

工程硕士  
学位论文

# 移动订餐 APP 系统的设计与实现

覃志宏

廣西大學

二〇二一年三月

分类号 TP3111.56

密级 公开

UDC \_\_\_\_\_

## 工程硕士学位论文

# 移动订餐 APP 系统的设计与实现

覃志宏

专业学位 计算机技术

指导教师 陈宁江教授

张文博高级工程师

论文答辩日期 2021 年 3 月 6 日 学位授予日期 2021.3.30

答辩委员会主席 陈友初

## 广西大学学位论文原创性和使用授权声明

本人声明所呈交的论文，是本人在导师的指导下独立进行研究所取得的研究成果。除已特别加以标注和致谢的地方外，论文不包含任何其他个人或集体已经发表或撰写的研究成果，也不包含本人或他人为获得广西大学或其它单位的学位而使用过的材料。与我一同工作的同事对本论文的研究工作所做的贡献均已在论文中作了明确说明。

本人在导师指导下所完成的学位论文及相关的职务作品，知识产权归属广西大学。本人授权广西大学拥有学位论文的部分使用权，即：学校有权保存并向国家有关部门或机构送交学位论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或其它复制手段保存、汇编学位论文。

本学位论文属于：

保密，在        年解密后适用授权。

不保密。

(请在以上相应方框内打“√”)

论文作者签名：      覃志宏      日期： 2021-03-16

指导教师签名：      陈亮江      日期： 2021.3.16

作者联系电话：                          电子邮箱：

# 移动订餐 APP 系统的设计与实现

## 摘要

如何减少浪费、降低运营成本和提高工作效率及服务水平，提高员工用餐体验和减少订餐、用餐耗时，是企事业单位食堂共同面临的问题和需求。研究开发一种集订餐和交互功能于一体的手机订餐交易管理系统，将其应用于企事业单位的食堂订餐业务中，不仅可为单位员工提供一种灵活便捷的订餐和交流餐饮知识的方式，有效减少员工在订餐上所耗费的时间，还可节省食堂经营成本和工作效率，打造便捷的食堂订餐及就餐模式。

本论文阐述了基于 Android 移动平台的移动订餐 APP 系统的分析和设计过程。该系统基于 B/S 结构和 Eclipse 平台，主要采用 Java 进行开发设计，在 SSM 框架基础上设计了基于 MVC 架构的实现方案。手机客户端应用 Android UI 开发技术遵照 MVC 模式进行开发，运用 HTTP 协议来实现手机客户端与服务器端的通信。选用 MySQL 作为后台数据库，采用 Tomcat 作为 Web 服务器。本论文用系统业务流程图、系统功能模块图等分析了系统功能的运行流程和组织结构，并对系统的业务过程和分析过程进行 UML 建模，通过用例图、流程图等展示了系统的各个功能模块，给出了各主要功能的运用界面。最后对本系统的功能设计和实现进行了测试和总结。

本移动订餐 APP 系统的主要功能由手机用户端和服务端两部分组成。手机用户端包括注册和登录、关注、收藏、订餐、发布菜谱、菜谱评论、用户留言等功能。服务端包括用户管理、菜谱管理、分类管理、评论管理、公告管理等功能，管理员通过后台对系统进行管理和维护。

经测试，本移动订餐 APP 系统已基本实现预期的功能，系统运行效果稳定。

**关键词：**订餐 APP    Android    交互功能    SSM 框架    MVC 架构

# **DESIGN AND IMPLEMENTATION OF MOBILE ORDERING APP SYSTEM**

## **ABSTRACT**

How to reduce waste, reduce operating costs and improve work efficiency and service level, improve staff dining experience and reduce ordering, dining time-consuming, is the common problem and demand of enterprise canteens. The research and development of a set of ordering and interactive functions in one of the mobile phone ordering transaction management system, which is applied to the canteen ordering business of enterprises and institutions, can not only provide a flexible and convenient way for the unit staff to order and exchange catering knowledge, effectively reduce the staff's time in ordering, but also save the canteen operating costs and work efficiency, create a convenient environment Canteen ordering and dining mode.

This thesis describes the analysis and design process of mobile ordering app system based on Android mobile platform. The system is based on B / S structure and Eclipse platform, and mainly uses Java to develop and design. On the basis of SSM framework, an implementation scheme based on MVC architecture is designed. Mobile client application Android UI development technology is developed according to MVC mode, using HTTP protocol to realize the communication between mobile client and server. MySQL is selected as the background database and Tomcat as the web server. This paper analyzes

the operation process and organization structure of the system function with system business flow chart and system function module diagram, and carries out UML modeling for the business process and analysis process of the system, shows each function module of the system through use case diagram and flow chart, and gives the application interface of each main function. Finally, the function design and implementation of the system are tested and summarized.

The main function of the mobile ordering app system is composed of mobile phone client and server. Mobile terminal includes registration and login, attention, collection, ordering, publishing recipes, menu comments, user messages and other functions. The server includes user management, menu management, classification management, comment management, announcement management and other functions. The administrator manages and maintains the system through the background.

After testing, the mobile ordering app system has basically realized the expected function, and the system operation effect is stable.

**KEY WORDS:** ordering app; Android; interactive function; SSM framework; MVC architecture.

# 目 录

摘要.....	I
<b>ABSTRACT .....</b>	<b>III</b>
第1章 绪论.....	1
1.1 研究背景.....	1
1.2 研究现状.....	1
1.3 研究目的和意义.....	3
1.4 相关技术概述.....	4
1.5 论文的主要研究内容.....	9
1.6 论文组织结构.....	10
第2章 系统需求分析.....	12
2.1 业务过程分析.....	12
2.2 需求分析.....	13
2.2.1 功能需求.....	13
2.2.2 性能需求.....	15
2.3 用例分析和建模.....	15
2.5 业务流程分析.....	23
2.6 本章小结.....	25
第3章 系统设计.....	27
3.1 系统架构设计.....	27
3.2 主要功能模块设计.....	30
3.3 数据库设计.....	30
3.4 客户端详细设计.....	36
3.4.1 开发环境设计.....	36
3.4.2 用户端功能模块设计 .....	37
3.5 服务端详细设计.....	41
3.5.1 用户管理.....	41
3.5.2 菜谱管理.....	42

3.5.3 评论管理.....	43
3.5.4 公告管理.....	44
3.6 本章小结.....	45
第 4 章 系统实现.....	46
4.1 用户功能模块的实现.....	46
4.1.1 用户注册登录模块.....	46
4.1.2 用户订餐模块.....	47
4.1.3 首页模块.....	49
4.1.4 发布食谱模块.....	50
4.1.5 浏览食谱模块.....	52
4.1.6 查看用户信息和关注模块.....	53
4.1.7 用户聊天界面.....	55
4.1.8 查看公告界面.....	57
4.2 管理员功能模块的实现.....	57
4.2.1 用户管理界面.....	57
4.2.2 菜谱管理界面.....	61
4.2.3 公告管理界面.....	62
4.2.4 类别管理界面.....	62
4.3 手机 app 的实现和发布 .....	63
4.4 本章小结.....	64
第 5 章 系统测试.....	65
5.1 测试方法.....	65
5.2 系统主要功能测试.....	66
5.2.1 登录与登出 .....	66
5.2.2 食谱查询功能测试.....	67
5.2.3 订餐功能测试.....	67
5.2.4 菜谱分类功能测试.....	67
5.2.5 管理权限功能测试.....	68
5.2.6 修改密码功能测试.....	68
5.3 系统性能测试.....	69

5.4 本章小结.....	71
第 6 章 结束语.....	72
6.1 工作总结.....	72
6.2 后续工作展望.....	72
参考文献.....	74
致 谢.....	77
附 录.....	78

## 第 1 章 绪论

### 1.1 研究背景

人们的生活节奏随着国民经济的发展而加快, 忙碌的生活使人们花在制作餐食上的时间越来越少, 学生党抑或是上班族, 都更倾向于点外卖来解决三餐问题。互联网的不断发展给生活带来了极大的便利, 现在人们的生活已经离不开互联网, 所以餐饮行业也要跟上时代的步伐, 将餐饮与互联网相结合才能更好的实现其经营发展<sup>[1]</sup>。从有关数据来看, 餐饮行业现已成为深受移动互联网影响的行业之一。

如今智能手机已经在社会各类人群中得到广泛普及, 智能手机分为 Android 和 IOS 两大阵营, Android 的源码完全开放, 使得开发变得非常灵活, 可操作性和可塑性强, IOS 则较为封闭, 只有 Apple 公司在开发。在国内智能手机领域中, 现在已经摆脱了国外大牌占据半壁江山的状况, 诸多高性价比的国产 Android 系统智能手机让越来越多的人使用到了更好的智能手机, 现在中国成年人基本上已是人手一部智能手机。随着科技的发展, 智能手机的功能也越来越完善, 越来越强大, 用电脑可以完成的日常事务, 现在用智能手机基本上都可以完成, 而且比电脑更易于操作、更便捷、更灵活, 一部智能手机就能将诸多问题搞定。智能手机的这些特性, 使其理所当然的成为了人们日常生活中消费、网购的主要工具。

当今社会是一个充斥着数字技术的世界, 数字技术几乎覆盖了我们生活的每个细节<sup>[2]</sup>, 网络和通信技术高速发展, 移动互联网越来越发达。这也为智能手机的全面性提供了基础条件。移动技术是当今商业、生活的需求, 移动系统现已成为很流行的创新技术系统。俗话说“民以食为天”, 餐饮作为高频的需求, 与人们的生活息息相关。随着经济的发展, 纯物质需求已不再是人们关注的第一要点, 开发一种集订餐功能与用户交互功能于一体的订餐交易类手机 APP, 既能为消费者提供便捷的订餐方式, 通过用户间餐饮知识的交流、学习, 又能够丰富用户的科学餐饮知识, 提高食品安全意识, 让消费者们享受到便捷购物、注重餐饮安全的乐趣<sup>[3]</sup>, 此外还有利于找到志趣相投的伙伴。

### 1.2 研究现状

#### (1) 国内外研究现状

随着电子商务的快速发展, 网络销售的占比亦逐年持续增大。网购在社会消费中占有不小的比重, 餐饮类消费在吃、穿、用类实物商品网上零售额中所占的比重最大, 现阶段我国

餐饮业无论是在规模、经营理念还是管理水平上都有了很大的进步<sup>[3]</sup>。在国内外，随着人们生活水平的提高和电子商务的迅猛发展，人们对自己的饮食、健康日益注重，在紧张工作之余很多人会选择享受美食进行放松<sup>[4]</sup>，以往的效率低下的电话订餐等方式已逐渐被淘汰，人们更加倾向于简便快捷且效率高的网上订餐。网上订餐这个行业，国内发展水平远远超出国外，国内庞大的用户群体促使相关人员对网上订餐这一活动进行研究，随着用户数量的增加，在产生的庞大数据中所隐含的信息，势必会影响整个行业的发展，洞悉这些数据背后的意义以及信息，可以有效地促进外卖市场的发展<sup>[5]</sup>。

在国外，英美等发达国家的大型外卖平台多数采用的是外卖旧模式，即只做平台而不介入配送，但近些年来，统一配送的外卖新模式已在许多国家发展得轰轰烈烈，在以中国、印度等为代表的新兴国家，外卖新模式更为普及。Just Eat 是英国最大的外卖平台，DoorDash、Uber Eats 和 GrubHub 则是美国的三大外卖平台。其中 GrubHub 是美国的老牌外卖网站，创立于 2004 年，是美国电子商务界杰出的代表。GrubHub 每天的外卖用户数量及其产生的数据非常大，GrubHub 的系统开发者在设计时采用了 B/S 结构，只需把所有的功能都实现在 Web 服务器上，并就不同的功能为各个组别的用户设置权限即可。GrubHub 的数据库软件技术很强大，每天系统会有大量用户使用，并且随着用户的活动，会产生大量的数据，正是其强大的数据库，保证了 GrubHub 的正常运营。DoorDash 则是由 Stanley Tang、Andy Fang 及 Tony Xu 三位华裔所创立，DoorDash 是美国目前发展最快、最潮的外卖平台，2021 年的市场调查报告显示，DoorDash 现已在美国餐饮外卖市场上跻身第一位，超过了老牌平台 Uber Eats 和 GrubHub。DoorDash 的操作简单、明了，能让用户快速找到喜欢的外卖。

在国内，外卖可以说是渗入了千家万户，目前较受消费者青睐的是美团外卖和饿了么外卖，外卖行业形成了两强争霸的格局。美团外卖是一个由腾讯控股支持的独立于美团网的网上订餐服务平台，于 2013 年 11 月正式上线，其核心功能是外卖，美团外卖提供美食、水果生鲜、甜点饮品、超市便利等众多品类的配送功能。相较于其他外卖软件，美团外卖的软件风格简洁明了，核心功能明确，用户下单流程流畅快捷，并且有强大的智能提示搜索功能。作为刮起“互联网+外卖”风潮的领头羊，抢占市场时间早是饿了么外卖的最大优势<sup>[6]</sup>，饿了么外卖拥有自建的短途物流团队，与第三方公司进行合作，注重配送速度和食品保鲜程度。

## （2）国内订餐交易软件发展现状

当前，国内订餐交易软件的发展前景喜人，研发、更新换代的速度很快。依靠移动技术和支付手段的进步，国内的手机订餐业务迅速兴起，成为电子商务的重要业务领域<sup>[7]</sup>。受大众青睐的订餐 APP 迎合其核心消费群体的时尚需求，实现了手机移动订餐、订位、选餐、

约餐、配送等诸多服务功能。虽然现在市场上的订餐 APP 产品已经得到了大多数用户的肯定，但是随着人们需求的不断增长与细分，APP 也需要不断更新发展。最近几年由于市场和客户的需求，多种多样的 APP 产品纷纷出现，比如购物杰、下厨房等移动产品深得用户的青睐，这些订餐 APP 产品主要功能是向用户推荐菜谱<sup>[8]</sup>。

国内目前的订餐交易 APP 产品发展具备以下特点：

①APP 产品起初是作为分享购物和交流订餐的工具，但是为了适应用户日益增加的需求，各大产品 APP 已经从最初单一的菜谱推荐，转变为多功能社区，多平台、多功能的综合性软件。

②APP 系统可以将用户与用户之间通过购物连接在一起，不仅可以促使用户之间切磋订餐经验，还可以促使一些志同道合的用户结交为朋友。电商利用已有的人气，继续推广其 APP，承接广告，最终达到普及和赚钱的目的。

③支付方式使用手机支付宝、微信支付和网银等多种支付渠道；

从订餐交易 APP 的发展途径中分析得到，其庞大的用户量具有很好的条件来适应这种新型的购物 APP，订餐交易不仅成为了用户获取购物、交流购物、分享购物、切磋订餐的途径，更成为了他们生活中必不可少的一种生活方式和用餐理念，其提供的先天性优势充分的得到了体现<sup>[9]</sup>。最为重要的是，订餐交易 APP 这一套吸引并留住用户的功能和手段，最终都会和用户的“网购”紧密的联系在一起，促使用户们频繁的消费，以满足自己的需求，从而达到了所有企业以盈利为目的的经营理念。由于目前 APP 的发展并没有形成一家独大的场面，并且由于客户的需求量极大，所以发展空间依然非常可观<sup>[10]</sup>。在开发基于手机的订餐交易系统时，如果能将购物功能和社交功能结合起来，在发展上会有很大空间。

### 1.3 研究目的和意义

通过对课题背景的了解以及国内外现状的研究，开发一个集订餐购物与用户交互功能于一体的手机订餐交易管理系统的意义如下：

(1) 订餐 APP 是基于手机的应用，其操作简单易学，不需要太多的业务操作能力即可轻松实现订餐操作和交流学习，而且订餐 APP 小巧轻便易安装，其占用的手机内存少，不易导致手机卡顿，因此对于用户来说使用成本低且操作方便；

(2) 手机支付堪称继卡类支付、网络支付后消费者的新宠，手机订餐已逐渐成为餐饮消费的主要方式。手机订餐交易管理系统除了能满足用户随时随地订餐的需要，既便

捷又省时，还能让使用此订餐 APP 的企事业单位食堂提高工作效率、减少经营成本，提升服务水平和用户订餐体验；

（3）将交互功能与订餐购物功能相结合，为手机订餐用户提供一个学习、交流餐饮知识，结识志趣相投伙伴的平台，促进大家重视科学饮食，践行更健康的生活方式<sup>[11]</sup>。

## 1.4 相关技术概述

本移动订餐 APP 系统是基于 Android 平台开发的，分为客户端和服务端两部分。手机客户端主要应用 Android UI 开发技术遵照 MVC 模式进行开发。服务端使用 Java 和 JSP（java server page）技术来达到效果，通过 SSM（Spring、SpringMVC 和 Mybatis）框架的相关技术来实现 MVC 架构（Model View Controller）。本系统主要使用 Java 语言进行开发，运用 JS、HTML5 优化页面，数据库采用的是 MySQL 数据库，用 Tomcat 作为 Web 服务器，服务器端界面设计则是运用 jQuery EasyUI（是一个基于 jQuery 的前端框架，集成了各种用户界面插件），另外还采用了 JDBC 技术，利用了 MyEclipse sdk、ADT、AVD 、DDMS 等开发环境及开发工具。

在服务器端，管理员界面的设计所采用的是 jQuery EasyUI 接口框架。Query EasyUI 框架是一个基于 jQuery 的用户界面的插件集，为构建现代化、交互式的 JavaScript 应用程序提供必要的功能、条件。在使用这套接口框架时，程序员只需要编写一些 HTML 标签来实现用户界面的定义，大大提高了开发效率。

JAVA，中文名称为超文本预处理语言。是一种比较流行的编程语言，多用在移动的应用开发及企业应用技术开发<sup>[12]</sup>。因为其强大的通用功能而被广泛的应用，它的特点在 web 的开发中得到完美的融合，并在 web 开发中被定义为 HTML 的内嵌式语言，JAVA 对操作系统和 WEB 服务器的要求很低，所以操作者可以根据自己的喜好自由的选择服务器抑或是操作系统。除此之外，还可以根据需求选择单个编程类型，或者编程方式。

MyEclipse 是对 EclipseIDE 的扩展，利用它可以在数据库和 Java EE 的开发、发布以及应用程序服务器的整合方面极大的提高工作效率。它是功能丰富的 Java EE 集成开发环境，包括了完备的编码、调试、测试和发布功能，完整支持 HTML, Struts, JSP, CSS, Javascript, Spring, SQL, Hibernate<sup>[13]</sup>。

Tomcat 是一个免费的开源的 Serlvet 容器，其也提供了一个应用 manager，访问这个应用需要用户名和密码，用户名和密码存储在一个 xml 文件中<sup>[14]</sup>。Tomcat 作为一款轻量级

的服务器,在很多中小型的开发中都会使用,其具有免费、开源等优点,同时也具有良好的扩展性,另外 Tomcat 还具有易于部署、移植和升级三大特性。

MySQL 作为一个开源数据库,性能稳定,国内拥有众多的企业级用户,如淘宝,新浪,腾讯等,采用它既能满足企业需求,又能避免因企业版权问题而产生纠纷。MySQL 是微软开发的较为完善的数据库系统,而且 MySQL 体积小、查询速度快,本订餐管理系统基于对设计的简洁性和简单易上手的需求而选择它。

JSP (Java Server Page, 中文名为 Java 服务器页面) 是一种动态网页技术标准,其主要目的是将表示逻辑从 Servlet 中分离出来,是一种使用 Java 来实现前端效果的技术。JSP 可以用来封装产生动态网页的处理逻辑,用 JSP 编写的网页是能够访问服务端存放资源的应用逻辑的网页。JSP 技术把网页设计的逻辑和页面显示实现分离,界面作为 MVC 框架里的 V 层,而后台的处理逻辑则是 MVC 框架内的 C 层。JSP 是支持可重用的基于组件的设计,这一特性使得用 JSP 开发 web 应用程序更加便捷。

SSM 框架是 spring、spring MVC 和 mybatis 框架的整合,是标准的 MVC 模式,它将整个系统划分为四层:表现层, controller 层, service 层, DAO 层。在 SSM 框架中, spring 实现业务对象管理,用 spring MVC 负责请求的转发和视图管理, mybatis 作为数据对象的持久化引擎<sup>[15]</sup>。

### (1) Spring

Spring 是一个 Java 开发框架,是一个轻量级的控制反转 IoC(或依赖注入 DI)和面向切面 AOP 的容器框架,致力于简化 Java 开发。选择使用 Spring 能够很好的满足项目后期的维护与扩展。Spring 是为了解决企业应用开发的复杂性而创建的,使用 Spring 开发可以将 Bean 对象、Dao 组件对象、Service 组件对象等交给 Spring 容器来管理。为了提高开发效率,简化开发和减少代码量, Spring 引用 JavaBean 机制来替代 EJB 的工作, Spring 不仅可用于服务端的开发,其简单性、高内聚、低耦合的特性还可满足绝大多数应用场景。Spring 有二大特性:一个是 IOC(即控制反转),另一个是 AOP(面向切面)。IOC 的使用可以将大部分的 JavaBean 交给容器进行创建和管理,从而减少开发人员的代码量,程序员只需使用注解或者配置即能完成 JavaBean 的创建管理;在使用 AOP 前,如果需要对现有的代码进行增强,必须将代码写到相应的实现逻辑处,或者使用静态动态代理等方法来实现增强,但引入 AOP 后,我们只需要对需要增强的代码使用 AOP 注解,注解指定相应的业务代码类方法,使用非常简单<sup>[16]</sup>。

Spring 框架是一个分层架构,它包含一系列的功能要素。Spring 总共大约有 20 个模

块，由 1300 多个不同的文件构成，这些组件被分别整合在 Core Container（核心容器）、AOP（Aspect Oriented Programming）、Instrumentation（设备支持）、Data Access/Integration（数据访问与集成）、Web、Messaging（通信报文）、Test 等 6 个模块中。Spring 的核心容器是其他模块建立的基础，其它的模块都是基于此核心模块进行开发，以组件的方式集成进来的。Spring 整体架构如图 1-1 所示。

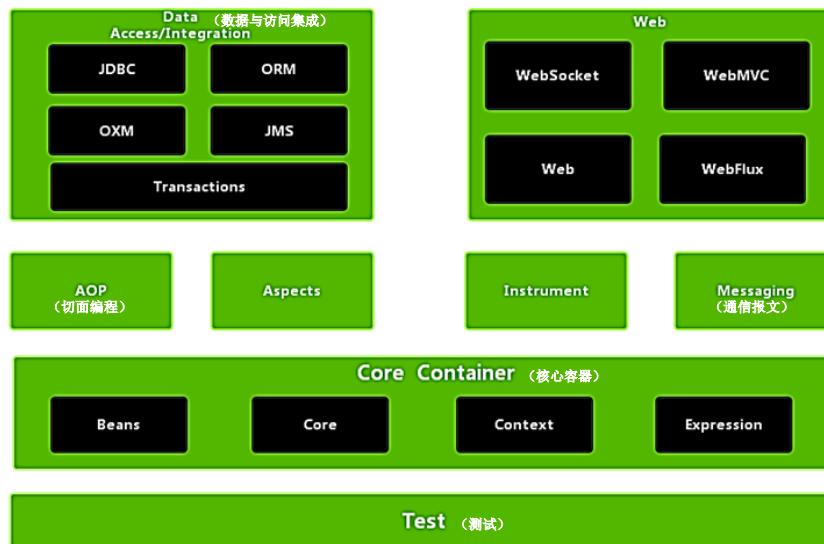


图 1-1 Spring 体系结构图

Figure 1-1 Spring architecture diagram

图 1-1 中包含了 Spring 框架的主要模块，这些模块可以满足企业级应用开发的需求，在开发过程中可以根据需求有选择性地使用所需要的模块。组成 Spring 框架的每个模块（组件）既可以独自存在，也可以与其他模块联合实现。

### 1) Core Container（核心容器）

Spring 的核心容器是其他模块建立的基础，它提供 Spring 框架的基本功能，由 core、beans、context 和 expression（Spring 表达式语言）等模块组成。

**Core 核心模块：**提供了 Spring 框架的基本组成部分，包括控制反转（Inversion of Control, IOC）和依赖注入（Dependency Injection, DI）功能，是 IOC 与 DI 的最基本实现。

**Beans 模块：**提供了 BeanFactory，是工厂模式的经典实现，Spring 将管理对象称为 Bean，是 Bean 工厂与 Bean 的装配。

**context 上下文模块：**定义基础的 Spring 的 Context 上下文即 IOC 容器，提供一个框架式的对象访问方式，是访问定义和配置的任何对象的媒介，它建立在 Core 和 Beans 模块的基础之上。

Expression 模块：是支持运行时查询和操作对象图的强大的表达式语言，它是对 JSP 2.1 规范中规定的统一表达式语言（Unified EL）的扩展。

### 2) Data Access/Integration（数据访问与集成）

包括 JDBC、ORM、OXM、JMS 和 Transactions 等 5 个模块模块。

JDBC 模块：是一个 JDBC 抽象框架的主要实现模块，用于简化 Spring JDBC 操作，大幅度减少了在开发过程中对数据库操作的编码。

TX 模块：Spring JDBC 事务控制实现模块。

ORM 模块：对流行的对象关系映射 API，包括 JPA、JDO、Hibernate 和 iBatis 提供了的集成层。

OXM 模块：提供了一个支持对象 XML 映射的抽象层实现，如 JAXB、Castor、XMLBeans、JiBX 和 XStream，将 Java 对象映射成 XML 数据，或者将 XML 数据映射成 Java 对象。

JMS（Java Messaging Service）模块：提供 Java 消息服务，能够发送和接收信息。。

Transactions 事务模块：支持编程和声明式事务管理实现特殊接口类，并为所有的 POJO。

### 3) Web 层

Spring 的 Web 层包括由 Web、WebMVC、WebSocket、WebFlux 等组件。

Web 模块：提供了最基础 Web 支持，提供基本的 Web 开发集成特性，主要建立于核心容器之上，通过 Servlet 或者 Listeners 来初始化 IOC 容器。

WebMVC 模块：实现了 Spring MVC(model-view-Controller) 的 Web 应用。提供了最基础 Web 支持，主要建立于核心容器之上，通过 Servlet 或者 Listeners 来初始化 IOC 容器。

WebSocket 模块：主要是与 Web 前端的全双工通讯的协议。

WebFlux 模块：一个新的非堵塞函数式 Reactive Web 框架，可以用来建立异步的，非阻塞，事件驱动的服务。

### 4) AOP(切面编程) and Instrumentation

由 aop、aspects 和 instrument 等 3 个模块组成。

AOP 模块：面向切面编程的应用模块，整合 Asm、CGLIB、JDKProxy，是 Spring 的另一个核心模块，提供了一个符合 AOP 要求的面向切面的编程实现。作为继 OOP（面向对象编程）后，对程序员影响最大的编程思想之一，AOP 极大地开拓了人们对于编程的思路。

aspects 模块：提供了与 AspectJ 的集成功能，AspectJ 是一个功能强大的 AOP 框架。

instrument 模块：是 AOP 的一个支援模块，提供了类检测支持和类加载器的实现，可以在特定的应用服务器中使用。主要作用是在 JVM 启用时，生成一个代理类，程序员通过代理类在运行时修改类的字节，从而改变一个类的功能，实现 AOP 的功能。

### 5) Messaging (通信报文)

只包含 messaging 模块。该模块是从 Spring4 开始新加入的一个模块，主要职责是为 Spring 框架集成一些基础的报文传送应用。

### 6) Test (测试)

由 test 模块组成。主要功能为测试提供支持。

#### (2) Spring MVC

Spring MVC 是 Spring 基本体系结构的一个组成部分，相当前于 Spring 框架的扩展和延续，它现已成为 Web 应用开发的首选框架。Spring MVC 是类似于 Struts2（基于 MVC 的轻量级的 web 应用框架）的 MVC 框架，它的诞生是为了 Spring Web 的开发，它用于 web 层，并且已集成到 Spring Web Flow 中，因此我们几乎不需要任何其他配置即可与以后的 Spring 集成。Spring MVC 相当于 controller（控制器），可用来处理用户请求。以前的 web 是业务、逻辑与页面三者混写的，而 Spring MVC 分离了控制器、模型对象、分派器以及处理程序对象的角色，引入 Spring MVC，可以使用三层架构来开发 web 项目、控制层、业务层、视图层，这种分层的思想更易于开发上的定制<sup>[17]</sup>。SpringMVC 工作原理如图 1-2 所示。



图 1-2 SpringMVC 工作原理图

Figure 1-2 Working principle of spring MVC

SpringMVC 的工作流程如下：

- 1) 客户端发送请求；
- 2) 请求交由核心控制器处理；
- 3) 核心控制器找到映射器，映射器查看请求路径是什么；
- 4) 核心控制器再找到适配器，查看有哪些类实现了 Controller 接口或者对应的 bean 对象；
- 5) 将带过来的数据进行转换、格式化等操作；
- 6) 找到控制器 Action，处理完业务之后返回一个 Model And View 对象；
- 7) 通过视图解析器来对 Model And View 进行解析；
- 8) 最后跳转到对应的 JSP/html 页面，把 View 返回给请求者，至此一个流程结束。

### (3) Mybatis

Mybatis 是 apache 的一个开源项目，它的前身是 ibatis，2010 年经 apache 开发组织对 ibatis 进行改进和升级，改名为 Mybatis，Mybatis 的主要业务功能是与数据库进行交互，实现对数据的增删改查，是一个基于 Java 的持久层框架，它支持自定义 SQL、存储过程以及高级映射。Mybatis 是对 jdbc 的封装，它让数据库底层操作变得透明，能有效提升数据库访问的效率。MyBatis 免除了几乎所有的 JDBC 代码以及手动设置参数和获取结果集的工作。Mybatis 非常的轻量，可以使用 xml 的方式来组织管理 sql，它的配置与使用也相当的简单，一般使用接口方式进行命名与传参定义，再结合 sql 语句进行数据库操作，开发效率和稳定性都很好。

## 1.5 论文的主要研究内容

本论文围绕“移动订餐 APP 系统”展开研究，在系统分析的阶段，应用面向对象开发方法和 UML 统一建模语言（Unified Modeling Language），对整个系统的功能需求和应用流程进行统一的分析。在此基础上，对系统进行了设计和实现。

具体完成的工作如下所述：

- (1) 研究当前的购物网站、APP 现状，找出目前存在的问题，提出自己的方案；
- (2) 对本系统开发过程中所涉及到的主要相关技术进行介绍，并对技术是如何运用到本系统中进行了说明分析；
- (3) 利用面向对象建模对系统的整个业务流程进行分析，采用用例图、流程图等对

业务详细过程进行说明分析；

- (4) 针对本系统的功能模块和需求分析确定其总体框架，并确定数据库实体，以及各个实体之间的联系，完成数据库的设计；
- (5) 对用户端功能中的注册与登录模块、菜谱展示、菜谱查询、菜谱上传、用户中心模块进行详细设计；
- (6) 对后台管理功能中的菜谱上传、用户信息、菜谱种类等管理模块进行详细设计；
- (7) 通过编码实现上述功能，并对系统进行测试。

## 1.6 论文组织结构

本论文的主要研究内容是移动订餐 APP 系统的设计与实现，论文分为以下六章，其框架结构如图 1-3 所示：

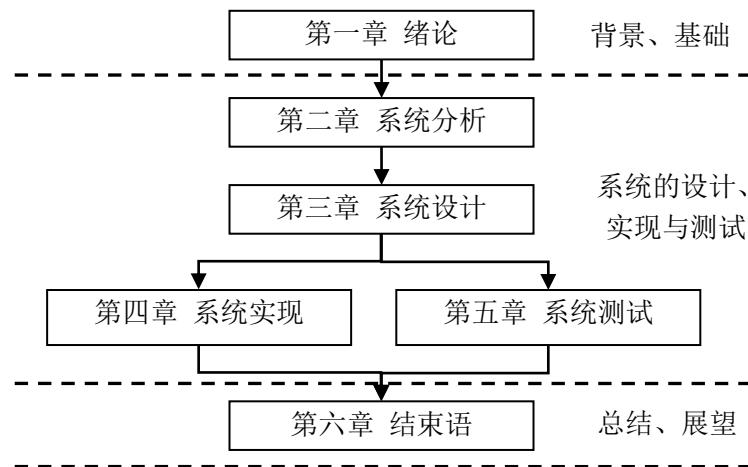


图 1-3 论文框架结构图  
Figure 1-3 Framework of the thesis

第一章 绪论，主要对选择此课题的原因、背景、意义、研究方向以及前景做一个简单的概括和描述，介绍了本论文的主要研究内容和组织结构。对本系统所使用的主要相关技术如 JAVA、MySQL、JSP 等进行了简要概述，介绍了本系统所采用的 SSM 框架所包含的相关技术。

第二章 系统需求分析，介绍了系统需求分析过程，包括业务过程的建模和需求分析的建模，对本系统的用例图和活动图进行了介绍。

第三章 系统设计，介绍了系统设计方案，包括总体框架设计和系统功能模块的设计，

以及数据库设计和用户端与服务端的功能模块的详细设计等。

第四章系统实现，介绍了系统的实现过程，包括界面设计的介绍和功能实现的过程。

第五章系统测试，说明了测试的软件、硬件环境与测试方法，编写客户端和服务器端功能模块的测试用例对系统进行测试，并显示测试结果，对系统进行了简要的总结评价。

第六章总结与展望，对本论文进行总结，总结了本课题研究完成的工作和成果，并且说明了在设计过程中该系统存在的不足以及后续的改进目标。

## 第 2 章 系统需求分析

本章主要对移动订餐 APP 系统的设计目标和业务需求进行了分析与介绍。首先是对移动订餐 APP 系统的业务流程进行了分析概述，然后从系统的用户模块、管理模块、支付功能等方面对系统的功能需求进行了详细分析，另外总结了系统的性能需求。

### 2.1 业务过程分析

经过分析移动订餐 APP 系统的业务流程，明确了系统的两大角色，分别是用户和管理员。其中将用户分成了两个种类，一种是未进行注册直接进入系统的用户，只能在线浏览菜谱信息和系统通告及订餐，但不能够发布自己的评价、不能留言、发布、收藏菜谱，也不能关注用户；另一种是已经注册的用户，用户可以浏览产品的详细信息和系统公告、订餐、发布和收藏菜谱、留言、对菜谱评价、关注用户，和用户互动等<sup>[18]</sup>。

管理员的主要功能是对系统后台进行管理和维护。可以对已经注册的用户信息、菜谱的分类、用户评论、公告信息进行增加、删除、修改、查询等管理工作。同时，管理员也可以对自己的信息进行管理，也可以增加管理员。此项目的系统业务流程如图 2-1 所示。

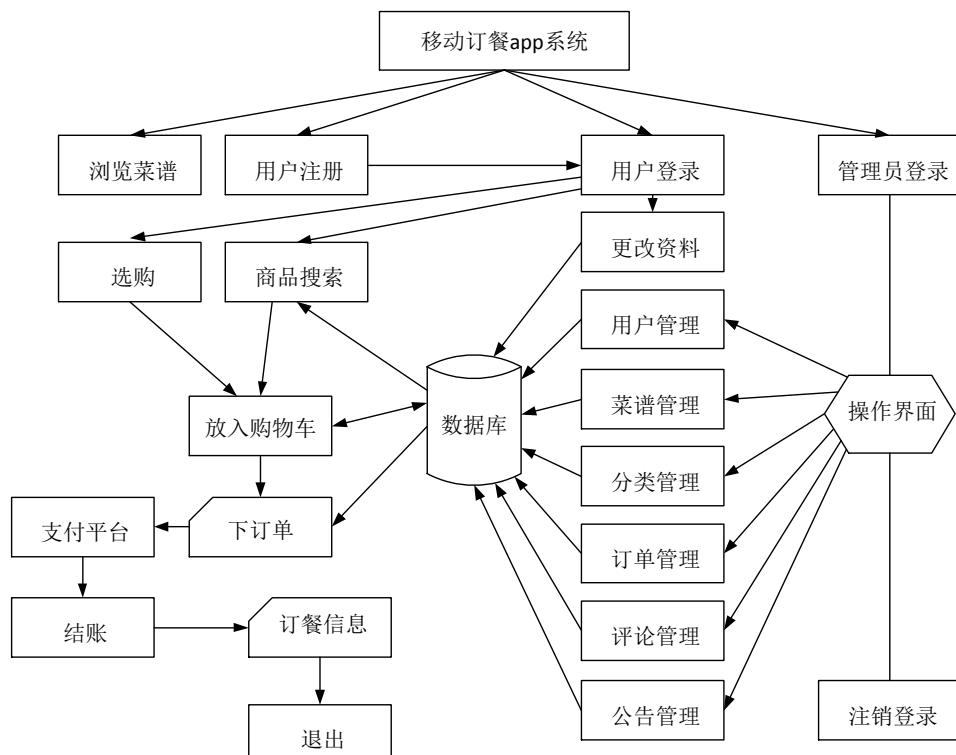


图 2-1 系统业务流程图  
Figure 2-1 System business flow chart

业务流程图是对业务功能分析的进一步细化，将系统的业务处理过程描述出来，它能更好的分析理解业务功能各要素之间的关系。业务流程图就是按照系统的信息流逐步进行的，主要包括信息来源、业务处理过程和方法、信息去向等<sup>[19]</sup>。

## 2.2 需求分析

需求分析是确定系统功能模块之前非常重要的一个环节，其不需要考虑到该系统如何实现，而是通过调查分析比较来明确系统的目标。需求分析的最终目的就是把用户的要求进行整合，形成表述完整的文档，确定任务<sup>[20]</sup>。

### 2.2.1 功能需求

本移动订餐 APP 系统主要包括手机用户端和后台管理模块两部分，下面将对这两大模块的功能做具体分析。

#### (1) 手机用户端功能需求

##### (1) 未进行注册的游客功能比较单一，主要有以下功能：

①注册需求：注册功能可以根据用户手机号进行注册，也可以使用微信与支付宝完成自动注册，注册完成之后需要绑定手机号才算真正完成注册功能；

②菜谱查询需求：可以通过分类查询，也可以通过输入关键字来查询；

③浏览公告：浏览公告相关的信息，并获得公告想传达的资讯；

##### (2) 注册成功的用户，主要有以下功能：

①登录需求：用户在成功登录系统后，能体验到区别于游客的其他功能；

②菜谱查询需求：这一模块用于菜谱列表的展示，方便用户对菜品有所了解。可以通过分类查询，也可通过输入的关键字来获得相应的菜谱。用户除浏览菜品做法外还可以评论，与其他用户一起切磋交流；

③个人信息管理需求：在这个模块，用户可对各种个人信息的修改和密码进行修改；

④菜谱上传需求：该模块实现用户菜谱上传功能，实现菜谱的共享；

⑤关注需求：这一模块实现用户之间的交流功能；

#### (2) 后台管理功能需求

##### (1) 用户管理模块需求：通过这个模块管理一切注册的用户的信息；

(2) 菜谱管理需求：对所有上传的菜谱进行管理，包括浏览、删除产品，添加、删除菜谱分类；

(3) 公告管理需求：通过发布系统公告，来保证用户能够第一时间的得到系统相关资讯，注意相关事宜。

### (3) 支付功能

对于在线交易的餐饮平台，支付功能是不可缺少的，因为平台提交的是一种交易行为，只有通过购买，支付才能算是一种商品行为，对买卖双方才算是一种交换与等值；在线支付要能够满足用户通过第三方平台来达到将款项正确结算到商家的功能，并且支持结算与退款功能，便于用户与商家进行查账。

目前市场上得到大众认可的主流的支付方式有支付宝和微信支付，下面简要描述一下支付宝<sup>[21]</sup>。

支付宝支付流程如下：

- 用户若已安装支付宝 App，用户在商户 APP 中选用支付宝支付方式后系统会跳转到支付宝中完成支付，支付完后跳返回到商户 APP 内展示支付结果。
- 用户若未安装支付宝 APP，在商户 APP 内选用支付宝支付方式会后调起支付宝网页支付界面，用户登录支付宝账户完成支付后展示支付结果。

应用程序对接支付宝，不仅能完成付款功能，还能对已支付的款进行退款，将支付宝与支付宝之间的钱进行互转，介于这些功能，可以在我们的应用程序上做一些推广类的应用，比如抢红包，当然前提是用户绑定了支付宝的帐号信息，这样可以直接在线提现到支付宝。支付宝的退款过程如图 2-2 所示。

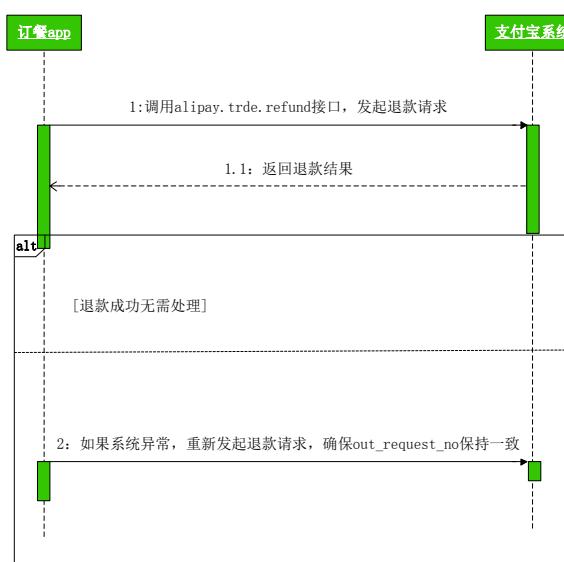


图 2-2 支付宝退款时序图  
Figure 2-2 Alipay payment sequence diagram Fig.

## 2.2.2 性能需求

在实现此基于手机的订餐交易管理系统开发过程中，首先要考虑下面几点，充分保证系统的性能<sup>[22]</sup>。

(1) 安全性。保证网络环境下的信息安全尤为重要，软件的设计应该保证使用者的个人信息不会被泄露。

(2) 稳定性。系统运行应该快速、稳定、高效和可靠。

(3) 扩展性。基于移动订餐 APP 系统的开发，用户后期可能还会有别的需求，所以不仅要考虑到系统现在需要满足用户的哪些需求，也要考虑到以后要根据用户需求的不断变化来完善系统，系统的设计与整体架构要有一定的灵活性，保证能够适应这些变化，为将来系统的后期维护和功能扩展提供便利。

## 2.3 用例分析和建模

在分析系统的过程中首先要明确系统的用例，使用用例图和用例规约表来对其进行更加明了的介绍。通过参与者来对系统功能模块进行分析，而用例规约则是详细介绍每个功能的实现步骤。

通过分析系统功能性需求，可以得到系统用例图如图 2-3 所示，游客（未注册用户）的用例图如图 2-4 所示，已注册用户和管理员的用例图分别如图 2-5 和图 2-6 所示。

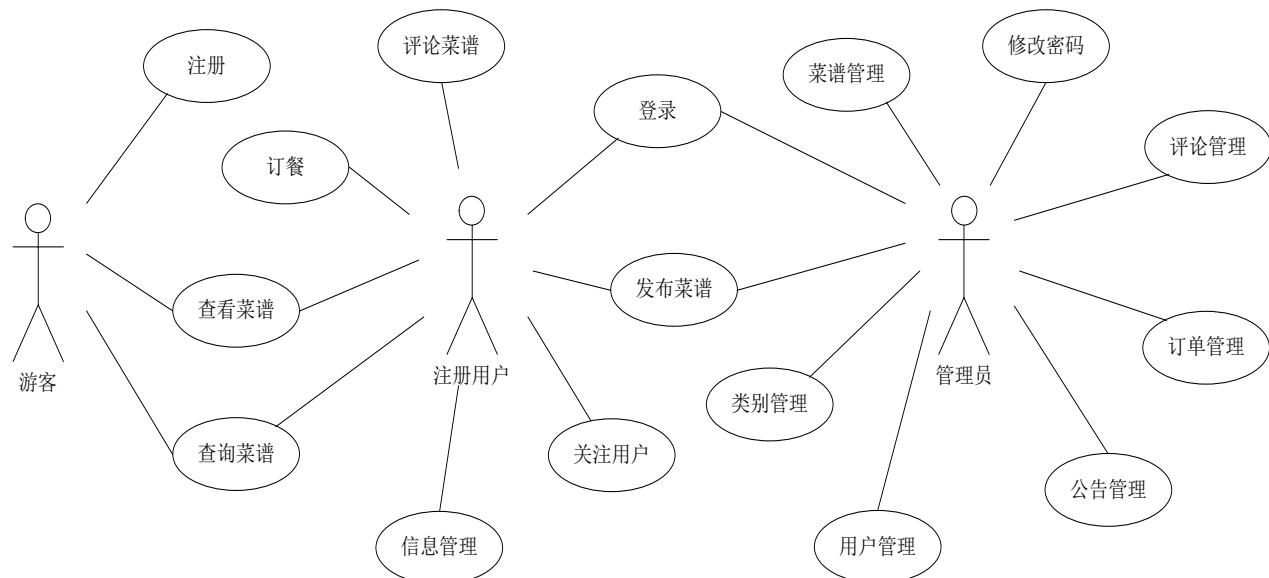


图 2-3 系统用例图

Figure 2-3 Registered user use case diagram

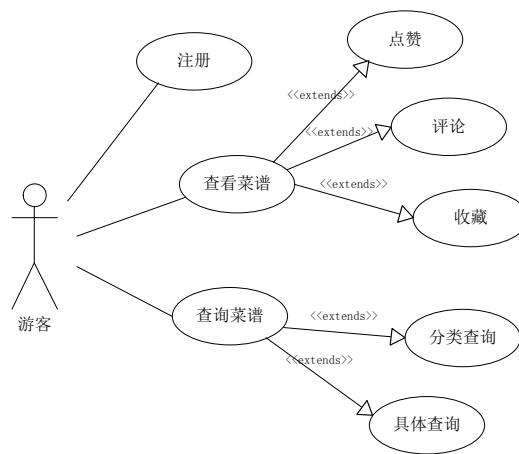


图 2-4 游客（未注册用户）用例图

Figure 2-4 Use case diagram of tourists (unregistered users)

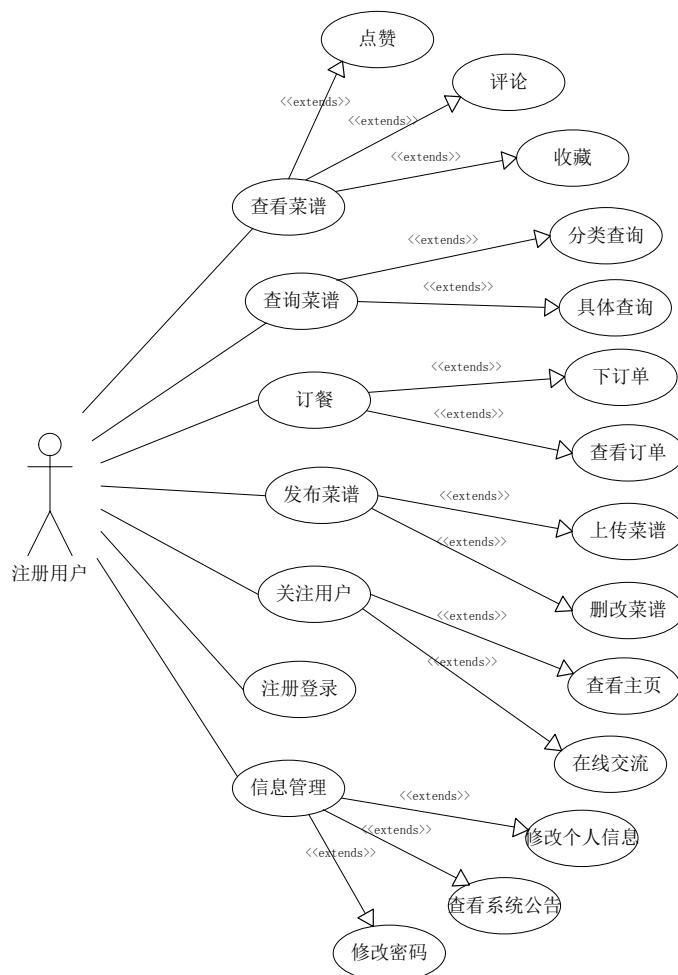


图 2-5 已注册用户用例图

Figure 2-5 Registered user use case diagram

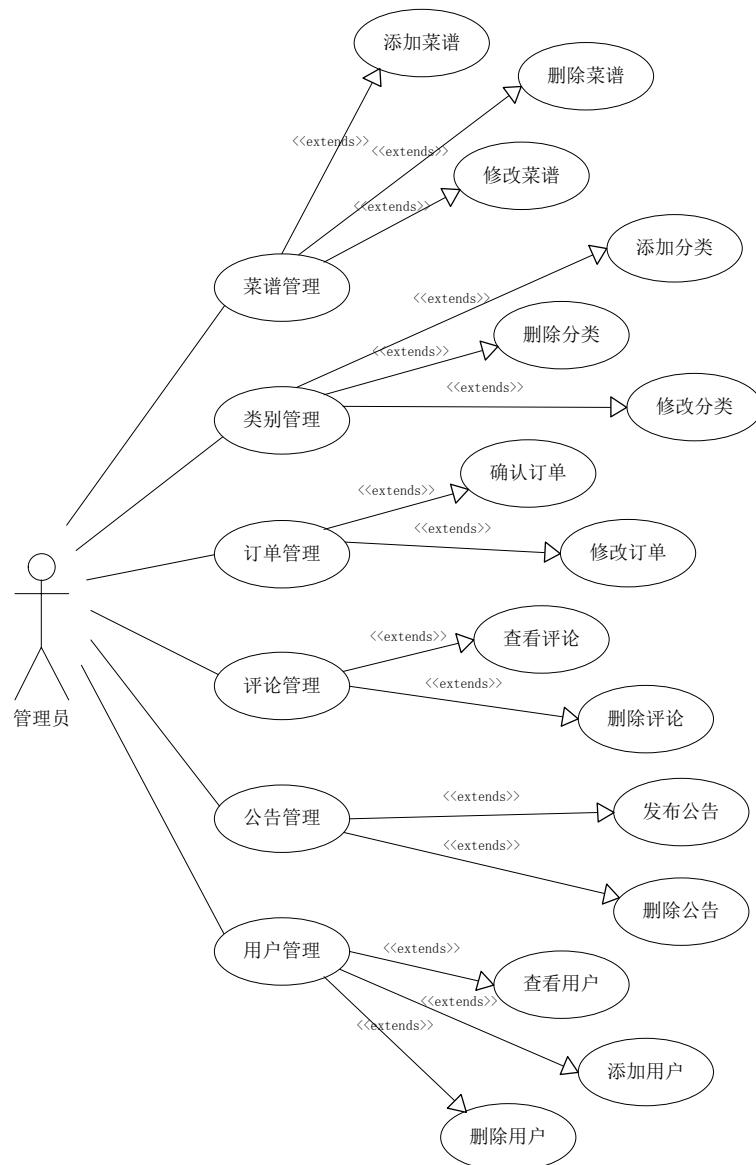


图 2-6 管理员用例图

Figure 2-6 Administrator use case diagram

### (1) 用户用例规约

主要用例的规约描述如下文所列。

1) 登录用例规约, 如表 2-1 所示:

表 2-1 登录用例规约

Table 2-1 Login use case specification

参与者	用户
简要说明	本用例说明用户的登录过程, 开启本次用例的说明是已经注册过的用户。
前置要求	在开始本用例之前需先注册账号。

基本事件流	(1) 用户输入用户名和密码，系统验证用户所输入的信息，验证通过方可登录。 (2) 对比数据库中的匹配项。 (3) 进入到该用户的首页并显示相应信息。
备用事件流	(1) 当用户输入的信息与数据库中对比出错，则系统将会给出错误提示。 (2) 重新输入或者取消操作，本次用例结束。 (3) 提供找回密码路径。
后置要求	系统会在屏幕中央弹出对话框显示成功登录。
扩展内容	选择自动登录按钮，下次可直接自动登录。
特殊需求	无与本用例内容相关的特别需求。

2) 订餐用例规约, 如表 2-2 所示:

表 2-2 订餐用例规约  
Table 2-2 Ordering use case specification

参与者	用户
简要说明	本用例说明用户在前端对菜谱进行订购的过程, 开启本次用例的为已注册并成功登录的用户。
前置要求	网络连接正常, 可成功进入首页; 用户已经登录
基本事件流	(1) 进入选餐页面, 浏览菜谱信息。 (2) 将选定的菜谱放入购物车, 点“结算”进行付款。 (3) 系统经核准生成订单后显示订餐成功信息, 用户可在订单列表区页面查看该订单信息。
备用事件流	(1) 因网络等原因导致下订单失败时, 系统会给出订餐失败提示, 用户重新选购菜谱。 (2) 在点击“确认付款”按钮前用户可随时点击“取消”按钮, 取消订餐操作。
后置要求	成功生成订单, 保存在数据库中
扩展内容	在按“结算”按钮前, 用户可以浏览选择其他菜谱
特殊需求	无与本用例内容相关的特别需求。

3) 发表评论用例规约, 如表 2-3 所示:

表 2-3 发表评论用例规约  
Table 2-3 Comment use case specification

参与者	用户
简要说明	本用例说明用户在前端对菜谱评论的过程, 开启本次用例的则可确定为已注册

	并成功登录的用户。
前置要求	网络连接正常，可成功进入首页；用户已经登录
基本事件流	(1) 浏览菜谱信息，在下方点击“评论”或“回复”，输入评论的内容。 (2) 点击“发表”按钮，发表评论。 (3) 评论区页面显示用户发表的评论内容。
备用事件流	(1) 当用户发表的评论内容包含违禁信息字词时，系统将会给出警示。 (2) 在点击“确认更改”按钮之前，用户可以随时点击“取消”按钮，取消操作。
后置要求	成功发表评论，保存在数据库中
扩展内容	在按“发表”按钮前，用户可以浏览评论信息
特殊需求	无与本用例内容相关的特别需求。

4) 个人信息管理用例规约，如表 2-4 所示：

表 2-4 个人信息管理用例规约  
Table 2-4 Personal information management use case specification

参与者	用户
简要说明	本用例说明用户重新编辑个人详细资料信息的过程，开启本次用例的则可确定为注册并成功登录的用户。
前置要求	网络连接正常，可成功进入首页；用户已经登录
基本事件流	(1) 用户在导航栏点击“我的”，进入用户中心页面。 (2) 编辑个人信息，点击“修改”按钮修改信息。 (2) 系统给出个人信息修改成功或失败的提示。
备用事件流	(1) 若修改个人信息保存失败，系统会给出提示让用户重新输入修改信息。 (2) 在点击“确认更改”按钮之前，用户可以随时点击“取消”按钮，取消操作。
后置要求	个人信息已经更新为修改后的内容
扩展内容	修改信息输入一半或为完全输入点击确认，方可取消恢复之前内容。
特殊需求	无与本用例内容相关的特别需求。

5) 发布菜谱用例规约，如表 2-5 所示：

表 2-5 发布菜谱用例规约  
Table 2-5 Publish recipe use case specification

参与者	用户
简要说明	本用例说明用户在前端上传菜谱的过程, 开启本次用例的则可确定为注册并成功登录的用户。
前置要求	网络连接正常, 可成功进入首页; 用户已经登录
基本事件流	(1) 进入菜谱发布页面, 点击下方的“上传”按钮, 在弹出的编辑界面按提示添加、编辑要上传的菜谱图片、文档, 然后点击“确定”按钮完成菜谱上传。 (2) 菜谱发布页面显示用户上传的菜谱资料。
备用事件流	(1) 当用户发布的菜谱资料包含违禁信息时, 系统将会给出警示。 (2) 在点击“确认”按钮之前, 用户可以随时点击“取消”按钮, 取消上传操作。
后置要求	成功上传菜谱, 保存在数据库中
扩展内容	在按“上传”按钮前, 用户可以浏览其他用户上传的菜谱资料
特殊需求	无与本用例内容相关的特别需求。

6) 注册用例规约, 如表 2-6 所示:

表 2-6 注册用例规约  
Table 2-6 Registration use case specification

参与者	游客
简要说明	本用例说明游客注册为系统用户的过程, 开启本次用例的说明是首次注册的用户
前置要求	网络连接正常, 可成功进入首页
基本事件流	(1) 用户填写个人信息, 如选择头像, 输入用户名, 重复设置两次密码, 填写其他注册信息后点击“注册”按钮进行注册。 (2) 系统验证用户所填写的信息的格式、内容。 (3) 系统核准注册, 保存用户信息。
备用事件流	(1) 当用户输入的信息不符合系统的要求, 则系统将会给出错误提示。 (2) 重新输入或者取消操作, 本次用例结束。
后置要求	用户注册成功, 系统将增加该用户信息, 用户可以根据用户名和密码登录
扩展内容	提示注册成功
特殊需求	无与本用例内容相关的特别需求。

7) 查询菜谱用例规约, 如表 2-7 所示:

表 2-7 查询菜谱用例规约  
Table 2-7 Query menu use case specification

参与者	游客、用户
简要说明	本用例说明对产品查询操作的过程，启用此用例的可以是游客也可以是已注册的用户
前置要求	网络连接正常；能成功进入首页
基本事件流	(1) 用户进入菜谱页面，输入查找的关键字，点击“搜索”。 (2) 系统显示相关的菜谱信息。
备用事件流	(1) 先选择菜谱的分类，然后在分类中输入菜谱名，点击“搜索”。 (2) 如果用户输入的关键字无效，则系统将会给出提示。
后置要求	找到符合查找要求的菜谱或菜谱列表
扩展内容	无与本用例内容相关的扩展内容。
特殊需求	无与本用例内容相关的特别需求。

## (2) 管理员用例规约

1) 管理员登录用例规约，如表 2-8 所示：

表 2-8 登录用例规约  
Table 2-8 Login use case specification

参与者	管理员
简要说明	本用例说明管理员在后台的登录过程，开启次本用例的说明是已在数据库中存储的管理员。
前置要求	在开始本用例之前需是系统中已有的管理员。
基本事件流	输入管理员用户名和密码，点击“登录”按钮进行登录
备用事件流	当管理者输入的信息与数据库中对比出错，则系统将会给出错误提示。重新输入或者取消操作，本次用例结束。
后置要求	登录成功，以动态效果进入后台。
扩展内容	无与本用例内容相关的扩展内容。
特殊需求	无与本用例内容相关的特别需求。

2) 用户管理用例规约, 如表 2-9 所示:

表 2-9 用户管理用例规约  
Table 2-9 User management use case specification

参与者	管理员
简要说明	本用例说明对用户信息进行各类操作的过程, 启用此用例的为已登录的账户管理员。
前置要求	在开始本用例之前需是系统中已有的管理员。
基本事件流	点击导航栏“用户管理”, 点击“用户列表”进入管理页面。选中数据后, 点击“新增”按钮添加用户信息, 点击“删除”可以删除用户信息
备用事件流	①点击导航栏“用户管理”, 点击“用户列表”进入管理页面, 输入用户名, 点击“查询”可以查询用户信息; ②点击导航栏“用户管理”, 点击“用户列表”进入管理页面, 点击“修改”可以修改用户信息;
后置要求	系统会在屏幕中央弹出话框显示操作信息。
扩展内容	无与本用例内容相关的扩展内容。
特殊需求	无与本用例内容相关的特别需求。

3) 评论管理用例规约, 如表 2-10 所示:

表 2-10 评论管理用例规约  
Table 2-10 Comments on management use case specifications

参与者	管理员
简要说明	本用例说明对菜谱评论信息进行各类操作的过程, 启用此用例的为已登录的账户管理员。
前置要求	在开始本用例之前需是系统中已有的管理员。
基本事件流	点击导航栏“评论管理”显示管理菜单项, 点击“评论列表”进入管理界面, 选择要删除的评论, 点击“删除”按钮删除相关信息
备用事件流	在点击“确认删除”之前, 可取消前面的选择
后置要求	系统会在屏幕中央弹出话框显示操作信息。
扩展内容	无与本用例内容相关的扩展内容。

特殊需求	无与本用例内容相关的特别需求。
------	-----------------

4) 公告管理用例规约, 如表 2-11 所示:

表 2-11 公告管理用例规约  
Table 2-11 Announcement management use case specification

参与者	管理员
简要说明	本用例说明对菜谱的进行分类的各类操作的过程, 启用此用例的为已登录的账户管理员。
前置要求	在开始本用例之前需是系统中已有的管理员。
基本事件流	账户管理员选择已存在的菜谱分类做出维护或者删除操作, 对未存在需上传的菜谱分类做出新增操作并保存。
备用事件流	如果有某处操作不合规, 则会给出错误提示, 则菜谱分类管理用例结束。
后置要求	系统会在屏幕中央弹出话框显示操作信息。
扩展内容	无与本用例内容相关的扩展内容。
特殊需求	无与本用例内容相关的特别需求。

## 2.4 业务流程分析

前面的用例图和用例规约是对系统总体的静态功能描述。通过活动图可以对系统结构有一个很好的补充<sup>[23]</sup>。本小节通过活动图建模, 对系统业务流程进行描述。下面给出管理员管理、用户操作的活动图。

管理员可以在后台进行管理操作, 对用户、菜谱、公告和评论等信息可进行增加、删除、修改、查询等操作。管理员活动图如图 2-7 所示。

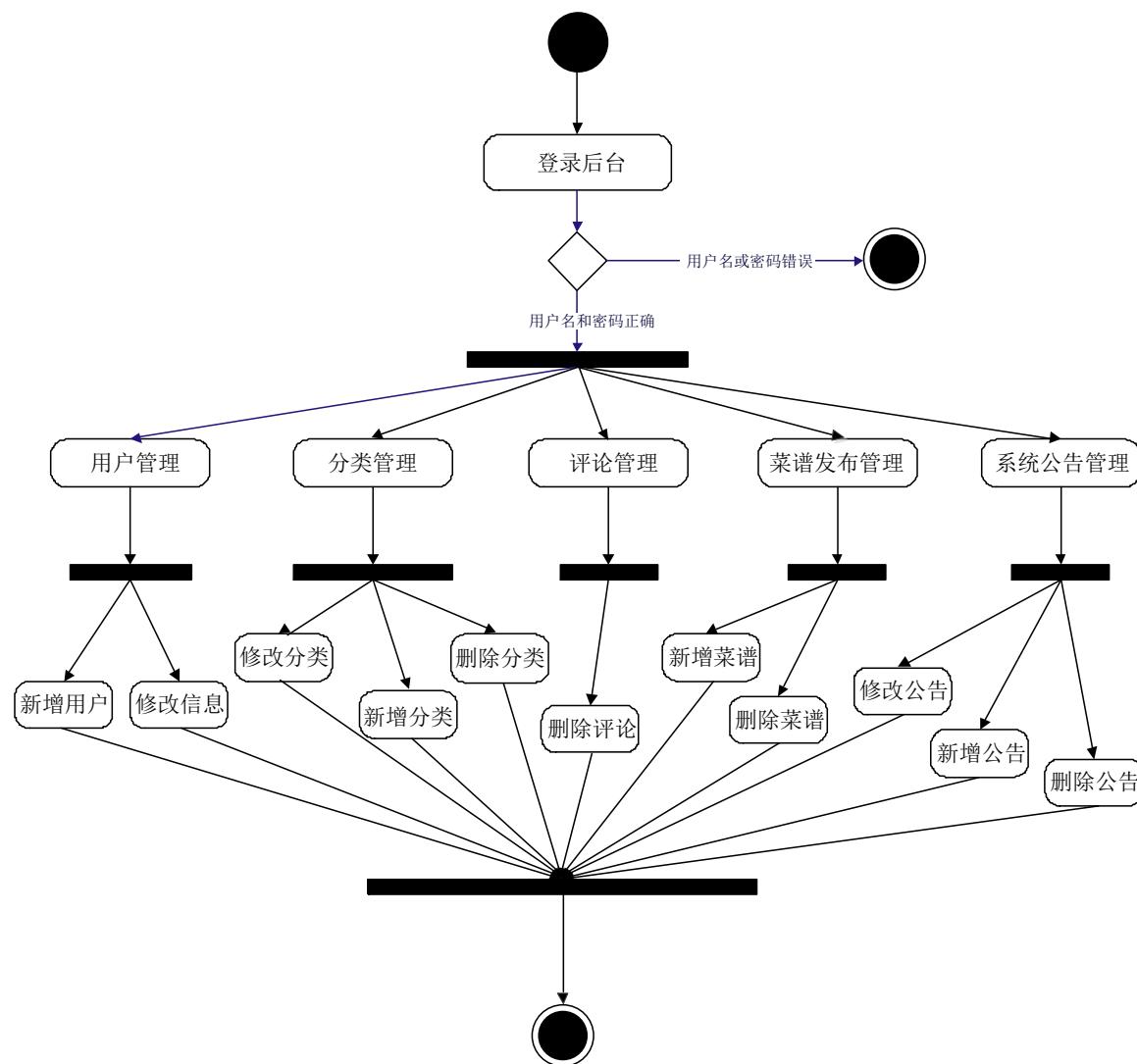


图 2-7 管理员活动图

Figure 2-7 Administrator activity chart

图 2-7 所表示的管理员活动流程如下：管理员输入用户名和密码登录系统，如输入错误则终止登录，如正确则登录系统；登录系统后管理员对用户、分类、评论、菜谱发布、系统公告等功能版块进行增加、删除、修改等管理操作，操作完成后生效保存至系统数据库。

用户可以查询、浏览菜谱，上传菜谱，关注用户，收藏菜谱、管理个人信息等，用户业务流程的活动图如图 2-8 所示。

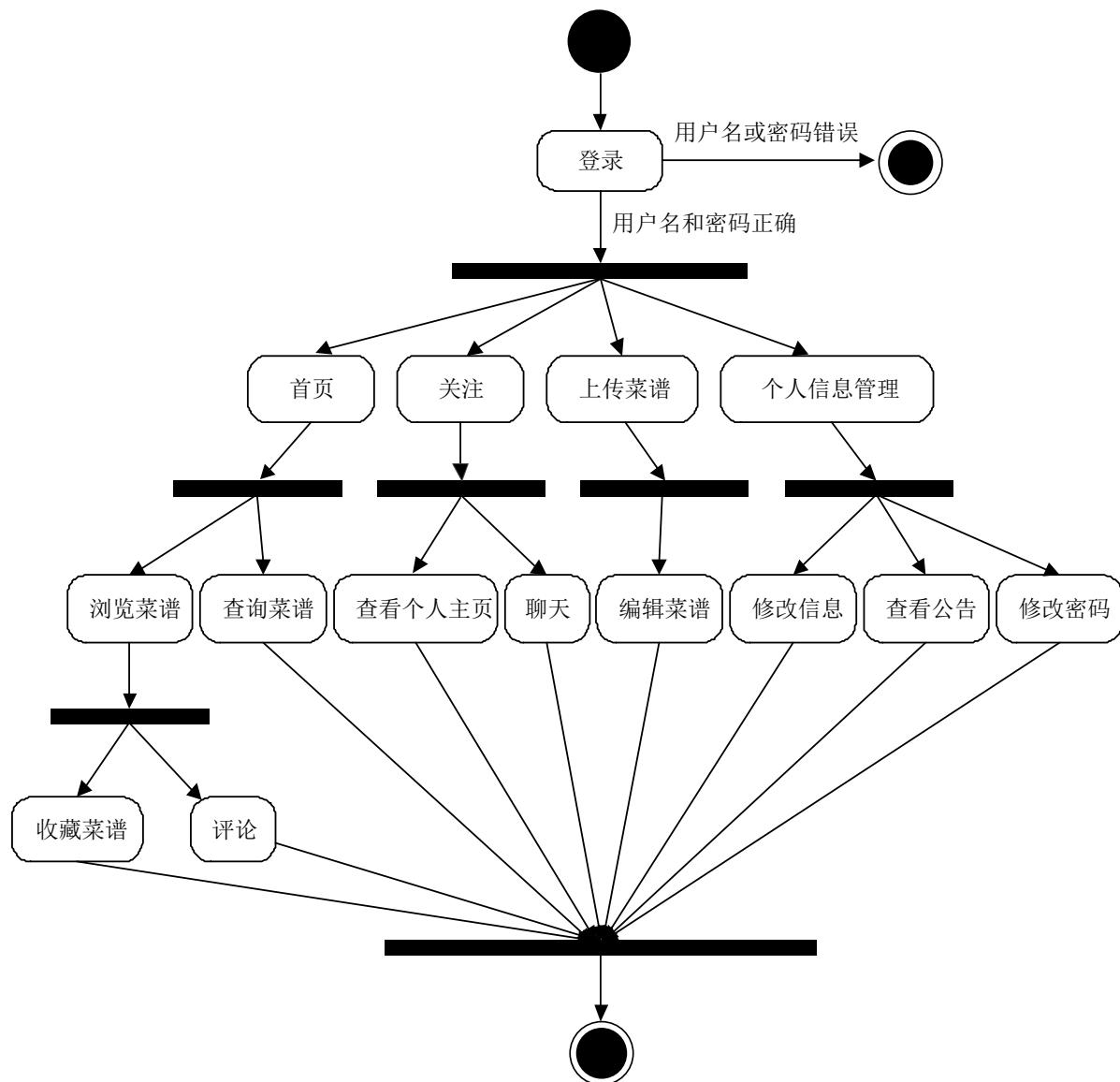


图 2-8 用户活动图  
Figure 2-8 User activity diagram

图 2-8 用户活动图流程：用户输入用户名和密码，如错误则退出登录，如正确则登录系统；登录成功后用户可到首页、关注、上传菜谱、个人信息管理等功能版块进行浏览、查询、修改等相应的操作；操作完成后生效保存至系统数据库。

## 2.5 本章小结

本章主要对基于 Android 平台的手机订餐系统的需求分析工作进行了描述，在描述时首先对系统主要的功能性需求、非功能性需求以及数据库需求给出了较为详细的分析介绍。

本章节首先从用户和管理员这两大角色的角度去对系统的整体功能需求进行了分析，通过系统功能模块图展示了系统的整体基本功能，接下来分别对本系统的用户端和服务端的功能需求进行了分析，确定了用户端前台所包含的功能模块能够满足用户订餐的相关需求，明确了服务端后台需求具备的管理功能。另外对系统的性能需求进行了分析，明确了系统的性能要求，并且对系统功能需求的主要用例给出了用例规约说明。通过对系统做出的以上分析，基本上确定了此系统应该实现哪些功能<sup>[24]</sup>。

## 第 3 章 系统设计

本章主要在系统需求分析的基础上，对订餐交易手机 app 系统进行详细设计。首先对系统的总体架构进行设计，分析介绍了系统所采用的 SSM 技术框架；然后介绍了主要功能模块的设计和菜谱发布时序图，最后对系统数据库进行了详细的分析和设计。

### 3.1 系统架构设计

本手机订餐交易管理系统采用 MVC 设计模式设计架构，使用 spring MVC 框架。MVC 的全名是 Model-View-Controller，也就是把一个应用程序划分为视图层、模式层和控制层<sup>[25]</sup>，将系统分为这三个大层能有效降低耦合度，便于系统的维护和扩展<sup>[26]</sup>。MVC 模型关系如图 3-1 所示。

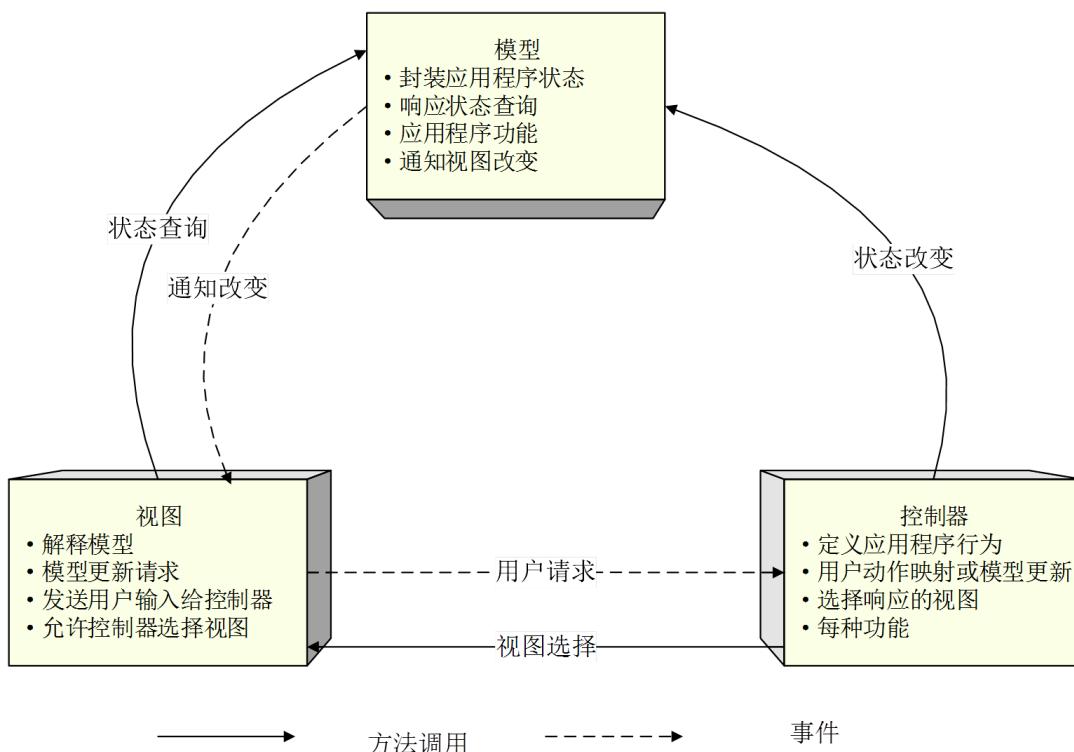


图 3-1 MVC 结构示意图  
Figure 3-1 MVC structure diagram

本系统采用前后台分离的 MVC 模式，后台负责数据生成与业务逻辑的处理，前台 UI 界面负责数据展示与收集用户输入的数据，并将用户输入数据返回给后台处理。前后台通过 Json 数据交互进行耦合，非常便于项目后期的维护与功能扩展。

下面说明如何使用 SOA (Service-Oriented Architecture, 面向服务体系架构) 思想和 SpringCloud (是 Spring 提供的一套微服务标准) 进行服务的拆分。在点餐餐饮系统中, 可以将功能模块从主体方面进行拆分, 主要有以下子系统功能:

用户系统: 用户注册, 登录, 用户信息查看, 用户信息修改, 用户手机号绑定

商品系统: 商品的上传, 修改, 删除, 查看, 上架, 下架

订单系统: 用户下单, 取消订单

支付系统: 支付宝支付, 微信支付

交流系统: 信息的发送, 信息的接收

统计系统: 各种纬度数据的统计

内容系统: 发布文章(增删改查), 发布视频(增删改查)

上面的每个功能下的子功能都对应相应的一个接口, 这个接口可以供该功能模块使用, 也可以为其它功能模块提供相应的服务, 比如: 订单模块的用户下单功能需要获取用户模块中下单用户的信息, 就需要调用相应的接口<sup>[27]</sup>。

订餐 app 系统框架如图 3-2 所示。

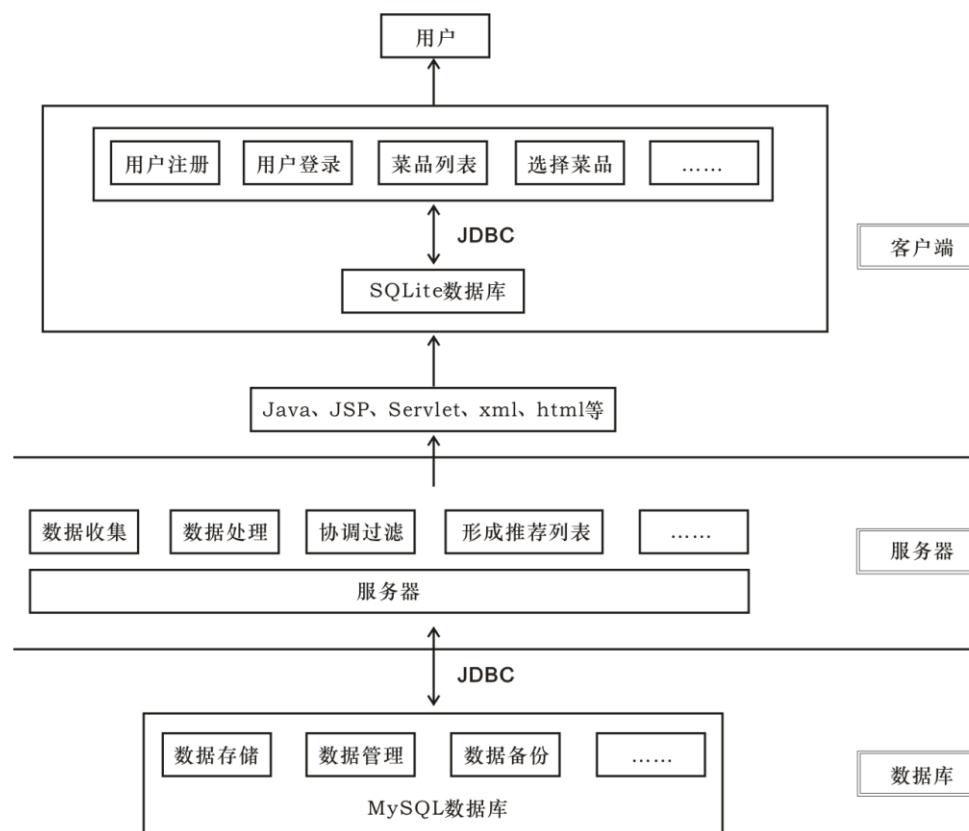


图 3-2 订餐交易 app 系统框架图

Figure 3-2 Framework of ordering transaction app system

本订餐 app 系统设计分为前端与后端两大模块，并且分别对应前台与后台，前台主要是满足客户的使用需求，后台则供管理者对系统的优化与功能维护操作。

本订餐 app 系统在终端涉及到管理员和消费者这两个用户群，在进行业务处理时则用到服务器，数据进行存储时则要用到本地数据库或远程数据库，具体的网络拓扑图如图 3-3 所示。

在本手机订餐交易管理系统整体架构中，服务器组包括 WEB 服务器（Apache Tomcat）和系统数据库服务器（MySQL），Web 服务器向基于此订餐管理系统的各级用户提供网页访问服务。客户端发出 HTTP 请求后 WEB 服务器做出响应，然后将与数据库服务器交互结果传递给客户端。客户端、服务器端交互主要采用无线网络链接，各级用户通过智能手机客户端订餐交易管理系统与服务器组通信联系。数据库服务器采用 MySQL，与 WEB 服务器相连，实现对数据的操作。在本手机订餐交易管理系统的网络拓扑结构中，智能手机客户端不管是在食堂局域网内或远程网络中，都是使用无线网络与 web 应用服务器及数据库服务器进行链接。

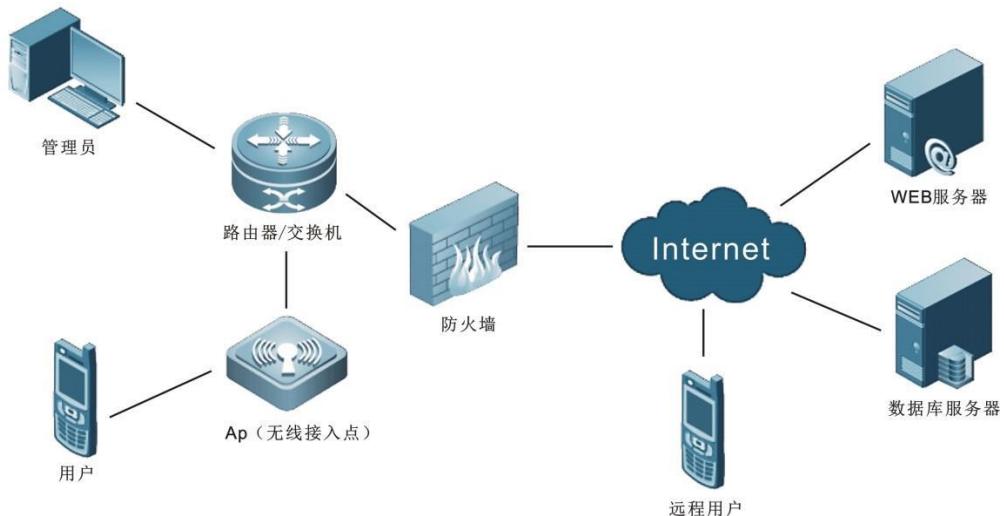


图 3-3 订餐交易 app 系统网络拓扑图  
Figure 3-3 Network topology of ordering transaction app system

本手机订餐交易管理系统的服务端负责数据的提供与维护，用户端负责信息的发布和显示。该系统的物理架构主要由 Web 服务器与 MySQL 数据库服务器、无线网络及手机订餐交易管理系统客户端等组成，Android 智能手机客户端通过无线网络来访问后台服务器的数据信息。

在本订餐管理系统中，采用 MySQL、Eclipse 和 Tomcat 的组合来开发 Web 服务器端，

并以 JSON 格式进行前后台数据交互。客户端建立了 Android Studio 的开发环境，将菜谱信息分类发布到 Android 手机客户端中，用户只需要下载客户端软件 (.apk 文件) 安装至手机中即可。Web 服务器和数据库服务器由开发者在服务器中进行搭建和发布。

### 3.2 主要功能模块设计

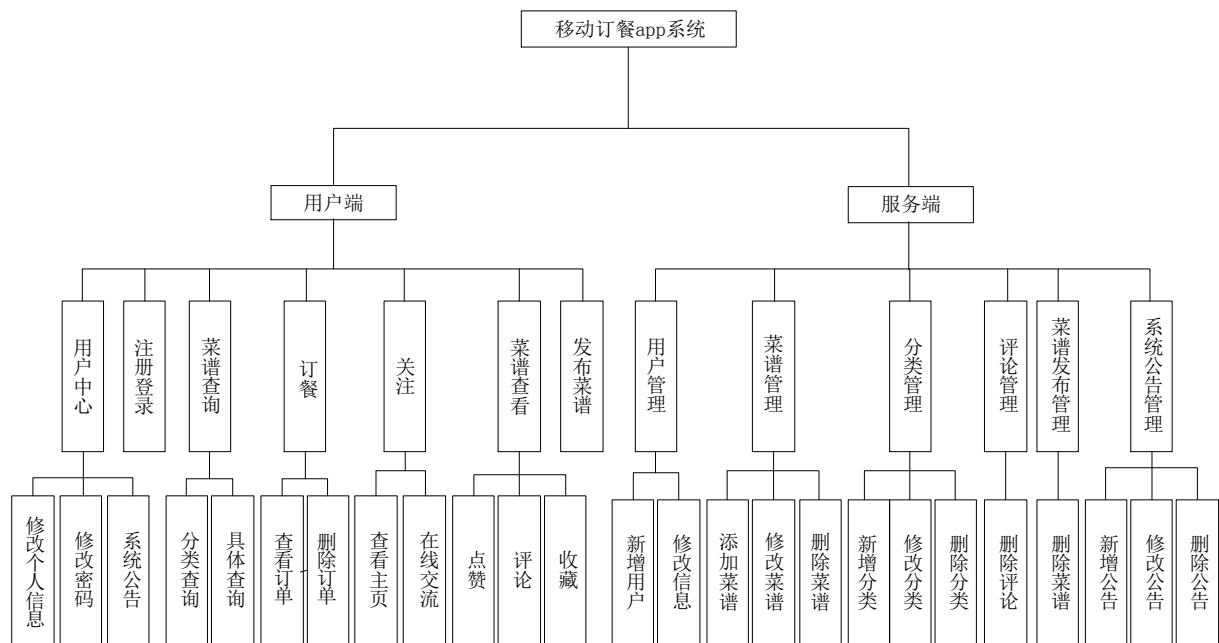


Figure 3-4 Module diagram of ordering app system

根据系统功能需求，本订餐交易管理系统由用户端和服务端两部分组成。用户端的功能主要包括发布菜谱功能、查看菜谱功能、用户中心功能、订餐功能、注册登录功能、关注功能；服务器端是管理员对后台信息进行管理，包括用户管理、分类管理、分类管理、评论管理、菜谱发布管理、系统公告管理等功能。

### 3.3 数据库设计

依据以上分析可以对数据库进行设计。根据订餐交易系统的需求分析和功能模块划分，本系统具有以下实体类，其属性为：

- (1) 公告实体类：属性有 ID、标题、内容、时间；
- (2) 评论实体类：属性有用户名、ID、评论时间、发布人、评价内容、状态等；

- (3) 菜谱实体类：属性有 ID、菜谱名、做法、类型、图片、用户名；  
(4) 用户实体类：属性有 ID、用户名、密码、性别、电话、地址、QQ、喜欢菜系、邮箱、生日等；  
(5) 订单实体类：属性有 ID、菜谱名、时间、数量、总额、状态、用户名；  
(6) 上传菜谱实体类：属性有用户名、ID、上传时间、发布人、菜谱名、上传内容、上传图片等；  
(7) 消息实体类：属性有 ID、用户名、朋友名、发布时间、消息等。

主要的实体-关系图分别如下所述。

- (1) 公告实体图，如图 3-5 所示：

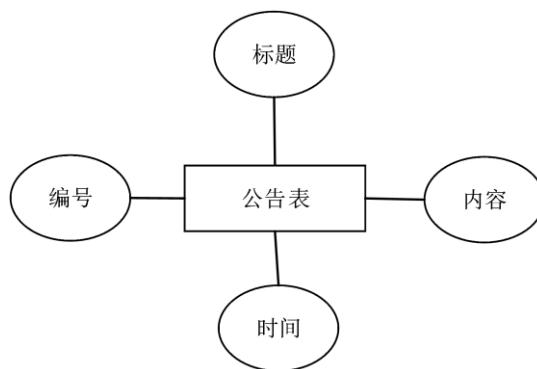


图 3-5 公告实体图

Figure 3-5 Entity diagram of announcement

- (2) 评论实体图，如图 3-6 所示：

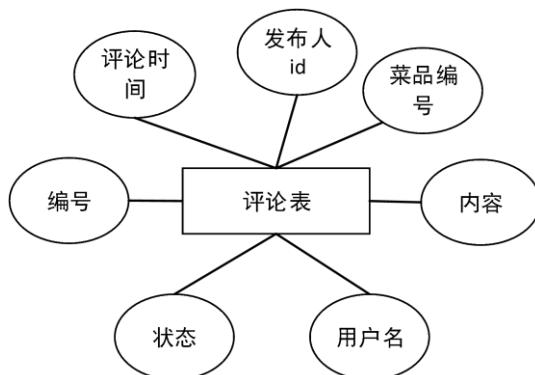


图 3-6 评论实体图

Figure 3-6 Comment entity diagram

- (3) 菜谱实体图，如图 3-7 所示：

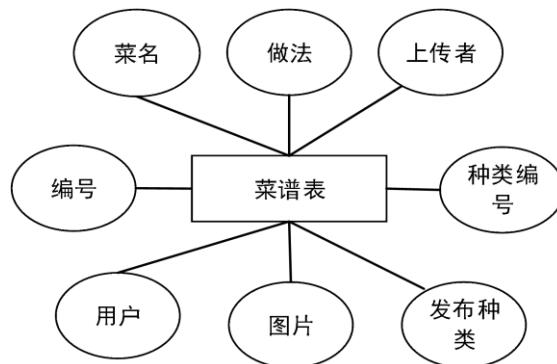


图 3-7 菜谱实体图

Figure 3-7 Menu entity diagram

(4) 用户实体图, 如图 3-8 所示:

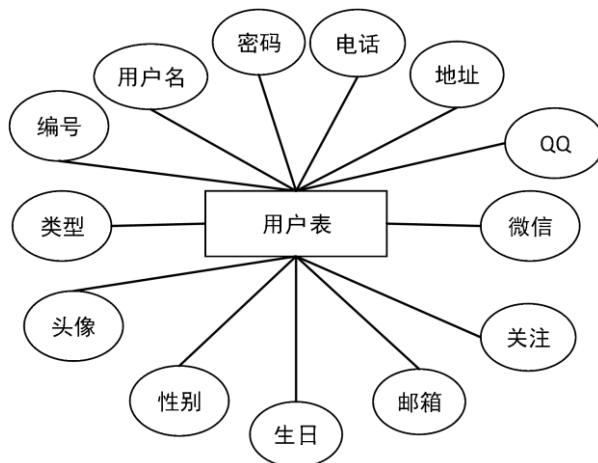


图 3-8 用户实体图

Figure 3-8 User entity diagram

(5) 订单实体图, 如图 3-9 所示:

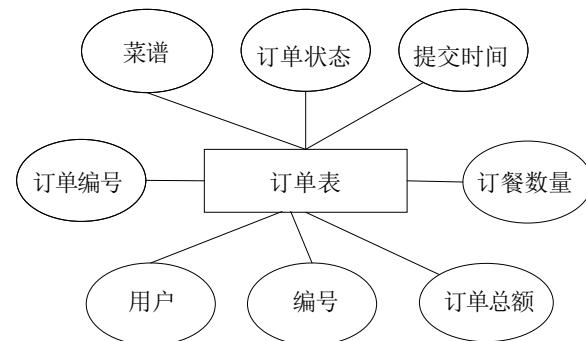


图 3-9 订单实体图

Figure 3-9 Order entity diagram

(6) 上传菜谱实体图, 如图 3-10 所示:

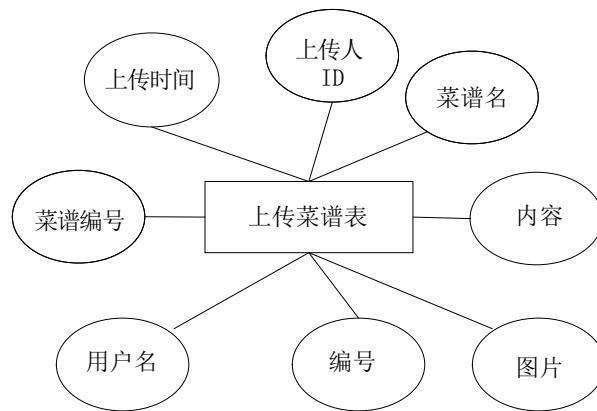


图 3-10 上传菜谱实体图  
Figure 3-10 Upload recipe entity map

(7) 消息实体图, 如图 3-11 所示:

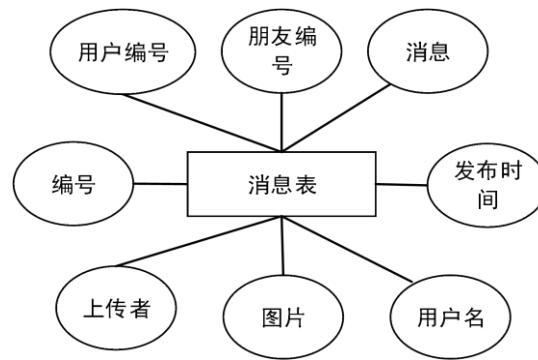


图 3-11 消息实体图  
Figure 3-11 Message entity diagram

根据实体间的联系, 可以画出如下图 3-12 所示的总体 E-R 图:

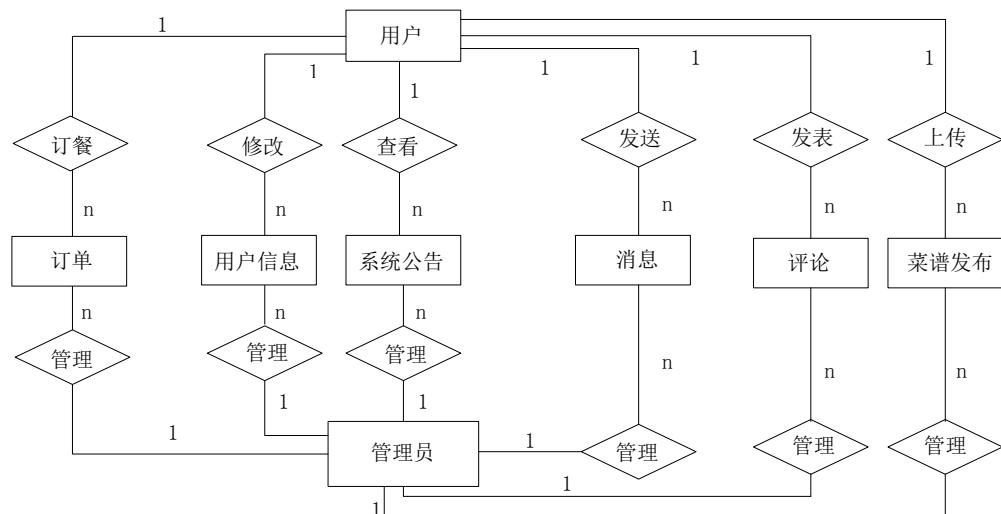


图 3-12 总体 E-R 图  
Figure 3-12 Overall E-R diagram

本订餐交易 app 系统需要后台数据库，根据以上实体-关系图设计，下面列举一些主要的数据库表的设计结果。

表 3-1 系统公告表  
Table 3-1 System announcement

列名	数据类型	长度	主键	允许空	说明
id	int	11	是	否	编号
title	varchar	50	否	否	标题
note	varchar	50	否	否	内容
ndate	varchar	50	否	否	时间

表 3-2 评论信息表  
Table 3-2 Comment information table

列名	数据类型	长度	主键	允许空	说明
id	int	11	是	否	ID
ndate	varchar	50	否	否	评论时间
uid	varchar	50	否	否	发布人 id
note	varchar	50	否	否	内容
username	varchar	50	否	否	用户名
type	varchar	50	否	否	状态
pid	varchar	50	否	否	菜谱编号

表 3-3 菜谱表  
Table 3-3 Menu

列名	数据类型	长度	主键	允许空	说明
id	int	11	是	否	编号
gname	varchar	50	否	是	菜名
note	varchar	50	否	是	做法
type	varchar	50	否	是	类型
img	varchar	50	否	是	图片
username	varchar	50	否	是	用户
uid	int	11	否	否	上传者
gtype	varchar	255	否	是	发布种类

表 3-4 用户信息表  
Table 3-4 User information table

列名	数据类型	长度	主键	允许空	说明
id	int	11	是	否	编号
username	varchar	50	否	否	用户名
password	varchar	50	否	否	密码
roletype	varchar	50	否	是	类型
email	varchar	50	否	是	邮箱
tel	varchar	50	否	是	电话
qq	varchar	50	否	是	QQ
sex	varchar	50	否	否	性别
birth	varchar	50	否	是	生日
img	varchar	50	否	是	头像
address	varchar	50	否	是	地址
guanzhu	varchar	50	否	是	关注

表 3-5 订单表  
Table 3-5 Order form

列名	数据类型	长度	主键	允许空	说明
id	int	11	是	否	ID
username	varchar	50	否	否	用户名
gname	varchar	50	否	是	菜谱名
pid	varchar	50	否	否	订单编号
pnumber	varchar	11	否	否	数量
pamount	varchar	11	否	否	金额
type	varchar	50	否	否	状态
udate	varchar	50	否	否	提交时间

表 3-6 上传菜谱表  
Table 3-6 Upload menu

列名	数据类型	长度	主键	允许空	说明
id	int	11	是	否	ID
udate	varchar	50	否	否	上传时间
uid	varchar	50	否	否	上传人 id

note	varchar	50	否	否	内容
img	varchar	50	否	是	图片
username	varchar	50	否	否	用户名
gname	varchar	50	否	是	菜谱名
pid	varchar	50	否	否	菜谱编号

表 3-7 消息表

Table 3-7 Message table

列名	数据类型	长度	主键	允许空	说明
id	int	11	是	否	编号
uid	int	11	否	是	用户编号
fid	int	11	否	是	朋友编号
note	varchar	500	否	是	消息
ndate	varchar	50	否	是	发布时间
username	varchar	50	否	是	用户名
fusername	varchar	50	否	是	朋友名
img	varchar	200	否	是	图片

系统使用 Navicat for MySQL 实现数据库的可视化。打开 Navicat（数据库管理与设计工具）后链接到数据库，从服务器上新建数据库。为数据库命名，然后按照上述分析与设计建表。

### 3.4 客户端详细设计

#### 3.4.1 开发环境设计

(1) 开发环境

1) 操作系统

Windows 7 以上操作系统或 Linux 操作系统

2) 程序开发工具

采用 MyEclipse, Android SDK, JDK

3) 数据库管理工具

MySQL Community Server 6.0 以上 Windows 数据库版本

4) Web 界面设计工具

使用 jQuery EasyUI

### 3.4.2 用户端功能模块设计

用户端的功能模块包括注册、登录、订餐、查询订单、发布菜谱、关注等模块，是一个各模块之间功能相互衔接的整体。

#### (1) 用户注册

功能描述：用户完成注册帐号。

输入：用户名、密码和电话等。

输出：用户 ID。

实现逻辑：用户注册的业务流程如图 3-13 所示。

数据库：user\_info 表、operation\_log 表。

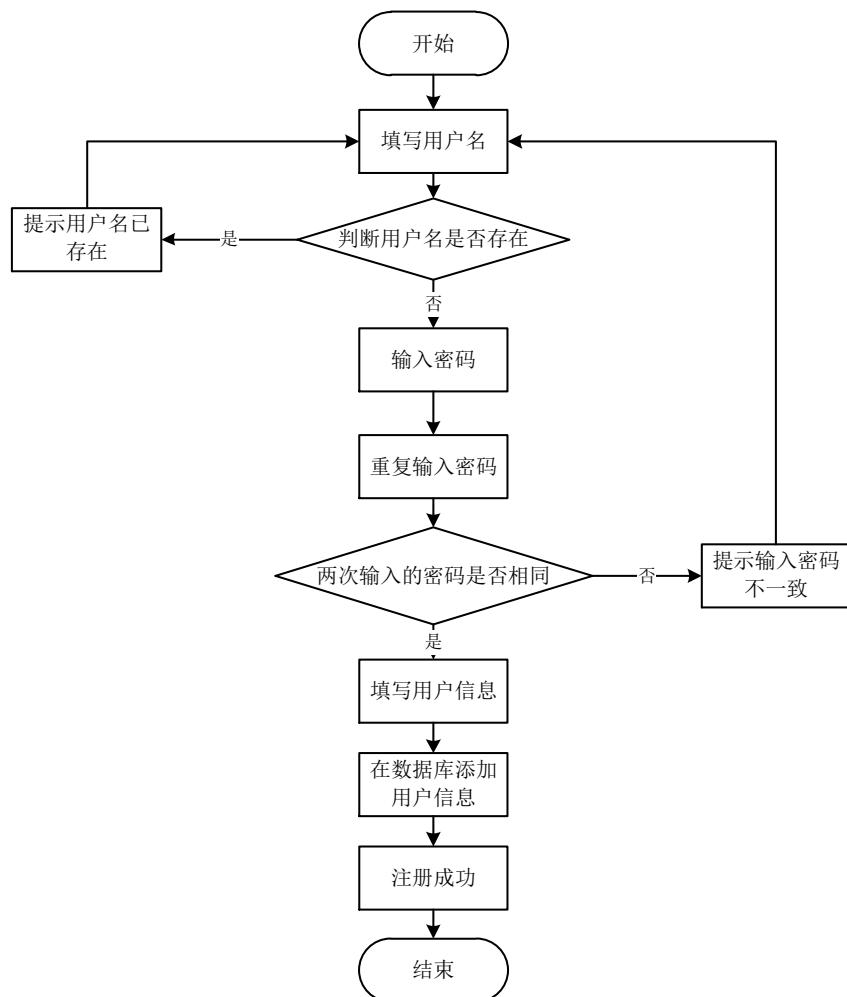


图 3-13 注册流程图  
Figure 3-13 Registration flow chart

## (2) 用户登录

功能描述：用户输入帐号、密码登录系统。

输入：帐号、密码。

输出：系统首页。

实现逻辑：用户注册的业务流程如图 3-14 所示。

数据库：user\_info 表、operation\_log 表。

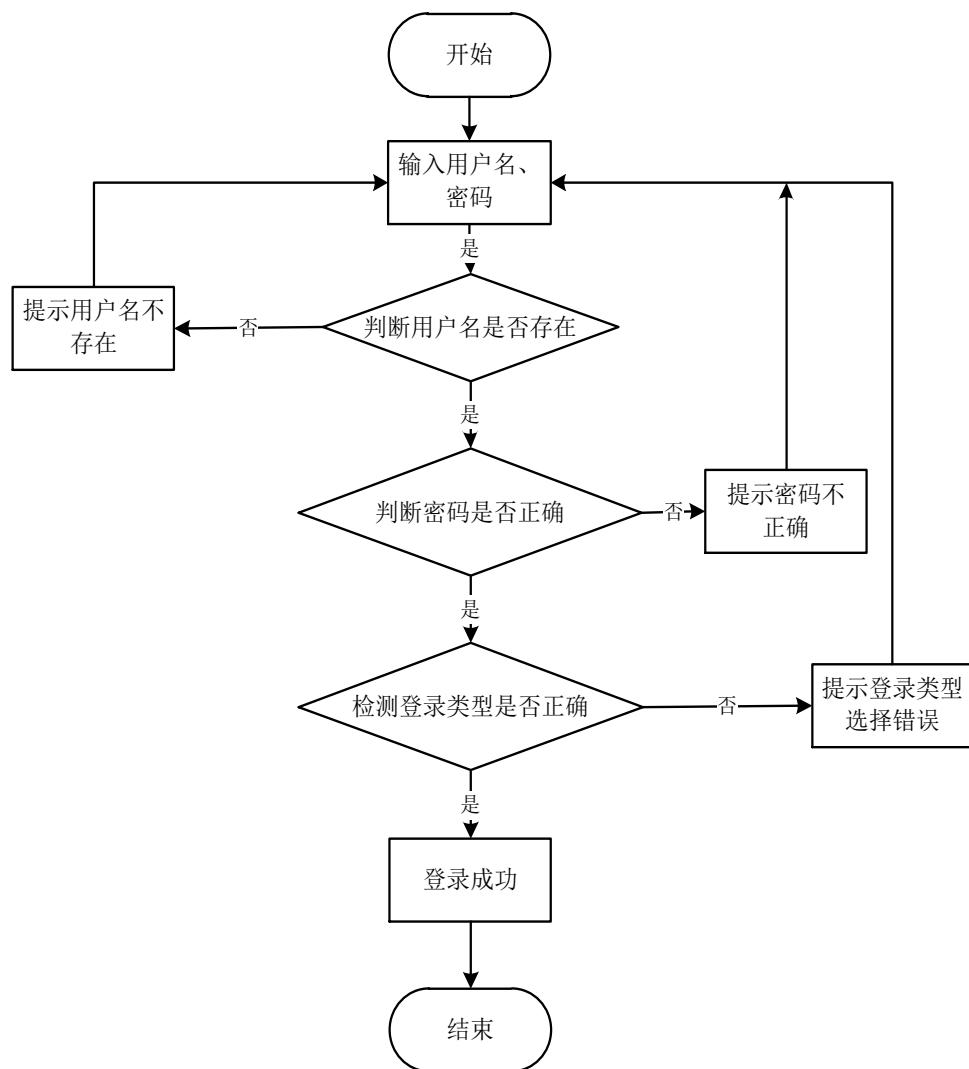


图 3-14 登录流程图

Figure 3-14 Login flow chart

### (3) 用户订餐

功能描述：用户浏览菜单、订购想要菜谱。

输入：菜谱放入购物车、结算。

输出：订单 ID。

实现逻辑：用户订餐的业务流程如图 3-15 所示。

数据库：user\_info 表、order\_info 表、operation\_log 表、search\_info 表及 dish\_info 表。

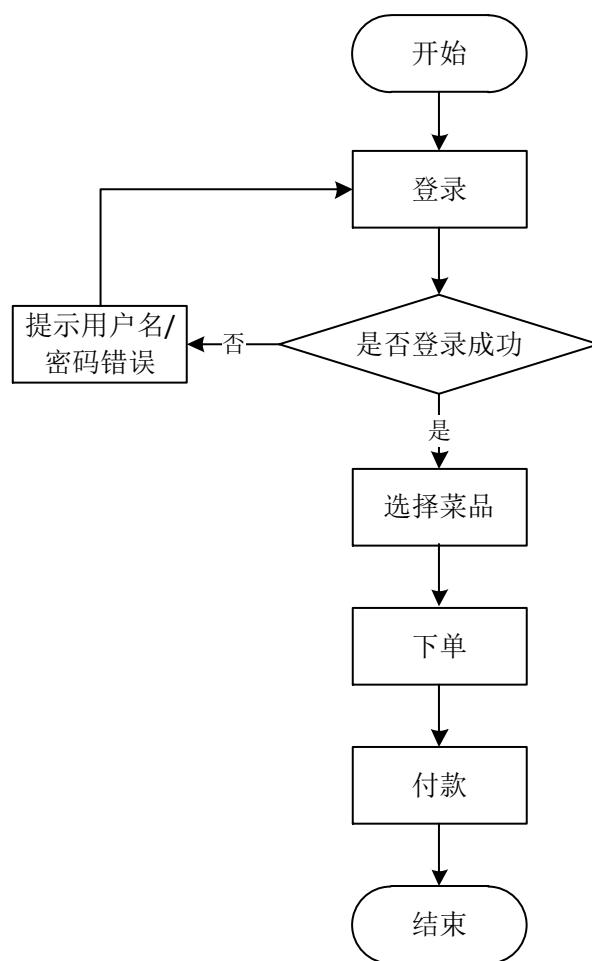


图 3-15 订餐流程图  
Figure 3-15 Ordering flow chart

### (4) 订单查询

功能描述：用户查看自己订单的执行流程和进度。

输入：订单 ID。

输出：在订单跟踪列表中显示所查订单信息。

实现逻辑：用户查询订单的订单跟踪流程如图 3-16 所示。

数据库：user\_info 表、order\_info 表、operation\_log 表、search\_info 表及 dish\_info 表。

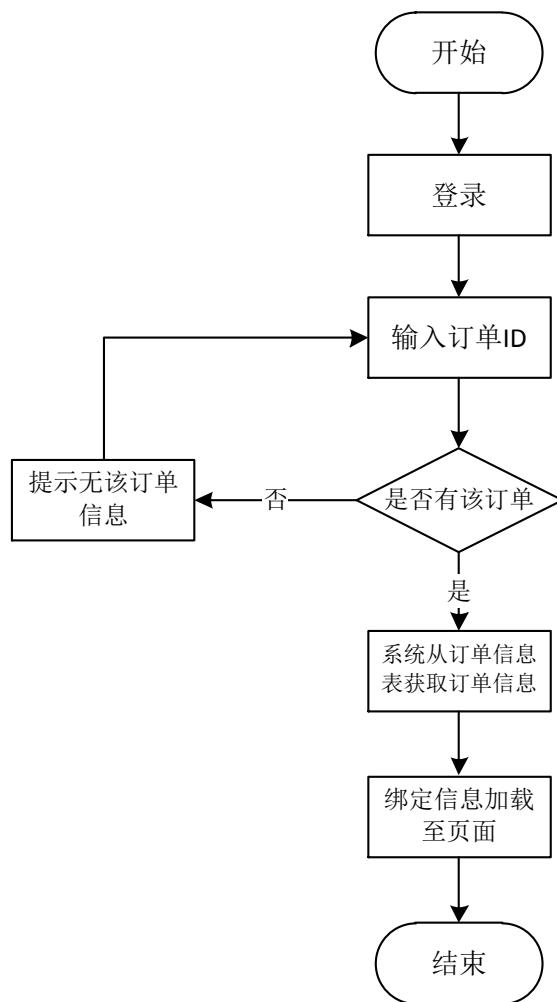


图 3-16 订单查询流程图  
Figure 3-16 Order query flow chart

### (5) 发布菜谱

功能描述：用户上传发布菜谱。

输入：用户 id，餐品 id，餐品名。

输出：在菜谱交流页面中显示所上传的菜谱信息。

实现逻辑：用户登录进入首页，选择底部导航栏的“发布”，进入菜谱编辑页面。

用户可以选择发布菜谱，可以选择分类（如川菜、家常菜、甜品、西餐等），上传步骤图片并附上文字说明等<sup>[28]</sup>。发布后，信息会存储在数据库中，并且菜谱列表也会更新。用户发布菜谱时序如图 3-17 所示。

数据库：user\_info 表、operation\_log 表、category\_info 表及 dish\_info 表。

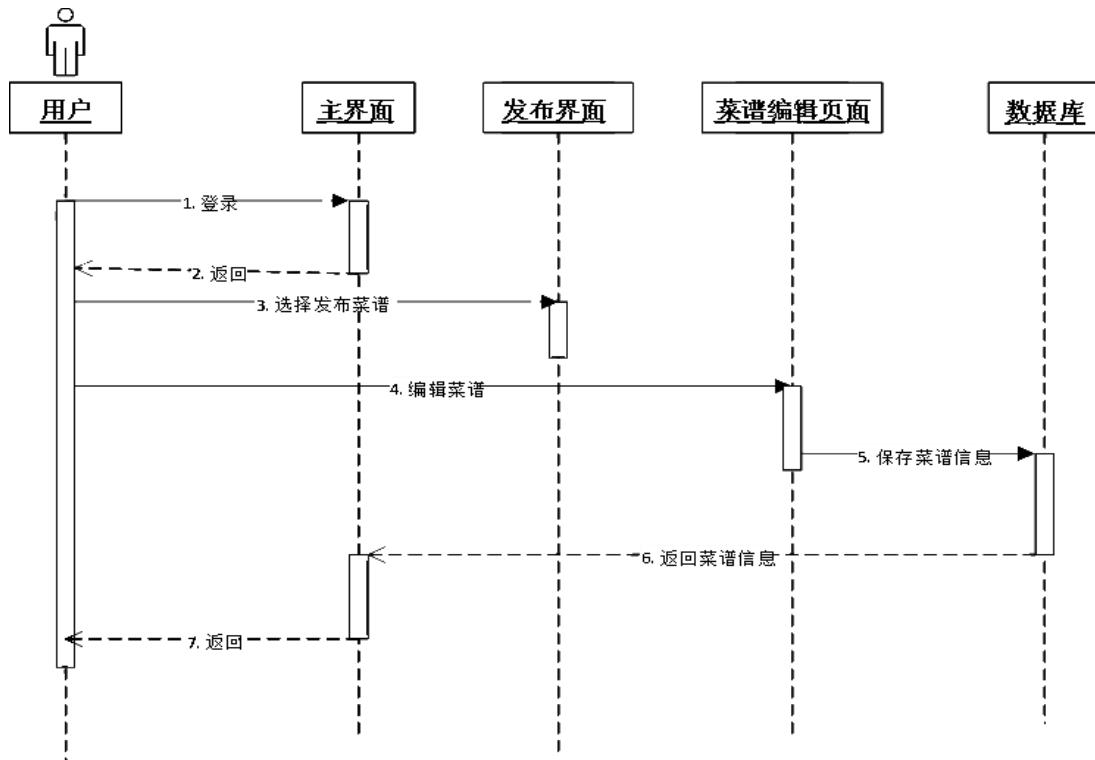


图 3-17 菜谱发布时序图  
Figure 3-17 Menu publishing sequence chart

### 3.5 服务端详细设计

服务端分为用户管理、分类管理、菜谱发布管理、评论管理、系统公告管理等模块。

用户管理模块主要负责管理用户的信息，比如修改密码等；分类管理模块主要负责餐品的分类和餐品的增、删、改等；菜谱发布管理模块主要负责对用户发布的菜谱进行管理、维护；评论管理模块主要负责管理用户发表的评论，对评论进行删、改等操作；进行系统公告管理模块主要负责系统公告的发布、修改、删除等。

服务端采用目前流行的 MVC 三层架构模式，采用 MVC 模式有利于快速开发和实现代码的解耦，也便于后期的功能扩展。

#### 3.5.1 用户管理

(1) 功能描述：对注册用户的信息进行增、改、删等操作。

- (2) 输入：用户 id，用户名，密码，姓名，电话，地址。
- (3) 输出：系统将用户信息保存至 MySQL 数据库中。
- (4) 实现逻辑：用户管理的主要功能是对注册用户信息的查看、修改和删除。这三个功能分别用 viewUser、updateUser、disableUser 来实现。增加、删除用户流程如图 3-18、图 3-19 所示：

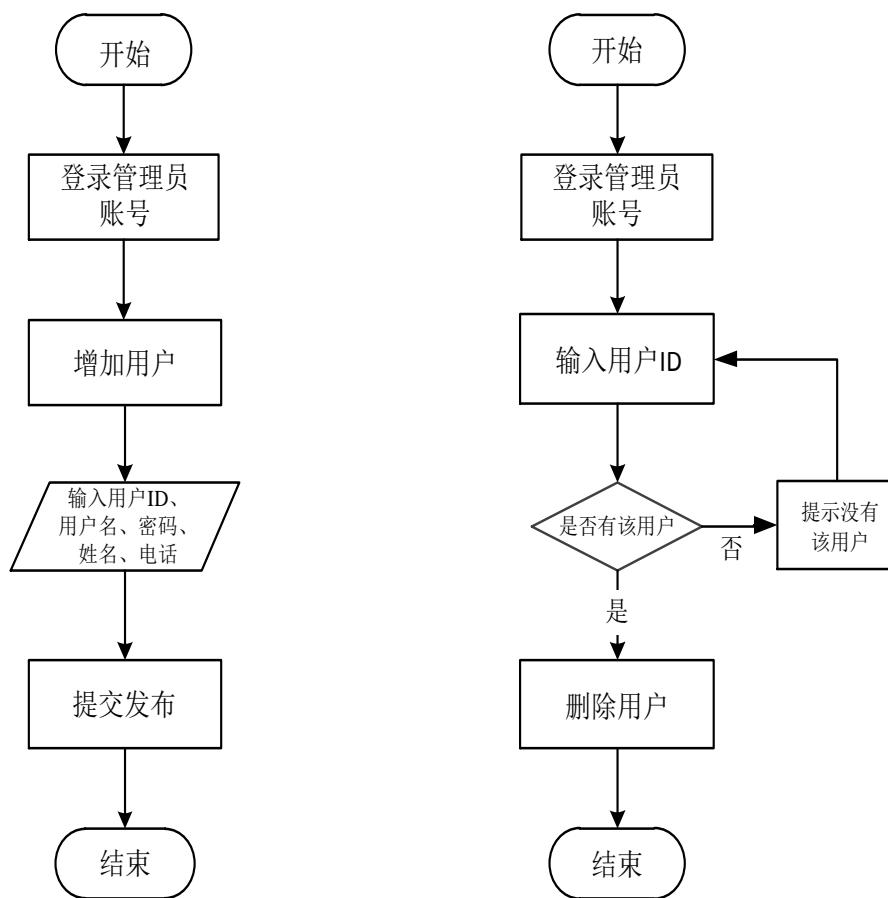


图 3-18 增加用户流程图

Figure 3-18 Add user flow chart

图 3-19 删除用户流程图

Figure 3-19 Delete user flowchart

### 3.5.2 菜谱管理

- (1) 功能描述：实现对餐品的分类和增、删、改操作。
- (2) 输入：餐品名，餐品类别，餐品 id，价格，数量。
- (3) 输出：系统将餐品信息保存至 MySQL 数据库中。
- (4) 实现方式：菜谱管理的主要有四个功能，即让管理员查看系统现有的各类餐品、上架新餐品、下架冷销餐品和调整餐品价格。这四个功能分别用 viewFood、upFood、

downFood、updateFood 来实现。餐品添加、删除流程图如图 3-20、图 3-21 所示：

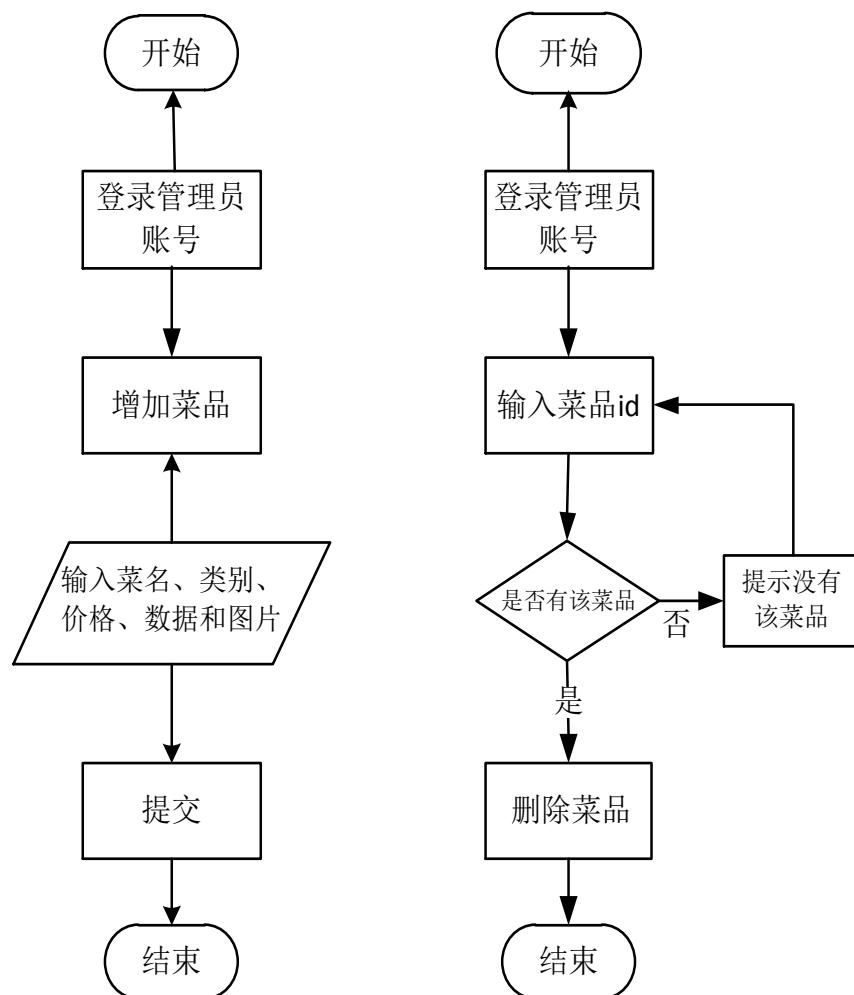


图 3-20 增加菜品流程图

Figure 3-20 Add food flow chart

图 3-21 删除菜品流程图

Figure 3-21 Delete menu flowchart

### 3.5.3 评论管理

- (1) 功能描述：实现对用户评论的增、删、查操作。
- (2) 输入：评论 id，用户名，发布人 id，内容，状态
- (3) 输出：系统将评论信息保存至 MySQL 数据库中。
- (4) 实现方式：评论管理的主要功能是让管理员查看和删除注册用户发布的评论。

这些功能的实现方法分别是 viewComments 和 deleteComments。评论删除流程图如图 3-22 所示：

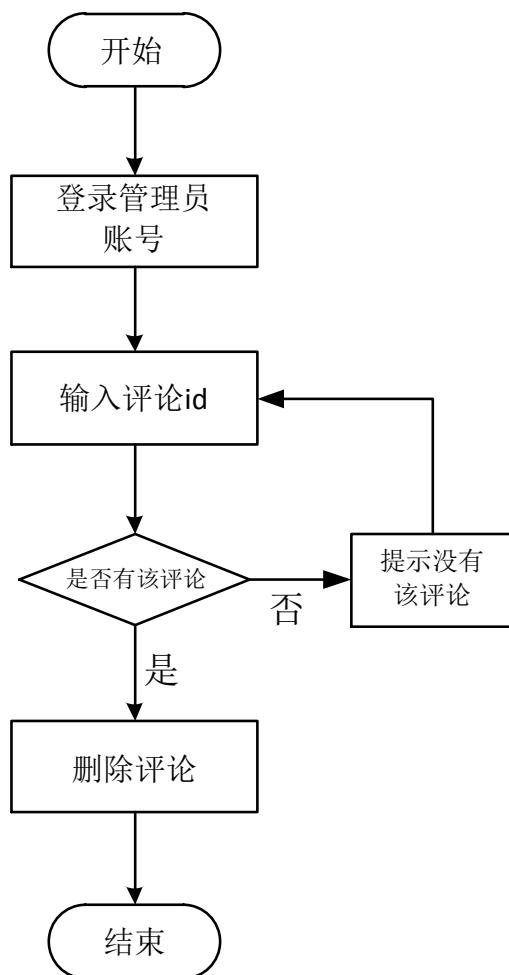


图 3-22 删 除评论流程图  
Figure 3-22 Delete comments flowchart

### 3.5.4 公告管理

- (1) 功能描述：实现对系统公告的增、删、改操作。
- (2) 输入：公告 id，标题，内容，时间
- (3) 输出：系统将公告信息保存至 MySQL 数据库中。
- (4) 实现方式：公告管理的主要功能是让管理员查看、发布、更新和删除系统公告。这四个功能的实现方法分别是 viewNotice, upNotice, updateNotice 和 downNotice。发布公告和删除公告的流程图分别如图 3-23 和图 3-24 所示。

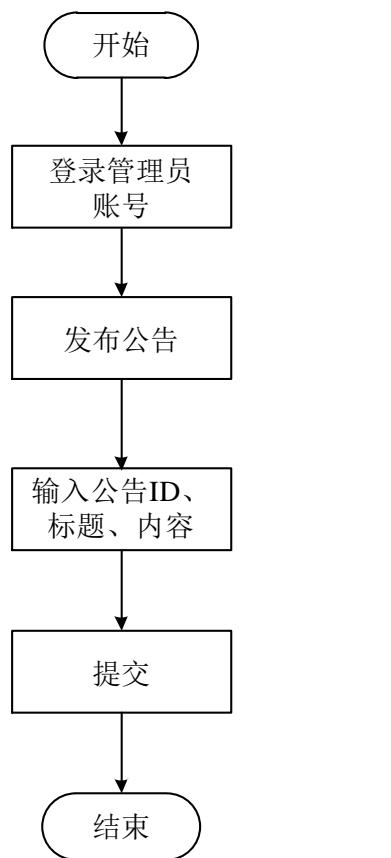


图 3-23 发布公告流程图

Figure 3-23 Announcement flow chart

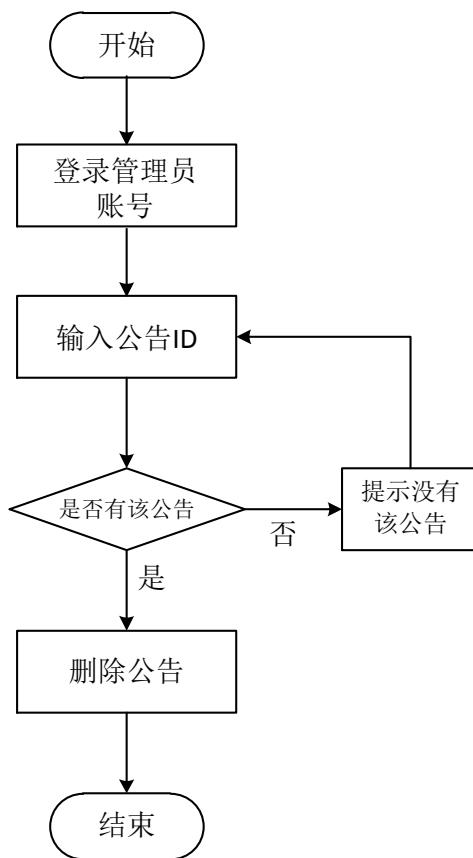


图 3-24 删除公告流程图

Figure 3-24 Delete announcement flowchart

### 3.6 本章小结

本章首先对系统总体架构和功能模块进行了概要设计，然后对数据库、用户端功能模块和服务端功能模块进行了详细设计，实现了对前台与后台主要功能模块的设计。

**总体架构设计：**对系统开发采用到的 Spring MVC 架构体系做出介绍；

**功能模块设计：**通过功能模块图对系统的整体功能进行设计，用功能流程图对用户端和服务端的功能模块进行设计及介绍功能模块的具体组成；

**数据库设计：**通过分析数据库需求、绘制实体图和 E-R 图（也称实体-联系图(Entity Relationship Diagram)，提供了表示实体类型、属性和联系的方法，用来描述现实世界的概念模型。），设计数据库表<sup>[29]</sup>。

## 第 4 章 系统实现

系统功能模块设计完成后，接下来要做的是各模块功能的实现，系统实现是将应用软件系统从逻辑概念上的设计转变为具体的可操作可使用的业务系统的过程。本章首先介绍了用户功能模块所包含的主要功能的实现，展示了实现模块功能的相关代码和功能模块的运行界面。接着介绍了管理功能模块所包含的主要功能的实现，展示了实现模块功能的相关代码和功能模块的运行界面。最后，介绍了手机 app 的开发机制和发布流程。

### 4.1 用户功能模块的实现

本移动订餐 app 系统的 Android 手机用户端通过服务器 IP 地址、端口号与服务器实现连接。

#### 4.1.1 用户注册登录模块

##### (1) 注册

###### 1) 注册流程

用户在注册账号的时候，首先进入注册界面，填写用户信息，包括用户名、密码和电话等，如果用户名已经被使用，那么将会提醒用户返回到注册页面，重新填写信息，如果未被使用，且两次密码输入一致即可注册成功，并且相关数据将会自动上传到数据库中<sup>[30]</sup>。

###### 2) 注册界面如下图所示：



图 4-1 注册界面

Figure 4-1 Registration interface

## (2) 登录

### 1) 登录流程

进入系统先看到的是首页，但只是游客身份，只能查看产品。如果想上传产品、收藏产品等必须先登录。注册成功的用户可以直接输入用户名和密码登录。

### 2) 登录界面



图 4-2 登录界面  
Figure 4-2 Login interface

### 4.1.2 用户订餐模块

#### 1) 订餐流程

注册用户输入用户名和密码登录系统后即可进入系统菜谱界面浏览、查询菜谱，选择好菜品后下单订餐，订餐成功提交后系统会打开支付界面，用户可选用以下多种付款方式，如支付宝、微信支付和全民付等，现最常用支付方式是支付宝和微信。本订餐系统用支付宝支付的时序如图 4-3 所示。

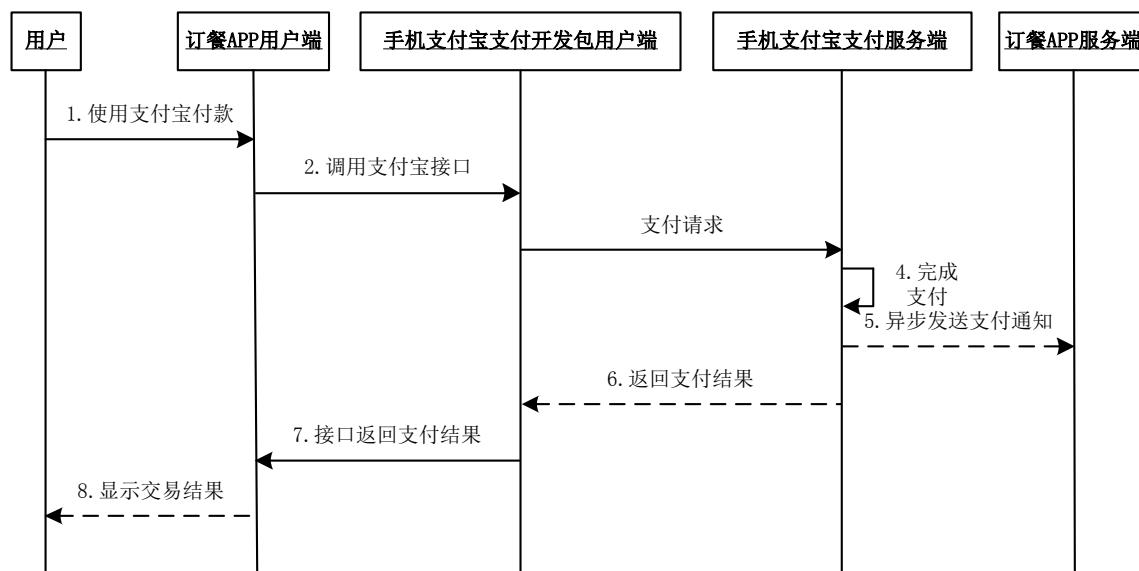


图 4-3 支付宝支付时序图  
Figure 4-3 Alipay payment sequence diagram

## 2) 付款界面

下图 4-4 是本订餐系统选用支付宝方式进行支付的付款界面。



图 4-4 付款界面  
Figure 4-4 Payment interface

#### 4.1.3 首页模块

用户登录成功后，转入首页。首页如图 4-5 所示。页面色彩搭配为经典的红白。首页中间主体的部分是菜谱列表。顶部是选择按钮，用户可以选择显示食谱列表或菜谱列表。下面是一个滚动的菜谱推荐模块。推荐内容为最新菜谱，用户可以直接点击滚动图片进入菜谱页面。用户可以按照自己的喜好选择食谱的分类查询菜谱，或者在搜索栏输入食谱的关键字来搜索。底部是整个应用的导航栏，可通过导航栏进入不同的模块，体验不同的功能<sup>[31]</sup>。

菜谱轮播推荐部分，将几张图片放在容器里。图片的样式默认为隐藏，第一张图片额外增加显示到前面的样式。设置定时器，更改图片的样式，实现自动轮播。



图 4-5 首页界面

Figure 4-5 Home page interface

首页上根据选择的“菜谱”或“食谱”来决定列表显示内容。具体是根据 listShouye(gtype)传过来的 gtype（是 GLib 运行时类型认证和管理系统）值，处理 AJAX 封装的请求。查询出数据，传到前端，并刷新首页列表。

```
function listShouye(gtype){
```

```
ajaxCallback("listFood",{bigtype:2,gtype:gtype},function(data)
    foodlist = data;
    ("foods").refreshShowListView(data);
});
```

查询数据代码如下：

```
function listFood(sid){
    ajaxCallback("listFood",{ sid:sid,bigtype:1 },function(data){
        foodlist = data;
        S(" foods").refreshShowListView(data);
    });
}
```

#### 4.1.4 发布食谱模块

用户点击底部导航栏的“发布食谱”，或首页右上角的“发布”即可进发布食谱的界面，如图 4-6 所示。

在编辑食谱时，与其他各类购物网站和 APP 相同，用户可以选择菜品分类（如小网购、海鲜、西餐、家常菜等），上传各步骤图片，添加文字步骤说明。食谱发布后会刷新首页的菜谱列表，同时将信息保存在数据库中。

信息分类，分为菜谱和食谱。用户可以自主选择。食谱的发布可以不断添加步骤和详细说明。发布的菜谱，通常是一张图片，配些简短说明，类似于朋友圈中的动态分享。



图 4-6 发布食谱界面

Figure 4-6 Publish recipe interface

发布食谱后的一系列运转：

```
<a href="#" onclick="saveFoodItem();class="ui-btn ui-btn-c">提交</a>
function saveFoodItem(){//发布食谱后,返回到首页,更新食谱列表
var fdata = serializeObject($("#fooditemform"));
fdata.gid = focusobj.id;
ajaxFormUploadFile(function(r){
    fdata.img = r;
    ajaxCallback("saveFooditem",fdata,function(data){
        showLoadert("操作成功!",true);
        $("#fooditemform")[0].reset();
        $("#gitemimg").attr("src","");
    });
});
```

```

});  

});  

toShouye();

```

点击“提交”后，通过 `serializeObject()` 将菜谱信息序列化存储在数组中，并通过处理 AJAX 异步请求刷新列表，返回首页。

#### 4.1.5 浏览食谱模块

在首页食谱列表中选择一个食谱，即可浏览详细信息。如图 4-7。在菜谱的末处，用户可以对菜谱点赞，收藏菜谱、评价，如图 4-8。在这个模块中，用户可以先浏览菜谱，根据下方的评论来判断菜谱的好坏和难易程度。若是因材料准备不够等原因无法立刻按照菜谱烹制，可以收藏菜谱，以备日后查看。收藏的菜谱可以在个人中心-我的收藏中查看<sup>[32]</sup>。

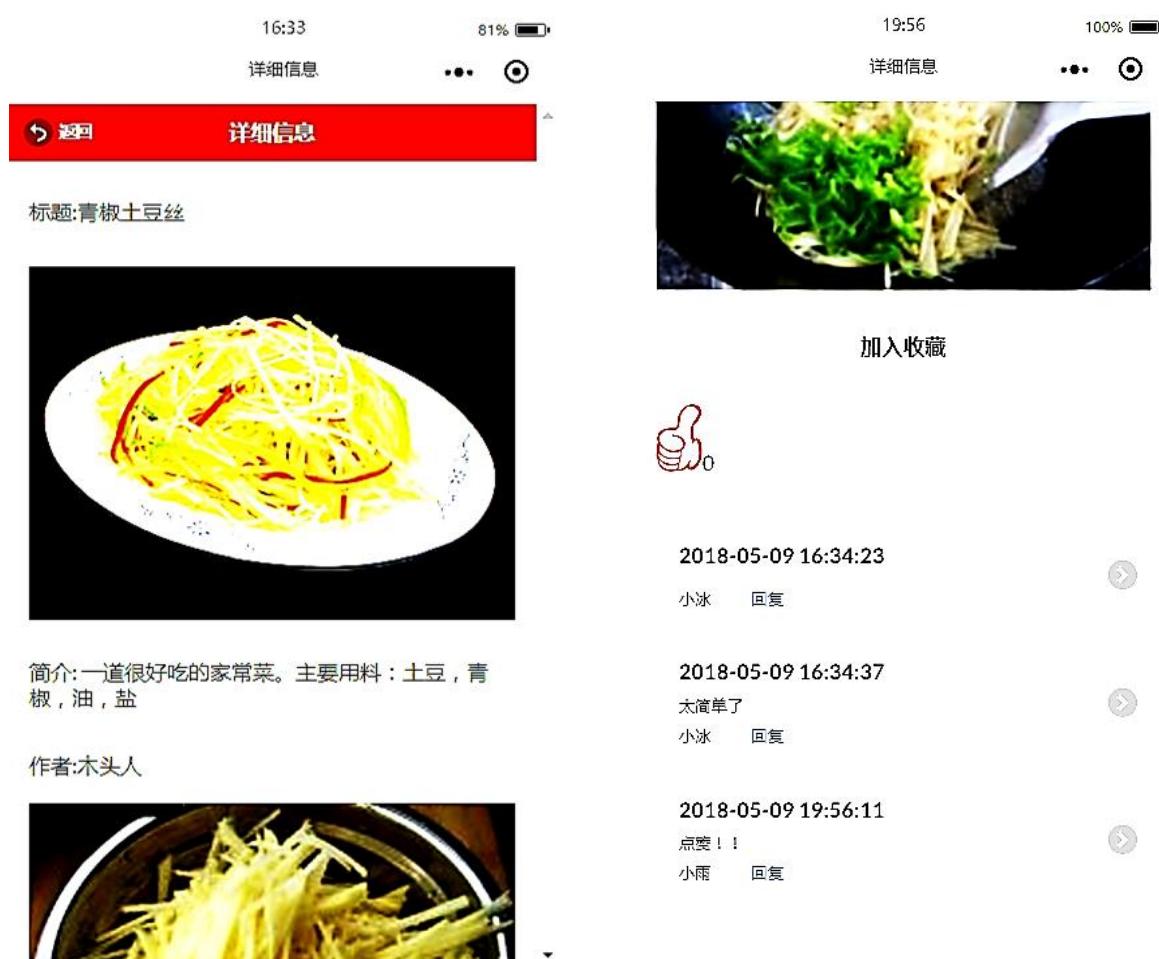


图 4-7 浏览食谱界面

Figure 4-7 Browse recipe interface

图 4-8 评论、点赞界面

figure 4-8 Comment and like interface

#### 4.1.6 查看用户信息和关注模块

在浏览菜谱时，点击作者名可以跳转到该用户的个人基本信息页面。如图 4-9 所示。在这里除了可以看到用户的基本信息外，还可以查看他发布过的菜谱抑或是关注该用户。关注列表如下图 4-10 所示，另外，还可以和关注的用户线上交流。在关注列表界面可以查找用户实现交友功能<sup>[33]</sup>。

在关注用户列表中，可以选择用户进行一对一交流，有点类似微博的私信功能，这也是系统为用户设计的比较私密的交流空间。

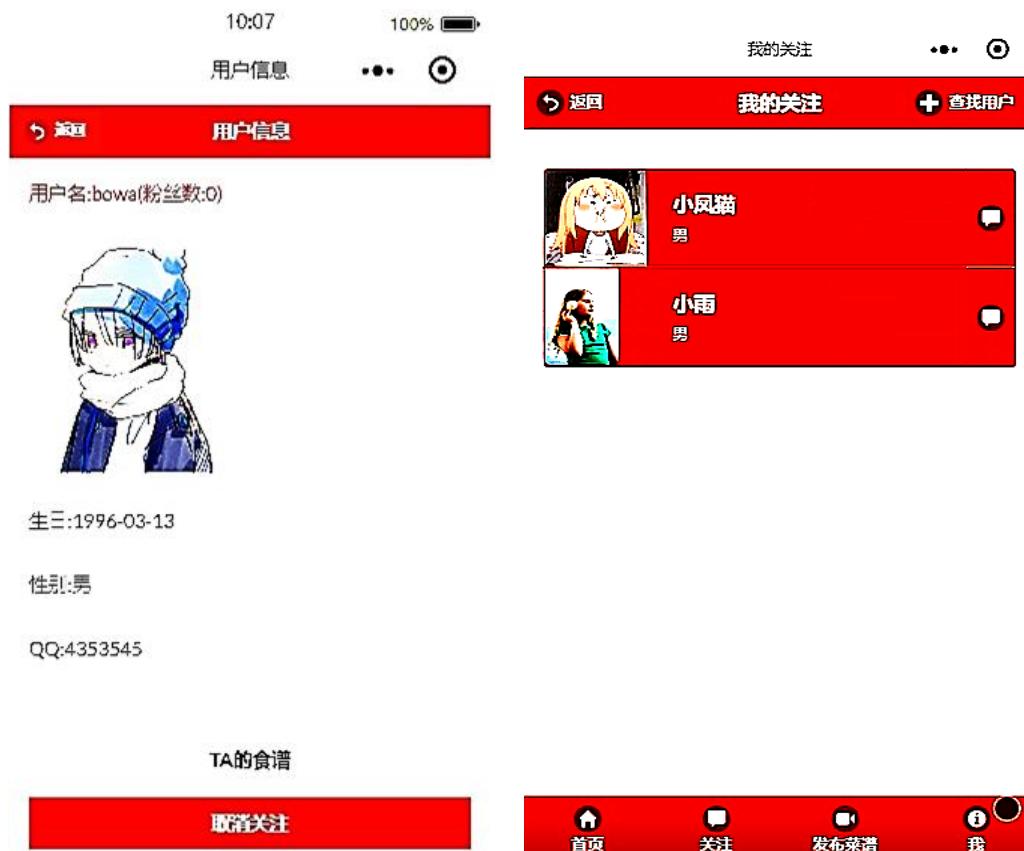


图 4-9 用户信息界面  
Figure 4-9 User information interface

图 4-10 关注列表界面  
Figure 4-10 Follow list interface

关注页面的跳转过程：

点击导航栏中的“关注”，切换到关注页面 friendpage。接着调用 listMyFriend()方法，对 ajaxCallback 封装请求进行处理<sup>[34]</sup>。先是将用户的“id”命名为“uid”，然后链接数据库中的 user 表，根据“uid”查出一条数据信息。用户名合法，则在这条数据中找出 fids

字段，fid 字段中存储的是关注用户的“id”。根据字段中的“id”显示出关注用户的数据，并以 json 的形式输出给前台，刷新关注列表的同时传入数据。代码如下：

```

function listMyFriendo(){
    ajaxCallback("listMyFriend",{uid:userinfo.id},function(data){
        focuslist = data;
        $("#myfriendlist").refreshShowListView(data); //刷新列表,并将数据传进去
        setTimeout();
        $("#friendpage").fadeIn(2000);
        console.log("显示");
    });
}

public function listMyFriend(){
    $uid=$_REQUEST['uid'];
    $model=M("User");//连接数据库中的 user 表
    $user=$model->find($uid); //查出一条信息
    if($user){
        $fids=empty($user['fids'])?":$user['fids'];
        $db=M('User');
        $list=$db->query("select * from wct_user where id in($fids)");
        if($list== false){
            $list=array();
        }
        $this->renderJsonpObj($list); //以 Json 形式输出给前台
    }
}

```

查询数据库中数据的操作代码如下：

```
public function find($options=array()){
```

```
if(is_numeric($options) || is_string($options)){
    $where[$this->getPk()] = $options;
    $options = array();
    $options['where'] = $where;
}

$options['limit'] = 1;//总是查找一条记录

//分析表达式

$options = $this->_parseOptions($options);

$resultset = $this->db->select(($options);

if(false === $resultset){

    return false;
}

if(empty($resultSet){//查询结果为空

    return null;
}

$this->data = $rsultSet[0];

$this-> after find($this->data,$options);

if(!empty($this->options('result')){

    return $this->returnResult($this->data,$this->options['result']);
}

return $this->data;
}
```

#### 4.1.7 用户聊天界面

关注用户后，可以进行一对一的聊天，界面如图 4-11 所示：



图 4-11 聊天界面

Figure 4-11 Chat interface

用户之间信息发送的实现：

通过 addMessage()方法，将信息和时间插入到数据库的 wct\_message 表中。

```
public function addMesage(){
    $_REQUEST['ndate']=date('Y-m-d H:i:s');
    $model=M("Message");
    $model->add($_REQUEST);
    $this->renderJsonpObj($_REQUEST);
}
```

用户端通过 refreshServerChatList()刷新对话列表。其中列表的刷新部分由 refreshChatList(data)实现。通过设置全局变量 \_oldlength 记录数组长度，每次刷新 \_oldlength 都会记录上一次的长度，两次长度不等时，实现列表的刷新。

```
function refreshServerChatList(){
    var queryobj=fuid:userinfo.id,fid:focusobj.id
    var ajaxurl="listMyMessage";
```

```
ajaxCallback.ajaxurl,queryobj,function(data){  
    refreshChatList(data);  
},true);  
}
```

#### 4.1.8 查看公告界面

用户可以在底部导航栏选择“我”，然后查看系统的公告内容，如图 4-12。用户可以在搜索框输入关键字查询相关公告。在公告列表中，选择一条，点击可以进入详细页面。个人中心除可以查看公告信息外，还可以修改自己的个人信息和密码，查看个人收藏的产品信息。



图 4-12 公告查看界面  
Figure 4-12 Announcement view interface

#### 4.2 管理员功能模块的实现

##### 4.2.1 用户管理界面

用户管理界面如图 4-13 所示。进入用户管理界面中选择用户列表，可以看到最上方

为用户名的搜索框。管理员可以在输入框输入用户名进行模糊查询，列表中会筛选出相应符合条件的信息。“重置”按钮则是用来清空搜索框中的内容。列表的最上面一栏有三个选项，分别是新增、修改和删除。以上的三个选项分别对应着管理员对用户信息的三种操作<sup>[35]</sup>。

新增和修改业务逻辑相似，下面以修改用户信息为例进行说明：

点击“新增”按钮，会弹出“用户管理”的页面。如图 4-14 所示。用户名不可为空，权限分为普通用户和管理员。用户表中，用户权限字段管理员为 1，普通用户的为 2。状态分为正常和封号两种。若被封号，则会失去体验某些功能的权利。密码和确认密码需输入一致，否则会报错。

删除操作相对简单。管理员只需要选择一条记录，点击“删除”按钮即可删除该条记录。执行操作前，系统会弹出确认窗口供管理员确认。

管理员列表									
		Id	用户名	电话	生日	QQ	介绍	密码	用户权限
1	18	张三						12345qwert	2
2	17	婷婷						qwer123	2
3	16	小冰	12345678911	1998-03-31	1234567891			123456	2
4	15	小美						111111	2
5	14	小雨						123456	2
6	13	bozai2	15123385885					111111	2
7	12	xiaowang	15123385885	1996-12-15				111111	2
8	11	ideabobo	15123385885	1995-04-22	545488484			123456	2
9	10	bowa	15123385885	1996-03-13	4353545			111111	2

图 4-13 用户管理界面图  
Figure 4-13 User management interface

用户名：	小波仔
权限：	普通用户
状态：	正常
密码：	<span style="color: red;">!</span>
密码确认：	<span style="color: red;">!</span>
<input type="button" value="保存"/> <input type="button" value="取消"/>	

图 4-14 信息修改界面  
Figure 4-14 Information modification interface

删除用户的功能实现：

首先，在 Spring 的配置文件 beans.xml 中，使用<bean>节点创建实例，  
id=sessionFactory。代码如下：

```
<bean id="sessionFactory">  
    class=org.springframework.orm.hibernate3.LocalSessionFactoryBean">  
        <property name="dataSource" ref="datasource"/>  
        <property name="mappingResources">  
            <list>  
                <value>com/ideabobo/model/Bill.hbm.xml</value>  
                <value>com/ideabobo/model/Dingzuo.hbm.xml</value>  
                <value>com/ideabobo/model/Good.hbm.xml</value>  
                <value>com/ideabobo/model/GoodItem.hbm.xml</value>  
                <value>com/ideabobo/model/User.hbm.xml</value>  
                <value>com/ideabobo/model/Type.hbm.xml</value>  
                <value>com/ideabobo/model/Vip.hbm.xml</value>  
                <value com=ideabobo/model/Shop.hbm.xml</value>  
                <value>com/ideabobo/model/Posts.hbm.xml</value>  
                <value>com/ideabobo/model/Replay.hbm.xml</value>  
                <value>com/ideabobo/model/Notice.hbm.xml</value>  
                <value>com/ideabobo/model/Message.hbm.xml</value>  
            </list>  
        </property>
```

界面发送用户删除请求， structs.xml 会拦截 post 请求，并将请求提交给 UserAction 进行业务逻辑处理。

```
function deleteItem(uuid){  
    $.post(<%=_APP_%>/User!deleteItem',(id:uuid),function(data){  
        closeFlush();  
    });
```

```
}
```

```
<action name="User" class="com.ideabobo.action.Useraction">
```

在逻辑处理类中，需要调用 userService 的 delete()方法根据 “id” 对数据进行删除。而 userService 对 delete()方法的实现是调用 Hibernate 根据字段 “uid” 到数据库中查询删除。最后会根据结果直接返回界面。代码如下：

```
public void deleteItem(){  
    String id=request.getParameter("id");  
    userService.delete(Integer.parseInt(id));  
    render("操作成功");  
  
    public void delete(Integer uid){  
        sessionFactory.getCurrentSession().delete(  
            sessionFactory.getCurrentSession().load(User.class,uid));  
    }  
}
```

Hibernate 对数据库的处理：

在 Spring 配置文件中可以找到 User.hbm.xml 是 Hibernate 的模型配置。他会将 User.java 这个模型类自动关联到数据库表 wtc\_user。User.java 里的每一个属性都对应着数据库表中的每一个字段。在 User.hbm.xml 文件中，将本地模型 property 对应表中的 column，实现 Hibernate 的本地映射与数据库的持久化。

```
<hibernate-mapping>  
    <class name="com.ideabobo.model.User" table="wtc_user">  
        <id name="id" type="java.lang.Integer">  
            <column name="id"/>  
            <generator class="identity"/>  
        </id>  
        <property name="username" type="java.lang.String">  
            <column name="username" length="50"/>  
        </property>  
        <property name="passwd" type="java.lang.String">
```

```

<column name="passwd" length=50"/>
</property>
.....
</property>
</class>
</hibernate-mapping>

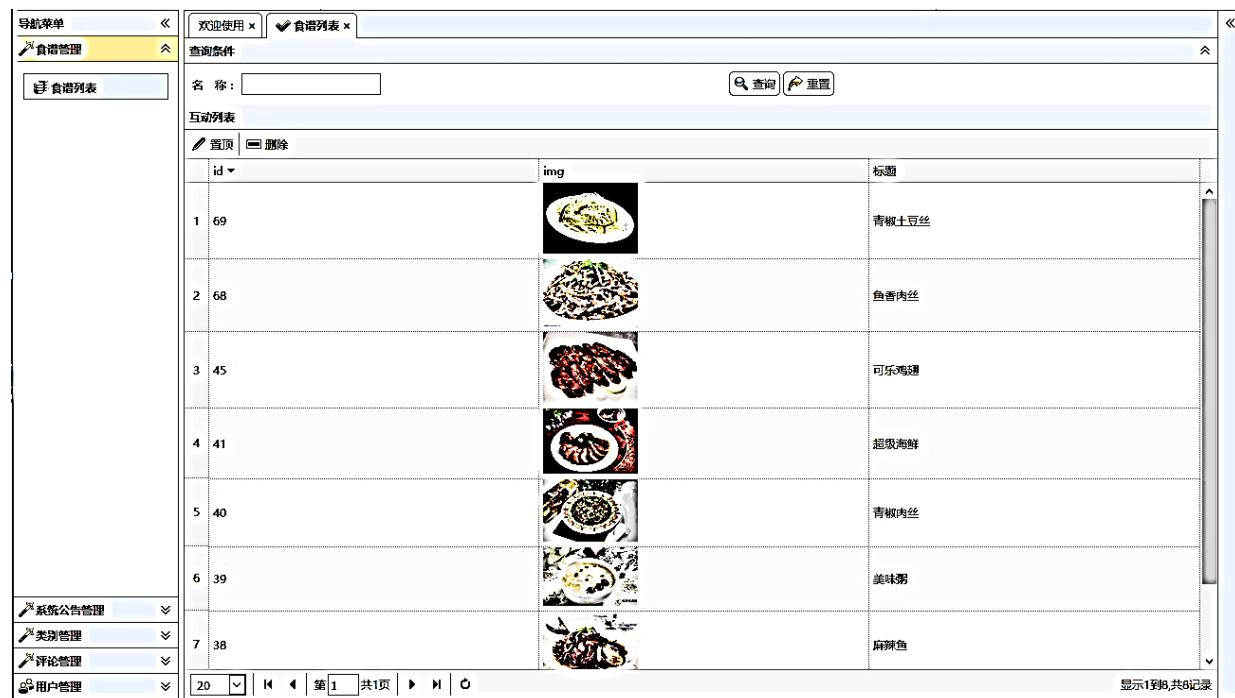
```

其他功能实现与此功能类似，下面不作赘述。

#### 4.2.2 菜谱管理界面

管理员可以增加、查看、删除菜谱信息，其界面展示如图 4-15 所示。菜谱管理界面与用户管理界面大致相同<sup>[36]</sup>。只是在菜谱管理的列表中，通过 img 的字段来存储菜谱的封面图片，使菜谱信息更容易被识别，使界面更加美观。这样使列表不至于因全是文字而太过枯燥乏味，也缓解了管理员的视觉疲劳<sup>[37]</sup>。

以删除菜谱为例，对其业务逻辑进行介绍。管理员点击“删除”，系统会首先判断是否已经选择一条记录，若没有，则提示“请选中一条记录进行删除！”。若已选中，则弹出确认窗口。确定后，执行 deleteItem()方法，structs.xml 拦截器根据请求去后台 servlet 寻找 action 处理，调用 Hibernate 模板进行数据库的更新。



The screenshot shows a web-based menu management system. At the top, there's a navigation bar with links like '系统公告管理', '类别管理', '评论管理', '用户管理', and '食谱管理'. Below the navigation is a search bar with fields for '名称' (Name) and buttons for '查询' (Search) and '重置' (Reset). The main area is titled '互动列表' (Interactive List) and contains a table with the following data:

ID	img	标题
1 69		青椒土豆丝
2 68		鱼香肉丝
3 45		可乐鸡翅
4 41		超级海鲜
5 40		青椒肉丝
6 39		美味粥
7 38		麻辣鱼

At the bottom of the interface, there are pagination controls (20, <, <<, 第 1, 共 1 页, >, >>, >) and a note indicating '显示1到8共8记录' (Displaying 1 to 8 of 8 records).

图 4-15 菜谱管理界面图  
Figure 4-15 Menu management interface

#### 4.2.3 公告管理界面

公告信息管理界面如图 4-16 所示。管理员可以查询历史系统公告、对公告信息进行修改抑或是发布新的系统公告。在查询公告时，可以在搜索框中输入公告名称，点击“查询”按钮，则列表中显示相应记录；点击“重置”按钮，则清空搜索框。管理员可以选择一条或多条系统公告进行删除操作<sup>[38]</sup>。



图 4-16 公告管理界面图

Figure 4-16 Announcement management interface

#### 4.2.4 类别管理界面

类别管理界面如图 4-17 所示。管理员打开类别管理列表后，可以在搜索框输入要操作的菜谱类别名，列表界面将会显示通过模糊查询得到的相关信息，以便于管理员或者超级用户对数据进行操作处理，如添加、删除、修改分类等。

菜谱分类的添加：点击“新增”按钮，屏幕中间会弹出操作的界面。鼠标移到输入框时，会有“该项为必输项”的提示。保存后，会成功添加到分类列表。若选择取消，则会弹出对话框确认是否取消。

菜谱分类的删除和修改：管理员必须先选中一条记录才能执行删除或修改操作。若直接点击“删除”或“修改”按钮，系统会弹出提示框，提示管理员选择一条记录。

管理员对菜谱分类的一系列管理可以在发布菜谱时得到较好的优化。选择也更加多样。在查询菜谱时，菜谱也被更好地归纳分类。

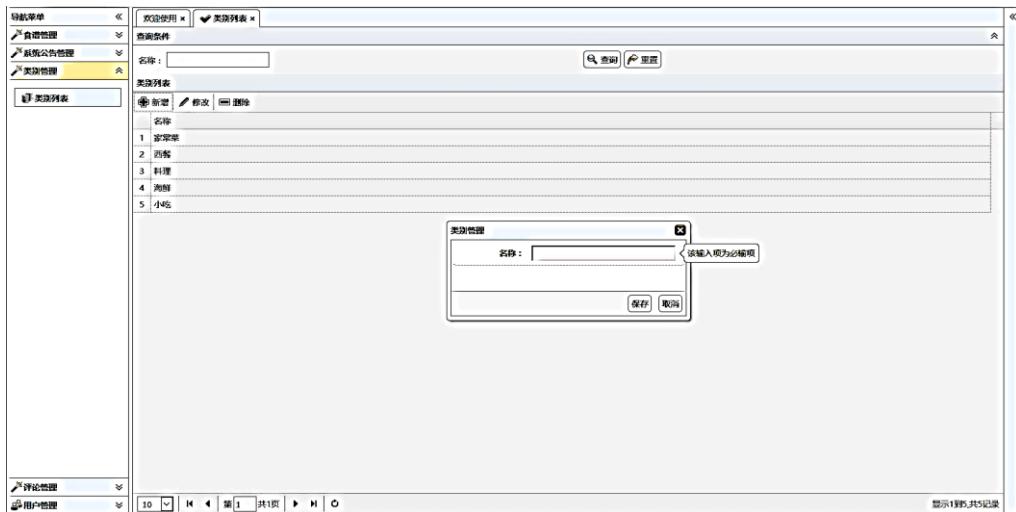


图 4-17 类别管理界面图

Figure 4-17 Category management interface

### 4.3 手机 app 的实现和发布

本订餐交易类手机 app 前端主要采用 JavaScript、HTML5 进行开发，JavaScript（简称“JS”）是一种具有函数优先的轻量级，解释型或即时编译型的高级编程语言。虽然它是作为开发 Web 页面的脚本语言而出名的，但是它也被用到了很多非浏览器环境中，JavaScript 基于原型编程、多范式的动态脚本语言，并且支持面向对象、命令式和声明式（如函数式编程）风格。本订餐 app 前端使用 JavaScript 进行开发后，使用 web-view 组件进行调用，如下图 4-18 所示：



图 4-18 内嵌网页

Figure 4-18 Embedded web page

前端的数据处理和后台一样，请求都会被 structs 拦截，对数据库做出数据处理后将结果返回到前端。

手机 app 的发布流程如下所述：

- (1) 申请账号，获取 APPID。首先在官网上注册一个属于自己的 app 账号，获得 APPID；
- (2) 根据需求，编码实现功能，编译预览效果；

(3) 在开发者工具的顶部工具栏有一个预览按钮，点击后，开发工具会自动将当前项目打包，将程序代码上传到手机的服务器。上传成功后，手机用户可以扫描界面上的二维码真实预览；

(4) 代码上传后需要提交审核，发布 app。

#### 4.4 本章小结

本章主要介绍的是系统功能模块的实现，并展示了用户端和后台服务端的主要界面。对于用户端模块，除展示页面外，还对流程做出分析。对于几个主要功能实现的过程以及操作流程进行了介绍，对实现的方法进行了分析。对于管理端模块，主要是展示管理界面并对相关的业务逻辑详细介绍。最后就 app 的开发和发布做了简要概述。着重介绍了用户端具有交互、聊天作用的关注功能的实现。本系统功能界面友好，操作简洁明了。

## 第 5 章 系统测试

软件测试在开发过程中是不可或缺的环节，不仅能验证功能是否达到预期目标，还能及时发现开发的系统所存在的问题和漏洞，以便及时改进不足之处，最终达到预期的理想效果。总的来说就是一个查漏补缺的过程<sup>[39]</sup>。本章将对需求分析里提到的主要功能进行测试，目的是确认本订餐管理系统是否实现了需求分析提到的功能和要求。

本章主要是对移动订餐 APP 系统的功能和性能进行测评，测试的部署的软硬件测试环境如下表 5-1 所示：

表 5-1 软硬件测试环境  
Table 5-1 Software and hardware test environment

语言	Java
数据库	MySQL
服务器	Apache、Tomcat
软件	Android 模拟器, Tomcat, JDK, MySQL 等
硬件设备	笔记本电脑 (ASUS i7, windows 10 系统)

### 5.1 测试方法

本系统在测试中所采用的测试方法是黑盒测试，黑盒测试又称为数据驱动测试，这种方法是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明，它是站在使用软件或程序的角度，从输入数据与输出数据的对应关系出发进行的测试。该类测试注重于测试软件的功能性需求<sup>[40]</sup>，主要针对软件界面和软件功能进行测试<sup>[41]</sup>。

软件测试按测试过程一般分为三个步骤进，即单元测试 (Unit Testing)、集成测试 (Integrated Testing) 和系统测试 (System Testing)。单元测试又称模块测试，是针对软件设计的最小单位——程序模块，进行正确性检验的测试工作。其目的在于发现各模块内部可能存在的各种差错。集成测试是把已测试过的模块组装起来，主要对与设计相关的软件体系结构的构造进行测试。系统测试则是当应用作为整体运行时的测试执行阶段，本章介绍的测试即为此类测试。

## 5.2 系统主要功能测试

本节主要是对系统部分的具体功能进行测试，保证订餐交易类手机 app 的核心功能可以正常实现。测试的功能包括登录与登出、食谱查询、订餐、菜谱分类、管理权限、密码修改等功能。以下是对菜谱进行浏览、发布、评论与收藏进行测试预期实现功能的具体描述：

(1) 登录后，跳转到首页，可以看到首页的菜谱列表、轮播推荐、顶部搜索栏和底部导航栏。可以上下滑动浏览菜谱列表，点击菜谱可以进入菜谱的详细页面，点击返回，可以正常跳回首页。

(2) 对于发布食谱模块。点击发布食谱，跳转到编辑页面，填写菜谱名后，可以添加食谱的封面以及步骤说明。提交后，页面跳回首页，菜谱列表刷新。

(3) 浏览菜谱时，在下方能够正常点赞，评论。收藏菜谱后，可以在我的收藏列表中查看。

### 5.2.1 登录与登出

本测试是为了测试用户是否能够正常登录与登出。测试消费者、管理员两种用户在用户名、密码输入正确或错误、登陆失败次数超过 3 次等情况下系统的状态。经测试，确定本手机订餐 app 在安全登录登出上尚未发现问题，能够满足要求。下表是登录登出测试用例及其结果：

表 5-2 用户登录登出测试用例  
Table 5-2 User login and logout test cases

编号	用例说明	检查点	期望输出	结果
L1	输入的用户名、密码都对	用户登录功能	登陆成功，跳转到订餐系统首页。	符合预期
L2	输入用户名对、密码错	系统登录的安全性	提示用户名与密码不匹配，输入错三次后帐号被锁定，24 小时内无法登录。	符合预期
L3	输入用户名错、密码对	系统登录的安全性。防止黑客破解用户密码	提示用户名与密码不匹配，输入错三次后帐号被锁定，24 小时内无法登录。	符合预期
L4	退出系统	系统登出的安全性	弹窗提示已安全退出系统。	符合预期

### 5.2.2 食谱查询功能测试

本测试的目的是测试用户是否能够正常查询食谱。食谱查询功能测试用例如表 5-3 所示：

表 5-3 食谱查询功能测试用例表  
Table 5-3 Recipe query function test case table

说明	测试食谱查询功能步骤及预期输出	
步骤	输入数据	预期输出
1	“青椒牛肉”	显示“青椒牛肉”菜谱
2	“牛”	显示所有含“牛”字的菜谱
3	输入错误名称	界面显示为空

### 5.2.3 订餐功能测试

主要测试的是订餐系统的重要功能：订餐功能。经测试，确认本订餐 app 能够满足消费者在订餐功能上的需求。其测试用例及结果如下表：

表 5-4 订餐功能测试用例  
Table 5-4 Ordering function test cases

编号	用例说明	检查点	期望输出	结果
Ord1	选中两样菜谱	菜谱加入购物车	购物车中加入被选菜谱	符合预期
Ord2	输入“麻辣豆腐”后点击“搜索”	搜索菜谱功能	显示包含“麻辣”两字的菜谱	符合预期
Ord3	对购物车中的菜谱进行结算	结算收款功能	跳转至订单结算页面，所结算的菜谱信息显示在屏幕上	符合预期
Ord4	订单填写完整并提交	订单管理功能	客户端显示订单详情，管理员端收到订单提醒信息	符合预期

### 5.2.4 菜谱分类功能测试

本测试检查的是本手机订餐交易管理系统中的菜谱分类管理功能，测试管理员在对菜品进行分类删改操作时系统的状态。经测试，系统能正确反馈信息，符合测试预期。其测试用例及结果如表 5-5 所示：

表 5-5 管理员管理菜谱分类测试用例  
Table 5-5 Administrator managed product classification test cases

用例编号	测试用例	预计结果	实际结果	测试结果
Test 01	选择一项餐品类别后点“删除”按钮	所选分类被删除	用户端的菜谱分类列表中没有已删除的类别	符合预期
Test 02	点击“新增”按钮后根据提示输入信息	成功添加分类	成功添加分类，用户端的分类列表被刷新	符合预期
Test 03	选一条分类记录后点“修改”按钮，填改信息	成功修改分类信息	用户端列表分类名被修改	符合预期

### 5.2.5 管理权限功能测试

本测试是检查此订餐系统的管理权限功能，测试在以管理员和非管理员身份登录时系统的反馈状态。经测试，测试结果符合预期。其测试用例及结果如表 5-6 所示：

表 5-6 管理权限测试用例  
Table 5-6 Manage permission test cases

编号	用例说明	检查点	期望输出	结果
Test01	非管理员登陆	管理权限控制	不能查看商户信息与其他用户信息	符合预期
Test02	管理员登陆	管理权限控制	可以查看商家信息与其他用户信息	符合预期

### 5.2.6 修改密码功能测试

本测试是为了测试用户进行密码修改操作是否正常。用户修改密码测试用例如下表 5-7 所示：

表 5-7 用户修改密码测试用例  
Table 5-7 Test cases of user modifying password

用例编号	测试用例	预计结果	实际结果	测试结果
Test 01	密码输入错误	修改失败	提示密码错误	符合预期
Test 02	密码首次输对，二次输错	修改失败	提示两次密码不一致	符合预期
Test 03	密码首次输错，二次输错	修改失败	提示密码错误	符合预期
Test 03	密码首次输对，二次输错	修改成功	提示密码修改成功	符合预期

### 5.3 系统性能测试

本订餐 app 系统开发完成后在测试机中进行了测试, 测试预期目标是系统能够支持 100 人同时在线。该测试采用测试工具 LoadRunner (是一种预测系统行为和性能的负载测试工具) 进行自动化测试, 以 100 人同时登录访问系统且每秒请求次数为 100 作为测试条件, 所得测试结果如表 5-8 所示。从测试结果来看, 本系统在该测试条件下能够承受 100 人同时在线访问, 具备较好的抗压能力。

(1) 系统整体运行速度的测试: 通过不同品牌的手机, 同品牌但是配置不同的手机, 同时打开运行、使用、关闭等一系列操作反复测试, 看看是否有卡滞、死机、不灵等状况的出现。

(2) 整体响应时间的测试: 根据过程可将响应时间分为: 打开响应时间、操作反应时间、和关闭响应时间。因此需要在不同的手机上测试记录相关参数的数据, 并进行整理分析和改进。除此之外, 按照不同过程的数据具体来分, 又可以分为: 平均响应时间, 最大和最小响应时间, 平均响应时间是整机 APP 运行时的流畅度, 最大响应时间关系到客户的容忍极限, 在这方面需要多做调查, 逐步优化, 以达到客户满意的结果。

平均事务响应时间图直接反映系统的性能情况, 本订餐系统在测试机上设定在一分钟内以每秒发送 15 至 20 个请求来进行测试, 检查 5 分钟内系统持续在此种测试状态下的响应情况。测试结果表明服务器在此种测试状态下可对所有的请求做出响应, 系统运行正常, 这表明本订餐系统稳定性良好。Web 服务器的平均事务响应时间如图 5-2 所示。

(3) 系统稳定性的测试: 运行时的稳定性极为重要, 稳定性的好坏关系着软件使用的可靠与否, 是客户比较在意的一个评估点, 所以也要着重处理<sup>[42]</sup>。

上述系统性能的测试用例及结果如下表所示:

表 5-8 系统稳定性测试用例  
Table 5-8 System stability test cases

编号	用例说明	检查点	期望输出	结果
Per1	网络良好, 登录 1 个用户	响应速度	2 秒内进入系统	符合预期
Per2	网络良好, 登录 100 个用户	并发性	5 秒内进入系统	符合预期
Per3	网络良好, 100 个用户同时提交订单	并发性	5 分钟内全部订单发送至管理员(商户), 经管理员处理后发送反馈信息给用户	符合预期
Per4	网络良好, 1 个用户多次点击“提交”按钮提交订单	稳定性	只提交订单一次, 付一次款	符合预期

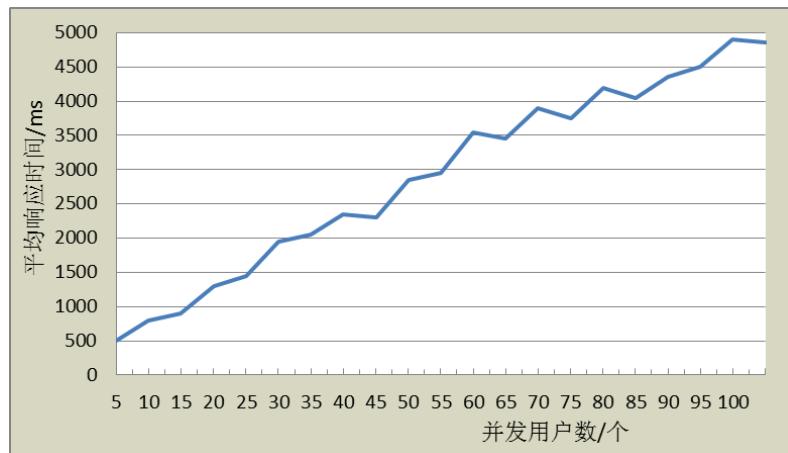


图 5-1 系统压力测试结果  
Figure 5-1 System pressure test results

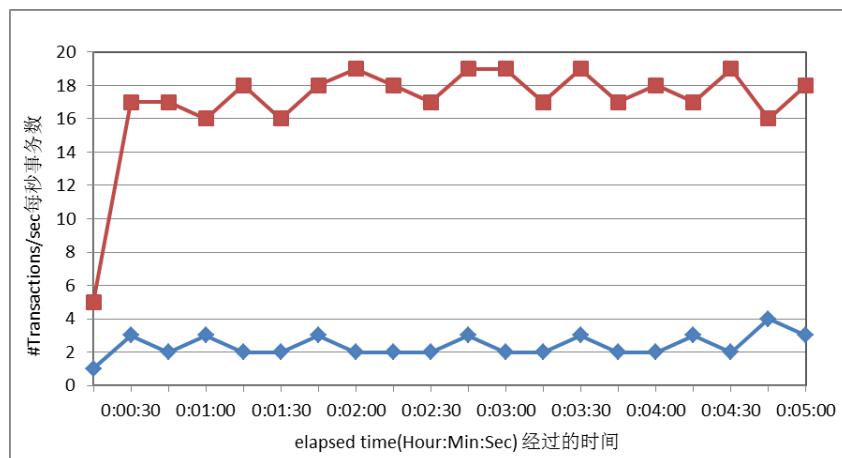


图 5-2 平均事务响应时间图  
Figure 5-2 Average transaction response time graph

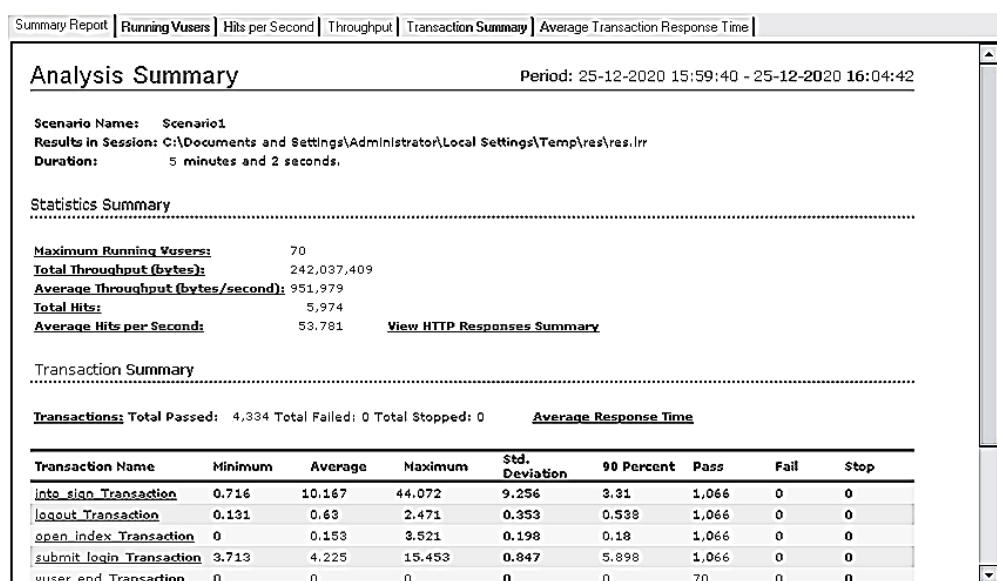


图 5-3 Web 服务器性能测试图  
Figure 5-3 Web server performance test chart

## 5. 4 本章小结

测试在系统的开发设计过程中不可或缺。本章对手机订餐交易管理系统的各个模块进行了测试用例的编写和测试，测试用例通过表格的形式表现。另外对系统的功能和性能进行了部分测试。测试表明，本手机订餐交易管理系统在系统性能、健壮性和安全性方面均符合实际应用的需求，能够为消费者提供订餐便利，并可以在多用户的情况下稳定运行。测试结果也反映出系统中存在的一些问题和不足，往后将根据测试的结果对系统功能进行调整，使系统运行的更加顺畅，优化用户体验效果<sup>[43]</sup>。

## 第 6 章 结束语

### 6.1 工作总结

本手机订餐交易管理系统是基于安卓平台开发的线上订餐 app 系统，本订餐系统采用 SSM 框架进行开发，既能满足软件的解耦，也有利于系统后期的功能扩展与维护。从测试运行情况来看，该系统的主要功能已经基本实现，达到了预期目标。本订餐系统分为用户端和服务端两大部分，分别对应系统的前台与后台。系统的前台功能主要有个人信息管理、注册登录、查看菜谱、订购菜谱、发布菜谱、关注和聊天等；后台功能则主要包括用户管理、菜谱分类管理、菜谱发布管理、评论管理、系统公告管理等。

本系统在测试过程中运行平稳，系统稳定性和强壮性较好，具有较强的实用性，能够提供便捷的订餐功能。但由于时间不足、水平有限等的原因，致使本系统现仍存在不少需要改进之处，今后我会继续努力学习相关开发知识，力争使所开发的系统日加完善。

### 6.2 后续工作展望

系统开发虽然已经完成，但由于自身能力的不足、开发经验的有限，有些功能还需要后续完善。在此，对后续工作提出一些展望：

(1) 引入一些智能化的功能，如可以添加拍照识别食材并推荐相关菜谱的功能。首页上的滚动推荐部分，功能主要是为用户推荐最新的菜谱。考虑推荐部分研究以下几种实现方案：

- 1) 通过对每个菜谱的评论量做出统计，评论数量 Top10 的菜谱可以被推荐，形成热门菜谱推荐；还可以根据时令的不同、个人喜好的不同推荐菜谱；
- 2) 通过余弦相似性来计算文本的相似性，提取特征值，计算菜谱的相似度，为用户提供个性化的菜谱推荐；
- 3) 通过深度学习来做菜谱的个性化推荐。

(2) 在功能方面，考虑做以下扩展：

1) 本系统的查询功能依旧是传统的手动输入查询，可以借助如科大讯飞语音开放平台等来实现语音识别，这样用户可以解放双手和键盘，通过语音来输入想要查询的内容。在浏览菜谱页面模块，本系统仅仅是简单的实现了浏览图片和文字。在实际生活中，用户在依照菜谱学习订餐交易时往往是不方便用手去滑动手机页面的。后期可以在步骤的

文字描述旁新添一个语音按钮，通过语音合成技术，获取步骤说明的文字内容并转换成语音。如此一来，用户就可以根据语音步骤提示进行订餐交易。语音合成的实现将会使本系统的操作更加智能化、人性化。

2) 可以考虑添加上传视频菜谱等功能，使系统的功能更加多元化，实现一个界面布局更符合用户操作习惯、能够根据浏览习惯进行个性化菜谱推荐的订餐 app 系统。

## 参考文献

- [1] 刘玉佳. 手机“app”开发的系统实现及前景分析 [J]. 信息通信, 2017. (1):260-261.
- [2] Balabanovic M, Shoham Y. Fab. Content-based, Collaborative Recommendation[J]. Communication of the ACM, 1997, 40. (3): 66-72.
- [3] Thomas Erl. Cloud Computing, Concepts Technology and Architecture. 北京:机械工业出版社. 2014.
- [4] 张建忠. 基于 WEB 的客户关系管理系统的分析、设计与实现 [D]. 济南:山东大学, 2012.
- [5] 甘春华, 王伟. 社交媒体的使用与满足:微博与微信的比较 [J]. 系统与信息技术学报, 2015, 17(4):351-363.
- [6] 林小艺. 当前主流外卖 APP 竞品分析 [J]. 人人都是产品经理, 2016.
- [7] 中商经济研究院, 中国网上订餐行业现状分析及投资前景预测报告 [R]. 2013-5
- [8] Reto Meier. Professional Android 4 Application Development. 北京:清华大学出版社. 2012.
- [9] Karl E. Wiegers 著, 陆丽娜, 王忠民, 王志敏等译. 软件需求. 北京:机械工业出版社, 2013.
- [10] 杨丰盛. Android 应用开发揭秘. 北京:机械工业出版社. 2011.
- [11] Patrick McNeil. TheWeb Designers' Idea Book. 北京:人民邮电出版社. 2014.
- [12] 杨静宇. Java 程序设计与应用开发(第 2 版) [M]. 北京:清华大学出版, 2009. 5
- [13] 张桂元, 贾燕枫. MyEclipse 开发入门与项目实践 [ M ] . 北京: 人民邮电出版社, 2006.
- [14] 孙卫琴, 李洪成. Tomcat 与 JSP Web 开发技术详解 [M] . 北京: 电子工业出版社, 2003.
- [15] 史万鑫, 文明远, 何兆松, 余琴. 高校食堂 APP 的设计与开发 [J]. 信息与电脑(理论版), 2019, 31(18):96-98.
- [16] 刘伟. 基于 Android 平台的手机订餐系统的设计与实现 [D]. 扬州:扬州大学, 2018.
- [17] 刘丙为. 基于 Spring MVC 的公司订餐系统的研究与开发 [D]. 乌鲁木齐:新疆大学, 2018.
- [18] 孙霞. 基于安卓技术的订餐系统开发 [J]. 电子测试, 2016(21):91+97.

- [19] 张海藩. 软件工程导论 [ M ] . 北京: 清华大学出版社, 2008.
- [20] 宋磊, 董国范. 基于 Android 技术开发订餐系统 [J]. 电子测试, 2016, (3) :5-6.
- [21] 宫意惟, 何逸仙, 马佳芸, 黄芳琳, 范鹏, 卢乔嘉. 基于 Android 平台的双模式点餐系统设计 [J]. 工业控制计算机, 2019, 32(7) :104-106.
- [22] 付乐, 刘雨婷. O2O 餐饮店掌店宝 Android APP 开发 [J]. 计算机时代, 2019(2) :39-42.
- [23] 邓春丽. 基于 Android 平台无线点餐系统的开发与实现 [D]. 成都: 电子科技大学, 2018.
- [24] 蔡敢民. 基于 Android 的大型综合餐饮服务平台 app 应用设计和开发 [J]. 南方农机, 2019, 50(1) :151-152.
- [25] 刘申晓. 基于 MVC 框架的网上订餐系统的设计与实现 [D]. 解放军信息工程大学, 2010.
- [26] 崔成生. 基于 Android 平台的空中餐厅点餐系统的设计与实现 [D]. 南京: 东南大学, 2018.
- [27] 胡向颖. 基于 Android 的餐厅点餐系统的设计与实现 [D]. 青岛: 青岛科技大学, 2018.
- [28] 马强. 基于 Android 的智能手机网络订餐系统的开发与应用 [D]. 长春: 吉林大学, 2018.
- [29] 张俊翔. 基于 Android 平台的 WiFi 点菜系统的设计与实现 [D]. 南宁: 广西大学, 2017.
- [30] 唐南沙. 基于 Android 的智能点餐系统 [D]. 长沙: 湖南大学, 2017.
- [31] 蔡丹. 基于 Android 的校园外卖系统设计与实现 [D]. 南昌: 江西财经大学, 2018.
- [32] 王天泥. 当图书馆遇上手机 app [J]. 图书与情报, 2016(6) :83-86.
- [33] 陈振宇. 基于 Android 平台的手机点餐客户端设计实现 [D]. 西安: 西安电子科技大学, 2017.
- [34] 陈凯宇, 程旭兰. 豆果食谱类 APP 市场推广分析 ——以“豆果购物”为例 [J]. 信息与电脑, 2013(1) :32-34.
- [35] 孙浩. 基于 Android 无线点餐系统的设计与实现 [J]. 电脑知识与技术, 2016, 12(36) :106-107+110.
- [36] 胡郁. 人工智能与语音识别技术 [J]. 电子菜谱世界, 2016, 23(4) :23-25.
- [37] 陈勇, 杜卉婷, 骆迁, 张灿灿, 陈琪, 李璋, 黄兆祥, 黄万雄. 基于 Android 的餐饮服务系统分析 [J]. 信息通信, 2016(12) :127-128.
- [38] 刘欣宜. Android 平台下餐饮信息管理应用 [J]. 科技展望, 2016, 26(30) :8.
- [39] Anandamurugan S, Priyaa T, Babu M C. Cloud Computing for Linux and Android [M]. Mercury Learning & Information. 2020.

- [40] 孟磊, 张姝, 李航, 孙阳, 吴鹏. 软件质量与测试[M]. 西安: 西安电子科技大学出版社. 2015.
- [41] 杨胜利. 软件测试技术[M]. 广州: 广东高等教育出版社. 2015.
- [42] Coppola Riccardo, Ardito Luca, Torchiano Marco, Alégroth Emil. Translation from layout-based to visual android test scripts: An empirical evaluation[J]. Journal of Systems and Software, 2021, 171: 110845.
- [43] 曹秉岩. 基于 Android 的网络订餐系统的设计与实现[D]. 沈阳: 东北大学, 2016.

## 致 谢

光阴荏苒，转眼间在读工程硕士的学习即将结束，在攻读工程硕士的学习期间，我学习了不少的理论知识，既拓宽了我的知识面，也在学习中增强了我的自学能力和动手能力。经历大半年时间的磨砺，经过反复的修改添加，这篇毕业论文终于完稿，这篇论文让我内心是五味杂陈，在选题、收集资料、论文的编写过程中，我喜悦过、痛苦过、彷徨过、无助过。我因受家庭变故（父亲近几年因脑梗瘫痪在床至去年年底病逝）和工作繁忙的影响，导致我的硕士学位论文的写作工作长时间停滞不前。回首大半年来收集、整理、思索、停滞、修改直至最终完成的过程，我得到了不少的关怀和帮助，这次论文的完成，不止有我自己的努力，同时也有老师的指导，同学的帮助，在此我要向他们表达我诚挚的谢意。

首先，我要衷心感谢我的论文导师陈宁江教授。他为人谦和，平易近人。在论文的写作过程中，每当我有所疑问向他请教时，他总会放下繁忙的工作，不厌其烦、及时高效的给与指导，为我的论文设计提出了宝贵的意见和建议。他严谨的治学之风和他对我的关心、教诲我更将永远铭记。

其次，我要感谢在这几年里陪伴我的亲人、朋友和同学们、同事们，感谢他们在我感到迷茫和懈怠时给我的敦促和鼓励，在我的论文写作遇到瓶颈时提供的支持和帮助，有了他们我的人生才更丰富、才不孤独、才有不懈的动力，谢谢他们。

最后，我要感谢广西大学计算机与电子信息学院的各位领导和老师的辛勤付出，是他们让我学到了更多的知识，使我接触到了更为广阔的平台。我会在今后的工作当中继续努力学习，认真钻研，奉行“活到老，学到老”的学习理念，不断提升自身的知识与技能，努力为国家和社会做出有益的贡献。

感谢各位老师审阅我的论文。

覃志宏

2021 年 3 月

## 附 录

《基于推荐算法的移动订餐系统设计》，发表于《中国科技人才》2020 年第 24 期。