



**Java考核**

课程名称 Java程序设计

题目名称 学生信息管理系统

学生学院 计算机学院

专业班级 计算机17(7)班

学 号 3217004729

学生姓名 丁哲芙

指导教师 赵锐

2018 年 11 月 27 日

|  |  |
| --- | --- |
| 难度系数 |  |
| 独立完成工作量占总工作量的比例 | 90% |
| 程序功能完成情况 |  |
| 报告内容与格式 |  |
| 总评成绩 |  |

目录

1. [课程题目](#_一、课程题目_1)
2. [题目分析与设计](#_二、题目分析与设计_1)

[1、开发环境](#二、1)

[2、题目需求分析](#二、2)

[3、界面设计分析](#二、3)

[4、数据库设计分析](#二、4)

[5、程序实现分析](#二、4)

1. [测试分析](#三)

[1、扩展包阐述](#三、1)

[2、测试预期成果](#三、2)

[3、测试情况](#三、3)

1. [设计体会](#四)

## 一、课程题目

**(2) 光盘借阅管理**

设计以下几个类：**光盘类：用户类：借阅记录类**，包括下列模块：

1) 新建、修改、删除光盘信息；

2) 新建、修改、删除用户信息；

3) 光盘借阅/归还信息记录；

4) 查询光盘名称，用户的借阅历史；

要求：基于JFrame窗体设计、使用菜单和数据库。

补充：类似的题目大家还可以选择学生信息管理系统、超市收银系统、仓库管理系统、医院挂号系统、ATM系统等。这些系统至少有权限管理，如用户、管理员权限。信息的增删改查功能；报表打印功能等。

## 二、题目分析与设计

1、开发环境

系统环境：Windows

开发工具：IntelliJ IDEA

Java版本：JDK 1.8

数据库：MySQL 5.7

系统采用技术：Swing+JDBC+面向对象编程

2、题目需求分析

(1)该系统的用户分为管理员、教师和学生。

管理员的权限包括：对学生、班级、教师的增删查改；

教师的权限包括：对学生、班级的增删查改，修改自己的个人信息但不包括姓名；

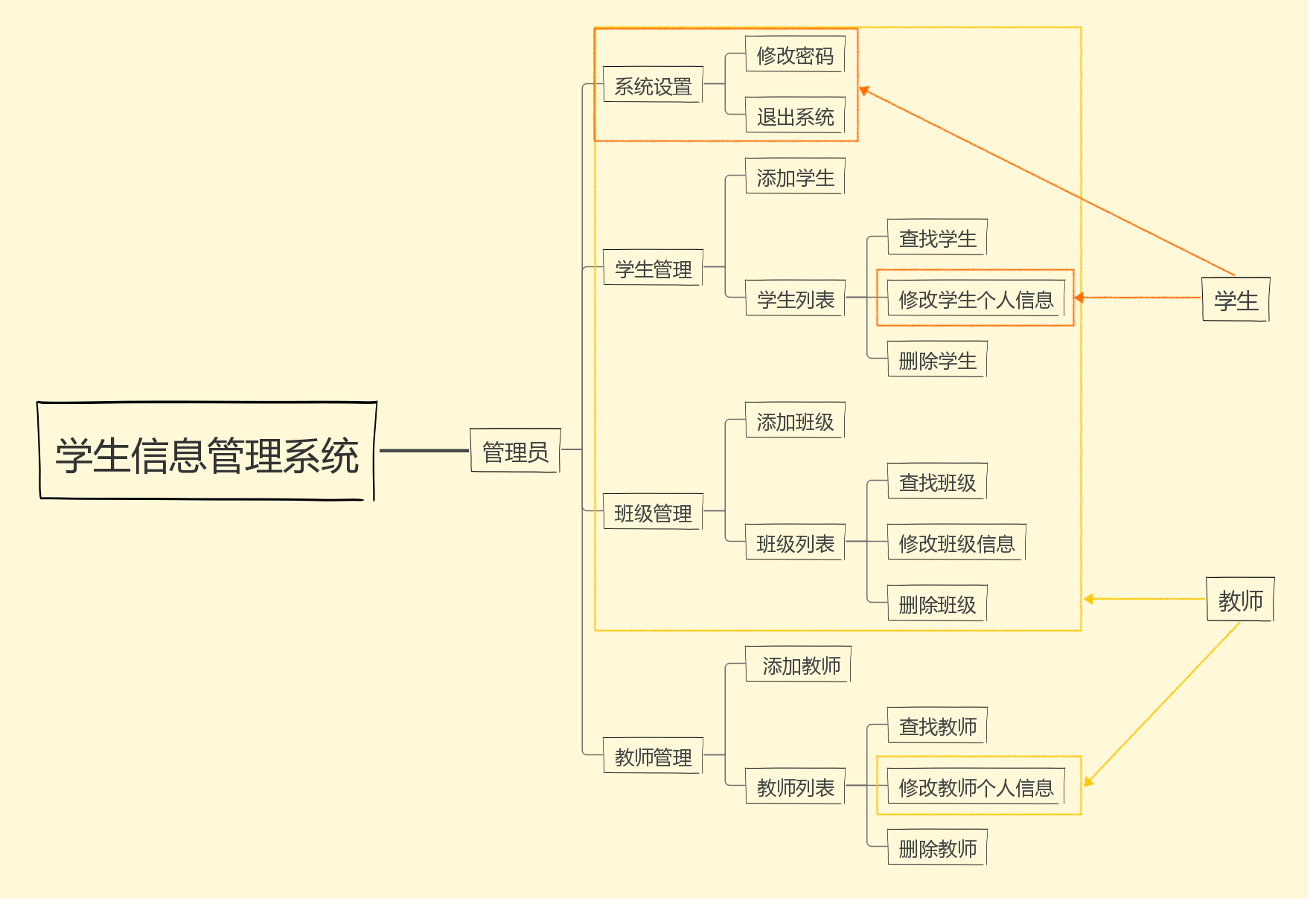
学生的权限包括：修改自己的个人信息但不包括姓名、班级；

所有用户均可在此系统上修改自己的登录密码；

（2）功能需求模块化分析

|  |  |
| --- | --- |
| 登录模块 | 用于管理员、教师、学生的登录 |
| 系统设置模块 | 用于修改当前登录用户的密码，以及退出系统 |
| 学生管理模块 | 在此模块中，学生可修改除姓名、班级外的的个人信息，教师和管理员可以增删查改学生信息 |
| 班级管理模块 | 在此模块中，学生不具备权限，教师和管理员可以增删查改班级信息 |
| 教师管理模块 | 在此模块中，学生不具备权限，教师可以修改除姓名外的个人信息，管理员可以增删查改教师信息 |
| 帮助 | 查看更多 |

2、软件功能架构图



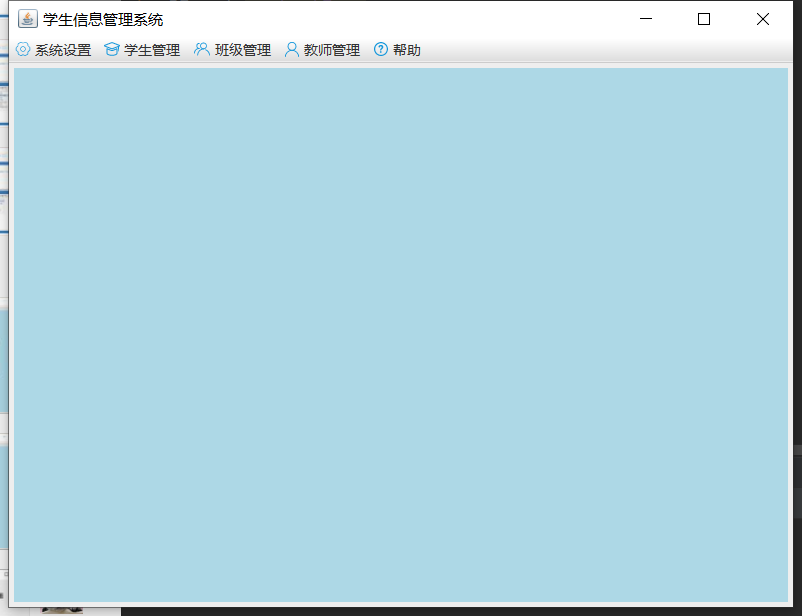
3、界面设计分析

（1）登录界面

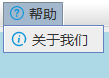
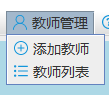
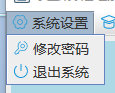


（2）主界面

主界面共分为系统设置、学生管理、班级管理、教师管理及帮助五个模块

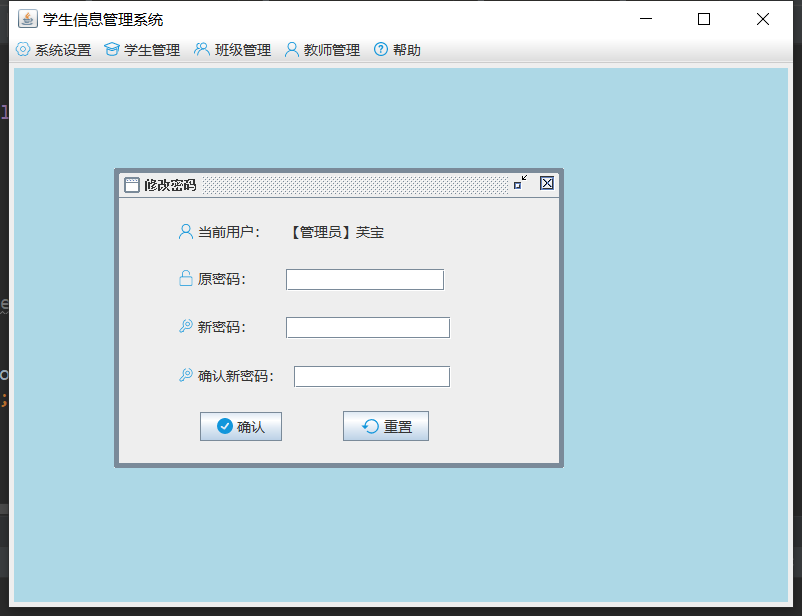


每个模块依次点击后会出现下拉菜单

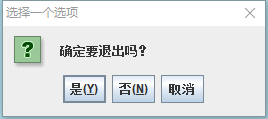


（3）系统设置

3.1系统设置->修改密码



3.2系统设置->退出系统

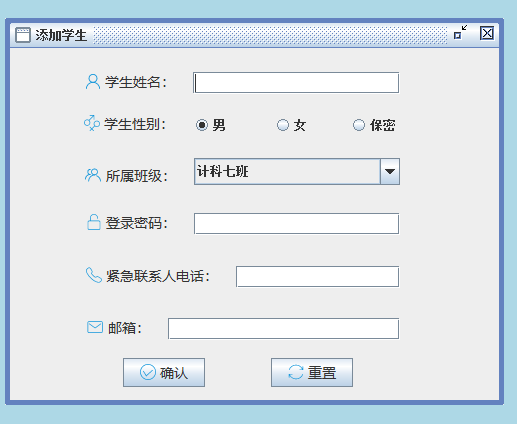


（4）学生管理

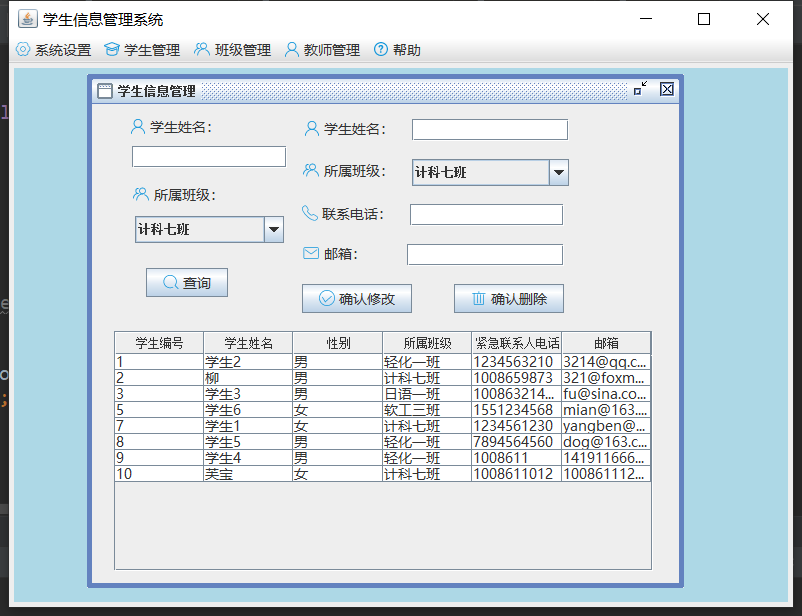
4.1学生管理->添加学生

此表单中有单选按钮（性别选择）、下拉表单（班级选择）、及多个文本框。

点击重置可将所有已填、已选的内容变为空或默认初始值。

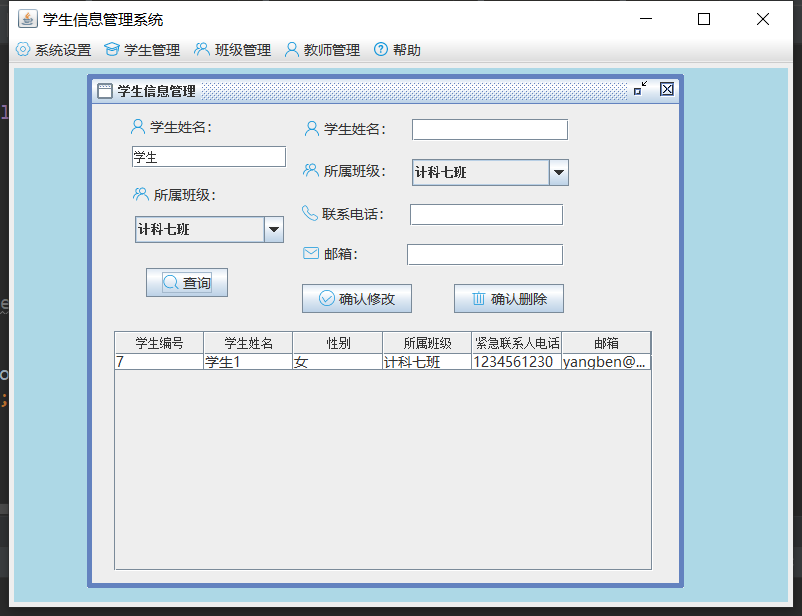


4.2学生管理->学生列表



下半部分为学生列表显示。

上半部分的左边为搜索栏，可填写学生姓名查询，支持姓名模糊查询，必须知道搜索结果的班级，即班级选对了、姓名模糊才能搜索出相应结果。

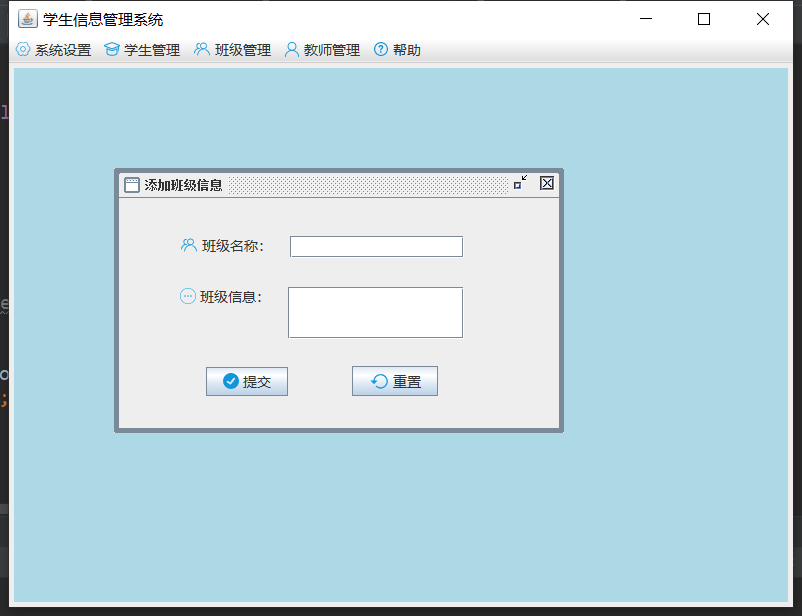


上半部分的右边为修改栏，需选中学生列表的一人，选中后该学生信息在修改栏依次呈现，此时可以修改。也可以选中学生后点击删除学生。

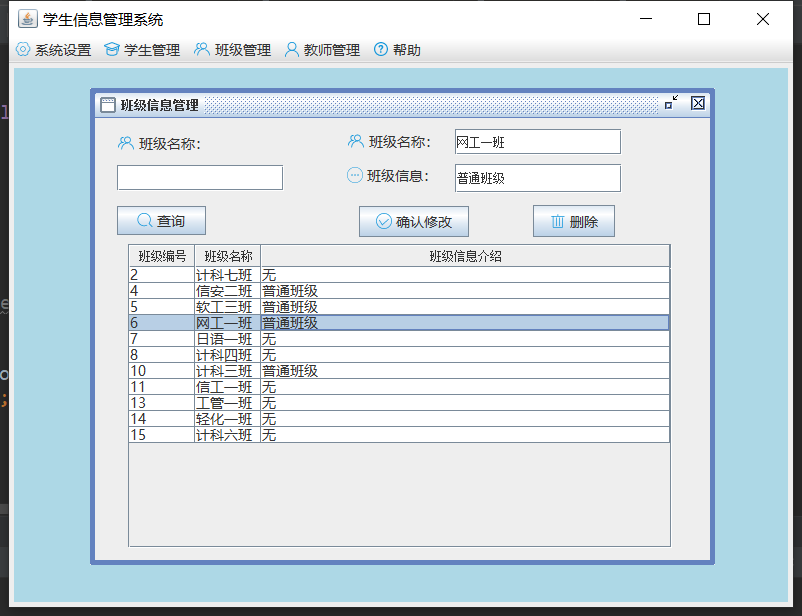


（5）班级管理

5.1班级管理->班级添加



5.2班级管理->班级列表

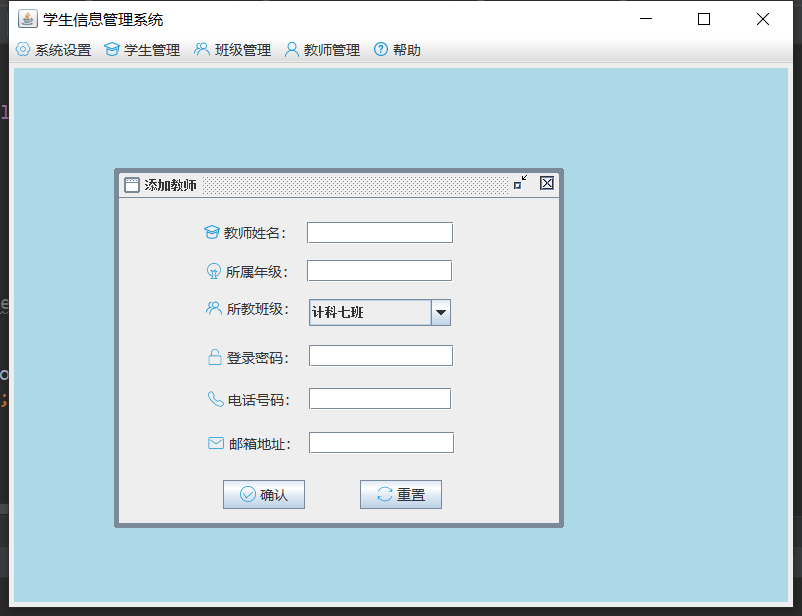


与学生列表相似，班级列表的上半部分的左边为搜索栏，可支持模糊查询。

上半部分的右边为修改栏，需选中一个班级，被选中的班级的信息会依次呈现在TextFiled中。选中后可进行修改和删除操作。

（6）教师管理

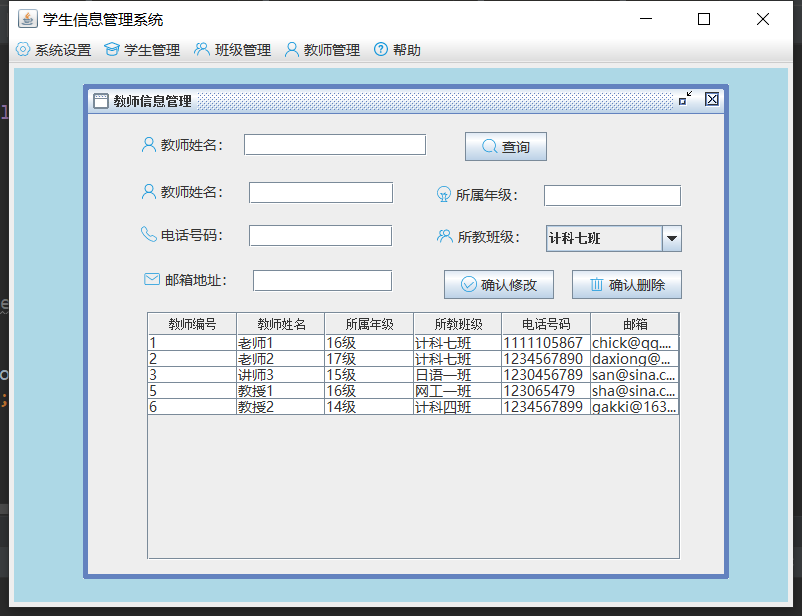
6.1教师管理->添加教师



6.2教师管理->教师列表

上面第一栏是用来搜索教师，输入教师姓名即可模糊搜索。

下面的当选中教师后，教师的个人信息会依次呈现，可供修改和删除。



（7）帮助->关于我们

点击后会有弹窗链接到广东工业大学教务系统。



4、数据库设计描述

（1）管理员表(s\_admin)

|  |  |
| --- | --- |
| **Field** | **Type** |
| id | int(5) |
| name | varchar(32) |
| password | varchar(32) |
| createDate | int(5) |

(2)班级表(s\_class)

|  |  |
| --- | --- |
| **Field** | **Type** |
| id | int(5) |
| name | varchar(32) |
| info | varchar(32) |

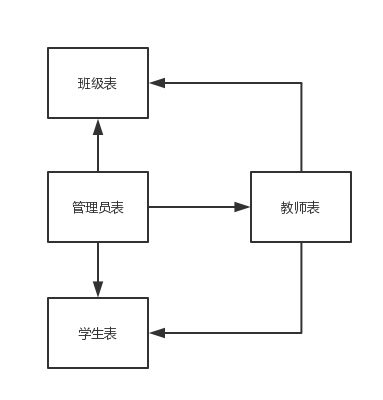
(3)学生表（s\_class）

|  |  |
| --- | --- |
| **Field** | **Type** |
| id | int(5) |
| name | varchar(32) |
| sex | varchar(8) |
| classId | int(5) |
| password | varchar(32) |
| cellphoneNumber | varchar(32) |
| mail | varchar(32) |

(4)教师表（s\_teacher）

|  |  |
| --- | --- |
| **Field** | **Type** |
| id | int(5) |
| name | varchar(32) |
| grade | varchar(32) |
| classId | int(5) |
| password | varchar(32) |
| phoneNumber | varchar(32) |
| mail | varchar(32) |

E-R图



首先，在util包里面创建DbUtil.java，目的是利用这个对象获取数据库连接。

然后，在dao包里面创建BaseDao.java，目的是创建数据库连接对象，整个项目与数据库打交道都用这个对象。

5、论述程序实现

（1）数据库连接及操作类的编写

<1>在util包中的DbUtil.java类中创建数据库连接操作。在此类中首先定义数据库连接驱动包名、数据库连接路径、数据库连接用户名和密码等静态变量，然后在构造方法中实现数据库连接操作。关键代码如下：

public class DbUtil {

private String dbUrl = "jdbc:mysql://localhost:3306/db\_student?useUnicode=true&characterEncoding=utf8&useSSL=false";

private String dbUsersName = "root";

private String dbPassword = "root";

private String jdbcName = "com.mysql.jdbc.Driver";

/\*\*

\* 获取数据库连接

\*/

public Connection getCon() {

try {

Class.forName(jdbcName);

}catch (ClassNotFoundException e) {

e.printStackTrace();

}

Connection con = null;

try {

con = DriverManager.getConnection(dbUrl, dbUsersName,dbPassword);

}catch (SQLException e) {

e.printStackTrace();

}

return con;

}

public void closeCon(Connection con) throws Exception{

if(con != null) {

con.close();

}

}

<2>为了避免运行程序时资源浪费，优化项目的运行速度，需要再完成数据库操作后关闭数据库连接。在dao包的BaseDao.java中，创建了BaseDao对象，整个项目连接都用这个对象，并定义了构造方法关闭数据库连接。关键代码如下：

public class BaseDao {

public Connection con = new DbUtil().getCon();

public void closeDao() {

try {

con.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

<3>因为我设计这个学生信息管理系统涉及到多个数据库的表，所以在dao包中创建了AdminDao.java,ClassDao.java,StudentDao.java,TeacherDao.java，其中，AdminDao、ClassDao、StudentDao、TeacherDao都继承了BaseDao，也就是说，这些类可以通过BaseDao连接数据库和关闭数据库。以AdminDao中的查询为例，用户登录时执行查询语句的方法executeQuery()，并返回结果集。关键代码如下：

public class AdminDao extends BaseDao {

public Admin login(Admin admin) {

String sql = "select \* from s\_admin where name=? and password=?";

Admin adminRst = null;

try {

PreparedStatement prst = con.prepareStatement(sql);

prst.setString(1, admin.getName());

prst.setString(2, admin.getPassword());

ResultSet executeQuery = prst.executeQuery();

if(executeQuery.next()) {

adminRst = new Admin();

adminRst.setId(executeQuery.getInt("id"));

adminRst.setName(executeQuery.getString("name"));

adminRst.setPassword(executeQuery.getString("password"));

adminRst.setCreateDate(executeQuery.getString("createDate"));

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return adminRst;

}

}

（2）登录模块设计

登录模块是学生信息管理系统的入口，可在该窗体中，输入正确的用户名称和密码、符合相应的身份的话（管理员、教师、学生），就可以登录成功进去系统。如果没有输入用户名或密码或者输入错误都会弹出相应的提示信息。登录模块窗体继承了JFrame类。在设计登录窗体前，初始化JPanel组件，设置JPanel布局。关键代码如下：

public class LoginFrm extends JFrame {

private JPanel contentPane;

private JTextField usersNameTextField;

private JPasswordField passwordTextField;

private JComboBox usersTypeComboBox;

}

窗体设计完成后，验证登录操作需要为“登录”按钮添加监听事件。将按钮监听事件写入内部类中，实现了ActionListener接口，在内部类中重写actionPerformed()方法，实现登录验证操作，获取了文本栏里的用户名称和密码的字符串，再调用dao包继而连接数据库查询数据库中的用户名和密码，一一对应的话，则可成功登录。关键代码如下：

JButton loginButton = new JButton("\u767B\u5F55");//“登录”的unicode编码

loginButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

loginAct(ae);

}

});

protected void loginAct(ActionEvent ae) {

// TODO Auto-generated method stub

String usersName = usersNameTextField.getText().toString();

String password = passwordTextField.getText().toString();

UsersType selectedItem = (UsersType)usersTypeComboBox.getSelectedItem();

if(StringUtil.isEmpty(usersName)) {

JOptionPane.showMessageDialog(this,"用户名不能为空！");

return;

}

if(StringUtil.isEmpty(password)) {

JOptionPane.showMessageDialog(this,"密码不能为空！");

return;

}

if("管理员".equals(selectedItem.getName())) {

//管理员登录

AdminDao adminDao = new AdminDao();

Admin adminTmp = new Admin();

adminTmp.setName(usersName);

adminTmp.setPassword(password);

admin = adminDao.login(adminTmp);

adminDao.closeDao();

if(admin == null) {

JOptionPane.showMessageDialog(this,"用户名或密码或身份错误！");

return;

}

this.dispose();

JOptionPane.showMessageDialog(this, "欢迎 【 "+selectedItem.getName()+" 】 : "+admin.getName()+"登录本系统！");

new MainFrm(selectedItem, admin).setVisible(true);

}else if("教师".equals(selectedItem.getName())) {

//教师登录

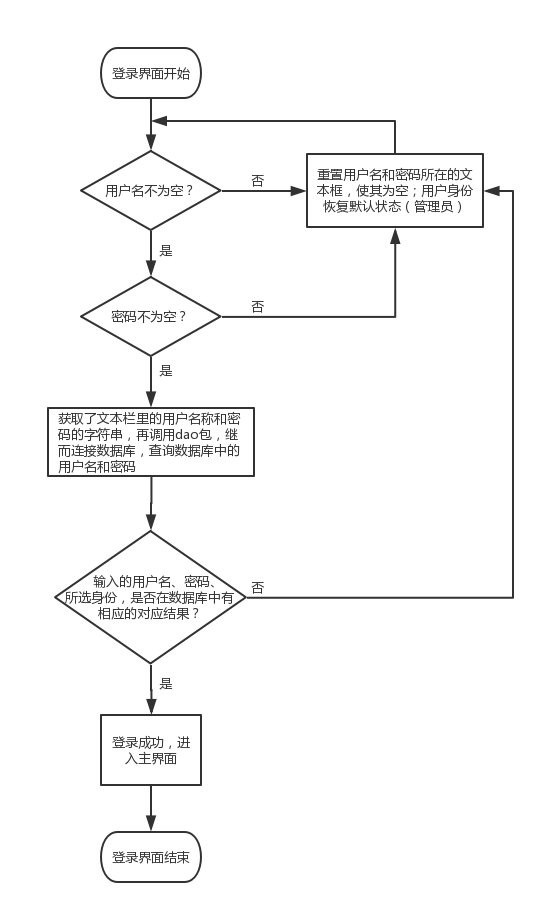
}else {

//学生登录

}

}

对应的流程图如下：



（3）主界面模块设计

在主界面中使用了BorderLayout布局管理器，使用JMenuBar类来创建菜单栏，使用JMenuBar的构造函数初始化菜单栏。在主页面中，针对登录用户身份的不同实现了权限设置。关键代码如下：

private void setAuthority() {

if("学生".equals(usersType.getName())) {

//对学生用户来说，添加学生、班级管理、教师管理的菜单不可按

addStudentMenuItem.setEnabled(false);

manageClassMenu.setEnabled(false);

manageTeacherMenu.setEnabled(false);

}

if("教师".equals(usersType.getName())) {

//对教师用户来说，添加教师的菜单不可按

addTeacherMenuItem.setEnabled(false);

}

}

（4）学生管理模块设计

由于学生管理、班级管理、教师管理三个菜单下的功能大同小异，以学生管理菜单为例。

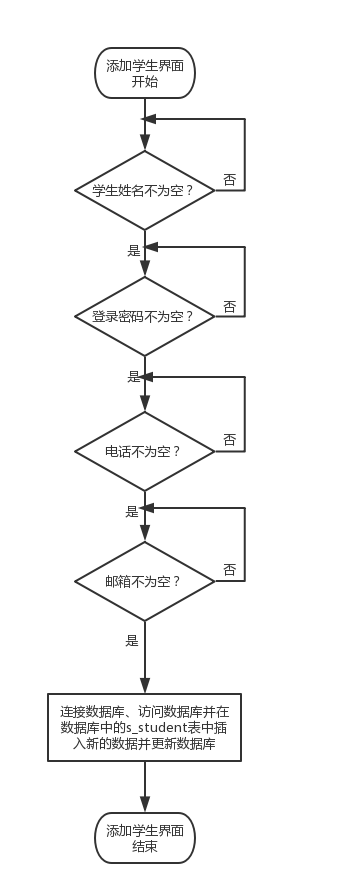
添加学生操作与登录操作部分相同，相同的是文本框不可为空，为空时则显示提示。不同的是添加学生的操作需要连接数据库、访问数据库并在数据库中的s\_student表中插入新的数据，sql语句如下：

insert into s\_student values(null,?,?,?,?,?,?)

插入后实现更新操作，sql语句如下：

update s\_student set name=? , classId=? , cellphoneNumber=? ,mail=? where id=?

对应流程图如下：



学生列表窗体继承了JInternalFrame类，初始化了JTable以显示学生列表信息。

查询操作是获取了填进文本栏的字符串，进行筛选显示，重新生成学生列表信息。当获取到的字符串为空时，显示初始的学生列表。

修改和删除操作是为学生列表的每一栏添加了监听事件，重写了addMouseListener()方法，当鼠标点击其中某一栏时，该栏显示被选中，关键代码如下：

classListTable = new JTable();

classListTable.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent me) {

selectedTableRow(me);

}

});

选中后，该学生的信息会在上半部分的修改栏显示，获取了JTable中的分栏数据，依次设置到各个文本栏、下拉栏中。关键代码如下：

protected void submitEditAct(ActionEvent ae) {

String studentName = editStudentNameTextField.getText().toString();

String cellphoneNumber = editCellphoneNumberTextField.getText().toString();

String mail = editMailTextField.getText().toString();

int row = studentListTable.getSelectedRow();

/\*\*\*\*

为了报告简洁，省略一系列检验是否选中学生、是否填写了姓名、电话号码、邮箱的操作

\*\*\*\*/

Student student = new Student();

student.setName(studentName );

GradeClass gc = (GradeClass)editStudentClassComboBox.getSelectedItem();

student.setId(Integer.parseInt(studentListTable.getValueAt(row, 0).toString()));

student.setCellphoneNumber(cellphoneNumber);

student.setMail(mail);

student.setClassId(gc.getId());

StudentDao studentDao = new StudentDao();

//检验修改数据库中的s\_student表是否成功

if(studentDao.update(student)) {

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

}else {

JOptionPane.showMessageDialog(this, "修改失败！");

}

studentDao.closeDao();

setTable(new Student());

}

选中某一学生后，点击删除键时，会弹窗询问是否确认删除该学生，如确认则删除。sql语句如下：

delete from s\_student where id=?

## 三、测试分析

1、

（1）阐述扩展包

jgoodies-forms-1.8.0-sources.jar

jgoodies-forms-1.8.0.jar

miglayout15-swing.jar

（2）数据库配置说明

1)数据库E-R图如图所示

2)s\_admin（管理员表）

管理员表包括管理员id、管理员姓名、密码、创建日期。

3)s\_class（班级表）

班级表包括班级id、班级名称、班级描述

4)s\_student（学生表）

学生表包括学生id、学生姓名、性别、密码、所属班级id、联系电话、邮箱

5)s\_teacher（教师表）

教师表包括教师id、教师姓名、密码、所教年级、所教班级id、联系电话、邮箱

2、典型测试与预计测试结果

1)登录测试

2)修改密码测试

3)添加学生测试

4)学生管理->学生列表中查询、修改、删除学生测试

5)添加班级测试

6)班级管理->班级列表->查询、修改、删除班级测试

7)添加教师测试

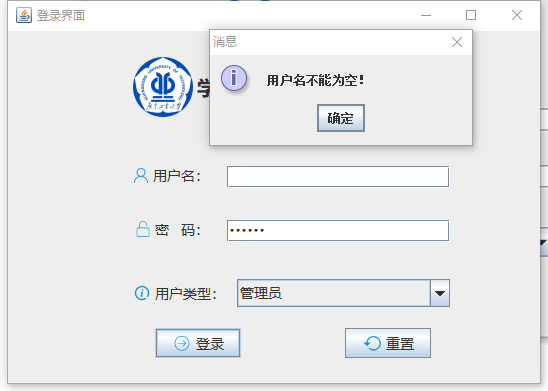
8)教师管理->教师列表->查询、修改、删除教师测试

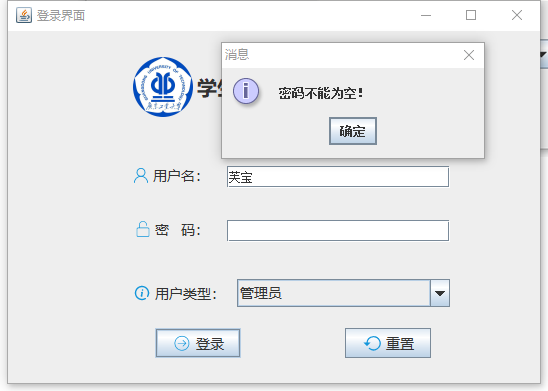
9)关于我们->了解更多跳转测试

预计成果：上述功能全部测试成功

3、测试结果

1)登录测试

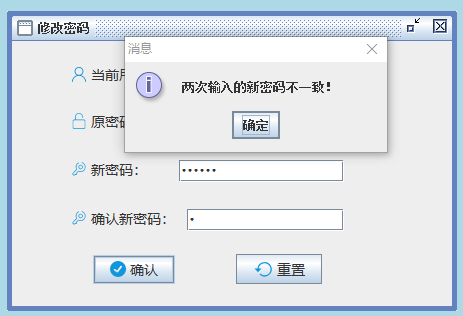


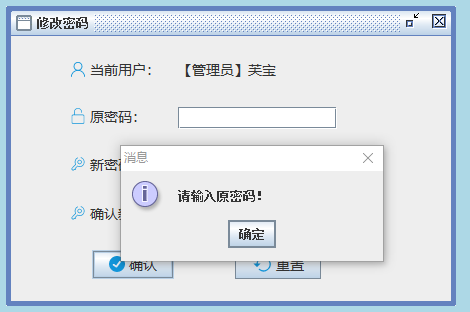


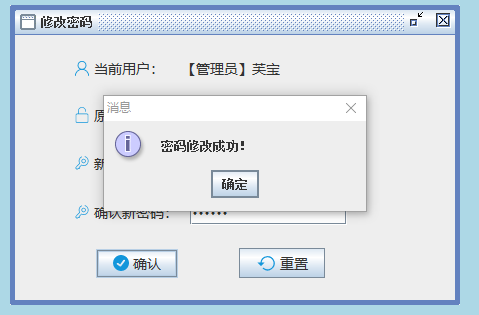


成功登录

2)修改密码测试







修改密码成功

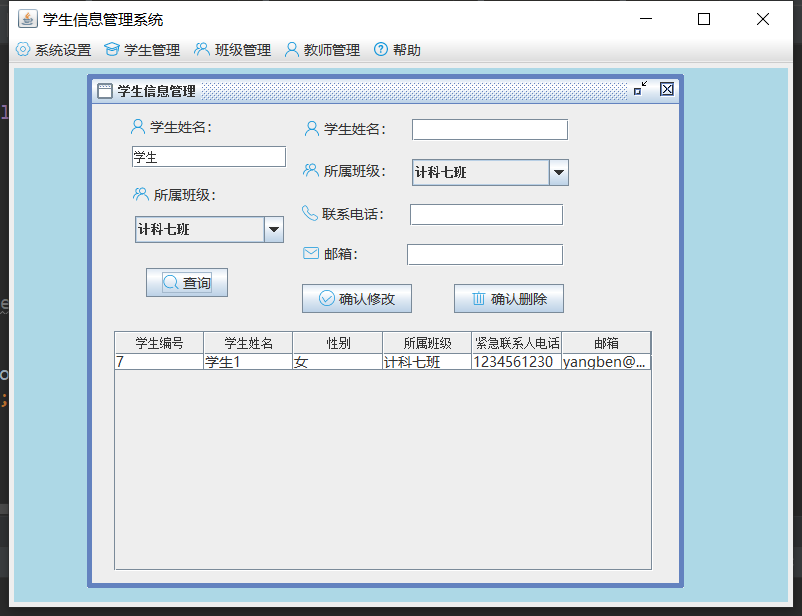
3)添加学生测试



添加成功

4)学生管理->学生列表中查询、修改、删除学生测试

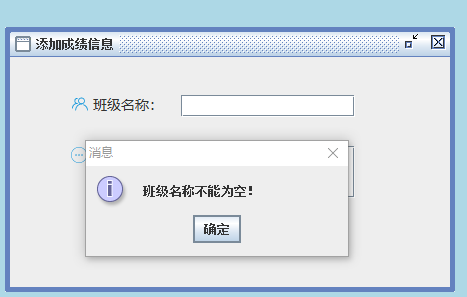


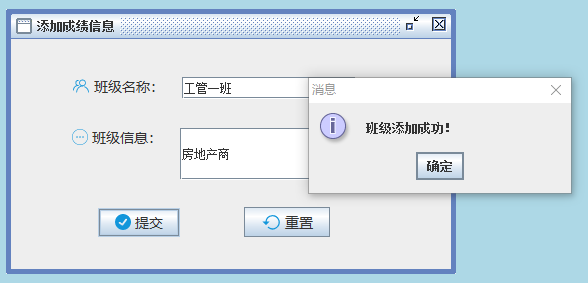


查询、修改、删除均成功

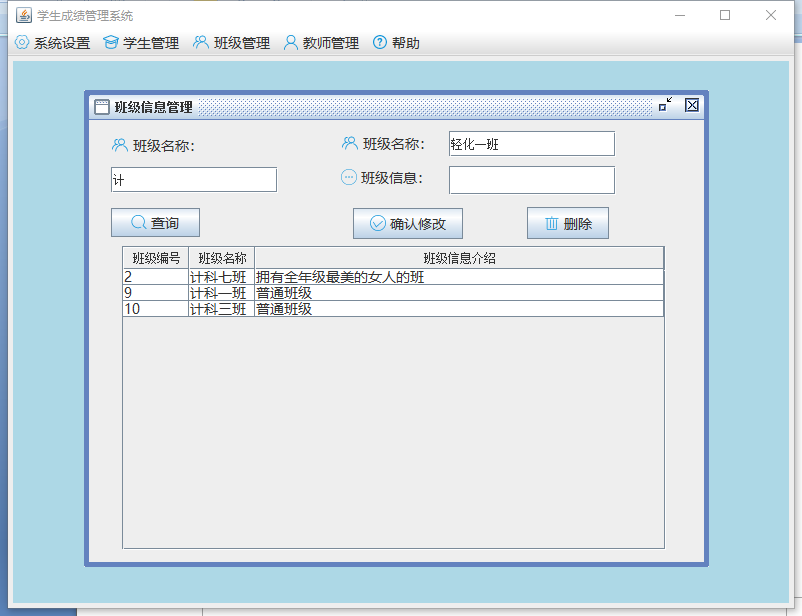
5)添加班级测试

添加班级成功



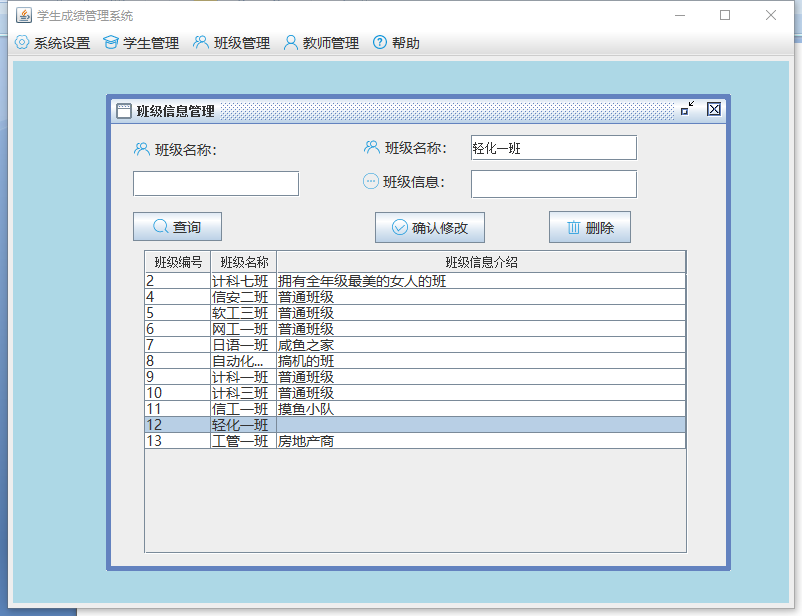


6)班级管理->班级列表->查询、修改、删除班级测试

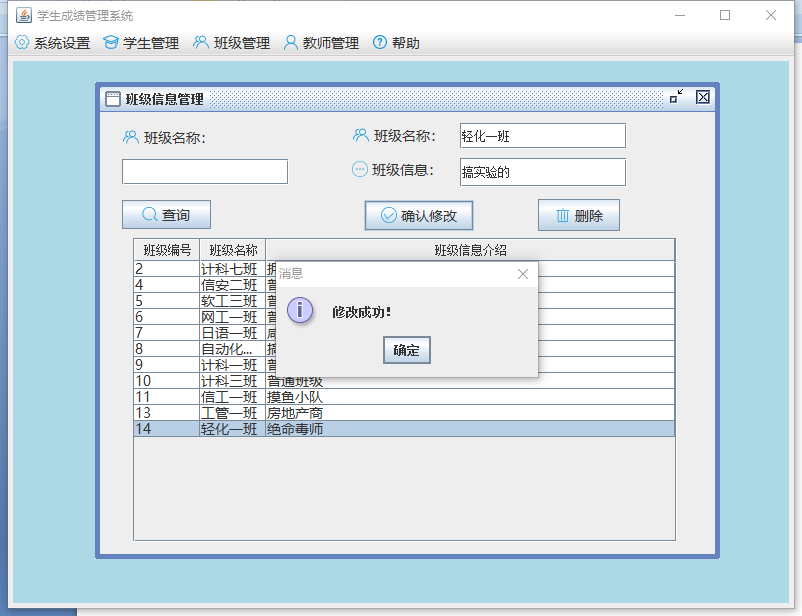


查询成功

修改之前

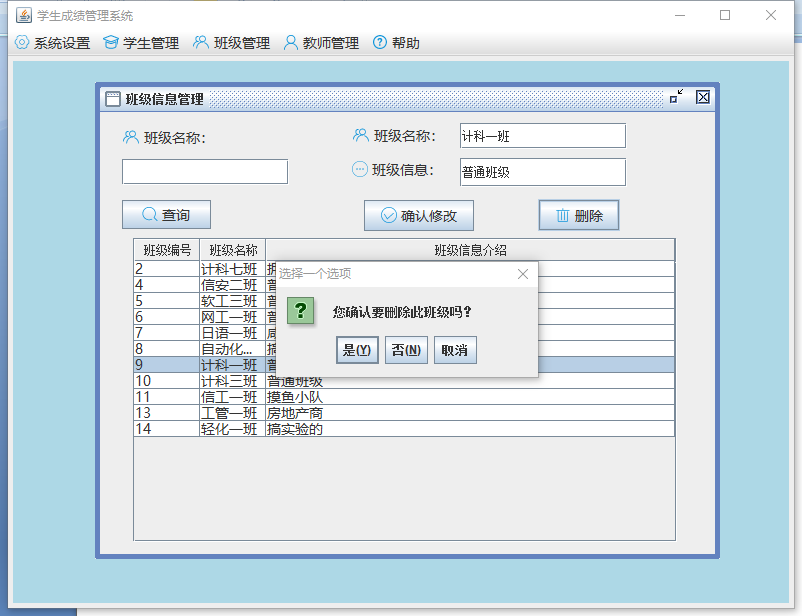


修改之后

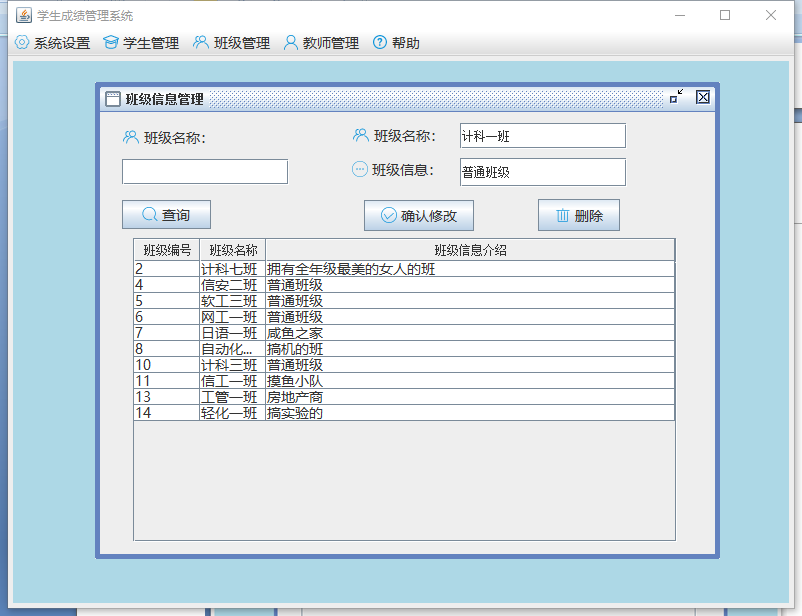


修改操作成功

删除之前



删除之后

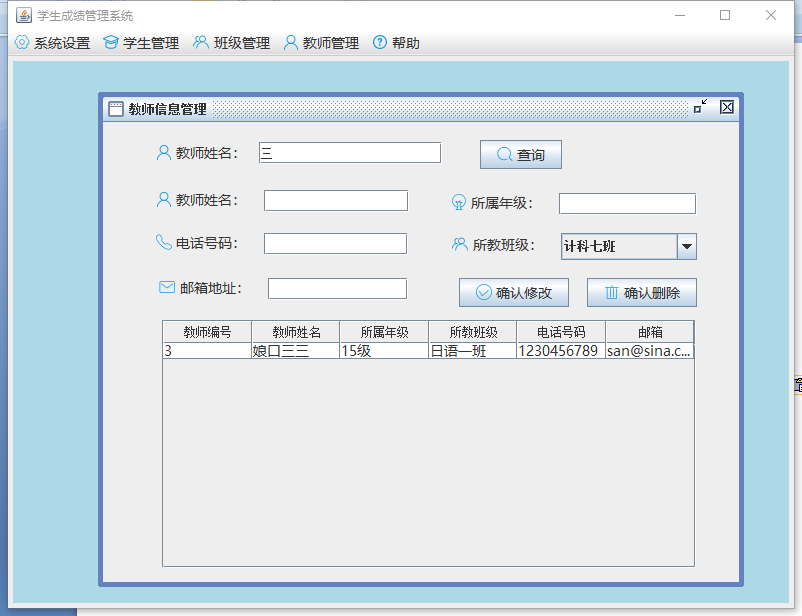


删除操作成功

7)添加教师测试

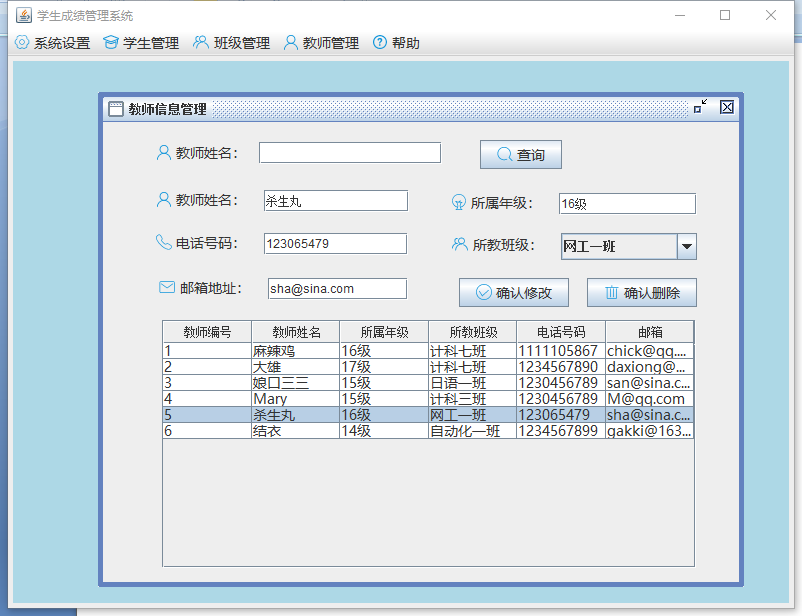
添加教师成功

8)教师管理->教师列表->查询、修改、删除教师测试

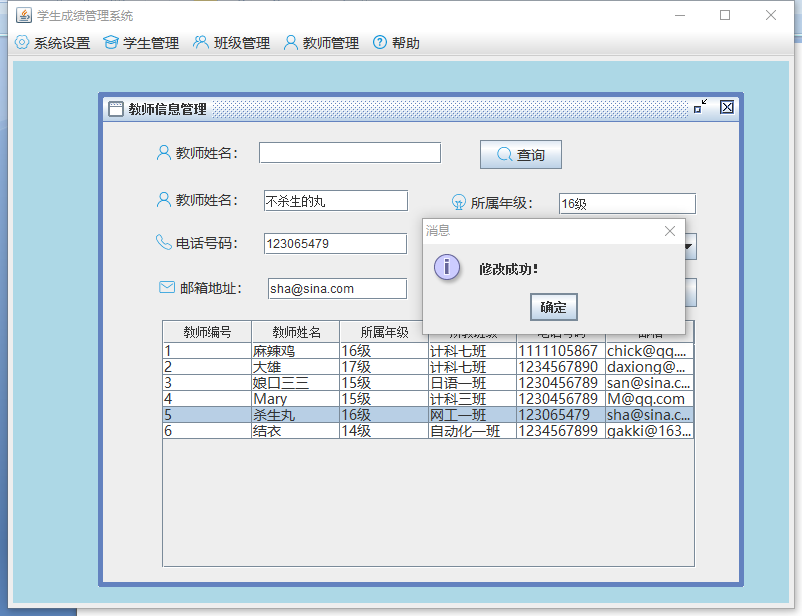


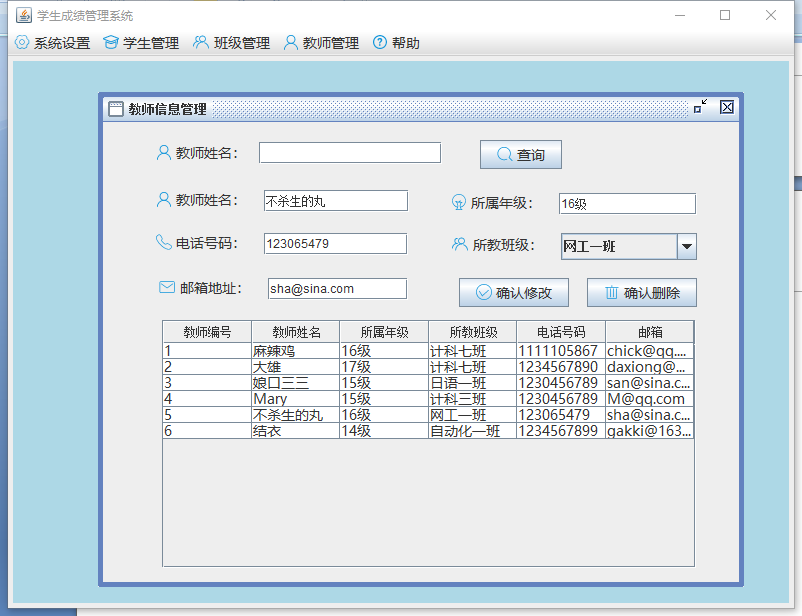
模糊查询成功

修改之前

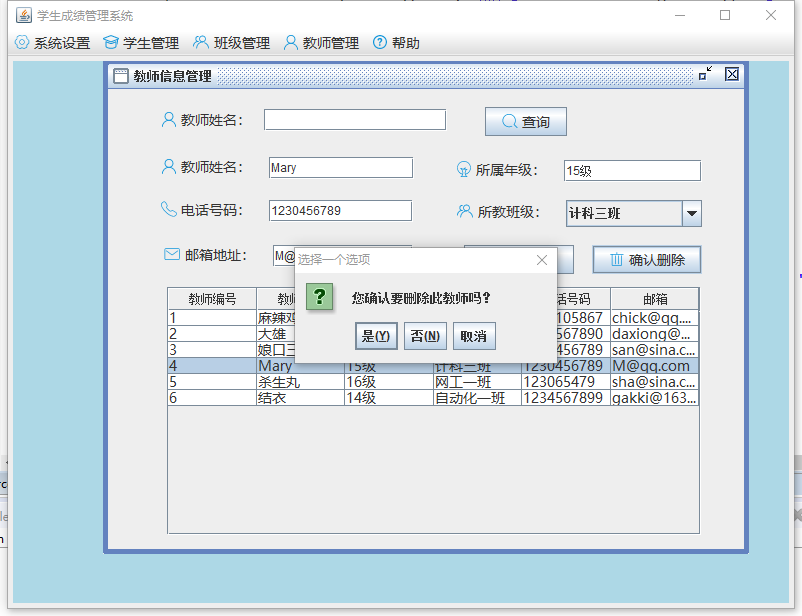


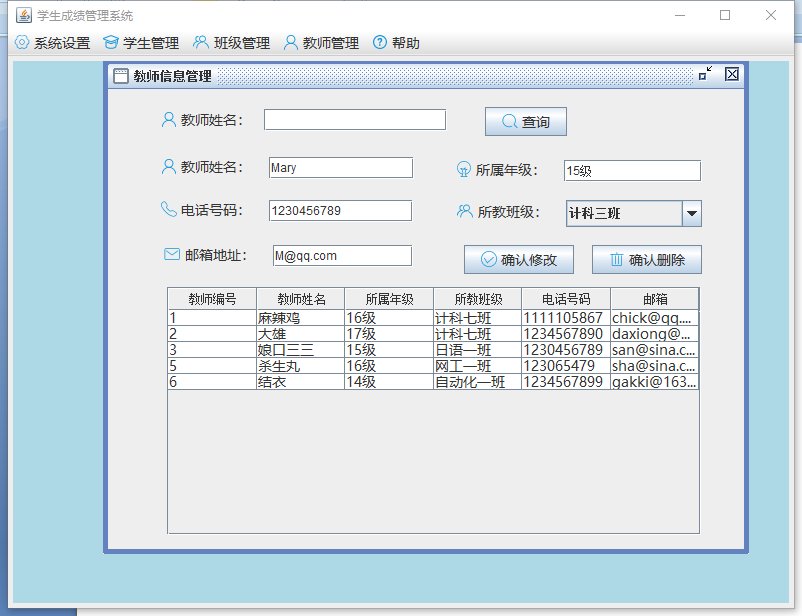
修改之后





删除之前

  
删除之后



9)关于我们->了解更多跳转测试



点击了解更多可跳转到默认浏览器打开广东工业大学教务系统，跳转测试成功

四、设计体会

改进之处：

1. 根据我写的学生信息管理系统，由于用到了数据库，发现在删除班级时，有时候会删除成功，有时候会报错，这是因为在其他表中（s\_student、s\_teacher）中引用了外键classId关联到了s\_class中的主键id，当所要被删除的班级，是某个学生或者教师所属班级的话，就会报错。
2. 数据库连接时，每一次new一个dao类的时候都会获得一个连接，然后用完之后关掉，理论上这个操作没问题，但是创建连接和销毁连接的代价比较昂贵，如果操作频繁的话就意味着浪费很多性能，可以改进。