

**本 科 生 毕 业 论 文**

.

题 目：基于Vue的美食购物商城的设计与实现

院 系：计算机学院

专 业：计算机科学与技术

学生姓名：阳孟芳

学 号：201810157

指导教师：林德树（讲师）

二〇二零 年 五 月

**学术诚信声明**

本人所呈交的毕业论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。本毕业论文的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

本人签名：

日期：

摘 要

食达人主要是一款基于在线购物的快速便捷的美食网站，它的核心功能是美食商城的实时定位和支付。在城市切换的过程中，我们可以迅速的定位到自己当前的城市，或者手动切换到理想中的城市，从而进行产品的选购。利用多种便捷的在线支付方式，给我们用户提供了巨大的便利。它具体是由首页、切换城市页、产品列表页、产品详情页、购物车、支付页、订单页七个模块来实现。

该平台使用了当前主流的Vue2.5框架来进行客户端的开发，主要使用了Nuxt.js实现Web网站的页面设计和页面开发。利用来进行数据库的编写。前端技术上主要是使用Nuxt.js和服务端框架Koa2，这两者之间通过SSR做服务端渲染，这样极大的增强了用户视觉上的体验感。同时在数据和状态的同步上我们是选择使用Vuex，这样为我们节省掉过多网络请求，避免了多余的资源浪费。而且开启了高德开放平台的地图服务，室内外一体化地图查看。该程序的设计目的是为用户提供线上美食订购的工作，从而达到减轻用户选择困难，帮助人们大大的节约时间成本，有更多的时间享受我们的生活。

【关键词】：Vue2.5；在线支付；Nuxt.js；MongoDB和Redis；MySQL；Koa2

Abstract

Food Daren is mainly a fast and convenient food website based on online shopping. Its core function is real-time positioning and payment of food mall. In the process of city switching, we can quickly locate our current city, or manually switch to the ideal city, so as to purchase products. Using a variety of convenient online payment methods provides great convenience for our users. It is specifically implemented by seven modules: home page, switch city page, product list page, product details page, shopping cart, payment page, and order page.

The platform uses the current mainstream Vue2.5 framework for client-side development, and mainly uses Nuxt.js to implement web site page design and page development. Use to write the database. The front-end technology mainly uses Nuxt.js and the server-side framework Koa2, which performs server-side rendering through SSR, which greatly enhances the user's visual experience. At the same time, we chose to use Vuex for the synchronization of data and status, which saved us too many network requests and avoided waste of extra resources. And opened the map service of Gaode open platform, indoor and outdoor integrated map viewing. The design purpose of the program is to provide users with online food ordering work, so as to reduce the difficulty of user selection, help people greatly save time costs, and have more time to enjoy our lives.

**[Keyword]:** Vue2.5; online payment; Nuxt.js; MongoDB and Redis; MySQL; Koa2

目 录

1前言

1.1课题研究背景和意义

自人类进入21世纪以来，我国的互联网行业蓬勃发展，5G网络也变得如此流行。在过去的十年里，人类积累的知识已经超过了两千多年的总和。我们世界上的各个信息点已经被互联网连接到一个巨大的信息资源库中，这主要是依靠了互联网技术。我们从从前的线下实体产业，再到今天的线上电商产业。生活方式是变得越来越高效便捷。这种便捷，快速且日趋完善的互动形式已经完全渗透到我们生活的每一个方面。随着全球互联网的飞速发展，特别是移动互联网的，我国的互联网用户数量也呈指数增长。近年来，国家要求“加速和降低费用”，以及宽带技术的发展，这迅速的拉近我互联网与我们普通老百姓的距离。就目前而言上，已经基本实现了村对村、城对城的通信网络，这样便于人们可以随时随地通过Internet和全国各地进行各种信息交流和共享。

网络的崛起势必会带动整个经济的繁荣和进步。因此，一些线上服务行业也就应运而生，这种高效简洁的美食购物商城，势必会成为人们生活中不可或缺的一部分。用户可以随时在线上完成选购，不仅不受任何时间和场所的限制。同时在选购的过程中，我们能够实时的对城市进行定位，这样能够在很大程度上节省我们的时间成本。根据我们在网络上的美食行业的问卷调查中很容易得出一个结论。传统的美食网站信息量过于庞大，用户在提取自己选购的相关信息时干扰过大。因此，我们的出发点就是让用户顺心、放心的进行选购，而解决用户的需求使我们在开发和设计中最核心的要点。这种线上商城的交易在一方面也帮助商家降低了运营的成本和业务流程的复杂性。我们本次基于Vue的美食购物商城的项目开发，不仅是要解决了买家的需求，同时满足了卖家的痛点。

1.2国内发展趋势

国内一些大型的线上餐饮行业受到了人们极大的青睐，网上餐饮行业的普及，确实给我们全世界带来了便捷，而线上餐饮平台要带给人们的已不再是单一的食品供应，而是根据你所在的城市的位置不定期的新品推荐，以及详细的地图进行定位等服务。比如，我们国内比较流行的餐饮平台有饿了么、美团、大众点评等，他们在这一领域都有自己的特色，其中较为显著的几个特点有：用户粘性较强、回购率也较高、覆盖范围比较广。据某知名调查机构报道，在北上广等一线城市他们的市占率已超过70%,其他二线城市市占率已超过64%，这足以体现出我们的市场需求足够大，而有需求也就意味着有市场，也从侧面反映出人们的消费习惯在慢慢的发生转型。

与此同时，这也必将迎来一个巨大的挑战，消费者的不断迭代将带来消费者需求的变更，这会使消费行业变得更加的复杂化，只有不断顺应消费者需求的餐饮行业，才能真正取得用户的信任，才能在这个行业里面脱颖而出，而我们做线上美食选购网站，应该考虑到多方面的因素，除了要参考国内外已有的平台外，也应该具备自己的创新，要具有多元化、健康化、极致化。始终要以消费者的需求为中心。同时，要能够在国内的大趋势之下找到自己的立足之本。

1.3国外发展趋势

随着生活节奏的不断加快，人们没时间也没有耐心在家做饭，其中，国外的人们是更加喜欢外出就餐。这与他们的生活方式密切相关，因此国外虽然也有许多外卖平台，有的平台成立时间也比较早，但是他们的规模都不是很大。比如：Grubhub，美国最大的在线移动外卖平台，从2004年成立至今，经历了若干次投资、并购最终上市，Grubhub目前已经覆盖了美国600多个城市和5.5万家餐厅。日活跃用户在30万左右。Delivery Hero，成立于2011年，Delivery Hero主要在全球50多个高密度城市地区运营自己的外卖服务，该公司与34个国家的20多万家餐馆合作，市场占有率和订单数量都非常高，每个月处理1200万个外卖订单。国外还有许多类似的外卖平台，但是他们与国内的美团和饿了么相比还是差了很多。具体原因主要有三点：

1、劳动力成本，国外的人力，配送员的成本比较高，同样一份快餐自己到店里买可能比较便宜，但如果牵扯到配送服务你至少得多掏一倍的价格。

2、人口密度，我们中国的城市化已经过半，中国城市人口数量已跃居世界第一。中国在商圈、居民区单位面积的人口要远远高于美国等发达国家。在国外的大城市，一大片区域可能也就只有几十户人家，餐厅数量也相对比较少，所以不利于外卖行业的发展。

3、线上支付，由于中国线上支付的全民渗透率已经将近70%，无论是电商、外卖还是滴滴打车，用户都能够非常便捷的完成支付环节，而在国外很多用户还处于最原始的付现金和刷卡环节。

综上所述，国外的发展趋势相比国内来说还是比较缓慢。

1.4小结

2设计方法和所用技术

基于Vue的美食购物商城摈弃了传统的C/S（Client/Server）结构，采用了开发简单，共享性强的B/S（Browser/Server）架构，而前端主要采用了Vue全家桶和最好使用的Vue SSR框架，让我们在开发过程中有条不絮的进行。服务端主要使用Node.js中比较精简的框架Koa2。同时使用VScode作为项目的开发工具，在数据库上我们采用了非关系型数据库（MongoDB）、数据缓存（Redis）。在UI设计上我们主要使用了简洁大方的ElementUI框架。

## 2.1 本系统环境参数

2.1.1 本系统运行环境

|  |  |
| --- | --- |
| 系统配置类型 | 描述 |
| 操作系统 | Windows10 64位 |
| 运行浏览器 | Chrome浏览器 |

2.1.2 本系统开发环境

|  |  |
| --- | --- |
| 环境配置名 | 版本描述 |
| VScode | 1.44.2(user setup) |
| Robo 3T | Version 3 |
| Nuxt.js | 2.0.0 |
| Vue.js | 2.5 |
| Node.js | v10.16.3 |
| Koa2 | 2.6.2 |
| MongoDB | 5.9.6 |
| Redis | 4.0.1 |
| ElementUI | 2.4.11 |

2.2 Vue2.5框架

Vue.js是用于构建用户界面的渐进框架。 与其他重量级框架不同，Vue使用自下而上的增量开发设计。 Vue的核心库仅关注视图层，并且非常容易学习并与其他库或现有项目集成。 另一方面，Vue完全有能力驱动使用Vue生态系统支持的单文件组件和库开发的复杂单页应用程序，Vue的学习曲线比较平缓，适合初学者入门学习。

2.3 Nuxt.js

Nuxt.js 基于一个强大的模块化架构。你可以从 50 多个模块中进行选择，让你的开发变得更快、更简单。对 PWA 的支持、添加谷歌分析到你的网页或生成网站地图，这些功能都无需重新发明轮子来获得。Nuxt.js 默认优化你的应用程序。 我们尽可能地利用 Vue.js 和 Node.js 的最佳实践来构建高性能的应用程序。Nuxt 帮你把所有不需要的比特都从你的应用程序中剔除，并且还包含了一组分析器，以便更好地优化你的应用程序。

2.4 Koa2

Koa 是一个新的 web 框架，由 Express 幕后的原班人马打造， 致力于成为 web 应用和 API 开发领域中的一个更小、更富有表现力、更健壮的基石。 通过利用 async 函数，Koa 帮你丢弃回调函数，并有力地增强错误处理。 Koa 并没有捆绑任何中间件， 而是提供了一套优雅的方法，帮助您快速而愉快地编写服务端应用程序。

2.5 MongoDB

MongoDB是一个介于[关系数据库](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E6%95%B0%E6%8D%AE%E5%BA%93" \t "_blank)和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。它支持的数据结构非常松散，是类似[json](https://baike.baidu.com/item/json)的[bson](https://baike.baidu.com/item/bson" \t "_blank)格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是它支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立[索引](https://baike.baidu.com/item/%E7%B4%A2%E5%BC%95)。

2.6 Redis

Redis是一个键值存储系统。 它类似于Memcached，但是Redis支持的数据类型相对丰富，包括字符串，列表，集合，有序集合（zset）和哈希。 这些数据类型均支持推/弹出和其他操作。Redis 是一个高性能的键值数据库。 redis的出现，很大程度补偿了[memcached](https://baike.baidu.com/item/memcached" \t "_blank)这类key/value存储的不足，在部分场合可以对关系数据库起到很好的补充作用。

2.7 ElementUI

ElementUI是使用Vue.js 2.5作为基本框架实现的一组组件库。 它是一组专门为开发人员、设计人员，以及产品经理准备的组件库，它不仅为用户提供了大量的设计资源，并且帮助网站快速进行原型制作。它包含的组件有：单/复选框，面包屑、按钮，Rate评分，Dialog 对话框，消息提示，Form表单，滚动条和其他组件等。 这些组件使用方便，且复用性极高，在项目开发中很大程度的提高了团队的开发效率。

2.8小结

3需求分析

本次课题设计是一款基于Vue的美食购物商城，他最主要的目的是为我们的生活提供便捷服务，用户可以通过地图服务、热门推荐等功能迅速的解决自己的诉求。而我们的需求分析模块主要是由可行性分析、功能描述，以及参考者用例三部分来组成。整个美食购物商城交易系统主要分为两种状态，分别是游客和普通会员。

3.1 可行性分析

3.1.1 经济可行性

在资金方面是完全充裕的，在开发后需要投入的成本，主要体现在网站的维护和检测、系统开发、广告的营销和线上线下的推广，以及后期服务器所需要消耗的成本等。而投资回报可以通过商家在后期运营的过程中，接一些第三方的推荐广告，或者在影响力扩大之后，可以采用众多商家合作精心合作，从而达到获取利润的目的。

3.1.2 技术可行性

本系统选用的开发平台是VScode和SublimeText 3，同时采用ElementUI 2.4.7的框架，开发语言使用的是JavaScript、HTML和CSS。数据库采用的是MongoDB和Redis，采用这两种数据库合理的优化了项目中的一些弊端，MongoDB在操作上非常的简单，便于我们快速开发，而Redis可以快速读写数据库，特别适合在http没有状态的情况下， Redis通过服务器的session，能够知道下次访问的是哪一个session。在页面渲染上，我们采用了SSR和Vuex两种方式来进行，SSR的方式能够提高我们用户视觉上的体验感，采用Vuex的方式能够减少Ajax请求，最大化的节省我们的网络资源。另外配合Chrome浏览器进行开发，能够让我们的项目大大的提高效率。因此从技术的角度去考虑，本次系统的开发是可行的。

3.2 系统功能描述

3.2.1 游客部分

该系统的用户角色分为会员和游客，游客的功能有以下几点：

（1）注册成为会员

如果游客需要将身份注册为会员需要填写用户名、邮箱地址以及用户的密码。

（2）浏览美食购物商城的信息

成为普通会员后可以在美食购物商城的首页浏览商品展示的相关信息，同时可以进行不同城市的切换，以及登录注册等相关操作。

（3）搜索美食购物商城的信息

游客在首页的时候可以在搜索框里面输入对应的关键字，进行快速的店铺定位，也可以根据热门推荐进行快速搜索，同时也具备商品的价格和评论的快速排序等。

3.2.2 会员部分

该系统的用户角色分为会员和游客，会员的功能有以下几点：

（1）注册

如果还未注册是游客的状态，需要在网站的注册系统页面注册成为会员，同时需要填写对应的用户名、邮箱地址以及用户的密码。

（2）登录

如果已经注册，用户可以根据自己已注册好的用户名和密码在网站的登录系统中进行登录。

（3）找回密码

用户在的登录的过程中，如果忘记自己的密码，可以选择在使用用户名和邮箱地址来进行密码的充值从操作。

（4）搜索美食购物商城的信息

游客在首页的时候可以在搜索框里面输入对应的关键字，进行快速的店铺定位，也可以根据热门推荐进行快速搜索，同时也具备商品的价格和评论的快速排序等。

（5）切换城市管理

用户打开美食购物商城后，将会自动定位到当前的所在城市，若想要浏览其他城市的美食信息。可以通过切换城市来实现，也可以在切换城市页通过城市的首字母或者远程搜索来快速的定位。

（6）商品列表管理

在搜索到对应的关键词之后，将进入商品列表页，同时展示出对应关键字的相关推荐，当我们在滚动商品列表的时候，右侧的地图定位将会自动移动到店铺对应的位置。

（7）购物车管理

在用户登录的情况下，可以快速的讲自己想要的商品添加进购物车，也可以对购物车的数量进行删减。

（8）支付管理

当用户在购物车添加好自己想购买的商品之后，即可提交订单，在订单页面我们有超时管理，如果在规定的时间内未完成支付，订单将会自动取消，同时订单支付页面支持支付宝扫描二维码的方式进行在线交易。

（9）订单管理

当用户提交了自己的支付订单或者未支付的状态下，订单列表都会显示出订单当前的状态，可以通过分页按钮来查看订单的详细信息，也可以在订单列表删除相应的商品。

（10）退出

用户已完成商品的选购或者没有找到合适的商品，可以通过顶部的退出选项，退出当前的登录状态，同时自动跳回到首页。

3.3 参与者用例图

游客用例图如图3-1所示。

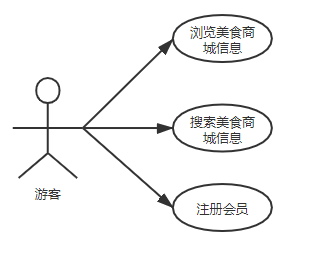


图3-1 游客用例图

会员用例图如图3-2所示

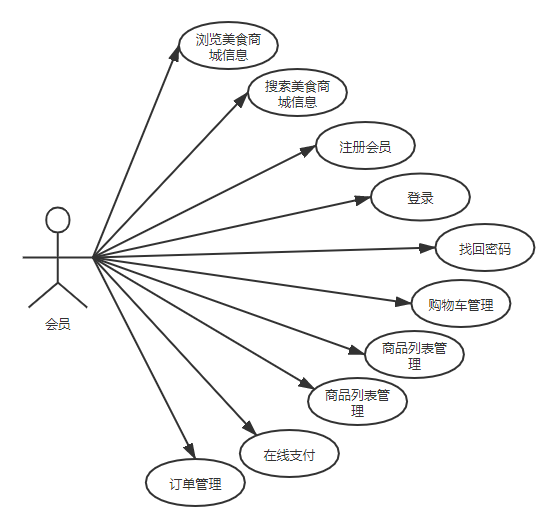


图3-2 会员用例图

3.3 小结

4.系统概要设计

4.1 系统功能概览

食达人是一款以用户为导向的美食购物商城，在整个设计和开发中，我们均把用户的诉求放在第一位，尽量让用户的操作变得简单，在功能方面我们也尽可能的完善，最终将系统划分为七个模块:首页、切换城市页、产品列表页、产品详情页、购物车，订单支付页，订单页。如图4-1所示。

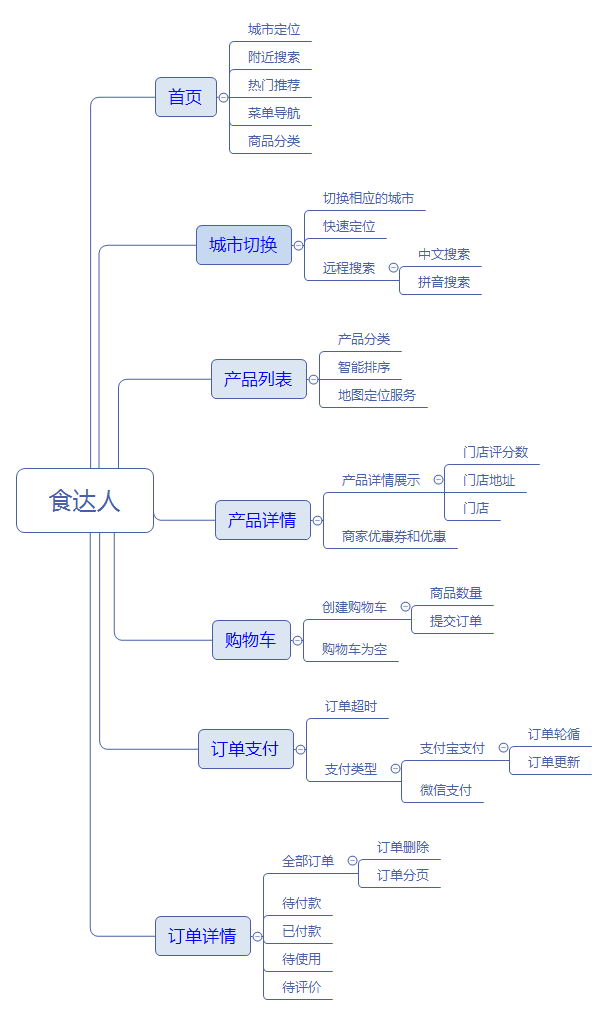


图4-1 思维导图

（1）首页：主要包括附近美食的搜索，以及根据当季的美食进行精准推荐，也可以通过商品的分类，快速的找到商品的类别，同时在商品的列表中能够快速的查看当下火爆的娱乐推荐，。

（2）切换城市：主要功能都是聚集在切换城市上，在这个页面提供了三种快速切换的方式，其一，通过二级联动来快速的切换，其二，通过远程搜索向服务端发出请求，从而达到切换的目的，其三，通过点击城市的首字母来进行实时定位。

（3）产品列表：主要分为四个模块，第一个模块：使用面包屑组件能够显示你当前的城市加对应平台，以及当前的城市加上你搜索的关键字。第二个模块：通过商品的分类和区域的筛选，能够得到所在区域的商品展示。第三个模块是商品列表的价格、人气、评论的排序。第四个模块主要是地图服务，我们可以在滚动列表的时候看到当前产品的店铺定位，也可以通过放大功能缩放地图。

（4）产品详情：主要是包括商品详细信息的展示和商家的团购和优惠，在团购和优惠模块做了用户拦截，如果用户没有进行登录，将无法查看到对应的美食条目，也就意味着不能进行购买。

（5）购物车：购物车是默认为空的状态，当我们在点击购买之前，购物车是不存在的，只有当我们提交了商品，购物车才开始创建。我们能够对商品的数量进行添加和删除操作。

（6）订单支付：主要由三部分组成，用户订单超时处理，以及购买的商品的基本信息和在线支付功能。

（7）订单详情：订单详情主要是对订单的状态进行查看，主要分为：全部订单、待付款和已付款、待使用、待评价五种状态，以及对订单进行更新和删除等操作。

4.1.1游客操作流程图

游客操作流程图如图4-2所示

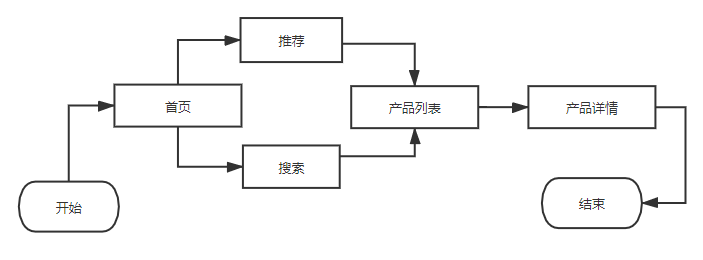


图4-2 游客操作流程图

4.1.2会员操作流程图

会员操作流程图如图4-3所示

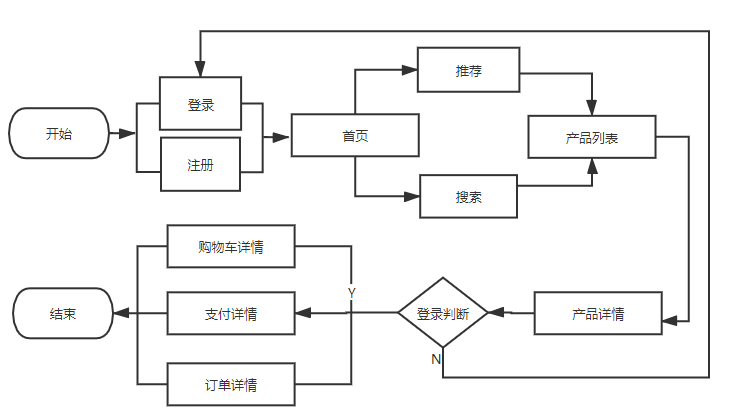


图4-3 用户操作流程图

4.2 数据库设计

4.2.1 数据库概念结构设计

在概念模式设计中，E-R模型法是最常用的设计方法。结合需求分析本系统一共建立了12张表，分别是users（用户表）、topsearchs（搜索信息表）、menus（菜单分类信息表）、cities（切换城市信息表）、category（列表分类信息表）、pois（首页搜索信息表）、province（切换省份信息表）、areas（切换区县信息表）、regions（列表区域信息表）、maps（列表地图信息表）、carts（购物车表）、orders（订单表）。

（1）用户信息实体类包括编码、用户名、密码、邮箱地址、头像、创建时间、性别。用户信息实体图如图4-4所示。

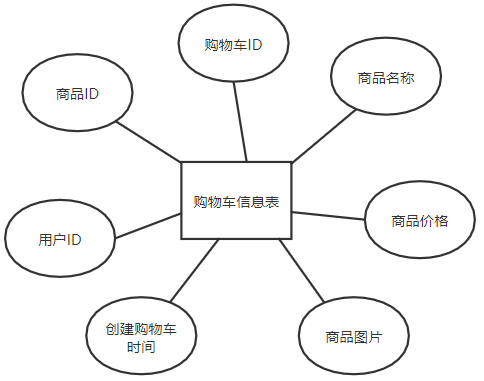


图4-4 用户信息实体图

（2）搜索信息表实体类包括商品ID、商品名称。搜索信息实体图如图4-5所示。

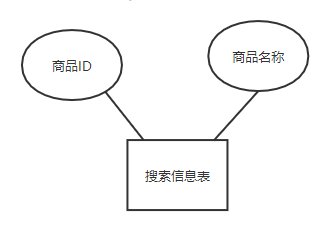


图4-5搜索信息实体图

（3）菜单分类信息实体类包括商品名称、商品类型、类型名称、类型列表。菜单分类信息实体图如图4-6所示。

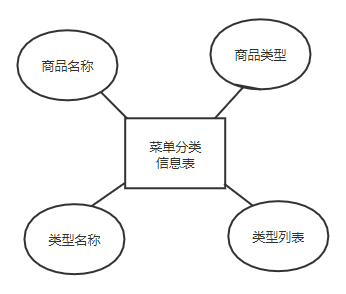


图4-6菜单分类信息实体图

（4）切换城市信息实体类包括省份编号、城市名称、省份名称、城市编号。切换城市信息实体图如图4-7所示。

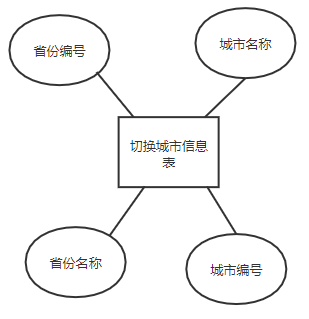


图4-7切换城市信息实体图

（5）列表分类信息实体类包括商品类型、商品类别、商品所在区县、商品所在城市。列表分类信息实体图如图4-8所示。

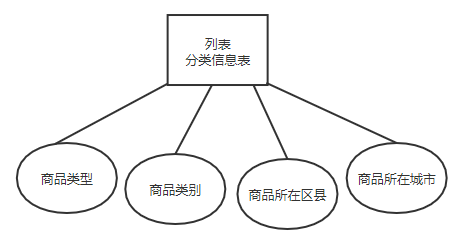


图4-8列表分类信息实体图

（6）首页搜索信息实体类包括商品ID、商品名称、省份名、城市、区县、区号、电话、商圈、地址、类型、经度、纬度。首页搜索信息实体图如图4-9所示。

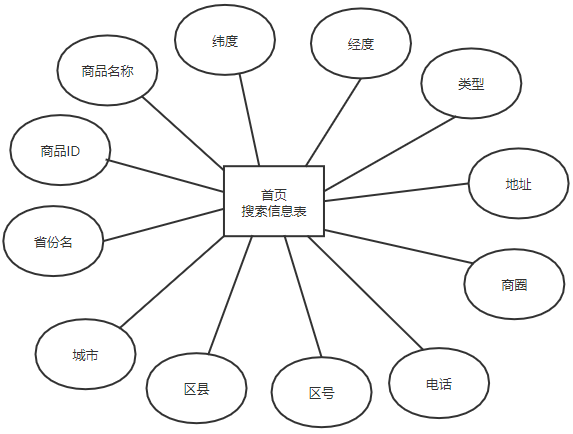


图4-9首页搜索信息实体图

（7）切换省份信息实体类包括省份编码、省份名称。切换省份信息实体图如图4-10所示。

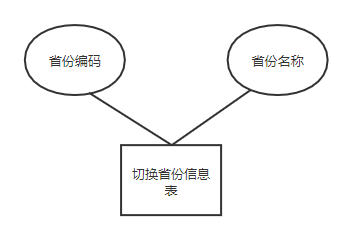


图4-10切换省份信息实体图

（8）切换区县信息实体类包括区域编号、区域名称、城市编号、城市名称。切换区县信息实体图如图4-11所示。

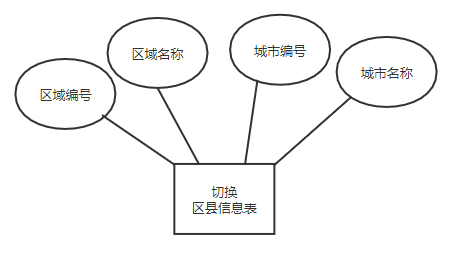


图4-11切换区县信息实体图

（9）商品列表区域信息实体类包括商品ID、商品名称、城市编码、商品所在城市。商品列表区域信息实体图如图4-12所示。

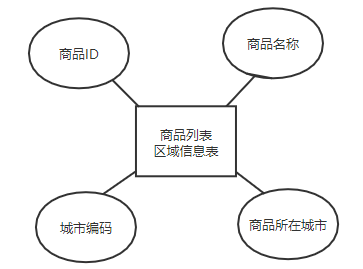


图4-12商品列表区域信息实体图

（10）列表地图信息实体类包括地图ID、城市列表、城市名称、城市ID、地图经度、地图纬度。列表地图信息实体图如图4-13所示。

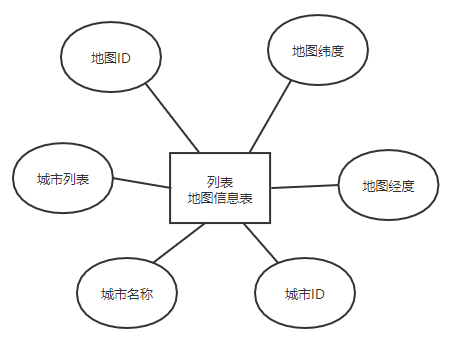


图4-13列表地图信息实体图

（11）购物车信息实体类包括商品ID、用户ID、创建购物车时间、商品图片、商品价格、商品名称、购物车ID。购物车信息实体图如图4-14所示。

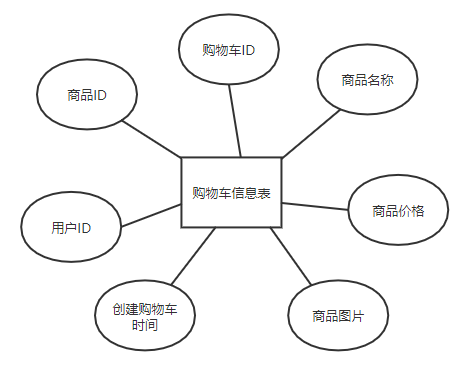


图4-14购物车信息实体图

（12）订单信息实体类包括编号、用户ID、订单编号、商品图片、商品数量、商品名称、商品价格、商品总金额、支付状态、订单创建时间、支付类型。订单信息实体图如图4-15所示。

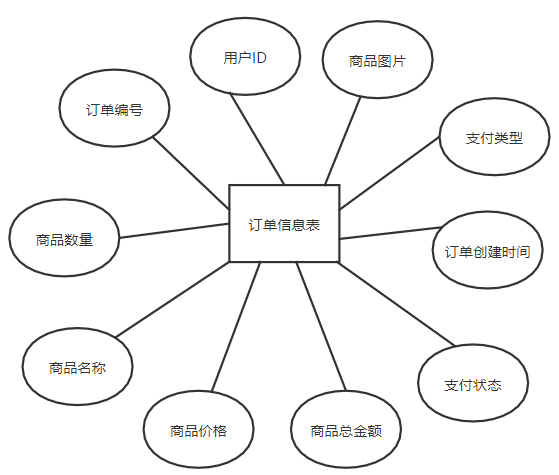


图4-15订单信息实体图

4.2.2 数据库逻辑结构设计

概念结构设计中的E-R图可以转换成关系模式，具体关系模式如下所示：

（1）用户（用户编号、用户名、密码、邮箱地址、头像、创建时间、性别）

（2）搜索（商品ID、商品名称）

（3）菜单分类（商品名称、商品类型、类型名称、类型列表）

（4）切换城市（省份编号、省份名称、城市名称、城市编号）

（5）列表分类（商品类型、商品类别、商品所在区县、商品所在城市）

（6）首页搜索（商品ID、商品名称、省份名、城市、区县、区号、电话、商圈、地址、类型、经度、纬度）

（7）切换省份（省份编码、省份名称）

（8）切换区县（区域编号、区域名称）

（9）城市（城市编号、城市名称、省份编号）

（10）列表区域（商品ID、商品名称、城市编码、商品所在城市）

（11）列表地图（地图ID、城市列表、城市名称、城市ID、地图纬度、地图经度）

（12）订单（订单编号、用户ID、商品数量、商品名称、商品价格、商品总金额、商品图片、支付状态、订单创建时间、支付类型）

（13）购物车（商品ID、用户ID、购物车ID、商品名称、商品价格、商品图片、创建购物车时间）

4.2.3 数据库数据表设计

数据项和数据结构如下：

（1）users表，用来存储用户账号信息。

表4-1 users用户信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据类型 | 长度 | 是否主键 | 是否允许空 |
| user\_id | 用户编号 | bigint | 20 | 是 | 否 |
| username | 用户名 | varchar | 200 | 否 | 是 |
| password | 密码 | varchar | 200 | 否 | 是 |
| email | 手机号 | varchar | 20 | 否 | 是 |
| create\_time  img  sex | 创建时间  头像  性别 | datetime  varchar  int | 0  300  1 | 否  否  否 | 是  是  是 |

（2）topsearchs表，用来存储商品搜索信息。

表4-2 topsearchs商品搜索信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据类型 | 长度 | 是否主键 | 是否允许空 |
| product\_id | 商品ID | int | 20 | 是 | 否 |
| name | 商品名称 | varchar | 200 | 否 | 是 |

（3）menus表，用来存储菜单分类信息。

表4-3 menus菜单分类信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据类型 | 长度 | 是否主键 | 是否允许空 |
| name | 商品名称 | varchar | 200 | 是 | 否 |
| type\_name | 类型名称 | varchar | 200 | 否 | 是 |
| type\_list | 类型列表 | varchar | 200 | 否 | 是 |
| product\_list | 商品类型 | varchar | 200 | 否 | 是 |

（4）cities表，用来存储切换城市信息。

表4-4 cities切换城市信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据类型 | 长度 | 是否主键 | 是否允许空 |
| \_id | 省份编号 | int | 20 | 是 | 否 |
| province\_name | 省份名称 | varchar | 200 | 否 | 是 |
| city\_id | 城市编号 | int | 20 | 否 | 是 |
| city\_name | 城市名称 | varchar | 200 | 否 | 是 |

（5）category表，用来存储列表分类信息。

表4-5 category列表分类信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据类型 | 长度 | 是否主键 | 是否允许空 |
| product\_type | 商品类型 | varchar | 200 | 是 | 否 |
| product\_category | 商品类别 | varchar | 200 | 否 | 是 |
| product\_city | 所在城市 | varchar | 200 | 否 | 是 |
| product\_area | 所在区县 | varchar | 1 | 否 | 是 |

（6）pois表，用来存储首页搜索信息。

表4-6 pois首页搜索信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据类型 | 长度 | 是否主键 | 是否允许空 |
| product\_id | 商品ID | Int | 20 | 是 | 否 |
| province | 省份名 | varchar | 200 | 否 | 是 |
| city | 城市 | varchar | 200 | 否 | 是 |
| county | 区县 | varchar | 200 | 否 | 是 |
| areaCode | 区号 | int | 20 | 否 | 是 |
| tel  area  addr  type  longitude  latitude | 电话  商圈  地址  类型  经度  纬度 | varchar  varchar  varchar  varchar  double  double | 11  200  200  200  8  8 | 否  否  否  否  否  否 | 是  是  是  是  是  是 |

（7）province表，用来存储切换省份信息。

表4-7 province订单详情信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据类型 | 长度 | 是否主键 | 是否允许空 |
| province\_id | 省份编码 | int | 20 | 是 | 否 |
| Province\_name | 省份名称 | varchar | 200 | 否 | 是 |

（8）areas表，用来存储切换区县信息。

表4-8 areas切换区县信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据类型 | 长度 | 是否主键 | 是否允许空 |
| areas\_id | 区域编号 | int | 20 | 是 | 否 |
| areas\_name | 区域名称 | varchar | 50 | 否 | 否 |
| city\_id  city\_name | 城市编号  城市名称 | int  varchar | 20  50 | 否  否 | 否  否 |

（9）regions表，用来存储列表分类信息。

表4-9 regions列表分类信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据类型 | 长度 | 是否主键 | 是否允许空 |
| product\_id | 城市编号 | varchar | 20 | 是 | 否 |
| city\_id  city\_name | 城市名称  省份编号 | varchar  varchar | 50  20 | 否  否 | 否  否 |
| product\_position | 所在城市 | varchar | 200 | 否 | 否 |

（10）maps表，用来存储列表地图信息。

表4-10 maps列表地图信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据类型 | 长度 | 是否主键 | 是否允许空 |
| map\_id | 地图ID | int | 20 | 是 | 否 |
| city\_list | 城市列表 | varchar | 50 | 否 | 否 |
| city\_name | 城市名称 | varchar | 20 | 否 | 否 |
| city\_id  map\_longitude  map\_latitude | 城市ID  地图经度  地图纬度 | int  double  double | 20  8  8 | 否  否  否 | 否  否  否 |

（11）orders表，用来存储用户订单信息。

表4-11 orders用户订单信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据类型 | 长度 | 是否主键 | 是否允许空 |
| order\_id | 订单编号 | int | 11 | 是 | 否 |
| user\_id | 用户ID | int | 11 | 否 | 是 |
| product\_count | 商品数量 | int | 20 | 否 | 是 |
| product\_name | 商品名称 | varchar | 20 | 否 | 是 |
| product\_price | 商品价格 | double | 8 | 否 | 是 |
| product\_imgs | 商品图片 | varchar | 20 | 否 | 是 |
| product\_total  status  create\_time  type | 商品总金额  支付状态  订单创建时间  支付类型 | double  char  datetime  char | 8  1  0  1 | 否  否  否  否 | 是  是  是  是 |

（12）carts表，用来存储购物车信息。

表4-12 carts购物车信息表

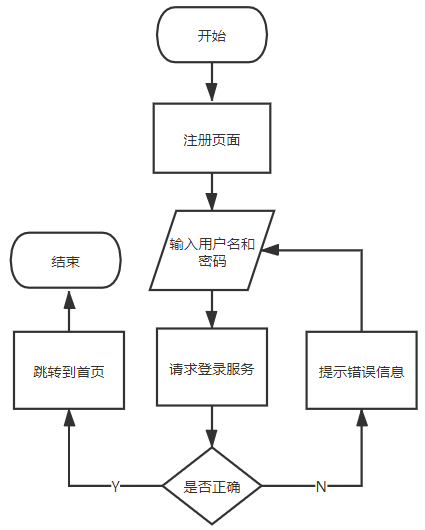
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 字段说明 | 数据类型 | 长度 | 是否主键 | 是否允许空 |
| product\_id | 商品ID | int | 11 | 是 | 否 |
| user\_id | 用户ID | int | 11 | 否 | 是 |
| cart\_time | 创建时间 | datetime | 0 | 否 | 是 |
| product\_imgs | 商品图片 | varchar | 0 | 否 | 是 |
| product\_price | 商品价格 | double | 8 | 否 | 是 |
| product\_name  cart\_id | 商品名称  购物车ID | varchar  int | 200  20 | 否  否 | 是  是 |

4.3 小结

5 详细设计

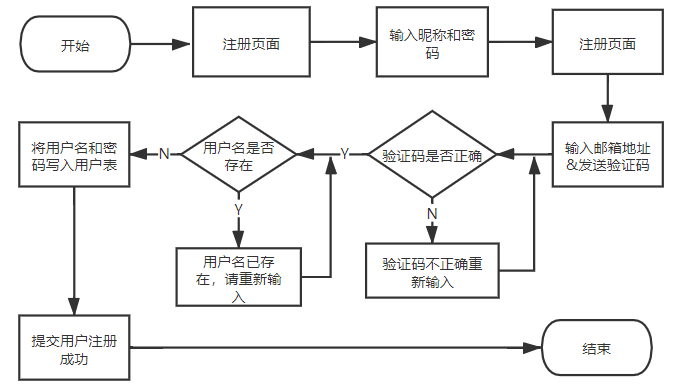
5.1 用户注册流程

游客在没有注册本商城的账号时，能够浏览首页的所有商品，同时能够进行城市切换，也可以搜索商品，若游客在商品详情页需要进行购买时，会有一个用户拦截的操作，这里我们将使用isAuthenticated来检测用户的登录状态。如果没有登录，将会进入登录页面，并且从这里进入到注册页面，整个注册过程中，用户需要输入自己的昵称，密码、邮箱地址，同时点击发送验证码，就会开启QQ邮箱的SMTP服务，即时我们的邮箱就会收到一条邮件，需要将邮箱中发送的验证码填回到注册页面去。在填写的过程中我们前端也简单的处理了在输入框失焦时，做一层简单的校验，例如邮箱地址是否合法，两次密码是否填写一致等。在完成一系列的操作之后，即可点击注册按钮，然后调用服务端的注册接口，此时服务端会进行检测用户名是否已被注册，邮箱验证码是否匹配，并返回到相应的提示信息，若成功注册，将会自动跳转到登录界面。用户注册流程图如图5-1所示。



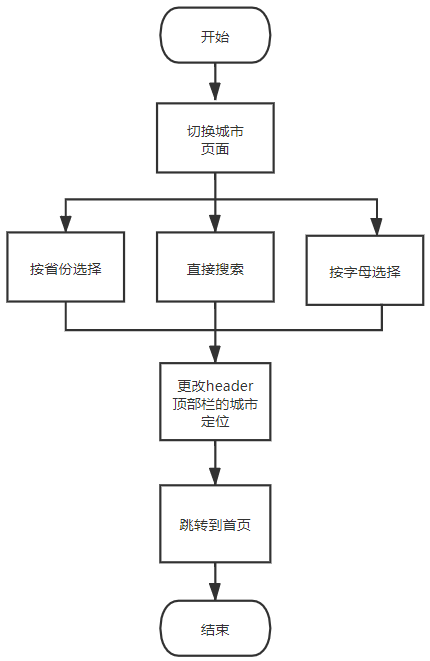
5.2 用户登录流程

若已经注册成为了食达人的用户，就可以在食达人的登录页面进行登录，登录页面主要会检测用户名和密码的合法性，然后会调用服务端的登录接口，此时我们的服务会去用户表(users)里面拉取该用户的一些基本信息来进行校验，如果验证成功将会把用户的数据存储到session里面去，否则验证失败，并且弹出相应的提示信息，要求重新验证。登录成功之后会自动跳转到本站的首页，同时用户的之后的每次请求中都会携带我们服务器返回的session，这样服务器就能识别当前状态是哪一个用户。用户登录流程图如图5-2所示。



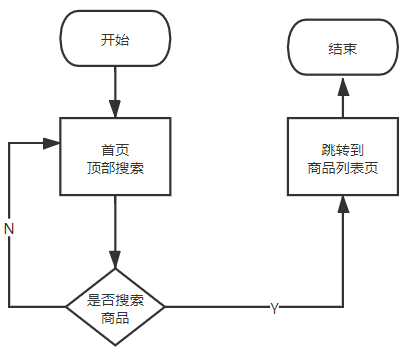
5.3 用户切换城市流程

切换城市的过程中，不管是游客状态还是普通用户状态均可以进行切换城市的操作。这里根据用户的使用行为，设计了三种切换的模式，其一，通过二级联动进行手动选择，在用户点击省份下拉箭头发出请求时，就开始请求线上的开源接口，在拿到相应的省份分类之前，城市切换默认是禁止使用的状态。在我们请求发回来的时候，省份所对应的城市变成活动状态。其二，远程搜索这里直接查询citys数据库里面的数据，同时还引用了拼音搜索的库，通过中文和拼音均可搜索。同时在远程搜索上我们为了节省资源请求，完成了延迟搜索。其三，字母的快速定位也可以实现同样的功能，也是查询citys数据里面的内容后，将数据渲染到页面上。切换城市流程图如图5-3所示。



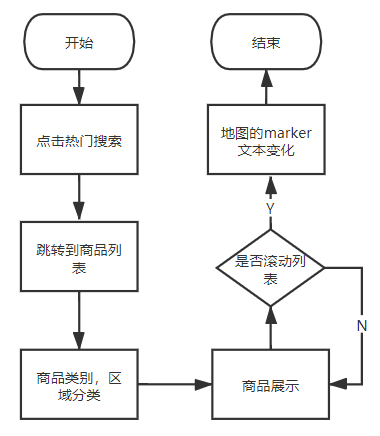
5.4 用户热门搜索流程

用户在首页的热门搜索的时候，主要是访问了数据库里面的pois表，在这个表中存储了很多用户要搜索的详细信息，在用户开始键入的时候，服务端就发出请求，并返回对应的下拉信息给用户，用户点击了对应的商品之后即可跳转到商品列表页。在首页的菜单部分，由于节点较多，为了提升用户的体验，我们采用SSR的方式渲染页面，他体验极好不会有页面闪烁的问题，同时很好的保护了接口，数据在页面展示的时候就已经下发，而用户并不知道接口的存在。首页顶部的城市定位也是使用同样的渲染方式。用户热门搜索流程图如图5-4所示。



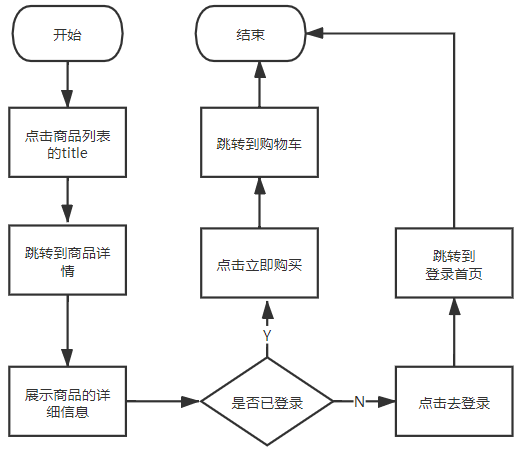
5.5 用户商品列表流程。

商品列表可以让用户快速的查看自己当前商品的店铺，在地图上显示的具体的方位。商品列表在渲染数据之前做了两层分类，一层是商品类别的分类，一种是商品区域的分类，他们两个都是访问了数据库中的categorys表，从而将同一类别的商品进行归类。而在商品列表中主要展示了所有商品的详细信息，这个列表主要是访问线上开源接口的数据，同时，我们在列表的顶部做了智能排序，分别是根据商品的价格，人气和评价。它主要是使用JS的sort方法来完成，右侧的地图主要是对左侧的列表的实时定位，当我们用户在滚动列表的时候，地图组件上的marker文本跟着实时变化，通过监听JS中的滚动事件来实现这一功能，地图组件主要是使用了高德开放平台的一个API。当用户想要查看某一商品的具体详细信息时，通过点击每个条目的标题，即可进入到商品的详情页，同时把商品的关键词和商品类型传递到详情页。用户商品列表流程图如图5-5所示。



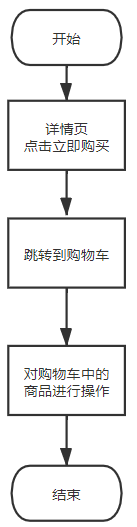
5.6 用户商品购买流程

商品购买之前会验证是否是登录状态，如果未登录商家团购及优惠底部想会显示让去登录的按钮，而如果是已登录的状态，将会显示底部显示商品列表页选中的商品的条目，我们在点击立即购买的时候，即可进入到商品购车的页面。用户商品购买流程图如图5-6所示。



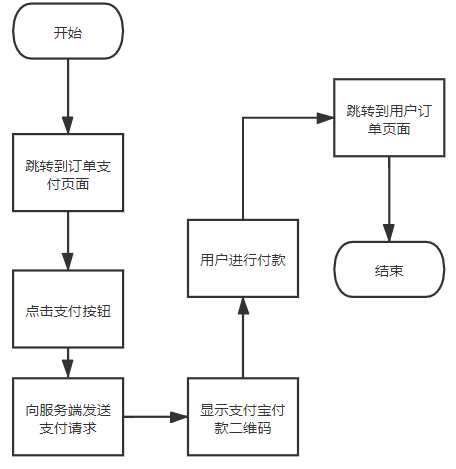
5.7 用户添加商品到购物车流程

购物车页的商品展示使用了el-table快速的搭建，这样方便我们总金额和数量使用自定义的scope去动态的计算，这样能够给我们的节省大量的代码，同时也能够避免在开发过程中多页面的相互传值的问题，购物车商品的数量我们也可以使用InputNumber计数器来快速的增减。当用户开始提交订单的时候，将自动跳转到支付订单页，同时将订单编号传递给写一个页面，用户添加商品到购物车流程图如图5-7所示。



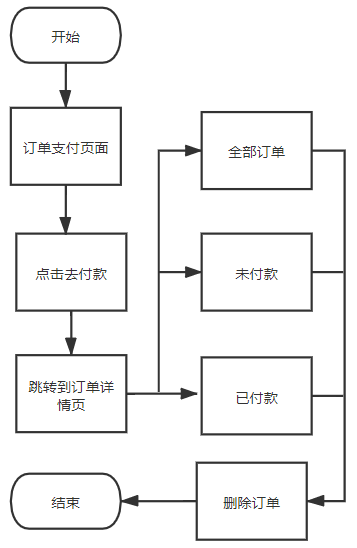
5.8 用户支付订单流程

在订单支付页面中，用户确认订单无误之后，即可提交当前的订单信息，在点击去付款按钮之后，向服务端发出一个支付请求，服务端在接收到请求之后返回给一串支付编码，开发者使用二维码生成工具将这串编码转换成对应的二维码图片，并将图片显示在用户付款的二维码框框中，在用户支付之前，前端做了一个订单轮循，每隔1000ms请求一下server服务，并查询一下当前的订单是否已支付，如果用户支付完成。则更新当前的订单状态，同时跳转到订单页面，如果用户未完成支付，可以在弹出的对话框中点击未付款继续留在当前页面，或者点击查看订单跳转到订单页面查看自己当前的支付状态。用户支付订单流程图如图5-8所示。



5.9 用户创建订单流程

用户添加的所有商品无论是已支付还是未支付的状态，都会进入到订单详情页，在这个页面可以点击分页按钮来查看全部订单，通过点击Tabs标签可以查看自己已支付和未支付的订单。若用户不想购买了，也可以通过点击删除按钮，服务端发起一个请求从orders表中删除相应的的订单信息。用户创建订单流程图如图5-9所示。



5.2小结

6 系统实现

6.1 网站主界面

6.1.1注册

注册界面主要是一个Form表单，用户可以输入自己的昵称，密码，并且填写自己的邮箱来进行注册，当用户输入了合法的邮箱后，点击发送验证码，此时会向服务端发起请求，服务端在接收到请求之后，再向腾讯的SMTP发送请求，此时，腾讯邮箱将会给我们的扣扣邮箱发送一个随机的字母数组组合的验证码。即可点击注册，此时也会调用注册接口，向服务器发起注册请求，如果没有在数据库中找到对应的用户名，则注册成功，若找到，返回提示信息给用户。用户注册界面如图6-1所示。



6.1.2登录

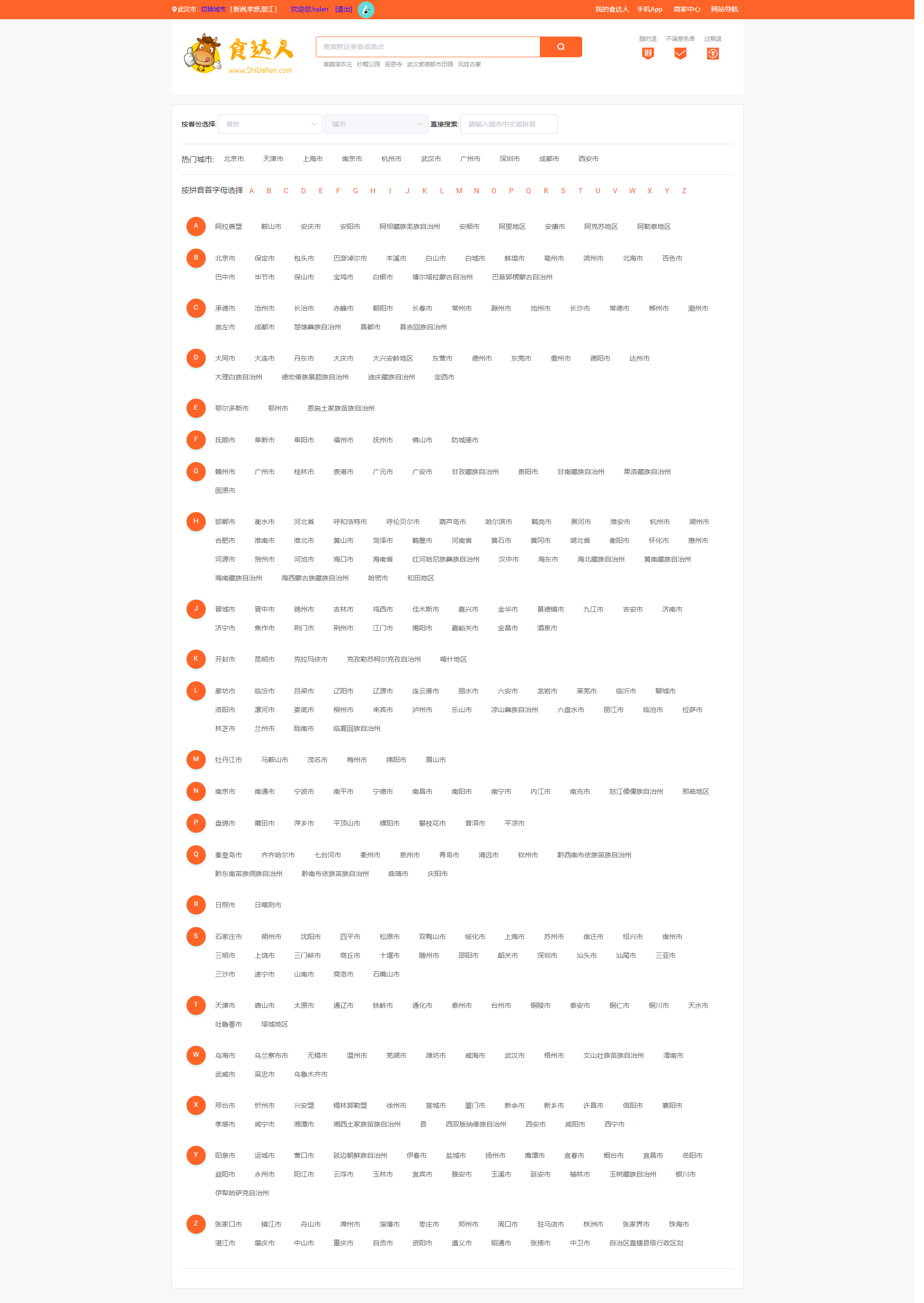
登录界面主要是一个Form表单，用户可以根据已有的账号来进行登录，若未登录可以通过点击注册，跳转到注册页面，在用户完成了用户名和密码的输入后，点击登录按钮，将会发送一个登录请求给服务端，服务端在接收到请求之后将会去查询数据库中是否已经存在此用户名和密码，如果存在，将会进入到商品首页，如果不存在，将返回相信的提示信息给用户，在登录的过程中，如果用户忘记自己的秘密或者用户名，也可以点击忘记密码，进入忘记密码页，来进行相应的修改。用户登录界面如图6-2所示。



6.1.3首页

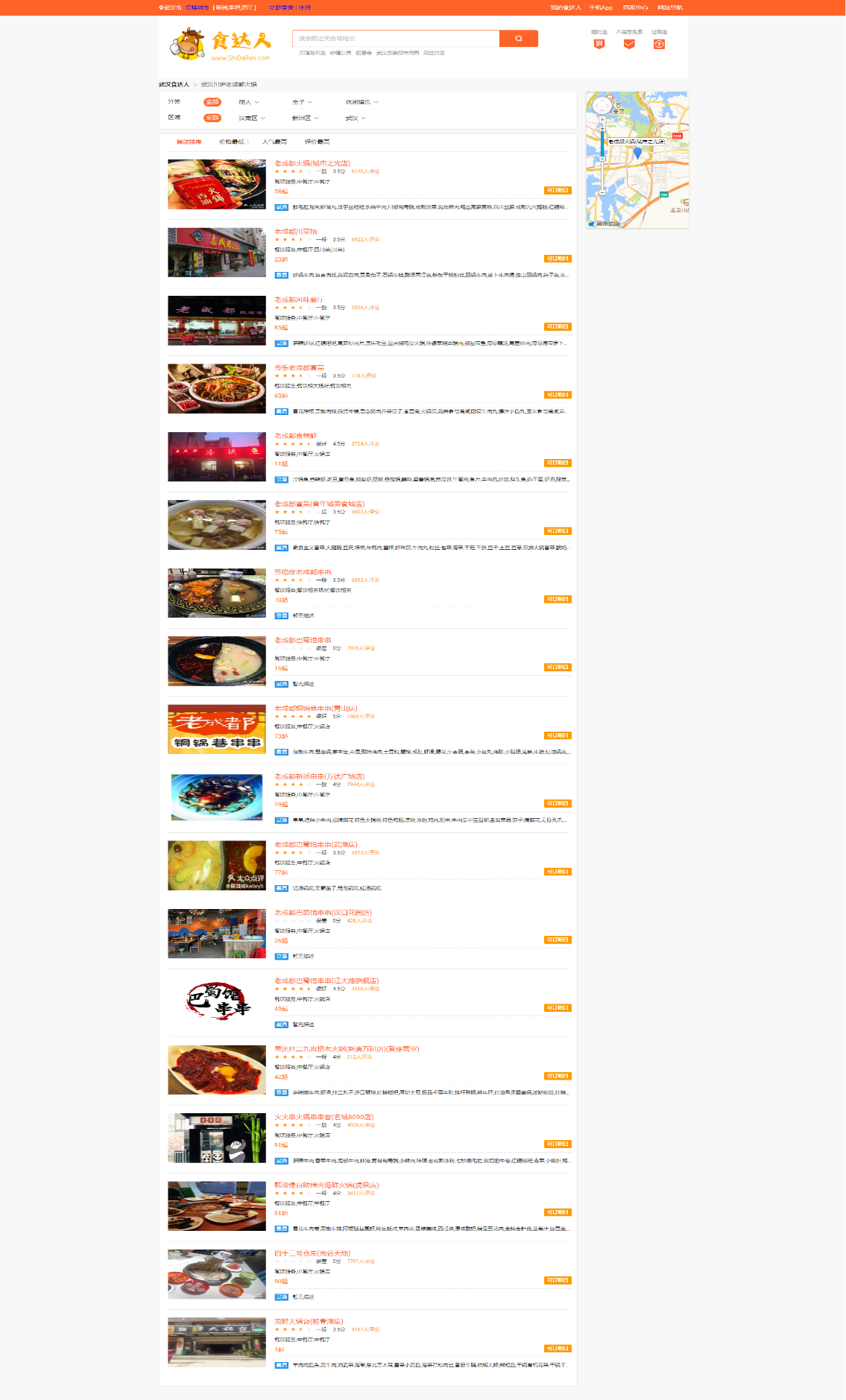
6.1.4切换城市

切换城市主要是协助用户快速的完成定位，可以通过二级联动来搜索，也可以通过远程搜索向服务器发起请求，同时我们也提供了拼音首字母的快速定位，用户可以挑选自己喜欢的方式来进行切换。用户切换城市界面如图6-4所示。



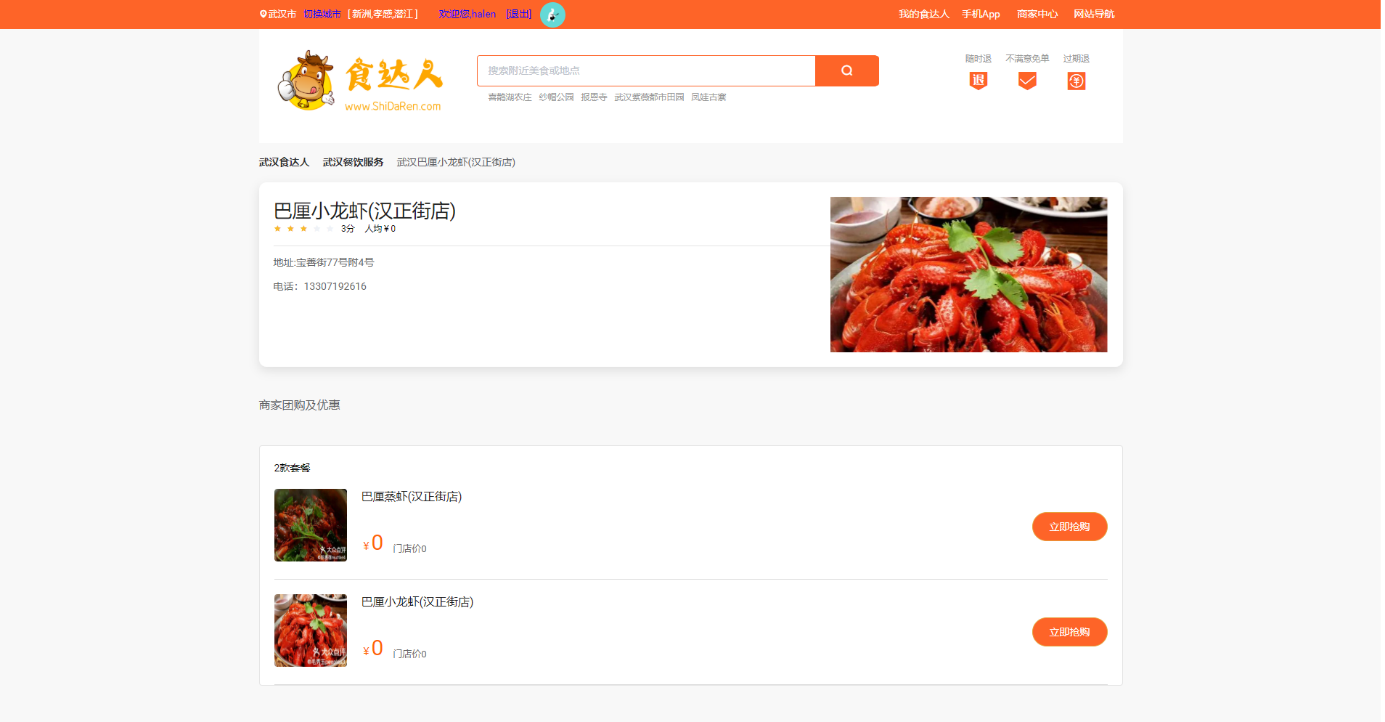
6.1.5商品列表

商品列表页主要是商品的类型和区域的分类，地图的详细定位，以及商品列表的展示，在商品列表提供了四种排序方式，分别是智能排序、价格最低、人气最高、评价最高，用户可以根据自己的需求来挑选。商品列表界面如图6-5所示。



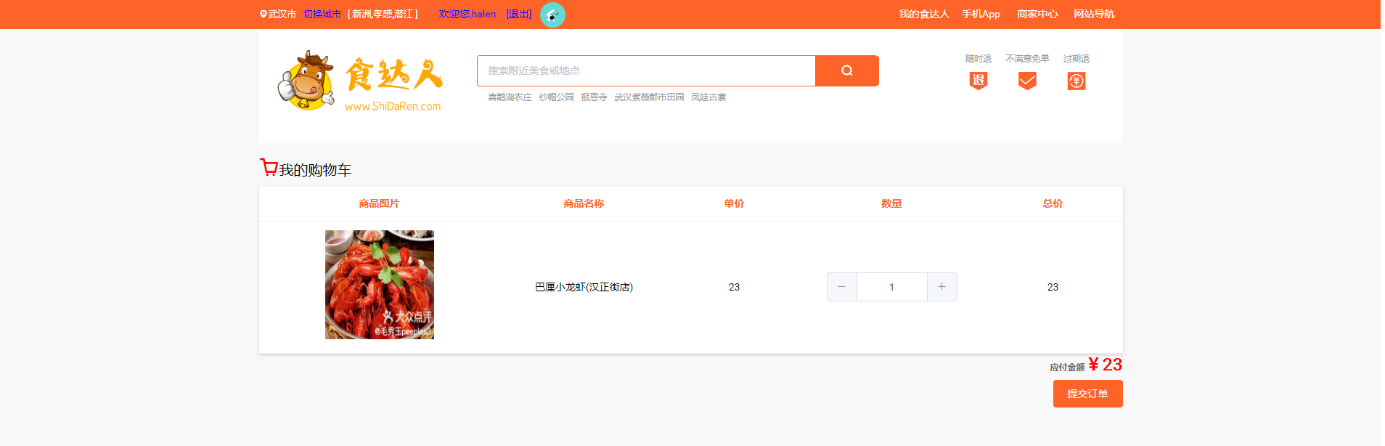
6.1.6商品详情

商品详情页主要展示商品的详细信息，包括商品名，商品地址及商品店铺的电话等信息，底部主要是商品的购买信息，若用户未注册，则显示让用户登录的一套模板，如果用户已注册，则显示让用户立即购买的一套模板。商品详情界面如图6-6所示。



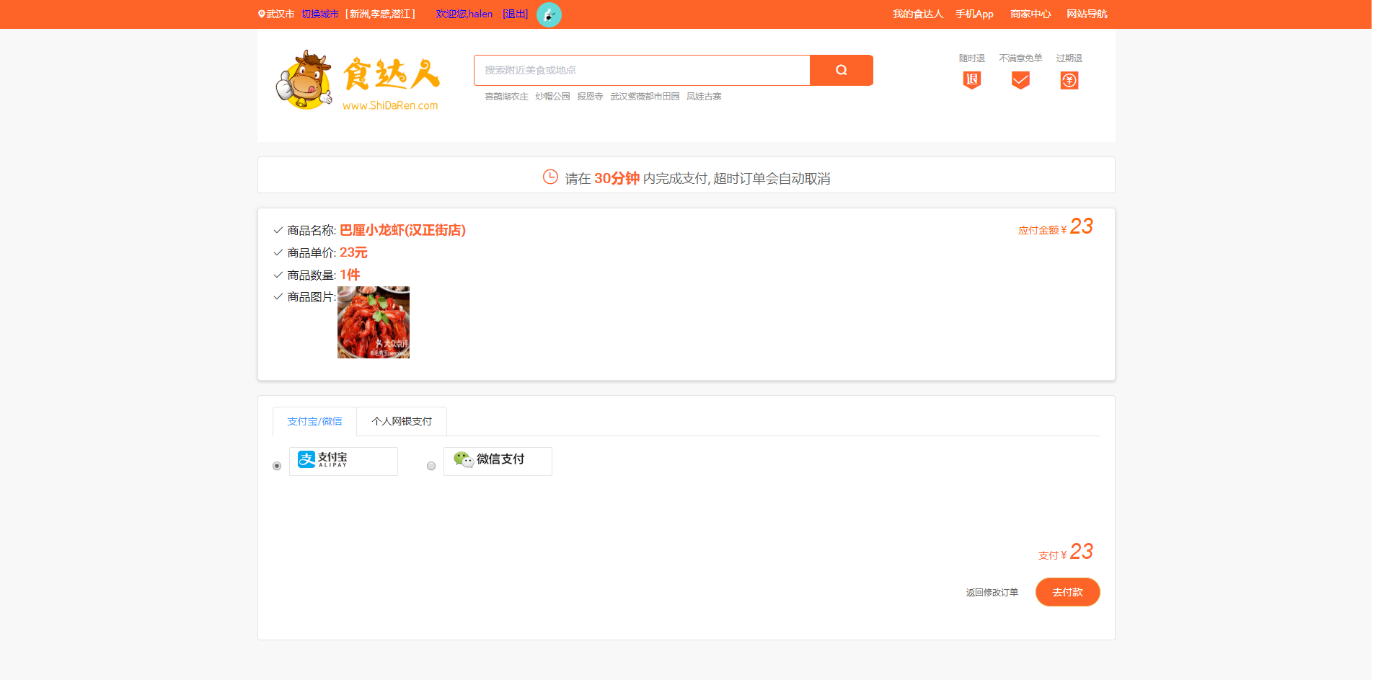
6.1.7购物车

购物车页主要是对商品的条目进行展示，用户可以根据自己的需求，添加和删减自己的订单的数量。购物车界面如图6-7所示。



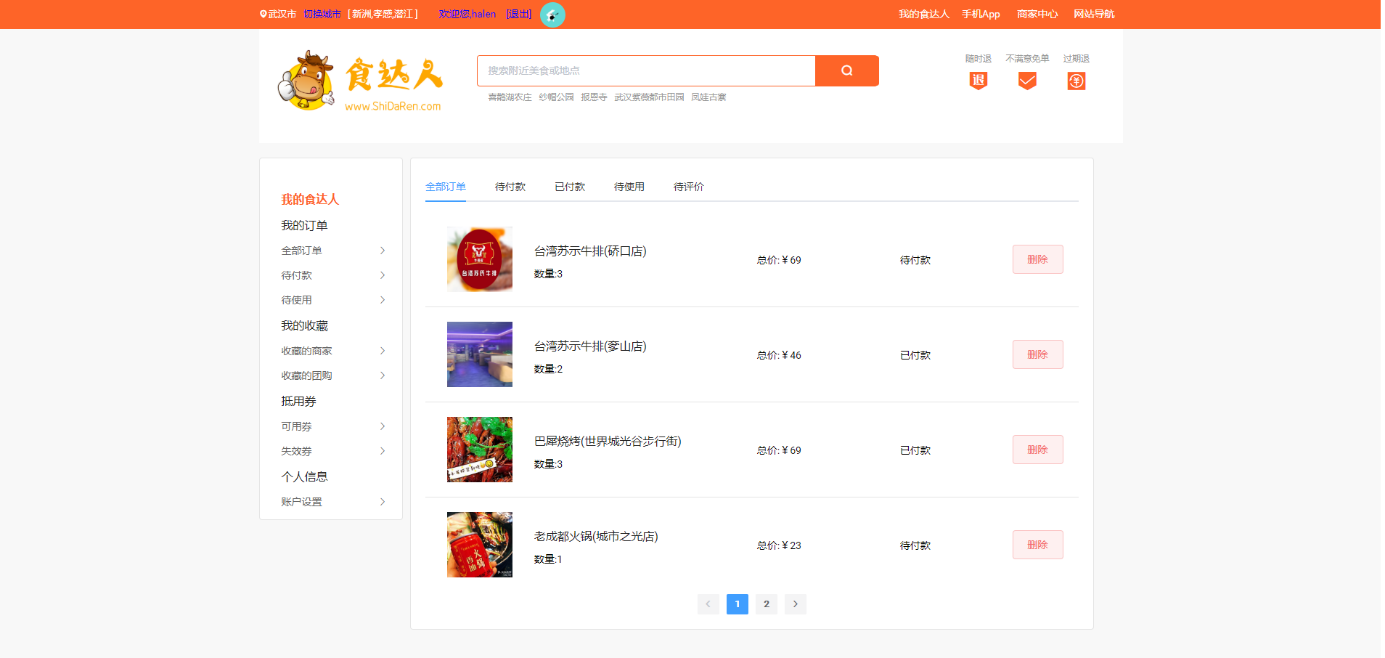
6.1.8订单支付页面

订单支付页主要是用户的购买信息，底部主要是提供支付宝的支付方式，用户在点击去支付款的时候即可弹出对应的二维码图片，可以通过扫描二维码进行支付。从而来完成相应的支付操作。订单支付界面如图6-8所示。



6.1.9订单详情

订单详情页主要是用户的订单状态，用户可以快速的查看自己当前商品的状态，如果用户不想购买了，点击删除按钮，就可以删除商品的条目信息，若订单数量较多，用户也可以通过点击分页按钮来查看。订单详情界面如图6-9所示。



参考文献

**致 谢**

时光飞逝，转眼间四年紧张而又充实的大学生生活即将画上句号。在这四年的学习期间，我得到了很多老师、同学和朋友的关怀和帮忙。在学位论文即将完成之际，我要向所有期间给予我支持、帮忙和鼓励的人表示我最诚挚的谢意。其中我要特别感谢的是我的指导老师张臣文老师，我有幸成为张老师门下的一名学生。

张老师对毕业设计做出了指导性的意见，在论文撰写过程中及时对我遇到的困难和疑惑给予悉心指点，提出了许多有益的改善性意见，投入了超多的心血和精力。在此，向老师对我的帮忙和关怀表示诚挚的谢意!

感谢导师组的各位老师，我的一点一滴的进步都离不开他们的悉心的指导和教诲，离不开他们的关心和帮助。

感谢各位同学，是你们的无私帮助让我感受到校园的温暖。

感谢我的父母，在我学习和论文期间给我许多的关心，理解和支持，才能让我安心学习，顺利完成学业。

最后再次感谢湖北文理学院为我带给了宝贵的学习机会，使我能够走上一个新的平台，幵始一段新的人生!

今后，我将继续努力，以美好的未来来报答所有关心我的老师、家人、同学和朋友！