**基于.NET应用程序开发实战报告**

**课程名称:** 基于.NET应用程序开发实战

**专 业:** 计算机科学与技术嵌入式培养

**班 级:**

**学 号:**

**学生姓名:**

**指导教师:**

**开课学期:**

2021年 4 月 6 日

目录

[1 引言（或绪论） 1](#_Toc9026)

[1．1 课程设计目的 1](#_Toc4725)

[1．2 课程设计的意义 1](#_Toc17193)

[2 可行性分析 2](#_Toc8090)

[2．1 技术可行性 2](#_Toc16975)

[2．2 经济可行性 2](#_Toc26374)

[2．3 社会可行性 2](#_Toc29582)

[3 系统分析设计 2](#_Toc14944)

[3．1 系统功能描述 2](#_Toc19290)

[3．2 系统功能表 2](#_Toc10254)

[3．3 E-R图分析 3](#_Toc25403)

[（1） 用户信息E-R图 3](#_Toc32073)

[（2） 收入信息E-R图 3](#_Toc19983)

[（3） 支出信息E-R图 4](#_Toc14808)

[3．4 数据库设计 4](#_Toc27488)

[（1）用户信息数据表（user） 4](#_Toc2707)

[（2） 收入信息数据表（Income） 4](#_Toc6730)

[（3） 支出信息数据表（Spend） 5](#_Toc18723)

[（4） 收支差额表（pay） 5](#_Toc28878)

[4 系统功能展示 6](#_Toc12952)

[4．1 登录注册功能 6](#_Toc647)

[4．2 主页设计 7](#_Toc21448)

[4．3 修改密码 10](#_Toc23053)

[4．4 收支录入功能 11](#_Toc24983)

[4．5 收入支出查询并导出Excel表 12](#_Toc8858)

[4．6 收入支出管理系统 15](#_Toc9372)

[结 论 19](#_Toc11176)

**1 引言（或绪论）**

财务管理系统是每一个管理家政财务收入和支出会计不可缺少的管理信息系统它的内容对于管理者来说是至关重要的,所以财务管理系统应该能够为每一个的管理者提供充足的信息和快捷的查询手段，大大的方便家政的账务管理。随着科学技术的不断提高,计算机科学日渐成熟, 网上银行的迅速发展，其强大的功能已为人们深刻认识它已进入人类社会的各个领域并发挥着越来越重要的作用。财务管理系统在每个公司都很重要，并通过使用财务管理系统，公司的者可以随时掌握公司的收支状况，同时可以通过财务管理系统对公司的经营状况分析，对公司的经营投资的决策提供财务方面的相关资料。不仅仅是公司，现在个人也需要财务管理系统，可以记录你的每一笔开销和收入，帮助人们更好的理财。系统主要功能有：系统管理，收支录入，记账管理。

**1．1 课程设计目的**

1. 培养综合运用所学知识独立完成课题的能力。
2. 更深入地理解和掌握该课程中的有关基本概念，程序设计思想和方法。
3. 提高对工作认真负责、一丝不苟，对同学团结友爱，协作攻关的基本素质。
4. 培养勇于探索、严谨推理、实事求是、有错必改，用实践来检验理论，全方位考虑问题等科学技术人员应具有的素质。
5. 培养从资料文献、科学实验中获得知识的能力，提高从别人经验中找到解决问题的新途径的悟性，初步培养工程意识和创新能力。
6. 对掌握知识的深度、运用理论去处理问题的能力、实验能力、课程设计能力、书面表达能力进行考核。

**1．2 课程设计的意义**

计算机已经成为我们学习和工作的得力助手，使用其可方便的管理财务科目，为其财务管理增加了方便。今天，计算机的价格已经十分低廉，同时性能有了长足的进步。它已经被应用于许多领域。

现在我国的财务管理水平还有纸质介质的存在，这样的机制已经不能适应时代的发展，因为它浪费了许多人力和物力，在信息时代这种传统的管理方法必然被计算机为基础的信息管理所取代。

我们团队作为软件工程专业的学生，希望可以在这方面有所贡献。改革的总设计师邓小平同志说过“科学技术是第一生产力”，我们希望能用我们所学的知识编制出一个实用的程序来帮财务人员进行财务管理。

**2 可行性分析**

**2．1 技术可行性**

本系统采用Windows10版本作为操作平台。数据库管理系统选用SQL Server 2012，该数据库管理系统,提供数据库之间的数据复制功能,可代替现有系统单据数据的手工传递工作，降低出错率，提高数据的可用性。本系统的应用程序开发平台选用Microsoft Visual studio 2019，使用的语言是c#窗体设计。

**2．2 经济可行性**

本系统开发成本低，一台可运行的电脑即可。大大缩减了人工财务管理所需要的人工成本。是非常方便的一个管理系统经济可行性高。

**2．3 社会可行性**

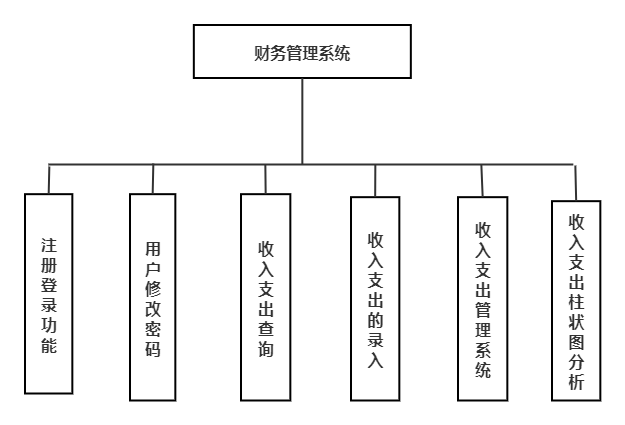
本系统操作简单方便，容易上手。适合人们使用。在财务管理上有很好的简便性，可以节约很多人力物力，适合流向社会，服务大众。

**3 系统分析设计**

**3．1 系统功能描述**

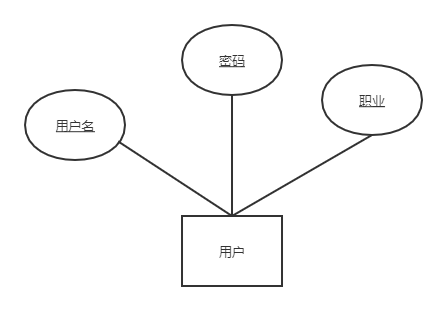
财务管理系统主要有用户登录注册功能，用户可以注册账号登录系统中，形成自己的收入支出账单，保存到数据库中。可以对数据进行增删改查等功能，了解自己的收支情况绘成出装图进行分析。

**3．2 系统功能表**

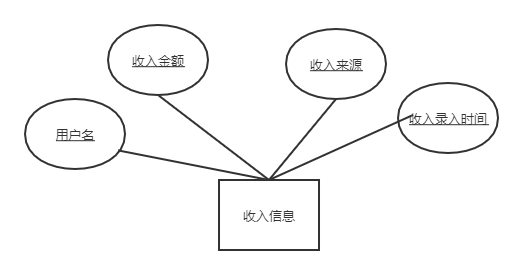
****

**3．3 E-R图分析**

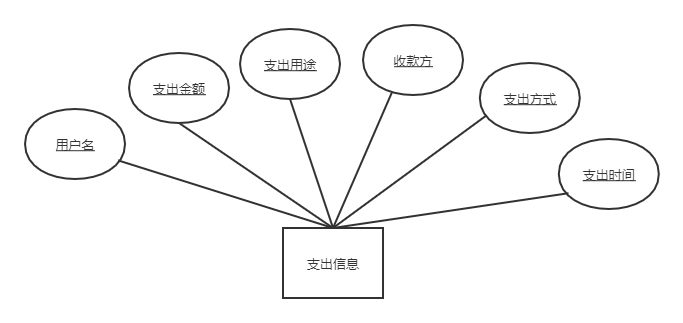
1. **用户信息E-R图**

****

1. **收入信息E-R图**

****

1. **支出信息E-R图**

****

**3．4 数据库设计**

（1）用户信息数据表（user）

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 字段描述 |
| UserName | varchar | 用户名 |
| Password | Varchar | 密码 |
| Juese | varchar | 职业 |

1. 收入信息数据表（Income）

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 字段描述 |
| Name | varchar | 用户名 |
| IncomeNum | Int | 收入金额 |
| IncomeSource | varchar | 收入来源 |
| IncomeDate | datetime | 日期 |

1. 支出信息数据表（Spend）

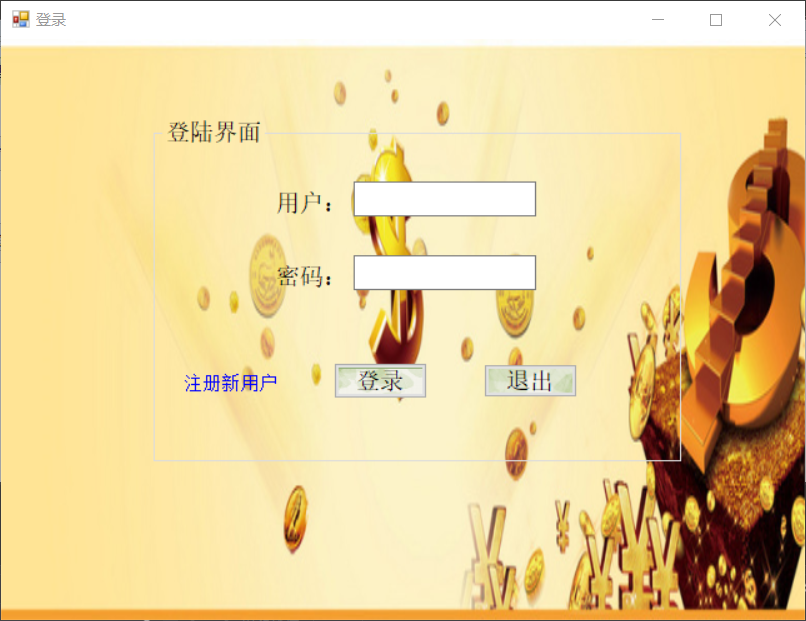
|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 字段描述 |
| spendName | varchar | 用户名 |
| spendNum | int | 支出金额 |
| spendDep | varchar | 支出用途 |
| receiveName | varchar | 收款方 |
| expenditure | varchar | 支付方式 |
| spendDate | datetime | 支出时间 |

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 字段描述 |
| month | int | 时间 |
| income | varchar | 收入 |
| spend | varchar | 支出 |
| payment | varchar | 收支差额 |

1. 收支差额表（pay）

**4 系统功能展示**

**4．1 登录注册功能**





用户可以在此界面同过相关信息的填写进行账号密码注册然后登录财务管理系统进行一些相关操作。

代码如下：

**登录功能：**

string sql = "select \* from users where UserName =@username";

SqlParameter p = new SqlParameter("@username",txtName.Text);

DataTable dt = new DataTable();

dt = DBHelper.ExecuteQuery(sql, p);

if (dt.Rows.Count>0)

{Register register = new Register();

register.UserName = dt.Rows[0][1].ToString();

register.Password = dt.Rows[0][2].ToString();

if(txtPwd.Text==register.Password)

{MessageBox.Show("登录成功！");

FormMain main = new FormMain();

main.Tag = register;

main.Show();

this.Hide();}

else{MessageBox.Show("密码错误！");

txtPwd.Text = "";}

**注册功能：**

SqlParameter[] ps ={new SqlParameter("@username",Newuser.Text),

new [SqlParameter("@password",Newpwd.Text),](mailto:SqlParameter(\"@password\",Newpwd.Text),)

new SqlParameter("@password2",Newpwd2.Text),

new SqlParameter("@juese",txtJuese.Text)};

if(Newuser.Text==""){

MessageBox.Show("请输入用户名");}

else{if(Newpwd.Text=="")

{ MessageBox.Show("请输入密码"); }

else{if(Newpwd.Text != Newpwd2.Text){

MessageBox.Show("请确认重复密码");}

else{if (Newuser.ReadOnly == false){//添加

string sql = "insert into users

values(@username,@password,@juese)";

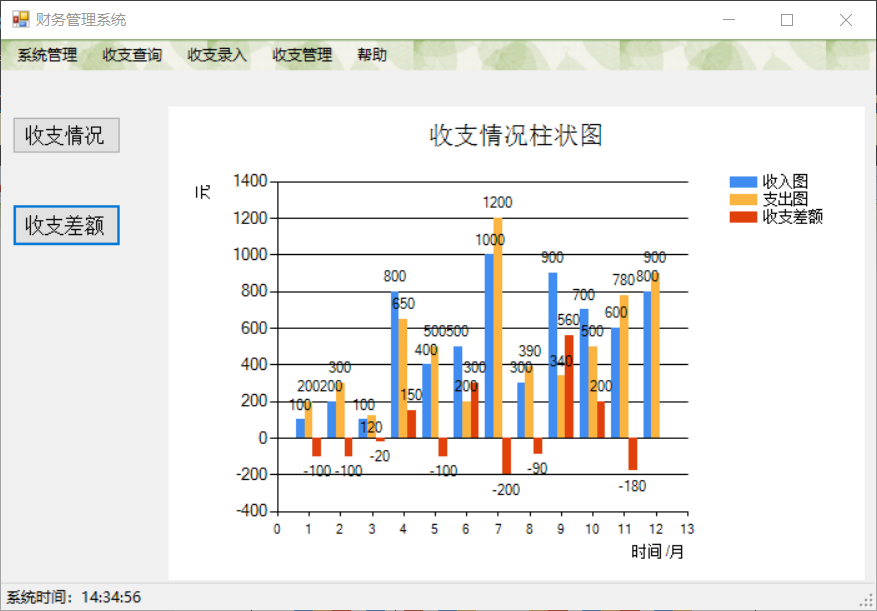
int res = DBHelper.ExecuteNonQuery(sql, ps);

if (res > 0){MessageBox.Show("注册成功");}

else{MessageBox.Show("注册失败");}}

else{}Newuser.ReadOnly = false;}}}}

**4．2 主页设计**



用户登录进系统主页面选择一些操作选择，有查询账单信息，和收支录入，收支系统管理等功能。还可以将自己的收支信息绘制成柱状图进行收支分析。也可以修改账号密码。

代码如下：

DataTable CreatData(){

DataTable dt = new DataTable();

//实例化SqlConnection对象

SqlConnection sqlCon = new SqlConnection();

//实例化SqlConnection对象连接数据库的字符串

sqlCon.ConnectionString = conn;

//定义SQL语句

//sql公用部分

string sqlQ = "select \* from pay";

dt = GetDataTable(sqlQ);

return dt;}

public static DataTable GetDataTable(string Sqlstr)

{DataTable dt;

SqlDataAdapter sda;

SqlConnection sqlCon = new SqlConnection();

//实例化SqlConnection对象连接数据库的字符串

sqlCon.ConnectionString = conn;

sqlCon.Open();

sda = new SqlDataAdapter(Sqlstr, conn);

dt = new DataTable();

sda.Fill(dt);

sqlCon.Close();

return dt;}

private void btn1\_Click(object sender, EventArgs e){

SqlConnection connection = new SqlConnection(conn);

DataTable dt = CreatData();

Chart1.DataSource = dt;//绑定数据

Chart1.Series["Series1"].ChartType = System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Column;//设置图表类型

Chart1.Series["Series2"].ChartType=System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Column;

Chart1.Series[0].XValueMember = "month";//X轴数据成员列

Chart1.Series[0].YValueMembers = "income";//Y轴数据成员列

Chart1.Series[1].XValueMember = "month";

Chart1.Series[1].YValueMembers = "spend";

Chart1.ChartAreas["ChartArea1"].AxisX.Title = "时间 /月";//X轴标Chart1.ChartAreas["ChartArea1"].AxisX.TitleAlignment=StringAlignment.Far;//设置Y轴标题的名称所在位置位远

Chart1.ChartAreas["ChartArea1"].AxisY.Title = "元";//Y轴标题

Chart1.ChartAreas["ChartArea1"].AxisY.TitleAlignment=StringAlignment.Far;//设置Y轴标题的名称所在位置位远

Chart1.ChartAreas["ChartArea1"].AxisX.Interval = 1;//X轴数据的间 Chart1.ChartAreas["ChartArea1"].AxisX.MajorGrid.Enabled = false;//不显示竖着的分割线

Chart1.Series[0].IsValueShownAsLabel = true;//显示坐标值

Chart1.Series[1].IsValueShownAsLabel = true;

Chart1.Series[0].LegendText = "收入图";

Chart1.Series[1].LegendText = "支出图";}

private void btn2\_Click(object sender, EventArgs e){

SqlConnection connection = new SqlConnection(conn);

// 折线图

DataTable dt = CreatData();

Chart1.DataSource = dt;//绑定数据

Chart1.Series["Series3"].ChartType = System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Column;//设置图表类型

Chart1.Series[2].XValueMember = "month";//X轴数据成员列

Chart1.Series[2].YValueMembers = "payment";//Y轴数据成员列

Chart1.ChartAreas["ChartArea1"].AxisX.Title = "时间 /月";//X轴标

Chart1.ChartAreas["ChartArea1"].AxisX.TitleAlignment = StringAlignment.Far;//设置Y轴标题的名称所在位置位远

Chart1.ChartAreas["ChartArea1"].AxisY.Title = "元";//Y轴标题

Chart1.ChartAreas["ChartArea1"].AxisY.TitleAlignment = StringAlignment.Far;//设置Y轴标题的名称所在位置位远

Chart1.ChartAreas["ChartArea1"].AxisX.Interval = 1;//X轴数据的间

Chart1.ChartAreas["ChartArea1"].AxisX.MajorGrid.Enabled = false;//不显示竖着的分割线

Chart1.Series[2].IsValueShownAsLabel = true;//显示坐标值

Chart1.Series[2].LegendText = "收支差额";}

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

toolStripStatusLabel2.Text = "系统时间：" +

DateTime.Now.ToLongTimeString();}

**4．3 修改密码**



用户进入用户修改密码界面，可以选择修改密码，必须记住原始密码，否则修改密码会失败。

代码如下：

string sql = "select \* from users where UserName =@username";

SqlParameter p = new SqlParameter("@username", txtUser.Text);

DataTable dt = new DataTable();

dt = DBHelper.ExecuteQuery(sql, p);

Register register = new Register();

register.UserName = dt.Rows[0][1].ToString();

register.Password = dt.Rows[0][2].ToString();

if(register.Password==txtPwd.Text)

{SqlParameter[] ps ={new SqlParameter("@user",txtUser.Text),

new SqlParameter("@pwd",txtPwd.Text),

new SqlParameter("@npwd",txtNewPwd.Text),};

string sql2 = "update users set Password=@npwd where UserName=@user";

int res = DBHelper.ExecuteNonQuery(sql2, ps);

if (res > 0){MessageBox.Show("修改成功,请重新登录！");

FormLogin login = new FormLogin();

login.Show();

this.Hide();}

else{MessageBox.Show("修改失败！");}}

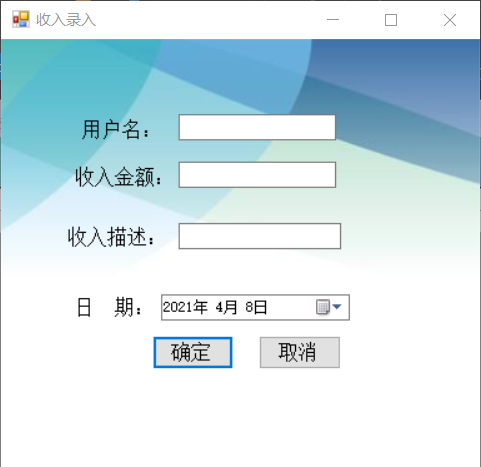
else{MessageBox.Show("原密码填写错误");}

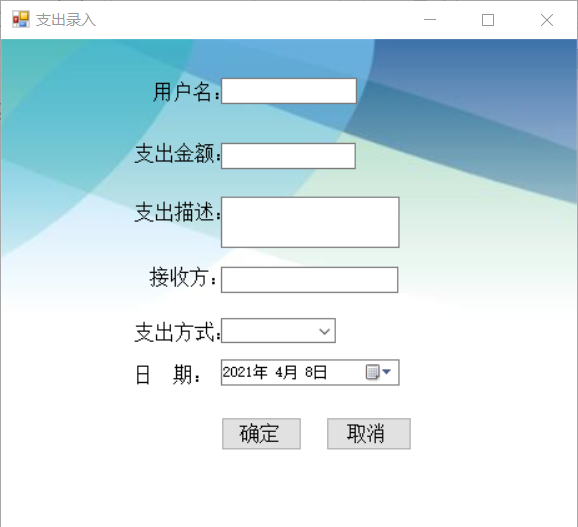
txtUser.Text = "";

txtPwd.Text = "";

txtNewPwd.Text = "";}

**4．4 收支录入功能**





这是收支录入界面，用户可以根据自己的每月收支情况进行财务登记，用户信息会保存到数据库中，安全性很高。

代码如下：

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

SqlParameter[] ps ={new SqlParameter("@name",txtName.Text),

new SqlParameter("@num",Convert.ToInt32(txtNum.Text)),

new SqlParameter("@dep",txtDep.Text),

new SqlParameter("@receive",txtAdd.Text),

new SqlParameter("@add",txtAdd.Text),

new SqlParameter("@date",txtDate.Value.Date)};

if (txtName.ReadOnly == false){

string sql = "insert into Spend

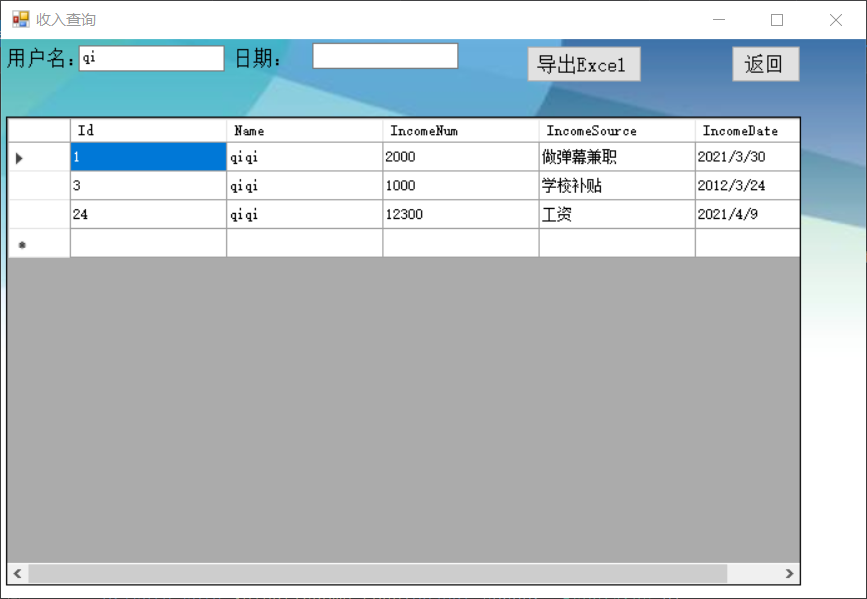
values(@name,@num,@dep,@receive,@add,@date)";

int res = DBHelper.ExecuteNonQuery(sql, ps);

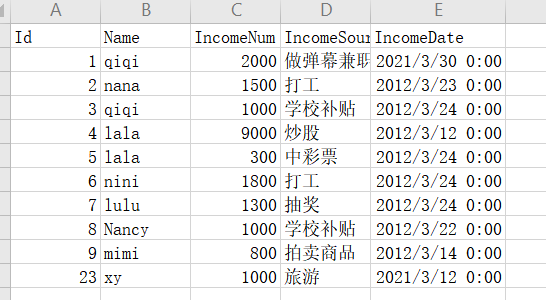
if (res > 0){MessageBox.Show("添加成功");}else{

MessageBox.Show("添加失败");}}}

**4．5 收入支出查询并导出Excel表**







在查询界面，用户可以进行账单的查询，找到自己每一笔钱的出处和去处。用户也可以将收支信息导出做成Excel表格。

查询代码：

public void LoadList(){

string sql = "select \* from Income";

DataTable dt = new DataTable();

dt = DBHelper.ExecuteQuery(sql);

dgvList.AutoGenerateColumns = false;

dgvList.DataSource = dt;}

public void LoadSearch(){

string sql = "select Id,Name,IncomeNum,IncomeSource,IncomeDate from

Income where InDelete=0";

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

SqlParameter p = new SqlParameter("@name", txtName.Text);

SqlParameter p2 = new SqlParameter("@date", txtDate.Text);

if(txtName.Text!=""&&txtDate.Text=="")

{

sql += "and Name like '%' + @name + '%'";

list.Clear();

list.Add(p);}

if (txtName.Text == "" && txtDate.Text!= ""){

sql += "and IncomeDate like '%' + @date + '%'";

list.Clear();

list.Add(p2);}

if(txtName.Text !="" && txtDate.Text!= "")

{

sql += "and Name like '%' + @name + '%'and IncomeDate like '%' + @date + '%'";

list.Clear();

list.Add(p);

list.Add(p2);}

DataTable dt = new DataTable();

dt = DBHelper.ExecuteQuery(sql, list.ToArray());

dgvList.AutoGenerateColumns = false;

dgvList.DataSource = dt;

}

导出Excel代码：

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

{

SaveFileDialog dlg = new SaveFileDialog(); //实例化一

个SaveFileDialog保存文件对话框

dlg.Filter = "Excel files (\*.xls)|\*.xls";

dlg.FilterIndex = 0;

dlg.RestoreDirectory = true;

dlg.CreatePrompt = true;

if (dlg.ShowDialog() == DialogResult.OK)

{

Stream myStream;

myStream = dlg.OpenFile();

StreamWriter sw = new StreamWriter(myStream,System.Text.Encoding.GetEncoding(-0));

string columnTitle = "";

try

{

//写入列标题

for (int i = 0; i < dgvList.ColumnCount; i++)

{

if (i > 0)

{

columnTitle += "\t";

}

columnTitle += dgvList.Columns[i].HeaderText;}

sw.WriteLine(columnTitle);

//写入列内容

for (int j = 0; j < dgvList.Rows.Count; j++)

{

string columnValue = "";

for (int k = 0; k < dgvList.Columns.Count; k++)

{

if (k > 0)

{

columnValue += "\t";

}

if (dgvList.Rows[j].Cells[k].Value == null)

columnValue += "";

else

columnValue += dgvList.Rows[j].Cells[k].Value.ToString().Trim();

}

sw.WriteLine(columnValue);}

sw.Close();

myStream.Close();

}

catch

{MessageBox.Show("导出不成功");}

finally

{

sw.Close();

myStream.Close();

}}}

**4．6 收入支出管理系统**





在收支管理系统界面，用户可以给自己的账单信息进行修改或者删除，本系统非常简单易操作，可以满足用户的财务需求。

代码如下：

public void LoadSearch(){

string sql = "select Id,Name,IncomeNum,IncomeSource,IncomeDate from Income where InDelete=0";

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

SqlParameter p = new SqlParameter("@name", txtName.Text);

SqlParameter p2 = new SqlParameter("@date", txtDate.Text);

if(txtName.Text!=""&&txtDate.Text=="")

{sql += "and Name like '%' + @name + '%'";

list.Clear();

list.Add(p);}

if (txtName.Text == "" && txtDate.Text!= "")

{sql += "and IncomeDate like '%' + @date + '%'";

list.Clear();

list.Add(p2);}

if(txtName.Text !="" && txtDate.Text!= "")

{sql += "and Name like '%' + @name + '%'and IncomeDate like '%' + @date + '%'";

list.Clear();

list.Add(p);

list.Add(p2);}

DataTable dt = new DataTable();

dt = DBHelper.ExecuteQuery(sql, list.ToArray());

dgvList.AutoGenerateColumns = false;

dgvList.DataSource = dt;}

public void LoadList(){string sql = "select \* from Income";

DataTable dt = new DataTable();

dt = DBHelper.ExecuteQuery(sql);

dgvList.AutoGenerateColumns = false;

dgvList.DataSource = dt;}

private void FormIncome\_Load(object sender, EventArgs e){

LoadList();}

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

{SqlParameter[] ps ={new SqlParameter("@id",txtId.Text),

new SqlParameter("@name",txtName2.Text),

new SqlParameter("@num",Convert.ToInt32(txtNum.Text)),

new SqlParameter("@source",txtSource.Text),

new SqlParameter("@date",txtDate.Value.Date)};

string sql = "update Income set

Name=@name,IncomeNum=@num,IncomeSource=@source,IncomeDate=@date

where Id=@id";

int res = DBHelper.ExecuteNonQuery(sql, ps);

if (res > 0){MessageBox.Show("修改成功");

LoadList();}

else{MessageBox.Show("修改失败");}}

private void button4\_Click(object sender, EventArgs e){

FormMain main = new FormMain();

main.Show();

this.Hide();}

private void btndel\_Click(object sender, EventArgs e){

DialogResult dialogRes = MessageBox.Show("确认删除吗？", "提示", MessageBoxButtons.OK);

if (dialogRes == DialogResult.OK){

int id = Convert.ToInt32(dgvList.SelectedRows[0].Cells[0].Value);

string sql = "delete from Income where Id=@id";

SqlParameter p = new SqlParameter("@id", id);

int res = DBHelper.ExecuteNonQuery(sql, p);

if (res > 0){LoadList();}

else{MessageBox.Show("删除失败");}}

else{return;}}

private void dgvList\_CellDoubleClick(object sender, DataGridViewCellEventArgs e){

//获取当前行的信息并显示在相应的位置

int id = Convert.ToInt32(dgvList.SelectedRows[0].Cells[0].Value);

string name = dgvList.SelectedRows[0].Cells[1].Value.ToString();

int num = Convert.ToInt32(dgvList.SelectedRows[0].Cells[2].Value);

string source = dgvList.SelectedRows[0].Cells[3].Value.ToString();

DateTime date= Convert.ToDateTime(dgvList.SelectedRows[0].Cells[4].Value.ToString());//显示

txtId.Text = id.ToString();

txtName2.Text = name;

txtNum.Text = num.ToString();

txtSource.Text = source;

txtDate.Value = date;}

结 论

通过本次的课程设计，加深了我对c#这门语言和数据库操作的了解，学到了窗体设计和一些控件，提高了综合运用课程所学知识的能力，在整个过程中，我遇到了许多的问题，包括对一些控件功能的不了解，调试时出现的种种错误，以及对两种语言的结合使用等等。最后通过网上查找资料和去学习相关视频，将所遇到的问题逐一解决。加深了我对数据库数据增删改查的理解。

从整个的窗体设计以及系统功能运用和系统测试等各个步骤，分步完成系统的各项任务，实现了系统中的用户密码修改，财务信息查询，财务信息录入，用户收支信息更新、以及将数据库通过柱状图来展示数据等模块的功能。通过实际编译系统的分析设计、编程调试，我也或多或少的掌握应用软件的分析方法和工程设计方法。这也是这次程序设计的收获。

这次的课程设计还有一些问题没有解决，做的不是很完美。整个窗体设计布局不是很协调也不够美观。总体能实现的功能还有待加强，我还要继续深入学习，掌握更多的知识，设计出功能更完善的系统管理。

总体上对这次的程序编写比较满意，希望以后可以学习到更多的相关知识，能提升自己的程序编写能力。

教师点评：

得 分:

教师签名:

批阅日期: