基于Java的快点餐系统的设计与实现

Design and Implementation of Quick Meal System Based on Java

学院（部）：计算机与信息工程学院

专 业：通信工程

姓 名：黄照光

学 号：1606735067

年级班级 ：16通信二班

指导教师 ：吴冉老师

完成日期： 年 月

目 录

[[摘 要] 3](#_Toc26722)

[1 绪论 1](#_Toc30948)

[1.1 引言 1](#_Toc13184)

[1.2点餐系统的开发背景 1](#_Toc21619)

[1.3 点餐系统的开发意义 1](#_Toc32697)

[1.4 网上点餐的市场趋势 1](#_Toc4663)

[1.5 网上点餐的市场优势 2](#_Toc8636)

[2 点餐系统相关的技术介绍 3](#_Toc21088)

[2.1 SpringBoot 3](#_Toc17997)

[2.2 JSP 3](#_Toc11947)

[2.3 Shiro 3](#_Toc26648)

[2.4 Redis 4](#_Toc26475)

[2.5 Git 4](#_Toc7440)

[2.6 MySql 4](#_Toc32244)

[2.7 Jmeter 4](#_Toc16322)

[3 点餐系统需求的调查以及实现 5](#_Toc20249)

[3.1 需求分析 5](#_Toc23621)

[3.2功能模块设计 5](#_Toc5852)

[3.2.1 商家管理模块 5](#_Toc17842)

[3.2.2 用户点餐模块 6](#_Toc20113)

[4 数据库设计 7](#_Toc8259)

[4.1概念结构设计 7](#_Toc8433)

[4.2逻辑结构设计 7](#_Toc19853)

[5 功能模块的实现 11](#_Toc19445)

[5.1登录模块的实现 11](#_Toc21365)

[5.2首页模块的实现 12](#_Toc18042)

[5.3商品管理的实现 14](#_Toc23497)

[5.4订单管理功能的实现 15](#_Toc28658)

[5.5修改密码功能的实现 16](#_Toc6566)

[5.6线上付款功能的实现 18](#_Toc688)

[6 系统测试 20](#_Toc2835)

[6.1 功能测试 20](#_Toc1430)

[6.2 性能测试 21](#_Toc123)

[总结 22](#_Toc8235)

[参考文献 23](#_Toc24813)

[致谢 24](#_Toc3170)

基于Java的快点餐系统的设计与实现

[摘 要]：随着社会的飞速发展，互联网凭借它普及性广，便捷度高的特点不断地改变着人们的生活习惯与行为方式，也在逐步迭代我们身边的每一个行业。传统模式的餐厅需要大量服务员以满足顾客的点餐需求，无法做到顾客随心点，服务随时到的需求，并且顾客无法及时有效的对本次消费进行评价，顾客与商家的交互十分受限，在互联网的今天用户普遍对于消费进行线上支付，但是大部分传统的餐厅无法满足顾客的这一新的行为习惯，导致客源被大量的掠夺。本次毕业设计后端主要使用SpringBoot框架，前端使用jsp技术，shiro作为登录权限认证，Tomcat作为中间件，redis做缓存，Jmeter做压力测试，Git实现版本控制和MySQL数据库完成点餐系统的设计和实现。系统包括了用户注册登录、卖家添加/修改商品、买家点餐、付款、商品配送、买家留言、修改密码等主要功能。

[关键词]：SpringBoot；jsp；shiro；redis；Git；MySQL

Design and Implementation of Quick Meal System Based on Java

**Abstract:** With the rapid development of society, the Internet is changing people's living habits and behavior with its wide popularity and high convenience. It is also gradually iterating every industry around us. The traditional restaurant needs a large number of waiters to meet the demand of customers' ordering, which can't meet the demand of customers' free order and service at any time, and customers can't evaluate the consumption timely and effectively. The interaction between customers and businesses is very limited. In today's Internet, users generally pay for consumption online, but most of the traditional restaurants can't meet customers This new habit of behavior leads to a large number of tourists being plundered. The back-end of this graduation project mainly uses the spring boot framework, the front-end uses JSP technology, Shiro as the login authority authentication, Tomcat as the middleware, redis as the cache, GIT to achieve version control and MySQL database to complete the design and implementation of the ordering system. The system includes the main functions of user registration and login, seller adding / modifying goods, buyer ordering, payment, goods distribution, buyer message, password modification and so on.

**Key Words:** SpringBoot；jsp；shiro；redis；Git；MySQL

# 1 绪论

## 引言

随着我国经济全球化的逐步深入，人们的生活水平也逐步提高，人们的消费观念于消费方式也有了明显的改变，顾客在购买喜爱的商品时，线上支付的方式也流行于我们的生活中。这种新的行为习惯也改变着餐饮行业，但是大多数餐饮店，由于模式观念转变较慢，技术支持不足或其他原因仍采用传统的消费模式，这种方式不但浪费人力物力，相比较其他已支持线上点餐的商家会流失大量客户。因此，线上点餐的模式成了当今人们在餐厅满足自己需求的趋势。而本系统就是建立这种需求之上，使用B/S模式，商家在后台直接上架商品，包括商品的价格、库存、口味、备注，商品上架后可在首页看到。顾客还能选择销量最高、最新上架的或者直接根据关键字搜索自己想要的商品，大大增加了用户体验感。

## 1.2点餐系统的开发背景

随着网络在生活中的普及，互联网已经成为人们目前最习惯使用的生活助手，人们可以足不出户的进行生活消费，甚至可以通过线上购买自己想要的商品。但是在传统的餐饮模式下，顾客只能通过服务员根据菜单将菜品展示给用户，而且流程十分繁琐，以及人力物力和门面的装修需要付出太多的心血，用户的体验度也并没有得到有效提升。不仅如此，餐饮业在日常对菜品的管理也是普遍采用手工的方式，整体技术含量和工作效率都较低，难以适应现代化经营管理的要求。线上点餐也由此成为了商家与用户迫切的需求。

## 1.3 点餐系统的开发意义

首先饮食是人们生活必不可少的一种行为，大多数用户为了提高生活水平都会到餐厅进行饮食消费。其次现代生活的快速化使得人们越来越倾向于在餐厅自由点餐，迅速浏览商家的热销菜品以及对应评价并进行线上支付。另一方面，由于计算机技术的不断发展，尤其是局域网技术的日渐成熟，为餐饮企业建立信息管理系统提供了技术上的支持；而且随着计算机硬件的成本下降，经济适用性也得到了保障。综合以上原因，开发点餐系统具有十分重要的现实意义，对餐饮业的发展有极大的推进作用。

## 1.4 网上点餐的市场趋势

互联网的普及为网上点餐提供了发展空间，同时，网上点餐服务的直观、便捷等优点是传统的点餐业务无法比拟的。数据调查显示，大城市白领更乐于选择网上点餐服务。网络点餐随着互联网的成长也在逐渐被人们所喜爱，正如几年前手机移动的短信一样，为企业带来的几百个亿的业务收入。在当今互联网世界里面，谁早一步在应用上创新，谁就掌握了未来市场的方向，因此，网上点餐更是新的方向。

## 1.5 网上点餐的市场优势

中国在线点餐服务平台有很大发展潜力，如果以2007年餐饮业零售额12352亿元的1%的在线点餐佣金计算，在线点餐服务的佣金市场就将近123亿元，再加上广告费等各类其他形式的收入，在线点餐服务平台的潜在市场将至少达到150亿元，这也使得这种商业模式受到风险投资的青睐，并得以迅速发展。

网上点餐通过文字说明、图片、动画加强对产品的宣传，大大加深餐饮业的“色型”要求。实现产品管理方便，起到立竿见影的效果，不用因为更改菜色而重新印刷。点餐途径也有了更好的改善，网上点餐成为电话点餐又一有力补充，扩大客流，从而整体提高业务量。另外如今餐店的定位更是网络化，开放化，缓解了食堂拥挤的困境，为食堂提供更为轻松的环境。

# 2 点餐系统相关的技术介绍

## 2.1 SpringBoot

说起SpringBoot框架，就要先说一下Spring。Spring是Java企业版的轻量级代替品。无需开发重量级的EnterpriseJavaBean（EJB），Spring为企业级Java开发提供了一种相对简单的方法，通过依赖注入和面向切面编程，用简单的Java对象实现了EJB的功能。Spring不仅仅局限于服务器端开发，任何Java应用都能在简单性、可测试性和松耦合性等方面从Spring中获益。而Spring Boot 是所有基于 Spring 开发的项目的起点，它对Spring进行了改善和优化，基于约定优于配置的思想，可以让开发人员不必在配置与逻辑业务之间进行思维的切换，全身心的投入到逻辑业务的代码编写中，从而大大提高了开发的效率，一定程度上缩短了项目周期。

## 2.2 JSP

JSP全称Java Server Pages，是一种动态网页开发技术。它使用JSP标签在HTML网页中插入Java代码。标签通常以<%开头以%>结束。JSP是一种Java servlet，主要用于实现Java web应用程序的用户界面部分。JSP通过网页表单获取用户输入数据、访问数据库及其他数据源，然后动态地创建网页。JSP标签有多种功能，比如访问数据库、记录用户选择信息、访问JavaBeans组件等，还可以在不同的网页中传递控制信息和共享信息。

## 2.3 Shiro

Apache Shiro是一种基于Java的安全框架，功能强大而且使用简单,它为开发人员提供一个直观而全面的认证，授权，加密及会话管理的解决方案。Shiro有3个核心组件分别为Subject，SecurityManager和Realms。其中Subject即“当前操作用户”，但是，在Shiro中，Subject这一概念并不仅仅指人，也可以是第三方进程、后台帐户（Daemon Account）或其他类似事物。它仅仅意味着“当前跟软件交互的东西”。Subject代表了当前用户的安全操作，SecurityManager则管理所有用户的安全操作。它是Shiro框架的核心，典型的Facade模式，Shiro通过SecurityManager来管理内部组件实例，并通过它来提供安全管理的各种服务。而Realm充当了Shiro与应用安全数据间的“桥梁”或者“连接器”。也就是说，当对用户执行认证（登录）和授权（访问控制）验证时，Shiro会从应用配置的Realm中查找用户及其权限信息。

## 2.4 Redis

Redis是当前比较热门的NOSQL系统之一，它是一个开源的key-value存储系统。它和Memcache类似，但很大程度补偿了Memcache的不足。Redis和Memcache一样，数据都是缓存在计算机内存中，不同的是，Memcache只能将数据缓存到内存中，无法自动定期写入硬盘，这就表示，一断电或重启，内存清空，数据丢失。所以Memcache的应用场景适用于缓存无需持久化的数据。而Redis不同的是它会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，实现数据的持久化。

## 2.5 Git

Git是目前世界上最先进的分布式版本控制系统（没有之一）。Git 是一个开源的分布式版本控制软件，可以有效、高速地处理从很小到非常大的项目版本管理。用于版本回退和多人同时开发。与常用的版本控制工具CVS，Subversion等不同，它采用了分布式版本库的方式，不必服务器端软件支持。在本系统中，把项目代码放到GitHub上，用Git做版本控制。

## 2.6 MySql

MySsql是一种[关系型数据库](http://baike.baidu.com/view/1450387.htm" \t "http://www.360doc.com/content/14/0213/16/_blank)，该数据库使用的是标准化数据库语言，它将数据保存在不同的数据表中，这样数据的关系将更加清晰，数据的查找也更加的方便，其特点为其体积小、速度快、总体拥有成本低。开发者对数据处理的效率会更加高效，大大方便我们的操作也方便于生产的实际应用。

## 2.7 Jmeter

Apache JMeter是Apache组织开发的基于Java的压力测试工具。用于对软件做压力测试，它最初被设计用于Web应用测试，但后来扩展到其他测试领域。 它可以用于测试静态和动态资源，例如静态文件、Java小服务程序、CGI 脚本、Java对象、数据库、FTP服务器等等。JMeter可以用于对服务器、网络或对象模拟巨大的负载，来自不同压力类别下测试它们的强度和分析整体性能。另外，JMeter能够对应用程序做功能/回归测试，通过创建带有断言的脚本来验证你的程序返回了你期望的结果。为了最大限度的灵活性，JMeter允许使用正则表达式创建断言。

# 3 点餐系统需求的调查以及实现

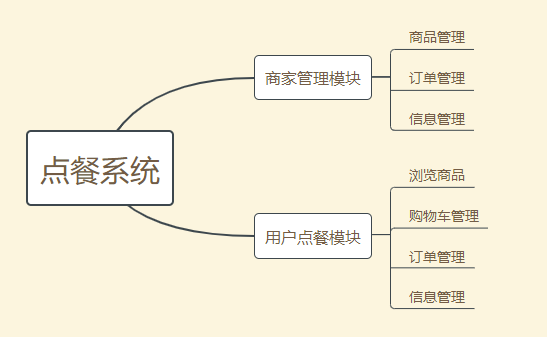
## 3.1 需求分析

在对大量线下实体店的调查后，该点餐系统中应当存在卖家和买家两种角色。点餐系统的卖家是饭店的老板，而买家一般为进入网站购买商品的顾客。对于购买商品的顾客而言，可能也只会产生浏览商品和购买商品的需求，以及管理与自己相关的用户信息的需求。对于卖家而言，则可能会产生对商品信息管理的需求、对商品分类管理的需求以及对用户信息管理的需求。

考虑到系统移植的普适性、轻量级开发和减少系统搭建难度，整体架构采用SpringBoot框架。以注解相的方式进行业务逻辑的开发，持久层使用Mybatis，可以使大部分SQL语句更加灵活，配置也相对于Hibernate比较简单轻便，对事物的控制更加灵活有效，也可有效的防止SQL注入攻击。

## 3.2功能模块设计

点餐系统的功能模块分为以下两大部分内容，具体如图1所示。

图1 扫码点餐系统总体结构图

### 3.2.1 商家管理模块

商家管理模块大致分为商品管理、订单管理和信息管理三部分。

商品管理：该模块是商家对本店铺菜品进行增加、删除、修改等功能的集合，可以设置菜品的数量，菜品上下架状态，还有菜品的口味，菜品的价格和库存，当然还可以增加菜品展示时的备注说明，同时在菜品展示时还有“已售数量”来表示该菜品的受欢迎程度。

订单管理：该模块是商家可以看到某笔订单的详情，包括菜品名称、菜品数量、用户的备注、下单时间、期望到达时间、订单地址、订单的支付方式以及订单状态。商家还可以进行订单状态的更新，及时将订单的处理通知给用户，比如已发货、已送达等物流信息。

信息管理：该模块是商家对个人信息管理的模块。可以修改门店的地址和商家的名称，商家手机号，以及商家登录该系统账号的密码，该系统对密码采用MD5加密处理，保证用户信息的安全性。

### 3.2.2 用户点餐模块

用户点餐模块大致分为浏览商品、购物车管理、订单管理和信息管理四部分。

浏览商品：该模块是用户进入该系统后的首页展示，用户可以看到展示的菜品，可以根据默认、最新、热销三种种类查看不同的菜品，同时也可以在首页按菜品的名字模糊搜索想吃的菜品。首页商品展示还显示已售数量供用户参考。

购物车管理：从首页可以直接添加商品进购物车，也可以进入商品详情页再添加购物车，也有对商品加入购物车后的一系列管理，包括下订单，再次加入购物车。

订单管理：用户的订单管理不同于商家的，商家的订单管理只能管理顾客的订单状态，而用户在下订单的时候，配送地址可以选择默认的，也可以选择新建，新建地址是在一个新的页面完成的。配送时间也有几个时间段可供选择，还可以加备注等功能。

信息管理：用户的信息管理同商家的一样。

# 4 数据库设计

## 4.1概念结构设计

好的数据库设计能够节省整个系统的开发时间，并且也可以规避上线运行之后的很多Bug，另外对系统的扩展性、延伸性、兼容性、并发性都有很大的帮助。数据库的结构是为了显示更多抽象实体和实体类之间的关系。这种关系表现在我们经常提到的E-R图的形式中。其中一对一联系（1：1）：如果对于实体集A中的每一个实体，实体集B中至多有一个（也可以没有）实体与之联系，反之亦然，则称实体集A与实体集B具有一对一联系，记为1：1。一对多联系（1：n）：如果对于实体集A中的每一个实体，实体集B中有n个实体（n>1）与之联系，反之，对于实体集B中的每一个实体，实体集A中至多只有一个实体与之联系，则称实体集A与实体集B有一对多联系，记为1：n。多对多联系（m：n）：如果对于实体集A中的每一个实体，实体集B中有n个实体（n>0）与之联系，反之，对于实体集B中的每一个实体，实体集A中也有m个实体（m>0）与之联系，则称实体集A与实体集B具有多对多联系，记为m：n。

在该系统中，主要存在八个实体，也就对应数据库八张表，分别是用户表、商品表、商品图片表、订单表、购物车表、订单详情表、菜单表、地址表。该点餐系统的E-R图如图2所示。

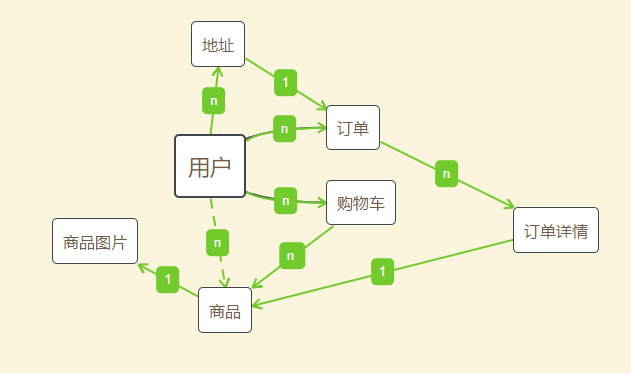


图2 点餐系统数据库E-R图

## 4.2逻辑结构设计

点餐系统使用MySql数据库进行数据表单的创建，根据分析，数据库表共有8张，由于篇幅的限制，在这里以其中五张进行说明：用户信息表（user）、用户地址表（address）、商品信息表（product）、购物车表（car）、订单表（orders），具体如下：

（1）用户信息表（user），主要存放用户注册时存储的信息，比如用户邮箱、用户密码、用户昵称、用户真实姓名、用户电话、用户角色、用户性别等。具体如表1所示。

表1 user（用户信息表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 数据名称 | 数据的类型 | 数据大小(bit) | 数据解释 |
| 1 | user\_id | bigint | 20 | 用户id |
| 2 | user\_email | varchar | 30 | 用户邮箱 |
| 3 | user\_password | varchar | 50 | 用户密码 |
| 4 | user\_nickname | varchar | 30 | 用户昵称 |
| 5 | user\_realname | varchar | 30 | 用户真实姓名 |
| 6 | user\_phone | varchar | 20 | 用户电话 |
| 7 | user\_rol | char | 1 | 角色（0顾客、1卖家） |
| 8 | user\_sex | char | 1 | 性别（0女、1男） |

（2）用户地址表（address），主要存放用户地址的相关信息，比如用户地址详情、是否是默认地址等，具体如表2所示。

表2 address（用户地址表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 数据名称 | 数据的类型 | 数据大小(bit) | 数据解释 |
| 1 | address\_id | bigint | 20 | 地址id |
| 2 | address\_buyer\_id | bigint | 20 | 对应用户id |
| 3 | address\_address | varchar | 200 | 用户地址详情 |
| 4 | address\_is\_default | char | 1 | 是否是默认地址 |

（3）商品信息表（product），主要存放商品的基本详情信息，比如商品编号、商品名字、商品价格、商品库存、商品口味、是否上架、商品描述、商品已售数量等，具体如表3所示。

表3 product（商品信息表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 数据名称 | 数据的类型 | 数据大小(bit) | 数据解释 |
| 1 | product\_id | bigint | 20 | 商品id |
| 2 | product\_no | varchar | 30 | 商品编号 |
| 3 | product\_name | varchar | 50 | 用户地址详情 |
| 4 | product\_seller\_id | bigint | 20 | 商家id |
| 5 | product\_price | double | 10 | 商品价格 |
| 6 | product\_stock | int | 255 | 商品库存 |
| 7 | product\_taste | char | 1 | 辣不辣（ 0否1是） |
| 8 | product\_is\_sell | char | 1 | 是否上架 （0否1是） |
| 9 | product\_description | varchar | 100 | 商品描述 |
| 10 | product\_picture | varchar | 50 | 商品图片位置 |
| 11 | product\_hass\_selled | int | 11 | 商品已售数量 |

（4）购物车表（car），主要存放购物车的详情信息，比如购物车对应的用户、购物车内的商品、商品数量、商品价格等，具体如表4所示。

表4 car（购物车表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 数据名称 | 数据的类型 | 数据大小(bit) | 数据解释 |
| 1 | car\_id | bigint | 20 | 购物车id |
| 2 | car\_buyer\_id | bigint | 20 | 对应用户id |
| 3 | car\_product\_id | bigint | 20 | 对应商品id |
| 4 | car\_product\_num | int | 11 | 对应商品数量 |
| 5 | car\_product\_price | double | 10 | 对应商品价格 |

（5）订单表（orders），主要存放订单的基本详情信息，比如订单编号、配送地址、订单开始时间、订单结束时间、要求送达时间、订单总价格、订单状态、订单备注等，具体如表5所示。

表5 orders（订单表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 数据名称 | 数据的类型 | 数据大小(bit) | 数据解释 |
| 1 | order\_id | bigint | 20 | 订单id |
| 2 | order\_buyer\_id | bigint | 20 | 用户id |
| 3 | order\_seller\_id | bigint | 20 | 商家id |
| 4 | order\_no | varchar | 30 | 订单编号 |
| 5 | order\_address | varchar | 255 | 配送地址 |
| 6 | order\_time | timestamp | 0 | 订单开始时间 |
| 7 | order\_delieve\_time | timestamp | 0 | 要求送达时间 |
| 8 | order\_finish\_time | timestamp | 0 | 订单结束时间 |
| 9 | order\_total\_price | double | 10 | 订单总价格 |
| 10 | order\_payment | varchar | 20 | 订单支付状态 |
| 11 | order\_status | char | 2 | 订单状态 |
| 12 | order\_comments | varchar | 1000 | 订单备注 |

# 5 功能模块的实现

## 5.1登录模块的实现

网站的登录模块一直都是系统中最关键的部分，用户只有通过验证账号密码才能够进入系统网站。用户在登录页面输入自己的信息，随后点击提交button进行登录操作。在login.jsp的登录页面，共调用了四种标签，使用div标签输入并展示用户信息输入框，第二种使用p标签并加粗加红为用户提示错误信息。第三种使用input标签，在form表单里面可以使用户输入其信息，并通过第四种button按钮实现点击登录事件。

在登录功能的实现原理上，当用户点击登录的时候，系统会将用户输入或选择的内容异步(ajax)传递给后台控制器LoginController。Login使用Shiro进行登录权限的控制，主要是使用Subject中的login方法对要登录的用户名密码进行验证，主要代码如下：

Subject subject = SecurityUtils.getSubject();

UsernamePasswordToken token = new UsernamePasswordToken (user.getUserEmail(), MD5Util.MD5(user.getUserPassword()));

try {

subject.login(token);

user.setUserPassword(MD5Util.MD5(user.getUserPassword()));

List<User> userList = loginService.getUser(user);

if (userList.size() <= 0) {

model.addAttribute("message", "用户名或密码不正确！");

return "login";

} else {

request.getSession().setAttribute("user", userList.get(0));

return "redirect:index";

}

 如代码所示，直接将input(text)中的value属性的值传递给后台Model对象。之后通过UsernamePasswordToken对象将用户名密码转换成token，再通过subject的login方法进行登录认证，认证成功后根据用户名查询得到用户对象的信息并存入session中以便页面显示使用，用户名密码在MD5加密后解析更具有安全性。如果身份验证成功跳转首页，如若校验失败则会提示“用户名密码不正确”。另外还可以在登录页面注册新用户和修改密码操作，如下图3所示：



图3 点餐系统登录页面

## 5.2首页模块的实现

登录后进入首页，查询商品表的全部信息，为了提高并发性，选择把商品分页显示在页面上，分页用的是pagehelper，它能简单高效的对查询数据库的结果进行封装。首页有最新、默认、销量最高三种商品分类，对应的是持久层框架mybatis的条件搜索，再加上在首页还能根据菜品的关键字进行模糊查询与价格的间距搜索，更加体现了持久层框架mybatis的灵活性。对应的代码如下：

//根据输入条件，进行组合查询

ProductExample example1 = new ProductExample();

ProductExample.Criteria criteria1 = example1.createCriteria();

//确定最高价和最低价

criteria1.andProductPriceBetween(min, max);

//上架的才显示 不上架的不显示

criteria1.andProductIsSellEqualTo("1");

//判断商品的状态

if (status != 0) {

if (status == 1) {

//按照最新查询

example1.setOrderByClause("product\_id DESC");

} else {

//按照卖出最多查询

example1.setOrderByClause("product\_hass\_selled DESC");

}

}

//根据搜索的关键词在进行查询

if (word != null) {

criteria1.andProductNameLike("%" + word + "%");

}

在跳转首页的时候，就会把分页、菜品分类状态、价格最小和最大、菜品关键字通过from表单或者url参数传到后端的PageController控制器，再根据mybatis分页查询后紧接着跳转回首页显示。另外在首页也可以直接查看菜品详情和加入购物车；用户信息入口，退出登录、购物车、订单列表都设计在在右上角，方便观察。点餐系统的首页如下图4所示。



图4 点餐系统首页页面

## 5.3商品管理的实现

商品管理功能是在商家角色下进入首页才有的点击入口，是商家对本店菜品的添加、修改、删除等一系列操作。在一般场景中，商品列表和商品详情页，应该是点击很频繁的页面，对于这些页面中的记录，一般是需要频繁查询数据库的，这样对数据库会产生很大的压力，有可能导致查询很慢，这对用户的交互感非常差。因此本系统对于一些需要经常查询数据库的记录，会把这些记录放进Redis缓存中，这样一来就会减少数据库的压力，而且访问Redis也是非常的快速。商品列表页面在前端代码层面大多数都是一些点击事件和点击提交函数，对应的是JavaScript中的一些click函数，还有ajax局部刷新请求数据。在商品详情获取的后端处理器中，首先通过session获取会话中的用户的信息，如果用户对象为空则获取要查看的商品id，根据商品id查询商品的各种信息，包括图片、价格、库存、口味、和备注等信息。点餐系统管理商品列表页面如下图5所示。



图5 点餐系统管理商品列表页面

在添加商品页面，其中上传图片部分，后端用的是MultipartFile，用它做上传图片显得非常简单，其中file.transferTo(imgFile)代码则是重点，把图片转成io流进行解析。本系统把上传后的图片保存在本地系统中，之后再根据图片的路径在前端页面显示图片。而对于上面说的Redis缓存，还有一个小问题，就是当数据库中数据更新的时候，Redis中相应的缓存就不能用了，会造成数据不一致的问题。假如一条记录在数据库更改后，再做查询就该是会走数据库，解决办法就是先删除Redis中的数据，在更新数据后插入Redis缓存中。点餐系统修改商品页面如下图6所示。



图6 点餐系统修改商品页面

## 5.4订单管理功能的实现

订单管理主要是对订单状态的改变。当点击时，首先根据该商品的id查询出该商品的相关信息，同时加载当前用户的收货地址信息。汇总了该商品和用户信息之后提交订单。这时会在redis中生成一个13位的订单id，同时将该订单id和商品信息存入订单表中。订单id为了保证不重复，该系统的做法是时间戳加5位随机数，保证订单id的唯一性。对于订单的状态总共有12种，包括0：已确定-未付款，1：已确定-已付款-未发货，2：已确定-已付款-已发货-待收货，3：已确定-已付款-已发货-已收货，4：已确定-货到付款-未发货，5：已确定-货到付款-已发货-未收货，6：已确定-货到付款-已发货-已收货，7：已经评价，8：取消订单，9：退款申请，10：已退款，11：用户不收货，12：超时10分钟未付款，订单取消。而其中的“12”状态超时10分钟未付款，订单取消，在该系统中是通过定时任务中@Scheduled(fixedRate = 600000)注解完成的，时间是10分钟。代码如下：

@Scheduled(fixedRate = 600000)

public void checkOrder() {

log.info("检测未支付订单是否超时...");

OrdersExample ordersExample = new OrdersExample();

OrdersExample.Criteria criteria = ordersExample.createCriteria();

criteria.andOrderStatusEqualTo("0");

List<Orders> orderList = ordersMapper.selectByExample(ordersExample);

orderList.forEach(order->{

Date finishTime = order.getOrderFinishTime();

Date date = new Date();

if (date.getTime() - finishTime.getTime() > 600000) {

order.setOrderStatus("12");

ordersMapper.updateByPrimaryKeySelective(order);

log.info("订单已被取消：" + order.getOrderNo());

}

});

}

另外在订单付完款后，用户还能对某个订单进行评价，而且商家也能看到顾客的评价跟反馈。完成支付之后需要在后台对订单进行审核，把状态改为已发货，返回前台页面可以看到订单的信息已经发生了更新。本系统没有模拟物流系统所以当我们点击确认收货，后台处理程序则更改本次订单的收货状态字段，并将评价字段置为待评价，并且付款方式也有显示支付和货到付款两种方式。

当收货之后，商品进入了待评价阶段，点击评价进入评价界面，点击发表评论按钮异步提交评论信息到后台controller控制器进行数据的插入，如果出现错误则根据返回的错误信息进行提示，如果执行成功则跳回到订单页，显示此订单的完整信息，包括订单编号、交易时间、名称、口味、数量等信息。点餐系统订单列表页面如下图7所示。



图7 点餐系统订单列表页面

## 5.5修改密码功能的实现

修改密码功能用到了qq的邮箱系统，在qq邮箱中设置完相应的发送邮件功能的配置后，还需要在代码配置文件中配置如下图8信息：

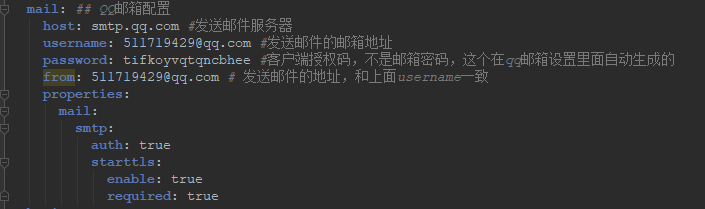


图8 点餐系统邮箱配置页面

当修改密码时发邮件给指定的账号，这一步是异步操作的，就是系统中修改密码页面会立即返回结果并跳转到消息成功页面，而发送邮箱跟返回结果不会有直接的关系，大概20秒后邮箱会收到邮件。这部分代码用到了Java8中的新特性Lambda表达式，stream流在书写上和观赏性上比较着Java8之前有很大提高。

//异步

CompletableFuture.runAsync(()->{

iMailService.sendHtmlMail(email, "主题：修改密码",

"<a href='http://localhost:82/editPassword?userId=" + user.getUserId() + "'>跳转重置密码页面</a>");

);

在改密码的时候需要发邮件给原账号，主要流程是：在页面填入需要修改的账号，点击“发送邮件”，用户会在邮箱收到邮件，在邮件中点击链接时会直接跳转修改密码页面，当用户修改完密码后等5秒跳转回登录页面进行重新登录。当初在选择邮箱系统的时候，有很多选择，但是根据大数据分析到当代大学生用qq的比较多，所以用户注册的时候可能以qq账号为本系统的账号进行注册，因此选择了调用qq的邮箱系统接口。点餐系统收到修改密码邮件页面如下图10所示。

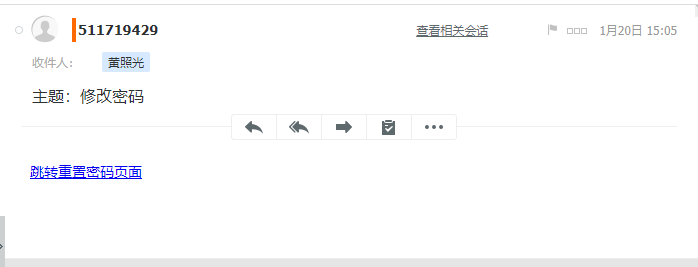


图9 点餐系统收到修改密码邮件页面

## 5.6线上付款功能的实现

该系统为了模拟付款，调用了支付宝的沙箱接口来进行线上付款，沙箱环境非常接近真实的付款环境，这也是该系统选择支付宝沙箱的重要原因之一，而且操作也很简单。付款方式也采用当下最流行的扫码付款，既便捷又安全。

在Java代码中，通过new DefaultAlipayClient创建沙箱环境对象，在传的七个参数中第一个参数是沙箱环境的回调网址，第二个参数是自己在沙箱管理平台申请的唯一的识别id，第三个参数也是在沙箱管理平台申请的私钥，第四个和第五个参数是传入的格式，第六个参数是支付公共键，第七个参数的固定参数代表加密方式。主要代码如下：

//获得初始化的AlipayClient

AlipayClient alipayClient = new DefaultAlipayClient ("https://openapi.alipaydev. com/gateway.do", APP\_ID, PRIVATE\_KEY, "json", "utf-8", ALIPAY\_PUBLIC\_KEY, "RSA2");

//创建API对应的request

AlipayTradePagePayRequest alipayRequest = new AlipayTradePagePayRequest();

if (all == 1) {

//成功后回调

alipayRequest.setReturnUrl(ww + "/payok?orderSn=" + ordersn\_string + "&all=1"); //在公共参数中设置回跳和通知地址

alipayRequest.setNotifyUrl(ww + "/payok?orderSn=" + ordersn\_string + "&all=1");

}

alipayRequest.setBizContent("{\"out\_trade\_no\":\"" + ordersn[0] + "\","

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

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

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

+ "\"product\_code\":\"FAST\_INSTANT\_TRADE\_PAY\"}");//填充业务参数

String form = "";

try {

//调用SDK生成表单

form = alipayClient.pageExecute(alipayRequest).getBody();

} catch (AlipayApiException e) {

e.printStackTrace();

}

上面为调用支付宝的同步调用接口，但是官方建议应在异步调用方法中处理付款成功后的操作，但因异步调用的路径必须为公网地址，支付宝才可以发送请求给我们，故这里不写异步调用的方法了，需要注意的是异步调用为post请求，且传递来的参数会多一些，但基本与同步调用的操作一致。系统调用支付宝沙箱环境进行付款购买商品更加具有真实性。点餐系统付款页面如下图10所示。

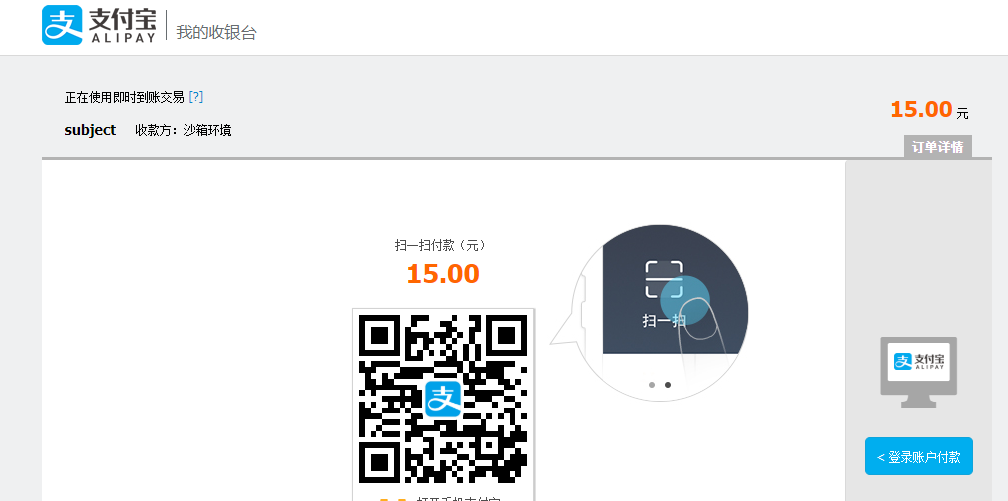


图10 点餐系统付款页面

# 6 系统测试

系统测试就是发现系统中存在的缺陷并将缺陷改正的过程，也是项目开发生命周期的最后一个步骤，是不能省略的一个重要步骤。系统测试主要分为功能测试和性能测试两大部分。

## 6.1 功能测试

在功能测试中，申请了多个商家和顾客账号用来测试每一个接口，主要是对登录、、注册、商品详情、订单、购物车、付款、修改密码发邮件等接口做重点测试，测试结果如下表6所示，跟预期的结果大致相同。

表6 测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 测试模块 | 测试功能 | 测试详情 | 测试结果 |
| 公共模块 | 登录 | 账号：admin@qq.com  密码：admin | 登陆成功，跳转首页 |
| 账号：511719429@qq.com  密码：123456 | 登陆成功，跳转首页 |
| 注册 | 账号：511719429@qq.com  密码：123456 | 注册成功，跳转登录页面 |
| 修改密码 | 账号：511719429@qq.com  原密码：123456  新密码：dian123!@# | 显示修改成功后，跳转登录页面 |
| 修改个人信息 | 修改部分信息 | 修改成功后跳转首页 |
| 退出登录 | 点击退出按钮 | 退出成功后跳转首页 |
| 搜索商品 | 在首页按分类搜索和关键字模糊搜索 | 搜索返回结果正确 |
| 订单管理 | 更改订单的状态（包括商家更改和用户更改） | 更新成功 |
| 商家模块 | 添加商品 | 添加新的商品（包括上传图片） | 添加成功 |
| 修改商品 | 对商品信息进行修改 | 修改成功 |
| 删除商品 | 选择商品并点击删除 | 删除成功 |
| 用户模块 | 购物车 | 添加商品到购物车 | 添加成功 |
| 删除已选择的商品 | 删除成功 |
| 下订单 | 从购物车结算 | 下单成功 |
| 支付 | 扫码支付任意金额 | 支付成功，跳转订单列表页面 |

## 6.2 性能测试

在性能测试中，该系统采用jmeter对一些重要的接口进行压力测试，下面着重介绍一下登录接口的压力测试。

对登录进行压力测试时，只需要设置一个简单的http登录，它会自动获取cookie信息，然后把post请求改为跟随重定向，添加线程组，添加循环控制器，接着填写http请求头和账号密码等信息，另外还需要在项目中修改配置文件jmeter.properties，设置CookieManager.save.cookies=true，然后点击运行就可以查看结果树了。在本次结果中，如下图11所示，可以看到error失败率为0%，在对登录接口代码做了一些调整好，平均响应时间降低了，并且吞吐量也有提高。为了做并发测试，添加了很多个线程数，并设置结果树、图形结果、聚合报告，把间隔时间设置为0，可以看做同时访问，也就是并发请求。

Jmeter还可以通过设置断言来验证结果，用来做回归测试很方便，本次系统就不做延伸了。

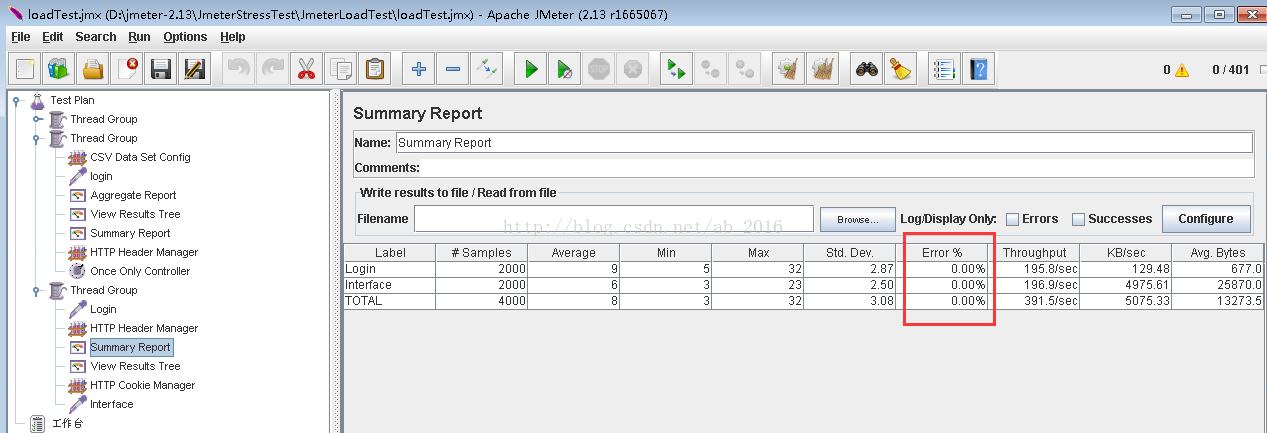


图11 点餐系统登录压测页面

# 总结

经过一段时间不懈努力和深入专研，终于将点餐系统设计并开发完成。该系统面向了广大有意愿将互联网与店铺营销模式进行结合的商家。该系统可以将商家的菜品以互联网页面的形式排列展现给顾客，告别传统的菜单，使本店特色菜品、最新菜品和热销菜品以醒目的方式推荐给顾客。顾客还可以通过进入系统迅速定位到商家推荐菜品，并选择自己喜爱的菜品进行下单，在支付的时候不只是柜台支付现金，也可以使用线上支付的方式进行消费，在顾客本次消费后可以对商家的菜品、服务进行评价与打分，使顾客可以更好的与商家进行交互，便于商家根据用户的意见进行菜品与管理的调整。而且该系统的功能经过测试证明已经可以满足大部分想改变传统销售模式的商家的需求，具有一定的实际意义。系统也存在着少量的缺陷，如点餐环节、线上支付、配送状态还需要再做完善，考虑在下一步中改进这些功能，会根据实际功能来优化代码以达到预期的效果。

# **参考文献**

1. 周兴宇. 微信自助点餐系统[D].大连海事大学,2016.
2. 黄鲤薇. 基于移动Web的点餐系统设计与实现[D].吉林大学,2016.
3. 黄文. 基于微信的叫号点餐系统的设计与实现[D].华中科技大学,2014.
4. 温立辉.Spring框架在模型层的应用及原理[J].福建电脑，2017-05-25：165-166.
5. 王旭.软件开发中JAVA编程的应用[J].中国管理信息化，2018-02-01：138-139.
6. 杨振华，杨明红.数据库编程中Java设计模式的应用研究[J].电脑迷，2017-06-15：245-246.
7. 刘正，赵鹤鸣.MVC模式下多层分布式软件系统架构设计[J].现代电子技术，2018：133-134.
8. 张桂珠，刘丽，陈爱国．Java面向对象程序设计（第2版）[M]北京邮电大学出版社，2013.
9. 张光平.Java多线程的分析[J].科技风，2018-01-10：188.
10. 王咏佳.校园订餐系统的设计与实现[D].江西财经大学,2017.
11. 张磊.网上订餐系统的设计与实现[D].吉林大学,2016.
12. 王东亮.网上点餐系统中购物车的设计与实现[J].电子技术与软件工程,2015(10):228.
13. 林子雨,邹权,赖永炫.关系数据库中的关键词查询结果动态优化[J].软件学报,2014.
14. 洪贵华.MVC设计模式在JSP开发Web应用程序中的应用探究[J].信息与电脑(理论版). 2017-12-23
15. Ramakrishnan Jagdish．A dynamic programming approach to adaptive fractionation[J]. Physics in Medicine and Biology [J]．Physics in Medicine and Biology，2012，57(5)：29-33
16. 冯颖凌, 廖子豪, 严豪. 基于Java的网上商城开发[J]. 电脑知识与技术, 2016, 12(24)：123-124
17. 刘睿文, 吴帝李. Java的B2C电子商城网站系统设计[J].电脑编程技巧与维护, 2010(22)：90-92.
18. 沈黄河. 电子商务平台网上商城系统的设计与实现[J]. 电脑与电信,2013(22):40-52.

**致谢**

该论文是在指导老师吴冉的悉心指导和帮助下完成的。她教学态度严谨，对我在论文的写作中提供了很大的帮助，严格要求我的论文质量，力争让我做到最好。在我有疑问的时候她都会第一时间给我解答，并给我正确的指引。在论文完成的过程中，老师给了我很多指导意见，细心地在论文中写了大量的批注告诉我应该怎么改，在我遇到瓶颈不知该怎么写的时候，老师的提点总会让我茅塞顿开。还要谢谢我的学长，论文中出现的一些技术问题在询问学长后也得以解决。现在毕业论文已经接近尾声，在此我向吴冉老师表示深深的感激！

在论文即将完成之际，心中不禁感慨万千。四年的大学生涯仿佛就在昨日，半年的实习经验让我学到了很多东西，也认识到在学校期间老师给我们的指导是多么的重要，让我们出校的瞬间就站在了比同龄人更高的起点上。我要感谢在这四年中给予我极大支持和帮助的老师和同学们以及父母。因为有了父母的支持，我才能有四年的时间来安心的学习，因为有老师和同学的帮助我才得以能在四年里学到更多的知识和技能，谢谢你们在生活和学习中对我的帮助和包容。是你们让我树立信心，技术上攻克一个又一个的难题，希望你们在以后的日子里生活和和美美。最后，请让我再次表达对你们的感谢，谢谢你们的帮助！