

# 中国计量大学

## 本科毕业设计（论文）

基于 Dubbo 的分布式电子商务平台应用

**Distributed e-commerce platform based on  
Dubbo**

学生姓名 孟仁梓 学号 1401400118

学生专业 计算机科学与技术 班级 14 工试 1 班

二级学院 量新学院 指导教师 陶海军

中国计量大学  
2018 年 5 月

# 郑 重 声 明

本人呈交的毕业设计论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片和资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

学生签名：\_\_\_\_\_

日期：\_\_\_\_\_

分类号: TP3  
UDC: 62

密 级: 公开  
学校代码: 10356

# 中国计量大学 本科毕业设计（论文）

基于 Dubbo 的分布式电子商务平台应用

**Distributed e-commerce platform based  
on Dubbo**

作 者 孟仁梓  
申请学位 工学学士  
学科专业 计算机科学与技术  
答辩委员会主席

学 号 1401400118  
指导教师 陶海军  
培养单位 中国计量大学  
评 阅 人

2018 年 5 月

## 致 谢

本论文能够顺利完成要大力感谢陶海军老师在百忙中抽出时间给予指导和帮助。另外，也要感谢给予转载权和引用权的资料、图片、文献、研究思想的所有者，还有感谢中国计量大学对我的培养。最后，向这些给予我帮助的老师，同学和组织致以衷心的感谢。

孟仁梓  
2018.5.14

## 基于 Dubbo 的分布式电子商务平台应用

**摘要：**自 21 世纪以来，随着个人电脑、智能手机的普及，互联网产业开始迅速发展，网站的访问量呈指数型爆炸增长，对并发访问技术提出了更高的要求，例如，2016 年 11 月 11 日，“淘宝网”的并发访问量高达 12 万笔每秒，适应高并发的网络架构亟需建立，因此，大型网站架构为适应并发需求，从单一应用架构演进为垂直应用架构，最后演进为面向服务（SOA）的分布式应用架构，SOA 体系的不同实现方式有 SOAP、RPC、REST 等，本文采用的 Dubbo 服务框架正是基于 RPC 技术的 SOA 体系中的典型框架。

本文基于 Dubbo 服务框架打造电子商务平台应用，按照软件开发基本步骤，进行了需求分析、系统设计、系统实现，最终构建了分布式电子商务平台应用，论文按照以下工作步骤进行展开：

1. 需求分析：此阶段针对目标客户，进行需求分析，确定技术选型和完成可行性分析。

2. 系统设计：此阶段完成了系统的整体架构的设计，如何利用 Dubbo 框架打造分布式系统，最后完成了数据库的设计。

3. 系统实现：此阶段实现了系统设计中需求的功能。

基于 Dubbo 的分布式电子商务平台应用的搭建，在用户层面，丰富了大众购物的选择，在商家层面，为商家带来了更多的销售机会。

**关键词：** Dubbo、分布式、电子商务

**中图分类号：** G633.67

## **Distributed e-commerce platform based on Dubbo**

**Abstract:** Since the 21st century, with the popularity of personal computers, smart phones, the Internet industry rapid development, web site traffic exponential type explosive growth, puts forward higher requirements on concurrent access to technology, for example, on November 11, 2016, "Taobao" concurrent traffic is as high as 120000 transactions per second, to adapt to the high concurrency network architecture needs to be established, therefore, large website structure, meeting the needs of concurrent, evolution from a single application architecture for application of vertical structure, the evolution of distributed application architecture, SOA system, different ways of implementation of SOA system such as SOAP, RPC, REST, The Dubbo service framework adopted in this paper is a typical framework in the SOA system based on RPC technology.

Based on the Dubbo service framework to build e-commerce application platform, according to the basic steps in software development, the demand analysis, system design, system implementation, eventually to build a distributed e-commerce platform, paper work according to the following steps:

1. Demand analysis: this stage is aimed at the target customers, conducts demand analysis, determines the technology selection and completes the feasibility analysis.
2. System design: this phase completes the design of the overall architecture of the system, how to build a distributed system using the Dubbo framework, and finally completes the design of the database.
3. System realization: this stage realizes the function of requirements in system design.

Based on the application of Dubbo's distributed e-commerce platform, it has enriched the choice of shopping at the user level and brought more sales opportunities to merchants at the merchant level.

**Keywords:** Dubbo、Distributed、Electronic commerce

**Classification:** G633.67

# 目录

摘要 .....	5
Abstract.....	6
第一章 绪论 .....	9
1.1 研究背景 .....	9
1.2 国内外研究现状.....	9
1.2.1 单一应用架构.....	9
1.2.2 垂直应用架构.....	10
1.2.3 分布式应用架构.....	11
1.2.4 SOA 架构.....	12
1.2.5 远程过程调用.....	12
1.3 论文组织架构.....	13
第二章 系统需求分析.....	14
2.1 可行性分析 .....	14
2.1.1 技术可行性.....	14
2.1.2 经济可行性.....	14
2.2 功能需求分析.....	14
2.2.1 并发访问需求.....	14
2.2.2 可扩展性需求.....	14
2.2.3 角色分析.....	15
2.2.4 功能分析.....	15
2.2.5 用例图.....	16
2.3 业务流程 .....	17
第三章 系统概要设计.....	18
3.1 系统主体设计.....	18
3.2 基于 Dubbo + Zookeeper 部署架构 .....	18
3.2.1 部署架构设计.....	18
3.2.2 Dubbo 工作框架流程介绍.....	19
3.2.3 基于 Dubbo + Zookeeper 部署的意义.....	20
3.3 系统整体流程.....	21
3.4 数据库设计 .....	21
3.4.1 用户表 (mmall_user) .....	21
3.4.2 分类表 (mmall_category) .....	22
3.4.3 产品表 (mmall_product) .....	22
3.4.4 购物车表 (mmall_cart) .....	23
3.4.5 支付信息表 (mmall_pay_info) .....	23
3.4.6 订单表 (mmall_order) .....	24
3.4.7 订单明细表 (mmall_order_item) .....	25
3.4.8 收货地址表 (mmall_shipping) .....	25
3.5 系统软件列表.....	26
第四章 系统实现 .....	27
4.1 开发环境 .....	27
4.2 系统的实现 .....	27
4.2.1 分布式框架的搭建.....	27
4.2.2 用户注册.....	28
4.2.3 用户登录.....	29
4.2.4 下订单.....	30
4.2.5 网站展示图.....	32
第五章 系统测试 .....	33

5.1 系统功能测试.....	33
5.1.1 用户注册功能测试.....	33
5.1.2 购物功能测试.....	34
5.1.3 后台管理功能测试.....	34
5.2 总结 .....	35
第六章 总结与展望 .....	36
6.1 总结 .....	36
6.2 展望 .....	36
第七章 参考文献 .....	37



# 第一章绪论

## 1.1 研究背景

据相关报告显示，电子商务所占传统零售消费市场份额逐年显著增长，而相比之下，传统的零售销售正在持续下降，自 2007 年以来，电子商务产业持续增长，仅 2013 年第一季度就完成了 8% 的增长<sup>[1]</sup>。伴随着高速发展的电子商务产业，大型网站的访问量呈指数性激增，大型网站的高访问量对于公司、用户和开发人员既是机遇又是挑战。

在互联网公司层面，更大的用户量意味着更大的流量，而巨大的流量意味着诱人的利益，抓住电商发展即走在了互联网的快车道，这是属于所有互联网公司的机遇，而巨大的利益面前，会导致更残酷的竞争，技术落后的公司会被淘汰出局，这是互联网公司所面临的挑战。

在用户层面，电子商务为大众购物带来了更加便捷的购物体验、更加丰富的购物选择，网上购物也逐渐成为了越来越多人的选择。

在技术人员层面，高聚合、低耦合、便于维护的系统是值得技术人员去了解学习，技术人员跟上时代的步伐才不至于被时代所遗弃。

综上所述，电子商务网站是互联网中最普及的应用之一，而分布式架构<sup>[2]</sup>则是未来发展的趋势，基于 Dubbo + Zookeeper<sup>[3]</sup>架构的分布式电子商务网站是富有极大研究意义的。

## 1.2 国内外研究现状

随着当今网站对并发量要求越来越高，大型网站架构从单一应用架构演进为垂直应用架构，最后演进为分布式应用架构<sup>[4]</sup>和 SOA 体系。四大架构情况如下所示：

### 1.2.1 单一应用架构

传统的单一应用架构如图 1-1 所示，将服务全部运行一个应用服务器之中，所有用户访问任何服务都通过单一的路径进行。

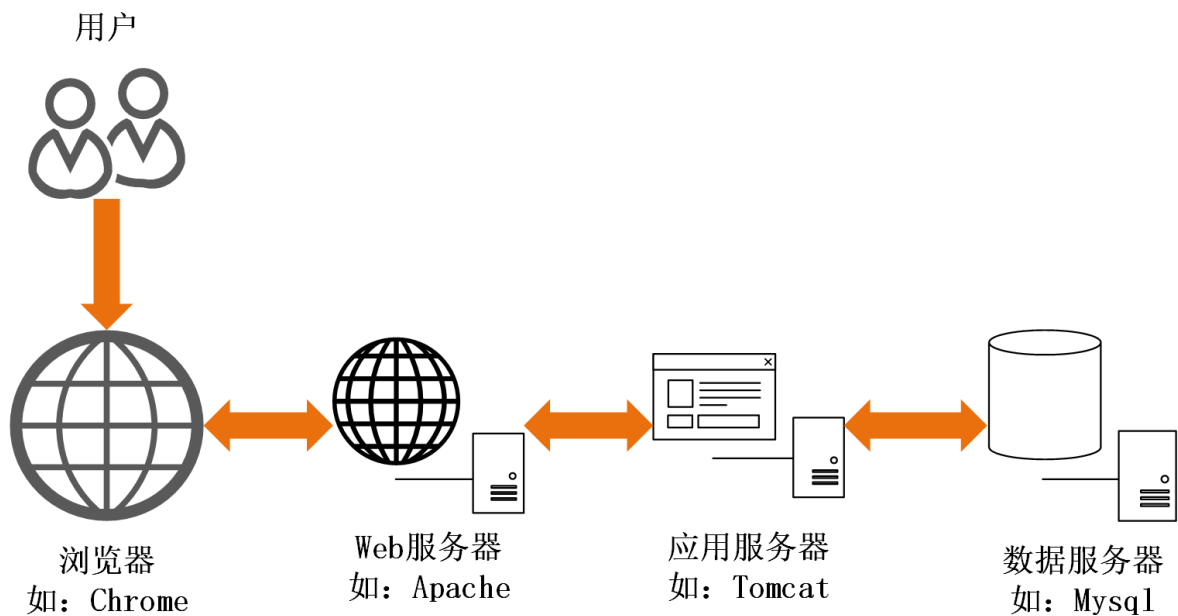


图 1 - 1 单一应用架构

而业务增长到一定程度时，该架构有诸多潜在的问题：

1. 因为代码过于冗杂，系统耦合高，难以进行正常的日常维护，更难以进行后续开发。
2. 在访问量增大时，用户接收了过多不需要的资源，该架构会导致资源开销过大。

如果用单一应用架构搭建电商网站，下单服务、用户注册服务、用户服务等集中于单一的应用服务器中，随着用户的增大访问会导致系统崩溃。

### 1.2.2 垂直应用架构

垂直应用架构是当今最受企业欢迎的架构之一，其中 MVC 体系<sup>[5]</sup>是垂直应用架构中的典范。MVC 体系如图 1-2 所示：

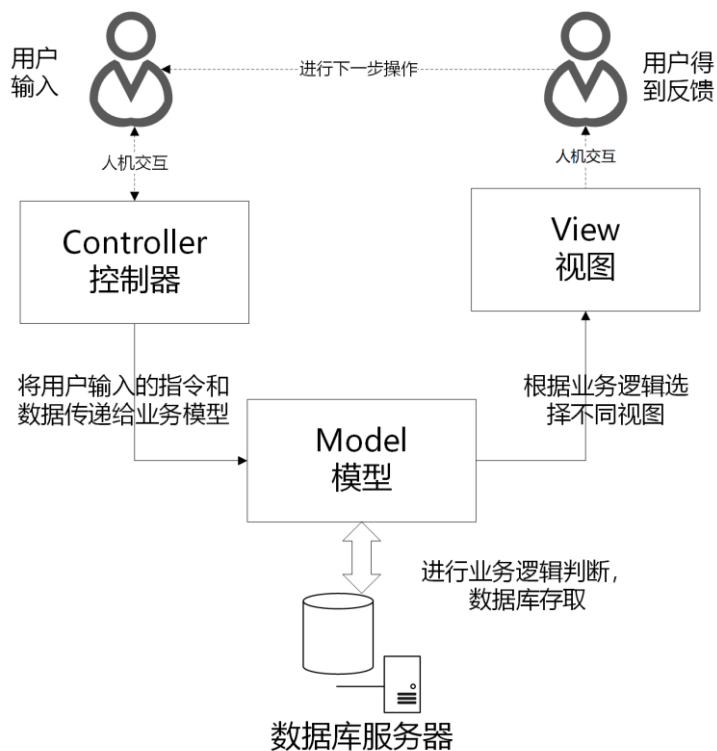


图 1 - 2 MVC 架构

MVC 架构中，M 是 Model，指的是模型，处理各个组件之间的逻辑关系  
V 是 View，指的是视图，负责显示视图。  
C 是 Controller，指的是控制器，负责处理用户请求。

### 1.2.3 分布式应用架构

分布式应用架构<sup>[6]</sup>完成服务的过程不再是以一个整体出现，而是将用于不同服务抽取出，系统运行时需要调用何种服务，就在服务中心调用某种服务。

架构的演进如图 1 - 3 所示：

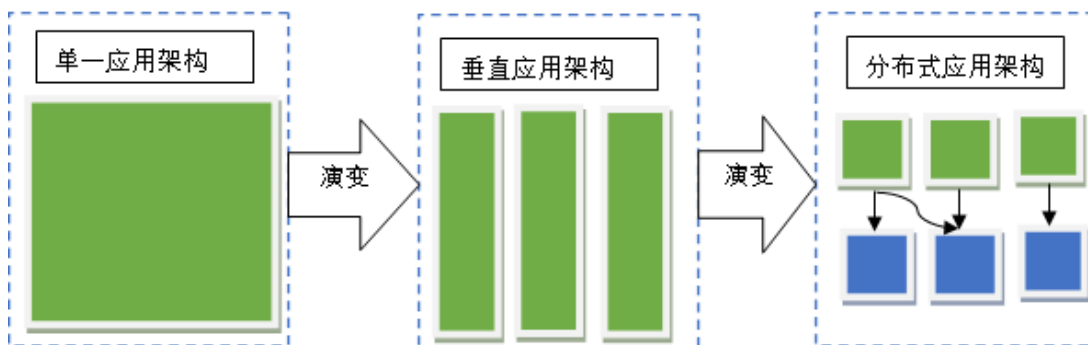


图 1 - 3 大型网站演进

而本文运用分布式应用架构<sup>[7]</sup>搭建易于维护且能经受高并发的网站。

1.2.4 SOA 架构

SOA 为 Service-Oriented Architecture，即面向服务的体系结构，SOA 将大型应用程序分解为不同的服务再将服务和其间的接口联系起来，SOA 的优势在于，通过搭建在系统中不同的服务，通过统一的接口进行交互，实现松耦合的系统，如图 1-4 即为 SOA 架构体系。

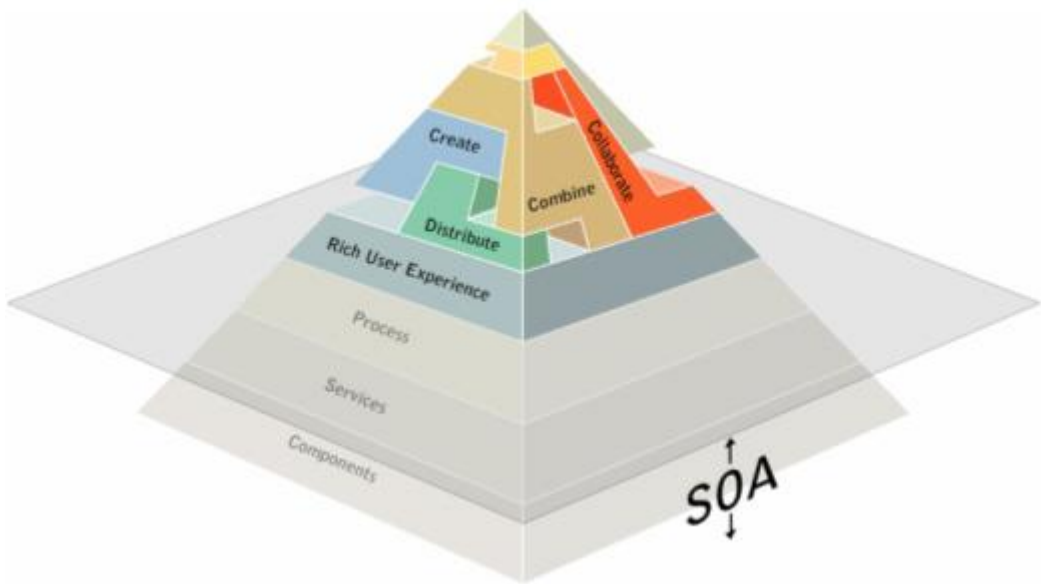


图 1 - 4 SOA 架构体系

1.2.5 远程过程调用

RPC 为 Remote Procedure Call，即远程过程调用，忽略底层网络技术的基础上实现远程计算机程序上的请求服务，远程过程调用的工作流程如图 1-5 所示：

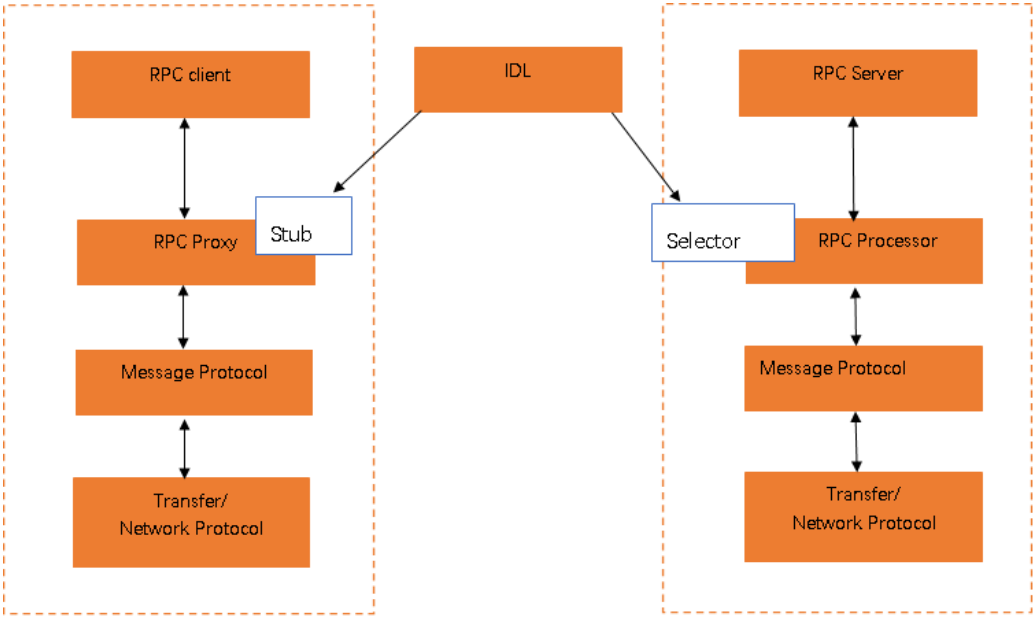


图 1 - 5 远程过程调用工作流程

- RPC Client: 调用方，负责在 RPC 框架忽略 RPC 框架的情况下进行远程过程调

用。

- **RPC Server:** 提供方，提供服务的方法实现。
- **Stub/Proxy:** 代理层，负责处理消息格式、网络传输协议，判断服务的调用过程是否出现异常。
- **Message Protocol:** 消息格式，帮助实现一次完整的 client-server 的交互。
- **Transfer/Network Protocol:** 传输协议层协议、网络层协议。比如，Thrift 传输协议是典型的传输层协议。
- **Selector/Processor:** 选择器和处理器，工作包括管理接口服务的注册、判断 Server 端的权限等工作。
- **IDL:** IDL 即 Interface description language，为接口描述语言。

典型的 RPC 远程调用框架有：

- **Netty 框架:** 封装了多路复用 IO 和阻塞式/非阻塞式 同步 IO 的封装
- **Dubbo 框架:** Dubbo 是国内知名分布式开源框架<sup>[8]</sup>，不但可以为系统提供远程调用服务<sup>[9]</sup>，还为系统提供了访问权限、版本控制、性能优化等功能。
- **Apache Thrift:** Apache Thrift 是 Facebook 所支持的 RPC 框架，为系统提供了远程调用服务。

### 1.3 论文组织架构

本文主要分为六个章节：

第一章为绪论，绪论在宏观上描绘了选题的研究背景和研究内容。

第二章为系统需求分析，本章分析了系统的可行性，从技术方面、用户需求、非功能需求等方面分析了系统的需求。

第三章为系统概要设计，本章分析了后端系统主体设计、Dubbo 框架的工作流程和数据库设计。

第四章为系统实现，本章讲解了系统的实现过程。

第五章为系统测试，本章描述了测试用例的组成以及测试结果，验证了系统地可靠性。

第六章为总结与期望，对项目的工作进行了总结，并对未来进行了展望，总结了网站尚需要完善的功能与模块。

## 第二章 系统需求分析

### 2.1 可行性分析

在设计开发软件之前，为确保顺利完成软件开发，通常会进行可行性分析，本文从以下两个方面进行可行性分析：

#### 2.1.1 技术可行性

本项目的开发重点是网站后端架构，请求的处理采用成熟的 SpringMVC 框架接收请求、响应请求，用户界面则采用 JSP 进行展示，最后利用 Dubbo 对外提供的 API 接口进行远程服务调用，另外，使用 MyBatis 操作数据持久化。

上述技术已经在大型互联网企业中广泛应用，所以此技术架构是可行的。

#### 2.1.2 经济可行性

在系统开发中，所使用的 Dubbo、SpringMVC、MyBatis 均已经开源，开发者只需要调用其 API 接口即可对其进行访问和使用。

在硬件配置方面，只需一台 x64 的个人计算机即可进行开发，无需购买额外的硬件设备。

综上所述，无论是在系统开发层面还是在硬件配置层面，经济层面完全可行。

### 2.2 功能需求分析

#### 2.2.1 并发访问需求

电子商务系统会有大量的用户访问，大量的用户访问必然会有高并发量的需求，进行合理的假设，电子商务系统在全国范围内正常推广开来，预计有 50 万左右的商家入驻本平台，上千万的个人使用该系统，那日均 UV（Unique Visitor，一个用户在一对时间内对网站进行访问的次数）可达上百万次，而 PV（Page View 网页浏览量）可以达到上亿次，为了满足平均上亿次的日均 PV 量，我们将对网站实现分布式架构，将数据洪峰分担到各个子节点上，而基于 Dubbo 的分布式架构为了分担服务器数据洪峰的压力应运而生。

#### 2.2.2 可扩展性需求

为了更好的迎合用户的需求，电商平台必然会有功能的迭代与演进，而在系统开发的过程必须考虑系统的可扩展性来满足不断变化的用户需求，Dubbo 分布式架构作为一套满足 SOA 思想的服务框架，统一管理配置各个系统间服务调用的框架，通过不同服务器间服务调用实现了系统地低耦合性。

2.2.3 角色分析

角色分为两个方面：游客、管理员、普通用户。

- 1. 游客：游客可以浏览商城中商品的信息，包括查看商品的图片、价格、详细信息等，不能进行购买、加入购物车、评论等操作。
- 2. 普通用户：成功注册后即可成为普通用户，用户享有游客所有的功能且可以使用购物车，进行购买操作，对个人信息进行更新等操作。
- 3. 管理员：管理员只能通过后台添加，负责对平台进行运营和维护工作，包括留言回复、订单处理等操作。

如图 2-1 为角色对应的功能模块：

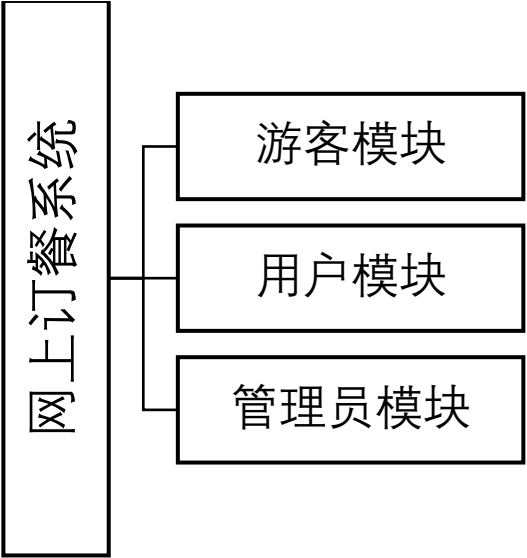


图 2-1 角色模块

2.2.4 功能分析

功能分析是软件工程中里程碑式的一步，其明确了不同角色之间的权限，为后续的软件开发指明了方面。

主要参与者	用例
游客	1. 用户注册 2. 商品信息浏览
普通用户	1. 用户登录/退出 2. 商品信息浏览 3. 购物车使用 4. 个人信息变更 5. 修改密码 6. 商品购买

	7. 评论
系统管理员	1. 管理员登录/退出 2. 商品信息更改 3. 订单管理

表 2-1 系统主要用例

### 2.2.5 用例图

用例图用于系统的功能建模，方便程序员了解系统和用户之间是如何交互的，下面是介绍本平台中各个角色的用例图：

游客用例图如图 2-2 所示：

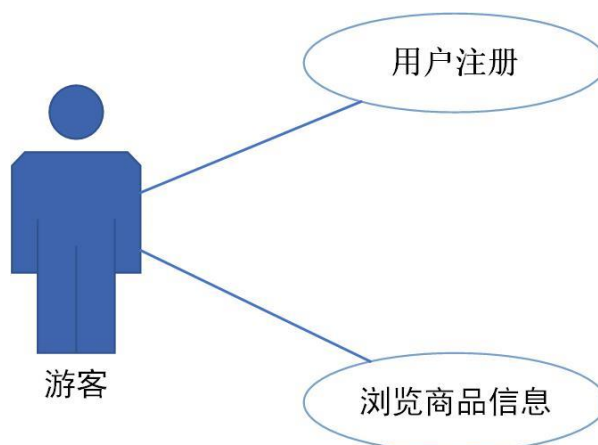


图 2-2 游客用例图

用户用例图如图 2-3 所示：

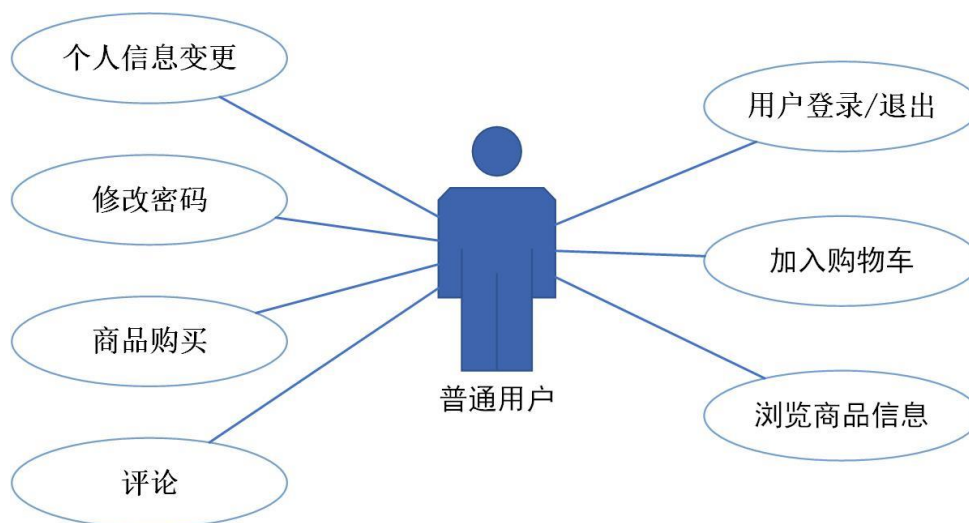


图 2-3 普通用户用例图



管理员用例图如图 2-4 所示：

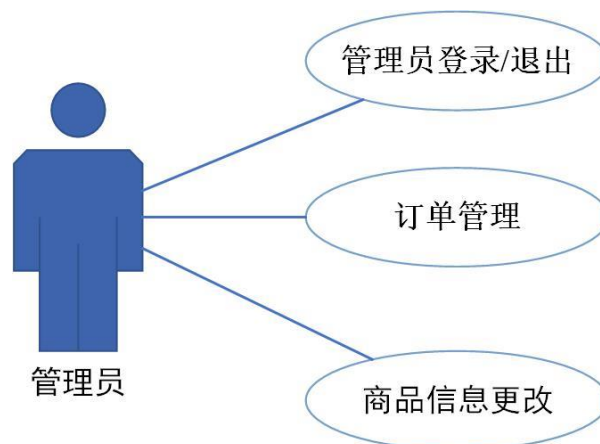


图 2-4 管理员用例图

## 2.3 业务流程

业务流程可以帮助梳理出软件的工作运行的具体流程，直观的观察到目前可能出现的问题。下面依次对各个角色进行分析：

### 1. 游客流程分析

游客只能在网站首页浏览所有的商品，然而没有权限购置商品，也不能将商品放入购物车，拥有的权限受限，如果升级为普通用户，需要进行用户注册操作。

### 2. 用户流程分析

用户首先登陆进入网站，首次登陆需要进行完善收货地址等相关信息，完善个人信息后就可以开始网上购物了，用户可以将喜欢的商品放入购物车中，选择购买就会完成下单流程，购物后可以对已购买的商品进行评价。

### 3. 管理员流程分析

管理员通过后台进行登录，登陆成功后，在订单列表能够检查一切订单信息，管理员可以更新订单的状态。

综上所述，通常情况下，订单的流程主要经过以下四个步骤：一、普通用户选择商品后进行下单。二、管理员对订单进行处理。三、用户确认收货。四、用户评价。

## 第三章 系统概要设计

### 3.1 系统主体设计

系统设计是在需求分析的基础上，由抽象转化为具体的过程，而根据上述需求，最终决定选择了基于 MVC 模式的 SpringMVC 框架，采用分层模式，实现了分布式架构。而通过 Dubbo 框架连接接入层和业务逻辑层，构建分布式框架，在业务逻辑层中加入了如图 3-1 的四个模块：购物车、商品、用户、订单。而基础服务位系统提供了辅佐的服务，如图 3-1 的五个模块：搜索、图片、日志、缓存、消息，而数据存储层为数据的持久性提供必要的服务，利用 Mysql 数据库进行实现。

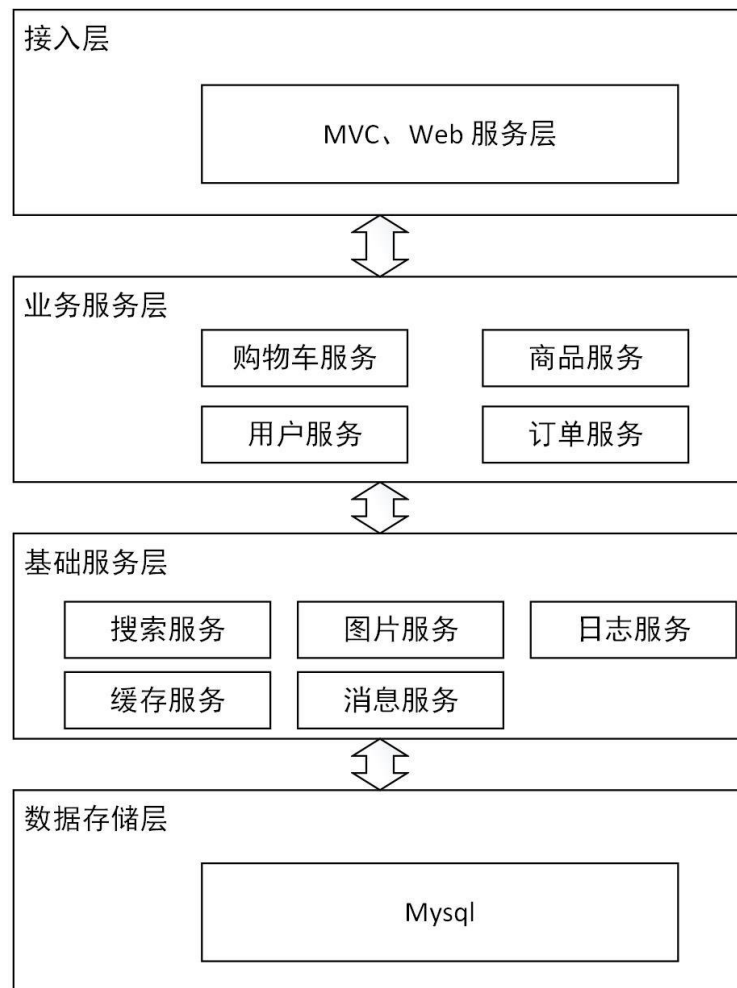


图 3 - 1 系统主体设计

### 3.2 基于 Dubbo + Zookeeper 部署架构

#### 3.2.1 部署架构设计

系统主体设计完成后，为本系统涉及部署架构<sup>[10]</sup>，保证系统能够杂部署架构中稳定且高效的运行。项目的部署架构如图 3 - 2 所示：

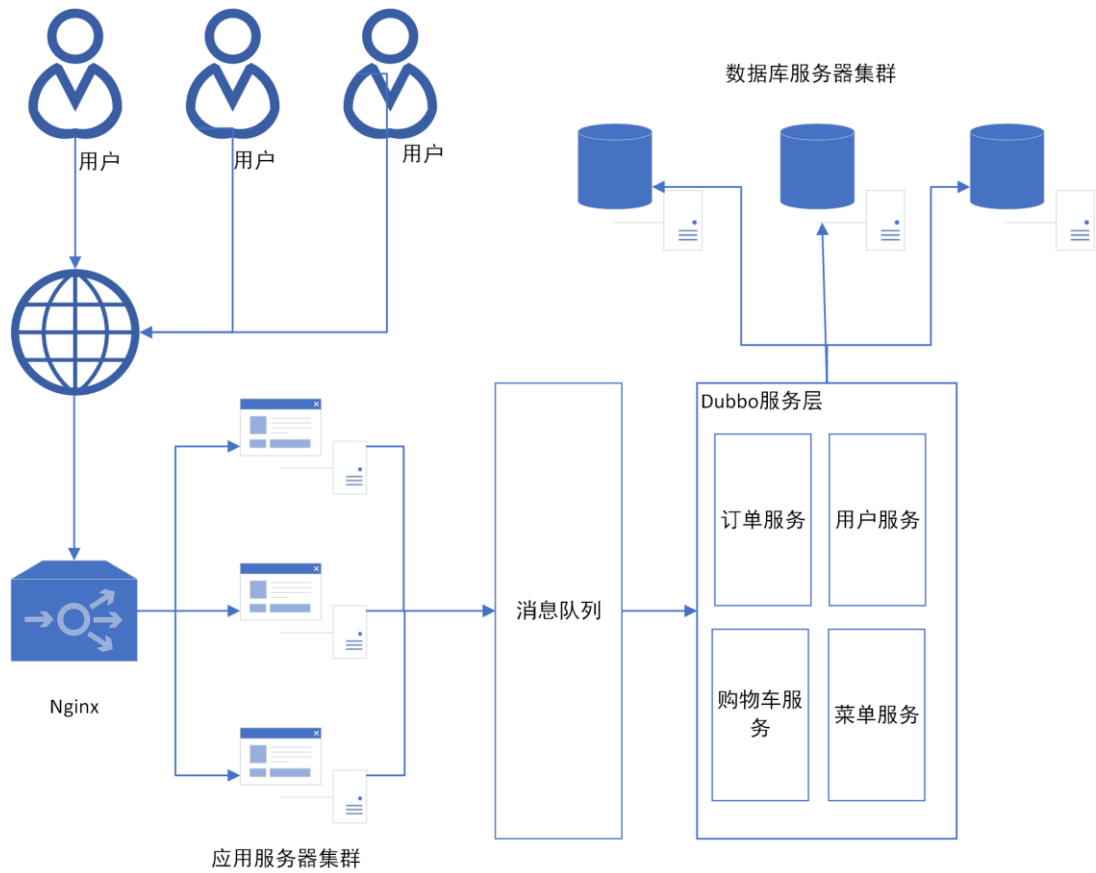


图 3 - 2 项目部署架构

- 1) 不同的用户通过个人计算机访问本系统网页，Nginx<sup>[11]</sup>负责域名解析，将服务分发到合适的应用服务器中，并且 Nginx 还为系统提供资源缓存等服务。
- 2) 应用服务器收到的请求写入消息队列中，而 Dubbo 服务层中的服务则订阅消息队列中的消息，消息队列负责将消息转发到合适的服务模块进行处理。
- 3) Dubbo 服务层对所收到的请求提供低耦合的分布式服务。
- 4) Mysql 关系型数据库，实现读写分离，为数据安全提供保障。

### 3.2.2 Dubbo 工作框架流程介绍

Dubbo 中各节点介绍，如图 3-3 所示：

- Service Provider: 服务的提供者
- Service Consumer: 调用远程服务者
- Container: 运行容器
- Dubbo Monitor: 监控中心
- Dubbo Registry: 发现注册、服务注册中心

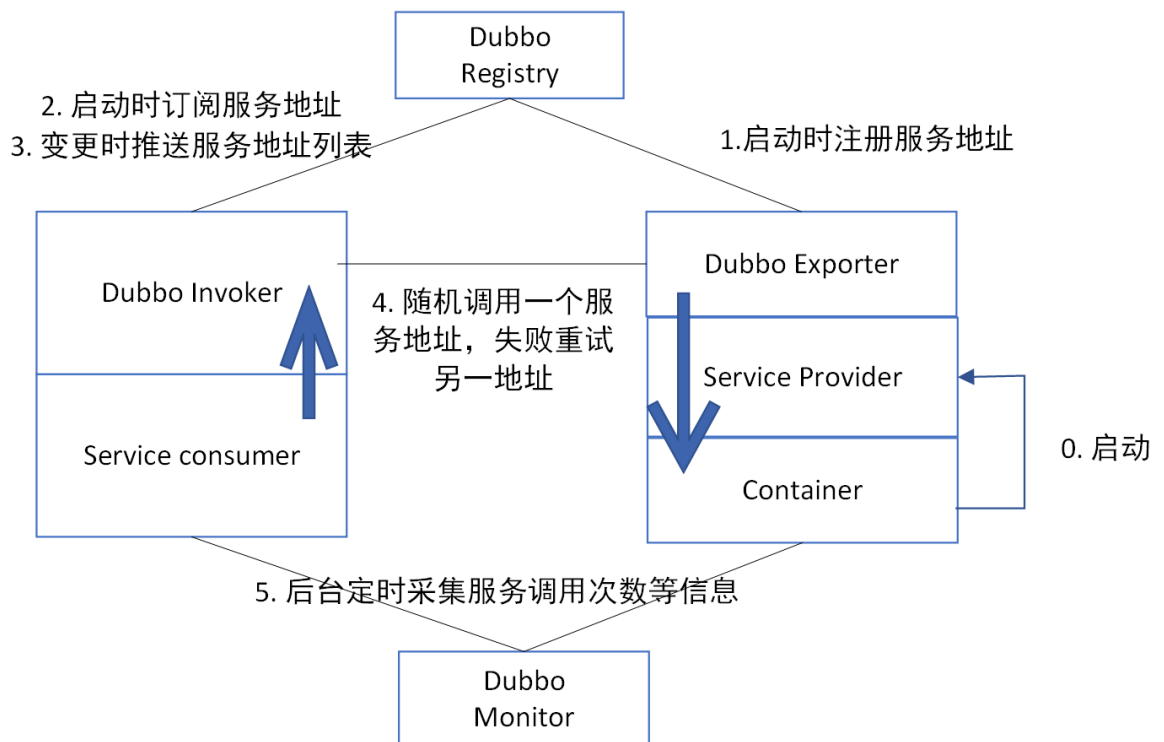


图 3 - 3 Dubbo 工作原理

Dubbo 工作与 Zookeeper<sup>[12]</sup>之间相互调用方式如下：

- （1）首先由 Service Provider 启动和加载 Container，如图 3 - 3 中步骤 0。
- （2）Service Provider 启动运行后，向 Dubbo Monitor 注册自己的服务，如图 3 - 3 步骤 1。
- （3）Service Consumer 在启动后，向 Dubbo Register 订阅自己需要的服务，如图 3 - 3 步骤 2
- （4）Dubbo Register 返回 Service Provider 地址给 Service Consumer，如图 3 - 3 步骤 3
- （5）Service consumer 从 Container 中调用 Service Provider 提供的服务，如图 3 - 3 步骤 4
- （6）Service consumer 和 Provider 的每段时间的调用次数，定时发送给 Service Register 做统计，如图 3 - 3 步骤 5。

### 3.2.3 基于Dubbo + Zookeeper部署的意义

基于 Dubbo + Zookeeper 框架将 Client 和 Server 端分离，做到了业务的解耦，有利于系统功能的扩展和维护，同时也实现了高并发的访问，有利于帮助电商网站应对数据洪峰。

### 3.3 系统整体流程

用户根据网址进入网站后首先打开的是网站的首页，游客有权限浏览商品信息，如果需要更进一步的操作，需要进行注册操作，游客注册成功，成为平台的普通用户，普通用户可以进行个人信息的修改，查看商品的详情，将商品加入购物车中，并可以选择继续购物，如果选择结束购物，跳转到生成订单界面，支付成功后订单随即更新；而成为管理员后，可以登录管理后台，可以进行商品管理，对订单的增删改查操作。系统整体流程设计如图 3-3 所示：

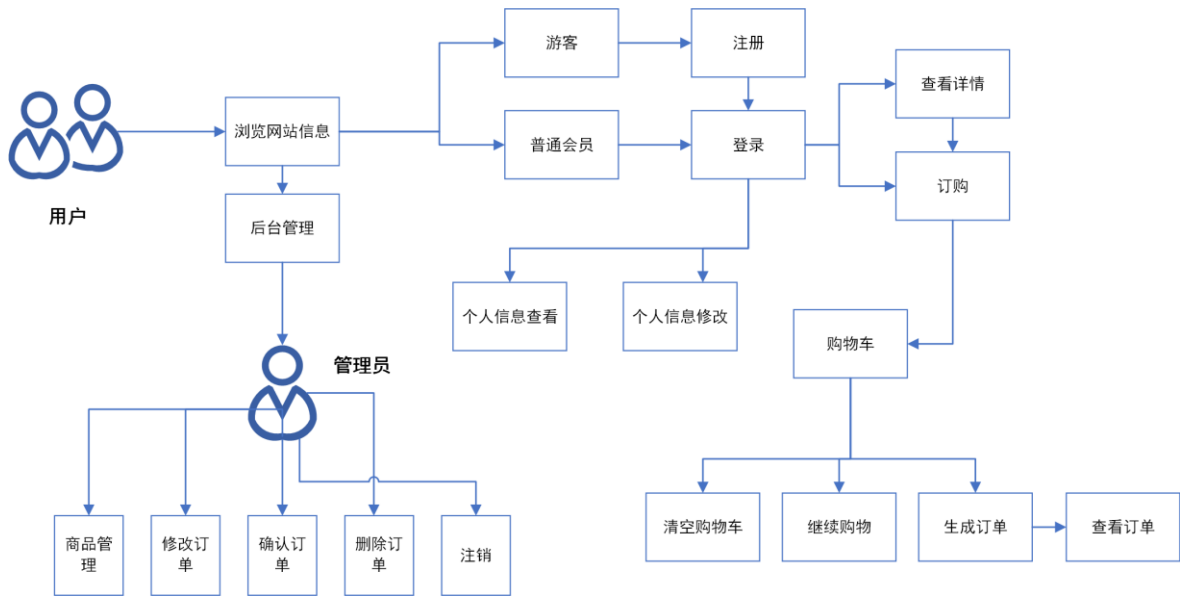


图 3 - 4 系统整体流程

### 3.4 数据库设计

建立结构合理的数据库相当必要，根据需求分析，设计了以下几张数据库表：用户表（mmall\_user）、分类表（mmall\_category）、产品表（mmall\_product）、购物车表（mmall\_cart）、支付信息表（mmall\_pay\_info）、订单表（mmall\_order）、订单明细表（mmall\_order\_item）、收货地址表（mmall\_shipping）如下：

#### 3.4.1 用户表（mmall\_user）

名称	字段名称	数据类型	主键	非空	备注
编号	id	INT（11）	是	是	自增
用户名	username	VARCHAR（50）	否	是	UNIQUE_KEY
密码	password	VARCHAR（50）	否	是	

电子邮件	email	VARCHAR（50）	否	否	
电话号码	phone	VARCHAR（20）	否	否	
问题	question	VARCHAR（100）	否	否	
答案	answer	VARCHAR（100）	否	否	
角色判定	role	INT（4）	否	是	角色 0-管理员， 1-普通用户
创建时间	create_time	DATETIME	否	是	
更新时间	update_time	DATETIME	否	是	

表 3 - 1 用户表

#### 3.4.2 分类表（mmall\_category）

名称	字段名称	数据类型	主键	非空	备注
编号	id	INT（11）	是	是	自增
父类别编号	parent_id	INT(11)	否	否	
类别名称	name	VARCHAR(50)	否	否	
类别状态	status	TINYINT(1)	否	否	类别状态 1-正 常，2-废弃
排序编号	sort_order	INT(4)	否	否	
创建时间	create_name	DATETIME	否	否	
更新时间	update_time	DATETIME	否	否	

表 3 - 2 分类表

#### 3.4.3 产品表（mmall\_product）

名称	字段名称	数据类型	主键	非空	备注
商品编号	id	INT（11）	是	是	自增
分别编号	category_id	VARCHAR（50）	否	是	
商品名称	name	VARCHAR（100）	否	是	

商品副标题	subtitle	VARCHAR (50)	否	否	
产品主图	main_image	VARCHAR (50)	否	否	
图片地址	sub_images	VARCHAR (20)	否	否	
商品详情	detail	VARCHAR (100)	否	否	
价格	price	VARCHAR (100)	否	否	
库存数量	stock	INT (4)	否	否	
商品状态	status	INT (6)	否	否	1-在售, 2-下架, 3-删除
创建时间	create_time	DATETIME	否	否	
更新时间	update_time	DATETIME	否	否	

表 3 - 3 产品表

#### 3.4.4 购物车表 (mmall\_cart)

名称	字段名称	数据类型	主键	非空	备注
编号	id	INT (11)	是	是	自增
用户编号	user_id	INT (11)	否	是	
产品编号	product_id	INT (11)	否	否	
数量	quantity	INT (11)	否	否	
是否选择	Checked	INT (11)	否	否	1=已勾选,0=未勾选
创建时间	create_time	DATETIME	否	是	
更新时间	update_time	DATETIME	否	是	

表 3 - 4 购物车表

#### 3.4.5 支付信息表 (mmall\_pay\_info)

名称	字段名称	数据类型	主键	非空	备注
编号	id	INT (11)	是	是	自增

用户编号	user_id	INT (11)	否	否	
订单号	order_no	BIGINT (20)	否	否	
支付平台	pay_platform	INT (10)	否	否	1- 支付宝,2- 微信
支付宝支付流水号	platform_number	VARCHAR (200)	否	否	
支付宝支付状态	platform_status	VARCHAR (20)	否	否	
创建时间	create_time	DATETIME	否	否	
更新时间	update_time	DATETIME	否	否	

表 3 - 5 支付信息表

#### 3.4.6 订单表 (mmall\_order)

名称	字段名称	数据类型	主键	非空	备注
编号	id	INT (11)	是	是	自增
订单号	order_no	BIGINT (20)	否	否	UNIQUE KEY
用户编号	user_id	INT	否	否	
地址号	shipping_id	INT	否	否	
实际付款金额	payment	DECIMAL	否	否	
支付类型	payment_type	INT	否	否	1-在线支付
运费	postage	INT	否	否	
订单状态	status	INT	否	否	
支付时间	payment_time	DATETIME	否	否	
发货时间	send_time	DATETIME	否	否	
交易完成时间	end_time	DATETIME	否	否	
交易关闭时间	close_time	DATETIME	否	否	



创建时间	create_time	DATETIME	否	否	
更新时间	update_time	DATETIME	否	否	

表 3 - 6 订单表

#### 3.4.7 订单明细表（mmall\_order\_item）

名称	字段名称	数据类型	主键	非空	备注
编号	id	INT（11）	是	是	自增
用户编号	user_id	INT(11)	否	否	
订单号	order_no	BIGINT(20)	否	否	
商品编号	product_id	INT（11）	否	否	
商品名称	product_name	VARCHAR（100）	否	否	
商品图片地址	product_image	VARCHAR（500）	否	否	
商品单价	current_unit_price	DECIMAL（20，2）	否	否	
商品数量	quantity	INT（10）	否	否	
商品总价	total_price	INT（20，2）	否	否	
创建时间	create_time	DATETIME	否	否	
更新时间	update_time	DATETIME	否	否	

表 3 - 7 订单明细表

#### 3.4.8 收货地址表（mmall\_shipping）

名称	字段名称	数据类型	主键	非空	备注
编号	id	INT（11）	是	是	自增
用户编号	user_id	INT（11）	否	否	
收货姓名	receiver_name	VARCHAR（20）	否	否	
收货固定电话	receiver_phone	VARCHAR（20）	否	否	
收货移动电话	receiver_mobile	VARCHAR（20）	否	否	

省份	receiver_province	VARCHAR (20)	否	否	
城市	receiver_city	VARCHAR (20)	否	否	
区/县	receiver_distinct	VARCHAR (20)	否	否	
详细地址	receiver_zip	VARCHAR (200)	否	否	
邮编	receiver_zip	VARCHAR (6)	否	否	
创建时间	create_time	DATETIME	否	否	
更新时间	update_time	DATETIME	否	否	

表 3 - 8 收货地址表

### 3.5 系统软件列表

系统开发中，采用的产品如图 3 - 5 所示：

类别	软件产品	版本
操作系统	Windows	10
开发语言	Java	
开发架构	SpringMVC + MyBatis	Spring-webmvc 3.2.0
Java 应用服务器	Tomcat	7.0
数据库	Mysql	5.1.73
开发工具及版本	IDEA	15.0.6

图 3 - 5 系统清单表

## 第四章 系统实现

### 4.1 开发环境

- 系统开发工具：IDEA 15.06， JDK 1.8, Tomcat 7.0
- 相关技术：Maven, Spring MVC, JSP, Dubbo
- 数据库：MySQL
- 操作系统：Win 10
- 浏览器：Chrome

### 4.2 系统的实现

根据上文中框架选择、需求分析以及系统概要设计，确定了系统的功能模块，也确定了开发本系统所使用的框架，而系统的实现是实现所做的所有需求与计划，搭建出电子商务系统所具备的基本功能。

#### 4.2.1 分布式框架的搭建

本系统采用 Dubbo + Zookeeper 的分布式架构进行部署，将业务层和服务层解耦，降低系统的耦合性，分布式框架的实现如图 4-1 所示：

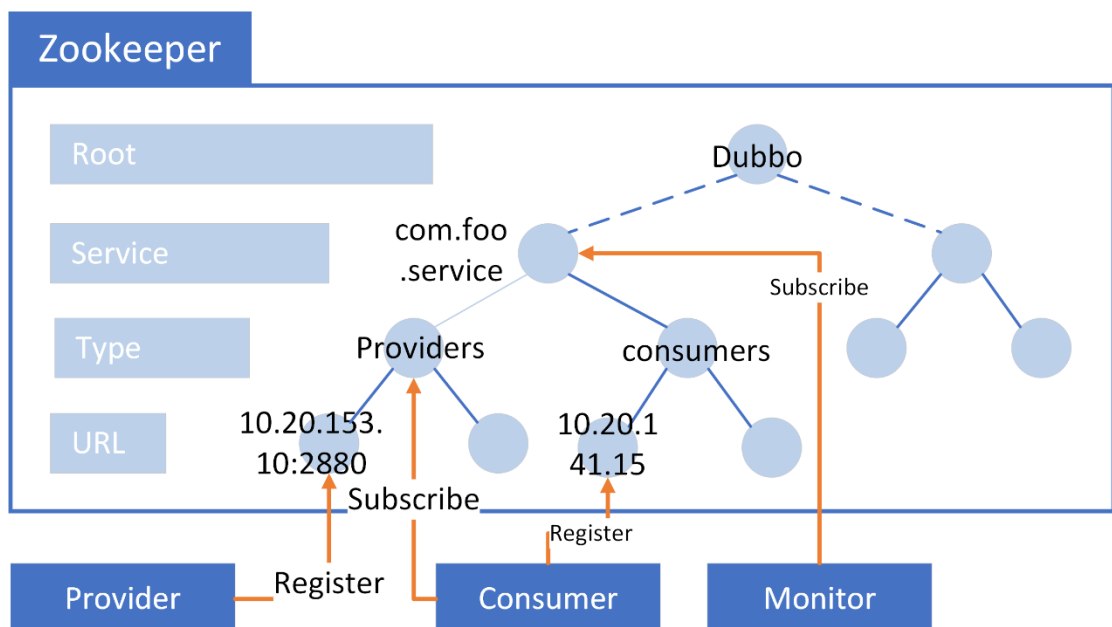


图 4 - 1 分布式框架

1. 系统启动时，服务提供者（Provider）做出响应，Zookeeper 进行服务注册，向 dubbo/com.foo.service/providers 写入 URL 地址。
2. 接着，服务消费者（Consumer）启动后做出响应，触发两个动作：

订阅 dubbo/com.foo.service/providers 服务提供者（Provider）的地址，且向 dubbo/com.foo.service/consumers 写入自身的地址。

3. 监控中心（Monitor）启动后，订阅 dubbo/com.foo.service/consumers 提供者和消费者的地址。

### 4.2.2 用户注册

用户注册成为平台的会员后才可以进行下单操作，用户注册需要用户名（username）、密码（password）、电子邮件(email)、电话号码(phone)、密码找回问题(question)和其答案(answer)。

账号（username）：必填项，长度保证在 50 个字符以内。

密码（password）：必填项，长度保证在 50 个字符以内。

电子邮件（email）：必填项，长度在 50 个字符以内

电话号码（phone）：必填项，长度在 20 个字符以内

密码找回问题（question）：必填项，长度在 100 字符以内

答案（answer）：必填项，长度在 100 字符以内

用户注册流程图如下图 4 -2 所示：

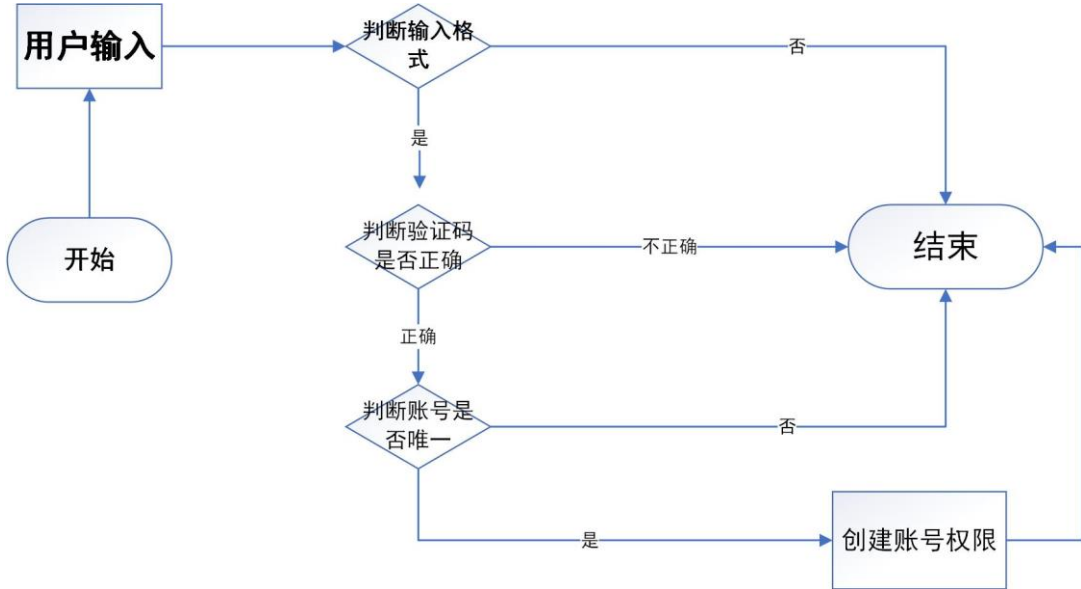


图 4 - 2 用户注册

➤ 后端用户登录接口如表 4 - 1 所示：

Http 通信	请求内容 (http body)	类型	备注
Post 请求 (Request)	username	varchar	Admin
	password	Varchar	admin
	Email	Varchar	<a href="mailto:Sdada@163.com">Sdada@163.com</a>
	Phone	Varchar	Daa
	Question	varchar	Dadzv

		Answer	Varchar	dsadad
响应 (Response)	响应成功	Status	Int	0
		Msg	Varchar	校验成功
	响应失败	Status	Int	1
		Msg	Varchar	用户已存在

表 4 - 1 用户登录表

➤ 检查用户名是否有效接口如表 4-2 所示：

Http 通信		请求内容 (http body)	类型	备注
Post 请求 (Request)		Str	varchar	admin
		Type	Varchar	Username
响应 (Response)	响应成功	Status	Int	0
		Msg	Varchar	校验成功
	响应失败	Status	Int	1
		Msg	Varchar	用户已存在

表 4 - 2 有效性检查

#### 4.2.3 用户登录

注册成功后，普通用户通过用户名和密码进行登录。

账号 (username)：必填项，长度保证在 50 个字符以内。

密码 (password)：必填项，长度保证在 50 个字符以内。

用户注册流程图如下图 4 -3 所示：

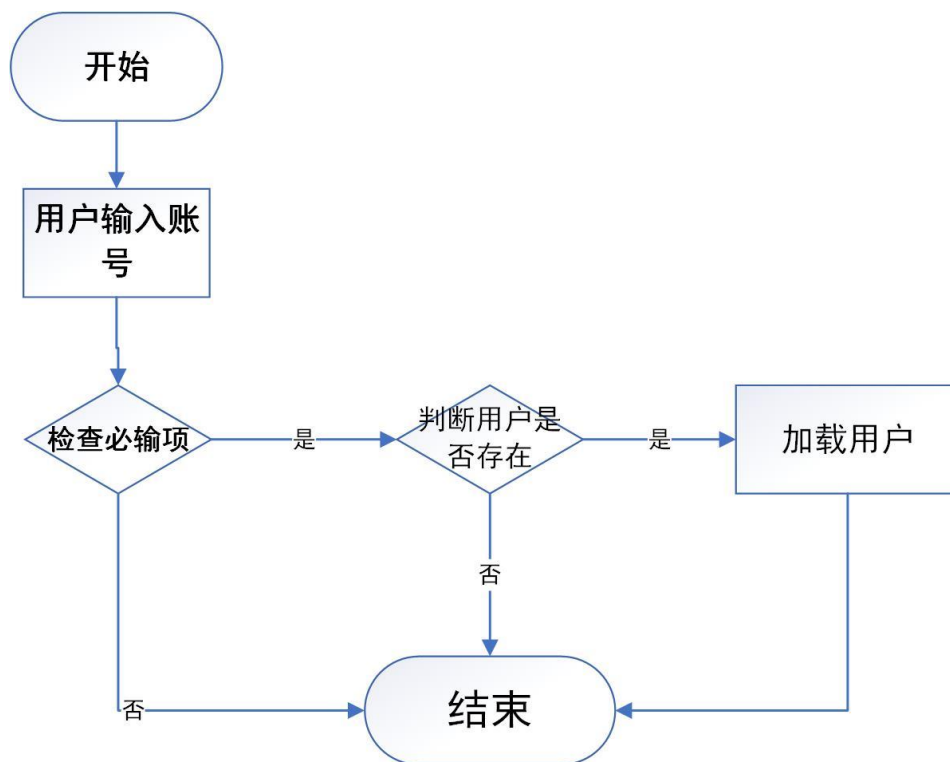


图 4 - 3 用户登录

➤ 用户登录接口：

Http 通信		请求内容 (http body)	类型	备注
Post 请求 (Request)		Username	varchar	admin
		Password	Varchar	Username
响应 (Response)	响应成功	Status	Int	0
		Id	Int	12
		Username	Varchar	Admin
		Email	Varchar	<a href="mailto:asd@163.com">asd@163.com</a>
		Phone	Varchar	Null
		role	Varchar	0
		createTime	Varchar	1479045343500
		updateTime	Datetime	1479045343500
		Msg	Varchar	校验成功
	响应失败	Status	Int	1
		Msg	Varchar	密码错误

表 4 - 3 用户登陆表

#### 4.2.4 下订单

成功登录后，普通用户通过电商平台上的众多商品展示图，挑选喜欢的商品，并将其加入购物车，接着重复此过程直到购买好所需的所有物品，然后核对购物车中所选商品，填写收货地址，选择快递信息等方式进行下单。

用户注册流程图如下图 4 -4 所示：

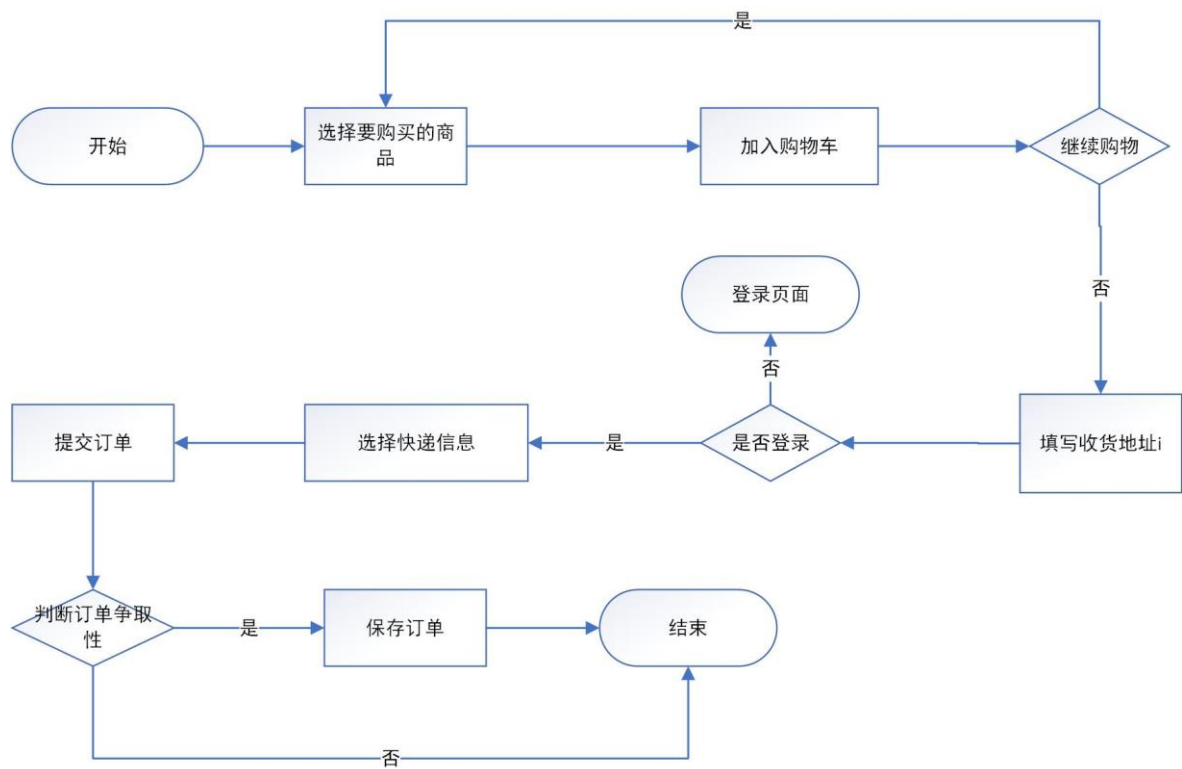


图 4 - 4 下订单

➤ 下订单接口如表 4 -4 所示：

Http 通信		请求内容 (http body)	类型	备注
Post 请求 (Request)		ShippingId	int	12
响应 (Response)	响应成功	OrderNo	Int	1351234214095
		Payment	Double	30021.11
		postage	Int	1
		Status	Int	0
		paymentTime	Datetime	Null
		sendTime	Datetime	Null
		endTime	Datetime	Null
		CloseTime	Datetime	Null
		createTime	Datetime	1351234214095
		ShippingId	Int	5
		shippingNo	int	Null
	响应失败	Status	Int	1
		Msg	Varchar	创建订单失败

表 4 - 4 创建订单

➤ 购物车接口如表 4 -5 所示：

Http 通信	请求内容 (http body)	类型	备注
	productId	int	12

Post 请求 (Request)		count	Int	10
响应 (Response)	响应成功	status	Int	0
		Id	Int	1
		userId	Int	13
		Quantity	Int	1
		productName	Int	12
		productSubtitle	VARCHAR	Iphone7
		productMainImage	VARCHAR	双十一促销
		productPrice	VARCHAR	Main.jpg
		productStatus	Double	7192.33
		ProductStock	Int	1
		productTotalPrice	Double	895454.90
	响应失败	Status	Int	1
		Msg	Varchar	用户未登录, 请登录

表 4 - 5 购物车

#### 4.2.5 网站展示图

下图为网站首页实例图 4-6 所示：

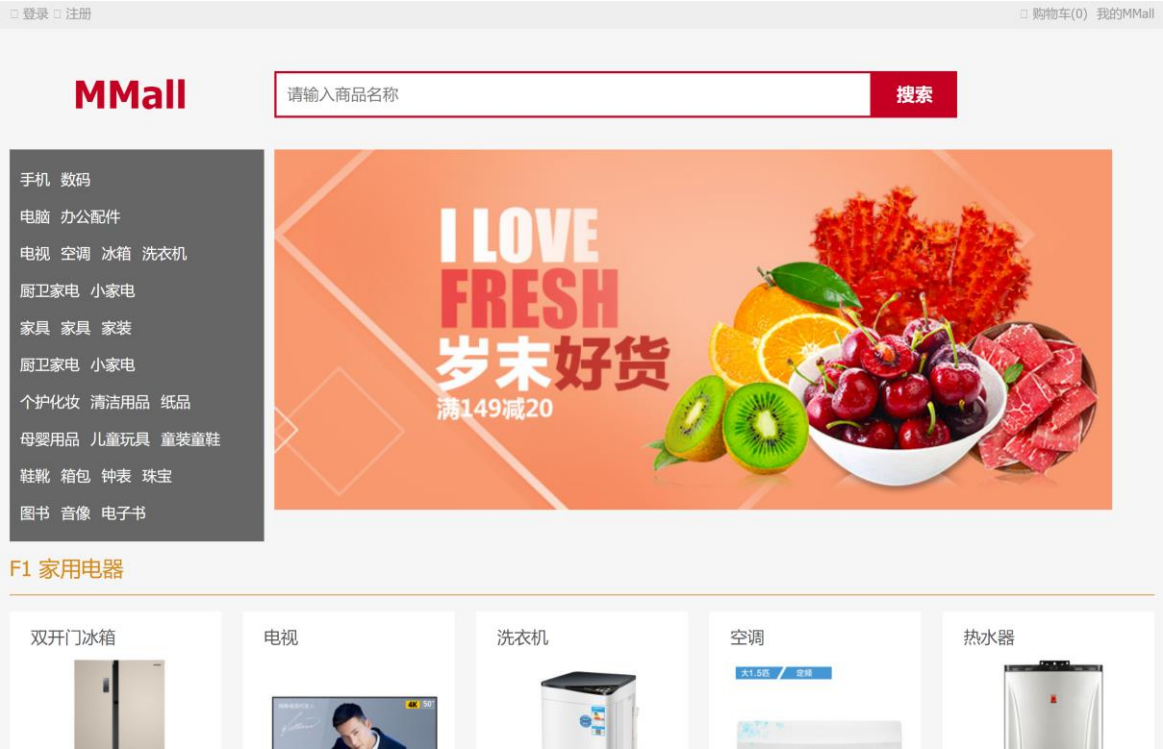


图 4 - 6 网站首页图



## 第五章 系统测试

在软件开发中，随着功能的迭代，项目变得越来越复杂，难免会使庞大的项目陷入泥淖中，本章中软件测试对各个模块进行测试，以确保整个系统正常运行。

### 5.1 系统功能测试

本节对客户端各个功能模块进行测试，保证每个模块的可靠性，才能保证整个软件的顺利运行，测试的模块包括：用户模块、分类管理模块、商品管理模块、购物车模块、收货地址模块等。

#### 5.1.1 用户注册功能测试

用户注册模块如表 5-1 所示：

前提条件	测试步骤	预期测试结果	实际测试结果
用户未登录	点击注册	跳转至注册界面	正常
用户成功登录	用户注销	注销成功	正常
	点击用户个人信息	显示用户个人信息	正常
	更改个人信息数据	成功更改且保存	正常

表 5 - 1 用户注册模块

Register.do 接口测试如图 5 - 1 所示：

The screenshot displays a REST client interface for testing the Register.do endpoint. The request is a POST method to the URL http://localhost:8088/user/register.do. The request body is a form with the following parameters: username (mrz), password (mr), email (adsda), phone (2313), question (dad), and answer (dz). The Content-Type is set to application/x-www-form-urlencoded. The response is a 200 OK status with a JSON body: {"status": 0, "msg": "注册成功", "data": null, "success": true}. The response headers show Server: Apache-Coyote/1.1, Content-Type: application/json; charset=UTF-8, Transfer-Encoding: chunked, and Date: Tue, 15 May 2018 05:55:42 GMT.

图 5 -1 注册接口测试

5.1.2 购物功能测试

购物功能测试如表 5-2 所示：

前提条件	测试步骤	预期测试结果	实际测试结果
用户未注册	购买商品	购买失败	正常
用户已注册	购买商品	购买成功且订单模块更新	正常
	加入购物车	加入购物车成功	正常
	个人信息设置	个人信息设置成功	正常
	查看商品详情	进入商品查看详情界面	正常
	支付订单	弹出支付页面	正常

表 5 - 2 购物功能模块

购物车接口测试如图 5 - 2 所示：

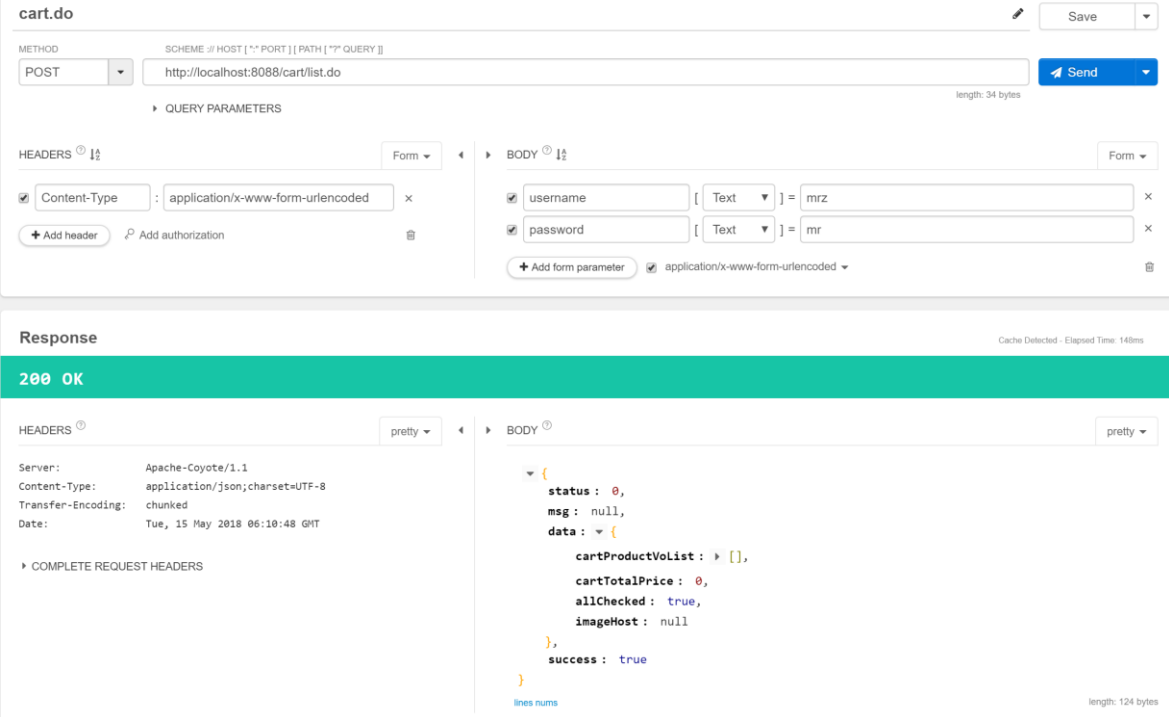


图 5 - 2 购物车接口测试

5.1.3 后台管理功能测试

后台管理功能如表 5-3 所示：

前提条件	测试步骤	预期测试结果	实际测试结果
管理员成功登录	管理员注销	返回登录页面	正常
	删除订单	订单已删除	正常
	留言回复	回复成功	正常
	商品信息更新	更新成功	正常
	添加商品	商品添加成功	正常

	修改订单状态	修改成功	正常
--	--------	------	----

表 5 - 3 后台管理模块

添加品类接口测试如图 5 - 3 所示

add\_category.do

METHOD

POST

SCHEME // HOST [ "\*" PORT ] [ PATH [ "?" QUERY ] ]

http://localhost:8088/manage/category/add\_category.do

length: 63 bytes

Send

QUERY PARAMETERS

HEADERS

Content-Type

:

application/x-www-form-urlencoded

×

+

Add header

?

Add authorization

BODY

categoryName

[

Text

]

=

冰淇淋

×

+

Add form parameter

?

application/x-www-form-urlencoded

Response

Cache Detected - Elapsed Time: 23ms

200 OK

HEADERS

pretty

Server: Apache-Coyote/1.1

Content-Type: application/json;charset=UTF-8

Transfer-Encoding: chunked

Date: Tue, 15 May 2018 06:32:30 GMT

COMPLETE REQUEST HEADERS

BODY

pretty

{

status: 0,

msg: null,

data: "添加品类成功",

success: true

}

length: 66 bytes

图 5 - 3 商品添加接口测试

5.2 总结

本章针对本系统的功能性进行了测试，测试结果反映出该系统各个模块正常运转。

## 第六章 总结与展望

### 6.1 总结

随着互联网的快速发展,电子商务已经成为了大众消费中必不可少的一环,本系统基于 MVC 模式的 SpringMVC 框架,采用分层模式,实现了分布式架构,通过 Dubbo 框架连接接入层和业务逻辑层,构建分布式框架,利用 Mysql 数据库进行实现了数据的持久性。

另外就我个人而言,在完成本次毕业设计中,我将我在专业中学到的知识,转化为实际的开发能力,在此过程中,我认识到自身实践能力不足,比如我在 Java 课程中可以取得不错的成绩,但是在写 Java 工程的时候,却总是 bug 不断,这次毕业设计是我大学所学知识的一次大练兵吧。

### 6.2 展望

相对于市面上的电商系统来讲,本系统有着很多待完善的内容,主要分为以下几点:

#### (1) UI 界面过于简陋

本系统的界面较为单一,UI 采用 JSP 中原生图标,界面的美观性过于简陋,如果考虑到软件的用户体验,应该请专业的美术人员对 UI 进行重新设计。

#### (2) 功能模块单一

相较于淘宝等电子商务平台,本平台缺少精品商品讯息的推送,优化搜索算法,关注的店铺等功能。在日后的开发中,可以借鉴淘宝等产品,适当增加可以为产品带来流量的功能。

## 第七章 参考文献

- [1] Fabian Aulkemeier, Milan Schramm, Maria-Eugenia Iacob, Jos van Hillegersberg. A Service-Oriented E-Commerce Reference Architecture [J] ProQuest. 2016. 26-45
- [2] 郭正敏. 基于 SOA 架构的分布式服务化治理方案的研究[D].南京邮电大学,2016
- [3] 任女尔,张庆余,林盛海.基于 Dubbo+ZooKeeper 的 CAMDS 协同业务改造[J].电脑知识与技术,2016,12(29):103-105
- [4] 李智慧. 大型网站技术架构: 核心原理与案例分析[M]. 第 1 版. 北京: 电子工业出版社. 2003: 25-146
- [5] 刘意凡. 基于 Spring MVC 的亚马逊物流信息平台的设计与实现[D].北京交通大学,2017.
- [6] 武装,李洪奇. 一种分布式 SOA 体系结构的设计与实现 [J].微计算机信息,2008,24(36):253-255.
- [7] Delin Hou. Design of Distributed Architecture based on Java Remote Method Invocation Technology[A]. Intelligent Information Technology Application Research Association, Hong Kong. Environmental Science and Information Application Technology Volume 2[C].Intelligent Information Technology Application Research Association, Hong Kong:,2009:4.
- [8] Alibaba Tech. Dubbo Developer Guide[EB/OL]. 2018-03-19. <http://dubbo.io/Developer+Guide.htm>
- [9] 严静. 基于数据分发服务的远程过程调用机制的研究与实现[D].东南大学,2017.
- [10] 张炜森,陈涛,李康. Nginx 高并发负载均衡原理与策略比较研究[J].工业控制计算机,2018,31(01):85-86+89.
- [11] 叶婷. 基于 Java 的农村电子商务系统的设计与实现 [硕士学位论文] [D] 太原: 中北大学, 2017
- [12] 陈天伟,彭凌西.基于 ZooKeeper 的一种分布式系统架构设计与实现[J].通信技术,2018,51(01):87-91.

关键词*		密级*	中图分类号*	UDC
Dubbo、分布式、电子商务		公开	G633.67	62
论文赞助				
学位授予单位*		学位授予单位代码*	学位类别*	学位级别*
中国计量大学		10356	工学	学士
论文题名*	基于 Dubbo 的分布式电子商务平台应用			论文语种*
并列题名*	无			简体中文
作者姓名*	孟仁梓	学号*	1401400118	
培养单位名称*	培养单位代码*	培养单位地址		邮编
中国计量大学	10356	浙江省杭州下沙高教园区学源街		310018
学科专业*	研究方向*		学制*	学位授予年*
计算机科学与技术	计算机科学与技术		4	2018 年
论文提交日期*				
导师姓名*	陶海军	职称*	副教授	
评阅人		答辩委员会主席*		
答辩委员会成员				
电子版论文提交格式 文本 (√) 图像 ( ) 视频 ( ) 音频 ( ) 多媒体 ( ) 其他 ( ) 推荐格式: application/msword; application/pdf				
电子版论文出版 (发布者)	电子版论文出版 (发布) 地		权限声明	
论文总页数*				
注: 共 33 项, 其中带“*”为必填数据。				