C:\Users\chenyu\AppData\Local\Temp\ksohtml\wpsEF83.tmp.jpg

毕业论文（设计）初稿

题 目 商城管理系统的设计与实现

学生姓名 谭佳利

学 号 201814160116

学 院 计算机与人工智能学院

专 业 网络工程

届 别 2018级

指导教师 张霞

职 称 副教授

2021 年 10 月 25 日

摘要

随着计算机和网络的日趋普及，越来越多的商家在网上建起商店，向消费者展示出一片新鲜的购物大地。在美、日等信息化程度较高的国家和地区，网络商店发展速度迅猛，美国的世界级超一流的零售商，如沃尔玛凯玛特家庭仓储、科罗格、J.C培尼等纷纷跻身于网络经商的行列。调查表明:美国的家庭已越来越习惯于在家中从网上购物。

在我国，网上购物从无到有也不过短短几年时间。网购商品也非常丰富:产品涉及图书、影视、音乐、软件、教育音像、游戏/娱乐、消费电子、商品/通讯、家电、电脑/配件、摄影/摄像、MP3/MP4 ，视听/车载、日用消费品、个人护理、钟表首饰、礼品箱包、玩具、厨具、母婴产品、化妆、家居、运动健康等等。可以用一句话来形容网购——只有你想不到的商品或服务，没有提供不了的商品或服务。网购，正在成为一种消费时尚。

网络购物已经成为一种趋势，且网购时代已经来临。但是中国的网购途径依旧表现的一家独大，对商品是否能够出售，是否合法，管理的繁琐程度，都是短时间不可解决的问题。再者，网络安全问题日益严重，一款系统对用户信息的保护是否可靠也被人们日益关注。

关键词：网络购物、消费、商品管理

目录

[第一章 绪论 5](#_Toc26432)

[1.1. 研究背景与意义 5](#_Toc177)

[1.2. 研究现状 5](#_Toc15344)

[1.3. 文本主要工作 6](#_Toc5225)

[第二章 关键技术介绍 6](#_Toc18831)

[2.1. 后端技术介绍 6](#_Toc5853)

[2.2. 前端技术运用 7](#_Toc12125)

[2.3. 系统架构设计模型 7](#_Toc20624)

[第三章 系统分析 7](#_Toc10516)

[3.1. 需求分析 7](#_Toc13988)

[3.1.1. 功能需求 7](#_Toc9808)

[3.1.2. 用户需求分析 7](#_Toc13603)

[3.1.3. 硬件需求 8](#_Toc14927)

[3.1.4. 性能分析 8](#_Toc29773)

[3.2. 系统可行性分析 8](#_Toc3710)

[3.2.1. 技术可行性分析 8](#_Toc14680)

[3.2.2. 经济可行性分析 8](#_Toc5812)

[3.2.3. 操作可行性分析 8](#_Toc18003)

[3.3. 功能性需求分析 9](#_Toc31757)

[3.4. 数据库分析 9](#_Toc31244)

[第四章 系统设计 9](#_Toc7098)

[4.1. 软件设计原则 9](#_Toc14369)

[4.2. 系统架构设计 9](#_Toc8521)

[4.3. 系统模块设计 10](#_Toc8874)

[4.4. 功能详情设计 10](#_Toc16431)

[4.5. 数据库设计 12](#_Toc14874)

[4.6. 本章小结 12](#_Toc28312)

[第五章 系统实现 12](#_Toc23885)

[5.1. 系统开发环境 12](#_Toc21454)

[5.2. 主要功能模块实现 12](#_Toc13020)

[5.2.1. 管理员登录页面功能实现： 13](#_Toc28336)

[5.2.2. 资源分配与功能菜单显示： 15](#_Toc3625)

[第六章 系统测试 17](#_Toc23827)

[6.1. 测试目的和分类 17](#_Toc13766)

[6.2. 测试环境 17](#_Toc13063)

[6.3. 主要功能测试 17](#_Toc24103)

[6.3.1. 功能性测试 17](#_Toc2382)

[6.3.2. 集成性测试 18](#_Toc696)

[6.3.3. 性能测试 18](#_Toc25739)

[6.4. 测试结果 18](#_Toc22735)

[第七章 结论 18](#_Toc25335)

[致谢 18](#_Toc13081)

[参考文献 18](#_Toc19604)

# 绪论

## 研究背景与意义

随着计算机和网络的日趋普及，越来越多的商家在网上建起商店，向消费者展示出一片新鲜的购物大地。在美、日等信息化程度较高的国家和地区，网络商店发展速度迅猛，美国的世界级超一流的零售商，如沃尔玛凯玛特家庭仓储、科罗格、J.C培尼等纷纷跻身于网络经商的行列。调查表明:美国的家庭已越来越习惯于在家中从网上购物。

在我国，网上购物从无到有也不过短短几年时间。网购商品也非常丰富:产品涉及图书、影视、音乐、软件、教育音像、游戏/娱乐、消费电子、商品/通讯、家电、电脑/配件、摄影/摄像、MP3/MP4 ，视听/车载、日用消费品、个人护理、钟表首饰、礼品箱包、玩具、厨具、母婴产品、化妆、家居、运动健康等等。可以用一句话来形容网购——只有你想不到的商品或服务，没有提供不了的商品或服务。网购，正在成为一种消费时尚。

网络购物已经成为一种趋势，且网购时代已经来临。但是中国的网购途径依旧表现的一家独大，对商品是否能够出售，是否合法，管理的繁琐程度，都是短时间不可解决的问题。再者，网络安全问题日益严重，一款系统对用户信息的保护是否可靠也被人们日益关注。

## 研究现状

现如今，我们所处于互联网时代，[电子](http://www.youerw.com/tongxin/" \t "http://www.youerw.com/yanjiu/_blank)商务的流行必不可少，消费者已拥有了一种全新的新潮购物思想。在国外等一些信息化水平较高的国家而言，网上购物的方式以迅雷不及掩耳之势的速度发展着。

调查[研究](http://www.youerw.com/yanjiu/" \t "http://www.youerw.com/yanjiu/_blank)表明，美国人已经越来越习惯于在家中购物了。据《中国商界》杂志报道，美国人[网络](http://www.youerw.com/jisuanji/" \t "http://www.youerw.com/yanjiu/_blank)购物成时尚，美国是世界上电子商务最发达的国家，也拥有全球最大的网上购物市场。

美国人推崇时尚，追求效率，这是美国网上购物市场发展迅猛的主要原因，但更重要的是，由于法制上的完善，越来越多消费者对在线购物感到放心.同时，美国人讲究实惠，而越来越多的网上商店千方百计不断满足消费者的需要，使网上购物成为一种时尚和乐趣。

当前，英国网上购物方兴未艾，势头正猛。英国几家机构前几年的调查结果显示，同欧洲其他国家和美国相比，英国网上购物者占人口比例最高，购买商品种类最多，购物增长速度最快。

法国人网上购物成习惯，据法国电子商务及远程销售联合会2008年1月24日公布的数据，去年法国网上购物继续保持强劲上涨势头，网上购物总额大幅增加。数据显示，法国去年网上购物交易总额达161亿欧元，增长了35%，而2004年网上购物总额还只有57亿欧元。该联合会预计，到2010年，法国网上购物交易总额将达到314亿欧元 24226  
我国的网络购物的历史不长，自1998年3月6日下午3:20国内的第一笔网上交易成功，中国的网络购物开始出现在人们的视线中。

1999年底，互联网开始沸腾，国内开始有300多家网络公司从事B2C这个行业。2000年，递增至700余家，但是，让人印象深刻的却仅三四家，那是一段漫长的谷底期。直至SARS的出现，人们不敢出门，便开始热爱上了足不出户的购物方式，于是，购物网站开始逐渐的成功上线，从而一步步到了今天。

## 文本主要工作

本论文综合运用Java Web开发的各项技术，深入研究使用Spring Boot、Mybatis和进行企业级开发的基本方法，设计实现了商城管理系统。实现登录注册、个人中心、购物车、搜索商品、商家对话、评论等基本功能。

第一章介绍了课题相关的发展现状以及目前面临的一些问题。第二章确定基本的设计思想和需要采用的技术，并对相关的技术做出了基本的介绍。第三章根据需求对系统的功能做出了具体的规划。第四章是系统总体设计，包括系统的数据库设计和各模块的详细设计，介绍了本系统中主要模块的实现方式.第五章讲述的是系统实现，详细介绍了本系统中主要技术的运用以及一些配置文件的设计．第六章进行系统的测试。

# 关键技术介绍

## 后端技术介绍

* **Spring Boot：**Spring Boot可以轻松创建独立的、生产级的、基于Spring的应用程序，更简洁的开发框架。 它对Spring平台和第三方库持一种固执己见的观点，使得开发者可以无需过多配置就可以开始开发。大多数Spring引导应用程序需要最少的Spring配置。还集成了多种三方框架及中间件，促使框架功能性的强大。在我们开发过程中提供了更好的开发体验。
* **MyBatis|MyBatis Plus：**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。MyBatis 可以通过简单的 XML 或注解来配置和映射原始类型、接口和 Java POJO（Plain Old Java Objects，普通老式 Java 对象）为数据库中的记录。而国人开发的Plus增强版，又将这个框架再次集成封装，是一个 MyBatis (opens new window)的增强工具，在 MyBatis 的基础上只做增强不做改变，为简化开发、提高效率而生。
* **Mysql：**最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。
* **Redis：**是一个由 Salvatore Sanfilippo 写的 key-value 存储系统，是跨平台的非关系型数据库。Redis 是一个开源的使用 ANSI C 语言编写、遵守 BSD 协议、支持网络、可基于内存、分布式、可选持久性的键值对(Key-Value)存储数据库，并提供多种语言的 API。Redis 通常被称为数据结构服务器，因为值（value）可以是字符串(String)、哈希(Hash)、列表(list)、集合(sets)和有序集合(sorted sets)等类型。

## 前端技术运用

* **Vue：**是一套构建用户界面的渐进式框架。只关注视图层， 采用自底向上增量开发的设计。Vue 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件。
* **Element UI：**一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库，配合Vue使用。

## 系统架构设计模型

客户端——服务器——云服务器——后台管理

# 系统分析

## 需求分析

### 功能需求

系统主要服务于售卖商品的个体商户，以及购买该商户产品的普通群众。系统主要分为前端商城页面和后端可视化商品管理页面。普通用户通过访问商城页面，可正常选购相关商品，包括在线商城必备的功能，如商品点赞、收藏、评论、加入购物车、支付等常用功能，还支持发布动态，退货系统、实时聊天等。可视化管理页面供商户及管理员使用，有资源分配，文件上传，商品管理与审核，商品销售指标总览，产品运维等功能。

### 用户需求分析

* 普通用户：登录、注册、支付、收藏、点赞、评论、发布动态、人工客服、个人中心设置、售后服务。
* 销售商：商品管理、商城通知、销量统计
* 商品销售运维人员：产品分析、销量预测、商品状态跟踪
* 系统运维人员：资源管理、产品审核

### 硬件需求

* 普通用户：PC端、手机端适配
* 管理员：PC端

### 性能分析

* 商城系统：普通用户浏览商品，存在峰值，需要保证某个时间段内，服务的高可用。
* 后台管理可视化工具：受用群体小，使用量低。

## 系统可行性分析

### 技术可行性分析

spring boot作为一个web网站简易开发的服务框架，拥有不错的性能以及开发可操作性、可扩展性，兼容当下市场的绝大部分服务框架、中间件。对于商城开发的后续功能改动和产品维护提供了各种监控工具，在当下开发任务下十分契合。

### 经济可行性分析

开发所使用的绝大多数工具以及中间件都是开源且免费的，几乎可以实现零成本开发。

### 操作可行性分析

系统使用的是传统商城样式模板UI，对于用户来说没有对购物界面做出任何改动，其功能使用方式也十分简洁，无需过多操作。对于管理员端界面，采用的是数据可视化处理，更直观的统计与管理。

## 功能性需求分析

1. 系统界面需比较美观，界面布局需紧凑合理，方便用户的使用.
2. 系统可维护性、可靠性：系统中各功能在维护时，应保证其易维护，且不易崩溃，增加其可靠程度。
3. 系统应该具有一定的扩展能力。系统需要具备一定的技术性。
4. 通过主页和登录建立良好的印象，让界面看起来更专业简洁，实现可公布信息的信息透明，符合用户预期,以此消除用户的不确定感。
5. 传达安全策略，建立安全范围。在一些操作上，增强用户的控制感。系统整体架构需要以业务模块进行分组，方便后期的服务化与扩展。

## 数据库分析

E-R图：

# 系统设计

## 软件设计原则

在进行软件设计过程中，遵循如下原则：

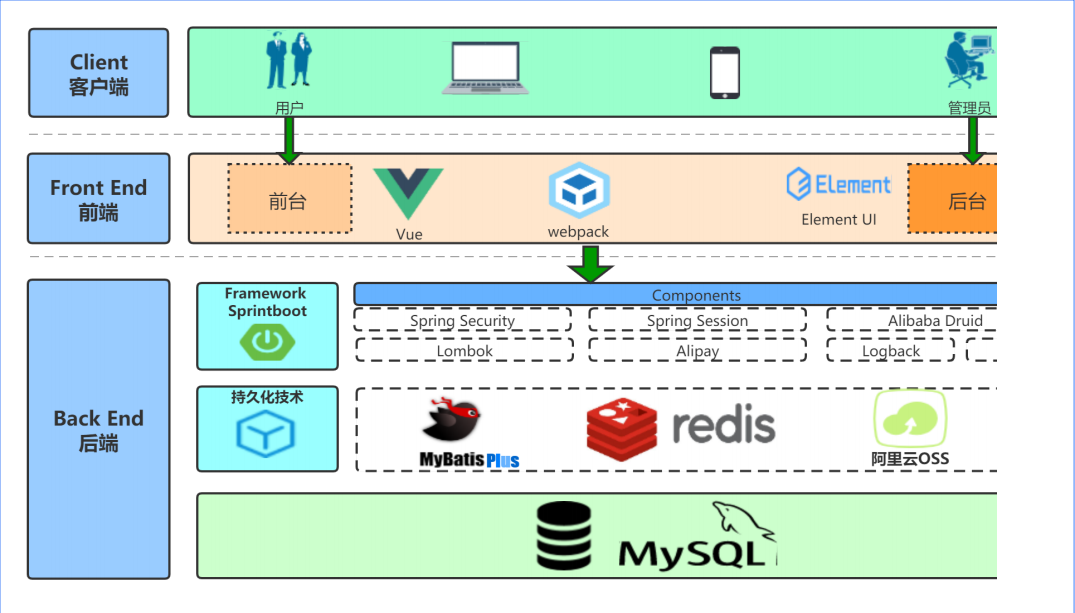
* 软件的可靠性原则，即在任何情况下，不论系统奔溃与否，都将保证系统能正常反馈给用户信息。
* 软件的信息安全性原则，即客户端向服务器发送请求，服务器响应数据，这个过程，需要对必要的信息进行特殊处理。
* 软件高可用性原则，对于大流量的商城前端页面，需要做到服务断融和服务降级。

## 系统架构设计

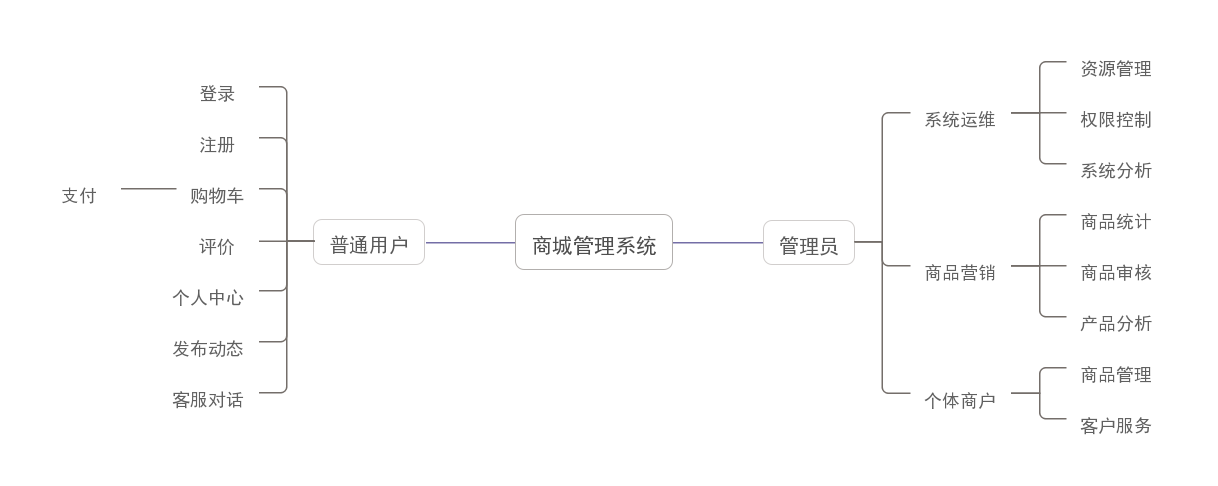
系统结合了实际情况的需求，以及自身技术的掌握情况对于本系统的后台采用了Spring Boot+Mybatis+MySQL作为后台的基础开发框架。Spring Boot，它简化了后台开发的流程，能快速的为项目的构建搭好基础的框架，而Mybatis作为优秀的持久层框架，为后台与数据库的交互提供了强有力的帮助，使用 Mybatis plus,能快速的将其需要的连接文件生成。数据库是使用的MySQL，它是一个多线程的数据库，可以以特别高的效率来处理大量的数据。

前端系统使用Vue.js作为基础的开发框架，运用了Element前端UI库作为后台管理PC端的开发。Element UI能帮助网站快速的成型，使用框架开发使用已经预制好的UI组件能够极大的减小开发者的工作量，同时该框架的配色也较友好，开发者也不用过多关系其主体颜色。前期开发，则是使用的Github上的脚手架快速搭建项目结构。

综上所述，用于编写本系统的技术架构图所示：



## 系统模块设计



## 功能详情设计

普通用户：

* 个人中心：属于用户的个性化设置
* 发布动态：用户可以在个人主页发布一些相关的信息，如视频，图片，文字等。
* 客服服务：其中包括有关的售后服务，如退货，修改收货地点，咨询聊天等。
* 登录注册：用户可以以游客、会员、普通顾客身份对其使用系统。
* 支付：使用三方程序，实现在线支付。

管理员：

系统运维：

* 资源管理：对数据的请求，响应路径进行统一管理、统一认证。
* 权限控制：对登录的角色身份进行判断处理，角色权限统一管理认证。
* 系统分析：展示系统使用情况，查看正在使用系统的用户，可进行强制上线下线，账号封禁管理。

商品销售人员：

* 商品统计：展示商品时间段内的销售情况。
* 商品审核：对即将上架或已经上架的商品进行统一管理。
* 产品分析：针对产品，展示物流情况，出货进货记录，仓库内数据。

个体商户：

* 商品管理：拥有商品上架，商品添加功能，修改商品信息，删除商品等。
* 客服服务：针对用户购物需求，处理客户需求的功能，包括退货申请，订单处理等功能。

## 数据库设计

数据库表对前端后端统一管理：

* cms\_\*：内容管理模块相关表
* oms\_\*：订单管理模块相关表
* pms\_\*：商品模块相关表
* sms\_\*：营销模块相关表
* ums\_\*：会员模块相关表

## 本章小结

本章节对系统进行了一个整体的设计，包括系统的架构，系统拥有的功能模块，各个功能模块功能相互之间的关系，以及系统开发需要用到的数据库表做了个简单的预估，为接下来的系统实现打下一个坚实的基础。

# 系统实现

## 系统开发环境

开发工具：IDEA、webstrome

JDK: 1.8

Mysql：版本为5.7+

Redis：版本3.2.100

操作系统：win10环境开发

## 主要功能模块实现

现阶段，暂时只实现了后台管理工具登录和菜单分权功能：

### 管理员登录页面功能实现：



登录采用Redis数据库进行缓存用户信息的方式实现，而Mysql表持久化特殊用户，且只开启一级缓存。

**核心代码：**

Controller层：

@ApiOperation(value = "登录")

@RequestMapping(value = "/login", method = RequestMethod.POST)

@ResponseBody

public CommonResult login(@Validated @RequestBody UmsAdminLoginParam umsAdminLoginParam) {

UmsAdmin login = adminService.login(umsAdminLoginParam.getUsername(), umsAdminLoginParam.getPassword());

if (login == null) {

return CommonResult.validateFailed("用户名或密码错误");

}

session.setAttribute(ComConstants.FLAG\_CURRENT\_USER,login);

System.out.println(session.getId());

Map<String, String> tokenMap = new HashMap<>();

return CommonResult.success(login);

}

Service层：

public UmsAdmin login(String username, String password) {

//密码需要客户端加密后传递

UmsAdmin umsAdmin=null;

try {

umsAdmin = loadUserByUsername(username);

if(!BCrypt.checkpw(password,umsAdmin.getPassword())){

Asserts.fail("密码不正确");

}

if(!userDetails.isEnabled()){

Asserts.fail("帐号已被禁用");

}

insertLoginLog(username);

} catch (Exception e) {

Asserts.fail("登录异常:"+e.getMessage());

}

return umsAdmin;

}

@Override

public void delResourceListByRole(Long roleId) {

QueryWrapper<UmsAdminRoleRelation> wrapper = new QueryWrapper<>();

wrapper.lambda().eq(UmsAdminRoleRelation::getRoleId,roleId);

List<UmsAdminRoleRelation> relationList = adminRoleRelationService.list(wrapper);

if (CollUtil.isNotEmpty(relationList)) {

String keyPrefix = REDIS\_DATABASE + ":" + REDIS\_KEY\_RESOURCE\_LIST + ":";

List<String> keys = relationList.stream().map(relation -> keyPrefix + relation.getAdminId()).collect(Collectors.toList());

redisService.del(keys);

}

}

@Override

public void delResourceListByRoleIds(List<Long> roleIds) {

QueryWrapper<UmsAdminRoleRelation> wrapper = new QueryWrapper<>();

wrapper.lambda().in(UmsAdminRoleRelation::getRoleId,roleIds);

List<UmsAdminRoleRelation> relationList = adminRoleRelationService.list(wrapper);

if (CollUtil.isNotEmpty(relationList)) {

String keyPrefix = REDIS\_DATABASE + ":" + REDIS\_KEY\_RESOURCE\_LIST + ":";

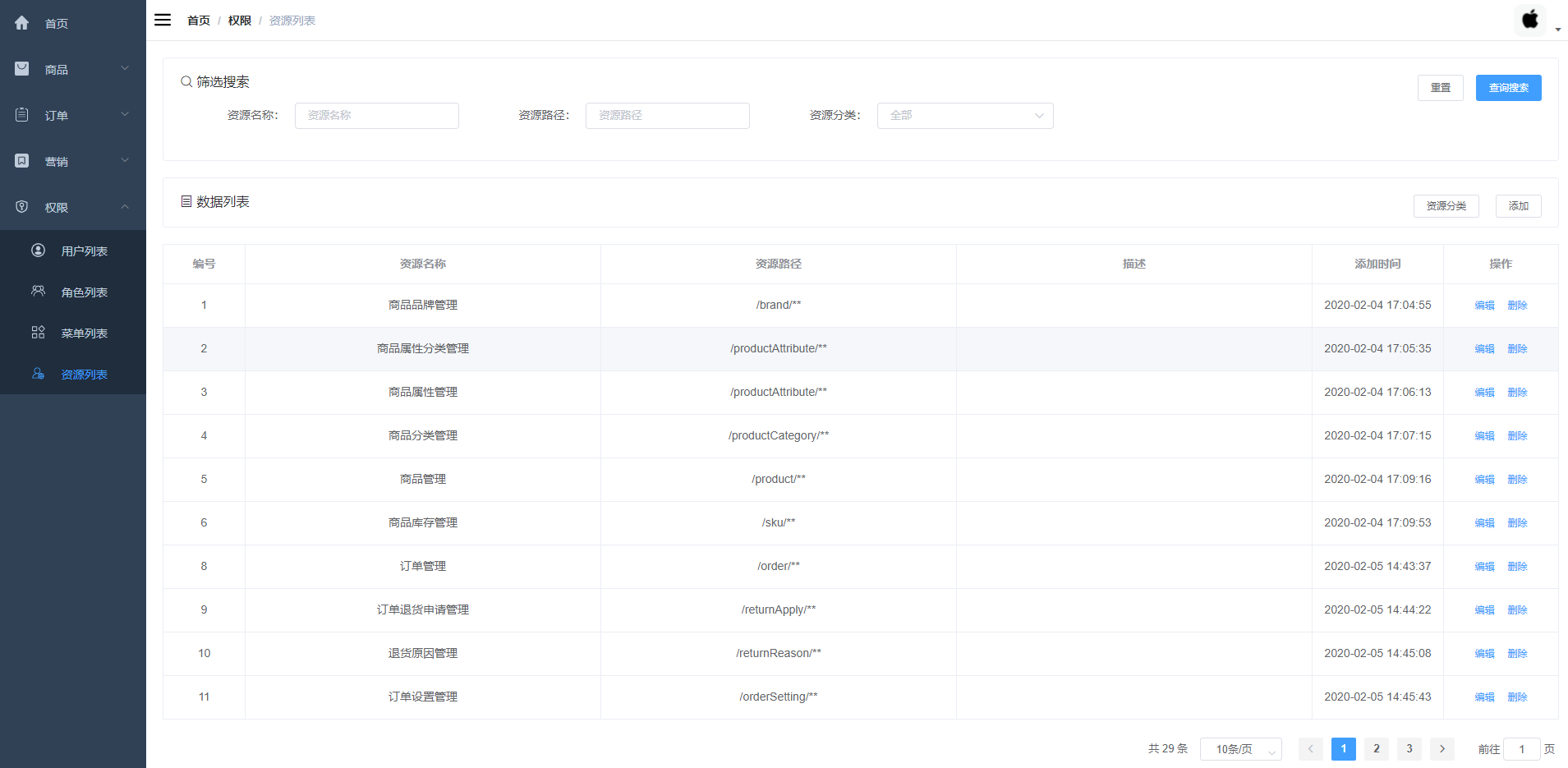
List<String> keys = relationList.stream().map(relation -> keyPrefix + relation.getAdminId()).collect(Collectors.toList());

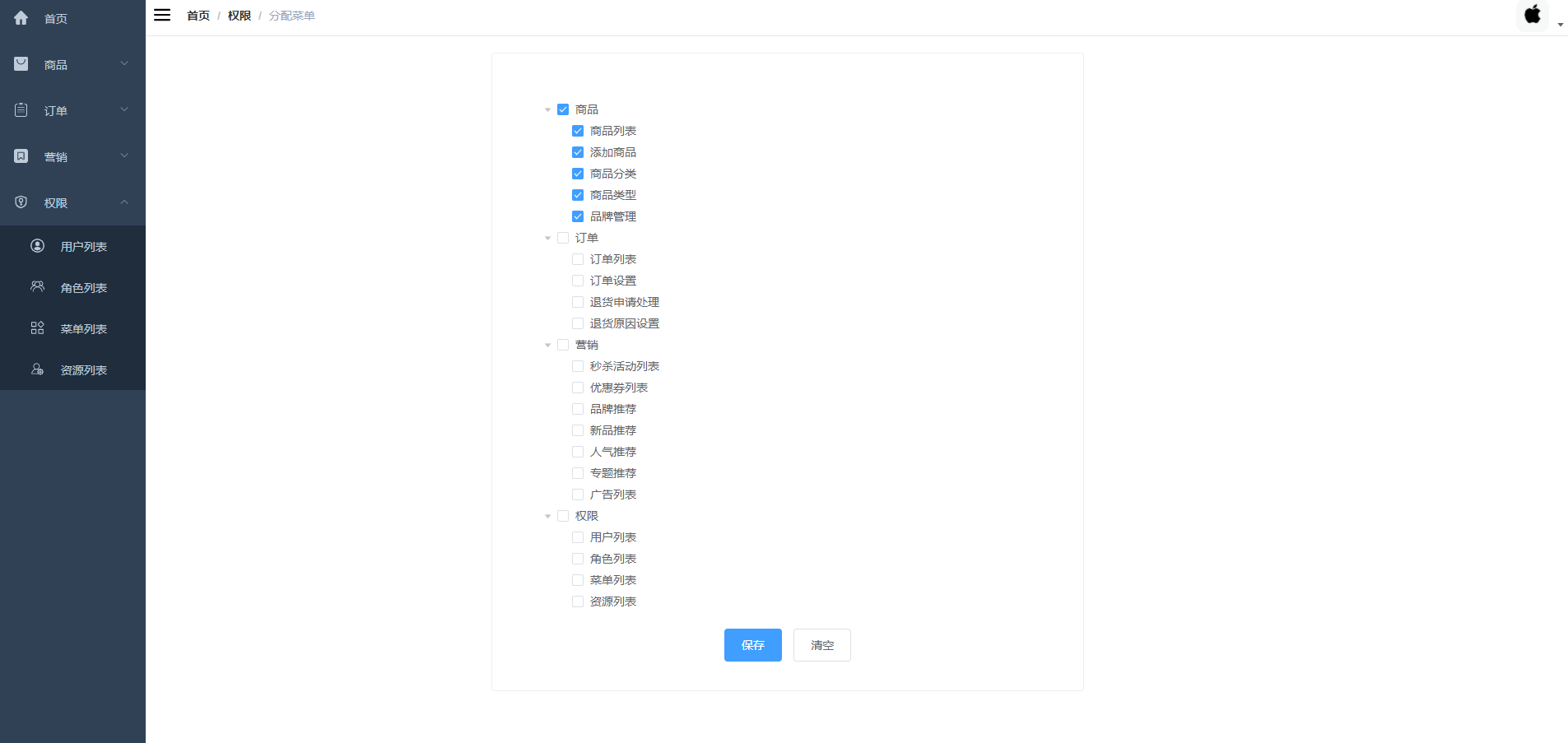
redisService.del(keys);

}

}

### 资源分配与功能菜单显示：





这里我使用了Spring Secure，在配置文件中配置了登录请求的白名单，然后登录时同时发送请求数据库查询该用户所分配的资源路径及菜单功能展示。

**核心代码：**

拦截器功能代码：

public class AuthInterceptor extends HandlerInterceptorAdapter {

// 配置文件中的白名单secure.ignored.urls

private List<String> urls;

@Autowired

private UmsAdminService umsAdminService;

@Override

public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {

//1、不需要登录就可以访问的路径——白名单

// 获取当前请求 /admin/login

String requestURI = request.getRequestURI();

// Ant方式路径匹配 /\*\* ？ \_

PathMatcher matcher = new AntPathMatcher();

for (String ignoredUrl : urls) {

if(matcher.match(ignoredUrl,requestURI)){

return true;

}

}

//2、未登录用户，直接拒绝访问

if (null == request.getSession().getAttribute(ComConstants.FLAG\_CURRENT\_USER)) {

throw new ApiException(ResultCode.UNAUTHORIZED);

} else {

//3、已登录用户，判断是否有资源访问权限

UmsAdmin umsAdmin = (UmsAdmin) request.getSession().getAttribute(ComConstants.FLAG\_CURRENT\_USER);

// 获取用户所有可访问资源

List<UmsResource> resourceList = umsAdminService.getResourceList(umsAdmin.getId());

for (UmsResource umsResource : resourceList) {

if(matcher.match( umsResource.getUrl(),requestURI)){

return true;

}

}

throw new ApiException(ResultCode.FORBIDDEN);

}

}

public List<String> getUrls() {

return urls;

}

public void setUrls(List<String> urls) {

this.urls = urls;

}

}

# 系统测试

## 测试目的和分类

对软件测试是我们开发软件中后期的一个必然过程，它能够使我们的程序在日后的使用过程中更具有趋近于完美的用户体验。

系统检测的主要目的是判断系统是否能正常运行，功能模块是否能够正常操作，代码程序是否报错。系统检测能够提高软件的质量，能够检测我们开发过程中被我们忽视的部分，能够提高软件的健壮性。

软件测试主要分为三种大类。功能性测试类、集成性测试类、性能性测试类。

功能性测试的主要内容包括:所有的系统页面的元素是否正常的显示，能否与数据库的数据对得上，页面提交的数据是否正常的传输，是否在传输工程中出现故障和数据存储是否正确，能否按照约定的格式去存储数据。

集成性测试的主要内容是:使用自定义的测试数据去检验系统的业务能力，同时检测其在运行中这些测试数据是否正确的计算，最后检测整个系统的运行流程是否符合设计，能否提高用户的体验。

性能性测试的主要内容包括:在一定数量的用户同时对系统进行操作的情况下，系统能否正常的对请求进行响应．系统的响应速度能否达到设计时的需求。

## 测试环境

测试环境是在win10电脑下，火狐浏览器测试

## 主要功能测试

登录和资源分配、菜单管理功能：

数据库预存的账户为admin/123456，资源分配为管理员的所以资源，菜单为展示所有，登录成功后，资源路径和菜单均展示正常。

### 功能性测试

登录过程：

输入错误密码，密码框下显示红色字体提示；正确输入密码，则在登录按钮上显示等待状态。查看数据库，日志表记录登录日志，但部分字符出现乱码。

### 集成性测试

登录功能集成测试：

### 性能测试

登录性能测试：

由于登录使用了Mysql和Redis，在登录过程中会等待几秒，存在延迟，当登录同一个账户，却多次输错密码，Redis会产生大量废数据，且自动持久化在Redis文件夹的.rdb文件里，使得响应速度急剧下降。

## 测试结果

# 结论

本次毕业设计的初稿，主要任务是完成了对程序编写前的设计。在本次毕业设计的过程中，从一开始的茫然到后来的各种学习、参考、查阅资料，每一步都走的极其艰难。最终在各方面的请教、指导，最终完成了初稿的编写。

通过这个商城管理系统的开发过程，我不仅仅将毕生所学的知识应用到了本次系统中，还通过查阅资料获取了许多更深层次的知识，将所学的知识融会贯通形成了自己的知识链。在这个过程中，我不仅仅学了很多专业知识，还学会了如何查阅资料，学习新知识。经过这段时间的努力，由于能力的提升也让我的自信心得到了提升，让我能更加自信的面临工作中遇到的问题，在本次设计过程中汲取的东西，是一笔宝贵的财富，让我受益匪浅。

# 致谢

# 参考文献