

## 摘要

当前信息技术的发展及其迅猛，各方面的发展，用户消费升级，传统电商等方面的线上推广遇到了瓶颈。高额的店铺成本投入以及维护，加上技术和成本的门槛高也是比较高的。这时选择微信小程序，开发的微信小程序，对于传统商品通用，而且在消费者方面，需能够体验到方便快捷，应用范围更广。本论文是实现基于微信的微商城小程序，充分利用面向对象的开发思路。

基于微信的微商城小程序旨在实现以下功能模块。 首页模块：其中有轮播图，商品展示及搜索功能。商品分类模块，将商品实现分类，点击可达对应的分类页面。购物车模块，实现购物车商品具备一些基本的操作，比如增删操作。除此之外，还有的就是订单及地址管理模块。

该系统在布局方面利用了移动端最常用的 flex 布局，该布局方式容易上手，对小程序的兼容也很友善。采用了小程序的原生框架以及结合了微信小程序提供的云开发技术。微商城小程序运用小程序云开发技术，业务逻辑都可在小程序端完成，但不够强大的地方就是无法实现强大的 Web 管理界面。

**关键词：**微信小程序 商城 化妆品 云开发

## **Abstract**

The current development of information technology and its rapid development, the development of all aspects of user consumption upgrade, traditional e-commerce and other aspects of online promotion encountered a bottleneck. High store cost inputs and maintenance, coupled with high technology and cost thresholds are also high. At this time, choose the WeChat app, developed by WeChat app, for traditional goods, and in the consumer side, need to be able to experience the convenience and speed, the application range is wider. This thesis is to implement a WeChat-based WeChat applet, making full use of object-oriented development ideas.

The WeChat-based WeChat applet is designed to implement the following functional modules. Home module: with rotating map, product display and search function. The product classification module enables you to classify products and click the corresponding classification page. The shopping cart module, which implements some basic operations for shopping cart products, such as adding and deleting operations. In addition to this, there is the order and address management module.

The system makes use of the most common flex layout on the mobile side in terms of layout, which is easy to use and friendly to applet compatibility. The native framework of the applet and the cloud development technology provided by the WeChat applet are used. The micro-mall applet using the applet cloud development technology, business logic can be done in the applet side, but not powerful enough is not able to achieve a powerful web management interface.

**Key words:** Wechat Public Platform tourism Travel information  
Lightweight Sharing Community

# 目录

<b>第一章 绪论</b> .....	1
1.1 课题背景与研究意义 .....	1
1.1.1 课题背景 .....	1
1.1.2 研究意义 .....	1
1.2 研究现状 .....	1
1.3 本课题的研究方法 .....	2
1.4 论文结构 .....	2
1.5 本章小结 .....	3
<b>第二章 系统相关技术研究</b> .....	4
2.1 开发工具及技术选型 .....	4
2.2 微信小程序 .....	4
2.3 相关技术 .....	4
2.3.1 JavaScript .....	4
2.3.2 CSS 技术 .....	4
2.3.3 HTML 语言 .....	5
2.3.4 小程序云开发 .....	5
2.4 本章小结 .....	5
<b>第三章 系统分析</b> .....	6
3.1 微商城小程序业务流程分析 .....	6
3.2 微商城小程序总体功能分析 .....	7
3.3 微商城小程序详细需求分析 .....	7
3.3.1 用户登录模块 .....	7
3.3.2 商品信息展示模块 .....	8
3.3.3 商品分类模块 .....	9
3.3.4 购物车模块 .....	10
3.3.5 下单支付模块 .....	11

3.3.6 个人中心模块 .....	12
3.4 本章小结 .....	13
<b>第四章 微商城小程序的设计 .....</b>	<b>14</b>
4.1 微商城小程序项目的搭建 .....	14
4.1.1 准备工作 .....	14
4.1.2 新建项目 .....	14
4.2 微商城小程序的数据库设计 .....	15
4.2.1 数据库概念结构设计 .....	15
4.2.2 数据库的逻辑结构设计 .....	18
4.3 本章小结 .....	20
<b>第五章 微商城小程序的具体实现 .....</b>	<b>21</b>
5.1 商品信息展示模块设计与实现 .....	21
5.1.1 首页轮播图 banner .....	21
5.1.2 首页搜索框 .....	22
5.1.3 首页热门商品展示 .....	23
5.2 商品分类模块设计与实现 .....	25
5.2.1 分类模块 .....	25
5.2.2 商品详情模块 .....	26
5.3 购物车模块设计与实现 .....	26
5.4 下单模块设计与实现 .....	28
5.5 微商城小程序个人中心管理模块的设计与实现 .....	29
5.5.1 登录获取用户基本信息 .....	29
5.5.2 个人地址管理 .....	30
5.5.3 个人订单管理 .....	31
5.6 商品管理模块的设计与实现 .....	32
5.7 本章小结 .....	32
<b>第六章 系统测试 .....</b>	<b>34</b>

6.1 软件测试的目的和流程 .....	34
6.1.1 权限测试 .....	35
6.1.2 功能测试 .....	35
6.1.3 网络测试 .....	37
6.2 本章小结 .....	37
<b>总结</b> .....	<b>38</b>
<b>参考文献</b> .....	<b>39</b>
<b>致 谢</b> .....	<b>41</b>



## 第一章 绪论

### 1.1 课题背景与研究意义

#### 1.1.1 课题背景

微信自推出到现在，已经成为即时通讯软件的佼佼者。微信小程序，它依托微信软件，无需另外安装下载即可快速使用，一经发布，就掀起了互联网的新热潮。

微信小程序，具备用完即走，无需安装的特性，在当前世界发展环境下，它的出现是非常有意义的。由于小程序的特性，它使用的时候对手机占用内存是很友好的，用户也不需要去担心加载速度问题，它的出现，是顺应时代发展趋势的，它存在的优势，是不容小觑的。

2016年，张小龙向人们解读微信所存在的意义。微信小程序在17年的1月发布，新华社微悦读，是最先发布的，同时它也是首款新闻类的小程序。

微信小程序具有以下的几种应用模式：

(1) 微信小程序自带模糊搜索。微信公众号和小程序之间是能够相互跳转的。根据这一特点，它们之间也相互促进，公众号的关注度高了，那么与之所绑定的小程序，也会受益。

(2) 微信小程序的推广。微信小程序的推广，跟其他的一些应用相比，是存在优势的，它是基于微信的，能够实现在好友间快速分享，推广也更加方便。

(3) 微信小程序可以与公众号建立绑定。在进入相关微信公众号，倘若与其存在绑定关系的小程序，拥有着类似的主题，就可以在小程序看到相关的一些内容的推送。

#### 1.1.2 研究意义

如今现代信息技术发展是及其迅速的，现在的日常生活中，手机的发挥的重用也是及其重要，其中，手机软件也是不可或缺的一部分。微信小程序，它具有操作方便、共享方便、随时可用等特点，使得购物更加的方便。微信小程序的火热发展，也在一定程度上推动了网上购物的发展，个人商家在拥有了各自的商城小程序之后，借助其来推动了发展。由此可以看出，商城小程序的优势是存在的。

本课题要探究制作的是化妆类的微商城小程序，小程序的主要目标是简单高效，回归简单。小程序的意义在于方便用户搜索、浏览、购买化妆品，减少外出购物不便。

### 1.2 研究现状

现在的发展跟以往相比，可谓是翻天覆地。互联网的普及，还有在支付方式上的各种改变。从传统的 PC 端，到移动端后来居上，还有其他各式各样的智能化得以普及发展，这些都在一定程度上加速了发展效率，对于人们的生活来说，也感受到了明显的便利。

微信发布以后，也迅速的成为了社交平台的巨头，紧接着，微信公众平台等各种新的活力应运而生。借助着微信这一大社交平台，各种生活服务也得到不一样的体验。然后就是微信小程序的发布，这一款轻量级的应用，随用随走，开启了全新的体验。微信小程序是一种新的软件模式，一种用户与服务相对应的新模式，它与传统 PC 端以及移动端的 APP 是存在区别的，是一个新的物种。

当前发展的趋势，是趋向更加智能化，更加便利的，更加方便人们生活，也能推动进一步的发展。伴随着智能化的发展，人们也愈加离不开手机，由于生活需要，购物这一环节也是不可避免的。许多智能化应用的出现，推动了线上以及线下两种购物方式得以有相互结合，相互促进。更加的去关注两者的结合，也成了非常看重的一方面。

借助微信的庞大的用户量，微信小程序的轻量，然后再去将线上线下购物结合，成为了一个新的机遇，一场新的变革。商家们借助微信小程序，拥有了各种的商城小程序，与时代前进步伐统一，进一步推动发展。

放眼未来，可以看得出，小程序的光明前景是显而易见的，在这个互联网发展的时代，商城小程序也更值得去关注。

### 1.3 本课题的研究方法

本人通过了以下的方法对本课题进行了研究：

- (1) 文献研究法：查阅并初步了解现有文献，初步掌握选题检索的相关内容和技巧，确定选题的目标、任务和内容；
- (2) 定位分析法：通过对文献的研究，深入了解开发的相关技术，调查基于微信的微商城小程序业务流程，从而作出对应的方案，并确定开发流程；
- (3) 技术研究：使用小程序云开发技术，开发本系统；
- (4) 测试研究：对开发的系统进行必要测试，以确保系统能够正常运行。

### 1.4 论文结构

第 1 章：论绪。阐述本文的研究背景以及意义，指出利用到的研究方法，并且分析了研究现状，还给出了本文的组织。

第 2 章：系统开发相关理论和技术的研究。本章主要介绍一个项目系统开发的具体开发过程和环境，以及对本系统所需的关键技术进行介绍。

第 3 章：微商城小程序的需求分析。



第4章：微商城小程序的设计。本章是设计系统的各个模块。

第5章：微商城小程序的具体实现。本章是基于各个模块去阐述微商城小程序的具体实现过程。

第6章：软件测试。

第7章：总结。

## **1.5 本章小结**

本章是论绪部分。本章对微信小程序的背景、优点进行了阐述，同时还列出了本课题的研究方法、思路以及论文的整体纲要。前两节对项目的背景、意义，研究现状等进行阐述。后面两节指出研究方法，还有论文撰写的组织结构，确认好组织框架，进一步确定项目模块设计实现的计划。

## 第二章 系统相关技术研究

### 2.1 开发工具及技术选型

- (1) 开发工具：微信开发者工具、Sublime。
- (2) 技术选型：基于微信小程序的原生框架，结合小程序云开发 API 。

### 2.2 微信小程序

微信小程序，它能够与 App 进行比较，相比较之下，能发现两者是比较类似的。微信小程序可以视为是服务，我们使用的时候它就出现，不用了它就消失，用完即走。如今微信小程序拥有相当高的应用范围，但是对于普通开发者来说，小程序目前的变现渠道还是比较单一的，即通过广告点击。<sup>[6]</sup>

### 2.3 相关技术

#### 2.3.1 JavaScript

JavaScript，它是一种解释性语言，简称是“JS”。JavaScript 作为开发 Web 页面的一种脚本语言，这种脚本语言，它是基于原型编程的，也是动态的，它不仅应用于浏览器的环境，对于不少非浏览器的环境，它也是同样适用的。

1995 年，Netscape 公司的 Brendan Eich，JavaScript 首次设计成功，并将它应用在网景导航者的浏览器上。同时，Netscape 公司还与 Sun 公司有所合作，管理层希望其外观看起来像 Java，因此将其取名为 JavaScript，但是实际上这两者在语法风格上面，是有着较大差异的。

JavaScript 主要是用来向 HTML 页面添加交互行为的，它同时还具备跨平台的一个特性。它与其他语言一样，也拥有着自身的一些基本数据类型，表达式等，它还可以实现 web 页面的人机交互。

#### 2.3.2 CSS 技术

在 HTML 出现之后，承载 HTML 的浏览器得以迅速发展，CSS 也应运而生。随着 HTML 的发展，增加了很多的显示功能，这都是为了去满足设计师的一些要求，然而这些功能的增加，使得外部样式语法的作用愈加无意义。

在 1994 年 10 月，Hankon Wium Lie 最初提出 CSS 的建议，随后的 96 年 12 月，推出了第一版的 CSS。

随后 CSS 技术不断发展，也拥有了自己的一套规则，它被称为层叠样式表，可以为结构化文档来添加样式。它不需要编译，浏览器是能够直接解释执行的。CSS 是由 W3C 的 CSS 工作组所产生和维护的。

CSS 语言有以下的几个特点：

（1）容易使用以及修改。通过 CSS 样式表，能够统一存放所有的样式声明，来进行统一的管理。

（2）具备丰富的样式定义。CSS 提供丰富的文档样式外观，允许更加需求随意改变文本大小、修饰方式等来实现各种页面效果。

（3）多页面应用。CSS 样式表是能够单独存放到一个文件当中的，这样方便多个页面使用同一个样式，来实现多个页面的风格一致。

### 2.3.3 HTML 语言

HTML，它是一种网页语言，也是一种超文本标记语言，其中，超文本的含义是指超出文本的范畴能够使用 html 来轻松实现。HTML 会自己的一套遵循的规范，至于它的操作思路，根据网页中不同的数据需要不同显示效果，需使用对应的标签将需要操作到的数据封装起来，然后通过其属性，来进一步的实现内部数据样式的改变。

### 2.3.4 小程序云开发

小程序云开发，它是无服务器的一种服务，为开发人员提供了“云函数”，“云数据库”以及“云文件存储”，并且还将这些功能封装到了 wx 的特定接口当中，开发者能够通过 cloud.xxx 的形式来进行调用。整套功能是基于腾讯云全新推出的云开发（Tencent Cloud Base）所研发出来的一套完备的小程序后台开发方案。<sup>[16]</sup>

## 2.4 本章小结

在本章中，主要研究说明了微商城小程序开发所需要用到的关键技术。包括微信小程序的框架，对小程序开发所需的一些基础的简要介绍，以及小程序云开发的介绍。这些都是本次项目开发中所需要用到的关键理论基础。

## 第三章 系统分析

### 3.1 微商城小程序业务流程分析

在开发微商城小程序之前，首先要进行分析，完成开发和设计微商城小程序客户端以及数据库。在微商城小程序的客户端部分，客户是能够进行浏览商品操作，还能对商品的分类进行查看、同时也能够执行添加商品至购物车、商品快速搜索操作，还有一些其他的功能，比如管理购物车，管理个人地址信息等。然后呢，对应数据库的部分，确保通过合理可行的设计，从而使得对应的数据表之间，拥有合理可靠的关联关系，当然也确保了数据库可扩展性。

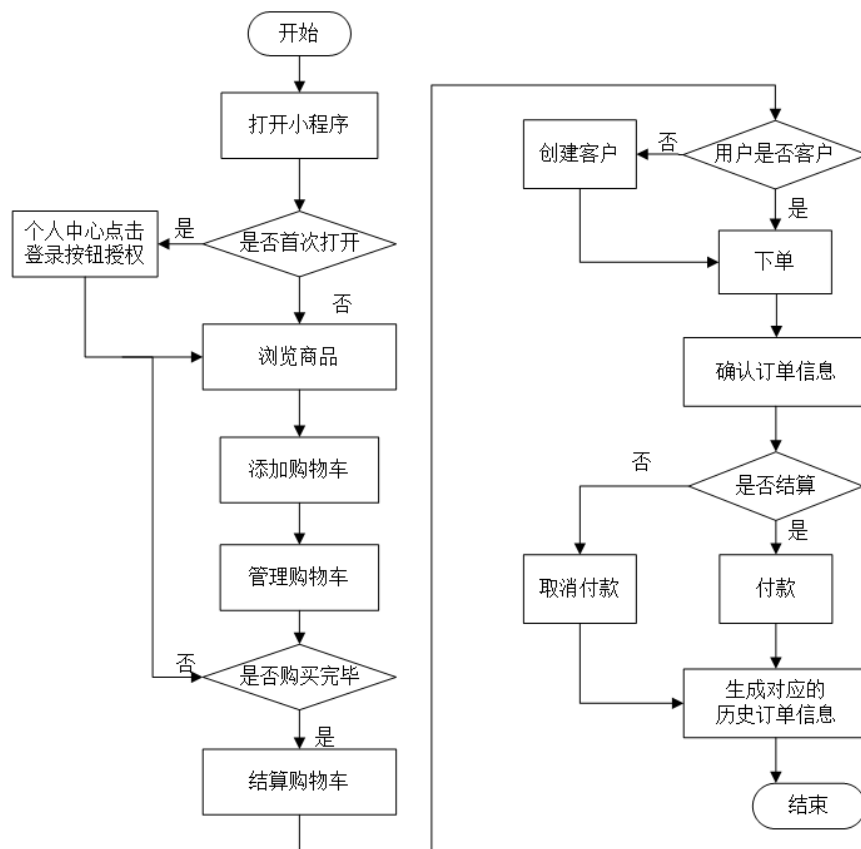


图 3.1 微商城小程序的购物流程图

如图 3.1 所示，是基于微信的微商城小程序的购物流程图，用户获取并且进入到小程序，以游客身份或客户端用户的身份，都具备浏览商品的权限。获取授权登录之后，客户端用户可以选择商品添加到购物车，并且能够进行管理。用户选择需要购买的商品后，即可进行商品的结算，确认完订单信息，执行付款操作，不管用户是否付款，都会生成历史订单信息，生成的历史订单信息也会根据支付结果有所不同。到这里之后，购物流程也就结束了。

### 3.2 微商城小程序总体功能分析

微商城小程序的总系统功能模块图为下图 3.2:

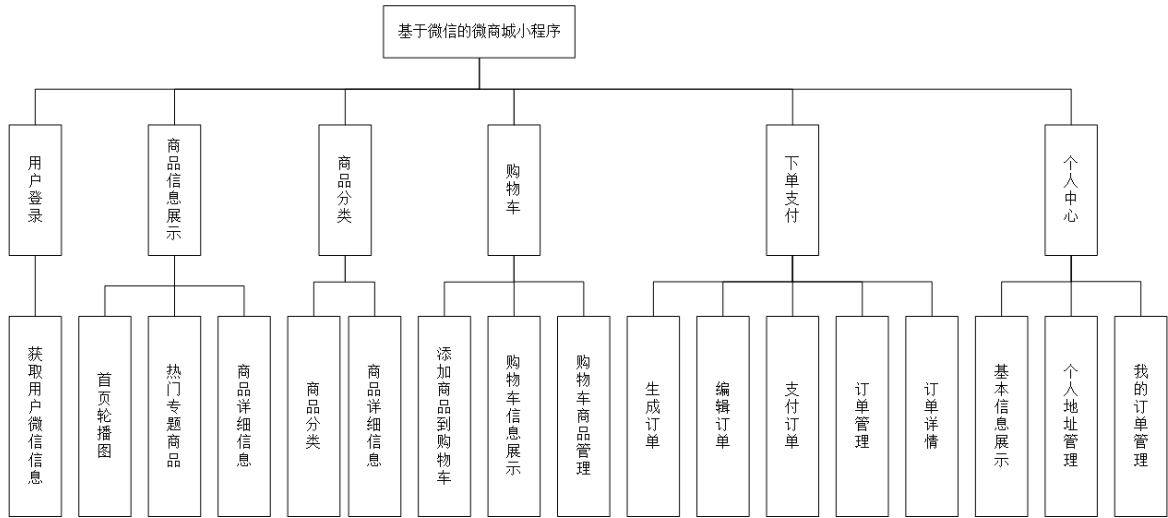


图 3.2 微商城小程序的总体功能结构

### 3.3 微商城小程序详细需求分析

#### 3.3.1 用户登录模块

微商城小程序的用户系统功能用例图如下图 3.3 所示。



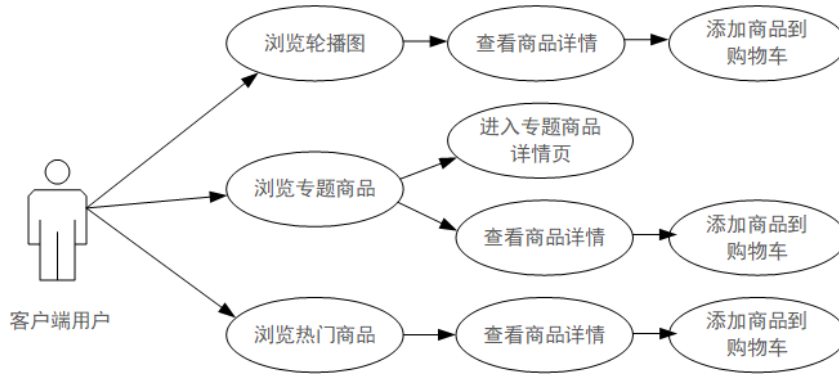


图 3.4.1 用户浏览商品的用例图（客户端用户）

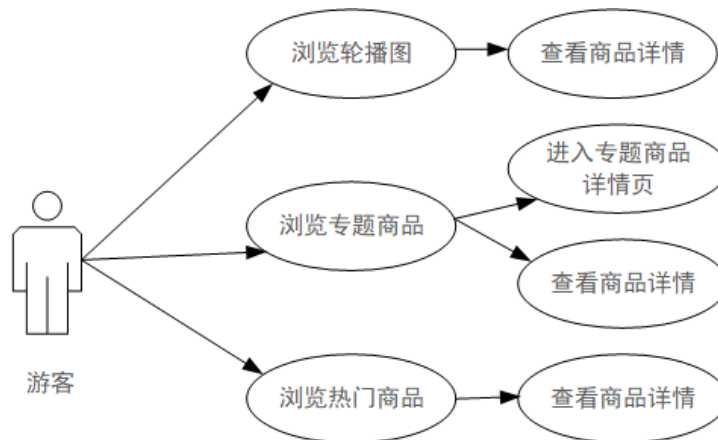


图 3.4.2 用户浏览商品的用例图（游客）

用户浏览商品信息功能需求分析：

- (1) 用户进入小程序，拥有浏览商品权限。
- (2) 用户在首页可以浏览的内容有：轮播图，专题商品，热门商品。
- (3) 轮播图在首页的头部，实现其自动播放滚动的效果。

用户微商城小程序的“首页”页面，可以选择执行上拉刷新操作，去获取并浏览最新的商品。

(4) 点击商品，可以到达对应的商品详情展示，其中包括商品的图片、价格、产品参数等内容，用户也能够在详情页面，选择立即购买或者先将商品添加到购物车。

### 3.3.3 商品分类模块

用户在微商城小程序进行商品浏览的同时，能够通点通过击底部导航栏的分类图标，进入到微商城小程序的商品分类页面。在商品分类页面，对商品进行了对应的分类，方便用户寻找相应商品。图 3.5.1 以及图 3.5.2 为微商城小程序的商品分类用例图。

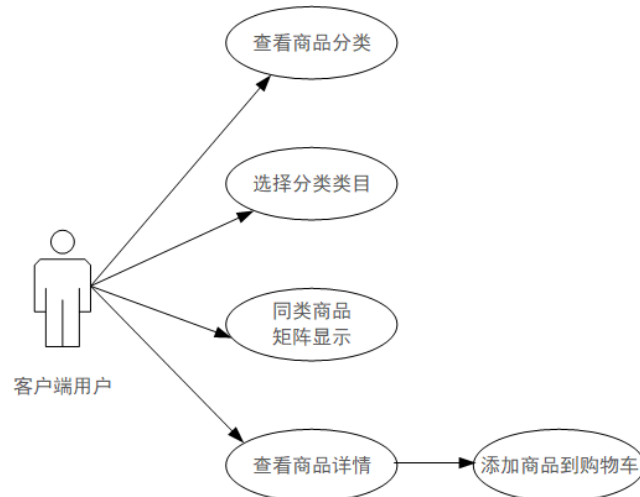


图 3.5.1 微商城小程序用户的商品分类用例图（客户端用户）

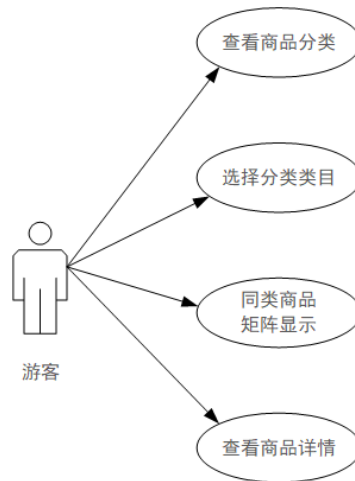


图 3.5.2 用户商品分类用例图（游客）

商品分类功能需求分析：

- (1) 用户能够在分类页面浏览展示的分类商品。
- (2) 商品是根据商品的功效等进行划分的，用户通过点击“分类”页面的分类栏左半部分对应的分类词，右半部分会矩阵显示对应的商品类目。
- (3) 右半部分商品分类类目展示后，用户可以通过点击类目名称来跳转到对应的分类专题。
- (4) 用户在对应的商品分类专题可以查看商品的详细信息。

### 3.3.4 购物车模块

客户端用户通过选择，让自己想要的商品加入其购物车当中，微商城小程序的底部导航栏，有“购物车”图标，可以方便用户快捷的进入其购物车进行对应的管理。微商城小程序的购物车的用例图如图 3.6 所示。



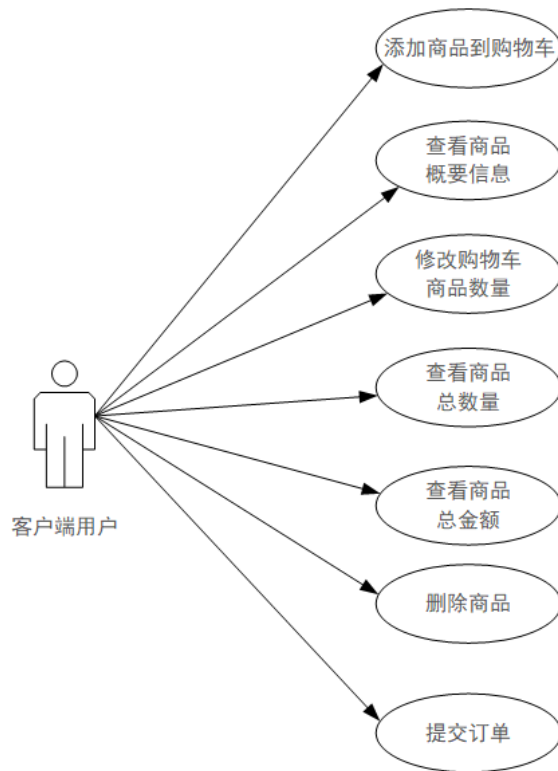


图 3.6 微商城小程序购物车的用例图

购物车模块具体的功能需求分析如下：

- (1) 用户只有允许登录授权了，才能够选择商品添加到购物车。
- (2) 用户允许登录授权之后，可以在购物车查看到已经添加的商品，购物车页面主要展示内容设计包括商品的概要信息（商品图片、价格、数量等），全选、单选、结算按钮。
- (3) 在购物车管理页面，用户能够手动处理商品的数量，在商品的选择方面，有单选、多选（全选），用户选中对应的商品左侧的复选框之后，右下角结算按钮上方就会出现删除按钮，可供用户执行删除操作。

### 3.3.5 下单支付模块

图 3.7 所示是微商城小程序的用户下单支付对应用例图

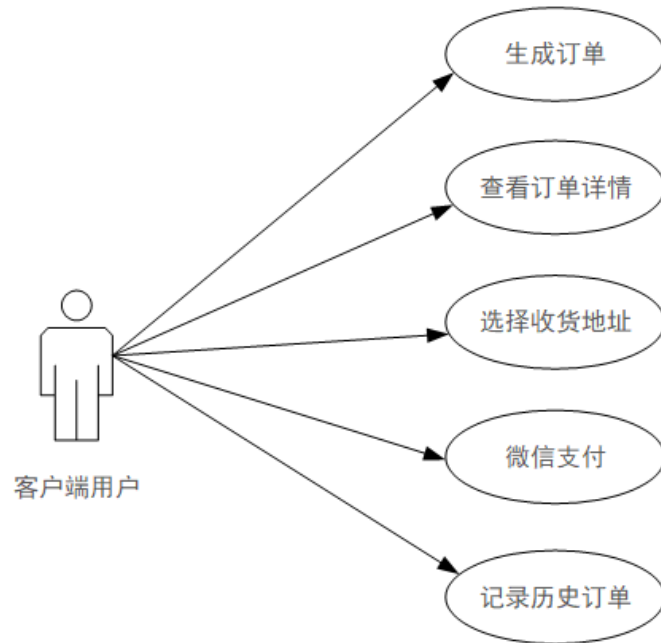


图 3.7 微商城小程序下单支付的用例图

下面是下单支付的功能需求分析：

- (1) 用户在购物车的列表当中进行选择，选择完毕，可以选择执行结算。
- (2) 用户只有允许登录授权了，才能够执行结算下单。进入到对应的确认订单页面，首先进行的操作是地址的选择。如果登录用户已有默认地址，则直接选择默认地址进行显示；如果没有默认地址，则是会提示用户进行选择地址的操作。
- (3) 当用户在确认订单页面选择地址完毕，点击付款按钮，生成订单，订单生成，云数据库中的订单表就会增加一条订单记录，购物车中选中的商品也删除。
- (4) 不管用户是否付款，都会生成历史订单信息。

### 3.3.6 个人中心模块

用户进入微商城小程序，点击底部导航栏的“我的”，可以进入到个人中心页面。图 3.8 为微商城小程序用户个人中心管理的用例图。

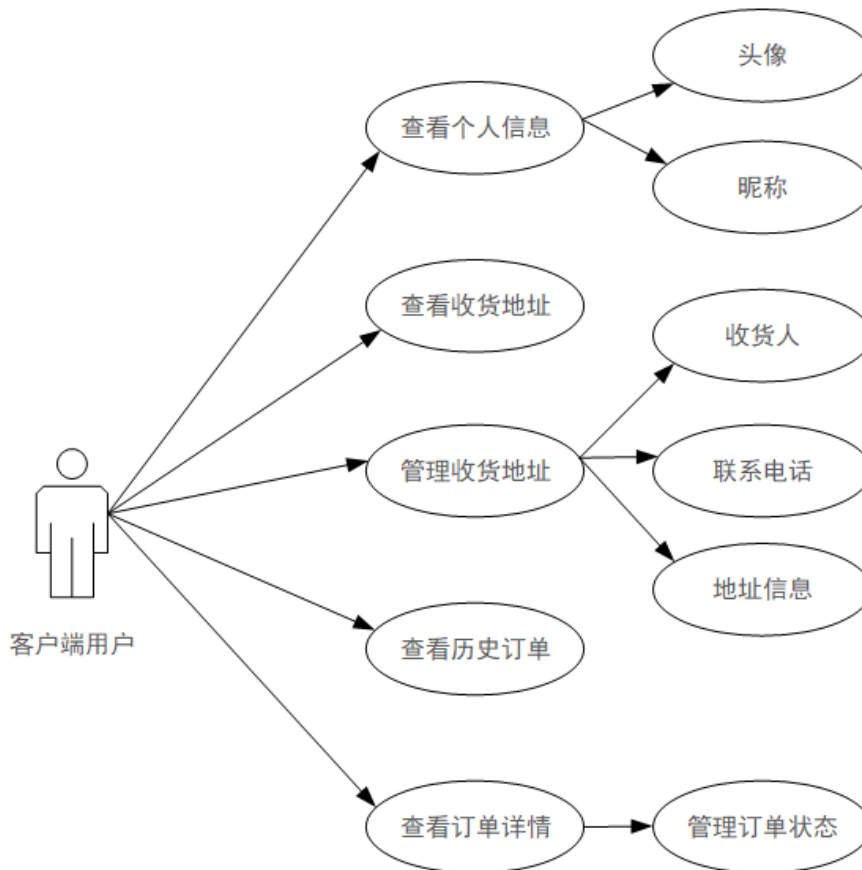


图 3.8 微商城小程序用户个人中心管理的用例图

个人中心模块具体的功能需求分析如下：

(1) 用户进入微商城小程序后，可以通过点击底部导航栏的“我的”，进入到个人中心页面。

(2) 用户未允许授权登录，用户为游客身份，个人中心页面会有“登录”按钮显示，以及下方的地址管理，订单管理等。当用户点击登录按钮允许授权了，就会展示用户微信头像以及昵称，个人地址管理和订单管理以及客服只有在用户允许登录了才可以进行查看。

(3) 用户能够通过点击个人地址管理，去进入到相关页面执行操作。

(4) 用户通过点击订单管理，能够去到相关的页面执行订单管理的操作。

### 3.4 本章小结

该章节主要是进行了微商城小程序的系统需求分析。其中 3.1 小节是对微商城小程序整体的业务流程分析，而 3.2 小节则是微商城小程序的总体功能分析，3.3 小结节是对微商城小程序进行详细的需求分析。分析过程中，充分利用了 UML 用例图，开发微商城小程序，需求分析这一关是至关重要的，通过本章节也为下面的微商城小程序功能的设计与实现搭好一定的基础。

## 第四章 微商城小程序的设计

### 4.1 微商城小程序项目的搭建

#### 4.1.1 准备工作

- (1) 申请个人小程序，获取到小程序 AppID。
- (2) 下载安装并且熟悉开发工具。
- (3) 点击开发者工具中的云开发，开通小程序云开发。
- (4) 通过在终端安装 node 环境，方便后续使用小程序云开发的云函数。

#### 4.1.2 新建项目

- (1) 在开发者工具新建不使用云服务的项目，如图 4.1 所示：

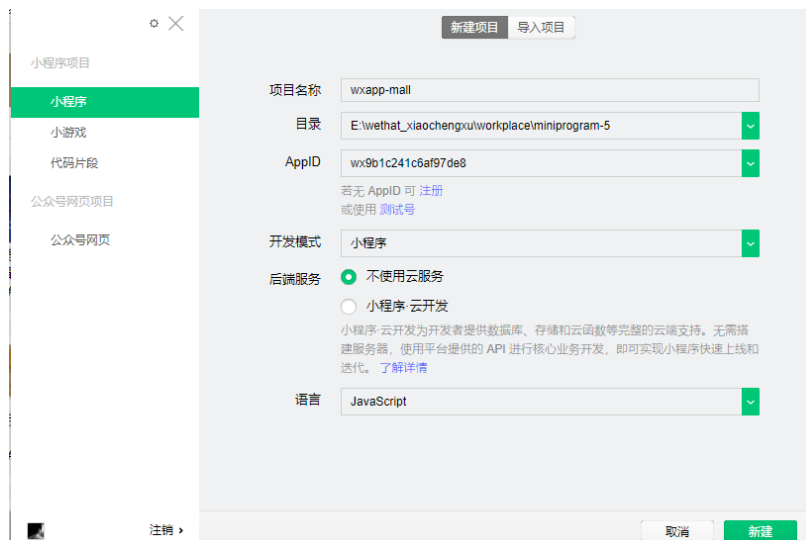


图 4.1 微商城小程序的项目新建

- (2) 删除官方的图片和模板，创建好项目所需的文件，进行全局配置，导入项目所需图片，并且初始化云开发环境。图 4.2.1 为项目的文件结构，图 4.2.2 为初始化云开发环境。

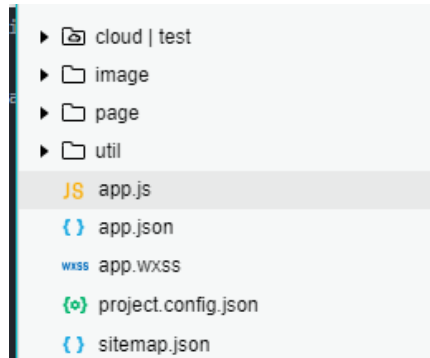


图 4.2.1 微商城小程序的项目文件结构

```
2  App({
3  onLaunch: function () {
4      this.setTabbar()
5      console.log('App Launch')
6      // 云开发环境初始化
7  wx.cloud.init({
8      env: 'test-x13ah',
9      traceUser: true
10 })
```

图 4.2.2 云开发环境初始化

## 4.2 微商城小程序的数据库设计

微商城小程序使用云数据库，它是小程序云开发所提供的。它在权限方面并不会很复杂，如下：

- (1) 仅创作者可写作。
- (2) 仅创建者可读写。
- (3) 仅管理端可写，所有人可读。
- (4) 仅管理端才能读写。

在很多情况下，简单使用这四种权限是无法完全满足开发人员的需求的，因此还需要在代码级别去进行一些适当的判断，来确保具体的表现与开发人员的想法相符合。

### 4.2.1 数据库概念结构设计

- (1) 商品信息的实体图如图 4.3:

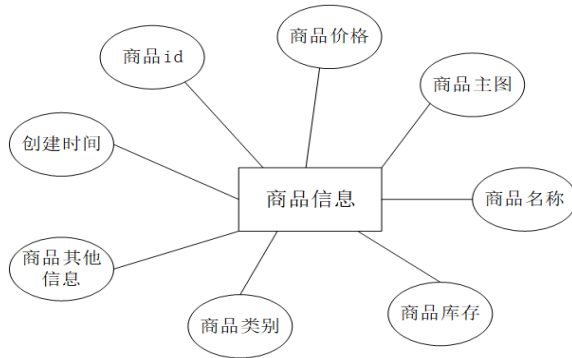


图 4.3 商品信息实体图

(2) 用户信息实体图如图 4.4:

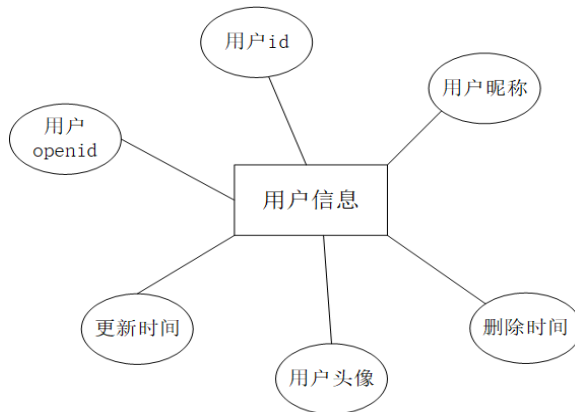


图 4.4 用户信息实体图

(3) 地址信息实体图如图 4.5:

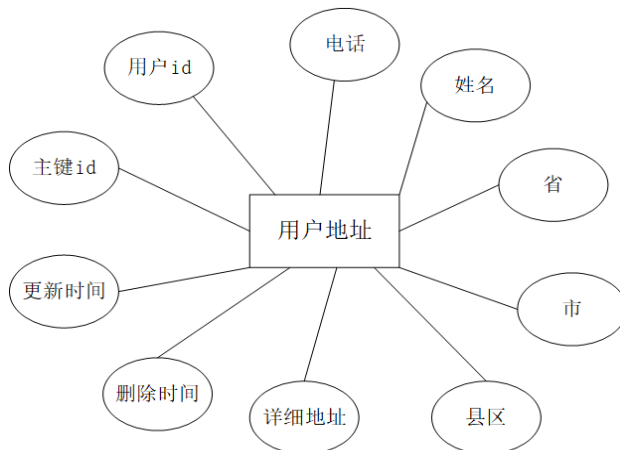


图 4.5 地址信息实体图

(4) 商品分类实体图如图 4.6:

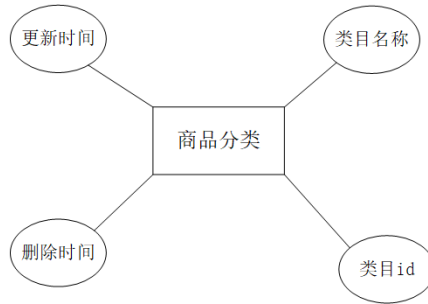


图 4.6 商品分类实体图

(5) 订单商品信息的实体图如图 4.7:

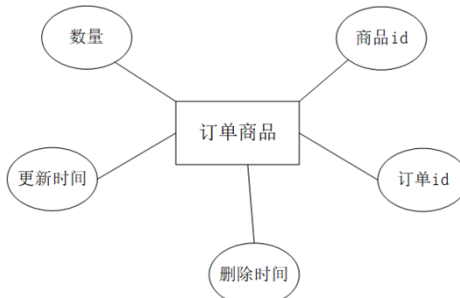


图 4.7 订单商品信息实体图

(6) 用户订单信息的实体图如图 4.8:

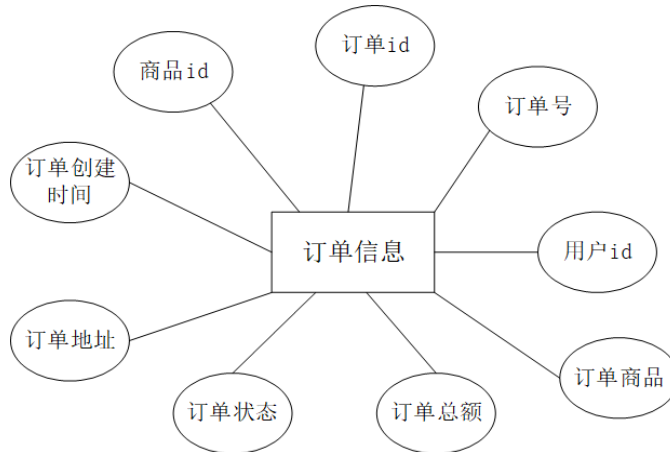


图 4.8 用户订单信息实体图

(7) 由系统实体与实体之间的联系能够得出系统 E-R 图，微商城小程序的系统总体 E-R 图如图 4.9 所示。

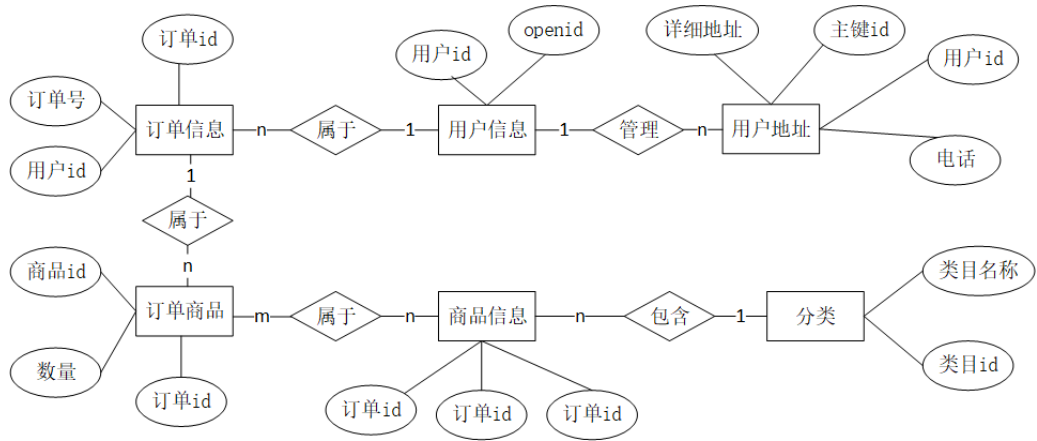


图 4.9 微商城小程序系统总体 E-R 图

## 4.2.2 数据库的逻辑结构设计

(1) 轮播图表 (banner), 如表 4.1 所示:

4.1 轮播图表

字段名	类型	能否为空	是否为主键	描述
Id	string	否	是	轮播图 id
Img	String	否	否	主图
Name	String	否	否	轮播图名称
Show	Boolean	否	否	是否显示
Create_data	data	否		创建时间
Update_data	data	否		更新时间

(2) 商品信息表 (products) 如表 4.2 所示:

4.2 商品信息表

字段名	类型	能否为空	是否为主键	描述
id	string	否	是	商品 id
product_name	String	否	否	商品名称
product_price	Number	否	否	商品价格
Product_img	String	否	否	商品主图
Product_stock	Number	否	否	商品库存
Category_type	Number	否	否	商品分类



Description	Object	否	否	商品描述
Create_data	Data		否	创建时间

(3) 用户信息表 (userInfo) 如表 4.3 所示:

4.3 用户信息表

字段名	类型	能否为空	是否为主键	描述
id	string	否	是	用户 id
openid	String	否	否	用户 openid
User_img	String	否	否	用户头像
Nick_name	Number	否	否	用户昵称
Update_time	data		否	更新时间
Delete_time	Data		否	删除时间

(4) 订单信息表 (orders) 如表 4.4:

4.4 地址信息表

字段名	类型	能否为空	是否为主键	描述
id	string	否	是	订单 id
Order_num	String	否	否	订单号
User_id	string	否	否	用户 id
Order_product	Object	否	否	订单商品
Order_status	String	否	否	订单状态
Order_address	Object	否	否	订单地址
Order_amount	Number	否	否	订单总额
Create_data	Data		否	创建时间

(5) 用户地址信息表 (address) 如表 4.5:

4.5 用户信息表

字段名	类型	能否为空	是否为主键	描述
id	string	否	是	地址 id
User_id	String	否	否	用户 id
name	String	否	否	收货人姓名
telNumber	String	否	否	手机号
provinceName	String	否	否	省
cityName	String	否	否	市
countyName	String	否	否	县区
detailInfo	String	否	否	详细地址
Update_data	data		否	更新时间
Delete_data	Data		否	删除时间

(6) 商品分类表 (category) 如表 4.6 所示:

4.6 商品分类表

字段名	类型	能否为空	是否为主键	描述
id	string	否	是	类目 id
Name	String	否	否	类目名称
Update_data	Data		否	更新时间
Delete_time	Data		否	删除时间

### 4.3 本章小结

本章节为微商城小程序的设计。在 4.1 小节进行了微商城小程序的项目搭建，接着在 4.2 小节完成了数据库的设计与实现，微商城小程序使用的是小程序云开发技术，数据库即为小程序云开发的云数据库。

## 第五章 微商城小程序的具体实现

### 5.1 商品信息展示模块设计与实现

用户进入到微商城小程序“首页”页面，页面中的内容为头部的 banner 轮播图，轮播图下面是搜索框，接着为专题商品，底部的是热门商品的展示。

#### 5.1.1 首页轮播图 banner

“首页”页面头部的轮播图模块，使用的是 swiper 组件和 swiper-item 组件，其中设置了固定时间进行自动循环播放，也显示了随之切换的小圆点。

界面实现的代码如下：

```

<!-- 轮播图模块 -->
<swiper indicator-dots="true" autoplay="true" interval="{{interval}}" duration="{{duration}}
" circular="true">
  <!-- 遍历展示轮播图数据 -->
  <block wx:for="{{imgUrls}}" wx:key="index">
    <!-- 点击轮播图跳转到分类栏 -->
    <navigator url="/page/category/category" open-type="switchTab">
      <swiper-item>
        <image src="{{item.image}}" class="slide-image" width="100%"/>
      </swiper-item>
    </navigator>
  </block>
</swiper>

```

图 5.1.1 微商城小程序首页轮播图

轮播图的数据是存放在小程序云开发后台的云数据库中，在首页页面首次加载时，就调用 API 获取云数据库中的轮播图数据，调用 this.setData 将获取到的轮播图数据绑定到页面初始数据中，并在首页页面进行渲染。

```

// 获取轮播图的数据
db.collection('banner').get().then(res=>{
  // 获取成功
  console.log('获取轮播图数据成功', res.data)
  // 将轮播图数据展示到首页
  this.setData({
    imgUrls: res.data
  })
})

```

图 5.1.2 微商城小程序首页轮播图



图 5.1.3 微商城小程序首页轮播图运行效果图

### 5.1.2 首页搜索框

首页的搜索框界面设计是利用了 view 组件，对图标的使用，能够点击跳转到搜索界面，进行精准搜索。

主要是利用小程序云开发的正则查询方法，利用此实现模糊搜索。执行效果图如图 5.2.1 和图 5.2.2。



5.2.1 微商城小程序搜索运行效果图



5.2.2 微商城小程序搜索运行效果图

### 5.1.3 首页热门商品展示

首页热门商品的展示，利用了小程序的列表渲染，点击某个商品，可以跳转到对应的商品详情页面，利用了页面跳转传参，将对应的商品 id 传到商品详情页面，即可展示对应的商品详情。图 5.3.1，图 5.3.2，图 5.3.3 分别为界面实现代码图、逻辑实现的一部分代码图、首页热门商品展示效果图。

```

<!-- 热门推荐商品模块 -->
<view class="newest">
  <view class="newest-title"><text>热门推荐</text></view>
  <view class="newest-box">
    <block wx:for-items="{{hotGoods}}" wx:key="index">
      <view class="newest-list" id="{{item.id}}" bindtap="switchToDetail">
        <image src="{{item.product_img}}" lazy-load</image>
        <view class="newest-text"><text>{{item.product_name}}</text></view>
        <view class="newest-text"><text>¥{{item.product_price}}</text></view>
        <button class="newest-add" data-index="{{index}}" data-item="{{item}}"
open-type="getUserInfo" bindgetUserinfo="getUserInfo" catchtap="addToCart">+</button>
      </view>
    </block>
  </view>
</view>

```

图 5.3.1 微商城小程序首页热门商品展示

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/308040130033006052>