# 首先构建三层架构

1. **创建好UI显示层,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数据库中的表，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：

添加学员中，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、保存学员对象：

1、首先要判断，在添加学生数据的ui中，所有必填的学生信息已填写，生日对应相应的身份证信息，并且电话号码，和卡号必须是正整数，数据库中没有对应新添加的学生的身份证号和卡号。

2、将所有学生信息，封装在student的对象类中，要将图片信息，转化为二进制的字符串信息，然后存入到数据库中去。

# 9、查询学员信息，可以更改删除学员信息

1、按照班级查询学员信息，按照学员姓名和学员学号进行排序。对于学生的排序，采用的是Icommparer接口。

在C#中，IComparer是一个接口，用于比较两个对象的大小。它通常用于对集合进行排序，例如List<>和Array。  
要使用IComparer，您需要实现该接口并提供一个Compare方法，该方法接受两个对象作为参数，并返回一个整数值，指示它们的相对顺序。如果第一个对象小于第二个对象，则返回负整数；如果第一个对象大于第二个对象，则返回正整数；如果两个对象相等，则返回零。

//姓名排序

class NameDESC : IComparer<StudentModel>

{

public int Compare(StudentModel x, StudentModel y)

{

return y.StudentName.CompareTo(x.StudentName);

}

}

//班级排序

class StuIDDESC : IComparer<StudentModel>

{

public int Compare(StudentModel x, StudentModel y)

{

return y.StudentID.CompareTo(x.StudentID);

}

}

if (this.dgvStudentList.RowCount == 0) return;

stulist.Sort(new StuIDDESC());

This.dgvstudentlist.Refrseh();重新排序

1. 对于选定的信息进行修改，利用到DataGridVIew中的

DataGridVIew.CurrentRow.Cell[“studentid”].value.tostring()；来获得选定行数据的某个列单独的数据。

3、修改学员信息，已知学号是主键，无法修改，需要将其所在的控件的属性Enabled=false,不启动该控件；在修改之前还需要判断是否输入了相对应的数据，判断重新输入的学生身份证是否在数据库中，已存在；修改之后，需要再次执行相对应的查询，保证数据的跟新。

4、删除学员信息，可以会出现以下情况，删除的主表中的信息，对应的主副关系的副表无法删除会报错，对应的Sqlexception ex的ex.number==547错误，单独显示。

# 10、查询学员成绩

1、首先查询出来所有的学员成绩，用DataSet将查询出来存储，并且存放在DataSet.Table [0]，表格中，因为只查询了一个表格。

2、对应成绩，班级进行查询，是利用的DataSet的属性

DataSet.Table[0].DefaultView.Rowfilter=对应不同的要求，对应数字是，”Csharp>”+this.textscore.text.trim();

对应班级时string Format(“ClassName=’{0}’”，this.cboClass.text.trim());当然，在对应班级查询的时间SelectIndexChanged的时候，需要先判断对应DataGridVIew对应的数据DataSet的对象是否含有数据，要是没有会报错，所以需要先判断是否存有数据。

# 查询全校，班级学员成绩，并且将缺考人员，以及全校，班级每科的平均分全部查询出来

1. 将按照班级查询的方式将每个班级的学生查询出来，此时要注意的时候，调用了ComboBox的事件SelectIndexChanged的事件是，只要在ComboBox添加数据的时候，就会自动执行一遍SelectIndexChanged事件，所以要将其在窗体里面添加这个事件的代码，同时在窗体另一个部分类中添加这个事件的代码删除掉，放在创建窗体的构造函数里面。也就是在窗体类是由两个部分类组成的。
2. 对于查询，考试人数，缺考人数和每科平均分，需要采用sqlserver中的常用函数的方法count，avg，表中的行数和平均分，对于缺考人数，在查询的时候还需要查用子查询，我们使用的SQL中的read生成虚拟表，所以读取数据时，第二次读取的数据，需要采用nextresult方法，来读取下一个结果集。
3. 对于查询学员成绩，分为按照班级查询学员成绩；再就是查询参考人数，缺考人数，每科的平均分；最后是将缺考的人员名单查询出来。
4. 将查询的时候，不同的存储数据对应不同的对象，全部放在扩展对象里面（扩展对象，就是将所创建好的类，变为扩展对象里面的属性），之后再DataGridView中的事件CellFormatting，设定单元格内容，因为单元格中的数据来源，只能是对象的第一级，也就是对象的属性，扩展对象里面将对象当作属性，其对象的属性就不在是第一级。所以DataGridVIew中的数据来源要写为扩展对象的属性的命名对象。

# 考勤打卡

1. 显示时间，需要利用到DateTime.Now.Day(Year,Month,TolongTimestring(),SonDay)这个方法可以得到当前的日期。需要通过Switch（DateTime.Now.DayofWeek）{dayofweek.sonday来判断周几}
2. 记录考勤信息（允许重复打卡），在打卡时，需要先从数据库中读取出来，实际有多少学员，有多少学员已打卡，在考勤表中读取，缺考人员，因为在考勤表中可能存在很久之前打卡的信息，而且有当天打卡的信息，在查询当天的打卡里面有不有重复的，需要利用到sqlserver中between查询方法，表示范围查询，一般用于日期和数字筛选，一般会写一个读取当前数据库的信息。用sqlserver中的Distinct card 之后between 查询当天不同卡号的考勤的总人数。
3. 显示学员信息，同时将学员信息在列表中展示
4. 同时将学员信息在列表中，通过卡号在学员表中查询出来的是为空，显示不存在，讲上一个打卡的学员信息，全部清空。
5. 将添加成功时的服务器时间，给定到表格中，当窗体在加载时，就将鼠标聚焦在this.txtstucardno.fouce()上。
6. 在添加学员，首先要判断是否输入为空并且是否按下回车键，第二要判断输入的卡号是否成功添加，成功之后要将其对应学员信息，反映在表格中，并且同时更新实到和缺勤人数。
7. DataTime.Now.ToShortDateString()是将现在的日期转为只有年月日，ToShortTimeString()是将现在的日期转化为年月日之后还包含，时分秒。

# 考勤查询

1. 通过日期和姓名查询学员考勤情况
2. 需要查询当天考勤的总人数，需要当天打卡的人数将你查询的日期，为限制去查询数据表中，来查询。

**当a窗体存在datagridview控件，并且将a窗体插入到b窗体的Panel空间中，不显示所有数据列，而是通过下滑线来显示所有的数据列，可以将datagridview的属性AutoSizeColumnsMode改为fill自动填充。**

# 外部Excel导入学员信息

1. **编写一个读取Excel的通用访问方法类OleDbHelper类**，using System.Data.OleDb;这个是写Ecxel方法的命名空间。

// 将 'YourExcelFile.xlsx' 替换为你的 Excel 文件的路径。建立问chatgot，要明确自己excle是多少版本的。

string filePath = @"C:\路径\到\你的Excel文件.xlsx";这是连接某单个表的文件

// Excel 2020 的 OLEDB 连接字符串

string connectionString = $"Provider=Microsoft.ACE.OLEDB.12.0;Data Source={filePath};Extended Properties=\"Excel 12.0;HDR=YES;\"";之后对于Excel的读取就和SQLserver差不多啦

1. **编写能够导入Excel数据的ImportDataFromExcel类，**添加查询Excel数据表的方法（要求数据封装成对象）
2. **在窗体DataGridView中展示读取好的数据。首先需要注意一点，在读取表格时，需要点名时Excel中的表格Select \* from [Sheets$],必须点名$.其余和读取Sqlserver表格一样.**
3. **在SQLhelper类中编写同时插入多条SQL语句的事务方法。首先需要在DAL中将需要执行的sql语句,全部添加为List<string>的集合,然后再SQLhelper中调用transaction的事务方法,之后开启事务con.BeginTransaction()开启事务,将所有sql添加到事务之后执行完成之后,还需要提交事务cmd.transaction.commit(),在报错中,要明确事务回滚cmd.transaction.RollBack(),finally中将事务清空,因为不管报错和事务完成事务是不清空的 cmd.transaction=null; con.close().**
4. 在ImportDataFromExcel类编写保存“多个集合对象的方法”
5. 在ui中将数据保存到sqlserver中。对此需要在保存之后，将在ui层建的List<StudentModel>的对象的内容Clear,将DataGridView的DataSource清空。

# 打印学生信息(实现有Excel模板报表的基本思路)

1. 提前准备好Excel模板，只需要明确需要填充数据的地方。
2. 通过Excel操作类，来操作当前的Excel模板。

特别注意，因为安装了Excel软件

所以在UI层需要引用Microsoft Excel 14.0/16.0bject Library。

把Excel模板复制到Debug当中

1、第一步创建Excel工作簿对象

2、因为存在Excel模板，所以要获取到Excel模板的绝对路径

3、将模板路径加入到所创建的Excel模板的对象当中去。

4、Microsoft.Office.Interop.Excel.Worksheet sheet = excelApp.Worksheets[1];将模板中指定的表放在所创建的表对象当中

5、添加数据，将数据存放到指定的行列当中（图片除外）

6、显示Execl可见，同时预览表格

7、释放资源

excelApp.Quit()；

System.Runtime.InteropService.Marshal.ReleaseComObject(excelApp);

excelApp=null;

# 打印学生成绩信息（无模板Excel表格的基本思路）

1. 创建Excel工作簿
2. 创建Sheet表格

Microsoft.Office.Interop.Excel.WorkSheets sheet=excelApp.WorkWords.Add().WorkSheets[1];

1. 设置标题的样式
2. 获取总列数和总行数
3. 最后将所有数据放入Sheet表格中