

四、概要设计说明书

1. 引言.....	2
1.1 编写目的.....	2
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.....	3
2.2 运行环境.....	3
2.3 需求概述.....	3
2.4 条件与限制.....	5
3. 总体设计.....	5
3.1 处理流程.....	5
3.2 总体结构和模块外部设计.....	8
3.3 功能分配.....	11
4. 接口设计.....	12
4.1 外部接口.....	12
4.2 内部接口.....	13
5. 数据结构设计.....	14
5.1 逻辑结构设计.....	14
5.2 物理结构设计.....	22
5.3 数据结构与程序的关系.....	22
6. 运行设计.....	24
6.1 运行模块的组合.....	24
6.2 运行控制.....	24
6.3 运行时间.....	25
7. 出错处理设计.....	25
7.1 出错输出信息.....	25
7.2 出错处理对策.....	26
8.安全保密设计.....	26
9.维护设计.....	28

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此项目做服务，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，店铺也可有限参与管理

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

管理员：Admin，电商平台的管理员，负责日常维护和监控

最高管理员：ADMIN，电商平台创建之初的用户，最高权限的管理员，拥有所有权限

客户端：客户访问系统的途径，在本项目中客户端无用户界面，用户通过 API 进行访问

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 V11》
- 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 需求概述

本学期的课程设计内容是采用面向对象的方法设计和实现一个高负载大并发的电子商城的后端系统。系统的需求基本基于京东系统，在此基础上增加或简化以下需求：

（黄色底色为其他模块内容，灰色为必做模块内容）

1. **修改团购功能**：对于特定商品设定不同等级的团购数量（如 1-100 件 9 折，101-200 件 8 折，201 件以上 7 折）。用户在提交团购订单时按照商品的正常价格支付订单，在规定时间内达到不同等级团购数量则团购成功，按照不同的团购等级原渠道退回用户多支付的金额（团购不支持使用优惠券和返点）。未达到团购门槛则全额退款。管理员可以选择让正在进行中团购下线，下线的团购即可终止团购活动，无论是否成团都不能享受折扣，按照未达到成团门槛处理。

2. **修改商品分享功能**：支持店铺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购买了该产品，则在清算时计入分享用户的返点。当商品有多个分享成功的情况，返给在活动有效时间内最早分享成功的用户。每次分享成功只能享受一次返点。可以为店铺定义默认的分享规则

3. **修改管理员的后台管理功能**：管理员后台所有的功能均需要记住由谁在何时新建、修改

了数据

4. **支持用户用积点和优惠券**，以及通过微信和支付宝平台支付以及退款。支付前端确定为小程序。用户在使用返点购买商品时，一点抵扣一分，不限制用户在一笔订单中使用返点的上限。退货时优先退回返点。

5. **添加平台级优惠券和店铺优惠券**

平台级优惠券：如限品类东券：满 300 减 30，第二或第三件半价（选择价钱最低的半价），商品满 3 件 7 折等等，我们要求这一部分的设计可扩展未来的新的优惠券种类，所有商铺都可以使用。我们的优惠券不可叠加

店铺优惠券：种类同平台级优惠券，但只能限定在本店铺使用，优惠券不能叠加优惠券应支持分类和指定商品使用，可支持各种不同类型的优惠券。优惠券需设定起止期限，在起止期限中才可以领用或使用优惠券。管理员可以作废优惠活动，优惠活动作废后，活动所有领出未用的优惠券也一并作废，已经使用的优惠券不受影响。

优惠券不支持在团购和预售中使用

6. **增加商品预售功能**：商品的付款分为预售款和尾款两部分支付。商品在同一时刻预售和团购只能有一个活动生效且只能是单纯预售或团购活动，不能同时附加优惠活动和分享活动。在一个团购或预售订单中只能有一条商品规格。管理员可以选择让正在进行中预售下线，下线的预售全额退款。

7. **修改商品评论功能**：商品的评论需要经过平台管理员审核后才能显示在商城中。删除商品亦一并删除商品的评论、用户在购物车内的商品、用户所有分享的商品

8. **修改查询功能**：删除关键字查询功能，改为以商品名称查询，只能查询起头的名称，删除查询历史的功能

9. **修改加密功能**：用户的密码、电话、Email 需要用 ASE 算法加密存储在数据库中

10. **定义商铺（商户）功能**：支持多商户的电子商城，每个商户仅仅可以管理自己商户的商品。商品的所有权是各个商户的，但需统一放在电子商城的仓库中进行销售和配送。每个商户需缴纳设定的保证金。保证金低于设定比例时，限制商户的退款交易。商户可定义多个账户，可设定清算时从平台转账优先顺序。

11. **支持用户敏感信息的保护功能**：如对于没有权限的用户，返回的订单信息中，用户的姓名只显示姓，名字用**代替，电话号码仅显示前三位和后四位，其他位用****代替

12. **增加优惠活动**：优惠活动与优惠券相似，但不需要发行优惠券，只需指定商品或分类的商品适用于某种活动（如满 300 减 30），同样优惠活动有起止时间

13. **修改发货功能**：在发货之前可以修改配送地址，修改地址不影响运费

14. **修改订单功能**：当订单中有多个店铺的商品时，用户支付订单后，需要分成多个店铺的订单。将支付金额和支付的积点按比例分摊给各子订单。团购和预售商品不能放入购物车中，既不能和其他商品合并在一个订单中。

15. **修改售后功能**：售后支付采用通过虚拟商品订单的方式支付。即允许商铺定义单价为 1 元的虚拟商品，让顾客通过购买虚拟商品的方式支付售后费用。

16. **添加对账清算功能**：

支持第三方支付平台的对账，以及第三方商铺的清算和个人用户的分享返点清算。

对账是每天凌晨都会自动进行的操作。对前一天的账目进行对账。清算是在对账完成后再做的操作，可以定义 T+n 的清算，即在对账完成后 n 天进行清算。

平台按照不同商品的类别抽取支付佣金。由于平台代为管理仓储和配送的商品，平台还需收取仓商品的物流费用。

清算时需分别计算商铺的订单收款、订单退款、售后退款、平台的支付佣金以及对分享用户的返点（一点抵扣一分钱），如果清算出的转账金额为负数则从商铺保证金中扣除。多

余款项汇出到商铺账户。如果商铺保证金低于设定比例时，则会在售后流程中暂停商铺订单的退款。清算的时候，如果商铺保证金低于设定比例，先补足商铺保证金，再把钱转账到商家指定账户。

2.4 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：

系统只能运行在指定的服务器上，一般情况下，拥有五台服务器，其各自硬件限制如下：

Slave1: Ubuntu 18.04 服务器 2 核 4G 内存虚拟机一台，支持图形界面

Slave2-5: Ubuntu 18.04 服务器 2 核 2G 内存虚拟机，仅支持命令行界面

所有服务器需安装 maven、jdk 11、git，硬件必须支持以上需求

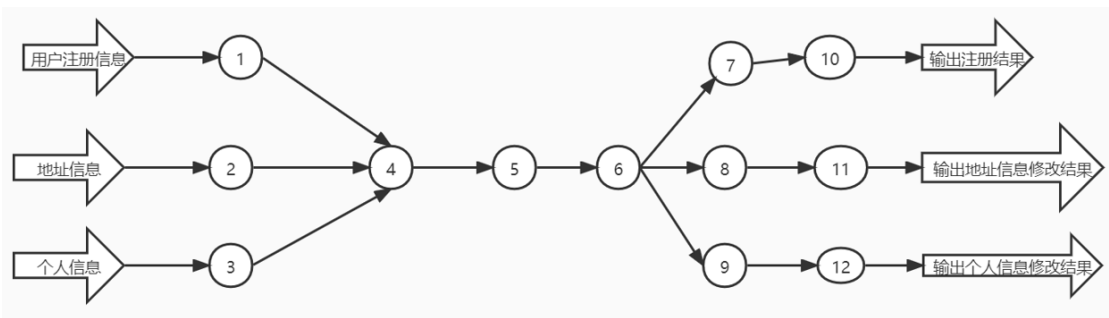
3. 总体设计

3.1 处理流程

下面将使用面向数据流的方法对 OOMALL 系统其他模块的处理流程进行分析。

系统可分为以下部分：

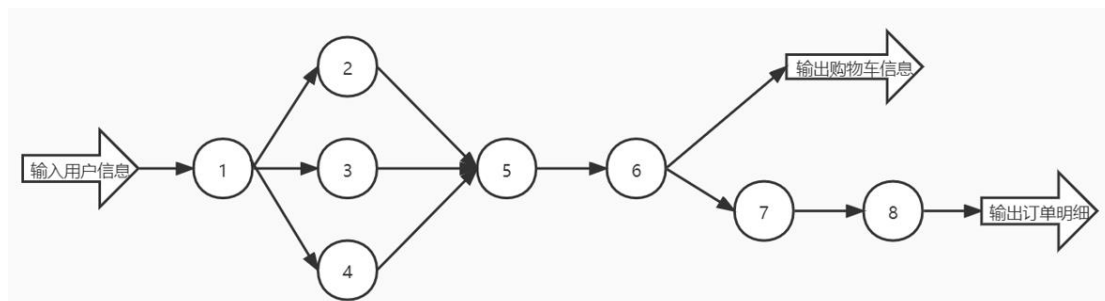
用户系统：



- 1: 读入用户注册信息，将其格式化，送 4
- 2: 读入地址信息，将其格式化，送 4
- 3: 读入个人信息，将其格式化，送 4
- 4: 检验格式化信息，初步检查输入是否有误，有误则转 1 或 2 或 3 再次读入，检测无误则送 5
- 5: 将合法的输入数据经网络送服务器程序，送 6
- 6: 解开 5 的数据包，如果是注册请求送 7，地址请求送 8，个人信息修改请求送 9
- 7: 运行 SQL 语句，将用户注册信息写入数据库，将执行结果送 10
- 8: 运行 SQL 语句，将地址信息写入数据库，将执行结果送 11
- 9: 运行 SQL 语句，查询并修改对应数据库表，将执行结果送 12

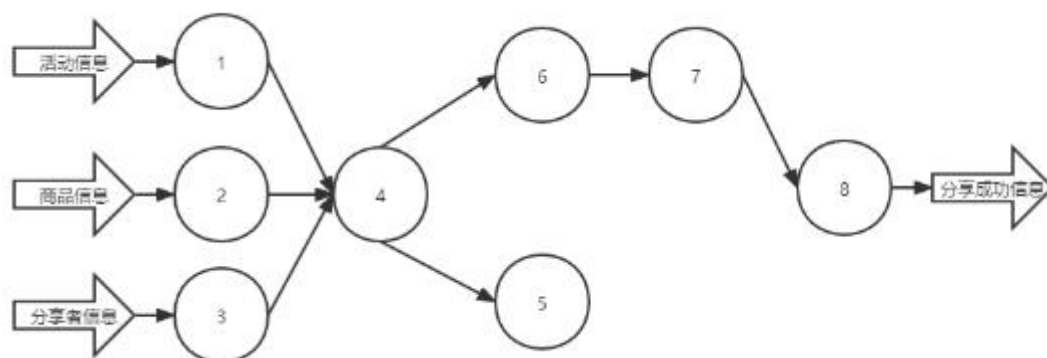
- 10: 由注册结果输出 json 格式的结果信息
- 11: 由地址信息写入结果输出 json 格式的结果信息
- 12: 由个人信息修改结果输出 json 格式的结果信息

购物车系统:



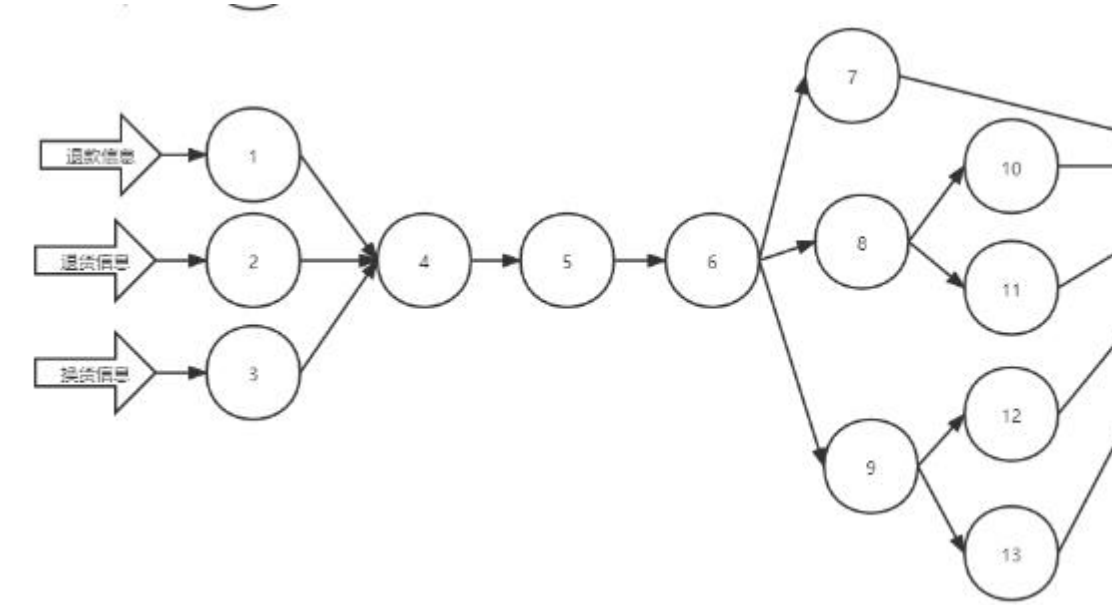
- 1: 读入用户信息，将其格式化，将用户信息送 2、3、4
- 2: 根据用户信息，运行 SQL 语句，查询商品信息，送 5
- 3: 根据用户信息，运行 SQL 语句，读优惠券信息，送 5
- 4: 根据用户信息，运行 SQL 语句，读商品活动优惠信息，送 5
- 5: 根据 345 出购物车中信息，送 6
- 6: 判断请求类型，若是获取购物车信息，输出购物车信息，若为生成订单明细，将对应购物车信息送 7
- 7: 根据购物车中商品明细，在业务逻辑层计算商品价格，生成订单明细，送 8
- 8: 输出订单明细

分享系统处理流程



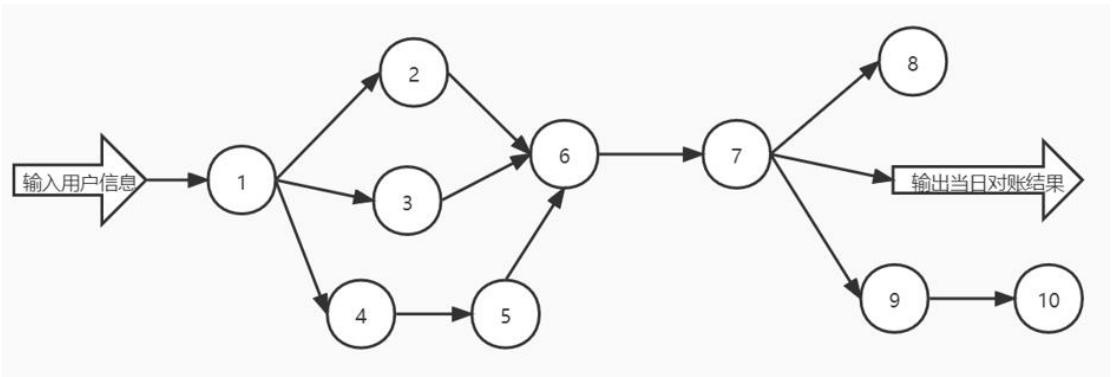
- 1. 获取用户分享的活动信息。
- 2. 获取用户分享的商品信息。
- 3. 获取分享者信息。
- 4. 分析分享的信息，若分享的是优惠活动，则将分享的活动信息和分享者信息传入 5，若分享的是商品，则将分享的商品信息和分享者信息传入 6。
- 5. 分析数据，分享的是优惠活动，根据活动信息和分享者的信息生成分享的 URL。
- 6. 分析数据，分享的是商品，根据商品信息和分享者的信息生成分享的 URL。
- 7. 用户通过分享的 URL 下单时，将对应的订单传入分享系统。
- 8. 生成分享成功的记录，并通过 SQL 语句写入数据库。
- 9. 输出分享成功信息。

售后系统处理流程



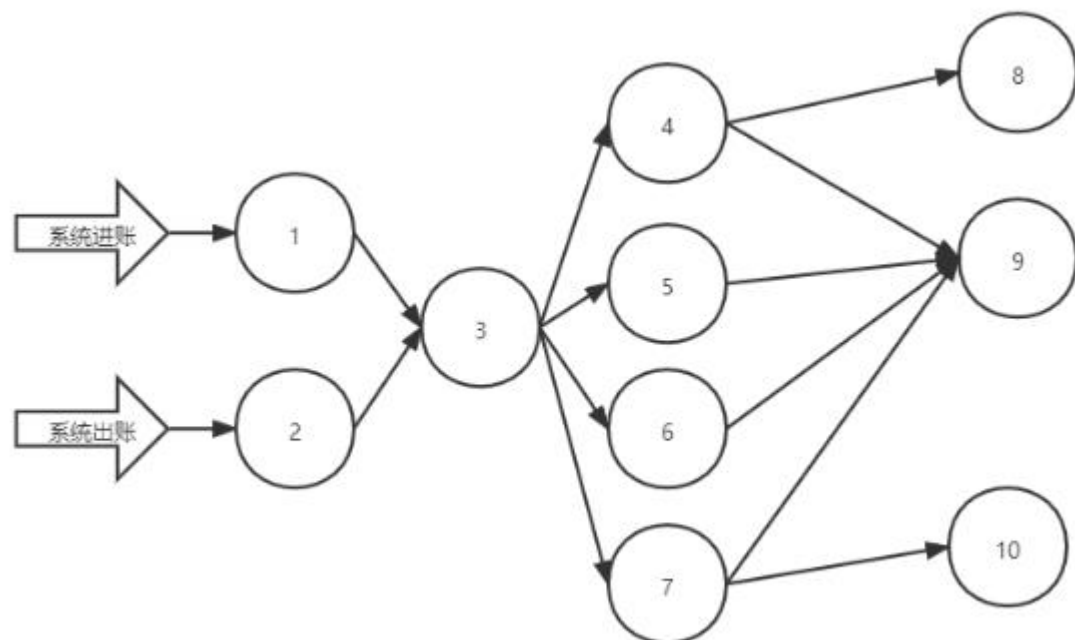
- 1. 获取用户退款请求信息。
- 2. 获取用户退货请求信息。
- 3. 获取用户换货请求信息。
- 4. 根据退款信息获取对应的订单信息。
- 5. 管理员人工审核售后信息,若审核通过则继续，不对则退出。
- 6. 分析请求信息，并获取对应的订单，若是退款请求，则将退款请求信息和订单信息传入 7。若是退货请求，则将退款请求信息和订单信息传入 8，若是换货请求，则将换货请求信息和订单信息传入 9。
- 7. 处理退款，将货款退还给用户。
- 8. 验收买家返回的货品,检验合格转入 10，检验不合格转入 11。
- 9. 验收买家返回的货品,检验合格转入 12，检验不合格转入 13。
- 10. 将货款退还给用户。
- 11. 将商品重新发给用户。
- 12. 将货款退还给用户。
- 13. 将商品重新发给用户。
- 14. 生成显示处理结果。

对账系统：



- 1: 接收对账请求, 将请求时间送 2、3、4
- 2: 根据请求时间, 运行 SQL 语句, 获取当日订单信息, 送 6
- 3: 根据请求时间, 运行 SQL 语句, 获取当日退款信息, 送 6
- 4: 根据请求时间, 调用外部 API, 获得当日第三方平台账单, 送 5
- 5: 根据当日第三方平台账单, 在业务逻辑层计算当日平台收支, 送 6
- 6: 根据订单信息、退款信息、收支信息, 格式化为当日对账信息, 送 7
- 7: 执行 SQL 语句, 查询平台账单, 与对账信息进行对比, 若核对出错, 送 8, 若核对无误, 送 9。同时输出当日对账结果
- 8: 执行 SQL 语句, 记录出错账单, 留待后续人工处理
- 9: 获取核对无误账单, 累计 n 天金额, 在 n 天后送 10
- 10: 根据 n 天累计结果, 将应清算金额送清算模块, 并执行 SQL 语句, 将对应订单标记为已清算, 将应清算金额写入清算模块数据表

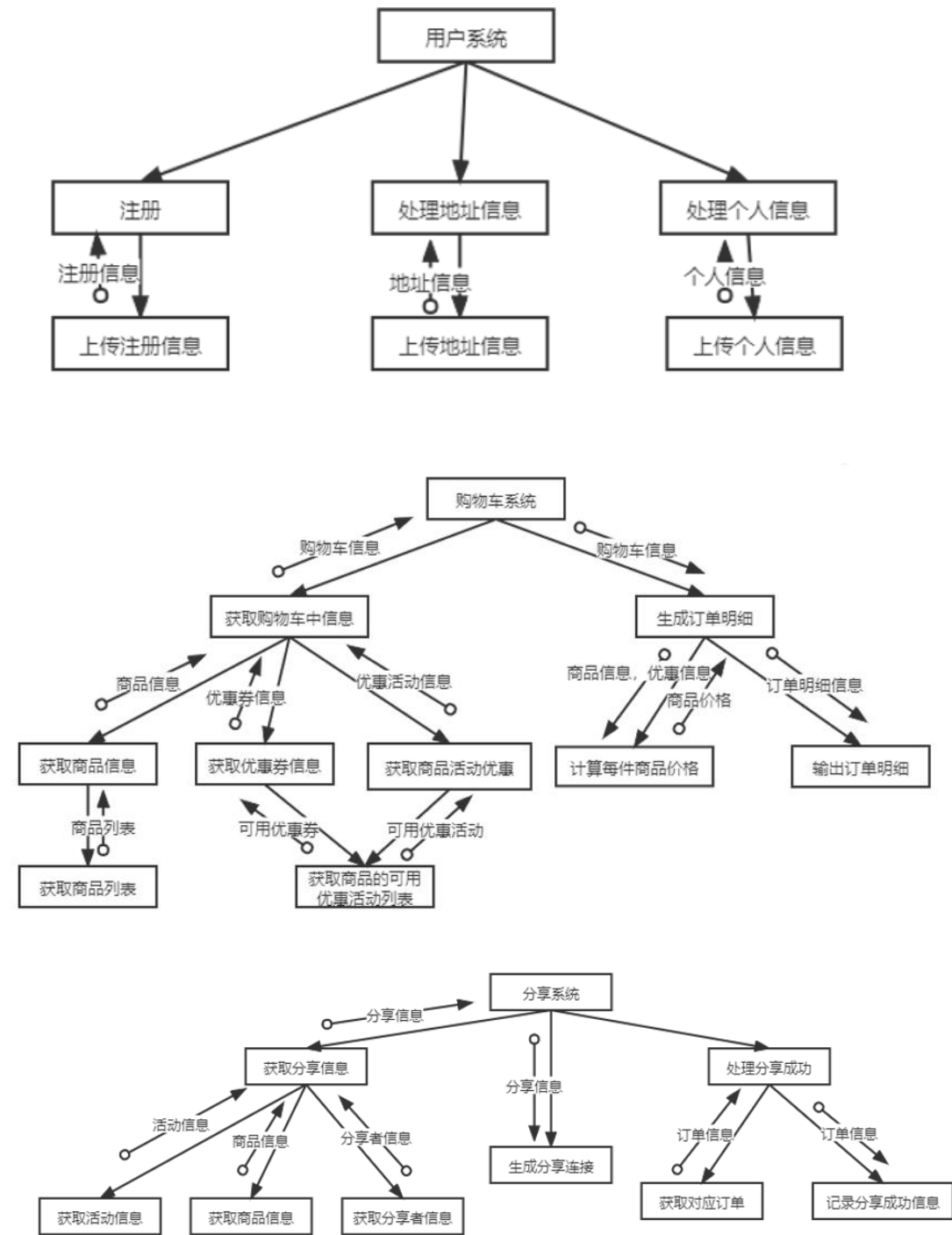
清算系统处理流程

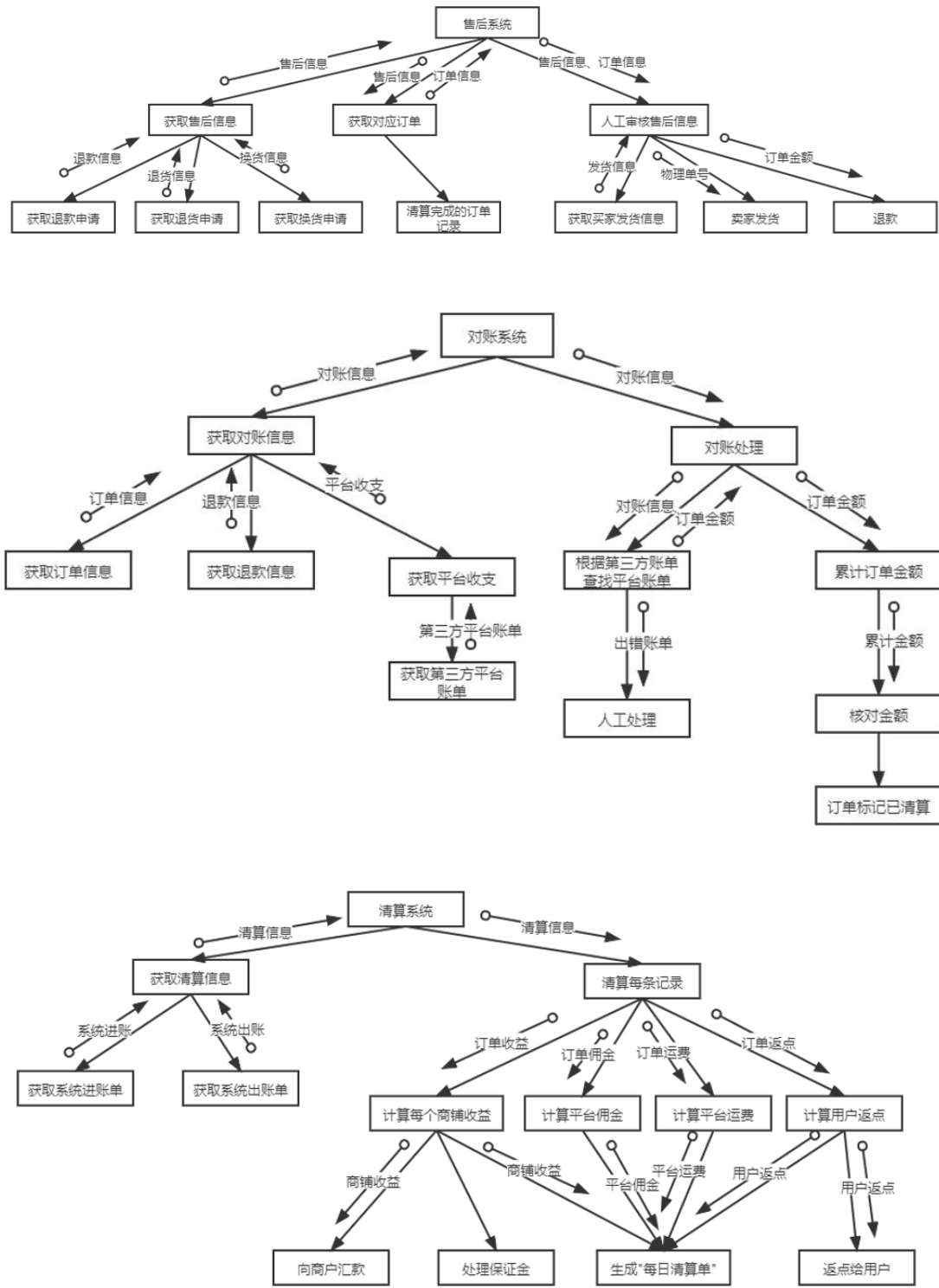


1. 获取系统准备清算进账。
2. 获取系统准备清算出账。
3. 将系统进账与出账整理为一份系统账单, 逐一清算每一条账单记录, 将账单计算所得商铺收益传入 4, 平台佣金传入 5, 平台运费传入 6, 订单的用户返点传入 7.
4. 计算本次清算每个商铺的总收益, 根据商铺的保证金处理商铺的总收益。
5. 计算本次清算的平台佣金收入。
6. 计算本次清算的平台运费收入。
7. 计算本次清算的用户返点。
8. 将收益款汇入对应的商户账户。
9. 以商铺为粒度, 根据商户的收益, 商户支付的平台佣金, 商户支付的平台运费, 商铺给用户的返点生成每日的清算清单。
10. 将返点打给用户。

3.2 总体结构和模块外部设计

3.2.1 总体结构





3.2.2 模块外部设计

模块名	输入参数	输出参数	给系统造成的影响
用户系统	用户名、密码、地址、邮箱、昵称、性别等	登录、修改个人信息、注册等操作的成功	用户验证个人信息后，可以根据自己权

		功与否	限进行相应的操作
分享系统	活动、商品、分享者、被分享者、对应订单等	分享链接、分享状态、分享基本信息等	可以实现买家间的分享，若被分享者购买该商品可以给分享者清算相应积点
购物车系统	用户 token、想删除的货品、想添加的货品、想修改货品的相应参数	购物车中货品信息	要求货品模块对购物车的操作提供接口
售后系统	退款申请、退货申请、换货申请、订单信息、发货（收货）状态、退款金额	售后信息、售后状态、收货（发货）单号、退款金额	要求订单模块为售后操作提供订单，用户系统为售后提供信息等
对账系统	支付单据、订单信息、商铺信息、退款信息、快递费、清算金额、佣金、返点、商铺收入	挂账单（长账、短账）、父订单（进账单、出账单）	为清算模块提供进账出账信息、需要平台提供当日的订单、退款和第三方的平台账单
清算系统	系统进账、系统出账	每日清算结果（包括平台运费、平台佣金、商铺收益、用户返点）	需要对账模块提供进账单和出账单、清算后可以将结算每日的流水、并输出清算结果

3.3 功能分配

各模块功能可参照 3.1 与 3.2 中的功能说明

用户的登录、注册、信息修改由用户系统负责

分享链接生成、分享信息的获取、分享成功记录由分享系统负责

购物车信息获取、订单明细生成由购物车系统负责

不同售后请求获取、售后处理、售后处理结果获取由售后模块负责

获取对账信息、对账处理、生成对账信息、计算清算金额由对账系统负责

获取清算信息、实际清算操作、生成清算信息由清算系统负责

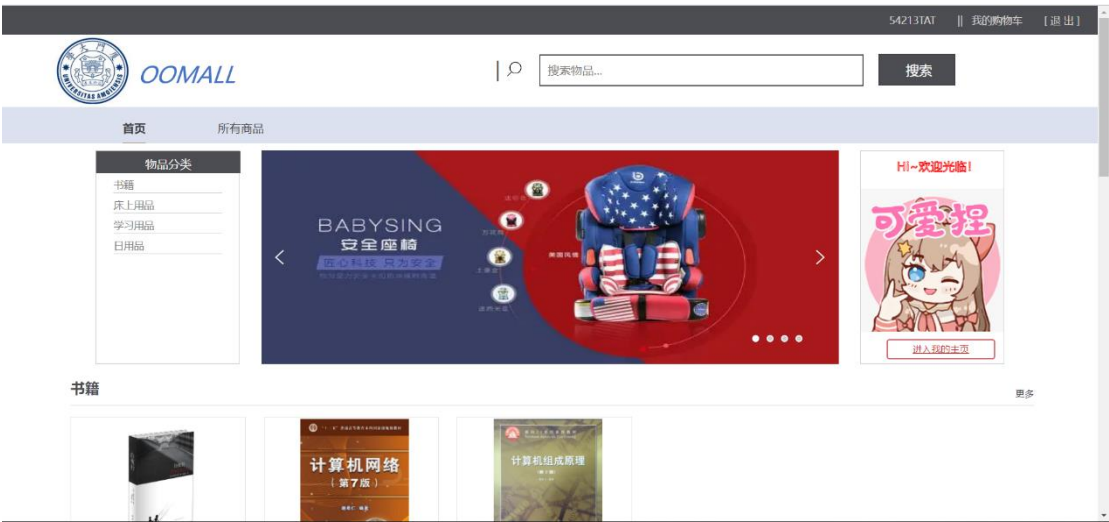
4. 接口设计

4.1 外部接口

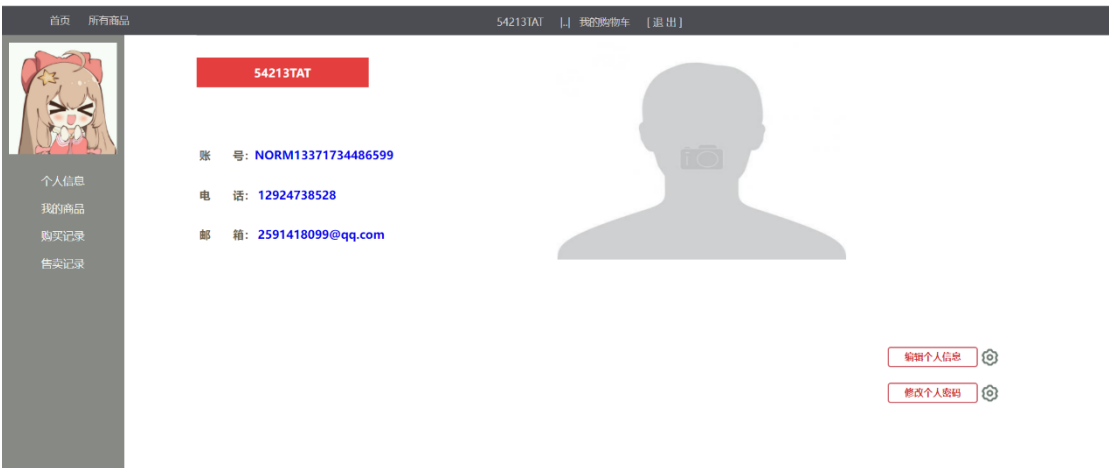
4.1.1 用户界面

本系统无前端界面部分，所有外部请求通过定义的外部 api 对系统进行访问
若要求界面设计，本系统大致界面如下：

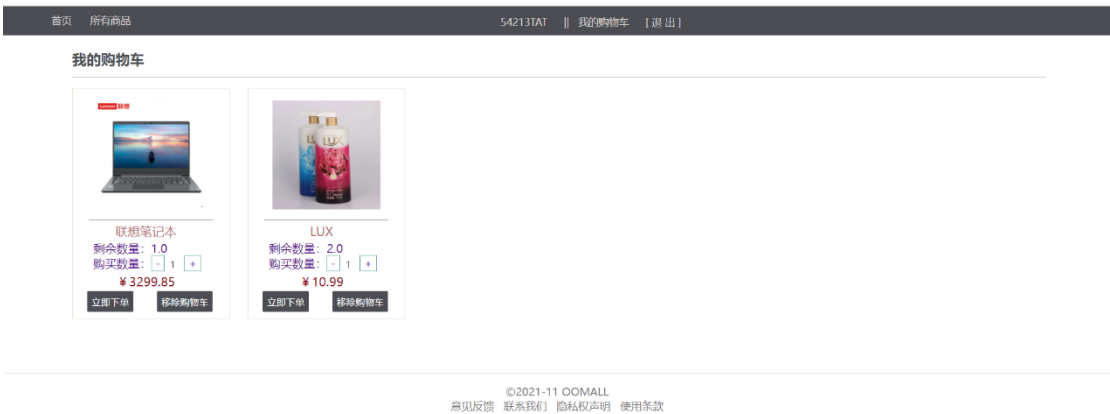
主页：



用户主页：



购物车界面：



4.1.2 软件接口

对于数据库，可通过 MyBatis 提供的对 MySQL 的访问接口，对数据库进行访问；可通过 MySQL 的备份命令，以做到对数据的备份

对于访问记录保存，可通过 logback 提供的输出接口，进行日志保存

对于模块功能调用，可通过输入 api 进行调用，服务端通过 Java 语言对 api 进行解析，以对系统各功能进行调用。Api 需符合 Resultful 风格

对于网络软件接口，使用一种无差错的传输协议，采用滑动窗口方式对数据进行网络传输及接收

4.1.3 硬件接口

在输入部分，仅支持使用键盘输入 api 进行访问，输入需符合 Resultful API 风格，采用 Java 语言对输入进行处理

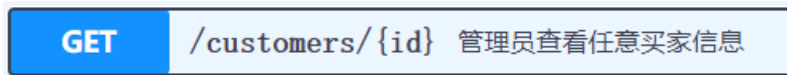
在输出部分，输出信息将输出在控制台和日志中，控制台信息可通过显示器直接查看，日志文件将存储在硬盘中。输出信息多为 json 风格字符串，可采用 Java 语言对输出进行处理

在网络传输部分，为了实现高速传输，将采用高速 ATM (Asynchronous Transfer Mode)，即异步传输模式

4.2 内部接口

内部接口方面，各模块之间通过 api 调用接口、参数传递、返回值的方式进行传递。具体参数与返回值的接口将在数据结构设计的内容中说明。接口传递的信息是以数据接口封装了的数据，以参数传递或返回值的方式在各模块间传递。

一个内部接口示例如下：



通过/customer/{id}进行“查看对应买家信息”功能调用，参数为目标买家 id，返回值为包含 Customer 对象属性的 json 字符串

5. 数据结构设计

5.1 逻辑结构设计

（其中所有表都有 create_by、create_name、modified_by、modified_name、gmt_create、gmt_modified 六个字段，分别描述了创建者用户 id、创建者用户名、修改者用户 id、修改者用户名、创建时间、修改时间，数据类型分别为 BIGINT、VARCHAR(128)、BIGINT、VARCHAR(128)、DATETIME、DATETIME，在表中不在赘述）

商品模块：

Oomall_advance_sale 商城预售基本信息表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	预售信息标识（主键）
Shop_id	BIGINT	NULL	开展预售的商铺 id
Shop_name	VARCHAR(128)	NULL	开展预售的商铺名称
Name	VARCHAR(128)	NULL	预售活动名称
Pay_time	DATETIME	NULL	预售活动时间
Advance_pay_price	BIGINT	NULL	预售价格
State	TINYINT	NULL	预售状态

Oomall_category 商品分类信息表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	类别信息标识（主键）
Name	VARCHAR(128)	NULL	类别名称
Commission_ratio	INT	NULL	平台佣金占比
Pid	BIGINT	NULL	上级分类 id

Oomall_comment 商品评论表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	商品评论信息标识（id）
Product_id	BIGINT	NULL	货品 id
Orderitem_id	BIGINT	NULL	订单明细 id
Type	TINYINT	NULL	评论类型
Content	VARCHAR(500)	NULL	评论内容

四、概要设计说明书

State	TINYINT	NULL	评论状态
--------------	---------	------	------

Oomall_coupon_activity 优惠信息表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	优惠信息标识 (id)
Name	VARCHAR(128)	NULL	优惠名称
Shop_id	BIGINT	NULL	优惠活动店铺 id
Shop_name	VARCHAR(128)	NULL	店铺名称
Coupon_time	DATETIME	NULL	开始领优惠券时间
Begin_time	DATETIME	NULL	活动开始时间
End_time	DATETIME	NULL	活动结束时间
Quantity	INT	NULL	优惠券数目
Quantity_type	TINYINT	NULL	0: 每人数量 1: 总数控制
Valid_term	TINYINT	NULL	优惠券有效时间: 0: 同活动时间; 其他: 从领券开始有效天数
Image_url	VARCHAR(255)	NULL	图片路径
Strategy	VARCHAR(255)	NULL	活动规则 json
State	TINYINT	NULL	活动状态

Oomall_coupon_onsale 优惠活动中的商品信息表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	优惠活动中的商品 (id)
Activity_id	BIGINT	NULL	优惠活动 id
Onsale_id	BIGINT	NULL	货品上架 id (0: 所有有效的上架货品, 不包括团购预售上架货品)

Oomall_freight_model 运费模板信息表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	运费模板信息标识 (id)
Name	VARCHAR(128)	NULL	运费模板名称
Default_model	TINYINT	NULL	默认模板 (0: 非默认; 1: 默认)
Type	TINYINT	NULL	运输类型 (0: 以重量计算; 1: 以数量计算)
Unit	INT	NULL	计重单位 500g

Oomall_goods 商品信息表

列名	数据类型	能否为空	说明
----	------	------	----

Id	BIGINT	NOT NULL	商品信息标识 (id)
Name	VARCHAR(128)	NULL	商品名称
Shop_id	BIGINT	NULL	店铺 id

Oomall_groupon_activity 团购信息表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	团购信息标识 (id)
Name	VARCHAR(128)	NULL	团购活动名称
Shop_id	BIGINT	NULL	团购活动店铺 id
Shop_name	VARCHAR(128)	NULL	店铺名称
Strategy	VARCHAR(500)	NULL	团购活动规则 json
Begin_time	DATETIME	NULL	活动开始时间
End_time	DATETIME	NULL	活动结束时间
State	TINYINT	NULL	活动状态

Oomall_onsale 货品销售价格和数量信息表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	货品销售价格和数量标识 (id)
Shop_id	BIGINT	NULL	店铺 id
Product_id	BIGINT	NULL	货品 id
price	BIGINT	NULL	货品价格
begin_time	DATETIME	NULL	开始销售时间
end_time	DATETIME	NULL	结束销售时间
quantity	INT	NULL	货品数量
type	TINYINT	NULL	0: 无活动; 1: 秒杀; 2: 团购; 3: 预售
activity_id	BIGINT	NULL	活动 id
share_act_id	BIGINT	NULL	活动分享 id
state	BIGINT	NULL	货品状态

Oomall_piece_freight 以数量计算的运输信息表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	运输信息标识 (id)
freight_model_id	BIGINT	NULL	运输模板 id
first_items	INT	NULL	首件数
first_item_piece	BIGINT	NULL	首费
additional_items	INT	NULL	续件数
additional_items_price	BIGINT	NULL	续费

region_id	BIGINT	NULL	抵达地区码
------------------	--------	------	-------

Oomall_product 货品信息表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	商品信息标识 (id)
shop_id	BIGINT	NULL	商铺 id
shop_name	VARCHAR(128)	NULL	店铺名称
goods_id	BIGINT	NULL	商品 id
category_id	BIGINT	NULL	分类 id
freight_id	BIGINT	NULL	运费模板 id
sku_sn	VARCHAR(128)	NULL	序号
name	VARCHAR(128)	NULL	货品名称
original_price	BIGINT	NULL	货品原价
weight	BIGINT	NULL	重量
image_url	VARCHAR(255)	NULL	图片链接
state	TINYINT	NULL	状态
barcode	VARCHAR(128)	NULL	条码
unit	VARCHAR(64)	NULL	计量单位
origin_place	VARCHAR(128)	NULL	货品原产地

oomall_product_draft 未审核的货品信息表

列名	数据类型	能否为空	说明
id	BIGINT	NOT NULL	未审核货品信息 id
shop_id	BIGINT	NULL	商铺 id
product_id	BIGINT	NULL	修改的货品 id, 0 为新货
goods_id	BIGINT	NULL	商品 id
category_id	BIGINT	NULL	分类 id
freight_id	BIGINT	NULL	运费模板 id
sku_sn	VARCHAR(128)	NULL	sku 序号
name	VARCHAR(128)	NULL	商品型号名称
original_price	BIGINT	NULL	该型号原价
weight	BIGINT	NULL	种类
image_url	VARCHAR(255)	NULL	图片链接
barcode	VARCHAR(128)	NULL	条码
unit	VARCHAR(64)	NULL	计量单位
origin_place	VARCHAR(128)	NULL	产地

oomall_region 地区信息表

四、概要设计说明书

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	地区信息标识（id）
pid	BIGINT	NULL	上级地区 id
name	VARCHAR(128)	NULL	地区名称
state	TINYINT	NULL	状态（0：有效；1：停发；2：废弃）

oomall_share_activity 分享活动信息表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	商品信息标识（id）
shop_id	BIGINT	NULL	店铺 id
shop_name	VARCHAR(128)	NULL	店铺名称
name	VARCHAR(128)	NULL	活动名称
begin_time	DATETIME	NULL	开始时间
end_time	DATETIME	NULL	结束时间
strategy	VARCHAR(500)	NULL	活动规则 json
state	TINYINT	NULL	0：草稿；1：上线；2：下线

oomall_shop 商铺信息表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	商铺信息标识（id）
name	VARCHAR(128)	NULL	商铺名称
deposit	BIGINT	NULL	店铺保证金
deposit_threshold	BIGINT	NULL	保证金门槛
state	TINYINT	NULL	店铺状态

oomall_shop_account 店铺账户信息表

列名	数据类型	能否为空	说明
Id	BIGINT	NOT NULL	店铺账户信息标识（id）
shop_id	BIGINT	NULL	店铺 id
type	TINYINT	NULL	支付渠道，0：支付宝；1：微信
account	VARCHAR(256)	NULL	账户号
name	VARCHAR(128)	NULL	账户名称
priority	TINYINT	NULL	汇入优先级

oomall_weight_freight 以重量计算的运输信息表

列名	数据类型	能否为空	说明
----	------	------	----

四、概要设计说明书

Id	BIGINT	NOT NULL	信息标识（id）
freight_model_id	BIGINT	NULL	运输模板 id
frist_weight	BIGINT	NULL	首次重量
frist_weigth_freight	BIGINT	NULL	首重价格
ten_price	BIGINT	NULL	10kg 以下每 0.5kg 价格
fifty_price	BIGINT	NULL	50kg 以下每 0.5kg 价格
hundred_price	BIGINT	NULL	100kg 以下每 0.5kg 价格
trihun_price	BIGINT	NULL	300kg 以下每 0.5kg 价格
above_price	BIGINT	NULL	300kg 以上每 0.5kg 价格
region_id	BIGINT	NULL	抵达地区 id

其他模块：

oomall_address 地址表

列名	数据类型	能否为空	说明
id	BIGINT	NOT NULL	主键
customer_id	BIGINT	NULL	买家 id
region_id	BIGINT	NULL	地区码
detail	VARCHAR(500)	NULL	详细地址
consignee	VARCHAR(128)	NULL	联系人
mobile	VARCHAR(128)	NULL	电话
default	TINYINT	NULL	设为默认

oomall_aftersale_service 售后服务表

列名	数据类型	能否为空	说明
id	BIGINT	NOT NULL	主键
order_id	BIGINT	NULL	售后订单 id
order_item_id	BIGINT	NULL	订单明细 id
customer_id	BIGINT	NULL	顾客 id
shop_id	BIGINT	NULL	店铺 id
service_sn	VARCHAR(128)	NULL	售后单序号
type	TINYINT	NULL	服务类型
reason	VARCHAR(500)	NULL	原因
conclusion	VARCHAR(500)	NULL	处理意见
price	BIGINT	NULL	金额
quantity	BIGINT	NULL	数量
region_id	BIGINT	NULL	地区码
detail	VARCHAR(255)	NULL	详细地址
consignee	VARCHAR(128)	NULL	联系人
mobile	VARCHAR(255)	NULL	电话
customer_log_sn	VARCHAR(128)	NULL	寄出运单号

四、概要设计说明书

shop_log_sn	VARCHAR(128)	NULL	寄回运单号
state	TINYINT	NULL	状态
be_deleted	TINYINT	NULL	买家逻辑删除

oomall_coupon 优惠券表

列名	数据类型	能否为空	说明
id	BIGINT	NOT NULL	主键
coupon_sn	VARCHAR(128)	NULL	优惠券编号
name	VARCHAR(128)	NULL	优惠券名称
customer_id	BIGINT	NULL	顾客 id
activity_id	BIGINT	NULL	优惠活动 id
begin_time	DATETIME	NULL	优惠券开始时间
end_time	DATETIME	NULL	优惠券过期时间
state	TINYINT	NULL	优惠券状态

oomall_customer 顾客表

列名	数据类型	能否为空	说明
id	BIGINT	NOT NULL	主键
user_name	VARCHAR(128)	NULL	用户名
password	VARCHAR(255)	NULL	密码
real_name	VARCHAR(128)	NULL	真实姓名
point	BIGINT	NULL	返点
state	TINYINT	NULL	状态
email	VARCHAR(255)	NULL	邮箱
mobile	VARCHAR(255)	NULL	电话
be_deleted	TINYINT	NULL	逻辑删除

oomall_expenditure_item 出账表

列名	数据类型	能否为空	说明
id	BIGINT	NOT NULL	主键
liquid_id	BIGINT	NULL	清算单 id
refund_id	BIGINT	NULL	退款单据 id
shop_id	BIGINT	NULL	商铺 id
revenue_id	BIGINT	NULL	进账清算 id
name	VARCHAR(128)	NULL	名称
amount	BIGINT	NULL	清算金额
express_fee	BIGINT	NULL	快递费
commission	BIGINT	NULL	佣金
point	BIGINT	NULL	分享返点
sharer_id	BIGINT	NULL	分享者 id
shop_revenue	BIGINT	NULL	商铺收入

oomall_liquidation 清算单表

列名	数据类型	能否为空	说明
id	BIGINT	NOT NULL	主键
shop_id	BIGINT	NULL	商铺 id
liquid_date	DATETIME	NULL	清算日期
express_fee	BIGINT	NULL	快递费
commission	BIGINT	NULL	佣金
shop_revenue	BIGINT	NULL	商铺收入
point	BIGINT	NULL	分享返点
state	TINYINT	NULL	商铺收入状态

oomall_revenue_item 进账表

列名	数据类型	能否为空	说明
id	BIGINT	NOT NULL	主键
liquid_id	BIGINT	NULL	清算单 id
payment_id	BIGINT	NULL	支付单据 id
shop_id	BIGINT	NULL	商铺 id
order_id	BIGINT	NULL	订单 id
orderitem_id	BIGINT	NULL	订单明细 id
name	VARCHAR(128)	NULL	名称
amount	BIGINT	NULL	清算金额
express_fee	BIGINT	NULL	快递费
commission	BIGINT	NULL	佣金
point	BIGINT	NULL	分享返点
sharer_id	BIGINT	NULL	分享者 id
shop_revenue	BIGINT	NULL	商铺收入

oomall_share 分享表

列名	数据类型	能否为空	说明
id	BIGINT	NOT NULL	主键
sharer_id	BIGINT	NULL	分享者 id
share_act_id	BIGINT	NULL	分享活动 id
product_id	BIGINT	NULL	货品 id
quantity	BIGINT	NULL	成功返点的商品件数
state	TINYINT	NULL	状态

oomall_successful_share 分享成功表

列名	数据类型	能否为空	说明
id	BIGINT	NOT NULL	主键
share_id	BIGINT	NULL	分享 id
product_id	BIGINT	NULL	货品 id
customer_id	BIGINT	NULL	点击者 id
state	TINYINT	NULL	状态

oomall_shopping_cart 购物车表

列名	数据类型	能否为空	说明
id	BIGINT	NOT NULL	主键
customer_id	BIGINT	NULL	顾客 id
product_id	BIGINT	NULL	加入货品 id
quantity	BIGINT	NULL	加入的数量
price	TINYINT	NULL	加入时单价

5.2 物理结构设计

系统中，针对每一个数据库中的表，建立 po 对象（persistent object），从数据取出的数据直接构建成为 po 对象

业务逻辑涉及到 po 对象时，需要将 po 对象转换为 bo 对象（business object），bo 对象中存储业务需要的属性。业务逻辑只能调用 bo 对象

向数据库中写入数据时，需要将写入对象转换为 po 对象，再进行写操作

5.3 数据结构与程序的关系

系统的数据结构由标准数据库语言 SQL 生成

例如：

oomall_address 地址表的建表语句如下：

```
CREATE TABLE `oomall_address` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `customer_id` bigint(20) DEFAULT NULL,  
  `region_id` bigint(20) DEFAULT NULL,  
  `detail` varchar(500) DEFAULT NULL,  
  `consignee` varchar(128) DEFAULT NULL,  
  `mobile` varchar(128) DEFAULT NULL,  
  `default` tinyint(4) DEFAULT NULL,  
  `creator_id` bigint(20) DEFAULT NULL,  
  `creator_name` varchar(128) DEFAULT NULL,  
  `modifier_id` bigint(20) DEFAULT NULL,  
  `modifier_name` varchar(128) DEFAULT NULL,  
  `gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `gmt_modified` datetime DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

在系统中，PO 对象与数据库表转换所需的代码由 MyBatis 自动生成
例如：

oomall_goods 所对应的 GoodsPo 对象代码如下:

```
public class GoodsPo {
    private Long id;
    private String name;
    private Long shopId;
    private Long creatorId;
    private String creatorName;
    private Long modifierId;
    private String modifierName;
    private LocalDateTime gmtCreate;
    private LocalDateTime gmtModified;
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name == null ? null : name.trim();
    }
    public Long getShopId() {
        return shopId;
    }
    public void setShopId(Long shopId) {
        this.shopId = shopId;
    }
    public Long getCreatorId() {
        return creatorId;
    }
    public void setCreatorId(Long creatorId) {
        this.creatorId = creatorId;
    }
    public String getCreatorName() {
        return creatorName;
    }
    public void setCreatorName(String creatorName) {
        this.creatorName = creatorName == null ? null : creatorName.trim();
    }
    public Long getModifierId() {
        return modifierId;
    }
}
```

```
public void setModifierId(Long modifierId) {
    this.modifierId = modifierId;
}
public String getModifierName() {
    return modifierName;
}
public void setModifierName(String modifierName) {
    this.modifierName = modifierName == null ? null : modifierName.trim();
}
public LocalDateTime getGmtCreate() {
    return gmtCreate;
}
public void setGmtCreate(LocalDateTime gmtCreate) {
    this.gmtCreate = gmtCreate;
}
public LocalDateTime getGmtModified() {
    return gmtModified;
}
public void setGmtModified(LocalDateTime gmtModified) {
    this.gmtModified = gmtModified;
}
}
```

6. 运行设计

6.1 运行模块的组合

客户机程序在有输入时启动接收数据模块，通过个模块之间的调用，读入并对输入进行格式化。在接收数据模块得到充分的数据时，将调用网络传输模块，将数据通过网络送到服务器，并等待接收服务器返回的信息。接收到返回信息后随机调用接收到返回信息后随机调用数据输出模块，对信息进行处理，产生相应的输出。

服务器程序的接收网络数据模块必须始终处于活动状态。接收到数据后，调用数据处理/查询模块对数据库进行访问，完成后调用网络发送模块，将信息返回客户机。

6.2 运行控制

运行控制将严格按照各模块建函数调用关系来实现。在各事务中心模块中，需对运行控制进行正确的判断，选择正确的运行控制路径。

在网络传输方面，客户机在发送数据后，将等待服务器的确认收到信号，收到后，再次等待服务器发送回答数据，然后对数据进行确认。服务器在接到数据后发送确认信号，在对数据处理、访问数据库后，将返回信息送回客户机，并等待确认。

6.3 运行时间

系统的运行时间基本能够到达同时涌入 1000 请求而不崩溃

7. 出错处理设计

7.1 出错输出信息

当某请求出错时，服务器后端返回对应的错误码和错误描述
对应的出错信息将输出在服务端的控制台与日志文件中，并且给予客户端用户以对应提示
自定义的返回码原则上不与 Http 自带返回码冲突，如 403、404
本系统目前设计以下错误码：

1. 系统级别错误
(服务端崩溃都会返回此错误码)

错误码	对应代码	对应描述
500	INTERNAL_SERVER_ERR	服务器内部错误

2. 因登录导致的错误

错误码	对应代码	对应描述
501	AUTH_INVALID_JWT	JWT 不合法
502	AUTH_JWT_EXPIRED	JWT 过期

3. 其余自定义错误码

错误码	对应代码	对应描述
503	FIELD_NOTVALID	字段不合法
504	RESOURCE_ID_NOTEXIST	操作的资源 id 不存在
505	RESOURCE_ID_OUTSCOPE	操作的资源 id 不是自己的对象
506	FILE_NO_WRITE_PERMISSION	目录文件夹没有写入的权限
507	STATE_NOT_ALLOW	当前状态禁止此操作
508	IMG_FORMAT_ERROR	图片格式不正确
509	IMG_SIZE_EXCEED	图片大小超限
510	PARAMETER_MISSED	缺少必要参数
511	RESOURCE_FALSIFY	信息签名不正确
601	ADDRESS_OUTLIMIT	达到地址簿上限
605	SHAREACT_CONFLICT	分享活动时段冲突
606	ORDERITEM_NOTSHARED	订单明细无分享记录
608	ADVERTISEMENT_STATENOTALLOW	广告状态禁止
801	ORDER_STATENOTALLOW	订单状态禁止
804	REFUND_MORE	退款金额超过支付金额
805	REGION_NOT_REACH	该地区不可达
900	GOODS_STOCK_NOTENOUGH	商品规格库存不够

901	GOODS_CATEGORY_SAME	类目名称已存在
902	GOODS_PRICE_CONFLICT	商品销售时间冲突
903	GOODS_CATEGORY_NOTALLOW	不允许加入到一级分类
909	COUPON_NOTBEGIN	未到优惠券领取时间
910	COUPON_FINISH	优惠券领罄
911	COUPON_END	优惠券活动终止
921	CATEALTER_INVALID	对 SPU 类别操作无效
924	ACTIVITY_NOTFOUND	无符合条件的优惠活动
925	SHOP_NOTOPERABLE	不可对该商铺进行操作
941	COMMENT_EXISTED	该订单条目已评论
947	LATE_BEGINTIME	开始时间不能晚于结束时间
948	ACT_LATE_PAYTIME	尾款支付时间晚于活动结束时间
949	ACT_EARLY_PAYTIME	尾款支付时间早于活动开始时间
950	COUPON_LATE_COUPONTIME	优惠券领卷时间晚于活动开始时间
966	SHOP_HASDEPOSIT	店铺仍有保证金未结算
967	SHOP_CATEGORY_NOTPERMIT	不允许增加新的下级分类
968	SHOP_NOT_RECON	店铺尚有支付未清算完毕
969	SHOP_USER_HASSHOP	用户已经有店铺
970	COMMENT_USER_NOORDER	用户没有购买此商品
995	FREIGHT_REGIONOBSOLETE	地区已废弃
996	FREIGHT_WRONGTYPE	该运费模板类型与内容不符
997	FREIGHT_REGIONEXIST	该运费模板中该地区已经定义
998	FREIGHT_NOTDELETED	存在上架销售商品，不能删除运费模板
999	FREIGHT_REGIONSAME	运费模板中该地区已经定义

7.2 出错处理对策

1. 所有服务端应保持持续开机，以防止停电或电压不稳定造成数据丢失。若服务器意外断电，可根据 SQL SERVER 的日志文件，使用 ROLLBACK 机制进行恢复
2. 在网络传输方面，保持网关与外部网络的通信畅通，可考虑为其建立一条成本较低的后备网络，以保证在主网络意外中断时的通信
3. 在硬件方面，应保证服务器数量不少于已划分模块数量，以确保每个模块能相对独立的运行环境，减少相互干扰。同时，设置备用服务器，以保证主服务器崩溃时可使用备用服务器继续运行系统
4. 在返回错误码时，将出错信息返回给客户端、输出在服务端控制台以便查看，同时记录错误码、出错信息、出错时间到日志文件中，以便后续查看

8.安全保密设计

本系统的安全保密主要有三大部分组成——权限系统、数据加密、网关设计

1. 权限系统

权限系统的设计目标如下：

- 支持 JWT 登录

采用 Session 功能记录登录状态会导致安全漏洞，且用户量过大会导致系统崩溃，因此我们采用 jwt 方式，支持令牌

令牌方式是在用户登录之后颁发令牌，从令牌可以获取用户信息。令牌可实现防伪造，且不可修改

jwt 的方式另一个好处是服务端无需留存任何东西，减轻服务器压力

- 支持唯一登录，即一个账号只能在一台设备上登录
- 支持基于角色的权限管理
- 支持用户的权限代理
- 支持用户的正常和失败的访问日志
- 支持多个网关并行工作
- 支持关键数据的加密以及防篡改
- 支持性能的度量

2. 数据加密工作

针对安全问题，我们主要采用加密和签名的方式，前者可防止信息泄露，后者可防止篡改

- 数据加密

对于系统中的关键信息以及隐私信息，需要进行加密处理，主要在于用户模块的用户信息，若其余模块有加密需求，可采用相同的方法

例：

因用户登录需要使用电话号码、邮箱、密码，所以对于用户表中的 `mobile`、`email`、`password` 字段进行 AES 加密

- 签名

签名即摘要签名，选取我们需要保护的一个或多个信息，对其进行操作（如相加、插入某些字符）后得到结果值，对结果值进行加密，存储为 `signature` 字符串

每当读取对应数据时，都进行一次签名计算，并进行结果比对，如果一致即可认为是正确操作，否则为非法操作

例：

对于优惠活动的开始日期和结束日期需要进行防篡改保护，所以我们可以对优惠活动表中的 `begin_time` 和 `end_time` 进行摘要签名，以此完成防篡改目的

3. 网关设计

网关，又称 `gateway`，外部请求只能访问网关，由网关进行相关权限校验以检查该请求是否合格，检查合格方可对系统进行访问，否则直接拒绝该请求。

对于所有请求，首先检查其是否为内部请求（内部接口）。因为内部接口是系统内部各模块进行跨模块进行访问所预留的接口，其 `api` 都带有 `internal` 字样，所有内部接口无法开放给外部使用。因此，如果网关检测到有外部请求试图访问内部接口，则直接拒绝该请求，防止外部请求直接调用内部 `api`

其次，要对请求进行权限校验，如果当前请求没有对应接口（`api`）的访问权限（如是否登录），则直接拒绝该请求。权限由专门的 `Privilege` 模块负责，权限信息存储在一个独立的 `Redis` 服务器中，对于系统内部的相互调用请求是否具有对应权限，以及网关的校验权限，主要有 `Redis` 服务器负责

通过网关的请求，网关查看路由表，将请求分发到对应模块所在服务器上进行处理

9.维护设计

【说明为方便维护工作的设施，如维护模块等。】

维护设计主要分为两部分：代码维护、数据库维护

代码维护：

1. 为方便后续代码的修改，设计中尽量保持开闭原则，以减少修改造成的影响，同时减轻修改代码所需的工作量
2. 代码采用 Git 方式进行保存，建立统一的 git 仓库，编程人员可建立分支 clone 到本地，修改完毕后发起 merge request，通过审核后并入主仓库。主仓库保留修改记录，方便回滚与历史查看

数据库维护：

1. 使用 MySQL 的事务功能，结合回滚机制，以便在意外情况发生时可进行数据恢复
2. 定期导出 MySQL 数据进行备份，防止数据丢失
3. 为数据库设立专门的服务器，关注数据库的日志文件，出错后可查看日志进行错误定位
4. 设立合理的隔离级别，注意防止数据库死锁问题
5. 设立合理的事务属性，合理使用分布式缓存 Redis，保持数据库数据的一致性