

UML Assignment3

空中画园

——线上画作分享交易系统

团队名称：发家致富

组长：梁伊雯

组员：黄金坤 赵敏 李林飞 杨茗溟

指导教师：孙萍

目录

一、项目简介

- 1.1 文档用途
- 1.2 项目背景
- 1.3 项目目标
- 1.4 项目进展
- 1.5 修改内容
- 1.6 术语表

二、架构细化与子系统设计

- 2.1 平台架构图
 - 2.1.1 系统架构图
 - 2.1.2 技术架构图
- 2.2 子系统和接口设计
 - 2.2.1 用户管理系统
 - 2.2.2 画展管理系统
 - 2.2.3 SFExpress子系统
 - 2.2.4 权限管理系统
 - 2.2.5 直播管理系统
 - 2.2.6 举报管理系统
 - 2.2.7 申请管理系统
 - 2.2.8 交易管理系统
 - 2.2.9 数据库系统
- 2.3 第三方平台接口规范——第三方快递平台
 - 2.3.1 路由注册接口
 - 2.3.2 路由查询接口
 - 2.3.3 数据传输MD5加密
 - 2.3.4 API接口函数调用
 - 2.3.5 接口调用示例
- 2.4 子系统示范与接口明确

三、用例图更新

四、设计模型

- 4.1 五个详细的用例实现
 - 4.1.1 举办画展用例实现
 - 4.1.2 购买画作用例实现
 - 4.1.3 观看直播用例实现
 - 4.1.4 审核申请用例实现
 - 4.1.5 登录账号用例实现
- 4.2 类设计

五、组员分工

一、项目简介

1.1 文档用途

本次系统设计文档改进了用例模型，并对上次系统设计文档中的的类做了调整与更新，将其分化成了不同的子系统，同时提供了子系统的接口列表，使其更适用于具体实现。同时文档对于软件架构做出了细化，呈现了依赖平台的设计架构。此外，还通过一个具体示例演示了本系统与第三方平台顺丰快递之间的接口规范。在针对分析模型进行改进、完善后提出设计模型，并通过举例出的几个用例实现和类设计来展示设计模型。

1.2 项目背景

伴随着科学技术的发展，社会文化逐渐繁荣，专业画师与绘画爱好者的群体日益壮大，但专注于绘画交流与交易的完善系统尚未出现。为满足专业画师展示绘画作品、提高艺术名气、获取画作收益的需求，满足广大绘画爱好者欣赏学习绘画艺术、购买心仪绘画作品的需求，我们需要一个专业完善的线上画作分享交易管理系统，来创造和谐纯净的绘画交流交易环境，进一步促进绘画艺术的繁荣。

1.3 项目目标

“空中画园线上画作分享交易管理系统”是集画作分享、绘画直播、画作定制等功能于一身的处理系统。该系统旨在为专业画师与绘画爱好者的群体提供绘画交流与交易的平台。对于不同的使用者会提供不同的功能

- 对于使用本系统的游客来说，系统提供注册账号和浏览画作的功能。
- 对于使用本系统的注册用户来说，系统能够支持浏览画作，观看直播的功能。
- 对于管理员来说，系统可以提供审核画作发布申请，审核直播发布申请，处理举报信息等基本功能。
- 对于认证为画师的使用者来说，系统可以提供申请上传画作，申请绘画直播，申请开办画展等基本功能。

1.4 项目进展

我们对上次的软件架构进行了细化，并分成了系统架构图和技术架构图。系统架构图用来表现不同层级内要实现的功能，分为界面区、业务区和数据区。技术架构图用来展现不同层级的功能实现所需要依赖的具体技术。

然后我们根据架构所提出的功能将系统中的每一层分为不同的子系统，并根据子系统要实现的功能为子系统定义了它的行为与元素，之后我们在上次分析类的基础上，在充分分析各个分析类之间的属性、方法以及各个分析类之间的内在关系和调用关系之后，经过重新整合改进了分析类，并根据具体需求定义了一些新的必要的类，两者结合便是更适用于具体实现的设计类，用于完成子系统的功能。紧接着依据子系统对用例模型进行了更新改进。然后我们重点分析并指定了为实现子系统功能调用而要求各个子系统提供的接口类以及其需要的接口类。我们选择其中（一个子系统）为例，详细分析并实现了其接口的设计。

接着我们查阅了相关资料，了解了如何在我们的系统中调用其他第三方平台子系统API，在充分研究了其API后，着重选择了其中一个第三方平台的子系统，详细设计了本系统与这个系统的接口实现。

经过以上步骤后提出了我们的设计模型。

1.5 修改内容

对系统架构图进行了扩充，本次环节新增加的技术架构图介绍了实现系统所依赖的技术，这些技术不仅可以用来实现系统的基本功能还可以用来实现本系统的一些设计机制。另外还对设计模型进行了改进形成了分析模型，并依此对用例模型按照子系统进行了重新划分，同时还加入了某些在分析设计环节提出的新用例。除此之外还研究并详细说明了与具体第三方平台的API设计规范。

1.6 术语表

术语	定义
游客	未注册本系统的用户，但可以浏览网页界面
注册用户	在本系统有注册账号的用户，可以参与观看主播画展、评论等
画师	通过画师认证的用户，拥有发布直播、举办线上画展的特权
管理员	对平台信息进行审核、系统进行维护升级的用户
系统	与用户进行交互的界面控制中枢
直播	指网络直播，用户双方可以在同一时间通过网络系统在平台上相互交流的一种新兴网络社交方式

术语	定义
惩罚	管理员对违反平台相关规定的用户进行权限的限制或注销账号
处理举报	画师根据相关用户提供的信息进行调查后再对违规用户进行惩罚
线上画展	画师向系统上传3D画作模型，系统进行渲染烘托生成模拟画展
实名认证	用户或画师通过提供本人的证件照和人脸识别进行身份认证
直播间	在互联网时代，直播间主要指某一个人或一个组织开通的网络直播节目
直播条例	具有规范性的、帮助构建良好网络直播秩序的相关条例
信誉积分	指定量考量用户状态的分数，信誉积分越高，表明该用户诚信度越高

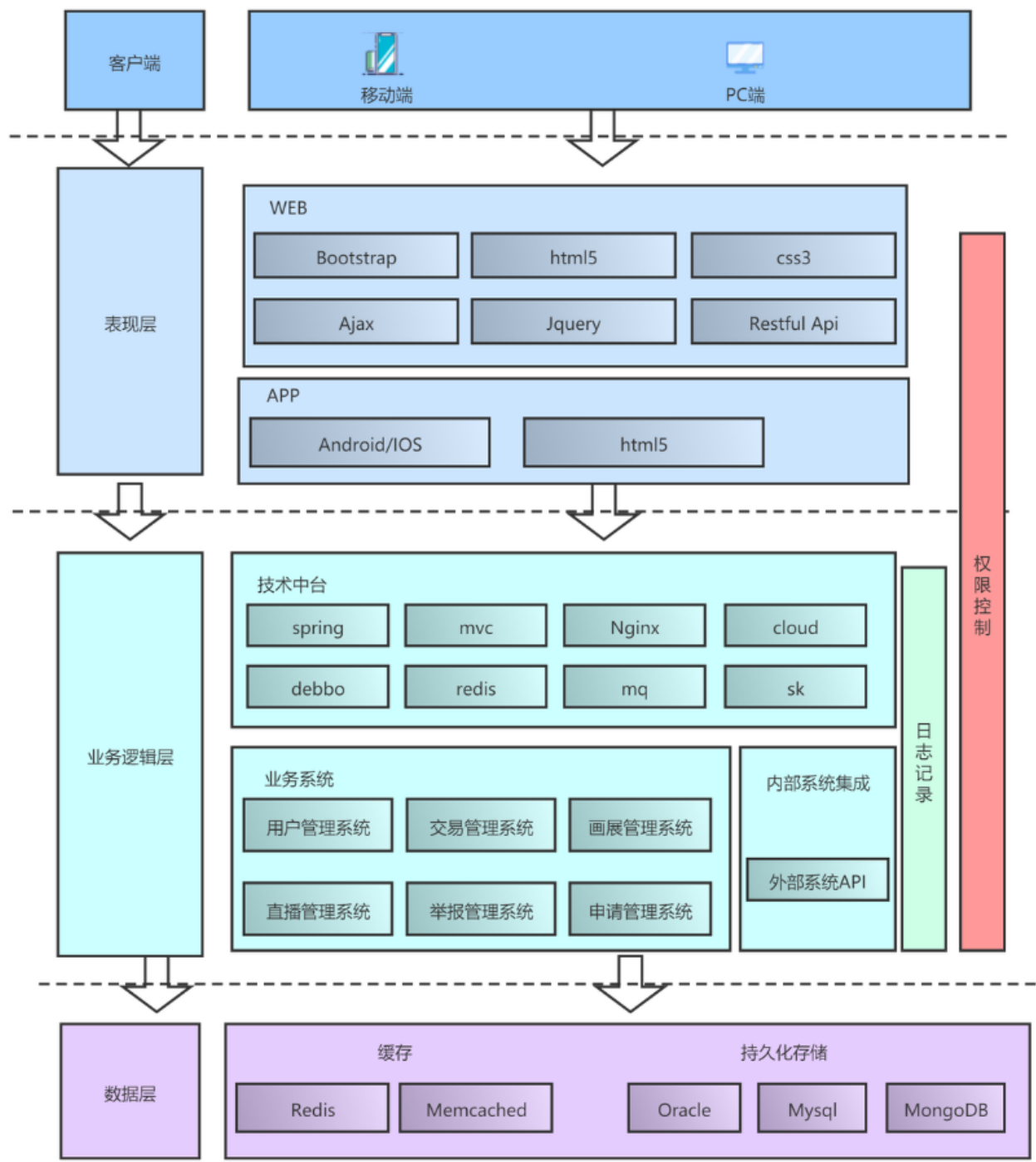
二、架构细化与子系统设计

2.1 平台架构图

2.1.1 系统架构图



2.1.2 技术架构图



系统逻辑架构主要分为3层：表现层、业务逻辑层、数据层。客户端提供移动端和PC端两种系统访问方式，表现层由此分为WEB和APP两种结构。在WEB端，我们主要采用Bootstrap框架。

• Bootstrap

Bootstrap是基于HTML5和CSS3开发的，它在jQuery的基础上进行了更为个性化的完善，形成一套自己独有的网站风格，并兼容大部分jQuery插件。Bootstrap 提供了一个带有网格系统、链接样式、背景的基本结构。使用Bootstrap能够快速搭建出网站。Bootstrap包含css和一些jquery插件，依赖于jquery，所以必须一起使用。

• jquery

jQuery是JavaScript库,能够极大地简化JavaScript编程,能够更方便的处理DOM操作和进行Ajax交互。jQuery十分轻巧,且具有强大的选择器和出色的DOM操作的封装。jQuery的事件处理机制吸收了JavaScript专家Dean Edwards编写的事件处理函数的精华, jQuery在处理事件绑定的时候相当可靠。

- **Ajax和RestfulAPI**

WEB端通过Ajax构造快速动态的交互式网页, 调用restful接口向后端传送Json格式数据。Restful API对uri进行了限制, 只用于定义资源, 易于理解, 通俗易懂。

- **APP端开发**

APP端采用Android或IOS混合html5进行开发, 这是一种取长补短的开发模式, 原生代码部分利用WebView插件或者其它框架为H5提供容器, 程序主要的业务实现、界面展示都是利用与H5相关的Web技术进行实现的。混合开发效率高, 更新、部署、代码维护都比较方便, 每次升级版本只需要在服务器端升级即可。

- **微服务架构**

业务中台采用微服务架构, 以保障系统高可用性, 应对高频海量业务访问场景。微服务架构采用前后端分离设计, 前端页面逻辑和后端微服务业务逻辑独立开发、独立部署, 通过网关实现前后端集成。前台应用接入中台微服务的技术组件一般是API网关。本系统采用Nginx搭建网关, 实现分流、负载均衡、服务路由和访问日志等功能。

- **业务层框架**

业务层主要使用Spring MVC框架。Spring MVC 角色划分清晰, 分工明细, 并且和 Spring 框架无缝结合。

其工作流程为:

1. 客户端请求提交到 DispatcherServlet。
2. 由 DispatcherServlet 控制器寻找一个或多个 HandlerMapping, 找到处理请求的 Controller。
3. DispatcherServlet 将请求提交到 Controller。
4. Controller 调用业务逻辑处理后返回 ModelAndView。
5. DispatcherServlet 寻找一个或多个 ViewResolver 视图解析器, 找到 ModelAndView 指定的视图。
6. 视图负责将结果显示到客户端。

其他如Dubbo、Spring Cloud等为常见的微服务治理技术体系。

- **日志记录**

采用AOP分离业务逻辑和日志记录, 将打印日志的功能封装起来, 减少系统的重复代码, 降低模块间的耦合度。

- **数据层**

在数据层中引入Redis和memcached作为缓存，加速读写，提高并发性，降低后端持久化层数据库的负载。Redis作为缓存服务器，在service层去调用DAO层去查询时先去缓存服务器查询，如果存在则直接返回该数据，否则再去查询数据库。这样大量减少了对磁盘I/O的操作，减轻了数据库的压力。

使用mongoDB搭配mysql做持久化存储。在web页面事务用mysql表来存储，实际存储的对象如图片文字等存入mongoDB。mongodb优势是处理大数据、高并发，目前来看主要还是一个完整项目的辅助存储。相反mysql更加灵活，技术更加成熟，可根据不同需求选择不同模块和数据分别存储。

2.2 子系统和接口设计

我们在这个公共安全系统管理设计中采用MVC(Model-View-Controller)模型, 并对每个模块进行了子系统设计, 每个子系统提供了与其它系统交流的接口

2.2.1 用户管理系统

子系统名称: UserManagerManagement

接口类: IUserManager

功能描述: 该子系统处理所有涉及用户信息变化、读取的业务

接口函数:

(1)注册

getRegistrstionInfo(string nickName,string password,string telephone):bool

(2)登录

getLoginInfo(string userID,string password):bool

(3)关注画家

subscribeRequest(string userID,string painterID):bool

(4)取消关注

unsubscribeRequest(string userID,string painterID):bool

(5)注销

logout(string userID):bool

(6)修改信息

modifyInfo(string userID ,string infoType, string info):bool

2.2.2 画展管理系统

子系统名称: PaintingExhibitionManagement

接口类: IPaintingExhibitionManager

功能描述: 该子系统提供线上3D画展的创建、查询、管理、观看等服务。

接口函数:

PaintingExhibitionInformation (查询信息, 查询画展信息)

(1) 查看画家的画展信息

viewPERoom(string painterID) : string

(2) 查看画展简介

getPEIntroduction(string roomID, string painterID) : string

(3) 查看画展详细信息

getPEDetails(string roomID, string painterID) : string

PaintingExhibitionManagement (画展管理, 画家增删改查画展)

(1) 上传画作

uploadPaintings(image paintingImage, string descriptions) : file

(2) 创建画展房间

createPERoom(string painterID, string roomName, list <painting, description>, int type, string theme, int capacity, DateTime startTime, DateTime endTime) : PERoom

(3) 更新画展房间

updatePERoom(string roomID, string newContent) : PERoom

(4) 删除画展房间

deletePERoom(string roomID) : bool

(5) 发布画展

postPE(string roomID, string painterID) : bool

PaintingExhibitionInteractivity (画展交互, 用户浏览画展)

(1) 观看画展

joinPERoom(string roomID, string userID)

(2) 退出画展

dropOutPERoom(string roomID, string userID)

(3) 添加评论

`addComment(roomID, userID, comment) : bool`

(4) 删除评论

`deleteComment(string roomID, string userID, string commentID) : bool`

(5) 点赞(关注/收藏)

`likePE(string roomID, string userID) : bool`

(6) 取消点赞(关注/收藏)

`deletelikePE(string roomID, string userID) : bool`

2.2.3 SFExpress子系统

子系统名称:SFExpressManagement

接口类: ISFExpressManager

功能描述: 该子系统调用顺丰平台接口函数实现购买画作后发货、快递状态、查收等功能。实现方法 为用户确认订单后系统将为订单生成唯一的标识(**orderID**用户不可见), 将该值作为参数传入**SFExpressManagement**子系统中, 系统内部先通过**orderID**到数据库中找到订单的详细信息, 包括订单号、电话号码、收获地址、查询号类别、查询号、路由查询类别等, 这些信息是**SF**第三方接口所需要的参数。为保证信息的安全性, 先用**MD5**加密方式将订单信息进行加密, 然后传给**SF**第三方平台, 解密后通过相应的接口函数创建快递包裹, 并实时更新快递信息传递到该子系统中, 调用该系统的接口函数实现快递状态的查询等。**SF**第三方平台接口规范详见下文。

接口函数:

Order2Package (通过订单生成顺丰快递包裹)

(1) 生成SF包裹

`createSFPackage(string orderID) : package`

(2) 取消包裹

`cancelSFPackage(string orderID) : bool`

(3) 更新包裹信息

`updateSFPackage(string orderID, string updateContent) : package`

SearchExpress (快递查询)

(1) 查询快递位置

`getPosition(string orderID) : string`

(2) 查询历史记录

getHistoryInformation(string orderID) : list

(3) 查询快递状态

getStatus(string orderID) : string

2.2.4 权限管理系统

子系统名称: AuthorityManagement

接口类: IAuthorityManager

功能描述: 该子系统提供画家身份认证、注册用户身份认证、实名认证等功能

接口函数:

RealDentityAuthentication (实名认证)

(1) 申请(需提供身份证照片和面部实时识别信息)

applyRealDentityAuthentication(image IDcardImage, image faceRecognition) : bool

(2) 查询

checkRealDentityAuthentication(string userID) : bool

RegisteredUserAuthentication (注册用户身份认证)

(1) 申请(注册用户时系统自动赋予该身份权限)

applyRegisteredUserAuthentication() : bool

(2) 查询

checkRegisteredUserAuthentication(string userID) : bool

PainterDentityAuthentication (画家身份认证, 只有注册用户可以)

(1) 申请 (注册用户ID, 信誉积分, 点赞数, 活跃积分, 画技积分)

applyPainterDentityAuthentication(string userID, int creditScore, int likeNums, int activePoints, int paintingPoints) : bool

(2) 查询

checkPainterDentityAuthentication(string userID) : bool

2.2.5 直播管理系统

子系统名称: LivestreamManagement

接口类: ILivestreamManager

功能描述: 该子系统处理与直播相关的业务

接口函数:

(1) 发布直播

releaseLivestream(string painterID):bool

(2) 进入观看直播

enterLivestream(string UserID, string LivestreamRoomID) :bool

(3) 举报直播

reportLivestream(string reporterID,string reportedID,string content):void

(4) 发布直播弹幕

popupLivestream(string UserID, string popup):bool

(5) 退出直播

quitLivestream(string painterID, string LivestreamRoomID):bool

(6) 结束直播

endLivestream(string painterID, string LivestreamRoomID):bool

2.2.6 举报管理系统

子系统名称: ReportManagement

接口类: IReportManager

功能描述: : 该子系统负责对举报内容的审核、画作售后问题的解决业务

接口函数:

(1) 生成举报

createReport(Report report):void

(2) 查看举报

getReport(string reportID):Report

(3) 设置举报结果

setReportResult(string reportID,string result, string administrationID):void

(4) 删除举报

`deleteReport(string reportID):void`

2.2.7 申请管理系统

子系统名称: RequestManagement

接口类: IRequestManager

功能描述: 该子系统完成与申请相关的业务

功能描述:

接口函数:

(1) 生成申请

`createRequest(Request request):void`

(2) 获取申请信息

`getRequest(string requestID):Request`

(3) 设置申请结果

`setRequestResult(string requestID,string result,string administrationID):void`

(4) 删除申请

`deleteRequest(string requestID):void`

2.2.8 交易管理系统

子系统名称: TransactionManagement

接口类: ITransactionManager

功能描述: 该子系统处理画作买卖、定制方面的业务

接口函数:

(1) 购买画作

`BuyPainting(string userID, string paintingID):void`

(2) 退货

`Return(string OrderID):void`

(3) 修改订单

`ModifyOrder(string OrderID):bool`

(4)取消订单

OrderCancellation(string OrderID):bool

(5)获得所有订单

GetAllOrders(string UserID):list

2.2.9 数据库系统

子系统名称: DataBaseSystem

接口类: IDatabaseSystem

功能描述: 该系统负责为其他子系统提供必要的数据库

接口函数:

(1) 创建用户信息

createUserInfo(string nickName,string password,string phone):string

(2) 修改用户信息

modifyUserInfo(string userID,string infoType,string info):bool

(3) 获取用户信息

getUserInfo(string userID):User

(4) 删除用户

deleteUser(string userID):bool

(5) 创建订单信息

createOrderInfo(string userID,string paintingID):string

(6) 修改订单信息

modifyOrderInfo(string orderID,string infoType,string info):bool

(7) 获取订单信息

getOrderInfo(string orderID): Orders

(8) 删除订单

deleteOrder(string orderID):bool

(9) 创建画作信息

createPaintingInfo(Paintings paintings):string

(10) 获取画作信息

getPaintingInfo(string paintingID):Paintings

(11) 修改画作信息

modifyPaintingInfo(string paintingID,string infoType,string info):bool

(12)删除画作

deletePainting(paintingID):bool

(13)创建举报信息

createReportInfo(Report report):string

(14) 删除举报

deleteReport(string reportID):bool

(15) 创建申请信息

createRequestInfo(Request request):string

(16) 删除申请

deleteRequest(string requestID):bool

(17) 创建画展信息

createPEInfo(string painterID, PaintingExhibitionRoom paintingexhibitionroom):
string

(18) 修改画展信息

modifyPEInfo(string painterID,string roomID,string infoType,string info):bool

(19) 删除画展

deletePE(string painterID,string roomID):bool

(20) 获取画展信息

getPEInfo(string painterID,string roomID):PaintingExhibitionRoom

(21) 创建直播信息

createLSInfo(string painterID,LiveRoom liveroom):string

(22) 修改直播信息

modifyLSInfo(string painterID,string infoType,string info):bool

(23) 删除直播

deleteLS(string painterID):bool

(24) 获取直播信息

getPEInfo(string painterID):LiveRoom

2.3 第三方平台接口规范——第三方快递平台

本系统为完成画作交易的顺利进行，我们和第三方平台顺丰快递进行交互，以实现快递的发送、实时查询与运输状态、确认签收等功能。

通过阅读顺丰平台的API接入方案、第三方软件接入方案等文档，我们设计了本系统与顺丰平台的API之间的接口。

申请API接口：<https://qiao.sf-express.com/index.html>

2.3.1 路由注册接口

此接口主要是连接顺丰快递的路由注册功能，向客户推送物流信息报文。

- 接口基本信息

名称	值
批量交易	不支持
接口类型	接入
报文类型	JSON

- 元素<请求>msgData

元素名	类型（约束）	必填	默认值	描述
type	String(1)	是	1	1-按订单号注册, 2-按运单号注册
attributeNo	String(100)	是		订单号或者运单号
checkPhoneNo	String(20)	否		电话号码后四位
orderId	String(100)	否		
clientIp	String(60)	否		

元素名	类型（约束）	必填	默认值	描述
language	String(20)	否	0	0：中文 1：英文 2：繁体
address	String(20)	否		地址信息

- 请求报文（**msgData**字段）：

```
{
  "attributeNo": "SF1040300507426",
  "type": "2"
}
```

- 元素<响应>**Response**

属性名	类型	是否必传	描述
apiResultCode	String(10)	是	API平台结果代码
apiErrorMsg	String(200)	是	API平台异常信息
apiResponseID	String(40)	是	API响应唯一标识UUID
apiResultData	String	否	业务处理详细结果

- 元素<响应>**apiResultData**

元素名	类型(约束)	必填	默认值	描述
success	boolean	是		成功返回true;失败返回false
errorCode	String	否		失败原因代码，如4001
errorMsg	String	否		失败原因，失败时记录具体原因，如系统异常
msgData	String	否		成功时返回的订单数据数据

- 响应报文

成功相应:

```
{
  "apiErrorMsg": "",
  "apiResponseID": "000178D44ED4BF3FD4846AA604E0763F",
  "apiResultCode": "A1000",
  "apiResultData": {
    "success": true,
    "errorCode": "S0000",
    "errorMsg": null,
    "msgData": null
  }
}
```

订单处理失败:

```
{
  "apiErrorMsg": "",
  "apiResponseID": "00016ABEC9ECCB3FE1C04106BA87EF3F",
  "apiResultCode": "A1000",
  "apiResultData": "{***}"
}
```

2.3.2 路由查询接口

客户可通过此接口查询顺丰运单路由，顺丰会在响应Json报文返回当时点要求的全部路由节点信息。

- 接口基本信息

名称	值
接口服务代码	EXP_RECE_SEARCH_ROUTES
批量交易	最多10个tracking_number
接口类型	接入
协议类型	HTTP/POST
报文类型	JSON/XML

- 元素<请求> QuerySFRoute

字段说明：

属性名	类型(约束)	必填	默认值	描述
language	Number(1)	否	0	返回描述语语言 0: 中文 1: 英文 2:繁体 指错误描述语言， 非路由数据语言
trackingType	Number(2)	否	1	查询号类别： 1:根据顺丰运单号查询， trackingNumber 将被当作 顺丰运单号处理 2:根据客户订单号查询， trackingNumber 将被当作 客户订单号处理
trackingNumber	List	是		查询号: trackingType=1, 则此值为顺丰运 单号如果trackingType=2, 则此值为客户订单号
methodType	Number(1)	否	1	路由查询类别： 1:标准路由查询 2:定制路由查询
checkPhoneNo	String(30)	否	0	电话号码验证 0:不需要认证 1: 需要认证

• JSON请求示例：

请求报文1:

```
{
  "language": "0",
  "trackingType": "1",
  "trackingNumber": ["444003077898", "441003077850"],
  "methodType": "1",
}
```

请求报文2:

```
{
  "language": "0",
  "trackingType": "1",
  "trackingNumber": ["444003077898"],
  "methodType": "1",
  "checkPhoneNo": "3809"
}
```

请求报文3:

```
{
  "language": "0",
  "trackingType": "1",
  "trackingNumber": ["运单号1", "运单号2", "运单号3"],
  "methodType": "1",
  "checkPhoneNo": "电话号码后四位_1, 电话号码后四位_2, 电话号码后四位_3"
}
```

- 元素<响应> **QuerySFRouteResponse**:

字段说明:

属性名	类型	必填	描述
success	String	是	true 请求成功, false 请求失败
errorCode	String	是	错误编码, S0000成功
errorMsg	String	是	错误描述
routeResps	List	是	顺丰运单号对应的路由详情 (列表)
mailNo	String	是	顺丰运单号
routes	List	是	路由信息 (列表)
acceptTime	Date	是	路由节点发生的时间, 格式: YYYY-MM-DD HH24:MM:SS, 示例: 2012-7-30 09:30:00
acceptAddress	String(100)	否	路由节点发生的地点
remark	String(150)	是	路由节点具体描述
opCode	String(20)	是	路由节点操作码

- **JSON响应报文示例——路由查询成功:**

```
{
  "success": true,
  "errorCode": "S0000",
  "errorMsg": null,
  "msgData": {
    "routeResps": [{
      "mailNo": "SF1011603494291",
      "routes": [{
        "acceptTime": "2019-05-09 10:11:26",
        "acceptAddress": "深圳",
        "opCode": "50",
        "remark": "已派件"
      },
      {
        "acceptTime": "2019-05-09 18:11:26",
        "acceptAddress": "深圳",
        "opCode": "80",
        "remark": "已签收"
      }
    ]
  }
}
```

2.3.3 数据传输MD5加密

为保证接口中信息传递的安全性，我们采用MD5进行加密，认证码生成实例如下(Java):

```
import java.io.FileInputStream;
import java.io.InputStream;
import java.security.MessageDigest;
import sun.misc.BASE64Encoder;

public class Util {
    public static String loadFile(String fileName) {
        InputStream fis;
        try {
            fis = new FileInputStream(fileName);
            byte[] bs = new byte[fis.available()];
            fis.read(bs);
            String res = new String(bs);
            fis.close();
            return res;
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```

```

    public static String md5EncryptAndBase64(String str) {
        return encodeBase64(md5Encrypt(str));
    }

    private static byte[] md5Encrypt(String encryptStr) {
        try {
            MessageDigest md5 = MessageDigest.getInstance("MD5");
            md5.update(encryptStr.getBytes("utf8"));
            return md5.digest();
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }

    private static String encodeBase64(byte[] b) {
        sun.misc.BASE64Encoder base64Encode = new BASE64Encoder();
        String str = base64Encode.encode(b);
        return str;
    }

    public static void main(String[] args) {
        String json= loadFile(args[0]);
        String checkword = loadFile(args[1]);
        String timeStamp = String.valueOf(System.currentTimeMillis());
        System.out.println(md5EncryptAndBase64(URLEncoder.encode((json +
timeStamp + checkword), "UTF-8")));
    }
}

```

2.3.4 API接口函数调用

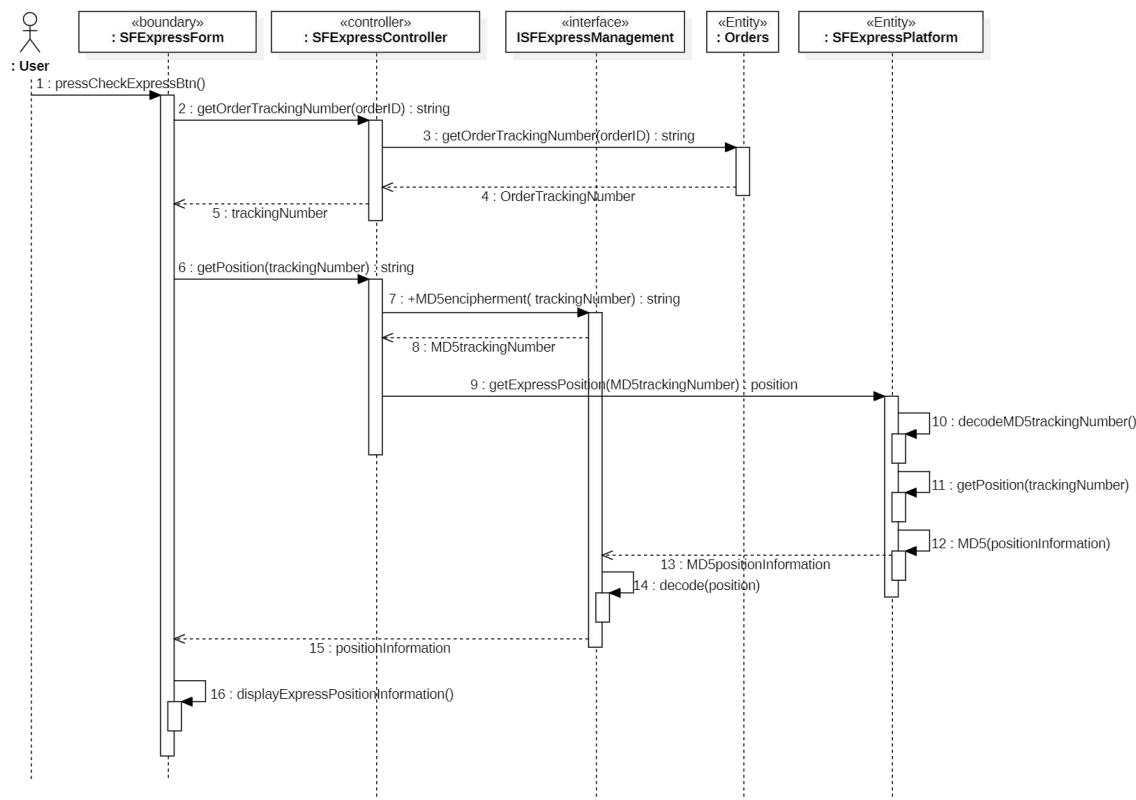
```

requestJSON = getExpressCreateServiceRequestJSON(params);
methodName = "expressCreateSerivce";
requestJSON = getExpressSearchServiceRequestJSON(params);
methodName = "expressSearchService";
requestJSON = getExpressRouteServiceRequestJSON(params);
methodName = "expressRouteService";
// post request
SFExpressResponse response =
(SFExpressResponse) JSONUtil.convertToObject(SFExpressResponse.class, re
sultStr);

```

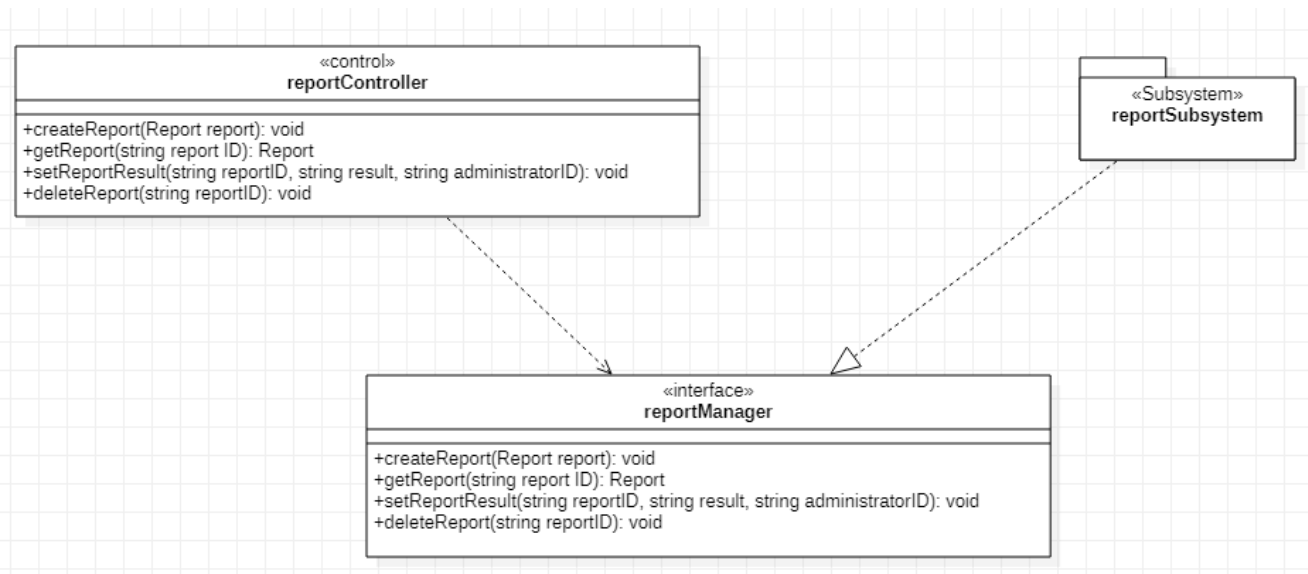
2.3.5 接口调用示例

快递查询顺序图：



2.4 子系统示范与接口明确

举报管理子系统有接口类ReportManager，包含四个接口函数，主要负责对数据库中举报的增删改查以及审查。



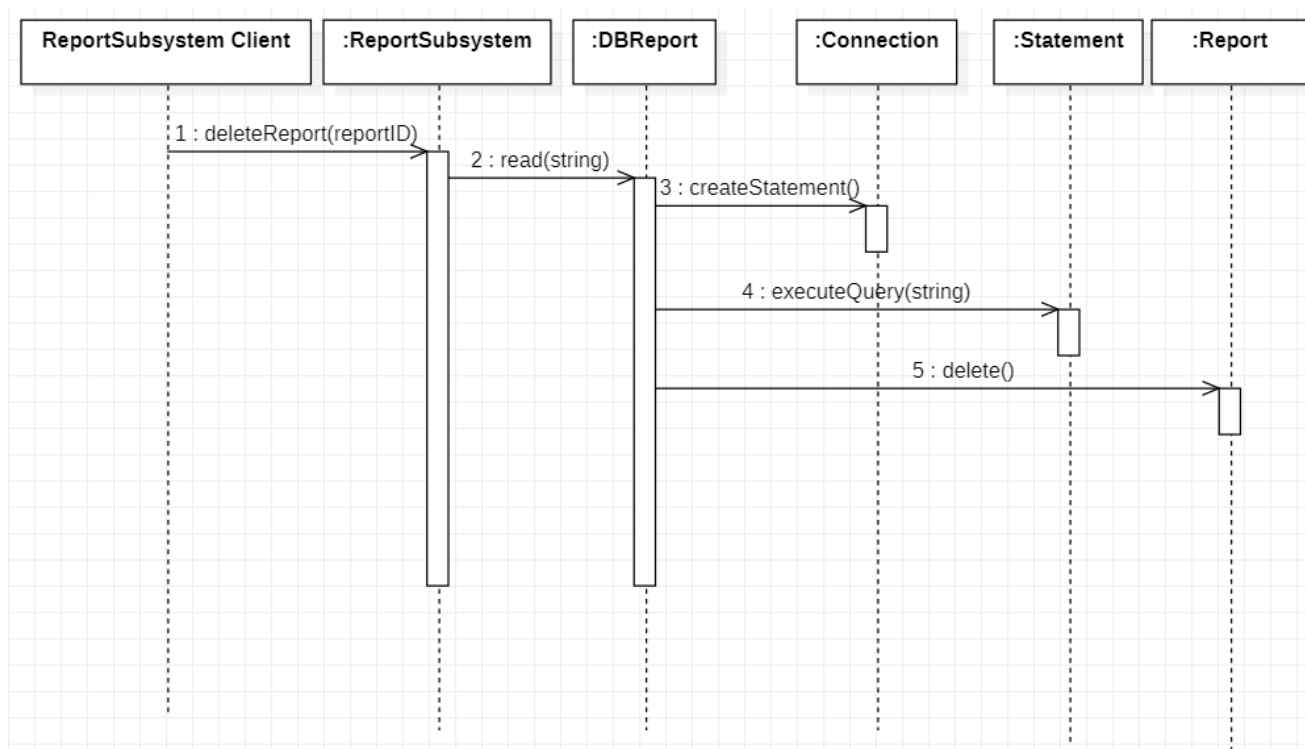
createReport(string reporterID,string reportedID,string content,string materials):void，该接口在用户进行举报中被实现，当用户填好举报表单后点击举报按钮后被调用，为用户上传举报信息；

`getReport(string reportID):report`，该接口在审核举报中被实现，当管理员点击某一举报的查看按钮后被调用，为管理员提供举报信息。它将接收举报id，然后将带有举报id的HTTP请求发送到后端API。API把举报id集成到SQL中并在数据库中进行查询，并将查询结果作为json对象返回，该接口函数对其分析后返回举报信息。

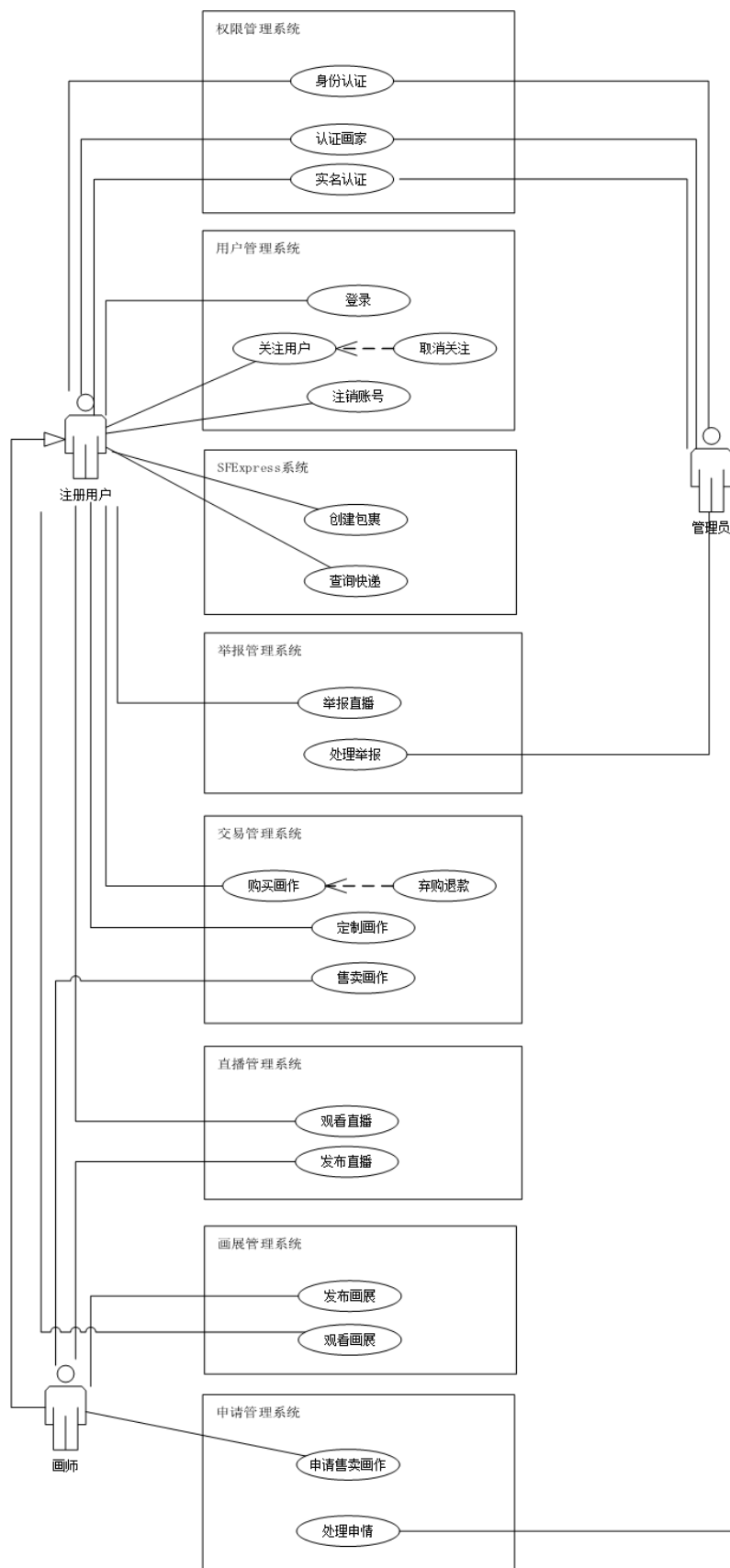
```
{  
  "reportID": "reportIDNumber",  
  "reporterID": "reporterIDNumber",  
  "reportedID": "reportedIDNumber",  
  "content": "content",  
  "materials": "materials",  
  "createTime": "createTime",  
  "dealStatus": true/false,  
}
```

`setReportResult(string reportID,string result,string administrationID):void`，该接口在审核举报中被实现，当管理员在查看某一举报后点击通过或驳回按钮后被调用，为管理员上传举报结果信息。将接收举报id和举报结果，然后将带有举报id和举报结果的HTTP请求发送到后端API。API把举报id集成到SQL中并在数据库中进行查询并更新举报结果。

`deleteReport(string reportID):void`，该接口在审核举报中被实现，当管理员在查看某一举报后点击删除按钮后被调用，提供给管理员使用，为管理员删除无效举报信息。它将接收举报id，然后将带有举报id的HTTP请求发送到后端API。API把举报id集成到SQL中并在数据库中进行查询并删除。



三、用例图更新

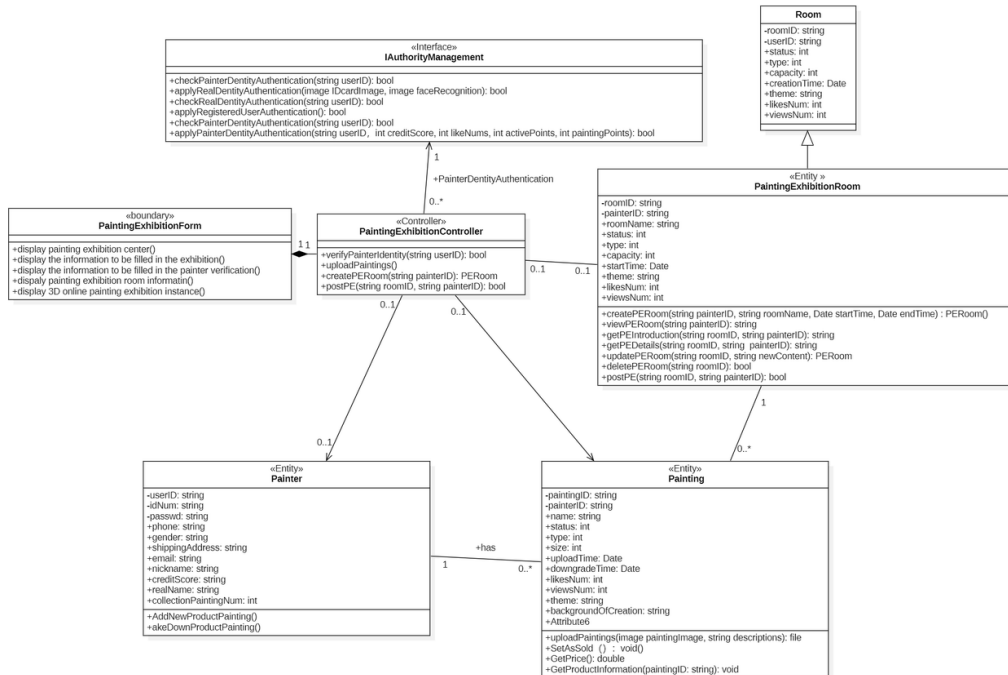


四、设计模型

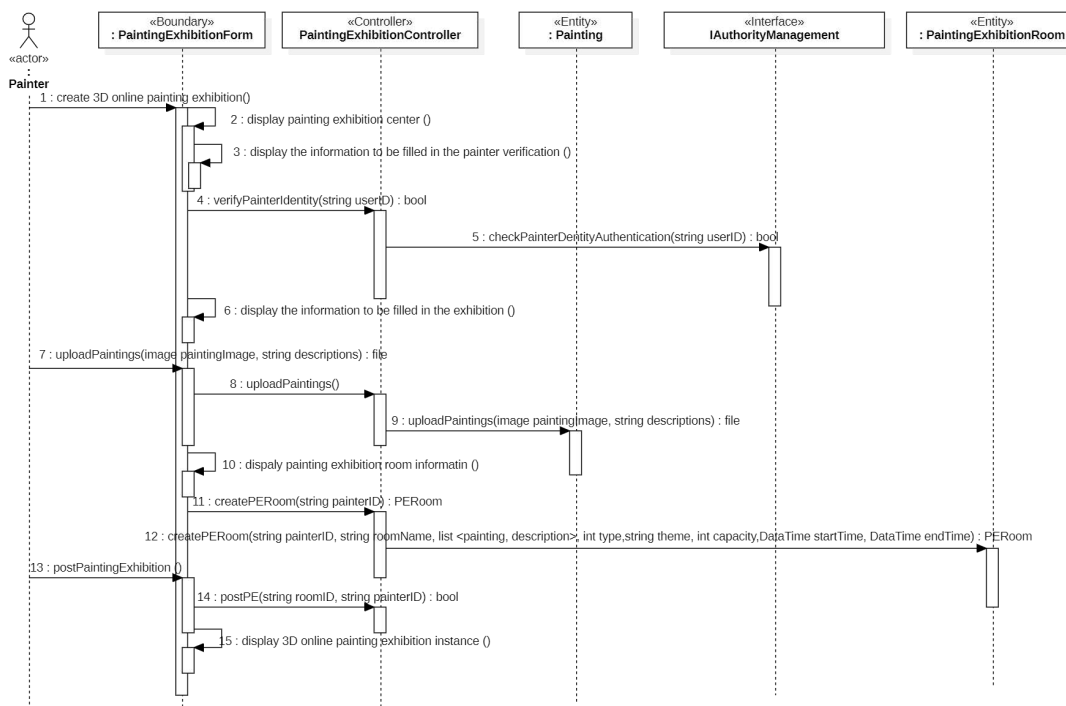
4.1 五个详细的用例实现

4.1.1 举办画展用例实现

- 举办画展类图

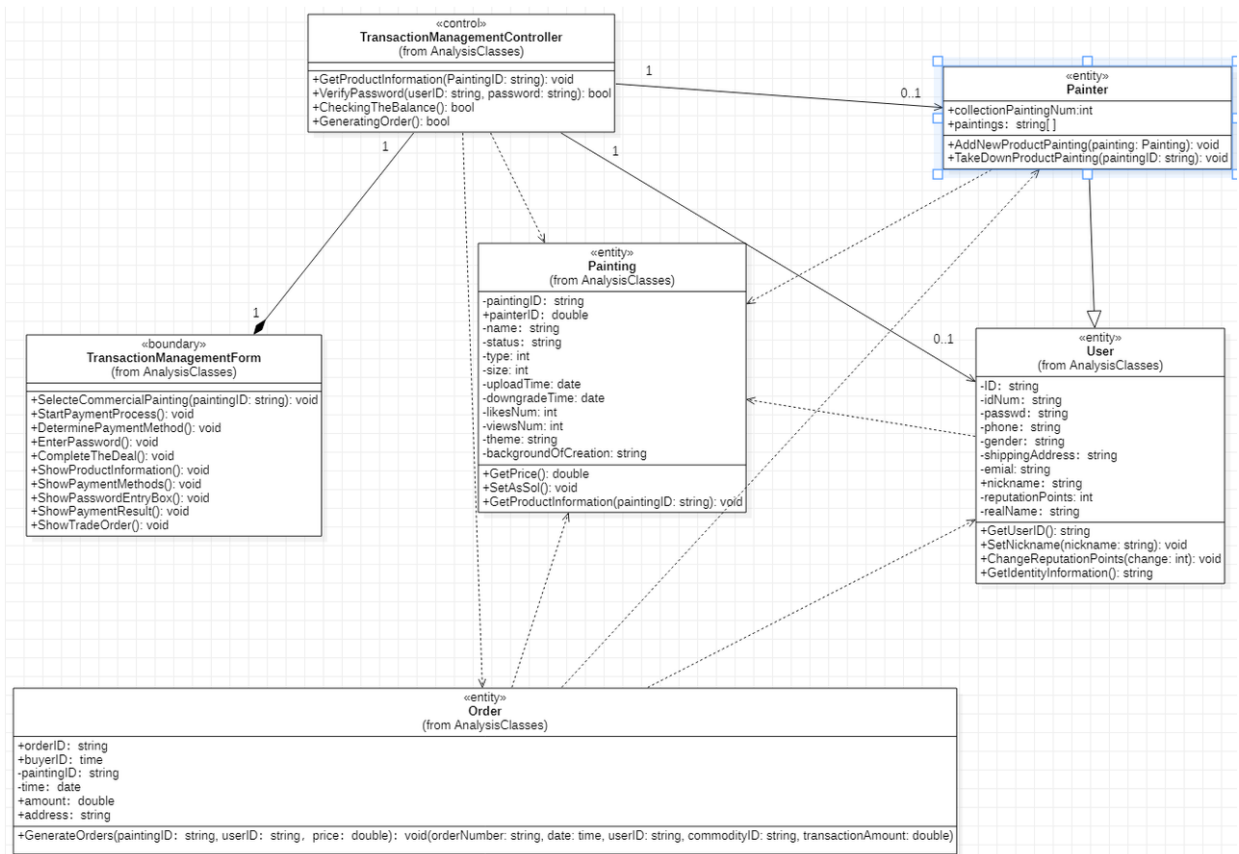


- 举办画展时序图

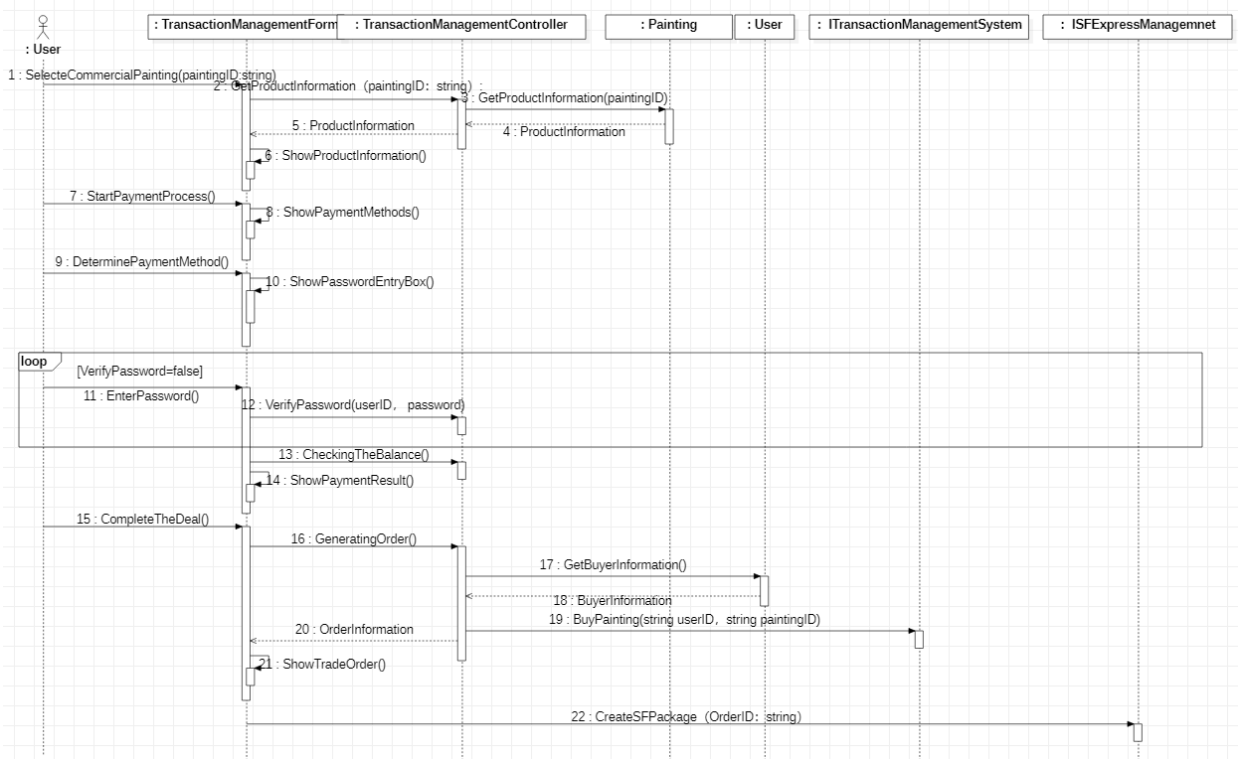


4.1.2 购买画作用例实现

• 购买画作类图

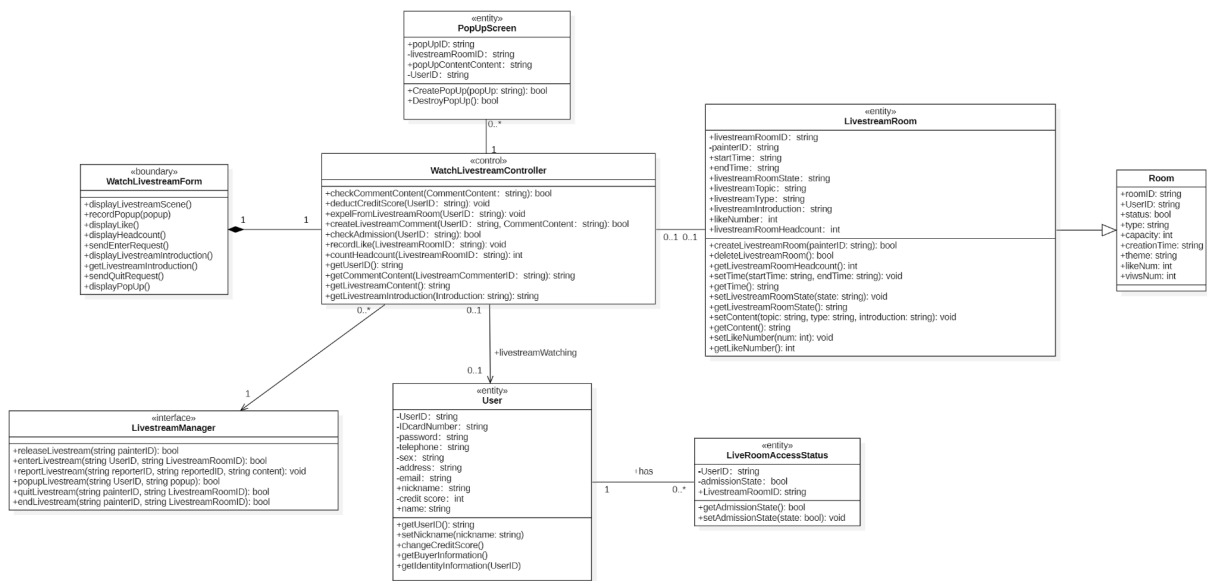


• 购买画作时序图

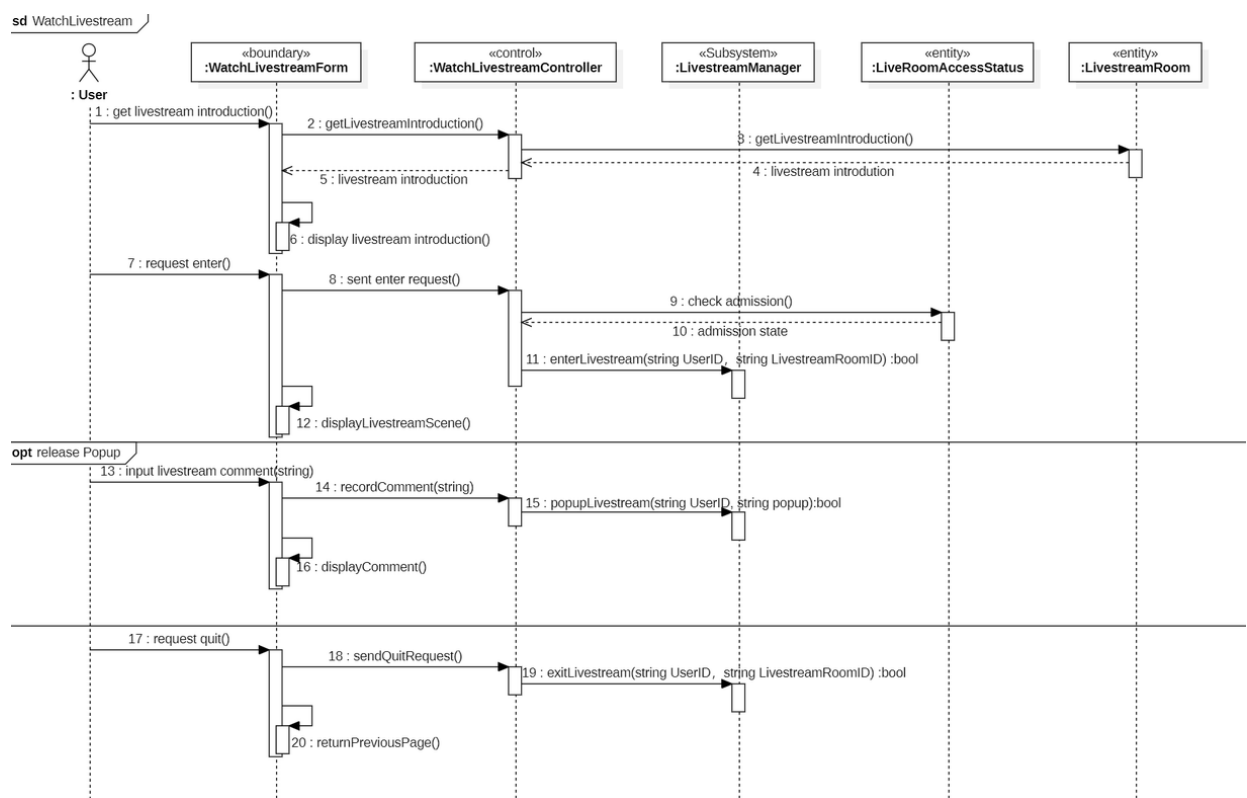


4.1.3 观看直播用例实现

- 观看直播类图

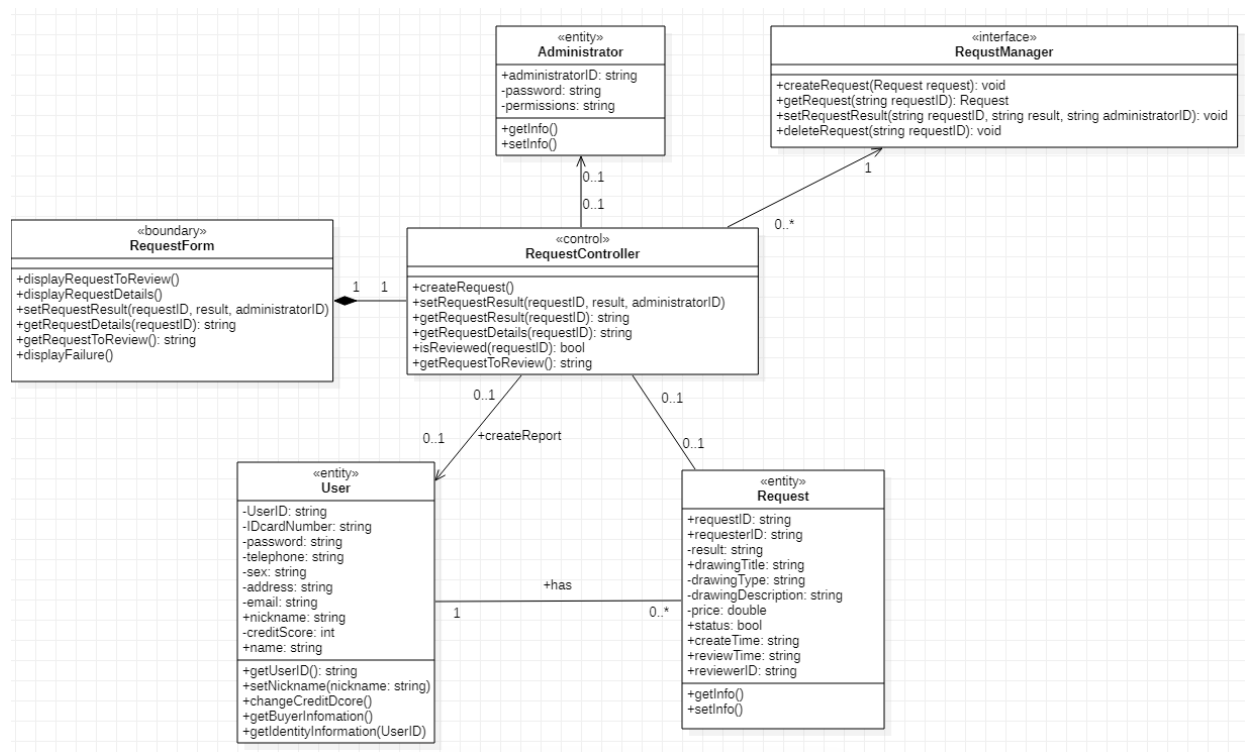


- 观看直播时序图

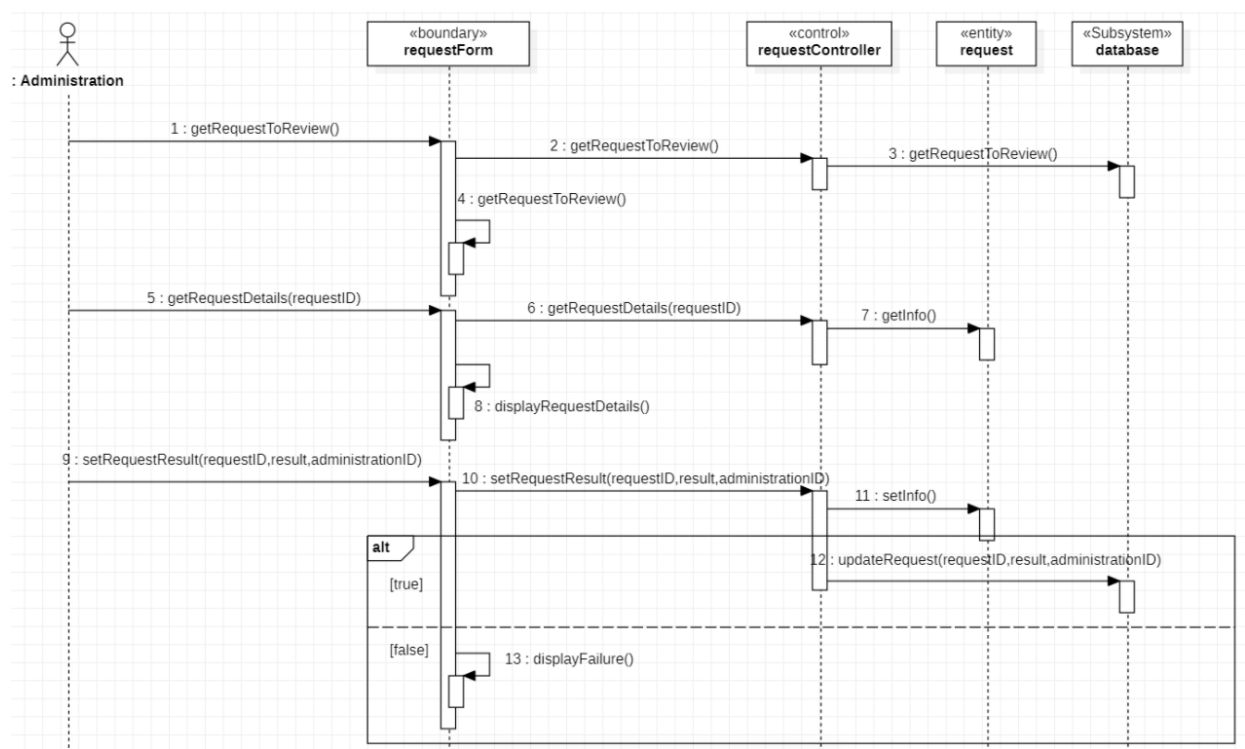


4.1.4 审核申请用例实现

- 审核申请类图

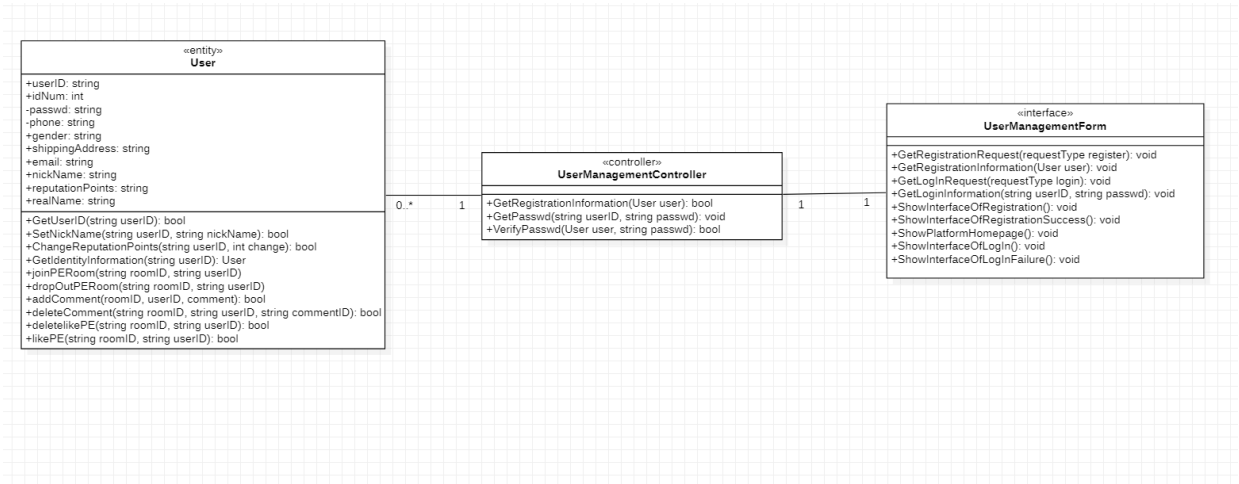


- 审核申请时序图

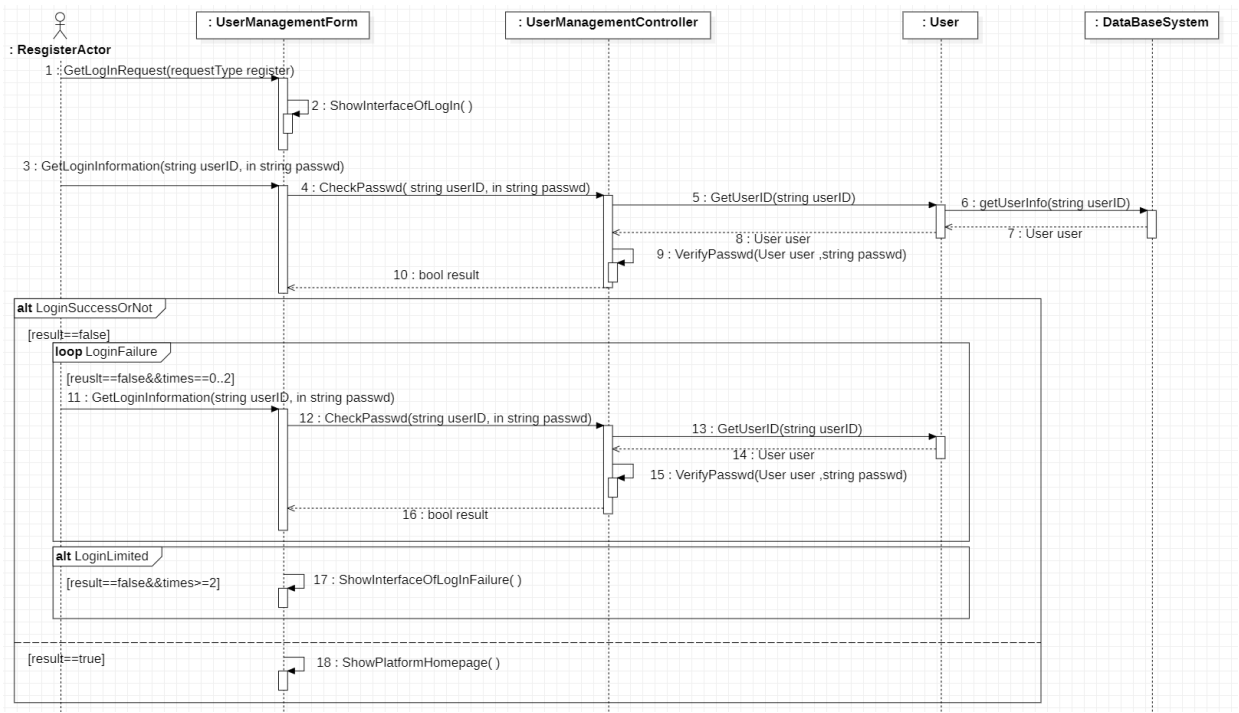


4.1.5 登录账号用例实现

- 登陆账号类图



- 登陆账号时序图



4.2 类设计

Class Name	Attribute	Methods
Entity		
User	userID, idNum, passwd, phone, gender, shippingAddress, email, nickName, reputationPoints, realName	+getUserID(string userID): bool +setNickname(string userID, string nickName): bool +changeReputationPoints(string userID, int change): bool +getIdentityInformation(string userID): User +joinPERoom(string roomID,string userID):bool +dropOutPERoom(string roomID,string userID):bool +addComment(roomID, userID, comment) : bool +deleteComment(string roomID, string userID,string commentID) : bool +deletelikePE(string roomID,string userID) : bool +likePE(string roomID,string userID) : bool
BrowsingHistory	userID, browsingHistory	+getBrowsingHistory(string userID):string +deleteBrowsingHistory(string userID,string browseHistory):bool
Painter	userID, idNum, passwd, phone, gender, shippingAddress, email, nickname, creditScore, realName, collectionPaintingNum	+addNewProductPainting(painting:Painting):void +takeDownProductPainting(paintingID:string): void
Administractor	administratorID, passport, permissions	+getInfo() +setInfo()
Report	reportID, whistleblowerID, reportedPersonID, content, result, material, generatedTime, processingStatus, processingPersonID, processingTime	+getInfo() +setInfo()
Application	applicationID, applicantID, paintingName, paintingType, paintingDescription, price, result, generationTime, reviewStatus, reviewerID, reviewTime	+getInfo() +setInfo()
Painting	paintingID, painterID, name, status, type, size, uploadTime, downgradeTime, likesNum, viewsNum, theme, backgroundOfCreation	+setAsSold():void +getPrice():double +getProductInformation(paintingID string):void +uploadPaintingImage() : bool

Class Name	Attribute	Methods
Order	sellerID, buyerID, paintingID, time, amount, address	+generateOrders(paintingID :string, userID :string, price: double): void
Room	roomID, userID, status, type, capacity, creationTime, theme, likesNum, viewsNum	
PaintingExhibitionRoom	roomID, painterID, roomName, status, type, capacity, startTime, theme, likesNum, viewsNum	+viewPERoom(string painterID) : string +getPEIntroduction(string roomID, string painterID) : string +getPEDetails(string roomID, string painterID) : string +createPERoom(string painterID, string roomName, list <painting, description>, int type,string theme, int capacity,dataTime startTime, DataTime endTime) : PERoom +updatePERoom(string roomID, string newContent) : PERoom +deletePERoom(string roomID) : bool +postPE(string roomID, string painterID) : bool
LivestreamRoom	livestreamRoomID, painterID, startTime, endTime, livestreamRoomState, livestreamTopic , livestreamType, livestreamintroduction, likeNumber, livestreamRoomHeadcount	+createLivestreamRoom(painterID:string): bool +deleteLivestreamRoom():bool +getLivestreamRoomHeadcount(): int +setTime(startTime:string, endTime:string): void +getTime(): string +setLivestreamRoomState(state:string): void +getLivestreamRoomState(): string +setContent(topic:string, type:string, in +introduction:string): void +getContent(): string +setLikeNumber(num:int): void +getLikeNumber(): int
LiveRoomAccessStatus	userID, admissionState, LivestreamRoomID	+getAdmissionState():bool +setAdmissionState():bool
PopUpScreen	livestreamRoomID, userID, popUpID, popUpContent	+createPopUp(string popUp):bool +destroyPopUp():bool
Comment	userID, commentContent	+createComment(string comment):bool +destroyComment():bool

Controller

UserManagementController	+getRegistrationInformation(User user):bool +getPasswd(string userID,string passwd):void +verifyPasswd(User user,string passwd):bool
TransactionManageController	+getProductInformation (paintingID: string) :void +verifyPassword(userID, password): bool +checkingTheBalance(): bool +generatingOrder(): bool
PaintingExhibitionController	+verifyPainterIdentity(string userID): bool +uploadPaintings() +createPERoom(string painterID): PERoom +postPE(string roomID, string painterID): bool

Class Name	Attribute	Methods
WatchLiveStreamController		+checkCommentContent(CommentContent: string):bool +deductCreditScore(userID: string):bool +expelFromLivestreamRoom(userID: string):bool +createLivestreamComment(userID: string, CommentContent: string):bool +checkAdmission(serID: string):bool +recordLike(LivestreamRoomID: string) :bool +countHeadcount(LivestreamRoomID: string):bool +getUserID():bool +getCommentContent(LivestreamCommenterID: string):bool +getLivestreamContent() :bool +getLivestreamIntroduction(Introduction:string):bool

View

UserManagementForm	+getRegistrationRequest(requestType register):void +getRegistrationInformation(User user):void +getLogInRequest(requestType login):void +showInterfaceOfRegistration():void +showInterfaceOfRegistrationSuccess():void +showPlatformHomepage():void +showInterfaceOfLogIn(): void +showInterfaceOfLogInFailure():void
TransactionManageForm	+selecteCommercialPainting(paintingID:string) +startPaymentProcess(): void +determinePaymentMethod(): void +enterPassword(): void +completeTheDeal(): void +showProductInformation(): void +showPaymentMethods(): void +showPasswordEntryBox(): void +showPaymentResult(): void +showTradeOrder(): void
PaintingExhibitionForm	+displayPaintingExhibitionCenter():void +displayThe InformationToBeFilledInTheExhibition():void +displayTheInformationToBeFilledInThePainterVerification():void +dispalyPaintingExhibitionRoomInformatin():void +display3DOnlinePaintingExhibitionInstance():void
LiveStreamForm	+displayLivestreamScene() :void +recordPopup(in popup) :void +displayLike():void +displayHeadcount():void +sendEnterRequest():void +displayLivestreamIntroduction():void +getLivestreamIntroduction():void +sendQuitRequest():void +displayPopUp():void

五、组员分工

在完成本次任务的过程中，所有组员积极参与讨论，努力完成自己的任务。组员之间配合融洽，遇到问题及时沟通、共同解决。组内分工明确，具体如下：

- 项目简介：由杨茗溟书写
- 平台架构图：由梁伊雯构思并书写
- 子系统和接口设计：经所有组员共同讨论确定划分，每个子系统接口交由个人书写，具体如下：

梁伊雯——直播管理系统

赵敏 ——举报管理系统、申请管理系统

黄金坤——交易管理系统

杨茗溟——用户管理系统、数据库系统

李林飞——权限管理系统、画展管理系统、SFExpress子系统

- 第三方平台接口规范：由李林飞构思并书写
- 子系统示范及接口明确：由赵敏构思并书写
- 五个详细用例实现：小组五人每个人一个，均为对分析文档中用例实现的改进

梁伊雯——观看直播用例实现

赵敏 ——审核申请用例实现

黄金坤——购买画作用例实现

杨茗溟——登陆账号用例实现

李林飞——举办画展用例实现

- 类设计：由小组成员共同设计，并由黄金坤整理、归纳并书写
- 文档排版：由杨茗溟整合美化

得分权重：

学号	姓名	贡献比例
1951095	梁伊雯	20%
1952211	黄金坤	20%
1951976	李林飞	20%

学号	姓名	贡献比例
1953714	杨茗溟	20%
1951326	赵敏	20%