

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

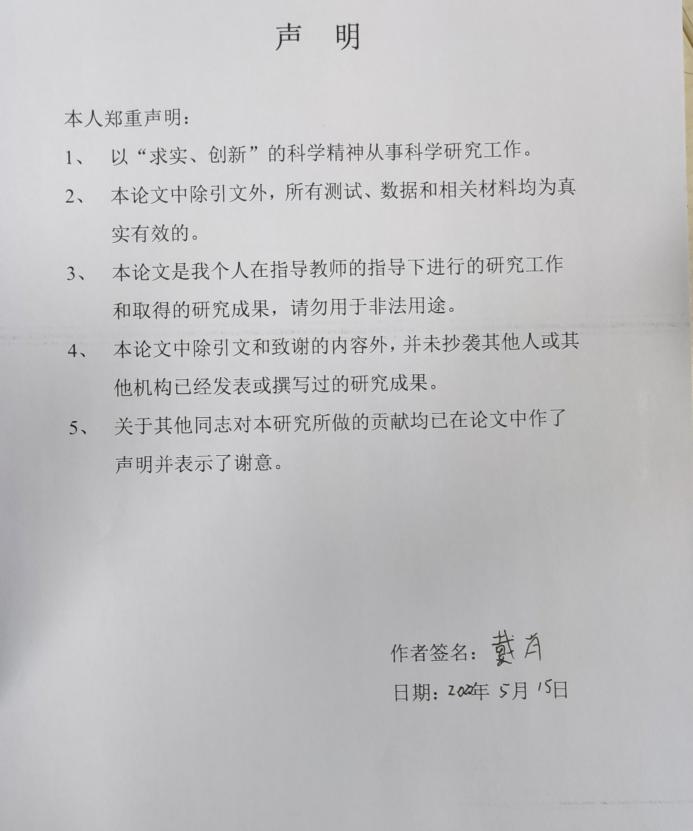
学 号 201633070041

院 系应用技术学院

专 业软件工程

指导教师许小龙

**二Ｏ二Ｏ 年 五月 十五 日**



目 录

[1 绪论 1](#_Toc39088715)

[2 系统概论与工具选择 1](#_Toc39088716)

[2.1 B/S架构 1](#_Toc39088717)

[2.2 JavaScript语言 2](#_Toc39088718)

[2.3 NodeJS 2](#_Toc39088719)

[2.4 MongoDB数据库 2](#_Toc39088720)

[2.5 VueJS 2](#_Toc39088721)

[2.6 Visual Studio Code工具 2](#_Toc39088722)

[3 需求分析 3](#_Toc39088723)

[3.1 前台系统需求分析 3](#_Toc39088724)

[3.2 后台系统需求分析 3](#_Toc39088725)

[3.3 可行性分析 4](#_Toc39088726)

[4 系统概要设计 4](#_Toc39088727)

[4.1 前台系统设计 4](#_Toc39088728)

[4.1.1 注册/登录功能模块 5](#_Toc39088729)

[4.1.2 首页功能模块 5](#_Toc39088730)

[4.1.3 商品详情功能模块 6](#_Toc39088731)

[4.1.4 购物车管理功能模块 7](#_Toc39088732)

[4.1.5 商品结算功能模块 7](#_Toc39088733)

[4.1.6 订单管理功能模块 8](#_Toc39088734)

[4.1.7 地址管理功能模块 9](#_Toc39088735)

[4.2 后台系统设计 9](#_Toc39088736)

[4.2.1 登录功能模块 9](#_Toc39088737)

[4.2.2 商品管理功能模块 10](#_Toc39088738)

[4.2.3 运营管理功能模块 10](#_Toc39088739)

[4.2.4 用户管理功能模块 11](#_Toc39088740)

[4.2.5 订单管理功能模块 11](#_Toc39088741)

[4.2.6 系统管理功能模块 12](#_Toc39088742)

[4.3 数据库设计 12](#_Toc39088743)

[4.3.1 概念结构设计 12](#_Toc39088744)

[4.3.2 逻辑结构设计 16](#_Toc39088745)

[5 系统详细设计 20](#_Toc39088746)

[5.1 前台系统详细设计 21](#_Toc39088747)

[5.1.1 注册/登录功能模块 21](#_Toc39088748)

[5.1.2 首页功能模块 22](#_Toc39088749)

[5.1.3 商品详情功能模块 23](#_Toc39088750)

[5.1.4 购物车管理功能模块 24](#_Toc39088751)

[5.1.5 商品结算功能模块 25](#_Toc39088752)

[5.1.6 订单管理功能模块 26](#_Toc39088753)

[5.1.7 地址管理功能模块 27](#_Toc39088754)

[5.2 后台系统详细设计 28](#_Toc39088755)

[5.2.1 商品管理功能模块 29](#_Toc39088756)

[5.2.2 运营管理功能模块 32](#_Toc39088757)

[5.2.3 用户管理功能模块 32](#_Toc39088758)

[5.2.4 订单管理功能模块 33](#_Toc39088759)

[5.2.5 系统管理功能模块 33](#_Toc39088760)

[6 系统测试 34](#_Toc39088761)

[6.1 测试用表 34](#_Toc39088762)

[6.1.1 测试计划进度表 34](#_Toc39088763)

[6.1.2 模块测试用例表 35](#_Toc39088764)

[6.2 黑盒测试 36](#_Toc39088765)

[6.2.1 前台注册测试 36](#_Toc39088766)

[6.2.2 前台登录测试 36](#_Toc39088767)

[6.2.3 前台商品搜索测试 37](#_Toc39088768)

[6.2.4 前台商品结算测试 37](#_Toc39088769)

[6.2.5 前台购物车测试 37](#_Toc39088770)

[6.2.6 前台订单测试 38](#_Toc39088771)

[6.2.7 前台地址测试 38](#_Toc39088772)

[6.2.8 后台登录测试 39](#_Toc39088773)

[6.2.9 后台商品管理测试 39](#_Toc39088774)

[6.2.10 后台运营管理测试 40](#_Toc39088775)

[6.2.11 后台用户管理测试 40](#_Toc39088776)

[6.2.12 后台订单管理测试 40](#_Toc39088777)

[6.2.13 后台系统管理测试 40](#_Toc39088778)

[总结 41](#_Toc39088779)

[参考文献 42](#_Toc39088780)

[致谢 43](#_Toc39088781)

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却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 ZOL and various mobile phone malls, I began to try to combine the mobile phone comparison parameters in ZOL 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。上诉几个方面中，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[1]。

1. 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[2]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[3]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。

* 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，都是使用此架构，用任意浏览器打开即可，与其他架构比较而言，用不着安装多余的的客户端，让用户使用起来更加的轻量便捷。

* 1. JavaScript语言

JavaScript目前是用的最多的脚本语言，起源于上世纪90年代中期.，刚开始命名为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，在当时有许多脚本语言，都没有打过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更趋向于编程语言而不单单是脚本语言。

* 1. NodeJS

NodeJS它不像Java、C，它根本不是一种语言，是一个JavaScript的运行环境或者是平台，可以称作为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，运行在谷歌浏览器的V8引擎上，运行效率特别高。当NodeJS用JavaScript进行编程时，是没有DOM这种概念的，只有EcmaScript基本的JavaScript语法部分。

* 1. MongoDB数据库

MongoDB[4]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。

* 1. 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框架，诞生虽晚，但与老牌Angular、React齐肩，造成三足鼎立的局面，现在已经出现了Vue3.0开发版本，效率更加高效。

* 1. 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。

1. 需求分析
   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、品牌的分类，用户或者游客点击轮播图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，用户或游客可以排序商品列表。

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。

* 1. 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，当点击发货会变为待收货。

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：管理员可以对管理员进行增删改。

* 1. 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。

1. 系统概要设计
   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。

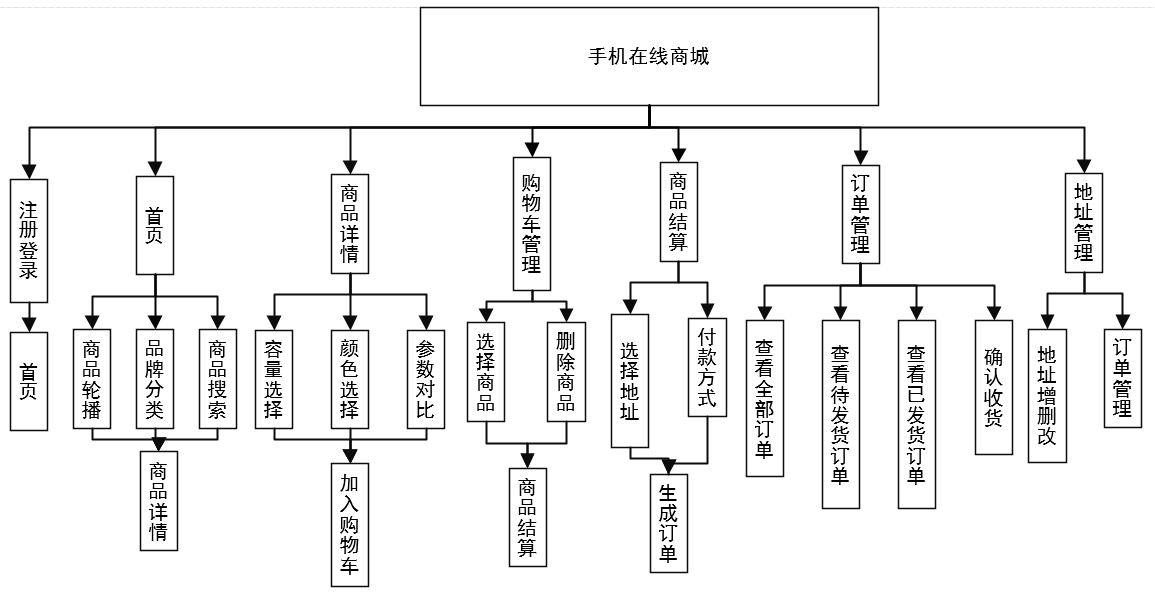


图 4.1 前台系统层次结构

* + 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，要进行数据检查，若通过，前端把账户密码发送给后端，后端去数据库中寻数据是否相符，如果相符则告诉前端链接到其他检查页，若不相符则有相关提示；当用户进行注册的时候，首先要进行表单，如果检查不成功，会进行相关的提示，如果检查通过，前端会把用户数据发送给后端，进行二次检查，若用户名已存在，则会在页面上会有用户名已存在的提示；若全部检查通过，则写入数据库，并跳转登录页，让用户去登录。注册/登录功能模块如图4.2所示。

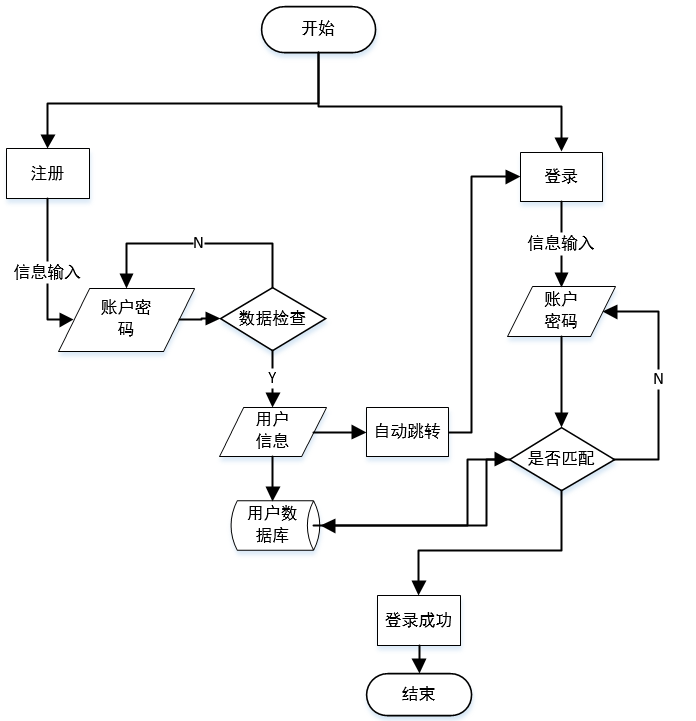


图4.2 注册/登录功能模块流程图

* + 1. 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。

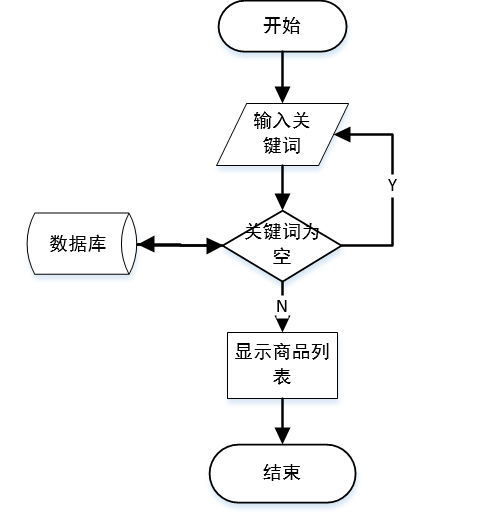


图.4.3 商品搜索的流程图

* + 1. 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。商品详情如图4.4所示。

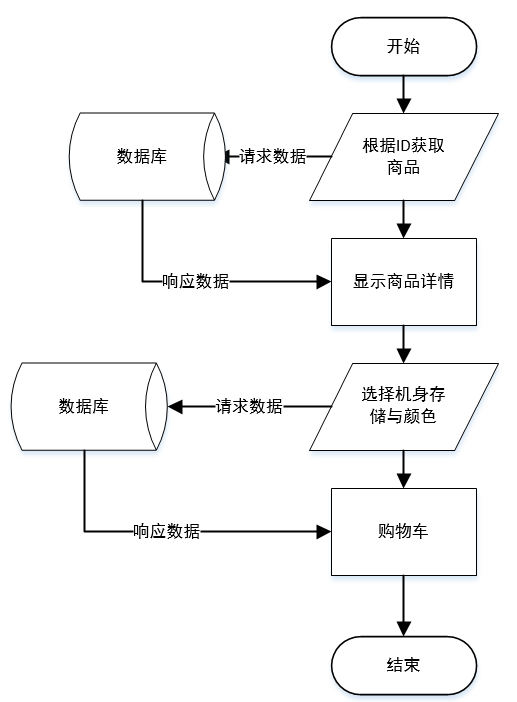


图4.4 商品详情功能模块流程图

* + 1. 购物车管理功能模块

当用户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。

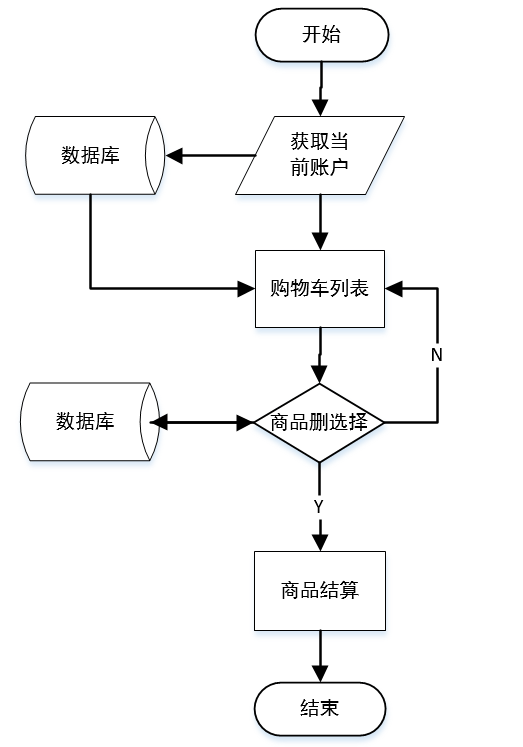


图4.5 购物车管理功能模的块流程图

* + 1. 商品结算功能模块

当用户到商品结算页时，会看到结算清单、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。如果没有地址，可以在此页面新建，也可以设置某一地址项为默认地址，其实这里直接内嵌了地址管理功能。其次就是，用户则可以两种有付款方式进行选择，第一种是货到付款的方式，第二种是在线支付的方式，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。

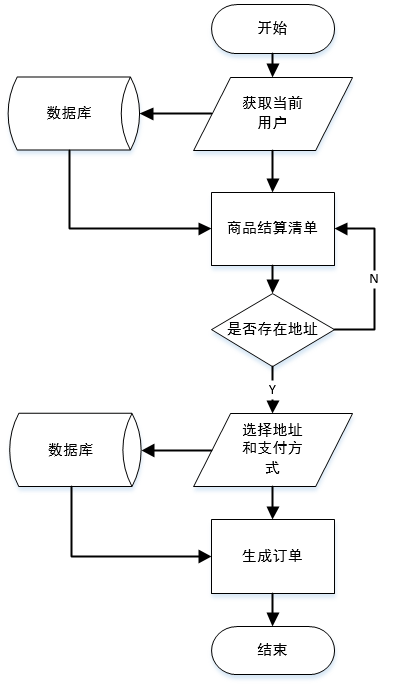


图4.6 商品结算功能模块的流程图

* + 1. 订单管理功能模块

订单管理功能模块分为全部订单，通过筛选可以分为三种模块。当为待收货时，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。

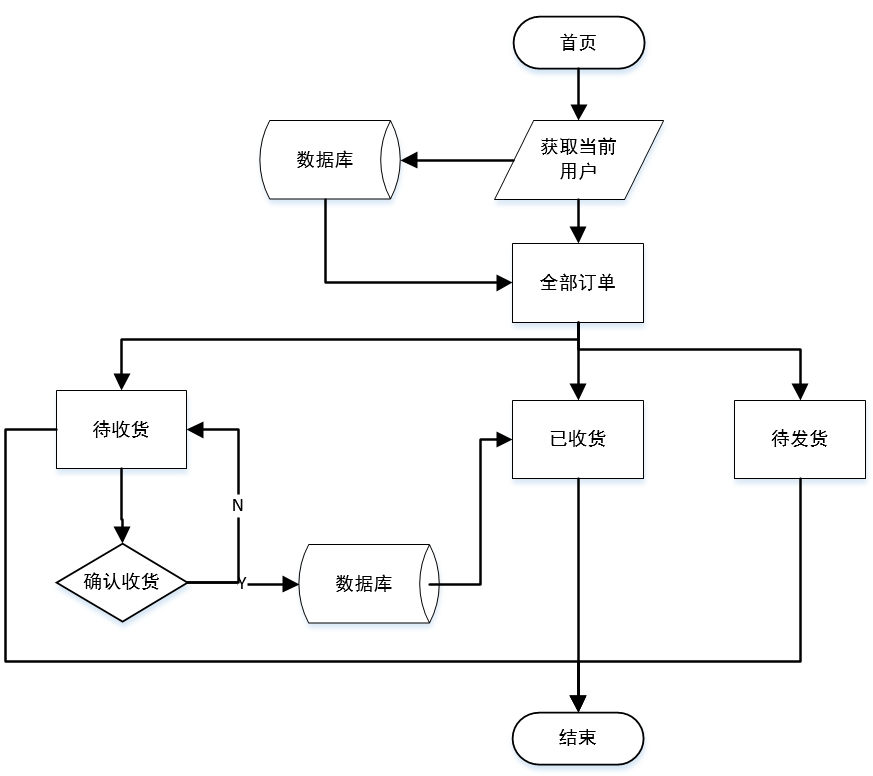


图4.7 订单管理功能模块的流程图

* + 1. 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增删改，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。

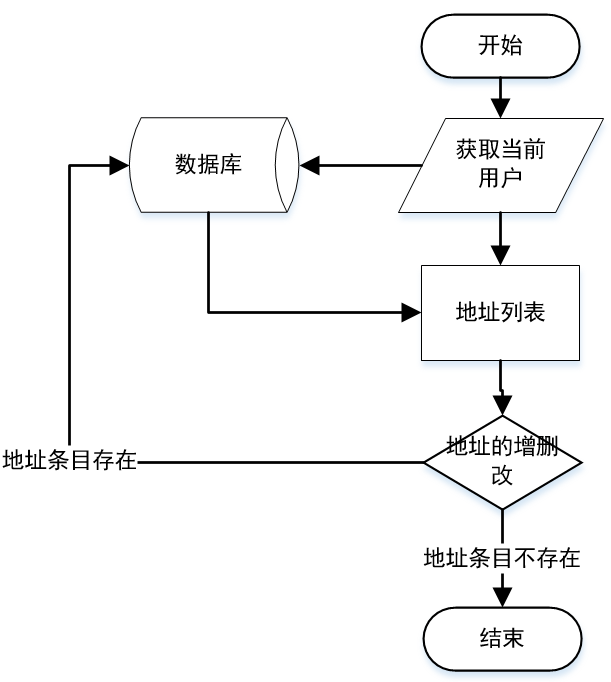


图4.8 地址管理功能模块流程图

* 1. 后台系统设计
     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，前端组件进行对表单信息进行相应的检查，如果检查成功，则会把账户密码发送给后端，后端则会把信息去数据库进行相应的查找并检查，若存在则给前端反馈登录成功，如果不存在，会有相应提示，并一直停留在此页面。登录功能模块如图4.9所示。

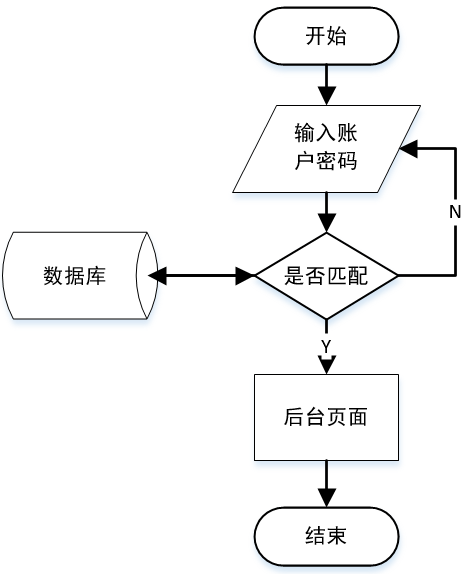


图4.9 登录功能模块的流程图

* + 1. 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，这里的销量和库存都是用百度官方的echarts 图表进行渲染的。当管理员输入品牌名，新建品牌成功后，页面会链接到品牌列表，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；对商品的操作与对品牌操作一样，但是新建商品包括基本信息、参数配置、详细配置三大模块。商品管理功能模块如图4.10所示。

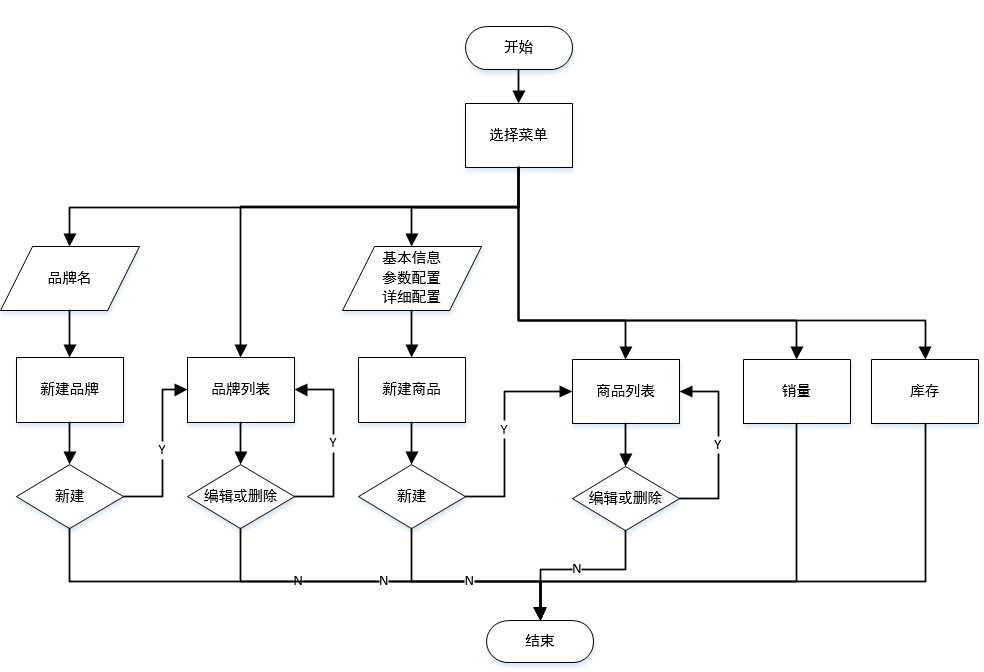


图4.10 商品管理功能模块

* + 1. 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，也就是对前台轮播图详细信息的填写，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，并且可以清晰地查看每个广告位条目，包括封面图也会加载出来，同样的也可以对其进行编辑或者删除。运营管理功能模块如图4.11所示。

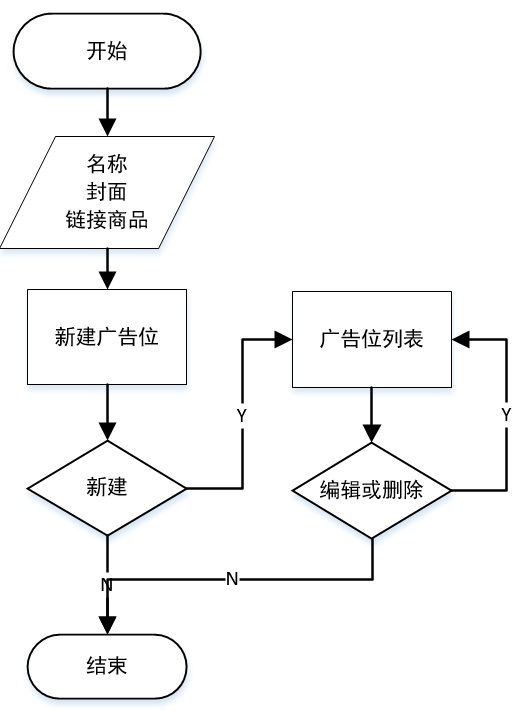


图4.11 运营管理功能模块的流程图

* + 1. 用户管理功能模块

此模块只能查看ID和用户名，管理员不能对用户的信息进行任何的详细操作。

* + 1. 订单管理功能模块

管理员对订单列表可以订单详情，与前台对应，把订单列表过滤划分，分为三个子项。管理员点击发货按钮，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。

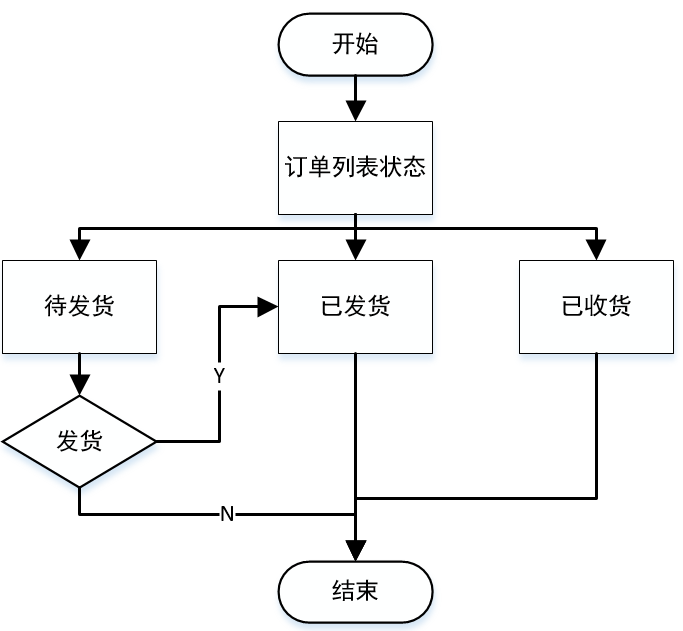


图4.12 订单管理功能模块流程图

* + 1. 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表单数据保存写入成功以后就链接到与之相对应的列表，也可以对此列表的每个条目进行编辑或者删除操作以及查看详情功能。系统管理功能模块如下图4.13所示。

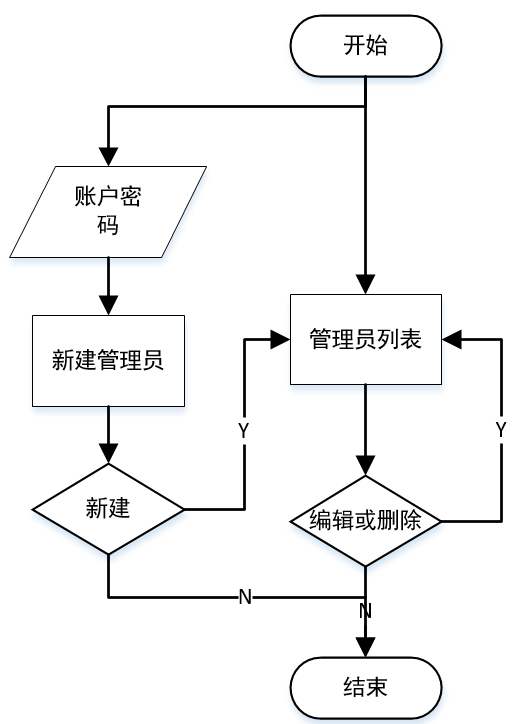


图 4.13 系统管理功能模块的流程图

* 1. 数据库设计
     1. 概念结构设计
        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。

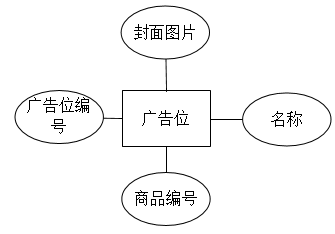


图 4.14 首页广告位实体图

（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。

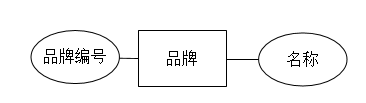


图4.15 品牌实体图

（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。

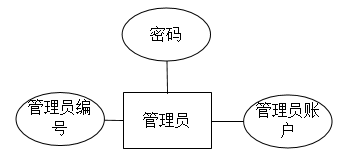


图4.16 管理员实体图

（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，地址编号等为外键。用户实体如图4.17所示。

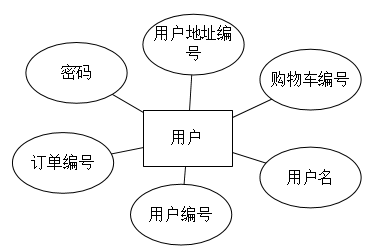


图4.17 用户实体图

（5）用户地址实体

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

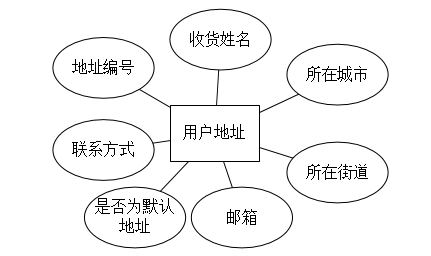


图4.18 地址实体图

（6）购物车实体

包括商品名称、商品封面、商品价格、商品副标题、购物车编号、商品编号、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、机身颜色等为外键。购物车实体如图4.19所示。

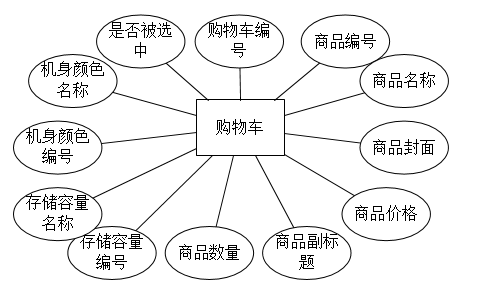


图4.19 购物车实体图

（7）订单实体

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

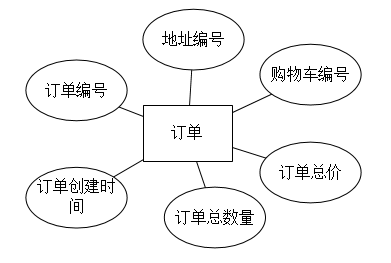


图4.20 订单实体图

（8）商品实体

包括品牌编号、商品编号、商品名 、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。

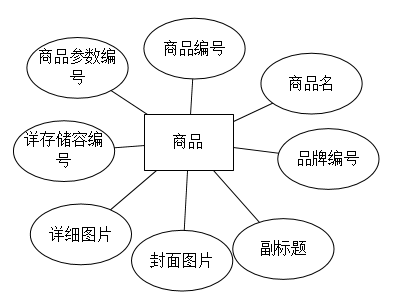


图4.21商品实体图

（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。

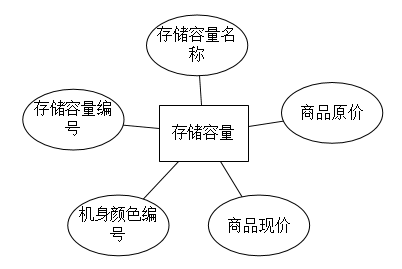


图4.22 存储容量实体图

（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。

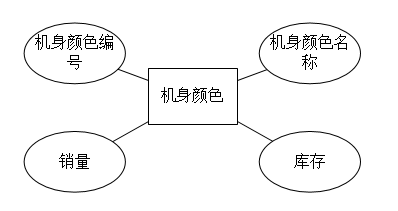


图4.23 机身颜色实体图

（11）商品参数实体

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

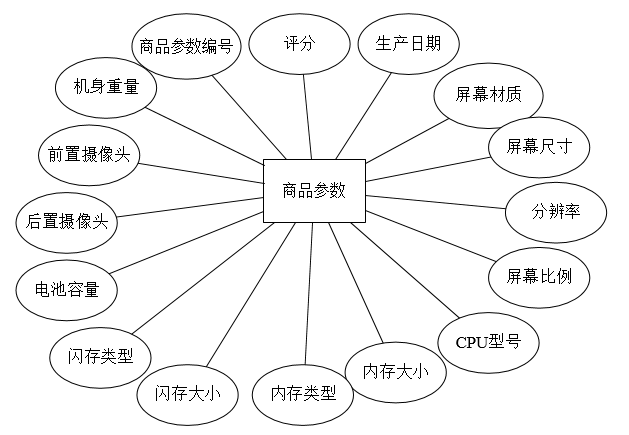


图4.24 商品参数实体图

* + - 1. 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。

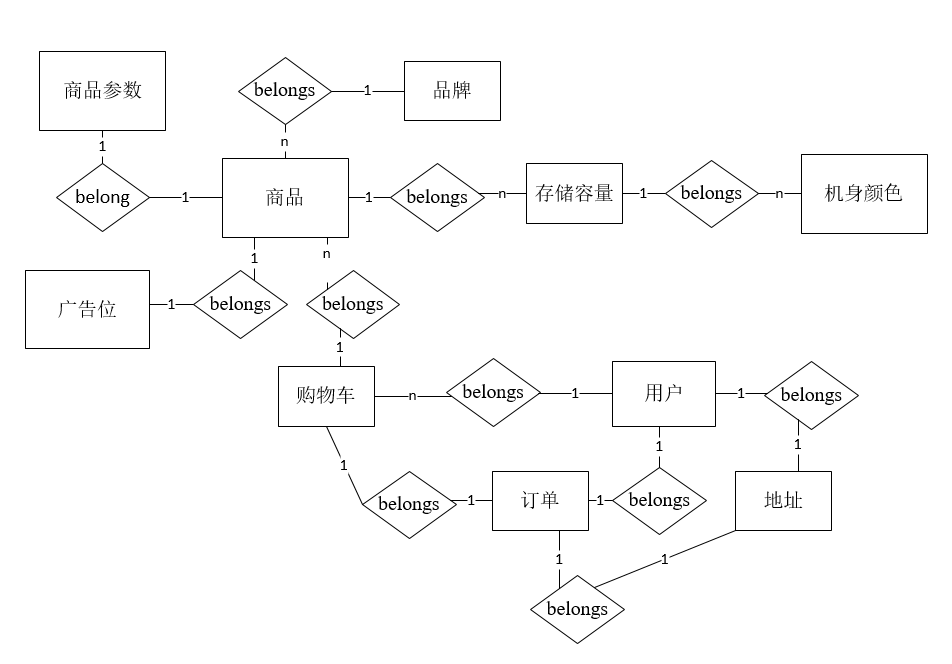


图4.25 E-R图

* + 1. 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 |
| 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 |
| productCoverIamge | 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，1是已发货/待收货，2是收货成功） | 否 |
| 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 |
| createTine | 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 |
| coverIamge | 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 |
| socre | 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 |
| rosolution | 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 |
| 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 |

1. 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，后端再把用户的信息进行加密，当用户登录成功时，后端会把token发送给前端，前端通过cookie、localStorage甚至是sessionStorage进行token数据的存储；每次前端发起请求的时候，就会把token挂载到请求头上，然后把token传送到后端，后端有相关拦截，进行校验用户是否一致，如果一致才会让用户可以去跳转到哪些路由。前端也要进行二次校验，对每个路由进行相关的标记，通过设置axios响应拦截，拦截到token是否存在或者失效，如果不存在或者已经失效，则会有进行强制路由跳转。其实这里的操作可以后端就可以完成强制路由跳转即可，但是为了安全起见，则前后端都进行相关配置。

* 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。

* + 1. 注册/登录功能模块
       1. 注册功能

注册页面会有用户名、密码、确认密码三个input标签；当输入用户名时，前端就会实时去把数据发送给后端，后端会进行相关查找，如果发现用户名字段已经存在了，则会提示用户名存在；若三个input标签都不输入，则会提示用户完善表单内容；若已完善并且数据写入成功，则会注册成功并链接到其他页。注册功能如图5.1所示。

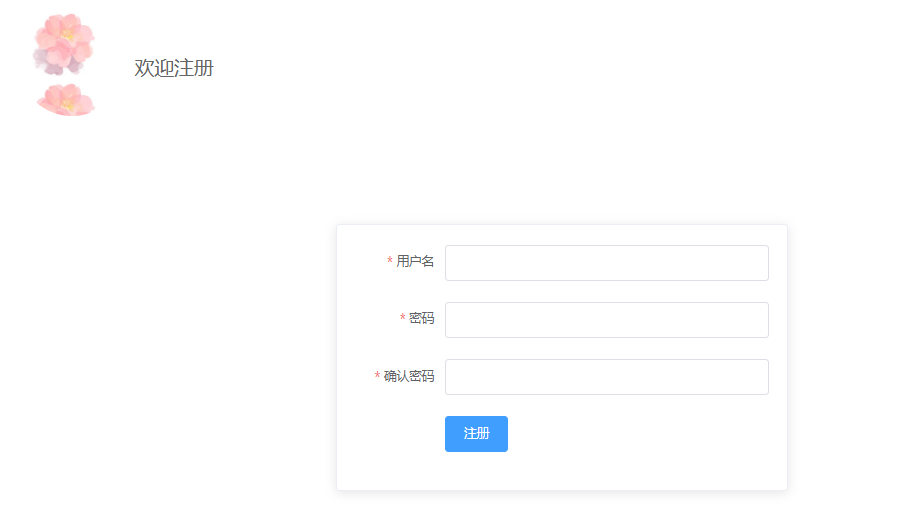


图5.1 注册功能

* + - 1. 登录功能

当用户若未做任何动作，直接登录话，会有表单检查，并弹出相关提示让用户完善，若两个input表单项都输入完整，点击登录，则会提交表单到后端，如果数据库检查成功的话，会链接相关页。登录功能如图5.2所示。

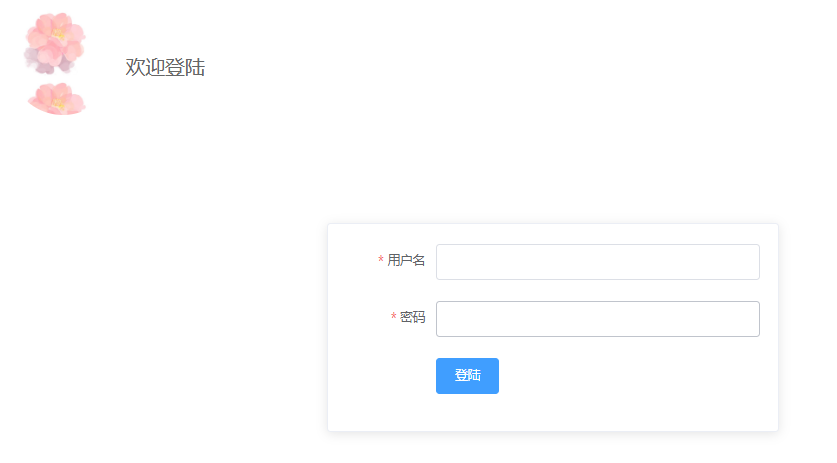


图5.2 登录功能

* + 1. 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

* + - 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，综合排序以及销量和价格，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

* + 1. 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、生产日期等等。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。



图5.6 商品详情

* + 1. 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数组写入数据对象，本人在这里写了简版的商品列表，也就是一个快捷入口，几乎每个页面都会显示，用户可以在页面的右侧看到一个侧边列表，点击可以查询到用户的已添加的商品，可以在侧边栏进行相关删除；点击侧边栏的结算按钮，就会链接到完整的购物车管理界面，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

* + 1. 商品结算功能模块

当用户到达商品结算页时，页面有三个显示模块，分别是支付方式的选择、收货地址的选择和送货清单的详细信息。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，完完全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。

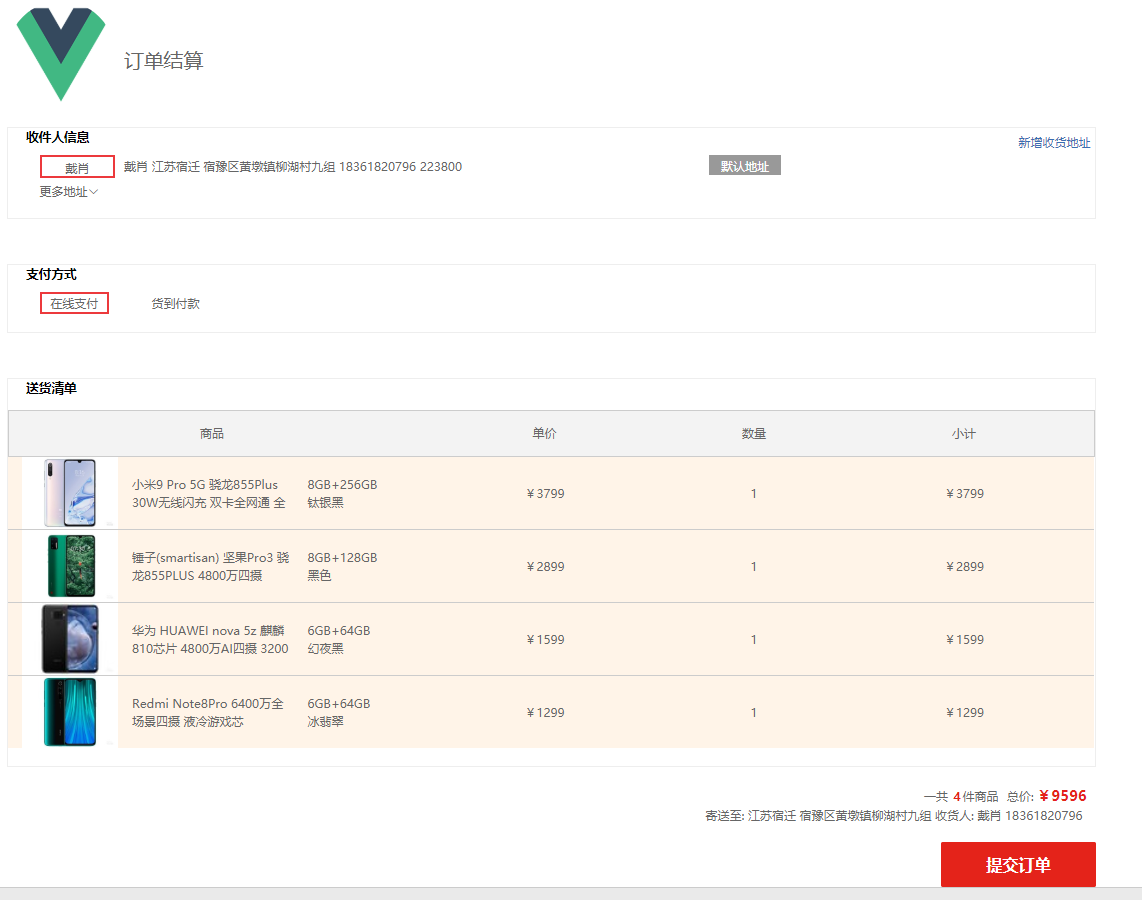


图5.9 商品结算

* + 1. 订单管理功能模块

当到订单管理页面时，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[7]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。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。



图5.10 订单管理

* + 1. 地址管理功能模块

当在地址管理页面时，用户可以查看列表，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。



图5.11 新建收货地址



图5.12 编辑收货地址

.



图5.13 地址列表

* 1. 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。

* + 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。

* + - 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。



图5.14 品牌模块

* + - 1. 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，当确定填写对应的信息后，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、生产日期等等。商品模块如图5.15和图5.16所示。



图5.15 新建商品



图5.16 商品列表

* + - 1. 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，把数据通过图表的方式渲染到页面上。销量模块如图5.17所示。

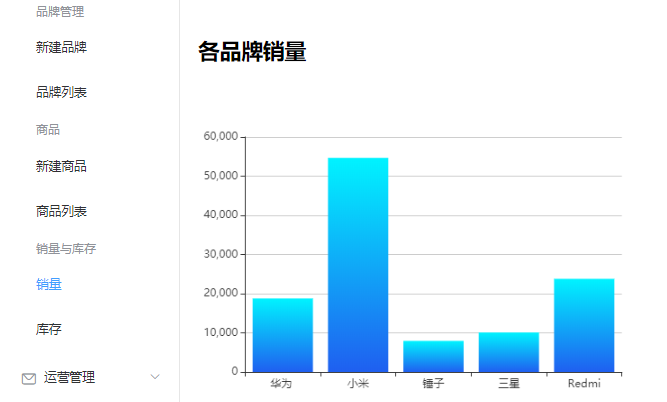


图5.17 各品牌销量

* + - 1. 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。

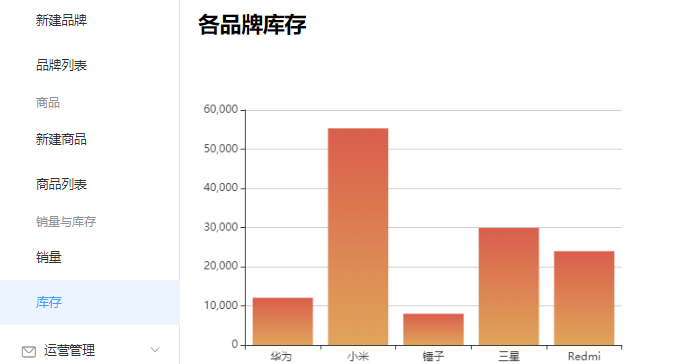


图5.18 各品牌库存

* + 1. 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。



图5.19 运营管理

* + 1. 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息进行任何的修改等操作。用户管理模块如图5.20所示。

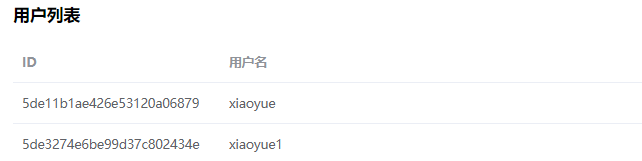


图5.20 用户列表

* + 1. 订单管理功能模块

管理员可以查看订单列表详情。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。

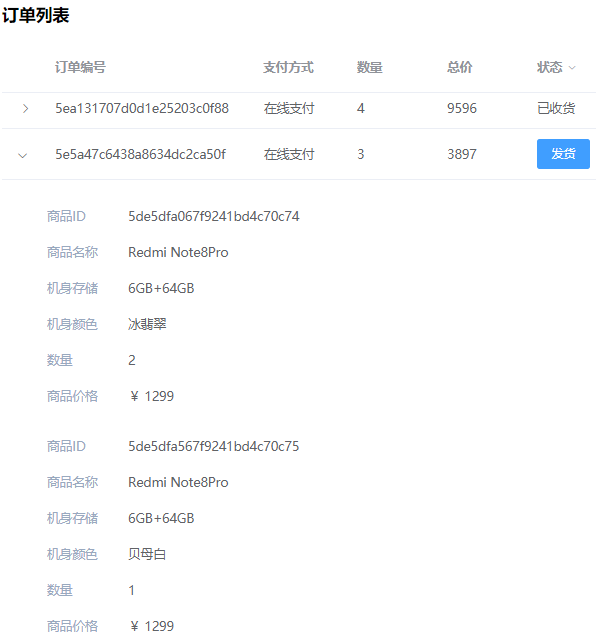


图5.21 订单列表

* + 1. 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。管理员在管理员列表进行管理员详情的查看，可以查看管理员的id与管理员名称，管理员可以在此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。

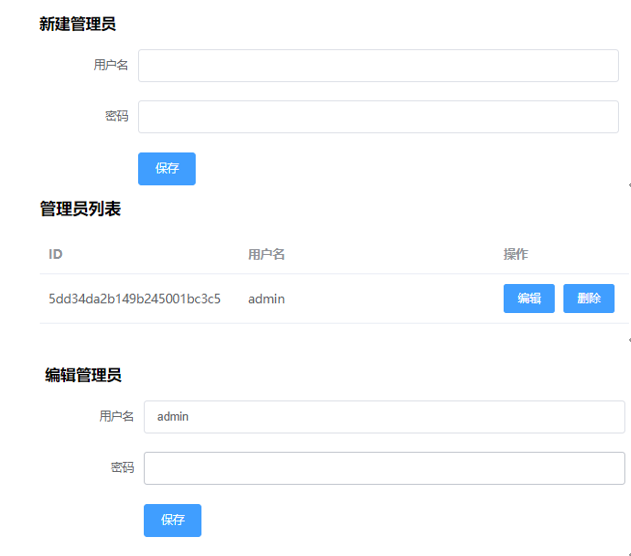


图5.22 管理员管理

1. 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。

* 1. 测试用表
     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 |

* + 1. 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 |

* 1. 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。

* + 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 | 表单未完整 | 请完善表单内容 | 请完善表单内容 |
| 3 | 用户名6位以下 | 请输入正确用户名 | 请输入正确用户名 |
| 4 | 用户名超过16位 | 请输入正确用户名 | 请输入正确用户名 |
| 5 | 两次密码冲突 | 两次密码冲突 | 两次密码冲突 |
| 6 | 正确格式的用户名和密码 | 注册成功 | 注册成功 |

* + 1. 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 | 对的用户名，错的密码 | 用户名或密码错误 | 用户名或密码错误 |
| 6 | 错的用户名，对的密码 | 用户名或密码错误 | 用户名或密码错误 |
| 7 | 错的用户名，错的密码 | 用户名或密码错误 | 用户名或密码错误 |

* + 1. 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 |

* + 1. 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 |

* + 1. 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、取消全部选择商品等。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 |
| 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 |

* + 1. 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 |

* + 1. 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 |

* + 1. 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 | 用户名密码都符合 | 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 |

* + 1. 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 |

* + 1. 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 |

* + 1. 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 |

* + 1. 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 |

* + 1. 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 |
| 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有选择的选项，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[1]。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，为自已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。