大学闲置物品交易平台

项目文档



2051970 赖羿龙 2052225 张勤杭 2052727吕天成

1953729 吴浩泽 2053049 黄远宏

2022年春季系统分析与设计

[设计模型 3](#_Toc19874)

[简介和概述 3](#_Toc22175)

[架构分析图 4](#_Toc17679)

[子系统和接口 5](#_Toc26105)

[用例实现 7](#_Toc25918)

[用例实现——[检索商品] 赖羿龙 7](#_Toc3416)

[用例实现——[购买商品] 吕天成 10](#_Toc32637)

[用例设计——用户中心 张勤杭 13](#_Toc5410)

[用例实现——出售商品 黄远宏 18](#_Toc19879)

[安全性实现 21](#_Toc16042)

[团队分工 24](#_Toc18204)

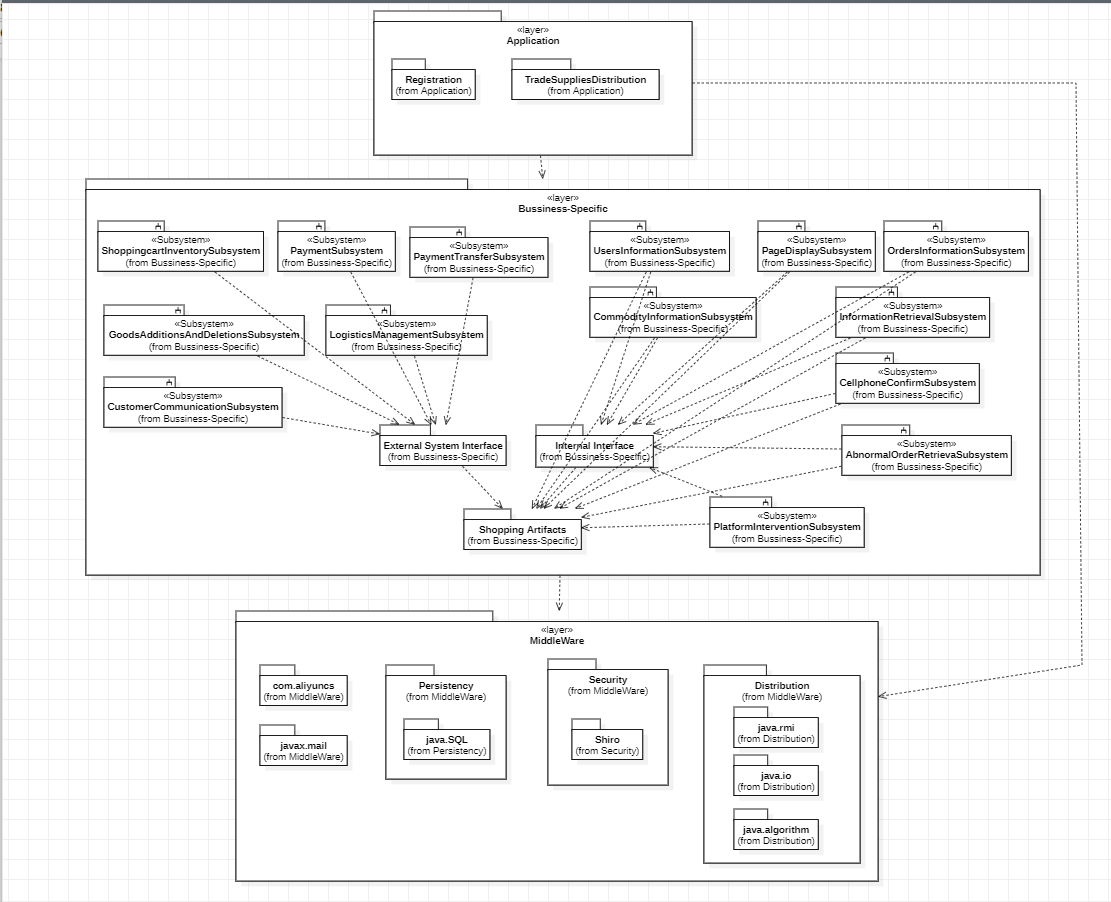
[用例实现： 24](#_Toc22976)

# 设计模型

## 简介和概述

在之前的需求分析、用例实现、分析机制之后，按照老师的建议，我们进一步对项目架构里的子系统接口以及部分机制的实现进行了分析和讨论，按照用例分工，每个人都对自己用例需要的子系统和接口进行了分析和描述，包括系统内部接口和利用现有平台（如阿里云）的第三方借接口。并绘制了更新后的用例类图、时序图以及部分子系统实现的类图和时序图，体现了和第三方信息服务平台间的通信之间的接口规范。最终实现了设计模型中的详细的用例实现。

## 架构分析图



## 子系统和接口

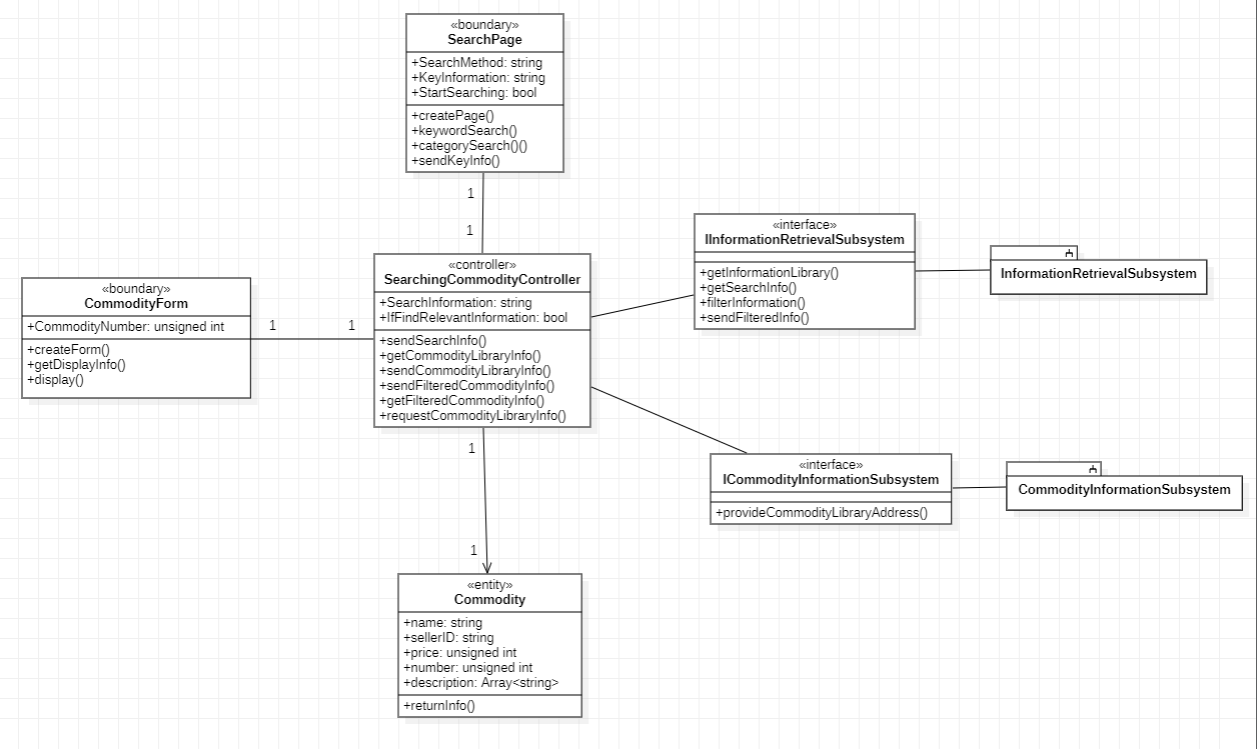
本项目所用的子系统和接口如下：

|  |  |  |
| --- | --- | --- |
| 子系统 | 子系统接口 | 接口功能 |
| Information Retrieval | getInformationLibrary() | 获取信息库 |
| getSearchInfo() | 获取搜索信息 |
| filterInformation() | 筛选获得的信息 |
| sendFilteredInfo() | 发送筛选后的信息 |
| Commodity Information | sendAllCommodityInfo() | 发送所有商品信息 |
| Customer Communication | selectCustomer() | 选择交流客户 |
| sendMessages() | 发送消息 |
| checkMessages() | 查看消息 |
| Goods Additions And Deletions | checkPostedGoods() | 查看已上架商品 |
| applyPostGoods() | 申请发布商品 |
| deleteGoods() | 删除已发布商品 |
| editProductDescription() | 编辑商品描述 |
| Payment Transfer | checkPayment() | 确认支付情况 |
| storePayment() | 暂存货款至后台 |
| Users Information | getuserInformation() | 获取用户信息 |
| updateuserInformation() | 更新用户信息 |
| addUser() | 添加用户 |
| deleteUser() | 删除用户 |
| showuserinformation() | 展示用户信息 |
| Shopping cart Inventory | showItemsinformation() | 展示商品信息 |
| changeItemsinformation() | 更改商品信息 |
| Orders Information | getorderInformation() | 获取订单信息 |
| updateorderInformation() | 更新订单信息 |
| Addorder() | 添加订单 |
| deleteorder() | 删除订单 |
| showorderinformation() | 展示订单信息 |
| Cellphone Confirm | VerifyCellPhoneNumber() | 确认电话号码 |
| Page Display | jumpToSpecifiedPage() | 跳转到特定界面 |
| Payment | EnquiryPaymentAmount() | 支付金额查询 |
| Payment() | 金额支付 |
| paymentResultConfirm() | 支付结果确认 |
| Logistics Management | getLogisticsInformation() | 获取物流信息 |
| updateLogisticsInforma  tion() | 更新物流信息 |
| LogisticsInformationVisualization() | 物流情况展示 |
| Platform Intervention | ProofHandling() | 处理举证 |
| VoucherApproval() | 审核凭证 |
| Abnormal Order Retrieva | GetAbnormalOrder() | 获取异常订单 |
| UpdateAbnormalOrder() | 修改异常订单 |
| DeleteAbnormalOrder() | 删除异常订单 |

## 用例实现

### 用例实现——[检索商品] 赖羿龙

类设计



子系统设计

本用例（检索商品）的实现需要两个子系统的合作，本节中展示一个（InformationRetrieval Subsystem）的实现。

商品信息子系统（CommodityInformationSubsystem）

本子系统的1个主要接口设计如下：

提供商品库访问地址：provideCommodityLibraryAddress()

信息检索子系统（InformationRetrieval Subsystem）

(1)本子系统的4个主要接口设计如下：

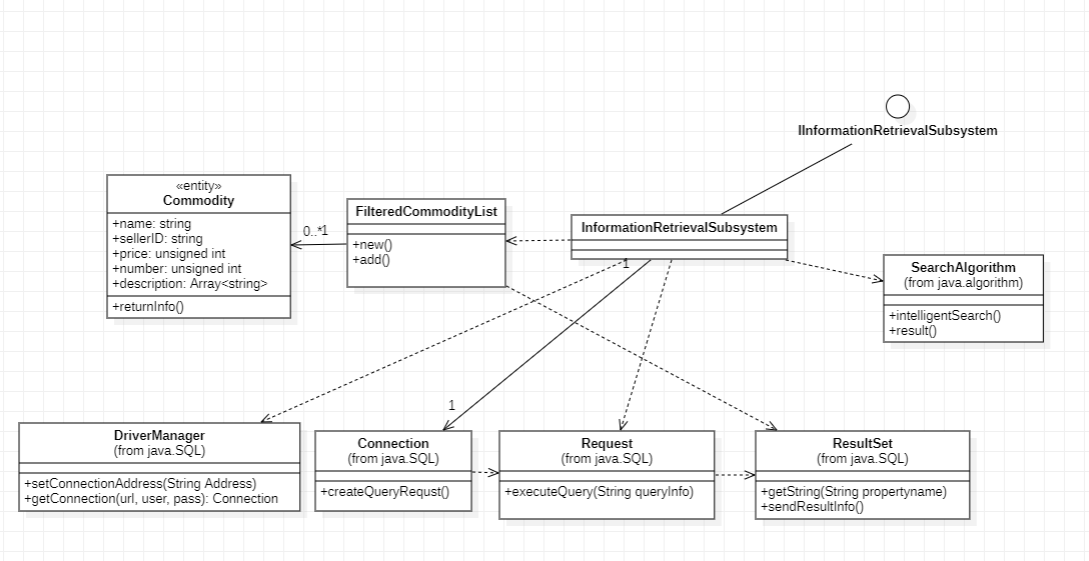
获取信息库：getInformationLibrary()

获取搜索信息：getSearchInfo()

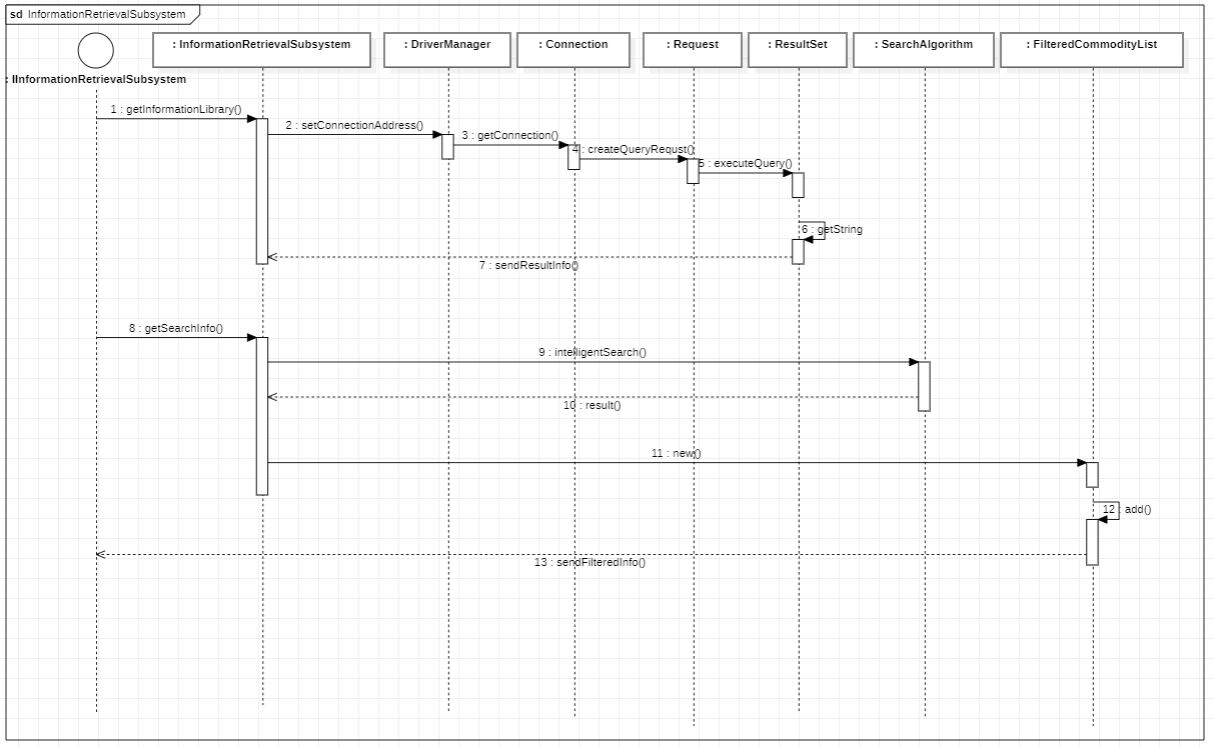
筛选信息：filterInformation()

发送筛选后的信息：sendFilteredInfo()

(2)为实现子系统功能，本子系统类图如下：

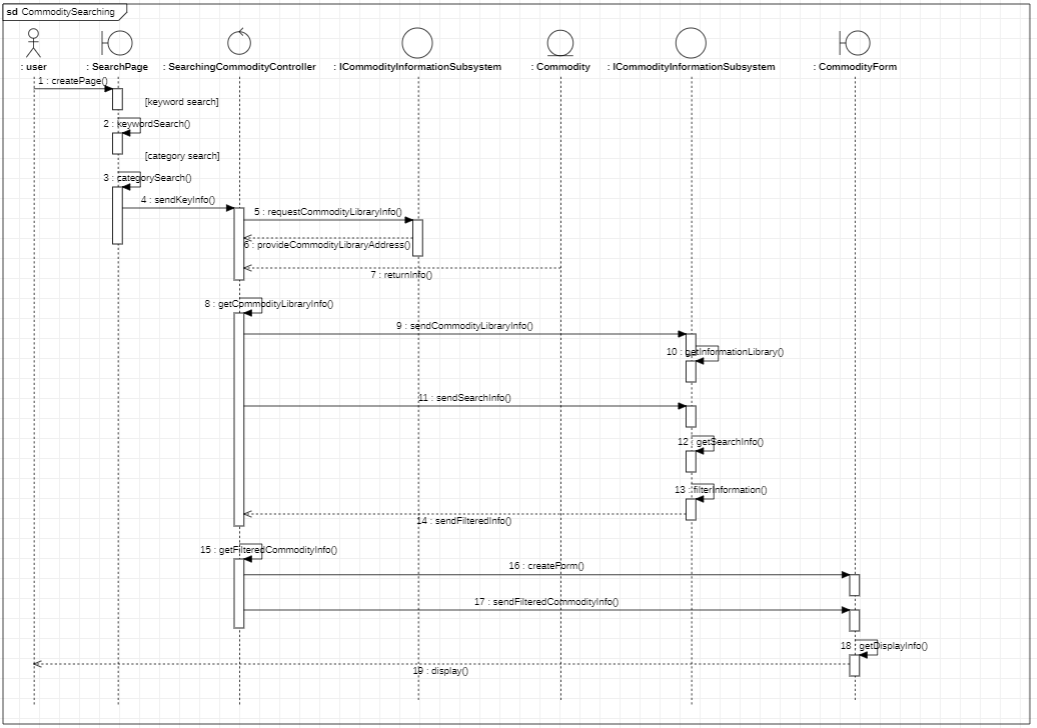


(3)本子系统时序图如下



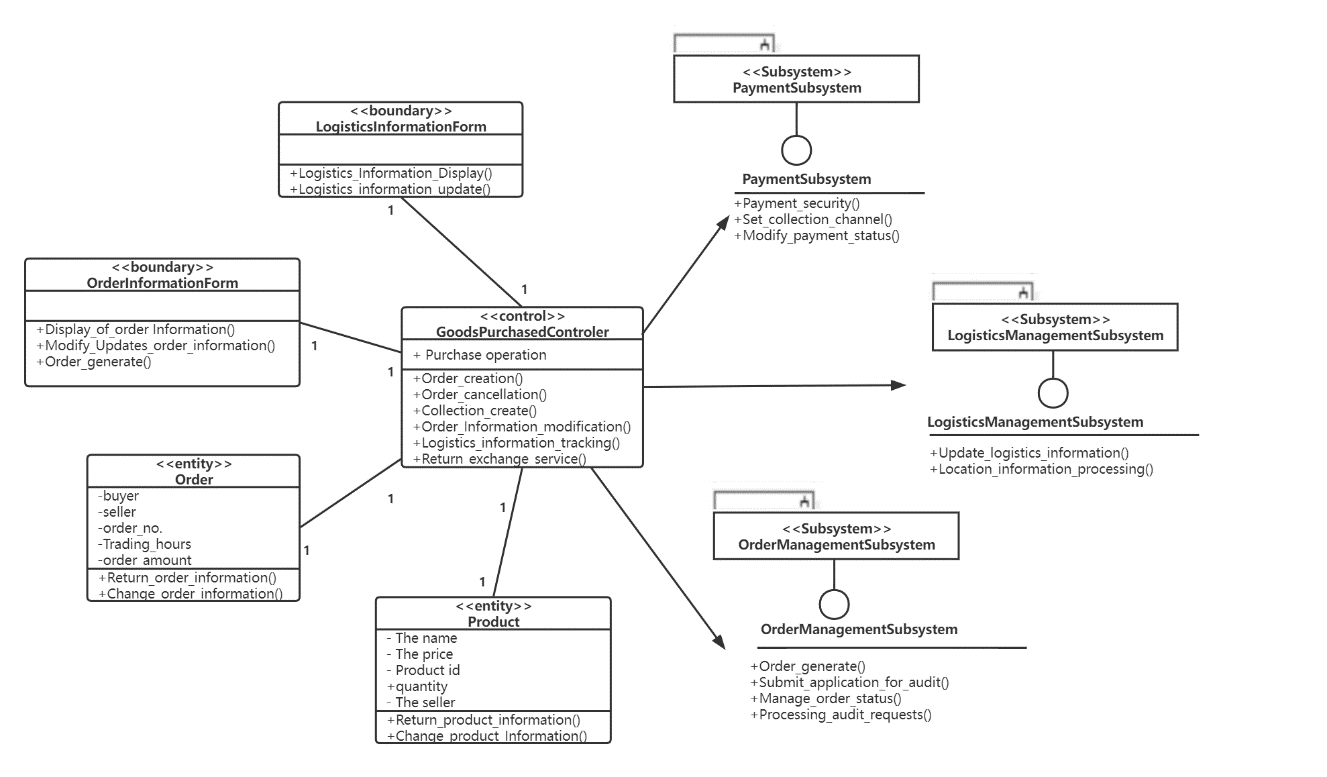
用例时序图设计

在上述子系统的基础上，本用例的时序图如下：



### 用例实现——[购买商品] 吕天成

类设计



子系统设计

本用例（购买商品）的实现需要PaymentSubsystem，LogisticsManagementSubsystem，OrderManagementSubsystem三个子系统的合作，本节中展示一个（OrderManagementSubsystem）的实现。

支付子系统（PaymentSubsystem）

本子系统的3个主要接口设计如下：

支付金额查询：EnquiryPaymentAmount()

金额支付：Payment()

支付结果确认：paymentResultConfirm()

2. 物流管理子系统（LogisticsManagementSubsystem）

(1) 本子系统的3个主要接口设计如下：

获取物流信息：getLogisticsInformation()

更新物流信息：updateLogisticsInformation()

物流情况展示：LogisticsInformationVisualization()

3.订单管理子系统（OrderManagementSubsystem）

(1)本子系统的4个主要接口设计如下：

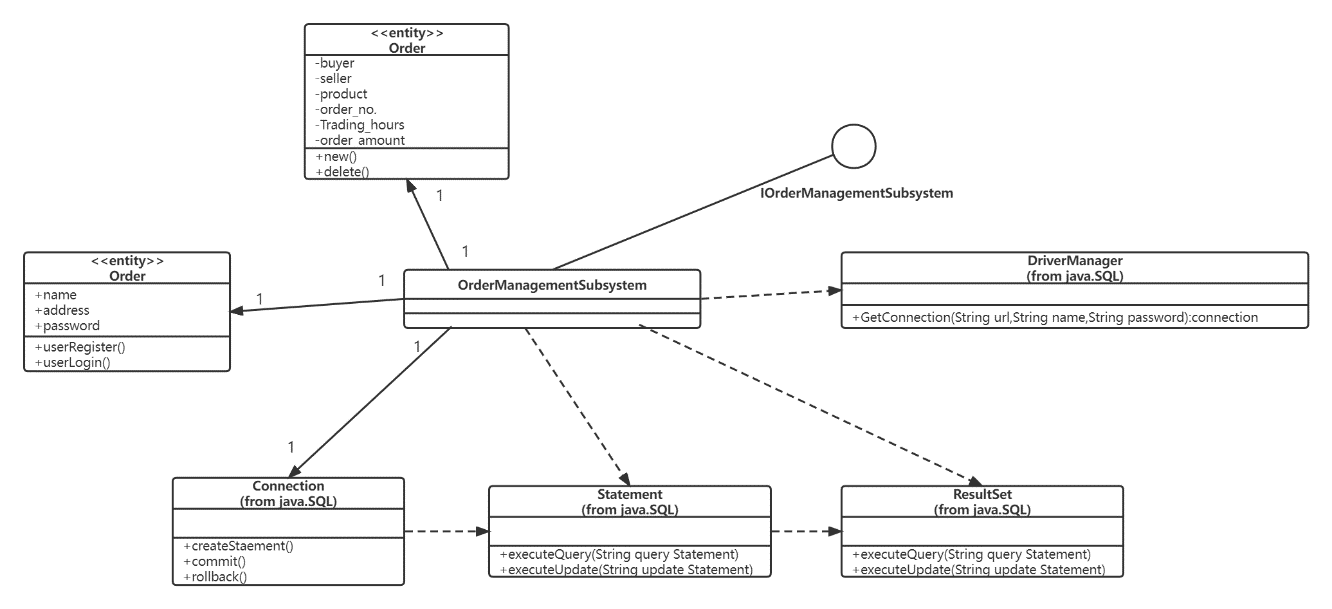
获取订单库：getOrderLibrary()

更新订单信息：updateOrder()

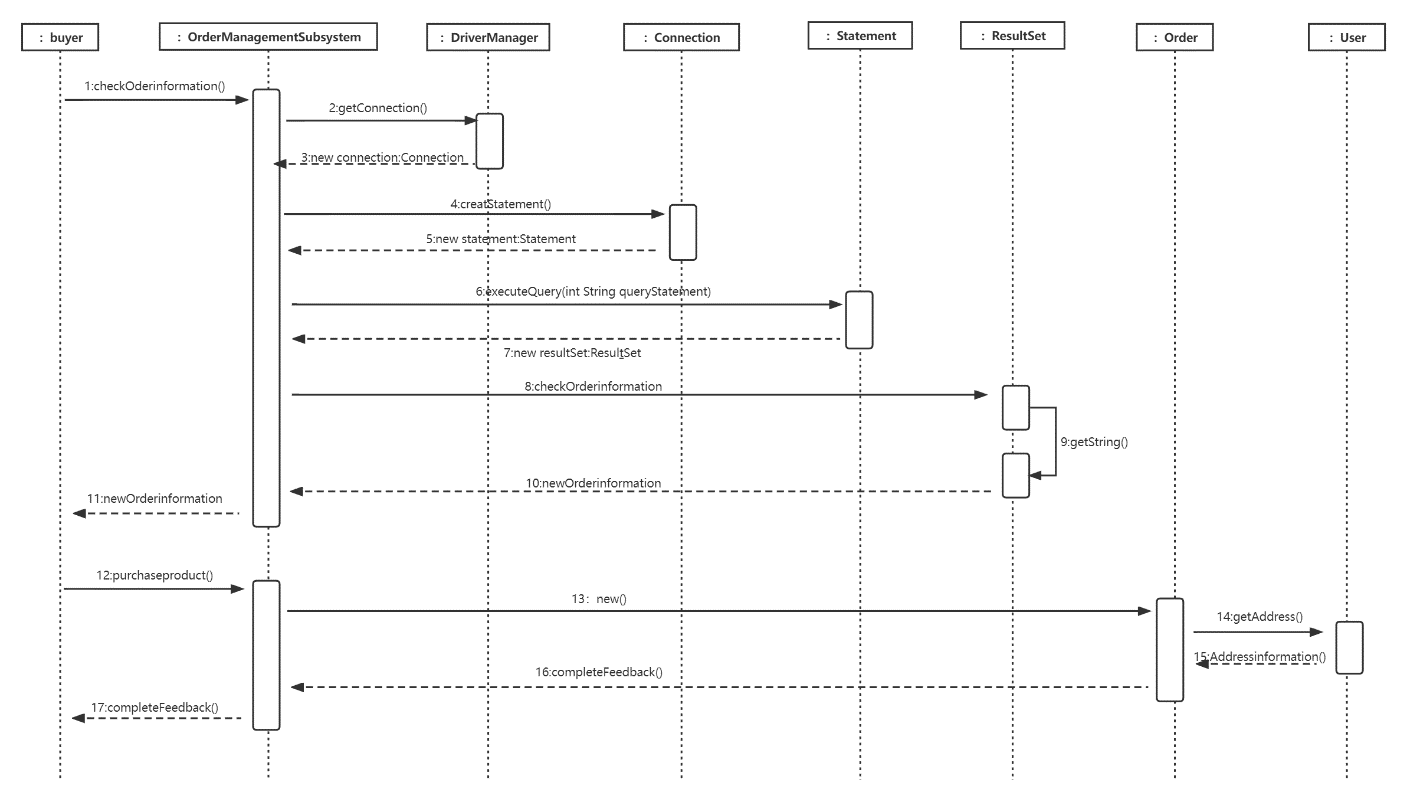
删除订单：deleteOrder()

添加订单：addOder()

(2)为实现子系统功能，本子系统类图如下：

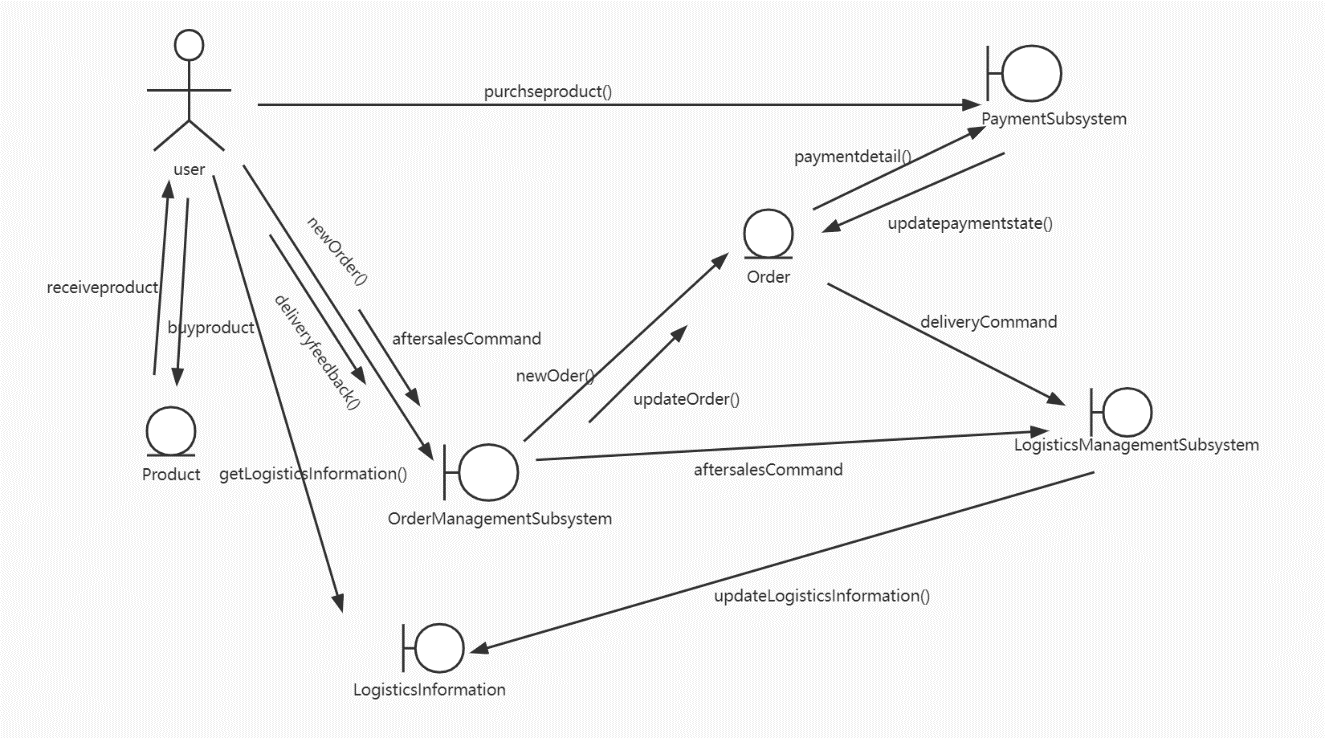


(3)本子系统时序图如下



用例通信图设计

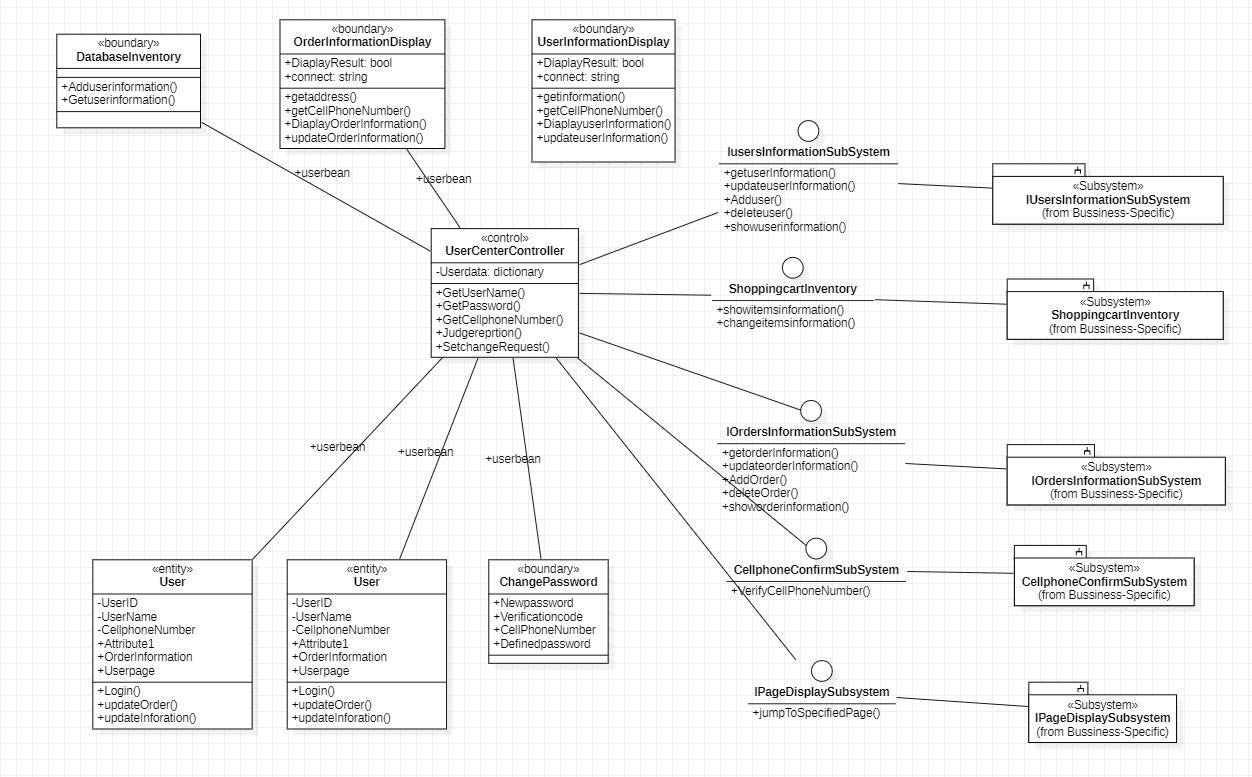
在上述子系统的基础上，本用例的通信图如下：



### 用例设计——用户中心 张勤杭

（用户信息修改、用户信息展示、商品订单查看删除、购物车查看修改）

（一）.类设计



（二）.子系统设计

本用例（添加社交圈）的实现需要五个子系统的合作，分别是1.IUsersInformationSubSystem（用户信息子系统）：

本子系统与平台信息系统的用户目录相关联，实现了将用户信息添加到数据库的功能，以及增加删除用户、增加更新信息等功能，具有持久性的机制。主要接口如下：

获得用户信息：getuserInformation()

更新用户信息：updateuserInformation()

添加用户： Adduser()

删除用户： deleteuser()

展示用户信息：showuserinformation()

2.ShoppingcartInventory（购物车展示子系统）：

本子系统是为了实现对购物车的访问以及购物车的更新，可以展示修改购物车信息，具有持久性的分析机制。主要接口如下：

展示商品信息：showitemsinformation()

更改商品信息：changeitemsinformation()

3.IOrdersInformationSubSystem（订单信息子系统）：

本子系统与平台信息系统的订单目录相关联，实现了将订单信息添加到数据库的功能，以及增加删除订单、增加更新订单信息等功能，具有持久性的机制。主要接口如下：

获得订单信息：getorderInformation()

更新订单信息：updateorderInformation()

添加订单： Addorder()

删除订单： deleteorder()

展示订单信息：showorderinformation()

4.CellphoneConfirmSubSystem（电话确认子系统）：

本子系统用于确认用户的手机信息是真实的，本子系统调用第三方验证码收验的API，向用户发送验证码，根据用户的输入来判断是否正确。主要接口如下：

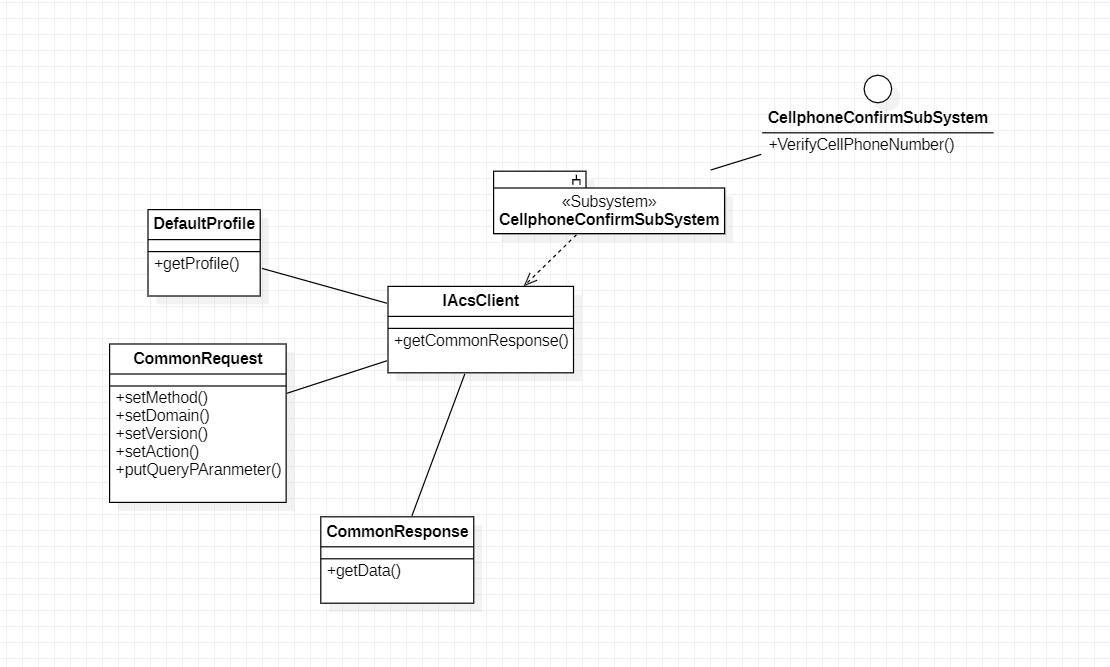
确认电话号码：VerifyCellPhoneNumber()

5.IPageDisplaySubsystem（页面跳转子系统）：

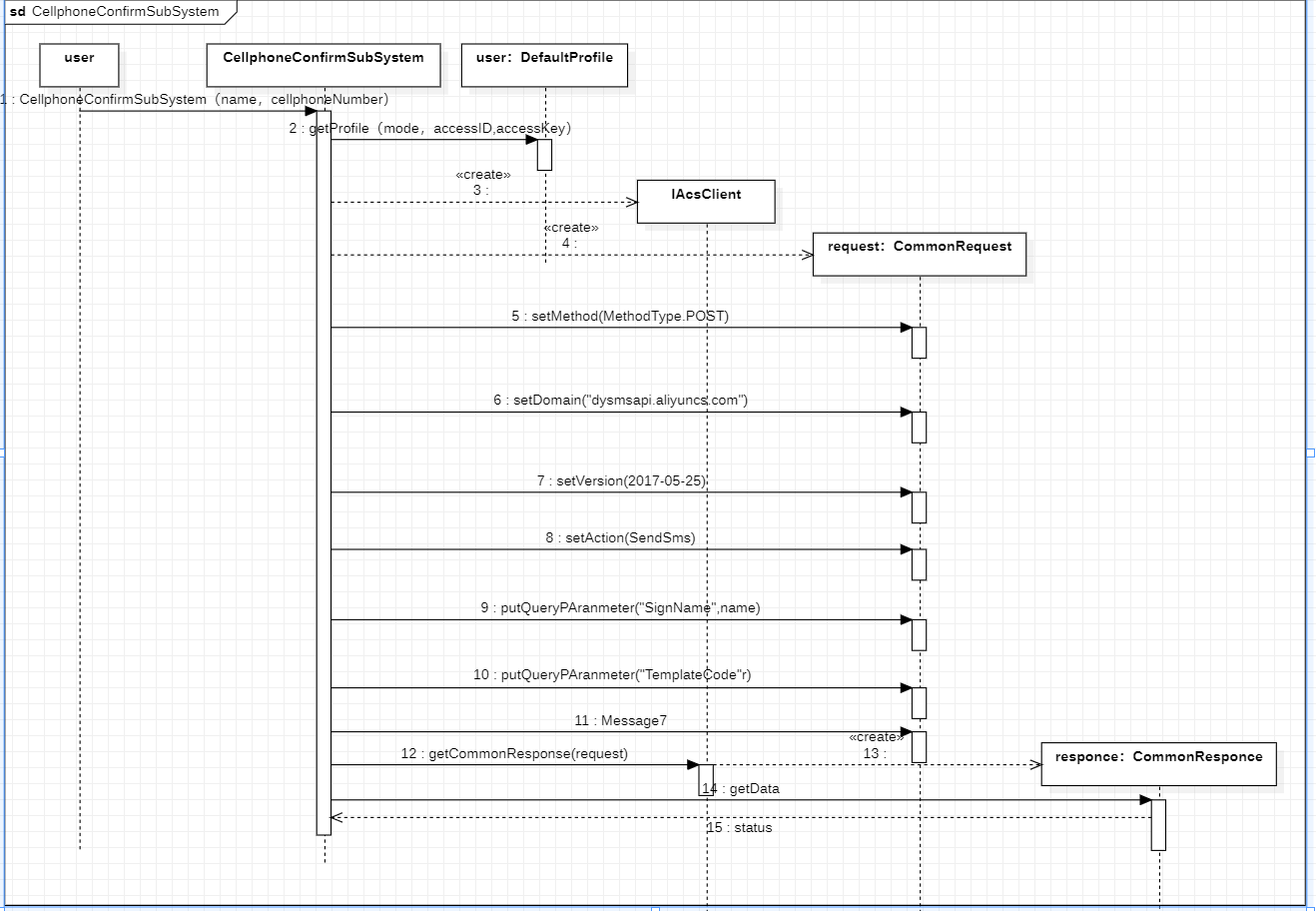
本子系统用于在用户查看个人信息以及购物车以及订单信息时页面的跳转。主要接口如下：

跳转特定界面：jumpToSpecifiedPage()

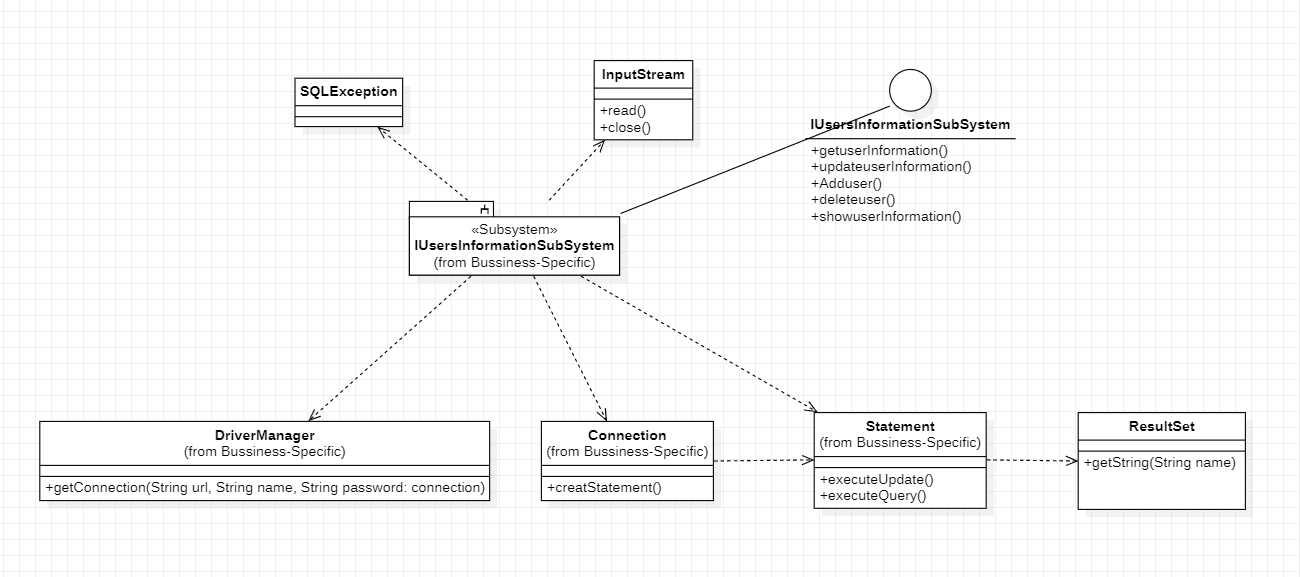
其中为了实现CellphoneConfirmSubSystem子系统，子系统类图如下：



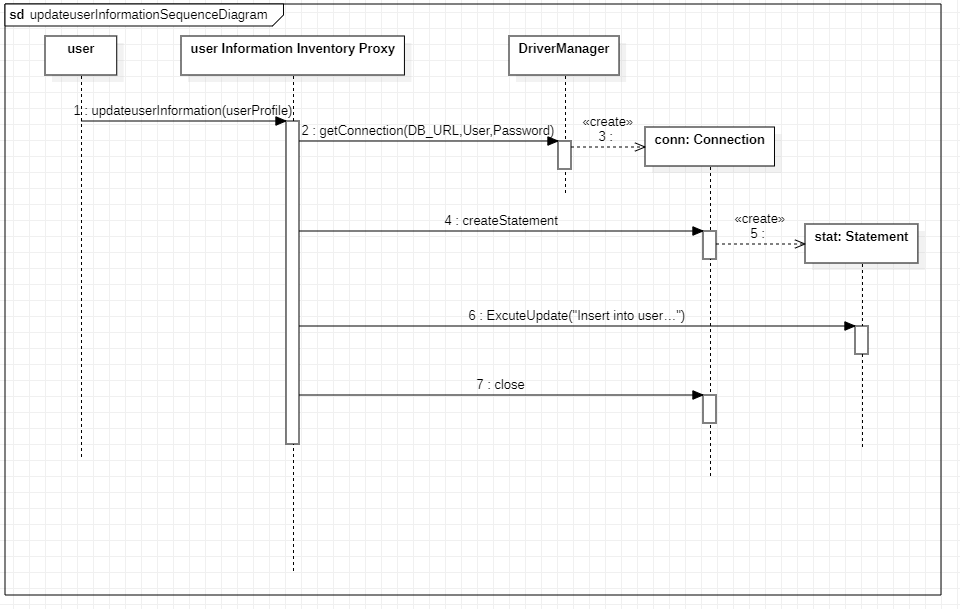
为了实现 CellphoneConfirmSubSystem接口的序列图如下：



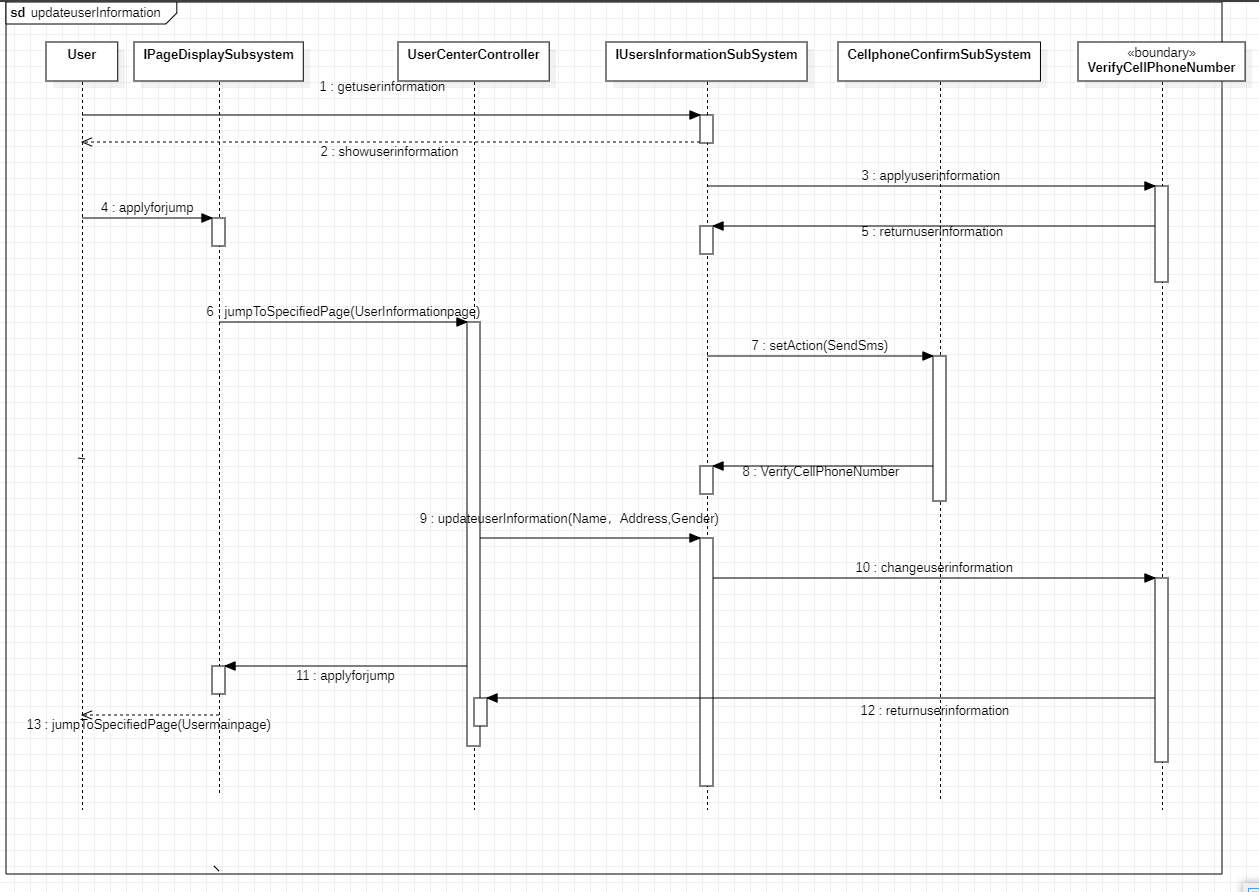
其中为了实现IUsersInformationSubSystem子系统，子系统类图如下：



为了实现 updateuserInformation接口的序列图如下：

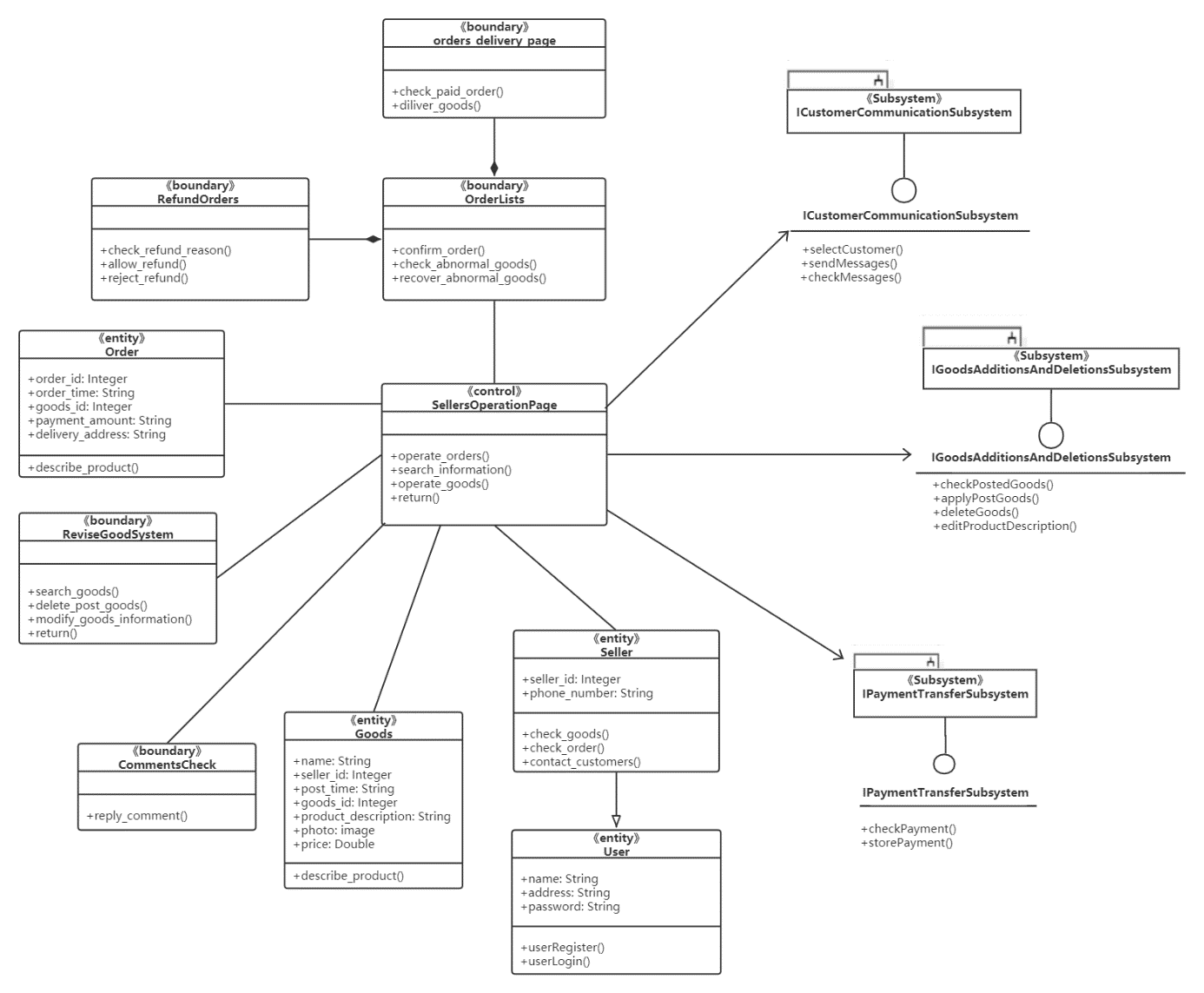


用例在添加子系统更新类图后的修改个人信息的时序图如下：



### 用例实现——出售商品 黄远宏

类设计



（二）所需子系统

本用例（出售商品）的实现需要CustomerCommunication Subsystem、GoodsAdditionsAndDeletions Subsystem、PaymentTransfer Subsystem这三个子系统的合作。

1. CustomerCommunication Subsystem

该子系统用于建立起卖家和顾客的联系，其3个主要接口设计如下：

选择交流客户：selectCustomer()

发送消息：sendMessages()

查看消息：checkMessages()

2. GoodsAdditionsAndDeletions Subsystem

该子系统用于商品的增删改查，其4个主要接口设计如下：

查看已上架商品：checkPostedGoods()

申请发布商品：applyPostGoods()

删除已发布商品：deleteGoods()

编辑商品描述：editProductDescription()

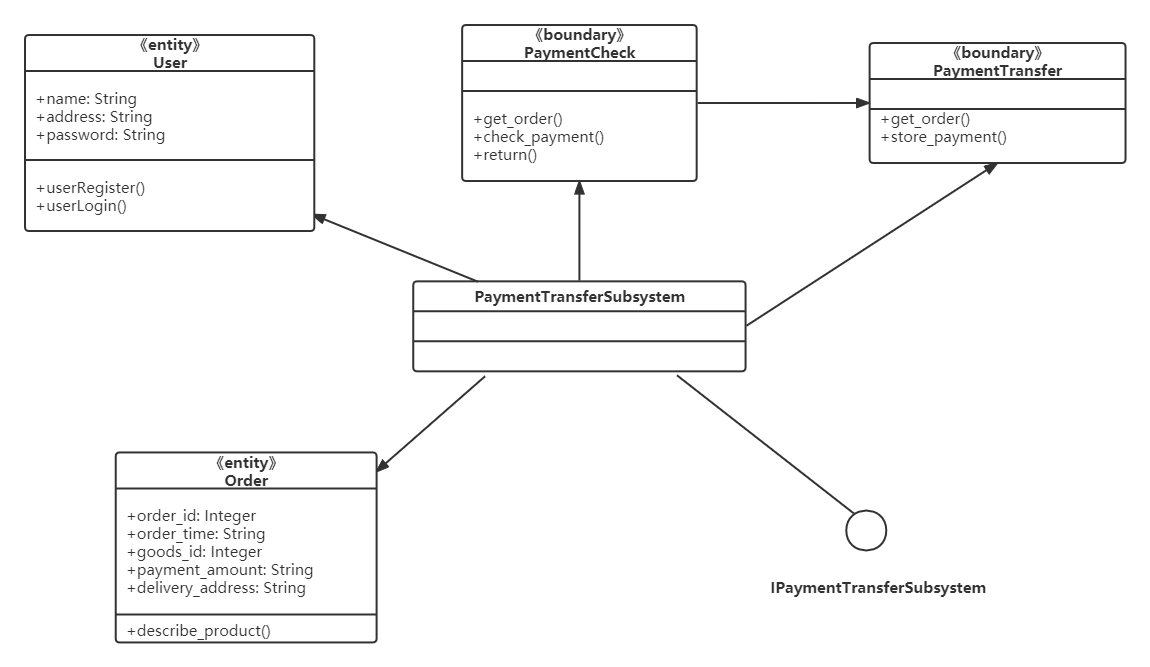
3. PaymentTransfer Subsystem

（1）该子系统用于接收和暂存用户付款，其2个主要接口设计如下：

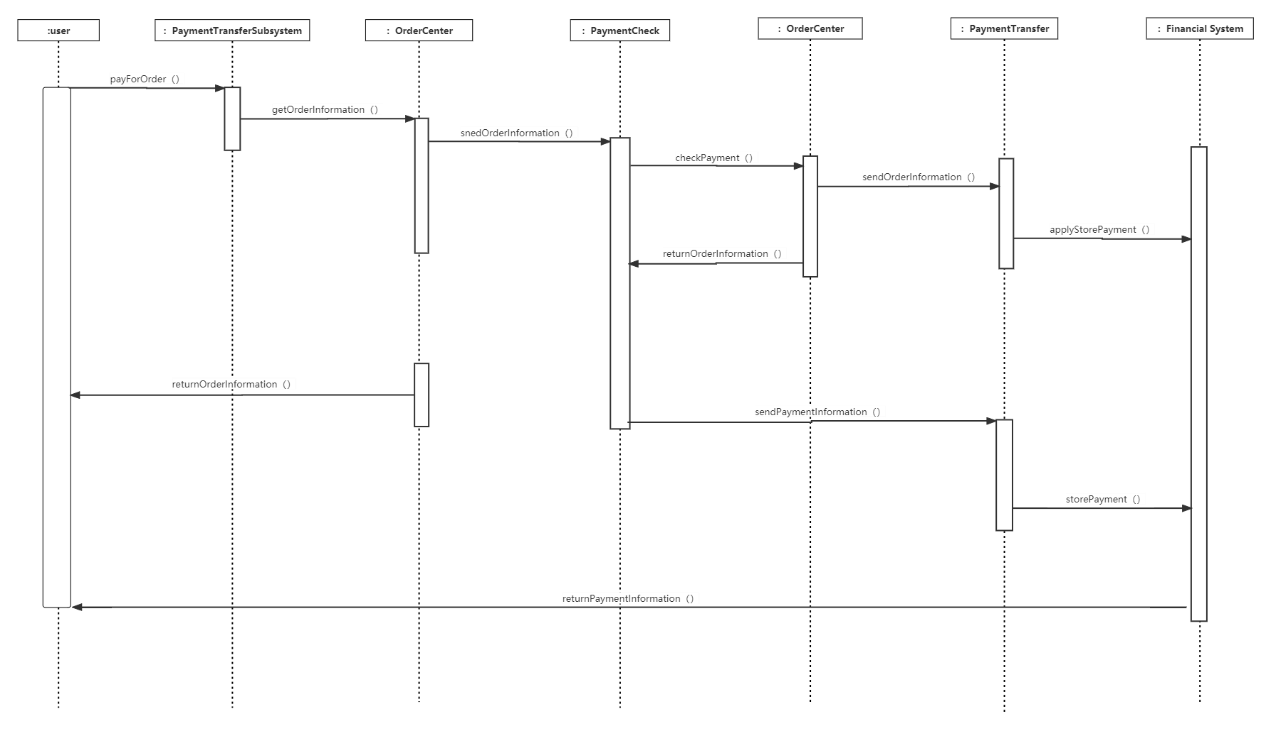
确认支付情况：checkPayment()

暂存货款至后台：storePayment()

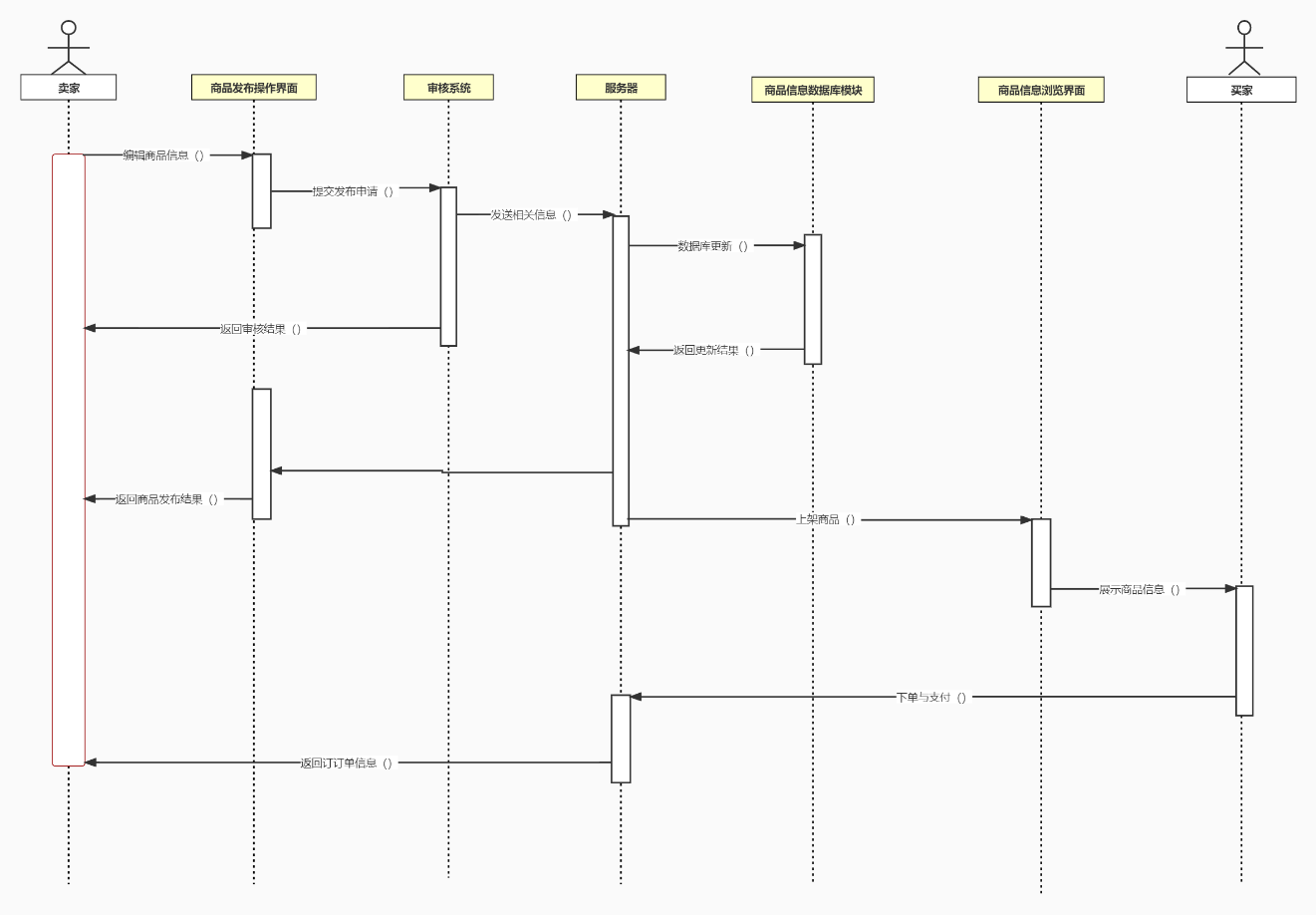
（2）为了实现子系统功能，本子系统的类图设计如下：



（3）本子系统的时序图设计如下：

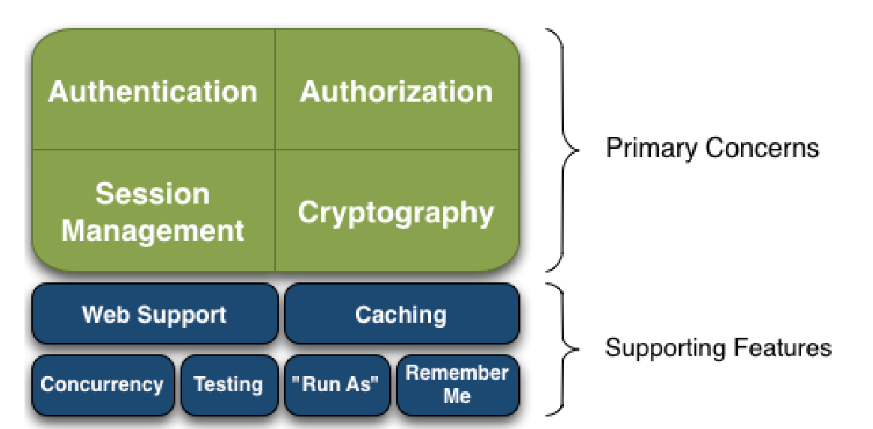


（三）本用例时序图



## 安全性实现

用户中心设计满足了安全性的分析机制（使⽤Shiro）。



Authentication：身份认证/登录，验证用户是不是拥有相应的身份,例如账号密码登陆；

Authorization：授权，即权限验证，验证某个已认证的用户是否拥有某个权限；即判断用户是否能做事情，常见的如：验证某个用户是否拥有某个角色。或者细粒度的验证某个用户对某个资源是否具有某个权限；

Session Manager：会话管理，即用户登录后就是一次会话，在没有退出之前，它的所有信息都在会话中；会话可以是普通JavaSE环境的，也可以是如Web环境的；

Cryptography：加密，保护数据的安全性，如密码加密存储到数据库，而不是明文存储；

Web Support：Web支持，可以非常容易的集成到Web环境；

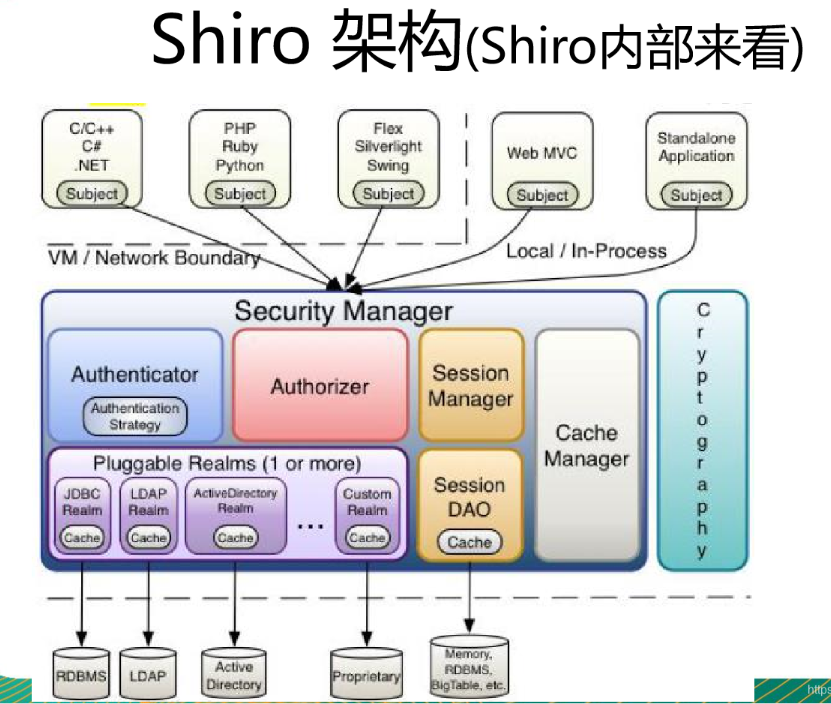
Caching：缓存，比如用户登录后，其用户信息、拥有的角色/权限不必每次去查，这样可以提高效率；

Concurrency：shiro支持多线程应用的并发验证，即如在一个线程中开启另一个线程，能把权限自动传播过去；

Testing：提供测试支持；

Run As：允许一个用户假装为另一个用户（如果他们允许）的身份进行访问；

Remember Me：记住我，这个是非常常见的功能，即一次登录后，下次再来的话不用登录了。



Subject：主体，可以看到主体可以是任何可以与应用交互的“用户”；

SecurityManager：相当于SpringMVC中的DispatcherServlet或者Struts2中的FilterDispatcher；是Shiro的心脏；所有具体的交互都通过SecurityManager进行控制；它管理着所有Subject、且负责进行认证和授权、及会话、缓存的管理。

Authenticator：认证器，负责主体认证的，这是一个扩展点，如果用户觉得Shiro默认的不好，可以自定义实现；其需要认证策略（Authentication Strategy），即什么情况下算用户认证通过了；

Authrizer：授权器，或者访问控制器，用来决定主体是否有权限进行相应的操作；即控制着用户能访问应用中的哪些功能；

Realm：可以有1个或多个Realm，可以认为是安全实体数据源，即用于获取安全实体的；可以是JDBC实现，也可以是LDAP实现，或者内存实现等等；由用户提供；注意：Shiro不知道你的用户/权限存储在哪及以何种格式存储；所以我们一般在应用中都需要实现自己的Realm；

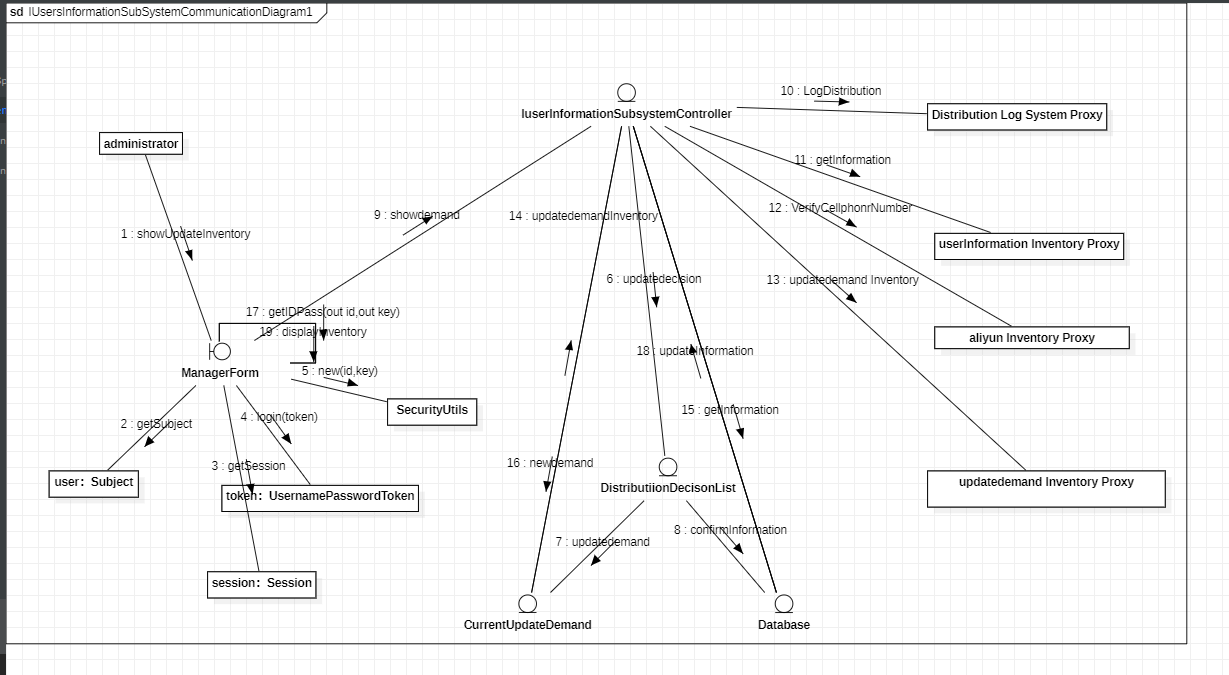
SessionManager：如果写过Servlet就应该知道Session的概念，Session呢需要有人去管理它的生命周期，这个组件就是SessionManager；而Shiro并不仅仅可以用在Web环境，也可以用在如普通的JavaSE环境、EJB等环境；所有呢，Shiro就抽象了一个自己的Session来管理主体与应用之间交互的数据；这样的话，比如我们在Web环境用，刚开始是一台Web服务器；接着又上了台EJB服务器；这时想把两台服务器的会话数据放到一个地方，这个时候就可以实现自己的分布式会话（如把数据放到Memcached服务器）；

SessionDAO：DAO大家都用过，数据访问对象，用于会话的CRUD，比如我们想把Session保存到数据库，那么可以实现自己的SessionDAO，通过如JDBC写到数据库；比如想把Session放到Memcached中，可以实现自己的Memcached SessionDAO；另外SessionDAO中可以使用Cache进行缓存，以提高性能；

CacheManager：缓存控制器，来管理如用户、角色、权限等的缓存的；因为这些数据基本上很少去改变，放到缓存中后可以提高访问的性能

Cryptography：密码模块，Shiro提高了一些常见的加密组件用于如密码加密/解密的

据此绘制用户中心用例通信图如下：



# 团队分工

|  |  |
| --- | --- |
| **概述和简介** | 由张勤杭同学完成 |
| **架构分析图** | 子系统和架构确认后由赖羿龙同学绘制 |
| **子系统和接口** | 共同完成 |

## 用例实现：

|  |  |
| --- | --- |
| 吕天成 | **购买商品** |
| 赖羿龙 | **检索商品** |
| 张勤杭 | **用户中心** |
| 吴浩泽 | **订单介入** |
| 黄远宏 | **出售商品** |