

# 文本复制检测报告单(全文标明引文)

№:BC202105182243561033774269

检测时间:2021-05-18 22:43:56

检测文献: 基于spring boot框架的网上商场购物平台的设计与实现

作者: 蒲俊鹏

检测范围: 中国学术期刊网络出版总库

中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库

中国重要会议论文全文数据库

中国重要报纸全文数据库

中国专利全文数据库

图书资源

优先出版文献库

大学生论文联合比对库

互联网资源(包含贴吧等论坛资源)

英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)

港澳台学术文献库

互联网文档资源

源代码库

CNKI大成编客-原创作品库

时间范围: 1900-01-01至2021-05-18

## 检测结果

去除本人文献复制比: 5.5%

去除引用文献复制比: 5.5%

总文字复制比: 5.5%

单篇最大文字复制比: 1% (571224\_周伟铭\_基于SSM框架的学做菜网站的设计与实现)

重复字数: [1309]

总段落数: [2]

总字数: [23637]

疑似段落数: [2]

单篇最大重复字数: [225]

前部重合字数: [557]

疑似段落最大重合字数: [721]

后部重合字数: [752]

疑似段落最小重合字数: [588]



指标: ☐ 疑似剽窃观点 ☒ 疑似剽窃文字表述 ☐ 一稿多投 ☐ 疑似整体剽窃 ☐ 过度引用 ☐ 重复发表

表格: 0 公式: 没有公式 疑似文字的图片: 0 脚注与尾注: 0

5% (588) 5% (588) 基于spring boot框架的网上商场购物平台的设计与实现\_第1部分 (总11710字)

6% (721) 6% (721) 基于spring boot框架的网上商场购物平台的设计与实现\_第2部分 (总11927字)



(注释: 无问题部分 文字复制比部分 引用部分)

## 1. 基于spring boot框架的网上商场购物平台的设计与实现\_第1部分

总字数: 11710

### 相似文献列表

去除本人文献复制比: 5% (588)

文字复制比: 5% (588)

疑似剽窃观点: (0)

1	571224_周伟铭_基于SSM框架的学做菜网站的设计与实现 周伟铭 - 《大学生论文联合比对库》 - 2020-05-20	1.9% (225) 是否引证: 否
2	周伟铭_软件开发_5620163043_基于SMM框架的学做菜网站的设计与实现 软件开发 - 《大学生论文联合比对库》 - 2020-05-25	1.3% (149) 是否引证: 否
3	周伟铭_软件开发_5620163043_基于SMM框架的学做菜网站的设计与实现 软件开发 - 《大学生论文联合比对库》 - 2020-05-27	1.3% (149) 是否引证: 否
4	施昂-2015124089-《几行》APP的设计与实现 施昂 - 《大学生论文联合比对库》 - 2019-05-20	1.1% (134) 是否引证: 否
5	施昂-2015124089-《几行》安卓APP的设计与实现 施昂 - 《大学生论文联合比对库》 - 2019-06-10	1.1% (134) 是否引证: 否

6	施昂-2015124089-《几行》安卓APP的设计与实现 施昂 - 《大学生论文联合比对库》 - 2019-06-14	1.1% (134) 是否引证: 否
7	移动数据生成与查询测试平台的设计与实现 刘海燕(导师: 王波涛) - 《东北大学硕士论文》 - 2012-06-01	0.8% (95) 是否引证: 否
8	内容管理系统的设计与实现 梁文幸 - 《大学生论文联合比对库》 - 2018-05-23	0.8% (95) 是否引证: 否
9	基于BS模式的商务酒店管理系统的设计与实现 孙荣(导师: 孙美君;付劝业) - 《天津大学硕士论文》 - 2014-12-01	0.6% (70) 是否引证: 否
10	权限管理的分析与设计 周凯强 - 《大学生论文联合比对库》 - 2018-05-12	0.6% (69) 是否引证: 否
11	基于JSF+Spring+Hibernate的人事管理系统 钱峰; - 《电脑知识与技术》 - 2012-02-25	0.6% (65) 是否引证: 否
12	失物招领系统的建立 董伟东; - 《计算机光盘软件与应用》 - 2014-01-01	0.6% (65) 是否引证: 否
13	基于Android智能手表计步器的研究与设计 于广艳; - 《科学技术创新》 - 2017-07-05	0.6% (65) 是否引证: 否
14	包头市财经学校网络题库与考试系统的设计与实现 任树理(导师: 黄迪明;付广林) - 《电子科技大学硕士论文》 - 2011-05-01	0.6% (65) 是否引证: 否
15	基于SSH框架的教学资源管理系统的设计与实现 李晖(导师: 肖创柏) - 《北京工业大学硕士论文》 - 2012-12-01	0.6% (65) 是否引证: 否
16	特色农产品电子交易平台设计实现 宋光正(导师: 喻国平) - 《南昌大学硕士论文》 - 2013-12-02	0.6% (65) 是否引证: 否
17	基于OLAP技术的供电公司经营风险系统的研究与开发 刘宁(导师: 申晓留) - 《华北电力大学(北京) 硕士论文》 - 2008-02-01	0.6% (65) 是否引证: 否
18	基于重庆电信SF系统的小灵通短信预警系统设计 肖鹏(导师: 孙跃;雷勇) - 《重庆大学硕士论文》 - 2007-10-01	0.6% (65) 是否引证: 否
19	网格计算环境下的虚拟企业管理相关问题研究 樊宁(导师: 张润彤) - 《北京交通大学硕士论文》 - 2006-12-01	0.6% (65) 是否引证: 否
20	四川文理学院学生管理系统的设计与实现 贺建英(导师: 彭寿全;周兰) - 《电子科技大学硕士论文》 - 2007-04-01	0.6% (65) 是否引证: 否
21	智能交通诱导信息服务平台设计与验证 王珏(导师: 刘杰) - 《北京邮电大学硕士论文》 - 2015-03-11	0.6% (65) 是否引证: 否
22	基于智能手机的规模定制版本控制与实现 张赛(导师: 杨述斌;张琪) - 《武汉工程大学硕士论文》 - 2015-04-04	0.6% (65) 是否引证: 否
23	基于统一目录系统的单点登录系统的实现 张晓琴(导师: 任长明;张冠忠) - 《天津大学硕士论文》 - 2008-08-01	0.5% (63) 是否引证: 否
24	便携式运动检测仪的设计与实现 韩熙熙(导师: 周又玲) - 《海南大学硕士论文》 - 2016-05-01	0.5% (63) 是否引证: 否
25	基于J2EE和HTTPS通信技术的专用汽车集成信息系统研究 冯志强(导师: 杨宗霄) - 《河南科技大学硕士论文》 - 2008-06-01	0.5% (59) 是否引证: 否
26	常用加工工艺资源环境属性数据库系统研究 陈超(导师: 张华) - 《武汉科技大学硕士论文》 - 2008-05-02	0.5% (59) 是否引证: 否
27	基于PHP的电子购物网站的设计与实现 戴伟超 - 《大学生论文联合比对库》 - 2014-04-19	0.5% (59) 是否引证: 否
28	基于BP神经网络的制造过程资源环境属性诊断的方法研究 鄢威(导师: 张华) - 《武汉科技大学硕士论文》 - 2008-12-20	0.5% (58) 是否引证: 否
29	辽宁工业大学教务管理系统的设计与实现 蒋志敏(导师: 甘露;陈晨) - 《电子科技大学硕士论文》 - 2014-03-01	0.3% (31) 是否引证: 否

#### 原文内容

分类号: TP311.1 U D C: D10621-408-(2021)4300-0 密级: 公开编号: 2017121106  
 成都信息工程大学  
 学位论文  
 基于springboot框架的网上商场购物平台的设计与实现

论文作者姓名:	蒲俊鹏
申请学位专业:	网络工程
申请学位类别:	工学学士
指导教师姓名(职称):	李舸(讲师)
论文提交日期:	2021年06月09日

论文作者姓名: 蒲俊鹏

申请学位专业: 网络工程

申请学位类别: 工学学士

指导教师姓名(职称): 李舸(讲师)

论文提交日期: 2021年06月09日

基于springboot框架的网上商场购物平台的设计与实现

**摘要:** 随着社会经济和科学技术的发展,人们的消费能力不断提高,同时人群大众对各类商品购买的需求也不断增加。以往的实体店购物模式不能满足人们的购物需求,消费者对购物流程提出了更高的需要,同时商家也想利用方便规范的管理工具来管理商店。本网上商城购物系统是基于springboot框架下的网页开发平台,其运行流程主要是:前端根据地址映射向后台发送异步请求,后台则根据需求处理参数,完成请求目的并返回数据,最后前端根据返回信息进行渲染页面或者响应用户。其中本系统利用本地mysql数据库和七牛云服务器存储数据和图片资源,有方便用户登录的单点登录功能和减少对数据频繁交互的redis缓存功能。这些应用满足广大消费者的购物需求,同时提升了本系统的运行效率,降低了系统运行的维护成本。

**关键词:** springboot; 分离; 效率; 需求

Design and Implementation of Online Shopping Platform Based on Springboot Framework

**Abstract:** With the development of social economy and science and technology, people's spending power continues to improve, and at the same time, the public's demand for all kinds of commodities is also increasing. In the past, the physical store shopping model could not meet people's shopping needs. Consumers have put forward higher demands on the shopping process. At the same time, merchants also want to use convenient and standardized management tools to manage the store. This online shopping system is a web development platform based on the springboot framework. The system is developed in a distributed mode, and the management module and the portal module are developed separately. The operating process of this system is mainly: the front end sends an asynchronous request to the background according to the address mapping, the background processes the parameters according to the requirements, completes the request purpose and returns data, and finally the front end renders the page or responds to the user according to the returned information. Among them, this system uses the local mysql database and Qiniu cloud server to store data and image resources, has a single sign-on function that facilitates user login, and a redis cache function that reduces frequent interactions with data. These applications meet the shopping needs of consumers and the management needs of merchants, while improving the operating efficiency of the system and reducing the maintenance costs of system operation.

**Key words:** springboot; separation; efficiency; demand

## 目录

论文总页数: 38页

1 引言	1
1.1 课题背景	1
1.2 国内外研究现状	1
1.3 本课题研究的意义	1
1.4 本课题研究的方法	2
2 开发环境及相关技术工具介绍	3
2.1 开发环境	3
2.2 相关技术	3
2.2.1 SpringBoot	3
2.2.2 Mysql	3
2.3 开发工具	3
2.3.1 JDK	3
2.3.2 Maven	3
2.3.3 Redis	3
2.3.5 Mybatis	3
3 系统需求分析和设计	4
3.1 系统可行性分析	4
3.1.1 经济可行性分析	4
3.1.2 技术可行性分析	4
3.1.3 操作可行性分析	4
3.2 功能需求	4

3.3 功能模块图.....	5
3.4 系统流程图.....	5
4 系统设计.....	8
4.1 系统模块设计.....	8
4.1.1 公共类模块.....	8
4.1.2 商品管理模块.....	8
4.1.3 门户模块.....	8
4.1.4 生成文件模块.....	8
4.2 数据库设计.....	8
4.2.1 数据库实体图.....	8
4.2.2 数据表设计.....	10
5 系统的实现.....	13
5.1 商品管理模块的实现.....	13
5.1.1 管理员登录子模块.....	13
5.1.2 管理商品分类子模块.....	15
5.1.3 管理商品子模块.....	21
5.2 门面模块的实现.....	23
5.2.1 登录子模块.....	23
5.2.2 搜索子模块.....	25
5.2.3 添加购物车子模块.....	26
6 系统展示.....	28
6.1 后台管理模块展示.....	28
6.1.1 登录界面展示.....	28
6.1.2 商品分类列表展示.....	28
6.1.3 商品列表展示.....	31
6.2 门户模块展示.....	32
6.2.1 登录界面展示.....	32
6.2.2 首页展示.....	33
6.2.3 搜索展示.....	33
6.2.4 添加购物车展示.....	34
结论.....	35
参考文献.....	36
致谢.....	37
声明.....	38

## 1 引言

### 1.1 课题背景

当代网络技术高速发展，网络覆盖越来越宽阔，大众人群消费水平普遍提升，快递行业发展蓬勃，人们更倾向足不出户地在网上购买自己所需要各种各样的商品。其中，网上购物服务流程完善，消费者能方便快捷地达到购物目的，同时网上购物的法律规章十分健全，这样购物者的消费能得到完善的法律保护。另外，网上售货模式也为大量人群提供了自主创业的机会和就业岗位，于是一个完整全面的购物平台由消费者和卖家需求油然而生。

### 1.2 国内外研究现状

在国内外有很多优秀的网上购物商城。

亚马逊公司（Amazon）是美国的一家覆盖全球的电子商务公司。在最开始，亚马逊仅仅是一个网上书店，但是很快它跟随时代潮流开始全方面多领域发展，其包含业务覆盖：食品、书籍、DVD销售、视频、软件、电视游戏、化妆品、服装、家具、玩具、保健品、珠宝、电子产品等。

沃尔玛百货公司（Walmart）是一家全世界连锁店和仓储式商店的跨国销售公司。该公司大规模涉足零售业，同时还是世界上雇佣员工最多的公司。该公司运营着了一款名为沃尔玛商城的Android平台应用。该应用的客户端能够为用户提供数码、食品、日用品、服饰、家具、婴幼儿用品、图书杂志等相关产品信息，用户可以参考商品信息挑选自己所需求的商品，再到沃尔玛门店自购商品。

淘宝网是亚太地区深受大众人群喜爱的网络零售，该网站由阿里巴巴公司成立于2003年5月。淘宝网是中国深受人群欢迎的网购零售平台，现如今它拥有近5亿之多的注册用户，而且每天至少6000万的固定访客。

### 1.3 本课题研究的意义

很多消费者和商店管理者都会购物平台反馈用户体验，但这些知名网站在多个版本更新中都未完善。要想在众多优秀的网上购物商场中脱颖而出，本系统必须弥补这些网站现未注意到的瑕疵细节。

本系统不仅能满足消费者的基本购物需要，还能在全方面考虑消费者感受、让消费者得到更好的购物体验。同时开发者也能根据网上商店管理者提出的管理需求，为管理者提供具有完整、可靠的管理功能的网上购物管理平台。除此之外，图片开发者而言，本系统开发工作量小，但所含功能完善，足以满足小众人群使用。

### 1.4 本课题研究的方法

首先去各大网上购物网站调研，参考网站门户界面框架、功能，体验多次完整的购物流程，模拟多种购物场景，构想设计本系统的独特之处。然后向网购人群发起问卷调查和网上查阅资料[1][2]，根据问卷报告和参考论文进行需求分析，搭建数据库，引入测试数据。下一步选择开发工具和搭建开发环境，开始本系统的开发。在开发过程中，主要分布式开发门户模块

(ShopOnline-portal)和商店管理模块(ShopOnline-management)两个主要模块，同时搭建公用代码模块(ShopOnline-common)、利用mybatis逆向工程根据数据库信息自动生成实体类和mapper文件模块(ShopOnline-generator)。随后进行完整的开发，在这一过程中，不断测试代码的正确性，不断完善代码的健壮性。最后完成系统开发，完成[论文编撰说明本系统。](#)

2 开发环境及相关技术工具介绍

2.1 开发环境

[IDEA](#) 总称是 IntelliJ IDEA，其是多高级编程语言开发的集成环境。该环境在智能代码助手、JUnit、JavaEE支持、代码自动提示、各类版本工具(git、svn等)、重构、CVS整合、代码分析、创新的GUI设计等这些方面的功能可以说是十分优异的。

2.2 相关技术

2.2.1 SpringBoot

[Spring Boot](#)框架是由Pivotal团队开发提供的一种全新框架，该框架的设计目的是用来简化新Spring应用的初始搭建以及开发过程。

2.2.2 Mysql

[MySQL](#) 是一款高效的、安全、跨平台，并与 PHP、Java 等主流编程语言紧密结合的数据库系统。

2.3 开发工具

2.3.1 JDK

[JDK](#)是一个程序开发环境。它由一个处于操作系统层之上的运行环境还有开发者编译，调试和运行用Java语言写的applet和应用程序所需的工具组成。

2.3.2 Maven

[Maven](#) 是来自Apache下的一个开源项目，它是一个创新的项目管理工具，它用于对Java项目进行项目构建、依赖管理及项目信息管理。

2.3.3 Redis

[Redis](#) 是开源的，其遵守 BSD 协议，是一个高性能优化的 key-value 数据库。

2.3.5 Mybatis

[MyBatis](#) 是一款优秀的持久层框架，它支持自定义 SQL、存储过程以及高级映射。[MyBatis](#) 免除了几乎所有的 JDBC 代码以及设置参数和获取结果集的工作。

3 系统需求分析和设计

3.1 系统可行性分析

3.1.1 经济可行性分析

本文所述的网上购物商场仅需要以个人计算机作为运行平台，所需开发工具和存储服务器都可以使用免费版本，无高成本要求，而且利用浏览器便能访问测试本系统搭建的网站。因为网站的较低工作量，其运行和维护成本会变得很低。同时，商家的入住和广告商的投资会给本系统带来一些收益。综上所述，本系统完全符合经济可行性要求。

3.1.2 技术可行性分析

本系统选用IntelliJ IDEA作为软件开发环境，该平台功能十分强大，整合多种开发工具，满足各种编程需求，完全足够本系统的开发。对于数据库方面，本系统选用mysql数据库来保存各种数据。在编程语言方面，本系统采用java语言开发，该编程语言是当前热门、十分成熟的高级编程语言，能保证编写的系统能稳定开发和运行。本系统运行所需的硬件要求极低，现市场上绝大主流配置的计算机都能轻易满足运行该系统的要求。综上所述，本系统完全满足技术可行性方面。

3.1.3 操作可行性分析

本系统同现各大网上购物商场有相似门户界面，且根据多数用户偏好构建页面框架，友好便利。对用商店管理界面设计，本系统引导语言简单易懂，商店管理员能立刻熟悉各个功能的使用方法，本系统能全方面满足管理员对商品等方面进行便利地管理。

3.2 功能需求

本系统网上购物商城主要功能有商店管理员对商品分类和商品进行增删改查，有消费者登录和商店管理者登录功能。消费者能浏览商城门户，根据商品分类或者搜索关键字查询自己所需的商品，并将选的商品加入购物车[3] [4]。门户首页的采用静态资源渲染，以便随时更新热门商品显示。而搜索界面的商品展示，则根据所搜索商品存储在数据库中的url地址查询对应在七牛云服务器存储图片进行动态渲染，以减少本地计算机存储资源量，增强本系统运行效能。同时商品分类信息和商品信息在第一次查询过后，会存储在redis缓存中，避免频繁访问同一资源，降低了对数据库的请求频率。

3.3 功能模块图

本系统主要有后台管理模块和门户模块构成，这些模块共同构建了整个系统功能，其具体功能由图3.1所示：

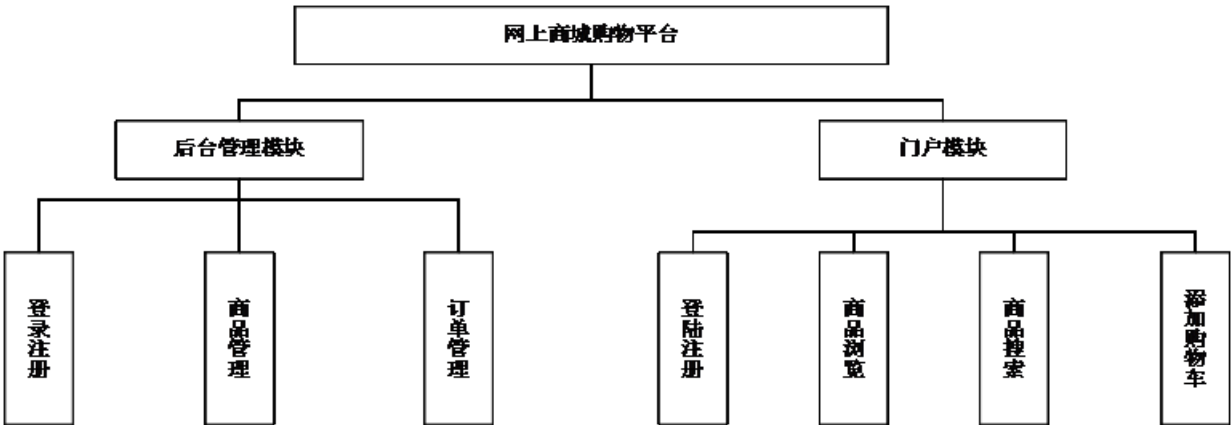


图3.1 功能模块



3.4 系统流程图

消费者完成购物流程如图3.2所示：

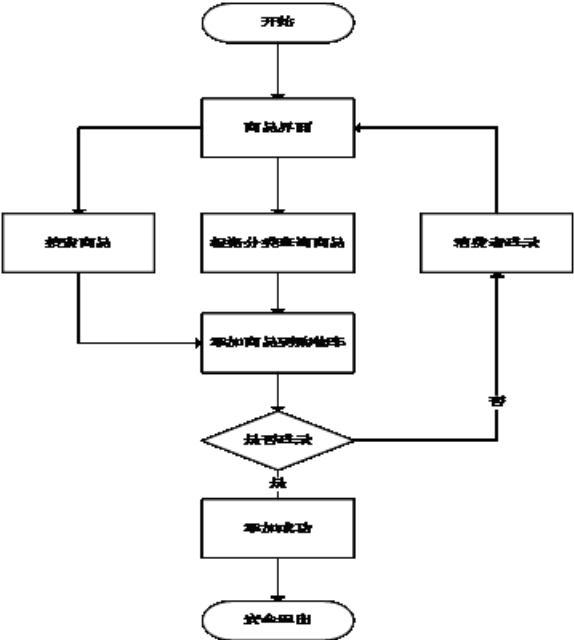
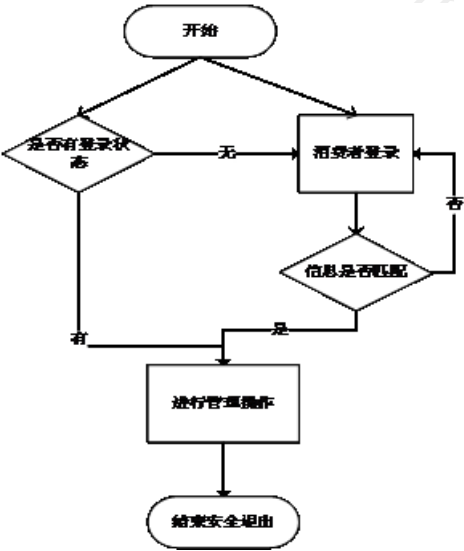


图3.2 消费者流程

商店管理者管理商品流程如图3.3所示：  
图



3.3管理流程

4 系统设计

4.1系统模块设计

4.1.1 公共类模块

公共类模块主要存放公共依赖文件、公共返回状态类(其中包括分页信息)、工具类。门户模块和商品管理模块依赖继承于公共模块，该两主模块仅需声明依赖名称，减少了依赖文件的冗余。同时两模块在调用方法后，都需要返回状态信息以标志方法是否执行成功。Cookie工具类用于存储用户登录信息，Json工具类用于对象与字符串之间的转化，Md5工具类用于加密用户密码保障用户信息安全性，UUID工具类用于生成用户登录的唯一标志的凭据。

4.1.2 商品管理模块

**商品管理模块是商家所有者对自家商品进行管理的模块。**商家在该模块可以安全单点登录，同时商家处于未登录状态无法直接通过url地址访问管理界面。商家能够对商品分类信息和商品信息进行增删改查操作。也能根据商品分类、商品品牌、商品名关键字搜索指定商品信息，同时能调节每页显示数据条数，分页查询商品信息。最后管理者能够安全退出以保证账号的安全性。

4.1.3 门户模块

门户模块是满足消费者进行自主购物的模块。消费者在未登录状态下，可以任意浏览商品信息，搜索指定商品。若消费者要查看自我购物车信息或者将商品添加到购物车中，需要登录个人信息才能执行成功。消费者的登录也会处于安全的单点登录状态，最后消费者也能安全退出以保证账号的安全性。

4.1.4生成文件模块

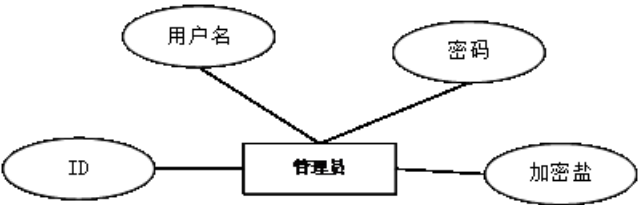
该模块通过配置mybatis-generator.xml文件来根据数据库表名和表中主键名，逆向生成对应的实体类和mapper文件[5]。

4.2 数据库设计

4.2.1 数据库实体图

商品分类(ID, 分类名, 父分类id, 排序, 是否热门, 是否显示, 分组, 图片地址, 分佣比例)

图4.1 商品分类  
管理员(ID, 用户名, 密码, 加密盐)  
图



4.2 管理员  
消费者(ID, 用户名, 密码, 加密盐)

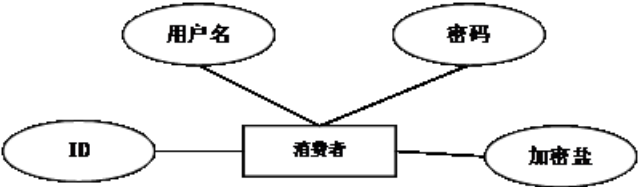


图4.3 消费者  
商品(ID, 商品名称, 商品分类id, 品牌id, 库存数量, 详细描述, 商品简介, 是否包邮)

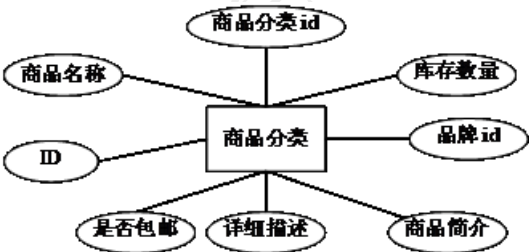


图4.4 商品  
品牌(ID, 品牌名, 图片地址, 排序, 是否热门)

图4.5 品牌

4.2.2 数据表设计

管理员表主要存储管理员注册信息。本系统管理员表t\_admin如表4.1所示：

表4.1 管理员表

列名	数据类型	允许NULL值	说明
admin_id	smallint	否	主键
user_name	varchar(60)	否	用户名
password	varchar(32)	否	密码
ec_salt	varchar(10)	是	密钥

列名数据类型允许NULL值说明

admin\_id smallint 否主键

user\_name varchar(60) 否用户名

password varchar(32) 否密码

ec\_salt varchar(10) 是密钥

商品分类表主要存储商品分类信息。本系统商品分类表t\_goods\_category如表4.2所示：

表4.2 商品分类表

列名	数据类型	允许NULL值	说明
id	smallint	否	主键
name	varchar(90)	否	分类名称
parent_id	smallint	否	父分类id
level	tinyint	否	分类等级
sort_order	tinyint	否	排序
is_show	tinyint	否	是否显示
image	varchar(512)	是	图片地址
is_hot	tinyint	是	是否热门
cat_group	tinyint	是	分类分组
commission_rate	tinyint	是	分佣比例

列名数据类型允许NULL值说明

id smallint 否主键

name varchar(90) 否分类名称

parent\_id smallint 否父分类id

level tinyint 否分类等级

sort\_order tinyint 否排序

is\_show tinyint 否是否显示

image varchar(512) 是图片地址  
is\_hot tinyint 是是否热门  
cat\_group tinyint 是分类分组  
commission\_rate tinyint 是分佣比例  
品牌表主要存储品牌信息。本系统品牌表t\_brand如表4.3所示：  
表4.3 品牌表

列名	数据类型	允许NULL值	说明
id	smallint	否	主键
name	varchar(60)	否	品牌名称
logo	varchar(80)	否	图片地址
sort	tinyint	否	排序
is_hot	tinyint	是	是否热门

列名数据类型允许NULL值说明

id smallint 否主键  
name varchar(60) 否品牌名称  
logo varchar(80) 否图片地址  
sort tinyint 否排序  
is\_hot tinyint 是是否热门

商品表主要存储商品信息。本系统商品表t\_goods如表4.4所示：  
表4.4 商品表

列名	数据类型	允许NULL值	说明
goods_id	mediumint	否	主键
goods_name	varchar(120)	否	商品名称
cat_id	int	否	分类id
store_count	smallint	否	库存数量
market_price	decimal	否	市场价
goods_remark	varchar(255)	否	商品简介
goods_content	text	是	详细描述
is_free_shipping	tinyint	否	是否包邮
brand_id	smallint	否	品牌id

列名数据类型允许NULL值说明

goods\_id mediumint 否主键  
goods\_name varchar(120) 否商品名称  
cat\_id int 否分类id  
store\_count smallint 否库存数量  
market\_price decimal 否市场价  
goods\_remark varchar(255) 否商品简介  
goods\_content text 是详细描述  
is\_free\_shipping tinyint 否是否包邮  
brand\_id smallint 否品牌id

消费者表主要存储消费者信息。本系统消费者表t\_users如表4.5所示：  
表4.5 消费者表

列名	数据类型	允许NULL值	说明
id	user_id	否	主键
email	varchar(60)	否	用户名
password	varchar(32)	否	密码

列名数据类型允许NULL值说明

id user\_id 否主键  
email varchar(60) 否用户名  
password varchar(32) 否密码

## 5 系统的实现

### 5.1 商品管理模块的实现

#### 5.1.1 管理员登录子模块

通过js、css和image静态资源文件对项目resource包下index.ftl渲染出一个管理员登录界面，其界面主要有登录账号、密码和验证码输入框。然后系统将用户输入的账号、密码和验证码发送至UserController调用方法验证信息的正确性。

```
//ticketString ticket = ssoService.login(admin);// cookieif(!StringUtil.isEmpty(ticket)){
System.out.println("username:"+admin.getUserName()); //save in session
request.getSession().setAttribute("user",admin); return
cookieService.setCookie(request,response,ticket)?BaseResult.success():BaseResult.error();}System.out.println("a
dmin is null ");return BaseResult.error();
```

其中ssoService服务的login()方法实现验证信息正确性和实现单点登录。ssoService首先验证账号是否存在。若存在，则



用该账号在数据库中存放的md5加密盐加密登录密码，再对比密码是否匹配。若全部匹配，则用UUIDUtil工具类生成一个随机的ticket并返回作为该用户的登录唯一票据，同时将该存入redis和cookie中以备单点登录使用[6]。

```
        //login()
        if(!admin1.getPassword().equals(Md5Util.getMd5WithSalt(admin.getPassword(),admin1.getEcSalt()))){
            System.out.println("pwd is invalid"); return null;}ValueOperations<String,String> valueOperations =
            redisTemplate.opsForValue();String ticket =
            UUIDUtil.getUUID();System.out.println("ticket:"+ticket);valueOperations.set(userTicket+":"+ticket,JsonUtil.objec
            ct2JsonStr(admin1),30, TimeUnit.MINUTES);
```

当用户处于未登录状态，直接通过url地址访问管理界面，会被设置的拦截器拦截至登录界面。该功能在interceptor包下的ManagerLoginInterceptor.java类实现，同时在config包下的MvcConfig.java类进行添加配置该拦截器，同时分离出静态资源设置允许该类通过。当用户已经登录时，通过cookie中存储信息得到票据，并通过validate()方法验证票据的正确性取得用户对象，再将对象存入session以备输出个人信息使用[7] [8] [9]。

```
        //config interceptor:
        String ticket = CookieUtil.getCookieValue(request,
        "userTicket");System.out.println("ticket:"+ticket);if(!StringUtil.isEmpty(ticket)){ Admin admin =
        ssoService.validate(ticket); request.getSession().setAttribute("user",admin); ValueOperations<String, String>
        valueOperations = redisTemplate.opsForValue(); valueOperations.set(userTicket+":"+ticket,
        JsonUtil.object2JsonStr(admin),30, TimeUnit.MINUTES); return true;}System.out.println("turn to
        login");response.sendRedirect(request.getContextPath()+"/login");
```

当用户访问其他网页或者关闭浏览器后，在缓存生效时间30分钟时间内，再次访问管理主页，都无需登录验证，同时缓存时间重新刷新。

若用户点击主页的安全退出，则调用logout()方法删除redis中的缓存、调用deleteCookie()方法删除cookie中缓存的票据信息、调用remove()方法删除session中的用户对象，则用户再次访问管理界面需要重新验证登录信息。

```
        //logout safely:

        ssoService.logout(ticket);request.getSession().removeAttribute("user");cookieService.deleteCookie(request, respo
        nse);
```

这就是本系统实现的完整的管理员单点登录功能，极大的减除了用户多次访问管理界面的需要重新验证信息的麻烦问题，同时安全退出则保障了用户信息安全。

### 5.1.2 管理商品分类子模块

点击进入商品模块的管理商品分类界面，需要渲染商品分类页面。首先通过地址映射调用GoodsCategoryController中的categoryList()方法获得所有的分类信息。其中调用goodsCategoryService中的selectCategoryListView()方法先查询redis缓存中是否已存在缓存的分类信息，若有则直接返回分类信息。若无，则先查询所有的顶级分类，用list接收查询返回对象，然后将顶级分类对象循环转化为vo对象，其中vo对象增加的List<GoodsCategoryVo> children字段以放置子分类信息。再根据父分类id循环查询各自的子分类(二级分类)信息并存放至父分类的children字段中。之后依次类推查询三级分类并存放。最后将顶级分类信息放入redis缓存中，返回list。

```
        //looking for buffer:
        String gcvListJson = valueOperations.get(goodsListKey);if(!StringUtil.isEmpty(gcvListJson)){ return
        JsonUtil.jsonToList(gcvListJson, GoodsCategoryVo.class);}
        //save in redis
        valueOperations.set(goodsListKey,JsonUtil.object2JsonStr(gcvList));
        GoodsCategoryController
        //save in model
        model.addAttribute("gcvList",goodsCategoryService.selectCategoryListView());
        前端freemarker模板利用<#list></list>标签得到存放在model中list的数据信息，通过三层标签循环输出数据，渲染分类
        列表信息。
```

```
        //print the information
        <#list gcvList as gcv1>
        <td>${gcv1.id}</td>
        <#list gcv1.children as gcv2>
        <td>${gcv2.id}</td>
        <#list gcv2.children as gcv3>
        <td>${gcv3.id}</td>
        </#list> </#list></#list>
```

利用图片渲染分类信息是否推荐和是否显示的方法是：给每个对应标签加入唯一的id，再添加js方法在加载界面资源时执行循环渲染对应图片。

```
        //set id
        <input type="hidden" id = "isHot${gcv1.id}" value="${gcv1.isHot}">
        <input type="hidden" id = "isShow${gcv1.id}" value="${gcv1.isShow}">
        //set images
```

```
if ("1" == $("#isHot"+i).val()) { //$("#imgHot"+i).attr("src","${ctx}/static/images/yes.png");
document.getElementById("imgHot"+i).src = "${ctx}/static/images/yes.png";
console.log($("#imgHot").attr("src")); } else $("#imgHot"+i).attr("src","${ctx}/static/images/cancel.png"); if
("1" == $("#isShow"+i).val()) { document.getElementById("imgShow"+i).src = "${ctx}/static/images/yes.png"; }else
$("#imgShow"+i).attr("src","${ctx}/static/images/cancel.png");}

在实现添加商品分类功能过程中，最主要的两个部分是多选框级联查询和图片上传服务。对于实现两个级联选择的多选框
，首先需要在进入添加商品分类界面时，将顶级分类信息查询并循环添加到顶级分类多选框中，给每个<option>设置value字段
值并添加onchange()方法。当进行选择顶级分类后，本系统去查询以该顶级分类为父分类的子分类，请求成功后，将返回数据
进行拼接成html代码，根据下一级多选框id将拼接的<option>标签设置进入多选框中，同时将一隐藏输入框的值设置为该顶级
分类的id作为父分类id、将另一隐藏输入框的值设置为2作为分类等级。选择二级分类时，只需更新两个隐藏输入框的值。

//选择顶级分类返回成功时
if (result.length > 0) { for (i = 0; i < result.length; i++) { htmlStr += "<option value=' " + result[i].id
+ "'>" + result[i].id + "," + result[i].name + "</option>" } $("#"+ next).html(htmlStr); $("##level").val(2);}
else { $("##level").val(2); layer.alert("无子分类");}

//选择二级分类后修改parentId和levelif (0==parentId) {
$("#parentId").val($("#parent_id_1").val());$("##level").val(2);return;}// 修改parentId和
level$("#parentId").val(parentId);$("##level").val(level);

对于图片上传服务，本系统编单独编写了一个上传服务。
```

指 标	
疑似剽窃文字表述	
1.	Abstract: With the development of social economy and science and technology,
2.	论文编撰说明本系统。 2 开发环境及相关技术工具介绍 2.1 开发环境 IDEA
3.	Spring Boot框架是由Pivotal团队开发提供的一种全新框架，该框架的设计目的是用来简化新Spring应用的初始搭建以及开发过程。
4.	2 Mysql MySQL 是一款高效的、安全、跨平台，并与 PHP、Java 等主流编程语言紧密结合的数据库系统。 2.3 开发工具 2.3.1 JDK JDK是一个程序开发环境。它由一个处于操作系统层之上的运行环境还有开发者编译，调试和运行用Java语言写的 applet和应用程序所需的工具组成。
5.	2. Maven Maven 是来自Apache下的一个开源项目，它是一个创新的项目管理工具，它用于对Java项目进行项目构建、依赖管理及项目信息管理。
6.	2.3.5 Mybatis MyBatis 是一款优秀的持久层框架，它支持自定义 SQL、存储过程以及高级映射。MyBatis 免除了几乎所有的 JDBC 代码以及设置参数和获取结果集的工作。
3	

2. 基于spring boot框架的网上商场购物平台的设计与实现_第2部分		总字数: 11927
相似文献列表		
去除本人文献复制比: 6%(721) 文字复制比: 6%(721) 疑似剽窃观点: (0)		
1	基于位置的O2O服务推荐系统研究与实现 张志翱(导师: 夏宇) - 《江西师范大学硕士论文》- 2018-05-01	1.2% (145) 是否引证: 否
2	IPO审计风险研究 郑鼎伦(导师: 傅頔) - 《浙江财经大学硕士论文》- 2019-06-30	1.2% (143) 是否引证: 否
3	核糖体蛋白RPL34及其假基因RPL34-PS1促进B细胞淋巴瘤生长及其机制研究 周芳超(导师: 郁多男) - 《扬州大学硕士论文》- 2020-06-30	1.2% (143) 是否引证: 否
4	基于客货运需求预测的赤峰玉龙机场发展路径研究 孙小丽(导师: 毛佳;尹贵军) - 《吉林大学硕士论文》- 2013-12-01	1.2% (143) 是否引证: 否
5	我国中央和地方政府间财权与事权的平衡性研究 刘园(导师: 侯宇) - 《郑州大学硕士论文》- 2016-05-01	1.2% (143) 是否引证: 否
6	我国创业板并购重组绩效及影响因素研究	1.2% (143)

	荀宇(导师: 丁骋骋) - 《浙江财经大学硕士论文》 - 2016-12-01	是否引证: 否
7	高校大学生创新创业工作研究 孙黎明(导师: 钱明辉) - 《云南师范大学硕士论文》 - 2017-05-24	1.2% (143) 是否引证: 否
8	围治疗期使用内分泌抑制药对功能性垂体腺瘤伽玛刀治疗后内分泌缓解影响的Meta分析 涂德文(导师: 梁日生) - 《福建医科大学硕士论文》 - 2017-06-01	1.2% (143) 是否引证: 否
9	新疆物流对经济增长影响因素空间计量分析 马军(导师: 王合玲) - 《新疆财经大学硕士论文》 - 2018-05-19	1.2% (143) 是否引证: 否
10	关于中美电子信息产品贸易的研究 许晓珊(导师: 牛飞亮) - 《云南大学硕士论文》 - 2019-03-01	1.2% (143) 是否引证: 否
11	马尾松树轮不同组分稳定碳同位素对气候变化响应敏感性的研究 雷超(导师: 王建;商志远) - 《南京师范大学硕士论文》 - 2017-05-13	1.2% (143) 是否引证: 否
12	公立医院员工绩效考核研究 张谦(导师: 祝捷) - 《云南师范大学硕士论文》 - 2020-06-07	1.2% (143) 是否引证: 否
13	智慧高校综合管理系统的设计与实现 张翼(导师: 朱世杰;王全福) - 《东北大学硕士论文》 - 2015-11-01	1.2% (143) 是否引证: 否
14	职业倦怠对强迫购买影响机制研究 岳中林(导师: 易牧农) - 《天津财经大学硕士论文》 - 2017-05-01	1.2% (141) 是否引证: 否
15	尼罗罗非鱼胚胎干细胞系的建立及白血病抑制因子维持其未分化状态的研究 范振华(导师: 魏静) - 《西南大学硕士论文》 - 2017-04-10	1.2% (138) 是否引证: 否
16	醴陵文化园景观设计 何静(导师: 王海峰) - 《湖南工业大学硕士论文》 - 2017-06-03	1.2% (138) 是否引证: 否
17	基于H5的手表官网的设计与实现 孙腾 - 《大学生论文联合比对库》 - 2020-04-27	1.0% (120) 是否引证: 否
18	微信点餐系统的设计与开发——毕业设计论文 - 《大学生论文联合比对库》 - 2018-05-24	0.9% (108) 是否引证: 否
19	银企关系对我国中小企业运行稳定性的影响研究 廖乐一(导师: 罗传建) - 《华中科技大学硕士论文》 - 2015-05-01	0.9% (107) 是否引证: 否
20	欣德米特调性理论引申的结构力研究 刘晶晶(导师: 张璟) - 《武汉音乐学院硕士论文》 - 2017-06-30	0.9% (107) 是否引证: 否
21	出血性关节炎动物模型建立及血友病关节炎康复疗效观察 李豫(导师: 姚黎清) - 《昆明医科大学硕士论文》 - 2019-04-01	0.9% (107) 是否引证: 否
22	五大连池市污水处理二期工程环境影响评价 刘立雪(导师: 林永波) - 《东北林业大学硕士论文》 - 2014-04-01	0.6% (71) 是否引证: 否
23	北京市地铁移动票务系统的设计与实现 樊肖锦(导师: 张军) - 《首都经济贸易大学硕士论文》 - 2019-06-30	0.6% (70) 是否引证: 否
24	国产新能源汽车产业发展及竞争策略研究 关琦(导师: 夏友富) - 《对外经济贸易大学硕士论文》 - 2017-05-01	0.6% (69) 是否引证: 否
25	基于android的“晓”视频APP的设计与实现 张志明 - 《大学生论文联合比对库》 - 2019-05-03	0.5% (64) 是否引证: 否
26	滇池流域村镇污水处理WebGIS平台的设计与开发 刘圣雅(导师: 梁虹) - 《云南大学硕士论文》 - 2018-05-01	0.5% (61) 是否引证: 否
27	“截贿”行为刑法规制研究 李广大(导师: 袁林) - 《西南政法大学硕士论文》 - 2019-03-27	0.5% (57) 是否引证: 否
28	3994734_林丽_网上服装销售管理系统设计与实现 林丽 - 《大学生论文联合比对库》 - 2017-04-16	0.4% (45) 是否引证: 否
29	基于Java的网上中医药店管理系统 苏宏 - 《大学生论文联合比对库》 - 2020-04-23	0.3% (32) 是否引证: 否
30	西南石油大学桶装水配售系统分析与设计 夏洋 - 《大学生论文联合比对库》 - 2018-06-20	0.2% (29) 是否引证: 否

原文内容

首先需要在七牛云服务器注册申请一个储存对象,更新dns服务器地址以解析上传图片的域名。当进行上传图片服务时,在本系统中利用传入的图片名和当下本地时间拼接成新的图片名,再根据与七牛云服务器的存储对象凭据对应,完成上传图片服

务，最后使用上传成功的回调方法将图片的url设置在一个隐藏输入框中。

```
//UploadController调用上传服务和修改图片名
String filename = file.getOriginalFilename();String date = DateTimeFormatter.ofPattern("yyyy-MM-dd").format(LocalDate.now());filename = date + "-" + System.currentTimeMillis() + "-" + filename;
//.substring(filename.lastIndexOf("."))return uploadService.upload(file.getInputStream(),filename);
//上传图片凭据
Configuration cfg = new Configuration(Region.region2());UploadManager uploadManager = new UploadManager(cfg);
//...生成上传凭证，然后准备上传String accessKey = "z3Ssv1YyolMz1e671cv5PFRoT7QtEf01LpkbjrEj";String secretKey = "3pqTW5qk3mRRIPnEyGQi4EhEAfaNiUy00gKxq08i";String bucket = "shoponline-pjp";
//上传图片
fileResult.setSuccess("success");fileResult.setMessage("上传图片成功");
fileResult.setFileUrl("http://app2.merborn.fun/"+fileName);
//上传成功回调设置
$("#image").val(data.response.fileUrl);
最后点击提交按钮，调用serialize()方法对表单中各个输入框的值进行序列化，然后利用ajax请求将数据传入后台，后台根据传入数据与实体的字段名匹配封装成完整对象[10]，再删除redis中分类信息缓存并将该对象存储到数据库中，完成添加商品分类操作。
//前端发送ajax请求存储分类数据
$.ajax({ url: "${ctx}/goods/category/save", type: "post", data: $("#category_form").serialize(), dataType: "JSON", success: function (result) { if (result.code == 200) { layer.confirm("保存成功", {btn: ['继续新增', '返回列表']}, function () { window.location.href = "${ctx}/goods/category/add"; }, function () { window.location.href = "${ctx}/goods/category/list"; }); } else { layer.alert("保存失败"); } },
//GoodsCategoryController调用方法存储新增分类
int resultCol = goodsCategoryService.categorySave(goodsCategory);return resultCol>0?BaseResult.success():BaseResult.error();
//删除Redis缓存，数据存入数据库redisTemplate.delete(redisTemplate.keys("goods*"));return goodsCategoryMapper.insertSelective(goodsCategory);
对商品分类删除功能，采用前端和后台同步删除的方式，保证了数据的一致性。同时前端异步删除标签，避免了删除后需要重新刷新页面的不便利操作。点击删除按钮，用调用del_fun()方法，该方法调用一个递归方法returnResult()以便在删除父分类时，先删除子分类，到达根据外键关系级联删除数据的目的。
具体流程是：先将删除分类的id传入后台，后台查明该分类属于几级分类。若该分类为三级分类，后台则删除数据库中该分类信息，然后直接返回一个仅存一个对象且对象id为-1的列表，前端判断该分类无子分类，直接调用document.getElementById().remove()方法删除该分类所属的父标签。若该分类有子分类，则查询子分类信息并返回，前端循环递归查询每个子分类情况，直至从最底层分类开始往上删除。最后删除redis中缓存分类信息。
//前端递归删除标签
if(result[0].id != -1 ){ for(var i = 0;i < result.length;i++){ //obj是当使用
$(obj).parent().parent().remove();方法是使用的，现无需。 returnResult(null,del_url,result[i].id);
console.log(result[i].id);}else console.log("无子类数据");document.getElementById("gcvList"+id).remove();
//后台创建最底层分类标志信息
goodsCategory3.setId((short)-1);goodsCategories.add(goodsCategory3);
//后台查询预删除分类的信息GoodsCategory goodsCategory = goodsCategoryService.categorySelectOne(deleteId);
//后台删除分类信息
goodsCategoryService.categoryDelete(deleteId);//如果该分类为一级分类if(goodsCategory.getLevel() == 1){ //查询该分类的二级分类列表 List<GoodsCategory> list2 = goodsCategoryService.selectCategoryByParentId(deleteId);
if(list2.isEmpty()) return goodsCategories; else return list2;
//如果该分类为二级分类}else if(goodsCategory.getLevel() == 2) { List<GoodsCategory> list3 = goodsCategoryService.selectCategoryByParentId(deleteId); if(list3.isEmpty()) return goodsCategories; else return list3;
//如果该分类为三级分类}else{return goodsCategories;}//删除Redis缓存
redisTemplate.delete(redisTemplate.keys("goods*"));return goodsCategoryMapper.deleteByPrimaryKey(id);
修改商品分类信息分为两个部分完成。对于修改商品分类是否推荐和是否显示仅需点击显示状态图片就可以切换，修改分组和排序信息仅需在显示页面修改数值便可。但对于修改商品分类的具体信息，管理员需前往category-update.ftl页面才能修改。该页面查询到该分类的信息并填充到对应输入框中，管理员仅需修改输入框中数值，选择提交便可，后台将表单数据序列化并传入后台，后台进行更新数据操作达到修改分类信息目的。最后后台将存储在redis中的分类信息删除。
//修改分类列表ishot，同修改分类是否显示function changeIshot(id,isHot,obj){var src = "";
if($("#obj").attr('src').indexOf("cancel.png") > 0 ) {src = '${ctx}/static/images/yes.png';var hot = 1; }else{src = '${ctx}/static/images/cancel.png';var hot = 0; } $.ajax({ url:"${ctx}/goods/category/changeIshot", type: "POST", data: { id: id, isHot: hot,}, dataType: "JSON", success: function(data) { $(obj).attr('src',src);}, error:function (result) {console.log(result);}});
/// 修改分类列表的分组，同修改分类排序function updateCatGroup(id,obj){ var value = $(obj).val();
console.log(value); $.ajax({ url:"${ctx}/goods/category/changeCatGroup", type: "POST", data: { id: id,
```



```
catGroup: value, }, dataType: "JSON", success: function(data) { layer.msg('更新成功', {icon: 1}); }, error: function(result) { console.log("error"); console.log(result); } });
```

### 5.1.3 管理商品子模块

该模块功能类同与商品分类模块，除此之外主要添加了三多选框级联查询、添加在线编辑器、商品列表显示分页、三个搜索条件查询商品信息等功能。

三多选框级联查询分类为新增商品时添加商品分类标签使用，功能实现类似于两多选框级联查询。

添加在线编辑器的目的是管理者在添加新商品时，管理者能用多样的工具来具体描述商品的信息，后台将详细信息描述的html代码转义成字符串存储在数据库中。

//在线编辑器

```
<textarea rows="6" cols="80" class="span12 ckeditor" id="goods_content" name="goodsContent" title=""></textarea>
```

//html文本转义if(!StringUtil.isEmpty(goods.getGoodsContent())){

```
goods.setGoodsContent(HtmlUtils.htmlEscape(goods.getGoodsContent(),"UTF-8"));
```

在添加商品完成后，本系统会将代表商品id的输入框值更新，当管理员重复提交增加商品时，本系统会禁止重复提交相同商品信息。

//判断是否连续添加了两次相同信息的商品if(null != goods.getGoodsId()){ return BaseResult.error();}

分页查询返回分页对象，前台拿到数据进行分页渲染

//创建redis存储格式

```
String[] goodsKeyArr = new String[]{ "goods:pageNum_"+page+":pageSize_"+size+":", "catId:", "brandId:", "goodName_:" };//构建分类对象PageHelper.startPage(page, size);
```

```
goodsListKey = Arrays.stream(goodsKeyArr).collect(Collectors.joining());ValueOperations<String, String>
```

```
valueOperations = redisTemplate.opsForValue();String pageInfoGoodsJson =
```

```
valueOperations.get(goodsListKey);if(!StringUtil.isEmpty(pageInfoGoodsJson)){ return
```

```
BaseResult.success(JsonUtil.jsonStr2JsonObject(pageInfoGoodsJson, PageInfo.class));}List<Goods> list =
```

```
goodsMapper.selectByExample(goodsExample);if(!CollectionUtils.isEmpty(list)) { PageInfo<Goods> pageInfo = new
```

```
PageInfo<>(list); valueOperations.set(goodsListKey, JsonUtil.object2JsonStr(pageInfo)); return
```

```
BaseResult.success(pageInfo);}else { valueOperations.set(goodsListKey, JsonUtil.object2JsonStr(new PageInfo<>
```

```
(new ArrayList<Goods>()), 60, TimeUnit.SECONDS);}
```

//前台获得返回分页数据进行分页渲染

```
if (result.pageInfo.list.length > 0) { //获取商品列表模板 var goodsTemp = doT.template($("#goodsTemplate").text()); //填充数据 $("#goodsContent").html(goodsTemp(result.pageInfo.list));
```

```
//获取分页模板 var pageTemp = doT.template($("#pageTemplate").text()); //填充数据
```

```
$("#pageContent").html(pageTemp(result.pageInfo));} else { fyAlert.msg("该分类或品牌暂无商品信息!", {icon: 3,
```

```
animateType: 3}); //获取商品列表模板 var goodsTemp = doT.template($("#goodsTemplate").text()); //填充数据
```

```
$("#goodsContent").html(goodsTemp(result.pageInfo.list));}
```

三条件选择查询

```
//查询选择的分类if(null !=goods.getCatId() && 0 !=goods.getCatId()){ criteria.andCatIdEqualTo(goods.getCatId()); goodsKeyArr[1] = "catId_"+goods.getCatId()+":";}//同时查询选择的品
```

```
牌if(null !=goods.getBrandId() && 0 !=goods.getBrandId()) { criteria.andBrandIdEqualTo(goods.getBrandId());
```

```
goodsKeyArr[2] = "brandId_"+goods.getBrandId()+":";}//再同时查询选择的关键词
```

```
if(!StringUtil.isEmpty(goods.getGoodsName())) { criteria.andGoodsNameLike("%" + goods.getGoodsName() + "%");
```

```
goodsKeyArr[3] = "goodName"+goods.getGoodsName()+":";}
```

## 5.2 门面模块的实现

### 5.2.1 登录子模块

用户在未登录状态先可以随意浏览商品和搜索商品信息，当用户将商品添加到购物车或者访问购物车信息时，本系统会将消费者拦截至登录界面，以便登录个人信息。当消费者登录成功后，本系统生成唯一标志登录票据，同时把用户信息存入session、把票据信息存入redis和cookie中。

//配置拦截器拦截路径

```
public void addInterceptors(InterceptorRegistry registry) { registry.addInterceptor(loginInterceptor)
```

```
.addPathPatterns("/cart/**") .excludePathPatterns("/images/**", "/js/**", "/css/**")
```

```
.excludePathPatterns("/login/**") .excludePathPatterns("/user/login/**")
```

```
.excludePathPatterns("/user/logout/**");}
```

//拦截器条件设置

```
String ticket = CookieUtil.getCookieValue(request, "myTicket");System.out.println("拦截器
```

```
ticket:"+ticket);if(!StringUtil.isEmpty(ticket)){ Users users = ssoService.validate(ticket);
```

```
request.getSession().setAttribute("users",users); ValueOperations<String, String> valueOperations =
```

```
redisTemplate.opsForValue(); valueOperations.set(myTicket+":"+ticket, JsonUtil.object2JsonStr(users), 30,
```

```
TimeUnit.MINUTES); return true;}System.out.println("跳到登录界面
```

```
");response.sendRedirect(request.getContextPath()+"/login");
```

//验证登录信息是否匹配，成功生成票据

```
if(!users1.getPassword().equals(Md5Util.getMd5WithSalt(users.getPassword(),"a2ck"))){
System.out.println("密码不匹配");return null;}ValueOperations<String,String> valueOperations =
redisTemplate.opsForValue();String ticket =
UUIDUtil.getUUID();System.out.println("ticket:"+ticket);valueOperations.set(myTicket+":"+ticket,
JsonUtil.object2JsonStr(users1),30,TimeUnit.MINUTES);
//验证票据是否合法
ValueOperations<String,String> valueOperations = redisTemplate.opsForValue();String usersJson =
valueOperations.get(myTicket + ":" + ticket);if(StringUtil.isEmpty(usersJson)){System.out.println("票据信息不
匹配");return null;}return JsonUtil.jsonStr2Object(usersJson,Users.class);
```

### 5.2.2 搜索子模块

前端进行ajax请求将搜索商品的名字传入后台。后台进行模糊查询并返回查询数据给前端，最后前端将返回数据进行页面信息渲染，完成搜索商品功能实现。

```
//再同时查询选择的关键词if(!StringUtil.isEmpty(searchStr)) {
goodsExample.createCriteria().andGoodsNameLike("%" + searchStr + "%");}List<Goods> list =
goodsMapper.selectByExample(goodsExample);for (Goods goods : list) {
System.out.println(goods.getMarketPrice());}if(!CollectionUtils.isEmpty(list)) {
System.out.println("SearchService: 查询到商品");return list;}else {System.out.println("查询无信息");return
list;}
```

前端请求查询，然后将查询结果进行模板渲染，页面加载的时候执行

```
$(function () { doSearch();});// 搜索商品function doSearch() { $.ajax({ url: "${ctx}/search/searchGoods",
type: "POST", data: $("#s_search_form").serialize(), dataType: "JSON", success: function (result) {
console.log(result); // 调用模板 var templ = doT.template($("#goodsTemplate").text()); // 填充内容
$("#s_search_content").html(templ(result)); }, error: function (result) {alert("系统正在升级中，请稍后再试
!");}});}
```

### 5.2.3 添加购物车子模块

点击添加购物车，将调用点击事件addToCart()，后台将传参数据封装成cartVo对象，进行一系列验证将商品信息加入redis中。

首先查询该用户的购物车信息。

```
Integer userId = users.getUserId();hashOperations = redisTemplate.opsForHash();Map<String, String> cartMap
= hashOperations.entries(userCart + ":" + userId);
```

当购物车为空时，则以商品id为键值将商品对象存入map中。

```
//购物车为空cartMap = new HashMap<>();System.out.println("CartService: 新增购物车信息
");cartMap.put(String.valueOf(cartVo.getGoodsId()),JsonUtil.object2JsonStr(cartVo));
```

当购物车不为空，但所添加商品为第一次添加时

```
System.out.println("CartService: 现购物车无商品,新增商品信息到购物车
");cartMap.put(String.valueOf(cartVo.getGoodsId()),JsonUtil.object2JsonStr(cartVo));
```

当购物车不为空，且购物车内已添加过该商品时

```
if(!StringUtil.isEmpty(cartStr)){ //查询到商品信息 CartVo vo = JsonUtil.jsonStr2Object(cartStr,
CartVo.class); if(vo != null) { //修改商品数量 vo.setGoodsNum(vo.getGoodsNum() + cartVo.getGoodsNum());
System.out.println("CartService: 商品数量"+vo.getGoodsNum()); //修改商品最新的价格
vo.setMarketPrice(cartVo.getMarketPrice()); //覆盖以前的商品对象 cartMap.put(String.valueOf(vo.getGoodsId()),
JsonUtil.object2JsonStr(vo)); System.out.println("CartService: 商品存在"); }else
System.out.println("CartService: 商品不存在");
```

## 6 系统展示

### 6.1 后台管理模块展示

#### 6.1.1 登录界面展示

访问管理平台登录界面或者未登录状态下访问后台管理界面，如图6.1所示：

图6.1 登录页面

#### 6.1.2 商品分类列表展示

点击管理商品分类界面如图6.2所示：

图6.2 商品分类界面

当点击新增按钮时，页面跳转到新增分类界面，如图6.3所示：

图6.3 新增商品分类

点击渲染分类的是否推荐或者是否展示的状态图片，就能在前台和后端同步修改数据。如图6.4所示：

图6.4 修改推荐和显示状态

修改分组输入框或者排序输入框的数值，便能前台和后端同步修改数据。如图6.5所示：

图6.5 修改分组和排序

点击修改按钮进入修改商品分类详细信息界面，如图6.6所示：

图6.6 进入修改商品分类界面

点击删除按钮，后台根据分类是否有子分类，递归从下往上逐一删除，如图6.7和图6.8删除新希望牛奶分类所示：

图6.7 删除分类前



图6.8 删除分类后

### 6.1.3 商品列表展示

点击商品列表进入商品展示界面，如图6.9所示：

图6.9 商品展示界面

根据搜索条件，进行搜索商品信息查询，如图6.10所示：

图6.10 搜索显示

新增商品界面如图6.11和图6.12所示

图6.11 新增商品界面

图6.12 新增商品界面

删除商品信息如图6.13和图6.14删除id为242的商品所示：

图6.13 删除之前

图6.14 删除之后

## 6.2 门户模块展示

### 6.2.1 登录界面展示

当用户选择登录或者想将商品加入购物车时，都会跳到登录界面，如图6.15所示：

图6.15 登录界面

### 6.2.2 首页展示

商品首页显示如图6.16所示：

图6.16 商城首页

### 6.2.3 搜索展示

搜索功能展示由图6.17搜索huawei所示：

图6.17 搜索页面

### 6.2.4 添加购物车展示

将商品加入购物车如图6.18所示，购物车商品数量随着改变：

图6.18 加入购物车

结论

通过近三个多月的学习和努力，在指导老师的耐心辅导下，网上购物商城系统的开发和论文写作最终得到了完成。在开发本系统在过程中，出现了许多问题，这些问题都记录在开发日志中，根据查询的资料和请教指导老师一一解决了这些难题，同时把解决方法也记录日志之中。本系统相比较其他大型网上购物商城优化了系统使用者的使用体验，弥补了细节瑕疵，也运用了如递归删除商品分类这样的个性化算法写法。但由于成本和时间的限制，本系统还有许多地方有待改善，但本系统功能基本完善，用户有完整的使用流程。本系统也统筹考虑了开发工作量、开发和维护成本、功能鲁棒性等问题，均衡各个问题的关键进行了完整开发过程。

参考文献

- [1] 王力平. 商务网站“秒杀”营销对消费者网上购物影响的实证研究——以京东网上商城“夺宝岛”“秒杀”营销为例[J]. 消费经济, 2012, 000(005):59-61.
- [2] 王景河. 网上商店购物者心理分析[J]. 商场现代化, 2002(6):12-12.
- [3] 蒋道霞, 秦媛媛, 何玉林. 基于“线下+线上”的大数据编程环境构建实践[J]. 职业教育(中旬刊), 2020, 19(12):66-68.
- [4] 张振球. 基于SpringBoot的校园文章发布系统设计与实现[J]. 无线互联技, 2019, 16(23):41-43.
- [5] 黄艳秀. 基于mybatis的面向数据库自动生成技术[J]. 河南科技, 2014, 000(004):21-22.
- [6] 秦漂, 艾青. 基于Redis技术的页面缓存研究[J]. 科技风, 2021(12):74-76.
- [7] 张峰. 应用SpringBoot改变web应用开发模式[J]. 科技创新与应用, 2017, 000(023):193-194.
- [8] 陈钊. SSM框架在Web应用开发中的设计与实现研究[J]. 电脑知识与技术, 2021, 17(10):226-227.
- [9] 王永和, 张劲松, 邓安明, 等. Spring Boot研究和应用[J]. 信息通信, 2016, 000(010):91-94.
- [10] 杜波瓦. MySQL网络数据库指南[M]. 机械工业出版社, 2000.

致谢

本论文在李舸导师的悉心指导下完成的。在开发过程中，导师无时无刻的细心指导不仅使本人寻找到了系统开发的方向、探索出一条成功的路径、掌握了基本的研究实现方法，还使本人明白了许多为人处事的道理。本次论文从选题到完成，每一步都是在导师的悉心指导下完成的，倾注了导师大量的心血。在写论文的过程中，遇到了很多的问题，在老师的耐心指导下，问题都得以解决。

作者简介：

姓名：蒲俊鹏性别：男

出生年月：1997年11月10日民族：汉

E-mail:pjp\_19102834276@163.com

声明

本论文的工作是2020年12月至2021年06月在成都信息工程大学网络空间安全学院完成的。文中除了特别加以标注地方外，不包含他人已经发表或撰写过的研究成果，也不包含为获得成都信息工程大学或其他教学机构的学位或证书而使用过的材料。

关于学位论文使用权和研究成果知识产权的说明：

本人完全了解成都信息工程大学有关保管使用学位论文的规定，其中包括：

- (1) 学校有权保管并向有关部门递交学位论文的原件与复印件。
- (2) 学校可以采用影印、缩印或其他复制方式保存学位论文。

(3) 学校可以学术交流为目的复制、赠送和交换学位论文。  
(4) 学校可允许学位论文被查阅或借阅。  
(5) 学校可以公布学位论文的全部或部分内容(保密学位论文在解密后遵守此规定)。  
除非另有科研合同和其他法律文书的制约, 本论文的科研成果属于成都信息工程大学。  
特此声明!  
作者签名:  
2021年月日

指 标
疑似剽窃文字表述
1. 加入购物车时, 都会跳到登录界面, 如图6.15所示: 图6.15 登录界面 6.2.2
2. 掌握了基本的研究实现方法, 还使本人明白了许多为人处事的道理。本次论文从选题到完成, 每一步都是在导师的悉心指导下完成的, 倾注了导师大量的心血。在写论文的过程中, 遇到了很多的问题, 在老师的耐心指导下, 问题都得以解决。

说明:

1. 总文字复制比: 被检测论文总重合字数在总字数中所占的比例
2. 去除引用文献复制比: 去除系统识别为引用的文献后, 计算出来的重合字数在总字数中所占的比例
3. 去除本人文献复制比: 去除作者本人文献后, 计算出来的重合字数在总字数中所占的比例
4. 单篇最大文字复制比: 被检测文献与所有相似文献比对后, 重合字数占总字数的比例最大的那一篇文献的文字复制比
5. 指标是由系统根据《学术论文不端行为的界定标准》自动生成的
6. 红色文字表示文字复制部分; 绿色文字表示引用部分; 棕灰色文字表示作者本人文献部分
7. 本报告单仅对您所选择比对资源范围内检测结果负责



 [amlc@cnki.net](mailto:amlc@cnki.net)

 <http://check.cnki.net/>

 <http://e.weibo.com/u/3194559873/>