详细设计规约

目录

[1. 引言 3](#_Toc123729765)

[1.1 编写目的 3](#_Toc123729766)

[1.2 背景及依据 3](#_Toc123729767)

[1.3 参考资料 3](#_Toc123729768)

[2. 系统软件结构 4](#_Toc123729769)

[3. 类设计 4](#_Toc123729770)

[3.1 登录注册微服务 4](#_Toc123729771)

[3.1.1 User类 5](#_Toc123729772)

[3.1.2 SignupManager类 6](#_Toc123729773)

[3.1.3 AuthentificationChecking类 8](#_Toc123729774)

[3.1.4 LoginInterface类 9](#_Toc123729775)

[3.2 数据统计微服务 10](#_Toc123729776)

[3.2.1 OrderManager类 10](#_Toc123729777)

[3.2.2 StatisticsInterface类 11](#_Toc123729778)

[3.3 用户信息微服务 11](#_Toc123729779)

[3.3.1 User 类 13](#_Toc123729780)

[3.3.2 PersonInfoManager 类 14](#_Toc123729781)

[3.3.3 PersonInfoInterface 类 16](#_Toc123729782)

[3.4 管理微服务 17](#_Toc123729783)

[3.4.1 Complaint类 18](#_Toc123729784)

[3.4.2 DeliverymanVerification类 19](#_Toc123729785)

[3.4.3 ComplaintManager类 20](#_Toc123729786)

[3.4.4 AdministrationInterface类 21](#_Toc123729787)

[3.5 订单微服务 22](#_Toc123729788)

[3.5.1 Order类 23](#_Toc123729789)

[3.5.2 Comment类 25](#_Toc123729790)

[3.5.3 Complaint类 26](#_Toc123729791)

[3.5.4 OrderManager类 28](#_Toc123729792)

[3.5.5 ComplaintManager类 31](#_Toc123729793)

[3.5.6 CommentManager类 32](#_Toc123729794)

[3.5.7 OrderInterface类 34](#_Toc123729795)

[3.6 支付微服务 35](#_Toc123729796)

[3.6.1 OrderPayInfo类 36](#_Toc123729798)

[3.6.2 OrderPaySerice类 37](#_Toc123729799)

[3.6.3 OrderPayUpdate类 38](#_Toc123729800)

[3.6.4 IorderSerice类 39](#_Toc123729801)

[3.7 交易微服务 40](#_Toc123729802)

[3.7.1 OrderInfo类 40](#_Toc123729804)

[3.7.2 OrderSerice类 41](#_Toc123729805)

[3.7.3 IOrderSerice类 43](#_Toc123729806)

# 引言

## 1.1 编写目的

本文档用于细化本系统的详细设计，通过微服务设计、接口设计、类详细设计三个角度规范系统开发中的编码方向、不同微服务之间的接口调用以及方法设计。

## 1.2 背景及依据

本项目名为飞递——校园跑腿平台，在PC端提供一个能够实现完整功能、用户体验良好的校园跑腿交易平台，用户可以通过该网站随时下单派人跑腿，也可以自己成为骑手接下他人的跑腿订单，还有其他相关的使用功能。

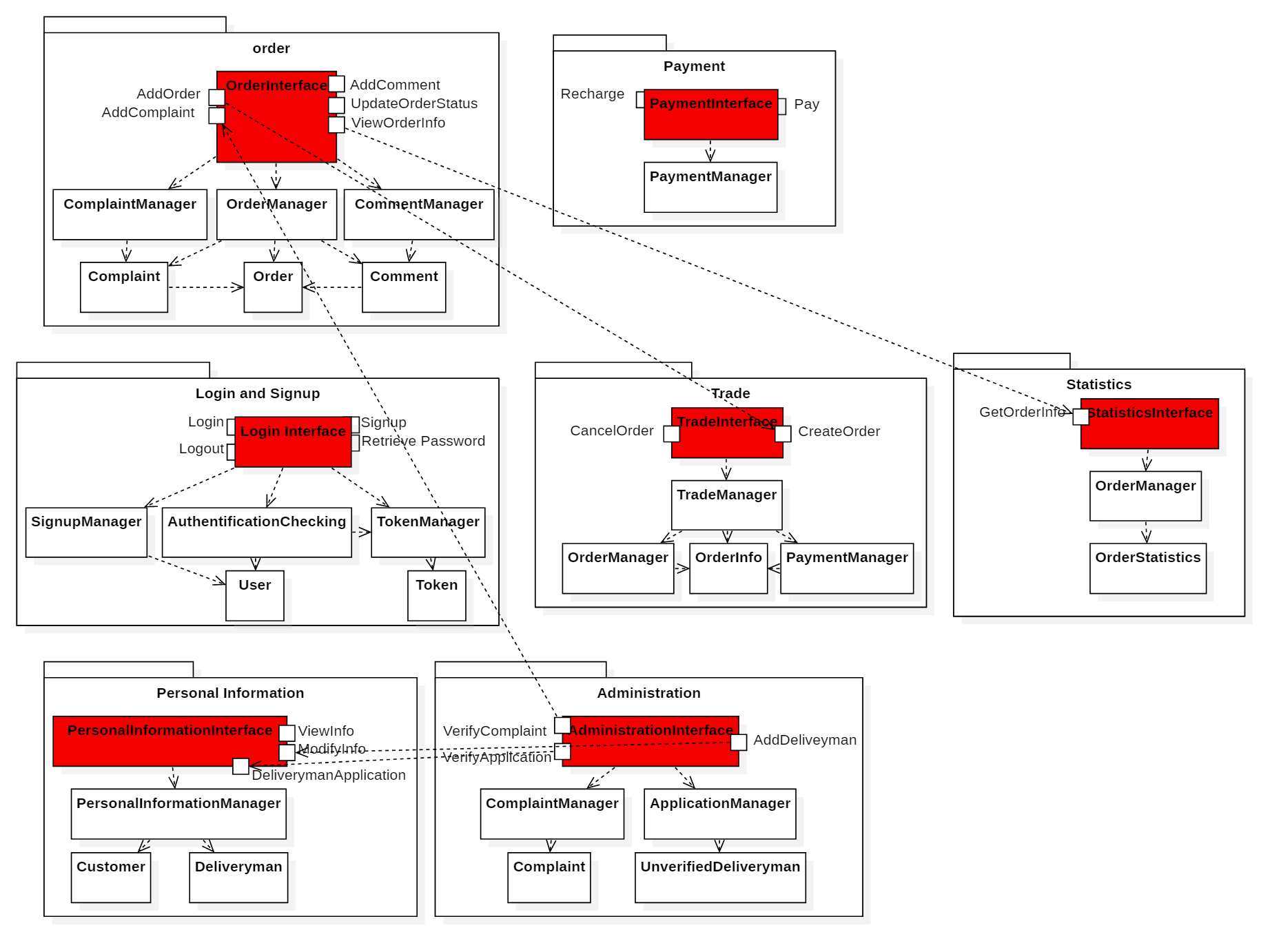
本项目任务的提出者为软件工程课程设计需要，开发者为本小组的四名同学，项目预期用户是有校园内有递送物品需求的同学和愿意通过跑腿赚取金钱的同学。

本文档的依据是《飞递-软件需求规约》、《飞递-软件需求分析规约》及《飞递-软件概要设计规约》。

## 1.3 参考资料

1. 《软件工程：实践者的研究方法》
2. 《设计模式：可复用面向对象软件的基础》

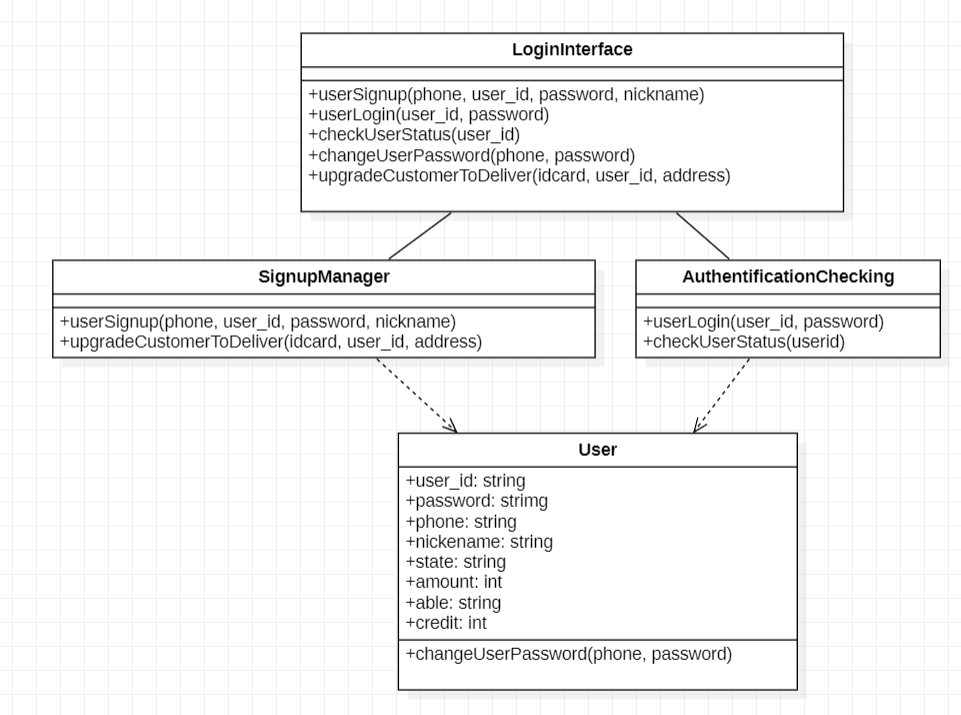
# 系统软件结构



# 类设计

## 3.1 登录注册微服务

在登录注册微服务中，一共有一个实体类User用来管理用户实体记录。同时通过两个控制类来完成用户登录、注册。两个控制类连接一个边界接口类来完成用户交互。如下图所示：



下文详细介绍这几个类。

### 3.1.1 User类

#### 功能

登录注册模块中的User类主要负责记录用户的登录账号和密码。并且通过内部的方法实现检查用户登录账号和密码是否匹配，并且通过修改密码方法管理自身密码。

#### 性能

无约束

#### 输入项

该类通过用户的账号以及密码进行初始化。修改密码时，需要提供手机号进行验证。

#### 输出项

无输出项

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| user\_id | String | 用户账号，不允许超过20个字符 |
| password | String | 用户密码，不允许超过20个字符 |
| phone | String | 手机号码，不允许超过20个字符 |
| nickname | String | 昵称，不允许超过20个字符 |
| state | String | 状态，用来区分用户和管理员 |
| amount | Int | 账户余额 |
| able | String | 账户状态，是否启用 |

#### 方法详细描述

##### checkPassword

方法签名：public Boolean checkPassword(String pwd)

功能说明：通过参数pwd与该用户的实际密码进行对比，查看密码是否正确。

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Boolean | |
| 返回值 | 密码是否正确，true为正确，false为错误。 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| pwd | String | 输入的待比较的密码。 |

##### changePassword

方法签名：public Boolean changePassword(String newPwd, String phonenumber)

功能说明:在验证手机号码正确后修改用户的密码为用户指定的新密码。

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Boolean | |
| 返回值 | 修改密码是否成功，true为成功，false为失败。 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| newPwd | String | 输入的新密码 |
| phoneNumber | String | 输入手机号码，需要手机号码正确 |

### 3.1.2 SignupManager类

#### 功能

管理用户的注册行为，提供用户通过手机号码、用户昵称、账号和密码注册的功能。

#### 性能

无约束

#### 输入项

输入手机号码、用户昵称、账号和密码进行注册。

#### 输出项

用户注册是否成功，布尔值

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| users | List | 系统中当前注册的所有用户，注册成功添加用户。 |

#### 方法详细描述

##### signup

方法签名：public Boolean signup(String phone, String pwd, String nickname,String user\_id)

功能说明：通过参数用户手机号码、用户昵称、账号和密码来进行注册。

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Boolean | |
| 返回值 | 用户是否登录成功，true为正确，false为错误。 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| phone | String | 用户的手机号码。 |
| pwd | String | 用户的待设定密码。 |
| nickname | String | 用户的昵称。 |
| user\_id | String | 用户账号。 |

##### upgradeToDeliver

方法签名：public Boolean upgradeToHost(String phone, String residentId, String user\_id)

功能说明：验证和记录用户身份证号并将用户升级成骑手。

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Boolean | |
| 返回值 | 用户是否成功升级成骑手。 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| phone | String | 用户手机号。 |
| residentId | String | 用户的身份证号。 |
| user\_id | String | 用户的账号。 |

### 3.1.3 AuthentificationChecking类

#### 功能

用于用户登录时检查用户账号和密码是否正确的类型。

#### 性能

无约束

#### 输入项

接受用户的账号和密码，通过账号和密码来验证、登录用户；

#### 输出项

用户登录是否成功，布尔值

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| users | List | 系统中当前注册的所有用户，注册成功添加用户。 |

#### 方法详细描述

##### userLogin

方法签名：public void userLoginInfo(String userid, String password)

功能说明：通过账号获取用户，然后比对用户的信息，判断密码是否正确，若正确则登录成功。

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Boolean | |
| 返回值 | 用户是否成功登录，true为成功，flase为失败。 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| phone | String | 用户的手机号码。 |
| pwd | String | 用户输入用于验证的密码。 |
| 返回类型 | Boolean | 返回类型 |
| 返回值 | 用户是否成功登录，true为成功，flase为失败。 | 返回值 |

##### checkUserStatus

方法签名：public Boolean checkUserStatus(String user\_id)

功能说明：判断用户类型是否为管理员

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Boolean | |
| 返回值 | 是否为管理员，true是，false否。 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| user\_id | String | 用户账号。 |

### 3.1.4 LoginInterface类

#### 功能

完成模块之间的接口调用交互。

#### 接口方法详细描述

##### userSignup

方法签名：public Boolean userSignup(String phone, String nickname, String password，String user\_id )

功能说明：传入用户注册的信息进行注册

实际调用方法：SignupManager.signup

##### userLogin

方法签名：public Boolean userLogin(String user\_id, String pwd)

功能说明：用户登录

实际调用方法：AuthenticationCkecking.checkLoginInfo

##### checkUserStatus

方法签名：public Boolean checkUserStatus(String user\_id)

功能说明：判断用户类型是否为管理员

实际调用方法：TokenManager.checkUserState

##### changeUserPassword

方法签名：public Boolean changeUserPassword(String phone, String newPwd)

功能说明：更改登录密码

实际调用方法：SignupManager.changePassword

##### upgradeCustomerToDeliver

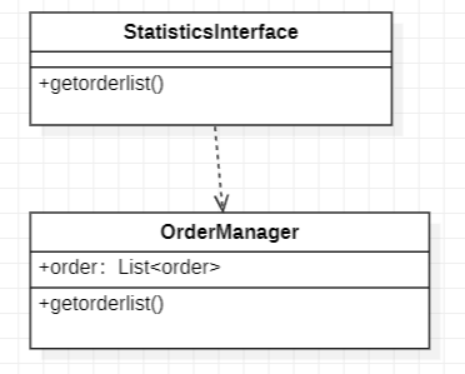
方法签名：public Boolean upgradeCustomerToDeliver(String user\_id,String idcard, String Address)

功能说明：用户申请成为骑手

实际调用方法：SignupManager.upgradeToDeliver

## 3.2 数据统计微服务

数据统计微服务中，包含一个控制类OrderManager类来实现对于订单数据的统计，然后通过StatisticsInterface实现与其他微服务之间的交互。



### 3.2.1 OrderManager类

#### 功能

管理订单信息，顾客点击查看订单获取与自己相关的订单统计信息。

#### 性能

无约束

#### 输入项

点击查看订单

#### 输出项

输出统计结果list。

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| Orders | List<Order> | 订单列表。 |

#### 方法详细介绍

##### getOrderlist

方法签名：public ActionResult getOrderlist()

功能说明：通过点击查看订单，获取用户id找到对应的订单的统计信息，并返回统计结果。

详细说明：

|  |  |
| --- | --- |
| 返回类型 | ActionResult |
| 返回值 | 统计结果 |
| 方法限定符 | public |

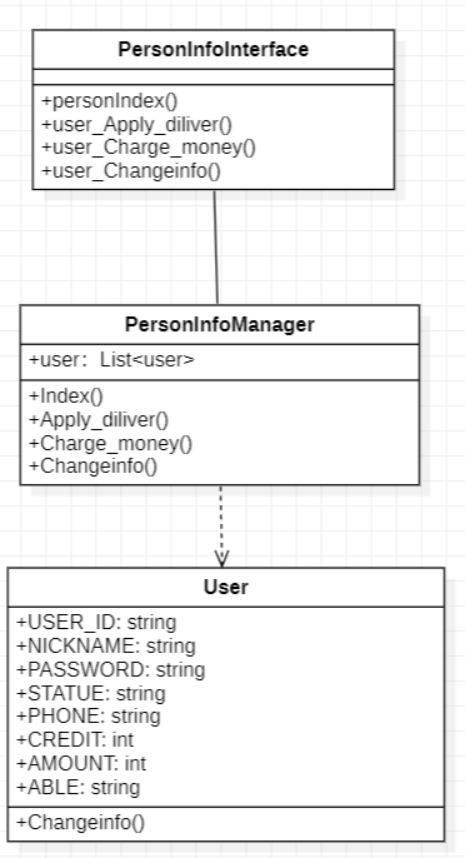
### 3.2.2 StatisticsInterface类

#### 功能

完成不同系统之间的接口调用交互。

## 3.3 用户信息微服务

该微服务包括user实体类,用于管理用户信息。在该微服务中包含一个控制类对实体做操作和管理，同时使用边界类PersonInfoInterFace来提供其他子系统调用的接口和与其他子系统接口做交互的接口，如下图所示：



接下来将会详细介绍这几个类的描述。

### 3.3.1 User 类

#### 功能

用户信息模块中的User类主要负责用户管理和修改个人信息，通过类内的方法来实现对用户信息的修改和管理。

#### 性能

无约束

#### 输入项

用户进入自己的信息页面，输入要修改的信息

#### 输出项

顾客完成信息的修改

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| USER\_ID | String | 唯一标识顾客实体的标识符，在顾客实体实例化后不会改变。 |
| NICKNAME | String | 用户的昵称，不允许超过20个字符。 |
| STATUE | String | 用户的身份，用来区分顾客，骑手，管理员。 |
| PHONE | String | 用户的电话。 |
| CREDIT | Int | 用户的信用。 |
| AMOUNT | Int | 用户的余额，用来完成订单等需求。 |
| ABLE | String | 顾客的限制，账户可用或不可用 |

#### 方法详细描述

##### Changeinfo

方法签名：public ActionResult changeinfo()

功能说明：更新顾客的基本信息

详细说明：

|  |  |
| --- | --- |
| 返回类型 | ActionResult |
| 返回值 | 无 |
| 方法限定符 | public |

#### 限制条件及出错处理

· 部分属性不能作出修改，用户Id，用户余额，用户身份等。

### 3.3.2 PersonInfoManager 类

#### 功能

个人模块中的PersonInfoManager类主要负责管理与顾客信息有关的内部业务逻辑，包括修改个人信息，获取基本的个人信息等。

#### 性能

无约束

#### 输入项

· 在获得用户的基本信息时，输入项用户点击个人信息页面，从token信息中获取的顾客id。

#### 输出项

· 在获得用户的基本信息时，输出项为用户的基本信息。

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| Users | List<users> | 用户实体类的列表 |

#### 方法详细描述

##### Index( )

方法签名：public ActionResult Index()

功能说明：通过从token信息中获取的userId获取对应的用户个人基本信息

接口调用：

· 调用登录注册微服务中的getUserFromToken(in token)函数获取token信息

· 调用自身系统Customer实体类中的getCustomer()接口

详细说明：

|  |  |
| --- | --- |
| 返回类型 | ActionResult Index() |
| 返回值 | 用户基本信息的条目集合 |
| 方法限定符 | public |

##### Apply\_Deliver()

方法签名：public ActionResult Apply\_Deliver()

功能说明：用户申请为骑手。

接口调用：

· 调用登录注册微服务中的getUserFromToken(in token)函数获取token信息

详细说明：

|  |  |
| --- | --- |
| 返回类型 | ActionResult |
| 返回值 | 申请成功 |
| 方法限定符 | public |
| 参数列表（参数名 / 类型 / 说明） | |

##### Charge\_money()

方法签名：public ActionResult Charge\_money()

功能说明：用户充值余额

接口调用：

· 调用登录注册微服务中的getUserFromToken(in token)函数获取token信息

详细说明：

|  |  |
| --- | --- |
| 返回类型 | ActionResult |
| 返回值 | 充值成功 |
| 方法限定符 | public |

### 3.3.3 PersonInfoInterface 类

#### 功能

个人模块中的边界类，负责提供其他微服务和外部系统（如web端）调用的接口，是一个只有方法没有属性的接口类，其完成的接口功能包括获取个人基本信息、更新个人基本信息、充值，申请骑手等功能。

#### 性能

每个接口的处理响应时间需在1000ms以内。

#### 输入项

· 用户点击相关操作，系统获取用户id，完成相关操作

#### 输出项

· 输出用户个人信息

#### 接口描述

##### Index( )

方法签名： public ActionResult Index()

功能说明：获取当前登录的用户的用户基本信息

接口调用：

· 需要调用PersonInfoManager类的Index()方法

##### Changeinfo( )

方法签名：public ActionResult changeinfo()

功能说明：更新用户的基本信息，返回是否更新成功

接口调用：

· 需要调用系统内的PersonInfoManager类的changeinfo()方法

##### Apply\_Deliver()

方法签名：public ActionResult Apply\_Deliver()

功能说明：用户申请为骑手

接口调用：

· 需要调用系统内的Apply\_Deliver()方法

##### Charge\_money()

方法签名：public ActionResult Charge\_money()

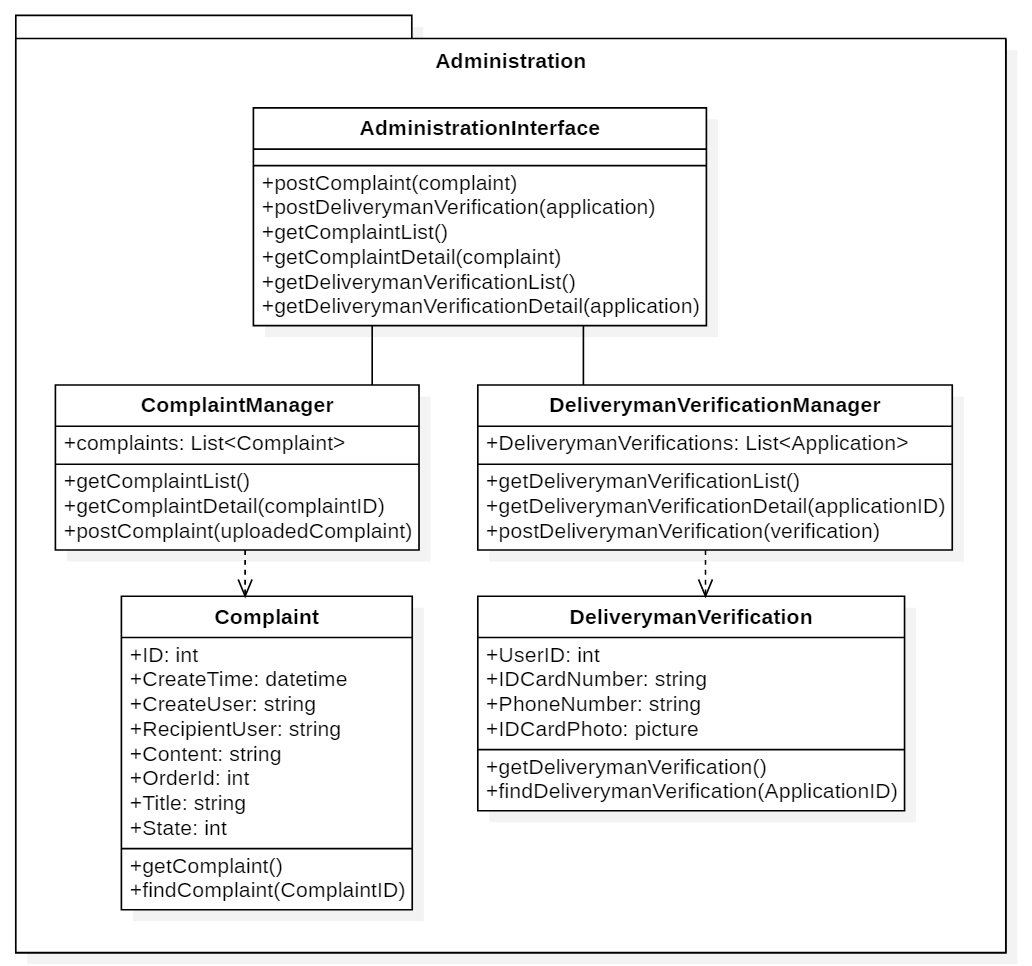
功能说明：用户充值余额

接口调用：

需要调用系统内的Charge\_money()方法

## 3.4 管理微服务

该微服务一共包括两个实体类：Complaint和UnverifiedDeliveryman，用于管理用户的投诉、骑手资质的审核。同时，在该微服务中包含两个控制类，用于完成对用户投诉的审核、处理和回复、待审核骑手资质的处理，同时使用边界类AdministrationInterface来提供其他子系统调用的接口和与其他子系统接口做交互的接口，如下图所示：



下文详细介绍这几个类。

### 3.4.1 Complaint类

#### 功能

管理模块中的Complaint类主要负责管理用户的投诉，完成对投诉的审核和反应。

#### 性能

保证每个方法的执行时间在100ms以内。

#### 输入项

查询用户投诉信息所需的具体的投诉ID。

#### 输出项

输出查询到的投诉信息或投诉信息列表。

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| complaintUserId | Int | 投诉信息投诉方的用户id |
| beComplaintedUserId | Int | 投诉信息被投诉方的用户id |
| complaintReason | String | 投诉信息中的投诉原因 |
| complaintTime | Date | 投诉时间 |
| isDealt | Boolean | 该投诉信息当前是否已经处理 |
| dealtTime | Date | 该投诉信息处理的时间 |
| reply | String | 投诉信息的回复内容 |

#### 方法详细描述

##### getComplaint

方法签名：public static List<Complaint> getComplaint()

功能说明：获取当前所有的投诉信息列表

详细说明：

|  |  |
| --- | --- |
| 返回类型 | List<Complaint> |
| 返回值 | 当前所有待处理的投诉信息列表 |
| 方法限定符 | public static |

##### findComplaint

方法签名：public Complaint findComplaint(Int ComplaintID)

功能说明:根据投诉ID查询某一个投诉信息

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Complaint | |
| 返回值 | 投诉信息 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| ComplaintId | Int | 投诉ID |

### 3.4.2 DeliverymanVerification类

#### 功能

管理模块中的DeliverymanVerification类主要负责管理用户对成为骑手的申请，审核用户的骑手资质。

#### 性能

保证每个方法的执行时间在100ms以内。

#### 输入项

查询用户提交的申请，需要输入申请用户的ID。

#### 输出项

输出查询到的申请信息或申请信息列表。

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| deliverymanVerificationId | Int | 申请信息的id |
| userId | Int | 申请用户的id |
| adminId | Int | 处理该信息的管理员id |
| vertificationResult | Bool | 该申请是否审核通过 |
| vertificationReply | String | 审核回复 |
| replyTime | Date | 审核回复时间 |

#### 方法详细描述

##### getDeliverymanVerification

方法签名：public static getDeliverymanVerification()

功能说明:获取当前所有待审核的骑手申请信息

详细说明：

|  |  |
| --- | --- |
| 返回类型 | DeliverymanVerification |
| 返回值 | 待处理的申请信息 |
| 方法限定符 | public static |

##### findDeliverymanVerification

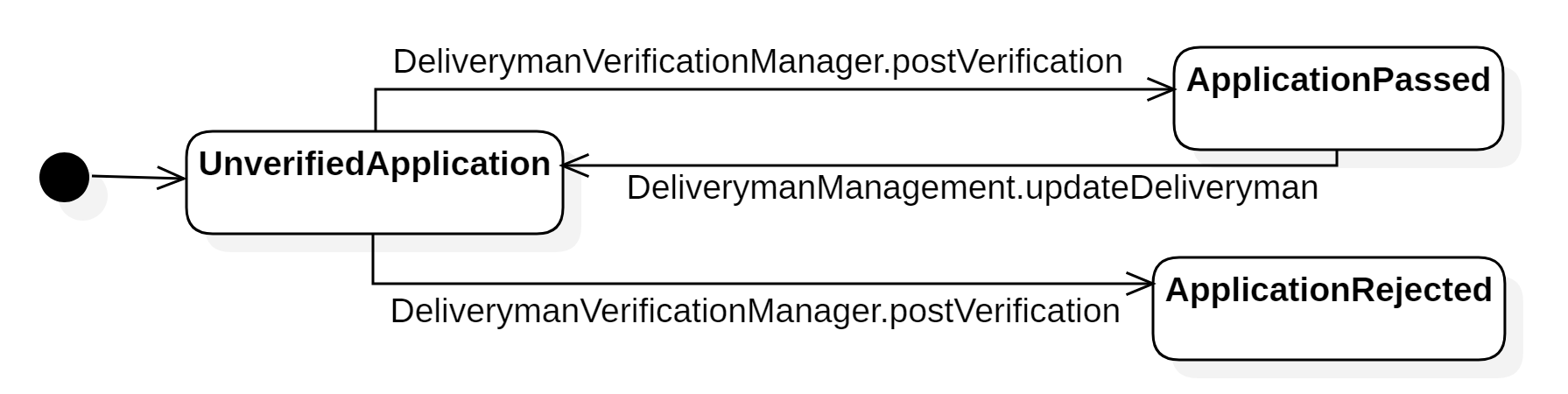
方法签名：public DeliverymanVerification findDeliverymanVerification(Int VerificationID)

功能说明:根据申请ID查询某一个骑手申请信息

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | DeliverymanVerification | |
| 返回值 | 申请信息 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| VerificationId | Int | 申请ID |

#### 类内类外方法调用状态图



### 3.4.3 ComplaintManager类

#### 功能

管理模块中的ComplaintManager类主要负责管理和维护用户的投诉。

#### 性能

保证每个方法的执行时间在100ms以内。

#### 输入项

查询用户投诉信息所需的具体的投诉ID，或者用户填写的投诉信息。

#### 输出项

输出查询到的投诉信息实体。

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| complaints | List<Complaints> | 维护的投诉信息的列表 |

#### 方法详细描述

##### getPostComplaintDetail

方法签名：public Complaint getPostComplaintDetail(Int ComplaintedId, Int beComplaintedId)

功能说明：获取某一投诉信息的详细内容

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Complaint | |
| 返回值 | 投诉信息 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| ComplaintedId | Int | 投诉方的用户id |
| beComplaintedId | Int | 被投诉方的用户id |

##### postPostComplaint

方法签名：public void postPostComplaint(uploadedComplaint)

功能说明: 上传处理好的投诉信息

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | void | |
| 返回值 | 无 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| uploadedComplaint | Complaint | 要更新的投诉信息 |

### 3.4.4 AdministrationInterface类

#### 功能

管理模块中的AministrationInterface负责提供给外部系统（如web端）和其他微服务调用的接口，功能包括获取待审核的投诉信息和骑手申请信息，修改相应的信息并上传。

#### 性能

保证每个方法的执行时间在1000ms以内。

#### 输入项

处理相关的投诉信息和申请信息时，需要输入投诉id或申请id。

#### 输出项

输出查询到的投诉信息和申请信息。

#### 方法详细描述

##### postVerification

方法签名：public Boolean postVerification(UnverifiedApplication application)

功能说明：更新管理员完成审核的投诉内容

接口调用：

需要调用系统内的VerificationManager的postVerification方法

##### postPostComplaint

方法签名：public Boolean postPostComplaint(Complaint Complaint)

功能说明：更新管理员完成的投诉处理信息

接口调用：

·       需要调用系统内的ComplaintManager的postPostComplaint方法

##### getApplicationList

方法签名：public List<UnverifiedApplication> getApplicationList()

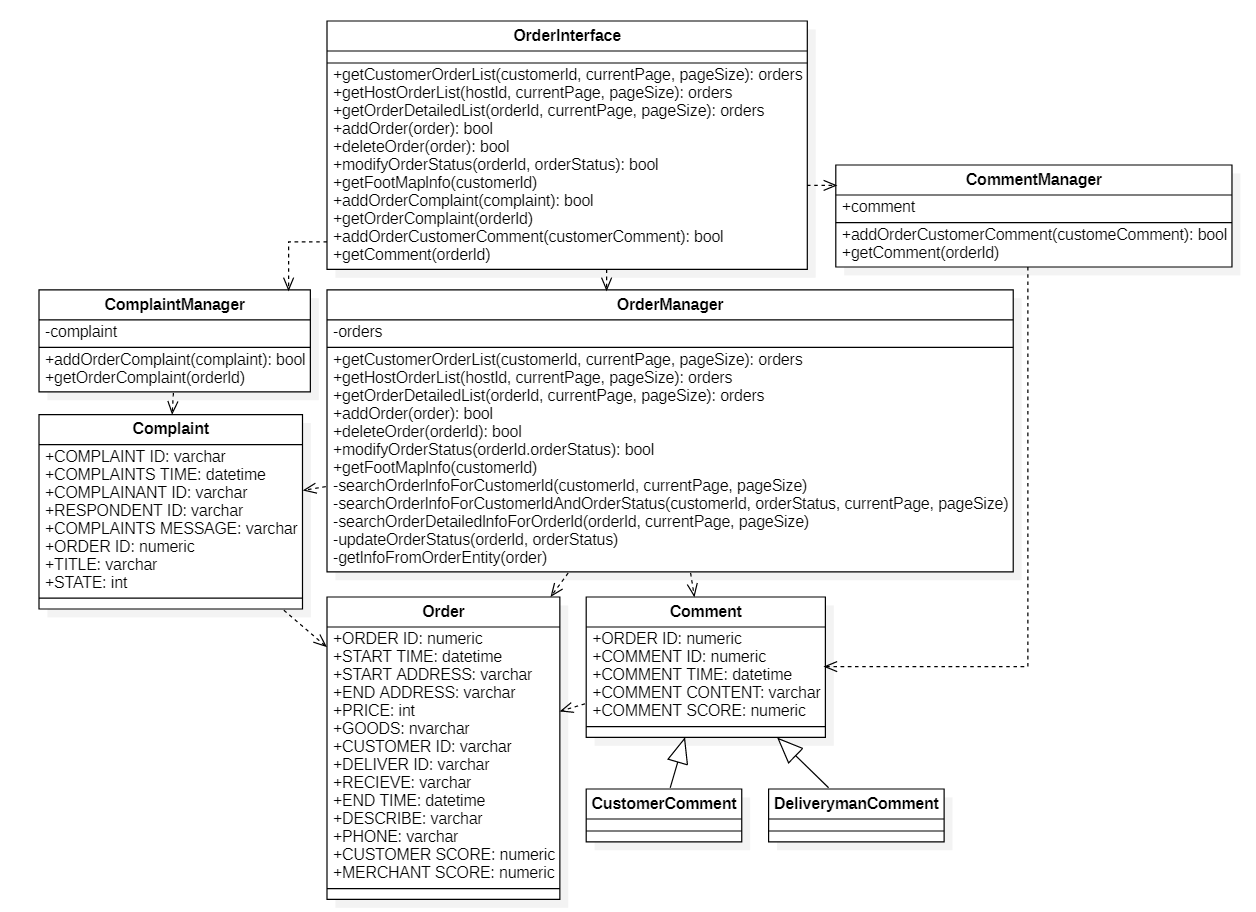
功能说明：获取管理员当前待处理的所有待审核骑手申请信息列表

接口调用：

·       需要调用系统内的VerificationManager的getApplicationList()方法

## 3.5 订单微服务

订单微服务主要负责订单、订单评论以及订单投诉信息的维护以及相关业务逻辑的实现，为上层微服务提供底层的订单信息的管理。在该微服务中主要存在3个实体类，分别为负责订单主体信息的Order类，负责订单评论的Comment类，其中根据评论的主体不同，将Comment泛化为CustomerComment以及DeliverymanComment类。除此之外，负责订单投诉信息为Complaint类。在此之上，通过三个控制类Manager分别管理对应相关实体类的操作与业务实现，在最上层提供一个总体对外的接口供其他微服务调用，设计类图如下：



接下来详细介绍这几个类。

### 3.5.1 Order类

#### 功能

订单微服务中的Order类并不提供复杂的业务逻辑处理，但是为上层控制类处理业务提供订单数据支持。

#### 性能

无约束

#### 输入项

1. 输入详细的订单下单时间、下单用户ID、订单总价格、订单状态，完成对订单的初始化操作。

2. 输入订单ID与订单状态进行订单状态的修改。

3. 输入订单ID实现相应订单的查找。

#### 输出项

1. 接受详细订单信息的输入，输出初始化订单操作的完成标志

2. 接受订单ID与订单状态值的输入，输出修改订单状态完成标志

3. 接受订单ID输入，返回该ID的订单信息

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| ORDER ID | numeric | 订单的ID |
| START TIME | datetime | 订单的开始时间 |
| START ADDRESS | varchar | 订单的开始地址 |
| END ADDRESS | varchar | 订单的结束地址 |
| PRICE | int | 订单的价格 |
| GOODS | nvarchar | 商品名称 |
| CUSTOMER ID | varchar | 顾客ID |
| DELIVER ID | varchar | 骑手ID |
| RECIEVE | varchar | 订单状态 |
| END TIME | datetime | 订单的结束时间 |
| DESCRIBE | varchar | 订单的描述 |
| PHONE | varchar | 顾客的手机号码 |
| CUSTOMER SCORE | numeric | 顾客得分 |
| MERCHANT SCORE | numeric | 骑手得分 |

#### 方法详细描述

##### Order

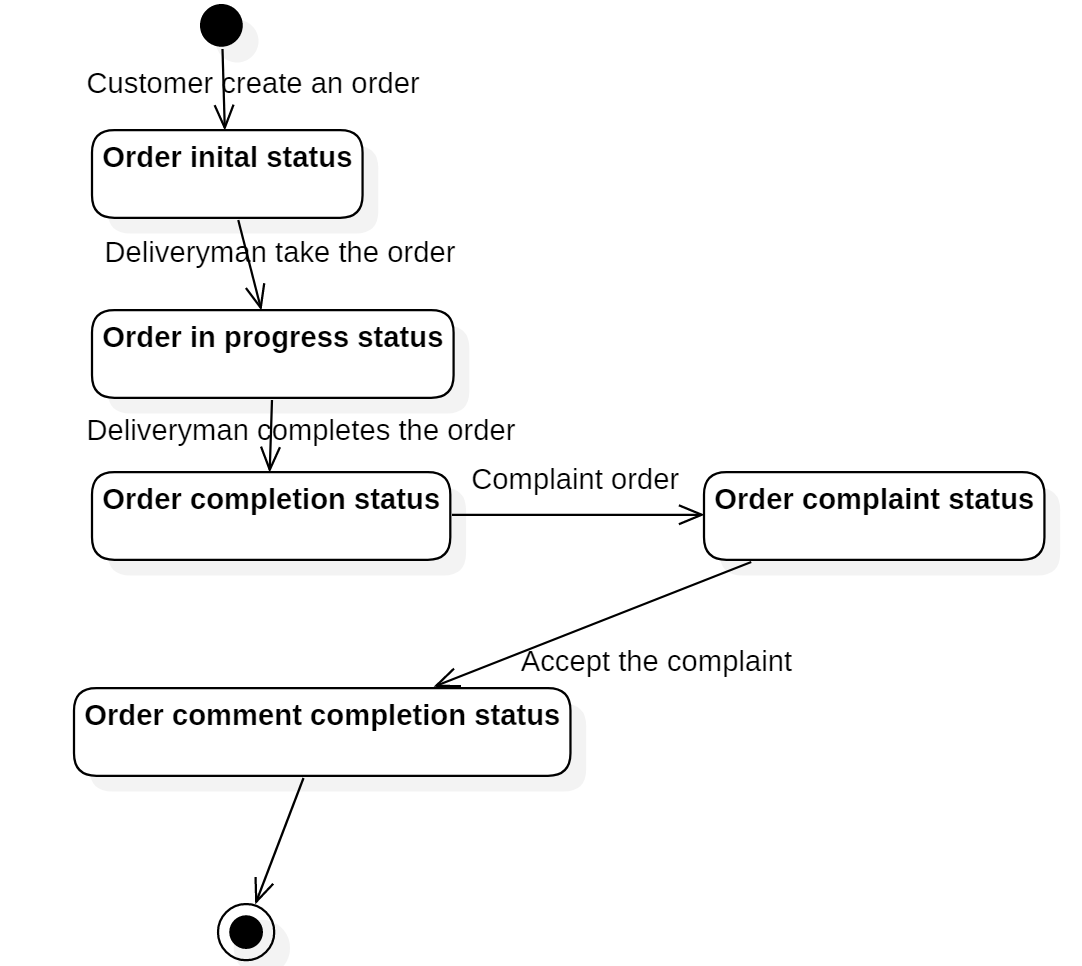
方法签名：public Order(long customerId, double totalCost)

功能说明：Order类的构造函数，初始化一个默认订单。

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | 无返回值 | |
| 返回值 | 无返回值 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| customerId | long | 待创建订单的用户标识符 |
| totalCost | double | 待创建订单的总花费金额 |

#### 类内、类外方法调用状态图



#### 限制条件及出错处理

对于不存在的订单查询任务要返回NullException空值错误提示。

### 3.5.2 Comment类

#### 功能

订单微服务中的Comment类不提供复杂的业务逻辑处理，为上层的控制类提供基础的评论数据支持。

#### 性能

无约束。

#### 输入项

1. 输入详细的订单的标识符、评论时间、评论内容、评论分数完成对订单评论的初始化操作。

2. 输入评论标识符查找相应的评论

#### 输出项

1. 接受详细订单评论信息的输入，输出初始化订单评论操作的完成标志

2. 接受订单评论标识符输入，返回该标识符对应的订单评论信息

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| COMMMENT ID | varchar | 订单评论的标识符 |
| ORDER ID | varchar | 订单的标识符 |
| COMMENT TIME | datetime | 评论时间 |
| COMMENT CONTENT | varchar | 订单评论的内容 |
| COMMENT SCORE | numeric | 订单分数 |

#### 方法详细描述

##### Comment

方法签名：public Comment(long orderId, String orderContent, int orderScore)

功能说明：Comment类的构造函数，初始化一个默认的订单评论。

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | 无返回值 | |
| 返回值 | 无返回值 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| orderId | long | 订单评论的指向订单标识符 |
| orderContent | String | 订单评论信息的文本内容 |
| orderScore | int | 订单评论信息的分数 |

#### 类内、类外方法调用状态图

该类无状态变化过程。

#### 限制条件及出错处理

对于不存在的订单评论查询任务要返回NullException空值错误提示

### 3.5.3 Complaint类

#### 功能

订单微服务中的Complaint类不提供复杂的业务逻辑处理，为上层的控制类提供基础的订单投诉类的数据支持。

#### 性能

无约束。

#### 输入项

1. 输入详细的订单的标识符、投诉时间、投诉原因完成对订单投诉信息的初始化操作。

2. 输入投诉标识符查找相应的订单投诉

3. 输入投诉标识符、处理投诉的管理员的标识符、处理时间、处理结果与回复完成对订单投诉内容的更新。

#### 输出项

1. 接受详细订单投诉信息的输入，输出初始化订单投诉信息操作的完成标志

2. 接受订单投诉标识符输入，返回该标识符对应的订单投诉信息

3. 接受详细的订单投诉处理信息的输入，输出订单投诉处理操作后的完成标志

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| COMPLAINT ID | long | 订单投诉信息的标识符 |
| COMPLAINT TIME | long | 投诉时间 |
| COMPLAINANT ID | long | 投诉人ID |
| RESPONDENT ID | long | 被投诉人ID |
| COMPLAINTS MESSAGE | varchar | 投诉信息 |
| ORDER ID | varchar | 订单ID |
| TITLE | varchar | 投诉标题 |
| STATE | int | 投诉状态 |

#### 方法详细描述

##### Complaint

方法签名：public Complaint(long orderId, String ComplaintReason)

功能说明：Complaint类的构造函数，初始化一个默认的订单投诉信息。

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | 无返回值 | |
| 返回值 | 无返回值 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| orderId | long | 订单投诉信息的指向订单标识符 |
| ComplaintReason | String | 订单投诉信息的文本内容 |

##### dealComplaint

方法签名：public void dealComplaint(long ComplaintId, int adminId, String reply)

功能说明：处理存在的订单投诉信息，添加处理的订单投诉处理结果

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | 无返回值 | |
| 返回值 | 无返回值 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| ComplaintId | long | 订单投诉信息的标识符 |
| adminId | int | 订单投诉信息的处理人员标识符信息 |
| reply | String | 订单投诉信息的回复内容 |

#### 类内、类外方法调用状态图

该类无状态变化过程。

#### 限制条件及出错处理

对于不存在的订单投诉信息查询任务要返回NullException空值错误提示。

对于已经处理好的投诉信息，如果再次调用处理投诉函数会提示重复订单投诉的错误提示。

### 3.5.4 OrderManager类

#### 功能

订单微服务中的OrderManager类是对订单信息进行业务逻辑处理的主要类，主要负责对根据不同的输入数据字典结构对订单数据进行业务操作，包含对订单的筛选、修改、增添与删除等基本操作，为了提高订单筛选过程的通用性，提取部分通用函数实现复杂业务逻辑。

#### 性能

无约束。

#### 输入项

1. 输入用户标识符、筛选过程中的当前页码、每页展示的记录数量实现订单筛选。

2. 输入完整订单信息实现订单插入操作

3. 输入订单标识符信息实现订单删除操作

4. 输入订单标识符、订单状态实现修改订单状态

#### 输出项

1. 接受用户标识符、筛选过程中的当前页码、每页展示的记录数量的输入，输出符合筛选条件的订单信息列表

2. 接受输入的订单详细信息，返回插入订单操作的成功标志信息

3. 接受输入的订单标识符与订单状态信息，返回修改订单状态操作的成功标志信息

4. 接受输入的订单标识符信息，返回删除订单操作的成功标志信息

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| orders | List<Order> | OrderManager类维护的订单信息列表。 |

#### 方法详细描述

##### getCustomerOrderList

方法签名：public List<Orders> getCustomerOrderList(long customerId, int currentPage, int pageSize)

功能说明：根据输入的用户标识符分页筛选该用户的订单信息

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | List<Order> | |
| 返回值 | 筛选出的该用户的订单信息列表 | |
| 方法限定符 | Public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| customerId | long | 用户标识符 |
| currentPage | int | 筛选订单列表的当前页码 |
| pageSize | int | 筛选订单列表的每一页的记录数量 |

实现过程：

该函数通过提供的customerId进行条件筛选存储的订单信息，并且根据分页信息对查询时的信息进行分页筛选。

##### getDeliverymanOrderList

方法签名：public List<Orders> getDeliverymanOrderList(long DeliverymanId, int currentPage, int pageSize)

功能说明：根据输入的骑手标识符分页筛选该骑手相关的订单全部订单信息

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | List<Order> | |
| 返回值 | 筛选出的该骑手订单相关的订单信息列表 | |
| 方法限定符 | Public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| DeliverymanId | long | 骑手标识符，非空且参照于Deliveryman表 |
| currentPage | int | 筛选订单列表的当前页码，取值不小于0 |
| pageSize | int | 筛选订单列表的每一页的记录数量 |

实现过程：

该函数首先是根据筛选该骑手的全部订单信息，然后根据订单标识符直接筛选该骑手的全部订单信息，并且根据分页条件进行部分数据返回。

##### addOrder

方法签名：public boolean addOrder(Order order)

功能说明：根据提供的Order信息新建一个订单

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | boolean | |
| 返回值 | 新建订单操作的成功标志 | |
| 方法限定符 | Public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| order | Order | 包含必要信息的订单实体，来源于表单输入，参数非空 |

实现过程：

该函数接受一个包含必要信息的订单实体，但是需要在创建新订单前补足订单信息，如订单标识符，下单时间等信息。并且在判断订单编号唯一后将该订单插入到数据库中。

##### deleteOrder

方法签名：public boolean deleteOrder(long orderId)

功能说明：根据输入的订单标识符进行筛选订单并删除相应的订单信息

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | boolean | |
| 返回值 | 删除订单操作的成功标志 | |
| 方法限定符 | Public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| orderId | long | 待删除订单的标识符信息，非空 |

实现过程：

该函数接受orderId后筛选该标识符指向的订单实体，在找到后将该订单从数据库中删除，反之，提示未存在该订单的报错信息。

##### modifyOrderStatus

方法签名：public boolean modifyOrderStatus(long orderId, int orderStatus)

功能说明：修改指定订单的订单状态

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | boolean | |
| 返回值 | 修改订单状态操作的执行标志 | |
| 方法限定符 | Public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| orderId | long | 待修改订单的标识符信息，非空 |
| orderStatus | int | 待修改订单的状态，状态取值为0-2，非空 |

实现过程：函数首先接受待修改的订单标识符信息，经过筛选如果存在该订单内容，则根据输入的订单状态值对订单进行状态修改，反之，则提示未存在该订单的报错信息。

##### searchOrderInfoForCustomerIdAndOrderStatus

方法签名：private List<Order> searchOrderInfoForCustomerIdAndOrderStatus(long customerId, long orderStatus, long currentPage, long pageSize)

功能说明：通过用户的标识符与订单的状态分页的筛选返回该用户的订单信息

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | List<Order> | |
| 返回值 | 筛选后该用户的订单信息列表 | |
| 方法限定符 | Private | |
| 参数列表（参数名 / 类型 / 说明） | | |
| customerId | long | 查询订单列表的用户标识符信息，非空，参照于customer顾客表 |
| orderStatus | int | 查询订单列表的筛选订单状态信息，取值范围为0~5，如果该值为空，则表示不进行订单状态条件约束 |
| currentPage | int | 当前页码的分页信息，取值不小于0 |
| pageSize | int | 每一页包含的记录数量，取值不小于0 |

实现过程：首先判断customerId指向的用户是否存在，如果存在则首先根据currentPage与pageSize创建分页信息，再根据用户标识符与订单状态共同筛选订单信息，并返回分页后的订单信息列表，反之，如果不存在该用户，则提出报错信息“不存在该用户”，如果不存在相应的订单状态，则提出报错信息“输入的订单状态异常”。

##### getInfoFromOrderEntity

方法签名：private OrderInfo getInfoFromOrderEntity(Order order)

功能说明：作为私有函数，该函数作用为接受一个订单实体，从该订单中抽取相应的信息供其他函数的使用

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | OrderInfo(data dictionary) | |
| 返回值 | 抽取后的经过扩展的部分订单列表信息 | |
| 方法限定符 | Private | |
| 参数列表（参数名 / 类型 / 说明） | | |
| order | Order | 输入的订单实体，该订单实体为经过查询后得到的订单信息 |

实现过程：该函数的主要作用为接受订单实体并抽取部分的内容经过拓展后包装返回给需要使用的函数，当接受到的订单实体为空时，返回NullException的错误信息提示。

#### 类内、类外方法调用状态图

该类无状态变化过程。

#### 限制条件及出错处理

对于输入的参数如果存在Null的对象，则返回NullException错误提示

对于输入的标识符信息进行筛选，如果并不存在筛选实体则返回“不存在用户/订单”的错误提示信息。

对于输入的参数不符合要求的数据类型与取值范围，则返回“输入参数类型不正确”的错误提示信息。

### 3.5.5 ComplaintManager类

#### 功能

订单微服务中的ComplaintManager负责对订单投诉信息进行管理，如搜索某一个订单的投诉信息，添加新的投诉信息等相关功能。

#### 性能

无约束。

#### 输入项

1. 输入详细的订单的标识符、投诉时间、投诉原因经过初始化的订单投诉信息 ，实现新增订单投诉信息

2. 输入订单的标识符信息，实现对订单投诉信息的筛选与查找

#### 输出项

1. 接受输入详细的订单的标识符、投诉时间、投诉原因经过初始化的订单投诉信息，输出增加新的订单投诉的成功标志

2. 接受输入订单的标识符信息，输出经过该标识符查找对应的订单投诉信息

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| Complaint | Complaint | ComplaintManager维护的一个订单投诉信息实体 |

#### 方法详细描述

##### addOrderComplaint

方法签名：public boolean addOrderComplaint(Complaint Complaint)

功能说明：为订单插入一条订单投诉信息。

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | boolean | |
| 返回值 | 插入订单投诉信息的操作标志 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| Complaint | Complaint | 待插入的订单投诉信息，仅包含基础的投诉信息，尚未被管理员处理 |

实现过程：

该函数实现增加一个订单投诉信息，首先需要生成唯一的订单投诉信息标识码，然后对订单信息的合理性进行筛选，如果合理，则将该订单投诉信息插入到数据库中并等待后续处理，如果不合理，如缺少订单标识符，则提示错误信息为“订单投诉信息不合理”。

##### getOrderComplaint

方法签名：public Complaint getOrderComplaint(long orderId)

功能说明：根据订单的标识符获得该订单对应的订单投诉信息

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Complaint | |
| 返回值 | 该订单的投诉信息，如果没有则返回NULL空值 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| orderId | long | 待查询的订单标识符信息，与订单投诉信息是唯一对应的，参照于order表 |

实现过程：首先通过订单标识符对订单信息进行筛选，然后判断是否存在这个订单，如果存在订单则根据该订单标识符寻找是否存在订单投诉信息，如果存在则返回，不存在投诉信息则返回空值并提示信息防止误用，如果不存在订单，则返回报错信息“不存在相应订单”。

#### 类内、类外方法调用状态图

该类无状态变化过程。

#### 限制条件及出错处理

对于缺少必要属性的输入投诉信息参数，返回报错信息“输入投诉信息不合理”

对于输入的订单未有对应订单投诉信息的情况，应当返回“该订单未被投诉”的提示信息，防止误引用空值

### 3.5.6 CommentManager类

#### 功能

订单微服务中的CommentManager负责对订单评论信息进行管理，如添加用户评论，添加骑手评论等内容，或者根据订单编号返回评论信息。

#### 性能

无约束。

#### 输入项

1. 输入详细的订单的标识符、评论内容、评论得分经过初始化的订单评论信息 ，实现新增订单评论信息

2. 输入订单的标识符信息，实现对订单评论信息的筛选与查找

#### 输出项

1. 接受输入详细的订单的标识符、评论内容、评论得分经过初始化的订单评论信息，输出增加新的订单评论的成功标志

2. 接受输入订单的标识符信息，输出经过该标识符查找对应的订单评论信息

#### 属性描述

|  |  |  |
| --- | --- | --- |
| 属性名 | 类型 | 说明 |
| comment | Comment | commentManager维护的一个订单评论信息实体 |

#### 方法详细描述

##### addOrderCustomerComment

方法签名：public boolean addOrderCustomerComment(Comment comment)

功能说明：为订单插入一条顾客订单评论信息。

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | boolean | |
| 返回值 | 插入一条顾客订单评论信息操作标志 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| comment | Comment | 待插入的订单顾客评论信息 |

实现过程：

该函数实现增加一个订单顾客评论信息，首先需要生成唯一的订单评论信息标识符与评论时间，然后对订单信息的合理性进行筛选，如果合理，则将该订单评论信息插入到数据库中并等待后续处理，如果不合理，如缺少订单标识符，则提示错误信息为“订单评论信息不合理”。

##### getComment

方法签名：public Comment getComment(long orderId)

功能说明：根据订单的标识符获得该订单对应的订单评论信息

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Comment | |
| 返回值 | 该订单的评论信息，如果没有则返回NULL空值 | |
| 方法限定符 | public | |
| 参数列表（参数名 / 类型 / 说明） | | |
| orderId | long | 待查询的订单标识符信息，与订单评论信息是唯一对应的，参照于order表 |

实现过程：首先通过订单标识符对订单信息进行筛选，然后判断是否存在这个订单，如果存在订单则根据该订单标识符寻找是否存在订单评论信息，如果存在则返回，不存在评论信息则返回空值并提示信息防止误用，如果不存在订单，则返回报错信息“不存在相应订单”。

#### 类内、类外方法调用状态图

该类无状态变化过程。

#### 限制条件及出错处理

对于缺少必要属性的输入评论信息参数，返回报错信息“输入评论信息不合理”。

对于输入的订单未有对应订单评论信息的情况，应当返回“该订单未添加评论信息”的提示信息，防止误引用空值。

### 3.5.7 OrderInterface类

#### 功能

订单微服务中的OrderInterface类是对外的接口类，是订单微服务整体功能的对外体现，其他的微服务通过该与该接口进行交互实现相应的功能与业务逻辑。

#### 性能

无约束。

#### 输入项

接受来自于外部微服务或者前端表单提交的多种多样的数据字典完成对订单、订单评论、订单投诉信息的业务逻辑处理。

#### 输出项

输出包含内部查询与处理后的结果与HttpStatus共同形成的ResponEntity。

#### 属性描述

接口类中不维护属性。

#### 方法详细描述

##### getCustomerOrderList

方法签名：public ResponEntity getCustomerOrderList(int orderStatus, int currentPage, int pageSize)

接口调用：public User getUserFromToken(in token) 获得customerId

实际调用方法:

OrderManager.getCustomerOrderList(customerId,orderStatus,currentPage,pageSize)

##### getDeliverymanOrderList

方法签名：public ResponEntity getDeliverymanOrderList(int orderStatus, int currentPage, int pageSize)

接口调用：public User getUserFromToken(in token) 获得DeliverymanId

实际调用方法:

OrderManager.getDeliverymanOrderList(DeliverymanId,orderStatus,currentPage,pageSize)

##### addOrder

方法签名：public HttpStatus addOrder(Order order)

接口调用：public User getUserFromToken(in token) 获得customerId

实际调用方法:

OrderManager.addOrder(order)

##### deleteOrder

方法签名：public HttpStatus deleteOrder(long orderId)

接口调用：无外部接口调用

实际调用方法:

OrderManager.deleteOrder(orderId)

##### modifyOrderStatus

方法签名：public HttpStatus modifyOrderStatus(long orderId, int orderStatus)

接口调用：无外部接口调用

实际调用方法:

OrderManager.modifyOrderStatus(orderId,orderStatus)

##### addOrderReport

方法签名：public HttpStatus addOrderReport(Report report)

接口调用：无外部接口调用

实际调用方法:

ReportManager.addOrderReport(report)

##### getOrderReport

方法签名：public ResponseEntity getOrderReport(long orderId)

接口调用：无外部接口调用

实际调用方法:

ReportManager.getOrderReport(orderId)

##### addOrderCustomerComment

方法签名：public HttpStatus addOrderCustomerComment(long orderId)

接口调用：无外部接口调用

实际调用方法:

CommentManager.addOrderCustomerComment(orderId)

##### getComment

方法签名：public ResponseEntity getComment(long orderId)

接口调用：无外部接口调用

实际调用方法:

ReportManager.getComment(orderId)

#### 类内、类外方法调用状态图

该类无状态变化过程。

#### 限制条件及出错处理

HttpStatus 401 表示未登录状态，应提示用户“请先登录系 统”的信息。

## 3.6 支付微服务

支付微服务主要负责在该跑腿项目的订单中调用阿里云支付接口，实现该系统中的支付相关功能，并为上层微服务提供底层的支付功能实现。在该微服务中主要存在1个实体类，负责为最后完成订单时的支付功能提供基本订单信息。通过两个控制类管理订单完成的付款功能以及订单取消的退款的回调功能实现。对于更上层的微服务有一个对外接口方便其他微服务进行调用。支付微服务的总类图如下：

### descript

### 3.6.1 OrderPayInfo类

功能：

为外部支付接口提供所需要支付的订单信息，通过上层微服务对订单信息的查询创建来实现对订单信息的维护。

性能：

无约束

输入项：

创建OrderPayInfo，通过订单类查询相应待支付订单的订单id、订单价格、订单的商品名称以及订单的配送者id等基本信息实现初始化操作。

输出项：

接受输入项的订单基本信息，输出相应的类初始化方便后续的订单支付、订单退款等需要获取订单相应信息的操作。

属性描述：

|  |  |  |
| --- | --- | --- |
| 属性名 | 数据类型 | 说明 |
| Order\_Id | Int | 订单标识符，非空 |
| Price | Int | 订单支付金额，非空 |
| OrderGoods | String | 订单配送商品，非空 |
| DeliveryName | String | 配送者，订单支付受益方 |

方法详细描述：

OrderPayInfo(int Order\_Id, int Price, string OrderGoods, string DeliveryName)

功能说明：OrderPayInfo类的构造函数，对待支付订单的基本信息初始化

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | 无 | |
| 返回值 | 无 | |
| 方法限定符 | Public | |
| 参数列表（参数名称 / 数据类型 / 说明） | | |
| Order\_Id | Int | 订单标识符，非空 |
| Price | Int | 订单支付金额，非空 |
| OrderGoods | String | 订单配送商品，非空 |
| DeliveryName | String | 配送者，订单支付受益方 |

实现过程：

该函数为OrderPayInfo的有参构造函数，通过待支付订单的基本信息完成对象构造，若订单参数缺失或订单状态不为已接单状态则返回错误信息。

类内类外方法调用状态图：

该类无状态变化过程。

### 3.6.2 OrderPaySerice类

功能：

OrderPaySerice类实现接收订单的支付基本信息实现对订单的支付以及订单取消的退款功能。

性能：

无约束。

输入项：

获取待支付订单相关的信息进而实现对订单的支付以及订单取消的退款功能。

输出项：

接收必要的待支付订单信息，输出订单完成支付成功的布尔值；

接收必要的退款订单信息，输出订单取消退款成功的布尔值。

属性描述：

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 说明 |
| payOrder | OrderPayInfo | 支付及退款订单的所需信息 |
| payClient | PayClient | Pay接口是实现对请求的提交的客户端实体 |

方法详细描述：

OrderPay(OrderPayInfo): bool

功能说明：实现对待支付订单的提交功能，并且返回是否支付成功的布尔值；

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Bool | |
| 返回值 | Pay支付接口的支付是否成功信息 | |
| 方法限定符 | Public | |
| 参数列表（参数名称 / 数据类型 / 说明） | | |
| payOrder | OrderPayInfo | 支付及退款订单的所需信息 |

实现过程：

该函数实现对订单完成后的支付信息提交。获得参数OrderPayInfo的订单支付相关信息，将该对象转换为json信息格式并封装为Request请求由PayClient进行提交获取相应是否成功信息。

orderRefund(OrderPayInfo): bool

功能说明：实现对取消订单的退款提交功能，并且返回是否退款成功的布尔值；

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Bool | |
| 返回值 | Pay支付接口退款是否成功信息 | |
| 方法限定符 | Public | |
| 参数列表（参数名称 / 数据类型 / 说明） | | |
| payOrder | OrderPayInfo | 支付及退款订单的所需信息 |

实现过程：

该函数实现对订单取消后的退款信息提交。获得参数OrderPayInfo的订单支付相关信息，将该对象转换为json信息格式并封装为Request请求由PayClient进行提交获取相应是否成功信息。

类内、类外方法调用状态图：

该类无状态变化过程。

限制条件及出错处理：

参数为空以及接口调用错误返回错误提示信息。

### 3.6.3 OrderPayUpdate类

功能：

该类负责对支付成功后更新操作以及相关业务的处理。

性能：

无约束；

输入项：

输入来自Pay接口传输的httpServeRequest信息，执行支付相关更新函数处理。

输出项：

接收http请求，返回经过验证和业务逻辑处理后的结果是否成功布尔值；

属性描述：

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 说明 |
| payClIent | PayClient | 基于Pay接口实现对请求提交的客户端实体 |

方法详细描述：

orderPayUpdate(in httpSericeRequest): string

功能说明：在支付或退款成功后，接受外部支付远端发送的http请求，在验签通过后执行相应业务逻辑处理，并返回是否更新成功的布尔值；

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Bool | |
| 返回值 | 验证处理结果并返回更新成功的布尔值 | |
| 方法限定符 | Public | |
| 参数列表（参数名称 / 数据类型 / 说明） | | |
| httpServeRequest | HttpServeRequest | 外部支付远端发送http请求 |

实现过程：

该函数为被动调用过程，在用户支付或退款成功后，外部支付远端会自动向配置完成的异步通知地址发送POST请求，在接收到该请求后首先执行验签操作，验签完成后执行自身的业务逻辑处理，最后返回更新订单信息成功的布尔值。

类内类外方法调用状态图：

该类无状态变化过程；

限制条件及出错处理：

无

### 3.6.4 IorderSerice类

功能：

该类为支付微服务向外部微服务提供的总接口，支付微服务通过整合外部支付接口提供的功能提供给其他微服务使用，并完成相应的业务逻辑；

性能：

无约束；

输入项：

接收OrderPayInfo信息与来自网络的http请求实现对服务的调用

输出项：

返回支付界面的表单信息与相应的接口调用;

属性描述：

接口类不提供属性

方法详细描述：

OrderPay(OrderPayInfo): bool

接口调用：alipayClient.excutePage(tradeRequest) 通过外部客户端执行相应的提交

实际调用方法：OrderPaySerice.OrderPay(OrderPayInfo)

orderRefund(OrderPayInfo): bool

接口调用：alipayClient.excutePage(tradeRequest) 通过外部客户端执行相应的提交

实际调用方法：OrderPaySerice. orderRefund(OrderPayInfo)

orderPayUpdate(in httpSericeRequest): string

接口调用：alipayClient.rsaCheckV1(alipayConfig) 通过外部客户端执行验签

实际调用方法：OrderPayUpdate. orderPayUpdate(in httpSericeRequest)

类内、类外方法调用状态图

该类无状态变化过程

限制条件及出错处理

无

## 3.7 交易微服务

交易微服务主要负责整合订单管理以及支付微服务提供的接口完成对订单交易的业务流程实现。在该微服务中主要存在一个实体类负责接收前端返回的订单信息，并将该信息调用支付微服务接口实现业务功能，该微服务中存在一个控制器类负责整合对外部接口功能调用实现自身业务逻辑。最后，通过对外提供一个整体的接口类实现服务的调用。交易微服务总类图如下：

### descript

### 3.7.1 OrderInfo类

功能：

该实体类用于接收来自前端传输的表单订单数据，存储后将该信息分别调用订单与支付的接口实现数据支持;

性能：

无约束；

输入项：

顾客Id、订单的开始时间、订单价格、订单状态等信息为该类进行初始化操作；

输出项：

接收初始化参数并进行构造函数返回；

属性描述：

|  |  |  |
| --- | --- | --- |
| 属性名称 | 数据类型 | 说明 |
| CustomerId | String | 顾客标识符 |
| Start\_Time | TimeStamp | 订单开始时间 |
| Price | Int | 订单金额 |
| Status | String | 订单状态 |

方法详细描述：

OrderInfo(string CutomerId, Timstamp Start\_Time, int Price, string Status)

功能说明：接收参数后对OrderInfo类进行初始化构造;

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | 无 | |
| 返回值 | 无 | |
| 方法限定符 | Public | |
| 参数列表（参数名称 / 数据类型 / 说明） | | |
| CustomerId | String | 顾客标识符 |
| Start\_Time | TimeStamp | 订单开始时间 |
| Price | Int | 订单金额 |
| Status | String | 订单状态 |

实现过程：

该函数为基础的OrderInfo的构造函数，通过接收参数实现对OrderInfo类对象的生成，如果参数错误，则返回用户“输入参数有误”的错误信息。

类内、类外方法调用状态图

该类无状态变化过程；

限制条件及出错处理：

参数输入错误，则返回用户“输入参数有误”信息;

### 3.7.2 OrderSerice类

功能：

控制整个涉及交易的过程的执行，实现对外部接口的调用，实现完整的订单的复杂业务流程。

性能：

无约束；

输入项：

接收来自前端的表单信息实现对订单交易业务逻辑的处理；

输出项：

接收订单表单信息实现对订单交易的业务逻辑的处理；

属性描述：

无；

方法详细描述：

OrderSet(OrderInfo): bool

功能说明：创建订单，并需要利用调用订单微服务插入相应信息的订单，调用支付微服务接口提供支付的功能；

接口调用：

OrderPay(OrderPayInfo) 实现支付功能调用

AddOrder(Order) 实现对新建订单的插入

详细说明

|  |  |  |
| --- | --- | --- |
| 返回类型 | Bool | |
| 返回值 | 支付页面的支付成功结果 | |
| 方法限定符 | Public | |
| 参数列表（参数名称 / 数据类型 / 说明） | | |
| orderInfo | OrderInfo | 输入的订单信息 |

实现过程：

调用初始订单创建的过程，首先将订单微服务获取的订单信息插入到数据库中，再生成对应的OrderPayInfo调用支付微服务支付接口实现对支付订单的提交，最后通过实现支付完成整体订单的创建。

OrderReFund(OrderInfo): bool

功能说明：实现订单退款的业务逻辑，并需要利用调用支付微服务的退款接口，退款完成后调用订单微服务接口提供改变订单状态的功能；

接口调用：

OrderRefund(OrderPayInfo) 实现退款功能调用

UpdateOrderStatus(Order) 实现对订单状态的更新

详细说明

|  |  |  |
| --- | --- | --- |
| 返回类型 | Bool | |
| 返回值 | 退款操作结果的布尔值 | |
| 方法限定符 | Public | |
| 参数列表（参数名称 / 数据类型 / 说明） | | |
| orderInfo | OrderInfo | 取消订单的待退款订单信息 |

实现过程：

调用该函数实现对订单退款的业务过程。首先调用支付微服务的退款接口实现对订单的退款功能，然后再调用订单微服务的更新订单状态的操作，完成取消订单退款整体流程。

OrderComplete(OrderId): bool

功能说明：支付完成后会调用该接口实现对订单支付流程的确定与订单状态的修改

接口调用：

UpdateOrderStatus(orderId, Status) 实现对订单状态的修改

详细说明：

|  |  |  |
| --- | --- | --- |
| 返回类型 | Bool | |
| 返回值 | 订单完成操作的是否成功布尔值 | |
| 方法限定符 | Public | |
| 参数列表（参数名称 / 数据类型 / 说明） | | |
| orderId | OrderId | 支付完成的订单Id |

实现过程：

该函数实现订单完成的业务流程，首先函数调用订单微服务完成对订单状态的修改，然后并在数据库中进行相应的修改。

类内、类外方法调用状态图

该类无状态变化过程

限制条件及出错处理

无

### 3.7.3 IOrderSerice类

功能：

该类为交易微服务对外提供的整体接口，通过暴露接口实现对订单交易过程的控制操作；

性能：

无约束

输入项：

接收订单信息实现对订单交易的业务逻辑；

输出项：

输出经过订单交易业务操作后的执行状态；

属性描述：

无；

方法详细描述：

OrderSet(OrderInfo): bool

接口调用：

OrderPay(OrderPayInfo) 实现支付功能调用

AddOrder(Order) 实现对新建订单的插入

实际调用方法：

OrderSerice. OrderSet(OrderInfo)

OrderReFund(OrderInfo): bool

接口调用：

OrderRefund(OrderPayInfo) 实现退款功能调用

UpdateOrderStatus(Order) 实现对订单状态的更新

实际调用方法：

OrderSerice. OrderReFund(OrderInfo)

OrderComplete(OrderId)

接口调用：

UpdateOrderStatus(orderId, Status) 实现对订单状态的修改

实际调用方法：

OrderSerice. OrderComplete(OrderId)

类内、类外方法调用状态图：

该类无状态变化过程;

限制条件及出错处理：

无