|  |
| --- |
| **本科毕业论文**  基于WEB的网上订餐系统  学生姓名：陈逸  指导教师：王文永  所在学院：信息与软件工程学院  所学专业：软件工程  中国·长春  2017 年 5 月 |

# 摘要

订餐网站在当下年轻人中备受推崇，其中的根本原因就是互联网的普及和科学技术的不断创新。科学技术是第一生产力，订餐网站给人们的生活带来的不仅仅是方便便捷，让人们足不出户就能吃到想吃的饭菜，更是创造了新兴的外卖骑手职业，解决了很多人的就业民生问题。商家可以通过网上订餐系统扩大自己的销售渠道，减少人工开销，为顾客提供更高质量的服务。顾客也可以通过网上订餐系统“货比三家”，省时省力地购买到自己想要的服务。在网上订餐系统这一环境中，进行准确、高效率的信息传递，给不论何时何地有就餐需求的顾客，提供一份便捷的服务，这与传统的餐厅营业模式相比，无疑是能够更进一步地迎合顾客的需求的。

本文详细地介绍了“基于WEB的网上订餐系统”的设计方案、各模块的功能分配，并分析了相关技术。本文将针对系统中各个功能模块进行详细的论述，包括顾客所能浏览的网站的首页、餐品分类页、餐品筛选页、顾客评价以及商家的餐品发布页面等。本文所介绍的“网上订餐系统”的前端页面应用了HTML语言、JSP开发语言；后台的设计应用了MySQL数据库的服务平台，利用Spring MVC框架，通过实现Model-View-Controller模式来分离数据、业务与展现，逐个完成各层次模块的开发，实现模块的功能。

**关键字：**网上订餐系统；HTML语言；JSP开发语言；MySQL数据库；Spring MVC框架；

# Abstract

The web site is highly prized among young people, the root cause of which is the popularity of the internet and the innovation of science and technology. Science and technology is the first productivity, ordering websites to bring people's life not only is convenient and convenient, people can choose the food they want to eat, but also create a new delivery career, and solve many people's employment and people's livelihood problems. Business can through the online ordering system to expand their sales channels, reduce labor costs, to provide customers with higher quality of service. Customers can also through the online ordering system " shop around ", save time and effort to buy their desired services. In the environment of online ordering system, accurate and efficient information transmission, to provide a convenient service, no matter when and where the customer has the demand, this is more to meet the customer demand than the traditional restaurant business model.

This paper introduces in detail the design scheme of " web-based online reservation system ", the functional allocation of each module, and analyzes the related technology. This paper will discuss the function modules of the system in detail, including the front page of the website, the classification page of the meal, the screening page of the meal, the customer evaluation and the merchant's table, etc. The front page of " online ordering system" introduced in this paper applies html language and JSP development language The background design and application of the MySQL database service platform, using the spring MVC framework, through the implementation of the model-view-controller pattern to separate the data, business and display, individual completes each level module development, realization module function.

**Keyword:** Online ordering system; Html language; Jsp development language; Mysql database; Spring MVC framework;

# **目录**

[摘要 2](#_Toc23848)

[Abstract 3](#_Toc22799)

**[目录](#_Toc9681)** [I](#_Toc9681)

[第一章 绪论 1](#_Toc6681)

[1.1问题提出 1](#_Toc6601)

[1.2 国内外研究现状 1](#_Toc144)

[1.3目标与意义 2](#_Toc26725)

[1.4研究的内容和方法 3](#_Toc13147)

[1.4.1研究内容 3](#_Toc1611)

[1.4.2研究方法 3](#_Toc120)

[第二章 基于WEB的网上订餐系统的理论基础 4](#_Toc19879)

[2.1 前端开发工具及框架 4](#_Toc13540)

[2.1.1 Dreamweaver CC 2017 4](#_Toc19715)

[2.1.2 Angular框架 5](#_Toc11413)

[2.2后台开发工具及框架 5](#_Toc8641)

[2.2.1 IDEA 5](#_Toc9094)

[2.2.2 Spring MVC框架 5](#_Toc25485)

[2.3 MySQL数据库 6](#_Toc22502)

[2.4 本章小结 6](#_Toc4163)

[第三章 基于WEB的网上订餐系统的需求分析和设计 6](#_Toc20192)

[3.1系统分析 6](#_Toc9243)

[3.1.1“顾客”模块需求分析 6](#_Toc22690)

[3.1.2“商家”模块需求分析 8](#_Toc10264)

[3.1.3“配送员”模块需求分析 10](#_Toc5507)

[3.1.4“系统管理员”模块需求分析 12](#_Toc4365)

[3.1.5“商家后台”模块需求分析 14](#_Toc3452)

[3.2系统设计 16](#_Toc23481)

[3.2.1架构设计 16](#_Toc13396)

[3.2.2概要设计 17](#_Toc30402)

[3.2.3数据库设计 25](#_Toc18470)

[第四章 系统功能的实现 29](#_Toc20896)

[4.1系统开发环境配置 29](#_Toc9522)

[4.2网上订餐系统的界面设计 30](#_Toc24494)

[4.2系统功能实现 34](#_Toc16660)

[4.2.1“顾客”模块实现 34](#_Toc6400)

[4.2.2“商家”模块实现 38](#_Toc20696)

[4.2.3“商家后台”模块实现 43](#_Toc3757)

[第五章 系统测试与维护 47](#_Toc30850)

[5.1功能测试 47](#_Toc5811)

[5.2 界面测试 47](#_Toc6148)

[5.3性能测试 47](#_Toc29591)

[第六章 总结 48](#_Toc10424)

[参考文献 49](#_Toc15987)

[致谢 50](#_Toc30681)

# 第一章 绪论

## 1.1问题提出

网上订餐系统，顾名思义，就是让人们能够在网上订餐的一个渠道。人们通过互联网平台，选择自己心仪的餐品，使用网银支付下单后，平台另一端的餐馆就开始着手准备好这些餐品，接着安排配送员，将外卖送达顾客所指定的地点。

网上订餐系统的存在给人们的生活带来了极大的便捷，最显而易见的就是，从前人们想要吃到某家店的食物，必须到固定的店面去吃，而现在足不出户就可以吃到想吃的饭菜。从另一方面来讲，传统的商家可能因为店面无法容纳过多的食客，而不得不限制每天的供应量，有了订餐系统这一平台后，可以使得更多的食客吃到想吃的饭菜，也可以使得商家接受更多的订单。对于商家而言，网上订餐节约了扩大店铺的成本，可以以低廉的价格为顾客提供高品质的餐品；对于顾客而言，他们可以方便地浏览餐品信息，自由地选择能够满足自己口味的餐品。无论从餐馆经营者的角度，还是从顾客的角度来看，网上订餐系统的出现都是给传统餐饮业注入生机的一次双赢的机会。

网上订餐系统的发展势如破竹，深受广大学生党、上班族的追捧。而广大学生党和上班族点外卖时，往往正是用餐高峰期，餐厅实体店内正在用餐的客户人数也达到饱和状态，在这一情况下，商家并不能及时将餐品送达顾客的手中。人们对于等待送餐的心理承受时间大约是一小时，超过这个时间段的商家极有可能会流失大量潜在顾客。另外我们知道，在网上订餐平台发展之初，为了吸引商家，抢占市场份额，网上订餐平台允许商家“无门槛”入驻。由于缺少第三方平台的监管，网上订餐的食品安全现状令人堪忧。

为了解决这些问题，本文介绍的“网上订餐系统”将一方面为保证送餐效率，首先要求商家给出预计送达时间，实行超时赔付机制；其次提供一键生成昨天菜单的功能；再将配送员与餐厅分离，一个配送员对应一片区域的商家的餐品的配送，配送员的工资由区域内商家共同支付。另一方面为保障食品安全，“网上订餐系统”将核查商家的营业执照，保证给顾客提供一个较高的用餐水准。

## 1.2 国内外研究现状

餐饮业历来被称为“百业之王”。因为宴席存在的必要性，外卖订餐永远不可能取代传统的餐馆经营模式，只能起到补充的作用。明确外卖市场的定位后，我们可以确定了网络订餐系统的目标是补充优化餐馆的传统经营模式，是一种新的盈利模式。划分餐饮市场的标准有很多，按照就餐人群、菜品档次、就餐时间、菜品风味来看，外卖市场无疑能够适宜最广的就餐人群，能够提供最平民的价格，并能够灵活适应各类就餐人群的就餐时间、用餐口味。

从近几年网上订餐系统发展可见网络订餐的兴起是大势所趋，从美团到饿了么，从大众点评到百度糯米，现在网络订餐的平台越来越贴近大众的生活习惯，渐渐融入到人们的日常生活中。不仅仅是在中国，目前网上订餐平台在国外的发展也势如破竹：英国著名的外卖创业公司Deliveroo、美国最大的在线外卖平台Grubhub、德国的外卖点餐平台Delivery Her等等，这些网上订餐平台的发展状态已经趋于成熟，其中不乏有些平台，例如英国的Deliveroo，已经拥有自己专业的外卖配送团队。

据了解，目前国内外订餐网站的有着一个共同的盈利模式，他们给顾客提供操作简单、界面吸引人的互联网点餐平台，并给予顾客一定的折扣以吸引顾客。这其中，这些网络订餐平台所扮演的就是导购商的角色，导购商与餐厅经营者达成接近成本价的价格协议，在订餐系统与顾客的交易中将自己获得的利润提成的一部分返还给顾客，自己获得中间的差价。在这类交易中，餐饮店获得了专业并且效果很好的宣传，顾客可以以几乎最低利润化的价格得到餐品。

虽然目前网上订餐平台数目众多，餐饮业的商家们也纷纷展开网上订餐项目，但是整个体系却缺乏相关标准和规定，例如饭菜分量的标准、支付方式的标准、送达时间的规定、饭菜卫生情况的检测标准等等，这一系列规则的缺乏常常会引发出商家和顾客之间的矛盾。

## 1.3目标与意义

本文所讨论的基于WEB的网上订餐系统，弥补了传统电话订餐模式的缺陷，在实现网上订餐模式的基础上，给商家和顾客之间建立了一个稳定安全且可靠的数据库环境；并在参考各大网上订餐系统后，建立出一个全面综合的网上订餐系统，将顾客、配送员以及商家的管理科学化、程序化、规范化。本文所介绍的网上订餐网站的体系架构的设计，充分考虑到顾客的个性化的需求、多并发的业务性以及扩容性，提供了多重接口，也为今后的维护预留了足够的空间。

具体来说，网上订餐系统将会解决当下餐厅没有统一的标准规定的问题，将商家的营业情况、餐品分量状况、餐具卫生情况等纳入系统管理的范围中。并且，网上订餐系统将把餐厅的商家和配送员实行分开的管理，将配送员从传统的送外卖模式中分离开来，使得外卖的送达时间能够控制在预计的范围中，以此来缩短顾客的等待时间。与此同时，网上订餐系统还会提供一键复制上个订单的功能，并且将购物过的店铺放在搜索后的首位，让顾客能够减少思考如何选择外卖餐品的时间，以此提供更多的贴心服务。

应用科学技术来提升人们的生活水平是一个必然的趋势，民以食为天，网上订餐系统的实现给人们带来的是生活中的极大的便利。在很多大城市中，快节奏的生活方式让很多人顾不上解决自己的吃饭问题，人们忙碌于自己的工作，鲜少有空闲自己做顿饭吃，因此随着网络订餐的普及，及时送到的外卖能够让忙于工作的人们吃上热腾腾的饭菜，避免了更多的人因为来不及吃饭而患上胃病的问题。

然而，美中不足的是，“无门槛”的网上订餐平台的准入机制，让一些不法商家有机可乘，一些不卫生的小作坊混迹在网上订餐平台中，鱼目混珠，以次充好，给一些顾客的身体健康状况造成了隐患，因此，当务之急是解决网上订餐平台里，商家的食品安全卫生问题，不能因小失大，让不卫生的食物影响了顾客的身体健康。在本文所介绍的基于WEB的网上订餐系统中，商家必须持有营业执照，餐品必须符合食品卫生标准，包装和餐具必须达到食品级标准。入驻网上订餐系统的外卖商家，在注册成为网上订餐平台里的商家时，必须附带自身的营业执照以及商家的身份信息，以此来保障追究责任到个人实行的可能性。

## 1.4研究的内容和方法

### 1.4.1研究内容

本文的主要内容是对基于WEB的网上订餐系统的设计与实现，网上订餐系统的设计与实现主要需要完成以下几个工作：

1. 调查并研究网上订餐系统的发展趋势，确定网上订餐系统的目标人群，确定网站的设计框架设计、开发平台以及开发工具；
2. 学习研究网上订餐系统相关的理论技术，运用软件工程的思想来开发网上订餐系统，对网上订餐系统进行详细的需求分析和概要设计，对用户需求进行细化和逐步求精，明确各模块的功能，以及各模块功能之间的动态调用关系，构建总体的框架；
3. 在Dreamweaver上设计网页的前端，包括首页、外卖分类页、筛选页、外卖详情页、订单页、登录页、注册页等等。用Java语言开发该网上订餐系统的后台，根据需求分析和概要设计，解决开发中的各种问题；
4. 测试该网上订餐系统，并且根据测试的结果完善和维护这个系统。

### 1.4.2研究方法

1. 观察法

以设计一个主要面向广大大学生、上班一族的基于WEB的网上订餐系统为目的，通过自己的感性认识，分析经常点外卖的人群的需求，体验现有的网上订餐平台的流程，征求外卖商家的建议，仔细观察了顾客、配送员、外卖商家之间的运作模式和网站开发的基本方法。通过启发自己的思维，在现有的网上订餐系统的基础上，增加了创新的内容。

1. 文献研究法

通过查找相关文献获得所需要的相关资料，了解网上订餐系统在国内外的研究现状，全面、客观地掌握网上订餐系统所需要掌握的技术等研究时遇到的问题。文献研究法能够在设计之初，给研究者提供研究对象的历史发展以及现状，帮助建立对研究对象的直观的、全局的印象，并且为研究者后续的创新奠定了坚实的基础。

1. 统计分析研究法

通过统计身边大学生对外卖市场的需求和提出的建议，以及调查所知的上班族点外卖的次数占的比重，整理分析后，可以得出当下社会人们对于网上点餐系统的功能需求。经过统计分析比较，当下社会，人们对外卖的追捧导致了外卖市场的火热，能够让人们点外卖的网上订餐平台更是大家所需要的。而且，比之曾经的传统电话订餐方式，网上订餐平台可以让人们看到所点餐品的实物图，给人们带来了更广泛的选择。

1. 经验总结法

网上订餐系统从出现发展到现在，已经逐步具备了完善的开发体系，因此总结前人的经验显得尤为重要。对比其他网上订餐系统的优劣，批判地看待这些网上订餐系统，设计本订餐网站的各个功能模块。在实际的开发过程中，将遇到的问题进行分析、归纳、总结，将总结所得出的结论理论化和系统化，内化为必不可缺的经验，以此用于更好地开发出一个基于WEB的网上订餐系统。

1. 实践研究法

在整理了用户的需求之后，设计网上订餐系统的基本框架，从总体思想、结构模型具体到各个模块的详细功能，采用了Spring MVC框架，运用MySQL数据库，开发网上订餐系统。

# 基于WEB的网上订餐系统的理论基础

## 2.1 前端开发工具及框架

### 2.1.1 Dreamweaver CC 2017

随着互联网的普及，[HTML](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=340826" \t "http://baike.sogou.com/_blank)技术也随之不断地发展完善。同时，这个互联网高度发达的时代也催生了众多的网页编辑器。人们将网页编辑器从性质上分为了所见即所得的网页编辑器和原始代码编辑器，其中，原始代码编辑器也被称为非所见即所得编辑器。这两种网页编辑器各有所长，而所见即所得网页编辑器的优点就是容易上手，操作简单便捷，开发时能够直观地看到自己所写的网页。Dreamweaver CC 2017则是所见即所得网页编辑器的代表。

Adobe Dreamweaver的简称是“DW”，它的中文名是"梦想编织者"，开发商是美国MACROMEDIA公司。Dreamweaver是同时具备网站管理和制作网页功能的网页编辑器。DW是一款视觉化网页开发工具，是一款为专业网页设计师特别设计的开发工具，我们可以利用它制作出不受平台和浏览器的限制的动态网页。它的优势在于支持网站开发者使用多种编程语言进行开发，拥有最佳的工作效率，网站开发者可以将Fireworks、[Photoshop](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=66049" \t "http://baike.sogou.com/_blank)等档案快速移至网页上，此外，网站开发者还可以将DW与很多设计工具，比如：[Play](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=64514516" \t "http://baike.sogou.com/_blank)back [Flash](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=55941" \t "http://baike.sogou.com/_blank)、[Shockwave](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7748275" \t "http://baike.sogou.com/_blank)等搭配使用，不需要离开Dremweaver开发工具便可完成。

在使用Adobe Dreamweaver开发网站的过程中，网站开发者可以通过使用[网站地图](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7648688" \t "http://baike.sogou.com/_blank)来设计制作网站的外观雏形、更新和重组网页，便于网站的管理。Dreamweaver是唯一提供了Roundtrip HTML、视觉化编辑与原始码编辑同步的设计工具。DW还拥有优秀的代码自动完成功能，提供给网站开发者轻松的开发体验。DW在开发过程中，可以随时切换代码编辑、实时视图编辑以及上半部分是实时视图，下半部分是代码编辑的拆分编辑，这三种编辑方式可以同步使用，十分方便。

Dreamweaver拥有可响应的自适应网格版面，可用于基于Intel或者PowerPC的Mac、MicrosoftWindowsXP以及WindowsVista系统。使用DW设计的网页，可以在所有主流的浏览器上打开，并且网站开发者可以通过在DW上编写代码，解决网站不兼容的问题，使用不同的浏览器，检测模块的功能。

### 2.1.2 Angular框架

Angular JS是一款由Misko Hevery 等人创建，后为Google所收购的优秀的前端JavaScript框架。AngularJS的特点有很多，其中核心的是：MVC模块化、语义化标签、自动化双向数据绑定、依赖注入等等。它使用简单，易于维护，通过使用了指令(directives)的结构来弥补HTML在构建应用这一方面的缺陷，让新语法能够为不同的浏览器所识别。例如：进行数据绑定时使用的双大括号{{}}语法、实现迭代或者隐藏DOM片段所使用的DOM控制结构等等。Angular JS把成为WEB应用中的一种客户端的解决方案作为目标，因此，网站开发者在使用Angular JS时，没有太多其他的方式来创建增加、删除、修改、查询应用时，但是在构建完增删改查后，还是能够灵活变通的。

## 2.2后台开发工具及框架

### 2.2.1 IDEA

IDEA 的全称是IntelliJ IDEA。它是[java语言](http://baike.baidu.com/view/229611.htm" \t "http://baike.baidu.com/_blank)开发的集成环境，被公认为是业界内最好的java开发工具之一，尤其在智能代码助手、代码自动提示等方面有着超常的的功能。其中，最超常的功能就是它的调试功能。它可以设置断点，甚至可以设置多个断电开启调试，也可以开启调试会话，还可以进行单步调试。IntelliJ IDEA对XML有着完美的支持，它可以智能选取某个方法，使用快捷键Ctrl+W就可以不断扩充选取的范围，对重构十分有利。

IntelliJ IDEA还提供了导航模式的特色功能，可以显示最近打开过的文件，以及可以根据网站开发者的需要，提供所需显示的类名查找框。它的历史记录功能也让不小心丢失文件的网站开放者能够轻松便捷地恢复文件。它的灵活的排版功能也让网站开发者能够针对不同的项目进行排版。此外，它还具有智能代码的功能，能够根据网站开发者所写的代码给出优化过后的代码，如果程序员同意，则会将优化过后的代码自动替换掉原来的代码，以此来提高编程的效率。

### 2.2.2 Spring MVC框架

Spring MVC 是MVC的一个流程框架，是Spring框架中最重要的模块，它能够更加灵活地进行流程处理，与Spring框架进行无缝集成。Spring MVC框架提供了大量的功能，比如将每一个角色划分给专门的对象来实现，Spring MVC将角色划分为控制器、命令对象、处理器映射、视图解析器等。Spring的框架高度可配置，Spring MVC对使用的视图是透明的，因此，在这个框架内，网站开发者可以使用JSP之外的技术。Spring MVC通过实现模块-视图-控制器模式，即Model-View-Controller模式将控制器、模型对象、过滤器和处理程序对象的角色进行分离。

Spring MVC比单纯的MVC构架更加地灵活，因此它很容易就能够和其他的视图框架进行集成。它的基础是功能强大的Spring IOC容器，简化了Spring MVC，使它便于测试。在现代的WEB开发中，MVC占据了不容小觑的一席之地，网站开发者在选择Spring MVC进行WEB项目的开发时，采用Spring框架的过滤器，可以解决出现乱码的问题，提高网站开发的效率。

## 2.3 MySQL数据库

MySQL是由瑞典MySQL AB公司开发的一个小型关系型数据库管理系统，它具有速度快、占用少、体积小、成本低的优势。其中，MySQL开放源码这一做法，受到很多为了降低网站总体拥有成本的一些中小型网站的追捧。不仅仅是很多中小型网站使用MySQL作为网站的数据库，大型网站也可以使用MySQL这一能够处理上千万条记录的大型数据库。

MySQL在使用过程中，是需要用C或C++进行代码的编写的，因为编译时使用了多种编译器进行编译，因此它的可移植性非常高。同时，MySQL提供了API，让网站开发者可以使用更多种类的编程语言在MySQL数据库中进行编译例如：Java、PHP等等。MySQL具备用于优化数据库的管理工具，且能够支持多线程运行，最大化利用CPU资源，提高使用效率。不仅如此，它优秀的SQL查询算法，也为查询速度的提高做出了贡献。

## 2.4 本章小结

本章介绍了设计和实现基于WEB的网上订餐系统所需要的开发工具和关键技术，包括前端的开发工具：Dreamweaver CC 2017和前端的JS框架：Angular JS，以及后台所用到的集成环境：IntelliJ IDEA、后台的框架：Spring MVC框架和开发工具：MySQL。

# 第三章 基于WEB的网上订餐系统的需求分析和设计

## 3.1系统分析

为了设计出功能完善的目标系统，首先需要基组成于WEB的网上订餐系统的各个模块进行需求分析。功能模块是一个包含了数据说明和可执行语句等对象的集合，每个功能模块负责实现一个子功能。这些许许多多的功能模块组合起来，可以形成一个网上订餐系统的整体。在通过自顶而下细化求精分析网上订餐系统的需求后，可以将网上订餐系统划分为以下五个功能模块：“顾客”模块、“商家”模块、“配送员”模块、“系统管理员”模块、“商家后台”模块。

### 3.1.1“顾客”模块需求分析

顾客是网上订餐系统的一个主体，作为网上订餐系统的使用者，顾客的体验感受尤为重要。在面向数据流对网上订餐系统进行了自定而下细化求精分析之后，我们可以从顾客的角度来设计，顾客功能模块中应当包含以下的几个功能：

1. 注册、登录；顾客在主页右上角点击登录/注册进入登录/注册页进行注册个人信息，登录网站；
2. 修改个人资料，包括修改头像、完善个人信息、增加收货地址、修改联系方式、修改密码、注销账户等等；
3. 浏览餐品；在主页上浏览餐品；
4. 收藏餐品；将喜爱的餐品添加到收藏中；
5. 搜索餐品；根据关键词，在餐品分类页和餐品筛选页搜索和选择餐品；
6. 提交评价；在评价页给已买商品写评价，评价内容分为文字评价和星级评价；
7. 提交订单；顾客可以查看待支付或已支付状态的订单；
8. 支付订单；按照订单上的价格在支付页进行支付，支付的款项在确认收货之前将暂存在网上订餐平台中；
9. 确认送达；在配送员将餐品送达后，点击确认送达，网上订餐平台内暂存的款项会被调入到商家后台管理平台中。

整理了以上顾客模块的功能需求后，网上订餐系统的“顾客”模块功能结构图，如图3-1所示：

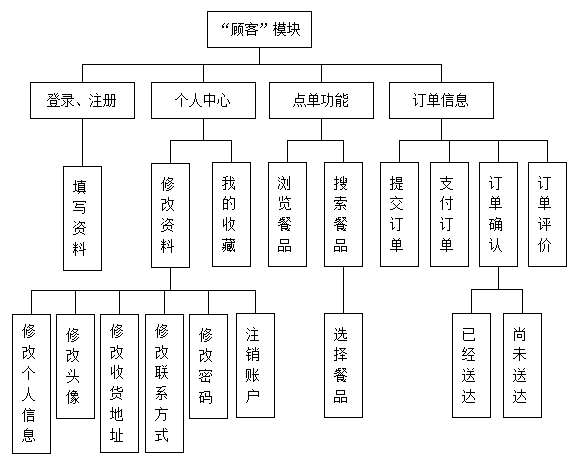


图3-1 网上订餐系统“顾客”模块功能结构图

用例规约：

|  |  |
| --- | --- |
| 用例名称 | 顾客 |
| 角色 | 买家 |
| 用例说明 | 该用例描述了顾客在该模块中可以进行注册、登录、修改个人信息、浏览餐品、根据关键词搜索餐品、选择餐品、提交订单、支付订单、评价订单的功能。 |
| 前置条件 | 顾客必须先注册一个账户，才能进行购买餐品的活动。 |
| 基本事件流 | 1. 顾客浏览餐品信息，选择想要的餐品； 2. 点击提交外卖的订单； 3. 顾客支付订单，卖家接收订单后开始安排餐品的制作； 4. 顾客可以查看订单，并对外卖商家进行催单； 5. 顾客收到外卖后可以对本次用餐的体验进行评价。 |
| 其他事件流 | 1. 修改个人资料、设置头像、修改密码；  2. 注销账户；  3. 与外卖配送员共同确认订单送达情况。 |

表3-1“顾客”模块用例规约

“顾客”模块的用例图如图3-2所示：

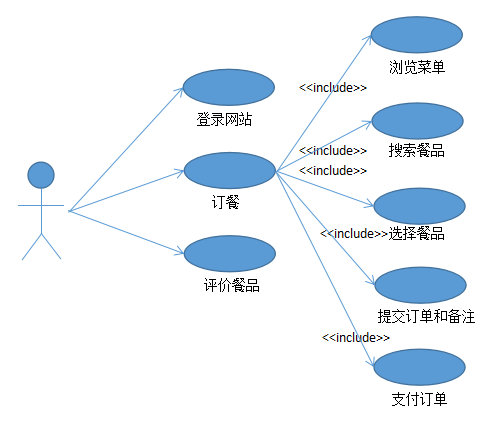


图3-2 网上订餐系统“顾客”模块用例图

### 3.1.2“商家”模块需求分析

商家作为网上订餐系统的另一个主体，也是网上订餐系统必不可少的组成部分。商家通过在网上订餐平台上推广自己的外卖餐品，增加顾客人数，并给顾客让利，实现双方双赢的局面。从商家的角度来看，清晰的功能划分、简单的操作方法可以吸引商家在网上订餐系统上的入驻。秉持着为顾客服务，对顾客负责的精神，在商家登录系统这一平台上，网上订餐系统平台应该把好关，杜绝一些黑心的不法商家浑水摸鱼。现在将“商家”模块的功能细分为如下几点：

1. 登录、注册；商家在注册时，必须持有本人的身份证明、食品生产安全许可证等相关证件，并且必须签订保证食品安全的合约；
2. 注销账户；商家可以在条约许可的条件下注销自己在网上订餐系统上的账户；
3. 发布和更新菜单；商家在网上订餐系统的平台上发布自己所能提供的餐品，上传菜品的真实图片，不得作假，商家也可以受季节变化、食材供应等外在因素的影响来变换每天提供的菜品，及时作出调整和修改；
4. 接收订单；在顾客对订单进行支付完成操作后，商家可以根据实际情况选择接收订单或拒接订单；
5. 查看订单；商家在接收顾客订单后，可以查看到订单的订单号、顾客选择的菜品内容、顾客的送餐地址等相关信息；
6. 订单反馈；顾客对商家售出的外卖商品进行评价后，商家可以在网上订餐平台上查看顾客的建议，在今后的销售过程中进行改进，为顾客提供更好的服务。

整理了以上商家模块的功能需求后，网上订餐系统的“商家”模块功能结构图，如图3-3所示：

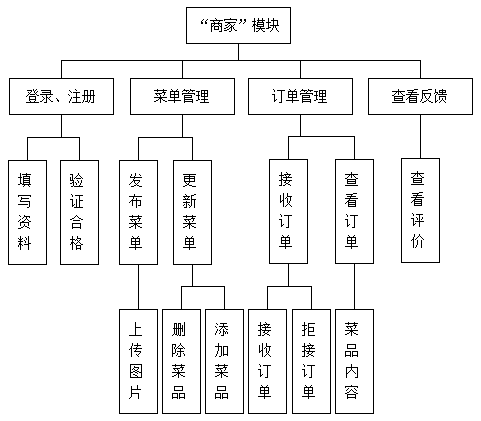


图3-3 网上订餐系统“商家”模块功能结构图

用例规约：

|  |  |
| --- | --- |
| 用例名称 | 商家 |
| 角色 | 卖家 |
| 用例说明 | 该用例描述了商家在该功能模块中进行注册和登录，验证商家制作的食品安全性、发布和更新菜单、接收订单、查看订单以及在网上订单平台上查看顾客的用餐体验的功能，来改进自己的服务质量。 |
| 前置条件 | 商家必须先验证自己提供的食品安全生产许可证的真实性，签订保证生产出的食品安全的条约，才能在网上订餐平台上发布自己店面的菜单。 |
| 基本事件流 | 1. 商家填写自己的注册信息，提供真实的食品安全生产许可证，并验证成功； 2. 商家在网上订餐平台上发布菜单详情； 3. 顾客提交并支付订单后，商家可以根据天气情况、道路情况等外在情况选择是接收订单还是拒接订单； 4. 商家在接收订单之后，可以查看订单，记录顾客所点的菜肴，安排出时开始制作外卖； 5. 商家同时配送员来取餐。 |
| 其他事件流 | 1. 商家在网上订餐平台上注销账户； 2. 商家根据实际情况更新菜单，上传菜品的实物图，以及列出菜肴所用的食材； 3. 商家在网上订餐系统的平台上查看顾客的用餐评价和用餐体验，以此提升自己的服务质量。 |

表3-2“商家”模块用例规约

“商家”模块的用例图如图3-4所示：

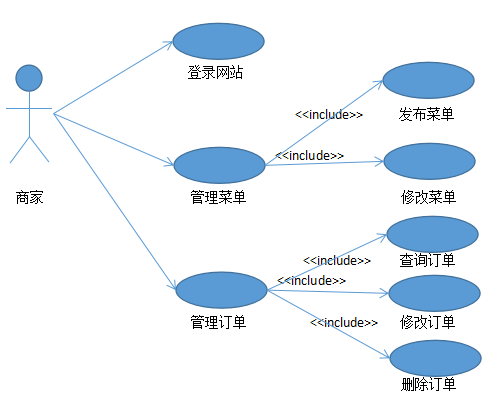


图3-4 网上订餐系统“商家”模块用例图

### 3.1.3“配送员”模块需求分析

配送员在网上订餐系统中扮演的角色是连接其中两个重要的主体，即顾客和商家。配送员从网上订餐系统中接收到商家的通知配送信息后，到达商家所在地提取外卖，再从系统中查看顾客的地址，将外卖送达目的地，注意送达时间不能超过预计时间，否则将要进行超时赔付。从配送员的角度来看，在接收到外卖商家的通知时，配送员需要从网上订餐系统中获得三个信息，即：外卖商家的地址、顾客所在地的地址、最迟所需的送达时间。配送员还需要与顾客一起确认外卖是否送达。从外卖配送员的角度来看，配送员功能可以划分为如下几个部分：

1. 接收通知；餐厅在准备好外卖后通过网上订餐系统通知配送员前来取餐，配送员接收到通知后，立刻前往商家所在地；
2. 查看订单；外卖配送员在接收到商家的取餐通知后，在网上订餐系统平台上查看商家的地址、顾客的地址、最迟送达时间这三条重要信息；
3. 确认送达；配送员在将外卖送达顾客的手中时，与顾客同时确认外卖的送达、订单最后一步的完成，在遇到特殊情况时，比如外卖餐品受损、联系不到顾客时，可以选择取消送餐。

整理了以上配送员模块的功能需求后，网上订餐系统的“配送员”模块功能结构图，如图3-5所示：

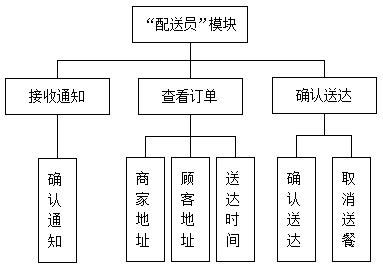


图3-5 网上订餐系统“配送员”模块功能结构图

用例规约：

|  |  |
| --- | --- |
| 用例名称 | 配送员 |
| 角色 | 送货者 |
| 用例说明 | 该用例描述了配送员在该功能模块中接收商家通知取餐、确认收到通知、查看订单详情，获取商家地址信息、顾客地址信息、最迟送达时间信息、确认送达的功能。在这个环节中，由于每一个配送员对应几个商家，所以需要配送员自己进行统筹安排，保证能在最后送达时间到来之前将外卖送到顾客手中。 |
| 前置条件 | 配送员必须提供健康证明才能在网上订餐系统中进行注册，继而使用网上订餐系统，查询信息。 |
| 基本事件流 | 1. 商家将外卖准备好后，通知配送员取餐，给配送员发送通知，配送员确认收到通知； 2. 配送员在网上订餐系统中查询商家地址信息、顾客地址信息、最迟送达时间信息，并前往商家所在地取餐，送至顾客所在地； 3. 配送员将外卖送到顾客手中后，与顾客一起在网上订餐系统里确认外卖已经送达。 |
| 其他事件流 | 1. 配送员注销自己的账户； 2. 配送员因为外卖餐品损毁、联系不到顾客或者其他原因选择取消送餐。 |

表3-3“配送员”模块用例规约

“配送员”模块的用例图如图3-6所示：

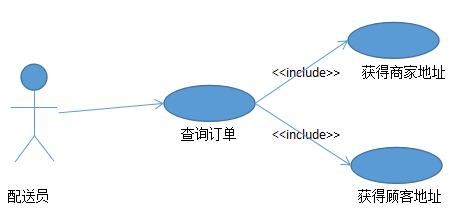


图3-6 网上订餐系统“配送员”模块用例图

### 3.1.4“系统管理员”模块需求分析

系统管理员在网上订餐系统中起到了一个综合信息、传输信息的作用。系统管理员存储了顾客的信息、商家的信息，对系统进行维护。顾客在提交订单后由系统管理员负责生成订单，在外卖餐品确认送达之前，系统管理员将暂时保管顾客支付订单的款额，在顾客确认送达后，商家从系统管理员处获取暂存在系统管理员处的款项。从系统管理员的角度来看，系统管理员的功能被分为如下几个方面：

1. 登录、注册；系统管理员权限很大，因此必须由可信的人作为系统管理员登录网上订餐系统；
2. 维护系统；
3. 信息管理；顾客个人信息、商家注册信息以及配送员信息都保存在系统管理员中，系统管理员将对这三个主体的信息进行维护管理；
4. 生成订单；顾客在提交并支付订单后，系统管理员将生成一份包含订单编号、顾客信息、顾客地址、点餐商家、餐品信息、预计送达时间等的订单。

整理了以系统管理员模块的功能需求后，网上订餐系统的“系统管理员”模块功能结构图，如图3-7所示：

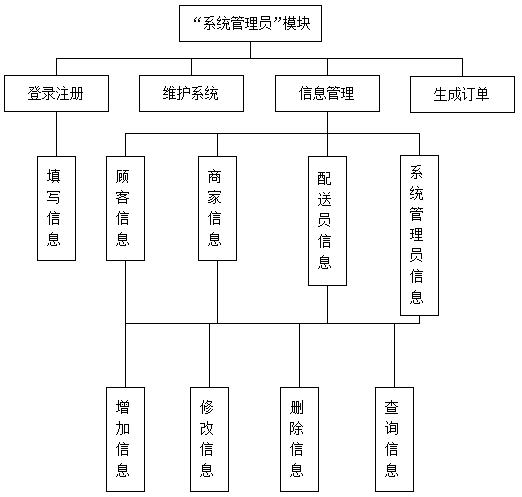


图3-7 网上订餐系统“系统管理员”模块功能结构图

用例规约：

|  |  |
| --- | --- |
| 用例名称 | 系统管理员 |
| 角色 | 管理者 |
| 用例说明 | 该用例描述了系统管理员在该功能模块中登录注册、维护系统、进行信息管理以及生成订单的功能；系统管理员可以管理顾客的个人信息、商家的信息、配送员的个人信息、系统管理员自己的个人信息，并且调度给不同的对象，比如：将顾客的地址信息和商家的地址信息提供给外卖配送员，将顾客所点餐品信息通过生成订单的形式传递给商家的厨师进行外卖的制作。 |
| 前置条件 | 系统管理员必须有可靠的信用记录，才能够注册登录为网上订餐系统的系统管理员。 |
| 基本事件流 | 1. 系统管理员注册并登陆网上订餐系统； 2. 系统管理员在顾客注册账户后，记录顾客的信息； 3. 系统管理员在顾客点单后，生成订单，并将将顾客的餐品信息发送给对应商家； 4. 在商家完成外卖制作，通知外卖配送员进行取餐时，将商家的地址信息，顾客的地址信息、送达时间信息调度给配送员使用。 |
| 其他事件流 | 1. 系统管理员对网站进行维护； 2. 系统管理员对商家信息、顾客信息、配送员信息进行管理。 |

表3-4“系统管理员”模块用例规约

“系统管理员”模块的用例图如图3-8所示：

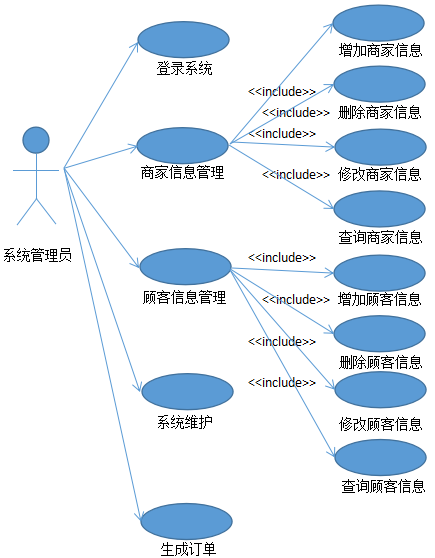


图3-8 网上订餐系统“系统管理员”模块用例图

### 3.1.5“商家后台”模块需求分析

商家在给交易过程当中，库存等也都在变化，因此需要一个商家后台的功能模块来对商家的每日营业情况进行统计和结算。商家每销售出一份外卖，库存就减少一份，相应的，当日的收入额也会随之增加。“商家后台”这一功能模块将会对商家的每日销售收入进行结算，结算出日收入、月收入、年收入后，商家可以根据销售情况的规律，调整下一阶段的菜单，以达到最好的服务状态。因此，从商家的视角来看，网上订餐系统还需要一个后台管理的功能，具体可以细化为以下的两个功能：

1. 查看库存；商家通过在网上订餐系统上的商家后台模块中查看库存，当库存不够时，可以对库存进行及时补充；
2. 每日结算；外卖商家的整体运作是离不开员工的，所以在进行每日结算时，要算入员工的工资成本。因为外卖配送员是一片区域的餐厅共同拥有的，所以每个餐厅应当给配送员一定的薪水，这也将算入成本中。总结说来，每日结算由员工工资、配送员工资、店铺租金、销售额、食材成本、餐具成本、其他损坏组成。

整理了以商家后台模块的功能需求后，网上订餐系统的“商家后台”模块功能结构图，如图3-9所示：

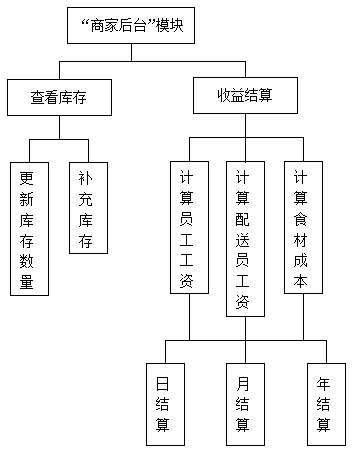


图3-9 网上订餐系统“商家后台”模块功能结构图

用例规约：

|  |  |
| --- | --- |
| 用例名称 | 商家后台 |
| 角色 | 后台 |
| 用例说明 | 该用例描述了商家在网上订餐系统中可以通过商家后台查看食材、餐具等商品的总库存、结算每日收益的功能。 |
| 前置条件 | 商家必须是网上订餐系统中的注册商家才能够使用系统中的商家后台 |
| 基本事件流 | 1. 商家接收订单后，商家后台模块自动将订单上的菜品所需要的食材从库存中扣除所需的分量； 2. 在顾客确认送达后，系统管理员处的暂存的销售金额自动转入商家后台的销售额中； 3. 商家后台根据销售额、各种成本计算当日营业额。 |
| 其他事件流 | 1. 在商家对库存进行补充后，手动更新库存的数量； 2. 商家后台对商家的销售情况每月进行月结算，每年进行年结算。 |

表3-5“商家后台”模块用例规约

“商家后台”模块的用例图如图3-10所示：

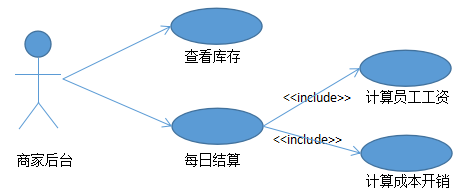


图3-10 网上订餐系统“商家后台”模块用例图

## 3.2系统设计

### 3.2.1架构设计

在设计基于WEB的网上订餐系统时，采用了Spring MVC框架，根据面向对象思想，将系统架构分为三层架构，分别是：表现层、业务层、持久层。在面向对象分析阶段，为了提高代码的可读性，增强代码地可维护性，将控制代码封装为控制类，即controller类。在分析阶段划分类，我们可以将类划分为：控制类、界面类、实体类及调用关系。Spring MVC 作为MVC的一个流程框架，是Spring框架中最重要的模块，Spring MVC 可以比MVC更加灵活地进行流程处理，与Spring框架进行无缝集成。在Spring MVC框架内，网站开发者可以使通过实现模块-视图-控制器模式，即Model-View-Controller模式，将控制器、模型对象、过滤器和处理程序对象的角色进行分离。

架构设计需要让系统具有强大的适应性、很好的可靠性、良好的安全性、可维护性、良好的扩展性等等。本网上订餐系统采用的软件构架是Spring MVC框架，Spring MVC框架的工作原理是如下：首先，用户给服务器发出一个请求，然后由DispatcherServlet来捕获这一请求。接着，DispatcherServlet对用户的URL进行解析，获取请求资源的标识符：URI。等到Handler执行完毕后，DispatcherServlet将得到一个返回的ModelAndView对象。视图（View）和模块（model）由视图处理器组合起来用于渲染视图，之后将结果返回给客户端。因为网上订餐系统这一WEB应用是基于MVC模式的，所以可以把Hibernate用作模型层/数据访问层。利用模型层将Java对象映射到数据库中，再用操作对象增添、删除、修改、查询数据表中的数据。

### 3.2.2概要设计

通过对网上订餐系统的功能模块的需求分析，在此基础上设计了网上订餐系统中的重要的功能模块的概要设计。由上文可知，本文将这个网上订餐系统划分了五个功能模块，分别是：“顾客”模块、“商家”模块、“配送员”模块、“系统管理员”模块、“商家后台”模块。以下将对这五个功能模块的概要设计进行详细的介绍。

**（1）“顾客”模块概要设计**

这个模块的功能主要是顾客在注册账户并登录账户后，在订餐网站上浏览餐品、搜索菜名、选择餐品、提交订单、支付订单后网购外卖商家出售的外卖餐品，并且可以收藏自己喜欢的店家或者食物，在消费后，对本次用餐进行公开的评价。其中顾客可以根据食物的品种分类逐步选择所要点的外卖，在提交并支付订单后，顾客可以随时查看外卖的状态，外卖的三个状态分别是：卖家已接单、正在配送、已经送达。以下是“顾客”模块中的某些重要功能的业务流程图：

①顾客登录网站流程图如图3-11所示：

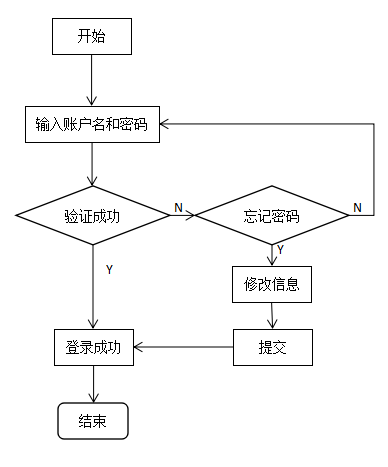


图3-11顾客登录流程图

②顾客提交外卖订单流程图

顾客提交外卖订单的流程中，包含了几个功能，首先，顾客在订餐网站中可以看到外卖商家发布的餐品，点击进入外卖商家发布菜品的页面后，顾客可以选择店铺中已经发布的餐品，提交订单。顾客也可以通过网站提供的分类，根据自己的要求，逐步选择自己喜欢的外卖，或者通过网站提供的筛选关键字，筛选出自己喜欢的食物，再提交订单。如果在提交订单的过程中，网络出现故障，或因为其他原因，导致提交订单失败，则返回上一层页面，重新选择餐品；如果在支付订单时支付失败，则返回上一层页面，即支付页面，顾客可以不用再次提交订单，直接重新支付已经提交过的订单即可。顾客提交外卖订单流程图如图3-12所示：

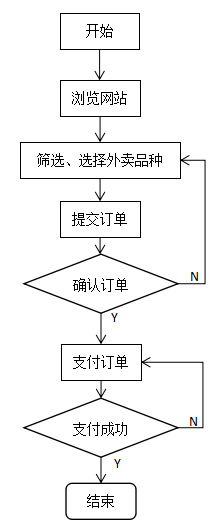


图3-12顾客提交外卖订单流程图

③顾客评价用餐情况流程图

顾客对用餐情况的评价对于其他顾客的选择有着至关重要的影响，从确认餐品送达开始，用餐顾客在用餐之后，登录网上订餐系统，填写自己本次的用餐体验，用餐体验部分分为两部分，先进行文字语言描述的评价，再给本次用餐体验打分，进行星级评价。顾客评价用餐情况流程图如图3-13所示：

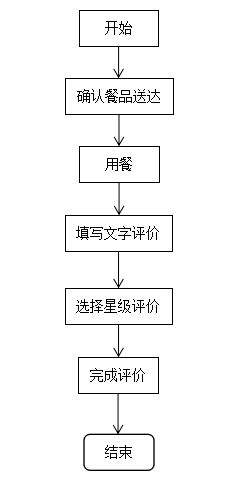


图3-13顾客评价用餐情况流程图

1. **“商家”模块概要设计**

“商家”模块的主要功能是商家登录、注册，选择是否接单、管理菜单、管理订单，这个模块也允许商家在网上订餐平台上查看顾客的评价和反馈。商家在这个模块可以通过注册账户得到注册商家编号，登录网上订餐系统后，根据食材的库存决定所发布的菜单内容，并且根据库存数量、种类即使增加、修改、删除菜品。商家可以根据自身情况决定是否接收顾客的订单，在接收顾客的订单后，根据订单上的菜品内容、顾客的备注内容，比如口味等要求，制作外卖。在顾客填写完评价后，商家可以在网上订餐系统上查看顾客的评价，根据顾客的评价改善商家的各项服务，以期得到更多的利润。以下是“商家”模块的一些重要功能的业务流程图：

①商家登录、注册网站流程图

商家入驻网上订餐系统的唯一前提就是保障制作的外卖从食材到餐具都能符合食品安全的标准。因此，在注册成为网上订餐网站上的商家，需要提供真实的食品安全许可证的编号，验证许可证真实性后，才可以成为网上订餐系统里的注册商家，发布菜品。商家登录、注册网站流程图如图3-14所示：

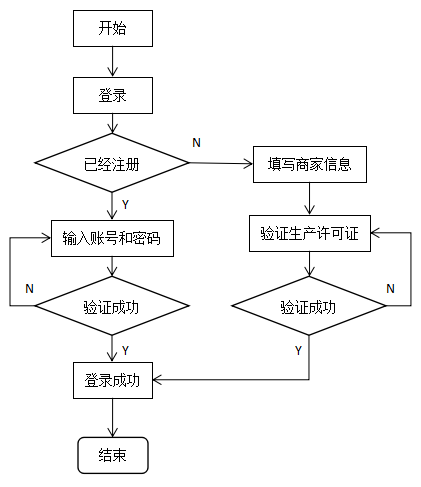


图3-14商家登录、注册网站流程图

②商家菜单管理流程图

商家对菜单进行管理，涉及到“商家后台”功能模块中的库存数量的问题。商家登录网站后，首先在“商家后台”模块中查看库存数量，当库存数量足够时，商家可以按照食材的数量和种类搭配出外卖菜品，发布到网上订餐系统上，当库存显示数量不足时，商家将在“商家”模块中，将数量不足的菜品进行下架操作，或者修改成为有足够库存的餐品。商家在发布菜单时，需要将外卖菜品的真实图片和所用食材发布到网上订餐系统中。商家菜单管理流程图如图3-15所示：

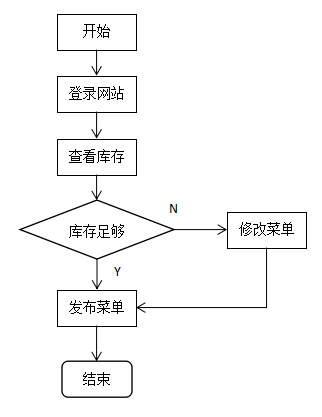


图3-15商家菜单管理流程图

③商家查看评价反馈情况流程图如图3-16所示：

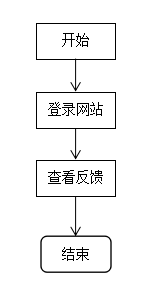


图3-16商家查看评价反馈情况流程图

1. **“配送员”模块概要设计**

“配送员”模块的主要功能是接收通知、查看订单、确认送达。配送员的登录、注册过程与顾客的登录注册过程类似，在此便不一一赘述。配送员收到商家的通知后，查看商家的地址，前往商家所在地取餐，配送员从网上订餐系统中还能查看顾客的地址以及预计送达时间，在到达顾客的所在地后通知顾客前来取餐，并与顾客一起进行外卖的确认送达。因此，配送员的主要业务流程图如图3-17所示：

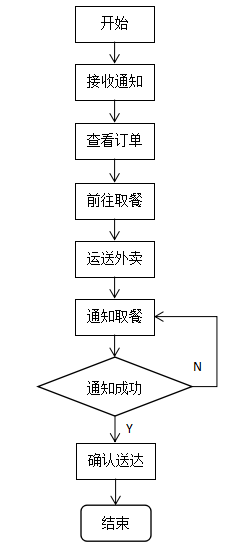


图3-17配送员的主要业务流程图

1. **“系统管理员”模块概要设计**

“系统管理员”模块的功能除了基本的登录、注册之外，还有维护系统、管理顾客、商家、配送员的注册信息、生成订单等。生成订单是指顾客在提交并且支付订单后，系统生成一个订单，将顾客选择的餐品信息和备注信息提供给外卖商家，将顾客的地址信息、商家的预计送达时间信息提供给配送员。系统管理员的主要业务流程图如图3-18所示：

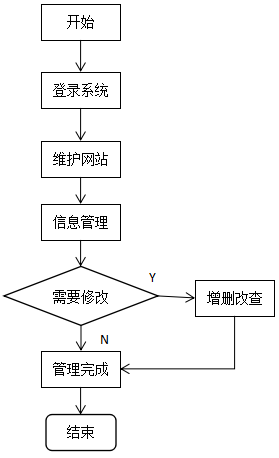


图3-18系统管理员的主要业务流程图

1. **“商家后台”模块概要设计**

“商家后台”模块的功能就是查看库存和结算收益。商家每日需要在商家后台查看当日的库存，再在“商家”模块中发布每日的餐品，每销售出一份餐品，库存将减少一份食材，当库存不足时，商家将外出购买食材来补充库存。在结算收益时，商家会根据库存中消耗的食材成本以及销售额，结合员工工资、配送员工资来计算当日的利润，并且给以每月总结和年终总结。以下是“商家后台”模块中的两个功能的业务流程图：

①商家后台查看库存流程图如图3-11所示：

商家及时查看库存不仅能够保障外卖制作者有源源不断的食材来进行外卖的制作，同时也能保证食材的新鲜度，及时更换不新鲜的食材。商家在看到外卖订单的菜品后，首先查看库存是否足够，如果食材库存充足，则接收订单，否则取消订单。接收订单后，商家安排厨师进行餐品的制作，并更新库存数量。商家后台查看库存流程图如图3-19所示：

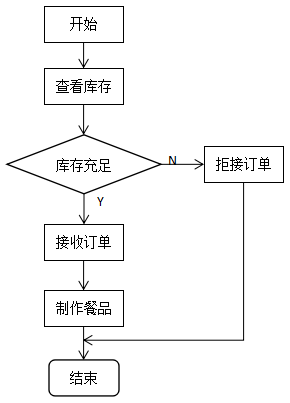


图3-19商家后台查看库存流程图

②商家后台结算收益流程图

商家后台在顾客确认送达后，可从网上订餐管理系统中获得订单的金额，商家后台根据开销各项成本和订单收益计算日、月、年收益。商家后台结算收益流程图如图3-20所示：

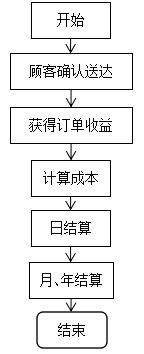


图3-20商家后台结算收益流程图

### 3.2.3数据库设计

网上订餐系统的数据库设计是按照商家、顾客、配送员的需求来设计的建立网上订餐系统的数据库，在网上订餐系统数据库中存储数据结构的一个过程。基于WEB的网上订餐系统采用的是MySQL这一关系型数据库来对系统中的数据进行存储处理。

（1）**实体-属性图：**

顾客的实体-属性图如图3-21所示：

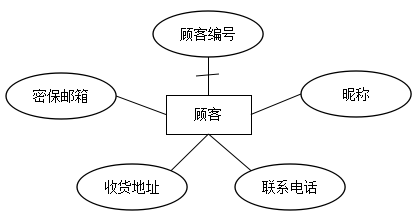


图3-21 顾客 实体-属性图

商家的实体-属性图如图3-22所示：

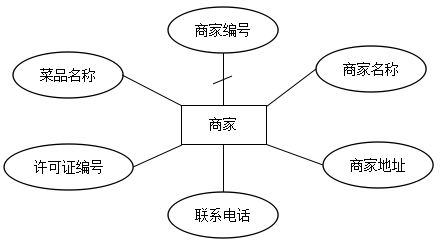


图3-22 商家 实体-属性图

配送员的实体-属性图如图3-23所示：

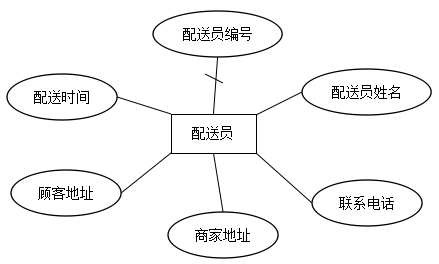


图3-23 配送员 实体-属性图

订单的实体-属性图如图3-24所示：

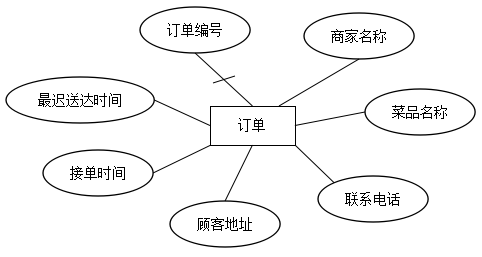


图3-24 订单 实体-属性图

菜品的实体-属性图如图3-25所示：

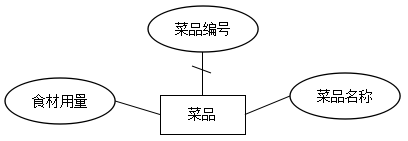


图3-25 菜品 实体-属性图

**（2）实体关系图：**

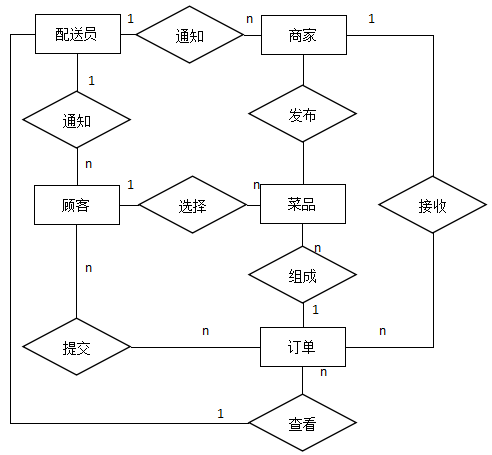


图3-26模块实体关系图

**（3）UML类图：**

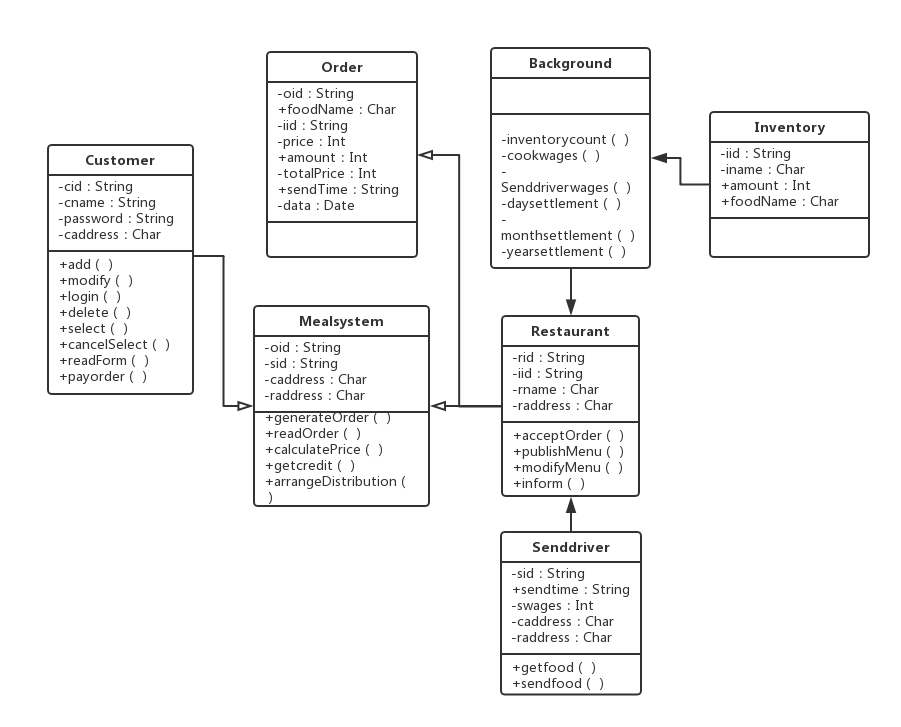


图3-27 UML类图

备注：

|  |  |
| --- | --- |
| 类名 | 说明 |
| Customer | 顾客类 |
| Order | 订单类 |
| Mealsystem | 点餐系统类 |
| Inventory | 存货类 |
| Restaurant | 餐厅类 |
| Senddriver | 送餐员类 |
| Background | 后台 |

表3-6 类名备注

|  |  |
| --- | --- |
| 方法名 | 说明 |
| Customer的add（）方法 | 增加顾客信息 |
| Customer的modify（）方法 | 修改顾客信息 |
| Customer的login（）方法 | 登录 |
| Customer的delete（）方法 | 注销 |
| Customer的select（）方法 | 选择餐品 |
| Customer的cancelSelect（）方法 | 取消选择 |
| Customer的readForm（）方法 | 查看表单 |
| Customer的payorder（）方法 | 支付订单 |
| Mealsystem的generateOrder（）方法 | 生成订单 |
| Mealsystem的readOrder（）方法 | 查看订单 |
| Mealsystem的calculatePrice（）方法 | 计算价格 |
| Mealsystem的arrangeDistribution（）方法 | 安排配送 |
| Mealsystem的getcredit（）方法 | 获取金额 |
| Restaurant的acceptOrder（）方法 | 接受订单 |
| Restaurant的publishMenu（）方法 | 发布菜单 |
| Restaurant的modifyMenu（）方法 | 修改菜单 |
| Restaurant的inform（）方法 | 通知收货 |
| Senddriver的getfood（）方法 | 餐馆取餐 |
| Senddriver的sendfood（）方法 | 送餐至目的地 |
| Background的inventorycount（）方法 | 后台计算库存 |
| Background的cookwages（）方法 | 计算厨师工资 |
| Background的Senddriverwages（）方法 | 送餐员工资 |
| Background的daysettlement（）方法 | 营业额日结算 |
| Background的monthsettlement（）方法 | 营业额月结算 |
| Background的yearsettlement（）方法 | 营业额年结算 |

表3-7 方法名备注

（4）网上订餐系统的活动图

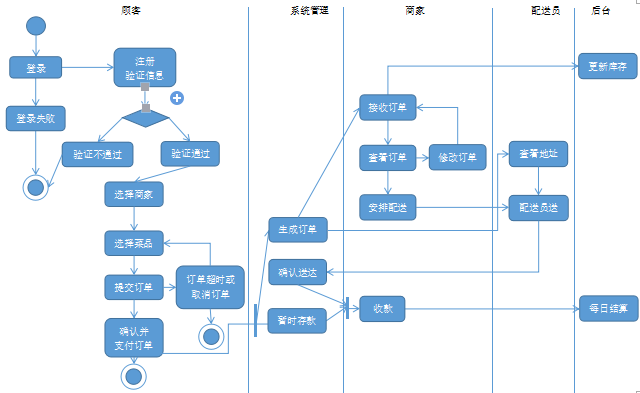


图3-28 网上订餐系统活动图

（5）逻辑结构设计

根据网上订餐系统的实体关系图，可以知道实体主要有：顾客、商家、配送员、订单、餐品这五项，针对这五类实体设计出如下关系数据模型：

顾客（**顾客编号**，顾客昵称，联系电话，收货地址，密保邮箱）

商家（**商家编号**，商家名称，商家地址，联系电话，许可证编号，菜品名称）

配送员（**配送员编号**，配送员姓名，联系电话，商家所在地址，顾客收货地址，最迟送达时间）

订单（**订单编号**，顾客编号，商家编号，商家名称，菜品名称，菜品价格，顾客联系电话，顾客地址，商家接单时间，最迟配送时间）

菜品（**菜品编号**，菜品名称，所属商家，食材用量）

# 系统功能的实现

## 4.1系统开发环境配置

一个恰当的、良好的开发环境是创建一个系统的必要前提和重要保障，基于WEB网上订餐系统这一项目的开发在经过对系统的需求分析、架构分析、概要设计和详细设计之后，决定采用以下开发环境进行项目的开发。基于WEB网上订餐系统使用B/S结构类型作为软件系统结构。

前端页面采用的是Angular框架。Angular是一款易于维护、操作简便的优秀的JavaScript框架。AngularJS的核心特点有MVC模块化、语义化标签、自动化双向数据绑定、依赖注入等等。作为一款具有优秀性能的JavaScript框架，Angular框架使用指令结构弥补了HTML在构建应用等方面的缺陷，它能够保障程序能够在全终端运行顺畅。

本网上订餐系统的后台框架采用的是Spring MVC框架，选用Java语言对订餐系统的后台进行开发。Spring MVC将角色划分为控制器、视图解析器、命令对象、处理器映射等等。Spring MVC通过实现Model-View-Controller模式完美地分离了控制器、模型对象、过滤器。在开发本网上订餐系统时，使用Spring MVC框架，能够灵活地将URL映射在页面控制器上。Spring框架高度可配置，自带绑定机制、异常处理方法、数据验证、单元测试，让开发过程变得简洁清晰。

网上订餐系统数据库选用了MySQL，结合Hibernate对象关系映射框架，用JDBC连接MySQL数据库，对JDBC进行轻量级数据对象的封装，让程序员能够在开发时运用面向对象的编程思维来随心所欲地对数据库进行相关操作。MySQL的可移植性很高，拥有用于优化数据库的管理工具，可以支持多线程运行，最大化利用CPU资源，具有很高的使用效率。MySQL有着优秀的查询算法，因此查询速度很快。

本系统使用集成开发环境IntelliJ IDEA进行开发，IntelliJ IDEA是用Java语言开发的集成环境。具有灵活的排版功能、优秀的调试功能。本网上订餐系统使用的就是IntelliJ IDEA集成开发环境。

## 4.2网上订餐系统的界面设计

本节主要从基于WEB的网上订餐系统的顾客端、商家端等几个功能模块中选取了两个重要的功能模块，介绍网上订餐系统的界面设计，并且展示这些功能模块的部分代码。基于WEB的网上订餐系统的界面设计是在网上订餐系统的基础上，对“顾客”功能模块、“商家”功能模块、“配送员”功能模块、“系统管理员”功能模块、“商家后台”功能模块进行的界面设计，并且保证网上订餐系统的各项功能能够实现。以下展示的是其中几个重要模块：“顾客”模块、“商家后台”的界面设计：

1. “顾客”模块界面设计

“顾客”模块中将展示顾客登录的界面、顾客浏览的首页的界面、顾客筛选菜品的界面、顾客评价外卖餐品的界面。

①顾客登录和注册的界面如图4-1、图4-2所示：

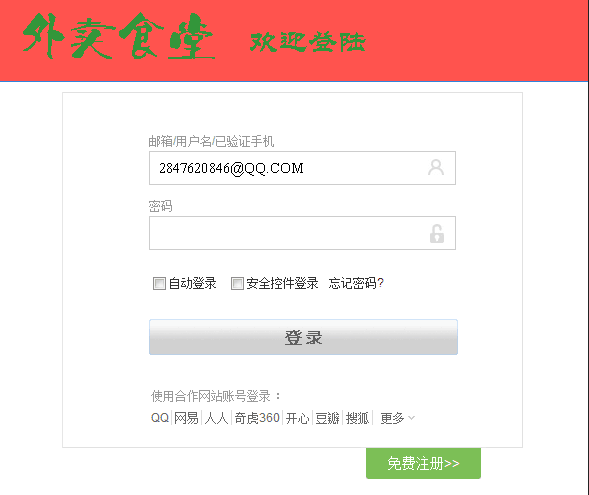


图4-1顾客登录的界面

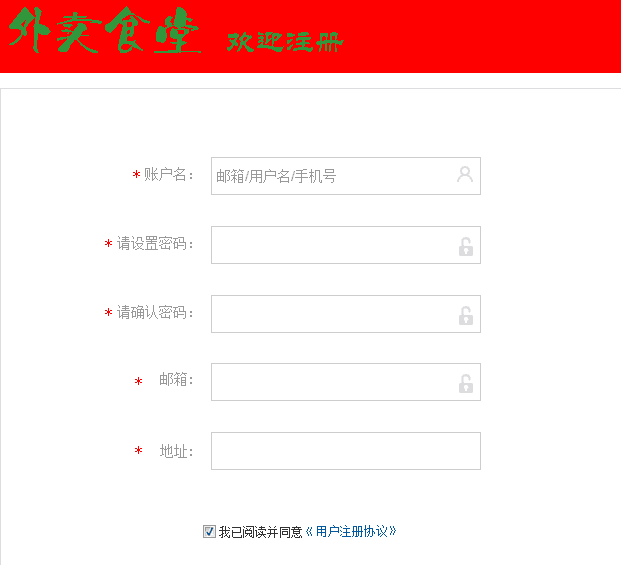


图4-2顾客注册的界面

②顾客浏览的网站首页界面如图4-2所示：

图4-2顾客浏览的网站首页界面

③顾客筛选菜品的界面如图4-3所示：

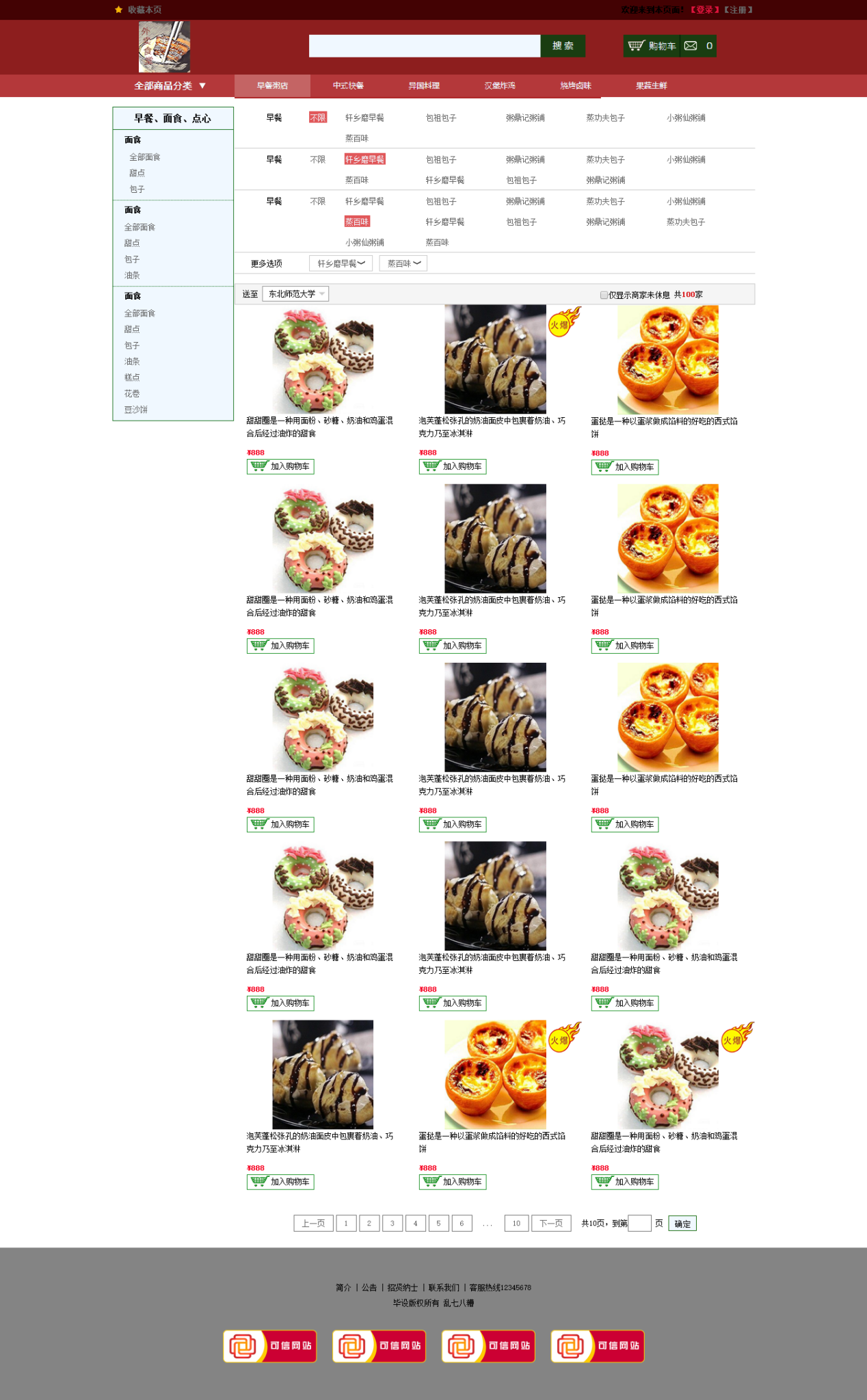


图4-3顾客筛选菜品的界面

（2）“商家后台”模块界面设计

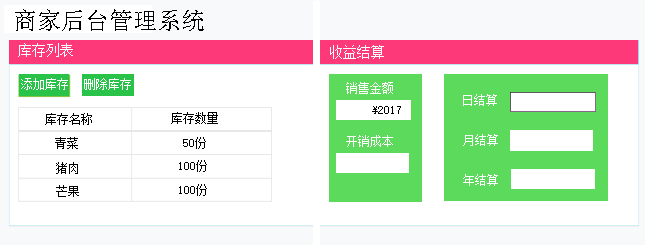


图4-4“商家后台”模块界面设计

## 4.2系统功能实现

### 4.2.1“顾客”模块实现

（1）顾客“登录注册”模块功能简介

顾客在“顾客”模块注册账号的操作，注册完账号后，顾客可以进行登录操作，继续在网上订餐系统上进行浏览选择菜品、提交订单、支付订单等后续操作。

1. 顾客“登录注册”模块核心代码

package com.emenu.web.interceptor;

import com.emenu.common.annotation.IgnoreLogin;

import com.emenu.common.entity.party.security.SecurityUser;

import com.emenu.common.utils.WebConstants;

import com.emenu.service.party.login.LoginManageService;

import com.emenu.service.party.security.SecurityUserService;

import com.pandawork.core.common.log.LogClerk;

import org.apache.shiro.subject.Subject;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.method.HandlerMethod;

import org.springframework.web.servlet.handler.HandlerInterceptorAdapter;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.util.Enumeration;

/\*\*

\* 登录拦截器

\*

\* @author: zhangteng

\* @time: 2017/5/7 15:26

\*\*/

public class LoginInterceptor extends HandlerInterceptorAdapter {

@Autowired

private LoginManageService loginManageService;

@Autowired

private SecurityUserService securityUserService;

@Override

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

String ssoServerURL = (String) request.getAttribute("website");

if (!(handler instanceof HandlerMethod)) {

// 若不是方法请求地址，则不拦截

return true;

}

Subject subject = loginManageService.isLogined(request);

// 已经登录过

if (subject != null) {

SecurityUser user = securityUserService.queryByLoginName((String) subject.getPrincipal());

// 将当前登录用户的LogiName、id放入cookie中

/\*CookieUtil.setCookie(response, CookieNameEnums.WebLoginName, String.valueOf(user.getUserName()));

CookieUtil.setCookie(response, CookieNameEnums.WebUserId, );\*/

this.setRequestAttribute(user, request);

return true;

}

// 判断类和方法是否有忽略登录

IgnoreLogin ignoreClazz = ((HandlerMethod) handler).getBean().getClass().getAnnotation(IgnoreLogin.class);

IgnoreLogin ignoreMethod = ((HandlerMethod) handler).getMethod().getAnnotation(IgnoreLogin.class);

if (ignoreClazz != null

|| ignoreMethod != null) {

return true;

}

LogClerk.sysout.debug("class or method IgnoreLogin annotation is null, means this request need interceptor");

// 执行到这里，说明用户没有记住登录

// ajax请求

if (((HandlerMethod) handler).getMethod().getAnnotation(ResponseBody.class) != null) {

response.getWriter().append("0");

response.getWriter().flush();

return false;

}

// 不是ajax请求

String serviceURL = this.getBasePath(request);

String rURL = serviceURL + request.getServletPath();

RequestMethod[] requestMethods = ((HandlerMethod) handler).getMethod().getAnnotation(RequestMapping.class).method();

boolean fountGet = false;

for (RequestMethod requestMethod : requestMethods) {

if (requestMethod.equals(RequestMethod.GET)) {

fountGet = true;

break;

}

}

if (!fountGet) {

rURL = "";

} else {

Enumeration<String> paramsEnum = request.getParameterNames();

StringBuffer sb = new StringBuffer();

while (paramsEnum.hasMoreElements()) {

String paramName = paramsEnum.nextElement();

String[] values = request.getParameterValues(paramName);

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

sb.append("&" + paramName + "=" + values[i]);

}

}

if (sb.length() > 0) {

String params = "?" + sb.toString().substring(1);

rURL += params;

}

}

String redirectURL = constructSspLoginUrl(ssoServerURL, rURL);

response.sendRedirect(redirectURL);

return false;

}

private String constructSspLoginUrl(String sUrl, String returnUrl) {

sUrl = sUrl + "admin/login?returnURL=" + returnUrl;

LogClerk.sysout.debug("sURL for login is " + sUrl);

return sUrl;

}

private String getBasePath(HttpServletRequest request) {

String basePath = request.getScheme() + "://" + request.getServerName() + ":" + request.getServerPort() + request.getContextPath();

return basePath;

}

private void setRequestAttribute(SecurityUser user, HttpServletRequest request) {

// 用户的userId

request.setAttribute(WebConstants.WEB\_SECURITY\_USER\_ID, user.getId());

// 用户partyId

request.setAttribute(WebConstants.WEB\_PARTY\_ID, user.getPartyId());

// 登录名 loginName

request.setAttribute(WebConstants.WEB\_USER\_LOGIN\_NAME, user.getLoginName());

}

}

### 4.2.2“商家”模块实现

1. “商家”餐品管理模块功能简介

商家在“商家”模块进行餐品的管理。

**“商家”餐品管理模块核心代码**

package com.emenu.web.controller.admin.meal;

import com.alibaba.fastjson.JSONObject;

import com.emenu.common.annotation.Module;

import com.emenu.common.entity.meal.MealPeriod;

import com.emenu.common.enums.meal.MealPeriodIsCurrentEnums;

import com.emenu.common.enums.meal.MealPeriodStatusEnums;

import com.emenu.common.enums.other.ModuleEnums;

import com.emenu.common.utils.URLConstants;

import com.emenu.web.spring.AbstractController;

import com.pandawork.core.common.exception.SSException;

import com.pandawork.core.common.log.LogClerk;

import com.pandawork.core.common.util.Assert;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.\*;

import org.springframework.web.servlet.mvc.support.RedirectAttributes;

import javax.servlet.http.HttpServletRequest;

import java.util.List;

/\*\*

\* MealPeriodController

\*

\* @author Wang Liming

\* @date 2015/11/11 9:31

\*/

@Controller

@Module(ModuleEnums.AdminRestaurantMealPeriod)

@RequestMapping(value = URLConstants.ADMIN\_MEAL\_PERIOD\_URL)

public class AdminMealPeriodController extends AbstractController{

/\*\*

\* 去餐品管理页面

\*

\* @param model

\* @return

\*/

@Module(ModuleEnums.AdminRestaurantMealPeriodList)

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

public String toMealPeriodPage(Model model){

try {

MealPeriod currentMealPeriod = mealPeriodService.queryByCurrentPeriod(MealPeriodIsCurrentEnums.Using);

List<MealPeriod> mealPeriodList = mealPeriodService.listAll();

model.addAttribute("currentMealPeriod", currentMealPeriod);

model.addAttribute("mealPeriodList", mealPeriodList);

} catch (SSException e) {

LogClerk.errLog.error(e);

sendErrMsg(e.getMessage());

return ADMIN\_SYS\_ERR\_PAGE;

}

return "admin/restaurant/meal/period/list\_home";

}

/\*\*

\* 新增餐品

\*

\* @param mealPeriod

\* @return

\*/

@Module(ModuleEnums.AdminRestaurantMealPeriodNew)

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

public String newMealPeriod(MealPeriod mealPeriod, HttpServletRequest httpServletRequest,

RedirectAttributes redirectAttributes){

try {

mealPeriodService.newMealPeriod(mealPeriod);

String successUrl = "/" + URLConstants.ADMIN\_MEAL\_PERIOD\_URL;

redirectAttributes.addFlashAttribute("msg", NEW\_SUCCESS\_MSG);

return "redirect:" + successUrl;

} catch (SSException e) {

LogClerk.errLog.error(e);

sendErrMsg(e.getMessage());

String failedUrl = "/" + URLConstants.ADMIN\_MEAL\_PERIOD\_URL;

redirectAttributes.addFlashAttribute("msg", e.getMessage());

return "redirect:" + failedUrl;

}

}

/\*\*

\* ajax修改餐品

\*

\* @param mealPeriod

\* @return

\*/

@Module(ModuleEnums.AdminRestaurantMealPeriodUpdate)

@RequestMapping(value = "ajax", method = RequestMethod.PUT)

@ResponseBody

public JSONObject updateMealPeriod(MealPeriod mealPeriod){

try {

mealPeriodService.updateMealPeriod(mealPeriod);

return sendJsonObject(AJAX\_SUCCESS\_CODE);

} catch (SSException e) {

LogClerk.errLog.error(e);

return sendErrMsgAndErrCode(e);

}

}

/\*\*

\* ajax删除餐品

\*

\* @param mealPeriodId

\* @return

\*/

@Module(ModuleEnums.AdminRestaurantMealPeriodDel)

@RequestMapping(value = "ajax/{mealPeriodId}", method = RequestMethod.DELETE)

@ResponseBody

public JSONObject delById(@PathVariable("mealPeriodId") Integer mealPeriodId){

try {

mealPeriodService.delById(mealPeriodId);

return sendJsonObject(AJAX\_SUCCESS\_CODE);

} catch (SSException e) {

LogClerk.errLog.error(e);

return sendErrMsgAndErrCode(e);

}

}

/\*\*

\* ajax修改启用状态

\*

\* @param mealPeriodId

\* @param status

\* @return

\*/

@Module(ModuleEnums.AdminRestaurantMealPeriodUpdate)

@RequestMapping(value = "ajax/status/{mealPeriodId}", method = RequestMethod.PUT)

@ResponseBody

public JSONObject updateStatusById(@PathVariable("mealPeriodId") Integer mealPeriodId,

@RequestParam("status") Integer status){

try {

mealPeriodService.updateStatusById(mealPeriodId, MealPeriodStatusEnums.valueOf(status));

return sendJsonObject(AJAX\_SUCCESS\_CODE);

} catch (SSException e) {

LogClerk.errLog.error(e);

return sendErrMsgAndErrCode(e);

}

}

/\*\*

\* ajax修改当前餐品

\*

\* @param mealPeriodId

\* @return

\*/

@Module(ModuleEnums.AdminRestaurantMealPeriodUpdate)

@RequestMapping(value = "ajax/current/{mealPeriodId}", method = RequestMethod.PUT)

@ResponseBody

public JSONObject updateCurrentMealPeriod(@PathVariable("mealPeriodId") Integer mealPeriodId){

try {

mealPeriodService.updateCurrentMealPeriod(mealPeriodId, MealPeriodIsCurrentEnums.Using);

return sendJsonObject(AJAX\_SUCCESS\_CODE);

} catch (SSException e) {

LogClerk.errLog.error(e);

return sendErrMsgAndErrCode(e);

}

}

/\*\*

\* ajax获得当前最后排序加1的值

\*

\* @return

\*/

@RequestMapping(value = "ajax/weight", method = RequestMethod.GET)

@ResponseBody

public JSONObject getLastWeight(){

try {

List<MealPeriod> mealPeriodList = mealPeriodService.listAll();

MealPeriod mealPeriod = mealPeriodList.get(mealPeriodList.size() - 1);

JSONObject jsonObject = new JSONObject();

jsonObject.put("weight", mealPeriod.getWeight() + 1);

return sendJsonObject(jsonObject, AJAX\_SUCCESS\_CODE);

} catch (SSException e) {

LogClerk.errLog.error(e);

return sendErrMsgAndErrCode(e);

}

}

}

### 4.2.3“商家后台”模块实现

（1）“商家后台”模块功能简介

在商家后台的餐品管理中，商家每销售出一份餐品就对餐品库存进行增、删、改、查的操作。

**（2）“商家后台”库存模块核心代码**

package com.emenu.web.controller.admin.other;

import com.alibaba.fastjson.JSON;

import com.alibaba.fastjson.JSONArray;

import com.alibaba.fastjson.JSONObject;

import com.emenu.common.annotation.IgnoreAuthorization;

import com.emenu.common.annotation.IgnoreLogin;

import com.emenu.common.entity.storage.StorageItem;

import com.emenu.common.utils.StringUtils;

import com.emenu.common.utils.URLConstants;

import com.emenu.web.spring.AbstractController;

import com.pandawork.core.common.exception.SSException;

import com.pandawork.core.common.log.LogClerk;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.ResponseBody;

import java.math.BigDecimal;

import java.util.Collections;

import java.util.List;

/\*\*

\* 后台管理一些通用方法的controller

\*

\* @author: zhangteng

\* @time: 2015/11/11 9:21

\*\*/

@IgnoreLogin

@Controller

@RequestMapping(value = URLConstants.ADMIN\_COMMON\_URL)

public class AdminCommonController extends AbstractController {

/\*\*

\* 根据汉字获取拼音

\*

\* @param str

\* @param type

\* @return

\*/

@IgnoreAuthorization

@RequestMapping(value = "tool/str2py/ajax", method = RequestMethod.GET)

@ResponseBody

public JSON ajaxStr2Py(@RequestParam("str") String str,

@RequestParam(value = "type", defaultValue = "head") String type) {

try {

String res = "";

if ("head".equals(type)) {

res = StringUtils.str2Pinyin(str, StringUtils.PINYIN\_TYPE\_HEAD\_CHAR);

} else {

res = StringUtils.str2Pinyin(str, StringUtils.PINYIN\_TYPE\_FULL);

}

JSONObject jsonObject = new JSONObject();

jsonObject.put("pinyin", res);

return sendJsonObject(jsonObject, AJAX\_SUCCESS\_CODE);

} catch (Exception e) {

LogClerk.errLog.error(e);

return sendJsonObject(AJAX\_FAILURE\_CODE);

}

}

/\*\*

\* 根据关键字获取库存物品名字列表

\*

\* @param keyword

\* @return

\*/

@IgnoreAuthorization

@RequestMapping(value = "tool/storage/item/search/ajax", method = RequestMethod.GET)

@ResponseBody

public JSON ajaxSearchStorageItem(@RequestParam("keyword") String keyword) {

List<StorageItem> storageItemList = Collections.emptyList();

try {

storageItemList = storageItemService.listByKeyword(keyword);

} catch (SSException e) {

LogClerk.errLog.error(e);

return sendErrMsgAndErrCode(e);

}

JSONArray jsonArray = new JSONArray();

for (StorageItem storageItem : storageItemList) {

JSONObject jsonObject = new JSONObject();

jsonObject.put("id", storageItem.getId());

jsonObject.put("name", storageItem.getName());

jsonObject.put("assistantCode", storageItem.getAssistantCode());

BigDecimal price = new BigDecimal("0.00");

if (storageItem.getTotalStockInQuantity().equals(BigDecimal.ZERO)) {

price = storageItem.getTotalStockInMoney().divide(storageItem.getTotalStockInQuantity(), 10, BigDecimal.ROUND\_HALF\_DOWN);

price = price.setScale(2, BigDecimal.ROUND\_HALF\_DOWN);

}

jsonObject.put("price", price);

jsonArray.add(jsonObject);

}

return sendJsonArray(jsonArray);

}

}

# 第五章 系统测试与维护

在对基于WEB的网上订餐系统进行测试维护的过程中，首先要检测浏览器的页面显示是否正常，接着分别检测“顾客”模块、“商家”模块、“配送员”模块、“系统管理员”模块、“商家后台”模块这五个功能模块是否能够实现设计时预计实现的功能。因此这与传统的软件测试相比，难度要增加不少。在对基于WEB的网上订餐系统进行测试时，需要从用户的角度对各类需求进行处理，保证订餐系统的安全性和可用性。测试必须全面覆盖整个系统的方方面面，尽量不出现遗漏。

## 5.1功能测试

系统的功能测试就是验证系统的各项功能是否能够实现，只考虑功能的实现问题，不考虑整个软件的结构和代码。功能测试包括了表单测试、链接测试及数据库测试。

（1）链接测试：进行链接测试时，点击页面中的所有超链接，查看每个超链接的链接是否正确，是否出现无效链接。

（2）表单测试：在网上订餐系统中，顾客提交的外卖订单、商家后台的库存数量、配送员端的一些操作等等都是通过提交表单来实现的。例如：在顾客登录时，对比顾客登录时输入的信息和注册时记录在系统内的信息是否吻合。

（3）数据库测试：在本订餐系统中，专门针对数据存储是否一致进行了测试，反复测试了信息显示的输入错误情况。

## 5.2 界面测试

界面测试的目的是检验网站界面的风格是否符合预计的目标，网上订餐系统的主要适用人群是大学生和上班族，因此风格应当是活泼轻松的。系统的界面测试包括：整体界面测试、内容测试以及图形测试。

（1）整体界面测试：分别模拟顾客、商家、配送员的身份，从头到尾将系统的整个流程过一遍，以测试用户体验是否良好。

（2）内容测试：按照从小到大，从局部到整体的流程来测试模块的各个功能。首先测试一个模块的功能，再逐步增加模块，联合起来进行检测。 根据每个模块的检测结果来判断整个系统的的功能是否完整、复合目标。

（3）图形测试：检查网页中字体、颜色是否和谐，确保所有的图形用途都是明确的，检查图形色彩的搭配是否协调。

## 5.3性能测试

性能测试的目的是为了在运行程序时，能够保证系统对用户的各类请求都能做出及时的响应。确保程序在运行过程中不会发生中断、溢出等问题，并且确保系统的运行速度能够满足顾客的要求。

在对网上订餐系统进行性能测试时，主要测试其链接、负载、安全性能等等。测试多个用户同时登陆是否会导致系统运行缓慢以及图片加载的速度快慢。

# 第六章 总结

基于WEB的网上订餐系统实现了顾客点餐、选择餐品、提交订单、支付订单，商家接收订单、通知配送，配送员确认收货等功能。在论文的第一章，阐述了网上订餐系统的研究背景与研究意义，国内外发展现状，研究订餐系统的目标与意义，介绍了开发本网上订餐系统用到的研究方法和技术路线。在论文的第二章里，介绍了网上订餐系统开发所用到的开发工具，网页前端和后台用到的框架以及数据库的选择。在第三章中，介绍了网上订餐系统的需求分析，并且绘制了订餐系统的功能框图、用例图、实体-属性图、实体-关系图、类图、活动图，详细而清晰地介绍了本网上订餐系统的“顾客”、“商家”、“配送员”、“系统管理员”、“商家后台”这五个模块的功能，并确定了网上订餐系统的整体功能。此外，通过对功能需求的逐步求精、不断细化梳理出了业务逻辑，完成了数据库的设计。在论文的第四章中，介绍了网站的主要模块的界面，以及开发网上订餐系统的部分代码，简述了系统开发环境的配置和各功能模块的具体实现。在第五章中，介绍了网上订餐系统完成后，对系统进行的各种测试，简述了测试的主要过程以及一些要点。

# 参考文献

［1］学位论文 网上订餐系统的设计与实现 [J]．- 2015

[2]期刊论文 基于JSP技术的网上自助点餐系统的设计与实现 [J]．电脑知识与技术 - 2015, 11(33)

[3]赛奎春.《JSP工程应用与项目实践》.［M］.机械工业出版社,2002年

[4]期刊论文 在线订餐管理系统的设计与开发 电脑知识与技术 [J]．- 2015(12)

[5]张良银，浅论C/S和 B/S 体系结构[J]．工程地质计算机应用，2006，4．

[6]期刊论文 基于JSP的网上订餐系统研究 才智 [J]．- 2014(36)

[7]罗布森.《Head First HTML与CSS》（第2版）[M].中国电力出版社2013,9

[8]弗里曼.《Head First HTML5 Programming》[M].中国电力出版社2012,9

[9]巴萨姆.《Head First Servlets and JSP》（第2版）[M].中国电力出版社2012,9

[10]Brown等.《JSP编程指南（第二版）》.［M］.电子工业出版社 ,2003年

[11]清宏计算机工作室.《JSP编程技巧》.［M］.机械工业出版社,2004年

[12]朱红,司光亚.《JSP Web编程指南》.［M］.电子工业出版社,2001年

[13]任泰明.《基于B/S结构的软件开发技术》[M].西安电子科技大学出版社，2006，11

[14]李昭原,刘又诚.《数据库系统原理与技术》[M].航空航天大学出版社,2002

[16]王珊,萨师煊.《数据库系统概论》[M].高等教育出版社,2006.

[17]学位论文 网上订餐系统的设计与实现 [J]．- 2014

[18]孙卫琴.《精通struts》［M］.电子工业出版社，2004年

[19]周启海.网站全程设计技术.北京交通大学出版社,2003.6.

[20]周晓娟,王法能,毛劲松.基于B/S的WEB数据库技术[J].航空计算技术,2005.4.

# 致谢

转眼大四即将结束，这篇毕业设计论文也将为我的大学生涯画上一个句点。回顾四年的时光，与老师、同学之间的点点滴滴仿佛还是刚刚发生的事情，但却俨然成为了过去。我很幸运，在这四年中，承蒙了学院老师和学长学姐们的诸多照拂，在此，我想由衷地对您们道一声感谢！

特别是我的毕设指导老师王文永教授，感谢您每周都从百忙之中抽出宝贵的时间，指导我们毕设小组里每个人的项目和论文的撰写，以及感谢您在交流时对我们亲切地鼓励关怀，给我们传授实用的人生经验。王老师不但具有非常深厚的专业素养，而且十分耐心负责，善于因材施教。毕业设计是我人生中完成的第一个完整的项目，正是在王教授的悉心指点下，我才能够在毕业前顺利完成我的毕业论文。真的很感谢您对我的帮助！我将在今后的人生中，努力学习，勤奋用功，绝不辜负您对我的帮助和期望！

最后，我还要感谢的是在我论文撰写和项目开发期间帮助我，支持我的可爱的同学和朋友，你们的帮助对我而言真的很重要，谢谢你们！