

三、需求规格说明书

- 1. 引言..... 1
 - 1.1 编写目的..... 1
 - 1.2 项目背景..... 2
 - 1.3 定义..... 2
 - 1.4 参考资料..... 2
- 2. 任务概述..... 2
 - 2.1 目标..... 2
 - 2.2 运行环境..... 3
 - 2.3 条件与限制..... 3
- 3. 数据描述..... 3
 - 3.1 静态数据..... 3
 - 3.2 动态数据..... 3
 - 3.3 数据库介绍..... 4
 - 3.4 数据词典..... 4
 - 3.5 数据采集..... 8
- 4. 功能需求..... 8
 - 4.1 功能划分..... 8
 - 4.2 功能描述..... 9
- 5. 性能需求..... 17
 - 5.1 数据精确度..... 17
 - 5.2 时间特性..... 18
 - 5.3 适应性..... 18
- 6. 运行需求..... 18
 - 6.1 用户界面..... 18
 - 6.2 硬件接口..... 20
 - 6.3 软件接口..... 20
 - 6.4 故障处理..... 20
- 7. 其它需求..... 21

1. 引言

1.1 编写目的

本项目为厦门大学信息学院软件工程专业软件工程系大三上学期《面向对象分析与设计》、《软件工程导论》、《JavaEE 平台技术》课程大作业。本文档编写目的在于介绍课程大作业

的其他模块（包括用户模块、优惠券模块、购物车模块、分享模块、售后模块、对账清算模块）中的需求。

此次编写为其他模块第一次迭代，目前项目进度仍处于设计阶段，尚未开始实际代码的编写。根据课程要求，我们小组计划制作一个能支持高并发、大负载的电子商城系统，其命名为 OOMALL。其旨在能让用户能使用此系统完成和目前主流电商平台相似的购物流程，如购物、发货、售后等。编写此需求说明书是为该项目做服务，预期读者为开发人员和审核人员。

1.2 项目背景

- a. 本项目来源于《JavaEE 平台技术》、《软件工程导论》、《面向对象分析与设计》课程设计大作业需求
- b. 项目开发者与设计者：袁佳哲、刘赫昭、程昊天、徐荪睿、谢健祥
- c. 客户端用户：有电商系统使用需求的用户，包括买家与卖家（店铺）
- d. 后台管理用户：后台管理人员，店铺也可优先参与管理

1.3 定义

客户：Customer，使用电商系统进行购买等功能的用户，是系统的主要用户

卖家：Shop，也称之为店铺，是系统的第二主要用户

客户端：客户访问系统的途径，在本项目中客户端主要由 API 组成

服务端：集业务逻辑、数据存储等功能与一身的服务器

1.4 参考资料

- a. OOAD 课程文件《成绩计算办法 V8》
- b. 阿里巴巴 Java 开发手册
- c. Roger S.Pressman , Bruce R.Maxin. 软件工程 实践者的研究方法[M]：机械工业出版社.2018.8
- d. 中国大学慕课厦门大学课程《JavaEE 平台技术》

2. 任务概述

2.1 目标

市场目标：高并发大负载电商系统主要源自于当前人们对网络购物的需求。在模仿当前主流电商平台（如京东）的基础上，我们希望借助于本学期的课程所学内容，包括设计原则、设计思想、JavaEE 平台技术等，实现我们想要的功能，以及在基础功能上做出扩展，使用户能便捷的满足自身需求

技术目标：针对客户端，希望用户能正常登录、注册，并完成购物、售后等功能，用户主要通过 API 对系统进行访问。针对后台管理系统，是以控制端界面呈现，通过大量测试用例，进行代码测试，并接受测试结果，管理者可以查看控制端信息，进行相应操作。针对服务器，主要是用 Ubuntu 服务器，实现数据存储、共享与同步。总体而言，以上技术的实现主要是前端请求与后台方面的知识。

2.2 运行环境

操作系统：Ubuntu 18.04

支持环境：IIS 5.0

数 据 库：MySQL 8.0、Redis 6.2.4

2.3 条件与限制

经费限制：云服务器租用费用

开发期限：两个月

硬件限制：客户端只能通过 API 方式进行访问；后台管理系统通过 Terminal 终端进行查看，以及可借助浏览器以网站方式进行性能监控等；服务端运行于 Ubuntu Linux 系统

3. 数据描述

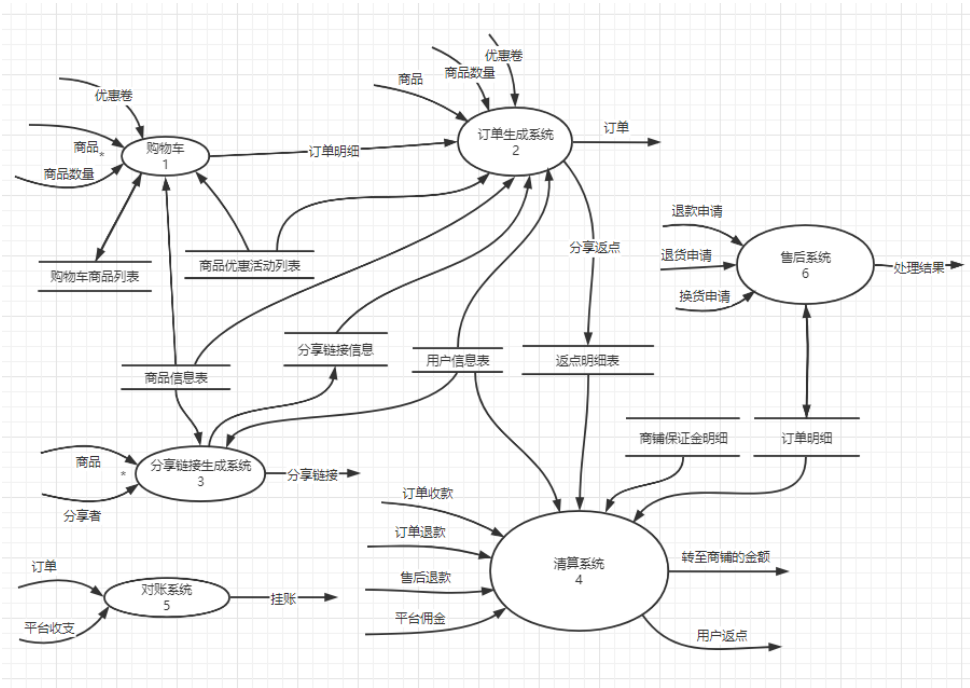
3.1 静态数据

用户：用户(卖家，卖家)的账号密码、邮箱、状态(正常、封禁等)、创建时间、修改时间
店铺：店铺名称、创建时间、修改时间等
以及数据字典部分

3.2 动态数据

1. 用户登录时提供的账号与密码。
2. 修改个人信息时输入的昵称、邮箱等。
3. 优惠券的使用范围、使用时间
4. 用户分享商品时生成的分享链接
5. 通过分享链接购买时，分享者的返点
6. 商铺生成的每日流水信息
7. 用户购物车里包含的项目
8. 用户在评论时提供的，评论信息

数据流图



3.3 数据库介绍

关系型数据库 MySQL8.0

3.4 数据词典

名称：顾客表

别名：用户表

描述：记录系统用户(卖家)的基本信息

定义：用户表=主键+用户名+性别+真实姓名+密码+邮箱+状态+电话+创建时间+修改时间

属性描述：

主键 Id：自增数字。

用户名：至少 4 位字符，系统内唯一。

性别：男 女。

真实姓名：2~8 个汉字。

密码：长度至少 8 位，由字母加数字组成。

邮箱：邮箱用户名+@+域名。

状态：0,1 注：“0”表示正常用户，“1”表示封禁用户。

电话：11 位数字。

创建时间：年+月+日+24 时制小时+分钟+秒。

修改时间：年+月+日+24 时制小时+分钟+秒。

名称：我的优惠券表

别名：无

描述：记录此用户拥有的优惠券

定义：我的优惠券表=主键+用户 id+持有数量+优惠券 id+获取时间

属性描述：

主键 Id：自增数字。

用户 id：外键，与用户表的 id 相同。

持有数量：数字，0~+∞。

优惠券 id：与优惠券的 id 相同(商品模块中)。

获取时间：年+月+日+24 时制小时+分钟+秒。

名称： 我的评论

别名： 无

描述： 记录此用户对商品的评论。

定义： 我的评论=主键+用户 id+评论 id。

属性描述：

主键 Id：自增数字。

用户 id：外键，与用户表的 id 相同。

评论 id：外键，与评论表中的 id 相同。

名称： 用户积点表

别名： 无

描述： 记录用户的积点数据

定义： 用户积点表=主键+用户 id+用户积点+是否可用+产生时间+修改时间。

属性描述：

主键 Id：自增数字。

用户 id：外键，与用户表的 id 相同。

用户积点：数字，0~+∞。

是否可用：0，1。0 表示可用，1 表示不可用。

产生时间：年+月+日+24 时制小时+分钟+秒。

修改时间：年+月+日+24 时制小时+分钟+秒。

名称： 收货信息表

别名： 收货地址表

描述： 记录用户的收货信息

定义： 收货信息表=主键+用户 id+收货人+手机号码+详细地址+邮政编码+是否为默认。

属性描述：

主键 Id：自增数字。

用户 id：外键，与用户表的 id 相同。

手机号码：11 位数字

详细地址：至多 100 位字符

邮政编码：6 位数字

是否位默认：0，1；1 表示默认，0 表示不为默认。

名称： 地区表

别名： 无

描述： 记录用户的地区信息

定义：地区表=主键+上级地区 id+地区名称+邮政编码+状态

属性描述：

主键 Id：自增数字。

上级地区 id：自增数字，为地区表中的 id。

地区名称：2~20 位字符。

邮政编码：6 位数字

状态：0，1；0 表示不可发货 1 表示可发货。

名称： 购物车细则表

别名： 购物车表

描述： 记录此用户的购物车信息

定义： 购物车明细表=主键+用户 id+货品 id+货品数量+加入单价

属性描述：

主键 Id：自增数字。

用户 id：外键，与用户表的 id 相同。

货品 id：外键，与货品表中的 id 对应

货品数量：数字，0~+∞。

加入单价：数字，0~+∞。

名称： 分享表

别名： 无

描述： 记录分享相关的信息

定义： 分享表=主键+分享者 id+分享来源商品 id +成功返点的商品件数+创建时间+修改时间

属性描述：

主键 Id：自增数字。

分享者 id：外键，与用户表的 id 相同。

分享来源商品 id：外键，与商品表中 id 对应。

成功返点的商品件数：数字，0~+∞。

创建时间：年+月+日+24 时制小时+分钟+秒。

修改时间：年+月+日+24 时制小时+分钟+秒。

名称： 分享成功表

别名： 无

描述： 记录分享成功的相关信息，用于计算用户的返点

定义： 分享表=主键+货品 id+分享者 id+分享 id +分享活动 id+点击者 id+订单明细 id+返点数+创建时间+修改时间

属性描述：

主键 Id：自增数字。

货品 id：外键，与商品表中 id 对应。

分享者 id：外键，与用户表的 id 相同。

分享 id：自增数字。。

分享活动 id：外键，对应活动表中的 id。

点击者 id：外键，与用户表的 id 相同。

订单明细 id: 外键, 与订单表中的 id 对应。

返点数: 数字, $0 \sim +\infty$ 。

创建时间: 年+月+日+24 时制小时+分钟+秒。

修改时间: 年+月+日+24 时制小时+分钟+秒。

名称: 售后表

别名: 售后信息表

描述: 记录用户的售后信息

定义: 售后表=主键+售后订单 id+申请售后用户 id+商铺 id+售后单序号+类型+原因+处理意见+金额+数量+地区 id+详细地址+联系人+电话+寄出运单号+状态+逻辑删除+创建时间+修改时间。

属性描述:

主键 id: 自增数字。

售后订单 id: 外键, 与订单表中的 id 对应。

申请售后用户 id: 外键, 与用户表的 id 相同。

商铺 id: 外键, 与商铺表的 id 相同。

售后单序号: 20 数字字母随机组合, 在系统中唯一。

类型: 服务类型: 0 换货, 1 退货, 2 退款。

原因: 0-100 个字符

处理意见: 0-100 个字符

金额: 数字, $0 \sim +\infty$ 。

数量: 数字, $1 \sim +\infty$ 。

地区 id: 外键, 与地区表的 id 相同。

详细地址: 0-200 个字符。

联系人: 2-20 个字符

电话: 11 位数字。

寄出运单号: 20 位数字字母随机组合, 由快递方产生。

状态: 0 表示未处理, 1 表示已处理

逻辑删除: 0 表示未删除, 1 表示已经删除。

创建时间: 年+月+日+24 时制小时+分钟+秒。

修改时间: 年+月+日+24 时制小时+分钟+秒。

名称: 商铺流水表

别名: 商铺流水

描述: 记录商铺所有流水

定义: 商铺流水表=主键+订单 id+流水状态+创建时间+修改时间

属性描述:

主键 Id: 自增数字。

订单 id: 外键, 与订单表的 id 相同。

流水状态: 0、1; 0 表示未对账; 1 表示已经对账。

创建时间: 年+月+日+24 时制小时+分钟+秒。

修改时间: 年+月+日+24 时制小时+分钟+秒。

名称: 清算记录表

别名：清算记录

描述：记录系统清算详情

定义：清算记录表=主键+订单记录 id+订单明细 id+运费 id+店铺 id+顾客 id+顾客积点 id

属性描述：

主键 id：自增数字。

订单记录 id：外键，与当天订单记录表的 id 相同。

订单明细 id：外键，与订单明细表的 id 相同。

运费 id：外键，与运费表的 id 相同。

店铺 id：外键，与店铺表的 id 相同。

顾客 id：外键，与用户表的 id 相同。

顾客积点 id：外键，与用户积点表的 id 相同。

名称：当天订单记录表

别名：当天订单记录

描述：记录当天对账状态

定义：当天订单记录表=主键+订单 id+对账完成状态

属性描述：

主键 id：自增数字。

订单 id：外键，与订单表的 id 相同。

对账完成状态：0、1；0 表示未对账完成；1 表示已经对账完成。

名称：挂帐单表

别名：挂帐单

描述：记录挂账处理状态

定义：挂帐单表=主键+订单 id+挂账处理状态

属性描述：

主键 id：自增数字。

订单 id：外键，与订单表的 id 相同。

挂账处理状态：0、1；0 表示挂账未处理；1 表示挂账已处理。

3.5 数据采集

菜单对应指令编号

4. 功能需求

4.1 功能划分

系统其他模块划分为：

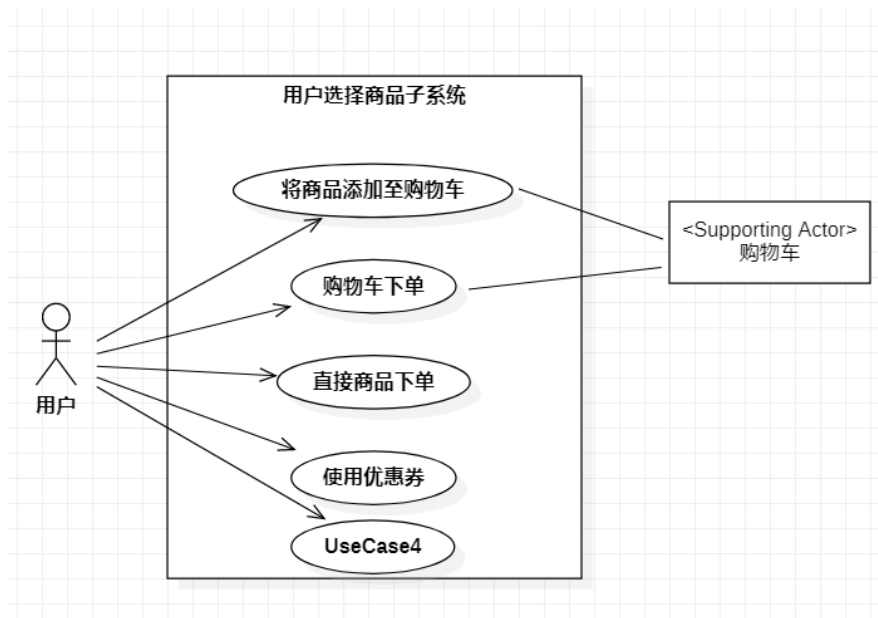
1. 用户选择商品：负责获得用户选择的商品以及对应是否使用优惠券。
2. 用户分享返点：负责用户分享商品后，生成分享链接，当有其他用户通过该链接购买商

- 品时，计算出分享者应该获得的返点。
3. 对账：负责对各个商品的每日流水和平台的收支进行对账。
 4. 清算：负责和店铺清算 n 天前对完账的账目，同时负责清算用户分享该获得的返点。
 5. 售后：用户退款或退换货

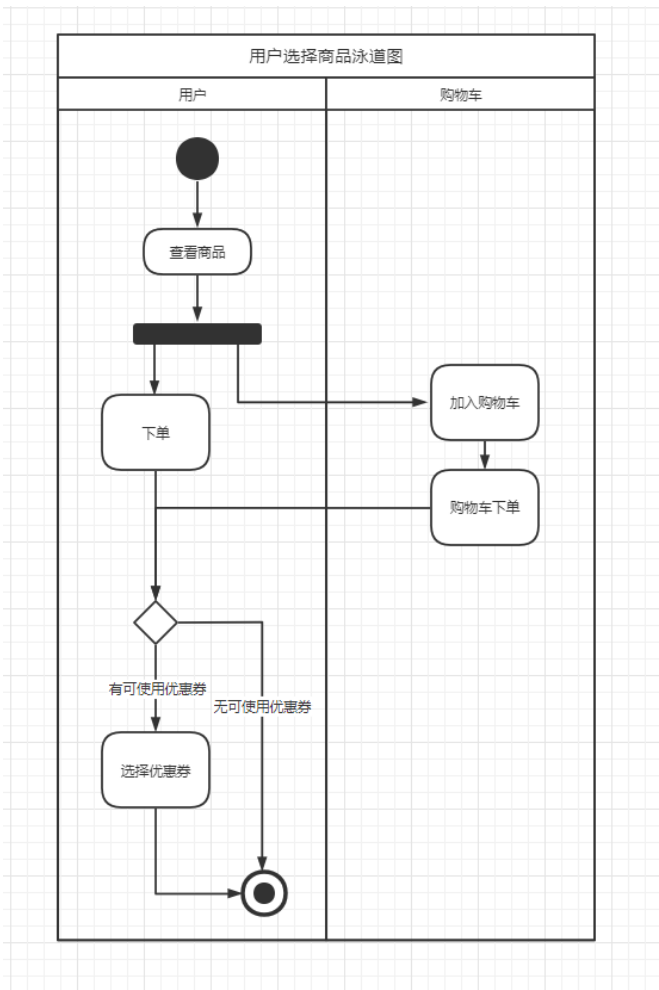
4.2 功能描述

(一) 用户选择商品

- a) 外部用户
 - 用户
- b) 功能
 - 用户选择商品加入购物车
 - 用户直接下单
 - 用户决定是否使用优惠券
- c) 用例图

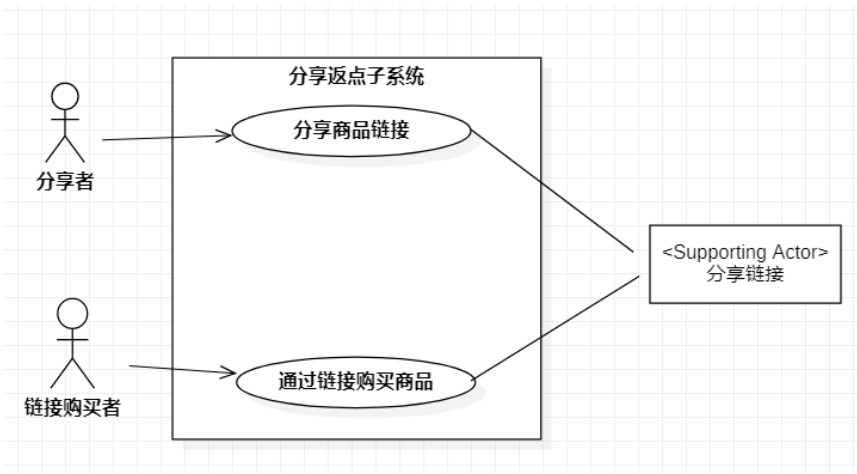


d) 泳道图

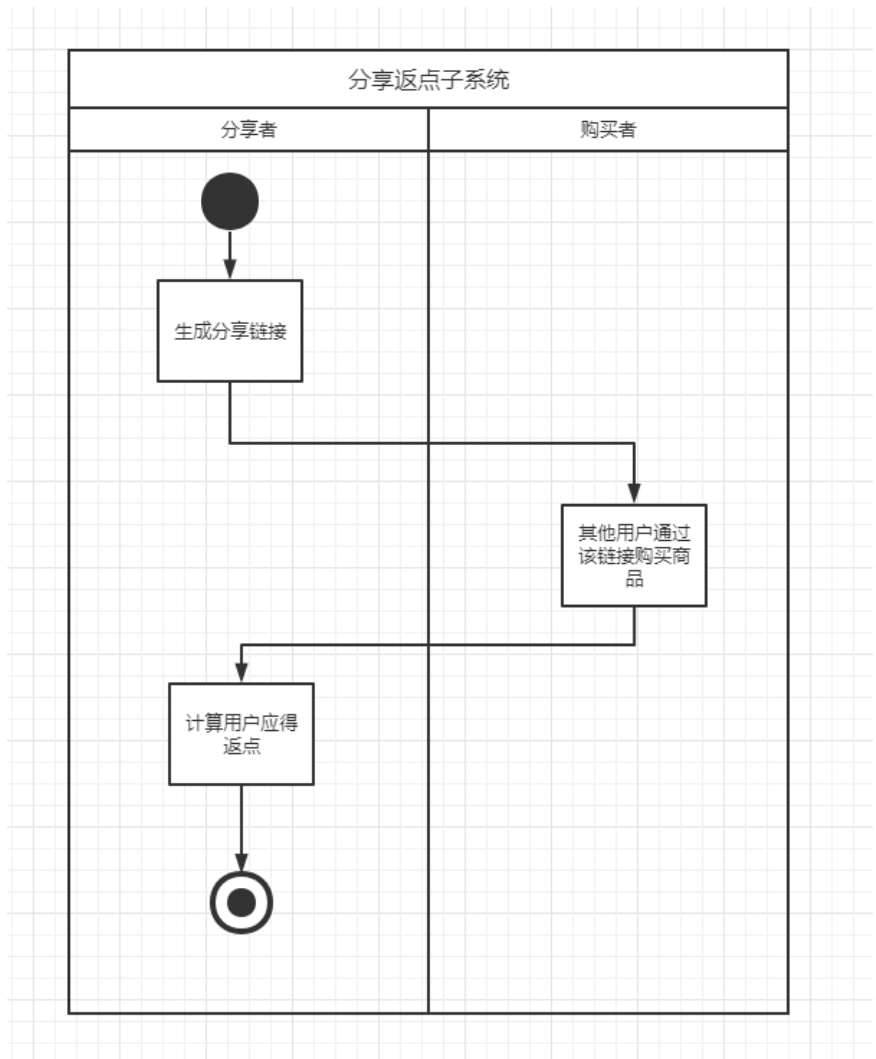


(二) 用户分享返点

- a) 外部用户
 - 用户
- b) 功能
 - 生成用户分享商铺的链接
 - 计算用户的返点
- c) 用例图

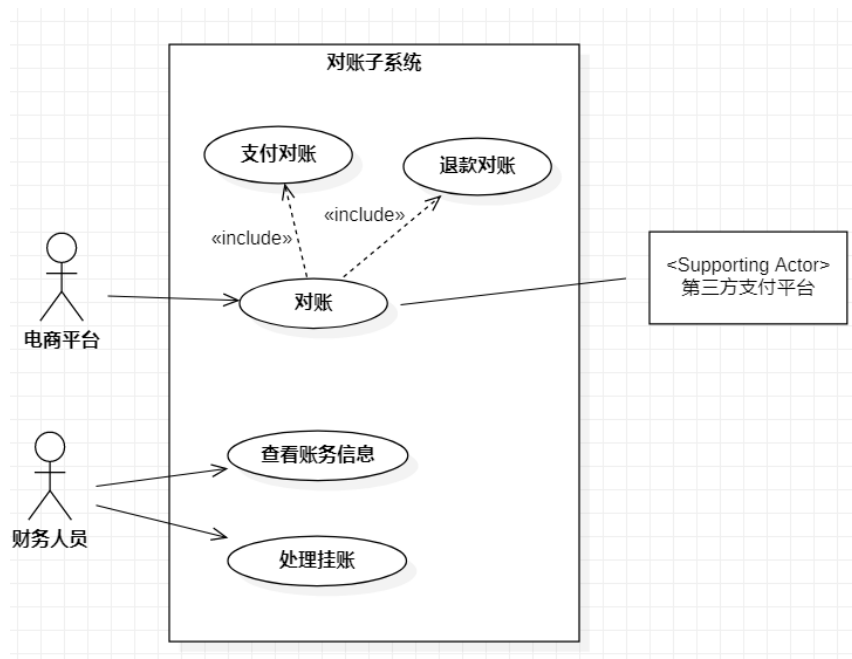


d) 泳道图

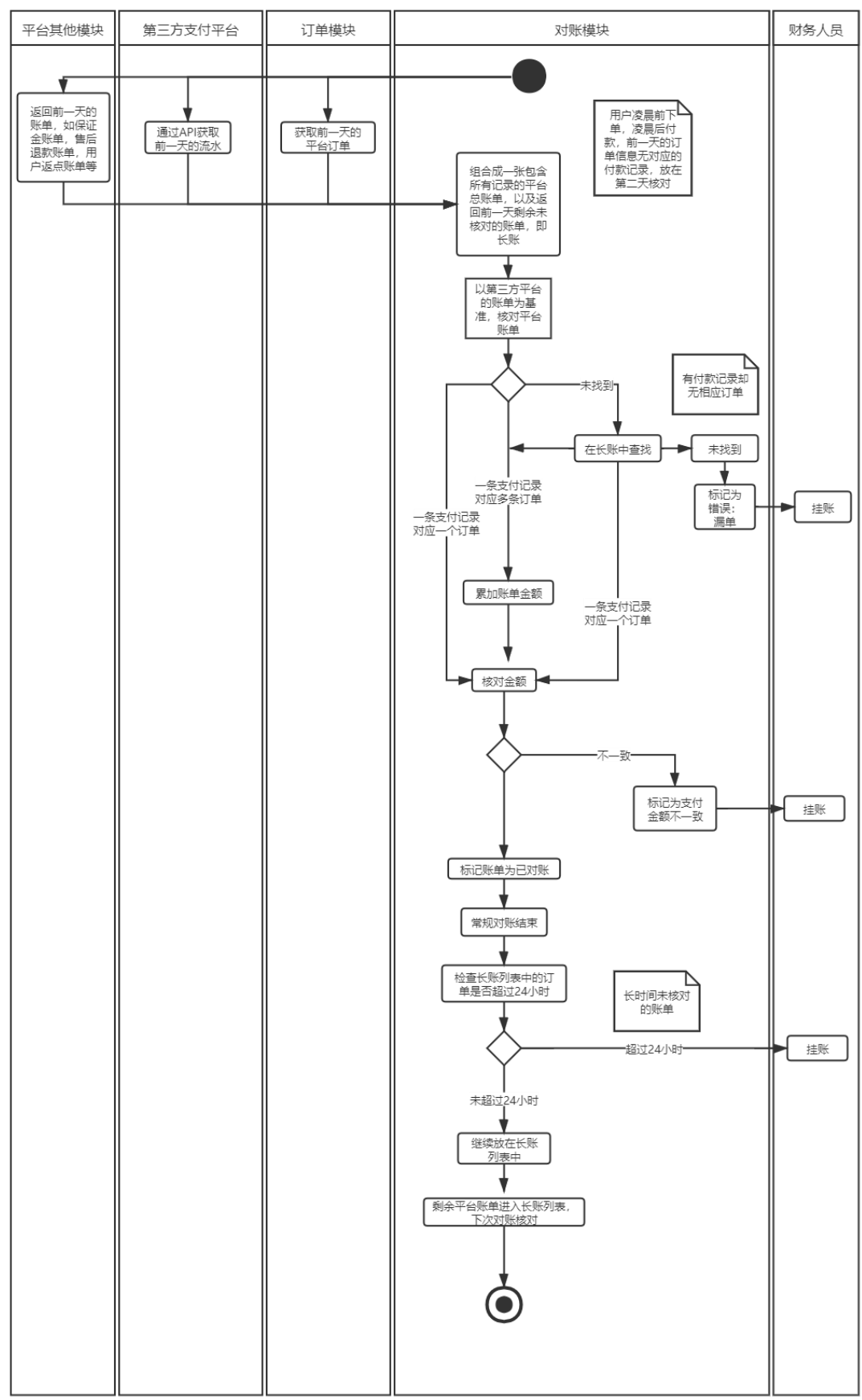


(三) 每日对账

- a) 外部用户
 - 商铺
- b) 功能
 - 核对商铺流水与平台收支
- c) 用例图

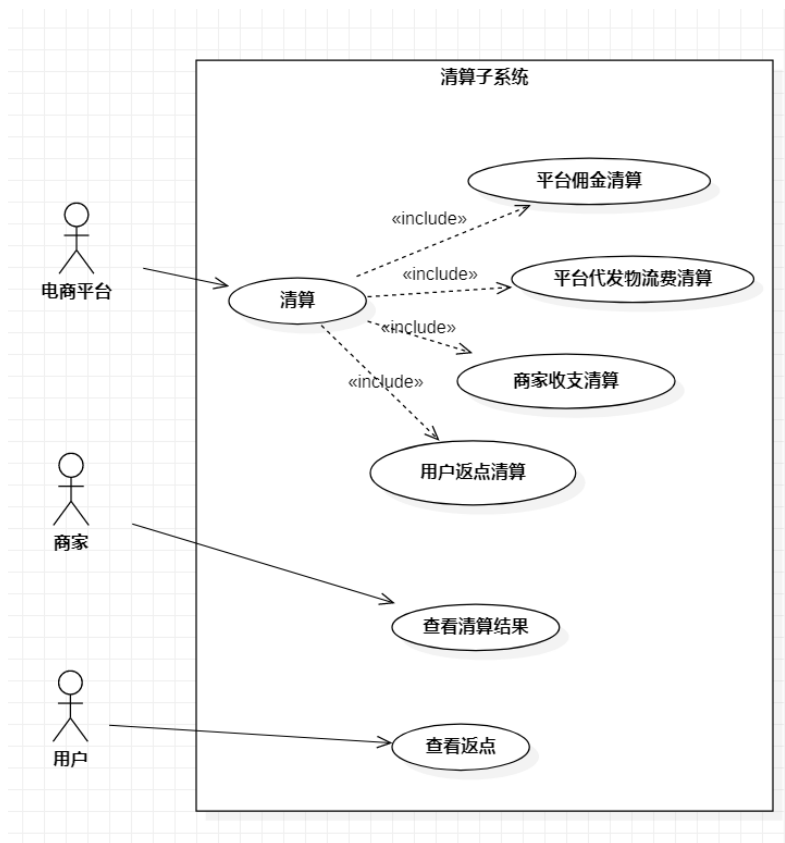


d) 泳道图



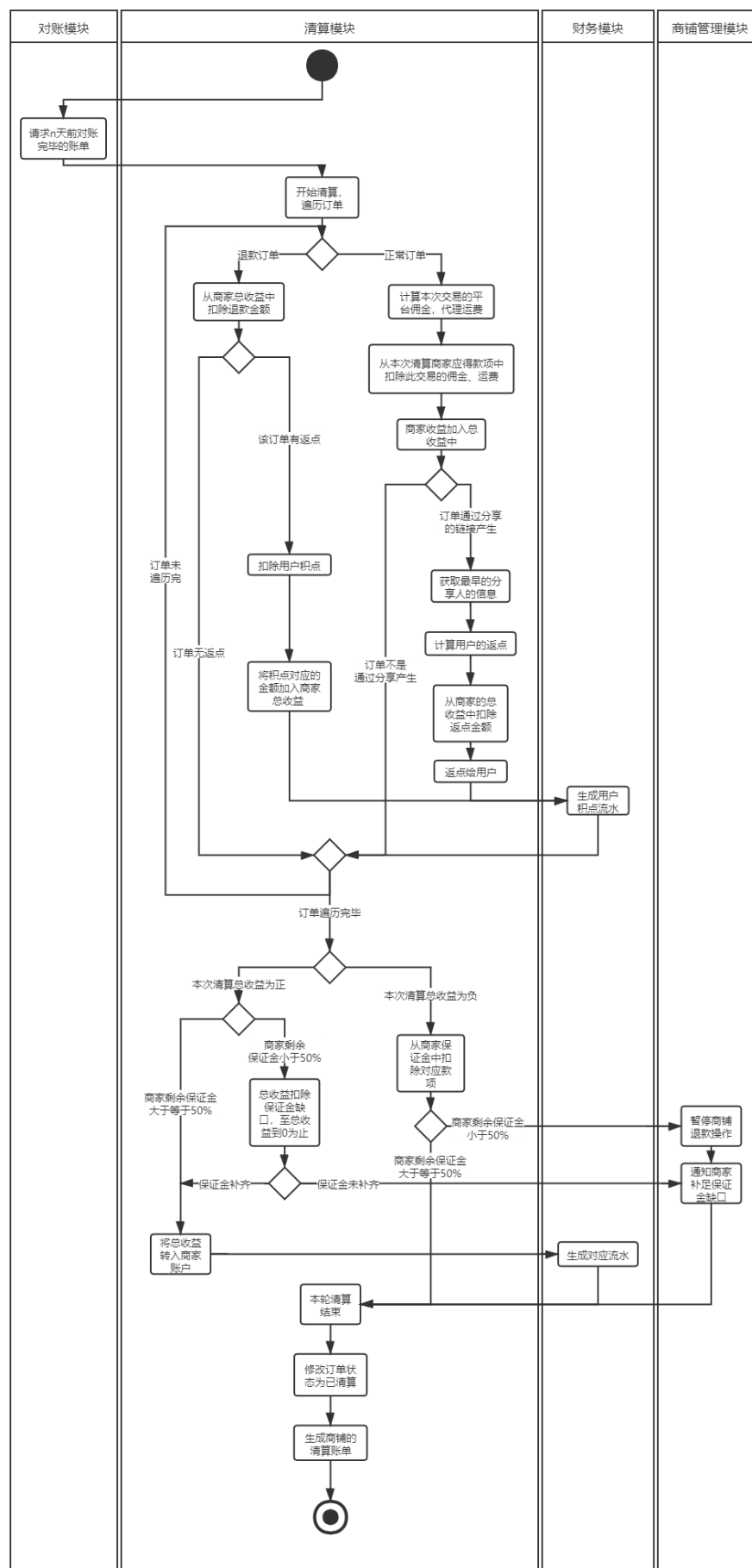
(四) 清算

- a) 外部用户
 - 用户
 - 商铺
- b) 功能
 - 清算 n 天前与商铺对好的账目
 - 清算用户应获得的返点
- c) 用例图



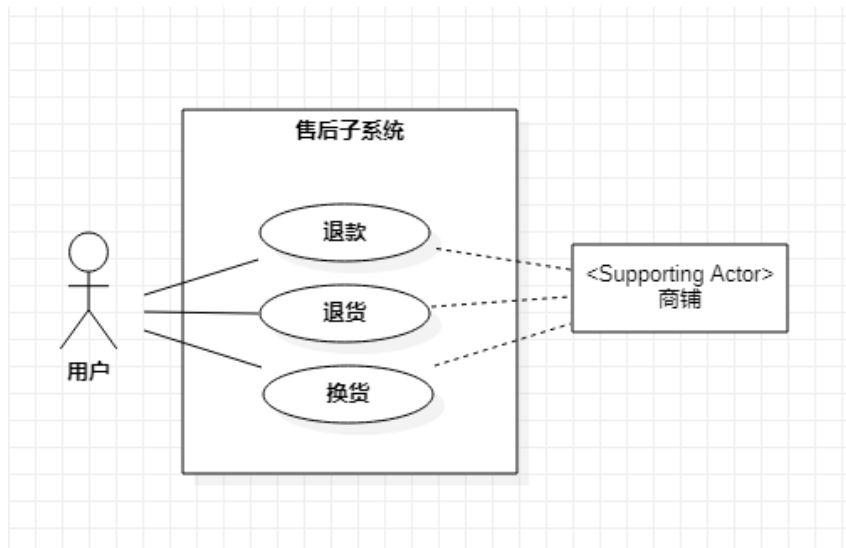
- d) 泳道图

三、需求规格说明书

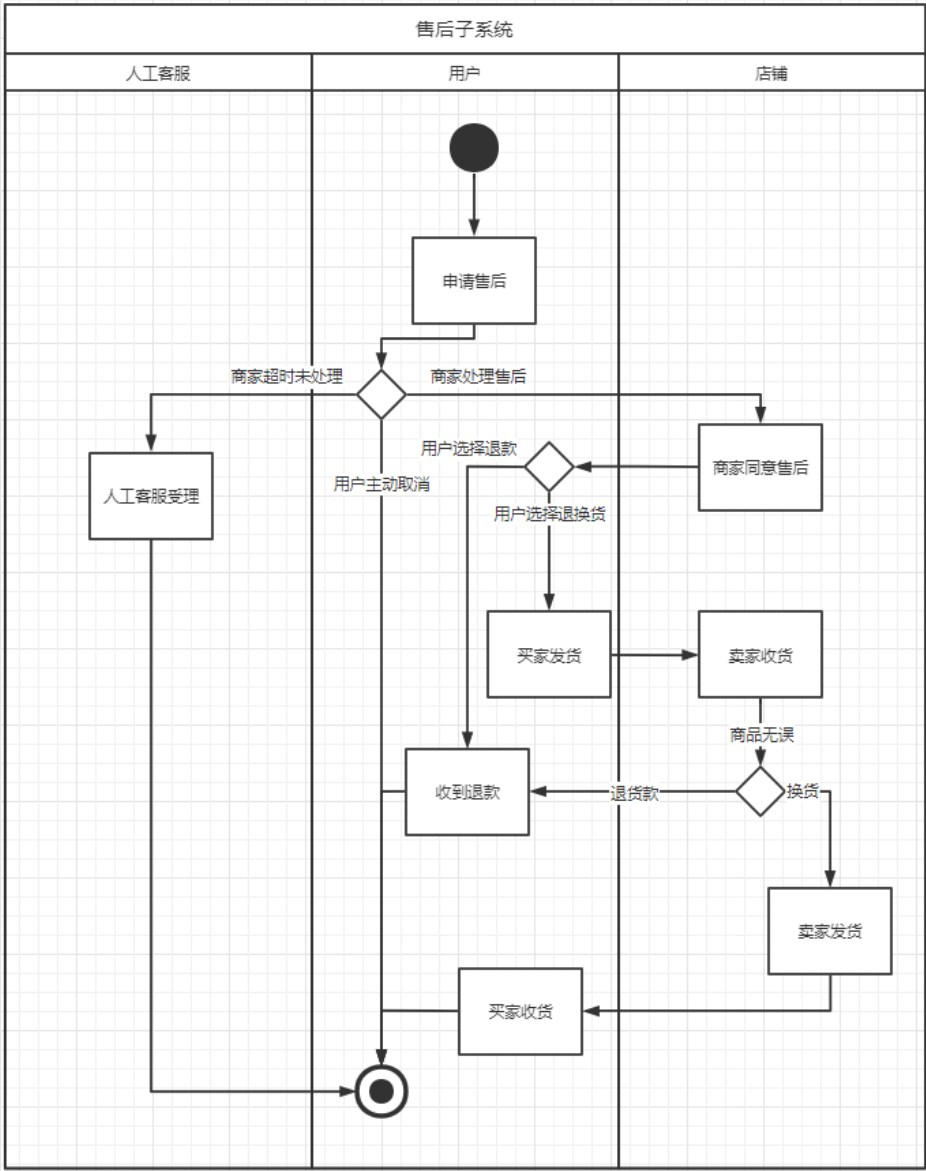


(五) 售后

- a) 外部用户
 - 用户
 - 商铺
- b) 功能
 - 用户退款
 - 用户退货
 - 用户换货
- c) 用例图



- d) 泳道图



5. 性能需求

5.1 数据精确度

该系统中涉及到的输入、输出数据均在数据库存储，数据类型和数据精度等详细信息可参见如上 ER 图和数据字典。

该软件主要涉及的输入、输出数据主要以 json 字符串格式呈现。

数据主要存储于数据库中，数据类型和数据精度可参见数据字典与 E-R 图

5.2 时间特性

本项目的一些具体时间特性要求如下：

可在 1 秒内通过 1000 个测试用例

5.3 适应性

本项目的适应性部分要求如下：

1. 最基本的设计原则符合面向对象的高内聚低耦合原则
2. 保证开闭原则，即进行版本迭代时只需要新增或修改部分代码，其余部分不需要更改
3. 运行环境：打包好的 jar 包，只要在安装 jdk 环境的系统即可正常运行
4. 功能插拔：部分功能最终可以开源，以及支持热插拔，即在系统正常运行过程中也可增加或修改部分功能，如支付功能可实现热插拔

6. 运行需求

6.1 用户界面

本项目无前端页面，用户通过 Resultful 风格 API 对系统进行访问

例：

用户服务：

customer 买家用户服务		^
GET	/customers/states 获得买家的所有状态	✓ ↩
POST	/customers 注册用户	✓ ↩
GET	/customers 买家查看自己信息	✓ ↩
PUT	/customers 买家修改自己的信息	✓ ↩
PUT	/customers/password 用户修改密码	✓ ↩
PUT	/customers/password/reset 用户重置密码	✓ ↩
GET	/customers/all 平台管理员获取所有用户列表	✓ ↩
POST	/customers/login 用户名密码登录	✓ ↩

售后：

aftersale 售后服务			^
GET	/aftersales/states	获得售后单的所有状态	v ↩
POST	/orderitems/{id}/aftersales	买家提交售后单	v ↩
GET	/aftersales	买家查询所有的售后单信息（可根据售后类型和状态选择）	v ↩
GET	/shops/{id}/aftersales	管理员查看所有售后单（可根据售后类型和状态选择）	v ↩
GET	/aftersales/{id}	买家根据售后单id查询售后单信息	v ↩
PUT	/aftersales/{id}	买家修改售后单信息（店家生成售后单前）	v ↩
DELETE	/aftersales/{id}	买家取消售后单和逻辑删除售后单	v ↩

6.2 硬件接口

客户端要求系统须支持浏览器访问，或支持 jmeter 测试计划运行
后台管理系统无硬件要求
服务端要求服务器为 Linux 系统，至少 2GB 内存

6.3 软件接口

客户端要求使用浏览器访问 API，或使用 jmeter 对 url 进行测试
后台管理系统使用 terminal 终端查看，或者使用浏览器进行后台监控
服务端要求可运行 jvm，部署 MySQL、Redis、RocketMQ 等

6.4 故障处理

1. 用户无法正常登录
2. 用户无法查看商城信息
3. 用户无法正常使用商城功能
4. 系统因宕机无法对请求作出响应

处理要求：尽可能快速、彻底解决用户遇到的问题

7. 其它需求

7.1 检测或验收标准

可支持每秒 1000 个测试用例，白盒测试 100%覆盖率，只允许最多 30 个测试用例出错

7.2 安全性要求

对关键信息进行加密，对重要信息进行签名验证

对于所有外部 API,使用网关进行验证,如果外部 API 对内部 API 进行访问(带有 internal),直接在网关对其进行拦截。非访问内部且合法的外部 API 才可通过网关对系统进行正常访问。

7.3 开发要求

使用 MySQL 实现数据库存储，并使用 Redis、RocketMQ 等对数据库读写进行性能提升辅助
主要代码使用 Java 开发，使用 Spring-boot 进行项目构建，使用 Maven 进行代码管理

7.4 质量属性

拥有较好的可用性、可靠性、可维护性、安全性、可移植性等