



**本科毕业设计（论文）**

基于SSM的乐高糕点网上销售系统的设计与实现

学 号：11

学生姓名：李 常

指导教师：孙 亚 楠

所在学院：计算机科学技术学院

所学专业：软件工程专业

2018 年 6 月

摘 要

随着当今互联网的飞速发展，网上购物的需求量增加日益明显，虽然市面上也存在着诸如京东、淘宝、亚马逊之类的网上购物平台，但这些网站都是大型的，商品类型繁多，我们的目标的是做到小而精，由于人们的生活水平的提升，蛋糕成为一种不可或缺的商品，以至于把更美好的蛋糕展现给用户成为了我们的首选目标。本文主要阐述了使用前端最新技术和效果配合Java完成乐高糕点官方网站的设计与实现过程。包括从可行性到网站发布的相关文档。本网站基于Jquery类库，前端使用Bootstrap,Thymeleaf，Html等，后台使用SSM框架，Mysql等相关技术，实现了主页，商品详情，商品列表，购物车，收货地址详情，订单相信等页面。由于本系统使用前端Bootstrap一些新的元素和Thymeleaf的新增属性完成页面的设计和实现，体现了创新。使用Ajax的技术来与后台进行数据交互，局部刷新和异步提交的方式，减少了页面响应时间。使得用户的体验效果更加美好。体现了电商网站最重要的一点，用户体验第一。

**关键字**：Thymeleaf；SSM；Ajax；Bootstrap

Abstract

With the rapid development of the Internet, increased demand for online shopping is increasingly obvious, although there are on the market, such as jingdong, taobao, amazon, the online shopping platform, but the site is large, the commodity type is various, our goal is to be small and exquisite, due to the ascension of the people's standard of living, the cake became a indispensable commodity, that give the user a better cake show became our first target. This paper mainly describes the design and implementation process of the official website of lego cakes with the latest technology and effect of the front end. Includes documentation from feasibility to site release. This web site based on the jQuery library, front end using the Bootstrap, Thymeleaf, Html, etc., the background using Springboot, Mybatis, Mysql and other related technologies, has realized the home page, product details, list of goods, shopping cart, receiving details page. As the system USES the new elements of the front-end Bootstrap and the new properties of Thymeleaf to complete the design and implementation of the page, it embodies the innovation. Using AJAX techniques to interact with the background data, local refresh and asynchronous submission, reduces page response time. Make the user experience more beautiful. It reflects the most important point of e-commerce website, user experience first.

**Key words**: Thymeleaf,；SSM；Ajax；Bootstrap

目 录

[摘 要 1](#_Toc514929347)

[Abstract 2](#_Toc514929348)

[第1章 概述 1](#_Toc514929349)

[1.1 引言 1](#_Toc514929350)

[1.2 国内外现状研究 1](#_Toc514929351)

[1.3 研究内容和拟解决问题 2](#_Toc514929352)

[1.3.1 布局设计 2](#_Toc514929353)

[1.3.2 交互性 2](#_Toc514929354)

[1.3.3 性能 2](#_Toc514929355)

[第2章 方案论证 3](#_Toc514929356)

[2.1 可行性分析 3](#_Toc514929357)

[2.1.1 经济可行性 3](#_Toc514929358)

[2.1.2 技术可行性 3](#_Toc514929359)

[2.1.3 操作可行性 4](#_Toc514929360)

[2.1.4 法律可行性 4](#_Toc514929361)

[2.2 开发方案 4](#_Toc514929362)

[2.2.1 方案一 4](#_Toc514929363)

[2.2.2 方案二 4](#_Toc514929364)

[2.2.2 方案三 5](#_Toc514929365)

[2.3 方案确定 5](#_Toc514929366)

[第3章 需求分析 6](#_Toc514929367)

[3.1 需求描述 6](#_Toc514929368)

[3.2 功能性需求分析 6](#_Toc514929369)

[3.3 数据字典 9](#_Toc514929370)

[3.5 用例分析 11](#_Toc514929371)

[3.6 非功能需求分析 13](#_Toc514929372)

[3.5 运行环境需求分析 13](#_Toc514929373)

[第4章 设计论述 14](#_Toc514929374)

[4.1 概要设计 14](#_Toc514929375)

[4.1.1 系统总体开发流程及设计 14](#_Toc514929376)

[4.1.2 系统功能设计 14](#_Toc514929377)

[4.1.3 系统E-R图 15](#_Toc514929378)

[4.2 数据库设计 18](#_Toc514929379)

[4.3 详细设计 21](#_Toc514929380)

[4.3.1 系统环境实现 21](#_Toc514929381)

[4.3.2 登录模块 21](#_Toc514929382)

[4.3.3 注册模块 23](#_Toc514929383)

[4.3.4 购物车模块 25](#_Toc514929384)

[4.3.5 收货地址功能 26](#_Toc514929385)

[4.3.6 订单支付页面 28](#_Toc514929386)

[4.3.7 我的订单中心 29](#_Toc514929387)

[4.3.8 商品管理 30](#_Toc514929388)

[4.3.9 订单处理中心 32](#_Toc514929389)

[第5章 软件测试与结果分析 33](#_Toc514929390)

[5.1 测试的目的和原则 33](#_Toc514929391)

[5.1.1 测试的目的 33](#_Toc514929392)

[5.1.1 测试的原则 33](#_Toc514929393)

[5.2 测试计划 33](#_Toc514929394)

[5.3 测试过程 34](#_Toc514929395)

[5.3.1 测试用例 34](#_Toc514929396)

[5.4 测试结论 36](#_Toc514929397)

[结 论 37](#_Toc514929398)

[致 谢 38](#_Toc514929399)

[参考文献 39](#_Toc514929400)

[附 录 40](#_Toc514929401)

第1章 概述

1.1 引言

随着计算机的发展和普及,人民的生活变得越来越方便。互联网时代的来临为我们诸多方面带来了无限商机与便利。电子商务一词大家肯定不陌生,从我们的衣物、家居,再到生活用品。我们都可以通过各大电商网站提供的网络平台,点击购买自己喜欢的商品就可以了。极大地方便了我们的生活。对于电商网站来说,它只需要在网络上创建一个销售平台,将自己所有的商品展示在网络上。而用户可以查看、选购。而供货商只需要生产,然后通过物流便可以送到用户手中。没有中间商、零售商。这样一来商品就没有层层的中间商加价。这样用户就能以最低的价格获得自己心仪的商品，或者买到性价比最高的商品。节约成本,提高了速度,缩小了用户等待时间。

在2015年国家总理李克强提出“互联网+”的思想概念，更是掀起了一波大众创业、万众创新的浪潮，各种互联网企业如雨后春笋，形成了以互联网为基石的经济发展新形态。随着移动互联网和智能手机接近峰值，用户的需求也接近个性化，乐高糕点，它是一个专门为个性化量身定制的电商网站，不追求商品种类的数目，只卖蛋糕，一个能够买到全国最优质蛋糕的电商网站，以方便人们的生活，满足用户的需求为己任，为用户挑选优质的商品。

1.2 国内外现状研究

电商网站因其方便快速以及在用户和商家直接构建起一条方便快捷的渠道，得到广大用户的青睐从而迅速发展成为如今互联网最炙手可热的在线网站,随着移动互联网的到来，基本上智能手机已经普及，人们完全可以利用琐碎的时间上网，这大大减少了人们上网的成本，据统计在2017年我国成年人平均每天花1小时38分钟使用智能手机，这主要来源于手机的廉价以及网速的全面覆盖，所以电商网站这是一个很大的商机和市场。

在购物服务领域，Amazon 是国外人们最喜爱的购物网站。它是美国最大的电子商务公司，也是美国最火的，也可以说是全球最大的电子商务网站。目前它已在全球都有用网络平台来为用户提供购买服务。它从以前的全球第一书店成为现在涉及基本所有的商品行业。

而在中国，如今中国的物联网已经是世界一流了，我们不仅实现了“上午购物下午到家”，还普及了无现金的生活。例如现在我们日常生活中的支付方式支付宝和微信，正因为有了无现金支付的方式更加促进了电子商务的发展。在中国现在有最大的两家电商网站淘宝和京东，在去年双十一那天，阿里和京东从公布的数据显示交易额达2953亿元，这是一个惊人的数字，这也侧面的体现出电商网站是一个发展前景

但乐高糕点网也是一家在中国占有巨大销售量的电商网站。乐高糕点结合线上电商网站和线下的门店，以自己的物流为优势，不论用户需要的美食在何地，乐高糕点都可以提供。

1.3 研究内容和拟解决问题

1.3.1 布局设计

一个好的网站，必须要有一个好的布局，使得用户使用起来既方便又美观，体现了良好的用户体验，可以吸引用户的眼球。以前的网站，仅为了应付搜索引擎的搜索，对用户体验不是很重视。随着互联网的发展，重中之重是用户体验，许多网站都是把用户放在第一位考虑的，留住用户才能留住资本。所以，能留住用户首先需要网站美观，其次才是网站的功能。如果一个网站连基本的美观都做不到，那么第一眼已经失败了，因为现代社会生活节奏很快，用户没有耐心体验一个不美观的网站的功能，设计一个美观的网站才能吸引潜在的用户在网站上完成成交。

1.3.2 交互性

前端开发有几样任务是需要完成的，比如动画特效，复杂的交互功能，UI设计，可拓展性和后期维护等。

在前端开发过程中，有几项任务是必须完成的。例如U设计，动画特效，复杂的交互功能和后期维护等。

网站交互设计的目的在于增加网站界面的友好性，提高可用性，从而使用户能够方便快速和有效地网站特定的功能和用户自己想要完成的功能。

交互设计包含的内容：

1.界面设计：界面设计不仅仅是UI方面的设计，还涉及到此网站为用户能提供何种功能，用户可以完成何种操作。

2.导航设计：导航设计是一个网站设计尤为重要的地方。因为用户不是编写代码人员。没有简单易懂的样式提示用户如何操作，那么你这个网站便是失败的产品。

3.信息设计：信息设计就是在用户使用网站的过程中，会有一些温馨或必要的提示，避免用户误操作或不会操作。

1.3.3 性能

网站的性能，由于我国的人口基数的庞大，网站的访问量也增加许多的用户，网站的性能也是决定用户是否在此网站停留的重要原因。用户浏览一个网页，我们应该以最快的速度反应数据，所以我们应该在前端采用分布式的架构和在后台采用缓存的机制。根据web优化的黄金法则:前端程序大部分时间则花在各种页面元素，如图像、样式表和脚本等的下载上，因此这占用了80%的最终用户响应时间。因此在编写代码的时候，减少HTTP请求次数，这是解决页面快速显示的主要途径之一

第2章 方案论证

2.1 可行性分析

2.1.1 经济可行性

系统在开发中所有用到的软件和技术都是免费开源的，都是已经以开源的形式开源了，其中用到的类库和技术都可以在官网上下载，所以不存在开发费用问题，开发人员只需要一台电脑，可以连接上网即可，综上所述，开发的经济是可行的。

2.1.2 技术可行性

本系统使用Java语言开发，从2018年编程语言排行榜来看Java依旧是最受欢迎的语言，由于它的面向对象的思想和简单行、健壮性以及可移植行让我们有了选择它的理由。

后台采用SSM框架，SSM（SpringMVC+Spring+Mybatis）框架集由SpringMVC、Spring、Mybatis三个开源框架整合而成，其中SpringMVC 做控制器(controller)，Spring 管理各层的组件，MyBatis 负责持久化层，由于SSM框架越来越轻量级目前已经成为中小企业的主流框架，为程序员们所喜爱。

（1）SpringBoot

Spring boot基于Spring生态系统之上，把SpringMVC和Spring常用的配置进行封装，目的是为了减少开发人员的学习成本以及提高开发效率。众所周知Spring框架需要进行大量的配置，Spring Boot引入自动配置的概念，让项目设置变得很容易。Spring Boot本身并不提供Spring框架的核心特性以及扩展功能，只是用于快速、敏捷地开发新一代基于Spring框架的应用程序。换句话说，它并不是用来替代Spring的解决方案，而是和Spring框架紧密结合用于提升Spring开发者体验的工具。同时它集成了大量常用的第三方库配置(例如Jackson, JDBC, Mongo, Redis, Mail等等)，Spring Boot应用中这些第三方库几乎可以零配置的开箱即用(out-of-the-box)，大部分的Spring Boot应用都只需要非常少量的配置代码，开发者能够更加专注于业务逻辑。

对开发人员来说，换用Spring Boot以后，项目初始化方法变了，配置文件变了，另外就是不需要单独安装Tomcat这类容器服务器了，maven打出jar包直接跑起来就是个网站，但你最核心的业务逻辑实现与业务流程实现没有任何变化。

所以，用简单的语言概括Spring，SpringMVC，SpringBoot就是:

Spring 是一个“引擎”；

Spring MVC 是基于Spring的一个 MVC 框架；

Spring Boot 是基于Spring4的条件注册的一套快速开发整合包。

（2）Mybatis

Mybatis是一个一流的基于Java的持久层框架，支持自定义SQL，存储过程和高级Mappings。 Mybatis减少了很多JDBC代码和手工设置的参数和提取结果步骤。Mybatis可以使用一个简单的XML配置文件或者Annotation来配置和匹配实体，把接口和对应数据库记录的JAVA POJO匹配起来。

前端采用BootStrap JQuery框架，数据库使用MYSQL，建模工具使用StartUML,开发工具使用IntelliJ IDEA ，并用Git进行代码的托管。以上都是最流行的开发框架，官方网站都提供了技术demo,并且都有详细的官方文档。整合了开源社区很多人的贡献，所以能够提供稳定的技术支持，遇到难题可以在很多技术博客或则开源社区上很容易找到解决方案，综上所述，技术是可行的。

2.1.3 操作可行性

乐高糕点平台是一个为客户提供最简洁的购买全国优质蛋糕的平台,以WEB界面的形式暴露给用户，用户不需要有太多的计算机知识，和使用淘宝京东的购物网站一样，登录上系统就可以根据导航栏的提示进行商品的购买，整个网站具备完整独立的选购功能，网站结构简洁明了，容易使用，操作简单。无需处理复杂使用逻辑，即可使用，综上所述，操作是可行的。

2.1.4 法律可行性

从以上的分析可以看出，这次毕设是从事相关的学习活动，无商业利益。并且采用的相关技术都是开源免费版本，比如BootStrap就是Twitter公司免费开发的一个框架，我们可以对源码下载使用，其他少数的商业软件也是社区版，无需涉及授权问题，所以不存在法律不可行问题，综上所述，法律是可行的。

2.2 开发方案

2.2.1 方案一

方案一:前端 Angular2 + 后端 NodeJS Express。在该课程设计题目确定后我就确定了采用B/S架构的方案来开发本系统，原因有很多，例如B/S架构可以很好的支撑高并发的网络请求等。第一种方案设计采用前后端分离的技术，前端采用Angular2框架快速构建Web应用，Angular2 是Google 开源的前段框架，Google 有很多大型的应用都通过该框架构建，这些大应用验证了Angular2 的高可用性。NodeJS 是同样Google对开源社区的技术贡献，NodeJS是Google将Chrome 的V8引擎再次封装后的运行环境，有了该运行环境可以使用javascript 编写跨平台的应用包括后端服务。本方案采用 Angular2 和 NodeJS 实现该系统考录到都是运用javascript语言或其子类语言，这样有利于缩减前后端技术栈。

2.2.2 方案二

方案二:前端 React Reflux + 后端 SOS。经过对方案一的技术栈认真查阅过后，提出此方案：前端React Reflux和后端 SOS。React 是著名社交网站Facebook 开源出来的前段框架，React 是整个前端框架的一场技术革命。React 采用组件化构建整个单页面应用，组件化不仅有利于提高代码的重用度，而且有写于编写搞内聚低耦合的组件是代码。使用React不仅可以将前段组件化，同时有利于构建复杂业务逻辑应用。Reflux 是在React 基础上提出的一种前段数据存储方案，将不同的数据存储到不同的store里面，已实现MVC模式。SOS （Spring One Stop）是Spring 社区提出的后端开发一站式解决方案，包括Spring MVC,Spring 和Spring Data JPA，即可通过SOS 快速构建后端业务服务，并且暴露给前段Restful API。

2.2.2 方案三

方案三:前端 Bootstrap JQuery + 后端 SSM。经过对方案一和方案二的技术查阅和之前实习的那家公司用到的技术提出此方案：前端Bootstrap JQuery和后端SSM。Bootstrap也是著名社交网站Twitter开源出来的热门的前端框架，由于它的简单灵活、开发快捷使得它一直是GitHub上热门的项目。从BootStrap3开始框架的库包含了所有的移动设备的样式支持，由于的BootStrap的响应式布局CSS能够自适应台式、手机和平板电脑，这样使得我们可以将我们开发的页面嵌套在我们的移动应用程序里,并且它支持所有的主流浏览器，开发人员能够很容易上手。JQuery可以通过一行简单的标记添加都页面里面，由于它是一个JavaScript函数库，以至于更加方便我们函数功能的调用，并且Google 和 Microsoft对它都有很好的支持，如果我们不想下载JQuery库，直接就可以从CDN JQuery加载核心的文件。SSM(SpringMVC,Spring,Mybatis)是目前中小企业主流的后端开发框架，SpringMVC和Spring在方案二也有提到，相比于SSH(Struts2,Spring,Hibernate)框架来说SSM它的开发更加轻量级，用Java的注解将开发简化到极致，并且它的数据持久实现更加灵活，SQL语句的优化更加简便。SpringMVC相比于Struts2，前者是基于方法级别的并且是单列模式，后者是基于类级别的原型模式，每次请求都会实例化一个对象，所以SpringMVC在开发性能上远高于Struts2。

2.3 方案确定

方案的确定确实让我考虑了很久，从前端 Angular2 + 后端 NodeJS Express方案到前端 React Reflux + 后端 SOS方案再到最后的前端 Bootstrap JQuery + 后端 SSM方案，每一种方案都有各自的优点，结合技术方案和自己目前掌握技术的熟练度，以及自己在实习期间开发用到的技术，最后选择了前端 Bootstrap JQuery + 后端 SSM这一方案，毕设将采用Windows作为开发系统，使用B/S架构模式，用Java 开发语言，数据库使用Mysql数据库，开发的工具使用IntelliJ IDEA，使用Maven进行项目架包管理，建模工具用StartUML,并且使用Git作为版本迭代器，将代码托管在GitHub上，前端的模板渲染引擎使用Thymeleaf。

第3章 需求分析

3.1 需求描述

本系统主要是提供各种精美的蛋糕供用户选择购买，最终的目的是让用户完成商品的购买，系统分为普通游客、注册用户和后台管理员，当游客要购买自己心仪的商品时首先要注册，注册用户则要登录，登录成功后用户可以将自己喜爱的商品放入购物车或则直接购买，然后在购物车结算，生成订单。用户可以在自己的我的订单里查看自己购买的商品详情信息。管理员则可以登录后台系统对商品添加或则下架，以及对商品信息的修改和一些商品要促销时折扣信息的修改，注册用户还可查看自己的收货地址，修改自己的收货地址以及添加自己的收货地址。

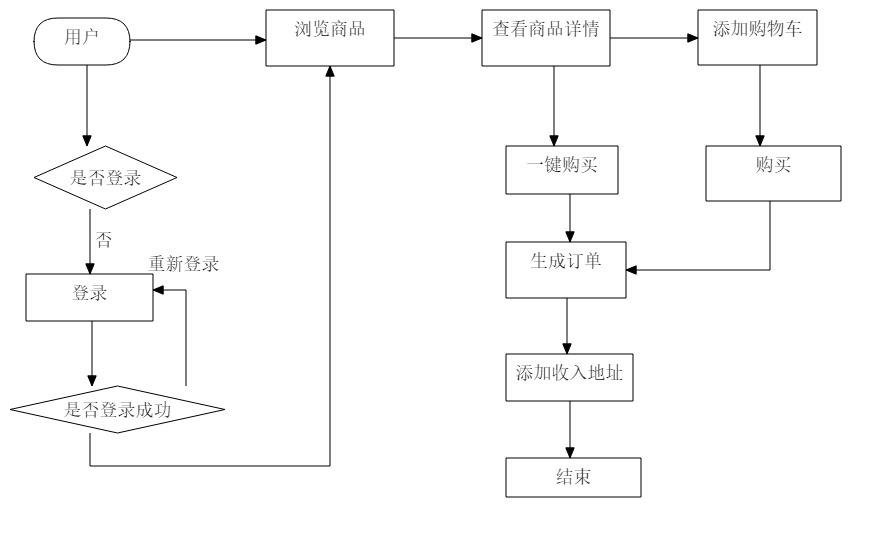


图 3-1 业务流程图

## 3.2 功能性需求分析

1. 管理员操作

（1）添加商品详细信息：

管理员可以添加商品的图片、商品名称、商品价格、商品颜色、商品尺寸，图片的上传可以做一下图片大小的限制，将图片的大小为5M，并且图片的格式仅支持png和jpeg。并且图片不应存在后台数据库，会影响数据库的性能，我们将图片路径保存在数据库，图片则保存在硬盘。发送请求后会返回添加的结果信息，结果信息分为添加成功和添加失败，管理可以通过结果信息判断商品是否添加成功。

修改商品信息：

管理员在商品列表里根据商品的ID可以修改商品的图片、名称、价格、颜色、尺寸进行修改，商品列表的商品ID为唯一主键，不可以被修改的，还有创建时间也不能被修改，创建时间是商品添加的时候获取的系统时间。发送请求会放回修改状态的结果信息，可以通过结果查看是修改成功，也可以在商品列表查看修改商品ID的商品信息时候修改。

删除商品

管理员可以在商品列表查看所有商品的商品信息，可以根据商品ID进行删除相应的商品，发送请求会提示你确定要删除该商品的消息提示，这更加增添用户的友好度，同样会返回删除的结果状态信息，删除成功和删除失败的信息。

查看商品列表

管理员可以查看所有商品的商品信息，包括商品图片、商品名称、商品价格、商品ID、商品尺寸、颜色以及注册商品的创建时间。管理员还可以根据商品的ID来指定查询商品的信息。

订单处理

管理员需要对用户提交的订单进行处理，管理员看到的订单信息包括订单ID，商品图，商品名称，数量，总金额，订单创建时间，支付状态，以及发货状态，如果该条订单为已支付状态则管理员需要将发货状态改为正在发货，否则不用修改。

2. 游客操作

（1）浏览网站

游客可以在没有登录的状态对网站进行访问，具体可以操作的地方为浏览网站首页，以及在商品中心里面浏览所有的商品。

查看商品详情

游客可以在商品中心浏览所有的商品，如果有喜欢的商品可以点击查看商品详情，商品详情里会展示商品的图片，一般为三张，图片可以被放大观看，还有商品的名称，商品的价格，商品的详细介绍等信息。

注册个人信息

游客可以注册自己的个人账户，需要注册的字段为姓名、昵称、性别、邮箱、电话号码、以及密码，其中电话号码和密码为必填字段，为了交互的友好性，密码与确认密码相匹配，如果两次密码不一样则不会发送注册请求，发送请求后会返回注册的结果信息注册成功或则注册失败信息。

3. 注册用户操作

（1）登录

注册用户的登录功能需要用户输入电话号码和密码或则邮箱和密码进行登录，点击登录按钮后会返回登录的结果信息，结果成功或则密码错误、或则该账户不存在等信息。

查看个人信息

注册用户在登录状态下可以查看自己的个人信息，个人信息包括用户名，昵称，电话号码，邮箱，性别以及注册日期。

修改个人信息

注册人员可以修改自己的个人信息，由于用户ID为唯一主键，用户在登录状态下可以拿到自己的ID号，可以修改的字段为用户名、昵称、电话号码、邮箱，但注册日期不能被修改。

查看订单详情

注册用户在购买商品并生成订单后就会有订单信息，用户可以查看自己的订单信息，包括订单号、商品图片、商品名称、数量、总价格、支付状态。订单状态分为已支付和未支付，如果为未支付可以在操作按钮点击立即支付按钮，如果为已支付就会显示发货状态，发货状态为已发货和未发货状态。

修改订单信息

用户可以根据订单号修改订单信息，但是订单信息只能修改商品的数量。修改商品数量或相应的支付总价也变为商品单价乘以数量的结果，点击确认修改按钮会返回修改的结果信息修改成功或则删除失败信息。

修改订单

用户可以把不需要的订单或者已经购买完成的订单信息删除，删除时根据用户的订单ID删除。当点击删除按钮是会提示确定要删除该条订单信息的提示，如果点击确定会返回删除的结果信息删除成功或则删除失败消息，同时在订单列表查看会发现该条订单信息不存在了。

添加收货地址

在用户需要购买商品时需要填收货人的地址，收货地址包括收货人姓名、收货人省市、收货人详细地址、收货人电话号码等信息。添加完成后会显示添加的结果信息保存成功或则保存失败。

修改收货地址

用户在已经添加的收货人信息的基础上进行收货人地址的修改，修改的时候收货人的姓名不能修改，其他字段可以修改。

删除收货地址

用户可以将不用的收货人地址删除，删除时根据收货人地址的ID进行删除。

添加商品到购物车

用户在登录状态下将自己喜爱的商品添加到自己的购物车，购物车列表里会显示购物车的商品名称、商品价格、商品图片、小计，操作等信息，用户在购物里可以勾选自己要购买哪些商品，当勾选上自己要购买的商品后会生成想要的支付总金额信息。

将商品从购物车移除

用户可将添加到购物车的商品进行移除，移除分为两种，一种是移出单个商品一种为清空购物车，移出时按照购物车里商品的ID进行移出，移出购物车后我在查看我的购物车会发现之前要移出的商品不存在了。

3.3 数据字典

从功能性需求分析可以得出以下数据数据字典

数据结构：用户信息

含义说明：记录用户的信息

组成：用户ID，用户名，密码，性别，电话号码，邮箱，注册日期

① 数据项：用户ID含义说明：用户的唯一区别，好比我们的身份证号码，。

② 数据项：用户名含义说明：用户的名称。

③ 数据项：密码含义说明：用户登录时的唯一凭证。

④ 数据项：性别含义说明：用户身份的鉴别。

⑤ 数据项：电话号码含义说明：用户的联系方式。

⑥ 数据项：注册日期含义说明：记录用户是什么时候注册账号的。

数据结构：商品信息

含义说明：记录商品的信息

组成：商品ID、商品名称、价格、图片地址、商品详细信息、标记

① 数据项：商品ID含义说明：商品的唯一区别，和用户ID类似，唯一标识。

② 数据项：商品名称含义说明：商品的名称。

③ 数据项：价格含义说明：记录商品的价格。

④ 数据项：图片地址含义说明：记录商品图片保存的位置。

⑤ 数据项：商品详细信息含义说明：记录商品的详细信息。

⑥ 数据项：标记含义说明：标记商品的状态是否删除。

数据结构：购物车信息

含义说明：记录购物车的信息

组成：购物车ID、用户ID、商品ID、数量、总价格、创建时间、标识

① 数据项：购物车ID含义说明：购物车的唯一区别，和用户ID类似，唯一标识。

② 数据项：用户ID含义说明：外键，用于关联用户信息表。

③ 数据项：商品ID含义说明：外键，用于关联商品信息表。

④ 数据项：数量含义说明：记录购物车里每个商品的数量。

⑤ 数据项：总价格含义说明：记录购物车里每一个商品乘以数量的价格。

⑥ 数据项：创建时间含义说明：标记购物车里添加商品的时间。

⑦ 数据项：标识含义说明：用于后续拓展使用。

数据结构：地址表

含义说明：记录用户收货地址的信息

组成：地址ID，用户ID，地址名称，地址，详细地址，电话号码，创建时间，标识

① 数据项：地址ID含义说明：用户收货地址的唯一区别，主键

② 数据项：用户ID含义说明：外键，用户关联用户信息表。

③ 数据项：地址名称含义说明：记录收货地址的名称。

④ 数据项：地址含义说明：记录收货地址的一级地址。

⑤ 数据项：详细信息含义说明：记录收货地址的详细地址。

⑥ 数据项：电话号码含义说明：记录收货人的联系方式。

⑦ 数据项：创建时间含义说明：记录收货信息的创建时间。

⑧ 数据项：标识含义说明：记录收货地址是否可用。

数据结构：订单表

含义说明：记录订单表的信息

组成：订单ID，用户ID，地址ID，创建时间，总价格，付款标记，标识

① 数据项：订单ID含义说明：订单表的唯一区别，主键

② 数据项：用户ID含义说明：外键，用户关联用户信息表。

③ 数据项：地址ID含义说明：外键，用于关联收货地址表。

④ 数据项：创建时间含义说明：记录订单创建的时间。

⑤ 数据项：总价格含义说明：记录订单里的所有商品的价格之和。

⑥ 数据项：付款标识含义说明：记录订单表是否已经付款。

⑦ 数据项：标识含义说明：记录订单表的订单是否失效。

数据结构：订单详细表

含义说明：记录订单详细表的信息

组成：ID，订单ID，商品ID，创建时间，标识

① 数据项: ID含义说明：订单详细表的唯一区别，主键

② 数据项：商品ID含义说明：外键，用户关联商品信息表。

③ 数据项：订单ID含义说明：外键，用于关联订单表。

④ 数据项：创建时间含义说明：记录订单创建的时间。

⑤ 数据项：标识含义说明：标记订单详细表是否可用。

3.5 用例分析

用户角色分为三种，一种为管理员，一种为普通游客，一种为注册用户。管理员对商品后天管理系统进行后台的管理，添加商品，修改商品信息，删除商品。普通游客只能对乐高糕点网浏览商品和查看商品详情信息。注册用户可以查看自己的个人信息，修改自己的信息，将商品添加到购物车，将商品移除购物车，查看订单，删除订单，添加收货人地址，修改收货人地址，删除收货人地址。

管理员用户图：

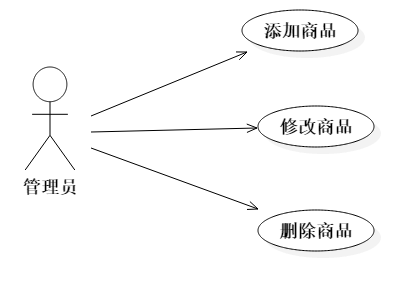


图3-2 管理员用例图

对功能做如下描述：

添加商品：由于时间的不同或者节日的不同，我们需要对一些商品进行添加替换，这部分商品有管理员进行添加。

修改商品：一些线上在售卖的商品需要进行相应的修改，比如修改商品的名称、图片或则价格等需要由管理员进行操作。

删除商品：网站上再卖的一些商品进行下架处理，或则说该商品不再卖了，可以进行删除。

普通游客用例图：

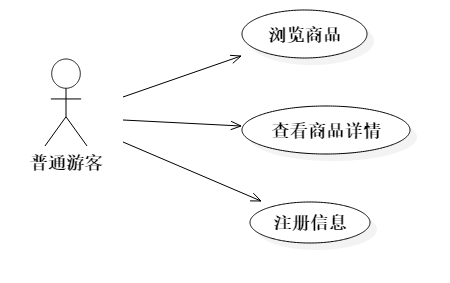


图3-3 普通游客用例图

对功能做如下描述：

浏览商品：游客对自己感兴趣的商品进行浏览

查看商品详情：游客可以对自己喜欢的商品查看详细信息

注册信息：游客可以注册自己的账号信息。由于购买商品必须在登录状态下，所以游客可以注册信息。

注册用户用例图：

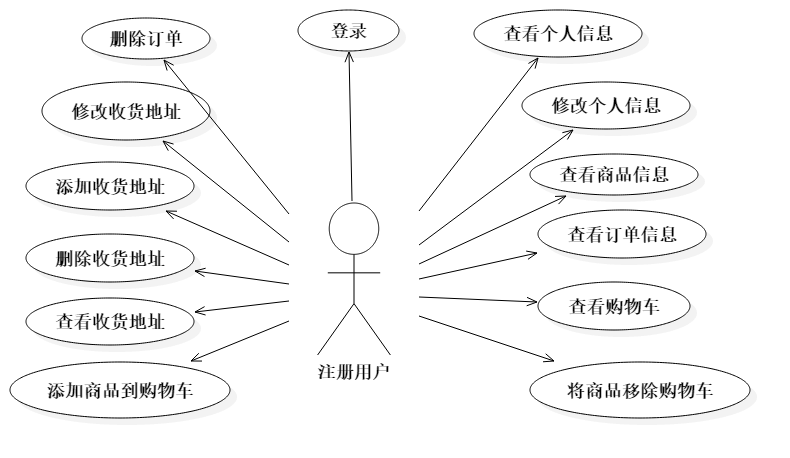


图 3-4 注册用户用例图

对功能做如下描述：

修改个人信息：用户可以对之前填写的一些个人信息进行修改。

查看商品信息：用户可以浏览网站上展示的商品信息。

添加购物车商品：用户可以对购买自己喜欢的商品，想将选中的商品放入购物车。

删除购物车商品：用户可以对已放入在购物车里的商品进行移除。

添加收货地址：用户可以添加新的收货人的信息。

删除收货地址：用户可以删除不在使用的收货人的信息。

修改收货地址：用户可以修改收货人已经变动的信息字段。

查看收货地址：用可可以查看已经存在的一些收货人地址信息。

查看订单信息：用可可以查看已经购买过的一些订单信息。

下订单：用户可以对购物车的商品购买，也可在商品展示列表直接购买，产生订单。

3.6 非功能需求分析

软件为满足用户业务需求以为的必须具有的除功能以外的需求称为非功能需求。包括系统性能，系统的适应性，维护下，可移植性、有效性和可扩充性等，下面对某些指标予以说明。

系统的可靠性、适应性和可维护性紧密联系在一起。系统应该支持故障和错误操作后的恢复，用户在运行过程中出错应在短时间内定位到问题所在。网络异常和异常中断应有容错机制。

除此之外，系统还要有较好的可维护性，可移植性，安全性、可靠性、互操作性、健壮性、易使用性、可重用性、可扩充性。

3.5 运行环境需求分析

操作系统：Windows10的操作系统

开发环境：Mysql5.1.7、JDK1.8 和 IntelliJ IDEA

第4章 设计论述

4.1 概要设计

4.1.1 系统总体开发流程及设计

本系统主要采用B/S框架结构，用户从浏览器里发出HTTP请求,服务器接收相应的请求然后做出处理，将数据通过SQL的方式存入数据库，再从数据库里返回数据或结果到服务器，服务器做出响应，最后返回相应的html界面给浏览器，这样给我们带来的好处是分布性强、维护方便、总体成本比较低、共享性强而且开发简单。

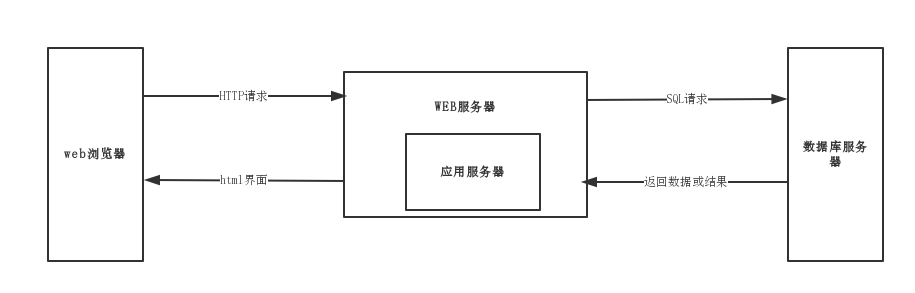
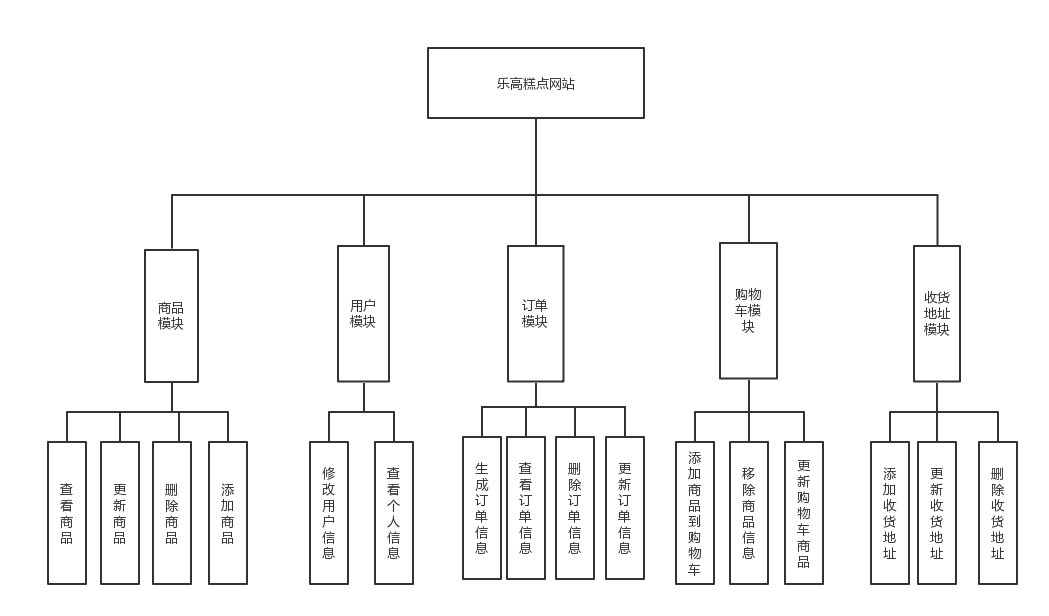


图4-1 开发总体流程图

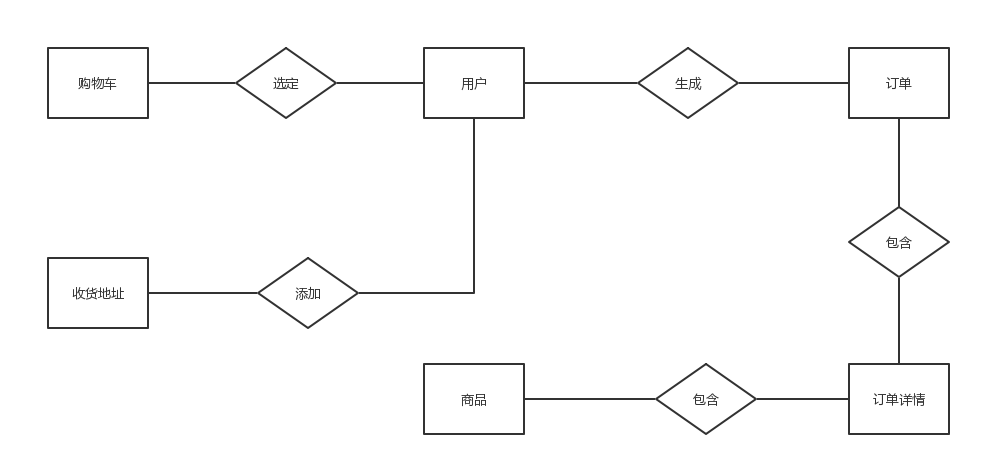
4.1.2 系统功能设计

系统主要分为管理员、普通游客、注册用户三个角色，其中主要的能功能模块主要如下几个：个人信息模块，商品信息管理模块，购物车模块，收获地址模块，订单信息模块。个人信息模块主要有对个人信息的查看、修改功能，商品管理模块主要有对商品信息的添加、修改、删除功能，购物车模块主要有查看购物车列表、添加商品到购物、删除购物车商品功能，收获地址模块主要有添加收获地址信息、修改收货地址信息、修改收货地址信息，订单信息模块主要有查看订单列表、删除订单、立即支付订单功能。乐高糕点网站系统整体功能模块图如下：

图4-2 功能模块图

4.1.3 系统E-R图

根据以上的分析可以，可以得出以下的系统E-R图。

1. 系统E-R图如下：

4-3 E-R图

2. 用户实体图，其中属性值包括用户名、密码、性别、电话号码、邮箱、注册日期，如图4-4图：

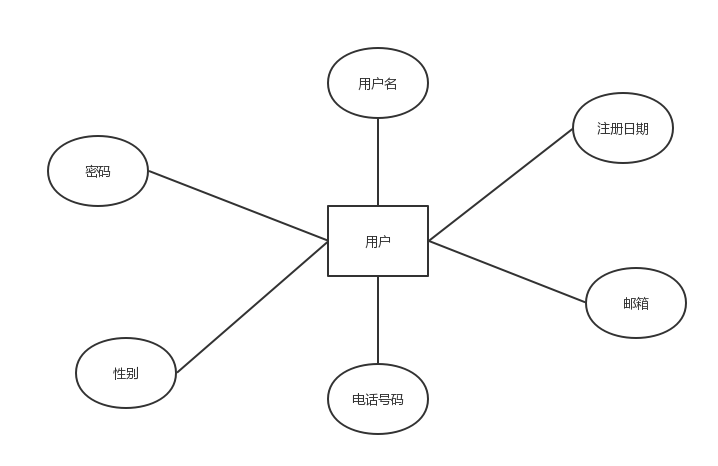


图4-4 用户实体图

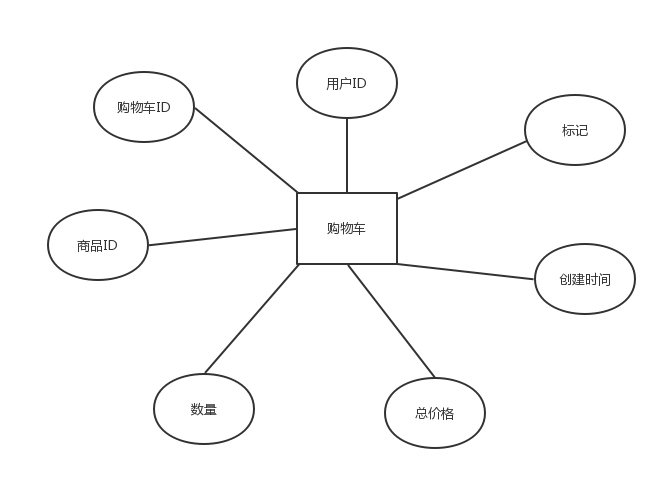
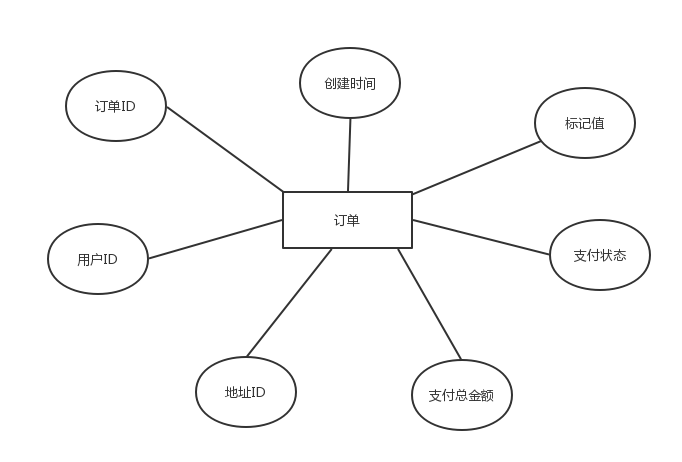
3. 购物车实体图，其中实体属性值为用户ID、购物车ID、商品ID、数量、总价格、创建时间、标记，如图4-5图：

图4-5 购物车实体图

4. 订单实体图，其中实体属性值为订单ID、用户ID、收货地址ID、支付总金额、支付状态、创建时间、标记值，如图4-6图：

图4-6订单实体图

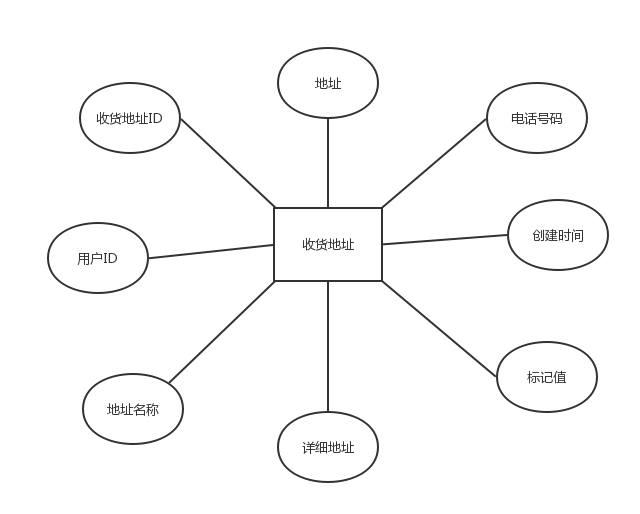
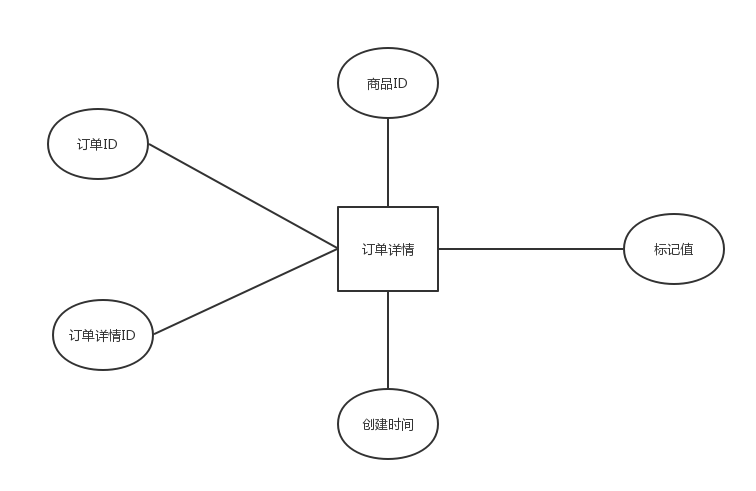
5. 收货地址实体图，其中实体属性值收货地址ID、用户ID、地址、地址名称、详细地址、电话号码、创建时间、标记值，如图4-7图：

图4-7 收货地址实体图

6. 订单详情实体图:其中实体属性值包括订单ID、商品ID、订单详情ID、创建时间、标记值，如图4-8图：

图4-8 订单详情实体图

4.2 数据库设计

乐高糕点系统实现的表结构设计，由于系统不是很大，我们主要设计了六张表，用户信息表，商品信息表，购物车表，用户地址表，订单表，订单商品详情表。

1．用户信息表（User）

用户信息表主要保存了用户的基本信息，用作用户登录系统以及购买商品，表字段包括用户ID、用户名、登录密码、性别、邮箱、电话号码、注册时间，用户ID为唯一主键，结构如表4-1所示。

表 4-1 用户信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 主外键 | 类型 | 长度 | 允许空 | 字段说明 |
| 1 | uid | 是 | int |  | 否 | 用户ID |
| 2 | Username | 否 | varchar | 50 | 否 | 用户名 |
| 3 | Password | 否 | varchar | 50 | 否 | 密码 |
| 4 | Sex | 否 | Int |  | 是 | 性别 |
| 5 | Mobile | 否 | char | 11 | 是 | 电话号码 |
| 6 | Email | 否 | varchar | 50 | 是 | 邮箱 |
| 7 | register\_time | 否 | datetime |  | 是 | 注册时间 |

2．商品信息表（product）

商品信息表（product）主要保存的是商品的基本属性信息，表字段主要包括商品ID、商品名称、图片路径、价格、商品详情、标记字段，其中商品ID为唯一主键，由于考虑到性能原因没有将图片存入数据库，数据只是存的图片的路径，标记字段用作于该商品是否下架或删除等用，结构如表4-2所示。

表4-2 商品信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **主键** | **类型** | **长度** | **允许空** | **字段说明** |
| 1 | Pid | 是 | Int |  | 否 | 商品ID |
| 2 | Pname |  | varchar | 50 | 是 | 商品名称 |
| 3 | Image |  | varchar | 50 | 是 | 图片路径 |
| 4 | Price |  | double | 50 | 是 | 价格 |
| 5 | Detail |  | varchar | 50 | 是 | 商品详情 |
| 6 | Flag |  | Int | 50 | 是 | 标记字段 |

3．地址信息表(address)

地址信息表（address）主要保存的是用户购买商品后的收货人地址信息，表字段主要包括地址ID、用户ID、地址名称、地址、详细地址、收货人电话号码、创建时间、标记字段，其中地址ID为唯一主键，用户ID为外键，对应用户信息表的用户ID字段，结构如表4-3所示。

表4-3地址信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **主键** | **类型** | **长度** | **允许空** | **字段说明** |
| 1 | Aid | 主 | Int |  | 否 | 地址ID |
| 2 | Uid |  | Int |  | 否 | 用户ID |
| 3 | Add\_name |  | varchar | 50 | 是 | 地址名称 |
| 4 | Address |  | varchar | 50 | 是 | 地址 |
| 5 | Detail\_address |  | varchar | 50 | 是 | 详细地址 |
| 6 | Phone |  | char | 11 | 是 | 电话号码 |
| 7 | Create\_time |  | dateTime |  | 是 | 创建时间 |
| 8 | Flag |  | Int |  | 是 | 标记字段 |

4．购物车表(cart)

购物车表（cart)主要保存的是用户将自己的喜爱的商品保存的数据，表字段主要包括购物车ID、用户ID、商品ID、商品数量、商品总价格、创建时间、标记字段，其中购物车ID为唯一主键，用户ID为外键，对应用户信息表的用户ID字段，商品ID为外键，对应商品信息表的商品ID，商品的总价格为商品的单价乘以商品的数量。结构如表4-4所示。

表4-4购物车标

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **主键** | **类型** | **长度** | **允许空** | **字段说明** |
| 1 | Cid | 是 | int |  | 否 | 购物车ID |
| 2 | Uid | 外 | int |  | 否 | 用户ID |
| 3 | Pid | 外 | int |  | 否 | 商品ID |
| 4 | Num |  | int |  | 是 | 商品数量 |
| 5 | Total\_price |  | double |  | 是 | 商品总价格 |
| 6 | Create\_time |  | datetime | 50 | 是 | 创建时间 |
| 7 | Flag |  | int | 50 | 是 | 标记字段 |

5．订单表（orderInfo）

订单表（orderInfo）主要保存的是用户用户购买商品的信息和收货地址信息，表字段主要包括订单ID、用户ID、地址ID、支付总金额、创建时间、支付状态、标记字段，其中订单ID为唯一主键，用户ID为外键，对应用户信息表的用户ID，地址ID为外键，对应地址信息表的地址ID。结构如表4-5所示。

表4-5订单表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **主键** | **类型** | **长度** | **允许空** | **字段说明** |
| 1 | Oid | 是 | int |  | 否 | 订单ID |
| 2 | Uid | 外 | int |  | 否 | 用户ID |
| 3 | Aid | 外 | int |  | 否 | 地址ID |
| 4 | Total\_price |  | double |  | 是 | 总金额 |
| 5 | Create\_time |  | datetime | 50 | 是 | 创建时间 |
|  | Pay\_mak |  | int |  | 是 | 支付状态 |
| 6 | Flag |  | int | 50 | 是 | 标记字段 |

6. 订单详细表（orderDetailInfo）

订单详细表（orderDetailInfo）主要保存的是一个订单里面的每一个商品的信息，由于一个订单里面有多个商品，表字段主要包括订单详细ID、订单ID、商品ID、创建时间、标记字段，其中订单详细ID为唯一主键，订单ID对应订单表的订单ID，商品ID对应的是商品信息表的商品ID。结构如表4-6所示。

表4-6 订单详细表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **字段名** | **主键** | **类型** | **长度** | **允许空** | **字段说明** |
| 1 | Id | 是 | int |  | 否 | 订单详细ID |
| 2 | Oid | 外 | int |  | 否 | 订单ID |
| 3 | pid | 外 | int |  | 否 | 商品ID |
| 4 | Create\_time |  | datetime | 50 | 是 | 创建时间 |
| 5 | Flag |  | int | 50 | 是 | 标记字段 |

由于我们用的是关系型数据库Mysql,所以表与表之间有一定的关联关系，对表之间的联系我们做如下的说明，如图4-7：

用户信息表和购物车表：由于一个用户具有一个购物车，所以用户表和购物车表为一对一的关系。

用户信息表和地址信息表：由于一个用户可以有多个收货地址，所以用户表和地址信息表为一对多的对应关系。

用户信息表和订单表：由于一个用户可以创建多个订单，所以用户信息表和订单表为一对多的对应关系。

订单表和地址信息表：由于一个订单里面只能包括一个收货地址，所以订单表和地址信息表对应关系为一对一的关系。

订单表和商品详情表：由于一个订单里面有多个商品，所以生成一个中间表订单详情表记录每一个订单的商品详情信息，所以订单详情和订单表的关系为多对一的关系，订单详情表和商品信息表的对应关系为一对多的关系。

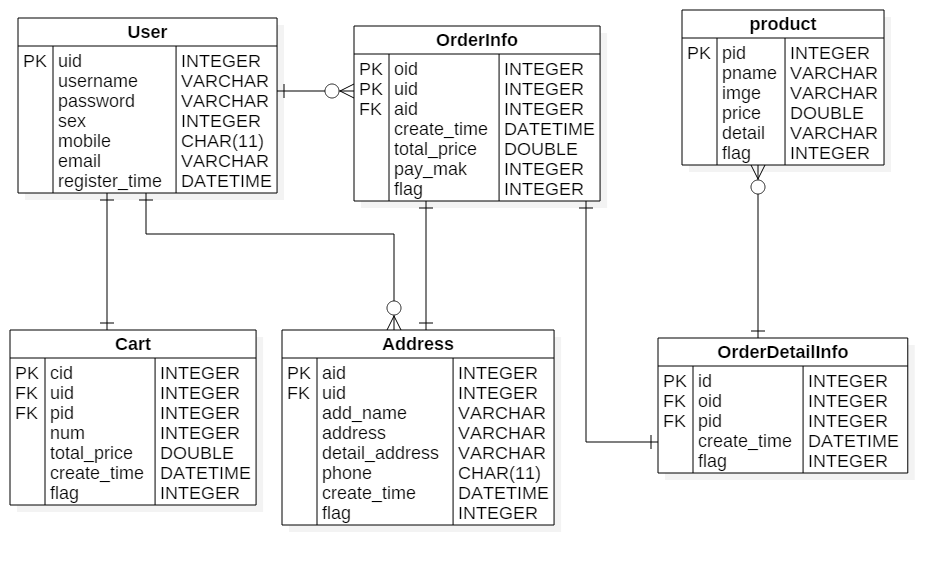


图 4-9 表的关系图

4.3 详细设计

4.3.1 系统环境实现

系统的实现建立在需求分析和系统设计之上，在IntelliJ IDEA和MySQL的开发工具之上实现乐高糕点系统的开发。操作系统采用Windows10；发布服务器采用SpringBoot自带的Tomcat；开发和编译环境采用IntelliJ IDEA+JDK1.8；CPU采用Intel core i7；网络环境无特别要求；浏览器采用谷歌的Chrome。

4.3.2 登录模块

用户登录网站后可以看见首页，用户可以在右上角点击登录的按钮，然后弹出一个登录的模态框，为了是用户的体验度更好，登录功能没有做成单独的页面，然后用户可以输入用户名或邮箱和密码进行登录操作

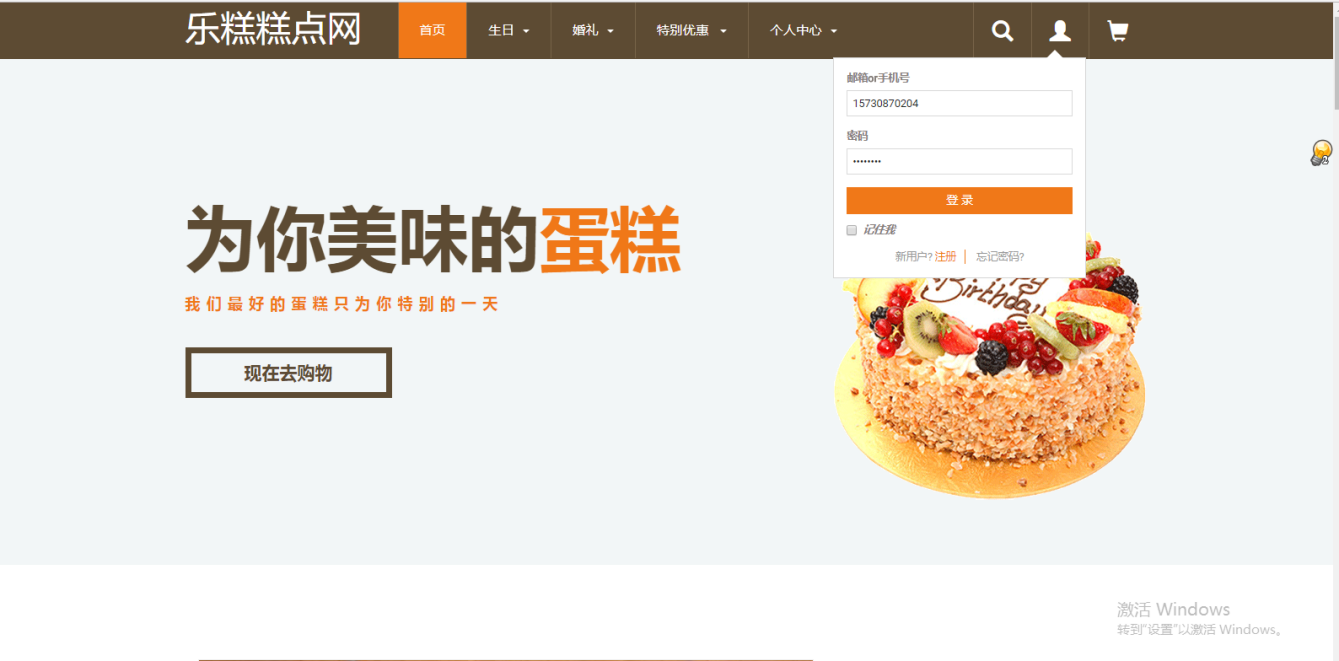


图4-10 系统登录界面图

代码如下：

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

public String login(Model model,

@RequestParam(value = "username") String name,

@RequestParam(value = "password") String password) {

//判断参数是否为空

if (name == null || name.equals("") || password == null || password.equals("")) {

model.addAttribute(Constant.STATUS, Constant.STATUS\_CODE\_ERROR);

model.addAttribute(Constant.MESSAGE, Constant.MESSAGE\_STR\_ERROR);

return "login";

}

//构建登录数据

UsernamePasswordToken token = new UsernamePasswordToken(name, password);

Subject subject = SecurityUtils.getSubject();

try {

subject.login(token);

if (subject.isAuthenticated()) {

//设置 登录信息

String principal = (String) subject.getPrincipal();

session.setAttribute(Constant.USER\_INFO, adminService.getUserByPhone(principal));

return "redirect:/";

}

} catch (IncorrectCredentialsException e) {

model.addAttribute(Constant.MESSAGE, "登录密码错误");

} catch (ExcessiveAttemptsException e) {

model.addAttribute(Constant.MESSAGE, "登录失败次数过多");

} catch (LockedAccountException e) {

model.addAttribute(Constant.MESSAGE, "帐号已被锁定");

} catch (DisabledAccountException e) {

model.addAttribute(Constant.MESSAGE, "帐号已被禁用");

} catch (ExpiredCredentialsException e) {

model.addAttribute(Constant.MESSAGE, "帐号已过期");

} catch (UnknownAccountException e) {

model.addAttribute(Constant.MESSAGE, "帐号不存在");

} catch (UnauthorizedException e) {

model.addAttribute(Constant.MESSAGE, "您没有得到相应的授权！");

}

model.addAttribute(Constant.STATUS, Constant.STATUS\_CODE\_SUCCESSED);

return "login";

}

4.3.3 注册模块

用户在浏览网站之后想要购买商品必须得注册一个用户，然后登录后就可购买了，主页页面包括个人信息和登录信息，个人信息里面需要填写姓名、昵称、性别、邮箱等字段，登录信息需要输入用户的密码，让后可以点击注册按钮进行注册。



图4-11 注册界面

代码如下：

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

@ResponseBody

public Map addUser(@RequestParam(value = "username")String username,

@RequestParam(value = "password")String password){

System.out.println(username+password);

Map map=new HashMap();

if (username==""||username==null||password==""||password==null){

map.put(Constant.STATUS,Constant.FAIL);

map.put(Constant.MESSAGE,"用户名或者密码不能为空！");

return map;

}

//判断电话号码是否存在

if (adminService.isExistPhone(username)) {

map.put(Constant.STATUS, Constant.FAIL);

map.put(Constant.MESSAGE, "对不起，该电话号码已经注册了！");

return map;

}

if (adminService.addUser(username, password) > 0) {

map.put(Constant.STATUS, Constant.SUCCESS);

map.put(Constant.MESSAGE, "恭喜你，注册成功！");

} else {

map.put(Constant.STATUS, Constant.FAIL);

map.put(Constant.MESSAGE, "对不起，注册失败，请重新注册！");

}

return map;

}

4.3.4 购物车模块

该模块主要为注册用户将自己喜欢的商品添加到购物车里面，如果用户想要买多个该商品可以在数量上选择自己想要的数量，用户如果不想要改商品可以将该商品移除购物车，用户还可将自己的购物车清空，在该页面用户还可以看到要购买该商品的小计，并且还会显示商品的总计为多少，让后用户可以点击去结算按钮进行结算，也可以点击继续购物按钮去挑选自己心仪的商品。

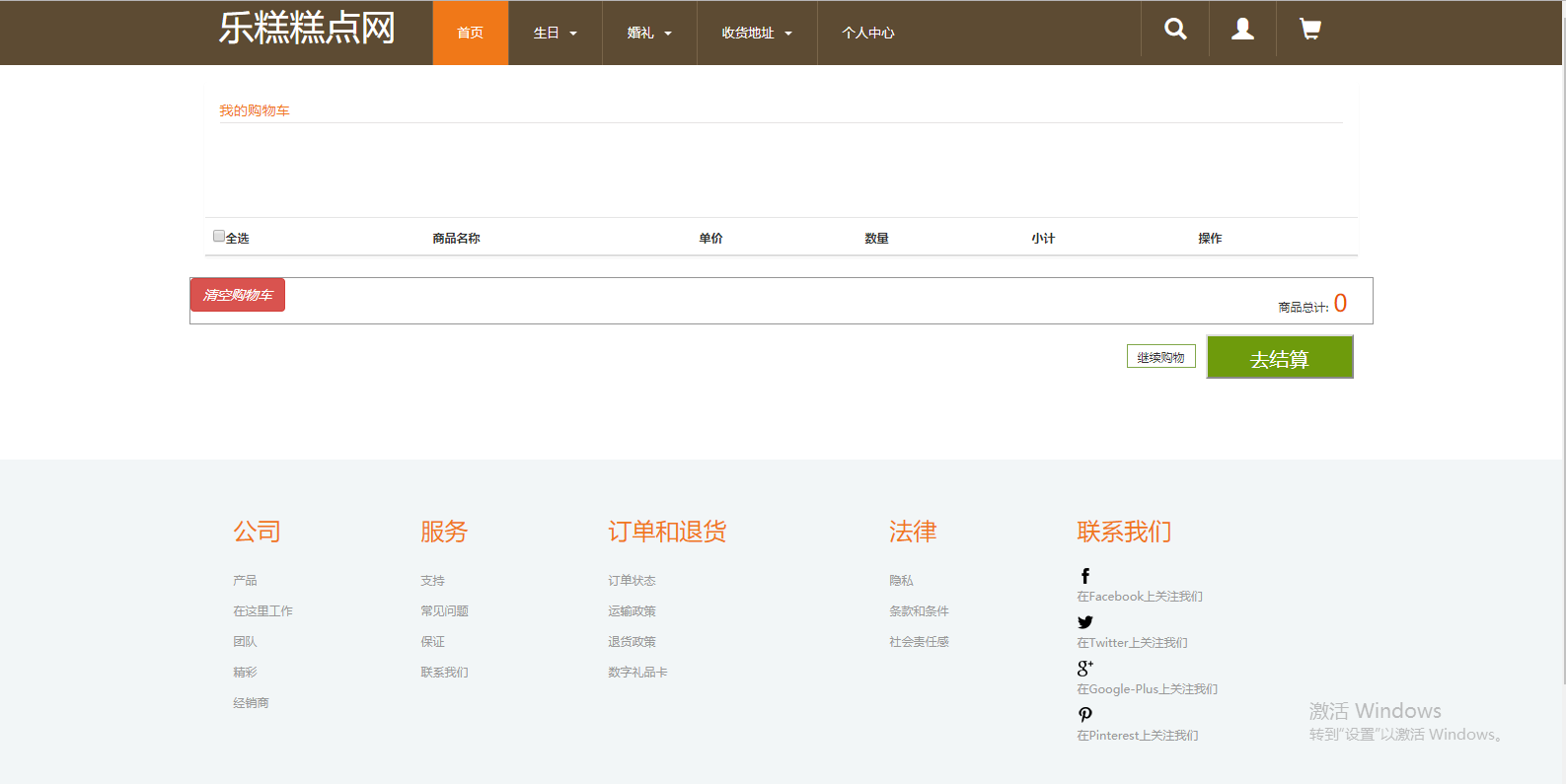


图4-12 购物车界面效果

代码如下

@Override

public Cart addCart(String userid, String proid, Integer num) {

//准备数据

Cart cart = new Cart();

Prodution prodution = produtionMapper.selectByPrimaryKey(proid);

if (prodution == null) {

return null;

}

//填充数据

cart.setCartid(RandomUtils.randomId10());

cart.setProid(proid);

cart.setUserid(userid);

cart.setNum(num);

cart.setCartcreatetime(new Date());

cart.setCartupdatetime(new Date());

cart.setCartmark(Constant.MARK\_CODE\_OK);

//计算小计

Double total = num \* prodution.getPrice();

cart.setTotal(total);

//添加到购物车

int reuslt = cartMapper.insert(cart);

if (reuslt > 0) {

return cart;

} else {

return null;

}

}

4.3.5 收货地址功能

该页面由购物车页面的去结算按钮跳转而来，主要是让用户输入收货人地址的一些信息，比如收货人姓名，收货人省份、收货人详细地址、收货人电话等信息，用户可以点击保存联系人信息将信息存入后台数据库里面，用户还可以添加多个联系人方式，下面还会有用户选购的商品信息等待着用户去确认。然后用户输入完收货人信息后可以点击提交订单按钮。如图4-6所示。



图4-13 收货地址界面

代码如下

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

public String toPrepareOrder(Model model, @RequestParam(value = "cartid", required = false) String[] cartids) {

//判断有没有选择商品

if (cartids == null) {

model.addAttribute(Constant.STATUS, Constant.STATUS\_CODE\_ERROR);

model.addAttribute(Constant.MESSAGE, Constant.MESSAGE\_STR\_NULL\_PARAM);

return "redirect:/cart/all";

}

//拿到用户信息

User user = (User) session.getAttribute(Constant.USER\_INFO);

model.addAttribute(Constant.USER\_INFO, user);

//拿到地址信息

List<Address> addressList = addressService.getAddressByUserId(user.getUserid());

model.addAttribute(Constant.MODEL\_KEY\_USER\_ADDRESSES, addressList);

//添加商品信息

List<CartInfo> cartInfos = cartService.getCartInfoByCartIds(cartids);

model.addAttribute(Constant.MODEL\_KEY\_CART\_INFOS, cartInfos);

//应付金额

Double total = cartService.countTotalByCartInfos(cartInfos);

model.addAttribute(Constant.MODEL\_KEY\_CART\_TOTAL, String.format("%.2f", total));

return "indent";

}

4.3.6 订单支付页面

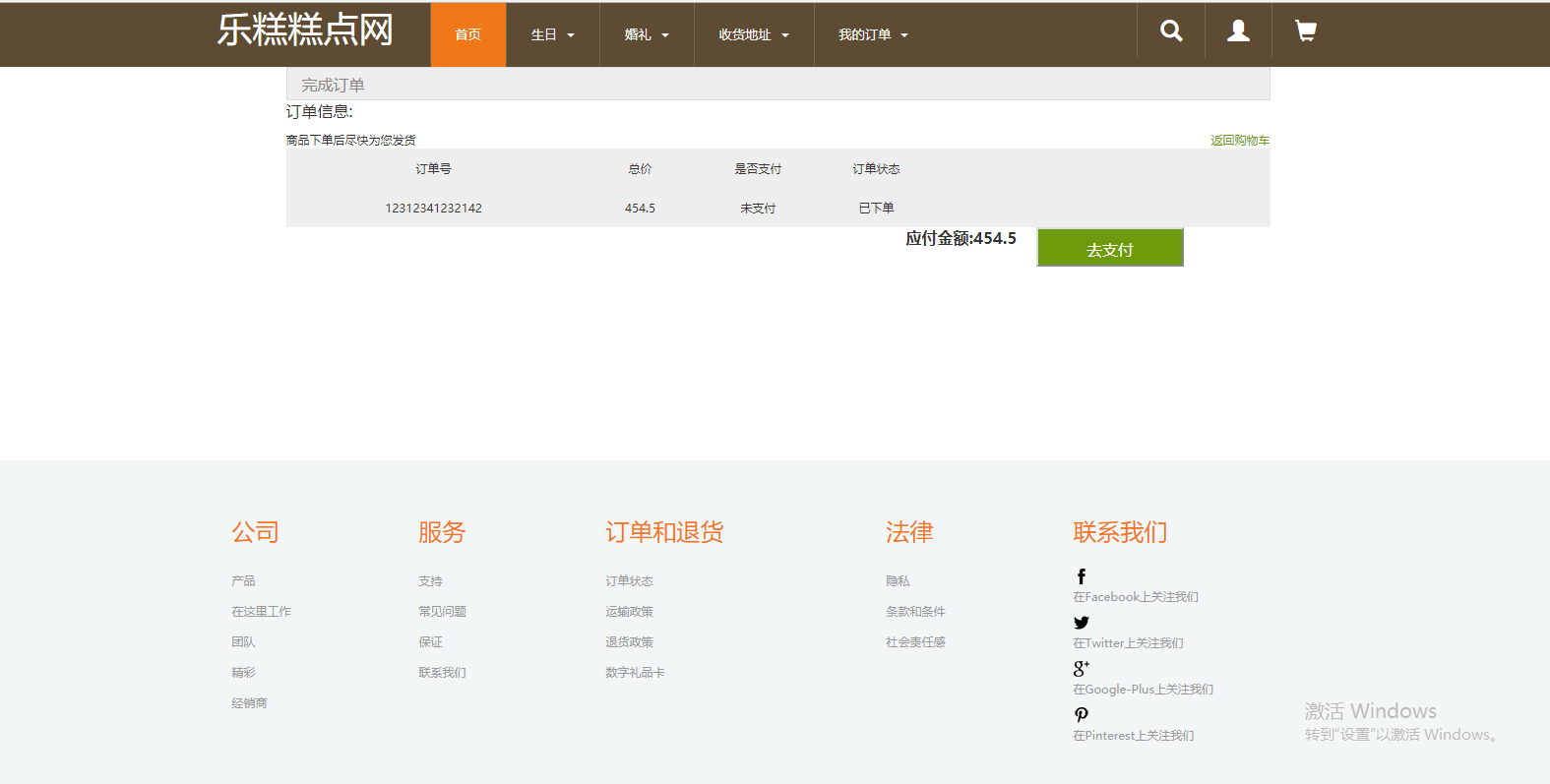
用户从上面的点击提交订单后系统会生成一个订单信息，上面会显示订单号，总价，是否支付，订单状态和应付金额等信息，在页面如果用户不想去支付可以点击返回购物车，如果用户觉得没有问题或则说没有什么其他想够买的商品就可以去点去支付按钮进行支付。如图4-7所示。

图4-14 支付订单页面

代码如下

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

public String createOrder(Model model,

@RequestParam(value = "addid") String addid,

@RequestParam(value = "cartid") String[] cartids) {

//当前登录用户

User user = (User) session.getAttribute(Constant.USER\_INFO);

model.addAttribute(Constant.USER\_INFO, user);

//判断是否登录

if (user == null) {

model.addAttribute(Constant.STATUS, Constant.STATUS\_CODE\_NOT\_LOGIN);

return "indent";

}

//添加订单

Order result = orderService.addOrderByAddIdAndCartIds(addid, cartids, user.getUserid());

if (result != null) {

model.addAttribute(Constant.STATUS, Constant.STATUS\_CODE\_SUCCESSED);

model.addAttribute(Constant.MESSAGE, "订单添加成功！");

//删除购物车

cartService.deleteChoose(cartids);

//应付金额

model.addAttribute(Constant.MODEL\_KEY\_CART\_TOTAL, String.format("%.2f", result.getTotalprice()));

model.addAttribute(Constant.MODEL\_KEY\_ORDER, result);

return "order";

} else {

model.addAttribute(Constant.STATUS, Constant.STATUS\_CODE\_ERROR);

model.addAttribute(Constant.MESSAGE, "订单添加失败!");

return "indent";

}

}

4.3.7 我的订单中心

由于用户可以很多的订单，然后用户可以点击我的订单中心查看自己完成的和未完成的订单，我的订单中心里展示的是商品图片，由于一个订单可能有多个商品，所以商品图片有多个，订单号，数量，创建时间，收货人，金额，状态，状态为待支付和已支付，如果是待支付的话用户还可以在操作里面点立即支付。如图

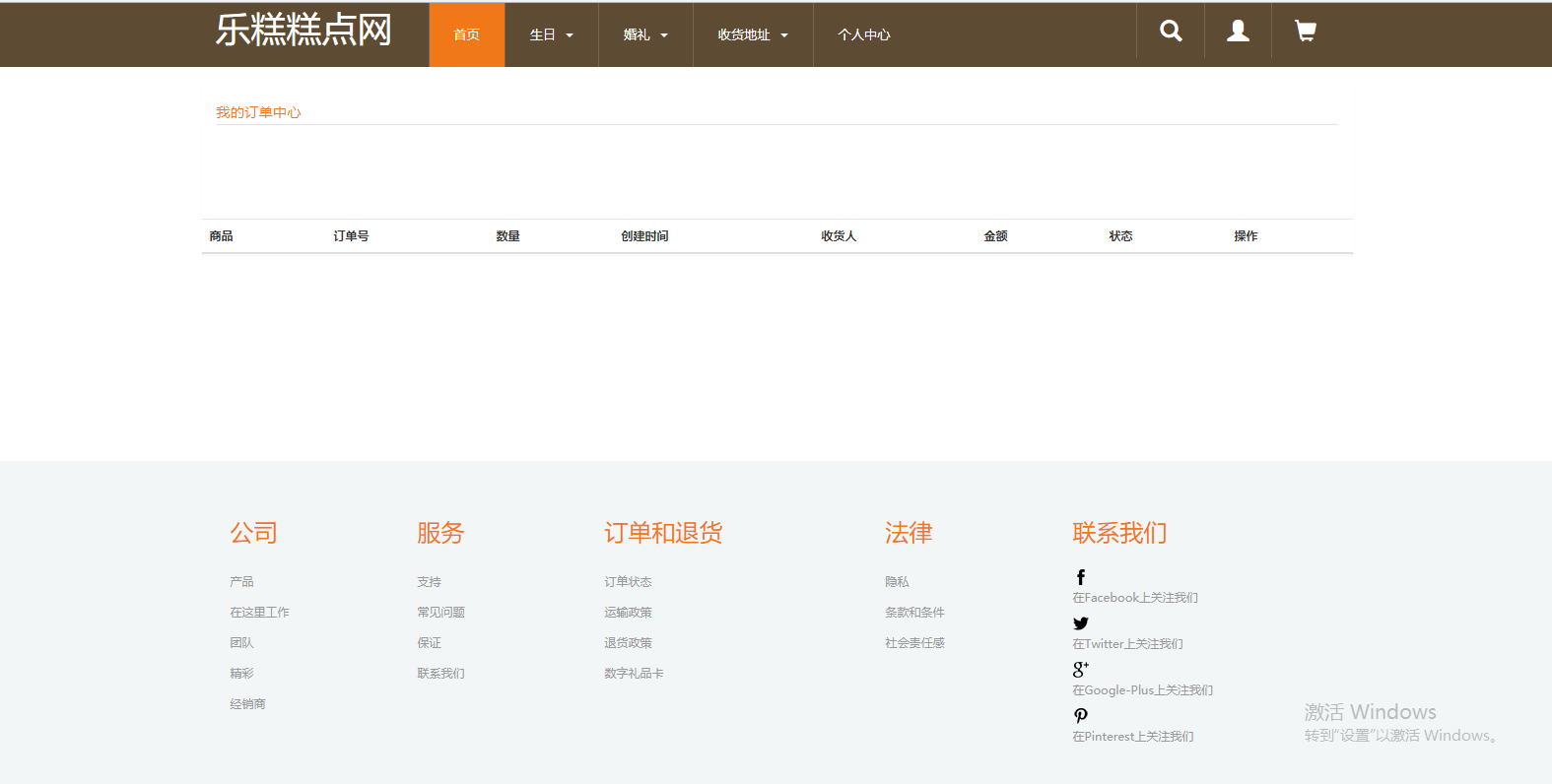
****

图4-15 订单信息页面

主要代码如下：

@RequestMapping(value = "showorderinfo" ,method = RequestMethod.GET)

public String showOrderInfo(Model model){

//显示个人信息

User user=(User) session.getAttribute(Constant.USER\_INFO);

if (user==null){

//重定向到 login

return "redirect:/user/login";

}

//显示订单信息

List<OrderInfo> orderInfos=orderService.showOrderInfo(user.getUserid());

int cartNum=(int)session.getAttribute(Constant.CART\_NUM);

model.addAttribute(Constant.ODERINFO\_LIST,orderInfos);

model.addAttribute(Constant.USER\_INFO,user);

model.addAttribute(Constant.CART\_NUM,cartNum);

return "orderinfo";

}

4.3.8 商品管理

该模块为管理员操作的界面，主要用户管理对商品信息的管理，由于用户没有权限对商品信息的操作，所以该模块主要有四个功能，添加商品信息功能，修改商品信息功能，查询商品信息功能和删除商品信息功能。其中添加商品信息主要需要输入商品的图片、商品名称、价格、颜色以及尺寸等属性。

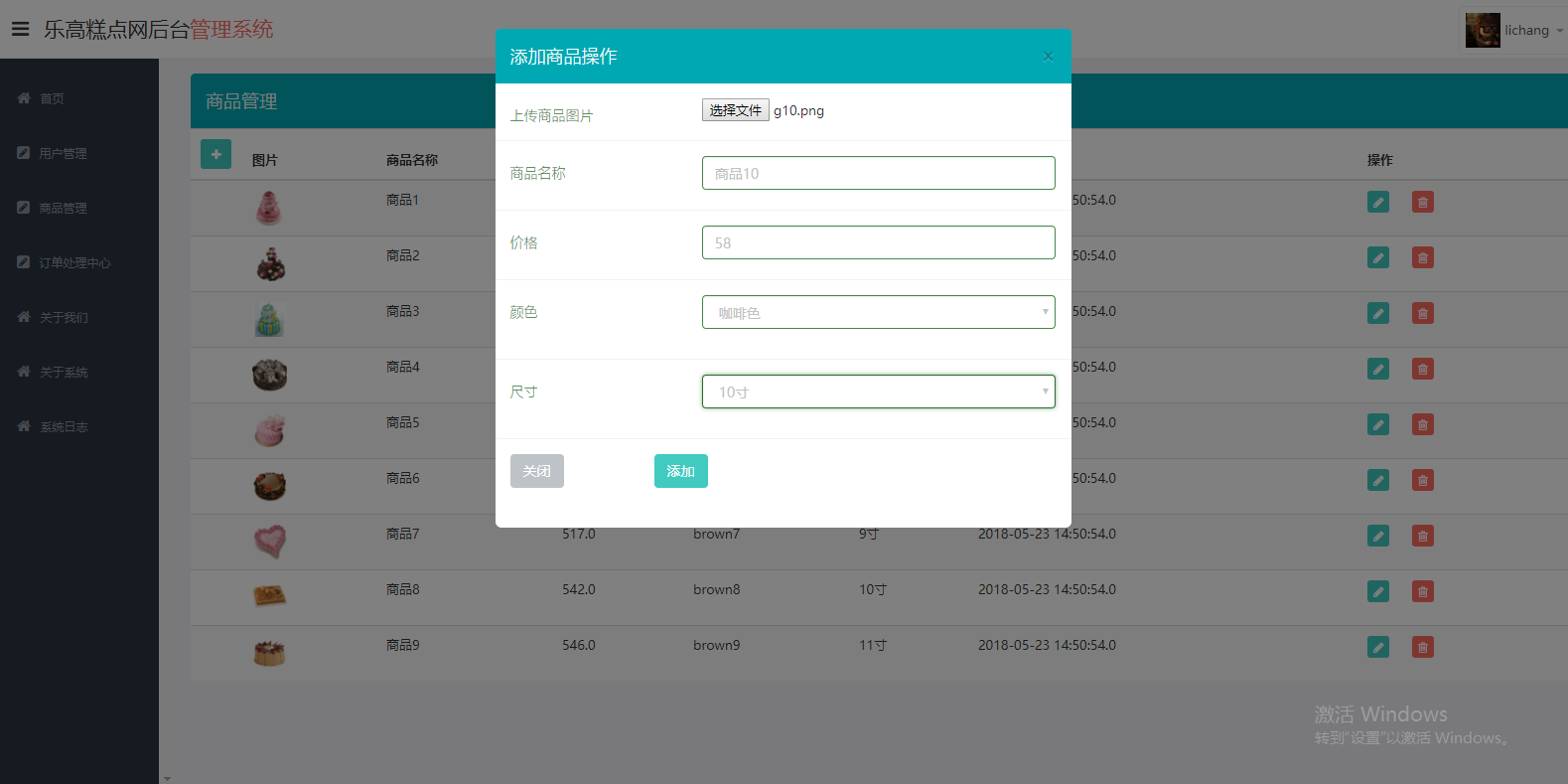


图4-16 商品添加页面

主要代码如下：

@RequestMapping(value = "/add\_product",method = RequestMethod.POST)

@ResponseBody

public Map addProduct(@RequestParam(value = "image") MultipartFile image,

@RequestParam(value = "pname") String pname,

@RequestParam(value = "price") String price,

@RequestParam(value = "color") String color,

@RequestParam(value = "weight") String weight){

Map map=new HashMap();

System.out.println(weight);

try {

String imageName = FileUploadUtil.uploadHeadImage(IMAGEUPLOADPATH,image);

if (imageName != null) { //判断文件是否存在文件夹里

Product product=new Product();

product.setPname(pname);

product.setPrice(Double.valueOf(price));

product.setColor(color);

product.setWeight(weight);

product.setImage("/upImage/"+imageName);

product.setStatdate(DateUtil.date2Str(new Date(),"yyyy-MM-dd HH:mm:ss"));

product.setFlag(1);

productService.addProduct(product);

map.put(STATUS, 1);

map.put(MESSAGE, "添加商品成功！！");

return map;

}

} catch (IOException e) {

map.put(STATUS, 0);

map.put(MESSAGE, "添加商品失败！！");

}

return map;

}

4.3.9 订单处理中心

该模块为管理员处理订单的界面，主要对用户生成的订单进行发货处理，主要展示的订单字段为，商品图片，商品名称，数量，金额，支付状态，发货状态。在订单的支付状态为已支付状态下管理员可以对该条订单的发货状态进行处理，处理状态分为已发货、正在处理两种状态。

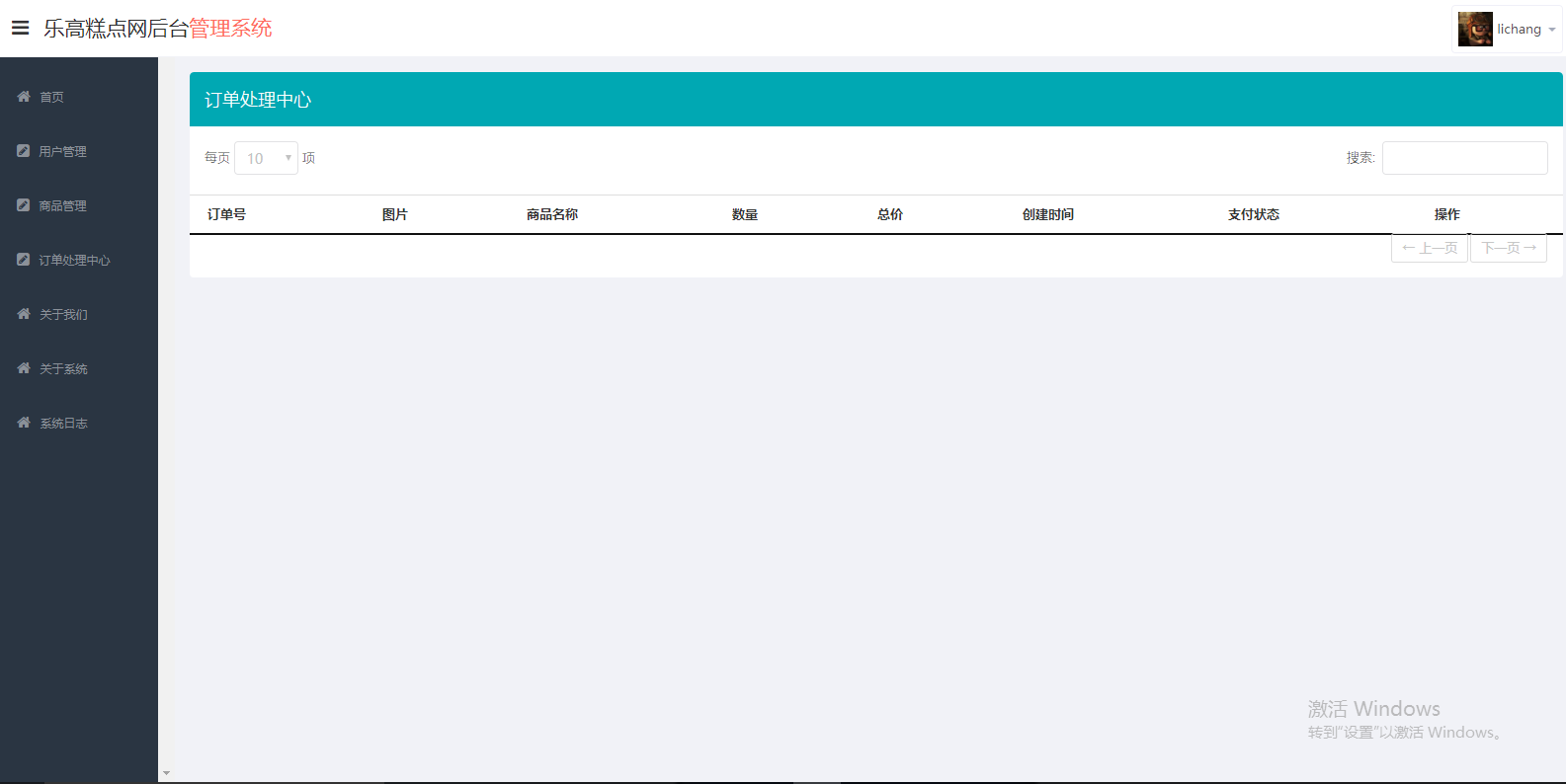


图4-17 订单处理页面

第5章 软件测试与结果分析

我们在真正项目的开发过程中，首先得要有自己的开发团队人员进行测试，因为任何一个软件不可能是绝对的完美，或多或少会有一些意想不到的bug,当我们把项目交付给第三方的时候还有第三专业的人进行测试，其实测试这一环节是最为重要的，如果不把测试做好，然后系统交由用户，当用户测出软件的漏洞之后，一方面可能对我们的系统造成很大的危害，另一方面会失去用户对我们软件的安全信任，所以做好软件测试必不可少。

5.1 测试的目的和原则

5.1.1 测试的目的

测试的目的在于找到或则发现我们在开发中没有发现的错误，并且找到错误出现的地方以及该错误由什么产生的，最终的目的是让我们的软件的健壮性更加的强，其实一个好的测试在于它的测试用户找到了至今为止没有发现的错误，一个成功的测试在于测试出了至今为止没有发现的错误测试，但是并不一定没有测试出错误就说明测试没有价值。

5.1.1 测试的原则

测试的原则应该是尽快的尽早的不断的测试，测试用例应该需要测试数据和测试预期结果想匹配，如果测试的数据和预期结果不一样，说明我们程序有问题，测试的另一个原则是程序员应该避免测试自己开发的程序，还有一个是就是用极端的方法，输入一个合理的数据和输入一个不合理的数据进行测试，当我们测试完之后应该对每一个测试用户的结果做一个全面的检查。

5.2 测试计划

一个安全且良好的软件应该能过够各种各样的测试，包括功能测试和性能测试，本网站主要测试的是功能测试，软件测试的方法分为白盒测试和黑盒测试，但对于动能测试我们主要采用黑盒测试即可，软件测试计划表如下表所示。

表5-1 系统测试计划表

|  |  |
| --- | --- |
| 目标 | 按照需求分析完成各个功能测试 |
| 内容 | 登录模块，注册模块，购物车模块，收获地址模块，商品管理模块，订单信息模块 |
| 技术 | 白盒测试和黑盒测试，采用黑盒测试 |
| 前提 | 软硬件配置完成 |
| 完成标准 | 主要功能都能实现并通过 |
| 特殊事项 | 不同用户权限问题 |

5.3 测试过程

本文对系统的测试主要是确定系统的运行能力，确保系统满足初期开发功能，并寻找系统的不足之处。具体测试环境如表5-2所示：

**表5-2环境介绍表**

|  |  |  |
| --- | --- | --- |
| 测试系统 | 基于SSM的乐高糕点网站 | 测试硬件环境：  普通笔记本电脑 |
| 应用服务器 | Tomcat8.0 |
| 浏览器 | 谷歌chrome |
| 数据库 | MySQL5.7 |
| 测试网络环境 | 无特殊要求 |
| 测试方法 | 黑盒测试 |

5.3.1 测试用例

对于了乐高糕点网站测试主要采用黑盒测试，测试的主要用例为：注册模块，登录模块，添加购物车模块，添加收获地址模块，商品管理模块，具体如表5-3至表5-7所示：

表5-3游客注册模块测试用例

|  |  |
| --- | --- |
| 用例名称 | 游客注册功能测试 |
| 用例ID | T1 |
| 基本描述 | 游客注册主要是将游客的电话号码与后台数据库匹配 |
| 测试方案 | 先用一个数据库已经存在的电话号码注册，再用一个数据库不存在的电话号码注册。 |
| 测试数据 | 输入电话15730870204；输入电话15110148779 |
| 测试结果 | 第一次注册失败，返回电话号码已经被注册消息，第二次注册成功，并用登录功能进行登录测试 |
| 测试人 | 李常 |

表5-4注册用户登录模块测试用例

|  |  |
| --- | --- |
| 用例名称 | 注册用户登录功能测试 |
| 用例ID | T2 |
| 基本描述 | 用用户输入的电话号码和密码与后台数据库匹配，如果在数据库里能找到这条记录则为登录成功，否则登录失败。 |
| 测试方案 | 先用电话存在并且密码不正确的字段与后台数据库匹配，再用电话号码不存在和密码正确的字段与后台数据库匹配，最后用电话号码存在且密码正确的字段与后台数据库匹配。 |
| 测试数据 | 15110148779,123456；13110143237,666666；15110148779,666666 |
| 测试结果 | 第一次登录失败；第二次登录失败，第三次登录成功。 |
| 测试人 | 李常 |

表5-5添加购物车模块测试用例

|  |  |
| --- | --- |
| 用例名称 | 添加购物车模块功能测试 |
| 用例ID | T3 |
| 基本描述 | 测试用户是否是在登录状态下添加商品，测试用户是否修改购物车数量总金额是否变化，测试用户是否能将商品从购物车移除 |
| 测试方案 | 用户在没有登录状态下将商品添加的购物车，用户在登录状态下将商品添加到购物车，用户修改购物车商品的数量，用户将商品从购物车移除。 |
| 测试数据 | 1.在未登录状态添加商品到购物车，2.在登录状态添加商品到购物车，3.修改购物车商品的数量从1到2，4.移除一个购车的商品。 |
| 测试结果 | 1.添加失败，提示请先登录消息，2.添加成功，购物车有该商品，3.总金额从23.8变47.6。4.该商品在购物车列表不存在。 |
| 测试人 | 李常 |

表5-6收获地址模块测试用例

|  |  |
| --- | --- |
| 用例名称 | 收获地址模块功能测试 |
| 用例ID | T4 |
| 基本描述 | 录入收入人地址主要测试的是收货人的电话号码格式是否正确 |
| 测试方案 | 输入一个格式正确的电话号码，输入一个非法格式的电话号码。 |
| 测试数据 | 15110148779；1573087020422 |
| 测试结果 | 正确保存；提示电话号码格式错误 |
| 测试人 | 李常 |

表5-7商品管理模块测试用例

|  |  |
| --- | --- |
| 用例名称 | 商品管理模块功能测试 |
| 用例ID | T5 |
| 基本描述 | 测试能否将一个商品添加保存在数据库，测试能否修改一个商品的信息，测试能否删除一个商品。 |
| 测试方案 | 添加一个商品的信息，然后再商品列表看是否有该商品；修改一个商品信息然后在商品列表看信息是否修改；删除一个商品，然后在商品列表是否还有该商品。 |
| 测试数据 | 添加商品名为康奈尔的商品；修改商品ID为1123的价格为87；删除商品ID为1123的商品。 |
| 测试结果 | 添加成功，商品列表有该商品；商品列表商品1123的价格由52变为87；商品列表没有商品ID为1123的商品。 |
| 测试人 | 李常 |

5.4 测试结论

通过测试，网站大体功能与预期期望一致，测试中我们发现了许多的小bug，然后通过不断的完善，项目整体上没有什么很大的问题，由于测试的方式有限，可能还有很多地方没有测试到位，测试其实是一门很高深的课程，就好比当今的黑客们，他们总能找到系统的漏洞，然后进行攻击，所以，对于普通的软件或则网站而言，虽然安全性并没有那么高，但是我们决不能允许出现逻辑上的错误以及软件自身上的错误。

结 论

本次毕业主要是模拟当前主流的电商网站，经过此次毕业设计对当前电商网站的业务逻辑有了深刻的理解，并且对电商的未来前景充满了期待，虽然做这次毕设的时间很短暂，但还是学到了很多新的技术，比如当前新型的流行的SpringBoot框架，虽然它不是一个新的技术，只是对spring生态系统进行了一系列的封装，减少了开发人员的配置，提高了开发人员的开发效率，但是，通过此次毕设也只到了凡事都是有利也有弊的，SpringBoot虽然封装的很好了，但是我们队spring生态系统的内部实现原理却了解的不是那么的深入，作为大学生的我们，我不是不应该只知道知其然，还应该知其所以然。

然而，SSM框架已经成为当今中小企业开发的主流框架，所以通过此次毕设的实现有助于马上要毕业的我们找工作，也让我们把之前的学过的java基础得到了进一步的实践，当然，在开发过程中也是遇到了不少的问题和困难，对于表的设计考虑的不是那么的全面，有些业务逻辑想的不是很周到，但是这难不倒一个程序员的开发斗志，在通过网上的资料查询也是能够得到解决，还有就是在这次写论文的过程中，发现以前学过的很多软件开发流程不是很清楚，比如，什么时候改写需求分析，什么时候改写概要设计，需求分析里该写什么，ER图该画到什么位置等等问题，通过这次论文的书写得到了全面的巩固，总之自己还是有很多不足的地方，还有很多要学习的地方，我们不应该不遗余力，厚积薄发。

致 谢

乐高糕点网站的开发，不仅让我将之前学过的知识实践了一遍，也让我对现在新的技术得到了认识，然而大学已经将要接近尾声了，对于自己来说总是有那么一丝的遗憾，遗憾的是用到的一些技术之前老师有讲过，但是自己没有认真听过，但还是得感谢大学教过我的每一位老师，是他们传播的知识才让我能把这次毕设完满的完成。在这特别得感谢我的指导老师孙亚楠老师，从论文的选题，开题报告，程序的实现，到最后的论文书写，她一直陪伴着我们，在毕设的实现上给了我跟多的指导，在我代码实现上遇到困难的时候给予我解决方案上的指导。总是很负责人的监督我们完成我们的毕设，再次真心的感谢你，老师您辛苦了。没有你的细心帮助我真的完不成这次毕业设计。

最后祝愿所有的老师工作顺利，所有的同窗好友前途似锦。

参考文献

[1] 刘升华.Java从入门到实践[M].北京：清华大学出版社，2009，7~78.

[2] 林信良.Spring技术手册[M].北京：电子工业出版社，2006，12~22.

[3] 陆荣幸，郁洲，阮永良等.J2EE平台上MVC设计模式的研究与实现[J].计算机应用研究，2003，03(03):23~46.

[4] 邹竹彪.JSP宝典[M].北京：电子工业出版社，2007，56~108.

[5] 耿祥义，张跃平.JSP实用教程[M].北京：清华大学出版社，2003，20~78.

[6] 福塔.SQL必知必会[M].北京：人民邮电出版社，2007，45~75.

[7] 张海藩.软件工程导论[M].北京;清华大学出版社，2003,17~98.

[8] 王珊.数据库系统概论[M].北京：高等教育出版社，2006，43~90.

[9] 冯晓强，程晓昕.基于MVC模式的网上购物系统的设计与实现[J].现代计算机(专业版)，2009，09(07):11~50.

[10] 毕广吉. Java程序设计实例教程[M].北京：冶金工业出版社，2007，13~69.

[11] James Turner，Kevin Bedell.Struts Kick Start[M].SAMS，2004，23~30.

[12] Ron Patton.Software Test[M].SAMS，2008，12~23.

[13] Mateosian R. PHP and MySQL for dynamic web sites: Visual QuickProguide [J].IEEE, 2003,23(5):33~67.

[14] Cornel，Horstmann.Core Java[M].Prentice\_Hall,1997,22~41.

[15] 肖汉.软件测试[M]. 北京：电子工业出版社，2013：56-287.

[16] 朱少民.软件测试[M]. 北京:人民邮电出版社，2009：103-152.

[17] 张保军. 综合业务系统软件测试管理体系的建立[J]. 中国金融电脑，2003年第11期.

[18] 赵斌. 软件测试技术经典教程（第二版）[M]. 北京:科学出版社，2016：153-205.

[19] 黄莹. 软件测试技术与测试管理[J]. 工业控制计算机，2003年16卷第5期.

[20] 许静，陈宏刚，王庆人. 软件测试方法简述与展望[J]. 计算机工程与应用，2003.13.

附 录

1．环境配置 ：

（1）硬件环境

CPU：最低Pentium II以上，提倡PIII以上

内存：最低128M以上，提倡256M

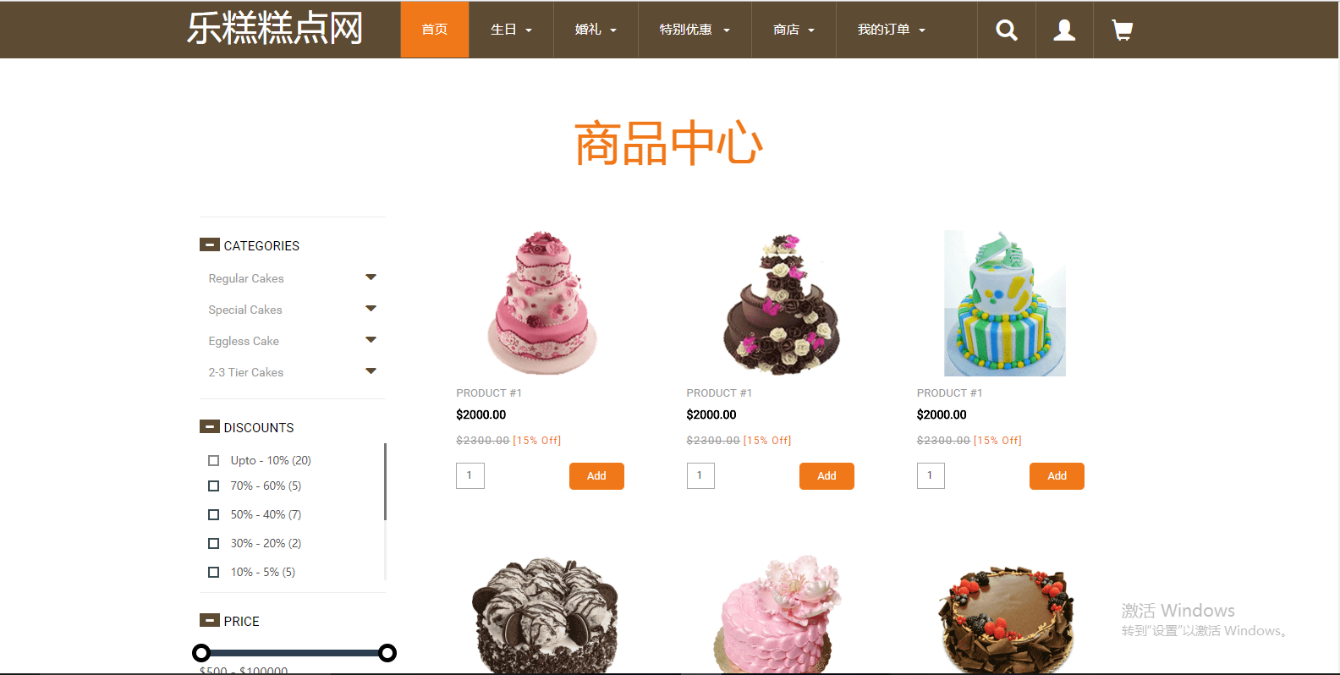
硬盘：最低500M以上的可用硬盘安装空间

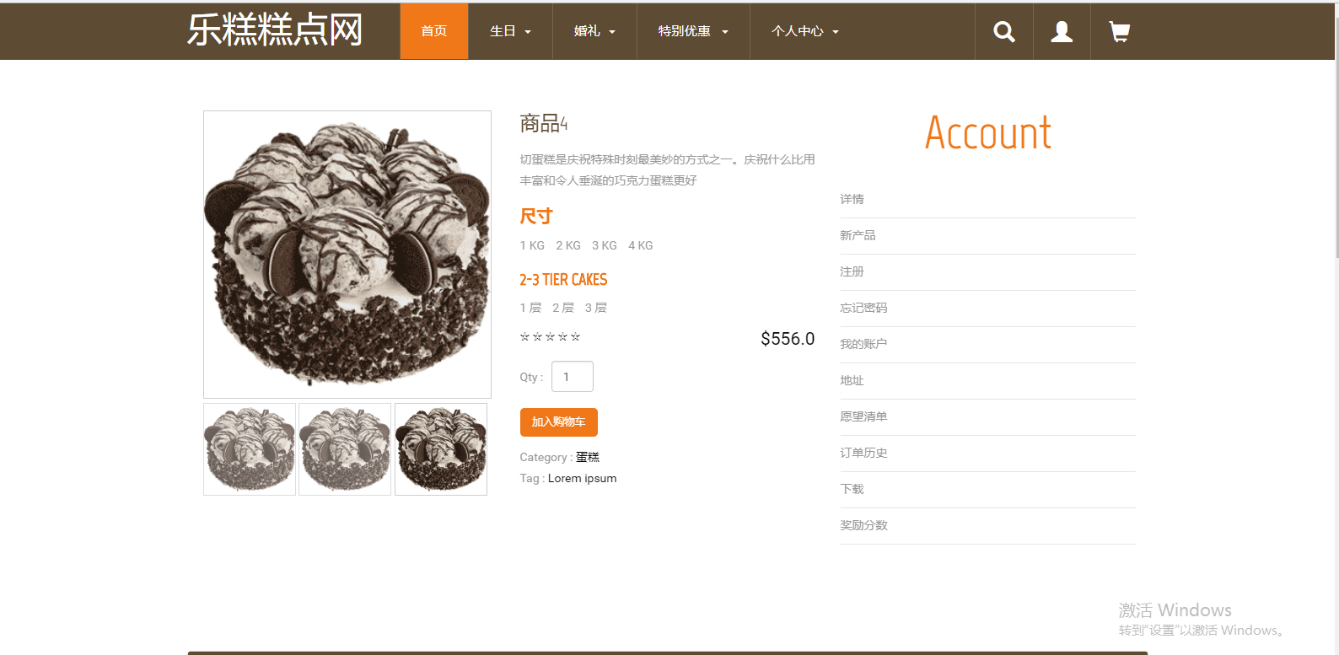
（2）软件环境

可在Windows,Linux等多个操作系统上运用。Mysql5.1版本以上。安装JDK1.5以上版本。Maven3.0版本以上。

系统部分运行界面

附图1 个人信息页面

附图2 商品列表页面



附图3 商品详情页面

附图4 后台管理主界面

pom.xml配置文件:

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.douguo</groupId>

<artifactId>springbootTest</artifactId>

<version>1.0-SNAPSHOT</version>

<!-- 定义公共资源版本 -->

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.5.8.RELEASE</version>

<relativePath />

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncodin>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<!--shiro-->

<dependency>

<groupId>org.apache.shiro</groupId>

<artifactId>shiro-spring</artifactId>

<version>1.4.0</version>

</dependency>

<dependency>

<groupId>org.apache.shiro</groupId>

<artifactId>shiro-ehcache</artifactId>

<version>1.2.5</version>

</dependency>

<dependency>

<groupId>com.github.theborakompanioni</groupId>

<artifactId>thymeleaf-extras-shiro</artifactId>

<version>1.2.1</version>

</dependency>

<!--/shiro-->

<!-- redis -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-redis</artifactId>

</dependency>

<dependency>

<groupId>org.mybatis.spring.boot</groupId>

<artifactId>mybatis-spring-boot-starter</artifactId>

<version>1.1.1</version>

</dependency>

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>druid-spring-boot-starter</artifactId>

<version>1.1.2</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<!-- /redis -->

<!-- 热部署模块 -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<optional>true</optional> <!-- 这个需要为 true 热部署才有效 -->

<scope>true</scope>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.github.servanter</groupId>

<artifactId>netsfjson-support-spring</artifactId>

<version>1.0.0</version>

</dependency>

<!--不检查thymelef 的语法 配置spring.thymeleaf.mode=LEGACYHTML5-->

<dependency>

<groupId>net.sourceforge.nekohtml</groupId>

<artifactId>nekohtml</artifactId>

<version>1.9.21</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<configuration>

<fork>true</fork> <!-- 如果没有该配置，devtools不会生效 -->

</configuration>

</plugin>

</plugins>

</build>

</project>