****

《面向对象程序设计》

课程设计报告书

**题目： 客户信息管理系统设计**

**学  院： 数学学院**

**专  业： 数学与应用数学**

**学生姓名： 雷皓兰**

**学生学号： 202130320717**

**指导教师： 梁景鸿**

**课程学分： 2**

**起止日期：** 2022年12月15日—2022年12月31日

目录

[一、 题目简介 2](#_Toc123398859)

[二、 系统概述 2](#_Toc123398860)

[三、 系统设计 2](#_Toc123398861)

[四、 关键技术 5](#_Toc123398862)

[五、 效果及存在问题 5](#_Toc123398863)

[六、 心得体会 6](#_Toc123398864)

[七、 附录 6](#_Toc123398865)

[八、 参考文献 17](#_Toc123398866)

# 题目简介

我的课程设计是客户信息管理系统，用来管理客户的基本信息、销售员信息、商品及其购买信息、售后信息。客户、销售员和管理员可以分别通过前端界面，执行相应操作。

# 系统概述

登录和注册界面客户、销售员和管理员可以进行注册和登录。

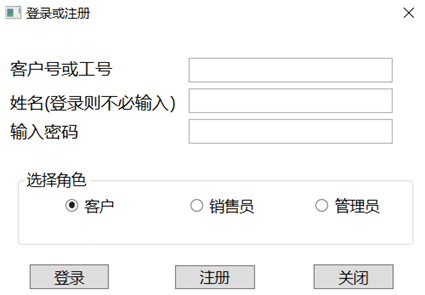
客户能查询到自己的购买信息，并对购买的商品提出售后问题。

销售员可以录入客户的信息、商品购买信息；可以回复售后问题。

管理员可以增加、修改销售员信息；对客户、销售员、商品、购买、售后信息进行条件查询、汇总、排序。

# 系统设计

首先打开运行是登录和注册界面如下，客户、销售员、管理员可登录可注册。



注册后显示注册成功：



客户注册登录后可直接查看自己的购买信息，并直接提出售后问题、查看售后回复。比如：



销售员登录后可看到自己信息并有三个窗口选项：

文本, 电子邮件

描述已自动生成

点击“录入客户信息”：

图形用户界面, 应用程序

描述已自动生成

点击“录入商品信息”：（考虑到《西游记》里面的东西大多都是孤品，所以商品没有余量，只有购买与否和购买者）



点击“回复售后问题”，可查看商品的售后问题并直接回复。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

管理者注册并登录后可看到自己的信息并有两个窗口选项：

文本

描述已自动生成

点击“管理销售员信息”，可对管理员的信息进行增、删、改：

表格

中度可信度描述已自动生成

点击“查询、汇总、排序各信息”可对客户和商品的信息进行查询和汇总：

文本

中度可信度描述已自动生成

# 关键技术

1. 使用了SQLConnection类和数据库建立连接，连接了本地的SQL Server文件，包括若干个表，可在表中添入数据(若干记录)。实现了用户的密码注册登录，并简单地显示数据库中信息。
2. 使用了SQLCommand类查询对象方法，可直接在数据库的表中查询指定记录。
3. 建立“项目数据源”，使用SQLDataAdapter和DataSet类实现数据的选择、插入、删除、等功能。用Update（）方法实现信息的更新，并保存到数据库；用FillBy和GetDataBy实现查询功能。
4. 使用DataTable类实现编辑、显示数据功能。
5. 利用RadioBottom控件实现对不同对象的选择，不同的对象有不同的操作、跳出不同界面。
6. 使用Listview显示数据，用TextBox控件的Text属性绑定DataSet的数据，使其能够显示出来。
7. 在汇总数据时利用“主从关系”，因为不同商品由不同客户购买，选定某一客户时，便只会显示该客户购买的商品。
8. 利用MassageBox显示各种错误或者意外情况。
9. 利用ComboBox控件下拉列表显示查询条件。
10. 使用MouseDoubleClick和MouseLeave事件辅助实现数据编辑功能。

# 效果及存在问题

效果如上述系统设计所示。

存在问题：

1. 对商品、客户、销售员绑定相应的图片失败，显示不够生动形象。
2. 商品只能是孤品，无法显示余量，系统稍显僵硬。
3. 查询、排序的功能还需完善，系统功能还需完善。

# 心得体会

虽然过程有点坎坷，但总算是可以自己做出指定功能的系统了，可增删改、查询等，可以和不同的对象互动，有灵活性，在实际生活中的也可以应用，有收获，做完有成就感。

# 附录

连接数据库：

private bool DBConn()

{ StreamReader sr = null;

bool blnConnSuccess = true;

try

{ sr = File.OpenText(@".\sqlconn.txt");

while (sr.Peek() != -1)

{ SqlConnTxt += sr.ReadLine();}

if (SqlConnTxt == "")

throw new Exception("数据库（文件）连接字串定义异常！");

}

catch (DirectoryNotFoundException e1)

{ MessageBox.Show(e1.Message + "配置文件出错", "出错！", MessageBoxButton.OK, MessageBoxImage.Error);

blnConnSuccess = false; }

catch (FileNotFoundException e1)

{MessageBox.Show(e1.Message + "配置文件出错！", "出错！", MessageBoxButton.OK, MessageBoxImage.Error);

blnConnSuccess = false; }

catch (Exception e1)

{ MessageBox.Show(e1.Message + "退出程序！", "出错！", MessageBoxButton.OK, MessageBoxImage.Error);

blnConnSuccess = false; }

finally

{if (sr != null)

sr.Close();}

if (blnConnSuccess)

{conn = new SqlConnection(SqlConnTxt);

try

{ conn.Open();

conn.Close();}

catch (SqlException e)

{ MessageBox.Show(e.Message, "连接不成功，请检查环境配置再试。", MessageBoxButton.OK, MessageBoxImage.Error);

blnConnSuccess = false; }}

return blnConnSuccess; }

private void Window\_Loaded(object sender, RoutedEventArgs e)

{if (!DBConn())

{ MessageBox.Show("连接数据库失败,程序中止，请与开发者联系！", "出错", MessageBoxButton.OK, MessageBoxImage.Error);

Application.Current.Shutdown();

this.Close();}}

注册：

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

{if (textBox1.Text == "" || textBox2.Text == "" || passwordBox1.Password == "")

{ MessageBox.Show("所有项都必须填写！");

return; }

conn = new SqlConnection(SqlConnTxt);

string s = RadioButtonState;

switch (s)

{ case "销售员":

string txt2 = "Insert Into Saler(SalerNum,SalerName,SalerPassword) Values(' ";

txt2 += textBox1.Text + "',N'";

txt2 += textBox2.Text + "','";

txt2 += passwordBox1.Password + "')";

conn.Open();

da = new SqlCommand(txt2, conn);

da.ExecuteNonQuery();

MessageBox.Show("注册成功！", "提示", MessageBoxButton.OK, MessageBoxImage.Asterisk);

textBox1.Text = "";

textBox2.Text = "";

passwordBox1.Password = "";

conn.Close();

break;

case "客户":

string txt1 = "Insert Into Customer(CustomerNum,CustomerName,CustomerPassword) Values(' ";

txt1 += textBox1.Text + "',N'";

txt1 += textBox2.Text + "','";

txt1 += passwordBox1.Password + "')";

conn.Open();

da = new SqlCommand(txt1, conn);

da.ExecuteNonQuery();

MessageBox.Show("注册成功！", "提示", MessageBoxButton.OK, MessageBoxImage.Asterisk);

textBox1.Text = "";

textBox2.Text = "";

passwordBox1.Password = "";

conn.Close();

break;

case "管理员":

string txt3 = "Insert Into Manager(ManagerNum,ManagerName,ManagerPassword) Values(' ";

txt3 += textBox1.Text + "',N'";

txt3 += textBox2.Text + "','";

txt3 += passwordBox1.Password + "')";

conn.Open();

da = new SqlCommand(txt3, conn);

da.ExecuteNonQuery();

MessageBox.Show("注册成功！", "提示", MessageBoxButton.OK, MessageBoxImage.Asterisk);

textBox1.Text = "";

textBox2.Text = "";

passwordBox1.Password = "";

conn.Close();

break; }}

登录：

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

{string s = RadioButtonState;

DataTable table, CoreTable;

switch (s)

{case "客户":

CustomerMisDataSetTableAdapters.CustomerTableAdapter CuTa =

new CustomerMisDataSetTableAdapters.CustomerTableAdapter();

int n = int.Parse(textBox1.Text);

table = CuTa.GetDataByCustomerNum(n);

if (table.Rows.Count != 0)

{ if (table.Rows[0]["Customerpassword"].ToString() == passwordBox1.Password)

{ CustomerMisDataSetTableAdapters.CommodityTableAdapter comTa = new CustomerMisDataSetTableAdapters.CommodityTableAdapter();

CoreTable = comTa.GetDataByCustomerNum(n);

ShowOneCustomerData sd = new ShowOneCustomerData(table, CoreTable);

sd.Show();}

else MessageBox.Show("密码错误");}

else MessageBox.Show("查无此人");

break;

case "销售员":

CustomerMisDataSetTableAdapters.SalerTableAdapter saTa =

new CustomerMisDataSetTableAdapters.SalerTableAdapter();

int i = int.Parse(textBox1.Text);

table = saTa.GetDataBySalerNum(i);

if (table.Rows.Count != 0)

{

if (table.Rows[0]["Salerpassword"].ToString() == passwordBox1.Password)

{ ToSaler id = new ToSaler(table);

id.Show(); }

else MessageBox.Show("密码错误"); }

else

MessageBox.Show("查无此人");

break;

case "管理员":

CustomerMisDataSetTableAdapters.ManagerTableAdapter maTa =

new CustomerMisDataSetTableAdapters.ManagerTableAdapter();

int m = int.Parse(textBox1.Text);

table = maTa.GetDataByManagerNum(m);

if (table.Rows.Count != 0)

{ if (table.Rows[0]["Managerpassword"].ToString() == passwordBox1.Password)

{ ToManager md = new ToManager(table);

md.Show(); }

else

MessageBox.Show("密码错误"); }

else

MessageBox.Show("查无此人");

break; } }

客户界面：

public ShowOneCustomerData(DataTable table1,DataTable table2)

{ InitializeComponent();

textBlock3.DataContext = table1;

textBlock4.DataContext = table1;

string sc = @"Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=D:\CustomerMis.mdf;

Integrated Security=True;

Connect Timeout=30";

conn = new SqlConnection(sc);

string s = "select \* from Commodity";

da = new SqlDataAdapter(s, conn);

SqlCommandBuilder scb = new SqlCommandBuilder(da);

ds = new DataSet();

da.Fill(ds, "Commodity");

view = new DataView();

view.Table = ds.Tables["Commodity"];

view.AllowEdit = true;

view.AllowDelete = true;

view.AllowNew = true;

listview1.ItemsSource=table2.DefaultView;

}

private void TextBox\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{TextBox textBox = (TextBox)sender;

textBox.IsReadOnly = false; }

private void TextBox\_MouseLeave(object sender, MouseEventArgs e)

{TextBox textBox = (TextBox)sender;

textBox.IsReadOnly = true;

if (ds.HasChanges())

da.Update(ds, "Commodity"); } }}

销售员录入客户信息：public partial class InputCustomer : Window

{

SqlCommand da;

SqlConnection conn;

SqlDataAdapter Da;

DataSet ds;

public InputCustomer()

{InitializeComponent();}

private void Button\_Click(object sender, RoutedEventArgs e)

{ if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == "")

{

MessageBox.Show("所有项都必须填写！");

return;

}

string sc = @"Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=D:\CustomerMis.mdf;

Integrated Security=True;

Connect Timeout=30";

SqlConnection conn = new SqlConnection(sc);

string txt1 = "Insert Into Customer(CustomerNum,CustomerName,CustomerPassword) Values(' ";

txt1 += textBox1.Text + "',N'";

txt1 += textBox2.Text + "','";

txt1 += textBox3.Text + "')";

conn.Open();

da = new SqlCommand(txt1, conn);

da.ExecuteNonQuery();

MessageBox.Show("录入成功！", "提示", MessageBoxButton.OK, MessageBoxImage.Asterisk);

conn.Close();

string txtCommand = "SELECT \* FROM Customer";

Da = new SqlDataAdapter(txtCommand, conn);

ds = new DataSet();

Da.Fill(ds, "Customer");

grid1.DataContext = ds.Tables["Customer"].DefaultView; }

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

} }}

录入商品信息：

public partial class AddCommodity : Window

{

SqlConnection conn;

SqlDataAdapter da;

DataSet ds;

DataView view;

public AddCommodity()

{

InitializeComponent();

string sc = @"Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=D:\CustomerMis.mdf;

Integrated Security=True;

Connect Timeout=30";

conn = new SqlConnection(sc);

string s = "select \* from Commodity";

da = new SqlDataAdapter(s, conn);

SqlCommandBuilder scb = new SqlCommandBuilder(da);

ds = new DataSet();

da.Fill(ds, "Commodity");

view = new DataView();

view.Table = ds.Tables["Commodity"];

view.AllowEdit = true;

listView1.ItemsSource = ds.Tables["Commodity"].DefaultView;

}

private void TextBox\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

TextBox textBox = (TextBox)sender;

textBox.IsReadOnly = false;

}

private void TextBox\_MouseLeave(object sender, MouseEventArgs e)

{

TextBox textBox = (TextBox)sender;

textBox.IsReadOnly = true;

if (ds.HasChanges())

da.Update(ds, "Commodity");

}

private void ClearAddBox()

{

textBox1.Text = ""; textBox2.Text = ""; textBox3.Text = ""; textBox4.Text = "";

}

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

{

DataRowView drv = (DataRowView)listView1.SelectedItem;

drv.Delete();

da.Update(ds, "Commodity");

}

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

{

string s1 = textBox1.Text, s2 = textBox2.Text, s3 = textBox3.Text;

if (s1.Length \* s2.Length \* s3.Length > 0)

{

DataView dv = (DataView)listView1.ItemsSource;

DataRowView drv = dv.AddNew();

drv["CommodityNum"] = textBox1.Text;

drv["CommodityName"] = textBox2.Text;

drv["Price"] = textBox3.Text;

drv["CustomerName"] = textBox4.Text;

drv.EndEdit();

da.Update(ds, "Commodity");

ClearAddBox();

}

else MessageBox.Show("请输入全部信息！", "新增记录资料不齐", MessageBoxButton.OK, MessageBoxImage.Error);

} }}

回复售后信息：

public partial class ReplyPropose : Window

{

SqlConnection conn;

SqlDataAdapter da;

DataSet ds;

DataView view;

public ReplyPropose()

{

InitializeComponent();

string sc = @"Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=D:\CustomerMis.mdf;

Integrated Security=True;

Connect Timeout=30";

conn = new SqlConnection(sc);

string s = "select \* from Commodity";

da = new SqlDataAdapter(s, conn);

SqlCommandBuilder scb = new SqlCommandBuilder(da);

ds = new DataSet();

da.Fill(ds, "Commodity");

view = new DataView();

view.Table = ds.Tables["Commodity"];

view.AllowEdit = true;

listView1.ItemsSource = ds.Tables["Commodity"].DefaultView;

}

private void TextBox\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

TextBox textBox = (TextBox)sender;

textBox.IsReadOnly = false;

}

private void TextBox\_MouseLeave(object sender, MouseEventArgs e)

{

TextBox textBox = (TextBox)sender;

textBox.IsReadOnly = true;

if (ds.HasChanges())

da.Update(ds, "Commodity");

} }}

管理员管理销售员信息：

public partial class ManageSalerData : Window

{

SqlConnection conn;

SqlDataAdapter da;

DataSet ds;

DataView view;

public ManageSalerData()

{

InitializeComponent();

string sc = @"Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=D:\CustomerMis.mdf;

Integrated Security=True;

Connect Timeout=30";

conn = new SqlConnection(sc);

string s = "select \* from Saler";

da = new SqlDataAdapter(s, conn);

SqlCommandBuilder scb = new SqlCommandBuilder(da);

ds = new DataSet();

da.Fill(ds, "Saler");

view = new DataView();

view.Table = ds.Tables["Saler"];

view.AllowDelete = true;

view.AllowEdit = true;

view.AllowNew = true;

listView1.ItemsSource = ds.Tables["Saler"].DefaultView;

}

private void TextBox\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

TextBox textBox = (TextBox)sender;

textBox.IsReadOnly = false;

}

private void TextBox\_MouseLeave(object sender, MouseEventArgs e)

{

TextBox textBox = (TextBox)sender;

textBox.IsReadOnly = true;

if (ds.HasChanges())

da.Update(ds, "Saler");

}

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

{

DataRowView drv = (DataRowView)listView1.SelectedItem;

drv.Delete();

da.Update(ds, "Saler");

}

private void ClearAddBox()

{

textBox1.Text = ""; textBox2.Text = ""; textBox3.Text = "";

}

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

{

string s1 = textBox1.Text, s2 = textBox2.Text, s3 = textBox3.Text;

if (s1.Length \* s2.Length \* s3.Length > 0)

{

DataView dv = (DataView)listView1.ItemsSource;

DataRowView drv = dv.AddNew();

drv["SalerNum"] = textBox1.Text;

drv["SalerName"] = textBox2.Text;

drv["SalerPassword"] = textBox3.Text;

drv.EndEdit();

da.Update(ds, "Saler");

ClearAddBox();

}

else MessageBox.Show("请输入全部信息！", "新增记录资料不齐", MessageBoxButton.OK, MessageBoxImage.Error);

} } }

查询、汇总：

public partial class Gather : Window

{

CustomerMisDataSet ds;

CustomerMisDataSetTableAdapters.CustomerTableAdapter customerAdapter;

CustomerMisDataSetTableAdapters.CommodityTableAdapter commodityAdapter;

DataView view;

public Gather()

{

InitializeComponent();

ds = new CustomerMisDataSet();

customerAdapter = new CustomerMisDataSetTableAdapters.CustomerTableAdapter();

commodityAdapter = new CustomerMisDataSetTableAdapters.CommodityTableAdapter();

customerAdapter.Fill(ds.Customer);

commodityAdapter.Fill(ds.Commodity);

view = new DataView();

view.Table = ds.Tables["Commodity"];

listView1.ItemsSource = ds.Tables["Customer"].DefaultView;

listView1.SelectedIndex = 0;

image1.DataContext = ds.Tables["Customer"].DefaultView;

}

private void listView1\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

DataRowView drv = (DataRowView)listView1.SelectedItem;

view.RowFilter = "CustomerNum = " +

drv["CustomerNum"].ToString();

listView2.ItemsSource = view;

}

private void comboBox1\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

comboBox2.Items.Clear();

ListBoxItem lbi = comboBox1.SelectedItem as ListBoxItem;

string s = lbi.Content.ToString();

if (s == "客户号")

{

comboBox2.Items.Add(">");

comboBox2.Items.Add(">=");

comboBox2.Items.Add("<");

comboBox2.Items.Add("<=");

comboBox2.Items.Add("=");

comboBox2.Items.Add("<>");

}

else if (s == "姓名")

{

comboBox2.Items.Add("=");

comboBox2.Items.Add("like");

}

else comboBox2.Items.Add("=");

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

string filterString;

if (comboBox1.Text == "" || comboBox2.Text == "" || textBox1.Text == "")

{

MessageBox.Show("请将过滤条件填写完整！");

return;

}

if (comboBox1.Text == "客户号")

{ filterString = "CustomerNum " + comboBox2.Text + " " + textBox1.Text; }

else if (comboBox1.Text == "姓名")

{

if (comboBox2.Text == "=")

filterString = "CustomerName " + comboBox2.Text + " '" +textBox1.Text + "'";

else

filterString = "CustomerName " + comboBox2.Text + " '" + textBox1.Text + "%'";

}

else filterString = "studentsex " + comboBox2.Text + " '" + textBox1.Text + "'";

string sc = @"Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=D:\CustomerMis.mdf;

Integrated Security=True;

Connect Timeout=30";

SqlConnection conn = new SqlConnection(sc);

string s = "SELECT \* FROM Customer WHERE " + filterString;

SqlDataAdapter da = new SqlDataAdapter(s, conn);

DataSet ds1 = new DataSet();

da.Fill(ds1, "Customer");

CustomerMisDataSetTableAdapters.CustomerNumTableTableAdapter snta = new

CustomerMisDataSetTableAdapters.CustomerNumTableTableAdapter();

snta.DeleteAll();

snta.Fill(ds.CustomerNumTable);

DataTable dt = ds1.Tables["Customer"];

for (int i = 0; i < dt.Rows.Count; i++)

{

DataRow row = ds.Tables["CustomerNumTable"].NewRow();

row["CustomerNum"] = dt.Rows[i]["CustomerNum"];

ds.Tables["CustomerNumTable"].Rows.Add(row);

}

snta.Update(ds.CustomerNumTable);

customerAdapter.FillInCustomerNumTable(ds.Customer);

commodityAdapter.FillInCustomerNumTable(ds.Commodity);

listView1.DataContext = ds.Tables["Customer"];

listView2.DataContext = ds.Tables["Customer"];

listView1.SelectedIndex = 0;

} } }

# 参考文献

【1】《C#应用程序设计教程（第二版）（人民邮电出版社）》

|  |  |
| --- | --- |
| 教  师  评  语 | 教师签名：  日期： |
| 成  绩  评  定 |  |
| 备  注 |  |