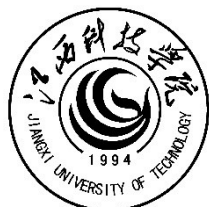


密级：

学号：2017007890



江西科技學院

## 本科生毕业论文（设计）

### 基于微信小程序的电商平台的设计与实现

学 院： 信息工程学院

专 业： 网络工程

班 级： 18 本网络 1 班

学 号： 2017007890 姓 名： 符锦翰

指导教师 1： 饶苏敏 指导教师 2：

## 学士学位论文原创性声明

本人郑重声明：所呈交的论文（设计）是本人在指导老师的指导下独立进行研究，所取得的研究成果。除了文中特别加以标注引用的内容外，本论文（设计）不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式表明。本人完全意识到本声明的法律后果由本人承担。

学位论文作者签名（手写）：黄锦翰 签字日期：2024 年 4 月 26 日

## 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权江西科技学院可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 ☐ 保 密 ☐，在 年解密后适用本授权书。  
☒ 不保密 ☒.

（请在以上相应方框内打“√”）

学位论文作者签名（手写）：黄锦翰 指导老师签名（手写）：陈强

签字日期：2024 年 4 月 26 日 签字日期：2024 年 4 月 26 日

## 摘要

随着移动互联网的蓬勃发展，电子商务已经成为现代社会不可或缺的一部分。传统电商平台通常基于网页或独立应用程序，但随着微信小程序的兴起，基于微信生态系统的电商平台成为新趋势。微信小程序作为轻量级应用，无需下载安装，大大节省了时间和空间。其维护成本较低，开发者可通过微信开放平台轻松开发和发布。

本研究通过对微信小程序的应用进行分析，针对电子商务领域，规划并开发了一款功能丰富、用户友好的电商平台。在实际开发过程中，通过接口文档的深入分析，成功实现了核心功能，该系统的主要功能模块包括首页展示、商品详情、规格选择、授权登录、购物车、地址管理、提交订单、订单支付、订单管理。采用了 Vue3+TypeScript 和 UniApp 框架，迅速搭建了整体架构，为用户提供了便捷的购物体验。

本研究的目标在于提供一个优质的电商平台，通过微信小程序的便利性和微信生态系统的支持，为用户提供丰富的社交和支付功能，从而满足用户的购物需求。未来，将继续扩展和优化平台，提升用户体验，以适应电子商务市场不断变化的需求。

**关键词：**电商平台；微信小程序；Vue3；TypeScript；UniApp

## Abstract

With the booming development of mobile internet, e-commerce has become an integral part of modern society. Traditional e-commerce platforms are usually based on web pages or standalone apps, but with the rise of WeChat apps, e-commerce platforms based on the WeChat ecosystem have become a new trend. WeChat applets, as lightweight applications, do not need to be downloaded and installed, which greatly saves time and space. Its maintenance cost is low, and developers can easily develop and distribute it through the WeChat open platform.

By analysing the application of WeChat applets, this study plans and develops a feature-rich and user-friendly e-commerce platform for the field of e-commerce. In the actual development process, the core functions were successfully implemented through the in-depth analysis of the interface documents. The main functional modules of the system include homepage display, product details, specification selection, authorised login, shopping cart, address management, order submission, order payment and order management. Vue3+TypeScript and UniApp framework were used to quickly build the overall architecture and provide users with a convenient shopping experience.

The goal of this study is to provide a high-quality e-commerce platform that meets the shopping needs of users by offering rich social and payment features through the convenience of WeChat apps and the support of the WeChat ecosystem. In the future, the platform will continue to be expanded and optimised to enhance the user experience to meet the changing needs of the e-commerce market.

**Key Words:** e-commerce platform; WeChat applets;Vue3; TypeScript;UniApp

## 目录

<b>第 1 章 引言</b>	<b>1</b>
1.1 选题背景及意义	1
1.1.1 选题背景	1
1.1.2 选题意义	1
1.2 国内外发展现状	2
1.2.1 国外发展现状	2
1.2.2 国内发展现状	2
1.3 论文研究目标和内容	3
1.4 论文结构安排	3
1.5 本章小结	4
<b>第 2 章 系统需求分析</b>	<b>5</b>
2.1 系统功能需求分析	5
2.1.1 用户需求分析	5
2.1.2 管理员需求分析	6
2.1.3 用户角色管理分析	7
2.2 系统非功能需求分析	7
2.3 本章小结	8
<b>第 3 章 系统设计</b>	<b>9</b>
3.1 系统总体设计	9
3.1.1 系统架构设计	9
3.2 系统功能模块设计	9
3.2.1 功能模块设计	9
3.3 数据库设计	13
3.3.1 数据库 E-R 模型设计	13
3.3.2 实体属性设计	14
3.3.3 数据库详细表设计	16

3.4 本章小结.....	25
<b>第 4 章 系统开发与实现 .....</b>	<b>26</b>
4.1 系统技术选型.....	26
4.2 开发环境建立.....	26
4.3 核心功能模块的实现.....	27
4.3.1 首页自定义导航栏设计.....	27
4.3.2 推荐模块设计.....	28
4.3.3 商品分类设计.....	29
4.3.4 商品详情设计.....	31
4.3.5 购物车页面设计.....	32
4.3.6 订单页面设计.....	33
4.3.7 订单状态列表实现.....	34
4.3.8 登录界面的实现.....	35
4.3.9 用户界面实现.....	37
4.4 本章小结.....	38
<b>第 5 章 测试.....</b>	<b>39</b>
5.1 系统功能测试.....	39
5.1.1 系统功能测试.....	39
5.1.2 商品模块测试.....	40
5.1.3 用户登录模块测试.....	40
5.1.4 用户购物车模块测试.....	41
5.2 系统非功能测试.....	42
5.3 本章小结.....	43
<b>第 6 章 总结与展望 .....</b>	<b>44</b>
6.1 总结.....	44
6.2 展望.....	44
<b>参考文献 .....</b>	<b>45</b>
<b>致谢.....</b>	<b>46</b>

# 第1章 引言

## 1.1 选题背景及意义

### 1.1.1 选题背景

随着移动互联网的迅速发展和智能手机的普及，电子商务行业蓬勃发展。微信小程序作为一种新型的移动应用形式，其轻量级、便捷性和强大的社交属性使其成为电商平台的重要入口之一。然而，当前微信小程序电商平台的开发仍面临着一些困扰。

首先，微信小程序平台的开发和运营需要考虑到不同平台的兼容性和特性差异，这给开发者带来了一定的挑战。其次，微信小程序在用户体验和功能扩展方面存在一些局限，例如对于微信登录的支持不足以及滚动驱动动画等功能受限。再者，电商平台在面对用户规模庞大的同时，需要保证系统的性能稳定和加载速度快，而在微信小程序开发中，性能问题常常成为开发者头疼的难题。

为解决这些问题，本论文旨在基于微信小程序平台设计和实现一款高性能、用户体验友好的电商平台，通过针对性的技术和策略来提升小程序电商平台的开发效率和用户体验。

本论文的研究和实现将促进微信小程序电商平台的发展和完善，为电商行业提供了更加高效、便捷的解决方案。通过解决微信小程序电商平台开发中的兼容性、功能扩展和性能优化等问题，提高了开发效率和用户体验，为用户提供了更加优质的购物体验。同时，优化后的小程序电商平台将提高系统的稳定性和加载速度，从而提升了电商平台的竞争力和用户满意度，为电商企业带来了更多的商业价值和利润增长。

### 1.1.2 选题意义

解决了微信小程序电商平台开发中面临的兼容性、功能扩展和性能优化等问题，为电商行业提供了更加高效、便捷的解决方案，推动了微信小程序电商平台的发展和完善。

提高了开发效率和用户体验，为用户提供了更加优质的购物体验，促进了电商行业的发展和用户满意度。

通过优化小程序电商平台的性能和功能，提升了系统的稳定性和加载速度，从而提高了电商平台的竞争力和商业价值，为电商企业带来了更多的利润增长。

和市场份额。

本论文的研究和实现将为相关领域的研究者和开发者提供参考和借鉴，促进了技术的交流和共享，培养了社会责任感和创新意识，推动了信息技术的应用和发展。

## 1.2 国内外发展现状

### 1.2.1 国外发展现状

在国外，电子商务行业已经取得了长足的发展，并且电商平台基于移动应用的趋势也日益明显。在微信小程序出现之前，国外已经存在着一些类似的移动应用电商平台，例如美国的 Amazon、eBay，以及中国的淘宝国际版等。这些平台在用户体验、功能丰富性、技术支持等方面都已经较为成熟，为用户提供了便捷的购物体验和丰富的商品选择。

在移动应用方面，国外的电商平台多采用原生应用开发或跨平台开发技术，例如使用 React Native、Flutter 等技术进行开发<sup>[1]</sup>。这些平台通常拥有强大的技术团队和丰富的开发经验，能够充分利用移动端的硬件资源和操作系统特性，为用户提供流畅、稳定的购物体验。

国外的电商平台非常重视用户数据的分析和挖掘。它们利用先进的大数据技术和人工智能算法对用户行为进行深入分析和预测，以实现个性化的推荐和服务。这种个性化的服务能够显著提升用户的黏性，促进用户与平台的互动，进而提高购买转化率。

### 1.2.2 国内发展现状

在国内，随着移动互联网的普及和电子商务的快速发展，电商行业呈现出蓬勃的发展态势。中国的电商巨头如阿里巴巴集团、京东集团等在国内电商市场占据主导地位<sup>[2]</sup>，其电商平台在用户规模、交易量、技术支持等方面都达到了国际先进水平。

而微信小程序作为一种新兴的移动应用形式，也逐渐在国内电商领域崭露头角。越来越多的电商企业和个体工商户开始将重点放在微信小程序上，通过微信生态圈的流量入口和社交属性，吸引用户进行购物消费。国内的微信小程序电商平台在功能和用户体验上不断创新和完善，逐渐成为电商行业的重要组成部分。

此外，国内的微信小程序电商平台也积极探索技术创新和商业模式创新，例如采用云开发技术、智能推荐算法等，为用户提供更个性化、智能化的购物



体验。同时，国内的电商平台也注重社交化营销和用户互动，通过社交分享、短视频营销等方式吸引用户参与和传播，提高用户参与度和购买转化率。

综上所述，国内外电商平台在移动应用方面都呈现出不断创新和发展的态势，微信小程序作为一种新兴的移动应用形式，正在逐渐成为电商行业的主流趋势。

### 1.3 论文研究目标和内容

#### （1）本研究的目标

是设计和实现一个基于微信小程序的电商平台，旨在解决当前微信小程序电商平台开发中存在的一系列问题和需求。通过采用先进的前端技术（Vue3+TypeScript 和 UniApp 框架）和后端技术（unicloud 云开发及云数据库），结合前端后端一体化的开发方式，实现了高性能、用户体验友好的电商平台。

#### （2）本系统的核心功能模块

主要包括以下部分，首页展示、商品详情、规格选择、登录授权、购物车、地址管理、订单提交、订单支付、订单管理<sup>[3]</sup>。

通过以上功能模块的设计和实现，本系统旨在解决用户在微信小程序电商平台使用过程中遇到的痛点和需求，提升用户体验，促进电商行业的发展和进步。同时，采用 Vue3+TypeScript 和 UniApp 框架以及 Uniclass 云开发技术，使得系统具备了高性能、易维护和可扩展性的特点，为电商平台的长期发展奠定了良好的基础。

### 1.4 论文结构安排

#### （1）论文基础架构

首先由用户需求分析部分将对电商平台的用户群体和需求进行深入分析，包括用户的购物习惯、偏好、行为特点等方面。通过对用户需求的分析，为后续系统设计和功能实现提供指导和依据。

其次是系统设计部分将详细介绍基于微信小程序的电商平台的整体设计方案，包括系统架构、模块设计、数据库设计等方面。在这一部分中，将对系统的功能模块和技术实现方案进行详细阐述，以确保系统能够满足用户需求并具备良好的性能和扩展性。

再者是系统实现与开发部分将详细描述基于微信小程序的电商平台的具体实现过程，包括前端界面设计、后端逻辑开发、数据库搭建等方面。在这一部分中，将介绍所采用的开发工具、技术栈和开发流程，以及遇到的问题和解决方法。

最后是进行功能测试、性能测试和安全测试等方面的评估和验证。通过测试，发现系统存在的问题和不足之处，并进行相应的优化和改进，以确保系统的稳定性和可靠性

## （2）vue 架构

开发环境的搭建，使用操作系统 Windows 10 64 位，Node.js v20.11.1，pnpm 版本 8.15.4，开发语言 TypeScript，并且作为主要的开发框架使用 Vue.js、UniApp 以及 uncloud。开发工具包括微信开发者工具、VS Code 以及 HbuilderX。

HbuilderX 与 Vue3：HbuilderX 专注于前端开发，尤其是移动端应用的开发。为了在 HbuilderX 中支持 Vue3 项目的开发，安装 Vue3 编译插件，以确保在项目正确编译和运行 Vue3 代码。

VS Code 插件配置：为了提高开发效率，可安装 uni-app 插件、uni-helper，以及专门为 Vue3 设计的语言服务插件 Volar，它提供代码智能提示、语法高亮与自动补全等功能。

电子商务平台案例：介绍如何使用 Vue3+TypeScript 和 UniApp 框架快速搭建电子商务平台的整体架构，并提供便捷的购物体验。通过微信小程序的便利性以及微信生态系统的支持，为用户提供丰富的社交和支付功能。

代码风格和错误检测：采用 ESLint 作为 JavaScript 代码检查工具，Prettier 作为代码格式化工具，这有助于团队统一代码风格和发现代码问题。

## 1.5 本章小结

首先，从选题背景和选题意义的角度出发，探讨了基于微信小程序的电商平台设计与实现的重要性和必要性。随着移动互联网的快速发展，微信小程序作为一种新型的移动应用形式，为电商平台的发展提供了新的机遇和挑战。

接着，分析了国内外电商平台发展的现状。在国外，电商平台已经得到了广泛的应用和发展，各大电商巨头通过不断创新和技术提升，提升了用户体验和平台功能。而在国内，微信小程序电商平台也逐渐崭露头角，但仍面临着一些挑战和不足之处。

最后，介绍了论文的结构和架构。论文主要分为摘要、引言、总结和四个主要章节：用户需求分析、系统设计、系统实现与开发、系统测试。

通过本章的阐述，读者清晰地了解论文的整体框架和内容安排，为后续的阅读和研究提供了指导和指引。

## 第2章 系统需求分析

### 2.1 系统功能需求分析

#### 2.1.1 用户需求分析

用户需求分析是系统设计的基础，通过深入了解用户的需求和期望，确保系统功能的准确性和实用性。在设计基于微信小程序的电商平台时，需要考虑到不同用户的需求，以提供个性化的用户体验。

##### （1）用户注册与登录

用户希望能够方便快捷地注册和登录平台，以便享受购物、下单等功能。因此，系统应该提供简洁明了的注册流程，同时支持使用微信账号进行登录。

##### （2）商品浏览与搜索

商品浏览与搜索，用户需要在平台上方便地查看商品信息，并能够通过简单快捷的搜索功能迅速找到所需商品。为此，系统应提供清晰直观的商品展示页面，并支持关键词搜索和筛选功能。

##### （3）商品详情查看

用户希望能够查看商品的详细信息，包括商品描述、价格、规格参数等。因此，系统应该提供商品详情页，展示商品的全部信息，并允许用户查看商品图片和相关评价<sup>[4]</sup>。

##### （4）购物车管理

用户需要能够方便地管理购物车中的商品，包括添加、删除、修改商品数量等操作。因此，系统应该提供购物车页面<sup>[5]</sup>，显示用户已选商品列表，并支持实时更新购物车内容。

##### （5）订单提交与支付

用户希望能够顺利提交订单并选择合适的支付方式进行支付。因此，系统应该提供订单提交页面，让用户填写收货信息并选择支付方式，同时支持多种支付方式的接入。通过对用户需求的分析<sup>[6]</sup>，确定系统需要具备的核心功能，以及用户期望的使用体验，从而指导后续系统设计和开发工作。

以下是用户需求分析具体用例如图 2.1:

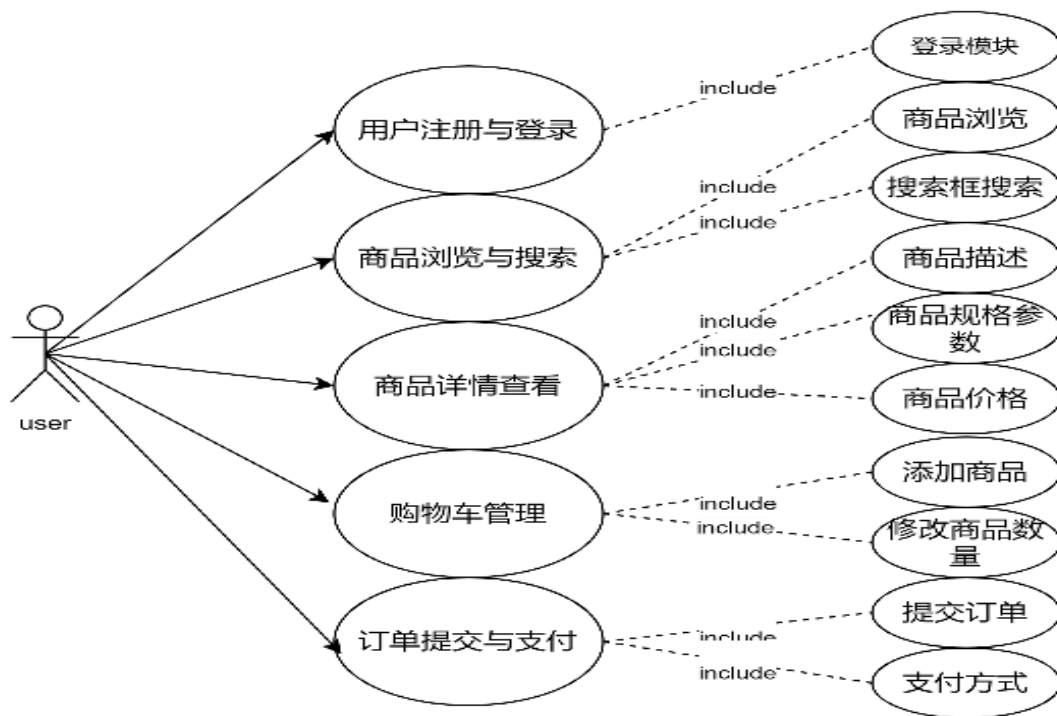


图 2.1 用户需求分析用例图

### 2.1.2 管理员需求分析

管理员是电商平台的重要角色，他们负责管理平台上的商品、订单、用户信息等核心内容。因此，管理员需要一系列功能来支持他们有效地管理电商平台。

以下是管理员需求分析用例图 2.2:

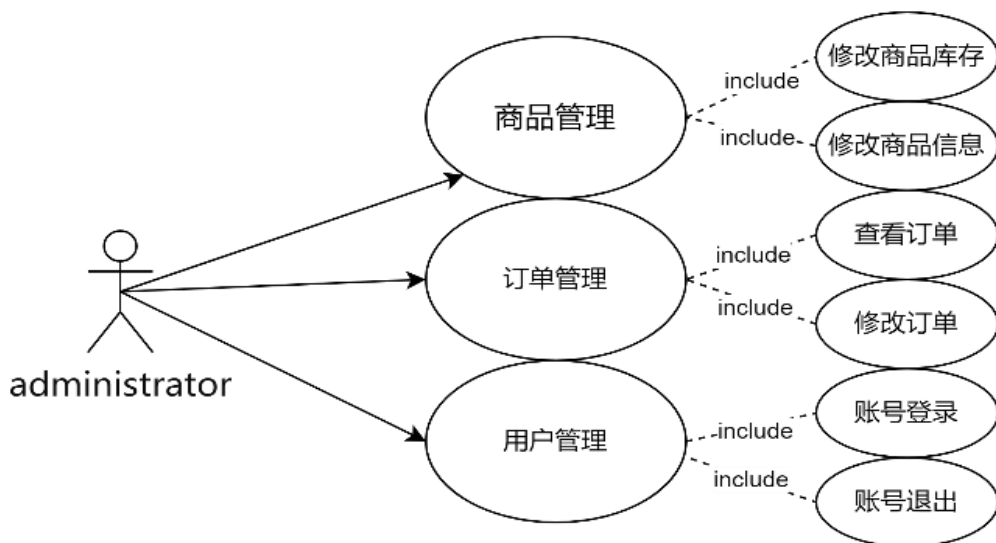


图 2.2 管理员需求分析用例图

商品管理功能，管理员具有添加商品的权限，新增商品信息，包括商品名称、价格、库存数量、详细描述等内容。编辑商品信息，管理员对已有的商品信息进行编辑和更新，包括价格调整、库存修改等<sup>[7]</sup>。商品管理，管理员有权删除不再销售的商品，以确保商品列表的及时更新。订单管理功能，查看订单列表，管理员查看所有用户的订单列表，包括订单状态、订单金额、下单时间等信息。处理订单，管理员对订单进行处理，包括确认订单、发货、取消订单等操作<sup>[8]</sup>。

### 2.1.3 用户角色管理分析

在电商平台的设计与实现中，用户角色管理是一个至关重要的功能模块，它旨在管理和区分不同类型的用户，并为每种用户角色提供相应的功能和权限。通过细致而有效地管理用户角色，提高系统的安全性、灵活性和用户体验。

以下是用户角色管理需求分析用例图 2.3：

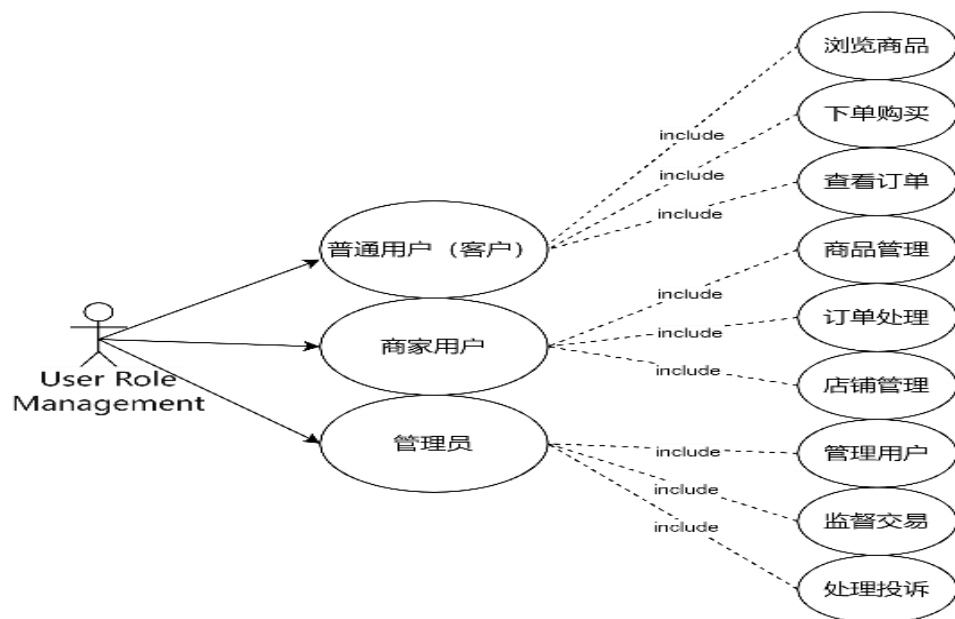


图 2.3 用户角色管理需求分析用例图

用户角色定义分为：普通用户，拥有浏览商品、下单购买、查看订单等功能权限。商家用户，具备商品管理、订单处理、店铺管理等功能权限，发布商品、管理库存、处理订单等操作。管理员，拥有系统管理权限，管理用户、监督交易、处理投诉等，对整个系统的运行进行监管和维护。

## 2.2 系统非功能需求分析

### (1) 性能需求

在此电子商务体系内，必须提供稳定迅速的响应性能，优化页面加载的时效性质，以增强消费者交互体验的顺滑度和连续性。本系统顺应当代电商平台用户规模的扩张，预设了承载高并发交互场景的能力，通过技术架构的优化保障平台的性能及稳定性。

#### （2）安全性需求

数据的安全保障是系统主干架构中的关键节点。通过实施端到端的数据加密策略、身份认证协议与细粒度的访问控制列表，系统旨在为用户隐私信息提供坚不可摧的保护屏障。此外，对于网络攻击的预防机制——包括但不限于 SQL 注入与 XSS 攻击——被纳入系统核心安全计划，确保全局安全和稳定<sup>[9]</sup>。

#### （3）可用性需求

为确保终端的流畅交互，系统在界面设计上追求极致的简约性与直观一致性，同时融入了反馈机制与错误预警系统。用户在不同终端与环境下的便捷操作性和系统的辅助反馈被设计成为系统可用性的核心支柱。

#### （4）可维护性需求

系统的构造原则立足长远，采用了模块化与组件化方案，以适应不断进化的技术要求和场景拓展。代码编纂的透明度和文档建设的完备性构成了系统翻新与维护的坚实基础，为开发团队提供了迅速理解和迭代的平台<sup>[10]</sup>。

综合以上分析，非功能需求作为支持微信小程序电商平台运作的底盘结构，其对性能、安全性、可用性以及可维护性的全面涵盖，确保了在面临稳定运行与原有用户体验提升的双重挑战下，系统能够保持高效和敏捷的服务响应。

### 2.3 本章小结

本章对基于微信小程序的电商平台的系统需求进行了详细分析，主要包括系统功能需求和系统非功能需求两个方面。

在系统功能需求分析阶段，首先对用户需求进行了详细分析，包括用户注册、登录、浏览商品、购物下单等功能，来管理个人信息方面；然后对管理员需求进行了详细分析，包括商品管理、订单管理、用户管理等方面；最后对用户角色管理进行了详细分析，明确了不同用户角色的权限和功能<sup>[11]</sup>。

在系统非功能需求分析中，主要考虑了系统的性能、安全性、可用性和可维护性等方面。对于性能需求，包括系统的响应速度、并发能力和稳定性<sup>[12]</sup>；安全性需求主要考虑用户数据的保护和系统的防攻击能力；可用性需求主要包括系统的易用性和界面友好性；可维护性需求主要考虑系统的易维护性和扩展性。

## 第3章 系统设计

### 3.1 系统总体设计

#### 3.1.1 系统架构设计

系统总体设计是对整个系统进行结构化和规划的过程，其中系统架构设计是至关重要的一部分。在系统架构设计中，需要考虑到系统的各个组成部分之间的关系以及它们如何协同工作来实现系统的功能。

前端架构：基于 Vue3+TypeScript 和 UniApp 框架进行前端开发，利用 Vue3 提供的组件化和响应式特性，实现前端页面的高效开发和管理。

应用 UniApp 框架进行开发，实现了一套代码多端运行的特性，以确保在微信小程序平台上的兼容性和稳定性。

后端架构：采用 uncloud 云开发平台提供的云函数和云数据库实现后端逻辑的开发和管理。

系统集成：前端和后端通过 API 进行通信，实现数据的传输和交互。后端通过调用云数据库来获取和存储数据，为前端提供数据支持。

以下是系统架构设计交互图 3.1：

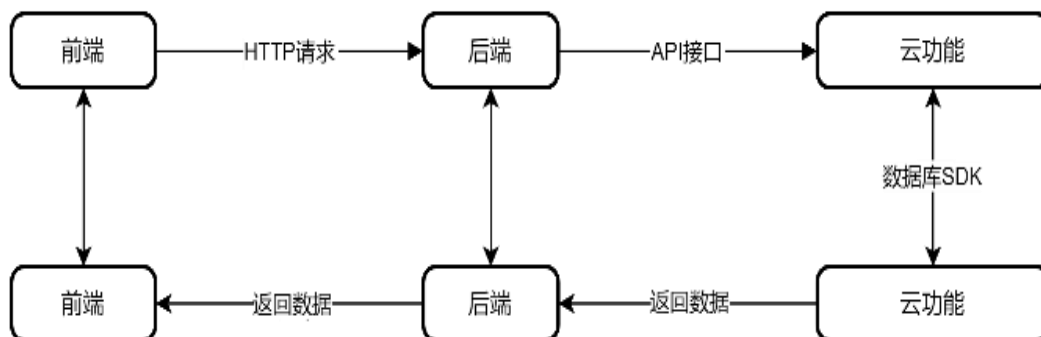


图 3.1 系统架构图

### 3.2 系统功能模块设计

#### 3.2.1 功能模块设计

系统的功能模块设计是根据电商平台的业务需求和用户行为进行设计的，目的是为了用户在微信小程序上进行商品浏览、购物、下单等操作时能够获得流畅的体验。

首页模块通过轮播图、推荐商品等形式展示精选商品，提供快捷入口，引

导用户浏览和购买商品。

商品详情模块展示商品的详细信息，包括商品图片、名称、价格、规格参数等<sup>[13]</sup>，为用户提供全面的商品了解和选择。商品规格选择模块允许用户选择商品的规格和数量，并实时更新商品的价格和库存信息，以使用户按需购买。登录授权模块提供用户登录和注册功能，以确保用户信息的安全，并为用户提供个性化的购物体验。

购物车模块记录用户已选商品，支持增删改查等操作，方便用户管理购物车中的商品。地址管理模块用户添加、编辑、删除收货地址，确保订单准确配送到用户指定地址<sup>[14]</sup>。

订单提交模块允许用户填写订单信息，包括收货地址和支付方式等，确认订单后将生成订单号，准备支付。订单支付模块支持用户选择支付方式进行订单支付操作，以确保交易的安全可靠性。订单管理模块允许用户查看自己的订单列表，包括待付款、待发货、待收货和已完成等订单状态，方便用户跟踪订单状态和查看历史订单记录。

系统流程设计主要包括用户操作流程和系统功能执行流程两个方面。以下是电商平台的业务需求各功能模块图 3.2:

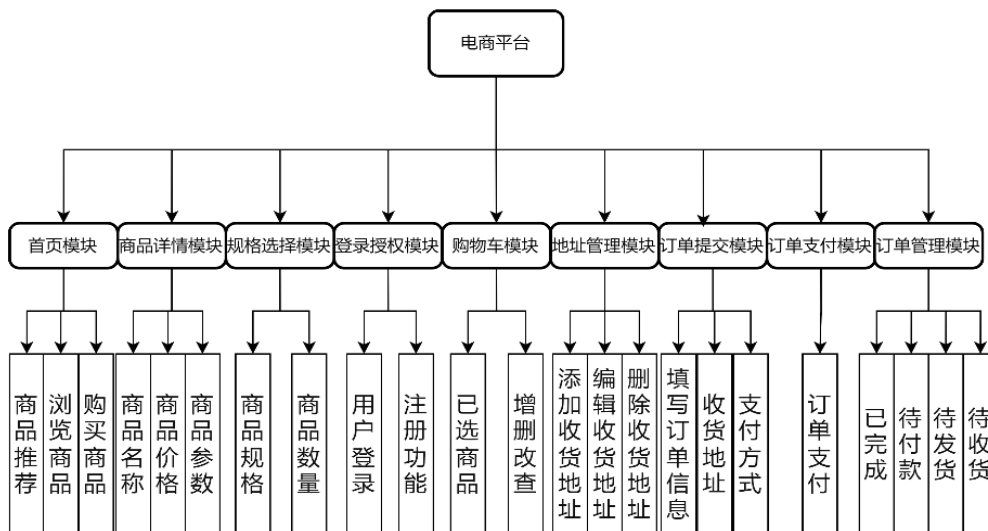


图 3.2 电商平台的业务需求模块图

### (1) 用户操作流程



在系统中，用户的操作流程是整个系统运行的基础。用户操作流程应该包括用户从进入小程序开始到完成各项操作所经历的整个过程。用户浏览商品、将商品添加到购物车并提交订单等操作流程如图 3.3:

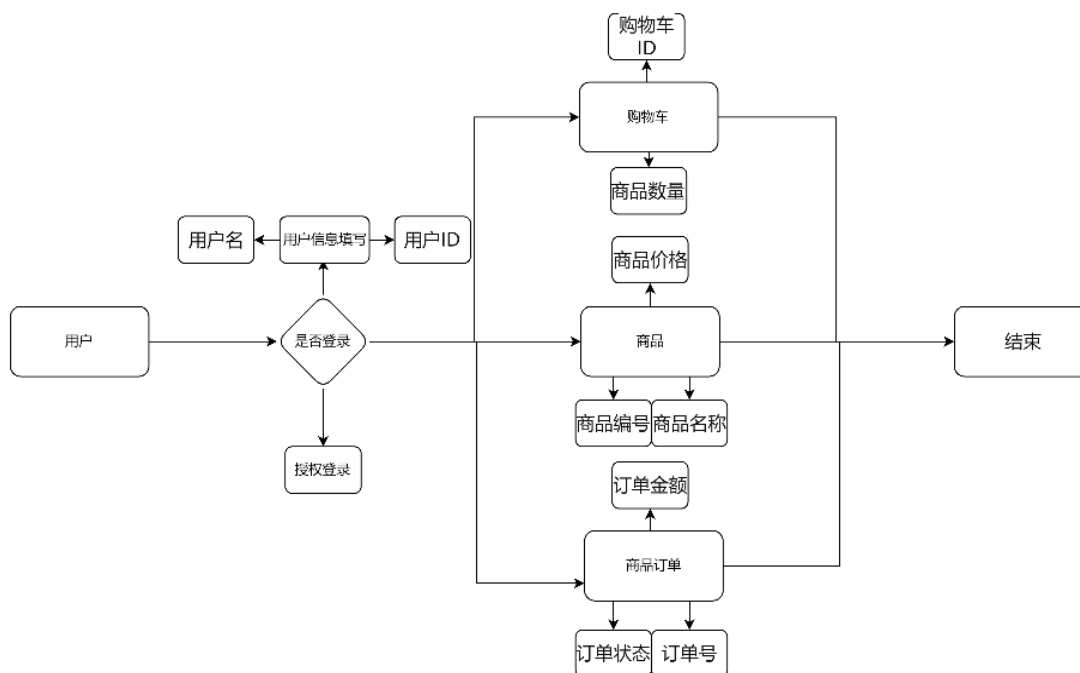


图 3.3 用户操作流程图

## (2) 系统功能执行流程

系统功能执行流程是指系统在接收到用户操作请求后，各个功能模块是如何相互配合完成用户请求的。这包括了系统内部的逻辑处理流程、数据传递流程以及系统响应用户操作。具体步骤如下系统功能执行流程时序图 3.4:

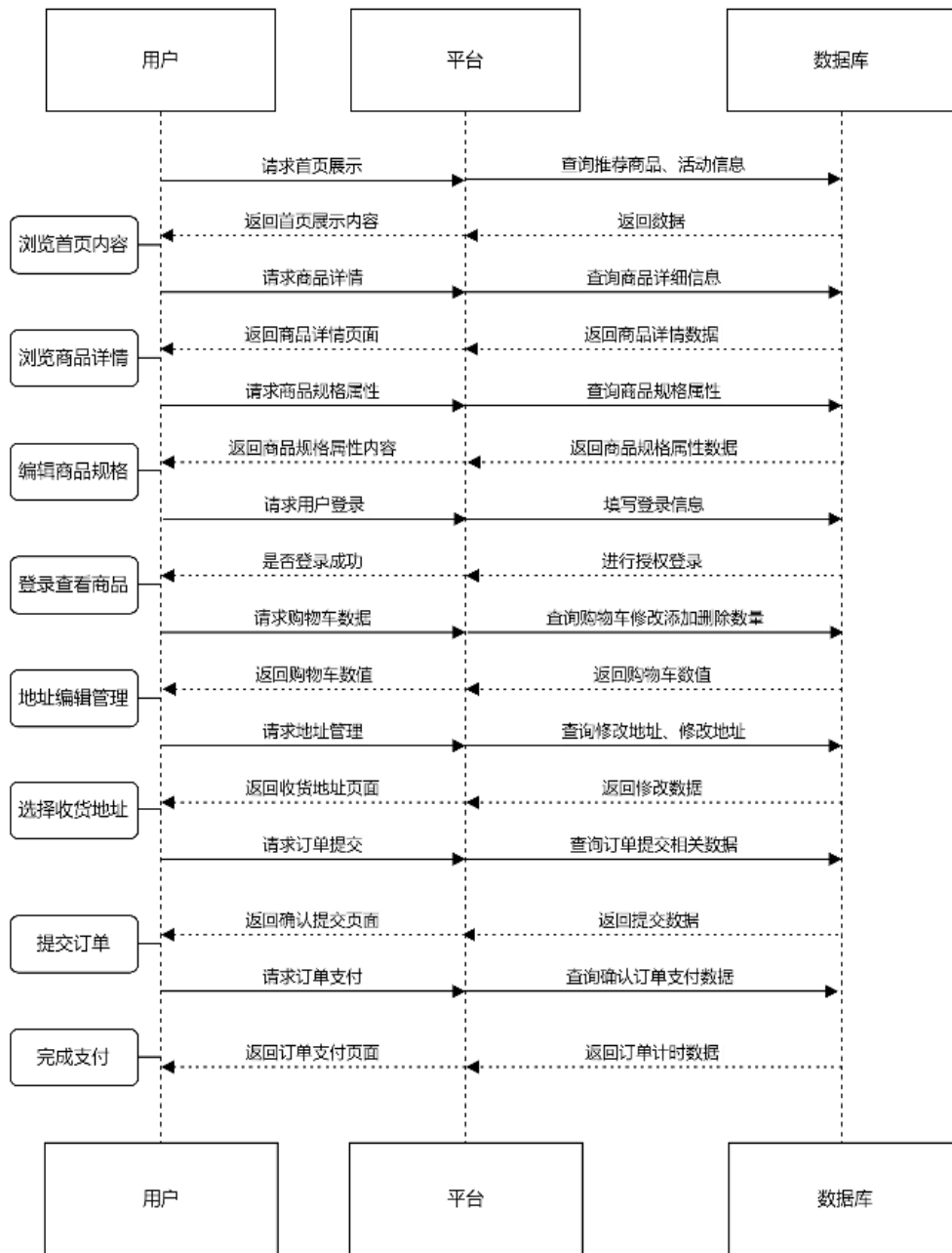


图 3.4 系统功能执行流程时序图

系统流程设计需要充分考虑用户体验和系统性能，确保用户能够顺利完成操作，并且系统能够高效地响应和处理用户请求。为了提高系统的可扩展性和可维护性，系统的流程设计应尽可能简洁明了，避免过多的冗余步骤和复杂的逻辑判断。通过合理的系统流程设计，使系统更加健壮、高效，从而为用户提供良好的使用体验，满足用户的需求。从而提升系统的竞争力和市场地位。

### 3.3 数据库设计

#### 3.3.1 数据库 E-R 模型设计

基于需求分析和功能设计，抽象出四个主要实体：用户（User）、订单（Order）、地址（Address）、商品（Product）。在设计的 ER 图中，这些实体以矩形表示，他们之间的关系以菱形表示，清晰地展示了他们如何相互联系：

User 与 Order 之间的关系是一对多（1: n），表明可以创建多个订单。

User 与 Address 也是一对多（1: n）关系，反映了可以拥有多个收货地址。

Order 与 Product 的是多对多（m: n）关系，显示了可以创建的订单包含多种商品，同时一个商品也属于多个订单。

系统数据库 E-R 模型如下图 3.5：

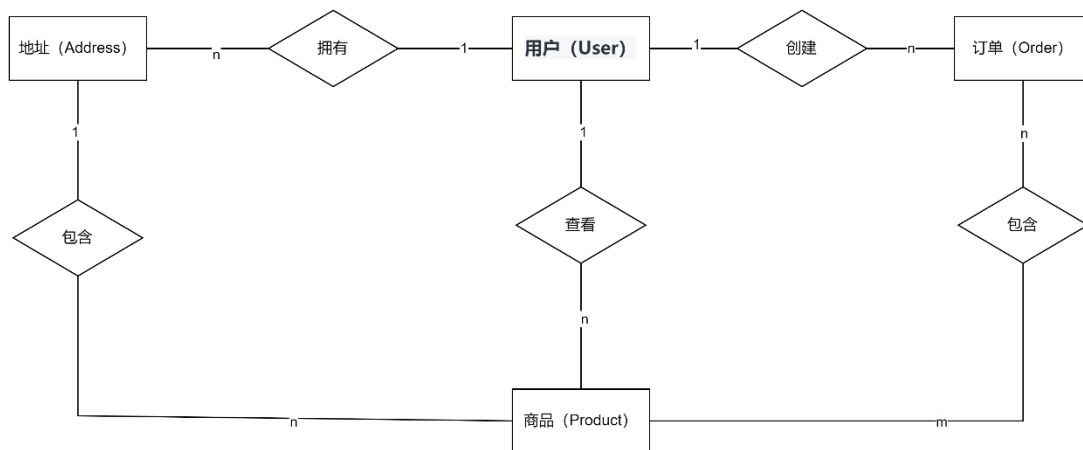


图 3.5 系统数据库 E-R 模型

### 3.3.2 实体属性设计

作为用户，用户的身份和联系方式通过用户名、密码保护以及用户的电子邮件和电话号码并存于系统中。不仅如此，用户的加入时刻和网站活动也被记载，辅助于用户的身份确认和账户的安全。用户实体属性设计如图 3.6:

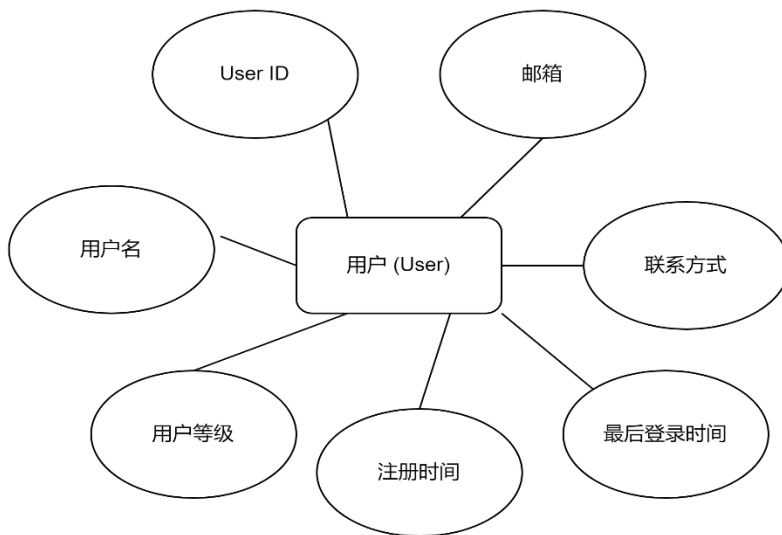


图 3.6 用户实体属性图

在作为买家的旅程中，用户的购物体验从挑选商品到最终将它们收到手中被视为一个过程。这个过程通过订单，以唯一的订单编号和用户作为买家的信息贯穿始终。系统不仅追踪用户支付的具体时刻，还标识了订单的发货时间与状态，确保用户对购物流程的每一步都有清晰的了解。商品实体属性设计如图 3.7:

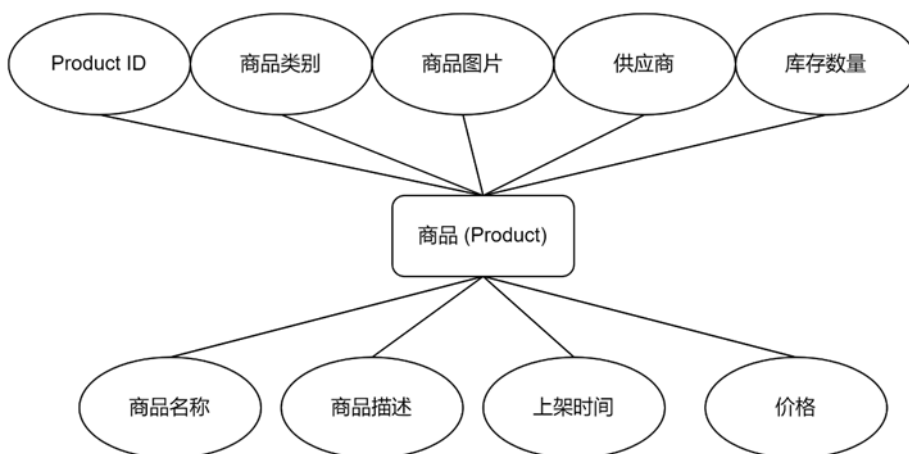


图 3.7 商品实体属性图

在用户的个人信息中，收货地址占有一席之地。每一个地址都有它独有的编号，以及绑定的姓名和联系电话。这些详细信息包含街道号码和邮编，让每次的配送都准确无误。其中，系统还辨识哪一个地址为用户偏好的默认选项，使用户的购物更加便捷。地址实体属性设计如图 3.8：

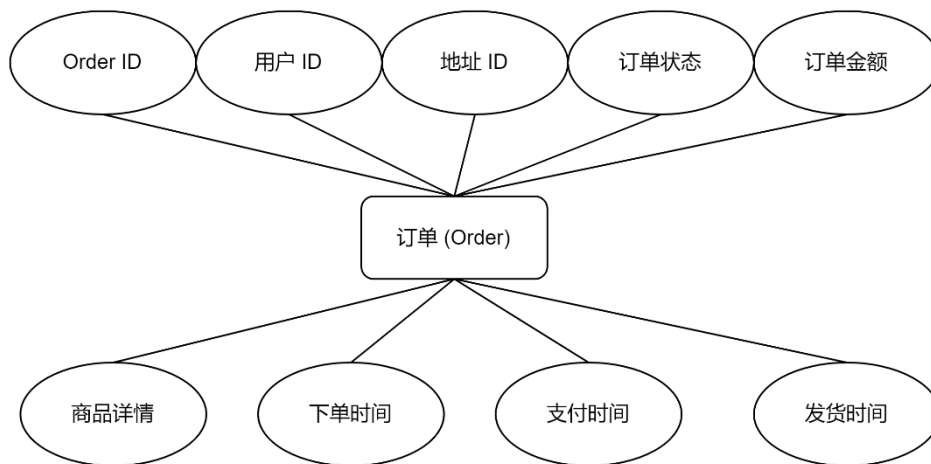


图 3.8 地址实体属性图

关于商品本身，每一件商品都被赋予了特性：从其名称、详细描述中，清晰了解其特征，库存量和价格情况促使购买决策。更进一步，商品的上架时刻与生产来源也为用户选择提供参考。这些关键信息的呈现不仅增强了用户的购物体验，也助力库存的有效管理。订单实体属性设计如图 3.9：

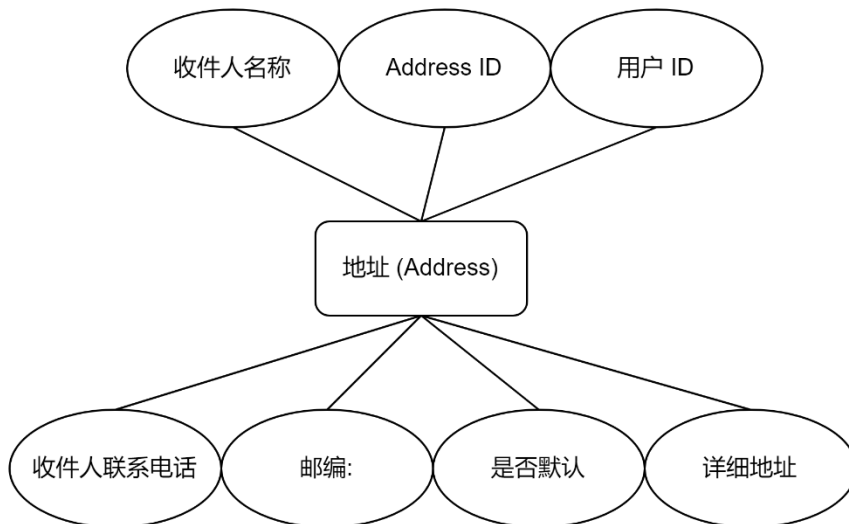


图 3.9 订单实体属性图

### 3.3.3 数据库详细表设计

数据库表概览在整个系统设计中，共构建了十张精心设计的数据表，每张表都围绕特定的模块分担着特定的数据管理职责。

**Banners (首页模块):** 这张表拥有 5 个字段，用于展示和管理首页轮播的广告内容。**Hot\_Recommendations (热门推荐):** 该表具备 6 个字段，聚焦于展示系统中热门推荐的商品，其字段设计能够支持动态的内容展示。**Search\_Module (搜索模块)** - 此表含有 4 个字段，是处理用户搜索请求的核心，它能够确保用户在寻找商品时的高效体验。**Product\_Details (商品详情):** 拥有 14 个字段的这张表提供了全面的商品信息，为用户提供详细的商品数据。**Member\_Authentication (会员登录):** 该表通过其 9 个字段实现了会员的认证机制，是用户登录和身份验证的数据支撑点。**Member\_Profile (会员中心-个人信息):** 含有 9 个字段，专注于存储和管理用户的个人信息，如偏好设置和历史记录。**Member\_Addresses (会员中心-地址管理)** - 包含 10 个字段，负责收集和管理用户的地址信息，以保证订单的正确配送。**Member\_Orders (会员中心-订单管理):** 具备 11 个字段，详细记录和管理用户的订单信息，支持订单跟踪和状态更新。**Shopping\_Cart (购物车):** 这张表有着 4 个字段，维护用户在购物过程中的选择和偏好。**Payments (支付):** 最简约的表，仅有 2 个字段，却是财务交易处理的关键部分，确保了支付操作的安全和效率。

以下是本系统的数据库关系模型 DMD 如图 3.10:

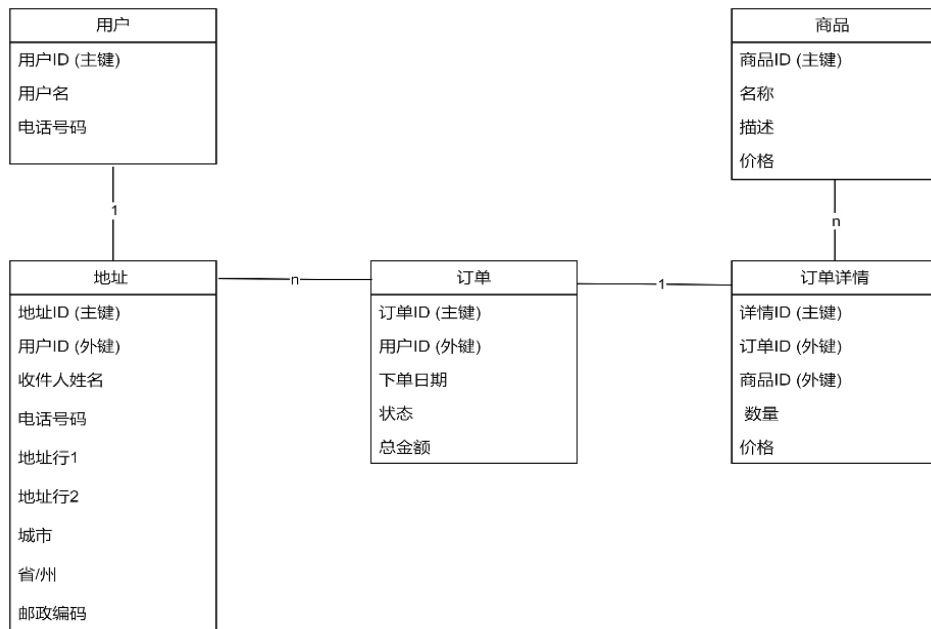


图 3.10 本系统的数据库关系模型 DMD

以下是数据库表概览在整个系统设计完成如以下表 3.1:

表 3.1 本系统数据库表概览

表序号	表名	表含义	字段数	关联表
1	Banners	首页模块	5	
2	Hot_Recommendations	热门推荐	6	
3	Search_Module	搜索模块	4	
4	Product_Details	商品详情	14	
5	Member_Authentication	会员登录	9	
6	Member_Profile	会员中心-个人信息	9	
7	Member_Addresses	会员中心-地址管理	10	
8	Member_Orders	会员中心-订单管理	11	
9	Shopping_Cart	购物车	4	
10	Payments	支付	2	

#### （1）首页模块的设计

信息涵盖了用户的个性化设置和显示偏好。在数据表中，设置项由唯一标识 `setting_id` 作为主键来管理，关联到特定的用户通过 `user_id` 字段。商品显示数量可通过 `limit` 字段控制，而 `additional_properties` 字段用于存储额外的用户配置信息，以序列化文本形式保存。此外，`pageSize` 字段定义了网页默认分页大小，其默认值设为 10，以提供标准化的浏览体验。以下是首页模块的设计如表 3.2:

表 3.2 首页模块的设计

字段	数据类型	字段名	备注
setting_id	int	设置项的唯一标识	主键
user_id	int	关联到用户的 ID	
limit	int	商品数量限制	
additional_properties	text	存储序列化的任意其他属性	
pageSize	Int default 10	默认页大小	

### （2）热门推荐数据表设计

专注于用户的浏览和交互习惯，允许为用户提供个性化内容。表中的 `id` 字段作为主键，唯一标识每一条设置记录。`user_id` 字段与用户表相关联，标记各项设置属于哪个用户。`page` 字段定义用户访问的当前分页页码，默认为第一页。`pageSize` 字段定义一页显示的记录数，默认为 10 条。`subType` 字段用作不同分类或页面类型的标识符。为了增加数据表的适应性和扩展性，`additional_properties` 字段采用 JSON 格式来存储可能随时间发展而增加的附加属性，以下是热门推荐数据表的设计如表 3.3：

表 3.3 热门推荐数据表的设计

字段	数据类型	字段名	备注
id	int	唯一标识每条设置记录	主键
user_id	user_id	用户 ID，假定与用户表相关联	
page	int default 1	分页页码	
pageSize	Int (10)	分页数据每页条数	
subType	varchar (255)	Tab 项的 id，用来区分不同的分类或者页面类型	



additional_properties	json	用来存储额外未知的属性字段，为了灵活性使用 JSON 类型
-----------------------	------	-------------------------------

### （3）搜索模块的设计

反映了对请求内容种类及用户输入的精细管理。数据表通过 `id` 字段指派每条记录一个独一无二的标记，该字段作为主键。字段 `content_type` 的作用是保存“Content-Type”的值，此值指明了请求的媒体类型。`keyword` 字段用于存储用户输入的搜索关键字，该字段为空，因为并非所有的请求都附带关键字。为了增加数据表的灵活性和未来的适用性，使用 `additional_properties` 字段采用 JSON 格式来保存其他可能的属性键值对。以下是搜索模块的设计如表 3.4：

表 3.4 搜索模块的设计

字段	数据类型	字段名	备注
id	int	为每条记录提供一个唯一标识符	主键
content_type	varchar(255)	存储“Content-Type”的值，表明请求的媒体类型	
keyword	varchar(255)	存储输入的搜索关键词，可为 NULL 因为并不是所有请求都需要关键词	
additional_properties	json	用于存储其他任意的属性键值对	

### （4）商品详情设计

以存储关于每款产品的关键信息。每一条记录通过 `id` 字段赋予一个唯一标识符，并将其设置为主键。产品通过 `brand_id` 与品牌表建立外键关联。`name` 字段记录商品名称，而 `desc` 字段用于详细描述产品。定价方面，`price` 字段展示当前价格，`old_price` 提供原价以供比较。用户互动数据通过 `collect_count`、`comment_count` 和 `sales_count` 字段分别记录收藏、评价数量和销售量。`inventory` 字段提供库存信息，`is_pre_sale` 标识商品是否为预售。此外，`discount` 字段显示产品折扣信息，而 `spu_code` 为商品分配 SPU 编码。`additional_properties` 以 JSON 格式存储其他可能的属性键值对，以备未来扩展。以下是商品详情的设计如表

## 3.5:

表 3.5 商品详情的设计

字段	数据类型	字段名	备注
id	varchar(255) PRIMARY KEY	商品 id	主键
brand_id	varchar(255)	品牌 id, 外键 关联至 Brands Table	
name	varchar(255)	商品名称	
desc	text	商品描述	
price	decimal	当前价格	
old_price	decimal	原价	
collect_count	int	收藏数量	
comment_count	int	评价数量	
sales_count	int	销量	
inventory	int	库存	
is_pre_sale	boolean	是否为预售商 品	
discount	decimal	折扣	
spu_code	varchar(255)	SPU 编码	
additional_properties	json	存储其他任意 的属性键值对	

## (5) 会员登录的设计

是为处理用户身份认证和保存登录信息设计的。这个数据表通常包括 `user_id` 作为主键，它唯一识别每一个用户。`username` 和 `password` 字段分别存储用户的登录名和加密后的密码，确保安全性。另外，`email` 字段用于存储用户的电子邮箱地址，便于找回密码或发送重要通知。`login_count` 字段记录用户的登录次数，有助于分析用户行为。`last_login` 字段记录上一次登录的时间，用于监测用户活跃度。为了应对未来的需求变化，`additional_info` 字段采用 JSON 格式，用于存储额外的用户信息，如用户偏好设置。以下是商品详情的设计如表 3.6:

表 3.6 会员登录的设计

字段	数据类型	字段名	备注
id	varchar(255)	用户的唯一标识符	主键
code	varchar(255)	由 wx.login() 获取的微信登录凭证	
encrypted_data	text	通过 getphonenumber 事件回调中获取的加密数据	
iv	varchar(255)	通过 getphonenumber 事件回调中获取的初始化向量	
authorization	text	用于存储用户登录后获取的 token	
phone_number	varchar(255)	用户的手机号码	
account	varchar(255)	用户的用户名或手机号码	
password	varchar(255)	用户的密码	
additional_properties	json	用于存储其他任意的属性键值对	

#### （6）会员中心的个人信设计

精细地规划，以保障会员信息的全面和安全管理。该表以 `authorization` 字段作为主键，该字段用作用户鉴权的授权令牌，保障访问安全性。会员的基本信息，如生日(`birthday`)、所在城市(`city_code`)、县区(`county_code`)以及省份(`province_code`)编码，均以精确字段存储，以确保信息准确性。用户的性别(`gender`)以枚举类型存在，提供了‘男’、‘女’及‘未知’选项，以满足不同用户的需求。此外，用户昵称(`nickname`)和职业(`profession`)的记录，进一步丰富了会员的个人资料。对于扩展性的考量，`additional_properties` 字段采用 JSON 格式，这样灵活地添加或修改会员可能拥有的其他任意信息。以下是会员中心-个人信息的设计如表 3.7:

表 3.7 会员中心-个人信息的设计

字段	数据类型	字段名	备注
----	------	-----	----

authorization	varchar(255) primary key	用于用户鉴权的授权令牌，设置为主键	主键
birthday	date	用户的生日，符合格式 YYYY-MM-DD	
city_code	varchar(255)	用户所在城市的编码	
county_code	varchar(255)	用户所在区/县的编码	
gender	enum('男', '女', '未知')	用户的性别，为枚举类型	
nickname	varchar(255)	用户的昵称	
profession	varchar(255)	用户的职业	
province_code	varchar(255)	用户所在省份的编码	
additional_properties	json	用于存储其他任意的属性键值对	

#### （7）会员中心的地址管理设计

经过精心设计，用于高效安全地存储和处理会员的收货地址信息。在该表中，每个收货地址记录都由一个 id 字段唯一标识，并设为主键。这确保了每个地址的唯一性与识别。user\_authorization 字段关联至用户的授权令牌，以确保地址与特定用户的安全链接。详细地址(address)、城市(city\_code)、区县(county\_code)及省份(province\_code)的编码信息分别存储在独立的字段中，以保证信息的准确性和可靠性。以下是会员中心-地址管理的设计如表 3.8:

表 3.8 会员中心-地址管理的设计

字段	数据类型	字段名	备注
id	varchar(255) primary key	收货地址的唯一标识符	主键
user_authorization	varchar(255)	关联到用户的授权令牌	
address	varchar(255)	详细地址	
city_code	varchar(255)	所在城市的编码	
county_code	varchar(255)	所在区/县的编码	

province_code	varchar(255)	所在省份的编码
contact	varchar(255)	联系方式
receiver	varchar(255)	收货人姓名
is_default	boolean	是否为默认地址（0 否，1 是）
additional_properties	json	用于存储其他任意的属性键值对

#### （8）会员中心的订单管理设计

部分关键依赖于一个精密构建的数据表，旨在管理和追踪顾客所下订单的全方位信息。每个订单条目均由 `id` 字段作为主键识别，具备唯一性。`address_id` 字段与 `ShippingAddresses` 表相联结，确保了所选收货地址信息的正确关联。买家可通过 `buyer_message` 字段留下任何特殊指示或消息。

配送时间的偏好由 `delivery_time_type` 字段指示，包括几种选项：无时间限制、只在工作日、或仅限于双休日或假期间配送。订单的当前阶段由 `order_state` 字段记录，它通过一系列的整数值分别标识了订单的多种状态，从待支付到已取消等。若订单被取消，相关原因将在 `cancel_reason` 字段中被详细说明。以下是会员中心-订单管理的设计如表 3.9：

表 3.9 会员中心-订单管理的设计

字段	数据类型	字段名	备注
id	varchar(255) PRIMARY KEY	订单的唯一标识符	主键
address_id	varchar(255)	所选的地址 ID，参照 <code>ShippingAddresses</code> 表	
buyer_message	varchar(255)	买家留言	
delivery_time_type	int	配送时间类型（1 为不限，2 为工作日，3 为双休或假日）	
order_state	int	订单状态（1 为待付款、2 为待发货、3 为待收货、4 为待评价、5 为已完成、6 为已取消）	
cancel_reason	varchar(255)	订单取消理由	

pay_type	int	支付方式（1 为在线支付，2 为货到付款）
pay_channel	int	支付渠道（1 为支付宝、2 为微信）
order_id	varchar(255)	订单 id
sku_id	varchar(255)	商品 skuId
count	int	购买商品的数量

### （9）购物车的设计

构建是网络购物体系中不可或缺的一环。在该表中，每个商品条目的核心都是一个 `sku_id`，由 `sku_id` 字段负责标识，同时确立为主键以保障唯一性。用户与其购物车内容的安全链接由 `user_authorization` 字段担保，此字段存储用户授权令牌，并作为外键，与用户表产生联系。商品的选购数量由 `count` 字段实时记录，而顾客是否决定购买特定商品的状态则通过 `selected` 字段的布尔值显示。以下是购物车的设计如表 3.10:

表 3.10 购物车的设计

字段	数据类型	字段名	备注
sku_id	varchar(255) PRIMARY KEY	商品的 SKU ID	主键
user_authorization	varchar(255)	用户的授权令牌，作为外键连接至用户表	
count	int	商品数量	
selected	boolean	商品是否被选中	

支付模块的数据设计是电子商务平台安全性和用户体验的关键。每笔支付的数据记录以订单的唯一标识 `order_id` 作为主键，确保了每一笔交易的独立性和可查询性。与此同时，`user_authorization` 字段存储着用户的授权令牌，此字段是与用户表相连接的外键，提供了对用户身份的验证和链接。以下是支付的设计如表 3.11:

表 3.11 支付的设计

字段	数据类型	字段名	备注
----	------	-----	----

order_id	varchar(255) PRIMARY KEY	订单的唯一标识符	主键
user_authorization	varchar(255)	用户的授权令牌，用于链接到用户表	

### 3.4 本章小结

在系统设计这一章节中，对基于微信小程序的电商平台进行了全面的设计与规划。首先，讨论了系统的整体设计，包括系统的架构和功能模块的设计，以确保系统能够高效稳定地运行。然后，通过系统功能执行流程 ER 图的设计，清晰地展示了系统内部各个功能模块之间的交互流程，为系统的实现提供了重要参考<sup>[15]</sup>。

通过这一章节的努力，为电商平台开发奠定了坚实的基础。不仅细致考虑到了系统的功能需求，更是周全考虑了如性能、安全性等非功能需求，确保了开发的系统能够满足用户的多元化需求，同时也具备了良好的用户体验。此外，还采用了当前流行的技术框架和云开发平台，为系统的顺利实现提供了强有力的技术支持。

## 第4章 系统开发与实现

### 4.1 系统技术选型

在开发微信小程序电商平台时，挑选了一些顶尖的技术和工具。这里，详细介绍它们及它们带来的独特优势：

选择了 Node，当前版本是 v20.11.1，它赋予了项目跨平台的运行能力，能显著提升网络应用的拓展性。为了管理项目依赖，使用 pnpm，版本 8.15.4，它的空间节省与安装加速功能为开发流程带来了极大便利。

编码时，依赖于 TypeScript 的静态类型检查特性，它极大地提升了代码质量和稳定性。同时，采用了 Vue.js，因为它的数据驱动与组件化使得用户界面变得即灵活又高效。

为了实现多端运行，选择了 UniApp 框架，它让一套代码便可适用于各种前端平台，包括微信小程序。在后端开发方面，选择 unicloud，它提供了便捷的云端开发服务，使得前后端协作更加顺畅。

在工具的选择上，用官方的微信开发者工具进行小程序开发和调试，它让开发过程更为流畅。同时，还选用了 VS Code 和 HbuilderX 这两款强大的编辑器，它们优秀的插件生态系统和对多端项目的原生支持使开发效率有了飞跃性的提升。

### 4.2 开发环境建立

**HbuilderX 配置：**HbuilderX 是由 DCloud 推出的一款集成开发环境，专注于前端开发，特别是针对移动端应用的开发。它提供了丰富的功能和插件，适用于多种前端开发场景，包括 Vue3 项目的开发。

**安装 Vue3 编译插件。**为了在 HbuilderX 中支持 Vue3 项目的开发，安装 Vue3 编译插件，以确保能够在项目中正确编译和运行 Vue3 代码。该插件通常通过 HbuilderX 的插件市场或者扩展商店进行安装。在开发过程中，选择适合的开发工具并配置相关插件是非常重要的。针对你提到的 VS Code，以下是对插件的选择和配置建议。

**VS Code 配置。**uni-app 插件，uni-helper 提供了一些 uni-app 开发常用的代码片段与模板相结合，提高开发效率。与 UniApp 小程序扩展一起使用，该插件提供了 Uni-App 项目的代码片段、语法高亮以及语法检查等功能，方便开发者快速编写 uni-app 项目。

**vue3 插件。**Vue Language Features (Volar)：Volar 是专门为 Vue3 设计的语言



服务插件，提供了丰富的代码智能提示、语法高亮、自动补全等功能，能够极大地提高 Vue3 项目的开发效率。

TypeScript Vue Plugin (Volar): 针对 Vue3 项目中使用 TypeScript 进行开发，该插件提供了与 Volar 的集成，更好地支持 Vue3 项目中的 TypeScript 代码编写。

### 4.3 核心功能模块的实现

#### 4.3.1 首页自定义导航栏设计

以下展示了一个简单的导航栏结构，包含了 Logo 和搜索功能。具体的样式和交互逻辑可能需要在外部的 CSS 和 JavaScript 文件中定义的实现和实现图的展示如图 4.1:



图 4.1 导航栏实现界面

其核心代码如下:

```
<script setup lang="ts">
</script>
<template>
  <view class="navbar">
    <!-- logo 文字 -->
    <view class="logo">
      <image class="logo-image" src="@/static/images/logo.png"></image>
      <text class="logo-text">新鲜 · 亲民 · 快捷</text>
    </view>
    <!-- 搜索条 -->
```

```
<view class="search">
  <text class="icon-search">搜索商品</text>
  <text class="icon-scan"></text>
</view>
</view>
</template>
```

#### 4.3.2 推荐模块设计

以下实现热门推荐页面的功能，包括页面标题的动态设置和异步获取热门推荐数据等。推荐模块实现的展示如图 4.2:

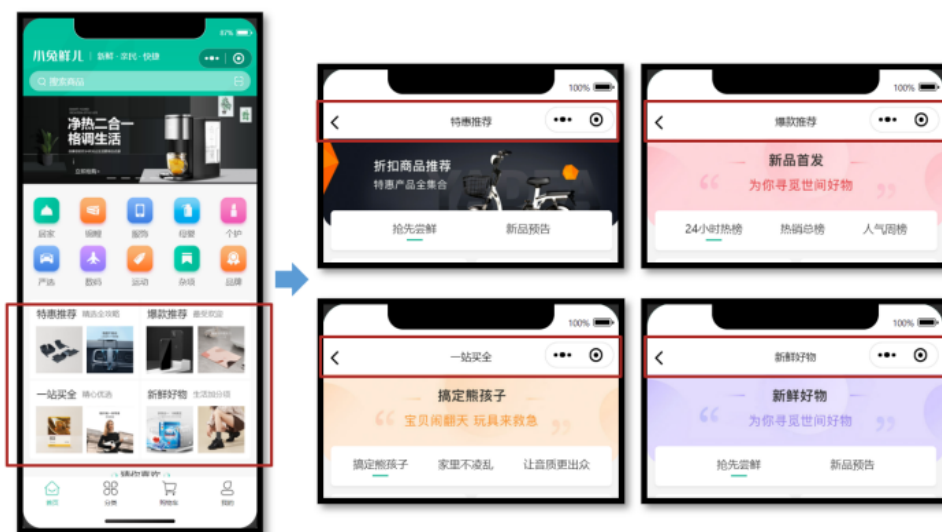


图 4.2 推荐模块实现界面

其核心代码如下:

```
<script setup lang="ts">
import { getHotRecommendAPI } from '@services/hot'
import type { SubTypeItem } from '@types/hot'
import { onLoad } from '@dcloudio/uni-app'
import { ref } from 'vue'
// 热门推荐页 标题和 url
const hotMap = [
  { type: '1', title: '特惠推荐', url: '/hot/preference' },
  { type: '2', title: '爆款推荐', url: '/hot/inVogue' },
  { type: '3', title: '一站买全', url: '/hot/oneStop' },
```

```
{ type: '4', title: '新鲜好物', url: '/hot/new' },  
]  
// uniapp 获取页面参数  
const query = defineProps<{  
  type: string  
}>()  
// 获取当前推荐信息  
const currHot = hotMap.find((v) => v.type === query.type)  
// 动态设置标题  
uni.setNavigationBarTitle({ title: currHot!.title })  
// 推荐封面图  
const bannerPicture = ref("")  
// 推荐选项  
const subTypes = ref<(SubTypeItem & { finish?: boolean })[]>([])  
// 高亮的下标  
const activeIndex = ref(0)  
// 获取热门推荐数据  
const getHotRecommendData = async () => {  
  // 技巧：环境变量，开发环境，修改初始页面方便测试分页结束  
  page: import.meta.env.DEV ? 30 : 1,  
  pageSize: 10,  
})
```

#### 4.3.3 商品分类设计

以下代码用于在 **Vue** 组件中实现获取首页轮播图数据和分类列表数据的功能。它通过异步函数从相应的服务接口获取数据，并使用响应式变量来存储数

据，以便在组件中动态展示。分类实现的展示如图 4.3:

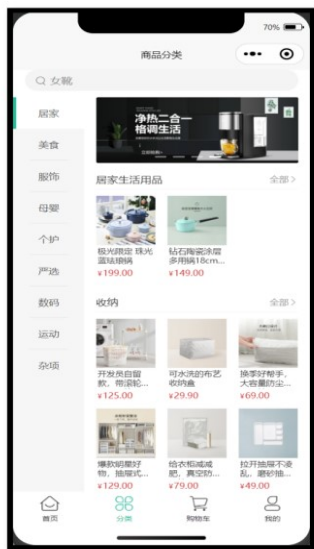


图 4.3 商品分类模块实现界面

其核心代码如下:

```
<script setup lang="ts">
import { getCategoryTopAPI } from '@services/category'
import { getHomeBannerAPI } from '@services/home'
import type { CategoryTopItem } from '@types/category'
import type { BannerItem } from '@types/home'
import { onLoad } from '@dcloudio/uni-app'
import { computed, ref } from 'vue'
import PageSkeleton from './components/PageSkeleton.vue'
// 获取轮播图数据
const bannerList = ref<BannerItem[]>([])
const getBannerData = async () => {
  const res = await getHomeBannerAPI(2)
  bannerList.value = res.result
}
// 获取分类列表数据
const categoryList = ref<CategoryTopItem[]>([])
const activeIndex = ref(0)
const getCategoryTopData = async () => {
  const res = await getCategoryTopAPI()
```

```
categoryList.value = res.result
}
```

#### 4.3.4 商品详情设计

在以下整个代码片段中，通过响应式变量、事件处理函数和 API 调用，实现了轮播图变化时更新当前索引和点击图片预览大图的功能，这对于商品详情页面的交互体验非常有用。详情实现页如图 4.4:

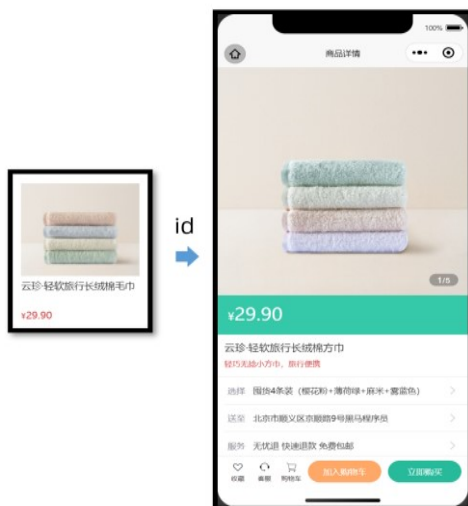


图 4.4 商品详情实现界面

其核心代码如下:

```
<script setup lang="ts">
// 轮播图变化时
const currentIndex = ref(0)
const onChange: UniHelper.SwiperOnChange = (ev) => {
  currentIndex.value = ev.detail.current
}
// 点击图片时
const onTapImage = (url: string) => {
  // 大图预览
  uni.previewImage({
    current: url,
    urls: goods.value!.mainPictures,
  })
}
```

```
}
```

#### 4.3.5 购物车页面设计

以下代码实现了一个简单的购物车列表布局，包含了商品图片、名称、属性、价格等信息，并且支持左滑操作。购物车实现如图 4.5：

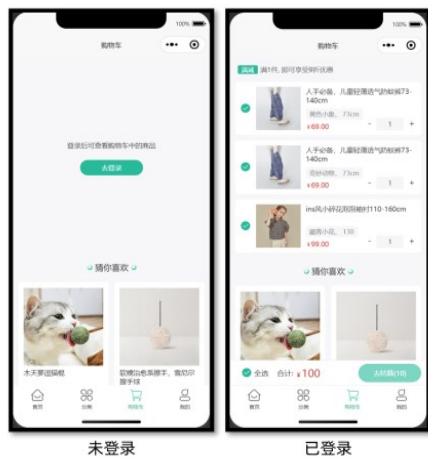


图 4.5 购物车模块实现界面

其核心代码如下：

```
<script setup lang="ts">
</script>
<template>
  <scroll-view scroll-y class="scroll-view">
    <!-- 已登录: 显示购物车 -->
    <template v-if="true">
      <!-- 购物车列表 -->
      <view class="cart-list" v-if="true">
        <!-- 滑动操作分区 -->
        <uni-swipe-action>
          <!-- 滑动操作项 -->
          <uni-swipe-action-item v-for="item in 2" :key="item" class="cart-swipe">
            <!-- 商品信息 -->
          </uni-swipe-action-item>
        </uni-swipe-action>
      </view>
    </template>
  </scroll-view>
</template>
```

#### 4.3.6 订单页面设计

以下这段代码提供了一个函数 `getMemberOrderByIdAPI`，用于获取特定订单的信息。它通过 HTTP GET 请求发送到指定的 URL (`/member/order/${id}`)，并返回一个 Promise 对象，用于异步获取订单信息。获取订单详情如图 4.6:

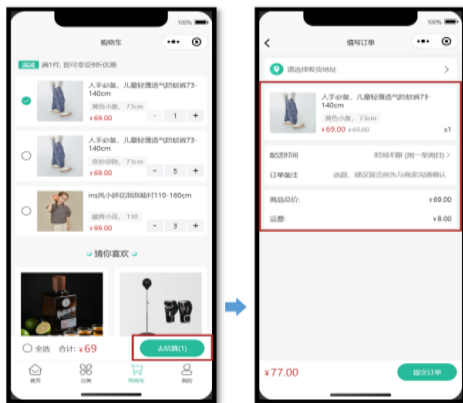


图 4.6 获取订单详情实现界面

其核心代码如下:

```
<script setup lang="ts">
import { getGoodsByIdAPI } from '@services/goods'
import type { GoodsResult } from '@types/goods'
import { onLoad } from '@dcloudio/uni-app'
import { ref } from 'vue'
import AddressPanel from './components/AddressPanel.vue'
import ServicePanel from './components/ServicePanel.vue'
// 获取屏幕边界到安全区域距离
const { safeAreaInsets } = uni.getSystemInfoSync()
// 接收页面参数
const query = defineProps<{
  id: string
}>()
// 获取商品详情信息
const goods = ref<GoodsResult>()
const getGoodsByIdData = async () => {
  const res = await getGoodsByIdAPI(query.id)
  goods.value = res.result
}
```

```
}
```

#### 4.3.7 订单状态列表实现

以下这段代码提供了一个方便的方式来管理和获取订单状态的信息，通过订单状态的唯一标识符快速识别订单状态，而订单状态的文本描述则方便用于展示给用户。订单状态列表详情如图 4.7

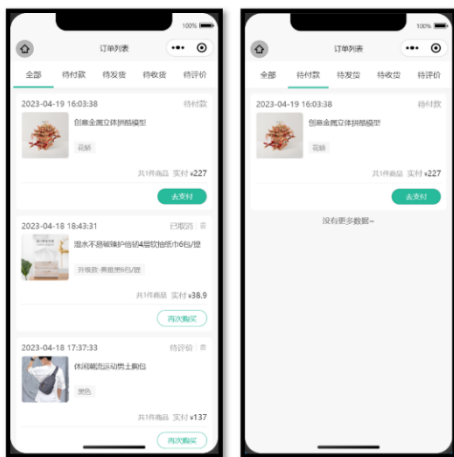


图 4.7 可视化推荐模块相关实现界面

其核心代码如下：

```
export const orderStateList = [
  { id: 0, text: " " },
  { id: 1, text: '待付款' },
  { id: 2, text: '待发货' },
  { id: 3, text: '待收货' },
  { id: 4, text: '待评价' },
  { id: 5, text: '已完成' },
  { id: 6, text: '已取消' },
]
</script>
<template>
  <view class="viewport">
    <!-- tabs -->
    <view class="tabs">
      <text class="item" v-for="item in 5" :key="item"> 待付款 </text>
```



```
<!-- 游标 -->
<view class="cursor" :style="{ left: 0 * 20 + '%' }"></view>
</view>
<!-- 滑动容器 -->
<swiper class="swiper">
  <!-- 滑动项 -->
  <swiper-item v-for="item in 5" :key="item">
    <!-- 订单列表 -->
    <scroll-view scroll-y class="orders">
      <view class="card" v-for="item in 2" :key="item">
        <!-- 订单信息 -->
        <view class="status">
          <text class="date">2023-04-14 13:14:20</text>
          <!-- 订单状态文字 -->
          <text>待付款</text>
          <!-- 待评价/已完成/已取消 状态: 展示删除订单 -->
          <text class="icon-delete"></text>
        </view>
      </view>
    </scroll-view>
  </swiper-item>
</swiper>
```

#### 4.3.8 登录界面的实现

在探索登录和注册的不同途径时，遇到了几种常见的方法，包括使用用户名或手机号加密码、手机号加验证码，以及授权登录等。每种方法都有其优势，但考虑到用户便捷性和平台的安全性，最终决定主要采用授权登录方式。授权登录的过程相当直接。以下是实现的登录如图 4.8

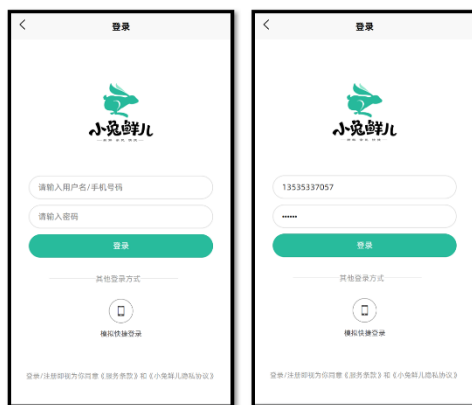


图 4.8 可视化推荐模块相关实现界面

其核心代码如下：

```

<script setup lang="ts">
</script>
<template>
  <view class="viewport">
    <view class="logo">
      <image
        src="https://pcapi-xiaotuxian-front-
devtest.itheima.net/miniapp/images/logo_icon.png"
      ></image>
    </view>
    <view class="login">
      <!-- 网页端表单登录 -->
      <!-- <input class="input" type="text" placeholder="请输入用户名/手机号码
" /> -->
      <!-- <input class="input" type="text" password placeholder="请输入密码" />
-->

      <!-- <button class="button phone">登录</button> -->
      <!-- 小程序端授权登录 -->
      <button class="button phone">
        <text class="icon icon-phone"></text>
        手机号快捷登录
      </button>
      <view class="extra">
        <view class="caption">
          <text>其他登录方式</text>
        </view>
        <view class="options">
          <!-- 通用模拟登录 -->
          <button>
            <text class="icon icon-phone">模拟快捷登录</text>
          </button>
        </view>
      </view>
    </view>
  </view>

```

#### 4.3.9 用户界面实现

以下这段代码主要用于在个人页面中显示用户的个人资料信息以及订单选项，通过 `orderTypes` 变量来定义订单选项的内容，以便在页面中展示。用户界面实现如图 4.9：



图 4.9 用户模块实现界面

其核心代码如下：

```
// src/pages/my/my.vue
<script setup lang="ts">
// 订单选项
const orderTypes = [
  { type: 1, text: '待付款', icon: 'icon-currency' },
  { type: 2, text: '待发货', icon: 'icon-gift' },
  { type: 3, text: '待收货', icon: 'icon-check' },
  { type: 4, text: '待评价', icon: 'icon-comment' },
]
</script>
<template>
<scroll-view class="viewport" scroll-y enable-back-to-top>
  <!-- 个人资料 -->
  <view class="profile" :style="{ paddingTop: safeAreaInsets!.top + 'px' }">
    <!-- : 已登录 -->
    <view class="nickname">程序员</view>
    <view class="meta">
```

#### 4.4 本章小结

在系统实现与开发这一章节中，首先进行了开发环境和工具的准备，确保了系统开发所需的软硬件环境的完备和配置。选择了合适的操作系统、开发语言、开发框架和开发工具，并进行了相应的配置，以确保开发过程的顺利进行。

在核心功能模块的开发方面，完成了首页页面的布局等模块。

通过本章的开发工作，成功设计并实现了系统的核心功能模块，为接下来的系统测试和上线工作奠定了坚实的基础。在实现过程中，充分利用了现有的开发工具和适合的技术框架，以确保系统的稳定性和出色性能。同时，还十分重视用户体验的设计，旨在提高用户的满意度和舒适度。

## 第5章 测试

### 5.1 系统功能测试

#### （1）单元测试

**系统功能测试：**集中于验证用户认证流程、商品的呈现和详情展示、购物车功能、订单处理、支付流程以及订单追踪管理等关键环节的性能。**商品模块测试：**着重于检验商品搜索、详细信息展示、分类导航及购物车添加等基础功能的准确性和响应性。**用户登录模块测试：**评估用户的登录、登出流程，及在未登录状态下访问受限部分的反馈机制是否正常。**用户购物车模块测试：**确保用户能顺利进行商品添加、数量调整、选中/取消选中以及从购物车中删除商品的操作，并能够成功结算。

#### 5.1.1 系统功能测试

**测试已注册用户的登录：**模拟已注册用户使用正确的用户名和密码进行登录，验证系统能够准确识别用户身份并成功登录系统。**测试登录失败情况：**模拟用户使用错误的用户名或密码进行登录，验证系统能够正确提示用户登录失败并给出相应的错误信息。汇总如表 5.1：

表 5.1 系统功能测试用例表

序号	测试项	测试操作	预期结果	测试结果
1	用户登入	用户通过微信账号登入	登入成功，显示用户的个人信息	成功
2	商品显示	主页显示商品列表，并且对每个商品的信息展示完整	列表显示所有商品，包括商品的图片、名称和价格等信息	成功
3	搜索功能	用户通过搜索框对商品进行检索	显示所有包含关键词的商品	成功
4	商品详情	用户点击商品查看商品详情	页面跳转到该商品的详情页面，显示商品的所有信息	成功
5	购物车	用户将商品添加至购物车	商品成功添加到购物车，购物车图标上显示商品数量	成功

6	订单提交	用户对购物车中的商品进行结算提交	订单提交成功，跳转到支付页面	成功
7	支付功能	用户完成支付	支付成功，跳转到订单详情页面，订单状态变为已支付	成功
8	订单管理	用户查看自己的订单信息	显示用户所有的订单信息，包括订单状态和商品信息等	成功

### 5.1.2 商品模块测试

在进行微信小程序电商平台的商品模块功能测试，针对该模块的主要功能设计一系列测试用例。如下表 5.2:

表 5.2 商品模块测试用例表

序号	测试项	测试操作	预期结果	测试结果
1	搜索框搜索商品	在搜索框中输入商品关键词，点击搜索	列表显示所有包含该关键词的商品	成功
2	查看单个商品的具体信息	从商品列表中选择一项，点击进入详情页	商品详情页成功加载，显示该商品的所有相关信息	成功
3	商品分类的浏览	在首页点击进入商品分类页面	显示所有商品分类，并且每个分类下面有相应的商品	成功
4	将商品加入购物车	在商品详情页，点击“加入购物车”按钮	商品成功添加到购物车，购物车图标数量增加	成功

### 5.1.3 用户登录模块测试

对于基于微信小程序的电商平台的登录模块，主要从用户登录、注册、密码找回等方面进行详细的测试。以下是一份示例的测试用例如表 5.3:

表 5.3 用户登录模块测试用例表

序号	测试项	测试操作	预期结果	测试结果
1	用户正常登录	用户打开小程序，输入正确的微信授权登录信息，点击登录按钮	用户成功登录，跳转到主页，并在个人中心能看到自己的个人信息	成功

2	用户登录信息错误	用户打开小程序，输入错误的微信授权登录信息，点击登录按钮	提示用户“登录信息错误，请重新输入”	成功
3	用户退出登录	用户进入个人中心，点击退出登录按钮	成功退出登录，跳转到登录页面	成功
4	用户未登录状态下访问购物车	用户未登录，尝试点击进入购物车	弹出提示框，提示用户需要登录才能访问，跳转到登录页面	成功
5	用户未登录状态下下单购买	用户未登录，对商品进行下单操作	弹出提示框，提示用户需要登录才能购买，跳转到登录页面	成功

#### 5.1.4 用户购物车模块测试

对于微信小程序电商平台的购物车模块，主要功能包括添加商品、删除商品、修改商品数量、商品选中/反选、查看商品信息等。以下是购物车模块的一份示例测试用例如表 5.4:

表 5.4 用户购物车测试用例表

序号	测试项	测试操作	预期结果	测试结果
1	添加商品到购物车	用户在商品详情页，选择商品规格和数量，点击添加到购物车按钮	商品成功添加到购物车，购物车图标上显示商品数量	成功
2	在购物车修改商品数量	用户在购物车内，点击增加或减少商品数量按钮	商品数量成功增加或减少，小计价格同时改变	成功
3	用户从购物车删除商品	用户在购物车内，点击删除商品按钮	商品成功从购物车中删除	成功
4	在购物车选中和取消选中商品	用户在购物车内，点击选中或取消选中商品	商品选中或者取消选中状态，购物车底部的总计价格同时改变	成功
5	从购物车结算选中的商品	用户在购物车内，点击结算按钮	跳转到确认订单页面，展示选中的商品、价格和总数量	成功

#### (2) 集成测试

通过应用黑盒测试方法对微信小程序电商平台进行了软件测试。主要基于系统的各个主要功能模块，如用户登录功能、商品模块、购物车添加模块等，

设计了一系列详尽的功能测试用例。通过模拟用户使用行为，检验系统各项功能是否能正确且有效地服务用户。

在测试过程中，重点评估系统的性能表现和稳定性，通过模拟大量用户同时在线的情况和提升并发用户数量来压测系统的稳定程度，同时检测系统在各种条件下的容错性和兼容性。以下是系统功能测试用例和用户登录模块测试用例如表 5.5 和表 5.6:

表 5.5 系统功能测试用例

序号	测试项	测试操作	预期结果	测试结果
1	用户登录	用户通过微信账号登录	登录成功，显示用户的个人信息	成功
2	商品显示	主页显示商品列表，并且对每个商品的信息展示完整	列表显示所有商品，包括商品的图片、名称和价格等信息	成功
3	商品显示	用户通过搜索框对商品进行检索	显示所有包含关键词的商品	成功

表 5.6 用户登录模块测试用例

序号	测试项	测试操作	预期结果	测试结果
1	用户正常登录	用户打开小程序，输入正确的微信授权登录信息，点击登录按钮	用户成功登录，跳转到主页，并在个人中心能看到自己的个人信息	成功
2	用户登录信息错误	用户打开小程序，输入错误的微信授权登录信息，点击登录按钮	提示用户“登录信息错误，请重新输入”	成功

## 5.2 系统非功能测试

### （1）性能测试

通过模拟大量用户同时在线，观察系统能否正常运行，响应速度等是否达标。比如，设置一个用例，同时有 1000 用户在线浏览商品，系统是否运行正常，查看商品、下单、支付等操作是否会明显变慢。

### （2）稳定性测试（压力测试）

用于量化系统的崩溃阈值，评估它的稳定性。比如，模拟用户并发数不断增加，观察系统在何时会崩溃或响应速度变得无法接受。

### （3）可用性测试



测试系统是否用户友好，是否能够为用户提供便捷性操作？比如，检查和评估用户在使用微信小程序过程中的操作是否方便，界面设计是否合理。

#### （4）安全性测试

测试系统是否存在安全漏洞，比如登录功能是否对密码进行加密处理，用户在进行支付等操作时，数据是否通过安全协议进行传输，是否有防止非法入侵的设施等。

#### （5）容错性测试

检查当输入数据出错或者硬件、网络等出现故障时，系统是否能正确的处理，并提供相应的错误提示信息。

#### （6）适应性或兼容性测试

测试微信小程序在不同手机型号、不同操作系统版本、不同网络环境下的表现是否一致，体验是否流畅。

### 5.3 本章小结

在这一章节中，主要阐述了如何应用黑盒测试方法在微信小程序电商平台中进行软件测试。首先，基于系统的各个主要功能模块，如用户登录功能、商品模块、购物车添加模块等，均设定了一系列详尽的功能测试用例。通过模拟用户使用行为，来检验系统各项功能是否能正确且有效地服务用户。

接着，在非功能性测试部分，重点是评估系统的性能表现和稳定性。模拟了大量用户同时在线的情况，以及提升并发用户数量来压测系统的稳定程度，同时检测系统在各种条件下的容错性和兼容性。

由于小程序涉及到用户敏感信息的操作，同样十分注重系统安全性测试。检查了用户密码的安全处理、支付等关键操作的数据传输安全等方面。

进行以上所有测试工作的初衷，就是为了尽可能地发现并修复系统可能存在的问题，提升系统的稳定性和可靠性，并让用户更信任、满意于小程序。在整个测试过程中，始终保持严谨、细致的工作态度，以确保测试的质量和有效性，并为微信小程序电商平台的所有功能开发和优化提供可靠、有参考价值的数据和建议。

## 第6章 总结与展望

### 6.1 总结

本文通过深入的探讨和实践，成功构建了一款基于微信小程序的电商平台。在此过程中，不仅充分考察了电商领域的现行发展状况，还深入分析了用户的需求及行为模式，从而设计并实现了一套既具备丰富功能又能够提供顺畅用户体验的电商系统。此系统涵盖了从商品展示、搜索、详情查看，到用户认证、购物车管理、订单处理等众多方面。

研究强调了系统的性能优化、安全保障、以及如何提高用户体验这几个关键方面。通过采用 Vue3+TypeScript 和 UniApp 框架，并结合 Uncloud 云开发技术，不仅保证了系统的高性能和稳定性，也提升了系统的维护和扩展能力。此外，对用户界面进行了细心的设计，使其既美观又易于操作，确保了用户的高满意度。

本文的研究成果显示，我们成功提升了系统的性能表现，同时增强了平台的用户友好性，为用户提供了更加顺畅和愉悦的购物体验。

### 6.2 展望

展望未来，我们认为电商平台的发展将围绕多渠道融合和服务多元化展开。我们预期，将微信小程序与其他移动应用、如 H5、APP 等整合，将大幅提升业务的覆盖范围和用户的便利性。同时，通过对商业模式的持续创新，我们能够探索更丰富的服务和产品形态，实现市场的多样化需求。个性化推荐引擎和智能客服系统的研发将成为未来工作的重点，而社交媒体元素的融入则预计将带动用户参与度的显著提升。除此之外，电商平台的界面和交互设计同样需不断演进，以适应用户日益增长的审美需求和操作习惯。跨境电商业务的拓展更是开辟了一片潜力巨大的全球市场，我们将通过这些渠道向用户提供更广泛的产品选项和更精致的服务体验。究其根本，持续研究和技术创新是推动电商平台不断前行的源泉。

总而言之，虽然已经完成了一套功能完善的电商平台的设计与实现，但意识到在电子商务这一不断进化的领域里，总有进步和创新的空间。期待着未来能够继续在这个充满挑战和机遇的领域中进行更深入的研究和探索。

## 参考文献

- [1] 李杨攀. 输电线路巡检信息采集系统的设计与实现[D]. 电子科技大学, 2021.
- [2] 洪春美. 跨境电商贸易平台商业模式解析——以网易考拉为例[J]. 宿州教育学院学报, 2019, 22(06): 12-14+27.
- [3] 佟晶博. 基于神经网络预测算法的分销系统应用研究[D]. 大连交通大学, 2020.
- [4] 陈吉力. 基于神经网络的商品推荐系统的设计与实现[D]. 电子科技大学, 2019.
- [5] 渠渊惠. 基于 Spring Cloud 的全品类电商系统的设计与实现[D]. 华中科技大学, 2020.
- [6] 崔怡荷. 基于 Unity 3D 的虚拟陶艺体验系统设计与研究[D]. 西安电子科技大学, 2021.
- [7] 马潇. 基于 J2EE 和 Lucene 搜索架构的网络购物平台的设计与实现[D]. 东南大学, 2021.
- [8] 王晨. 基于微服务架构的电商平台的设计与实现[D]. 西安电子科技大学, 2021.
- [9] 冯一. 安全异常学习感知系统的设计与实现[D]. 北京邮电大学, 2020.
- [10] 胡康. 病案首页辅助系统的设计与实现[D]. 华中科技大学, 2021.
- [11] 龚瑾珂. 基于机器学习的僵尸网络检测系统的设计与实现[D]. 北京邮电大学, 2021.
- [12] 王甜甜. 研发管理平台的设计与实现[D]. 北京交通大学, 2022.
- [13] 邹文兵. 面向多营销模式的分布式电商平台设计与实现[D]. 厦门大学, 2019.
- [14] 蒋欢欢. 基于数据挖掘的医药电商平台的设计与实现[D]. 中国科学院大学(中国科学院沈阳计算技术研究所), 2022.
- [15] 赵陆. 支持多人协同工作的真实感虚拟装配训练系统[D]. 山东大学, 2019.

## 致 谢

在此，我愿用最温暖的文字，向那些在我学术旅程中给予我无限支持与帮助的老师、同学们表达我的深深感激。

首先，我要向我的指导教师，表达最深切的敬意与感谢。从研究伊始到论文完成，您不仅以您深邃的学识和丰富的经验为我指引方向，更以无比的耐心和细致的关怀，为我撑起成长与探索的天空。在您的指导下，我不仅收获了宝贵的知识，更学会了如何成为一个更好的研究者和思考者。

其次，提及我亲爱的家人，您们是我坚强的后盾。在这段漫长的学习路程中，是您们的理解和鼓励，让我能够在压力中保持平衡，在困惑中寻得方向。在此，我特别感谢我的父母，感谢您们无微不至的爱和对我的研究工作的支持，是您们的智慧和见识，常常在我的讨论中点燃思想的火花。

我也要感谢我的同学们，您们是我宝贵的精神财富。在我处于思路困顿时，是您们耐心的倾听和宝贵的建议，让我看到了问题的另一面，迸发了创新的思考。您们的陪伴与支持，是我研究道路上最美好的风景。

最后，我向所有给予我帮助和启发的教授和同学表示真挚的感谢。在各种学术交流和讨论中，您们的见解让我受益良多，我对此深感荣幸和感激。

在这个特殊的时刻，所有感激的话语似乎都不能完全表达我内心的情感。但请相信，这一路上的支持与陪伴，将永远镌刻在我心中，并鼓励我在未来的研究路上继续前进。

再次感谢您们。

符锦翰

2024年3月20日