# 目 录

[**目 录** 1](#_Toc472455025)

[**摘 要** 2](#_Toc472455026)

[**1.** **家庭记账软件背景及需求分析** 3](#_Toc472455027)

[**1.1．开发背景及意义** 3](#_Toc472455028)

[**1.2．适应人群** 3](#_Toc472455029)

[**1.3．主要开发内容** 3](#_Toc472455033)

[**1.4．需求分析** 3](#_Toc472455034)

[**2.** **家庭记账软件系统设计** 4](#_Toc472455035)

[**2.1．家庭记账软件总体结构设计** 4](#_Toc472455037)

[**2.2．SQL Server数据库概念结构设计** 5](#_Toc472455038)

[**2.3．SQL Server数据库逻辑结构设计** 6](#_Toc472455039)

[**3.** **家庭记账软件系统开发平台** 6](#_Toc472455040)

[**3.1．Visual Studio 2010简介** 6](#_Toc472455041)

[**3.2．C#开发语言简介** 7](#_Toc472455042)

[**3.3．SQL Server 2008 简介** 7](#_Toc472455043)

[**3.4．Access 2010简介** 7](#_Toc472455044)

[**4.** **家庭记账软件系统实现** 8](#_Toc472455045)

[**4.1．登录界面的实现** 8](#_Toc472455046)

[**4.2．选项功能的实现** 10](#_Toc472455047)

[**4.3．财务功能的实现** 12](#_Toc472455048)

[**4.4．报表功能的实现** 17](#_Toc472455049)

[**4.5．账户管理功能的实现** 20](#_Toc472455050)

[**4.6．程序设置功能的实现** 23](#_Toc472455051)

[**5.** **家庭记账软件系统测试** 24](#_Toc472455052)

[**5.1．登录界面的测试** 24](#_Toc472455053)

[**5.2．主要功能模块的测试** 26](#_Toc472455054)

[**5.3．辅助功能的测试** 42](#_Toc472455055)

[**6.** **总结** 46](#_Toc472455056)

[**6.1．设计过程中遇到的问题** 46](#_Toc472455057)

[**6.2．心得体会** 46](#_Toc472455058)

[**附 录** 48](#_Toc472455059)

**摘 要**

随着科学技术和信息技术的发展，个人计算机在家庭中的使用变得越来越常见了，家庭的记账工作因此也能通过计算机软件来方便快捷地实现。

家庭记账软件是一款运行在PC端的.NET应用程序，它使用SQL Server数据库和C#程序开发语言作为其软件主要的实现方式，它给用户带来了方便、快捷的使用感受以及全新的记账体验。

在软件的前期设计阶段，需要对软件的使用对象进行一些必要的需求分析，分析其使用的系统环境、使用需求等，以便确定软件应使用哪种数据库、哪种.NET版本等先决条件，再根据开发需求，对软件进行整体的系统设计。进入开发阶段后，则需对软件进行功能模块的划分，使软件开发更有效率的完成。在代码编写环节，更需注意对代码风格的控制，以提高每个变量的辨识度；对于数据库语句的编写则必须做到没有差错，如果数据库语句错误，势必将会对软件的功能产生影响，甚至使软件无法正常使用。

在软件使用方面，软件分为普通用户和管理员用户两类，管理员用户可对其它用户进行一些必要的修改，普通用户仅能对其本身进行使用、修改以及查询。为了方便用户的使用，软件还提供了四个辅助功能，计算器功能、日历功能、汇率转换功能以及邮件发送功能，可在用户使用记账功能的同时方便用户临时使用。

**关键词：数据库、记账软件、.NET、C#、SQL Server**

1. **家庭记账软件背景及需求分析**

**1.1．开发背景及意义**

随着信息科技的发展，人们的生活水平日益提高，越来越多的家庭都会或多或少的拥有家用个人计算机。平时，人们通常都会在日常生活中进行消费活动，在工作中或者在其他方面获得收入。通过家庭记账软件的记账，我们能十分方便的记录日常的消费行为以及所获得的收入情况，并且通过统计报表清晰的了解自己及家庭的收支情况。

通过这款记账软件，可以灵活地修改、删除一些错误的记账数据，也可以补记一些以往由于种种原因忘记记录的账单，同时又能使家庭记账变得更加方便、快捷，避免了纸质账单频繁涂改以及由于保存不当造成的污损，也能使一些平时不太记账、懒于记账、理财水平欠佳的人对家庭记账产生一定的兴趣，对自己的收支情况产生一定的了解，使他们能更好地管理自己的生活花销，提升自己金钱的利用效能。

**1.2．适应人群**

本软件适合用于家庭成员之间的日常生活记账，能够较为详实的方式记录各家庭成员之间的收支行为。



**1.3．主要开发内容**

本软件的开发主要包括SQL Server数据库表格及字段定义、软件主体程序的编写，以及相关附加程序的编写，并包括用于存储连接字符串及程序设置的Access数据库的mdb文件。在软件主体方面，将采用C#语言进行程序编写，软件大致分为主要功能和辅助功能两部分。主要功能实现记账软件的记账、报表查询功能，辅助功能则主要实现一些基本的辅助性小功能：例如计算器、日历、汇率转换等。

**1.4．需求分析**

对于家庭用户而言，数据的吞吐量并不是很大，因此，使用SQL Server这类中型数据库可完全满足用户需求。为了方便家庭用户更便捷的去使用这个软件，本程序可在登录界面单击连接设置按钮自行配置数据库连接设置，并测试是否连接成功。用户日常的记账主要包括收支类型、金额、收支项目、备注等，能较大程度的满足家庭用户的记账需求。对于报表查询功能，软件将提供每日收支统计情况、每月收支统计情况、年度收支统计情况以及收支类别统计情况四个报表查询功能`，能基本满足家庭用户的需求。

1. **家庭记账软件系统设计**

**2.1．家庭记账软件总体结构设计**

在登录界面设置清空、连接设置、登录三个按钮。清空按钮可清空登录窗体用户名、密码文本框当中的文字内容；连接设置按钮可打开数据库连接设置窗体，进行相应的数据库连接的设置；登录按钮则可判断用户名、密码是否输入正确，若正确则进入记账软件的主界面。同时，该登录界面还给用户提供了更为快捷的登录方式，即通过键盘的回车换行，就可直接切换到下一个文本框的输入，最后一个密码文本框回车键则实现登录按钮的功能。结构示意图如图2.1所示。

清空

连接设置

登录

**图2.1 登录界面结构设计示意图**

在主界面设置菜单栏设置选项、财务、报表、账户管理、辅助功能、帮助几个较大的功能模块。在选项中设置了密码重设、切换用户、退出三个菜单选项，在财务菜单中设置了添加收支记录、删除收支记录、修改收支记录、管理收支记录四个功能菜单。在报表中设置了每日收支情况、每月收支情况、年度收支情况、收支类别统计情况。在账户管理中设置了添加账户、删除账户、修改账户信息三个功能菜单，管理员权限用户可全部显示上述菜单，而普通用户则仅显示修改账户信息菜单。在辅助功能中设置了计算器、日历、汇率转换、邮件发送功能。

选项

财务

报表

账户管理

辅助功能

帮助

密码重设

切换账户

退出

添加收支记录

删除收支记录

修改收支记录

管理收支类型

每日收支情况

每月收支情况

年度收支情况

收支类别统计

添加账户

删除账户

修改账户信息

计算器

日历

汇率换算

邮件发送

关于程序

程序设置

**图2.2主界面结构设计示意图**

在主界面的工具栏中设置了各菜单栏中较为常用的选项，例如切换用户、记账、计算器、日历、关于、设置等功能。并可在程序设置中可设置是否显示该工具栏。

在主界面的状态栏则显示“欢迎使用家庭记账软件”的欢迎词以及当前登录该软件的当前用户。该状态栏可在程序设置中选择是否显示。

主界面的背景也可在程序设置中进行选择，本程序共提供三个主界面背景以供用户选择。

**2.2．SQL Server数据库概念结构设计**

在数据库的设计中，数据库的名称定为AccountingSystem，数据库中共分为三个表，分别为：User表、Account表和Item表。

在User表中，主要由username、password、nickname、familycall、accountstyle五个字段组成。User表字段细节如下表2.1所示。

表2.1 用户数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键 | 非空 | 说明 |
| username | nvarchar(50) | Y | Y | 用户名 |
| password | nvarchar(50) | N | Y | 密码 |
| nickname | nvarchar(50) | N | Y | 昵称 |
| familycall | nvarchar(50) | N | N | 家庭称呼 |
| accountstyle | nvarchar(50) | N | Y | 账户类型 |

在Account表中，主要由username、date、money、item、type、memo、id七个字段组成。Account表字段细节如下表2.2所示。

表2.2 记账数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键 | 非空 | 说明 |
| username | nvarchar(50) | N | Y | 用户名 |
| date | Datetime | N | Y | 日期 |
| money | Money | N | Y | 金额 |
| item | nvarchar(50) | N | Y | 收支项目 |
| type | nvarchar(50) | N | Y | 收支类型 |
| memo | nvarchar(50) | N | N | 备注 |
| id | int | N | Y | 数据号（自动增加） |

在Item表中，主要由item、type两个字段组成。Item表字段细节如下表2.3所示。

表2.3 记账数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键 | 非空 | 说明 |
| item | nvarchar(50) | N | Y | 收支项目 |
| type | nvarchar(50) | N | Y | 收支类型 |

**2.3．SQL Server数据库逻辑结构设计**

在AccountingSystem数据库中，User表与Account表互相关联，username字段为User表的主键，User表中的一个username字段对应Account表中的多个username。

数据库E-R图的结构如图2.3所示。当User表中的username字段被删除后，将触发触发器，删除Account表中该用户的记账数据，下面代码为该触发器的创建代码。

Create TRIGGER trigUserDelete

On [User] FOR DELETE

as

Begin

declare @Sname nVARCHAR(50)

select @Sname=deleted.username from deleted

delete from [Account] where username=@Sname

end

由于Item表中的item以及type字段删除后，为了能让Account表中的数据保持不变，保证用户已记录的记账数据不丢失，将保留已删除的item字段，因此Item表在数据库中单独存在，仅供程序下拉菜单填充使用，不与其他两表形成逻辑上的关联。

用户

记录

记账数据

**1**

**n**

**图2.3 AccountSystem数据库的E-R图结构**

1. **家庭记账软件系统开发平台**

**3.1．Visual Studio 2010简介**

Visual Studio是微软公司开发的一款基于.Net的可视化开发工具。它是目前最流行的Windows平台应用程序开发环境。Visual Studio 2010版本于2010年4月12日上市，其集成开发环境的界面被重新设计和组织，变得更加简单明了。Visual Studio 2010同时带来了.NET Framework  4.0、Microsoft Visual Studio 2010 CTP(Community Technology Preview--CTP)，用其开发的应用程序可在多数版本的Windows操作系统上运行。除了支持SQL Server和Access数据库，它还支持 IBM DB2和Oracle数据库。

**3.2．C#开发语言简介**

C#语言是一款由微软设计、研发并用于.NET程序开发的面向对象程序设计语言。其语言风格简洁明了，既给应用程序的快速开发提供了方便，又保留了C++强大的开发功能，同时还支持跨平台开发，是一款强大、易用的程序开发语言。

Visual C#集成开发环境是一款基于C#语言的.NET程序开发平台，也是Visual Studio的重要组成部分。这款集成开发环境是微软为.NET应用程序开发而量身定制的开发环境，为.NET应用程序的开发提供了可靠的开发环境支持。在Windows窗体程序开发方面，其主要由项目创建、界面设计、代码编辑等方面构成，给开发者提供了良好的开发体验。

**3.3．SQL Server 2008 简介**

SOL Server 2008是微软公司开发的一款关系型数据库软件。它在微软的数据平台上发布，可以组织管理任何数据。可以将结构化、半结构化和非结构化文档的数据直接存储到数据库中。可以对数据进行查询、搜索、同步、报告和分析之类的操作。数据可以存储在各种设备上，从数据中心最大的服务器一直到桌面计算机和移动设备，它都可以控制数据而不用管数据存储在哪里。

SQL Server 2008 允许用户使用 .NET 和Visual Studio开发的自定义应用程序中使用SQL Server数据，在面向服务的架构（SOA）和通过 Microsoft BizTalk Server 进行的业务流程中使用数据。信息工作人员可以通过日常使用的工具直接访问数据，使数据的访问即方便又快捷。

**3.4．Access 2010简介**

Access 2010是微软公司开发的一款文件型数据库软件。Microsoft Office Access是微软把数据库引擎的图形用户界面和软件开发工具结合在一起的一个数据库管理系统。它是微软Office中的一个成员, 在包括专业版和更高版本的Office版本里面被单独出售。用户开发的程序可通过.NET直接访问数据库文件，而不需要安装及配置数据库服务，给人们带来了轻便快捷的感受。但其本身并不支持视图、存储过程、触发器等功能，在功能上有一定的局限性。本程序仅将其用于存储程序的设置。

1. **家庭记账软件系统实现**

**4.1．登录界面的实现**

登录窗体主要由关闭按钮、用户名框、密码框、登录按钮等构成。窗体界面如图所示。



**图4.1 登录窗体界面**

清空按钮可清空用户名、密码文本框内的用户名、密码。清空按钮的实现代码如下：

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

{

txtUsername.Text = ""; //清空用户名文本框

txtPassword.Text = ""; //清空密码文本框

}

登录按钮可实现对用户名和密码的查验，正确后打开主界面。其主要实现代码如下：

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

{

if (this.GetSqlConnectString() == false)

{

return; //获取连接字符串，判断是否成功，不成功则退出该事件

}

string strUsername, strPassword;

try

{

string connectstring = sqlConectstring;

string sqlcommand = "select username,password from [User]";//从User表中查找用户名、密码的数据库命令

SqlConnection sqlConnection = new SqlConnection(connectstring);

sqlConnection.Open();

SqlCommand sqlCommand = new SqlCommand(sqlcommand, sqlConnection);

SqlDataReader reader = sqlCommand.ExecuteReader();

while (reader.Read())

{

strUsername = reader.GetString(0);//读取用户名

strPassword = reader.GetString(1);//读取密码

if (txtUsername.Text.Trim() == strUsername.Trim() && txtPassword.Text.Trim() == strPassword.Trim())//比较文本框与数据库中的用户名、密码是否一致

{

reader.Close();

sqlConnection.Close();

frmMain MainWindow = new frmMain();

MainWindow.Tag = strUsername.Trim();//将当前登录的用户名传入主窗体

MainWindow.Show();//打开主窗体

this.Hide();//隐藏登录窗体

if (SetConnect.fSC!=null)//判断数据库连接设置窗体是否关闭

{

SetConnect.fSC.Close();//未关闭，则将其关闭

}

return;//退出该事件

}

}

MessageBox.Show("用户名或密码错误，登录失败", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "系统错误");

}

}

连接设置则可打开数据库连接的设置窗体，在该窗体内可设置该程序的SQL数据库连接并将数据库连接进行保存，其主要通过向Access数据库文件读取和存储数据实现程序功能。其窗体界面如下图4.2所示。

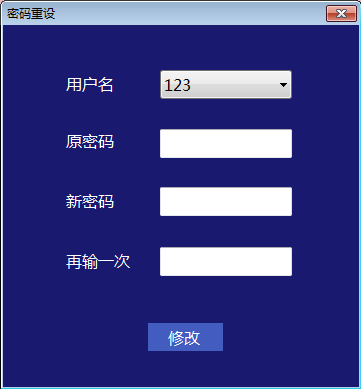


**图4.2 设置数据库连接的窗体界面**

**4.2．选项功能的实现**

选项主要由密码重设、切换用户、退出构成。

密码重设可重设用户的密码，普通用户仅能重设自己的密码，管理员用户则能重置其他用户的密码。其窗体界面如下图所示。



**图4.3 密码重设的窗体界面**

密码重设的主要实现代码如下：

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

{

if (TxtIsNull() == true)//调用窗体类中定义的判断是否为空的方法

{

return;

}

this.GetOldPassword();//调用窗体类中定义的获取原密码的方法

if (strOldPassword != txtOldPassword.Text)//判断原密码是否正确

{

MessageBox.Show("原密码错误", "错误");

return;

}

if (txtNewPassword.Text != txtNewPasswordAgain.Text)

{

MessageBox.Show("两次密码输入不一致", "错误");

return;

}

try

{

string connectstring = frmLogOn.f0.sqlConectstring;

string sqlcommand = "update [User] set [password]='" + txtNewPassword.Text + "'where [username]='"+cmbUsername.Text+"'";//更新User表中的密码字段的数据库命令

SqlConnection aConnection = new SqlConnection(connectstring);

aConnection.Open();//打开数据库连接

SqlCommand aCommand = new SqlCommand(sqlcommand, aConnection);

aCommand.ExecuteNonQuery();//执行数据库命令

aConnection.Close();

txtOldPassword.Text = "";

txtNewPassword.Text="";

txtNewPasswordAgain.Text = "";

MessageBox.Show("修改成功", "恭喜");

}

catch (Exception ex)//捕获异常事件

{

MessageBox.Show(ex.Message, "系统错误");

return;

}

}

切换用户主要是关闭当前主窗体以及辅助功能窗体，并将隐藏的登录窗体重新显示出来，并将密码框置为空。其主要实现代码如下：

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

{

this.Dispose();//关闭本窗体

if (ExchangeRate.fER != null)//判断汇率换算窗体是否关闭

{

ExchangeRate.fER.Close();

ExchangeRate.fER = null;

}

if (ExchangeRateSet.fERS != null)//判断汇率设定窗体是否关闭

{

ExchangeRateSet.fERS.Close();

ExchangeRateSet.fERS = null;

}

if (SendEmail.fSM != null)//判断邮件发送窗体是否关闭

{

SendEmail.fSM.Close();

SendEmail.fSM = null;

}

if (SendEmailSet.fSMS != null)//判断邮件发送设置窗体是否关闭

{

SendEmailSet.fSMS.Close();

SendEmailSet.fSMS = null;

}

if (frmcalc.IsDisposed == false)//判断计算器是否关闭

{

frmcalc.Close();

}

frmLogOn.f0.Show();//显示处在隐藏状态的窗体

frmLogOn.f0.ClearPassword();//调用登录窗体中定义的清除密码文本框方法

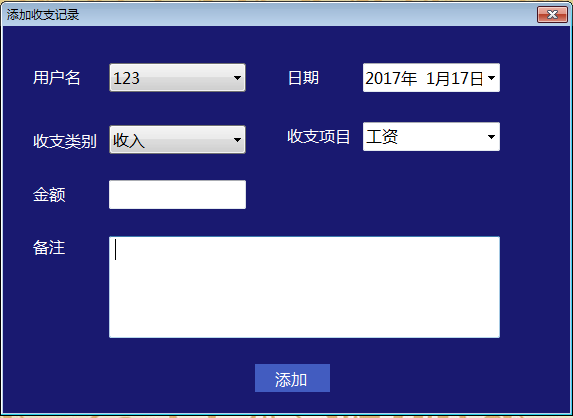
}

退出则是关闭软件，在软件被关闭前会自动弹出对话框提示是否关闭该软件。

**4.3．财务功能的实现**

财务主要由添加收支记录、删除收支记录、修改收支记录、管理收支类型构成。

添加收支记录主要实现用户对日常收支数据的格式化录入，通过添加按钮将用户录入的数据保存至数据库中。其窗体界面如图4.4所示。



**图4.4 添加收支记录的窗体界面**

添加收支记录的主要实现代码如下：

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

{

if (txtMoney.Text == "" || cmbItem.Text == "")//判断是否为空

{

MessageBox.Show("请将金额和收支项目填写完整", "系统错误");

return;

}

DateTime dt = dtpDate.Value;//定义时间类型的变量，将日期控件的值赋予其

string dtStr = dt.ToString("yyyy-MM-dd");//以格式化的方式将日期转化为字符串

try

{

string connectstring = frmLogOn.f0.sqlConectstring;//获取连接字符串

string sqlcommand = "insert into [Account]([username],[date],[money],[item],[type],[memo]) values('" + cmbUsername.Text + "','" + dtStr + "'," + txtMoney.Text + ",'" + cmbItem.Text + "','" + cmbType.Text + "','" + txtMemo.Text + "')";//向Account表中添加数据的数据库命令

SqlConnection aConnection = new SqlConnection(connectstring);

aConnection.Open();

SqlCommand aCommand = new SqlCommand(sqlcommand, aConnection);

aCommand.ExecuteNonQuery();

aConnection.Close();

txtMoney.Text = "0";//将文本框置为0

txtMemo.Text = "";//将文本框置为空

MessageBox.Show("添加成功", "恭喜");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "系统错误");

}

}

删除收支记录主要实现将数据库中已有的收支数据选择后，通过删除按钮将数据库中的收支数据进行删除。其窗体界面如图4.5所示。



**图4.5 删除收支记录的窗体界面**

删除收支记录的主要实现代码如下：

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

{

if (cmbId.Text == "")//判断用户是否已选要删除的数据

{

MessageBox.Show("请选择数据号", "系统错误");

return;

}

try

{

string connectstring = frmLogOn.f0.sqlConectstring;

string sqlcommand = "delete from [Account] where id=" + cmbId.Text;

SqlConnection aConnection = new SqlConnection(connectstring);

aConnection.Open();

SqlCommand aCommand = new SqlCommand(sqlcommand, aConnection);

aCommand.ExecuteNonQuery();

aConnection.Close();

this.GetAuthority();//获取当前用户权限

this.FillDgv();//填充数据表

this.FillId();//填充数据号下拉列表

this.ClearContent();//清空自动填写的已删除文本框内容

MessageBox.Show("删除成功", "恭喜");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "系统错误");

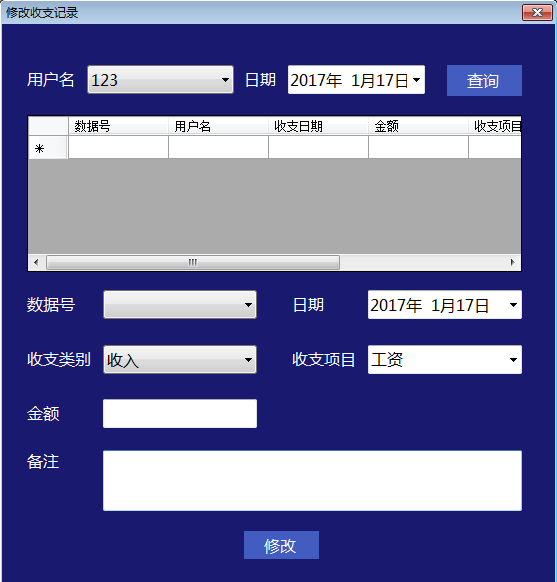
this.DeleteRecord\_Load(sender, e);

return;

}

}

修改收支记录主要实现将数据库中已有的收支数据更新为现有的收支数据，用户对原有收支数据的筛选，选择需要修改的收支数据，通过修改按钮将原有的收支记录进行更新操作。其窗体界面如图4.6所示。



**图4.6 修改收支记录的窗体界面**

管理收支类型主要实现对收支项目列表的删除与添加。通过添加按钮向数据库的Item表中添加收支类型及项目类型，通过删除按钮将软件中已选择的收支类型及收支项目删除。其窗体界面如图4.7所示。



**图4.7 管理收支类型的窗体界面**

管理收支类型的添加按钮主要实现代码如下：

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

{

string str1, str2;

if (cmbTypeAdd.Text == "" || txtAddItem.Text == "")

{

MessageBox.Show("所添加的内容为空", "系统错误");

return;

}

try

{

string connectstring = frmLogOn.f0.sqlConectstring;

string sqlcommand = "select [type],[item] from [Item]";

SqlConnection aConnection = new SqlConnection(connectstring);

aConnection.Open();

SqlCommand aCommand = new SqlCommand(sqlcommand, aConnection);

SqlDataReader reader = aCommand.ExecuteReader();

while (reader.Read())

{

str1 = reader.GetString(0);//读取收支类型

str2 = reader.GetString(1);//读取收支项目

if (cmbTypeAdd.Text.Trim() == str1.Trim() & txtAddItem.Text.Trim() == str2.Trim())//判断是否与已有的重复

{

reader.Close();

aConnection.Close();

MessageBox.Show("所添加的内容重复", "系统错误");

return;

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "系统错误");

}

try

{

string connectstring = frmLogOn.f0.sqlConectstring;

string sqlcommand = "insert into [Item]([item],[type]) values('" + txtAddItem.Text + "','" + cmbTypeAdd.Text + "')";

SqlConnection aConnection = new SqlConnection(connectstring);

aConnection.Open();

SqlCommand aCommand = new SqlCommand(sqlcommand, aConnection);

aCommand.ExecuteNonQuery();

aConnection.Close();

this.FillList();//调用窗体类中定义的填充列表方法

cmbTypeAdd.Text = "";

txtAddItem.Text = "";

MessageBox.Show("添加成功", "提示");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "添加失败");

return;

}

this.FillDeleteItem();

}

**4.4．报表功能的实现**

报表主要由每日收支情况、每月收支情况、年度收支情况以及收支类别统计情况构成。

每日收支情况主要实现个人以及家庭每日的收支情况统计，通过对数据库Account表的条件查询填充程序中的数据表，利用SQL中的sum函数统计共收入以及共支出的金额数量，并计算差值得出当日的盈余情况。其窗体界面如图4.8所示。



**图4.8 每日收支情况的窗体界面**

每日收支情况填充个人范围数据表的实现代码如下：

public void FillDgvPer()

{

DateTime dt = dtpDate.Value;

string dtStr = dt.ToString("yyyy-MM-dd");

try

{

string connectstring = frmLogOn.f0.sqlConectstring;

string sqlcommand = "select [id] as 数据号,[username] as 用户名,[date] as 收支日期,[money] as 金额,[item] as 收支项目,[type] as 收支类型,[memo] as 备注 from [Account] where [username]='" + cmbUsername.Text + "' and [date]='" + dtStr + "' order by [id]";

SqlConnection aConnection = new SqlConnection(connectstring);

aConnection.Open();

SqlDataAdapter adp = new SqlDataAdapter(sqlcommand, connectstring);

DataSet ds = new DataSet();//定义数据集

adp.Fill(ds, "Account");//将Account表中的数据填充至ds中

dgvEachDay.DataSource = ds.Tables["Account"];

SqlCommandBuilder cbAccount = new SqlCommandBuilder(adp);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "系统错误");

}

}

每月收支情况主要实现个人以及家庭每月的收支情况统计，也是通过对数据库Account表的条件查询填充程序中的数据表，利用SQL中的sum函数统计共收入以及共支出的金额数量，并计算差值得出当月的盈余情况。其窗体界面如图4.9所示。



**图4.9 每月收支情况的窗体界面**

年度收支情况主要实现个人以及家庭每年的收支情况统计，也是通过对数据库Account表的条件查询填充程序中的数据表，利用SQL中的sum函数统计共收入以及共支出的金额数量，并计算差值得出当年的盈余情况。其窗体界面如图4.10所示。



**图4.10 年度收支情况的窗体界面**

收支类别统计情况主要实现个人以及家庭按照收支类别对收支记录进行统计。其窗体界面如图4.11所示。



**图4.11 收支类别统计情况的窗体界面**

**4.5．账户管理功能的实现**

账户管理主要由添加账户、删除账户以及修改账户信息构成。

添加账户功能主要实现用户向数据库User表中添加一个用户。在添加之前程序会先判断文本框是否为空，然后自动判断所添加的用户是否已经存在，若存在则弹出对话框说明已添加的用户已存在。该功能仅能在管理员权限下使用，普通用户此功能菜单在主界面中不可见。其窗体界面如图4.12所示。



**图4.12 添加用户的窗体界面**

添加用户的主要实现代码如下：

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

{

string username = (string)this.ParentForm.Tag;//从父窗体获取当前的用户名

if(txtPassword.Text!=txtPasswordRepeat.Text)

{

MessageBox.Show("两次密码不一致", "输入错误");

}

if (txtUsername.Text == "" || txtPassword.Text == "" || txtNickname.Text == ""|| txtPasswordRepeat.Text == "")

{

MessageBox.Show("请将所有必填空格填写完整", "填写错误");

return;

}

string str1, str2;

try

{

string connectstring = frmLogOn.f0.sqlConectstring;

string sqlcommand = "select username,nickname from [User]";

SqlConnection aConnection = new SqlConnection(connectstring);

aConnection.Open();

SqlCommand aCommand = new SqlCommand(sqlcommand,aConnection);

SqlDataReader reader = aCommand.ExecuteReader();

while (reader.Read())

{

str1 = reader.GetString(0);

str2 = reader.GetString(1);

if (txtUsername.Text.Trim() == str1.Trim() || txtNickname.Text.Trim() == str2.Trim())//判断用户名或昵称是否与已有帐号重复

{

reader.Close();

aConnection.Close();

MessageBox.Show("用户名或昵称与已有帐号重复", "系统错误");

return;

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "系统错误");

}

try

{

string connectstring = frmLogOn.f0.sqlConectstring;//从登录窗体获取数据库的连接字符串

string sqlcommand = "insert into [User]([username],[password],[nickname],[familycall],[accountstyle]) values('" + txtUsername.Text + "','" + txtPassword.Text + "','" + txtNickname.Text + "','" + txtFamilyCall.Text + "','" + cmbAccountStyle.Text + "')";

SqlConnection aConnection = new SqlConnection(connectstring);

aConnection.Open();

SqlCommand aCommand = new SqlCommand(sqlcommand, aConnection);

aCommand.ExecuteNonQuery();

aConnection.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "添加失败");

return;

}

MessageBox.Show("添加成功，您所添加的账户用户名为："+txtUsername.Text+"，密码为："+txtPassword.Text+"（用户名不可更改）", "恭喜");

txtUsername.Text = "";

txtNickname.Text = "";

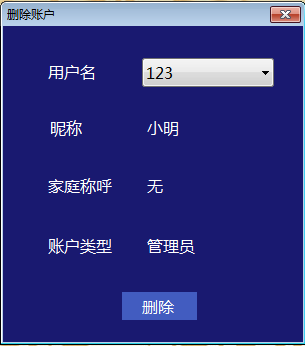
txtPassword.Text = "";

txtPasswordRepeat.Text = "";

txtFamilyCall.Text = "";

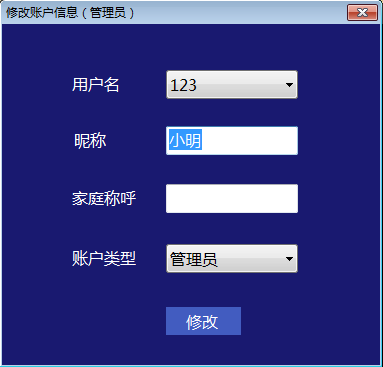
}

删除账户功能主要实现用户从数据库User表中删除一个已有的用户，若该用户是当前已登录的账户，程序将弹出对话框提示当前已登录的用户不能删除。该功能仅能在管理员权限下使用，普通用户此功能菜单在主界面中不可见。其窗体界面如图4.13所示。



**图4.13 删除用户的窗体界面**

修改账户信息可对昵称、家庭称呼、账户权限等账户信息进行修改。管理员用户权限下可对其他用户进行修改，而普通用户权限仅能修改当前普通用户账户信息，且账户类型不能被修改。当管理员账户把当前用户的权限设为普通用户时，软件会弹窗提示是否继续，若继续，当前用户的权限将降为普通用户。其窗体界面如图4.14所示。



**图4.14 修改账户信息的窗体界面**

**4.6．程序设置功能的实现**

程序设置可设置主界面的工具栏以及状态栏是否显示以及主界面的背景图片，应用按钮可立即应用当前设置但并不保存，保存按钮可将当前设置存入Access数据库文件中。程序设置的主要实现代码如下：

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

{

try

{

string connectstring = "provider=Microsoft.jet.OLEDB.4.0;Data Source=accountsystem.mdb;Jet OleDb:Database Password=lxw618";

string sqlcommand = "update [Set] set [toolbar]='" + cmbTool.Text + "',[statusbar]='" + cmbStatus.Text + "',[background]='" + cmbBackground.Text + "'";//更新Access数据库设置表的相应字段

OleDbConnection aConnection = new OleDbConnection(connectstring);

aConnection.Open();

OleDbCommand aCommand = new OleDbCommand(sqlcommand, aConnection);

aCommand.ExecuteNonQuery();

MessageBox.Show("保存成功", "恭喜");

aConnection.Close();

frmMain.f1.LoadSet();//调用主窗体定义的载入设置方法

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "系统错误");

}

}

1. **家庭记账软件系统测试**

**5.1．登录界面的测试**

该软件测试主要采用逐一运行的方式进行测试。运行家庭记账软件首先进入的是登录界面，其软件界面如下图5.1所示。



**图5.1 登录界面**

当输入完用户名密码后，密码框将以“●”形式显示密码，显示效果如图5.2所示，点击“清空”按钮后可将输入的用户名密码清空。



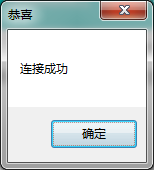
**图5.2 输入后的显示效果**

当点击连接设置时，将会打开连接设置窗体，连接设置窗体如下图5.3所示。



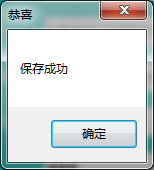
**图5.3 设置数据库连接窗体**

单击测试连接按钮，将测试文本框中所填写的数据库连接是否可用，若可用将弹出如下图5.4所示的对话框。



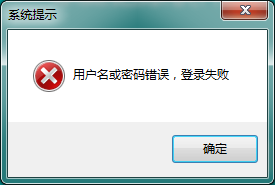
**图5.4 连接成功提示对话框**

单击保存按钮便可将该SQL数据库连接设置保存在Access数据库文件中。若保存成功将弹出如图5.5所示的对话框，并自动关闭当前界面。



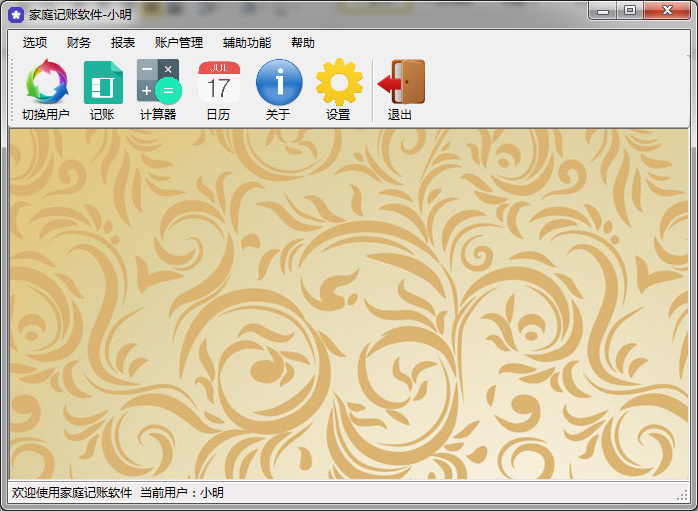
**图5.5 保存成功提示对话框**

当单击登录窗体中的登录按钮后，若用户名或密码错误，则会弹出如图5.6所示的用户名或密码错误的对话框。



**图5.6 用户名密码输入错误提示**

若用户名或密码输入正确，将直接进入程序主界面，程序主界面如下图5.7所示。若单击登录界面的白色叉按钮，整个软件将被关闭。

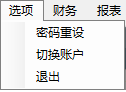


**图5.7 主界面窗体**

**5.2．主要功能模块的测试**

一、选项菜单模块

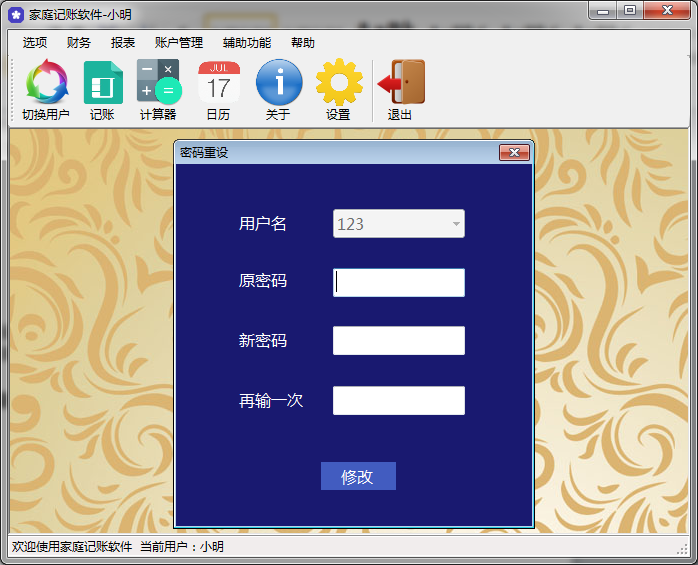
选项功能菜单主要由密码重设、切换用户和退出组成，其菜单列表如下图5.8所示。



**图5.8 选项菜单**

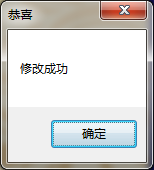
**1.密码重设模块**

单击密码重设菜单，密码重设窗体将在主窗体中被打开，其界面如下图5.9所示。



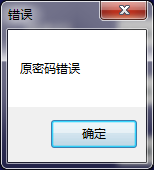
**图5.9 密码重设窗体**

输入原密码以及新密码并再输一次新密码，单击修改按钮则会弹出修改成功的对话框。对话框如图5.10所示。



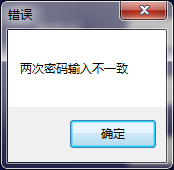
**图5.10修改成功提示对话框**

若原密码输入不正确，则会弹出原密码错误的对话框。对话框如图5.11所示。



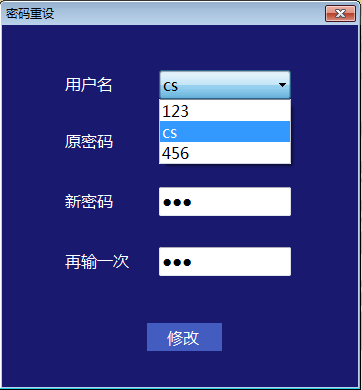
**图5.11 原密码错误提示对话框**

若密码与再输一次的密码不一致，则会弹出两次密码输入不一致的提示对话框，对话框如图5.12所示。



**图5.12 两次密码输入不一致提示对话框**

在管理员权限下，还可修改其他用户的密码，但仍需要输入原密码。如下图5.13所示，用户名的下拉列表可选择其他用户。



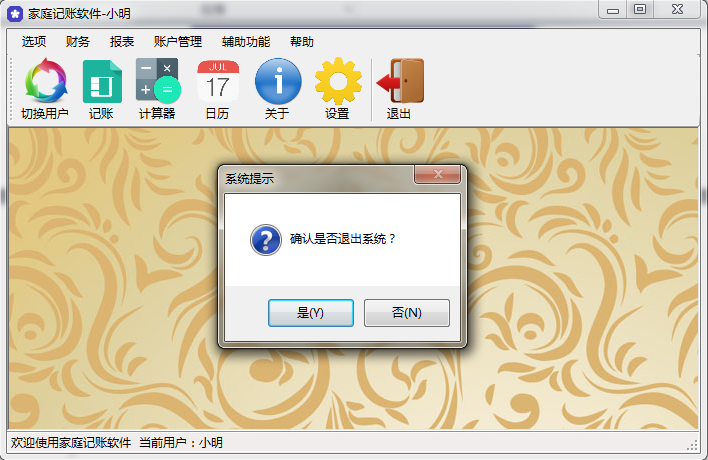
**图5.13 下拉列表选择其他用户**

**2.切换账户模块**

当单击切换账户后，程序将关闭主界面和辅助功能界面，并显示登录界面，将密码框清空。

**3.退出模块**

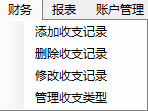
单击退出菜单项后，主窗体将弹出确认是否退出系统的对话框。若确认则关闭整个程序，单击否则恢复原状。弹出效果如下图5.14所示。



**图5.14 确认是否退出对话框**

二、财务菜单模块

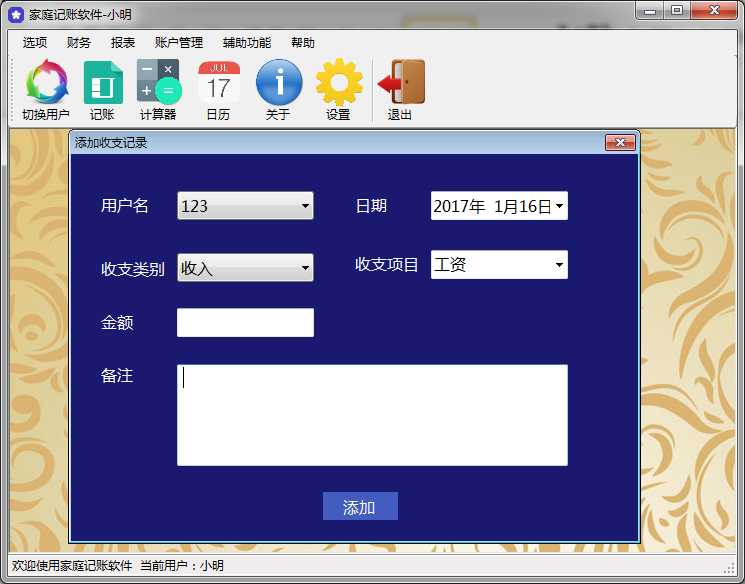
财务功能菜单主要由添加收支记录、删除收支记录、修改收支记录和管理收支类型组成，其菜单列表如下图5.15所示。



**图5.15 财务菜单**

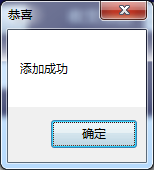
**1.添加收支记录模块**

单击添加收支记录菜单项，其界面如下图5.16所示。



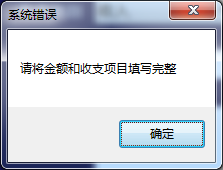
**图5.16 添加收支记录界面**

向文本框输入内容，并单击添加按钮，若添加成功，则显示添加成功对话框。添加成功对话框如下图5.17所示。



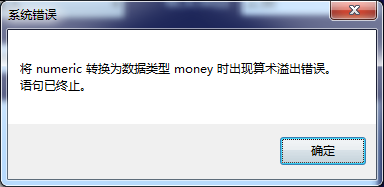
**图5.17 添加成功对话框**

若输入的金额或收支项目为空，则会弹出请将金额或收支项目填写完整的对话框。对话框如下图5.18所示。



**图5.18 提示将金额和收支项目填写完整对话框**

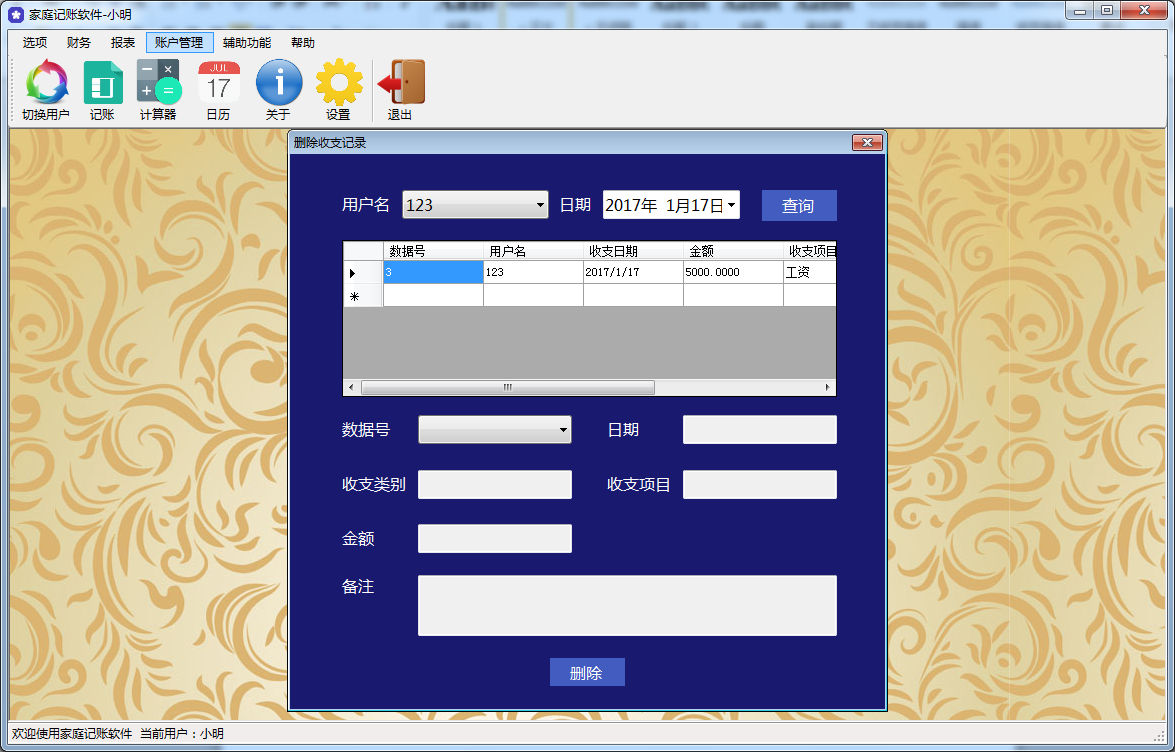
若输入的金额过大，系统则会捕获异常并提示溢出，对话框如下图5.19所示。



**图5.19 算术溢出对话框**

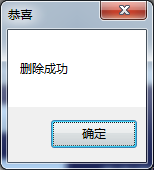
**2.删除收支记录模块**

单击删除收支记录菜单项，其界面如下图5.20所示。



**图5.20 删除收支记录界面**

在数据号下拉列表中选择要删除的收支记录，单击删除按钮，若删除成功，便会弹出删除成功对话框。弹出的对话框如下图5.21所示。



**图5.21 删除成功对话框**

**3.修改收支记录模块**

单击修改收支记录菜单项，其界面如下图5.22所示。



**图5.22 修改收支记录界面**

从中查询一个数据，如下图5.23所示。



**图5.23 查询后的数据**

选择要修改的数据号7，程序将自动填充如下图5.24所示的文本框及下拉列表框。

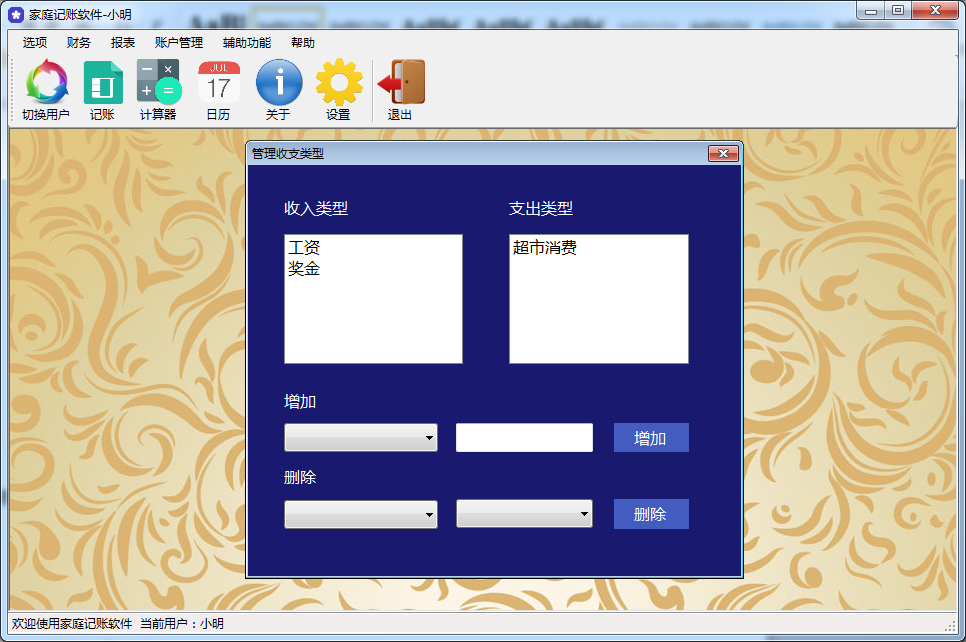


**图5.24 自动填充的效果**

修改需要修改的数据后单击修改按钮，若修改成功便会弹出修改成功的对话框。

**4.管理收支类型模块**

单击管理收支类型菜单项，其界面如下图5.25所示。

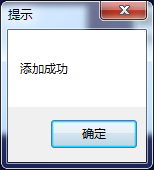


**图5.25 管理收支类型界面**

添加一条支出类型的项目，如图5.26所示，然后单击添加按钮。若添加成功，则会弹出添加成功的对话框，对话框如图5.27所示，支出类型的列表中也会增加一条记录，如图5.28所示。



**图5.26 添加一条支出类型的项目**

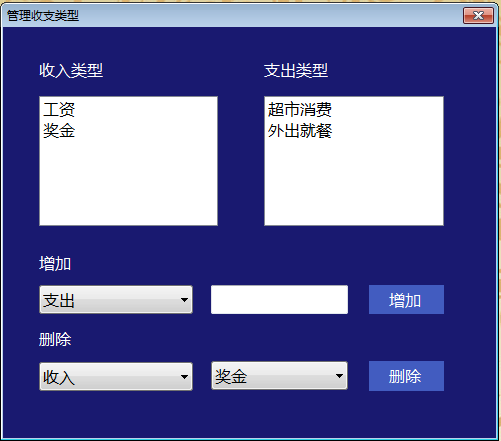


**图 5.27 添加成功对话框**



**图 5.28 添加成功后的效果**

选择一条需要删除的收入类型，例如奖金，如图5.29所示。



**图5.29 选择的收入类型**

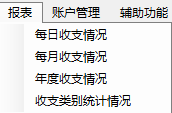
单击删除按钮，若删除成功便会弹出，删除成功的对话框，收入类型列表中奖金记录便会消失，如图5.30所示。



**图5.30删除后的效果**

三、报表菜单模块

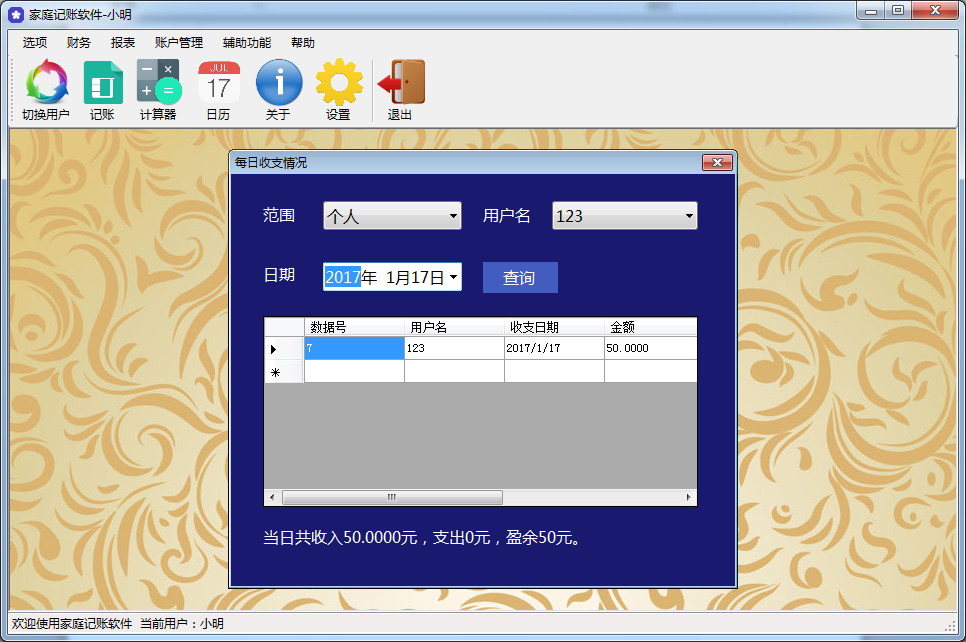
报表功能菜单主要由每日收支情况、每月收支情况、年度收支情况和、收支类别统计情况组成，其菜单列表如下图5.31所示。



**图5.31 报表菜单**

**1.每日收支情况模块**

单击每日收支情况菜单项，其界面如下图5.32所示。



**图5.32 每日收支情况界面**

选择所要查询的范围，单击查询按钮，即可查询到所需要的数据，如下图5.33所示。



**图5.33 查询后的结果**

**2.每月收支情况模块**

单击每月收支情况菜单项，其界面如下图5.34所示。



**图5.34 每月收支情况的界面**

选择所要查询的范围，单击查询按钮，即可查询到所需要的数据，如下图5.35所示。



**图5.35 查询后的结果**

**3.年度收支情况模块**

单击年度收支情况菜单项，其界面如下图5.36所示。



**图5.36 年度收支情况的界面**

选择所要查询的范围，单击查询按钮，即可查询到所需要的数据，如下图5.37所示。

****

**图5.37 查询后的结果**

**4.收支类别统计情况模块**

单击收支类别统计情况菜单项，其界面如下图5.38所示。



**图5.38 收支类别统计情况的界面**

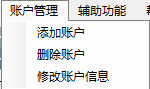
选择所要查询的范围，单击查询按钮，即可查询到所需要的数据，如下图5.39所示。



**图5.39 查询后的结果**

四、账户管理模块

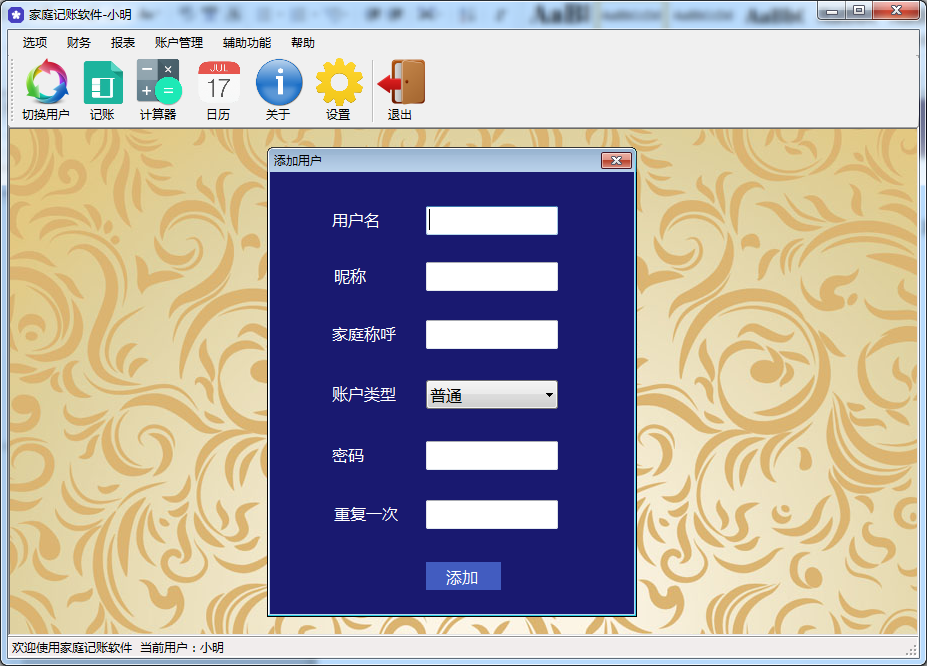
账户管理菜单主要由添加账户、删除账户、修改账户信息组成，其菜单列表如下图5.40所示。



**图5.40 账户管理菜单**

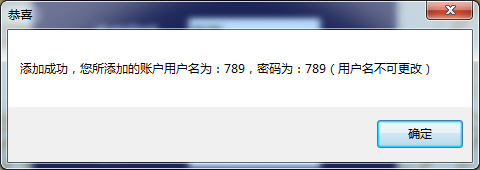
**1.添加账户模块**

单击添加账户菜单项，其界面如下图5.41所示。



**图5.41 添加用户的界面**

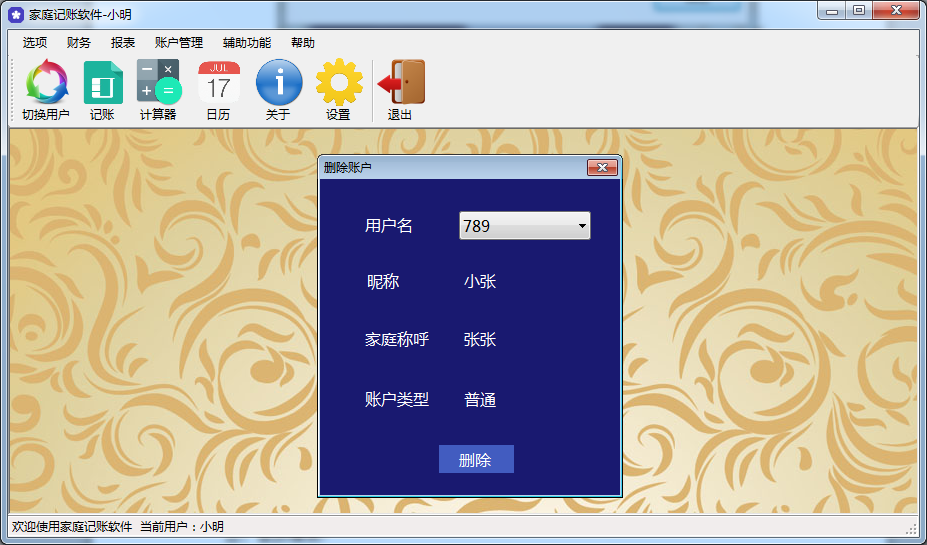
在界面填写所要添加的用户，单击添加按钮，如添加成功将弹出添加成功对话框，并清空所填写的内容，对话框如下图5.42所示。



**图5.42 添加成功**

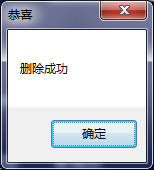
**2.删除账户模块**

单击删除账户菜单项，其界面如下图5.43所示。



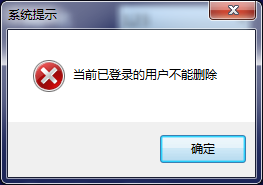
**图5.43 删除账户的界面**

在界面填写所要删除的用户，单击删除按钮，如删除成功将弹出删除成功的对话框，并自动重新载入用户名及内容，对话框如下图5.44所示。



**图5.44 删除成功的对话框**

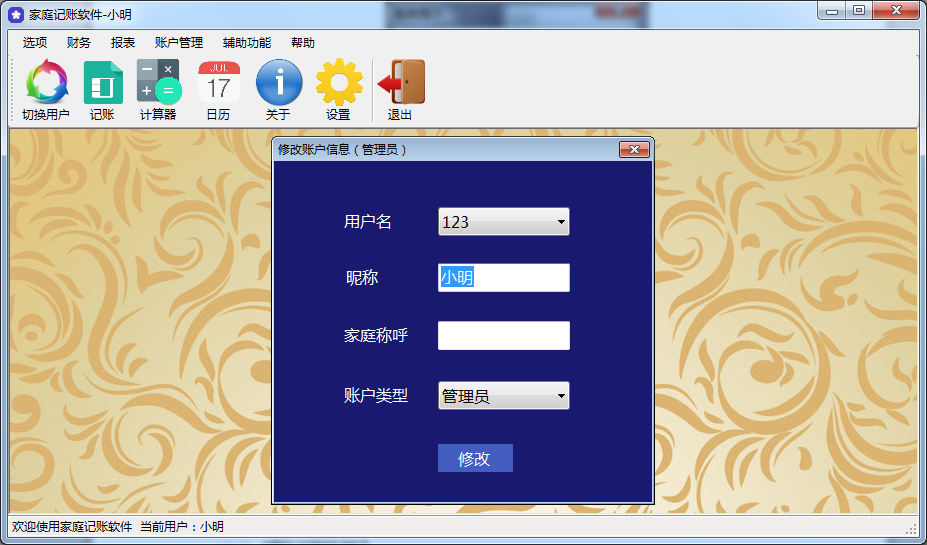
若删除的是当前账户，系统则会自动提示当前已登录的账户不能删除，提示对话框如下图5.45所示。



**图5.45 当前已登录的用户不能删除对话框**

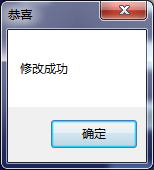
**3.修改账户信息模块**

单击修改账户信息菜单项，其界面如下图5.46所示。



**图5.46 修改账户信息界面**

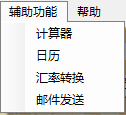
在界面修改所要修改的用户信息，单击修改按钮，如修改成功将弹出修改成功的对话框，对话框如下图5.47所示。



**图5.47 修改成功的对话框**

**5.3．辅助功能的测试**

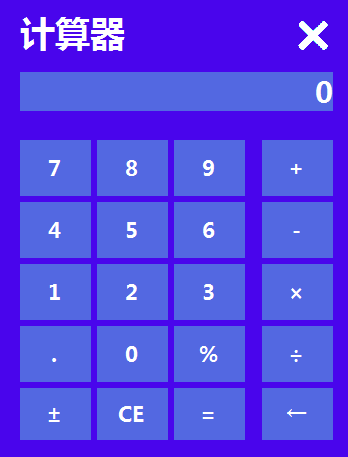
辅助功能菜单主要由计算器、日历、汇率转换和邮件发送组成，其菜单列表如下图5.48所示。



**图5.48 辅助功能菜单**

一、计算器模块

单击计算器菜单项，其界面如下图5.49所示。



**图5.49 计算器界面**

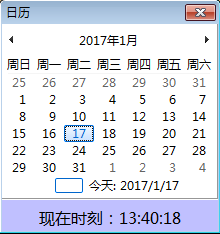
当除数为零时，则显示除数不能为零，如图5.50所示。



**图5.50 除数不能为零**

二、日历模块

单击日历菜单项，其界面如下图5.51所示。



**图5.51 日历界面**

三、汇率转换模块

单击汇率转换菜单项，其界面如下图5.52所示。



**图5.52 汇率换算**

任意输入一个想要输入的数值，单击转换按钮，兑换后的金额便在程序界面显示。如下图5.53所示。



**图5.53 换算后结果**

单击汇率设定按钮，汇率设定的窗体便被打开。如下图5.54所示。



**图5.54 汇率设定界面**

单击保存设定按钮，该汇率设定即被保存。

四、邮件发送模块

单击邮件发送菜单项，其界面如下图5.55所示。



**图5.55 邮件发送界面**

输入收件人及内容后，单击发送按钮即可发送，若收件人、内容为空，则会弹出提示对话框收件人、文本内容不能为空，对话框如图5.56所示。



**图5.56 不能为空提示**

单击设置按钮按钮，即可打开邮件发送设置窗体。其界面如下图5.57所示。



**图5.57 邮件发送界面**

单击测试按钮，即可向程序指定邮箱发送测试邮件，验证邮件发送设置是否填写正确，单击保存即可保存设置并关闭邮件发送设置。

1. **总结**

**6.1．设计过程中遇到的问题**

在设计的过程中，我也遇到了一系列问题，例如登录时需要连带关闭数据库连接设置窗体，在窗体已经关闭的情况下，继续关闭窗体，会导致程序报错，通过判断窗体的指针变量是否为空，可有效解决这个问题。在程序访问数据库时，DataReader未关闭导致再同一数据库连接下，无法再次使用DataReader读取数据。在数据库中用到有关于日期的查询语句时，SQL Server数据库查询语句的字段内容用到的是字符串类型的两个单引号，而Access数据库则使用的两个“#”号来标识日期字段的内容，当我在编写此类代码时，“#”号无法被SQL Server数据库识别，因而产生了错误。在编写程序的过程中，当我需要访问另一个窗体的特定控件的属性时，通常会遇到该控件为私有，无法访问，经过一系列学习研究发现，当将另一个窗体的窗体设计代码中的控件定义代码前的private改为public，或在另一个窗体的代码中添加一个访问器，再去调用这个访问器，即可解决相应的问题。

**6.2．心得体会**

使用.NET 实现PC端的应用程序开发，需要首先对这个软件项目进行合理的需求分析，确定程序的开发语言以及.NET的版本，以及采用哪种数据库。然后，划分整个软件的功能模块，根据每个模块的功能特点，设计相应的程序界面，并编写它的代码。

在代码编写过程中，良好的代码编写习惯同样也是十分重要的，好的代码风格不仅能使自己在修改代码过程中迅速理解自己代码的意思，也能使代码外观显得简洁统一，使别人快速理解代码的意思，给他人提供一定的阅读方便。

在运行调试过程中，对已出现的问题，要及时修正，如果暂时不能解决，可将其记录下来，并将错误的代码段使用注释语句屏蔽，在其旁边用注释语句注解；对于尚未发现的错误，则需要积极防范，主动考虑一些可能产生错误的情况，以提高程序的稳定性、可靠性。

通过程序的编写，这款家庭记账软件主要实现了对于家庭用户简单记账的功能，能使家庭记账变得更加方便、快捷，同时操作也比较简单，易于上手。该系统不足之处主要在于，对数据处理未使用可视化的图表、饼图之类的表现方式，代码的编写有些地方仍然存在冗余的现象，需要以后做进一步的改进。

**附 录**

登录窗体从mdb文件获取SQL数据库连接字符串的代码（accountsystem.mdb位于程序所在位置的同一目录下）：

public Boolean GetSqlConnectString()

{

try

{

string connectstring = "provider=Microsoft.jet.OLEDB.4.0;Data Source=accountsystem.mdb;Jet OleDb:Database Password=lxw618";

string sqlcommand = "select connect from [Set]";

OleDbConnection aConnection = new OleDbConnection(connectstring);

aConnection.Open();

OleDbCommand aCommand = new OleDbCommand(sqlcommand, aConnection);

OleDbDataReader reader = aCommand.ExecuteReader();

reader.Read();

sqlConectstring = reader.GetString(0);

reader.Close();

aConnection.Close();

return true;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "系统错误");

return false;

}

}

**参考文献**

1. 孙践知,张迎新,肖媛媛著.C#程序设计[M]. 北京:清华大学出版社，2010.11.
2. 胡锦丽主编，雷筱珍，刘芳副主编.数据库原理及应用案例教程（SQL Server版）[M].北京:北京大学出版社，2008.8.
3. 彭顺生等.C# Windows项目案例开发教程[M].北京:清华大学出版社，2014.
4. 刘克成，张凌晓.C#程序设计[M].中国铁道出版社，2007.
5. 崔建江.C#编程和.NET框架[M].机械工业出版社，2012.
6. 王爱莲，郭淑馨.Access数据库基础案例教程[M]. 北京:清华大学出版社，2016