

# 空中画园——线上画作分享交易管理系统

- 学号：1951976
- 姓名：李林飞

## Assignment 1

工作：

用例规约：举办画展 & 发布直播

活动图：发布直播

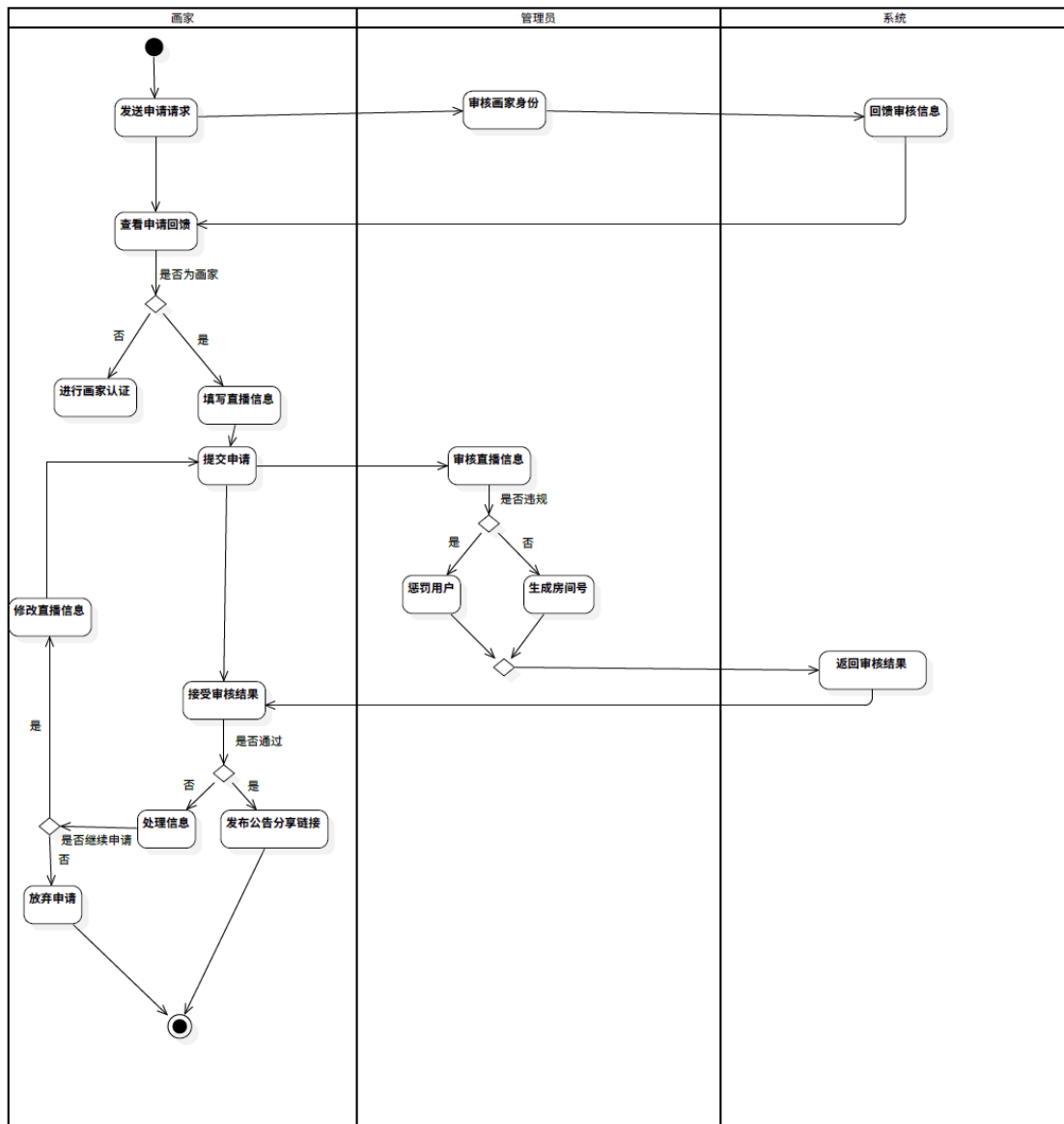
其他：整合与美化文档

- 用例规约

用例名称	举办画展
用例编号	UC014
用例说明	该用例允许画师上传3D画廊模型举办线上画展。
参与者	画师、普通用户、管理员
前置条件	画师已经完成了画师认证，获得画师权限。
基本事件流	1、画师进入个人画展管理界面，并向系统发送申请举办线上画展的请求。
	2、管理员接收到申请，审核申请者身份，系统打开画展申请信息界面。
	3、画师根据界面引导填写画展信息，并上传画廊3D模型。
	4、管理员根据相应的标准审核画展信息，通过后系统将对3D画廊模型进行渲染烘培，并将生成画展房间号返回给画师。
	5、画师收到画展房间号，可以将其发布到公告栏或通过链接分享给其他人。
异常事件流	<b>申请者是非画师用户</b> 如果提出申请发布直播的用户身份不是画师，系统将返回提示信息，提示申请者需先进行画师认证才能申请发布直播。
	<b>画师画展信息不能通过审核</b> 如果画师申请的画展内容含有不正当的信息，系统将拒绝该申请要求，并提醒画师修改后重新提交。
	<b>银行卡绑定</b> 如果画师设置线上画展允许观众打赏、送礼等行为，则需要画师绑定银行卡，获得的收益将直接流入银行卡账号。如果画师该线上画展是公益性质的，或者只为分享，不允许观众打赏，则忽略此过程。
特殊要求	用户必须完成画师认证。
后置条件	画师成功举办了线上画展。

用例名称	发布直播
用例编号	UC008
用例说明	该用例允许已认证为画师的用户通过发布直播分享自己的创作过程或教学视频。
参与者	画师、普通用户、管理员
前置条件	画师已经完成了画师认证，获得画师权限。
基本事件流	1、画师进入个人直播管理界面，并向系统发送申请直播的请求。
	2、管理员接收到直播申请，审核申请者身份，系统打开直播申请信息界面。
	3、画师根据界面引导填写直播信息，包括内容、时间、人数等，并提交审核。
	4、管理员根据相应的标准审核直播信息，通过后生成房间号并返回到用户界面。
	5、画师收到直播房间号，可以将其发布到公告栏或通过链接分享给其他人。
	6、画师在直播开始时间前进入直播间，进行自己的直播。
异常事件流	<b>申请者是非画师用户</b> 如果提出申请发布直播的用户身份不是画师，系统将返回提示信息，提示申请者需先进行画师认证才能申请发布直播。
	<b>画师直播信息不能通过审核</b> 如果画师申请的直播含有不正当的信息，系统将拒绝该申请要求，并提示其涉及的不正当内容。
	<b>绑定银行卡</b> 如果画师该直播允许观众打赏、送礼等行为，则需要画师绑定银行卡，获得的收益将直接流入银行卡账号。如果画师该直播是公益性质的，或者只为分享，不允许观众打赏，则忽略此过程。
特殊需求	无

- 活动图



## Assignment 2

工作:

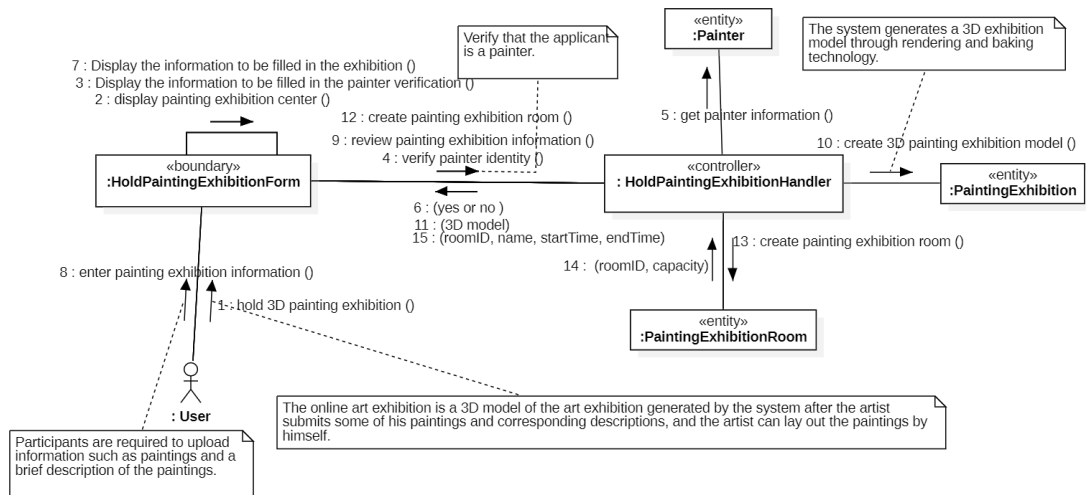
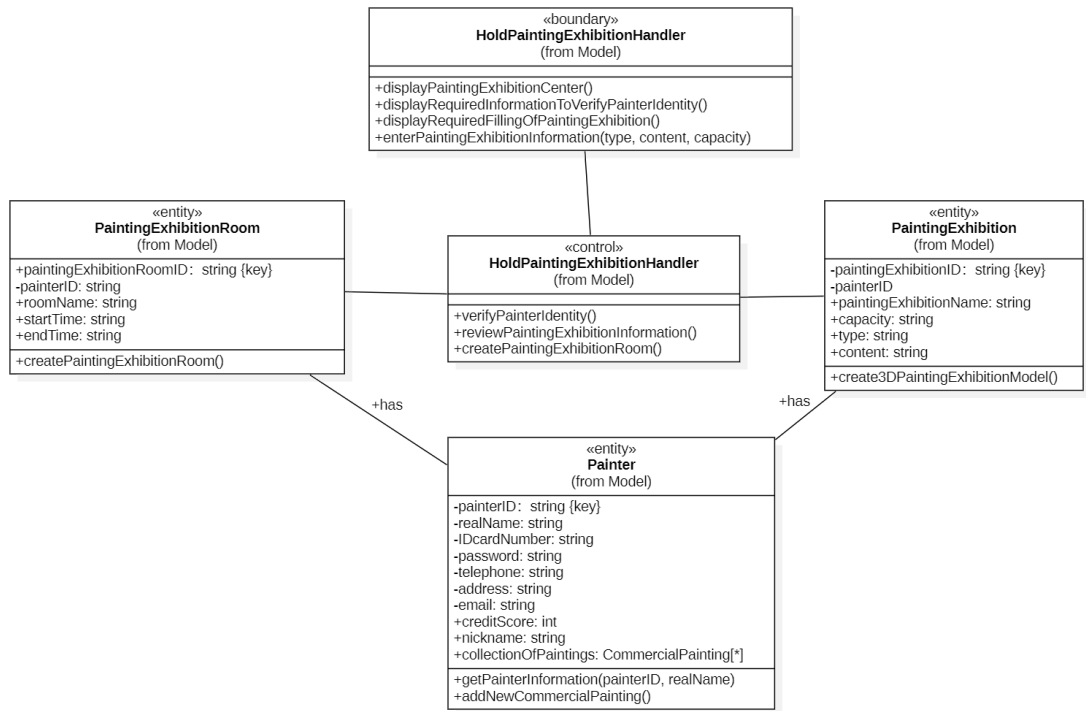
类图: 举办画展 & 发布直播

时序图: 发布直播

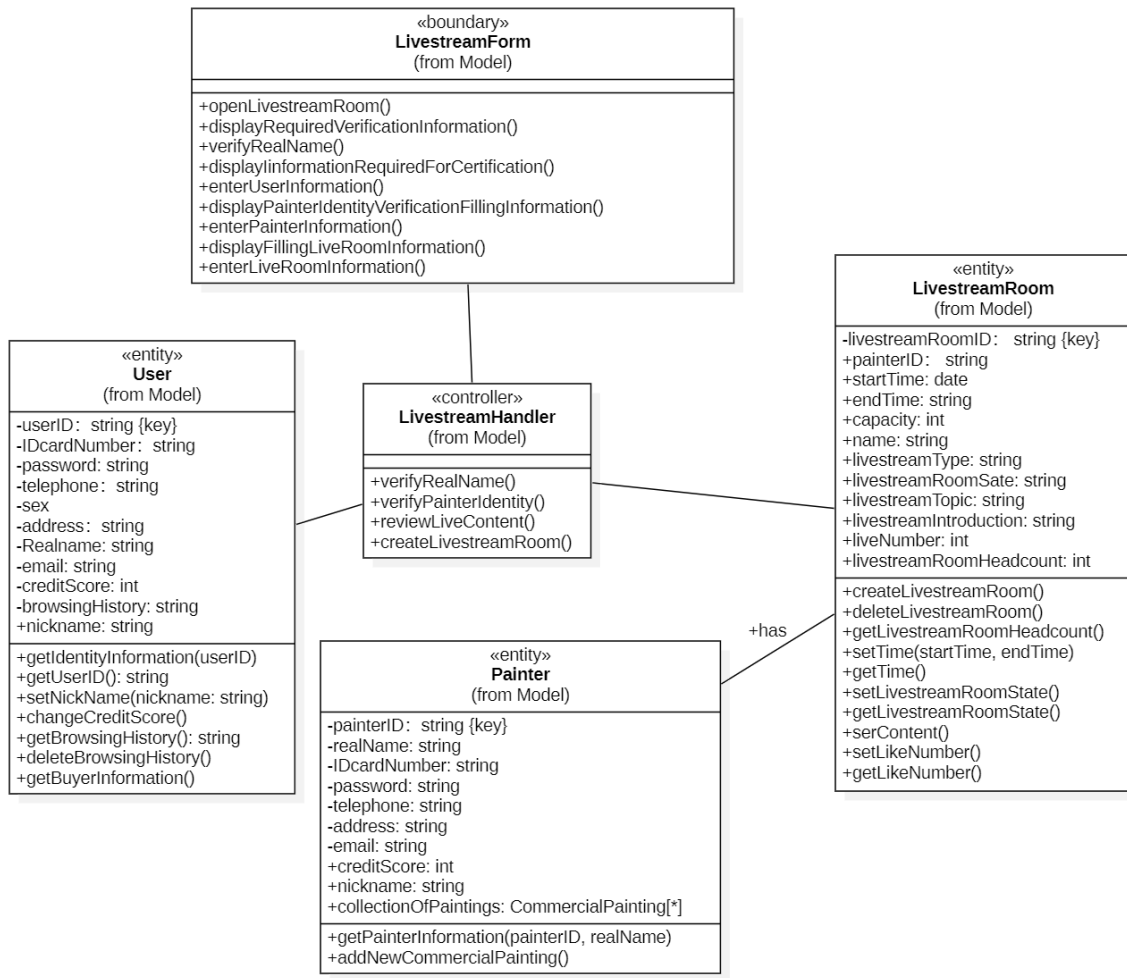
通信图: 举办画展

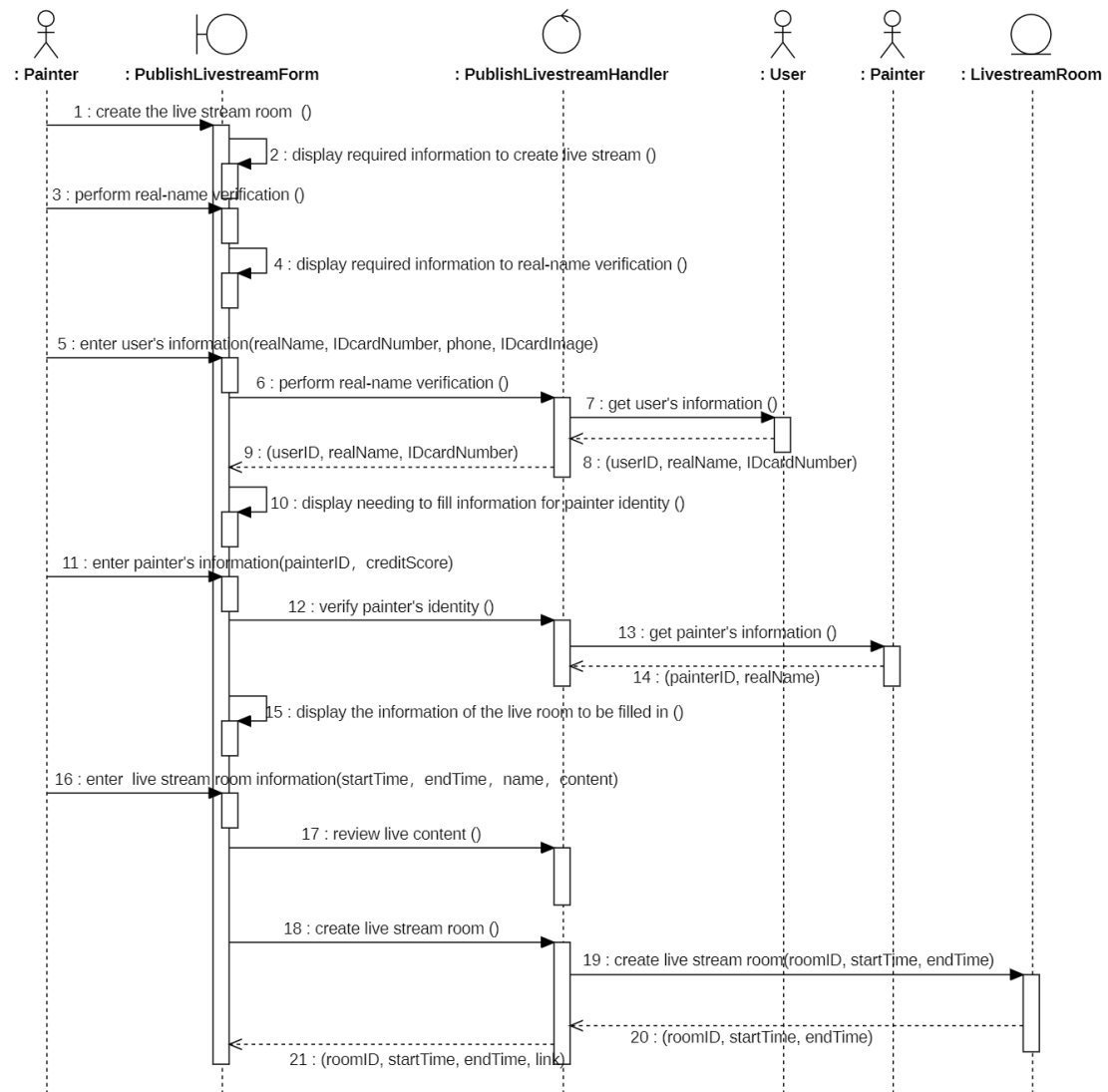
其他: 架构分析与设计

- 分析模型
  - 举办画展

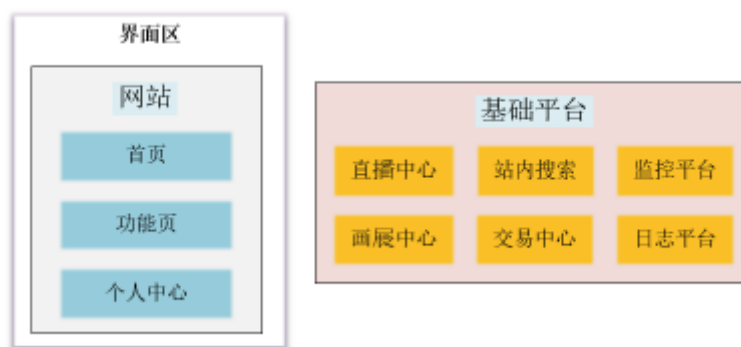


○ 发布直播



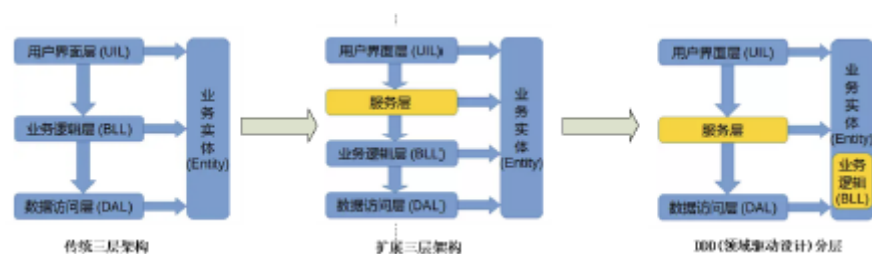


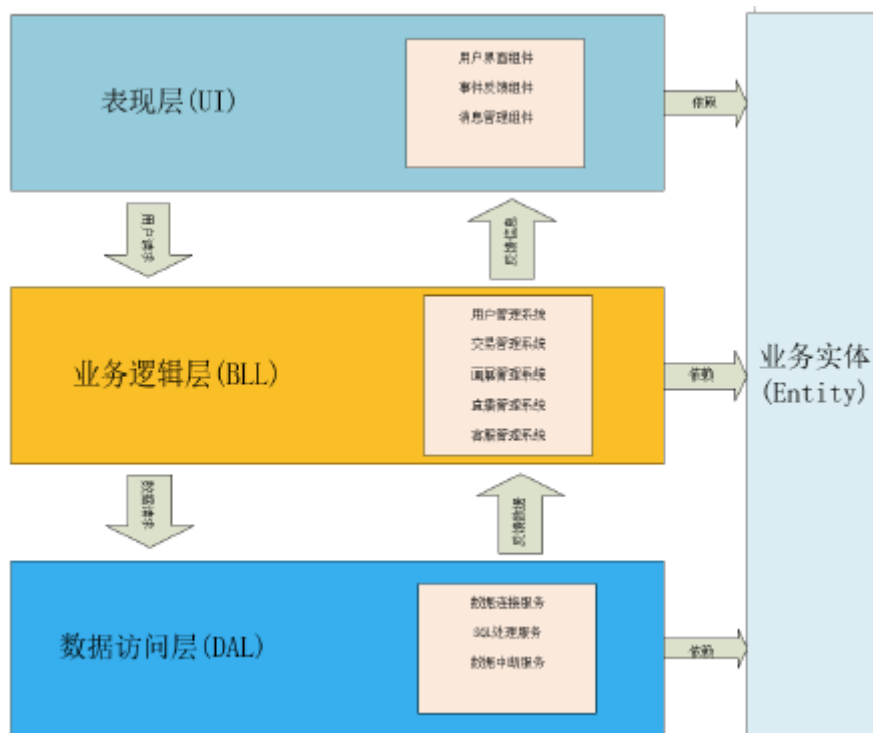
- 架构设计
  - 架构分析



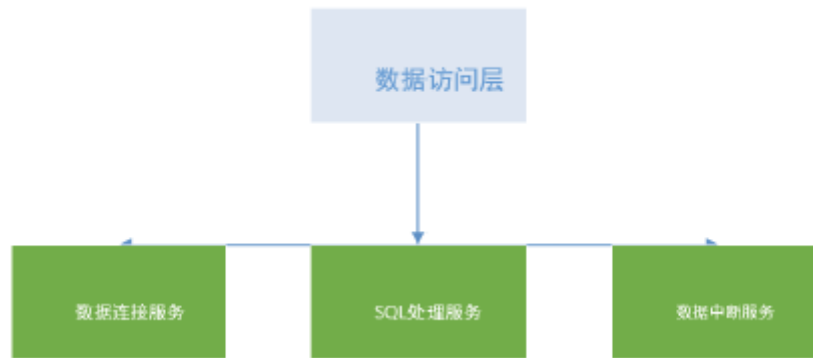


◦ 架构设计









## Assignment 3

---

### 工作

子系统：画展管理系统 & 权限管理系统 & SFExpress系统

用例实现：举办画展类图 & 举办画展时序图

其他：第三方系统接口设计规范

- 子系统
  - 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)

◦ 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

- SFExpress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

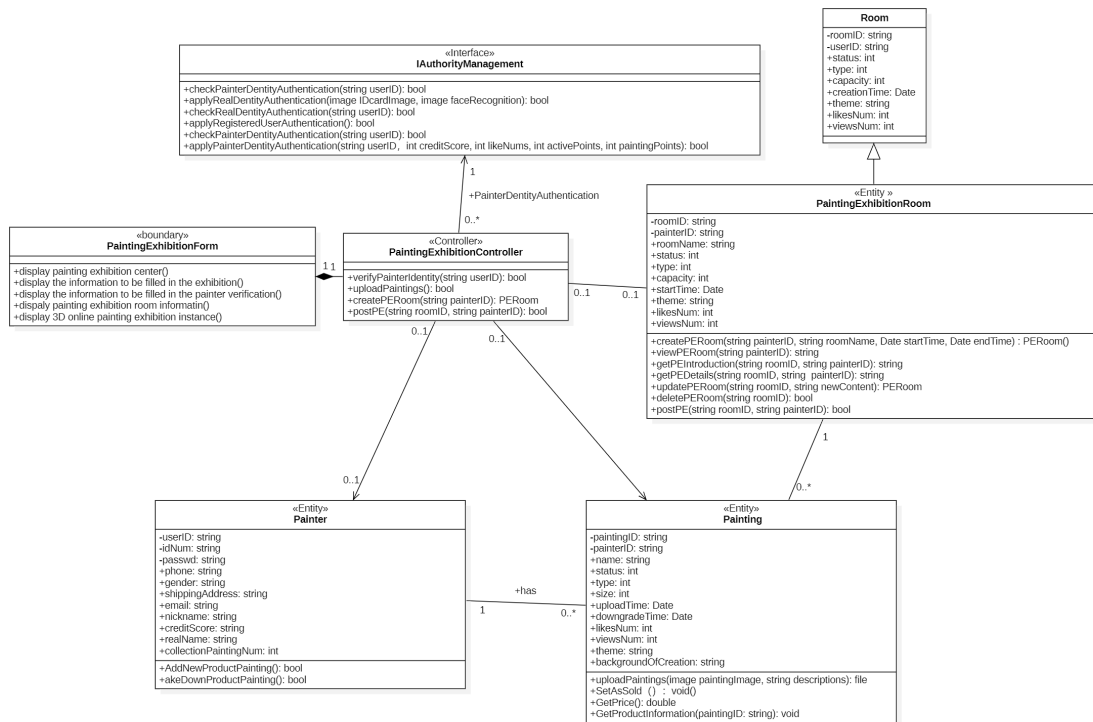
---

getHistoryInformation(string orderID) : list

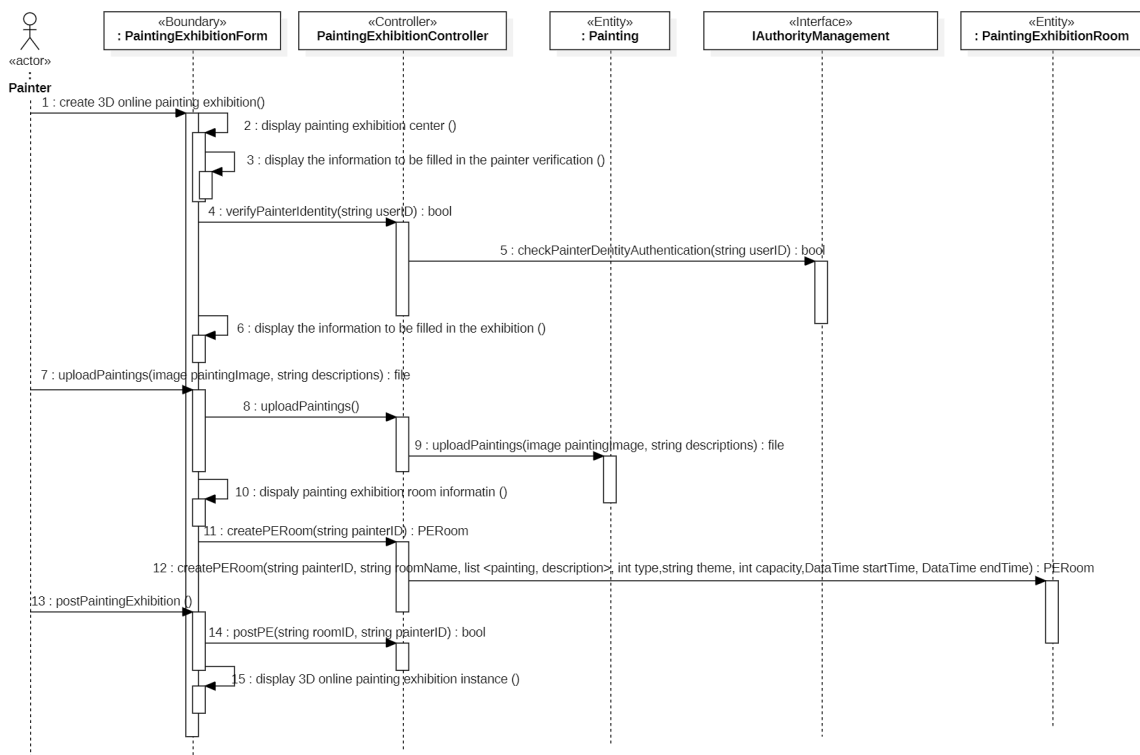
- (3) 查询快递状态

getStatus(string orderID) : string

- 用例实现
  - 类图



## 时序图



## 第三方系统接口规范设计

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

## 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，为用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

字段说明

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		
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

字段说明

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

字段说明

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:

- 成功响应:

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

单处理失败:

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

### 路由查询接口

用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 (msgData字段) :

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

- 请求报文2(msgData字段)

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

- 请求报文3(msgData字段):

```
{
  "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 (约束)	必填	默认值	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

#### 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": "上海",
        "opCode": "50",
        "remark": "已派件"
      },
      {
        "acceptTime": "2019-05-09 18:11:26",
        "acceptAddress": "上海",
        "opCode": "80",
        "remark": "已签收"
      }
    ]
  }
}
```

- 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对接口所需数据进行加密后传输, 快递后经过解密后才能正常使用。认证码生成示例如下(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"))));
    }
}

```

- 接口调用函数

```

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

```

#### 接口调用时序图

