|  |
| --- |
| **本 科 毕 业 设 计（论文）**  **题 目：**海大二手商品c2c电商网站开发与实现  **姓 名：罗宏鑫**  **学 号：201611921119**  **班 级：计数1165**  **学 院：数学与计算机学院**  **专 业：信息与计算科学** |

**本 科 毕 业 设 计（论文）任 务 书**

**海大二手商品c2c电商网站开发与实现**

**摘 要**

如今已经开始迈入二十一世纪20年代，我国的电子商务早在上个世界就已经开始发展，尤其是在10年代左右，由淘宝、天猫和京东等为主力军的电子商务平台将我国完全的带入了电子商务时代，随之蓬勃而出的各种电商交易形式也随之出现，例如：企业与企业（BTB）、企业与用户（BTC）以及用户与用户（CTC）之间通过专用的网络或Internet来进行数据信息交换、传递或开展交易活动的一种新型商业模式。在我国，电子商务平台上的交易极大的带动国民的消费水平，拉高我国GDP增长。在这几种交易形式中，我将主要围绕CTC即用户与用户之间的交易形式去讲解我是海大二手交易平台的开发过程和一些细节。

在大学里，每年都会有很多毕业生要毕业，离开学校，但是面对住在大学四年的时间的生活，我们很多学生无可避免的会有了很多闲置的东西，例如各种闲置的书籍、电子产品或生活用品等。这些东西里，很多即将毕业的同学可能因为不方便会选择拒绝将其带走，在这样的情况下，多数同学可能都是拿去送给自己认识的师弟师妹们，甚至是拿去丢掉，送人还好，但丢弃闲置用品这样会无形中也造成了很大的浪费。而且不仅是针对于即将毕业的毕业生，很多的大学生在大学生活中因为网购或是离开父母拥有难得的经济独立情况下，也是购买和存有了很多闲置的东西。平时，我看见不少同学都是在微信群或是朋友圈里将这些闲置的东西进行贩卖和销售，但是这样无法将其商品推荐给认识以外的人这样就导致面向客户的群体比较小，而且在价格方面，可能是朋友或熟人的关系又很难有个妥协的价格。如果能专门为海大的学子提供一个免费的交易平台来处理各种闲置物品，让学生们能够在网上二手交易平台免费进行交易，让学生们能以低廉的价格买到优质的商品和解决闲置物品烦恼的问题，杜绝丢弃浪费行为和让同学与同学之间有更多相识接触的机会，因此开发一个专门为海大学生服务的二手交易平台我认为还是很有必要的。

整个电商平台的前端开发是采用基于最新的渐进式MVVM开发模式的前端三大框架之一的vue框架去开发的，vue是以数据驱动和组件化的思想构建的，采用自底向上增量开发的设计。后端方面则是采用Node.js去开发，因为其使用了事件驱动、非阻塞式I/O的模型，在后端语言中具有轻量且高效的特点。数据库方面则是采用MongoDB，这是一个基于分布式文件存储的数据库，用C++语言编写，是为WEB应用提供可扩展的高性能数据库存储解决方案，是一个介于关系数据库和非关系数据库之间的产品。

关键词：二手电商平台；JavaScript；MongoDB；Vue全家桶 ；HTML；Node.js **ABSTRACT**

Now has begun to enter the twenty-first century 20 s, the electronic commerce in our country has started as early as in the development, especially in the 10 s, by taobao, jingdong and Tmall as the mainstay of e-commerce platform will be fully into the era of electronic commerce in our country, as out of a variety of electronic business transaction forms have emerged, such as: enterprise and enterprise (BTB), with the user (BTC) and the user and the user (CTC) through a dedicated network or the Internet for information exchange, transfer, or to carry out trading activities of a new business model.In China, the transaction on the e-commerce platform greatly drives the consumption level of the citizens and boosts the GDP growth.Among these transaction forms, I will mainly explain the development process and some details of my haida second-hand trading platform around the transaction form of CTC between users.

In the university, every year there will be a lot of graduates to graduate and leave the school, but in the face of living in the university for four years of life, many of us students will inevitably have a lot of idle things, such as a variety of idle books, electronic products or daily necessities.In these things, a lot of students who are about to graduate may choose to refuse to take it away because of the inconvenience, in this case, most of the students may be to take to give their understanding of the younger brothers and sisters, or even take to throw away, give people good, but discard the idle things will also cause a great waste.And it's not just for those who are about to graduate. Many college students have bought and stored a lot of idle things in their college life because they bought online or left their parents and had rare financial independence.At ordinary times, I saw a lot of students are in the WeChat group or circle of friends, will these unused things to sell and sales, but it can't recommend their products to meet people outside thus become customer oriented group is small, and in terms of price, may be friends or acquaintances relationship would have a compromise price again.If we can specifically for students to provide a free head trading platform to deal with all kinds of idle items, let the students can free trading second-hand trading platform on the Internet, can let the students with low price to buy high quality goods and solve the problem of idle goods trouble, put an end to discarding the waste and make more get to know each other contact between students and students, thus developing a specifically for second-hand trading platform of student services of head I think or very be necessary.

The front-end development of the whole e-commerce platform is developed by the vue framework, one of the three front-end frameworks based on the latest progressive MVVM development mode. Vue is built with the idea of data-driven and componentization, and adopts the design of bottom-up incremental development.The back-end side is developed with node. js, because it USES an event-driven, non-blocking I/O model and is lightweight and efficient in the back-end language.In terms of database, MongoDB is adopted, which is a database based on distributed file storage and written in C++ language. It is a scalable high-performance database storage solution for WEB applications and a product between relational database and non-relational database.

**Key words:** Second-hand e-commerce platform；JavaScript；MongoDB；Vue buckets

；HTML ；Node.js

目 录

第一章 绪论

1.1 课题研究背景

1.2 课题研究意义

1.3 国内外现状

1.4 论文的研究内容

1.5 论文的组织结构

第二章 相关技术介绍

2.1 HTML5、CSS和JavaScript

2.2 Vue框架

2.3 Node.js

2.4 本章总结

第三章 系统分析

3.1 功能性需求

3.1.1 用户功能

3.1.2 管理员功能

3.2 可行性分析

3.3 本章总结

第四章 系统设计

4.1 概要设计

4.1.1 体系结构设计

4.1.2 系统模块设计

4.2 系统详细设计

4.2.1数据库设计

第五章 系统实现

5.1 后端接口实现

5.1.1 用户信息模块的功能接口实现

5.1.2 商品信息模块的功能接口实现

5.1.3 轮播图模块的功能接口实现

5.1.4 管理员模块的功能接口实现

5.2 前端页面功能实现

5.2.1 首页

5.2.2 商品分类页面

5.2.3 收藏页面

5.2.4 个人中心页面

5.2.5 用户登录页面

5.2.6 用户注册页面

5.2.7 个人信息修改页面

5.2.8 个人商品管理页面

5.2.9个人商品发布和修改页面

5.2.10 商品详情页面

第六章 系统测试

6.1 测试内容

6.2 测试用例

6.3 测试结果感悟

第七章 总结

参考文献

# 绪论

如今计算机及网络技术的发展已经有很长的一段时间了，电脑已经走入千家万户，不再是稀有的商品，尤其是在中国，在中国互联网络信息中心（CNNIC）在京发布第44次《中国互联网络发展状况统计报告》，据《统计报告》统计显示，截至2019年6月，我国网民规模达8.54亿人，所以说上网已经是我们国民的常态，而在其中的网上购物更是日渐普及，许多人在日常生活中都喜欢上网购物来满足自己的消费需求。近年来，随着国民的环保节约意识的普及和传播，网购交易平台也出现了专门交易二手是商品的平台，其中二手交易市场更是迅猛发展，二手商品的交易无论是在规模上还是发展趋势都在不断增大。但是当前比较缺乏专门的校园二手交易平台，二手交易市场的需求空间还有很大程度没被发挥。

## 课题研究背景

现代中国的许多家庭主要支出很大一部分是落在孩子的身上，可以说学生虽然不是家庭的主要的经济来源，但却是多数家庭的主要消费力量所在。其中对于离开父母身边，掌握了财务自由的大学生更是中国消费力量的主要分支之一，据调查显示，大学生的年均消费额度已经超过了全国居民的年均可支配额度。也就是说大学生作为主要的消费力量之一是不可小觑的。而在网购这种消费模式下，据搜狐网的调查报告显示，19年国家统计局北京调查总队从网购活跃人群中抽取820个样本，调查其在2018年9至11月的网购消费情况，调查数据显示，网购主力消费群体为女性群体、高学历群体和30至49岁年龄段群体，其中高学历群体主要就是大学生，可以说大学生是网购不折不扣的生力军。

对于学生群体而言，虽然手上会掌握有来自父母的一定的经济条件，但是往往是不能完全满足学生的消费欲望，所以许多的学生都喜欢在网购时不停的挑选便宜又实惠的商品，但是一手的商品往往是按出厂价来判，所以很多时候我们大学生可能遇到自己喜欢的东西但是因为价格原因而退步。但是二手商品却截然不同，许多二手商品虽然已经被他人使用过，但是对于一些类似数码电子产品来讲，二手不会比一手差到哪去，但是价格却是极其令人心动。因此在日常中，学生是最容易接受二手这一概念，而且学习用品和在外的生活用品更新速度的加快已经学生消费能力的增强，还有一些市场经济形势的驱动等客观条件都在为校园二手市场的发展和出现奠定了基础。而且虽然像淘宝和京东亦或是拼多多等等这些大型的网络交易平台已经占领了绝大部分市场，但是其中的不足之处就是缺乏地域特色，没能够很好的满足一个地方消费人群的需求，而且因为商品杂多，有时还会让人眼花缭乱，并且这些大型的网购交易平台针对性不强，一定程度上忽略了大学生这一人群的消费特点。

## 课题研究意义

随着党领导的金山银山就是绿水青山的概念不断普及和传播，学生的环保和节约意识也有在增强，建立一个二手交易平台专门为某所学校服务是很有必要的。发展一种规范合理的校园电子商务也是一种创新的新形势。

二手交易平台能够提供许多方便学生学习和生活的商品信息，例如:便宜的学习资料，课外书籍，还有体育用品和耳机或手机及笔记本等电子商品。我们学校地域扩大，许多学生都买有个人的电动车，方便自己上下课使用，最近甚至因为电动车过多而采取限制电动车出入的管理规定。所以电动车可以说在大学生中是十分抢手的，但是买新的电动车多数人会嫌太贵，买二手的就刚刚好，用个三年五载的再便宜转手卖掉，方便经济又实用。四六级或专业英语，又或者实计算机等级考试已经考研等等一系列各种考试，每年都会有一大堆学生去参加这些考试，但是很多人都是买一手的考试资料，但是二手除了差上一次的考试真题外，其实和一手的没有太多不同，而且很多人可能只是看了几眼就去考试了，所以买一手资料往往很亏，因为师兄师姐们就有一大堆学习资料无处安放，一手的往往都有好几十块钱，但如果二手入手的话可能只要一半不到的价格，而且有些还有师兄师姐的真传笔记，这就好比获取前辈们的学习经验，这可比一手的好多了。

建立一个我校专用的二手交易平台，可以让大学生们解决生活的闲置物品，避免宿舍物品堆积，其中可以让学生主动去接触自己讲要出售的商品的二手信息，让学生了解更多二手交易和市场交易规则，尝试自己经营一家网店，这不仅让学生创造一定的经济价值，还可以学会如何去经营一家小店。

因为交易的买卖双方都是我校的学生或老师用户，在他们发布供求信息的时候都要提供真实的联系方式，商品质量及价格问题也可以当面对质，这是校园间的交易，不会像网络上的其他交易平台一样，买卖双方可能要承担一定的风险，就算有无理由退货，也是要承担运费自付的风险。

## 国内外现状

我国目前还没有听说有过什么特别著名的校园二手交易平台，但目前国内各大高校已经开始认识到建设校园电子商务的重要性，并且已经开始着手实施了。目前国内的网购平台主要还是阿里巴巴的淘宝、京东、拼多多以及一些短视频的网购交易平台，但是普遍还是存在诚信、安全、物流等问题，法律政策上还有待改善。其中二手交易方式与传统的网络交易有些许不一样，所以一定程度上会比传统的网络交易遇到的问题会多一点。在国外，各国的政府和企业已经意识到电子商务的重要性，清楚的认识到电子商务对经济增长的推动作用，许多国家都有在大力发展电子商务经济在各领域的应用。在电子商务的技术、市场和法律方面，美国都处于较为领先的地位，其中比较著名的网路交易平台有亚马逊和eBay等。虽然国外的电子商务系统已经相对完善，但是也存在一些问题，与国内一样，交易平台主要是不断的扩张和接纳更多的人群从而忽略了针对性这一个问题。马云也经常说过，现在主流的B2C交易模式会往C2B模式发展，也就是说未来将会是商品针对于专门的客户去设计，而不是客户去不得不选择那几类商品。交易平台也是一样，针对性越强，客户体验才会更好，因为二手交易的商务模式当前尚处在探索的阶段，因此真正成熟的二手交易平台不多，市场需求在全球来讲还是处于较大的状态。

## 论文的研究内容

整个系统采用的是前后端分离的结构形式去设计，前端采用web开发，使用HTML语言，其中包括三大层级，结构层（HTML）、表现层（CSS）、行为层（JavaScript）三个层构成web的标准。后端主要采用Node.Js负责系统的逻辑，对数据库进行读写和文件处理，Node是一个服务器端JavaScript解释器并采用最新的ES6规范去编写，数据库方面则是以基于分布式文件存储的MongoDB数据库。

具体的研究内容有以下几个方面：

1. 根据现有主流的计算机网络技术和软件开发技术，选择适合自身个人开发的技术法案。
2. 研究本系统所需的关键技术：HTML、JavaScript、Node.js、Vue、webpack、JSON、AJAX、MongoDB、CSS，其中Vue全家桶的MVVM设计模式，JavaScript客户端脚本技术，Node.js的模块及规范都将是重点。
3. 采用UML对本系统开发进行详细设计，使用Microsoft office Visio画图工具进行用例图、活动图、功能模块图、数据流图来辅助开发
4. 利用软件测试对系统进行测试，保证系统的质量，如单元测试、集成测试、验收测试、系统测试、黑盒测试和白盒测试等方法

## 论文的组织结构

1. 绪论。介绍本课题的研究背景和意义还有国内外的现状以及论文的简单描述
2. 相关技术介绍。讨论前端开发的三大框架react、Vue、angular，JavaScript的特点和ajax技术的重要性，MVVM框架的特点，年轻的后端开发语言node.js的发展和使用
3. 分析系统，用过用例的方式对大学生二手交易平台系统开发进行需求分析，得出系统的用例模型
4. 系统的设计，对系统进行架构设计，对系统进行分成，主要分为模型层（M）、视图层（V）、特殊的控制层（VM）
5. 系统的开发与实现。给出系统的各个功能的具体实现，系统的功能界面图，以上所诉技术是如何搭配使用来开发系统的
6. 系统测试，最终对测试结果进行分析和评价
7. 总结，对工作进行简要的总结，并对往后的维护与升级提出新的构思和设想。

# 第二章 相关技术介绍

## 2.1 HTML5、CSS和JavaScript

HTML本身指的就是超文本标记语言，而HTML5指的是HTML的第五次重大修改，也就是第五个HTML版本。介绍它的功能之前，首先介绍一下它的发展历史，HTML首次以因特网草案的形式发布是在1993年，可以说20世纪90年代就是HTML的1.0到4.0版本的快速发展的年代，随着HTML的发展，W3C（万维网联盟）控制了HTML的规范。在这四个版本陆续快速的发布之后，许多人都认为HTML已经到了末路，对web的标准也开始转移到XML和XHTML上，也就是可拓展的超文本标记语言。不过那时，许多的网站主要还是基于HTML。为了能支持Web应用，HTML当时需要添加一些新的功能和制定新的规范。2004年，WHATWG（Web Hypertext Application Technology Working Group，Web超文本应用技术工作组）创立了HTML5规范。Web2.0这个新词也就是那时被发明出来，旧的静态网站逐渐被具有更多特性的动态网站和社交网站取代。2006年，W3C又重新介入HTML，并于2008年发布了HTML5的工作草案。虽然正式的规范还未发布，但是各大浏览器厂家就已经开始对自己旗下的产品进行升级并支持HTML5的新功能。因为基于浏览器的实验性反馈，HTML5规范也得到了持续性的完善，但是HTML5的规范始终还是草案的形式发布，并不是最终版的。虽然2012年已经有候选推荐版的规范，但计划推荐版还是要等到2022年。如果因为觉得规范没有正式发布而忽略HTML，那就大错特错。因为2012年发布的虽然是候选推荐版的，但这也意味着HTML5的规范其实已经大致编写完成。更重要的是现在浏览器厂家已经支持HTML5很多新的功能。

HTML5其实严格意义上来讲就是单指HTML的第五个版本，但是在IT界来说，HTML却不仅仅是指HTML，它还包括CSS和JavaScript，其中HTML和CSS都是由W3C（万维网联盟制定结构和表现的标准），JavaScript由ECMA（欧洲电脑网商联合会制定标准），一个网页就是由HTML（结构）、CSS（表现）、JavaScript（行为）组成。其中HTML已经介绍过它的历史，而它的使用也是很简单，就是用各种标签去创建网页的结构，例如：<p>这是一个p标签</p>，p标签主要是段落标签，一个网页需要什么结构，类似标题结构用H标签，链接用a标签，一块区域用div等等，架构一个网页先是分析它的构成，然后用不同的标签去构建不同的部分，最终把网页的结构给展示出来，这就是HTML。

CSS负责网页的表示层，CSS可以去定义HTML结构的样式，比如宽度、高度、颜色、字体类型等等，例：div{color：red; width:10px; height:20px;}，这是给一个div元素添加宽度10像素，高度20像素和字体颜色为红色的样式，在最新的CSS3中，还添加了许多的动画效果，但是CSS3的样式功能是不兼容IE8及其以下的浏览器的，所以如果考虑兼容问题的话，在动画效果方面可能需要其它方式去实现。

网页最重要的行为层，JavaScript，这是一个网页的逻辑部分，相当于一个人的思想，没有JavaScript就像是没有思想的人。JavaScript是Netscape公司和Sun公司合作开发的语言。在JavaScript出现之前，web浏览器不过是一种能够显示超文本文档软件的一部分，而在JavaScript出现之后，网页的布局不再局限于枯燥的文本，而是有了交互性。在JavaScript1.0发布时，微软的IE为了跟上脚步，在推出IE3时也发不了自己的VBScript语言并以Jscript为名发布，最后ECMA对JavaScript语言进行了标准化，其结果就是ECMAScript语言，这使得同一种语言又多了一个名字。实际上，现在所说的JavaScript就是ECMAScript。但值得注意的是，JavaScript与Sun公司开发的Java程序语言没有任何联系。最初JavaScript是叫LiveScript，后来选择JavaScript作为正式名称是为了能让大家更容易记住它，但是反而让大家把它和Java混为一谈。事实上，Java在理论上几乎可以部署在任何环境中，但JavaScript却只局限于Web浏览器。JavaScript是一种脚步语言，JavaScript脚步通常只能通过Web浏览器进行解释和执行，所以JavaScript不像Java和C++等编译型程序设计那样用途广泛。但JavaScript基于浏览器提供了BOM（浏览器对象模型）和DOM（文档对象模型）的方法，其中BOM提供了window对象，能够控制浏览器的url挑战和浏览器窗口尺寸和滚动等，DOM提供了对节点的操作，也就是对HTML的标签的操作。

可以说HTML、CSS和JavaScript这三个是相助相辅的，单独拿一个出来都很难有大的作为，但是这三个结合起来则是完全占据前端开发的主要部分。

## 2.2 Vue框架

近年来，国内最火的前端框架当属Vue了，作为一个开源的框架，而它的开发者是位中国人-尤雨溪。Vue是一个基于MVVM模式的一套渐进式框架。它是数据驱动和组件化的思想构建的，采用自底向上增量开发的设计。在讲Vue框架的同时，不得不说下前端另外的几个主要框架。目前Vue、react、angular这三个可以说是前端的三大框架，几乎所有公司的前端项目都是基于这三个框架去开发的。但是在它们三个出现之前，就有一个独霸前端开发，它就是jQuery，前面介绍JavaScript的时候就已经说明它是控制DOM节点来实现页面逻辑的，但是如果全部都用JavaScript去写，则会很麻烦，所以在2006年起，jQuery出现了，它作为一个开源的框架，很多开发者都可以在里面共享自己编写的jQuery方法，所以jQuery会不断进步，它的主要特点就是节点操作简单，浏览器的兼容性特别强。在2009年，angular出现了，它是一种MVC模式的框架，双向数据绑定，脱离了JavaScript的节点操作，是一次大的突破。而在2013年，Facebook公司因为对当时市场上大多数的JavaScript MVC框架不满意，所以就自己编写一套语言去架设他们公司的Instagram ，于是react作为Facebook的内部框架出现了，它提出了virtual DOM的概念，一种基于diff算法进行操作的虚拟节点，大大避免了许多真实节点的操作，在性能上碾压了angular和jQuery。而react出现的第二年，也就在2014年，华人尤雨溪综合了angular与react的优点，他和他的团队们开发了Vue框架，一款高性能、高效率的前端框架。目前Vue已经出了2.0版本，并且已经在各大国内市场成为前端开发的主流之一，而且在近期Vue3.0也即将发布。

对习惯于原生JavaScript和jQuery操作的开发者来说，第一次接触react和Vue这种MVVM开发模式的框架是会有点不习惯，因为它的核心思想由节点操作变化为数据操作，我们将不再以节点操作为主，而是关注Model的变化，让MVVM框架利用自己的机制去自动更新DOM，从而让开发者脱离对DOM节点的操作。例如一个div标签里的内容是1，我们通过JavaScript去改变这个div标签的内容，首先就是要先获取这个div节点，再去给它重新赋值，达到改变div内容的操作。但是在Vue中，我们只需给它赋值一个Vue的属性变量，之后我们只需要改变这个属性变量就可以达到同样的效果，不需要获取节点操作。

Vue框架不止是Vue对象而已，搭建一个基于Vue的前端框架需要用到Vue全家桶，除了Vue对象以外，还包括VueCLI即Vue的脚手架，能够帮助我们快速搭建一个Vue的项目，还有VueRouter（Vue的路由插件），Vue是一个单页面应用简称SPA，而Vue-Router允许我们通过不同的 URL 访问不同的内容。可以实现多视图的单页Web应用。

## 2.3 Node.js

Node是一个轻量又高效，事件驱动、非阻塞式的I/O模型，其实就是JavaScript的兄弟，JavaScript本来只能是Web前端开发的语言，它不能像Java、C++这类后端语言一样，能够对文本或数据库进行增删改查的操作，但是node的出现却打破了这个局面。前端语言可以说是被HTML独霸，但后端语言却是百花齐放，地位高的就有Java、C++、C#、Python等，有地位高的，那也就有地位低的，人都喜欢比个高低，编程语言也一样，像是PHP也是作为一门后端语言，但是市场占有率低，而且不被Java、C++那些大佬看得起，而node这个新生代后端语言的地位就像是PHP一样。Node就是把谷歌浏览器的控制台拿出来，目前市面上最好的解析引擎就是谷歌的Chrome了。只要这个引擎核心安装在哪，哪就可以作为node的开发环境。Node是一个抛弃了HTML和CSS的JavaScript语言，所以它也就没有BOM和DOM控制浏览器的方法，但是它却换来了处理文件和对数据库进行增删改查的各种模块。

Node虽然和JavaScript的语言几乎一样，但是有几个最重要的区别，除了上面所说的BOM和DOM，两者的规范也是不一样的。好的开发都是有自己的一套规范，而node.js是遵循CommonJS的模块化规范，利用require引入模块和module.exports导出模块。前面也说到，node虽然没有了对浏览器的控制，但是却换来了许多模块，例如内置模块fs、http、path，这些都是node自带的模块，不需要安装即可使用，可对数据库进行操作和开启服务器。还有express、jsdom、request等第三方模块，可快速搭建服务器、能够进行爬虫，当然也能够实现自定义的模块。Node就是通过各种强大的模块实现了后端的功能。

## 2.4 本章总结

本章主要介绍了项目所使用的几个关键技术，包括前端页面构成和使用框架的介绍，还有后端开发所用到的语言介绍，并且分别说明了它们的各自的优点和缺点，还延伸去谈到了一些历史和相关技术的发展，有些地方也会举例子去更详细的介绍了使用方法。

# 第三章 系统分析

## 3.1 功能性需求

下面主要分用户功能和管理员功能来介绍功能性需求

### 3.1.1 用户功能

（1）用户登录

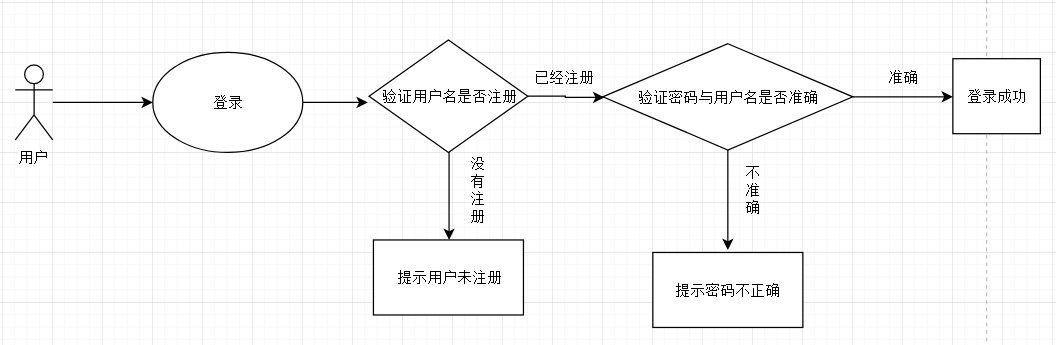


图3.1用户登录活动图

如图3.1所示为用户登录的活动图，用户在浏览商品的时候可不登录，但在收藏商品和想要发布商品的时候，需要登录才可进行这些操作。登录的账号不能为空，密码必须是六位数字以上，登录成功之后会回到登录之前的页面，失败会给对应的提示。若忘记密码，可在登录页面点击找回密码功能，用户只需提供用户名和注册时对应的手机号即可找回密码。

（2）用户注册

用户注册需提供注册的用户名，学号，密码，宿舍地址，手机号码，微信号，班级，为保证避免重复注册的情况存在，已注册过的用户名、学号和手机号码以及微信号不能再重新注册。

（3）用户查看其它用户的信息

用户点击其他用户头像或昵称时可以查看该用户的详细信息，包括用户名、学号、电话、班级，微信号等。

（4）用户修改个人信息

用户在我的界面可以进入修改个人信息页面，除了学号不可修改以外，其它信息均可修改。

（5）用户管理自己的商品

用户可在我的页面发布自己想要销售的商品，发布商品需提供商品名字、图片、价格、商品数量、商品的属性，联系信息将会自动根据当前用户注册时的信息进行填充，如果需要修改也可以进行联系信息修改，最好点击确认即可发布商品。用户在我的商品中可以对商品信息进行修改，也可以删除发布的商品信息。

（6）用户收藏喜欢的商品

用户在浏览商品的详情页的时候，可点击收藏进行商品的收藏，到时候可以在收藏页面查看自己收藏过的商品。点击收藏的商品也可以立即跳转到商品详情页，如果卖家已经下架该商品，则无法跳转并给出对应的提示。

（7）用户退出

用户可在我的页面点击退出来退出当前用户，退出后会跳到登录页面，并且只能查看首页，分类页和商品的详情页。

### 3.1.2 管理员功能

（1）管理员登录

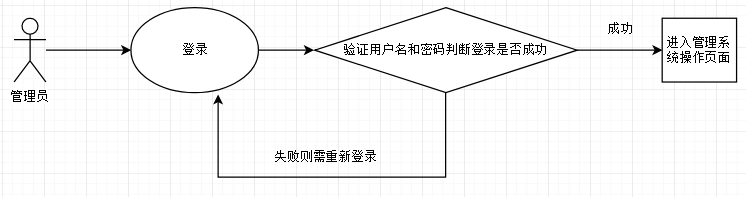


图3.2管理员登录的活动图

管理员进入管理系前必须先登录。系统管理员初始账号为admin，密码为admin，登录之后才可以修改管理员密码。

（2）管理员修改登录密码

管理员登录成功之后可以修改在修改密码界面可以修改登录的密码，在修改密码页面需填写旧密码、新密码和确认新密码三个表单。三个表单填写准确之后，验证通过才能修改密码成功。

（3）管理员管理用户和商品信息

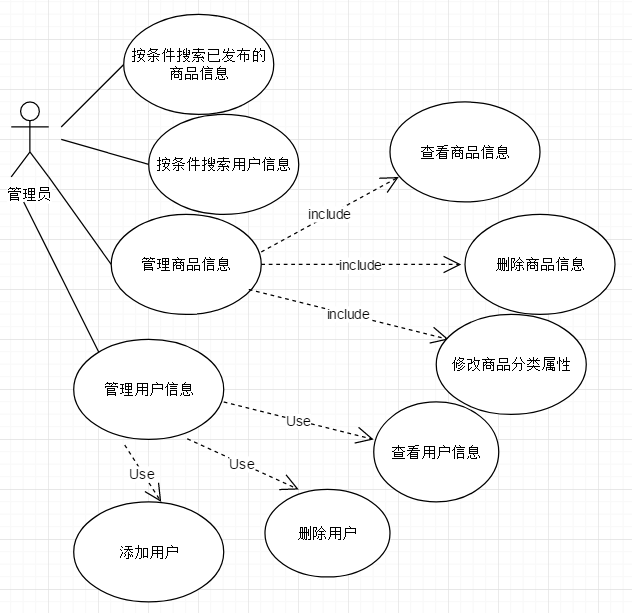


图3.3管理用操作用户信息用例图

管理员可在商品信息列表中查看商品信息详情，包括发布者信息用户名、手机号、微信号、班级、学号和所留的联系信息宿舍地址、联系电话还有商品的名称、价格、数量、图片等信息。

管理员也可查看用户信息，查看用户发布的商品信息，如果用户发布的信息不合法或包含有虚假成分等，管理员可直接删除该用户，那这个用户及其所发布的商品信息都会一并删除。删除后的用户将不能再登录交易平台，其所发布的商品信息也将在交易平台上下架。若某些用户的账号存在问题，用户可联系管理员，管理员就可根据添加用户来重新帮他设置账号信息。

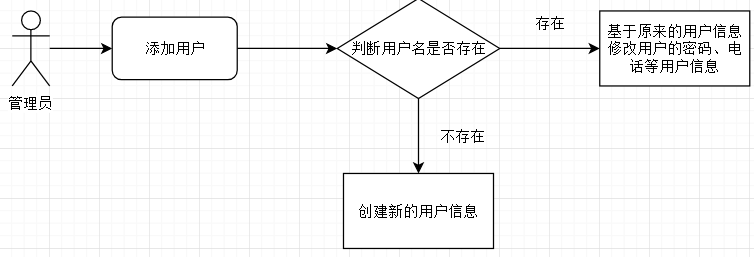


图3.4管理员添加用户活动图

（4）管理员退出

管理员点击退出之后，管理系统将会退出操作页面并跳转到登录页面。

## 3.2 可行性分析

该系统主要是提供一个新的校园交易发布方式给同学，提供一个学生与学生之间进行交易的平台，让校园内的各种交易信息整合在一起，方便学生用户去浏览所需的交易。因为该系统只是针对单个学校的学生群体，所以对于用户负载来讲规模不会是很大。从开发条件来说，因为不会直接在平台上设计金钱交易和拥有短信验证之类的操作（支付功能需要企业级的手续申请才行，个人项目开发是不能申请支付宝支付和微信支付的，短信验证也需要购买第三方短信验证接口才行），加上选择的开发工具都是开源免费的，所以开发条件是可行的。从技术上来件，各大编程平台有提供技术分享，而且也有类似的电子商务网站开发经验借鉴，所以技术上也是可行的。从运维方面来考虑，登录该平台只需一个浏览器即可，数据库和项目都可以放在一台服务器上运行服务器可在阿里云或腾讯云等国内各大知名服务器提供平台购买，而且学生购买有学生价，价格不贵。

通过以上需求分析和可行性分析，可以肯定设计和开发这个校园二手交易平台是可行的，能为校园网络环境增添新的色彩。

## 3.3 本章总结

本章介绍了功能性需求和提供可行性分析，而功能性需求分为用户和管理员两个角色，管理员功能主要是利用后台管理系统去管理用户、商品等信息，用户功能主要是浏览商品信息、发布商品信息、找回密码和注册用户等。

# 第四章 系统设计

海大二手交易平台是一个为海大学子提供寻找和发布商品的平台。根据常规的网上购物模式，系统的主要功能模块分为前台的浏览和查找商品、用户登录和注册及收藏，还有查看卖家联系方式等，后台则为管理员维护用户、商品的合法性，避免虚假交易出现。下面主要用面向对象的方法给出系统的设计。

## 4.1 概要设计

本章概要设计主要分为系统体系结构设计和系统的模块设计。

### 4.1.1 体系结构设计

编程开发主要有两种开发模式，一种是面向过程的开发模式，还有一种是面向对象的开发模式。本次体系结构设计主要是围绕面向对象的开发思想去构思的。面向对象的思维模式主要包含封装、继承、多态这三大特性。封装主要是为了能让我们的代码能够得到复用性，避免相同的模块重复去写代码。继承则是继承上一个对象的属性和方法，方便更好的维护。多态表示一个函数若同一操作作用于不同的对象展示出不同的功能，产生不同的执行结果。由于这三个特性，设计出的系统结构具有高内聚、低耦合的特点，能够让我们的项目更灵活、拓展性更高。

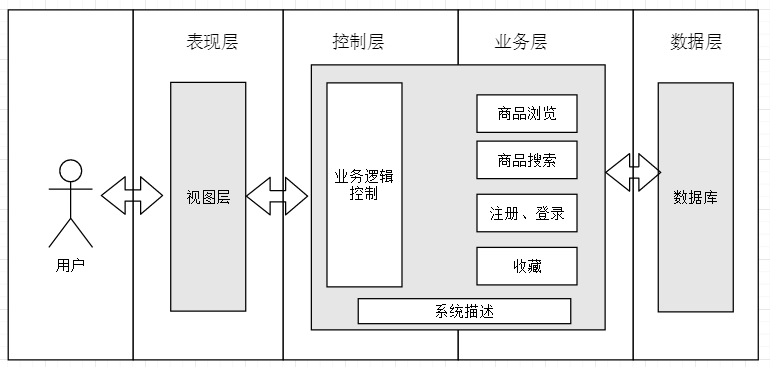


图4.1 概要功能架构图

项目的框架设计上，主要采用4层架构的方式去设计和实现的，主要包括：表现层、控制层、业务层、数据层，如图4.1.的架构图所示。

表现层：特指前端页面，主要负责根据不同的业务逻辑控制去展示不同的信息，利用了JavaScript和Ajax异步请求与服务器进行交互。页面则用html和css去布局美化，让用户有更好的界面体验。

控制层：主要负责根据用户在页面的操作去控制页面有对应的响应。主要是控制页面的跳转，搜索、登录、查看商品信息等的请求，再根据请求去渲染页面，在同一个页面里展示更多的信息给用户。有些涉及权限的页面还会用路由拦截去实现，比如在没有登录的情况下，用户是不能点击收藏的。

业务层：将整个项目分为几大模块，不同的模块根据自己所在业务的需求去调用不同的接口，从数据库请求不同的数据来显示到自己的业务中。

数据层：根据不同的业务将数据库划分不同的集合，在对应的集合里去存放对应业务所需的数据。

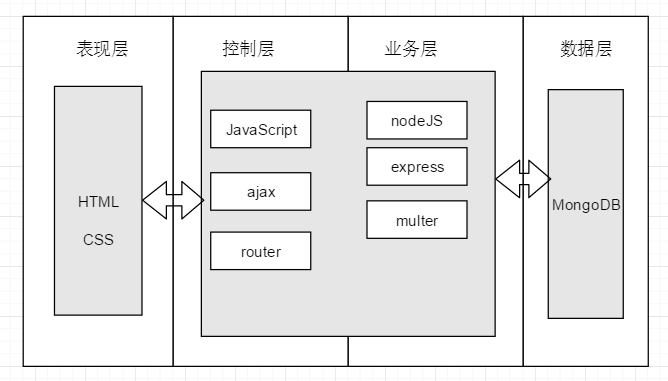


图4.2 技术架构

具体的每层所用到的技术如图4.2所示，html和css展示界面，JavaScript控制页面的逻辑，ajax负责异步请求与服务器进行数据交互，router路由负责页面的跳转。系统的开发环境为windows操作系统，node作为后端接口开发环境，express搭建服务器，multer接受图片上传，MongoDB作为数据库存储数据。

### 4.1.2 系统模块设计

整个系统模块可以分为两大部分，一个是前台的用户功能模块如图4.3所示，还有一个是后台管理员系统功能操作模块如图4.4所示。

前台功能模块再划分的话，可以分为登录和未登录两个状态，未登录的状态下，用户同样可以享受到商品浏览和搜索的功能，但是用户不能在商品详情页收藏喜欢的商品到收藏页面，而且用户在未登录状态是不能发布商品去销售的，只有用户登录之后才能享受到收藏商品和发布商品的功能。修改个人信息也是要在用户登录之后才能操作。用户登录之后，在享受到发布商品的功能下，用户自然就可以管理自己发布的商品，例如下架商品，修改商品信息，查看自己发布的商品等功能，可以查看自己收藏的商品。

后台功能模块，管理员必须登录才能进入后台管理系统页面对商品和用户信息进行操作管理。商品管理方面，管理员不能随意修改商品信息，但是遇到不合法或不合理的商品信息，管理员有资格对商品信息进行下架删除处理。对于商品类别用户发布不合理的情况，管理员可以更改商品的类别，达到更好的分类效果。用户管理方面，管理员可以修改用户的信息，若修改的用户已存在，则在原有的用户信息上修改，若想要修改的用户不存在，则会新建一个用户或是提示用户不存在，无法修改用户信息等。对于发布不良商品，恶意破坏平台交易环境的用户，管理员可以对其进行用户删除处理，使该用户不得再次在平台发布恶意信息，而且被管理员删除的用户是无法再用相同的手机号码或是账号进行注册和登录。管理员还可以对用户关键信息进行搜索，若接受到其他用户的举报，管理员可以搜索到对应用户，并查看他的用户信息，假如举报信息核实无误，管理员可对其进行用户删除操作。管理员在点击退出之后，后台管理系统将立即退出并跳转到登录界面。

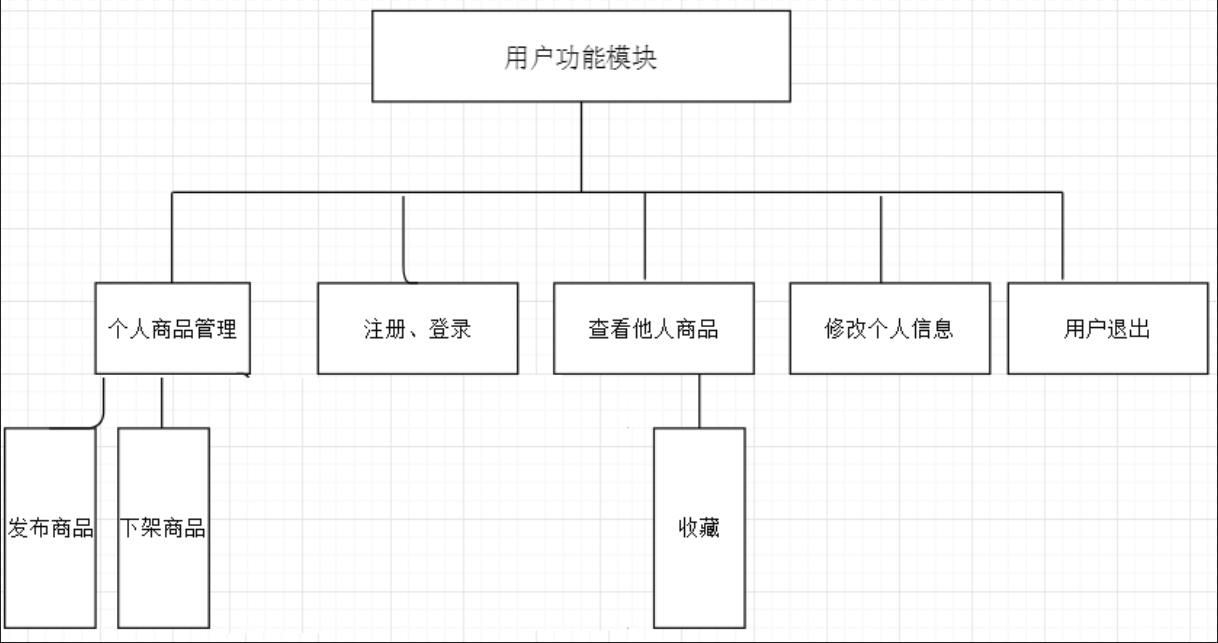


图4.3前台功能模块

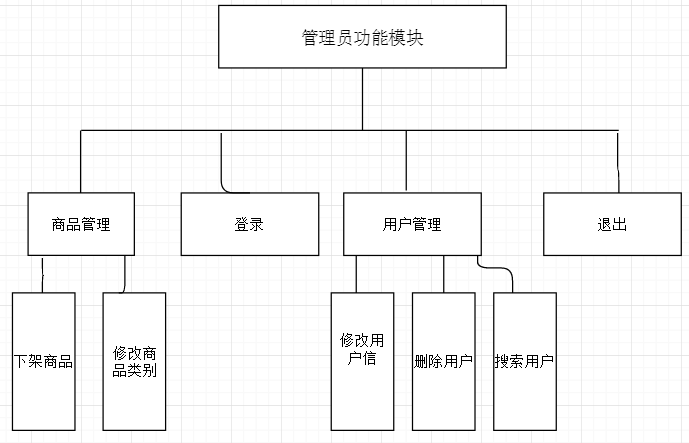


图4.4后台功能模块

## 4.2 系统详细设计

系统详细设计包括数据库设计、项目目录设计以及功能设计

### 4.2.1 数据库设计

数据库我们采用MongoDB的数据库去实现数据的存储，数据库的名称为GDOU\_SHTP，MongoDB数据库和MySQL数据库不一样，MySQL是一个关系型数据库，而MongoDB是一个非关系型的数据，在MySQL中的表table相对于MongoDB就是集合collection，数据row就相对于文档document。海大二手交易平台的数据库会有首页的轮播图推荐商品数据（slideshow）、发布的商品信息数据（products）、用户信息数据（users）、管理员信息数据（）四大集合，如下有商品信息表、用户信息表、商品属性表来介绍数据库的字段含义。

表4-1 商品信息表

|  |
| --- |
| 字段名 数据类型 允许非空 |
| PRODUCT\_NAME string no  PRODUCT\_PRICE string no  PRODUCT\_NUM number no  PRODUCT\_PIC array yes  PRODUCT\_CLASS string no  SELLER\_NAME string no  SELLER\_DORM string no  SELLER\_ID number no  SELLER\_PHONE number no  SELLER\_WECHAT string yes  SELLER\_CLASS string no  SELLER\_SNO string no  RELEASE\_DATE date no  DEADLINE date no  DESCRIBE string no |

商品信息表如表4-1所示，PRODUCT\_NAME（商品名称，不能为空，无默认值）、PRODUCT\_PRICE（商品价格，不能为空，无默认值）、PRODUCT\_NUM（商品数量，不能为空，默认值为1）、PRODUCT\_PIC（商品图片，允许为空，无默认值）、PRODUCT\_CLASS（商品属性，不允许为空，默认值为others其它）、SELLER\_NAME（卖家姓名，不允许为空，无默认值）、SELLER\_DORM（卖家宿舍地址，不允许为空，无默认值）、SELLER\_ID（卖家id，不允许为空，系统会自动填充该信息）、SELLER\_PHONE（卖家联系电话，不允许为空，无默认值）、SELLER\_WECHAT（卖家微信，允许为空，无默认值）、SELLER\_CLASS（卖家所在班级，不允许为空，无默认值）、SELLER\_SNO（卖家学号，不允许为空，无默认值）、RELEASE\_DATE（商品发布日期，不允许为空，值为发布时系统时间）、DEADLINE（发布截至日期，不允许为空，默认值为发布日期的三天后，如果卖家不再操作商品延长发布时间，商品会自动表示已售出，避免用户已售出却忘记下架的情况，若用户想要继续上架，需在发布商品后的第三天延长发布时间，这样商品会自动延长时间）、DESCRIBE（对商品的介绍，不允许为空，无默认值）。在数据库中的数据格式如下所示：

{

"\_id" : ObjectId("5e92e665c12cea49871a635d"),

"PRODUCT\_NAME" : "小强",

"PRODUCT\_PRICE" : "9999",

"PRODUCT\_NUM" : 1,

"PRODUCT\_PIC" : [

"https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695402667&di=5ba644fa8e2080f9c4a2ab9f34c4e136&imgtype=0&src=http%3A%2F%2Fi0.sinaimg.cn%2Fedu%2F2012%2F0428%2FU3834P42DT20120428095633.jpg",

"https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695492598&di=137287d78f302546a47abbeb22e0e1a2&imgtype=0&src=http%3A%2F%2Fgss0.baidu.com%2F-fo3dSag\_xI4khGko9WTAnF6hhy%2Fzhidao%2Fpic%2Fitem%2Ff9198618367adab4dcde5f9b8cd4b31c8601e4d8.jpg"

],

"PRODUCT\_CLASS" : "pet",

"SELLER\_NAME" : "帝王东",

"SELLER\_PHONE" : "1357924610",

"SELLER\_WECHAT" : "wxid\_842yjadwjwt721",

"SELLER\_CLASS" : "信计1161",

"SELLER\_SNO" : "201611921119",

"RELEASE\_DATE" : "2020-04-12 08:09:04",

"DEADLINE" : "2021-04-12 08:09:04",

"DESCRIBE" : "很可爱的小强，希望大家会喜欢，只要给它吃垃圾就行，生命力强不怕养死，给你一只还能带给你更多只，只赚不亏的宠物"

}

表4-2 用户信息表

|  |
| --- |
| 字段名 数据类型 允许非空 |
| USER\_NAME string no  USER\_SNO string no  USER\_PSW number no  USER\_DORM string no  USER\_CLASS string no  USER\_PHONE number no  USER\_WECHAT string yes  USER\_COLLECT array yes |

用户信息表如表4-2所示，USER\_NAME（用户名，不允许为空，无默认值）、USER\_SNO（用户学号，不允许为空，无默认值）、USER\_PSW（用户密码，不允许为空，无默认值）、USER\_DORM（用户宿舍地址，不允许为空，无默认值）、USER\_CLASS（用户所在班级，不允许为空，无默认值）、USER\_PHONE（用户联系电话，不允许为空，无默认值）、USER\_WECHAT（用户微信，允许为空，无默认值）、USER\_COLLECT（用户收藏的商品，允许为空，这里将存放商品信息的\_ID，用户登录之后点击我的收藏页面时，可根据用的USER\_COLLECT中存放的商品信息\_ID去商品信息集合查找用户收藏的商品信息）。用户信息在数据库的数据格式为：

{

"\_id" : ObjectId("5e92e3dcc12cea49871a6270"),

"USER\_NAME" : "帝王东",

"USER\_SNO" : "201611921119",

"USER\_PSW" : "123456",

"USER\_DORM" : "海安A309",

"USER\_CLASS" : "计数1165",

"USER\_PHONE" : "13579246810",

"USER\_WECHAT" : "wxid\_842yjadwjwt721",

"USER\_COLLECT" : [

"5e92e1e6c12cea49871a620d"

]

}

表4-3商品属性表

|  |
| --- |
| 值 属性 |
| digital 数码  sports 运动  books 书籍  lives 生活用品  traffic 交通工具  dress\_shoes 衣鞋  music 音乐  pet 宠物  others 其它 |

表4-3为商品属性表，digital、sports、books、lives、traffic、dress\_shoes、music、pet、others代表的值是商品信息里的PRODUCT\_CLASS商品属性的值，用户在发布商品的时候，在商品属性的表单里会被提供以上属性值供其选择，用户可以选择对应的属性值，分类页面会根据PRODUCT\_CLASS商品属性的值去划分商品，这样买家在分类页面查找自己想要的相关商品时可以更快的找到自己合适心仪的商品。

表4-4 管理员信息表

|  |
| --- |
| 字段名 数据类型 允许非空 |
| ADMIN\_NAME string no  ADMIN\_PSW number no |

表4-4为管理员信息表，因为管理员的权重很大，所以管理员除了更改数据库之外没有其它方式去添加管理员，管理员只有ADMIN\_NAME（管理员名称，不允许为空，用来验证管理员的名称）、ADMIN\_PSW（管理员登录所用的密码，不允许为空，无默认值），后台管理系统根据这两个去验证管理员信息。在数据库的格式如下所示：

{

"\_id" : ObjectId("5e92dd5dc12cea49871a6181"),

"ADMIN\_NAME" : "admin",

"ADMIN\_PSW" : "suntek"

}

表4-5首页轮播图商品推荐信息表

|  |
| --- |
| 字段名 数据类型 允许非空 |
| SLIDESHOW array no |

{

"\_id" : ObjectId("5e92ea4fc12cea49871a64fc"),

"SLIDESHOW" : [

"5e92e665c12cea49871a635d",

"5e92e53bc12cea49871a62f1",

"5e92e1e6c12cea49871a620d "

]

}

表4-5为首页轮播图的商品推荐信息表，管理员可选择几款商品作为商品推荐放到首页的轮播图，让大家更容易看到，这个集合只有一个字段，该字段存放的是商品的\_ID，轮播图渲染的时候在请求slideshow之后，会根据这些\_ID在商品信息集合中把商品的第一张图给请求回来，并且点击轮播图跳转商品信息详情页时，也可以根据商品的\_ID去渲染商品的信息详情。

# 第五章 系统实现

## 5.1 后端接口实现

### 5.1.1 用户信息模块的功能接口实现

用户在注册时，填写信息的时候，需要对一些信息进行判断，避免信息重复注册，例如用户名、学号、手机号、微信号这几个在被注册之后是不允许重复注册的。后端接口node.js代码如下：

（1）检测用户名是否存在

router.get('/checkUsername',async (req,res)=>{

    let {username} = req.query;

    let result = await mongo.find(colName,{USER\_NAME:username});

    if(result.length>0){//有搜到数据的话，代表存在该用户了

        res.send(formatData({data:1}));

    }else{

        res.send(formatData({data:0}));

    }

})

请求url：/users/ checkUsername

请求方法：get

参数：

username ：将要注册的用户名

返回：

{

"code": 1,

"msg": "success",

"data": 1 //返回为1代表用户名已经注册，返回0代表未注册

}

（2）检测学号是否存在

router.get('/checkUsersno',async (req,res)=>{

    let {usersno} = req.query;

    let result = await mongo.find(colName,{USER\_SNO:usersno});

    if(result.length>0){//有搜到数据的话，代表存在该用户了

        res.send(formatData({data:1}));

    }else{

        res.send(formatData({data:0}));

    }

})

请求url：/users/ checkUsersno

请求方法：get

参数：

usersno：将要注册的用户学号

返回：

{

"code": 1,

"msg": "success",

"data": 1 //返回为1代表用户学号已经注册，返回0代表未注册

}

（3）检测手机号是否存在

router.get('/checkUserphone',async (req,res)=>{

    let {userphone} = req.query;

    let result = await mongo.find(colName,{USER\_PHONE:userphone});

    if(result.length>0){//有搜到数据的话，代表存在该用户了

        res.send(formatData({data:1}));

    }else{

        res.send(formatData({data:0}));

    }

})

请求url：/users/ checkUserphone

请求方法：get

参数：

userphone：将要注册的用户手机号

返回：

{

"code": 1,

"msg": "success",

"data": 1 //返回为1代表用户手机号已经注册，返回0代表未注册

}

（4）检测微信号是否存在

router.get('/checkUserwechat',async (req,res)=>{

    let {userwechat} = req.query;

    let result = await mongo.find(colName,{USER\_WECHAT:userwechat});

    if(result.length>0){//有搜到数据的话，代表存在该用户了

        res.send(formatData({data:1}));

    }else{

        res.send(formatData({data:0}));

    }

})

请求url：/users/ checkUserwechat

请求方法：get

参数：

userwechat：将要注册的用户微信号

返回：

{

"code": 1,

"msg": "success",

"data": 1 //返回为1代表用户微信号已经注册，返回0代表未注册

}

在通过上面的接口，用户注册的信息返回的都是0未注册的情况下，允许用户注册，注册的后端接口代码如下：

（5）用户注册

router.post('/register',async (req,res)=>{

    let {

        username,

        usersno,

        userpsw,

        userdorm,

        userphone,

        userwechat,

        userclass

    } = req.body;

    if(username&&usersno&&userpsw&&userdorm&&userphone&&userclass){

        try{

           let result = await mongo.create(colName,[{

             USER\_NAME : username,

             USER\_SNO : usersno,

             USER\_PSW : userpsw,

             USER\_DORM :userdorm,

             USER\_CLASS : userclass,

             USER\_PHONE : userphone,

             USER\_WECHAT : userwechat,

             USER\_COLLECT : []

             }]);

           res.send(formatData({data:result}));

        }catch(err){

           res.send(formatData({code:0,data:err}))

        }

    }else{

        res.send(formatData({code:0,data:"必要信息不能为空"}))

    }

})

请求url：/users/ register

请求方法：post

参数：

username：用户名

usersno:用户学号

userpsw：用户密码

userdorm：用户宿舍地址

userphone：用户手机号

userwechat：用户微信号

userclass：用户所在班级

返回：

{

"code": 1,//code为1代表注册成功，为0代表注册失败

"msg": "success",

"data": {

"result": {

"ok": 1,

"n": 1

},

"ops": [

{

"USER\_NAME": "吴作龙",

"USER\_SNO": "202011921119",

"USER\_PSW": "wng",

"USER\_DORM": "海安A309",

"USER\_CLASS": "计数1165",

"USER\_PHONE": "13560982882",

"USER\_WECHAT": "wuzhuolong803",

"USER\_COLLECT": [],

"\_id": "5e9c353b1de54b4cc4232efc"

}

],

"insertedCount": 1,

"insertedIds": {

"0": "5e9c353b1de54b4cc4232efc"

}

}

}

用户注册成功之后就自动跳转到登录页面，用户输入用户名和密码即可登录，用户登录的代码如下：

（6）用户登录

router.post('/login',async (req,res)=>{

    let {

        username,

        userpsw

    } = req.body;

    let result = await mongo.find(colName,{USER\_NAME:username,USER\_PSW:userpsw});

    if(result.length>0){//有搜到数据的话，代表存在该用户了

        res.send(formatData({data:result}));

    }else{

        res.send(formatData());

    }

})

请求url：/users/ login

请求方法：post

参数：

username：用户名

userpsw：用户密码

返回：

{

"code": 1,

"msg": "success",

"data": [//data数组有数据代表登录成功，并返回当前登录用户的用户信息，登录失败data数组为空

{

"\_id": "5e9c353b1de54b4cc4232efc",

"USER\_NAME": "吴作龙",

"USER\_SNO": "202011921119",

"USER\_PSW": "wng",

"USER\_DORM": "海安A309",

"USER\_CLASS": "计数1165",

"USER\_PHONE": "13560982882",

"USER\_WECHAT": "wuzhuolong803",

"USER\_COLLECT": []

}

]

}

用户在登录成功之后，可修改自己的用户名，用户密码，宿舍地址，用户手机号，用户微信号，用户班级，用户的商品收藏，而管理员则可以修改用户的所有信息，用户信息修改的接口代码如下：

（7）修改用户信息

router.post('/update',async (req,res)=>{

    let {

        id,

        username,

        usersno,

        userpsw,

        userdorm,

        userphone,

        userwechat,

        userclass,

        usercollect

    } = req.body;

    let obj = {

        USER\_NAME:username,

        USER\_SNO:usersno,

        USER\_PSW:userpsw,

        USER\_DORM:userdorm,

        USER\_PHONE:userphone,

        USER\_WECHAT:userwechat,

        USER\_CLASS:userclass,

        USER\_COLLECT:usercollect

    }

    let data = {}

    for(var key in obj){

        if(obj[key]){

          data[key] = obj[key]

        }

    }

    try{

        let result = await mongo.update(colName,{\_id:id},data);

        res.send(formatData({data:result}));

    }catch(err){

        res.send(formatData({data:err}));

    }

})

请求url：/users/ update

请求方法：post

参数：

id:用户\_id，用户信息修改是根据这个去查找唯一用户信息修改

username：用户名

usersno:用户学号

userpsw：用户密码

userdorm：用户宿舍地址

userphone：用户手机号

userwechat：用户微信号

userclass：用户所在班级

usercollect：用户的收藏

返回：

{

"code": 1,

"msg": "success",

"data": {//data对象不为空代表修改成功，修改失败data对象为空对象

"result": {

"n": 1,

"nModified": 1,

"ok": 1

},

"connection": {

"id": 0,

"host": "localhost",

"port": 27017

},

"modifiedCount": 1,

"upsertedId": null,

"upsertedCount": 0,

"matchedCount": 1,

"n": 1,

"nModified": 1,

"ok": 1

}

}

在后台管理系统中，管理员为了方便管理用户，可以根据用户名、学号、班级、宿色地址、手机号、微信号查找相关用户并支持模糊查找，用户信息查找的接口代码如下：

（8）根据用户名、学号、班级、宿色地址、手机号、微信号查找相关用户

router.post('/search',async (req,res)=>{

    let {

      keywords,

      pagesize,

      pagenum

    } = req.body;

    let searchUserList;

    if(keywords){

        searchUserList = await mongo.find(colName,[

            {USER\_NAME:{$regex:keywords}},

            {USER\_SNO:{$regex:keywords}},

            {USER\_CLASS:{$regex:keywords}},

            {USER\_DORM:{$regex:keywords}},

            {USER\_PHONE:{$regex:keywords}},

            {USER\_WECHAT:{$regex:keywords}}

        ])

    }else{

        if(pagenum&&pagesize){

            searchUserList = await mongo.find(colName,{},pagenum,pagesize);

        }else{

            searchUserList = await mongo.find(colName);

            res.send(formatData({data:searchUserList.length}));

            return;

        }

    }

    if(searchUserList.length>0){

        res.send(formatData({data:searchUserList}));

    }else{

        res.send(formatData({data:[]}));

    }

})

请求url：/users/ search

请求方法：post

参数：

Keywords：查找的关键字 ，不传的时候则根据pagesize和pagenum查找所有用户中对应数量和页码的用户，如果都不传，则返回所有用户的数量

Pagesize：请求一次的数据数量

Pagenum：页码

返回：

{

"code": 1,

"msg": "success",

"data": [//查找到相关用户会返回查找到的用户，查找为空则会提示没有查到相关用户

{

"\_id": "5e9c353b1de54b4cc4232efc",

"USER\_NAME": "吴卓龙",

"USER\_SNO": "202011921119",

"USER\_PSW": "wng",

"USER\_DORM": "海安A309",

"USER\_CLASS": "计数1165",

"USER\_PHONE": "13560982882",

"USER\_WECHAT": "wuzhuolong803",

"USER\_COLLECT": []

}

]

}

在后台管理系统中，管理员可根据用户的举报和调查，若发现相关用户存在违法和不良的交易行为，可删除该用户账号，避免该用户继续利用平台环境传播违法犯罪的信息，删除用户的接口代码如下：

（9）根据\_id删除用户

router.post('/del',async (req,res)=>{

    let {

        id

    } = req.body;

    try{

       await mongo.remove(colName,{\_id:id})

       res.send(formatData({data:"用户删除成功"}))

    }catch (err){

       res.send(formatData({code:0,data:err}))

    }

})

请求url：/users/ del

请求方法：post

参数：

id：要删除的用户的id

返回：

{

"code": 1,

"msg": "success",

"data": "用户删除成功"

}

在收藏页面，需要根据用户的id去请求接口查询数据库得到收藏的商品详细信息，具体的接口代码如下：

（10）根据用户\_id返回用户收藏的商品

router.post('/collect',async (req,res)=>{

    let {

        id

    } = req.body;

    try{

        let users = await mongo.find(colName,{\_id:ObjectId(id)});

        let collectIdList = users[0].USER\_COLLECT;

        if(collectIdList.length>0){

            let collectList=[];

            let reqId = [];

            await collectIdList.forEach(async function(item){

                reqId.push({\_id:ObjectId(item)})

            })

            let data = await mongo.find('products',reqId);

            collectList = collectList.concat(data);

            res.send(formatData({code:1,data:collectList}))

        }else{

            res.send(formatData({code:1,data:[]}))

        }

    }catch (err) {

        res.send(formatData({code:0,data:err}))

    }

})

请求url：/users/ collect

请求方法：post

参数：

id：要查询的用户的id

返回：

{

    "code": 1,

    "msg": "success",

    "data": [//有收藏商品则返回的data数组有数据，否则为空数组

        {

            "\_id": "5e92e1e6c12cea49871a620d",

            "PRODUCT\_NAME": "浩东牌电动车",

            "PRODUCT\_PRICE": "309",

            "PRODUCT\_NUM": 1,

            "PRODUCT\_PIC": [

                "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586693622248&di=08fedcc7f0d14d8a5534a9419fea1749&imgtype=0&src=http%3A%2F%2Fimg.jdzj.com%2FUserDocument%2Fwfwangge%2FPicture%2F201110259548.jpg"

            ],

            "PRODUCT\_CLASS": "traffic",

            "SELLER\_NAME": "帝王东",

            "SELLER\_DORM": "海安A309",

            "SELLER\_ID": "5e92e3dcc12cea49871a6270",

            "SELLER\_PHONE": "1357924610",

            "SELLER\_WECHAT": "wxid\_842yjadwjwt721",

            "SELLER\_CLASS": "信计1161",

            "SELLER\_SNO": "201611921119",

            "RELEASE\_DATE": "2020-04-12 08:09:04",

            "DEADLINE": "2021-04-12 08:09:04",

            "DESCRIBE": "当年我就是靠这台爱马仕电动车上的学校危险驾驶通报名单第一名，因为学校内没有一台电动车能追得上我这台，而且有了它还能撩妹兜风，我就是靠它把到隔壁班的翠花，校园内任你飚，不怕撞环校车,我撞了好多次车都没事，就是我有点事，可以说车绝对牛逼克拉斯，因要毕业了，所以忍痛割爱卖掉这台浪里小白龙，想要就联系我"

        },

        {

            "\_id": "5e92e53bc12cea49871a62f1",

            "PRODUCT\_NAME": "华为P40",

            "PRODUCT\_PRICE": "1165",

            "PRODUCT\_NUM": 1,

            "PRODUCT\_PIC": [

                "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695199935&di=3386c6f8bd13a8452ed43238c116d728&imgtype=0&src=http%3A%2F%2Fdingyue.ws.126.net%2F2020%2F0410%2Fe9a0a0f2j00q8kkgk001nc000hs00adm.jpg",

                "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695199928&di=df3f5f42072f330a82ede78179a0a9a4&imgtype=0&src=http%3A%2F%2Fn.sinaimg.cn%2Fsinacn20190510s%2F492%2Fw1920h972%2F20190510%2F9df1-hwsffzc2076045.jpg",

                "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695199927&di=af7d2965d6d737e700ec811a1b139428&imgtype=0&src=http%3A%2F%2Fdingyue.ws.126.net%2F2020%2F0325%2F4671b4fdj00q7qd9c002ec001jk00rsm.jpg"

            ],

            "PRODUCT\_CLASS": "digital",

            "SELLER\_NAME": "帝王东",

            "SELLER\_DORM": "海安A309",

            "SELLER\_ID": "5e92e3dcc12cea49871a6270",

            "SELLER\_PHONE": "1357924610",

            "SELLER\_WECHAT": "wxid\_842yjadwjwt721",

            "SELLER\_CLASS": "信计1161",

            "SELLER\_SNO": "201611921119",

            "RELEASE\_DATE": "2020-04-12 08:09:04",

            "DEADLINE": "2021-04-12 08:09:04",

            "DESCRIBE": "大一时候托哆啦A梦给我买的，因为要发布了，所以现在不想要了，我已经叫多啦A梦给我买华为P100了，这台老式机就给你们吧"

        }

    ]

}

### 5.1.2 商品信息模块的功能接口实现

用户在登录之后，可在我的页面里，我发布的商品入口进入用户自己发布的商品页面，能够查看自己当前发布的商品，具体的接口代码如下：

（1）根据用户id查找该用户销售的商品

router.post('/sellgoods',async (req,res)=>{

    let {sellerid} =req.body;

    let sellgoodsResult = await mongo.find(colName,{SELLER\_ID:sellerid});

    if(sellgoodsResult.length>0){

        res.send(formatData({data:sellgoodsResult}));

    }else{

        res.send(formatData({data:"没有商品发布"}));

    }

})

请求url：/products / sellgoods

请求方法：post

参数：

sellerid：用户\_id

返回：

{

"code": 1,

"msg": "success",

"data": [//返回用户发布的所有商品信息，若未发布会提示没有商品发布

{

"\_id": "5e92e1e6c12cea49871a620d",

"PRODUCT\_NAME": "电动车",

"PRODUCT\_PRICE": "309",

"PRODUCT\_NUM": 1,

"PRODUCT\_PIC": [

"https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586693622248&di=08fedcc7f0d14d8a5534a9419fea1749&imgtype=0&src=http%3A%2F%2Fimg.jdzj.com%2FUserDocument%2Fwfwangge%2FPicture%2F201110259548.jpg"

],

"PRODUCT\_CLASS": "traffic",

"SELLER\_NAME": "帝王东",

"SELLER\_DORM": "海安A309",

"SELLER\_ID": "5e92e3dcc12cea49871a6270",

"SELLER\_PHONE": "1357924610",

"SELLER\_WECHAT": "wxid\_842yjadwjwt721",

"SELLER\_CLASS": "信计1161",

"SELLER\_SNO": "201611921119",

"RELEASE\_DATE": "2020-04-12 08:09:04",

"DEADLINE": "2021-04-12 08:09:04",

"DESCRIBE": "当年我就是靠这台爱马仕电动车上的学校危险驾驶通报名单第一名，因为学校内没有一台电动车能追得上我这台，而且有了它还能撩妹兜风，我就是靠它把到隔壁班的翠花，校园内任你飚，不怕撞环校车,我撞了好多次车都没事，就是我有点事，可以说车绝对牛逼克拉斯，因要毕业了，所以忍痛割爱卖掉这台浪里小白龙，想要就联系我"

},

{

"\_id": "5e92e53bc12cea49871a62f1",

"PRODUCT\_NAME": "华为P40",

"PRODUCT\_PRICE": "1165",

"PRODUCT\_NUM": 1,

"PRODUCT\_PIC": [

"https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695199935&di=3386c6f8bd13a8452ed43238c116d728&imgtype=0&src=http%3A%2F%2Fdingyue.ws.126.net%2F2020%2F0410%2Fe9a0a0f2j00q8kkgk001nc000hs00adm.jpg",

"https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695199928&di=df3f5f42072f330a82ede78179a0a9a4&imgtype=0&src=http%3A%2F%2Fn.sinaimg.cn%2Fsinacn20190510s%2F492%2Fw1920h972%2F20190510%2F9df1-hwsffzc2076045.jpg",

"https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695199927&di=af7d2965d6d737e700ec811a1b139428&imgtype=0&src=http%3A%2F%2Fdingyue.ws.126.net%2F2020%2F0325%2F4671b4fdj00q7qd9c002ec001jk00rsm.jpg"

],

"PRODUCT\_CLASS": "digital",

"SELLER\_NAME": "帝王东",

"SELLER\_DORM": "海安A309",

"SELLER\_ID": "5e92e3dcc12cea49871a6270",

"SELLER\_PHONE": "1357924610",

"SELLER\_WECHAT": "wxid\_842yjadwjwt721",

"SELLER\_CLASS": "信计1161",

"SELLER\_SNO": "201611921119",

"RELEASE\_DATE": "2020-04-12 08:09:04",

"DEADLINE": "2021-04-12 08:09:04",

"DESCRIBE": "大一时候托哆啦A梦给我买的，因为要发布了，所以现在不想要了，我已经叫多啦A梦给我买华为P100了，这台老式机就给你们吧"

}

]

}

在首页头部，会有个搜索框对商品信息进行搜索，可根据商品名、卖家的用户名、卖家的班级、卖家的宿色地址、商品描述中的关键字查找商品并支持模糊查询，具体的接口代码如下：

（2）根据商品名、用户名、班级、宿色地址、描述中的关键字查找商品

router.post('/search',async (req,res)=>{

    let {

      keywords,

      pagesize,

      pagenum

    } = req.body;

    let searchProductsList;

    if(keywords){

        searchProductsList = await mongo.find(colName,[

            {PRODUCT\_NAME:{$regex:keywords}},

            {SELLER\_NAME:{$regex:keywords}},

            {SELLER\_CLASS:{$regex:keywords}},

            {SELLER\_DORM:{$regex:keywords}},

            {DESCRIBE:{$regex:keywords}}

        ])

    }else{

        if(pagesize&&pagenum){

            searchProductsList = await mongo.find(colName,{},pagenum,pagesize);

        }else{

            searchProductsList = await mongo.find(colName);

            res.send(formatData({data:searchProductsList.length}));

            return;

        }

    }

    if(searchProductsList.length>0){

        res.send(formatData({data:searchProductsList}));

    }else{

        res.send(formatData({data:"没有查到相关商品"}));

    }

})

请求url：/products / search

请求方法：post

参数：

keywords：商品信息查找关键字 ，若没有keywords则根据pagesize和pagenum查找对应的商品信息，都没有则返回所有商品的数量

pagesize：查找的页数

pagenum：一页的数量

返回：

{

"code": 1,

"msg": "success",

"data": [//找到对应商品信息会以数组的形式返回查找到的所有商品信息，找不到会提示没有找到相关商品

{

"\_id": "5e92e1e6c12cea49871a620d",

"PRODUCT\_NAME": "电动车",

"PRODUCT\_PRICE": "309",

"PRODUCT\_NUM": 1,

"PRODUCT\_PIC": [

"https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586693622248&di=08fedcc7f0d14d8a5534a9419fea1749&imgtype=0&src=http%3A%2F%2Fimg.jdzj.com%2FUserDocument%2Fwfwangge%2FPicture%2F201110259548.jpg"

],

"PRODUCT\_CLASS": "traffic",

"SELLER\_NAME": "帝王东",

"SELLER\_DORM": "海安A309",

"SELLER\_ID": "5e92e3dcc12cea49871a6270",

"SELLER\_PHONE": "1357924610",

"SELLER\_WECHAT": "wxid\_842yjadwjwt721",

"SELLER\_CLASS": "信计1161",

"SELLER\_SNO": "201611921119",

"RELEASE\_DATE": "2020-04-12 08:09:04",

"DEADLINE": "2021-04-12 08:09:04",

"DESCRIBE": "当年我就是靠这台爱马仕电动车上的学校危险驾驶通报名单第一名，因为学校内没有一台电动车能追得上我这台，而且有了它还能撩妹兜风，我就是靠它把到隔壁班的翠花，校园内任你飚，不怕撞环校车,我撞了好多次车都没事，就是我有点事，可以说车绝对牛逼克拉斯，因要毕业了，所以忍痛割爱卖掉这台浪里小白龙，想要就联系我"

}

]

}

在分类页面，商品信息将会根据类别进行分类，在分类页面点击类别切换的时候需要根据类别查找类别对应的商品信息，为了更好的用户体验，分类页将提供懒加载功能，避免一次性渲染太多信息导致的页面卡顿不会的用户体验，具体的接口代码如下：

（3）根据类别查找商品

router.get('/type',async (req,res)=>{

    let {

        type,

        pagesize,

        pagenum

    } = req.query;

    let typeProductsList;

    if(type){

        typeProductsList = await mongo.find(colName,{PRODUCT\_CLASS:type},pagenum,pagesize);

    }else{

        typeProductsList = await mongo.find(colName,{},pagenum,pagesize);

    }

    if(typeProductsList.length>0){

        res.send(formatData({data:typeProductsList}));

    }else{

        res.send(formatData());

    }

})

请求url：/products / type

请求方法：get

参数：

type：商品类别，具体的值有digital、sports、books、lives、traffic、dress\_shoes、music、pet、others，可不传，不传则查找所有商品

pagesize：懒加载一次请求的商品数量

pagenum：懒加载的次数

返回：

{

"code": 1,

"msg": "success",

"data": [//以数组的形式返回商品具体信息，类别商品信息为空时会返回空数组

{

"\_id": "5e92e1e6c12cea49871a620d",

"PRODUCT\_NAME": "电动车",

"PRODUCT\_PRICE": "309",

"PRODUCT\_NUM": 1,

"PRODUCT\_PIC": [

"https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586693622248&di=08fedcc7f0d14d8a5534a9419fea1749&imgtype=0&src=http%3A%2F%2Fimg.jdzj.com%2FUserDocument%2Fwfwangge%2FPicture%2F201110259548.jpg"

],

"PRODUCT\_CLASS": "traffic",

"SELLER\_NAME": "帝王东",

"SELLER\_DORM": "海安A309",

"SELLER\_ID": "5e92e3dcc12cea49871a6270",

"SELLER\_PHONE": "1357924610",

"SELLER\_WECHAT": "wxid\_842yjadwjwt721",

"SELLER\_CLASS": "信计1161",

"SELLER\_SNO": "201611921119",

"RELEASE\_DATE": "2020-04-12 08:09:04",

"DEADLINE": "2021-04-12 08:09:04",

"DESCRIBE": "当年我就是靠这台爱马仕电动车上的学校危险驾驶通报名单第一名，因为学校内没有一台电动车能追得上我这台，而且有了它还能撩妹兜风，我就是靠它把到隔壁班的翠花，校园内任你飚，不怕撞环校车,我撞了好多次车都没事，就是我有点事，可以说车绝对牛逼克拉斯，因要毕业了，所以忍痛割爱卖掉这台浪里小白龙，想要就联系我"

}

]

}

在后台管理中心中页码跳转和商品信息总数都需要请求商品的数量，于是就提供了请求商品数量的接口，具体的接口代码如下：

（4）查找商品数量

router.get('/productnum',async (req,res)=>{

    let {

        type

    } = req.query;

    var query = {};

    if(type){//传类型就查找这个类型的商品数量，不穿就查看所有商品的数量

        query.PRODUCT\_CLASS = type;

    }

    try{

        let typeProductsList = await mongo.find(colName,query);

        res.send(formatData({data:typeProductsList.length}));

    }catch{

        res.send(formatData({code:0}));

    }

})

请求url：/products / productnum

请求方法：get

参数：

type：商品类别，具体的值有digital、sports、books、lives、traffic、dress\_shoes、music、pet、others，若传了type就返回对应类别的商品数量，没有传则返回所有商品的数量

返回：

{

"code": 1,

"msg": "success",

"data": 3//返回对应的商品数量

}

用户登录之后可在我的页面修改发布商品信息对自己发布的商品信息进行修改，用户可以修改自己发布的商品的商品名称、商品价格、商品数量、卖家姓名、手机号码、微信号码、班级、商品描述，以及可延长发布日期。用户若在三天之内，不再操作自己发布的商品，商品也会根据时间三天后自动下架，等用户重新操作一下，商品又可以继续上架了。而管理员同样可以利用这个借口修改商品的所有信息。

（5）根据商品\_id修改商品信息

router.post('/update',async (req,res)=>{

    let {

        productid,

        porductname,

        productprice,

        productnum,

        productclass,

        sellername,

        sellerdorm,

        sellerphone,

        sellerwechat,

        sellerclass,

        sellersno,

        deadline,

        describe

    } = req.body;

    var obj = {

        PRODUCT\_NAME:porductname,

        PRODUCT\_PRICE:productprice,

        PRODUCT\_NUM:productnum,

        PRODUCT\_CLASS:productclass,

        SELLER\_NAME:sellername,

        SELLER\_DORM:sellerdorm,

        SELLER\_PHONE:sellerphone,

        SELLER\_WECHAT:sellerwechat,

        SELLER\_CLASS:sellerclass,

        SELLER\_SNO:sellersno,

        DEADLINE:deadline,

        DESCRIBE:describe

    }

    var data = {};

    for(var key in obj){

        if(obj[key]){

            data[key] = obj[key]

        }

    }

    if(data){

        try{

            await mongo.update(colName,{\_id:productid},data)

            res.send(formatData());

        }catch(err){

            res.send(formatData({code:0}));

        }

    }else{

        res.send(formatData({code:0}));

    }

})

请求url：/products / update

请求方法：post

参数：

    productid：商品id，必传，商品信息修改将根据商品id作为key来修改对应商品

   porductname：修改的商品名称，可不传

productprice：修改的商品价格，可不传

   productnum：修改的商品数量，可不传

   productclass：修改的商品类别，可不传

   sellername：修改的卖家姓名，可不传

  sellerdorm：修改的卖家宿舍地址，可不传

   sellerphone：修改的卖家联系电话，可不传

   sellerwechat：修改的卖家微信号码，可不传

   sellerclass：修改的卖家班级，可不传

   sellersno：修改的卖家学号，可不传

   deadline：修改的下架时间，可不传

   describe：修改的商品描述，可不传

返回：

{

"code": 1,//修改成功返回1，失败则返回0

"msg": "success",//修改成功success，失败则提示fail

"data": []

}

用户在发布商品之后，完成交易用户可以自己手动下架商品，避免商品还在上架被买家继续找，然后管理员也可以下架一些违法不良的商品，可以彻底删除商品。接口代码如下：

（6）根据商品\_id删除商品

router.post('/del',async (req,res)=>{

    let {

        productid

    } = req.body;

    try{

        await mongo.remove(colName,{\_id: productid });

        res.send(formatData({data:"删除成功"}))

    }catch(err){

        res.send(formatData({code:0,data:err}))

    }

})

请求url：/products / del'

请求方法：post

参数：

    productid：商品id，必传，商品信息修改将根据商品id作为key来修改对应商品

返回：

{

    "code": 1,//返回1为删除成功，返回0为删除不成功

    "msg": "success",

    "data": "删除成功"//删除成功或失败都会有提示信息

}

在其它商品列表里点击跳转到商品详情页面时，需要根据商品的id去获取商品的详情信息，接口：

（7）根据商品ID获取商品信息

router.post('/info',async (req,res)=>{

    let {

        productid

    } = req.body;

    try{

        let data = await mongo.find(colName,{\_id:ObjectId(productid)});

        res.send(formatData({data:data}))

    }catch(err){

        res.send(formatData({code:0,data:err}))

    }

})

请求url：/products / info'

请求方法：post

参数：

    productid：商品id，必传，商品信息修改将根据商品id作为key来修改对应商品

返回：

{

    "code": 1,

    "msg": "success",

    "data": [//搜索到该商品则会返回有数据的数组，否则返回空数组

        {

            "\_id": "5e92e53bc12cea49871a62f1",

            "PRODUCT\_NAME": "华为P40",

            "PRODUCT\_PRICE": "1165",

            "PRODUCT\_NUM": 1,

            "PRODUCT\_PIC": [

                "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695199935&di=3386c6f8bd13a8452ed43238c116d728&imgtype=0&src=http%3A%2F%2Fdingyue.ws.126.net%2F2020%2F0410%2Fe9a0a0f2j00q8kkgk001nc000hs00adm.jpg",

                "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695199928&di=df3f5f42072f330a82ede78179a0a9a4&imgtype=0&src=http%3A%2F%2Fn.sinaimg.cn%2Fsinacn20190510s%2F492%2Fw1920h972%2F20190510%2F9df1-hwsffzc2076045.jpg",

                "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695199927&di=af7d2965d6d737e700ec811a1b139428&imgtype=0&src=http%3A%2F%2Fdingyue.ws.126.net%2F2020%2F0325%2F4671b4fdj00q7qd9c002ec001jk00rsm.jpg"

            ],

            "PRODUCT\_CLASS": "digital",

            "SELLER\_NAME": "帝王东",

            "SELLER\_DORM": "海安A309",

            "SELLER\_ID": "5e92e3dcc12cea49871a6270",

            "SELLER\_PHONE": "1357924610",

            "SELLER\_WECHAT": "wxid\_842yjadwjwt721",

            "SELLER\_CLASS": "信计1161",

            "SELLER\_SNO": "201611921119",

            "RELEASE\_DATE": "2020-04-12 08:09:04",

            "DEADLINE": "2021-04-12 08:09:04",

            "DESCRIBE": "大一时候托哆啦A梦给我买的，因为要发布了，所以现在不想要了，我已经叫多啦A梦给我买华为P100了，这台老式机就给你们吧"

        }

    ]

}

在发布商品的表单中，多数都是普通的字符串和数字类型的数据，而图片上传是转出formdata类型的数据，这个比较特殊，需上传到服务器上，并返回图片保存的URL地址，具体的接口代码如下：

（8）上传图片

router.post('/upload', multer({

    //设置文件存储路径

    dest: 'static'   //upload文件如果不存在则会自己创建一个。

// eslint-disable-next-line no-unused-vars

}).single('file'), function (req, res, next) {

    if (req.file.length === 0) {  //判断一下文件是否存在，也可以在前端代码中进行判断。

        res.render("error", { message: "上传文件不能为空！" });

        return

    } else {

        let file = req.file;

        let fileInfo = {};

        fs.renameSync('./static/' + file.filename, './static/' + file.originalname);//这里修改文件名字，比较随意。

        // 获取文件信息

        fileInfo.mimetype = file.mimetype;

        fileInfo.originalname = file.originalname;

        fileInfo.size = file.size;

        fileInfo.path = file.path;

        // 设置响应类型及编码

        res.set({

            'content-type': 'application/json; charset=utf-8'

        });

        res.send('http://47.98.245.185:4399/static/' + fileInfo.originalname);

    }

})

登录后的用户可以发布自己的二手商品，发布商品的接口如下：

（9）发布商品

router.post('/release',async (req,res)=>{

    let {

        PRODUCT\_NAME,

        PRODUCT\_PRICE,

        PRODUCT\_NUM,

        PRODUCT\_PIC,

        PRODUCT\_CLASS,

        SELLER\_NAME,

        SELLER\_DORM,

        SELLER\_ID,

        SELLER\_PHONE,

        SELLER\_WECHAT,

        SELLER\_CLASS,

        SELLER\_SNO,

        RELEASE\_DATE,

        DEADLINE,

        DESCRIBE,

    } = req.body;

    if(PRODUCT\_NAME&&PRODUCT\_PRICE&&PRODUCT\_NUM&&PRODUCT\_PIC&&PRODUCT\_CLASS&&SELLER\_NAME){

        try{

           let result = await mongo.create(colName,[{

            PRODUCT\_NAME : PRODUCT\_NAME,

            PRODUCT\_PRICE : PRODUCT\_PRICE,

            PRODUCT\_NUM : PRODUCT\_NUM,

            PRODUCT\_PIC :PRODUCT\_PIC,

            PRODUCT\_CLASS : PRODUCT\_CLASS,

            SELLER\_NAME : SELLER\_NAME,

            SELLER\_DORM : SELLER\_DORM,

            SELLER\_ID : SELLER\_ID,

            SELLER\_PHONE:SELLER\_PHONE,

            SELLER\_WECHAT:SELLER\_WECHAT,

            SELLER\_CLASS:SELLER\_CLASS,

            SELLER\_SNO:SELLER\_SNO,

            RELEASE\_DATE:RELEASE\_DATE,

            DEADLINE:DEADLINE,

            DESCRIBE:DESCRIBE,

             }]);

           res.send(formatData({data:result}));

        }catch(err){

           res.send(formatData({code:0,data:err}))

        }

    }else{

        res.send(formatData({code:0,data:"必要信息不能为空"}))

    }

})

请求url：/products / release'

请求方法：post

参数：

    PRODUCT\_NAME,

        PRODUCT\_PRICE,

        PRODUCT\_NUM,

        PRODUCT\_PIC,

        PRODUCT\_CLASS,

        SELLER\_NAME,

        SELLER\_DORM,

        SELLER\_ID,

        SELLER\_PHONE,

        SELLER\_WECHAT,

        SELLER\_CLASS,

        SELLER\_SNO,

        RELEASE\_DATE,

        DEADLINE,

        DESCRIBE,

返回：

{  
    "code":1,//发布成功返回1，发布失败返回0  
    "msg":"success",  
    "data":{  
        "result":{  
            "ok":1,  
            "n":1  
        },  
        "ops":[  
            {  
                "PRODUCT\_NAME":"爱爱",  
                "PRODUCT\_PRICE":"112",  
                "PRODUCT\_NUM":"1",  
                "PRODUCT\_PIC":[  
                    "[http://47.98.245.185:4399/static/手机拍的图.jpg](http://47.98.245.185:4399/static/%E6%89%8B%E6%9C%BA%E6%8B%8D%E7%9A%84%E5%9B%BE.jpg)"  
                ],  
                "PRODUCT\_CLASS":"others",  
                "SELLER\_NAME":"帝王东",  
                "SELLER\_DORM":"海安A309",  
                "SELLER\_ID":"5e92e3dcc12cea49871a6270",  
                "SELLER\_PHONE":"13579246810",  
                "SELLER\_WECHAT":"wxid\_842yjadwjwt721",  
                "SELLER\_SNO":"201611921119",  
                "RELEASE\_DATE":"2020-05-05 20:31:17",  
                "DEADLINE":"2020-05-05T12:31:04.000Z",  
                "DESCRIBE":"123123123",  
                "\_id":"5eb15ce009564632b8751e0d"  
            }  
        ],  
        "insertedCount":1,  
        "insertedIds":{  
            "0":"5eb15ce009564632b8751e0d"  
        }  
    }  
}

### 5.1.3 轮播图模块的功能接口实现

轮播图作为整个平台商品信息最为显眼的地方，任何用户都可以看得到。轮播图的获取接口原理是先获取slideshow集合里面的商品id，然后根据商品id再去查找products集合里面的商品信息，并将它返回给前端渲染，接口代码如下：

（1）根据productid获取所有轮播图

router.get('/all',async (req,res)=>{

    let slideshowId = await mongo.find(colName);

    slideshowIdArr = slideshowId.map(function(item){

        return {

            \_id:ObjectId(item.PRODUCTSID.trim())

        }

    })

    let slideshowResult = await mongo.find("products",slideshowIdArr);

    if(slideshowResult.length){

        res.send(formatData({data:slideshowResult}));

    }else{

        res.send(formatData({code:0}));

    }

})

请求url：/slideshow/all

请求方法：get

参数：

   无需传参

返回：

{

    "code": 1,//返回为1代表查找成功

    "msg": "success",

    "data": [//存放商品信息的数组

        {

            "\_id": "5e92e1e6c12cea49871a620d",

            "PRODUCT\_NAME": "浩东牌电动车",

            "PRODUCT\_PRICE": "309",

            "PRODUCT\_NUM": 1,

            "PRODUCT\_PIC": [

                "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586693622248&di=08fedcc7f0d14d8a5534a9419fea1749&imgtype=0&src=http%3A%2F%2Fimg.jdzj.com%2FUserDocument%2Fwfwangge%2FPicture%2F201110259548.jpg"

            ],

            "PRODUCT\_CLASS": "traffic",

            "SELLER\_NAME": "帝王东",

            "SELLER\_DORM": "海安A309",

            "SELLER\_ID": "5e92e3dcc12cea49871a6270",

            "SELLER\_PHONE": "1357924610",

            "SELLER\_WECHAT": "wxid\_842yjadwjwt721",

            "SELLER\_CLASS": "信计1161",

            "SELLER\_SNO": "201611921119",

            "RELEASE\_DATE": "2020-04-12 08:09:04",

            "DEADLINE": "2021-04-12 08:09:04",

            "DESCRIBE": "当年我就是靠这台爱马仕电动车上的学校危险驾驶通报名单第一名，因为学校内没有一台电动车能追得上我这台，而且有了它还能撩妹兜风，我就是靠它把到隔壁班的翠花，校园内任你飚，不怕撞环校车,我撞了好多次车都没事，就是我有点事，可以说车绝对牛逼克拉斯，因要毕业了，所以忍痛割爱卖掉这台浪里小白龙，想要就联系我"

        },

        {

            "\_id": "5e92e53bc12cea49871a62f1",

            "PRODUCT\_NAME": "华为P40",

            "PRODUCT\_PRICE": "1165",

            "PRODUCT\_NUM": 1,

            "PRODUCT\_PIC": [

                "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695199935&di=3386c6f8bd13a8452ed43238c116d728&imgtype=0&src=http%3A%2F%2Fdingyue.ws.126.net%2F2020%2F0410%2Fe9a0a0f2j00q8kkgk001nc000hs00adm.jpg",

                "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695199928&di=df3f5f42072f330a82ede78179a0a9a4&imgtype=0&src=http%3A%2F%2Fn.sinaimg.cn%2Fsinacn20190510s%2F492%2Fw1920h972%2F20190510%2F9df1-hwsffzc2076045.jpg",

                "https://timgsa.baidu.com/timg?image&quality=80&size=b9999\_10000&sec=1586695199927&di=af7d2965d6d737e700ec811a1b139428&imgtype=0&src=http%3A%2F%2Fdingyue.ws.126.net%2F2020%2F0325%2F4671b4fdj00q7qd9c002ec001jk00rsm.jpg"

            ],

            "PRODUCT\_CLASS": "digital",

            "SELLER\_NAME": "帝王东",

            "SELLER\_DORM": "海安A309",

            "SELLER\_ID": "5e92e3dcc12cea49871a6270",

            "SELLER\_PHONE": "1357924610",

            "SELLER\_WECHAT": "wxid\_842yjadwjwt721",

            "SELLER\_CLASS": "信计1161",

            "SELLER\_SNO": "201611921119",

            "RELEASE\_DATE": "2020-04-12 08:09:04",

            "DEADLINE": "2021-04-12 08:09:04",

            "DESCRIBE": "大一时候托哆啦A梦给我买的，因为要发布了，所以现在不想要了，我已经叫多啦A梦给我买华为P100了，这台老式机就给你们吧"

        }

    ]

}

轮播图不可能永远是不变的，管理员可以在后台管理系统中去配置轮播图信息，可选择几个需要展示的商品放在轮播图中，于是需要根据商品id去添加商品到轮播图集合中，接口代码如下：

（2）根据productid添加商品到轮播图

router.post('/add',async (req,res)=>{

    let {productid} = req.body;

    try{

       mongo.create(colName,{PRODUCTSID:productid});

       res.send(formatData());

    }catch{

       res.send(formatData({code:0}))

    }

})

请求url：/slideshow/add

请求方法：post

参数：

   Productid,需要添加到轮播图的商品id

返回：

{

    "code": 1,//添加成功返回1，添加失败返回0

    "msg": "success",

    "data": []

}

在用户下架商品之后，slideshow也应该删除掉已下架的商品，管理员也可以通过后台管理平台去控制轮播图并可删除部分轮播图的商品信息，接口代码如下：

（3）根据productid删除轮播图,只能单个删除

router.delete('/del',async (req,res)=>{

    let {productid} = req.body;

    try{

        mongo.remove(colName,{PRODUCTSID:productid})

        res.send(formatData());

    }catch{

        res.send(formatData({code:0}))

    }

})

请求url：/slideshow/del

请求方法：delete

参数：

   Productid，需要删除的商品的id

返回：

{

    "code": 1,//返回1代表删除成功，返回0代表删除失败

    "msg": "success",

    "data": []

}

### 5.1.4管理员模块的功能接口实现

管理员是一个特殊的身份，管理员账户可以登录平台的后台管理系统去进行一些特殊权限才能执行的操作，这个账户非常特殊，所以没有管理员注册的接口，管理员的数量是极少的，一般管理员的添加都是直接通过服务器上的数据库去操作，管理员的删除也是同样的道理，只有服务器的搭建者才能去操作，但是管理员还是需要有登录的后台管理系统的权限认证接口，接口的代码如下：

（1）管理员登录

router.post('/login',async (req,res)=>{

    let {adminname,adminpsw} = req.body;

    let result = await mongo.find(colName,{ADMIN\_NAME:adminname,ADMIN\_PSW:adminpsw});

    if(result.length>0){//有搜到数据的话，代表存在该用户了

        res.send(formatData({data:1}));

    }else{

        res.send(formatData({data:0}));

    }

})

请求url：/admin/login

请求方法：post

参数：

  adminname//管理员的用户名

adminpsw//管理员的登录密码

返回：

{

    "code": 1,//返回1的时候验证通过，允许登录，返回0的时候，验证不通过，登录失败

    "msg": "success",

    "data": 1

}

## 5.2 前端页面功能实现

用户平台的前端页面有首页、商品分类页、商品详情页、用户商品收藏页、个人中心页、用户登录页、用户注册页、用户个人商品管理页面、用户个人商品修改和新增页面、用户个人信息修改页面，共10个页面。用户平台的前端页面会主要根据两种类型的用户进行不同的展示。一种的未登录的游客，这种类型的用户主要看到的只有首页、分类页、商品详情页三种页面，并且其中的部分功能也不能享受到。另一种就是已经登录的用户，登录后的用户可以看到用户平台的所有页面和使用所有的功能。而后台管理系统的页面主要是登录页、用户信息管理页面和商品信息管理页面共三个页面。在进入后台管理系统平台时，页面会根据浏览器的存储查找是否有登录，若没登录将会退出管理系统并跳转到登录页面。

### 5.2.1 首页

本平台支持没有登录的游客在平台上浏览商品信息，所以无论是已登录的用户还是未登录的游客都可以在首页享受到相同的待遇。首页是进入整个平台第一个展示给用户的页面。首页有商品信息的简单浏览，有懒加载的效果，每次加载一定数量的商品，滚动到底部的时候会再请求新的商品信息，直到数据库的商品信息全部加载完毕，其中配合了elementUI的vue组件去实现。具体的代码如下：

export default {

    props:{

        pdSize:Number

    },

    data () {

      return {

        loading:false,

        pdList:[],

        pagenum:1,

        pdNum:999,

        nowTime:new Date().getTime()

      }

    },

    async created(){

        let {data}= await getProductNum();

        this.pdNum = data;

    },

    computed: {

      noMore () {

        return this.pdList.length >= this.pdNum

      },

      disabled () {

        return this.loading || this.noMore

      }

    },

    methods: {

      load () {

        this.loading = true

        setTimeout(async () => {

          let data = await getLazyProductList (this.pdSize,this.pagenum);

          this.pagenum+=1;

          this.pdList=this.pdList.concat(data);

          this.loading = false

        }, 0.1)

      },

      godetail2(val){

        this.$emit("godetail",val);

      },

      compareTime2(val){

        return compareTime(val);

      }

    }

}

首页头部的输入框是跳转商品关键词搜索的页面的入口，点击输入框将跳转到搜索页面，对商品信息进行搜索。跳转的代码是利用了vue的路由router跳转，代码如下：

goSearch(){

      this.$router.push('/search');

    }

首页中间部分还有以轮播图形式展示的商品。无论是轮播图的商品还是列表展示的商品点击都可以跳转对应的商品信息的详情页面。这两个的跳转也是利用了vue的路由实例并进行传参，代码如下：

godetail(val){

            this.$router.push({path:'/productDetail',query:{id:val}})

    }

首页面效果如下：



### 5.2.2 商品分类页面

除了首页，商品分类的页面和首页一样，无论是登录的用户还是未登录的游客都可以访问，在商品分类的页面里不同分类展示不同的商品信息，每个商品信息都具有跳转到对应商品的详情页面功能，而在侧边栏，不同分类的菜单可根据用户的选择去渲染对应分类的商品信息，代码如下：

export default {

  data(){

    return{

      pdTypeList:[],

      typeloading:true,

      loading:false,

      nowTypeName:'数码',

      nowType:"digital",

      nowTypeList:[],

      pagenum:1,

      pdNum:999,

      nodata:false

    }

  },

  async created(){

      let {data} = await getProductType();

      this.pdTypeList = data;

      this.typeloading=false;

      let num = await getProductNum();

      this.pdNum = num.data

  },

  computed: {

      noMore () {

        return this.nowTypeList.length >= this.pdNum

      },

      disabled () {

        return this.loading || this.noMore

      }

  },

  components:{

    navjump

  },

  methods: {

    load () {

        this.loading = true

        setTimeout(async () => {

            let data = await getLazyProductList (15,this.pagenum,this.nowType);

            this.nowTypeList = this.nowTypeList.concat(data);

            this.pagenum+=1;

            if(this.nowTypeList.length<=0||this.nowTypeList.filter(item =>{

              return this.compareTime2(item.DEADLINE);

            }).length<=0){

               this.nodata=true

            }else{

              this.nodata=false

            }

            this.loading = false;

        }, 0.1)

      },

    navLick(item){

       this.nowTypeList = [];

       this.nodata = false;

       this.nowTypeName = item.NAME;

       this.nowType = item.TYPE;

       this.pagenum = 1;

       this.load();

    },

    compareTime2(val){

        return compareTime(val);

      },

    godetail(val){

        this.$router.push({path:'/productDetail',query:{id:val}})

    }

}}

具体的效果图如下：



### 5.2.3 收藏页面

进入收藏页面，初始化页面时会自动查找浏览器的用户登录cookie，根据是否登录来判断是否显示具体的页面信息，代码如下：

 async created(){

    let userId = lookCookie("userId");

    this.userId = userId;

    this.loading = true;

    if(userId){

      let {data} = await usercollect({id:userId});

      window.console.log(data);

      if(data.length>0){

         this.collectList = data;

      }else{

        this.hasCollect = false;

        this.hasCollectText ='赶紧去逛逛吧！您暂时还没有收藏商品'

      }

    }else{

      this.hasCollect = false;

    }

   this.loading = false;

  }

在收藏页面提供有收藏商品的删除功能，这个功能具体是利用更新用户信息的接口去实现的，根据用户的id去查找对应的用户信息，并将新的该用户的收藏信息进行更新，就可以达到删除的功能，具体的代码如下：

delCollect(){

     this.loading = true;

     let checkList = this.checkList;

     let newCollcetList = this.collectList.filter(function(item){

         let isok = true;

         checkList.forEach(function(item2){

           if(item.\_id==item2){

             isok = false;

           }

         })

         if(isok){return item}

     })

     let newCollcetIdList = newCollcetList.map(function(item){

       return item.\_id

     })

     modifyUser({id:this.userId,usercollect:newCollcetIdList});

      this.collectList = newCollcetList;

      this.loading = false;

   }

在收藏页面看到的商品列表点击对应的商品是可以利用vue的路由实例router进行跳转并带商品的id，代码同首页如下：

 godetail(val){

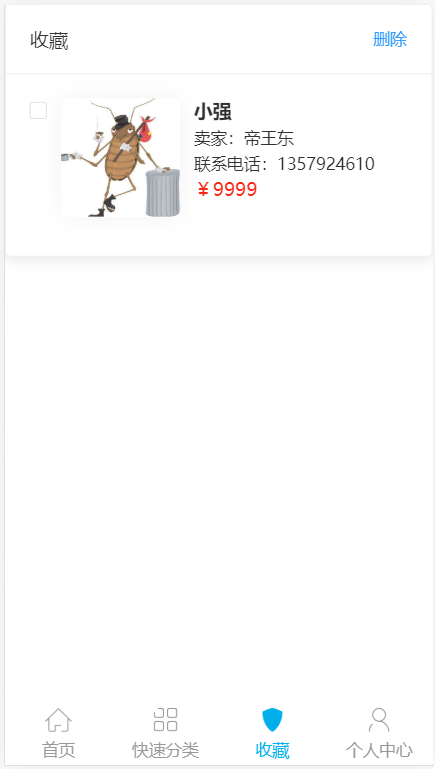
        this.$router.push({path:'/productDetail',query:{id:val}})

    }

收藏页面需要用户登录之后才能使用收藏页面的功能，若是未登录的游客点击收藏菜单跳转到收藏页面，页面效果如下：



用户登录之后，用户可以管理自己收藏的商品，可以删除，也可以点击商品跳转到商品的详情页面，页面的效果图如下：



### 5.2.4 个人中心页面

个人中心页面也需要用户登录之后才能使用个人中心页面的功能，未登录的游客将会展示跳转到用户登录页面的按钮和跳转到用户注册的按钮，页面初始化代码如下：

   <div>

        <div class="head" v-show="isLog">

            <el-avatar :size="70" :src="PicUrl" style="box-shadow: 0 0 8px 0 rgba(0, 0, 0, 0.7);"></el-avatar>

            <div style="margin-top:10px;font-weight:600">{{username}}</div>

        </div>

        <div class="box-body"  v-show="isLog">

           <el-card class="box-card">

                <div v-for="item in navList"  @click="goto(item.click)" :key="item.name" class="text item">

                   <i :class="item.class" style="margin-right:3vw"></i>

                   {{item.name}}

                   <i class="el-icon-arrow-right"></i>

                </div>

            </el-card>

        </div>

        <div class="nolog" v-show="!isLog">

            <img src="../assets/wdl.jpg">

            <div style="margin-bottom:9vh;font-size:1rem;">抱歉~,您还未登录,请选择以下操作</div>

            <el-button type="primary" style="margin-bottom:5vh" @click.native="goto('log')">用户登录</el-button>

            <el-button type="primary" style="margin-left:0" plain @click.native="goto('zhuce')">用户注册</el-button>

        </div>

    </div>

  created(){

        let userId = lookCookie("userId");

        if(userId&&userId!='undefined'){

            this.username = lookCookie("userName")

            this.isLog = true;

        }

    }

未登录看到的页面效果如下：



登录后看到的页面效果如下：



其中的用户登录、注册和退出用的是用一个方法，登录和注册都是利用vue的路由实例进行跳转，而退出是根据isLog 这个字段直接切换页面的标签显示，代码如下：

goto(val){

            if(val == 'logOut'){

                removeCookie('userId');

                this.isLog = false;

            }else{

                this.$router.push('/user/'+val);

            }

        }

### 5.2.5 用户登录页面

用户登录页面主要有两个表单，需输入用户名和密码进行注册，在输入表单的时候会进行非空验证，为空的话是不能点击登录按钮发起登录请求的，验证代码如下：

var checkname = (rule,value,callback)=>{

           if(!value||!value.trim()){

               callback(new Error('用户名不能为空'));

           }else{

               callback();

           }

        };

        var checkpassword = (rule,value,callback)=>{

           if(!value||!value.trim()){

               callback(new Error('密码不能为空'));

           }else{

               callback();

           }

        };

在用户名和密码输入不为空的情况下，用户可以点击登录按钮发起请求，后端接口接受到请求会查找数据库用户的用户名和密码是否匹配正确，若正确则登录成功并跳转到个人中心页面，若不正确则会提示用户输入的用户名和密码可能有误需重新输入，代码如下：

async submitForm () {

           this.loading = true;

           let {username,password} = this.$data.ruleForm;

           let {data} = await userLog({username,password});

           if(data.length>0){

                 setCookie ('userId',data[0].\_id,1);

                 setCookie ('userName',data[0].USER\_NAME,1);

                 this.loading = false;

                 this.$router.push("/user");

           }else{

               this.loading = false;

               this.$notify({

                    title: '登录失败',

                    message: `请检查信息是否填写正确和完整`,

                    type: 'warning'

               })  } }

用户登录页面的效果图如下：



### 5.2.6 用户注册页面

用户注册页面共有8个表单，用户需填写好表单的信息并验证成功后才能点击注册按钮，验证的代码如下：

var checkUsername = (rule, value, callback) => {

        if (!value.trim()) {

          return callback(new Error('用户名不能为空'));

        }

        setTimeout(async () => {

          if (value.length<3||value.length>8) {

            callback(new Error('用户名应为3到8位字符'));

          } else {

              let data =await checkname(value);

              if(data.data===0){

                  callback();

              }else{

                  callback(new Error('用户名已存在，请重新命名'));

              }

            }

        }, 1000);

      };

         var validatePass = (rule, value, callback) => {

        if (value.trim()<6) {

          callback(new Error('密码必须为6位以上'));

        } else {

          if (this.ruleForm.checkPass !== '') {

            this.$refs.ruleForm.validateField('checkPass');

          }

          callback();

        }

      };

         var validatePass2 = (rule, value, callback) => {

        if (value.trim() === '') {

          callback(new Error('密码不能为空哦'));

        } else if (value.trim() !== this.ruleForm.pass) {

          callback(new Error('两次输入密码不一致!'));

        } else {

          callback();

        }

      };

         var checkdorm = (rule,value,callback)=>{//宿舍验证

        if(!(/^[\u4e00-\u9fa5]{2}\S{0,2}\d{3,6}$/.test(value.trim()))){

            callback(new Error('请输入宿舍号,例:海安A309,海滨1栋104'));

        }else{

            callback();

        }

      };

        var checkphone = async (rule,value,callback)=>{//手机号验证

        if(!(/^1[3456789]\d{9}$/.test(value.trim()))){

            callback(new Error('请输入正确的手机号'));

        }else{

            let {data} = await checkuserphone(value);

            if(data==1){

                callback(new Error('该手机号已被注册,若有问题请联系管理员'));

            }else{

              callback();

            }

        }

      };

         var checkgrade = (rule,value,callback)=>{//班级验证

        if(!(/^[\u4e00-\u9fa5]{2,}\d{4}$/.test(value.trim()))){

            callback(new Error("专业+班别,例:'信计1161'"));

        }else{

            callback();

        }

      };

         var checkstudynum =async (rule,value,callback)=>{//学号验证

        if(!(/^20\d{10}$/.test(value.trim()))){

            callback(new Error('请输学号,例如201611921119'));

        }else{

            let data = await checksno(value);

            if(data.data===1){

              callback(new Error('该学号已被人注册,若不是本人操作,请联系管理员'));

            }else{

              callback();

            }

        }

      };

      var checkwechat = async (rule,value,callback)=>{//微信号

        if(!value){

          callback(new Error('请输入微信号'));

        }else if(!value.trim()){

          callback(new Error('微信号不能为空'));

        }else{

            let {data} = await checkuserwechat(value);

            if(data==1){

              callback(new Error('该微信号已被注册,若不是本人操作,请联系管理员'));

            }else{

              callback();

            }

        }

      }

用户通过上面的代码进行验证后就可以点击注册按钮，注册成功后会提示注册成功并跳转到个人中心页面，同时会在浏览器的cookie中存储用户的id和用户名username两个关键信息，若用户注册失败则会出现弹窗并提示用户检查信息是否都填写准确，具体的代码如下：

      submitForm(formName) {

        this.loading = true;

        this.$refs[formName].validate(async (valid) => {

          if (valid) {//表单全正确，valid为true

          let {pass,Username,dorm,phone,grade,studynum,wechat} = this.$data.ruleForm;

           let data = await createuser({pass,Username,dorm,phone,grade,studynum,wechat});

            if(data.code==1){//注册成功

                 this.$notify({

                  title: '恭喜注册成功',

                  message: `欢迎${Username}来到海大二手交易平台`,

                  type: 'success'

                });

                setCookie('userId',data.data.ops[0].\_id,1);

                setCookie('userName',Username,1);

                this.loading = false;

                this.$router.push('/user')

            }else {

            this.$notify({

                  title: 'sorry,注册失败',

                  message: `请检查信息是否填写正确和完整`,

                  type: 'warning'

                });

            return false;

          }

          }

        });

        this.loading = false;

      }

### 5.2.7 个人信息修改页面

本平台提供用户个人信息修改的功能，用户修改个人信息是为了方便一些在大学四年过程中出现更换宿舍、电话、班级等情况，个人信息修改的页面可以在登录之后的个人信息页面有入口进入。个人信息修改页面主要有六个表单，分别是用户名、手机号、微信号、宿舍、班级、姓名。在页面初始化的时候会根据浏览器的cookie值去回填表单的内容，代码如下：

async created(){

        this.loading = true;

        let username = lookCookie('userName');

        let {data} = await userSearch({keywords:username});

        this.ruleForm.Username = data[0].USER\_NAME;

        this.ruleForm.pass = data[0].USER\_PSW;

        this.ruleForm.dorm = data[0].USER\_DORM;

        this.ruleForm.phone = data[0].USER\_PHONE;

        this.ruleForm.grade = data[0].USER\_CLASS;

        this.ruleForm.wechat = data[0].USER\_WECHAT;

        this.loading = false;

}

在用户修改对应的表单的时候，不同的表单会有自己的验证方法，验证代码如下：

 var checkUsername = (rule, value, callback) => {

              if(value){

                  setTimeout(async () => {

                    if (value.length<3||value.length>8) {

                      callback(new Error('用户名应为3到8位字符'));

                    } else {

                        let data =await checkname(value);

                        if(data.data===0){

                            callback();

                        }else{

                            callback(new Error('用户名已存在，请重新命名'));

                        }

                      }

                  }, 1000);

              }else{

          callback();

      }};

       var validatePass = (rule, value, callback) => {

        if (value && value.trim()<6) {

          callback(new Error('密码必须为6位以上'));

        }else{

          callback();

      }};

       var checkdorm = (rule,value,callback)=>{//宿舍验证

       if(value){

           if(!(/^[\u4e00-\u9fa5]{2}\S{0,2}\d{3,6}$/.test(value.trim()))){

               callback(new Error('请输入宿舍号,例:海安A309,海滨1栋104'));

           }else{

               callback();

           }

       }else{

          callback();

      }};

      var checkphone = async (rule,value,callback)=>{//手机号验证

      if(value){

          if(!(/^1[3456789]\d{9}$/.test(value.trim()))){

              callback(new Error('请输入正确的手机号'));

          }else{

              let {data} = await checkuserphone(value);

              if(data==1){

                  callback(new Error('该手机号已被注册,若有问题请联系管理员'));

              }else{

                callback();

              }

          }

      }else{

          callback();

      }};

       var checkgrade = (rule,value,callback)=>{//班级验证

       if(value){

           if(!(/^[\u4e00-\u9fa5]{2,}\d{4}$/.test(value.trim()))){

               callback(new Error("专业+班别,例:'信计1161'"));

           }else{

               callback();

           }

       }else{

          callback();

      } };

       var checkwechat = async (rule,value,callback)=>{//微信号

        if(value){

                let {data} = await checkuserwechat(value);

                if(data==1){

                    callback(new Error('该微信号已被注册,若不是本人操作,请联系管理员'));

                }else{

                    callback();

                }

        }else{

          callback();

      }}

通过上面的表单的方法验证后，点击信息修改按钮才会发起请求，请求接口更改数据库的用户信息，按钮提交的代码如下：

     async submitForm(formName) {

          this.loading = true;

           this.$refs[formName].validate(async (valid) => {

               if(valid){

                   let {pass,Username,dorm,phone,grade,wechat} = this.$data.ruleForm;

                //    window.console.log({pass,Username,dorm,phone,grade,wechat})

                   let id = lookCookie('userId');

                   if(id){

                       let {data} = await modifyUser({

                         id:id,

                         username:Username,

                         userpsw:pass,

                         userdorm:dorm,

                         userphone:phone,

                         userwechat:wechat,

                         userclass:grade,

                       });

                       if(data.modifiedCount>0){

                           this.$notify({

                             title: '修改成功',

                             message: `恭喜你，您的信息修改成功`,

                             type: 'success'

                         }); }

                   }else{

                       this.$notify({

                           title: '请先登录用户',

                           message: `用户未登录`,

                           type: 'warning'

                         });  } }

           });

           this.loading = false; }

个人信息修改页面的效果图如下：



### 5.2.8 个人商品管理页面

个人商品管理页面主要有右上角的按钮和个人发布的商品展示列表中的商品都可以跳转去到商品发布页面。跳转到个人商品发布和修改页面需要判断是点击右上角的商品发布，还是列表中的商品去修改该商品的发布信息，代码如下：

goMUProduct(val){

            if(val){

                this.$router.push(

                    {path:'/user/personalProduct/MUProduct',

                    query:{id:val}

                    })

            }else{

                this.$router.push('/user/personalProduct/MUProduct')

            }

        }

除了右上角的商品发布和列表的商品点击跳转外，列表中的商品信息中带有商品下架按钮，用户可以根据自己的需求下架自己曾发布的商品，代码如下：

 async delPd(id){

            this.loading = true;

            let data = await delProduct(id);

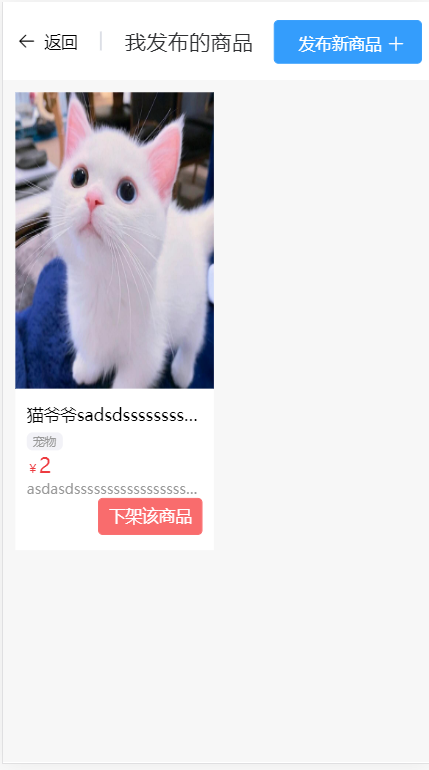
            if(data.code==1){

               this.sellProductList= this.sellProductList.filter((item)=>{ return item.\_id != id; })

                this.$notify({title: '下架成功', message: `您的商品已成功下架`,  type: 'success'});

            }else{  this.$notify({ title: '下架失败',message: data.data, type: 'warn'  });}this.loading =false; }

个人商品管理页面的效果图如下：



### 5.2.9个人商品发布和修改页面

商品发布和修改页面是共用同一个页面，根据跳转到这个页面是否带有商品id进行判断，若是带有商品id则显示修改商品发布的内容并启用修改商品信息的功能并回填商品信息，若没有则显示新发布商品的内容并回填发布商品时需要的用户信息，默认是登录用户的信息，代码如下：

async created(){

        this.loading = true;

        let typedata = await getProductType();

        this.productType = typedata.data;

        if(this.$route.query&&this.$route.query.id){

            this.isfb = false;

          this.productId = this.$route.query.id

          this.title = '修改商品发布信息';

          let {data} = await getProductInfo(this.productId);

          this.ruleForm = Object.assign(this.ruleForm,data[0]);

        //   window.console.log(data);

        }else{

            this.isfb = true;

            this.productId='';

            this.userName = lookCookie("userName");

            this.userId = lookCookie("userId");

            let {data} = await userSearch({keywords:this.userName});

            this.title = '发布新商品';

            this.ruleForm.SELLER\_NAME = data[0].USER\_NAME;

            this.ruleForm.SELLER\_DORM = data[0].USER\_DORM;

            this.ruleForm.SELLER\_CLASS = data[0].USER\_CLASS;

            this.ruleForm.SELLER\_PHONE = data[0].USER\_PHONE;

            this.ruleForm.SELLER\_WECHAT = data[0].USER\_WECHAT;

            this.userSno = data[0].USER\_SNO;

        }

        this.loading = false;

}

其中无论是发布新的商品信息还是修改原来的商品信息，都会进行表单的验证，而验证的函数代码如下：

 var checkSELLER\_NAME = (rule,value,callback) =>{

           if(value){

                  setTimeout(async () => {

                    if (value.length<3||value.length>8) {

                      callback(new Error('用户名应为3到8位字符'));

                    } else {

                        callback();

                    }

                  }, 1000);

              }else{

                    callback(new Error('不能为空'));

                }

        };

        var checkSELLER\_DORM = (rule,value,callback) =>{

           if(value){

                  if(!(/^[\u4e00-\u9fa5]{2}\S{0,2}\d{3,6}$/.test(value.trim()))){

                        callback(new Error('请输入宿舍号,例:海安A309,海滨1栋104'));

                    }else{

                        callback();

                    }

              }else{

                    callback(new Error('不能为空'));

                }

        };

        var checkSELLER\_PHONE = (rule,value,callback) =>{

           if(value){

                if(!(/^1[3456789]\d{9}$/.test(value.trim()))){

                    callback(new Error('请输入正确的手机号'));

                }else{

                    callback();

                }

              }else{

                    callback(new Error('不能为空'));

                }

        };

        var checkSELLER\_CLASS = (rule,value,callback) =>{

           if(value){

                if(!(/^[\u4e00-\u9fa5]{2,}\d{4}$/.test(value.trim()))){

                    callback(new Error("专业+班别,例:'信计1161'"));

                }else{

                    callback();

                }

              }else{

                    callback(new Error('不能为空'));

                }

        };

        var checkSELLER\_WECHAT = (rule,value,callback) =>{

           if(value){

                    callback();

              }else{

                    callback(new Error('不能为空'));

            }

        };

        var checkPRODUCT\_NAME = (rule,value,callback) =>{

           if(value){

                    callback();

              }else{

                    callback(new Error('不能为空'));

            }

        };

        var checkPRODUCT\_NUM = (rule,value,callback) =>{

           if(value>=1){

                    callback();

              }else{

                    callback(new Error('不能小于1'));

            }

        };

        var checkDEADLINE = (rule,value,callback) =>{

           if(value){

                    callback();

              }else{

                    callback(new Error('不能为空'));

            }

        };

        var checkDESCRIBE = (rule,value,callback) =>{

           if(value){

                    callback();

              }else{

                    callback(new Error('不能为空'));

            }

        };

        var checkPRODUCT\_PRICE = (rule,value,callback) =>{

           if(value){

                    callback();

              }else{

                    callback(new Error('不能为空'));

            }

        };

在通过以上的验证之后才能点击商品发布按钮，商品发布的提交需要重新验证一遍表单信息是否符合标准，并发起对应的请求，修改商品信息则发起商品信息修改的请求，发布商品则发起商品信息发布的请求，具体的代码如下：

async submitForm(formName) {

          this.$refs['ruleForm'].validate(async (valid) => {

              if(valid){

              this.loading = true;

               if(formName=='update'){

                   let datas = this.ruleForm;

                   let data = await updateProduct({

                       productid:this.productId,

                        porductname:datas.PRODUCT\_NAME,

                        productprice:datas.PRODUCT\_PRICE,

                        productnum:datas.PRODUCT\_NUM,

                        productclass:datas.PRODUCT\_CLASS,

                        sellername:datas.SELLER\_NAME,

                        sellerdorm:datas.SELLER\_DORM,

                        sellerphone:datas.SELLER\_PHONE,

                        sellerwechat:datas.SELLER\_WECHAT,

                        sellerclass:datas.SELLER\_CLASS,

                        deadline:datas.DEADLINE,

                        describe:datas.DESCRIBE

                   })

                   if(data.code==1){

                       this.$notify({

                        title: '修改成功',

                        message: `商品信息修改成功`,

                        type: 'success'

                        });

                   }else{

                       this.$notify({

                        title: '修改不成功',

                        message: `商品信息修改失败`,

                        type: 'warn'

                        });

                   }

               }else if(formName=='release'){

                   if(this.ruleForm.PRODUCT\_PIC.length<=0){

                       this.$notify({

                        title: '上传商品图片',

                        message: `请上传商品图片,能更好的帮你将商品推荐给其他用户`,

                        type: 'warn'

                        });

                       return;

                   }else{

                       let objfrom = {

                           SELLER\_ID:this.userId,

                           RELEASE\_DATE:dateFormat("YYYY-mm-dd HH:MM:SS", new Date()),

                           SELLER\_SNO:this.userSno

                       }

                       this.ruleForm = Object.assign(this.ruleForm,objfrom);

                       let data = await releaseProduct(this.ruleForm);

                       if(data.code==1){ this.$notify({title: '发布成功',

                            message: `恭喜你,你的商品已发布成功,可在我发布的商品中查看`,

                            type: 'success'

                            });this.goBack();

                       }else{

                           this.$notify({itle: '发布失败', message: `请查看发布信息是否填写完整`,

                            type: 'warn'

                            }); }} }  this.loading = false; } })}

发布新商品的页面效果如下：



已发布的商品信息修改的页面效果如下：



### 5.2.10商品详情页面

商品详情页面是展示商品详细信息的页面，而且从首页、分类页、收藏页的商品信息列表都可以跳转到这个页面，并且这个页面也是用户去详细了解商品的页面。商品详情页主要是渲染和点击收藏按钮进行商品的收藏。

页面初始化的时候，不仅需要根据商品id请求商品详细信息进行渲染还要请求登录的用户是否收藏过当前商品，若收藏了则收藏按钮需显示已收藏状态，未收藏则显示未收藏的状态，具体的代码如下：

页面代码结构：

<div

     v-loading="loading">

        <el-page-header @back="goBack" content="商品详情">

        </el-page-header>

        <div class="lunbo">

             <el-carousel :interval="5000" arrow="always">

                <el-carousel-item v-for="item in productInfo.PRODUCT\_PIC" :key="item">

                  <img :src="item" style="width:100%;height:100%">

                </el-carousel-item>

            </el-carousel>

        </div>

        <div class="info-body">

            <el-card class="box-card">

                <div class="text item">

                   <span class="label"> 商品名称 </span>

                    <span class="text-content pdname">{{productInfo.PRODUCT\_NAME}} </span>

                </div>

                <div class="text item">

                   <span class="label">  商品价格   </span>

                    <span class="text-content pdprice">

                        ￥ <span>{{productInfo.PRODUCT\_PRICE}}</span>

                    </span>

                    <span style="float:right"

                     v-loading="collectLoading"

                      @click="upadteCollect()">

                       <i class="el-icon-star-off" v-if="!isCollect"></i>

                       <i class="el-icon-star-off isCollect" v-else></i>

                       收藏

                    </span>

                </div>

                <div class="text item">

                   <span class="label">商品数量 </span>

                    <span class="text-content">

                        {{productInfo.PRODUCT\_NUM}}

                    </span>

                </div>

                <div class="text item">

                   <span class="label"> 商品类型</span>

                    <span class="text-content">

                        {{translateType2(productInfo.PRODUCT\_CLASS)}}

                    </span>

                </div>

                 <div class="text item">

                   <span class="label">卖家</span>

                    <span class="text-content"> {{productInfo.SELLER\_NAME}} </span>

                </div>

                 <div class="text item">

                   <span class="label">

                        卖家宿舍

                    </span>

                    <span class="text-content"> {{productInfo.SELLER\_DORM}}  </span>

                </div>

                 <div class="text item">

                   <span class="label">  联系电话 </span>

                    <span class="text-content"> {{productInfo.SELLER\_PHONE}} </span>

                </div>

                 <div class="text item">

                   <span class="label">微信 </span>

                    <span class="text-content">{{productInfo.SELLER\_WECHAT}} </span>

                </div>

                 <div class="text item">

                   <span class="label">  班级</span>

                    <span class="text-content"> {{productInfo.SELLER\_CLASS}} </span>

                </div>

                 <div class="text item">

                   <span class="label"> 发布日期

                    </span>

                    <span class="text-content"> {{productInfo.RELEASE\_DATE}}

                    </span>

                </div>

                <div style="height:auto;line-height:50px">

                   <span class="label">  商品描述

                    </span>

                    <div style="font-size:16px;line-height:25px">{{productInfo.DESCRIBE}}

                    </div>

                </div>

            </el-card>

        </div>

    </div>

Js请求商品详情和用户收藏信息进行渲染的代码：

async created(){

        this.loading = true;

        this.productId = this.$route.query.id;

        let {data} =await getProductInfo(this.productId);

        this.productInfo = data[0];

        let username = lookCookie('userName');

        this.userId = lookCookie('userId');

        if(username){

            let data2 = await userSearch({keywords:username});

            this.usercollect = data2.data[0].USER\_COLLECT;

            let pdid = this.productId;

            let iscl = this.isCollect;

            this.usercollect.map(function(item){

                if(item==pdid){

                    iscl = true;

                }})this.isCollect = iscl; } this.loading = false; }

点击收藏按钮的时候需判断用户是否登录，并且根据已登录的用户在数据所保存的商品收藏信息进行增删，是点击收藏还是取消收藏，具体的代码如下：

async upadteCollect(){

            this.collectLoading = true;

           if(this.userId){

                if(this.isCollect){

                  this.usercollect= this.usercollect.filter((item)=>{ return item!=this.productId; })

                }else{this.usercollect.push(this.productId); }

                this.isCollect = !this.isCollect;

                modifyUser({id:this.userId,usercollect:this.usercollect});

           }else{

               this.$notify({title: '不可收藏', message: `请先登录,登陆之后才可以收藏商品哦`,

                    type: 'warning' });

           }this.collectLoading = false; }

# 第六章 系统测试

本次测试的目的主要是为了验证海大二手交易平台的开发结果是否实现了相应的功能，用户的登录、注册、商品信息的发布和修改、用户信息的修改、商品信息的收藏、首页商品信息的展示、商品信息的搜索、分类页面根据类别渲染不同的商品等用户功能。以及管理员的用户信息修改和删除、商品信息的修改和删除等功能。

## 6.1 测试内容

（1）用户测试

为了能让学生用户更方便的测试该平台，我将会把平台的代码放到服务器上提供服务地址给海大学生作为测试用户对平台进行测试，以测试海大二手交易平台的用户的功能，操作性以及用户体验等为主，参与测试对象主要为海大的学生用户。

（2）功能测试

对已开发好的功能一个一个进行测试，测试是否实现了预计结果，需测试的功能主要参考上诉需求分析提出的功能。

（3）性能测试

测试平台发起请求的时间是否过长，cpu的占用率会否过高，内存占用大小等。

## 6.2 测试用例

（1）正确输入用户名和密码，查看是否登录成功

预置条件：用户登录页面

测试输入：用户名“我是卡米”，密码：“123456”；

预期结果：登录成功，跳转到个人中心页面

实际结果：同预期结果一致。

（2）错误输入用户名和密码，查看是否登录成功

预置条件：用户登录页面

测试输入：用户名“123123”，密码：“123456”；

预期结果：登录失败，提示“用户名或密码错误”

实际结果：同预期结果一致。

（3）正确填写用户注册表单，查看是否注册成功

预置条件：用户注册页面

测试输入：用户名“海大小哥”，密码：“123456”，学号：“201611911199”，微信号：“123123”，手机号：“13454565667”，宿舍：“海安a309”，班级：“计数1165”，密码：“123456”；

预期结果：注册成功，提示“恭喜注册成功”

实际结果：同预期结果一致。

（4）错误填写用户注册表单，查看是否注册成功

预置条件：用户注册页面

测试输入：用户名“海大”，密码：“12345”，学号：“20161911199”，微信号：“123123”，手机号：“1345456567”，宿舍：“海安”，班级：“计数65”，密码：“12345”；

预期结果：注册失败，对应的表单会提示对应信息

实际结果：同预期结果一致。

（5）首页懒加载查看商品信息，查看加载是否成功

预置条件：首页

测试输入：将页面滚动到底部进行数据请求并记载渲染；

预期结果：页面滚动到底部，数据记载成功并渲染，直到数据全部请求提示没有更多了

实际结果：同预期结果一致。

（6）首页点击轮播图跳转商品详情页

预置条件：首页

测试输入：点击首页的轮播图跳转到对应的商品详情页；

预期结果：页面成功跳转到对应商品的详情页并展示商品信息

实际结果：同预期结果一致。

（7）首页点击商品列表中的商品跳转商品详情页

预置条件：首页

测试输入：点击首页的商品列表中的商品跳转到对应的商品详情页；

预期结果：页面成功跳转到对应商品的详情页并展示商品信息

实际结果：同预期结果一致。

（8）首页搜索框跳转到搜索页面并进行搜索，查看根据搜索条件是否能成功搜索到对应商品

预置条件：搜索页面

测试输入：点击首页搜索框跳转到搜索页面，在搜索页面输入发布过商品的用户我是卡米的用户名；

预期结果：页面成功跳转到搜索页，并根据我是卡米关键词搜索对应用户的商品或包含我是卡米商品名的商品

实际结果：同预期结果一致。

（9）在搜索页面中点击商品跳转到商品详情页

预置条件：搜索页面

测试输入：点击搜索页面的商品跳转到商品详情页；

预期结果：页面成功跳转到商品详情页，并根据跳转是所带的商品id请求对应的商品详情信息展示。

实际结果：同预期结果一致。

（10）未登录状态去查看收藏

预置条件：收藏页面

测试输入：点击导航跳转到收藏页面；

预期结果：页面根据浏览器的cookie查看用户是否登录，未登录显示登录后同步收藏商品。

实际结果：同预期结果一致。

（11）已登录状态去查看收藏

预置条件：收藏页面

测试输入：在某个商品详情页面点击商品收藏，在收藏页面里可展示，并在收藏页面里点击跳转可跳到详情页；

预期结果：用户在详情页收藏商品时会发起请求，并将收藏的商品录入该用户的数据库信息，在收藏页面请求用户收藏的数据库数据并对商品进行渲染，并可点击渲染的商品跳转到商品详情页。

实际结果：同预期结果一致。

（12）未登录状态去查看个人中心

预置条件：个人中心页面

测试输入：点击底部的导航去到个人中心页面；

预期结果：个人中心页面根据浏览器的cookie查找用户是否登录，用户未登录显示用户未登录的提示并提供登录按钮和注册按钮。

实际结果：同预期结果一致。

（13）已登录状态查看个人中心

预置条件：个人中心页面

测试输入：点击底部导航跳转到个人中心；

预期结果：个人中心页面根据浏览器的cookie查找用户是否登录，用户登录显示用户的用户名在头部和提供修改个人信息、个人商品管理和用户退出的按钮。

实际结果：同预期结果一致。

（14）个人中心点击修改个人信息跳转到修改个人信息页面

预置条件：修改个人信息页面

测试输入：已登录状态在个人中心页点击个人信息修改按钮跳转到个人信息修改页面；

预期结果：个人信息修改页面根据浏览器的cookie判断用户是否登录，并根据用户id请求用户信息回填到个人信息页面表单。

实际结果：同预期结果一致

（15）个人信息页面填写正确规范信息更改个人信息

预置条件：修改个人信息页面

测试输入：按照规范修改用户名、手机号码、微信号码、宿舍、班级、密码并点击确认修改；

预期结果：个人信息修改页面会将表单中新的内容请求到后端接口，接口接受新的信息修改数据个人信息达到更新个人信息的功能。

实际结果：同预期结果一致

（16）个人信息页面填写不规范的信息更改个人信息

预置条件：修改个人信息页面

测试输入：不规范的修改用户名、手机号码、微信号码、宿舍、班级、密码并点击确认修改；

预期结果：不同的信息的表单会在表单下方提示对应的信息。

实际结果：同预期结果一致

（17）跳转到个人商品管理页面

预置条件：个人商品管理页面

测试输入：点击个人中心中的个人商品管理按钮跳转到个人商品管理页面；

预期结果：页面成功跳转，并根据用户id请求数据库用户发布的商品信息进行渲染。

实际结果：同预期结果一致

（18）跳转发布个人商品页面

预置条件：发布个人商品页面

测试输入：在个人商品管理页面点击发布新商品按钮跳转到商品信息表单页面；

预期结果：页面成功跳转，并根据用户id请求数据库用户信息自动回填必要的发布者信息。

。

实际结果：同预期结果一致

（19）正确填写个人商品表单发布商品

预置条件：发布个人商品页面

测试输入：正确填写发布信息并点击发布按钮；

预期结果：提示发布成功并跳转到个人商品管理页面。

实际结果：同预期结果一致

（20）错误填写个人商品表单发布商品

预置条件：发布个人商品页面

测试输入：错误填写发布信息并点击发布按钮；

预期结果：对应的表单会提示不成功的信息。

实际结果：同预期结果一致

（21）跳转个人商品信息修改

预置条件：个人商品信息修改页面

测试输入：点击个人商品管理页面的商品信息跳转到个人商品信息修改页面；

预期结果：个人商品信息修改页面会根据商品id自动回填商品信息和卖家信息。

实际结果：同预期结果一致

（22）按规范修改个人商品信息

预置条件：个人商品信息修改页面

测试输入：按规范修改个人商品信息的商品名称、商品价格、截至时间、商品数量、商品类别、卖家称呼、宿舍、联系方式、微信号、班级、商品描述；

预期结果：对应的商品信息将会更新到数据库，其它用户查看时看到是修改后的商品信息。

实际结果：同预期结果一致

（23）按规范修改个人商品信息

预置条件：个人商品信息修改页面

测试输入：按规范修改个人商品信息的商品名称、商品价格、截至时间、商品数量、商品类别、卖家称呼、宿舍、联系方式、微信号、班级、商品描述；

预期结果：对应的商品信息将会更新到数据库，其它用户查看时看到是修改后的商品信息。

实际结果：同预期结果一致

（24）不按规范修改个人商品信息

预置条件：个人商品信息修改页面

测试输入：不按规范修改个人商品信息的商品名称、商品价格、截至时间、商品数量、商品类别、卖家称呼、宿舍、联系方式、微信号、班级、商品描述；

预期结果：对应的商品信息表单会提示错误。

实际结果：同预期结果一致

（24）已发布的商品在分类页查看

预置条件：分类页

测试输入：商品发布时规定分类，点击分类页面查看对应分类；

预期结果：在对应分类看到刚刚自己在此分类发布的商品。

实际结果：同预期结果一致

（25）收藏商品

预置条件：商品详情页

测试输入：在商品详情页面点击收藏；

预期结果：该商品收藏按钮标亮，并在收藏页面查看到该商品。

实际结果：同预期结果一致

## 6.3测试结果感悟

本次测试有用到浏览器的控制台查看发起的请求所需时间和请求参数，根据请求的时间判断页面性能是否稳定，请求参数则有助于查找请求是否出错。其中所有的功能已基本实现，已基本能满足用户查看商品、收藏商品、登录、注册、修改个人信息、修改商品信息、发布商品等基本功能。而在管理系统中，管理员可进行用户信息修改、用户删除、商品信息修改、商品删除等基本操作。从测试结果来看，并没有特别严重的系统缺陷。不足之处就是页面样式不够规范，部分接口响应的时间稍微过长，页面数据处理逻辑有待优化。

# 第七章 总结

海大二手交易平台的开发从项目设计、项目分析、项目实现、再到测试，整个项目已基本实现了二手交易过程中的商品信息展示、收藏、用户登录、注册、修改信息等一系列功能。前端页面用的是vue框架的单页面应用，所以在数据渲染方面根据diff算法减少页面节点的操作，比起一般的页面渲染具有极好的页面性能，页面的逻辑处理能更快实现，带给用户更好的体验。页面逻辑处理部分主要采用JavaScript语言处理，调用接口的请求是是利用基于ajax异步请求封装的axios组件，因为JavaScript代码的执行是单线程的，异步的请求避免了接口响应所用时间可能带来的阻塞，一定程度上提高了页面的动态交互功能。系统已基本实现了预期的所有功能，实现需求中的主要设计功能，在测试的过程中也是发现不少问题并都解决了，经过测试这个过程，已保证海大二手交易平台上线后能够正常运行。

# 参考文献

[1] 张跃. C2C电子商务平台盈利模式研究. 吉林大学

[2] 王世伟. 校园学生自助交易平台的设计与实现,东北大学

[3]Steve Souders.高性能网站建设指南.电子工业出版社

[4]Nicholas C.Zakas. JavaScript高级程序设计.人民邮电出版社

[5]Michael McMillan 数据结构与算法JavaScript描述 人民邮电出版社.

[6]Richard Blum Linux命令行和shell脚本编程宝典 人民邮电出版社.

[7]BYVoid Node.js开发指南.人民邮电出版社 2012

[8].Alexander Stepanov Paul McJones 编程原本 机械工业出版社