# 首先构建三层架构

1. **创建好UI显示层，BLL(逻辑层),DAL(数据访问层)**
2. **建立好UI层的界面**
3. **创建好Model(模型)和Common(工具类)，model是在其中创建对应sql数据库中的，common创建连接sql数据库的sqlhelper类和加密数据库的MD5类以及其他工具类**

# 对应需要进行的功能，完善DAL和BLL

1. **首先SqlHelper连接数据库**

**通过输入的密码和用户名，判断是否查询的到。在ui层调用DAL，DAL调用BLL，之后判断是否查询的到。**

**1）、其中在SqlHelper中，有两种连接SQL数据库的方法**

**其一：//连接时，必须对应数据库的名称，服务器以及用户名和密码，是静态私有的**

**private static string sqlstr = "server=.;uid=jgw;password=123456;database=学生学习系统管理DB";**

**其二：//还可以再ui层的中APP.config,创建一个<connectionString><数据库连接地址></connectionString>**

**//再调用system.configuration框架，最后调用其方法**

**private static string sqlStr = ConfigurationManager.ConnectionStrings["sqlcon"].ConnectionString;**

1. **在DAL中编程设定分为三步**

**1）、创建sql语句：**

**对于sql语句中的变量我们需要进行参数化，避免出现敏感字符**

string sql = "select adminname from admins where username=@id and password=@pwd ";

//将查询语句的限制参数化

SqlParameter[] sp = new SqlParameter[]

{

new SqlParameter("id",objAdmin.LogInId),

new SqlParameter("pwd",objAdmin.Logirpwd)

};

//还有一种发方法是

string sql1 = "select adminname from admins where username={0} and password='{1}'";//因为密码是字符串必须使用单引号

sql1 = string.Format(sql1,objAdmin.LogInId,objAdmin.Logirpwd);//也是将其限制的参数化

**2）、调用SqlHelper中的查询方法：**

DataRow dataRow = SqlHelper.GetDataRow(sql,sp);

**3）、返回结果：**

**判断连接是否成功和是否有满足查询的结果**

**判断是否接收到信息，或者为null。**

if (dataRow != null)

{

objAdmin.AdminName = dataRow["adminname"].ToString();

return objAdmin;

}

else

{

return null;

}

1. **在ui层调用BLL（因为BLL调用了DAL）**

**在登入界面：**

**1）、验证输入信息：不能为空，对于user不能有字符，需要以用正则来判断**

//先判断是否输入用户名和密码

if (this.txtLoginId.Text==null)

{

MessageBox.Show("请输入用户名");

}

//利用正则函数判断输入是否是正整数

if (!DataValidate.IsInteger(this.txtLoginId.Text.Trim()))

{

MessageBox.Show("请输入正整数");

}

if (this.txtLoginPwd.Text == null)

{

MessageBox.Show("请输入密码");

}

**2）、封装对象：将输入信息**

**创建对应数据库的类来见将其接受，调用对应的方法，当作参数输入。**

AdmisModel ModeL = new AdmisModel();

ModeL.LogInId=Convert.ToInt32(this.txtLoginId.Text);

ModeL.Logirpwd = this.txtLoginPwd.Text;

**3）、后台交互**

**在main中创建对应数据库的全局变量**

Program.model= AdminBLL.Get(ModeL);

**4）、处理交互结果**

if (Program.model != null)//登入成功，关闭登入窗体，显示主窗体

{

this.DialogResult = DialogResult.OK;

this.Close();

}

else

{

MessageBox.Show("请输入正确的用户名和密码");

# 3、窗体的嵌入：

**1、首先要创建嵌入窗体的对象**

**FrmAddStudent frmAddStudent = new FrmAddStudent();**

**2、判断容器中是否已经存在了窗体，需要将其关闭**

//判断容器中是否有别的窗体存在，将其他窗体关闭

foreach (Control item in this.panelForm.Controls)

{

if (item is Form)//判断容器中的是否是窗体，因为窗体嵌入，容器变为父级，可以使用里氏变换

{

Form frm = (Form)item;//将其转换为窗体对象，之后将其关闭

frm.Close();

}

}

**3、嵌入窗体**

**其一：将需要嵌入的窗体取消顶格显示，方面放在容器上方**

**其二：将窗体的边框取消掉**

**其三：选择嵌入窗体的父容器(因为这一个学员对应一个照片所以不用让文件夹对象可以多选图片路径)**

**其四：将嵌入的窗体，选择随着容器大小变化**

**其五：显示窗体**

//嵌入窗体的步骤

//1、将子窗体设置成非顶级控件，这样窗体才可以显示在容器上方

objfrm.TopLevel = false;

//2、去掉窗体的边框

objfrm.FormBorderStyle = FormBorderStyle.None;

//3、指定子窗体显示的容器，指定窗体的父级

objfrm.Parent = this.panelForm;

//4、设置窗体随着容器的大小自动给调整窗体的大小

objfrm.Dock = DockStyle.Fill;

//5、显示窗体

this.lblTitle.Visible = false;

objfrm.Show();

# 程序退出前的确认：

1. DialogResult dialogResult = MessageBox.Show("退出询问","确认退出",MessageBoxButtons.OKCancel,
2. MessageBoxIcon.Question );//messageboxbuttons.okcancel用来增加确认和取消按钮，
3. //messageboxlcon.question用来增加一个问号图片
4. if (dialogResult!=DialogResult.OK)
5. {
6. e.Cancel= true; //告诉窗体关闭这个事件取消

# 5、修改密码：

1. **创建DAL方法，用来修改数据库中管理员表格的方法**
2. **判断旧密码输入正确，新密码是否输入，长度是否满足要求，新旧密码是否相等。**
3. **调用DAL方法，修改密码，将修改的密码返回到全局变量Admin的属性中**

# 6、基于泛型集合动态填充下拉框：

添加学院中，在执行此窗体之前，需要读取出来的时所存在数据库中的班级表，将其存放在下拉框中，也就是在窗体的构造函数中调用DAL查询到，之后将其List<>集合放在comboBox.datasoucre，combobox.data

# 7、在添加学员信息里面，选择照片：

1、首先需要创建打开文件夹的对象

2、确定初始目录

3、确定打开文件夹的名字

4、确定可以选择文件的扩展名

5、显示文件夹窗体，同时给定返回值

6、用返回值，判断在文件夹中是否选择图片，选择图片就将其传给PictureBox.Image的路径，取消了文件夹窗体防止

PictureBox.Image=iamge.FromFile(open.filename);没有得到路径而报错。

//创建打开文件夹对象

OpenFileDialog open=new OpenFileDialog();

open.InitialDirectory = @"F:\wallhaven壁纸下载地";//打开文件夹的初始目录

open.Title = "选择图片";//打开对话框的名字

open.Filter = "所有文件|\*.\*|\*.jpg|";//可以选择文件的扩展名

DialogResult result= open.ShowDialog();//通过打开文件夹的返回值

if (result==DialogResult.OK)//当文件夹取消时，不给图片空间路径

{

//则将选择图片的路径给到图片控件

pbStu.Image = Image.FromFile(open.FileName);

}

# 8、添加学员对象：

1、首先要判断是否输入必须要求输入的数据，第二判断输入的数据，如身份证信息和卡号信息是否已经存在

2、调用sql语句存放学员对象，返回学号，并在表格中显示出啦

#region 数据验证

if (this.txtStudentName.Text.Trim().Length == 0)

{

MessageBox.Show("学生姓名不能为空");

this.txtStudentName.Focus();

return;

}

if (this.txtStudentIdNo.Text.Trim().Length == 0)

{

MessageBox.Show("身份证不能为空");

this.txtStudentIdNo.Focus();

return;

}

if (this.txtPhoneNumber.Text.Trim().Length == 0)

{

MessageBox.Show("手机号码不能为空");

this.txtPhoneNumber.Focus();

return;

}

if (this.txtCardNo.Text.Trim().Length == 0)

{

MessageBox.Show("卡号不能为空");

this.txtCardNo.Focus();

return;

}

if (this.txtAddress.Text.Trim().Length == 0)

{

MessageBox.Show("家庭住址不能为空");

this.txtAddress.Focus();

return;

}

if (this.cboClassName.SelectedIndex == -1)

{

MessageBox.Show("班级不能为空");

this.cboClassName.Focus();

return;

}

if (this.pbStu.Image == null)

{

MessageBox.Show("学生图像不能为空");

this.btnChoseImage.Focus();

}

if (this.txtPhoneNumber.Text.Trim().Length == 0)

{

MessageBox.Show("手机号码不能为空");

this.txtPhoneNumber.Focus();

return;

}

if (!this.rdoMale.Checked && !this.rdoFemale.Checked)

{

MessageBox.Show("请选择性别");

this.rdoFemale.Focus();

return;

}

//年龄限定18~35，用电脑运行程序的时间，减去所选择的时间

//或者利用数据库的时间，减去选择的时间

int age = DateTime.Now.Year - Convert.ToDateTime(this.dtpBirthday.Text).Year;

if (age > 35 || age < 18)

{

MessageBox.Show("年龄不符合要求");

return;

}

//身份证格式的验证，用正则表达式来，判断身份证格式

if (!DataValidate.IsIdentityCard(this.txtStudentIdNo.Text))

{

MessageBox.Show("身份证格式不对");

this.txtStudentIdNo.SelectAll();

this.txtStudentIdNo.Focus();

return;

}

//验证身份证和出生日期是否对的上，将选择的日期改为字符串的格式，

//通过字符串包含的方法来确定是否对的上

string drithday = Convert.ToDateTime(this.dtpBirthday.Text).ToString("yyyyMMdd");

if (!this.txtStudentIdNo.Text.ToString().Contains(drithday))

{

MessageBox.Show("身份证和生日不符");

this.txtStudentIdNo.SelectAll();

this.txtStudentIdNo.Focus();

return;

}

//输入的卡号和身份证号不能和数据库重复

string[] str = new string[] { this.txtStudentIdNo.Text, this.txtCardNo.Text };

if (StudnetBLL.BoolInStudent(str))

{

MessageBox.Show("学生表里面已经存在了身份证或者卡号");

this.txtStudentIdNo.Focus();

return;

}

//验证卡号必须是正整数

if (!DataValidate.IsInteger(this.txtCardNo.Text))

{

MessageBox.Show("卡号必须是正整数");

this.txtCardNo.SelectAll();

this.txtCardNo.Focus();

return;

}

#endregion

#region 添加学生

StudentModel stuModel = new StudentModel(this.txtStudentName.Text, this.rdoMale.Checked ? "男" : "女",

Convert.ToDateTime(this.dtpBirthday.Text), this.txtStudentIdNo.Text, this.txtCardNo.Text,

this.pbStu.Image!=null? new SerializeObjectToString().SerializeObject(this.pbStu.Image):"",

DateTime.Now.Year - Convert.ToDateTime(this.dtpBirthday.Text).Year, this.txtPhoneNumber.Text, this.txtAddress.Text, Convert

.ToInt32(this.cboClassName.SelectedValue));

int number = StudnetBLL.GetStu(stuModel);

if (number > 1)

{

//同步显示添加的学员

stuModel.ClassName = this.cboClassName.Text;

stuModel.StudentID = number;

this.liststu.Add(stuModel);

this.dataGridView1.DataSource = null;

this.dataGridView1.DataSource = this.liststu;

#region 添加学员成功之后需要判断是否继续添加学员

//询问是否继续添加,首先显示信息，之后生成按钮，表示继续还是结束

try

{

DialogResult result = MessageBox.Show("新学员添加成功", "是否继续添加", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

//因为所有的文本框都存放在groupbox容器中

foreach (Control item in this. groupBox1.Controls)

{

if (item is TextBox)

{

item.Text = "";

}

else if (item is RadioButton)

{

((RadioButton)item).Checked = false;

}

}

//日期控件不用初始化

this.cboClassName.SelectedIndex = -1;

this.pbStu.Image = null;

this.txtStudentName.Focus();

}

}

catch (Exception ex)

{

MessageBox.Show("添加学员失败的原因"+ex.Message);

}

#endregion

}

else

{

MessageBox.Show("添加失败");

return;

}

# 9、查询学员对象

**1、两种查询方式，第一种是班级查询，第二种是学号查询。**

**将查询的学员信息，在DataGridView。**

**（第一需要判断是否输入班级和学号信息，才可以查询，学号信息是否正整数）**

**2、查询之后，对其进行姓名排序和学号排序。**

**3、按照班级查询，需要将其所有的数据都显示出来，一下只能查询一个。**

**4、对班级查询出来的数据，选择其中某个学员选择修改**

**5、优化学号查询（用回车键快速查询），对于DataGridView中的数据，采用右键查询。**

# 10、对于学员成绩的查询

1、首先将所有学员的成绩读取出来（通过DataSet将数据传给DataGridView，这样两者会成为绑定关系，当DataSet的数据改变，DataGridVIew也会改变），并且将班级信息读取到ComboBox的下拉框中去。此时需要注意的是可以查询的sql语句有多张表，所以要选择对应的表，当只有一张表时

则为ds.tables[0],表示唯一的表。

2、通过成绩和班级，分别进行查询。第一在查询之前需要进行的判断是否拥有数据可以进行查询，第二输入的成绩是否是正整数。此时需要注意的是