

51 购物网站系统设计

1、系统概述及开发原则

(1)开发意义：实现一个买家与卖家可以与之交互的购物网站。

a.买家可以进行的操作：

注册登录 浏览商品 看商品详情页 加入购物车 生成订单 查看，修改个人信息 修改地址

b.卖家可以进行的操作：

注册登录 个人信息维护 上架商品

(2)开发过程指导思想：

a.开发核心架构：Windows 系统→MySQL 数据库→Tomcat 中间件
→MyEclipse 开发平台

b.指导思想：DAO MVCII

2、系统功能

(1) 用户注册

用户通过输入用户 ID，用户名，用户密码和用户电话的信息完成注册。然后系统将新注册的用户信息存入用户的数据库。

(2) 用户登录

注册成功的用户可以输入相应的用户 ID 和密码进行登录。系统从数据库获取信息，将该登录信息与数据库的信息进行核对，完全相对应的话即登录成功，否则登录失败，该页面登录内容清空需要重新输入正确的 ID 和密码。卖家登录成功，页面跳转到卖家信息界面；买家登录成功，页面跳转到商品详情页面。

(3) 新增商品功能

在个人信息页面右下方可以点击“新增商品”，之后会转跳至新增商品页面，卖家可以进行商品增加功能，根据商品编号、商品名称、商品颜色、商品容量、商品数量等标签，对商品进行增加的操作。

(4) 商品列表

从数据库提取每个商品的图片、名字、价格和销量，分别组成每个商品框显示在商品列表中，点击每个商品框可以跳转到该商品的详情页。

(5) 商品详情页加入购物车

商品详情页中显示该商品详细信息，若用户需要选购该商品，可通过加减按钮修改数量或者直接文本框输入数量，确认数量后点击“开始选购”后可以开始选购商品颜色，内存和尺寸等。选购完可以点击“加入购物车”，加入购物车后跳转。

(6) 购物车列表

购物车中显示着选购的商品内容以及数量，并通过单种商品的价格和数量算出该商品全部价格。在购物车最后会计算出全部购物车中商品的价格，点击“结算”按钮后形成订单。

(7) 生成订单

根据购物车信息以及用户个人地址信息等形成订单，确认个人信息，若需要新增地址，可以点击“使用新地址”后跳转到个人信息页面来新增。订单中显示商品个数，价格等，最终形成总价格。

(8) 用户个人中心，信息及地址维护

用户登录成功后，可以转跳到个人中心，在该页面会显示该用户的用户头像、用户编号、用户昵称、所在城市和详细地址，方便用户进行检查个人信息是否有误，并可以点击页面右下方的“修改信息”，及时从数据库后台更改相关个人信息，确保准确性。

3、技术架构

DAO: 用户注册，新增商品功能，商品列表，用户个人中心，信息及地址维护

哈希表: 加入购物车功能，购物车列表，订单

Servlet: 用户登录

Session: 全部

MVCII: 登录和注册，Servlet+ISP 实现登录功能

JavaScript: 按钮与超链接的应用,密码与用户名输入的限制

4、开发过程

(1) 进行需求分析

a.买家需求分析: 能够浏览商品列表, 浏览商品详情页, 加入购物车, 将多个加入购物车, 生成订单, 登录注册, 修改个人信息 (包括对地址的修改)

b.卖家需求分析: 登录注册, 修改个人信息, 上架商品。

(2) 规划静态内容

共设计 10 个页面。包括 index (首页), login (登录), register (注册), shopList (商品列表), shopInfo (商品详情页), shopCart(购物车), pay (订单界面), 卖家个人中心, 卖家个人中心, 新增商品页面。

(3) 美工设计阶段

风格和主题问题: 在制作“新增商品”和“个人中心”网页的过程中, 我们参考很多的网站风格, 最后根据我们的“51shop”风格敲定了橙色风为商城主题。

(4) 程序开发阶段

a.设计商品和用户数据库

b.包的设计, JavaBean, DAO, Servlet 类

c.jsp 页面设计

eg:哈希表: 将商品详情页的商品信息通过超链接传输参数到 addCart.jsp 后将商品各项信息对应放进 goods 的 JavaBean 中, 之后形成哈希表。将哈希表每项对应显示在购物车页面, 并计算各项价格。订单页面的地址部分通过 session 从该用户的个人地址信息中取得, 商品部分通过购物车的哈希表取得, 并计算该订单总价。

商品详情页的数量加减问题: 使用 JavaScript 编写方法实现数量的加减后再用 form 自身提交来获取文本框中数量值。

(5) 测试

语言乱码问题: getBytes.iso,utf-8

Bug 问题: 询问老师与同学, 上网查找资料

5、界面展示

(1)用户注册

1) 注册：用户通过输入用户 ID，用户名，用户密码和用户电话的信息完成注册。然后系统将新注册的用户信息存入用户的数据库。

A user registration form titled "注册" (Register). It contains four input fields: "请输入ID" (Please enter ID), "请输入昵称" (Please enter nickname), "设置密码" (Set password), and "请输入手机号" (Please enter mobile number) with a red asterisk indicating it is required. Below the fields is a blue "注册" (Register) button. At the bottom, there is a checkbox labeled "点击表示您同意商城《服务协议》" (Click to indicate you agree to the mall's Service Agreement) and a "登录" (Login) link.

用户注册界面图

2) 注册成功：注册完成后显示用户信息添加成功。

laptop-hmth5lj3:8080 显示

用户信息添加成功

确定

用户注册成功界面图

(2)用户登录：登录成功则实现跳转，登录不成功则提示用户登录失败。

登录

卖家登录失败，错误的用户名和密码！

邮箱/手机/用户名

请输入密码

登录

☐记住密码

注册

合作账号

QQ登录

微博登录

微信登录

用户登录界面图

(3)新增商品功能：在新增商品界面，卖家可以进行商品增加或重置功能，根据对应的商品编号、商品名称、商品颜色、商品容量、商品数量等标签，对商品进行增加的操作，如果输入的信息有误也可以点击重置按钮进行清空。

S1购商城

商品编号：

商品名称：

商品价格：

商品库存：

商品颜色：

商品容量：

商品尺寸：

商品图片：

增加

重置

新增商品界面图

(4)商品列表：从数据库读取图片名称，商品名称，商品价格，商品销量，进行列表显示。



商品列表界面图

(5)商品详情页加入购物车：商品列表通过超链接+参数的方式将需要的参数传到商品详情页进行显示。通过 js 可以对数量进行加减。



商品详情页界面图

(6)购物车列表：将商品详情页中用户选择的商品、数量等加入购物车

车。通过把商品详情页中的商品各项信息放入哈希表中，之后在购物车页面显示哈希表中的内容，显示该商品数量，单价以及总价等。



购物车页面图

(7)生成订单

1) 订单-地址：根据 session 提取用户个人信息的地址部分并显示在订单对应的地方，若需要新增地址可以点击“使用新地址”去添加并存到数据库。



订单-地址页面图

2) 订单-信息确认：订单页面中通过哈希表提取到所需商品信息后显示在页面中，并计算出总价。最后用红框标注提醒用户确认价格和地址准确无误后可下单。

确认订单信息

商品信息	单价	数量	金额	配送方式
 魅族 颜色: 黑色 包装: 裸装	2000.0	1	2000.0	包邮

买家留言:

优惠券

红包

合计 (含运费) ¥ 2000.0

实付款: **¥2000.0**

寄送至: 上海市南京东路33号

收货人: 李兴华 1976435890

提交订单

订单-信息确认页面图

(8)用户个人中心，信息及地址维护

1) 买家：以橙色为主题的买家个人中心界面，在此页面上会转跳出买家先前在注册登录界面时输入的数据库信息，如果信息有误还可以进行修改个人信息。

买家个人中心

买家头像



买家编号

606060

买家昵称

李兴华

所在城市

上海市

详细地址

上海市南京东路33号

电话号码

1976435890

修改

买家个人中心页面图

在卖家用户信息修改界面，买家可以根据界面上的类目进行对自己编号、昵称、所在城市、详细地址和电话号码的修改，确保自己的信息最终在个人中心展示时是正确的。

买家 李兴华 的信息修改

买家编号: 606060

买家昵称: 李兴华

所在城市: 上海市

详细地址: 上海市南京东路33号

电话号码: 1976435890

修改

买家用户信息修改界面图

2) 卖家：以橙色为主题的卖家个人中心界面，在此页面上会转跳出卖家先前在注册登录界面时输入的数据库信息，并可以进行添加商品，如果信息有误还可以进行修改个人信息。



卖家个人中心

卖家头像

卖家编号

6060606060

卖家昵称

卖家1

所在城市

上海市

详细地址

上海市静安区豪华别墅

电话号码

6844684458

添加商品

修改小主信息

卖家个人中心界面

在卖家用户信息修改界面，卖家可以根据界面上的类目进行对自己编号、昵称、所在城市和详细地址的修改，确保自己的信息最终在个人中心展示时是正确的。

卖家 卖家1 的信息修改

卖家编号:	<input type="text" value="6060606060"/>
卖家昵称:	<input type="text" value="卖家1"/>
所在城市:	<input type="text" value="上海市"/>
详细地址:	<input type="text" value="上海市静安区豪华别墅"/>
<input type="button" value="修改"/>	

卖家用户信息修改界面

6、系统特色及总结

登录界面：不仅可以用用户 ID 和密码登录，还可以实现 qq 邮箱登录。

导航条：设置的每一个导航栏都可以实现跳转。

商品列表：可以读取图片，新上传的商品在最前面显示。

商品详情页：js 实现数量加减。可以随时返回其他界面。

购物车：通过哈希列表取得信息，可以根据每样商品单价和数量计算该商品总价，以及计算该购物车所有商品总价。

订单：通过 session 提取用户的地址信息，哈希表提取所购商品信息并显示在页面上并计算出总价。若需要增加新地址，可以点击“新增”按钮后去个人中心新增地址并存储到用户个人信息数据库中，在提交前再一次罗列订单总价和用户地址信息，让用户确认无误后

提交订单。

界面：界面精美大气，操作体验极佳。

多个用户角色：多用户商城系统，覆盖了商家、消费者等多个用户角色，关系复杂，所以系统需要具备更精细的管理方案，为企业运营电商平台带来更高的效率。我们在登录注册时便区分了卖家与买家的不同信息，使得用户可以根据自己的所选身份进行更多相应的操作。此外，我们在个人中心界面还做了卖家/买家不同的头像，体现用户个性化定制，使得操作感更逼真。

7、代码

1) shopList 商品列表

```
1.
<% request.setCharacterEncoding("utf-8");%>
<%
    String keyWord="";
    List<Goods> all=DAOFactory.getIGoodsDAOInstance(
    ).findAll(keyWord);
    Iterator<Goods> iter=all.iterator();
    %>

2.
<%
    while(iter.hasNext()) {
        Goods goods=iter.next(
    );
    %>

        <li>
            <div
                class="i-pic limit">
                    <a href="shopInfo1.jsp?gid=<
                        %=goods.getId()%>&name=<%=goods.getName()%>&price=<
                            %=goods.getPrice()%>&
                        gnumber=<
                            %=goods.getNumber()%>
```

```
&gstock=<
                        %=goods.getGstock()%>&
                    gcolor=<
                        %=goods.getGcolor()%>&
                    gmemory=<
                        %=goods.getGmemory()%>
                    &gsize=<
                        %=goods.getGsize()%>&
                    picture=<
                        %=goods.getPicture() %
                    "></a>

                    <p class="title fl"><%=g
                    oods.getName() %></p>

                    <p class="price fl"> <b>
                        &yen;</b> <strong><%= goods.
                        getPrice()%></strong> </p>

                    <p class="number fl"> 销
                        量
                        <span></span><%=goods.getNumber() %> </p>

                    </div>
                v> </li>
            <%
                }
            %>
```

2) shopInfo 商品详情页

```
1. <%
```

```
String gid=new String(request.getParameter("gid"));
```

```
String gname=new String(request.getParameter("gname").getBytes("ISO-8859-1"), "utf-8");
```

```
Float gprice=Float.parseFloat(request.getParameter("gprice"));
```

```
Integer gnumber=Integer.parseInt(request.getParameter("gnumber"));
```

```
Integer gstock=Integer.parseInt(request.getParameter("gstock"));
```

```
String gcolor=new String(request.getParameter("gcolor").getBytes("ISO-8859-1"), "utf-8");
```

```
String gmemory=new String(request.getParameter("gmemory").getBytes("ISO-8859-1"), "utf-8");
```

```
String gsize=new String(request.getParameter("gsize").getBytes("ISO-8859-1"), "utf-8");
```

```
String picture=new String(request.getParameter("picture"));
```

```
%>
```

2.

```
<form class="theme-signin" name="loginform" action="" method="post">
```

```
<div class="theme-options">
```

```
<div class="cart-title number">数量</div>
```

```
<dd>
```

```
<input type="button" onclick="sub()" value="-"/>
```

```
<input type="text" value="1" class="xpf" name="count" id="txt" >
```

```
<input type="button" onclick="add()" value="+"/>
```

```
<input type="submit" value="开始选购"/>
```

3.

```
<a id="LikBasket" title="加入购物车"
```

购物车

```
" href="addCart.jsp?gid=<%=gid%>
&gname=<%=gname%>&gprice=<%=gpri
ce%>&gnumber=<%=gnumber%>&gstock
=<%=gstock%>&gcolor=<%=gcolor%>&
gmemory=<%=gmemory%>&gsize=<%=gs
ize%>&picture=<%=picture%>&count
=<%=Integer.parseInt(request.get
Parameter("count"))%>"/><i></i>
加入购物车</a>
```

3) addcart

```
<%
String gid=new String(request.getParameter("gid"
));
String gname=new String(request.getParameter("gn
ame").getBytes("ISO-8859-
1"),"utf-8");
Float gprice=Float.parseFloat(request.getParameter("gprice"));
Integer gnumber=Integer.parseInt(request.getParameter("gnumber"));
Integer gstock=Integer.parseInt(request.getParameter("gstock"));
String gcolor=new String(request.getParameter("g
color").getBytes("ISO-8859-
```

```
1"),"utf-8");
```

```
String gmemory=new String(request.getParameter("
gmemory").getBytes("ISO-8859-
1"),"utf-8");
```

```
String gsize=new String(request.getParameter("gs
ize").getBytes("ISO-8859-
1"),"utf-8");
```

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

```
Integer count=Integer.parseInt(request.getParame
ter("count"));
```

```
Goods goods = new Goods();
```

```
goods.setGid(gid);
```

```
goods.setGname(gname);
```

```
goods.setGprice(gprice);
```

```
goods.setGnumber(gnumber);
```

```
goods.setGstock(gstock);
```

```
goods.setGsize(gsize);
```

```
goods.setGcolor(gcolor);
```

```
goods.setGmemory
```

```

(gmemory);
        goods.setPicture
        (picture);
        Hashtable cart=(
        Hashtable)session.getAttribute("
        cartList");
        if(cart==null)ca
        rt=new Hashtable();
        Integer fnum;
        if(cart.contains
        Key(goods)) {
            fnum=(In
            teger)cart.get(goods);
            fnum+=co
            unt;
        }
        else{
            fnum=new
            Integer(count);
        }
        cart.put(goods,f
        num);
        session.setAttri
        bute("cartList",cart);
        %>
        <jsp:forward page="shopC
        art.jsp"/>

```

4) shopCart 购物车

```

1.
    <%
        request.setC
        haracterEncoding("utf-8");
        Hashtable ca
        rtList=(Hashtable)session.getAtt
        ribute("cartList");
        float sum=0;
        if(cartList!
        =null) {
            Set<Go
            ods> ks=cartList.keySet();
            Iterat
            or<Goods> it=ks.iterator();
            %>
2.
    <%
        Float total=Float.parseFloat(c
        artList.get(i).toString())*i.g
        etGprice();
        sum+=total;
        %>
3.
    <%
        }
        session.setAttribute("order",car
        tList);
    }
    %>

```

5) pay 生成订单

```
<%
    request.setCharacterEncoding("utf-8");
    Hashtable orderList=(Hashtable)session.getAttribute("order");
    float sum=0;
    if(orderList!=null) {
        Set<Goods> ks=orderList.keySet();
        Iterator<Goods> it=ks.iterator();
    }
%>
```

6) Addorder

```
<%
    ShopCart shopcart=new ShopCart();
    shopcart.setGid(request.getParameter("gid"));
    shopcart.setGname(request.getParameter("gname"));
    shopcart.setGcolor(request.getParameter("gcolor"));
    shopcart.setGprice(Float.parseFloat(request.getParameter("gprice")));
    shopcart.setGnumber(Integer.parseInt(request.getParameter("gnumber")));
```

```
    shopcart.setGallprice(Float.parseFloat(request.getParameter("gallprice")));
    shopcart.setPicture(request.getParameter("picture"));
    String msg="结算失败";
    if(DAOFactory.getIShopCartDAOInstance().doCreate(shopcart))
    {
        msg="开始结算! ";
    }else{
        msg="结算失败! ";
    }
}
```

7) 增加商品

```
1.
商品编号:
<input type = "text" name="gid">
<br><br><br>
商品名称:
<input type = "text" name="gname">
<br><br><br>
商品价格:
<input type = "text" name="gprice">
<br><br><br>
商品库存:
<input type = "text" name="gstock">
<br><br><br>
```


;

```
ter("gcolor"));
```

```
eter("gmemory"));
```

```
er("gsize"));
```

```
eter("picture"));
```

$$\text{try}\{$$

```
ce().doCreate(good)) {
```

 $\%>$

商品信息添加成功! ☐

 ≤ 0

```
} else {
```

 $\%>$

<0%

}

```
} catch (Exception e) {
```

```
e.printStackTrace();
```

}

8) 登录 login

```
function validate(f) {
```

```
if(! (/^\w{5,15}$/).test(f.userid.
```

$$\text{value})) \{$$

```
alert("用户 ID 必须是 5—15 位！
```

```

");
f.userid.focus();
return false;
}
if(!(/^\w{5,15}$/.test(f.userpass.value))) {
    alert("密码必须是 5—15 位!");
    f.userpass.focus();
    return false;
}
return true;
}
</script>

2.
<%=request.getAttribute("info")==null?"":request.getAttribute("info")%>

3.
List<String> info=(List<String>)
request.getAttribute("info");
    if(info!=null){
        Iterator<String> iter=info.iterator();
        while(iter.hasNext()){
            %>
            <h4><%=iter.next() %></h4>
4>
            <%
            }

}
    }
    %>
}

4.
<script>
    function login() {
        var user=document.getElementById("user");
        //获取账户信息
        var password=document.getElementById("password");
        //获取密码信息
        if(user.value!=='mr' && password.value!=='mrsoft' ) {
            alert('您输入的账户或密码错误! ');
        }else{
            alert('登录成功! ');
        }
    }
</script>

9) LoginServlet
public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    String path="login.jsp";

```

```

String userid=request.getParameter("userid");
String userpass=request.getParameter("userpass");
List<String>info=new ArrayList<String>();
if(userid==null||"".equals(userid)) {
    info.add("用户 id 不能为空!");
}
if(userpass==null||"".equals(userpass)) {
    info.add("密码不能为空!");
}
if(info.size()==0) {
    User user=new User();
    user.setUserid(userid);
    user.setUpassword(userpass);
    try {if(userid.length()==10){
        if(DAOFactory.getIUserDAOInstance().findLogin(user)) {
            //<jsp:forward page="success.jsp" />
            HttpSession session=request.getSession();
            path="usersale.jsp";
            info.add("卖家登陆成功, 欢迎"+user.getUserName()+"光临!");
            session.setAttribute("user", user);
        }else {
            info.add("卖家登录失败, 错误的用户名和密码!");
        }
    }catch(Exception e) {
        e.printStackTrace();
    }
    request.setAttribute("info", info);
    request.getRequestDispatcher(path).forward(request,response);
}
}
info.add("卖家登录失败, 错误的用户名和密码!");
}
}
else {
    if(DAOFactory.getIUserDAOInstance().findLogin(user)) {
        //<jsp:forward page="success.jsp" />
        HttpSession session=request.getSession();
        path="shopList.jsp";
        info.add("买家登陆成功, 欢迎"+user.getUserName()+"光临!");
        session.setAttribute("user", user);
    }else {
        info.add("买家登录失败, 错误的用户名和密码!");
    }
}
}catch(Exception e) {
    e.printStackTrace();
}
}
request.setAttribute("info", info);
request.getRequestDispatcher(path).forward(request,response);
}
}

```