厦門大學

面向对象分析与设计课程

顾客订单模块详细设计

组　　别　3-4 组

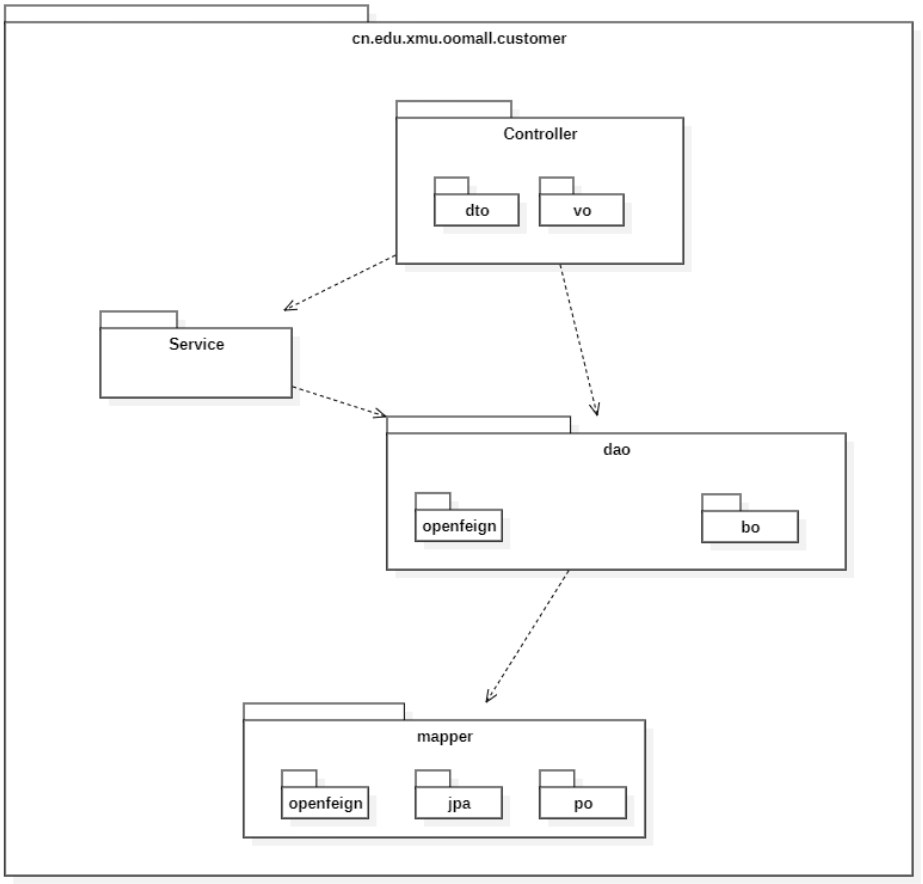
组　　员　黄子安 崔方博 范周喆 侯妤欣 徐森彬

2023　年　**12**　月　**16**　日

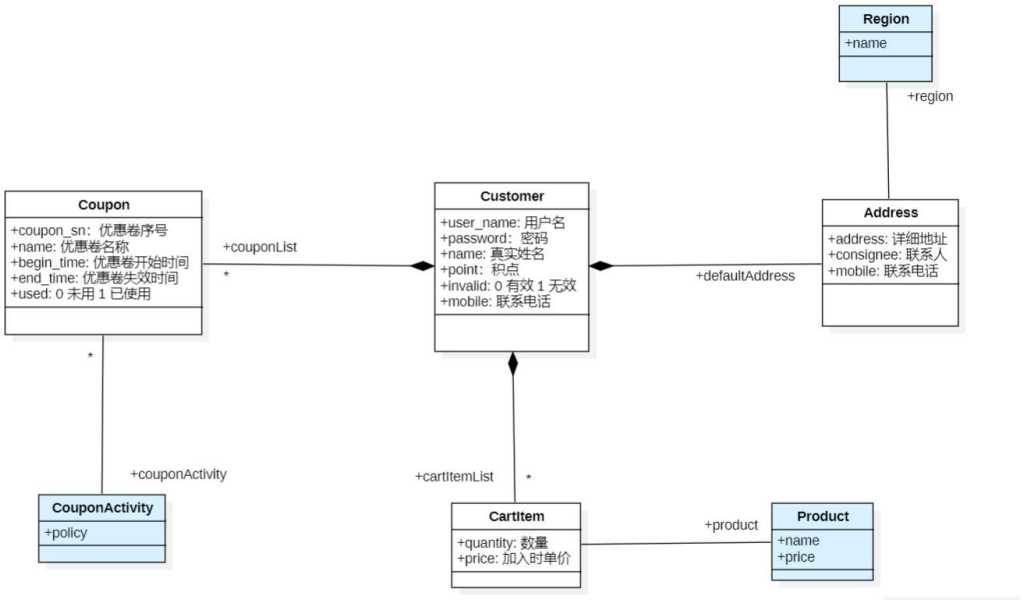
目录

1、顾客模块

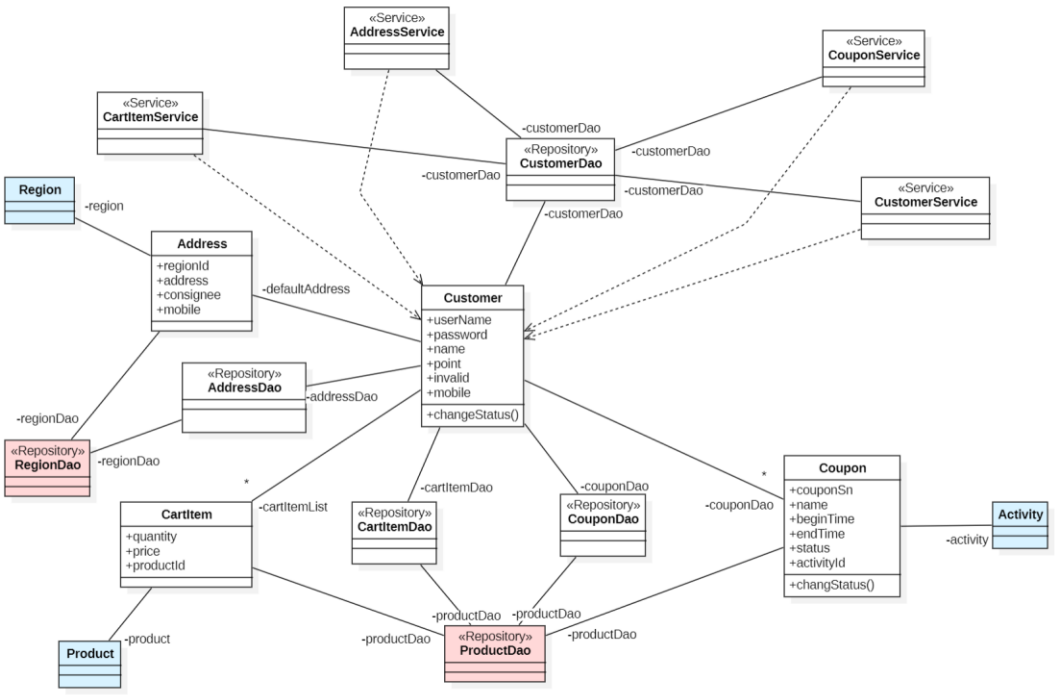
1.1 包图


cn.edu.xmu.oomall.customer
Controller
dto
vo
Service
dao
openfeign
bo
mapper
openfeign
jpa
po

1.2 对象模型


Region
+name
+region
Coupon
+coupon_sn：优惠卷序号
+name: 优惠卷名称
+begin_time: 优惠卷开始时间
+end_time: 优惠卷失效时间
+used: 0 未用 1 已使用
+couponList
*
Customer
+user_name: 用户名
+password：密码
+name: 真实姓名
+point：积点
+invalid: 0 有效 1 无效
+mobile: 联系电话
+defaultAddress
Address
+address: 详细地址
+consignee: 联系人
+mobile: 联系电话
*
+couponActivity
CouponActivity
+policy
+cartItemList
*
CartItem
+quantity: 数量
+price: 加入时单价
+product
Product
+name
+price

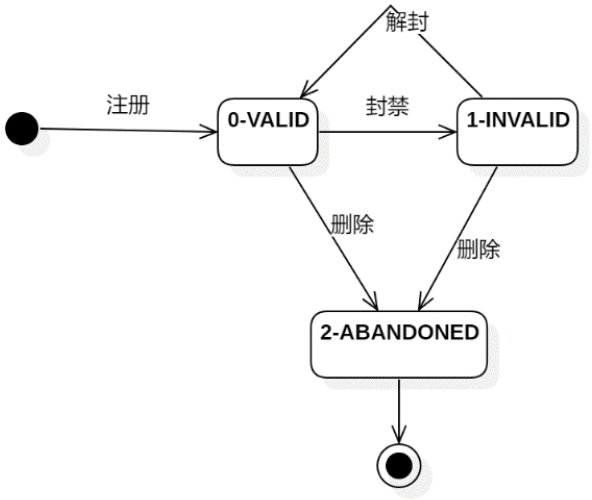
1.3 类图



顾客因为顾客和优惠券、收货地址、购物车明细三者都是组合关系，因此大量使用了创建者模式，所以需要在 customer 对象中注入 AddressDao、CouponDao、CartItemDao，这里粉色的 Dao 表示 openFeign 用于访问其他模块对应的 Dao 层，之后蓝色的三个部分是其余模块的对象，ProductDao、RegionDao 注入到对应的 CartItem、Coupon、Address 中用于实现在获取对应对象（Region、Product、Activity）时能发送请求获取对象，即将 CartItem、Coupon、Address 变为满血模型

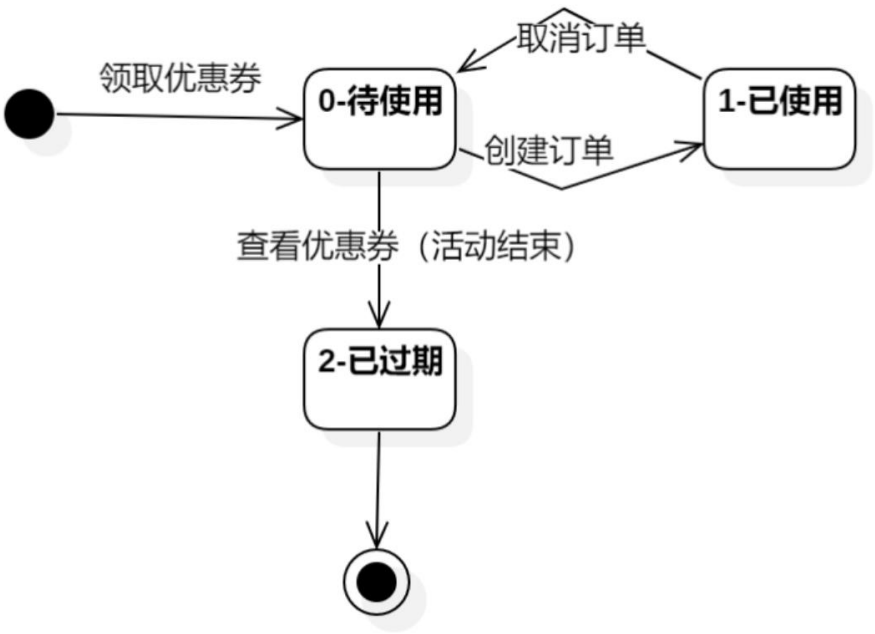
1.4 状态图

顾客状态机



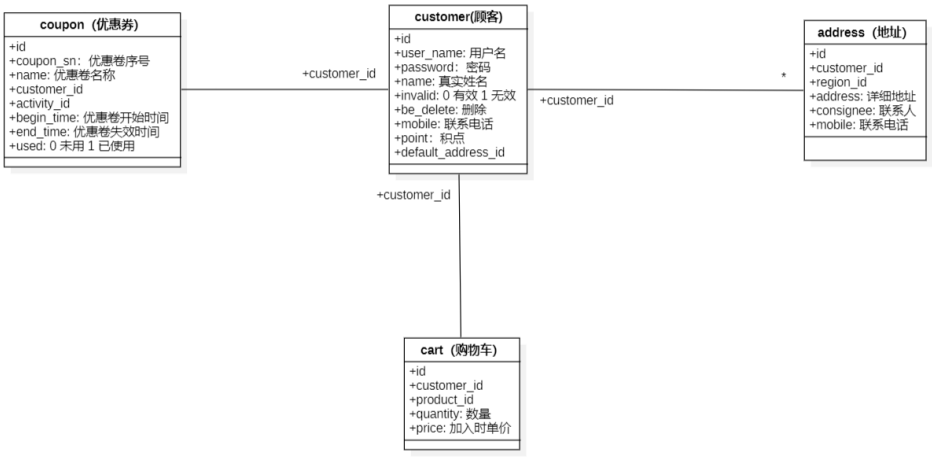
事件	API
注册	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/customer/registerUser
封禁	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/customer/banUser
解禁	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/customer/releaseUser
删除 *	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/customer/delUser

优惠券状态机



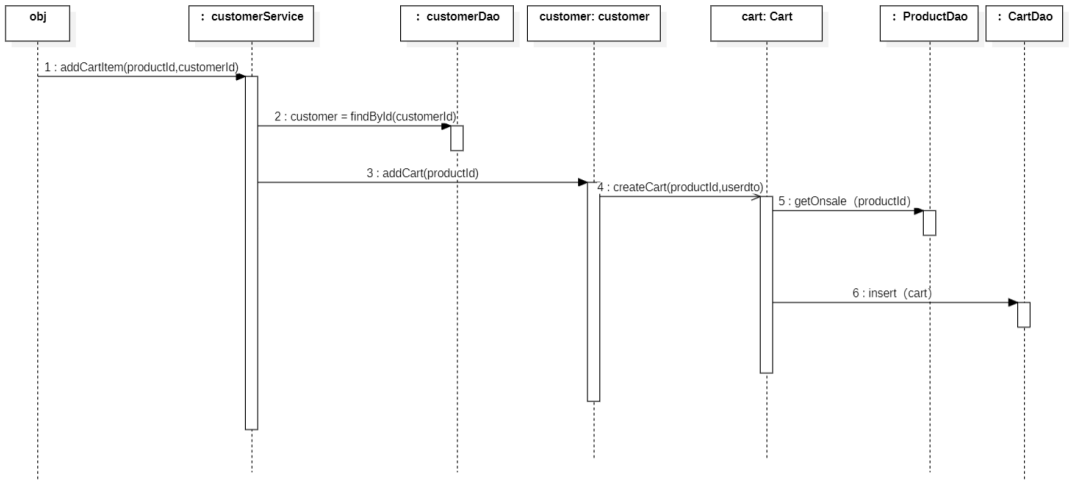
事件	API
领取优惠券	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/customer/post_couponactivities__id__coupons
创建订单*	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/customerPostNewNormalOrder
取消订单	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/delete_shops__shopId__orders__id_（商家） https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/delete_orders__id_（顾客）
查看优惠券列表	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/customer/showCoupons

1.5 数据库



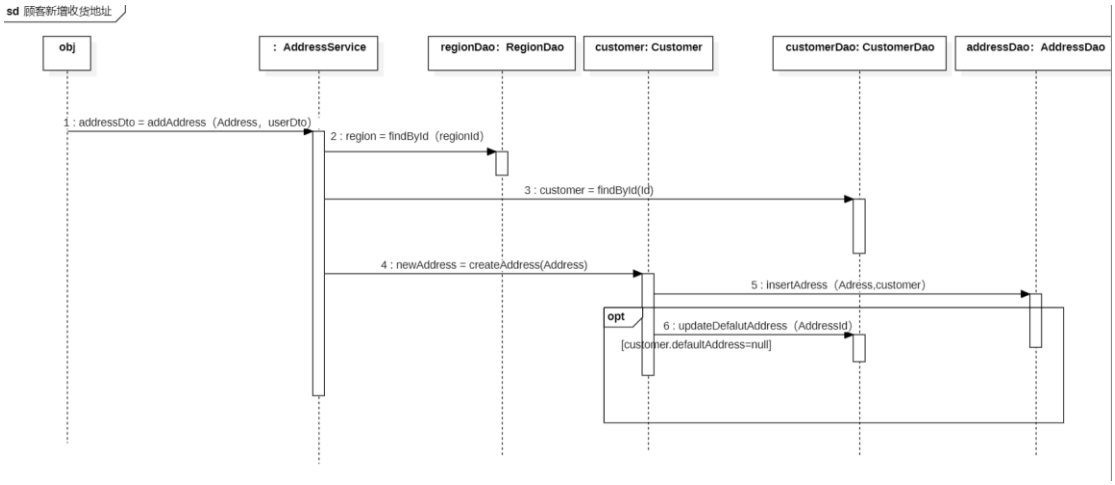
1.6 程序逻辑

顾客将商品加入购物车

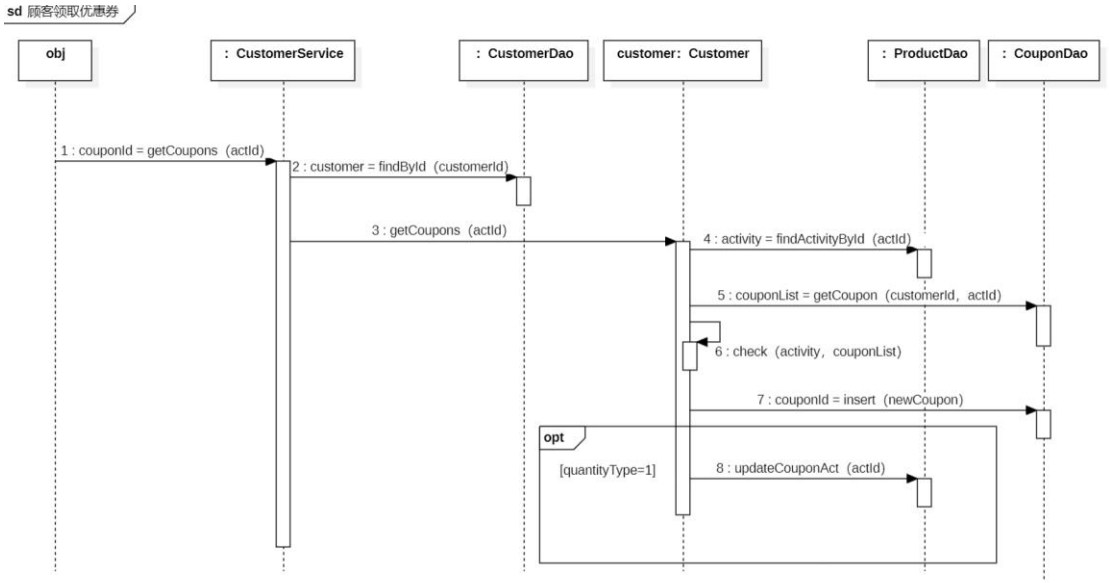


顾客和购物车是组合关系，因此使用创建者模式将创建购物车明细的职责分配给顾客

顾客新增收货地址

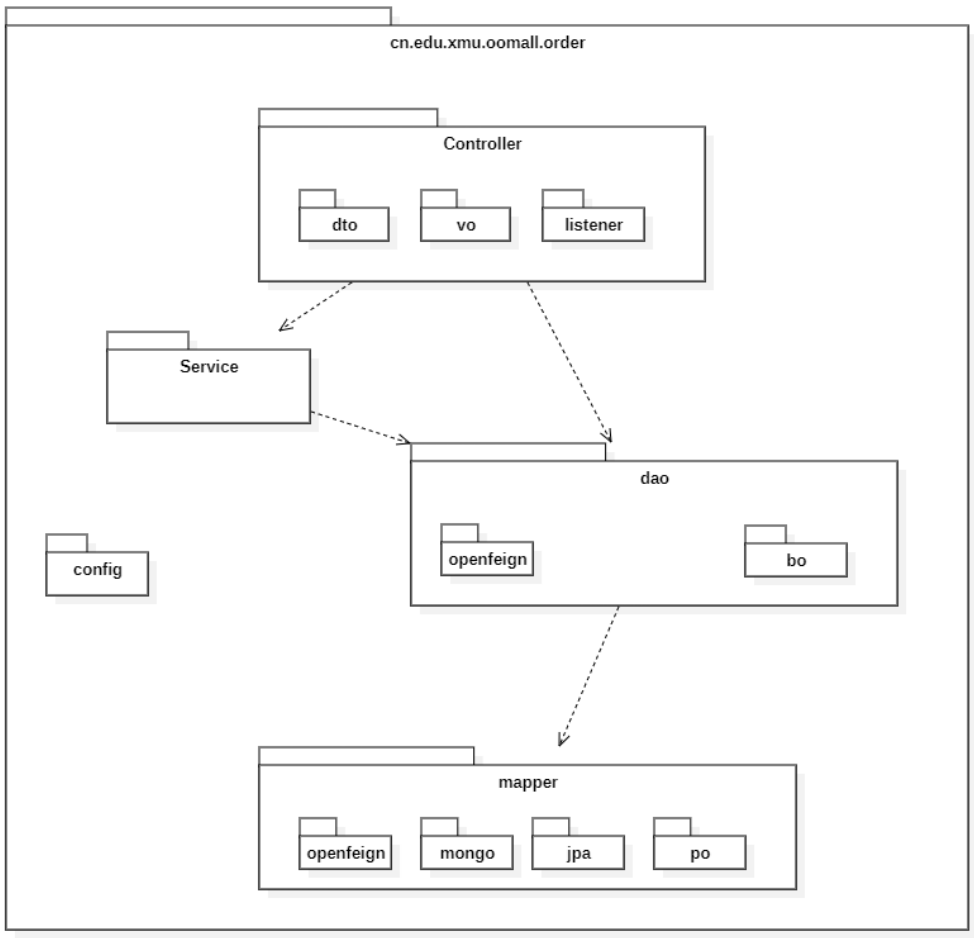


顾客领取优惠券



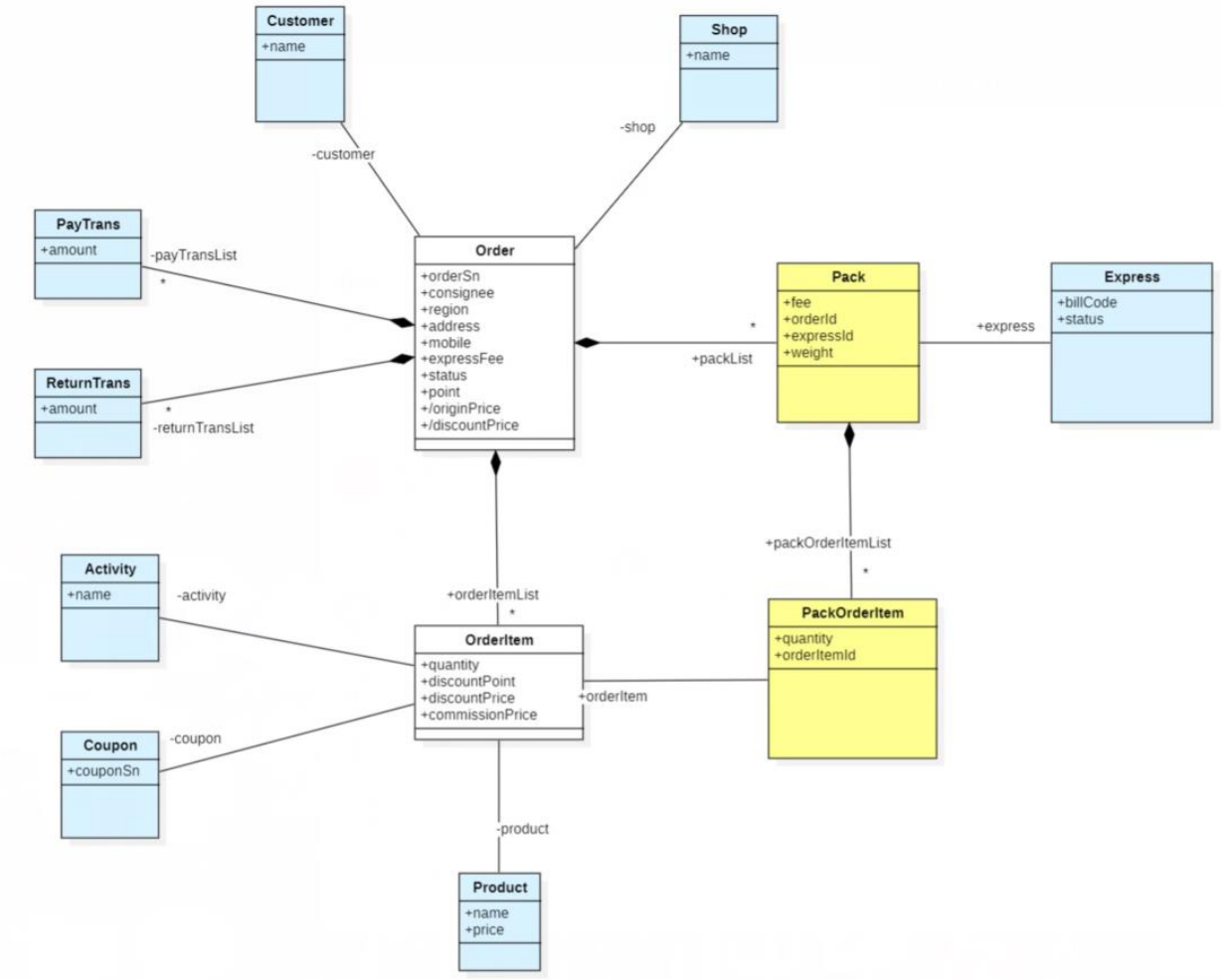
2、订单模块

2.1 包图



listerner：支付订单需要使用 RocketMQ，所以需要对应的包来存放相关类，另外 listerner 位于控制器层，因为根据六边形体系结构与外部系统交互的消息队列服务器应该位于系统的最外层，这里不是 Mapper 层因为不能向上传递订阅的消息，因此只能放在控制器；另外 RocketMQ 实际上调用了 Service 的方法，从功能业务看也应该位于控制器层
config：该模块使用了 RocketMQ 和 Mongo，所以需要存放对应的配置类

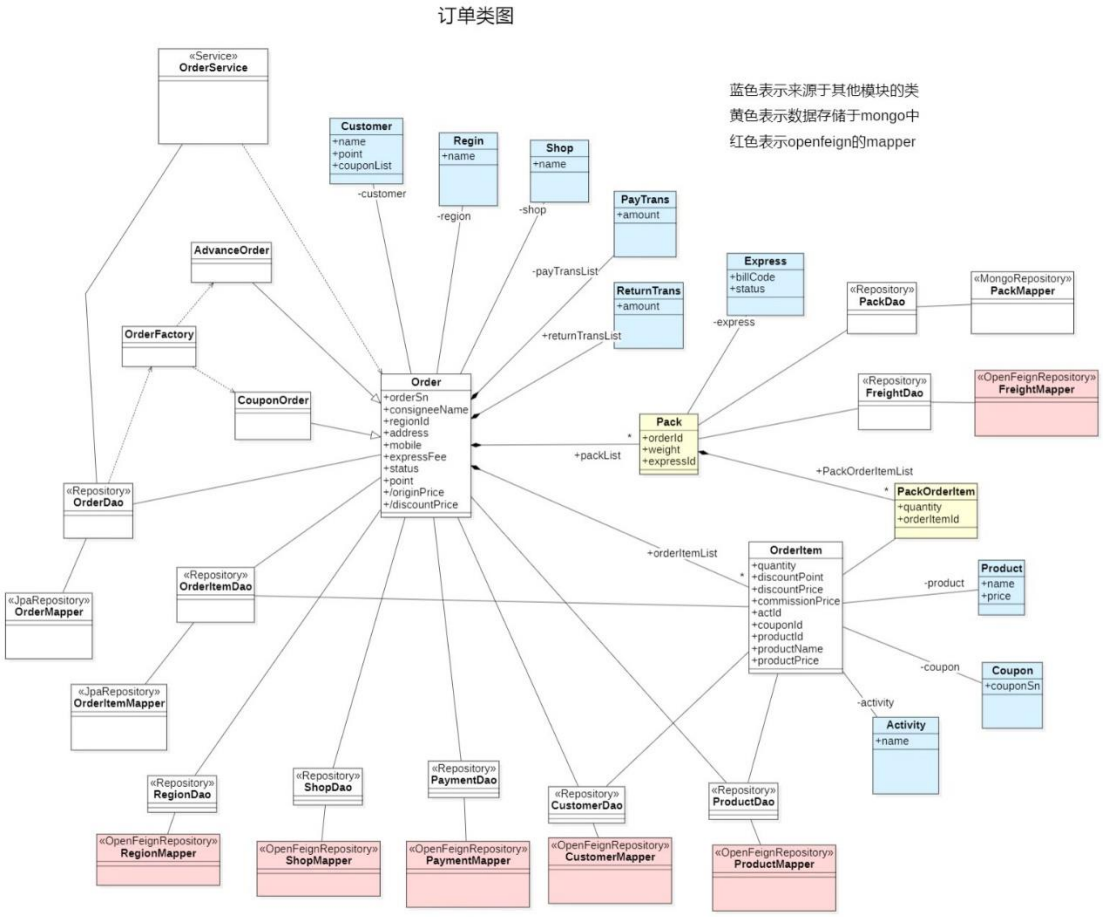
2.2 对象模型



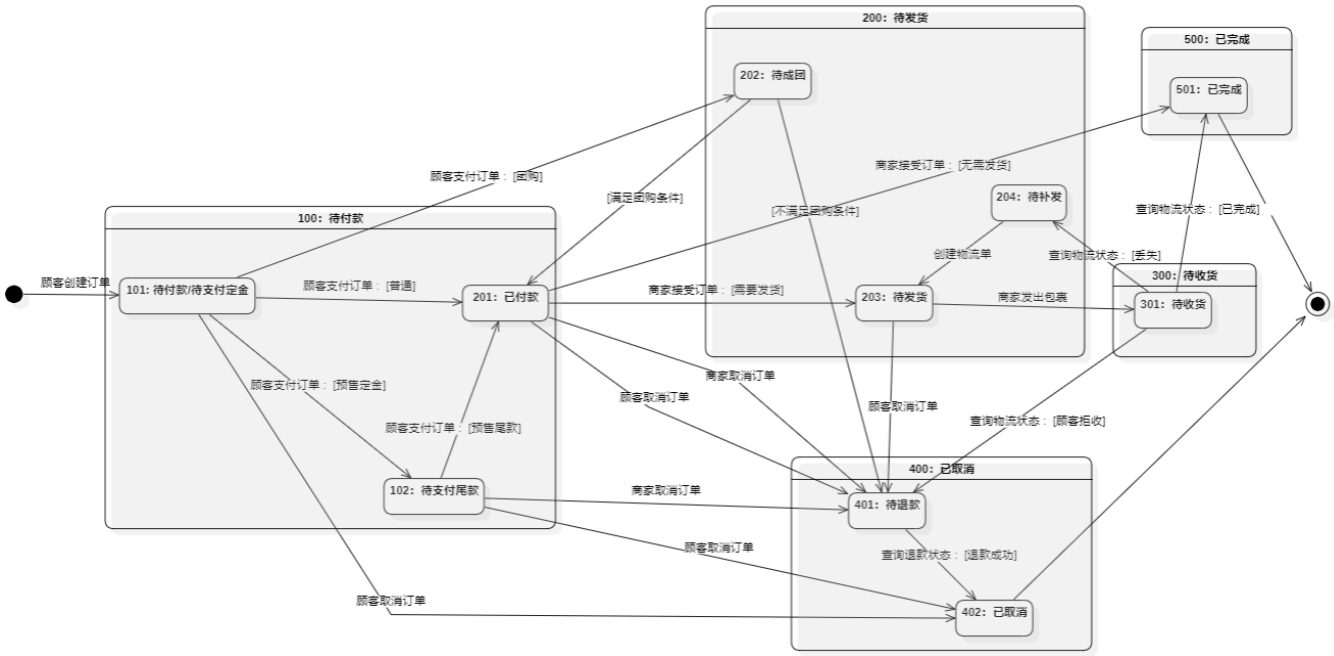
对象模型修改：

- 增加两个新的对象 Pack 和 PackOrderItem
- 增加 Pack：原本的对象模型无法实现分包裹发货的功能，因为没有一个对象用于记录包裹里面包含的物品信息
- 增加 PackOrderItem：PackOrderItem 是对应特定的 OrderItem 的，一个包裹中未必存入了 OrderItem 中所有数量的商品（即可能 100 件的商品被分成了两个 50 件在不同包裹）
- 改进之后的好处：Order 与 Express 由原本的一对多的关系转化为了 Order 与 Pack 一对多关系，Pack 与 Express 一对一关系，可以知道一个包裹中有多少数量的商品，利于实现多包裹发货和包裹丢失补发

2.3 类图

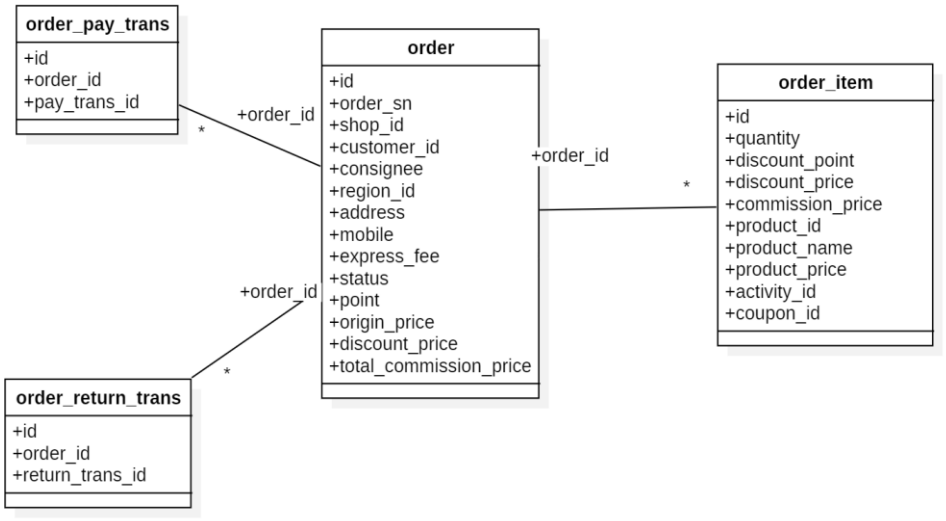

订单类图
蓝色表示来源于其他模块的类
黄色表示数据存储于mongo中
红色表示openfeign的mapper
«Service»
OrderService
Customer
+name
+point
+couponList
Regin
+name
Shop
+name
PayTrans
+amount
ReturnTrans
+amount
Express
+billCode
+status
«Repository»
PackDao
«MongoRepository»
PackMapper
AdvanceOrder
OrderFactory
CouponOrder
Order
+orderSn
+consigneeName
+regionId
+address
+mobile
+expressFee
+status
+point
+/originPrice
+/discountPrice
Pack
+orderId
+weight
+expressId
«Repository»
FreightDao
«OpenFeignRepository»
FreightMapper
«Repository»
OrderDao
PackOrderItem
+quantity
+orderItemId
«JpaRepository»
OrderMapper
«Repository»
OrderItemDao
OrderItem
+quantity
+discountPoint
+discountPrice
+commissionPrice
+actId
+couponId
+productId
+productName
+productPrice
Product
+name
+price
Coupon
+couponSn
Activity
+name
«JpaRepository»
OrderItemMapper
«Repository»
RegionDao
«Repository»
ShopDao
«Repository»
PaymentDao
«Repository»
CustomerDao
«Repository»
ProductDao
«OpenFeignRepository»
RegionMapper
«OpenFeignRepository»
ShopMapper
«OpenFeignRepository»
PaymentMapper
«OpenFeignRepository»
CustomerMapper
«OpenFeignRepository»
ProductMapper
-customer
-region
-shop
-payTransList
+returnTransList
-express
+packList
+PackOrderItemList
+orderItemList
-product
-coupon
-activity

2.4 状态图



事件	API
顾客创建订单*	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/customerPostNewNormalOrder
顾客支付订单	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/payOrder
顾客取消订单	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/delete_orders__id_
商家取消订单	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/delete_shops__shopId__orders__id_
商家接受订单*	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/postFreights
商家发出包裹*	https://app.swaggerhub.com/apis/hyxHome/oomall/1.3.1#/order/putFreights

2.5 数据库

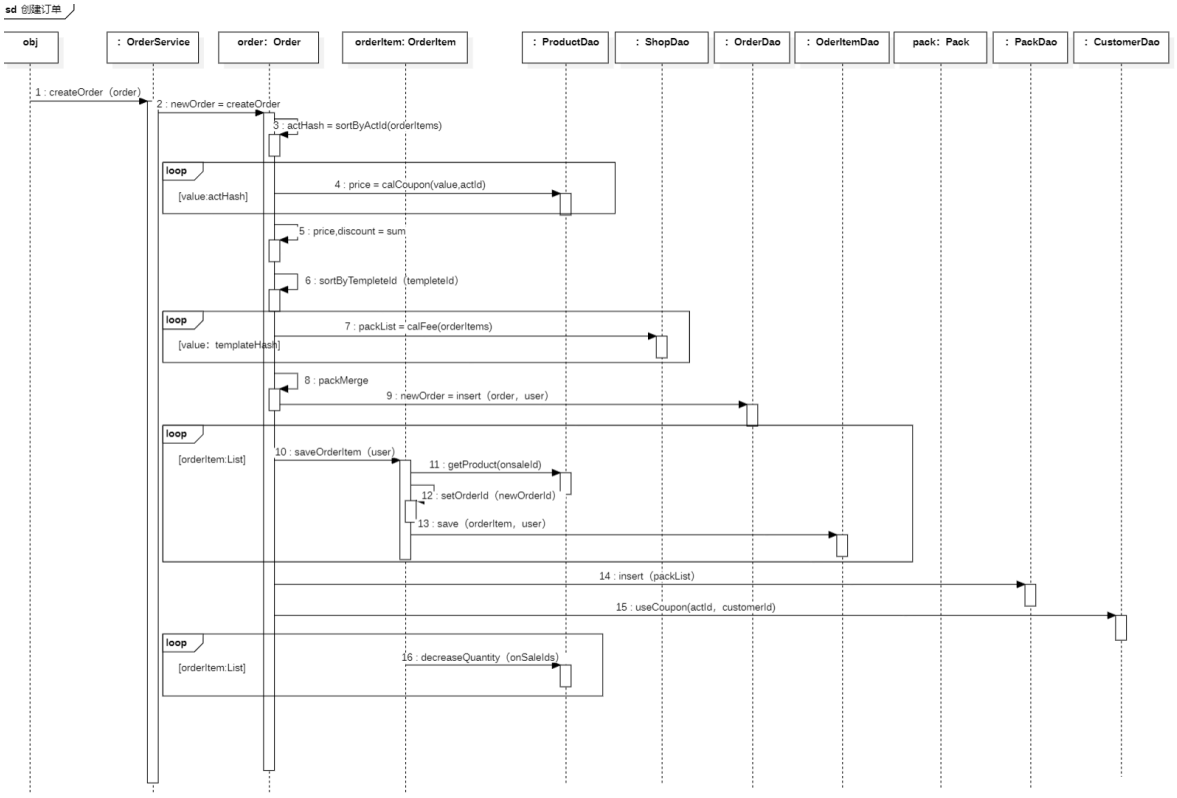


根据对象模型的颜色可以知道新增的两个对象放在了 MongoDB 之中，主要考虑以下几点

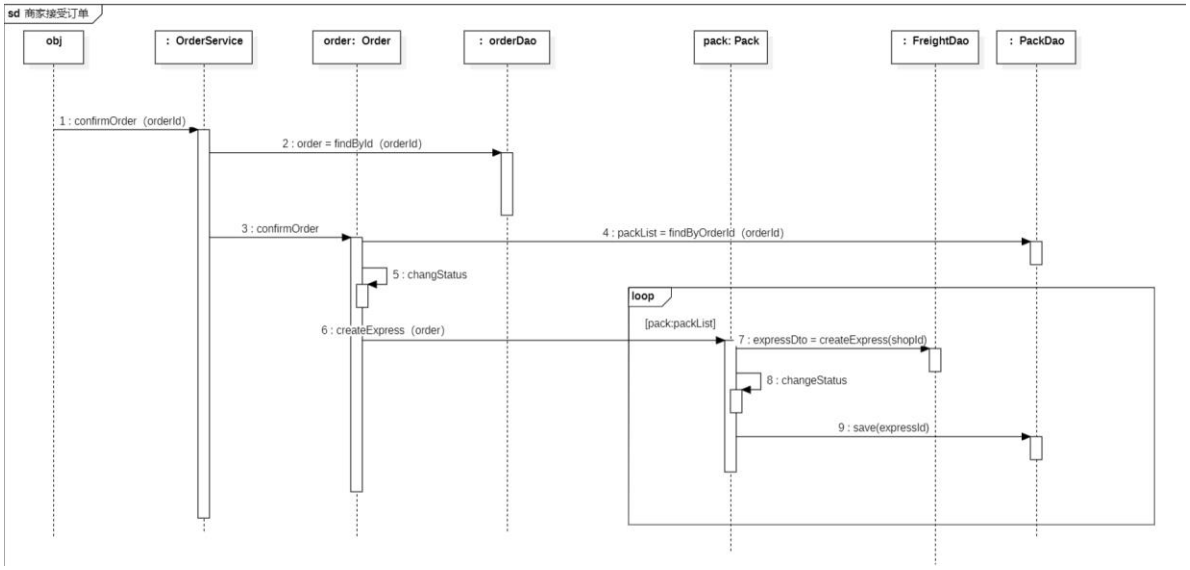
- Pack 和 PackOrderItem 与其他对象的关系相对复杂，且后续使用（主要在创建运单的时候）Pack 的时候也会需要使用 PackOrderItem（即两个对象一般是共同使用的），使用 MongoDB 的对象存储有助于简化存取流程
- Order 还是存在 MySQL：订单模块需要用到两个包裹对象的场合比较有限，不太希望在每次从数据库中读取 Order 的时候都把一连串的包裹全找出来，必然会影响性能，因此 Order 放在 MySQL 和包裹考虑分开存放

2.6 程序逻辑

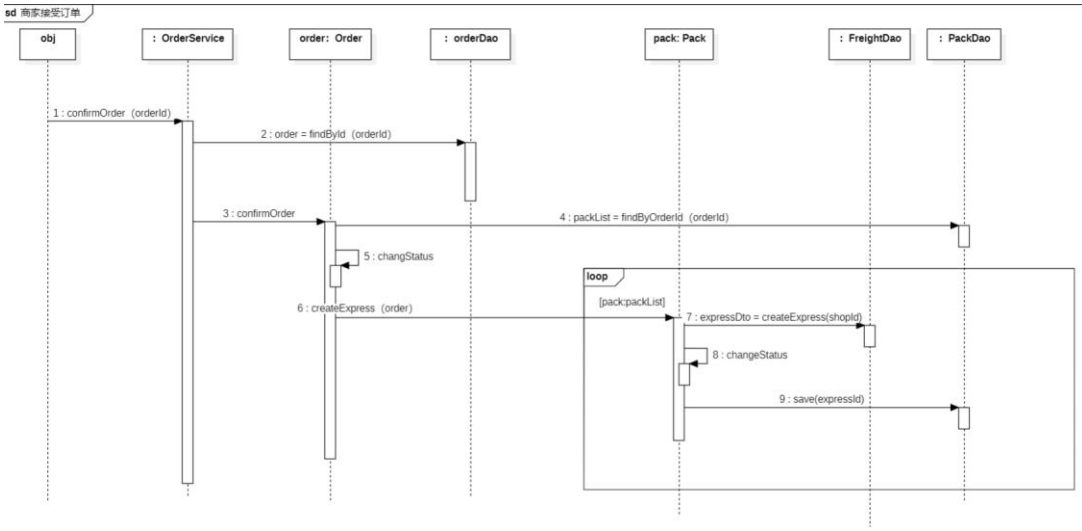
创建订单



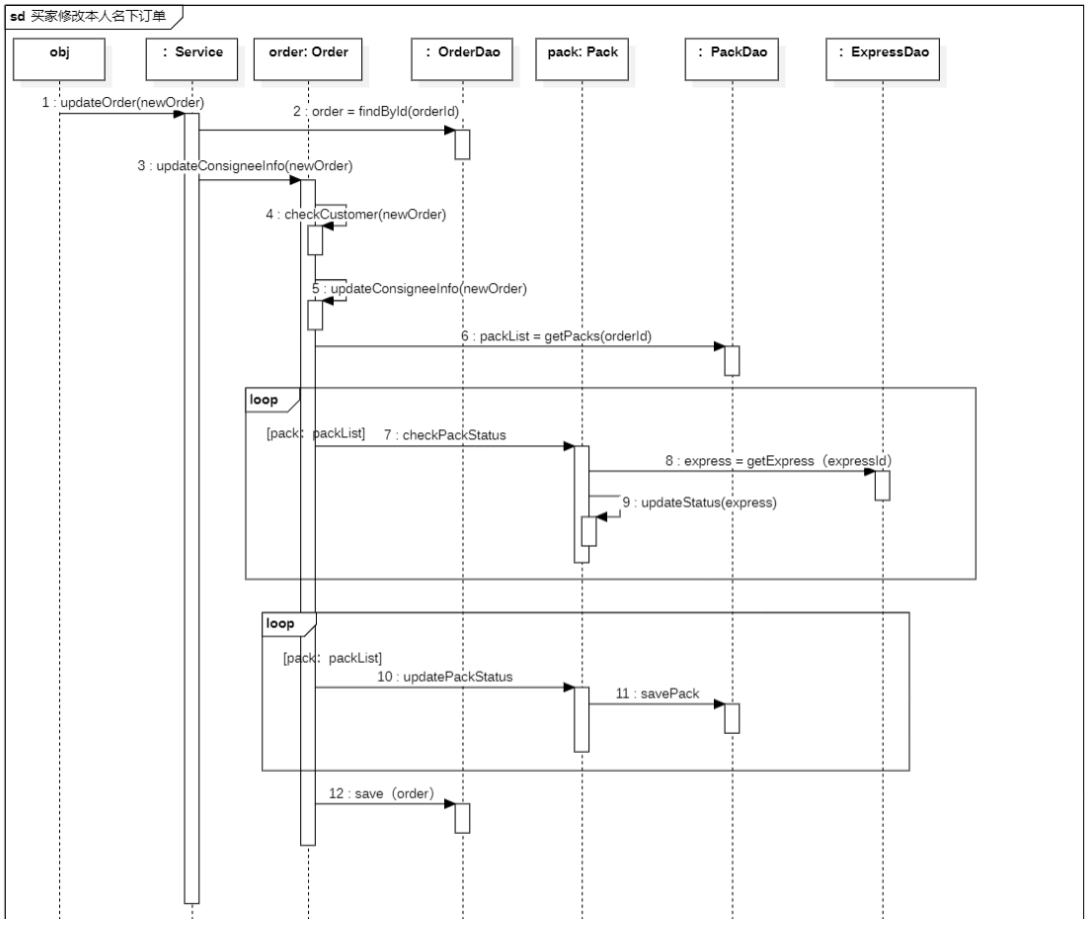
支付订单



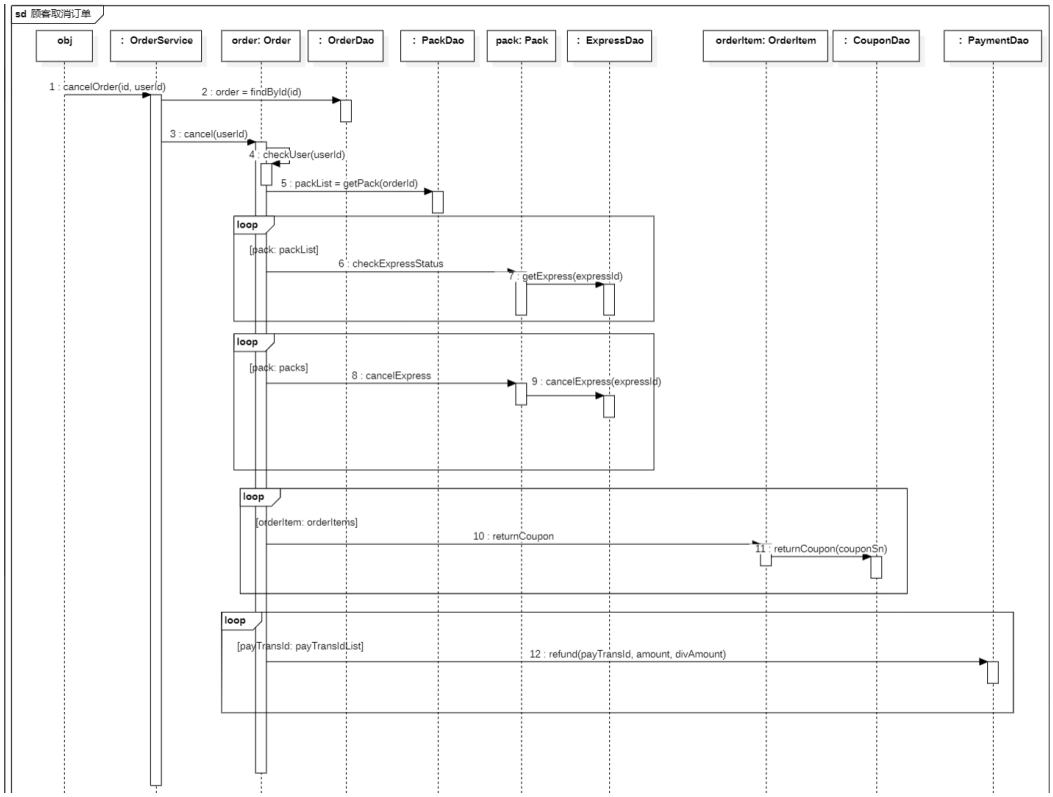
商家接受订单



买家修改订单



买家取消订单



商家取消订单