

计算机科学与技术学院、大数据学院



课程：软件设计模式

综合实验报告

题目：设备保管及查询系统

姓 名：黄磊

学 号：201824113330

班 级：18 级软件工程 3 班

任课教师：许盛贵

2020 年 12 月 15 日

1. 系统开发目的

本系统将建立一个简易 WEB 设备保管及查询系统，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进一步实施设备资源的共享，以避免不必要的重覆购入相同设备；同时记录企业内所有员工信息和部门信息，供系统管理员进行信息查询和管理。

2. 系统需求

1. 功能性需求：

(1) 员工分系统管理人(也是一般员工)及一般员工，所有人员都可进行设备保管。人员均需编列于特定部门中，如财务部或销售部，一般企业部门名称请自订。

(2) 所有人员使用系统需先行登陆，并按人员身份呈现不同首页。

(3) 系统管理人可作 A. 部门，B. 设备及 C. 人员的 C R U D；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下列条件查询，包含设备编号、设备名称、购入日期（年度）、存放位置、设备负责人姓名、部门名称。

(5) 管理人查询：除了前项查询功能外，尚需包含部门查询及人员查询。

2. 非功能性需求：

(1) 需采 Java 编程语言及 MySQL 数据库开发一个网站型管理系统。

(2) 需采用 M V C + D A O 设计模式或 S S H 框架作系统开发。

(3) 各项编码均需采 utf-8 编码，含数据库及各类程序代码或网页。

3. 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)（作为登陆用）、密码、姓名、联络电话、是否为管理人、部门代号(FK)。

(4) 需有 E - R 图设计，并请合理设计上述数据型态及限制。

(5) 数据库需使用 MySQL 数据库管理系统。

(6) 各项命名均需使用英文，如设备数据表可使用 Equipment，员工可使用 Employee，部门可使用 Department。所有字段名亦同。英文使用请用意译。

4. Web 界面设计实作要求：(15%)

(1) 建置一个 Web 项目，项目名称为 EMS+学号，学号为你的真实学号，格式如

EMS2016011012, 并完成下列各项功能。

(2) 设计一个首页 index.jsp, 初次连结至本页时必须先登入, 登入成功后能以菜单(menu)方式连结至各功能页面。

(3) 用户选取某项功能时系统需检查使用者是否有完成登陆或是否具有权限。

(4) WEB 界需具初步美观设计。

3. 系统分析及设计

3.1 开发及运行环境

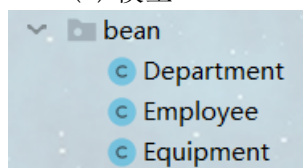
开发环境: idea、jdk1.8、mysql8.0

运行环境: tomcat9.0.35、windows10

3.2 系统功能架构设计

本项目的系统功能架构采用 MVC+DAO 模式, 并使用 filter 过滤器进行权限拦截。

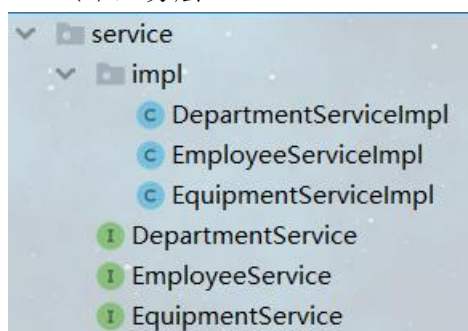
(1) 模型 Model



(2) 持久层 Dao



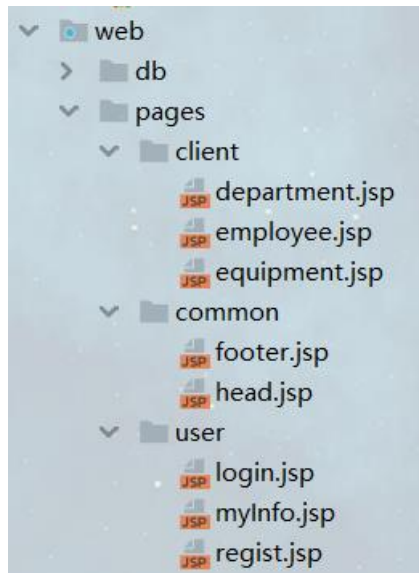
(3) 业务层 Service



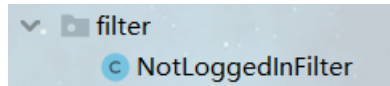
(4) 控制层 Controller



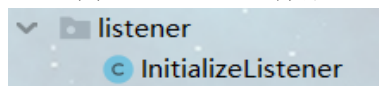
(5) 视图层 View



(6) Filter 过滤器



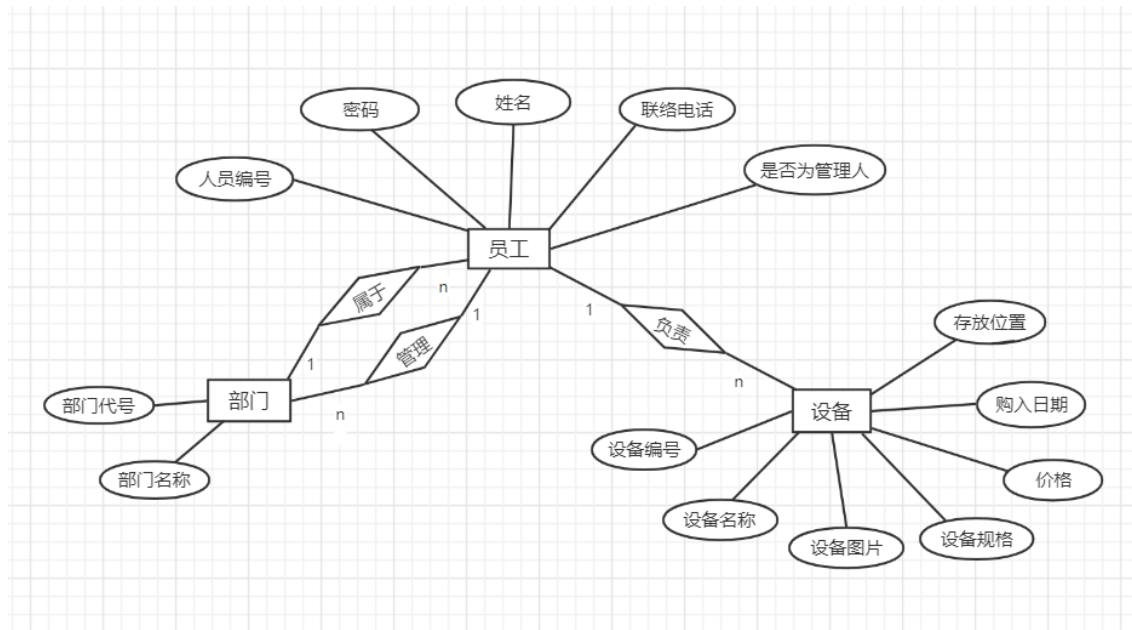
(7) Listener 监听器



(8) Utils 工具类



3.3 数据库概念设计



员工实体：共有 5 个属性，分别为人员编号、密码、姓名、联络电话、是否为管理人。

部门实体：共有 2 个属性，分别为部门代号、部门名称。

设备实体：共有 4 个属性，分别为设备编号、设备名称、设备图片、设备规格、价格、购入日期、存放位置。

实体间的联系：一个员工可以管理多个部门，多个员工可以属于一个部门，一个员工可以负责多台设备。

3.4 数据库逻辑设计

(1) 创建数据库 EquipmentStorage:

```
CREATE DATABASE EquipmentStorage DEFAULT CHARSET utf8mb4;
```

(2) 创建 employee 员工表:

```
CREATE TABLE IF NOT EXISTS employee (
    `id` CHAR(10) PRIMARY KEY,
    `password` VARCHAR(32) NOT NULL,
    `name` VARCHAR(50),
    `phone` CHAR(13),
    `usertype` CHAR(8),
    `department` VARCHAR(20)
)
```

(3) 创建 department 部门表:

```
CREATE TABLE IF NOT EXISTS department (
    `id` VARCHAR(20) PRIMARY KEY,
```

```

`name` VARCHAR(50) NOT NULL,
`manager` CHAR(10),
FOREIGN KEY(`manager`) REFERENCES employee(`id`)
)

```

(4) 创建 equipment 设备表:

```

CREATE TABLE IF NOT EXISTS equipment (
  `id` VARCHAR(30) PRIMARY KEY,
  `name` VARCHAR(50),
  `specification` VARCHAR(120),
  `price` DECIMAL(11, 2),
  `buydate` DATE,
  `position` VARCHAR(120),
  `img_path` VARCHAR(200),
  `manager` CHAR(10),
  FOREIGN KEY(`manager`) REFERENCES employee(`id`)
)

```

(5) 添加 employee 员工表的外键约束

```

ALTER TABLE employee ADD FOREIGN KEY(department) REFERENCES
department(`id`);

```

4. 软件操作说明

4.1 一般员工之功能操作

(1) 查询设备信息及部门信息

设备

部门

欢迎您! 贾克斯!

设备信息表

全部

请输入要搜索的信息

查询

图片	编号	名称	规格	购入日期	价格	存放位置	负责人
	0001	键盘	70*30	2020-12-10	265.0	仓库A	admin
	0002	硬盘	50*50	2020-11-05	566.5	仓库B	15233
	0003	鼠标	15*9	2020-12-02	76.48	仓库A	21824
	0004	音响	70*100	2020-12-12	235.0	仓库C	admin
	0005	打印机	60*40	2020-07-04	524.0	仓库A	11511

设备

部门

欢迎您! 贾克斯!

设备信息表

存放位置

仓库A

查询

图片	编号	名称	规格	购入日期	价格	存放位置	负责人
	0001	键盘	70*30	2020-12-10	265.0	仓库A	admin
	0003	鼠标	15*9	2020-12-02	76.48	仓库A	21824
	0005	打印机	60*40	2020-07-04	524.0	仓库A	11511

设备

部门

欢迎您! 贾克斯!

部门信息表

全部

请输入要搜索的信息

查询

部门代号	部门名称	部门主管
101	财务部	11439
102	销售部	15233
103	运营部	14362
104	策划部	11511
105	公关部	21824
106	人事部	admin

(2)查看自己的用户信息

设备

部门

欢迎您! 贾克斯!

我的信息

工号

21824

姓名

贾克斯

联系电话

14523695266

用户类型

普通员工

所属部门

105

修改信息

修改密码

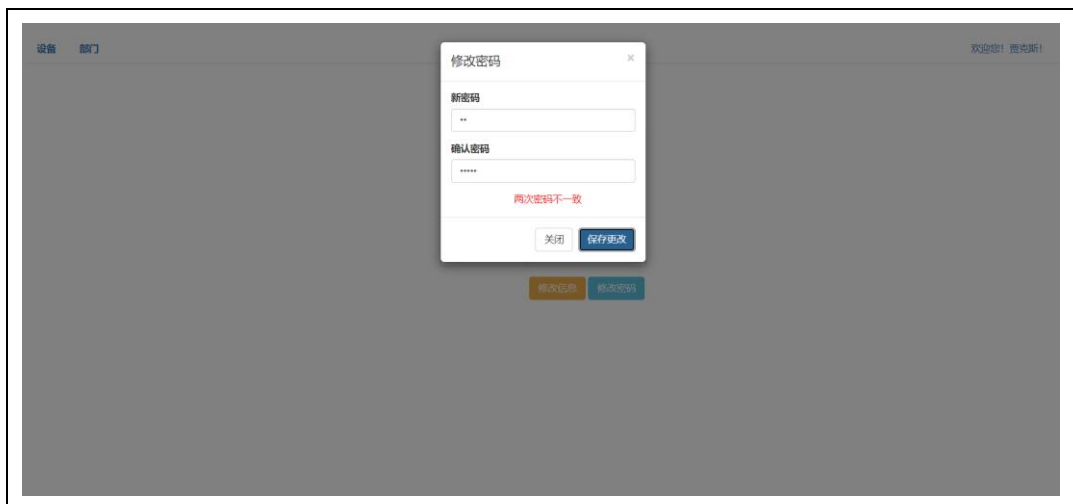
我的信息

注销

(3)修改个人信息



(4)修改个人密码



4.2 系统管理员之功能操作

(1) 查询所有员工信息

设备信息表								
<div> <div>全部</div> <div>请输入要搜索的信息</div> <div>查询</div> <div>添加</div> </div>								
图片	编号	名称	规格	购入日期	价格	存放位置	负责人	操作
	0001	键盘	70*30	2020-12-10	265.0	仓库A	admin	修改 删除
	0002	硬盘	50*50	2020-11-05	566.5	仓库B	15233	修改 删除
	0003	鼠标	15*9	2020-12-02	76.48	仓库A	21824	修改 删除
	0004	音响	70*100	2020-12-12	235.0	仓库C	admin	修改 删除
	0005	打印机	60*40	2020-07-04	524.0	仓库A	11511	修改 删除

(2) 查询所有部门信息

设备

部门

员工

欢迎您! 黄磊!

部门信息表

全部

请输入要搜索的信息

查询

添加

部门代号	部门名称	部门主管	操作
101	财务部	11439	修改 删除
102	销售部	15233	修改 删除
103	运营部	14362	修改 删除
104	策划部	11511	修改 删除
105	公关部	21824	修改 删除
106	人事部	admin	修改 删除

(3) 查询所有员工信息

设备

部门

员工

欢迎您! 黄磊!

员工信息表

全部

请输入要搜索的信息

查询

添加

工号	姓名	密码	联系电话	用户类型	所属部门	操作
11439	李四	123	15842632569	普通员工	101	修改 删除
11511	张三	123	1423652452	系统管理员	104	修改 删除
14362	李白	123	15489632369	普通员工	103	修改 删除
15233	薇恩	123	15362547166	系统管理员	102	修改 删除
17362	科加斯	123	14362596325	普通员工	104	修改 删除
21824	贾克斯	000	14523695266	普通员工	105	修改 删除
admin	黄磊	123	15046253961	系统管理员	106	修改 删除

(4) 按条件查询信息

设备

部门

员工

欢迎您! 黄磊!

员工信息表

用户类型

系统管理员

查询

添加

工号	姓名	密码	联系电话	用户类型	所属部门	操作
11511	张三	123	1423652452	系统管理员	104	修改 删除
15233	薇恩	123	15362547166	系统管理员	102	修改 删除
admin	黄磊	123	15046253961	系统管理员	106	修改 删除

(5) 查询个人信息

设备 部门 员工

欢迎您! 黄磊!

我的信息

工号	admin
姓名	黄磊
联系电话	15046253961
用户类型	系统管理员
所属部门	106

修改信息

修改密码

(6) 添加设备

设备 部门 员工






欢迎您! 黄磊!

添加设备信息

编 号	0006
名 称	扫描仪
规 格	340*220
购入日期	2019/02/09
价 格	563.43
图 片	选择文件 扫描仪.png
存放位置	仓库B
负 责 人	admin

关闭

提交

图片	编号	名称	规格	购入日期	价格	存放位置	负责人	操作
	0001	键盘					admin	修改 删除
	0002	硬盘					15233	修改 删除
	0003	鼠标					21824	修改 删除
	0004	音响					admin	修改 删除
	0005	打印机	60*40	2020-07-04	524.0	仓库A	11511	修改 删除

设备 部门 员工

欢迎您! 黄磊!

localhost:8080 显示
添加成功!

确定







设备信息管理

全部

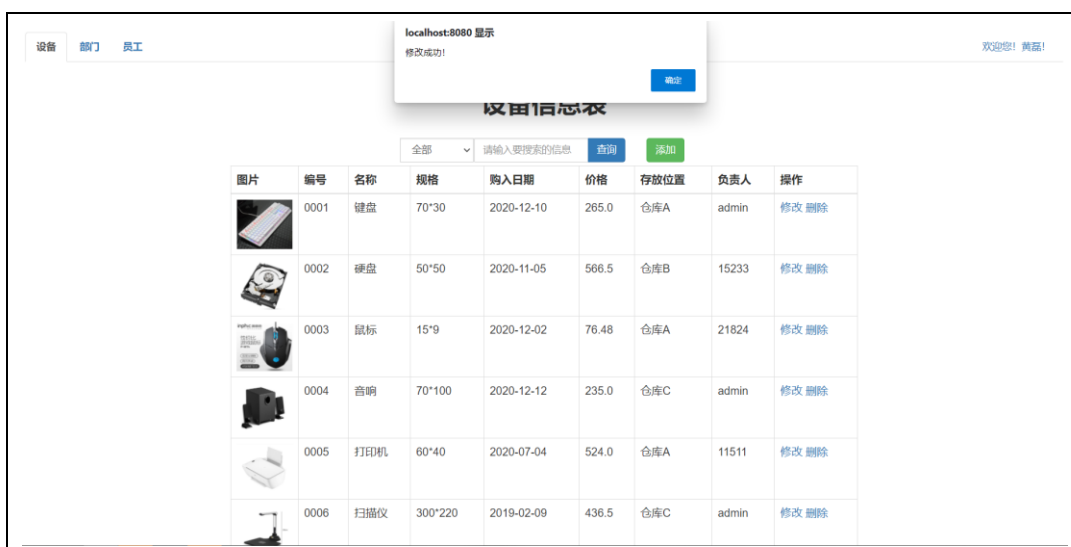
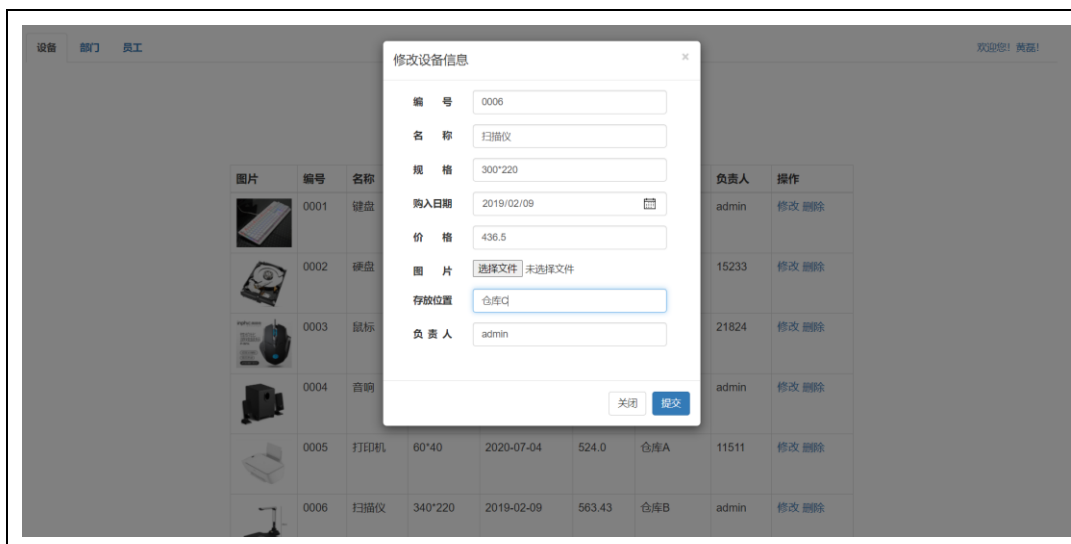
请输入要搜索的信息

查询

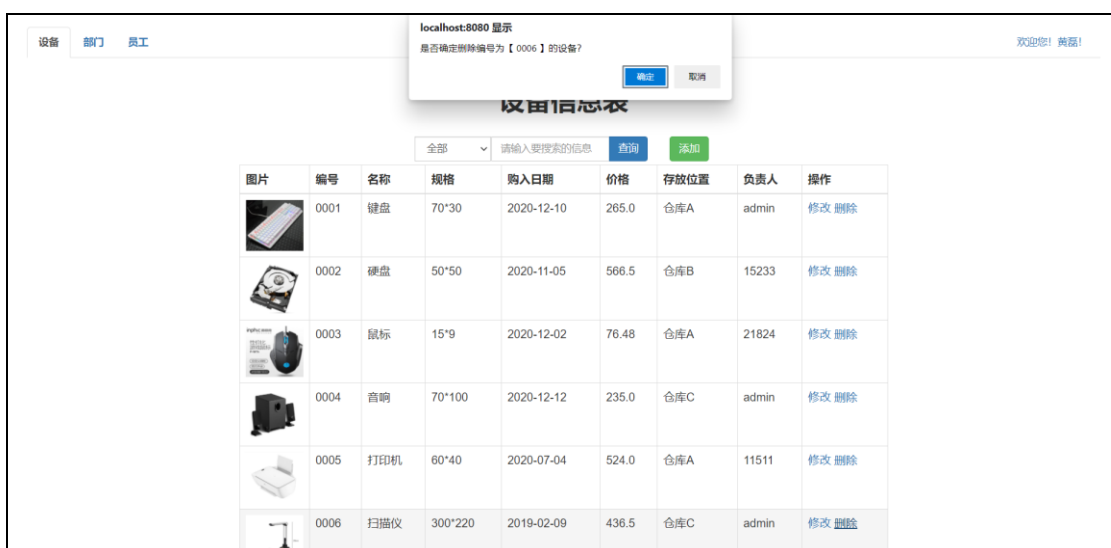
添加

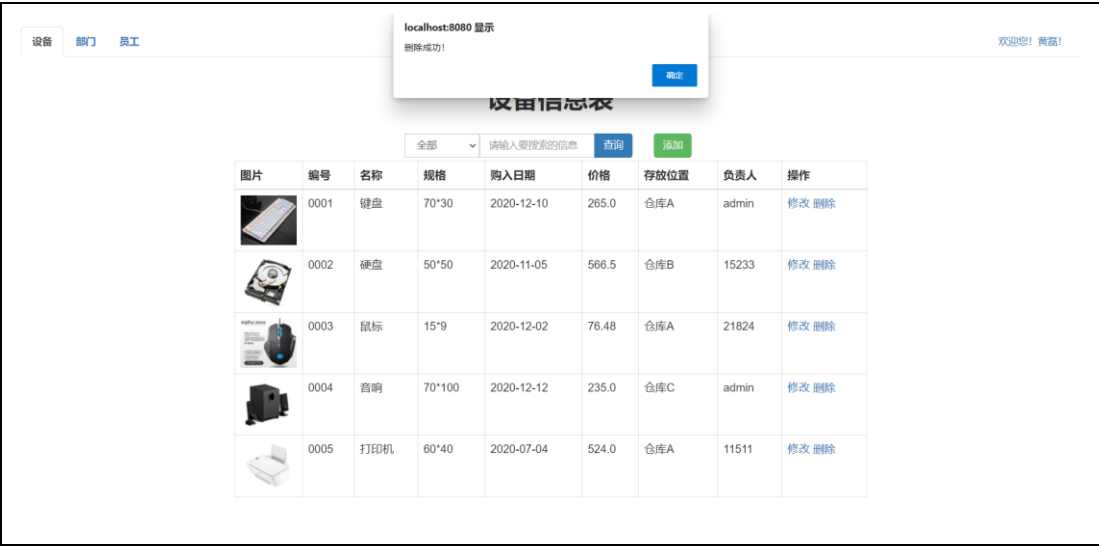
图片	编号	名称	规格	购入日期	价格	存放位置	负责人	操作
	0001	键盘	70*30	2020-12-10	265.0	仓库A	admin	修改 删除
	0002	硬盘	50*50	2020-11-05	566.5	仓库B	15233	修改 删除
	0003	鼠标	15*9	2020-12-02	76.48	仓库A	21824	修改 删除
	0004	音响	70*100	2020-12-12	235.0	仓库C	admin	修改 删除
	0005	打印机	60*40	2020-07-04	524.0	仓库A	11511	修改 删除
	0006	扫描仪	340*220	2019-02-09	563.43	仓库B	admin	修改 删除

(7) 修改设备信息

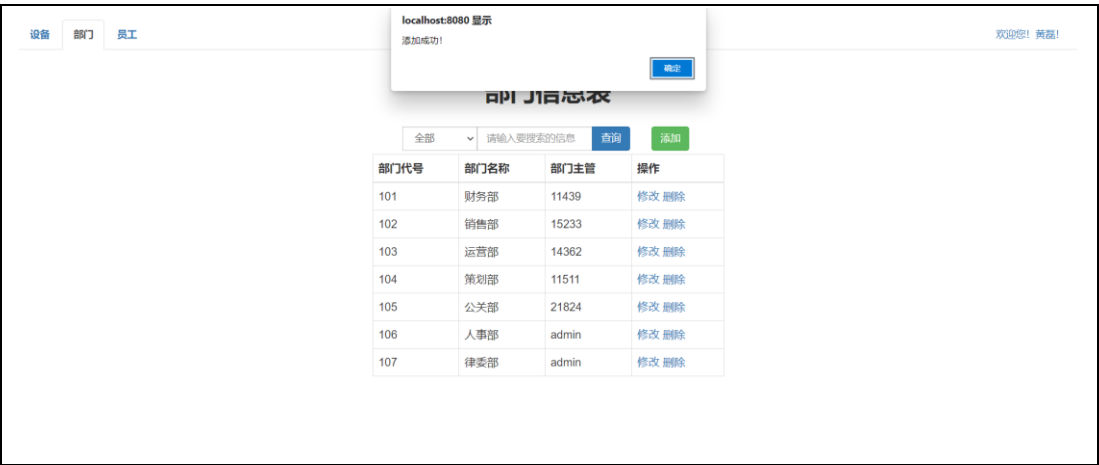
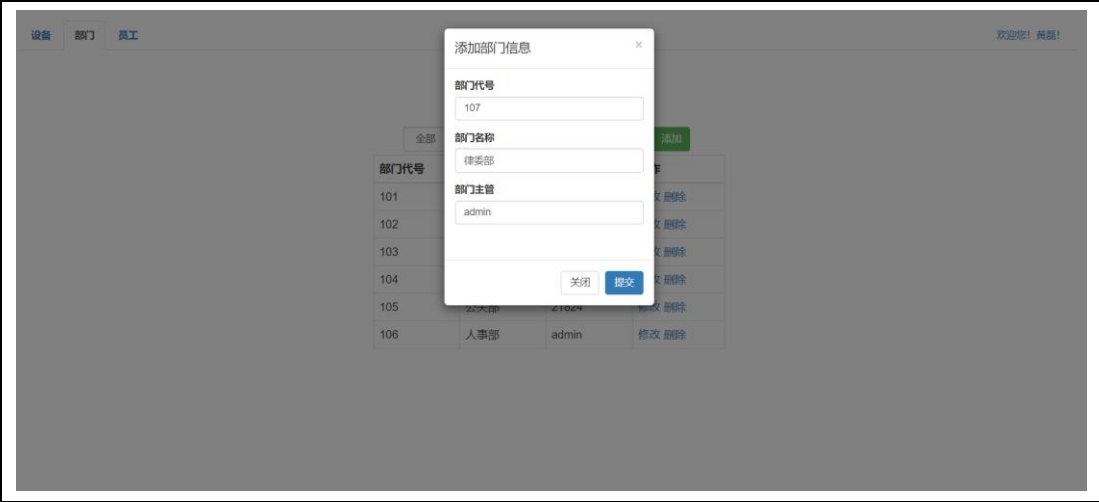


(8) 删除设备

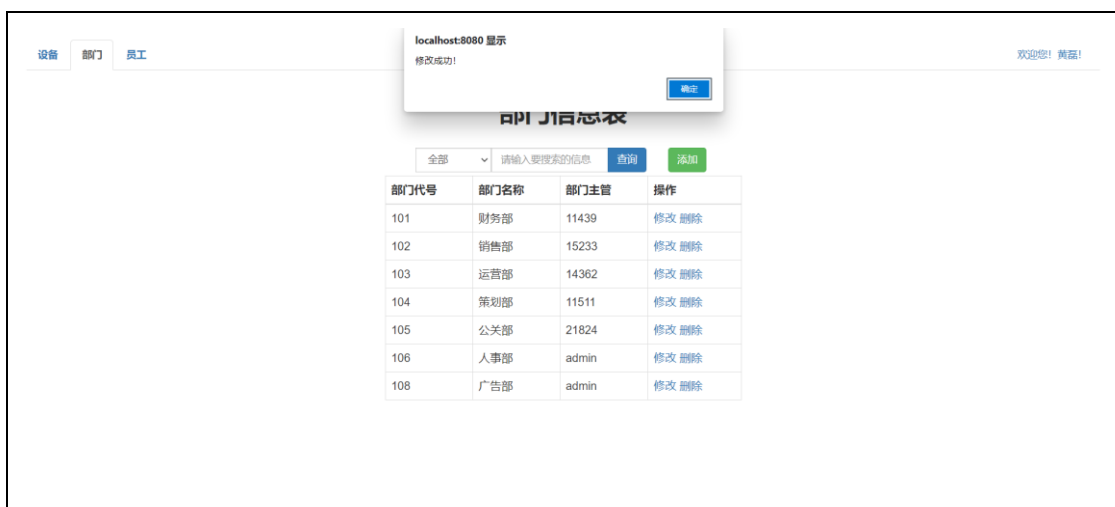




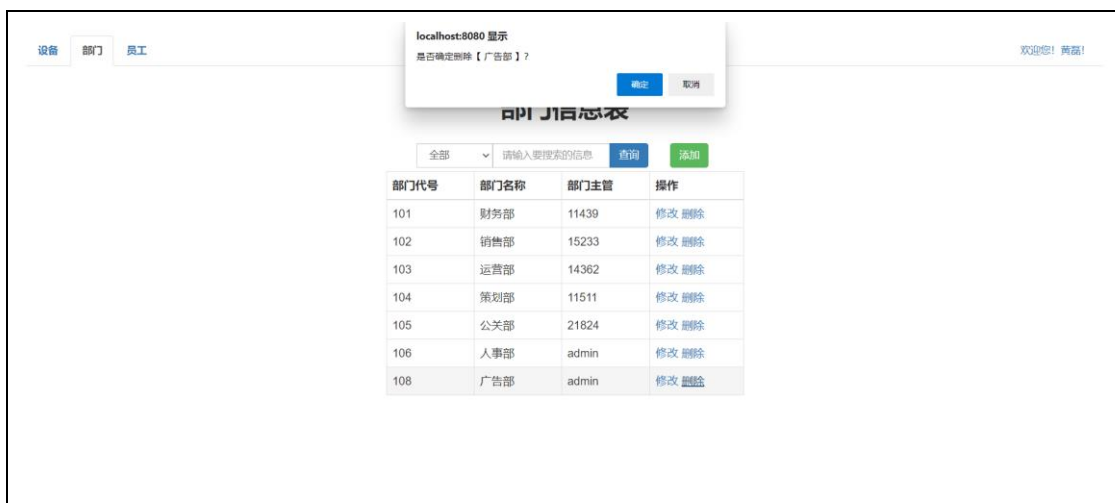
(9) 添加部门



(10) 修改部门信息

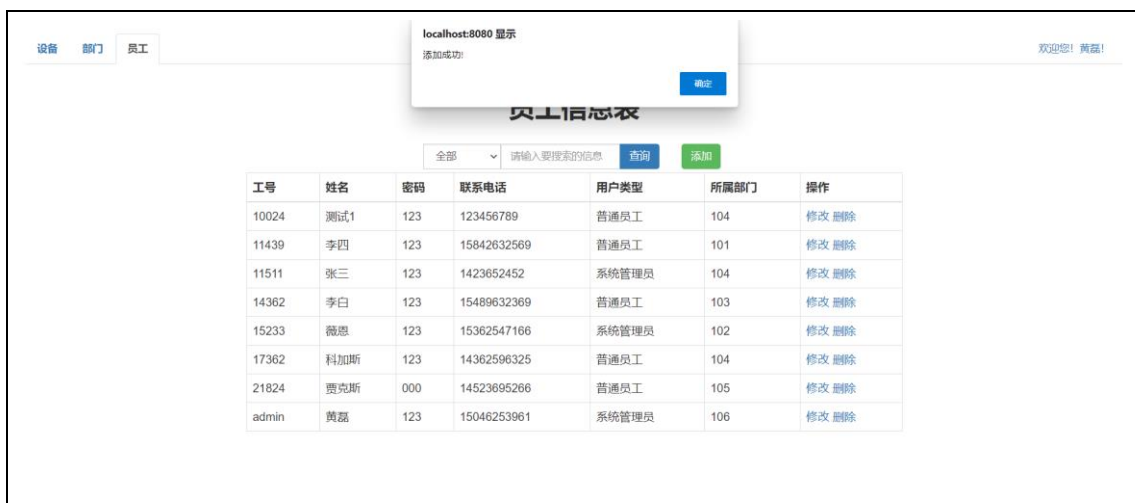
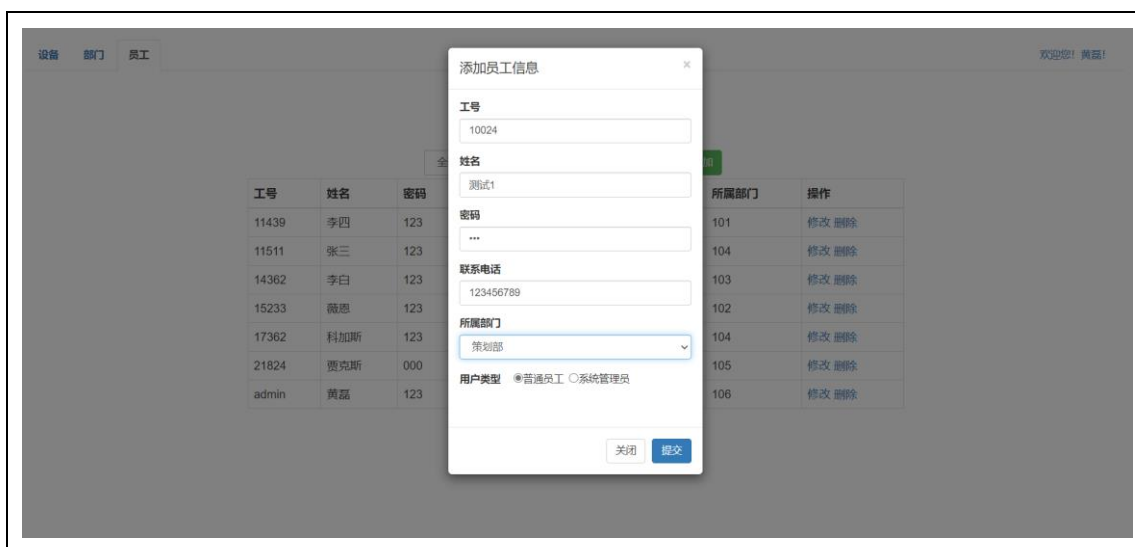


(11) 删除部门

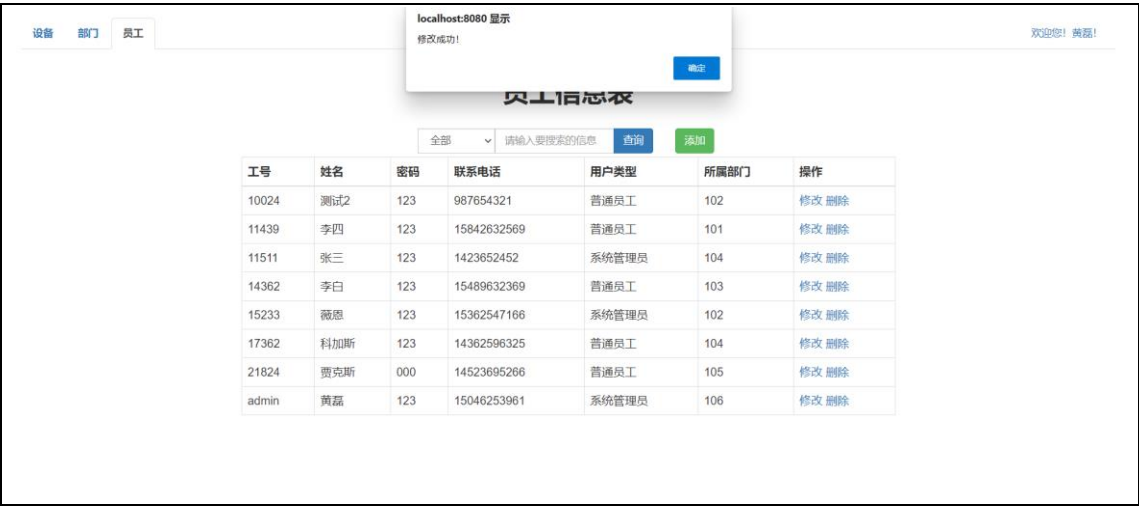
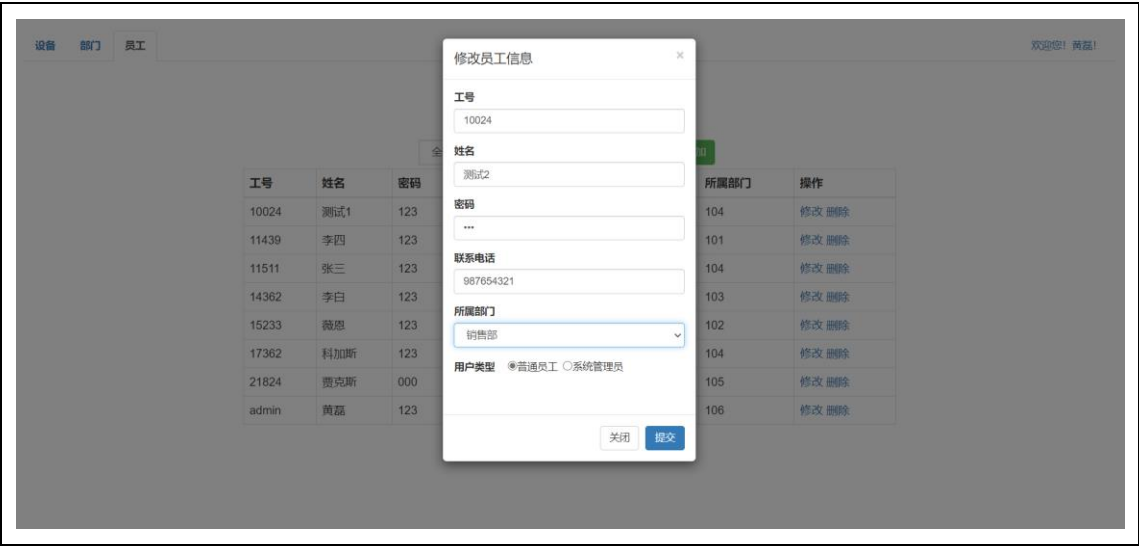




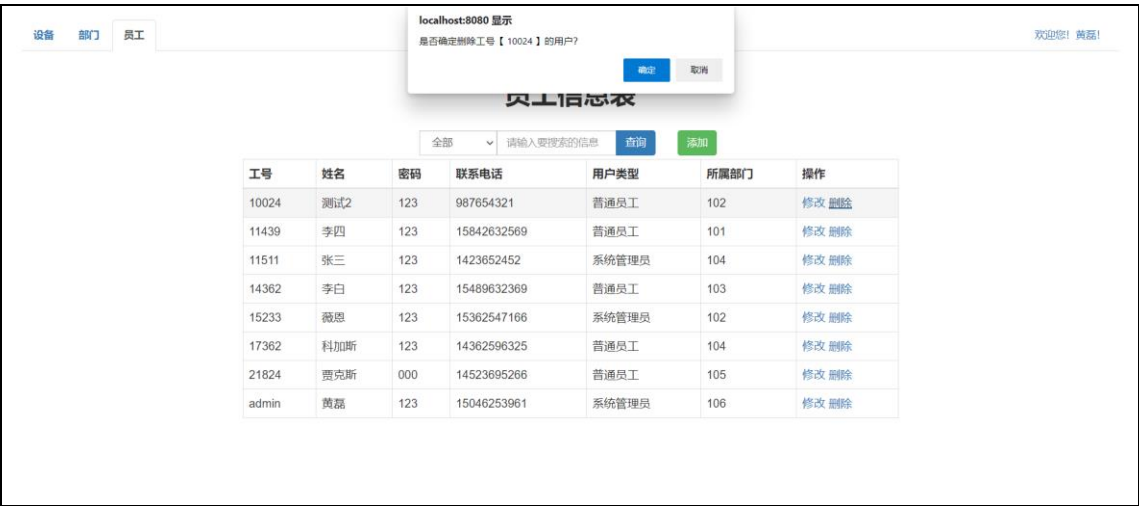
(12) 添加员工



(13) 修改员工信息



(14) 删除员工





4.3 其它功能操作

(1) 用户登录



(2) 用户注册

员工 ID

14785

✓

姓 名

王五

✓

密 码

...

✓

确认密码

...

✓

联系电话

14563254785

✓

所属部门

人事部

▼

用户类型

☒ 普通员工 ☐ 系统管理员

立即注册

已有账号, 立即登录

设备保管查询系统
Author: 黄磊 · 201824113330

5. 重要程序源码说明

5.1 前端界面源码

1. head.jsp

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>

<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<%

    String basePath = request.getScheme() //获取 ip 协议
        + "://"
        + request.getServerName() //获取服务器主机名
        + ":"
        + request.getServerPort() //获取端口号
        + request.getContextPath() //获取项目路径
        + "/";

%>
<base href="<%=basePath%>">
<!-- Bootstrap -->
<link href="static/css/bootstrap.min.css" rel="stylesheet">
```

```

<script src="static/js/jquery-3.2.1.min.js"></script>
<script src="static/js/bootstrap.min.js"></script>

<!-- 加载我的js 文件-->

<script src="static/js/myjs.js"></script>

```

2. footer.jsp

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<br/><br/><br/>
<div class="text-center" style="margin-bottom:0px">

    设备保管查询系统<br>

    Author : 黄磊 - 201824113330

</div>

```

3. 用户登录界面代码 login.jsp

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <%@include file="/pages/common/head.jsp"%>
    <title>登录</title>

    <style type="text/css">

        .container {
            margin: 110px auto;
            width: 430px;
        }
        .form-horizontal {
            padding-top: 28px;
            padding-left: 31px;
            padding-bottom: 6px;
            background-color: white;

            border-radius: 15px; /* 边框圆角 */

            box-shadow: 0px 2px 2px 2px #e6e6e6; /* 边框阴影 */

            width: 400px;
        }
        .alert-danger {
            width: 400px;
        }
    </style>

```

```

        visibility: hidden;
    }
    .body {
        background-color: #f4f0ff;
    }
    .form-control {
        width: 260px;
    }
    .btn {
        margin-right: 15px ;
    }
</style>

<script type="text/javascript">
    $(function () {
        window.onload = function() {
            getLoginCookie();
        };

        $("#login_btn").click(function () {
            var id = $("#id").val();
            var password = $("#password").val();
            var patter = /\S/;
            if (!patter.test(id) || !patter.test(password)) {

                $("#span.msg").text("请正确输入账号和密码!");

                $("#prompt").css('visibility', 'visible');
                return false;
            }
        });

        var msg = '${requestScope.msg}';
        if (msg) {
            $("#span.msg").text(msg);
            $("#prompt").css('visibility', 'visible');
        }

        $("#id").focus(function () {
            $("#prompt").css('visibility', 'hidden');
        });

        $("#regist_btn").click(function () {
            location.href = "pages/user/regist.jsp";
        })
    })

```

```

    })
  </script>
</head>
<body class="body">
  <div class="container">
    <div class="row clearfix">
      <div class="col-md-12 column">
        <h1 style="text-align: center;">设备保管及查询系统</h1><br/><br/>
        <div class="alert alert-danger" id="prompt">
          <strong>警告！</strong> <span class="msg"> ${requestScope.msg}
        </span>
      </div>
      <div>
        <form class="form-horizontal" role="form" action="userServicelet"
method="post">
          <input type="hidden" name="action" value="login">
          <div class="form-group">
            <label for="id" class="col-sm-2 control-label">账号</label>
            <div class="col-sm-10">
              <input type="text" class="form-control" id="id" placeholder="
请输入账号" name="id" value="${requestScope.id}">
            </div>
          </div>
          <div class="form-group">
            <label for="password" 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" name="password">
            </div>
          </div>
          <div class="form-group">
            <div class="col-sm-offset-2 col-sm-10">
              <div class="checkbox">
                <label><input type="checkbox" name="remember"
checked/>Remember me</label>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <div class="form-group">
            <div class="col-sm-offset-2 col-sm-10">
                <button type="submit" class="btn btn-default" id="login_btn">
登录</button>

                <button type="button" class="btn btn-default" id="regist_btn">
注册</button>

            </div>
        </div>
    </form>
</div>
</div>
<%@include file="/pages/common/footer.jsp" %>
</body>
</html>

```

4. 用户注册界面 regist.jsp

```

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <%@include file="/pages/common/head.jsp"%>

    <title>注册</title>

    <style type="text/css">
        .container {
            margin: 80px auto;
            width: 600px;
        }
        .body {
            background-color: #f4f0ff;
        }
        .form-horizontal {
            padding-top: 30px;
            padding-left: 60px;
            padding-bottom: 5px;
            background-color: white;

            border-radius: 15px; /*边框圆角*/
        }
    </style>

```

```

        box-shadow: 0px 2px 2px 2px #e6e6e6; /*边框阴影*/
    }
    #img {
        width: 23px;
        height: 23px;
    }
    .form-control {
        width: 260px;
    }
    .control-label {
        width: 90px;
    }
    .btn {
        margin-right: 15px ;
    }
    a {
        text-decoration: none;
        font-size: 1.1em;
    }
    a:hover{
        color: red;
    }
</style>
<script type="text/javascript">
    $(function () {
        getDepartmentOptions();

        var ids = $("#id");
        var names = $("#name");
        var pwds = $("#password");
        var repwds = $("#repwd");
        var phones = $("#phone");
        var idExists = 0;

        // 注册页面， 用户名输入框， 光标丢失焦点事件， ajax 判断用户名是否已存在

        ids.blur(function () {
            var id = this.value;
            if (id !== "") {
                $.getJSON("userService", "action=existsId&id=" + id,
function (data) {
                    if (data.existsId != null) {
                        $("#span_id_msg").append("<img
src='static/img/wrong.jpg' id='img' alt=''>");

```

```

        idExists = 1;
    } else {
        $("span.id_msg").append("<img
src='static/img/right.jpg' id='img' alt=''>");
        idExists = 0;
    }
    })
}
});

// 判断输入的密码和确认密码是否一致
repwds.blur(function () {
    var password = pwds.val();
    var repwd = repwds.val();
    if (repwd !== "") {
        if (password === repwd) {
            $("span.repwd_msg").append("<img
src='static/img/right.jpg' id='img' alt=''>");
        } else {
            $("span.repwd_msg").append("<img
src='static/img/wrong.jpg' id='img' alt=''>");
        }
    }
});
names.blur(function () {
    var name = names.val();
    if (name !== "") {
        $("span.name_msg").append("<img
src='static/img/right.jpg' id='img' alt=''>");
    }
});
pwds.blur(function () {
    var pwd = pwds.val();
    if (pwd !== "") {
        $("span.pwd_msg").append("<img
src='static/img/right.jpg' id='img' alt=''>");
    }
});
phones.blur(function () {
    var phone = phones.val();
    if (phone !== "") {
        $("span.phone_msg").append("<img
src='static/img/right.jpg' id='img' alt=''>");
    }
}

```

```

});
ids.focus(function () {
    $("#span.id_msg img").remove()
});
names.focus(function () {
    $("#span.name_msg img").remove()
});
pwds.focus(function () {
    $("#span.pwd_msg img").remove()
});
repwds.focus(function () {
    $("#span.repwd_msg img").remove()
});

phones.focus(function () {
    $("#span.phone_msg img").remove()
});
$("#regists_btn").click(function () {
    var patter = /\S/;
    var id= ids.val();
    var name= names.val();
    var password = pwds.val();
    var repwd = repwds.val();
    var phone = phones.val();
    if (!patter.test(id)||!patter.test(name)
||!patter.test(password)||!patter.test(repwd)||!patter.test(phone)||passw
ord!=repwd||idExists===1) {
        return false;
    }
});
})
</script>
</head>
<body class="body">
<div class="container">
    <div class="row clearfix">
        <div class="col-md-12 column">
            <form class="form-horizontal" role="form" action="userServlet"
method="post">
                <input type="hidden" name="action" value="regists">
                <div class="form-group">
                    <label for="id" class="col-sm-2 control-label">员&ensp;

```



```

        </div>
        <div class="form-group">
            <label for="phone" class="col-sm-2 control-label">联系电
话</label>

            <div class="col-sm-6">
                <input type="text" class="form-control" id="phone"
placeholder="请输入联系电话" name="phone">
            </div>
            <span class="col-sm-4 phone_msg"></span>
        </div>
        <div class="form-group">
            <label class="col-sm-2 control-label">所属部门</label>
            <div class="col-sm-10">
                <select class="form-control department_option"
name="department"> </select>
            </div>
        </div>
        <div class="form-group">
            <label class="col-sm-2 control-label">用户类型</label>
            <div class="radio col-sm-10">
                <label>
                    <input type="radio" name="userType"
id="optionsRadios1" value="普通员工" checked>普通员工
                </label>
                <label>
                    <input type="radio" name="userType"
id="optionsRadios2" value="系统管理员">系统管理员
                </label>
            </div>
        </div>

        <div class="form-group">
            <div class="col-sm-offset-2 col-sm-10">
                <button type="submit" class="btn btn-default"
id="regists_btn">立即注册</button>
            </div>
        </div>
    </div>
</div>

```

```

        <a href="pages/user/login.jsp">已有账号，立即登录</a>
    </div>
</div>
</form>
</div>
</div>
</div>
<%@include file="/pages/common/footer.jsp" %>
</body>
</html>

```

5. 个人信息界面 myInfo.jsp

```

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <%@include file="/pages/common/head.jsp"%>
    <title>我的信息</title>

    <style type="text/css">
        .container-fluid {
            margin: 20px auto;
        }
        .table {
            width: 300px;
            margin: 0 auto;
        }
        .cls {
            margin: 15px auto;
            width: 300px;
        }
        #changePwdPrompt, #changeInfoPrompt{
            visibility: hidden;
            margin-left: 92px;
        }
        #changeInfo_btn {
            margin-left: 130px;
        }
    </style>

    <script type="text/javascript">

```

```

$(function () {
    getDepartmentOptions();

    var changePwdPrompt = $("#changePwdPrompt");
    var changeInfoPrompt = $("#changeInfoPrompt");

    $("#changePwd_btn").click(function () {
        $("#new_pwd, #repwd").val("");
        changePwdPrompt.css('visibility', 'hidden');
    });
    $("#changeInfo_btn").click(function () {
        $("#department").find("option[value=" +
        ${sessionScope.loginUser.department} + "']").attr("selected", true);
        changeInfoPrompt.css('visibility', 'hidden');
    });
    $("#save_change").click(function () {
        var newPwd = $("#new_pwd").val();
        var rePwd = $("#repwd").val();
        if (newPwd === "" || rePwd === "") {
            changePwdPrompt.html("请正确输入");
            changePwdPrompt.css('color', 'red');
        } else if (newPwd !== rePwd) {
            changePwdPrompt.html("两次密码不一致");
            changePwdPrompt.css('color', 'red');
        } else {
            $.getJSON("employeeServlet",
            "action=changePassword&id=${sessionScope.loginUser.id}&newPassword=" +
            newPwd, function (data) {
                //console.log(data)
                if (data.result === 1) {
                    changePwdPrompt.text("密码修改成功!");
                    changePwdPrompt.css('color', 'green');
                }
            })
        }
        changePwdPrompt.css('visibility', 'visible');
    });
    $("#new_pwd, #repwd").focus(function () {
        changePwdPrompt.css('visibility', 'hidden');
    });
    $("#save_change_2").click(function () {

```

```

        var name = $("#name").val();
        var phone = $("#phone").val();
        if (name === "" || phone === "") {

            changeInfoPrompt.text("信息不得为空");

            changeInfoPrompt.css("color", "red");
        } else {

$.getJSON("employeeServlet", "action=changeBasicInfo&id=${sessionScope.loginUser.id}&" + $("#changeInfo_form").serialize(), function (data) {
    //console.log(data);
    if (data.result === 1) {

        changeInfoPrompt.text("信息修改成功!");

        changeInfoPrompt.css("color", "green");

    }
    });
    changeInfoPrompt.css('visibility', 'visible');
});
$("#close").click(function () {
    location.href =
"employeeServlet?action=refreshLoginUserInfo&id=${sessionScope.loginUser.id}";

    });
})
</script>
</head>
<body class="body">
<div class="container-fluid">
    <div class="row-fluid">
        <div class="span12">
            <ul class="nav nav-tabs">
                <li>

                    <a href="equipmentServlet?action=showAll"><strong>设备
</strong></a>

                </li>
                <li>

                    <a href="departmentServlet?action=showAll"><strong>部门
</strong></a>

                </li>
            </ul>
        </div>
    </div>
</body>

```

```

        <c:if test="${sessionScope.loginUser.userType == '系统管理员'
    '}">

        <li>

            <a href="employeeServlet?action=showAll"><strong>员工

        </strong></a>

        </li>
        </c:if>
        <li class="dropdown pull-right">
            <a class="dropdown-toggle" data-toggle="dropdown"
href="#">欢迎您! ${sessionScope.loginUser.name}! </a>

            <ul class="dropdown-menu">
                <li>

                    <a href="pages/user/myInfo.jsp">我的信息</a>

                </li>
                <li class="divider"></li>
                <li>

                    <a href="userServlet?action=logout">注销</a>

                </li>
            </ul>
        </li>
    </ul>

    <h3 style="text-align: center; margin-top: 40px">
        <strong>我的信息</strong>
    </h3>
    <br/>
    <table class="table table-hover table-bordered">
        <thead>
            <tr>

                <th width="80px"> 工号 </th>

                <td> ${sessionScope.loginUser.id} </td>

            </tr>
        </thead>
        <tbody>
            <tr>

                <th> 姓名 </th>

```

```

        <td> ${sessionScope.loginUser.name} </td>
    </tr>
</tbody>
<tbody>
<tr>
    <th> 联系电话 </th>
    <td> ${sessionScope.loginUser.phone} </td>
</tr>
</tbody>
<tbody>
<tr>
    <th> 用户类型 </th>
    <td> ${sessionScope.loginUser.userType} </td>
</tr>
<tr>
    <th> 所属部门 </th>
    <td> ${sessionScope.loginUser.department} </td>
</tr>
</tbody>
</table>
<div class="cls">
    <!-- Button trigger modal -->
    <input class="btn btn-warning" type="submit"
id="changeInfo_btn" value="修改信息" data-toggle="modal"
data-target="#alter_msg">
    <input class="btn btn-info " type="submit" id="changePwd_btn"
value="修改密码" data-toggle="modal" data-target="#alter_pwd">
</div>

<!-- Modal -->
<div class="modal fade" id="alter_pwd" tabindex="-1"
role="dialog" aria-labelledby="alter_pwd_Label">
    <div class="modal-dialog modal-sm" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close"
data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>

```

```

        <h4 class="modal-title" id="alter_pwd_Label">修改
密码</h4>

    </div>
    <div class="modal-body">
        <form method="post" id="form">
            <div class="form-group">
                <label for="new_pwd">新密码</label>
                <input type="password"
class="form-control" id="new_pwd" placeholder="请输入新密码">
            </div>
            <div class="form-group">
                <label for="repwd">确认密码</label>
                <input type="password"
class="form-control" id="repwd" placeholder="请再次输入密码"
name="newPassword">
            </div>
        </form>
        <span id="changePwdPrompt">&nbsp;</span>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-default"
data-dismiss="modal">关闭</button>
        <button type="button" class="btn btn-primary"
id="save_change">保存更改</button>
    </div>
</div>
</div>
</div>
</div>
<!-- Modal -->
<div class="modal fade" id="alter_msg" tabindex="-1" role="dialog"
aria-labelledby="alter_msg_Label">
    <div class="modal-dialog modal-sm" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close"

```



```

data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>

        <h4 class="modal-title" id="alter_msg_Label">修改信息
</h4>
    </div>
    <div class="modal-body">
        <form method="post" id="changeInfo_form">
            <div class="form-group">
                <label for="name">姓名</label>

                <input type="text" class="form-control"
id="name" placeholder="" name="name"
value="${sessionScope.loginUser.name}">
            </div>
            <div class="form-group">
                <label for="phone">电话</label>

                <input type="tel" class="form-control"
id="phone" placeholder="" name="phone"
value="${sessionScope.loginUser.phone}">
            </div>
            <div class="form-group">
                <label>部门</label>

                <select class="form-control"
department_option id="department" name="department"> </select>
            </div>
        </form>
        <span id="changeInfoPrompt">&nbsp;</span>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-default"
data-dismiss="modal" id="close">关闭</button>

        <button type="button" class="btn btn-primary"
id="save_change_2">保存更改</button>
    </div>
</div>
</div>
</div>
</div>
</div>

```

```
</body>
</html>
```

6. 员工管理及查询界面 employee.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
  <%@include file="/pages/common/head.jsp"%>
  <title>设备信息</title>
  <style type="text/css">
    .container-fluid {
      margin: 20px auto;
    }
    .table {
      margin: 0 auto;
      width: 900px;
    }
    .modal-dialog {
      width: 460px;
    }
    .form-control {
      width: 280px;
    }
    img {
      height: 70px;
      width: 80px;
    }
    .prompt {
      display: block;
      text-align: center;
    }
  </style>

  <script type="text/javascript">

    $(function () {

      /* 页面完全加载后才出 alert 弹窗*/

      window.onload = function() {
        if (${not empty sessionScope.msg}) {
          alert("${sessionScope.msg}");
        }
      }
    }
  </script>
</head>
</html>
```

```

    });

    var prompt = $(".prompt");

    $("#add_equipment_btn, #alter_equipment_btn").click(function ()
{
    $("#id, #name, #specification, #buyDate, #price, #imgPath,
#position, #manager").val("");
    prompt.css('visibility', 'hidden');
});
    $("#id, #new_id").focus(function () {
        prompt.css('visibility', 'hidden');
    });

    $(".submit_btn").click(function () {
        var menthod = $(this).val();
        var id;
        //console.log(menthod);
        if (menthod === "add") {
            id = $("#id").val();
            if (id === "") {
                console.log(id)

                prompt.text("设备编号不得为空");

                prompt.css('color', 'red');
            } else if (fileIsLegal != null && fileIsLegal === false)
{

                prompt.text("文件不合法");

                prompt.css('color', 'red');
            } else {
                $("#add_form").submit();
            }
        } else if (menthod === "modify") {
            id = $("#new_id").val();
            if (id === "") {

                prompt.text("设备编号不得为空");

                prompt.css('color', 'red');
            } else if (fileIsLegal != null && fileIsLegal === false)
{

                prompt.text("文件不合法");

                prompt.css('color', 'red');
            }
        }
    });
}

```

```

        } else {
            $("#modify_form").submit();
        }
    }
    prompt.css('visibility', 'visible');
});

$("#a.delete").click(function () {
    return confirm("是否确定删除编号为【" +
$(this).parent().parent().find("td").eq(1).text() + "】的设备？")
});

$("#alter_equipment").on('show.bs.modal', function (event) {
    var btn = $(event.relatedTarget);
    var id = btn.data("id");
    var name = btn.data("name");
    var specification = btn.data("specification");
    var price = btn.data("price");
    var buyDate = btn.data("buydate");
    var position = btn.data("position");
    var imgPath = btn.data("imgpath");
    var manager = btn.data("manager");
    console.log(imgPath + " " + id + " " + name + " " +
specification + " " + buyDate + " " + position + " " + price);
    var modal = $(this); //Update the modal's content.
    modal.find("#oldId").val(id);
    modal.find("#new_id").val(id);
    modal.find("#new_name").val(name);
    modal.find("#new_specification").val(specification);
    modal.find("#new_price").val(price);
    modal.find("#new_buyDate").val(buyDate);
    modal.find("#new_position").val(position);
    modal.find("#new_manager").val(manager);
    modal.find("#originalImgPath").val(imgPath);
    })
    })
</script>
</head>
<body class="body">
<c:if test="${not empty sessionScope.msg}">
    <c:set var="msg" value="" scope="session"> </c:set>
</c:if>

```

```

<div class="container-fluid">
  <div class="row-fluid">
    <div class="span12">
      <ul class="nav nav-tabs">
        <li class="active">
          <a href="equipmentServlet?action=showAll"><strong>设备
</strong></a>
        </li>
        <li>
          <a href="departmentServlet?action=showAll"><strong>部门
</strong></a>
        </li>
        <c:if test="${sessionScope.loginUser.userType == '系统管理员
'}">
          <li>
            <a href="employeeServlet?action=showAll"><strong>员工
</strong></a>
          </li>
        </c:if>
        <li class="dropdown pull-right">
          <a class="dropdown-toggle" data-toggle="dropdown"
href="#">欢迎您! ${sessionScope.loginUser.name}! </a>
          <ul class="dropdown-menu">
            <li>
              <a href="pages/user/myInfo.jsp">我的信息</a>
            </li>
            <li class="divider"></li>
            <li>
              <a href="userServlet?action=logout">注销</a>
            </li>
          </ul>
        </li>
      </ul>

      <h1 style="text-align: center; margin-top: 40px">
        <strong>设备信息表</strong>

```

```

</h1>
<br/>

<!-- 搜索框和添加按钮 -->

<div class="<!-- col-xs-offset-4 -->" style="margin: 0 auto;
width: 470px;">
    <div class="row" style="margin-bottom: 5px; padding-left:
28px">

        <c:if test="${sessionScope.loginUser.userType == '系统管
理员'}"><div class="col-xs-9"></c:if>

        <c:if test="${sessionScope.loginUser.userType == '普通员
工'}"><div class="col-xs-11"></c:if>

            <div class="input-group">
                <span class="input-group-btn">
                    <select class="form-control" style="width:
106px;" id="select">

                        <option value="all">全部</option>

                        <option value="id">设备编号</option>

                        <option value="name">设备名称</option>

                        <option value="specification">设备规格
</option>

                        <option value="buyDate">购入年份</option>

                        <option value="price">价格</option>

                        <option value="position">存放位置
</option>

                        <option value="manager">负责人</option>

                    </select>
                </span>
                <input type="text" class="form-control"
id="search" placeholder="请输入要搜索的信息" >

                <span class="input-group-btn">

```

```

        <button class="btn btn-primary" type="button"
id="inquire" value="equipmentServlet" onclick="queryByCondition()">查询
</button>

        </span>
    </div>
</div>

<c:if test="${sessionScope.loginUser.userType == '系统管
理员'}">

        <div class="col-xs-3">
            <span class="input-group">
                <input class="btn btn-success" type="submit"
id="add_equipment_btn" value="添加" data-toggle="modal"
data-target="#add_equipment">
            </span>
        </div>
    </c:if>
</div>
</div>

<!-- Modal 响应添加按钮-->

    <div class="modal fade" id="add_equipment" tabindex="-1"
role="dialog" aria-labelledby="add_equipment_Label">
        <div class="modal-dialog modal-lg" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <button type="button" class="close"
data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>
                    <h4 class="modal-title"
id="add_equipment_Label">添加设备信息</h4>
                </div>
                <div class="modal-body">
                    <form
action="equipmentServlet?action=addEquipment" method="post"
enctype="multipart/form-data" class="form-horizontal" id="add_form">
                        <div class="form-group">
                            <label for="id" class="control-label

```

```

col-sm-3">编&nbsp;&nbsp;&nbsp;号</label>

        <div class="col-sm-9">
            <input type="text"

class="form-control" id="id" placeholder="请输入编号" name="id" required>

        </div>
    </div>
    <div class="form-group" >
        <label for="name" class="control-label

col-sm-3">名&nbsp;&nbsp;&nbsp;称</label>

        <div class="col-sm-9">
            <input type="text"

class="form-control" id="name" placeholder="请输入名称" name="name">

        </div>
    </div>
    <div class="form-group">
        <label for="specification"

class="control-label col-sm-3">规&nbsp;&nbsp;&nbsp;格</label>

        <div class="col-sm-9">
            <input type="text"

class="form-control" id="specification" placeholder="请输入规格"

name="specification">

        </div>
    </div>
    <div class="form-group">
        <label for="buyDate" class="control-label

col-sm-3">购入日期</label>

        <div class="col-sm-9">
            <input type="date"

class="form-control" id="buyDate" name="buyDate">

        </div>
    </div>
    <div class="form-group">
        <label for="price" class="control-label

col-sm-3">价&nbsp;&nbsp;&nbsp;格</label>

        <div class="col-sm-9">
            <input type="text"

```



```

class="form-control" id="price" placeholder="请输入价格" name="price">
    </div>
</div>
<div class="form-group" >
    <label class="control-label col-sm-3">图
    &ensp;&ensp;片</label>
    <div class="col-sm-9">
        <input type="file" id="imgPath"
name="imgPath" onchange="checkFileSuffix(this.value)">
    </div>
</div>
<div class="form-group">
    <label for="position"
class="control-label col-sm-3">存放位置</label>
    <div class="col-sm-9">
        <input type="text"
class="form-control" id="position" placeholder="请输入存放位置"
name="position">
    </div>
</div>
<div class="form-group">
    <label for="manager" class="control-label
col-sm-3">负&ensp;责&ensp;人</label>
    <div class="col-sm-9">
        <input type="text"
class="form-control" id="manager" placeholder="请输入负责人"
name="manager">
    </div>
</div>
</form>
<span class="prompt">&nbsp;</span>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-default"
data-dismiss="modal">关闭</button>
    <button type="button" class="btn btn-primary

```

```

submit_btn" value="add">提交</button>

    </div>
</div>
</div>
</div>

<table class="table table-hover table-bordered">
    <thead>
    <tr>

        <th width="90px">图片</th>

        <th>编号</th>

        <th>名称</th>

        <th>规格</th>

        <th>购入日期</th>

        <th>价格</th>

        <th>存放位置</th>

        <th>负责人</th>

        <c:if test="${sessionScope.loginUser.userType == '系统管
理员'}">

            <th>操作</th>

        </c:if>
    </tr>
    </thead>
    <tbody>
    <c:forEach items="${sessionScope.allEquipments}"
var="equipment">
        <tr>
            <td width="90px">
                <c:if test="${not empty equipment.imgPath}">
                    
                </c:if>
            </td>

```

```

        <td> ${equipment.id} </td>
        <td> ${equipment.name} </td>
        <td> ${equipment.specification} </td>
        <td> ${equipment.buyDate} </td>
        <td> ${equipment.price} </td>
        <td> ${equipment.position} </td>
        <td> ${equipment.manager} </td>

        <c:if test="${sessionScope.loginUser.userType == '系统管理员'}">

            <td>
                <a id="alter_equipment_btn" data-toggle="modal"
data-target="#alter_equipment" data-id="${equipment.id}"
data-name="${equipment.name}"
data-specification="${equipment.specification}"
data-price="${equipment.price}"
data-buyDate="${equipment.buyDate}"
data-position="${equipment.position}" data-imgPath="${equipment.imgPath}"
data-manager="${equipment.manager}">修改</a>

                <a class="delete"
href="equipmentServlet?action=delete&id=${equipment.id}" >删除</a>

            </td>
        </c:if>
    </tr>
</c:forEach>
</tbody>
</table>
</div>
</div>
</div>

<!-- Modal 响应修改按钮-->

<div class="modal fade" id="alter_equipment" tabindex="-1"
role="dialog" aria-labelledby="alter_equipment_Label">
    <div class="modal-dialog modal-lg" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close"
data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>

```

```

        <h4 class="modal-title" id="alter_equipment_Label">
修改设备信息</h4>

        </div>
        <div class="modal-body">
            <form
action="equipmentServlet?action=changeEquipmentInfo" method="post"
enctype="multipart/form-data" class="form-horizontal" id="modify_form">
                <input type="hidden" id="oldId" name="oldId"
value="" >
                <input type="hidden" id="originalImgPath"
name="originalImgPath" value="" >

                <div class="form-group">
                    <label for="id" class="control-label
col-sm-3">编&emsp;&emsp;号</label>

                    <div class="col-sm-9">
                        <input type="text" class="form-control"
id="new_id" placeholder="请输入编号" name="id">
                    </div>
                </div>
                <div class="form-group" >
                    <label for="name" class="control-label
col-sm-3">名&emsp;&emsp;称</label>

                    <div class="col-sm-9">
                        <input type="text" class="form-control"
id="new_name" placeholder="请输入名称" name="name">
                    </div>
                </div>
                <div class="form-group">
                    <label for="specification"
class="control-label col-sm-3">规&emsp;&emsp;格</label>

                    <div class="col-sm-9">
                        <input type="text" class="form-control"
id="new_specification" placeholder="请输入规格" name="specification">
                    </div>
                </div>
            </div>
        </div>
    </div>

```



```

id="new_manager" placeholder="请输入负责人" name="manager">

        </div>
    </div>
</form>
<span class="prompt">&nbsp;</span>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-default"
data-dismiss="modal">关闭</button>

    <button type="button" class="btn btn-primary
submit_btn" value="modify">提交</button>

</div>
</div>
</div>
</div>
</body>
</html>

```

7. 部门管理及查询界面

```

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <%@include file="/pages/common/head.jsp"%>

    <title>部门信息</title>

    <style type="text/css">
        .container-fluid {
            margin: 20px auto;
        }
        .table {
            width: 500px;
            margin: 0 auto;
        }
        .prompt {
            display: block;
            text-align: center;
        }
    </style>

    <script type="text/javascript">

```

```

$(function () {

    /* 页面完全加载后才出 alert 弹窗*/

    window.onload = function() {
        if ({not empty sessionScope.msg}) {
            alert("${sessionScope.msg}");
        }
    };

    var prompt = $(".prompt");
    $("#add_department_btn, #alter_department_btn").click(function

() {

        prompt.css('visibility', 'hidden');
    });
    $(".submit_btn").click(function () {
        var menthod = $(this).val();
        //alert(menthod);
        var id, name;
        if (menthod === "add") {
            id = $("#id").val();
            name = $("#name").val();
            if (id === "" || name === "") {

                prompt.text("部门代号和名称不得为空!");

                prompt.css('color', 'red');
            } else {
                $("#add_form").submit();
            }
        } else if (menthod === "modify") {
            id = $("#new_id").val();
            name = $("#new_name").val();
            if (id === "" || name === "") {

                prompt.text("部门代号和名称不得为空!");

                prompt.css('color', 'red');
            } else {
                $("#modify_form").submit();
            }
        }
        prompt.css('visibility', 'visible');
    });
    $("#a.delete").click(function () {

```

```

        return confirm("是否确定删除【" +
$(this).parent().parent().find("td").eq(1).text() + "】 ?")
    });
    $("#alter_department").on('show.bs.modal', function (event) {
        var btn = $(event.relatedTarget); // Button that triggered the
modal
        var id = btn.data("id"); // Extract info from data-* attributes
        var name = btn.data("name");
        var manager = btn.data("manager");
        //console.log(id + " " + name + " " + manager);
        var modal = $(this); //Update the modal's content.
        modal.find("#oldId").val(id);
        modal.find("#new_id").val(id);
        modal.find("#new_name").val(name);
        modal.find("#new_manager").val(manager);
    });
})

</script>
</head>
<body class="body">
<c:if test="${not empty sessionScope.msg}">
    <c:set var="msg" value="" scope="session"> </c:set>
</c:if>
<div class="container-fluid">
    <div class="row-fluid">
        <div class="span12">
            <ul class="nav nav-tabs">
                <li>
                    <a href="equipmentServlet?action=showAll"><strong>设备
</strong></a>
                </li>
                <li class="active">
                    <a href="departmentServlet?action=showAll"><strong>部门
</strong></a>
                </li>
                <c:if test="${sessionScope.loginUser.userType == '系统管理员'
' }">
                    <li>

```



```

        <a href="employeeServlet?action=showAll"><strong>员工
</strong></a>
    </li>
</c:if>
<li class="dropdown pull-right">
    <a class="dropdown-toggle" data-toggle="dropdown"
href="#">欢迎您! ${sessionScope.loginUser.name}! </a>
    <ul class="dropdown-menu">
        <li>
            <a href="pages/user/myInfo.jsp">我的信息</a>
        </li>
        <li class="divider"></li>
        <li>
            <a href="userServlet?action=logout">注销</a>
        </li>
    </ul>
</li>
</ul>

<h1 style="text-align: center; margin-top: 40px">
    <strong>部门信息表</strong>
</h1>
<br/>
<!-- 搜索框和添加按钮 -->
<div class="col-xs-offset-2" style="margin: 0 auto;
width: 470px">
    <div class="row" style="margin-bottom: 5px; padding-left:
28px">
        <c:if test="${sessionScope.loginUser.userType == '系统管
理员'}"><div class="col-xs-9"></c:if>
        <c:if test="${sessionScope.loginUser.userType == '普通员
工'}"><div class="col-xs-11"></c:if>
            <div class="input-group">
                <span class="input-group-btn">

```

```

        <select class="form-control" style="width:
106px" id="select">

            <option value="all">全部</option>

            <option value="id">部门代号</option>

            <option value="name">部门名称</option>

            <option value="manager">部门主管

        </option>

    </select>
</span>
<input type="text" class=" form-control"
placeholder="请输入要搜索的信息" id="search" >

    <span class="input-group-btn">
        <button class="btn btn-primary" type="button"
id="inquire" value="departmentServlet" onclick="queryByCondition()">查询
    </button>

    </span>
</div>
</div>

<c:if test="${sessionScope.loginUser.userType == '系统管
理员'}">

    <div class="col-xs-3">
        <span class="input-group">
            <input class="btn btn-success" type="submit"
id="add_department_btn" value="添加" data-toggle="modal"
data-target="#add_department">
        </span>
    </div>
</c:if>
</div>
</div>

<!-- Modal 响应添加按钮-->

    <div class="modal fade" id="add_department" tabindex="-1"
role="dialog" aria-labelledby="add_department_Label">

```

```

        <div class="modal-dialog modal-sm" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <button type="button" class="close"
data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>
                    <h4 class="modal-title"
id="add_department_Label">添加部门信息</h4>
                </div>
                <div class="modal-body">
                    <form action="departmentServlet" method="post"
id="add_form">
                        <input type="hidden" name="action"
value="add">
                        <div class="form-group">
                            <label for="id">部门代号</label>
                            <input type="text" class="form-control"
id="id" placeholder="请输入部门代号" name="id">
                        </div>
                        <div class="form-group">
                            <label for="name">部门名称</label>
                            <input type="text" class="form-control"
id="name" placeholder="请输入部门名称" name="name">
                        </div>
                        <div class="form-group">
                            <label for="manager">部门主管</label>
                            <input type="text" class="form-control"
id="manager" placeholder="请输入部门主管工号" name="manager">
                        </div>
                    </form>
                    <span class="prompt">&nbsp;</span>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-default"
data-dismiss="modal">关闭</button>
                    <button type="button" class="btn btn-primary

```

```

submit_btn" value="add">提交</button>

    </div>
</div>
</div>
</div>
</div>

<table class="table table-hover table-bordered">
    <thead>
        <tr>
            <th> 部门代号 </th>

            <th> 部门名称 </th>

            <th> 部门主管 </th>

            <c:if test="${sessionScope.loginUser.userType == '系
统管理员'}">

                <th> 操作 </th>
            </c:if>
        </tr>
    </thead>
    <tbody>
        <c:forEach items="${sessionScope.departments}"
var="department">
            <tr>
                <td> ${department.id} </td>
                <td> ${department.name} </td>
                <td> ${department.manager} </td>
                <c:if test="${sessionScope.loginUser.userType ==
'系统管理员'}">
                    <td>
                        <a id="alter_department_btn"
data-toggle="modal" data-target="#alter_department"
data-id="${department.id}" data-name="${department.name}"
data-manager="${department.manager}">修改</a>

                        <a class="delete"

```

```

href="departmentServlet?action=delete&id=${department.id}" >删除</a>

        </td>
    </c:if>
</tr>
</c:forEach>

<!-- Modal 响应修改事件-->

<div class="modal fade" id="alter_department"
tabindex="-1" role="dialog" aria-labelledby="alter_department_Label">
    <div class="modal-dialog modal-sm" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close"
data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>
                <h4 class="modal-title"
id="alter_department_Label">修改部门信息</h4>
            </div>
            <div class="modal-body">
                <form action="departmentServlet"
method="post" id="modify_form">
                    <input type="hidden" name="action"
value="modify">
                    <input type="hidden" id="oldId"
name="oldId" value="">
                    <div class="form-group">
                        <label for="new_id">部门代号
</label>
                        <input type="text"
class="form-control" id="new_id" placeholder="请输入部门代号" name="id">
                    </div>
                    <div class="form-group">
                        <label for="new_name">部门名称
</label>
                        <input type="text"
class="form-control" id="new_name" placeholder="请输入部门名称" name="name">
                    </div>
                    <div class="form-group">

```



```

        width: 900px;
    }
    .modal-dialog {
        width: 460px;
    }
    .form-control {
        width: 280px;
    }
    img {
        height: 70px;
        width: 80px;
    }
    .prompt {
        display: block;
        text-align: center;
    }
</style>
<script type="text/javascript">
    $(function () {

        /* 页面完全加载后才出 alert 弹窗*/

        window.onload = function() {
            if ({not empty sessionScope.msg}) {
                alert("${sessionScope.msg}");
            }
        };
        var prompt = $(".prompt");
        $("#add_equipment_btn, #alter_equipment_btn").click(function ()
{
            $("#id, #name, #specification, #buyDate, #price, #imgPath,
#position, #manager").val("");
            prompt.css('visibility', 'hidden');
        });
        $("#id, #new_id").focus(function () {
            prompt.css('visibility', 'hidden');
        });

        $(".submit_btn").click(function () {
            var method = $(this).val();
            var id;
            //console.log(method);
            if (method === "add") {
                id = $("#id").val();
                if (id === "") {

```

```

        console.log(id)

        prompt.text("设备编号不得为空");

        prompt.css('color', 'red');
    } else if (fileIsLegal != null && fileIsLegal === false)
    {

        prompt.text("文件不合法");

        prompt.css('color', 'red');
    } else {
        $("#add_form").submit();
    }
} else if (menthod === "modify") {
    id = $("#new_id").val();
    if (id === "") {

        prompt.text("设备编号不得为空");

        prompt.css('color', 'red');
    } else if (fileIsLegal != null && fileIsLegal === false)
    {

        prompt.text("文件不合法");

        prompt.css('color', 'red');
    } else {
        $("#modify_form").submit();
    }
}
prompt.css('visibility','visible');
});

$("a.delete").click(function () {

    return confirm("是否确定删除编号为【" +

$(this).parent().parent().find("td").eq(1).text() + "】的设备？")

});

$("#alter_equipment").on('show.bs.modal', function (event) {
    var btn = $(event.relatedTarget);
    var id = btn.data("id");
    var name = btn.data("name");
    var specification = btn.data("specification");
    var price = btn.data("price");
    var buyDate = btn.data("buydate");

```



```

        var position = btn.data("position");
        var imgPath = btn.data("imgpath");
        var manager = btn.data("manager");
        console.log(imgPath + " " + id + " " + name + " " +
specification + " " + buyDate + " " + position + " " + price);

        var modal = $(this); //Update the modal's content.
        modal.find("#oldId").val(id);
        modal.find("#new_id").val(id);
        modal.find("#new_name").val(name);
        modal.find("#new_specification").val(specification);
        modal.find("#new_price").val(price);
        modal.find("#new_buyDate").val(buyDate);
        modal.find("#new_position").val(position);
        modal.find("#new_manager").val(manager);

        modal.find("#originalImgPath").val(imgPath);
    })
})
</script>
</head>
<body class="body">
<c:if test="${not empty sessionScope.msg}">
    <c:set var="msg" value="" scope="session"> </c:set>
</c:if>
<div class="container-fluid">
    <div class="row-fluid">
        <div class="span12">
            <ul class="nav nav-tabs">
                <li class="active">
                    <a href="equipmentServlet?action=showAll"><strong>设备
</strong></a>
                </li>
                <li>
                    <a href="departmentServlet?action=showAll"><strong>部门
</strong></a>
                </li>
                <c:if test="${sessionScope.loginUser.userType == '系统管理员
'}">
                    <li>

```

```

        <a href="employeeServlet?action=showAll"><strong>员工
</strong></a>
    </li>
</c:if>
<li class="dropdown pull-right">
    <a class="dropdown-toggle" data-toggle="dropdown"
href="#">欢迎您! ${sessionScope.loginUser.name}! </a>
    <ul class="dropdown-menu">
        <li>
            <a href="pages/user/myInfo.jsp">我的信息</a>
        </li>
        <li class="divider"></li>
        <li>
            <a href="userServlet?action=logout">注销</a>
        </li>
    </ul>
</li>
</ul>

<h1 style="text-align: center; margin-top: 40px">
    <strong>设备信息表</strong>
</h1>
<br/>
<!-- 搜索框和添加按钮 -->
<div class="col-xs-offset-4" style="margin: 0 auto;
width: 470px;">
    <div class="row" style="margin-bottom: 5px; padding-left:
28px">
        <c:if test="${sessionScope.loginUser.userType == '系统管
理员'}"><div class="col-xs-9"></c:if>

        <c:if test="${sessionScope.loginUser.userType == '普通员
工'}"><div class="col-xs-11"></c:if>
            <div class="input-group">
                <span class="input-group-btn">

```

```

<select class="form-control" style="width:
106px;" id="select">
    <option value="all">全部</option>
    <option value="id">设备编号</option>
    <option value="name">设备名称</option>
    <option value="specification">设备规格
</option>
    <option value="buyDate">购入年份</option>
    <option value="price">价格</option>
    <option value="position">存放位置
</option>
    <option value="manager">负责人</option>
</select>
</span>
<input type="text" class=" form-control"
id="search" placeholder="请输入要搜索的信息" >
    <span class="input-group-btn">
        <button class="btn btn-primary" type="button"
id="inquire" value="equipmentServlet" onclick="queryByCondition()">查询
</button>
    </span>
</div>
</div>
<c:if test="${sessionScope.loginUser.userType == '系统管
理员'}">
    <div class="col-xs-3">
        <span class="input-group">
            <input class="btn btn-success" type="submit"
id="add_equipment_btn" value="添加" data-toggle="modal"
data-target="#add_equipment">
        </span>

```



```

class="control-label col-sm-3">规格</label>
        <div class="col-sm-9">
            <input type="text"
class="form-control" id="specification" placeholder="请输入规格"
name="specification">
        </div>
    </div>
    <div class="form-group">
        <label for="buyDate" class="control-label
col-sm-3">购入日期</label>
        <div class="col-sm-9">
            <input type="date"
class="form-control" id="buyDate" name="buyDate">
        </div>
    </div>
    <div class="form-group">
        <label for="price" class="control-label
col-sm-3">价格</label>
        <div class="col-sm-9">
            <input type="text"
class="form-control" id="price" placeholder="请输入价格" name="price">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-sm-3">图
        &nbsp;&nbsp;&nbsp;</label>
        <div class="col-sm-9">
            <input type="file" id="imgPath"
name="imgPath" onchange="checkFileSuffix(this.value)">
        </div>
    </div>
    <div class="form-group">
        <label for="position"
class="control-label col-sm-3">存放位置</label>
        <div class="col-sm-9">
            <input type="text"

```

```

class="form-control" id="position" placeholder="请输入存放位置"
name="position">
    </div>
</div>
<div class="form-group">
    <label for="manager" class="control-label
col-sm-3">负&ensp;责&ensp;人</label>
    <div class="col-sm-9">
        <input type="text"
class="form-control" id="manager" placeholder="请输入负责人"
name="manager">
    </div>
</div>
</form>
<span class="prompt">&nbsp;</span>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-default"
data-dismiss="modal">关闭</button>
    <button type="button" class="btn btn-primary
submit_btn" value="add">提交</button>
</div>
</div>
</div>
</div>
<table class="table table-hover table-bordered">
    <thead>
    <tr>
        <th width="90px">图片</th>
        <th>编号</th>
        <th>名称</th>
        <th>规格</th>
        <th>购入日期</th>

```

```

<th>价格</th>

<th>存放位置</th>

<th>负责人</th>

<c:if test="${sessionScope.loginUser.userType == '系统管
理员'}">

    <th>操作</th>

</c:if>
</tr>
</thead>
<tbody>
<c:forEach items="${sessionScope.allEquipments}"
var="equipment">
    <tr>
        <td width="90px">
            <c:if test="${not empty equipment.imgPath}">
                
            </c:if>
        </td>
        <td> ${equipment.id} </td>
        <td> ${equipment.name} </td>
        <td> ${equipment.specification} </td>
        <td> ${equipment.buyDate} </td>
        <td> ${equipment.price} </td>
        <td> ${equipment.position} </td>
        <td> ${equipment.manager} </td>

        <c:if test="${sessionScope.loginUser.userType == '系
统管理员'}">
            <td>
                <a id="alter_equipment_btn" data-toggle="modal"
data-target="#alter_equipment" data-id="${equipment.id}"
data-name="${equipment.name}"
data-specification="${equipment.specification}"
data-price="${equipment.price}"
data-buyDate="${equipment.buyDate}"
data-position="${equipment.position}" data-imgPath="${equipment.imgPath}"

```



```

id="new_id" placeholder="请输入编号" name="id">

    </div>
</div>
<div class="form-group" >
    <label for="name" class="control-label
col-sm-3">名&emsp;&emsp;称</label>

    <div class="col-sm-9">
        <input type="text" class="form-control"
id="new_name" placeholder="请输入名称" name="name">

        </div>
    </div>
    <div class="form-group">
        <label for="specification"
class="control-label col-sm-3">规&emsp;&emsp;格</label>

        <div class="col-sm-9">
            <input type="text" class="form-control"
id="new_specification" placeholder="请输入规格" name="specification">

            </div>
        </div>
        <div class="form-group">
            <label for="buyDate" class="control-label
col-sm-3">购入日期</label>

            <div class="col-sm-9">
                <input type="date" class="form-control"
id="new_buyDate" name="buyDate">

                </div>
            </div>
            <div class="form-group">
                <label for="price" class="control-label
col-sm-3">价&emsp;&emsp;格</label>

                <div class="col-sm-9">
                    <input type="text" class="form-control"
id="new_price" placeholder="请输入价格" name="price">

                    </div>
                </div>
            <div class="form-group" >

```

```

        <label class="control-label col-sm-3">图
&emsp;&emsp;片</label>

        <div class="col-sm-9">
            <input type="file" id="new_imgPath"
name="imgPath" onchange="checkFileSuffix(this.value)">
        </div>
    </div>
    <div class="form-group">
        <label for="position" class="control-label
col-sm-3">存放位置</label>

        <div class="col-sm-9">
            <input type="text" class="form-control"
id="new_position" placeholder="请输入存放位置" name="position">
        </div>
    </div>
    <div class="form-group">
        <label for="manager" class="control-label
col-sm-3">负&ensp;责&ensp;人</label>

        <div class="col-sm-9">
            <input type="text" class="form-control"
id="new_manager" placeholder="请输入负责人" name="manager">
        </div>
    </div>
</form>
<span class="prompt">&nbsp;</span>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-default"
data-dismiss="modal">关闭</button>

    <button type="button" class="btn btn-primary
submit_btn" value="modify">提交</button>
</div>
</div>
</div>
</div>

```

```
</body>
</html>
```

9. myjs.js

```
// 通过 ajax 获取所有部门选项数据，添加到 <select> 中

function getDepartmentOptions() {
    $.getJSON("userServlet", "action=getDepartmentOptions", function
(data) {
        //console.log(data);
        if (data !== null) {
            for (var i in data) {
                for (var j = 0; j < data[i].length; j++) {
                    //console.log(data[i][j]["id"]);
                    $(".department_option").append("<option value=" +
data[i][j]["id"] + ">" + data[i][j]["name"] + "</option>")
                }
            }
        }
    })
}

// 根据条件查询

function queryByCondition() {
    var XXXServlet = $("#inquire").val();
    var condition = $("#select").val();
    var parameter = $("#search").val();
    location.href = XXXServlet + "?action=queryByCondition&condition=" +
condition + "&parameter=" + parameter;
}

// 判断上传的文件是否合法

var fileIsLegal = null;
function checkFileSuffix(filename) {
    console.log(filename);

    var flag = false; //状态

    var arr = ["jpg", "png", "jpeg", "gif"];

    //取出上传文件的 后缀名

    var suffix = filename.substr(filename.lastIndexOf(".") + 1);

    //循环比较
```

```

    console.log(suffix);
    for (var i = 0; i < arr.length; i++) {
        if (suffix === arr[i]) {

            flag = true; // 一旦找到合适的，立即退出循环

            break;
        }
    }

    // 条件判断

    if (!flag) {

        $(".prompt").text("文件不合法");

        $(".prompt").css('color', 'red');
        $(".prompt").css('visibility', 'visible');
        fileIsLegal = false;
    } else {
        $(".prompt").text("");
        $(".prompt").css('visibility', 'hidden');
        fileIsLegal = true;
    }
}

// 实现登录页面 remember me 功能

function getLoginCookie() {
    $.getJSON("userService", "action=getLoginCookie", function (data) {
        console.log(data);
        $("#id").val(data.id.value);
        $("#password").val(data.pwd.value);
    })
}

```

5.2 各项功能源码

1. 模型 model

(1) 部门实体类 Department.java

```

package com.lab.bean;

public class Department {
    private String id;
    private String name;
}

```

```

private String manager;
public Department() {
}
public Department(String id, String name, String manager) {
    this.id = id;
    this.name = name;
    this.manager = manager;
}
public String getId() {
    return id;
}
public void setId(String id) {
    this.id = id;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public String getManager() {
    return manager;
}
public void setManager(String manager) {
    this.manager = manager;
}
@Override
public String toString() {
    return "Department{" +
        "id=" + id +
        ", name='" + name + '\'' +
        ", manager='" + manager + '\'' +
        '}';
}
}

```

(2) 员工实体类 Employee.java

```

package com.lab.bean;
public class Employee {
    private String id;
    private String password;
    private String name;
    private String phone;
    private String userType;
}

```

```

private Integer department;
public Employee() {
}
public Employee(String id, String password, String name, String phone,
String userType, Integer department) {
    this.id = id;
    this.password = password;
    this.name = name;
    this.phone = phone;
    this.userType = userType;
    this.department = department;
}
public String getId() {
    return id;
}
public void setId(String id) {
    this.id = id;
}
public String getPassword() {
    return password;
}
public void setPassword(String password) {
    this.password = password;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public String getPhone() {
    return phone;
}
public void setPhone(String phone) {
    this.phone = phone;
}
public String getUserType() {
    return userType;
}
public void setUserType(String userType) {
    this.userType = userType;
}
public Integer getDepartment() {
    return department;
}

```

```

    }
    public void setDepartment(Integer department) {
        this.department = department;
    }
    @Override
    public String toString() {
        return "employee{" +
            "id='" + id + '\'' +
            ", password='" + password + '\'' +
            ", name='" + name + '\'' +
            ", phone='" + phone + '\'' +
            ", userType='" + userType + '\'' +
            ", department=" + department +
            '}';
    }
}

```

(3) 设备实体类 Equipment.java

```

package com.lab.bean;
public class Equipment {
    private String id;
    private String name;
    private String specification;
    private Double price;
    private String buyDate;
    private String position;
    private String imgPath;
    private String manager;
    public Equipment() {
    }
    public Equipment(String id, String name, String specification, Double
price, String buyDate, String position, String imgPath, String manager) {
        this.id = id;
        this.name = name;
        this.specification = specification;
        this.price = price;
        this.buyDate = buyDate;
        this.position = position;
        this.imgPath = imgPath;
        this.manager = manager;
    }
    public String getId() {
        return id;
    }
}

```

```

}
public void setId(String id) {
    this.id = id;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public String getSpecification() {
    return specification;
}
public void setSpecification(String specification) {
    this.specification = specification;
}
public Double getPrice() {
    return price;
}
public void setPrice(Double price) {
    this.price = price;
}
public String getBuyDate() {
    return buyDate;
}
public void setBuyDate(String buyDate) {
    this.buyDate = buyDate;
}
public String getPosition() {
    return position;
}
public void setPosition(String position) {
    this.position = position;
}
public String getImgPath() {
    return imgPath;
}
public void setImgPath(String imgPath) {
    this.imgPath = imgPath;
}
public String getManager() {
    return manager;
}
public void setManager(String manager) {

```



```

        this.manager = manager;
    }
    @Override
    public String toString() {
        return "Equipment{" +
            "id='" + id + '\'' +
            ", name='" + name + '\'' +
            ", specification='" + specification + '\'' +
            ", price=" + price +
            ", buyDate=" + buyDate +
            ", position='" + position + '\'' +
            ", imgPath='" + imgPath + '\'' +
            ", manager='" + manager + '\'' +
            '}';
    }
}

```

2. 持久层 Dao：与数据库进行交互

(1) BaseDao：对第三方 jar 包 Dbutils 进行一些方法的封装，方便后面 sql 语句直接调用执行。

BaseDao.java

```

package com.lab.dao.impl;

import com.lab.utils.JdbcUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.handlers.BeanHandler;
import org.apache.commons.dbutils.handlers.BeanListHandler;
import org.apache.commons.dbutils.handlers.ScalarHandler;

import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;

public abstract class BaseDao {

    private QueryRunner queryRunner = new QueryRunner();
    /**
     * 执行 insert / update / delete 语句
     * @param sql
     * @param args
     * @return 若返回-1，说明执行失败；返回其他表示影响的行数
     */
}

```

```

    */
    public int update(String sql, Object ...args) {
        Connection conn = JdbcUtils.getConnection();
        try {
            return queryRunner.update(conn, sql, args);
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            JdbcUtils.close(conn);
        }
        return -1;
    }

    /**
     * 查询返回一个javaBean的sql语句
     *
     * @param type
     * @param sql
     * @param args
     * @param <T>
     * @return
     */
    public <T> T queryForOne(Class<T> type, String sql, Object ...args) {
        Connection conn = JdbcUtils.getConnection();
        try {
            return queryRunner.query(conn, sql, new BeanHandler<T>(type),
args);
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            JdbcUtils.close(conn);
        }
        return null;
    }

    /**
     * 查询返回多个javaBean的sql语句, sql语句有参数
     *
     * @param type
     * @param sql
     * @param args
     * @param <T>
     * @return
     */

```

```

    public <T> List<T> queryForList(Class<T> type, String sql,
Object ...args) {
    Connection conn = JdbcUtils.getConnection();
    try {
        return queryRunner.query(conn, sql, new
BeanListHandler<T>(type), args);
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        JdbcUtils.close(conn);
    }
    return null;
}
/**
 * 查询返回多个javaBean的 sql 语句, sql 语句无参数
 * @param type
 * @param sql
 * @param <T>
 * @return
 */
public <T> List<T> queryForList(Class<T> type, String sql) {
    Connection conn = JdbcUtils.getConnection();
    try {
        return queryRunner.query(conn, sql, new
BeanListHandler<T>(type));
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        JdbcUtils.close(conn);
    }
    return null;
}
/**
 * 执行返回一行一列的 sql 语句
 * @param sql
 * @param args
 * @return
 */
public Object queryForSingleValue(String sql, Object ...args) {
    Connection conn = JdbcUtils.getConnection();
    try {

```

```

        return queryRunner.query(conn, sql, new ScalarHandler(), args);
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        JdbcUtils.close(conn);
    }
    return null;
}
}

```

(2) 部门: 接口 DepartmentDao / 实现类 DepartmentDaoImpl

DepartmentDao.java

```

package com.lab.dao;
import com.lab.bean.Department;
import java.util.List;
public interface DepartmentDao {
    public List<Department> queryAllDepartment();
    public int addDepartment(Department department);
    public int updateDepartment(Department department, String id);
    public int deleteDepartment(String id);
    public Department queryById(String id);
    public List<Department> queryByName(String name);
    public List<Department> queryByManager(String manager);
}

```

DepartmentDaoImpl.java

```

package com.lab.dao.impl;
import com.lab.bean.Department;
import com.lab.dao.DepartmentDao;
import java.util.List;
public class DepartmentDaoImpl extends baseDao implements DepartmentDao {
    @Override
    public List<Department> queryAllDepartment() {
        String sql = "select * from department";
        return queryForList(Department.class, sql);
    }
    @Override
    public int addDepartment(Department department) {
        if ("".equals(department.getManager())) {
            String sql = "insert into department(`id`,`name`) values(?,?)";
            return update(sql, department.getId(), department.getName());
        } else {
            String sql = "insert into department(`id`,`name`,`manager`) values(?,?,?)";
            return update(sql, department.getId(), department.getName(),

```

```

department.getManager());
    }
}
@Override
public int updateDepartment(Department department, String id) {
    String sql = "update department set id=?, name=?, manager=? where
id=?";
    return update(sql, department.getId(), department.getName(),
department.getManager(), id);
}
@Override
public int deleteDepartment(String id) {
    String sql = "delete from department where id=?";
    return update(sql, id);
}
@Override
public Department queryById(String id) {
    String sql = "select * from department where id=?";
    return queryForOne(Department.class, sql, id);
}
@Override
public List<Department> queryByManager(String manager) {
    String sql = "select * from department where manager=?";
    return queryForList(Department.class, sql, manager);
}
@Override
public List<Department> queryByName(String name) {
    String sql = "select * from department where name=?";
    return queryForList(Department.class, sql, name);
}
}
}

```

(3) 员工：接口 EmployeeDao / 实现类 EmployeeDaoImpl

Employee.java
<pre> package com.lab.dao; import com.lab.bean.Employee; import java.util.List; public interface EmployeeDao { public Employee queryByIdAndPassword(String id, String password); public Employee queryById(String id); public int saveEmployee(Employee employee); public List<Employee> queryAllEmployees(); public int updatePasswordById(String id, String newPassword); public int updateBasicInfoById(Employee employee); } </pre>

```

    public int updateAllInfoById(Employee employee, String oldId);
    public int deleteById(String id);
    public List<Employee> queryByName(String name);
    public List<Employee> queryByphone(String phone);
    public List<Employee> queryByUserType(String userType);
    public List<Employee> queryByDepartment(String department);
}

```

EmployeeDaoImpl.java

```

package com.lab.dao.impl;
import com.lab.bean.Department;
import com.lab.bean.Employee;
import com.lab.dao.EmployeeDao;
import java.util.List;
public class EmployeeDaoImpl extends baseDao implements EmployeeDao {
    @Override
    public Employee queryByIdAndPassword(String id, String password) {
        String sql = "select * from employee where id=? and password=?";
        return queryForOne(Employee.class, sql, id, password);
    }
    @Override
    public Employee queryById(String id) {
        String sql = "select * from employee where id=?";
        return queryForOne(Employee.class, sql, id);
    }
    @Override
    public int saveEmployee(Employee employee) {
        String sql = "insert into
employee(`id`,`password`,`name`,`phone`,`usertype`,`department`)
values(?,?,?,?,?,?)";
        return update(sql, employee.getId(), employee.getPassword(),
employee.getName(), employee.getPhone(), employee.getUserType(),
employee.getDepartment());
    }
    @Override
    public List<Employee> queryAllEmployees() {
        String sql = "select * from employee order by id";
        return queryForList(Employee.class, sql);
    }
    @Override
    public int updatePasswordById(String id, String newPassword) {
        String sql = "update employee set password=? where id=?";
        return update(sql, newPassword, id);
    }
}

```

```

@Override
public int updateBasicInfoById(Employee employee) {
    String sql = "update employee set name=?, phone=?, department=? where id=?";
    return update(sql, employee.getName(), employee.getPhone(), employee.getDepartment(), employee.getId());
}

public int updateAllInfoById(Employee employee, String oldId) {
    String sql = "update employee set id=?, name=?, phone=?, password=?, userType=?, department=? where id=?";
    return update(sql, employee.getId(), employee.getName(), employee.getPhone(), employee.getPassword(), employee.getUserType(), employee.getDepartment(), oldId);
}

@Override
public int deleteById(String id) {
    String sql = "delete from employee where id=?";
    return update(sql, id);
}

@Override
public List<Employee> queryByName(String name) {
    String sql = "select * from employee where name=?";
    return queryForList(Employee.class, sql, name);
}

@Override
public List<Employee> queryByphone(String phone) {
    String sql = "select * from employee where phone=?";
    return queryForList(Employee.class, sql, phone);
}

@Override
public List<Employee> queryByUserType(String userType) {
    String sql = "select * from employee where userType=?";
    return queryForList(Employee.class, sql, userType);
}

@Override
public List<Employee> queryByDepartment(String department) {
    String sql = "select * from employee where department=?";
    return queryForList(Employee.class, sql, department);
}
}

```

(4) 接口 EquipmentDao / 实现类 EquipmentDaoImpl

EquipmentDao.java

```

package com.lab.dao;
import com.lab.bean.Employee;
import com.lab.bean.Equipment;
import java.sql.Date;
import java.util.List;
public interface EquipmentDao {
    public List<Equipment> queryAllEquipments();
    public int addEquipment(Equipment equipment);
    public int deleteEquipmentById(String id);
    public int updateEquipmentInfo(Equipment equipment, String oldId);
    public Equipment queryById(String id);
    public List<Equipment> queryByName(String name);
    public List<Equipment> queryBySpecification(String specification);
    public List<Equipment> queryByPrice(String price);
    public List<Equipment> queryByBuydate(String year);
    public List<Equipment> queryByPosition(String position);
    public List<Equipment> queryByManager(String manager);
    public List<Equipment> queryByDepartment(String Department);
}

```

EquipmentDaoImpl.java

```

package com.lab.dao.impl;
import com.lab.bean.Equipment;
import com.lab.dao.EquipmentDao;
import java.util.List;
public class EquipmentDaoImpl extends baseDao implements EquipmentDao {
    @Override
    public List<Equipment> queryAllEquipments() {
        String sql = "select * from equipment";
        return queryForList(Equipment.class, sql);
    }
    @Override
    public int addEquipment(Equipment equipment) {
        String sql = "insert into
equipment(`id`,`name`,`specification`,`price`,`buydate`,`position`,`imgPa
th`,`manager`) values(?, ?, ?, ?, ?, ?, ?, ?)";
        return update(sql, equipment.getId(), equipment.getName(),
equipment.getSpecification(), equipment.getPrice(),
equipment.getBuyDate(), equipment.getPosition(), equipment.getImgPath(),
equipment.getManager());
    }
    @Override
    public int deleteEquipmentById(String id) {
        String sql = "delete from equipment where id = ?";
        return update(sql, id);
    }
}

```



```

    }
    @Override
    public int updateEquipmentInfo(Equipment equipment, String oldId) {
        String sql = "update equipment set
id=?,name=?,specification=?,price=?,buydate=?,position=?,imgPath=?,manager=? where id=?";
        return update(sql, equipment.getId(), equipment.getName(),
equipment.getSpecification(), equipment.getPrice(),
equipment.getBuyDate(), equipment.getPosition(), equipment.getImgPath(),
equipment.getManager(), oldId);
    }
    @Override
    public Equipment queryById(String id) {
        String sql = "select * from equipment where id=?";
        return queryForOne(Equipment.class, sql, id);
    }
    @Override
    public List<Equipment> queryByName(String name) {
        String sql = "select * from equipment where name=?";
        return queryForList(Equipment.class, sql, name);
    }
    @Override
    public List<Equipment> queryByBuydate(String year) {
        String sql = "select * from equipment where year(buydate)=?";
        return queryForList(Equipment.class, sql, year);
    }
    @Override
    public List<Equipment> queryByPosition(String position) {
        String sql = "select * from equipment where position=?";
        return queryForList(Equipment.class, sql, position);
    }
    @Override
    public List<Equipment> queryByManager(String manager) {
        String sql = "select * from equipment where manager=?";
        return queryForList(Equipment.class, sql, manager);
    }
    @Override
    public List<Equipment> queryByDepartment(String department) {
        String sql = "select * from equipment where department=?";
        return queryForList(Equipment.class, sql, department);
    }
    @Override
    public List<Equipment> queryBySpecification(String specification) {
        String sql = "select * from equipment where specification=?";
    }

```

```

        return queryForList(Equipment.class, sql, specification);
    }
    @Override
    public List<Equipment> queryByPrice(String price) {
        String sql = "select * from equipment where price=?";
        return queryForList(Equipment.class, sql, price);
    }
}

```

3. 业务层 Service

(1) 部门：接口 DepartmentService / 实现类 DepartmentServiceImpl

DepartmentService.java

```

package com.lab.service;
import com.lab.bean.Department;
import java.util.List;
public interface DepartmentService {
    public int add(Department department);
    public int delete(String id);
    public int update(Department department, String id);
    public List<Department> queryAll();

    public Department queryById(String id);
    public List<Department> queryByCondition(String condition, String
parameter);
}

```

DepartmentServiceImpl.java

```

package com.lab.service.impl;
import com.lab.bean.Department;
import com.lab.dao.DepartmentDao;
import com.lab.dao.impl.DepartmentDaoImpl;
import com.lab.service.DepartmentService;
import java.util.List;
public class DepartmentServiceImpl implements DepartmentService {
    private DepartmentDao departmentDao = new DepartmentDaoImpl();
    @Override
    public int add(Department department) {
        return departmentDao.addDepartment(department);
    }
    @Override
    public int delete(String id) {
        return departmentDao.deleteDepartment(id);
    }
    @Override
    public int update(Department department, String id) {

```

```

        return departmentDao.updateDepartment(department, id);
    }
    @Override
    public List<Department> queryAll() {
        return departmentDao.queryAllDepartment();
    }
    @Override
    public Department queryById(String id) {
        return departmentDao.queryById(id);
    }
    @Override
    public List<Department> queryByCondition(String condition, String
parameter) {
        List<Department> departments = null;
        if (condition.equals("name")) {
            departments = departmentDao.queryByName(parameter);
        } else if (condition.equals("manager")) {
            departments = departmentDao.queryByManager(parameter);
        }
        return departments;
    }
}

```

(2) 员工：接口 EmployeeService / 实现类 EmployeeServiceImpl

EmployeeService.java
<pre> package com.lab.service; import com.lab.bean.Employee; import java.util.List; public interface EmployeeService { public Employee login(String id, String password); public void regist(Employee employee); public Employee existsId(String id); public List<Employee> showAll(); public List<Employee> queryByCondition(String condition, String parameter); public int changePassword(String id, String newPassword); public int changeBasicInfo(Employee employee); public int changeAllInfo(Employee employee, String id); public int deleteEmployee(String id); } </pre>

EmployeeServiceImpl.java

```
package com.lab.service.impl;
import com.lab.bean.Department;
import com.lab.bean.Employee;
import com.lab.dao.EmployeeDao;
import com.lab.dao.impl.EmployeeDaoImpl;
import com.lab.service.EmployeeService;
import java.util.List;
public class EmployeeServiceImpl implements EmployeeService {
    private EmployeeDao employeeDao = new EmployeeDaoImpl();
    @Override
    public Employee login(String id, String password) {
        return employeeDao.queryByIdAndPassword(id,password);
    }
    @Override
    public void regist(Employee employee) {
        employeeDao.saveEmployee(employee);
    }
    @Override
    public Employee existsId(String id) {
        return employeeDao.queryById(id);
    }
    @Override
    public List<Employee> showAll() {
        return employeeDao.queryAllEmployees();
    }
    @Override
    public int changePassword(String id, String newPassword) {
        return employeeDao.updatePasswordById(id, newPassword);
    }
    @Override
    public int changeBasicInfo(Employee employee) {
        return employeeDao.updateBasicInfoById(employee);
    }
    @Override
    public int changeAllInfo(Employee employee, String oldId) {
        return employeeDao.updateAllInfoById(employee, oldId);
    }
    @Override
    public int deleteEmployee(String id) {
        return employeeDao.deleteById(id);
    }
    @Override
    public List<Employee> queryByCondition(String condition, String
```

```

parameter) {
    List<Employee> employees = null;
    if (condition.equals("name")) {
        employees = employeeDao.queryByName(parameter);
    } else if (condition.equals("phone")) {
        employees = employeeDao.queryByphone(parameter);
    } else if (condition.equals("userType")) {
        employees = employeeDao.queryByUserType(parameter);
    } else if (condition.equals("department")) {
        employees = employeeDao.queryByDepartment(parameter);
    }
    return employees;
}
}

```

(3) 设备：接口 EquipmentService / 实现类 EquipmentServiceImpl

EquipmentService.java

```

package com.lab.service;
import com.lab.bean.Equipment;
import java.util.List;
public interface EquipmentService {
    public List<Equipment> showAll();
    public int add(Equipment equipment);
    public int delete(String id);
    public int update(Equipment equipment, String oldId);
    public Equipment queryById(String id);
    public List<Equipment> queryByCondition(String condition, String
parameter);
}

```

EquipmentServiceImpl.java

```

package com.lab.service.impl;
import com.lab.bean.Employee;
import com.lab.bean.Equipment;
import com.lab.dao.EquipmentDao;
import com.lab.dao.impl.EquipmentDaoImpl;
import com.lab.service.EquipmentService;
import java.util.List;
public class EquipmentServiceImpl implements EquipmentService {
    private EquipmentDao equipmentDao = new EquipmentDaoImpl();
    @Override
    public List<Equipment> showAll() {
        return equipmentDao.queryAllEquipments();
    }
}

```

```

@Override
public int add(Equipment equipment) {
    return equipmentDao.addEquipment(equipment);
}
@Override
public int delete(String id) {
    return equipmentDao.deleteEquipmentById(id);
}
@Override
public int update(Equipment equipment, String oldId) {
    return equipmentDao.updateEquipmentInfo(equipment, oldId);
}
@Override
public Equipment queryById(String id) {
    return equipmentDao.queryById(id);
}
public List<Equipment> queryByCondition(String condition, String
parameter) {
    List<Equipment> equipments = null;
    if (condition.equals("name")) {
        equipments = equipmentDao.queryByName(parameter);
    } else if (condition.equals("specification")) {
        equipments = equipmentDao.queryBySpecification(parameter);
    } else if (condition.equals("price")) {
        equipments = equipmentDao.queryByPrice(parameter);
    } else if (condition.equals("position")) {
        equipments = equipmentDao.queryByPosition(parameter);
    } else if (condition.equals("manager")) {
        equipments = equipmentDao.queryByManager(parameter);
    } else if (condition.equals("buyDate")) {
        equipments = equipmentDao.queryByBuydate(parameter);
    }
    return equipments;
}
}

```

4. Controller 控制层

(1) BaseServlet.java: 重写 doGet/doPost 方法, 利用反射机制执行传递过来前端页面传递过来的 action 参数的方法, 是用于被其他 Servlet 继承的基础 Servlet。

BaseServlet.java

```

package com.lab.servlet;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;

```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.lang.reflect.Method;

public class BaseServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
        doPost(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
        req.setCharacterEncoding("utf-8");
        resp.setContentType("text/html;charset=UTF-8");

        String action = req.getParameter("action");
        try {
            Method method = this.getClass().getDeclaredMethod(action,
                HttpServletRequest.class, HttpServletResponse.class);
            method.invoke(this, req, resp);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

(2) DepartmentServlet.java: 处理部门增删改查功能

DepartmentServlet.java

```

package com.lab.servlet;

import com.lab.bean.Department;
import com.lab.service.DepartmentService;
import com.lab.service.impl.DepartmentServiceImpl;
import com.lab.service.impl.EmployeeServiceImpl;
import com.lab.utils.WebUtils;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;

```

```

import java.util.ArrayList;
import java.util.List;

@WebServlet(name = "DepartmentServlet", urlPatterns = "/departmentServlet")
public class DepartmentServlet extends BaseServlet {
    DepartmentService departmentService = new DepartmentServiceImpl();
    protected void showAll(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        List<Department> departments = departmentService.queryAll();
        HttpSession session = req.getSession();
        session.setAttribute("departments", departments);
        req.getRequestDispatcher("pages/client/department.jsp").forward(req,
resp);
    }

    protected void add(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        Department department =
WebUtils.copyParamToBean(req.getParameterMap(), new Department());
        if (departmentService.queryById(department.getId()) != null) {
            req.getSession().setAttribute("msg", "添加失败，部门代号已存在！");
        } else if (department.getManager() != null && new
EmployeeServiceImpl().existsId(department.getManager()) == null) {
            req.getSession().setAttribute("msg", "添加失败，部门管理员不存在！");
        } else {
            departmentService.add(department);
            req.getSession().setAttribute("msg", "添加成功！");
        }
        resp.sendRedirect(req.getContextPath() +
"/departmentServlet?action=showAll");
    }

    protected void delete(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        String id = req.getParameter("id");
        int result = departmentService.delete(id);
        if (result > 0) {

```



```

        req.getSession().setAttribute("msg", "删除成功!");
    } else {
        req.getSession().setAttribute("msg", "删除失败!");
    }
    resp.sendRedirect(req.getContextPath() +
"/departmentServlet?action=showAll");
}

    protected void modify(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        Department department =
WebUtils.copyParamToBean(req.getParameterMap(), new Department());
        String oldId = req.getParameter("oldId");
        if (!oldId.equals(department.getId()) &&
departmentService.queryById(department.getId()) != null) {
            req.getSession().setAttribute("msg", "修改失败，部门代号已存在！");
        } else if (department.getManager() != null && new
EmployeeServiceImpl().existsId(department.getManager()) == null) {
            req.getSession().setAttribute("msg", "修改失败，部门管理员不存在！");
        } else {
            departmentService.update(department, oldId);
            req.getSession().setAttribute("msg", "修改成功!");
        }
        resp.sendRedirect(req.getContextPath() +
"/departmentServlet?action=showAll");
    }

    protected void queryByCondition(HttpServletRequest req,
HttpServletResponse resp) throws ServletException, IOException {
        String condition = req.getParameter("condition");
        String parameter = req.getParameter("parameter");
        List<Department> departments = new ArrayList<>();
        if (condition.equals("all")) {
            showAll(req, resp);
            return;
        } else if (condition.equals("id")) {
            Department department = departmentService.queryById(parameter);

```

```

        if (department != null) {
            departments.add(department);
        }
    } else {
        departments = departmentService.queryByCondition(condition,
parameter);
    }
    req.getSession().setAttribute("departments", departments);
    resp.sendRedirect(req.getContextPath() +
"/pages/client/department.jsp");
    }
}

```

(3) EmployeeServlet.java: 处理员工增删改查功能

EmployeeServlet.java

```

package com.lab.servlet;

import com.google.gson.Gson;
import com.lab.bean.Employee;
import com.lab.service.EmployeeService;
import com.lab.service.impl.EmployeeServiceImpl;
import com.lab.utils.WebUtils;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

@WebServlet("/employeeServlet")
public class EmployeeServlet extends BaseServlet {
    EmployeeService employeeService = new EmployeeServiceImpl();

    /*处理系统管理员添加员工用户功能*/

    protected void addEmployee(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
        Employee employee = WebUtils.copyParamToBean(req.getParameterMap(),
new Employee());
        employeeService.regist(employee);
    }
}

```

```

        req.getSession().setAttribute("msg", "添加成功!");
        resp.sendRedirect(req.getContextPath() +
"/employeeServlet?action=showAll");
    }

    /*显示所有用户*/

    protected void showAll(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        List<Employee> employees = employeeService.showAll();
        req.getSession().setAttribute("employees", employees);
        req.getRequestDispatcher( "pages/client/employee.jsp").forward(req,resp);
    }

    /* 修改密码*/

    protected void changePassword(HttpServletRequest req,
HttpServletResponse resp) throws ServletException, IOException {
        String id = req.getParameter("id");
        String newPassword = req.getParameter("newPassword");
        int result = employeeService.changePassword(id, newPassword);
        Map<String, Object> resultMap = new HashMap<>();
        resultMap.put("result", result);
        resp.getWriter().write(new Gson().toJson(resultMap));
    }

    /* 用户修改自己的个人信息 */

    protected void changeBasicInfo(HttpServletRequest req,
HttpServletResponse resp) throws ServletException, IOException {
        Employee employee = WebUtils.copyParamToBean(req.getParameterMap(),
new Employee());
        int result = employeeService.changeBasicInfo(employee);
        Map<String, Object> resultMap = new HashMap<>();
        resultMap.put("result", result);
        resp.getWriter().write(new Gson().toJson(resultMap));
    }

    /*处理管理员修改员工用户信息功能*/

    protected void changeAllInfo(HttpServletRequest req,
HttpServletResponse resp) throws ServletException, IOException {
        String oldId = req.getParameter("oldId");
        Employee employee = WebUtils.copyParamToBean(req.getParameterMap(),
new Employee());

```

```

        int result = employeeService.changeAllInfo(employee, oldId);
        if (result > 0) {
            HttpSession session = req.getSession();

            session.setAttribute("msg", "修改成功!");

            Employee loginUser = (Employee)
session.getAttribute("loginUser");
            if (loginUser.getId().equals(oldId)) {
                session.setAttribute("loginUser", employee);
            }
        } else {
            req.getSession().setAttribute("msg", "修改失败!");
        }
        resp.sendRedirect(req.getContextPath() +
"/employeeServlet?action=showAll");
    }

    /*更新我的信息 (即当前登录用户) */

    protected void refreshLoginUserInfo(HttpServletRequest req,
HttpServletResponse resp) throws ServletException, IOException {
        String id = req.getParameter("id");
        Employee loginUser = employeeService.existsId(id);
        req.getSession().setAttribute("loginUser", loginUser);
        resp.sendRedirect(req.getContextPath() + "/pages/user/myInfo.jsp");
    }

    /*处理系统管理员删除员工用户功能*/

    protected void deleteEmployee(HttpServletRequest req,
HttpServletResponse resp) throws ServletException, IOException {
        String id = req.getParameter("id");
        int result = employeeService.deleteEmployee(id);
        if (result > 0) {
            req.getSession().setAttribute("msg", "删除成功!");
        } else {
            req.getSession().setAttribute("msg", "删除失败! 该员工为部门主管!");
        }
        resp.sendRedirect(req.getContextPath() +
"/employeeServlet?action=showAll");
    }

```

```

    /*按条件查找*/

    protected void queryByCondition(HttpServletRequest req,
    HttpServletResponse resp) throws ServletException, IOException {
        String condition = req.getParameter("condition");
        String parameter = req.getParameter("parameter");
        //System.out.println(condition + " " + parameter);
        List<Employee> employees = new ArrayList<>();
        if (condition.equals("all")) {
            showAll(req, resp);
            return;
        } else if (condition.equals("id")) {
            Employee employee = employeeService.existsId(parameter);
            if (employee != null) {
                employees.add(employee);
            }
        } else {
            employees = employeeService.queryByCondition(condition,
parameter);
        }
        req.getSession().setAttribute("employees", employees);
        resp.sendRedirect(req.getContextPath() +
"/pages/client/employee.jsp");
    }
}

```

(4) EquipmentServlet.java: 处理设备增删改查功能

EquipmentServlet.java

```

package com.lab.servlet;

import com.lab.bean.Equipment;
import com.lab.service.EquipmentService;
import com.lab.service.impl.EmployeeServiceImpl;
import com.lab.service.impl.EquipmentServiceImpl;
import com.lab.utils.FileUploadUtils;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

```

```

@WebServlet(name = "EquipmentServlet", urlPatterns = "/equipmentServlet")
public class EquipmentServlet extends BaseServlet {
    private EquipmentService equipmentService = new EquipmentServiceImpl();

    protected void showAll(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        List<Equipment> allEquipments = equipmentService.showAll();
        req.getSession().setAttribute("allEquipments", allEquipments);
        req.getRequestDispatcher( "pages/client/equipment.jsp").forward(req,
resp);
    }

    protected void addEquipment(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
        Equipment equipment = new FileUploadUtils(req).copyParamToBean(new
Equipment());
        if (equipmentService.queryById(equipment.getId()) != null) {
            req.getSession().setAttribute("msg", "添加失败！设备编号已存在！
");
        } else if (equipment.getManager() != null && new
EmployeeServiceImpl().existsId(equipment.getManager()) == null) {
            req.getSession().setAttribute("msg", "添加失败！设备负责人不存在！
");
        } else{
            equipmentService.add(equipment);
            req.getSession().setAttribute("msg", "添加成功！");
        }
        resp.sendRedirect(req.getContextPath() +
"/equipmentServlet?action=showAll");
    }

    protected void delete(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        String id = req.getParameter("id");
        Equipment equipment = equipmentService.queryById(id);
        int result = equipmentService.delete(id);
        FileUploadUtils.deleteFile(req, equipment.getImgPath());
        if (result > 0) {
            req.getSession().setAttribute("msg", "删除成功！");
        } else {

```

```

        req.getSession().setAttribute("msg", "删除失败!");
    }
    resp.sendRedirect(req.getContextPath() +
"/equipmentServlet?action=showAll");
}

    protected void changeEquipmentInfo(HttpServletRequest req,
    HttpServletResponse resp) throws ServletException, IOException {
        FileUploadUtils fileUpload = new FileUploadUtils(req);
        String oldId = fileUpload.getParameterValue("oldId");
        String originalImgPath =
fileUpload.getParameterValue("originalImgPath");
        Equipment equipment = fileUpload.copyParamToBean(new Equipment());
        if (!"".equals(originalImgPath) && equipment.getImgPath() == null)
    {
        equipment.setImgPath(originalImgPath);
    }
        int result = equipmentService.update(equipment, oldId);
        if (result > 0) {
            req.getSession().setAttribute("msg", "修改成功!");
        } else {
            req.getSession().setAttribute("msg", "修改失败!");
        }
        resp.sendRedirect(req.getContextPath() +
"/equipmentServlet?action=showAll");
    }

    /* 按条件查找*/

    protected void queryByCondition(HttpServletRequest req,
    HttpServletResponse resp) throws ServletException, IOException {
        String condition = req.getParameter("condition");
        String parameter = req.getParameter("parameter");
        List<Equipment> equipments = new ArrayList<>();
        if (condition.equals("all")) {
            showAll(req, resp);
            return;
        } else if (condition.equals("id")) {
            Equipment equipment = equipmentService.queryById(parameter);
            if (equipment != null) {
                equipments.add(equipment);
            }
        }
    }

```

```

    } else {
        equipments = equipmentService.queryByCondition(condition,
parameter);
    }
    req.getSession().setAttribute("allEquipments", equipments);
    resp.sendRedirect(req.getContextPath() +
"/pages/client/equipment.jsp");
    }
}

```

(5) UserServlet.java: 处理用户登录注册以及记住账号密码等功能

```

UserServlet.java

package com.lab.servlet;
import com.google.gson.Gson;
import com.lab.bean.Department;
import com.lab.bean.Employee;
import com.lab.service.DepartmentService;
import com.lab.service.EmployeeService;
import com.lab.service.impl.DepartmentServiceImpl;
import com.lab.service.impl.EmployeeServiceImpl;
import com.lab.utils.CookieUtils;
import com.lab.utils.WebUtils;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

@WebServlet(name = "UserServlet" ,urlPatterns = "/userServlet")
public class UserServlet extends BaseServlet {
    EmployeeService employeeService = new EmployeeServiceImpl();

    /*处理登录功能 */

    protected void login(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        String id = req.getParameter("id");
        String password = req.getParameter("password");
    }
}

```



```

String remember = req.getParameter("remember");
Cookie userId = null;
Cookie userPwd = null;
if (remember != null) {
    userId = CookieUtils.findCookie("userId", req.getCookies());
    userPwd = CookieUtils.findCookie("userPwd", req.getCookies());
    if (userId != null && userPwd != null) {
        userId.setValue(id);
        userPwd.setValue(password);
        resp.addCookie(userId);
        resp.addCookie(userPwd);
    } else if (userId == null && userPwd == null) {
        userId = new Cookie("userId", id);
        userPwd = new Cookie("userPwd", password);
        resp.addCookie(userId);
        resp.addCookie(userPwd);
    }
} else {
    userId = CookieUtils.findCookie("userId", req.getCookies());
    userPwd = CookieUtils.findCookie("userPwd", req.getCookies());
    if (userId != null && userPwd != null) {
        userId.setMaxAge(0); // 马上删除 cookie
        userPwd.setMaxAge(0);
        resp.addCookie(userId);
        resp.addCookie(userPwd);
    }
}
Employee loginUser = employeeService.login(id, password);
if (loginUser == null) {
    System.out.println("登录失败");

    req.setAttribute("msg", "账号或密码错误!");
    req.setAttribute("id", id);
    req.getRequestDispatcher("pages/user/login.jsp").forward(req,
resp);
} else {
    System.out.println("登录成功");
    req.getSession().setAttribute("loginUser", loginUser);
    resp.sendRedirect(req.getContextPath() +
"/equipmentServlet?action=showAll");
}

```

```

    }
    protected void getLoginCookie(HttpServletRequest req,
    HttpServletResponse resp) throws ServletException, IOException {
        Cookie userId = CookieUtils.findCookie("userId", req.getCookies());
        Cookie userPwd = CookieUtils.findCookie("userPwd",
    req.getCookies());
        if (userId != null && userPwd != null) {
            Map<String, Object> resultMap = new HashMap<>();
            resultMap.put("id", userId);
            resultMap.put("pwd", userPwd);
            resp.getWriter().write(new Gson().toJson(resultMap));
        }
    }

    /* 处理注销功能*/

    protected void logout(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        req.getSession().invalidate();
        resp.sendRedirect(req.getContextPath());
    }

    /*处理注册功能*/

    protected void registers(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException {
        Employee employee = WebUtils.copyParamToBean(req.getParameterMap(),
    new Employee());
        if (employeeService.existsId(employee.getId()) != null) {

            System.out.println("账号已存在");
        } else {
            employeeService.regist(employee);

            System.out.println("注册成功");

            // 注册成功顺带登录进入

            req.getSession().setAttribute("loginUser", employee);
            resp.sendRedirect(req.getContextPath() +
"/equipmentServlet?action=showAll");
        }
    }

    /* 检查指定账号是否存在*/

    protected void existsId(HttpServletRequest req, HttpServletResponse
    resp) throws ServletException, IOException {

```

```

        String id = req.getParameter("id");
        Employee existsId = employeeService.existsId(id);
        Map<String, Object> resultMap = new HashMap<>();
        resultMap.put("existsId", existsId);
        Gson gson = new Gson();
        String json = gson.toJson(resultMap);
        resp.getWriter().write(json);
    }

    /* ajax 获取所有部门选项用于部门多选框的选择 */

    protected void getDepartmentOptions(HttpServletRequest req,
        HttpServletResponse resp) throws ServletException, IOException {
        List<Department> departments = new
        DepartmentServiceImpl().queryAll();
        Map<String, Object> resultMap = new HashMap<>();
        resultMap.put("result", departments);
        resp.getWriter().write(new Gson().toJson(resultMap));
    }
}

```

5. Utils 工具类

(1) JdbcUtils.java: 创建 druid 数据库连接池, 封装 druid 连接池的获取连接、归还连接等方法

```

JdbcUtils.java

package com.lab.utils;

import com.alibaba.druid.pool.DruidDataSource;
import com.alibaba.druid.pool.DruidDataSourceFactory;

import java.io.InputStream;
import java.sql.Connection;
import java.util.Properties;

public class JdbcUtils {
    private static DruidDataSource dataSource;
    static {
        try {
            Properties properties = new Properties();

            // 读取 druid.properties 属性配置文件

            InputStream inputStream =
            JdbcUtils.class.getClassLoader().getResourceAsStream("druid.properties");
            properties.load(inputStream);

```

```

        // 创建数据库连接池

        dataSource = (DruidDataSource)
DruidDataSourceFactory.createDataSource(properties);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/*获取数据库连接池中的连接*/

public static Connection getConnection() {
    Connection conn = null;
    try {
        conn = dataSource.getConnection();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return conn;
}

/ * 关闭连接，释放资源*/

public static void close(Connection conn) {
    if (conn != null) {
        try {
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
}
}

```

(2) FileUploadUtils.java: 用于文件上传的一些相关功能处理

FileUploadUtils.java

```

package com.lab.utils;

import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileItemFactory;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;
import javax.servlet.http.HttpServletRequest;
import java.io.File;
import java.util.HashMap;
import java.util.List;

```

```

import java.util.Map;
public class FileUploadUtils {
    private Map<String, String> mp = new HashMap<>();

    /*用于 enctype="multipart/form-data" 的 form 表单，上传文件，并将所有表单
数据项封装到 map 中*/

    public FileUploadUtils(HttpServletRequest req) {
        try {
            req.setCharacterEncoding("utf-8");
            if (ServletFileUpload.isMultipartContent(req)) {
                FileItemFactory fileItemFactory = new DiskFileItemFactory();
                ServletFileUpload servletFileUpload = new
ServletFileUpload(fileItemFactory);
                List<FileItem> fileItemList =
servletFileUpload.parseRequest(req);

                // 循环判断，每一个表单项，是普通类型，还是上传的文件

                for (FileItem fileItem : fileItemList) {

                    if (fileItem.isFormField()) { // 普通表单项

                        mp.put(fileItem.getFieldName(),
fileItem.getString("UTF-8"));
                    } else {
                        if
("".equals(fileItem.getName()) || fileItem.getName() == null) {
                            continue;
                        }
                        String savePath =
req.getServletContext().getRealPath("/upload/");

                        System.out.println("图片文件保存路径： " + savePath);

                        File savePathDir = new File(savePath);
                        if (!savePathDir.exists()) {
                            savePathDir.mkdir();
                        }
                        String fileName = System.currentTimeMillis() + "_" +
fileItem.getName();

                        String filePath = savePath + fileName;
                        fileItem.write(new File(filePath));

                        System.out.println("文件上传成功");

                        String fileRelativePath = "upload/" + fileName;
                        mp.put(fileItem.getFieldName(), fileRelativePath);
                    }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

        }
    }
} catch (Exception e) {
    e.printStackTrace();
}
}

/* 将所有表单数据注入到 javabean 中*/

public <T> T copyParamToBean(T bean) {
    WebUtils.copyParamToBean(this.mp, bean);
    return bean;
}

/* 获取表单中指定参数名的参数值*/

public String getParameterValue(String parameterName) {
    for (Map.Entry<String, String>entry : this.mp.entrySet()) {
        if (entry.getKey().equals(parameterName)) {
            return entry.getValue();
        }
    }
    return null;
}

/* 删除设备的同时也把上传到服务器的图片文件删除*/

public static void deleteFile(HttpServletRequest req, String imgPath) {
    String savePath = req.getServletContext().getRealPath("/upload/");
    File[] files = new File(savePath).listFiles();
    if (files == null || "".equals(imgPath) || imgPath == null) {
        return;
    }
    for (File file : files) {
        if (imgPath.contains(file.getName())) {
            file.delete();

            System.out.println("图片已从服务器删除!");

            break;
        }
    }
}
}
}
}

```

(3) WebUtils.java: 封装第三方 jar 包 beanUtils 的一个方法，用于一次性将前端传过来的参数注入到 javabean 中。

WebUtils.java

```
package com.lab.utils;
import org.apache.commons.beanutils.BeanUtils;
import java.lang.reflect.Field;
import java.math.BigDecimal;
import java.util.Map;
public class WebUtils {

    /* 把 Map 中的值注入到对应的 JavaBean 属性中。*/

    public static <T> T copyParamToBean(Map value, T bean) {
        try {
            BeanUtils.populate(bean, value);

            System.out.println("注入之后 1: " + bean);

            Field[] fields = bean.getClass().getDeclaredFields();
            for (Field field : fields) {
                field.setAccessible(true);
                Object attributeValue = field.get(bean);
                if ("".equals(attributeValue)) {
                    field.set(bean, null);
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return bean;
    }
}
```

(4) CookieUtils.java: 处理查找指定名称的 Cookie 对象功能

CookieUtils.java

```
package com.lab.utils;
import javax.servlet.http.Cookie;
public class CookieUtils {

    /* 查找指定名称的 Cookie 对象*/

    public static Cookie findCookie(String name , Cookie[] cookies){
        if (name == null || cookies == null || cookies.length == 0) {
            return null;
        }
        for (Cookie cookie : cookies) {
            if (name.equals(cookie.getName())) {
```

```

        return cookie;
    }
}
return null;
}
}

```

6. Filter 过滤器

(1) NotLoggedInFilter.java: 处理未登录状态下的拦截功能

```

NotLoggedInFilter.java

package com.lab.filter;
import javax.servlet.*;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
@WebFilter(filterName = "NotLoggedInFilter",
    urlPatterns = {"/pages/client/*", "/pages/common/*",
        "/pages/user/myInfo.jsp", "/departmentServlet",
        "/employeeServlet", "/equipmentServlet"})
public class NotLoggedInFilter implements Filter {
    public void doFilter(ServletRequest req, ServletResponse resp,
        FilterChain chain) throws ServletException, IOException {
        HttpServletRequest request = (HttpServletRequest) req;
        Object loginUser = request.getSession().getAttribute("loginUser");
        if (loginUser == null) {
            request.getRequestDispatcher("/pages/user/login.jsp").forward(req, resp);
        } else {
            chain.doFilter(req, resp);
        }
    }
}

```

5.3 各项配置文件 (*.xml 等)

本节请贴上各项配置文件内容 (文本贴上, 不截图) 并简要说明。

1. web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"

```



```
version="4.0">

<welcome-file-list>
  <welcome-file>/pages/user/login.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

2. 数据库连接池配置文件 druid.properties

```
driverClassName=com.mysql.cj.jdbc.Driver
url=jdbc:mysql://127.0.0.1:3306/equipmentstorage?serverTimezone=UTC
username=root
password=123456
initialSize=5
maxActive=10
maxWait=3000
minIdle=3
```

5.4 数据库源码（数据库创建及定义 script）

本节请贴上（文本贴上，不截图）并简要说明相关数据库源码。

```
// 创建数据库
CREATE DATABASE EquipmentStorage DEFAULT CHARSET utf8mb4;
// 创建员工表
CREATE TABLE IF NOT EXISTS employee (
  `id` CHAR(10) PRIMARY KEY,
  `password` VARCHAR(32) NOT NULL,
  `name` VARCHAR(50),
  `phone` CHAR(13),
  `usertype` CHAR(8),
  `department` VARCHAR(20)
)
// 创建部门表
CREATE TABLE IF NOT EXISTS department (
  `id` VARCHAR(20) PRIMARY KEY,
  `name` VARCHAR(50) NOT NULL,
  `manager` CHAR(10),
  FOREIGN KEY(`manager`) REFERENCES employee(`id`)
)
```

```
// 给员工表添加外键约束
ALTER TABLE employee ADD FOREIGN KEY(department) REFERENCES department(`id`);
// 创建设备表
CREATE TABLE IF NOT EXISTS equipment (
    `id` VARCHAR(30) PRIMARY KEY,
    `name` VARCHAR(50),
    `specification` VARCHAR(120),
    `price` DECIMAL(11, 2),
    `buydate` DATE,
    `position` VARCHAR(120),
    `img_path` VARCHAR(200),
    `manager` CHAR(10),
    FOREIGN KEY(`manager`) REFERENCES employee(`id`)
)
```

6. 实验总结

本次实验是进行一个设备管理系统的独立开发，这是对这学期该课程所学内容的一个综合应用，通过这次实验，我对这学期所学内容和相关知识都有了更为深刻的认识，对代码层面的编写也更加熟练。在该项目中，架构模式上我采用了MVC+DAO进行开发，数据库连接池采用阿里Druid连接池，前端采用bootstrap样式库，技术上一些必要之处采用ajax进行页面的异步更新。通过对登录页面记住账号密码功能的实现，我对Cookie的相关知识更为了解，通过前端页面与Servlet的交互，我也更加明白了请求转发和请求重定向的区别之处和各自用途，还有对request域和session域的作用范围和生命周期等知识内容都更加熟悉。在这次项目中，我也逐渐培养出了对代码尽可能的进行优化的习惯，比如对一些可复用的代码块，可以抽取出来，直接在多处进行引入而不必每处都写一样的代码；还有利用反射机制直接执行前端传递到Servlet的action参数的方法，而不必在Servlet写一堆冗余的if/else去判断要执行Servlet的哪个方法。当然，在完成该项目的开发过程中也遇到了一些难题。比如在对员工部门或设备进行添加修改删除操作时，当提交完表单执行完成操作后，此时进行刷新会出现表单重复提交的情况。这个问题困扰了我一段时间，经过长时间的思考分析，最终发现这是因为我采用了请求转发进行增删改操作后的页面跳转，由于请求转发是一次请求且浏览器会记录表单的上一次提交的信息，故出现刷新会重复提交表单的问题。这时将请求转发改为请求重定向，利用请求重定向是两次请求大的特性便可解决这个问题。通过这个实验的独立开发完成，我获益匪浅，对相关内容知识以及各种bug异常的解决都有了更为丰富的经验。