基于java的电商秒杀网站的设计与实现

**摘 要:**随着购物方式的多元化，网上购物成了人们越来越喜欢的消费方式，设计一种既能实现电商网站的基本购买功能，而且带有商品秒杀功能的电商网站对我国消费经济有极大的促进作用。

本文通过对当前市场的一些电子商务网站如淘宝、当当等网站的研究，分析出了电子商务发展的趋势，随着快递物流的不断增加，未来电子商务必然会有更大的市场和前景。本文首先完成了电商系统的总体框架的设计，然后详细的进行需求分析，明确和完善了各个模块需要完成的功能，最后完成各个模块的功能实现。

此电商秒杀系统主要是从消费者角度出发，对登录、注册，再到商品详情页的展示，商品的秒杀等功能进行了深度的研究。此电商秒杀系统采用Java语言开发和MySQL数据库，开发工具使用IntelliJ IDEA。使用MVC三层架构为设计模式为系统进行分层设计。在界面和数据的交换上采用Servlet和JSP以及Ajax异步请求来完成数据和页面的展示。此系统实现了电商系统的简易功能，模拟用户购物，为网上购物提供便利。

关键词：电子商务; Java; MySQL数据库; JSP

Java-based electricity kill website design and implementation

Abstract:Along with our country economy development, online shopping become more and more like spending patterns, and with the function of commodity seconds kill e-commerce sites of great role in promoting consumer economy of our country.

In this paper, through the study of the current market of some e-commerce site, analyze the current situation of electronic commerce and the trend of future development, with the continuous increase of express logistics, the future of e-commerce will have a larger market and prospects.And then the detailed demand analysis, clear and perfect the function of each module needs to be completed, the final realization of the function of the various modules.

This electrical contractor seconds kill system is mainly from the perspective of consumers, to login, register, and then to the commodity details page display, kill, and the order of payment, and other functions to carry on the depth of research.This electrical contractor seconds kill system USES Java language and MySQL database development, development tools using IntelliJ IDEA. MVC design pattern is used to design the system hierarchically. In the interface and data exchange, servlet, JSP and Ajax asynchronous request are used to complete the display of data and pages. This system realizes the simple function of e-commerce system, mainly from the perspective of consumers, to provide convenience for online shopping.

**Keywords:** Electronic Commerce；Java; MySQL; JSP

# 第1章 绪 论

## 1.1 课题背景与意义

随着互联网的普及，越来越多的人们喜爱在互联网上查阅资料，享受生活。互联网时代已经悄然来临。且随着世界经济的迅猛发展，电子商务作为经济发展的前提下催生的产物。中国进入改革开放后经济开始腾飞，电子商务的蓬勃发展同时也推动了互联网的发展。互联网和电子商务相辅相成，相互促进，以至于它们迈入了一个腾飞的阶段。

步入二十一世纪以来，在计算机技术迅猛发展的同时， 我国经济也不断发展，国民的消费观念慢慢发生了转变，从老一辈的缝缝补补，到现在已经很少有用坏去修的思想，一般都是去置办新的产品，而且对一些新型产品的追求日益加剧。类似电脑、手机这类电子产品，大概在半年到一年就会更新换代，一些实体旗舰店还没有收到公司的产品来货，有的消费者就需求买入，而实体店面一般库存都是一些老一代的产品，更加不能满足当代年轻人的挑剔要求。这个时候就能突出电子商务的重要性和意义了。第一，你可以不用出家门就能买到最新的产品；第二，你可以根据自己的喜好去选择商品，购买不存在局限性了，不用听推销员的啰嗦了更不用碍于面子购买自己不喜欢的产品了；第三，产品的售后有保障。总的来说，电子商务相较于实体购物是非常有优势的。

## 1.2 电商秒杀系统的发展前景和展望

纵观互联网发展到今天，网络消费观念将会成为一种必然，电子商务是一个全新的、不断变化的领域，随着新技术的飞跃发展，也益于Internet技术的不断发展。我国的电子商务从马云创建阿里巴巴网站，到阿里成交的第一笔订单，到2020年双十一阿里实现了4982亿元的成交额。随着电子商务的不断发展，后续出现的一些列电商企业比如：京东、当当、国美、苏宁等。都获得了不错的发展，给了我们良好的网上购物体验，也为我们提供了便利。国家越来越提及”互联网+”的战略，这一定程度上也促进了电子商务的发展，也促进了一些电商企业的成长，我觉得现在的电商企业正朝着以下趋势发展：

1、从价格战到提高服务质量

在电商企业发展之初，为了吸引用户量，也为了彼此间的竞争，他们经常采用以极低的价格去获取用户，降低利润来获取高订单量。以此来达到与同行竞争的目的。随着电商企业的逐渐发展，很多电商企业逐渐改变了观念，不再一味的去压低价格，而是想办法给消费者提供更好的服务，比如京东成立的自营店铺，虽价格稍贵，但是产品质量

有保障，假一赔十；为了提高用户体验，他们更是采用京东物流，让消费者更快收到自己所购买的商品。

2、线上线下同时扩展影响力

电商企业经过稳定的发展，市场份额逐渐稳定下来，越来越多的电商企业家明白，要想进一步扩大自己的影响力，那么需要线上线下同时扩展，而且所销售的产品应该向综合类去改变，所以才有了京东便利店、京东大药房、京东母婴、盒马生鲜、当当实体书店等等。一系列的线下店铺应运而生。

支撑电商企业发展的，就是他们的一整套电商系统，没有这个系统或是系统出现问题，他们的很多业务都将受到影响，从而导致消费者的消费体验不佳。那么就会流失掉一些客户，这是企业所不想看到的。所以，设计和完善电商系统是非常重要的，它支撑着企业的发展，未来也必不能少了这个系统。

## 1.3 课题的目标和内容

本课题的研究主要目标是实现一个电商秒杀网站，让用户能够实现从登录、注册，到选购商品，再到支付的一整套流程。电商秒杀系统的主要功能有登录注册、商品分类、商品搜索、商品秒杀、购物车和支付宝支付等。

本文的研究内容主要如下：

1、电商秒杀网站的需求分析

2、电商秒杀系统的系统可行性的分析，以及数据库的设计

3、完成电商秒杀系统的各个模块的功能实现

4、对各个模块的功能进行测试

## 1.4 论文的组织结构

第一章，绪论。介绍了对电商秒杀系统的研究背景和意义、以及未来的一些发展趋势。表明了课题的目标的内容。

第二章，开发相关的工具以及技术介绍。阐述了在开发过程所用到的一些开发软件和开发技术。并说明软件和技术分别有哪些优缺点，以及个人的一些看法。

第三章，系统的需求分析以及可行性分析，通过各个方面对电商秒杀系统的可行性做出了分析，确保系统能顺利完成。以及对系统的需求做了详细的分析，阐明使用了模块化、分层化的思想进行划分。

第四章，系统的总体设计。在需求分析的基础上，阐明了系统的开发环境，更加详细的说明了系统每个功能模块的设计，并且讲述了数据库E-R图和数据库表的设计。

第五章，系统的实现。通过对功能模块的介绍后，绘制功能流程图，再使用代码将功能具体的实现。

第六章，系统的测试。使用不同的测试用例对系统的功能进行全方位的测试，保证系统的顺利运行。

第七章，总结与展望。总结电商秒杀系统的研究过程以及研究成果，对系统研究过程中出现的问题以及不足之处做出说明。

# 第2章 开发工具及相关技术

## 2.1 常用开发工具

### 2.1.1 IntelliJ IDEA简介

IDEA 全称 IntelliJ IDEA，是java编程语言开发的集成环境。是大多数java程序员喜爱的java开发工具，其功能丰富，有代码自动提示，英文自动检查，项目重构方便，支持各类版本控制工具包括Git和SVN，有丰富的快捷键令开发效率倍增。还支持多语言环境，包括前端的HTML、CSS、JavaScript。包括一些配置文件比如properties、yaml等。其丰富的插件功能，使得开发更加便捷比如自带的数据库连接工具、阿里巴巴代码规范Alibaba Java Coding Guidelines、代码右侧缩略图工具CodeGlance，都会给我们比较好的开发体验。更值得一提的是它的UI界面，非常的简约，在写代码的时候给人一种毫无压力的感觉。

### 2.1.2 Navicat Premium

Navicat premium 是一款数据库管理工具。它可以通过连接来获取本地的数据库，通过可视化界面来显示出来。这样避免了我们经常在命令行使用命令来操作数据库。Navicat premium也有自身的优势，比如你可以从中看到各种数据库的详细信息、包括报错等等，你也可以使用Navicat premium进行创建数据库、创建表等一些列操作，都是可以使用界面操作。这样操作会变得更加简单。Navicat premium还有一个优势在于它能支持多种数据库，包括MySQL、SQLite、Oracle 及 PostgreSQL，而且可以单一程式同时连接，让不同数据库之间的管理更加方便。

## 2.2 相关技术

### 2.2.1 Java语言

Java是一门[面向对象](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1" \t "https://baike.baidu.com/item/Java/_blank)编程语言，面向对象的语言的特点包括封装、继承、多态。这种面向对象导致了Java语言的代码层次感很强。

Java语言的用途也是非常广泛。Java可以通过GUI编写桌面程序、和前端结合可以做各种网站的后台系统等。

通过对很多电商系统的研究，发现淘宝、京东等知名电商企业的电商系统都是使用Java语言来开发，更加说明Java语言对电商系统的契合程度非常高。

### 2.2.2 MySQL数据库

MySQL是一个[关系型数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511" \t "https://baike.baidu.com/item/mySQL/_blank)，具有速度快并且灵活性高的特点，缺点就是容易出现数据冗余的情况。MySQL是开源免费的，一般中小型网站的开发都选择 MySQL 作为网站的数据管理，用作网站数据的持久化，这样开发效率很快而且对小型网站来说非常的经济，但是安全性有待加强。

### 2.2.3 JSP

JSP是Java的服务器页面。JSP部署于网络服务器上，可以响应客户端发送的请求，JSP的主要作用是代替Servlet程序回传HTML页面的数据。

JSP还有内置九大对象，用来发起请求和相应请求，和后端进行数据交换。JSP还有四大域对象，四个域对象功能一样，只是对数据的存取范围不同。包括全局的application域、会话管理的session域、当前请求request域、pageContext域。主要作用是用来存储服务器后端传递到前端来的数据。application域在整个web工程内都有效；session域在一个会话范围内有效；request域在一次请求内有效；pageContext域在同一个JSP页面范围内有效。JSP技术中最常用的还有C标签和EL表达式。C标签又叫核心标签，我们可以在C标签中嵌入一些我们想要书写的Java代码。EL表达式是在JSP中用于做判断的表达式比如做空值判断、逻辑判断、算术判断都可以使用EL表达式来实现。

总的来说JSP的功能是非常强大的，只要我们合理的运用JSP中的一些对象和域，可以非常方便的实现前后端的数据交换，对我们的开发效率有极大的提升。

### 2.2.4 Servlet

Servlet（Server Applet），全称Java Servlet，Servlet运行于支持Java的应用服务器中。在编写实现Servlet的Servlet类时，直接实现它。在扩展实现这个这个接口的类时，间接实现它。我们使用最多的还是HttpServlet，因为需要使用HttpServletRequest和HttpServletResponse对象来获取前端传给后端的数据。这样才方便后端进行数据的处理。

### 2.2.5 JQuery框架

随着互联网的不断发展，人们都希望用更少的代码实现更多的功能，所以一些优秀的程序员把一些基础的代码进行了封装。JQuery就是这样应运而生。JQuery设计的宗旨是倡导写更少的代码，做更多的事情。JQuery兼容各种主流浏览器。这个是JQuery框架的优势。

JQuery在学习过程中会让你感觉非常舒服，因为JQuery的指导文档写的十分清晰，JQuery仅仅对底层的Javascript代码进行了简单的封装，所以可读性很高，而且JQuery的文件大小仅仅在20K左右，非常轻量级便于我们在项目中引入。不管是刚入行的小白还是非常老练的程序员都喜欢使用JQuery框架进行前端页面的一些动态操作。

### 2.2.6 Bootstrap框架

众所周知的是，一个项目从0到1完完全全使用一些底层的代码实现是非常困难的。这样不仅仅开发效率低下，而且尤其对于后端程序员来说要他们去写一些CSS的样式简直就是非常痛苦的。所以需要一种对于叠层样式表进行封装的框架，程序员在使用的时候仅仅需要调用，就可以写出非常漂亮的样式，这样对于我们的项目开发非常有帮助。

BootStrap就是一个基于html、css、Javascript的前端框架，相当于说它是个半成品，开发人员可以在BootStrap的基础上进行开发，简化我们的代码，以至于减少工作量。

BootStrap也就特点，第一个特点是它定义了许多的CSS样式和Javascript插件，开发者可以直接引用这些样式和插件，比如一些按钮、图片、表格、表单等等。第二个特点是BootStrap使用了响应式布局。这样移动端和PC端就可以使用同一套前端了。减少了开发人员的工作量。

### 2.2.7 Ajax技术

Ajax全程叫Asynchronous JavaScript and XML，是一种异步请求，不是一种新的编程语言，而是一种用于创建更好更快以及交互性更强的Web应用程序的技术。我们常使用Ajax是因为它可以异步的向后端发起请求，再不刷新整个网页的前提下能够动态的修改网页上的内容，极大的提高了用户体验。

### 2.2.8 本章小节

本章主要介绍了电商秒杀系统所用到的一些开发工具和使用到的框架以及一些技术的介绍，以及这些技术的用途。对Java语言、MySQL数据库、JSP、Servlet、JQuery、Bootstrap、Ajax都进行了相关介绍，还对Intellij IDEA和Navicat一些常用的开发工具进行了简介和我一自己的一些看法。

# 第3章 系统可行性分析与需求分析

## 3.1系统的可行性分析

### 3.1.1经济可行性

通过对实体店铺和电子商务的研究，我发现对于实体店铺而言，经济性是个值得思考的问题，因为你开个实体店铺，店铺租金就是一笔不小的开销，而且你还需要招聘人员去看管店铺和服务顾客，在经济和人力上面都是很大的消耗。而对于电子商务来说，仅仅需要开发人员去开发系统，后期派人维护，商家入驻也仅是把自己的商品添加到数据库中，这样可以大大减少相比实体店铺中一些不必要的开支。所以说电商秒杀系统在经济上是可行的。

### 3.1.2市场可行性

前文有说明在互联网不断发展的今天，已经有许许多多的成功电商企业的案例，阿里巴巴集团从创立到今天据港交所显示市值已经约4.45万亿港元；京东集团从创立到今天据港交所显示市值已经约8838.54亿港元亿港元。而且越来越多的企业都开始打造自己的电子商务，乃至于街边的奶茶店都可以使用线上点单，线上销售的模式已经是越来越受欢迎。当代年轻人酷爱线上消费的模式，况且根据研究表明年轻人在国民消费的比重不断上升，又因为6.18以及双11等等大型购物节的推动，当前的市场是非常广阔的，如果你有好的产品，加上一定的广告营销是不愁卖不出去的。所以企业搭建自己的电商系统在市场是非常可行的。

### 3.1.3时间可行性

电商秒杀系统的开发周期有3个月，主要我在大二大三的时候就已经开始学习Java语言了，现在已经掌握了电商秒杀系统开发所需要的一些技术，我准备在第一个月完成电商秒杀系统前端的设计，然后再花一周的时间数据库的分析与设计，最后再开发电商系统的后台。因为有足够的开发周期，所以我有信心去完成这个毕业设计。

### 3.1.4技术可行性

随着Java的火热，大量的Java教学视频在网上涌现出来，还有一些大牛写的博客、技术总结等等都让我觉得使用Java语言来实现一个电商秒杀系统是可行的，从登录到商品的浏览到下单支付的过程是可以是实现的。而且网上有些博客非常全面的总结了开发一个电商秒杀系统中会遇到的一些难题，这也让我受益良多。我使用核心技术的是JSP加上Servlet，对于前端的一些提示是使用Javascript做出的，在前后端数据交互我使用了Form表单提交、转发以及Ajax异步请求等，前端不需要提示的我就使用转发，需要提示的我就使用Ajax，目的明确，极大的提高了开发效率。电商秒杀系统开发采用的是Windows10系统，服务器使用的是Tomcat，数据库采用的是关系型数据库MySQL，使用Intellij IDEA作为代码编译器，因此电商秒杀系统是切实可行的。

## 3.2 系统功能需求分析

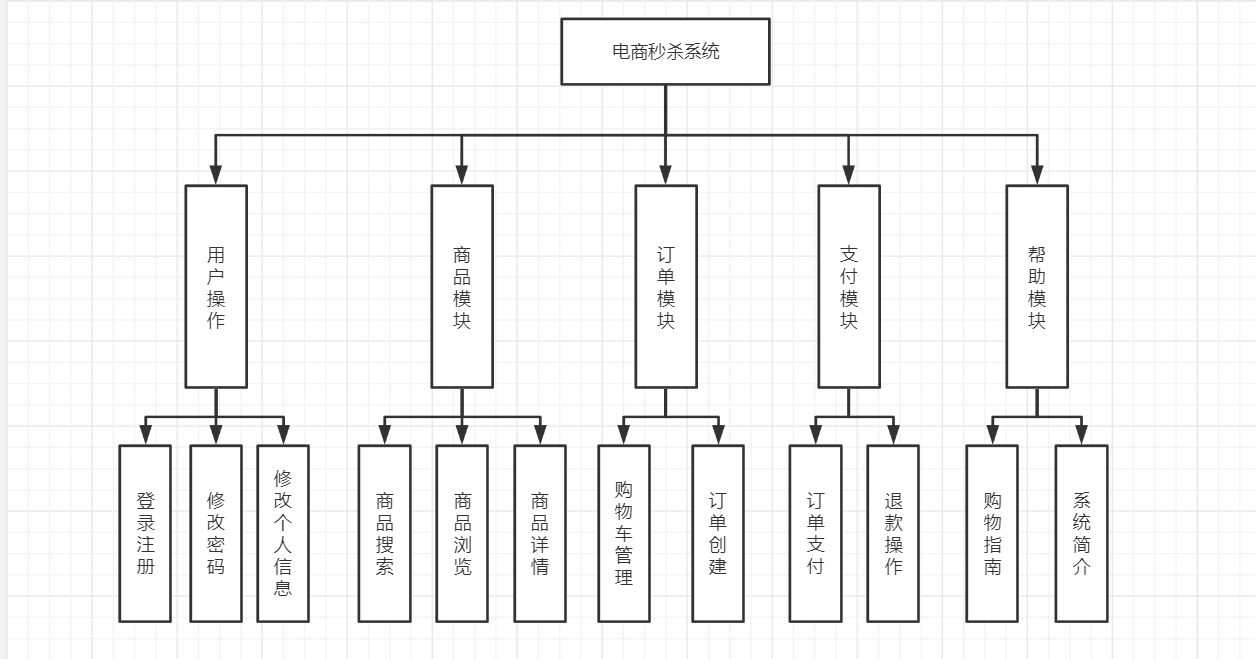
本系统的目的是为了给企业搭建一个电商秒杀系统，为了让企业的产品可以多渠道的销售，也让消费者更加方便的可以买到符合心意的产品。电商系统是一个非常庞大的系统，所以我在设计的时候采用了分层开发和分模块开发，将一个庞大的系统分为几个小的子系统，然后将子系统中的模块又进行详细的分解，再确定每个模块具体的功能，这样我们每个功能的任务就非常明确清楚了。因为我们系统采用的是MVC三层架构，所以在分层开发的过程中，我们可以先写Model层的代码，再上升到Controller层，最后再去写View层的代码，这样会让我们在写代码的时候逻辑非常清晰，减少代码出Bug的情况，而且分层开发也有一下优势：

1、隐藏细节，提供抽象，隐藏的细节包括数据的表示、功能的实现

2、接口清晰，修改维护方便

3、方便开发，将软件分为若干层次，逐层实现。

经过我们的需求分析，我们总结出电商秒杀系统的整体模块如图3.1所示。

图3.1 整体模块图

接下我对电商秒杀的系统的所有模块进行一个简介

1、用户模块

用户模块主要是包括对用户的一些操作，有用户的登录功能、注册功能、以及修改个人信息包括修改密码、修改收货地址等。

2、商品模块

商品模块的主要功能有商品推荐、商品的展示、商品浏览、商品的分类、商品的搜索、商品的秒杀等功能。商品的浏览又分为商品的分级浏览以及秒杀商品的浏览；商品的搜索是根据关键词进行全局的检索再定位到具体的商品；商品的秒杀功能是如果存在秒杀活动，就可以去以秒杀活动的价格去购买指定的商品。

3、订单模块

订单模块主要是订单的创建、订单的删除、订单的支付、以及购物车的管理，将某一商品添加到购物车，从购物车创建订单等。

4、支付模块

支付模块最主要的功能就是订单的支付，支付采用的支付宝的沙箱支付模型，通过沙箱模型来模拟支付场景，还有一个退款操作，如果消费者对商品不满意，可以使用退款操作。

5、帮助模块

帮助模块主要的功能是张贴一些文章，比如购物指南、售后如何操作、电商系统的简介、消费者购买的流程等等。

## 3.3 性能需求分析

对于任何一款系统来说，性能测试都是必不可少的一部分。设想一个情景，你所开发电商秒杀系统，消费者在使用的过程中，访问一个具体商品详情页，需要5秒以上的时间，在这等待的过程中，消费者就可能会回心转意，不在你这个平台内购买了。这在一定程度上会严重影响消费者的体验，而且消费者对于你这个系统有非常差的评价，导致在心中不信任你这个平台，从而因为性能的影响流失了一些客户，这是得不偿失的。所有我们在设计系统的时候尤其注意的是系统的性能。

1、安全性分析

从安全分析来说，系统最重要的就是数据库的安全问题，所有我们在设计代码的时候一定要尤其注意防止SQL注入，还有各种窃取我们数据库的方式。我在编写代码的时候采用的是预编译的SQL，可以有效的防止SQL注入问题，我们的第二重保障是对用户密码的加密功能，存储在数据库的密码是加密后的，如果不慎被黑客破解了数据库，那么他也无法获取我们的用户信息，极大的保障的用户的安全。对于我们这种面向消费者的系统来说，消费者的信息安全是尤为重要的。其次容易出现密码泄露的地方在于登录过程，因为客户端和服务器的交互请求时无状态的，用户每次登录都会带上身份信息，用户登录成功后，会将身份信息返回给客户端，一些心怀不轨的人就可以去劫持在客户端上的身份信息，然后获取到用户的密码，对于这样的情况我们可以使用HTTPS协议来代替HTTP协议，但是HTTPS协议是要收费的，不适于我们此次毕业设计，还有一种办法就是使用Session来保存用户的登录信息，因为Session是存在于服务器端的，安全性比较高。

2、数据完整性分析

防止数据库完整性被破坏，我们可以从下面几点来防范。第一个可以防范的点是在我们的数据库设计中，许多字段我们可以设置为非空，这样就可以防止空数据进入我们的数据库，从而保障了数据的完整性。第二个点是在我们的Controller层的代码中，我们可以进行一些非法数据的判断，如果前端传过来的数据是非法的，那么我们就将其舍弃并告诉前端数据非法，这样一些非法数据也不会进入我们的数据库里面。对于一个系统来说，数据是系统中尤为重要的信息，如果一个系统连数据的安全性都不能保证的话，那么这个系统必然是失败的，是不会让用户满意的。

3、可拓展性和可维护性分析

在系统的开发过程中，我们的需求经常在设计完以后，出现一些其他的改动，比如增加一些新鲜的功能或者是删掉原有性能比较低下的功能，这就对于我们的系统的可扩展性有了一定的要求，在我们架构系统设计的时候应该要考虑这方面的因素，而且在编码的过程中应用面向对象的思想，封装一些具体实现代码，以至于实现代码的复用，这样可以降低代码的耦合性，模块之间的代码互不影响，这样系统的可扩展性就变高了。

4、界面要求

界面对一个用户而言是最直观的，因为用户使用你这个系统第一眼就是看到系统的界面，界面的要求就是美观、简约。要让用户看了你的系统界面觉得舒适，而不是产生没有访问下去的欲望的感觉，简约就是说用户可以通过简单的观察，就可以发现你的界面的功能，而不是通过半天的点击达不到用户所想要的结果。如果一个界面符合了美观加上简约两个要求，那么就是一个合格的界面。

5、并发性分析

对于一个电商秒杀系统而言，在开发完成以后必然是要给许多用户去使用的。系统的并发性是衡量一个系统非常重要的指标。并发性指的是一个系统在使用的过程中最高能承受多少的访问量。如果某一时间超过你系统所能承载的并发，会出现访问不到或者是访问页面崩溃等一些列情况。我在设计系统的时候，采用了分层架构，这是有利于提高系统的并发的，而且我们在写代码的时候尽量减少磁盘的IO操作，与数据库连接的时候采用数据库连接池进行水平扩展，这样也有利于提高我们系统的并发量。

# 第4 章 系统设计

## 4.1系统开发环境搭建

系统开发环境如下：

1、硬件环境

电商秒杀系统部署在本地服务器上，服务器的硬件配置越好，电商秒杀系统就越可以承载越多的负载，从而来支持业务逻辑。本地服务器的硬件配置如下：

CPU：Intel Corei7

内存：8.00GB

硬盘：500GB

2、软件环境

电商秒杀系统的开发工具是Intellij IDEA，IDEA是当今最为流行的开发JavaEE的最为便利、效率最高的开发工具，其简约的界面风格，我非常喜欢。同时支持JSP语法，前后端一体化开发，效率非常高，再也不用再几个软件中切换。

操作系统：Win10专业版64位

开发工具：Intellij IDEA

服务器：Tomcat 9.0

数据库：MySQL 5.5

数据库管理工具：Navicat

开发语言：Java

JDK：jdk1.8

页面使用JSP展示，采用前后端不分离开发。其中很多页面使用了HTML5、CSS3、Javascript、JQuery、Bootstrap。浏览器采用的是最新的谷歌浏览器。是因为谷歌浏览器的兼容性非常好，而且运行速度很快。

## 4.2系统功能模块设计

根据上文的需求分析、可行性分析、性能分析可知，我们要怎么对系统的功能模块进行一个划分。根据图3.1可知，电商秒杀系统的整体模块划分。我将根据电商秒杀系统的整体模块图，再仔细说明一下具体功能的操作步骤。

1、用户模块

用户也就是我们的买家，可以不用注册账号，就可以进入我们系统的主页进行一些商品的浏览和查看。但是如果想将商品收藏或者是下单购买商品，系统就会判断当前用户是否登录，如果没有登录就跳转到登录页面，进行注册了登录，才可以使用商城的一些功能。在登录进系统以后，我们可以去使用用户模块的所有功能。

2、商品模块

商品模块里面的功能有商品的展示，在用户访问主页的时候，我们会将一些商品呈现给用户，按照商品的一、二级分类展示给用户，我们再主页还做了轮播图处理，对新型产品进行一个推广，顶部还有一个搜索框，我们可以按照关键字进行检索得到对应的商品信息，还有一个热搜产品推荐、以及秒杀中心，开展秒杀活动以较低的价格购买到心意的商品。

3、购物车模块

购物车是购买流程中必不可少的部分，因为你所需要的商品有时候可能不止一件，你可以先都收藏到购物车中。然后再去对购物车里面的商品进行查看、删除某件不喜欢的商品、修改商品的购买数量、清空购物车等等。确定好以后再提交给订单模块去生成订单。

4、订单模块

在生成订单的时候，会要用户选择收货人姓名和收获地址，再确定所要购买的商品以及数量是否正确，如果无误，则提交订单，订单生成，订单号是按照时间的毫秒数来生成，还有订单的时间，订单的金额，订单的状态等等都会展示。

5、支付模块

根据订单所要支付的金额，跳转到支付页面，根据支付宝沙箱模型，进行模拟支付，支付成功跳转到主页面，如果15分钟内未支付成功，则会自动删除未支付的订单并将库存返回。还附带一个退款操作，用户提交退款申请，管理员通过以后，金额会通过支付宝沙箱模原路返回到用户的账户中。可以使用一张图来诠释支付和退款的一个过程。

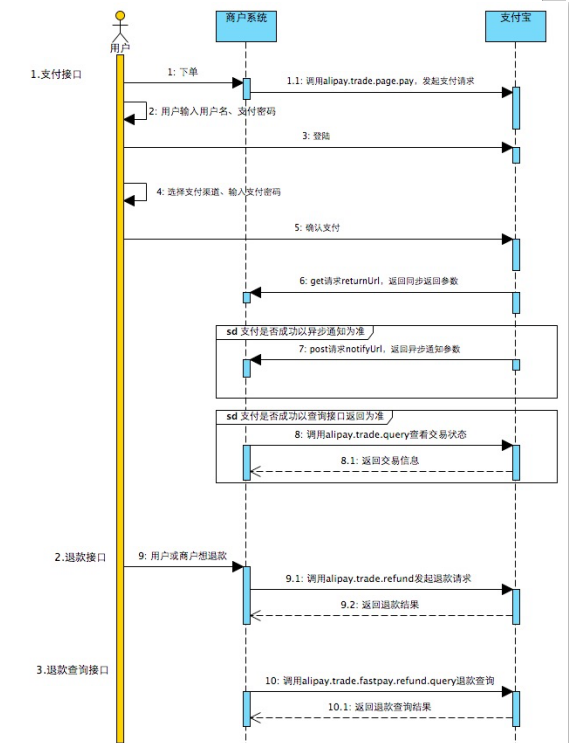


图4.1 支付、退款流程图

6、帮助模块

在帮助模块中，主要是展示了一些文章内容，为了让用户更加方便的去使用系统，在帮助模块内我们编辑了购物指南、售后服务介绍、商城简介等等文章，希望提高用户的购物体验。

## 4.3数据库模块设计

### 4.3.1数据库设计概述

数据库设计是每一个系统必不可少的一部分，也是尤为重要的一部分，因为数据库的设计一旦确定后期要修改起来的难度就非常高，因为数据库对应着我们的实体类，牵一发而动全身。在数据库设计的时候我们要注意基本表与其字段之间的关系要满足数据库设计的三范式、满足数据库设计的一对一原则、双向使用原则等。在设计的时候，我们还应该注意，第一明确系统需求；第二，重视数据的维护；第三，增强命名规范；第四，合理的使用索引。以上都是在设计数据库时应该注意的问题，合理的解决这些问题，设计出来的系统会非常高效。

### 4.3.2数据库设计步骤

1、在进行数据库设计之前，设计人员必须充分了解系统的功能需求

2、了解需求分析后，定义出实体以及定义实体的属性

3、绘制实体与实体之间的关系图E-R图

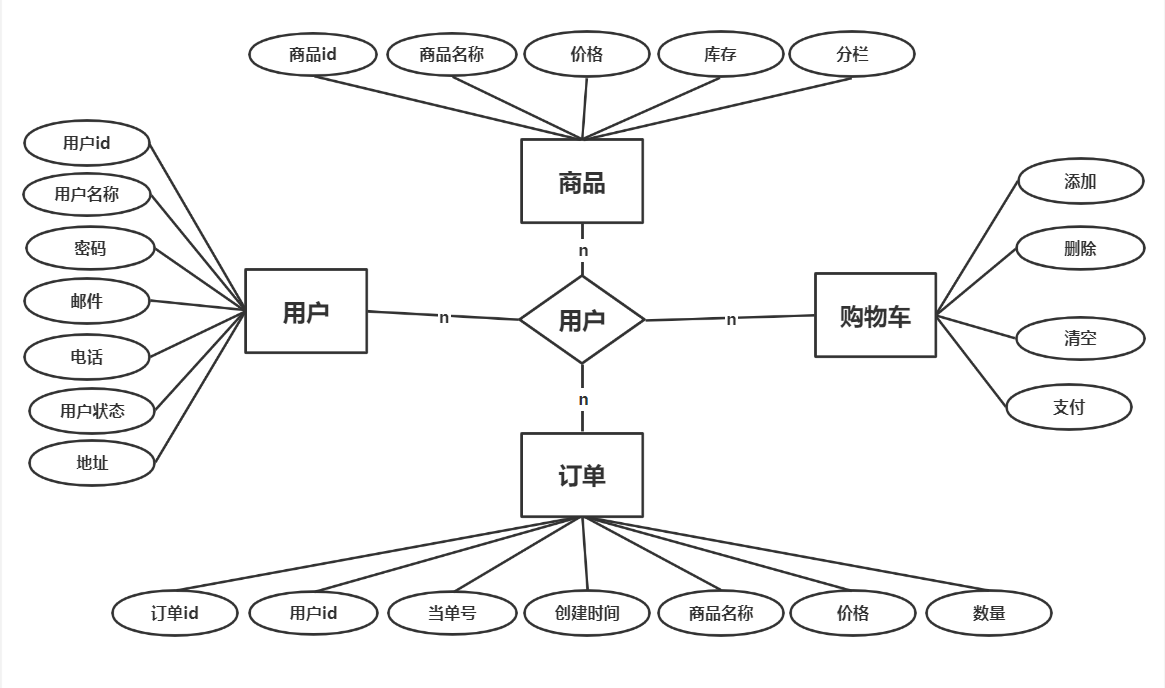
4、将E-R图转化成物理模型

5、检查物理模型是否符合数据库三范式

6、根据模型定义数据库

### 4.3.3 ER图设计

数据库关系图就是ER图，ER图提供了表示实体、属性和联系的方法，用来描述现实世界的概念模型。简单来说就是将抽象的模型转换成可以表示的图形。

图4-2 用户E-R图

### 4.3.4基本表设计

数据库设计中最为基础最为重要的缓解就是基本表的设计了，因为一切数据的操作都需要依靠表和字段来进行操作。

1、用户表的基本信息

表4.1 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 键 | 注释 |
| id | int | 11 | 非空 | 主键 | 用户id |
| username | varchar | 255 | 非空 |  | 用户名 |
| password | varchar | 255 | 非空 |  | 用户密码 |
| nick\_name | varchar | 255 |  |  | 昵称 |
| postcode | int | 11 |  |  | 邮政编码 |
| real\_name | varchar | 255 | 非空 |  | 真实姓名 |
| phone\_number | varchar | 255 | 非空 |  | 电话号码 |
| id\_card | varchar | 255 | 非空 |  | 身份证号码 |
| address | varchar | 255 | 非空 |  | 地址 |

2、商品表的基本信息

表4.2 商品表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 键 | 注释 |
| id | int | 11 | 非空 | 主键 | 商品id |
| name | varchar | 255 | 非空 |  | 商品名称 |
| price | double | 10 | 非空 |  | 商品价格 |
| discount | double | 10 | 非空 |  | 商品折扣 |
| properties | varchar | 255 | 非空 |  | 商品属性 |
| description | varchar | 255 |  |  | 商品描述 |
| stock | int | 11 | 非空 |  | 商品库存 |
| goods\_source | varchar | 255 | 非空 |  | 商品货源 |

3、订单表的基本信息

表4.3 订单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 键 | 注释 |
| id | int | 11 | 非空 | 主键 | 订单id |
| number | varchar | 255 | 非空 |  | 订单编号 |
| state | varchar | 255 | 非空 |  | 订单状态 |
| price | double | 10 | 非空 |  | 订单价格 |
| count | int | 11 | 非空 |  | 商品数量 |
| order\_date | varchar | 255 | 非空 |  | 订单日期 |
| good\_id | int | 11 | 非空 |  | 商品id |
| user\_id | int | 11 | 非空 |  | 用户id |
| order\_info | varchar | 255 |  |  | 订单详情 |
| address\_id | int | 11 | 非空 |  | 收获地址id |

4、秒杀活动表的基本信息

表4.4 秒杀活动表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 键 | 注释 |
| id | int | 11 | 非空 | 主键 | 秒杀活动id |
| promo\_name | varchar | 255 | 非空 |  | 秒杀活动名称 |
| start\_time | datetime | 0 | 非空 |  | 开始时间 |
| good\_id | int | 11 | 非空 |  | 商品id |
| promo\_price | double | 10 | 非空 |  | 秒杀价格 |
| end\_time | datetime | 0 | 非空 |  | 结束时间 |

1. 收货地址表的基本信息

表4.5 收货地址表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 键 | 注释 |
| id | int | 11 | 非空 | 主键 | 收货地址id |
| user\_id | int | 11 | 非空 |  | 用户id |
| rev\_name | varchar | 255 | 非空 |  | 收货人姓名 |
| address | varchar | 255 | 非空 |  | 区域地址 |
| detail\_address | varchar | 255 | 非空 |  | 详细地址 |
| postcode | int | 11 | 非空 |  | 邮政编码 |
| phone\_number | varchar | 255 | 非空 |  | 电话号码 |

1. 购物车表的基本信息

表4.6 购物车表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 键 | 注释 |
| id | int | 11 | 非空 | 主键 | 序列id |
| good\_id | int | 11 | 非空 |  | 商品id |
| price | double | 10 | 非空 |  | 商品单价 |
| total\_price | double | 10 | 非空 |  | 选中总价 |
| count | int | 11 | 非空 |  | 商品数量 |
| user\_id | int | 11 | 非空 |  | 用户id |
| is\_buy | int | 11 | 非空 |  | 是否选中 |

1. 商品图片表的基本信息

表4.7 商品图片表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 键 | 注释 |
| id | int | 11 | 非空 | 主键 | 序列id |
| good\_id | int | 11 | 非空 |  | 商品id |
| address | varchar | 255 | 非空 |  | 商品图片地址 |

1. 退款表的基本信息

表4.8 退款表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 不是null | 键 | 注释 |
| id | int | 11 | 非空 | 主键 | 退款id |
| user\_id | int | 11 | 非空 |  | 用户id |
| number | varchar | 255 | 非空 |  | 订单编号 |
| price | double | 10 | 非空 |  | 订单价格 |
| reason | varchar | 25 | 非空 |  | 退款原因 |
| state | varchar | 255 | 非空 |  | 退款状态 |
| time | datetime |  | 非空 |  | 退款时间 |
| flag | varchar | 255 | 非空 |  | 退款标志 |

# 第5章 系统实现

本系统主要包括用户的基本操作、商品模块、购物车模块、订单模块、支付模块、帮助模块构成。 我将在第5章中将以上模块具体实现的详细的进行说明。

## 5.1 用户模块的实现

### 5.1.1 登录的实现

电商秒杀系统的用户登录功能的实现原理是将输入的账号密码与数据库内的用户信息进行比对。我们的登录页面有三个输入框，账号、密码、验证码。点击登录按钮，首先我们将会对验证码进行判断，如果验证码不正确，直接不去查询数据库，返回给前端验证码错误的结果。如果验证码正确，则将账号密码去数据库比对，如果查到有对应的用户，则将其返回。存储到Session域中以便后续的操作。如果没有查到，则返回为空，返回前端说明用户或者密码错误。整个流程如5.1图所示。

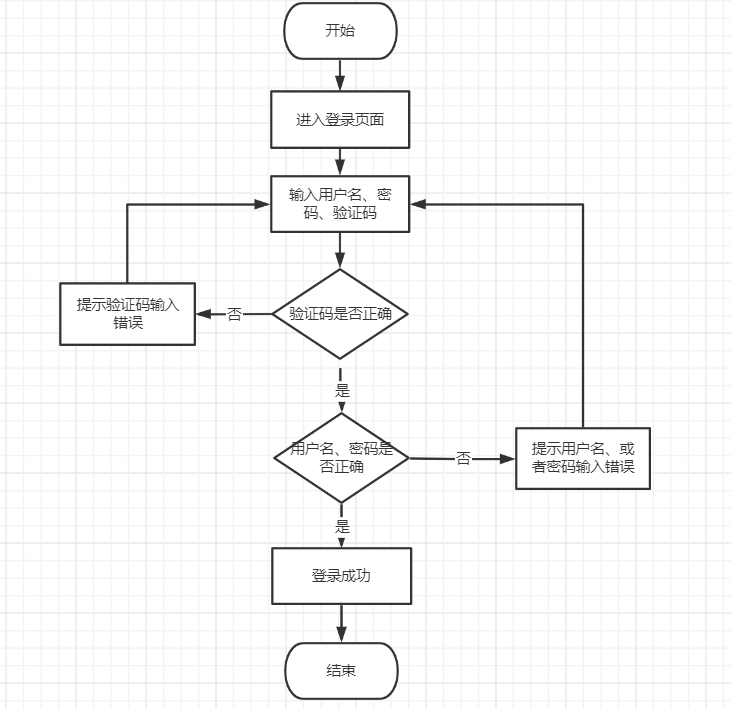


图5.1 登录流程图

核心代码如下：

public void login(HttpServletRequest request, HttpServletResponse response) {

HttpSession session = request.getSession();

String username = request.getParameter("username");

String password = request.getParameter("password");

User user = new User(username, password);

PrintWriter out = response.getWriter();

User loginUser = userService.login(user);

//清空整个jsonObject对象的内容

jsonObject.clear();

if (loginUser==null){

jsonObject.fluentPut("code","-1");

}else {

jsonObject.fluentPut("code","1");

session.setAttribute("user",loginUser);

}

out.print(jsonObject.toString());

out.flush();

out.close();

}

### 5.1.2 注册的实现

电商秒杀系统的注册功能的实现原理是将用户填写的注册信息提交给我们的用户表内存储。在数据库用户表内添加一条新的数据才算注册成功。在我们的注册页面中，有三个输入框吗，输入我们想要注册的用户名、密码、以及确认密码，在输入框后会有对应的提示输入信息，直到三个输入框都合法的，并且同意我们的注册协议的时候，才能提交注册请求。注册流程如图5.2所示。

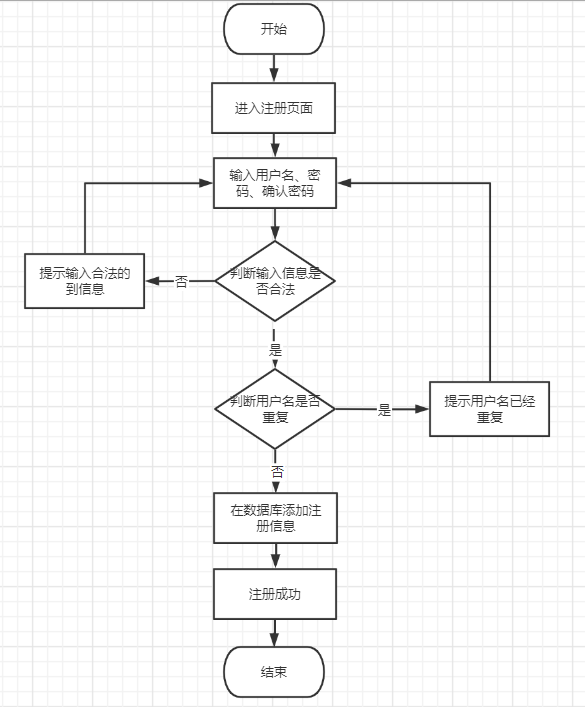


图5.2 注册流程图

核心代码如下：

private void registerUser(HttpServletRequest request, HttpServletResponse response) {

String username = request.getParameter("username");

String password = request.getParameter("password");

User user = new User(username,password);

User checkUser = userService.checkUser(user);

if (checkUser == null){

int i = userService.addUser(user);

response.sendRedirect("login.jsp");

}else {

response.sendRedirect("register.jsp");

}

}

### 5.1.3 注销的实现

电商秒杀系统的注销功能有2种方式，第一种是自动注销，当用户30分钟未操作电脑，则触发事件自动注销。第二种是手动注销，用户点击注销按钮，即可退出系统回到登录页面。注销流程如图5.3所示

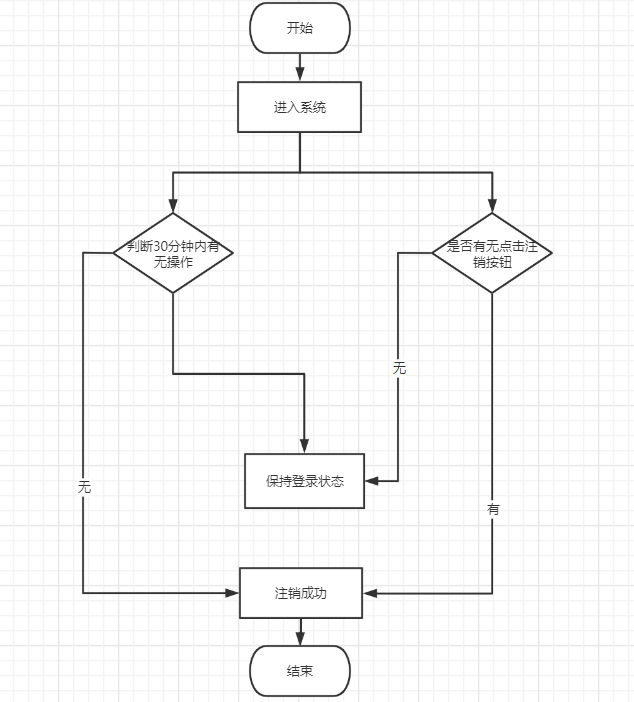


图5.3 注销流程图

核心代码如下：

private void logOut(HttpServletRequest request, HttpServletResponse response) {

HttpSession session = request.getSession();

session.invalidate();

response.sendRedirect("login.jsp");

}

## 5.2 商品模块的实现

### 5.2.1 搜索的实现

电商秒杀系统的商品搜索功能是根据关键字进行搜索，检索的是全文商品的名称。根据用户输入的文字信息进行检索，将得到的结果返回给商品展示页，将查询的结果展示到商品展示页面，就完了对应的搜索功能。

核心代码如下：

private void queryAllGoods(HttpServletRequest request, HttpServletResponse response) throws Exception {

int pn = NumberUtils.parseStr(request.getParameter("pn"));

String name = request.getParameter("name");

int totalCount = service.getTotalCount(name);

limit = totalCount > 100?12:8;

PageManager pm = new PageManager(pn,totalCount,limit);

List<Goods> goods = service.searchGoods(pm,name);

request.setAttribute("searcherGoods",goods);

request.setAttribute("pm",pm);

request.getSession().setAttribute("name",name);

request.getRequestDispatcher("searcherAllPage.jsp").forward(request, response);

}

### 5.2.2 秒杀的实现

在主页中有秒杀中心，其中有正在举行的秒杀活动，我们可以点击进入，查看想要秒杀的商品，在确定好想要秒杀的商品以后，点击秒杀购买按钮，完成秒杀订单的支付，即参与秒杀活动。流程如图5.4所示

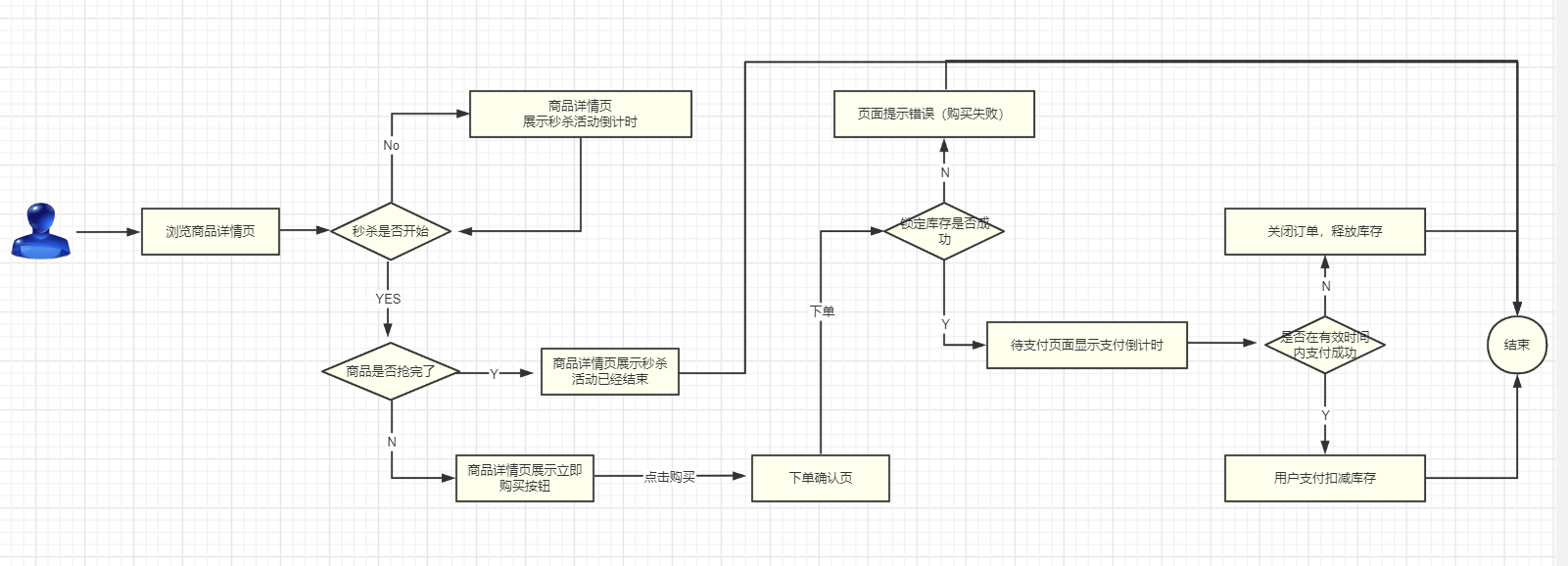


图5.4 秒杀流程图

核心代码如下：

private void queryPromo(HttpServletRequest request, HttpServletResponse response) throws Exception {

List<Promo> promoList = promoService.queryPromo();

List<Promo> jingxingzhong = new ArrayList<>();

List<Promo> weikaishi = new ArrayList<>();

List<Goods> goodsListjinxing = new ArrayList<>();

List<Goods> goodsListweikaishi = new ArrayList<>();

Date date = new Date();

for (int i=0; i < promoList.size();i++) {

if((date.getTime()-promoList.get(i).getStart\_time().getTime()>=0)&& (date.getTime() - promoList.get(i).getEnd\_time().getTime()<=0)){

jingxingzhong.add(promoList.get(i));

Goods goods = goodsService.queryGoods(promoList.get(i).getGood\_id());

List<GoodsPic> goodsPics = goodsService.queryGoodsPic(goods.getId());

goods.setGoodsPics(goodsPics);

goods.setDiscount(promoList.get(i).getPromo\_price());

goodsListjinxing.add(goods);

}else if (date.getTime() - promoList.get(i).getStart\_time().getTime()<0){

weikaishi.add(promoList.get(i));

Goods goods = goodsService.queryGoods(promoList.get(i).getGood\_id());

List<GoodsPic> goodsPics = goodsService.queryGoodsPic(goods.getId());

goods.setDiscount(promoList.get(i).getPromo\_price());

goods.setGoodsPics(goodsPics);

goodsListweikaishi.add(goods);

}

}

request.getSession().setAttribute("promoList",promoList);

request.getSession().setAttribute("jingxingzhong",jingxingzhong);

request.getSession().setAttribute("weikaishi",weikaishi);

request.getSession().setAttribute("goodsListjinxing",goodsListjinxing);

request.getSession().setAttribute("goodsListweikaishi",goodsListweikaishi);

request.getRequestDispatcher("product.jsp").forward(request, response);

}

## 5.3 订单模块的实现

因为订单模块和商品模块息息相关，所以我们需要绘制一个总的流程图才能将2个模块的流程讲述清楚。如图5.5所示

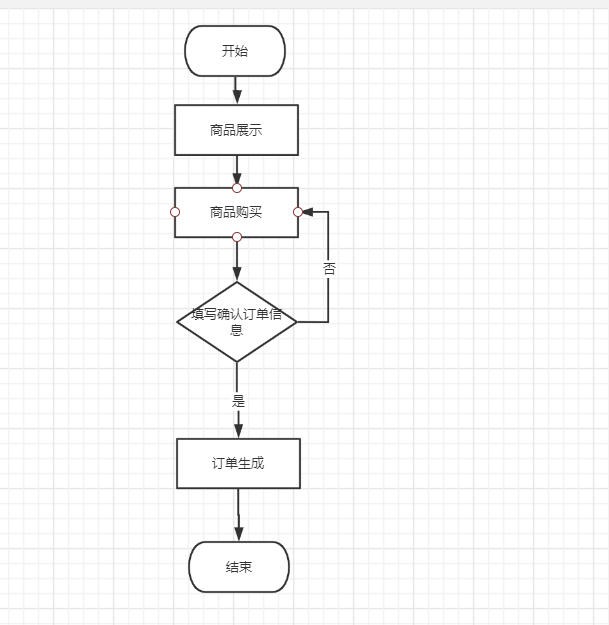


图5.5 订单流程图

### 5.3.1 订单创建的实现

在用户确认需要购买某种商品时，点击立即购买，则会在后台默认生成一个订单。订单的主要信息就是用户所选择的商品信息以及所购买的数量以及购买的时间等。

核心代码如下：

private void addOrder(HttpServletRequest request, HttpServletResponse response) throws Exception {

HttpSession session = request.getSession();

User user = (User)session.getAttribute("user");

String[] listgood = request.getParameterValues("goodsid");

String[] counts = request.getParameterValues("counts");

//前端数据 通过这个方法来传递 数组 后台接收数组参数

String price = request.getParameter("price");

String addressid = request.getParameter("addressid");

jsonObject.clear();

PrintWriter out = response.getWriter();

//将string数组转成int

goodsid = new int[listgood.length];

for (int i=0;i<listgood.length;i++){

goodsid[i] = Integer.parseInt(listgood[i]);

}

goodcount = new int[counts.length];

for (int i = 0; i < counts.length; i++) {

goodcount[i] = Integer.parseInt(counts[i]);

}

//商品总数

int allcount = 0;

for (int i = 0; i < goodcount.length; i++) {

allcount = goodcount[i]+allcount;

}

RecvAddress address = addressService.queryOneAddress(Integer.parseInt(addressid));

Order order = new Order();

Date date = new Date();

SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

String s = String.valueOf(date.getTime());

long l = Long.parseLong(s);

session.setAttribute("number",s);

int i = ordersGoodsService.addOrdersGoods(goodsid,goodcount,s);

order.setNumber(s);

order.setState("未付款");

order.setPrice(Double.parseDouble(price));

order.setCount(allcount);

order.setOrderDate(simpleDateFormat.format(date));

order.setUserId(user.getId());

//随便第一个商品的id传进去

order.setGoodId(Integer.parseInt(listgood[0]));

order.setAddressId(Integer.parseInt(addressid));

order.setPayWays("支付宝");

order.setOrderInfo(address.getRevName()+"-"+address.getAddress()+address.getDetailedAddress()+"-"+address.getPhoneNumber());

int i1 = orderService.addOrder(order);

if (i1>0){

jsonObject.fluentPut("code", "1");

jsonObject.fluentPut("number",s);

orderService.deleteshoppingCar(user.getId(),goodsid);

orderService.deleteStock(goodsid,goodcount);

}else {

jsonObject.fluentPut("code", "-1");

}

out.print(jsonObject.toString());

out.flush();

out.close();

new PayTimeThread(l,s).start();

}

### 5.3.2 订单查询的实现

在我们下了订单以后，我们可以在我们的个人中心进行订单的查询，包括查看订单的编号、下单时间、订单金额等等。

private void queryOrders(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

HttpSession session = request.getSession();

User user = (User) session.getAttribute("user");

Integer userid = user.getId();

int pn = NumberUtils.parseStr(request.getParameter("pn"));

int limit = NumberUtils.parseStr(request.getParameter("limit"));

int totalCount = orderService.getTotalCount(userid);

PageManager pm = new PageManager(pn,totalCount,limit);

//当前用户的订单并分页

List<Order> orders = orderService.queryOrders(userid,pm);

request.setAttribute("orders",orders);

request.setAttribute("pm", pm);

request.getRequestDispatcher("/vipOrder.jsp").forward(request,response);

}

## 5.4 支付模块的实现

### 5.4.1 订单支付的实现

创建订单以后，我们需要进入支付宝沙箱模型中去支付，当我们从创建订单开始就开启了一个线程进行倒计时，如果15分钟内还没支付订单，那么将自动取消此次订单。流程如图5.6所示

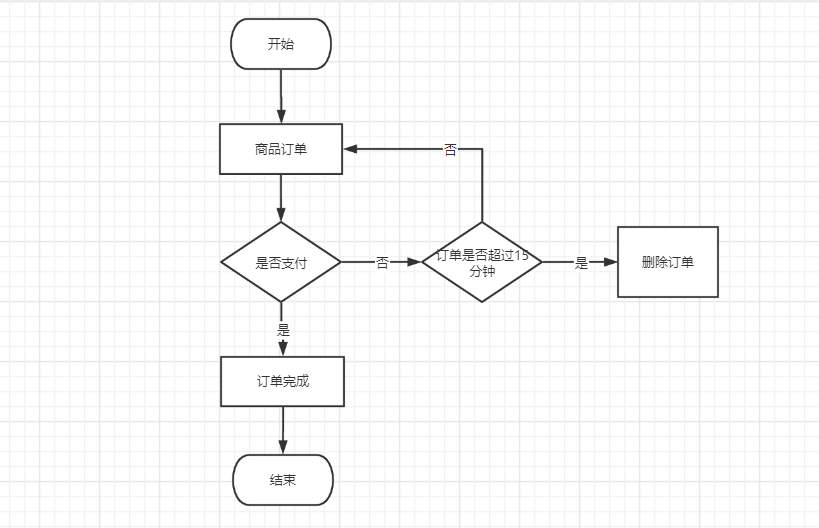


图5.6 订单支付流程图

核心代码如下：

AlipayClient alipayClient = new DefaultAlipayClient(AlipayConfig.gatewayUrl, AlipayConfig.app\_id,AlipayConfig.merchant\_private\_key,"json",AlipayConfig.charset,AlipayConfig.alipay\_public\_key, AlipayConfig.sign\_type);

AlipayTradePagePayRequest alipayRequest = new AlipayTradePagePayRequest();

alipayRequest.setReturnUrl(AlipayConfig.return\_url);

alipayRequest.setNotifyUrl(AlipayConfig.notify\_url);

Stringout\_trade\_no=newString(request.getParameter("number").getBytes("ISO-8859-1"),"UTF-8");

Stringtotal\_amount=newString(request.getParameter("totalPrice").getBytes("ISO-8859-1"),"UTF-8");

alipayRequest.setBizContent("{\"out\_trade\_no\":\""+ out\_trade\_no +"\","

+ "\"total\_amount\":\""+ total\_amount +"\","

+ "\"subject\":\""+ "\*" +"\","

+ "\"body\":\""+ "\*" +"\","

+ "\"product\_code\":\"FAST\_INSTANT\_TRADE\_PAY\"}");

//请求

String result = alipayClient.pageExecute(alipayRequest).getBody();

//输出

out.println(result);

### 5.4.2 订单退款的实现

在完成订单以后，会出现用户不满意商品的情况，电商秒杀系统应该支持退款操作。退款的流程如图5.7所示

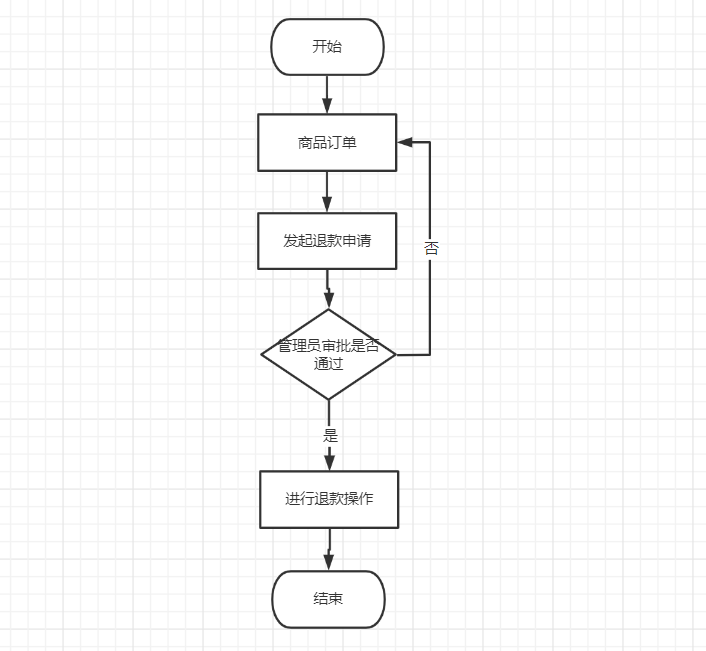


图5.7 订单退款流程

核心代码如下:

private void refund(HttpServletRequest request, HttpServletResponse response) {

Refund refund = new Refund();

refund.setNumber(request.getParameter("WIDTRout\_trade\_no"));

refund.setPrice(Double.valueOf(request.getParameter("WIDTRrefund\_amount")));

refund.setTime(new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date()));

refund.setUserId(((User)request.getSession().getAttribute("user")).getId());

refund.setReason(request.getParameter("WIDTRrefund\_reason"));

refund.setState("退款中");

service.addRefund(refund);

service.updateOrderState(refund.getNumber(),refund.getState());

try {

request.getRequestDispatcher("/order?method=queryOrders").forward(request,response);

} catch (ServletException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

# 第6章 系统测试

## 6.1 用户模块的测试

表6.1 用户模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 主要行为 | 基本输入 | 预期结果 | 测试结果 |
| 1-1 | 未登录点击主页 | 无 | 显示主页内容 | 显示主页内容 |
| 1-2 | 注册信息填写 | 用户名、密码 | 注册成功，跳转登录页 | 注册成功，跳转登录页 |
| 1-3 | 登录信息填写 | 用户名、密码、验证码 | 登录成功，跳转到主页 | 登录成功，跳转到主页 |
| 1-4 | 点击注销按钮、或者30分钟不操作系统 | 无 | 自动注销，跳转登录页 | 自动注销，跳转登录页 |

当用户未执行登录操作时，只能去访问首页，只有登录过后，才能使用电商秒杀系统的全部功能。未登录访问如图6.1



图6.1 未登录访问主页图

当用户没有账号时，可以点击注册按钮，进行账号的注册。在注册的时候，会根据输入的信息给出对应的提示，而且输入错误提交后也不会覆盖原来输入的内容，这让用户在注册的时候非常好的体验。进而想进入我们的网站进行购物。注册测试如图6.2所示



图6.2 注册测试图

在拥有了账号以后，我们可以前往登录页面进行登录操作。如图6.3、图6.4是我们登录测试的结果



图6.3 登录测试图



图6.4 登录成功结果图

## 6.2商品模块的测试

表6.2 商品模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 主要行为 | 基本输入 | 预期结果 | 测试结果 |
| 2-1 | 点击主页按钮 | 无 | 显示主页内容 | 显示主页内容 |
| 2-2 | 点击商品分类 | 无 | 跳转到指定类别的商品页面 | 跳转到指定类别的商品页面 |
| 2-3 | 点击某一商品 | 无 | 跳转到商品详情页 | 跳转到商品详情页 |
| 2-4 | 输入关键字搜索的商品 | 要查的商品名称 | 展示搜索出的商品 | 展示搜索出的商品 |
| 2-5 | 修改购买数量为0或者负数 | 0 | 提示数量不能为0 | 提示数量不能为0 |

用户登录电商秒杀系统以后，首先展示给用户的就是我们的首页，首页中各种商品的展示如图6.5

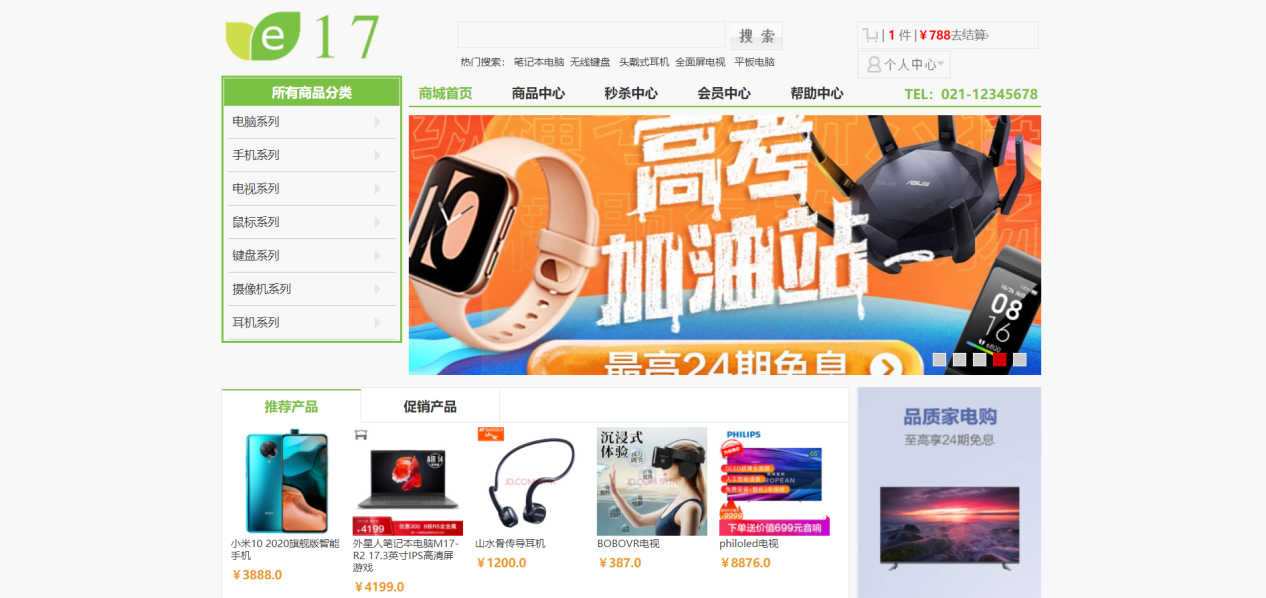


图6.5 首页商品展示

我们又将商品进行了分类，我们可以测试商品的分类，得到的结果如图6.6所示

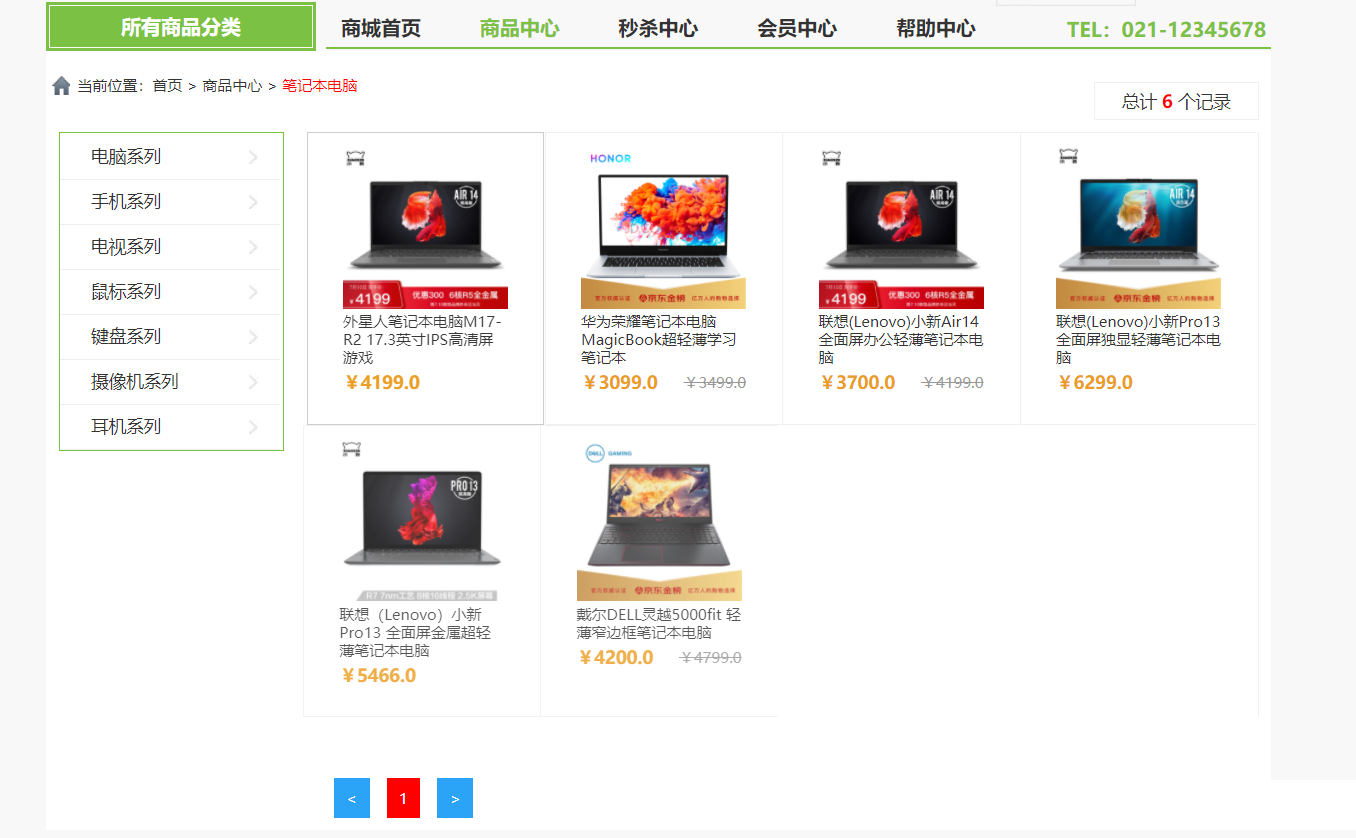


图6.6 商品分类结果

点击商品后，我们就跳转到商品详情页。商品详情页中有对商品的展示，并且介绍了商品的名称、价格等一系列参数。图6.7是我们对商品详情页的测试



图6.7 商品详情页测试

我们的主页还带有商品的搜索功能，图6.8是我们对商品搜索功能的测试

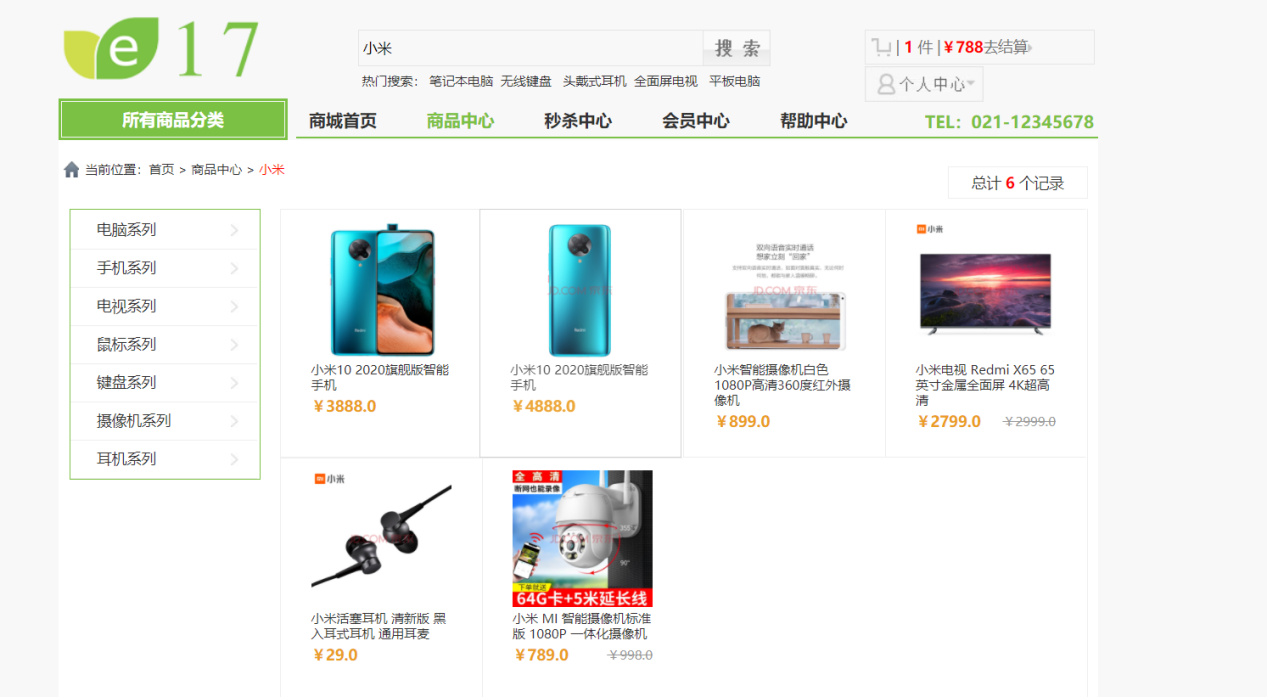


图6.8 搜索功能测试

## 6.3 订单模块的测试

表6.3 订单模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 主要行为 | 基本输入 | 预期结果 | 测试结果 |
| 3-1 | 点击立即支付按钮 | 无 | 跳转到支付页面 | 跳转到支付页面 |
| 3-2 | 点击我的订单 | 无 | 展示自己购买的所有订单 | 展示自己购买的所有订单 |
| 3-3 | 点击订单的详情 | 无 | 跳转到订单详情页 | 跳转到订单详情页 |

订单模块是我们系统的一个很重要的模块，订单模块的测试主要包括订单的核对、用户订单的查询、订单详情的展示等功能。图6.9是订单核对的测试、图6.10是订单查询的测试、图6.11是订单详情的测试。



图6.9 订单核对测试



图6.10 订单查询测试



图6.11 订单详情测试

## 6.4支付模块的测试

支付模块主要测试的功能就是订单支付和退款功能。支付功能测试的结果如图6.12、6.13所示。退款测试的结果如图6.14所示。



图6.12 支付测试



图6.13 支付完成测试



图6.14 退款结果测试

## 6.5本章小结

本章是对电商秒杀系统的模块功能进行了的测试，并列举了一系列的功能测试的结果和截图。

# 第7章 总结与展望

## 7.1总结

经过快1个多月的系统开发，电商秒杀系统的基本功能总算是完成了，在这一个多月中，有苦也有甜，有解决Bug时的欣喜，也有遇到问题时的愁容，总算还好有强大的互联网和学校的老师以及同学们的帮助。选做这个课题的初衷是为了帮助消费者去更加方便的购买商品以及提高消费者的网购体验。既能促进当今电商企业的发展，也能帮助消费者获得美好的购物体验，同时也带动了我国经济的发展，一举三得。接下来我将汇报一下我在研究过程中所做的一些工作：

1、在绪论部分中，我分析了当前中国电子商务的发展形势，阐明了我要研究的目标和要研究的内容，也说明了电商秒杀系统今后会怎样的发展。接着谈论的是实现电商秒杀系统所要用到的一些技术，简单的介绍了技术的作用。随后在需求分析上面对电商秒杀系统应有的功能进行了详细分析。

2、在第四章中，我们展开了对系统设计的分析，对于具体的功能分析，我们绘制了功能的流程图，然后再用代码实现具体的流程，完成对应功能的开发。并对用户在系统中应该如何进行功能的使用进行了说明。我们还对系统数据库的设计进行了详细的说明，从数据设计的概要再到设计的步骤，然后绘制E-R图，表明了实体与属性的联系，再到具体的表设计到每一个字段的设计，并且说明了每个表之间的联系，为系统的数据持久化做出了保证。

3、在第五章中，我们开始实现我们系统的功能，我们的系统是基于浏览器/服务器的，并且采用了MVC的三层架构模式，三层模型为系统后期的优化以及扩展提供了良好的基础。

4、在最后我们对系统的功能进行了测试，基本实现了我们前文所述的功能，能够实现消费者的购物或者是秒杀要求，通过对多个模块的测试，我们的系统和用户之间的交互是没问题的，基本完成了所要研究的内容。

## 7.2展望

课题完成了需求分析中的预定模块，但是因为技术的原因，有些功能做的还是不是很完善，和淘宝、京东那些大型的电商网站来说差距还是非常的大，而且在页面的美化这上面设计的不到位，系统的性能也可以进行一些优化，比如采用秒杀商品可以使用缓存，这样可以大大提高系统的并发性能，数据库可以使用分库分表，有助于减小数据库的压力等等。我期望在后期能够把系统完善的更好。

在整个开发过程中，让我觉得最重要的能力就是自学能力以及面对问题的处理能力。因为在开发过程中你可能遇到各种千奇百怪的问题，以及没有学过的技术，我们应该怎么样把技术学会然后应用到我们的项目中去，以及解决我们所遇到的问题，这个是非常关键的。我期望大家都能养成这样的能力，这将对我们的工作会非常的有帮助。

随着现在技术更新迭代的越来越快，终生学习成为我们当代程序员的一种标签，不要停下脚步，不然就会像个井底之蛙，不懂得天空是有多么的广阔，努力学习吧，去提高自己，学习才能帮助我们在路上越走越远。