



Litemail 系统 需求规格说明书

厦门大学软件学院 2017 级

普通 00 是 AD 而且还是 OOAD 这样的 J2EE 你喜欢吗 小组

组长：郑钦文

组员：颜倩 刘佳硕 蔡郭平 叶颖津

目录

1. 引言	4
1.1 编写目的	4
1.2 项目背景	4
1.3 定义	4
1.4 参考资料	6
2. 任务概述	6
2.1 目标	6
2.2 运行环境	6
2.3 条件与限制	6
3. 数据描述	7
3.1 静态数据	7
3.1.1 类图	7
3.1.2 实体关系图	16
3.2 动态数据	17
3.2.1 活动图	17
3.2.2 状态机图	19
3.2.3 时序图	20
3.2.4 协作图	23
3.2.5 数据流图	28
3.3 数据库介绍	29
3.4 数据词典	29
3.4.1 数据流定义表	29
3.4.2 数据元素定义	30
3.5 数据采集	36
4. 功能需求	37
4.1 功能划分	37
4.1.1 用户目标分析	37
4.1.2 用例图	38
4.2 功能描述	50
4.2.1 用户用例分析	50
4.2.2 管理员用例分析	53
5. 性能需求	56
5.1 数据精确度	56
5.2 时间特性	62
5.3 适应性	63
6. 运行需求	63
6.1 用户界面	63
6.2 硬件接口	64
6.3 软件接口	64
6.4 故障处理	65
6.4.1 硬件故障分析	65

6.4.2 软件故障处理	65
7. 其它需求	66
7.1 安全性需求	66
7.2 可用性需求	66
7.3 可维护性需求	66

1. 引言

1.1 编写目的

为提高 Litemall 系统的易维护性、运行效率、易扩展性、易管理性，并降低代码重复性，我们对 Litemall 系统进行基于面向对象设计思想的重新设计。

本文档基于面向对象的思想，提出系统功能、性能、运行等方面的需求，并以 UML 图对所需数据进行多维描述，为明确系统需求、安排项目规划与进度、组织软件开发与测试等后续工作提供基础性指导。

1.2 项目背景

- a. 开发单位：
厦门大学信息学院 2017 级“普通 00 是 AD 而且还是 OOAD 这样的 J2EE 你喜欢吗”小组。
- b. 主管部门：
厦门大学信息学院。
- c. 项目背景：
Litemall 系统是一个 B2C 网络购物平台，其商品涵盖家居、服装、日用品、母婴产品等多种门类。还拥有团购优惠、组合商品优惠以及不定时发放的商品优惠券等优惠策略，能有效地刺激用户消费。平台还提供了广告平台以及专题平台等商品宣传平台。为在满足繁杂需求的同时保证平台的运行效率和易扩展性、易维护性、易管理性，我们将对 Litemall 系统进行面向对象的重新设计，为系统搭建新的运行框架。

1.3 定义

名词名称	定义
Litemall 系统	本次面向对象设计的主要设计对象。这是一个非面向对象设计的 B2C 网上商城应用，包括许多复杂的功能，将被改造成面向对象设计的系统。
J2EE	Java2Platform, EnterpriseEdition, 是一个为大企业主机级的计算类型而设计的 Java 平台
面向对象设计	面向对象(Object Oriented)是软件开发方法。面向对象是相对于面向过程来讲的，面向对象方法，把相关的数据和方法组织为一个整体来看待，从更高的层次来进行系统建模，更贴近事物的自然运行模式。

团购	团体购物，指认识或不认识的消费者联合起来，加大与商家的谈判能力，以求得最优价格的一种购物方式。根据薄利多销的原理，商家可以给出低于零售价格的团购折扣和单独购买得不到的优质服务
预购	预购即预付货款，是指商业企业为得到所需产品而向生产厂家提前支付部分或全部货款，以支持其生产，稳定产销关系，同时也保证了自己能尽快地得到所需产品
分享	分享指某一用户将某一特定的商品链接转发给其他用户的行为，在本系统的需求中，用户可通过分享来获利，当被分享者点入所分享链接、通过所分享链接进入指定商品并购买等行为发生时，发送链接的分享者将获取一定返利。
返点	管理员在特定商品上设定推广提成规则（如推广 10 件，可获得 1%的返点，推广 11-50 件可获得 1.3%的返点）。允许用户通过微信用二维码或链接分享商品，如果其他用户通过二维码点击浏览商品，则认为分享成功。如果用户在分享成功后设定的有效时间内（如七天）购买了该产品，且未退货。则计入分享用户的返点。当商品有多个分享成功的情况，返给在有效时间内最早分享成功的用户，或者平分给所有有效时间内分享成功的用户。
组合商品	即把两种以上的商品组合成套装购买，套装商品不允许单独退货，但允许单独换货的服务。
优惠券	优惠券是一种可降低购买商品所花费用的工具，本系统中优惠券分为三类，面向所有商品的优惠券，面向某一类商品的优惠券，面向某一特定商品的优惠券，通过适当的让利幅度，从而达到增加销售的目的，优惠券有一定的使用数量限制，用户可通过本电商平台系统进行领取。

1.4 参考资料

文档名称	发布日期	作者	来源
《讨论课管理系统需求规格说明书》	2018 年 11 月 5 日	厦门大学软件学院 2016 级 3-2 小组	作者授权
《OOAD_Grading_2019》	2019 年 10 月 27 日	邱明	作者授权

2. 任务概述

2.1 目标

为了对 Litemall 系统进行面向对象设计,我们需要设计若干个对象来满足系统的需求,并保证对象之间符合高内聚、低耦合的原则。

在目前阶段,本文档设计这些对象所包含的属性以及类之间的联系,不涉及类的方法设计。

首先对原本的 Litemall 系统进行四个维度的需求分析。根据已有的前端反向分析 Litemall 系统应该具有功能,并为新需求设计功能,按照面向对象的方法,产出对应的类图、实体关系图、活动图、对象状态机图、数据流图。

在完成面向对象的需求分析之后,为提高开发效率,课程成员应进行模块分工,除此之外需要形成标准工作小组来对模块工作进行宏观调整和控制。

2.2 运行环境

操作系统: Microsoft Windows 7/10、Mac、IOS、Android

支持环境: Web 端、微信小程序

数据库: Microsoft SQL Server 2000

2.3 条件与限制

1. 必须使用 JavaEE 或者 .NET 进行开发。
2. 必须能在不同平台上稳定运行,如能在 IE10、IE edge16、Chrome、Safari、FireFox、微信小程序等平台上稳定运行。
3. 根据《OOAD_Grading_2019》,对原有的 Litemall 系统,增加若干新需求,修改若干原有需求。对于这些需求的增加和修改,要求保证能满足条件与限制 2。
4. 项目必须能在本学期期末以前完成部署并投入实用。
5. 本项目的开发过程中产生的所有内容以及产品均须符合中华人民共和国相关法律规定。

3. 数据描述

3.1 静态数据

3.1.1 类图

3.1.1.1 类与属性

类名	需求及功能	属性
User	<ol style="list-style-type: none">User 类要存储用户名、头像等用户信息；User 类要有一个相应的账号和对应的密码；User 类要记录该用户得到了多少返点，来满足分享商品获利的功能需求；	userName account password profilePicture backPoint
Address	<ol style="list-style-type: none">为了满足用户设定不同的收货地址，以及对收货地址进行更改的需求，建立了 Address 类；Address 类应该存储用户 ID、地址的各级详细信息以及用户的手机号和姓名；	name phoneNumber province addressDetail usrID
SharedLog	<ol style="list-style-type: none">为了满足用户查询自己被分享了哪些商品，建立了 SharedLog 类；SharedLog 类应该存储某次分享的产生时间，该记录的 ID 以及 usrID；SharedLog 类为了满足其有时效性的特点，需要记录状态；	status BirthTime sharedID shareID
BackPointLog	<ol style="list-style-type: none">为了满足记录用户通过分享而获得返点、以及使用返点购买商品的行为，建立了 backPointLog 类；backPointLog 类应该储存消耗或获得返点的来源商品 ID、要存储消耗或获得返点的时间以及数量、还要记录用户的 ID；	spendHistoryDate reciveHistoryDate comeFrom pendPointHistory recivePointHistory usrID

Share	<ol style="list-style-type: none"> 1. 为了满足用户分享商品获得返利的需求，建立 Share 类； 2. Share 类应该存储商品 ID、分享成功的次数以及用户 ID； 3. Share 类为了满足分层返利策略的需求，并且提高运行效率，需要记录到下一级的返利层级需要的分享成功件数以及当前所处于的层级； 	goodsID successTimes currentLevel nextHighLimit usrID
ShareRules	<ol style="list-style-type: none"> 1. 为了满足管理员发布针对商品的推广返利政策，用户的分享层级更新有据可循，建立了 ShareRules 类； 2. ShareRules 类应该记录商品 ID； 3. ShareRules 类为了满足分层返利策略的需求，需要记录下一级别的推广次数下限、当前级别以及返点百分比； 	lowLimits rebatePercentage goodID level
Order	<ol style="list-style-type: none"> 1. Order 类需要储存总价、下单时间、地址、物流单号、用户 ID、状态等订单基本信息； 2. Order 类为了满足云端同步需求，需要记录是否已经被同步以及是否被修改； 3. Order 类为了满足退换货需求，需要记录该订单是否是退换货订单以及类型； 4. Order 类为了满足分享返利需求，需要记录是否是被分享后下的订单； 	totalPrice orderTime address orderStatus isReturned status isShared espressID isModified isSync usrID
OrderItem	<ol style="list-style-type: none"> 1. 为了满足 Order 订单拆分存储而建立 OrderItem 类。该类的每个实例都是主 Order 中的每个商品的订单信息，主码是 Order 的 ID 以及商品的 ID； 2. 为了满足单个商品退换等 	discountPrice soldPrice quantity combineID

	操作, 单独记录该子订单的折扣、合计销售、商品 ID、主订单 ID、组合商品 ID、以及数量	
OrderShareItem	<ol style="list-style-type: none"> 1. 为了满足完成订单增加分享成功次数的需求, 建立了 OrderShareItem 类; 2. OrderShareItem 类应该存储订单状态、订单 ID 以及分享 ID; 	status orderID shareID
Goods	<ol style="list-style-type: none"> 1. Goods 类中需要储存商品名称、价格、类别、库存、商品细节、浏览图片、详细图片等商品信息; 2. Goods 类为了满足上下架以及缺货的需求, 需要记录商品状态; 3. Goods 类为了满足组合销售的需求, 需要记录是否有商品组合; 4. Goods 类为了防止超卖, 需要有预扣库存的功能; 5. Goods 类为了满足增加参数的功能, 需要存储其参数; 	goodsStock goodsName goodsPrice category details browseImage detailImage status isCompositionGoods preReducingStock hasSKU
SKU	<ol style="list-style-type: none"> 1. 为了满足商家能为商品设定参数, 提高人机交互体验, 建立了 SKU 类; 2. SKU 类需要记录其属于哪个商品; 3. SKU 类需要记录其库存、价格、以及各种参数组合 4. SKU 类为了提高人机交互体验, 允许商家为 SKU 自定义 ID; 	Parameter1~n Stock Price goodsID customerID
PresaleGoods	<ol style="list-style-type: none"> 1. 为了满足商家设立预售商品的活动, 建立了商品类 Goods 的子类 PresaleGoods 类; 2. PresaleGoods 类应该存储定金以及尾款、缴纳定金的支付开始结束时间、补付尾款的开始结束时间、以及该预售处于的状态; 	deposit depositStartTime depositEndTime finalPayment finalPaymentStartTime finalPaymentEndTime presaleStatus goodsNumberLimit

	3. 为了防止超卖，在 PresaleGoods 类中设定预售商品的购买上限数量；	
PresaleOrder	1. 为了满足预售商品的特殊支付状态，建立订单类 Order 的子类 PresaleOrder； 2. PresaleOrder 类需要记录当前订单的状态；	status
GoodsCategory	1. 为了满足商品类别的增删查改，建立了 GoodsCategory 类； 2. GoodsCategory 类需要记录 ID 以及名字； 3. GoodsCategory 类为了满足分级分类，需要记录父类的 ID；	categoryID categoryName fatherCategory
CompositionGoodsItems	1. 为了满足管理员新建商品组合购买优惠策略，建立了 CompositionGoodsItems； 2. CompositionGoodsItems 类应该记录该组合 ID 以及包含的商品 ID；	compositionGoodsID goodsID
Comment	1. 为了满足对商品以及专题广告进行评论的功能，建立了 Comment 类； 2. Comment 类需要记录评价的内容、评论时间、以及状态等信息；	content commentaryTime status
GoodsComment	1. 由于商品评论和专题评论的消亡机制不一样，故建立 Comment 的子类 GoodsComment； 2. GoodsComment 类需要记录商品 ID；	goodsID
SpecialTopicComment	1. 由于商品评论和专题评论的消亡机制不一样，故建立 Comment 的子类 SpecialTopicComment； 2. SpecialTopicComment 类需要记录专题 ID	specialTopicComment
SpecialTopic	1. 为了满足商家撰写专题的需求，建立了 SpecialTopic 类；	titles image

	2. SpecialTopic 类应该记录标题以及内容图片;	
CartItem	<ol style="list-style-type: none"> 为了满足用户使用购物车一次性购买多种商品的需求, 建立了 CartItem 类; CartItem 类应该储存购物车的用户 ID、商品 ID、商品数量; 	quantity usrID goodsID
FootPrints	<ol style="list-style-type: none"> 为了满足用户查看过去自己的浏览足迹的需求, 建立了 FootPrints 类; FootPrints 类应该储存时间、用户 ID 以及商品 ID; 	browseTime usrID goodsID
Favorites	<ol style="list-style-type: none"> 为了满足用户收藏某件商品, 日后再浏览或购买的需求, 建立了 Favorites 类; Favorites 类应该存储用户 ID 以及商品 ID 	usrID goodsID
Brand	<ol style="list-style-type: none"> 为了满足商家设立品牌分类, 让用户能根据品牌检索商品, 建立 Brand 类 Brand 类应该记录该品牌名字以及品牌描述、图片等信息; 	brandName brandDescription image
CouponTopParent	<ol style="list-style-type: none"> 为了满足管理员创建不同级别的优惠券, 建立 CouponTopParent 类, 它拥有两个子类 CouponForOneGoods 类和 CouponForOneCategory 类; CouponTopParent 类全场通用, 它应该记录这种优惠券的 ID、折扣、有效时间是否在有效期内、是否已经可用等信息; CouponTopParent 类为了防止超量发放优惠券, 应该储存用户该种优惠券的总发行量、剩余数量; 	couponID discount startDate endDate numberLimitForOne totalCoupons remainingCoupons isUsed isActive
CouponForOneGoods	<ol style="list-style-type: none"> 为了满足优惠券分级, 建立 CouponTopParent 的子类: CouponForOneGoods 类, 该级别优惠券只适用于一种商品; CouponForOneGoods 应该记录商品 ID; 	goodsID

CouponForOneCategory	<ol style="list-style-type: none"> 为了满足优惠券分级, 建立 CouponTopParent 的子类: CouponForOneCategory 类, 该级别优惠券只适用于一种商品类别; CouponForOneCategory 应该记录商品类别 ID; 	categoryID
CouponItem	<ol style="list-style-type: none"> 为了满足用户领取并拥有优惠券的需求, 建立了 CouponItem 类; CouponItem 类应该记录用户 ID、优惠券 ID、用户拥有的同种优惠券数量、以及使用了多少张; 	usrID couponID holding used
Refund	<ol style="list-style-type: none"> 为了满足用户在购物完成或未完成的情况下进行退换货的复杂操作, 建立 Refund 类; Refund 类应该存储退款金额、退款类型、订单详细信息、退款状态、退款理由、回寄物流单、照片、以及用户的描述; 	refundFee status refundType orderInfomation refundResons refundExpressID usrImage userDescription
Administrator	<ol style="list-style-type: none"> 为了满足用户和管理员的权限分离, 建立 Administrator 类; Administrator 类应该记录该管理员所拥有的权限 	authority
Advertisement	<ol style="list-style-type: none"> 为了满足管理员能上线广告创造营收的需求, 建立 Advertisement 类; Advertisement 类应该记录该广告的有效时间、图片、跳转链接、以及是否生效; Advertisement 类为了满足在没人买广告的时候播放默认招商广告应记录是否为默认广告; 	startTime endTime isActive isDefault jumpURL mage
Log	<ol style="list-style-type: none"> 为了提高安全性, 建立记录管理员行为的 Log 类; Log 类应该记录生成这条 log 记录的动作者 ID、时间日期、状态、内容等信息; 	createID date status content
Groupon	<ol style="list-style-type: none"> 为了满足管理员能根据团 	startTime

	<p>购优惠规则创建团购活动，建立了 Groupon 类；</p> <p>2. Groupon 类应该记录团购的开始结束时间、团购状态、现在已经参团的商品数量、优惠折扣；</p> <p>3. Groupon 类为了满足分级团购的要求，还要记录当前的级别以及下一级别需要的参团数量；</p>	<p>endTime</p> <p>status</p> <p>currentGoodsNumber</p> <p>discountPhase</p> <p>nextHighLimit</p>
GrouponOrder	<p>1. 因为团购订单具有特殊性，当团购活动时间限制到了之后，用户可以选择当场取消订单或者按照当前折扣等级继续购买，这过程中可能涉及补差价或者退差款的活动，因此建立 Order 类的子类 GrouponOrder 类；</p> <p>2. GrouponOrder 类应该存储当前团购状态以及团购活动 ID；</p>	<p>status</p> <p>grouponID</p>
GrouponRules	<p>1. 为了满足团购活动分级优惠的需求，建立了 GrouponRules 类；</p> <p>2. GrouponRules 类应该存储团购活动 ID，优惠级别、这个级别的数量下限、这个级别下的折扣；</p>	<p>GrouponID</p> <p>discount</p> <p>lowLimits</p> <p>phase</p>

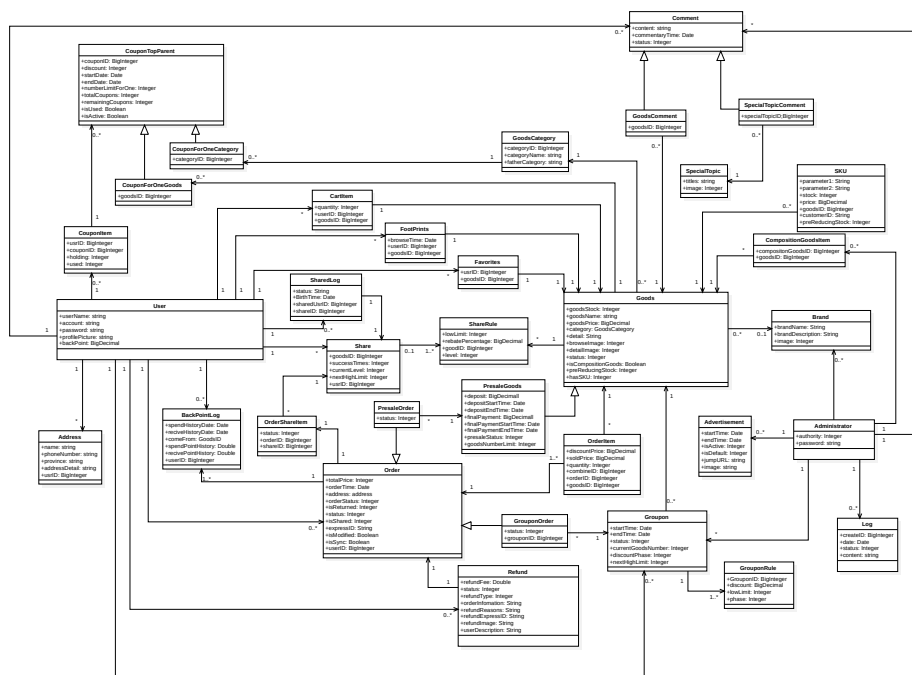
3.1.1.2 类之间的关系

类之间有以下 6 种联系方式:

联系	类之间的关系
1...* — 1	一或多对一关系。指该类 A 的所有实例都与类 B 的某一个实例关联，且类 A 和类 B 都必须有实例；
1—0...1	一对零或一关系。指该类 A 的每一个实例，都与类 B 的某一个实例关联，且类 A 必须有实例、类 B 可以没有实例；
0...*—1	零或多对一关系。指该类 A 的所有实例都与类 B 的某一个实例关联，且类 A 可以没有实例，类 B 必须有实例；
1—0...*	一对零或多关系。指该类 A 的每一个实例，都与类 B 的多个实例关联，且类 A 必须有实例、类 B 可以没有实例；
1—1...*	一对一或多关系。指该类 A 的每一个实例，都与类 B 的多个实例关联，且类 A 和类 B 都必须有实例；

3.1.1.3 概念类图

Model1::ClassDiagram1



概念类图描述了系统中类的形态以及类之间的关系。概念类图对于系统的数据库的建立以及数据结构的设计有着重要的指导意义。本部分会描述一些本系统的特别的类的数据结构,

没提及的类请对照概念类图阅读。概念类图所确定的数据库结构会转化为实体关系模型，具体请见下文的实体关系模型。

1. 对于商品类`Goods`，其有一个子类预售商品类`PresaleGoods`与其形成继承关系。当管理员要上架预售商品的时候可以新建一个预售商品`PresaleGoods`实例，其有`Goods`部分的所有属性外加`PresaleGoods`中的所有属性。

2. 对于商品类`Goods`，其有一个属性是标记位变量`hasSKU`，这个标记位会记录该商品是否有`SKU`，以满足上架同一个商品拥有多种规格的需求。当`hasSKU`为`true`的时候，则会去查找`SKU`类。`SKU`类的实例都有`GoodsID`，因此某个拥有`SKU`的商品都能通过`GoodsID`查找到自己的`SKU`。

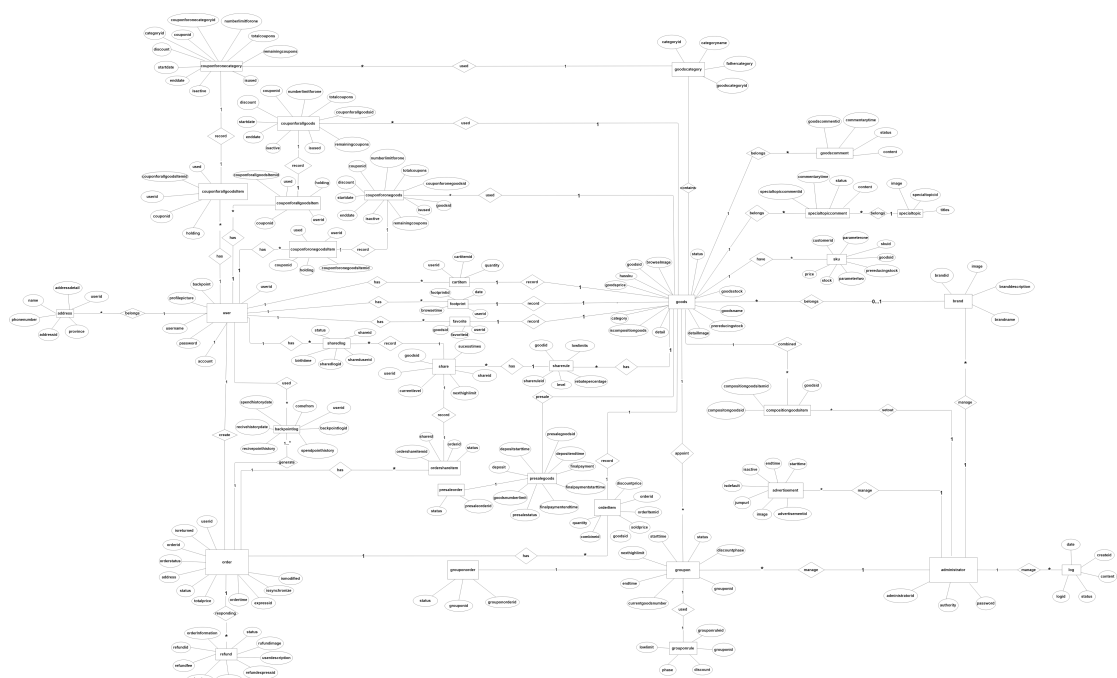
3. 对于订单类`Order`，其有两个标记位变量分别是`isModified`和`isSync`。这两个标记位会记录该订单是否被修改过以及是否已经被同步。当订单要进行云端同步的时候，会检查所有未被同步过的订单，也就是`isSync`为`false`的订单，进行同步，然后将`isModified`置为`false`，`isSync`置为`true`。然后会检索所有`isSync`为`true`，而`isModified`为也为`true`的订单，进行同步，然后将`isModified`置为`false`。这样则能提高订单同步的效率。

4. 对于优惠券类`CouponTopParent`，其有两个子类`CouponForOneGoods`、`CouponForOneCategory`。这是为了满足优惠券分级别的需求。父类以及两个子类，一共可以创建三种范围的优惠券：“全场适用”、“类别适用”、“商品适用”。而`CouponItem`类则是表达用户所拥有的优惠券的类。用户每次领取同种优惠券，则会创建`CouponItem`类的实例。这样的设计能消除用户和优惠券之间的多对多关系，缓解高并发的情况。

5. 对于团购类`Groupon`，里面记录了团购的大部分规则，其实例是每一个团购活动，这个团购活动会记录当前的折扣以及有效时间。为了实现分级团购的需求，我们还设计了分级团购规则类`GrouponRules`。`GrouponRules`的实例中会记录`GrouponID`，以便于团购活动根据自己的`ID`来查询团购规则。`GrouponRules`的实例中的属性“到达下一级别所需要的件数”这个常量也会被记录在`Groupon`的实例中，当团购数量达到了这一个量，`Groupon`实例就会查询下一条团购规则，也就是下一级别的团购规则，并更新自己的折扣和“到达下一级别所需要的件数”。以此来满足分级团购的需求。

6. 对于分享类`Share`，其实例是一个用户对一个商品的分享行为统计。因为分享返利也有一个分级的情况，于是分享的类关系中也有与`Groupon`和`GrouponRules`类似的`Share`和`ShareRules`关系。

3.1.2 实体关系图



此次课程设计中，E-R 图由概念类图转换而来，整体上严格遵循三个模型的一致性和设计可追踪性。

1. 本设计文档中，MySQL 数据库的设计严格遵循阿里规约：

(1) 在表达与“是否”概念有关时，均使用`is_xxx`的方式，如`isused`，`isactive`等属性。

(2) 在`pojo`类`中的任何布尔类型的变量都未加`is`前缀。

(3) 所画 E-R 图中所有的表名，字段名均用小写字母表示，没有出现`desc`、`range`，`match`，`delayed`等保留字，表名均未使用复数。

(4) 设计小数类型的均采取了`decimal`，而未使用如`float`，`double`等数据类型。

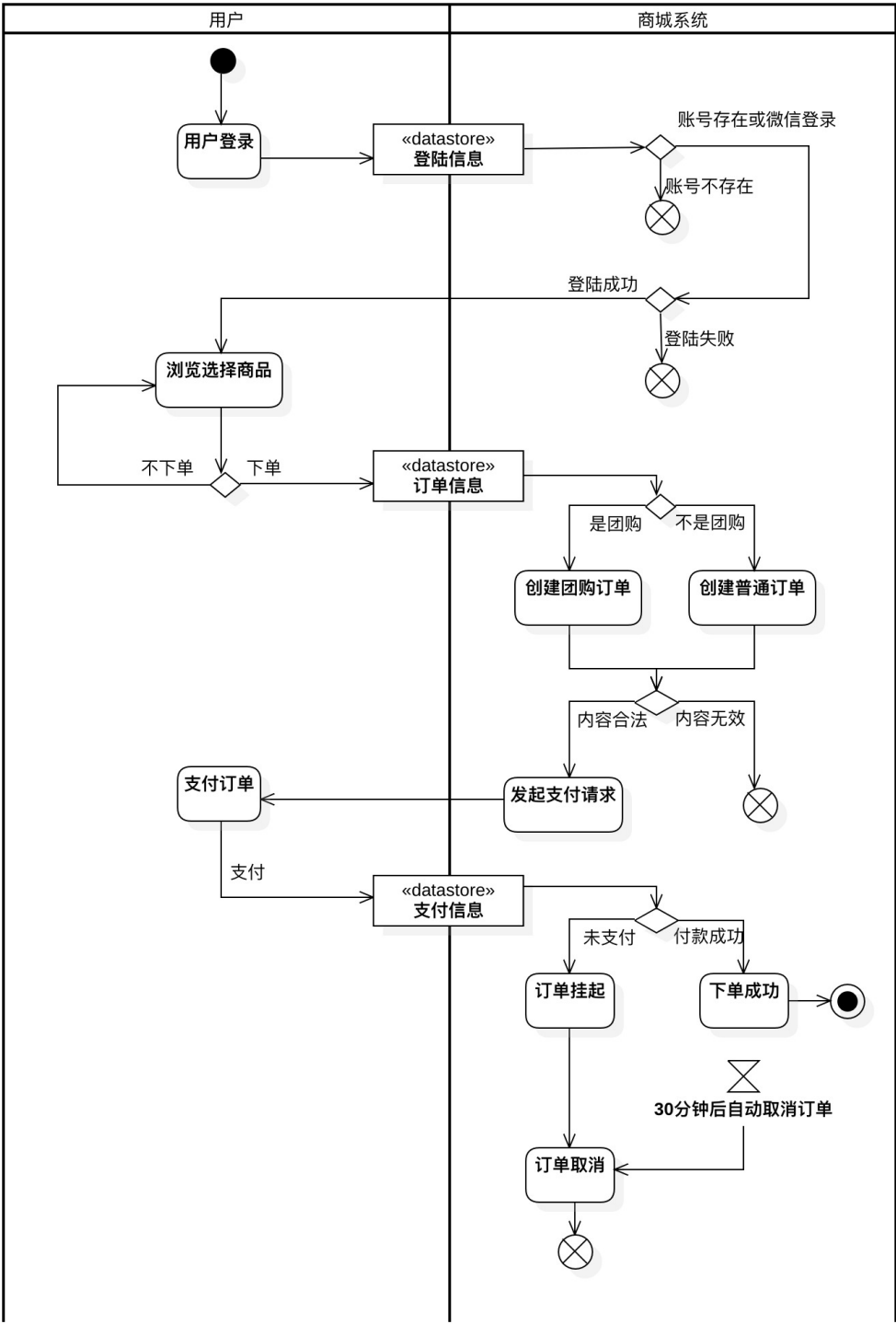
2. 在对象模型向`E-R`图`转换的过程中，`orderitem`等`item`类`转化为关系表。

3. 相比对象模型，`E-R`图在`coupon`的继承关系进行了部分调整，将`coupon`和两个子类当作三个表进行存储。同理，在处理`order`和`goods`的子类上也采取相同方法，但是由于拆开会导致连线数量大量增加，因此不在`E-R`图`中画出。

3.2 动态数据

3.2.1 活动图

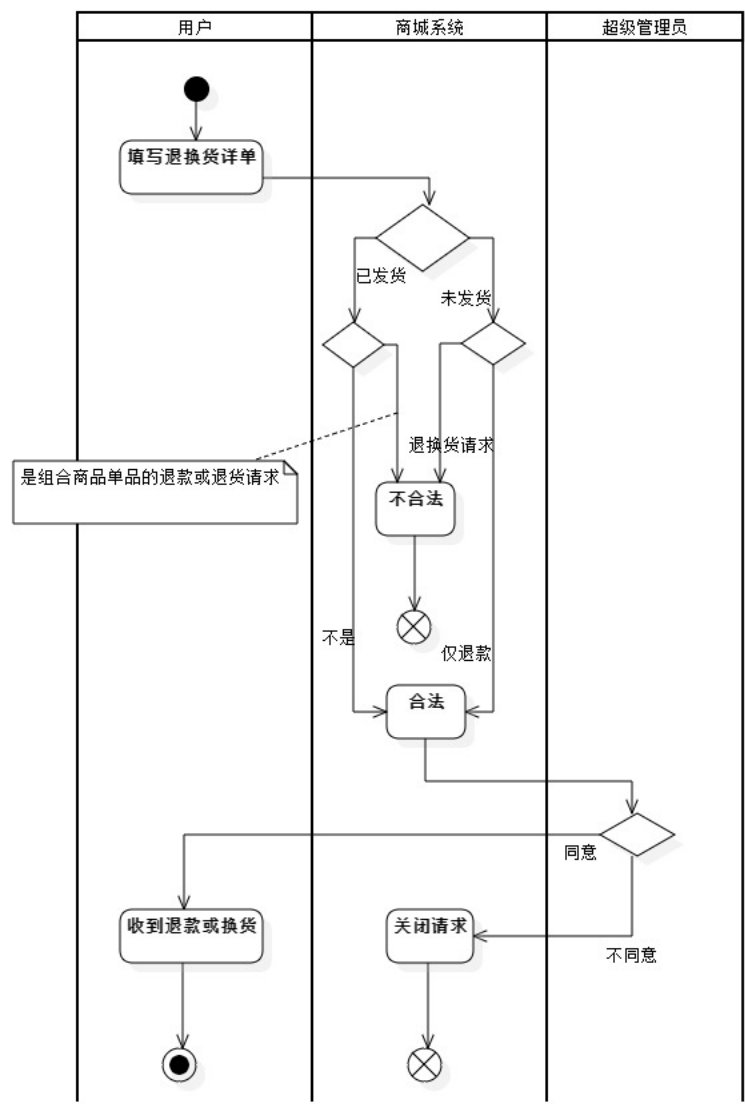
3.2.1.1 用户购物活动图



用户购物活动图包括用户从登陆商城到完成订单支付之间进行的活动。

1. 用户登陆商城，可以选择注册账号、账号登陆或微信登录，系统针对不同登陆方式进行验证。登陆成功后用户开始浏览商城。
2. 用户看中商品后选择加入购物车或直接购买
3. 决定购买后有两种情况：（1）参加团购，此时不可使用其他折扣（2）未参加团购，普通订单，可以使用优惠券和返点。
4. 生成订单或团购订单，进行付款，付款后下单成功。此时取消付款订单失败，订单未付款超过 30 分钟自动取消。

3.2.1.2 用户退换货活动图



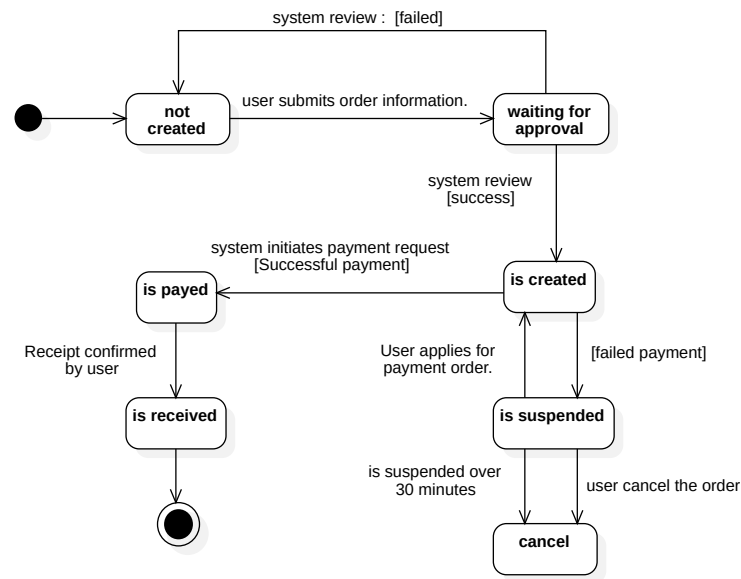
用户退换货活动图从提交退换货申请到商城管理员同意请求之间进行的活动。

1. 用户填写并提交退换货详单，内容不合法则流程结束，合法将表单信息提交给管理员。
2. 管理员审核退换货请求，同意则进入退换货流程，用户收到退款退货凭证；不同意则关闭请求，流程结束。

3.2.2 状态机图

在活动图的基础上,可以发现订单的逻辑较为复杂,活动图对订单状态的描述并不直观,因此绘制订单状态机图,直观展示订单的各种可能状态以及之间的转换关系。

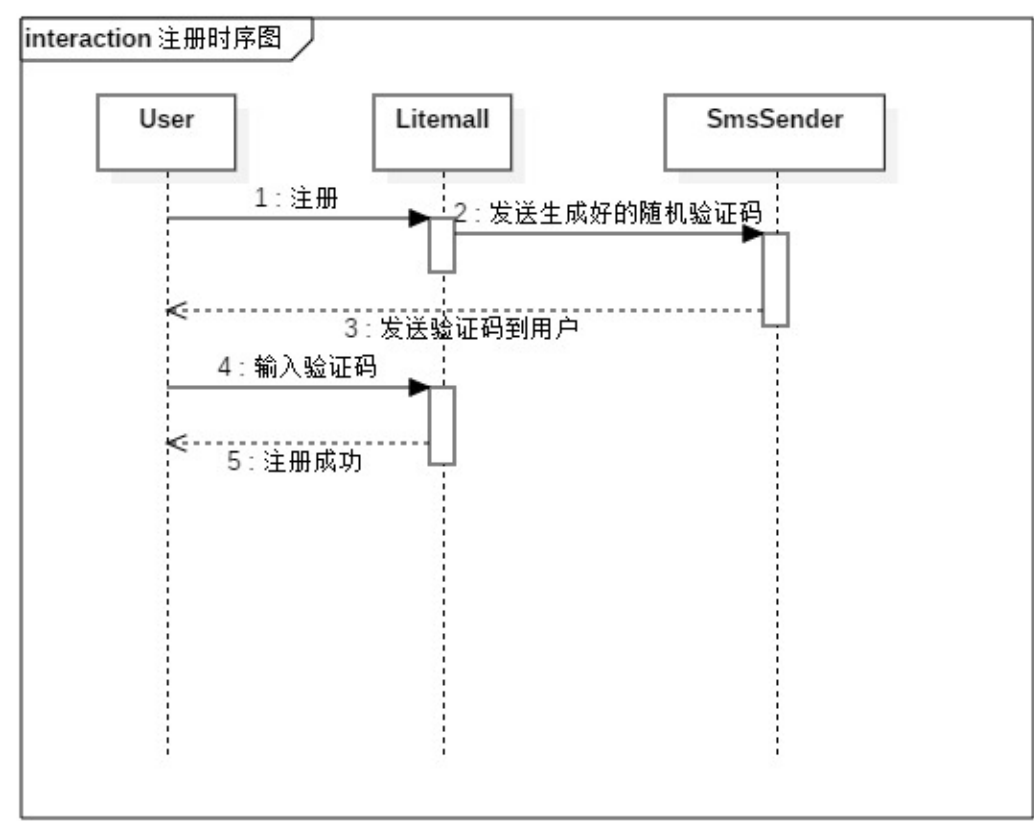
StateMachine1::StatechartDiagram1



3.2.3 时序图

全部时序图数量众多，本文档只画出和第三方系统有交互的时序图。

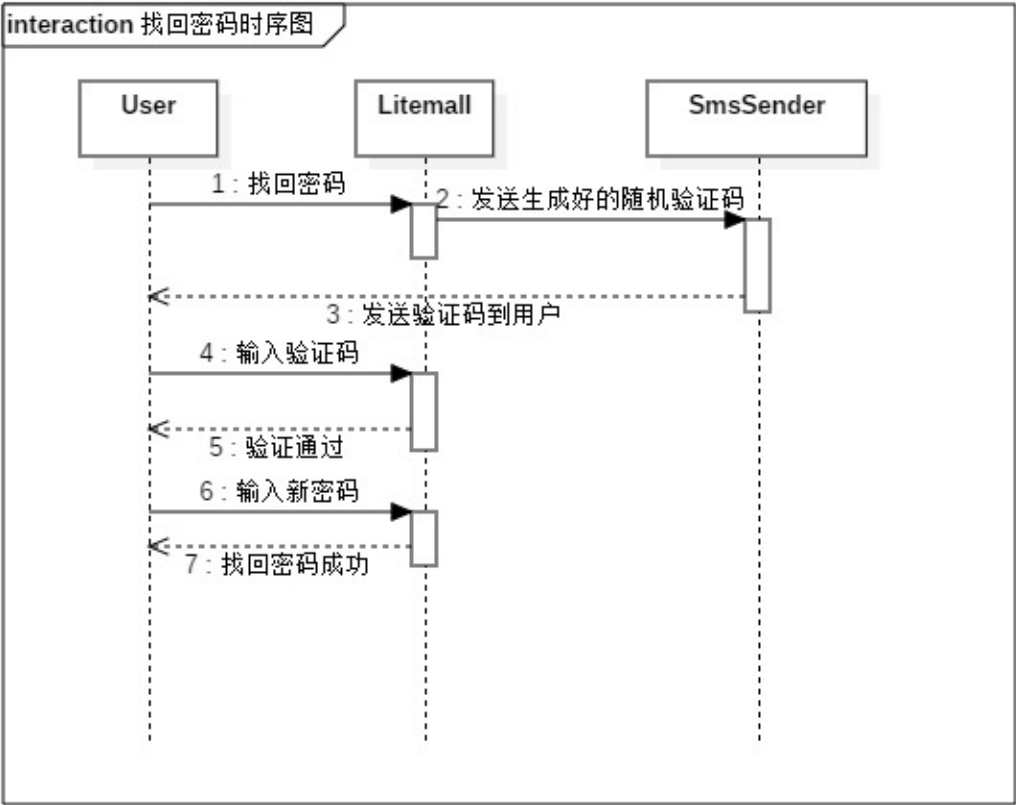
3.2.3.1 用户登陆时序图



用户注册时序图为用户注册时事件的流程

1. 用户发起注册请求
2. 服务器向第三方请求代发验证码
3. 用户输入合法验证码，注册成功，否则注册失败

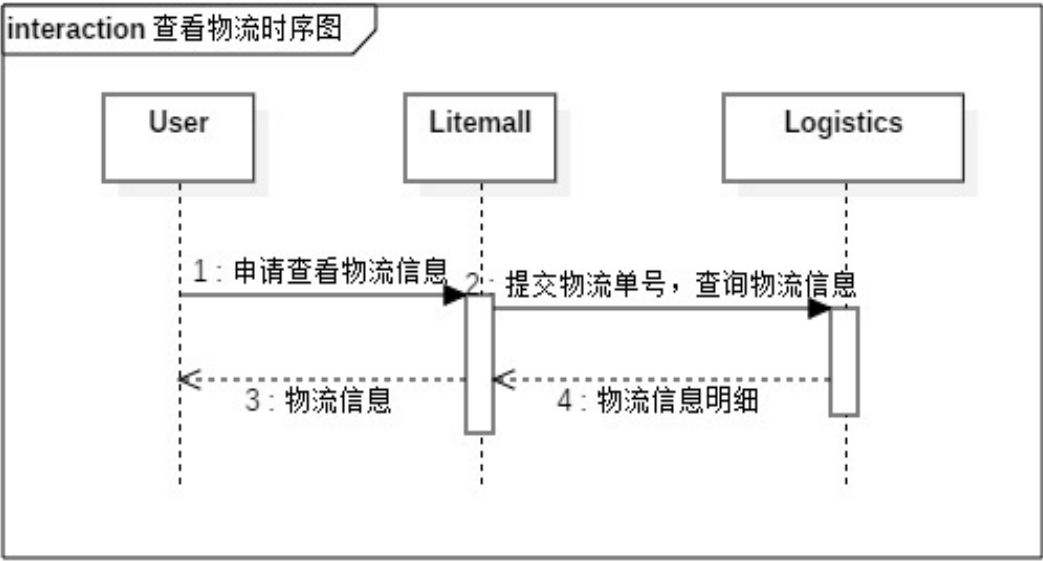
3.2.3.2 用户找回密码时序图



用户找回密码时序图为用户找回密码时事件的流程

- 1. 用户发起找回密码请求
- 2. 服务器向第三方请求代发验证码
- 3. 用户输入合法验证码，则验证通过，否则验证不通过
- 4. 用户输入新密码，找回密码成功

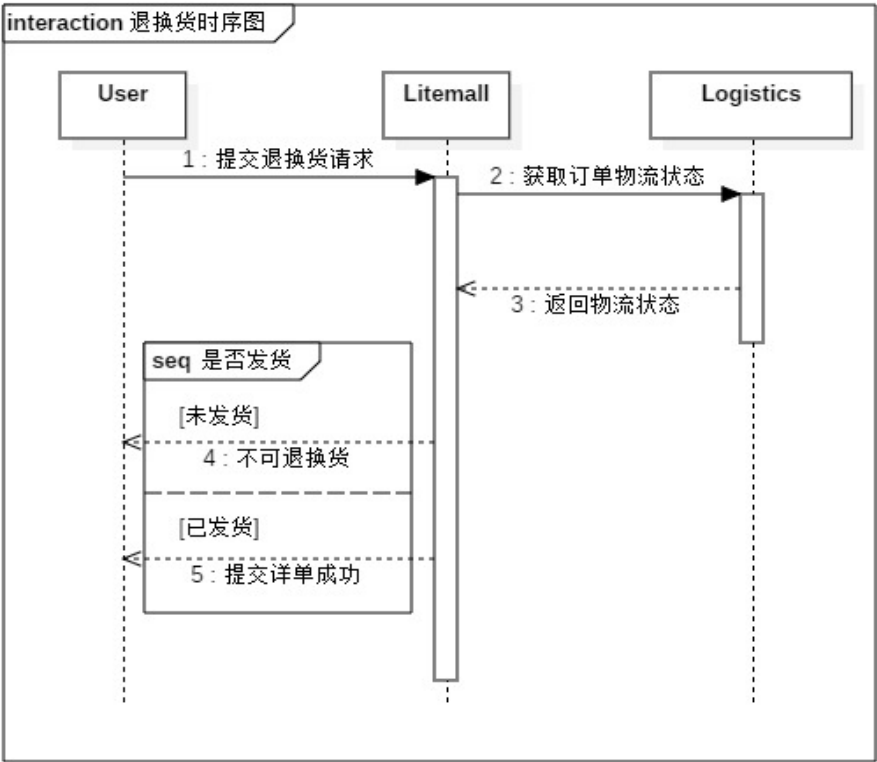
3.2.3.3 用户查看物流时序图



用户查看物流时序图为用户查看物流时事件的流程

- 1. 用户发起查看物流请求
- 2. 服务器向第三方提交订单绑定的物流单号，申请查询物流信息
- 3. 第三方返回物流信息，服务器处理后显示给用户

3.2.3.4 用户退换货时序图

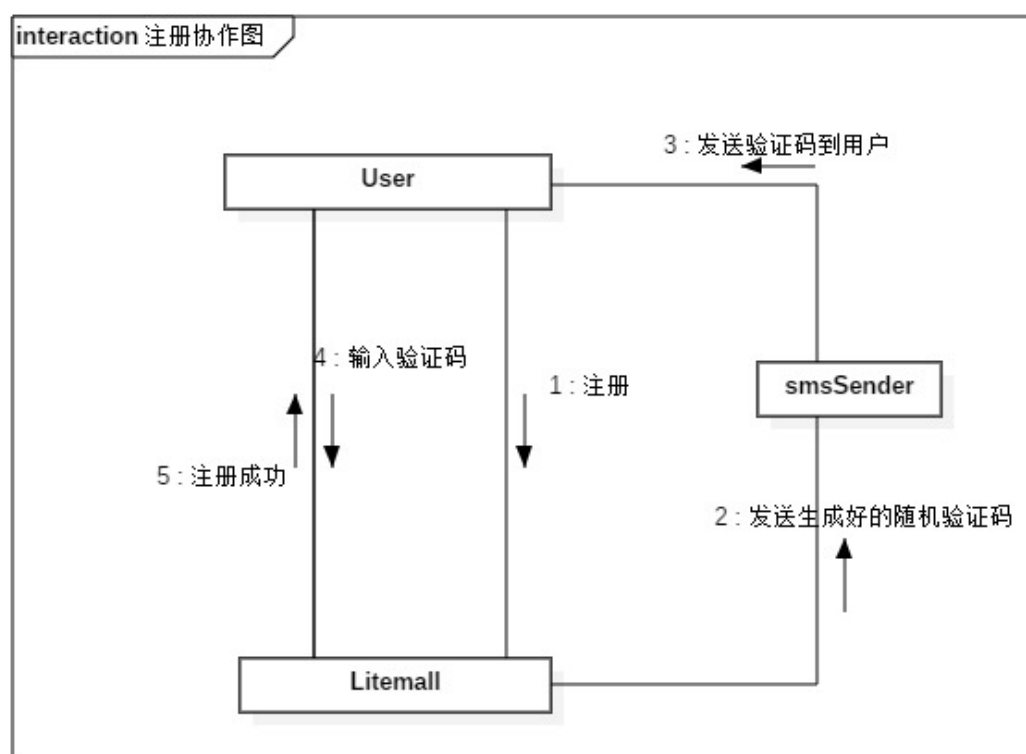


用户注册时序图为用户注册时事件的流程

1. 用户发起退换货请求
2. 服务器向第三方提交订单绑定的物流单号，申请查询物流信息
3. 若未发货，则服务器拒绝用户请求，否则用户成功提交退换货请求

3.2.4 协作图

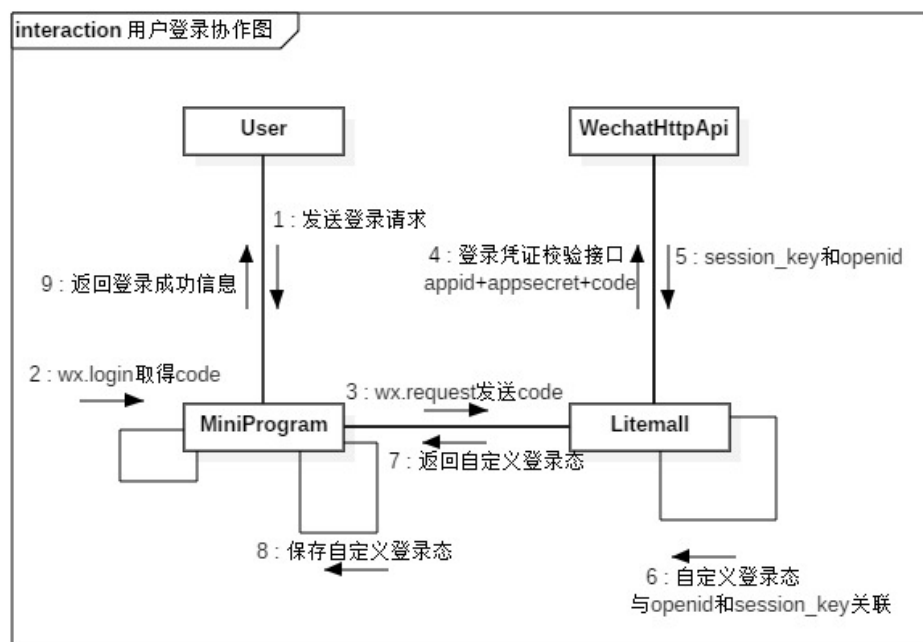
3.2.4.1 用户注册协作图



用户注册协作图为用户注册时各类的协作关系

1. 用户发起注册请求
2. 服务器向第三方请求代发验证码
3. 用户输入合法验证码，注册成功，否则注册失败

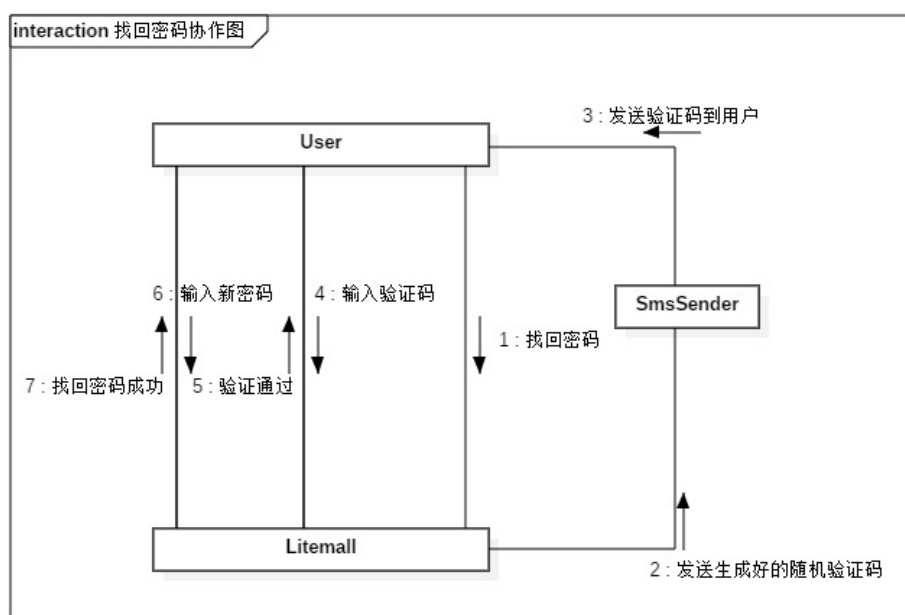
3.2.4.2 用户登录协作图



用户登录协作图为用户登录时各类的协作关系

1. 用户发起登录请求
2. 小程序在本地获得识别码并发送给服务器
3. 服务器登录凭证校验接口，获得微信登录凭证
4. 服务器处理凭证，返回自定义登录态到小程序
5. 小程序保存自定义登录态，用户登录成功

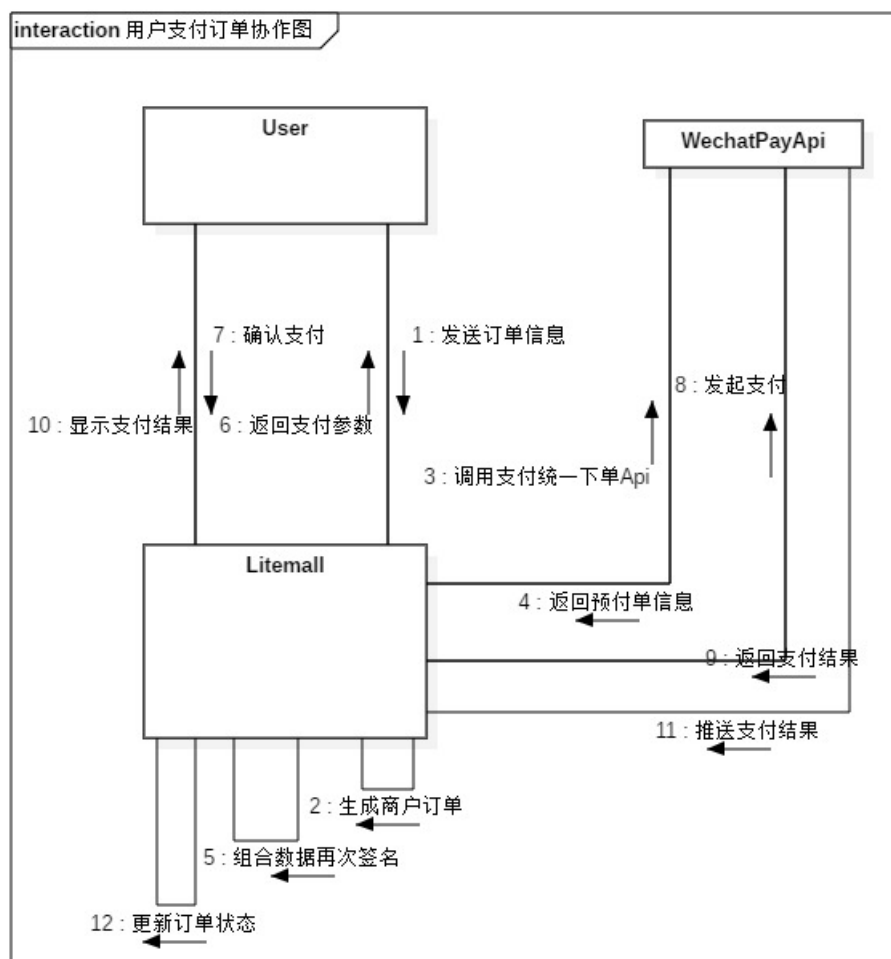
3.2.4.3 用户找回密码协作图



用户找回密码协作图为用户找回密码时各类的协作关系

1. 用户发起找回密码请求
2. 服务器向第三方请求代发验证码
3. 用户输入合法验证码，则验证通过，否则验证不通过
4. 用户输入新密码，找回密码成功

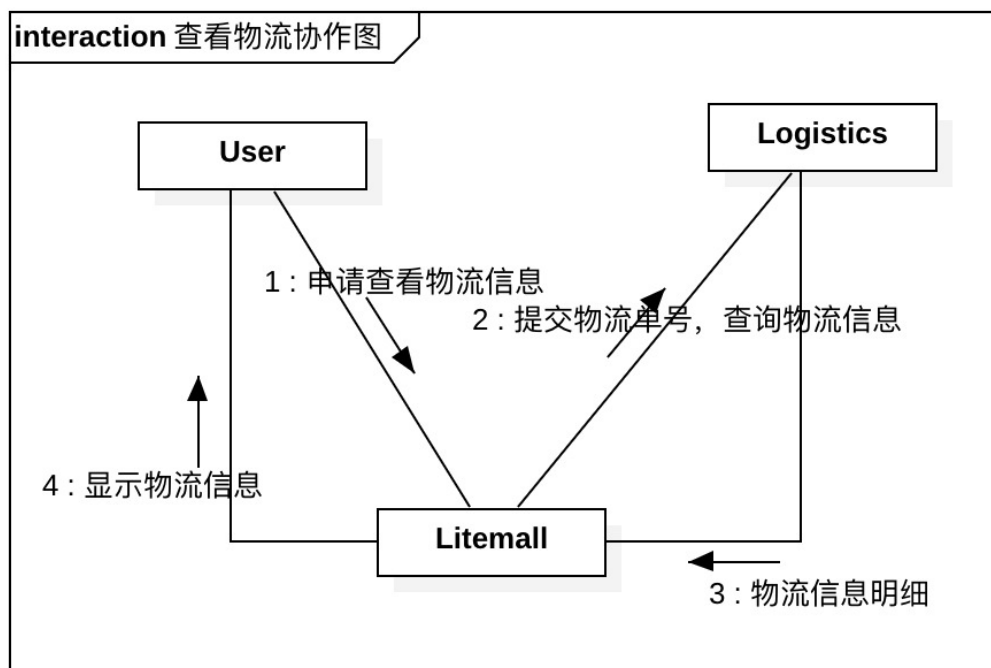
3.2.4.4 用户支付订单协作图



用户支付订单协作图为用户支付订单时各类的协作关系

1. 用户提交订单信息到服务器
2. 服务器生成订单，调用第三方支付 API
3. 第三方返回预付单信息，服务器组合数据再次签名
4. 服务器组合支付参数生成支付页面显示给用户
5. 用户确认支付，服务器代为向第三方发起支付
6. 支付成功，第三方返回登录结果，服务器通知用户支付成功
7. 第三方持续推送支付结果，直至服务器收到、更新订单状态为止

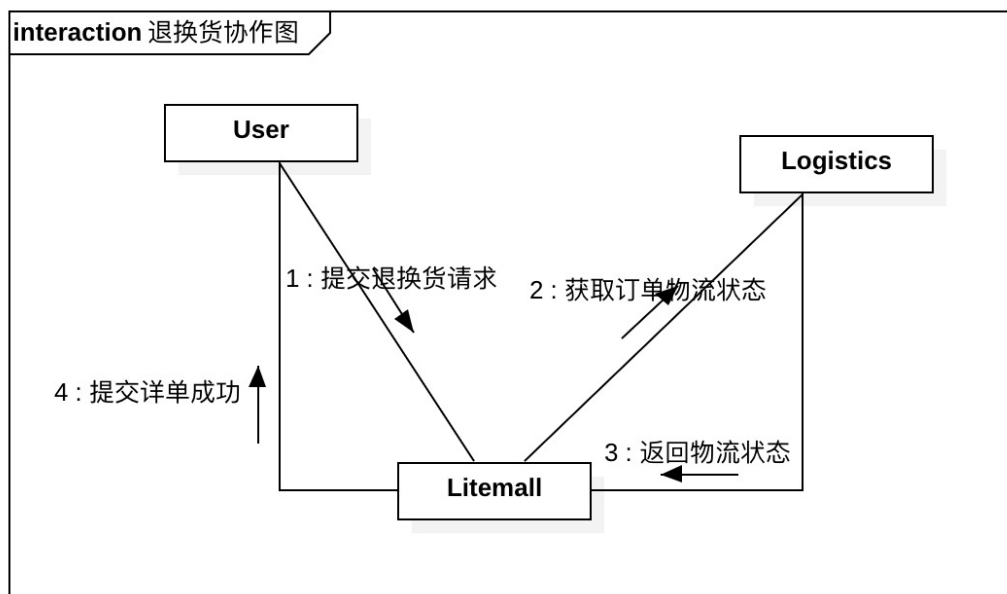
3.2.4.5 用户查看物流协作图



用户查看物流协作图为用户查看物流时各类的协作关系

1. 用户发起查看物流请求
2. 服务器向第三方提交订单绑定的物流单号, 申请查询物流信息
3. 第三方返回物流信息, 服务器处理后显示给用户

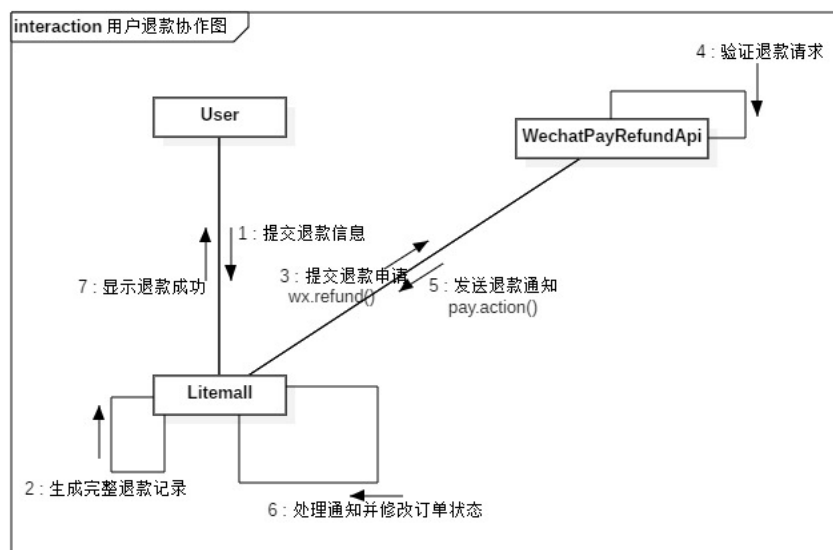
3.2.4.6 用户退换货协作图



用户退换货协作图为用户退换货时各类的协作关系

1. 用户发起退换货请求
2. 服务器向第三方提交订单绑定的物流单号，申请查询物流信息
3. 若未发货，则服务器拒绝用户请求，否则用户成功提交退换货请求

3.2.4.7 用户退款协作图



用户退款协作图为用户退款时各类的协作关系

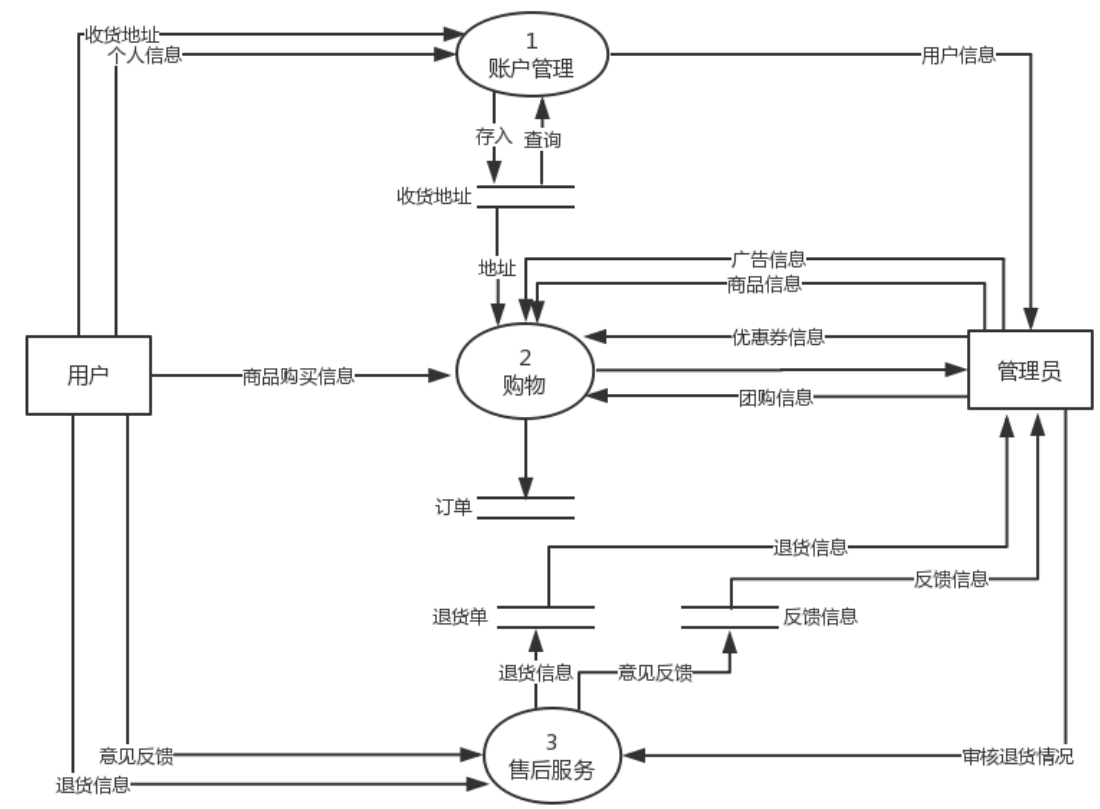
1. 用户发起退款请求，提交退款信息
2. 服务器处理退款信息，生成完整退款记录，向第三方提交退款申请
3. 第三方验证退款请求合法性，发送退款成功的通知给服务器
4. 服务器处理通知、修改订单状态，并显示给用户

3.2.5 数据流图

3.2.5.1 顶层数据流图



3.2.5.2 0层数据流图



3.3 数据库介绍

本设计使用 Oracle 公司旗下的关系数据库 MySQL，并将在后续设计中使用 MyBatis 对数据库建立持久层框架。

3.4 数据词典

数据词典部分所涉及数据类型的表示约定如下：

数据类型符号	描述
N	数值型
V	可变长字符型
D	日期型
B	逻辑性
DM	小数型

3.4.1 数据流定义表

编号	数 据 流 名	内 部 名	组 成	流 量	备 注
L01	用 户 登 录		userid+userpassword		
L02	浏 览 商 品		Userid+footprint+goodsname+ goodsprice+goodsdetail		
L03	购 买 一 般 商 品		Goodsid+goodsprice+goodsname+userid +addressed+orderid		
L04	参 与 团 购		Userid+goodsid+groupon+grouponrule+ Currentgoodsnumber++nexthighlimit+discount		
L05	分 享 商 品		Userid+shareid+sharedid+goodsid+successtimes		
L06	预 购 商 品		Userid+deposit+goodsnumberlimit+presalstatus		
L07	购 买 组 合 商 品		Userid+compositiongoodsid+goodsid		
L08	评 论 商 品		Userid+goodsid+commented+commentcontent+ Commentarytime+status		
L09	商 品 退 换		Userid+goodsid+refundimage+refundid+orderid		

3.4.2 数据元素定义

表	属性名称	标识符	类型	长度	小 数 位	备注
User	用户标识符	userID	N	10	--	
	用户昵称	userName	V	10	--	
	账户	account	V	20	--	
	用户密码	password	V	10	--	
	用户头像	profilePicture	V	255	--	
	返点	backPoint	DM	10	2	
Goods	商品库存	goodsStock	N	10	--	
	商品名称	goodsName	V	10	--	
	商品价格	goodsPrice	N	10	--	
	商品种类	GoodsCategory	V	10	--	
	团购	Groupon	V	20	--	
	商品描述	specification	V	255	--	
	浏览图片	browseImage	N	255	--	
	细节图片	detailImage	N	255	--	
	状态	status	B	5	--	
	是否是组合商品	isCompositionGoods	B	5	--	
	预减库存	preReducingStock		20	--	
	是否有 sku	hasSKU	B	5	--	
Order	总价	totalPrice	N	10	--	
	订单时间	orderTime	D	24	--	

	配送地址	address	V	255	--	
	订单状态	orderStatus	N	5	--	
	商品清单	goodsList	V	255	--	
	是否退回	isReturned	B	5	--	
	状态	status	B	5	--	
	是否分享	isShared	B	5	--	
	快递单号	ExpressID	N	10	--	
	是否被修改	isModified	B	5	--	
	是否同步	isSync	B	5	--	
Groupon	开始时间	startTime	D	24	--	
	结束时间	endTime	D	24	--	
	状态	status	N	5	--	
	目前商品数量	currentGoodsNumber	N	20	--	
	折扣阶段	discountPhase	N	20	--	
Coupon	优惠券标识符	couponID	N	10	--	
	折扣	discount	N	10	--	
	开始时间	startDate	D	24	--	
	截止时间	endDate	D	24	--	
	使用数量限制	numberLimitForOne	N	10	--	
	所有优惠券	totalCoupons	N	10	--	
	剩余优惠券	remainingCoupons	N	10	--	
	是否被使用	isUsed	B	5	--	

	是否有效	isActive	B	5	--	
CouponItem	用户标识符	userID	N	10	--	
	优惠券标识符	couponID	N	10	--	
	已拥有	holding	N	10	--	
	已使用	used	N	10	--	
Advertisement	开始时间	startTime	D	24	--	
	截止时间	endTime	D	24	--	
	是否有效	isActive	N	5	--	
	是否默认	isDefault	N	5	--	
	跳转链接	jumpURL	V	255	--	
	广告图片	image	V	255	--	
GrouponRules	团购标识符	GrouponID	N	10	--	
	折扣	discount	N	10	--	
	下限	lowLimits	N	10	--	
GrouponOrder	状态	status	N	5	--	
	团购标识符	grouponID	N	10	--	
OrderItem	折扣价格	discountPrice	N	10	--	
	所售价格	soldPrice	N	10	--	
	数量	quantity	N	10	--	
	组合标识符	CombineID	N	10	--	
PresaleGoods	定金	deposit	N	10	--	
	定金起始时间	depositStartTime	D	24	--	

	定金结束时间	depositEndTime	D	24	--	
	最终支付	finalPayment	N	10	--	
	最终支付开始时间	finalPaymentStartTime	D	24	--	
	最终支付截止时间	finalPaymentEndTime	D	24	--	
	预售状态	presaleStatus	N	10	--	
	商品数量限制	goodsNumberLimit	N	10	--	
PresaleOrder	状态	status	N	5	--	
Refund	退款标识符	RefundID	N	10	--	
	退款费用	Fee	N	10	--	
	阶段	Phase	N	10	--	
	种类	Type	N	10	--	
	订单信息	OrderInfos	V	255	--	
	退款理由	Resons	V	255	--	
	退货快递单号	ReturnExpressID	V	10	--	
	退货图片	Image	V	255	--	
	退货理由	Reason	V	255	--	
Administrator	管理员标识符	AdministratorID	N	10	--	
	管理员权限	authority	N	10	--	
CompositionGoodsItems	组合商品标识符	compositonGoodsID	N	10	--	
	商品标识符	goodsID	N	10	--	
Brand	品牌名称	brandName	V	20	--	
	品牌描述	brandDescription	V	255	--	

	品牌图片	image	V	255	--	
Share	商品标识符	goodsID	N	10	--	
	成功分享次数	sucessTimes	N	10	--	
	状态	status	N	5	--	
	下阶段的限制	nextHighLimit	N	10	--	
	用户标识符	userID	N	10	--	
ShareRules	分享下限	lowLimits	N	10	--	
	目前返点	persentPoint	N	20	--	
	商品标识符	goodID	N	10	--	
	状态	status	N	5	--	
SharedLog	状态	status	N	5	--	
	产生时间	BirthTime	D	24	--	
	被分享者标识符	sharedID	N	10	--	
	分享者标识符	shareID	N	10	--	
OrderShareItem	状态	status	N	5	--	
Favorites	用户标识符	userID	N	10	--	
	商品标识符	goodsID	N	10	--	
FootPrints	浏览时间	browsetime	D	24	--	
	用户标识符	userID	N	10	--	
	商品标识符	goodsID	N	10	--	
CartItem	数量	quantity	N	10	--	
	用户表示股	userID	N	10	--	

	商品标识符	goodsID	N	10	--	
backPointLog	历史发送时间	spendHistoryDate	D	24	--	
	历史收取时间	recieveHistoryDate	D	24	--	
	来源	comeFrom	N	10	--	
	历史消耗返点	spendPointHistory	DM	10	2	
	历史获取返点	recivePointHistory	DM	10	2	
address	地址民称	Name	V	255	--	
	电话号码	phonenumber	V	11	--	
	行政分区	Province	V	20	--	
	地址详情	Addressdetail	V	255	--	
	用户标识符	userid	N	10	--	
log	创建标识符	Createid	N	10	--	
	创建时间	Date	D	24	--	
	状态	status	N	5	--	
	内容	content	V	255	--	
goodscategory	商品种类标识符	categoryid	N	10	--	
	种类名称	Categoryname	V	10	--	
	父种类	fathercategory	V	10	--	
goodscomment	商品标识符	goodsid	N	10	--	
	评论时间	commentarytime	D	24	--	
	状态	status	N	5	--	
	内容	content	V	255	--	

specialtopcomment	专题讨论标识符	specialtopicid	N	10	--	
	评论时间	commentarytime	D	24	--	
	状态	status	N	5	--	
	内容	content	V	255	--	
specialtopic	专题标识符	specialtopicid	N	10	--	
	专题题目	title	N	10	--	
	专题图片	image	V	255	--	
sku	销售属性标识符	skuid	N	10	--	
	规格一	parameter1	V	20	--	
	规格二	parameter2	V	20	--	
	库存	stock	N	10	--	
	价格	Price	N	10	--	
	商品标识符	Goodsid	N	10	--	
	顾客标识符	customerid	N	10	--	
	预扣库存	prereducingstock	N	20	--	

3.5 数据采集

1. 与购买有关的数据采集
数据来源：商品信息，用户信息来源于用户的输入。
数据输入所用的媒体和硬设备：手机、平板
输出数据接受者：订单
输出数据的形式和设备：形式：多张表 设备：显示器
输入的承担者：用户
2. 与评论有关的数据采集
数据来源：评论详情，评论打分等均来源于用户的输入。
数据输入所用的媒体和硬设备：手机、平板
输出数据接受者：商品

输出数据的形式和设备：形式：表 设备：显示器

输入的承担者：用户

3. 与团购有关的的数据采集

数据来源：团购规则由管理员指定，团购由用户所下单商品构成，及来源于用户和管理员的共同输入。

数据输入所用的媒体和硬设备：手机、平板

输出数据接受者：用户和管理员

输出数据的形式和设备：形式：表 设备：显示器

4. 与优惠券的的数据采集

数据来源：管理员制定优惠券使用规则，用户领取并使用优惠券，在购买商品时进行应用

数据输入所用的媒体和硬设备：手机、平板

输出数据接受者：商品

输出数据的形式和设备：形式：表 设备：显示器

5. 与分享有关的数据采集

数据来源：发送分享链接的分享者，以及制定分享规则的管理员

数据输入所用媒体和硬设备：手机、平板

输出数据接收者：被分享者

输出数据的形式和设备：形式：链接 设备：显示器

4. 功能需求

4.1 功能划分

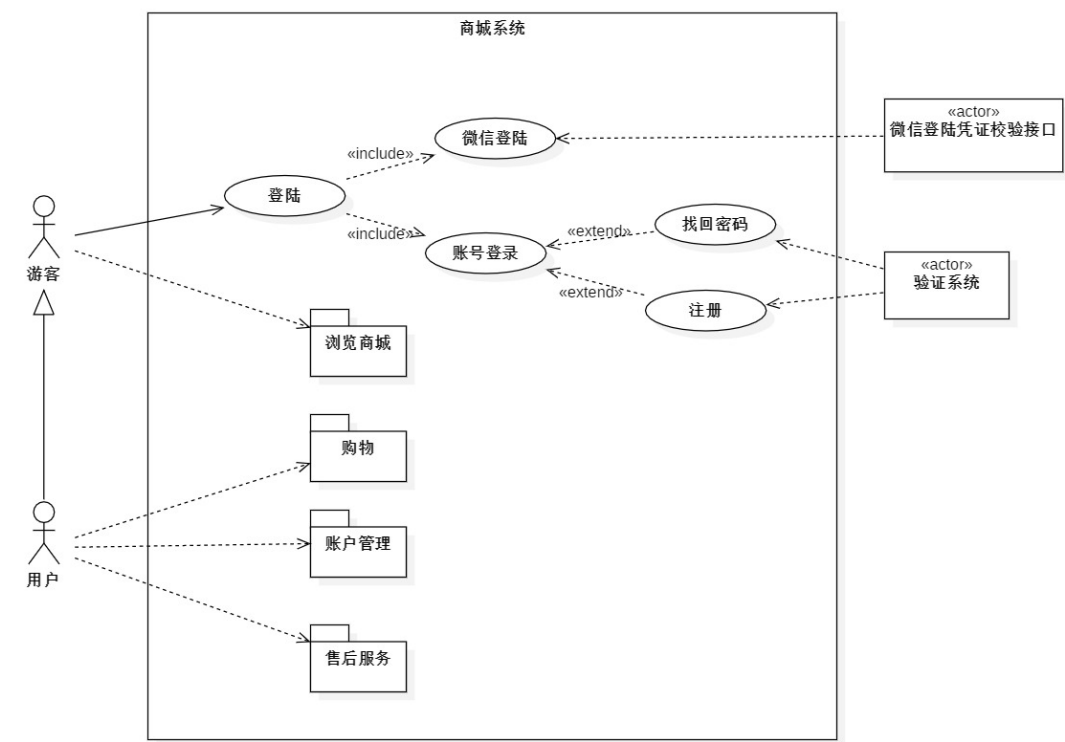
4.1.1 用户目标分析

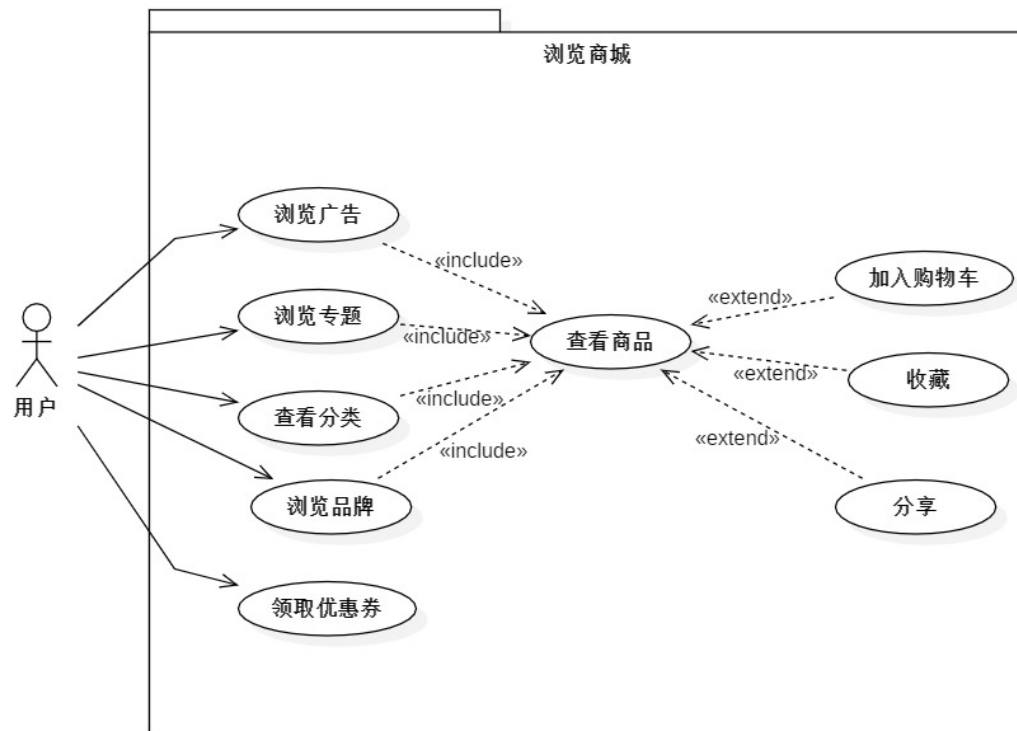
主要参与者	目标
用户	主要分为浏览商城、购买、信息管理、售后服务和其他功能： （1）购物模块主要达到的目标为：浏览商品、下订单、购物车管理。 （2）账户管理主要达到的目标为：优惠券管理、收藏夹管理、订单管理、查看足迹、地址管理。 （3）售后服务主要达到的目标为：退货、换货、联系客服、意见反馈。 （4）其他功能模块主要达到的目标为：登陆、密码找回、注册、退出。
管理员	主要分为用户管理、商场管理、商品管理、推广管理、系统管理、配置管理、统计报表和其他功能： （1）用户管理模块主要达到的目标为：查看和导出用户、用户的收货地址、收藏、足迹、搜索历史和意见反馈。 （2）商场管理模块主要达到的目标为：查看行政区域，查看、添加和导出品牌制造商、订单，查看、添加、编辑和删除商品类目、通用问题、关键词。 （3）商品管理模块主要达到的目标为：添加和导出商品列表，编辑、删除商品，查看商品详情，管理商品上架，查看、审核、回复、删除商品评论。

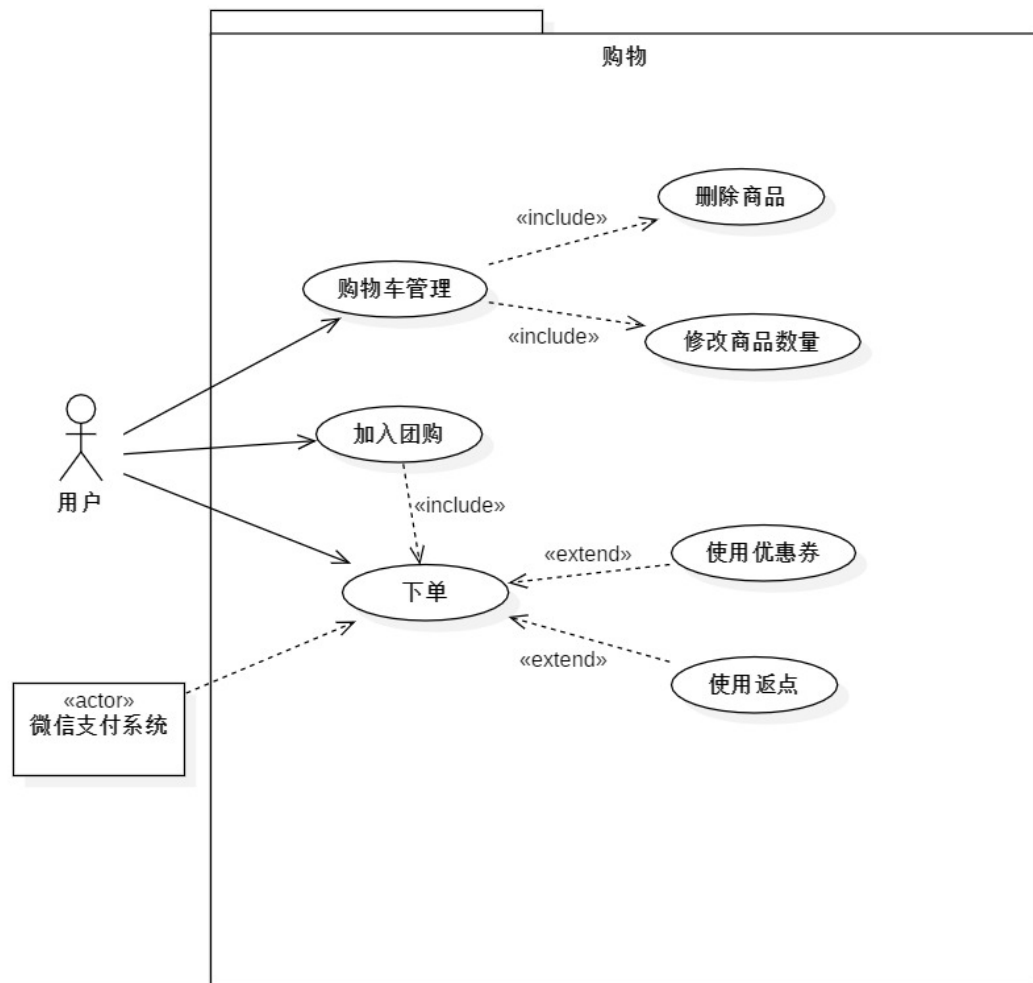
	<p>(4) 推广管理模块主要达到的目标为：添加、编辑广告，设置广告的上线时间和下线时间，将广告下线，优惠券、专题、团购规则、团购活动的管理。</p> <p>(5) 系统管理模块主要达到的目标为：查看管理员、操作日志，角色的授权、对象存储。</p> <p>(6) 配置管理模块主要达到的目标为：配置商场、运费、订单、小程序信息。</p> <p>(7) 统计报表模块主要的功能为：统计用户、订单、商品。</p> <p>(8) 其他功能模块主要达到的目标为：外链管理、管理员登陆。</p>
--	--

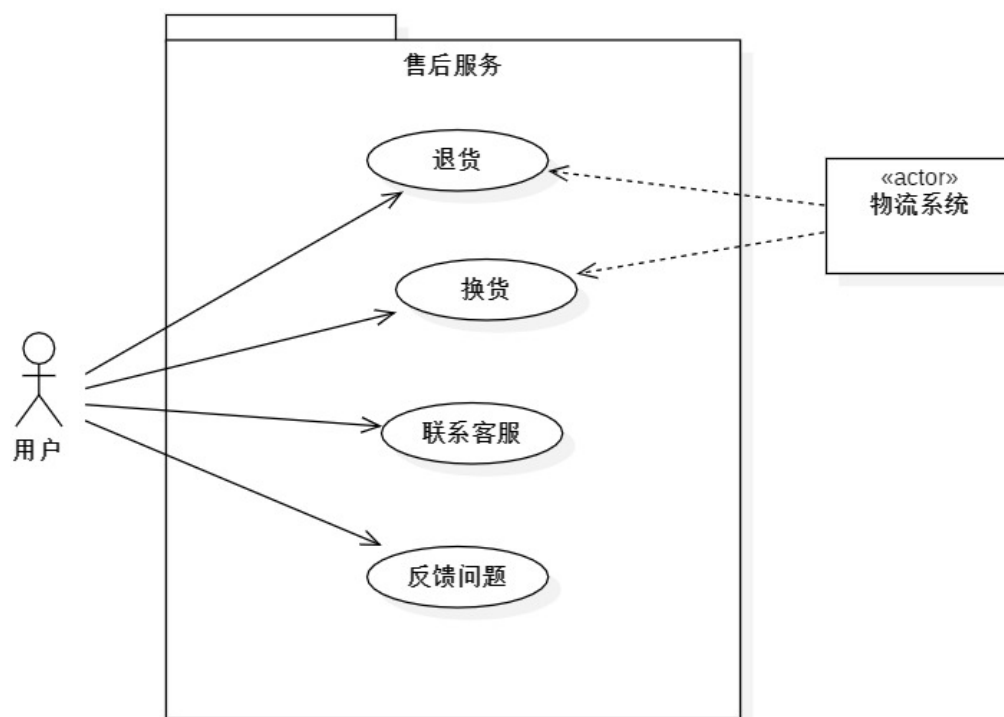
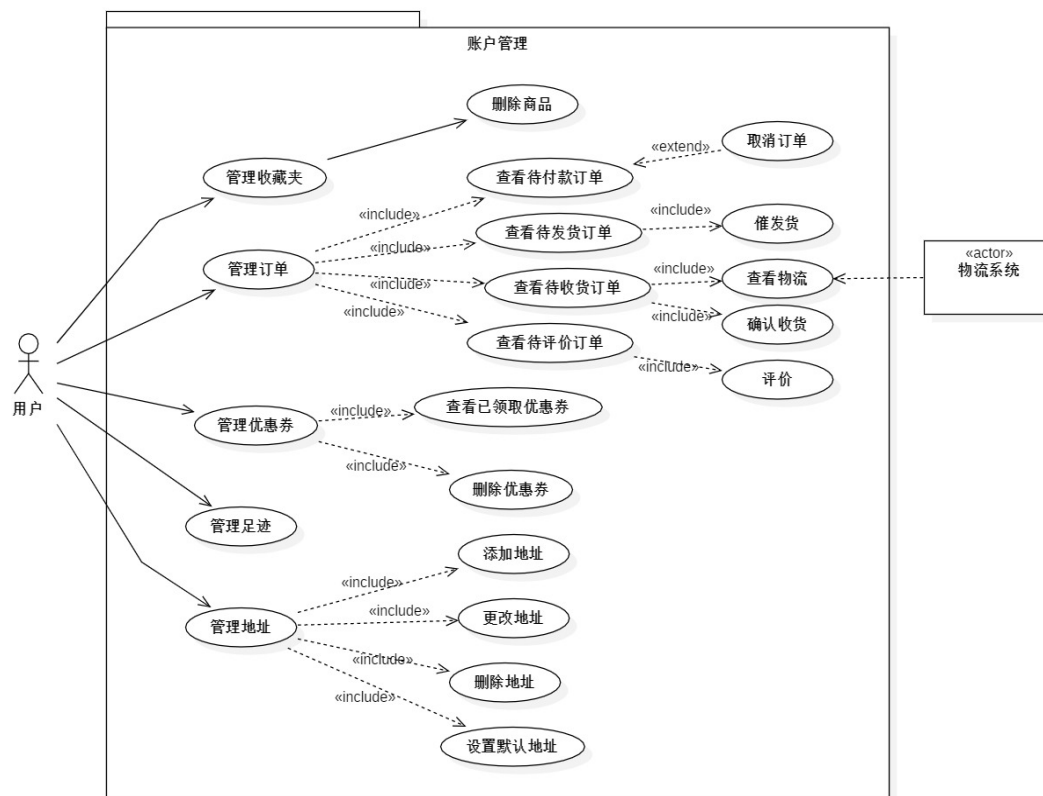
4.1.2 用例图

4.1.2.1 用户用例图

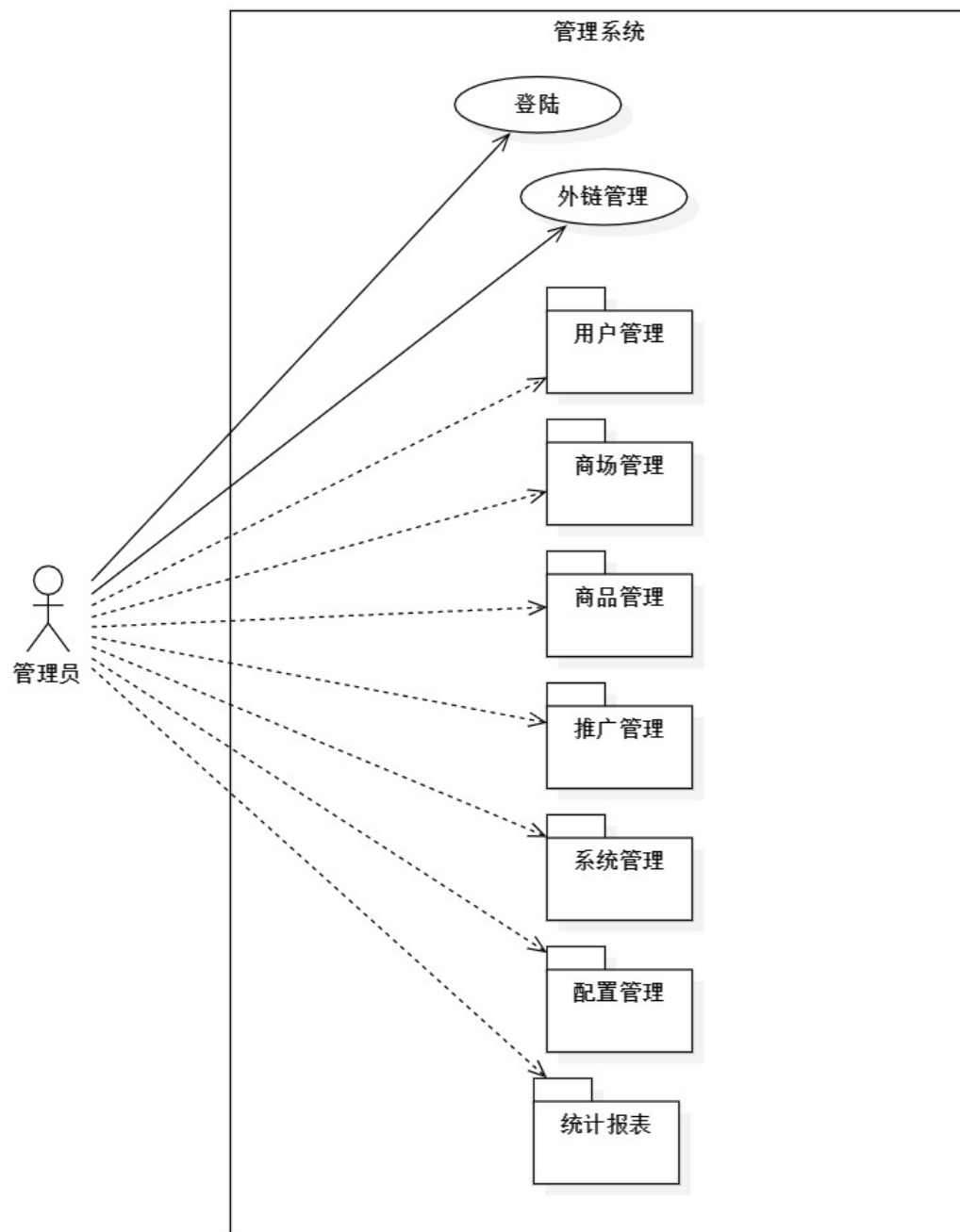


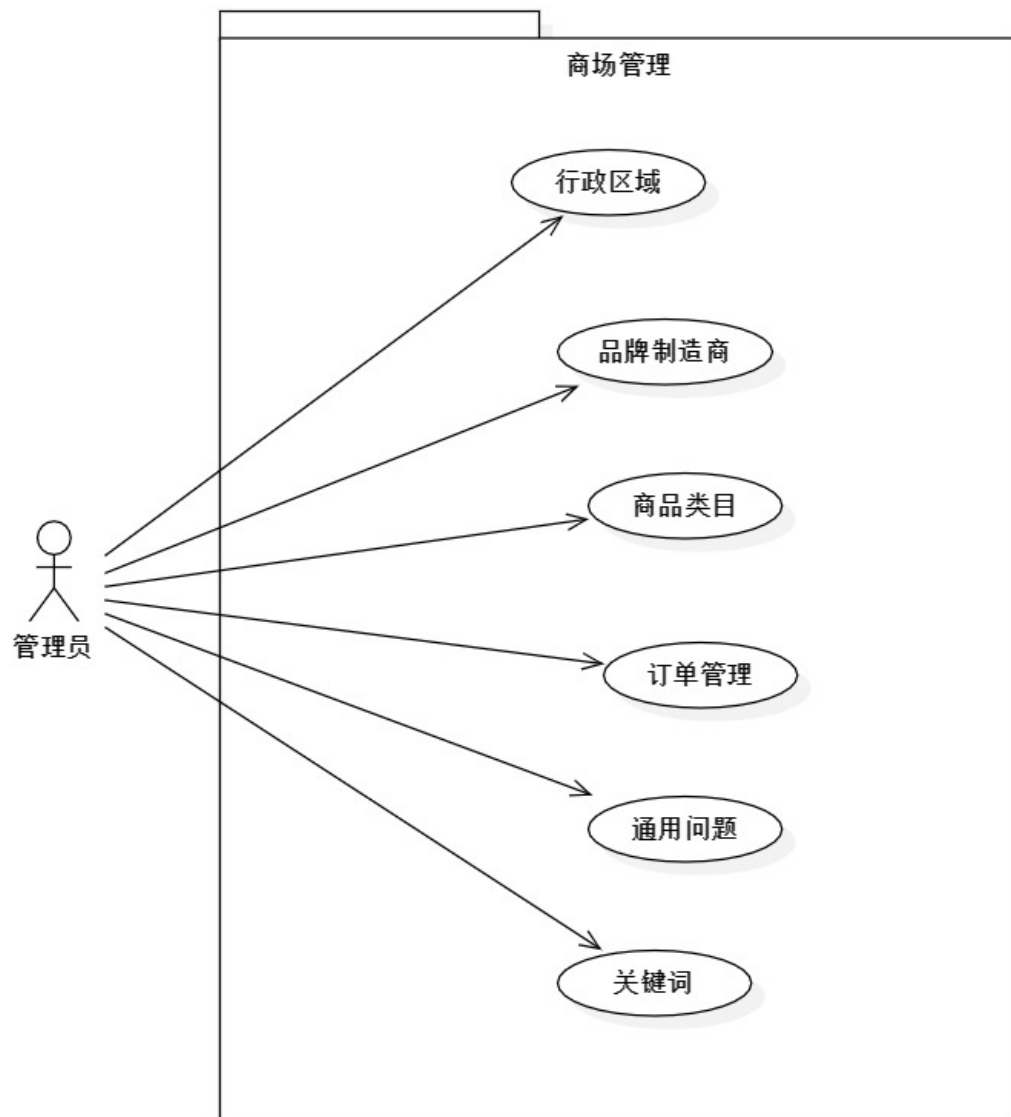


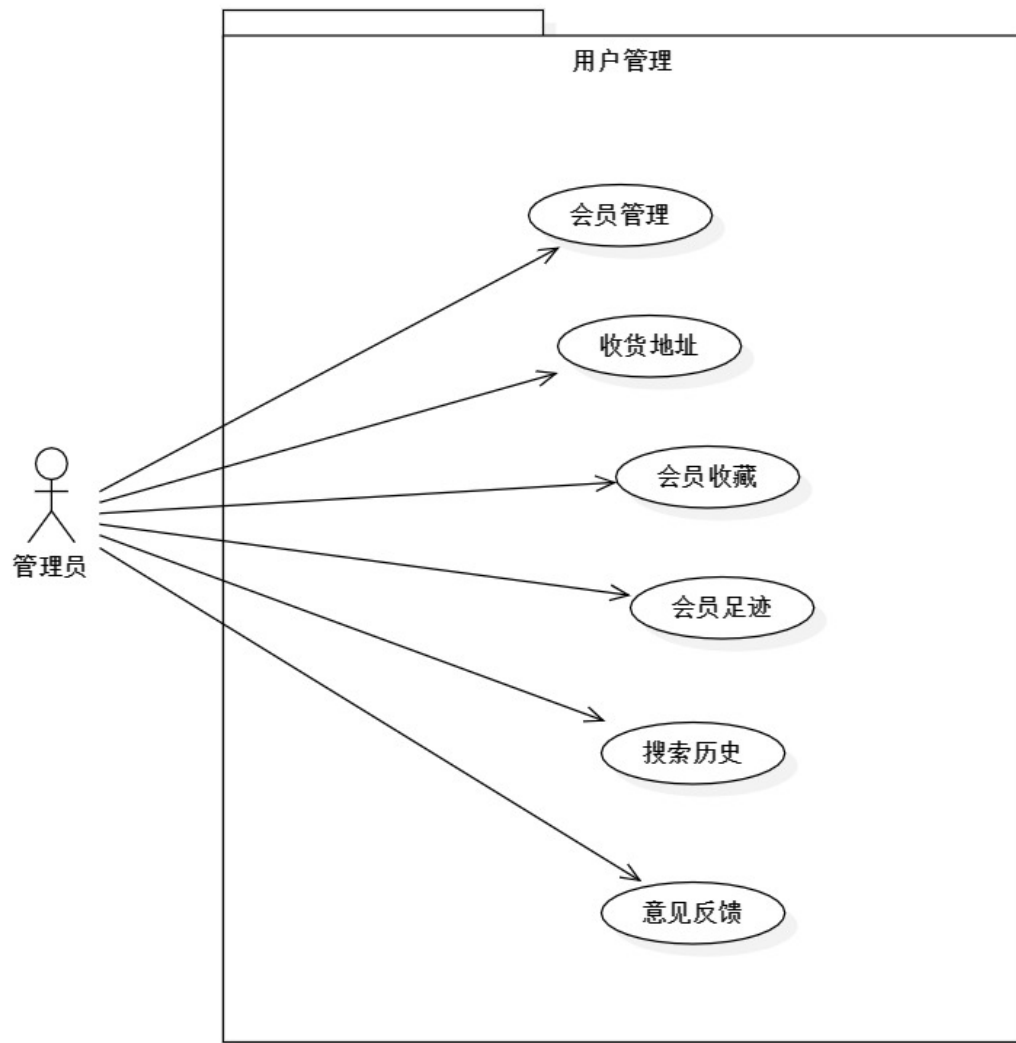


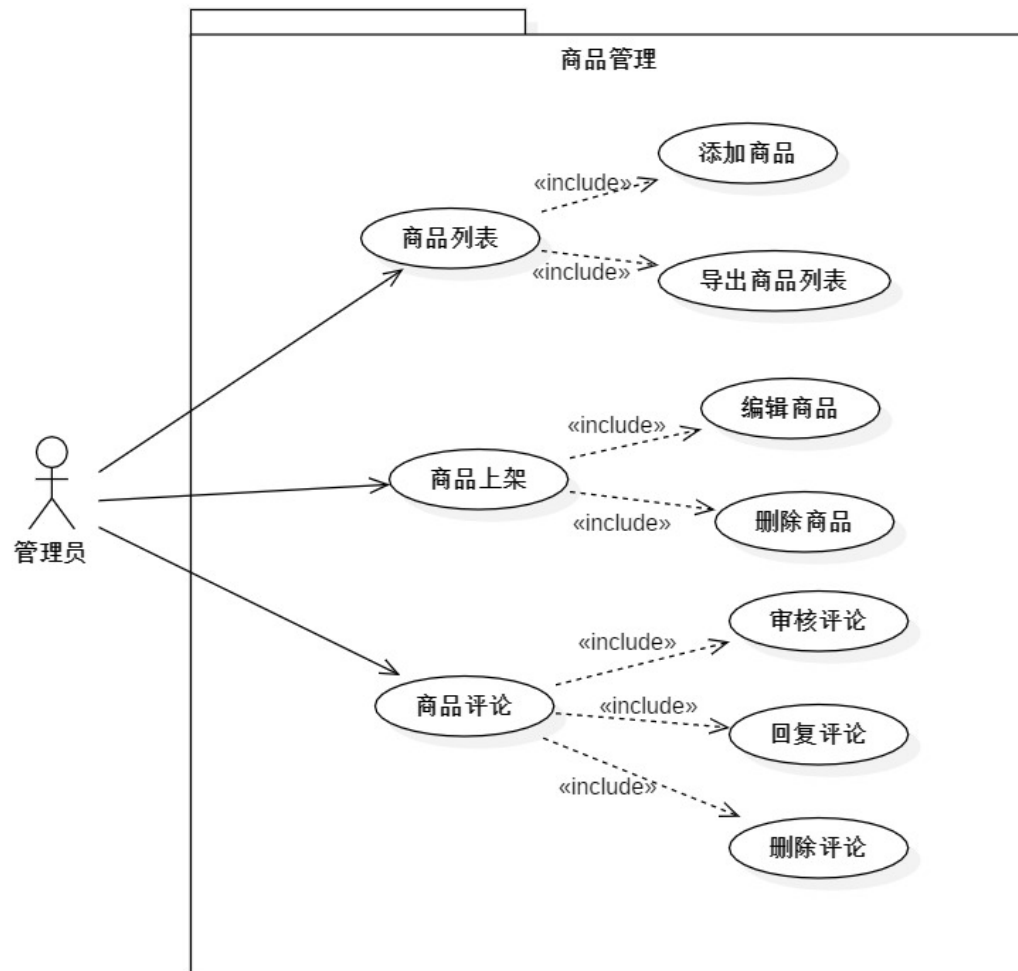


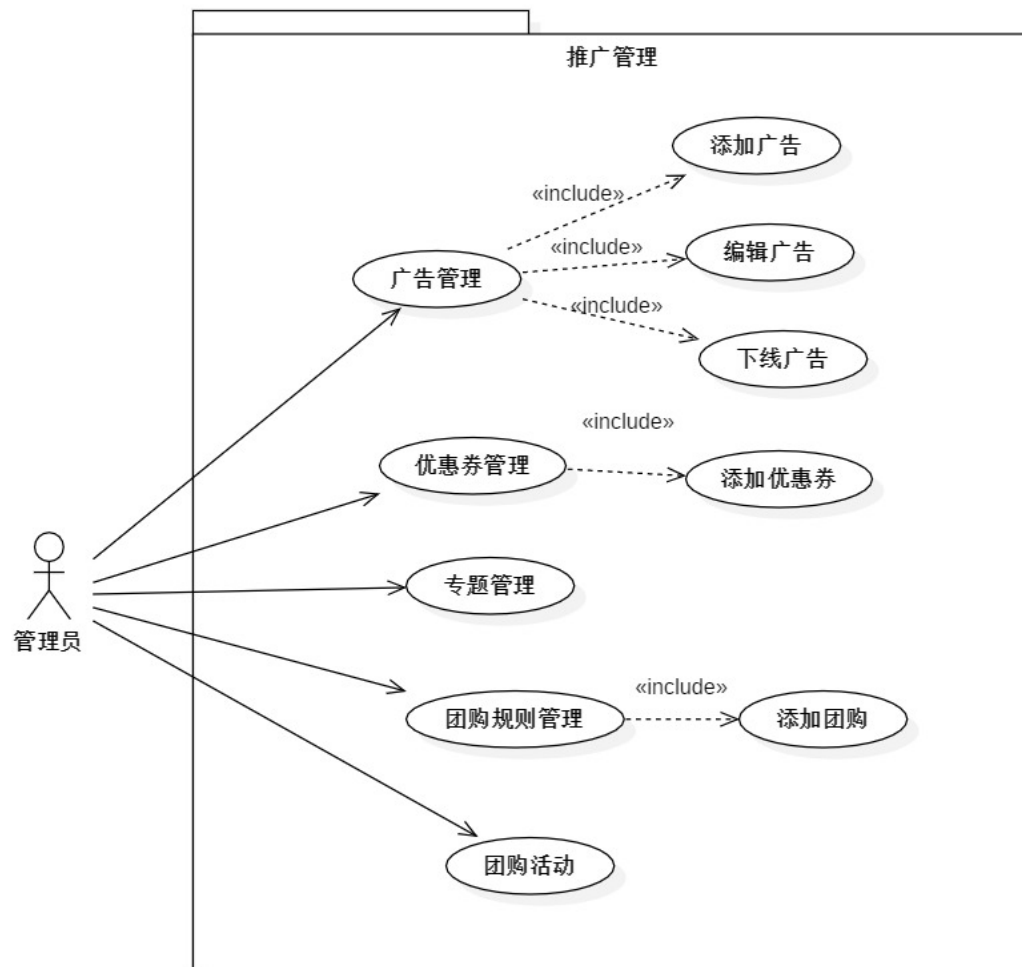
4.1.2.2 管理员用例图

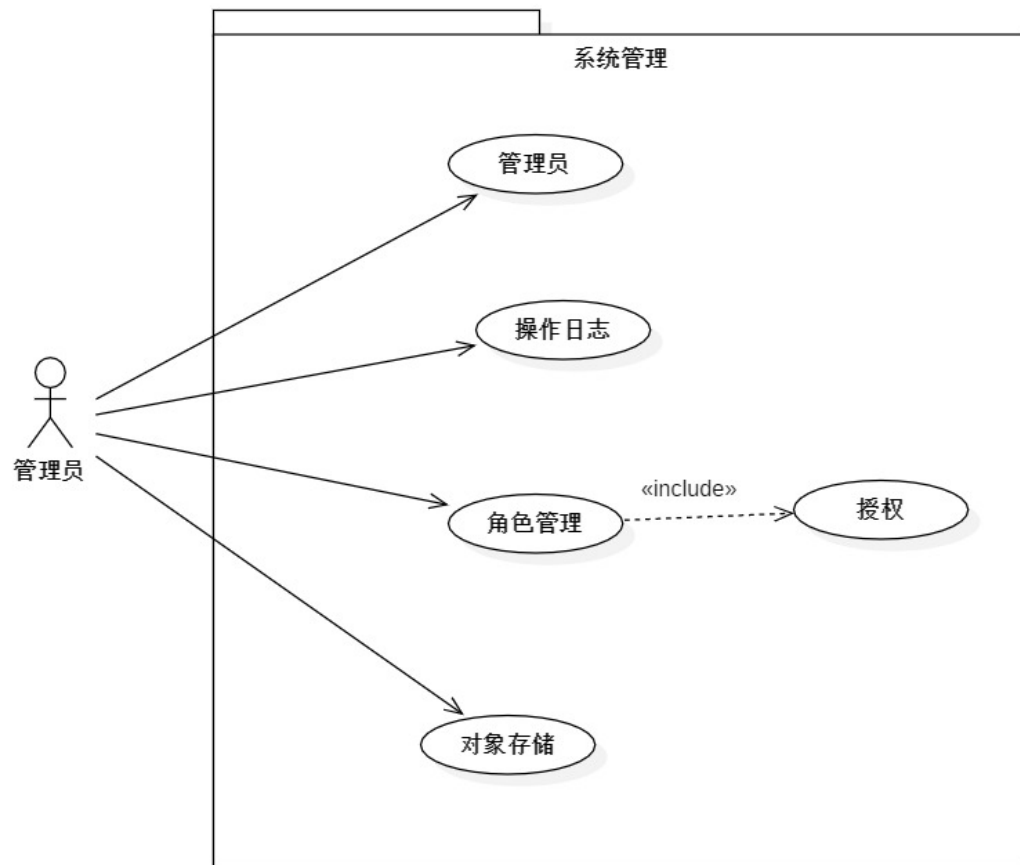


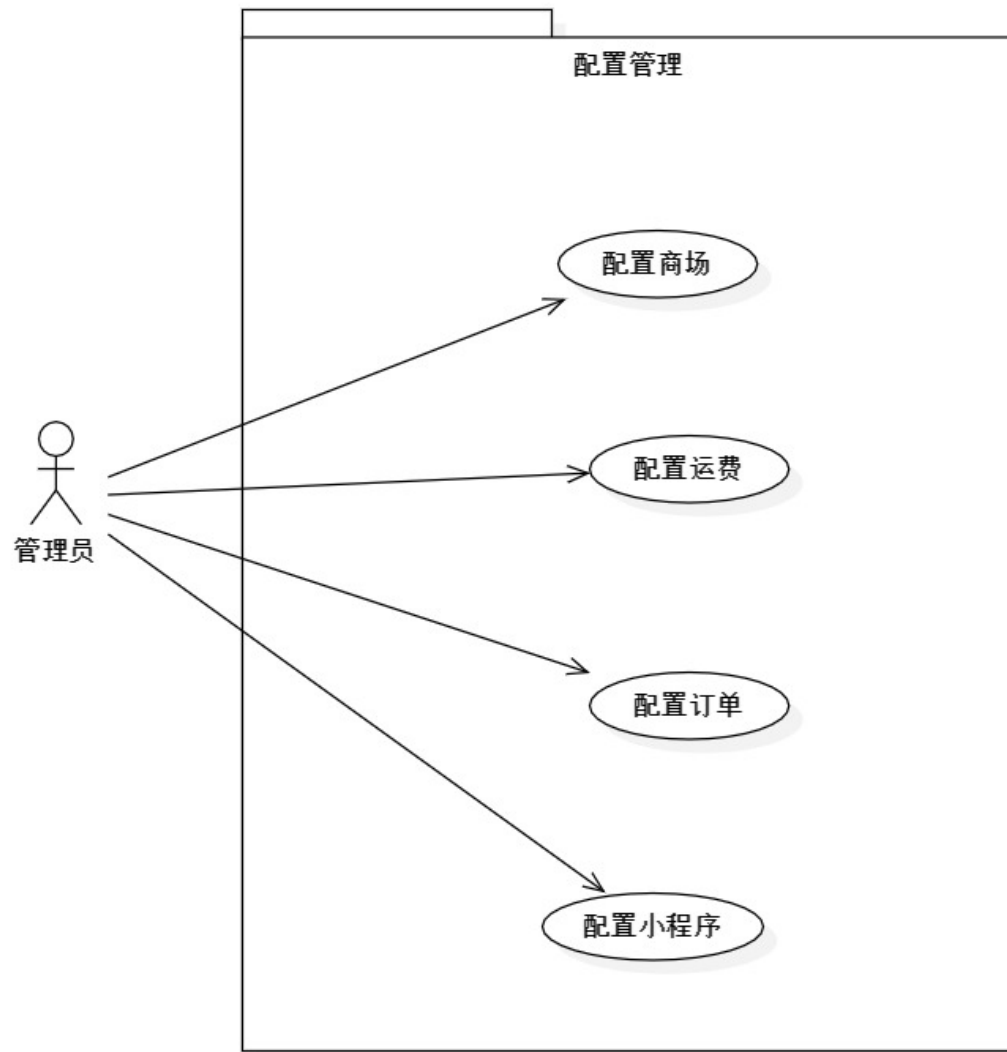


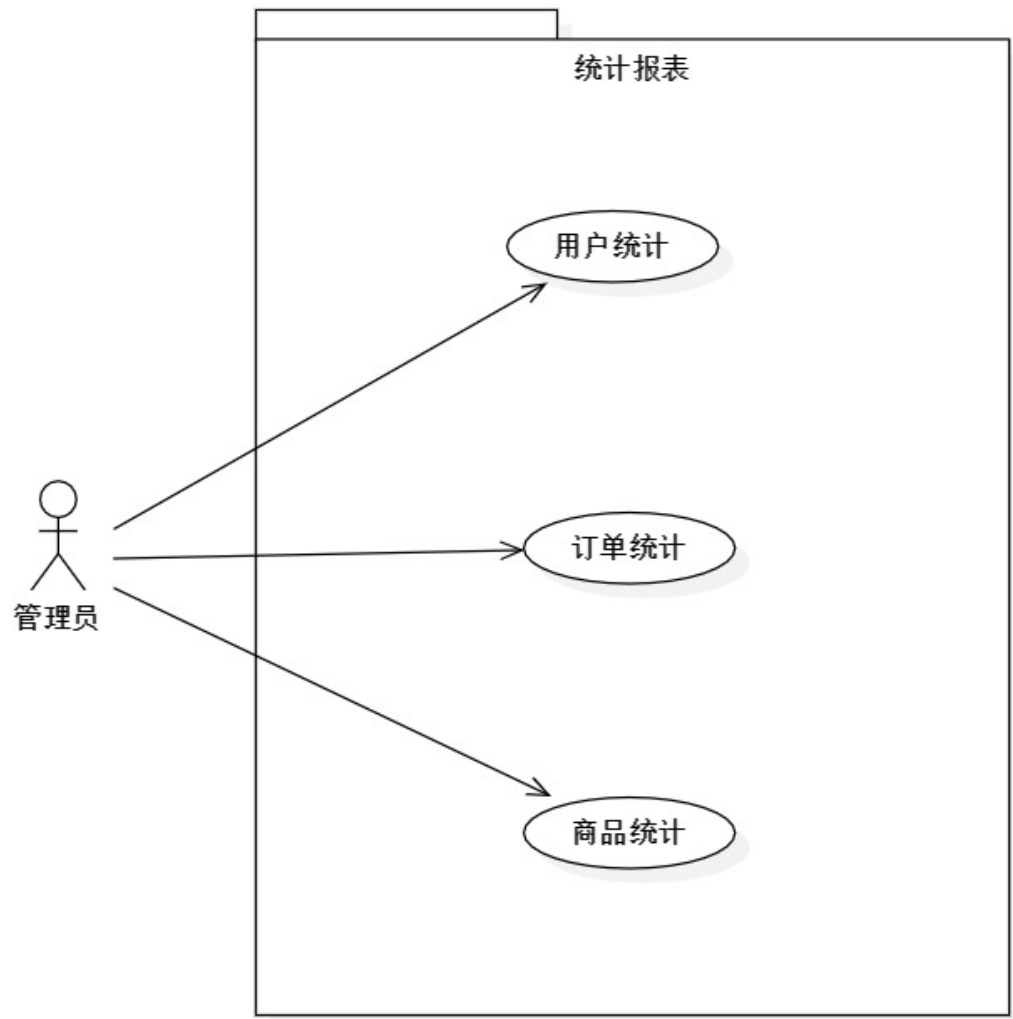












4.2 功能描述

4.2.1 用户用例分析

用例一：登陆

主要流程：

用户进入登陆页，选择微信登陆或者账号登陆方式，进入商城主页。

分支流程：

1. 用户选择账号登陆，用户输入账号密码；
2. 用户选择忘记密码，进入密码重置页面，填写验证信息；
3. 用户选择注册账号，用户进入注册页面，填写注册信息。

用例二：浏览广告

主要流程：

用户可以滑动广告栏查看广告，并进入广告页面查看商品。

用例三：浏览专题

主要流程：

用户进入某个专题页面，可以浏览专题描述，专题下的商品，专题留言，并可以在留言区进行留言。

用例四：查看分类

主要流程：

用户可以在分类板块查看一级分类标签，通过选择一级标签可以进入二级分类标签，并浏览二级分类标签下的商品。

用例五：浏览品牌

主要流程：

用户可以选择进入任一品牌制造商页面，并浏览品牌制造商下的商品。

用例六：领取优惠券

主要流程：

用户进入首页可以领取系统发放的优惠券。

分支流程：

1. 若达到单人领取优惠券上限，则领取失败；
2. 若达到优惠券领取张数上限，则领取失败。

用例七：查看商品

主要流程：

用户进入商品页面，可以查看商品描述、商品的评论、商品的规格。

分支流程：

1. 用户可以选择将商品加入购物车；
2. 用户可以选择将商品加入收藏夹；
3. 用户可以选择加商品分享给好友，好友通过链接在七天之内购买商品且不退货，用户可以获得返点。

用例八：购物车管理

主要流程：

用户在购物车页面可以选择删除商品，修改商品数量和属性，并在购物车选择商品进行结算。

用例九：团购

主要流程：

用户选择团购商品，按照正常商品的价格下单，在团购规定时间内达到团购数量则完成团购，并按照团购等级规则，原渠道收回团购优惠的金额。

分支流程：

在团购规定时间内没有达到最低团购数量，则团购失败，订单自动取消，用户将收回支付金额。

用例十：下单

主要流程：

用户选择购买商品或者通过购物车结算进入下单页面，填写收货地址，可以查看满足商品使用条件的优惠券，选择使用一张优惠券，并可以选择使用用户拥有的返点数抵扣商品支付金额，提交订单，完成支付。

分支流程：

1. 用户提交订单后未完成付款，订单被挂起，并于 30 分钟后自动取消订单；
2. 用户提交订单后商品库存不足，则无法进行支付，订单不能提交；
3. 用户在支付过程中账户余额不足，则付款失败，订单失败作废。

用例十一：管理收藏夹

主要流程：

用户进入收藏夹，可以浏览收藏夹内商品，也可以选择将商品删除。

用例十二：管理订单

主要流程：

用户可以查看自己的订单，包括待付款订单，待发货订单，待收货订单和待评价订单

分支流程：

1. 用户可以在待付款订单中选择取消订单；
2. 用户可以在待发货订单中催商家发货；
3. 用户可以查看待收货订单的物流信息，也可以确认收货；
4. 用户可以对待评价订单进行评价，评价发布完成后等待管理员审核，审核成功后才能在页面上显示。

用例十三：管理优惠券

主要流程：

用户可以查看自己已领取的优惠券，删除优惠券。

用例十四：管理足迹

主要流程：

用户可以查看过去几天中每天的商品浏览足迹。

用例十五：管理地址

主要流程：

用户可以管理自己的收货地址，包括添加一个新地址，更改现有的地址信息，删除一条收货地址，设置某个收货地址为默认地址。

用例十六：退货

主要流程：

用户可以提交退货申请，填写退货理由，并等待管理员审核，在审核完成后用户在退货订单中填写物流信息，管理员确认收货后将在七天之内将用户申请退款金额返回给用户。

分支流程：

1. 管理员未审核退货申请则用户重新提交；
2. 管理员未收获成功则退货失败；

用例十七：换货

主要流程：

用户可以提交换货申请，选择要换货的商品，填写换货理由，并等待管理员审核，审核完成后用户在换货订单中填写物流信息，管理员确认收货后必须在七天之内重新发货，并填写物流信息，用户重新收货后确认商品信息后则换货成功。

分支流程：

1. 用户提交换货申请后管理员未审核或拒绝换货则换货失败，用户可以再次提交换货申请；
2. 用户重新收货后检查商品时发现商品有误则换货失败，再次提交换货申请。

用例十八：联系客服

主要流程：

用户可以在购买前跟客户咨询商品信息、购买建议、协商，也可以在收货后联系客服询问关于商品使用的注意点，咨询退换货流程。

用例十九：反馈问题

主要流程：

用户可以对商品相关、功能异常、优化建议或者其他进行反馈，填写对商城、商品、服务的个人建议，并填写用户的手机号码。

4.2.2 管理员用例分析

用例一：会员管理

主要流程：

管理员可以通过手机号或者用户名来查找用户，并查看用户的相关信息。

用例二：收货地址

主要流程：

管理员可以查看或导出用户的收货地址。

用例三：会员收藏

主要流程：

管理员可以查看或导出用户的商品收藏信息。

用例四：会员足迹

主要流程：

管理员可以查看或导出用户的商品浏览记录。

用例五：搜索历史

主要流程：

管理员可以查看或导出用户的商品搜索历史。

用例六：意见反馈

主要流程：

管理员可以查看或导出用户的意见反馈，从而对商城进行调整。

用例七：行政区域

主要流程：

管理员可以查看全国的所有行政区区域编码，所有省级行政区下的所有市级行政区、区县级行政区的区域编码。

用例八：品牌制造商

主要流程：

管理员可以根据品牌商 ID 或者品牌商名称查找对应的品牌制造商，也可以添加、编辑、删除或者导出品牌商信息。

用例九：商品类目

主要流程：

管理员可以查看所有一级商品类目列表，以及一级目录下的二级商品类目，也可以添加、编辑、删除一级类目、二级类目。

用例十：订单管理

主要流程：

管理员可以根据用户 ID、订单编号、订单状态查找所有订单列表，还可以查看订单的详细信息，导出订单列表、打印单个订单的详细信息。

用例十一：通用问题

主要流程：

管理员可以输入问题内容可以查询相关通用问题，可以添加、编辑、删除通用问题信息，支持导出列表。 关键词：输入关键字或者跳转链接可以查询相应的关键词标签，可以添加、编辑、删除关键词，支持导出关键词列表。

用例十二：商品列表

主要流程：

管理员可以根据商品编号、商品名称查询商品信息，也可以添加商品，对商品进行上架。还可以查看商品的详细信息，编辑修改商品的信息，删除商品，导出商品信息。

用例十三：商品上架

主要流程：

管理员可以填写新的商品信息，将商品上架。

用例十四：商品评论

主要流程：

管理员可以通过用户 ID 查找到用户的所有的评论，以及通过商品 ID 查找到该商品的所有评论。对评论进行审核，还可以回复评论，删除不合法的评论。

用例十五：广告管理

主要流程：

管理员可以将广告上线，设定广告的上线时间、下线时间，在广告上线后可以编辑广告，暂停广告的播放，也可以直接下线广告。

用例十六：优惠券管理

主要流程：

管理员可以向商场发放优惠券，包括设置优惠券的数量，优惠券的抵扣金额，有效期限，单人领取上限，也可以删除优惠券。

用例十七：专题管理

主要流程：

管理员可以进行专题的推广（专题是以某一个主题进行商品推荐的软文），可以编辑专题，查看专题的访问数量。

用例十八：团购管理

主要流程：

管理员可以发起团购活动，定制团购活动的开始时间、结束时间、团购的具体规则，可以查看团购活动的订单，在团购活动结束后可以进行删除。

用例十九：角色管理

主要流程：

管理员拥有不同的权限，超级管理员（最高权限的管理员）能够设置其他管理员的权限，控制其他管理员只能操作特定的功能（例如：可以设置管理员为推广管理员，负责首页的广告、专题等活动的推广；也可以设置管理员为商场管理员，负责商品的商家和销售管理）。

用例二十：操作日志

主要流程：

管理员（拥有最高权限）能够查看一段时间内系统中商城管理员、推广管理员的操作日志，保证管理后台系统安全运行。

用例二十一：商场配置

主要流程：

管理员可以设置、修改商场名称、商场地址、联系电话、联系 qq 等。

用例二十二：运费配置

主要流程：

管理员可以设置、修改运费的价格，以及包邮所需的最低消费价格。

用例二十三：订单配置

主要流程：

管理员可以设置用户下单后未付款系统的等待时间，否则订单就会自动取消。也可以设置订单发货后用户未确认收货时系统的等待时间，否则订单就会自动确认收货；及确认收货后用户未评价商品，系统的等待时间，否则系统默认好评。

用例二十四：小程序配置

主要流程：

管理员可以设置、修改首页中各栏目显示的数量，还可以开启和关闭商城中的商品分享功能。

用例二十五：统计用户

主要流程：

管理员可以查看各个时间段用户的增长情况。

用例二十六：统计订单

主要流程：

管理员可以查看订单量、下单用户、订单总额和客单价四个统计量随时间的关系变化图。

用例二十七：统计商品

主要流程：

管理员可以查看订单量、下单货物数量和下单货品总额三个统计量随时间的关系变化图。

5. 性能需求

5.1 数据精确度

为了提高系统安全性以及保证系统正常运行,我们针对系统中所有需要人手动输入的数据都作出界定,这其中包括有用户输入的以及管理员输入的数据。

字段名称	储存位置	数据精确度	备注
优惠券有效期开始时间	CouponTopParent.startDate	格式为****年**月**日**时,精确至小时,采用24小时制,如2019年11月2日13时	时间不得晚于CouponTopParent.endDate
优惠券有效期结束时间	CouponTopParent.endDate	格式为****年**月**日**时,精确至小时,采用24小时制,如2019年11月2日13时	时间不得早于CouponTopParent.startDate
优惠券总发行量	CouponTopParent.totalCoupons	大于0的正整数	
优惠券折扣	CouponTopParent.discount	大于0小于100的整数	存储的是百分比,如该数据为10,则表示返点额度为10%

详细地址	Address.addressDetail	50 个字符的字符串，可为各语言文字与数字的组合	
收件人姓名	Address.name	10 个字符的字符串，可为各语言文字与数字的组合	
收件人手机号	Address.phoneNumber	11 个字符的字符串，只能由数字组成	
用户名	User.userName	10 个字符的字符串，可为各语言文字与数字的组合	
用户账户	User.account	20 个字符的字符串，只能是英文字母与数字与特殊字符的组合	
用户密码	User.password	20 个字符的字符串，只能是英文字母与数字与特殊字符的组合，必须同时含有英文字母和数字	
用户头像	User.profilePicture	不大于 1M 的 JPEG、PNG 格式图片	将会裁剪成正方形进行保存
商品类别名称	GoodsCategory.categoryName	10 个字符的字符串，只能由汉字组成	
下一级	ShareRules.lowLimits	大于 0 的正整数	不能比当前级别的分享级别下限少

分享级别下限			
当前分享级别返点百分比	ShareRules.rebatePercentage	大于 0 小于 100 的整数	存储的是百分比，如该数据为 10，则表示返点额度为 10%
预售商品定金	PresaleGoods.deposit	大于等于 0 的浮点数，精确至小数点后二位	
预售商品尾款	PresaleGoods.finalPayment	大于 0 的浮点数，精确至小数点后二位	
定金支付开始时间	PresaleGoods.depositStartTime	格式为****年**月**日**时，精确至小时，采用 24 小时制，，如 2019 年 11 月 2 日 13 时	时间不得晚于 PresaleGoods.depositEndTime、PresaleGoods.finalPaymentEndTime、PresaleGoods.finalPaymentStartTime
定金支付结束时	PresaleGoods.depositEndTime	格式为****年**月**日**时，精确至小时，采用 24 小时制，，如 2019 年 11 月 2 日 13 时	时间不得早于 PresaleGoods.depositStartTime、时间不得晚于 PresaleGoods.finalPaymentEndTime、PresaleGoods.finalPayment

3-1 小组需求规格说明书

间			tStartTime
尾款支付开始时间	PresaleGoods.finalPaymentStartTime	格式为****年**月**日**时，精确至小时，采用 24 小时制，，如 2019 年 11 月 2 日 13 时	时间不得早于 PresaleGoods.depositStartTime、PresaleGoods.depositEndTime，时间不得晚于 PresaleGoods.finalPaymentEndTime
尾款支付结束时间	PresaleGoods.finalPaymentEndTime	格式为****年**月**日**时，精确至小时，采用 24 小时制，，如 2019 年 11 月 2 日 13 时	时间不得晚于 PresaleGoods.depositStartTime、PresaleGoods.depositEndTime、PresaleGoods.finalPaymentStartTime
预售数量限制	PresaleGoods.goodsNumberLimit	大于 0 的正整数	
商品库存	Goods.goodsStock	大于 0 的正整数	数量不得少于 Goods.preReducingStock
商品名称	Goods.goodsName	50 个字符的字符串，可为各语言文字与数字的组合	
商品价格	Goods.goodsPrice	大于等于 0 的浮点数，精确至小数点后二位	
商品细节	Goods.details	100 个字符的字符串，可为各语言文字与数字的组合	
商品详情图	Goods.detailImage	JPEG、PNG 格式图片	

3-1 小组需求规格说明书

片			
商品浏览图片	Goods. browseImage	不大于 2M 的 JPEG、PNG 格式图片	
评论内容	Comment. content	100 个字符的字符串，可为各语言文字与数字的组合	
专题标题	SpecialTopic. title	50 个字符的字符串，可为各语言文字与数字的组合	
专题内容图片	SpecialTopic. image	JPEG、PNG 格式图片	
退款金额	Refund. refundFee	大于等于 0 的浮点数，精确至小数点后二位	
退款描述	Refund. userDescription	100 个字符的字符串，可为各语言文字与数字的组合	
退款详情图片	Refund. usrImage	JPEG、PNG 格式图片	
退货回寄物流单	Refund. refundExpressID	50 个字符的字符串，只能是英文字母与数字的组合	
规格	SKU. parameter1 SKU. parameter2	20 个字符的字符串，可为各语	

参数		言文字与数字的组合	
规格价格	SKU. price	大于等于 0 的浮点数，精确至小数点后二位	
规格库存	SKU. stock	大于 0 的正整数	
自定义规格 ID	SKU. customerID	20 个字符的字符串，只能是英文字母与数字的组合	
品牌名称	Brand. brandName	10 个字符的字符串，只能是英文字母与数字的组合	
品牌描述	Brand. brandDescription	50 个字符的字符串，只能是英文字母与数字的组合	
品牌图片	Brand. image	JPEG、PNG 格式图片	
广告上线时间	Advertisement. startTime	格式为****年**月**日**时，精确至小时，采用 24 小时制，，如 2019 年 11 月 2 日 13 时	时间不能晚于 Advertisement. endTime
广告下线时间	Advertisement. endTime	格式为****年**月**日**时，精确至小时，采用 24 小时制，，如 2019 年 11 月 2 日 13 时	时间不能早于 Advertisement. startTime
广告跳转	Advertisement. jumpURL	长度为 100 的字符串，只能由数字和英文字母组成	

UR L			
广 告 图 片	Advertisement.image	JPEG、PNG 格式 图片	
团 购 开 始 时 间	Goupon.startTime	格式为****年 **月**日**时， 精确至小时，采 用 24 小时制， 如 2019 年 11 月 2 日 13 时	时间不能晚于 Goupon.endTime
团 购 结 束 时 间	Goupon.endTime	格式为****年 **月**日**时， 精确至小时，采 用 24 小时制， 如 2019 年 11 月 2 日 13 时	时间不能早于 Goupon.startTime
团 购 折 扣	GouponRules.discount	大于 0 小于 100 的整数	存储的是百分比，如该数据 为 10，则表示返点额度为 10%
团 购 下 一 优 惠 级 别 数 量 下 限	GouponRules.lowLimits	大于 0 的正整数	不能比当前级别的分享级别 下限少

以上表格所展示的数据精确度数据仅包含用户及管理员需要手动输入的数据，在数据库中会有更多的约束，这里不一一列出，详细可见数据字典。

5.2 时间特性

1. 订单创建之后，必须 30 分钟以内完成支付，30 分钟内未完成支付则自动取消订单；这样可以防止恶意刷单导致占用过多数据库资源。
2. 不考虑故障因素的情况下，上传资料的时间与网速呈正相关。本系统有若干功能需要上传较大规格的图片。理想情况下上传命令的响应速度以及上传结果的反馈信息应该与一

般响应时间相近，大约为 1-2 秒以内。

3. 在热门商品的预售以及热门优惠券的发放的情景下，可能会同时有数量庞大的用户同时访问系统造成并发问题；我们系统应该要支持同时 10,000 人对系统某一个功能进行访问，应在这个限度内保证性能不受影响。

4. 用户在前端浏览商品时图片的加载是本系统中人机交互体验的重点部分。预览图的加载和预加载速度是越快越好，希望系统能在 1-2 秒以内完成浏览页面所需的加载。

5.3 适应性

1. 该系统应该支持 web 端。针对此项，要求同时兼容移动端、pad 端以及 pc 端的 web 端；目标能在 IE10、IE edge16、Chrome、Safari、FireFox 等不同平台上稳定运行，这要求系统要有良好的可移植性以及可维护性。

2. 该系统应该支持微信小程序端。

3. 在软件方面，响应时间，更新处理时间，数据传送时间，数据转换时间都应该满足用户需求

4. 当出现新的需求，软件有能力及时调整。有能力修改、增加、删除接口以及数据结构；有能力修改、增加、删除数据精度；有能力适应不同的运行环境。这要求有较高的灵活性。

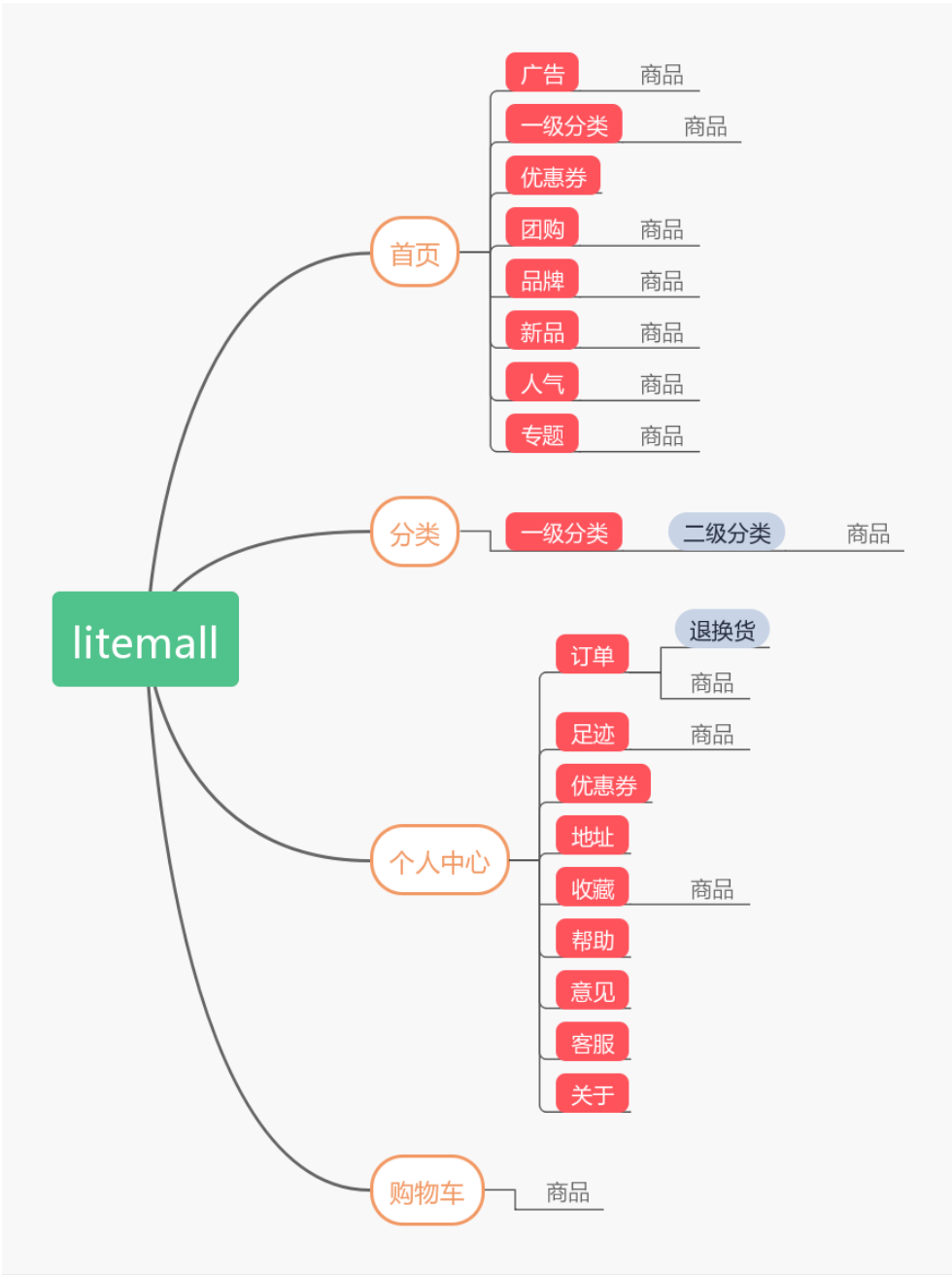
6. 运行需求

6.1 用户界面

用户界面是程序的一部分，用户可以看到并与之交互。目前的 Litemall 设计将为用户提供一个美观、大方、直观、简单的用户界面。具体接口要求如下：

1. 系统采用 b/s 结构和 html 编程。所有接口都使用 Web 样式。设计和实现遵循易用性原则，用户界面布局清晰易懂，操作步骤简单明了。

2. 系统界面主要分为首页、商品详情页、分类、购物车、个人中心等四部分，整个界面的思路图如下：



6.2 硬件接口

服务器端使用阿里云服务器，用于系统的运行和维护

6.3 软件接口

1. 本系统运行于 safari 浏览器或其他的安卓兼容浏览器，要求支持任意 iOS 系统及各版本 Android 系统。
2. 服务器与数据库的接口：通过 ADO 和 JDBC 进行联接，用于查询、上传和下载文件，与数据库连接，进行读出写入。

6.4 故障处理

6.4.1 硬件故障分析

当软件崩溃时，应保证软件功能的完整性，不能发生软件运行故障排除故障后软件不能继续使用的情况。

1. 软件运行过程中产生的数据库错误和网络传输错误，由系统自动记录在错误日志中，提交系统管理员或软件开发人员解决。

2. 当运行设备在运行过程中出现故障时，应按以下方式解决，经处理后软件应处于可运行状态：

（1）设备断电：用户恢复设备电源并重启设备后，可以通过访问本软件恢复软件的正常运行和使用。

（2）设备硬件损坏：用户可以更换损坏的硬件。

（3）内存溢出：用户应检查并解决手机内存不足的问题。内存使用正常后，软件将恢复正常运行和使用状态。

（4）软件运行过程中的其他未知错误，应提交给软件开发人员或由软件开发人员在系统管理员的协助下解决。

6.4.2 软件故障处理

在软件使用过程中，如果出现软件故障，应显示报警信息。

1. 当用户输入一些不符合规定的的数据时，软件会显示一些合理的提示信息，并且不会因用户输入错误而导致系统错误或程序停止运行。问题包括但不限于：

（1）输入的用户名不存在：数据库没有这个用户名。软件将提示无用户。

（2）密码错误：用户名和密码不匹配。将弹出警告消息，用户需要重新登录。

（3）忘记密码：用户可以通过认证重置密码。

（4）因不可抗力造成的未知损失：由用户承担。

2. 软件应具有数据备份机制，以防止数据意外丢失。当软件所依赖的文件损坏或丢失时，可以对话框的形式进行提示，报告损坏的具体相关错误，并使用备份数据修复软件的正常功能。

3. 为了防止由于用户未能及时保存而造成的信息丢失，软件除了提供数据备份机制外，还应提供定期保存机制，并在每隔一定时间自动保存信息，以确保用户数据的安全和软件的方便。

7. 其它需求

7.1 安全性需求

1. 权限控制

根据不同的用户角色设置相应的权限，并对用户的重要操作做相应的日志记录以供查看，如不同等级的管理员能管理的范围也不一样。

2. 重要数据加密

该系统根据一定的算法对一些重要数据进行加密，如用户名、登录密码、重要参数等。

3. 数据备份

在后台，应备份登录信息、日志记录等关键信息，以弥补数据的损坏和丢失。

4. 日志

系统应能记录系统运行过程中发生的错误，包括本地错误和网络错误。这些错误记录便于以后查找错误原因，改进和更新系统。同时，日志还应记录用户的关键操作信息。

7.2 可用性需求

1. 系统的设计应符合美学和人体工程学，且易于使用。

2. 支持没有使用计算机或终端经验、使用计算机或终端经验较少、使用计算机或终端经验较多的用户方便地使用系统。

3. 控制所需条目

系统可以对必须录入的项目进行控制，使用户能够保证信息录入的完整性和正确性同时对必须输入的项目进行有效统一的提示。对于错误的输入，应该有合理的提示和解决方案。

4. 容错性

系统具有一定的容错和抗干扰能力。在非硬件故障或非通信故障的情况下，系统能保证正常运行，并有足够的提示信息帮助用户有效、正确地完成任务。

5. 每个操作完成后统一规范的提示信息

例如，商家修改商品信息后，系统会提示警告框“确定要提交报告吗？该操作不可恢复！”用户点击“确认”后，系统即可开始上传相应信息。上传完成后，系统会将上传结果反馈给用户端。用户确认上传成功后，可以直接返回相应页面。如果上传失败，如（4）所述，应该有相应的解决方案来处理各种错误情况。

6. 用户自定义

为了满足业务需求的变化，需要灵活设置一些重要的参数。

7. 在线帮助和操作手册应便于用户阅读和理解。

7.3 可维护性需求

1. 保留系统相应版本的源代码。代码必须有注释，特别是关于 if、else 和每个条件的含义。它必须有一个清晰的描述。

3. 应该有清晰的系统结构和命名规范、界面规范、提示和帮助信息规范、良好的错误提示信息，可以帮助用户找到自己的原因，自己维护系统。