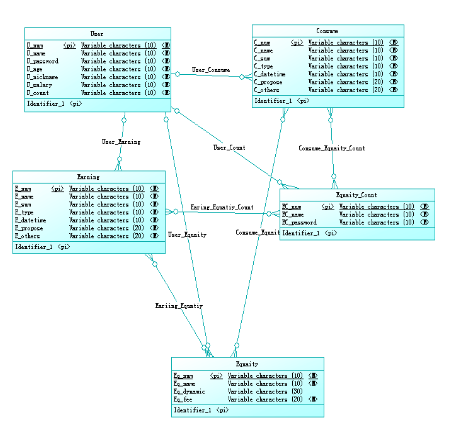
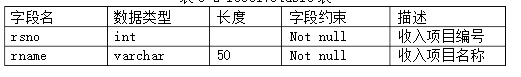
**家庭财务管理系统详细设计**

**1数据库设计**  
**1.1 数据库需求分析** 本项目需要从数据库中查询大量的信息，所有模块功能的实现都需要和数据库进行交互。用户权限的实现、以及页面上各个功能图标的显示全是查询数据库动态生成的，具有不同角色的用户登录到系统中所看到的主界面中功能的选项是有区别的，对某些用户而言，一些功能是完全屏蔽的。鉴于系统数据交互的复杂性，一个良好的数据库设计就成为了我们必须要关注的问题，在需求阶段本人仔细设计了数据库中的表以及各表的关系。我所负责的系统管理模块所用到的表的关系，如图所示：  
  
  
**1.2角色的分配和表空间的分配**  
数据库依照mysql数据库的使用规范，应先创建了一个用户，并为之分配权限，表空间使用了默认的USERS表空间。  
  
  
**1.3 数据库中各主要表的结构**  
 本系统是实现家庭财务管理系统，主要设计了11张表，分别为fmtable，receivetable，outlaytable，dailyincometable，dailyexpensetable，currentaccounttable, currentaccountimfortable, depositaccounttable，lendtable，borrowtable,record表。分别用于存储家庭成员信息、收入项目信息、支出项目信息、日常收入信息、日常支出信息、活期账户信息、存取款信息、定期账户信息，借出款信息表，借入款信息表,记录表。

1. 家庭成员表用于记录家庭成员的基本信息。



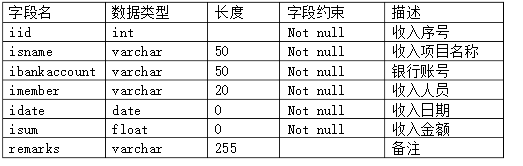
1. 收入项目表用于记录收入项目信息。



1. 支出项目表用于记录支出项目信息。



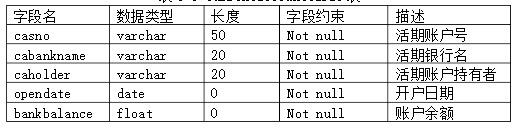
1. 日常收入信息表用于记录日常收入的详细信息 。



1. 日常支出信息表用于记录日常支出的详细信息。



1. 活期账户表用于记录活期账户的信息。



1. 存取款信息表用于记录每次的存取款记录信息。



1. 定期账户信息表用于记录定期账户的详细信息。



1. 借出款信息表用于记录借出款的详细信息。

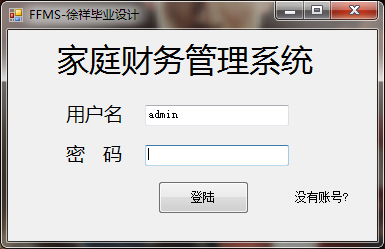


1. 借入款信息表用于记录借入款的详细信息。

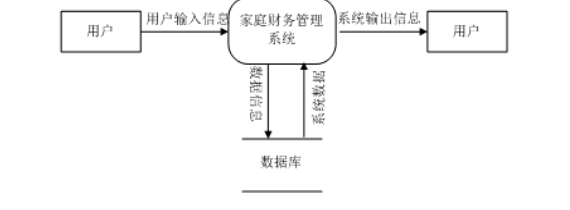
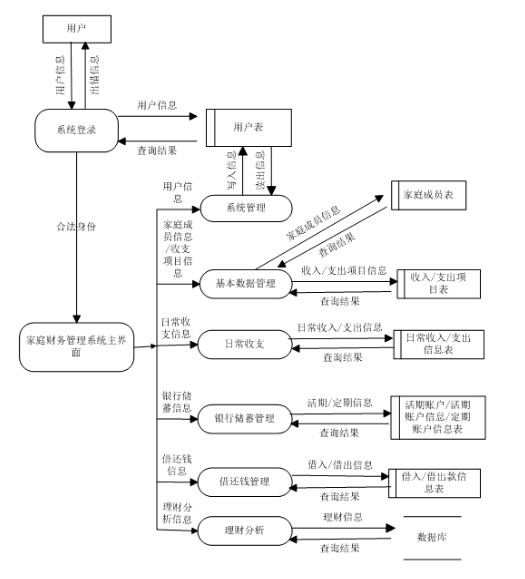
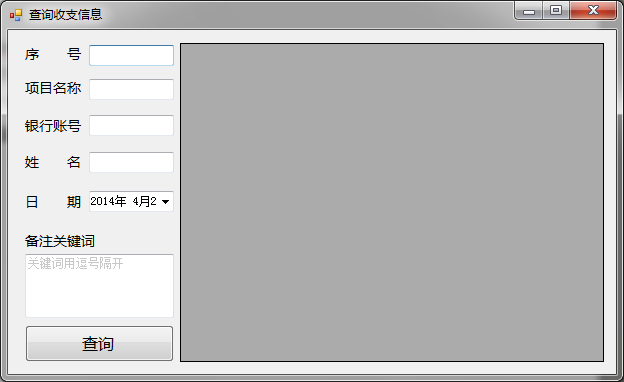


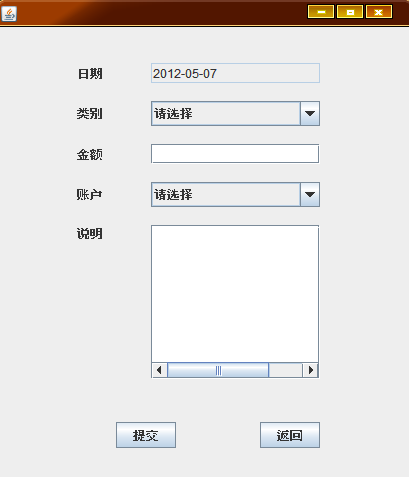
1. 记录表用于记录每次银行账号更新的信息，用于每月总的余额查询。

  
**2系统功能实现**  
  
**2.1 系统管理模块**  
 系统登录如图所示：

  
先进行用户登录，然后输入用户名或者密码进入家庭理财管理项目。  
**2.2 系统管理模块**  
 从登陆页面进入，用户输入自己的用户名和密码后，单击“提交”按钮或回车进入系统选择功能画面（登录之后可看到主页面上的最新提示，不同的角色会有不同的提示），如下图所示：如果用户名或密码错误，则不能进入系统。在系统的主页面中，左边是功能树，右边是工作区。如图所示：  
 

系统管理模块部分主要是系统管理员对权限和基础数据的数据字典进行管理，包括组织机构管理、角色管理、用户管理、信息上报、系统公告、文件管理、年度编程、数据备份。  
具体操作流程如下：  
系统管理员首先建立组织机构，然后再建立角色，同时给角色分配相应的权限。  
然后建立用户，把此用户分给需要的组织机构和角色。  
用户登录后，可以根据自己所具备的角色对相应的功能模块进行操作。而对于用户所属角色外的功能，对该用户则是不可见的，这样就限制了用户的权限，保证了系统的安全性。  
用户就是本系统的所有使用者，角色就是本系统中具有特定功能的一个虚拟的用户。这个虚拟的用户具有系统管理员给他分配的各种可管理的和可操作的功能，当系统管理员把这个角色赋给一个或多个用户，这些用户就可以像这个虚拟的用户一样对相应的功能进行管理。它的数据流，如图所示：

****  
****  
**2.3 信息管理功能的实现**     
 包括理财收支管理、基本资料管理、理财债务基本信息管理以及对显示信息进行管理。在进入理财主管理界面后显示了所有的理财债务基本信息管，管理员可以根据实际需要选择理财类别名称，包括添加项目，修改以及注销理财分类信息。在基本资料管理模块中，管理员可以进行理财收入、理财支出、理财债务、理财投资、债务利率、投资回报、月收支统计，在点击任何一个按钮后，系统显示相应的全部信息，供管理员进行管理操作。在理财规则管理功能中，管理员可以设置对理财管理员的密码进行管理，以及重新登陆，退出等基本功能，主管理界面主要包含有收入，支出，债务，投资的全部信息查询。  
  
**2.4 查询功能的实现**  
 在主界面中，管理员通过选择查询的条件，可以为日期或债务归还，投资出售等信息，同时在本模块中还可以查询各大网站的相关信息，在进行相应信息的添加，点击查询按钮，系统给出所有符合条件的家庭理财信息。如图所示：  


**2.5 收入增删改功能实现**     
 进行收入的增删改功能界面如图所示：  
  
**2.6 支出增删改功能实现**    
 支出增删改界面如图所示：  
  
  
**2.7 更改密码功能的实现**  
 进行修改密码功能界面如图所示：  


**3.数据库分析**  
**3.1系统配置**  
（1）服务器端  
 服务器端的配置是由建立站点所需的软件来决定的。在最低配置下，服务器虽然能正常工作，但其性能往往不尽如人意，如果我们想使站点服务器工作得更好的话，我们需要更好的配置，特别是服务器的内存应该大于 256MB。下面为服务器端的最低配置：  
  
处理器：Intel Pentium 486以上  
内存：256MB以上  
硬盘空间：80G 以上  
光驱：需要从光盘上安装操作系统和其它开发站点所需的软件  
网卡：ISA 或者 PCI 接口均可  
显卡：SVGA 显示适配器  
（2）客户端  
内存：8MB  
硬盘空间：300MB  
网卡：ISA 或者 PCI  
显卡：SVGA 显示适配器  
**3.2代码设计**     
 本家庭理财管理系统的代码设计遵循标准化原则，对代码的很多方面进行了规范，例如：在该系统地设计中，局部变量使用时，尽量避免使用全局变量，因为全局变量的作用范围比较大，再对代码进行修改时，不小心就会修改到不必要修改的部分，使得系统部能够正常使用。另外，代码设计还遵从了可读性高，可使用性强等的设计原则，对代码进行了必要的注释，在代码的编写过程中，尽量对代码的格式进行了规范，因为一个好的代码必须要有一个比较清新的书写过程和思路，而这些都可以从代码的格式上体现出来，当然，代码的格式也是给读代码的人第一印象的关键因素，就如同一个系统软件的界面一样有着十分重要的作用。

1. **系统主要模块的实现**  
   **4.1 系统登陆界面的实现**  
    进入登陆界面，将提示用户输入用户名和密码，调用数据库存储的用户信息进行校验，只有当用户名和密码相匹配时，才允许用户使用该系统。  
    用户登录系统后能够允许使用的系统功能和用户的权限有关。普通家庭成员只能进行查询和浏览，家长还可以进行增、删、改操作和对普通家庭成员的管理。权限的设置在家庭成员信息中进行限制，权限的判定在家庭成员登录模块进行。  
    登陆界面实现过程：现在VS下创建一个新的Windows窗口，在窗口中添加Tabel表格，用于页面的布局，在Table表格中添加相关的控件，用于具体的功能实现。  
    登录代码的实现：首先导入必要的命名空间，然后在page\_load单击事件中，利用Session对象判断用户登录的状态，若登录成功则跳转到主页面，最后在双击功能控件编辑具体代码，主要代码如下：

//登录按钮  
private void btSure\_Click(object sender, System.EventArgs e)  
{  
try  
{  
string sql="select uName,PWD from family where uName='"+this.comboName.Text +"'";  
if(oleConnection1==null)  
oleConnection1 = MainForm.getConnection();  
OleDbCommand cmd = oleConnection1.CreateCommand();  
cmd.CommandText =sql;                  
OleDbDataReader dr = cmd.ExecuteReader();                 
if(!dr.Read())  
{  
MessageBox.Show("无此用户，请重试！","提示");  
}  
else     
{     
if(!dr.GetString(1).Equals (this.password.Text.ToString()))  
{  
MessageBox.Show("\n密码错误，请重试！","提示");  
dr.Close();cmd.Dispose();  
return;  
}                     
mainmenu=((MainForm)(this.MdiParent)).getMainMenu();  
for (int i=0;i<6;i++)  
{  
mainmenu.MenuItems[i].Visible=true;  
}  
StatusBar statusBar = ((MainForm)(this.MdiParent)).getStatusBar();  
statusBar.Panels[0].Text="当前登录用户";  
statusBar.Panels[1].Text=this.comboName.Text.Trim();  
statusBar.Panels[3].Text = "家庭理财系统";  
dr.Close();cmd.Dispose();this.Close();  
}                         
}  
catch(Exception ee)  
{  
MessageBox.Show("\n"+ee.ToString());  
}  
}

**4.2修改密码的实现**

//修改密码

Public void actionPerformed(ActionEvent e)

{

If(e.getSourse()==CancelBtn){}else

If(e.getSourse()==UpdateBtn)

Db.closeConnection();

This.dispose();

try[ char[] password=PasswordTextField.getPassword

String passwordSTR=new String(password); char[]

newPassword=NewPasswordTextField.getPassword; String

newPasswordSTR=new String(newPassword);

PasswordConfirmTextField.getPassword char[]

confirmPassword=StringconfirmPasswordSTR=newString(confirmPassword);

String strSQL="select \* from usertable where username=&#39;"+

UserTextField.getText()trim)+" &#39;and password=&#39;"+

passwordSTR+" &#39;"; if(UserTextField.getTexttrimquals(") 1 ] else

if(passwordSTR.equals("') ] else

if(!newPasswordSTR.equals(confrmPasR [] else

{ if(!db.getResult(strSQL).first())

JOptionPane.showMessageDialog(ull,"此用户不存在或者原密码不正确! ");

{JOptionPane.showMessageDialog(null,"原密码不能为空! "); JOptionPane.showMessageDialog(ull,"用户名不能为空! ");

JOptionPane.showMessageDialog(null,"两次输入的新密码不一致! ");

}

else { strSQL="update usertable set} password=&#39;"+ newPasswordSTR+" &#39;where

username=&#39;" +User TextField.getText()

.trim()+" &#39;"; }

if(db.updateSq(strSQL)) { this.dispose();

JOptionPane.showMessageDialog(null,"更新密码成功! "); }

} else

JOptionPane.showMessageDialog(null I #T E 195JX! "); .

this.dispose(); } db.closeConnection();

catch(SQLException sqle) {} catch(Exception ex) i

}

System.out.printlnex.toString; System.out.println(sqle.toString());}

**4.3存取记录的核心代码**

**//**存取记录的核心代码

String strsql="select bankbalance from currentaccounttable where  
casno="+" &#39;"+caisnoJcomboBox.getSelectdlStri().trim()+"&#39;";  
rs=db.getResult(strsql);  
rs.first();

String bankbalance=rs.getString1);

if(caitypeJcomboBox.getSelecttigquals("))

String strSQL="insert into

currentaccountimfortable(caiid,caisno,cabankbalance,cai

type,caisum,ncabankbalance,caidate,caimember

values("· +caiidTextField.getTexttrim)+" ,&#39;"+ caisnoJcomboBox.getSelectedltem().toString.trim+"&#39;, &#39;" +bankbalance.toString().trim &#39;, &#39;"+caitypeJcomboBox.getSelectedtm(.toStrigt +" &#39;,&#39;" +caisumTextField.getText().trim()+"&#39,"+

Float.parseFloat(bankbalance.toString()+""+””+”

Float.parseFloat(caisumTextFieldgetTttim+" ,&#39;"

+caidate' TextField.getText().trim()+" &#39;, &#39;"+

caimemberJcomboBox. getSelectedltem().toString().trim(+"&#39;)";

if(db.updateSql(strSQL))

When casno=" +" &#39;"+caisnoJcomboBox

.getSelectedltem().toString().trim" &#39;";

System.out.println(strsql1); String strsql1="select

bankbalance from currentaccounttable

rs=db.getResult(strsql1); . rs.first();

String bankbalance1=rs.getString(1);

Stringrbanksno=caisnoJcomboBox .getSelectedltem().toString;

String rdate=caidateTextField.getText; intrbankbalance=lnteger.parselntbankbalance+lnteger.parselnt(caisumTextField.getet() Record r=new

Record(rbanksno,rdate,rbankbalance); r.record();

String strSQL1 = "update currentaccounttable set

bankbalance=bankbalance+"+"&#39;"+caisumTextField.getText().trim()+" &#39;"+" where casno="+"&#39;" +caisnoJcomboBox

.getSelectedltem().toString.trim"&#39;";

{

db.updateSql(strSQL1);

System.out.println(strSQL1); this.dispose();

JOptionPane.showMessageDialog(ull,"存款记录添加成功! "); }

else JOptionPane showMessageDialog(null,"存取款记录添加失败! ");

this.dispose(); } }

else{

String strSQL="insert into

currentaccountimfortable(caiid,caisno,cabankbalance,caitype,caisumnaala caidate,caimember)values("caiidTextField.getnext().trim()+",&#39;"+caisnoJcomboBox.getSelectedltem().toString().trim()+" &#39;,&#39;"+

bankbalance.toString().trim(" &#39;,&#39;"toString().trim()+" &#39;, &#39;"+

caisumTextField.getText().trim()+"&#39;,"++"-"+Float.parseFloat(caisumTextFieldgetTt.trim())+",&#39;"+caidateTextField.getTexttri+"&#39;,&#39;"+caimemberJcomboBox.getSelectedltem().toString).trim()+" &#39;)";

if(db.updateSql(strSQL)&amp;&amp;(Float.parseFloatbankbatS().trim())-Float.parseFloatcaimTiet(trim

Float.parseFloat(bankbalance.toString().trim(())

caitypeJcomboBox.getSelectedltem()))&gt;=0)

{

String strsql1="select bankbalance from currentaccounttable

When casno="+" &#39;"+caisnoJcomboBox .getSelectedltem().toString.trim" &#39;";

System.out.println(strsql1 rs=db.getResult(strsql1);

rs.first(); String bankbalance1=rs.getString1); String

rbanksno=caisnoJcomboBox .getSelectedltem().toString; String

rdate=caidateTextField.getText(); -Integer.parselnt(caisumTextField.get()

int rbankbalance=lnteger.parselntbankbala Record r=new

Record(rbanksno,rdate,rbankbalance); r.record();

String strSQL1 = "update currentaccounttable set

bankbalance=bankbalance-"+"&#39;"+caisumTextField. getText().trim()+ ' &#39;"+" Where casno="+"&#39;"+caisnoJcomboBox.getSelectedltem().toString().trim()"";

else db.updateSql(strSQL1);

JOptionPane.showMessageDialog(null,"取款记录添加成功! ");

this.dispose(); }

if(Float.,parseFloat(bankbalance.toString().trim())&lIt;0| I (Float.

parseFloat(bankbalance.toString().trim())-Float.parseFloat

(caisumTextField.getTextl).trim())&lt;0){

JOptionPane.showMessageDialog(null,"存取款记录添加失败!，余额

不足，不能取款");

{ }} } db.closeConnection();JOptionPane.showMessageDialog(null,"存取款记录添加失败! ");  
this.dispose();  
this.dispose(); }