·

**沈阳工学院**

**C#程序设计**

**实验报告**

题目： 学生成绩管理系统

院 系： 信息与控制学院

专 业： 计科二班

班级学号： 1530110331

学生姓名： 张东源

指导教师： 吴晓艳

成 绩：

2017年5月11 日

**实验目的：**

通过本项目的完成，使学生掌握C#程序设计课程的核心内容：包括编程基础、流程控制语句、面向对象、数据访问和Windows窗体的应用。通过综合实验项目的完成，既能将理论知识综合贯通，加深理论知识的理解和掌握，又能锻炼学生的动手能力及实际应用能力。

**实验要求：**

该综合实验项目延续课堂上所讲授的学生成绩管理系统项目的功能，要求学生分组完成系统管理、教师管理、专业管理、班级管理、课程管理和成绩管理的功能，学生能够根据课程所学，完成数据类、数据操作类、窗体界面的设计及功能的实现。

# 目 录

[1 系统设计 2](#_Toc407175310)

[1.1系统功能设计 2](#_Toc407175311)

[1.2 数据库设计 2](#_Toc407175312)

[2 系统实现 5](#_Toc407175313)

[2.1 用户管理功能的实现 5](#_Toc407175314)

2.2 学生管理功能的实现……………………………………………………… ..10

2.3 课程管理功能的实现………………………………………………………...15

2.4 教室管理功能的实现………………………………………………………...20

2.5 成绩管理功能的实现………………………………………………………...25

2.6 班级管理功能的实现……………………………………………………….. 30

2.7 专业管理功能的实现 ……………………………………………………… 35

# 1 系统设计

## 1.1系统功能设计

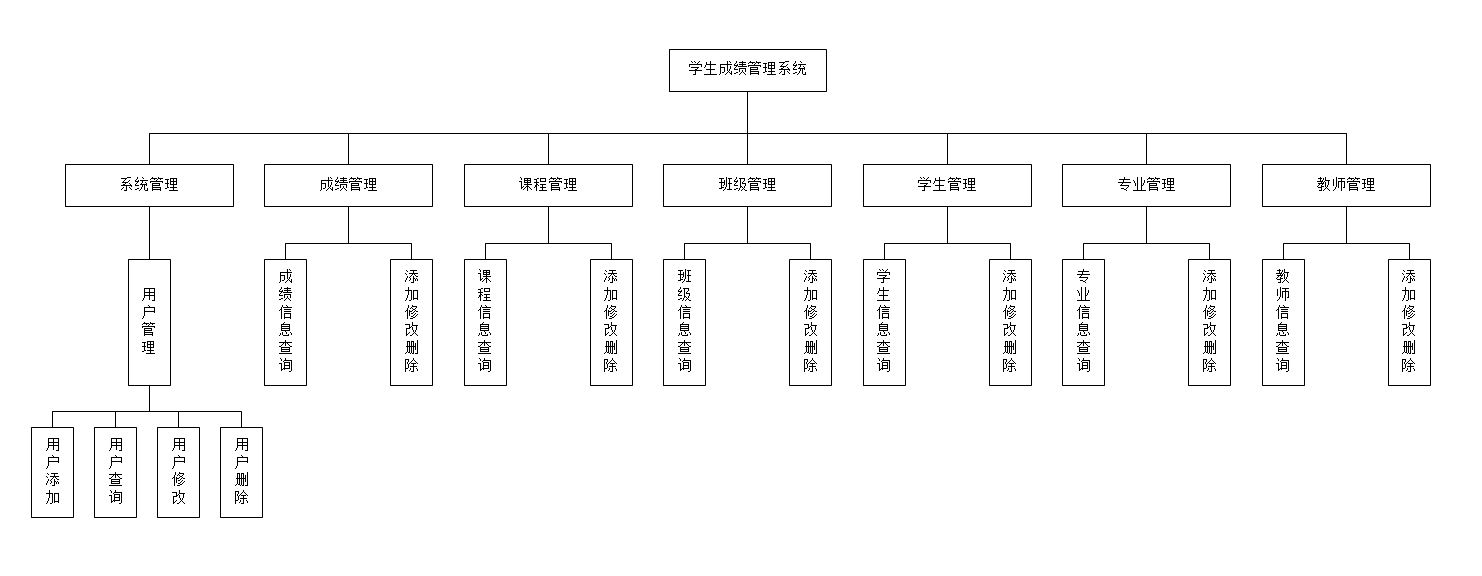
学上成绩管理系统的功能结构图如图1.1所示。

图1.1学生成绩管理系统系统功能模块图

## 1.2 数据库设计

学生成绩管理系统数据库的名称为：xscjgl，为完成如上功能，共设计了7个数据表，分别为：用户信息表、图书类别表、图书信息表、购物车表、订单表和新闻表。

用户表结构设计如表1.1所示。

表1.1用户表

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型（长度）** | **说明（包括约束）** |
| userid | varchar(10) | 用户编号（主键） |
| username | varchar(10) | 用户名称 |
| userlevel | varchar(10) | 用户名称 |

表1.2专业表

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型（长度）** | **说明（包括约束）** |
| specialtyid | varchar(4) | 专业编号（主键） |
| specialtymc | varchar(50) | 专业名称 |

表1.3班级表

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型（长度）** | **说明（包括约束）** |
| classid | varchar(10) | 班级编号（主键） |
| specialtyid | varchar(4) | 专业编号 |
| studentnumber  remark | int  varchar(100) | 班级人数  备注 |

表1.4教师表

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型（长度）** | **说明（包括约束）** |
| teaid | varchar(10) | 教师编号（主键） |
| teaname | varchar(20) | 教师名称 |
| teasex  teloffice  Address  teabirthdaay | varchar(2)  varchar(11)  varchar(100)  datetime | 教师性别  教师办公电话  教师地址  教师出生日期 |

表1.5学生表

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型（长度）** | **说明（包括约束）** |
| sno | varchar(8) | 学生编号（主键） |
| sname | varchar(20) | 学生名称 |
| sex  birthday  classid  tel  address | varchar(2)  datetime  vatchar(10)  varchar(20)  varchar(100) | 学生性别  学生出生日期  班级号  电话号码  地址 |

表1.6成绩表

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型（长度）** | **说明（包括约束）** |
| sno | varchar(8) | 学生编号（主键） |
| Cno  gradepeacetime | varchar(10)  Number(9,1) | 课程号  平时成绩 |
| gradeexpriment  gradelast  grade | Number(9,1)  Number(9,1)  Number(9,1) | 实验成绩  期末成绩  总成绩 |

表1.7课程表

|  |  |  |
| --- | --- | --- |
| **字段名称** | **数据类型（长度）** | **说明（包括约束）** |
| kid | varchar(10) | 课程编号（主键） |
| kcname  periodexpriment | varchar(50)  int | 课程名称  课程实验学时 |
| periodteaching  creadit  coursetype | int  Number(9,1)  varchar(50) | 课程讲课学时  课程学分  课程类型 |

# 2 系统实现

## 2.1 用户管理功能的实现

用户管理是为了管理现有的用户。在该模块中可以新增、修改、删除、查询用户。添加用户需添加用户的编号、用户姓名、用户密码。查询的时候，输入相应的用户编号或用户姓名，则会在datagridview界面中显示对应的用户。删除和修改用户的时候，如果删除或修改成功，则会提示删除成功或修改成功。实现界面如下图：

2.1.1 用户信息的增加如图所示：

图2.1.1 用户管理增加界面

实现过程：

该界面主要有button控件、lable控件、textbox控件。通过UserInfoOperation连接数据库，将textbox的内容响应到数据库中。其中用户名不能为空，若是空值，则无法添加。其中添加用户的主要代码为：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using 学生成绩管理系统.Classes;

namespace 学生成绩管理系统

{

public partial class UserAdd : Form

{

public UserAdd()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

UserInfoData data = new UserInfoData();

data.Userid = txtUserid.Text;

data.Userpwd = txtUserpwd.Text;

data.Userlevel = txtUserlevel.Text;

if (txtUserid.Text == "" || txtUserid.Text == null)

{

MessageBox.Show("用户名不能为空");

txtUserid.Focus();

return;

}

Classes.UserInfoOperation op = new Classes.UserInfoOperation();

if (op.InsertUserInfo(data))

MessageBox.Show("插入成功");

else

MessageBox.Show("失败");

}

}

}

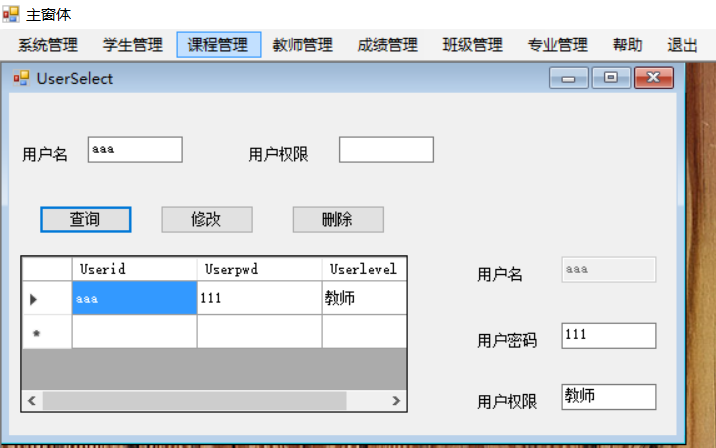
2.1.2 用户信息的查询如图所示：

图2.1.2 用户管理查询界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。lable控件的enable属性设置为True，通过UserInfoOperation连接数据库，输入用户姓名或用户权限，点击查询，数据库中的数据就会相应到datagridview中。其中查询用户的主要代码为:

private void UserSelect\_Load(object sender, EventArgs e)

{dataGridView1.DataSource = Bind().Tables[0]; }

DataSet Bind()

{

UserInfoData data = new UserInfoData();

data.Userid = txtUserid.Text;

data.Userlevel = txtUserlevel.Text;

Classes.UserInfoOperation op = new Classes.UserInfoOperation();

ds = op.GetUserInfo(data);

return ds;

}

//查询

private void button1\_Click(object sender, EventArgs e)

{

dataGridView1.DataSource = Bind().Tables[0];}

//dataGridView显示数据

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int i = e.RowIndex;

txtUserid1.Text = ds.Tables[0].Rows[i][0].ToString();

txtUserpwd1.Text = ds.Tables[0].Rows[i][1].ToString();

txtUserlevel1.Text = ds.Tables[0].Rows[i][2].ToString();

}

2.1.3 用户信息的修改如图所示：

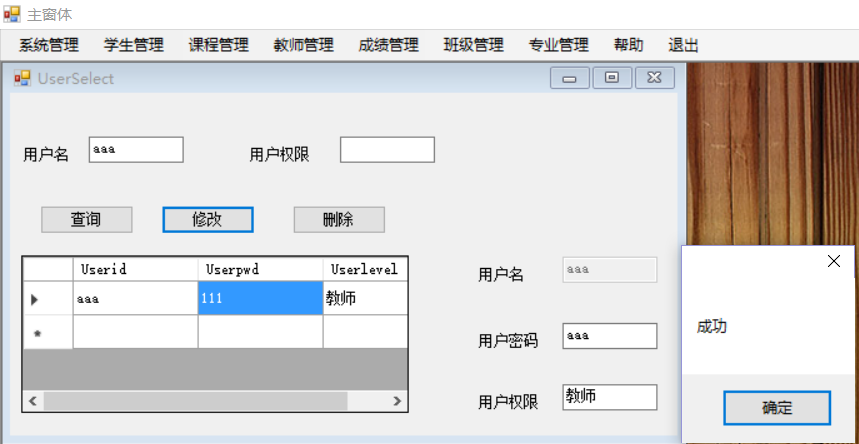


图2.1.3 用户管理修改界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过UserInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，响应到右边的文本框中，则可以进行修改，其中用户名不能修改，因为lable控件的用户名的enable属性设置为True。其中修改用户的主要代码为:

//修改

private void button2\_Click(object sender, EventArgs e)

{

UserInfoData data = new UserInfoData();

data.Userid = txtUserid1.Text;

data.Userpwd = txtUserpwd1.Text;

data.Userlevel = txtUserlevel1.Text;

Classes.UserInfoOperation op = new Classes.UserInfoOperation();

if (op.UpdateUser(data))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];

}

2.1.4 用户信息的删除如图所示：

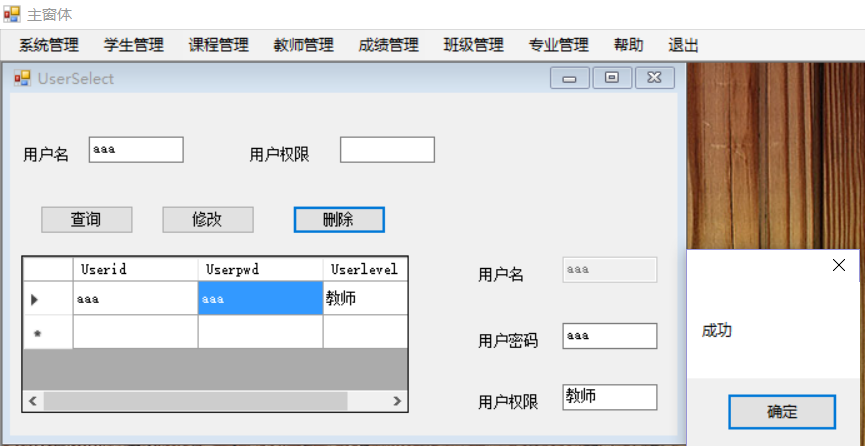


图2.1.4 用户管理删除界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过UserInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，即可以删除该数据。其中删除用户的主要代码为:

private void button3\_Click(object sender, EventArgs e)

{

string sno = txtUserid1.Text;

Classes.UserInfoOperation op = new Classes.UserInfoOperation();

if (op.DeleteUser(sno))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];

}

## 2.2 学生管理功能的实现

学生管理是为了管理现有的学生。在该模块中可以新增、修改、删除、查询学生信息。添加学生需添加学生的学号、姓名、性别、出生日期、班级、电话、地址。查询的时候，输入相应的学生学号、姓名、性别或班级，则会在datagridview界面中显示对应的用户。删除和修改学生信息的时候，如果删除或修改成功，则会提示删除成功或修改成功。实现界面如下图：

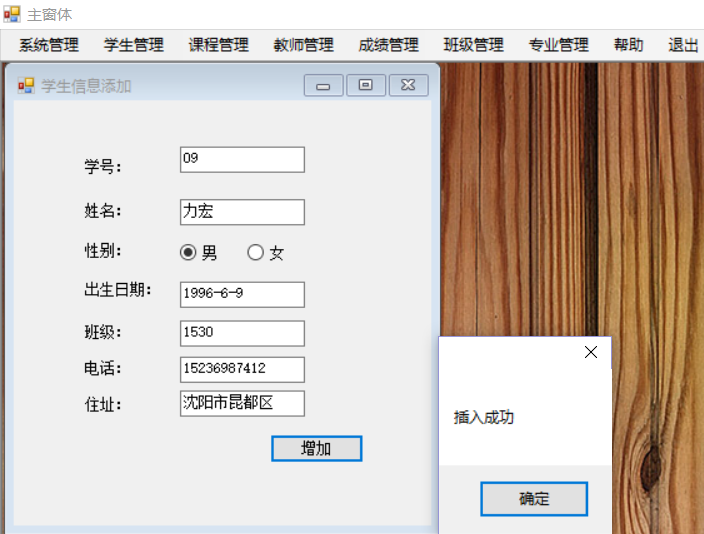
2.2.1 学生信息的增加如图所示：

图2.2.1 学生管理增加界面

实现过程：

该界面主要有button控件、lable控件、textbox控件。通过StudentInfoOperation连接数据库，将textbox的内容响应到数据库中。其中学生姓名不能为空，若是空值，则无法添加。其中添加用户的主要代码为：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace 学生成绩管理系统

{

public partial class StudentAdd : Form

{public StudentAdd()

{InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

StudentInfoData data = new StudentInfoData();

data.Sno = txtSno.Text;

data.Sname = txtSname.Text;

if (rdoMale.Checked)

data.Sex = rdoMale.Text;

else

data.Sex = rdoFemale.Text;

data.Birthday = txtBirthday.Text;

data.Classid = txtClassid.Text;

data.Tel = txtTel.Text;

data.Address = txtAddress.Text;

if (txtSno.Text == "" || txtSno.Text == null)

{

MessageBox.Show("学号不能为空");

txtSno.Focus();

return;

}

Classes.StudentInfoOperation op = new Classes.StudentInfoOperation();

if (op.InsertStudentInfo(data))

MessageBox.Show("插入成功");

else

MessageBox.Show("失败");

}

}}

2.2.2 学生信息的查询如图所示：

图2.2.2 学生管理查询界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。lable控件的enable属性设置为True，通过studentInfoOperation连接数据库，输入学号、姓名、性别或班级，点击查询，数据库中的数据就会相应到datagridview中。其中查询用户的主要代码为:

private void StudentSelect\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = Bind().Tables[0];

}

DataSet Bind()

{

StudentInfoData data = new StudentInfoData();

data.Sno = txtSno.Text;

data.Sname = txtSname.Text;

data.Sex = txtSex.Text;

data.Classid = txtClass.Text;

Classes.StudentInfoOperation op = new Classes.StudentInfoOperation();

ds = op.GetStudentInfo(data);

return ds;

}

//查询

private void button1\_Click(object sender, EventArgs e)

{

dataGridView1.DataSource = Bind().Tables[0];

}

//显示数据

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int i = e.RowIndex;

txtSno1.Text = ds.Tables[0].Rows[i][0].ToString();

txtSname1.Text = ds.Tables[0].Rows[i][1].ToString();

txtSex1.Text = ds.Tables[0].Rows[i][2].ToString();

txtBirthday1.Text = ds.Tables[0].Rows[i][3].ToString();

txtClassid1.Text = ds.Tables[0].Rows[i][4].ToString();

txtTel1.Text = ds.Tables[0].Rows[i][5].ToString();

txtAddress1.Text = ds.Tables[0].Rows[i][6].ToString();

}

2.2.3 学生信息的修改如图所示：

图2.2.3 学生管理修改界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过studentInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，响应到右边的文本框中，则可以进行修改，其中学号不能修改，因为lable控件的用户名的enable属性设置为True。其中修改用户的主要代码为:

private void button2\_Click(object sender, EventArgs e)

{

StudentInfoData data = new StudentInfoData();

data.Sno = txtSno1.Text;

data.Sname = txtSname1.Text;

data.Sex = txtSex1.Text;

data.Birthday = txtBirthday1.Text;

data.Classid = txtClass.Text;

data.Tel = txtTel1.Text;

data.Address = txtAddress1.Text;

Classes.StudentInfoOperation op = new Classes.StudentInfoOperation();

if (op.UpdateStudent(data))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];

}

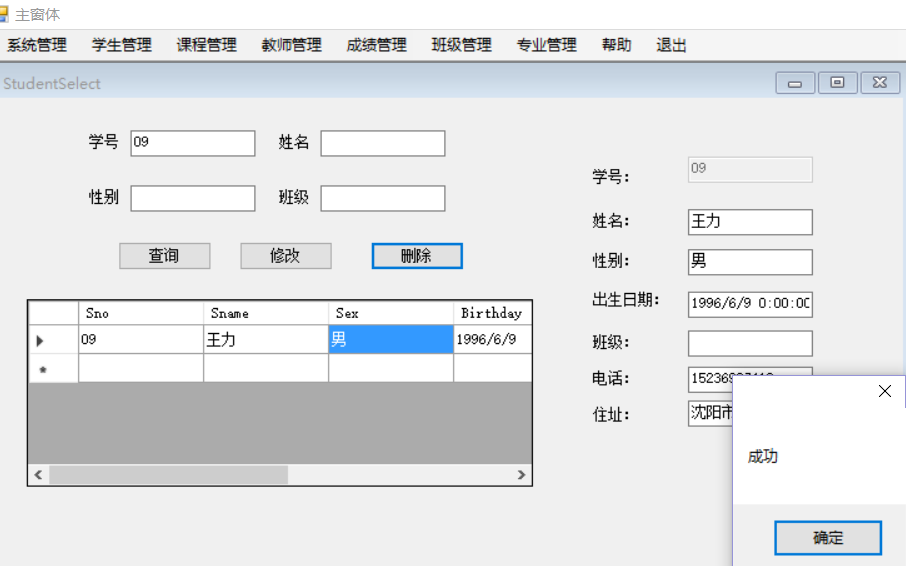
2.2.4 学生信息的删除如图所示：

图2.2.4 学生管理删除界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过UserInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，即可以删除该数据。其中删除学生信息的主要代码为:

private void button3\_Click(object sender, EventArgs e)

{

string sno = txtSno1.Text;

Classes.StudentInfoOperation op = new Classes.StudentInfoOperation();

if(op.DeleteStudent(sno))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];

}

## 2.3 课程管理功能的实现

课程管理是为了管理现有的课程。在该模块中可以新增、修改、删除、查询课程信息。添加课程需添加课程的课程编号、课程名称、课程实验学时、课程讲科学时、课程学分、课程类型。查询的时候，输入相应的课程编号、课程实验学时，则会在datagridview界面中显示对应的课程。删除和修改课程信息的时候，如果删除或修改成功，则会提示删除成功或修改成功。实现界面如下图：

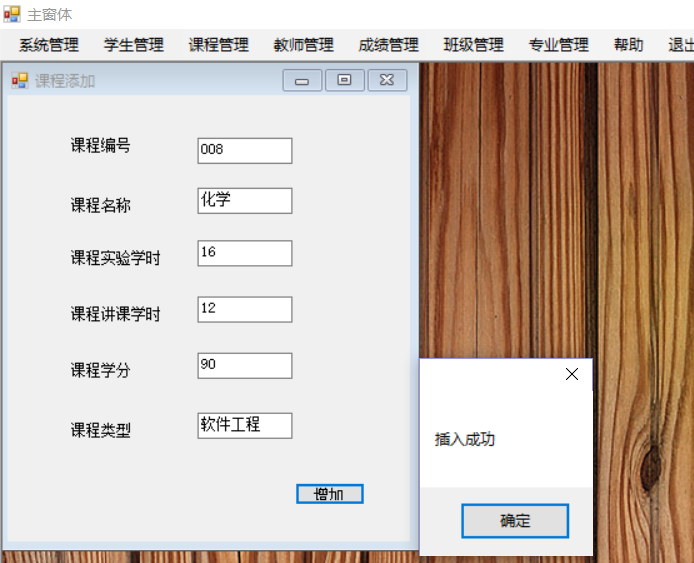
2.3.1 课程信息的增加如图所示：

图2.3.1 课程管理增加界面

实现过程：

该界面主要有button控件、lable控件、textbox控件。通过courseInfoOperation连接数据库，将textbox的内容响应到数据库中。其中学生姓名不能为空，若是空值，则无法添加。其中添加用户的主要代码为：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using 学生成绩管理系统.Classes;

namespace 学生成绩管理系统

{

public partial class CourseAdd : Form

{ public CourseAdd()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

CourseInfoData data = new CourseInfoData();

data.Kcid = txtKcid.Text;

data.Kcname = txtKcname.Text;

data.Periodexpriment = int.Parse(txtPeriodexpriment.Text);

data.Periodteaching = int.Parse(txtPeriodteaching.Text);

data.Credit = float.Parse(txtCredit.Text);

data.Coursetype = txtCoursetype.Text;

if (txtKcid.Text == "" || txtKcid.Text == null)

{

MessageBox.Show("课程编号不能为空");

txtKcid.Focus();

return;

}

Classes.CourseInfoOperation op = new Classes.CourseInfoOperation();

if (op.InsertCourseInfo(data))

MessageBox.Show("插入成功");

else

MessageBox.Show("失败");

} }}

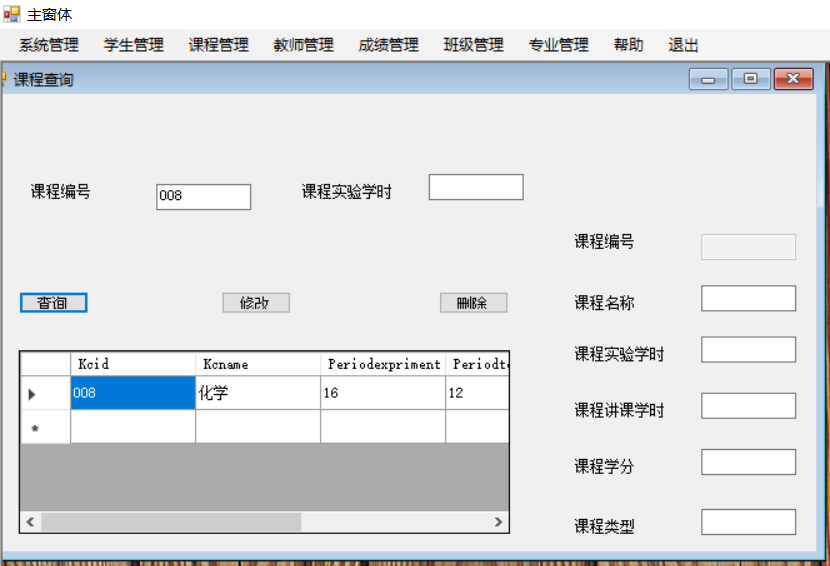
2.3.2 课程信息的查询如图所示：

图2.3.2 课程管理查询界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。lable控件的enable属性设置为True，通过courseInfoOperation连接数据库，输入课程编号、课程实验学时，点击查询，数据库中的数据就会响应到datagridview中。其中查询用户的主要代码为:

private void CourseSelect\_Load(object sender, EventArgs e)

{dataGridView1.DataSource = Bind().Tables[0];

}

DataSet Bind()

{

CourseInfoData data = new CourseInfoData();

data.Kcid = txtKcid.Text;

data.Kcname = txtKcname.Text;

Classes.CourseInfoOperation op = new Classes.CourseInfoOperation();

ds = op.GetCourseInfo(data);

return ds;

}

private void button1\_Click(object sender, EventArgs e)

{

dataGridView1.DataSource = Bind().Tables[0];

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int i = e.RowIndex;

txtKcid1.Text = ds.Tables[0].Rows[i][0].ToString();

txtKcname1.Text = ds.Tables[0].Rows[i][1].ToString();

txtPeriodexpriment1.Text = ds.Tables[0].Rows[i][2].ToString();

txtPeriodteaching1.Text = ds.Tables[0].Rows[i][3].ToString();

txtCredit1.Text = ds.Tables[0].Rows[i][4].ToString();

txtCoursetype1.Text = ds.Tables[0].Rows[i][5].ToString();}

2.3.3 课程信息的修改如图所示：

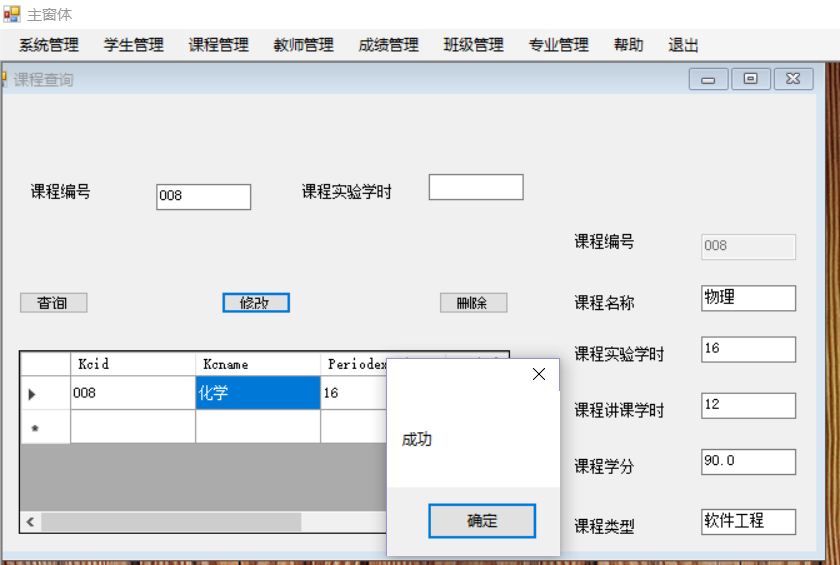


图2.3.2 课程管理修改界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过courseInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，响应到右边的文本框中，则可以进行修改，其中学号不能修改，因为lable控件的用户名的enable属性设置为True。其中修改用户的主要代码为:

private void button2\_Click(object sender, EventArgs e)

{

CourseInfoData data = new CourseInfoData();

data.Kcid = txtKcid1.Text;

data.Kcname = txtKcname1.Text;

data.Periodexpriment = int.Parse(txtPeriodexpriment1.Text);

data.Periodteaching = int.Parse(txtPeriodteaching1.Text);

data.Credit = float.Parse(txtCredit1.Text);

data.Coursetype = txtCoursetype1.Text;

Classes.CourseInfoOperation op = new Classes.CourseInfoOperation();

if (op.UpdateCourse(data))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];}

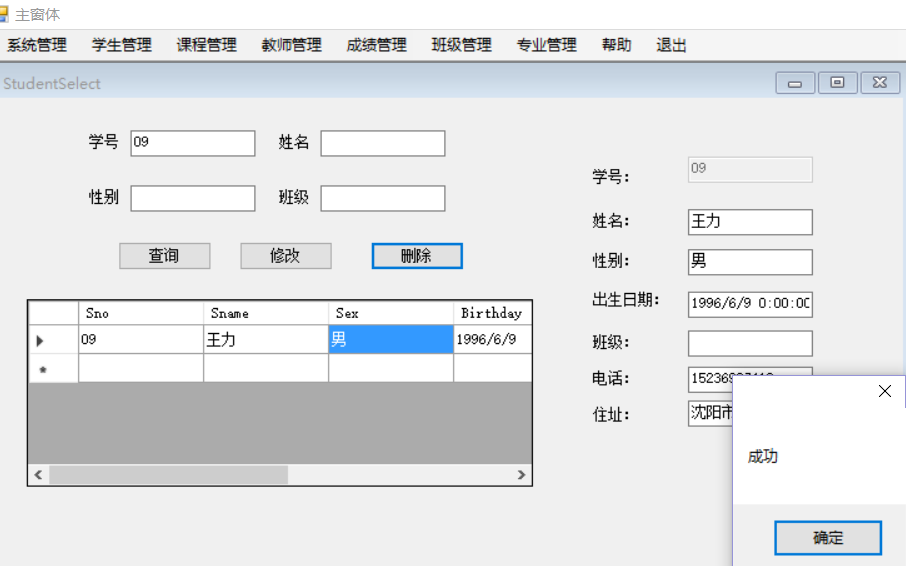
2.3.4 课程信息的删除如图所示：

图2.3.2 课程管理删除界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过UserInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，即可以删除该数据。其中删除课程信息的主要代码为:

private void button3\_Click(object sender, EventArgs e)

{string kcid = txtKcid1.Text;

Classes.CourseInfoOperation op = new Classes.CourseInfoOperation();

if (op.DeleteCourse(kcid))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];}

## 2.4 教师管理功能的实现

教师管理是为了管理现有的教师。在该模块中可以新增、修改、删除、查询教师信息。添加教师需添加教师编号、教师名称、教师实验学时、教师讲科学时、教师学分、教师类型。查询的时候，输入相应的教师编号、教师实验学时，则会在datagridview界面中显示对应的教师。删除和修改教师信息的时候，如果删除或修改成功，则会提示删除成功或修改成功。实现界面如下图：

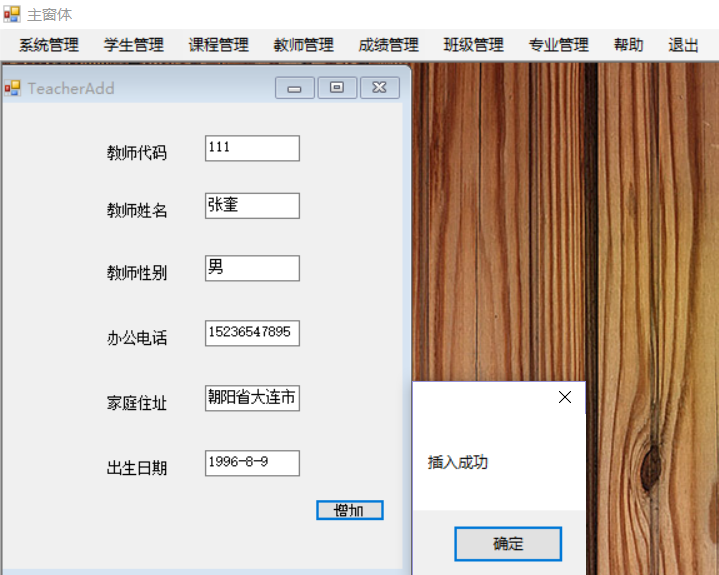
2.4.1 教师信息的增加如图所示：

图2.4.1 教师管理增加界面

实现过程：

该界面主要有button控件、lable控件、textbox控件。通过teacherInfoOperation连接数据库，将textbox的内容响应到数据库中。其中教室编号不能为空，若是空值，则无法添加。其中添加教师的主要代码为：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using 学生成绩管理系统.Classes;

namespace 学生成绩管理系统

{

public partial class TeacherAdd : Form

{

public TeacherAdd()

{

InitializeComponent();}

private void button1\_Click(object sender, EventArgs e)

{

TeacherInfoData data = new TeacherInfoData();

data.Teaid = txtTeaid.Text;

data.Teaname = txtTeaname.Text;

data.Teasex = txtTeasex.Text;

data.Teloffice = txtTeloffice.Text;

data.Address = txtAddress.Text;

data.Teabirthday = txtTeabirthday.Text;

if (txtTeaid.Text == "" || txtTeaid.Text == null)

{

MessageBox.Show("用户名不能为空");

txtTeaid.Focus();

return;}

Classes.TeacherInfoOperation op = new Classes.TeacherInfoOperation();

if (op.InsertTeacherInfo(data))

MessageBox.Show("插入成功");

else

MessageBox.Show("失败");}}}

2.4.2 教师信息的查询如图所示：

图2.4.2 教师管理查询界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。lable控件的enable属性设置为True，通过teacherInfoOperation连接数据库，输入教师编号、教师性别、教师姓名或出生日期，点击查询，数据库中的数据就会响应到datagridview中。其中查询教师的主要代码为:

private void TeacherSelect\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = Bind().Tables[0];}

DataSet Bind()

{

TeacherInfoData data = new TeacherInfoData();

data.Teaid = txtTeaid.Text;

data.Teaname = txtTeaname.Text;

data.Teasex = txtTeasex.Text;

data.Teabirthday = txtTeabirthday.Text;

Classes.TeacherInfoOperation op = new Classes.TeacherInfoOperation();

ds = op.GetTeacherInfo(data);

return ds; }

private void button1\_Click(object sender, EventArgs e)

{

dataGridView1.DataSource = Bind().Tables[0];

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int i = e.RowIndex;

txtTeaid1.Text = ds.Tables[0].Rows[i][0].ToString();

txtTeaname1.Text = ds.Tables[0].Rows[i][1].ToString();

txtTeasex1.Text = ds.Tables[0].Rows[i][2].ToString();

txtTeloffice1.Text = ds.Tables[0].Rows[i][3].ToString();

txtAddress1.Text = ds.Tables[0].Rows[i][4].ToString();

txtTeabirthday1.Text = ds.Tables[0].Rows[i][5].ToString();}

2.4.3 教师信息的修改如图所示：

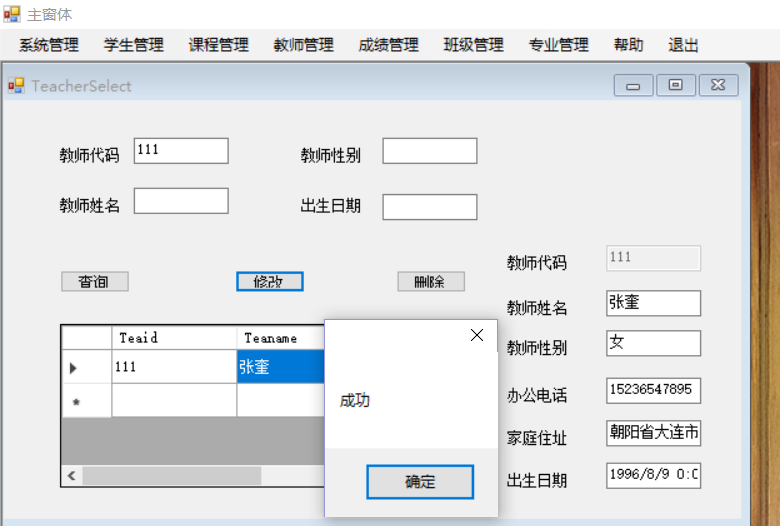


图2.4.2 教师管理修改界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过teacherInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，响应到右边的文本框中，则可以进行修改，其中学号不能修改，因为lable控件的教师编号的enable属性设置为True。其中修改教师信息的主要代码为:

private void button2\_Click(object sender, EventArgs e)

{

TeacherInfoData data = new TeacherInfoData();

data.Teaid = txtTeaid1.Text;

data.Teaname = txtTeaname1.Text;

data.Teasex = txtTeasex1.Text;

data.Teloffice = txtTeloffice1.Text;

data.Address = txtAddress1.Text;

data.Teabirthday = txtTeabirthday1.Text;

Classes.TeacherInfoOperation op = new Classes.TeacherInfoOperation();

if (op.UpdateTeacher(data))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];}

2.4.4 教师信息的删除如图所示：



图2.4.2 教师管理删除界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过teacherInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，即可以删除该数据。其中删除教师信息的主要代码为:

private void button3\_Click(object sender, EventArgs e)

{

string sno = txtTeaid1.Text;

Classes.TeacherInfoOperation op = new Classes.TeacherInfoOperation();

if (op.DeleteTeacher(sno))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];}

## 2.5 成绩管理功能的实现

成绩管理是为了管理现有的成绩。在该模块中可以新增、修改、删除、查询成绩信息。添加成绩需添加成绩编号、教师名称、教师实验学时、教师讲科学时、教师学分、教师类型。查询的时候，输入相应的教师编号、教师实验学时，则会在datagridview界面中显示对应的教师。删除和修改教师信息的时候，如果删除或修改成功，则会提示删除成功或修改成功。实现界面如下图：

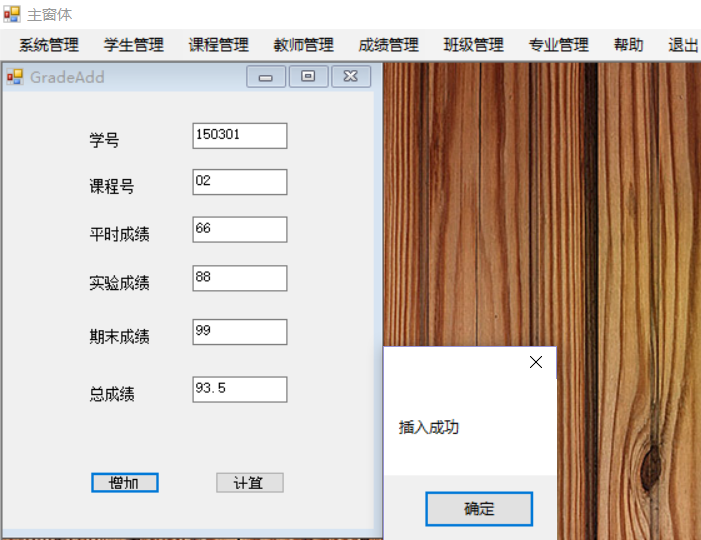
2.5.1 成绩信息的增加如图所示：

图2.5.1 成绩管理增加界面

实现过程：

该界面主要有button控件、lable控件、textbox控件。通过teacherInfoOperation连接数据库，将textbox的内容响应到数据库中。其中学号不能为空，若是空值，则无法添加。其中添加成绩的主要代码为：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using 学生成绩管理系统.Classes;

namespace 学生成绩管理系统

{

public partial class GradeAdd : Form

{

public GradeAdd()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

StuGradeData data = new StuGradeData();

data.Sno = txtSno.Text;

data.Cno = txtCno.Text;

data.Gradepeacetime = float.Parse(txtGradepeacetime.Text);

data.Gradeexpriment = float.Parse(txtGradeexpriment.Text);

data.Gradelast = float.Parse(txtGradelast.Text);

data.Grade = float.Parse(txtGrade.Text);

if (txtSno.Text == "" || txtSno.Text == null)

{

MessageBox.Show("学号不能为空");

txtSno.Focus();

return; }

Classes.StuGradeOperation op = new Classes.StuGradeOperation();

if (op.InsertStuGrade(data))

MessageBox.Show("插入成功");

else

MessageBox.Show("失败");

}

private void button2\_Click(object sender, EventArgs e)

{

int a = int.Parse(txtGradepeacetime.Text);

int b = int.Parse(txtGradeexpriment.Text);

int c = int.Parse(txtGradelast.Text);

double x = a \* 0.1 + b \* 0.2 + c \* 0.7;

float y = (float)x;

txtGrade.Text = y.ToString();

} }}

2.52 成绩信息的查询如图所示：

图2.5.2 成绩管理查询界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。lable控件的enable属性设置为True，通过teacherInfoOperation连接数据库，输入学号、课程号，点击查询，数据库中的数据就会响应到datagridview中。其中查询成绩的主要代码为:

private void GradeSelect\_Load(object sender, EventArgs e)

{dataGridView1.DataSource = Bind().Tables[0]; }

DataSet Bind()

{

StuGradeData data = new StuGradeData();

data.Sno = txtSno.Text;

data.Cno = txtCno.Text;

Classes.StuGradeOperation op = new Classes.StuGradeOperation();

ds = op.GetStuGrade(data);

return ds;}

private void button1\_Click(object sender, EventArgs e)

{dataGridView1.DataSource = Bind().Tables[0];}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int i = e.RowIndex;

txtSno1.Text = ds.Tables[0].Rows[i][0].ToString();

txtCno1.Text = ds.Tables[0].Rows[i][1].ToString();

txtGradepeacetime1.Text = ds.Tables[0].Rows[i][2].ToString();

txtGradeexpriment1.Text = ds.Tables[0].Rows[i][3].ToString();

txtGradelast1.Text = ds.Tables[0].Rows[i][4].ToString();

txtGrade1.Text = ds.Tables[0].Rows[i][5].ToString();}

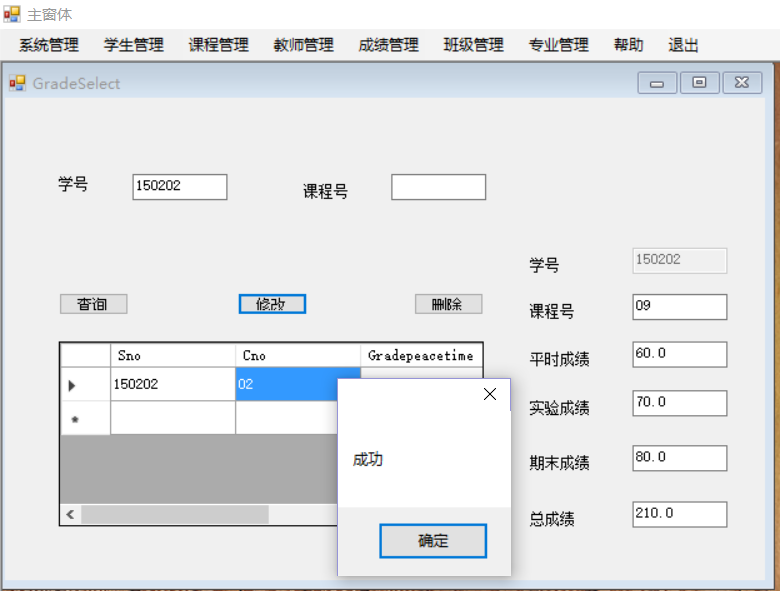
2.5.3 成绩信息的修改如图所示：

图2.5.2 成绩管理修改界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过teacherInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，响应到右边的文本框中，则可以进行修改，其中学号不能修改，因为lable控件的教师编号的enable属性设置为True。其中修改成绩信息的主要代码为:

private void button2\_Click(object sender, EventArgs e)

{

StuGradeData data = new StuGradeData();

data.Sno = txtSno1.Text;

data.Cno = txtCno1.Text;

data.Gradepeacetime = float.Parse(txtGradepeacetime1.Text);

data.Gradeexpriment = float.Parse(txtGradeexpriment1.Text);

data.Gradelast = float.Parse(txtGradelast1.Text);

data.Grade = float.Parse(txtGrade1.Text);

Classes.StuGradeOperation op = new Classes.StuGradeOperation();

if (op.UpdateStuGrade(data))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];}

2.5.4 成绩信息的删除如图所示：



图2.5.2 成绩管理删除界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过teacherInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，即可以删除该数据。其中删除成绩信息的主要代码为:

private void button3\_Click(object sender, EventArgs e)

{

string sno = txtSno1.Text;

Classes.StuGradeOperation op = new Classes.StuGradeOperation();

if (op.DeleteStuGrade(sno))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];

}

## 2.6 班级管理功能的实现

成绩管理是为了管理现有的成绩。在该模块中可以新增、修改、删除、查询成绩信息。添加成绩需添加成绩编号、教师名称、教师实验学时、教师讲科学时、教师学分、教师类型。查询的时候，输入相应的教师编号、教师实验学时，则会在datagridview界面中显示对应的教师。删除和修改教师信息的时候，如果删除或修改成功，则会提示删除成功或修改成功。实现界面如下图：

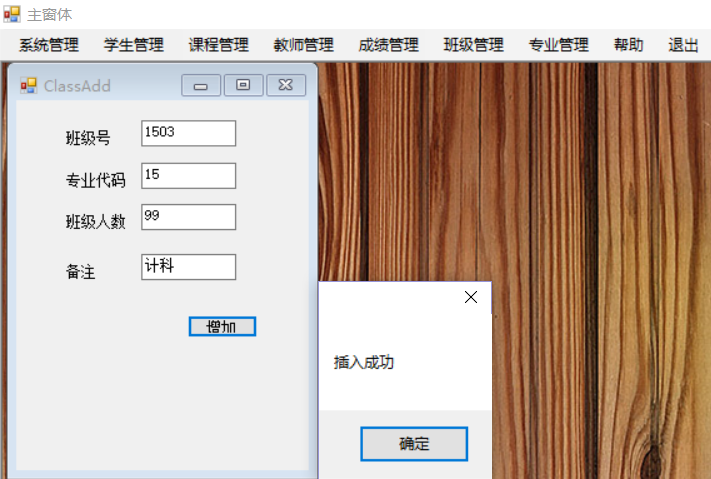
2.6.1 班级信息的增加如图所示：

图2.6.1 班级管理增加界面

实现过程：

该界面主要有button控件、lable控件、textbox控件。通过teacherInfoOperation连接数据库，将textbox的内容响应到数据库中。其中学号不能为空，若是空值，则无法添加。其中添加班级的主要代码为：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using 学生成绩管理系统.Classes;

namespace 学生成绩管理系统

{

public partial class ClassAdd : Form

{

public ClassAdd()

{

InitializeComponent();}

private void button1\_Click(object sender, EventArgs e)

{

ClassInfoData data = new ClassInfoData();

data.Classid = txtClassid.Text;

data.Specialtyid = txtSpecialtyid.Text;

data.Studentnumber = int.Parse(txtStudentnumber.Text);

data.Remark = txtRemark.Text;

if (txtClassid.Text == "" || txtClassid.Text == null)

{

MessageBox.Show("用户名不能为空");

txtClassid.Focus();

return;}

Classes.ClassInfoOperation op = new Classes.ClassInfoOperation();

if (op.InsertClassInfo(data))

MessageBox.Show("插入成功");

else

MessageBox.Show("失败");} }}

2.6.2 班级信息的查询如图所示：

图2.6.2 班级管理查询界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。lable控件的enable属性设置为True，通过classInfoOperation连接数据库，输入班级号、专业代码，点击查询，数据库中的数据就会响应到datagridview中。其中查询班级的主要代码为:

private void ClassSelect\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = Bind().Tables[0];

}

DataSet Bind()

{ClassInfoData data = new ClassInfoData();

data.Classid = txtClassid.Text;

data.Specialtyid = txtSpecialtyid.Text;

Classes.ClassInfoOperation op = new Classes.ClassInfoOperation();

ds = op.GetClassInfo(data);

return ds;

}

private void button1\_Click(object sender, EventArgs e)

{

dataGridView1.DataSource = Bind().Tables[0];

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int i = e.RowIndex;

txtClassid1.Text = ds.Tables[0].Rows[i][0].ToString();

txtSpecialtyid1.Text = ds.Tables[0].Rows[i][1].ToString();

txtStudentnumber1.Text = ds.Tables[0].Rows[i][2].ToString();

txtRemark1.Text = ds.Tables[0].Rows[i][3].ToString();}

2.6.3 班级信息的修改如图所示：



图2.6.2 班级管理修改界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过classInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，响应到右边的文本框中，则可以进行修改，其中学号不能修改，因为lable控件的教师编号的enable属性设置为True。其中修改班级信息的主要代码为:

private void button2\_Click(object sender, EventArgs e)

{

ClassInfoData data = new ClassInfoData();

data.Classid = txtClassid1.Text;

data.Specialtyid = txtSpecialtyid1.Text;

data.Studentnumber = int.Parse(txtStudentnumber1.Text);

data.Remark = txtRemark1.Text;

Classes.ClassInfoOperation op = new Classes.ClassInfoOperation();

if (op.UpdateClass(data))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];}

2.6.4 班级信息的删除如图所示：

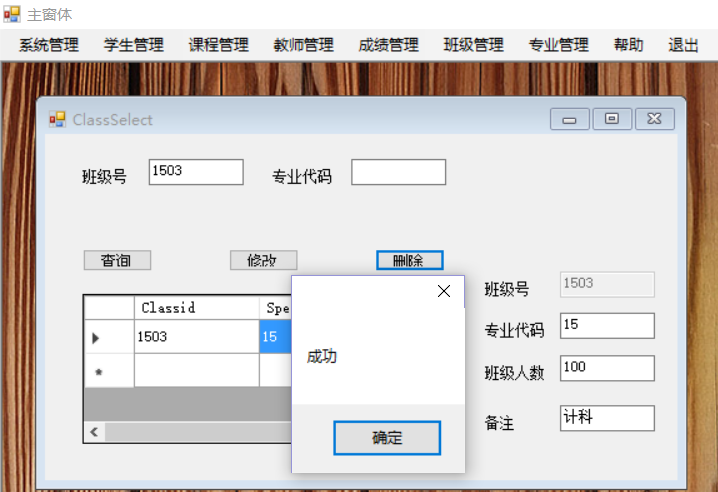


图2.6.2 班级管理删除界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过classInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，即可以删除该数据。其中删除班级信息的主要代码为:

private void button3\_Click(object sender, EventArgs e)

{

string classid = txtClassid1.Text;

Classes.ClassInfoOperation op = new Classes.ClassInfoOperation();

if (op.DeleteClass(classid))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];}

## 2.7 班级管理功能的实现

专业管理是为了管理现有的专业。在该模块中可以新增、修改、删除、查询专业信息。添加专业需添加专业代码、专业名称。查询的时候，输入相应的专业代码或专业名称，则会在datagridview界面中显示对应的教师。删除和修改专业信息的时候，如果删除或修改成功，则会提示删除成功或修改成功。实现界面如下图：

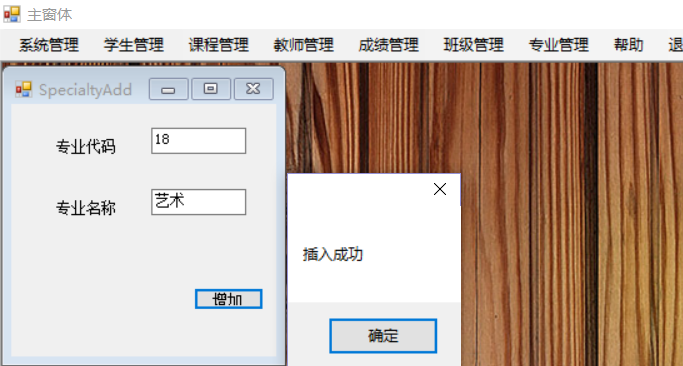
2.7.1 专业信息的增加如图所示：

图2.7.1 专业管理增加界面

实现过程：

该界面主要有button控件、lable控件、textbox控件。通过specialtyInfoOperation连接数据库，将textbox的内容响应到数据库中。其中学号不能为空，若是空值，则无法添加。其中添加专业的主要代码为：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using 学生成绩管理系统.Classes;

namespace 学生成绩管理系统

{

public partial class SpecialtyAdd : Form

{

public SpecialtyAdd(){

InitializeComponent();}

private void button1\_Click(object sender, EventArgs e)

{

SpecialtyInfoData data = new SpecialtyInfoData();

data.Specialtymc = txtSpecialtymc.Text;

data.Specialtyid = txtSpecialtyid.Text;

if (txtSpecialtyid.Text == "" || txtSpecialtyid.Text == null)

{

MessageBox.Show("用户名不能为空");

txtSpecialtyid.Focus();

return;}

Classes.SpecialtyInfoOperation op = new Classes.SpecialtyInfoOperation();

if (op.InsertSpecialtyInfo(data))

MessageBox.Show("插入成功");

else

MessageBox.Show("失败");} }}

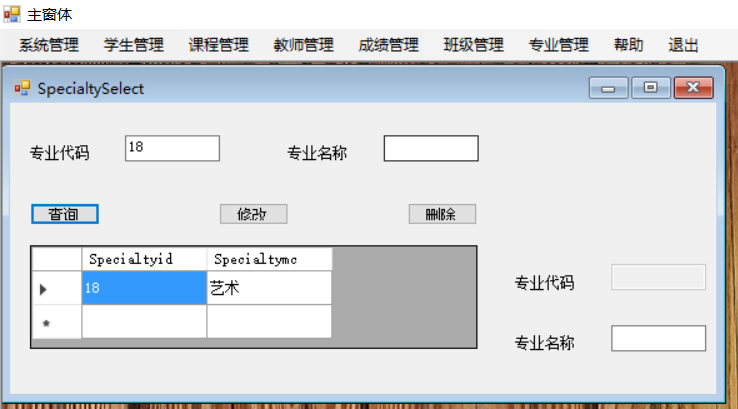
2.7.2 专业信息的查询如图所示：

图2.7.2 专业管理查询界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。lable控件的enable属性设置为True，通过specialtyInfoOperation连接数据库，输入班级号、专业代码，点击查询，数据库中的数据就会响应到datagridview中。其中查询专业的主要代码为:

private void SpecialtySelect\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = Bind().Tables[0];

}

DataSet Bind()

{

SpecialtyInfoData data = new SpecialtyInfoData();

data.Specialtyid = txtSpecialtyid.Text;

data.Specialtymc = txtSpecialtymc.Text;

Classes.SpecialtyInfoOperation op = new Classes.SpecialtyInfoOperation();

ds = op.GetSpecialtyInfo(data);

return ds;

}

private void button1\_Click(object sender, EventArgs e)

{

dataGridView1.DataSource = Bind().Tables[0];

}

//dataGridView显示数据

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

int i = e.RowIndex;

txtSpecialtyid1.Text = ds.Tables[0].Rows[i][0].ToString();

txtSpecialtymc1.Text = ds.Tables[0].Rows[i][1].ToString();

}

2.7.3 专业信息的修改如图所示：

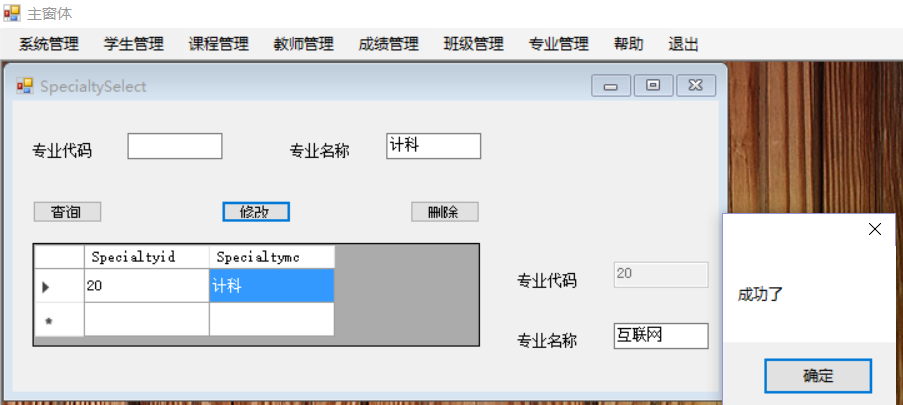


图2.7.2 专业管理修改界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过specialtyInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，响应到右边的文本框中，则可以进行修改，其中专业代码不能修改，因为lable控件的教师编号的enable属性设置为True。其中修改专业信息的主要代码为:

private void button2\_Click(object sender, EventArgs e)

{

SpecialtyInfoData data = new SpecialtyInfoData();

data.Specialtyid = txtSpecialtyid1.Text;

data.Specialtymc = txtSpecialtymc1.Text;

Classes.SpecialtyInfoOperation op = new Classes.SpecialtyInfoOperation();

if (op.UpdateSpecialty(data))

MessageBox.Show("成功了");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];

}

2.7.4 专业信息的删除如图所示：

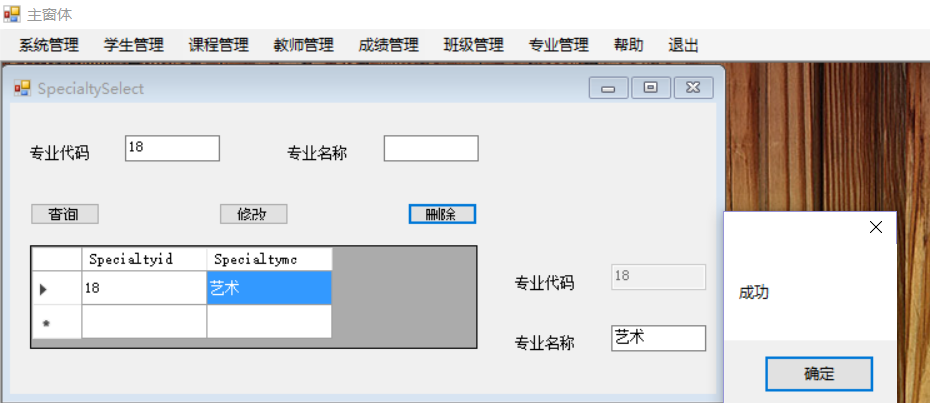


图2.7.2 专业管理删除界面

实现过程：

该界面主要有button控件、lable控件、textbox控件、datagridview控件。通过specialtyInfoOperation连接数据库，通过查询的到的数据会相应到datagridview中，单击某个数据，即可以删除该数据。其中删除专业信息的主要代码为:

private void button3\_Click(object sender, EventArgs e)

{

string specialtyid = txtSpecialtyid1.Text;

Classes.SpecialtyInfoOperation op = new Classes.SpecialtyInfoOperation();

if (op.DeleteSpecialty(specialtyid))

MessageBox.Show("成功");

else

MessageBox.Show("失败了");

dataGridView1.DataSource = Bind().Tables[0];

}