

计算机科学与技术学院、大数据学院



课程：Web 开发技术 I

综合实验报告

题目：设备保管及查询系统

姓 名： 黎振龙

学 号： 201724112120

班 级： 17 网络工程 1 班

任课教师： 许盛贵

2020 年 6 月 26 日

1. 系统开发目的

本系统建立一个简易的 WEB 设备保管系统，通过该系统可以对员工、员工设备、部门进行统一的管理，根据不同角色拥有不同的系统管理权限。记录企业内所有设备的状况及保管人信息以提供企业内部了解设备的保管状况并可以进一步实施设备资源的共享，以避免不必要的重置购入相同的设备。同时更方便企业对人员进行管理，能快速、高效的实现企业的管理。

2. 系统需求

功能需求：

- (1) 将员工分为系统管理人及一般员工，所有人员都可以进行设备的保管。人员均需要编列于特定的部门中，如财务部和销售部。
- (2) 所有人员使用系统前需先登录，并按人员身份呈现不同的首页。
- (3) 只有系统管理人可以作 部门，设备，人员的增删查改操作；其他人员只能进行各项查询。
- (4) 一般查询：设备查询时能分别按以下条件查询：设备编号、设备名称、购入日期、存放位置、设备负责人姓名、部门名称。
- (5) 管理员查询：除一般查询外，还需要包含部门查询以及人员查询。

数据库设计要求：

- (1) 设备信息：设备编号 (PK)、设备名称、设备规格、设备图片 (1 张)、价格、购入日期、存放位置、设备负责人 (FK)。
- (2) 部门信息：部门代号 (PK)、部门名称、部门主管 (FK)。
- (3) 员工信息：人员编号 (PK)、密码、姓名、联系电话、是否为管理人。
- (4) 需要有 E-R 图设计，并合理设计上述数据形态及限制。
- (5) 数据库使用 VS2017 自带的 SQL Express。
- (6) 各项命名均使用英文。

其他需求：

- (1) 前端整合 Bootstrap 框架。
- (2) 界面简洁、美观。

3. 系统分析及设计

3.1 开发及运行环境

本系统是使用 VS2017 环境进行开发，数据库使用 VS2017 自带的数据库 LocalDB。运行环境为 Win10。本系统使用 .net MVC 进行开发，使用 PageList 分页工具进行分页管理，使用 Linq+EF 进行数据库操作。系统中使用各工具版本号如下：

EntityFramework.zh-Hans 版本：6.2.0
Microsoft.AspNet.Mvc 版本：5.2.7
Microsoft.AspNet.Razor 版本：3.2.7
分页工具 PageList.Mvc 版本：4.5.0
Newtonsoft.Json 版本：12.0.3

3.2 系统功能架构设计

设备管理系统的功能架构图如图 1：

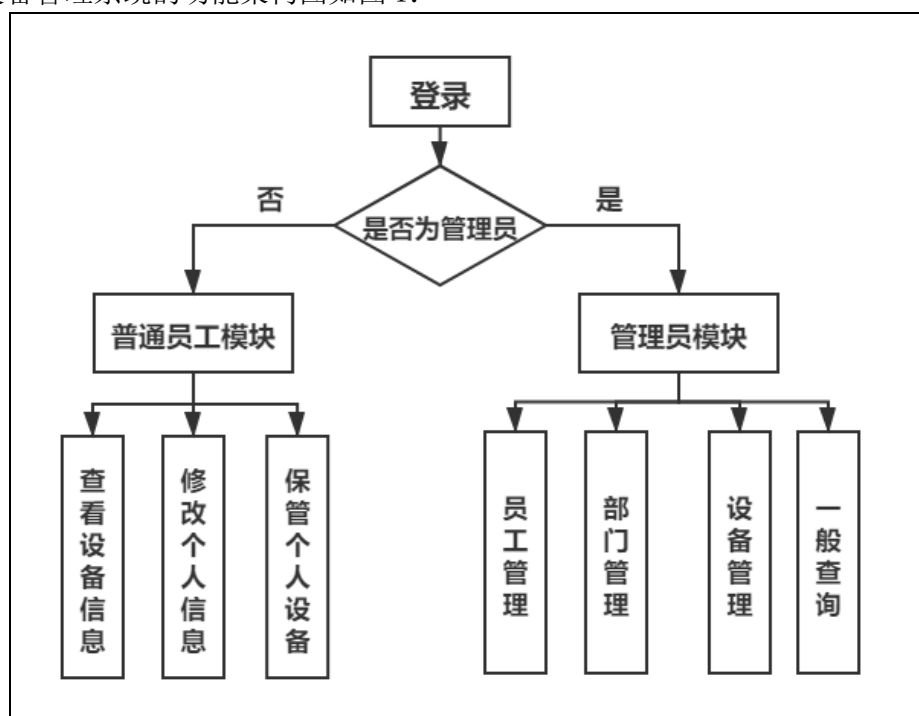


图 1 设备管理系统功能架构图

员工需要先进行系统登录才能操作系统，登录时，系统会判定当前登录人是一般员工还是系统管理员。如过是一般员工，就跳转到一般员工界面，一般员工登录后可以对设备进行查询（但不能进行增删改操作）、修改个人信息、保管个人设备等功能。如果是系统管理员，则跳转到系统管理员界面，系统管理员可以对员工、部门、设备进行管理（增删改查）操作。

3.3 数据库概念设计

本系统数据库概念设计（E-R 图）如图 2 所示

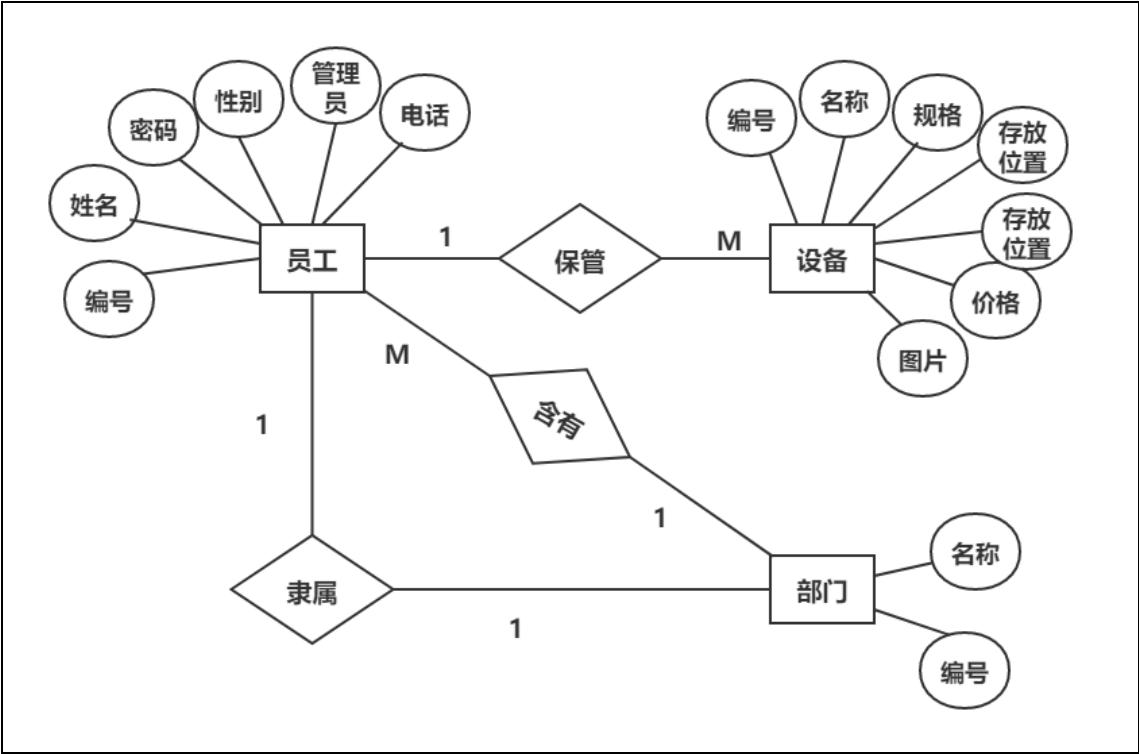


图 2 设备管理系统数据库概念设计

在概念设计中：一个员工隶属一个部门，而一个部门含有多个员工；同时一个员工可以保管多个设备。

3.4 数据库逻辑设计

员工信息逻辑设计如表 1 所示：

员工信息表（Employee）				
字段名	数据类型	长度	说明	注释
Id	int		不为空，主键	员工编号
name	nvarchar	50	不为空	员工姓名
password	nvarchar	50	不为空	密码
sex	nvarchar	50	可以为空	性别
phone	nchar	12	可以为空	电话号码
admin	bit		可以空，默认 0	是否为管理员
departmentId	int		可以为空，外键	部门编号

表 1 员工信息逻辑设计表

设备信息逻辑设计如表 2 所示：

设备信息表 (Employee)				
字段名	数据类型	长度	说明	注释
Id	int		不为空，主键	设备编号
name	nvarchar	50	不为空	设备名称
specs	nvarcahr	50	可以为空	设备规格
image	image		可以为空	设备图片
price	float		可以为空	设备价格
datetime	date		可以为空	录入日期
location	nvarchar	50	可以为空	存放位置
manager	int		可以为空，外键	管理人编号

表 2 设备信息逻辑设计表

部门信息逻辑设计如表 3 所示：

部门信息表 (Employee)				
字段名	数据类型	长度	说明	注释
Id	int		不为空，主键	部门编号
name	nvarchar	50	不为空	部门名称
admin	int		可以为空	管理员编号

表 3 部门信息逻辑设计表

4. 软件操作说明

4.1 一般员工之功能操作

进行操作之前先登录，输入非管理员账号：2020104，密码：root

请登录

账号

密码

登录成功，正在跳转




登录

登录完成后，进入一般用户首页（只能进行设备查询，不能进行设备及其他管理）

设备管理系统

欢迎你: 王五退出

请选择条件 请输入搜索信息 搜索

编号	设备名称	设备规格	设备图片	价格	录入日期	存放位置	设备管理员
20201011	联想电脑	300		6000	2020-01-01	销售部	黎振龙
20201012	打印机	5		5000	2020-02-03	开发部	黎振龙
20201013	显示器	300		4500	2020-01-02	销售部	无管理员

每页 3 条记录, 共 2 页, 当前第 1 页

1 2 »

一般用户可以对设备进行一般查询，选择编号进行搜索

编号

1011

搜索

	存放位置	设备管理员
--	------	-------

搜索结果:

设备管理系统

欢迎你: 王五退出

请选择条件 请输入搜索信息 搜索

编号	设备名称	设备规格	设备图片	价格	录入日期	存放位置	设备管理员
20201011	联想电脑	300		6000	2020-01-01	销售部	黎振龙

每页 10 条记录, 共 1 页, 当前第 1 页

1

选择设备名称进行查询:

设备名称

打印机

搜索

	存放位置	设备管理员
--	------	-------

搜索结果:

编号	设备名称	设备规格	设备图片	价格	录入日期	存放位置	设备管理员
20201012	打印机	5		5000	2020-02-03	开发部	黎振龙

每页 10 条记录, 共 1 页, 当前第 1 页

1

4.2 系统管理员之功能操作

使用管理员账号登录, 账号: 2020101, 密码: admin

登录成功后, 跳到管理员首页, 进行员工管理

员工管理

设备管理

部门管理

欢迎你: 黎振龙

退出

添加员工

请选择条件

请输入搜索信息

搜索

编号	姓名	密码	性别	电话	管理员	所属部门	操作
2020101	黎振龙	admin	男	13172619481	是	销售部	删除 修改
2020102	张三	admin	男	13289017812	是	开发部	删除 修改
2020104	王五	root	女	13112123455	否	行政部	删除 修改
2020106	张益达	123123	男	13172619481	否	销售部	删除 修改
2020111	周星星	12312313	男	123123	是	开发部	删除 修改
2020112	曹达华	123123	男	12312312312	否	行政部	删除 修改
2020113	周星驰	admin	男	19302981921	否	销售部	删除 修改
2020116	赵六	root	男	13128762109	否	行政部	删除 修改
2020117	张大张	123123	男	12381231	否	开发部	删除 修改

每页 10 条记录, 共 1 页, 当前第 1 页

1

在员工管理页面可以对员工进行增删查改操作。

演示添加员工操作, 点击添加员工按钮, 弹出输入员工信息模态框:

新增 信息

姓名

我是测试数据

密码

12345

性别

☒男 ☐女

电话

13172619481

管理员

☐是 ☒否

所属部门

重置

提交

执行添加操作，主页中增加一条数据

2020117	张大张	123123	男	12381231	否	开发部	删除	修改
2020118	我是测试数据	12345	男	13172619481	否	无所在部门	删除	修改

修改 2020118 员工，点击修改按钮，跳转到修改页面，输入修改信息，提交：

修改员工信息

修改

姓名:

我是测试数据2

密码

admin

性别

☐ 男 ☒ 女

电话

13172619481

设为管理员

☐ 是 ☒ 否

部门:

销售部

销售部

开发部

行政部

人事部

市场部

确认修改

返回

修改成功后，跳转到主页，显示修改后的数据：

2020117	张大张	123123	男	12381231	否	开发部	
2020118	我是测试数据2	admin	女	13172619481	否	销售部	

点击删除按钮，删除 2020118 号员工：

localhost:63036 显示

确认要删除这条数据吗？

确定

取消

确认后，直接删除：

2020117	张大张	123123	男	12381231	否	开发部	删除	修改
空								每页 10 条记录，共 1 页，当前第 1 页

1

员工管理功能演示完成。

设备管理功能演示

点击设备管理，跳转的设备管理页面：

<div> 添加 <div>请选择条件</div> <div>请输入搜索信息</div> 搜索 </div>								
编号	设备名称	设备规格	设备图片	价格	录入日期	存放位置	设备管理员	操作
20201011	联想电脑	300		6000	2020-01-01	销售部	黎振龙	删除 修改
20201012	打印机	5		5000	2020-02-03	开发部	黎振龙	删除 修改
20201013	显示器	300		4500	2020-01-02	销售部	无管理员	删除 修改

每页 3 条记录，共 2 页，当前第 1 页

[1](#)
[2](#)
[»](#)

点击添加按钮，在弹出的模态框中输入设备信息：

新增 信息

设备名称

JAVA程序设计

设备规格

200

设备图片



选择文件 java.jpg

价格

69

录入日期

2020/12/1

存放位置

开发部

设备管理员

请输入管理员编号

重置

提交

添加成功后，返回设备管理页面，看到新增信息

20201015	机械键盘	200		200	2000-01-01	广东	无管理员	删除 修改
20201016	JAVA程序设计	200		69	2020-12-01	开发部	无管理员	删除 修改

新增数据

点击修改按钮，输入修改信息就可以修改设备信息：

修改设备信息


修改	
设备名称:	<input type="text" value="JAVA程序设计"/>
设备规格	<input type="text" value="200"/>
设备价格	<input type="text" value="69"/>
录入时间	<input type="text" value="2020/12/01"/>
存放位置	<input type="text" value="开发部"/>
管理员	<input type="text"/>
<input type="button" value="确认修改"/> 返回	

修改基本信息

设备图片	
	<input type="button" value="选择文件"/> 未选择任何文件
	<input type="button" value="上传图片"/> 返回

修改设备图片

点击删除按钮，确认删除后即可删除设备信息：

20201015	机械键盘	200		200	2000-01-01	广东	无管理员	删除 修改
----------	------	-----	---	-----	------------	----	------	---------------------------------------

空

每页 3 条记录，共 2 页，当前第 2 页

1 2

按设备名称搜索设备：

设备名称

打印机

搜索

查询结果：

编号	设备名称	设备规格	设备图片	价格	录入日期	存放位置	设备管理员	操作
20201012	打印机	5		5000	2020-02-03	开发部	黎振龙	删除 修改

每页 10 条记录，共 1 页，当前第 1 页

1

设备管理功能演示完成。

点击部门管理，跳转到部门管理页面：

添加

请选择条件 请输入搜索信息 搜索

编号	部门名称	部门管理员	操作
101	销售部	黎振龙	删除 修改 查看所有员工
102	开发部	张三	删除 修改 查看所有员工
103	行政部	王五	删除 修改 查看所有员工
105	人事部	暂无部门管理员	删除 修改 查看所有员工
106	市场部	暂无部门管理员	删除 修改 查看所有员工

每页 10 条记录，共 1 页，当前第 1 页

1

可以查看某个部门的所有员工：

编号	姓名	密码	性别	电话	管理员	所属部门	操作
2020101	黎振龙	admin	男	13172619481	是	销售部	删除 修改
2020106	张益达	123123	男	13172619481	否	销售部	删除 修改
2020113	周星驰	admin	男	19302981921	否	销售部	删除 修改

添加

新增部门：

新增 信息

部门名称

测试部门

管理员

2020101

重置

提交

新增成功:

106	市场部	暂无部门管理员	<div>删除</div> <div>修改</div> <div>查看所有员工</div>
107	测试部门	黎振龙	<div>删除</div> <div>修改</div> <div>查看所有员工</div>

每页 10 条记录, 共 1 页, 当前第 1 页

修改部门, 输入部门信息进行修改:

修改部门信息

修改

部门名称:

测试部门

部门管理员:

2020101

确认修改

返回

按部门名字进行部门查询:

部门名称

开发

搜索

查询结果:

编号	部门名称	部门管理员	操作
102	开发部	张三	<div>删除</div> <div>修改</div> <div>查看所有员工</div>

每页 10 条记录, 共 1 页, 当前第 1 页

部门管理功能演示完成。

4.3 其他功能演示

验证登录功能: 未登录时, 不能直接在地址栏输入网址访问系统
复制网址, 退出当前登录状态:

localhost:63036/Employee/

地址栏输入，如果未登录，直接跳转道登录页面，不能进入系统：

请登录

账号

请输入账号

密码

请输入密码

登录

角色验证功能，一般员工不能进入系统管理员管理页面：

设备管理系统							
欢迎你：王五 退出							
请选择条件 请输入搜索信息 搜索							
编号	设备名称	设备规格	设备图片	价格	录入日期	存放位置	设备管理员
20201011	联想电脑	300		6000	2020-01-01	销售部	黎振龙
20201012	打印机	5		5000	2020-02-03	开发部	黎振龙

其他功能演示完成。

5. 重要程序源码说明

5.1 用户界面源码

登录页面: Login.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Login.aspx.cs"
Inherits="EMS201724112120.Login" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title>用户登录</title>
    <link href="lib/bootstrap/bootstrap.min.css" rel="stylesheet" />
    <script src="lib/bootstrap/jquery.min.js"></script>
    <script src="lib/bootstrap/bootstrap.min.js"></script>
    <style>
        .login {
            color: white;
            height: 38px;
            width: 300px;
            background-color: #2b669a;
        }
        #flag{
            margin:60px auto;
        }
    </style>
</head>
<body>
    <div class="container" id="flag">
        <h2 class="text-center">请登录</h2>
        <div class="row">
            <form class="form-horizontal col-md-offset-4 col-md-4" onsubmit="return
func()">
                <div class="form-group">
                    <label for="inputEmail3" class="col-sm-2 control-label">账号</label>
                    <div class="col-sm-10">
                        <input type="text" class="form-control" id="username" placeholder="
请输入账号">
                    </div>
                </div>
            </div>
            <div class="form-group">
```

```

        <label for="inputPassword3" class="col-sm-2 control-label" >密码
</label>

        <div class="col-sm-10">
            <input type="password" class="form-control" id="password"
placeholder="请输入密码">
            <span id="message" ></span>
        </div>
    </div>
    <div class="form-group">
        <div class="col-sm-offset-2 col-sm-10">
            <button type="submit" class="btn btn-primary" style="width:290px"
onclick="login()">登录</button>
        </div>
    </div>
</form>
</div>
</div>
</body>
</html>
<script type="text/javascript">

    function func() {
        return false;
    }

    function login() {
        var username = $("#username").val();
        var password = $("#password").val();
        var message = $("#message");
        console.log(username);
        console.log(password);
        console.log("点击了登录");
        if (username == "") {
            message.text("用户名不能为空");
        } else if (password == "") {
            message.text("密码不能为空");
        } else {
            message.text("正在验证, 请稍后...");
            $.ajax({
                url: "/Login/Index",
                type: "POST",
                dataType: "json",
                data: {
                    id: username,
                    pwd: password,

```

```

    },
    success: function (data) {
        console.log(data);
        message.text(data.message);
        //管理员登录
        if (data.code == 0) {
            window.location.href = data.url;
        }
        //非管理员登录
        if (data.code == 1) {
            window.location.href = data.url;
        } else {
            message.text(data.message);
        }
    },
    error: function (data) {
        console.log(data.message);
    }
});
}
}
</script>

```

设备首页视图, index.cshtml

```

@using EMS201724112120.Models;
@using Newtonsoft.Json;
@model PagedList.IPagedList < EMS201724112120.Models.Equipment>
@using PagedList.Mvc;
@using System.Globalization;
@{
    ViewBag.Title = "Index";
    Layout = null;
}
<html>
<head>
    <title>设备信息管理</title>
    <link href="~/lib/bootstrap/bootstrap.min.css" rel="stylesheet" />
    <script src="~/lib/bootstrap/jquery.min.js"></script>
    <script src="~/lib/bootstrap/bootstrap.min.js"></script>
    <style>
        #table1 th{
            text-align:center;
        }
    </style>

```



```

        #table1 td{
            line-height:150px;
            text-align:center;
        }
    </style>
</head>
<body>
    <!--用户界面导航-->
    <nav class="navbar navbar-default">
        <div class="container-fluid">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle collapsed"
data-toggle="collapse" data-target="#bs-example-navbar-collapse-1"
aria-expanded="false">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="/Employee/">员工管理</a>
                <a class="navbar-brand" href="/Equipment/">设备管理</a>
                <a class="navbar-brand" href="/Department/">部门管理</a>
            </div>

            <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
                <ul class="nav navbar-nav navbar-right">
                    <li><a href="#">欢迎你: @Session["name"]</a></li>
                    <li><a href="/Login/Logout">退出</a></li>
                </ul>
            </div>
        </div>
    </nav>
    <!--主题模块-->
    <div class="row" style="margin: 10px 10px 0 10px;">
        <div class="col-md-12">
            <button class="btn btn-primary" data-toggle="modal" data-target="#add">添
加</button>
            <form class="navbar-form navbar-right">
                <div class="form-group">
                    <select id="select" class="selectpicker">
                        <option id="0">请选择条件</option>
                        <option id="1">编号</option>
                        <option id="2">设备名称</option>
                    </select>

```

```

        <input type="text" id="selecttext" class="form-control"
placeholder="请输入搜索信息">
    </div>
    <button type="button" class="btn btn-primary" onclick="searchbytext()">
搜索</button>
</form>
<table class="table table-bordered table-hover table-striped" id="table1">
    <tr style="text-align:center">
        <th>编号</th>
        <th>设备名称</th>
        <th>设备规格</th>
        <th>设备图片</th>
        <th>价格</th>
        <th>录入日期</th>
        <th>存放位置</th>
        <th>设备管理员</th>
        <th>操作</th>
    </tr>
    @foreach (Equipment equipment in Model)
    {
        <tr>
            <td>@equipment.Id</td>
            <td>@equipment.name</td>
            <td>@equipment.specs</td>
            @if (equipment.image == null)
            {
                <td>暂无图片</td>
            }
            else
            {
                <td>
                    
                </td>
            }
            <td>@equipment.price</td>
            <td>
                @Convert.ToDateTime(@equipment.datatime).ToString("yyyy-MM-dd")
            </td>
            <td>@equipment.location</td>
            @if (@equipment.Employee == null)
            {

```



```

        <td><input type="text" name="name" id="name" placeholder="
请输入设备名称" /></td>
    </tr>
    <tr>
        <td>设备规格</td>
        <td><input type="text" name="specs" id="specs" placeholder="
请输入设备规格" /></td>
    </tr>
    <tr>
        <td>设备图片</td>
        <td>
            <!-- 预览 -->
            <img id="img1" src="" alt=""
style="width:200px;height:100px" /><br />
            <input type="file" name="filename" id="file1" />
        </td>
    </tr>
    <tr>
        <td>价格</td>
        <td><input type="text" name="price" id="price" placeholder="
请输入价格" /></td>
    </tr>
    <tr>
        <td>录入日期</td>
        <td><input type="text" name="datetime" id="datetime"
placeholder="请输入录入日期" /></td>
    </tr>
    <tr>
        <td>存放位置</td>
        <td><input type="text" name="location" id="location"
placeholder="请输入存放位置" /></td>
    </tr>
    <tr>
        <td>设备管理员</td>
        <td><input type="text" name="manager" id="manager"
placeholder="请输入管理员编号" /></td>
    </tr>
</table>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-default" onclick="reset()">重
置</button>

```

```

        <button type="button" class="btn btn-primary" onclick="add()">提交
</button>

        </div>
    </div>
    <!-- /.modal-content -->
</div>
<!-- /.modal -->
</div>
<!-- 模态框 (Modal) end -->

</body>

</html>
<script type="text/javascript">
    function searchbytext() {
        var keyword = $("#selecttext").val();           //获取搜索框关键字
        var selecttext = $("#select option:selected").attr("id");//获取查询条件
        if (keyword == null || keyword == "") {

        } else {
            window.location.href = "/Equipment/findBy?word=" + keyword + "&num=" +
selecttext;
        }
    }
    $("#file1").on("change", function () {
        var file = this.files[0];
        if (this.files && file) {
            var reader = new FileReader();
            reader.onload = function (e) {
                $("#img1").attr("src", e.target.result);
            }
            reader.readAsDataURL(file);
        }
    });

    function add() {
        images = $("#img1").attr("src");
        images = images.substring(images.indexOf(',') + 1);
        var equipment = {
            Id: 0,
            name: $("input[name='name']").val(),
            specs: $("input[name='specs']").val(),
            image:"",

```

```

        price: $("input[name='price']").val(),
        datetime: $("input[name='datetime']").val(),
        location: $("input[name='location']").val(),
        manager: $("input[name='manager']").val()

    }
    console.log(equipment);
    $.ajax({
        url: "/Equipment/Add/",
        type: "POST",
        dataType: "text",
        contentType: "application/json;charset=utf-8",
        data: JSON.stringify({ equipment: equipment, imageBase:images }),
        success: function (data) {
            console.log("添加成功");

            console.log(data);
            if (data == "success") {
                $('#add').modal('hide');
                window.location = "/Equipment/"
            }
        },
        error: function (data) {
            console.log("添加失败");
        }
    });
}

function del(id){
    console.log(id);
    if (confirm("确认要删除这条数据吗? ")) {
        window.location = "/Equipment/Delete/" + id;
    }
}

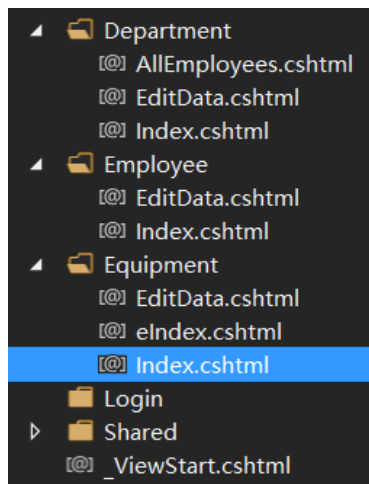
}

function Modify(id) {
    window.location = "/Equipment/EditData/" + id;
}

</script>

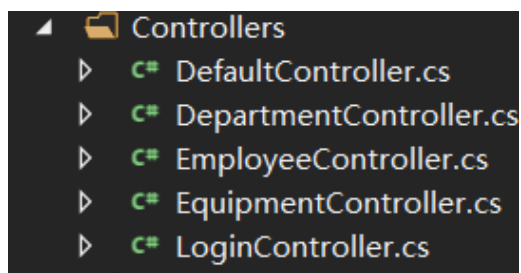
```

其他主页视图基本相似，篇幅原因不一一展示



5.2 各项功能源码

控制器设计如下：



DefaultController 为**登录验证控制器**，重写 `OnActionExecuting` 方法，进行登录 Session 验证，要进行登录验证的控制器都要继承该控制器，具体代码如下；

```
namespace EMS201724112120.Controllers
{
    public class DefaultController : Controller
    {
        //登录验证控制器
        protected override void OnActionExecuting(ActionExecutingContext
filterContext)
        {
            base.OnActionExecuting(filterContext);
            var controllerName =
filterContext.ActionDescriptor.ControllerDescriptor.ControllerName;

            var name = Session["name"] as String;
            if (String.IsNullOrEmpty(name))
            {
                filterContext.Result = RedirectToAction("../Login.aspx");
            }
        }
    }
}
```

```

    }
}
}
}

```

DepartmentController 用于进行**部门信息管理**（增删查改），具体代码如下：

```

namespace EMS201724112120.Controllers
{
    public class DepartmentController : DefaultController
    {
        private SystemEntities db = new SystemEntities();
        // GET: Department
        //查找所有部门
        public ActionResult Index(int? page)
        {
            if (check())
            {
                return RedirectToAction("eIndex", "Equipment");
            }
            var departments = from a in db.Departments select a;
            int pageNumber = page ?? 1;
            int pageSize = 10;
            departments = departments.OrderBy(a => a.Id);
            IPagedList<Department> pagelist =
departments.ToPagedList(pageNumber, pageSize);
            return View(pagelist);
        }
        //条件查询
        public ActionResult findBy(int? page, String word, int? num)
        {
            if (check())
            {
                return RedirectToAction("eIndex", "Equipment");
            }
            var departments = from s in db.Departments where (s.name.Contains(word))
select s;
            if (num == 1)
            {
                departments = from s in db.Departments where
(s.Id.ToString().Contains(word)) select s;
            }
            int pageNumber = page ?? 1;
            int pageSize = 10;
            departments = departments.OrderBy(a => a.Id);

```



```

        IPagedList<Department> pagedList = departments.ToPagedList(pageNumber,
pageSize);
        return View("Index", pagedList);
    }
    //增加一个部门
    public ActionResult Add(Department department)
    {
        if (check())
        {
            return RedirectToAction("eIndex", "Equipment");
        }
        try
        {
            db.Departments.Add(department);
            db.SaveChanges();
            String message = "success";
            return Content(message);
        }catch(Exception ex)
        {
            return Content("增加失败!" + ex.Message);
        }
    }
    //回显要修改的数据
    public ActionResult EditData(int id)
    {
        if (check())
        {
            return RedirectToAction("eIndex", "Equipment");
        }
        try
        {
            Department department = (from a in db.Departments where a.Id == id select
a).FirstOrDefault();
            return View(department);
        }catch (Exception ex)
        {
            return Content("回显数据失败" + ex.Message);
        }
    }
    //修改部门信息
    public ActionResult EditDataed(Department department)
    {
        if (check())
        {

```

```

        return RedirectToAction("eIndex", "Equipment");
    }
    try
    {
        DbEntityEntry<Department> entry = db.Entry<Department>(department);
        entry.State = EntityState.Unchanged;
        entry.Property(a => a.Id).IsModified = true;
        entry.Property(a => a.name).IsModified = true;
        entry.Property(a => a.admin).IsModified = true;
        db.SaveChanges();
        return RedirectToAction("Index", "Department");
    }catch(Exception ex)
    {
        return Content("修改失败" + ex.Message);
    }
}

//删除部门信息
public ActionResult Delete(int id)
{
    if (check())
    {
        return RedirectToAction("eIndex", "Equipment");
    }
    try
    {
        Department department = new Department() { Id = id };
        db.Departments.Attach(department);
        db.Departments.Remove(department);
        db.SaveChanges();
        return RedirectToAction("Index", "Department");
    }
    catch(Exception ex)
    {
        return Content("删除失败" + ex.Message);
    }
}

//获取当前部门下的所有员工
public ActionResult AllEmployees(int id)
{
    if (check())
    {
        return RedirectToAction("eIndex", "Equipment");
    }
}

```

```

        //根据id查找当前部门
        Department department = (from a in db.Departments where a.Id == id select
a).FirstOrDefault();
        //得到当前部门的所有员工集合
        ICollection<Employee> list = department.Employees;
        ViewData["datalist"] = list;
        return View();
    }
    public Boolean check()
    {
        var admin = Session["admin"] as String;
        return admin == "true" ? false : true;
    }
}

```

EmployeeController 用于**员工信息管理**（增删查改），具体代码如下：

```

namespace EMS201724112120.Controllers
{
    public class EmployeeController : DefaultController
    {
        private SystemEntities db = new SystemEntities();
        //主页进行查询员工操作
        public ActionResult Index(int? page)
        {
            if (check())
            {
                return RedirectToAction("eIndex", "Equipment");
            }
            var employees = from s in db.Employees select s;
            int pageNumber = page ?? 1;
            int pageSize = 10 ;
            employees = employees.OrderBy(a => a.Id);
            IPagedList<Employee> pagedList = employees.ToPagedList(pageNumber,
pageSize);

            return View(pagedList);
        }
        //条件查询
        public ActionResult findBy(int? page,String word,int? num)
        {
            if (check())
            {
                return RedirectToAction("eIndex", "Equipment");
            }
        }
    }
}

```

```

        var employees = from s in db.Employees where (s.name.Contains(word)) select
s ;

        if (num == 1)
        {
            employees = from s in db.Employees where (s.Id.ToString().Contains(word))
select s;
        }
        int pageNumber = page ?? 1;
        int pageSize = 10;
        employees = employees.OrderBy(a => a.Id);
        IPagedList<Employee> pagedList = employees.ToPagedList(pageNumber,
pageSize);
        return View("Index",pagedList);
    }
    //执行增加操作
    public ActionResult Add(Employee employee)
    {
        try
        {
            if (check())
            {
                return RedirectToAction("eIndex", "Equipment");
            }
            db.Employees.Add(employee);
            db.SaveChanges();
            String message = "success";
            return Content(message);
        }
        catch(Exception ex)
        {
            return Content("增加失败! " + ex.Message);
        }
    }

    //执行删除操作
    public ActionResult Delete(int id)
    {
        try
        {
            if (check())
            {
                return RedirectToAction("eIndex", "Equipment");
            }
        }
    }

```

中

```
Employee employee = new Employee() { Id = id };
db.Employees.Attach(employee); //通过主键实例化的对象添加到EF管理容器

db.Employees.Remove(employee); //将对象状态标记为删除状态
db.SaveChanges(); //操作更新到数据库
return RedirectToAction("Index", "Employee");
}
catch(Exception ex)
{
    return Content("删除失败! "+ex.Message);
}

}

//显示要修改的数据
public ActionResult EditData(int id)
{
    try
    {
        if (check())
        {
            return RedirectToAction("eIndex", "Equipment");
        }
        Employee employee = (from a in db.Employees where a.Id ==id select
a).FirstOrDefault();
        IEnumerable<SelectListItem> listItem = (from c in db.Departments select
c).ToList().Select(c => new SelectListItem { Value = c.Id.ToString(), Text = c.name });
        ViewBag.CateList = listItem;
        return View(employee);
    }catch(Exception ex)
    {
        return Content("回显修改信息失败" + ex.Message);
    }
}

//执行修改操作
public ActionResult EditDataed(Employee employee)
{
    try
    {
        if (check())
        {
            return RedirectToAction("eIndex", "Equipment");
        }
    }
}
```

```

        DbEntityEntry<Employee> entry = db.Entry<Employee>(employee);
        entry.State = EntityState.Unchanged;
        entry.Property(a => a.name).IsModified = true;
        entry.Property(a => a.password).IsModified = true;
        entry.Property(a => a.sex).IsModified = true;
        entry.Property(a => a.phone).IsModified = true;
        entry.Property(a => a.admin).IsModified = true;
        entry.Property(a => a.departmentId).IsModified = true;
        db.SaveChanges();
        return RedirectToAction("Index", "Employee");
    }
    catch(Exception ex)
    {
        return Content("修改失败" + ex.Message);
    }
}
public Boolean check()
{
    var admin = Session["admin"] as String;
    return admin=="true"?false:true;
}
}

```

EquipmentController 用于设备信息管理（增删查改），具体代码如下：

```

namespace EMS201724112120.Controllers
{
    public class EquipmentController : DefaultController
    {
        private SystemEntities db = new SystemEntities();
        // GET: Equipment
        //查询所有设备信息
        public ActionResult Index(int? page)
        {
            if (check())
            {
                return RedirectToAction("eIndex", "Equipment");
            }
            var equipments = from a in db.Equipments select a;
            int pageNumber = page ?? 1;
            int pageSize = 3;
            equipments = equipments.OrderBy(a => a.Id);
            IPagedList<Equipment> pagedList = equipments.ToPagedList(pageNumber,

```

```

pageSize);

        return View(pagedList);
    }
    //条件查询
    public ActionResult findBy(int? page, String word, int? num)
    {
        if (check())
        {
            return RedirectToAction("eIndex", "Equipment");
        }
        var equipments = from s in db.Equipments where (s.name.Contains(word)) select
s;

        if (num == 1)
        {
            equipments = from s in db.Equipments where
(s.Id.ToString()).Contains(word)) select s;
        }
        int pageNumber = page ?? 1;
        int pageSize = 10;
        equipments = equipments.OrderBy(a => a.Id);
        IPagedList<Equipment> pagedList = equipments.ToPagedList(pageNumber,
pageSize);
        return View("Index", pagedList);
    }

    //增加设备
    public ActionResult Add(Equipment equipment,String imageBase)
    {
        if (check())
        {
            return RedirectToAction("eIndex", "Equipment");
        }
        try
        {
            byte[] bytes = Convert.FromBase64String(imageBase);
            equipment.image = bytes;
            db.Equipments.Add(equipment);
            db.SaveChanges();
            String message = "success";
            return Content(message);
        }catch(Exception ex)
        {
            return Content("增加失败,错误信息: " + ex.Message);
        }
    }

```

```

    }
}
//删除设备
public ActionResult Delete(int id)
{
    if (check())
    {
        return RedirectToAction("eIndex", "Equipment");
    }
    try
    {
        Equipment equipment = new Equipment() { Id = id };
        db.Equipments.Attach(equipment);
        db.Equipments.Remove(equipment);
        db.SaveChanges();
        return RedirectToAction("Index", "Equipment");
    }
    catch (Exception ex)
    {
        return Content("删除失败,错误信息: " + ex.Message);
    }
}
//回显修改设备信息
public ActionResult EditData(int id)
{
    if (check())
    {
        return RedirectToAction("eIndex", "Equipment");
    }
    try
    {
        Equipment equipment = (from a in db.Equipments where a.Id == id select
a).FirstOrDefault();
        return View(equipment);
    }catch(Exception ex)
    {
        return Content("回显修改数据信息失败,错误信息: " + ex.Message);
    }
}
//执行修改操作
public ActionResult EditDataed(Equipment equipment)
{
    if (check())
    {

```



```

        return RedirectToAction("eIndex", "Equipment");
    }
    try
    {
        DbEntityEntry<Equipment> entry = db.Entry<Equipment>(equipment);
        entry.State = EntityState.Unchanged;
        entry.Property(a => a.name).IsModified = true;    //名字
        entry.Property(a => a.specs).IsModified = true;    //规格
        entry.Property(a => a.price).IsModified = true;    //价格
        entry.Property(a => a.datatime).IsModified = true; //录入时间
        entry.Property(a => a.location).IsModified = true; //产地
        entry.Property(a => a.manager).IsModified = true; //设备管理员
        db.SaveChanges();
        return RedirectToAction("Index", "Equipment");
    }
    catch (Exception ex)
    {
        return Content("修改数据失败,错误信息: " + ex.Message);
    }
}
//更新图片操作
public ActionResult updateImage(int id,String imageBase)
{
    if (check())
    {
        return RedirectToAction("eIndex", "Equipment");
    }
    try
    {
        byte[] bytes = Convert.FromBase64String(imageBase);
        Equipment equipments = db.Equipments.Find(id);
        db.Entry(equipments).State = EntityState.Unchanged;
        equipments.image = bytes;
        db.Entry(equipments).Property("image").IsModified = true;
        db.SaveChanges();
        String message = "success";
        return Content(message);
    }catch(Exception ex)
    {
        return Content("更新图片失败,错误信息: "+ex.Message);
    }
}
}

```

```

//非管理员主页
public ActionResult eIndex(int? page)
{
    var equipments = from a in db.Equipments select a;
    int pageNumber = page ?? 1;
    int pageSize = 3;
    equipments = equipments.OrderBy(a => a.Id);
    IPagedList<Equipment> pagedList = equipments.ToPagedList(pageNumber,
    pageSize);

    return View(pagedList);
}
//条件查询
public ActionResult findBye(int? page, String word, int? num)
{
    var equipments = from s in db.Equipments where (s.name.Contains(word)) select
    s;

    if (num == 1)
    {
        equipments = from s in db.Equipments where
        (s.Id.ToString().Contains(word)) select s;
    }
    int pageNumber = page ?? 1;
    int pageSize = 10;
    equipments = equipments.OrderBy(a => a.Id);
    IPagedList<Equipment> pagedList = equipments.ToPagedList(pageNumber,
    pageSize);
    return View("eIndex", pagedList);
}

public Boolean check()
{
    var admin = Session["admin"] as String;
    return admin == "true" ? false : true;
}
}

```

LoginController 用于用户登录验证和角色验证及用户登出，具体代码如下：

```

namespace EMS201724112120.Controllers
{
    //登录验证控制器
    public class LoginController : Controller
    {
        private SystemEntities db = new SystemEntities();
    }
}

```

```

// GET: Login
//登录验证及转发
public ActionResult Index(int id,String pwd)
{
    Employee employee = (from a in db.Employees where a.Id == id select
a).FirstOrDefault();

    if (employee == null)
    {
        var result = new { url = "", message = "用户不存在, 请重新输入", code = 202 };
        return Json(result);
    }
    if (employee.password != pwd)
    {
        var result = new { url = "", message = "密码错误, 请重新输入", code = 202 };
        return Json(result);
    }
    if (employee.admin==true)
    {
        Session["name"] = employee.name;
        Session["admin"] = "true";
        var result = new { url = "/Employee", message = "登录成功, 正在跳转
" ,code=0};
        return Json(result);
    }
    else
    {
        Session["name"] = employee.name;
        Session["admin"] = "false";
        var result = new { url = "/Equipment/eIndex", message = "登录成功, 正在
跳转", code = 1 };
        return Json(result);
    }
}
//登出
public ActionResult Logout()
{
    Session["name"] = "";
    Session["admin"] = "";
    return RedirectToAction("../Login.aspx");
}
}

```

5.3 数据库源码（数据库创建及定义 script）

员工信息表：

```
CREATE TABLE [dbo].[Employee] (  
    [Id]          INT          IDENTITY (1, 1) NOT NULL,  
    [password]    NVARCHAR (50) NOT NULL,  
    [name]        NVARCHAR (50) NOT NULL,  
    [sex]         NVARCHAR (10) NULL,  
    [phone]       NCHAR (12)   NULL,  
    [admin]       BIT          NULL,  
    [departmentId] INT          NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC),  
    FOREIGN KEY ([departmentId]) REFERENCES [dbo].[Department] ([Id]) ON  
DELETE SET NULL  
);
```

设备信息表：

```
CREATE TABLE [dbo].[Equipment] (  
    [Id]          INT          IDENTITY (1, 1) NOT NULL,  
    [name]        NVARCHAR (50) NOT NULL,  
    [specs]       NVARCHAR (50) NULL,  
    [image]       IMAGE        NULL,  
    [price]       FLOAT (53)   NOT NULL,  
    [datetime]    DATE          NULL,  
    [location]    NVARCHAR (50) NULL,  
    [manager]     INT          NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC),  
    FOREIGN KEY ([manager]) REFERENCES [dbo].[Employee] ([Id]) ON DELETE SET  
NULL  
);
```

部门信息表：

```
CREATE TABLE [dbo].[Department] (  
    [Id]          INT          IDENTITY (1, 1) NOT NULL,  
    [name]        NVARCHAR (50) NOT NULL,  
    [admin]       INT          NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC),  
    FOREIGN KEY ([admin]) REFERENCES [dbo].[Employee] ([Id]) ON DELETE SET NULL  
);
```

6. 实验总结

本系统使用传统的 MVC 进行开发，前端使用 bootstrap 进行页面美化，使用 jquery 进行页面控制和数据验证，数据验证后才能进入后台，进行数据库操作。熟悉了 MVC 的操作流程：控制器对数据进行处理，将结果绑定为 model 传给对应视图，由于本系统不是很复杂，没有多余的逻辑代码，只是对数据库简单的增删查改，故很好实现。系统使用验证 Session 方式来验证用户登录情况，根据 Session 来判定用户角色。

不足之处：前端使用 .net 代码嵌入方式嵌入 C# 代码到页面中，造成很多的代码冗余，后面应该配合 jquery 的 ajax 对页面进行动态渲染。还有就时对 C# 的控件不是很熟悉，很少使用，都是使用 JavaScript 进行页面控制。

通过本课程的学习以及设备管理系统的开发，了解了 .net 的基本开发方式，了解了基于 C# 的 WEB 网站开发。

本文件说明（上传学者网前请删除本段文字）

- 本文格式及字体大小请勿修改（除非有必要），字体则可适度调整。
- 本文所列章节不可缺少，若有必要可适度增加或修改。
- 必要源码不可以截图贴上，需以文本贴上，并调小段间距，以免篇幅过大。
- 本文件完成后请另存为 P D F 格式再提交。