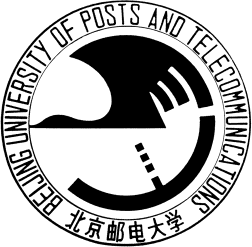


**本 科 毕 业 设 计**



**题目：购物小程序设计与实现**

姓 名 陈海霞

学 院

专 业 计算机信息系统管理

班 级

学 号

班内序号

指导教师

2022 年 4月

# 摘 要

本文首先对选题背景及微信小程序的发展现状进行系统分析，结合当今主流网络商城小程序所应具备的功能完成的系统需求分析工作，包括功能需求、性能需求以及可行性分析内容。其中系统前端功能包括注册登录、商品展示、购物车、订单结算、在线客服和用户信息管理模块；系统后台功能包括商品管、仓库管理以及系统管理模块。

其次完成系统设计工作。本小程序包括逻辑层、视图层和客户端三部分，采用类似 MVC 的开发模式完成，由客户端、小程序 API、数据库、web 服务器四部分部分组成。客户端采用 Felx 布局以确保页面在不同屏幕尺寸以及设备类型运行时可以处于恰当的位置，用户在前端完成操作后通过 Ajax 技术向服务器发送请求，服务器向数据库访问相应的数据以及访问相关的端口和 PHP 文件，进而进行数据处理操作。处理之后数据返回到客户端，经过页面视图进行渲染，用户可以查看结果。

再次完成小程序开发工作。小程序前端设计采用 WXML、WXSS 和 JavaScript 语言完成。其中 WXML 负责的是数据绑定、组件样式等设计的实现；WXSS 负责的是对 WXML 的组件样式进行渲染；JavaScript 负责的是应用逻辑层的设计。小程序后台功能采用 PHP + MySQL 的方案解决。采用 MySQL 数据库技术对归纳整理之后的国家科技支撑计划项目中相关资源进行数据库建设，来为本套网络商城小程序作数据支撑。小程序客户端的页面实现通过网络与服务器 REST API 接口通信来获取 MySQL 数据。

最后完成系统测试工作。在研究和开发完本小程序后，采用对小程序的功能、操作系统兼容性、屏幕兼容性、微信兼容性、性能以及后台接口等方面分别进行测试工作。对测试结果进行分析后，得出本小程序基本实现了各个模块功能，基本达到上线标准，可以满足用户的购物需求的结论。

本套网络商城微信小程序系统的设计虽然已经基本满足既定的功能需求，但是依然存在这一些需要增进的地方。在后续的过程中，主要应着眼于维护系统的稳定性，完善后续系统内容，提出一些系统改进方案。

关键词：网络商城，微信小程序，微信 web 开发者工具

**ABSTRACT**

So on。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。

# 目 录

关于学位论文使用授权的声明 I

原创性声明 I

[摘 要 II](#_bookmark0)

[ABSTRACT III](#_bookmark1)

[第1 章 绪论 1](#_bookmark2)

* 1. [选题背景 1](#_bookmark3)
  2. [国内外研究现状 2](#_bookmark4)
  3. [研究目的和意义 2](#_bookmark5)
  4. [论文结构 3](#_bookmark6)

[第2 章 相关理论与技术 5](#_bookmark7)

* 1. [微信 web 开发者工具 5](#_bookmark8)
  2. [微信小程序开发组件 6](#_bookmark9)
  3. [微信小程序开发 API 6](#_bookmark10)
  4. [前端开发技术 7](#_bookmark11)
  5. [后台开发技术 7](#_bookmark12)
  6. [数据库技术 8](#_bookmark13)
  7. [本章小结 9](#_bookmark14)

[第3 章 需求分析 10](#_bookmark15)

* 1. [功能需求 10](#_bookmark16)
     1. [系统功能模型 10](#_bookmark17)
     2. [系统核心业务 10](#_bookmark18)
  2. [系统性能需求 12](#_bookmark19)
  3. [可行性分析 12](#_bookmark20)
  4. [本章小结 13](#_bookmark21)

[第4 章 系统设计 14](#_bookmark22)

* 1. [系统设计原则 14](#_bookmark23)
  2. [系统架构设计 14](#_bookmark24)
  3. [系统逻辑结构设计 15](#_bookmark25)
  4. [系统功能结构设计 16](#_bookmark26)
     1. [前端功能设计 16](#_bookmark27)
     2. [后台功能模块 19](#_bookmark28)
  5. [数据库设计 19](#_bookmark29)
     1. [数据库模型 19](#_bookmark30)
     2. [数据库详细设计 20](#_bookmark31)
  6. [本章小结 23](#_bookmark32)

[第5 章 系统实现 25](#_bookmark33)

* 1. [开发环境搭建 25](#_bookmark34)
     1. [获取微信小程序 ID 25](#_bookmark35)
     2. [搭建开发环境 25](#_bookmark36)
  2. [前端功能模块实现 27](#_bookmark37)
     1. [注册登录模块 27](#_bookmark38)
     2. [商品展示模块 28](#_bookmark39)
        1. [商品分类展示 28](#_bookmark40)
     3. [购物车模块 33](#_bookmark41)
     4. [订单结算模块 36](#_bookmark42)
     5. [在线客服模块 38](#_bookmark43)
     6. [用户信息管理模块 39](#_bookmark44)
  3. [后台功能模块的实现 39](#_bookmark45)
     1. [商品管理模块 39](#_bookmark46)
     2. [仓库管理模块 40](#_bookmark47)
     3. [系统管理模块 40](#_bookmark48)
  4. [本章小结 41](#_bookmark49)

[第6 章 系统测试 42](#_bookmark50)

* 1. [测试方法 42](#_bookmark51)
  2. [系统测试环境 42](#_bookmark52)
  3. [测试结果 42](#_bookmark53)
     1. [系统功能测试结果 42](#_bookmark54)
     2. [系统兼容性测试结果 44](#_bookmark55)
     3. [系统性能测试结果 44](#_bookmark56)
  4. [本章小结 44](#_bookmark57)

[第7 章 总结与展望 45](#_bookmark58)

* 1. [工作总结 45](#_bookmark59)
  2. [未来展望 45](#_bookmark60)

[参考文献 47](#_bookmark61)

[致谢 50](#_bookmark62)

# 第 1 章 绪论

## 选题背景

从移动互联网浪潮开启以来，信息技术的发展日新月异，这也潜移默化地影响着人们的购物观念以及购物方式，特别是网上购物成为了一种新型的购物方式

[1]。传统的购物方式让我们总需要考虑时间和空间的局限性，然而网上购物的方

式让我们改变了传统购物的方法，使之变得便利、简略、快捷，可以达到足不出户即可以购买所需商品的购物方式。

2017 年 1 月 9 日，小程序正式上线的标志是“微信之父”——张小龙在广州亚运城综合体育馆进行的微信 2017 公开课 Pro 上以“下一站”为主题的发布会[5]。这个微信平台上的程序全称微信小程序，简称小程序，缩写 XCX，英文名mini program，小程序的使用优势在于其几乎是不占用手机内存的，而且在使用小程序之前是不必再应用商店中下载相信的应用软件的。用户进入小程序只需要在装有微信的手机中打开微信，直接搜索相应的小程序名称就可以直接进入小程序。同时小程序的提供给消费者的用户体验也是不逊色传统购物软件的，甚至用户提还要更高。[6]微信平台淋漓尽致地发挥了小程序“微、轻、小”等主要特点。

微信网络商城小程序的发以京东商城微信小程序的上线为开始起点的，到现 在微信平台上已经上线了各种各样样的网络商城，进一步改变了人们的购物方式。

## 国内外研究现状

小程序的主要开发语言是JavaScript，在国外 JavaScript 基本上是应用于网页开发，Google、YouTube、Facebook 等的都是利用 JavaScript 语言进行的前端开发[6]。与小程序相比，两者有很大的相似性同时还是有些许区别的。第一点长时间的脚本运行可能会导致页面失去响应的原因，是因为网页开发渲染线程和脚本线程是互斥的。而在小程序中，二者是分开的，分别运行在不同的进程中。其次， 网页开发者需要面对的环境是各式各样的浏览器，PC 端需要面对 IE、Chrome 等， 在移动端需要面对 Safari、Chrome 以及各种操作系统中的各式 WebView[7]。而小程序开发过程需要面对的是两大操作系统 iOS 和 Android 的微信客户端。

虽然小程序上线才两年的时间，但从技术的维度来看，小程序并非凭空冒出来的一个概念[8]。小程序只是一种新型的程序类型，关于小程序的构思的第一次出现其实不是在微信平台上。但是由于微信用户的规模庞大以及微信重视小程序的开发工作，使得微信小程序日渐成为我们消费者的主流应用系统的发展模式。

在当今社会我们对网络商城的概念并不感到生疏了，随着网络时代日新月异的发展，不仅网络商城的系统逐步完善，而且电子商务也成为了当今经济发展的重要因素[9]。

在国外，当前网络商城发展的领军者当属北美市场了，其中美国网络商城的快速发展走在了北美市场的前沿。具体体现在虽然美国市场仅仅经历了一年的时间就从 B2C 结算转变成为了 B2B 结算，但是通过 B2B 结算就已经超过了 B2C 结算的交易额[10]。在全球网络商城发展的过程中，即使以美国为代表的北美市场的网络商城发展突飞猛进，但是网络商城在欧洲地区的发展也在全球网络商城的发展中占有重要地位，其中欧洲依靠欧盟这一特殊的机构，整合资源，结合全球网络商城发展优点，走出了自己独特的道路，为全球网络商城发展贡献自己的力量。将目光转向国内市场，我国互联网发展的起步较晚，目前我国内的网络商城的结算依旧采用 B2C 结算[11]。我国网络上出现的首批网络商城服务到现在不过才二十多年。但是近年来我们国家计算机相关行业的迅猛发展，其中网络商城是其中的佼佼者。线上购物逐渐赶上甚至超过了线下购物，成为我国消费者的主要消费方式，同时网络商城的发展也逐渐改变我们的消费方式、生活习惯[12]。是我们国家未来消费方式的趋势。

## 研究目的和意义

本选题来源于在移动互联网快速发展的时期，用户对电子类书籍日益剧增需求，依托微信云开发的敏捷优势，设计开发一个基于微信平台的网络商城小程序，主要目的一是便于项目的研发者、组织者进行

业务推广，扩大支撑项目平台成果的影响力；二是符合当今人们高节奏的生活方式，使应用实现了触手可及，能够让消费者得到更加快捷，便捷的服务。将国家科技支撑计划课题的成果做一个更为全面、更为深入的推广[13]。

# 第 2 章 相关理论与技术

## 微信小程序开发组件

微信小程序的组件是小程序开发视图层的基本组成单元，它自带微信风格UI 样式和特定功能效果。例如，用户在小程序页面上所看到的图片、文本、按钮等都属于小程序组件[21]。微信小程序为开发者提供了一系列基础组件，通过组合这些组件可以进行更高效的开发。

一般来说我们将组件按照功能主要分为以下七类： 1. 视图容器（ View Container）组件，主要用于规划布局页面内容；2.基础内容（Basic Content）组件，主要用于显示图标、文字等常用基础内容；3.表单（Form）组件，主要用 于制作表单；4.导航（Navigation）组件，主要用于跳转指定页面；5.媒体（Media） 组件：主要用于显示图片、音频、视频等多媒体内容；6.地图（Map）组件，主要用于显示小程序的地图效果；7.画布（Canvas）组件，主要用于绘制画图内容。

## 微信小程序开发API

微信小程序开发平台给开发者提供了非常全面的 API，开发者可以很方便的通过这些 API 调用微信平台提供的能力，以便我们开发者更好地更方便地去开发小程序。小程序提供的 API 按照功能主要分为以下几大类包括：网络、媒体、文件、数据缓存、位置、设备、界面、界面节点信息等。

## 微信云开发

微信云开发是微信团队联合腾讯云推出的专业的小程序开发服务。开发者可以使用云开发快速开发小程序、小游戏、公众号网页等，并且原生打通微信开放能力。开发者无需搭建服务器，可免鉴权直接使用平台提供的 API 进行业务开发。其拥有无需搭建后端服务、快速构建、免鉴权、免登陆、统一多端应用、不限语言框架等特点。在目前敏捷开发背景具有十分突出的优势，能快速应对版本变化，极大的减少开发人员的负担与压力。

## 开发技术

当前小程序开发采用UNI-APP，UNI-APP 是一个使用 Vue.js 开发所有前端应用的框架，开发者编写一套代码，可发布到iOS、Android、Web（响应式）、以及各种小程序（微信/支付宝/百度/头条/飞书/QQ/快手/钉钉/淘宝）、快应用等多个平台。其拥有跨平台、页面反应速度快、生态丰富十分活跃、快速构建等特点，自小程序生态开展以来，UNI-APP风靡各大应用场景。

在微信小程序的开发中微信平台为开发人员引入了 WXML 和 WXS语法。其中WXML（WeiXin Markup Language）是框架设计的一套标签语言，结合基础组件、事件系统，可以构建出页面的结构。WXS（WeiXin Script）是小程序的一套脚本语言，结合 WXML，可以构建出页面的结构。但WXS 与 JavaScript 是不同的语言，有自己的语法，并不和 JavaScript 一致。

## 数据库技术

MongoDB是一个基于分布式文件存储的数据库。由C++语言编写。拥有高性能、丰富的语言查询、高可用、水平可拓展性、支持多个存储引擎等特点，旨在为WEB应用提供可扩展的高性能数据存储解决方案,MongoDB随着物理网时代的带来，应用越来广泛。

MongoDB是一个介于关系数据库和非关系数据库之间的产品。MongoDB从3.0版本开始默认使用的是WiredTiger引擎， WiredTiger引擎可以针对单个文档来保证ACID特性，MongoDB从4.0开始支持副本集多文档事务，从4.2版本开始支持分布式事物，因此可以跨多个操作，集合，数据库，文档和分片使用事物。

MongDB中的记录是一个文档，它是由字段和值对组成的数据结构。MongoDB文档类似于JOSN对象，其结构如下：



# 第 3 章 需求分析

## 业务分析

在整个基于小程序的商城中，首先由本商城的管理人员在后台进行电子类书籍的上架，待书籍上架后，客户刻在小程序前台界面进行数据购买，待客户成功付款，商家应在24小时内发货，若客户未付款，则代付款订单24小时内自动取消。其业务流程图如下：



## 功能需求

* + 1. 系统功能模型

本选题研究的主要内容包括网络商城小程序的系统设计。结合当今电子商务平台的微信小程序设计的优点。得出所设计的网络商城小程序系统主要的功能模块应该包括前端功能模块和后台功能模块这两部分。其中前端功能模块主要有注册登录、商品展示、购物车、订单结算、个人信息等模块的功能； 后台功能模块主要有注册登录、商品管理、用户管理、订单管理等模块的功能。系统功能模块如图3.1所示



图 3.1 购物小程序功能模块图

* + 1. 系统核心业务
       1. 前台功能

用户进入小程序后使用正确的用户名和登录密码成功进入本系统，可以对自己的信息进行修改和维护。可以更改用户昵称、密码登录信息、用户名、支付密码、出生日期、性别、验证的手机号码等信息。

当用户选中心仪书籍商品，将先关书籍商品加入购物车。用户可对购物车内的书籍商品添加、删除、增减数目。再其对商品操作的同时，购物车下方价格应随之变化等。

客户确认收货后可以对商品举行评价，可对商品文字是否清晰，章节是否有完整，客服服务态度，进行打分，同时还提供发布图片与追加评价等功能，其用例图如下：



用户用例图

* + - 1. 后台功能

在本系统中，商家即是管理员，商家把所要销售商品的详细信息全部发布到网站供用户浏览选购。陈列的信息包括商品名称、商品描述、商品图片、商品价格、商品类别、销售量、作者、评价数、好频率等。其中销售量、评价数、好评率是系统自动计算，管理员无法填写。商品陈列功能是系统非常基础的功能，是用户完成商品购买这一过程的第一步。

用户可以检查在本系统中购买的商品的全部订单，只有付款的定单商家才会发货，待付款的订单，如用户超过24小时未付款，则系统会自动逻辑删除该笔未付款的订单。系统需要对每种状态的订单进行管理，以便会员和商家都能够清楚高效地了解商品的动态，其用例图如下：



管理员用例图

## 系统性能需求

一个系统成功实现的标志是运行顺畅的界面。相反如果系统卡顿，伴随而来的是，打开界面的时候动作缓慢，响应不及时，轻则带来糟糕的用户体验，导致客户流失，重则在卡顿的过程中极可能导致收到外部攻击，从而导致客户资金损失。

1. 安全性需求：由于该网络商城小程序必然会涉及到金钱交易，因此保 证消费者的利益是至关重要的。保障消费者的利益的重点是提高该系统的安全性， 通过采用一系列最先进的安全防范技术来保障消费者的账户资金安全。同时需要 考虑的是安全性和系统便捷性之间的平衡。而不是一昧的最求安全，放弃用户体验和网站的流畅性。
2. 稳定性需求：在消费者通过该网络商城购物时不希望看到的是系统卡顿、界面乱码、提供信息错误。这些问题均为系统没有提供较高稳定性所导致的。因此，如果想让消费者拥有更好的线上购物体验，我们提高系统稳定性这一过程就是必不可少的。
3. 可靠性需求：该系统投入生产之后，可能随着业务的飞速拓展，用户量急剧上升，这种高并发带来大巨大压力，可能导致用户信息，订单数据错乱，这些均是可靠性不足导致的，因此可通过负载均衡等一些列解决方案来弥补该问题。
4. 响应时间要求：对一个系统的评价标准，首先是响应速度。当消费者使用该网络商城小程序的时候如果连UI 界面都十分卡顿，那么就会丧失用户体验，不利于这个小程序的进一步推广，可通过优化程序本身或者CDN等技术来提升用户体验。

## 可行性分析

本系统立足于微信小程序同时基于小程序的云开发展开，在技术方面具有可行性。在移动互联网技术发展迅速，大众对于电子类书籍的需求日益剧增，同时商城主要是经营电子类书籍，正好切入市场巨大需求之中，通过电子类书籍展开突破口，吸纳大量的用户，同时由于云开发带来的便捷性，高效性和低成本的特点，故在经济上具有可行性。

# 第 4 章 系统设计

## 系统设计原则

微信小程序设计的基本原则是在完成自己的既定要求的情况下针对微信客 户端进行小程序的优化。以下设计原则都是基于对用户的尊重的基础上的，旨在微信生态类建立高效、一致的用户体验的同时，最大程度顺应和支持各业务需求设计，实现用户与程序的共赢。包括：1. 友好礼貌（为了避免用户在微信中使用小程序服务时，注意力被周围复杂环境干扰，小程序在设计时应该注意减少无关的设计元素对用户目标的干扰，礼貌地向用户展示程序侧提供的服务，友好地引导用户进行操作。）；2. 清晰明确（作为一个负责任的开发者，一旦用户进入我们的小程序页面，就有责任和义务清晰明确地告知用户身在何处、又可以往何处去，确保用户在页面中游刃有余地穿梭而不迷路，这样才能为用户提供安全的愉悦的使用体验。）；3. 便捷优雅（从 PC 时代的物理键盘鼠标到移动端时代手指，虽然输入设备极大精简，但是手指操作的准确性却大大不如键盘鼠标精确。为了适应这个变化，需要开发者在设计过程中充分利用手机特性，让用户便捷优雅的操控界面。）等。

## 系统架构设计

微信小程序的框架包含两个部分，分别为视图渲染View 层 和代码逻辑 AppService层。

视图渲染层用于渲染页面结构，代码逻辑层用于运行 JS 脚本。视图层和 逻辑代码层采用了双线程方式进行管理。视图层运行于一个 Webview 线程。在 Webview 线程中，会将 wxml 转化成 html, 将 wxss 转化为 css，最终展示成我们的视图。

代码逻辑层运行在另一个 Webview 线程，即 JS 执行引擎线程（不同的环境中，实现的引擎不一样，在 IOS 中，使用 JavaScriptCore ， 在 安卓中使用 V8 引擎或者 x5 JS 解析器，在 开发工具中，使用 nwjs Chrome 内核 ，我们统称为 JSCore）。在 Jscore 中，提供了 javaScript 的运行环境，在这个线程中执行小程序主要执行小程序的逻辑代码。而视图层和代码逻辑层是两个单独的执行线程，而这两个线程之间的通信则是通过我们的 JSBridge。



图 4.1 系统架构图

## 数据库设计

* + 1. 数据库模型

基于微信小程序的网络商城系统研究和开发的时候最重要的内容就是针对商品、用户等信息进行数据库的设计，网络商城小程序提供给消费者最基本的内容就是商品的各种信息。一旦数据库的设计发生问题或者是数据库设计的不够合适，会导致消费者在网络商城小程序中购物时体验不好，浪费时间同时还给后期增减商品信息等数据库操作带来很多麻烦。在数据库设计的时候，我们必须牢记数据库设计中的相关规则规定，这样下来我们设计的数据库才能给小程序的使用者带来良好的用户体验以及在后期对数据库操作的时候也会十分方便不必浪费过多的时间成本等。

* + 1. 数据库详细设计
       1. 数据库概念设计：在进行数据库设计的时候首先要进行的工作就是进 行数据库概念设计，优秀的数据库概念设计是创建一个优秀的数据库的前提保证。数据库概念设计就是将我们现实中的具体事物转换为虚拟的数字信息，在这里就 是将网络商城中的商品转换为抽象的数字信息。在这款网络商城小程序数据库概 念设计的时候使用的方法是目前概念设计中最常用的方法——创建 E-R 图。我 们创建了包括商品信息、用户信息、订单信息和商品分类信息等多种信息实体的E-R 图，数据E-R图如下。

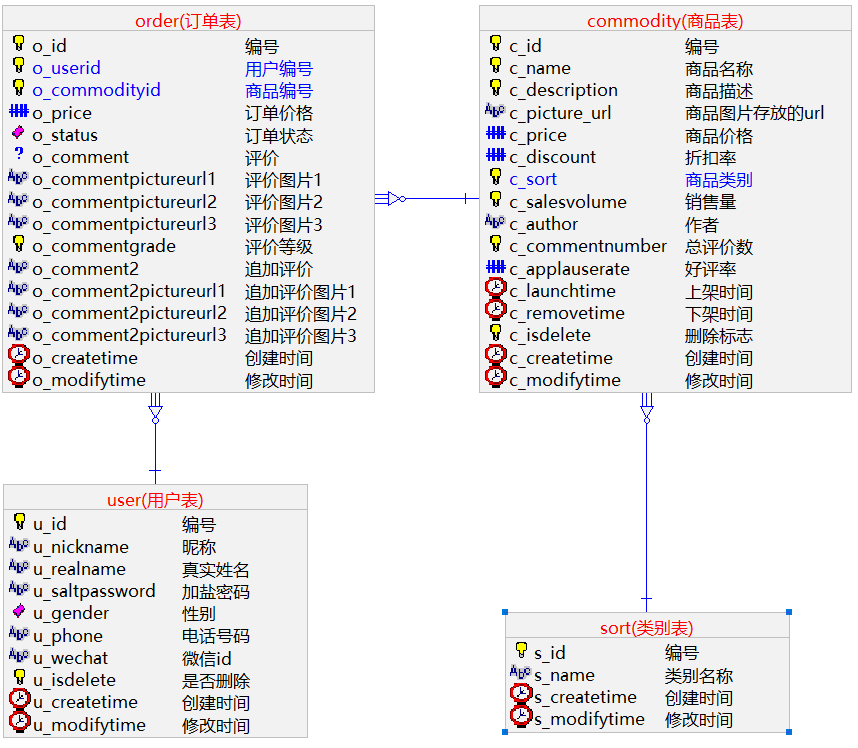


图 4.6 数据库E-R图

* + - 1. 逻辑结构设计：在进行网络商城小程序数据库设计的第二阶段就是进 行数据库的逻辑方式设计。现实世界中的具体事物映射到数据库中就是关系模型、数字信息等。上文刚刚介绍了这套网络商城小程序中相关实体的概念设计以及

E-R 图的创建，结合上文所述内容我们方可以进行数据库的逻辑方式这个阶段的设计。也就是要做的工作具体为结合上文所创建的 E-R 图来设计相应的关系表来供数据库使用，其中 E-R 图中的实体、属性和联系等都要与表一一对应。在制作信息表示一定要标明 key，如果涉及到与其他表得联系时还设置 foreign key。接下来要仔细介绍的就是具体关系表的绘制。详细信息表的情况如下文所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用户信息表（user）基本数据表，如表 4.2。  表 4.2 用户信息表（user） | | | | |
| 字段名 | 类型 | 大小 | 为空 | 字段描述 |
| u\_id | String | 20 | 否 | 自增id |
| u\_nickname | String | 60 | 否 | 昵称 |
| u\_realname | String | 60 | 否 | 真实姓名 |
| u\_password | String | 10 | 否 | 密码 |
| u\_gender | Enum | 1 | 否 | 性别 |
| u\_phone | String | 15 | 否 | 电话号码 |
| u\_wechat | String | 30 | 否 | wechatid |
| u\_isdelete | Integer | 1 | 否 | 是否删除 |
| u\_createtime | Date |  | 否 | 创建时间 |
| u\_modifytime | Date |  | 否 | 修改时间 |
| 分类信息表（sort）基本数据表，如表 4.3。  表 4.3 分类信息表（sort） | | | | |
| 字段名 | 类型 | 大小 | 为空 | 字段描述 |
| s\_id | String | 20 | 否 | 自增id |
| s\_name | String | 60 | 否 | 名称 |
| s\_createtime | Date |  | 否 | 创建时间 |
| s\_modifytime | Date |  | 否 | 修改时间 |

订单信息表（order）基本数据表，如表 4.4。

表 4.4 订单信息表（order）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 大小 | 为空 | 字段描述 |
| o\_id | String | 20 | 否 | 自增id |
| o\_userid | String | 20 | 否 | 用户id |
| o\_commodityid | String | 10 | 否 | 商品id |
| o\_price | double | 20 | 否 | 商品价格 |
| o\_comment | String | 10 | 否 | 评论 |
| o\_commentpictureurl1 | String | 500 | 否 | 评论图片1 |
| o\_commentpictureurl2 | String | 500 | 否 | 评论图片2 |
| o\_commentpictureurl3 | String | 500 | 否 | 评论图片3 |
| o\_commentgrade | Integer | 1 | 否 | 评论等级（1到5级） |
| o\_comment2 | String | 500 | 否 | 追评 |
| o\_comment2pictureurl1 | String | 500 | 否 | 追评图片1 |
| o\_comment2pictureurl2 | String | 500 | 否 | 追评图片2 |
| o\_comment2pictureurl3 | String | 500 | 否 | 追评图片3 |
| o\_createtime | Date |  | 否 | 创建时间 |
| o\_modifytime | Date |  | 否 | 修改时间 |

订单信息表（commodity）基本数据表，如表 4.4。

表 4.4 商品表（commodity）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 大小 | 为空 | 字段描述 |
| c\_id | String | 10 | 否 | 自增id |
| c\_name | String | 20 | 否 | 商品名称 |
| c\_description | String | 10 | 否 | 商品描述 |
| c\_picture\_url | String | 10 | 否 | 图片url |
| c\_price | Double | 10 | 否 | 商品价格 |
| c\_discount | Double | 10 | 否 | 折扣率 |
| c\_salesvolume | Integer | 11 | 否 | 销售量 |
| c\_author | String | 50 | 否 | 作者 |
| c\_commentnumber | Integer | 10 | 否 | 评论数 |
| c\_applauserate | Double | 5 | 否 | 好评率 |
| c\_launchtime | Date |  | 否 | 上架时间 |
| c\_removetime | Date |  | 否 | 下架时间 |
| c\_isdelete | Integer | 1 | 否 | 是否删除 |
| c\_createtime | Date |  | 否 | 创建时间 |
| c\_modifytime | Date |  | 否 | 修改时间 |

# 第 5 章 系统实现

## 开发环境搭建

* + 1. 获取微信小程序 ID

获取小程序AppID为开发微信小程序的开启步骤，后续的所有开发流程都是基于这个 AppID来完成的。需在微信公众号平台注册并获取AppID，其内容如下：



* 1. 查看 AppID 详情
     1. 搭建开发环境

在获取微信小程序AppID后，需要完善开发环境。在开发者界面下载对应的依赖包。本文开发的基于微信小程序的网络商城 MAC 上进行开发的[39]。安装成功后，详细开发环境配制情况如下表5.1所示。

表 5.1 开发环境配置表

|  |  |
| --- | --- |
| 操作系统 | MAC |
| 微信 web 开发者工具 | V1.02.1812271 |
| MongoDB | V4.2.0 |

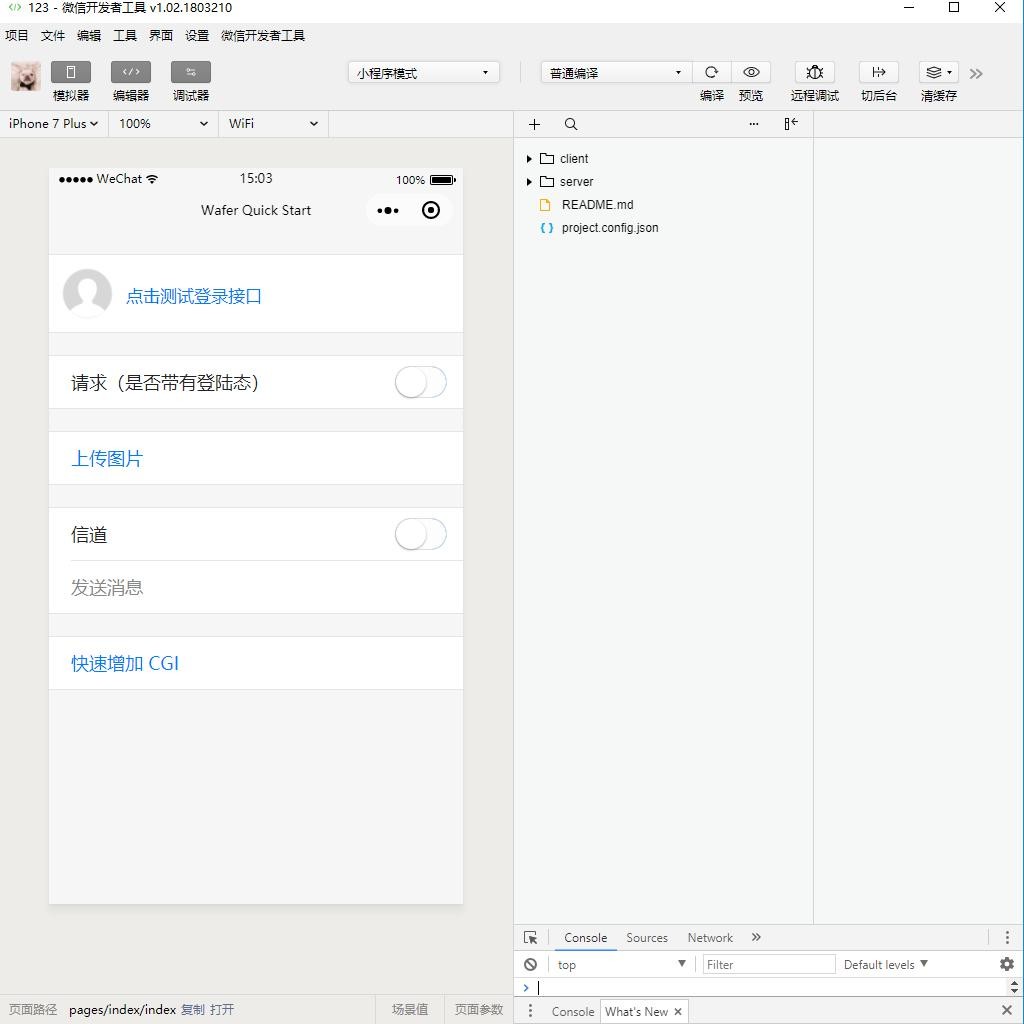
在安装完之后，打开微信 web 开发者工具图表，管理员使用微信扫描二维码后进入菜单界面，完后单击菜单中的“小程序项目”选项，进入小程序项目管理界面，然后开发者依次填写项目目录、AppID 和项目名称。将上述内容填写完毕后单击“确定”按钮完成操作，跳转到微信小程序开发界面，如图 5.2 所示。

图 5.2 小程序项目开发页面

## 前端功能模块实现

* + 1. 注册登录模块

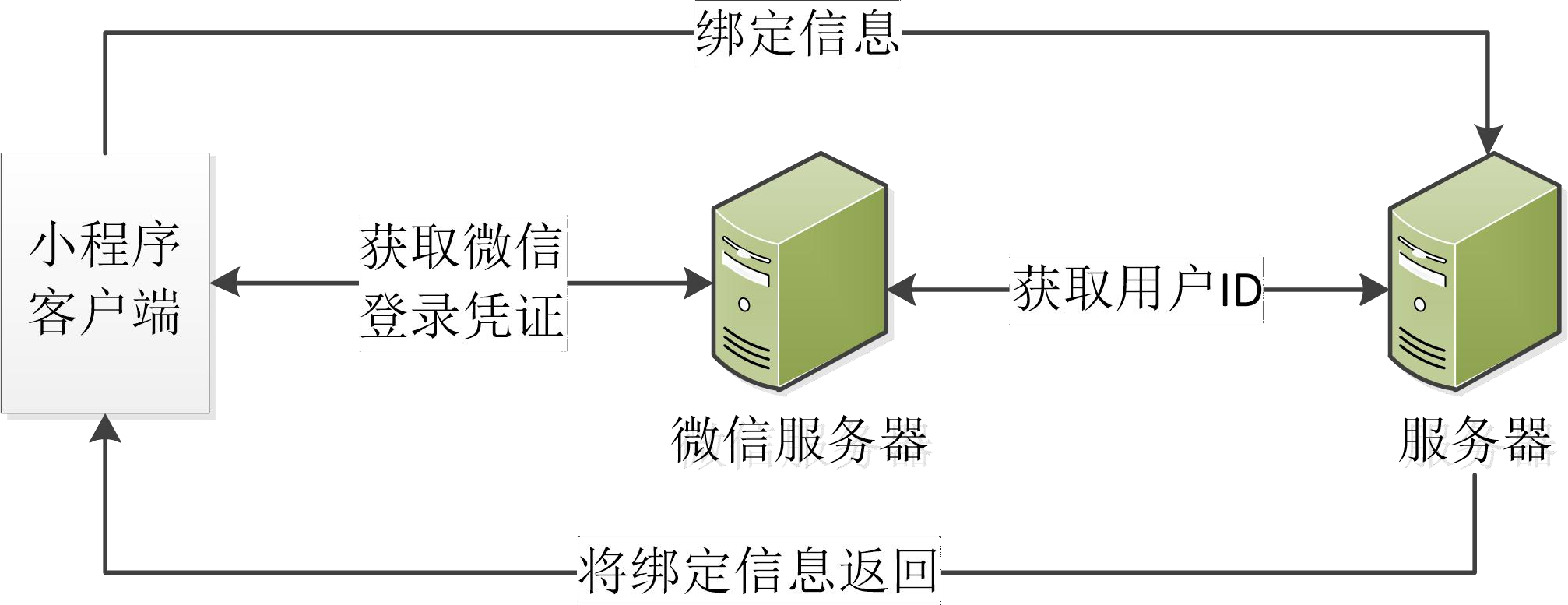
在消费者第一次使用这套网络商城小程序的时候，首先要进行的就是注册登录功能，在微信平台上我们可以直接接入官方提供的 wx.login()API 端口来确认消费者的实名信息、联系电话等内容[40]。这套网络商城小程序在实现注册登录功能方面的具体流程如下图 5.3 所示。

图 5.3 注册登录功能实现流程图

这个注册登陆的流程大体上是通过上述的 wx.login()API 端口，我们可以得到一个可以唯一确定消费者身份的代码，这里称之为code 码。然后通过将这个code 码来调用也是微信官方提供的 wx.login()API 端口。通过这个端口我们就可以拥有得到消费者个人相关信息的权利了。

通过这一系列操作就能完成网络商城小程序获取消费者信息的工作了。这样下来不仅免去了消费者注册账号的复杂流程节省了大量时间进而提高了用户体验，而且也节省了网络商城小程序管理者储存消费者个人信息的相关成本。

<view wx:if="{{canIUse}}">

<view class='header'>

<image src='/images/wx\_login.png'></image>

</view>

<view class='content'>

<view>申请获取以下权限</view>

<text>获得你的公开信息(昵称，头像等)</text>

</view>

<button class='bottom' type='primary' open-type="getUserInfo" lang="zh\_CN" bindgetuserinfo="bindGetUserInfo"> 授权登录

</button>

</view><view wx:else>请升级微信版本</view>

上述内容即为网络商城小程序在注册登录这个过程中具体实现这一功能的相关代码实现。这套网络商城小程序在实现注册登录功能方面的实现效果如下图

5.4 所示。

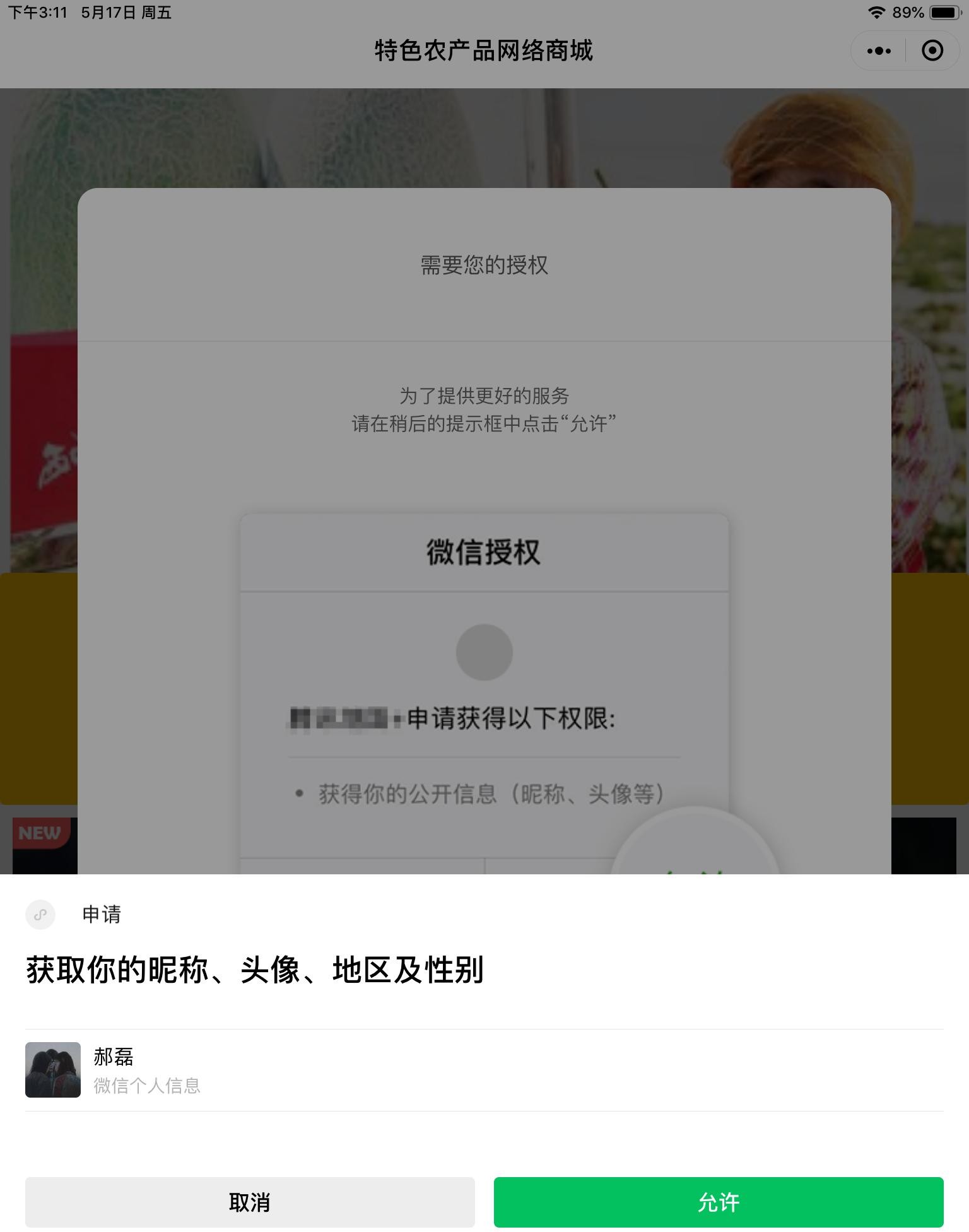


图 5.4 注册登陆功能实现效果图

* + 1. 商品展示模块
       1. 商品分类展示

消费者在使用这套网络商城微信小程序时，可以通过选择商品产地进行浏览商品的操作，这里就涉及到对商品按照产地来进行分类展示。本文是将商品产地分为甘肃、贵州、新疆、宁夏、重庆、西藏等六大类，通过动态 API 来实现分类的功能。具体操作包括：1.在小程序前端界面设计上述六类地区的页面；2.通过使用动态 AP 来进行商品数据的获取； 3.消费者通过选择相应的商品分类选项进入相应的商品集合并加载相应数据信息。实现的关键代码如下所示。

<!--主盒子-->

<view class="container">

<!--左侧栏-->

<view class="nav\_left">

<block wx:for="{{classifyItems}}">

<!--当前项的 id 等于 item 项的 id，那个就是当前状态-->

<!--用 data-index 记录这个数据在数组的下标位置，使用 data-id 设置每个 item 的 id 值，供打开 2 级页面使用-->

<view class="nav\_left\_items {{curNav == item.id ? 'active' : ''}}" bindtap="switchRightTab" data-index="{{index}}" data-id="{{item.id}}">

{{item.name}}

</view>

</block>

</view>

<!--右侧栏-->

<view class="nav\_right">

<!--如果有数据，才遍历项-->

<view wx:if="{{classifyItems[curIndex].ishaveChild}}">

<block wx:for="{{classifyItems[curIndex].shopClassifyDtoList}}">

<view class="nav\_right\_items">

<!--界面跳转 -->

<navigator url="/pages/classifyGoods/classifyGoods?classifyId={{item.id}}">

<image src="{{item.imgUrl}}"></image>

<text>{{item.name}}</text>

</navigator>

</view>

</block>

</view>

<!--如果无数据，则显示数据-->

<view class="nodata\_text" wx:else>该分类暂无数据</view>

</view>

</view>

完成上述配置之后，消费者就可以按照商品分类进行浏览心怡的商品了。商品分类展示功能实现效果如图 5.5 所示。

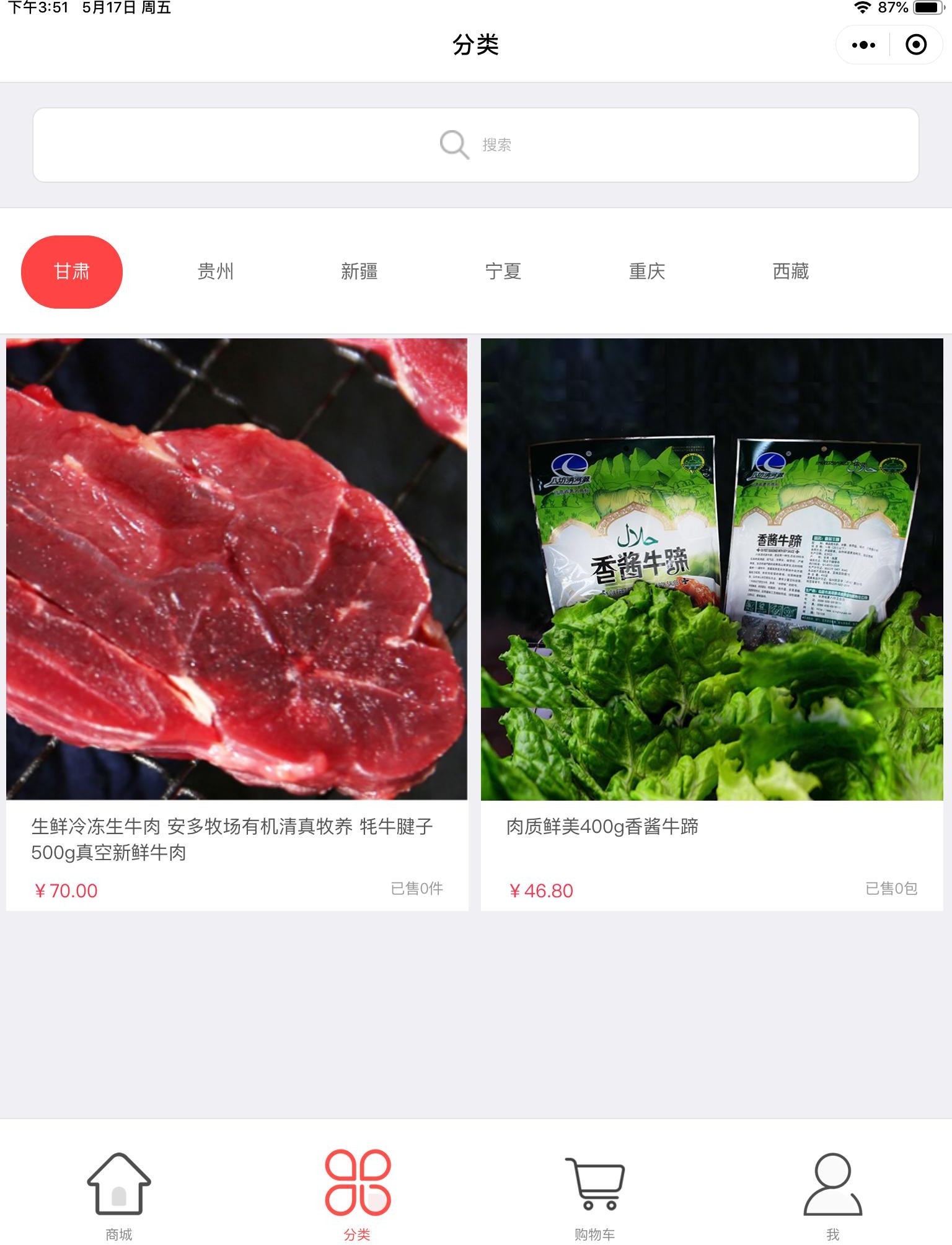


图 5.5 商品分类展示功能效果图

* + - 1. 商品关键字检索

消费者在浏览商品时，可以通过在检索栏中对商品的关键字检索从而快速找到相关产品。实现商品关键字检索功能的关键步骤包括：1.设计商品关键字检索的 UI 界面，这里采用的是创建 wxml 文件。设计商品检索输入框，将输入的商品检索信息传入到后台数据库进行存储。然后通过 JavaScript 文件、循环遍历算法来进行对搜索数据的查找；2.设计 UI 检索页面的样式以及检索结果的呈现， 这里采用的是创建 wxss 文件；3.通过第一步创建的 JavaScript 文件，将输入的检索数据进行分析处理。通过将 wx.request、URL 等方法，在后台数据库中查找相应商品编号； 4.通过在服务器端建立 PHP 搜索服务文件根据 URL 与商品编号联系，在后台数据库中对信息表中的数据进行遍历查找，最后得出搜索结果。并且将搜索结果的数据通过 JavaScript 文件、json 文件等形式反馈到小程序客户端。5.将反馈回小程序客户端的的数据信息通过页面视图层的渲染，最后将渲染结果呈现给用户，从而完成整个商品关键字检索过程[41]。实现的关键代码如下所示。

<!-- 标题 -->

<view class="title">小程序搜索</view>

<!-- 搜索框 view -->

<view class="search\_con">

<!-- 表单 -->

<form bindsubmit="formSubmit">

<!-- 记得设置 name 值，这样 JS 才能接收 name=keyword 的值 -->

<input type="text" name="keyword" class="search\_input" placeholder=' 你要找什么呢？'/>

<button formType="submit" class="search\_btn">搜索</button>

</form>

</view>

<!-- 搜索结果展示 -->

<view wx:for="{{re}}" wx:key="re" class="search\_result">

<!-- 当提交空白表单的时候 -->

<view class="empty">{{item.empty}}</view>

<!-- 当有搜索结果的时候 -->

<view class="resname">{{item.resname}}</view>

<!-- 当查询不到结果的时候 -->

<view class="noresult">{{item.noresult}}</view>

</view>

在完成以上配置之后，我们的网络商城小程序就能够实现按商品关键字检索商品的功能了，具体实现效果如图 5.6 所示。

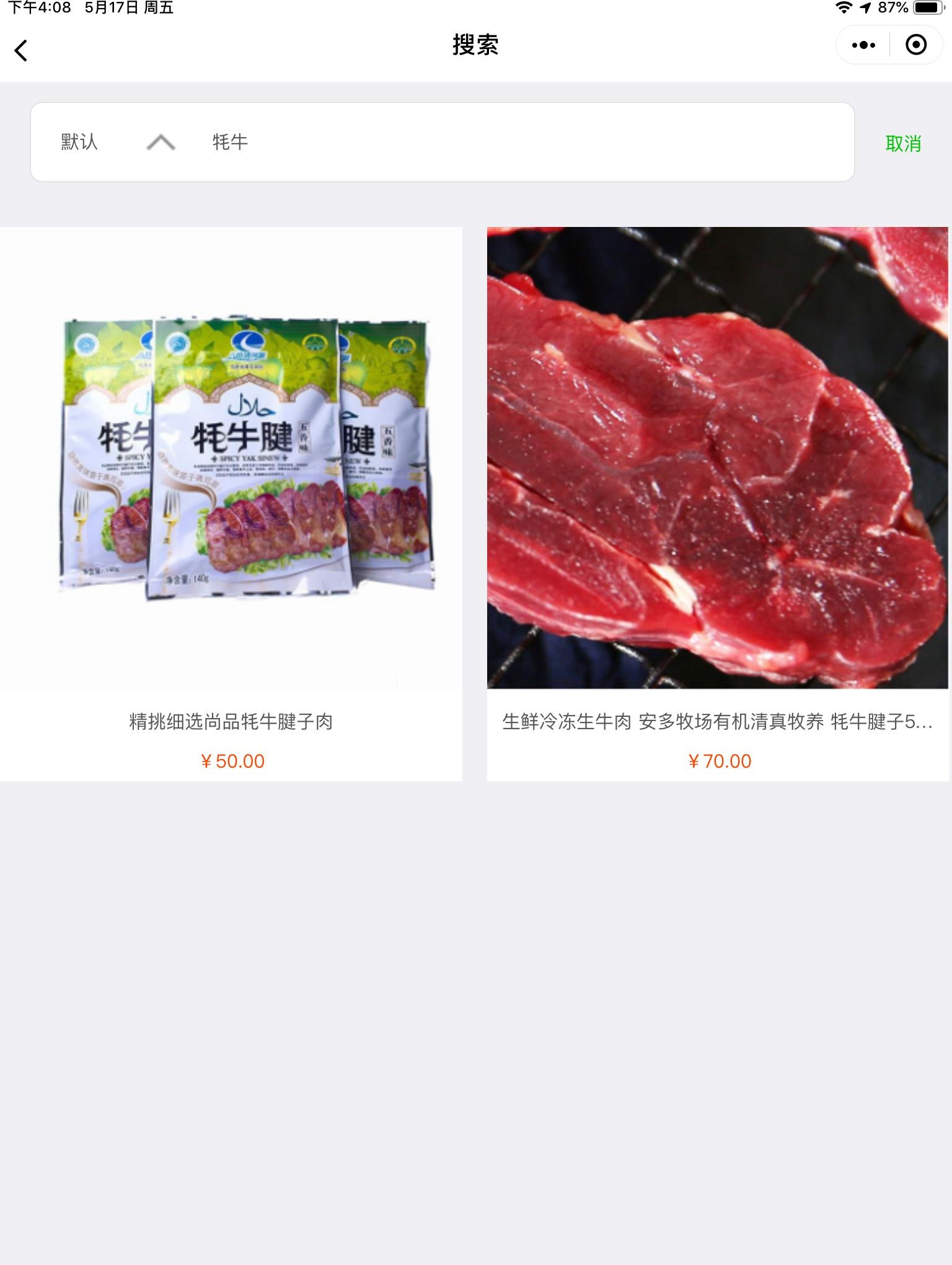


图 5.6 商品关键字检索效果图

* + - 1. 多语言展示

由于本文所设计研发的网络商城小程序是源于国家科技支撑项目源于国家 科技支撑计划项目“民族特色农产品多语言网络交易展示平台关键技术集成与应用示范”，其目的是为了更好地推广这个平台。因此开发多语言展示这项功能是很有必要的。实现这一功能的主要步骤是：1.对网络商城的 UI 界面中的信息分别创建藏语、维吾尔语、蒙语等三门语言与汉语的对应语料库；2.通过调用wx.T.get anguage()的端口来实现 UI 前端显示语言的切换。3.微信中的 langIndex 缓存模块，在完成小程多语言功能切换的时候都需要将这个缓存模块重新存储。实现多语言展示功能的关键代码如下。

import event from '../../utils/event';

data: {language: '', //多语言

},

onLoad: function(options) {//多语言wx.T.setLocaleByIndex(globalData.langIndex); that.setLanguage();

event.on('languageChanged', that, that.setLanguage);

}

setLanguage() {

this.setData({

language: wx.T.getLanguage()

});

}

* + 1. 购物车模块
       1. 购物车功能

消费者在网络商城小程序上进行完注册登录之后，接下来就是消费者的购物功能的实现了。即网络商城小程序购物车功能的实现。再参考完当下的几个主流购物网站之后，我总结出了在设计购物车功能模块的时候需要注意的几点要求就是：（1）购物车的功能不要太复杂，界面应该尽量简洁。（2）在设计购物车模块的时候应该站在消费者的角度上考虑来进行设计工作。

在该套网络商城小程序的购物车设计上，应该达到消费者可以自由地方便地增减所要购买商品的件数，在商品结算的时候可以直接通过相关 API 端口直接进入微信支付界面。同时还可以存放有意向的商品暂时又不需要购买的商品。综上所述，我们应该设计简洁的购物车UI 界面再有一点就是在业务逻辑设计的时候应该多站在方便消费者购物体验的角度上分析设计。接下来我将针对这两个方面进行详细介绍。

在这套网络商城小程序购物车的UI 界面设计的时候，是通过上文介绍的mvc 的视图层来完成设计的，即在这一层的页面的 cart.wxml 和 cart.wxss 文件中添加与逻辑层的联系。在浏览商品时我们可以直接点击添加到购物车将意向商品加入到购物车内，同时我们还可以在购物车界面中实现商品数量的增减，这些功能的实现的处理是经过回调时间控制。这样下来我们就能得到一个界面简洁但内容有十分丰富的购物车 UI 界面[43]。

在这套网络商城小程序购物车的业务逻辑设计的时候，最重要的就是将产品的详细信息等数据应用到购物车数据信息当中，当购物车的数据信息发生变化的时候，为了保存购物车当中的商品数据需要添加本地数据缓存功能，在本地数据缓存中，存放购物车当中的商品信息。这些功能的实现主要是在 mvc 架构中的是业务逻辑层来实现的，具体的业务逻辑是在 cart.js 文件中加以控制。实现本网络商城小程序购物车模块功能的关键代码如下。

<import src="/template/quantity/index.wxml" />

<scroll-view class="scroll" scroll-y="true">

<view class="separate"></view>

<view wx:for="{{carts}}">

<view class="cart\_container">

<image class="item-select" bindtap="switchSelect" data-index="{{index}}" data-id="{{index}}"

src="{{item.isSelect?'../../images/cart/comment\_select.png':'../../images/cart/comment

\_normal.png'}}" />

<image class="item-image" src="{{item.pic}}"></image>

<view class="column">

<text class="title">{{item.name}}</text>

<view class="row">

<text class="sku-price">￥</text>

<text class="sku-price">{{item.price}}</text>

<view class="sku">

<template is="quantity" data="{{ ...item.count, componentId: index }}" > </view> </view>

</view> </view>

<view class="separate"></view>

</view>

</scroll-view>

<view class="bottom\_total">

<view class="bottom\_line"></view>

<view class="row">

<image class="item-allselect" bindtap="allSelect" src="{{isAllSelect?'../../images/cart/comment\_select.png':'../../images/cart/comment\_n ormal.png'}}" />

<text class="Mini \_text">全选</text>

<text>合计:￥ </text>

<text class="price">{{totalMoney}}</text>

<button class="button-red" bindtap="toBuy" formType="submit">去结算

</button>

</view>

</view>

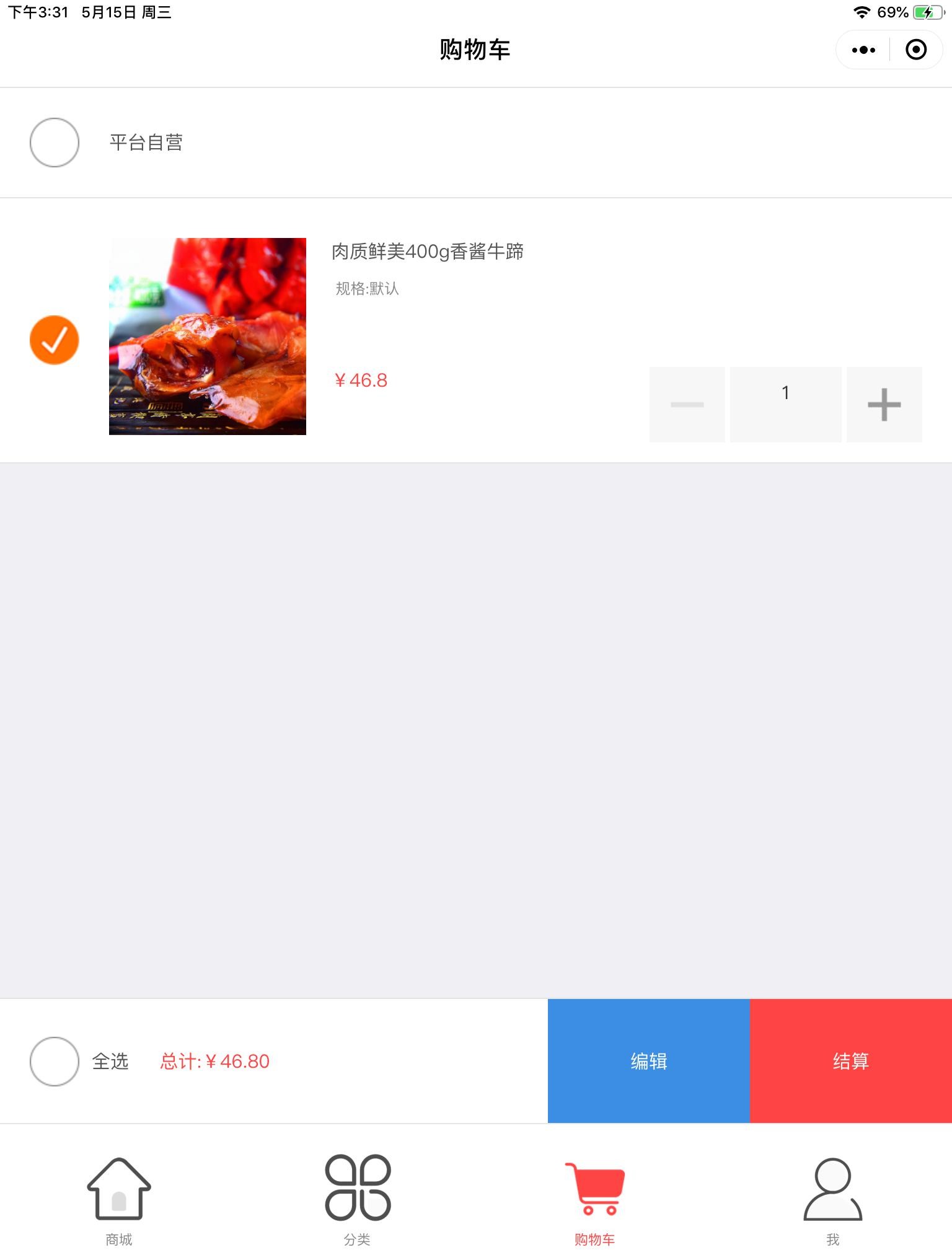
完成以上配置之后，网络商城小程序的购物车功能得以实现。实现的效果如图 5.7 示。

图 5.7 购物车实现效果

* + - 1. 商品收藏功能

消费者在这套网络商城小程序中浏览商品的过程中，商品收藏功能的实现是通过给商品设置一个按钮，按钮绑定一个点击事件，点击事件通过接口，在数据库或者缓存（建议使用 redis）中，存储商品的标识（id 或者其他）+用户的标识（id 或者其他）。用户查看“我的收藏”，也是通过接口，用自身的标识（id或者其他）去数据库或者缓存里查询到所有收藏的商品标识。同时取消收藏也是同一道理，通过接口，然后删除对应数据。实现该功能的关键代码如下。

data: {detailObj:{}, index:null, //是否收藏isCollected:false },

handleCollection(){

let isCollected = !this.data.isCollected

this.setData({//下面本来是这样子的:isCollected=isCollected,可以简写 isCollected })//提示用户

wx.showToast({

title: isCollected ? '收藏成功' : '取消收藏', icon:'success'

})}

完成以上配置，网络商城小程序的商品收藏功能就得以实现。实现效果如图

5.8 所示。

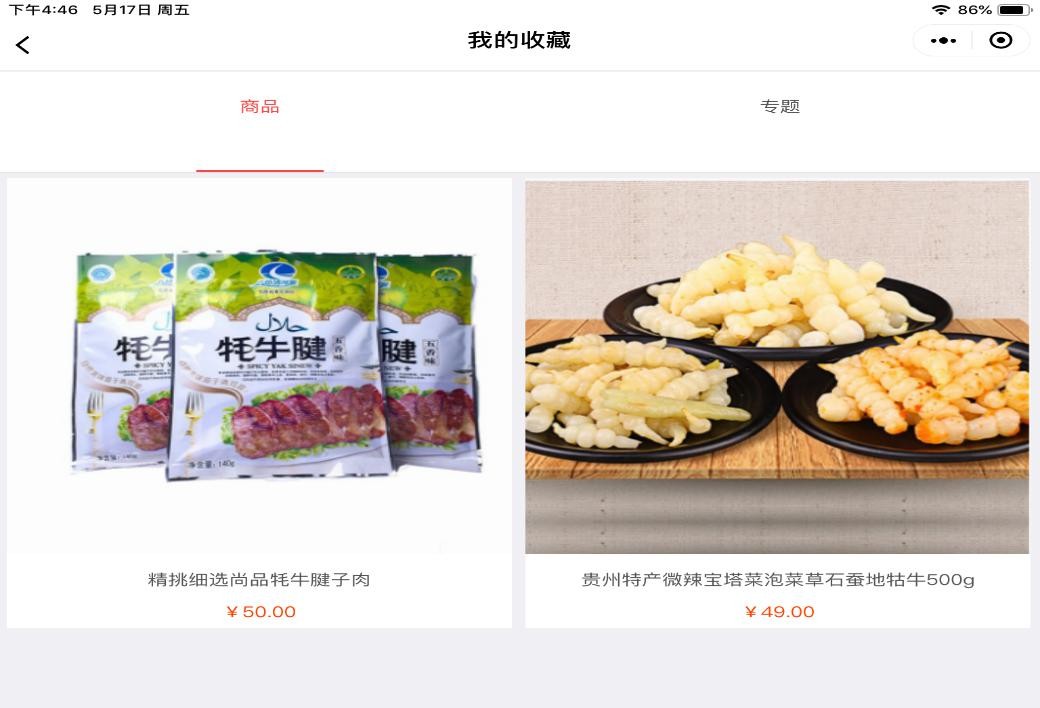


图 5.8 商品收藏功能效果图

* + 1. 订单结算模块

在本套网络商城小程序的使用时消费者应是在购物车页面可以发起生成订单请求，然后根据自己的现实需求来对购物车中的商品进行支付结算操作。

在订单结算的实现过程中大致分为以下几个步骤：1.提交购物车里的商品信息；2.点击支付之后系统调用微信支付模块，通过微信支付完成对商品的结算； 3.支付成功后，微信支付返回支付结果到消费者微信当中。主要的实现代码如下文所述。

//在进行订单支付操作的时候，首先是进行订单号的生成，接下来进行的才是商品支付结算。

Order.do Order（订单信息，（数据）=> {//调用库存监控的订单 API如果（data.pass）{//订单生成成功

Var id = data.order\_id; //更新订单状态

That.data.id = id;

That.\_exec Pay（id）; //通过库存检测通过，调用服务器预订单接口} 否则{that.\_order 失败（数据）; //如果下订单，则返回结果}}）。

同时在订单结算模块中还应该有消费者支付失败时，系统返回订单支付之前的状态，支付成功后确定订单收货地址是否正确。经过这一系列操作后订单结算的功能才得以完全实现。

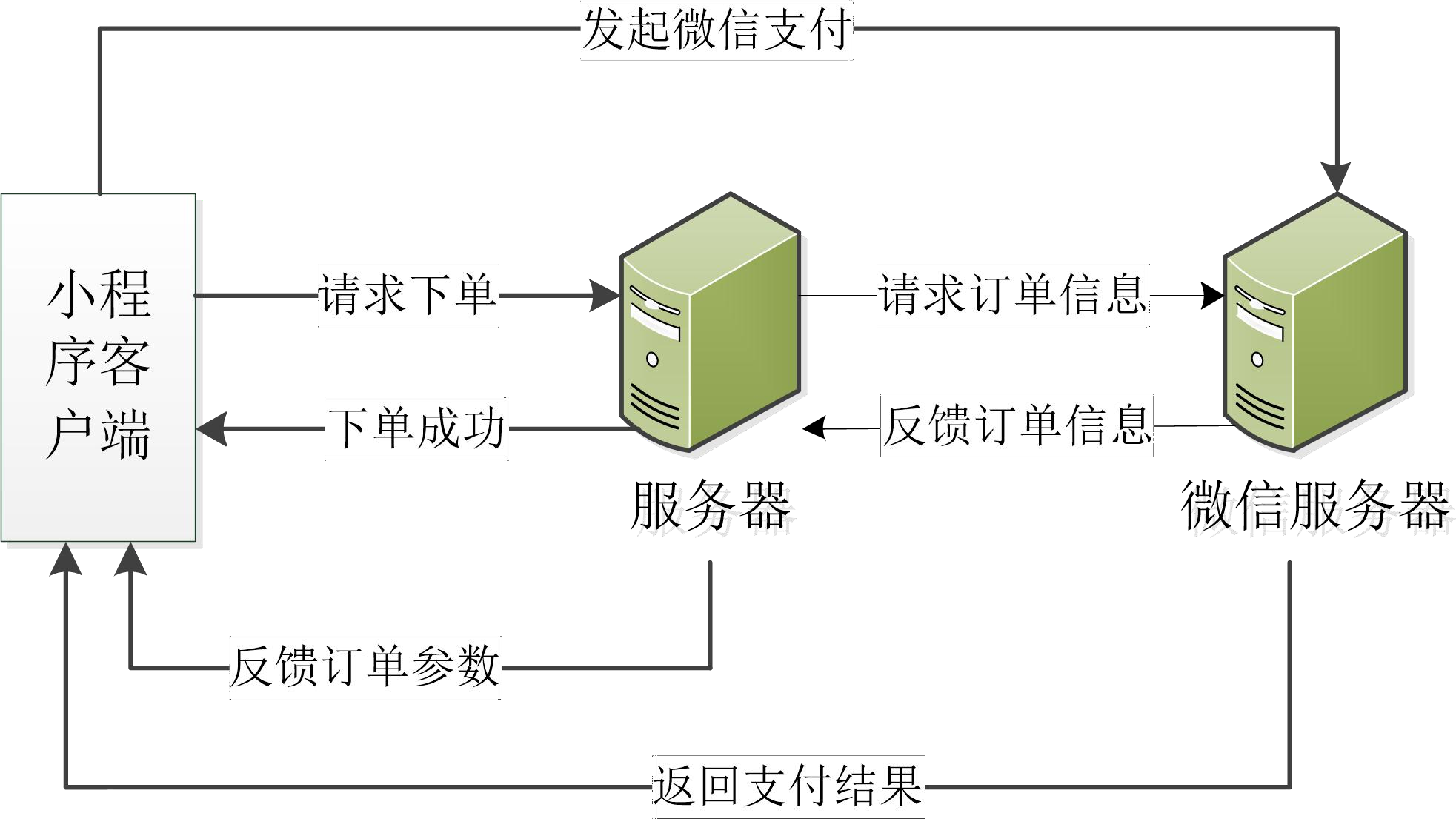
在网络商城小程序的订单结算模块是需要小程序客户端、服务器以及微信服务器的配合才能够实现的。消费者在小程序客户端的购物车中提出下单请求到服务器端，服务器端再将请求信息发送到微信服务器，之后微信服务器在通过相关API 的调用来完成支付任务，最后微信服务器再将订单支付的结果经由小程序客户端传达给消费者。具体的小程序客户端，开发人员服务器和支付微信服务器的业务逻辑关系如下图 5.9 所示。

图 5.9 下单支付的业务逻辑图

经上文所述，网络商城小程序订单结算模块的实现运行效果如下图 5.10。



图 5.10 订单结算模块的运行效果图

* + 1. 在线客服模块

消费者在这套网络商城微信小程序中的在线客服功能是通过接入微信平台提供的客服入口来实现，具体流程包括：1.消费者通过网络商城小程序中的在线客服入口进入在线客服聊天界面；2.消费者在聊天界面发送的文字、语音以及图片等消息会被打包成 json 形式的数据包；3.json 形式的数据包包被发送到我们开发者填写的 URL 中；4.系统设置的客服管理人员接收到上述数据后通过调用相关接口进行回复。

实现该功能的关键代码如下。

// 通过组件方式

<contact-button type="default-light" size="20"

session-from="weapp">

</contact-button>

// 通过按钮方式

<button open-type="contact">进入客服会话</button>

完成以上配置之后，该功能的实现效果如图 5.11 所示。



图 5.11 在线客服功能实现效果图

* + 1. 用户信息管理模块

用户信息管理主要是消费者对个人账户内收货地址的管理，包括对收货地址的添加、修改以及删除等操作。实现这一功能的具体要求：1.设计收货地址以及新增收货地址的 UI 界面，这里采用的是创建 wxml 文件；2.用户填写收货地址分为省、市、县（区）三个级别；3.收货地址的存储管理是通过微信平台提供的数据缓存来实现的。

为了实现对收货地址列表的管理，本文采用的是如果收货地址发生添加、修改、删除等操作的时候，客户端将 request 请求发送至系统管理员，微信接口获取到要修改的信息后，通过 ajax 提交后台，后台获取到信息根据用户请求，对数据库进行增、删、改、查，更新用户个人信息表。成功后返回成功信息，前端[回调函数](https://www.baidu.com/s?wd=%E5%9B%9E%E8%B0%83%E5%87%BD%E6%95%B0&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)得到后台返回的成功信息后，对页面进行数据更新操作[45]。将更新后的个人信息保存至服务端数据库中。

## 后台功能模块的实现

* + 1. 商品管理模块

在网络商城小程序中商家可将在售的商品资源，根据产地进行分类结果。在商品管理设计过程中，小程序客户端商品信息的呈现与后台数据库之间是通过

Ajax 技术来实现数据交互的。后台的商品管理模块可以实现对商品详情、商品分类以及商品库存信息等的管理。在后台数据库中的数据信息不知道第三方服务器上，然后通过微信服务器相关 API 接口连接到第三方服务器。这样就可以实现商品管理模块中的相关数据信息与小程序客户端的联系。从而将商品管理模块的相关功能呈现到小程序前端页面[45]。实现这一功能的关键步骤是：1.通过 WXML 将需要提交的商品详情的数据信息进行编辑；2.通过微信 image 标签将商品详情中的图片进行上传；3.通过 JavaScript 文件以及相关 API 接口，将要上传的数据信息上传至服务器。

* + 1. 仓库管理模块

我们设计的网络商城微信小程序在进行数据库管理的时候，系统管理员要及时有效地在后台更新商品的数据信息。为了保证数据库能够正常的运行，并且不会产生问题。在进行上述过程的同时为了达到数据库的规范化，及时修改那些已经作废的数据或者不规范的数据。当商品信息发生增、删、改的时候，客户端将request 请求发送至系统管理员，微信接口获取到要修改的信息后，通过 ajax 提交后台，后台获取到信息根据用户请求，对数据库进行增、删、改，更新商品信息表。成功后返回成功信息，前端[回调函数](https://www.baidu.com/s?wd=%E5%9B%9E%E8%B0%83%E5%87%BD%E6%95%B0&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)得到后台返回的成功信息后，对页面进行数据更新操作。将更新后的商品信息保存至服务端数据库中。

* + 1. 系统管理模块

系统管理模块功能的实现是为网络商城小程序设置系统管理员。系统管理员可以被授予相应的权限来对系统的商品信息进行维护，更好地保证网络商城正常工作。主要实现的功能包括：1.可以查看普通用户的信息，并且有权限删除普通用户信息。2、对产品管理（比如某个产品下架（删除），产品价格的更改等管理），增删改查。3、后台订单管理等。其中管理商品资源，对商品进行分类管理，审核商品的库存信息，可通过商品产地及商品名进行查询，及时掌握所有商品的相关情况，保障商品资源的可靠性。并且对商品信息进行增、删、查的操作，保证网络商城小程序内商品的丰富性。通过系统管理模快可以更好地提高消费者消费者的购物体验以及提供给消费者一个安全、可靠的购物环境。对商品的具体操作如图 5.12 所示。

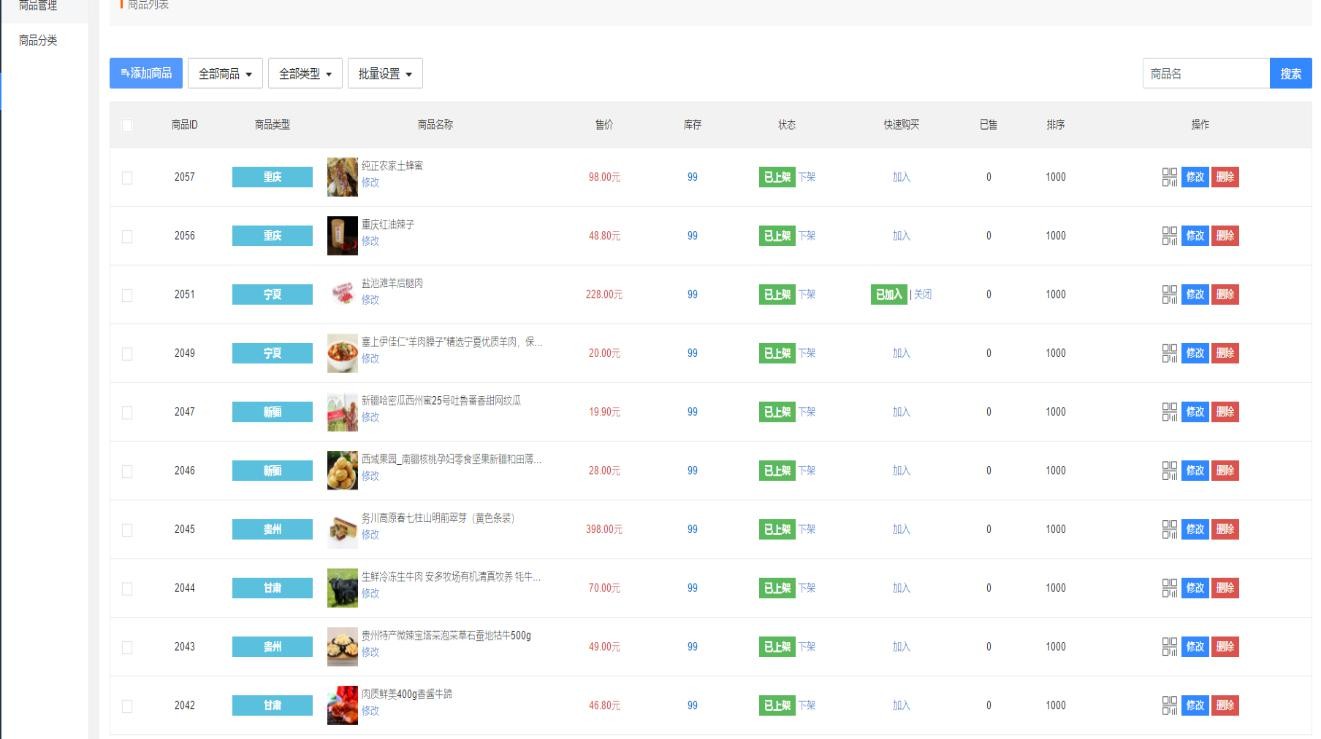


图 5.12 后台商品操作效果图

# 第 6 章 系统测试

## 测试方法

微信小程序是微信平台近两年推出新型产品，因此在测试内容上也与传统app 有所不同[46]。本文对小程序的测试主要分为 4 个方面，即功能测试、兼容性测试、性能测试、后台接口测试。对于安全性测试由于小程序集成在微信客户端内，相比于传统的网页来说安全性能够更有保障。只要在后台接口测试上保证数据的安全性，客户端的安全性由微信 app 来替我们保证[47]。

因此本次系统测试主要分为以下四个方面的内容：1.系统功能测试；2.系统兼容性测试（包括：操作系统兼容性、屏幕兼容性以及微信版本兼容性等）；3.系统性能测试（包括：页面的白屏时间，首屏时间，资源占用，页面渲染时间， 帧率等）。

## 系统测试环境

系统测试是完成系统后最重要的一个环节，在这里我们根据 6.1 节的测试方法，来进行微信小程序测试环境的搭建。本网络商城系统是基于微信小程序开发的，测试入口应该为微信应用。考虑到各平台的兼容性等，本次将分别测试安卓系统与苹果系统。详细硬件设备信息如下所示。

安卓系统：小米10，MIUI12，微信版本 Version8.0.9，CPU 为骁龙 835，RAM 为 4GB。

苹果系统：iphone12Max，iOS12.1.4，微信版本 Version8.0.9，CPU 为 A14，RAM

为 6GB。

电脑：MacBookPro 13，系统为 Mac OS，内存大小为 16GB，浏览器为：谷歌浏览器， CPU 为因特尔酷睿 i7。

## 测试结果

* + 1. 系统功能测试结果

下面主要对系统的发送注册登录、商品浏览、购物车、订单支付和地址管理等功能进行功能性、安全性和稳定性进行测试[47]。以下是各个功能的测试情况。

* + - 1. 基于微信小程序的网络商城的注册登录功能测试详情如表 6.1 所示。

表 6.1 注册登录功能测试

测试项目 基于微信小程序的网络商城的注册登录功能测试

目的 测试用户打开小程序时微信登陆和微信授权的功能。

前提 用户初次打开小程序。

测试流程 获取小程序，进入“我的”页面；弹出微信授权获取公开信息页面，选择允许或拒绝。

测试结果 当用户选择允许时，个人信息显示用户的微信头像和昵称； 当用户拒绝时，显示系统默认的头像和昵称。

* + - 1. 基于微信小程序的网络商城的商品浏览功能测试详情如表 6.2 所示。

表 6.2 商品浏览功能测试

测试项目 基于微信小程序的网络商城的商品浏览功能 目的 测试用户打开小程序时浏览商品时是否流畅。

测试流程 进入小程序，浏览商品。

测试结果 界面运行流畅。

* + - 1. 基于微信小程序的网络商城的购物车测试如表 6.3 所示。

表 6.3 购物车测试

测试项目 基于微信小程序的网络商城的购物车测试目的 测试购物车功能是否实现。

前提 商品有无库存的两种情况都要存在。 测试流程 用户浏览商品，并且添加到购物车中。测试结果 购物车功能均能实现。

* + - 1. 基于微信小程序的网络商城的订单结算测如表 6.4 所示。

表 6.4 订单支付测试

测试项目 基于微信小程序的网络商城的订单支付测试目的 测试订单结算功能。

前提 支付成功且商品有库存。

测试流程 用户成功支付购物车里的订单。

测试结果 支付成功后，微信发送支付成功信息到用户微信账户中；支付失败后，系统显示支付失败字样，同时返回到订单详情页面当中。

* + - 1. 基于微信小程序的网络商城的地址管理功能测试详情如表 6.5 所示。

表 6.5 地址管理功能测试

测试项目 基于微信小程序的网络商城的地址管理功能测试目的 测试地址管理功能。

前提 进入网络商城小程序。

测试流程 在我的页面点击“地址管理”；点击新增地址。

测试结果 跳转到地址管理页面，可以接受微信授权获取当前卫星定位信息；点击新增地址可按照规范新增收货信息。地址编辑完毕后保存，订单详情页面就可以实时定位当前收货地址信息[50]。

* + 1. 系统兼容性测试结果

系统的兼容性测试结果如下所示：

测试内容 结果

操作系统兼容性 兼容

屏幕兼容性 兼容

微信版本兼容性 兼容

* + 1. 系统性能测试结果

系统的性能测试结果如下所示：

|  |  |
| --- | --- |
| 测试内容 | 结果 |
| 页面的白屏时间 | 0.5 秒 |
| 首屏时间 | 1 秒 |
| 资源占用 | 2MB |
| 页面渲染时间 | 1 秒 |
| 帧率 | 1 帧/秒 |

# 第 7 章 总结与展望

## 工作总结

本文经过对基于微信小程序的网络商城详细分析，完成了这套基于微信小程序的网上购物系统。本文的主要研究工作包括以下几个方面的内容：1.通过对“民族特色农产品多语言网络交易展示平台关键技术集成与应用示范”这一项目的成果进行了研究分析，并且从中梳理、归纳了适用于网络商城微信小程序开发的相关数据信息。以及分析了当前微信小程序的发展现状，从而得到了本文所研究内容的现实意义；2.结合小程序的开发，整理总结了当今微信小程序开发过程中所应用到的微信 web 开发者工具、微信小程序开发组件、微信小程序开发 API、HTML+CSS+JavaScript、PHP 开发语言等相关理论与技术；3.从系统功能需求、系统性能需求以及系统可行性分析等方面完成了对网络商城小程序的需求分析； 4.从系统的架构、系统的逻辑结构以及结合微信小程序开发的特点，完成了网络商城小程序的系统设计与实现等章节的内容，实现了小程序各项前端功能和后台功能；5.通过对系统功能测试、系统兼容性测试、系统性能测试等方面的工作， 以及分析各项测试的结果，基本验证了网络商城小程序的各项功能达到了需求分析和系统设计中相关要求，完成网络商城小程序系统的测试工作，保证了系统可以被用户正常地使用。从而形成本篇以绪论、相关技术与理论、需求分析、系统设计、系统实现、系统测试及本章的总结与展望等七大章节构成的毕业论文，进行由浅及深，从理论、技术到实现、测试的逻辑思路进行具体的描述。

在完成本篇论文的过程中，我严格按照软件开发流程是按需求分析、系统设计、系统实现、系统测试的步骤进行的。并且在开发这套网络商城小程序的过程中严格遵这些步骤，最终实现了这套功能完善小程序系统，并且保证了系统的平稳运行。

## 未来展望

由于时间有限，本套网络商城微信小程序系统的设计虽然已经满足了推广民族特色农产品的基本需求，但是依然存在这一些需要增进的地方。在后续的过程中，主要应着眼于维护系统的稳定性，完善后续系统内容，提出一些系统改进方案。

* + 1. 基于微信小程序的网络商城系统的客户端和服务器等方面相对于目前主流网络商城小程序来说功能还不够完善，后期应当对此进行改进和优化。参考现阶段流行的即时通讯系统，对本系统中的一些细节性的功能加以完善，以达到

更好的用户体验。

* + 1. 本系统并没有对同时多用户并发使用进行测试。接下来应该对此问题进行深入的分析和研究，找到提升系统性能的办法。
    2. 本系统的 UI 页面还需要进一步优化，使其更加美观。

# 参考文献

1. 张迪,黄森,朱珺,朱烨.基于微信小程序的第二课堂学分系统设计研究[J].电脑知识与技 术,2018,14(36):54-57+71.
2. 刘文元.Web API 中的 GraphQL[J].电脑迷,2019(01):107.
3. 李洁.主流 JavaScript 框架——AngularJS、React 和 Vue 使用体会[J].电脑迷,2019(01):77.
4. 黄小华,傅永华,赵莉,郑丹,姚思奇.浅析题库类微信小程序设计及应用[J].信息技术与 信息化,2018(12):51-53.

[5]王婷婷.微信小程序开发[J].信息技术与信息化,2018(12):62-63.

[6]彭守镇.微信小程序应用探究[J].信息与电脑(理论版),2018(22):22-23.

1. 周浩,赵修彦.基于 ArcGIS JavaScript API 的物业全覆盖管理信息系统设计与实现[J]. 测绘与空间地理信息,2018,41(11):160-162.
2. 劳诗尧,胡必波,宋冠相.MVC 设计模式在互联网+农副产品创新电商平台应用研究[J].电脑编程技巧与维护,2018(11):136-138.
3. Huifang Wang. Effect of Psychological Intervention of King's Interactive Standard on the Rehabilitation Process and Nursing Satisfaction of Patients with Severe Acute Pancreatitis in ICU[A]. Institute of Management Science and Industrial Engineering.Proceedings of 2018 3rd International Conference on Life Sciences,Medicine,and Health(ICLSMH 2018)[C].Institute of Management Science and Industrial Engineering:计算机科学与电子技术国际学会(Computer Science and Electronic Technology International Society),2018:5.
4. 方红亮,解颐,许宁.基于 JavaScript 的藏汉双语内容分享 WebAPP 设计与开发[J].电脑编程技巧与维护,2018(08):81-82+85.
5. Zheng Liu. A Study of Cockpit HMI Simulation Design Based on the Concept of MVC Design Pattern[A]. Advanced Science and Industry Research Center.Proceedings of 2018 3rd International Conference on Modelling, Simulation and Applied Mathematics（MSAM 2018） [C].Advanced Science and Industry Research Center:Science and Engineering Research Center,2018:3.
6. 洪世勇.新时期网页设计中计算机图像处理技术的应用[J].信息与电脑(理论 版),2018(11):171-172+175.
7. 田俊峰. 多语言文本语义相似度的计算及其应用研究[D].华东师范大学,2018. [14]刘正,张书锋,赵鹤鸣.MVC 模式下多层分布式软件系统架构设计[J].现代电子技术,2018,41(07):135-139+146.

[15]张亚召. 基于行列式点过程的多语言多文档摘要研究[D].北京邮电大学,2018.

# 致谢

时间如白驹过隙，转瞬即逝。转眼之间。我即将离开这所给了我两年美好时光的北京邮电大学，去面对社会的考验，步入人生的另一个阶段。

完成本论文撰写之际，我要特别感谢母校北京邮电大学，谢谢您汇集了如此之多学识渊博的老师，配备了优秀的教学设施以及良好的生活设施，让我接收了良好的教育和丰富的研究生生活。今后步入社会，我将永远以母校为荣，也会将您的教育铭记于心，努力工作，做一个对社会有贡献的人。