目 录

[1、项目分析 3](#_Toc43624606)

[1.1、项目背景 3](#_Toc43624607)

[1.2、需求分析 3](#_Toc43624608)

[2、系统概要设计 3](#_Toc43624609)

[2.1、系统设计思想 4](#_Toc43624610)

[2.2、开发环境 4](#_Toc43624611)

[2.3、总体结构设计 4](#_Toc43624612)

[2.4、数据库设计 5](#_Toc43624613)

[3、系统界面模块设计 6](#_Toc43624614)

[3.1、登录界面设计 7](#_Toc43624615)

[3.2、主窗口界面设计 7](#_Toc43624616)

[3.3、系统设置界面设计 8](#_Toc43624617)

[3.4、部门管理界面设计 9](#_Toc43624618)

[3.5、员工管理界面设计 11](#_Toc43624619)

[3.6、设备管理界面设计 11](#_Toc43624620)

[3.7、客户管理界面设计 13](#_Toc43624621)

[3.8、关于我们界面设计 14](#_Toc43624622)

[3.9、快捷访问设计 16](#_Toc43624623)

[4、系统详细设计 16](#_Toc43624624)

[4.1、数据库连接 16](#_Toc43624625)

[4.2、登录功能实现 17](#_Toc43624626)

[4.3、主窗口界面实现 20](#_Toc43624627)

[4.4、用户操作功能实现 25](#_Toc43624628)

[4.5、添加员工实现 31](#_Toc43624629)

[4.6、修改客户信息实现 37](#_Toc43624630)

[5、系统测试 38](#_Toc43624631)

[5.1、功能测试 38](#_Toc43624632)

[5.2、系统主窗口界面 39](#_Toc43624633)

[5.3、增删改查功能实现 39](#_Toc43624634)

[6、课程设计总结 45](#_Toc43624635)

# 1、项目分析

## 1.1、项目背景

随着科技的进步与社会的发展，越来越多的技术依赖计算机实现，在众多的计算机系统中，尤以windows为代表的窗体应用最受广大用户的欢迎，PC终端行业一直是科技发展的一大重点行业，无论是用户使用还是程序开发都离不开电脑的操作。经过数年的发展，windows窗体应用已经成为了人类社会发展的重要产品。而对于企业而言，打造一款完美的网络协议窗体应用也是其不懈最求的目标。对于个人而言，掌握一种网络协议窗体应用已然是程序开发的必然趋势。

## 1.2、需求分析

互联网技术的飞速发展的当下，以Windows、Macos和Linux为主的两大PC终端操作系统成为了计算机行业三足鼎立的局面。而应用最广、最受广大用户接收的则是windows窗体应用。Microsoft windows操作系统问世于1985年，起初仅仅是为Microsoft-DOS模拟环境，后续的系统版本由于微软不断的更新升级，不但易用，也当是当前应用最为广泛的操作系统。现如今windows已经发展成为了一个用户量极大，服务最为广泛的窗体化应用操作系统，以Visual Studio为代表的一系列windows应用开发工具得到了广泛的利用。本次课程设计既是以Visual Studio软件为工具，以 C#语言为依托，学习windows系统环境下网络协议的分析于运用。通过windows窗口化程序，动态的代理技术、TCP/UDP的协议分析，掌握网络协议的产生于传递，学会使用C#语言，灵活的运用网络协议知识完成程序的开发设计。

# 2、系统概要设计

随着Windows窗体应用的深入人心和网络技术的不断发展，以华为为代表的第五代通讯技术(5G)一度成为了人们关注的热点。PC端应用一直以来都是IT界的一片热土。其中以windows系统环境下的窗体应用经过了数年的发展，已经发展成了强大的生态体系。提供了众多的插件、组件和开发接口吸引着众多的开发者和用户。因此，在互联网时代掌握网络协议分析技术已经是形势所趋。对于众多的企业来讲，普遍有着复杂的企业资源，例如：多样的部门、众多的员工、海量的数据资源和广泛的客户资源。这些复杂的关系网络，迫切的需要一款程序解决。在网络协议分析场景下的C#程序设计以windows窗体应用为依托，通过数据库的加持，因地制宜分析企业的资源关系网络，量身打造一个适合的管理系统，同时满足广大用户的需求既是本次程序设计的初衷。

## 2.1、系统设计思想

在大数据时代任何事物都是数据，都对企业存在着已知的或者潜在的价值，发现价值利用价值正是企业不断发展的动力。易软企业资源管理系统从企业的资源出发，瞄准当下企业业务层级的复杂多变的现状，从企业的产生到不断发展过程出发研究，对各个层面展开数据库设计，从部门关系、员工管理、企业设备管理和客户关系网络分析到整体设计，在根源上理清企业资源的关系网络，采用C#程序设计语言，通过windows窗体应用为用户提供了一套完美的交互体验。

## 2.2、开发环境

|  |  |  |
| --- | --- | --- |
| 分类 | 名称 | 版本 |
| 操作系统 | Windows | 10 |
| 数据库 | Mysql | 8.0.12 |
| 开发工具 | Visual Studio | 2017 |
| 数据库操作工具 | Navicat for Mysql | 10.1.7 |
| 数据库集成环境 | Phpstudy | 8.0.9..1 |
| 数据库生成工具 | PowerDesigner | 15.1 |

## 2.3、总体结构设计

易软企业资源管理系统是当前IT行业发展的产物，旨在建立一个完备的企业管理系统，在锻炼自身网络协议分析能力以及C#程序设计能力的同时，检验自身这一阶段的学习成果。

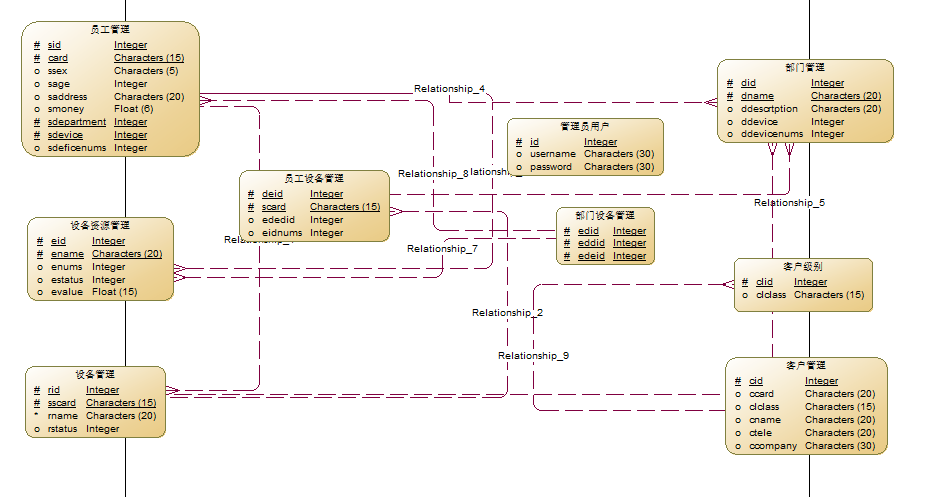
“易软企业资源管理系统”主要分为六个模块。其中包括：系统设置、部门管理、员工管理、设备管理、客户管理和关于我们等几个模块。通过复杂的代码逻辑将着几个模块联系在一起，在不影响功能的同时，充分优化的UI界面，力图达到最佳的交互体验。

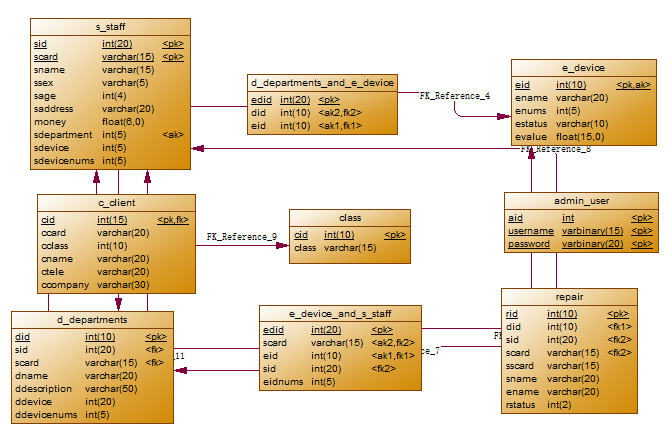
使用xmind设计系统的整体框架图如下：



## 2.4、数据库设计

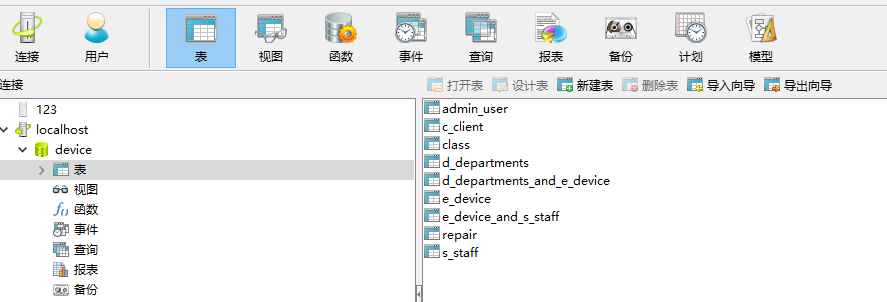
首先根据企业管理系统的业务模型设计实体，针对每个实体填充属性，确定每个属性的值以及主键、外键等信息。每个实体之间的键值关系，根据设计好的数据模型，使用Power Designer设计数据库的概念模型、逻辑模型和物理模型进而导出数据库报告，生成数据库源码。在Mysql官方网站下载MySql.Data.dll，方便以后使用C#语言调用数据库。





打开PhpStudy集成环境启动Mysql，打开navicat连接数据库并将sql文件导入到mysql中。





# 3、系统界面模块设计

本次系统界面设计用到的图片素材来自网络资源，下载后用Photo Shop进行整合加工而成。用到的图标logo来自于“阿里巴巴矢量图标库”。按钮、输入框等组件设计，复选框设计充分考虑交互体验；如：让页面不可放大，复选框不可编辑等内容。

## 3.1、登录界面设计

登录界面采用网络素材图片通过PhotoShop自行设计而成。



## 3.2、主窗口界面设计

输入用户名和密码正确后就会进入主窗口程序，主窗口界面使用多个菜单栏和一个图片式的设计构成一个简约的欢迎页面。还有几个logo图标。



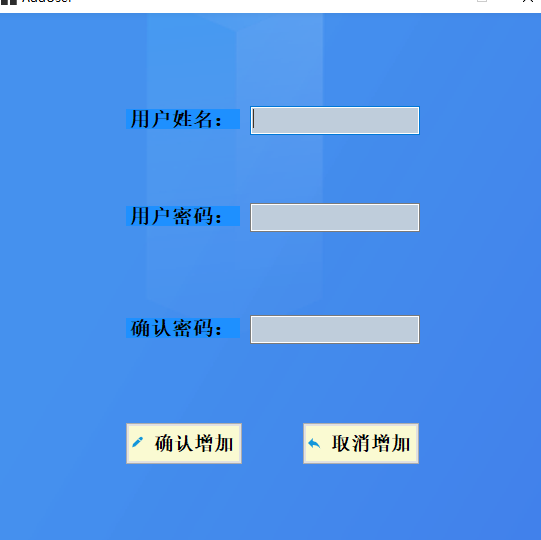
## 3.3、系统设置界面设计

系统设置采用下拉菜单设计，用户可以通过点击菜单选择修改管理员密码、新增管理员和退出系统。

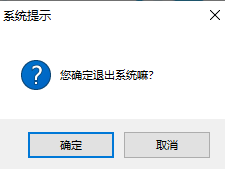
修改管理员密码页面设计图片如下



添加管理员用户UI设计图片



退出系统界面直接给出一个提示框让用户点击确认即可退出系统。



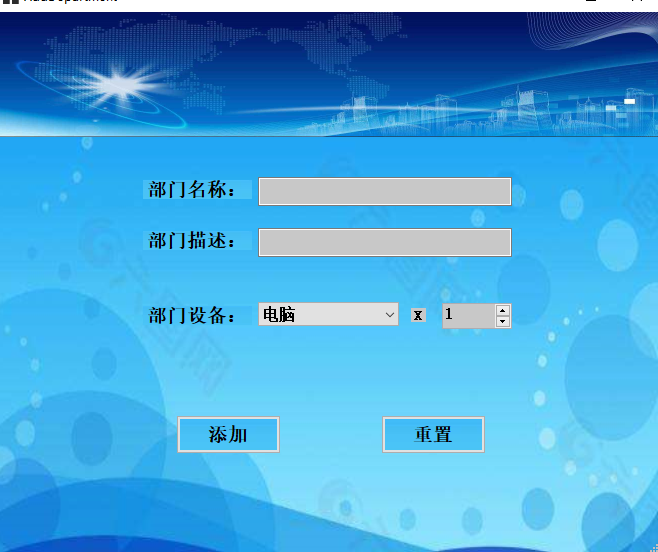
## 3.4、部门管理界面设计

部门管理页面设计两个下拉菜单分别为部门列表查看和添加部门两个选项。

部门列表页面设计图片



部门添加页面设计图片



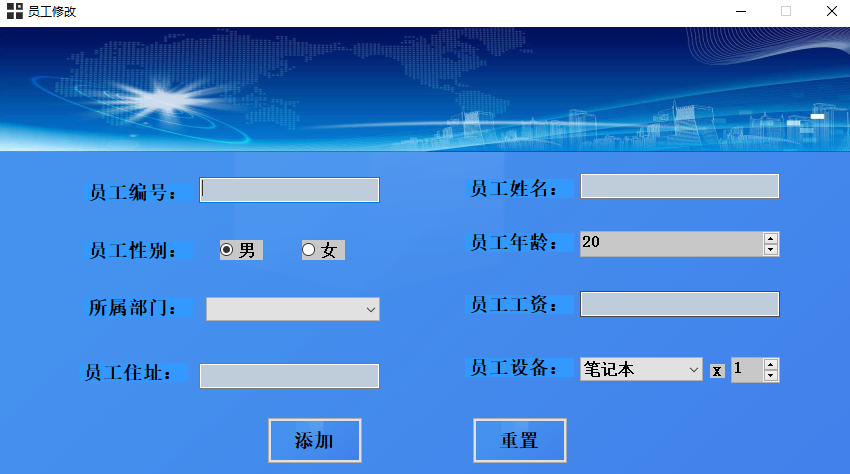
## 3.5、员工管理界面设计

员工管理页面为下拉菜单设计，下设有两个下拉菜单，分别为员工列表和员工添加两个选项。

员工列表页面设计图片



添加用户设计图片



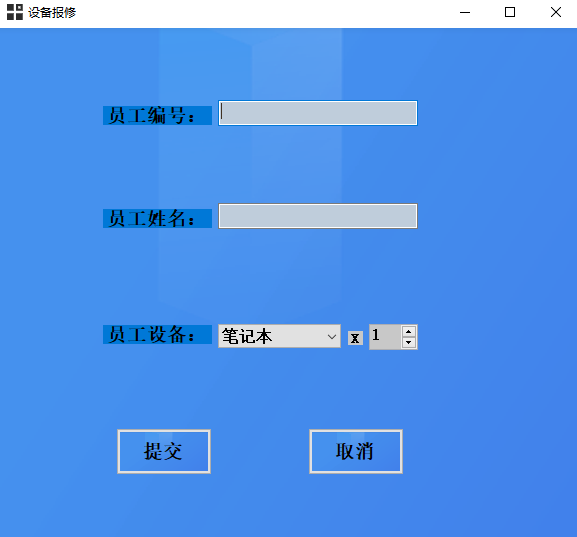
## 3.6、设备管理界面设计

设备管理页面下设三级下拉菜单，分别为设备列表、设备维修和设备添加。

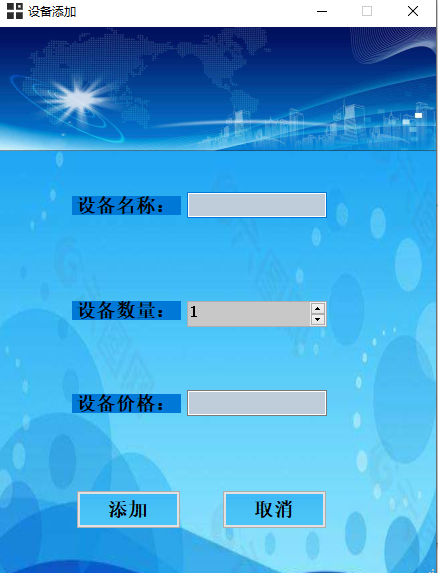
设备管理列表页面设计图



设备维修页面设计图



添加设备页面设计图



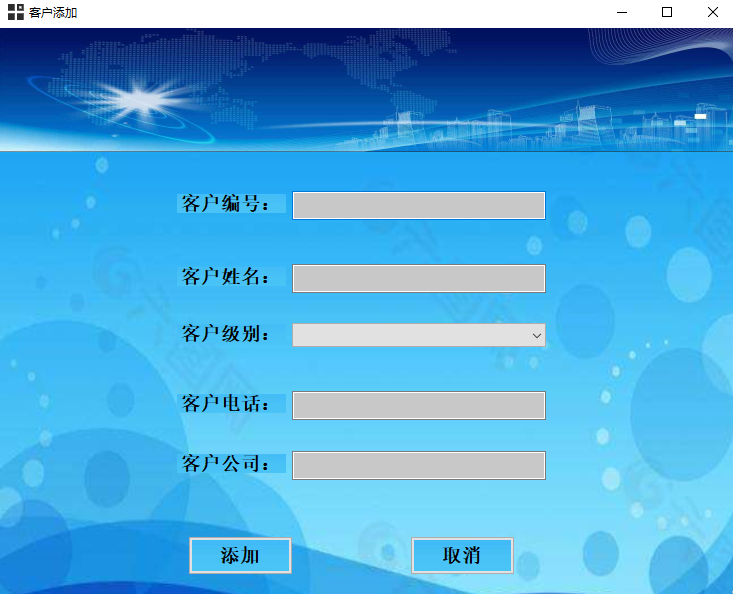
## 3.7、客户管理界面设计

客户管理菜单分为两级下拉菜单，分别为客户管理列表和客户添加。

客户管理列表页面设计图



客户添加页面设计图



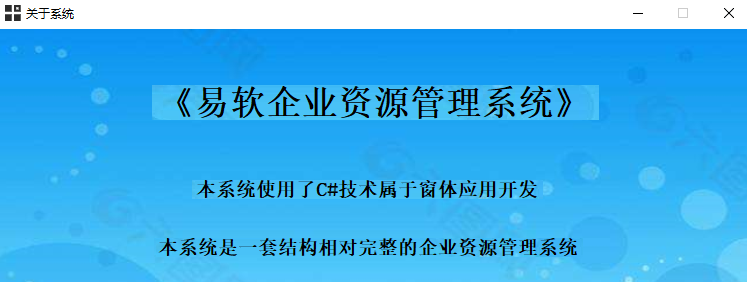
## 3.8、关于我们界面设计

最后一个模块为关于我们，关于我们菜单栏下设了三级菜单选项，分别为获取帮助、关于系统和关于开发者。

获取帮助页面设计



关于系统界面设计图



关于开发者界面设计图



## 3.9、快捷访问设计

菜单栏下边还设有快速连接选项，分别是部门列表、员工列表、设备列表、客户列表和退出系统。分别引用了已有的页面窗体链接。

# 4、系统详细设计

## 4.1、数据库连接

本次数据库设计采用Power Designer设计数据库模型，然后导出数据库到PHP Study\_Pro数据库集成环境中，使用navicat测试数据库可用性。

在Visual Studio中新建数据库连接类，将数据库的连接、打开、关闭等操作封装到一个类中方便以后操作。

DbUtils.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using MySql.Data.MySqlClient;

namespace DeviceManagers

{

class DbUtils

{

private static string dbUrl = @"

Database=device;DataSource=127.0.0.1;UserId=root;Password=root;pooling=false;CharSet=utf8;port=3306";

private MySqlConnection connection;

// 连接数据库

public MySqlConnection Connection

{

get

{

if (connection == null)

{

connection = new MySqlConnection(dbUrl);

}

return connection;

}

}

// 打开数据库

public void OpenConnection()

{

string s = Connection.State.ToString();

if (Connection.State == System.Data.ConnectionState.Closed)

{

connection.Open();

}

else if (Connection.State == System.Data.ConnectionState.Broken)

{

connection.Close();

connection.Open();

}

}

//关闭数据库

public void CloseConnection()

{

if (Connection.State == System.Data.ConnectionState.Closed || Connection.State == System.Data.ConnectionState.Broken || Connection.State == System.Data.ConnectionState.Open)

{

connection.Close();

}

}

}

}

## 4.2、登录功能实现

登录窗口代码获取用户页面输入的信息，首先进行校验，判断用户输入是否为空，如果为空则返回提示信息重新输入并登录，通过代码连接数据库查询用户名和密码的正确性，如果验证查询成功则会进入主页面。

LoginForm.cs页面

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace DeviceManagers

{

public partial class UserLogin : Form

{

#region 引入变量

public User user = new User();

private DbUtils dbUtil = new DbUtils();

#endregion

#region 构造函数

public UserLogin()

{

InitializeComponent();

}

#endregion

#region 事件

#region 重置事件

private void button\_reset\_Click(object sender, EventArgs e)

{

SetFormNull();

}

#endregion

#region 登录事件

private void button\_login\_Click(object sender, EventArgs e)

{

//验证登录

//数据验证

if (CheckInput())

{

Login();

}

}

#endregion

#endregion

#region 方法

#region 重置方法

private void SetFormNull()

{

text\_username.Text = "";

text\_password.Text = "";

}

#endregion

#region 验证表单

private bool CheckInput()

{

if (text\_username.Text.Trim().Length == 0)

{

MessageBox.Show("用户名不为空", "登录提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

return false;

}

if (text\_password.Text.Trim().Length == 0)

{

MessageBox.Show("密码不为空", "登录提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

return false;

}

return true;

}

#endregion

#region 登录方法

private void Login()

{

try

{

//创建sql查询

string sql = string.Format("select \* from admin\_user where username='{0}' and password='{1}'", text\_username.Text.Trim(), text\_password.Text.Trim());

//执行数据库连接

MySqlCommand com = new MySqlCommand(sql, dbUtil.Connection);

//打开数据库

dbUtil.OpenConnection();

//开始查询

MySqlDataReader reader = com.ExecuteReader();

//开始验证

if (reader.Read())

{

//记录用户信息

user.Uid = Convert.ToInt32(reader[0]);

user.Username = reader[1].ToString();

user.Password = reader[2].ToString();

//登录成功

MainForm mainform = new MainForm();

mainform.user = user;

mainform.Show();

this.Hide();

}

else

{

MessageBox.Show("用户名或密码不存在", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据库操作异常", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

dbUtil.CloseConnection();

}

}

#endregion

#endregion

}

}

## 4.3、主窗口界面实现

用户登录后进入主窗口界面，主窗口界面主要通过精美的UI设计和各个窗口事件的调用实现系统整体的页面调度。

MainForm.cs页面

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DeviceManagers

{

public partial class MainForm : Form

{

#region 变量

public User user;

#endregion

#region 构造函数

public MainForm()

{

InitializeComponent();

}

#endregion

#region 事件

#region 退出系统

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

{

DialogResult dia = MessageBox.Show("您确定退出系统嘛？", "系统提示", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);

//返回值为OK则退出系统

if (dia == DialogResult.OK)

{

Application.Exit();

}

}

#endregion

#region 修改密码

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

{

viewForm.UpdatePassword updatepass = new viewForm.UpdatePassword();

//传递参数

updatepass.user = this.user;

//显示修改窗口

updatepass.ShowDialog();

}

#endregion

#region 新增用户

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

{

viewForm.AddUser adduser = new viewForm.AddUser();

//显示窗口

adduser.ShowDialog();

}

#endregion

#region 查看员工列表

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

{

viewForm.StaffManagement staffManagement = new viewForm.StaffManagement();

staffManagement.Show();

}

#endregion

#region 添加员工

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

{

viewForm.AddStaff addStaff = new viewForm.AddStaff();

//0表示添加

addStaff.commodityID = "0";

addStaff.ShowDialog();

}

#endregion

#region 部门列表

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

{

viewForm.DepartmentManagement departmentManagement = new viewForm.DepartmentManagement();

departmentManagement.ShowDialog();

}

#endregion

#region 部门添加事件

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

{

viewForm.AddDepartment addDepartment = new viewForm.AddDepartment();

addDepartment.ShowDialog();

}

#endregion

#region 设备加载事件

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

{

viewForm.DeviceManagerment deviceManagerment = new viewForm.DeviceManagerment();

deviceManagerment.ShowDialog();

}

#endregion

#region 设备保修事件

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

{

viewForm.DeviceRepair deviceRepair = new viewForm.DeviceRepair();

deviceRepair.ShowDialog();

}

#endregion

#region 添加设备事件

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

{

viewForm.AddDevice addDevice = new viewForm.AddDevice();

addDevice.ShowDialog();

}

#endregion

#region 获取帮助事件

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

{

viewForm.Help help = new viewForm.Help();

help.ShowDialog();

}

#endregion

#region 关于开发者事件

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

{

viewForm.AboutDeveloper aboutDeveloper = new viewForm.AboutDeveloper();

aboutDeveloper.ShowDialog();

}

#endregion

#region 关于系统

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

{

viewForm.AboutSys aboutSys = new viewForm.AboutSys();

aboutSys.ShowDialog();

}

#endregion

#region 部门列表

private void toolStripButton1\_Click(object sender, EventArgs e)

{

viewForm.DepartmentManagement departmentManagement = new viewForm.DepartmentManagement();

departmentManagement.ShowDialog();

}

#endregion

#region 员工列表

private void toolStripButton2\_Click(object sender, EventArgs e)

{

viewForm.StaffManagement staffManagement = new viewForm.StaffManagement();

staffManagement.ShowDialog();

}

#endregion

#region 设备列表

private void toolStripButton3\_Click(object sender, EventArgs e)

{

viewForm.DeviceManagerment deviceManagerment = new viewForm.DeviceManagerment();

deviceManagerment.ShowDialog();

}

#endregion

#region 退出系统

private void toolStripButton4\_Click(object sender, EventArgs e)

{

DialogResult dr = MessageBox.Show("您是否要退出系统" , "系统提示", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if(dr == DialogResult.Yes)

{

this.Close();

Application.Exit();

}

}

#endregion

#region 客户列表

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

{

viewForm.ClientManagement clientManagement = new viewForm.ClientManagement();

clientManagement.ShowDialog();

}

#endregion

#region 添加客户事件

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

{

viewForm.AddClient addClient = new viewForm.AddClient();

addClient.ShowDialog();

}

#endregion

#region 客户管理事件

private void toolStripButton5\_Click(object sender, EventArgs e)

{

viewForm.ClientManagement clientManagement = new viewForm.ClientManagement();

clientManagement.ShowDialog();

}

#endregion

#endregion

}

}

## 4.4、用户操作功能实现

修改用户密码主要代码实现

#region 修改密码

private void UpdatePass()

{

try

{

//创建数据库语句

string sql = string.Format(@"update admin\_user set password='{0}' where id='{1}'", textBox\_newPassword1.Text.Trim(), user.Uid);

//执行数据库连接

MySqlCommand comm = new MySqlCommand(sql, dbUtil.Connection);

//打开数据库

dbUtil.OpenConnection();

//开始写入数据

int result = comm.ExecuteNonQuery();

//结果提示

if(result == 1)

{

MessageBox.Show("修改成功", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Close();

}

else

{

MessageBox.Show("修改失败", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据库操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

dbUtil.CloseConnection();

}

}

#endregion

* + 1. 新增用户主要代码实现

#region 增加用户方法

private void AddAdmin()

{

try

{

//创建数据库语句

string sql = string.Format(@"insert into admin\_user(username,password) values('{0}','{1}')", textBox\_adduser.Text.Trim(), textBox\_addpassword1.Text.Trim());

//执行数据库连接

MySqlCommand comm = new MySqlCommand(sql, dbUtil.Connection);

//打开数据库

dbUtil.OpenConnection();

//开始写入数据

int result = comm.ExecuteNonQuery();

//结果提示

if (result == 1)

{

MessageBox.Show("添加成功", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Close();

}

else

{

MessageBox.Show("添加失败", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据库操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

dbUtil.CloseConnection();

}

}

#endregion

* + 1. 部门管理列表主要代码实现

#region 初始化部门信息列表事件

private void SetCombox\_department()

{

try

{

//ds = new DataSet();

ds = new DataSet();

//创建sql查询

StringBuilder sb = new StringBuilder();

sb.AppendFormat(@"select d.did as '部门编号', d.dname as '部门名称', d.ddescription as '部门表述',e.ename as '部门设备',d.ddevicenums as '设备数量' from d\_departments as d,e\_device as e where e.eid = d.ddevice");

//赋值

this.dataGridView\_departmentList.DataSource = ds.Tables["Comminty"]; // 绑定数据源

//搜索条件 按照员工id排序

sb.AppendFormat(" order by d.did asc");

try

{

//执行数据库连接

MySqlDataAdapter adapter = new MySqlDataAdapter(sb.ToString(), dbUtil.Connection);

//开始填充表格

adapter.Fill(ds, "Comminty");

this.dataGridView\_departmentList.DataSource = this.ds.Tables["Comminty"];

}

catch (Exception)

{

MessageBox.Show("数据库操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据库操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

#endregion

#region 查找部门

private void FindDepartment()

{

try

{

ds = new DataSet();

//创建sql查询

StringBuilder sb = new StringBuilder();

sb.AppendFormat(@"select d.did as '部门编号', d.dname as '部门名称', d.ddescription as '部门表述',e.ename as '部门设备',d.ddevicenums as '设备数量' from d\_departments as d,e\_device as e where e.eid = d.ddevice");

if(textBox\_findDepartment.Text.Trim() != "")

{

sb.AppendFormat(" and d.dname like '%{0}%'", textBox\_findDepartment.Text.Trim());

}

//赋值

this.dataGridView\_departmentList.DataSource = ds.Tables["Comminty"]; // 绑定数据源

//搜索条件 按照员工id排序

sb.AppendFormat(" order by d.did asc");

try

{

//执行数据库连接

MySqlDataAdapter adapter = new MySqlDataAdapter(sb.ToString(), dbUtil.Connection);

if(adapter.ToString() != "")

{

//开始填充表格

adapter.Fill(ds, "Comminty");

this.dataGridView\_departmentList.DataSource = this.ds.Tables["Comminty"];

}

else

{

MessageBox.Show("没有这部门", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据库操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据库操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

#endregion

#region 删除部门

private void DeleteDepartById()

{

//判定是否选中了数据

//给出提示是否要删除

if(this.dataGridView\_departmentList.CurrentRow != null)

{

DialogResult dr = MessageBox.Show("确定要删除部门为："+ dataGridView\_departmentList.CurrentRow.Cells[1].Value, "系统提示", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if(dr == DialogResult.Yes)

{

try

{

//创建sql语句

string sql = string.Format("delete from d\_departments where did ={0}",dataGridView\_departmentList.CurrentRow.Cells[0].Value);

//执行数据库连接

MySqlCommand comm = new MySqlCommand(sql,dbUtil.Connection);

//打开数据库连接

dbUtil.OpenConnection();

//开始执行

int result = comm.ExecuteNonQuery();

if (result == 1){}

else

{

MessageBox.Show("操作失败，请重试", "操作提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

dbUtil.CloseConnection();

}

try

{

//创建sql语句

string sql = string.Format("delete from d\_departments\_and\_e\_device where did ={0}", dataGridView\_departmentList.CurrentRow.Cells[0].Value);

//执行数据库连接

MySqlCommand comm = new MySqlCommand(sql, dbUtil.Connection);

//打开数据库连接

dbUtil.OpenConnection();

//开始执行

int result = comm.ExecuteNonQuery();

if (result == 1){}

else

{

MessageBox.Show("操作失败，请重试", "操作提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

dbUtil.CloseConnection();

}

try

{

//创建sql语句

string sql = string.Format("delete from s\_staff where sdepartment ={0}", dataGridView\_departmentList.CurrentRow.Cells[0].Value);

//执行数据库连接

MySqlCommand comm = new MySqlCommand(sql, dbUtil.Connection);

//打开数据库连接

dbUtil.OpenConnection();

//开始执行

int result = comm.ExecuteNonQuery();

if (result == 1)

{

MessageBox.Show("删除成功", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.SetCombox\_department();

}

else

{

MessageBox.Show("操作成功", "操作提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

dbUtil.CloseConnection();

}

}

}

}

#endregion

## 4.5、添加员工实现

#region 添加员工

private void AddSta()

{

string sex = "";

if (radioButton\_staffBoy.Checked)

{

sex = "男";

}

else if (radioButton\_staffGirl.Checked)

{

sex = "女";

}

//int dep = FindDepartIndex(comboBox\_staffdepartment.Text.Trim());

//int dev = FindDeviceIndex(comboBox\_staffDevice.Text.Trim());

try

{

//创建sql语句

string sql = string.Format("insert into s\_staff values(null,'{0}','{1}','{2}',{3},'{4}',{5},{6},{7},{8})",

textBox\_staffNum.Text.Trim(),

textBox\_staffName.Text.Trim(),

sex,

Convert.ToInt32(numericUpDown\_staffAge.Text.Trim()),

textBox\_staffAddress.Text.Trim(),

Convert.ToInt32(textBox\_staffMoney.Text.Trim()),

FindDepartIndex(comboBox\_staffdepartment.Text.Trim()),

FindDeviceIndex(comboBox\_staffDevice.Text.Trim()),

Convert.ToInt32(numericUpDown\_nums.Text.Trim())

);

//执行数据库连接

MySqlCommand comm = new MySqlCommand(sql, dbUtil.Connection);

//打开数据库连接

dbUtil.OpenConnection();

//开始执行

int result = comm.ExecuteNonQuery();

if (result == 1)

{

//MessageBox.Show("添加成功", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

//this.SetNull();

}

else

{

MessageBox.Show("添加失败", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据库操作错误", "错误提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

dbUtil.CloseConnection();

}

try

{

//创建sql语句

string sql = string.Format("insert into e\_device\_and\_s\_staff values(null,'{0}',{1},{2})",

textBox\_staffNum.Text.Trim(),

FindDeviceIndex(comboBox\_staffDevice.Text.Trim()),

Convert.ToInt32(numericUpDown\_nums.Text.Trim())

);

//执行数据库连接

MySqlCommand comm = new MySqlCommand(sql, dbUtil.Connection);

//打开数据库连接

dbUtil.OpenConnection();

//开始执行

int result = comm.ExecuteNonQuery();

if (result == 1)

{

MessageBox.Show("添加成功", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.SetNull();

}

else

{

MessageBox.Show("添加失败", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据库操作错误", "错误提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

dbUtil.CloseConnection();

}

}

#endregion#region 查找部门

private void FindDepartment()

{

try

{

ds = new DataSet();

//创建sql查询

StringBuilder sb = new StringBuilder();

sb.AppendFormat(@"select d.did as '部门编号', d.dname as '部门名称', d.ddescription as '部门表述',e.ename as '部门设备',d.ddevicenums as '设备数量' from d\_departments as d,e\_device as e where e.eid = d.ddevice");

if(textBox\_findDepartment.Text.Trim() != "")

{

sb.AppendFormat(" and d.dname like '%{0}%'", textBox\_findDepartment.Text.Trim());

}

//赋值

this.dataGridView\_departmentList.DataSource = ds.Tables["Comminty"]; // 绑定数据源

//搜索条件 按照员工id排序

sb.AppendFormat(" order by d.did asc");

try

{

//执行数据库连接

MySqlDataAdapter adapter = new MySqlDataAdapter(sb.ToString(), dbUtil.Connection);

if(adapter.ToString() != "")

{

//开始填充表格

adapter.Fill(ds, "Comminty");

this.dataGridView\_departmentList.DataSource = this.ds.Tables["Comminty"];

}

else

{

MessageBox.Show("没有这部门", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据库操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据库操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

#endregion

#region 删除部门

private void DeleteDepartById()

{

//判定是否选中了数据

//给出提示是否要删除

if(this.dataGridView\_departmentList.CurrentRow != null)

{

DialogResult dr = MessageBox.Show("确定要删除部门为："+ dataGridView\_departmentList.CurrentRow.Cells[1].Value, "系统提示", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if(dr == DialogResult.Yes)

{

try

{

//创建sql语句

string sql = string.Format("delete from d\_departments where did ={0}",dataGridView\_departmentList.CurrentRow.Cells[0].Value);

//执行数据库连接

MySqlCommand comm = new MySqlCommand(sql,dbUtil.Connection);

//打开数据库连接

dbUtil.OpenConnection();

//开始执行

int result = comm.ExecuteNonQuery();

if (result == 1){}

else

{

MessageBox.Show("操作失败，请重试", "操作提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

dbUtil.CloseConnection();

}

try

{

//创建sql语句

string sql = string.Format("delete from d\_departments\_and\_e\_device where did ={0}", dataGridView\_departmentList.CurrentRow.Cells[0].Value);

//执行数据库连接

MySqlCommand comm = new MySqlCommand(sql, dbUtil.Connection);

//打开数据库连接

dbUtil.OpenConnection();

//开始执行

int result = comm.ExecuteNonQuery();

if (result == 1){}

else

{

MessageBox.Show("操作失败，请重试", "操作提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

dbUtil.CloseConnection();

}

try

{

//创建sql语句

string sql = string.Format("delete from s\_staff where sdepartment ={0}", dataGridView\_departmentList.CurrentRow.Cells[0].Value);

//执行数据库连接

MySqlCommand comm = new MySqlCommand(sql, dbUtil.Connection);

//打开数据库连接

dbUtil.OpenConnection();

//开始执行

int result = comm.ExecuteNonQuery();

if (result == 1)

{

MessageBox.Show("删除成功", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.SetCombox\_department();

}

else

{

MessageBox.Show("操作成功", "操作提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception)

{

MessageBox.Show("数据操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

dbUtil.CloseConnection();

}

}

}

}

#endregion

#endregion

## 4.6、修改客户信息实现

#region 点击修改客户事件

private void button\_modify\_Click(object sender, EventArgs e)

{

viewForm.AddClient addClient = new viewForm.AddClient();

//获取用户点击索引

addClient.commodityID = Convert.ToString(dataGridView\_clientList.CurrentRow.Cells[0].Value);

addClient.ShowDialog();

this.SetDataClient();

}

#endregion

#region 修改客户信息

private void UpdateCli()

{

try

{

//创建sql语句

StringBuilder sb = new StringBuilder();

sb.AppendFormat("update c\_client set ccard='{0}'", textBox\_clientNum.Text.Trim());

sb.AppendFormat(",cclass={0}", FindClass(comboBox\_clientClass.Text.Trim()));

sb.AppendFormat(",cname='{0}'", textBox\_clientName.Text.Trim());

sb.AppendFormat(",ctele='{0}'", textBox\_clientTele.Text.Trim());

sb.AppendFormat(",ccompany='{0}'", textBox\_clientCompany.Text.Trim());

sb.AppendFormat(" where ccard='{0}'", commodityID);

//执行数据库连接

MySqlCommand comm = new MySqlCommand(sb.ToString(), dbUtil.Connection);

//打开连接

dbUtil.OpenConnection();

//开始执行

int result = comm.ExecuteNonQuery();

if (result == 1)

{

MessageBox.Show("修改成功", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Close();

}

}

catch (Exception)

{

MessageBox.Show("数据库操作错误", "系统提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

finally

{

dbUtil.CloseConnection();

}

}

#endregion

# 5、系统测试

## 5.1、功能测试

本次网络协议程序设计开发中没完成一部分代码，都要通过Visual Studio编译测试功能的实现的实际情况，有没有影响其他组件的正常运行。遇到代码执行错误等情况，通过断点测试等方法找到错误所在，并成功解决Bug。因此易软企业资源管理系统是在不断的测试中完成的。完成系统整体设计后通过程序编译测试程序整体的可用性和功能完整情况。

主要测试系统的交互特性，给人友好的交互体验。遇到用户输入错误等情况，系统会弹出提示窗口。



## 5.2、系统主窗口界面

登入系统后即可看到欢迎页面，用户可以对系统内容进行操作。当点击退出系统的时候，系统弹出提示是否要退出系统，点击是方则退出系统。



## 5.3、增删改查功能实现

用户可以对系统的任何数据进行操作，主要包括部门、员工、设备和客户的增删改查。如：相对“公共部”进行修改，则点击修改。



系统会根据“公共部”的索引自动到数据库中查询其相关信息，并将其返回到部门修改页面的特定栏目中。那么我们将部门描述改为“这是一个公共部门”，点击确定。



点击确定后，窗口自动关闭，返回到部门管理页面，显示列表自动刷新，我们可以看到修改后的“公共部”信息。

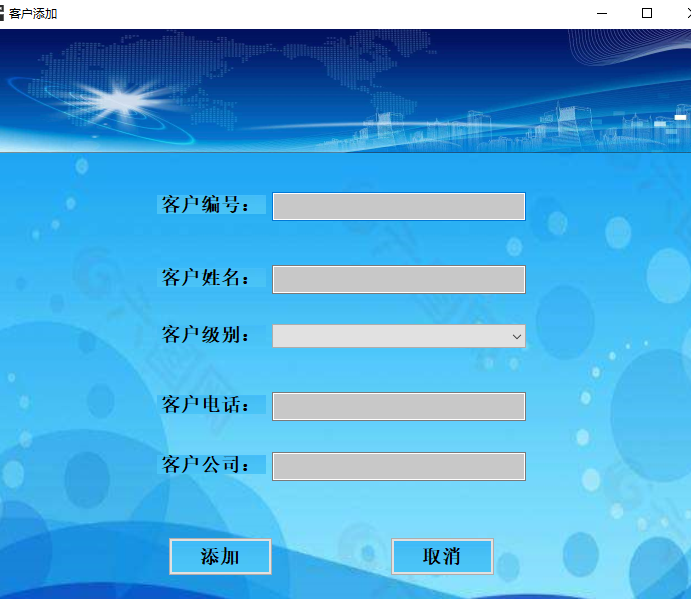


操作完成后，我们只需要点击右上角关闭窗口即可。

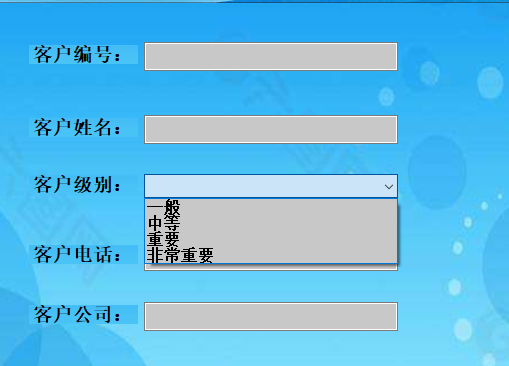
下面测试客户管理列表，客户信息的添加。



可以看到列表中已有了三条信息，那么测试客户信息的添加情况。



本次系统设计，将客户设置了特定的级别，作为必选项；在窗口调用时即加载到了页面，供用户选择。



例如我们填写如下信息，点击添加。



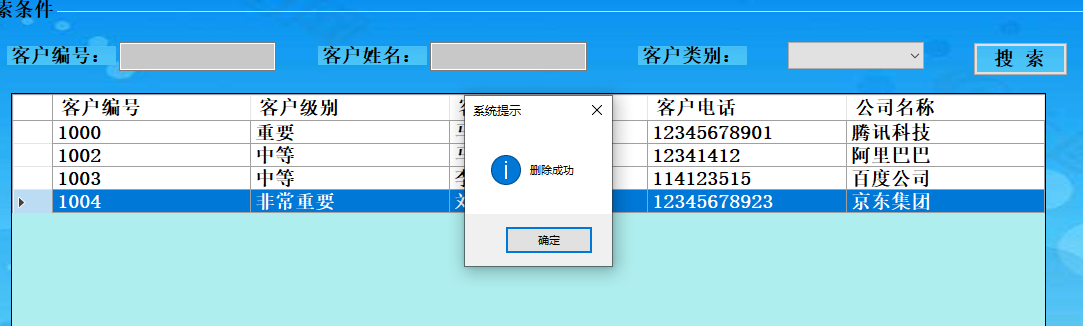
返回系统客户列表查看。



那么验证删除事件，直接选中刚刚添加的客户点击删除。



系统给出提示是否要删除，点击确定则执行删除事件。



系统客户列表刷新，重现加载客户列表。



按照此步骤对员工管理、设备管理等各个页面进行功能测试和交互测试。在不断的调试中系统得到了最终的完善。最终“易软企业资源管理系统V1.0.6“版本上线。

# 6、课程设计总结

此次网络协议分析课程的学习使我受益良多，深层的了解掌握了网络协议的独特魅力，了解了C#语言的程序设计和应用可发。从零开始紧跟老师步伐全方位的学习了网络协议课程的讲解和案例分析，充分理解了网络协议的原理和分析过程。在老师我同学的帮助下，我学习完成了本次网络协议分析课程，完成了易软企业资源管理系统的设计与开发，并做为课程设计检验自己的学习成果。

非常感谢老师能够给我这次机会，让我能够独立的完成一个网络协议系统的设计与开发的全过程，以此来结合所学知识对我之前学到的知识有了更深层次的感悟，对今后的学习与工作都有着及其重要的作用。