

分 类 号：TP311.5

单位代码：10183

研究生学号：201353R699

密 级：公开



# 吉林大学 硕士学位论文 (专业学位)

微信拼团商城的设计与实现

**The Design and Implementation of  
WeChat Group-Buying Mall**

作者姓名：宋志鹏

类 别：工程硕士

领域(方向)：软件工程

指导教师：张家晨 教授

陈 伟 副教授

培养单位：软件学院

2016年6月

## 论文情况说明

该论文为吉林大学软件学院在职（半脱产）软件工程硕士（单证）学位论文，论文形式为项目开发、实验技术、课程设计类论文，不属于学术型论文。

特此说明。



吉林大学软件学院  
2016 年 6 月

微信拼团商城的设计与实现

The Design and Implementation of  
WeChat Group-Buying Mall

作者姓名：宋志鹏

领域（方向）：软件工程

指导教师：张家晨 教授

陈 伟 副教授

类别：在职工程硕士

答辩日期：2016 年 5 月 20 日

未经本论文作者的书面授权，依法收存和保管本论文书面版本、电子版本的任何单位和个人，均不得对本论文的全部或部分内容进行任何形式的复制、修改、发行、出租、改编等有碍作者著作权的商业性使用（但纯学术性使用不在此限）。否则，应承担侵权的法律责任。

吉林大学硕士学位论文原创性声明

本人郑重声明：所呈交学位论文，是本人在指导教师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：宋志明

日期：2016 年 6 月 7 日

## 摘 要

### 微信拼团商城的设计与实现

随着微信用户规模的扩大,逐渐衍生出另一种商业模式:微商。微商基于微信公众平台,结合其接口开发相应的商城,通过微信公众号或二维码扫描技术进行关注,并在微信朋友圈推送商品销售信息。

相比于淘宝之类的传统互联网营销,微商的经营手段灵活,消息推送及时,采购和支付均在手机上完成,方便易用。用户也不需要安装各式各样的APP,对于用户智能设备的硬件配置要求很低。对于微商运营者而言,该模式也操作简捷。用户随时随地可以通过手机进行商品推广、查看订单情况、在线客服等。另一个更加吸引广大用户的原因,是其去品牌化特点。传统的电商平台是中心化、平台化、品牌化。微商转变原有的以产品为中心方式,而注重其渠道,它强调人之间的交流沟通。如果你的人品不好,哪怕你的商品再好,也一样卖不出去。所以微商首先要卖的是人品、信任、逼格。

如何把握该商业模式,研发具有特色的微商城,是目前一个很热门的课题。本人结合多年的软件开发经验,重点研究微信拼团商城的业务逻辑,设计合理的用户界面,研发稳定高效的后台管理中心,提出了基于Spring MVC+AngularJS 技术的系统设计方案。通过Spring MVC实现业务逻辑,并实现业务解耦;通过AngularJS实现界面事件的双向绑定。同时结合Lodop实现物流单的打印,微信API完成微信支付,kuaidi100接口实现物流跟踪。

本论文的研究工作主要有以下几个方面:

1. 深入研究Spring MVC的技术应用。
2. 研究AngularJS技术。
3. 研究微信公众平台开发技术。
4. 研究面向接口开发技术。

**关键字:**

微商; Spring MVC; AngularJS; 微信公众平台;

## Abstract

### **The Design and Implementation of WeChat Group-Buying Mall**

With the expansion of the size of WeChat users, there is derived another business model: WeChat-Business. WeChat-Business based on the WeChat platform, combined with its interface to develop the corresponding mall, through pay attention to the WeChat public number or scanning two-dimensional code build the connection, and push commodity sales information in self WeChat friends.

Compared to others traditional Electronic Commerce, such as Taobao, WeChat business has flexible management means, timely message send, procurement and payment are completed on the phone by conveniently. Users do not need to install a wide variety of APP, hardware configuration requirements for users of intelligent devices is very low. As for the WeChat shop, the model is more forthright. Users can push message、check orders、online communication as whenever and wherever possible. Another reason to attract the majority of users is unbound characteristics. Traditional electronic business platform is focus on the centralization 、platform、brand. But the WeChat business is emphasize the channel and more effective communication ,neither nor the product. If your character is not good, even if your product is good, also nobody will buy it. So the WeChat business is the first to sell character, trust, BIG.

How to grasp the business model, research and development with the characteristics of micro mall, is becoming a very popular topic. Combined with some years of software development experience, I focus on the business logic of WeChat group mall, design a reasonable user interface, research and develop a stable and efficient background management center. A system design scheme based on Spring MVC technology, and realize the business decoupling. Through AngularJS to achieve two-way binding interface events, at the same time combined with Lodop assembly to achieve the logistics of a single print, with WeChat API to complete WeChat payment, with kuaidi100 interface realization of logistics tracking .

The research work of this paper mainly has the following several aspects:

1. In-depth study of MVC Spring technology application.
2. Investigative AngularJS techniques.
3. Research on WeChat public platform development technology.

4. Research on interface development technology.

**Keywords:**

WeChat Business; Spring MVC; AngularJS; WeChat public platform;



目 录

第 1 章 绪 论 .....	1
1.1 课题意义及背景.....	1
1.2 目前微信商城现状.....	2
1.3 本文研究内容与主要工作.....	3
1.4 论文组织结构.....	4
第 2 章 软件架构相关理论与方法 .....	5
2.1 软件架构 .....	5
2.2 Spring MVC 框架.....	6
2.3 ibatis 框架.....	7
2.4 AngularJS 框架.....	8
2.5 开发平台搭建.....	9
2.6 本章小结.....	10
第 3 章 系统需求分析 .....	11
3.1 总体业务需求.....	11
3.2 主要功能分析.....	17
3.2.1 微信公众号申请 .....	17
3.2.2 拼团规则 .....	18
3.2.3 商品发布流程 .....	18
3.2.4 用户拼团采购 .....	19
3.2.5 订单管理 .....	20
3.3 非功能性分析.....	22
3.4 数据关系分析.....	22
3.5 本章小结.....	24
第 4 章 系统架构设计 .....	24

4.1 系统硬件体系结构.....	25
4.2 系统架构设计.....	25
4.3 网站 UI .....	27
4.4 AngularJS 指令设计.....	29
4.5 系统性能优化.....	30
4.5.1 分布式内存方式 .....	30
4.5.2 分表方案 .....	30
4.5 本章小结.....	31
第5章 系统实现 .....	33
5.1 开发环境与技术 .....	33
5.2 微信公众号配置 .....	33
5.3 AngularJS 指令.....	35
5.4 商品管理 .....	36
5.5 订单管理 .....	40
5.6 微信用户商品采购 .....	42
5.7 Memcached 配置.....	44
5.8 本章小结 .....	46
第6章 总结与展望 .....	47
6.1 总 结 .....	47
6.2 展 望 .....	47
参考文献 .....	49
致 谢 .....	52

## 第1章 绪论

### 1.1 课题意义及背景

随着智能手机的普及，原来必须在电脑上进行的一些工作已可迁移到手机上完成。手机端由于其具有的灵活便携性，随时随地可浏览采购支付，成为许多商家争抢的新领地。

淘宝<sup>[1]</sup>的电子商务模式已经证明是相当成功的，数以千万计的商家入驻并进行商务活动。随着微信<sup>[2]</sup>的兴起，到2016年官方公布的数据显示，日登录用户量已经达到5.7亿。腾讯公司又结合它的微信产品，及时的推出了微信公布平台，供用户进行信息的推广和商品的交易。许多有眼光的商户已经捕捉到了这份商机，研发微信商城<sup>[3]</sup>，借助微信朋友圈进行推广，并实现商品的销售。

如果微信商城仅仅类似淘宝完成商品的移动交易方式，那它也不具备足够的吸引用户的资本。为了体现微商城的灵活性，在原有的商城交易上形成了一种新的交易模式：拼团购买。该模式类似于捆绑销售方式，必须当用户的购买人数达到一定的数量才可成团，表示购买成功。当然，该种销售模式商品的价格比个人单件购买要低的多。

各商家为了体现自己的特色，又在该基础上添加了许多具有个性色彩的商业规则。比如为了吸引新用户关注自己的微信号，会定期的搞活进行抢红包、抢优惠券、抽奖。相关的优惠券在支付时可以低用一定的费用。对于第一个开团的用户，对于其成交价的8.5折优惠等等一系列的活动。所有这些，都体现了微信商城的与众不同，灵活多变。

本课题的主要目的就是研究微信拼团商城的业务功能，并结合相关技术实现之。要求系统具有良好的用户体验，并同时具有高可扩展、可维护性<sup>[4]</sup>，以方便后期的功能变动。该系统的特点主要体现在以下几个方面：

1. 结合微信公众号，实现手机拼团活动。
2. 团购业务流程新颖。
3. 前台用户界面美观，操作流畅。
4. 自动订单打印和跟踪。

## 5. 多种不同的表格数据生成。

## 1.2 目前微信商城现状

微信商城<sup>[5]</sup>是从属于微信产品的一个第三方开放平台，从其出现到目前大概有了 5 年时间，目前各大企业几乎都开通了微信公众号来进行信息的垂直推送<sup>[6]</sup>。

随着移动设备和通讯网络的快速发展，互联网+的概念进一步深入社会的方方面面，商业模式有了进一步的演化，由原来的大而全，向小而精转化，向个性化转向。微信商城正好可以满足社会的这一需求。微信商城相比于传统的淘宝商城之类的电子商务平台，其最大的区别就在于产品的分类布局比较单一，用户专注于一类产品的专业营销。

从微信商城平台的运营来讲，它也不同于传统的电子商务营销。传统的电子商务营销主要是通过百度推广之类的方式，让互联网用户可以快速的搜索互本产品。而微信商城则完全不同。它是通过自己的微信朋友圈<sup>[7]</sup>扩散，类似于“人脉”。只要你的粉丝够多，你的人脉够广，那么你的影响就更多，潜在的买在也就越多。它的这种推广模式是爆炸式信息推广，购买过程不是支付完毕该商务活动就结束。如果某个用户觉得某个商品不错，他下单购买，然后他再将该商品信息发布到自己的朋友圈，吸引自己的好友继续购买，这样才能形成拼团效应。如果当前的购买活动达不到商家规定的拼团人数，则购买活动失败，商家自动退款给用户。

微商城从产品的类型来将，一般都比较精致，信息量不大，针对性强。这个与传统的电子商务平台追求产品丰富，分类全面又有区别。以淘宝为例，淘宝上的服装商铺，为了方便用户的商品选择，会提供各种不同的分类，各种不同的套餐，多种不同的分类搜索。而微商城因其主要是在移动平台操作，则界面布局简捷，只有几个大的分类和一个关键字搜索，用户主要的操作就是通过滑屏上下浏览商品。

从开发技术上讲，微商城的开发重点关注其跨平台性和快捷性。因为相对于 PC 而言，移动设备的数据处理能力、通讯能力、存储能力毕竟要弱一些。要想让大多数用户均可以方便的使用微商城，则必须进行技术的优化。每次请求的数据量要尽可能的少，尽量少占用客户端资源，业务处理放在服务器端去完成。

目前微信商城的发展有以下几个特点：

### 1. 开发技术不规范<sup>[8]</sup>。

2. 结构僵化，不方便扩展。

3. 稳定性差。

针对上述的种种问题，本课题在研究与设计时候，侧重于技术的选型和实现，尽量利用现在新的技术手段来达到高效率、方便扩展的目标，其主要特点有：

1. 解耦<sup>[9]</sup>。

最大限度的实现数据层、业务层、UI层的分解。保证UI层仅仅只负责数据呈现，事件的逻辑处理在业务层完成，处理完之后的数据进入数据层保存。当以后商务规则进行变动后，只需要修改业务层的实现过程，而不用修改其接口部分。

2. 所见即所得。

用户的UI体验<sup>[10]</sup>上，突出用户体验。在实现业务功能的同时，减少与服务器的交互次数。

3. 提高安全稳定性。

安全稳定是系统运行的基本要求。为了保证商务过程的安全性，所有的会话过程均加密，交易过程结合用户手机校验码的方式进行综合验证。对每一步资金的往来，不管是入帐还是出账，均记录其日志<sup>[11]</sup>，并且日志文件只供管理员查看不可修改或删除。

### 1.3 本文研究内容与主要工作

本文研究在当前微商火爆的形势下，研发出一套商务模式新颖，运行稳定，框架灵活的微信商城平台，进行商品的拼团式销售。该销售模式通过捆绑式销售的方式，加大商品的销售量。同时，利用捆绑销售的价格优势，吸引更多的用户关注和购买商品。销售渠道也由原来的商家找客户，转变为由客户来拉客户。

本课题研究的工作重点有如下几点：

1. 新颖的商业逻辑研究。

好的商业模式或思想，决定了平台运营的市场前景。本课题在传统的电子商务基础上，研究拼团购买、梯形价格、产品赠送等模式，大大的吸引用户关注。

2. 研究微信公众平台。

微商城的运作是基于微信公众平台的，需要通过申请、审核、动态菜单定制、UAuth<sup>[12]</sup>认证等一系列过程，才能搭建成功。对于微信支付，要研究其用户secretkey和API接口<sup>[13]</sup>，

为了保证交易安全性，还要进行https加密<sup>[14]</sup>。

### 3. 研究Spring MVC架构设计<sup>[15]</sup>。

分析各业务流程，将其业务逻辑和数据访问进行解耦。业务层和数据访问之间的访问均通过接口访问，屏蔽具体的实现细节。对于业务的变动，仅仅是修改其实现过程，而各层与层之间的接口访问相对稳定。

### 4. AngularJS技术<sup>[16]</sup>。

AngularJS提出了一种双向绑定思想，即用户的任何数据改变或动作，均会实时的在界面呈现，不需要过多的用户业务操作。该技术大大简化前台设计。对于复杂的业务事件，可能扩展directive指令<sup>[17]</sup>，功能相当强大。

## 1.4 论文组织结构

本文共分六章，对微信拼团商城的分析与设计章节安排如下：

第一章绪论。描述选题的实用性和特色性。

第二章系统设计理论。从理论上介绍本系统采用的相关技术和工具，以及为什么要选择这些技术和工具。

第三章业务功能分析。对系统进行业务流程分析，并进行模块划分，功能详细说明。

第四章系统架构设计。结合第二章的架构理论，进行系统架构方案设计。

第五章系统实现。选择重要环节，讲解系统的前台和后台实现，给出关键代码。

第六章总结与展望。列出课题的成果以及不足之处。

## 第2章 软件架构相关理论与方法

信息化的飞速发展，软件产品作为社会生产管理的一种手段，日益深入民众生活。软件产品的研发具有一定的难度，具有复杂性、不可捉摸、思维产物等固有特点，使其在质量、工期、维护方面很难完全满足用户需求。软件开发者一直致力于追求一种高效的、可靠的监管方式，一种灵活的框架设计，以开发出高质量的产品。本章结合软件工程的相关思想，研讨系统架构的设计理论与方法。

### 2.1 软件架构

软件架构（Software Architecture）<sup>[18]</sup>是一个软件的草图，它由一系列相关的设计模式组成，确定了软件设计思路。软件架构的选型与项目有关，根据不同的项目特点，突出应用不同的技术。软件的架构设计一般由系统分析师和构架设计师来完成，好的架构，就是一个强壮的母体，为项目的成功打下坚实的基础。

软件的架构设计时要满足3大原则<sup>[19]</sup>：

1. 单一职责原则。即一个模块或一个组件，完成其核心的业务。
2. 接口隔离原则。设计过程要体现业务性，可不是关注业务的具体实现细节。
3. 开闭原则。方便项目的维护，维护过程尽量不要影响已有的功能模块，防止错误的蔓延。

软件架构设计要达到如下的目标：

- 简明性<sup>[11]</sup>。通过架构设计，将原来复杂的业务问题组成一个有机的框架，形成一个整体解决方案。
- 可行性。架构设计要可行，可适用，不是所有的项目都要追求高大上的架构。
- 可扩展性(Scalable)<sup>[20]</sup>。鉴于软件项目的不确定性，需要架构能容纳一定的业务修改。
- 技术前瞻性(Time to Market)。架构要充分考虑到现有的软硬件资源，以及当前流行的开发思想，以便软件具有更长的生命力<sup>[21]</sup>。

根据关注的不同角度，将架构分成三种<sup>[22]</sup>：

1. 逻辑架构。

从逻辑思想上抽象业务，形成其业务关系。根据软件的组件之间的关系，按一定的逻辑形式划分为许多层：UI 界面、BLL、DAL、Service 层等。按照 MVC 的思想，将业务与呈现分面，将操作与数据分开，可以分为 Model 数据模型层、View 视图层、Control 业务控制层，DAO 数据持久化层<sup>[23]</sup>。

## 2. 物理架构

为了实现软件的逻辑而已，需要在物理硬件上进行设计。传统的服务器资源是集中式架构，随着社会的发展，分布式、大数据、云平台日益成熟。物理架构考虑的是，如何结合硬件资源，将业务瓶颈分流，利用分布式或平衡负载思想转移到多台设备上去执行<sup>[24]</sup>。

## 3. 系统架构

系统架构整合软硬件资源。在全面分析系统业务功能的同时，也关注其非功能性特征或潜在业务需求，注意全面的分析系统的生态环境，以及其与外界的对接等。系统架构是架构设计中难度最大的一项工作，对后期的设备选型、开发技术确定、系统实现都有指导性意义。

## 2.2 Spring MVC 框架

Spring 是一个轻量级控制反转 (IoC) 和面向切面 (AOP) 的容器框架<sup>[25]</sup>。该框架是为了处理复杂项目而产生的，其目的是降低各层之间的耦合度。Spring MVC 将控制器、模型对象、分派器进行分离。设计是围绕分发器 (DispatcherServlet) 展开的，DispatcherServlet 负责将请求派发到特定的 handler，在派发时，进行业务的判断与控制。Spring 具有如下特点<sup>[26]</sup>：

- 角色功能明确。验证器 (Validator)、控制器 (Controller)、模型对象 (model object)、均由独立的对象来实现，各层之间相互调用，也可由一个对象提供多个业务逻辑 (不推荐)。
- 配置文件管理方式。应用类和框架类都作为 **JavaBean** 配置，支持在一个上下文中引用其他上下文中 **JavaBean**，调用灵活。配置细节按规定的格式存放为一个文件，后期改动方便。
- 可适配控制器 Controller。用户制定不同的业务控制器，用户根据需要选择配置，而



不是从单一控制器继承。

- 代码重用。封装业务对象和业务规则子类，在需要的地方引用即可。
- 定制绑定。可定义数据与界面的绑定规则，定义业务逻辑来实现不同数据类型之间的转化。
- 自定义验证。它已经提供了 6 类基本的验证机制，用户也可以继续其 `validation` 接口，定义特点的验证方式，如：座机号码有效性的检验，对录入日期格式的检验。
- 可定制路径映射。Spring 提供 URL 映射与解析机制，可以按不同的参数，将其解板映射到不同的页面上。
- model 转换灵活。它使用基于 Map 的 Key-value 键值对来保存数据，并根据 Key 与各种 UI 控件集成。
- 支持主题模式。主题 theme 是当前流行的样式管理方式，通过更换主题可以快速的更换 WEB 的界面，不需要修改其他任何的业务功能，部署维护方便。

## 2.3 ibatis 框架

该组件是 apache 的一个开源项目<sup>[27]</sup>。它在原来普通 SQL 语句的基础上，进一步将 SQL 语句文本化，并支持相关参数传递。它利用 xml 配置文件，将接口和 JavaBean 业务对象映射成数据库中的记录。

它具有如下优点：

- 上手简单。相对于 Hibernate 的会自动化 ORM 模式，它是半自动的，用户可以在 XML 配置文件中直接书写 SQL 语句来实现相关数据操作。
- SQL 管理规范，直观性强，方便与其他系统的对接。
- 将 SQL 从原来的数据访问库剥离出来。
- 支持部分 Hibernate 功能，可以将业务对象与数据库的表关系一一映射。
- 支持动态表关系映射。
- 支持编写动态 SQL 语句。

相对 Hibernate，它的缺点也很明显<sup>[28]</sup>：

- 对于表单字段多的应用环境，SQL 工作量很大。
- SQL 语句依赖于数据库，移植到不同数据库工作量大。

- Xml 配置文件里标签 id 必须唯一，DAO 方法重载困难。
- 与 SQL 语法紧密相关，具体实现仍然依赖于 SQL。
- DAO 层过于简单，后期需要进一步的分解。
- 多表操作时，不支持级联操作。
- 编写动态复杂 SQL 不方便调试。
- 提供的动态 SQL 且可读性低。
- 使用不当，容易导致 SQL 性能问题。
- 支持有限的参数类型。
- 多参数时，传参方式生硬。

总结：

ibatis 的优点其实也是 ibatis 的缺点。正因为 ibatis 使用简单，直接将 SQL 语句以明文存放在配置文件里，所以对于 SQL 的性能便更多依赖于程序员的编程水平了。

ibatis 适用于小型且项目使用，对于中大型项目还是推荐使用 Hibernate。如果觉得 Hibernate 因为 ORM 映射而效率低下，则可直接使用 Spring 提供的 JDBC 简单框架<sup>[29]</sup>（Template），同样支持对象映射，并且支持现成的数据连接池。

## 2.4 AngularJS 框架

AngularJS<sup>[30]</sup>是谷歌的一个 JavaScript 框架，旨在简化前端应用程序的开发。现在微软的 Asp.net MVC5 版本也集成了 AngularJS。市面上还有一个比较成熟的组件：EasyUI<sup>[31]</sup>该组件也可灵活的实现数据绑定。之所以选择 AngularJS 框架，是因为其具有良好的设计思想，可以完整的实现 UI 层与业务层分离，简化界面复杂度。同时，具有强大的扩展指令。

AngularJS 具有如下 5 个明显的优点：

### 1. 良好的应用程序结构。

JavaScript 是脚本程序，依附于 Html 页面，书写灵活，没有明确的结构。虽然在编写小应用程序的时候没有问题，但如果界面数据量大，并且业务逻辑复杂时，可能 JS 代码就有上千行。该脚本对于后期的维护调试都是相当费劲。使用 AngularJS，您可以通过 MVC（模型 - 视图 - 控制器）或 MVVM（模型 - 视图 - 视图模型）模式来组织

源代码。AngularJS 是一个 MVW 框架，你可以将业务逻辑封闭成独立的代码块，在页面中引入，它可显著提高应用程序的可测试性和可维护性。

## 2. 双向数据绑定。

HTTP 的特性是不保持状态数据的，每次数据的传递必须要通过重新发送请求，进行页面刷新来完成。为了避免页面的不断刷新，现在开发都采用 JS 进行异步请求。但 JS 的异步请求必须要程序员书写大量的代码。而 AngularJS 的数据模本身就支持数据模型改变时的视图更新，反之也可。这样大大减少 JS 代码。

## 3. Directive 指令。

AngularJS 指令让用户自定义方法，进行页面的行为控制，也可以更进一步，扩展 API 函数。你可以扩展任何你所需要的数据检验、异步提交、文件上传、视频播放事件。并且该扩展的指令，是支持跨浏览器的。

## 4. HTML 模板<sup>[32]</sup>。

模板是电子商城设计中经常使用的技巧。结合模板技术，将应用程序的呈现与业务逻辑实现过程完整的分离，则程序人员与美工并行开发。后期如果需要进行界面更新，只需要重新设计一套模板，复制到相应目录下，修改一下配置文件即可。开发部署都相当方便。

## 5. 可嵌入、注入和测试。

AngularJS 支持开放式的依赖注入。如果用户在客户端调试时，需要临时的设置 JS 对象的值，可直接按 F12，进行客户端调试，并修改对象，而不用修改程序或重录入，再启动运行界面。该方式提供了很好的可测试性，测试人员可以直接利用依赖注入方式输入多组测试用例。

# 2.5 开发平台搭建

微信拼团商城的主体平台开发基于 Java 技术，在 MyEclipse10.0+ SSH+ ibatis + Tomcat7.0 + AngularJS + MySQL5.5 平台上实现<sup>[29]</sup>。SSH 经过多年的发展，已经完全成熟，具有其他技术无法比拟的优势：

- 开源组合。开发人员可全面了解其内部机制，并进行深度订制。网上各类组织为其提供了很多丰富的插件，开发效率高。

- 典型的 MVC 思想。Struts 实现<sup>[33]</sup>Web 层数据显示 View 部分，Hibernate 实现 DAO 层的数据访问控制，Spring 的作用贯穿了整个中间层，将 Web 层、Service 层、DAO 层及 PO 无缝整合，其数据服务层用来存放数据。

- 优秀的解耦性<sup>[30]</sup>。很少有软件产品的需求从一开始就完全是固定的。客户对软件需求，是随着软件开发过程的深入，不断明晰。SSH 完美的进行了项目解耦，使各部分独立，易于维护和扩展。

- 高效率。系统结合 Memcached 缓存方案，将相关资源调入内存缓存，大大加快系统的响应速度。

- 安全性。丰富的加密算法、文件部署方案、用户访问限制等措施，保证系统数据文件和数据库的安全。

## 2.6 本章小结

本章主要介绍了软件架构设计的相关概念，并讨论总结了软件框架设计的步骤、法则，重点解释了 Spring + MyBatis + AngularJS 框架的优点。充分阐述微信拼团商城设计过程中解耦思想、组件化开发思想，以及如何结合该模式以追求最高灵活性、最大扩展性、最快的响应时间，为后续的系统设计做好理论铺垫。

同时，本章简要的介绍了所采用的开发平台和开发框架版本组合，该组合是当前的主流搭配模式。

## 第3章 系统需求分析

高校宿舍管理工作涉及到的人员面很广，有教师、学生、保卫处、学生处、后勤管理处等，其工作职能参差交错。本人从的宿舍管理工作现状入手，分析现有的管理方式，同时调研各不同的部门和个人，结合本人多年的班主任管理经验，完整的梳理系统的功能需求。

另外，微信拼团商城的相关数据来源涉及到教务处、财务处、学工处、后勤管理处。分别调研各处的数据格式和数据接口，以便在微信拼团商城中实现各环节的数据对接。

### 3.1 总体业务需求

微信拼团商城主要功能是通过关注微信公众号，快速的在智能设备上实现商品的拼团购买过程。其最大特色是体现拼团购买，微信支付。为了吸引用户关注本微信公众号，购买过程以阶梯价格，即买的人越多其最终售价越低，同时对于团长（第一个开团的用户），实行无条件打折机制，鼓励用户抢着开团，多开团。

系统的主要用户有两类：系统管理员和微信用户。系统管理员进行商品信息的维护、发货、订单打印等；微信用户则关注本微信公众号，进行商品的浏览、收藏、采购、支付。

从功能角度划分，微信拼团商城包括 15 大功能模块，用例图<sup>[34]</sup>如图 3-1 所示，分别是：

- 系统管理(UC001)。主要是负责系统运行的权限管理，包括用户管理和权限管理，本系统的管理采用权限组管理方式。
- 平台管理(UC002)。该模块主要完成系统的运行参数设置、基本数据设置、微信关注用户管理、数据备份还原、日志管理。
- 商品管理(UC003)。管理商城的商品相关操作，以及订单管理，销售金额统计等。
- 微信前台 (UC004)。系统的核心部分，主要是用来呈现商品，供微信用户来拼团，支付等。包括的功能主要有：商品分类显示、查询、采购、支付、个人中心等。

整个系统的功能结构如下图 3-1 所示：

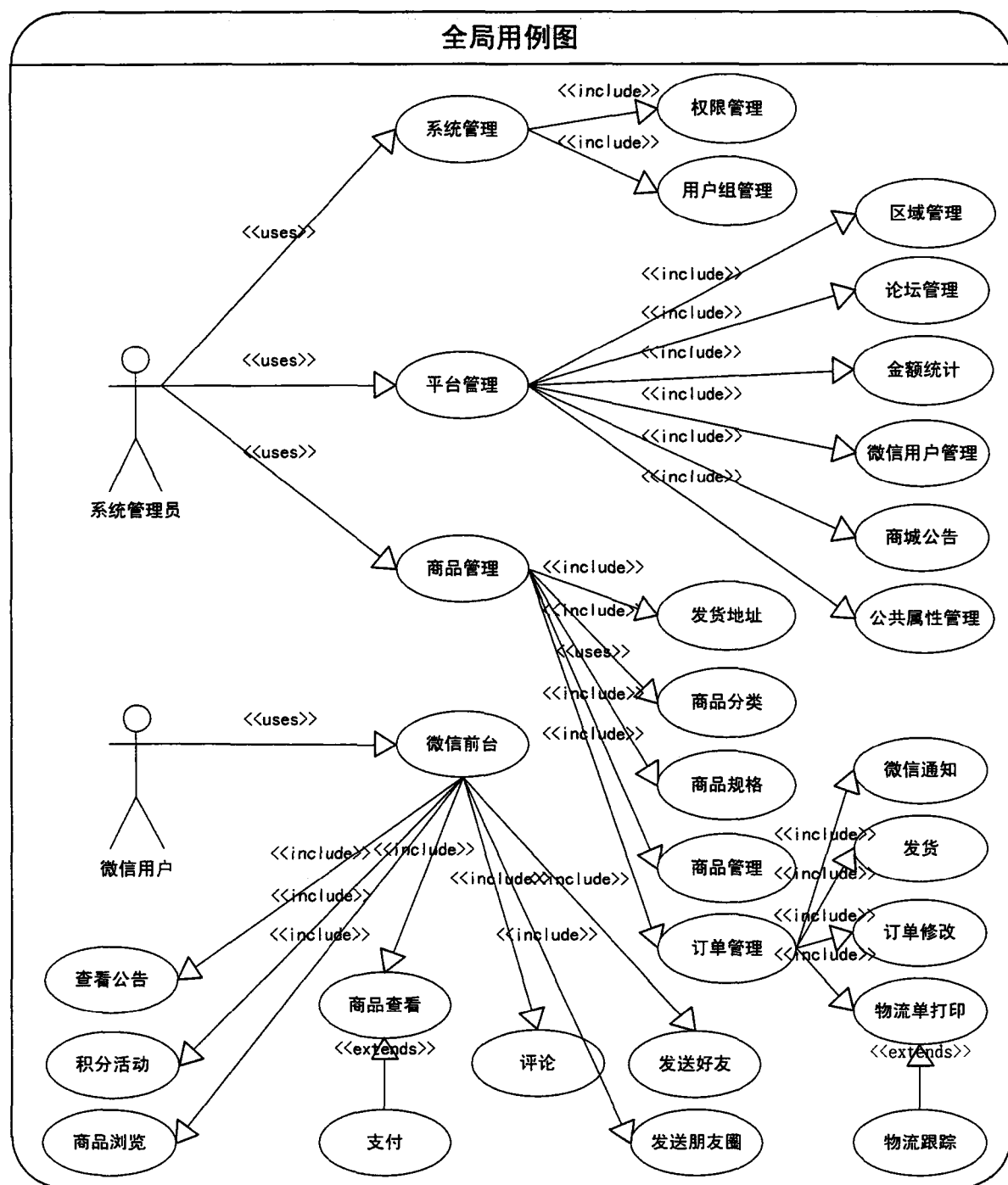


图 3-1 微信拼团商城全局用例图

为了保证系统的运行，前期要做好基础数据工作的导入或设置。比如：系统管理员、权限、全国区域信息、物流接口申请配置等，学工处、宿管员、学生进行协作，完成入住。先挑取核心的业务逻辑模块进行用例描述。商品发布用例描述如表 3-1 所示。

表 3-1 发布商品用例描述

用例编号	UC003-004（编号含义：表示第 3 个模块第 4 个用例）
用例名称	商品管理
用例描述	发布商品
参与者	管理员
前置条件	涉及的相关基础数据： <div>1. 商品分类以及规格参数。</div> <div>2. 区域信息。</div>
基本流程	<div>1. 管理员登录后台。</div> <div>2. 打开商品管理，点击“添加商品”，显示商品发布界面。</div> <div>3. 输入商品信息：全称、简称、简要描述、拼团人数、快递费，所售省份、商品规格。</div> <div>4. 上传商品图片：首页图、图例、视频信息。</div> <div>5. 设置价格参数：市场价、参团折扣价、折数、团长折数、单独购买价、总库存量。</div> <div>6. 设置交易时间：如 10 天有效。超过该时间拼团未达要求的，则表示拼团失败，自动退款。</div>
后续动作	<div>1. 可在微信公众号中查看到当前商品信息以及图片。</div> <div>2. 销售成功后，库存量会相应减少。</div> <div>3. 商品下架时，修改其状态，但不真实的删除该商品。</div>
例外流程	<div>1. 商品输入信息不正确或是不完善，及时提醒。</div> <div>2. 商品发布失败，提示管理员失败原因。</div> <div>3. 记录操作失败日志。</div>
非功能性需求	<div>1. 提示信息准确、丰富，有用户操作指南。</div>

在用户生成订单后，如果拼团成功，管理员可在后台订单管理中心查询到当前的订单，并进行订单物流单打印，以及发货处理。其物流面单<sup>[35]</sup>打印用例描述如表 3-2 所示。

表 3-2 物流单打印用例描述

用例编号	UC003-005-001(编号含义:表示第 3 个模块第 5 个用例子功能 1)
用例名称	物流单打印
用例描述	拼团成功,可选择打印物流单,以供后面发货。
参与者	管理员
前置条件	1. 订单信息。 2. 微信用户信息。 3. 打印规则配置。
基本流程	1. 检索交易订单,查看“交易中”的订单信息。 2. 若当前有“快递单号”,即可打印。 3. 选择物流分类,目前单位支持宅送、天天、中通、汇通。 4. 点击打印。 5. 输入当前物流单号。 6. 根据当前选的物流分类,自动打印成对应的物流单格式。 7. 操作结束。
后续动作	1. 查看物流打印单效果。 2. 打印效果不好时,可能重新打印。
例外流程	1. 物流单号不存在的,提醒打印失败。 2. 连接不上打印机的,提醒打印失败。 3. 打印机缺纸的,提醒打印失败。
非功能性需求	1. 打印格式一定要与物流公司的格式对应。

在微信用户购买过程中,为了方便用户购买,特别提醒两种购买模式:拼团购买与单独购买。拼团购买必须要满足拼团人数才会成功,其价格相对单独购买要优惠。单独购买一个人也可以进行购买,价格会适当的高于拼团价格,且没有任何的优惠活动。用户购买用例描述如表 3-3 所示。



表 3-3 用户购买用例描述

用例编号	UC004-005（编号含义：表示第 4 个模块第 5 个用例）
用例名称	用户购买
用例描述	微信关注用户在微商城进行商品的购买以及支付
参与者	微信关注用户
前置条件	1. 用户已经关注本微信公众号。
基本流程	1. 打开微信，进入本微信公众平台。 2. 进入微商城。 3. 浏览或查询商品。 4. 点击查看商品明细。 5. 点击开团。 6. 团购的数据缺省为 1，直接微信支付。 7. 若需要使用优惠券的，可以选择使用优惠券，以及当前使用的面额。 8. 输入微信支付口令，提示支付成功。 9. 支付成功后，可以将该商品发布到朋友圈或是发送给某个朋友，以吸引其他人参团。 10. 若想单独购买，则在第 5 步点击单独购买。 11. 若不想开团，则可点击附近的人或附近的团，检索已经开的团，并参团。 12. 购买过程完成，可以在个人中心查看到购买订单信息。
后续动作	1. 在个人中心查看购买订单信息。 2. 跟踪本单的物流信息。 3. 如果拼团交易失败，跟踪退款信息。
例外流程	1. 对于售罄的商品，不可开团或单独购买，点击时显示当前已经销售完毕。 2. 对于支付失败的，准确提示交易失败信息。 3. 如果当前购买人数太多，服务器繁忙的，提醒“服务器正忙，请耐心等待”。
非功能性需求	1. 支付时提醒正在支付友好界面。 2. 注意支付时的安全性和响应效率。

在微信推广过程中，为了前期的推广以吸引更多用户关注微信公众号，特别提供了

“参加活动有礼”栏目。用户可以点击进入相关的栏目，进行微信的问答，以及抽奖活动，以获取数量不等的优惠券。该优惠券有明确的面值以及使用期限，可支付时抵等值的金额。用户积分活动用例描述如表 3-4 所示。

表 3-4 用户购买用例描述

用例编号	UC004-002（编号含义：表示第 4 个模块第 2 个用例）
用例名称	积分活动
用例描述	微信关注用户进行积分活动游戏或问卷调查，以获取一定优惠券。
参与者	微信关注用户
前置条件	1. 用户已经关注本微信公众号。
基本流程	1. 打开微信，进入本微信公众平台。 2. 进入拼团商城微信公众号首页。 3. 点击推送的游戏栏目。 4. 回答问题或是领取优惠券。 5. 如果是回答问题类的调查式活动，则必须回答完所有问题，系统会根据当前的提交内容正确率，赠与当前用户一定的优惠券。 6. 领取优惠券的活动类似于大转盘的抽奖，抽取到什么就是什么。 7. 所有的优惠券都是独立的面值，独立的有效日期，用户可以查看自己的优惠券。
后续动作	1. 在个人中心查看优惠券。 2. 在支付时，选择利用优惠。
例外流程	1. 对于某些商品不支持优惠券支付的，或是有限定使用面额的，提醒用户。
非功能性需求	1. 优惠券到期提醒服务。

## 3.2 主要功能分析

下面对系统的主要功能进行分析，描述其要实现的业务功能和业务流程，为后期的开发提供实现思路。

### 3.2.1 微信公众号申请

系统的开发从微信公众号的申请开始。微信公众号分为订阅号、服务号、企业号。订阅号主要是个人信息发布的一种途径，只能通过认证的用户才可以定制主界面菜单，功能最弱。第二类是服务号，可支持消息群发，相关高级API接口等，可支撑微商城应用。第三类是企业号，可结合相关的企业生产管理，协作运营，支持高级接口的调用。

拼团商城根据其业务需要，可选择申请服务号或是企业号。本系统中使用的是服务号。微信公众号的申请评审费用一次 300，审核周期一般在一周左右。审核通过之后，用户登录微信公众号后台，进行以下配置：

1. 进入开发->基本配置，选择“开发模式”。
2. 填写接口配置，URL 以及 Token 参数，该参数保证服务器与微信公众平台的对接。

这里对后期服务部署要涉及的参数有：

- AppID：应用 ID。
- AppSecret：应用密钥，该值一个月可重置一次，供服务器连接微信公众号时身份认证用。

3. 配置服务器参数。

- URL：微商城服务器地址，如：  
`http://182.254.247.110/my/weixinjsclient/checkSignature`
- Token：令牌，如 `daifeifan123liufeng`
- EncodingAESKey：消息加解密密钥，保证通讯的安全性。  
如：`7ZZuXqDjFljchif6l7Y2falnevrlHnZxUh8erVhCptC`

4. 配置微商城服务器，部署项目。
5. 扫描关注微信，并进行相关功能操作。

### 3.2.2 拼团规则

为了吸引用户，系统特别引入独特的拼团规则。拼团规则即用户拼团购买商品的条件，按不同的条件进行商品购买，会产生不同的费用。

第一个独立购买规则。系统中商品提供3个价格，一个是市场价，一个是当前拼团价，一个是独立购买价。当用户因为人数限制不能拼团购买时可以独立购买，独立购买价设置为拼团价的1.3倍（该值可调）。

第二个拼团规则。如果用户拼团人数达到当前商品的销售阈值，则可享受拼团价。拼团价取市场价的7折（该值可调）。如果拼团失败，则所有费用如数返还。

第三个梯形价格。梯形价格的设置是这样的：对于某个拼团商品，假如预先设定的是5人成团，拼团价是30，团长8.5折优惠。在此基础之上，再添加一个业务规则：如果当前团人数达到10以人，则每多增加1人，价格少0.5元，最低价不得低于25元。该模式形成后，所有的用户的购买价格均由团购成功的最后一个人确定。如果本团的人数越多，那么本团的整体购买价格就越低。对于之前已经支付了相关价格的用户，在发货时，返还相应的金额。

第四个并订价格。所谓的并订是指当用户在本商城同时采购了多件商品时，管理员会与用户沟通，是否允许并单，即将多次购买的物品，一次性打包发送过来。对于并单节省的费用，50%（该值可调）返回给用户。

以上所有的购买行为，只要商品允许使用优惠券的，均可以按规则使用优惠券。并且，对于用户每一订达到某个量时，返回一定的值。如：当前某个订单达到100，为其返2%（该值可调）的优惠券。

### 3.2.3 商品发布流程

拼团商城的主要业务过程离不开商品，所以商品的类型设计，规格属性的设计，是系统的核心。商品发布时，用户需先准备好产品的图片（至少一张主图，可以配置若干干辅图）。为了更精彩的展现商品内容，还可以上传对应的视频文件，以更直观的展现商品。商品的发布与维护由管理员操作，所有商品的更新操作均需记录系统日志，以保证系统的安全性。其发布流程如下图3-2所示：

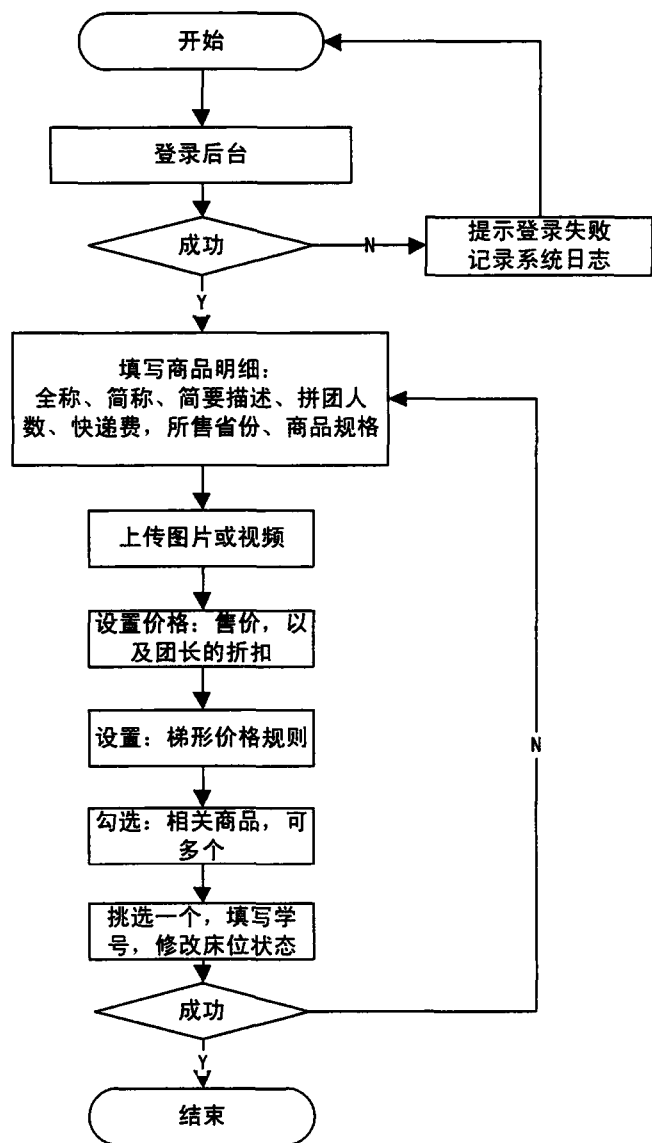


图 3-2 商品发布

梯形价格是本系统是提出的一种新型概念：即每个用户购买同一个商品的价格可以不一样。梯形价格的形成，可参见本章用户拼团流程。

3.2.4 用户拼团采购

用户通过微信公众号进入微商城后，即按分类、日期、关键字来检索商品，并进行商品的采购。为了方便移动客户的使用，商品的显示不采用传统的上一页，下一页的分页，而改为动态加载。即第一次显示 20 条记录，然后当用户浏览到最后一条时，再加异步加载 20 条，依次类推。

为了方便方便用户购买成功，进入商品明细页后，用户可以选择单独购买、开

团、附近的团三种方式。单独购买即按单独购买价，进行一次采购，与传统的采购过程一样；开团购买指的是本人作为“团长”，独立开一个新的团，然后再将该商品的购买链接发过去或共享到朋友圈，以供朋友圈的人参团，所有本人朋友圈参团购买的用户均属于本团。

附近的人或是附近的团，指的是通过查找已经存在的人或团来进行采购。附近的人，指的是当前本商品已经开团，但还没有拼团成功的，可以直接参团。附近的团指的是本商城中，当前还未结束，开团成功且销售量最大的团。供用户根据自己的需求，快速选择参与。

其操作流程如下图 3-3 所示。

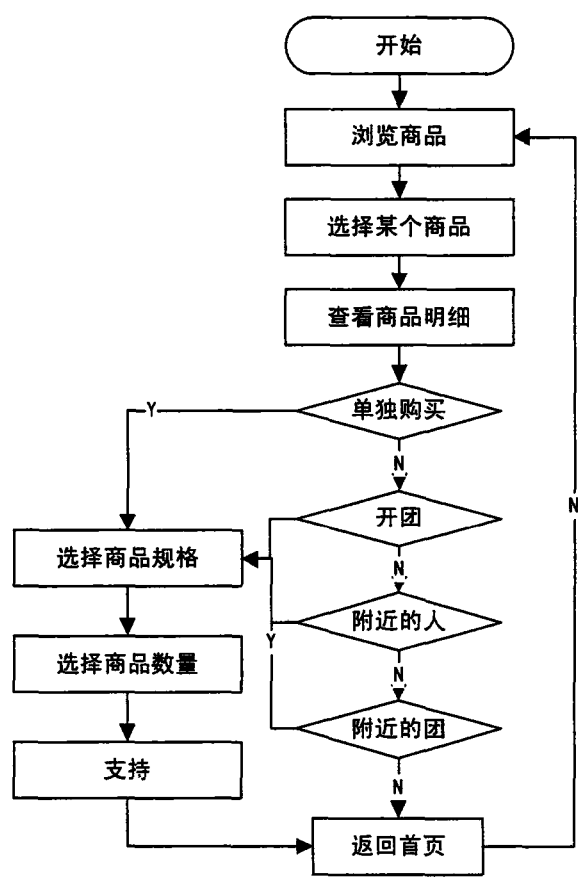


图 3-3 拼团购买流程图

3.2.5 订单管理

订单管理部分主要处理用户生成的订单，可以按区域、时间、手机号、订单不同

状态检索订单。其中订单的支付状态分为 5 类：待付款、已付款、付款失败、退款中、退款成功。订单的交易状态分为 3 类：交易中、交易成功、交易取消。订单支付状态转<sup>[36]</sup>化如图 3-4 所示。

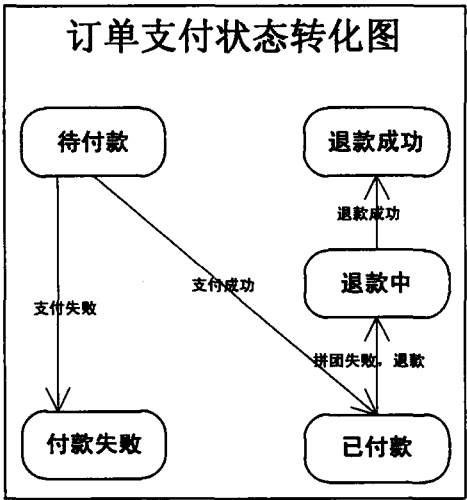


图 3-4 订单支付状态转化图

对于订单交易状态的变化，只要用户支付成功，则显示在交易中；当发货后，显示成功；若因为其他原因导致交易失败或中止时，显示为交易失败。交易失败一般由两个途径造成：第一个原因是用户与管理员协商取消订单。第二个原因是用户已经支付，但拼团失败。

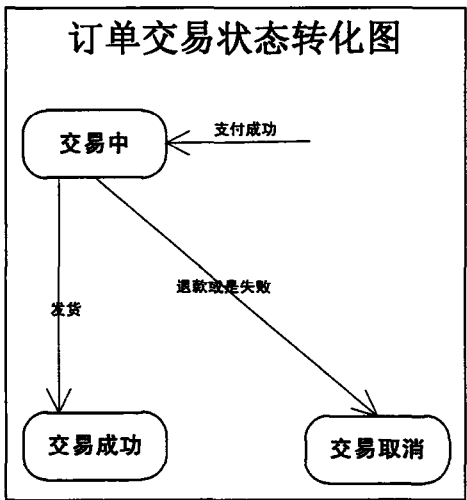


图 3-5 订单交易状态转化图

- 订单管理包括的主要功能有：查询、打印物流单、退款、微信通知、手动发货等。
- 查询就是按上面所述，多条件组合查询订单内容。
  - 打印物流单，就是根据不同物流公司的面单，将当前订单内容，打印到对应的位

置，以方便打印机直接打印，而不用手动去书写物流单。

- 退款。对于拼团失败，或是因为其他原因用户需要取消订单的，管理员可以通过后台直接退款。所退款项直接返还到其支付帐户里，退款成功后，向其微信里发信息通知。
- 微信通知。向用户定向发送微信内容。
- 手动发货。可以选中多个已经生成物流单事情的订单，可以批是为其货。
- 订单修改。点击某个订单，可以对其进行信息的完善与修改。对于发货成功的订单，可以跟踪其物流信息。

### 3.3 非功能性分析

为了保障系统的良好运行，除了其业务逻辑完善，实现预定的功能目标之外，还必须考虑其相关的潜在的非功能性需求，以保证系统在实际运维中方便管理。对于本微商城，其主要有以下几点潜在要求：

1. 性能需求。系统的运行必须快捷，界面流畅，给用户良好的体验。
2. 系统安全可靠。有关在线交易类的软件，首先任务是安全可靠。对于所有的业务往来记录操作日志，并且结合事务机制，保证其操作的完整性。绝不能允许一时成功，一时不成功的现象。
3. 易用性。界面操作简单，易懂，不需要进行用户培训。符合普通大众非专业人士的操作习惯。
4. 可维护性高。商务性的行为，根据市场的变化业务也会经常调整，软件必须做到易分析、易改变、易测试。
5. 方便扩展。鉴于微商城的特性，它本身就是基于第三开放平台二次开发的，部分功能是需要接口支持的。为了方便后期的功能扩展，相关的接口部分应利用文本配置方式，方便用户自行管理。

### 3.4 数据关系分析

良好的数据库系统，会大大提高系统的运行性能，降低数据维护维度。完整的数



数据库设计要提供 E-R 图和数据字典<sup>[37]</sup>。数据库的设计应尽量做到以下原则：

1. 原始单据与实体之间尽量保持一对一，特殊情况下，可以是一对多关系。
2. 所有的表一定要有主键。
3. 至少要满足 3NF。
4. 少建复杂表结构，多用 View 来替代。
5. 充分保证数据完整性。
6. 数据分开存储原则。一般来讲，一个表不推荐存放大量数据，超过百万条数据的表结构，一般按业务规则横向划分多个平等表结构。
7. 合理利用索引提高检索效率。
8. 多用触发器和存储过程来替代较复杂的 SQL 操作。

在本系统中，主要的数据库关系如下所述，ER 如图 3-11：

- 权限组具有多个不同操作功能。
- 一个用户可以属于多个权限组。
- 每个商品必须属于某个分类。
- 每个商品要有若干个属性，并且每个属性可具有若干个属性值。
- 每个商品可具有多个图片和多个附件。
- 一个发布的商品对应一个拼团规则。
- 一个拼团规则包含一个明细，指明当前商品的销售价、最低价，以及人数达到一定数量后，每增加一人，优惠多少。
- 提供两种类型的优惠券，一种是打折券，一种是现金抵用券，每个商品可以指明是否可以优惠券。
- 若商品支付时可使用优惠券，则按条件可能使用多张，记录下当前优惠券的券号。
- 每个新开团用户均生成一个唯一的团号，以后参与该团的用户均用该团号。
- 一个团可以具有多个用户，一个用户可以参与多个不同的团。
- 每个订单购买成功时，可以为其生成一个唯一的订单号，无论打印多次，该订单号都保持不变。



## 第 4 章 系统架构设计

本章研究系统的整体架构设计，以及物理部署方案。从总体上对系统进行架构设计，按 MVC 的思想，建立不同部件，以及明确各个不同部件之间接相关调用关系，达到高内聚低耦合的目的。力求在满足系统业务功能需求的同时，达到高可重复利用性，高灵活性，高可扩展性。

### 4.1 系统硬件体系结构

系统采用分布式服务器架构<sup>[38]</sup>，将静态页面、说明文档、帮助手册等放在 nginx 服务器上，而业务系统部署在 Web 服务器上。为了降低数据库压力，将数据库服务器与 Web 服务器分器、图片服务器分离。该方式是典型的商业运行模式，在数据量增大的同时，具有较大的扩容性。

分布式部署方案则将三者均分离部署在不同的服务器上，各服务器承担不同的职责，相互之间网络互联互通如图 4-1 所示。

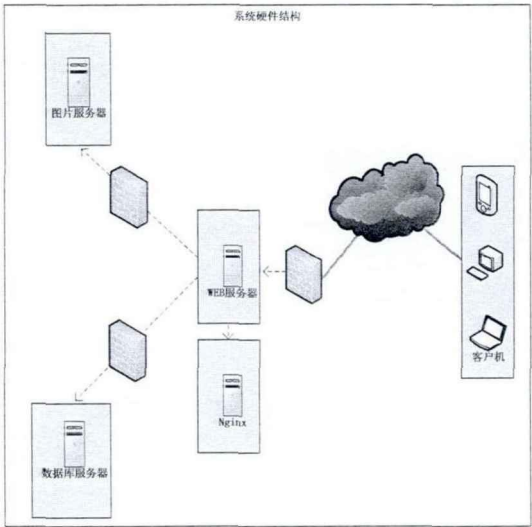


图 4-1 微信拼团商城服务器部署结构

### 4.2 系统架构设计

软件实现部分采用模块化划分思想，将界面与业务逻辑控制分离的前提下，实现

各模块的组件化。主要模块按功能需求完全分开，提取公用的抽象方法和接口。第三方组件的配置完全利用配置文件完成。

微信拼团商城共分为 4 个逻辑组成部分：

- basicFunc：用来存放基本的功能，如参数设置、公共数据维护等。
- client：用来存放微信端的内容。
- common：公共方法，系统服务器端。
- server：后台管理的方法。
- resources：存放相关资源文件，主要是配置文件。包括 spring 配置，hibernate 配置，memcached 配置，log4j 日志文件配置。

项目结构如图 4-2 所示：

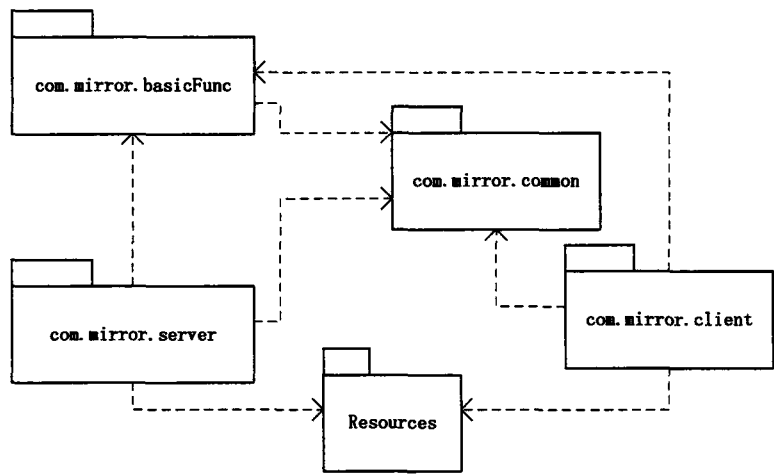


图 4-2 项目分层设计图

按照 MVC 的思想，将操作过程解耦分解为：数据访问层、业务层、实体层三个独立单元，三个单元通过消息传递实现松耦合调用。各组件功能划分清晰，调用关系明确，按逻辑思路划分如下：

- Model 层。存放 PO 对象，用来数据交互。
- Service 层。存放真正的业务实现方法，包括与数据库的交互，可以与 ORM 模式结合。
- Controller 层。包括业务方法，传递的数据是 Model 对象，数据的持久化等操作由 Service 完成。

下面以 Client 为例，描述按 MVC 模式进行项目划分布局，其包关系图如图 4-3 所示：

## Client结构

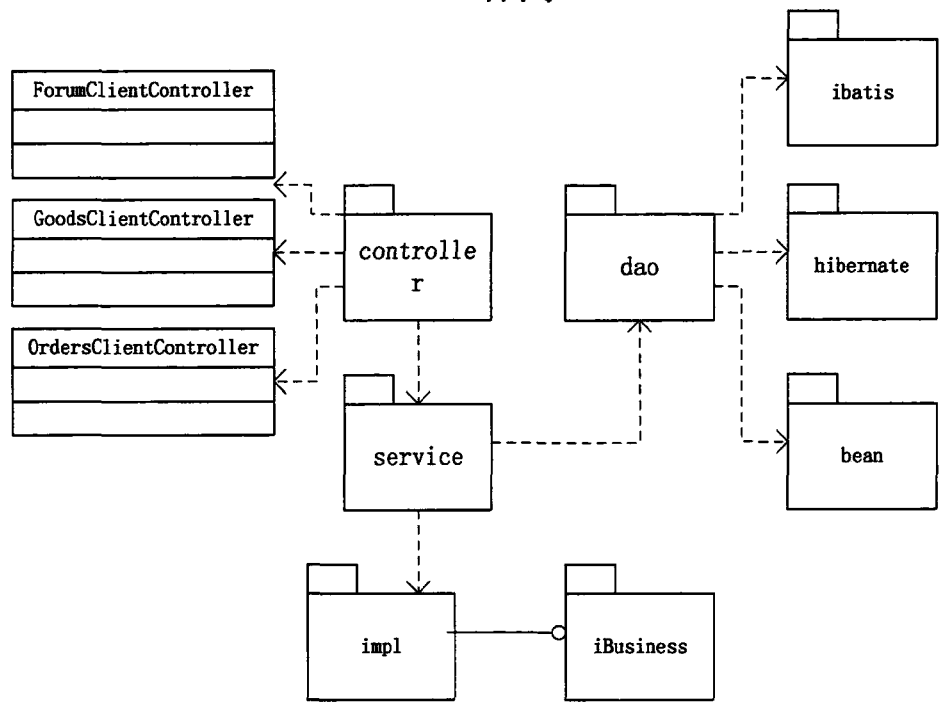


图 4-3 client 模块项目 MVC 结构

在上图中，**controller** 层提供业务方法，如：提供论坛管理、订单管理、商品管理等方法。其真正的数据修改、添加等功能，由 **service** 服务层来提供。本系统利用 **Hibernate** 实现数据持久化操作，故在 **service** 服务方法实现中调用 **dao** 的相关数据持久化方法。

为了便于业务的更新与升级，将所有的业务方法均封装为 **interface**，在 **controller** 中只是对接口的调用，而不用关心业务方法的具体实现细节。

在此对于数据库的操作提供两种 **ORM** 方式：**Hibernate** 与 **ibatis** 方式。**Hibernate** 方式适合简单 **SQL** 操作，**ibatis** 方式适合于复杂的 **SQL** 语句操作，特别是经常变动的 **SQL** 语句操作。**ibatis** 将复杂的 **SQL** 语句以配置文件的方式提供，在后期业务变动或是修改时，只需要对应的修改配置文件即可。不需要修改业务对象再重新编译。

### 4.3 网站 UI

站点的 **UI** 层主要存放用户界面。包括样式、图片、视频等。好的 **UI** 具有样式鲜明，操作简便，界面友好，响应快捷的特点。系统的 **UI** 分为两部分：后台管理员使用 **UI** 界面，以及前台微信用户访问 **UI** 界面。为了方便管理，将两部分主体内容进行分享存放。共享公用部分样式和脚本。同时，结合 **AngularJS** 的思想，每个页面对应一个事件控制

js 文件，其目录结构如图 4-4 所示。

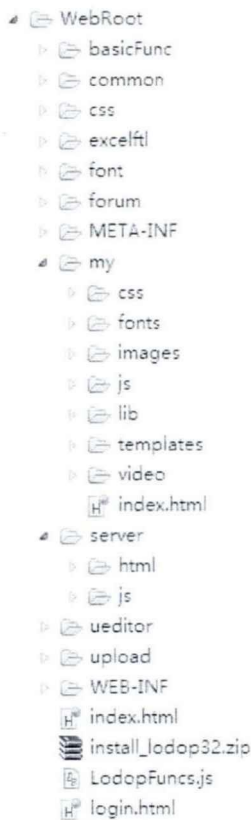


图 4-4 UI 部分项目结构

为了保证系统的安全性，将所有的页面文件存放到 WebRoot 根目录下，以防止用户的非法访问。各文件目录的功能与说明如下：

- basicFunc：存放基本页面，如版本信息、目录菜单等。
- common：系统出错文件、帮助使用手册等。
- css：公用样式等。
- excelftl:EXCEL 文件模版，借报表打印用。
- font：公用的字体。
- forum：存放论坛内容。
- my：存放微信端界面，为了方便修改，所有的微信端用户界面以模版方式提供，全部放到 template 目录下。
- server：存放管理员后台。
- ueditor：第三方百度富文本框。
- upload：存放用户上传资源。

- WEB-INF：资源目录和主配置文件。

## 4.4 AngularJS 指令设计

AngularJS 最主要的特点是“双向绑定”，即用户界面的操作结果，可以及时的反应给用户，而不需要再进行用户 JS 动作判断。该方式极大的增强了用户体验，减化界面控制逻辑。为了达到更好的控制效果，需要自定义指令行为，即当某事件发生时，界面如何做出相应的变化。其 directive 指令具有如下特点：

- 1、预翻译，运行效率高。
- 2、语法完善，方便扩展。
- 3、跨浏览器支持好。
- 4、支持跨域的数据绑定，呈现方式灵活多样。

系统设计过程中，根据其不同，规划设计以下几类指令：

1. Controller：存放所有的系统业务逻辑。
2. Directive：定义模版渲染效果动作，以及定时器、模版加载事件。
3. Filtler：数据查询显示的过滤动作指令。
4. Main：脚本的加载指令，某些文件采用异步加载。
5. Service：存放业务方法，如商品的展示、支付、用户中心等。
6. Slide：控制移动设备的上下移动动作事件。
7. 模版：每一个模版页面对应一个业务方法指令，以处理页面上的数据加载和动作。

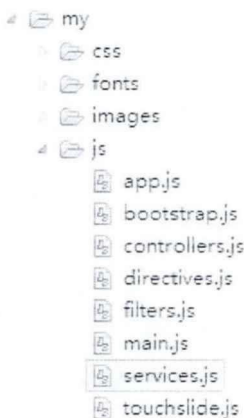


图 4-5AngularJS 指令文件

## 4.5 系统性能优化

为了提高系统的响应速度，利用 Memcached 机制<sup>[38]</sup>进行数据缓存。对于业务数据库中的表结构，采用分表查询方式。

### 4.5.1 分布式内存方式

Memcached 机制可以将数据缓存进内存，但其与直接的 Cache 不一样。Memcached 最大的优点是可能支持后期的分布式扩展。比如：当系统的用户业务量达到某个阈值，单台服务台已经不好处理时，这时就可以考虑进行分布式解决方案，而 Memcached 机制可以支持分布式内存。系统设计中，将初始的基础数据，以及用户初始访问的商品列表缓存。

Memcached 的优点主要有如下几点：

- 使用预申请和分组的方式管理内存空间，而不是每次都需要 new 一个内存块 slab。
- 自动内存释放。当一个 slab 不被引用时，就自动释放该块。
- 利用性能更好的 newHash 算法。
- Memcached 是一个独立的服务程序，可以结合其他框架进行灵活配置。

### 4.5.2 分表方案

对于系统中的业务数据，商品数据和订单数据，将其分表存储。商务系统的数据量会随着时间急剧膨胀，会严重影响后期的检索性能。系统预期 2 个月后的用户量为 20 万左右，半年后的关注用户达 50 万。按照平常一天 1% 的用户下单，则每天会产生几千个订单。对于订单表的数据维护将会是一个难题。

系统设计采用分表方案，即将不同的业务数据，进入不同的业务系统。首先，将订单数据根据其不同状态，插入不同的表。主要的业务状态有：交易中，交易成功，交易失败。建立三个订单业务表：交易中订单，交易成功订单，交易失败订单。三个订单中数据利用触发器机制，保持数据的一致性。

对于交易成功和交易失败的订单数据，是后期业务统计热点，需要进行频繁的统计



汇总。按年份，将同一类数据以年为单位再进一步进行横向切割，保证每次系统运行过程中，查询到的数据都只是一小部分。

划分之后的订单数据结构如图 4-6 所示：

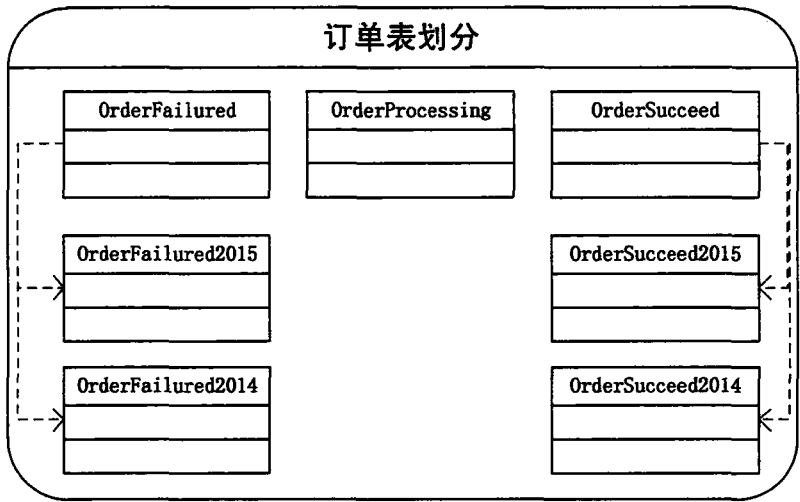


图 4-6 订单划分结构

其中：

- OrderFailed 表示交易失败订单。  
OrderFailed2014 表示 2014 年失败订单记录，按日期索引排序。  
OrderFailed2015 表示 2015 年失败订单记录，按日期索引排序。  
依次类推，不同年份会有不同的该表。
- OrderProcessing 表示正在交易中订单。
- OrderFSucceed 表示交易成功订单。  
OrderFSucceed2014 表示 2014 年成功订单记录，按日期索引排序。  
OrderFSucceed2015 表示 2015 年成功订单记录，按日期索引排序。  
依次类推，不同年份会有不同的该表。

4.6 本章小结

本章按项目特点，进行系统框架设计，以设计符合移动端使用的微商城。系统设计的思想按照 MVC 的思想，将各层进行分离。将各层之间的交互尽量利用配置文件来实现，配置文件利用 xml 标准文件进行存放。

为了提供系统的性能，特别结合了当前流行的 Memcached 机制，将常用的数据加载入内存中，以减少 IO 操作。对于数据库的优化方面，采用索引机制和分表机制，来实现查询的最优化。

## 第 5 章 系统实现

本章以系统的例心功能为例，从 UI 界面的设计，到核心功能的实现，描述项目的具体实现过程。

### 5.1 开发环境与技术

由于微商城的用户访问量会很大，一个成熟的微商城的关注用户量会在 50 万以上。且拼团价格计算逻辑很复杂，为保证系统高效稳定的运行，特选择 Centos [39]作为服务器。Centos 开源系统具有稳定性高、集成性好、费用便宜等特点，系统环境如下所示。

- 应用服务器环境：Centos6.5 + Tomcat7.0 + MySQL5.6。
- 系统开发工具：MyEclipse10.0 +Spring 4 + Tomcat7.0 + MySQL5.6+ AngularJS。
- 其他工具：jQuery1.8，Lodop 插件，Adobe PDF ActiveX 插件 11.0 以上。

### 5.2 微信公众号配置

为了将服务器与微信公众平台对接，需要同时对微信公众平台开发者中心进行配置，同时，对服务器中系统中的配置文件进行修改，设置其微信连接的应用程序接口和钥匙。

微信公众号配置如下图 5-1 所示：



图 5-1 微信公众平台配置参数

在后台应用程序中心也需要进行配置，本系统将所有的配置信息存放在 config.properties 文件中，并在 resources/spring/applicationContext-core.xml 中进行引用，以使 Spring 应用 config 中的相关配置参数。

### 1. 配置 config.properties

#### #配置数据源

```
database.driver = com.mysql.jdbc.Driver
hibernate.dialect=org.hibernate.dialect.MySQLDialect
database.url = jdbc:mysql://localhost:3306/ptmalldb
database.username = root
database.password = 123456
database.port = 3306
database.serverName = vmards001.mysql.rds.aliyuncs.com
```

#### #配置微信 APPID

```
weixin_app_id=wx3521d2f9c7335af4
weixin_app_secret=570ab5fe2134cd9c974191007d4e68bd
weixin_mch_id=1254370701
weixin_app_id=wx543e906d79c257ae
weixin_app_secret=1f48584d09e6882dc93e5d9eeb9d37d0
weixin_mch_id=1245445502
weixin_redirect_url=http://www.pintuan-mall.com/my/weixinjsclient/oauthResponse
weixin_token=520cca815cb55ae48e147b1dc56c0ee2
weixin_encodingAESKey=rNrWgPhjn4TMawB4gfpo4t4AU3Cicw8vfvauu9MouRH
weixin_home_path=http://www.pintuan-mall.com/my/my/index.html
weixin_notify_url=http://www.pintuan-mall.com/my/weixinjsclient/notify
weixin_pay_url=http://www.pintuan-mall.com/my/
```

### 2. 在 applicationContext-core.xml 中添加 beans 下的 bean 配置结点:

<!-- 加载数据库配置文件 -->

```

<context:property-placeholder
location="classpath:resources/config.properties" />
<!-- 微信基础配置 -->
<bean id="config" class="com.mirror.basicFunc.vo.Config">
    <property name="weixinAppId" value="\${weixin_app_id}"/>
    <property name="weixinAppSecret" value="\${weixin_app_secret}"/>
    <property name="redirectUrl" value="\${weixin_redirect_url}"/></property>
    <property name="weixinToken" value="\${weixin_token}"/></property>
    <property
                                name="weixinEncodingAESKey"
value="\${weixin_encodingAESKey}"/></property>
    <property name="weixinHomePath" value="\${weixin_home_path}"/></property>
    <property name="weixinMchId" value="\${weixin_mch_id}"/></property>
    <property name="weixinNotifyUrl" value="\${weixin_notify_url}"/></property>
    <property name="weixinPayUrl" value="\${weixin_pay_url}"/></property>
</bean>

```

### 5.3 AngularJS 指令

AngularJS 的指令动作分为 7 大模块（参见 4.4），分别来用存放不同的业务动作。下面展示 main 的配置，主要描述了如何去加载项目 js 资源：

```

require.config({
    //配置 AngularJS 的路径
    paths:{
        "AngularJS":"../lib/ionic/js/AngularJS/AngularJS.min",
        "AngularJS-ui-router":"../lib/ionic/js/AngularJS-ui/AngularJS-ui-router.min",
        "jquery":"../lib/jquery/jquery.min",
        "app":"../js/app",
        "controllers":"../js/controllers",
        "directives":"../js/directives",
        "filters":"../js/filters",

```

```
"services":"../js/services",

"touchslide":"../lib/jquery/touchslide",

"classie":"../lib/jquery/classie",

"sidebarEffects":"../lib/jquery/sidebarEffects",

"bootstrap":"../js/bootstrap",

"ngScroll":"../lib/ng-infinite-scroll/ng-infinite-scroll.min",

"jquery.lazyload":"../lib/jquery/jquery.lazyload.min"

},
```

5.4 商品管理

后台商品登录成功之后，商品列表如图 5-2 所示。

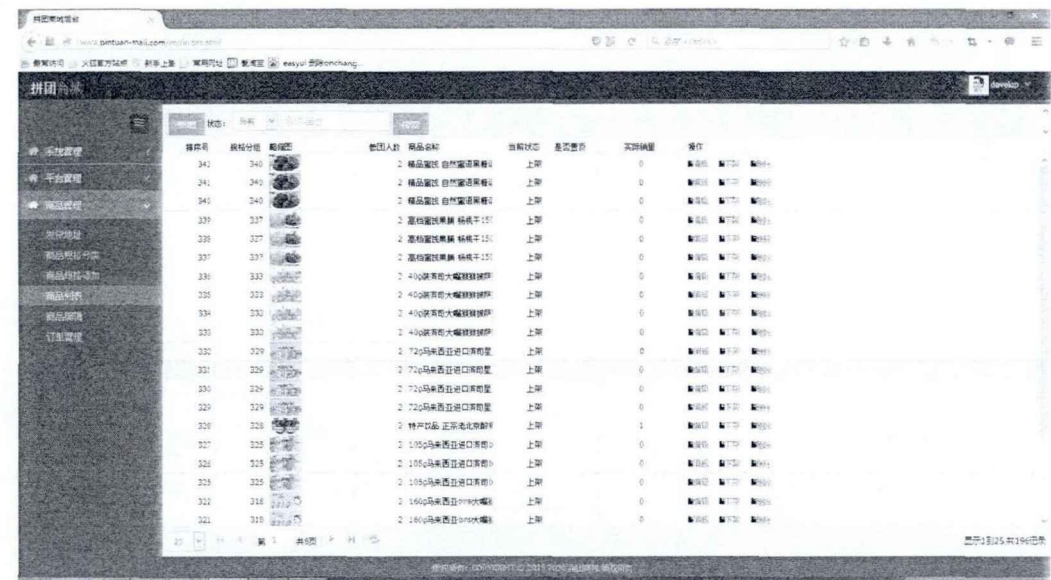


图 5-2 商品列表

商品的列表文件 WebRoot/server/html/goodsList.html 主要 UI 呈现，其页面对应的行为（数据查询、分页显示、数据绑定）存放在 WebRoot/server/js/goodsList.js 中，利用 AngularJS 来控制其行为。

goodsList.html 的主体为一个 table, 初始为空，在呈现时，通过 AngularJS 的数据绑定机制，动态加载数据列表：

```
<!DOCTYPE>

<html>

<!--头部：存放样式以及页面属性说明 -->
```

```

<body>
  <script type="text/javascript" src="server/js/goodsList.js"></script>
  <div id="cc" class="easyui-layout" fit="true">
    <div region="north" split="false" style="height:50px;" border="false" >
      <div class="toolbar-bg" style="height: none;">
        <table style="width: 100%; text-align: left; vertical-align: middle;">
          <tr>
            <td>
              <button type="button" class="btn blue"
                onclick='GoodsList.add();'>新增</button>
              状态:
              <select class="span1" id="search_state">
                <option value="">所有</option>
                <option value="1">上架</option>
                <option value="2">下架</option>
              </select>
              <input class="span2" type="text" placeholder="名称,描述"
                id="searchVal" />
              <button type="button" class="btn blue"
                onclick='GoodsList.reload();'>搜索</button>
            </td>
          </tr>
        </table>
      </div>
    </div>
    <div region="center" style="padding:5px;" >
      <table id="orderTable">
      </table>
    </div>
  </div>
</body>
</html>

```

```
/*表格定义*/
```

38



39

```
        frozenColumns : true,
        pageList : [ 25, 30, 40 ,50],
        loadMsg : "正在加载数据,请稍后...",
        columns : header,
        onDbClickRow:function(rowIndex, rowData){
            GoodsList.updateSort(rowIndex);
        }
    });
});

});

/*页面加载时,调用页面初始化,完成数据绑定*/
$(function(){
    editrow = null;
    GoodsList.init();
});
```

5.5 订单管理

管理员进入后台后,可进行订单维护。订单维护的主界面如图 5-3 所示,选择“中通快递”并打印单,生成的打印单如图 5-4 所示。该打印过程调用了 Lodop 控件,利用其图形打印特性,算准位置,将相关的数据“画”到图片上。若要查看订单明细或是跟踪物流信息,可点击“详情”,可查看到当前订单的明细以及物流信息,如图 5-5 所示。

图 5-3 订单列表



图 5-4 物流单打印

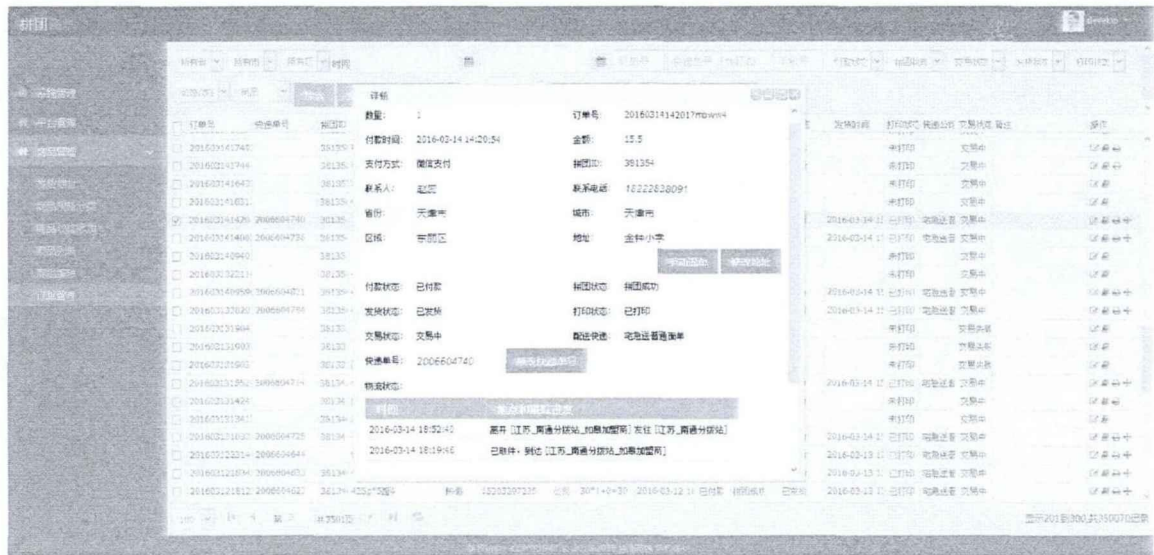


图 5-5 订单明细

## 5.6 微信用户商品采购

对于关注了本微信号的用户，可以进入当前的拼团商城。如图 5-6 所示。用户点某个商品，则跳转到商品的明细查看界面，同时可以支付如图 5-7 所示。支付成功之后的内容，可以转发到朋友圈。



图 5-6 商品明细



图 5-7 商品支付

微信支付先要用户申请开通其微信支付接口，支付时先预支付生成其支付签字，然后由用户确认一次后再真实的完成支付，其核心业务代码如下。

```
public static Map<String, Object> wxPay(Map<String, Object> param, Config config) {  
    Map<String, Object> returnResult=new HashMap<String, Object>();  
    UnifiedorderDto unifiedOrder=new UnifiedorderDto();
```

```
Map<String,String> signMap=new HashMap<String, String>();
//公众账号 appid, mch_id, nonce_str, body, orderNo
unifiedOrder.setAppid(config.getWeixinAppId());
signMap.put("appid", config.getWeixinAppId());
//总金额 total_fee
BigDecimal bd=new BigDecimal(param.get("price").toString());
Double price=bd.doubleValue()*100;
Long tempPrice=Math.round(price);
unifiedOrder.setTotal_fee(tempPrice.toString());
signMap.put("total_fee", tempPrice.toString());
//终端 IP, 通知地址, 交易类型, 用户标识 openId
unifiedOrder.setOpenid(param.get("openId").toString());
signMap.put("openid", param.get("openId").toString());
//签名
unifiedOrder.setSign(WXSignUtil.sign(signMap, config));
try{
    Map<String,String> result=WXPAYUtil.unifiedorderMap(unifiedOrder);
    if(result.get("result_code").equals("SUCCESS")
        &&result.get("return_code").equals("SUCCESS")){
        //返回成功 json 字符串
    }else{
        return Result.put("flag", true);
    }
}catch(Exception e){
    log.info("支付出错啦:", e);
}
return returnResult;
}
```



## 5.7 Memcached 配置

为了加进进一步优化系统性能，本系统利用 Memcache 来提高系统响应性能。Spring + Memcached 可以完成的集成，相关配置过程如下：

### 1. 创建配置 spring-memcached.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans .....>
<bean id="memcachedPool" class="com.mirror.MemCached.MyPool"
    factory-method="getInstance"
    init-method="initialize"
    destroy-method="shutDown">
    <constructor-arg>
        <value>myMemPool</value>
    </constructor-arg>
    <property name="servers">
        <list>
            <value>${memcache.server}</value>
        </list>
    </property>
<!-- 类似配置参数:
initConn: 最大连接数, 缺省为 30;
minConn: 最小连接数值, 缺省为 10;
maxConn: 最大连接数值, 缺省为 50, 根据并发量可以调整;
maintSleep: 主线程睡眠时间, 单位为毫秒, 缺省设置为 3000;
nagle: 是否启用 nagle 算法,
socketT0: 连接超时时间, 单位为毫秒, 缺省为 3000;
-->
</bean>
```

```

<bean id="memcachedClient" class="com.mirror.MemCached.MyMemlient">
    <constructor-arg>
        <value> myMemPool</value>
    </constructor-arg>
</bean>
</beans>

```

2. applicationContext-common.xml 配置文件，添加如下配置：

```

<bean
    class="org.springframework.beans.factory.config.PropertyPlaceholderConfir
gurer">
    <property name="locations">
        <list>
            <value>classpath: myMemCached.properties</value>
        </list>
    </property>
</bean>

```

3. Web.xml 文件中配置新建的文件

```

<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value> classpath:/spring/spring-memcached.xml</param-value>
</context-param>

```

5. myMemCached.properties 配置文件，如第 2 步中的文件名称对应：

```

mcache.server=local\12001
memcache.initConn=30
memcache.minConn=10
memcache.maxConn=50
memcache.maintSleep=3000
memcache.nagle=false

```

```
memcache.socketTO=3000
```

## 5.8 本章小结

本章在第四章系统架构设计基础上，以核心功能为例，叙述了微信拼团商城的设计实现以及运行效果。实现的过程突出了分层思想的体现，代码的规范化，以及微信编程技术的应用。为了达到更好的性能，也引入了log4j实现日志记录，memcached进行分布式内在管理。

本系统通过严格的单元测试、集成测试、用户体验测试三个阶段，运行平稳，目前已在正式投入使用，并且已有关注用户30多万，每天生成的订单达1000条。



## 第6章 总结与展望

### 6.1 总 结

本微信拼团商城的功能灵活，操作模式新颖，能极大的吸引微信用户进行拼团商品采购。同时结合先进的开发技术，界面运行流畅，功能扩展方便。系统的分析设计严格按照软件工程规范标准，数据设计符合国家电子商务数据安全规范。本课题主要完成了如下工作：

对微信拼团商城进行业务分析，分别调研各类用户，提炼最佳商业模式，结合微信公众平台，研发出一套完整的商业系统，并投入使用。

参考各优秀的国内外技术解决方案，在实现高可扩展性、高灵活性的同时，考虑其响应时间和用户操作体验，采用AngularJS技术操作前台UI呈现，更好实现用户事件交互。同时，结合MVC思想、抽象工厂模式和Memcached缓存技术，使网站的维护灵活，运行速度快捷。

对于网站系统中很重要的信息安全性问题，在利用硬件防火墙过滤的同时，制定了严格的访问控制机制，可以减少恶意的DDOS访问攻击。

### 6.2 展 望

由于商业模式的不断变革，本微信拼团商城新增的部分业务功能还有待完善研究，比如：拉人入团、购买赠送、并单等功能还未找到完美的解决方案。系统的稳定性上也还存在一定缺陷，系统在运行10天左右服务器资源会出现枯竭现象。主要还要进一步研究的方面有：

#### 1、图形化订单打印模版设计。

订单模版的打印目前是根据各快递公司不同的单子，依次计算其每个数据项的位置然后打印的。该位置计算过程相当麻烦，容易产生偏差，要是能做成图形化位置拖放就更完美了。

#### 2、查询优化。

目前数据库的查询优化是结合索引来完成，当用户数据达到40万时，每次分页显示

的查询要5秒左右，后期可以考虑分布式数据存储模式来提高数据并发量。

### 3. 页面分页加载。

现在进入商城时，缺省加载50条最新的商品信息。后期可改为异步加载模式，先加载3屏数据，然后当屏幕移到最下面时自动加载20条，依次类推。异步加载可以大大加快页面的打开速度，提供更好的用户体验。

## 参考文献

- [1] 孔秀丽. 基于微信公众平台的大学生移动商城的构建 [J]. 科技创业月刊, 2015(3) :25-26.
- [2] 倪懿. SoLoMo 模式下基于微信公众平台构建图书馆移动服务系统的研究[J]. 情报科学, 2016(1) :12-13.
- [3] 杨晨. 浅析基于微信的新型营销模式 [J]. 现代工业经济和信息化, 2014(16) :41-42
- [4] 黄冰,熊昌昱. 基于微信公众平台开发的校园微信助手[J]. 信息通信, 2015(16):13-14.
- [5] 朱义. Eclipse 实用教程[M]. 北京:电子工业出版社,2011:89-126.
- [6] 肖群. 基于 RBAC 的精品课程网站权限设计 [J]. 智能计算机与应用,2016(1):4-6.
- [7] 何光旭. 基于 jQuery 框架的医院 HIS 系统页面前端特效设计 [J]. 电子设计工程,2015(4):21.
- [8] 高昂. MVC 设计模式研究[J]. 电脑知识与技术,2016(2):32-33.
- [9] Deaders. Computer-based Creativity Enhanced Conceptual Design Model for Non-routine Design of Mechanical Systems [J]. Chinese Journal of Mechanical Engineering,2014(6):8-9.
- [10] 许娜. Network topology optimization of supply chain information systems [J]. Journal of Beijing Institute of Technology, 2015(7):28-29.
- [11] 刘宇宏. 设计模式在软件项目中的应用[J]. 计算机软件,2015(8):43-45.
- [12] 吴云芳. 基于 B/S 结构的高校毕业设计选题系统的设计与实现 [J]. 大学教育,2016(1):42.
- [13] 李文. 基于 SSH 架构的在线考试系统的研究[J]. 武汉大学学报,2014(12):53-54.
- [14] 张大亮. 网络攻击技术研究[J]. 武汉大学学报,2015(4):38.
- [15] 陆远. UML 的流程建模与软件测试[D]. 武汉大学,2015:12-13.
- [16] 刘华利. Java Web 中数据库访问技术的研究[J]. 计算机工程,2015(3):46-47.
- [17] 李容. 基于 MVC 模式的 WEB 应用研究[J]. 计算机应用与软件,2015(10):12-13.

- [18]顾小升. 软件体系结构[D]. 南京大学,2015:27-31.
- [19]倡腾飞,基于 RUP 的轻量级软件过程实践[D]. 北京:北京交通大学,2013:20-22.
- [20]谢琪林. 网上阅卷系统设计与实现[D]. 四川:电子科技大学,2011:11-12.
- [21]颜兆林. 系统安全性分析技术研究[D]. 长沙:中国人民解放军国防科学技术大学,2001:21-26.
- [22]杨瑞龙,朱征宇. 引入软件设计模式的面向对象程序设计教学方法[J]. 软件学报,2015(10):26-27.
- [23]陈端芝. 基于 JSF+Spring+Hibernate 的工程质量管理体系设计与实现[J]. 西安交通大学学报,2015(10):33-35.
- [24]马静. 基于 TAD 的电信定单处理子系统的设计[D]. 石家庄:河北科技大学,2011:12-15.
- [25]胡家镜. 旅游移动电子商务网站内容交付性比较[J]. 计算机工程与设计,2015(10):47-49.
- [26]张大新. B2C 电子商务网站设计研究[D]. 上海:华东理工大学,2013:13-14.
- [27]黄缙华. MySQL 入门与提高 [M]. 北京:清华大学出版社,2015:122-124.
- [28]林满佳,唐屹. OAuth 授权流程的安全建模研究 [J]. 广州大学学报,2015(3):14-16.
- [29]张天琪. OAuth 协议安全性研究 [J]. 信息安全,2013(3),44-45.
- [30]徐礼金. 基于 J2EE 框架的微信拼团商城分析与设计[硕士学位论文]. 广州:华南理工大学,2011,20-28.
- [31]高静. JSON 数据传输效率研究[J]. 2011,第 63 期,23-25.
- [32]冯晨超. 基于 AngularJS 的物品和用户查询模块的设计与实现 [硕士学位论文]. 南京大学,2014:37-28.
- [33]董英茹. 简谈 AngularJS 在下一代 Web 开发中的应用[J]. 软件工程,2015(5):29-30.
- [34]相晨萌. 基于 j2ee 多层架构的档案管理信息系统的开发[J]. 2011,第 35 期,16-17.
- [35]刘博. 在线预约挂号系统设计与实现 [J]. 中国科学院大学 2014: 37-39.
- [36]黄有福. Jsp WebStorage API 会话机制探索[J]. 2013,第 39 期,4-5.

- [37] 李晓玲. 电子商城系统的设计与实现[D]. 成都: 电子科技大学, 2013, 16-21.
- [38] 李洁. 南京机电公司电子商城系统的设计与实现[D]. 成都: 电子科技大学, 2013, 42-47.
- [39] Jonathan Hobson. CentOS 6 Linux Server Cookbook [M]. Packet Publishing, 2016: 60-125.

## 致 谢

感谢我的导师张家晨、陈伟老师，在百忙之中结合了我工作的情况，为我认真选题，分析论文中存在的问题，反反复复仔细地修改论文。他们严谨的作风，深厚的造诣，深深地打动了我，我一定铭记在心。在论文完成之际，谨向导师致以最崇高的敬意和由衷的感谢！

衷心感谢吉林大学的专家教授了我丰富的专业知识。是你们教育了我为人处事的原则和解决问题的方法！同时也感谢与我一起学习的同学们，是你们让我感受到学习知识的快乐！

感谢我的领导和同事，是你们给了我学习深造的机会，给予了我莫大的支持和关心，使我有更多的时间在教学、科研上有所突破。感谢我的家人这三年来为我默默的奉献，是你们为我提供了生活上的帮助，使我有更多的精力投入到研究中去，从而让我的科研水平好技术水平得到了提升。

感谢论文审稿专家和答辩专家付出了辛苦的劳动，为论文改进提出了宝贵的意见。

感谢那些默默关心支持我的人，祝你们健康幸福！