**信息科学与工程学院**

**2020－2021学年第一学期**

实 验 报 告

课程名称： 高级编程技术

实验名称： 实验三

专 业 班 级 通信工程 二班

学 生 学 号 201800121050

学 生 姓 名 孟麟芝

实 验 时 间 2020年10月17日

实验报告

## 【实验目的】

1.掌握GUI的编写方法，学会使用各种控件。

2.通过对GUI编写的深入思考，加深对C#面向对象特点及其结构的理解。

## 【实验要求】

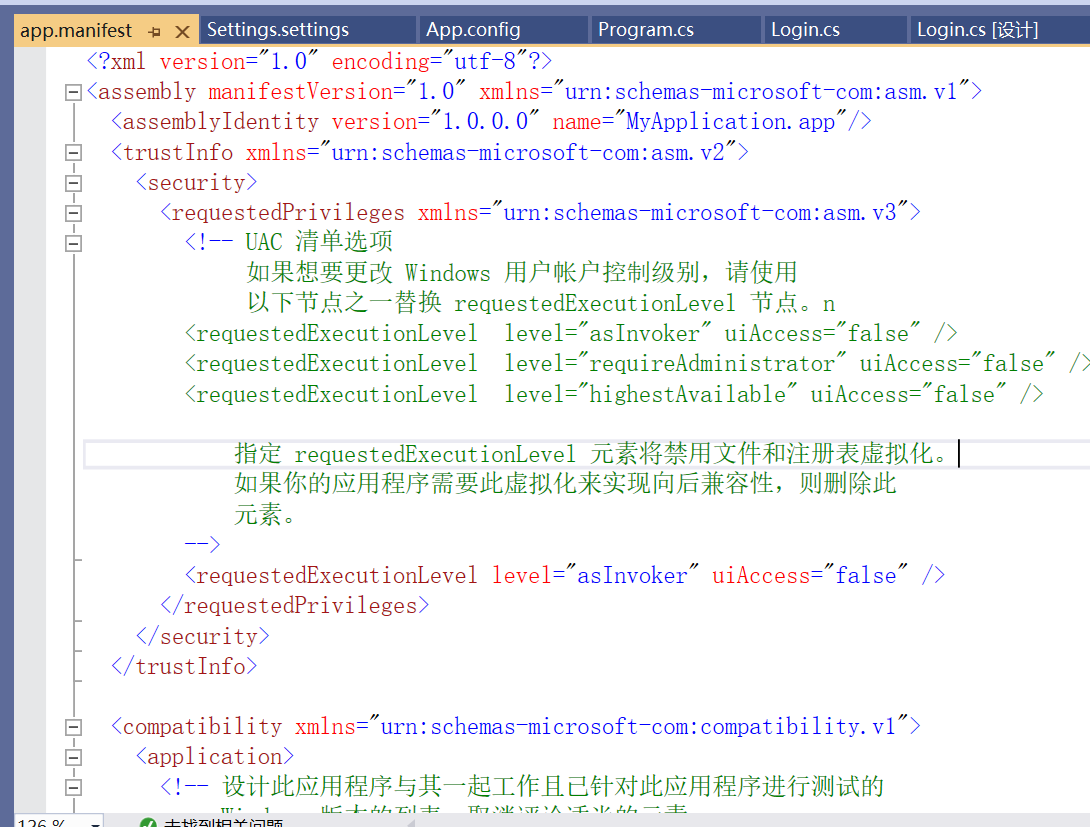
跟着书上给出的学生管理系统的例子，使用各种控件，加入自己的理解，搭建一个学生管理系统。

## 【功能演示】

由于直接对代码和分立的窗体的实现方式进行说明比较模糊，故先展示其功能，再对其中的细节进行一些说明。

另外，后面的说明过程中不再赘述课本上有的内容，主要叙述了一些学习过程中的发现。

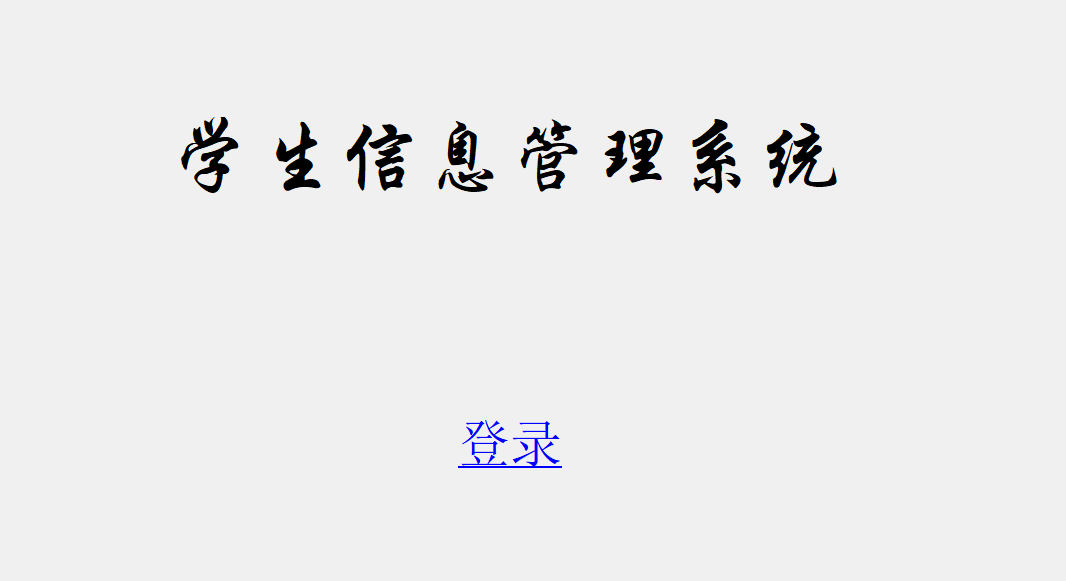
本程序通过引入微软自带的app.manifest，实现了对高分辨率屏幕的适配，使显示出的界面不至于模糊。



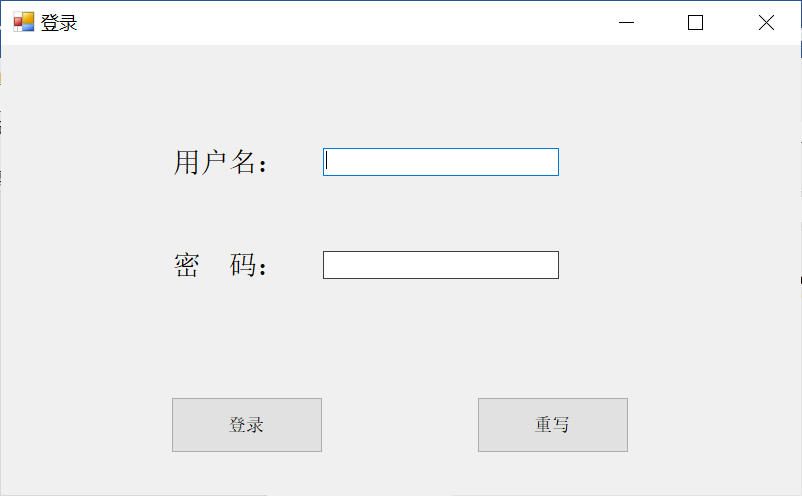
只需要将下面这一段代码取消注释即可实现对高分屏的适配。



首先，启动后可以看到下面这样的界面，点击登录这个超链接，超链接会变色，窗口隐藏，登录窗口展出。



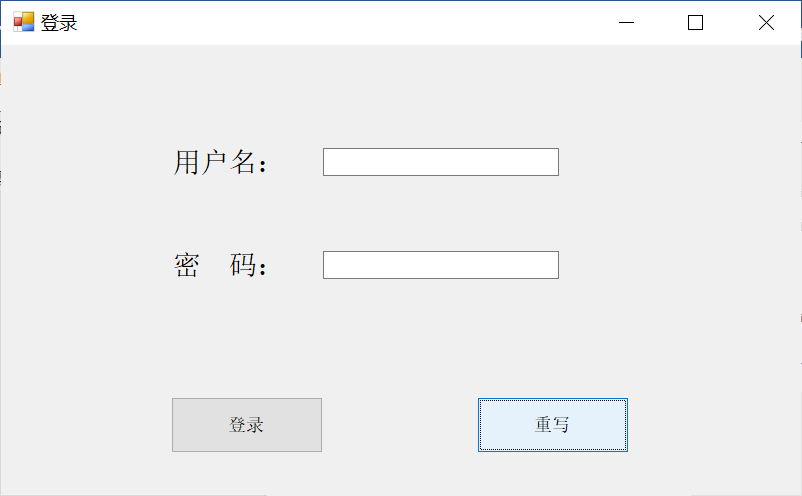
登录窗口如下：



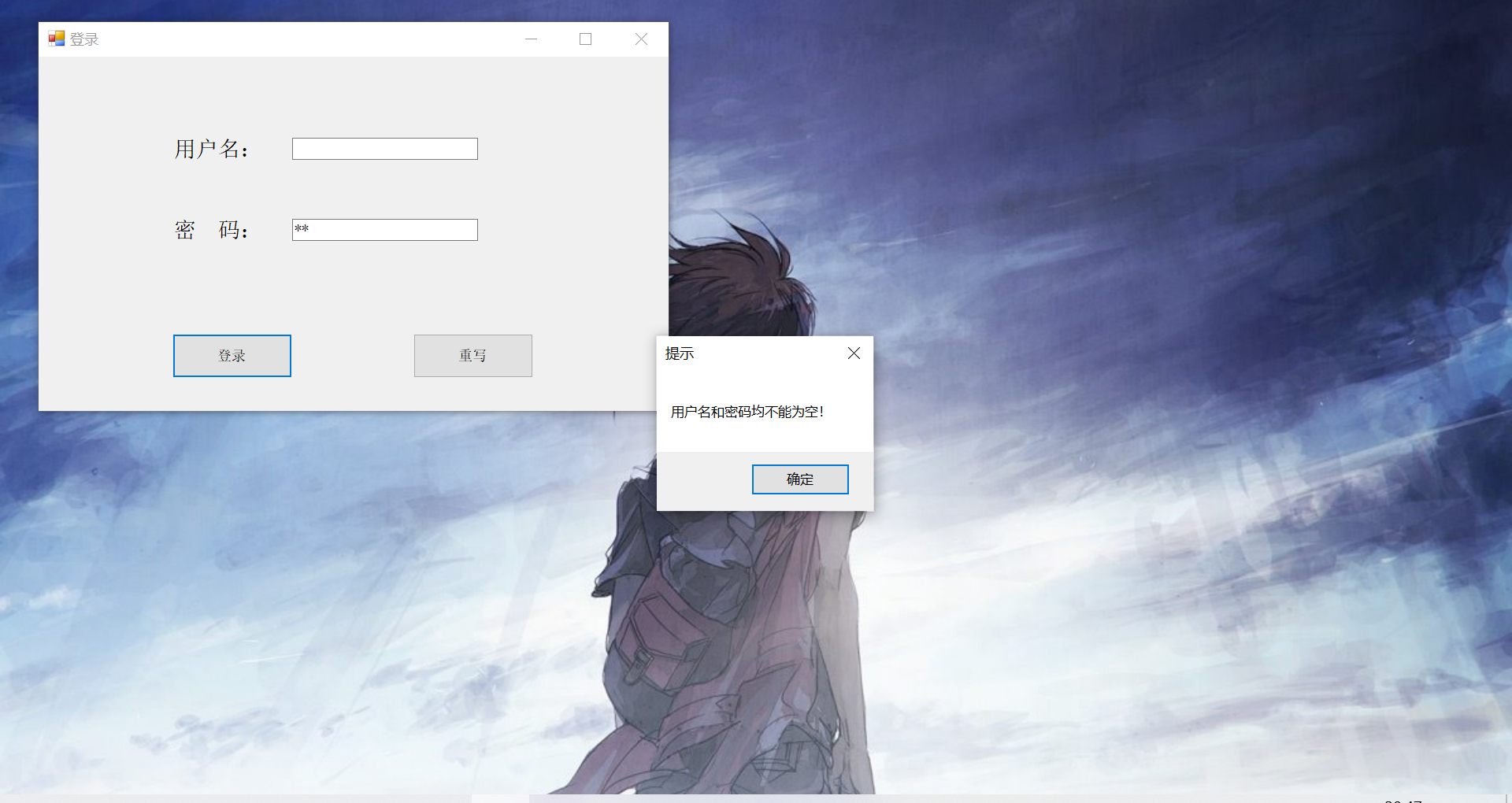
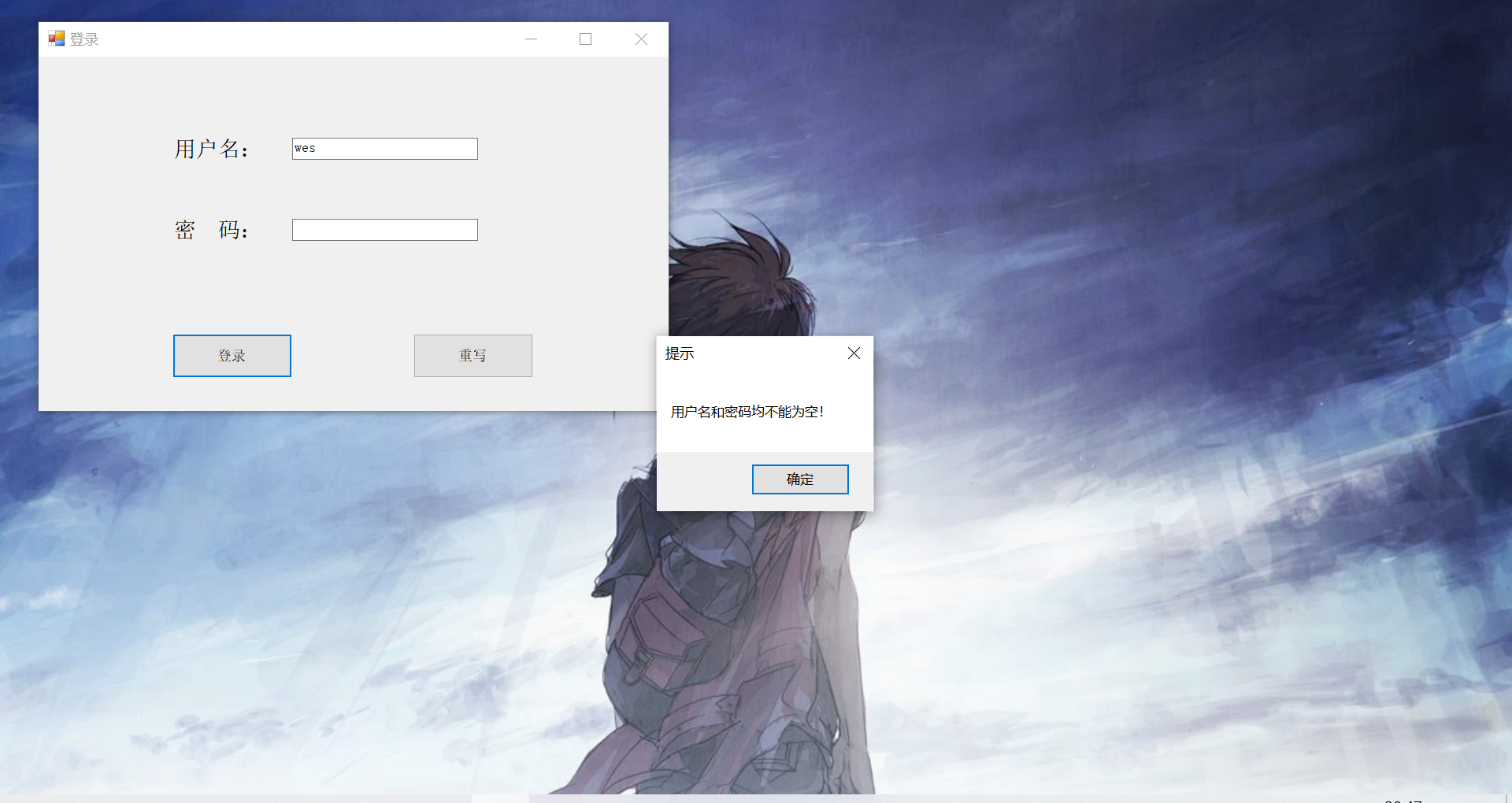
这一窗口中有一些细节，如密码输入后不会明确的显示出来，将加密成\*



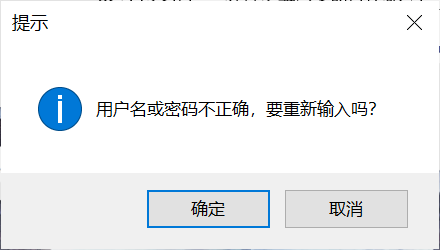
另外，点击重写，则可以清空已经输入的内容，重新输入



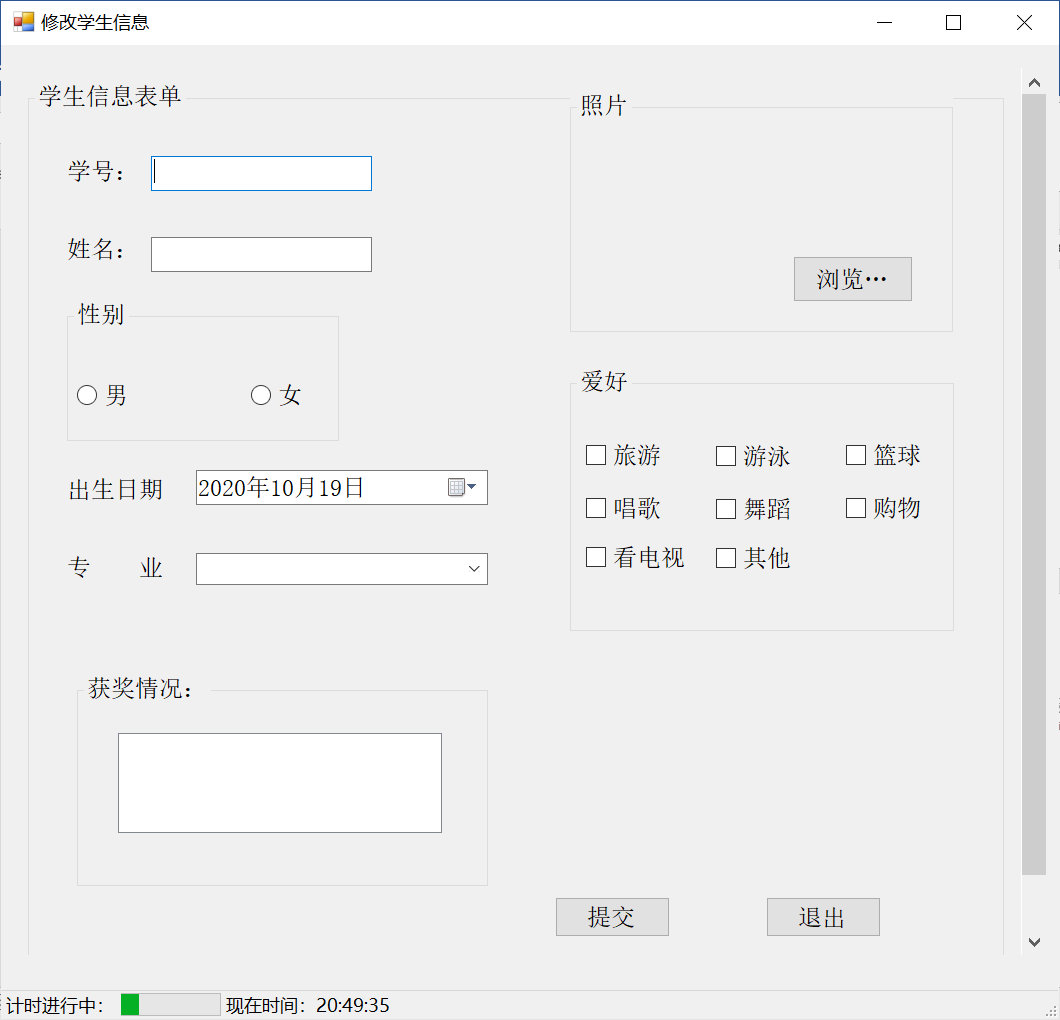
当用户名和密码有任意一个为空时，程序都会弹出警告



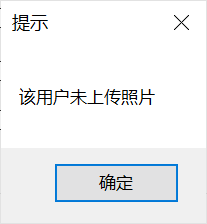
而若输入用户名或密码有误，则会提醒，若点击确定则将清空已输入的内容，点击取消将保留已经输入的内容



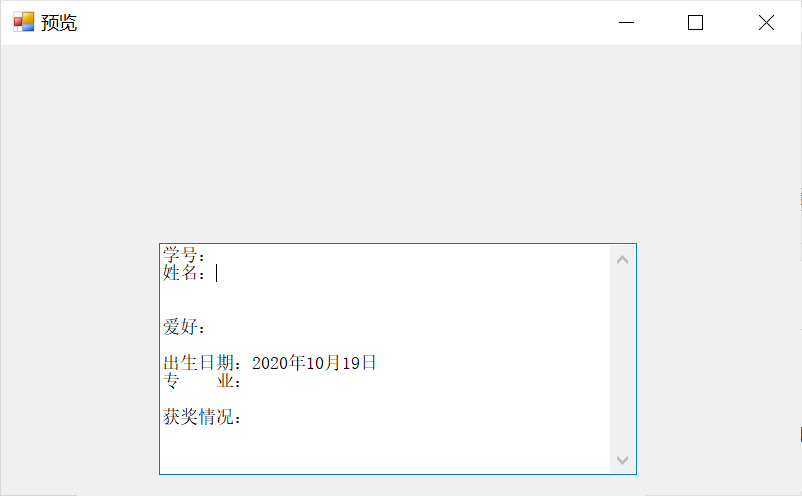
只有输对用户名和密码时，才可登录进上传信息界面



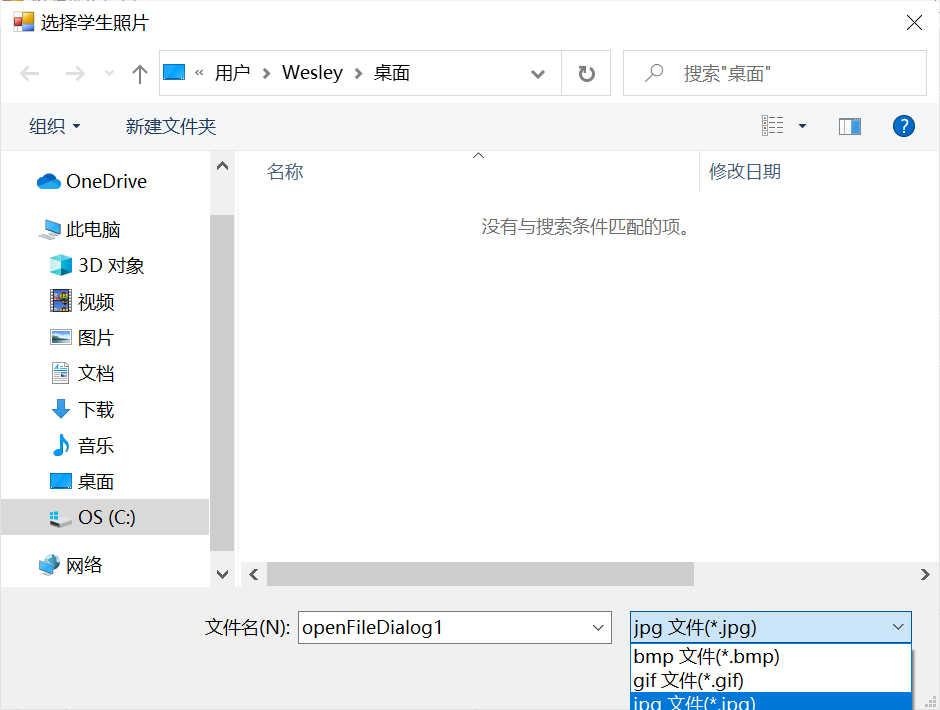
这一界面中只需要进行信息输入操作即可，若用户没有添加照片则也会弹出提醒。



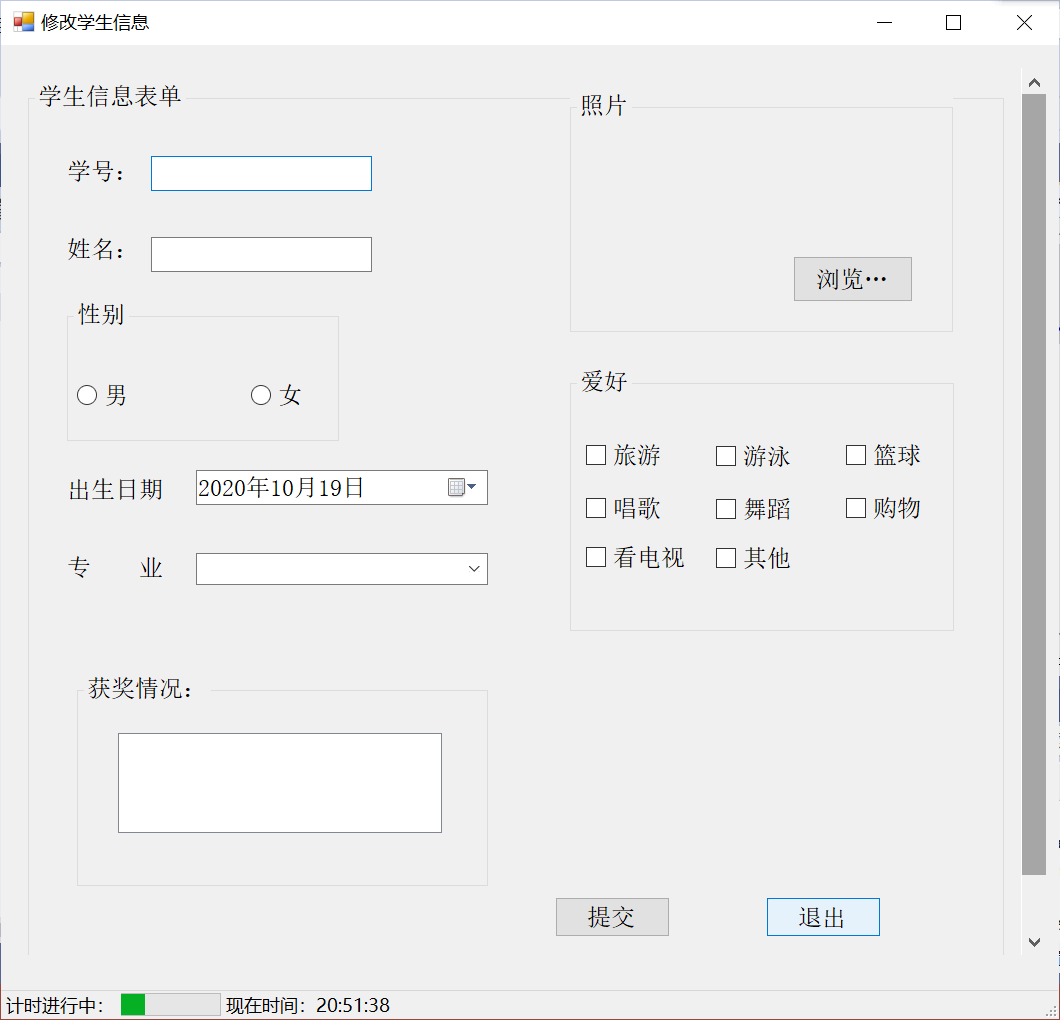
相应的，信息展示界面照片将为空。



点击浏览，则会弹出一个对话框，进行照片选择，默认打开的路径是桌面，可以选择添加几种图片文件



若点击退出，则会直接退出整个程序。



若正常填写信息，则有如下效果，为了防止信息过多显示不全，还贴心的设置了滚动条。



## 【欢迎窗口代码】

using System;

using System.Windows.Forms;

namespace StudentsManagement

{

public partial class Welcome : Form

{

public static Login loginForm;

public static Update updateForm;

public static Browser browserForm;

public Welcome()

{

//这里是构造函数，实例化出来了这两个对象

InitializeComponent();

loginForm = new Login();

updateForm = new Update();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

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

{

}

private void linkLabel1\_LinkClicked\_1(object sender, LinkLabelLinkClickedEventArgs e)

{

linkLabel1.LinkVisited = true;

loginForm.Show();

this.Hide();

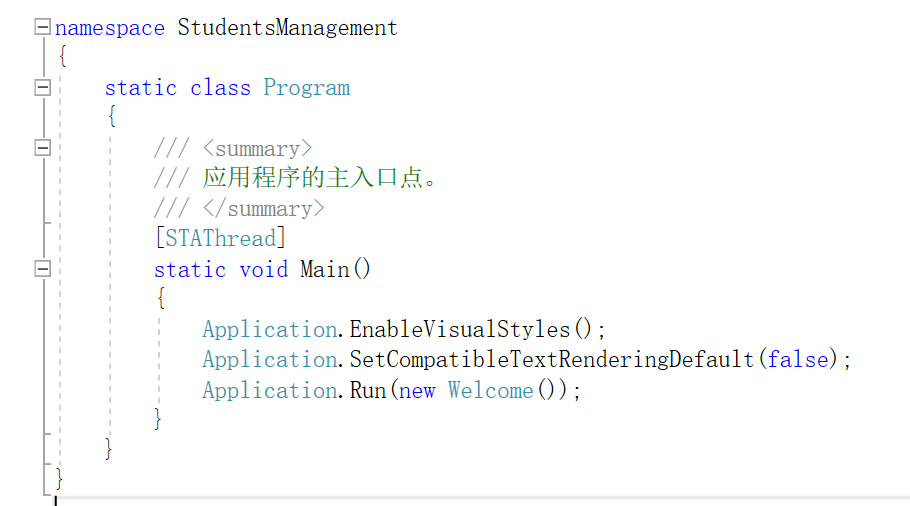
}

}

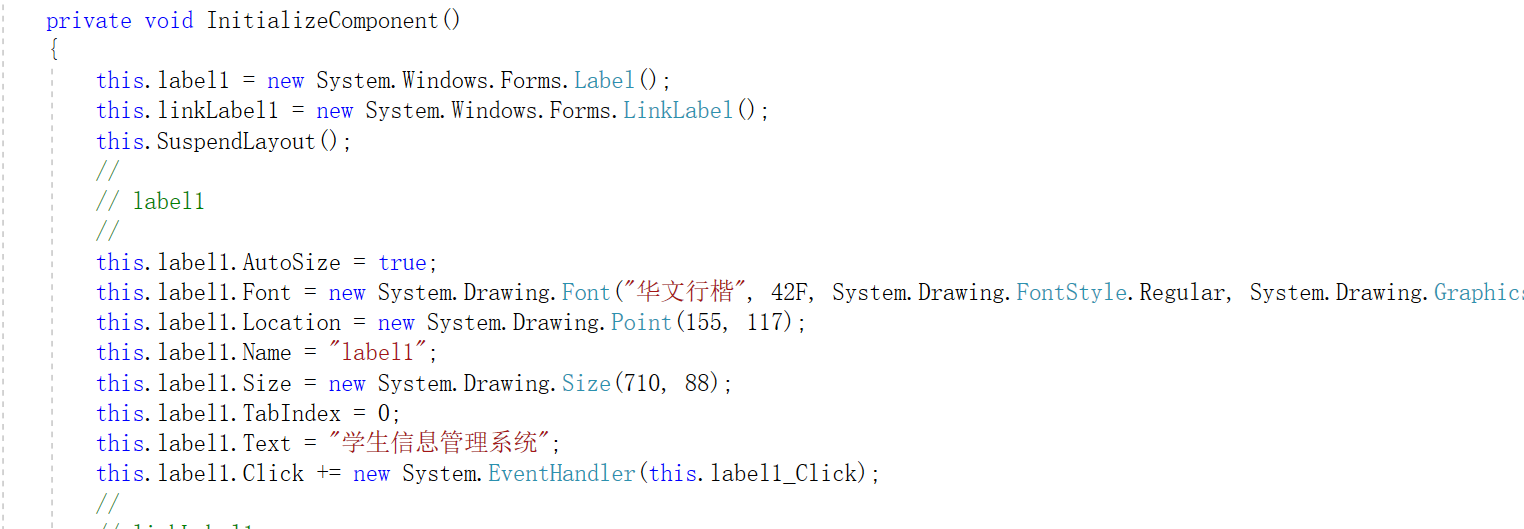
}

## 【欢迎窗口说明】

从其代码中可以看出，C#中处处皆对象，窗体本身是从Form这个基类派生出来的，为什么第一次打开我们会进入Welcome这个窗口呢，原因在于，开发GUI时集成开发环境为我们自动添加了一个程序入口，并且帮我们实例化了第一个窗口然后将其显示：



窗口中的控件又是如何显示出来的呢？我们看到在一个窗口的构造函数中含有一个InitializeComponent()，转到它的声明与实现，发现如下代码，可见拖入控件后，IDE会自动帮忙将其实例化，在图形界面中设置的控件属性也会被写入这一方法。



了解了这些，我们就大体明白了C#中的GUI是如何运作的，且它是如何体现C#面向对象的特征的。

## 【登录窗口代码】

using System;

using System.Windows.Forms;

namespace StudentsManagement

{

public partial class Login : Form

{

public Login()

{

InitializeComponent();

}

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

{

}

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

{

if (textUserName.Text == string.Empty || textPassword.Text == string.Empty)

{

MessageBox.Show("用户名和密码均不能为空！", "提示");

}

else

{

if (textUserName.Text == "wesley" && textPassword.Text == "123")

{

Welcome.updateForm.Show();

this.Hide();

}

else

{

//DialogResult是MessageBox.Show()的返回值，它是一个枚举类型

DialogResult result = MessageBox.Show("用户名或密码不正确，要重新输入吗？", "提示", MessageBoxButtons.OKCancel, MessageBoxIcon.Information);

if (result == DialogResult.OK)

{

textUserName.Clear();

textPassword.Clear();

}

}

}

}

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

{

textUserName.Clear();

textPassword.Clear();

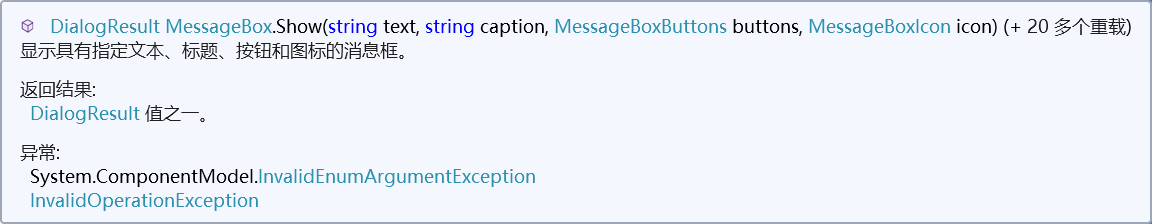
}

}

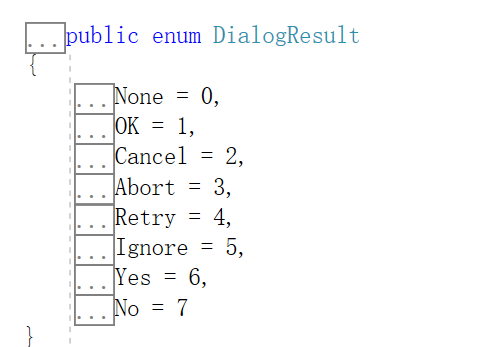
}

## 【登录窗口说明】

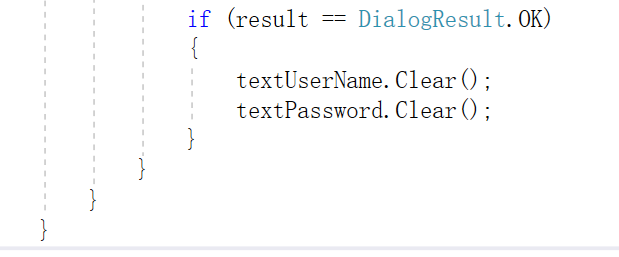
这一窗口主要编写好单击Button后的事件即可，其中有一个值得注意的地方：获取Messagebox.Show()的返回值。



直接尝试从其文档中获取一些答案，可见它的返回值是DialogResult的值之一，那这个DialogResult又是什么呢？转到它的声明，便豁然开朗了，这是一个枚举类型



可见一些看似高级的功能，其实是由课本上一些基本的东西拼接而成的。



那么我们只需要加一个if语句即可实现前面演示的按确定重写，按取消保留的功能了。

## 【修改信息窗口代码】

using System;

using System.ComponentModel;

using System.Drawing;

using System.Windows.Forms;

namespace StudentsManagement

{

public partial class Update : Form

{

//这两个成员是public static的，使之可以在其他窗口被直接用类访问

public static string path;

public static string info;

public Update()

{

InitializeComponent();

}

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

{

}

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

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

}

private void openFileDialog1\_FileOk(object sender, CancelEventArgs e)

{

}

private void pictureBox1\_Click\_1(object sender, EventArgs e)

{

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void timer1\_Tick(object sender, EventArgs e)

{

toolStripStatusLabelNowTime.Text = string.Format("现在时间：{0}", DateTime.Now.ToLongTimeString());

toolStripProgressBar1.PerformStep();

if (toolStripProgressBar1.Value == toolStripProgressBar1.Maximum)

{

toolStripProgressBar1.Value = toolStripProgressBar1.Minimum;

}

}

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

{

Environment.Exit(0);

}

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

{

this.openFileDialog1.InitialDirectory = @"C:\Users\Wesley\Desktop";

this.openFileDialog1.Filter = "bmp 文件(\*.bmp)|\*.bmp|gif 文件(\*.gif)|\*.gif|jpg 文件(\*.jpg)|\*.jpg";

//设置为3，默认打开对话框中显示的文件是所有文件

this.openFileDialog1.FilterIndex = 3;

this.openFileDialog1.RestoreDirectory = true;

this.openFileDialog1.Title = "选择学生照片";

if (this.openFileDialog1.ShowDialog() == DialogResult.OK)

{

path = this.openFileDialog1.FileName;

}

pictureBox1.Image = Image.FromFile(path);

}

private void listBox1\_SelectedIndexChanged\_1(object sender, EventArgs e)

{

}

private void listBox1\_MouseDoubleClick\_1(object sender, MouseEventArgs e)

{

listBox1.Items.Clear();

listBox1.Items.Add("2020年度校优秀团员");

listBox1.Items.Add("2020年度国家奖学金");

listBox1.Items.Insert(2, "2020年度校运动会10000米游泳冠军");

listBox1.SelectedIndex = 1;

string msg = string.Format("已填加奖惩记录{0}条", listBox1.Items.Count);

MessageBox.Show(msg, "提示");

}

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

{

//先对姓名和学号进行输出

//\r为移至行首，\n为换行

info = "学号：" + NumBox.Text + "\r\n";

info += "姓名：" + NameBox.Text + "\r\n";

//对性别识别输出

foreach (Control control in GroupSex.Controls)

{

if ((control as RadioButton).Checked)

{

info += "性别：" + (control as RadioButton).Text;

}

}

//对爱好识别输出

info += "\r\n\r\n爱好：";

foreach (Control control in GroupFavor.Controls)

{

if ((control as CheckBox).Checked)

{

info += (control as CheckBox).Text + " ";

}

}

//对出生日期与专业识别显示

info += "\r\n\r\n出生日期：" + dateTimePicker1.Text;

info += "\r\n专 业：" + comboBox2.Text;

//对获奖情况识别显示

info += "\r\n\r\n获奖情况：";

int count = 1;

foreach (string i in listBox1.Items)

{

if (i != "")

{

info += "\r\n " + count.ToString() + "." + i;

count++;

}

}

//打开预览窗口

Welcome.browserForm = new Browser();

Welcome.browserForm.Show();

//这里使用透明度调为100%，实现窗口隐藏，其实用Hide更好

this.Opacity = 0;

}

}

}

## 【修改信息窗口说明】

这一窗口的说明在代码的注释中已经比较清晰了，在实现过程中遇到一个问题，即如果使用panel的滚动条功能，滚动到底后滚动条会不正常的弹动，经过查阅信息后，这个问题仍未得到解决。

另外一个值得注意的地方是，在按提交按钮时将实例化出browserForm这个对象，同时调用了它的构造函数，那么想要在其中显示内容，只需要将browserForm的构造函数进行修改即可，一实例化出来便已经带有了我们想要的值。

## 【预览窗口代码】

using System;

using System.Drawing;

using System.Windows.Forms;

namespace StudentsManagement

{

public partial class Browser : Form

{

public Browser()

{

InitializeComponent();

//这里需要引入System.Drawing包

if (StudentsManagement.Update.path != null)

{

pictureBox1.Image = Image.FromFile(StudentsManagement.Update.path);

}

else

{

MessageBox.Show("该用户未上传照片", "提示");

}

textBox1.Text = StudentsManagement.Update.info;

}

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

{

}

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

{

}

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

{

}

}

}

## 【预览窗口说明】

如上所述，这个窗口的构造函数中做了两个操作：将textBox进行赋值初始化，将pictureBox初始化，之所以对未上传照片进行额外的考虑，是因为如果path为空路径，那么Image.FromFile()的参数将为NULL，程序会遇到错误而退出，为了避免这一情况，故加入了if语句，若path为空则弹出提醒，不为空时才传给Image.FromFile()。

## 【实验心得】

本次实验对C#有了更深刻的理解，即便是看起来与代码相去甚远的GUI，也离不开基本的类、对象等，GUI的代码与控制台程序的代码是一脉相承的，从基本概念上来讲，并没有什么“新花样”，然而，如名言所说，”More is different”，细小的零件拼接，产生了神奇的效果。

学好基本的知识是非常重要的，在使用一个事物的过程中要去关注它的本质，才能真正的把握他。编写GUI的过程中又要具有大局观，不能过分拘泥于某个细节具体如何实现，要对整体的框架有所把控。