

# 目 录

1、项目分析.....	2
1.1、项目背景.....	2
1.2、需求分析.....	2
2、系统概要设计.....	2
2.1、系统设计思想.....	3
2.2、开发环境.....	3
2.3、总体结构设计.....	3
2.4、数据库设计.....	4
3、系统界面模块设计.....	5
3.1、登录界面设计.....	6
3.2、主窗口界面设计.....	6
3.3、系统设置界面设计.....	7
3.4、部门管理界面设计.....	8
3.5、员工管理界面设计.....	10
3.6、设备管理界面设计.....	10
3.7、客户管理界面设计.....	12
3.8、关于我们界面设计.....	13
3.9、快捷访问设计.....	15
4、系统详细设计.....	15
4.1、数据库连接.....	15
4.2、登录功能实现.....	16
4.3、主窗口界面实现.....	19
4.4、用户操作功能实现.....	24
4.5、添加员工实现.....	30
4.6、修改客户信息实现.....	36
5、系统测试.....	37
5.1、功能测试.....	37
5.2、系统主窗口界面.....	38
5.3、增删改查功能实现.....	38
6、课程设计总结.....	44

## 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 为代表的窗体应用深受广大用户的欢迎，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主要产品。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

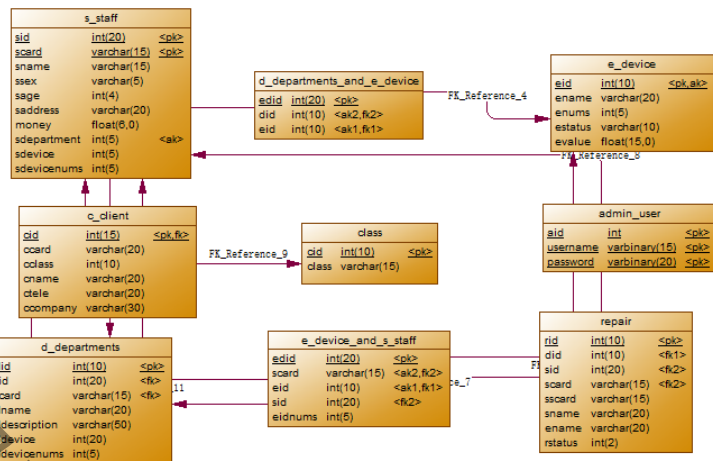
## 2.3、总体结构设计

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。

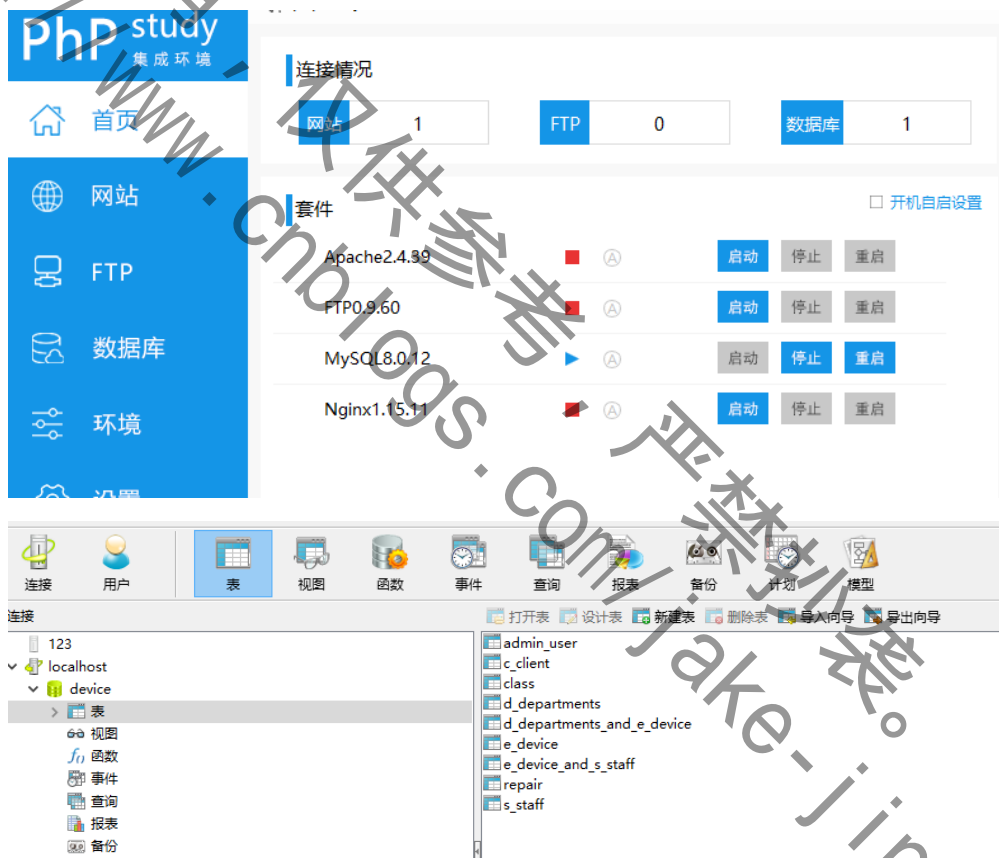
“易软企业资源管理系统”主要分为六个模块。其中包括：系统设置、部门管理、员工管理、设备管理、客户管理和关于我们等几个模块。通过复杂的代码逻辑将着几个模块联系在一起，在不影响功能的同时，充分优化的 UI 界面，力图达到最佳的交互体验。

使用 xmind 设计系统的整体框架图如下：





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

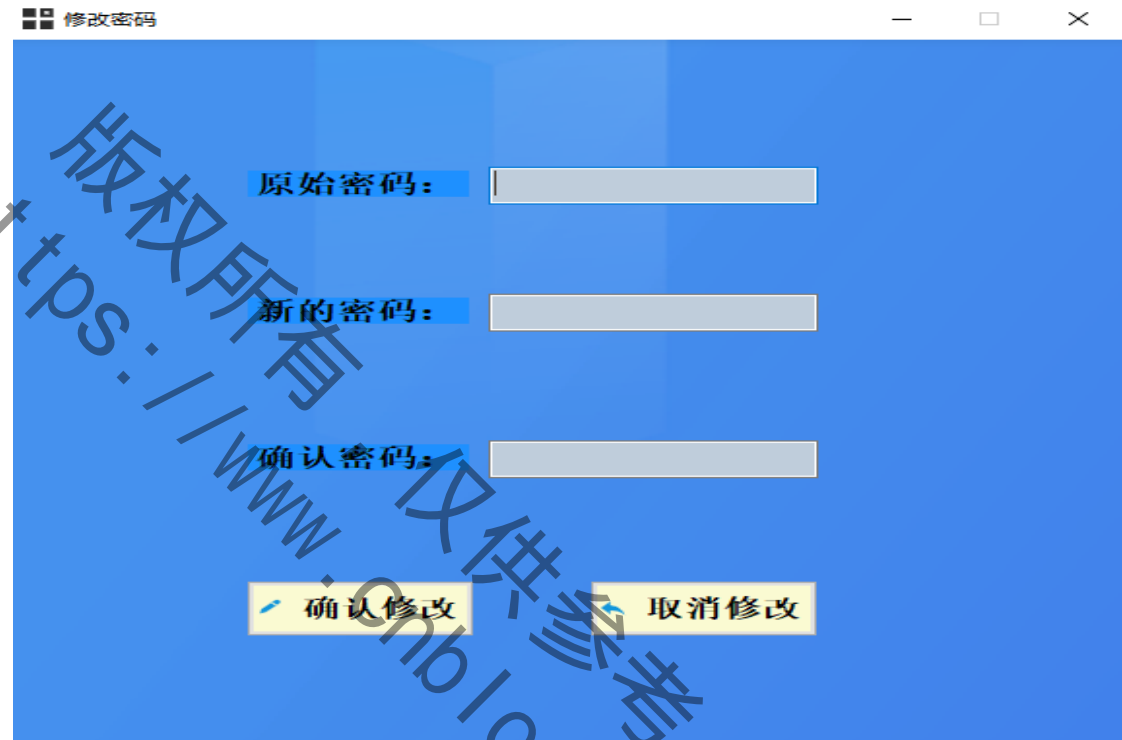
输入用户名和密码正确后就会进入主窗口程序，主窗口界面使用多个菜单栏和一个图片式的设计构成一个简约的欢迎页面。还有几个 logo 图标。



### 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



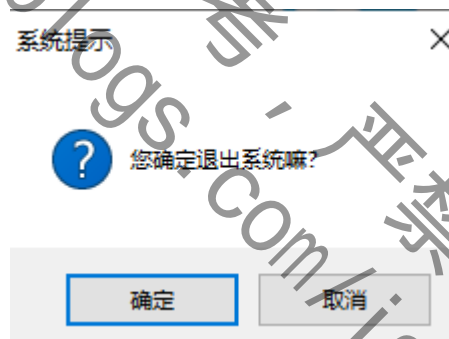
添加管理员用户 UI 设计图片

用户姓名:

用户密码:

确认密码:

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



部门管理

搜索条件

部门名称:

部门编号	部门名称	部门表述	部门设备	设备数量
1	财务部	这是个有钱的部门	笔记本	20
2	人事部	这是个用人的部门	笔记本	10
3	组织部	这是个用组织的部门	笔记本	12
4	公共部	请问	笔记本	1
5	事业部	请问	笔记本	1
6	科研部	这是一个搞事情的部门	笔记本	20
10	文娱部	这是一个娱乐的部门	笔记本	1

部门添加页面设计图片

部门名称:

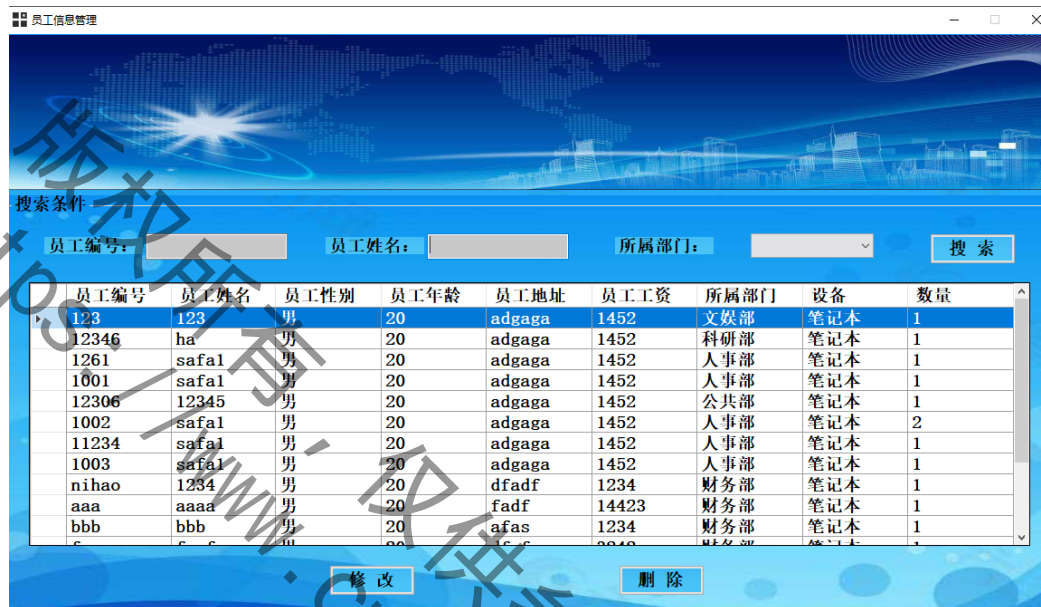
部门描述:

部门设备:

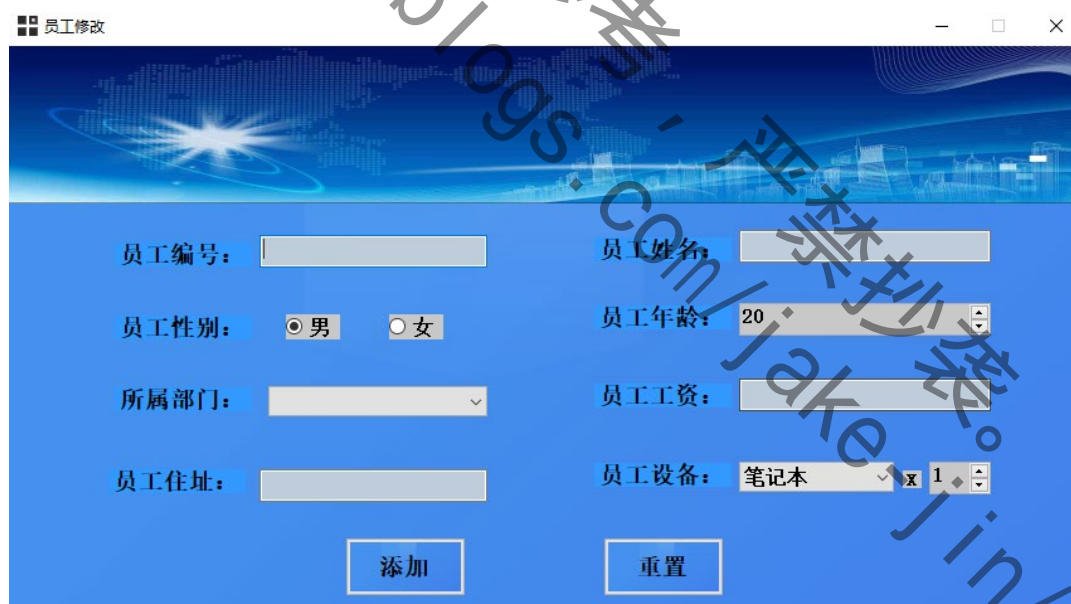
### 3.5、员工管理界面设计

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

设备管理列表

搜索条件

设备名称:

部门名称	设备名称	数量
文娱部	笔记本	1
财务部	笔记本	20
人事部	笔记本	10
组织部	笔记本	12
财务部	笔记本	20
财务部	服务器	20
财务部	服务器	20
设备部	一体机	10

设备维修页面设计图

设备报修

员工编号:

员工姓名:

员工设备: 笔记本  1

添加设备页面设计图



设备名称:

设备数量:

设备价格:

### 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



搜索条件

客户编号:  客户姓名:  客户类别:

客户编号	客户级别	客户姓名	客户电话	公司名称
1000	重要	马化腾	12341234432	腾讯科技
1002	中等	马云	12341412	阿里巴巴
1003	中等	李彦宏	114123515	百度公司

客户添加页面设计图



客户添加

客户编号:

客户姓名:

客户级别:

客户电话:

客户公司:

添加 取消

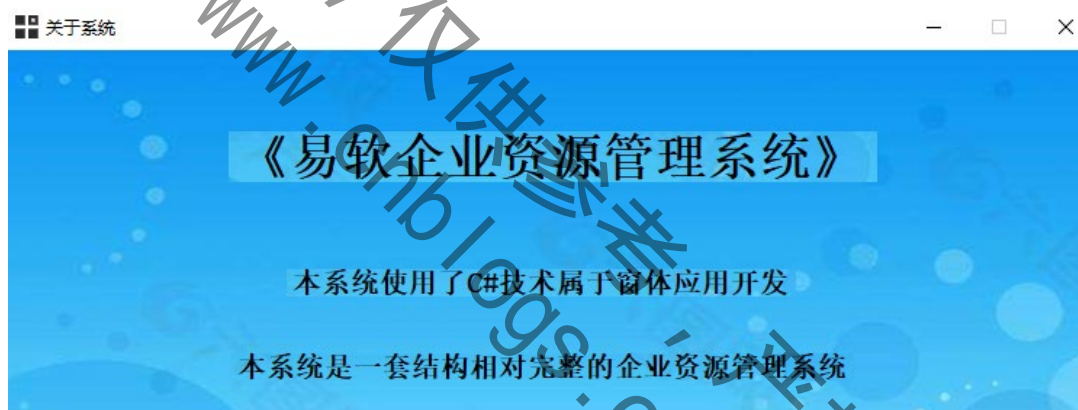
### 3.8、关于我们界面设计

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
    }
}

```

```

#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.ShowDialog();
}

#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.DeviceManagement deviceManagement = new viewForm.DeviceManagement();
    deviceManagement.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
```

- i. 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

```

ii. 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
    }
    catch (Exception)
    {
        MessageBox.Show(" 数据操作错误 ", " 系统提示 ", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
    }
}
#endregion

```

```

        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。

部门管理

搜索条件

部门名称:  搜索

部门编号	部门名称	部门表述	部门设备	设备数量
1	财务部	这是个有钱的部门	笔记本	20
2	人事部	这是个用人的部门	笔记本	10
3	组织部	这是个用组织的部门	笔记本	12
4	公共部	这是一个公共部门	笔记本	1
5	事业部	请问	笔记本	1
6	科研部	这是一个搞事情的部门	笔记本	20
10	文娱部	这是一个娱乐的部门	笔记本	1

修改 删除

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。

客户管理

搜索条件

客户编号:  客户姓名:  客户类别:  搜索

客户编号	客户级别	客户姓名	客户电话	公司名称
1000	重要	马化腾	12345678901	腾讯科技
1002	中等	马云	12341412	阿里巴巴
1003	中等	李彦宏	114123515	百度公司

修改 删除

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。



客户编号:

客户姓名:

客户级别:

客户电话:

客户公司:

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。

客户编号:

客户姓名:

客户级别:

客户电话:

客户公司:

一般  
中等  
重要  
非常重要

例如我们填写如下信息，点击添加。



客户编号: 1004

客户姓名: 刘强东

客户级别: 非常重要

客户电话: 12345678923

客户公司: 京东集团

添加 取消

系统提示

添加成功

确定

返回系统客户列表查看。



客户管理

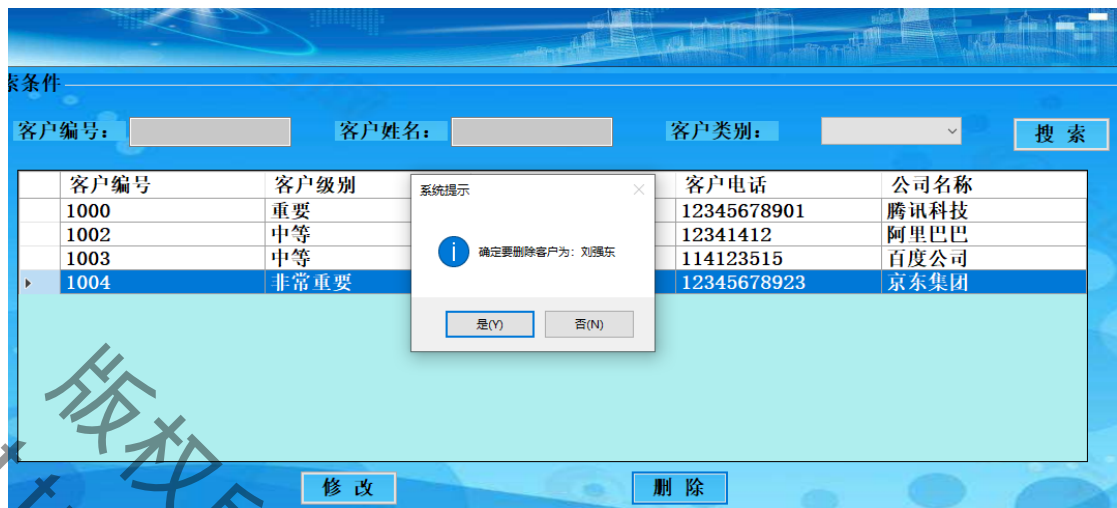
搜索条件

客户编号: 客户姓名: 客户类别: 搜索

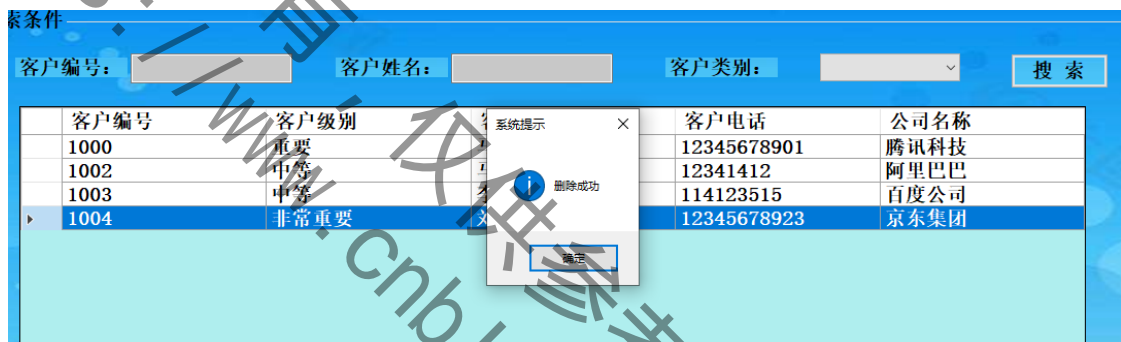
客户编号	客户级别	客户姓名	客户电话	公司名称
1000	重要	马化腾	12345678901	腾讯科技
1002	中等	马云	12341412	阿里巴巴
1003	中等	李彦宏	114123515	百度公司
1004	非常重要	刘强东	12345678923	京东集团

修改 删除

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开发。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。