

# 南京信息工程大学

## 本科生毕业论文(设计)



题 目 基于 NodeJS 的手机商城平台的设计与实现

学生姓名 戴肖

学 号 201633070041

院 系 应用技术学院

专 业 软件工程

指导教师 许小龙

二〇二〇 年 五月 十五 日

# 声 明

本人郑重声明：

1. 持以“求实、创新”的科学精神从事研究工作。
2. 本论文是我个人在导师指导下进行的研究工作和取得的研究成果。
3. 本论文中除引文外，所有实验、数据和有关材料均是真实的。
4. 本论文中除引文和致谢的内容外，没有抄袭他人或其他机构已经发表或撰写过的研究成果。
5. 其他同志对本研究所做的贡献均已在论文中作了声明并表示谢意。

作者签名：

日 期：2020.5.15

# 目 录

|                                 |    |
|---------------------------------|----|
| 1 绪论.....                       | 1  |
| 2 系统概论与工具选择.....                | 1  |
| 2.1 B/S 架构 .....                | 1  |
| 2.2 JavaScript 语言 .....         | 2  |
| 2.3 NodeJS .....                | 2  |
| 2.4 MongoDB 数据库 .....           | 2  |
| 2.5 VueJS .....                 | 2  |
| 2.6 Visual Studio Code 工具 ..... | 2  |
| 3 需求分析 .....                    | 3  |
| 3.1 前台系统需求分析.....               | 3  |
| 3.2 后台系统需求分析.....               | 3  |
| 3.3 可行性分析 .....                 | 4  |
| 4 系统概要设计 .....                  | 4  |
| 4.1 前台系统设计.....                 | 4  |
| 4.1.1 注册/登录功能模块.....            | 5  |
| 4.1.2 首页功能模块.....               | 5  |
| 4.1.3 商品详情功能模块.....             | 6  |
| 4.1.4 购物车管理功能模块.....            | 7  |
| 4.1.5 商品结算功能模块.....             | 7  |
| 4.1.6 订单管理功能模块.....             | 8  |
| 4.1.7 地址管理功能模块.....             | 9  |
| 4.2 后台系统设计.....                 | 9  |
| 4.2.1 登录功能模块.....               | 9  |
| 4.2.2 商品管理功能模块.....             | 10 |
| 4.2.3 运营管理功能模块.....             | 10 |
| 4.2.4 用户管理功能模块.....             | 11 |
| 4.2.5 订单管理功能模块.....             | 11 |
| 4.2.6 系统管理功能模块.....             | 12 |
| 4.3 数据库设计 .....                 | 12 |
| 4.3.1 概念结构设计 .....              | 12 |
| 4.3.2 逻辑结构设计 .....              | 16 |

|                       |    |
|-----------------------|----|
| 5 系统详细设计 .....        | 20 |
| 5.1 前台系统详细设计 .....    | 21 |
| 5.1.1 注册/登录功能模块 ..... | 21 |
| 5.1.2 首页功能模块 .....    | 22 |
| 5.1.3 商品详情功能模块 .....  | 23 |
| 5.1.4 购物车管理功能模块 ..... | 24 |
| 5.1.5 商品结算功能模块 .....  | 25 |
| 5.1.6 订单管理功能模块 .....  | 26 |
| 5.1.7 地址管理功能模块 .....  | 27 |
| 5.2 后台系统详细设计 .....    | 28 |
| 5.2.1 商品管理功能模块 .....  | 29 |
| 5.2.2 运营管理功能模块 .....  | 32 |
| 5.2.3 用户管理功能模块 .....  | 33 |
| 5.2.4 订单管理功能模块 .....  | 34 |
| 5.2.5 系统管理功能模块 .....  | 34 |
| 6 系统测试 .....          | 35 |
| 6.1 测试用表 .....        | 35 |
| 6.1.1 测试计划进度表 .....   | 35 |
| 6.1.2 模块测试用例表 .....   | 36 |
| 6.2 黑盒测试 .....        | 37 |
| 6.2.1 前台注册测试 .....    | 37 |
| 6.2.2 前台登录测试 .....    | 37 |
| 6.2.3 前台商品搜索测试 .....  | 38 |
| 6.2.4 前台商品结算测试 .....  | 38 |
| 6.2.5 前台购物车测试 .....   | 38 |
| 6.2.6 前台订单测试 .....    | 39 |
| 6.2.7 前台地址测试 .....    | 39 |
| 6.2.8 后台登录测试 .....    | 40 |
| 6.2.9 后台商品管理测试 .....  | 40 |
| 6.2.10 后台运营管理测试 ..... | 41 |
| 6.2.11 后台用户管理测试 ..... | 41 |
| 6.2.12 后台订单管理测试 ..... | 41 |
| 6.2.13 后台系统管理测试 ..... | 41 |
| 总结 .....              | 42 |

|           |    |
|-----------|----|
| 参考文献..... | 43 |
| 致谢 .....  | 44 |

# 基于 NodeJS 的手机商城平台的设计与实现

戴肖

南京信息工程大学应用技术学院，江苏 南京 210044

**摘要：**网络不断发展的今天，人们的生活方式也随之发生了改变，大家在网上交流、购物、娱乐等等，其中用的最多的设备就是智能手机，然而最困扰用户的就是如何挑选自己心仪的手机。本人经过充分的研究，通过浏览中关村在线以及各个手机商城，开始尝试把中关村在线里面手机对比参数与手机商城相结合，使得用户在浏览手机的时候，可以与其他手机进行精确的参数对比，可以在很短的时间内，挑选到自己心仪的手机，让消费者不在产生困扰。本系统在设计的时候对京东官网的布局样式以及交互进行了借鉴，操作更人性化。本系统是基于 NodeJS 环境的手机商城 PC 端，前后端使用的是目前主流的开发编译软件 VSCode，前端采用的技术框架是目前国内最火的框架 VueJS2.X 版本，为了页面的布局美观，页面部分采用了 elementUI 组件库，后端则才用比较冷门但比较新颖的网络框架 express 以及使用了很具有发展的前景的 MongoDB 数据库。前台的功能模块有：注册登录、首页、商品详情、购物车、商品结算、订单管理、地址管理；后台的功能模块有：登录、商品管理、运营管理、订单管理、用户管理、系统管理。

**关键词：**NodeJS；VueJS；MongoDB

# Design and Implementation of Mobile Phone Mall Platform Based on NodeJS

Dai Xiao

School of Applied Technology , Nanjing University of Information Science & Technology , Nanjing , 210044 , China

**Abstract:** Today, with the continuous development of the Internet, people's lifestyles have also changed accordingly. Everyone communicates online, shopping, entertaining, etc. The most used device is the smart phone, but the most troublesome user is how to choose their favorite mobile phone . After thorough research, by browsing Zhongguancun Online and various mobile phone malls, I began to try to combine the mobile phone comparison parameters in Zhongguancun Online with the mobile phone mall, so that when users browse mobile phones, they can accurately compare the parameters with other mobile phones. In a very short period of time, picking your favorite mobile phone, so that consumers are not in trouble. The design of the system borrowed from the layout style and interaction of the Jingdong official website, and the operation was more user-friendly. This system is based on the PC end of the mobile phone mall based on the NodeJS environment. The front and back ends use the current mainstream development and compilation software VSCode. The technical framework used at the front end is the most popular framework VueJS2.X version in China. The elementUI component library is used, and the backend uses the relatively unpopular but relatively novel network framework express and the very promising MongoDB database. The functional modules at the front desk are: registration login, home page, product details, shopping cart, commodity settlement, order management, address management; the functional modules at the backstage are: login, commodity management, operation management, order management, user management, system management.

**Keyword:** NodeJS;VueJS;MongoDB

## 1 绪论

本课题研究的方向是电子商务，电子商务源于发达国家。电子商务对传统行业和新兴企业注入了新的力量，带来了新的交易手段。节约成本，提高了劳动生产力，改变了传统的经营管理方式、思维和手段，也将改变公司的经营模式。第一，电子商务带来了新的销售方式，许多企业都改变了自己的销售理念，在网络上建立起销售点；第二，电子商务也影响金融证券的经营模式，电子商务已经深深的影响了经济发展；第三，电子商务同时也影响了制造业、机械行业等。

如今，我国的电子商务发展越来越迅速。随着我国计算机、互联网行业迅速崛起，电子商务也得到了极大的推广，受到了广大民众的喜爱，在年轻人中极为流行。我国发展电子商务的环境也在逐渐完善，同时国家对电子商务的政策，法规也相继出台，为电子商务的稳步发展建立了有利条件，网络带宽的增加，大大提高了网络用户的使用数量，各个地区的电子商务都取得了不小的发展，尤其在工业、交通企业等方面相对显著。上述几个方面中，可以看出我国对电子商务行业的重视，投入了大量的人力物力，不仅如此，也得到了广大的民众的喜爱和支持，在未来有不小的发展前景<sup>[1]</sup>。

## 2 系统概论与工具选择

本系统分为 PC 端的前台系统与后台管理系统。前台页面的样式主要是自己一点点手写，后台页面用的是 ElementUI 搭建而成，页面简洁，通俗易懂，而且使用的是目前前端最火的框架 VueJS，相对于传统开发而言，效率显著提高。后端的是采用 NodeJS<sup>[2]</sup>开发环境，使用的是 MongoDB 数据库，而 MongoDB 数据非传统数据库，是文档型数据库，编写方式类似于 JavaScript<sup>[3]</sup>的数组对象，总的来看，前后端采用的都是基于 JavaScript 语言编写，减少了学习成本。

### 2.1 B/S 架构

B/S 架构全称是 Browser/Server，意思用户只需要安装一个浏览器如 Chrome 或者 Firefox 浏览器或者使用本机自带的浏览器就可以使用系统，本系统无论是前台系统还是后台管理系统，均采用的是 B/S 架构，相比较 C/S 架构，无需安装额外的客户端，使用起来更加方便快捷。



## 2.2 JavaScript 语言

JavaScript 世界上使用最多的脚本语言，它诞生于 1995 年，由网景公司的布兰登·艾奇负责编写，起初命名为 LiveScript，但是由于当时的 Java 这个语言特别火，所以为了提高自身热度，决定改名为 JavaScript，类似于国内的“北大”与“北大青鸟”关系，在同时期还有其他的网页语言 VBScript、JScript 等等，但是后来均被 JavaScript 打败，所以在现在的浏览器中，只运行一种脚本语言就是 JavaScript，但是几年来延伸出了好多生态，不单单局限于浏览器，使得 JavaScript 更趋向于编程语言。

## 2.3 NodeJS

NodeJS 非语言，也非框架，是一个 JavaScript 的运行环境或者是平台，可以称为 JavaScript 虚拟机。与 PHP、JSP、Ruby 等等是平台也是语言不同，NodeJS 特点是 JavaScript 进行编程，运行在 Chrome 的 V8 引擎上，运行效率极快。当用 JavaScript 进行编程时，是不存在 BOM 和 DOM 的，只有 EcmaScript 基本的 JavaScript 语言部分。在 Node 中为 JavaScript 提供一些服务器级别的 API，如文件操作以及 http 服务等。

## 2.4 MongoDB 数据库

MongoDB<sup>[4]</sup>是为了能高效率开发 Web 应用而设计出来的数据库系统，其设计理念以极简、灵活著称，隶属于 Web 应用栈。MongoDB 与传统关系型数据库明显的不同在于，它是面向文档的，文档型数据库与 JSON 结构相似，不严地说，也可以理解为把不同的 JSON 数据存放在 MongoDB 数据库中。

## 2.5 VueJS

VueJS 于 2014 年诞生，作者是前华裔 Google 工程师尤雨溪，目前在中国是前端最火的的框架，诞生虽晚，但与老牌 Angular、React 齐肩，造成三足鼎立的局面，但是与 Angular 稍微不同的是，其核心概念是组件与双向数据流(基于 ES5 中的 defineProperty<sup>[5]</sup>来实现的)。

## 2.6 Visual Studio Code 工具

VS Code 由微软公司开发，几乎是目前前端工程师的标配，主要优势在于比 Atom 编辑器流畅，比 WebStorm 更轻量，其本身有着十分丰富的插件系统，无论我们是编辑 HTML、CSS、TS、Vue 等前端代码，还是 JAVA，Python 后端代码，我们都可以找到相对应的插件，在开发效率上有显著的提高。

## 3 需求分析

### 3.1 前台系统需求分析

用户注册及登录功能：用户通过输入账户密码进行注册，通过 `bcryptjs` 进行密码的加密，它是一个第三方加密库，用来实现 Node 环境下的密码加密，当密码验证成功后方可进入系统。

首页功能：用户或者游客可以从首页看到商品的轮播、商品的分类，用户或者游客点击轮播或者分类子项可以进入商品详情。

商品搜索功能：用户或者游客可以输入关键字进行商品搜索，商品的呈现可以是默认顺序排序、按销量升序、按销量降序、按价格升序、按价格降序等。

商品详情功能：用户或者游客可以对商品进行型号及颜色选择、商品参数对比，如果用户或者游客没有点击，系统会默认选择型号和颜色。

购物车管理功能：用户可以进入购物车进行商品的管理，包括选择商品、全部选择或者反选商品、删除商品以及商品的结算。

地址管理功能：用户可以新建地址、删除地址、修改地址、地址列表查看、选择默认地址，用户需要填写收货人、所在城市、所在街道、邮编、手机号码、是否默认。

商品结算功能：用户可以选择收货地址、新建地址、修改地址、删除地址、选择付款方式等，付款方式分为在线支付和货到付款。

订单管理功能：用户可以查看全部订单列表、查看待发货订单列表、查看已发货订单列表、确认收货订单列表。

### 3.2 后台系统需求分析

品牌管理功能：管理员可以进行品牌的添加、删除品牌、修改品牌以及查看品牌列表。

用户管理功能：管理员可以查看用户列表。

商品管理功能：管理员可以对商品进行添加、删除、修改、查看商品信息。

订单管理功能：管理员可以对订单进行查看、发货处理、查看已发货订单、查看待收货订单。

销量管理功能：管理员可以对查看各个品牌的销量。

库存管理功能：管理员可以对查看各个品牌的库存。

管理员管理功能：管理员可以添加管理员、删除管理员、修改管理员密码、查看管理员列表。

3.3 可行性分析

可行性分析是对在立项起到运营过程中可能遇到的一些因素进行分析，如社会因素、技术因素以及经济因素等等。

需求可行性：目前使用人们离不开智能手机，特别是青少年更把数码手机的关注当成一种潮流文化，又因为本系统属于电商，对于手机而言，线上的需求往往大于线下。

技术可行性：根据客户的需求，选择目前较流行且较稳定的技术，精确细化系统功能，进行可行性研究，并有效的利用现有技术确定并实现开发任务。

操作可行性：前台页面和后台页面部分使用最新的 UI 框架，大部分参考其他电商页面进行手写，界面清晰，通俗易懂，让用户很容易地清楚哪个功能是的作用，提高用户体验。

4 系统概要设计

4.1 前台系统设计

通过用户的需求分析，前台系统是主要分为注册登录功能模块、首页功能模块、商品详情功能模块、购物车管理功能模块、地址管理功能模块、商品结算功能模块、订单管理功能模块与地址管理功能模块。这些功能模块都是递进的，一个结果的输出是另一个条件的输入，功能模块如此划分，主要是把系统精确划分。如此一来，前台层次结构图如图 4.1 所示。

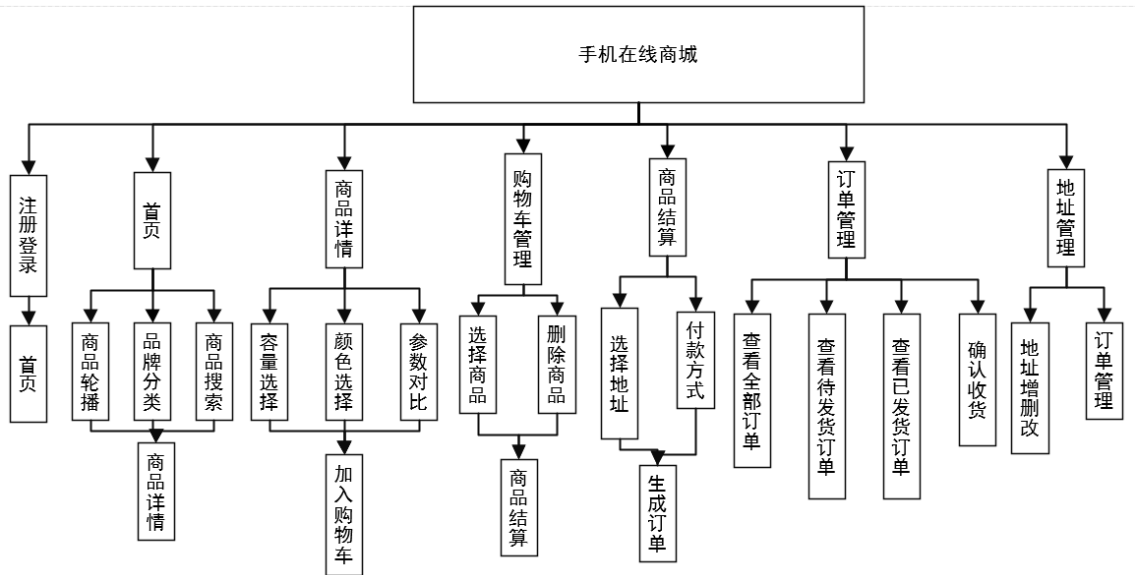


图 4.1 前台系统层次结构

4.1.1 注册/登录功能模块

当用户进行登录的时候，用户输入账户密码，然后前端把账户密码发送给后端，后端去数据库中寻找数据是否匹配，如果匹配则跳转首页，若不匹配则有相关提示；当用户进行注册的时候，用户输入用户名时，密码及其确认密码，首先要进行表单校验，如果校验不成功，会进行相关的提示，如果校验通过，如果点击注册，前端会把用户数据发送给后端，进行二次校验，如果用户名已存在，则会在页面上进行，用户名已存在提示；若全部检查通过，则跳转登录页，让用户去登录。注册/登录功能模块如图 4.2 所示。

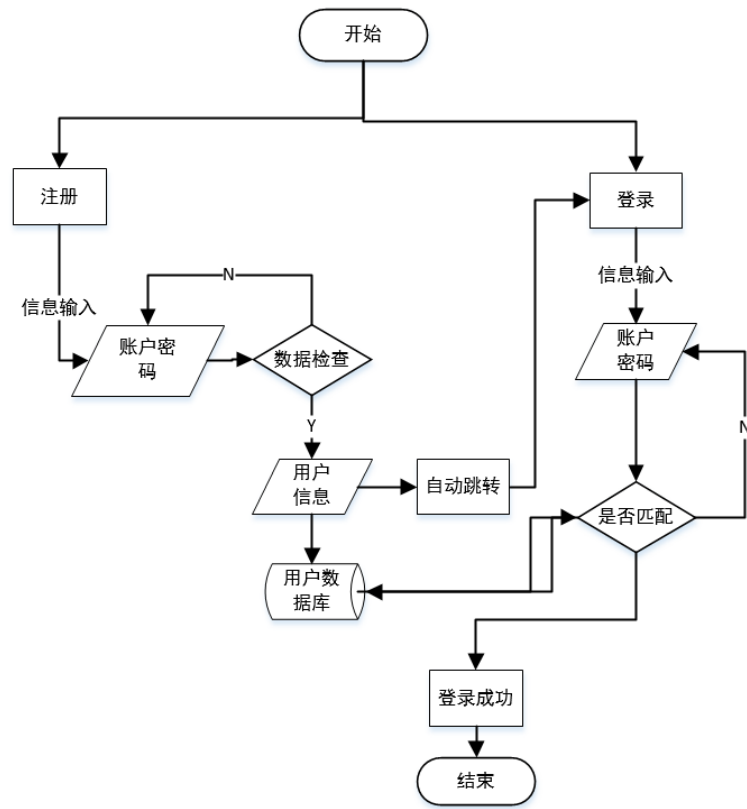


图 4.2 注册/登录功能模块流程图

4.1.2 首页功能模块

首页功能模块分为三个功能：首页轮播图、品牌的分类以及商品的搜索；首页轮播图就是轮播固定商品的宣传图，品牌分类就是各个品牌，鼠标移动到各个品牌的标签的时候，会自动显示该品牌下的全部商品；用户可以通过轮播图、品牌分类子项以及通过商品的搜索，都可以看到选择到喜欢的商品，当用户通过三种途径点击商品的时候，前端会根据用户点击商品的 id，根据商品 id 进行跳转，跳转商品详情页。当用户进行搜索的时候，如果输入关键词为空，不会发起搜索请求，如果搜索的关键词与数据库匹配，则会通过后端把匹配的商品列表传到前端，前端在通过 for 循环把商品列表渲染到页面上。商品搜索如图 4.3 所示。

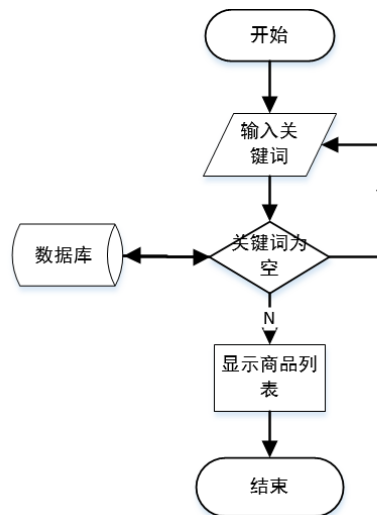


图.4.3 商品搜索的流程图

#### 4.1.3 商品详情功能模块

当进入商品详情功能模块的时候，用户选择商品的机身存储和颜色，也可以进行商品的参数的对比，若没有选择机身存储和颜色，则会选择默认选项，然后把商品加入购物车。商品详情如图 4.4 所示。

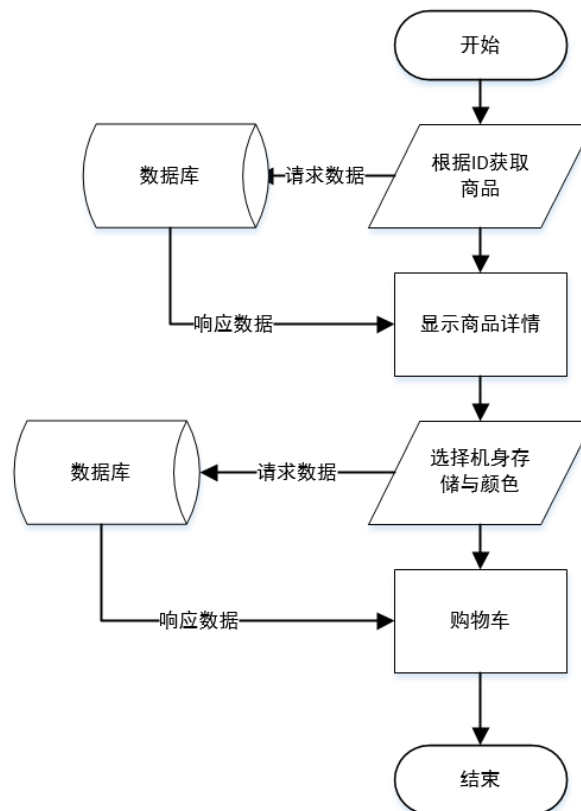


图 4.4 商品详情功能模块流程图

4.1.4 购物车管理功能模块

当用户把商品添加入购物车以后，点击跳转购物车管理列表，用户可以查看商品列表商品的详细信息包括单价、数量以及已选商品的总价等等，用户也可以对购物车里的商品进行选择、删除、全选、反选、某个商品数量的增加或减少等操作，若用户对商品没有做任何的选择，或者用户把购物车里的商品全部删除掉，系统是无法让用户进入结算页，并会利用 elementUI 组件库里的吐司弹框并发出警告提示未选择任何商品。购物车管理功能模块的流程图如图 4.5 所示。

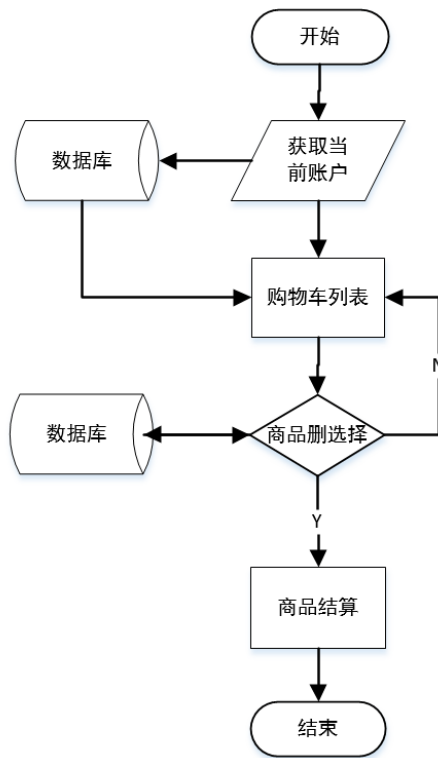


图 4.5 购物车管理功能模块流程图

4.1.5 商品结算功能模块

当用户从购物车页面进行商品的结算时候，会跳转到商品结算页，会有结算清单、用户地址选择以及付款的方式。用户必须要填写收货地址或者选择收货地址，否则无法提交表单，如果没有收货地址可以进行让用户进行新建地址、删除地址、编辑地址，设置某一地址项为默认地址。其次就是，用户则可以两种有付款方式进行选择，第一种是在线支付的方式，第二种是货到付款的方式，如果用户对付款方式没有做出任何的选择，系统会默认地为用户选择在线付款的方式为商品清单进行付款，如果所有的表单项都完整的输入完成，则系统允许用户可以提交表单，相应的，系统会自动地提示付款成功的消息提示，并生成订单。商品结算功能模块如图 4.6 所示。

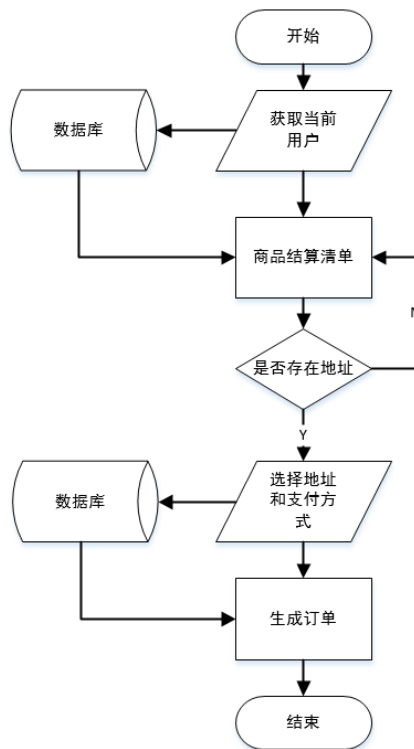


图 4.6 商品结算功能模块的流程图

#### 4.1.6 订单管理功能模块

订单管理功能模块分为全部订单，通过筛选可以分为已发货、待发货、待收货，当点击确认收货的时候，订单会从待收货转成已收货。订单管理功能模块如图 4.7 所示。

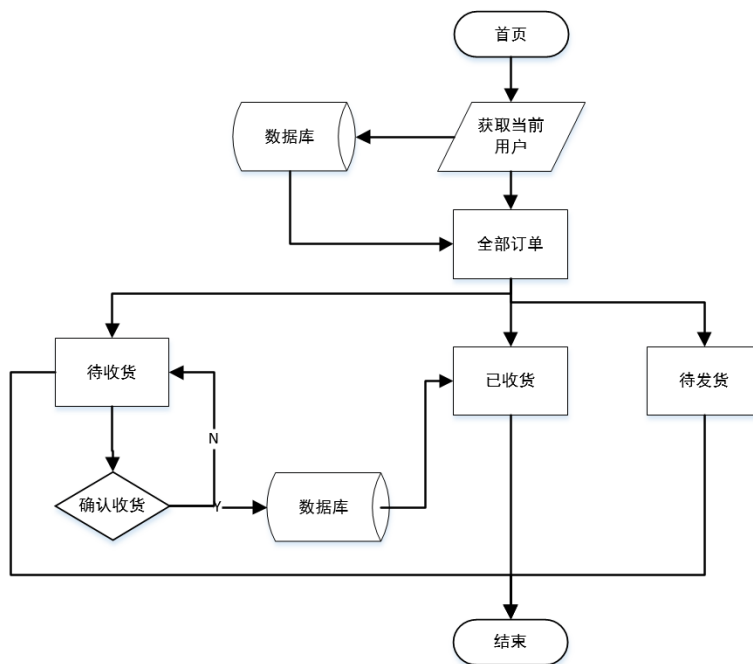


图 4.7 订单管理功能模块的流程图

4.1.7 地址管理功能模块

地址管理功能模块就是用户可以在地址管理中心进行新建地址、编辑地址、删除地址、查看地址列表，也可以设为某一地址项添加为默认地址。当地址列表中没有存在任何地址条目的时候，用户则无法进行默认地址的设置。地址管理功能模块如图 4.8 所示。

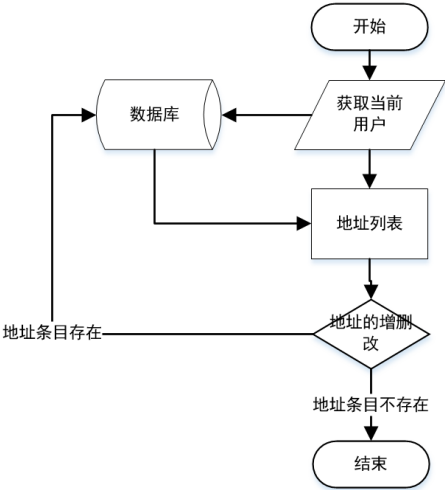


图 4.8 地址管理功能模块流程图

4.2 后台系统设计

4.2.1 登录功能模块

当管理员在进行登录的时候必须输入管理员账户和密码，前端组件进行对管理员账户和密码相应的校验，如果校验成功，则会把账户密码发送给后端，后端则会把账户与密码去数据库进行相应的匹配，如果匹配成功则登录成功，如果没有匹配成功，则会一直停留在登录页面。登录功能模块如图 4.9 所示。

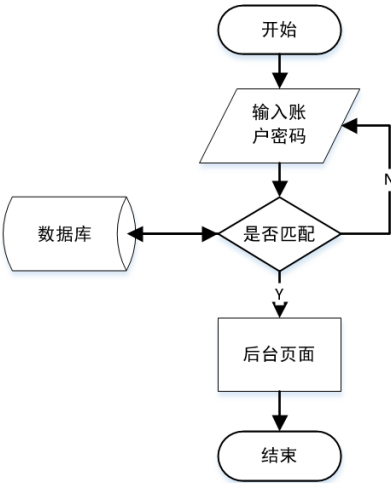


图 4.9 登录功能模块的流程图



4.2.2 商品管理功能模块

商品管理包括新建品牌、品牌列表、新建商品、商品列表、销量和库存。当管理员输入商品名，新建品牌成功后，页面会跳转到品牌列表，管理员可以查看品牌列表的详细信息，管理员也可以对品牌列表里的品牌进行编辑和删除；新建商品包括基本信息、参数配置、详细配置三大模块，当管理员新建商品并输入完整的商品信息以后会跳转到商品列表，管理员可以查看商品列表的详细信息，管理员可以对商品列表的商品进行编辑和删除。当点击销量的是，管理员可以查看是各个品牌的销售量，当点击库存的时候，管理员也可以查看各个品牌的库存。商品管理功能模块如图 4.10 所示。

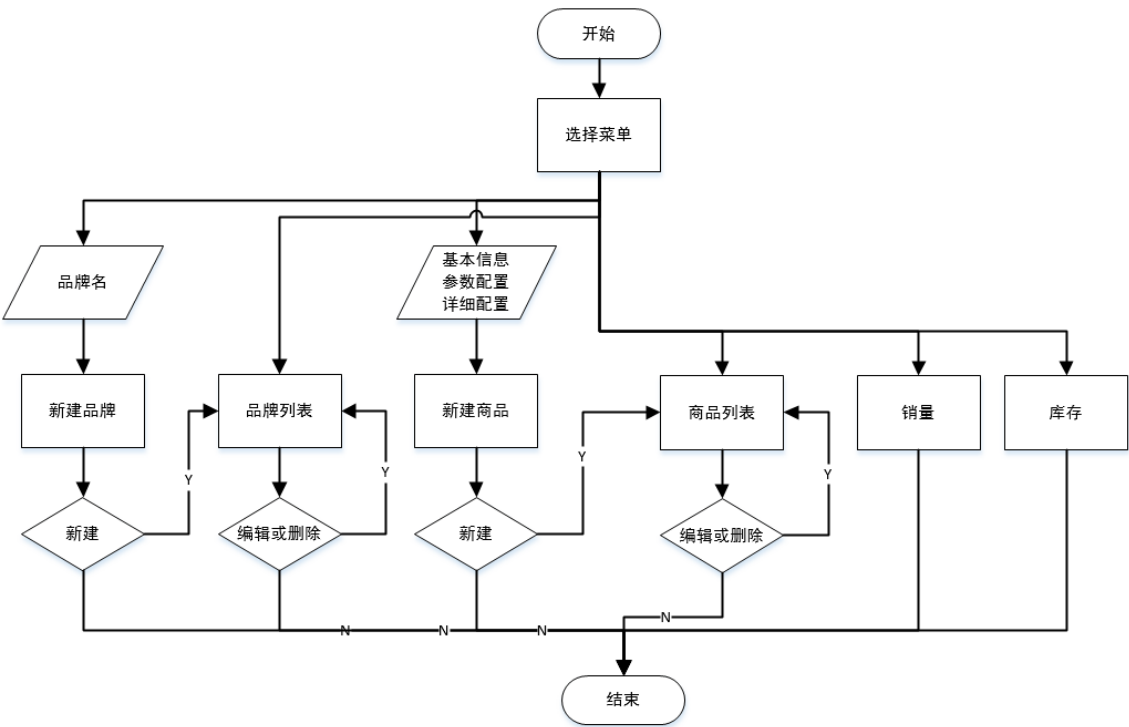


图 4.10 商品管理功能模块

4.2.3 运营管理功能模块

运营管理包括新建广告位和广告位列表，也就是对前台的轮播图的详细信息的填写，当管理员新建广告位的时候需要输入名称、与之相应广告位封面、以及管理员所要链接的到商品，链接商品是通过商品的对商品的 id 进行绑定，这样的话用户点击前台轮播图的时候就可以根据商品的 id 属性跳转到商品所对应的详情信息。当管理员输入完整的信息新建广告位成功以后，页面会跳转到广告位列表，管理员可以查看广告位列表的详细信息，也可以广告位列表的广告位进行编辑与删除。运营管理功能模块如图 4.11 所示。

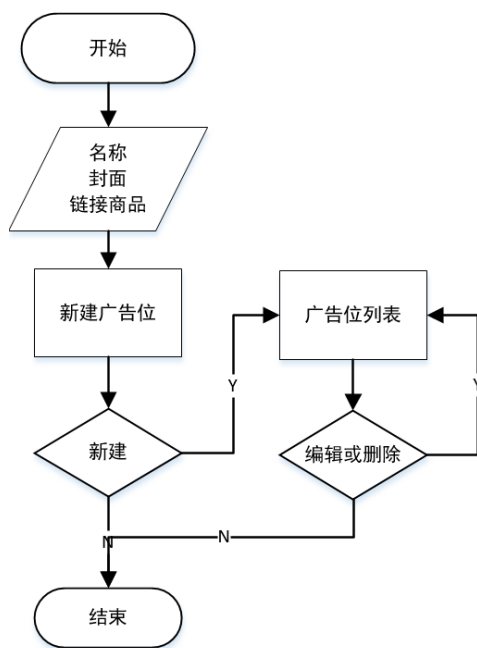


图 4.11 运营管理功能模块的流程图

#### 4.2.4 用户管理功能模块

用户管理功能模块只有用户列表，只能查看用户的 ID、用户名，管理员不能对用户的信息进行任何的详细操作。

#### 4.2.5 订单管理功能模块

管理员对订单列表可以看到订单编号、支付方式、数量、总价、及其状态。状态可分为待发货、已发货、已收货，当订单处于待发货时，管理员可以点击发货按钮，订单的状态就从待发货的状态转为已发货。订单管理功能模块如图 4.12 所示。

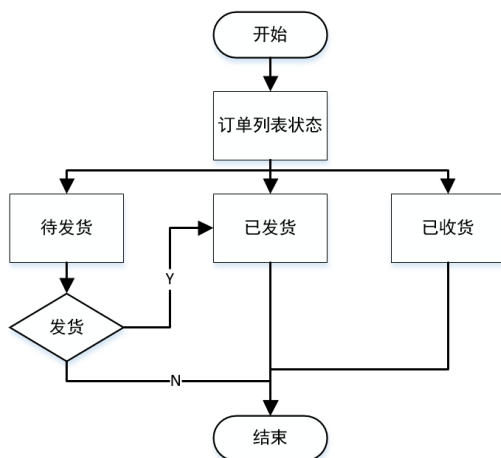


图 4.12 订单管理功能模块流程图

4.2.6 系统管理功能模块

系统管理功能模块包括新建管理员和管理员列表。新建管理员通过输入账户密码保存以后可以跳转到管理员列表，可以对管理员列表里的条目进行编辑与删除。系统管理功能模块如下图 4.13 所示。

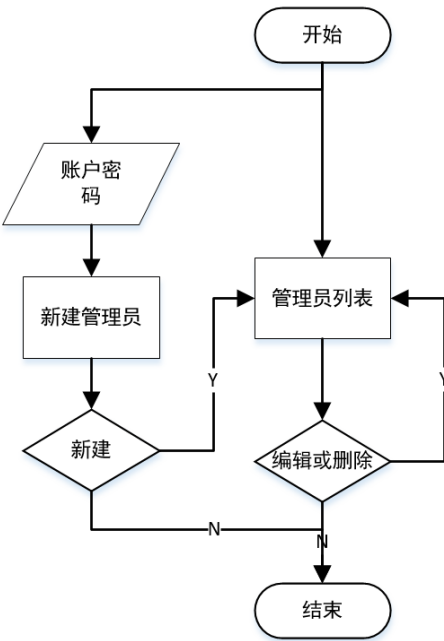


图 4.13 系统管理功能模块的流程图

4.3 数据库设计

4.3.1 概念结构设计

4.3.1.1 实体图

(1) 首页广告位实体。

包括封面图片、广告位编号、名称和商品编号。其中广告位编号为主键，商品编号为外键。首页广告位实体如图 4.14 所示。

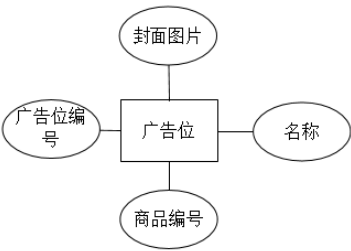


图 4.14 首页广告位实体图

## （2）品牌实体

包括品牌编号和名称。其中品牌编号为主键。品牌实体如图 4.15 所示。

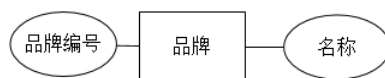


图 4.15 品牌实体图

## （3）管理员实体

包括管理员编号、密码、管理员账户。其中管理员编号为主键。管理员实体如图 4.16 所示。

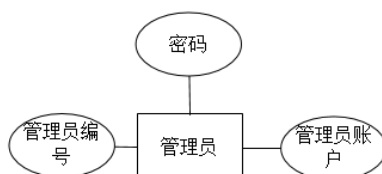


图 4.16 管理员实体图

## （4）用户实体

包括用户编号、订单编号、密码用户地址编号、购物车编号、用户名。其中用户编号为主键，订单编号、用户地址编号、购物车编号为外键。用户实体如图 4.17 所示。

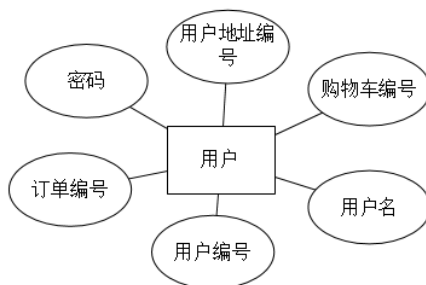


图 4.17 用户实体图

## （5）用户地址实体

包括地址编号、收货姓名、所在城市、所在街道、邮箱、联系方式、是否默认地址。其中地址编号为主键。用户地址实体如图 4.18 所示。



图 4.18 地址实体图

#### (6) 购物车实体

包括购物车编号、商品编号、商品名称、商品封面、商品价格、商品副标题、商品数量、存储容量编号、存储容量名称、机身颜色编号、机身颜色名称、是否被选中。其中购物车编号为主键，商品编号、存储容量编号、机身颜色编号为外键。购物车实体如图 4.19 所示。

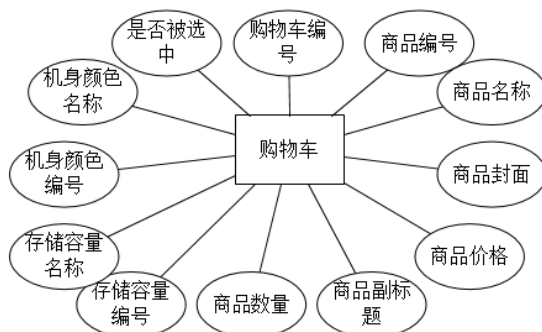


图 4.19 购物车实体图

#### (7) 订单实体

包括订单编号、地址编号、购物车编号、订单状态、订单总价、订单总数量、订单创建时间。其中订单编号为主键，地址编号、购物车编号为外键。订单实体如图 4.20 所示。

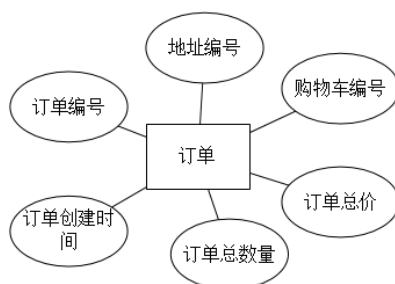


图 4.20 订单实体图

#### (8) 商品实体

包括商品编号、商品名、品牌编号、副标题、封面图片、详细图片、存储容量编号、商品参数编号。其中商品编号为主键，品牌编号、存储容量编号、商品参数编号为外键。商品实体如图 4.21 所示。

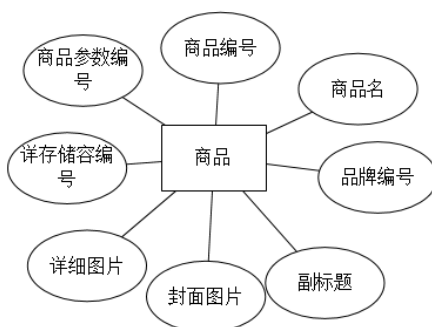


图 4.21 商品实体图

#### (9) 存储容量实体

包括存储容量编号、存储容量名称、商品原价、商品现价、机身颜色编号。其中存储容量编号为主键，机身颜色编号为外键。存储容量实体如图 4.22 所示。

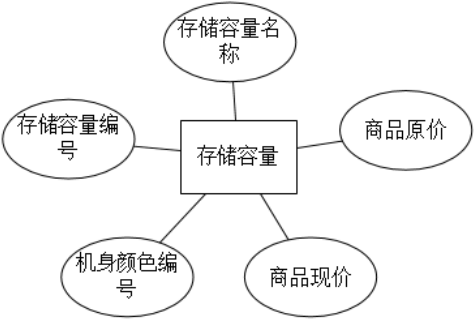


图 4.22 存储容量实体图

（10）机身颜色实体

包括机身颜色编号、机身颜色名称、库存、销量。其中机身颜色编号为主键。机身颜色实体如图 4.23 所示。

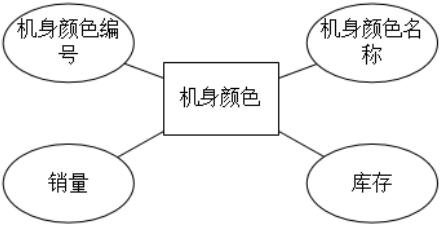


图 4.23 机身颜色实体图

（11）商品参数实体

包括商品参数编号、评分、生产日期、屏幕材质、屏幕尺寸、分辨率、屏幕比例、CPU 型号、内存大小、内存类型、闪存大小、闪存类型、电池容量、后置摄像头、前置摄像头、机身重量。其中商品参数编号为主键。商品参数实体如图 4.24 所示。

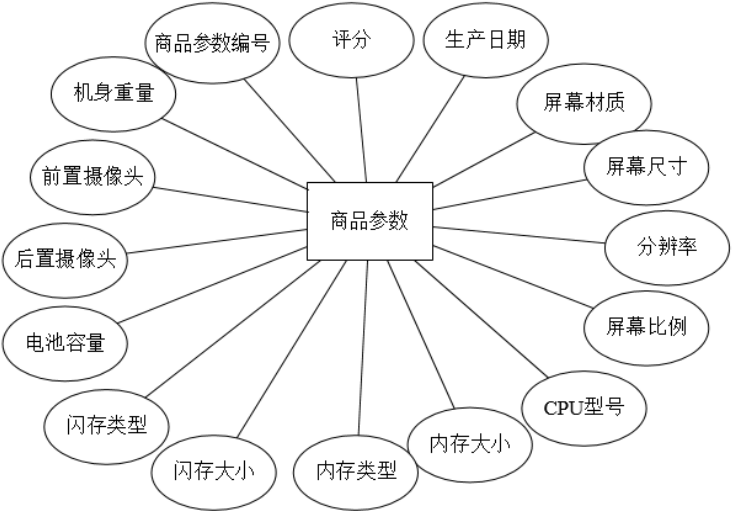


图 4.24 商品参数实体图

#### 4.3.1.2 E-R 图

数据库中实体之间的联系如下：

- (1) 一个品牌对应多个商品。
- (2) 一个广告位对应一个商品。
- (3) 一个商品对应一个商品参数。
- (4) 一个商品对应多个存储容量。
- (5) 一个存储容量对应多个机身颜色。
- (6) 一个用户对应多个购物车。
- (7) 一个用户对应多个地址。
- (8) 一个用户对应多个订单。
- (9) 一个订单对应一个地址。

实体之间关系图如图 4.25 所示。

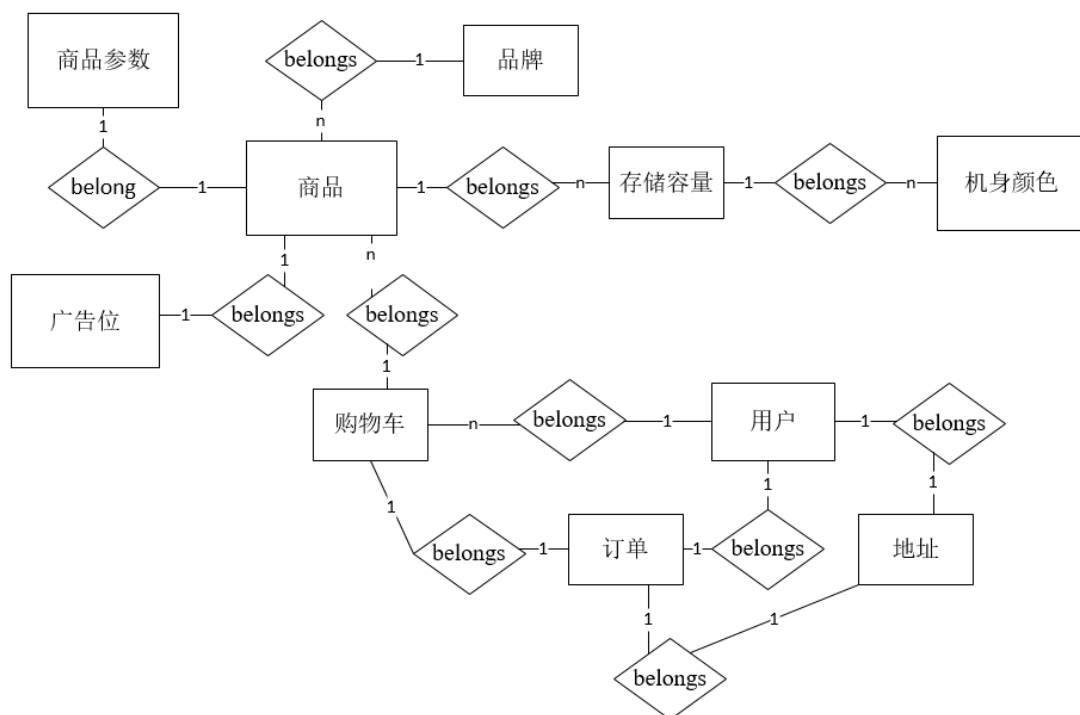


图 4.25 E-R 图

#### 4.3.2 逻辑结构设计

此系统用的是 MongoDB 数据库，首先把各个表罗列出来，最后进行合并。

- (1) 首页广告位表

表 4.1 首页广告位表

| 列名         | 数据类型   | 允许空值 | 说明   | 主外键 |
|------------|--------|------|------|-----|
| _id        | String | 否    | 默认编号 | 主键  |
| name       | String | 否    | 名称   | 否   |
| coverImage | String | 否    | 封面图片 | 否   |
| porductId  | String | 否    | 商品编号 | 外键  |

## (2) 品牌表

表 4.2 品牌表

| 列名   | 数据类型   | 允许空值 | 说明   | 主外键 |
|------|--------|------|------|-----|
| _id  | String | 否    | 默认编号 | 主键  |
| name | String | 否    | 名称   | 否   |

## (3) 管理员表

表 4.3 管理员表

| 列名       | 数据类型   | 允许空值 | 说明   | 主外键 |
|----------|--------|------|------|-----|
| _id      | String | 否    | 默认编号 | 主键  |
| username | String | 否    | 用户名  | 否   |
| password | String | 否    | 密码   | 否   |

## (4) 用户表

表 4.4 用户表

| 列名        | 数据类型   | 允许空值 | 说明    | 主外键 |
|-----------|--------|------|-------|-----|
| _id       | String | 否    | 默认编号  | 主键  |
| username  | String | 否    | 用户名   | 否   |
| password  | String | 否    | 密码    | 否   |
| addressId | String | 否    | 地址编号  | 外键  |
| cartId    | String | 否    | 购物车编号 | 外键  |
| orderId   | String | 否    | 订单编号  | 外键  |

## (5) 地址表

表 4.5 地址表

| 列名  | 数据类型   | 允许空值 | 说明   | 主外键 |
|-----|--------|------|------|-----|
| _id | String | 否    | 默认编号 | 主键  |



续表 4.5

| 列名         | 数据类型    | 允许空值 | 说明    | 主外键 |
|------------|---------|------|-------|-----|
| username   | String  | 否    | 收货姓名  | 否   |
| city       | String  | 否    | 所在城市  | 否   |
| streetName | String  | 否    | 所在街道  | 否   |
| postCode   | Number  | 否    | 邮编    | 否   |
| tel        | Number  | 否    | 联系方式  | 否   |
| isDefault  | Boolean | 否    | 是否为默认 | 否   |

## (6) 购物车表

表 4.6 购物车表

| 列名                | 数据类型    | 允许空值 | 说明     | 主外键 |
|-------------------|---------|------|--------|-----|
| _id               | String  | 否    | 默认编号   | 主键  |
| productId         | String  | 否    | 商品编号   | 外键  |
| productName       | String  | 否    | 商品名称   | 否   |
| productCoverImage | String  | 否    | 商品封面   | 否   |
| productPrice      | Number  | 否    | 商品价格   | 否   |
| productSubtitle   | String  | 否    | 商品副标题  | 否   |
| productNum        | Number  | 否    | 商品数量   | 否   |
| storageId         | String  | 否    | 存储容量编号 | 外键  |
| storageName       | String  | 否    | 存储容量名称 | 否   |
| colorId           | String  | 否    | 机身颜色编号 | 外键  |
| colorName         | String  | 否    | 机身颜色名称 | 否   |
| checked           | Boolean | 否    | 是否被选中  | 否   |

## (7) 订单表

表 4.7 订单表

| 列名        | 数据类型   | 允许空值 | 说明                                | 主外键 |
|-----------|--------|------|-----------------------------------|-----|
| _id       | String | 否    | 默认编号                              | 主键  |
| addressId | String | 否    | 地址编号                              | 外键  |
| cartId    | String | 否    | 购物车编号                             | 外键  |
| status    | Number | 否    | 状态（0 是待发货，<br>1 是已发货/待收货，2 是收货成功） | 否   |

续表 4.7

| 列名            | 数据类型   | 允许空值 | 说明                     | 主外键 |
|---------------|--------|------|------------------------|-----|
| paymentMethod | Number | 否    | 付款方式(0 是在线付款, 1 是货到付款) | 否   |
| totalPrice    | Number | 否    | 商品总价                   | 否   |
| totalNumber   | Number | 否    | 商品总数量                  | 否   |
| createTime    | String | 否    | 创建时间                   | 否   |

## (8) 商品表

表 4.8 商品表

| 列名          | 数据类型   | 允许空值 | 说明     | 主外键 |
|-------------|--------|------|--------|-----|
| _id         | String | 否    | 默认编号   | 主键  |
| name        | String | 否    | 商品名    | 否   |
| brandId     | String | 否    | 品牌编号   | 外键  |
| subtitle    | String | 否    | 副标题    | 否   |
| coverImage  | String | 否    | 封面图片   | 否   |
| detailImage | String | 否    | 详情图片   | 否   |
| storageId   | String | 否    | 存储容量编号 | 外键  |
| parameterId | String | 否    | 商品参数编号 | 外键  |

## (9) 商品参数表

表 4.9 商品参数表

| 列名             | 数据类型   | 允许空值 | 说明     | 主外键 |
|----------------|--------|------|--------|-----|
| _id            | String | 否    | 默认编号   | 主键  |
| score          | Number | 否    | 评分     | 否   |
| productionDate | String | 否    | 生产日期   | 否   |
| screenMaterial | String | 否    | 屏幕材质   | 否   |
| screenSize     | String | 否    | 屏幕尺寸   | 否   |
| resolution     | String | 否    | 分辨率    | 否   |
| screenRatio    | String | 否    | 屏幕比例   | 否   |
| cpuType        | String | 否    | CPU 型号 | 否   |
| ramSize        | String | 否    | 内存大小   | 否   |
| ramType        | String | 否    | 内存类型   | 否   |
| romSize        | String | 否    | 闪存大小   | 否   |

续表 4.9

| 列名              | 数据类型   | 允许空值 | 说明    | 主外键 |
|-----------------|--------|------|-------|-----|
| romType         | String | 否    | 闪存类型  | 否   |
| batteryCapacity | String | 否    | 电池容量  | 否   |
| rearCamera      | String | 否    | 后置摄像头 | 否   |
| frontCamera     | String | 否    | 前置摄像头 | 否   |
| weight          | String | 否    | 机身重量  | 否   |

## (10) 存储容量表

表 4.10 存储容量表

| 列名       | 数据类型   | 允许空值 | 说明     | 主外键 |
|----------|--------|------|--------|-----|
| _id      | String | 否    | 默认编号   | 主键  |
| name     | String | 否    | 存储容量名称 | 否   |
| oldPrice | Number | 否    | 商品原价   | 否   |
| nowPrice | Number | 否    | 商品现价   | 否   |
| colorId  | String | 否    | 机身颜色编号 | 外键  |

## (11) 机身颜色表

表 4.11 机身颜色表

| 列名          | 数据类型   | 允许空值 | 说明     | 主外键 |
|-------------|--------|------|--------|-----|
| _id         | String | 否    | 默认编号   | 主键  |
| name        | String | 否    | 机身颜色名称 | 否   |
| stock       | Number | 否    | 库存     | 否   |
| salesVolume | Number | 否    | 销量     | 否   |

## 5 系统详细设计

本系统的后端是用 NodeJS 进行编写的，其中安全策略则是由前后端一起完成，当用户登录的时候，前端会把用户的信息传给后端，后端再把用户的信息进行加密，用 token 挂载到请求头上，每次前端发起请求的时候，则会把携带者 token 传到后端，后端再次进行校验。前台分为游客与用户，当游客点击购买等用户操作，前端通过 axios 的响应拦截，去拦截后端传过来的 token，如果前端检测不到 token 或者 token 失效，说明不是用户，此时前端路由就会强制跳转到登录页面并让游客进行登录操作。后台系统也是这种做法，当前端检测不到或者 token 失效，也会进行强制跳转到登录页面。

## 5.1 前台系统详细设计

前台系统主要分为注册/登录功能模块、首页功能模块、商品详情功能模块、购物车管理功能模块、商品结算功能模块、订单管理功能模块、地址管理功能模块。当单纯的用于页面显示的话就是用 GET 请求，例如轮播图的轮播；与后端频繁交互的话，如增删改，则发起 POST/DELETE/PUT 请求，出于效率考量，处理数据大都是用通用 CRUD。

### 5.1.1 注册/登录功能模块

#### 5.1.1.1 注册功能

点击首页的免费注册按钮，调转到注册界面，输入用户名、密码、确认密码；当输入用户名时，后端会去数据库匹配，如果用户名匹配成功，则会提示用户名存在；若三个 input 标签都不输入，则会有相关提示，提示用户必须输入才能完成注册，注册成功后会跳转到登录页面。注册功能如图 5.1 所示。



\* 用户名

\* 密码

\* 确认密码

注册

图 5.1 注册功能

#### 5.1.1.2 登录功能

点击首页登录按钮跳转到登录页面，当用户无任何输入的时候，会有表单校验，并弹出相关提示，当输入账户密码通过校验，点击登录，则会提交表单到后端写入数据库中，如果写入成功的话，则跳转到首页。登录功能如图 5.2 所示。



欢迎登陆

\* 用户名

\* 密码

登陆

图 5.2 登录功能

### 5.1.2 首页功能模块

首页主要分为轮播图、全部品牌、商品搜索，轮播图与全部品牌都是用的 GET 请求，然后把请求的数据全部渲染到页面上。首页功能如图 5.3 所示。



图 5.3 首页图

5.1.2.1 商品搜索

(1) 当输入关键字时候，利用 vue 的特性，在搜索标签上绑定 input 方法的，并进行数据绑定数据名为 query，当 query 发生变化时，则会触发 input 方法。这里对 query 进行处理，如果用户输入的是空格，则不会向后台请求，如果输入的 query 是正常值，前端实时把 query 传到后端，后端根据正则表达式，把数据库中匹配成功的数据返回前端。输入关键字如图 5.4 所示。

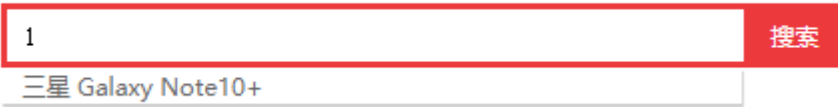


图 5.4 关键字搜索

(2) 当输入 query 点击搜索或者按回车键的时候，若根据模糊查询从数据库查询到商品数据，后端则会把商品数据渲染到前端页面上。这时候有三个功能可以选择，综合排序、销量排序（升序和降序）、价格排序（升序与降序），排序的原理就是先立一个 flag 为 true，每次点击的时候 flag 进行取反，对 flag 进行判断，利用商品数据数组的 sort 方法进行升序或者降序。商品排序如图 5.5 所示。

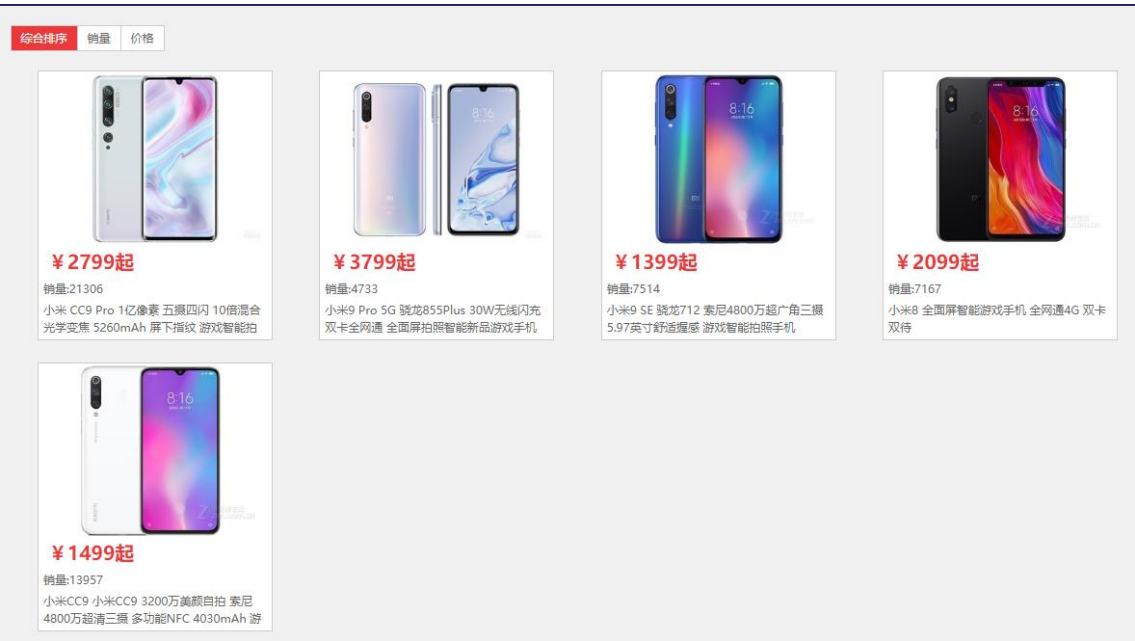


图 5.5 商品列表排序

5.1.3 商品详情功能模块

用户通过轮播图、品牌分类、商品的搜索这三种途径进行点击就可以达到商品详情页面，详情页的主要数据显示有商品的详情图片、商品的详细标题、商品的原价与现价、商品的机身

容量的选择、颜色、销量、库存。用户可根据自己喜好选择机身容量、颜色与数量加入购物车，当用户没有做出任何选择时，则会根据默认选择，把商品加入购物车；机身存储与机身颜色是父子级的关系，一个机身存储项会有独立的机身颜色，当进入详情界面的时候利用 `loadStorageAndColor` 方法进行默认加载。用户也可以查看商品的详情参数，包括 ZOL 评分、生产日期、屏幕材质、屏幕尺寸、屏幕分辨率、CPU 型号、RAM 容量、RAM 存储类型、ROM 容量、ROM 存储类型、电池容量、重量、前置摄像头、后置摄像头。用户可以输入关键词查询手机参数，与当前商品的参数进行对比。商品详情如图 5.6 所示。



小米9 Pro 5G 8GB+256GB 钛银黑 骁龙855Plus 30W无线闪充 双卡全网通 全面屏拍照智能新品游戏手机

价格 **¥3799**

选择版本 8GB+256GB 12GB+256GB

选择颜色 梦之白 钛银黑

销量: 1515件 库存: 5415件

- 1 + 加入购物车

商品参数

| 基本参数    |   | 基本参数    |   | 手机参数对比     |
|---------|---|---------|---|------------|
| ZOL评分   | 7   | ZOL评分   | 7   | 小米9 Pro 5G |
| 生产日期    | 2019/09/11  | 生产日期    | 2019/09/11  |            |
| 屏幕材质    | AMOLED  | 屏幕材质    | AMOLED  |            |
| 屏幕尺寸    | 6.39英寸  | 屏幕尺寸    | 6.39英寸  |            |
| 屏幕分辨率   | 2340x1080   | 屏幕分辨率   | 2340x1080   |            |
| CPU型号   | 骁龙855 Plus  | CPU型号   | 骁龙855 Plus  |            |
| RAM容量   | 6GB/8GB/12GB  | RAM容量   | 6GB/8GB/12GB  |            |
| RAM存储类型 | LPDDR4  | RAM存储类型 | LPDDR4  |            |
| ROM容量   | 64GB/128GB/256GB  | ROM容量   | 64GB/128GB/256GB  |            |
| ROM存储类型 | UFS 3.0   | ROM存储类型 | UFS 3.0   |            |
| 电池容量    | 4000mAh   | 电池容量    | 4000mAh   |            |
| 重量      | 196   | 重量      | 196   |            |
| 前置摄像头   | 4800万索尼主摄像头: 0.8μm, 1/2"感光面积, FOV 79° AF, 闭环马达, 支持4in1(12MP, 1.6μm) | 前置摄像头   | 4800万索尼主摄像头: 0.8μm, 1/2"感光面积, FOV 79° AF, 闭环马达, 支持4in1(12MP, 1.6μm) |            |

图 5.6 商品详情

### 5.1.4 购物车管理功能模块

当用户把商品加入购物车的时候，数据库会在用户的 `cartList` 数组插入数据对象，本人写了简版的商品列表，也就是购物车的快捷入口，即页面右侧会有一个侧边列表，可以查询到用户的已添加的商品，可以在侧边栏进行商品的删除；点击侧边栏的去购物车结算，就会跳转都购物车管理界面，用户在购物车管理页面可以选择商品，全选或反选，以及对商品数量的增加、减少以及删除。为了简化步骤，这里的所有的增删改查，都是通过更新购物车列表这个数组，也就是通过对购物车这个数组里面的条目进行增删改查得到的。购物车管理功能如图 5.7 和 5.8 所示。



图 5.7 侧边栏




图 5.8 购物车

### 5.1.5 商品结算功能模块

当用户通过购物车进行商品结算时，会跳转到商品结算页，页面有三个显示模块，分别是收货地址的选择、支付方式的选择以及送货清单查看。用户可以在收货地址进行地址的增加、地址的编辑、地址的删除，以及选择某一个条目为默认地址，这里的地址的增删改查，完全是把地址管理的整个模块内嵌到商品结算模块；用户可以在支付方式选择在线支付或者货到付款；商品清单就是选择要付款的商品，包括商品的数量，价格。当用户支付成功以后，会向数据库中用户的 orderList 插入新的订单对象，然后跳转到首页，并提示支付成功。商品结算功能如图 5.9 所示。





订单结算

收件人信息

戴肖

戴肖 江苏宿迁 宿豫区黄墩镇柳湖村九组 18361820796 223800

更多地址

默认地址





新增收货地址

支付方式

在线支付

货到付款

送货清单

| 商品   | 单价     | 数量 | 小计     |
|--|--------|----|--------|
|  <div>小米9 Pro 5G 骁龙855Plus 8GB+256GB 30W无线闪充 双卡全网通 全 钛银黑</div>    | ¥ 3799 | 1  | ¥ 3799 |
|  <div>锤子(smartisan) 坚果Pro3 骁龙855PLUS 4800万四摄 8GB+128GB 黑色</div>   | ¥ 2899 | 1  | ¥ 2899 |
|  <div>华为 HUAWEI nova 5z 麒麟810芯片 4800万AI四摄 3200 6GB+64GB 幻夜黑</div> | ¥ 1599 | 1  | ¥ 1599 |
|  <div>Redmi Note8Pro 6400万全场景四摄 液冷游戏芯 6GB+64GB 冰晶翠</div>          | ¥ 1299 | 1  | ¥ 1299 |

一共 4 件商品 总价: ¥9596

寄送至: 江苏宿迁 宿豫区黄墩镇柳湖村九组 收货人: 戴肖 18361820796

提交订单

图 5.9 商品结算

### 5.1.6 订单管理功能模块

用户点击个人订单的时候，会跳转到订单管理页面，此时前端会获取用户的详细信息，并根据用户的详细信息获取到与之对应的订单列表，然后把用户的订单列表通过 for 循环的形式遍历渲染到页面上，这个时候用户可以查看到全部订单、待发货、待收货、已发货四个选项，这里的订单列表类型的分类通过取巧的方法，目的是为了减少不必要的 http 请求，首先把所有的订单类型数据全部获取到前端，然后把所有的订单类型通过 JavaScript 里面数组里的 filter 函数<sup>[7]</sup>根据后端传过来的状态码进行订单类型的过滤和分解，如果状态是 0，就代表是待发货订单类型，如果状态是 1，就代表是待收货订单类型，如果状态是 2 的话，就代表着是已收货订单类型。用户可以查看到订单列表里的收货人、订单金额、支付方式为在线支付还是货到付款、订单的状态以及订单里商品的清单列表的详细信息。当用户点击收货按钮，会把当前订单的状态提交给后端，此订单的状态码则会从状态 1 转变为状态 2 即转变为已收货的状态，并且会有收货成功的提示，同时会像后端重新获取订单类型的数据，再次通过 for 循环的形式遍历渲染到页面上，订单管理的如图 5.10 所示。

|   |  |              |                               |
|---|--|--------------|-------------------------------|
| 全部订单 待发货 待收货 已收货  |  |              |                               |
| 2020-02-29 19:15:18 订单号: 5e5a47c6438a8634dc2ca50f                                 |  |              |                               |
| 收货人:戴肖  | 订单金额:¥ 3897                              | 支付方式:在线支付    | 订单状态:待发货                      |
|  | Redmi Note8Pro 6400万全场景四摄 冰翡翠            | 6GB+64GB     | x2                            |
| 总金额 ¥ 2598  |  |              | 在线支付                          |
|  | Redmi Note8Pro 6400万全场景四摄 贝母白            | 6GB+64GB     | x1                            |
| 总金额 ¥ 1299  |  |              | 在线支付                          |
| 2019-12-01 21:28:01 订单号: 5de3bfe1520971146ce84715                                 |  |              |                               |
| 收货人:戴肖  | 订单金额:¥ 1599                              | 支付方式:在线支付    | 订单状态:待收货 <a href="#">点击收货</a> |
|  | 华为 HUAWEI nova 5z 麒麟810芯片 4800万AI四摄 3200 | 幻夜黑 6GB+64GB | x1                            |
| 总金额 ¥ 1599  |  |              | 在线支付                          |
| 2019-12-01 21:26:18 订单号: 5de3bf7acff0883da01870a9                                 |  |              |                               |
| 收货人:戴肖  | 订单金额:¥ 1599                              | 支付方式:在线支付    | 订单状态:已收货                      |
|  | 华为 HUAWEI nova 5z 麒麟810芯片 4800万AI四摄 3200 | 幻夜黑 6GB+64GB | x1                            |
| 总金额 ¥ 1599  |  |              | 在线支付                          |
| 2019-12-01 21:25:21 订单号: 5de3bf41cff0883da0187088                                 |  |              |                               |
| 收货人:戴肖  | 订单金额:¥ 5898                              | 支付方式:在线支付    | 订单状态:待收货 <a href="#">点击收货</a> |

图 5.10 订单管理

### 5.1.7 地址管理功能模块

当在地址管理页面，用户可以查看用户的地址列表，对地址条目进行增删改，以及选择默认地址。用户填写地址详情的时候，需填写收货人、所在城市、所在街道、手机号以及邮编。这里的增删改的发起的请求，也都是通过 PUT 方式，通过更新地址数组来进行增删改。地址管理功能如图 5.11、图 5.12 和图 5.13 所示。

新增收货地址

戴肖 默认

收货人

所在城市

所在街道

手机号

邮编

新建收货地址

\* 收货人

\* 所在城市

\* 所在街道

\* 邮编

\* 手机号码

设为默认 ☐

保存

编辑

图 5.11 新建收货地址

新增收货地址

戴肖 默认

收货人  
所在城市  
所在街道  
手机号  
邮编

编辑收货地址

\* 收货人

戴肖

\* 所在城市

江苏宿迁

\* 所在街道

宿豫区黄墩镇柳湖村九组

\* 邮编

223800

\* 手机号码

18361820796

设为默认

☒

保存

编辑

图 5.12 编辑收货地址

新增收货地址

戴肖 默认地址

收货人: 戴肖  
所在城市: 江苏宿迁  
所在街道: 宿豫区黄墩镇柳湖村九组  
手机号: 18361820796  
邮编: 223800

编辑

图 5.13 地址列表

## 5.2 后台系统详细设计

除了登录模块与前台无异外，后台系统主要分为商品管理、运营管理、用户管理、订单管理、系统管理五大功能模块。后台与前台最大的不同是，前台局部用了通用的 **CRUD**，后台完全是用了通用的 **CRUD**，不需要额外地写请求接口。

5.2.1 商品管理功能模块

商品管理包括品牌、商品、销量、库存四个子模块。

5.2.1.1 品牌模块

当管理员新建一个品牌的是时候，需要填写的品牌的名称，点击保存后，会跳转到品牌列表，管理员可以查看品牌列表，包括品牌的 id 属性以及对品牌列表里的条目进行编辑与删除。品牌模块如图 5.14 所示。

新建品牌

名称

保存

品牌列表

| ID                       | 名称    | 操作                          |
|--------------------------|-------|-----------------------------|
| 5dca2148ab9393a5e0feeb12 | 华为    | <div>编辑</div> <div>删除</div> |
| 5dca32bf47b5b9becc02ba56 | 小米    | <div>编辑</div> <div>删除</div> |
| 5dccadfadc08e52fec69b30f | 锤子    | <div>编辑</div> <div>删除</div> |
| 5dd20d5335f41e63c0a09b70 | 三星    | <div>编辑</div> <div>删除</div> |
| 5dd3474798052b4004f6c9d1 | Redmi | <div>编辑</div> <div>删除</div> |

图 5.14 品牌模块

5.2.1.2 商品模块

当管理员用新建商品的时候，需要填写商品的所有信息，因为填写的信息量比较大，所以可以随时进行表单的提交，当管理员填写对应的信息后，会跳转到商品列表，可以对商品列表的条目进行编辑与删除。商品的信息分为三个模块，分别是基本信息、参数配置以及商品的详细配。；其中基本信息包括的信息为商品的所属品牌、商品名、副标题、封面图片、详情图片；参数配置包括评分、生产日期、屏幕尺寸、屏幕分辨率、CPU 型号、RAM 容量、RAM 存储类型、ROM 容量、ROM 存储类型、电池容量、机身重量、前置摄像头、后置摄像头；详细配置包括存储容量大小、原价、现价、机身颜色、库存、销量。商品模块如图 5.15 和图 5.16 所示。

新建商品

基本信息

参数配置

详细配置

所属品牌

请选择

商品名

副标题

封面图片

+

详情图片

+

保存

图 5.15 新建商品

| ID                       | 商品名称              | 所属品牌  | 输入关键字搜索 |    |
|--------------------------|-------------------|-------|---------|----|
| 5dd163849e51cb6f1867b1d4 | 小米 CC9 Pro        | 小米    | 编辑      | 删除 |
| 5dd20bba35f41e63c0a09b54 | 小米9 Pro 5G        | 小米    | 编辑      | 删除 |
| 5dd20e0835f41e63c0a09b83 | 三星 Galaxy Note10+ | 三星    | 编辑      | 删除 |
| 5dd33f9f98052b4004f6c9bb | Redmi Note8Pro    | Redmi | 编辑      | 删除 |

图 5.16 商品列表

5.2.1.3 销量模块

销量模块是显示的是各个品牌的销量，通过对各个品牌旗下手机的销量进行计算层层汇总，然后后端把品牌列表的数据返回到前端，前端调用百度官方的 echarts 图表中的柱状图的 api，把数据通过图标的方式渲染到页面上。销量模块如图 5.17 所示。

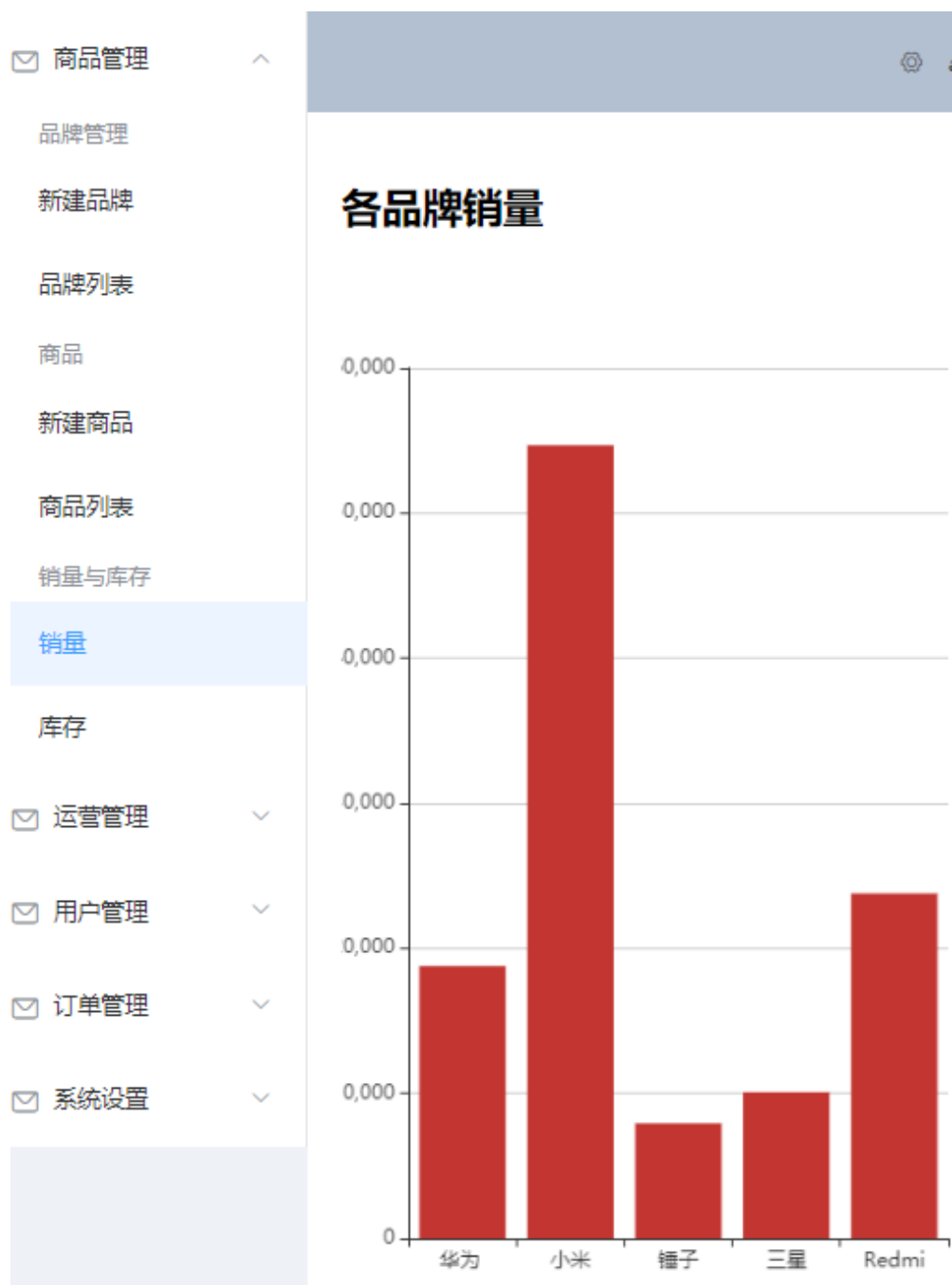


图 5.17 各品牌销量

#### 5.2.1.4 库存模块

库存模块与销量模块的实现原理基本上一样，库存模块如图 5.18 所示。

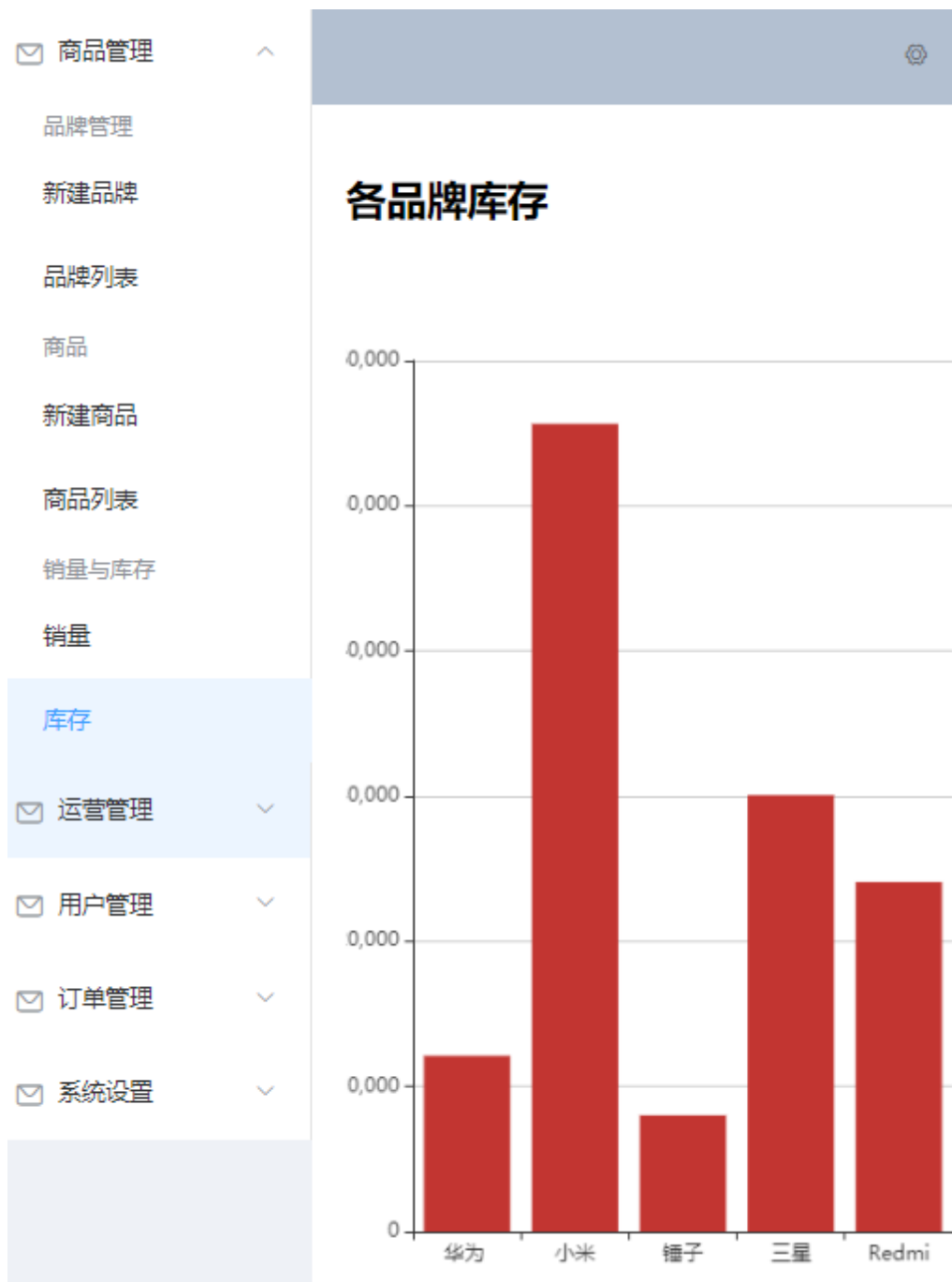


图 5.18 各品牌库存

### 5.2.2 运营管理功能模块

运营管理分为新建广告位和广告位列表两个部分，也就是前台显示的轮播图，当管理员保存的时候，会跳转到广告位列表，管理员可以对广告位列表里的条目进行编辑与删除。当用户新建广告位时，需要输入广告位的名称，广告位封面以及要链接的商品，这里根据商品的 id 的绑定进行实现的。运营管理如图 5.19 所示。

### 新建广告位

名称

广告位封面



链接商品

请选择



保存

### 广告位列表

| ID                       | 名称               | 广告位封面 | 操作                                    |
|--------------------------|------------------|-------|---------------------------------------|
| 5dd21845e8f8ac76dcd2e933 | 小米 CC9 Pro       |       | <a href="#">编辑</a> <a href="#">删除</a> |
| 5dd347ae98052b4004f6c9da | 2000万像素          |       | <a href="#">编辑</a> <a href="#">删除</a> |
| 5dd4ce51d327dc5324b445e6 | 华为 HUAWEI nova 5 |       | <a href="#">编辑</a> <a href="#">删除</a> |

图 5.19 运营管理

### 5.2.3 用户管理功能模块

用户管理模块只是纯粹地把用户的数据渲染到页面上，只包括在用户列表查看用户的 ID、用户名，为了安全起见，管理员无法对用户的信心进行任何的修改等操作。用户管理模块如图 5.20 所示。

### 用户列表

| ID                       | 用户名      |
|--------------------------|----------|
| 5de11b1ae426e53120a06879 | xiaoyue  |
| 5de3274e6be99d37c802434e | xiaoyue1 |

图 5.20 用户列表



5.2.4 订单管理功能模块

管理员可以查看订单列表，包括订单编号、支付方式、商品的数量，订单的总价、订单的状态以及详细条目。订单状态包括已发货、待发货、已收货。详细条目可以看到商品的 ID、商品名称、机身颜色、数量以及商品价格。管理员可以处理订单，进行商品的发货处理；这里的订单状态是根据状态码的改变实现的。订单管理如图 5.21 所示。

订单列表

| 订单编号                       | 支付方式 | 数量 | 总价   | 状态  |
|----------------------------|------|----|------|-----|
| > 5ea131707d0d1e25203c0f88 | 在线支付 | 4  | 9596 | 已收货 |
| 5e5a47c6438a8634dc2ca50f   | 在线支付 | 3  | 3897 | 发货  |

商品ID

5de5dfa067f9241bd4c70c74

商品名称

Redmi Note8Pro

机身存储

6GB+64GB

机身颜色

冰翡翠

数量

2

商品价格

¥ 1299

商品ID

5de5dfa567f9241bd4c70c75

商品名称

Redmi Note8Pro

机身存储

6GB+64GB

机身颜色

贝母白

数量

1

商品价格

¥ 1299

图 5.21 订单列表

5.2.5 系统管理功能模块

系统管理功能模块包括新建管理员与管理员列表。当管理员新建管理员，输入管理员账号与管理员密码，当表单数据输入完成以后，保存提交表单，保存完成时，则页面会跳转到管理员列表，管理员在管理员列表进行管理员页面的查看，可以查看管理员的 id 与管理员名称，管理员可以在管理员列表界面进行管理员条目编辑与删除；但是当管理员进行编辑的时候，只能对管理员的用户名称进行编辑，是无法对管理员的密码进行编辑的，因为密码加密采用的是 bcryptjs，是不可逆的，所以只可以重新输入密码。系统管理功能如图 5.22 所示。

新建管理员

用户名

密码

保存

管理员列表

| ID                       | 用户名   | 操作              |
|--------------------------|-------|-----------------|
| 5dd34da2b149b245001bc3c5 | admin | <div>编辑删除</div> |

编辑管理员

用户名

密码

保存

图 5.22 管理员管理

6 系统测试

在一个系统完成的每个阶段，系统测试都是必不可少的一环，一个系统的完成所需的时间，测试可以占到百分之三十左右。测试根据系统的需求文档进行数据的检测，一个系统是否合格的标准取决于测试。

6.1 测试用表

6.1.1 测试计划进度表

系统的测试需要一个有清晰明朗的计划，为之后具体实施提供方向。测试计划如表 6.1 所示。

表 6.1 测试计划进度表

| 序号 | 测试内容        | 时间(天) | 开始日期      | 结束日期      | 实施者 |
|----|-------------|-------|-----------|-----------|-----|
| 1  | 系统单元测试      | 2     | 2020.3.1  | 2020.3.3  | 戴肖  |
| 2  | 系统集成测试      | 3     | 2020.3.4  | 2020.3.7  | 戴肖  |
| 3  | 系统确认测试      | 3     | 2020.3.8  | 2020.3.11 | 戴肖  |
| 4  | 系统 Alpha 测试 | 4     | 2020.3.12 | 2020.3.16 | 戴肖  |
| 5  | 系统 Beta 测试  | 4     | 2020.3.17 | 2020.3.21 | 戴肖  |

### 6.1.2 模块测试用例表

在测试的时候需要对各个功能模块进行测试，出于严格需要，对每个功能模块建立一个测试用例，测试功能模块分为前台注册/登录模块、前台商品搜索模块、前台购物车模块、前台商品结算模块、前台订单模块、前台地址管理模块、后台登录模块、后台商品管理模块、后台运营管理模块、后台用户管理模块、后台订单管理模块、后台系统管理模块。测试用例如表 6.2 所示。

表 6.2 测试用例表

| 序号 | 测试模块及功能名  | 测试内容                              | 测试结果 |
|----|-----------|-----------------------------------|------|
| 1  | 前台注册/登录模块 | 确保账号唯一性以及密码的加密                    | 可实现  |
| 2  | 前台商品搜索模块  | 确保对关键词的正则搜索以及对搜索商品列表的排序方式         | 可实现  |
| 3  | 前台购物车模块   | 确保购物车的商品选择、全选、反选的成功               | 可实现  |
| 4  | 前台商品结算模块  | 确保结算商品是对用户地址以及用户付款方式进行检查          | 可实现  |
| 5  | 前台订单模块    | 查看全部订单、待发货订单、待收货订单已发货订单以及确保是否收货成功 | 可实现  |
| 6  | 前台地址管理模块  | 查看地址列表以及对地址进行增删改、选择默认地址           | 可实现  |
| 7  | 后台登录模块    | 确保账号唯一性以及密码的加密                    | 可实现  |
| 8  | 后台商品管理模块  | 对各个品牌、单个商品订单增删改查以及各个品牌的销量与库存的查看   | 可实现  |
| 9  | 后台运营管理模块  | 对首页广告位的增删改查                       | 可实现  |
| 10 | 后台用户管理模块  | 对用户列表进行查看                         | 可实现  |
| 11 | 后台订单管理模块  | 对订单列表进行查看以及进行发货处理                 | 可实现  |
| 12 | 后台系统管理模块  | 对管理员列表进行增删改查                      | 可实现  |

6.2 黑盒测试

黑盒测试是最普遍的一种测试方法，对于非开发者而言，也可以进行测试。核心理念是输入数据，返回结果，不探究内部结构。

6.2.1 前台注册测试

前台注册测试主要测试已存在的账户、表单未输入完整、不足 6 位的用户名、超过 16 位的用户名、密码和确认密码不一致、正确格式的用户名和密码。前台测试表如表 6.3 所示。

表 6.3 前台注册测试表

| 序号 | 测试用例        | 预期结果      | 实际结果      |
|----|-------------|-----------|-----------|
| 1  | 已存在的账户      | 用户名已存在    | 用户名已存在    |
| 2  | 表单未输入完整     | 请完善表单内容   | 请完善表单内容   |
| 3  | 不足 6 位的用户名  | 请输入正确用户名  | 请输入正确用户名  |
| 4  | 超过 16 位的用户名 | 请输入正确用户名  | 请输入正确用户名  |
| 5  | 密码和确认密码不一致  | 两次输入密码不一致 | 两次输入密码不一致 |
| 6  | 正确格式的用户名和密码 | 注册成功      | 注册成功      |

6.2.2 前台登录测试

前台登录测试的测试用例主要分为表单未输入完整、不足 6 位的用户名、超过 16 位的用户名、正确的用户名和密码、正确的用户名，错误的密码、错误的用户名，正确的密码、错误的用户名，错误的密码。前台登录测试如表 6.4 所示。

表 6.4 前台登录测试表

| 序号 | 测试用例         | 预测结果     | 实际结果     |
|----|--------------|----------|----------|
| 1  | 表单未输入完整      | 请完善表单内容  | 请完善表单内容  |
| 2  | 不足 6 位的用户名   | 请输入正确用户名 | 请输入正确用户名 |
| 3  | 超过 16 位的用户名  | 请输入正确用户名 | 请输入正确用户名 |
| 4  | 正确的用户名和密码    | 登录成功     | 登录成功     |
| 5  | 正确的用户名，错误的密码 | 用户名或密码错误 | 用户名或密码错误 |
| 6  | 错误的用户名，正确的密码 | 用户名或密码错误 | 用户名或密码错误 |
| 7  | 错误的用户名，错误的密码 | 用户名或密码错误 | 用户名或密码错误 |

6.2.3 前台商品搜索测试

商品搜索测试用例主要分为商品名称存在的关键字搜索、商品名称不存在的关键字搜索、商品列表销量升序、商品列表销量降序、商品列表价格升序、商品列表价格降序、商品列表综合排序。前台商品搜索测试如表 6.5 所示。

表 6.5 前台商品搜索测试表

| 序号 | 测试用例          | 预测结果       | 实际结果       |
|----|---------------|------------|------------|
| 1  | 商品名称存在的关键字搜索  | 相关商品列表显示   | 相关商品列表显示   |
| 2  | 商品名称不存在的关键字搜索 | 暂无结果       | 暂无结果       |
| 3  | 商品列表销量升序      | 商品根据销量升序排列 | 商品根据销量升序排列 |
| 4  | 商品列表销量降序      | 商品根据销量降序排列 | 商品根据销量降序排列 |
| 5  | 商品列表价格升序      | 商品根据价格升序排列 | 商品根据价格降序排列 |
| 6  | 商品列表价格降序      | 商品根据价格降序排列 | 商品根据价格降序排列 |
| 7  | 商品列表综合排序      | 商品根据默认排序   | 商品根据默认排序   |

6.2.4 前台商品结算测试

前台商品结算测试主要分为未添加地址进行结算、添加地址选择在线付款、添加地址选择货到付款。前台商品结算测试如表 6.6 所示。

表 6.6 前台商品结算测试表

| 序号 | 测试用例       | 预测结果      | 实际结果      |
|----|------------|-----------|-----------|
| 1  | 未添加地址进行结算  | 提示请添加收货地址 | 提示请添加收货地址 |
| 2  | 添加地址选择在线付款 | 结算成功      | 结算成功      |
| 3  | 添加地址选择货到付款 | 结算成功      | 结算成功      |

6.2.5 前台购物车测试

前台购物车测试用例主要分为选择商品、取消选择商品、全部选择商品、取消全部选择商品、对商品数量进行增减、删除商品。前台购物车测试如表 6.7 所示。

表 6.7 前台购物车测试表

| 序号 | 测试用例   | 预测结果 | 实际结果 |
|----|--------|------|------|
| 1  | 选择商品   | 选择成功 | 选择成功 |
| 2  | 取消选择商品 | 取消成功 | 取消成功 |

| 序号 | 测试用例      | 预测结果 | 实际结果 |
|----|-----------|------|------|
| 3  | 全部选择商品    | 选择成功 | 选择成功 |
| 4  | 取消全部选择商品  | 取消成功 | 取消成功 |
| 5  | 对商品数量进行增减 | 增减成功 | 增减成功 |
| 6  | 删除商品      | 删除成功 | 删除成功 |

### 6.2.6 前台订单测试

前台订单测试用例主要分为查看全部订单、查看待发货订单、查看待收货订单、查看已收货订单、对待收货点击确认收货。前台订单测试如表 6.8 所示。

表 6.8 前台订单测试表

| 序号 | 测试用例       | 预测结果      | 实际结果      |
|----|------------|-----------|-----------|
| 1  | 查看全部订单     | 显示当前全部订单  | 显示当前全部订单  |
| 2  | 查看待发货订单    | 显示当前待发货订单 | 显示当前待发货订单 |
| 3  | 查看待收货订单    | 显示当前待收货订单 | 显示当前待收货订单 |
| 4  | 查看已收货订单    | 显示当前已收货订单 | 显示当前已收货订单 |
| 5  | 对待收货点击确认收货 | 确认收货成功    | 确认收货成功    |

### 6.2.7 前台地址测试

前台地址测试用例主要分为查看地址列表、新建收货地址、编辑收货地址、设置默认地址、设置默认地址、删除地址条目。前台地址测试如表 6.9 所示。

表 6.9 前台地址测试表

| 序号 | 测试用例   | 预测结果     | 实际结果     |
|----|--------|----------|----------|
| 1  | 查看地址列表 | 显示当前地址列表 | 显示当前地址列表 |
| 2  | 新建收货地址 | 新建收货地址成功 | 新建收货地址成功 |
| 3  | 编辑收货地址 | 编辑收货地址成功 | 编辑收货地址成功 |
| 4  | 设置默认地址 | 设置成功     | 设置成功     |
| 5  | 删除地址条目 | 删除成功     | 删除成功     |

6.2.8 后台登录测试

后台登录测试用例主要分为表单未输入完整、不足 6 位的用户名、超过 16 位的用户名、正确的用户名和密码、正确的用户名，错误的密码、错误的用户名，正确的密码、错误的用户名，错误的密码。后台登录测试如表 6.10 所示。

表 6.10 后台登录测试表

| 序号 | 测试用例         | 预测结果     | 实际结果     |
|----|--------------|----------|----------|
| 1  | 表单未输入完整      | 请完善表单内容  | 请完善表单内容  |
| 2  | 不足 6 位的用户名   | 请输入正确用户名 | 请输入正确用户名 |
| 3  | 超过 16 位的用户名  | 请输入正确用户名 | 请输入正确用户名 |
| 4  | 正确的用户名和密码    | 登录成功     | 登录成功     |
| 5  | 正确的用户名，错误的密码 | 用户名或密码错误 | 用户名或密码错误 |
| 6  | 错误的用户名，正确的密码 | 用户名或密码错误 | 用户名或密码错误 |
| 7  | 错误的用户名，错误的密码 | 用户名或密码错误 | 用户名或密码错误 |

6.2.9 后台商品管理测试

后台商品管理测试用例主要分为新建品牌、查看品牌列表、编辑品牌、删除品牌、新建商品、查看商品列表、编辑商品、删除商品、查看各品牌销量列表、查看各品牌库存列表。后台商品管理测试如表 6.11 所示。

表 6.11 后台商品管理测试表

| 序号 | 测试用例      | 预测结果      | 实际结果      |
|----|-----------|-----------|-----------|
| 1  | 新建品牌      | 新建品牌成功    | 新建品牌成功    |
| 2  | 查看品牌列表    | 显示当前品牌列表  | 显示当前品牌列表  |
| 3  | 编辑品牌      | 编辑品牌成功    | 编辑品牌成功    |
| 4  | 删除品牌      | 删除品牌成功    | 删除品牌成功    |
| 5  | 新建商品      | 新建商品成功    | 新建商品成功    |
| 6  | 查看商品列表    | 显示当前商品列表  | 显示当前商品列表  |
| 7  | 编辑商品      | 编辑商品成功    | 编辑商品成功    |
| 8  | 删除商品      | 删除商品成功    | 删除商品成功    |
| 9  | 查看各品牌销量列表 | 显示各品牌销量列表 | 显示个品牌销量列表 |
| 10 | 查看各品牌库存列表 | 显示各品牌库存列表 | 显示各品牌库存列表 |

6.2.10 后台运营管理测试

后台运营管理测试用例主要分为新建广告位、查看广告位列表、编辑广告位、删除广告位。后台运营管理测试如表 6.12 所示。

表 6.12 后台运营管理测试表

| 序号 | 测试用例    | 预测结果    | 实际结果    |
|----|---------|---------|---------|
| 1  | 新建广告位   | 新建广告位成功 | 新建广告位成功 |
| 2  | 查看广告位列表 | 显示广告位列表 | 显示广告为列表 |
| 3  | 编辑广告位   | 编辑广告位成功 | 编辑广告位成功 |
| 4  | 删除广告位   | 删除广告位成功 | 删除广告位成功 |

6.2.11 后台用户管理测试

后台用户管理测试用例就是查看用户列表。如表 6.13 所示。

表 6.13 后台用户管理测试表

| 序号 | 测试用例   | 预测结果   | 实际结果   |
|----|--------|--------|--------|
| 1  | 查看用户列表 | 显示用户列表 | 显示用户列表 |

6.2.12 后台订单管理测试

后台订单管理测试用例分为查看订单列表、未发货订单点击发货。如表 6.14 所示。

表 6.14 后台订单管理测试表

| 序号 | 测试用例      | 预测结果   | 实际结果   |
|----|-----------|--------|--------|
| 1  | 查看订单列表    | 显示订单列表 | 显示订单列表 |
| 2  | 未发货订单点击发货 | 发货成功   | 发货成功   |

6.2.13 后台系统管理测试

后台系统管理测试用例主要分为新建管理员、查看管理员列表、编辑管理员、删除管理员。后台系统管理员测试如表 6.15 所示。

表 6.15 后台系统管理测试表

| 序号 | 测试用例    | 预测结果    | 实际结果    |
|----|---------|---------|---------|
| 1  | 新建管理员   | 新建管理员成功 | 新建管理员成功 |
| 2  | 查看管理员列表 | 显示管理员列表 | 显示管理员列表 |



| 序号 | 测试用例  | 预测结果    | 实际结果    |
|----|-------|---------|---------|
| 3  | 编辑管理员 | 编辑管理员成功 | 编辑管理员成功 |
| 4  | 删除管理员 | 删除管理员成功 | 删除管理员成功 |

## 总结

刚开始定论文题目的时候，本着能够顺利毕业的想法，就写了一个手机商城。手机商城的各个模块完全是按照自己的臆想来做，去模拟一个正常的电商网站的流程去写，里面的页面布局以及交互方式大部分参考了京东的运营方式。其次自己的主攻方向是前端，由于减轻学习成本，选择了与 JavaScript 有着强关联的后端与数据库，所以后端采用的是 NodeJS，NodeJS 用的都是 JavaScript 语言进行编写，数据库采用的是 MongoDB，MongoDB 数据库的表结构与 JavaScript 的对象结构基本上类似，所以如此的选择，上手相对于以 Java 为后端要容易一些，而且基本上在 VSCode 编译软件就可以完成整个系统的编写与运行。

本系统最大的难点在于安全策略，也就是 axios 的响应拦截，从后端传过来的 token 进行校验，如果是无效 token 或者没有 token 的时候，就会强制跳转路由。其次就是其他人的商城的商品属性的选择几乎是没选择的选项，即使有也就是一个层级的属性选择，如一辆自行车的颜色，都是单层次的；而本人写的手机上商城手机，手机有机身存储与颜色的选择，首先要确定机身存储与机身颜色的关系，为了前端接收数据容易渲染到页面上，本人给商品添加机身存储的数组，然后机身存储数组里再嵌套机身颜色数组，最后把详细的价格挂载到机身存储里，用存储来标识手机的价格，这样一来机身存储与机身颜色属于父子级的关系，容易进行 for 循环的遍历渲染。

本系统最不尽人意的地方，就是支付接口没有实现，本人去微信官方与支付宝官方查看了支付接口，发现自己没有能力去完成支付接口，其次就是支付需要商业户才可以而且要收取一定的手续费。

此系统花了将近 40 多天才完成，总的来说，遇到了很多棘手的问题，如 VueJS 上的一些异步回调的问题，以及获取 DOM 节点上的数据，发现获取不到，然后通过异步延迟得到。最后是 elementUI 自己封装的语法，很多新的语法，官网没有直接更新，需要手动去 CSND 上去查。编写此系统的过程中，对 JavaScript 语言比以前要熟练得多，尤其是 es6 的大部分语法，如箭头函数、结构赋值、一些数组的回调方法、async 与 await 的异步请求当同步请求的使用<sup>[1]</sup>。采坑的地方数不胜数，幸运的是自己基本上一点点的花时间解决了，为自己以后的工作打下了基础。技术更新太快，唯有不断学习。

## 参考文献

- [1] 徐晨,袁晓君,张曼丽. 社交网络时代跨境电商口碑营销模式研究[J]. 《合作经济与科技》,2020,000(001):83-85
- [2] Liang L , Zhu L , Shang W , et al. Express supervision system based on NodeJS and MongoDB[C]// IEEE/ACIS International Conference on Computer & Information Science. IEEE, 2017.
- [3] 李胜. 基于 JavaScript 的面向对象语言特性及其程序设计[J]. 科技与创新, 2019(6):112-113.
- [4] 霍多罗夫. MongoDB 权威指南[M]. 2014.
- [5] 张惠芬. JavaScript 的理论分析及其应用[J]. 衡水学院学报(1):51-52
- [6] Chodorow K, Dirolf M. MongoDB: The Definitive Guide[M]. 2010.
- [7] 扎卡斯. JavaScript 高级程序设计[M]. 2006.
- [8] Antani,Ved 《Mastering JavaScript》 [M]. Birmingham. UK : Packt Publishing, 2016.
- [9] Simon Holm Jensen, Anders Møller, Peter Thiemann. Type Analysis for JavaScript[C]// Static Analysis, 16th International Symposium, SAS 2009, Los Angeles, CA, USA, August 9-11, 2009. Proceedings. Springer-Verlag, 2009.
- [10] Pienaar J A , Hundt R . JSWhiz: Static analysis for JavaScript memory leaks[C]// Code Generation and Optimization (CGO), 2013 IEEE/ACM International Symposium on. ACM, 2013.

## 致谢

本系统对于其他人做的系统来说，甚是简陋，但是对于自己来说，是一次了不起的尝试。在这里，我要非常感谢黑马程序员以及尚硅谷，通过他们的视频，我学会不少前端知识，还要感谢 B 站全栈之巅 UP 主，通过他的视频，会使用 NodeJS 后端的技术以及 MongoDB 数据库的使用，最后我要衷心的感谢我的指导老师许小龙老师给我这次锻炼的机会，并花费了不少时间指导我对论文进行修改。