|  |
| --- |
| **基于ThinkPHP框架的点餐系统** |

|  |
| --- |
| Point meal system based on the ThinkPHP framework |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 杨超 | 学号 | | 201311671227 | |
| 所在学院 | 数学与计算机学院 | | 班级 | | 信管1131 |
| 所在专业 | 信息管理与信息系统 | | | | |
| 申请学位 | 工学学士学位 | | | | |
| 指导教师 | 涂超 | | 职称 | | 副教授 |
| 副指导教师 |  | | 职称 | |  |
| 答辩时间 | 2017 年 6 月 4 日 | | | | |

**目 录**

[摘要 I](#_Toc20092)

**[abstract](#_Toc24794)** [II](#_Toc24794)

[前言 III](#_Toc452340863)

[1 绪论 1](#_Toc31658)

[1.1 研究背景 1](#_Toc3883)

[1.2 研究目的和意义 2](#_Toc23498)

[1.2.1 研究目的 2](#_Toc1980)

[1.2.2 研究意义 2](#_Toc6373)

[2 可行性分析 2](#_Toc1818)

[2.1 技术的可行性 3](#_Toc30048)

[2.1.1 PHP的开发语言平台 3](#_Toc26129)

[2.1.2 ThinkPHP框架的支撑 4](#_Toc28384)

[2.2 操作的可行性 7](#_Toc23421)

[3 需求分析 7](#_Toc31098)

[3.1系统总体需求分析 8](#_Toc18745)

[3.2 数据流图 8](#_Toc7500)

[4 系统设计 9](#_Toc21970)

[4.1　系统概要设计 9](#_Toc20975)

[4.2　系统体系结构 10](#_Toc24763)

[4.3　功能模块设计  10](#_Toc3681)

[4.2.1 总代理模块的功能设计  11](#_Toc3329)

[4.2.2 代理模块的功能设计 11](#_Toc31065)

[4.2.3 店铺模块的功能设计 11](#_Toc31065)

[4.3 数据库设计 11](#_Toc16092)

[4.3.1 数据库的E-R图设计  12](#_Toc2703)

[4.3.2 数据库的详细设计    13](#_Toc23414)

[5 系统实现 17](#_Toc18288)

[5.1 总代理功能模块的实现 19](#_Toc5326)

[5.2 代理功能模块的实现 23](#_Toc240)

[5.2 店铺功能模块的实现 23](#_Toc240)

[6 系统测试 27](#_Toc31242)

[6.1系统整体测试步骤 27](#_Toc20719)

[6.2测试用例 27](#_Toc21910)

[6.3测试结果 27](#_Toc31693)

[7 总结 28](#_Toc15378)

[鸣 谢 29](#_Toc13368)

[参考文献 30](#_Toc27605)

[附录一：前端代码 31](#_Toc17412)

[附录二：后台代码 37](#_Toc24789)

# 摘要

随着人民生活水平的不断提高，以及科学技术的日新月异，餐饮行业作为第三产业中的服务业，其发展也是突飞猛进。传统的餐饮业纯人工服务模式已经不能满足互联网+时代消费者的消费需求。因此，一款适应时代发展，推陈出新的智能终端电子点餐系统应运而生。

本点餐系统结合当前互联网时代的形势，综合分析当前餐饮行业的发展，打破了传统的所有点餐工序都由人工完成的现状。

本点餐系统采用PHP语言，基于ThinkPHP框架，以PHPstorm为开发工具，MySQL为数据库进行开发。为实现点餐系统的一体化管理，本系统有针对性的对结构进行了合理、科学的划分，本系统共有三个后台管理系统，一个前端点餐平台。其中三个后台管理系统包括：总代理后台、代理后台、店铺后台。其中三个后台管理系统的主要关系为：总代理生成代理，代理生成店铺，店铺对商品进行管理。

本点餐系统具有以下优点：高效智能，节约餐饮行业的用人和管理成本，提高餐馆的服务效率，使整个点餐、支付、配送服务协调高效运转；满足用户的个性化消费需求，提升用户的体验度。提升企业形象，改善餐饮行业的服务水平，促进餐饮行业的蓬勃发展。

关键词：点餐系统；结构；优点

# **Abstract**

With the continuous improvement of people's living standard, as well as the progress of science and technology, services in the food industry as the third industry, its development is by leaps and bounds. The traditional catering service model has not satisfied the consumer demand of the Internet + era. As a result, a new intelligent terminal electronic ordering system that ADAPTS to the development of The Times has emerged.

This order system in combination with the current situation of the Internet era, the development of the comprehensive analysis of the current food and beverage industry, breaking the traditional all order process completed by manual.

This ordering system USES the PHP language, which is based on the ThinkPHP framework, and is developed using the PHPstorm as a development tool. Ordering system to realize the integration of management, this system is targeted to the division of the rational and scientific structure, there are three background management system, the system platform endpoint before a meal. Three of the backend management systems include the agent background, the agent background, and the backstage. The main relationship between the three backend management systems is that the general agent generates the agent, the agent generates the stores, and the store manages the goods.

This order system has the following advantages: high intelligence, save food industry of choose and employ persons and management costs, improve the service efficiency of the restaurant, make the whole order, payment, delivery service efficient operation and coordination; Meet the user's personalized consumer needs and improve the user's experience. Promote the image of the enterprise, improve the service level of the catering industry, and promote the development of the catering industry.

**Key words:** ordering system; Structure; advantages

前 言

作为我国第三产业中的一个传统服务性行业而言，餐饮行业始终保持着旺盛的增长势头，取得了不可小觑的发展，市场潜力非常大[1]。但是，普遍的餐饮企业还是沿用传统的经营模式，通过纯人工操作的方式为顾客提供销售服务。这种通过人工完成顾客点餐过程中的各个环节，呈现出效率差，服务质量低的短板。并且，伴随着餐饮行业经营规模的不断扩大，客户流量的不断增加，传统的纯人工的面对面点餐操作方式面临更加严峻的挑战。众所周知，当今社会的生活节奏非常快，生活方式也在不断的朝着现代化方向改变，物联网时代也在渐渐深入人民的日常生活，传统的消费观念和消费方式必然受到冲击，新时代下的消费观念迫使餐饮企业改进其运营模式，提高其经营效率，提升服务质量，满足客户对方便快捷和个性时尚等方面的个性化需求，以提高客户的消费体验度。在“智慧地球”的大环境下，要想提升餐饮行业的餐饮行业工作效率，降低人工成本和增加盈利能力，信息化建设无疑是实现以上要求需求的有效手段。尤其是近些年来终端技术的蓬勃发展和智能移动终端设备的普及，如平板电脑和其他的一些终端机技术等，为餐饮企业的信息化建设推波助澜。

通过信息化建设来扭转传统的餐饮行业纯人工化的点餐流程，这正是本智能点餐系统所要实现的任务。

凭借智能终端技术的高速发展，本文采用PHP开发技术，探究了关于智能终端设备的电子点餐系统的实现方案，深入分析了终端智能电子点餐系统的关键技术、功能结构和体系结构等领域的问题。基于终端设备的智能点餐系统不仅能够帮助餐饮企业实现高效管理，降低运营成本，提高销售竞争力和全局运营监控能力，而且还可以解决顾客便捷快速点餐的问题，给客户一个满意的消费体验。

**基于ThinkPHP框架的点餐系统**

信息管理与信息系统，201311671227，杨超

指导教师：涂超

毕业设计说明书

# 绪论

## 研究背景

在当今的这个时代，计算机技术和软件技术在信息化的发展中占主导地位，它覆盖了很多领域，例如经济、政治、军事和通信，并且在社会文化、艺术、家电、娱乐等范畴都能看见其身影。随着现代传媒和互联网的普及及应用，网络通信、电子办公、网上缴费、网络一体化的进程加速加速再加速。

最近几年，物物相连的物联网也开始逐渐进入了人们的视野，改变和提高了人们的生活方式，在数字化和智能化的这股浪潮的冲击下，网上点餐系统这种新型的点餐方式便运营而生了。这种终端的点餐系统，记录了商铺的菜谱种类，销售情况和好评度，让消费者一目了然，不但节省了点餐的时间，而且商家可以通过系统的数据推出最受消费者喜爱的菜肴，极大的减少了消费者时间成本和商家的经营成本。

在实习期间，每天工作很辛苦，最大的感触是生活节奏加快，时间根本不够。比如，中餐和午休时间只有仅仅的1.5小时，也就是说午饭时间越短，午休时间就越充足，然而，大多数餐馆都还是沿用传统的点餐模式，选好座位，手动写单，开始消费，收到支付，整个流程下来，不仅耗费了时间，而且无形间就耗费了不少的宝贵时间，比如：从下单到接单的这一段时间差。并且，我们在点餐的时候，只能凭借一本菜谱，上面只有寥寥数笔的单价、名称和图片，我们根本不知道其他消费者的评价，有时候可能就点了自己不喜欢吃的菜也是很正常的。

商品的购买不再是传统低效的纸笔记录了，那些传统的营销模式也已经不能够满足现在社会的需要。当人们收集了大量的数据后，会先把它们保存起来然后再做近一步的处理进而抽取出有用的信息。在如今数据爆炸的时代，数据量急剧增长，借助计算机和数据库技术来保存大量的数据，以便能更好的利用这些数据资源。

所以，设计出一个有电子商务功能的企业网站是有非常好的发展前途与应用市场的。

本系统就是面向购物者的商城系统。此系统是一种比较智能化的管理系统，它面向所有喜爱看购物的用户。它能够实现商城的基本功能，包括新商品信息的录入、查询等操作以及后台数据库的备份和恢复。用户合法注册后必须输入有效密码才能成功进入此系统，可以进行购买商品，查询信息，统计信息等操作。对于非法操作，系统有识别和防护措施。

## 研究目的和意义

现代化的社会生活，使市场的动态走向发生了很大的变化，人们已经不再对生活的用品满足于丰衣足食的低度要求，很多人往往并不是为了满足生活的需要去购买商品，而是喜欢去购买。因而开发本系统的目的就是要帮助消费者更好的选择商品。

在开发一个商城的系统，所需要考虑到的是它面向所有用户，但具有安全性能。它能够实现商品购买的基本功能，包括新商品信息的录入、查询等操作。用户合法注册后必须输入有效密码才能成功进入此系统。系统在设计过程中，为了克服查询和统计方式不同等困难，需要使程序代码标准化，软件统一化，确保软件的可维护性和实用性；删除不必要的管理冗余，实现管理规范化、科学化；界面友好、简单、实用做到实用、方便，尽量满足商品的需求。可以通过本次课程设计加深对数据库系统、软件工程、Java语言、以及三大框架的理论知识的理解和应用水平；在理论和实践中再进一步的巩固已学基本理论知识并加以综合提高同时将学会的知识应用于实际的方法，提高分析和解决问题的能力，增强动手的能力。本次课程设计，使同学能够主动查阅与三大框架的相关资料，更好的巩固课堂上老师未曾教授的知识，从而达到培养学生自学以及主动解决问题的能力目的。

21世纪是—个以网络为核心的信息时代。数字化、网络化与信息化是21世纪的时代特征。商城是以一个网上销售、出库、入库及人员管理、商品管理、供应商管理的工作为核心，它的内容对于商店的决策者和管理者都是至关重要，在此基础上，我们开发了这套系统。目的就是为了解决商品管理的繁乱，使商店日常经营及销售中，能够有迹可寻，随时跟踪市场，而且减少不必要的劳力资本，使得经营者能够实现用最小的成本获得最大的利润。因此设计和开发商城，具有十分重要的社会意义和现实意义。

从系统的背景来看，可以看出开发一个商城有以下几点重大的意义：

**1**.**不关店的营业时间**。网上商店的营业时间，基本是每天开着不会关而且无须有专人值班看店，都可照常营业。传统店铺的营业时间而是不固定的，遇到特殊情况也会说不准。

**2.不受店铺大小的限制**。哪怕只是街边的小店铺，在网上却可以拥有像百货大楼那么大的店面，只要开店者愿意的话，可以尽情的摆商品。

**3**.**不受地理位置影响**。不管客户离店有多远，也不管顾客是国内还是国外的。只要顾客能够上网就可以很方便地找到并购买商品。

因为现代社会愈发的依赖于网络，所以为企业或政府设计一个网站是一件必不可少的事情。

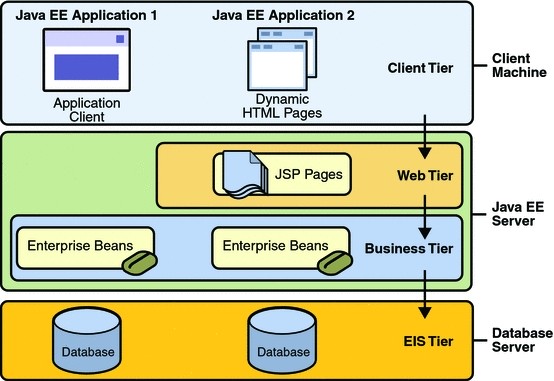
# 可行性分析

## 技术的可行性

### PHP的开发平台

Javaee是java企业版平台，它让模块化的java程序可以在服务器上部署。Javaee是一种多层体系的结构，而其是基于java语言，所以系统也和java语言一样，具有平台无关性，不受任何平台的限制，同时javaee也拥有它的特征就是：可扩张性，稳定的可用性，良好的安全性和可靠性。

Javaee一直遵循着四层架构模式，web层，业务逻辑层，数据层还有实体层。而相对应的层级关系如图2-1所示：



Javaee在现在的开发模式中一直沿用着MVC的开发模式：模型（Model）、视图（View）、控制器（Controller）。其主要目的其实就是将系统按照组成部分划分为三个层级，由中间的控制层进行对视图层和模型层的衔接，从而实现各层级之间相对独立的状态。控制层的作用是接收请求，获取用户的参数；调用业务逻辑处理方法；获取结果，导航视图，把结果显示视图中。

视图层的作用则是使用jsp技术把需要显示的数据在页面上进行显示。模型层的作用是使用 javabean技术进行对业务数据的封装。

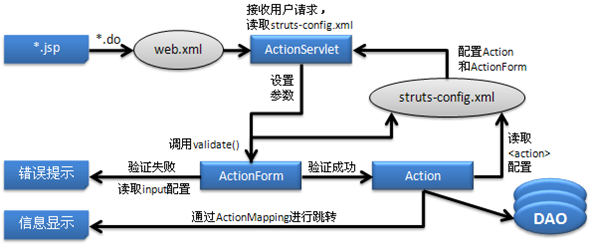
MVC可以理解为是一种设计模式，一种设计思想；而javaee架构是一种应用体系架构是在javase的基础上构建的，能够更好的帮助企业开发和部署可移植、可伸缩且安全的服务端java应用程序。现在来说，javaee的体系结构基本都是基于MVC模式开发的，层级的设计也是按照MVC三层的关系设计的。

### ThinkPHP框架的支撑

1. **Struts框架：**

Struts是一种基于MVC设计模式，普遍应用在JavaEE平台上的框架，该框架开发的初衷是在于方便开发人员开发，缩短项目的开发时间。它基本上集成了MVC设计模式的各项特征，并且结合JavaEE的技术类型，进行小范围的调整和扩展。Struts是一种轻量级的框架，使用Java Server Pages、标签库还有Servlet这三种技术组合实现。Struts操作简单，开发高效以及功能全面的特点，所以现在使它成为了最常用的框架之一。

Struts的框架流图如图2-2所示：



Struts的工作原理和体系结构：

模型部分（Model）：

Struts框架的模型层是由处理JSP页面同Action之间数据交互的层级，在业务层处理具体业务逻辑的层级实现的。这两种方式都是为了把业务逻辑数据封装起来，从而降低了系统的维护复杂程度以及提升了程序的可重用性。

控制器（Controller）：

ActionServlet和Action是Struts框架中控制层的两大主要组成部分，而ActionServlet是系统核心控制器，继承了HttpServlet的主要功能，是通过Struts.xml的主配置文件来实现Action和JSP页面的映射关系，达到拦截和分发用户请求。而Action是业务逻辑控制器，它主要是分发业务不是处理业务。

视图部分（View）：

在JSP页面通过Struts的标签库轻松灵活的实现了视图中的数据显示。在页面不需要添加java的代码，只是需要简单的标签就可以实现页面中的数据传递，还同时简化了页面的显示功能和控制页面显示的逻辑。

1. **Spring框架：**

Spring 是一个开源框架，是为了解决企业应用程序开发复杂性而创建的。框架的主要优势之一就是其分层架构，分层架构允许您选择使用哪一个组件，而且同时为J2EE 应用程序开发提供集成的框架。

Spring 框架是一个分层架构，由 7 个定义良好的模块组成。Spring 模块 构建在核心容器之上，核心容器定义了创建、配置和管理 bean 的方式。

组成 Spring 框架的每个模块（或组件）都可以单独存在，或者是和其他一个或多个模块联合实现。每个模块的功能如下：

核心容器：核心容器提供 Spring 框架的基本功能。核心容器的主要组件是 BeanFactory，它是工厂模式的实现。BeanFactory 使用控制反转 （IOC）模式将应用程序的配置和依赖性规范与实际的应用程序代码分开。

Spring Context：Spring Context是一个配置文件，向 Spring 框架提供Context信息。Spring Context包括企业服务，例如 电子邮件、JNDI、EJB、国际化。同时也需要在javaee项目中进行配置信息来获取Context信息。

Spring DAO：JDBC DAO 抽象层提供了有意义的异常层次结构，可用该结构来管理异常处理和不同数据库供应商抛出的错误消息。异常层次结构简化了错误处理，并且极大地降低了需要编写的异常代码数量（例如打开和关闭连接）。Spring DAO 的面向 JDBC 的异常遵从通用的 DAO 异常层次结构。

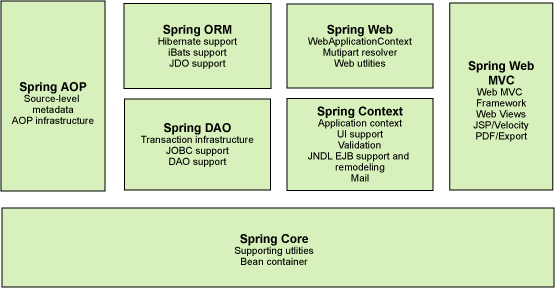
Spring AOP：通过配置管理特性，Spring AOP 模块直接将面向方面的编程功能集成到了 Spring 框架中。所以，可以很容易地使 Spring 框架管理的任何对象支持 AOP。Spring AOP 模块为基于 Spring 的应用程序中的对象提供了事务管理服务。

Spring Web 模块：Web 上下文模块建立在应用程序上下文模块之上，为基于 Web 的应用程序提供了上下文。所以，Spring 框架支持与 JakartaStruts的集成。Web 模块还简化了处理多部分请求以及将请求参数绑定到域对象的工作。

Spring ORM：Spring 框架插入了若干个 ORM 框架，从而提供了 ORM 的对象关系工具，其中包括 JDO、Hibernate 和 MyBatis SQL Map。所有这些 都遵从 Spring 的通用事务和 DAO 异常层次结构。

Spring MVC 框架：MVC 框架是一个全功能的构建 Web 应用程序的 MVC 实现。MVC 容纳了大量视图技术，其中包括 JSP、Velocity、Tiles、iText 和 POI。

spring的整体框架如图2-3所示：



1. **Hibernate框架：**

Hibernate是一个轻量级的处理数据持久层的框架，主要是通过对JDBC的封装，屏蔽了底层的数据库操作，实现了实体对象与数据库的映射关系，而实现可以去以面向对象的设计理念操作关系型数据库，有助于提高开发效率。访问数据库所需要做的就是读取持久化对象定义的xml映射文件。Hibernate不仅可以基本实现对于所有JDBC应用场景的覆盖，还可以适用在JSP等Web应用和java客户端程序Applets中使用。

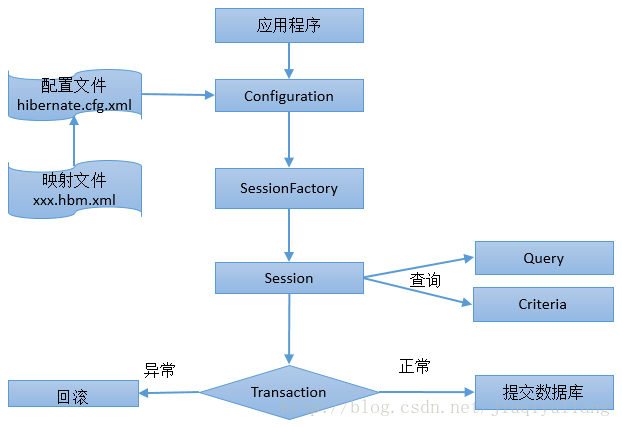
想要改变数据库的操作很简单，只需要改变配置文件（hibernate.cfg.xml或者hibernate.properties）上的参数即可，而对程序没有影响。

Hibernate另一个强大的地方是有自己的面向对象的查询语言HQL。HQL支持了大部分主流的数据库，比如Oracle，MySQL，SQL Server等。而Hibernate是基于对象关系映射O/R Mapping的持久层框架，因为大部分的数据库都是关系型数据库，所以需要通过ORM这种中间件的技术去建立程序对象和关系型数据库数据的映射关系。具体的过程是：先读取映射配置文件建立Java对象和数据库表之间的映射关系，然后就按照这种关系进行两者间的数据转换。

在HIbernate框架中，分为Session、SessionFactory、Transaction、Query、Criteria和Configuration六大核心接口。

1. Session接口：是通过SessionFactory生成的，是用于对持久对象进行增删改查操作的。
2. SessionFactory接口：是通过Configuration创建的，完成对Hibernate的初始化和对存储代理数据。可以看成是一种工厂模式，在通常情况下，只需要创建一个SessionFactory就可以对其进行管理。
3. Transaction接口：Transaction的作用是对系统的事务管理，根据自己的系统需求对底层数据库的事务进行管理。可以提高Hibernate框架在不同系统环境间的可移植性。
4. Query接口：是通过持久化对象对数据库进行查询，可以支持HIbernate的HQL语言；同时也可以支持SQL查询语言。
5. Criteria接口：该接口和Query接口相似，也是创建和执行面向对象的标准化查询。
6. Configuration接口：是配置参数和启动Hibernate框架，同时也是完成对SessionFactory对象的创建工作。通过映射文件里面的配置参数实现对SessionFactory的实例创建。

HIbernate的整体框架如图2-4所示：



## 操作的可行性

此系统具备友好的用户界面,使用方便，易于维护，操作简单便于被用户接受,用户只需熟练操作计算机，和对此系统使用做简单的了解即可方便使用,而且使用此系统可以减少管理人员的繁琐工作。因此从使用操作方面看，此系统的开发是可行的。

# 需求分析

## 3.1系统总体需求分析

商城系统开发的还有一个比较重要的阶段是需求分析阶段。需求分析就是描述系统的需求，需要对业务管理进行深入研究和分析，以便于构建业务原型，建立原型的目的是为了方便开发者和提出需求的人(用户，管理员)理解和沟通，以至于去机制的分析。

设计和实现商城系统，使用库存管理从而达到提高管理商品的效率。使程序既能满足材料库存管理的需要，又能方便操作。屏幕显示要直观与方便。

一般来说系统应有数据的修改、查询、录入、删除等操作集成在一起，这样，管理人员就可以很方便地进行有效地操作。另外，一般还应支持模糊查询和组合查询。这样不仅提高了操作的速度，也有效地减少了出错的几率。界面风格应当尽量向大众化方向发展。这样做的好处是能够使用户更容易掌握。

商城系统，首先必须具备的功能是记录仓库存货、销售以及进货情况，通过该系统去了解商品价格、库存数量还有商品的种类等信息，以便管理员根据以上信息做出营销对策。

商品管理：商品信息管理，商品信息包括商品数量、单价。商品管理功能完成信息登记、修改和删除等。

订单管理：订单信息的管理，包括订单中的商品信息，商品数量、总价等。

分类查看：可以根据商品类型来查看某种商品的销售情况海鸥更新库存

系统管理：用户密码修改，用户权限修改。

用户管理：用户信息、用户名、用户ID。

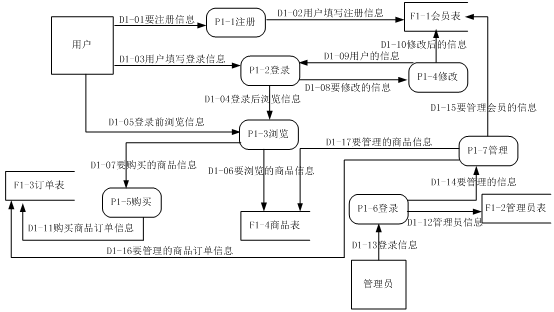
## 数据流图

数据流程图简称DFD，它用来描述目标系统的逻辑结构，它是由实体部分，处理部分，数据存储部分和数据流四部分组成。

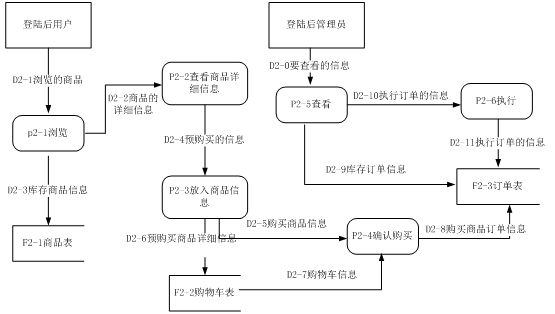
数据流程图主要是一种可以对信息系统进行比较全面的描述的工具，可以用最少的符号来表示信息在系统中的一个流动方向，以及储存情况等等。该数据流程图具有极强的概括性。抽象性主要的表现就是可以不看到具体的物质，关注的是数据的流动以及存储情况；概括性就是指把不同的信息通过各种业务流程相互联系起来，最终通过计算机来完成对于信息的处理。

它是表示系统逻辑模型的常用工具，图中不存在任何具体的物理元素，只表示信息在系统中流动和处理的情况。数据流图是逻辑系统的图形化表示。因此，它是系统分析员与用户进行交流的极好工具。

网上商城系统所设计的数据流图如图3-1所示：



用户可以以游客的身份进行搜索商品和浏览商品，还可以注册和进行搜索商品、浏览商品、查看商品的详细信息、加入购物车、提交订单、确认购买这些操作则需要用户登录，后台管理员可以对订单进行处理。订购商品数据流图如图3-2所示。



# 系统设计

## 4.1　系统概要设计

使用Mysql数据库和JSP进行对该网站的后台管理系统的设计开发，为了系统的安全性、可扩充性、实用性和易维护性的要求。前端页面采用了静态页面和动态页面相结合的设计。

本系统在设计时应该满足以下几个目标：

1.界面设计美观友好、信息查询灵活、方便、准确、数据存储安全可靠。

2.为顾客提供了一个方便、快捷的商品信息查询功能。

3.展示商城内的所有商品，并可以展示最新的商品和特价的商品。

4.用户可以修改自己的个人资料。

5.实现网上购物并可以真实的实现付款功能。

6.对用户输入的数据，系统进行数据检验，尽可能的排除不必要的错误。

7.用户随时都可以查看自己的订单。

8.系统运行稳定、安全可靠。

9.系统最大限度地实现了易维护性和易操作性。

## 4.2　系统体系结构

本网上购物系统是基于B/S架构，本网站系统的开发目的主要是为了方便用户购买商品。

本系统共有两类用户：

商城用户：这类用户可以查看商城的商品信息，查看各种分类商品信息，将商品添加到购物车中；根据用户名和密码登陆系统，登陆之后也可以修改自己的个人信息。修改购物车中商品数量，或者删除购物车中某些不想要的商品。

系统管理员：系统管理员主要负责管理系统后台信息；主要有订单管理，商品管理,类别管理和库存管理。订单管理有未付款订单和已付款订单管理还有未确认订单和已确认订单管理，未发货和已发货订单管理。商品管理主要有商品类别添加和商品类别管理，商品添加和商品管理。而库存管理则有对所有商品的库存数量进行管理包括是增加库存数量和修改库存数量等。

## 4.3　功能模块设计

层次模块结构是将系统划分为若干子系统，子系统下在划分为若干的模块。而模块是指具备逻辑功能、输入和输出功能、运行程序和内部数据一组程序。在结构化设计中，模块一般都是按功能划分的，通常称为功能模块。合理的功能模块的划分能够极大限度地提高开发工作的效率减少繁琐的程序编写并增大系统的可维护性。

综合前面的系统分析，商城系统的功能模块图如图4-1所示。

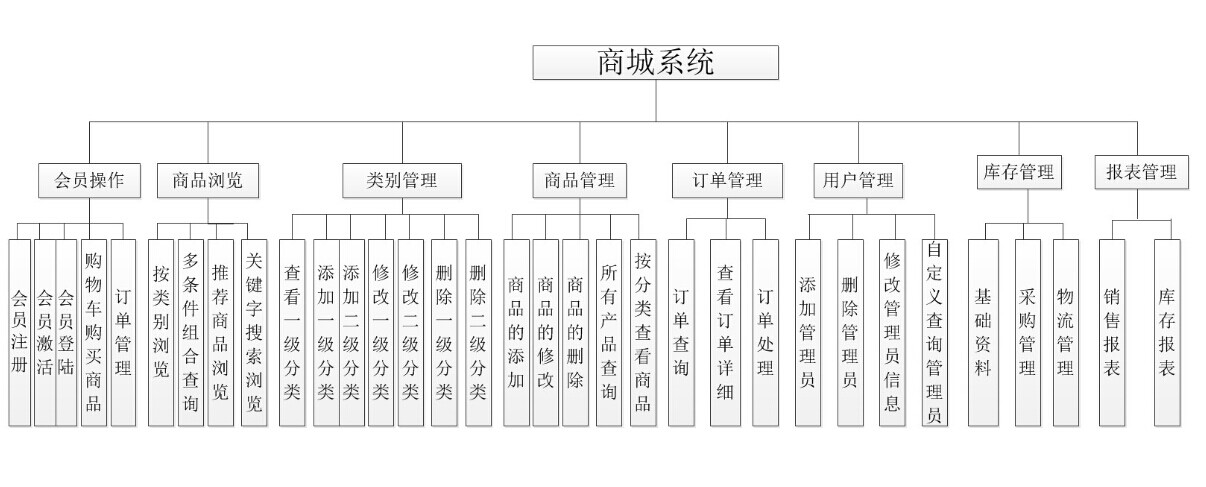


图4-1商城系统的功能模块图

### 总代理模块的功能模块设计

前台模块主要是为登录商城的用户提供商品信息和服务，添加商品并填写和提交订单的服务。如果把前台模块再细分的话可以在分为用户管理模块、修改个人资料、商品查询模块、购物车模块模块。下面会详细说明一下这些功能模块的设计：

1. 用户管理模块:为了方便于商城用户的管理。该商城用户管理模块主要实现用户的注册、登陆和修改密码还有查看并且修改个人信息等功能。  
    2.商品搜索模块：用户根据商品的名称可以搜索到所需要的商品信息。

3.购物车模块：在网上商城中，当用户选择了商品须通过购物车进行暂时的保存，然后生成订单。同时还可以随时添加、查看、修改、清空购物车中的商品。  
 4.个人密码管理模块：用户资料修改是让用户去修改个人信息而实现的，为了保护用户信息不受非法侵入。并且是只有当用户登录后才有权限修改自己的个人密码。

### 代理模块的功能模块设计

后台管理模块的功能是对商城的资源进行管理和维护。以后台的业务流程分析和前台各个模块的设计为基础，根据对商城系统的需求分析来确定后台管理模块应具有的功能。后台管理模块的各个子功能模块如下：

1.管理员身份验证模块：

为管理员提供了一个后台入口。该模块的功能是对管理员身份进行验证。管理员输入用户名和密码，如果通过验证则跳转后台主页，否则提示错误。

2.商品管理模块：向商品表插入商城首页中展示的商品信息，也就是添加商品信息的功能。

3.处理订单模块：管理者对用户的订单的执行情况可以进行处理。

4.用户信息管理模块：查询所有的注册用户，对一些非法或无效的用户进行删除操作。

### 4.2.3 店铺模块的功能模块设计

### 4.2.4 前台模块的功能模块设计

## 数据库设计

### 数据库E-R图设计

E-R图是描述系统数据库中各个表之间关系的图型，它能够直观的反应系统各个表之间的联系以及相互调用的关系。它的思想是用实体，实体的属性以及实体之间的关系来表示数据库系统的结构。

用矩形表示实体，矩形框内写明实体名；用椭圆表示实体的属性，并用无向边将其与相应的实体连接起来；用菱形表示实体之间的联系，在菱形框内写明联系名，并用无向边分别与有关的实体连接起来，同时在无向边旁标上联系的类型（1:1,1:n或m:n）即一对一，一对多和多对多。在E-R图中有如下四个成分：

1.矩形，表示实体，在框中记入实体名。

2.菱形，表示联系，在框中记入联系名。

3.椭圆，表示实体或联系的属性，将属性名记入框中。

4.连线 实体与属性之间；实体与联系之间；联系与属性之间用直线相连，并在直线上标注联系的类型。（对于一对一联系，要在两个实体的连线方向各写1； 对于一对多联系，要在一的一方写1，多的一方写N；对于多对多关系，则要在两个实体连线方向各写N和M，也可以写不同的字母来表示是多对多的关系）。

有正确的需求分析才能制造好的E-R图。本商城系统的E-R图如图4-2所示。



图4-2 E-R图

### 数据库的详细设计

根据对网上商城的实际情况分析，需要设计以下表。

**1. category（商品一级分类信息表）**

商品一级分类信息表主要用来保存商品类别中的一级类别信息。表category的结构如表4-1所示。

表4-1　商品一级分类信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| cid | varchar | 200 | 是 | 类别ID |
| cname | varchar | 255 | 否 | 类别名称 |
| cdesc | varchar | 100 | 否 | 类别描述 |

**2. categorysecond（商品二级分类信息表）**

商品二级分类信息表主要用来保存商品类别中的二级类别信息。表categorysecond的结构如表4-2所示。

表4-2　商品二级分类信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| csid | char | 200 | 是 | 类别ID |
| csname | varchar | 200 | 否 | 类别名称 |
| Csdesc | varchar | 100 | 否 | 类别描述 |

**3.cartItem（购物车表）**

购物车表用来保存用户在购买商品时添加的购物车信息。表cartItem如表4-3所示。

表4-3　购物车表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| cartItemId | char | 200 | 是 | ID |
| quantity | int | 11 | 否 | 数量 |
| Pid | char | 200 | 否 | 产品ID |
| Uid | char | 200 | 否 | 用户ID |

**4.orderitem（订单明细表）**

订单明细表是用来保存订单的详细信息。表orderitem如表4-4所示。

表4-4　订单明细表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| itemID | char | 200 | 是 | ID号 |
| count | int | 11 | 否 | 数量 |
| subtotal | double |  | 否 | 小计 |
| pid | char | 200 | 否 | 产品ID |
| oid | char | 200 | 否 | 订单ID |
| index | Int |  | 否 | 索引 |

**5.orders（订单信息主表）**

订单信息主表用来保存订单的相关信息。表orders如表4-5所示。

表4-5　订单信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| oid | char | 200 | 是 | 订单编号 |
| total | double |  | 否 | 合计 |
| Ordertime | datetime |  | 否 | 订单时间 |
| addr | varchar | 50 | 否 | 联系地址 |
| uid | char | 200 | 否 | 用户ID |
| phone | varchar | 50 | 否 | 电话 |

**6.t\_user（用户信息表）**

会员信息表主要用来保存所注册的用户的信息。表t\_user如表4-6所示。

表4-6　会员信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| uid | varchar | 200 | 是 | 会员ID |
| username | varchar | 255 | 否 | 用户名 |
| name | varchar | 255 | 否 | 真实姓名 |
| password | varchar | 255 | 否 | 密码 |
| addr | varchar | 255 | 否 | 联系地址 |
| state | int |  | 否 | 状态 |
| phone | varchar | 255 | 否 | 电话 |
| email | varchar | 255 | 否 | Email地址 |

**7.t\_admin（管理员信息表）**

管理员信息表用来保存管理员信息。表t\_admin如表4-7所示。

表4-7　管理员信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| uid | char | 32 | 是 | ID号 |
| username | varchar | 50 | 否 | 登录名 |
| password | varchar | 50 | 否 | 密码 |
| name | varchar | 50 | 否 | 真实姓名 |
| Sex | char | 10 | 否 | 性别 |
| phone | varchar | 50 | 否 | 电话 |
| Email | varchar | 50 | 否 | Emial |
| role | varchar | 20 | 否 | 管理员类型 |
| udate | datetime |  | 否 | 新增时间 |

**8.product（商品信息表）**

商品信息表主要用来保存商品的信息。表product如表4-8所示。

表4-8　商品信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| pid | char | 200 | 是 | 商品ID |
| pname | varchar | 255 | 否 | 商品名称 |
| Market\_price | double |  | 否 | 市场价格 |
| Shop\_price | varchar | 50 | 否 | 商城价格 |
| Image | varchar | 255 | 否 | 图片 |
| Image1 | varchar | 255 | 否 | 图片1 |
| Image2 | varchar | 255 | 否 | 图片2 |
| Image3 | varchar | 255 | 否 | 图片3 |
| Image4 | varchar | 255 | 否 | 图片4 |
| Is\_hot | int | 11 | 否 | 是否热门 |
| Is\_speciall | int | 11 | 否 | 是否推荐 |
| pdate | datetime |  | 否 | 上架时间 |
| csid | char | 200 | 否 | 所属二级 |
| abbreviate | varchar | 50 | 否 | 产品描述 |
| ReqCount | int | 11 |  | 库存数量 |

**9.sale（销售统计信息表）**

销售统计表主要用来统计商品的销售情况的数据信息。表sale如表4-9所示。

表4-9　销售统计信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| saleid | char | 32 | 是 | 商品ID |
| pname | varchar | 255 | 否 | 商品名称 |
| Shop\_price | double |  | 否 | 市场价格 |
| sdate | datatime |  | 否 | 销售时间 |
| pid | char | 200 | 否 | 商品pid |
| abbreviate | varchar | 50 | 否 | 缩略名 |
| scsid | varchar | 255 | 否 | 二级分类 |

**10.login\_log（后台登录日志表）**

后台登录日志表主要用来统计后台访问压力。表login\_log如表4-10所示。

表4-10　后台登录日志表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| login\_id | char | 32 | 是 | 登陆id |
| login\_date | datatime |  | 否 | 登陆时间 |
| login\_ip | varchar | 255 | 否 | 登陆ip |
| login\_name | varchar | 50 | 否 | 用户名 |

**11.login\_info（后台登录数据表）**

后台登录数据表主要用来统计后台访问压力。表login\_info如表4-11所示。

表4-11　后台登录日志表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| c | varchar | 50 | 是 | 时间 |
| d | int |  | 否 | 次数 |

**12.loginforuser\_log（前台登录日志表）**

前台登录数据表主要用来统计前台访问压力。表loginforuser\_info如表4-12所示。

表4-12　前台登录日志表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| login\_id | char | 32 | 是 | 登陆id |
| login\_date | datatime |  | 否 | 登陆时间 |
| login\_ip | varchar | 50 | 否 | 登陆Ip |
| login\_name | varchar | 50 | 否 | 用户名 |

**13.loginforuser\_info（前台登录数据表）**

前台登录数据表主要用来统计前台访问压力。表loginforuser\_info如表4-13所示。

表4-13　前台登录日志表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 | 描述 |
| a | varchar | 50 | 是 | 时间 |
| b | int |  | 否 | 次数 |

# 系统实现

在本章将讨论如何实现功能模块，最终实现整个商城系统。经过这个阶段的设计工作，应该得出对目标系统的详细描述。

## 5.1前台功能的实现

该商城系统一共分成了前台功能模块和后台功能模块。在每个模块的介绍中，将具体说明一些涉及要点以及核心的模块。

前台功能模块分为订单查询模块、用户管理模块、购物车模块、修改个人信息模块和商品展示模块。商城主页主要有以下几个部分：

1.**网站导航**：用于用户更方便的看到导航信息；

2.**用户登录**：用户能够在这里注册、登录或找回密码；

3.**商品分类搜索**：用户按商品类别进行搜索显示；

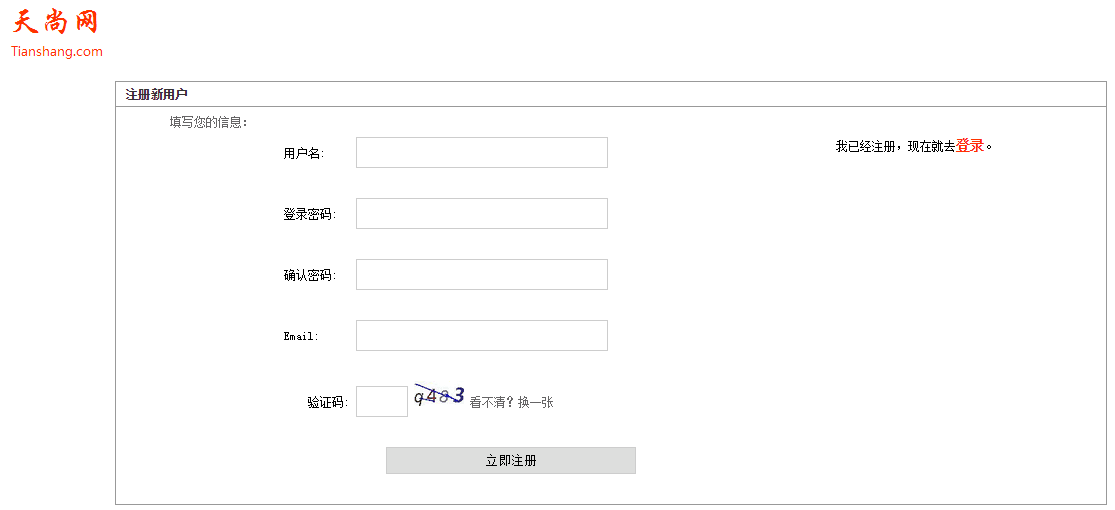
4.**热卖**：按照商品的热销情况显示商品；

5.**商品展示列表**：按后台添加商品的时间顺序展示商品信息。

首页运行页面如图5-1所示：



用户注册页面如图5-2所示：



为了便于网站对用户的管理，在用户注册页面接收到的注册信息要进行过滤，每个用户名只能是唯一的，如果注册时所填写的用户名出现了重复则会提示，在提交注册信息后需要对所填的数据在数据库中进行查询，如果没有找到，则插入数据库新的用户信息并在注册页面提示注册成功，否则提示此用户已被占用请重新注册。用户注册的程序流程图如图5-3所示：



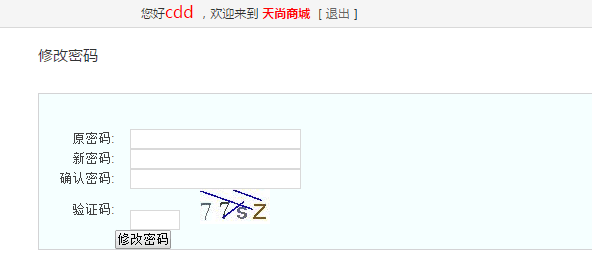
当用户登录系统的时候，登录页面如图5-4所示：



用户修改个人资料时是只有当用户只有登录商城后才有权限修改个人密码。用户修改个人信息模块在前台主页的导航条上，单击“我的信息”按钮，进入用户修改页面进行修改个人信息，用户修改个人信息页面如图5-5所示：



同时也可以对用户的密码进行修改，也是只有当用户登录了商城以后才可以进行对自己的密码进行修改，页面如图5-6所示：



修改密码的业务流程是如图5-7所示：



购物车模块中，主要包含以下几个功能，即：添加购物车、查看购物车、生成订单和清空购物车。

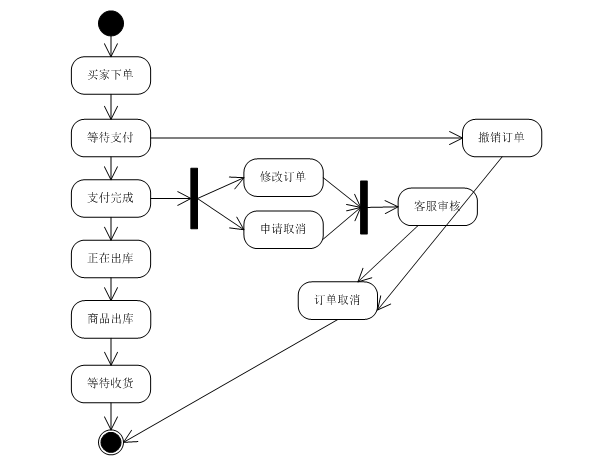
1.**添加购物车** 添加购物车是把用户选中的商品暂时存放在购物车中，当用户在商品明细页面中单击“购买”按钮的时候，系统会将该商品的详细信息展示在查看物品清单页面中。用户在单击物品清单页面下方的“放入购物车”链接，便可以将该商品放入购物车中，在购物车中的商品，如果没有该商品则添加进去，如果已经存在，则把商品数量修改。

2.**查看购物车** 为了方便用户随时查看购物情况，在个人的选择项那里添加了查看购物车的功能，通过它可以将用户放入购物车中的商品显示出来。对选中的商品想要修改或者删除的话，可以在数量那栏进行修改，除了修改商品的数量之外还可以通过输入数量为0把该选中的商品进行删除。

系统每次只会将1个商品放入购物车中，如果用户需要多个同种商品，可以通过修改商品信息右侧相应数量的值来完成。操作完成后需要通过单击“修改”按钮来保存操作。

1. **生成订单** 生成订单是网上购物商城的最终目的同时也是为了对该商品进行付款的前提，前面所有功能的实现都是为最后生成一个用户满意的订单作基础，在此要生成一个可供用户随时查询的订单号，还要保存用户订单中所购买的商品信息。当用户确认对购物车不再改变以后，就可以生成订单并到收银台结帐，完成商品的购买。对于生成唯一订单号方法有很多，主要就是确保订单号码的唯一性及方便用户记录以便于查询自己订单的执行状态即可。订单生成后，用户单击“提交”按钮，便可以将录入的订单信息保存到数据库中。
2. **清空购物车** 清空购物车是指当用户生成订单后，如果还想继续购物的话就要清空购物车再进行选择商品，这样能够防止重复购物。

购买商品的流程，如图5-8所示：



而在购物车的模块中，显示的商品信息则如图5-9所示：



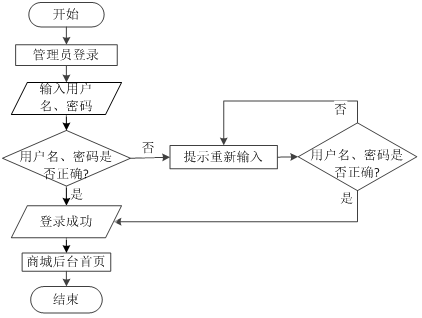
## 5.4后台功能的实现

后台功能是给管理员登录使用的。后台功能包括管理员身份验证、添加商品模块、订单信息管理模块、查询用户信息管理模块和测试压力图模块等。

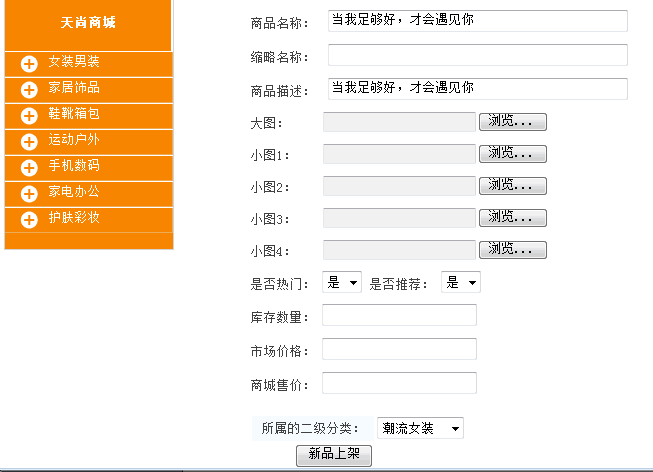
当管理员登录的时候需要验证用户名和密码，如果输入错了用户名或者错了密码，系统会提示错误信息，当输入正确的用户名和密码则显示登录成功，登录商城后台首页。管理员有权对网站进行维护，反之则提示错误。管理员登录页面如图5-10：



管理员登录的流程图如图5-11所示：



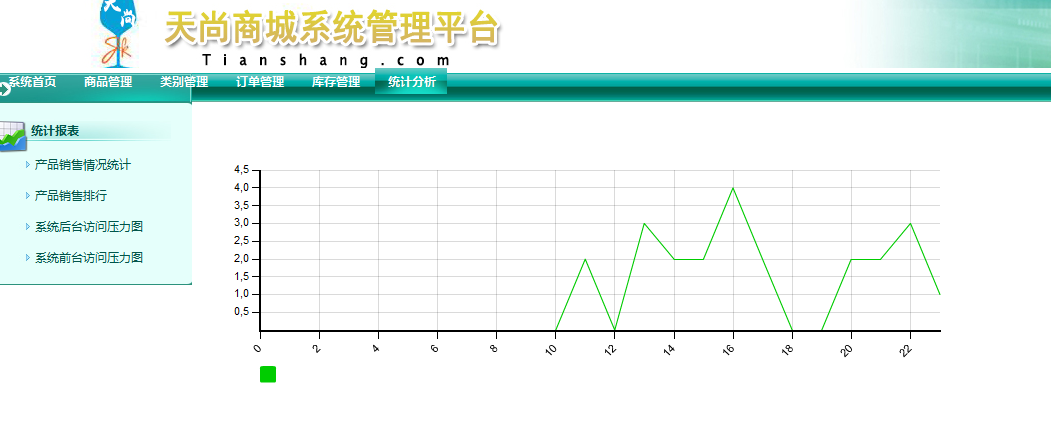
添加商品，通过管理员账号登录后台管理系统，然后选择“添加商品”的选项，系统会打开添加商品的页面，该页面主要是用于将管理员保存的商品信息保存在数据库中的，保存在数据库中的数据是用于提供到前端页面展示商品列表的。当用户点击了“商品添加”的按钮后添加商品及其信息，确认所添加的商品信息没有错之后，通过“新品上架”的按钮添加商品页面，详细如图5-12所示：



后台管理系统会根据每个登录后台的用户所在的时间段进行一个压力统计折线图，包括登录了前台页面的用户所在的时间段也会做一个压力统计折线图，主要是为了更加客观地显示压力数据。

考虑到如果每天都是用日期记录数据的话会显示的很臃肿的数据，所以采用了根据所登录的小时作为压力图的x轴的参数。

后台压力折线图如图5-13所示：



在后台管理系统，能够选择库存管理按钮，查询显示出所选择的库存数量的商品，同时也可以对该商品进行进货操作，实现库存管理。

后台库存管理操作如图5-14所示：



当用户对所选择的商品进行购买生成订单以后，会在后台管理系统可以查看到所有的订单，并能根据订单的状态去选择查看该选择的状态的订单，并能查看该订单的详细情况，点击详情之后会出现该订单所购买的商品。

订单管理操作的如图5-15所示：



对商品类别操作管理，是用于在前台的分类一栏显示，对于类别可以进行添加一级分类，也可以添加二级分类，也可以对分类进行修改和删除操作。

商品分类的操作管理页面如图5-16所示：



# 系统测试

## 6.1系统整体测试步骤

软件测试的主要步骤有单元测试、集成测试和确认测试这三个方面。

1．**单元测试**（Unit Testing）

单元测试。通常单元测试可放在编码阶段，开发人员在编写好一个模块后，总会对自己编写的模块进行测试，检查它是否实现了详细设计说明书中规定的模块功能和算法。单元测试主要发现编码和详细设计中产生的错误，通常采用白盒测试。

2．**集成测试**（Integration Testing）

集成测试也称组装测试，它是对由各模块组装而成的程序进行测试，主要检查模块间的接口和通信。集成测试主要发现设计阶段产生的错误，通常采用黑盒测试。

3．**确认测试**（Validation Testing）

确认测试的任务是检查软件的功能、性能及其他特征与用户的需求一致，它是需求规格说明书（即需求规约）作为依据的测试。确认测试通常采用黑盒测试。

## 6.2测试用例

软件测试的这一项工作是必须的。影响软件测试的因素有很多，比如软件自身的复杂程度、测试的方法和技术的运用等等。在软件测试中有些因素是客观存在的，无法避免，而有些因素则是波动的、不稳定的，例如开发队伍是流动的，在工作中人们也受情绪等影响，等等。有了测试用例，相对来说不管是谁来测试，参照测试用例上的操作一个个操作，都能保障测试的质量，可以把人为因素的影响减少到最小。即使一开始的测试用例考虑不周全，随着测试的不断进行和软件版本的不断更新，也将慢慢完善测试用例。

因此测试用例的设计和编制是软件测试活动中最重要的。本系统功能测试用例表如表6-2所示：

表6-2 测试用例表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试模块及功能名 | 测试内容 | 期望结果 |
| 1 | 系统管理模块 | 能够实现密码的修改和角色管理还有退出系统这三大操作，实现系统中基本信息的维护与管理。包括用户信息的维护、商品信息的维护； | 可以实现 |
| 2 | 商品管理模块 | 当点了子菜单时，可以按照上面的相关内容进行填写，能够进行商品信息和商品的查看、修改、添加的管理；等功能。 | 可以实现 |
| 3 | 会员管理模块 | 当点了子菜单时，可以按照上面的相关内容进行填写，可以分为进行包括个人信息的查看、修改。 | 可以实现 |

## 6.3测试结果

测试结果包括界面及功能测试和代码测试两部分，本系统对每个功能点做了详细的测试，测试结果正常。测试结果表如表6-3所示。

表6-3 系统功能测试结果表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 测试模块 | 功能点 | 质量保证标准 | 问题属性 | 测试结果 |
| F1 | 系统管理模块 | 修改密码 | 正确性 | 故障 | 正常 |
| 角色管理 | 正确性 | 缺陷 | 正常 |
| 退出系统 | 正确性 | 故障 | 正常 |
| F2 | 商品管理模块 | 添加商品 | 正确性 | 故障 | 正常 |
| 查看商品 | 正确性 | 故障 | 正常 |
| 修改商品 | 正确性 | 故障 | 正常 |
| 删除商品 | 正确性 | 故障 | 正常 |
| F3 | 会员管理模块 | 添加会员 | 正确性 | 故障 | 正常 |
| 查看会员信息 | 正确性 | 故障 | 正常 |
| 修改会员信息 | 正确性 | 故障 | 正常 |

# 总结

本篇论文研究的主题是以淘宝为原型的在线商城系统。该系统实现用户的注册和登录等功能，并将用户写入的数据保存在数据库中；商品列表的展示，将数据库中的商品信息读取出来并显示在页面上给用户浏览，主要的模块有商品的展示、商品的明细情况和分类搜索；购物车功能，能管理用户已经选择但还未结账的商品，分别实现了添加商品、删除列表、数量修改、清空购物车和结算功能；我的订单模块，主要是管理用户的购买订单，具体实现功能有订单确认、订单列表展示、删除订单；以及商品管理模块，是后台管理员的核心模块，能够控制整个网站中的商品列表、商品发布、商品修改和商品删除，为管理网上商城起到决定性的作用。

因为对在线商城管理的实际流程不太熟悉，还有就是对系统设计开发的经验缺乏和时间上的仓促等原因，系统开发过程中难免存在一些不足之处。首先，在系统设计方面存在不足，例如在线商城只有一些相关的商品，与实际的商城中的商品相差很远。其次，系统实现也存在不足，例如程序代码不够规范标准；只是去实现了系统要求的功能，但是程序中存在一些异常没有及时的处理，还有一些代码的很冗余。因此系统还有待进一步的改进。

通过这次项目也让我学习到了许多的知识，编程能力和分析解决问题能力也得到了很大的提高，学会了运用所学的计算机理论知识去解决项目开发中遇到的实际问题。我也相信这次项目会为我从事计算机工作积累很好的经验。

# 鸣 谢

在这次的项目设计过中要感谢我的同伴的对我的大力帮助。同伴对我的细心教导和每天的联调工作，没有因为枯燥而烦躁。在总体结构和功能的把握上给予了极大的帮助，并为我在编程、数据库设计等细节工作上给予了耐心的指导，为我对这些有些不熟悉的地方加强了巩固，对于我顺利的完成这次项目设计起到了关键性的作用。

在我遇到困难的时候同伴对我帮助是我的一大收获，这对提高我的编程技术有很大帮助。在此我一并向他们表示感谢，以及在大学生活中给予我关心和帮助的老师和同学们，是他们教会了我在学习上和生活上的知识，不止专业的知识还有做人做事的知识。通过这次的毕业设计我还明白了作为一名计算机类的大学毕业生，我们要会的不仅仅是编写代码，更重要的是要有整体把握系统设计的能力。我会在以后的工作和学习中不断完善自己，为自己以后的道路铺上更加坚固的地砖。

最后再一次感谢所有在这次毕业设计中曾经帮助过我的老师、朋友还有同学们，以及在设计中被我引用或参考的论文和著作，表示由衷的感谢。

# 参考文献

［1］ 简爱华，萧宇嘉，康建华． 2014－2018 年中国餐饮业投资分析及前景预测报告［Ｒ /OL］．［2014－04－15］． http: / /www ．ocn． com． cn /reports /2006083canyin． htm．

[2]  郝玉龙. 迟健男.轻松掌握Struts2.清华大学出版社，2010.

[3]  麦著. Spring攻略.人民邮电出版社，2011.

# 附录一：前端代码

list.jsp:

<%@ page language=*"java"* import=*"java.util.\*"* pageEncoding=*"UTF-8"*%>

<%@ taglib uri=*"/struts-tags"* prefix=*"s"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns=*"http://www.w3.org/1999/xhtml"* xml:lang=*"en"*>

<head>

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

<title>商品列表页</title>

<link rel=*"shortcut icon"* href=*"*${pageContext.request.contextPath}*/*${pageContext.request.contextPath}*/*${pageContext.request.contextPath}*/images/tianshang.ico"* type=*"image/x-icon"* />

<link rel=*"stylesheet"* href=*"*${pageContext.request.contextPath}*/css/base.css"* type=*"text/css"* />

<link rel=*"stylesheet"* href=*"*${pageContext.request.contextPath}*/css/shop\_common.css"* type=*"text/css"* />

<link href=*"*${pageContext.request.contextPath}*/css/common.css"* rel=*"stylesheet"* type=*"text/css"* />

<link rel=*"stylesheet"* href=*"*${pageContext.request.contextPath}*/css/shop\_header.css"* type=*"text/css"* />

<link rel=*"stylesheet"* href=*"*${pageContext.request.contextPath}*/css/shop\_list2.css"* type=*"text/css"* />

<link href=*"*${pageContext.request.contextPath}*/css/product.css"* rel=*"stylesheet"* type=*"text/css"* />

<script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/jquery.js"*></script>

<script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/topNav.js"*></script>

<script type=*"text/javascript"* src=*"*${pageContext.request.contextPath}*/js/shop\_list.js"*></script>

</head>

<body>

<!-- Header -wll-2013/03/24 -->

<%@ include file=*"header2.jsp"*%>

<div class=*"clear"*></div>

<div class=*"clear"*></div>

<!-- Header TopNav End -->

<div class=*"clear"*></div>

<!-- 面包屑 注意首页没有 -->

<div class=*"shop\_hd\_breadcrumb"*>

<strong>当前位置：</strong> <span> <a href=*"*${pageContext.request.contextPath }*/index.action"*>首页</a>&nbsp;›&nbsp; <a

href=*"*${pageContext.request.contextPath }*/index.action"*>商品分类</a>&nbsp;›&nbsp;

<a href=*"*${pageContext.request.contextPath}*/product\_findByCid?cid=*${category.cid}*&page=1"*>${category.cname}</a>

&nbsp;›&nbsp;

<s:if test=*"csid!=null"*>

<a href=*"*${pageContext.request.contextPath}*/product\_findByCsid?csid=*${categoryS.csid}*&page=1"*>${categoryS.csname }</a> </span>

</s:if>

</div>

<div class=*"clear"*></div>

<!-- 面包屑 End -->

<!-- Header End -->

<!-- List Body 2013/03/27 -->

<div class=*"shop\_bd clearfix"*>

<div class=*"shop\_bd\_list\_left clearfix"*>

<!-- 左边商品分类 -->

<div class=*"shop\_bd\_list\_bk clearfix"*>

<div class=*"title"*>商品分类</div>

<div class=*"contents clearfix"*>

<s:iterator var=*"sl"* value=*"#secondlist"*>

<dl class=*"shop\_bd\_list\_type\_links clearfix"*>

<dt>

<a href=*"*${pageContext.request.contextPath}*/product\_findByCid?cid=*<s:property value=*"#sl.cid"*/>*&page=1"*><strong><s:property value=*"#sl.cname"* /></strong></a>

</dt>

<s:iterator var=*"sll"* value=*"#sl.categorySeconds"*>

<dd>

<span><a href=*"*${pageContext.request.contextPath}*/product\_findByCsid?csid=*<s:property value=*"#sll.csid"*/>*&page=1"*><s:property value=*"#sll.csname"* /> </a>

</span>

</dd>

</s:iterator>

</dl>

</s:iterator>

</div>

</div>

<!-- 左边商品分类 End -->

<!-- 热卖推荐商品 -->

<div class=*"shop\_bd\_list\_bk clearfix"*>

<div class=*"title"*>热卖推荐商品</div>

<div class=*"contents clearfix"*>

<ul class=*"clearfix"*>

<s:iterator var=*"sp"* value=*"#specialProduct"*>

<li class=*"clearfix"*>

<div class=*"goods\_name"*>

<a href=*"*${pageContext.request.contextPath}*/product\_findByPid.action?pid=*<s:property value=*"#sp.pid"*/>*"*><s:property value=*"#sp.pname"*/></a>

</div>

<div class=*"goods\_pic"*>

<span class=*"goods\_price"*>¥ <s:property value=*"#sp.shop\_price"*/> </span>

<a href=*"*${pageContext.request.contextPath}*/product\_findByPid.action?pid=*<s:property value=*"#sp.pid"*/>*"*>

<img src=*"*${pageContext.request.contextPath }*/*<s:property value=*"#sp.image"*/>*"* /></a>

</div>

<div class=*"goods\_xiaoliang"*>

<span class=*"goods\_xiaoliang\_link"*><a href=*"*${pageContext.request.contextPath}*/product\_findByPid.action?pid=*<s:property value=*"#sp.pid"*/>*"*>去看看</a> </span> <span

class=*"goods\_xiaoliang\_nums"*>已销售<strong><s:if test=*"#sp.saleCount>0"*><s:property value=*"#sp.saleCount"*/></s:if><s:else>0</s:else></strong>笔</span>

</div>

</li>

</s:iterator>

</ul>

</div>

</div>

<!-- 热卖推荐商品 -->

<div class=*"clear"*></div>

<!-- 浏览过的商品 -->

</div>

<div class=*"shop\_bd\_list\_right clearfix"*>

<!-- 显示菜单 -->

<div class=*"sort-bar"*>

<div class=*"bar-l"*>

<!-- 查看方式S -->

<div class=*"switch"*>

<span class=*"selected"*><a

title=*"以方格显示*

*"*

ecvalue=*"squares"* nc\_type=*"display\_mode"* class=*"pm"*

href=*"javascript:void(0)"*>大图 </a> </span><span style="border-left:*none*;"><a

title=*"以列表显示"* ecvalue=*"list"* nc\_type=*"display\_mode"* class=*"lm"*

href=*"javascript:void(0)"*>列表</a> </span>

</div>

<!-- 查看方式E -->

<!-- 排序方式S -->

<ul class=*"array"*>

<li class=*"selected"*><a title=*"默认排序"*

class=*"nobg"* href=*"javascript:dropParam(['key','order'],'','array');"*>默认</a></li>

<li><a title=*"点击按销量从高到低排序"* onclick="javascript:replaceParam(['key','order'],['sales','desc'],'array');"

href=*"javascript:void(0)"*>销量</a></li>

<li><a title=*"点击按人气从高到低排序"* onclick="javascript:replaceParam(['key','order'],['click','desc'],'array');"

href=*"javascript:void(0)"*>人气</a></li>

<li><a title=*"点击按信用从高到低排序"* onclick="javascript:replaceParam(['key','order'],['credit','desc'],'array');"

href=*"javascript:void(0)"*>信用</a></li>

<li><a title=*"点击按价格从高到低排序"* onclick="javascript:replaceParam(['key','order'],['price','desc'],'array');"

href=*"javascript:void(0)"*>价格</a></li>

</ul>

<!-- 排序方式E -->

<!-- 价格段S -->

<div class=*"prices"*>

<em>¥</em> <input type=*"text"* value=*""* class=*"w30"*> <em>-</em>

<input type=*"text"* value=*""* class=*"w30"*> <input

type=*"submit"* value=*"确认"* id=*"search\_by\_price"*>

</div>

<!-- 价格段E -->

</div>

</div>

<div class=*"clear"*></div>

<!-- 显示菜单 End -->

<!-- 商品列表 -->

<div class=*"shop\_bd\_list\_content clearfix"*>

<ul>

<s:iterator var=*"p"* value=*"pageBean.list"*>

<li>

<dl>

<dt>

<a href=*"*${pageContext.request.contextPath}*/product\_findByPid.action?pid=*<s:property value=*"#p.pid"*/>*"*><img

width=*"30"* height=*"40"*

src=*"*${pageContext.request.contextPath}*/*<s:property value=*"#p.image"*/>*"* />

</a>

</dt>

<dd class=*"title"*>

<a href=*"*${pageContext.request.contextPath}*/product\_findByPid.action?pid=*<s:property value=*"#p.pid"*/>*"*><s:property value=*"#p.pname"* /> </a>

</dd>

<dd class=*"content"*>

<span class=*"goods\_jiage"*> ￥<strong><s:property

value=*"#p.shop\_price"* /> </strong> </span> <span class=*"goods\_chengjiao"*>最近销售<font color=*"red"*><s:property value=*"#p.saleCount"*/></font>笔</span>

</dd>

</dl>

</li>

</s:iterator>

</ul>

</div>

<div class=*"pagination"*>

<span>第 <s:property value=*"pageBean.page"* />/ <s:property

value=*"pageBean.totalPage"* /> 页</span>

<s:if test=*"cid != null"*>

<s:if test=*"pageBean.page != 1"*>

<a href=*"*${ pageContext.request.contextPath }*/product\_findByCid.action?cid=*<s:property value=*"cid"*/>*&page=1"*

class=*"firstPage"*>&nbsp;</a>

<a href=*"*${ pageContext.request.contextPath }*/product\_findByCid.action?cid=*<s:property value=*"cid"*/>*&page=*<s:property value=*"pageBean.page-1"*/>*"*

class=*"previousPage"*>&nbsp;</a>

</s:if>

<s:iterator var=*"i"* begin=*"1"* end=*"pageBean.totalPage"*>

<s:if test=*"pageBean.page != #i"*>

<a href=*"*${ pageContext.request.contextPath }*/product\_findByCid.action?cid=*<s:property value=*"cid"*/>*&page=*<s:property value=*"#i"*/>*"*><s:property

value=*"#i"* />

</a>

</s:if>

<s:else>

<span class=*"currentPage"*><s:property value=*"#i"* />

</span>

</s:else>

</s:iterator>

<s:if test=*"pageBean.page != pageBean.totalPage"*>

<a class=*"nextPage"* href=*"*${ pageContext.request.contextPath }*/product\_findByCid.action?cid=*<s:property value=*"cid"*/>*&page=*<s:property value=*"pageBean.page+1"*/>*"*>&nbsp;</a>

<a class=*"lastPage"* href=*"*${ pageContext.request.contextPath }*/product\_findByCid.action?cid=*<s:property value=*"cid"*/>*&page=*<s:property value=*"pageBean.totalPage"*/>*"*>&nbsp;</a>

</s:if>

</s:if>

<s:if test=*"csid != null"*>

<s:if test=*"pageBean.page != 1"*>

<a href=*"*${ pageContext.request.contextPath }*/product\_findByCsid.action?csid=*<s:property value=*"csid"*/>*&page=1"*

class=*"firstPage"*>&nbsp;</a>

<a href=*"*${ pageContext.request.contextPath }*/product\_findByCsid.action?csid=*<s:property value=*"csid"*/>*&page=*<s:property value=*"pageBean.page-1"*/>*"*

class=*"previousPage"*>&nbsp;</a>

</s:if>

<s:iterator var=*"i"* begin=*"1"* end=*"pageBean.totalPage"*>

<s:if test=*"pageBean.page != #i"*>

<a href=*"*${ pageContext.request.contextPath }*/product\_findByCsid.action?csid=*<s:property value=*"csid"*/>*&page=*<s:property value=*"#i"*/>*"*><s:property

value=*"#i"* />

</a>

</s:if>

<s:else>

<span class=*"currentPage"*><s:property value=*"#i"* />

</span>

</s:else>

</s:iterator>

<s:if test=*"pageBean.page != pageBean.totalPage"*>

<a class=*"nextPage"* href=*"*${ pageContext.request.contextPath }*/product\_findByCsid.action?csid=*<s:property value=*"csid"*/>*&page=*<s:property value=*"pageBean.page+1"*/>*"*>&nbsp;</a>

<a class=*"lastPage"* href=*"*${ pageContext.request.contextPath }*/product\_findByCsid.action?csid=*<s:property value=*"csid"*/>*&page=*<s:property value=*"pageBean.totalPage"*/>*"*>&nbsp;</a>

</s:if>

</s:if>

</div>

<!-- 商品列表 End -->

<!-- List Body End -->

<!-- Footer - wll - 2013/3/24 -->

<div class=*"clear"*></div>

<div class=*"shop\_footer"*>

<div class=*"shop\_footer\_link"*>

<p>

<a href=*""*>首页</a>| <a href=*""*>招聘英才</a>| <a href=*""*>广告合作</a>| <a

href=*""*>关于tianShang</a>| <a href=*""*>关于我们</a>

</p>

</div>

<div class=*"shop\_footer\_copy"*>

<p>Copyright 2004-2013 tianShang Inc.,All rights reserved.</p>

</div>

</div>

<!-- Footer End -->

</body>

</html>

# 附录二：后台代码

package com.cdd.shopping.index.action;

import java.util.List;

import com.cdd.shopping.category.domain.Category;

import com.cdd.shopping.category.service.CategoryService;

import com.cdd.shopping.categorysecond.domain.CategorySecond;

import com.cdd.shopping.product.domain.Product;

import com.cdd.shopping.product.service.ProductService;

import com.opensymphony.xwork2.ActionContext;

import com.opensymphony.xwork2.ActionSupport;

/\*\*

\* 主页Index

\* @author cdd

\*

\*/

public class IndexAction extends ActionSupport {

//注入userService

private CategoryService categoryService;

//注入productService

private ProductService productService;

public void setProductService(ProductService productService) {

this.productService = productService;

}

public void setCategoryService(CategoryService categoryService) {

this.categoryService = categoryService;

}

@Override

public String execute() throws Exception {

//查询一级分类

List<Category> cList=categoryService.findAll();

System.out.println(cList);

ActionContext.getContext().getSession().put("cList", cList);

//查询二级分类

//查询所有热卖商品

List<Product> hList=productService.findHotProduct();

ActionContext.getContext().getValueStack().set("hList", hList);

//查询最新商品

List<Product> nList=productService.findNewProduct();

ActionContext.getContext().getValueStack().set("nList", nList);

//查询所有推荐商品

List<Product> sList=productService.findSpecialProduct();

ActionContext.getContext().getValueStack().set("sList", sList);

//查询排行商品

List<Product> topList=productService.findTopProduct();

ActionContext.getContext().getValueStack().set("topList", topList);

//查找男装女装

Category category=categoryService.findByCategoryName();

String cid=category.getCid();

ActionContext.getContext().getValueStack().set("category", category);

List<Product> colthList=productService.findByCid(cid);

ActionContext.getContext().getValueStack().set("colthList", colthList);

//查询对应的二级服装分类

List<CategorySecond> caList=categoryService.findByCsid(cid);

ActionContext.getContext().getValueStack().set("caList", caList);

System.out.println(caList);

// System.out.println(topList);

return "index";

}

}