

JAVA web 课程设计报告

设计题目： 在线商城

专 业 计算机科学与技术

设计时间 2020/12/25

2020 年 12 月 25 日

目录

目录

1.概述.....	4
1.1 目的.....	4
1.2 课程设计的组成部分.....	4
1.2.1 需求分析.....	4
1.2.2 简介.....	5
2.在线商城设计内容.....	6
2.1 数据库设计.....	6
(1) 用户数据表.....	6
(2) 商品数据表.....	6
(3) 订单数据存储表.....	9
2.2 前端页面设计+后端 Servlet 实现	11
(1)主页面(index.jsp+ WareSetServlet.Java +WareDetailsServlet.java)	11
(2) 用户登录页面 (login.jsp+ LoginServlet.java)	15
(3) 用户注册页面 (regist.jsp+ RegistServlet.java)	18
(4) 测试界面——文件上传 (test.jsp)	21
(5) 结束支付后的感谢页面 (thanks.jsp)	22
(6) 商 品 详 细 信 息 查 看 页 面 (wareDetails.jsp+	

WareDetailsServlet.java)	23
(7) 购物车页面 (cart.jsp+ putInCartServlet.java)	27
(8) 购物车下单页面 (ordering.jsp+ OrderingServlet.java)	31
(9) 后台管理页面	35
(10) 进货入库页面 (jsp+ ockingServlet.java)	37
(11) 后台管理下架页面 (query.jsp+ QueryServlet.java)	50
(12) 客户查询订单情况页面 (name.jsp+ processRequest.java)	58
(13) 按时间查询订单页面	62
(14)其他 JAVA 类文件	67
3.心得体会	85

1.概述

1.1 目的

通过 JSP&JDBC 的综合运用，掌握中小型 JavaWeb 应用系统服务端数据库编程的基本

原理和常用技术，具备进行服务端数据库表设计和 JSP 页面 JDBC 编程的基本动手能力。具体包括：

- 1) 编写 Java 代码，熟悉并掌握 JDBC 的使用，包括 Connection、Statement、PreparedStatement 和 ResultSet 等对象的原理和常用方法；
- 2)编写 JSP 页面，熟悉并掌握 JDBC 在 JSP 页面中的应用语法、控制结构，掌握 JSP 内置对象的使用以及服务端调试技术；
- 3)实现 JSP&JDBC 等服务端编程技术的融汇贯通，掌握对单表和关联关系表进行 CRUD 等常规数据操作，并达到服务端页面的合理代码布局、动态网页呈现和实时性交互。

- 1.掌握 java-web 网站设计的全过程；
- 2.进一步熟练 Servlet 技术、数据库、标签等-般 java-web 应用技术；
- 3.掌握整个 java-web 应用架构、目录结构以及服务器使用。

1.2 课程设计的组成部分

1.2.1 需求分析

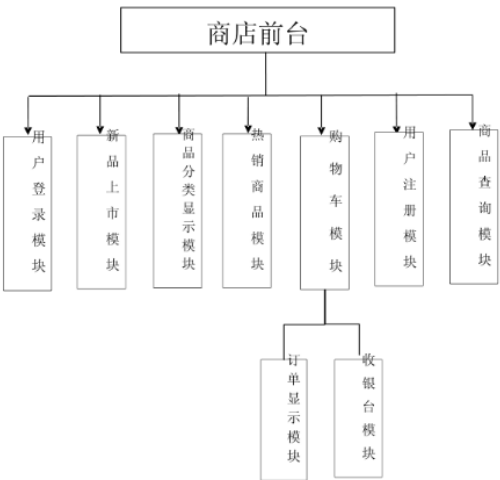
在客户端综合编程实验的基础上，结合 JSP&JDBC 等技术，完成商城后台管理系统的主体模块(登录、产品、产品信息等)，实现相应信息的数据动态显示和后台操作功能，主要包括相关数据库设计以及用户登录功能、产品列表动态数

数据库表内容列表展示功能、购物车添加及与数据库的后台交互功能等，并能够借助开发工具进行服务端代码调试。

- 1.设计一个较完善的网上商店;
- 2.账户模块:提供用户注册、登录，考虑用户数量增减，需要用数据库技术;
- 3.商品显示模块:出于增加销售量考虑，需要有简单广告功能，并提供商品详细参数;推广商品，网站需要对热销商品展示以及商品分类显示、商品查询;
- 4.订单管理模块、订单详细信息模块:显示已保存的订单的详细信息;
- 5.购物车模块:添加/删除商品，结帐，显示订单信息。
- 6.实现管理员登录后台实现一系列操作。

1.2.2 简介

本商城主要利用 **JavaServlet** 技术进行开发，有很强的逻辑性、可扩展性，便于维护。商城界面设计主要实用了 **CSS**，美化了店面。其次本商城设计还使用了 **JSP** 技术、**JDBC** 技术、**JavaBean** 技术。



系统整体功能图

2.在线商城设计内容

2.1 数据库设计

(1) 用户数据表

```
CREATE TABLE [dbo].[clients](          --客户表
    [code] [varchar](18) PRIMARY KEY,    --客户号 (email)
    [title] [varchar](18) NULL,          --客户名称
    [phone] [varchar](15) NULL,          --联系电话
    [wechat] [varchar](18) null,         --微信号
    [address] [varchar](36) NULL,        --住址
    [password] [varchar](15) NULL,       --登录密码
    [manager] [bit] default(0)           --是管理员吗 (默认为否)
) ON [PRIMARY]
GO

insert clients values('zhsan@mail.com','张三','13812345678','13812345678','昆明市青年路 100 号','abc123',0);

insert clients values('lisi@mail.com','李思','13912345678','13912345678','昆明市青年路 101 号','abc321',0);

insert clients values('admin@mail.com','管理员','13712345678','13712345678','昆明市青年路 102 号','abc111',1);

go
```

(2) 商品数据表

```
CREATE TABLE [dbo].[types](          --商品类别表
    [code] [int] IDENTITY(1,1) PRIMARY KEY, --类别号
    [title] [varchar](20) NULL          --类别名称
) ON [PRIMARY]
GO
```

```

CREATE TABLE [dbo].[wares](                                --商品表
    [code] [varchar](10) PRIMARY KEY,                        --商品号
    [title] [varchar](20) NULL,                               --品名
    [model] [varchar](20) NULL,                               --型号
    [depict] [varchar](50) NULL,                             --描述
    [photo] [varchar](20) NULL,                              --图片文件
    [price] [smallmoney] NULL,                               --单价
    [amount] int default 0,                                  --数量
    [type] [int] references [types](code)                   --所属类别
) ON [PRIMARY]
GO

```

```

INSERT [dbo].[types] VALUES (N'照相机')

```

```

INSERT [dbo].[types] VALUES (N'摄像机')

```

```

INSERT [dbo].[types] VALUES (N'手机')

```

```

INSERT [dbo].[types] VALUES (N'笔记本电脑')

```

```

GO

```

```

INSERT [dbo].[wares] VALUES (N'bjb151824', N'联想笔记本电脑', N'联想', N'4 核/2G/200G', N'bjb151824', 6800.0000, 20, 4)

```

```

INSERT [dbo].[wares] VALUES (N'bjb152019', N'索尼笔记本电脑', N'SONY', N'6 核/3G/300G', N'bjb152019', 10000.0000, 20, 4)

```

```

INSERT [dbo].[wares] VALUES (N'bjb152137', N'华硕笔记本电脑', N'ASUS', N'4 核/3G/300G', N'bjb152137', 8000.0000, 20, 4)

```

```

INSERT [dbo].[wares] VALUES (N'bjb152242', N'华硕笔记本电脑',

```

N'ASUS ', N'4 核/3G/400G', N'bjb152242', 8500.0000, 20, 4)

INSERT [dbo].[wares] VALUES (N'bjb152419', N'索尼笔记本电脑 ',
N'SONY ', N'6 核/3G/350G', N'bjb152419', 10500.0000, 20, 4)

INSERT [dbo].[wares] VALUES (N'bjb152610', N'联想笔记本电脑 ', N'联想
, N'5 核/3G/300G', N'bjb152610', 7500.0000, 20, 4)

INSERT [dbo].[wares] VALUES (N'bjb152717', N'华硕笔记本电脑 ',
N'ASUS ', N'4 核/2.5G/500G', N'bjb152717', 9000.0000, 20, 4)

INSERT [dbo].[wares] VALUES (N'sj123136', N'七喜手机 ', N'HEDY
, N'3 核/3G/500M', N'sj123136', 1800.0000, 20, 3)

INSERT [dbo].[wares] VALUES (N'sj123817', N'MOTOROLA 手机 ',
N'MOTOROLA ', N'3 核/3G/600M', N'sj123817', 2000.0000, 20, 3)

INSERT [dbo].[wares] VALUES (N'sj123929', N'三星手机 ',
N'SAMSUNG ', N'3 核/3G/650M', N'sj123929', 2500.0000, 20, 3)

INSERT [dbo].[wares] VALUES (N'sxj122152', N'SONY 摄像机 ', N'SONY
, N'3.5"/2000M/48 倍', N'sxj122152', 12500.0000, 20, 2)

INSERT [dbo].[wares] VALUES (N'sxj122618', N'Canon 摄像机 ', N'CANON
, N'4"/3000M/48 倍', N'sxj122618', 15000.0000, 20, 2)

INSERT [dbo].[wares] VALUES (N'xj120912', N'Premier 相机 ', N'PREMIER
, N'7.0DM-7365/3X Super Zoom', N'xj120912', 1200.0000, 20, 1)

INSERT [dbo].[wares] VALUES (N'xj121208', N'CASIO 相机 ', N'CASIO
, N'600 万像素 A700', N'xj121208', 1500.0000, 20, 1)

INSERT [dbo].[wares] VALUES (N'xj121349', N'SAMSUNG 相机 ',
N'SAMSUNG ', N'三星照相机的若干参数', N'xj121349', 1000.0000, 20, 1)

INSERT [dbo].[wares] VALUES (N'xj121618', N'SONY 相机 ', N'SONY

, N'这是索尼相机的有关参数', N'xj121618', 2000.0000, 20, 1)

INSERT [dbo].[wares] VALUES (N'xj121809', N'FUJIFILM 相机', N'FUJIFILM
, N'这是 FUJIFILM 相机的若干参数', N'xj121809', 1600.0000, 20, 1)

INSERT [dbo].[wares] VALUES (N'xj122028', N'OLYMPUS 相机',
N'OLYMPUS', N'有关参数: u760 All-Weather', N'xj122028', 1800.0000, 20, 1)

go

	code	title	model	depict	photo	price	amount	type
1	bjb151345	华为电脑	华为	8核/5G/500G	bjb151345	3000.00	2	4
2	bjb151824	联想笔记本电脑	联想	4核/2G/200G	bjb151824	6800.00	20	4
3	bjb152019	索尼笔记本电脑	SONY	6核/3G/300G	bjb152019	10000.00	20	4
4	bjb152137	华硕笔记本电脑	ASUS	4核/3G/300G	bjb152137	8000.00	20	4
5	bjb152242	华硕笔记本电脑	ASUS	4核/3G/400G	bjb152242	8500.00	20	4
6	bjb152419	索尼笔记本电脑	SONY	6核/3G/350G	bjb152419	10500.00	20	4
7	bjb152610	联想笔记本电脑	联想	5核/3G/300G	bjb152610	7500.00	20	4
8	bjb152717	华硕笔记本电脑	ASUS	4核/2.5G/500G	bjb152717	9000.00	20	4
9	sj123136	七喜手机	HEDY	3核/3G/500M	sj123136	1800.00	20	3
10	sj123817	MOTOROLA手机	MOTOROLA	3核/3G/600M	sj123817	2000.00	20	3
11	sj123929	三星手机	SAMSUNG	3核/3G/650M	sj123929	2500.00	20	3
12	sxj122152	SONY摄像机	SONY	3.5"/2000M/48倍	sxj122152	12500.00	20	2
13	sxj122618	Canon摄像机	CANON	4"/3000M/48倍	sxj122618	15000.00	20	2
14	xj120912	Premier相机	PREMIER	7.0DM-7365/3X Super Zoom	xj120912	1200.00	20	1
15	xj121208	CASIO相机	CASIO	600万像素A700	xj121208	1500.00	20	1
16	xj121349	SAMSUNG相机	SAMSUNG	三星照相机的若干参数	xj121349	1000.00	20	1
17	xj121618	SONY相机	SONY	这是索尼相机的有关参数	xj121618	2000.00	20	1
18	xj121809	FUJIFILM相机	FUJIFILM	这是FUJIFILM相机的若干...	xj121809	1600.00	20	1
19	xj122028	OLYMPUS相机	OLYMPUS	有关参数: u760 All-Weat...	xj122028	1800.00	20	1

查询已成功执行。

(3) 订单数据仓储表

```
CREATE TABLE [dbo].[orders](
    [code] [bigint] IDENTITY(1,1) PRIMARY KEY,
    [owner] [varchar](18) NULL,
    [date] [date] default getDate(),
```

--订单表
--订单号
--所属客户
--下单日期

```

        [total] [money] NULL,                --总金额

        [consignee] [varchar](18),           --收货人

        [address] [varchar](36),             --收货地址

        [phone] [varchar](15)                --联系电话

    ) ON [PRIMARY]

GO

```

```

CREATE TABLE [dbo].[details](              --订单明细表

    [order] [bigint],                        --所属订单

    [ware] [varchar](10) references wares(code), --所购商品

    [amount] [int] NULL,                    --数量

    [price] [smallmoney] NULL,              --单价

    PRIMARY KEY([order] ASC,[ware] ASC)

) ON [PRIMARY]

GO

```

2.2 前端页面设计+后端 Servlet 实现

(1) 主 页 面 (index.jsp+ WareSetServlet.Java
+WareDetailsServlet.java)

```
<%@page pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>电子商务网站开发实践</title>
        <link rel="stylesheet" href="css/index.css">
    </head>
    <body>
        <section>
            <header>
                <c:choose>
                    <c:when test="\${empty sessionScope.client}">
                        <a href="regist.jsp">注册</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
                        <a href="login.jsp">登录</a>
                    </c:when>
                    <c:otherwise>
                        欢迎\${sessionScope.client.title}归来!
                    </c:otherwise>
                </c:choose>
                <a href="cart.jsp"></a>
            </header>
```

```
<nav>

    <h3>商品类别</h3>

    <c:forEach items="${applicationScope.types}" var="type">

        <a
href='wareSet?code=${type.code}&title=${type.title}'>${type.title}</a>

    </c:forEach>

</nav>

<main>

    <address>${sessionScope.address}</address>

    <c:forEach items="${sessionScope.wares}" var="entry">

        <div>

            <a href="wareDetails?wareId=${entry.key}"></a>

            <b>商品名称: </b>${entry.value.title}

            <b>品牌: </b>${entry.value.model}

            <b>性能参数: </b>${entry.value.depict}

            <b>    单    价    :    </b><fmt:formatNumber
value="${entry.value.price}" type="currency"/>

        </div>

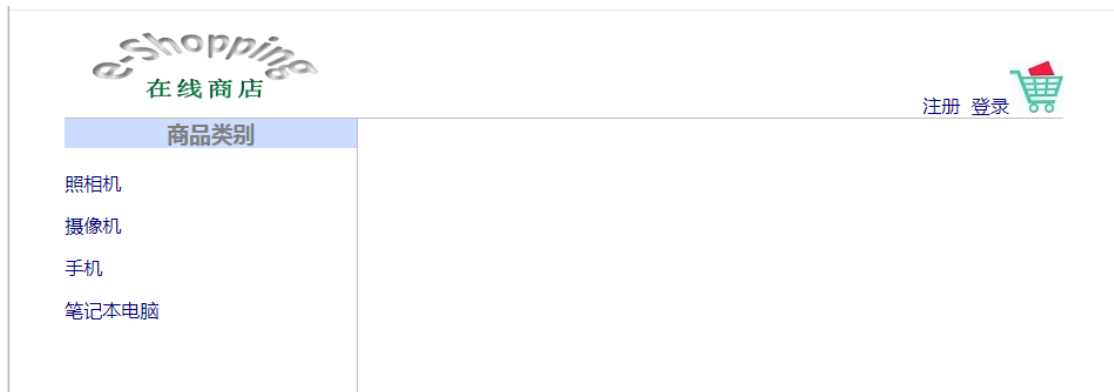
    </c:forEach>

</main>

</section>

</body>
```

</html>



登录数据库，将相应的数据取出来显示到 jsp 页面上

1) WareSetServlet.Java

```
@WebServlet(name = "WareSetServlet", urlPatterns = {"/wareSet"})
```

```
public class WareSetServlet extends HttpServlet {
```

```
    protected void processRequest(HttpServletRequest request,
    HttpServletResponse response)
```

```
    throws ServletException, IOException {
```

```
        request.setCharacterEncoding("UTF-8");
```

```
        int code= Integer.parseInt(request.getParameter("code"));
```

```
        String title= request.getParameter("title");
```

```
        HttpSession session= request.getSession(true);
```

```
        session.setAttribute("address", title);
```

```
        session.setAttribute("wares", DBUtils.queryWaresByType(code));
```

```
        response.sendRedirect("index.jsp");
```

```
}
```

2) WareDetailsServlet.java

```
@WebServlet(name = "WareDetailsServlet", urlPatterns = {"/wareDetails"})

public class WareDetailsServlet extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest request, HttpServletResponse
response)

        throws ServletException, IOException {

        request.setCharacterEncoding("utf-8");

        String wareId= request.getParameter("wareId");

        HttpSession session= request.getSession();

        HashMap<String,Ware> wares=
(HashMap<String,Ware>)session.getAttribute("wares");

        Ware ware= wares.get(wareId);

        session.setAttribute("ware", ware);

        request.getRequestDispatcher("wareDetails.jsp").forward(request,
response);

    }

    @Override

    public String getServletInfo() {
```

```

        return "Short description";

    } // </editor-fold>

}

```



(2) 用户登录页面 (login.jsp+ LoginServlet.java)

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        <style>
            div {
                background-color: skyblue;
                text-align: center;
            }

```

```

        margin-bottom: 20px;
    }
    form {
        display: grid;
        grid: auto-flow/auto auto;
        place-content: center;
        gap: 5px;
    }
    span {
        grid-area: auto/2;
        place-self: start end;
        font-size: 9pt;
        margin-bottom: 8px;
    }
    [type=submit] {
        grid-area: auto/span 2;
        margin-top: 10px;
    }
</style>
</head>
<body>
    <div>客户登陆</div>
    <form action="login" method="post">
        账号 <input name="acc">
        <span>没有账号? 请<a href="regist.jsp">注册</a></span>
        密码 <input type="password" name="pw">
        <span style="color:red">${requestScope.err}<input
type="submit"></span>
    </form>

```



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

登录数据库，将数据库中的客户信息取出来存在 session 里面

1)LoginServlet.java

```
@WebServlet(name = "LoginServlet", urlPatterns = {"/login"})

public class LoginServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        try(    Connection cn=DBUtils.connect();

            PreparedStatement ps= cn.prepareStatement("select * from
clients where code=? and password=?")) {

            ps.setString(1, request.getParameter("acc"));

            ps.setString(2, request.getParameter("pw"));

            try(ResultSet rs= ps.executeQuery()) {
```

```

        if(rs.next()) {

            Client                c=                new
Client(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.g
etBoolean(7));

            request.getSession().setAttribute("client", c);

            String                uri=
(String)request.getSession().getAttribute("aim");

            response.sendRedirect(null==uri?"index.jsp":uri);

        }else{

            request.setAttribute("err", "登录失败！请核对信息。");

request.getRequestDispatcher("login.jsp").forward(request, response);

        }

    }

} catch(SQLException se){}

}

```

(3) 用户注册页面 (regist.jsp+ **RegistServlet.java**)

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>

```

```
<style>
    form {
        display: grid;
        grid: auto-flow/auto auto;
        place-content: center;
        gap: 5px;
    }
    span {
        grid-area: auto/span 2;
        place-self: end;
        margin-top: 1em;
    }
    div {
        background-color: skyblue;
        text-align: center;
        margin-bottom: 1em;
    }
</style>
</head>
<body>
    <div>客户注册</div>
    <form action="regist" method="post">
        账号 <input name="code">
        客户名称 <input name="title">
        密码 <input type="password" name="pw1">
        确认密码 <input type="password" name="pw2">
        联系电话 <input name="phone">
        微信号 <input name="wechat">
        住址 <input name="address">
```

```

        <span>
            <input type="submit">
            <input type="reset">
        </span>
    </form>
</body>
</html>

```

客户注册	
账号	<input type="text"/>
客户名称	<input type="text"/>
密码	<input type="text"/>
确认密码	<input type="text"/>
联系电话	<input type="text"/>
微信号	<input type="text"/>
住址	<input type="text"/>
<input type="button" value="提交"/> <input type="button" value="重置"/>	

使用 `servlet` 将新客户所注册的账号信息存入到数据库中，方便以后购物时使用。

1)RegistServlet.java

```

public class RegistServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)

        throws ServletException, IOException {

        request.setCharacterEncoding("utf-8");

        response.setContentType("text/html;charset=UTF-8");

        try( Connection cn= DBUtils.connect());

```

```

        PreparedStatement ps= cn.prepareStatement("insert clients
values(?,?,?,?,?,0)") {

        ps.setString(1, request.getParameter("code"));

        ps.setString(2, request.getParameter("title"));

        ps.setString(3, request.getParameter("phone"));

        ps.setString(4, request.getParameter("wechat"));

        ps.setString(5, request.getParameter("address"));

        ps.setString(6, request.getParameter("pw1"));

        ps.executeUpdate();

        response.sendRedirect("login.jsp");

    } catch(SQLException se) {

    }
}

```

(4) 测试界面——文件上传 (test.jsp)

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>

```

```
<h1>Hello World!</h1>

<form action="test" method="post" enctype="multipart/form-data">

    <input type="file" name="upload">

    <input type="submit" >

</form>

</body>

</html>
```

Hello World!

未选择任何文件

（5）结束支付后的感谢页面（thanks.jsp）

```
<%@page pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>交易结束</title>
    </head>
    <body>
        <h1>Thank you very much!</h1>
    </body>
```

Thank you very much!

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        <style>
            #tip {
                background: #eee;
                text-align: center;
            }
            main {
                display: grid;
                grid: 2fr repeat(5,auto) 3fr/500px 86px auto;
                place-content: center;
                gap: 10px;
            }
        
```

```

img {
    grid-area: span 7/auto;
}

b {
    text-align: right;
}

b:first-of-type, a:first-of-type {
    align-self: end;
}

a[href] {
    grid-area: auto/3;
    place-self: start;
    margin: 32px 0 0 10px;
}

input[type=button] {
    width: 100%;
}
</style>
</head>
<body>
    <div id="tip">商品详细信息</div>
    <c:set var="w" value="\${sessionScope.ware}" />
    <main>
        
        <b>商品编号: </b><a>\${w.code}</a>
        <b>商品名称: </b>\${w.title}
        <b>品牌: </b>\${w.model}
        <b>性能参数: </b>\${w.depict}
        <b> 单 价 : </b><fmt:formatNumber value="\${w.price}"

```



```

type="currency"/>
        <b>库存量: </b>${w.amount}
        <a href="putInCart"><input type="button" value="放进购物车
"></a>

    </main>
</body>
</html>

```

商品详细信息	
商品编号 :	
商品名称 :	品牌 :
性能参数 :	单价 :
库存量 :	
<input type="button" value="放进购物车"/>	

客户单击某一个商品的图片利用 **Servlet** 就会将对应的商品信息从数据库中获取出来，并将其详细信息显示在商品详细页面上，其中运用了 **HashMap** 来快速读取。

1) WareDetailsServlet.java

```

@WebServlet(name = "WareDetailsServlet", urlPatterns = {"/wareDetails"})

public class WareDetailsServlet extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest request, HttpServletResponse
response)

        throws ServletException, IOException {

```

```

        request.setCharacterEncoding("utf-8");

        String wareId= request.getParameter("wareId");

        HttpSession session= request.getSession();

        HashMap<String,Ware> wares=
(HashMap<String,Ware>)session.getAttribute("wares");

        Ware ware= wares.get(wareId);

        session.setAttribute("ware", ware);

        request.getRequestDispatcher("wareDetails.jsp").forward(request,
response);
    }

    @Override

    public String getServletInfo() {

        return "Short description";

    }// </editor-fold>

}

```

比如我点击了笔记本电脑，它就会从数据库读取相关详细的信息出来并显示在页面上



商品编号：bjb152137
商品名称：华硕笔记本电脑
品牌：ASUS
性能参数：4核/3G/300G
单价：¥ 8,000.00
库存量：20

[放进购物车](#)

(7) 购物车页面 (cart.jsp+ **putInCartServlet.java**)

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="f"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>JSP Page</title>

        <link rel="stylesheet" href="css/cart.css">
```

```
<script src="js/cart.js" defer></script>
```

```
</head>
```

```
<body>
```

```
<div>查看和编辑购物车</div>
```

```
<table border="1">
```

```
<caption id="tot">总 金 额： <f:formatNumber value="${empty  
sessionScope.cart.total? 0.0 : sessionScope.cart.total}" type="currency" /></caption>
```

```
<caption>
```

```
<a href="index.jsp"><input type="button" value="继续浏览  
></a>
```

```
<a href="adm/ordering.jsp"><input type="button" value="下  
单支付"></a>
```

```
</caption>
```

```
<tr>
```

```
<th>商品名称</th>
```

```
<th>品牌型号</th>
```

```
<th>性能参数</th>
```

```
<th>单价</th>
```

```
<th>数量</th>
```

```
<th>小计</th>
```

```
</tr>
```

```
<c:forEach items="${sessionScope.cart.wares}" var="w">
```

```

<tr>

    <td data-code="{w.code}">{w.title}</td>

    <td>{w.model}</td>

    <td>{w.depict}</td>

    <td><f:formatNumber
                                value="{w.price}"
type="currency"/></td>

    <td><input type="number" value="{w.amount}"></td>

    <td><f:formatNumber
                                type="currency"
value="{w.money}"></td>

</tr>

</c:forEach>

</table>

</body>

</html>

```

查看和编辑购物车					
商品名称	品牌型号	性能参数	单价	数量	小计
					总金额: ¥ 0.00
		<input type="button" value="继续浏览"/>	<input type="button" value="下单支付"/>		

在商品的详细页面中，当客户单击“加入购物车”时，利用 servlet 处理相应的操作，将相关的商品信息放到购物车中。

1) putInCartServlet.java

```
@WebServlet(name = "putIntoCartServlet", urlPatterns = {"/putInCart"})
```

```
public class putInCartServlet extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest request, HttpServletResponse
response)

        throws ServletException, IOException {

        HttpSession session= request.getSession(true);

        Ware w= (Ware)session.getAttribute("ware");

        if(null==w) return;

        Ware ware=w.cloneWare();

        Cart cart=(Cart)session.getAttribute("cart");

        cart.putWare(ware);

        response.sendRedirect("cart.jsp");

    }
```

查看和编辑购物车

商品名称	品牌型号	性能参数	单价	数量	小计
OLYMPUS相机	OLYMPUS	有关参数：u760 All-Weather	¥ 1,800.00	1	¥ 1,800.00

总金额：¥ 1,800.00

继续浏览

下单支付

(8) 购物车下单页面 (ordering.jsp+ OrderingServlet.java)

```
<body>

    <div>支付与下单</div>

    <h2>总金额: <f:formatNumber value="${sessionScope.cart.total}"
type="currency"/></h2>

    <hr>

    <form action="../ordering" method="post">

        收      货      人      <input      name="getter"
value="${sessionScope.client.title}">

        收      货      地      址      <textarea
name="address">${sessionScope.client.address}</textarea>

        联      系      电      话      <input      name="phone"
value="${sessionScope.client.phone}">

        <input type="submit">

    </form>

</body>
```

利用 Servlet 将客户所购买的物品信息存入到数据库中

1) OrderingServlet.java

```
@WebServlet(name = "OrderingServlet", urlPatterns = {"/ordering"})
```

```

public class OrderingServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        request.setCharacterEncoding("utf-8");

        HttpSession session= request.getSession();

        String sql="insert orders(owner,total,consignee,address,phone)
values(?,?,?,?,?)";

        Client usr=(Client)session.getAttribute("client");

        Cart cart=(Cart)session.getAttribute("cart");

        long oid;

        try (Connection cn= DBUtils.connect();) {

            try(PreparedStatement ps= cn.prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS)) {

                ps.setString(1, usr.getCode());

                ps.setDouble(2, cart.getTotal());

                ps.setString(3, request.getParameter("getter"));

                ps.setString(4, request.getParameter("address"));

                ps.setString(5, request.getParameter("phone"));

                ps.executeUpdate();

```



```

        try(ResultSet rs=ps.getGeneratedKeys()) {

            rs.next();

            oid= rs.getLong(1);

        }

    }

    try(CallableStatement cs= cn.prepareCall("{call
addDetails(?,?,?,?)}") {

        for(Ware w: cart.getWares()) {

            cs.setLong(1, oid);

            cs.setString(2, w.getCode());

            cs.setInt(3, w.getAmount());

            cs.setDouble(4, w.getPrice());

            cs.addBatch();

        }

        cs.executeBatch();

    }

    request.getSession().invalidate();

    request.getSession(true);

    response.sendRedirect("thanks.jsp");

} catch(SQLException se) {}

}

```

```
// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on  
the + sign on the left to edit the code.">
```

```
@Override
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse  
response)
```

```
throws ServletException, IOException {
```

```
    processRequest(request, response);
```

```
}
```

```
@Override
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse  
response)
```

```
throws ServletException, IOException {
```

```
    processRequest(request, response);
```

```
}
```

```
@Override
```

```
public String getServletInfo() {
```

```
    return "Short description";
```

```
    } // </editor-fold>

}
```

支付与下单	
总金额: ¥ 8,000.00	
收货人	<input type="text" value="李思"/>
收货地址	<input type="text" value="昆明市青年路101号"/>
联系电话	<input type="text" value="13912345678"/>
<input type="button" value="提交"/>	

(9) 后台管理页面

```
<%@page pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <meta charset="UTF-8">

        <title>JSP Page</title>

        <link rel="stylesheet" href="../css/adm.css">

        <link rel="stylesheet" href="../css/mng.css">

    </head>

    <body>

        <div>
```

<header></header>

<nav>

<h3>后台管理</h3>

用户管理

客户

管理员

商品管理

进货入库

查询下架

订单管理


```

        <li>

            <a href="queryOrder.jsp">查询删除</a>

        </li>

    </ul>

</li>

</ul>

</nav>

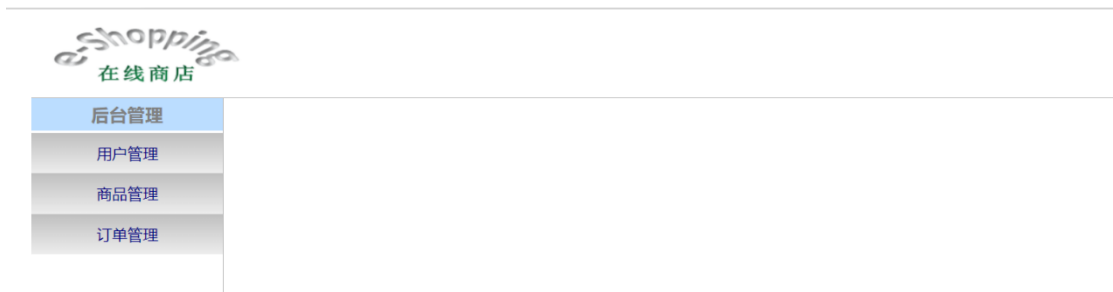
<main></main>

</div>

</body>

</html>

```



(10) 进货入库页面 (jsp+ **ockingServlet.java**)

当管理员单击“进货入库”时，就会进入到输入详细入库信息页面，该页面的编写过程中使用到了 ajax 技术。

```
<%@page pageEncoding="UTF-8"%>
```

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <meta charset="UTF-8">
```

```
        <title>JSP Page</title>
```

```
        <link rel="stylesheet" href="../css/adm.css">
```

```
        <link rel="stylesheet" href="../css/mng.css">
```

```
        <style>
```

```
            #fm {
```

```
                display: grid;
```

```
                grid: auto-flow/auto 200px auto;
```

```
                place-content: center;
```

```
                gap: 5px;
```

```
            }
```

```
            hr {
```

```
                width: 100%;
```

```
                border: 1px solid #eee;
```

```
                grid-area: auto/span 3;
```

```
            }
```

```
            #p1 {
```

```
margin-top: 0;

grid-area: auto/span 2;

text-align: center;

background: skyblue;
}

[value='保存'] {

    grid-area: auto/span 3;

    place-self: end center;

    margin-top: 16px;

    width: 85px;
}

[type=file] {

    display: none;
}

img {

    grid-area: span 6/auto;

    box-sizing: border-box;

    height: 100%;

    width: 160px;

    border: 1px solid silver;
}
```

</style>

</head>

<body>

<div>

<header></header>

<nav>

<h3>后台管理</h3>

用户管理

客户

管理员

商品管理

进货入库


```

        <li><a href="queryWares.jsp">查询下架</a></li>

    </ul>

</li>

<li>

    <a href="#">订单管理</a>

    <ul>

        <li>

            <a href="queryOrder.jsp">查询删除</a>

        </li>

    </ul>

</li>

</ul>

</nav>

<main>

    <p id="p1">进货入库</p>

    <form id="fm">

        商品类 <select id="type">

            <c:forEach    items="${applicationScope.types}"

var="type">

                <option

value="${type.code}">${type.title}</option>

            </c:forEach>

```

```
</select>
```

```
<hr>
```

```
商品号 <input id="code"><img alt="单击，选择图像"
"><input type="file" id="photo">
```

```
数量 <input type="number" id="amount">
```

```
单价 <input id="price">
```

```
品名 <input id="title">
```

```
型号 <input id="model">
```

```
描述 <textarea id="depict"></textarea>
```

```
<input type="button" value="保存">
```

```
</form>
```

```
</main>
```

```
</div>
```

```
<script>
```

```
document.querySelector("#code").onchange=function() {
```

```
var ajax=new XMLHttpRequest(),tp=document.querySelector("#type").value;
```

```
ajax.open("get",
"../existedWare?tp="+tp+"&cd="+this.value);
```

```
ajax.onload=function() {
```

```
var ware=eval(this.responseText),ele;
```

```
if(Object.keys(ware).length>0) {
```

```
document.querySelector("#title").value=ware.title;
```

```
document.querySelector("#amount").value=1;
```

```
document.querySelector("#price").value=ware.price.toFixed(2);
```

```
document.querySelector("#model").value=ware.model;
```

```
document.querySelector("#depict").value=ware.depict;
```

```
setDisable(true);
```

```
} else {
```

```
document.querySelector("#title").value="";
```

```
document.querySelector("#amount").value=0;
```

```
document.querySelector("#price").value="0.00";
```

```
document.querySelector("#model").value="";
```

```
document.querySelector("#depict").value="";
```

```
setDisable(false);
```

```
}
```

```
};
```

```
ajax.send();
```

```
};
```

```
function setDisable(dis) {
```

```
document.querySelector("#title").disabled=dis;
```

```

        document.querySelector("#model").disabled=dis;

        document.querySelector("#depict").disabled=dis;

        document.querySelector("#photo").disabled=dis;
    }

    document.querySelector("img").onclick= function() {

        this.nextSibling.click();

    };

    document.querySelector("[type=file]").onchange=function() {

        var rd=new FileReader();

        rd.onload=function() {

            document.querySelector("img").src=rd.result;

        };

        rd.readAsDataURL(this.files[0]);

    };

    document.querySelector("[value=保存]").onclick= function() {

        var ajax=new XMLHttpRequest();

        ajax.open("post", "../stocking");

        ajax.onload=function() {

            if(Number(ajax.responseText)!=1) return;

            document.querySelector("#code").value="";

            document.querySelector("#title").value="";

```

```
        document.querySelector("#amount").value=0;

        document.querySelector("#price").value="0.00";

        document.querySelector("#model").value="";

        document.querySelector("#depict").value="";

        document.querySelector("#photo").value="";

        document.querySelector("img").src=null;

        setDisable(false);

    };

    var fd=new FormData();

    fd.append("code", document.querySelector("#code").value);

    fd.append("price", document.querySelector("#price").value);

    fd.append("amount",
document.querySelector("#amount").value);

    if(document.querySelector("#title").disabled===false) {

        fd.append("title",
document.querySelector("#title").value);

        fd.append("model",
document.querySelector("#model").value);

        fd.append("depict",
document.querySelector("#depict").value);

        fd.append("photo",
document.querySelector("#photo").files[0]);
```

```

        fd.append("type",
document.querySelector("#type").value);

    }

    ajax.send(fd);

};

</script>

</body>

</html>

```

当管理员单击提交订单时，利用 Servlet 对其操作多出反映，并将所提交的购物数据存入到数据库中。

1) ockingServlet.java

```
@MultipartConfig
```

```
@WebServlet(name = "StockingServlet", urlPatterns = {"/stocking"})
```

```
public class StockingServlet extends HttpServlet {
```

@Override

```
protected void doPost(HttpServletRequest request, HttpServletResponse  
response)
```

```
throws ServletException, IOException {
```

```
response.setContentType("text/plain; charset=utf-8");
```

```
request.setCharacterEncoding("utf-8");
```

```
String title=request.getParameter("title"), sql,  
code=request.getParameter("code");
```

```
int k = -1;
```

```
if(null==title)
```

```
sql="update wares set price=?, amount=amount+? where code=?";
```

```
else
```

```
sql="insert wares values(?,?,?,?,?,?,?)";
```

```
try ( Connection cn= DBUtils.connect();
```

```
PreparedStatement ps=cn.prepareStatement(sql)) {
```

```
if(null==title) {
```

```
ps.setDouble(1,
```

```
Double.parseDouble(request.getParameter("price")));
```

```
ps.setInt(2,
```

```
Integer.parseInt(request.getParameter("amount")));
```

```
ps.setString(3, request.getParameter("code"));
```

```

    }else{

        ps.setString(1, code);

        ps.setString(2, request.getParameter("title"));

        ps.setString(3, request.getParameter("model"));

        ps.setString(4, request.getParameter("depict"));

        ps.setString(5, code); //用商品号作为图像文件名,确保文
件名不重复

        ps.setDouble(6,
Double.parseDouble(request.getParameter("price")));

        ps.setInt(7,
Integer.parseInt(request.getParameter("amount")));

        ps.setInt(8, Integer.parseInt(request.getParameter("type")));

    }

    ps.executeUpdate();

} catch(SQLException se) {

    se.printStackTrace();

}

Part part=request.getPart("photo");

if(null != part){

    Path
target=Paths.get(this.getServletContext().getRealPath(""),"images",code+".jpg");

    try (    InputStream source=part.getInputStream());

```



```

    ) {

        Files.copy(source,

                                target,

StandardCopyOption.REPLACE_EXISTING);

    }

}

k = 1;

try(PrintWriter out=response.getWriter()){

    out.print(k);

}

}

}

```

对象资源管理器

连接 (local) (SQL Server 10.50.1600 - stu72jq)

数据库

系统数据库

数据库快照

eshop

数据库关系图

表

系统表

dbo.clients

dbo.details

dbo.orders

dbo.types

dbo.wares

视图

同义词

可编程性

Service Broker

存储

安全性

ReportServer

ReportServerTempDB

UFDATA_001_2020

UFDATA_998_2012

UFDATA_999_2011

UFSub

UFSystem

安全性

服务端对象

复制

管理

SQL Server 代理

SQLQuery2.sql - (lo...op (stu72jq (53))*) eshop.sql - (local).eshop (stu72jq (52))

select * from clients

select * from wares

结果 消息

	code	title	model	depict	photo	price	amount	type
5	bpb152419	索尼笔记本电脑	SONY	6核/3G/350G	bpb152419	10500.00	20	4
6	bpb152610	联想笔记本电脑	联想	5核/3G/300G	bpb152610	7500.00	20	4
7	bpb152717	华硕笔记本电脑	ASUS	4核/2.5G/500G	bpb152717	9000.00	20	4
8	sq123136	七喜手机	HEDY	3核/3G/500M	sq123136	1800.00	20	3
9	sq123817	MOTOROLA手机	MOTOROLA	3核/3G/600M	sq123817	2000.00	20	3
10	sq123929	三星手机	SAMSUNG	3核/3G/650M	sq123929	2500.00	20	3
11	sq122152	SONY摄像机	SONY	3.5"/2000M/48倍	sq122152	12500.00	20	2
12	sq122618	Canon摄像机	CANON	4"/3000M/48倍	sq122618	15000.00	20	2
13	xj120912	Premier相机	PREMIER	7.00M-7365/3X Super Zoom	xj120912	1200.00	20	1
14	xj121208	CASIO相机	CASIO	600万像素A700	xj121208	1500.00	20	1
15	xj121349	SAMSUNG相机	SAMSUNG	三星照相机的若干参数	xj121349	1000.00	20	1
16	xj121618	SONY相机	SONY	这是索尼相机的有关参数	xj121618	2000.00	20	1
17	xj121809	FUJIFILM相机	FUJIFILM	这是FUJIFILM相机的若干...	xj121809	1600.00	20	1
18	xj122028	OLYMPUS相机	OLYMPUS	有关参数...	xj122028	1800.00	20	1
19	xj5001	索尼相机	sony1001	好	xj5001	5200.00	2	1



(11) 后台管理下架页面 (query.jsp+ QueryServlet.java)

<body>

<div>

<header></header>

<nav>

<h3>后台管理</h3>

用户管理

客户

管理员

商品管理

进货入库

查询下架

订单管理

账户查询删除

日期查询删除

浏览购物出

</nav>

<main>

<p>查询与下架</p>

<table>

```
<caption>

    商品类 <select name="type">

        <c:forEach
items="${applicationScope.types}" var="tp">

            <option
value="${tp.code}">${tp.title}</option>

        </c:forEach>

    </select>

</caption>

<caption><input type="button" value="下架所选商品"
id="btn"></caption>

<thead>

    <tr>

        <th><input type="checkbox" title="都选或都
不选" id="cb1"></th>

        <th>图像</th>

        <th>商品号</th>

        <th>品名</th>

        <th>型号</th>

        <th>描述</th>

        <th>单价</th>

        <th>数量</th>
```

```

        </tr>

    </thead>

    <tbody>

        </tbody>

    </table>

</main>

</div>

<script>

    document.querySelector("select").onchange=function() {

        var ajax=new XMLHttpRequest();

        ajax.open("get", "QueryServlet?type="+this.value);

        ajax.onload=function() {

document.querySelector("tbody").innerHTML=ajax.responseText;

        };

        ajax.send();

    };

    document.querySelector("select").onchange();    //显示第一个
类别的商品

    document.querySelector("#cb1").onchange=function() {

        var

```

```

cbs=document.querySelector("tbody").querySelectorAll("input");

        for(var cb of cbs)

            cb.checked= this.checked;

    };

    document.querySelector("#btn").onclick=function() {

        var
cbs=document.querySelector("tbody").querySelectorAll("input"),cds="";

        for(var cb of cbs)

            if(cb.checked){

                var rw = cb.parentElement.parentElement;

                cds+=","+rw.cells[2].textContent+"";

                rw.parentElement.removeChild(rw);

            }

        var ajax = new XMLHttpRequest(),fd=new FormData();

        ajax.open("post","QueryServlet");

        ajax.onload=function(){};

        fd.append("cds",cds.slice(1));

        ajax.send(fd);

    };

</script>

```

</body>



利用 Servlet 登录数据库，将数据库中的相应信息取出来，类似于在询与下架页面中。

1) QueryServlet.java

@MultipartConfig

@WebServlet(name = "QueryServlet", urlPatterns = {"/QueryServlet"})

public class QueryServlet extends HttpServlet {

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html; charset=UTF-8");

String fm="<tr><th><input type='checkbox'></th><td><img

```
src='images/%s.jpg'

width='60'></td><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td>%d</td></tr>"
```



```
tb="";

try(    Connection cn=DBUtils.connect();

        PreparedStatement ps=cn.prepareStatement("select * from
wares where type="+request.getParameter("type"));

        ResultSet rs=ps.executeQuery()) {

while(rs.next()) {

    tb+=String.format(fm, rs.getString("photo"),

                        rs.getString("code"),

                        rs.getString("title"),

                        rs.getString("model"),

                        rs.getString("depict"),

                        rs.getDouble("price"),

                        rs.getInt("amount"));

}

}catch(SQLException se) {

se.printStackTrace();

}

try(PrintWriter out=response.getWriter()) {

out.print(tb);
```



```
    }  
}
```

```
@Override
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse  
response)
```

```
throws ServletException, IOException {
```

```
try(    Connection cn=DBUtils.connect();
```

```
        PreparedStatement ps=cn.prepareStatement("delete wares  
where code in (" + request.getParameter("cds") + ")")){
```

```
    ps.executeUpdate();
```

```
    } catch(SQLException se){
```

```
        se.printStackTrace();
```

```
    }
```

```
}
```

```
@Override
```

```
public String getServletInfo() {
```

```
    return "Short description";
```

```
}// </editor-fold>
```

```
}
```

(12) 客户查询订单情况页面 (name.jsp+processRequest.java)

```
<body>

    <div>

        <header></header>

        <nav>

            <h3>后台管理</h3>

            <ul>

                <li>

                    <a href="#">用户管理</a>

                    <ul>

                        <li><a href="#">客户</a></li>

                        <li><a href="#">管理员</a></li>

                    </ul>

                </li>

                <li>

                    <a href="#">商品管理</a>

                    <ul>

                        <li><a href="restock.jsp">进货入库</a></li>

                        <li><a href="#">查询下架</a></li>

                    </ul>

                </li>

            </ul>

        </nav>

    </div>

</body>
```

```
</li>

<li>

    <a href="#">订单管理</a>

    <ul>

        <li><a href="name.jsp">账户查询删除</a>

        <li><a href="date.jsp">日期查询删除</a>

        <li><a href="../index.jsp">浏览购物出</a>

    </ul>

</li>

</ul>

</nav>

<main>

    <p id="p1">查询删除</p>

    <form action="../SelectServlet" method="get" style="text-align: center; margin-top: 100px">

        <p>请输入需要查询的账户号： <input type="text" name="owner" >

            <input type="submit" value="查询">

        </p>

    </form>

</main>

</div>
```

</body>

利用 servlet 登录数据库查询客户订单表，并将其相关的信息显示到这个页面上。

1) processRequest.java

```
@MultipartConfig

@WebServlet(name = "SelectServlet", urlPatterns = {"/SelectServlet"})

public class SelectServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

    }

    @Override

    protected void doGet(HttpServletRequest request, HttpServletResponse
response)

        throws ServletException, IOException {

        request.setCharacterEncoding("utf-8");

        response.setCharacterEncoding("utf-8");

        try (Connection cn=DBUtils.connect();
```

```

        PreparedStatement ps=cn.prepareStatement("select * from orders
where owner=?");{

        ps.setString(1, request.getParameter("owner"));

        try (ResultSet rs=ps.executeQuery()) {

                List<Order> orders=new ArrayList<Order>();

                for(;rs.next();) {

                        Order order = new Order();

                        order.setCode(rs.getLong("code"));

                        order.setOwner(rs.getString("owner"));

                        order.setDate(rs.getString("date"));

                        order.setTotal(rs.getDouble("total"));

                        order.setConsignee(rs.getString("consignee"));

                        order.setAddress(rs.getString("address"));

                        order.setPhone(rs.getString("phone"));

                        orders.add(order);

                }

                request.getSession().setAttribute("orders", orders);

request.getRequestDispatcher("/adm/select.jsp").forward(request, response);

        }

```

```

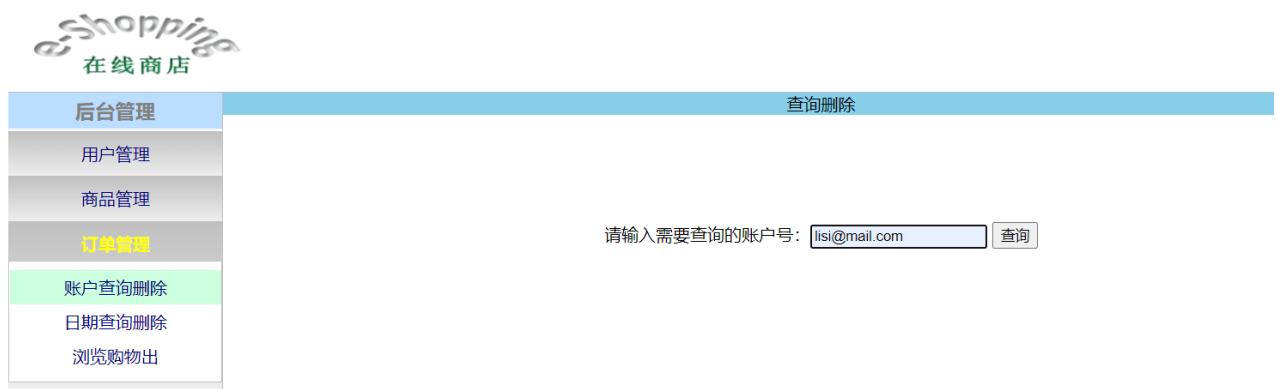
    }catch(SQLException se) {

        se.printStackTrace();

    }

}

```



(13) 按时间查询订单页面

```
<body>
```

```
<div>
```

```
<header></header>
```

<nav>

<h3>后台管理</h3>

用户管理

客户

管理员

商品管理

进货入库

查询下架

订单管理

账户查询删除

日期查询删除

浏览购物出

</nav>

<main>

<p id="p1">查询删除</p>

<form action=" ../TimeServlet" method="get"
style="text-align: center; margin-top: 100px">

<p> 输入需要查询的日期： <input type="text"
name="date" >

<input type="submit" value="查询">

</p>

</form>

</main>

</div>

</body>


在线商店

后台管理		订单查询						
用户管理		日期范围 <input type="text" value="年 / 月 / 日"/> ~ <input type="text" value="年 / 月 / 日"/> <input type="button" value="查询"/>						
商品管理		订单号	所属客户	下单日期	总金额	收货人	目的地址	联系电话
订单管理		<input type="button" value="退订所选"/>						

```
protected void doGet(HttpServletRequest request, HttpServletResponse  
response)
```

```
throws ServletException, IOException {
```

```
request.setCharacterEncoding("utf-8");
```

```
response.setCharacterEncoding("utf-8");
```

```
try (Connection cn=DBUtils.connect();
```

```
PreparedStatement ps=cn.prepareStatement("select * from  
orders where date=?");{
```

```
ps.setString(1, request.getParameter("date"));
```

```
try (ResultSet rs=ps.executeQuery()) {
```

```
List<Order> orders=new ArrayList<Order>();
```

```
for(;rs.next();) {
```

```
Order order = new Order();
```

```
order.setCode(rs.getLong("code"));
```

```
order.setOwner(rs.getString("owner"));
```

```
        order.setDate(rs.getString("date"));

        order.setTotal(rs.getDouble("total"));

        order.setConsignee(rs.getString("consignee"));

        order.setAddress(rs.getString("address"));

        order.setPhone(rs.getString("phone"));

        orders.add(order);

    }

    request.getSession().setAttribute("orders", orders);

request.getRequestDispatcher("/adm/select.jsp").forward(request, response);

    }

    }catch(SQLException se) {

        se.printStackTrace();

    }

}
```



后台管理

用户管理

商品管理

订单管理

查询删除

订单号	所属客户	下单日期	总金额	收货人	收货地址	联系电话	操作
10	zhsan@mail.com	2020-12-25	8000.0	张三	昆明市青年路100号	13812345678	删除订单
11	zhsan@mail.com	2020-12-25	1800.0	张三	昆明市青年路100号	13812345678	删除订单
22	admin@mail.com	2020-12-25	8000.0	管理员	昆明市青年路102号	13712345678	删除订单

(14)其他 JAVA 类文件

1) Cart.java

```
public class Cart {  
  
    private HashMap<String, Ware> wares= new HashMap<>();//对数据进行高速处理
```

```
    public Ware getWare(String code) {  
  
        return wares.get(code);  
  
    }
```

```
    public void putWare(Ware ware) {  
  
        String wid=ware.getCode();//获取状态码  
  
        if(wares.containsKey(wid)) { //判断对应的值存在吗
```

```
        Ware w=wares.get(wid);//  
  
        w.setAmount(w.getAmount()+1);  
  
    }else{  
  
        wares.put(wid, ware);//指定 wid 的值  
  
    }  
  
}
```

```
public void modifyAmount(String ware, int n) {  
  
    Ware w=wares.get(ware);  
  
    if(null==w) return;//为空停止  
  
    w.setAmount(n);//添加 n  
  
    if(n==0) wares.remove(ware);//当数量为 0 删了这个商品  
  
}
```

```
public double getTotal() {  
  
    double t=0;  
  
    for(Map.Entry<String,Ware> e: wares.entrySet()) {  
  
        t+= e.getValue().getMoney();  
  
    }  
  
    return t;  
  
}
```

```
public Collection<Ware> getWares() {  
  
    return wares.values();  
  
}  
  
}
```

2) Client.java

```
public class Client {  
  
    private String code, title, phone, wechat, address;  
  
    private boolean manager;  
  
    public Client() {}  
  
    public Client(String code, String title, String phone, String wechat, String  
address, boolean manager) {  
  
        this.code = code;  
  
        this.title = title;  
  
        this.phone = phone;  
  
        this.wechat = wechat;  
  
        this.address = address;  
  
        this.manager= manager;  
  
    }  
  
}
```

```
public String getCode() {  
  
    return code;  
  
}
```

```
public void setCode(String code) {  
  
    this.code = code;  
  
}
```

```
public String getTitle() {  
  
    return title;  
  
}
```

```
public void setTitle(String title) {  
  
    this.title = title;  
  
}
```

```
public String getPhone() {  
  
    return phone;  
  
}
```

```
public void setPhone(String phone) {  
  
    this.phone = phone;  
  
}
```

```
public String getWechat() {  
  
    return wechat;  
  
}
```

```
public void setWechat(String wechat) {  
  
    this.wechat = wechat;  
  
}
```

```
public String getAddress() {  
  
    return address;  
  
}
```

```
public void setAddress(String address) {  
  
    this.address = address;  
  
}
```

```
public boolean isManager() {
```

```
        return manager;
    }

    public void setManager(boolean manager) {

        this.manager = manager;

    }

}
```

3) DBUtils.java

```
public class DBUtils {

    private static String url;

    private static List<Type> types;


    public static void setUrl(String url) {

        DBUtils.url= url;

    }


    public static Connection connect() throws SQLException {

        return DriverManager.getConnection(DBUtils.url);

    }

}
```



```

public static boolean update(String sql, Object...args) {

    try(Connection cn= connect();//建立链接

        CallableStatement cs= cn.prepareCall(sql)) {//调用数据库存储过
程

        for(int i=0; i<args.length; i++) cs.setObject(i+1, args[i]);

        cs.executeUpdate();//上传 SQL 语句进行查询

        return true;

    }catch(SQLException se) {

        se.printStackTrace();

        return false;

    }

}

```

```

public static List<Type> queryTypes() {

    types= new ArrayList<>();

    try(Connection cn= DBUtils.connect();//建立链接

        PreparedStatement ps= cn.prepareStatement("select * from
types");//预处理 SQL 语句

        ResultSet rs= ps.executeQuery()) {//上传 SQL 语句进行查询

            while(rs.next()) {

                types.add(new Type(rs.getInt("code"),rs.getString("title")));//
在 types 中插入数据

```

```

    }

    } catch(SQLException se) {

        se.printStackTrace();

    }

    return types;

}

```

```

public static HashMap<String,Ware> queryWaresByType(int type) {

    HashMap<String,Ware> map= new HashMap<>();

    try(Connection cn= DBUtils.connect();//建立链接

        PreparedStatement ps= cn.prepareStatement("select * from wares
where type="+type);//预处理 SQL 语句

        ResultSet rs= ps.executeQuery()){//上传 SQL 语句进行查询

            for(String id; rs.next();) {

                Ware ware= new Ware(id=rs.getString("code"),

                    rs.getString("title"),

                    rs.getString("model"),

                    rs.getString("depict"),

                    rs.getString("photo"),

                    rs.getDouble("price"),

                    rs.getInt("amount"),

```

```

        rs.getInt("type"));

        map.put(id, ware);

    }

    }catch(SQLException se) {

        se.printStackTrace();

    }

    return map;

}

}

```

4) Order.java

```

public class Order {

    private long code;

    private String owner, date, consignee, phone, address;

    private double total;

    public Order(){}

    public Order(long code, String owner, String date, String consignee, String
phone, String address, double total){

        this.code = code;

        this.owner = owner;

```

```
        this.address = address;

        this.date = date;

        this.total = total;

        this.phone = phone;

        this.consignee = consignee;
    }

    public long getCode() {

        return code;
    }

    public void setCode(long code) {

        this.code = code;
    }

    public String getOwner() {

        return owner;
    }

    public void setOwner(String owner) {

        this.owner = owner;
```

```
}
```

```
public String getDate() {  
    return date;  
}
```

```
public void setDate(String date) {  
    this.date = date;  
}
```

```
public String getConsignee() {  
    return consignee;  
}
```

```
public void setConsignee(String consignee) {  
    this.consignee = consignee;  
}
```

```
public String getPhone() {  
    return phone;  
}
```

```
public void setPhone(String phone) {  
  
    this.phone = phone;  
  
}
```

```
public String getAddress() {  
  
    return address;  
  
}
```

```
public void setAddress(String address) {  
  
    this.address = address;  
  
}
```

```
public double getTotal() {  
  
    return total;  
  
}
```

```
public void setTotal(double total) {  
  
    this.total = total;  
  
}
```

```
}
```

5) Type.java

```
public class Type {  
  
    private int code;  
  
    private String title;  
  
    public Type() {}  
  
    public Type(int code, String title) {  
  
        this.code = code;  
  
        this.title = title;  
  
    }  
  
    public int getCode() {  
  
        return code;  
  
    }  
  
    public void setCode(int code) {  
  
        this.code = code;  
  
    }  
  
    public String getTitle() {
```

```
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }
}
```

6) Ware.java

```
public class Ware implements Serializable {

    private String code, title, model, depict, photo;

    private double price;

    private int amount, type;

    public Ware() {}

    public Ware(String code, String title, String model, String depict, String
photo, double price, int amount, int type) {

        this.code = code;

        this.title = title;
```



```
        this.model = model;

        this.depict = depict;

        this.photo = photo;

        this.price = price;

        this.amount = amount;

        this.type = type;
    }
}
```

```
public Ware(Ware w) {

    this.code = w.code;

    this.title = w.title;

    this.model = w.model;

    this.depict = w.depict;

    this.photo = w.photo;

    this.price = w.price;

    this.amount = 1;

    this.type = w.type;
}
}
```

```
public Ware cloneWare() {

    return new Ware(this);
}
```

```
}
```

```
public int getType() {  
    return type;  
}
```

```
public void setType(int type) {  
    this.type = type;  
}
```

```
public String getCode() {  
    return code;  
}
```

```
public void setCode(String code) {  
    this.code = code;  
}
```

```
public String getTitle() {  
    return title;  
}
```

```
public void setTitle(String title) {  
  
    this.title = title;  
  
}
```

```
public String getModel() {  
  
    return model;  
  
}
```

```
public void setModel(String model) {  
  
    this.model = model;  
  
}
```

```
public String getDepict() {  
  
    return depict;  
  
}
```

```
public void setDepict(String depict) {  
  
    this.depict = depict;  
  
}
```

```
public String getPhoto() {  
    return photo;  
}
```

```
public void setPhoto(String photo) {  
    this.photo = photo;  
}
```

```
public double getPrice() {  
    return price;  
}
```

```
public void setPrice(double price) {  
    this.price = price;  
}
```

```
public int getAmount() {  
    return amount;  
}
```

```
public void setAmount(int amount) {
```

```
        this.amount = amount;

    }

    public double getMoney() {

        return this.amount * this.price;

    }

}
```

3.心得体会

1.一个 Servlet 往往对应了不同的操作，一个对象的增删改查，都放在了一个 Servlet 里面实现，这样设计可有效减少 Servlet 数量；

2.对于网页页面效果要求较高的时候，多采用 JSP 来编写，对于后台处理求较高时，多采用 Servlet。

3.尽量使用和数据库版本一致的驱动， 否则 JDBC 不正常；.

4.在编写样式表之前一定要注意 CSS 文档的编码类型一定要和 HTML 文档的字符集类型一致。

5.对数据库的理解，也更熟悉 MySQL 的使用。数据库是整个网站的后台支持，没有合理的数据库，整个网站的性能就会降低，甚至影响网站的使用，合理的创建数据表，以及合理的进行划分，是网站设计的第一步。同时也熟悉了对数据库的相关操作。

6 学会使用 NetBeans IDE 和 Eclipse,本课题要求是用 NetBeans IDE 进行开发，但是在同时使用两着进行对比之后，发现 Eclipse 要更具优势，于是最终选择 Eclipse 进行开发。

7 深了 HTML 的熟悉程度,同时学了 Javascript 的相关知识,更深一步的了解 JSP 技术。巩固了 java 知识。.

总之,本次的课程设计,刚开始的时候面临着很多的困难,连开发环境的搭建都是问题,但是慢慢的,通过自己查阅相关资料和向其他人请教的方式,已经基本的完成本次课程设计的内容,从中获益匪浅。