

基于 MVC 模式的网上购物系统的设计与实现

冯晓强¹, 程晓昕²

(1. 四川大学计算机学院, 四川 610207; 2. 四川大学电子信息学院, 四川 610207)

摘要: 由于 MVC 模式的简单高效性, 越来越多的 Web 开发都采用了此模式。介绍基于 MVC 开发模式下的网上购物系统的系统构架以及系统设计, 此系统主要实现用户信息管理、订单管理、商品管理等功能, 并描述系统功能模块的实现, 介绍开发过程中所用的一些关键技术。

关键词: MVC; 网上购物; Servlet; JSP; JavaBean; JavaScript

0 引言

在 Internet 飞速发展的今天, 互联网成为人们快速获取、发布和传递信息的重要渠道, 它在人们政治、经济、生活等各个方面发挥着重要的作用。其中最具代表性之一的就是网上商店的诞生, 网上购物系统向消费者展示出一片新鲜的购物天地, 网上购物弥补了传统购物的一些局限性, 成为传统购物的补充。

1 系统构架

本系统采用了三层架构的 MVC 模式, MVC 有 3 个主要组件: 模型(Model)、视图(View)、控制器(Controller)。

Model: 模型是用于描述外部世界使用程序的业务逻辑和数据;

View: 视图主要涉及程序向用户界面的输出;

Controller: 控制器处理用户的输入。

本系统开发基于 JSP+JavaBean+Servlet 模型, JSP 充当视图, JavaBean 充当模型, Servlet 充当控制器。

MVC 模式的工作流程如图 1。

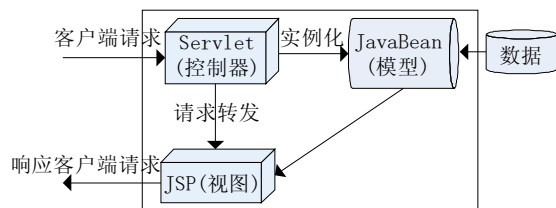


图 1 MVC 模式的工作流程图

2 系统设计

2.1 系统主要功能设计

该系统主要是为广大用户提供更加方便的购物途径, 其主要功能模块如图 2。

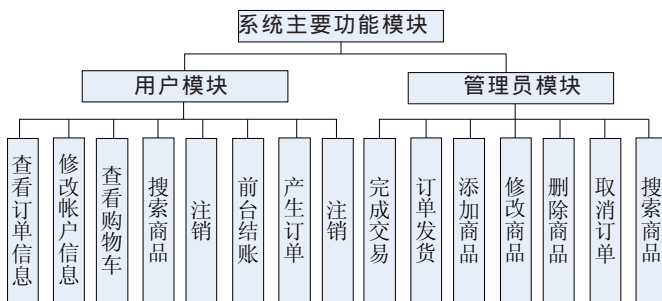


图 2 系统主要功能模块图

2.2 数据库设计

(1) 系统中各实体及其联系

本系统共有 5 个实体: 管理员 admin、商品信息 goods、订单 orders、订单详细信息 orders_detail、用户信息 userinfo。

系统 E-R 图如图 3。

(2) 根据转换算法可将 E-R 图转换为关系模式如下:

● 管理员: admin(adminname, passwords);

● 商品信息: goods(goods_id, goods_name, goods_bigtype, goods_smalltype, goods_describe, goods_price,

收稿日期: 2009-04-07 修稿日期: 2009-07-18

作者简介: 冯晓强(1987-), 男, 山西朔州人, 在读本科

goods_discount,goods_number,goods_pic,sale_number);

● 订单 :orders(order_id,username,order_date,order_address,contactman,postcode,order_sum,order_dealdate,order_state,order_phone);

● 订单详细信息 :orders_detail(order_id,goods_id,goods_name,goods_price,buy_number,total_money);

● 用户信息 :userinfo (username,passwords,real-name,email,gender phone,seleinfo);

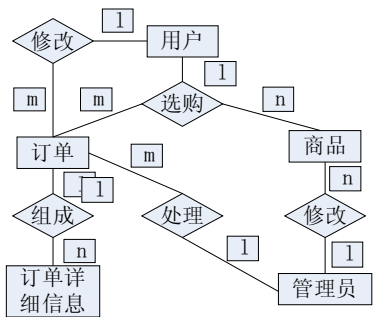


图 3 系统 E-R 图

3 主要功能模块的实现

该系统主要的功能模块有商品查询、商品购买、订单管理和用户登录与注册。数据库采用 Postgresql, 用 JDBC 实现与数据库的连接和操作。编写了一系列 Servlet 类集中处理事务,JavaBean 组织数据的结构, JSP 页面显示数据。

● 主要的 Servlet 类

AdminServlet.java: 负责处理管理员的事务, 包括商品管理和订单管理两大部分;

ConsumerServlet.java: 负责处理用户的日常请求, 包括页面跳转、查看商品详细信息、搜索商品等;

GoodscartServlet.java: 实现了购物车的功能, 主要处理添加商品到购物车、删除购物车内的商品、结账并产生定单等事务;

LoginServlet.java: 负责登录功能的事务处理, 包括管理员和普通用户的登录, 实现了对用户名和密码的检查;

RegisterServlet.java: 负责处理用户注册的请求, 实现了对用户名的检查使用户名唯一;

UpdateServlet.java: 负责对更新用户信息的功能。

● 主要 JavaBean 类

DB_Conn.java: 连接数据库, 获得数据库连接对象;

DB_User.java: 用户事务逻辑, 封装了对用户数据

库的操作, 包括增加、删除用户记录, 获得用户信息, 更新用户信息;

DB_Product.java: 商品事务逻辑, 封装了对商品数据库的操作, 包括查询商品、添加商品到数据库、删除商品和更新商品信息;

DB_Order.java: 订单事务逻辑, 封装了对订单数据库的操作, 包括查询订单信息、更新订单信息、添加和删除等;

DB_Order_detail.java: 封装了对订单详细信息的处理;

CartBean.java: 实现购物车, 用来处理购物车的请求;

CartProduct.java: 购物车商品信息类, 封装了关于购物车内商品的信息;

User.java: 用户信息类, 封装了用户的基本信息;

Product.java: 商品信息类, 封装了商品的基本信息;

Order.java: 订单信息类, 封装了订单的基本信息;

Order_detail.java: 订单信息信息类, 封装了订单的详细信息。

● 主要的 JSP 页面

index.jsp: 系统主页, 上方设计有导航条, 指导全站的视图;

myaccount.jsp: 显示我账户的信息;

searchview.jsp: 该页面用来查询商品, 显示查询结果;

cartview.jsp: 显示购物车中的商品;

checkout.jsp: 结算页面, 对购物车里的商品进行结算;

listorder.jsp: 显示已经产生的订单;

searchorder.jsp: 该页面用来查询订单, 显示查询结果;

detail_order.jsp: 显示订单的详细信息;

modifyorder.jsp: 修改订单的信息;

listproduct.jsp: 列出所有的商品并显示相关信息;

addproduct.jsp: 添加商品页面, 供用户输入关于商品的相关信息;

deleteproduct.jsp: 删除商品页面;

modifyproduct.jsp: 修改商品信息页面;

tuijian.jsp: 动态显示显示推荐商品信息;

tejia.jsp: 动态显示特价商品信息;

login.jsp: 登录页面。

4 开发过程中的关键技术

4.1 JavaScript

在该网上购物系统中就用到了这种脚本语言,增强了程序的交互性。以下是程序中用户注册部分的一小段 JavaScript 代码:

```
<script language="javascript">
    function Check()
    {
        if(document.register.username.value=="")
        {
            window.alert("请填写你的登录名!");
            window.register.username.focus();
            return false;
        }
        if(document.register.pwd.value=="")
        {
            window.alert("请填写密码!");
            window.register.pwd.focus();
            return false;
        }
    }
</script>
```

通过该小段代码,可以轻松地判断用户是否输入了用户名和密码,如果没有填写,点击“提交”时,就会自动地弹出一个对话框“请填写用户名”或“请填入密码”提示用户必须填写的内容,具有很好的实时交互性,而不用单独写一个 Servlet 去处理该情况,使程序更易编写,且效果较好。

4.2 CSS

传统 Web 站点中的 HTML 页面文档结构和显示格式一直是混合在一起的,这给页面开发和调整带来了许多不便。另外,仅仅通过 HTML 进行字体显示格式的调整是很粗糙的,并不能达到某些特定效果的要求。为了使页面显示格式完全独立于文档结构,以及为了能够更加精确地设置文档显示效果,W3C 为 HTML 制定了样式单机制,这就是 CSS。

运用 CSS 来统筹全站的面版,使页面代码更加简洁移动。当需要改动的时候,只需对 CSS 样式表进行修改,就可以使全站的页面元素得到修改,降低了网站维护的难度。

4.3 验证码的生成

验证码是指将一系列随机产生的数字或特殊符号叠加到一幅图像里,同时在图像里加上一些干扰信息,用于防止恶意破坏的验证信息。验证码的作用主要是为了防止恶意用户利用机器人程序自动注册、登录,以达到防止无限申请账号从而破坏服务器或暴力破解密码的目的。

该网上购物系统使用了验证码技术,程序中 Random.jsp 的功能是随机产生一幅验证码的图像,并发送回浏览器,其实质是一个在 JSP 中产生动态图像的程序,Random.jsp 实现了验证码的生成。

会员登录系统时验证码的界面如图 4。



图 4 会员登录系统界面图

5 结 语

本系统运用 MVC 模式实现了网上购物功能,基本可以满足广大上网购物用户的需求。MVC 开发模式中的三层各司其职,其中一层发生变化不会影响其他层,而且有利于责任分工,因此运用此模式开发相关系统可以提高工作效率。

参考文献

- [1]邹竹彪. JSP 宝典[M]. 电子工业出版社,2007
- [2]王珊,萨师煊. 数据库系统概论(第四版)[M]. 高等教育出版社,2006
- [3]夏慧军. JavaScript 典型应用与最佳实践[M]. 电子工业出版社,2008
- [4]Kynn Bartlett 著. 周哲,和亲玉等译. CSS 入门经典[M]. 人民邮电出版社,2007
- [5]张银鹤,冉小晏,刘治国等. JSP 完全学习手册[M]. 清华大学出版社,2008

Design and Implementation of Online Shopping System Based on MVC Model

FENG Xiao-qiang¹ , CHENG Xiao-xin²

(1. College of Computer, Sichuan University, Sichuan 610207;

2. College of Electronics and Information, Sichuan University, SiChuan 610207)

Abstract: Because of simpleness and high efficiency of MVC model, more and more Web developments adopt it. Describes system architecture and system design of online shopping system based on MVC model, the system mainly implements some functions such as user information management, order management, goods management and so on, describes the implementation of system function modules, describes some key technologies used in the development process.

Keywords: MVC; Onine Shopping; Servlet; JSP; JavaBean; JavaScript

(上接第 173 页)

A Heterogeneous Information Integration Solution Based on Web Service

ZHAO Yuan¹, LV Hong¹ , YAN Jian²

(1. Naval Aeronautical Engineering Academy, Yantai 264001; 2. Shandong Transport Vocational College, Weifang 261206)

Abstract: On the basis of the traditional heterogeneous information system, proposes a scheme for integration of heterogeneous information based on Web service, and also setups a sort of encapsulation protocol REML. The reuse of various heterogeneous information system is realized with Web service, so that the traditional information system can be accessed by the Web service users without the labor of redeveloping a new system and the sharing of heterogeneous information is realized.

Keywords: Heterogeneous Information; Web Service; ISS; REML