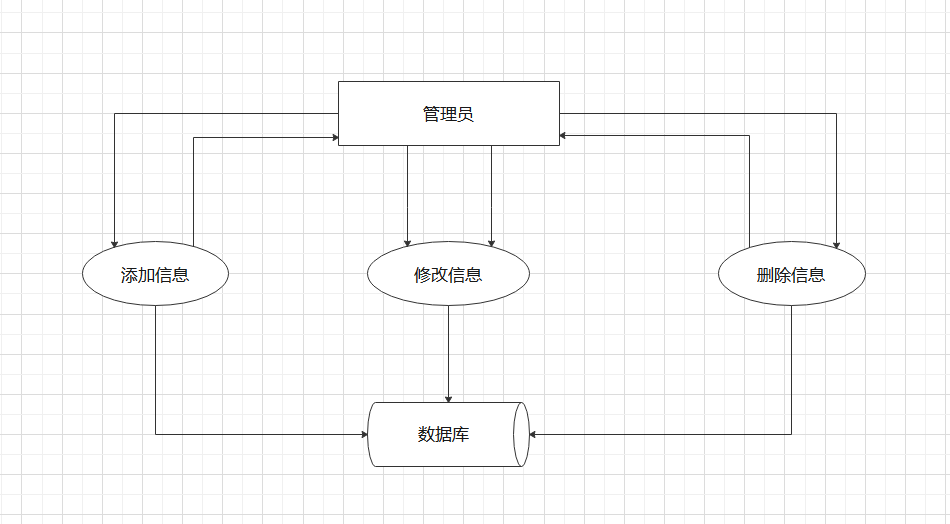


图3.4 信息管理流程图

3.3 状态转换图

学籍信息管理系统的脚本如下

正常情况脚本

————————————————————————————————

打开系统界面

学籍管理系统要求密码且用户输入密码

系统验证密码正确

系统显示所有服务

用户选择其中一个服务

系统提供此服务

用户浏览或操作完毕

系统请求是否继续，用户回答“不”

系统退出

—————————————————————————————————

异常情况脚本

—————————————————————————————————

打开系统界面

学籍管理系统要求密码且用户输入密码

系统验证密码错误，请求用户重新输入，再次输入后经验证正确。

系统显示所有服务

用户选择其中一个服务

系统提供此服务失败，显示错误信息

—————————————————————————————————

根据此脚本画出状态图。

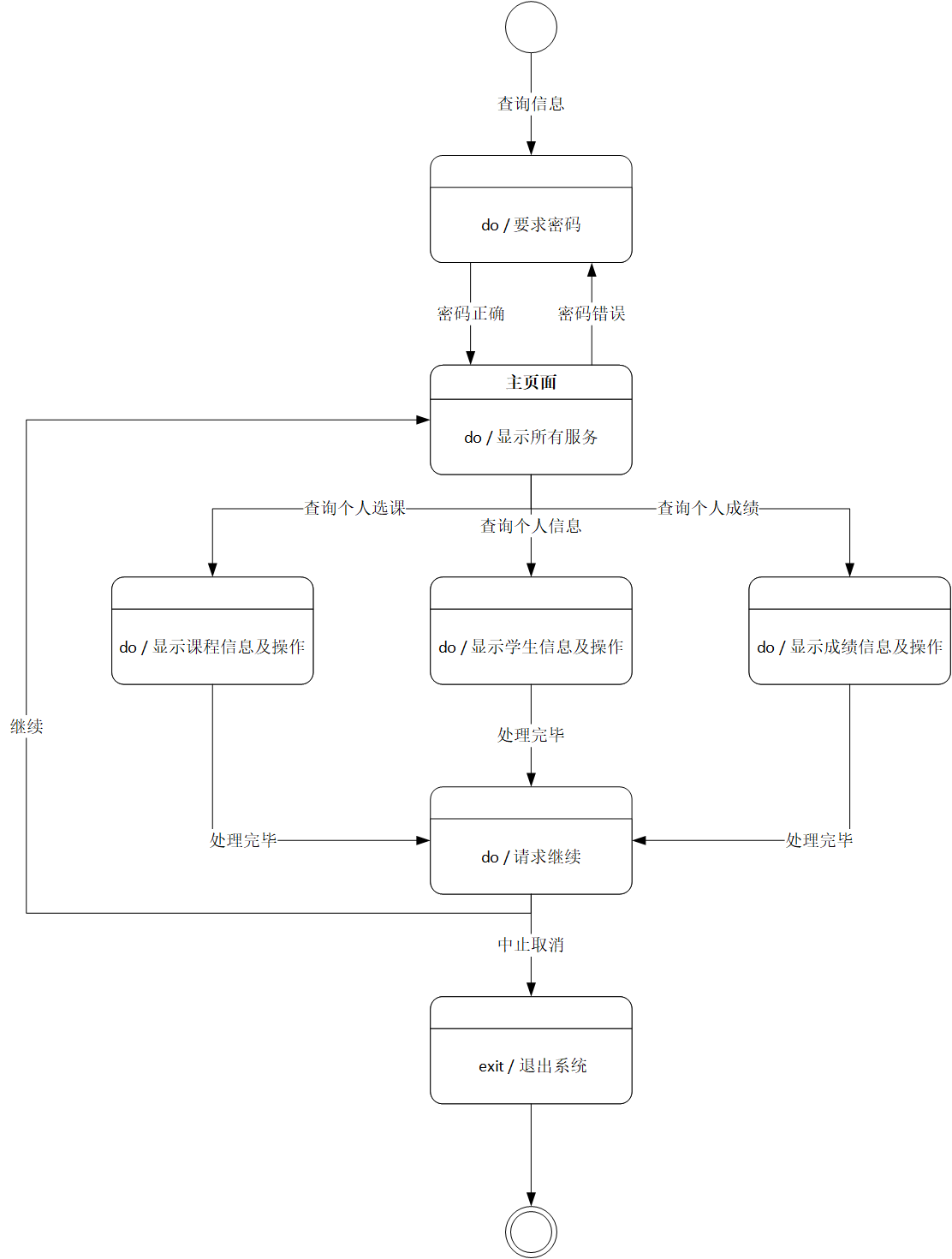


图3.5 状态转换图1

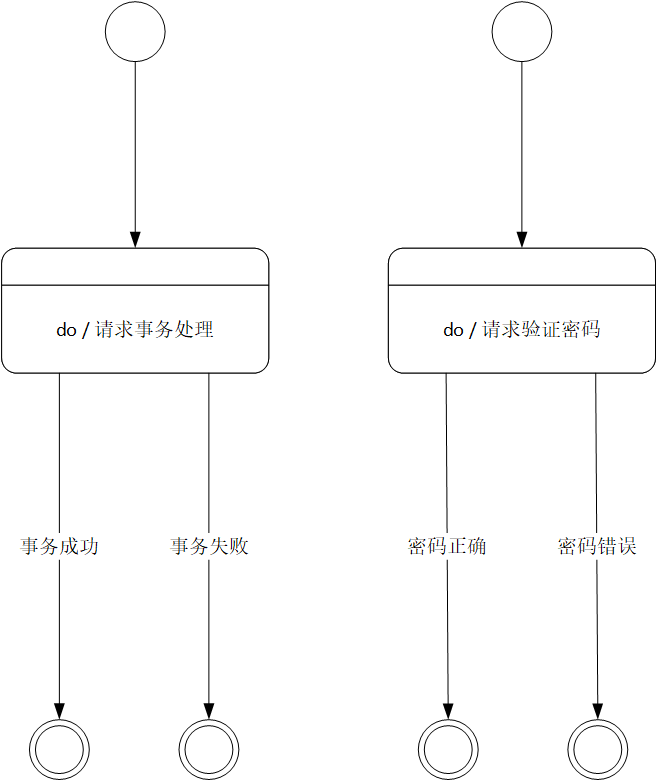


图3.6 状态转换图2

4数据库设计

负责人：刘振（代码由刘振、黄阳、刘城浩共同实现）

4.1 数据库概念结构设计

4.1.1 E-R图

根据学籍管理系统对学生、课程、管理员的要求，找到与之相关的属性，经过筛选，去除冗余后，分别得到学生、课程、管理员的属性，并根据它们的关系画出E-R图。

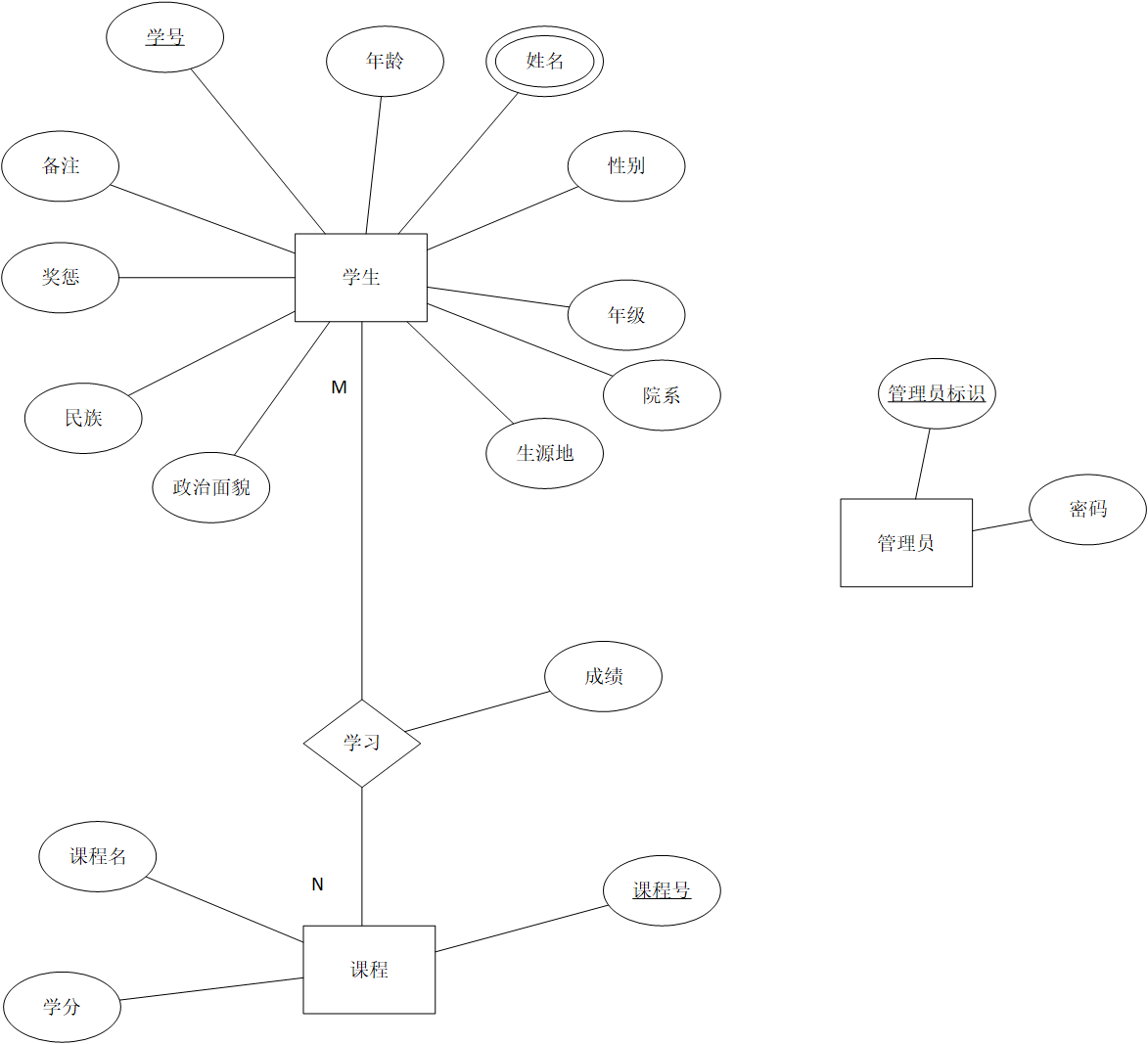


图4.1 E-R图

4.2 数据库逻辑结构设计

**课程数据表（course）：**

（课程号，课程名，学分） —— (cno, cname, credit)

主码：课程号（cno）

**成绩数据表（sc）：**

（~~学号~~，*~~课程号~~*，成绩） —— (id, *cno*, grade)

主码：学号，课程号（id，cno）

外码：学号（id）、课程号（cno）

索引：课程号（cno）

**学生数据表（student）：**

（学号，年龄，*姓名*，性别，年级，生源地，院系，政治面貌，民族，奖惩，备注） —— (id, age, *name*, sex, year, address, dept, political, nationality, record, flag)

主码：学号（id）

索引：姓名（name）

**管理员数据表（tb\_admin）：**

（管理员标识，密码） —— (admin\_id, pwd)

主码：管理员标识(admin\_id)

5 可视化界面

负责人：文华

5.1连接数据库

源代码

config文件：

url=jdbc:mysql://localhost/stumanager\_1?user=root&password=123&useUnicode=true&characterEncoding=UTF-8  
username=root  
password=123  
driver=com.mysql.jdbc.Driver

Java源码：

DBConnection.java：

package common;  
  
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
import java.util.Properties;  
  
public class DBConnection {  
 private Connection conn;  
   
 public void close() {  
 try {  
 conn.close();  
 } catch (SQLException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 }  
 }  
  
 public Connection getConn() {  
 if(conn!=null){  
 System.*out*.println("开启数据库连接--");  
 }  
 return conn;  
 }  
   
 public DBConnection(){  
 Properties prop = new Properties();  
 try {  
 //从外部文件读取数据库联接参数  
 prop.load(new FileInputStream("config/login.properties"));  
   
 String driver=prop.getProperty("driver");  
 String url=prop.getProperty("url");  
 String username=prop.getProperty("username");  
 String password=prop.getProperty("password");  
  
 System.*out*.println("url = " + url);  
 System.*out*.println("username = " + username);  
 System.*out*.println("password = " + password);  
   
 //加载驱动，获取数据库连接  
 Class.*forName*(driver);  
 conn=DriverManager.*getConnection*(url, username, password);  
   
 } catch (FileNotFoundException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 } catch (IOException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 } catch (ClassNotFoundException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 } catch (SQLException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 }  
 System.*out*.println("Done!");  
 }  
  
}

5.2登录界面

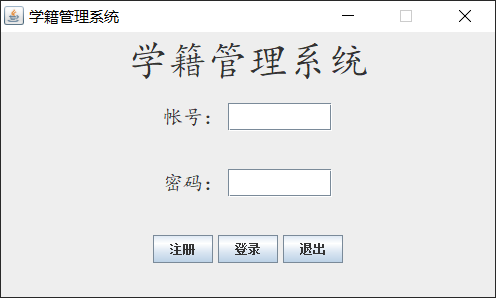


图5.1 登录界面

源代码：

package view;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.List;  
  
import javax.swing.JDesktopPane;  
import javax.swing.JFrame;  
import javax.swing.JInternalFrame;  
import javax.swing.JMenu;  
import javax.swing.JMenuBar;  
import javax.swing.JMenuItem;  
  
  
import javax.swing.JPanel;  
import javax.swing.JTable;  
import javax.swing.table.DefaultTableModel;  
  
import common.Common;  
import dao.Impl;  
  
  
public class AdminView extends JFrame {  
  
 JDesktopPane desk;  
 public AdminView(){  
   
 JMenuBar menubar = new JMenuBar();  
 this.setJMenuBar(menubar);  
 JMenu menu1 = new JMenu("基本信息登记修改");  
 JMenuItem item1\_1 = new JMenuItem("学生信息");  
 item1\_1.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new StudentView());  
 }  
 });  
 JMenuItem item1\_2 = new JMenuItem("课程信息");  
 item1\_2.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new CourseView());  
 }  
 });  
 JMenuItem item1\_3 = new JMenuItem("学生成绩信息");  
 item1\_3.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new StudentCourseView());  
 }  
 });  
 menu1.add(item1\_1);  
 menu1.add(item1\_2);  
 menu1.add(item1\_3);  
 menubar.add(menu1);  
 JMenu menu2 = new JMenu("基本信息更改");  
   
   
 JMenuItem item2\_1 = new JMenuItem("学生信息更改");  
   
 item2\_1.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new StudentEditDialog());  
 }  
 });  
   
   
 JMenuItem item2\_2 = new JMenuItem("学生成绩更改");  
   
 item2\_2.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new GradeEditDialog());  
 }  
 });  
   
   
   
 menu2.add(item2\_1);  
 menu2.add(item2\_2);  
 menubar.add(menu2);  
 JMenu menu3 = new JMenu("基本信息查询");  
 JMenuItem item3\_1 = new JMenuItem("按年龄统计人数");  
 item3\_1.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new SortByAgeView());  
 }  
 });  
   
   
   
 JMenuItem item3\_2 = new JMenuItem("按地区来源统计人数");  
   
 item3\_2.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new SortByRegionView());  
 }  
 });  
 JMenuItem item3\_3 = new JMenuItem("按政治面貌统计人数");  
 item3\_3.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new SortByPoliticalView());  
 }  
 });  
   
 JMenuItem item3\_4 = new JMenuItem("统计各专业某门课的学习成绩情况");  
 item3\_4.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new SortGradeView());  
 }  
 });  
   
 menu3.add(item3\_1);  
 menu3.add(item3\_2);  
 menu3.add(item3\_3);  
 menu3.add(item3\_4);  
 menubar.add(menu3);  
 desk = Common.*setDesk*("picture/2.jpg");  
 getContentPane().add(desk);  
 this.setTitle("管理员登录");  
 this.setSize(800, 500);  
 this.setLocationRelativeTo(null);  
 this.setResizable(true);  
 this.setVisible(true);  
 }  
   
   
}

6 系统测试

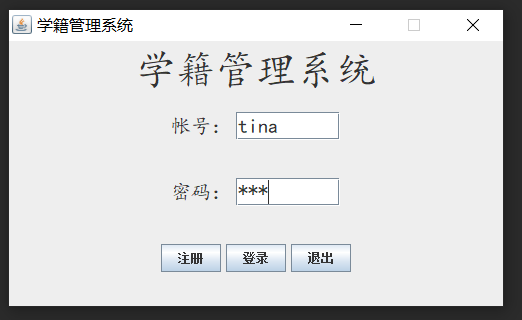
负责人：文华、黄阳、刘振、刘城浩

6.1各测试项目说明

1）用户登录窗口的测试

目的：保证登陆窗口的正确性与在错误发生时的容错与纠错性。

在登陆框中输入空用户名和错误的用户密码来检测系统的出错运行情况。要求系统在遇到这些情况时能给出正确的错误提示。





2）添加用户的测试

目的：保证添加用户的正确性与在数据输入不正确时的容错与纠错性。

在添加记录框中输入重复的用户名、空用户名和密码来检测系统的出错运行情况。要求系统在遇到这些情况时能给出正确的错误提示。

3）修改用户密码的测试

目的：保证修改用户密码窗口的正确性与在数据输入不正确时的容错与纠错性。

在修改用户密码框中输入错的原用户密码,以及空的新用户密码,来检测系统的出错运行情况。要求系统在遇到这些情况时能给出正确的错误提示。

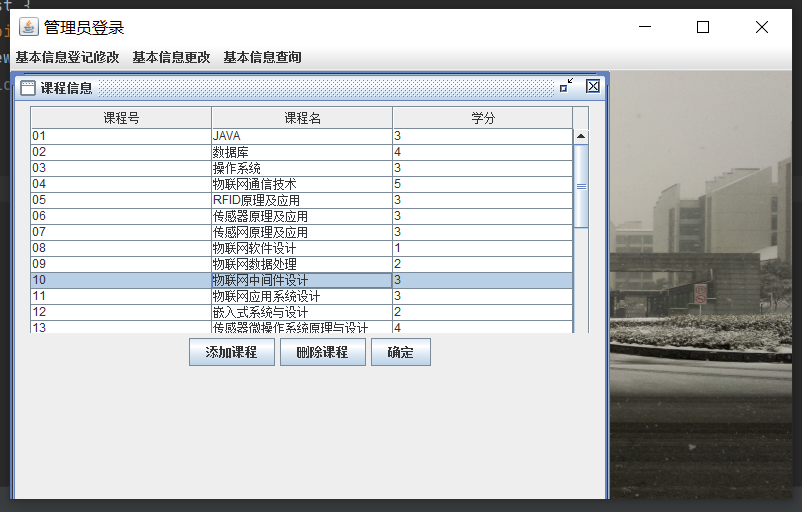
4）添加课程信息的测试

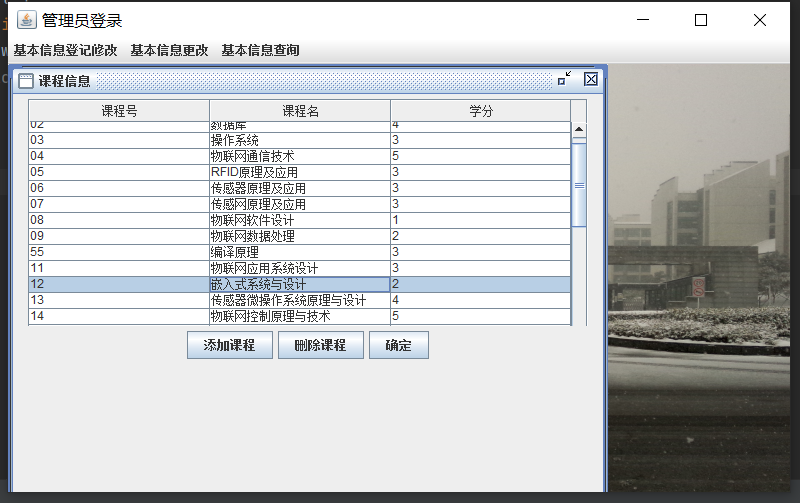
目的：保证添加课程信息的正确性与在数据输入不正确时的容错与纠错性。

在添加课程信息框中输入空的学号,已存在的学号，或者错误的日期格式,来检测系统的出错运行情况。要求系统在遇到这些情况时能给出正确的错误提示。

5）修改课程信息的测试

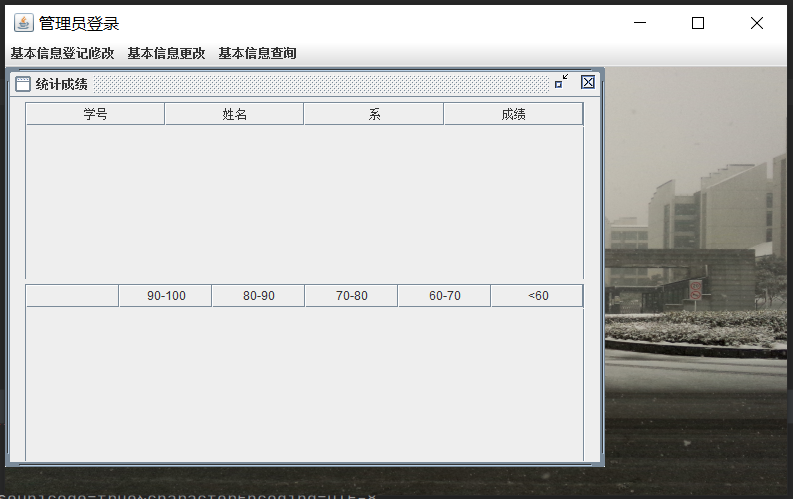
目的：保证修改课程信息的正确性与在数据输入不正确时的容错与纠错性。

在修改课程信息框中输入空的学号,已存在的学号，或者错误的日期格式,来检测系统的出错运行情况。要求系统在遇到这些情况时能给出正确的错误提示。



6）查询课程信息的测试

目的：是保证查询课程信息的正确性与在数据输入不正确时的容错与纠错性。

在查询课程的查询方式框中输入空的学号,空的姓名,或者空的班号,来检测系统的出错运行情况。要求系统在遇到这些情况时能给出正确的错误提示。

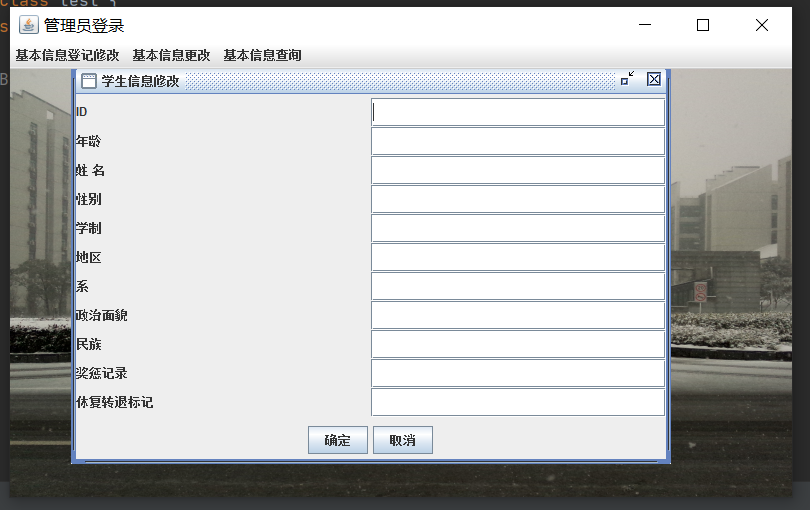
7）系统信息的测试

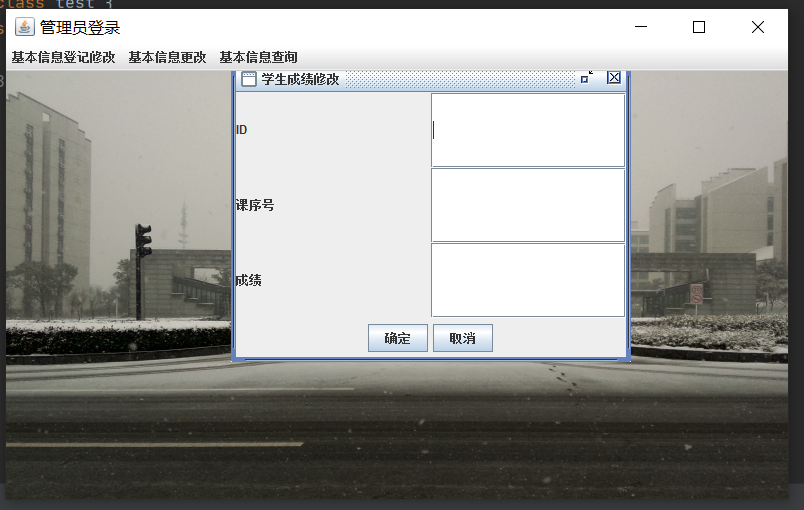
目的：保证输出的系统信息和作者信息的正确性与完整性。

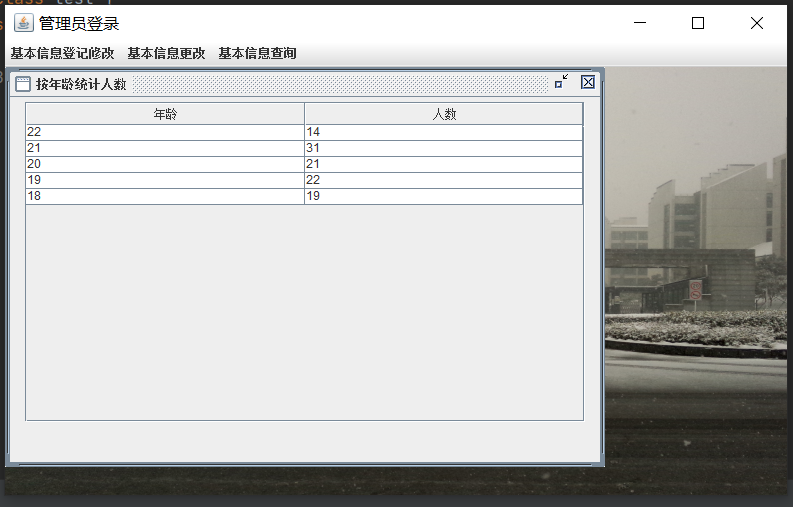




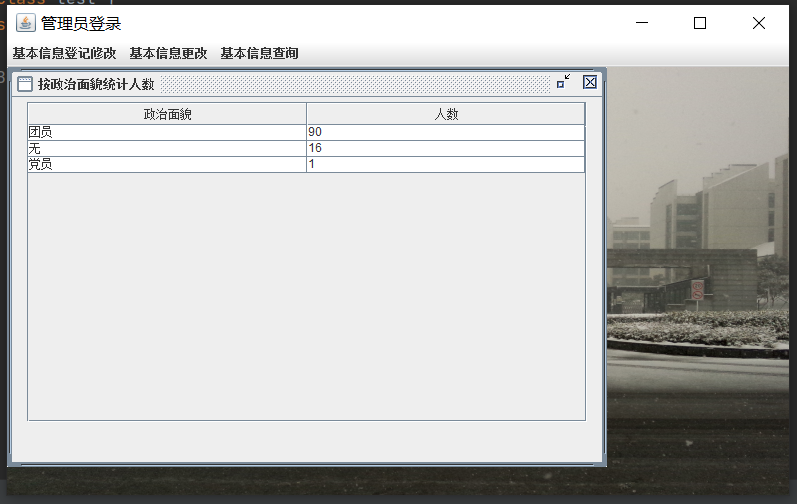


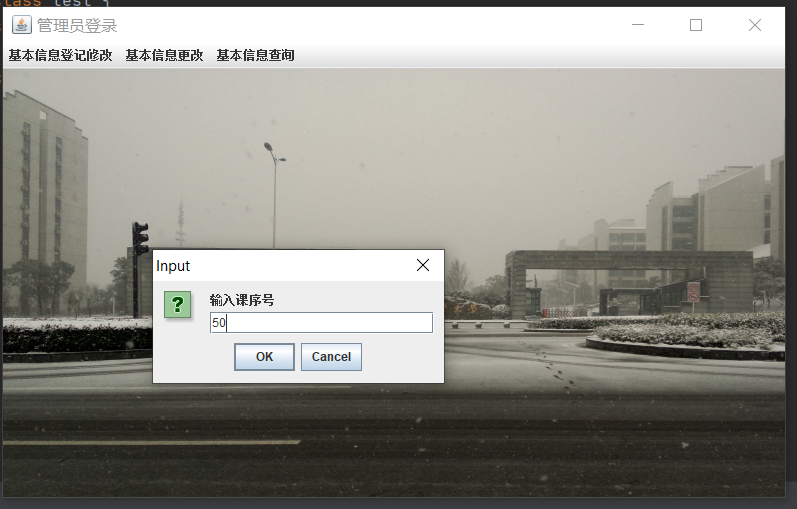












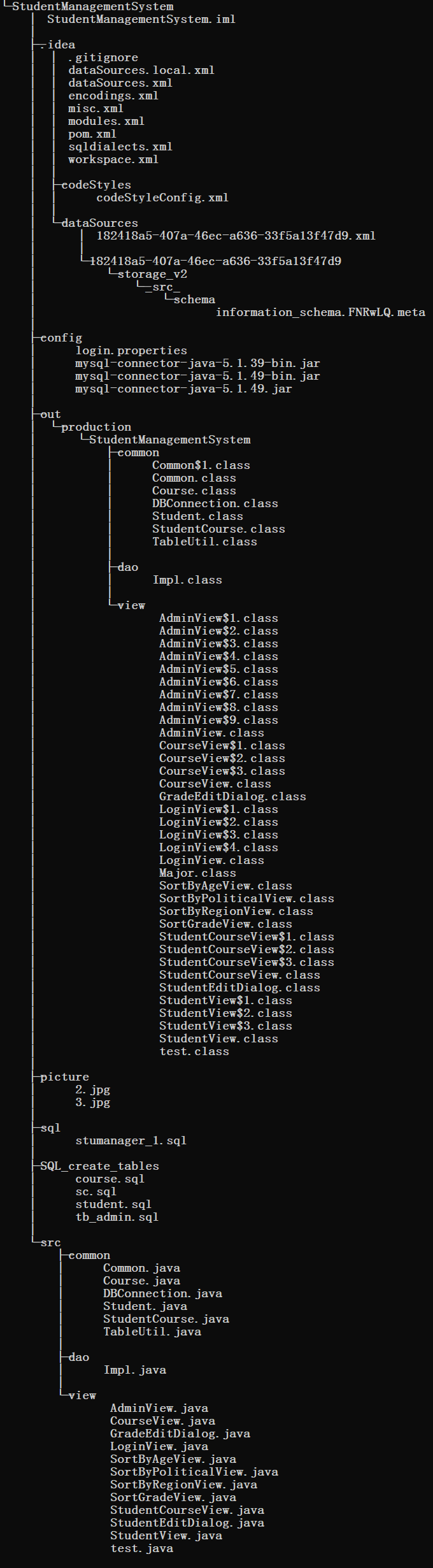


7 实验总结

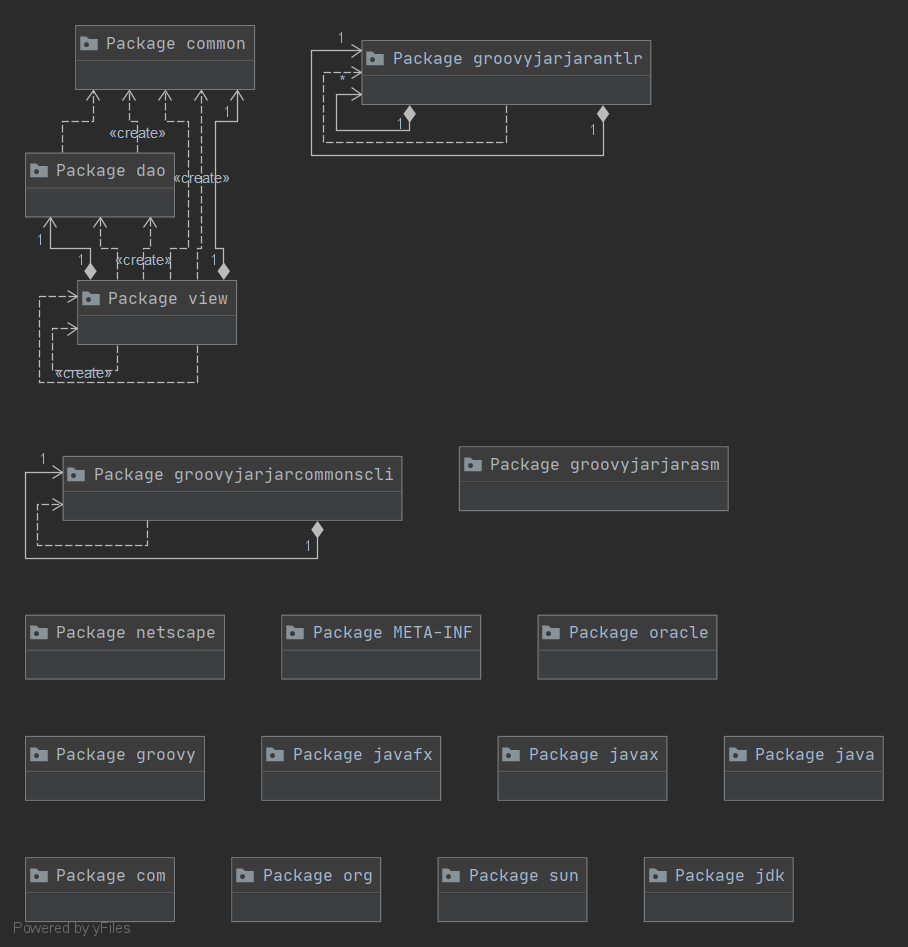
在本系统的开发过程中，小组成员在知识、经验方面都存在着许多的欠缺，此外在整个开发的过程中，时间也比较仓促。因此，该系统必然会存在一些缺陷和不足。由于对学生课程管理的整个流程不够熟悉，在需求分析时未能做到完全满足用户的需求。相信在今后的不断学习过程中我们小组成员会民族该系统的缺陷和不足

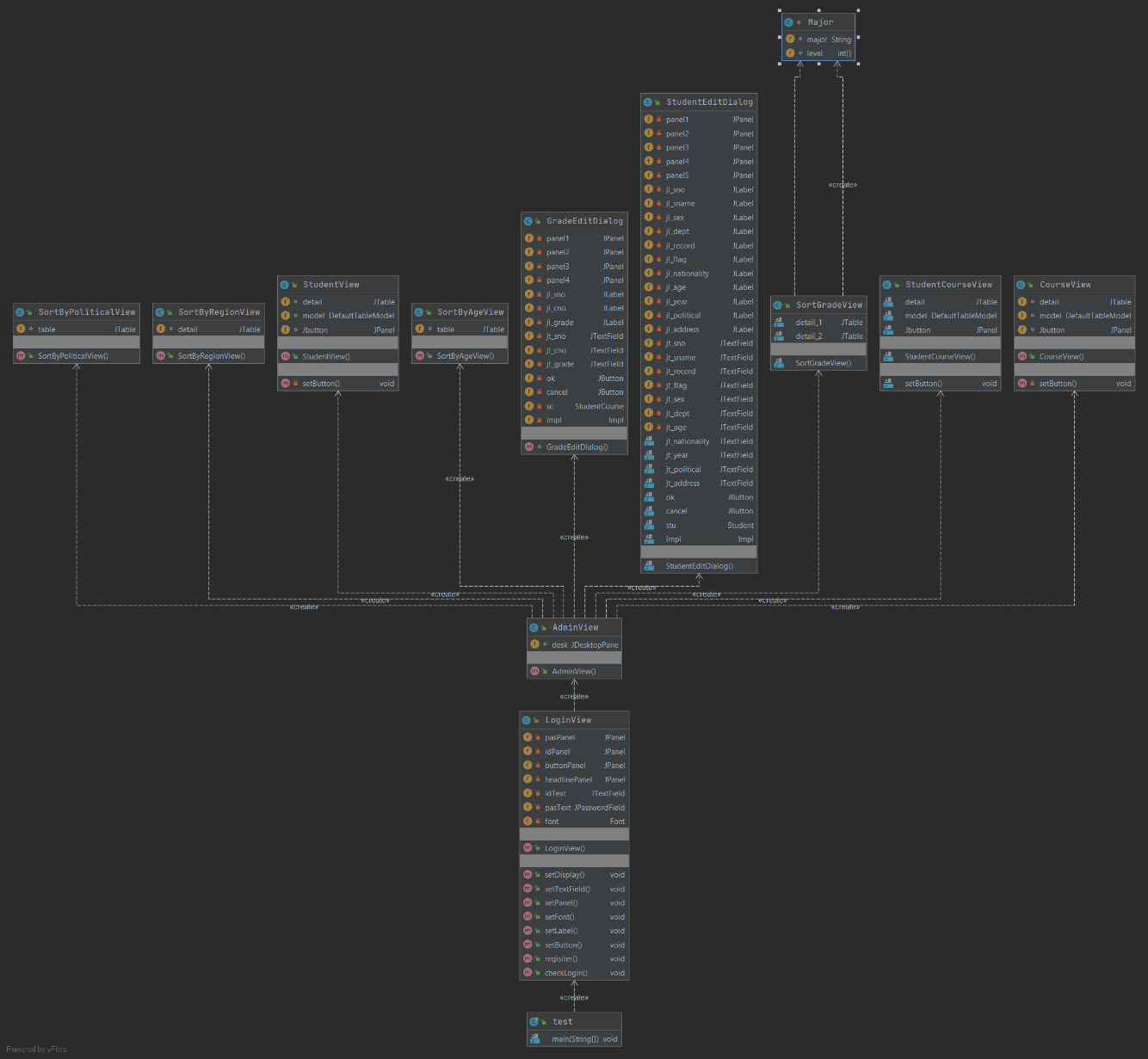
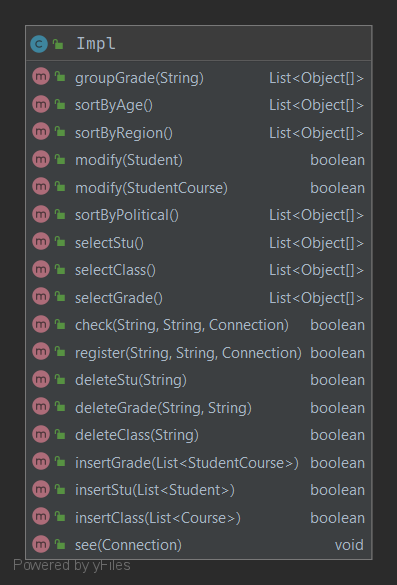
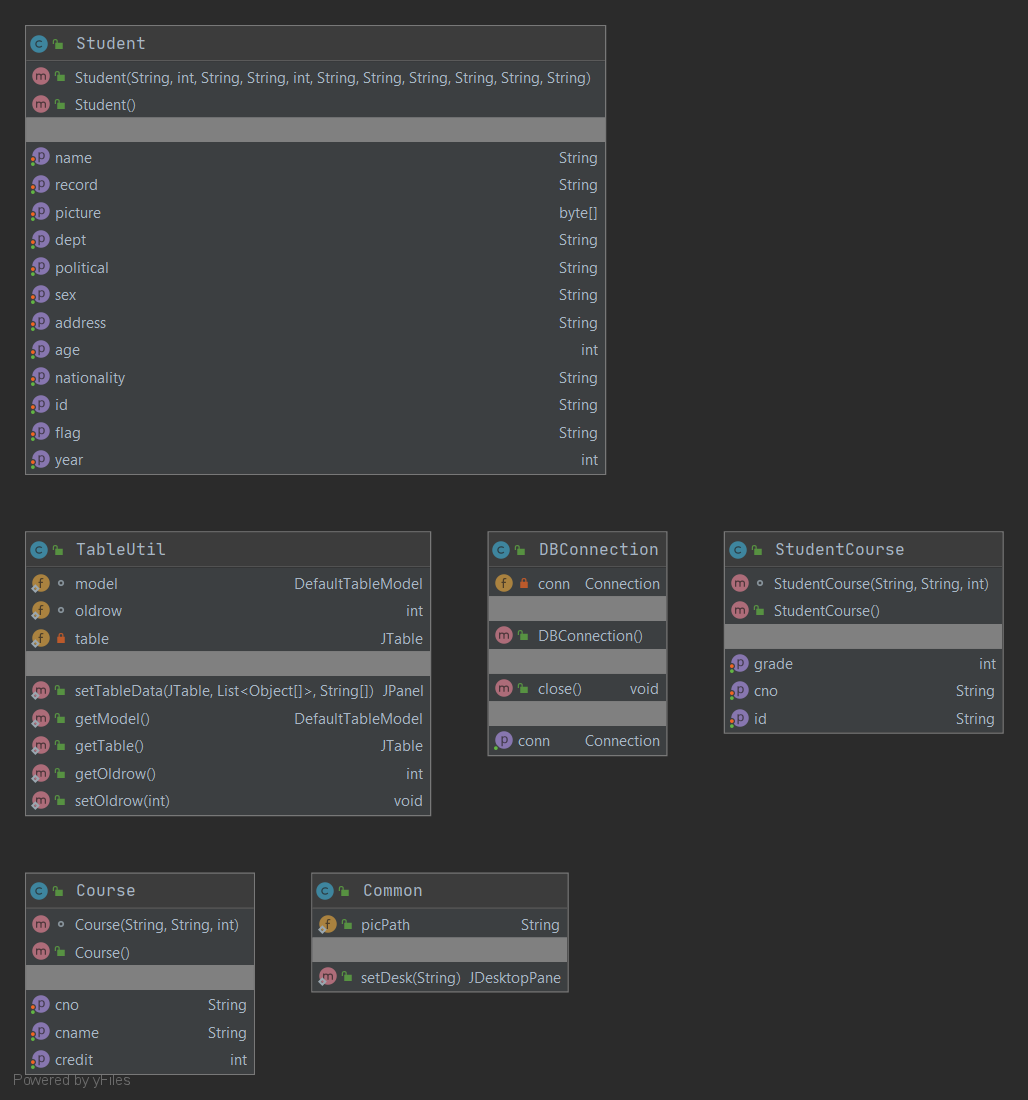
附录：程序清单及其他

项目代码数：



项目类图：





源代码：

common package：

Common.java：

package common;  
  
import java.awt.Graphics;  
import java.awt.Image;  
import java.io.File;  
import java.io.IOException;  
  
import javax.imageio.ImageIO;  
import javax.swing.JDesktopPane;  
  
public class Common {  
  
 public static String *picPath*;  
 public static JDesktopPane setDesk(String pic){  
 *picPath* = pic;  
 JDesktopPane deck = new JDesktopPane(){  
 public void paintComponent(Graphics g) {  
 Image image;  
 try {  
 image = ImageIO  
 .*read*(new File(*picPath*));  
 Image i = image.getScaledInstance(800, 500, Image.*SCALE\_SMOOTH*);  
 g.drawImage(i, 0, 0, null);  
 } catch (IOException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 }  
 }  
 };  
 return deck;  
 }  
}

Course.java：

package common;  
  
public class Course {  
 private String cno;  
 private String cname;// 姓名  
 private int credit;  
  
 Course(String cno, String cname, int credit) {  
 this.cno = cno;  
 this.cname = cname;  
 this.credit = credit;  
 }  
  
 public void setCno(String cno) {  
 this.cno = cno;  
 }  
  
 public void setCname(String cname) {  
 this.cname = cname;  
 }  
  
 public void setCredit(int credit) {  
 this.credit = credit;  
 }  
  
 public String getCno() {  
 return cno;  
 }  
  
 public String getCname() {  
 return cname;  
 }  
  
 public int getCredit() {  
 return credit;  
 }  
  
 public Course() {  
  
 }  
}

DBConnection.java：

package common;  
  
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
import java.util.Properties;  
  
public class DBConnection {  
 private Connection conn;  
   
 public void close() {  
 try {  
 conn.close();  
 } catch (SQLException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 }  
 }  
  
 public Connection getConn() {  
 if(conn!=null){  
 System.*out*.println("开启数据库连接--");  
 }  
 return conn;  
 }  
   
 public DBConnection(){  
 Properties prop = new Properties();  
 try {  
 //从外部文件读取数据库联接参数  
 prop.load(new FileInputStream("config/login.properties"));  
   
 String driver=prop.getProperty("driver");  
 String url=prop.getProperty("url");  
 String username=prop.getProperty("username");  
 String password=prop.getProperty("password");  
  
 System.*out*.println("url = " + url);  
 System.*out*.println("username = " + username);  
 System.*out*.println("password = " + password);  
   
 //加载驱动，获取数据库连接  
 Class.*forName*(driver);  
 conn=DriverManager.*getConnection*(url, username, password);  
   
 } catch (FileNotFoundException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 } catch (IOException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 } catch (ClassNotFoundException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 } catch (SQLException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 }  
 System.*out*.println("Done!");  
 }  
  
}

Student.java：

package common;  
  
public class Student {  
 private byte[] picture;  
 private String id;//学号  
 private int age;//年龄  
 private String name;//姓名  
 private String sex;//性别  
 private int year;//学制  
 private String address;//地区  
 private String dept;  
 private String political;  
 private String nationality;   
 private String record;   
 private String flag;   
 public Student(String id,int age,String name,String sex,int year,String address,String dept,  
 String political,String nationality,String record,String flag) {  
 this.id=id;  
 this.age=age;  
 this.name=name;  
 this.sex=sex;  
 this.year=year;  
 this.address=address;  
 this.dept=dept;  
 this.political=political;  
 this.nationality=nationality;  
 this.record=record;   
 this.flag=flag;  
 }  
 public void setFlag(String flag)  
 {  
 this.flag=flag;  
 }  
 public String getFlag(){  
 return flag;  
   
 }  
 public Student() {  
 // *TODO 自动生成的构造函数存根* }  
 public void setPicture(byte[] pic)  
 {  
 this.picture=pic;  
 }  
 public byte[] getPicture()  
 {  
 return picture;  
 }  
   
 public String getId() {  
 return id;  
 }  
 public void setId(String id) {  
 this.id = id;  
 }  
 public String getName() {  
 return name;  
 }  
 public void setName(String name) {  
 this.name = name;  
 }  
 public int getAge() {  
 return age;  
 }  
 public void setAge(int age) {  
 this.age = age;  
 }  
 public String getSex() {  
 return sex;  
 }  
 public void setSex(String sex) {  
 this.sex = sex;  
 }  
 public String getAddress() {  
 return address;  
 }  
 public void setAddress(String address) {  
 this.address = address;  
 }  
 public String getDept() {  
 return dept;  
 }  
 public void setDept(String dept) {  
 this.dept = dept;  
 }  
 public void setYear(int year)  
 {  
 this.year=year;  
 }  
 public int getYear()  
 {  
 return year;  
 }  
 public void setPolitical(String political)  
 {  
 this.political=political;  
 }  
 public String getPolitical()  
 {  
 return political;  
 }  
 public void setNationality(String nationality){  
 this.nationality=nationality;  
 }  
 public String getNationality()  
 {  
 return nationality;  
 }  
 public void setRecord(String record){  
 this.record=record;  
 }  
 public String getRecord()  
 {  
 return record;  
 }  
  
   
}

StudentCourse.java：

package common;  
  
public class StudentCourse {  
 private String cno;  
 private String id;  
 private int grade;  
  
 StudentCourse(String id, String cno, int grade) {  
 this.cno = cno;  
 this.id = id;  
 this.grade = grade;  
 }  
  
 public void setCno(String cno) {  
 this.cno = cno;  
 }  
  
 public void setId(String id) {  
 this.id = id;  
 }  
  
 public void setGrade(int grade) {  
 this.grade = grade;  
 }  
  
 public String getCno() {  
 return cno;  
 }  
  
 public String getId() {  
 return id;  
 }  
  
 public int getGrade() {  
 return grade;  
 }  
  
 public StudentCourse() {  
  
 }  
  
}

TableUtil.java：

package common;  
  
import java.awt.Dimension;  
import java.awt.FlowLayout;  
import java.util.List;  
  
import javax.swing.JPanel;  
import javax.swing.JScrollPane;  
import javax.swing.JTable;  
import javax.swing.table.DefaultTableModel;  
  
public class TableUtil {  
 static DefaultTableModel *model*;  
 static int *oldrow*;  
 private static JTable *table*;  
 public static JPanel setTableData(JTable detail,List<Object[]> ls,String[] title){  
 *model* = new DefaultTableModel();  
 *model*.setColumnIdentifiers(title);  
 detail = new JTable(*model*);  
 *table*=detail;  
 JScrollPane jsp = new JScrollPane(detail);  
 jsp.setPreferredSize(new Dimension(560, detail.getRowHeight()\*20));  
 JPanel tablePanel = new JPanel();  
 tablePanel.setLayout(new FlowLayout());  
 tablePanel.add(jsp);  
 for (int i = 0; i < ls.size();i++){  
 *model*.addRow(ls.get(i));  
 }  
 *oldrow* = ls.size();  
 return tablePanel;  
 }  
 public static DefaultTableModel getModel(){  
 return *model*;  
   
 }  
public static JTable getTable()  
{  
 return *table*;  
}  
 public static int getOldrow(){  
 return *oldrow*;  
 }  
 public static void setOldrow(int row){  
 *oldrow*=row;  
 }  
   
}

dao package：

Impl.java：

package dao;  
  
import java.sql.\*;  
import java.util.\*;  
  
import common.\*;  
  
  
  
public class Impl {  
 public List<Object[]> groupGrade(String cno)  
 {  
 Connection conn=new DBConnection().getConn();  
 List<Object[]> data = null;  
 String sql="select student.id,name,dept,grade from student,sc where student.id=sc.id and cno=? ;";  
 PreparedStatement pre=null;  
 ResultSet rs=null;  
 try {  
 pre=conn.prepareStatement(sql);  
 pre.setString(1,cno);  
 rs = pre.executeQuery();  
 data = new ArrayList<Object[]>();  
 while(rs.next()){  
 Object[] rowData=new Object[4];  
 rowData[0] = rs.getString("id");  
 rowData[1] = rs.getString("name");  
 rowData[2] = rs.getString("dept");  
 rowData[3] = rs.getString("grade");  
 conn.setAutoCommit(true);  
 //加入集合  
 data.add(rowData);  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return data;  
   
 }  
 public List<Object[]> sortByAge() {  
 Connection conn=new DBConnection().getConn();  
 List<Object[]> data = null;  
 String sql = "select age,*count*(\*) from Student group by age";  
 PreparedStatement pre=null;  
 ResultSet rs=null;  
 try {  
 pre=conn.prepareStatement(sql);  
 rs = pre.executeQuery(sql);  
 data = new ArrayList<Object[]>();  
 while(rs.next()){  
 Object[] rowData=new Object[2];  
 rowData[0] = rs.getString("age");  
 rowData[1] = rs.getString("count(\*)");  
 conn.setAutoCommit(true);  
 //加入集合  
 data.add(rowData);  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return data;  
 }  
 public List<Object[]> sortByRegion() {  
 Connection conn=new DBConnection().getConn();  
 List<Object[]> data = null;  
 String sql = "select address,*count*(\*) from Student group by address";  
 PreparedStatement pre=null;  
 ResultSet rs=null;  
 try {  
 pre=conn.prepareStatement(sql);  
 rs = pre.executeQuery(sql);  
 data = new ArrayList<Object[]>();  
 while(rs.next()){  
 Object[] rowData=new Object[2];  
 rowData[0] = rs.getString("address");  
 rowData[1] = rs.getString("count(\*)");  
 conn.setAutoCommit(true);  
 //加入集合  
 data.add(rowData);  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return data;  
 }  
 public boolean modify(Student s)  
 {  
   
 Connection conn=new DBConnection().getConn();  
 String sql = "update student set id=?,age=?,name=?,sex=?,year=?,address=?,dept=?,political=?,nationality=?,record=?,flag=? where id=?";  
 int flag=0;  
 PreparedStatement pre=null;  
 try {  
 pre=conn.prepareStatement(sql);  
 pre.setString(1,s.getId());  
 pre.setInt(2,s.getAge());  
 pre.setString(3,s.getName());  
 pre.setString(4,s.getSex());  
 pre.setInt(5,s.getYear());  
 pre.setString(6,s.getAddress());  
 pre.setString(7,s.getDept());  
 pre.setString(8,s.getPolitical());  
 pre.setString(9,s.getNationality());  
 pre.setString(10,s.getRecord());  
 pre.setString(11, s.getFlag());  
 pre.setString(12, s.getId());  
 flag=pre.executeUpdate();  
 if (pre!= null)pre.close();  
 if(conn!=null) conn.close();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }   
 if(flag!=0) return true;  
 else return false;  
  
  
 }  
 public boolean modify(StudentCourse sc)  
 {  
   
 Connection conn=new DBConnection().getConn();  
 int flag=0;  
 String sql = "update SC set id=?,cno=?,grade=? where id=? and cno=?";  
 PreparedStatement pre=null;  
 try {  
 pre=conn.prepareStatement(sql);  
 pre.setString(1,sc.getId());  
 pre.setString(2,sc.getCno());  
 pre.setInt(3,sc.getGrade());  
 pre.setString(4,sc.getId());  
 pre.setString(5, sc.getCno());  
 flag=pre.executeUpdate();  
 if (pre!= null)pre.close();  
 if(conn!=null) conn.close();  
   
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 if(flag!=0) return true;  
 else return false;  
  
   
 }   
   
   
   
   
 public List<Object[]> sortByPolitical() {  
 Connection conn=new DBConnection().getConn();  
 List<Object[]> data = null;  
 String sql = "select political,*count*(\*) from Student group by political";  
 PreparedStatement pre=null;  
 ResultSet rs=null;  
 try {  
 pre=conn.prepareStatement(sql);  
 rs = pre.executeQuery(sql);  
 data = new ArrayList<Object[]>();  
 while(rs.next()){  
 Object[] rowData=new Object[2];  
 rowData[0] = rs.getString("political");  
 rowData[1] = rs.getString("count(\*)");  
 conn.setAutoCommit(true);  
 //加入集合  
 data.add(rowData);  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return data;  
 }  
   
 public List<Object[]> selectStu() {  
 Connection conn=new DBConnection().getConn();  
 List<Object[]> data = null;  
 String sql = "select \* from student";  
 PreparedStatement pre=null;  
 ResultSet rs=null;  
 try {  
 pre=conn.prepareStatement(sql);  
 rs = pre.executeQuery(sql);  
 data = new ArrayList<Object[]>();  
 while(rs.next()){  
 Object[] rowData=new Object[13];  
 rowData[0] = rs.getString("id");  
 rowData[1] = rs.getInt("age");  
 rowData[2] = rs.getString("name");  
 rowData[3] = rs.getString("sex");  
 rowData[4] = rs.getInt("year");  
 rowData[5] = rs.getString("address");  
 rowData[6] = rs.getString("dept");   
 rowData[7] = rs.getString("political");   
 rowData[8] = rs.getString("nationality");   
 rowData[9] = rs.getString("record");   
 rowData[10] = rs.getString("flag");   
 conn.setAutoCommit(true);  
 //加入集合  
 data.add(rowData);  
 //if (pre!= null)pre.close();  
 //if(rs!=null) rs.close();  
 //if(conn!=null) conn.close();  
 }  
   
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return data;  
 }  
 public List<Object[]> selectClass() {  
 Connection conn=new DBConnection().getConn();  
 List<Object[]> data = null;  
 String sql = "select \* from course";  
 PreparedStatement pre=null;  
 ResultSet rs=null;  
 try {  
 pre=conn.prepareStatement(sql);  
 rs = pre.executeQuery(sql);  
 data = new ArrayList<Object[]>();  
 while(rs.next()){  
 Object[] rowData=new Object[3];  
 rowData[0] = rs.getString("cno");  
 rowData[1] = rs.getString("cname");  
 rowData[2] = rs.getInt("credit");  
   
 conn.setAutoCommit(true);  
 //加入集合  
 data.add(rowData);  
 }  
   
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return data;  
 }  
 public List<Object[]> selectGrade() {  
 Connection conn=new DBConnection().getConn();  
 List<Object[]> data = null;  
 String sql = "select \* from SC";  
 PreparedStatement pre=null;  
 ResultSet rs=null;  
 try {  
 pre=conn.prepareStatement(sql);  
 rs = pre.executeQuery(sql);  
 data = new ArrayList<Object[]>();  
 while(rs.next()){  
 Object[] rowData=new Object[3];  
 rowData[0] = rs.getString("id");  
 rowData[1] = rs.getString("cno");  
 rowData[2] = rs.getInt("grade");  
   
 conn.setAutoCommit(true);  
 //加入集合  
 data.add(rowData);  
 }  
   
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return data;  
 }  
 public boolean check(String id,String pwd,Connection conn){  
   
 PreparedStatement pre=null;  
 ResultSet rs=null;  
 String password=null;  
 try {  
 //创建sql语句  
 String sql="select pwd from tb\_admin where id=?";  
 pre=conn.prepareStatement(sql);  
 //动态传递参数并执行sql语句  
 pre.setString(1,id);  
 rs=pre.executeQuery();  
 while(rs.next())  
 password = rs.getString("pwd");  
 if(pwd.equals(password))   
 return true;  
 } catch (SQLException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 }  
 return false;  
 }  
public boolean register(String id,String pwd,Connection conn){  
   
 PreparedStatement pre=null;  
 try {  
 //创建sql语句  
 String sql="insert into tb\_admin values (?,?)";  
 pre=conn.prepareStatement(sql);  
 //动态传递参数并执行sql语句  
 pre.setString(1,id);  
 pre.setString(2, pwd);  
 pre.execute();  
 return true;  
 } catch (SQLException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 return false;  
 }  
   
 }  
 public boolean deleteStu(String id)  
 {  
 Connection conn=new DBConnection().getConn();  
 PreparedStatement pre=null;  
 String sql = "delete from student where id=?";  
 try {  
 conn.setAutoCommit(false);  
 pre = conn.prepareStatement(sql);  
 pre.setString(1,id);  
 pre.execute();  
 conn.setAutoCommit(true);  
 if (pre!= null)pre.close();  
 if(conn!=null) conn.close();  
 return true;  
 } catch (SQLException ex) {  
 // *TODO Auto-generated catch block* ex.printStackTrace();}  
 return false;  
   
 }  
  
 public boolean deleteGrade(String id,String cno)  
 {  
 Connection conn=new DBConnection().getConn();  
 PreparedStatement pre=null;  
 String sql = "delete from SC where id=? and cno=?";  
 try {  
 conn.setAutoCommit(false);  
 pre = conn.prepareStatement(sql);  
 pre.setString(1,id);  
 pre.setString(2, cno);  
 pre.execute();  
 conn.setAutoCommit(true);  
 if (pre!= null)pre.close();  
 if(conn!=null) conn.close();  
 return true;  
 } catch (SQLException ex) {  
 // *TODO Auto-generated catch block* ex.printStackTrace();}  
 return false;  
   
 }  
 public boolean deleteClass(String id)  
 {  
 Connection conn=new DBConnection().getConn();  
 PreparedStatement pre=null;  
 String sql = "delete from course where cno=?";  
 try {  
 conn.setAutoCommit(false);  
 pre = conn.prepareStatement(sql);  
 pre.setString(1,id);  
 pre.execute();  
 conn.setAutoCommit(true);  
 if (pre!= null)pre.close();  
 if(conn!=null) conn.close();  
 return true;  
 } catch (SQLException ex) {  
 // *TODO Auto-generated catch block* ex.printStackTrace();}  
 return false;  
   
 }  
   
   
   
 public boolean insertGrade(List<StudentCourse> ls){  
 Connection conn=new DBConnection().getConn();  
 PreparedStatement pre=null;  
 String sql = "insert into SC values(?,?,?)";  
 try {  
 conn.setAutoCommit(false);  
 pre = conn.prepareStatement(sql);  
 for (StudentCourse sc:ls){  
 pre=conn.prepareStatement(sql);  
 pre.setString(1,sc.getId());  
 pre.setString(2,sc.getCno());  
 pre.setInt(3,sc.getGrade());  
   
 pre.addBatch();  
   
 }  
 pre.executeBatch();  
 conn.setAutoCommit(true);  
   
 if (pre!= null)pre.close();  
 if(conn!=null) conn.close();  
 return true;  
 } catch (SQLException ex) {  
 // *TODO Auto-generated catch block* ex.printStackTrace();}  
 return false;  
 }  
   
   
   
 public boolean insertStu(List<Student> ls){  
 Connection conn=new DBConnection().getConn();  
 PreparedStatement pre=null;  
 String sql = "insert into student values(?,?,?,?,?,?,?,?,?,?,?)";  
 try {  
 conn.setAutoCommit(false);  
 pre = conn.prepareStatement(sql);  
 for (Student s:ls){  
 pre=conn.prepareStatement(sql);  
 pre.setString(1,s.getId());  
 pre.setInt(2,s.getAge());  
 pre.setString(3,s.getName());  
 pre.setString(4,s.getSex());  
 pre.setInt(5,s.getYear());  
 pre.setString(6,s.getAddress());  
 pre.setString(7,s.getDept());  
 pre.setString(8,s.getPolitical());  
 pre.setString(9,s.getNationality());  
 pre.setString(10,s.getRecord());  
 pre.setString(11, s.getFlag());  
 pre.addBatch();  
   
 }  
 pre.executeBatch();  
 conn.setAutoCommit(true);  
   
 if (pre!= null)pre.close();  
 if(conn!=null) conn.close();  
 return true;  
 } catch (SQLException ex) {  
 // *TODO Auto-generated catch block* ex.printStackTrace();}  
 return false;  
 }  
  
 public boolean insertClass(List<Course> ls){  
 Connection conn=new DBConnection().getConn();  
 PreparedStatement pre=null;  
 String sql = "insert into course values(?,?,?)";  
 try {  
 conn.setAutoCommit(false);  
 pre = conn.prepareStatement(sql);  
 for (Course c:ls){  
 pre=conn.prepareStatement(sql);  
 pre.setString(1,c.getCno());  
 pre.setString(2,c.getCname());  
 pre.setInt(3,c.getCredit());  
 pre.addBatch();  
 }  
 pre.executeBatch();  
 conn.setAutoCommit(true);  
 System.*out*.println("添加成功");  
 if (pre!= null)pre.close();  
 if(conn!=null) conn.close();  
 return true;  
 } catch (SQLException ex) {  
 // *TODO Auto-generated catch block* ex.printStackTrace();}  
 return false;  
 }  
   
 public void see(Connection conn)  
 {  
 PreparedStatement pre=null;  
 ResultSet rs=null;  
 String password=null;  
 String id=null;  
 try {  
 //创建sql语句  
 String sql="select \* from tb\_admin";  
 pre=conn.prepareStatement(sql);  
 //动态传递参数并执行sql语句  
 rs=pre.executeQuery();  
 while(rs.next()){  
 id=rs.getString("id");  
 password = rs.getString("pwd");  
 }  
   
 } catch (SQLException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 }  
   
   
 }  
}

view package：

AdminView.java：

package view;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.List;  
  
import javax.swing.JDesktopPane;  
import javax.swing.JFrame;  
import javax.swing.JInternalFrame;  
import javax.swing.JMenu;  
import javax.swing.JMenuBar;  
import javax.swing.JMenuItem;  
  
  
import javax.swing.JPanel;  
import javax.swing.JTable;  
import javax.swing.table.DefaultTableModel;  
  
import common.Common;  
import dao.Impl;  
  
  
public class AdminView extends JFrame {  
  
 JDesktopPane desk;  
 public AdminView(){  
   
 JMenuBar menubar = new JMenuBar();  
 this.setJMenuBar(menubar);  
 JMenu menu1 = new JMenu("基本信息登记修改");  
 JMenuItem item1\_1 = new JMenuItem("学生信息");  
 item1\_1.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new StudentView());  
 }  
 });  
 JMenuItem item1\_2 = new JMenuItem("课程信息");  
 item1\_2.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new CourseView());  
 }  
 });  
 JMenuItem item1\_3 = new JMenuItem("学生成绩信息");  
 item1\_3.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new StudentCourseView());  
 }  
 });  
 menu1.add(item1\_1);  
 menu1.add(item1\_2);  
 menu1.add(item1\_3);  
 menubar.add(menu1);  
 JMenu menu2 = new JMenu("基本信息更改");  
   
   
 JMenuItem item2\_1 = new JMenuItem("学生信息更改");  
   
 item2\_1.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new StudentEditDialog());  
 }  
 });  
   
   
 JMenuItem item2\_2 = new JMenuItem("学生成绩更改");  
   
 item2\_2.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new GradeEditDialog());  
 }  
 });  
   
   
   
 menu2.add(item2\_1);  
 menu2.add(item2\_2);  
 menubar.add(menu2);  
 JMenu menu3 = new JMenu("基本信息查询");  
 JMenuItem item3\_1 = new JMenuItem("按年龄统计人数");  
 item3\_1.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new SortByAgeView());  
 }  
 });  
   
   
   
 JMenuItem item3\_2 = new JMenuItem("按地区来源统计人数");  
   
 item3\_2.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new SortByRegionView());  
 }  
 });  
 JMenuItem item3\_3 = new JMenuItem("按政治面貌统计人数");  
 item3\_3.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new SortByPoliticalView());  
 }  
 });  
   
 JMenuItem item3\_4 = new JMenuItem("统计各专业某门课的学习成绩情况");  
 item3\_4.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 desk.add(new SortGradeView());  
 }  
 });  
   
 menu3.add(item3\_1);  
 menu3.add(item3\_2);  
 menu3.add(item3\_3);  
 menu3.add(item3\_4);  
 menubar.add(menu3);  
 desk = Common.*setDesk*("picture/2.jpg");  
 getContentPane().add(desk);  
 this.setTitle("管理员登录");  
 this.setSize(800, 500);  
 this.setLocationRelativeTo(null);  
 this.setResizable(true);  
 this.setVisible(true);  
 }  
   
   
}

CourseView.java：

package view;  
  
import java.awt.GridLayout;  
import java.awt.event.MouseAdapter;  
import java.awt.event.MouseEvent;  
import java.util.ArrayList;  
import java.util.List;  
  
import javax.swing.JButton;  
import javax.swing.JInternalFrame;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JTable;  
import javax.swing.table.DefaultTableModel;  
  
import common.Course;  
import common.Student;  
import common.TableUtil;  
import dao.Impl;  
  
public class CourseView extends JInternalFrame{  
 JTable detail;  
 DefaultTableModel model;  
 JPanel Jbutton;  
public CourseView(){  
 super("课程信息",true,true,false,true);  
 Impl impl = new Impl();  
 List<Object[]> ls = impl.selectClass();  
 String[] title = {"课程号","课程名","学分"};  
 JPanel tablePanel = TableUtil.*setTableData*(detail, ls,title);  
 model = TableUtil.*getModel*();  
 getContentPane().add(tablePanel);  
 setButton();  
 getContentPane().setLayout(new GridLayout(2,1));  
 this.setSize(600, 500);  
 this.setVisible(true);  
   
 }  
  
 private void setButton() {  
 JButton addButton = new JButton("添加课程");  
 addButton.addMouseListener(new MouseAdapter() {  
 public void mouseClicked(MouseEvent e){  
 Object[] rowdata = new Object[model.getColumnCount()];  
 model.addRow(rowdata);  
 }  
 });  
  
 JButton delButton = new JButton("删除课程");  
 delButton.addMouseListener(new MouseAdapter() {  
 public void mouseClicked(MouseEvent e){  
 JTable table=TableUtil.*getTable*();  
 int selectRow=table.getSelectedRow();  
 Impl impl=new Impl();  
 String id=(String)model.getValueAt(selectRow, 0);   
 if(impl.deleteClass(id)){  
 model.removeRow(selectRow);  
 model.fireTableDataChanged();  
 JOptionPane.*showMessageDialog*(null, "删除课程信息成功", "Addition",  
 JOptionPane.*PLAIN\_MESSAGE*);  
 int oldrow = TableUtil.*getOldrow*();  
 TableUtil.*setOldrow*(oldrow-1);  
 }  
 else {  
 JOptionPane.*showMessageDialog*(null, "删除课程信息失败", "Addition",  
 JOptionPane.*WARNING\_MESSAGE*);  
   
 }  
   
   
  
 }});   
 JButton confirmAddButton = new JButton("确定");  
 confirmAddButton.addMouseListener(new MouseAdapter() {  
 public void mouseClicked(MouseEvent e){  
 int oldrow = TableUtil.*getOldrow*();  
 int newrow = model.getRowCount();   
 List<Course> ls = new ArrayList<Course>();  
 for (int i = oldrow;i<newrow;i++){  
 Course c=new Course();  
 c.setCno((String)model.getValueAt(i, 0));  
 c.setCname((String)model.getValueAt(i, 1));  
 String credit=(String)model.getValueAt(i, 2);  
 c.setCredit(Integer.*parseInt*(credit));  
   
 ls.add(c);  
 Impl impl = new Impl();  
 if(impl.insertClass(ls)){  
 JOptionPane.*showMessageDialog*(null, "添加课程信息成功", "Addition",  
 JOptionPane.*PLAIN\_MESSAGE*);  
 TableUtil.*setOldrow*(oldrow+1);}  
 else {  
 JOptionPane.*showMessageDialog*(null, "添加课程信息失败", "Addition",  
 JOptionPane.*WARNING\_MESSAGE*);  
 JTable table=TableUtil.*getTable*();  
 model.removeRow(table.getRowCount()-1);  
 }  
   
 }  
   
 }  
 });  
 Jbutton = new JPanel();  
 Jbutton.add(addButton);  
 Jbutton.add(delButton);  
 Jbutton.add(confirmAddButton);  
 getContentPane().add(Jbutton);  
}  
   
  
}

GradeEditDialog.java：

package view;  
  
  
import javax.swing.\*;  
  
import java.awt.\*;  
  
import common.StudentCourse;  
import dao.Impl;  
  
  
  
public class GradeEditDialog extends JInternalFrame {  
 private JPanel panel1,panel2,panel3,panel4;  
 private JLabel jl\_sno, jl\_cno,jl\_grade;  
 private JTextField jt\_sno,jt\_cno,jt\_grade;  
 private JButton ok;  
 private JButton cancel;  
 private StudentCourse sc;  
 private Impl impl=new Impl();  
 GradeEditDialog(){  
 super("学生成绩修改",true,true,false,true);  
 jl\_sno=new JLabel("ID");  
 jt\_sno=new JTextField(15);  
   
 jl\_cno=new JLabel("课序号");  
 jt\_cno=new JTextField(15);   
 jl\_grade=new JLabel("成绩");  
 jt\_grade=new JTextField(15);   
 panel1=new JPanel(new GridLayout(3,1));  
 panel1.add(jl\_sno);   
 panel1.add(jl\_cno);  
 panel1.add(jl\_grade);  
   
  
   
 panel2=new JPanel(new GridLayout(3,1));  
 panel2.add(jt\_sno);   
 panel2.add(jt\_cno);  
 panel2.add(jt\_grade);  
  
 panel3=new JPanel(new GridLayout(1,2));  
 panel3.add(panel1);  
 panel3.add(panel2);  
   
 ok=new JButton("确定");  
 ok.addActionListener((e)->{  
 sc=new StudentCourse();  
   
   
   
 sc.setId(jt\_sno.getText().trim());   
 sc.setCno(jt\_cno.getText().trim());  
   
 sc.setGrade(Integer.*parseInt*(jt\_grade.getText().trim()));  
   
   
 if (impl.modify(sc) == false) {  
 JOptionPane.*showMessageDialog*(null, "修改失败", "Edition",  
 JOptionPane.*ERROR\_MESSAGE*);  
 jt\_sno.setText(null);  
 jt\_cno.setText(null);  
 jt\_grade.setText(null);  
   
   
   
 } else {  
 JOptionPane.*showMessageDialog*(null, "修改成功", "Edition",  
 JOptionPane.*PLAIN\_MESSAGE*);  
 this.dispose();  
 }  
 });  
 cancel=new JButton("取消");  
 cancel.addActionListener((e)->{  
 jt\_sno.setText(null);  
 jt\_cno.setText(null);  
 jt\_grade.setText(null);  
   
 });  
 panel4=new JPanel();  
 panel4.add(ok);  
 panel4.add(cancel);  
 this.setLayout(new BorderLayout());  
 this.add(panel3,BorderLayout.*CENTER*);  
 this.add(panel4,BorderLayout.*SOUTH*);  
 this.setSize(400, 300);  
 this.setVisible(true);  
 }  
}

LoginView.java：

package view;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.sql.Connection;  
import java.sql.SQLException;  
  
import javax.swing.\*;  
  
import dao.Impl;  
import common.DBConnection;  
  
public class LoginView extends JFrame{  
   
 private JPanel pasPanel;  
 private JPanel idPanel;  
 private JPanel buttonPanel;  
 private JPanel headlinePanel;  
   
 //文本框声明为成员变量的原因是在按钮单击时，需要从文本框里取值给emp  
 private JTextField idText;  
 private JPasswordField pasText;  
   
 private Font font;  
  
 public LoginView(){  
  
 this.setDisplay();  
 getContentPane().add(headlinePanel);  
 getContentPane().add(idPanel);  
 getContentPane().add(pasPanel);  
 getContentPane().add(buttonPanel);  
   
   
 this.setTitle("学籍管理系统");  
 this.setSize(500,300);  
   
 //设置网格布局 参数1代表行，参数2代表列  
 getContentPane().setLayout(new GridLayout(4, 1));  
 this.setResizable(false);//设置能否调整窗口大小，false不能调整大小  
   
 this.setLocationRelativeTo(null);//设置窗口相对于指定组件的位置，如果组件当前未显示，或者 c 为 null，则此窗口将置于屏幕的中央  
 this.setVisible(true);//当前窗口可见  
   
 }  
   
 //初期化各控件  
 public void setDisplay(){  
 this.setPanel();  
 this.setFont();  
 this.setLabel();  
 this.setButton();  
 this.setTextField();  
 }  
   
 //设置输入文本域  
 public void setTextField(){  
 // 设置文本框  
 idText = new JTextField(10);  
 idText.setFont(font);  
   
 pasText = new JPasswordField(10);  
 pasText.setFont(font);  
 pasText.setEchoChar('\*');  
   
 pasText.addKeyListener(new KeyAdapter() {  
 @Override  
 public void keyPressed(KeyEvent ee) {  
 //KeyEvent.VK\_ENTER：等价于回车键按下  
 if (ee.getKeyCode() == KeyEvent.*VK\_ENTER*) {  
 checkLogin();//自定义方法卡号和密码是否存在  
 }  
 }  
 });  
 //JPanel里添加文本框  
 idPanel.add(idText);  
 pasPanel.add(pasText);  
 }  
   
 // 设置面板  
 public void setPanel() {  
 this.idPanel=new JPanel();  
 this.idPanel.setLayout(new FlowLayout());  
 this.pasPanel=new JPanel();  
 this.pasPanel.setLayout(new FlowLayout());  
   
 this.buttonPanel=new JPanel();  
 this.buttonPanel.setLayout(new FlowLayout());  
   
 this.headlinePanel=new JPanel();  
 this.headlinePanel.setLayout(new FlowLayout());   
 }  
   
 //设置字体  
 public void setFont(){  
 font=new Font("楷体", Font.*PLAIN*, 20);  
 }  
   
 //设置标签  
 public void setLabel(){  
 //设置员工标签  
 JLabel idLabel=new JLabel("帐号：");  
 idLabel.setFont(font);  
 idPanel.add(idLabel);//标签必须要加到panel里  
   
 //设置密码标签  
 JLabel pasLabel=new JLabel("密码：");  
 pasLabel.setFont(font);  
 pasPanel.add(pasLabel);//标签必须要加到panel里  
   
 //设置显示文字;  
 JLabel headline = new JLabel("学籍管理系统");  
 headline.setFont(new Font("楷体", Font.*PLAIN*, 40));  
 headlinePanel.add(headline);  
 }  
   
 //设置按钮  
 public void setButton(){  
 JButton regisiter=new JButton("注册");  
 JButton submit=new JButton("登录");  
 regisiter.addMouseListener(new MouseAdapter()   
 {public void mouseClicked(MouseEvent e) {  
 regisiter();  
 }});  
 submit.addMouseListener(new MouseAdapter() {  
 @Override  
 public void mouseClicked(MouseEvent e) {  
 checkLogin();  
 }  
 });  
 JButton exit=new JButton("退出");  
   
 exit.addMouseListener(new MouseAdapter() {  
 @Override  
 public void mouseClicked(MouseEvent e) {  
 System.*exit*(0);  
 }  
 });  
 buttonPanel.add(regisiter);  
 buttonPanel.add(submit);  
 buttonPanel.add(exit);  
 }  
 public void regisiter(){  
 String id = String.*valueOf*(idText.getText());  
 String pas = String.*valueOf*(pasText.getPassword());  
 DBConnection db=new DBConnection();  
 Connection conn =db.getConn();  
 Impl impl = new Impl();  
 try{  
 if(impl.register(id, pas, conn)){  
   
 JOptionPane.*showMessageDialog*(null, "注册成功", "Register",  
 JOptionPane.*PLAIN\_MESSAGE*);  
 idText.setText(null);  
 pasText.setText(null);  
   
 } else {  
 JOptionPane.*showMessageDialog*(null, "用户名重复", "登陆错误",  
 JOptionPane.*ERROR\_MESSAGE*);  
 }  
  
 }catch(Exception e){  
 e.printStackTrace();  
 }  
 }  
  
  
   
 // 判断员工号和密码是否正确，并且根据不同的权限跳转相应的页面  
 public void checkLogin() {  
 // 从输入框里取得数据设置到成员变量emp上  
 String id = String.*valueOf*(idText.getText());  
 String pas = String.*valueOf*(pasText.getPassword());  
 DBConnection db=new DBConnection();  
 Connection conn =db.getConn();  
 Impl impl = new Impl();  
 try{  
 if(impl.check(id, pas, conn)){  
 this.setVisible(false);// 当前窗口不可见  
 new AdminView();  
 } else {  
 JOptionPane.*showMessageDialog*(null, "用户名不存在或密码不正确", "登陆错误",  
 JOptionPane.*ERROR\_MESSAGE*);  
 }  
  
 }catch(Exception e){  
 e.printStackTrace();  
 }  
 }  
  
}

SortByAgeView.java：

package view;  
  
import java.awt.GridLayout;  
import java.util.List;  
  
import javax.swing.JButton;  
import javax.swing.JInternalFrame;  
import javax.swing.JPanel;  
import javax.swing.JTable;  
  
import common.TableUtil;  
import dao.Impl;  
public class SortByAgeView extends JInternalFrame {  
 JTable table=new JTable();  
 public SortByAgeView(){  
 super("按年龄统计人数",true,true,false,true);  
 Impl impl = new Impl();  
 List<Object[]> ls = impl.sortByAge();  
 String[] title = {"年龄","人数"};  
 JPanel tablePanel = TableUtil.*setTableData*(table, ls,title);  
 this.getContentPane().add(tablePanel);  
 this.setVisible(true);  
 this.setSize(600, 400);  
 }  
}

SortByPoliticalView.java：

package view;  
  
  
  
import java.awt.GridLayout;  
import java.util.List;  
  
import javax.swing.JButton;  
import javax.swing.JInternalFrame;  
import javax.swing.JPanel;  
import javax.swing.JTable;  
  
import common.TableUtil;  
import dao.Impl;  
  
public class SortByPoliticalView extends JInternalFrame {  
 JTable table=new JTable();  
 public SortByPoliticalView(){  
 super("按政治面貌统计人数",true,true,false,true);  
 Impl impl = new Impl();  
 List<Object[]> ls = impl.sortByPolitical();  
 String[] title = {"政治面貌","人数"};  
 JPanel tablePanel = TableUtil.*setTableData*(table, ls,title);  
 this.getContentPane().add(tablePanel);  
 this.setVisible(true);  
 this.setSize(600, 400);  
 }  
}

SortByRegionView.java：

package view;  
import java.awt.\*;  
import java.util.ArrayList;  
import java.util.List;  
  
import javax.swing.JButton;  
import javax.swing.JInternalFrame;  
import javax.swing.JPanel;  
import javax.swing.JTable;  
import javax.swing.table.DefaultTableModel;  
  
import common.Course;  
import common.Student;  
import common.TableUtil;  
import dao.Impl;  
  
public class SortByRegionView extends JInternalFrame{  
 JTable detail;  
public SortByRegionView(){  
 super("按地区来源统计人数",true,true,false,true);  
 Impl impl = new Impl();  
 List<Object[]> ls = impl.sortByRegion();  
 String[] title = {"地区","人数"};  
 JPanel tablePanel = TableUtil.*setTableData*(detail, ls,title);  
 getContentPane().add(tablePanel);  
 this.setSize(600, 400);  
 this.setVisible(true);  
   
 }}

SortGradeView.java：

package view;  
  
import java.awt.GridLayout;  
import java.util.ArrayList;  
import java.util.HashSet;  
import java.util.Iterator;  
import java.util.List;  
import javax.swing.\*;  
  
import common.TableUtil;  
import dao.Impl;  
class Major{  
   
 String major=new String();  
 int[] level=new int[5];  
}  
public class SortGradeView extends JInternalFrame {  
 private JTable detail\_1;  
 private JTable detail\_2;  
 public SortGradeView(){  
 super("统计成绩",true,true,false,true);  
 String cno = JOptionPane.*showInputDialog*("输入课序号");  
 Impl impl = new Impl();  
 List<Object[]> ls = impl.groupGrade(cno);  
 String[] title = {"学号","姓名","系","成绩"};  
 JPanel tablePanel = TableUtil.*setTableData*(detail\_1, ls,title);  
 HashSet<String> hs=new HashSet<String>();  
 for(Object[] o:ls)  
 {  
 if(!hs.contains((String)o[2]))  
 hs.add((String)o[2]);  
 }  
 Major[] m=new Major[hs.size()];  
  
   
 int i=0;  
Iterator<String> it=hs.iterator();  
while(it.hasNext())  
{  
 m[i]=new Major();  
 String s=it.next();  
 m[i].major=s;  
 for(int j=0;j<5;j++)  
 m[i].level[j]=0;  
 i++;   
}  
  
 for(Object[] o:ls)  
 {  
 String major=(String)o[2];  
 for(Major mj:m){  
 if(major.equals(mj.major)){  
 int grade=Integer.*valueOf*((String)o[3]);  
   
 if(grade>=90&&grade<=100)  
 mj.level[0] +=1;  
 else if(grade>=80&&grade<90)  
 mj.level[1]+=1;  
 else if((grade>=70&&grade<80))  
 mj.level[2]+=1;  
 else if(grade>=60&&grade<70)  
 mj.level[3]+=1;  
 else if(grade<60)  
 mj.level[4]+=1;   
 }  
 }  
 }  
  
 List<Object[]> li=new ArrayList<Object[]>();  
   
 for(Major mj:m)  
 {  
 String[] nums=new String[6];  
 nums[0]=mj.major;  
 for(int j=1;j<6;j++)  
 nums[j]=String.*valueOf*(mj.level[j-1]);  
 li.add(nums);  
 }  
 String[] name = {" ","90-100","80-90","70-80","60-70","<60"};  
 JPanel Panel = TableUtil.*setTableData*(detail\_2, li,name);  
 JPanel mainPanel=new JPanel(new GridLayout(2,1));  
 mainPanel.add(tablePanel);  
 mainPanel.add(Panel);  
 this.getContentPane().add(mainPanel);  
 this.setVisible(true);  
 this.setSize(600, 400);  
   
   
   
   
 }  
   
}

StudentCourseView.java：

package view;  
  
  
import java.awt.GridLayout;  
import java.awt.event.MouseAdapter;  
import java.awt.event.MouseEvent;  
import java.util.ArrayList;  
import java.util.List;  
  
import javax.swing.JButton;  
import javax.swing.JInternalFrame;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JTable;  
import javax.swing.table.DefaultTableModel;  
  
import common.Course;  
  
import common.StudentCourse;  
import common.TableUtil;  
import dao.Impl;  
  
public class StudentCourseView extends JInternalFrame{  
 JTable detail;  
 DefaultTableModel model;  
 JPanel Jbutton;  
public StudentCourseView(){  
 super("学生选课信息",true,true,false,true);  
 Impl impl = new Impl();  
 List<Object[]> ls = impl.selectGrade();  
 String[] title = {"学号","课程号","成绩"};  
 JPanel tablePanel = TableUtil.*setTableData*(detail, ls,title);  
 model = TableUtil.*getModel*();  
 getContentPane().add(tablePanel);  
 setButton();  
 getContentPane().setLayout(new GridLayout(2,1));  
 this.setSize(600, 500);  
 this.setVisible(true);  
   
 }  
  
 private void setButton() {  
 JButton addButton = new JButton("添加学生成绩");  
 addButton.addMouseListener(new MouseAdapter() {  
 public void mouseClicked(MouseEvent e){  
 Object[] rowdata = new Object[model.getColumnCount()];  
 model.addRow(rowdata);  
 }  
 });  
  
 JButton delButton = new JButton("删除学生成绩");  
 delButton.addMouseListener(new MouseAdapter() {  
 public void mouseClicked(MouseEvent e){  
 JTable table=TableUtil.*getTable*();  
 int selectRow=table.getSelectedRow();  
 Impl impl=new Impl();  
 String id=(String)model.getValueAt(selectRow, 0);  
 String cno=(String)model.getValueAt(selectRow, 1);  
   
 if(impl.deleteGrade(id,cno)){  
 model.removeRow(selectRow);  
 model.fireTableDataChanged();  
 JOptionPane.*showMessageDialog*(null, "删除学生信息成功", "Addition",  
 JOptionPane.*PLAIN\_MESSAGE*);  
 int oldrow = TableUtil.*getOldrow*();  
 TableUtil.*setOldrow*(oldrow-1);  
 }  
 else {  
 JOptionPane.*showMessageDialog*(null, "删除学生信息失败", "Addition",  
 JOptionPane.*WARNING\_MESSAGE*);  
   
 }  
 }});   
   
   
 JButton confirmAddButton = new JButton("确定");  
 confirmAddButton.addMouseListener(new MouseAdapter() {  
 public void mouseClicked(MouseEvent e){  
 int oldrow = TableUtil.*getOldrow*();  
 int newrow = model.getRowCount();   
 List<StudentCourse> ls = new ArrayList<StudentCourse>();  
 for (int i = oldrow;i<newrow;i++){  
 StudentCourse sc=new StudentCourse();  
 sc.setId((String)model.getValueAt(i, 0));  
 sc.setCno((String)model.getValueAt(i, 1));  
 String grade=(String)model.getValueAt(i, 2);  
 sc.setGrade(Integer.*parseInt*(grade));  
 ls.add(sc);  
 }  
 Impl impl = new Impl();  
   
 if(impl.insertGrade(ls))  
 {  
 JOptionPane.*showMessageDialog*(null, "添加成绩信息成功", "Addition",  
 JOptionPane.*PLAIN\_MESSAGE*);  
 TableUtil.*setOldrow*(oldrow+1);}  
 else {  
 JOptionPane.*showMessageDialog*(null, "添加成绩信息失败", "Addition",  
 JOptionPane.*WARNING\_MESSAGE*);  
 JTable table=TableUtil.*getTable*();  
 model.removeRow(table.getRowCount()-1);  
 }  
   
 }  
 });  
 Jbutton = new JPanel();  
 Jbutton.add(addButton);  
 Jbutton.add(delButton);  
 Jbutton.add(confirmAddButton);  
 getContentPane().add(Jbutton);  
}  
   
  
}

StudentEditDialog.java：

package view;  
  
import javax.swing.\*;  
  
import java.awt.\*;  
  
import javax.swing.\*;  
  
import common.Student;  
import dao.Impl;  
  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
public class StudentEditDialog extends JInternalFrame {  
   
   
 private JPanel panel1,panel2,panel3,panel4,panel5;  
 private JLabel jl\_sno, jl\_sname,jl\_sex, jl\_dept,jl\_record,jl\_flag,jl\_nationality,jl\_age,jl\_year,jl\_political,jl\_address;  
 private JTextField jt\_sno,jt\_sname,jt\_record,jt\_flag,jt\_sex, jt\_dept,jt\_age,jt\_nationality,jt\_year,jt\_political,jt\_address;  
 private JButton ok;  
 private JButton cancel;  
 private Student stu;  
 private Impl impl=new Impl();  
 StudentEditDialog(){  
 super("学生信息修改",true,true,false,true);  
 jl\_sno=new JLabel("ID");  
 jt\_sno=new JTextField(15);  
 jl\_sname=new JLabel("姓 名");  
 jt\_sname=new JTextField(15);  
 jl\_sex=new JLabel("性别");  
 jt\_sex=new JTextField(15);  
 jl\_dept=new JLabel("系");  
 jt\_dept=new JTextField(15);  
 jl\_age=new JLabel("年龄");  
 jt\_age=new JTextField(15);  
 jl\_political=new JLabel("政治面貌");  
 jt\_political=new JTextField(15);  
 jl\_address=new JLabel("地区");  
 jt\_address=new JTextField(15);   
 jl\_year=new JLabel("学制");  
 jt\_year=new JTextField(15);   
 jl\_nationality=new JLabel("民族");  
 jt\_nationality=new JTextField(15);   
 jl\_record=new JLabel("奖惩记录");  
 jt\_record=new JTextField(15);   
 jl\_flag=new JLabel("休复转退标记");  
 jt\_flag=new JTextField(15);   
 panel1=new JPanel(new GridLayout(11,1));  
 panel1.add(jl\_sno);   
 panel1.add(jl\_age);  
 panel1.add(jl\_sname);  
 panel1.add(jl\_sex);  
 panel1.add(jl\_year);  
 panel1.add(jl\_address);  
 panel1.add(jl\_dept);  
 panel1.add(jl\_political);  
 panel1.add(jl\_nationality);  
 panel1.add(jl\_record);  
 panel1.add(jl\_flag);  
  
   
 panel2=new JPanel(new GridLayout(11,1));  
 panel2.add(jt\_sno);   
 panel2.add(jt\_age);  
 panel2.add(jt\_sname);  
 panel2.add(jt\_sex);  
 panel2.add(jt\_year);  
 panel2.add(jt\_address);  
 panel2.add(jt\_dept);  
 panel2.add(jt\_political);  
 panel2.add(jt\_nationality);  
 panel2.add(jt\_record);  
 panel2.add(jt\_flag);  
  
 panel3=new JPanel(new GridLayout(1,2));  
 panel3.add(panel1);  
 panel3.add(panel2);  
   
 ok=new JButton("确定");  
 ok.addActionListener((e)->{  
 stu=new Student();  
   
   
   
 stu.setId(jt\_sno.getText().trim());   
 stu.setAge(Integer.*parseInt*(jt\_age.getText().trim()));  
 stu.setName( jt\_sname.getText().trim());  
 stu.setSex(jt\_sex.getText().trim());  
 stu.setYear(Integer.*parseInt*(jt\_year.getText().trim()));  
 stu.setAddress(jt\_address.getText().trim());  
 stu.setDept(jt\_dept.getText().trim());  
 stu.setPolitical(jt\_political.getText().trim());  
 stu.setNationality(jt\_nationality.getText().trim());  
 stu.setRecord(jt\_record.getText().trim());  
 stu.setFlag(jt\_flag.getText().trim());  
   
   
   
 if (impl.modify(stu) == false) {  
 JOptionPane.*showMessageDialog*(null, "修改失败", "Edition",  
 JOptionPane.*ERROR\_MESSAGE*);  
 jt\_sno.setText(null);  
 jt\_age.setText(null);  
 jt\_sname.setText(null);  
 jt\_sex.setText(null);   
 jt\_year.setText(null);  
 jt\_address.setText(null);  
 jt\_dept.setText(null);  
 jt\_political.setText(null);  
 jt\_nationality.setText(null);  
 jt\_record.setText(null);  
 jt\_flag.setText(null);  
   
   
 } else {  
 JOptionPane.*showMessageDialog*(null, "修改成功", "Edition",  
 JOptionPane.*PLAIN\_MESSAGE*);  
 this.dispose();  
 }  
 });  
 cancel=new JButton("取消");  
 cancel.addActionListener((e)->{  
 jt\_sno.setText(null);  
 jt\_sname.setText(null);  
 jt\_sex.setText(null);  
 jt\_age.setText(null);  
 jt\_dept.setText(null);   
 });  
 panel4=new JPanel();  
 panel4.add(ok);  
 panel4.add(cancel);  
 this.setLayout(new BorderLayout());  
 this.add(panel3,BorderLayout.*CENTER*);  
 this.add(panel4,BorderLayout.*SOUTH*);  
  
 this.setSize(600, 400);  
 this.setVisible(true);  
 }  
}

StudentView.java：

package view;  
  
import java.awt.GridLayout;  
import java.awt.event.MouseAdapter;  
import java.awt.event.MouseEvent;  
import java.sql.Connection;  
import java.util.ArrayList;  
import java.util.List;  
  
import javax.swing.\*;  
import javax.swing.table.DefaultTableModel;  
  
import common.\*;  
import dao.Impl;  
  
  
public class StudentView extends JInternalFrame {  
 JTable detail;  
 DefaultTableModel model;  
 JPanel Jbutton;  
public StudentView(){  
 super("学生信息",true,true,false,true);  
 Impl impl = new Impl();  
 List<Object[]> ls = impl.selectStu();  
 String[] title = {"ID","年龄","姓名","性别","学制","地址","所在系","政治面貌","民族","奖惩标记","休复转退标记"};  
 JPanel tablePanel = TableUtil.*setTableData*(detail, ls,title);  
 model = TableUtil.*getModel*();  
 getContentPane().add(tablePanel);  
 setButton();  
 getContentPane().setLayout(new GridLayout(2,1));  
 this.setSize(600, 500);  
 this.setVisible(true);  
   
 }  
  
 private void setButton() {  
 JButton addButton = new JButton("添加学生");  
 addButton.addMouseListener(new MouseAdapter() {  
 public void mouseClicked(MouseEvent e){  
 Object[] rowdata = new Object[model.getColumnCount()];  
 model.addRow(rowdata);  
 }  
 });  
  
 JButton delButton = new JButton("删除学生");  
 delButton.addMouseListener(new MouseAdapter() {  
 public void mouseClicked(MouseEvent e){  
 JTable table=TableUtil.*getTable*();  
 int selectRow=table.getSelectedRow();  
 Impl impl=new Impl();  
 String id=(String)model.getValueAt(selectRow, 0);  
   
   
   
   
 if(impl.deleteStu(id)){  
 model.removeRow(selectRow);  
 model.fireTableDataChanged();  
 JOptionPane.*showMessageDialog*(null, "删除学生信息成功", "Addition",  
 JOptionPane.*PLAIN\_MESSAGE*);  
 int oldrow = TableUtil.*getOldrow*();  
 TableUtil.*setOldrow*(oldrow-1);  
 }  
 else {  
 JOptionPane.*showMessageDialog*(null, "删除学生信息失败", "Addition",  
 JOptionPane.*WARNING\_MESSAGE*);  
   
 }  
   
   
   
 }});   
 JButton confirmAddButton = new JButton("确定");  
 confirmAddButton.addMouseListener(new MouseAdapter() {  
 public void mouseClicked(MouseEvent e){  
 int oldrow = TableUtil.*getOldrow*();  
 int newrow = model.getRowCount();   
 Student stu = null;  
 List<Student> ls = new ArrayList<Student>();  
 for (int i = oldrow;i<newrow;i++){  
 System.*out*.println((String)model.getValueAt(i, 0));  
 stu = new Student();  
 stu.setId((String)model.getValueAt(i, 0));  
 String age=(String)model.getValueAt(i, 1);  
 stu.setAge(Integer.*parseInt*(age));  
 stu.setName((String)model.getValueAt(i, 2));  
 stu.setSex((String)model.getValueAt(i, 3));   
 String year=(String)model.getValueAt(i, 4);  
 stu.setYear(Integer.*parseInt*(year));  
 stu.setAddress((String)model.getValueAt(i, 5));  
 stu.setDept((String)model.getValueAt(i, 6));  
 stu.setPolitical((String)model.getValueAt(i, 7));  
 stu.setNationality((String)model.getValueAt(i, 8));  
 stu.setRecord((String)model.getValueAt(i, 9));  
 stu.setFlag((String)model.getValueAt(i, 10));  
 ls.add(stu);  
   
 }  
 Impl impl = new Impl();  
 if(impl.insertStu(ls)){  
 JOptionPane.*showMessageDialog*(null, "添加学生信息成功", "Addition",  
 JOptionPane.*PLAIN\_MESSAGE*);  
 TableUtil.*setOldrow*(oldrow+1);}  
 else {  
 JOptionPane.*showMessageDialog*(null, "添加学生信息失败", "Addition",  
 JOptionPane.*WARNING\_MESSAGE*);  
 JTable table=TableUtil.*getTable*();  
 model.removeRow(table.getRowCount()-1);  
 }  
   
 }  
 });  
 Jbutton = new JPanel();  
 Jbutton.add(addButton);  
 Jbutton.add(delButton);  
 Jbutton.add(confirmAddButton);  
 getContentPane().add(Jbutton);  
}  
   
  
}

测试主文件：

test.java：

package view;  
import common.DBConnection;  
  
public class test {  
public static void main(String[] args){  
 new LoginView();  
 //DBConnection x = new DBConnection();  
  
 }  
}