未命名

**2018届毕业设计(论文)**

|  |
| --- |
| **论文题目: 微信食品商城的设计与实现** |
|  |

|  |  |
| --- | --- |
| **学 院 :** | **信息与电子工程学院** |
| **专 业 :** | **软件工程** |
| **班 级 :** | **141班** |
| **学 号 :** | **1140299141** |
| **学生姓名 :** | **鲁钺锋** |
| **指导教师 :** | **孙奕鸣** |

**二○一八 年 五 月**

# 摘要

移动互联网蓬勃发展，即时通讯应用由个人电脑转向移动设备。而微信正是移动通讯的最鲜明代表，其作为一款跨平台及时聊天工具，在短短几年时间，凝聚了男女老少大批用户。商家利用微信的种种优点并以微信公众平台作为自己商城的展示工具，形成一种主流的线上营销模式，不仅省去了实体店铺的租金成本，而且有微信这个媒介，使得对于商城、商品的营销推广相较于传统方式更为简洁有效。

本系统前端主要采用现阶段最流行的vuejs框架，后端主要选择java作为编程语言采用spring、Mybatis、java、mysql等优秀的开发技术和数据库来实现本系统的开发。该微信食品商城主要包括商品展示、商品购买、用户管理、订单管理、购物车管理以及分享获利等功能。

关键词：电子商城，微信公众平台，vue，java

# ABSTRACT

The mobile Internet is booming, and instant messaging applications are switched from personal computers to mobile devices. And WeChat is the most vivid representative of mobile communications, as a cross platform and timely chat tool, in a few short years, a large number of men and women, young and young users. Using the advantages of WeChat and taking the WeChat public platform as the display tool of its own business city, the merchants form a mainstream online marketing model, which not only saves the rent cost of the entity stores, but also has the medium of WeChat, which makes the marketing and promotion of the commercial city and commodity more concise and effective than the traditional way.

The front end of this system mainly uses the most popular vuejs framework at the present stage, and the back end mainly selects Java as the programming language to use spring, Mybatis, Java, MySQL and other excellent development technology and database to realize the development of this system. The WeChat food mall mainly includes display, merchandise purchase, user management, order management, shopping cart management and sharing profit.

**Keywords**: electronic mall,WeChat public platform, vue , java.

目录

[摘要 I](#_Toc11648)

[ABSTRACT II](#_Toc20470)

[第1章 引言 1](#_Toc9336)

[1.1 背景 1](#_Toc8881)

[1.2 选题的目的与意义 1](#_Toc26230)

[1.3 主要问题 2](#_Toc2765)

[第2章 需求分析 3](#_Toc3588)

[2.1功能需求分析 3](#_Toc13847)

[2.2 可行性分析 4](#_Toc24937)

[2.1.1技术可行性 4](#_Toc2750)

[2.1.2经济可行性 5](#_Toc2835)

[2.1.3操作可行性 5](#_Toc23179)

[2.3 UML用例图 5](#_Toc27243)

[第3章 系统总体设计 8](#_Toc12515)

[3.1 功能模块设计 8](#_Toc26720)

[3.2程序流程图 10](#_Toc6574)

[3.3 数据库设计 11](#_Toc21796)

[第4章 详细设计与实现 19](#_Toc28823)

[4.1用户信息模块 19](#_Toc16621)

[4.2商品信息模块 20](#_Toc842)

[4.3购物车信息模块 21](#_Toc4035)

[4.4订单信息模块 22](#_Toc26262)

[4.5分享信息模块 23](#_Toc18584)

[4.6用户管理模块 24](#_Toc7574)

[4.7商品管理模块 26](#_Toc25146)

[4.8订单管理模块 27](#_Toc27487)

[4.9分享管理模块 28](#_Toc9455)

[第5章 测试 30](#_Toc10934)

[5.1测试环境 30](#_Toc11691)

[5.2黑盒测试 30](#_Toc9437)

[5.3验收测试 31](#_Toc18501)

[结束语 32](#_Toc26498)

[致谢 33](#_Toc16413)

[附录 源代码 35](#_Toc23367)

第1章 引言

1.1 背景

随着计算机技术在各个行业的发展与深入，基于Internet的信息化经营和管理以前所未有的速度迅猛发展，不仅改变着传统的社会生产方式，而且对经济结构的调整产生了极为深刻的影响，并成为世界经济新的增长点[1]。在计算机行业尚未发达时，商家主要以线下经营的模式来销售商品，推广、宣传也只能通过报纸、电视或者口头宣传，这种营销模式不仅需要更大的成本，也需要花费更多的时间。而如今，网络成为了商家展示自己的另一个途径，商家建立属于自己的电商平台，将商品信息、宣传信息与服务尽显与网络中，这不仅能为商家节约成本，也更好的帮助其管理。而对于顾客而言，随时随地只要有网络的存在就能上网选购自己心仪的商品，不需要再像传统模式亲临线下实体店，这也大大提高了商家与顾客之间的交易效益。

随着网络的发展同时兴起的还有移动电子商务，它更好地扩大了我们的电子商务行业的运行空间，实现了更广泛的电子商务。[2]而微信公众号正是最能体现移动电子商务的最好平台之一，它的大量兴起更使得基于移动端的电商又一次成为主流，基于微信的传播速度，及其简便等优点，更能为商家提供有效的宣传方式和更有利的商品推广媒介。在微信，有好玩的有趣的活动或者商品，大家都能通过转发或者分享到朋友圈，这也促使微信商城中的信息可以通过这种方式传递出去，相当于每个顾客都是一个免费的广告推销商，所有人都免费为您宣传。

一汽大众与切客的到店签到有礼，让我们看到了如何将顾客通过移动端推送到线下，木有美术馆通过微信传递每一束花的故事，让我们知道了其实产品是可以和顾客直接交流的。招商银行的微信爱心漂流瓶，启发我们可以通过移动互联网获取新的客户关系链；这些优秀案例给了其他企业很多启发，特别是对微信这一工具的运用让人眼前一亮，而且随着微信人群基数的扩大，大量企业开始尝试将微信作为营销工具。 [3]

1.2 选题的目的与意义

传统销售不仅需要考虑到雇员、实体店铺租金等成本问题，还要考虑到销售时间的选择，而基于互联网的销售能够大大减少人力、物力等成本，而且对于时间和空间的局限性也小了很多，用户可有网络的环境下随时随地进行网上购物。而且在电子商务模式下，人员管理更加的扁平化。因为信息传递的层次的所见，信息泄漏也比较少，这样可以很好的保证信息传递过程中的真是性和全面性，也可以对市场做出迅速的反应。[4]而且电子商务网站可以更方便的访问更广泛的商品，并且在线渠道能够为消费者带来最优惠的价格。[5]所以本系统的实现不仅能方便顾客的使用，也能为商家带来便利有效管理体系。

1.3 主要问题

安全性问题：包括数据库的安全性和电子支付的安全性。尤其是电子支付的安全性是电子商城系统中的重点，牵涉到的这些知识都需要深入的去了解。

用户反馈问题：如何了解用户对商品的评价及如何反馈信息给用户也是网站设计的一个重点，可能需要添加客户评论的模块，客户可对已购买的商品进行评论，真正实现电子商务平台的交互性。

服务器荷载问题：由于个人的能力、财力有限，项目部署的服务器是小型服务器，若同时间有较多用户同时访问本系统，可能会导致服务器荷载过重而奔溃，这会大大的影响用户的体验也会给本系统带来负面影响。

1. 需求分析

2.1功能需求分析

根据使用本系统内的普通用户和系统管理员的功能需求进行分析，系统将包括以下几个功能模块。

针对顾客的功能模块：

（1）用户信息模块

该模块主要包括用户注册、登录、查看和修改个人信息的功能。用户首次进入本系统需要进行注册。已有账号密码的用户可直接登录本系统，同时也可查看和修改个人信息。

（2）商品信息模块

该模块主要包括查看、搜索和购买商品的功能。用户进入系统后，可浏览系统中展示的商品，也可输入与商品相关的关键字进行商品的检索，当然，用户若有心仪的商品也可进行购买。

（3）购物车信息模块

该模块主要包括查看、添加和删除购物车信息的功能。用户可选择一个或多个商品，先将其添加到购物车中以便后续的购买。也可对购物车中的信息进行查看，若有不满意的商品，可以选择将其从购物车中移除。

（4）订单信息模块

该模块主要包括查看、添加和删除订单信息的功能。用户选择购买商品后，系统会自动生成一条订单信息，其中包括用户信息、收货信息以及商品信息。用户在订单页面可以查看、删除订单信息。

（5）分享信息模块

该模块主要包括分享商品信息的功能。用户可以将商品信息通过微信自带的分享功能以链接的形式分享到微信朋友圈，其他人可点击该分享链接进入到本系统，若用户第一次进入本系统并进行注册，则分享人可获得一定的返利。若用户购买分享的商品，则分享人也可获得一定的返利。

针对管理员的功能模块：

（1）管理信息模块

该模块主要包括管理员登录、查看和修改个人信息的功能。管理员登录后台管理系统后可对自己的个人信息进行查看和修改。

（2）用户管理模块

该模块主要包括查看、添加、删除用户信息的功能。管理员可查看所有注册用户的基本信息，包括用户名、密码、手机号码、注册时间和最近一次登录时间。也可添加新的用户和删除已存在的用户。并且还可导出所有用户的基本信息并生成excel表格。

（3）商品管理模块

该模块主要包括查看、添加、删除、修改商品信息的功能。管理员可查看所有的商品信息，包括商品名称、描述、价格等。也可添加新的商品、修改原有商品信息、和删除商品。并且还可导出所有商品的基本信息并生成excel表格。

（4）订单管理模块

该模块主要包括查看、修改订单状态的功能。管理员可查看所有订单信息，包括订单编号、状态、收货信息、订单详情等。也可修改订单状态。并且还可导出所有订单信息并生成excel表格。

（5）分享管理模块

包括查看所有分享信息和某个分享信息下的所有返利信息。还可以设置注册返利点和购买返利比。

2.2 可行性分析

2.1.1技术可行性

系统主要采用MVC架构进行设计，遵循低耦合，高聚合的模块划分原则，把具有复杂功能的系统通过设计分解为具有基本独立，功能简单，易于实现等特点[6]。

而且本系统秉持着前后端分离的设计理念，前端采用vue框架，其利用基于MVVM模式的轻量级响应式框架Vue.js开发,可以有效简化Web前端开发流程,降低开发难度,提高开发效率,实现了Web系统前、后端开发完全分离[7]。当然在网站实际设计中，html与css也是必不可少的工具,html通过标记符号来标记要显示的网页中的各个部分,而css能够对网页中元素位置的排版进行像素级精确控制[8]。后端采用java，其作为一个成熟的、稳定的编程语言，具有卓越的高效性、通用可移植性和逻辑安全性使它在PC、智能手机、互联网、游戏控制台等方面被广泛的使用[9]，而且Java程序可以只编译一次，就可以在各种系统中运行，这也与本系统的要求十分符合。数据库采用mysql数据库，由于其体积小、速度快、总体拥有成本低，尤其是[开放源码](http://baike.baidu.com/view/394804.htm" \t "http://baike.baidu.com/_blank)这一特点[10]也是我将其作为本系统数据库的原因。

2.1.2经济可行性

开发环境与开发工具的选择直接影响到系统制作成本的高低，所以我选择window操作系统作为开发环境，idea作为开发工具，tomcat作为服务器支持。由于上述软件都是免费的，这也大大降低了本系统开发的成本，为系统后续的盈利打下了基础。

2.1.3操作可行性

本系统的页面简约美观、易操作，用户不需要话费太多的时间来熟悉系统，各个功能的流程都一目了然，不拖泥带水，即便是第一次接触电商平台的用户也能轻易上手。

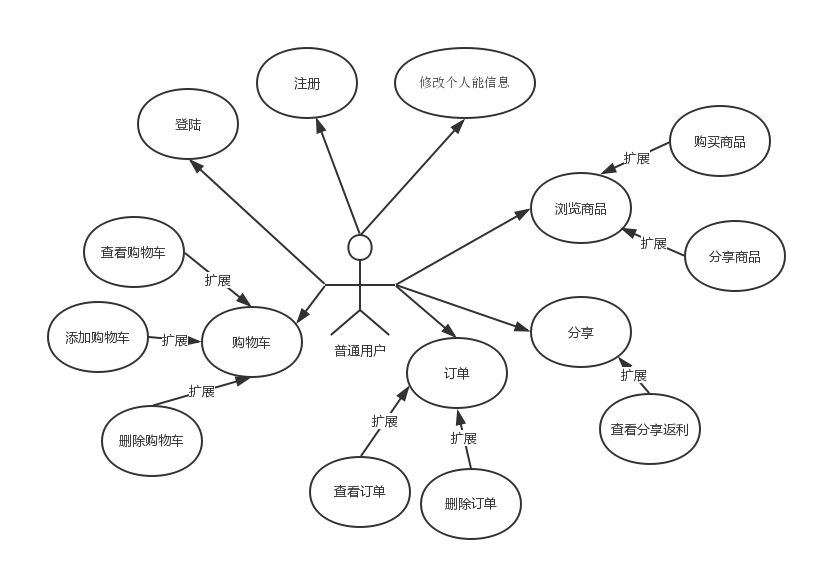
2.3 UML用例图

本系统主要针对的用户有两个，分别为普通用户和系统管理员。

普通用户即为消费者，其主要浏览本系统的前台页面，主要功能包括：

（1）用户信息模块，即用户注册、登录、查看和修改个人信息。（2）商品信息模块，即查看、搜索、购买商品。（3）购物车信息模块，即添加购物车、删除购物车、搜索购物车信息。（4）订单信息模块，即添加订单、删除订单、查看订单信息。（5）分享返利模块，即用户通过微信分享商品，他人通过分享连接进入本商城系统，注册成功则返利，购买分享商品也能返利。

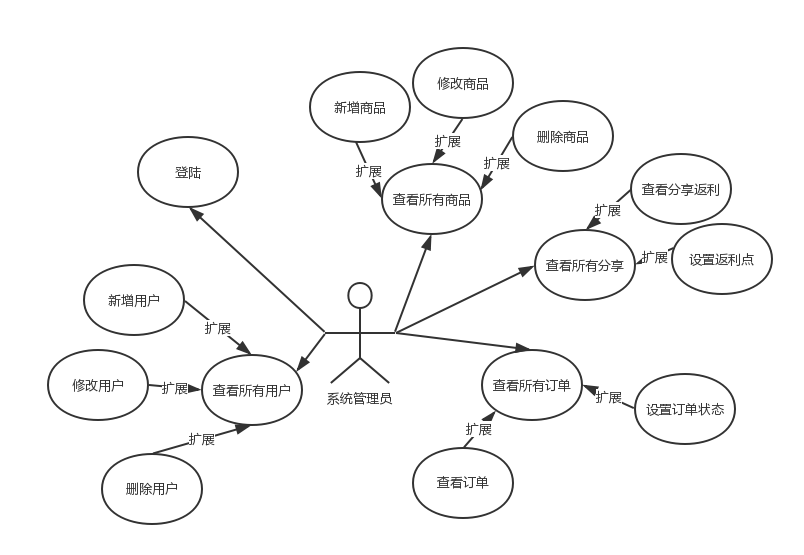
普通用户的用例图如图2-1所示。



**图2-1普通用户用例图**

系统管理员，其主要浏览本系统的后台管理页面，主要功能包括：（1）管理信息模块，即管理员登录、查看个人信息、修改个人信息。（2）用户管理模块，即查看所有用户信息、搜索用户信息、删除用户。（3）商品管理模块，即添加商品、查询商品、删除商品。（4）订单管理模块，即查看所有订单信息、搜索订单信息、导入订单和删除订单。（5）分享返利管理模块，即针对用户分享信息进行相应的积分返利。

系统管理员的用例图如图2-2所示。



**图2-2系统管理员用例图**

第3章 系统总体设计

3.1 功能模块设计

针对顾客的功能模块：

（1）用户信息模块

该模块主要包括用户注册、登录、查看和修改个人信息的功能。用户首次进入本系统需要进行注册。已有账号密码的用户可直接登录本系统，同时也可查看和修改个人信息。

（2）商品信息模块

该模块主要包括查看、搜索和购买商品的功能。用户进入系统后，可浏览系统中展示的商品，也可输入与商品相关的关键字进行商品的检索，当然，用户若有心仪的商品也可进行购买。

（3）购物车信息模块

该模块主要包括查看、添加和删除购物车信息的功能。用户可选择一个或多个商品，先将其添加到购物车中以便后续的购买。也可对购物车中的信息进行查看，若有不满意的商品，可以选择将其从购物车中移除。

（4）订单信息模块

该模块主要包括查看、添加和删除订单信息的功能。用户选择购买商品后，系统会自动生成一条订单信息，其中包括用户信息、收货信息以及商品信息。用户在订单页面可以查看、删除订单信息。

（5）分享信息模块

该模块主要包括分享商品信息的功能。用户可以将商品信息通过微信自带的分享功能以链接的形式分享到微信朋友圈，其他人可点击该分享链接进入到本系统，若用户第一次进入本系统并进行注册，则分享人可获得一定的返利。若用户购买分享的商品，则分享人也可获得一定的返利。

针对管理员的功能模块：

（1）管理信息模块

该模块主要包括管理员登录、查看和修改个人信息的功能。管理员登录后台管理系统后可对自己的个人信息进行查看和修改。

（2）用户管理模块

该模块主要包括查看、添加、删除用户信息的功能。管理员可查看所有注册用户的基本信息，包括用户名、密码、手机号码、注册时间和最近一次登录时间。也可添加新的用户和删除已存在的用户。并且还可导出所有用户的基本信息并生成excel表格。

（3）商品管理模块

该模块主要包括查看、添加、删除、修改商品信息的功能。管理员可查看所有的商品信息，包括商品名称、描述、价格等。也可添加新的商品、修改原有商品信息、和删除商品。并且还可导出所有商品的基本信息并生成excel表格。

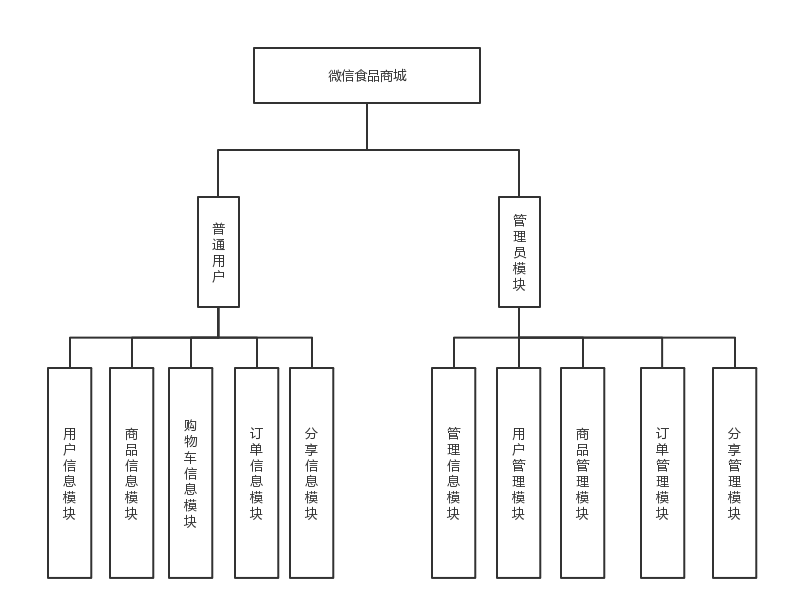
（4）订单管理模块

该模块主要包括查看、修改订单状态的功能。管理员可查看所有订单信息，包括订单编号、状态、收货信息、订单详情等。也可修改订单状态。并且还可导出所有订单信息并生成excel表格。

（5）分享管理模块

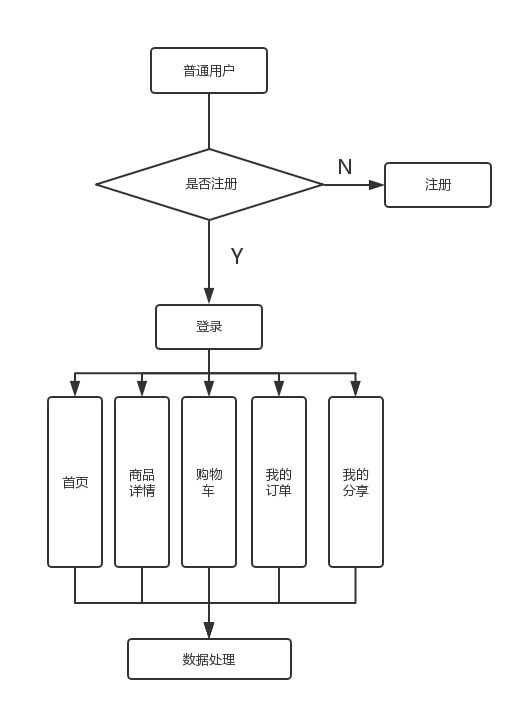
该模块主要包括查看分享信息、返利点的设置的功能。管理员可查看所有分享信息，也可添加修改返利点。并且还可导出所有分享信息并生成excel表格。

整体系统的功能模块示意图，如图3-1所示。

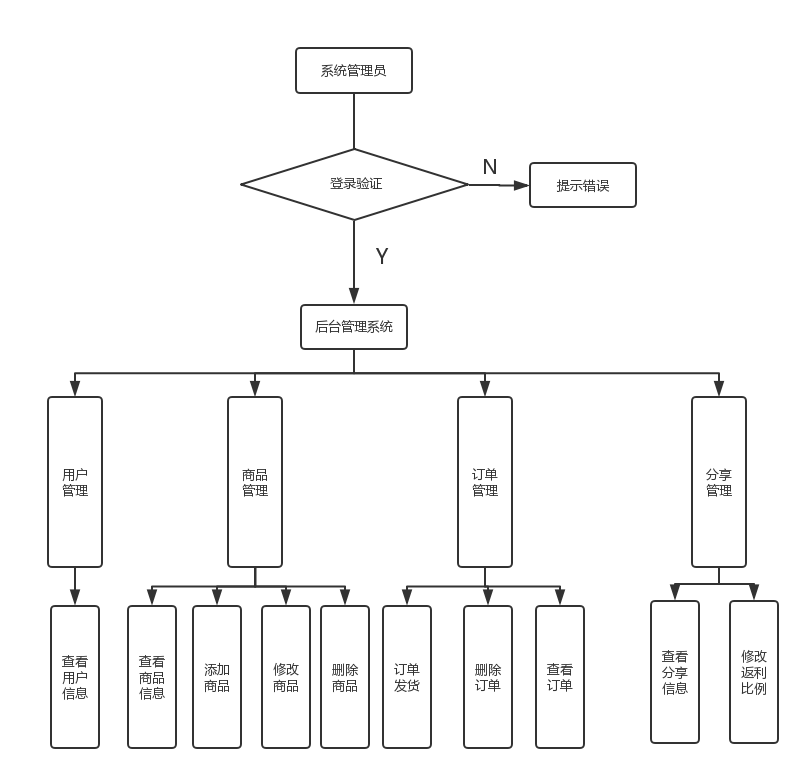
**图3-1系统功能模块图**

3.2程序流程图

（1）普通用户打开微信商城前台页面后，程序流程图如图3-2所示。



**图3-2普通用户流程图**

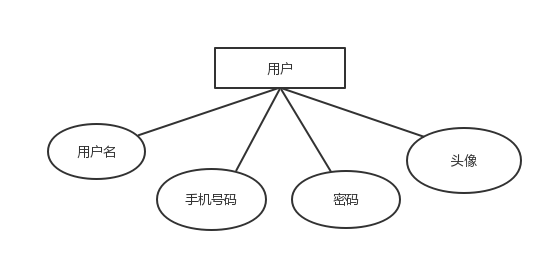
（2）系统管理员打开微信商城后台管理页面后，程序流程图如图3-3所示。

**图3-3系统管理员流程图**

3.3 数据库设计

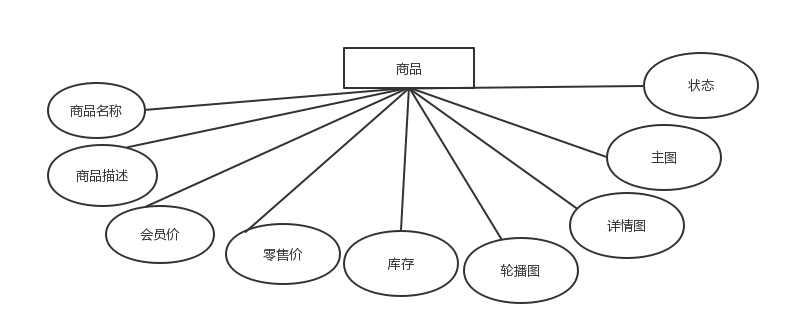
（1）数据库模型图设计。

用户表实体包括的信息主要有：用户名、手机号码、密码、头像。用户实体图如图3-4所示。

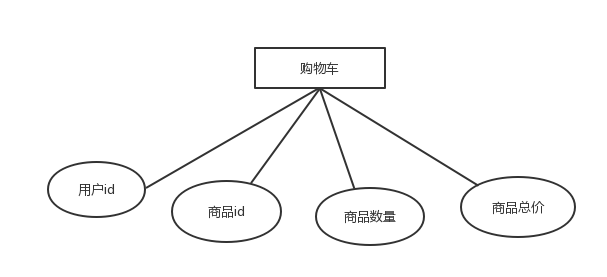


**图3-4用户实体图**

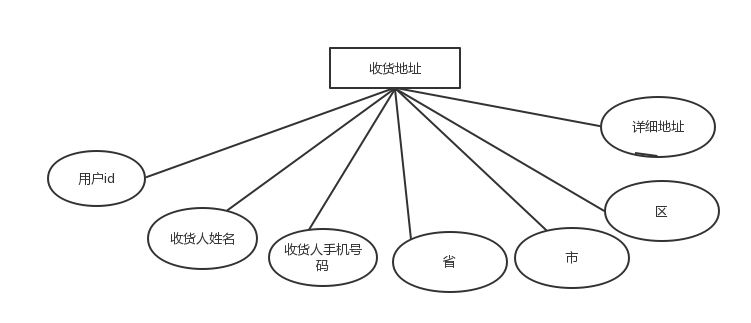
商品实体包括的信息有：商品名称、商品描述、会员价、零售价、库存、状态、主图、轮播图、详情图。商品实体图如图3-5所示。

**图3-5商品实体图**

购物车实体包括的信息有：用户id、商品id、商品数量、商品总价。购物车实体图如图3-6所示。

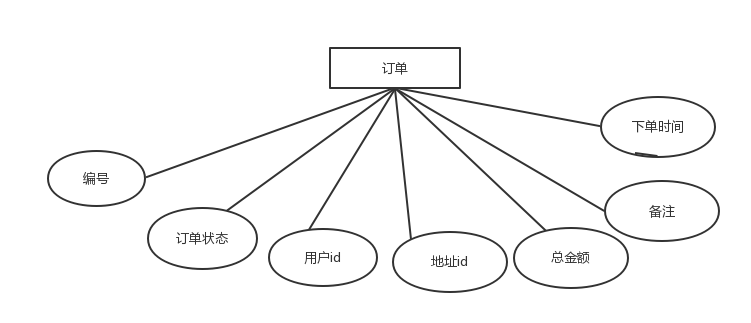
**图3-6购物车实体图**

收货地址实体包括的信息有：用户id、收货人姓名、收货人手机号码、省、市、区、详细地址。收货地址实体图如图3-7所示。

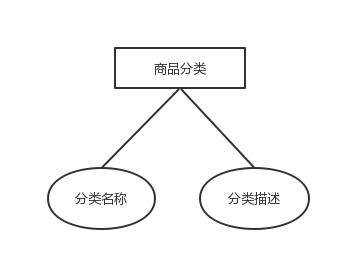


**图3-7收货地址实体图**

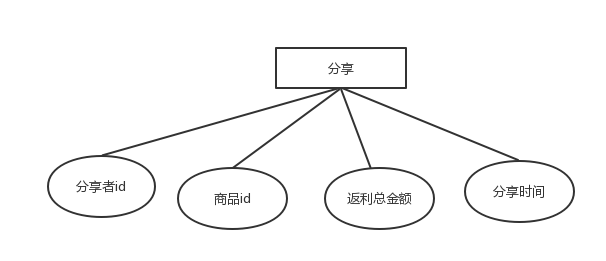
订单实体包括的信息有：订单编号、订单状态、用户id、地址id、总金额、备注、下单时间，订单实体图如图3-8所示。



**图3-8订单实体图**

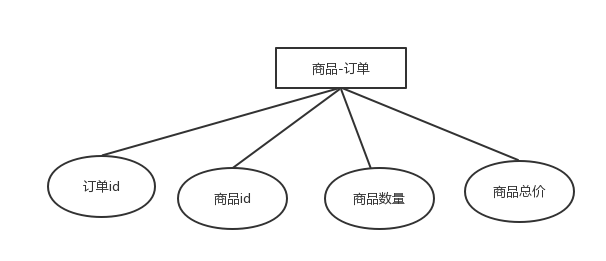
商品分类实体包括的信息有：分类名称、分类描述，商品分类实体图如图3-9所示。

**图3-9商品分类实体图**

分享实体包括的信息有：分享者id、商品id、、返利总额、分享时间，分享实体图如图3-10所示。

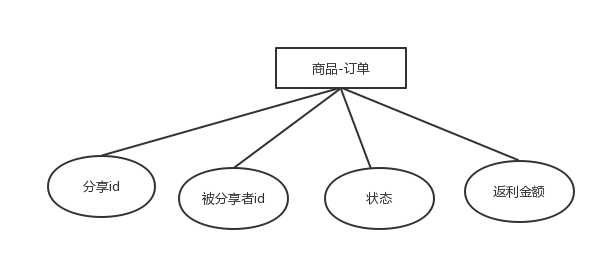
**图3-10分享实体图**

商品-订单关联实体包括的信息有：订单id、商品id、商品数量、商品总价，商品订单关联实体图如图3-11所示。



**图3-11商品-订单实体图**

用户-分享关联实体包括的信息有：分享id、被分享者id、状态、返利金额，用户-分享关联实体图如图3-12所示。

**图3-12用户-分享实体图**

（2）数据库表设计

用户信息表如表3-1所示。

**表3-1用户信息表**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 注释 |
| id | int | 用户id |
| phone | varchar | 手机号码 |
| password | varchar | 密码 |
| regist\_time | datetime | 注册时间 |
| recent\_login\_time | datetime | 最近登录时间 |

商品信息表如表3-2所示。

**表3-2商品信息表**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 注释 |
| id | int | 商品id |
| name | varchar | 商品名称 |
| vip\_price | float | 会员价 |
| retail\_price | float | 零售价 |
| subtitle | varchar | 商品描述 |
| stock | varchar | 库存 |
| main\_image | varchar | 主图 |
| carousel\_image | varchar | 轮播图 |
| detail\_image | varchar | 详情图 |
| type\_id | int | 分类id |

收货地址表如表3-3所示。

**表3-3收货地址表**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 注释 |
| id | int | 地址id |
| contacts | varchar | 联系人 |
| phone | float | 手机号码 |
| province | float | 省 |
| province\_city | varchar | 市 |
| area | varchar | 区 |
| detail\_address | varchar | 详细地址 |
| user\_id | varchar | 用户id |

商品分类表如表3-4所示。

**表3-4商品分类表**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 注释 |
| id | int | 分类id |
| type\_name | varchar | 分类名称 |
| type\_describe | varchar | 分类描述 |

购物车表如表3-5所示。

**表3-5购物车表**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 注释 |
| id | int | 地址id |
| user\_id | int | 用户id |
| goods\_id | int | 商品id |
| goods\_number | int | 商品数量 |

分享表如表3-6所示。

**表3-6分享表**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 注释 |
| id | int | 分享id |
| share\_user\_id | int | 分享者id |
| goods\_id | int | 商品id |
| share\_time | datetime | 分享时间 |
| amounts | float | 返利金额 |

订单表如表3-7所示。

**表3-7订单表**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 注释 |
| id | int | 订单id |
| number | varchar | 订单编号 |
| status | varchar | 订单状态 |
| user\_id | int | 用户id |
| address\_id | int | 地址id |
| total\_amount | float | 总金额 |
| remarks | varchar | 备注 |
| order\_time | datetime | 下单时间 |

商品-订单表如表3-8所示。

**表3-8商品-订单表**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 注释 |
| id | int | id |
| goods\_id | int | 商品id |
| order\_id | int | 订单id |
| goods\_number | int | 商品数量 |
| goods\_price | float | 商品总价 |

用户-分享表如表3-9所示。

**表3-9用户-分享表**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 注释 |
| id | int | id |
| share\_id | int | 分享id |
| user\_id | int | 被分享者id |
| status | int | 状态 |
| amounts | float | 返利金额 |

第4章 详细设计与实现

本章节主要介绍网上童装销售系统的详细设计与实现过程，根据系统总体设计中的描述，微信食品商城系统的功能主要划分普通用户的5个模块和系统管理员的5个模块，本章节将从微信食品商城系统功能的实现思路、系统技术应用过程、实现步骤和程序实现结果4个方面进行阐述。

4.1用户信息模块

该模块主要包括用户注册、登录、查看和修改个人信息的功能。用户首次进入本系统需要进行注册。已有账号密码的用户可直接登录本系统，同时也可查看和修改个人信息。

用户通过微信公众号进入微信食品商城系统，首先进入的是登录注册页面，首次进入系统的用户可以选择注册，注册需要填写手机号码、密码，点击注册按钮，前端通过axios调取后端regist接口，后端获取用户填写的手机号码和密码，查询用户表中是否已存在当前手机号码，若存在，则提示用户当前手机号码已存在，反之则注册成功，自动进入系统首页。老用户通过输入手机号码和密码直接登录系统，同理，通过调用login接口，在数据库用户表中比对手机号码和密码，若有匹配值，则跳转进入系统首页，若匹配失败，则提示用户登录名或密码有误。

用户成功进入系统后，点击底部导航条中的“我的”，跳转到我的页面，并选择个人资料，跳转进入个人资料页面，在该页面中用户可修改自己的密码。登录页面如图4-1所示，注册页面如图4-2所示，个人资料页面如图4-3所示。



**图4-1 登录页面**  **图4-2 注册页面**  **图4-3 个人资料页面**

4.2商品信息模块

该模块主要包括查看、搜索和购买商品的功能。用户进入系统后，可浏览系统中展示的商品，也可输入与商品相关的关键字进行商品的检索，当然，用户若有心仪的商品也可进行购买。

用户进入系统后，系统首先展示给用户的为首页，首页布局从上到下分别为，搜索条、商品分类、轮播图和推荐商品。用户通过点击搜索，进入搜索页面，在搜索条中输入商品信息，点击收缩按钮后，前端通过axios调用后端searchGoods接口，将搜索信息与商品表中的商品名称做模糊比较，若匹配到相应的商品，则将商品的信息，包括id、商品名称、商品主图、会员价、零售价、库存、描述，封装成json格式的数组，返回给前端，前端将这些信息展示到搜索页面中。

用户点击商品分类，前端调用selectType接口将用户选中的分类id传到后端，后端通获取商品表中商品分类id等于选中的分类id，并将这些匹配的商品信息，包id、商品名称、商品主图、会员价、零售价、库存、描述，封装成json格式的数组，返回给前端，前端将这些信息展示在首页。

用户点击商品，可进入到相应商品的详情页面，在详情页面，用户可以查看该商品的所有信息包括轮播图、名称、描述、会员价、零售价、详情图。用户可以选择加入购物车和立即购买，若选择加入购物车，则系统调用addShoppingCar接口，将商品id、用户id、商品数量存入到购物车表中生成一条新的记录。若选择立即购买，系统自动跳转到确认订单页面，在该页面中，用户可以选择收货信息，查看订单详情。若继续选择提交订单，则前端调取addOrder接口在数据库order表和goods-order表中生成相应的记录。首页页面如图4-4所示，商品详情页面如图4-5所示，确认订单页面如图4-6所示。



**图4-4 首页页面**  **图4-5 商品详情页面**  **图4-6 确认订单页面**

4.3购物车信息模块

该模块主要包括查看、添加和删除购物车信息的功能。用户可选择一个或多个商品，先将其添加到购物车中以便后续的购买。也可对购物车中的信息进行查看，若有不满意的商品，可以选择将其从购物车中移除。

用户可以在商品详情页面中点击“加入购物车”、选择商品数量并确认，前端调取addShoppingCar接口，将用户id、商品id、商品数量传到后端，后端将这些信息添加到购物车表中形成一条新的记录。

用户可以点击系统底部导航条中的“购物车”，跳转到购物车页面，前端将通过axios调取selectShoppingCar接口将用户id传到后端，后端通过查询购物车表中的user\_id字段和前端传过来的id进行匹配，将匹配到的购物车信息，包括商品id、商品主图、商品名称、商品单价、商品数量封装成json格式的数组返回给前端，前端再做数据展示给用户。

用户可在购物车页面修改添加到购物车的商品的数量，点击加号或者减号，，前端调用editShoppingCar接口将购物车id和商品数量传给后端，后端通过查询购物车表中id与前端传来的购物车id相匹配的信息，将此记录的中的商品数量修改为前端传来的商品数量。

用户在购物车页面点击右上角的删除图标，进行购物车信息的删除，先选中需要删除的购物车信息，点击删除按钮，前端通过调用deleteShoppingCar接口将购物车id传给后端，后端删除数据库表中id与其相匹配的记录。购物车页面如图4-7所示，删除购物车商品页面如图4-8所示。



**图4-7 购物车页面**  **图4-8 删除购物车商品页面**

4.4订单信息模块

该模块主要包括查看、添加和删除订单信息的功能。用户选择购买商品后，系统会自动生成一条订单信息，其中包括用户信息、收货信息以及商品信息。用户在订单页面可以查看、删除订单信息。

用户在商品详情页面点击立即购买并却提交订单或者在购物车页面选中商品信息点击结算并确认订单，前端通过调用addOrder接口，将用户id、地址id、订单总金额、备注、商品id、商品数量、商品单价传给后端，后端在订单表和商品-订单关联表中插入相应的信息生成一条新的记录。

用户通过点击导航条中的“我的”进入，我的页面，可选择点击全部订单、代付款、待发货、待收货、已完成进入相应的订单页面，前端通过调用selectOrder接口将用户id和订单状态传给后端。后端通过查询order表将匹配的数据封装成json格式的数组返回给前端，前端将订单数据显示在页面。

用户在订单页面点击订单可以跳转进入订单详情页面，订单详情页面显示订单的详细信息，当订单状态为未支付时，用户可以选择支付，支付完成后订单状态即变为代发货，当系统管理员确认发货后，订单状态即变为代收货，用户可以点击确认收获，此次订单流程即完成，订单状态也会变为已完成，用户可以选择删除该订单。我的页面如图4-9所示，订单页面如图4-10所示，订单详情页面如图4-11所示。

**图4-9 我的页面**  **图4-10 订单页面 图4-11 订单详情页面**



4.5分享信息模块

该模块主要包括分享商品信息的功能。用户可以将商品信息通过微信自带的分享功能以链接的形式分享到微信朋友圈，其他人可点击该分享链接进入到本系统，若用户第一次进入本系统并进行注册，则分享人可获得一定的返利。若用户购买分享的商品，则分享人也可获得一定的返利。

用户在浏览商品时，进入商品详情页面选择微信自带的分享功能，将某个商品详情页面以链接的形式分享到朋友圈或朋友，前端调用addShare接口将用户id、商品id传到后端，后端将此信息插入到分享表中，他人通过点击分享连接进入到本系统，若此人为新用户，并注册了帐号，则前端调用regist的接口同时调用clickShare接口，将分享id和用户id传给后端，后端将此信息插入到user\_share关联表中，同时将表中的status设置为注册，并将返利金额插入到表中的amounts中，同时更新分享表中将amounts返利金额，当点击链接的用户注册完成后，系统自动跳转到被分享商品的详情页面，点击购买该商品后，前端会调用clickShare接口，在数据库中user\_share插入一条新纪录，并将status设为购买，同时更新分享表中的amounts值。

用户点击底部导航条的“我的”进入我的页面，再次点击“我的分享”跳转进入到我的分享页面，前端调用selectShare接口将用户id传到后端，后端查询share表和user\_share表，匹配的信息封装成json格式的数组返回给前端，前端将返回信息展示在我的分享页面中。我的分享页面如图4-12所示，分享详情页面如图4-13所示。



**图4-12 我的分享 图4-13 分享详情**

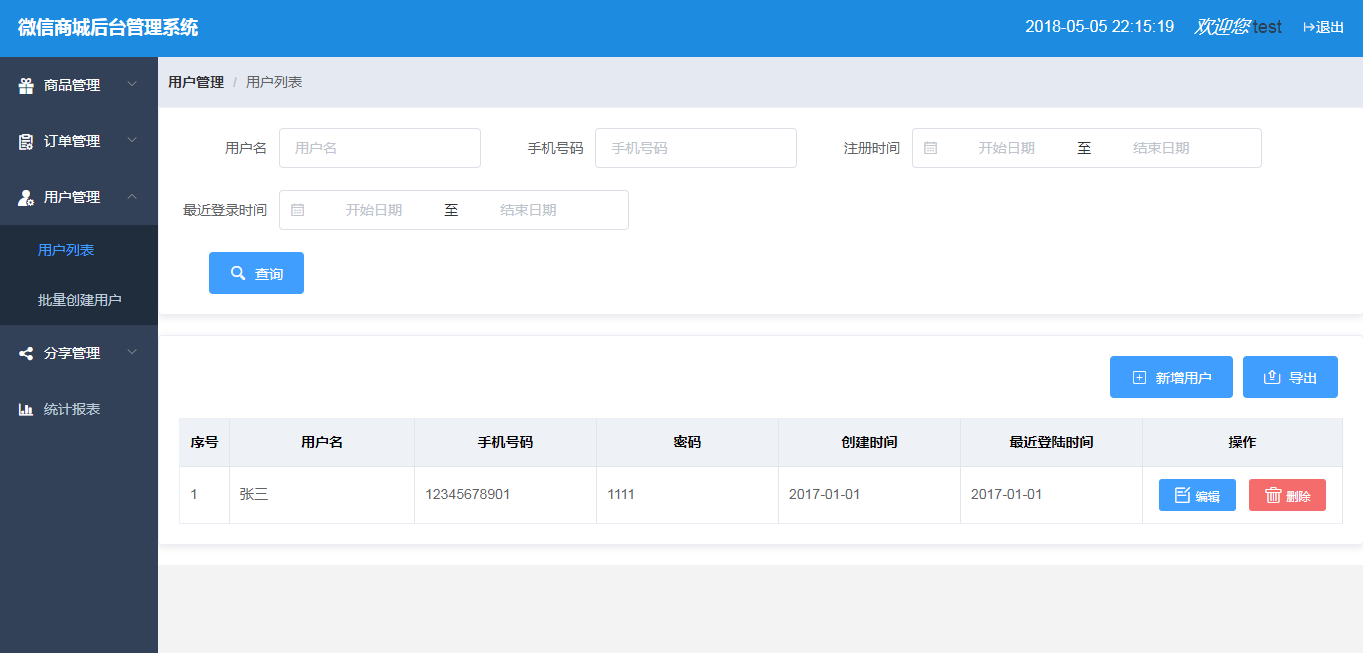
4.6用户管理模块

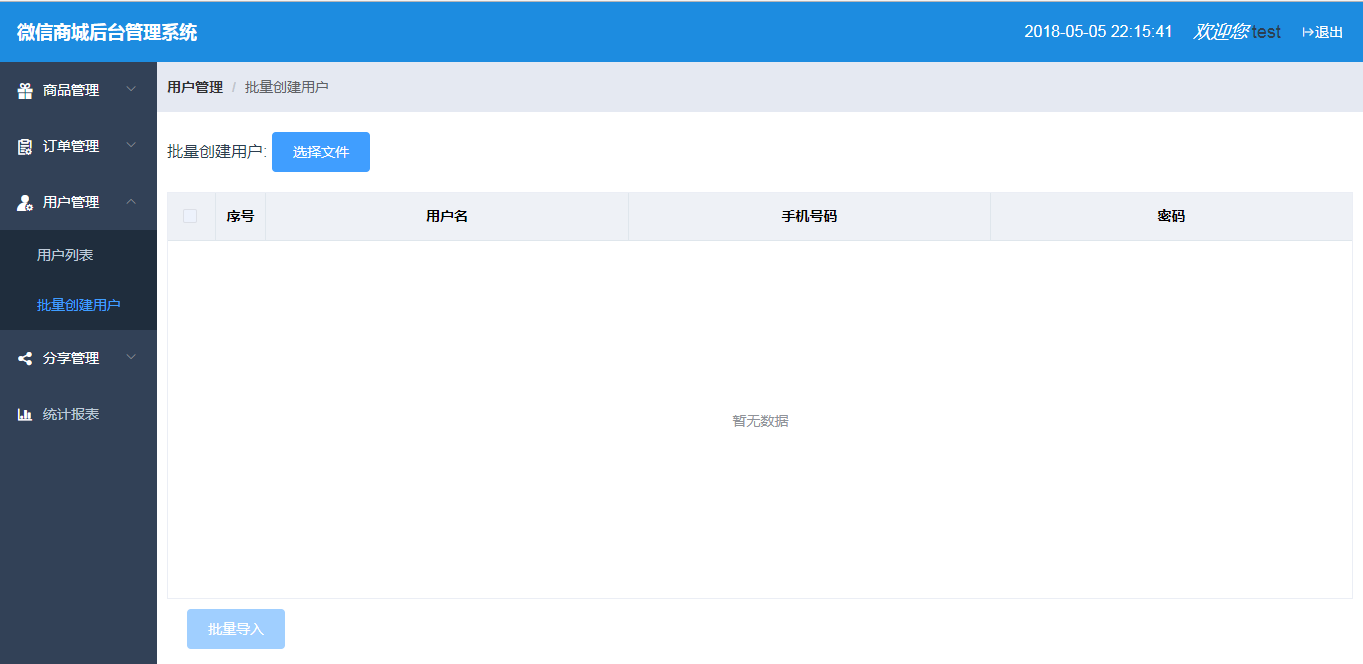
该模块主要包括查看、添加、删除用户信息的功能。管理员可查看所有注册用户的基本信息，包括用户名、密码、手机号码、注册时间和最近一次登录时间。也可添加新的用户和删除已存在的用户。并且还可导出所有用户的基本信息并生成excel表格。

系统管理员进入后台管理系统后，点击左侧导航条中的用户列表，进入用户列表页面，前端调取selectUser接口，后端将user表中所有数据封装成json格式的数组，前端将这些信息显示在页面上，用户可以通过输入用户名、手机号码、注册时间、最近登录时间来搜索用户，此时前端仍调取selectUser接口，并将搜索条件传给后端，后端在user表中查询匹配搜索条件的记录组装成json格式返回给前端，前端将这些信息展现在页面上。

管理员可点击新增用户，系统将弹出一个模态框，管理员输入用户名、手机号码、密码、确认密码，并点击添加按钮，此时前端调用addUser接口并将上述信息传给后端，后端首先在user表中查找该用户名是否已存在，若已存在，则提示用户“该用户已存在”,反之，将上述信息插入到user表中形成一个新的记录。管理员也可点击修改或删除，前端调用editUser或deleteUser接口，后端将更新user表中的信息或删除某条记录。

管理员点击左侧导航条中的批量创建用户页面，点击选择文件按钮，并选择需要导入的excel的表格，页面上会显示需要导入的数据，用户可以选择一条或者多条数据并点击批量导入，前端即调用batchAddUser接口，将批量用户的信息传给后端，后端先验证用户是否已存在，若已存在则提示管理员“某某用户已存在”，反之，则在user表中插入这些信息。用户列表页面如图4-14所示，批量创建用户页面如图4-15所示。

**图4-14 用户列表页面**

**图4-15 批量创建用户页面**

4.7商品管理模块

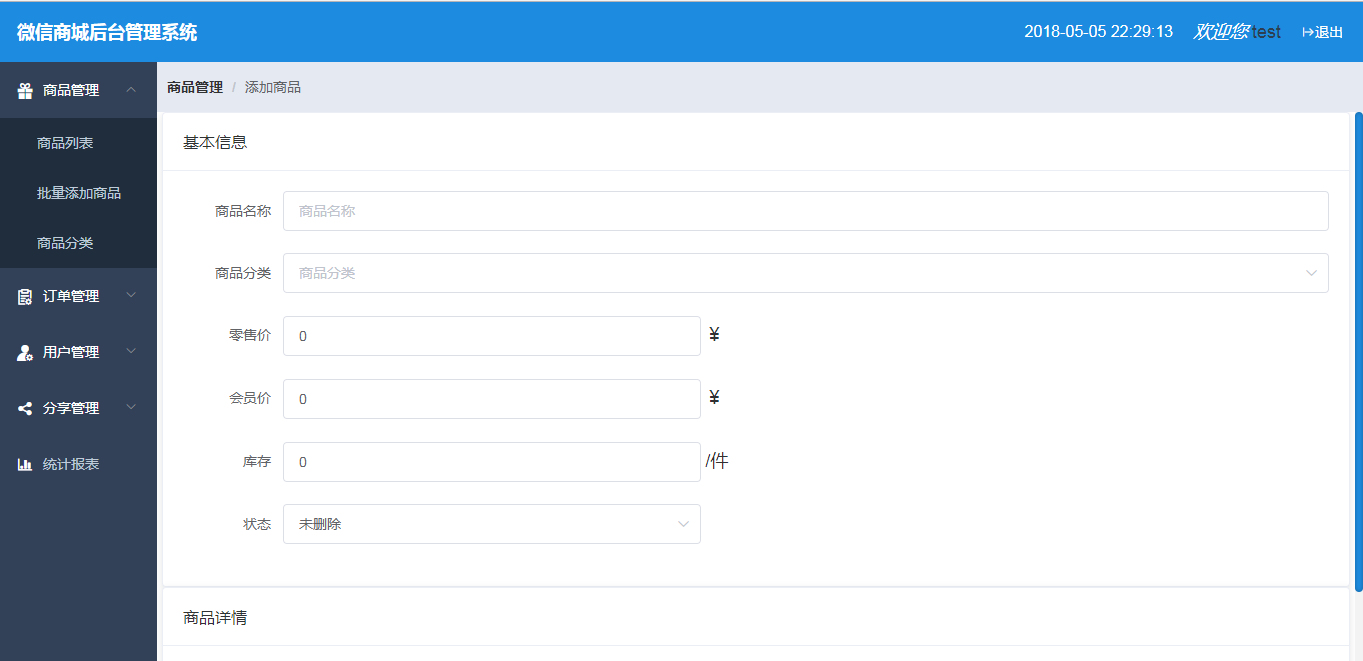
该模块主要包括查看、添加、删除、修改商品信息和管理商品分类的功能。管理员可查看所有的商品信息，包括商品名称、描述、价格等。也可添加新的商品、修改原有商品信息、和删除商品。并且还可导出所有商品的基本信息并生成excel表格。

系统管理员点击左侧导航条中的商品列表，进入商品列表页面，前端调取selectGoods接口，后端将goods表中所有数据封装成json格式的数组，前端将这些信息显示在页面上，用户可以通过输入商品名称、商品描述、商品分类、库存、状态来搜索商品，此时前端仍调取selectGoods接口，并将搜索条件传给后端，后端在goods表中查询匹配搜索条件的记录组装成json格式返回给前端，前端将这些信息展现在页面上。

管理员可点击新增商品，进入新增商品页面，管理员输入相应的信息，包括商品名称、描述、会员价、零售价等，并点击添加按钮，此时前端调用addGoods接口并将上述信息传给后端，后端上述信息插入到goods表中形成一个新的记录。管理员也可点击修改或删除，前端调用editGoods或deleteGoods接口，后端将更新goods表中的信息或删除某条记录。商品列表页面如图4-16所示，新增商品页面如图4-17所示。



**图4-16 商品列表页面**



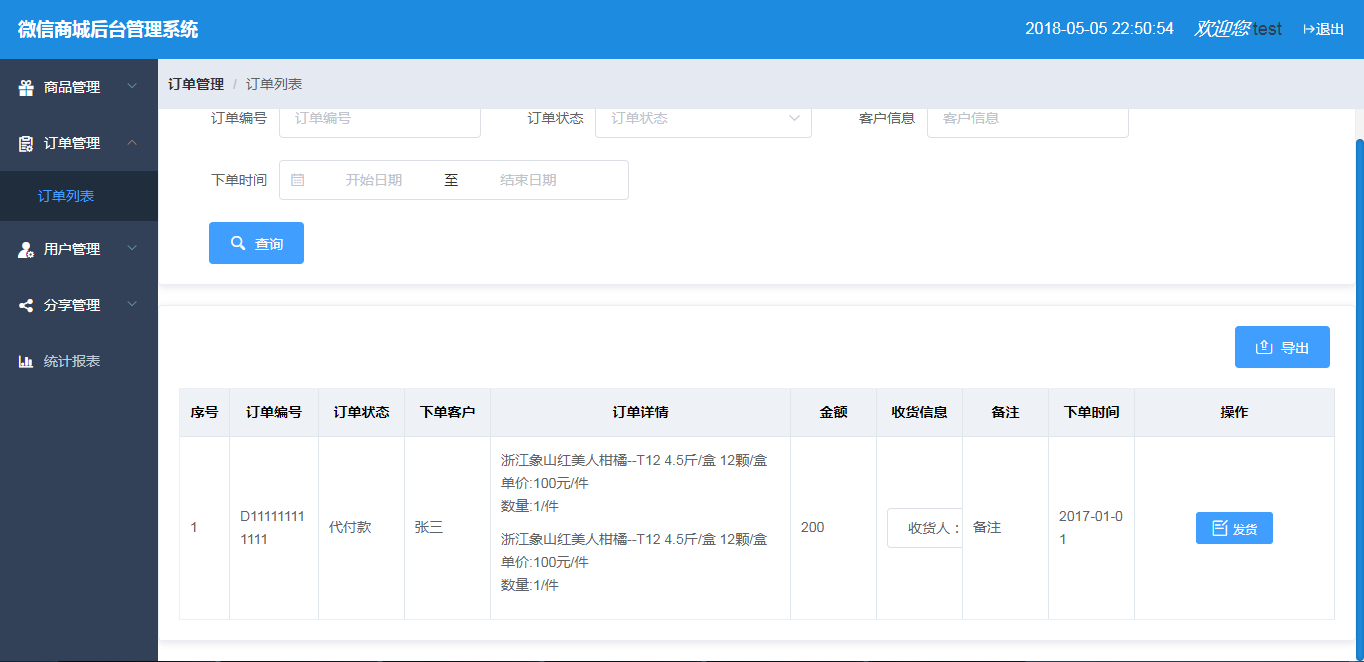
**图4-17 新增商品页面**

4.8订单管理模块

该模块主要包括查看、修改订单状态的功能。管理员可查看所有订单信息，包括订单编号、状态、收货信息、订单详情等。也可修改订单状态。并且还可导出所有订单信息并生成excel表格。

系统管理员点击左侧导航条中的订单列表，进入订单列表页面，前端调取selectOrder接口，后端将order表和goods\_order关联表中所有数据封装成json格式的数组，前端将这些信息显示在页面上，用户可以通过输入订单编号、订单状态、客户信息、下单时间搜索订单，此时前端仍调取selectOrder接口，并将搜索条件传给后端，后端在order表中查询匹配搜索条件的记录组装成json格式返回给前端，前端将这些信息展现在页面上。

管理员可点击编辑订单，可设置订单的状态，若订单状态为待发货，用户可点击发货，此时前端调用editOrder接口，将订单id传给后端，后端在order表中查到匹配记录，并将该记录的订单状态改为已发货。订单列表页面如图4-18所示。



**图4-18 订单列表页面**

4.9分享管理模块

该模块主要包括查看分享信息、返利点的设置的功能。管理员可查看所有分享信息，也可添加修改返利点。

系统管理员点击左侧导航条中的分享列表，进入分享列表页面，前端调取selectShare接口，后端将share表和user\_share关联表中所有数据封装成json格式的数组，前端将这些信息显示在页面上，用户可以通过分享用户用户名、商品名称、搜索分享信息，此时前端仍调取selectShare接口，并将搜索条件传给后端，后端在share表中查询匹配搜索条件的记录组装成json格式返回给前端，前端将这些信息展现在页面上。系统管理员也可设置注册返利点和购买商品返利点。分享列表页面如图4-19所示。

**图4-19 分享列表页面**

第5章 测试

随着软件应用领域越来越广泛，其质量的优劣也日益收到人们的重视。质量保证能力的强弱直接影响着软件业的生存和发展。软件测试是软件生命周期中一项非常重要且复杂的工作，对软件可靠性保证具有极其重要的意义。[11]

5.1测试环境

硬件环境：PC机。

软件环境：windows操作系统、mysql数据库、tomcat服务器和node服务器。

5.2黑盒测试

黑盒测试也称做功能测试和行为测试、数据驱动测试，主要是根据功能需求来测试程序是否按照预期工作。这里将以微信食品商城底部导航条和登录为例进行黑盒测试。底部导航条测试用例表如表5-1所示。

**表5-1 底部导航条测试用例**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 项目/软件 | 微信食品商城 | | | 功能模块名 | 底部导航条 | | |
| 功能特性 | 页面跳转测试 | | | | | | |
| 测试目的 | 消费者点击底部导航条的能否顺利进行页面的跳转 | | | | | | |
| 测试数据 | “首页”、“购物车”、“我的”字样 | | | | | | |
| 测试项 | 测试类别 | 描述/输入/操作 | | | | 期望结果 | 真实结果 |
| 首页页面跳转 | 功能测试 | 点击“首页”字样 | | | | 跳转到首页 | 跳转到首页 |
| 购物车页面跳转 | 功能测试 | | 点击“购物车”字样 | | | 跳转到购物车页面 | 跳转到购物车页面 |
| 我的页面跳转 | 功能测试 | | 点击“我的”字样 | | | 跳转到我的页面 | 跳转到我的页面 |

登录测试用例表如表5-2所示。

**表5-2 登录测试用例**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 项目/软件 | 微信食品商城 | | 功能模块名 | 登录模块 | | |
| 功能特性 | 登录测试 | | | | | |
| 测试目的 | 消费者能否正常登录 | | | | | |
| 测试数据 | 手机号码phone、密码password | | | | | |
| 测试项 | 测试类别 | 描述/输入/操作 | | | 期望结果 | 真实结果 |
| 用户登录 | 功能测试 | 输入手机号17826804660  输入密码123456 | | | 登录成功 | 登录成功 |

5.3验收测试

验收测试是部署软件之前的最后一个测试操作。其目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。这里将以微信食品商普通用户的登录注册、商品信息、购物车信息、订单信息、分享信息进行验收测试。功能验收情况表如表5-3所示。

**表5-3 功能验收测试情况表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 验收模块名称 | 开始时间 | 结束时间 | 用例数 | 用例通过数 | 问题数 | 用例通过率 |
| 个 | 个 | 个 | ％ |
| 登录注册 | 2018-04-30 | 2018-04-30 | 10 | 9 | 1 | 90% |
| 商品信息 | 2018-04-30 | 2018-04-30 | 10 | 8 | 2 | 80% |
| 购物车信息 | 2018-04-30 | 2018-04-30 | 10 | 7 | 3 | 70% |
| 订单信息 | 2018-04-30 | 2018-04-30 | 10 | 8 | 2 | 80% |
| 分享信息 | 2018-04-30 | 2018-04-30 | 10 | 6 | 4 | 60 |

结束语

在本次针对微信食品商城的设计与开发的过程中，我将我在上课所学到的知识及自学掌握的技能充分发挥了出来，虽不说该系统有多完美，但也能称得上是一个成熟稳定的电商系统了。这个系统是对我大学4年的肯定，也是对毕业后能否顺利找到工作的垫脚石。在整个设计与开发的过程中，我也遇到了种种困难。

例如我在本系统的前端开发过程中采用了vuejs框架，这是一个MVVM的前端框架，其开发理念完全不同于以前的比如jQuery之类的js库，它主要是一套用于构建用户界面的渐进式框架，其核心只关注视图层，以数据驱动视图，几乎不用直接操作DOM节点。由于之前一直接触都是jQuery或者是原生的js去直接操作DOM节点，动态的添加、删除、修改节点，所有刚开始上手的时候还是对vue的驱动模式非常不适应，不过后来通过上网查看其官方文档、浏览相应论坛博客也慢慢体会到了vuejs框架的魅力，大大加快了开发进度。而且对于支付，本系统还是采用的虚拟金钱及积分的形式，未采用支付宝、微信支付等功能，由于涉及到钱，我对本系统的安全性能还不是很肯定而且调取支付宝和微信支付等接口的申请需要企业身份，所以我只能积分的形式来模拟支付流程。

对于本系统的最终效果，我还是很满意的。不仅有基本的功能，我也添加了分享获利的创新点，这个功能的存在我相信能大大提升用户对于本系统的使用热情，用户通过分享可以获得相应的利润，而对于本系统而言，用户的分享等同于对宣传。这又何尝不是一个双赢的营销模式。

致谢

首先我要感谢我的指导老师，感谢他细心的教导并为我提出修改意见，在论文选题上也给予了我很大的帮助，及时帮我解答疑惑。感谢周建楠同学为我修改开题报告并指导我修改论文格式，在学习上我们共同进步，在生活上我们相互鼓励。同时也要感谢我的母校浙江科技学院和各位任课老师为我提供的良好的学习环境、让我在大学四年中学到不少专业知识和人生理念，使我在之后进入社会后能有一技之长，也让我在之后的为人处世中能够少走弯路。最后，特别要感谢的是我的家人和同学，他们四年来对我的关心和支持是我完成学业的精神动力，没有他们的支持，就没有我今天的成就。

**参考文献**

1. 李睿.移动互联营销--微信营销原则思考[J].广告人，2013
2. Huili Wang. On Development and Application Prospect of Mobile E-commerce[A]. Proceedings of 2016 2nd International Conference on Electronics,Network and Computer Engineering(ICENCE 2016)
3. 刘向明，柯晓雯.网上商城系统——一种B2C电子商务模式[J].广东通信技术，2001
4. 于奎,冯学勇等. 电子商务对传统商业的影响与挑战[J].河南社会科学2000(6):47-48
5. Zhu Zijun. E-commerce: A Young,Fast-Growing Industry in China[J].China's Foreign Trade,2012(01):52-55.
6. 毛娇,胡晓旭,荆晓菊等.网站产品开发项目可行性研究[J]. 现代营销(下旬刊). 2015(02):146-147
7. 朱二华.基于vue.js的Web前端应用研究[j].科技与创新，2017
8. 陆凌牛.HTML 5与CSS 3权威指南[M]. 机械工业出版社, 2015
9. 王明鑫，于立婷，孙仲霖。 简述Java编程语言的发展及优势[J]. 通讯世界，2016,01:204.
10. 唐汉明,翟振兴,关宝军,王洪全,黄潇. 深入浅出MySQL-数据库开发、优化与管理维护[M]. 人民邮电出版社,2014
11. 李海生，郭跃.软件测试技术案例教程[M]. 清华大学出版社, 2012

**附录 源代码**

微信商城首页vue单文件组件

<template>

<div class="home\_wrap">

<header>

<div class="search\_wrap">

<div class="search\_bar" @click="goToSearch">

<span>搜索</span>

</div>

</div>

<div class="product\_type">

<ul>

<li :class="{'type-active':typeId === 0}" @click="selectType(0)">推荐</li>

<li v-for="(item, index) in typeList" @click="selectType(item.id)" :class="{'type-active':typeId === item.id}">

{{item.typeName}}

</li>

</ul>

</div>

</header>

<div class="product\_wrap">

<div class="swipe\_wrap" v-if="typeId === 0">

<mt-swipe :auto="4000">

<mt-swipe-item><img src="../assets/images/lunbo.jpg"></mt-swipe-item>

<mt-swipe-item><img src="../assets/images/lunbo.jpg"></mt-swipe-item>

<mt-swipe-item><img src="../assets/images/lunbo.jpg"></mt-swipe-item>

</mt-swipe>

</div>

<div class="product\_list">

<div class="product" v-for="(item, index) in productList" :key="index" @click="goToDetail(item.id)">

<div class="product\_img">

<img :src="item.mainImage">

</div>

<div class="product\_info">

<div class="product\_name">

{{item.name}}

</div>

<div class="product\_price">

<div class="vip\_price">

<span>￥</span>

<!-- <span>会员价￥</span> -->

<span>{{item.vipPrice|formatMoney}}</span>

</div>

<div class="normal\_price">

<div class="line"></div>

<span>零售价￥</span>

<span>{{item.retailPrice|formatMoney}}</span>

</div>

</div>

</div>

</div>

</div>

</div>

<tab-bar></tab-bar>

</div>

</template>

<script>

import { SelectAllType, SelectCommodityByTypeId } from '@/js/api'

import TabBar from '@/components/TabBar'

import { Indicator, Toast } from 'mint-ui';

export default{

components:{

TabBar

},

data(){

return{

accountInfo: {},

//分类选中项

typeId:0,

productList:[],

typeList: [],

}

},

filters:{

formatMoney(val){

if(typeof(val)==="number"){

return val.toFixed(2);

}else{

return val;

}

}

},

methods:{

//获取所有商品分类

selectAllType(){

SelectAllType({id: '',typeName: ''}).then(data =>{

let { errMsg, errCode, value, success, extraInfo } = data;

if(success){

this.typeList = value;

}

else{

Toast(errMsg);

}

});

},

selectType(id){

Indicator.open();

this.typeId = id;

SelectCommodityByTypeId({typeId:this.typeId}).then(data =>{

let { errMsg, errCode, value, success, extraInfo } = data;

if(success){

this.productList = value;

}

else{

Toast(errMsg);

}

Indicator.close();

});

},

//跳转到商品详情页面

goToDetail(id){

this.$router.push({

path:'/productDetail',

query:{

commodityId:id,

userId: this.accountInfo.id

}

})

},

//跳转到搜索页面

goToSearch(){

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

},

},

mounted(){

//查询所有分类

this.selectAllType();

//查询推荐商品

this.selectType(this.typeId);

this.accountInfo = this.$store.state.accountInfo;

}

}

</script>

<style scoped>

.home\_wrap{

padding-bottom: 54px;

}

header{

position: fixed;

left: 0;

top: 0;

width:100%;

box-sizing: border-box;

background: #fff;

z-index:9999;

}

.search\_wrap{

padding: 10px 0;

border-bottom: 1px solid #f3f3f3;

}

.search\_bar{

width: 98%;

height: 35px;

margin: 0 auto;

line-height: 35px;

background: #f3f3f3;

border-radius: 20px;

color: #666666;

font-size: 14px;

}

.product\_wrap{

padding-top: 97px;

}

.product\_type{

box-shadow: 0 2px 3px 2px rgba(0, 0, 0, 0.4);

}

.product\_type ul{

list-style-type: none;

padding-left: 10px;

padding-right: 10px;

overflow-x: scroll;

white-space: nowrap;

}

.product\_type ul li{

display: inline-block;

display: inline-block;

height: 40px;

line-height: 40px;

padding: 0 20px;

box-sizing: border-box;

}

.type-active{

font-weight: bold;

border-bottom: 5px solid #ffe46f;

font-size: 18px;

}

.swipe\_wrap{

height: 180px;

}

.swipe\_wrap img{

width: 100%;

max-height: 180px;

}

.title{

height: 42px;

text-align: center;

}

.title\_line{

display: inline-block;

width: 172px;

height: 20px;

font-size: 18px;

line-height: 40px;

border-bottom: 1px solid #333;

}

.title\_name{

display: inline-block;

height: 40px;

padding: 0 10px;

background: #fff;

}

.type\_image img{

width:100%;

}

.product{

padding: 10px 0;

}

.product~.product{

border-top: 1px solid #e8e8e8;

}

.product\_img{

float: left;

margin-left: 10px;

width: 100px;

height: 100px;

}

.product\_img img{

width: 100%;

}

.product\_info{

position: relative;

margin-left: 100px;

padding: 0 5%;

height: 100px;

box-sizing: border-box;

}

.product\_name{

width: 100%;

height: 60px;

font-size: 14px;

line-height: 20px;

color: #666;

word-break: break-all;

text-align: left;

}

.product\_price{

position: absolute;

bottom: 0;

width: 90%;

font-size: 16px;

}

.vip\_price{

float: left;

color: rgb(171, 9, 35);

}

.vip\_price span {

display: inline-block;

}

.vip\_price span:nth-of-type(1) {

font-size: 10px;

}

.normal\_price {

float: right;

color: #ccc;

position: relative;

}

.normal\_price span {

display: inline-block;

}

.normal\_price span:nth-of-type(1) {

font-size: 10px;

}

.line {

position: absolute;

border-top: 1px solid #ccc;

width: 110%;

left: -5%;

top: 10px;

}

@media screen and (max-width:320px) {

.price {

font-size: 12px;

}

.vip\_price span:nth-of-type(1) {

font-size: 8px;

}

.normal\_price span:nth-of-type(1) {

font-size: 8px;

}

}

</style>

后台管理系统下单逻辑Controller代码

/\*  
 下单  
\*/  
@RequestMapping(value = "/addOrder")  
@ResponseBody  
public JSONObject addAddress(HttpServletRequest request, @RequestBody String json) {  
 JSONObject jsonIn = JSONObject.parseObject(json);  
 JSONObject jsonOut = new JSONObject();  
 try{  
 Order order = new Order();  
 order.setUserId(jsonIn.getIntValue("userId"));  
 order.setAddressId(jsonIn.getIntValue("addressId"));  
 order.setStatus(jsonIn.getIntValue("status"));  
 order.setTotalAmount(jsonIn.getFloatValue("totalAmount"));  
 order.setRemarks(jsonIn.getString("remarks"));  
 order.setNumber("lyf"+new Date().getTime());  
 order.setOrderTime(new Date());  
 order.setIsDelete(0);  
 order.setType(0);  
  
 if(jsonIn.getIntValue("status")==1){  
 User user = userDao.selectUserById(order.getUserId());  
 user.setMoney(user.getMoney()-order.getTotalAmount());  
 userDao.doOrder(user);  
 }  
 orderDao.addOrder(order);  
 JSONArray goodsList = jsonIn.getJSONArray("goodsList");  
 for(int i =0;i<goodsList.size();i++){  
 JSONObject goods = goodsList.getJSONObject(i);  
 GoodsOrder goodsOrder = new GoodsOrder();  
 goodsOrder.setGoodsId(goods.getIntValue("productId"));  
 goodsOrder.setOrderId(order.getId());  
 goodsOrder.setGoodsNumber(goods.getIntValue("amount"));  
 goodsOrder.setGoodsName(goods.getString("productName"));  
 goodsOrder.setGoodsVipPrice(goods.getFloatValue("vipPrice"));  
 goodsOrder.setGoodsMainImage(goods.getString("mainImage"));  
 //查询商品库存  
 int stock = commodityDao.selectCommodityById(goodsOrder.getGoodsId()).getStock();  
 if(stock-goodsOrder.getGoodsNumber()>0){  
 //更新商品库存  
 commodityDao.updateCommodityStockById(goodsOrder.getGoodsId(),stock-goodsOrder.getGoodsNumber());  
 goodsOrderDao.addGoodsOrder(goodsOrder);  
 }  
 else{  
 StaticOptionCode.setResult(jsonOut,21,"",false,"");  
 }  
 }  
 StaticOptionCode.setResult(jsonOut,17,order.getId(),true,"");  
 }catch (Exception e) {  
 e.printStackTrace();  
 StaticOptionCode.setResult(jsonOut,18,"",false,"");  
 }  
  
 return jsonOut;  
}

后台管理系统下单逻辑mybatis配置

<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd" >  
<mapper namespace="cn.com.lyf.wechat.dao.OrderDao">  
  
 <select id="selectOrderByUserId" resultType="cn.com.lyf.wechat.entity.Order">  
 select \* from t\_order where user\_id = #{userId} and is\_delete = '0'  
 <if test="status!=-1">  
 AND status = #{status}  
 </if>  
 ORDER BY order\_time DESC  
 </select>  
  
 <select id="selectOrderById" resultType="cn.com.lyf.wechat.entity.Order">  
 select \* from t\_order where id = #{id} and is\_delete = '0'  
 </select>  
  
 <select id="selectAllOrder" resultType="cn.com.lyf.wechat.entity.Order">  
 select \* from t\_order t where 1=1  
 <if test="order.number!=null and order.number!=''">  
 AND t.number = #{order.number}  
 </if>  
 <if test="order.status!=null and order.status!=-1">  
 AND t.status = #{order.status}  
 </if>  
 <if test="order.userId!=null and order.userId!=''">  
 AND t.user\_id = #{order.userId}  
 </if>  
 <if test="orderStartTime!=null and orderStartTime!=''">  
 AND t.order\_time >= #{orderStartTime}  
 </if>  
 <if test="orderEndTime!=null and orderEndTime!=''">  
 AND #{orderEndTime} >= t.order\_time  
 </if>  
 ORDER BY order\_time DESC  
 </select>  
  
 <update id="changeOrderStatus">  
 UPDATE t\_order SET status = #{status} WHERE id = #{id}  
 </update>  
  
 <update id="deleteOrder">  
 UPDATE t\_order SET is\_delete = 1 WHERE id = #{id}  
 </update>  
  
 <insert id="addOrder" parameterType="cn.com.lyf.wechat.entity.Order" useGeneratedKeys="true" keyProperty="order.id" keyColumn="id">  
 INSERT INTO t\_order (number,status,user\_id,address\_id,total\_amount,remarks,order\_time,is\_delete,type)  
 VALUES  
 (#{order.number},#{order.status},#{order.userId},#{order.addressId},#{order.totalAmount},#{order.remarks},#{order.orderTime},#{order.isDelete},#{order.type})  
 </insert>  
  
</mapper>