摘要：

# 摘 要

学生选课系统是一个学校不可缺少的部分,它的内容对于学校的决策者和管理者来说都至关重要,所以学生选课系统应该能够为用户提供充足的信息和快捷的查询手段。学生选课系统需要满足来自三方面的需求，这三个方面分别是学生、教师和管理员。本系统包含了这三个用户角色的重要功能，加强了用户之间的彼此联系。本系统围绕学生选课功能，规范了选课流程，增加了细节设置，使用方便，用户界面良好，使得教务管理更加方便。

关键字：C# ，选课管理，mysql

第一章:需求分析

第1章 需求分析

学生选课系统是一个学校不可缺少的部分,它的内容对于学校的决策者和管理者来说都至关重要,所以学生选课系统应该能够为用户提供充足的信息和快捷的查询手段。

学生选课系统需要满足来自三方面的需求，这三个方面分别是学生、教师和管理员。学生的需求是查询院系的课程、学生选课情况及学生信息的修改:教师对选课系统学生选课情况进行操作，同时形成学生选课查看确认;选课管理员的功能最为复杂，包括对学生、教师、选课进行管理和统计，及系统状态的查看、维护并生成选课报表。学生可以直接查看选课情况，学生可以根据本人学号和密码登录系统，还可以进行本人学科成绩情况的查询和维护部分个人信息。一般情况下，学生只应该查询和维护不入的选课情况和个人信息，若查询和维护其他学生的选课及成绩查询信息，就要知道其他坐生的学号和密码。这些是很难得到的，特别是密码。所以不但满足了学生的废求，还保护了学生的个人隐私。

学校管理员可以浏览、查询、修交、审核选课的信息结果；添加和修改学生、老师的信息，可以确保添加时数据库数据的独特性；对课程进行审核，在达到课程预定上限时限制添加；对必修课进行发布，添加到学生和老师的课表里。管理员与其他两个用户关系最为密切，要灵活使用各个数据库表，添加程序选课流程的限制（选课流程为：管理员添加课程，设置课程人数上限，发布必修课程，学生查询本专业限选课程以及学校任选课程，管理员对学生进行审核，达到人数上限则不能再审核，通过的学生课表增加该该门课，同时老师的学生名单中增加该学生），严格按照步骤进行，使用最为频繁。

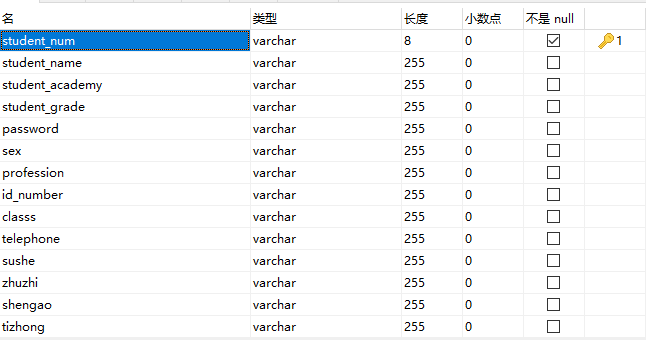
学校老师可以查看自己的课表，查看个人信息，录入所教授课程学生的成绩，按照课程编号查看学生名单以及修改自己的登陆密码。教师课程基本都是从数据库中读取数据，关键是要建立学生、老师、课程及选课结果几个关系表中的关联关系。

组员分工：

第2章 系统模块设计



第3章 数据库设计



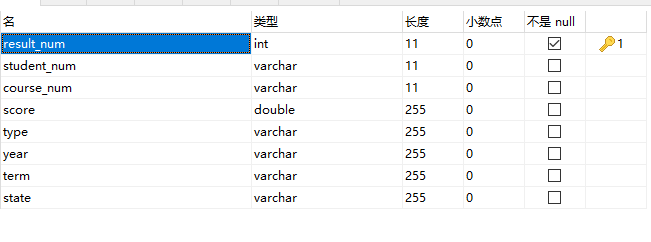
3-1学生信息表



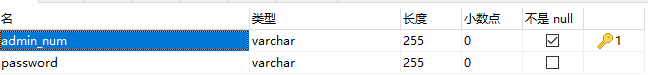
3-2教师信息表



3-3课程信息表



3-4选课结果表



3-5管理员信息表

第4章 系统实现

**学生名单导出：**

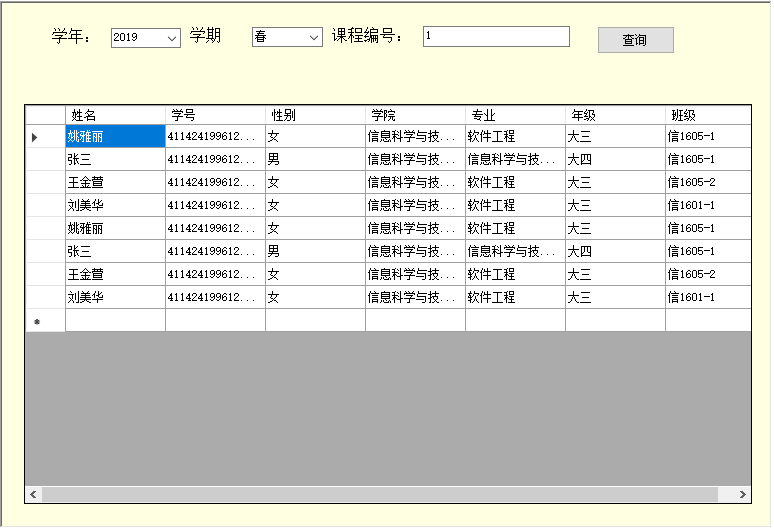


图4-1学生名单导出

**技术点及闪光点：**

需进行结果表，学生表的两表关联，通过DataGridView.add和List实现显示所有学生名单。

**核心代码：**

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

{

dao.resultDao dao = new dao.resultDao();

dao.studentDao sdao = new dao.studentDao();

List<model.Result> rs = new List<model.Result>();

rs = dao.selectStudent(textBox1.Text, comboBox1.Text, comboBox2.Text);//学生编号s

List<model.Student> s = new List<model.Student>();

s = sdao.selectStudent(rs);//根据学生编号s查询出所有的学生信息

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("姓名", typeof(string));

dt.Columns.Add("学号", typeof(string));

dt.Columns.Add("性别", typeof(String));

dt.Columns.Add("学院", typeof(string));

dt.Columns.Add("专业", typeof(String));

dt.Columns.Add("年级", typeof(String));

dt.Columns.Add("班级", typeof(String));

foreach (model.Student a in s)

{

dt.Rows.Add(a.Student\_name, a.Id\_number, a.Sex, a.Student\_academy, a.Profession, a.Student\_grade, a.Classs);

}

dataGridView1.DataSource = dt;

}

**教师添加：**



图4-2教师添加

**核心代码：**

bool wanzheng = true; ;//用户是否输入完整

int i = 添加学生框.RowCount;

// MessageBox.Show("" + i);

dao.teacherDao tdao = new teacherDao();

for (int j = 0; j < i - 1; j++)//检查时候有未填的空

{

string Teacher\_num = Convert.ToString(添加学生框.Rows[j].Cells[0].Value);//学号

string Teacher\_name = Convert.ToString(添加学生框.Rows[j].Cells[1].Value);//学号

string Sex = Convert.ToString(添加学生框.Rows[j].Cells[2].Value);//性别

string Teacher\_academy = Convert.ToString(添加学生框.Rows[j].Cells[3].Value);//学院

string Profession = Convert.ToString(添加学生框.Rows[j].Cells[4].Value);//专业

string Id\_number = Convert.ToString(添加学生框.Rows[j].Cells[5].Value);//身份证号

if (Id\_number.Length != 18)

{

MessageBox.Show("请输入正确的" + Teacher\_name + "的身份证号码");

wanzheng = false;

}

else if (Teacher\_num == "" || Teacher\_name == "" || Sex == "" || Teacher\_academy == "" || Profession == "" || Id\_number == "")

{

MessageBox.Show("请输入完整信息！");

wanzheng = false;

}

}

if (wanzheng == true)

{

for (int j = 0; j < i-1; j++)

{

model.Teacher teacher = new Teacher();

teacher.Teacher\_num = Convert.ToString(添加学生框.Rows[j].Cells[0].Value);//学号

teacher.Teacher\_name = Convert.ToString(添加学生框.Rows[j].Cells[1].Value);//学号

teacher.Sex = Convert.ToString(添加学生框.Rows[j].Cells[2].Value);//性别

teacher.Teacher\_academy = Convert.ToString(添加学生框.Rows[j].Cells[3].Value);//学院

teacher.Profession = Convert.ToString(添加学生框.Rows[j].Cells[4].Value);//专业

teacher.Id\_number = Convert.ToString(添加学生框.Rows[j].Cells[5].Value);//身份证号

string id\_number = teacher.Id\_number;

teacher.Password = id\_number.Substring(id\_number.Length - 6, 6);

bool isd = dao.teacherDao.add\_teacher(teacher);

if (isd == false)

{

MessageBox.Show("提交失败！");

break;

}

}

MessageBox.Show("提交成功！");

添加学生框.Rows.Clear();

}

**教师查看及修改：**

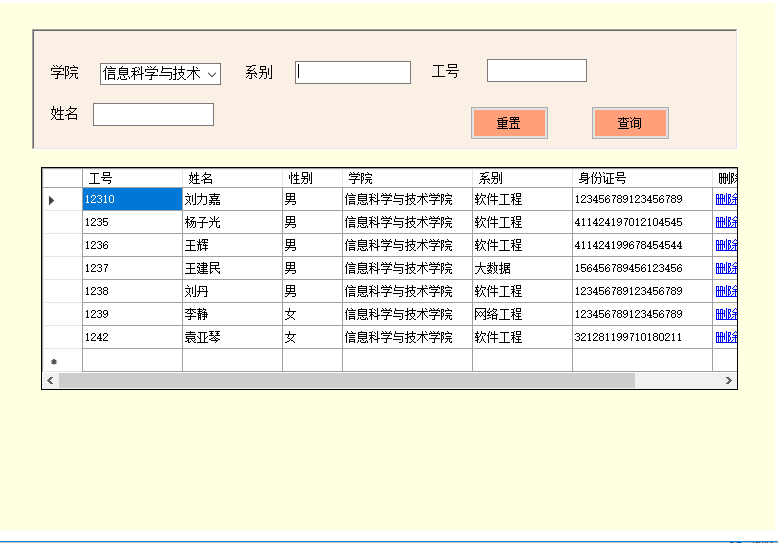


图4-3教师信息查看

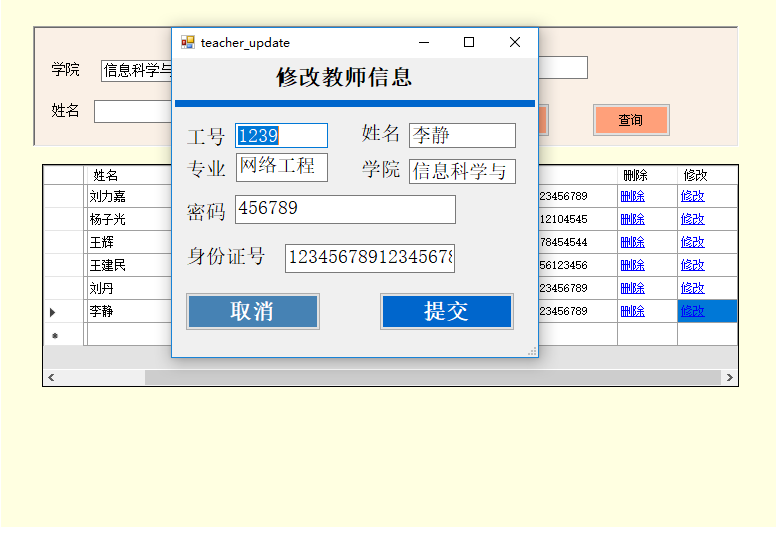


图4-4教师信息修改

**核心代码：**

private void 教师查询按钮\_Click(object sender, EventArgs e)

{

Teacher ss = new Teacher();

ss.Teacher\_num = 工号1.Text;

ss.Teacher\_name = 姓名1.Text;

ss.Teacher\_academy = 学院1.Text;

ss.Profession = 专业1.Text;

List<Teacher> teachers = new List<Teacher>();

if (ss.Teacher\_num == "" && ss.Teacher\_name == "" && ss.Teacher\_academy == "" && ss.Profession == "")//如果输入为空，则显示全部学生信息

{

teachers = dao.teacherDao.find\_teachers();

}

else//如果输入查询条件则更具条件查询

{

teachers = dao.teacherDao.find\_teachers2(ss);

}

教师显示.Rows.Clear();

int i = 0;

if (teachers != null)

{

foreach (Teacher s in teachers)

{

教师显示.Rows.Add();

教师显示.Rows[i].Cells[0].Value = s.Teacher\_num;

教师显示.Rows[i].Cells[1].Value = s.Teacher\_name;

教师显示.Rows[i].Cells[2].Value = s.Sex;

教师显示.Rows[i].Cells[3].Value = s.Teacher\_academy;

教师显示.Rows[i].Cells[4].Value = s.Profession;

教师显示.Rows[i].Cells[5].Value = s.Id\_number;

i++;

}

}

//清空原来输入的查询条件

学院1.Text = "";

专业1.Text = "";

姓名1.Text = "";

工号1.Text = "";

}

private void 查看ToolStripMenuItem\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel4.Visible = false;

panel2.Visible = false;

panel5.Visible = false;

panel添加教师页面.Visible = false;

panel9.Visible = false;

panel审核.Visible = false;

查看教师页面.Visible = true;

教师显示.Rows.Clear();

List<Teacher> teachers = new List<Teacher>();

teachers = dao.teacherDao.find\_teachers();

int i = 0;

if (teachers != null)

{

foreach (Teacher s in teachers)

{

教师显示.Rows.Add();

教师显示.Rows[i].Cells[0].Value = s.Teacher\_num;

教师显示.Rows[i].Cells[1].Value = s.Teacher\_name;

教师显示.Rows[i].Cells[2].Value = s.Sex;

教师显示.Rows[i].Cells[3].Value = s.Teacher\_academy;

教师显示.Rows[i].Cells[4].Value = s.Profession;

教师显示.Rows[i].Cells[5].Value = s.Id\_number;

i++;

}

}

}

private void 教师显示\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

if (e.ColumnIndex == 6)//删除

{

int i = e.RowIndex;

string teacher\_num = Convert.ToString(教师显示.Rows[i].Cells[0].Value);

MessageBox.Show("删除成功！" );

teacherDao.delete(teacher\_num);

教师显示.Rows.Remove(教师显示.Rows[i]);

}

else if (e.ColumnIndex == 7)//查询

{

int i = e.RowIndex;

Teacher s = new Teacher();

string teacher\_num = Convert.ToString(教师显示.Rows[i].Cells[0].Value);

Form f1 = new teacher\_update(teacher\_num);

f1.ShowDialog();

Teacher teacher = new Teacher();

teacherDao.update(teacher);

if (tea != null)//用户修改了信息

{

教师显示.Rows[i].Cells[0].Value = tea.Teacher\_num;

教师显示.Rows[i].Cells[1].Value = tea.Teacher\_name;

// 教师显示.Rows[i].Cells[2].Value = tea.Sex;//性别不能修改

教师显示.Rows[i].Cells[3].Value = tea.Teacher\_academy;

教师显示.Rows[i].Cells[4].Value = tea.Profession;

教师显示.Rows[i].Cells[5].Value = tea.Id\_number;

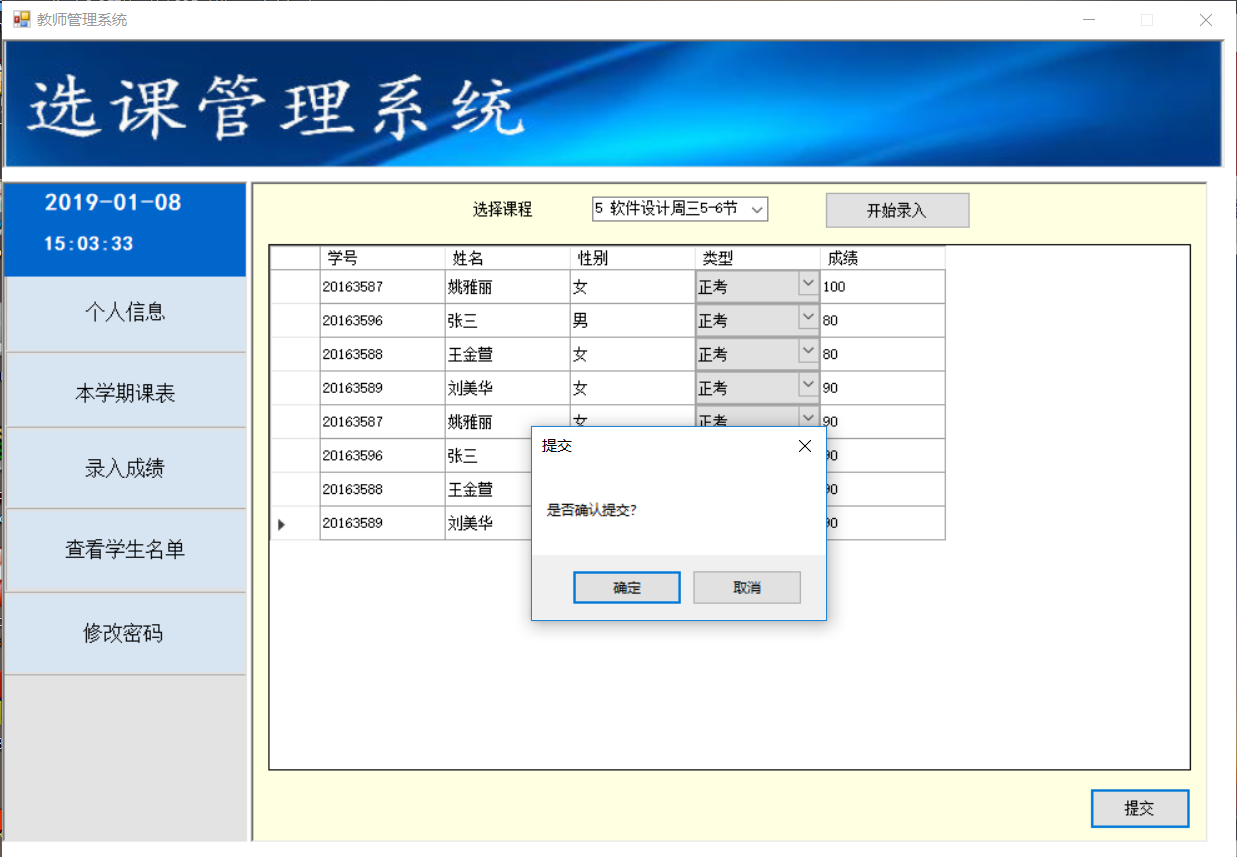
}

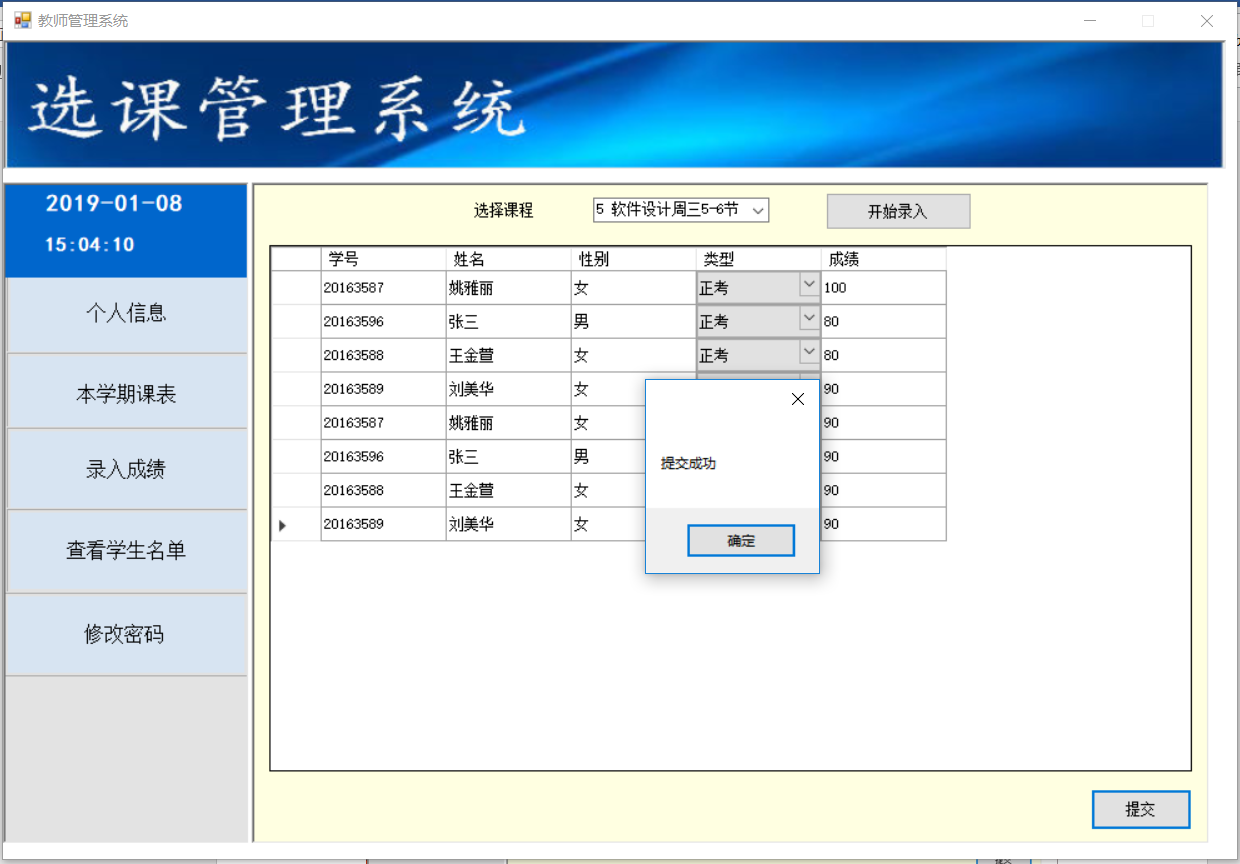
}

}

**教师录入成绩：**







**核心代码：**

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

{

string course = comboBox1成绩.Text;

if (course == "")

{

MessageBox.Show("请选择需要录入的课程！");

}

else

{

dataGridView1成绩.Rows.Clear();

//coursedao cdao = new coursedao();

string[] c = course.Split(' ');

string cour = c[0];

dao.resultDao rdao = new dao.resultDao();

List<model.Result> rs = new List<model.Result>();

rs = rdao.selectResult(cour);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dao.studentDao sdao = new dao.studentDao();

Console.WriteLine("姚雅丽");

dt.Columns.Add("学号", typeof(string));

dt.Columns.Add("姓名", typeof(string));

dt.Columns.Add("性别", typeof(string));

dt.Columns.Add("类型", typeof(string));

dt.Columns.Add("成绩", typeof(string));

foreach (DataGridViewColumn co in dataGridView1成绩.Columns)

{

if (co.Index != 4 && co.Index != 3)

co.ReadOnly = true;

}

foreach (model.Result r in rs)

{

model.Student s = new model.Student();

s = sdao.selectNum(r.Student\_num);

dataGridView1成绩.Rows.Add(r.Student\_num, s.Student\_name, s.Sex, "", "");

}

}

}

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

{

MessageBoxButtons messButton = MessageBoxButtons.OKCancel;

DialogResult dr = MessageBox.Show("是否确认提交？", "提交", messButton);

if (dr == DialogResult.OK)

{

string course = comboBox1成绩.Text;

dao.resultDao rdao = new dao.resultDao();

string[] c = course.Split(' ');

string cour = c[0];

bool b = true;

for (int i = 0; i < dataGridView1成绩.Rows.Count ; i++)

{

try

{

rdao.updateScore(dataGridView1成绩.Rows[i].Cells[0].Value.ToString(), cour, dataGridView1成绩.Rows[i].Cells[3].Value.ToString(), Convert.ToDouble(dataGridView1成绩.Rows[i].Cells[4].Value));

}

catch (SystemException)

{

MessageBox.Show("格式错误！");

b = false;

}

}

if (b == true)

{

MessageBox.Show("提交成功");

}

}

else

{

}

}

**学生查看成绩：**



**核心代码：**

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

{

dao.resultDao dao = new dao.resultDao();

dao.courseDao cdao = new dao.courseDao();

dao.teacherDao tdao = new dao.teacherDao();

List<model.Result> rs = new List<model.Result>();

if (checkBox2成绩.Checked == false && checkBox1成绩.Checked == false)

{

rs = dao.selectAll(student\_login.number, comboBox1成绩.Text, comboBox2成绩.Text);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("代码", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("性质", typeof(String));

dt.Columns.Add("教师", typeof(string));

dt.Columns.Add("学分", typeof(String));

dt.Columns.Add("成绩", typeof(String));

dt.Columns.Add("类型", typeof(String));

dt.Columns.Add("学年", typeof(String));

dt.Columns.Add("学期", typeof(String));

foreach (model.Result r1 in rs)

{

if (r1.Score != double.NaN)

{

model.Course c = new model.Course();

c = cdao.selectNum(r1.Course\_num);

model.Teacher t = new model.Teacher();

t = tdao.selectNum(c.Teacher\_num);

dt.Rows.Add(r1.Result\_num, c.Course\_name, c.Type, t.Teacher\_name, c.Credit\_hour, r1.Score, r1.Type, r1.Year, r1.Term);

}

}

dataGridView1成绩.DataSource = dt;

}

else if (checkBox1成绩.Checked == true && checkBox2成绩.Checked == false)

{

rs = dao.select1(student\_login.number, comboBox1成绩.Text, comboBox2成绩.Text);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("代码", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("性质", typeof(String));

dt.Columns.Add("教师", typeof(string));

dt.Columns.Add("学分", typeof(String));

dt.Columns.Add("成绩", typeof(String));

dt.Columns.Add("类型", typeof(String));

dt.Columns.Add("学年", typeof(String));

dt.Columns.Add("学期", typeof(String));

foreach (model.Result r1 in rs)

{

if (r1.Score != double.NaN)

{

model.Course c = new model.Course();

c = cdao.selectNum(r1.Course\_num);

model.Teacher t = new model.Teacher();

t = tdao.selectNum(c.Teacher\_num);

dt.Rows.Add(r1.Result\_num, c.Course\_name, c.Type, t.Teacher\_name, c.Credit\_hour, r1.Score, r1.Type, r1.Year, r1.Term);

}

}

dataGridView1成绩.DataSource = dt;

}

else if (checkBox1成绩.Checked == false && checkBox2成绩.Checked == true)

{

rs = dao.select2(student\_login.number, comboBox1成绩.Text, comboBox2成绩.Text);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("代码", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("性质", typeof(String));

dt.Columns.Add("教师", typeof(string));

dt.Columns.Add("学分", typeof(String));

dt.Columns.Add("成绩", typeof(String));

dt.Columns.Add("类型", typeof(String));

dt.Columns.Add("学年", typeof(String));

dt.Columns.Add("学期", typeof(String));

foreach (model.Result r1 in rs)

{

if (r1.Score != double.NaN)

{

model.Course c = new model.Course();

c = cdao.selectNum(r1.Course\_num);

model.Teacher t = new model.Teacher();

t = tdao.selectNum(c.Teacher\_num);

dt.Rows.Add(r1.Result\_num, c.Course\_name, c.Type, t.Teacher\_name, c.Credit\_hour, r1.Score, r1.Type, r1.Year, r1.Term);

}

}

dataGridView1成绩.DataSource = dt;

}

else if (checkBox1成绩.Checked == true && checkBox2成绩.Checked == true)

{

rs = dao.select3(student\_login.number, comboBox1成绩.Text, comboBox2成绩.Text);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("代码", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("性质", typeof(String));

dt.Columns.Add("教师", typeof(string));

dt.Columns.Add("学分", typeof(String));

dt.Columns.Add("成绩", typeof(String));

dt.Columns.Add("类型", typeof(String));

dt.Columns.Add("学年", typeof(String));

dt.Columns.Add("学期", typeof(String));

foreach (model.Result r1 in rs)

{

if (r1.Score != double.NaN)

{

model.Course c = new model.Course();

c = cdao.selectNum(r1.Course\_num);

model.Teacher t = new model.Teacher();

t = tdao.selectNum(c.Teacher\_num);

dt.Rows.Add(r1.Result\_num, c.Course\_name, c.Type, t.Teacher\_name, c.Credit\_hour, r1.Score, r1.Type, r1.Year, r1.Term);

}

}

dataGridView1成绩.DataSource = dt;

}

}

**查询所有可选课程:**



图4-5查询所有可选课程

**技术点及闪光点：**

通过课程表中的课程编号找到教师编号，再通过教师编号查询教师表中的教师姓名；限选仅显示学生所属学院的课程；从表中获取当前课程编号，再从选课表中计算该课程编号的被选取次数，得到当前选课人数。

**核心代码：**

private void toolStripLabel2\_Click\_1(object sender, EventArgs e)

{

string str = util.Util.sqlcon;

MySqlConnection conn = new MySqlConnection(str);

s string sql = "select course\_num,course\_name,(select teacher\_name from teacher where teacher\_num=course.teacher\_num) as teacher\_name,credit\_hour,type,address,time\_week,time\_jie,academy,maxcount,(select count(\*) from result where course\_num=course.Course\_num) as number from course where type='任选' or (type='限选' and academy='" + label2.Text + "')";

conn.Open();

MySqlCommand cmd = new MySqlCommand(sql, conn);

MySqlDataAdapter da = new MySqlDataAdapter(cmd);

DataSet ds = new DataSet();

da.Fill(ds, "course");

dataGridView1.DataSource = ds;

dataGridView1.DataMember = "course";

conn.Close();

}

**精确查询可选课程:**



图4-6精确查询可选课程

**技术点及闪光点：**

可通过课程名称、课程类型、学院来精确查询可选课程；限选只显示学生所属学院的课程。

**核心代码：**

private void toolStripLabel1\_Click(object sender, EventArgs e)

{

string str = util.Util.sqlcon;

String sql = "";

string tmp1 = toolStripComboBox1.Text;

string tmp2 = toolStripTextBox1.Text;

MySqlConnection conn = new MySqlConnection(str);

if (tmp1 == "课程名称")

sql = "select course\_num,course\_name,(select teacher\_name from teacher where teacher\_num=course.teacher\_num) as teacher\_name ,credit\_hour,type,address,time\_week,time\_jie,academy,maxcount,(select count(\*) from result where course\_num=course.Course\_num) as number from course where course\_name='" + tmp2 + "' and (type='任选' or (type='限选' and academy='" + label2.Text + "') )";

else if (tmp1 == "课程类型")

{

if (tmp2 == "限选")

sql = "select course\_num,course\_name,(select teacher\_name from teacher where teacher\_num=course.teacher\_num) as teacher\_name ,credit\_hour,type,address,time\_week,time\_jie,academy,maxcount,(select count(\*) from result where course\_num=course.Course\_num) as number from course where type='" + tmp2 + "'and academy='" + label2.Text + "'";

else

sql = "select course\_num,course\_name,(select teacher\_name from teacher where teacher\_num=course.teacher\_num) as teacher\_name ,credit\_hour,type,address,time\_week,time\_jie,academy,maxcount,(select count(\*) from result where course\_num=course.Course\_num) as number from course where type='任选'";

}

else if (tmp1 == "学院")

{

if (tmp2 == label2.Text)

sql = "select course\_num,course\_name,(select teacher\_name from teacher where teacher\_num=course.teacher\_num) as teacher\_name ,credit\_hour,type,address,time\_week,time\_jie,academy,maxcount,(select count(\*) from result where course\_num=course.Course\_num) as number from course where academy='" + tmp2 + "' and type!='必修'";

else

sql = "select course\_num,course\_name,(select teacher\_name from teacher where teacher\_num=course.teacher\_num) as teacher\_name ,credit\_hour,type,address,time\_week,time\_jie,academy,maxcount,(select count(\*) from result where course\_num=course.Course\_num) as number from course where academy='" + tmp2 + "' and type='任选' ";

}

else

{

sql = "select course\_num,course\_name,(select teacher\_name from teacher where teacher\_num=course.teacher\_num) as teacher\_name,credit\_hour,type,address,time\_week,time\_jie,academy,maxcount,(select count(\*) from result where course\_num=course.Course\_num) as number from course where type='任选' or (type='限选' and academy='" + label2.Text + "')";

MessageBox.Show("输入有误");

}

conn.Open();

MySqlCommand cmd = new MySqlCommand(sql, conn);

MySqlDataAdapter da = new MySqlDataAdapter(cmd);

DataSet ds = new DataSet();

da.Fill(ds, "course");

dataGridView1.DataSource = ds;

dataGridView1.DataMember = "course";

conn.Close();

}

**选课:**



图4-7选课1



图4-8选课2

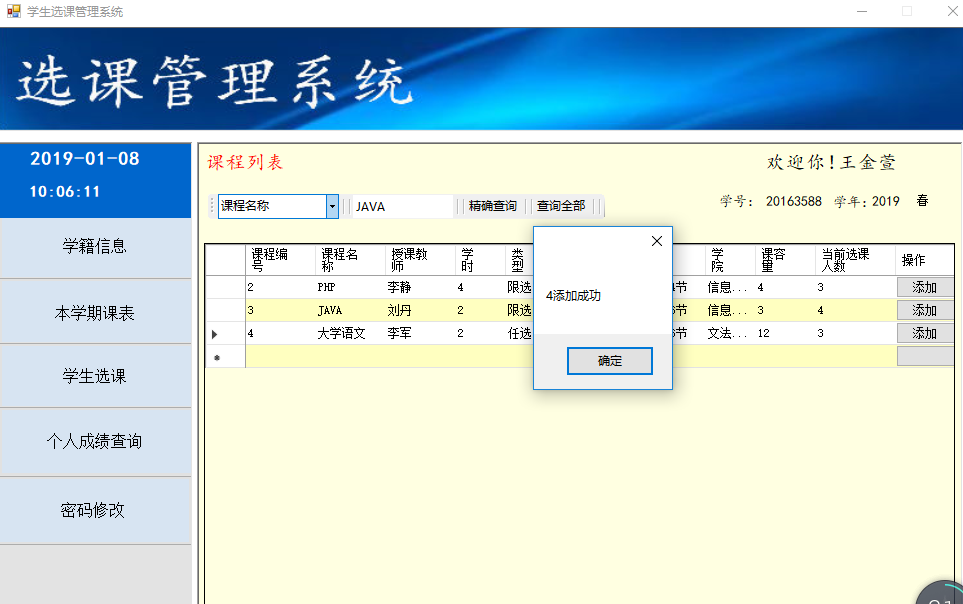


图4-9选课3

**技术点及闪光点：**

学生点击“添加”按钮，系统再次确定其选课意向，点击确定后选课成功。若已选择过该课程，则系统给出“已选择过该课程”的提示。添加成功后的结果会插入到选课表，此时课程状态默认为“待审核”。

**核心代码：**

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

{

if (e.RowIndex >= 0)

{

DataGridViewColumn column = dataGridView1.Columns[e.ColumnIndex];

if (column is DataGridViewButtonColumn)

{

string str = util.Util.sqlcon;

MySqlConnection conn = new MySqlConnection(str);

int r = this.dataGridView1.CurrentRow.Index;

string cNum = this.dataGridView1.Rows[r].Cells[1].Value.ToString();

conn.Open();

MySqlCommand mycmd = new MySqlCommand("select \* from result where student\_num='" + student\_login.number + "' and course\_num='" + cNum + "' ", conn);

MySqlDataReader mysdr = mycmd.ExecuteReader();

if (mysdr.HasRows)

{

MessageBox.Show("已添加过课程" + cNum);

conn.Close();

}

else

{

conn.Close();

DialogResult result = MessageBox.Show("确定添加课程" + cNum + "吗?", "添加", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

string sql = "insert into result(student\_num,course\_num,year,term,state) values('" + student\_login.number + "','" + cNum + "','" + label8.Text + "','春','待审核')";

conn.Open();

MySqlCommand cmd = new MySqlCommand(sql, conn);

MySqlDataAdapter da = new MySqlDataAdapter(cmd);

DataSet ds = new DataSet();

da.Fill(ds, "course");

conn.Close();

MessageBox.Show(cNum + "添加成功");

}

else { return; }

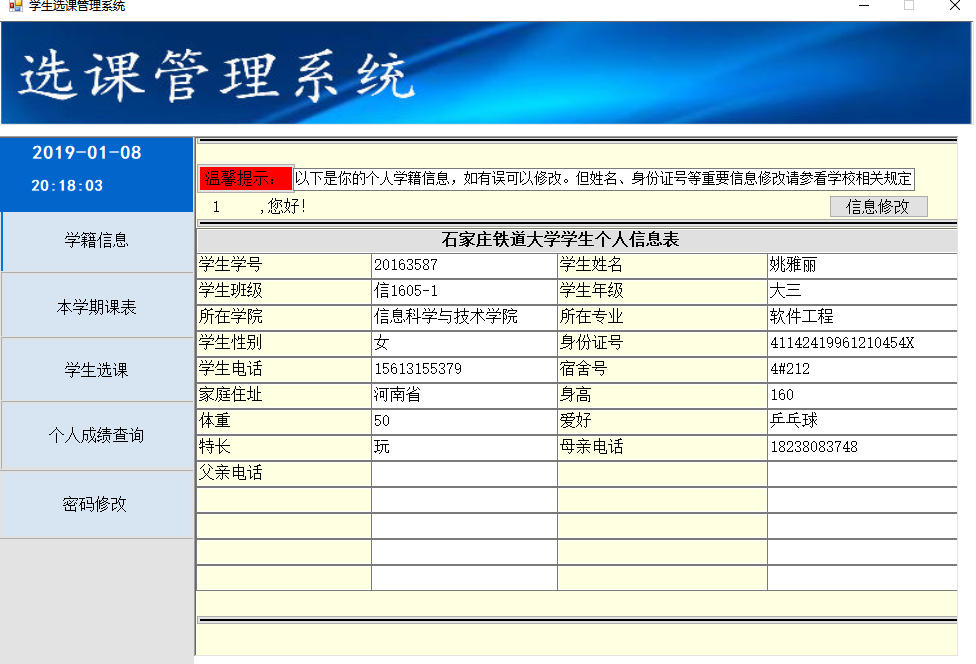
}

}

}

}

**学生个人学籍信息查看**



**核心代码：**

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

try

{

command = conn.CreateCommand();

command.CommandText = "select teacher\_num,teacher\_name,teacher\_academy,profession,sex,office,id\_number,zhuzhi,aihao,telephone from teacher where teacher\_num=" + teacher\_login.tnumber;

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

textBox2.Text = dataReader.GetString(0);

textBox5.Text = dataReader.GetString(1);

textBox8.Text = dataReader.GetString(2);

textBox10.Text = dataReader.GetString(3);

textBox12.Text = dataReader.GetString(4);

textBox14.Text = dataReader.GetString(5);

textBox16.Text = dataReader.GetString(6);

textBox18.Text = dataReader.GetString(7);

textBox20.Text = dataReader.GetString(8);

textBox22.Text = dataReader.GetString(9);

label2.Text = dataReader.GetString(1);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

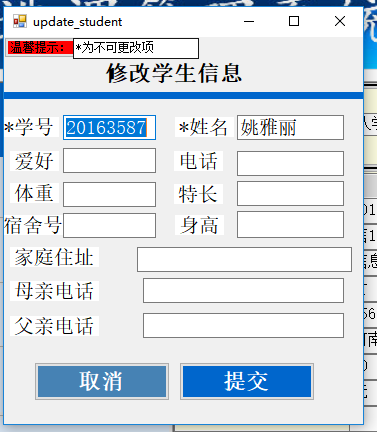
{

conn.Close();

}

}

**个人学籍信息修改**



**核心代码：**

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

{

MySqlConnection conn = Util.getConn();

MySqlCommand command = null;

try

{

if (text16.Text == "")

{

MessageBox.Show("办公室不能为空！");

}

else if (text17.Text == "")

{

MessageBox.Show("电话不能为空！");

}

else if (textBox1.Text == "")

{

MessageBox.Show("爱好不能为空！");

}

else if (text18.Text == "")

{

MessageBox.Show("家庭住址不能为空！");

}

else

{

command = conn.CreateCommand();

string sql = "update teacher set aihao='" + textBox1.Text + "',telephone='" +

text17.Text + "',office='" + text16.Text + "',zhuzhi='" +

text18.Text + "' where teacher\_num='" +

teacher\_login.tnumber + "'";

Console.Write("sql:" + sql);

command.CommandText = sql;

command.ExecuteNonQuery();

Console.WriteLine();

MessageBox.Show("已修改，请刷新页面！");

this.Close();

}

}

catch (Exception)

{

}

finally

{

if (conn.State == ConnectionState.Open)

{

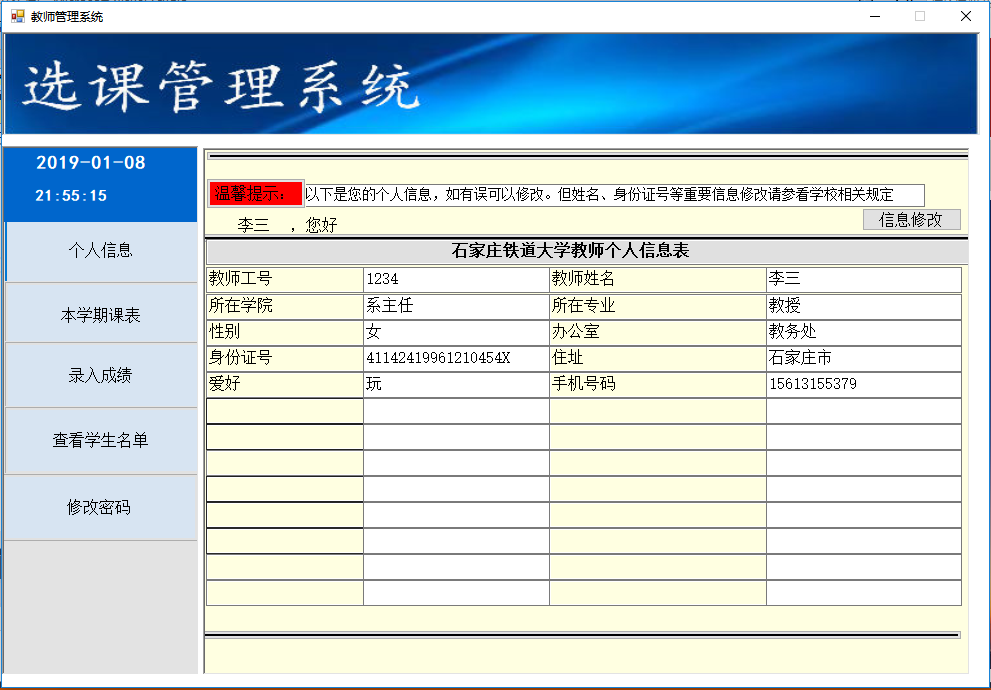
conn.Close();

}

}

}

**教师信息查看：**



**核心代码：**

private void teacher\_information\_Load(object sender, EventArgs e)

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

try

{

command = conn.CreateCommand();

command.CommandText = "select teacher\_num,teacher\_name,teacher\_academy,profession,sex,office,id\_number,zhuzhi,aihao,telephone from teacher where teacher\_num=" + teacher\_login.tnumber;

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

textBox2.Text = dataReader.GetString(0);

textBox5.Text = dataReader.GetString(1);

textBox8.Text = dataReader.GetString(2);

textBox10.Text = dataReader.GetString(3);

textBox12.Text = dataReader.GetString(4);

textBox14.Text = dataReader.GetString(5);

textBox16.Text = dataReader.GetString(6);

textBox18.Text = dataReader.GetString(7);

textBox20.Text = dataReader.GetString(8);

textBox22.Text = dataReader.GetString(9);

label2.Text = dataReader.GetString(1);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

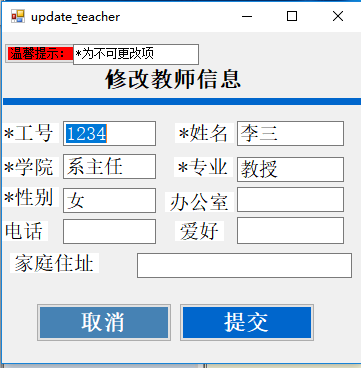
conn.Close();

}

}

}

**教师信息修改：**



**核心代码：**

private void update\_teacher\_Load(object sender, EventArgs e)

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

try

{

command = conn.CreateCommand();

command.CommandText = "select teacher\_num,teacher\_name,teacher\_academy,profession,sex from teacher where teacher\_num=" + teacher\_login.tnumber;

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

text11.Text = dataReader.GetString(0);

text12.Text = dataReader.GetString(1);

text13.Text = dataReader.GetString(2);

text14.Text = dataReader.GetString(3);

text15.Text = dataReader.GetString(4);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

}

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

{

this.Close();

}

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

{

MySqlConnection conn = Util.getConn();

MySqlCommand command = null;

try

{

if (text16.Text == "")

{

MessageBox.Show("办公室不能为空！");

}

else if (text17.Text == "")

{

MessageBox.Show("电话不能为空！");

}

else if (textBox1.Text == "")

{

MessageBox.Show("爱好不能为空！");

}

else if (text18.Text == "")

{

MessageBox.Show("家庭住址不能为空！");

}

else

{

command = conn.CreateCommand();

string sql = "update teacher set aihao='" + textBox1.Text + "',telephone='" +

text17.Text + "',office='" + text16.Text + "',zhuzhi='" +

text18.Text + "' where teacher\_num='" +

teacher\_login.tnumber + "'";

Console.Write("sql:" + sql);

command.CommandText = sql;

command.ExecuteNonQuery();

Console.WriteLine();

MessageBox.Show("已修改，请刷新页面！");

this.Close();

}

}

catch (Exception)

{

}

finally

{

if (conn.State == ConnectionState.Open)

{

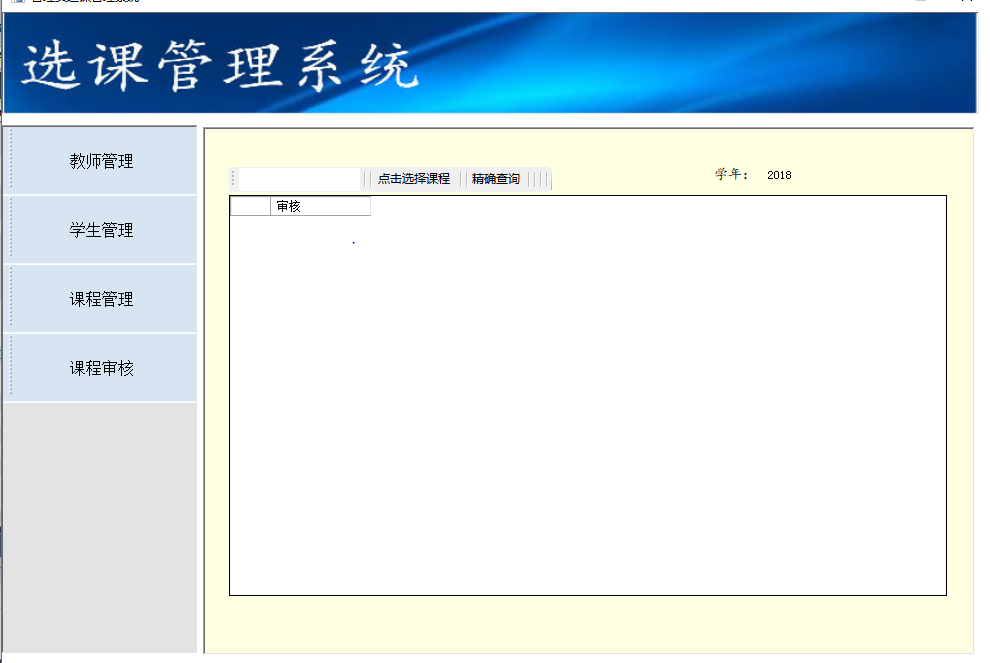
conn.Close();

}

}

}

**课程审核：**



**技术点及闪光点：**

课程审核，根据选择的课程名进行审核，在课程里查询出该课程编码，在result表里根据编码查出每个选择了该课程的数据，显示到data表里面。

**核心代码：**

private void toolStripLabel1\_Click(object sender, EventArgs e)

{

Console.Write("!!!!!!!!!!!!!!!!!!!!!!!!!");

string tmp1 = toolStripComboBox1.Text;

dao.shenheDao s = new shenheDao();

model.Course c = new Course();

c = s.find\_course(tmp1);

Console.Write("lalalal"+c.Course\_num);

List<model.Result> l = new List<model.Result>();

num= s.selectAll2(c.Course\_num);

Console.Write(num + "555555555555555555555");

List<model.Result> rs = new List<model.Result>();

rs = s.selectAll1(c.Course\_num);

Console.WriteLine(c.Course\_num+"33333333333333333333333333");

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("审核编码", typeof(string));

dt.Columns.Add("课程编号", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("学生学号", typeof(string));

dt.Columns.Add("学生姓名", typeof(string));

dt.Columns.Add("最大人数", typeof(int));

dt.Columns.Add("状态", typeof(string));

foreach (model.Result r1 in rs)

{

model.Student d = new model.Student();

Console.Write(r1.Student\_num + "111111111111111111");

d=s.find\_student(r1.Student\_num);

Console.Write("1111111111111111111");

dt.Rows.Add(r1.Result\_num,r1.Course\_num,tmp1, d.Student\_num, d.Student\_name,c.Maxcount,r1.State);

}

dataGridView1.DataSource = dt;

}

**点击选择课程：**





**技术点及闪光点：**

管理员点击查询课程按钮，跳出页面，查询所有已发布 的课程。

**核心代码：**

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

{

shenheDao s = new shenheDao();

List<Course> ss = new List<Course>();

ss = s.find\_courses();

Console.Write(ss);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("课程号", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("课程类型", typeof(string));

dt.Columns.Add("所属学院", typeof(string));

int i = 0;

foreach (Course s1 in ss)

{

dt.Rows.Add(s1.Course\_num, s1.Course\_name, s1.Type, s1.Academy);

i++;

}

dataGridView1.DataSource = dt;

foreach (DataGridViewColumn co in dataGridView1.Columns)

{

if (co.Index != 0)

co.ReadOnly = true;

}

data = dataGridView1;

}

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

{

int r = this.dataGridView1.CurrentRow.Index;

SH.toolStripComboBox1.Text = data.Rows[r].Cells[2].Value.ToString();

this.Close();

}

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

{

shenheDao s = new shenheDao();

Course c = new Course();

c = s.find\_course(textBox1.Text);

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("课程号", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("课程类型", typeof(string));

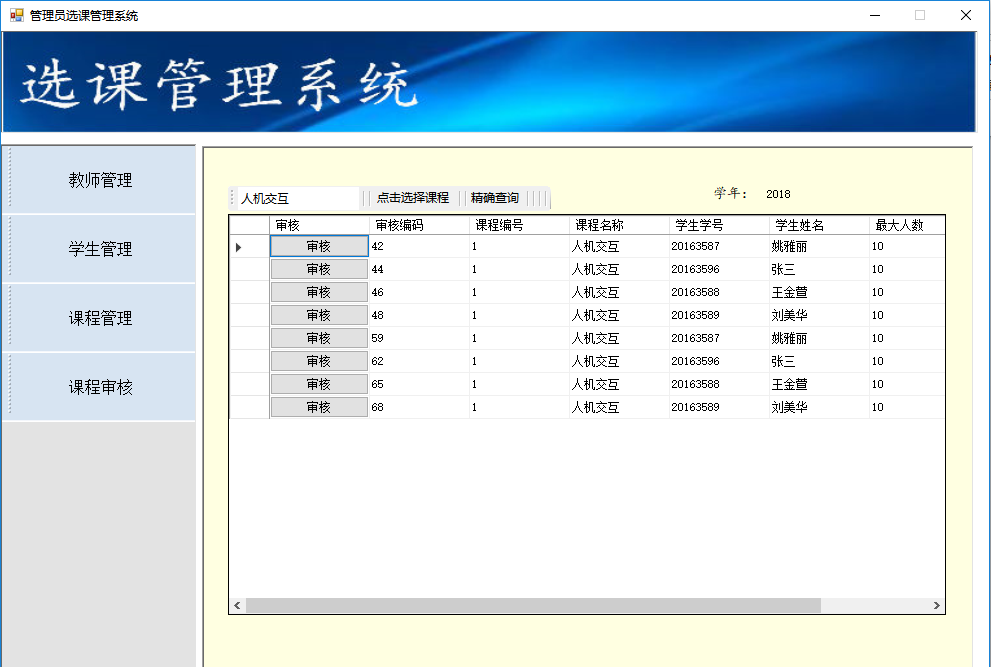
dt.Columns.Add("所属学院", typeof(string));

dt.Rows.Add(c.Course\_num, c.Course\_name, c.Type, c.Academy);

dataGridView1.DataSource = dt;

}

**精准查询：**



**技术点及闪光点：**

管理员点击审核，出现通过和未通过按钮，点击之后审核课程。因为有课程的最大容量，当审核已通过的数量达到课程最大容量的时候，会提示，该课程已达到人数上限。

**核心代码：**

private void toolStripLabel1\_Click(object sender, EventArgs e)

{

Console.Write("!!!!!!!!!!!!!!!!!!!!!!!!!");

string tmp1 = toolStripComboBox1.Text;

dao.shenheDao s = new shenheDao();

model.Course c = new Course();

c = s.find\_course(tmp1);

Console.Write("lalalal"+c.Course\_num);

List<model.Result> l = new List<model.Result>();

num= s.selectAll2(c.Course\_num);

Console.Write(num + "555555555555555555555");

List<model.Result> rs = new List<model.Result>();

rs = s.selectAll1(c.Course\_num);

Console.WriteLine(c.Course\_num+"33333333333333333333333333");

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("审核编码", typeof(string));

dt.Columns.Add("课程编号", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("学生学号", typeof(string));

dt.Columns.Add("学生姓名", typeof(string));

dt.Columns.Add("最大人数", typeof(int));

dt.Columns.Add("状态", typeof(string));

foreach (model.Result r1 in rs)

{

model.Student d = new model.Student();

Console.Write(r1.Student\_num + "111111111111111111");

d=s.find\_student(r1.Student\_num);

Console.Write("1111111111111111111");

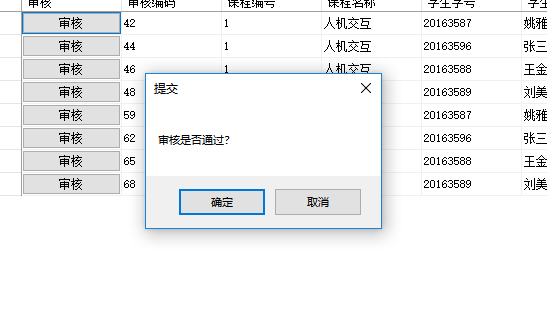
dt.Rows.Add(r1.Result\_num,r1.Course\_num,tmp1, d.Student\_num, d.Student\_name,c.Maxcount,r1.State);

}

dataGridView1.DataSource = dt;

}

**审核课程：**





**技术点及闪光点：**

管理员点击审核，出现通过和未通过按钮，点击之后审核课程。因为有课程的最大容量，当审核已通过的数量达到课程最大容量的时候，会提示，该课程已达到人数上限。

**核心代码：**

private void dataGridView1\_CellContentClick(object sender, System.Windows.Forms.DataGridViewCellEventArgs e)

{

if (e.RowIndex >= 0)

{

int i = e.RowIndex;

MessageBoxButtons messButton = MessageBoxButtons.OKCancel;

String number=Convert.ToString(dataGridView1.Rows[i].Cells[1].Value);

int n = Convert.ToInt32(dataGridView1.Rows[i].Cells[6].Value);

Console.Write(number);

DialogResult dr = MessageBox.Show("审核是否通过？", "提交", messButton);

if (dr == DialogResult.OK)

{

if(num<n)

{

dataGridView1.Rows[i].Cells[7].Value = "审核已通过";

shenheDao.update(number, "审核已通过");

num++;

}

else

{

MessageBox.Show("通过人数已达上限！");

}

}

else

{

dataGridView1.Rows[i].Cells[7].Value = "审核未通过";

shenheDao.update(number,"审核未通过");

}

}

}

**客容量限制：**

**技术点及闪光点：**

因为有课程的最大容量，当审核已通过的数量达到课程最大容量的时候，会提示，该课程已达到人数上限。

**核心代码**

private void dataGridView1\_CellContentClick(object sender, System.Windows.Forms.DataGridViewCellEventArgs e)

{

if (e.RowIndex >= 0)

{

int i = e.RowIndex;

MessageBoxButtons messButton = MessageBoxButtons.OKCancel;

String number=Convert.ToString(dataGridView1.Rows[i].Cells[1].Value);

int n = Convert.ToInt32(dataGridView1.Rows[i].Cells[6].Value);

Console.Write(number);

DialogResult dr = MessageBox.Show("审核是否通过？", "提交", messButton);

if (dr == DialogResult.OK)

{

if(num<n)

{

dataGridView1.Rows[i].Cells[7].Value = "审核已通过";

shenheDao.update(number, "审核已通过");

num++;

}

else

{

MessageBox.Show("通过人数已达上限！");

}

}

else

{

dataGridView1.Rows[i].Cells[7].Value = "审核未通过";

shenheDao.update(number,"审核未通过");

}

}

}

**学生课表查询：**

**技术亮点及闪光点：**

选择不同学年学期，点击查询，根据所写数据库查询语句，多表连接从选课结果表中查询出审核已通过状态的课程，为该学生所选学年学期的课表

**核心代码：**

private void 本学期课表\_Click(object sender, EventArgs e)

{

panel成绩查询.Visible = false;

panel3选课.Visible = false;

panel3学籍.Visible = false;

panel3课表.Visible = true;

}

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

{

dao.resultDao dao = new dao.resultDao();

dao.courseDao cdao = new dao.courseDao();

dao.teacherDao tdao = new dao.teacherDao();

List<model.Result> rs = new List<model.Result>();

rs = dao.selectAll1(student\_login.number, comboBox1课表.Text, comboBox2课表.Text);

Console.WriteLine(comboBox1课表.Text);

Console.WriteLine(comboBox2课表.Text);

// DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("代码", typeof(string));

dt.Columns.Add("课程编号", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("教师", typeof(string));

dt.Columns.Add("学分", typeof(string));

dt.Columns.Add("类型", typeof(string));

dt.Columns.Add("学年", typeof(string));

dt.Columns.Add("学期", typeof(string));

foreach (model.Result r1 in rs)

{

model.Course c = new model.Course();

c = cdao.selectNum(r1.Course\_num);

model.Teacher t = new model.Teacher();

t = tdao.selectNum(c.Teacher\_num);

Console.WriteLine(r1.Result\_num);

Console.WriteLine(c.Type);

Console.WriteLine(c.Course\_num);

Console.WriteLine(c.Type);

Console.WriteLine(r1.Year);

Console.WriteLine(r1.Term);

dt.Rows.Add(r1.Result\_num, c.Course\_num, c.Course\_name, t.Teacher\_name, c.Credit\_hour, c.Type, r1.Year, r1.Term);

}

dataGridView1课表.DataSource = dt;

}

public List<Result> selectAll1(string number, string year, string term)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Result r = null;

List<Result> rs = new List<model.Result>();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM result where student\_num='" + number + "' and year='" + year + "' and term='" + term + "'and state='审核已通过'";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

r = new Result();

r.Result\_num = dataReader.GetString(0);

r.Student\_num = dataReader.GetString(1);

// r.Teacher\_num = dataReader.GetString(2);

r.Course\_num = dataReader.GetString(2);

r.Year = dataReader.GetString(5);

r.Term = dataReader.GetString(6);

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

public Course selectNum(string number)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Course r = new Course();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM course where Course\_num ='" + number + "'";

dataReader = command.ExecuteReader();

Console.WriteLine(command.CommandText);

while (dataReader.Read())

{

r.Course\_num = dataReader.GetString(0);

r.Course\_name = dataReader.GetString(1);

r.Credit\_hour = dataReader.GetString(2);

r.Type = dataReader.GetString(3);

r.Address = dataReader.GetString(4);

r.Time\_week = dataReader.GetString(5);

r.Time\_jie = dataReader.GetString(6);

r.Academy = dataReader.GetString(7);

r.Teacher\_num = dataReader.GetString(9);

return r;

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return null;

}

public model.Teacher selectNum(string number)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

model.Teacher r = new model.Teacher();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM teacher where teacher\_num ='" + number + "'";

dataReader = command.ExecuteReader();

Console.WriteLine(command.CommandText);

while (dataReader.Read())

{

r.Teacher\_num = dataReader.GetString(0);

r.Teacher\_name = dataReader.GetString(1);

r.Teacher\_academy = dataReader.GetString(2);

//r.Question = dataReader.GetString(3);

//r.Answer = dataReader.GetString(4);

r.Password = dataReader.GetString(3);

r.Sex = dataReader.GetString(4);

r.Profession = dataReader.GetString(5);

r.Id\_number = dataReader.GetString(6);

return r;

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return null;

}

**教师课表查询：**



**技术亮点及闪光点:**

教师点击查询，根据所写查询语句，多表连接查询，从课程表中依据当前教师登录号查询课程信息，并在教师表中查询教师姓名，显示在界面中，即自己本学年学期的课程

**核心代码：**

private void 本学期课表\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel7成绩.Visible = false;

panel7选课.Visible = false;

panel3课表.Visible = false;

panel7个人.Visible = false;

panel课表.Visible = true;

}

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

{

dao.resultDao dao = new dao.resultDao();

dao.courseDao cdao = new dao.courseDao();

dao.teacherDao tdao = new dao.teacherDao();

List<model.Course> rs = new List<model.Course>();

rs = cdao.selectNum1(teacher\_login.tnumber);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("课程编号", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("教师", typeof(string));

dt.Columns.Add("上课地点", typeof(string));

dt.Columns.Add("时间", typeof(string));

dt.Columns.Add("课时", typeof(string));

dt.Columns.Add("学分", typeof(string));

dt.Columns.Add("类型", typeof(string));

foreach (model.Course r1 in rs)

{

model.Teacher t = new model.Teacher();

t = tdao.selectNum2(teacher\_login.tnumber);

Console.WriteLine("2:" + r1.Course\_num);

Console.WriteLine("3:" + r1.Course\_name);

Console.WriteLine("4:" + r1.Type);

Console.WriteLine("5:" + t.Teacher\_name);

Console.WriteLine("6:" + r1.Credit\_hour);

Console.WriteLine("7:" + r1.Type);

dt.Rows.Add(r1.Course\_num, r1.Course\_name, t.Teacher\_name,r1.Address,r1.Time\_week,r1.Time\_jie, r1.Credit\_hour, r1.Type);

}

dataGridView2.DataSource = dt;

}

public List<Course> selectNum1(string number)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Course r = new Course();

List<Course> rs = new List<model.Course>();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM course where teacher\_num ='" + number + "'";

dataReader = command.ExecuteReader();

Console.WriteLine(command.CommandText);

while (dataReader.Read())

{

r = new Course();

r.Course\_num = dataReader.GetString(0);

r.Course\_name = dataReader.GetString(1);

r.Credit\_hour = dataReader.GetString(2);

r.Type = dataReader.GetString(3);

r.Address = dataReader.GetString(4);

r.Time\_week = dataReader.GetString(5);

r.Time\_jie = dataReader.GetString(6);

r.Academy = dataReader.GetString(7);

r.Teacher\_num = dataReader.GetString(9);

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

public model.Teacher selectNum2(string number)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

model.Teacher r = new model.Teacher();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM teacher where teacher\_num ='" + number + "'";

dataReader = command.ExecuteReader();

Console.WriteLine(command.CommandText);

while (dataReader.Read())

{

r.Teacher\_num = dataReader.GetString(0);

r.Teacher\_name = dataReader.GetString(1);

return r;

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

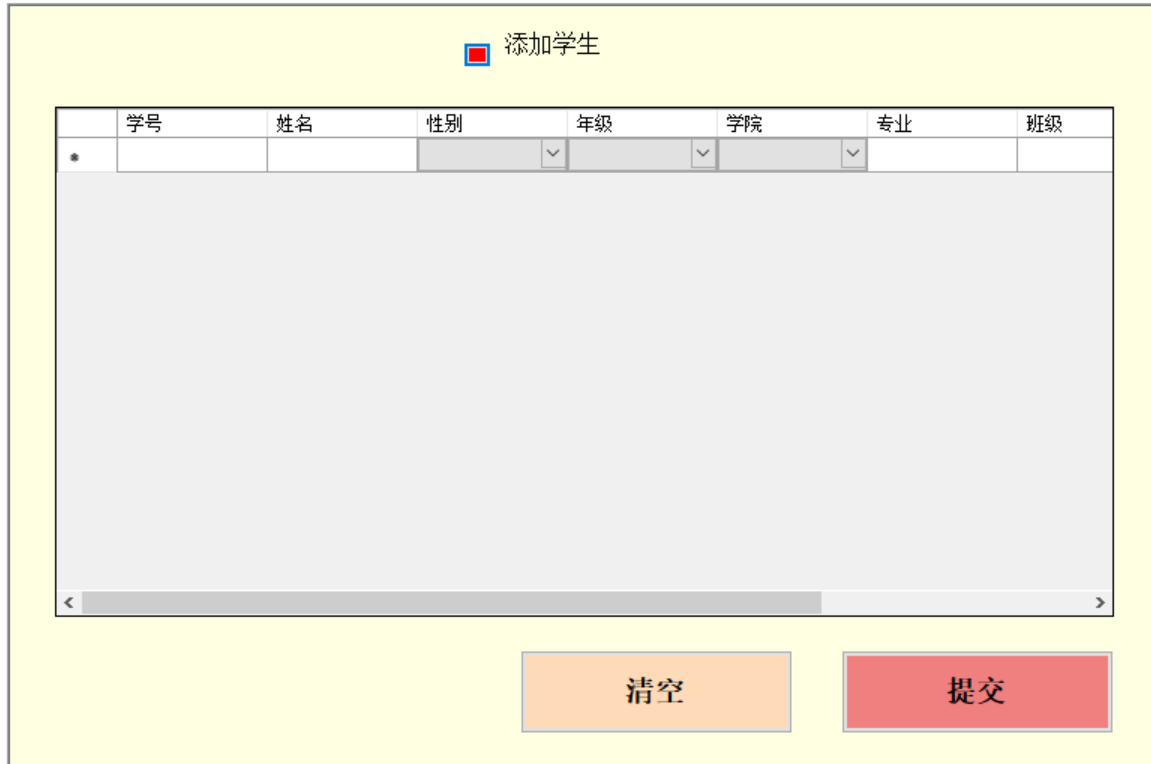
}

}

return null;

}

**添加学生：**

**技术点及闪光点：**

可以自动弹出学号，不用用户自己输入。用户可以自己选择性别，年级，学院，不会输入错误。用户按enter键或tab键光标自动跳到下一个表格。

**核心代码：**

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

{

bool wanzheng = true; ;//用户是否输入完整

int i = dataGridView1.RowCount;

// MessageBox.Show("" + i);

dao.studentDao Sdao = new studentDao();

for (int j = 0; j < i-1; j++)//检查时候有未填的空

{

string Student\_num = Convert.ToString(dataGridView1.Rows[j].Cells[0].Value);//学号

string Student\_name = Convert.ToString(dataGridView1.Rows[j].Cells[1].Value);//学号

string Sex = Convert.ToString(dataGridView1.Rows[j].Cells[2].Value);//性别

string Student\_grade = Convert.ToString(dataGridView1.Rows[j].Cells[3].Value);//年纪

string Student\_academy = Convert.ToString(dataGridView1.Rows[j].Cells[4].Value);//学院

string Profession = Convert.ToString(dataGridView1.Rows[j].Cells[5].Value);//专业

string Id\_number = Convert.ToString(dataGridView1.Rows[j].Cells[7].Value);//身份证号

string Classs = Convert.ToString(dataGridView1.Rows[j].Cells[6].Value);//身份证号

if(Student\_num==""||Student\_name==""||Sex==""||Student\_grade==""||Student\_grade==""||

Student\_academy == "" || Profession == "" || Id\_number == "" || Classs == "")

{

MessageBox.Show("请输入完整信息！");

wanzheng = false;

}

}

if (wanzheng == true)

{

for (int j = 0; j < i; j++)

{

model.Student student = new Student();

student.Student\_num = Convert.ToString(dataGridView1.Rows[j].Cells[0].Value);//学号

student.Student\_name = Convert.ToString(dataGridView1.Rows[j].Cells[1].Value);//学号

student.Sex = Convert.ToString(dataGridView1.Rows[j].Cells[2].Value);//性别

student.Student\_grade = Convert.ToString(dataGridView1.Rows[j].Cells[3].Value);//年纪

student.Student\_academy = Convert.ToString(dataGridView1.Rows[j].Cells[4].Value);//学院

student.Profession = Convert.ToString(dataGridView1.Rows[j].Cells[5].Value);//专业

student.Id\_number = Convert.ToString(dataGridView1.Rows[j].Cells[7].Value);//身份证号

student.Classs = Convert.ToString(dataGridView1.Rows[j].Cells[6].Value);//身份证号

string id\_number = student.Id\_number;

if (id\_number.Length < 18)

{

//MessageBox.Show(student.Id\_number.Length + "请输入正确的" + student.Student\_name + "的身份证号码");

break;

}

student.Password = id\_number.Substring(id\_number.Length - 6, 6);

bool isd = dao.studentDao.add\_student(student);

if (isd == false)

{

MessageBox.Show("提交失败！");

break;

}

}

MessageBox.Show("提交成功！");

dataGridView1.Rows.Clear();

s\_count = 0;

}

}

protected override bool ProcessCmdKey(ref Message msg, Keys keyData)

{

base.ProcessCmdKey(ref msg, keyData);

int WM\_KEYDOWN = 256;

int WM\_SYSKEYDOWN = 260;

bool \_disable = false;

if (msg.Msg == WM\_KEYDOWN || msg.Msg == WM\_SYSKEYDOWN)

{

switch (keyData)

{

case Keys.Enter:

SendKeys.Send("{Tab}");

\_disable = true;

break;

}

}

return \_disable;

}

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

{

if (e.ColumnIndex == 0 || e.ColumnIndex == -1)

{

string num = dao.studentDao.find\_num();

int num2 = int.Parse(num)+s\_count;

num2 = num2 + 1;

dataGridView1.CurrentRow.Cells[0].Value = num2;

s\_count++;

}

}

**查看学生：**



**技术点及闪光点：**

用户可以输入多条件，来查询学生，并且删除学生时会弹出提示框确认删除，用户修改学生信息时，修改完成后可以及时更新信息。

**核心代码：**

panel7.Visible = false;

panel4.Visible = false;

panel9.Visible = false;

panel5.Visible = false;

panel添加教师页面.Visible = false;

查看教师页面.Visible = false;

panel2.Visible = true;

List<Student> students = new List<Student>();

students = dao.studentDao.find\_students();

int i = 0;

if (students!= null)

{

foreach (Student s in students)

{

dataGridView2.Rows.Add();

dataGridView2.Rows[i].Cells[0].Value =s.Student\_num;

dataGridView2.Rows[i].Cells[1].Value = s.Student\_name;

dataGridView2.Rows[i].Cells[2].Value = s.Sex;

dataGridView2.Rows[i].Cells[3].Value = s.Student\_grade;

dataGridView2.Rows[i].Cells[4].Value = s.Student\_academy;

dataGridView2.Rows[i].Cells[5].Value = s.Profession;

dataGridView2.Rows[i].Cells[6].Value = s.Classs;

dataGridView2.Rows[i].Cells[7].Value = s.Id\_number;

i++;

}

}

dataGridView2.ReadOnly = true;

}

private void button4\_Click(object sender, EventArgs e)

{

Student ss = new Student();

ss.Student\_num = textBox3.Text;

ss.Student\_name = textBox4.Text;

ss.Student\_grade = comboBox1.Text;

ss.Student\_academy = comboBox2.Text;

ss.Classs = textBox2.Text;

ss.Profession = textBox1.Text;

List<Student> students = new List<Student>();

if (ss.Student\_num == "" && ss.Student\_name == "" && ss.Student\_grade == "" && ss.Student\_academy == ""

&& ss.Classs == "" && ss.Profession == "")//如果输入为空，则显示全部学生信息

{

students = dao.studentDao.find\_students();

}

else//如果输入查询条件则更具条件查询

{

students = dao.studentDao.find\_students2(ss);

}

dataGridView2.Rows.Clear();

int i = 0;

if (students != null)

{

foreach (Student s in students)

{

dataGridView2.Rows.Add();

dataGridView2.Rows[i].Cells[0].Value = s.Student\_num;

dataGridView2.Rows[i].Cells[1].Value = s.Student\_name;

dataGridView2.Rows[i].Cells[2].Value = s.Sex;

dataGridView2.Rows[i].Cells[3].Value = s.Student\_grade;

dataGridView2.Rows[i].Cells[4].Value = s.Student\_academy;

dataGridView2.Rows[i].Cells[5].Value = s.Profession;

dataGridView2.Rows[i].Cells[6].Value = s.Classs;

dataGridView2.Rows[i].Cells[7].Value = s.Id\_number;

i++;

}

}

//清空原来输入的查询条件

textBox3.Text="";

textBox4.Text="";

comboBox1.Text="";

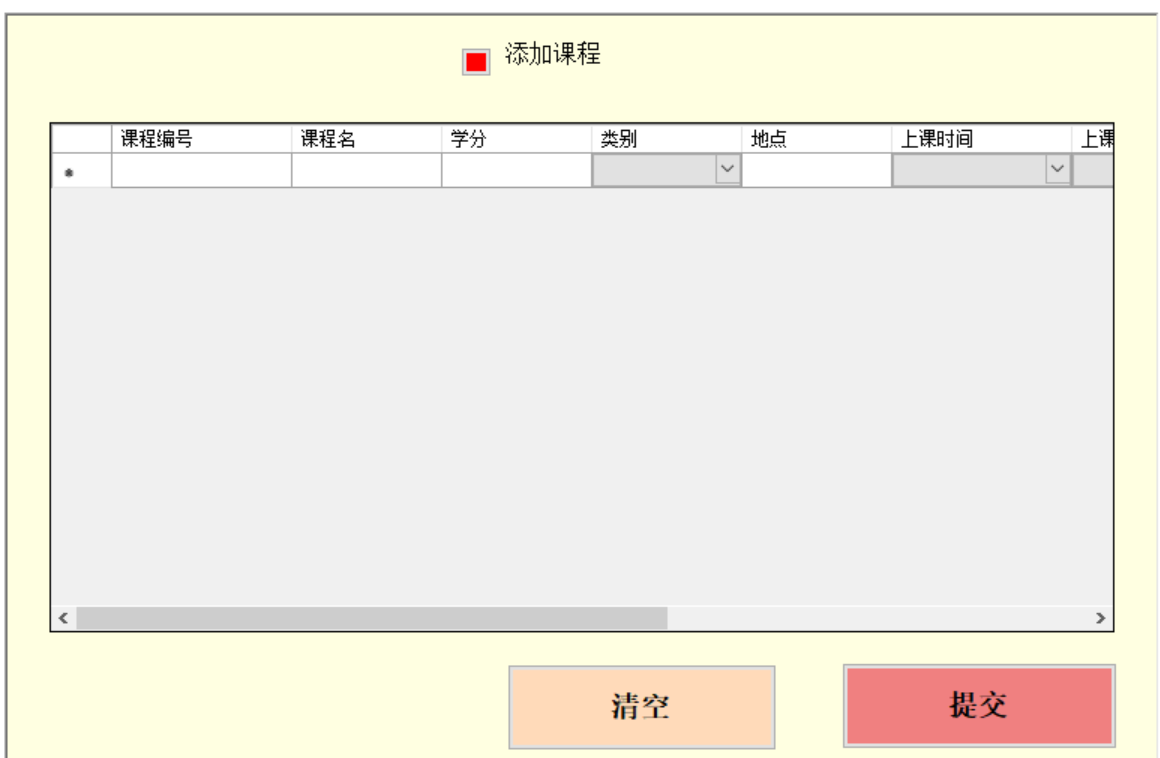
comboBox2.Text="";

textBox2.Text="";

textBox1.Text="";

}

**添加课程：**

**技术闪光点：**

可以自动弹出课程号，不用用户自己输入。用户可以自己选择类别，上课时间，学院，不会输入错误。用户按enter键或tab键光标自动跳到下一个表格。还可以弹出弹框来选择已有老师。

**核心代码：**

private void button6\_Click(object sender, EventArgs e)

{

bool wanzheng = true;//用户是否输入完整

int i = dataGridView3.RowCount;

dao.courseDao Sdao = new courseDao();

for (int j = 0; j < i - 1; j++)//检查时候有未填的空

{

string Course\_num = Convert.ToString(dataGridView3.Rows[j].Cells[0].Value);//课程号

string Course\_name = Convert.ToString(dataGridView3.Rows[j].Cells[1].Value);//课程名

string Credit\_hour = Convert.ToString(dataGridView3.Rows[j].Cells[2].Value);//学分

string Type = Convert.ToString(dataGridView3.Rows[j].Cells[3].Value);//类别

string Address = Convert.ToString(dataGridView3.Rows[j].Cells[4].Value);//地点

string Time\_week = Convert.ToString(dataGridView3.Rows[j].Cells[5].Value);//上课时间

string time\_jie = Convert.ToString(dataGridView3.Rows[j].Cells[6].Value);//上课时间

string Academy = Convert.ToString(dataGridView3.Rows[j].Cells[7].Value);//学院

string maxcount = Convert.ToString(dataGridView3.Rows[j].Cells[8].Value);//最大容量

string teacher\_num = teacher\_nums[j];//老师工号

if (Course\_num == "" || Course\_name == "" || Credit\_hour == "" || Type == "" || Address == "" ||

Time\_week == "" || time\_jie == "" || Academy == ""||maxcount==""||teacher\_num=="" )

{

MessageBox.Show("请输入完整信息！");

wanzheng = false;

}

}

if (wanzheng == true)

{

for (int j = 0; j < i-1; j++)

{

model.Course course = new Course();

course .Course\_num= Convert.ToString(dataGridView3.Rows[j].Cells[0].Value);//课程号

course.Course\_name = Convert.ToString(dataGridView3.Rows[j].Cells[1].Value);//课程名

course.Credit\_hour = Convert.ToString(dataGridView3.Rows[j].Cells[2].Value);//学分

course.Type = Convert.ToString(dataGridView3.Rows[j].Cells[3].Value);//类别

course.Address = Convert.ToString(dataGridView3.Rows[j].Cells[4].Value);//地点

course.Time\_week = Convert.ToString(dataGridView3.Rows[j].Cells[5].Value);//上课时间

course.Time\_jie = Convert.ToString(dataGridView3.Rows[j].Cells[6].Value);//上课时间

course.Academy = Convert.ToString(dataGridView3.Rows[j].Cells[7].Value);//学院

course.Maxcount = Convert.ToInt32(dataGridView3.Rows[j].Cells[8].Value);//最大容量

MessageBox.Show("容量1：" + course.Maxcount);

course.Teacher\_num = teacher\_nums[j];//老师工号

MessageBox.Show("工号2：" + teacher\_nums[j]+" j:"+j);

bool isd = dao.courseDao.add\_course(course);

if (isd == false)

{

MessageBox.Show("提交失败！");

break;

}

}

MessageBox.Show("提交成功！");

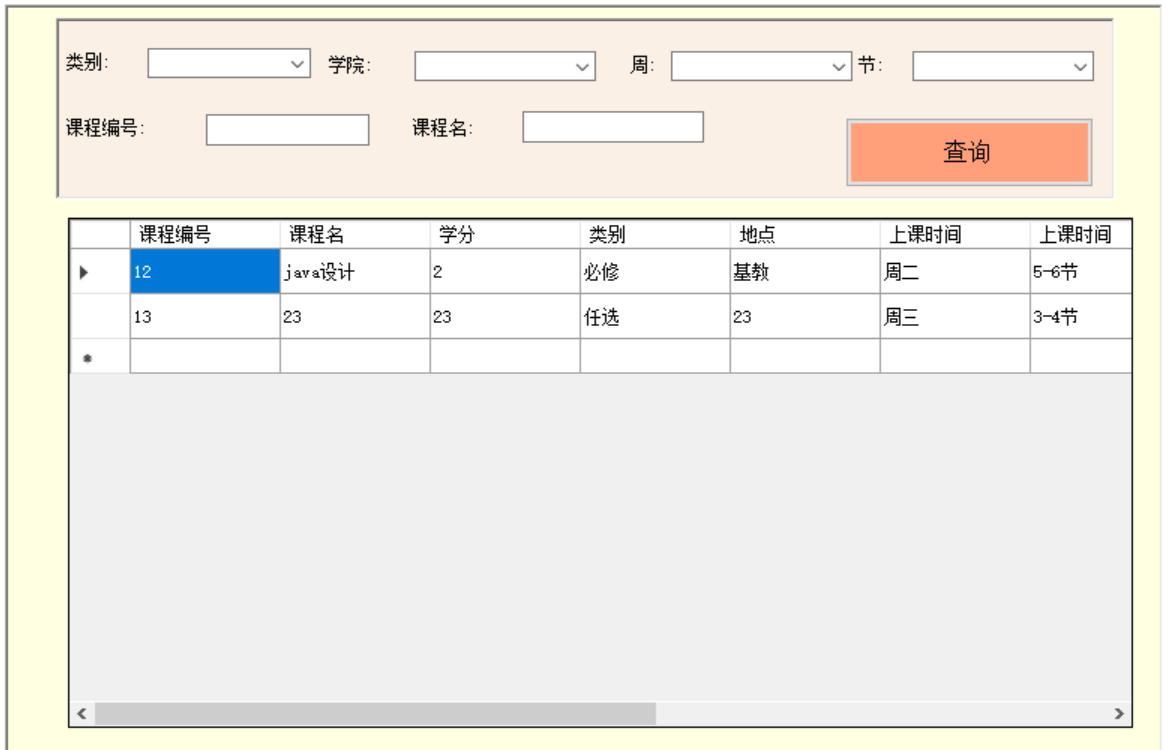
dataGridView3.Rows.Clear();

teacher\_nums.Clear();

c\_count = 0;

}

}

**查看课程：**

**技术闪光点**

用户可以输入多条件，来查询课程，并且删除课程时会弹出提示框确认删除，用户修改课程信息时，修改完成后可以及时更新信息。

**核心代码：**

private void button8\_Click(object sender, EventArgs e)

{

Course ss = new Course();

ss.Type = comboBox4.Text;

ss.Academy = comboBox3.Text;

ss.Time\_week = comboBox5.Text;

ss.Time\_jie = comboBox6.Text;

ss.Course\_num = textBox6.Text;

ss.Course\_name = textBox5.Text;

List<Course> courses = new List<Course>();

if (ss.Type == "" && ss.Academy == "" && ss.Time\_week == "" && ss.Time\_jie == ""

&& ss.Course\_num== "" && ss.Course\_name == "")//如果输入为空，则显示全部学生信息

{

courses = dao.courseDao.find\_courses();

}

else//如果输入查询条件则更具条件查询

{

courses = dao.courseDao.find\_courses2(ss);

}

dataGridView4.Rows.Clear();

int i = 0;

if (courses != null)

{

foreach (Course s in courses)

{

dataGridView4.Rows.Add();

dataGridView4.Rows[i].Cells[0].Value = s.Course\_num;

dataGridView4.Rows[i].Cells[1].Value = s.Course\_name;

dataGridView4.Rows[i].Cells[2].Value = s.Credit\_hour;

dataGridView4.Rows[i].Cells[3].Value = s.Type;

dataGridView4.Rows[i].Cells[4].Value = s.Address;

dataGridView4.Rows[i].Cells[5].Value = s.Time\_week;

dataGridView4.Rows[i].Cells[6].Value = s.Time\_jie;

dataGridView4.Rows[i].Cells[7].Value = s.Academy;

dataGridView4.Rows[i].Cells[8].Value = s.Maxcount;

dataGridView4.Rows[i].Cells[9].Value = s.Teacher\_num;

i++;

}

}

//清空原来输入的查询条件

comboBox4.Text="";

comboBox3.Text="";

comboBox5.Text="";

comboBox6.Text="";

textBox6.Text="";

textBox5.Text="";

}

private void toolStripMenuItem6\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel4.Visible = false;

panel2.Visible = false;

panel9.Visible = false;

panel添加教师页面.Visible = false;

查看教师页面.Visible = false;

panel5.Visible = true;

List<Course> courses = new List<Course>();

courses = dao.courseDao.find\_courses();

int i = 0;

if (courses != null)

{

foreach (Course s in courses)

{

dataGridView4.Rows.Add();

dataGridView4.Rows[i].Cells[0].Value = s.Course\_num;

dataGridView4.Rows[i].Cells[1].Value = s.Course\_name;

dataGridView4.Rows[i].Cells[2].Value = s.Credit\_hour;

dataGridView4.Rows[i].Cells[3].Value = s.Type;

dataGridView4.Rows[i].Cells[4].Value = s.Address;

dataGridView4.Rows[i].Cells[5].Value = s.Time\_week;

dataGridView4.Rows[i].Cells[6].Value = s.Time\_jie;

dataGridView4.Rows[i].Cells[7].Value = s.Academy;

dataGridView4.Rows[i].Cells[8].Value = s.Maxcount;

dataGridView4.Rows[i].Cells[9].Value = s.Teacher\_num;

i++;

}

}

dataGridView4.ReadOnly = true;

}

private void dataGridView4\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

if (e.ColumnIndex == 8)

{

int i = e.RowIndex;

string Course\_num = Convert.ToString(dataGridView4.Rows[i].Cells[0].Value);

MessageBox.Show("" + Course\_num);

courseDao.delete(Course\_num);

dataGridView4.Rows.Remove(dataGridView4.Rows[i]);

}

else if (e.ColumnIndex == 9)//查询

{

int i = e.RowIndex;

Course s = new Course();

string Course\_num = Convert.ToString(dataGridView4.Rows[i].Cells[0].Value);

Form f1 = new course\_update(Course\_num);

f1.ShowDialog();

if (course != null)//用户修改了信息

{

dataGridView4.Rows[i].Cells[0].Value = course.Course\_num;

dataGridView4.Rows[i].Cells[1].Value = course.Course\_name;

dataGridView4.Rows[i].Cells[2].Value = course.Credit\_hour;

dataGridView4.Rows[i].Cells[3].Value = course.Type;

dataGridView4.Rows[i].Cells[4].Value = course.Address;

dataGridView4.Rows[i].Cells[5].Value = course.Time\_week;

dataGridView4.Rows[i].Cells[6].Value = course.Time\_jie;

dataGridView4.Rows[i].Cells[7].Value = course.Academy;

dataGridView4.Rows[i].Cells[8].Value = course.Maxcount;

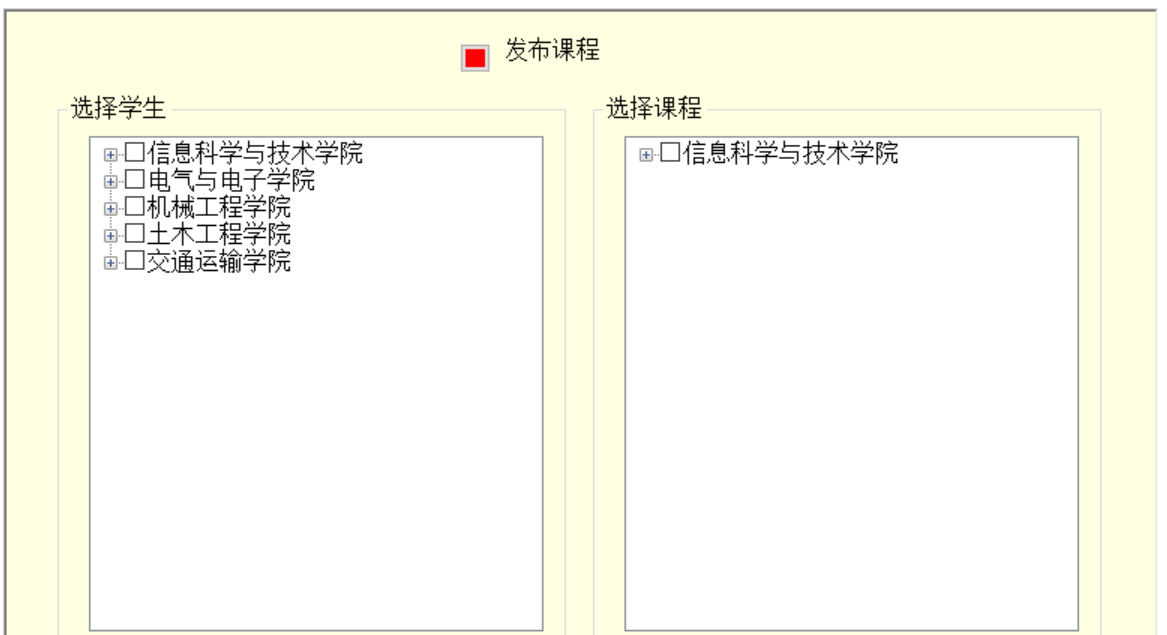
dataGridView4.Rows[i].Cells[9].Value = course.Teacher\_num;

}

}

}

**发布课程:**

**核心代码：**

private void 发布ToolStripMenuItem\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel4.Visible = false;

panel2.Visible = false;

panel5.Visible = false;

panel添加教师页面.Visible = false;

查看教师页面.Visible = false;

panel9.Visible = true;

List<string> student\_academys = dao.studentDao.find\_academys();

string[][] class2 = new string[student\_academys.Count][];

int i = 0;

foreach (string student\_academy in student\_academys)

{

List<string> classs = dao.studentDao.find\_classs(student\_academy);

foreach (string class1 in classs)

{

class2[i] = new string[] { student\_academy, class1 };

Console.WriteLine(student\_academy + " " + class1);

i++;

}

}

}

private void treeView1\_AfterCheck(object sender, TreeViewEventArgs e)

{

if (e.Node.Nodes.Count > 0)

{

foreach (TreeNode node in e.Node.Nodes)

{

node.Checked = e.Node.Checked;

}

}

}

private void treeView1\_AfterSelect(object sender, TreeViewEventArgs e)

{

}

private void 发布ToolStripMenuItem\_Click\_1(object sender, EventArgs e)

{

panel7.Visible = false;

panel4.Visible = false;

panel2.Visible = false;

panel5.Visible = false;

panel添加教师页面.Visible = false;

查看教师页面.Visible = false;

panel9.Visible = true;

//学生树

List<string> student\_academys = dao.studentDao.find\_academys();

int count = dao.studentDao.find\_classCount();

string[][] class2 = new string[count][];

int i = 0;

foreach (string student\_academy in student\_academys)

{

List<string> classs = dao.studentDao.find\_classs(student\_academy);

foreach (string class1 in classs)

{

class2[i] = new string[] { student\_academy, class1 };

i++;

}

}

foreach (var data in class2)

{

string date = data[0];

string value = data[1];

TreeNode datenode = new TreeNode(date);

datenode.Name = date;

TreeNode valuenode = new TreeNode(value);

if (!treeView1.Nodes.ContainsKey(date))

{

treeView1.Nodes.Add(datenode);

treeView1.Nodes[date].Nodes.Add(valuenode);

}

else

{

treeView1.Nodes[date].Nodes.Add(valuenode);

}

}

////课程树

List<string> course\_academys = dao.courseDao.find\_academys();

count = dao.courseDao.find\_countCount();

string[][] course2 = new string[count][];

int j = 0;

foreach (string academy in course\_academys)

{

List<string> classs = dao.courseDao.find\_courses3(academy);

foreach (string class1 in classs)

{

course2[j] = new string[] { academy, class1 };

j++;

}

}

foreach (var data in course2)

{

string date = data[0];

string value = data[1];

TreeNode datenode = new TreeNode(date);

datenode.Name = date;

TreeNode valuenode = new TreeNode(value);

if (!treeView2.Nodes.ContainsKey(date))

{

treeView2.Nodes.Add(datenode);

treeView2.Nodes[date].Nodes.Add(valuenode);

}

else

{

treeView2.Nodes[date].Nodes.Add(valuenode);

}

}

}

private void treeView1\_AfterCheck\_1(object sender, TreeViewEventArgs e)

{

if (e.Node.Nodes.Count > 0)

{

foreach (TreeNode node in e.Node.Nodes)

{

node.Checked = e.Node.Checked;

}

}

}

private void treeView2\_AfterSelect(object sender, TreeViewEventArgs e)

{

}

private void treeView2\_AfterCheck(object sender, TreeViewEventArgs e)

{

if (e.Node.Nodes.Count > 0)

{

foreach (TreeNode node in e.Node.Nodes)

{

node.Checked = e.Node.Checked;

}

}

}

private void button11\_Click(object sender, EventArgs e)

{

//获取选中的班级

List<string> classs=new List<string>();

foreach (TreeNode tn in treeView1.Nodes)

{

foreach(TreeNode tn2 in tn.Nodes){

if (tn2.Checked == true)

{

classs.Add(tn2.Text);

}

}

}

//获取选中的课程号

List<string> course\_num = new List<string>();

foreach (TreeNode tn in treeView2.Nodes)

{

foreach (TreeNode tn2 in tn.Nodes)

{

if (tn2.Checked == true)

{

string num = tn2.Text;

int IndexofA = num.IndexOf('(');

int IndexofB = num.IndexOf(')');

string course\_num2 = num.Substring(IndexofA + 1, IndexofB - IndexofA - 1);

//string course\_num2 = tn2.Text.Substring(1, 3);

course\_num.Add(course\_num2);

}

}

}

//查询出该班级内的所有学号

List<string> student\_num = dao.studentDao.find\_studentNums(classs);

//讲学号以及课程号写入result表

List<Result> results = new List<Result>();

foreach (string n in student\_num)

{

foreach (string c in course\_num)

{

System.DateTime currentTime=new System.DateTime();

currentTime=System.DateTime.Now;

Result result = new Result();

result.Course\_num = c;

result.Student\_num = n;

result.Term = "春";

result.Year = currentTime.Year.ToString();

dao.resultDao.add\_result(result);

}

}

MessageBox.Show("提交成功！");

//取消节点前的复选框选中状态

foreach (TreeNode tn in treeView1.Nodes)

{

if (tn.Checked == true) tn.Checked = false;

foreach (TreeNode tn2 in tn.Nodes)

{

if (tn2.Checked == true)

{

tn2.Checked = false;

}

}

}

foreach (TreeNode tn in treeView2.Nodes)

{

if (tn.Checked == true) tn.Checked = false;

foreach (TreeNode tn2 in tn.Nodes)

{

if (tn2.Checked == true)

{

tn2.Checked = false;

}

}

}

}

第5章 心得体会

学习一门新的语言时，总会遇到很多问题。与其他语言不同的是，C#界面是大多是直接绘制的，这是即方便又不方便的地方。首先需要我们去了解每一个控件的功能、效果、属性设置，要考虑到功能与功能，功能与界面，界面与界面之间的各种关系，还需要将控件与数据库以及效果显示代码结合起来。此外，由于这是一个完整的系统，我们要考虑的不仅仅是功能的实现，还得注意良好的界面设计，团队的分工合作，这些都花费了我们很多时间去实践。

技术上，我们再数据表的关联上花费了很多时间，首先我们需要明白选课的正确流程，然后我们才能正确的设计表，也是因为在这上面没有一开始弄清楚，导致后面再次返工。我们大量使用了DataGridView列表的显示，需要用到for循环结合数据库语言的List设计。

选课时的多重限制是我们一开始做得不足的地方，限制太少，导致系统不具备实用性。修改后，添加了课程人数限制，管理员审核等功能，是系统流程更加完善。

这次系统的开发，让我们看到了需求调研的重要新，提前准备的迫切性，团队合作的有效性以及思维的缜密重要性，也相信，我们之后的项目开发一定可以从中找到经验教训，不断提高。

参考文献

[1] 孙东梅： 《Dreamweaver 8 +ASP 完美动态网站设计》 ，电子工业出版社；2007 年 5 月第一次印刷

[2] 陈建伟 李美军 施建强： 《ASP 动态网站开发教程》 ，第一版：清华大学出版社； 2005 年一月

[3] 肖志刚 张维 韩璐： 《ASP 动态网页设计应用培训教程》 ，第一版：电子工业  
出版社； 2004 年 1 月

[4] 温明： 《ASP 网站建设实录》 ，红旗出版社；2005 年 2 月第一次印刷

[5] 苗雪兰 刘瑞新 王怀峰： 《数据库系统原理及应用教程》 ，第一版：机械工业出 版社；2001 年 7 月

[6] 邵丽萍 张后扬 郭春芳： 《网站编程技术实用教程》 ，第一版：清华大学出版社； 2005 年 1 月

附录

教师页面主要代码：

private void teacher\_home\_Load(object sender, EventArgs e)

{

Timer = new Timer();//显示当前时间

Timer.Interval = 1000;

Timer.Tick += new EventHandler(Timer\_Tick);

Timer.Start();

panel7成绩.Visible = false;

dao.resultDao rdao = new dao.resultDao();

dao.courseDao cdao = new dao.courseDao();

List<model.Course> cs = new List<model.Course>();

cs = cdao.selecttc(teacher\_login.tnumber);

foreach (model.Course r1 in cs)

{

model.Course c = new model.Course();

c = cdao.selectNum(r1.Course\_num);

//Console.WriteLine(c.Course\_num + " " + c.Course\_name + c.Time\_week + c.Time\_jie);

comboBox1成绩.Items.Add(c.Course\_num + " " + c.Course\_name + c.Time\_week + c.Time\_jie);

}

}

private void Timer\_Tick(object sender, EventArgs e)

{

this.label2.Text = DateTime.Now.ToString("HH:mm:ss");

this.label1.Text = DateTime.Now.ToString("yyyy-MM-dd");

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void tableLayoutPanel1\_Paint(object sender, PaintEventArgs e)

{

}

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

{

}

private void 评估课程\_Click(object sender, EventArgs e)

{

panel7选课.Visible = false;

panel7成绩.Visible = false;

panel3课表.Visible = false;

panel课表.Visible = false;

panel7个人.Visible = false;

panel7.Visible = true;

}

private void 个人成绩查询\_Click(object sender, EventArgs e)

{

panel7选课.Visible = false;

panel7.Visible = false;

panel3课表.Visible = false;

panel课表.Visible = false;

panel7个人.Visible = false;

panel7成绩.Visible = true;

}

private void 学生选课\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel7成绩.Visible = false;

panel3课表.Visible = false;

panel课表.Visible = false;

panel7个人.Visible = false;

panel7选课.Visible = true;

string str = util.Util.sqlcon;

MySqlConnection conn = new MySqlConnection(str);

string sql = "select teacher\_name,teacher\_academy from teacher where teacher\_num='" + teacher\_login.tnumber + "'";

conn.Open();

MySqlCommand cmd = new MySqlCommand(sql, conn);

MySqlDataReader thisread = cmd.ExecuteReader();

Console.WriteLine("sql:" + sql);

thisread.Read();

string temp1 = thisread.GetValue(0).ToString().Trim();

string temp2 = thisread.GetValue(1).ToString().Trim();

conn.Close();

TeacherSelection ts = new TeacherSelection();

ts.label6Text = temp1;

ts.label2Text = temp2;

ts.label9Text = label3.Text;

ts.label7Text = teacher\_login.tnumber;

ts.FormBorderStyle = FormBorderStyle.None;

ts.Dock = DockStyle.Fill;

ts.TopLevel = false;

panel7选课.Controls.Clear();

panel7选课.Controls.Add(ts);

ts.Show();

}

private void 本学期课表\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel7成绩.Visible = false;

panel7选课.Visible = false;

panel3课表.Visible = false;

panel7个人.Visible = false;

panel课表.Visible = true;

}

private void panel2\_Paint\_1(object sender, PaintEventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void label1\_Click\_1(object sender, EventArgs e)

{

}

private void panel1\_Paint\_1(object sender, PaintEventArgs e)

{

}

private void 学籍信息\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel7成绩.Visible = false;

panel7选课.Visible = false;

panel3课表.Visible = false;

panel课表.Visible = false;

panel7个人.Visible = true;

teacher\_information sAllPage = new teacher\_information();

sAllPage.FormBorderStyle = FormBorderStyle.None;

sAllPage.Dock = DockStyle.Fill;

sAllPage.TopLevel = false;

panel7个人.Controls.Clear();

panel7个人.Controls.Add(sAllPage);

sAllPage.Show();

}

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

{

dao.resultDao dao = new dao.resultDao();

dao.studentDao sdao = new dao.studentDao();

List<model.Result> rs = new List<model.Result>();

rs = dao.selectStudent(textBox1.Text, comboBox1.Text, comboBox2.Text);//学生编号s

List<model.Student> s = new List<model.Student>();

s = sdao.selectStudent(rs);//根据学生编号s查询出所有的学生信息

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("姓名", typeof(string));

dt.Columns.Add("学号", typeof(string));

dt.Columns.Add("性别", typeof(String));

dt.Columns.Add("学院", typeof(string));

dt.Columns.Add("专业", typeof(String));

dt.Columns.Add("年级", typeof(String));

dt.Columns.Add("班级", typeof(String));

foreach (model.Student a in s)

{

dt.Rows.Add(a.Student\_name, a.Id\_number, a.Sex, a.Student\_academy, a.Profession, a.Student\_grade, a.Classs);

}

dataGridView1.DataSource = dt;

}

//录入成绩

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

{

string course = comboBox1成绩.Text;

if (course == "")

{

MessageBox.Show("请选择需要录入的课程！");

}

else

{

dataGridView1成绩.Rows.Clear();

//coursedao cdao = new coursedao();

string[] c = course.Split(' ');

string cour = c[0];

dao.resultDao rdao = new dao.resultDao();

List<model.Result> rs = new List<model.Result>();

rs = rdao.selectResult(cour);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dao.studentDao sdao = new dao.studentDao();

Console.WriteLine("姚雅丽");

dt.Columns.Add("学号", typeof(string));

dt.Columns.Add("姓名", typeof(string));

dt.Columns.Add("性别", typeof(string));

dt.Columns.Add("类型", typeof(string));

dt.Columns.Add("成绩", typeof(string));

foreach (DataGridViewColumn co in dataGridView1成绩.Columns)

{

if (co.Index != 4 && co.Index != 3)

co.ReadOnly = true;

}

foreach (model.Result r in rs)

{

model.Student s = new model.Student();

s = sdao.selectNum(r.Student\_num);

dataGridView1成绩.Rows.Add(r.Student\_num, s.Student\_name, s.Sex, "", "");

}

}

}

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

{

}

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

{

MessageBoxButtons messButton = MessageBoxButtons.OKCancel;

DialogResult dr = MessageBox.Show("是否确认提交？", "提交", messButton);

if (dr == DialogResult.OK)

{

string course = comboBox1成绩.Text;

dao.resultDao rdao = new dao.resultDao();

string[] c = course.Split(' ');

string cour = c[0];

bool b = true;

for (int i = 0; i < dataGridView1成绩.Rows.Count ; i++)

{

try

{

rdao.updateScore(dataGridView1成绩.Rows[i].Cells[0].Value.ToString(), cour, dataGridView1成绩.Rows[i].Cells[3].Value.ToString(), Convert.ToDouble(dataGridView1成绩.Rows[i].Cells[4].Value));

}

catch (SystemException)

{

MessageBox.Show("格式错误！");

b = false;

}

}

if (b == true)

{

MessageBox.Show("提交成功");

}

}

else

{

}

}

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

{

//dao.resultDao dao = new dao.resultDao();

//dao.courseDao cdao = new dao.courseDao();

//dao.teacherDao tdao = new dao.teacherDao();

//List<model.Result> rs = new List<model.Result>();

//rs = dao.selectAll2(comboBox1课表.Text, comboBox2课表.Text);

//Console.WriteLine(comboBox1课表.Text);

//Console.WriteLine(comboBox2课表.Text);

//DataSet ds = new DataSet();

//DataTable dt = new DataTable("Table\_New");

//dt.Columns.Add("代码", typeof(string));

//dt.Columns.Add("课程编号", typeof(string));

//dt.Columns.Add("课程名称", typeof(string));

//dt.Columns.Add("教师", typeof(string));

//dt.Columns.Add("学分", typeof(string));

//dt.Columns.Add("类型", typeof(string));

//dt.Columns.Add("学年", typeof(string));

//dt.Columns.Add("学期", typeof(string));

//foreach (model.Result r1 in rs)

//{

// model.Course c = new model.Course();

//c = cdao.selectNum(r1.Course\_num);

//model.Teacher t = new model.Teacher();

// t = tdao.selectNum(r1.Teacher\_num);

/\*\*Console.WriteLine("1:" + r1.Result\_num);

Console.WriteLine("2:" + c.Course\_num);

Console.WriteLine("3:" + c.Course\_name);

Console.WriteLine("4:" + c.Type);

Console.WriteLine("5:" + t.Teacher\_name);

Console.WriteLine("6:" + c.Credit\_hour);

Console.WriteLine("7:" + r1.Type);\*\*/

//dt.Rows.Add(r1.Result\_num, c.Course\_num, c.Course\_name, t.Teacher\_name, c.Credit\_hour, c.Type, r1.Year, r1.Term);

//}

//dataGridView1课表.DataSource = dt;

}

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

{

dao.resultDao dao = new dao.resultDao();

dao.courseDao cdao = new dao.courseDao();

dao.teacherDao tdao = new dao.teacherDao();

List<model.Course> rs = new List<model.Course>();

rs = cdao.selectNum1(teacher\_login.tnumber);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("课程编号", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("教师", typeof(string));

dt.Columns.Add("上课地点", typeof(string));

dt.Columns.Add("时间", typeof(string));

dt.Columns.Add("课时", typeof(string));

dt.Columns.Add("学分", typeof(string));

dt.Columns.Add("类型", typeof(string));

foreach (model.Course r1 in rs)

{

model.Teacher t = new model.Teacher();

t = tdao.selectNum2(teacher\_login.tnumber);

Console.WriteLine("2:" + r1.Course\_num);

Console.WriteLine("3:" + r1.Course\_name);

Console.WriteLine("4:" + r1.Type);

Console.WriteLine("5:" + t.Teacher\_name);

Console.WriteLine("6:" + r1.Credit\_hour);

Console.WriteLine("7:" + r1.Type);

dt.Rows.Add(r1.Course\_num, r1.Course\_name, t.Teacher\_name,r1.Address,r1.Time\_week,r1.Time\_jie, r1.Credit\_hour, r1.Type);

}

dataGridView2.DataSource = dt;

}

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

{

panel7选课.Visible = false;

panel7.Visible = false;

panel7成绩.Visible = false;

panel3课表.Visible = false;

panel7个人.Visible = false;

update\_password\_teacher t = new update\_password\_teacher();

t.Show();

}

//

学生页面主要代码：

private void 本学期课表\_Click(object sender, EventArgs e)

{

panel成绩查询.Visible = false;

panel3选课.Visible = false;

panel3学籍.Visible = false;

panel3课表.Visible = true;

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void 个人成绩查询\_Click\_1(object sender, EventArgs e)

{

panel3课表.Visible = false;

panel3学籍.Visible = false;

panel3选课.Visible = false;

panel成绩查询.Visible = true;

}

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

{

dao.resultDao dao = new dao.resultDao();

dao.courseDao cdao = new dao.courseDao();

dao.teacherDao tdao = new dao.teacherDao();

List<model.Result> rs = new List<model.Result>();

if (checkBox2成绩.Checked == false && checkBox1成绩.Checked == false)

{

rs = dao.selectAll(student\_login.number, comboBox1成绩.Text, comboBox2成绩.Text);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("代码", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("性质", typeof(String));

dt.Columns.Add("教师", typeof(string));

dt.Columns.Add("学分", typeof(String));

dt.Columns.Add("成绩", typeof(String));

dt.Columns.Add("类型", typeof(String));

dt.Columns.Add("学年", typeof(String));

dt.Columns.Add("学期", typeof(String));

foreach (model.Result r1 in rs)

{

if (r1.Score != double.NaN)

{

model.Course c = new model.Course();

c = cdao.selectNum(r1.Course\_num);

model.Teacher t = new model.Teacher();

t = tdao.selectNum(c.Teacher\_num);

dt.Rows.Add(r1.Result\_num, c.Course\_name, c.Type, t.Teacher\_name, c.Credit\_hour, r1.Score, r1.Type, r1.Year, r1.Term);

}

}

dataGridView1成绩.DataSource = dt;

}

else if (checkBox1成绩.Checked == true && checkBox2成绩.Checked == false)

{

rs = dao.select1(student\_login.number, comboBox1成绩.Text, comboBox2成绩.Text);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("代码", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("性质", typeof(String));

dt.Columns.Add("教师", typeof(string));

dt.Columns.Add("学分", typeof(String));

dt.Columns.Add("成绩", typeof(String));

dt.Columns.Add("类型", typeof(String));

dt.Columns.Add("学年", typeof(String));

dt.Columns.Add("学期", typeof(String));

foreach (model.Result r1 in rs)

{

if (r1.Score != double.NaN)

{

model.Course c = new model.Course();

c = cdao.selectNum(r1.Course\_num);

model.Teacher t = new model.Teacher();

t = tdao.selectNum(c.Teacher\_num);

dt.Rows.Add(r1.Result\_num, c.Course\_name, c.Type, t.Teacher\_name, c.Credit\_hour, r1.Score, r1.Type, r1.Year, r1.Term);

}

}

dataGridView1成绩.DataSource = dt;

}

else if (checkBox1成绩.Checked == false && checkBox2成绩.Checked == true)

{

rs = dao.select2(student\_login.number, comboBox1成绩.Text, comboBox2成绩.Text);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("代码", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("性质", typeof(String));

dt.Columns.Add("教师", typeof(string));

dt.Columns.Add("学分", typeof(String));

dt.Columns.Add("成绩", typeof(String));

dt.Columns.Add("类型", typeof(String));

dt.Columns.Add("学年", typeof(String));

dt.Columns.Add("学期", typeof(String));

foreach (model.Result r1 in rs)

{

if (r1.Score != double.NaN)

{

model.Course c = new model.Course();

c = cdao.selectNum(r1.Course\_num);

model.Teacher t = new model.Teacher();

t = tdao.selectNum(c.Teacher\_num);

dt.Rows.Add(r1.Result\_num, c.Course\_name, c.Type, t.Teacher\_name, c.Credit\_hour, r1.Score, r1.Type, r1.Year, r1.Term);

}

}

dataGridView1成绩.DataSource = dt;

}

else if (checkBox1成绩.Checked == true && checkBox2成绩.Checked == true)

{

rs = dao.select3(student\_login.number, comboBox1成绩.Text, comboBox2成绩.Text);

DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("代码", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("性质", typeof(String));

dt.Columns.Add("教师", typeof(string));

dt.Columns.Add("学分", typeof(String));

dt.Columns.Add("成绩", typeof(String));

dt.Columns.Add("类型", typeof(String));

dt.Columns.Add("学年", typeof(String));

dt.Columns.Add("学期", typeof(String));

foreach (model.Result r1 in rs)

{

if (r1.Score != double.NaN)

{

model.Course c = new model.Course();

c = cdao.selectNum(r1.Course\_num);

model.Teacher t = new model.Teacher();

t = tdao.selectNum(c.Teacher\_num);

dt.Rows.Add(r1.Result\_num, c.Course\_name, c.Type, t.Teacher\_name, c.Credit\_hour, r1.Score, r1.Type, r1.Year, r1.Term);

}

}

dataGridView1成绩.DataSource = dt;

}

}

private void 学生选课\_Click\_1(object sender, EventArgs e)

{

panel成绩查询.Visible = false;

panel3课表.Visible = false;

panel3学籍.Visible = false;

panel3选课.Visible = true;

string str = util.Util.sqlcon;

MySqlConnection conn = new MySqlConnection(str);

string sql = "select student\_academy,student\_name from student where student\_num='" + student\_login.number + "'";

conn.Open();

MySqlCommand cmd = new MySqlCommand(sql, conn);

MySqlDataReader thisread = cmd.ExecuteReader();

thisread.Read();

string temp1 = thisread.GetValue(0).ToString().Trim();

string temp2 = thisread.GetValue(1).ToString().Trim();

conn.Close();

CourseSelection cs = new CourseSelection();

cs.label2Text = temp1;

cs.label6Text = temp2;

cs.label8Text = label3传参1.Text;

cs.label7Text = student\_login.number;

cs.FormBorderStyle = FormBorderStyle.None;

cs.Dock = DockStyle.Fill;

cs.TopLevel = false;

panel3选课.Controls.Clear();

panel3选课.Controls.Add(cs);

cs.Show();

}

private void label3传参1\_Click(object sender, EventArgs e)

{

}

private void panel成绩查询\_Paint(object sender, PaintEventArgs e)

{

}

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

{

dao.resultDao dao = new dao.resultDao();

dao.courseDao cdao = new dao.courseDao();

dao.teacherDao tdao = new dao.teacherDao();

List<model.Result> rs = new List<model.Result>();

rs = dao.selectAll1(student\_login.number, comboBox1课表.Text, comboBox2课表.Text);

Console.WriteLine(comboBox1课表.Text);

Console.WriteLine(comboBox2课表.Text);

// DataSet ds = new DataSet();

DataTable dt = new DataTable("Table\_New");

dt.Columns.Add("代码", typeof(string));

dt.Columns.Add("课程编号", typeof(string));

dt.Columns.Add("课程名称", typeof(string));

dt.Columns.Add("教师", typeof(string));

dt.Columns.Add("学分", typeof(string));

dt.Columns.Add("类型", typeof(string));

dt.Columns.Add("学年", typeof(string));

dt.Columns.Add("学期", typeof(string));

foreach (model.Result r1 in rs)

{

model.Course c = new model.Course();

c = cdao.selectNum(r1.Course\_num);

model.Teacher t = new model.Teacher();

t = tdao.selectNum(c.Teacher\_num);

Console.WriteLine(r1.Result\_num);

Console.WriteLine(c.Type);

Console.WriteLine(c.Course\_num);

Console.WriteLine(c.Type);

Console.WriteLine(r1.Year);

Console.WriteLine(r1.Term);

dt.Rows.Add(r1.Result\_num, c.Course\_num, c.Course\_name, t.Teacher\_name, c.Credit\_hour, c.Type, r1.Year, r1.Term);

}

dataGridView1课表.DataSource = dt;

}

private void 学籍信息\_Click(object sender, EventArgs e)

{

panel成绩查询.Visible = false;

panel3课表.Visible = false;

panel3选课.Visible = false;

panel3学籍.Visible = true;

student\_information sAllPage = new student\_information();

sAllPage.FormBorderStyle = FormBorderStyle.None;

sAllPage.Dock = DockStyle.Fill;

sAllPage.TopLevel = false;

panel3学籍.Controls.Clear();

panel3学籍.Controls.Add(sAllPage);

sAllPage.Show();

}

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

{

update\_password u = new update\_password();

u.Show();

}

//

管理员页面主要代码：

public partial class admin\_home : Form

{

public static new Student stud;//当前窗口和修改信息窗口用来传递修改的学生信息；

public static new Course course;//当前窗口和修改信息窗口用来传递修改的课程信息；

public static new Teacher tea;//当前窗口和修改信息窗口用来传递修改的老师信息；

public int s\_count=0;//添加第几行学生信息

public int c\_count = 0;//添加第几行课程信息

public int t\_count = 0;//调加第几行教师

public static string teacher\_num;

public List<string> teacher\_nums=new List<string>();//管理员添加课程时选择的老师工号;

public admin\_home()

{

InitializeComponent();

teacher\_num = dao.teacherDao.find\_num();

panel4.Visible = false;

panel2.Visible = false;

panel7.Visible = false;

panel5.Visible = false;

panel9.Visible = false;

panel添加教师页面.Visible = false;

查看教师页面.Visible = false;

panel审核.Visible = false;

}

private void menuStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

}

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

{

}

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

{

}

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

{

bool wanzheng = true; ;//用户是否输入完整

int i = dataGridView1.RowCount;

dao.studentDao Sdao = new studentDao();

for (int j = 0; j < i-1; j++)//检查时候有未填的空

{

string Student\_num = Convert.ToString(dataGridView1.Rows[j].Cells[0].Value);//学号

string Student\_name = Convert.ToString(dataGridView1.Rows[j].Cells[1].Value);//学号

string Sex = Convert.ToString(dataGridView1.Rows[j].Cells[2].Value);//性别

string Student\_grade = Convert.ToString(dataGridView1.Rows[j].Cells[3].Value);//年纪

string Student\_academy = Convert.ToString(dataGridView1.Rows[j].Cells[4].Value);//学院

string Profession = Convert.ToString(dataGridView1.Rows[j].Cells[5].Value);//专业

string Id\_number = Convert.ToString(dataGridView1.Rows[j].Cells[7].Value);//身份证号

string Classs = Convert.ToString(dataGridView1.Rows[j].Cells[6].Value);//身份证号

if (Id\_number.Length !=18)

{

MessageBox.Show("请输入" + Student\_name + "的正确身份证号！");

wanzheng = false;

}

else if(Student\_num==""||Student\_name==""||Sex==""||Student\_grade==""||Student\_grade==""||

Student\_academy == "" || Profession == "" || Id\_number == "" || Classs == "")

{

MessageBox.Show("请输入完整信息！");

wanzheng = false;

}

}

if (wanzheng == true)

{

for (int j = 0; j < i-1; j++)

{

model.Student student = new Student();

student.Student\_num = Convert.ToString(dataGridView1.Rows[j].Cells[0].Value);//学号

student.Student\_name = Convert.ToString(dataGridView1.Rows[j].Cells[1].Value);//学号

student.Sex = Convert.ToString(dataGridView1.Rows[j].Cells[2].Value);//性别

student.Student\_grade = Convert.ToString(dataGridView1.Rows[j].Cells[3].Value);//年纪

student.Student\_academy = Convert.ToString(dataGridView1.Rows[j].Cells[4].Value);//学院

student.Profession = Convert.ToString(dataGridView1.Rows[j].Cells[5].Value);//专业

student.Id\_number = Convert.ToString(dataGridView1.Rows[j].Cells[7].Value);//身份证号

student.Classs = Convert.ToString(dataGridView1.Rows[j].Cells[6].Value);//身份证号

student.Password = student.Id\_number.Substring(student.Id\_number.Length- 6, 6);

bool isd = dao.studentDao.add\_student(student);

if (isd == false)

{

MessageBox.Show("提交失败！");

break;

}

}

MessageBox.Show("提交成功！");

dataGridView1.Rows.Clear();

s\_count = 0;

}

}

protected override bool ProcessCmdKey(ref Message msg, Keys keyData)

{

base.ProcessCmdKey(ref msg, keyData);

int WM\_KEYDOWN = 256;

int WM\_SYSKEYDOWN = 260;

bool \_disable = false;

if (msg.Msg == WM\_KEYDOWN || msg.Msg == WM\_SYSKEYDOWN)

{

switch (keyData)

{

case Keys.Enter:

SendKeys.Send("{Tab}");

\_disable = true;

break;

}

}

return \_disable;

}

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

{

if (e.ColumnIndex == 0 || e.ColumnIndex == -1)

{

string num = dao.studentDao.find\_num();

int num2 = int.Parse(num)+s\_count;

num2 = num2 + 1;

dataGridView1.CurrentRow.Cells[0].Value = num2;

s\_count++;

}

}

private void dataGridView3\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

}

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

{

dataGridView1.Rows.Clear();

s\_count = 0;

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

}

private void toolStripMenuItem3\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel4.Visible = false;

panel9.Visible = false;

panel5.Visible = false;

panel添加教师页面.Visible = false;

查看教师页面.Visible = false;

panel审核.Visible = false;

panel2.Visible = true;

List<Student> students = new List<Student>();

dataGridView2.Rows.Clear();

students = dao.studentDao.find\_students();

int i = 0;

if (students!= null)

{

foreach (Student s in students)

{

dataGridView2.Rows.Add();

dataGridView2.Rows[i].Cells[0].Value =s.Student\_num;

dataGridView2.Rows[i].Cells[1].Value = s.Student\_name;

dataGridView2.Rows[i].Cells[2].Value = s.Sex;

dataGridView2.Rows[i].Cells[3].Value = s.Student\_grade;

dataGridView2.Rows[i].Cells[4].Value = s.Student\_academy;

dataGridView2.Rows[i].Cells[5].Value = s.Profession;

dataGridView2.Rows[i].Cells[6].Value = s.Classs;

dataGridView2.Rows[i].Cells[7].Value = s.Id\_number;

i++;

}

}

dataGridView2.ReadOnly = true;

}

private void dataGridView2\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void toolStripMenuItem2\_Click(object sender, EventArgs e)

{

panel2.Visible = false;

panel4.Visible = false;

panel9.Visible = false;

panel5.Visible = false;

panel添加教师页面.Visible = false;

查看教师页面.Visible = false;

panel审核.Visible = false;

panel7.Visible = true;

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_Click(object sender, EventArgs e)

{

textBox1.Text = " ";

}

private void button4\_Click(object sender, EventArgs e)

{

Student ss = new Student();

ss.Student\_num = textBox3.Text;

ss.Student\_name = textBox4.Text;

ss.Student\_grade = comboBox1.Text;

ss.Student\_academy = comboBox2.Text;

ss.Classs = textBox2.Text;

ss.Profession = textBox1.Text;

List<Student> students = new List<Student>();

if (ss.Student\_num == "" && ss.Student\_name == "" && ss.Student\_grade == "" && ss.Student\_academy == ""

&& ss.Classs == "" && ss.Profession == "")//如果输入为空，则显示全部学生信息

{

students = dao.studentDao.find\_students();

}

else//如果输入查询条件则更具条件查询

{

students = dao.studentDao.find\_students2(ss);

}

dataGridView2.Rows.Clear();

int i = 0;

if (students != null)

{

foreach (Student s in students)

{

dataGridView2.Rows.Add();

dataGridView2.Rows[i].Cells[0].Value = s.Student\_num;

dataGridView2.Rows[i].Cells[1].Value = s.Student\_name;

dataGridView2.Rows[i].Cells[2].Value = s.Sex;

dataGridView2.Rows[i].Cells[3].Value = s.Student\_grade;

dataGridView2.Rows[i].Cells[4].Value = s.Student\_academy;

dataGridView2.Rows[i].Cells[5].Value = s.Profession;

dataGridView2.Rows[i].Cells[6].Value = s.Classs;

dataGridView2.Rows[i].Cells[7].Value = s.Id\_number;

i++;

}

}

//清空原来输入的查询条件

textBox3.Text="";

textBox4.Text="";

comboBox1.Text="";

comboBox2.Text="";

textBox2.Text="";

textBox1.Text="";

}

private void dataGridView2\_CellContentClick\_1(object sender, DataGridViewCellEventArgs e)

{

if (e.ColumnIndex == 8)

{

int i = e.RowIndex;

string student\_num = Convert.ToString(dataGridView2.Rows[i].Cells[0].Value);

MessageBox.Show("" + student\_num);

studentDao.delete(student\_num);

dataGridView2.Rows.Remove(dataGridView2.Rows[i]);

}

else if(e.ColumnIndex == 9)//查询

{

int i = e.RowIndex;

Student s = new Student();

string student\_num = Convert.ToString(dataGridView2.Rows[i].Cells[0].Value);

Form f1= new student\_update(student\_num);

f1.ShowDialog();

if (stud != null)//用户修改了信息

{

dataGridView2.Rows[i].Cells[0].Value = stud.Student\_num;

dataGridView2.Rows[i].Cells[1].Value = stud.Student\_name;//性别不能修改

dataGridView2.Rows[i].Cells[3].Value = stud.Student\_grade;

dataGridView2.Rows[i].Cells[4].Value = stud.Student\_academy;

dataGridView2.Rows[i].Cells[5].Value = stud.Profession;

dataGridView2.Rows[i].Cells[6].Value = stud.Classs;

dataGridView2.Rows[i].Cells[7].Value = stud.Id\_number;

}

}

}

private void toolStripMenuItem5\_Click(object sender, EventArgs e)

{

panel2.Visible = false;

panel7.Visible = false;

panel5.Visible = false;

panel9.Visible = false;

panel添加教师页面.Visible = false;

查看教师页面.Visible = false;

panel审核.Visible = false;

panel4.Visible = true;

}

private void dataGridView3\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

if (e.ColumnIndex ==9 )//添加课程时选择老师

{

Form g = new select\_teacher();

g.ShowDialog();

dataGridView3.CurrentRow.Cells[9].Value = teacher\_num;

teacher\_nums.Add(teacher\_num);

}

}

private void button6\_Click(object sender, EventArgs e)

{

bool wanzheng = true;//用户是否输入完整

int i = dataGridView3.RowCount;

dao.courseDao Sdao = new courseDao();

for (int j = 0; j < i - 1; j++)//检查时候有未填的空

{

string Course\_num = Convert.ToString(dataGridView3.Rows[j].Cells[0].Value);//课程号

string Course\_name = Convert.ToString(dataGridView3.Rows[j].Cells[1].Value);//课程名

string Credit\_hour = Convert.ToString(dataGridView3.Rows[j].Cells[2].Value);//学分

string Type = Convert.ToString(dataGridView3.Rows[j].Cells[3].Value);//类别

string Address = Convert.ToString(dataGridView3.Rows[j].Cells[4].Value);//地点

string Time\_week = Convert.ToString(dataGridView3.Rows[j].Cells[5].Value);//上课时间

string time\_jie = Convert.ToString(dataGridView3.Rows[j].Cells[6].Value);//上课时间

string Academy = Convert.ToString(dataGridView3.Rows[j].Cells[7].Value);//学院

string maxcount = Convert.ToString(dataGridView3.Rows[j].Cells[8].Value);//最大容量

string teacher\_num = teacher\_nums[j];//老师工号

if (Course\_num == "" || Course\_name == "" || Credit\_hour == "" || Type == "" || Address == "" ||

Time\_week == "" || time\_jie == "" || Academy == ""||maxcount==""||teacher\_num=="" )

{

MessageBox.Show("请输入完整信息！");

wanzheng = false;

}

}

if (wanzheng == true)

{

for (int j = 0; j < i-1; j++)

{

model.Course course = new Course();

course .Course\_num= Convert.ToString(dataGridView3.Rows[j].Cells[0].Value);//课程号

course.Course\_name = Convert.ToString(dataGridView3.Rows[j].Cells[1].Value);//课程名

course.Credit\_hour = Convert.ToString(dataGridView3.Rows[j].Cells[2].Value);//学分

course.Type = Convert.ToString(dataGridView3.Rows[j].Cells[3].Value);//类别

course.Address = Convert.ToString(dataGridView3.Rows[j].Cells[4].Value);//地点

course.Time\_week = Convert.ToString(dataGridView3.Rows[j].Cells[5].Value);//上课时间

course.Time\_jie = Convert.ToString(dataGridView3.Rows[j].Cells[6].Value);//上课时间

course.Academy = Convert.ToString(dataGridView3.Rows[j].Cells[7].Value);//学院

course.Maxcount = Convert.ToInt32(dataGridView3.Rows[j].Cells[8].Value);//最大容量

course.Teacher\_num = teacher\_nums[j];//老师工号

bool isd = dao.courseDao.add\_course(course);

if (isd == false)

{

MessageBox.Show("提交失败！");

break;

}

}

MessageBox.Show("提交成功！");

dataGridView3.Rows.Clear();

teacher\_nums.Clear();

c\_count = 0;

}

}

private void dataGridView3\_CellMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

}

private void dataGridView3\_CellDoubleClick\_1(object sender, DataGridViewCellEventArgs e)

{

if (e.ColumnIndex == 0 || e.ColumnIndex == -1)

{

int num2 ;

string num = dao.courseDao.find\_num();

num2 = int.Parse(num)+c\_count;

num2 = num2 + 1;

dataGridView3.CurrentRow.Cells[0].Value = num2;

c\_count++;

}

}

private void button7\_Click(object sender, EventArgs e)

{

dataGridView3.Rows.Clear();

teacher\_nums.Clear();

c\_count = 0;

}

private void button8\_Click(object sender, EventArgs e)

{

Course ss = new Course();

ss.Type = comboBox4.Text;

ss.Academy = comboBox3.Text;

ss.Time\_week = comboBox5.Text;

ss.Time\_jie = comboBox6.Text;

ss.Course\_num = textBox6.Text;

ss.Course\_name = textBox5.Text;

List<Course> courses = new List<Course>();

if (ss.Type == "" && ss.Academy == "" && ss.Time\_week == "" && ss.Time\_jie == ""

&& ss.Course\_num== "" && ss.Course\_name == "")//如果输入为空，则显示全部学生信息

{

courses = dao.courseDao.find\_courses();

}

else//如果输入查询条件则更具条件查询

{

courses = dao.courseDao.find\_courses2(ss);

}

dataGridView4.Rows.Clear();

int i = 0;

if (courses != null)

{

foreach (Course s in courses)

{

dataGridView4.Rows.Add();

dataGridView4.Rows[i].Cells[0].Value = s.Course\_num;

dataGridView4.Rows[i].Cells[1].Value = s.Course\_name;

dataGridView4.Rows[i].Cells[2].Value = s.Credit\_hour;

dataGridView4.Rows[i].Cells[3].Value = s.Type;

dataGridView4.Rows[i].Cells[4].Value = s.Address;

dataGridView4.Rows[i].Cells[5].Value = s.Time\_week;

dataGridView4.Rows[i].Cells[6].Value = s.Time\_jie;

dataGridView4.Rows[i].Cells[7].Value = s.Academy;

dataGridView4.Rows[i].Cells[8].Value = s.Maxcount;

dataGridView4.Rows[i].Cells[9].Value = s.Teacher\_num;

i++;

}

}

//清空原来输入的查询条件

comboBox4.Text="";

comboBox3.Text="";

comboBox5.Text="";

comboBox6.Text="";

textBox6.Text="";

textBox5.Text="";

}

private void toolStripMenuItem6\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel4.Visible = false;

panel2.Visible = false;

panel9.Visible = false;

panel添加教师页面.Visible = false;

查看教师页面.Visible = false;

panel审核.Visible = false;

panel5.Visible = true;

List<Course> courses = new List<Course>();

dataGridView4.Rows.Clear();

courses = dao.courseDao.find\_courses();

int i = 0;

if (courses != null)

{

foreach (Course s in courses)

{

dataGridView4.Rows.Add();

dataGridView4.Rows[i].Cells[0].Value = s.Course\_num;

dataGridView4.Rows[i].Cells[1].Value = s.Course\_name;

dataGridView4.Rows[i].Cells[2].Value = s.Credit\_hour;

dataGridView4.Rows[i].Cells[3].Value = s.Type;

dataGridView4.Rows[i].Cells[4].Value = s.Address;

dataGridView4.Rows[i].Cells[5].Value = s.Time\_week;

dataGridView4.Rows[i].Cells[6].Value = s.Time\_jie;

dataGridView4.Rows[i].Cells[7].Value = s.Academy;

dataGridView4.Rows[i].Cells[8].Value = s.Maxcount;

dataGridView4.Rows[i].Cells[9].Value = s.Teacher\_num;

i++;

}

}

//dataGridView4.ReadOnly = true;

}

private void dataGridView4\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

if (e.ColumnIndex == 10)

{

int i = e.RowIndex;

string Course\_num = Convert.ToString(dataGridView4.Rows[i].Cells[0].Value);

MessageBox.Show( Course\_num+"学生已删除!");

courseDao.delete(Course\_num);

dataGridView4.Rows.Remove(dataGridView4.Rows[i]);

}

else if (e.ColumnIndex == 11)//查询

{

int i = e.RowIndex;

Course s = new Course();

string Course\_num = Convert.ToString(dataGridView4.Rows[i].Cells[0].Value);

Form f1 = new course\_update(Course\_num);

f1.ShowDialog();

if (course != null)//用户修改了信息

{

dataGridView4.Rows[i].Cells[0].Value = course.Course\_num;

dataGridView4.Rows[i].Cells[1].Value = course.Course\_name;

dataGridView4.Rows[i].Cells[2].Value = course.Credit\_hour;

dataGridView4.Rows[i].Cells[3].Value = course.Type;

dataGridView4.Rows[i].Cells[4].Value = course.Address;

dataGridView4.Rows[i].Cells[5].Value = course.Time\_week;

dataGridView4.Rows[i].Cells[6].Value = course.Time\_jie;

dataGridView4.Rows[i].Cells[7].Value = course.Academy;

dataGridView4.Rows[i].Cells[8].Value = course.Maxcount;

dataGridView4.Rows[i].Cells[9].Value = course.Teacher\_num;

}

}

}

// private void toolStripMenuItem5\_Click(object sender, EventArgs e)

private void 发布ToolStripMenuItem\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel4.Visible = false;

panel2.Visible = false;

panel5.Visible = false;

panel添加教师页面.Visible = false;

panel审核.Visible = false;

查看教师页面.Visible = false;

panel9.Visible = true;

List<string> student\_academys = dao.studentDao.find\_academys();

string[][] class2 = new string[student\_academys.Count][];

int i = 0;

foreach (string student\_academy in student\_academys)

{

List<string> classs = dao.studentDao.find\_classs(student\_academy);

foreach (string class1 in classs)

{

class2[i] = new string[] { student\_academy, class1 };

Console.WriteLine(student\_academy + " " + class1);

i++;

}

}

}

private void treeView1\_AfterCheck(object sender, TreeViewEventArgs e)

{

if (e.Node.Nodes.Count > 0)

{

foreach (TreeNode node in e.Node.Nodes)

{

node.Checked = e.Node.Checked;

}

}

}

private void treeView1\_AfterSelect(object sender, TreeViewEventArgs e)

{

}

private void 发布ToolStripMenuItem\_Click\_1(object sender, EventArgs e)

{

panel7.Visible = false;

panel4.Visible = false;

panel2.Visible = false;

panel5.Visible = false;

panel添加教师页面.Visible = false;

查看教师页面.Visible = false;

panel审核.Visible = false;

panel9.Visible = true;

treeView2.Nodes.Clear();

treeView1.Nodes.Clear();

//学生树

List<string> student\_academys = dao.studentDao.find\_academys();

int count = dao.studentDao.find\_classCount();

string[][] class2 = new string[count][];

int i = 0;

foreach (string student\_academy in student\_academys)

{

List<string> classs = dao.studentDao.find\_classs(student\_academy);

foreach (string class1 in classs)

{

class2[i] = new string[] { student\_academy, class1 };

i++;

}

}

foreach (var data in class2)

{

string date = data[0];

string value = data[1];

TreeNode datenode = new TreeNode(date);

datenode.Name = date;

TreeNode valuenode = new TreeNode(value);

if (!treeView1.Nodes.ContainsKey(date))

{

treeView1.Nodes.Add(datenode);

treeView1.Nodes[date].Nodes.Add(valuenode);

}

else

{

treeView1.Nodes[date].Nodes.Add(valuenode);

}

}

////课程树

List<string> course\_academys = dao.courseDao.find\_academys();

count = dao.courseDao.find\_countCount();

string[][] course2 = new string[count][];

int j = 0;

foreach (string academy in course\_academys)

{

List<string> classs = dao.courseDao.find\_courses3(academy);

foreach (string class1 in classs)

{

course2[j] = new string[] { academy, class1 };

j++;

}

}

foreach (var data in course2)

{

string date = data[0];

string value = data[1];

TreeNode datenode = new TreeNode(date);

datenode.Name = date;

TreeNode valuenode = new TreeNode(value);

if (!treeView2.Nodes.ContainsKey(date))

{

treeView2.Nodes.Add(datenode);

treeView2.Nodes[date].Nodes.Add(valuenode);

}

else

{

treeView2.Nodes[date].Nodes.Add(valuenode);

}

}

}

private void treeView1\_AfterCheck\_1(object sender, TreeViewEventArgs e)

{

if (e.Node.Nodes.Count > 0)

{

foreach (TreeNode node in e.Node.Nodes)

{

node.Checked = e.Node.Checked;

}

}

}

private void treeView2\_AfterSelect(object sender, TreeViewEventArgs e)

{

}

private void treeView2\_AfterCheck(object sender, TreeViewEventArgs e)

{

if (e.Node.Nodes.Count > 0)

{

foreach (TreeNode node in e.Node.Nodes)

{

node.Checked = e.Node.Checked;

}

}

}

private void button11\_Click(object sender, EventArgs e)

{

//获取选中的班级

List<string> classs=new List<string>();

foreach (TreeNode tn in treeView1.Nodes)

{

foreach(TreeNode tn2 in tn.Nodes){

if (tn2.Checked == true)

{

classs.Add(tn2.Text);

}

}

}

//获取选中的课程号

List<string> course\_num = new List<string>();

foreach (TreeNode tn in treeView2.Nodes)

{

foreach (TreeNode tn2 in tn.Nodes)

{

if (tn2.Checked == true)

{

string num = tn2.Text;

int IndexofA = num.IndexOf('(');

int IndexofB = num.IndexOf(')');

string course\_num2 = num.Substring(IndexofA + 1, IndexofB - IndexofA - 1);

//string course\_num2 = tn2.Text.Substring(1, 3);

course\_num.Add(course\_num2);

}

}

}

//查询出该班级内的所有学号

List<string> student\_num = dao.studentDao.find\_studentNums(classs);

//讲学号以及课程号写入result表

List<Result> results = new List<Result>();

foreach (string n in student\_num)

{

foreach (string c in course\_num)

{

System.DateTime currentTime=new System.DateTime();

currentTime=System.DateTime.Now;

Result result = new Result();

result.Course\_num = c;

result.Student\_num = n;

result.Term = "春";

result.Year = currentTime.Year.ToString();

dao.resultDao.add\_result(result);

}

}

MessageBox.Show("提交成功！");

//取消节点前的复选框选中状态

foreach (TreeNode tn in treeView1.Nodes)

{

if (tn.Checked == true) tn.Checked = false;

foreach (TreeNode tn2 in tn.Nodes)

{

if (tn2.Checked == true)

{

tn2.Checked = false;

}

}

}

foreach (TreeNode tn in treeView2.Nodes)

{

if (tn.Checked == true) tn.Checked = false;

foreach (TreeNode tn2 in tn.Nodes)

{

if (tn2.Checked == true)

{

tn2.Checked = false;

}

}

}

}

private void 教师查询按钮\_Click(object sender, EventArgs e)

{

Teacher ss = new Teacher();

ss.Teacher\_num = 工号1.Text;

ss.Teacher\_name = 姓名1.Text;

ss.Teacher\_academy = 学院1.Text;

ss.Profession = 专业1.Text;

List<Teacher> teachers = new List<Teacher>();

if (ss.Teacher\_num == "" && ss.Teacher\_name == "" && ss.Teacher\_academy == "" && ss.Profession == "")//如果输入为空，则显示全部学生信息

{

teachers = dao.teacherDao.find\_teachers();

}

else//如果输入查询条件则更具条件查询

{

teachers = dao.teacherDao.find\_teachers2(ss);

}

教师显示.Rows.Clear();

int i = 0;

if (teachers != null)

{

foreach (Teacher s in teachers)

{

教师显示.Rows.Add();

教师显示.Rows[i].Cells[0].Value = s.Teacher\_num;

教师显示.Rows[i].Cells[1].Value = s.Teacher\_name;

教师显示.Rows[i].Cells[2].Value = s.Sex;

教师显示.Rows[i].Cells[3].Value = s.Teacher\_academy;

教师显示.Rows[i].Cells[4].Value = s.Profession;

教师显示.Rows[i].Cells[5].Value = s.Id\_number;

i++;

}

}

//清空原来输入的查询条件

学院1.Text = "";

专业1.Text = "";

姓名1.Text = "";

工号1.Text = "";

}

private void 查看ToolStripMenuItem\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel4.Visible = false;

panel2.Visible = false;

panel5.Visible = false;

panel添加教师页面.Visible = false;

panel9.Visible = false;

panel审核.Visible = false;

查看教师页面.Visible = true;

教师显示.Rows.Clear();

List<Teacher> teachers = new List<Teacher>();

teachers = dao.teacherDao.find\_teachers();

int i = 0;

if (teachers != null)

{

foreach (Teacher s in teachers)

{

教师显示.Rows.Add();

教师显示.Rows[i].Cells[0].Value = s.Teacher\_num;

教师显示.Rows[i].Cells[1].Value = s.Teacher\_name;

教师显示.Rows[i].Cells[2].Value = s.Sex;

教师显示.Rows[i].Cells[3].Value = s.Teacher\_academy;

教师显示.Rows[i].Cells[4].Value = s.Profession;

教师显示.Rows[i].Cells[5].Value = s.Id\_number;

i++;

}

}

}

private void 教师显示\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

if (e.ColumnIndex == 6)//删除

{

int i = e.RowIndex;

string teacher\_num = Convert.ToString(教师显示.Rows[i].Cells[0].Value);

MessageBox.Show("删除成功！" );

teacherDao.delete(teacher\_num);

教师显示.Rows.Remove(教师显示.Rows[i]);

}

else if (e.ColumnIndex == 7)//查询

{

int i = e.RowIndex;

Teacher s = new Teacher();

string teacher\_num = Convert.ToString(教师显示.Rows[i].Cells[0].Value);

Form f1 = new teacher\_update(teacher\_num);

f1.ShowDialog();

Teacher teacher = new Teacher();

teacherDao.update(teacher);

if (tea != null)//用户修改了信息

{

教师显示.Rows[i].Cells[0].Value = tea.Teacher\_num;

教师显示.Rows[i].Cells[1].Value = tea.Teacher\_name;

// 教师显示.Rows[i].Cells[2].Value = tea.Sex;//性别不能修改

教师显示.Rows[i].Cells[3].Value = tea.Teacher\_academy;

教师显示.Rows[i].Cells[4].Value = tea.Profession;

教师显示.Rows[i].Cells[5].Value = tea.Id\_number;

}

}

}

private void 添加ToolStripMenuItem\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel4.Visible = false;

panel2.Visible = false;

panel5.Visible = false;

panel9.Visible = false;

查看教师页面.Visible = false;

panel审核.Visible = false;

panel添加教师页面.Visible = true;

}

private void 提交添加老师\_Click(object sender, EventArgs e)

{

bool wanzheng = true; ;//用户是否输入完整

int i = 添加学生框.RowCount;

// MessageBox.Show("" + i);

dao.teacherDao tdao = new teacherDao();

for (int j = 0; j < i - 1; j++)//检查时候有未填的空

{

string Teacher\_num = Convert.ToString(添加学生框.Rows[j].Cells[0].Value);//学号

string Teacher\_name = Convert.ToString(添加学生框.Rows[j].Cells[1].Value);//学号

string Sex = Convert.ToString(添加学生框.Rows[j].Cells[2].Value);//性别

string Teacher\_academy = Convert.ToString(添加学生框.Rows[j].Cells[3].Value);//学院

string Profession = Convert.ToString(添加学生框.Rows[j].Cells[4].Value);//专业

string Id\_number = Convert.ToString(添加学生框.Rows[j].Cells[5].Value);//身份证号

if (Id\_number.Length != 18)

{

MessageBox.Show("请输入正确的" + Teacher\_name + "的身份证号码");

wanzheng = false;

}

else if (Teacher\_num == "" || Teacher\_name == "" || Sex == "" || Teacher\_academy == "" || Profession == "" || Id\_number == "")

{

MessageBox.Show("请输入完整信息！");

wanzheng = false;

}

}

if (wanzheng == true)

{

for (int j = 0; j < i-1; j++)

{

model.Teacher teacher = new Teacher();

teacher.Teacher\_num = Convert.ToString(添加学生框.Rows[j].Cells[0].Value);//学号

teacher.Teacher\_name = Convert.ToString(添加学生框.Rows[j].Cells[1].Value);//学号

teacher.Sex = Convert.ToString(添加学生框.Rows[j].Cells[2].Value);//性别

teacher.Teacher\_academy = Convert.ToString(添加学生框.Rows[j].Cells[3].Value);//学院

teacher.Profession = Convert.ToString(添加学生框.Rows[j].Cells[4].Value);//专业

teacher.Id\_number = Convert.ToString(添加学生框.Rows[j].Cells[5].Value);//身份证号

string id\_number = teacher.Id\_number;

teacher.Password = id\_number.Substring(id\_number.Length - 6, 6);

bool isd = dao.teacherDao.add\_teacher(teacher);

if (isd == false)

{

MessageBox.Show("提交失败！");

break;

}

}

MessageBox.Show("提交成功！");

添加学生框.Rows.Clear();

}

}

private void 清空添加老师\_Click(object sender, EventArgs e)

{

添加学生框.Rows.Clear();

}

private void panel添加教师页面\_Paint(object sender, PaintEventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void 添加学生框\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

if (e.ColumnIndex == 0 || e.ColumnIndex == -1)

{

string num = dao.teacherDao.find\_num();

int num2 = int.Parse(num) + t\_count;

num2 = num2 + 1;

添加学生框.CurrentRow.Cells[0].Value = num2;

t\_count++;

}

}

private void admin\_home\_Load(object sender, EventArgs e)

{

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void toolStripMenuItem7\_Click(object sender, EventArgs e)

{

panel7.Visible = false;

panel4.Visible = false;

panel2.Visible = false;

panel5.Visible = false;

panel9.Visible = false;

查看教师页面.Visible = false;

panel添加教师页面.Visible = false;

panel审核.Visible = true;

shenhe sAllPage = new shenhe();

sAllPage.FormBorderStyle = FormBorderStyle.None;

sAllPage.Dock = DockStyle.Fill;

sAllPage.TopLevel = false;

panel审核.Controls.Clear();

panel审核.Controls.Add(sAllPage);

sAllPage.Show();

}

private void toolStripMenuItem4\_Click(object sender, EventArgs e)

{

}

//

数据库语言主要代码：（只列部分）

课程表：

class courseDao

{

//添加一条课程信息

//

public static bool add\_course(model.Course course)

{

MySqlConnection conn = Util.getConn();

MySqlCommand command;

try

{

command = conn.CreateCommand();

command.CommandText = "INSERT INTO course(Course\_num,Course\_name,credit\_hour,type,address,time\_week,time\_jie,academy,maxcount,teacher\_num) VALUES('"

+ course.Course\_num + "','" + course.Course\_name + "','" +

course.Credit\_hour + "','" + course.Type + "','" +

course.Address + "','" + course.Time\_week + "','" +

course.Time\_jie + "','" + course.Academy + "'," +

course.Maxcount + ",'" + course.Teacher\_num + "')";

Console.WriteLine("sql:" + command.CommandText);

command.ExecuteNonQuery();

}

catch (Exception)

{

return false;

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return true;

}

//查询出最后一条数据的学生学号

public static string find\_num()//查询最大的课程号

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

string num = null;

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT max(course\_num) FROM course";

Console.WriteLine("sql:" + command.CommandText);

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

num = dataReader.GetString(0);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return num;

}

public static List<Course> find\_courses()//查询所有课程

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

List<Course> cours = new List<Course>();

string num = null;

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM course";

dataReader = command.ExecuteReader();

while (dataReader.Read())

{

model.Course course = new Course();

course.Course\_num = dataReader.GetString("Course\_num");

course.Course\_name = dataReader.GetString("Course\_name");

course.Credit\_hour = dataReader.GetString("credit\_hour");

course.Type = dataReader.GetString("type");

course.Address = dataReader.GetString("address");

course.Time\_week = dataReader.GetString("time\_week");

course.Time\_jie = dataReader.GetString("time\_jie");

course.Academy = dataReader.GetString("academy");

course.Maxcount = dataReader.GetInt16("maxcount");

course.Teacher\_num = dataReader.GetString("teacher\_num");

cours.Add(course);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return cours;

}

public static List<Course> find\_courses2(Course s)//根据条件查询

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

List<Course> courses = new List<Course>();

string num = null;

try

{

command = conn.CreateCommand();

string sql = "SELECT \* FROM course where";

bool shuru = false;//sql语句中是否已添加条件

if (s.Type != "")

{

sql = sql + " type='" + s.Type + "' ";

shuru = true;

}

if (s.Academy != "")

{

if (shuru == true) sql = sql + " and ";

sql = sql + " academy='" + s.Academy + "' ";

shuru = true;

}

if (s.Time\_jie != "")

{

if (shuru == true) sql = sql + " and ";

sql = sql + " time\_jie='" + s.Time\_jie + "' ";

shuru = true;

}

if (s.Time\_week != "")

{

if (shuru == true) sql = sql + " and ";

sql = sql + " time\_week='" + s.Time\_week + "' ";

shuru = true;

}

if (s.Course\_num != "")

{

if (shuru == true) sql = sql + " and ";

sql = sql + " Course\_num='" + s.Course\_num + "' ";

shuru = true;

}

if (s.Course\_name != "")

{

if (shuru == true) sql = sql + " and ";

sql = sql + " Course\_name='" + s.Course\_name + "' ";

shuru = true;

}

Console.Write("sql:" + sql);

command.CommandText = sql;

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

model.Course course = new Course();

course.Course\_num = dataReader.GetString("Course\_num");

course.Course\_name = dataReader.GetString("Course\_name");

course.Credit\_hour = dataReader.GetString("credit\_hour");

course.Type = dataReader.GetString("type");

course.Address = dataReader.GetString("address");

course.Time\_week = dataReader.GetString("time\_week");

course.Time\_jie = dataReader.GetString("time\_jie");

course.Academy = dataReader.GetString("academy");

course.Maxcount = dataReader.GetInt16("maxcount");

course.Teacher\_num = dataReader.GetString("teacher\_num");

courses.Add(course);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return courses;

}

public static void delete(string num)//管理员删除学生

{

MySqlConnection conn = Util.getConn();

MySqlCommand command = null;

try

{

command = conn.CreateCommand();

string sql = "delete from course where Course\_num='" + num + "'";

Console.Write("sql:" + sql);

command.CommandText = sql;

command.ExecuteNonQuery();

Console.WriteLine();

}

catch (Exception)

{

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

}

public static Course find\_course(string Course\_num)//根据学号查询学生信息

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

model.Course course = new Course();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM course where Course\_num='" + Course\_num + "'";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

course.Course\_num = dataReader.GetString("Course\_num");

course.Course\_name = dataReader.GetString("Course\_name");

course.Credit\_hour = dataReader.GetString("credit\_hour");

course.Type = dataReader.GetString("type");

course.Address = dataReader.GetString("address");

course.Time\_week = dataReader.GetString("time\_week");

course.Time\_jie = dataReader.GetString("time\_jie");

course.Academy = dataReader.GetString("academy");

course.Maxcount = dataReader.GetInt16("maxcount");

course.Teacher\_num = dataReader.GetString("teacher\_num");

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return course;

}

public static void update(Course s)//修改学生信息

{

MySqlConnection conn = Util.getConn();

MySqlCommand command = null;

try

{

command = conn.CreateCommand();

string sql = "update course set Course\_name='" + s.Course\_name + "',credit\_hour='" +

s.Credit\_hour + "',type='" + s.Type + "',address='" +

s.Address + "',time\_week='" + s.Time\_week + "',time\_jie='" + s.Time\_jie + "',academy='" +

s.Academy + "',maxcount=" + s.Maxcount + ",teacher\_num='"+s.Teacher\_num+"' where Course\_num='" + s.Course\_num + "'";

Console.Write("sql:" + sql);

command.CommandText = sql;

command.ExecuteNonQuery();

Console.WriteLine();

}

catch (Exception)

{

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

}

public static List<string> find\_academys()//根据院名查询出所有的学院

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

model.Course course = new Course();

List<string> academys = new List<string>();

try

{

command = conn.CreateCommand();

command.CommandText = "select distinct academy from course ";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

string academy = dataReader.GetString("academy");

academys.Add(academy);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return academys;

}

public static List<string> find\_courses3(string academy)//根据院名查询出所有的班级

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

List<string> coueses = new List<string>(); ;//一共有18个学院

try

{

command = conn.CreateCommand();

command.CommandText = "select Course\_name,Course\_num from course where academy='" + academy + "' and type='必修'";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

string couese1 = dataReader.GetString("Course\_name");

string couese2 = dataReader.GetString("Course\_num");

couese1 = "("+couese2+")"+couese1;

coueses.Add(couese1);

}

if (!dataReader.IsClosed)

{

dataReader.Close();

}

}

catch (Exception)

{

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return coueses;

}

public static int find\_countCount()//查出班级的总个数

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

model.Student student = new Student();

int count = 0;//一共有18个学院

try

{

command = conn.CreateCommand();

command.CommandText = "select count(Course\_name) from course where type='必修'";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

count = dataReader.GetInt16(0);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return count;

}

//姚雅丽

public Course selectNum(string number)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Course r = new Course();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM course where Course\_num ='" + number + "'";

dataReader = command.ExecuteReader();

Console.WriteLine(command.CommandText);

while (dataReader.Read())

{

r.Course\_num = dataReader.GetString(0);

r.Course\_name = dataReader.GetString(1);

r.Credit\_hour = dataReader.GetString(2);

r.Type = dataReader.GetString(3);

r.Address = dataReader.GetString(4);

r.Time\_week = dataReader.GetString(5);

r.Time\_jie = dataReader.GetString(6);

r.Academy = dataReader.GetString(7);

r.Teacher\_num = dataReader.GetString(9);

return r;

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return null;

}

public List<Course> selecttc(string teacher\_num)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Course r = null;

List<Course> rs = new List<model.Course>();

try

{

command = conn.CreateCommand();

string sql = "SELECT \* FROM course where teacher\_num='" + teacher\_num + "'";

Console.WriteLine("sql:" + sql);

command.CommandText = sql;

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

r = new Course();

r.Course\_num = dataReader.GetString(0);

r.Course\_name = dataReader.GetString(1);

r.Credit\_hour = dataReader.GetString(2);

r.Type = dataReader.GetString(3);

r.Address = dataReader.GetString(4);

r.Time\_week = dataReader.GetString(5);

r.Time\_jie = dataReader.GetString(6);

r.Academy = dataReader.GetString(7);

r.Teacher\_num = dataReader.GetString(9);

Console.WriteLine("aaa:" + dataReader.GetString(0));

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

//寇肖萌部分

public List<Course> selectNum1(string number)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Course r = new Course();

List<Course> rs = new List<model.Course>();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM course where teacher\_num ='" + number + "'";

dataReader = command.ExecuteReader();

Console.WriteLine(command.CommandText);

while (dataReader.Read())

{

r = new Course();

r.Course\_num = dataReader.GetString(0);

r.Course\_name = dataReader.GetString(1);

r.Credit\_hour = dataReader.GetString(2);

r.Type = dataReader.GetString(3);

r.Address = dataReader.GetString(4);

r.Time\_week = dataReader.GetString(5);

r.Time\_jie = dataReader.GetString(6);

r.Academy = dataReader.GetString(7);

r.Teacher\_num = dataReader.GetString(9);

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

学生表：

class studentDao//对user进行的数据库操作

{

public static string find\_password(string student\_num)//查询出登录者的密码

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

string password=null;

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT password FROM student where student\_num=" + student\_num;

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

password = dataReader.GetString(0);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return password;

}

//添加一条学生信息

//

public static bool add\_student(model.Student student)

{

MySqlConnection conn = Util.getConn();

MySqlCommand command;

try

{

command = conn.CreateCommand();

command.CommandText = "INSERT INTO student(student\_num,student\_name,student\_academy,student\_grade,password,sex,profession,id\_number,classs) VALUES('"

+ student.Student\_num + "','" + student.Student\_name + "','" +

student.Student\_academy + "','" + student.Student\_grade + "','" +

student.Password + "','" + student.Sex + "','" +

student.Profession + "','" + student.Id\_number + "','"+

student.Classs+"')";

Console.WriteLine("sql:" + command.CommandText);

command.ExecuteNonQuery();

}

catch (Exception)

{

return false;

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return true;

}

//查询出最后一条数据的学生学号

public static string find\_num()//查询出登录者的密码

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

string num = null;

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT max(student\_num) FROM student";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

num = dataReader.GetString(0);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return num;

}

public static List<Student> find\_students()//查询所有学生

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

List<Student> studs = new List<Student>();

string num = null;

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM student";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

model.Student student=new Student();

student.Student\_num=dataReader.GetString("student\_num");

student.Student\_name=dataReader.GetString("student\_name");

student.Student\_academy=dataReader.GetString("student\_academy");

student.Student\_grade=dataReader.GetString("student\_grade");

student.Classs=dataReader.GetString("classs");

student.Profession= dataReader.GetString("profession");

student.Id\_number = dataReader.GetString("id\_number");

student.Password= dataReader.GetString("password");

student.Sex=dataReader.GetString("sex");

studs.Add(student);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return studs;

}

public static List<Student> find\_students2(Student s)//根据条件查询

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

List<Student> studs = new List<Student>();

string num = null;

try

{

command = conn.CreateCommand();

string sql="SELECT \* FROM student where";

bool shuru=false;//sql语句中是否已添加条件

if (s.Student\_num != "")

{

sql = sql + " student\_num='" + s.Student\_num + "' ";

shuru=true;

}

if (s.Student\_name != "")

{

if (shuru == true) sql = sql + " and ";

sql = sql + " student\_name='" + s.Student\_name + "' ";

shuru = true;

}

if (s.Student\_grade != "")

{

if (shuru == true) sql = sql + " and ";

sql = sql + " student\_grade='" + s.Student\_grade + "' ";

shuru = true;

}

if (s.Student\_academy != "")

{

if (shuru == true) sql = sql + " and ";

sql = sql + " student\_academy='" + s.Student\_academy + "' ";

shuru = true;

}

if (s.Profession != "")

{

if (shuru == true) sql = sql + " and ";

sql = sql + " profession='" + s.Profession + "' ";

shuru = true;

}

if (s.Classs != "")

{

if (shuru == true) sql = sql + " and ";

sql = sql + " classs='" + s.Classs + "' ";

shuru = true;

}

Console.Write("sql:" + sql);

command.CommandText =sql;

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

model.Student student = new Student();

student.Student\_num = dataReader.GetString("student\_num");

student.Student\_name = dataReader.GetString("student\_name");

student.Student\_academy = dataReader.GetString("student\_academy");

student.Student\_grade = dataReader.GetString("student\_grade");

student.Classs = dataReader.GetString("classs");

student.Profession = dataReader.GetString("profession");

student.Id\_number = dataReader.GetString("id\_number");

student.Password = dataReader.GetString("password");

student.Sex = dataReader.GetString("sex");

studs.Add(student);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return studs;

}

public static void delete(string num)//管理员删除学生

{

MySqlConnection conn = Util.getConn();

MySqlCommand command = null;

try

{

command = conn.CreateCommand();

string sql = "delete from student where student\_num='"+num+"'";

Console.Write("sql:" + sql);

command.CommandText = sql;

command.ExecuteNonQuery();

Console.WriteLine();

}

catch (Exception)

{

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

}

public static Student find\_student(string student\_num)//根据学号查询学生信息

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

model.Student student = new Student();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM student where student\_num='"+student\_num+"'";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

student.Student\_num = dataReader.GetString("student\_num");

student.Student\_name = dataReader.GetString("student\_name");

student.Student\_academy = dataReader.GetString("student\_academy");

student.Student\_grade = dataReader.GetString("student\_grade");

student.Classs = dataReader.GetString("classs");

student.Profession = dataReader.GetString("profession");

student.Id\_number = dataReader.GetString("id\_number");

student.Password = dataReader.GetString("password");

student.Sex = dataReader.GetString("sex");

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return student;

}

public static void update(Student s)//修改学生信息

{

MySqlConnection conn = Util.getConn();

MySqlCommand command = null;

try

{

command = conn.CreateCommand();

string sql = "update student set student\_name='"+s.Student\_name+"',student\_academy='"+

s.Student\_academy+"',student\_grade='"+s.Student\_grade+"',password='"+s.Password+"',profession='"+

s.Profession + "',id\_number='" + s.Id\_number + "',classs='" + s.Classs + "' where student\_num='"+

s.Student\_num+"'";

Console.Write("sql:" + sql);

command.CommandText = sql;

command.ExecuteNonQuery();

Console.WriteLine();

}

catch (Exception)

{

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

}

public static List<string> find\_classs(string academy)//根据院名查询出所有的班级

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

model.Student student = new Student();

List<string> classs = new List<string>(); ;//一共有18个学院

try

{

command = conn.CreateCommand();

command.CommandText = "select distinct classs from student where student\_academy='" + academy + "'";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

string class1 = dataReader.GetString("classs");

classs.Add(class1);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return classs;

}

public static List<string> find\_academys()//根据院名查询出所有的学院

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

model.Student student = new Student();

List<string> classs = new List<string>(); ;//一共有18个学院

try

{

command = conn.CreateCommand();

command.CommandText = "select distinct student\_academy from student ";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

string student\_academy = dataReader.GetString("student\_academy");

classs.Add(student\_academy);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return classs;

}

public static int find\_classCount()//查出班级的总个数

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

model.Student student = new Student();

int count = 0;//一共有18个学院

try

{

command = conn.CreateCommand();

command.CommandText = "select count(distinct classs) from student ";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

count = dataReader.GetInt16(0);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return count;

}

public static List<string> find\_studentNums(List<string> classs)//查询所有学生

{

MySqlConnection conn = Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

List<string> stud\_nums = new List<string>();

string num = null;

try

{

foreach (string c in classs)

{

command = conn.CreateCommand();

command.CommandText = "SELECT student\_num FROM student where classs='" + c + "'";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

string s\_n = dataReader.GetString("student\_num");

stud\_nums.Add(s\_n);

}

if (!dataReader.IsClosed)

{

dataReader.Close();

}

}

}

catch (Exception)

{

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return stud\_nums;

}

//根据学生编号查询学生信息

public List<Student> selectStudent(List<Result> student)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

List<Student> sts = new List<Student>();

Console.WriteLine("list:" + student.Count);

try

{

foreach (Result s1 in student)

{

command = conn.CreateCommand();

string sql = "SELECT student\_name,id\_number,sex,student\_academy,profession,student\_grade,classs FROM student where student\_num='" + s1.Student\_num + "'";

Console.WriteLine(sql);

command.CommandText = sql;

dataReader = command.ExecuteReader();

while (dataReader.Read())

{

Student s = new Student();

s.Student\_name = dataReader.GetString("student\_name");

s.Student\_academy = dataReader.GetString("student\_academy");

s.Student\_grade = dataReader.GetString("student\_grade");

s.Sex = dataReader.GetString("sex");

s.Profession = dataReader.GetString("profession");

s.Id\_number = dataReader.GetString("id\_number");

s.Classs = dataReader.GetString("classs");

sts.Add(s);

}

if (!dataReader.IsClosed)

{

dataReader.Close();

}

}

}

catch (Exception)

{

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return sts;

}

//根据学号查询学生信息

public Student selectNum(string number)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Student r = new Student();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM student where student\_num ='" + number + "'";

Console.WriteLine("aasql:" + command.CommandText);

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

r.Student\_num = dataReader.GetString("student\_num");

r.Student\_name = dataReader.GetString("student\_name");

r.Sex = dataReader.GetString("sex");

Console.WriteLine(r.Student\_num + "a " + r.Student\_name + " " + r.Sex);

return r;

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return null;

}

结果表：

class resultDao

{

public static bool add\_result(model.Result result)

{

MySqlConnection conn = Util.getConn();

MySqlCommand command;

try

{

command = conn.CreateCommand();

command.CommandText = "INSERT INTO result(student\_num,course\_num,year,term,state) VALUES('"

+ result.Student\_num + "','" + result.Course\_num + "','" +

result.Year + "','" + result.Term + "','审核已通过')";

Console.WriteLine("sql:" + command.CommandText);

command.ExecuteNonQuery();

}

catch (Exception)

{

return false;

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return true;

}

//根据课程号查询学生号

public List<Result> selectStudent(string course\_num, string year, string term)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Result r = null;

List<Result> rs = new List<model.Result>();

try

{

command = conn.CreateCommand();

string sql = "SELECT student\_num FROM result where course\_num='" + course\_num + "' and year='" + year + "' and term='" + term + "' and state='审核已通过'" ;

Console.WriteLine(sql);

command.CommandText = sql;

dataReader = command.ExecuteReader();

while (dataReader.Read())

{

r = new Result();

r.Student\_num = dataReader.GetString(0);

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

//姚雅丽代码部分

public List<Result> selectAll(string number, string year, string term)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Result r = null;

List<Result> rs = new List<model.Result>();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM result where student\_num='" + number + "' and year='" + year + "' and term='" + term + "'";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

r = new Result();

r.Result\_num = dataReader.GetString(0);

r.Student\_num = dataReader.GetString(1);

//r.Teacher\_num = dataReader.GetString(2);

r.Course\_num = dataReader.GetString(2);

r.Score = dataReader.GetDouble(3);

r.Type = dataReader.GetString(4);

r.Year = dataReader.GetString(5);

r.Term = dataReader.GetString(6);

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

//查询未通过课程

public List<Result> select1(string s\_num, string year, string term)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Result r = null;

List<Result> rs = new List<model.Result>();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM result where score<60 and year='" + year + "' and term='" + term + "'";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

r = new Result();

r.Result\_num = dataReader.GetString(0);

r.Student\_num = dataReader.GetString(1);

//r.Teacher\_num = dataReader.GetString(2);

r.Course\_num = dataReader.GetString(2);

r.Score = dataReader.GetDouble(3);

r.Type = dataReader.GetString(4);

r.Year = dataReader.GetString(5);

r.Term = dataReader.GetString(6);

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

//查询最高分

public List<Result> select2(string s\_num, string year, string term)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Result r = null;

List<Result> rs = new List<model.Result>();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM result where score=(select max(score) from result) and year='" + year + "' and term='" + term + "' and student\_num='" + s\_num + "'";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

r = new Result();

r.Result\_num = dataReader.GetString(0);

r.Student\_num = dataReader.GetString(1);

//r.Teacher\_num = dataReader.GetString(2);

r.Course\_num = dataReader.GetString(2);

r.Score = dataReader.GetDouble(3);

r.Type = dataReader.GetString(4);

r.Year = dataReader.GetString(5);

r.Term = dataReader.GetString(6);

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

public List<Result> select3(string s\_num, string year, string term)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Result r = null;

List<Result> rs = new List<model.Result>();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM result where score=(select max(score) from result) and year='" + year + "' and term='" + term + "' and student\_num='" + s\_num + "' and score<60";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

r = new Result();

r.Result\_num = dataReader.GetString(0);

r.Student\_num = dataReader.GetString(1);

//r.Teacher\_num = dataReader.GetString(2);

r.Course\_num = dataReader.GetString(2);

r.Score = dataReader.GetDouble(3);

r.Type = dataReader.GetString(4);

r.Year = dataReader.GetString(5);

r.Term = dataReader.GetString(6);

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

public List<Result> selectTeacher(string teacher\_num)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Result r = null;

List<Result> rs = new List<model.Result>();

try

{

command = conn.CreateCommand();

string sql = "SELECT \* FROM result where teacher\_num='" + teacher\_num + "'";

Console.WriteLine("sql:" + sql);

command.CommandText = sql;

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

r = new Result();

r.Result\_num = dataReader.GetString(0);

r.Student\_num = dataReader.GetString("student\_num");

//r.Teacher\_num = dataReader.GetString(2);

r.Course\_num = dataReader.GetString("course\_num");

//r.Score = dataReader.GetDouble(4);

//r.Type = dataReader.GetString(5);

r.Year = dataReader.GetString("year");

r.Term = dataReader.GetString("term");

Console.WriteLine("aaa:" + dataReader.GetString(0));

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

//根据课程号查询成绩表

public List<Result> selectResult(string course\_num)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Result r = null;

List<Result> rs = new List<model.Result>();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM result where course\_num='" + course\_num + "' and state='审核已通过' and score is null";

Console.WriteLine(command.CommandText);

dataReader = command.ExecuteReader();

Console.WriteLine(dataReader);

while (dataReader.Read())

{

r = new Result();

r.Result\_num = dataReader.GetString(0);

r.Student\_num = dataReader.GetString("student\_num");

//r.Teacher\_num = dataReader.GetString(2);

r.Course\_num = dataReader.GetString("course\_num");

//r.Score = dataReader.GetDouble(4);

//r.Type = dataReader.GetString(5);

r.Year = dataReader.GetString("year");

r.Term = dataReader.GetString("term");

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

//修改成绩

public void updateScore(string student\_num, string course\_num, string type, double score)

{

MySqlConnection conn = util.Util.getConn();

MySqlCommand command;

try

{

command = conn.CreateCommand();

command.CommandText = "update result set score=" + score + ", type='" + type + "' where student\_num='" + student\_num + "' and course\_num='" + course\_num + "'";

Console.Write(command.CommandText);

command.ExecuteNonQuery();

}

catch (Exception)

{

Console.WriteLine("添加失败！");

}

finally

{

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

}

//寇肖萌

public List<Result> selectAll1(string number, string year, string term)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Result r = null;

List<Result> rs = new List<model.Result>();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM result where student\_num='" + number + "' and year='" + year + "' and term='" + term + "'and state='审核已通过'";

dataReader = command.ExecuteReader();

Console.WriteLine();

while (dataReader.Read())

{

r = new Result();

r.Result\_num = dataReader.GetString(0);

r.Student\_num = dataReader.GetString(1);

// r.Teacher\_num = dataReader.GetString(2);

r.Course\_num = dataReader.GetString(2);

r.Year = dataReader.GetString(5);

r.Term = dataReader.GetString(6);

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

public List<Result> selectAll2(string year, string term)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Result r = null;

List<Result> rs = new List<model.Result>();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM result where year='" + year + "' and term='" + term + "'";

dataReader = command.ExecuteReader();

Console.WriteLine(command.CommandText);

while (dataReader.Read())

{

r = new Result();

r.Result\_num = dataReader.GetString(0);

r.Student\_num = dataReader.GetString(1);

r.Course\_num = dataReader.GetString(2);

r.Year = dataReader.GetString(5);

r.Term = dataReader.GetString(6);

rs.Add(r);

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return rs;

}

public Course selectNum1(string number)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Course r = new Course();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM course where Course\_num ='" + number + "'";

dataReader = command.ExecuteReader();

Console.WriteLine(command.CommandText);

while (dataReader.Read())

{

r.Course\_num = dataReader.GetString(0);

r.Course\_name = dataReader.GetString(1);

r.Credit\_hour = dataReader.GetString(2);

r.Type = dataReader.GetString(3);

r.Address = dataReader.GetString(4);

r.Time\_week = dataReader.GetString(5);

r.Time\_jie = dataReader.GetString(6);

r.Academy = dataReader.GetString(7);

r.Teacher\_num = dataReader.GetString(9);

return r;

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return null;

}

public Teacher selectNum2(string number)

{

MySqlConnection conn = util.Util.getConn();

MySqlDataReader dataReader = null;

MySqlCommand command = null;

Teacher r = new Teacher();

try

{

command = conn.CreateCommand();

command.CommandText = "SELECT \* FROM teacher where Teacher\_num ='" + number + "'";

dataReader = command.ExecuteReader();

Console.WriteLine(command.CommandText);

while (dataReader.Read())

{

r.Teacher\_num = dataReader.GetString(0);

r.Teacher\_name = dataReader.GetString(1);

return r;

}

}

catch (Exception)

{

}

finally

{

if (!dataReader.IsClosed)

{

dataReader.Close();

}

if (conn.State == ConnectionState.Open)

{

conn.Close();

}

}

return null;

}