实验五 数据库应用程序开发

一、实验目的

1. 掌握 C#配置、连接并操作数据库的方法。

2. 掌握数据库应用程序的设计和开发步骤；

二、实验要求 根据要求，编写 C#程序，并将程序代码和运行结果写入实验报告。

三、实验内容 【任务】 设计并编写一个数据库应用程序，要求如下：

λ 设计登录界面；

λ 采用多文档界面(MDI)，至少包含一个子窗体；

λ 主窗体上包含菜单、工具栏、状态栏；

λ 数据库可以采用 Access、SQL，至少包含 1 个数据表；

λ 程序中实现录入、浏览、查询的功能。

【要求】

λ 实验报告中将程序实现的功能描述清楚；

λ 实验报告中说明所连接的数据库和数据表；

λ 实验报告中包含界面布局和运行结果的截图；

λ 实验报告中包含事件代码。

【注意】

λ 实验报告中说明登录的用户名和密码；

λ 如程序执行有特殊要求，请在实验报告中说明。

数据库登录名：20142966密码：20142966

程序源代码：

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace StudentMis

{

public partial class Login : Form

{

private DBCL.DBEntities db = new DBCL.DBEntities();

public Login()

{

InitializeComponent();

}

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

{

this.Hide();

StudentAdd frmStudentAdd = new StudentAdd();

frmStudentAdd.ShowDialog(this);

}

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

{

string strUserName = "";

string strPWD = "";

lbMessage.Text = " ";

if (tbusername.Text.Trim() == "")

{

lbMessage.Text = "账号不能为空!";

//MessageBox.Show("学号不能为空!");

tbusername.Focus();

return;

}

if (tbpassword.Text.Trim() == "")

{

lbMessage.Text = "密码不能为空!";

tbpassword.Focus();

return;

}

strUserName = tbusername.Text.Trim();

strPWD = tbpassword.Text.Trim();

var model = db.student.FirstOrDefault(m => m.username == strUserName);

if (model == null)

{

lbMessage.Text = "账号不存在，请重新输入!";

return;

}

if (strPWD == model.password)

{

this.Hide();

Main frmMain = new Main();

frmMain.Show();

}

else

{

lbMessage.Text = "用户名与密码不匹配!";

}

}

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

{

lbMessage.Text = "";

}

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

{

Application.Exit();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace StudentMis

{

public partial class Main : Form

{

public Main()

{

InitializeComponent();

}

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

{

this.Close();

Login Login1 = new Login();

Login1.Show();

}

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

{

StudentAdd frmStudentAdd = new StudentAdd();

frmStudentAdd.ShowDialog(this);

}

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

{

StudentUpdate frmStudentAdd = new StudentUpdate();

frmStudentAdd.ShowDialog(this);

}

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

{

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace StudentMis

{

public partial class StudentAdd : Form

{

private DBCL.DBEntities db = new DBCL.DBEntities();

private void InitControls()

{

lbMessage.Text = " ";

}

public StudentAdd()

{

InitializeComponent();

}

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

{

string strID = "";

string strName = "";

lbMessage.Text = " ";

if (tbID.Text.Trim() == "")

{

lbMessage.Text = "学号不能为空!";

//MessageBox.Show("学号不能为空!");

tbID.Focus();

return;

}

if (tbName.Text.Trim() == "")

{

lbMessage.Text = "修改密码不能为空!";

tbName.Focus();

return;

}

// 获取数据

strID = tbID.Text.Trim();

strName = tbName.Text.Trim();

// 检验密码确认密码是否相同

try

{

DBCL.student model = new DBCL.student();

model.username = strID;

model.password = strName;

db.student.Add(model);

db.SaveChanges();

this.Close();

Login frmMain = new Login();

frmMain.Show();

}

catch

{

MessageBox.Show("添加失败!");

return;

}

}

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

{

this.Close();

Login frmMain = new Login();

frmMain.Show();

}

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

{

}

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

{

lbMessage.Text = "";

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace StudentMis

{

public partial class StudentUpdate : Form

{

// 变量 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

private string mstrID = "";

private DBCL.DBEntities db = new DBCL.DBEntities();

private void InitControls()

{

lbMessage.Text = " ";

}

public StudentUpdate()

{

InitializeComponent();

}

public StudentUpdate(string strID)

{

InitializeComponent();

mstrID = strID;

}

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

{

lbMessage.Text = "";

var model = db.student.FirstOrDefault(m => m.username == mstrID);

if (model == null)

{

return;

}

tbusername.Text = model.username;

tbusername.ReadOnly = true;

tbpassword.Text = model.password;

}

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

{

}

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

{

string strID = "";

string strName = "";

lbMessage.Text = "";

if (tbpassword.Text.Trim() == "")

{

lbMessage.Text = "密码不能为空!";

//MessageBox.Show("学号不能为空!");

tbpassword.Focus();

return;

}

if (tbfox.Text.Trim() == "")

{

lbMessage.Text = "确认密码不能为空!";

tbfox.Focus();

return;

}

// 检验密码确认密码是否相同

if (tbpassword.Text.Trim() != tbfox.Text.Trim())

{

lbMessage.Text = "密码与确认密码不相同!";

tbfox.Focus();

return;

}

// 获取数据

strID = tbusername.Text.Trim();

strName = tbfox.Text.Trim();

var modell = db.student.FirstOrDefault(m => m.username == strID);

if (modell == null)

{

MessageBox.Show("该学号在数据库中不存在!");

return;

}

modell.username = strID;

modell.password = strName;

try

{

db.SaveChanges();

}

catch

{

MessageBox.Show("编辑失败!");

return;

}

this.Close();

/\* try

{

DBCL.student model = new DBCL.student();

model.username = strID;

model.password = strName;

db.student.Add(model);

db.SaveChanges();

}

catch

{

MessageBox.Show("添加失败!");

return;

}\*/

}

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

{

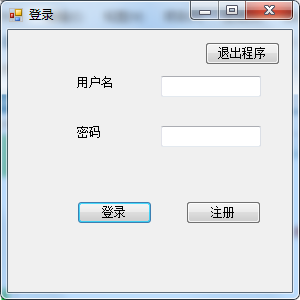
this.Close();

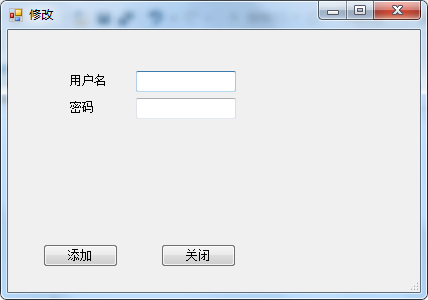
}

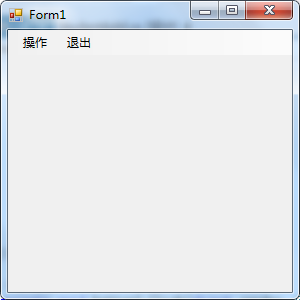
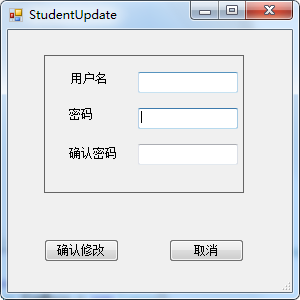
}

}

程序结果截图：







四、实验总结

这次的实验花了好久的时间才做完，主要是因为连接数据库不熟练，EF框架也不熟练的缘故，这次试验，大致上初步了解了EF框架的用法，感觉还是需要多多熟练的。