**\*\*\*\*\*\*\*\*\*大学**

**数学与计算机学院**

**2019-2020学年第一学期**

高级开发技术

**题 目： 学生成绩管理系统**

**专 业： 软件工程**

**班 级：**

**学 号：**

**姓 名： 范世纪**

**指导老师：**

**2019年　12月　6日**

# 系统简介

## 系统概述

（1）学生身份进入系统可实现课程信息查询、成绩查询、教学评价、选课、退课、个人信息维护等操作。

（2）教师身份进入系统可实现学生信息查询、课程信息查询、学生成绩管理、学生评教查询等操作。

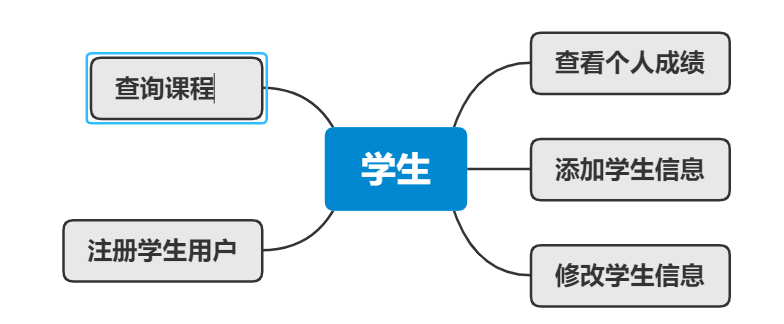
（3）教学秘书进入系统可查询学生信息、教师信息查询、学生成绩分析查询、教学评价查询、课程信息维护、开放选课等操作。

（4）系统管理员可实现用户管理、用户信息维护等功能。

## **功能模块说明**

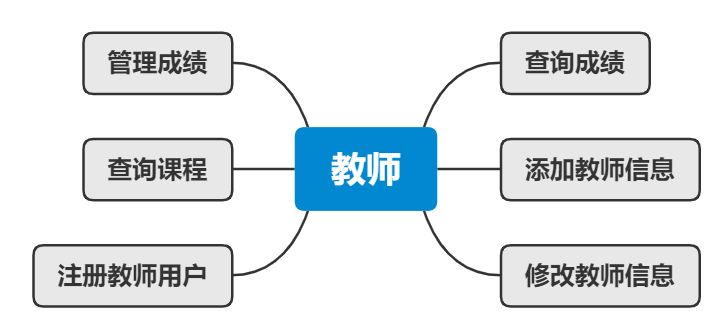
#### 2.1 学生功能模块

**学生功能模块**供学生权限使用，学生用户使用用户名以及密码登录系统。学生用户类型为“s”，以此与教师、教学秘书和系统管理员权限区分开。学生角色登录系统后可实现查询课程、查询个人成绩、查询个人信息、修改个人信息等功能。下图为学生功能模块设计。



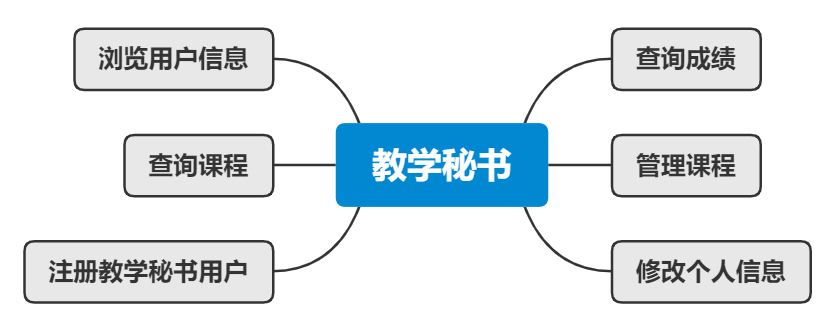
#### 2.2 教师管理模块

**教师功能模块**供教师权限使用，教师用户使用用户名以及密码登录系统。教师用户类型为“t”，以此与学生、教学秘书和系统管理员权限区分开。教师角色登录系统后可实现查询课程、查询学生成绩、查询个人信息、修改个人信息等功能。下图为教师功能模块设计。



#### 2.3 教学秘书管理

**教学秘书功能模块**可供教学秘书权限使用。教学秘书用户名以及密码登录系统，在录入的同时注意教学秘书用户类型为“ts”，以此与学生、教师和系统管理员权限区分开。教学秘书权限进入系统可进行查询学生信息、查询课程、管理课程、查询学生或教师用户、查询学生成绩、修改个人信息等功能。教学秘书模块设计如下图所示。



#### 2.4 系统管理员管理

**系统管理员功能模块**可供系统管理员权限使用。系统管理员的用户名可由系统提前录入数据库中，在录入的同时注意系统管理员用户类型为“a”，以此与学生、教师和教学秘书权限区分开。系统管理员权限进入系统可**新增用户、审核用户**。

## **数据库结构**

#### 3.1登录信息表



#### 3.2用户信息表



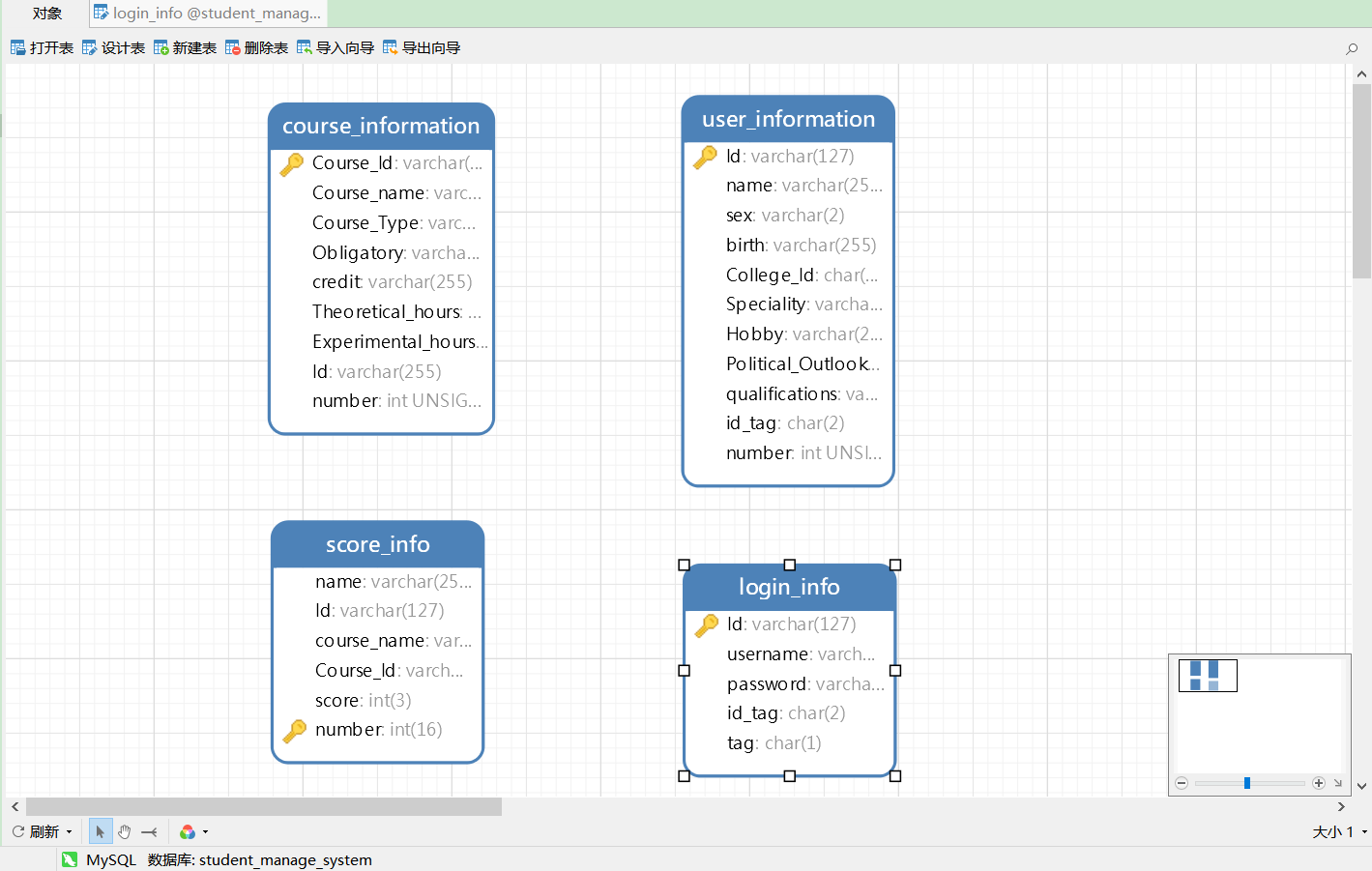
#### 3.3课程信息表



#### 3.4成绩信息表



#### 3.5数据库结构总览



# **功能实现**

## 登录界面

#### 1.1初始登录界面

采用验证码核对输入信息，字符串匹配正确，登录按钮才可用

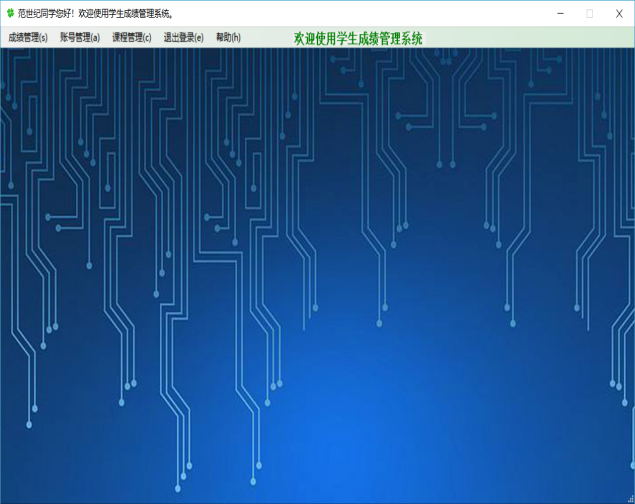
用户名不存在实时提示，密码错误实时提示



#### 1.2登录成功界面（区分管理员和普通用户）

普通用户：成绩管理、课程管理、信息管理

管理员：审核用户、添加用户



（普通用户界面） （管理员界面）

**核心代码：**

namespace ScoreManagement.FormApplication

{

public partial class login : Form

{

public login()

{

InitializeComponent();

}

//登录窗口初始化

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

{

label6.Text = "";

identity.SelectedIndex = 0; //登录系统的身份默认是学生

MaximizeBox = false; //关闭登录窗口的最大化功能

Verification\_Code.Text= Verification.Code(); //动态生成验证码

DoLogin.Enabled = false; //默认在不输入验证码时无法点击登录按钮

Globel.username = Tusername.Text;

}

//实时验证用户输入验证码是否正确

private void Verification\_Code\_in\_KeyUp(object sender, EventArgs e)

{

if (Verification\_Code\_in.Text != Verification\_Code.Text)

{

label6.Text = "验证码错误！";

}

else

{

label6.Text = "";

DoLogin.Enabled = true;

}

}

//验证用户登录

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

{

DBlink db = new DBlink();//创建数据库连接对象

Identification.identification(identity.SelectedIndex);//根据用户选择的值判断用户身份

if (db.DBconn()) {

//数据库连接成功

if (db.GetLoginData("select \* from login\_info where id\_tag='" + Globel.id\_tag + "'and username='" + Tusername.Text + "'"))

{

//用户存在

if (int.Parse(LoginInfo.tag) == 1)//账号已被审核允许登录

{

if (Tpass.Text.Equals(LoginInfo.password))//登录成功

{

Globel.username = Tusername.Text;

new MainMenu().ShowDialog();//显示主菜单界面

this.Hide();

this.Visible = false;

}

else { //密码错误

label6.Text = "密码错误！";

}

}

else { //账号未审核

label6.Text = "该账号未审核，请联系管理员";

}

}

else {

label6.Text = "登录失败，用户名不存在！";

}

}

db.DBclose();//关闭数据库连接

}

}

}

**动态生成验证码：**

namespace ScoreManagement.Tools

{

class Verification

{

//随机生成验证码

public static string Code()

{

string code = ""; //验证码字符串

Random ran = new Random();

for (int i = 0; i < 6; i++) //随机生成六位验证码功能

{

int n = ran.Next(9);

code += n;

}

return code;

}

}

}

## 注册界面（区分不同用户）

选择身份下拉选项框，默认为学生，选择不同的身份有相应的提示

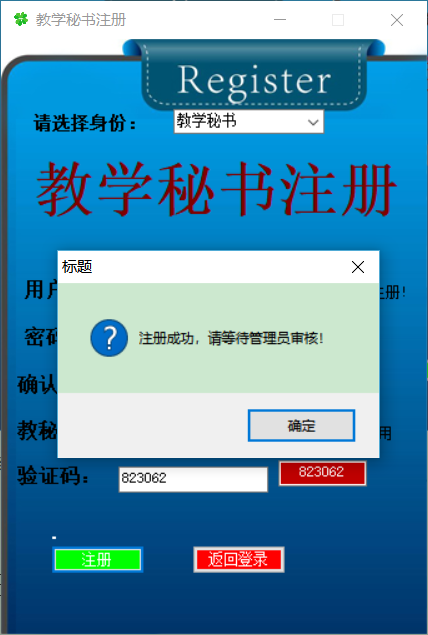
实时核对并提示是否能注册，用户名，编号不能重复

确认密码用KeyUp事件核对

#### 2.1注册主界面



#### 2.1注册成功



**核心代码：**

namespace ScoreManagement.FormApplication

{

//用户注册界面

public partial class regist : Form

{

public regist()

{

InitializeComponent();

}

//注册界面初始化

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

{

MaximizeBox = false; //关闭注册窗口的最大化功能

RegistIdentity.SelectedIndex = 3;//设置下拉菜单的默认选项

//将所有提示信息的lable置空

label8 .Text= "\*必填";

label9.Text = "\*必填";

label11.Text = "\*必填";

label12.Text = "";

label14.Text = "\*必填";

button1.Enabled = false; //用户未输入注册信息时，默认注册按钮不可点击

label2.Text = "选择身份";

this.Text = "注册";

Verification\_Code.Text = Verification.Code(); //加载验证码

panel1.Visible = false; //未选择注册身份之前默认不显示所有输入框

}

//当用户输入用户名完成时验证用户名是否可以注册

private void RGname\_Leave(object sender, EventArgs e)

{

DBlink db = new DBlink(); //创建数据库连接对象

if (db.DBconn()) { //创建数据库连接

string str = "select \* from login\_info where id\_tag='" + Globel.id\_tag + "'and username='" + RGname.Text + "'";

if (db.GetLoginData(str))//执行sql检测该用户是否已经被注册

{

label9.Text = "该用户已经存在!";

}

else {

label9.Text = "恭喜，可以注册!";

}

}

db.DBclose(); //关闭数据库连接

}

//当第二次输入密码被弹起时验证密码是否一致

private void RGpass2\_KeyUp(object sender, KeyEventArgs e)

{

if (!RGpass.Text.Equals(RGpass2.Text))

{

label8.Text = "密码不一致！";

}

else

{

label8.Text = "";

}

}

//当第二次输入密码被弹起时验证密码是否一致

private void code\_KeyUp(object sender, KeyEventArgs e)

{

if (!code.Text.Equals(Verification\_Code.Text))

{

label12.Text = "验证码错误！";

button1.Enabled = false;

}

else

{

label12.Text = "";

button1.Enabled = true;

}

}

//注册事件

private void button1\_Click\_1(object sender, EventArgs e)

{

DBlink db = new DBlink(); //创建数据库连接对象

if (db.DBconn()) { //连接数据库

string str ="insert into login\_info(Id,username,password,id\_tag) values('"+RGnumber.Text + "','"+RGname.Text + "','"+RGpass.Text + "','"+ Globel.id\_tag + "');";

string info = "注册成功，请等待管理员审核！";

//如果是管理员执行的添加用户时tag直接为1,即不用审核

if (LoginInfo.isadmin)

{

info = "添加用户成功，该用户可以直接登录！";

str = "insert into login\_info(Id,username,password,id\_tag,tag) values('" + RGnumber.Text + "','" + RGname.Text + "','" + RGpass.Text + "','"+Globel.id\_tag+"','" + 1 + "');";

}

if (db.UpdataDeletAdd(str)) {

DialogResult dr = MessageBox.Show(info, "标题", MessageBoxButtons.OK, MessageBoxIcon.Question);

if (dr == DialogResult.OK)

{

this.Visible = false;//隐藏当前窗口

}

}

}

db.DBclose();//关闭数据连接

}

## 成绩管理

#### 3.1学生界面

仅显示个人成绩

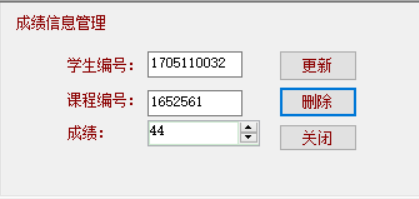


#### 3.2老师界面

可实现录入成绩、删除成绩和查询成绩，默认显示全部成绩项

查询学生编号为1705110032，且课程名为数据库的成绩

录入学生编号1705110032，课程编号为165261，成绩为44，并刷新控件DateGridView，实时显示成绩变化



删除学生编号1705110032，课程编号为165261的数据项

#### 3.3教学秘书界面（仅可用查询功能，不再赘述）

**核心代码：**

DataSet ds;

private void BindsocreData(string sql)

{

MySqlConnection con = new MySqlConnection();

con.ConnectionString = "server = localhost; uid = root; pwd = 052398; database = student\_manage\_system;";//连接字符串

Boolean DBtag = false;

try

{

con.Open();

DBtag = true;

if (DBtag)

{

MySqlCommand mySqlCommand = new MySqlCommand(sql, con);// 创建SqlCommand

MySqlDataAdapter dat = new MySqlDataAdapter(mySqlCommand); // 创建SqlDataAdapter

ds = new DataSet();

dat.Fill(ds);

dataGridView3.DataSource = ds.Tables[0];

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "error");

}

con.Close();

}

private void 成绩管理ToolStripMenuItem\_Click(object sender, EventArgs e)

{

panel3.Visible = false;

panel4.Visible = true; //隐藏成绩与课程信息

DBlink db = new DBlink(); //创建数据库连接对象

if (db.DBconn())

{ //创建数据库连接

string Str1 = "select Id from login\_info WHERE username = '" + Globel.username + "'";

Globel.Id = db.Getstringsearch(Str1);

}

db.DBclose(); //关闭数据库连接

string sql = "SELECT score\_info.Id as 学号, score\_info.name as 姓名,score\_info.course\_name as 课程,score\_info.score as 成绩 FROM score\_info";

if (LoginInfo.id\_tag.Equals("s"))

{

BindsocreData("select score\_info.Id as 学号, score\_info.name as 姓名,score\_info.course\_name as 课程,score\_info.score as 成绩 from score\_info where Id = '" + Globel.Id + "'");

}

else if(LoginInfo.id\_tag.Equals("t"))

{

BindsocreData(sql);

panel5.Visible = true;//教师可以管理成绩

panel6.Visible = true;//教师可以查询成绩

}

else if (LoginInfo.id\_tag.Equals("ts"))

{

BindsocreData(sql);

panel6.Visible = true;//教学秘书可以查询成绩

}

}

//删除成绩

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

{

ScoreInfo.id = textBox6.Text;

ScoreInfo.course\_id = textBox7.Text;

string delsql = "delete from score\_info where Id= '" + ScoreInfo.id + "'and Course\_Id = '" + ScoreInfo.course\_id + "'";

DBlink db4 = new DBlink(); //创建数据库连接对象

if (db4.DBconn())

{ //连接数据库

if (db4.UpdataDeletAdd(delsql))

{

MessageBox.Show("删除记录成功！");

}

else

{

MessageBox.Show("删除失败！");

}

}

db4.DBclose();//关闭数据连接

string sql = "SELECT score\_info.Id as 学号, score\_info.name as 姓名,score\_info.course\_name as 课程,score\_info.score as 成绩 FROM score\_info";

BindsocreData(sql);

}

//更新成绩

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

{

ScoreInfo.id = textBox6.Text;

ScoreInfo.course\_id = textBox7.Text;

ScoreInfo.score = numericUpDown1.Value.ToString();

string Str3 = "select count(\*) from score\_info where Id = '" + ScoreInfo.id + "'and Course\_Id = '" + ScoreInfo.course\_id + "'";

string str5 = "update score\_info set score = '" + ScoreInfo.score + "'where Course\_Id = '" + ScoreInfo.course\_id + "'and Id = '" + ScoreInfo.id + "'";

DBlink db = new DBlink(); //创建数据库连接对象

DBlink db4 = new DBlink(); //创建数据库连接对象

if (db4.DBconn())

{ //连接数据库

string Str1 = "select Course\_name from course\_information where Course\_Id = " + ScoreInfo.course\_id;

string Str2 = "select name from user\_information where Id = " + ScoreInfo.id;

ScoreInfo.course\_name = db4.Getstringsearch(Str1);

ScoreInfo.name = db4.Getstringsearch(Str2);

//MessageBox.Show(ScoreInfo.course\_id + ScoreInfo.course\_name);

int count = db4.GetIntsearch(Str3);

if (count == 0)

{

string str4 = "insert into score\_info(name,Id,Course\_name,Course\_Id,score) " +

"values('" + ScoreInfo.name + "','" + ScoreInfo.id + "','" + ScoreInfo.course\_name + "','" + ScoreInfo.course\_id + "','"

+ ScoreInfo.score + "');";

bool bool6 = db4.UpdataDeletAdd(str4);// 执行查询

if (bool6)

MessageBox.Show("更新记录成功！");

else

MessageBox.Show("更新记录失败！");

}

else

{

bool bool7 = db4.UpdataDeletAdd(str5);// 执行查询

if (bool7)

MessageBox.Show("更新记录成功！");

else

MessageBox.Show("更新记录失败！");

}

}

db4.DBclose();//关闭数据连接

string sql = "SELECT score\_info.Id as 学号, score\_info.name as 姓名,score\_info.course\_name as 课程,score\_info.score as 成绩 FROM score\_info";

BindsocreData(sql);

}

# **实验总结**

这次的数据库大作业是历时两周，我们小组内相互帮助，在小组外也与做相同项目的小组互相解决问题。报告整理包括需求分析、概念结构设计、逻辑结构设计、MySQL数据库实施、C#界面设计及程序逻辑代码设计、文档排版等。

通过课程设计的整个过程，我了解了一个简单数据库系统实现的全过程。从最开始的需求分析，概念结构设计，逻辑结构设计，到最后的实施，每一步都需要认真严谨，不能有半点的马虎，否则就会给下一阶段的工作造成麻烦。这次能够成功的实现系统的功能，与小组成员间的团结合作有关。小组的讨论让我得到了很多想法，真正感觉到集体力量的强大，同时系统功能的实现也得益于老师的严格要求和耐心指导。在系统设计的整个过程中，老师也给出了许多宝贵的意见，使系统的功能更加完善。

 在这次实训中，我对C#语言有了一个更深的了解认识，对c#连接MySQL进一步的掌握，也对这个学期学的知识得到巩固，还尝试运行编程，每次运行程序成功，让我对下面的项目就充满信心。通过自己与同学合作编写程序，最终把最初的理论知识转化基本技能。  
 总而言之，通过这次的课程设计，我受益匪浅。

# **附录**

参考文献

【1】C#连接mysql三种方式

<https://www.2cto.com/database/201508/437072.html>

【2】C#中Panel控件的使用

https://blog.csdn.net/aran\_wdx/article/details/81567552