****

**数据库系统课程设计报告**

题目:房屋租赁管理系统

课程代号: 0688116

课程名称：**数据库系统课程设计**

学号：20125580

姓名：时文涛

班级：计算机科学与技术2班

指导教师：郝刚、刘里

完成日期：2015年4月

1. 引言…………………………….3
2. 系统分析与设计……………….3
3. 系统开发及实现……………….9
4. 总结……………………………17
5. 心得体会及附录………...…….17

六、 参考文献…………….......…….19

1. 引言

随着房价的不断升高，人们对房屋的需求越来越难以满足，对于一些在外地工作的人来说租房子成为了一种常态，因此出现了越来越多的房屋中介，而对于房屋中介而言，管理日益增加租户的信息和房产的信息成了一个至关重要的问题。与此同时，环保也是大家关注的焦点。因此，开发一款能够实现无纸化的管理系统来管理房屋和租户的信息就显得十分必要。

房屋租赁管理系统的任务目标：简化房屋信息及租客信息的管理步骤，可以方便快捷地查询所需要的信息，系统自动进行水电等费用的计算。在系统启动时有对用户身份的检查，以保证信息的安全性。

房屋租赁管理系统的开发及运行环境：

开发环境：Windows7 64位

Visual Studio 2010

Mysql 5

运行环境：window

.NET4.0及以上

1. 系统分析与设计

2.1 需求分析

通过对任务的分析，可以发现运行的系统要能实现以下的功能：

（1）登录管理：实现以正确的员工号、密码和登录人员身份才能登陆系统的功能，正确登录之后才能进行之后的管理。

（2）房屋信息管理：包括添加新的房屋信息、修改房屋信息、查询房屋信息、删除房屋信息。

（3）用户信息管理：包括添加新的用户信息、修改用户信息、查询用户信息、删除用户信息。

（4）租赁信息管理：包括查询某个房屋的租赁状况、添加租赁关系、删除租赁关系。

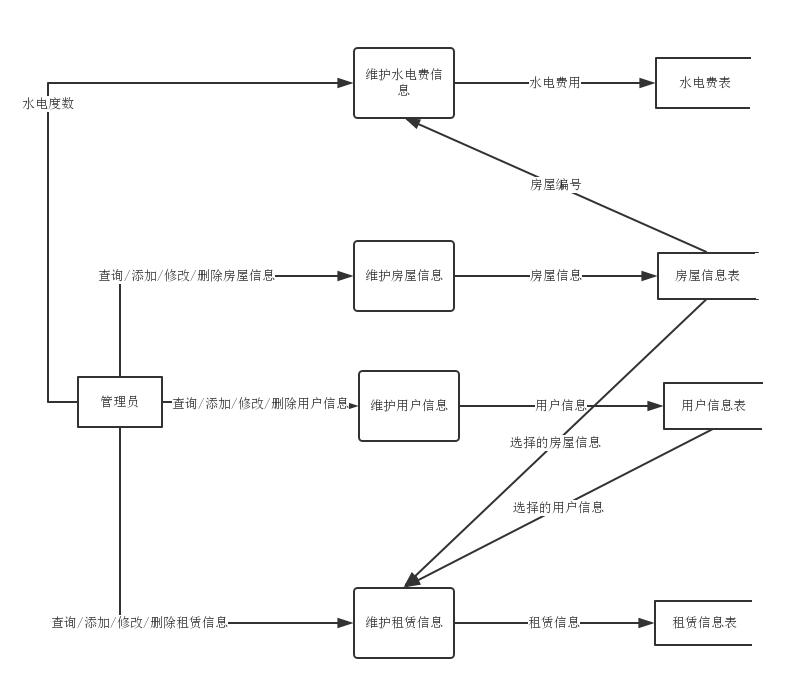
（5）水电费用管理：包括水电费用的查询、添加、修改、删除。

（6）财务收支记录：添加公司的财务记录（不允许修改和删除）。

（7）收费项目设置：设置水费、电费、物业费、宽带费单价及财务余额。

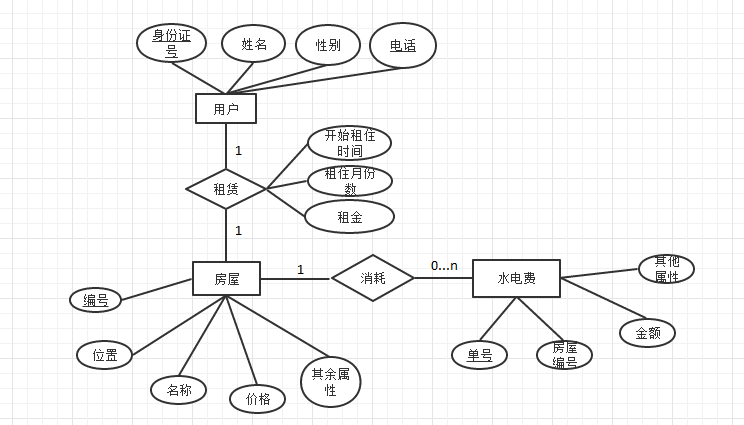
（8）管理员信息维护/密码修改：若登陆用户为超级管理员(admin)，则可以管理其他员工的信息，否则就只有修改自己密码的权限。

数据流图如下：



2.2数据库概念结构设计

E-R图



2.3数据库逻辑结构设计

关系模式（带有下划线的为主键，所有表中数据均为varchar类型）：

管理员（用户名、密码）

房屋信息（房屋编号、所属区域、房屋名称、房屋类型、房屋面积、装修情况、房屋设施、房屋用途、月租价格、房间状态、备注说明）

用户信息（身份证号、用户姓名、性别、电话）

租赁信息（房屋编号、租户身份证号、租赁开始时间、租赁月数 、月租金）

水电费信息（受理单号、房屋编号、上次电表数、本次电表数、上次水表数、本次水表数、收费金额）

财务收支报（编号、类型、金额、日期）

杂项设置（编号、水费单价、电费单价、宽带费单价、物业费单价、财务余额）

注：编号位数字等限制由程序中的函数完成，故所有数据类型均为varchar

管理员表（manageruser）：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 含义 | 数据类型 | 约束 |
| user | 管理员用户名 | varchar（20） | 主键 |
| password | 密码 | varchar（20） | 不为空 |

房屋信息表（roominf）：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 含义 | 数据类型 | 约束 |
| num | 房屋编号 | varchar（10） | 主键 |
| weizhi | 房屋位置 | varchar（30） | 不为空 |
| mingzi | 房屋名称 | varchar（20） | 不为空 |
| zhonglei | 房屋类型 | varchar（10） | 不为空 |
| mianji | 房屋面积 | varchar（11） | 不为空 |
| zhuangxiu | 装修状况 | varchar（4） | 不为空 |
| sheshi | 屋内设施 | varchar（20） | 不为空 |
| yongtu | 房屋用途 | varchar（4） | 不为空 |
| jiage | 房屋价格 | varchar（11） | 不为空 |
| zhuangtai | 房屋状态 | varchar（4） | 不为空 |
| beizhu | 备注信息 | varchar（40） | 不为空 |

用户信息表（userinf）：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 含义 | 数据类型 | 约束 |
| sfzid | 身份证号 | varchar（18） | 主键 |
| username | 姓名 | varchar（10） | 不为空 |
| xingbie | 性别 | varchar（2） | 不为空 |
| dianhua | 电话 | varchar（14） | 不为空 |

房屋租赁表（roomrent）：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 含义 | 数据类型 | 约束 |
| roomnum | 房屋编号 | varchar（10） | 主键 |
| sfznum | 身份证号 | varchar（18） | 主键 |
| starttime | 开始租住时间 | varchar（12） | 不为空 |
| monthnum | 租住月份数 | varchar（4） | 不为空 |
| monthlyrent | 月租金 | varchar（20） | 不为空 |

水电费表（billsinf）：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 含义 | 数据类型 | 约束 |
| danhao | 受理单号 | varchar（15） | 主键 |
| roomnum | 房屋编号 | varchar（10） | 不为空 |
| predianbiao | 上次电表数 | varchar（15） | 不为空 |
| dianbiao | 本次电表数 | varchar（15） | 不为空 |
| preshuibiao | 上次水表数 | varchar（15） | 不为空 |
| shuibiao | 本次水表数 | varchar（15） | 不为空 |
| jine | 金额 | varchar（20） | 不为空 |

财务报表(caiwu):

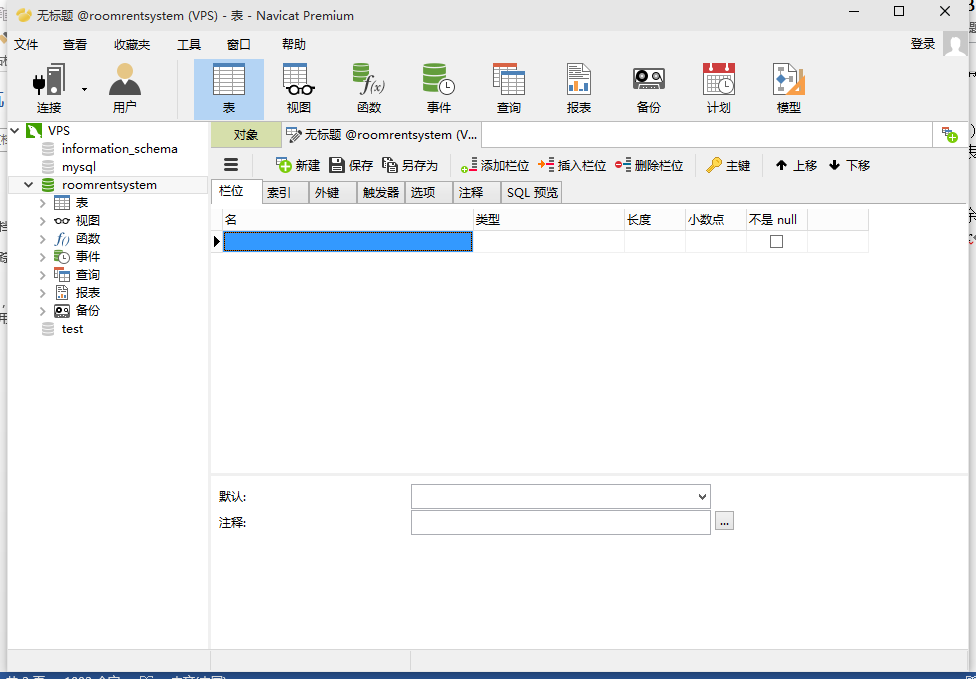
|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 含义 | 数据类型 | 约束 |
| bianhao | 编号 | varchar（20） | 主键 |
| leixing | 类型 | varchar（2） | 不为空 |
| jine | 金额 | varchar（15） | 不为空 |
| date | 日期 | varchar（12） | 不为空 |

杂项设置表（othersetting）：

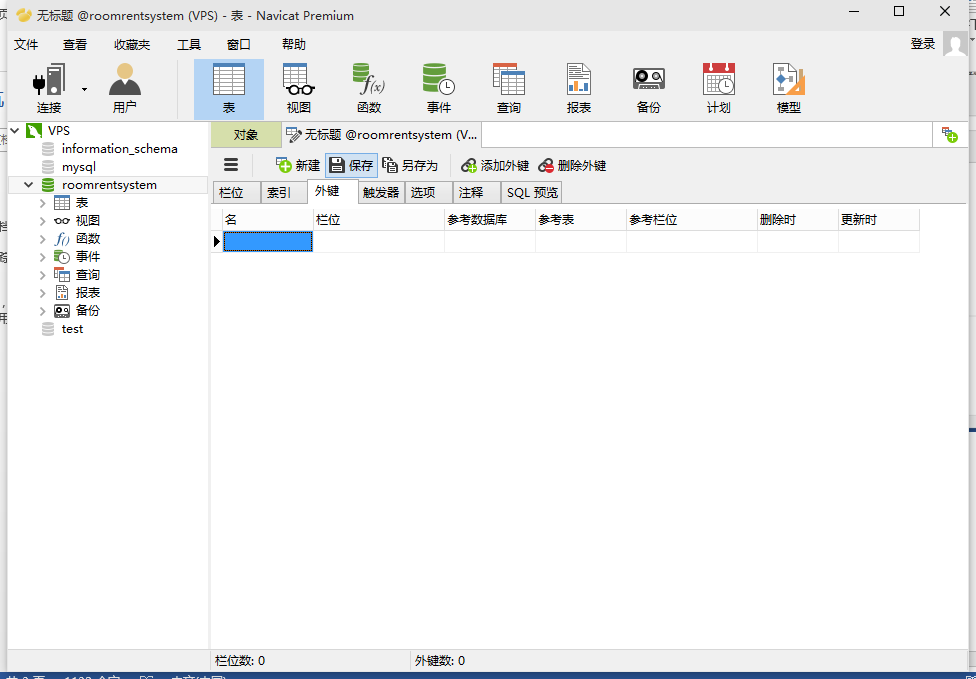
|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 含义 | 数据类型 | 约束 |
| bianhao | 编号 | varchar（10） | 主键 |
| shuiprice | 水价 | varchar（10） | 不为空 |
| dianprice | 电价 | varchar（10） | 不为空 |
| wangprice | 网费 | varchar（10） | 不为空 |
| wuyeprice | 物业费 | varchar（10） | 不为空 |
| yue | 余额 | varchar（10） | 不为空 |

2.4数据库创建

在实验室实验室使用的是phpadmin建立的数据库roomrentsystem，本机上没有，用Navicat演示、下图为创建表的界面：



下图为创建外码的界面：



若使用SQL命令建表则如下：

水电费表：

CREATE TABLE IF NOT EXISTS `billsinf` (

`danhao` varchar(15) COLLATE utf8\_unicode\_ci NOT NULL,

`roomnum` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,

`predianbiao` varchar(15) COLLATE utf8\_unicode\_ci NOT NULL,

`dianbiao` varchar(15) COLLATE utf8\_unicode\_ci NOT NULL,

`preshuibiao` varchar(15) COLLATE utf8\_unicode\_ci NOT NULL,

`shuibiao` varchar(15) COLLATE utf8\_unicode\_ci NOT NULL,

`jine` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL,

PRIMARY KEY (`danhao`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

财务表：

CREATE TABLE IF NOT EXISTS `caiwu` (

`bianhao` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL,

`leixing` varchar(2) COLLATE utf8\_unicode\_ci NOT NULL,

`jine` varchar(15) COLLATE utf8\_unicode\_ci NOT NULL,

`date` varchar(12) COLLATE utf8\_unicode\_ci NOT NULL,

PRIMARY KEY (`bianhao`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

管理员信息表：

CREATE TABLE IF NOT EXISTS `manageruser` (

`user` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL,

`password` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL,

PRIMARY KEY (`user`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

杂项设置表：

CREATE TABLE IF NOT EXISTS `othersetting` (

`bianhao` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,

`shuiprice` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,

`dianprice` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,

`wangprice` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,

`wuyeprice` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,

`yue` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,

PRIMARY KEY (`bianhao`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

房屋信息表：

CREATE TABLE IF NOT EXISTS `roominf` (

`num` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,

`weizhi` varchar(30) COLLATE utf8\_unicode\_ci NOT NULL,

`mingzi` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL,

`zhonglei` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,

`mianji` varchar(11) COLLATE utf8\_unicode\_ci NOT NULL,

`zhuangxiu` varchar(4) COLLATE utf8\_unicode\_ci NOT NULL,

`sheshi` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL,

`yongtu` varchar(4) COLLATE utf8\_unicode\_ci NOT NULL,

`jiage` varchar(11) COLLATE utf8\_unicode\_ci NOT NULL,

`zhuangtai` varchar(4) COLLATE utf8\_unicode\_ci NOT NULL,

`beizhu` varchar(40) COLLATE utf8\_unicode\_ci NOT NULL,

PRIMARY KEY (`num`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

租赁表：

CREATE TABLE IF NOT EXISTS `roomrent` (

`roomnum` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,

`sfznum` varchar(18) COLLATE utf8\_unicode\_ci NOT NULL,

`starttime` varchar(12) COLLATE utf8\_unicode\_ci NOT NULL,

`monthnum` varchar(4) COLLATE utf8\_unicode\_ci NOT NULL,

`monthlyrent` varchar(20) COLLATE utf8\_unicode\_ci NOT NULL,

PRIMARY KEY (`roomnum`,`sfznum`),

KEY `userinfyueshu` (`sfznum`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

用户信息表：

CREATE TABLE IF NOT EXISTS `userinf` (

`sfzid` varchar(18) COLLATE utf8\_unicode\_ci NOT NULL,

`username` varchar(10) COLLATE utf8\_unicode\_ci NOT NULL,

`xingbie` varchar(2) COLLATE utf8\_unicode\_ci NOT NULL,

`dianhua` varchar(14) COLLATE utf8\_unicode\_ci NOT NULL,

PRIMARY KEY (`sfzid`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

约束关系：

ALTER TABLE `roomrent`

ADD CONSTRAINT `roominfyueshu` FOREIGN KEY (`roomnum`) REFERENCES `roominf` (`num`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `userinfyueshu` FOREIGN KEY (`sfznum`) REFERENCES `userinf` (`sfzid`) ON DELETE CASCADE ON UPDATE CASCADE;

1. 系统开发及实现

论述设计开发采用的技术及设计效果（逐模块或主要模块）

对模块要论述要实现的功能、界面设计效果、实现方法）

开发本系统采用C#语言+Mysql数据库。数据显示使用了DataGridView控件

3.1登陆模块

界面设计如下：

登陆模块为Login.cs：当用户点击登陆按钮时用用户名和密码向数据库中的manageruser表提交查询，若校验通过则将用户名作为初始化参数传递到MainForm.cs、否则提示用户不存在或密码错误。

主要代码：

sqltext1 = "select password from manageruser where user='" + username + "'";

ds = MySqlHelper.GetDataSet(MySqlHelper.Conn, CommandType.Text, sqltext1, null);

MainForm mainform = new MainForm(username);

主界面代码：

在窗口初始化时通过函数重载，获得当前登陆的用户的用户名

public MainForm()

{

usernamestr = "admin";

InitializeComponent();

}

public MainForm(string user)

{

usernamestr = user;

InitializeComponent();

}

根据登陆的用户不同，顶部工具菜单第二项可能为管理员信息维护、修改密码，登陆用户若为admin则为管理员信息维护，否则为修改密码。

窗口的主体部分用SplitContainer空间分割为两部分，左侧为功能列表、右侧为具体的操作窗口。当点击功能能列表中的不同按钮时、右侧操作窗口内容随之改变。

关键代码（切换右侧操作窗口）：

splitContainer1.Panel2.Controls.Clear();

Rentinf rentinf = new Rentinf();

rentinf.TopLevel = false;

rentinf.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;

rentinf.WindowState = FormWindowState.Normal;

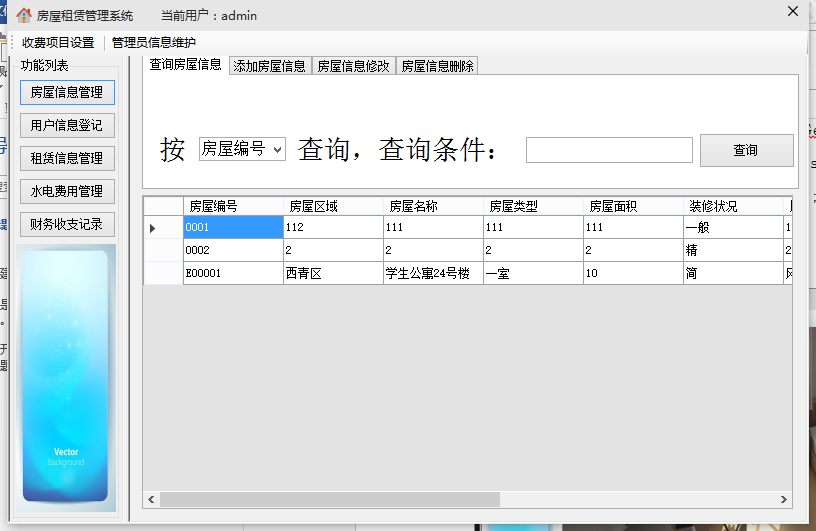
rentinf.Dock = DockStyle.Fill;

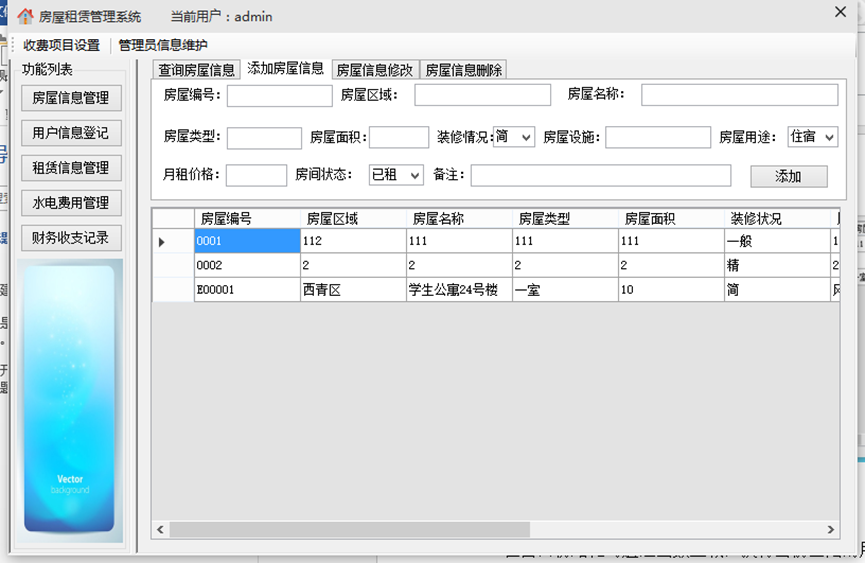
rentinf.KeyPreview = true;

rentinf.Parent = splitContainer1.Panel2;

rentinf.Show();

3.2房屋信息管理模块

界面设计如下：

可以按照房屋编号、房屋名称、房屋区域、房屋状态进行查询

查询关键代码：

sqltext1 = "select \* from roominf where " + chaxunfangshi + " like '%" + textBox1.Text + "%'";

roomdataGridView.DataSource = MySqlHelper.GetDataSet(MySqlHelper.Conn, CommandType.Text, sqltext1, null).Tables[0].DefaultView;

添加关键代码：

sqltext="insert into roominf(num,weizhi,mingzi,zhonglei,mianji,zhuangxiu,sheshi,yongtu,jiage,zhuangtai,beizhu)values('" + num + "','" + weizhi + "','" + mingzi + "','" + leixing + "','" + mianji + "','" + zhuangxiu + "','" + sheshi + "','" + yongtu + "','" + jiage + "','" + zhuangtai + "','" + beizhu + "')";

MySqlHelper.ExecuteNonQuery(MySqlHelper.Conn,CommandType.Text,sqltext, null);

roomdataGridView.DataSource = MySqlHelper.GetDataSet(MySqlHelper.Conn, CommandType.Text,"select \* from roominf ", null).Tables[0].DefaultView;

修改代码：

sqltext = "update roominf set weizhi='" + weizhi + "',mingzi='" + mingzi + "',zhonglei='" + leixing + "',mianji='" + mianji + "',zhuangxiu='" + zhuangxiu + "',sheshi='" + sheshi + "',yongtu='" + yongtu + "',jiage='" + jiage + "',zhuangtai='" + zhuangtai + "',beizhu='" + beizhu + "' where num='" + num + "'";

MySqlHelper.ExecuteNonQuery(MySqlHelper.Conn, CommandType.Text, sqltext, null);

roomdataGridView.DataSource = MySqlHelper.GetDataSet(MySqlHelper.Conn, CommandType.Text, "select \* from roominf ", null).Tables[0].DefaultView;

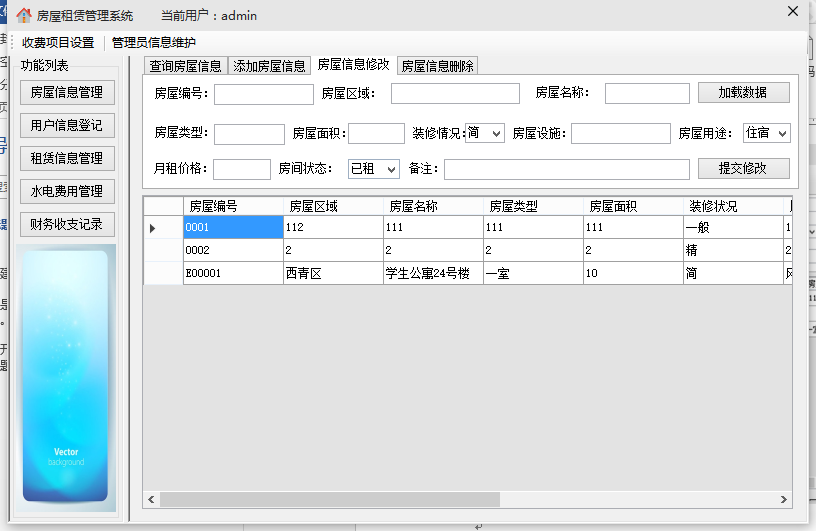
删除关键代码：

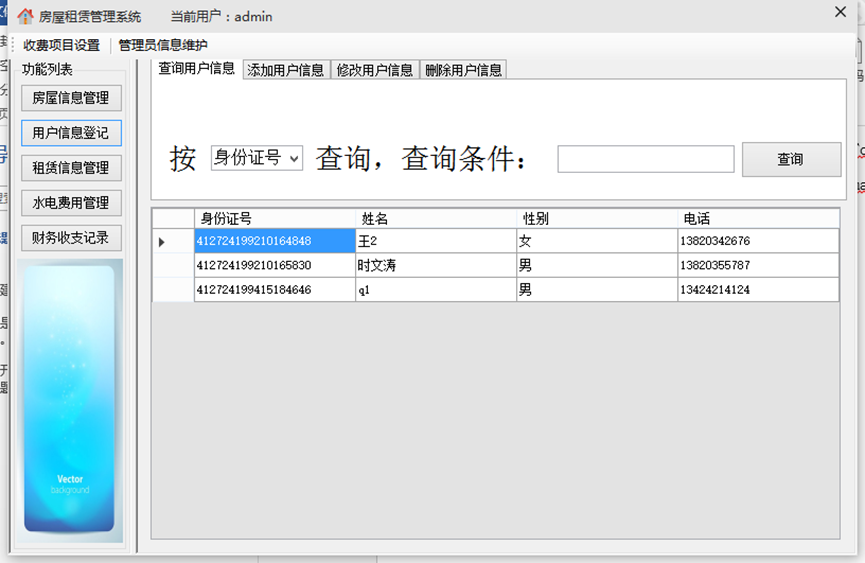
sqltext = "delete from roominf where num='" + num + "'"; MySqlHelper.ExecuteNonQuery(MySqlHelper.Conn,CommandType.Text, sqltext, null);

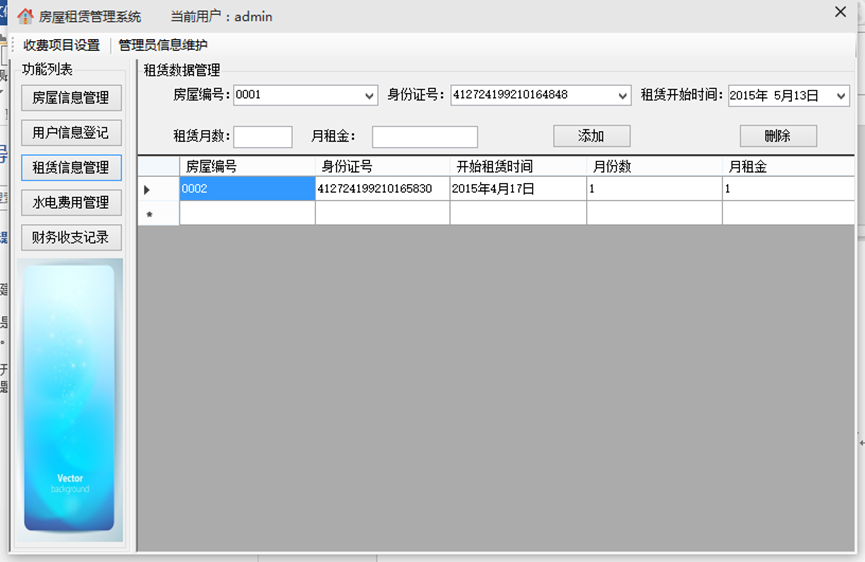
roomdataGridView.DataSource=MySqlHelper.GetDataSet(MySqlHelper.Conn,CommandType.Text, "select \* from roominf ", null).Tables[0].DefaultView;

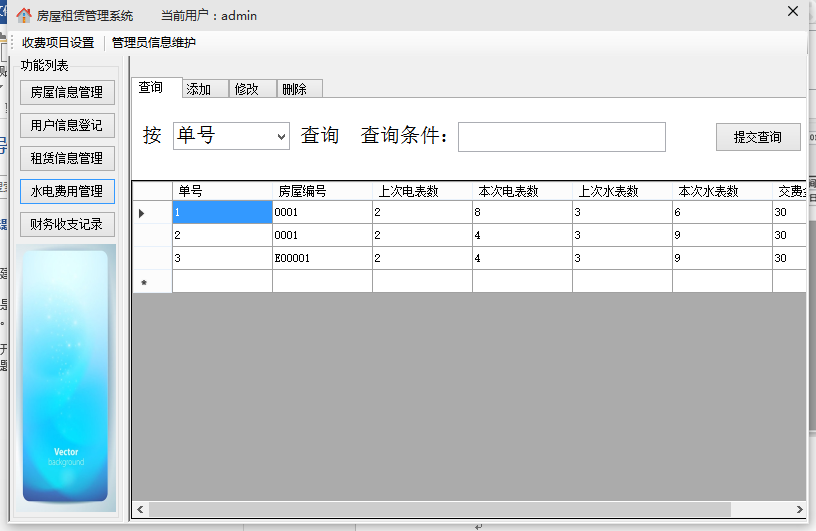
3.3用户信息管理模块

可以按照身份证号、姓名、电话查询相应的用户信息

添加、修改、删除操作与房屋信息管理类似。在代码中设置了身份证号和电话号码，必须为数字的函数进行处理。

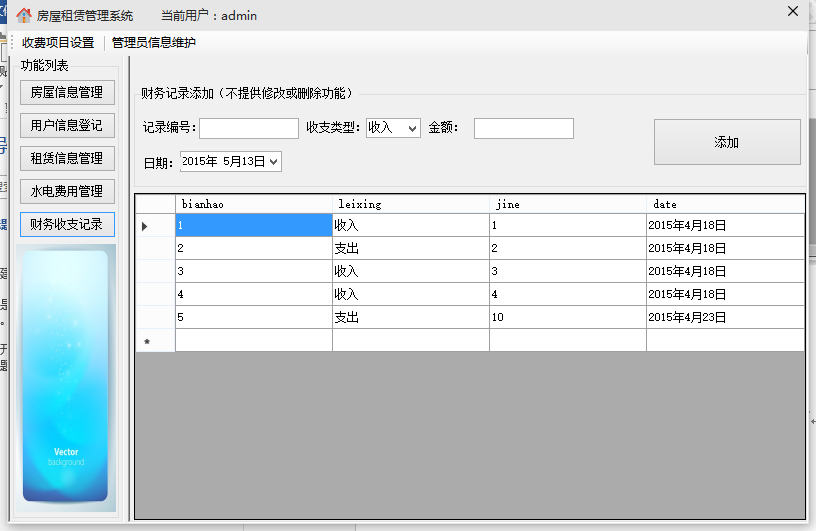
3.4租赁信息管理模块

用于查询、记录房屋租赁信息。期中由于数据房屋编号和身份证号是外键，分别参照房屋信息表和用户信息表，所以房屋信息和身份证号使用了comobox控件来完成选择，不允许手工输入。

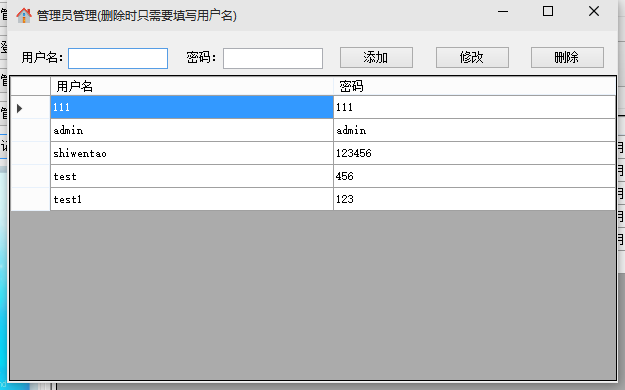
3.5水电费用管理模块

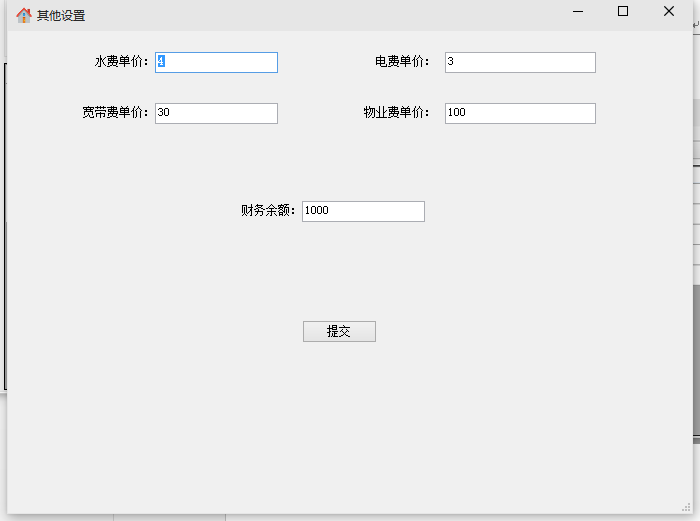
用于记录出租的房屋的水电费用信息。由电表数据来计算应缴费的金额，并存入数据库。

添加、修改、删除操作与房屋信息管理类似。

3.6财务收支模块

本模块用于记录公司的收支状况，为防止篡改，不允许进行修改和删除。

3.7其他

Admin用户登陆后工具菜单中显示相应的菜单可以进入，用于进行系统管理员信息的修改和删除、若非admin管理员登陆，则不显示该菜单，只显示修改密码的菜单。

其他设置模块用于设置水费电费等设置。仅与费用计算有关。

1. 总结

学习中，只学习了一些基本的SQL语句特点及操作方法，但这次课程设计是要求作出一个独立的系统，不仅要用到SQL知识，还有用到其他的软件，这些软件大部分对我来说都是陌生的，这让我在设计的过程中遇到了很大的困难，课设后我稍加总结，我在课程设计中遇到了以下方面的问题：

1. 本次课设我采用了Mysql 作为建立数据库的工具，C#作为开发界面的工具。Mysql对我来说都是不熟悉的，在这里我遇到了第一个难题，如何在C#中使用Mysql数据库，一度设计进度为零，但后来经过在网上查阅资料和自己摸索，慢慢的我能作出一些界面了。这一个麻烦算是解决了。
2. 在做好界面之后，又遇到了另一个问题，不知道怎么把设计好的界面和SQL建立的数据库连接在一起，如果不能连接在一起的话，那这次课设将是不成功的，后经去图书馆查阅资料，上网搜索资料，和同学讨论，在了解到想要把C#做的界面的Mysql建立的数据库连接在一起必须要用到数据库操作类，在从网上找到相应的操作类后，既能连接到数据库了，能够实现一些基本的功能。
3. 数据库中数据的显示也是困扰了我很久的问题，经过网络查询最终我选择了DataGridView来显示数据。

心得体会：

这次课程设计对我的意义很大。不仅巩固了就知识，还学会了新知识。更是激起了我对数据库的兴趣。

设计时，遇到了很多在书本中和课堂上没有机会遇到的问题，在课堂上只能学习到数据库中有一些最为基本的知识，但在课程设计中这些基本知识只是一个基础。这体现在课程设计中的是我花费近一半的时间去学习怎么把C#和Mysql数据库连接在一起，怎么能实现一个界面到另一个界面的跳转，怎么通过直观上界面的操作改变个显示数据库中的信息。这个过程是很痛苦的，一方面是对设计出一个系统的兴趣驱使我不断的去学习新知识和改正错误；另一方面是对设计出一个系统没有任何经验，甚至在设计过程中没有人可以指导你或是和你有一起讨论，一度都想过随便弄一个现有的程序修改一下交差。但是看到运用所学知识做出的成果时，还是选择继续做下去，哪怕不能实现太多的更能，但最起码要凭借自己去做出一个系统。

在课设中遇到了诸多苦难和千奇百怪的错误，一度灰心过。但当改正错误时，不由兴起一股成就感，这些改正后的错误一直激励我坚持到最后，从这次的课程设计中我巩固了以前所学，新学习到了C#知识，遇到了很多错误，但是最终也完成了一个系统，这次课程设计对我来说是使我受益匪浅！

附录（部分源代码）：

判断textbox中获取的string是否为数字的函数：

using System.Text.RegularExpressions;

public bool IsNum(String strNumber)

{

Regex objNotNumberPattern = new Regex("[^0-9.-]");

Regex objTwoDotPattern = new Regex("[0-9]\*[.][0-9]\*[.][0-9]\*");

Regex objTwoMinusPattern = new Regex("[0-9]\*[-][0-9]\*[-][0-9]\*");

String strValidRealPattern = "^([-]|[.]|[-.]|[0-9])[0-9]\*[.]\*[0-9]+$";

String strValidIntegerPattern = "^([-]|[0-9])[0-9]\*$";

Regex objNumberPattern = new Regex("(" + strValidRealPattern + ")|(" + strValidIntegerPattern + ")");

return !objNotNumberPattern.IsMatch(strNumber) && !objTwoDotPattern.IsMatch(strNumber) && !objTwoMinusPattern.IsMatch(strNumber) && objNumberPattern.IsMatch(strNumber);

}

数据库操作类MySqlHelper.cs：关键代码

public static int ExecuteNonQuery(string connectionString, CommandType cmdType, string cmdText, params MySqlParameter[] commandParameters)

{

MySqlCommand cmd = new MySqlCommand();

using (MySqlConnection conn = new MySqlConnection(connectionString))

{

PrepareCommand(cmd, conn, null, cmdType, cmdText, commandParameters);

int val = cmd.ExecuteNonQuery();

cmd.Parameters.Clear();

return val;

}

}

public static DataSet GetDataSet(string connectionString, CommandType cmdType, string cmdText, params MySqlParameter[] commandParameters)

{

//创建一个MySqlCommand对象

MySqlCommand cmd = new MySqlCommand();

//创建一个MySqlConnection对象

MySqlConnection conn = new MySqlConnection(connectionString);

//在这里我们用一个try/catch结构执行sql文本命令/存储过程，因为如果这个方法产生一个异常我们要关闭连接，因为没有读取器存在，

try

{

//调用 PrepareCommand 方法，对 MySqlCommand 对象设置参数

PrepareCommand(cmd, conn, null, cmdType, cmdText, commandParameters);

//调用 MySqlCommand 的 ExecuteReader 方法

MySqlDataAdapter adapter = new MySqlDataAdapter();

adapter.SelectCommand = cmd;

DataSet ds = new DataSet();

adapter.Fill(ds);

//清除参数

cmd.Parameters.Clear();

conn.Close();

return ds;

}

catch (Exception e)

{

throw e;

}

}

参考文献

[1]. 萨师煊著系统概论（第四版），2006：P91-P117。

[2].马俊等著C#程序设计及应用教程（第二版），2009:P220-P230。

[3].历小军等著软件开发过程及规范，2013:P178-P180.

[4]. 王华杰 黄山 著《精通C#数据库编程》 科学出版社。

[5]. 周峰 李德路 王征 著《Visual C#.NET 20。中文版基础与实践教程》电子工业出版社。

[6]. 康莉 著《新手学Visual C# 2008 程序设计》电子工业出版社。