

毕业设计说明书



|  |  |  |
| --- | --- | --- |
| 题目 | : | 购物导向性服装店网站设计 |
| 学院 | : | 计算机与软件工程学院 |
| 年级专业 | : | 2015软件工程 |
| 姓名 | : | 周鹏 |
| 学号 | : | 3120150407230 |
| 指导教师 | : | 李月卉 |
| 完成时间 | : | 年 月 日 |

**摘要**

随着计算机技术与电子商务的快速发展，带有推荐功能的导向性的购物网站已成为时代主流，一方面可以根据地理位置、浏览行为、购买行为等多个维度进行商品推荐，以更加个性化、差异化的服务用户。另一方面，随着个人用户量的增多，如何调整网站的技术架构、提升高并发处理能力也成为技术难点。而二者正是困扰许多中小型电商企业的问题。

本设计针对电商服装购物网站，通过采用Spring Cloud微服务框架整合Spring Boot后台开发框架、Mybatis数据库访问组件、Redis缓存、Maven依赖管理、前后端分离等技术手段来实现一个分布式、微服务的技术架构，从而提升网站的高并发处理能力。并且通过记录用户购物行为、IP定位等手段来实现多维度的推荐系统，达到一定的购物导向性。其主要功能包括用户管理、服装商品与店铺管理、服装推荐等、短信验证、服装店定位、公交路线规划等。一方面，达到了购物导向的效果，另一方面，实现了高并发微服务架构。

本论文详细阐述了本网站开发的分析、设计、实现等各阶段工作过程，可为同类网站的设计开发提供一定的说明，对于中小型电商企业实现导向性推荐系统、高并发技术架构两方面都具有较好的参考价值。

**【关键词】**导向性 微服务 服装网站

# 

Abstract

With the rapid development of computer technology and e-commerce, shopping websites with recommendation function have become the mainstream of the times. On the one hand, they can recommend goods according to geographical location, browsing behavior, purchase behavior and other dimensions, so as to provide more personalized and differentiated service users. On the other hand, with the increase of the number of individual users, how to adjust the technical framework of the website and improve the high concurrent processing capacity has become a technical difficulty. And they are the problems that perplex many small and medium-sized e-commerce enterprises.

This design aims at e-commerce clothing shopping website. It integrates Spring Boot backstage development framework, Mybatis database access component, Redis cache, Maven dependency management, front-end and back-end separation and other technical means to achieve a distributed and micro-service technology architecture, so as to enhance the high concurrent processing capacity of the website. And through recording users'shopping behavior, IP positioning and other means to achieve multi-dimensional recommendation system, to achieve a certain shopping orientation. Its main functions include user management, clothing merchandise and shop management, clothing recommendation, short message verification, clothing store positioning, bus route planning, etc. On the one hand, it achieves the effect of shopping orientation, on the other hand, it achieves a high concurrent micro-service architecture.

This paper elaborates the analysis, design, implementation and other stages of the website development process, which can provide a certain explanation for the design and development of similar websites, and has a good reference value for small and medium-sized e-commerce enterprises to achieve oriented recommendation system and high concurrent technology architecture.

**【Keywords】：**Orientation Micro Service Clothing Website

**目 录**

目录

[**摘要** 1](#_Toc5971972)

[Abstract 2](#_Toc5971973)

[1 绪论 7](#_Toc5971974)

[1.1 论文研究背景介绍 7](#_Toc5971975)

[1.2 电商服装购物网站现状与发展趋势 7](#_Toc5971976)

[1.3 电商服装购物网站开发过程中的特点分析 8](#_Toc5971977)

[1.3.1 缓存组件的使用 8](#_Toc5971978)

[1.3.2 前后端分离 8](#_Toc5971979)

[1.3.3 分布式部署架构 8](#_Toc5971980)

[1.3.4 推荐导向系统提升用户体验 9](#_Toc5971981)

[1.4 论文主要工作 9](#_Toc5971982)

[1.5 开发环境说明 10](#_Toc5971983)

[1.6 开发的目的和意义 11](#_Toc5971984)

[2 技术分析 11](#_Toc5971985)

[2.1 技术栈总览 11](#_Toc5971986)

[2.2 微服务架构分析 11](#_Toc5971987)

[2.2.1 原始单一架构分析 11](#_Toc5971988)

[2.2.2 传统分布式架构分析 12](#_Toc5971989)

[2.2.3 微服务架构 13](#_Toc5971990)

[2.2.4 微服务实现之Spring Cloud框架 14](#_Toc5971991)

[2.3 技术栈介绍 14](#_Toc5971992)

[2.3.1 开发框架Spring Boot分析 14](#_Toc5971993)

[2.3.2 开发框架Spring Boot分析 15](#_Toc5971994)

[3 软件可行性研究 15](#_Toc5971995)

[需求分析 16](#_Toc5971996)

[需求规范 16](#_Toc5971997)

[产品背景 16](#_Toc5971998)

[产品概述 16](#_Toc5971999)

[功能需求 16](#_Toc5972000)

[性能需求 17](#_Toc5972001)

[难点以及关键点分析 17](#_Toc5972002)

[设计思路 18](#_Toc5972003)

[初步方案 18](#_Toc5972004)

[概要设计 19](#_Toc5972005)

[设计目的 19](#_Toc5972006)

[架构设计 20](#_Toc5972007)

[游戏功能模块设计 20](#_Toc5972008)

[游戏功能模块设计 20](#_Toc5972009)

[游戏内容设计 21](#_Toc5972010)

[程序流程图 23](#_Toc5972011)

[数据库设计 24](#_Toc5972012)

[数据表 24](#_Toc5972013)

[命名规范 25](#_Toc5972014)

[数据库命名规范 25](#_Toc5972015)

[项目文件命名规范 25](#_Toc5972016)

[详细设计 25](#_Toc5972017)

[注册模块 25](#_Toc5972018)

[界面设计 26](#_Toc5972019)

[设计思想 26](#_Toc5972020)

[关键代码 26](#_Toc5972021)

[登录模块 29](#_Toc5972022)

[界面设计 29](#_Toc5972023)

[设计思想 29](#_Toc5972024)

[关键代码 29](#_Toc5972025)

[游戏初始化模块 31](#_Toc5972026)

[界面设计 31](#_Toc5972027)

[设计的思想 31](#_Toc5972028)

[关键代码 31](#_Toc5972029)

[地图初始化模块 32](#_Toc5972030)

[界面设计 32](#_Toc5972031)

[设计的思想 32](#_Toc5972032)

[关键代码 33](#_Toc5972033)

[游戏开始模块 34](#_Toc5972034)

[界面设计 34](#_Toc5972035)

[设计思想 34](#_Toc5972036)

[关键代码 34](#_Toc5972037)

[游戏结束模块 35](#_Toc5972038)

[界面设计 35](#_Toc5972039)

[设计思想 35](#_Toc5972040)

[关键代码 36](#_Toc5972041)

[其他关键技术思想 37](#_Toc5972042)

[软件测试 38](#_Toc5972043)

[游戏功能测试 38](#_Toc5972044)

[游戏初始化界面测试 38](#_Toc5972045)

[地图初始化界面测试 39](#_Toc5972046)

[重置游戏地图测试 40](#_Toc5972047)

[碰撞过程测试 41](#_Toc5972048)

[游戏结束界面测试 42](#_Toc5972049)

[游戏白盒测试 43](#_Toc5972050)

[碰撞过程白盒测试 43](#_Toc5972051)

[测试结论 45](#_Toc5972052)

[总结 45](#_Toc5972053)

[致谢 46](#_Toc5972054)

[参考文献 47](#_Toc5972055)

1 绪论

1.1 论文研究背景介绍

根据国内电商年度报告显示，以阿里巴巴、京东等为代表的电子商务企业，旗下都拥有自己的B2C、C2C个人端购物网站，如阿里巴巴的淘宝网、天猫商城以及京东的京东商城等。这些大型电商企业凭借其先进的互联网管理理念与运作方式，引进了高精尖的互联网技术人才，使其产品经得住时代的考研，能承受大量的并发访问，逐渐改变了人们的购物方式与生活方式。

而对于中小型企业来说，目前已经开通了网上商店的企业已经尝到了电子商务带来的甜头,国际与国内许多知名品牌大多已经在网络上有了自己的品牌专营店。对于一些知名度稍弱的品牌,也在酝酿为自己的产品在网络上打开另一条销售渠道。这样,就可以面向来自世界各地的消费群体,不仅是企业,越来越多的个人也将加入到网络销售中,变成电子商务大军中的一员。

对于中小型企业来说，因曾经开发OA系统、管理系统的经验，更偏向于使用传统技术与架构，如传统后端技术框架SSH、同程序多服务器部署等，这并不利于其开展B2C、B2C等针对个人用户电商业务，由于个人用户的用户量大，并发访问高，对于服务器性能、数据库性能都是一个考验，传统的技术架构不适用于电商购物网站的业务场景。

1.2 电商服装购物网站现状与发展趋势

根据《[2018-2022年中国服装零售市场深度评估及未来发展趋势报告](https://www.chyxx.com/research/201612/477178.html)》2018年，我国服装网购市场交易规模达到7232亿元，年复合增长率将达到14%左右。2015年以来，服装行业移动端销售额迅猛增长，移动端成交额已占到总体行业销售额一半以上，移动端渗透率在2017年以来左右突破50%，时间点晚于食品、家电数码及美妆行业，但早于鞋包行业。服装行业目前的移动渗透率在所有行业里处领先地位。

为了更好的体现其产品特色，满足用户差异化，定制化，个人化购物的需求，中小型电子商务企业、知名服装店在依附大型电商开始网站的同时，也在开发自己的专属购物网站。

1.3 电商服装购物网站开发过程中的特点分析

由于存在大量的个人用户注册、同时刻并发访问等情况，电商服装购物网站的开发都围绕着如何更好的处理并发请求、更快的响应速度来进行，往往存在以下特点，其目的是为了改善用户体验，来提高用户满意度，从而减少用户流失。

# 1.3.1 缓存组件的使用

由于并发量大，而关系型数据库实际存放位置都为硬盘，如果请求数据都从数据库去查，则会带来性能瓶颈。对于某些特定数据、更新频率低、查找频率高，往往使用内存数据库NO-SQL来进行缓存，如Redis、MongDB，数据直接从内存中获取，从而提高后端接口口响应速度。

# 1.3.2 前后端分离

传统的网站前端，如PHP、JSP等，采用脚本式的将PHP、Java语言嵌入前端网页当中，部署时也需要专门的服务器来运行。优点是开发方便，简化前端人员的开发工作，缺点是需要依赖特定的服务器、部署不够灵活，同样也存在性能瓶颈。而使用前后端分离的架构方式，前端由纯HTML、CSS、JS构成，则数据解析处理只需要在客户端浏览器进行，省去了在服务器的数据解析过程，从而数据解析压力由服务器转到客户端浏览器，则可以很大程度减轻服务器的压力，从而提高并发访问的性能瓶颈。前后端的数据交互一般使用XML、Json格式数据，现目前大部分企业都使用JSON数据。

# 1.3.3 分布式部署架构

传统的应用程序部署一台服务器，当并发访问量较高时，解决方式是通过加大带宽与服务器配置。这种方法会带来资源过剩与程序崩溃两个极端问题，一方面，加大服务器内存、CPU配置在一定程度的确可以提高响应速度，但是还受应用程序本身，如Tomcat线程限制，数据库连接数量限制等问题；另一方面，这种部署方式，一旦程序崩溃，则软件直接无使用。目前BS架构中、往往会将应用程序部署多个，分布式部署多台服务器，再由Nginx来实现轮询调度，负载均衡，共同承担用户的并发请求，从而提高并发量，不仅解决了单台服务器资源过剩、而且当一个程序崩溃时，其他程序依旧可以提供服务，软件依然可以使用。

微服务架构则在普通分布式架构上再次优化，将大型应用程序拆分成若干个服务，更细微的区分了存在性能瓶颈的点，为程序服务器资源分配提供了有效依据。

# 1.3.4 推荐导向系统提升用户体验

由市场上的淘宝网、天猫商城、京东购物等大型电商购物网站的引领作用，中小型企业的购物网站，也需要推荐系统来更加个性化的发现用户喜好，进行商品推荐、店铺推荐，带来更加人性化的购物体验。

1.4 论文主要工作

本论文，以服装购物网站为例，对电商购物网站中的推荐系统作了探索与开发、对微服务分布式架构的开发技术进行了研究与运用。从需求分析开始，到测试完成，详细的记录与解释了导向性购物网站的开发过程、生命周期、技术选型与分析等。主要分为以下几个方面：

（1）研究Redis缓存、微服务分布式架构、IP定位、推荐导向及开发框架、Git版本控制等相应技术点与实现方式。

（2）确立需求分析和设计方案，利用主流技术框Spring Boot、Mybatis数据库访问组件与C3P0连接池进行各个功能模块进行后台开发。

（3）前端开发，使用HTML5、CSS3、LayerUI、JavaScript进行开发，前后端分离部署。

（4）单元测试、集成测试、功能测试。对实现的功能和网页展现形式进行测试，并使用Spring Boot结合Junit进行单元测试，使用SwaggerUI自动化测试框架进行接口集成测试，保留测试代码与测试数据。

（5）使用Spring Cloud实现微服务架构与部署，使用Eureka注册中心实现服务集中管理。

（6）记录开发过程中遇到的困难以及难点，分析出网站的难点以及重点、实用价值及参考价值。

本文的研究内容包括7章：

第1章为绪论，主要对论文研究的背景进行了简单的介绍，对电商服装购物网站的发展现状以及发展趋势进行了简单的分析，对购物网站开发过程中的特点进行了简单的描述，以及对论文主要的工作进行了罗列，对软件开发环境进行了简单的介绍，以及介绍了本次购物服装网站的目的和意义。

第2章对该网站的开发技术栈作了较为详细的分析，包括技术本身的特征以及与传统技术的对比说明。

第3章为可行性研究，主要从技术可行性、经济可行性、法律可行性等方面对网站进行了可行性的研究。

第4章为需求分析，主要进行了需求分析，其中包括网站实现的功能与目标，功能的需求，性能的需求，以及设计思路和初步方案的选定。

第5章为网站的概要设计，主要阐述了设计的目的，架构设计，以及对功能的设计（网站功能设计，网站内容设计），程序流程图以及数据库的设计。

第6章为软件详细设计，对网站的每个模块进行了详细的设计，设计的思想以及关键的代码进行了罗列。

第7章为软件的测试，从单元测试、接口集成测试、到功能测试进行了测试，以保证游戏功能的完整性以及系统的稳定性。

1.5 开发环境说明

（1）操作系统：个人电脑Windows 10企业版、阿里云Ubuntu 16.04.4

（2）程序语言：Java、Html5、JavaScript

（3）后端开发工具：IntelliJ IDEA，目前业界最流行的后端开发工具，相对于Eclipse作了不少提升，其开发与调试都更加方便。

（4）前端开发工具：Visual Studio Code，一款界面美观，插件丰富的前端开发IDE。

（5）版本控制：Git，代码仓库使用Github托管

（6）其他辅助工具：Xshell远程连接工具、Navicat数据库视图工具、Photoshop图片处理、Chrome浏览器调试等。

1.6 开发的目的和意义

本设计针对电商服装购物网站，进行推荐导向系统、微服务实现分布式架构两个点关键点进行探索与实现。

在该开发过程中，对技术进行研究与学习，对代码风格进行规范，锻炼自己的学习能力，同时也为自己技术栈新增元素，提升了自己技术实力。另外，该开发可为同类网站的设计开发提供一定的说明，对于中小型电商企业实现导向性推荐系统、高并发技术架构两方面都具有较好的参考价值。

2 使用技术栈分析

* 1. 技术栈概览

（1）后端：Spring Cloud微服务框架、Spring Boot开发框架、Mybatis数据库访问组件、C3P0数据库连接池、Redis缓存、Maven依赖包管理、Gson格式化组件、秒嘀科技短信验证平台、IP定位高德地图API、Zxing二维码生成、HttpURLConnection网络请求与爬虫

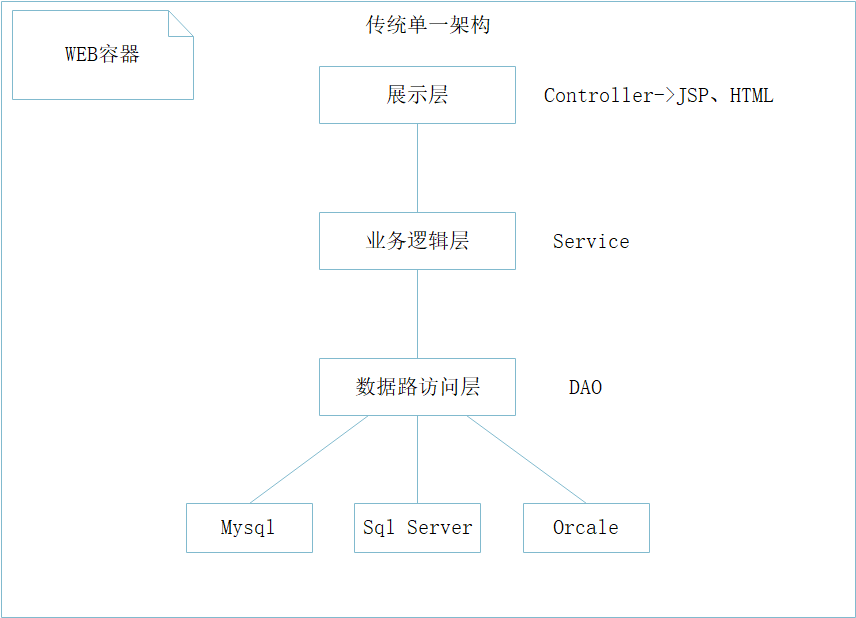
（2）前端：LayerUI前端框架、Html5、Css3、JavaScript

2.2 微服务架构分析

对于企业来说，随着公司业务量的飞速发展，平台面临的挑战已经远远大于业务，需求量不断增加，技术人员数量增加，面临的复杂度也大大增加。在这个背景下，平台的技术架构也完成了从传统的单体应用到微服务化的演进。下面将对传统原始单一架构，到传统分布式，到微服分布式，分别进行分析与说明。

# 2.2.1 原始单一架构分析

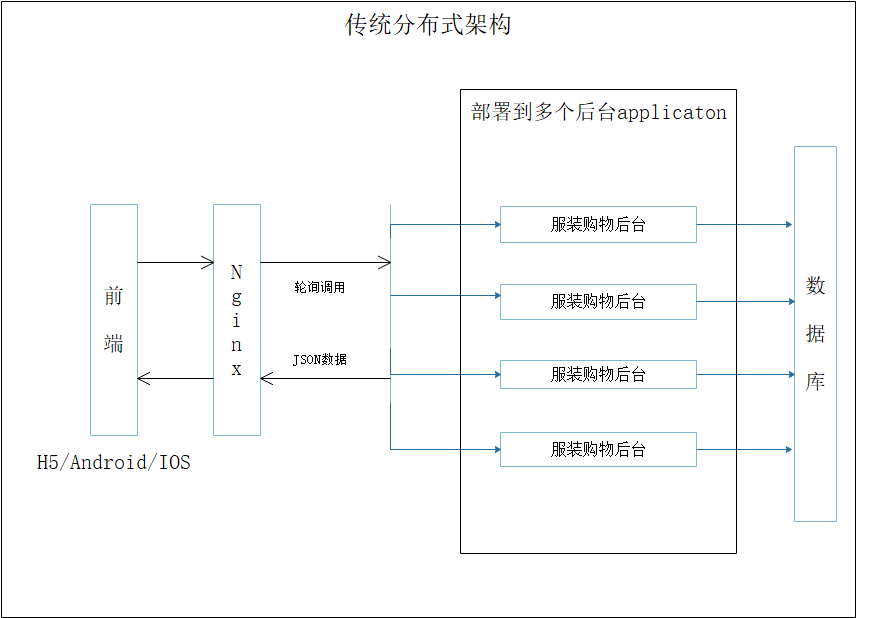
原始的单一的架构，将前端页面与后端代码放在同一个项目中，然后将其部署在一个WEB容器（如tomcat）中。如图2.1所示：

图2.1 传统单一应用程序架构图

通过上图可以看出，展示层使用JSP（PHP）脚本页面、HTML，使用MVC控制层控制器视图跳转。这样的架构是用户流量较小时，为了节约成本,并将所有应用都打包放到一个应用里面。缺点是并发承受能力较低，一旦服务器宕机，整个程序将无法运行。

# 2.2.2 传统分布式架构分析

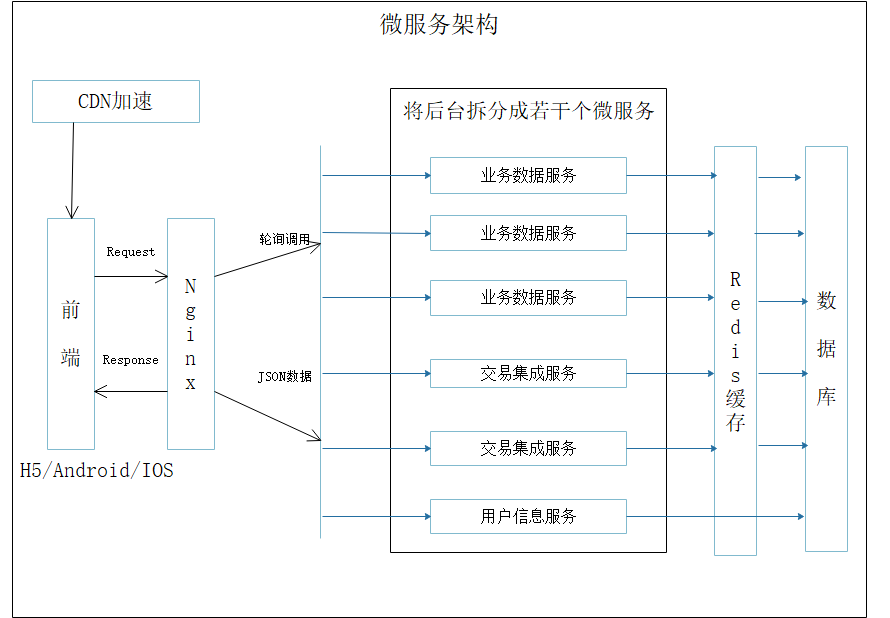
传统的分布式架构，为前后端分离，使用Json或XML格式数据进行交互。同时将后端程序部署多个，如图2.2所示：

图2.2 传统分布式架构

根据上图可以看出，传统分布式架构将后台程序部署了多个分别在不同的服务器上，而且前后端分离，如果其中一台服务器宕机，其后台程序仍然可以正常使用。相对于原始的单一应用程序架构，该方式可以提高并发处理能力，并且软件不会因为某一台服务器问题，而导致整个软件无法使用。

# 2.2.3 微服务架构

相对传统的分布式架构，微服务架构在应用程序级别将各个后台拆分为了若干个微服务，一个简单的微服务架构示例，如图2.3所示：

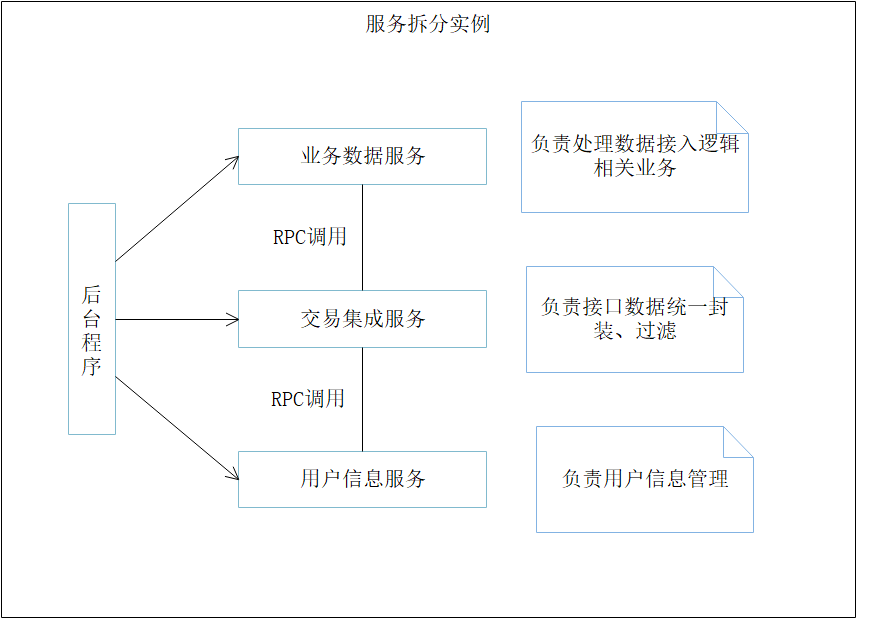
图2.3 微服务架构示例

由上图可见，后台程序拆分成了业务数据服务、交易集成服务、用户信息服务三种不同的微服务，并且业务数据服务部署的最多，因为其与数据库交互最为频繁，其次是交易集成服务，用来统一处理前端请求与响应。用户信息服务则只针对用户登录、注册等进行处理，相对功能单一，部署数量则较少，从而有针对性的进行部署，增加接口耗时长的部署服务数量，提高服务器资源利用率，更好的服务高并发业务场景。服务负责处理的业务示例如表2.1：

表2.1 服务对应业务场景表

|  |  |  |  |
| --- | --- | --- | --- |
| 服务名称 | 业务数据服务 | 交易集成服务 | 用户信息服务 |
| 性能瓶颈 | 容易产生性能瓶颈 | 易产生 | 不易产生 |
| 业务类型 | 数据库查询、更新 | 数据封装、接口处理 | 用户信息管理 |
| 业务示例 | 商品搜索 | 商品搜索结果过滤 | 用户登录 |
| 服务部署数量 | 多 | 较多 | 较少 |

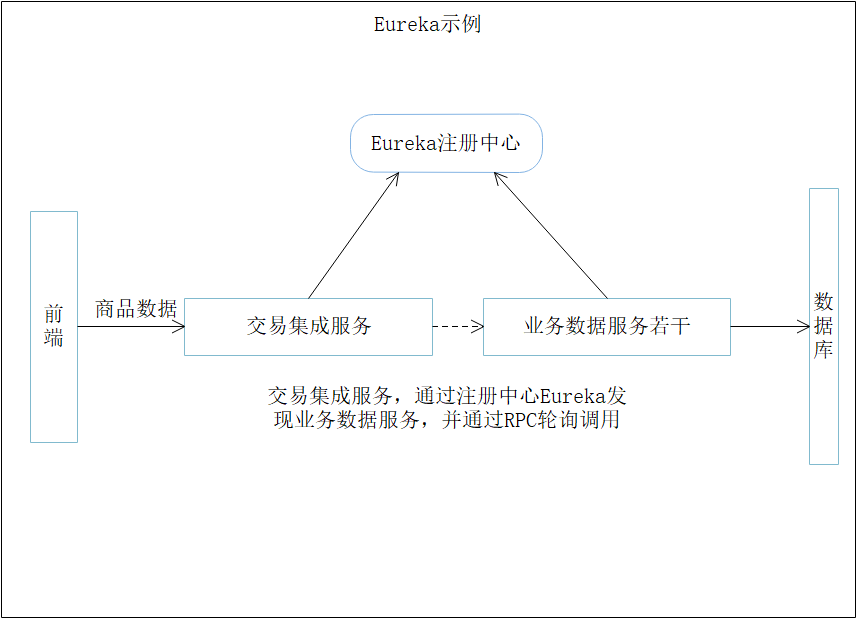
通过这种方式，则有多个微服务共同来构成整个后台程序，其服务之间通过RPC调用，来实现交互，一个后台程序的服务拆分实例如图2.4所示：

图2.4 服务拆分实例图

# 2.2.4 微服务实现之Spring Cloud框架

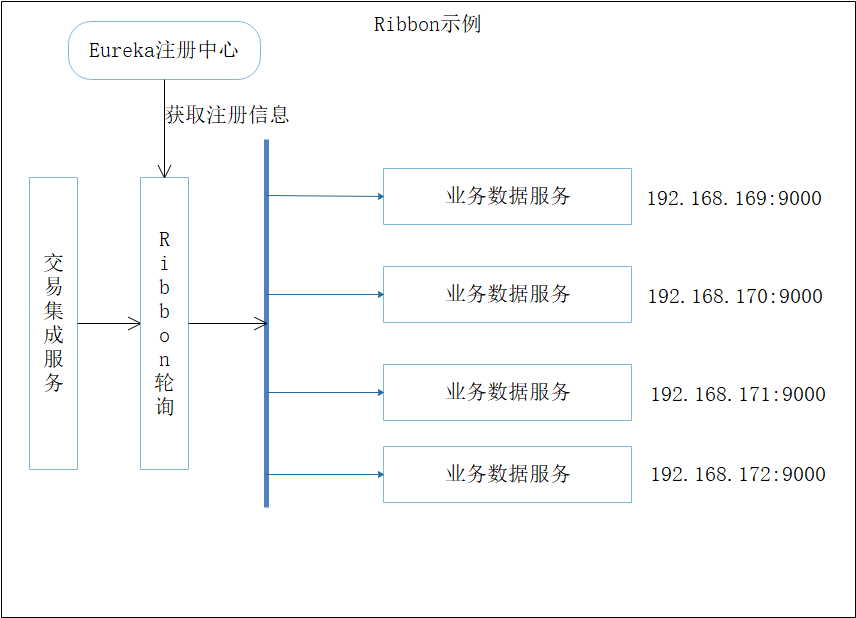
Spring Cloud是目前市场上主流的微服务实现技术栈，它是一个全家桶式的技术栈，包含了很多组件：Eureka、Ribbon、Feign、Hystrix、Zuul。

（1）Eureka是Spring注册中心，它本身也是一个服务，所有的服务运行以后，都需要在注册中心上去注册，Eureka将维护一个注册表，包含注册服务的IP地址与端口欣喜，并持续发送心跳包保持其活跃，一旦发现心跳检测不通，则将该服务从注册表中去除，当服务之间调用之前，会通过Eureka的注册表，去查找目标服务的IP地址与端口。Eureka注册中心示例如图2.5所示：

图2.5 Eureka注册中心示例

如上图所示，交易集成服务和业务数据服务，都在Eureka上面注册，Eureka保存了各服务所在的机器和端口号，并且会定时通过发送心跳检测服务是否正常运行。当服务需要互相调用时，比如交易集成服务，在接收到前端POST的商品发布数据，需要调用业务数据服务将商品数据存入数据库，则就需要通过Eureka去查找业务数据服务所在的机器IP地址和端口号，进行调用，再由业务数据服务执行对数据库的操作，而服务之间的RPC调用，则是通过Ribbon来进行。

（2）Ribbon的作用是实现负载均衡，它首先会去Eureka去查找所有服务的注册表信息，再使用轮询算法去请求调用的目标服务。示例如图2.6所示：

通过上图可见，有5台服务器都部署了5台业务数据服务，其IP地址都不同，而交易集成需要调用业务数据服务时，就会通过Ribbon去轮询调用，轮询的依据是来自Eureka提供的注册表信息。

**（3）Hystrix是一个服务熔断组件，作用是当发现服务调用超时，主动熔断调用，防止服务卡死等待，起到一定的并发容灾作用。假设在一个购物商城业务场景中，每产生一个订单就增加5个积分，订单服务需要调用积分服务，**现在假设积分服务自己最多只有 100 个线程可以处理请求积分服务不幸的挂了，每次订单服务调用积分服务的时候，都会卡住几秒钟，然后抛出—个超时异常。就是微服务架构中恐怖的服务雪崩问题。这么多服务互相调用，要是不做任何保护的话，某一个服务挂了，就会引起连锁反应，导致别的服务也挂。但是我们思考一下，就算积分服务挂了，订单服务也可以不用挂，结合业务来看：支付订单的时候，只要把库存扣减了，然后通知仓库发货就可以了，如果积分服务挂了，等它恢复之后，再手工恢复数据，不一定要因为一个积分服务挂了，就直接导致订单服务完全失去作用。Hystrix就可以解决这个问题，Hystrix 会搞很多个小小的线程池，比如订单服务请求库存服务是一个线程池，请求仓储服务是一个线程池，请求积分服务是一个线程池。每个线程池里的线程就仅仅用于请求那个服务，当积分服务挂了，前端请求订单服务，订单服务还是可以正常调用库存服务扣减库存，调用仓储服务通知发货，而对于积分服务，只要Hystrix检测调用超时，就直接在线程级别进行熔断，从而防止服务卡死等待，提高并发量。另外在熔断时，还可以作降级业务处理，当并发量较高时，积分服务被熔断，可以在熔断后添加积分记录，待恢复后根据记录来进行恢复。

（4）Zuul组件，也就是微服务网关，负责服务路由。如果前端、移动端要调用后端系统，统一从 Zuul 网关进入，再由 Zuul 网关转发请求给对应的服务。

在实际使用过程中，并不是所有的组件都需要使用，Ribbon组件与RPC调用通常由Feign来进行，Feign是将Ribbon与HTTP网络调用结合后的Spring Cloud组件。在一般中小型企业中，很少使用熔断，因为熔断的维护成本很高，而是直接使用请求超时的用户提示。

2.3 其他技术栈介绍

# 2.3.1 开发框架Spring Boot分析

Spring Boot简化了基于Spring的应用开发，通过少量的代码就能创建一个独立的、产品级别的Spring应用。 Spring Boot为Spring平台及第三方库提供开箱即用的设置，多数Spring Boot应用只需要很少的Spring配置。

Spring Boot其自身很好的融入了MVC模型、更清晰进行Controller、Service、Dao代码分层，采用简化的XML配置文件与注解，管理更为方便。使用Spring Boot可以实现很好的代替传统的SSH、SSM等企业级开发框架。

# 2.3.2 数据库访问组件Mybatis、C3P0分析

Mybatis是一个较为灵活的数据库访问组件，其内部原理仍然是采用的JDBC操作，其主要功能是将SQL语句从代码中抽离出来，而卸载xml配置文件中。程序员手写SQL更加灵活，适用于电商行业。不同于传统的信息管理系统，电商行业的版本迭代很快，需求也经常改变，传统的将SQL语句写在代码中，每次修改SQL语句都需要重新编译整个代码，则维护成本较高，而将SQL语句放在配置文件中，只需要替换配置文件，而不用重新编译整个程序，降低了维护成本。

C3P0是一个数据连接池，它是对JDBC中的数据库连接Connection进行复用，当连接使用完毕后，不再是调用Close关闭掉，而是放入连接池缓存起来，下次使用直接从连接池获取，从而提高数据库访问效率。

# 2.3.3 缓存组件Redis分析

Redis是一个NO-SQL数据库，用于缓存场景，支持存入string、hash、set、zset、list的排序，相对传统的Memache缓存数据结构更加丰富。对于一些查找频率高，更新频率低的信息，使用直接Redis存入内存，而不用通过数据库去硬盘查找，则明显提高了访问速度。

# 2.3.4 依赖包管理Maven说明

Maven的作用是实现外部jar包的自动化管理，Maven会

建立自己的jar包仓库，在程序中，只需要集成Maven的配置文件即可，填写jar包的groupId与ArtifactId即可。如数据库连接时，不用再去人为的将依赖jar包拷贝到项目中，使得程序结构管理更加方便。

# 2.3.5 秒嘀科技短信验证平台

秒嘀科技是一家提供短信验证码、语音验证码等验证手段的科技公司，其接入方便，使用POST请求，并且带有64位Hash加密签名，安全可靠。

由于本次设计开发的服装网站，注册账号采用了手机号。为了避免用户填写虚假的手机号，他人的手机号，提高注册用户的身份真实性，同时为了兼顾成本，采用了爬虫与HttpURLConnection接入了第三方的短信验证。

# 2.3.6 高德地图IP定位

高德地图IP定位，网站不同于手机APP，在PC端不能调用GPS进行定位，也无法调用手机号运营商基站的网络三角定位，故只能使用IP地址来进行定位。高德地图的IP精确地位，是基于其长年的数据服务累计的IP地址数据与地理位置数据，本次设计为了实现PC端定位功能，使用其IP定位。

# 2.3.7 技术Zxing、Gson、HttpURLConnection介绍

Zxing是一个二维码生成工具、Gson是一个Json数据格式化工具、HttpURLConnection是一个Java语言原生的网络请求类，用于Java实现网络爬虫。

1. 软件可行性研究

（1）从技术可行性的角度，在大学课程中学习了Java、JavaWeb、前端相关技术，对该设计使用到的技术都有涉猎；另外曾在实验室学习过Java后端技术，自学了Spring Boot开发框架、Mybatis数据库访问组件、C3P0数据库连接池、Redis缓存、Maven依赖包管理、Gson格式化组件、HttpURLConnection网络请求与爬虫、Html5、Css3、JavaScrip等，并且在企业实习的过程中，对本设计的核心技术Spring Cloud以及上述技术都进行过系统的再次学习，运用这些技术参与过企业级项目开发，有过较完备的实战开发经验。对于前端LayerUI前端框架、秒嘀科技短信验证平台、IP定位高德地图API平台，其官网上都有较为完备的开发文档供参考。综上，技术上是完全可行的。

（2）从法律可行性角度，服装购物网站适应时代潮流，是电商的爱好者、服装店的刚需，其网站内容健康、网站使用技术均为开源技术与自研，不会涉及相应的版权纠纷，且整个网站的研发、部署、运营都不会违反相关法律法规，故在法律上是完全可行的。

（3）从操作可行性角度，该网站采用Layerui前端框架进行开发，所实现的网站界面美观，并且带有对用户的友情界面提示，操作简单可行。

（4）从经济可行性角度，该网站开发所涉及的技术资料均来自互联网和相关书籍，使用的短信验证、地图定位均为免费接口与开源接口，并没有过多的经济开销，另外在开发的过程中，也不需要额外的付费软件、硬件支撑，使用自己的笔记本电脑，无需任何其他费用，故经济上是完全可行的。

4 需求分析

4.1 需求规范

# 4.1.1 产品背景

根据《[2018-2022年中国服装零售市场深度评估及未来发展趋势报告](https://www.chyxx.com/research/201612/477178.html)》2018年，我国服装网购市场交易规模达到7232亿元，年复合增长率将达到14%左右。2015年以来，服装行业移动端销售额迅猛增长，移动端成交额已占到总体行业销售额一半以上，移动端渗透率在2017年以来左右突破50%，时间点晚于食品、家电数码及美妆行业，但早于鞋包行业。服装行业目前的移动渗透率在所有行业里处领先地位。

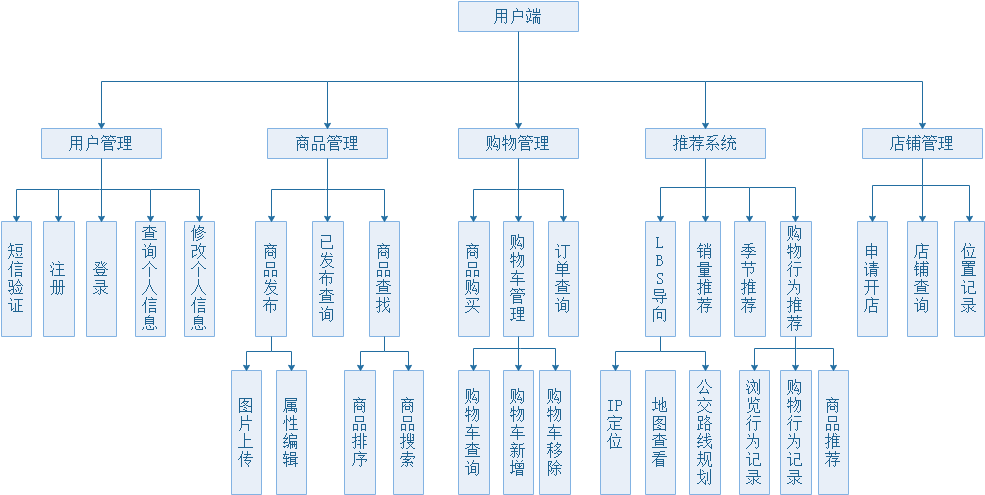
随着计算机技术与电子商务的快速发展，带有推荐功能的导向性的购物网站已成为时代主流，一方面可以根据地理位置、浏览行为、购买行为等多个维度进行商品推荐，以更加个性化、差异化的服务用户。另一方面，随着个人用户量的增多，如何调整网站的技术架构、提升高并发处理能力也成为技术难点。而二者正是困扰许多中小型电商企业的问题。为了更好的体现其产品特色，满足用户差异化，定制化，个人化购物的需求，购物导向性服装网站已经成为中小型电子商务企业、知名服装店的首要之选。

# 4.1.2 产品概述

购物导向性服装店网站设计，适用企业为：中小型电子商务企业、知名服装店。市场定位是一个B2C、C2C定位的网站，网站上既可以入驻普通商家、也支持实体店入驻，实体店自动定位。除了基本的用户信息管理以外，用户可以申请开店、发布服装、购买服装、购物车管理；系统会记录用户上次登录的地理位置信息以及上次购买行为，并且主动为用户推荐同类型、当下季节、浏览偏好的服装，同时可以附近的服装实体店，可以进行公交路线导航。本网站采用微服务分布式架构，是目前市场上主流的电商网站架构之一，相对于普通网站，先进的技术栈使用也为网站的维护、需求变更、运维部署提供了灵活的处理方式与入口。

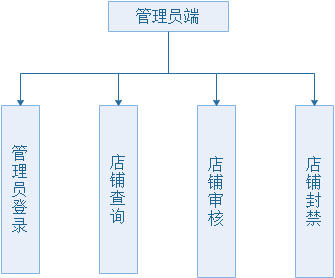
4.2 功能需求

该设计包含用户端和管理员端。其用户端功能模块如图4.1所示：

图4.1 用户端功能模块图

如上图所示，用户端主要分为4个模块，分别为：用户管理、商品管理、购物管理、推荐系统、店铺管理。对功能模块进行展开：用户管理可以分为：短信验证、注册、登录、查询个人信息、修改个人信息5个子模块；商品管理可以分为：商品发布、已发布查询、商品查找3个子模块；购物管理可以分为：商品购买、购物车管理、订单查询3个子模块；推荐系统可以分为LBS导向、销量推荐、季节推荐、购物行为推荐4个子模块；店铺管理可以分为：申请开店、店铺查询、位置记录3个子模块。

管理员端的功能主要为店铺管理，其功能模块结构如图4.2所示：

图4.2 管理员端功能模块结构图

如上图所示，管理员分为4个功能模块，分别为：登录、店铺查询、店铺审核、店铺封禁。

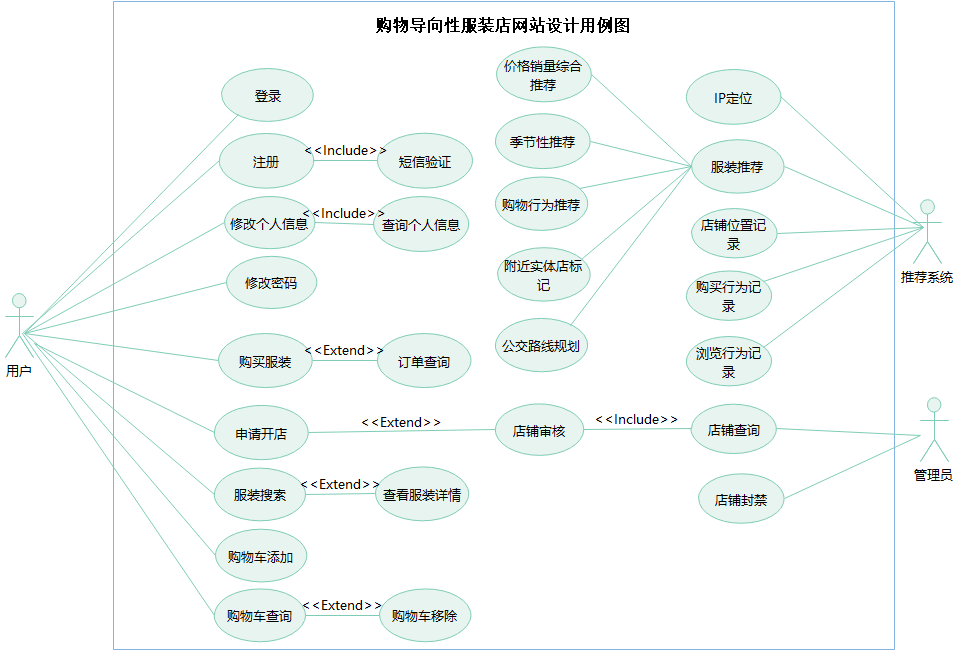
4.3 性能需求

一方面，要求界面清新美观，符合响应式布局理念，同时具有电商购物网站的商业风格与流行服饰的独特的时尚感。另一方面，要求后台接口响应时间均为毫秒级，除文件上传以外，前后端交互到前端渲染要求带给用户的感觉是响应速度快、操作流畅，架构要求采用Spring Cloud微服务分布式架构，将后台程序拆分为：业务数据服务、交易集成服务。

5 概要设计

5.1 用例分析

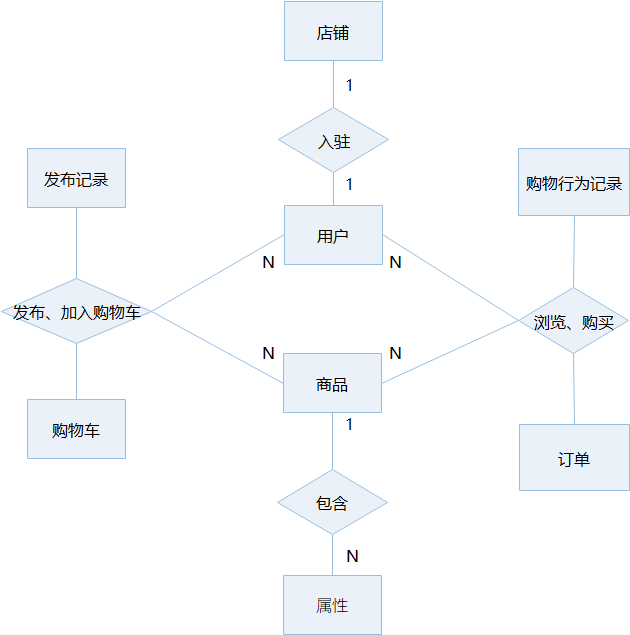
结合需求分析，分析用户与管理员关于用户管理、商品管理、推荐系统、店铺管理4个功能模块，画出用例图如5.1所示：

图5.1 购物导向性服装店网站设计用例图

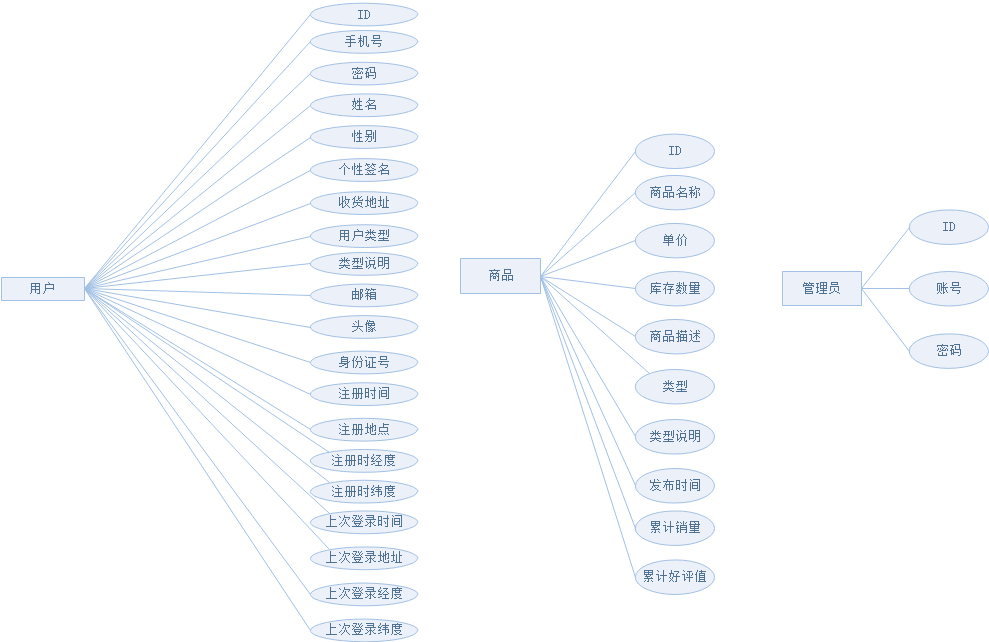
如上图，根据系统边界将用例中分为三个角色：用户、管理员、推荐系统。其中用户用例的为：短信验证、登录、注册、查询个人信息、修改个人信息、修改密码、购买服装、订单查询、申请开店、服装搜索、查看服装详情、购物车添加、购物车查询、购物车移除；推荐系统用例为：IP定位、服装价格销量综合推荐、服装季节性推荐、购物行为推荐、附近实体店标记、店铺位置记录、购物行为记录、浏览行为记录；管理员用例为：店铺审核、店铺查询、店铺封禁。

5.2 概要逻辑设计

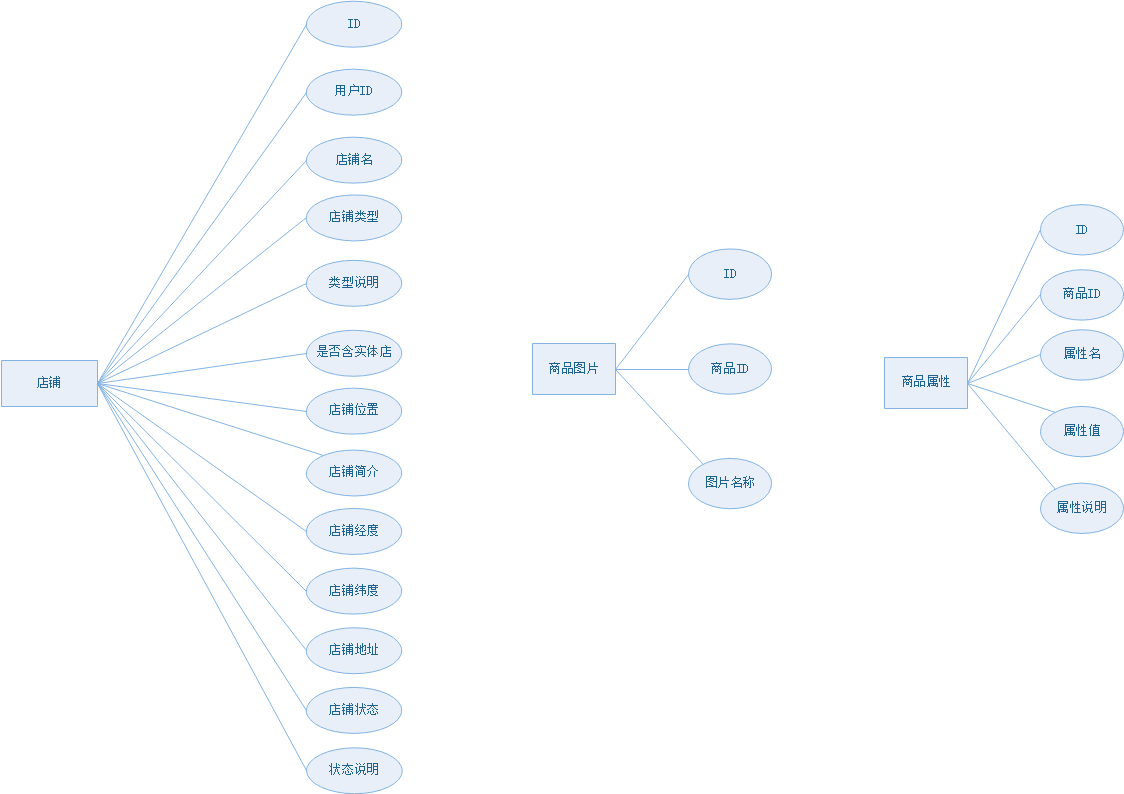
结合需求分析与用例设计，参考实体关系模型，画图E-R图，如图5.2所示：

图5.2 E-R图

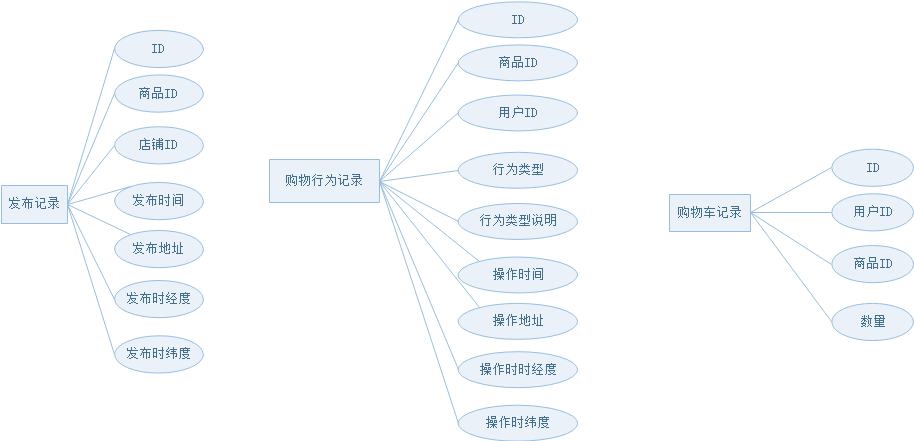
根据上述E-R图，将实体添加上属性。其中用户、商品、管理员E-R如图5.3所示：

图5.3 用户、商品、管理员相关E-R图

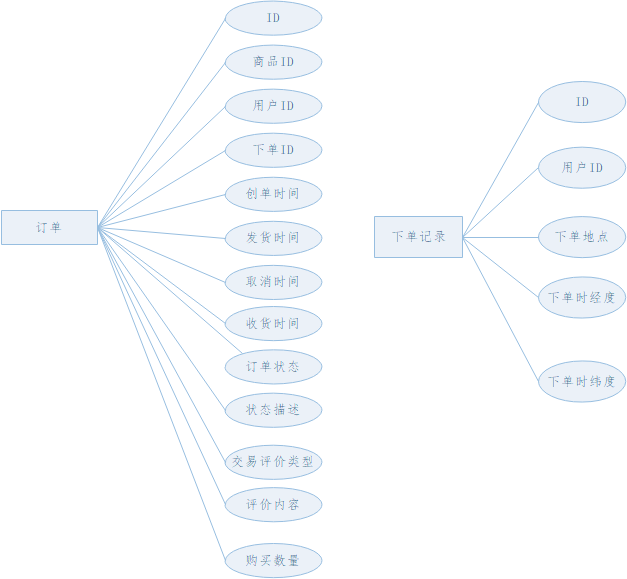
店铺、商品图片、商品属性相关E-R图如图5.4所示：

图5.4 店铺、商品图片、商品属性相关E-R图

发布记录、购物行为记录、购物车相关E-R图如图5.5所示：

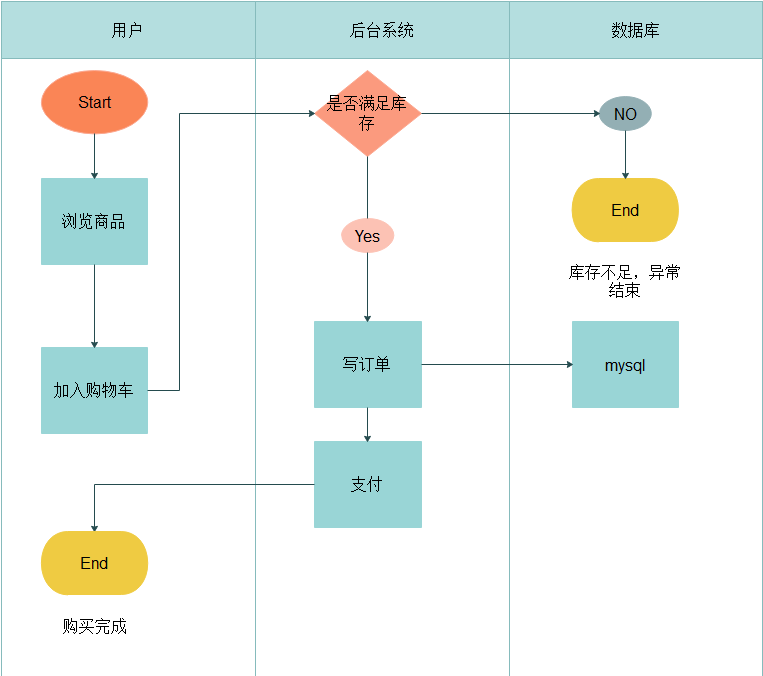
图5.5 发布记录、购物行为记录、购物车相关E-R图

订单、下单记录相关E-R图如图5.6所示：

图5.6 订单、下单记录相关E-R图

5.3 业务流程分析

本购物网站的购物流程如图5.7所示：

图5.7 购物流程图

如上图，用户进入网站，首先浏览商品，然后选择是否加入购物车，点击购买，后台事物检测是否库存满足，若库存不足则直接结束。当库存满足时系统执行减库存，写订单操作，购买完成。

6 软件详细设计

6.1 数据库设计

# 6.1.1 数据表设计

通过对E-R结合，对网站中的对象进行抽象提取，可得到以下数据表：

表6.1 用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 长度 | 允许为空 | 备注 |
| id | bigint | 20 | N | 用户ID |
| phone | char | 12 | N | 手机号 |
| password | varchar | 40 | N | 密码 |
| name | char | 20 | N | 姓名 |
| sex | char | 2 | Y | 性别 |
| signature | varchar | 100 | Y | 个性签名 |
| receive\_address | varchar | 120 | Y | 收货地址 |
| type | tinyint | 4 | N | 用户类型 |
| type\_explain | char | 20 | N | 类型说明 |
| email | char | 20 | Y | 邮箱 |
| head\_image | char | 36 | Y | 头像 |
| id\_card | char | 19 | Y | 身份证号 |
| register\_time | datetime | 0 | N | 注册时间 |
| register\_location | varchar | 120 | Y | 注册地址 |
| register\_longitude | decimal | 12,6 | Y | 注册纬度 |
| register\_latitude | decimal | 12,6 | Y | 注册经度 |
| last\_login\_time | datetime | 0 | Y | 登录时间 |
| last\_login\_location | varchar | 120 | Y | 登录地址 |
| last\_login\_longitude | decimal | 12,6 | Y | 登录纬度 |
| last\_login\_latitude | decimal | 12,6 | Y | 登录经度 |
| create\_time | timestamp | 0 | N | 创建时间 |
| update\_time | timestamp | 0 | N | 修改时间 |
| deleted | tinyint | 1 | N | 是否删除 |

用户表如图6.1所示，其中id为用户表的主键。

表6.2 店铺表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 长度 | 允许为空 | 备注 |
| id | bigint | 20 | N | 店铺ID |
| user\_id | bigint | 20 | N | 用户ID |
| name | char | 30 | N | 店铺名 |
| type | tinyint | 4 | N | 店铺类型 |
| type\_explain | char | 20 | N | 类型说明 |
| entity\_store | tinyint | 1 | N | 实体店 |
| address | varchar | 100 | Y | 所在城市 |
| introduce | varchar | 120 | Y | 店铺简介 |
| longitude | decimal | 12,6 | Y | 所在纬度 |
| latitude | decimal | 12,6 | Y | 所在经度 |
| location | varchar | 120 | N | 所在地址 |
| status | tinyint | 4 | N | 店铺状态 |
| status\_explain | char | 15 | N | 状态说明 |
| create\_time | timestamp | 0 | N | 创建时间 |
| update\_time | timestamp | 0 | N | 修改时间 |
| deleted | tinyint | 1 | N | 是否删除 |

店铺表如图6.2所示，其中id为店铺表的主键。

表6.3 管理员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 长度 | 允许为空 | 备注 |
| id | bigint | 20 | N | ID |
| account | char | 20 | N | 账号 |
| password | varchar | 40 | N | 密码 |
| create\_time | timestamp | 0 | N | 创建时间 |
| update\_time | timestamp | 0 | N | 修改时间 |
| deleted | tinyint | 1 | N | 是否删除 |

其中id为管理员表的主键。

表6.4 商品表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 长度 | 允许为空 | 备注 |
| id | bigint | 20 | N | 商品ID |
| name | char | 50 | N | 商品名称 |
| price | decimal | 18,2 | N | 单价 |
| amount | int | 11 | N | 库存数量 |
| introduce | varchar | 120 | Y | 商品描述 |
| type | int | 11 | N | 商品类型 |
| type\_explain | char | 20 | N | 类型说明 |
| introduce | varchar | 120 | N | 店铺简介 |
| add\_time | datetime | 0 | N | 发布时间 |
| sale\_volume | int | 11 | N | 累计销量 |
| comment\_sum | int | 11 | N | 好评值 |
| create\_time | timestamp | 0 | N | 创建时间 |
| update\_time | timestamp | 0 | N | 修改时间 |
| deleted | tinyint | 1 | N | 是否删除 |

商品表如表6.4所示，其中id为主键。

表6.5 商品图片表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 长度 | 允许为空 | 备注 |
| id | bigint | 20 | N | 图片ID |
| goods\_id | bigint | 20 | N | 商品ID |
| image\_name | char | 36 | N | 图片名称 |
| create\_time | timestamp | 0 | N | 创建时间 |
| update\_time | timestamp | 0 | N | 修改时间 |
| deleted | tinyint | 1 | N | 是否删除 |

商品图片表如表6.5所示，其中id为图片表主键。

表6.6 发布记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 长度 | 允许为空 | 备注 |
| id | bigint | 20 | N | 记录ID |
| goods\_id | bigint | 20 | N | 商品ID |
| store\_id | bigint | 20 | N | 店铺ID |
| release\_time | datetime | 0 | N | 发布时间 |
| release\_location | varchar | 120 | N | 发布地址 |
| release\_longitude | decimal | 12,6 | N | 发布纬度 |
| release\_latitude | decimal | 12,6 | N | 发布经度 |
| create\_time | timestamp | 0 | N | 创建时间 |
| update\_time | timestamp | 0 | N | 修改时间 |
| deleted | tinyint | 1 | N | 是否删除 |

发布记录表记录了用户与商品发布信息，如表6.6所示，其中id为主键

表6.7 商品属性表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 长度 | 允许为空 | 备注 |
| id | bigint | 20 | N | 属性ID |
| goods\_id | bigint | 20 | N | 商品ID |
| name | char | 15 | N | 属性名 |
| value | char | 36 | N | 属性值 |
| property\_explain | varchar | 100 | Y | 属性说明 |
| create\_time | timestamp | 0 | N | 创建时间 |
| update\_time | timestamp | 0 | N | 修改时间 |
| deleted | tinyint | 1 | N | 是否删除 |

商品属性表记录了商品与其属性，如表6.7所示，其中id为主键。

表6.8 订单记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 长度 | 允许为空 | 备注 |
| id | bigint | 20 | N | 记录ID |
| user\_id | bigint | 20 | N | 用户ID |
| location | varchar | 120 | N | 下单地点 |
| longitude | decimal | 12,6 | Y | 下单纬度 |
| latitude | decimal | 12,6 | Y | 下单经度 |
| sum\_money | decimal | 18,2 | N | 总金额 |
| create\_time | timestamp | 0 | N | 创建时间 |
| update\_time | timestamp | 0 | N | 修改时间 |
| deleted | tinyint | 1 | N | 是否删除 |

订单记录表记录了购买记录，如表6.8所示，其中id为主键。

表6.9 购物行为表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 长度 | 允许为空 | 备注 |
| id | bigint | 20 | N | 行为ID |
| user\_id | bigint | 20 | N | 用户ID |
| goods\_id | bigint | 20 | N | 商品ID |
| type | tinyint | 4 | N | 行为类型 |
| type\_explain | char | 20 | N | 行为说明 |
| operation\_time | datetime | 0 | N | 操作时间 |
| operation\_location | varchar | 120 | N | 操作地点 |
| operation\_longitude | decimal | 12,6 | N | 操作纬度 |
| operation\_latitude | decimal | 12,6 | N | 操作经度 |
| create\_time | timestamp | 0 | N | 创建时间 |
| update\_time | timestamp | 0 | N | 修改时间 |
| deleted | tinyint | 1 | N | 是否删除 |

购物行为表记录了用户的浏览、购买商品行为，如表6.9所示，其中id为主键。

表6.10 订单明细表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 长度 | 允许为空 | 备注 |
| id | bigint | 20 | N | 订单ID |
| goods\_id | bigint | 20 | N | 商品ID |
| user\_id | bigint | 20 | N | 用户ID |
| orders\_record\_id | bigint | 20 | N | 购买ID |
| add\_time | datetime | 0 | N | 创单时间 |
| deliver\_time | datetime | 0 | Y | 发货时间 |
| cancel\_time | datetime | 0 | Y | 取消时间 |
| receive\_time | datetime | 0 | Y | 收货时间 |
| status | tinyint | 4 | N | 订单状态 |
| status\_explain | char | 15 | N | 状态描述 |
| comment\_type | tinyint | 4 | N | 评价类型 |
| comment\_content | varchar | 120 | Y | 评价内容 |
| amount | int | 11 | Y | 购买数量 |
| sum\_money | decimal | 18,2 | N | 订单金额 |
| create\_time | timestamp | 0 | N | 创建时间 |
| update\_time | timestamp | 0 | N | 修改时间 |
| deleted | tinyint | 1 | N | 是否删除 |

订单明细表记录了用户购买商品时产生的订单明细，多条订单明细记录共同构成一次购买记录。订单明细表如表6.10所示，其中id为其主键。

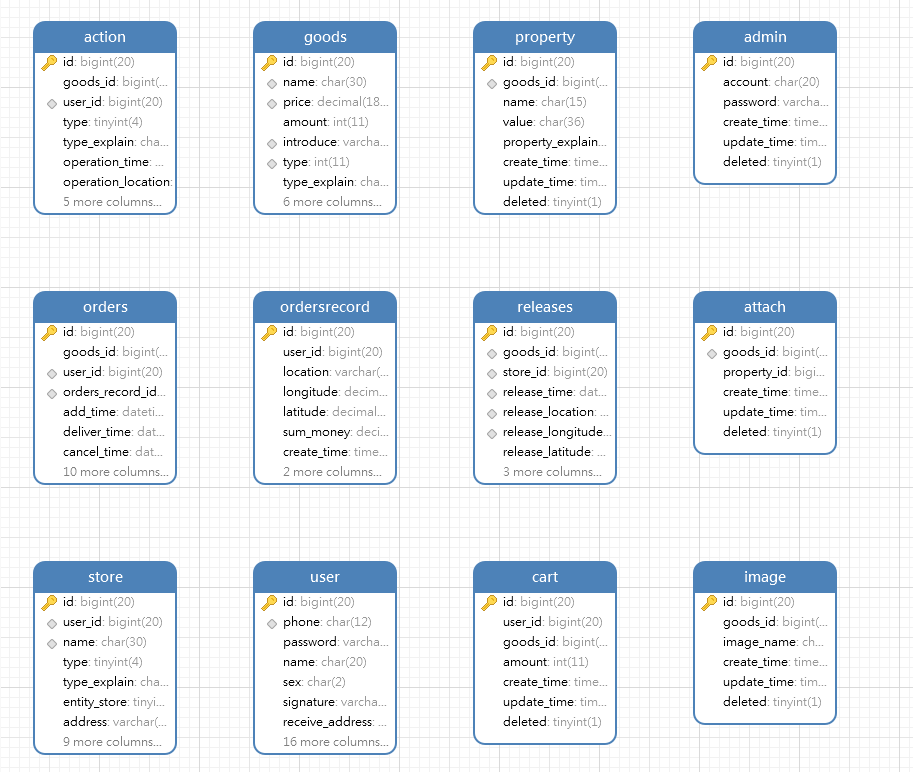
表6.11 购物车表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段类型 | 长度 | 允许为空 | 备注 |
| id | bigint | 20 | N | 购物车ID |
| user\_id | bigint | 20 | N | 用户ID |
| goods\_id | bigint | 20 | N | 商品ID |
| amount | int | 11 | N | 购物数量 |
| create\_time | timestamp | 0 | N | 创建时间 |
| update\_time | timestamp | 0 | N | 修改时间 |
| deleted | tinyint | 1 | N | 是否删除 |

购物车表记录了用户添加商品到购物车的记录，如表6.11所示，其中id为主键。

# 6.1.2 数据库物理设计

根据数据表的设计，在mysql中生成数据库，如图6.1所示：

图6.1 数据库详细设计

6.2 命名规范

# 6.2.1 数据库命名规范

（1）数据库名：英文单词全拼，所有字母均小写。

（2）数据表名：英文单词全拼，所有字母均小写。

（3）字段名称：英文单词缩写，所有字母均小写，下划线分隔。

# 6.2.2 项目文件命名规范

对项目的文件进行命名规范，方便对项目文件进行管理，有利于查阅，节省时间并且非常的高效。

（1）在后端设计中，包名均为com.clothing.xxx，严格按照dao、service、controller、util、domain、enums、consts等子文件夹存储对应的类，且类文件名首字母大写、驼峰命名。资源文件放在resources下，使用yml取代properties属性文件，均符合企业级专业项目命名规范。

（2）在前端设计中，html、css、js严格分离，存放于自己的文件夹中，方便资源的统一管理与维护。

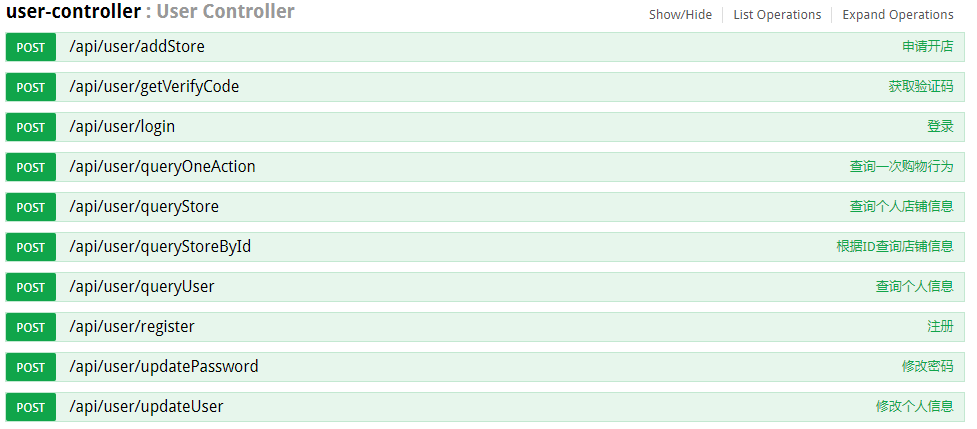
（3）项目名：英文单词全拼，所有单词均首字母大写，驼峰命名，如：ClothingDataBussinessService作为业务数据服务命名。

6.3 模块详细设计

# 6.3.1 用户管理模块

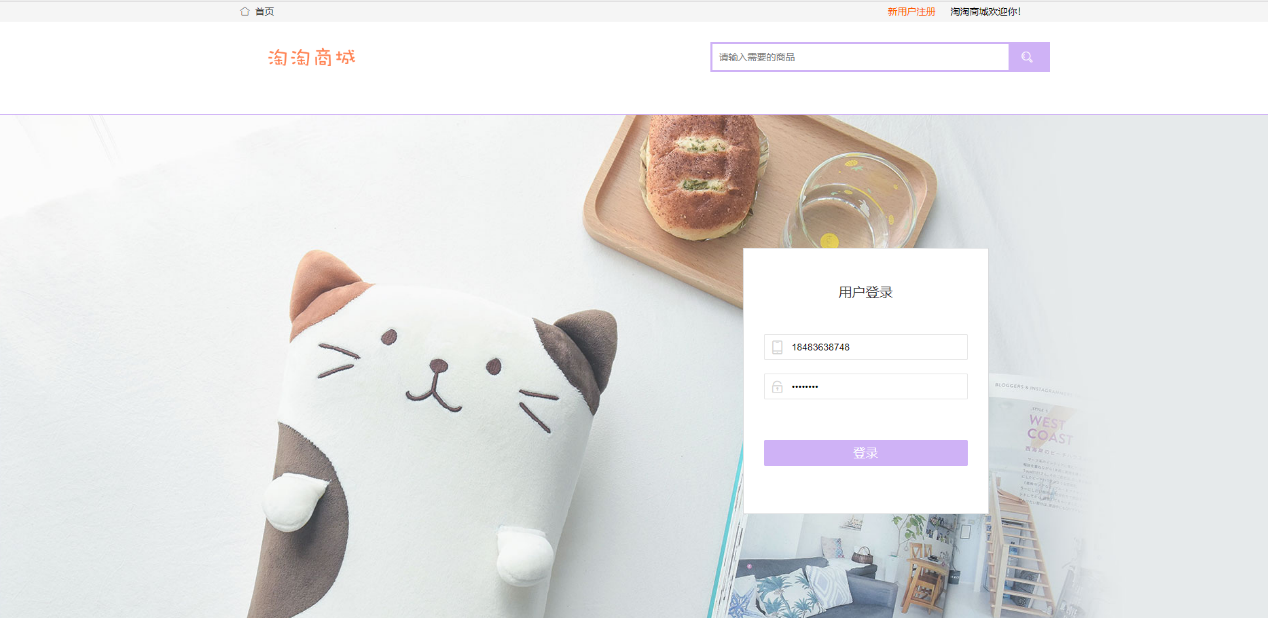
（1）模块描述：用户登录、注册、个人信息管理的相关模块。

（2）后台接口设计：获取验证码、登录、注册、查询个人信息、修改密码、修改个人信息。接口命名及请求方式参见用户控制层，如图6.2 所示：

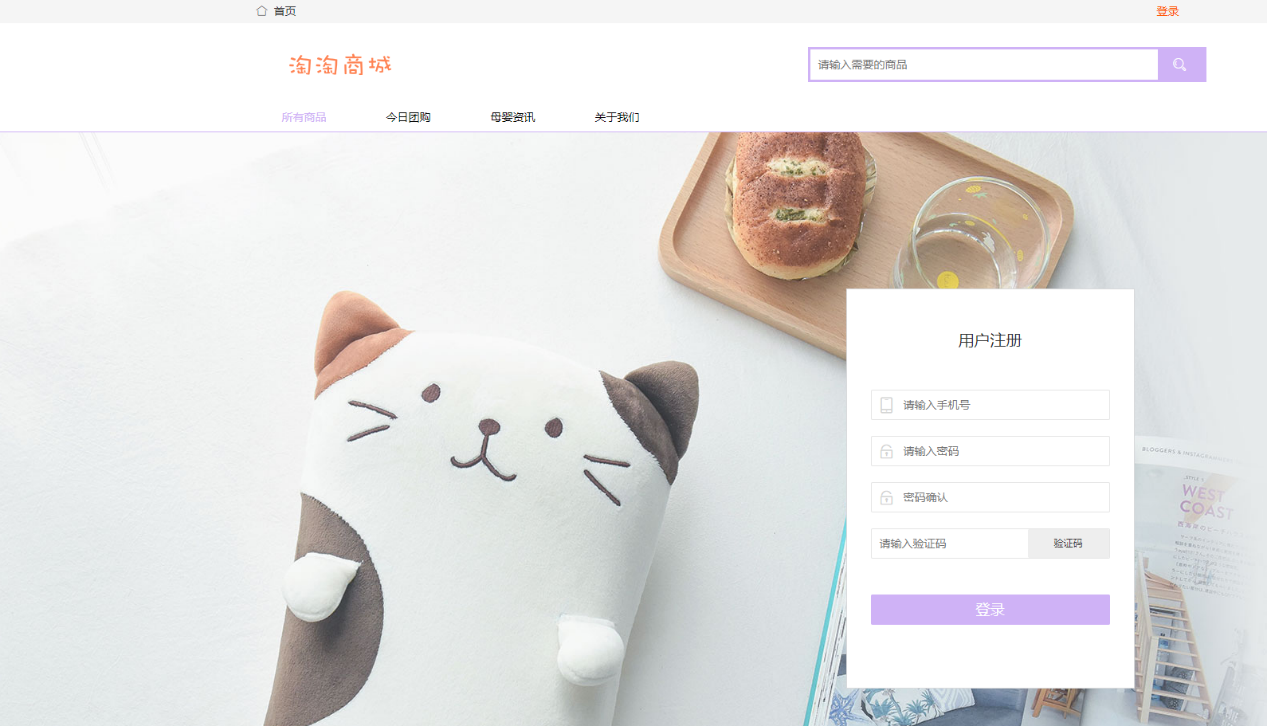
图6.2 用户控制层接口说明

（3）界面设计

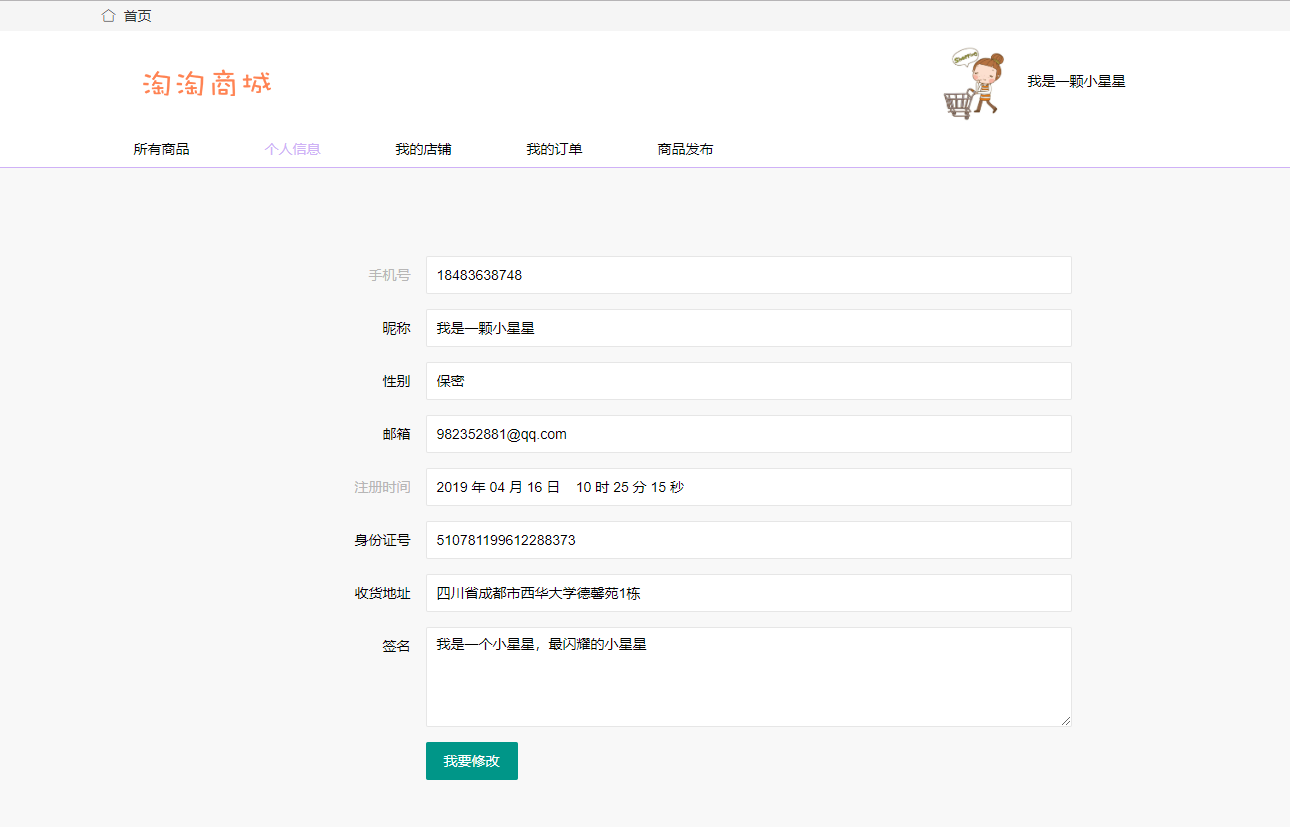
登录界面如图6.3所示：

图6.3 登录界面

注册界面如图6.4所示：

图6.4 注册界面

个人信息界面如图6.5所示

图6.5 个人信息界面

# 6.3.2 商品管理模块

（1）模块描述：商品发布、商品搜索、商品详情查看相关模块。

（2）后台接口设计：商品分页条件查询，根据ID查询商品信息，商品详情查询。查询条件分为：销量、价格、发布时间以及商品名称模糊查询。

# 总结

在这次毕业设计中我得到了许多收获，我第一次体会到Html5语言的奇妙，也发现了很多不曾注意到的细节，刚开始自己对毕业设计充满了恐惧，始终觉得自己的专业知识不够好，自己的编程也很差，在设计过程中，我总是很认真，从基本的做起，也让我收获了很多，相信在接下来的职业生涯里，我会以作为一名职业人的身份在这几个方面要求自己，严格自我。除了学到不少专业知识，也了解一些社会的现实性，包括人际交往，沟通方式及相关礼节方面的内容，对于团队开发来说，团结一致使我深有体会。团队的合作注重沟通和信任，不能不屑于做小事，永远都要保持亲和诚信，把专业理论运用到具体实践中，不仅加深我对理论的掌握和运用，还让我拥有了一次又一次难忘的开发经历，这是也是毕业设计最大的收获。

经过这次毕业设计，让我明白如果是一个完整项目的开发，必须由团队来分工合作，否则，即费力又费时还做不好这个项目。因此不管在工作中还是在生活中要和老师、同学保持良好的关系是很重要的。做事首先要学做人，要明白做人的道理，如何与人相处是现代社会的做人的一个最基本的问题。对于自己这样一个即将步入社会的人来说，需要学习的东西很多，他们就是最好的老师，正所谓“三人行，必有我师”，我们可以向他们学习很多知识、道理。遇到自己不能解决的问题时，要主动的寻求他们帮助，虚心请教。

设计虽然辛苦，但很充实，我想这就是工作。我也体会到成功地完成一个任务的那种兴奋，那种小有成就的感觉是只有置身其中的人才能体会的。只有我们亲身经历过，才能体会到别人所不能体会到的成功和喜悦。

# 致谢

在本次毕业设计过程中，遇到了很多的问题，首先自己选择了并不擅长的游戏，刚开始真的一点办法都没有，总是感觉自己做不出来，一味的抱怨，抱怨有什么用呢，还是要自己慢慢的去做，在老师和同学的指导下，我慢慢的开始做自己的毕业设计，从最开始安装软件开始，到后面慢慢的查看资料，尝试着书写代码，说实话，过程虽然很辛苦，但自己也感觉到了自己成长，首先，在这里我要感谢我的指导老师，是她每次不辞辛苦的指挥我去完善自己的软件，想方设法的为我们考虑软件的不足，让我们修改，真的非常感谢。刚开始还有点埋怨，老师每次都非常难严格的要求我们，叫我们写这写那，李老师每次都抽出自己的空余时间来知道我们，看我们软件完成的进度，指出我们设计的软件中的不足之处，让我们下来修改，真的感谢李老师。

我也想感谢一下我的父母，是他们不辞辛苦的抚养我长大，现在我就要毕业了，我会好好的找一个好工作，不让你们这么辛苦。

在这里我也要好好的感谢各位评审老师，感谢你们抽出时间来检验我的毕设成果，谢谢你们。

# 参考文献

[1] 郑逸凡.基于HTML5技术的Web游戏设计[J].计算机光盘软件与应用.2014.

[2] 刘再明.国产HTML5游戏路在何方?[J].互联网周刊.2013.

[3] 祝红涛，张钦．CSS网络大讲堂：清华大学出版社.2013.05：2-3.

[4] Mark Michaelis . C#本质论[M]. 人民邮电出版社. 2013.11.

[5] 梅洛尼，莫里森．HTML与CSS入门经典（第八版）.人民邮电出版社.2014.08.

[6] 黄扬子.基于Node. j s平台搭建REST风格Web服务[J].无线互联科技.2015.

[7] 蔡智澄,王志华.搜索引擎的主要特点及其检索策略[J].现代情报.2016.05(5)：150-152.

[8] 克罗福特. 搜索引擎：信息检索实践[M]. 机械工业出版社. 2013.6.

[9] 内格尔. C#高级编程[M]. 清华大学出版社. 2014.11.

[10] [昝辉](http://search.dangdang.com/book/search_pub.php?category=01&key2=%EA%C3%BB%D4&order=sort_xtime_desc). SEO实战编码[M]. 电子工业出版社. 2013.01.