实 验 报 告 单

**实验名称： 实验3：Windows窗体程序实验 1**

同组人：无 实验课时：4

实验室：计算机软件实验室 报告日期：2022年4月30日

**一、实验目的：**

1. 能综合使用多种Windows控件；

2. 能合理使用ADO.NET和EF框架等开发技术访问数据库；

3. 能够跨平台部署数据库，设计数据库时考虑安全因素，降低数据库安全风险。

1. **实验内容：**
2. **数据库设计：**

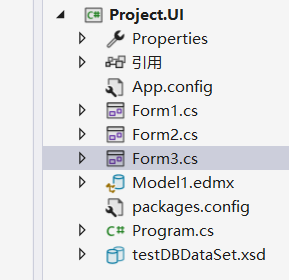
**表userInfo**

create table userInfo(uid int not null primary key identity,sname nvarchar(50) not null ,passwd nvarchar(50) not null)

**表test:**

create table test(sno int not null primary key identity,sname char(10) not null)

1. **程序结构：**



这里没有采用三层架构

Form 1是登录界面，form3是注册界面，form2是增删改查界面，model1.demx是数据库的连接文件和对应的数据库类；

三层架构

界面层ui

业务逻辑层，用来进行各种业务，增删改查

数据访问层，获取数据库数据，建立获取数据的类，使用ef框架来实现调用数据

另外还有model,建立用户类，用来与数据库建立联系。也即上面的model1.demx

1. **重点代码分析**
2. **访问数据库**

**UserInfo类**

public partial class userInfo

{

public int uid { get; set; }

public string passwd { get; set; }

public string sname { get; set; }

}

**TEST类**

public partial class test

{

public int sno { get; set; }

public string sname { get; set; }

}

这两个类主要是用来构造数据苦衷表的对象以进行数据库操纵

**按下增加一行数据**

test tes = new test

{

sno = 10,

sname = "曹操"

};

dbContext.Set<test>().Add(tes);

dbContext.SaveChanges();//做出更改后要保存

loadData();

**删除一行数据**

test tes = dbContext.Set<test>().Where(u=>u.sname == "刘七").FirstOrDefault();//

dbContext.Set<test>().Remove(tes);

dbContext.SaveChanges();

loadData();

**查找**

test tes = dbContext.Set<test>().Where(u=>u.sno == 1).FirstOrDefault();

MessageBox.Show(tes.sno + "," + tes.sname);

**展示数据库**

Void LoadData(){

dataGridView1.DataSource = dbContext.Set<test>().ToList();

}

**以下为剩余关键代码**

**窗体一：**

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

// 获取用户输入的信息

string username = tbx\_username.Text;

string password = tbx\_password.Text;

// 判断用户名和密码

if (username.Equals("zlc") && password.Equals("123"))

{

Form2 form2 = new Form2();

//form2.Show();

form2.Tag = tbx\_username.Text;

//form2.ShowDialog();

form2.Show();

//this.Close();

Hide();

}

else

{

MessageBox.Show("用户名或密码错误","错误",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

}

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

{

Close();

}

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

{

Form3 form3 = new Form3();

form3.Tag = tbx\_username.Text;

form3.Show();

}

}

**窗体二：**

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;

using System.Data.SqlClient;

namespace \_02窗体程序

{

public partial class Form2 : Form

{

MyTbContest dbContext = new MyTbContest();

public Form2()

{

InitializeComponent();

}

private void Form2\_Load(object sender, EventArgs e)

{

label1.Text = "欢迎" + this.Tag + label1.Text;

loadData();

}

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

{

Application.Exit();

}

private void Form2\_FormClosing(object sender, FormClosingEventArgs e)

{

}

List<test> list = new List<test>();

private void loadData()

{

//由于使用了ef框架使得原来的test类被重写

dataGridView1.DataSource = dbContext.Set<test>().ToList();

/\* // 1.创建连接字符串

string connStr = "Data Source=192.168.168.135;Initial Catalog=testDB;User ID=sa;PassWord=3712466ZLCjyajy";

// 2.创建连接对象

//SqlConnection conn = new SqlConnection(connStr);

using (SqlConnection conn = new SqlConnection(connStr))

{

// 3.编写sql语句

string sql = "select \* from test";

// 4.创建执行sql语句的命令对象

SqlCommand cmd = new SqlCommand(sql, conn);

// 5.打开数据库连接

conn.Open();

// 6.执行sql语句

//cmd.ExecuteNonQuery();

//cmd.ExecuteReader();

//cmd.ExecuteScalar();

SqlDataReader reader = cmd.ExecuteReader();

if (reader.HasRows)

{

list = new List<test>();

while (reader.Read())

{

// Console.WriteLine(reader["sno"] + " "// + reader["sname"]);

int id = reader.GetInt32(0);

string name = reader.GetString(1);

test test1 = new test(id, name);

list.Add(test1);

}

dataGridView1.DataSource = list;//数据绑定

}

else

{

Console.WriteLine("表中没有数据。");

}

}\*/

}

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

{

}

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

{

}

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

{

test tes = dbContext.Set<test>().Where(u=>u.sno == 1).FirstOrDefault();

MessageBox.Show(tes.sno + "," + tes.sname);

}

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

{

test tes = new test

{

sno = 10,

sname = "曹操"

};

dbContext.Set<test>().Add(tes);

dbContext.SaveChanges();//做出更改后要保存

loadData();

}

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

{

//tes是一条记录

test tes = dbContext.Set<test>().Where(u=>u.sname == "曹操").FirstOrDefault();

dbContext.Set<test>().Remove(tes);

dbContext.SaveChanges();

loadData();

}

private void button1\_Click\_2(object sender, EventArgs e)

{

test tes = dbContext.Set<test>().Where(u=>u.sno == 1).FirstOrDefault();

tes.sname = "张无忌";

dbContext.SaveChanges();

loadData();

}

}

}

**加密：**

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

{/\*

//这里没有分层，添加userinfo数据，加密，修改合为一体 其实应该分开的，这里输入完数据，注释掉添加

//添加数据

userInfo info = new userInfo

{

uid = Convert.ToInt32(textBox1.Text),

passwd = zhuceTb\_pass.Text

};

dbContest.Set<userInfo>().Add(info);

dbContest.SaveChanges();\*/

//调用加密方法

string md5 = Common.Md5Helper.GetMd5(zhuceTb\_pass.Text);

// MessageBox.Show(md5);

//更改内容为密文

//这里要注意 LINQ to Entities 不识别方法“Int32 ToInt32(System.String）

var y = Convert.ToInt32(textBox1.Text);//这样就能实现

userInfo res = dbContest.Set<userInfo>().Where(u => u.uid == 123).FirstOrDefault();

res.passwd = md5.Substring(0,15);

dbContest.SaveChanges();

userInfo info = dbContest.Set<userInfo>().Where(u => u.uid == 123 && u.passwd == md5).FirstOrDefault();

if(info != null)

{

//将得到的记录更改

MessageBox.Show(md5);

//这里出现了问题，发现是密文太长了，于是更改passwd类型

dbContest.SaveChanges();//做出更改后要保存

MessageBox.Show(info.uid.ToString());

}

//这里可以重新写一个登录项目，login类，并在此调用

Application.Exit();

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

namespace Common

{

public class Md5Helper

{

public static string GetMd5(string text)

{

//创建md5对象

MD5 md5 = MD5.Create();

//将字符串转成字节数组

byte[] bs = Encoding.UTF8.GetBytes(text);

//加密

byte[] bs2 =md5.ComputeHash(bs);

//将字节数组转换成字符串

StringBuilder stringBuilder=new StringBuilder();

for (int i = 0; i < bs2.Length; i++)

{

//0-255=>00-ff

stringBuilder.Append(bs2[i].ToString("x2").ToLower());

}

return stringBuilder.ToString();

}

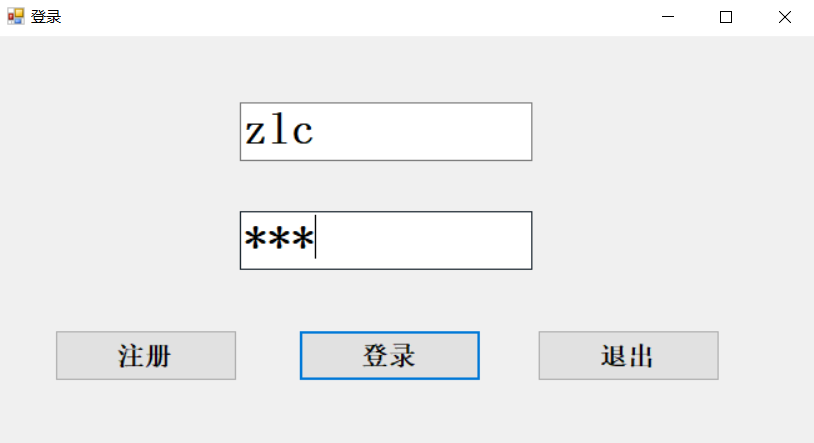
}

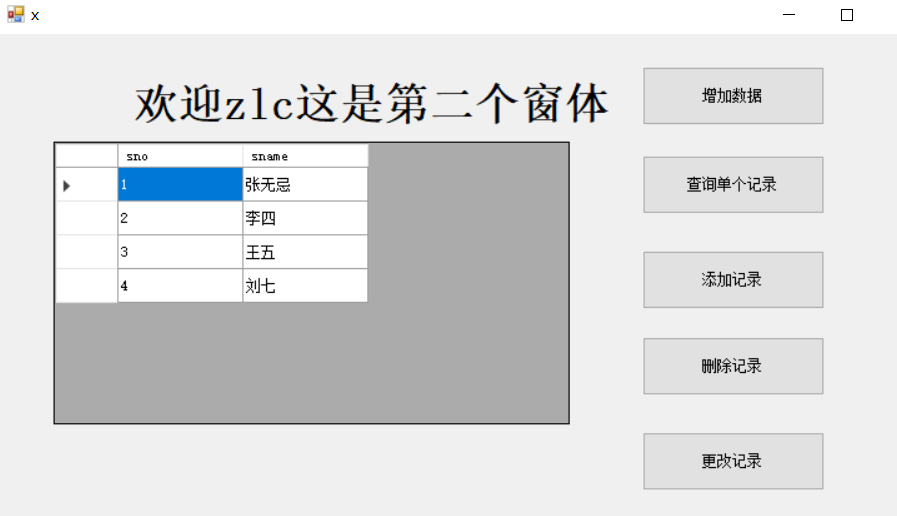
}

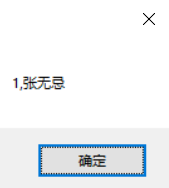
1. **实验结果：**

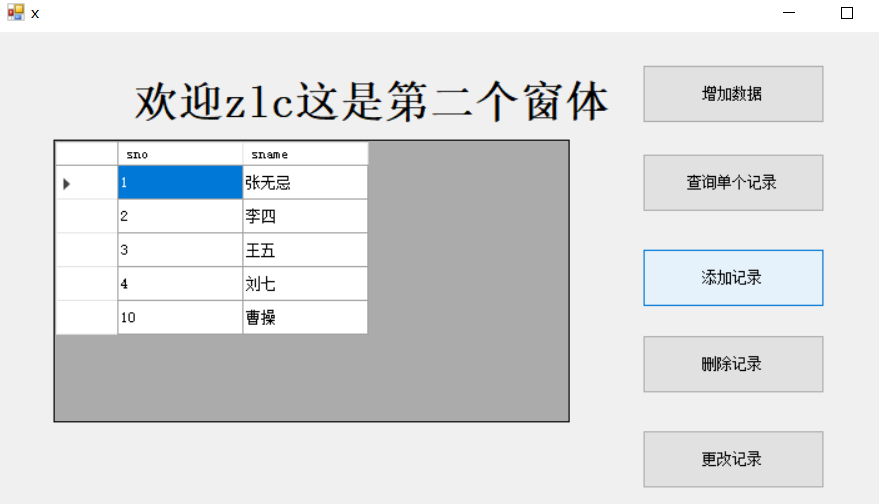
（带有登录功能的窗体应用程序，实现访问数据库的CRUD操作）

登录

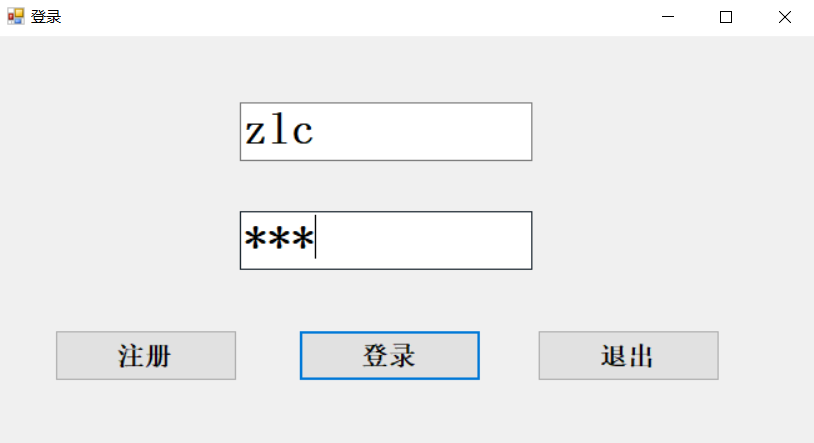


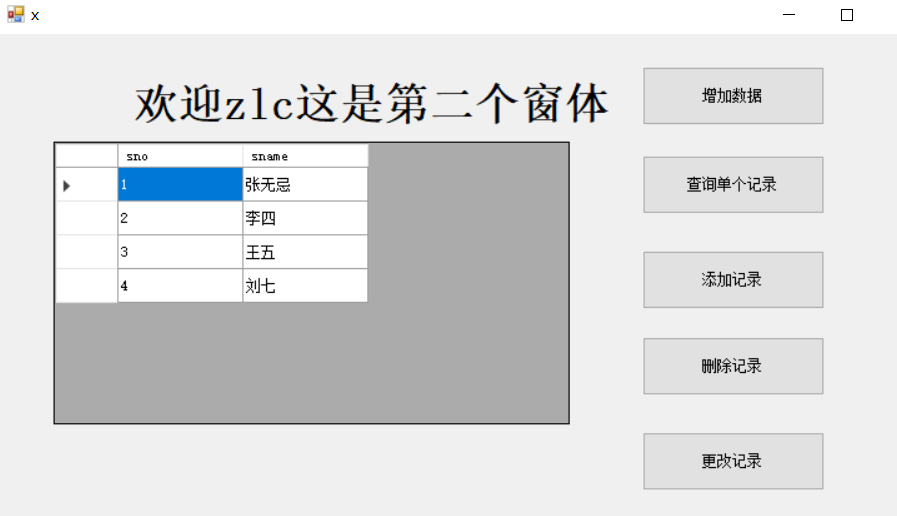


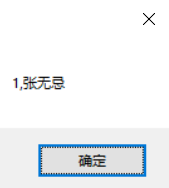




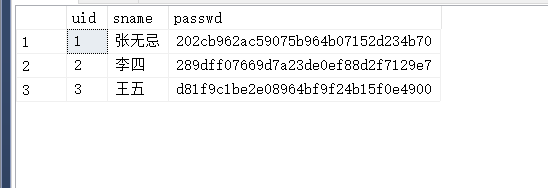
（带有登录功能的窗体应用程序，实现访问数据库的CRUD操作）



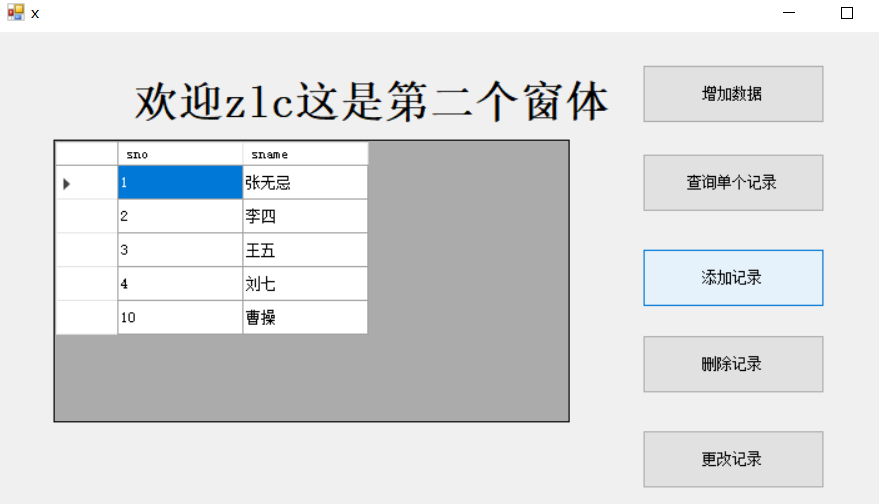




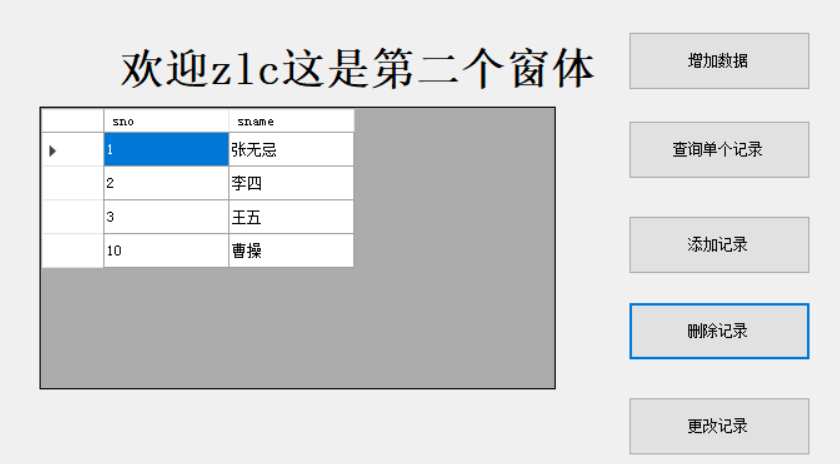
加密后

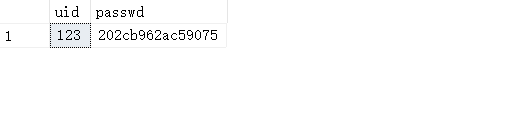


增加



删除





1. **实验总结：**

**通过这次实验，基本了解了vs连接虚拟机上的sqlserver数据库，使用ef框架进行数据库的增删改查。基本了解了创建登录界面的逻辑与其中的框架；了解了数据库敏感信息的加密原理。虽然没有使用三层架构，但也基本了解了其使用方法。**

———————————————————————————————

**成绩：**

批阅教师：

日 期：