



归宿——一个民宿预定平台

详细设计规约



1851055 汪明杰

1851049 陈中悦

1851231 王立友

1853572 梁 乔

1954098 香宁雨

目录

1	引言	4
1.1	编写目的	4
1.2	背景及依据	4
1.3	参考资料	4
2	系统软件结构	5
3	微服务设计	5
4	类设计	6
4.1	登陆注册微服务	6
4.1.1	User 类	6
4.1.2	SignupManager 类	9
4.1.3	SMSManager 类	1 3
4.1.4	ResidentVerifier 类	1 4
4.1.5	IdGenerator 类	1 6
4.1.6	AuthenticationCkecking 类	1 6
4.1.7	Token 类	1 8
4.1.8	TokenManager 类	2 0
4.1.9	LoginInterface 类	2 3
4.2	论坛微服务	2 5
4.2.1	Post 类	2 5
4.2.2	PostManager 类	2 6
4.2.1	Reply 类	2 9
4.2.2	ReplyManager 类	3 0
4.2.3	Report 类	3 2
4.2.4	ReportManager 类	3 4
4.2.5	PostInterface 类	3 6
4.3	房源管理微服务	3 7
4.3.1	Bed 类	3 8
4.3.2	Room 类	4 0
4.3.3	Label 类	4 1
4.3.4	Advertise 类	4 3
4.3.5	Stay 类	4 4
4.3.6	RoomManager 类	4 7
4.3.7	StayManager 类	5 0
4.3.8	StayInterface 类	5 2
4.4	数据统计微服务	5 3
4.4.1	StayManager 类	5 3
4.4.2	OrderManager 类	5 4
4.4.3	StatisticsInterface 类	5 6

4.5	用户信息微服务	5 6
4.5.1	Customer 类	5 7
4.5.2	Host 类	5 9
4.5.3	FavoriteDir 类	6 0
4.5.4	UserGroup 类	6 2
4.5.5	HostGroup 类	6 4
4.5.6	FavoriteManager 类	6 5
4.5.7	PersonInfoManager 类	6 8
4.5.8	OSSConfigureManager 类	7 1
4.5.9	PersonInfoInterface 类	7 2
4.6	管理微服务	7 5
4.6.1	Report 类	7 5
4.6.2	UnverifiedStay 类	7 7
4.6.3	ReportManager 类	7 9
4.6.4	AdministrationInterface 类	8 1
4.7	促销微服务	8 2
4.7.1	Coupon 类	8 3
4.7.2	CouponType 类	8 6
4.7.3	CouponUsageManager 类	8 7
4.7.4	CouponGrantManager 类	8 9
4.7.5	SaleInterface 类	9 1
4.8	订单微服务	9 2
4.8.1	Order 类	9 3
4.8.2	Comment 类	9 5
4.8.3	Report 类	9 7
4.8.4	OrderManager 类	9 9
4.8.5	ReportManager 类	1 0 4
4.8.6	CommentManager 类	1 0 6
4.8.7	OrderInterface 类	1 0 8
4.9	支付微服务	1 1 1
4.9.1	AlipayOrder 类	1 1 1
4.9.2	AlipayConfig 类	1 1 3
4.9.3	PaymentManager 类	1 1 4
4.9.4	CallbackManager 类	1 1 6
4.9.5	PaymentInterface 类	1 1 8
4.10	交易微服务	1 1 9
4.10.1	OrderInfo 类	1 2 0
4.10.2	RedisUtils 类	1 2 2
4.10.3	TradeManager 类	1 2 3
4.10.4	TradeInterface 类	1 2 6
5	索引	1 2 8

1 引言

1.1 编写目的

本文档是用于细化归宿平台系统的详细设计的，我们将通过微服务设计、接口设计、类详细设计三个角度规范归宿平台开发中的编码方向、不同微服务之间的接口调用以及方法设计等。

1.2 背景及依据

本项目名为**归宿**，旨在为 PC 端的用户提供一个体验良好，灵活性强，交互性好的轻量级民宿预订与服务平台，用户可以通过该网站随时预定民宿入住的服务，并通过其他相关的功能获得更好的体验。

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

1.3 参考资料

1. 《SpringBoot 更好的开发》
2. 《HTML5+CSS3 从入门到精通》
3. 《更好的软件架构，更好的设计》
4. 《Design Patterns》
5. 《Clean Architecture》
6. 《Clean Code》

2 系统软件结构

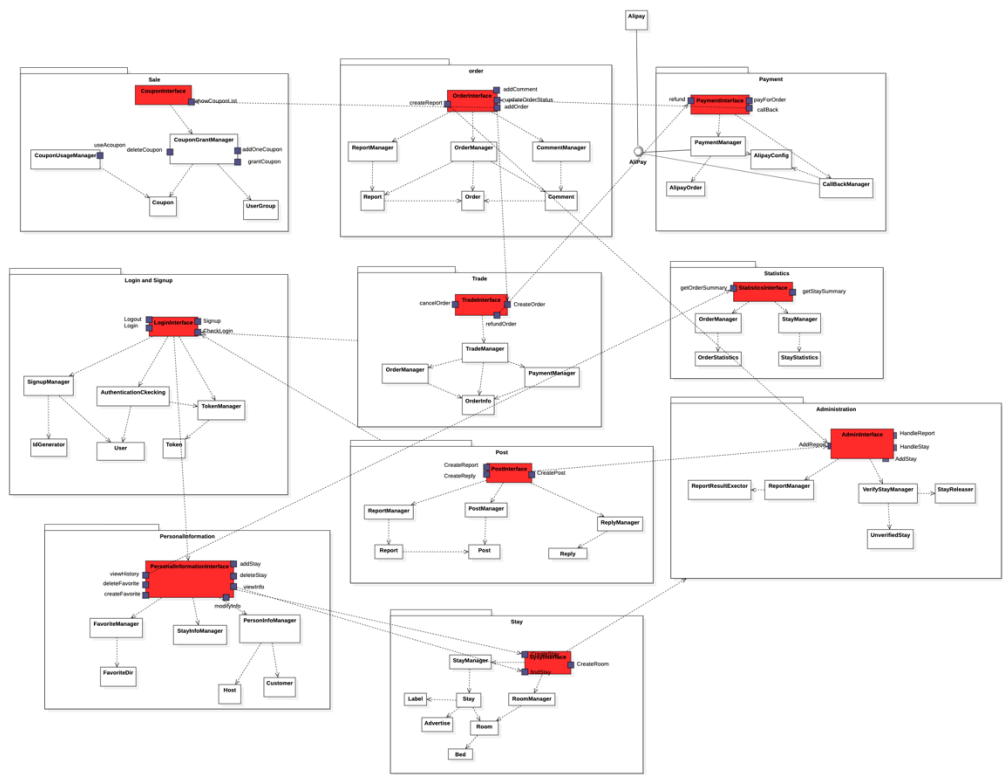


图 1 归宿平台系统软件结构图

3 微服务设计

归宿平台在微服务划分上一共分为 10 个微服务和一个 API Gateway、一个 Eureka 服务发现与注册中心。API Gateway 的功能是提供统一的后端访问入口，Eureka 的作用是提供统一的服务 IP 地址管理。其余微服务的功能将在接下来详细类设计中说明。

以下是基于微服务的系统架构划分图：

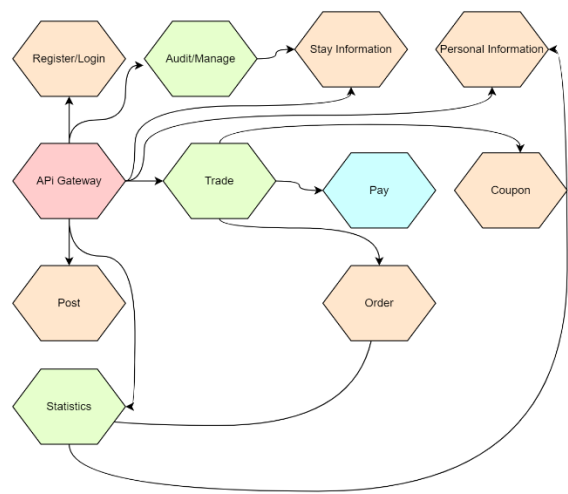


图 2 归宿平台微服务划分调用图

4 类设计

4.1 登陆注册微服务

这个在登陆注册微服务中，一共有两个实体类，User 与 Token，用来管理用户实体记录以及登录状态的管理。同时通过三个控制类来完成用户登陆、注册以及登陆标识管理的工作。三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：

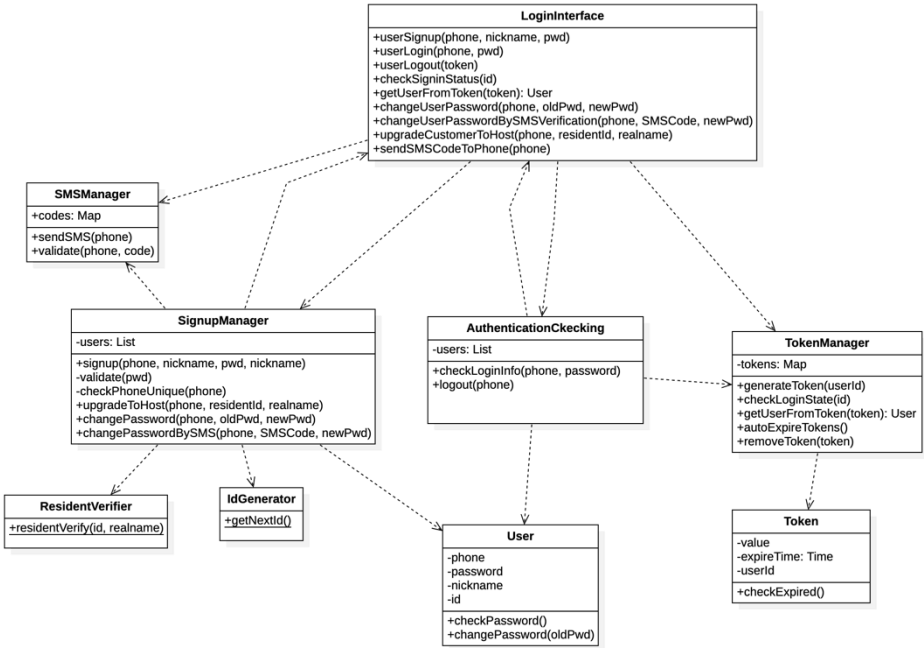


图 3 登陆注册微服务详细类设计

接下来将会详细介绍这几个类的描述。

4.1.1 User 类

4.1.1.1 功能

登陆注册模块中的 User 类主要负责记录用户的登录账号（在归宿系统中是手机号码或者昵称）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。

4.1.1.2 性能

无约束

4.1.1.3 输入项

该类通过用户的手机号、昵称以及密码进行初始化。修改密码时，需要提供新、旧两个密码进行验证。

4.1.1.4 输出项

检查密码时，只会返回“正确”或“错误”进行反馈（布尔值）。其余情况下没有输出项。

4.1.1.5 属性描述

属性名	类型	说明
phone	String	用户的电话号码。接受该参数初始化,之后不会再改变。初始化时会检查电话号码是否符合真实电话号码格式要求。
password	String	用户的密码。设置时要求满足一定安全强度（6位以上，由数字、大写字母、小写字母、和至少一个符号组成）。
nickname	String	用户的昵称。不允许超过 20 个字符。

表格 1 User 类属性表

4.1.1.6 方法详细描述

4.1.1.6.1 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。
-----	--------	------------

表格 2 checkPassword 方法说明

4.1.1.6.2 changePassword

方法签名：public Boolean changePassword(String newPwd, String oldPwd)
功能说明：在验证旧密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，对于改密码会做安全性认证，若过于简单不会更新密码。
oldPwd	String	输入的旧密码，需要与之前的密码匹配，否则也会修改失败。

表格 3 changePassword 方法说明

4.1.1.1 类内、类外方法调用状态图

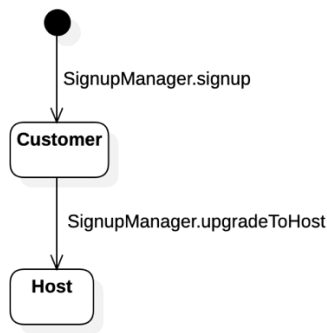


图 4 User 类方法调用状态图

4.1.1.2 限制条件及出错处理

- 用户电话号码需要符合中国真实电话号码的格式，第一位为 1，第二位是 3、6、9 中的一位，共 11 位组成。若不满足该条件，User 类会抛出一个FormatException 的自定义异常。
- 用户的密码要符合“6 位以上，由数字、大写字母、小写字母、和至少一个符号组

成”的规则，否则 User 类会抛出一个 FormatException 的自定义异常。

4.1.2 SignupManager 类

4.1.2.1 功能

管理用户的注册行为，提供用户通过手机号码、用户昵称和密码注册的功能。同时还会通过帮助类获取全局唯一的用户标识符（ID）赋予注册成功的用户。

4.1.2.2 性能

允许并发注册用户。

4.1.2.3 输入项

- 1. 该类维护了一个用户列表，通过接受用户的手机号码、用户昵称和密码来进行注册；
- 2. 同时该类还可以单独通过手机号码来判断用户是否已经注册成为系统的用户；
- 3. 用户输入密码可以在该类中检查是否符合要求。

4.1.2.4 输出项

- 1. 用户注册是否成功，布尔值；
- 2. 该手机号是否已经注册系统中的用户，布尔值；
- 3. 改密码是否符合格式要求，布尔值。

4.1.2.5 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，只要成功注册就会记录在此处。该列表不一定恒在内存中，可以通过 ORM 机制存储、或者从数据库中取回。

表格 4 SignupManager 类属性表

4.1.2.6 方法详细描述

4.1.2.6.1 signup

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

功能说明：通过参数用户手机号码、用户昵称和密码来进行注册。

接口调用：

- 调用自身 validate(pwd)方法，检查用户的密码是否符合“6 位以上，由数字、大写字母、小写字母、和至少一个符号组成”的规则，若不是则注册失败；
- 调用自身 checkPhoneUnique(phone)方法，检查想要注册账号的用户是否手机号是否是新注册的，若不是则注册失败。

详细说明：

返回类型	Boolean	
返回值	用户是否登陆成功，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。

表格 5 signup 方法说明

4.1.2.6.2 validate

方法签名：private Boolean validate(String Pwd)

功能说明：验证用户密码是否满足“6 位以上，由数字、大写字母、小写字母、和至少一个符号组成”的规则。

详细说明：

返回类型	Boolean	
返回值	用户密码是否满足规则。	
方法限定符	private	
参数列表（参数名 / 类型 / 说明）		
pwd	String	用户待检查的密码。

表格 6 validate 方法说明

4.1.2.6.3 checkPhoneUnique

方法签名：public Boolean checkPhoneUnique(String phone)

功能说明：验证手机号是否是新注册该系统的手机号。

详细说明：

返回类型	Boolean	
返回值	用户的手机号是否是新注册的，若是返回 true，若已有用户用相同的手机号注册过返回 false。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
phone	String	待检查的手机号。

表格 7 checkPhoneUnique 方法说明

4.1.2.6.4 upgradeToHost

方法签名：public Boolean upgradeToHost(String phone, String residentId, String realname)

功能说明：验证和记录用户身份证号并将用户升级成房东。

接口调用：

- 调用 ResidentVerifier 类中的 residentVerify 方法，验证房东的身份信息。

详细说明：

返回类型	Boolean	
返回值	用户是否成功升级成房东。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
phone	String	用户手机号，用来标识用户身份。
residentId	String	用户的身份证号，该号码会与用户的真实姓名一起通过外部的服务查验是否真实。
realname	String	用户的真实姓名。

表格 8 upgradeToHost 方法说明

4.1.2.6.5 changePassword

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

功能说明：在验证旧密码正确后修改用户的密码为用户指定的新密码，修改密码前要验证密码是否符合规范。

接口调用：

- 调用自身的 validate 方法，验证新密码格式的正确性。
- 调用目标用户的 changePassword 方法，完成对用户密码的修改。

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）	

phone	String	用户的手机号码，用于唯一标识用户。
newPwd	String	输入的新密码，对于改密码会做安全性认证，若过于简单不会更新密码。
oldPwd	String	输入的旧密码，需要与之前的密码匹配，否则也会修改失败。

表格 9 changePassword 方法说明

4.1.2.6.6 changePasswordBySMS

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

功能说明：在通过短信验证码验证后修改用户的密码为用户指定的新密码，修改密码前要验证密码是否符合规范。

接口调用：

- 调用自身的 validate 方法，验证新密码格式的正确性。
- 调用 SMSManager 的 validate 方法，用于验证手机验证码是否正确。只有正确的手机验证码才能完成成功修改密码。
- 调用目标用户的 changePassword 方法，完成对用户密码的修改。

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）		
phone	String	用户的手机号码，用于唯一标识用户。
SMSCode	String	用于验证的短信验证码，短信验证码会通过 SMSManager 类发送和记录，此处要通过 SMSManager 来检验验证码的正确与否。
newPwd	String	输入的新密码，对于改密码会做安全性认证，若过于简单不会更新密码。

表格 10 changePasswordBySMS 方法说明

4.1.2.7 限制条件及出错处理

- 用户电话号码需要符合中国真实电话号码的格式，第一位为 1，第二位是 3、6、9 中的一位，共 11 位组成。若不满足该条件，User 类会抛出一个 FormatException 的自定义异常。
- 用户的密码要符合“6 位以上，由数字、大写字母、小写字母、和至少一个符号组成”的规则，否则 User 类会抛出一个 FormatException 的自定义异常。

4.1.3 SMSManager 类

4.1.3.1 功能

接受用户的手机号码，给指定用户发送验证码。调用了第三方的 SMS 服务，并且会在系统中暂存所发送的用户手机号和配对的验证码。还可以接受用户对于手机验证码的验证。

4.1.3.2 性能

允许一秒钟发送验证码给 20 个用户。即允许并发发送验证码。

4.1.3.3 输入项

1. 接受用户的手机号码，自动生成验证码和短信内容发送给用户；
2. 接受用户的手机号码以及验证码，判断手机号码是否与发送的短信验证码记录相符。

4.1.3.4 输出项

1. 无输出，但是会调用第三方的短信服务；
2. 输出验证码是否正确；

4.1.3.5 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，只要成功注册就会记录在此处。该列表不一定恒在内存中，可以通过 ORM 机制存储、或者从数据库中取回。

表格 11 SMSManager 类属性表

4.1.3.6 方法详细描述

4.1.3.6.1 sendSMS

方法签名：public void sendSMS(String phone)

功能说明：自动生成验证码和短信内容发送给用户。

接口调用：

- 调用第三方的短信服务平台 API，通过 API 给用户发送指定内容的短信。

详细说明：

返回类型	void	
返回值	无返回值，若发送验证码失败会抛出异常。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
phone	String	想要发送验证码用户的手机号码。

表格 12 sendSMS 方法说明

4.1.3.6.2 validate

方法签名：private Boolean validate(String phone, String code)

功能说明：验证输入的验证码是否是系统最新的发送给用户手机的验证码。

详细说明：

返回类型	Boolean	
返回值	验证码是否正确，正确返回 true，错误返回 false。	
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。
code	String	用户提供的短信验证码内容。

表格 13 validate 方法说明

4.1.3.7 限制条件及出错处理

- 用户电话号码需要符合中国真实电话号码的格式，第一位为 1，第二位是 3、6、9 中的一位，共 11 位组成。且用户手机号是真实存在的，否则会抛出一个异常。
- 同一用户一分钟之内只能请求一次短信息服务，多余的请求会被直接忽略。

4.1.4 ResidentVerifier 类

4.1.4.1 功能

接受用户提供的身份证号码和真实姓名，判断该身份证是否属实且征信状态如何，是否能注册成房东。调用了第三方的身份认证服务（模拟）。

4.1.4.2 性能

允许一秒钟同时验证 5 名用户。

4.1.4.3 输入项

用户身份证号、用户真实姓名。

4.1.4.4 输出项

是否能注册成房东。

4.1.4.5 方法详细描述

4.1.4.5.1 residentVerify

方法签名：static public Boolean residentVerify(String id, String realname)

功能说明：判断该身份证是否属实且征信状态如何，是否能注册成房东。

接口调用：

- 调用第三方的身份认证服务平台 API，通过 API 认证身份信息真伪（模拟）。

详细说明：

返回类型	Boolean	
返回值	判断该身份证是否属实且征信状态如何，能够注册成房东返回 true，不能注册返回 false。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
residentId	String	用户填写的身份证号码。
realname	String	用户填写的真实姓名。

表格 14 residentVerify 方法说明

4.1.4.6 限制条件及出错处理

- 用户身份证号码不符合格式的会直接返回 false。

4.1.5 IdGenerator 类

4.1.5.1 功能

通过静态方法提供全局唯一的用户 ID。

4.1.5.2 输出项

通过雪花算法生成的全局唯一的用户 ID 值，保证不会重复。

4.1.5.3 方法详细描述

4.1.5.3.1 getNextId

方法签名：static public Long getNextId()
功能说明：通过雪花算法生成的全局唯一的用户 ID 值。
详细说明：

返回类型	Long
返回值	全局唯一的用户 ID 值。
方法限定符	public

表格 15 getNextId 方法说明

4.1.6 AuthenticationCkecking 类

4.1.6.1 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。还能通过用户的 ID 查看用户的登录情况。

4.1.6.2 性能

允许并发登录，一秒钟可以接受 50 个登陆请求。

4.1.6.3 输入项

1. 接受用户的手机号码和密码，通过手机号码和密码来验证、登录用户；
2. 接受用户的 ID，查询用户当前的登录情况。

4.1.6.4 输出项

1. 输出用户是否登录成功；
2. 输出用户的登录状态。

4.1.6.5 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，只要成功注册就会记录在此处。该列表不一定恒在内存中，可以通过 ORM 机制存储、或者从数据库中取回。

表格 16 AuthenticationCkecking 类属性表

4.1.6.6 方法详细描述

4.1.6.6.1 checkLoginInfo

方法签名：public void checkLoginInfo(String phone, String password)

功能说明：通过电话号码获取用户，然后比对用户的信息，判断密码是否正确，若正确则记录用户的登录状态。

接口调用：

- 若用户登录成功，则调用 TokenManager 中的 generateToken 方法。

详细说明：

返回类型	Boolean	
返回值	用户是否成功登陆，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）		
phone	String	用户的手机号码。
pwd	String	用户输入用于验证的密码。

表格 17 sendSMS 方法说明

4.1.6.6.2 logout

方法签名: private void logout(String phone)

功能说明: 登出一位用户。

接口调用:

- 调用 TokenManager 中的 removeToken 方法。

详细说明:

返回类型	void	
返回值	无返回值，若用户存在登录记录则登出，若没有记录则不会发生任何副作用。	
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。

表格 18 validate 方法说明

4.1.7 Token 类

4.1.7.1 功能

用户登陆后会产生一个 token 保存在系统中, 用户在登录之后, 与系统交互的过程中不需要每次都发送密码来验证, 只需要在 HTTP 请求头部包含一个 Cookie 信息, 里面有由系统发派的 token, 就能自动完成登录状态的判断。

4.1.7.2 性能

无约束

4.1.7.3 输入项

该类通过用户的 ID 来初始化, 通过当前系统时间和指定的过期时间 (该系统中是 2 小时不活跃状态), 通过算法来生成一枚 token。

4.1.7.4 输出项

用户可以通过调用 token 类的方法来进行查看 token 过期的情况。输出值表示 token 是否过期。

4.1.7.5 属性描述

属性名	类型	说明
value	String	token 的值，一个例子是：579024cb-bf39-4c9c-babb-e9a4af76b9bf。
expireTime	DateTime	token 过期的时间。
userId	Long	token 对应的登录用户的 ID。

表格 19 User 类属性表

4.1.7.6 方法详细描述

4.1.7.6.1 checkExpired

方法签名：public Boolean checkExpired()
功能说明：查看该 token 当前的过期状态。
详细说明：

返回类型	Boolean
返回值	token 是否过期，true 为未过期，false 为已过期。已过期的 token 会在被扫描到的同时被 tokenManager 实体移除。
方法限定符	public

表格 20 checkExpired 方法说明

4.1.7.7 类内、类外方法调用状态图

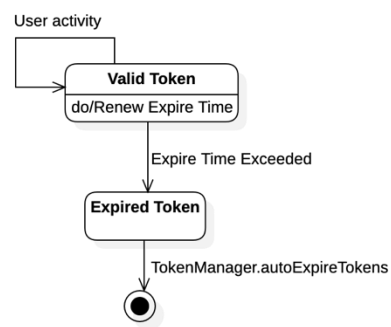


图 5 Token 类方法调用状态图

4.1.7.8 限制条件及出错处理

- token 的格式为 UUID，一个例子是：579024cb-bf39-4c9c-babb-e9a4af76b9bf。

4.1.8 TokenManager 类

4.1.8.1 功能

管理用户登陆后产生的 token，用户在登录之后，与系统交互的过程中不需要每次都发送密码来验证，只需要在 HTTP 请求头部包含一个 Cookie 信息，里面有由系统发派的 token，就能自动完成登录状态的判断。其余的服务会请求到该模块的 API 来完成 token 的验证。

4.1.8.2 性能

允许并发产生唯一的 token。

4.1.8.3 输入项

- 接受用户 ID，为该 ID 产生一个 token；
- 判断某个 ID 用户的登录状态；
- 接受一个 token，返回储存的用户信息；
- 移除一个 token。

4.1.8.4 输出项

- 与用户 ID 对应的 token 值；
- 用户的登录状态；
- Token 对应用户的详细信息；
- 无输出。

4.1.8.5 属性描述

属性名	类型	说明
tokens	Map	储存一个 token、用户信息组成的键值对，可以通过 token 查询到用户的详细信息，也可以通过遍历的方式来找到用户拥有的 token。

表格 21 TokenManager 类属性表

4.1.8.6 方法详细描述

4.1.8.6.1 generateToken

方法签名: `public String generateToken(Long userId)`

功能说明: 通过输入用户的 ID 来产生一个唯一的 token (20 位 UUID 格式)。

详细说明:

返回类型	String	
返回值	一个当前全局唯一的 token	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
userId	Long	用户的 ID。

表格 22 generateToken 方法说明

4.1.8.6.2 checkLoginState

方法签名: `private Boolean checkLoginState(Long UserId)`

功能说明: 通过输入用户的 ID 来判断用户当前的登录状态。

详细说明:

返回类型	Boolean	
返回值	用户是否登录且 token 未过期，true 为登录状态，false 为未登录或 token 过期已被自动清理掉。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
userId	Long	用户的 ID。

表格 23 checkLoginState 方法说明

4.1.8.6.3 getUserFromToken

方法签名: `public User getUserFromToken(String token)`

功能说明: 通过一个 token 值来获取当前登录状态的用户。

详细说明:

返回类型	User	
返回值	token 对应的当前登录状态的用户, 若未登陆或者 token 无效, 会抛出一个用户未登录的异常, 对于 HTTP 请求, 返回一个错误码 401 的相应。	
方法限定符	public	

参数列表（参数名 / 类型 / 说明）		
token	String	登录状态的一个 token。

表格 24 getUserFromToken 方法说明

4.1.8.6.4 autoExpireTokens

方法签名：public void autoExpireTokens()

功能说明：系统每隔一段时间自动调用的方法，用于自动过期登陆时长超过 2 小时并且没有活跃行为的用户 token。

详细说明：

返回类型	void
返回值	无返回值。
方法限定符	public

表格 25 autoExpireTokens 方法说明

4.1.8.6.5 removeToken

方法签名：public void removeToken(String token)

功能说明：手动移除登陆的 token。

详细说明：

返回类型	void	
返回值	无返回值，若存在 token 则移除，若没有记录则不会发生任何副作用。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
token	String	想要移除记录的 token 值。

表格 26 removeToken 方法说明

4.1.8.7 限制条件及出错处理

- token 的格式为 UUID，一个例子是：579024cb-bf39-4c9c-babb-e9a4af76b9bf。
- 通过 token 获取用户时，若 token 不存在或格式不对，系统会抛出一个未登录的异常（NotLoggedIn），对于 HTTP 响应，会返回一个 401（Unauthorized）状态码。

4.1.9 LoginInterface 类

4.1.9.1 功能

完成模块之间的接口调用交互。

4.1.9.2 接口方法详细描述

4.1.9.2.1 userSignup

方法签名：public Boolean userSignup(String phone, String nickname, String pwd)

实际调用方法：SignupManager.signup

4.1.9.2.2 userLogin

方法签名：public Boolean userSignup(String phone, String nickname, String pwd)

实际调用方法：SignupManager.signup

4.1.9.2.1 userLogin

方法签名：public Boolean userLogin(String phone, String pwd)

实际调用方法：AuthenticationCkecking.checkLoginInfo

4.1.9.2.1 userLogout

方法签名：public void userLogout(String token)

实际调用方法：AuthenticationCkecking.logout

4.1.9.2.1 checkSigninStatus

方法签名：public Boolean checkSigninStatus(String id)

实际调用方法：TokenManager.checkLoginState

4.1.9.2.1 `getUserFromToken`

方法签名: `public User getUserFromToken(String token)`

实际调用方法: `TokenManager.getUserFromToken`

4.1.9.2.1 `changeUserPassword`

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

实际调用方法: `SignupManager.changePassword`

4.1.9.2.1 `changeUserPasswordBySMSVerification`

方法签名: `public Boolean changeUserPasswordBySMSVerification(String phone, String SMSCode, String newPwd)`

实际调用方法: `SignupManager.changePasswordBySMS`

4.1.9.2.1 `upgradeCustomerToHost`

方法签名: `public Boolean upgradeCustomerToHost(String phone, String residentId, String realname)`

实际调用方法: `SignupManager.upgradeToHost`

4.1.9.2.1 `sendSMSCodeToPhone`

方法签名: `public Boolean sendSMSCodeToPhone(in phone)`

实际调用方法: `SMSManager.sendSMS`

4.2 论坛微服务

这个在论坛微服务中，一共有三个实体类，Post、Reply 与 Report，其中 Post 类是用户发送在系统中保存的帖子类，Reply 是针对帖子或其他回复的恢复，而 Report 类则是用户举报帖子的记录。下图所示是该微服务的详细设计：

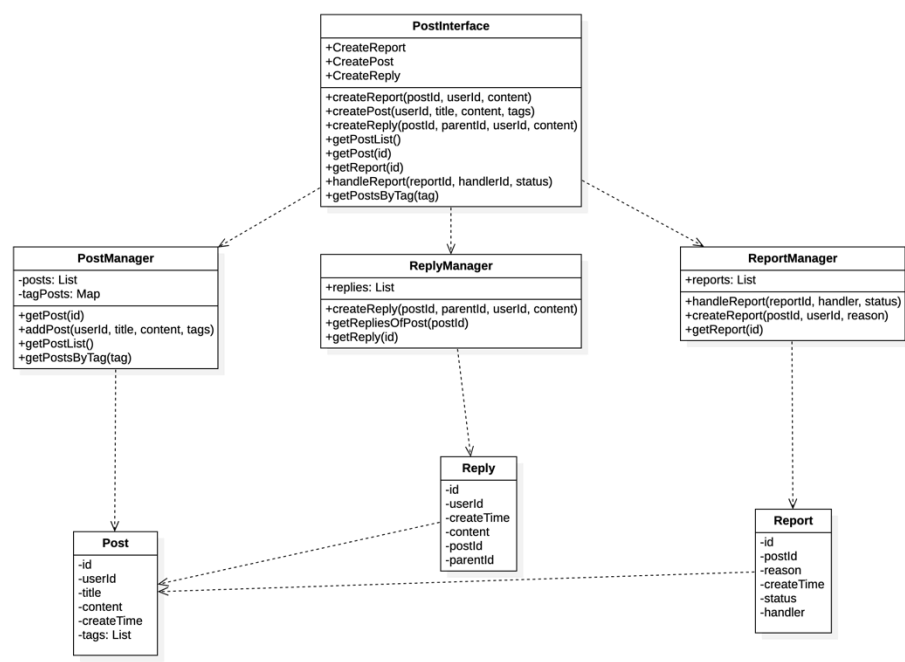


图 6 论坛微服务详细类设计

接下来将会详细介绍这几个类的描述。

4.2.1 Post 类

4.2.1.1 功能

Post 类储存了用户发送帖子的详细信息，如用户 ID、发帖时间、帖子标题、发帖内容等。这是一个 ORM 的实体类型，因此对应着数据库中的一张表，主要的职责就是用于记录数据，除了设置数据 (setter) 和获取数据 (getter) 之外没有其他操作。

4.2.1.2 性能

无约束

4.2.1.3 输入项

该类通过帖子作者的用户 ID、帖子标题、帖子内容、创建时间、帖子的标签等信息进行初始化。

4.2.1.4 输出项

通过读取数据方法（getters，类图中并未画出）可以获取相应的属性值。其余情况下没有输出项。

4.2.1.5 属性描述

属性名	类型	说明
id	Long	帖子的唯一标识符。
userId	Long	帖子的作者用户 ID。
title	String	帖子的标题。
content	String	帖子内容，要求不超过 1000 字。
createTime	Time	帖子的创建时间，由系统自动记录。
tags	List<String>	帖子的标签。

表格 27 Post 类属性表

4.2.1.6 限制条件及出错处理

- 帖子的长度不能超过 1000 字，标题不能超过 50 字，标签（tags）不能超过 5 个。所有内容超出限制的部分都会被截断。

4.2.2 PostManager 类

4.2.2.1 功能

管理帖子的类型，通过维持一个 ORM 映射和连接池来控制 Post 类实例的持久化和获取操作。可以完成帖子发布的处理、获取帖子、获取列表和按照标签来获取列表。

4.2.2.2 性能

允许并发发布、查询帖子。

4.2.2.3 输入项

1. 接受帖子 ID，返回对应的帖子；
2. 接受帖子的内容信息，创建新帖；
3. 接受分页页数即页帖子数量，调用方法即获取帖子列表；
4. 接受一个存在的 tag 标签，返回标签对应的所有帖子列表。

4.2.2.4 输出项

1. ID 对应的帖子；
2. 帖子 ID，出错时为 null；
3. 帖子的分页列表；
4. Tag 对应的所有帖子。

4.2.2.5 属性描述

属性名	类型	说明
posts	List	系统中当前的所有帖子，只要成功发布就会记录在此处。该列表不一定恒在内存中，可以通过 ORM 机制存储、或者从数据库中取回。

表格 28 PostManager 类属性表

4.2.2.6 方法详细描述

4.2.2.6.1 getPost

方法签名：public Post getPost (Long id)

功能说明：返回 ID 对应的帖子，若帖子不存在，则抛出异常。

详细说明：

返回类型	Post	
返回值	输入 ID 对应的帖子，若帖子不存在，则抛出异常。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
id	Long	帖子的 ID。

表格 29 getPost 方法说明

4.2.2.6.2 addPost

方法签名: private Long addPost(Long userId, String title, String content, List<String> tags)

功能说明: 根据输入的用户 (作者) ID、帖子标题、帖子内容、帖子的标签来创建一片新的帖子。新帖子的 ID 是一个流水号, 即为当前系统总帖子数量+1。

详细说明:

返回类型	Long	
返回值	新建帖子的 ID，若帖子创建失败（如：没有这个用户、标题内容为空等），则返回 null。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
userId	Long	用户的 ID。
title	String	帖子的标题。
content	String	帖子的内容。
tags	List<String>	帖子的标签列表，要求不超过 5 个。

表格 30 addPost 方法说明

4.2.2.6.3 getPostList

方法签名: public List<Post> getPostList(int page, int size)

功能说明: 获取系统所有的帖子。

详细说明:

返回类型	List<Post>	
返回值	系统所有的帖子按照分页页数和分页大小的限制返回。若该分页没有内容，也会返回一个空的列表。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
page	int	分页页数。
size	int	分页大小。

表格 31 getPostList 方法说明

4.2.2.6.4 getPostsByTag

方法签名: public List<Post> autoExpireTokens(String tag)

功能说明: 通过输入的帖子标签返回对应的帖子列表。

详细说明:

返回类型	List<Post>	
返回值	满足标签中含有输入值的帖子, 若没有帖子被查出, 也会返回一个空的列表。	

方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
tag	String	指定需要返回的帖子标签。

表格 32 autoExpireTokens 方法说明

4.2.2.7 限制条件及出错处理

- 帖子的标题和内容不能为空，否则会抛出一个格式异常。HTTP 会返回 400 状态码标识错误。

4.2.1 Reply 类

4.2.1.1 功能

Reply 类储存了用户发送帖子评论的详细信息，如用户 ID、发送时间、评论内容等。这是一个 ORM 的实体类型，因此对应着数据库中的一张表，主要的职责就是用于记录数据，除了设置数据（setter）和获取数据（getter）之外没有其他操作。

4.2.1.2 性能

无约束

4.2.1.3 输入项

该类通过帖子作者的用户 ID、回复内容、回复帖子的 ID、回复回复的 ID（系统内允许嵌套回复，即可以回复已有的回复）创建时间等信息进行初始化。

4.2.1.4 输出项

通过读取数据方法（getters，类图中并未画出）可以获取相应的属性值。其余情况下没有输出项。

4.2.1.5 属性描述

属性名	类型	说明
-----	----	----

id	Long	回复的唯一标识符。
userId	Long	回复的作者用户 ID。
postId	Long	回复属于的帖子。
content	String	回复内容，要求不超过 200 字。
createTime	Time	回复的创建时间，由系统自动记录。
parentId	Long	回复的父级回复，即用户希望回复的回复。

表格 33 Post 类属性表

4.2.1.6 限制条件及出错处理

- 回复的长度不能超过 200 字。所有内容超出限制的部分都会被截断。

4.2.2 ReplyManager 类

4.2.2.1 功能

管理回复的类型，通过维持一个 ORM 映射和连接池来控制 Reply 类实例的持久化和获取操作。可以完成帖子发布的处理、获取帖子、获取列表和按照标签来获取列表。

4.2.2.2 性能

允许并发发布、查询回复。

4.2.2.3 输入项

1. 接受回复的内容信息，创建新回复；
2. 接受帖子 ID 和分页参数，返回帖子的某页回复列表；
3. 接受一个回复 ID，返回对应回复的内容。

4.2.2.4 输出项

1. 新建回复 ID，出错时为 null；
2. 指定回复的分页列表；
3. 一个回复的详细内容。

4.2.2.5 属性描述

属性名	类型	说明
replies	List	系统中当前的所有帖子，只要成功发布就会记录在此处。该列表不一定恒在内存中，可以通过 ORM 机制存储、或者从数据库中取回。

表格 34 PostManager 类属性表

4.2.2.6 方法详细描述

4.2.2.6.1 createReply

方法签名：public Long createReply(Long postId, Long parentId, Long userId, String content)

功能说明：创建一个新回复，返回新回复的 ID。若出错，则返回 null。

详细说明：

返回类型	Long	
返回值	新回复的 ID，若出错，则返回 null。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
postId	Long	回复的帖子的 ID。
parentId	Long	回复的父级 ID。
userId	Long	回复用户（作者）的 ID。
content	String	回复内容。

表格 35 createReply 方法说明

4.2.2.6.2 getRepliesOfPost

方法签名：public List<Reply> getRepliesOfPost(Long postId, int page, int size)

功能说明：获取指定帖子下的所有回复的一个分页。

详细说明：

返回类型	List<Reply>	
返回值	某个帖子的所有回复内容按照分页页数和分页大小的限制返回。若该分页没有内容，也会返回一个空的列表。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
postId	Long	指定帖子 ID。
page	int	分页页数。

size	int	分页大小。
------	-----	-------

表格 36 getRepliesOfPost 方法说明

4.2.2.6.3 getReply

方法签名：public Reply getReply(Long id)

功能说明：通过输入的帖子 ID 返回对应的帖子列表。

详细说明：

返回类型	Reply	
返回值	ID 对应的帖子，若帖子不存在，则返回 null。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
id	Long	帖子 ID。

表格 37 getReply 方法说明

4.2.2.7 限制条件及出错处理

- 回复内容不能为空，否则会抛出一个格式异常。HTTP 会返回 400 状态码标识错误。

4.2.3 Report 类

4.2.3.1 功能

Report 类储存了用户举报帖子的详细信息，如用户 ID、帖子 ID、发送时间、举报原因等。这是一个 ORM 的实体类型，因此对应着数据库中的一张表，主要的职责就是用于记录数据，除了设置数据（setter）和获取数据（getter）之外没有其他操作。

4.2.3.2 性能

无约束

4.2.3.3 输入项

该类通过举报内容的帖子 ID、用户 ID、举报原因、创建时间、举报处理人（一个管理

员) 等信息进行初始化。

4.2.3.4 输出项

通过读取数据方法 (getters, 类图中并未画出) 可以获取相应的属性值。其余情况下没有输出项。

4.2.3.5 属性描述

属性名	类型	说明
id	Long	举报的唯一标识符。
userId	Long	举报的发起人用户 ID。
postId	Long	举报的帖子。
reason	String	举报原因内容, 要求不超过 200 字。
createTime	Time	举报的创建时间, 由系统自动记录。
handler	Long	举报处理着管理员的 ID。
status	int	0 为未处理、1 为已仲裁通过举报, 要求被举报方理赔、2 为驳回举报。

表格 38 Report 类属性表

4.2.3.1 类内、类外方法调用状态图

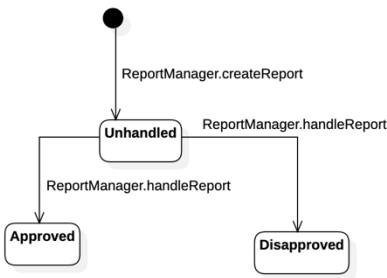


图 7 Report 类方法调用状态图

4.2.3.2 限制条件及出错处理

- 举报原因的 length 不能超过 500 字。所有内容超出限制的部分都会被截断。

4.2.4 ReportManager 类

4.2.4.1 功能

管理举报的类型，通过维持一个 ORM 映射和连接池来控制 Reply 类实例的持久化和获取操作。可以完成帖子发布的处理、获取帖子、获取列表和按照标签来获取列表。

4.2.4.2 性能

允许并发发布、查询回复。

4.2.4.3 输入项

1. 接受举报的内容信息，创建新举报；
2. 来自管理员系统的调用，填写举报被处理信息；
3. 接受一个回复 ID，返回对应回复的内容。

4.2.4.4 输出项

1. 新建举报 ID；
2. 无返回；
3. 一个举报的详细内容。

4.2.4.5 属性描述

属性名	类型	说明
reports	List	系统中当前的所有举报，只要成功发布就会记录在此处。该列表不一定恒在内存中，可以通过 ORM 机制存储、或者从数据库中取回。

表格 39 ReportManager 类属性表

4.2.4.6 方法详细描述

4.2.4.6.1 createReport

方法签名: public Long createReport(Long postId, Long userId, String reason)

功能说明: 创建一个新举报, 返回新举报的 ID。若出错, 则返回 null。

详细说明:

返回类型	Long	
返回值	新举报内容的 ID，若出错，则返回 null。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
postId	Long	举报的帖子的 ID。
userId	Long	发起举报用户的 ID。
reason	String	举报原因。

表格 40 createReport 方法说明

4.2.4.6.2 handleReport

方法签名: public void handleReport(Long reportId, Long handler, int status)

功能说明: 处理一个举报的接口, 管理员处理举报后, 由系统调用该接口完成举报状态的改变。

详细说明:

返回类型	void	
返回值	没有返回值，若用户指定了错误的状态，则会抛出一个异常，在 HTTP 请求中会返回状态码 400。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
reportId	Long	举报内容的 ID。
handler	Long	举报处理者管理员的 ID。
status	int	设定举报处理后的状态。

表格 41 handleReport 方法说明

4.2.4.6.3 getReport

方法签名: public Report getReport(Long id)

功能说明: 通过输入的举报 ID 返回对应的举报。

详细说明:

返回类型	Report	
返回值	ID 对应的举报内容，若举报不存在，则返回 null。	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
id	Long	举报内容 ID。

表格 42 getReport 方法说明

4.2.4.7 限制条件及出错处理

- 管理员或系统指定了错误的举报状态，则会抛出一个异常，在 HTTP 请求中会返回状态码 400。

4.2.5 PostInterface 类

4.2.5.1 功能

完成模块之间的接口调用交互。

4.2.5.2 接口方法详细描述

4.2.5.2.1 createReport

方法签名：public Long createReport(Long postId, Long userId, String reason)

实际调用方法：ReportManager.createReport

4.2.5.2.2 createPost

方法签名：public Boolean createPost(Long userId, String title, String content, List<String> tags)

实际调用方法：PostManager.signup

4.2.5.2.3 createReply

方法签名：public Boolean createReply(Long postId, Long parentId, Long userId, String content)

实际调用方法：ReplyManager.createReply

4.2.5.2.4 `getPostList`

方法签名: `public void getPostList(int page, int size)`

实际调用方法: `PostManager.getPostList`

4.2.5.2.5 `getPost`

方法签名: `public Boolean getPost(Long id)`

实际调用方法: `PostManager.getPost`

4.2.5.2.6 `getReport`

方法签名: `public User getReport(Long id)`

实际调用方法: `ReportManager.getReport`

4.2.5.2.7 `handleReport`

方法签名: `public Boolean handleReport(Long reportId, Long handlerId, int status)`

实际调用方法: `ReportManager.handleReport`

4.2.5.2.8 `getPostsByTag`

方法签名: `public Boolean getPostsByTag(String tag)`

实际调用方法: `PostManager.getPostsByTag`

4.3 房源管理微服务

在这个微服务中,共包含 5 个实体类,分别是 Room 类、Bed 类、Advertise 类、Label 类和 Stay 类,用来记录房源和与房源相关的信息。同时,通过两个控制类 RoomManager 类和 StayManager 类实现对于新增房源、新增房间、查找房源的管理。最终,通过边界类 StayInterface 实现与其他微服务之间的交互。

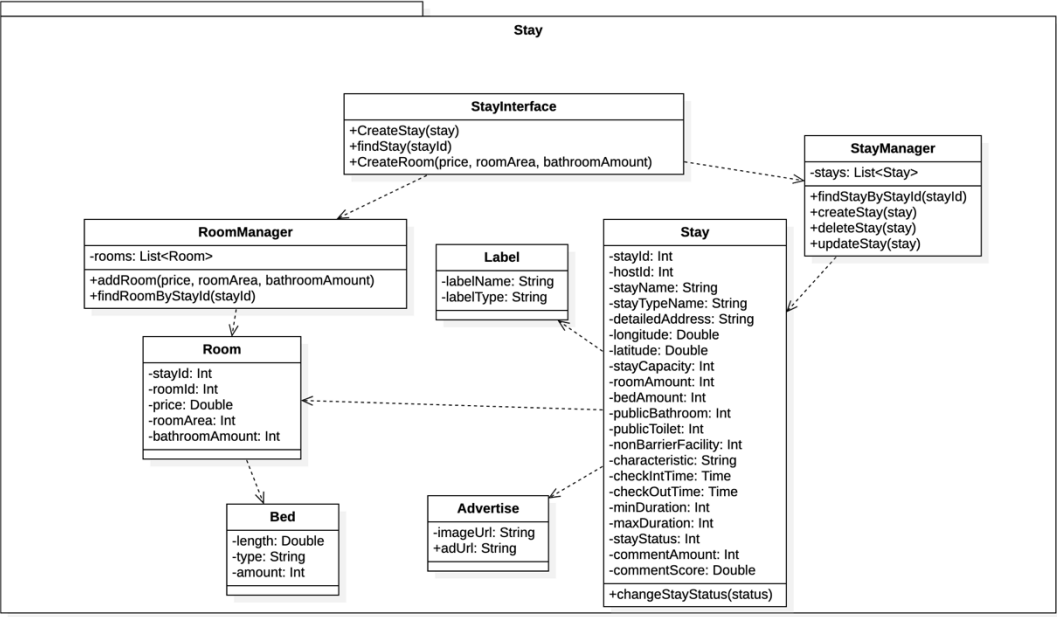


图 8 房源管理微服务详细类设计

4.3.1 Bed 类

4.3.1.1 功能

房源微服务中的 Bed 类记录了房源中床的信息，包括长度、类型和数量等。

4.3.1.2 性能

无约束

4.3.1.3 输入项

该类通过长度、类型和数量进行初始化。

4.3.1.4 输出项

无输出项

4.3.1.5 属性描述

属性名	类型	说明
length	Double	床的长度。接受该参数初始化，之后不会再改变。初始化时会检查长度，确认其大于 0。
type	String	床的类型，指明了该床的特征。例如：单人床、双人床等。
amount	Int	床的数量，指明了该类床的数量。初始化时会检查其值，确定其大于等于 0。

表格 43 Bed 类属性表

4.3.1.6 方法详细介绍

4.3.1.6.1 Bed

方法签名：public Bed(Double length, String type, Int amount)

功能说明：通过参数 length、type 和 amount 进行类的初始构造。

详细说明：

返回类型	无	
返回值	无返回值	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
length	Double	床的长度。
type	String	床的类型。
amount	Int	床的数量。

表格 44 Bed 方法说明

4.3.1.7 限制条件及出错

- 床的长度需要大于 0，否则 Bed 类会抛出 `FormatException` 的自定义异常。
- 床的数量需要大于等于 0，否则 Bed 类会抛出 `FormatException` 的自定义异常。

4.3.2 Room 类

4.3.2.1 功能

房源微服务中的 Room 类记录了房源中的每一个房间，包括对应房源 ID、房间 ID、价格 price、房间大小 roomArea 和浴室数量 bathroomAmount 等。

4.3.2.2 性能

无约束

4.3.2.3 输入项

该类通过房源 ID、房间 ID、价格 price、房间大小 roomArea 和浴室数量 bathroomAmount 进行初始化。

4.3.2.4 输出项

无输出项

4.3.2.5 属性描述

属性名	类型	说明
stayId	Int	房源 id，表示该房间所属的房源信息。
roomId	Int	房间 id，表示该房间是其对应房源的几号房间。
price	Double	价格，表示该房间的售价。初始化时会检查大小，确认其大于 0。
roomArea	Int	房间面积，表示该房间的尺寸。初始化时会检查大小，确认其大于 0。
bathroomAmount	Int	浴室数量，表示该房间中浴室的数量。初始化时会检查大小，确认其大于等于 0。

表格 45 Room 类属性表

4.3.2.6 方法详细介绍

4.3.2.6.1 Room

方法签名: `public Room(int stayId, int roomId, double price, int roomArea, int bathroomAmount)`

功能说明: 通过参数 `stayId`、`roomId`、`price`、`roomArea` 和 `bathroomAmount` 进行类的初始构造。

详细说明:

返回类型	无	
返回值	无返回值	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
stayId	Int	房源 ID。
roomId	Int	房间 ID。
price	Double	房间价格。
roomArea	Int	房间面积大小。
bathroomAmount	Int	浴室数量

表格 46 Room 方法说明

4.3.2.7 限制条件及出错

- 房间面积需要大于 0，否则 Room 类会抛出 `FormatException` 的自定义异常。
- 房间价格需要大于等于 0，否则 Room 类会抛出 `FormatException` 的自定义异常。
- 房源 ID 对应的需要房源需要存在，否则 Room 类会抛出 `ForeignKeyException` 的外键异常。

4.3.3 Label 类

4.3.3.1 功能

房源微服务中的 Label 类记录了房源的标签，包含两个属性标签名 `labelName` 和标签类型 `labelType`。

4.3.3.2 性能

无约束

4.3.3.3 输入项

该类通过标签名 `labelName` 和标签类型 `labelType` 进行初始化。

4.3.3.4 输出项

无输出项

4.3.3.5 属性描述

属性名	类型	说明
<code>labelName</code>	<code>String</code>	标签名称，表示该标签所显示的名称。
<code>labelType</code>	<code>String</code>	标签类型，表示该标签表示的类型。

表格 47 Label 类属性表

4.3.3.6 方法详细介绍

4.3.3.6.1 Label

方法签名：`public Label(String labelName, String labelType)`

功能说明：通过参数 `labelName` 和 `labelType` 进行类的初始构造。

详细说明：

返回类型	无	
返回值	无返回值	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
labelName	String	标签名称。
labelType	String	标签类型。

表格 48 Label 方法说明

4.3.3.7 限制条件及出错

- 标签名称和标签类型需要非空，否则会抛出 `FormatException` 的自定义异常。

4.3.4 Advertise 类

4.3.4.1 功能

房源微服务中的 `Advertise` 类记录了房源所展示广告的信息，包含两个属性广告图片链接 `imageUrl` 和广告链接 `adUrl`。

4.3.4.2 性能

无约束

4.3.4.3 输入项

该类通过广告图片链接 `imageUrl` 和广告链接 `adUrl` 进行初始化。

4.3.4.4 输出项

无输出项

4.3.4.5 属性描述

属性名	类型	说明
<code>imageUrl</code>	<code>String</code>	图片链接，表示该广告所展示图片的链接。
<code>adUrl</code>	<code>String</code>	广告链接，表示点击该广告后跳转到的链接。

表格 49 Advertise 类属性表

4.3.4.6 方法详细介绍

4.3.4.6.1 Advertise

方法签名: public Advertise(String imageUrl, String adUrl)
功能说明: 通过参数 imageUrl 和 adUrl 进行类的初始构造。
详细说明:

返回类型	无	
返回值	无返回值	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
imageUrl	String	图片链接。
adUrl	String	广告链接。

表格 50 Advertise 方法说明

4.3.4.7 限制条件及出错

- 图片链接和广告链接需要非空，否则会抛出 FormatException 的自定义异常。

4.3.5 Stay 类

4.3.5.1 功能

房源微服务中的核心业务类——房源 Stay 类。Stay 类记录了一个房源最基本的信息，包括房源 ID stayId、房东 ID hostId、房源名称 stayName、房源类型名称 stayTypeName、详细地址 detailedAddress、经度 longitude、纬度 latitude、可容纳人数 stayCapacity、房间数量 roomAmount、床的数量 bedAmount、公共浴室数量 publicBathroom、公共卫生间数量 publicToilet、是否有无障碍设施 nonBarrierFacility、房源描述 characteristic、入住时间 checkInTime、离房时间 checkOutTime、最小入住天数 minDuration、最大入住天数 maxDuration、房源状态 stayStatus、评论数量 commentAmount、评论得分 commentScore。

4.3.5.2 性能

无约束

4.3.5.3 输入项

被改变的状态 status。

4.3.5.4 输出项

无输出项

4.3.5.5 属性描述

属性名	类型	说明
stayId	Int	房源 id。
HostId	Int	房东 id，表示房源所属的房东。
stayName	String	房源名称。初始化时会检查其长度，确认其长度大于 0。同时，长度不能超过 10。
stayTypeName	String	房源类型名称。初始化时会检查大小，确认其长度大于 0。同时，长度不能超过 5。
detailedAddress	String	详细地址。初始化时会检查大小，确认其长度大于 0。
longitude	Double	经度。表示该房源所处位置的经度。
latitude	Double	纬度。表示该房源所处位置的纬度。
stayCapacity	Int	房源可以容纳的人数。初始化时会检查值，确认其大于等于 0。
roomAmount	Int	房间数量。初始化时会检查值，确认其大于 0。
bedAmount	Int	床的数量。初始化时会检查值，确认其大于等于 0。
publicBathroom	Int	公共浴室数量。初始化时会检查值，确认其大于等于 0。
publicToilet	Int	公共卫生间数量。初始化时会检查值，确认其大于等于 0。
nonBarrierFacility	Int	是否有无障碍设施。取值为 0 或 1，其中 0 表示没有无障碍设施，1 表示有无障碍设施。
characteristic	String	初始化时会检查大小，确认其长度大于 0。
checkInTime	Time	入住时间。
checkOutTime	Time	退房时间。
minDuration	Int	最小入住时间。初始化时会检查值，确认其大于 0。
maxDuration	Int	最大入住时间。初始化时会检查值，确认其大于最小入住时间。
stayStatus	Int	房源状态，取值为 0~3。0 表示房源待审核，1 表示房源审核未通过，2 表示房源审核通过，3

		表示房源处于封禁状态。
commentAmount	Int	评论数量，表示该房源所拥有的订单的全部评价。
commentScore	Double	评价得分，表示该房源所拥有的订单评价的平均得分。

表格 51 Stay 类属性表

4.3.5.6 方法详细介绍

4.3.5.6.1 changeStayStatus

方法签名：public void changeStayStatus(int status)

功能说明：通过参数 status，将房源的状态设置为新的状态。

详细说明：

返回类型	void	
返回值	无返回值	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
Status	Int	房源新状态。

表格 52 changeStayStatus 方法说明

4.3.5.1 类内、类外方法调用状态图

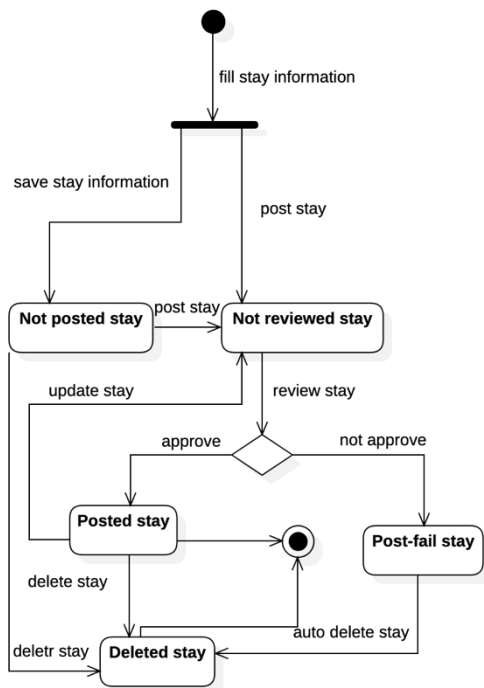


图 9 房源状态图

4.3.5.2 限制条件及出错

- 房东 ID 对应的需要房东需要存在，否则 Stay 类会抛出 `ForeignKeyException` 的外键异常。
- `changeStayStatus` 的参数 `status` 需要大于等于 0 并且小于等于 3，否则 Stay 类会抛出 `FormatException` 的外键异常。

4.3.6 RoomManager 类

4.3.6.1 功能

管理房间的查找和添加，通过内部 rooms 进行操作。

4.3.6.2 性能

无约束

4.3.6.3 输入项

输入房间价格 price、房间面积 roomArea 和浴室数量 bathroomAmount。
输入房源 ID。

4.3.6.4 输出项

输出查找到的房间列表。

4.3.6.5 属性描述

属性名	类型	说明
rooms	List<Stay>	房间列表。

表格 53 RoomManager 类属性表

4.3.6.6 方法详细介绍

4.3.6.6.1 addRoom

方法签名：public Room addRoom(double price, int roomArea, int bathroomAmount)

功能说明：给定参数房间价格 price、房间面积 roomArea 和浴室数量 bathroomAmount，
创建一个新的房间并返回。

详细说明：

返回类型	Room	
返回值	房间	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
price	Double	房间价格。
roomArea	Int	房间面积。
bathroomAmount	Int	浴室数量。

表格 54 addRoom 方法说明

4.3.6.6.2 findRoomByStayId

方法签名: public List<Room> findRoomByStayId(int stayId)

功能说明: 通过参数房源 stay, 查找到该房源下的全部房间。

详细说明:

返回类型	List<Room>	
返回值	房间列表	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
stayId	Int	需要查找的房源 ID。

表格 55 findRoomByStayId 方法说明

4.3.6.6.3 deleteStay

方法签名: public void deleteStay(Stay stay)

功能说明: 通过参数房源 stay, 删除对应的房源。

详细说明:

返回类型	void	
返回值	无返回值	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
stay	Stay	待删除的房源。

表格 56 deleteStay 方法说明

4.3.6.6.4 updateStay

方法签名: public void updateStay(Stay stay)

功能说明: 通过参数房源 stay, 更新对应的房源。

详细说明:

返回类型	void	
返回值	无返回值	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
stay	Stay	待更新的房源。

表格 57 updateStay 方法说明

4.3.6.7 限制条件及出错

- 房源 ID 对应的需要房源需要存在,否则 StayManager 类会抛出 ForeignKeyException 的外键异常。

4.3.7 StayManager 类

4.3.7.1 功能

管理房源的查找、发布、删除和发布，通过内部 stays 进行操作。

4.3.7.2 性能

无约束

4.3.7.3 输入项

输入房源 ID stayId。

输入指定房源 stay。

4.3.7.4 输出项

输出查找到的房源 stay。

4.3.7.5 属性描述

属性名	类型	说明
stays	List<Stay>	房源列表。

表格 58 StayManager 类属性表

4.3.7.6 方法详细介绍

4.3.7.6.1 findStayByStayId

方法签名：public Stay findStayByStayId(int stayId)

功能说明：通过参数房源 id，找到对应的房源并返回。

详细说明：

返回类型	Stay	
返回值	房源	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
stayId	Int	需要查找的房源 ID。

表格 59 findStayByStayId 方法说明

4.3.7.6.2 createStay

方法签名：public void createStay(Stay stay)

功能说明：通过参数房源 stay，创建一个新的房源。

详细说明：

返回类型	void	
返回值	无返回值	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
stay	Stay	待创建的新房源。

表格 60 createStay 方法说明

4.3.7.6.3 deleteStay

方法签名：public void deleteStay(Stay stay)

功能说明：通过参数房源 stay，删除对应的房源。

详细说明：

返回类型	void	
返回值	无返回值	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
stay	Stay	待删除的房源。

表格 61 deleteStay 方法说明

4.3.7.6.4 updateStay

方法签名：public void updateStay(Stay stay)
功能说明：通过参数房源 stay，更新对应的房源。
详细说明：

返回类型	void	
返回值	无返回值	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
stay	Stay	待更新的房源。

表格 62 updateStay 方法说明

4.3.7.7 限制条件及出错

- 房源 ID 对应的需要房源需要存在, 否则 StayManager 类会抛出 ForeignKeyException 的外键异常。

4.3.8 StayInterface 类

4.3.8.1 功能

完成模块之间的接口调用交互。

4.3.8.2 接口方法详细描述

4.3.8.2.1 CreateStay

方法签名： public Stay CreateStay(Stay stay)
实际调用方法： StayManager.createStay

4.3.8.2.2 findStay

方法签名： public Stay findStay(int stayId)
实际调用方法： StayManager.findStayByStayId

4.3.8.2.3 CreateRoom

方法签名： public Room CreateRoom(double price, int roomArea, int bathroomAmount)
实际调用方法： RoomManager.addRoom

4.4 数据统计微服务

在这个微服务中，共包含两个控制类 OrderManager 类和 StayManager 类实现对于订单数据和房源数据的统计。最终，通过边界类 StatisticsInterface 实现与其他微服务之间的交互。

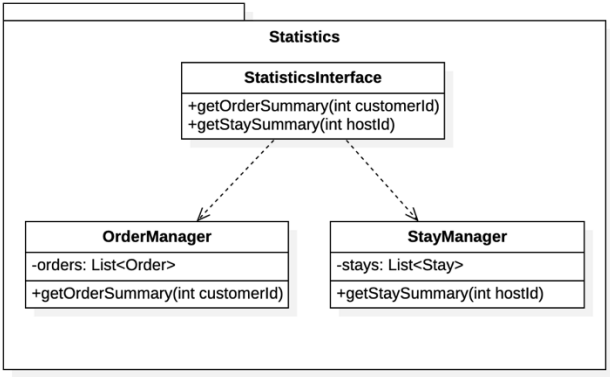


图 10 数据统计微服务详细类设计

4.4.1 StayManager 类

4.4.1.1 功能

管理房源信息，通过房东 ID 来查找与该房东相关的房源统计信息。

4.4.1.2 性能

无约束

4.4.1.3 输入项

输入房东 ID hostId。

4.4.1.4 输出项

输出统计结果 result。

4.4.1.5 属性描述

属性名	类型	说明
stays	List<Stay>	房源列表。

表格 63 StayManager 类属性表

4.4.1.6 方法详细介绍

4.4.1.6.1 getStaySummary

方法签名：public String getStaySummary(int hostId)

功能说明：通过参数房东 id，找到对应的房源的统计信息，并返回统计结果。

详细说明：

返回类型	String	
返回值	统计结果	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
hostId	Int	房东 ID。

表格 64 getStaySummary 方法说明

4.4.1.7 限制条件及出错

- 房东 ID 对应的需要房东需要存在, 否则 StayManager 类会抛出 ForeignKeyException 的外键异常。

4.4.2 OrderManager 类

4.4.2.1 功能

管理订单信息，通过顾客 ID 来查找与该顾客相关的订单统计信息。

4.4.2.2 性能

无约束

4.4.2.3 输入项

输入顾客 ID customerId。

4.4.2.4 输出项

输出统计结果 result。

4.4.2.5 属性描述

属性名	类型	说明
Orders	List<Order>	订单列表。

表格 65 OrderManager 类属性表

4.4.2.6 方法详细介绍

4.4.2.6.1 getOrderSummary

方法签名：public String getOrderSummary(int customerId)

功能说明：通过参数顾客 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	String	
返回值	统计结果	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
customerId	Int	顾客 ID。

表格 66 getOrderSummary 方法说明

4.4.2.7 限制条件及出错

- 顾客 ID 对应的需要顾客需要已经存在，否则 OrderManager 类会抛出

ForeignKeyException 的外键异常。

4.4.3 StatisticsInterface 类

4.4.3.1 功能

完成模块之间的接口调用交互。

4.5 用户信息微服务

该微服务一共包括五个实体类: FavoriteDir, Customer, Host, UserGroup, HostGroup, 用于管理用户信息、用户收藏夹信息、用户组信息。同时, 在该微服务中包含三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:

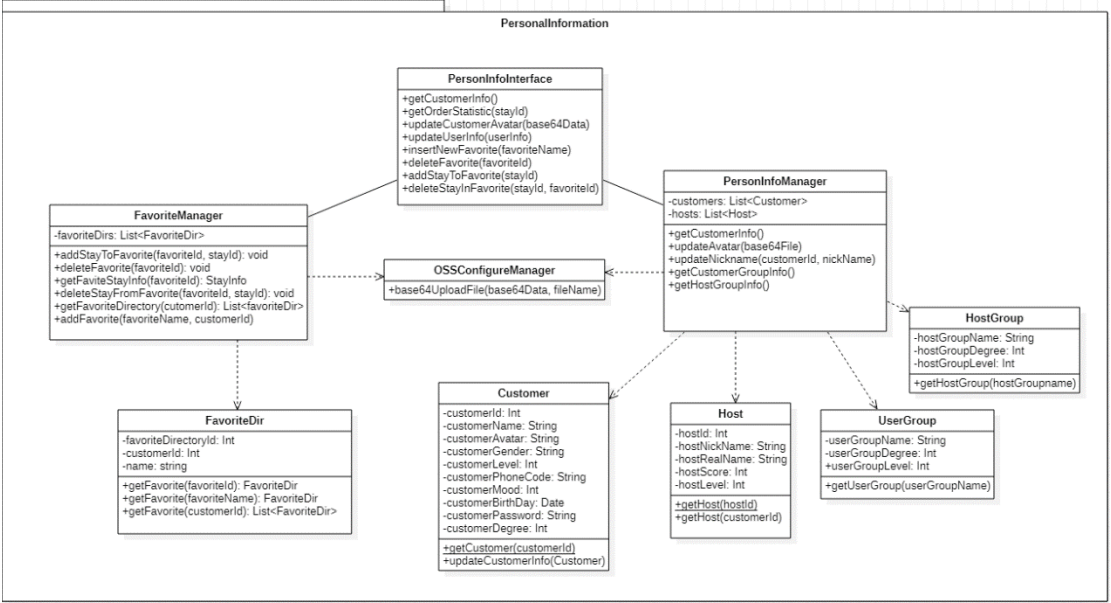


图 11 用户信息微服务详细类设计

接下来将会详细介绍这几个类的描述。

4.5.1 Customer 类

4.5.1.1 功能

用户信息模块中的 Customer 类主要负责管理和维护顾客个人信息，通过类内的方法来实现对用户顾客信息的修改和管理。

4.5.1.2 性能

无约束

4.5.1.3 输入项

查询顾客信息的条件属性，如用户 Id，用户名称。

4.5.1.4 输出项

输出查询的顾客实体。

4.5.1.5 属性描述

属性名	类型	说明
customerId	Int	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。
customerName	String	顾客的昵称，不允许超过 20 个字符。
customerAvatar	String	顾客的头像 URL 字符串。
customerGender	String	顾客的性别，只能取“m”或“f”。
customerLevel	Int	顾客的用户组等级，一个大于零的整数。
customerPhoneCode	Int	顾客的电话号码，必须符合正确的电话号码格式，在初始化顾客时会进行判断。
customerMood	Int	顾客的心情，取值 0-7

customerBirthday	Date	顾客的生日
customerPassword	String	顾客的密码
customerDegree	Int	顾客的积分, 会随着预定房源、发布帖子、评论和评价等操作更新积分

表格 67 Customer 类属性表

4.5.1.6 方法详细描述

4.5.1.6.1 getCustomer

方法签名: public static Customer getCustomer(Int customerId)

功能说明: 通过传入的顾客 id 找到对应的顾客实体

详细说明:

返回类型	Customer	
返回值	查询的顾客实体，若无此顾客则返回空值	
方法限定符	public static	
参数列表（参数名 / 类型 / 说明）		
customerId	Int	待返回的顾客 Id

表格 68 getCustomerId 方法说明

4.5.1.6.2 updateCustomerInfo

方法签名: public void updateCustomerInfo(Customer customer)

功能说明: 更新顾客的基本信息

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）		
customer	Customer	待更新的顾客信息

表格 69 updateCustomerInfo 方法说明

4.5.1.7 限制条件及出错处理

- 部分属性不能作出修改, 如顾客 Id, 顾客等级, 顾客积分。

4.5.2 Host 类

4.5.2.1 功能

用户信息模块中的 Host 类主要负责管理和维护房东个人信息，通过类内的方法来实现对房东顾客信息的修改和管理。

4.5.2.2 性能

无约束

4.5.2.3 输入项

查询房东信息的条件属性，如房东 Id，房东真名。

4.5.2.4 输出项

输出查询的房东实体。

4.5.2.5 属性描述

属性名	类型	说明
hostId	Int	唯一标识房东实体的标识符，在房东实体实例化后不会改变。
hostNickName	String	房东的昵称，不允许超过 20 个字符。
hostRealName	String	房东的真名。
hostScore	Int	房东的房东组积分
hostLevel	Int	房东的房东组等级，为大于零的整数。

表格 70 Host 类属性表

4.5.2.6 方法详细描述

4.5.2.6.1 getHost

方法签名: public static Host getHost(Int hostId)

功能说明: 通过传入的房东 id 找到对应的房东实体

详细说明:

返回类型	Host	
返回值	查询的房东实体，若无此房东则返回空值	
方法限定符	public static	
参数列表（参数名 / 类型 / 说明）		
hostId	Int	待返回的房东 Id

表格 71 getHostId 方法说明

4.5.2.7 限制条件及出错处理

- 部分属性不能作出修改, 如房东 Id, 房东等级, 房东积分。
- 房东的真名需通过调用身份认证系统的验证接口来确定其合法性, 即必须为用户上传的身份证的照片对应的真名, 一旦确定不能修改。

4.5.3 FavoriteDir 类

4.5.3.1 功能

用户信息模块中的 FavoriteDir 类主要负责维护顾客的每一个收藏夹的收藏夹信息。

4.5.3.2 性能

无约束

4.5.3.3 输入项

管理收藏夹信息的各种条件属性, 包括收藏夹 id, 收藏夹名称, 顾客 id

4.5.3.4 输出项

输出查询后的收藏夹实体

4.5.3.5 属性描述

属性名	类型	说明
favoriteDirId	Int	唯一标识收藏夹实体的标识符，在收藏夹实体实例化后不会改变。
customerId	Int	收藏夹所属的顾客的顾客 Id
name	String	收藏夹名称

表格 72 favoriteDir 类属性表

4.5.3.6 方法详细描述

4.5.3.6.1 getFavorite

方法签名：public FavoriteDir getFavorite(Int favoriteId)

功能说明：重载函数，通过传入的收藏夹 id 获取对应的收藏夹实体。

详细说明：

返回类型	FavoriteDir	
返回值	查询的收藏夹实体	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
favoriteId	Int	收藏夹 Id

表格 73 getFavorite 方法说明

4.5.3.6.2 getFavorite(override)

方法签名：public FavoriteDir getFavorite(String favoriteName)

功能说明：重载函数，通过传入的收藏夹名称获取对应的收藏夹实体。

详细说明：

返回类型	FavoriteDir	
返回值	查询的收藏夹实体	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
favoriteName	String	收藏夹名称

表格 74 getFavorite 方法说明

4.5.3.6.3 getFavorite(override)

方法签名：public List<FavoriteDir> getFavorite(Int customerId)
功能说明：重载函数，通过传入的顾客 id 获取对应的收藏夹实体。
详细说明：

返回类型	FavoriteDir	
返回值	查询的收藏夹实体	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
favoriteId	Int	顾客 id

表格 75 getFavorite 方法说明

4.5.4 UserGroup 类

4.5.4.1 功能

个人信息模块中的 UserGroup 类主要负责管理和维护系统内顾客的用户组等级信息。用户组等级就是当前顾客的等级信息，积分信息。在用户积分到达一定水平时可以升级。

4.5.4.2 性能

无约束

4.5.4.3 输入项

由于该类仅涉及查询某个顾客的用户组信息，因此输入项为某用户组名称。

4.5.4.4 输出项

获取用户组信息时，输出对应的用户组实体。

4.5.4.5 属性描述

属性名	类型	说明
userGroupName	String	用户组的用户组名称，即该用户等级名称
userGroupDegree	Int	用户的用户组等级积分。只有用户积分超过该值时才能达到该用户组等级
userGroupLevel	Int	对应的用户组等级数

表格 76 UserGroup 类属性表

4.5.4.6 方法详细描述

4.5.4.6.1 getUserGroup

方法签名：public UserGroup getUserGroup(String userGroupName)

功能说明：通过用户组名称获取对应的用户组实体

详细说明：

返回类型	UserGroup	
返回值	对应的用户组实体	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
userGroupName	String	用户组名称

表格 77 getUserGroup 方法说明

4.5.4.7 限制条件及出错处理

- 在正常情况下，该类的所有属性均不能被改变，且用户的用户积分均只能在该类的积分属性范围内变动，且当用户的用户组积分到达对应的阈值时，用户组等级自动升级。

4.5.5 HostGroup 类

4.5.5.1 功能

个人信息模块中的 HostGroup 类主要负责管理和维护系统内房东的房东组等级信息。房东组等级就是当前房东的等级信息，积分信息。在房东积分到达一定水平时可以升级。

4.5.5.2 性能

无约束

4.5.5.3 输入项

由于该类仅涉及查询某个房东房东组信息，因此输入项为某房东组名称。

4.5.5.4 输出项

获取房东组信息时，输出对应的房东组实体。

4.5.5.5 属性描述

属性名	类型	说明
hostGroupName	String	房东组的房东组名称，即该房东等级名称
hostGroupDegree	Int	房东的房东组等级积分。只有房东积分超过该值时才能达到该房东组等级
hostGroupLevel	Int	对应的房东组等级数

表格 78 HostGroup 类属性表

4.5.5.6 方法详细描述

4.5.5.6.1 getHostGroup

方法签名：public HostGroup getHostGroup(String hostGroupName)

功能说明：通过用户组名称获取对应的用户组实体

详细说明：

返回类型	HostGroup	
返回值	对应的用户组实体	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
hostGroupName	String	房东组名称

表格 79 getHostGroup 方法说明

4.5.5.7 限制条件及出错处理

- 在正常情况下，该类的所有属性均不能被改变，且房东的房东积分均只能在该类的积分属性范围内变动，且当房东的房东组积分到达对应的阈值时，房东组等级自动升级。

4.5.6 FavoriteManager 类

4.5.6.1 功能

个人模块中的 FavoriteManager 类主要负责完成子系统内与收藏夹创建、删除、修改、和添加房源到收藏夹等功能的业务逻辑，完成边界类相关接口的内部逻辑。

4.5.6.2 性能

无约束

4.5.6.3 输入项

- 在添加新房源到收藏夹或从某个收藏夹删除某个收藏的房源时，输入项为收藏夹 Id 和房源 Id 信息
- 在删除某一收藏夹和获得收藏夹内的房源信息时，输入项为收藏夹 id
- 在获得某位顾客的所有收藏夹时，输入项为顾客 Id
- 在创建新的收藏夹时，需要输入收藏夹的名称和顾客 Id

4.5.6.4 输出项

- 在删除某一收藏夹和获得收藏夹内的房源信息时，输出项为房源基本信息，包括房源的照片 URL，房源名称。
- 在获得某位顾客的所有收藏夹时，输出项为一个收藏夹实体的列表。

4.5.6.5 属性描述

属性名	类型	说明
favoriteDirs	List	维护的系统内各顾客所有的收藏夹列表，即数据库内的信息。

表格 80 FavoriteManager 类属性表

4.5.6.6 方法详细描述

4.5.6.6.1 addStayToFavorite

方法签名：public void addStayToFavorite(Int favoriteId,Int stayId)

功能说明：通过指定的收藏夹 id 和房源 id 添加对应的房源到收藏夹中

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）		
favoriteId	Int	要加入房源的收藏夹 Id
stayId	Int	要加入收藏夹的房源 Id

表格 81 addStayToFavorite 方法说明

4.5.6.6.2 deleteFavorite

方法签名：public void deleteFavorite(favoriteId)

功能说明：删除指定收藏夹 id 的收藏夹

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）	

favoriteId	Int	要删除的收藏夹 id
------------	-----	------------

表格 82 deleteFavorite 方法说明

4.5.6.6.3 deleteStayFromFavorite

方法签名：public void deleteStayFromFavorite(Int favoriteId, Int stayId)

功能说明：删除指定收藏夹内的指定房源

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）		
favoriteId	Int	指定的收藏夹 id
stayId	Int	要删除的房源 Id

表格 83 deleteStayFromFavorite 方法说明

4.5.6.6.4 getFavoriteDirectory

方法签名：public List<FavoriteDir> getFavoriteDirectory(Int customerId)

功能说明：获取某一顾客的所有收藏夹实体列表

详细说明：

返回类型	List<FavoriteDir>	
返回值	查询得到的收藏夹实体列表	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
customerId	Int	顾客 Id

表格 84 getFavoriteDirectory 方法说明

4.5.6.6.5 getFavoriteStayInfo

方法签名：public List<StayInfo> getFavoriteStayInfo(Int favoriteId)

功能说明：获取指定收藏夹 Id 的所有房源基本信息

详细说明：

返回类型	List<StayInfo>	
返回值	查询得到的房源基本信息列表	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
favoriteId	Int	带查询的收藏夹 Id

表格 85 getFavoriteStayInfo 方法说明

4.5.6.6.6 addFavorite

方法签名: public void addFavorite(String favoriteName, Inr customerId)

功能说明: 根据指定的收藏夹名称和顾客 Id 添加新的收藏夹

详细说明:

返回类型	void	
返回值		
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
favoriteName	String	待插入的收藏夹的收藏夹名称
customerId	Int	指定的顾客 Id

表格 86 addFavorite 方法说明

4.5.7 PersonInfoManager 类

4.5.7.1 功能

个人模块中的 PersonInfoManager 类主要负责管理与顾客信息和房东信息有关的内部业务逻辑, 包括修改个人昵称, 头像等信息, 获取基本的个人信息, 用户组信息等。

4.5.7.2 性能

无约束

4.5.7.3 输入项

- 在获得顾客或房东的基本信息时, 输入项为从 token 信息中获取的顾客 id 或房东 Id
- 在更新用户头像时, 输入项为要更新的图片的 base64 格式的编码字符串
- 在更新用户的基本信息时, 输入项为从 token 信息中获取的顾客 id 和要更新的用户基本信息, 如顾客昵称, 顾客生日等。

4.5.7.4 输出项

- 在获得顾客或房东的基本信息时，输出项为房东或顾客的基本信息。
- 在获得用户组信息时，输出项为用户组的基本信息。

4.5.7.5 属性描述

属性名	类型	说明
customers	List<customer>	维护的顾客实体类的列表
hosts	List<Host>	维护的房东实体类的列表

表格 87 PersonInfoManager 类属性表

4.5.7.6 方法详细描述

4.5.7.6.1 getCustomerInfo

方法签名：public HashMap<String,Object> getCustomerInfo()

功能说明：通过从 token 信息中获取的 customerId 获取对应的顾客个人基本信息

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	HashMap<String,Object>
返回值	用户基本信息的条目集合
方法限定符	public

表格 88 checkPassword 方法说明

4.5.7.6.2 updateAvatar

方法签名：public Boolean updateAvatar(String base64File)

功能说明：更新当前登录用户的头像图片

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 OSSConfigureManager 控制类的 base64UploadFile()将 base64Data 转化为 Url

详细说明：

返回类型	Boolean
------	---------

返回值	图片是否上传成功，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）		
String base64File	String	要更新的图片的 base64 字符串

表格 89 updateAvatar 方法说明

4.5.7.6.3 updateUserInfo

方法签名：public void updateUserInfo(CustomerInfo customerInfo)

功能说明：更新当前用户的基本信息

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

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）		
customerInfo	CustomerInfo	前端所需的需要更改的用户基本信息的 Dto 类，包括用户昵称，用户生日，用户性别的基本信息

表格 90 updateUserInfo 方法说明

4.5.7.6.4 getCustomerGroupInfo

方法签名：public CustomerGroup getCustomerGroupInfo()

功能说明：获取当前登录用户的用户组信息

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

详细说明：

返回类型	CustomerGroup	
返回值	顾客用户组实体类	
方法限定符	public	

表格 91 getCustomerGroupInfo 方法说明

4.5.7.6.5 getHostGroupInfo

方法签名：public HostGroup getHostGroupInfo()

功能说明：更新当前房东的基本信息

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

详细说明：

返回类型	HostGroup
返回值	房东组信息
方法限定符	public

表格 92 updateHostInfo 方法说明

4.5.7.7 限制条件及出错处理

- 当更新用户基本信息发现格式错误导致程序失败时，会返回一个错误码 417 并抛出对应的 `FormatException` 的自定义异常。

4.5.8 OSSConfigureManager 类

4.5.8.1 功能

个人模块中的 `OSSConfigureManager` 类主要负责完成图片修改或上传时将上传图片的 base64 格式的字符串上传至远端 OSS，处理为可直接存储到数据库的图片 URL 字符串。

4.5.8.2 性能

无约束

4.5.8.3 输入项

- 要处理的图片的 base64 格式的字符串

4.5.8.4 输出项

- 输出上传至远端 OSS 的图片的 URL 字符串

4.5.8.5 方法详细描述

4.5.8.5.1 base64UploadFile

方法签名：public String base64UploadFile(String base64Data, String fileName)

功能说明：将输入的 base64 格式的图片上传至服务器并返回图片的 URL

详细说明：

返回类型	String	
返回值	上传至远端 OSS 的图片 URL	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
base64Data	String	需要上传的图片的 base64 格式的字符串
fileName	String	图片名称，唯一标识该图片

表格 93 base64UploadFile 方法说明

4.5.8.6 限制条件及出错处理

- 输入的图片的 base64 格式的字符串开头必须以“data:”开头并且后跟该图片的类型，否则会抛出图片格式不正确的异常。
- 图片的名称必须唯一标识该图片，以免覆盖上传至远端的其他图片
- 对于错误格式的 base64 字符串，系统会抛出 WrongFormatException 异常并将错误原因作为输出字符串输出到数据库中

4.5.9 PersonInfoInterface 类

4.5.9.1 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 webd 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、上传个人头像、创建新的收藏夹、更新和管理收藏夹等功能。

4.5.9.2 性能

每个接口的处理响应时间需在 1000ms 以内。

4.5.9.3 输入项

- 若为处理与收藏夹信息有关的逻辑，则输入项包括收藏夹 id，收藏夹名称，房源 id。
- 若为处理与用户个人信息维护有关的逻辑，则输入项包括顾客 id，房东 id，需要更新的用户基本信息等。

4.5.9.4 输出项

- 若为处理与收藏夹信息有关的逻辑，则输出项为更新后的收藏夹基本信息，收藏夹内的房源基本信息
- 若为处理与个人信息维护有关逻辑，则输出项包括顾客和房东更新后的基本信息

4.5.9.5 接口描述

4.5.9.5.1 getCustomerInfo

方法签名：public CustomerInfo getCustomerInfo()

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 getCustomerInfo 方法

4.5.9.5.2 getOrderStatistic

方法签名：public OrderData getOrderStatistic(stayId)

功能说明：获取房东某个房源的订单数据，用于前端的展示，返回的是前端需要展示的数据类型的 DTO 类

接口调用：

- 需要调用 Statistic 微服务的 getOrderSummary 函数

4.5.9.5.3 updateCustomerAvatar

方法签名：public ResponseEntity updateCustomerAvatar(String base64Data)

功能说明：更新顾客的头像，返回错误码以方便下游处理

接口调用：

- 需要调用系统内 PersonInfoManager 的 updateAvatar 方法

4.5.9.5.4 updateUserInfo

方法签名: `public boolean updateUserInfo(userInfo)`

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 `updateUserInfo` 方法

4.5.9.5.5 getFavoriteDirectory

方法签名: `public FavoriteDirectory getFavoriteDirectory()`

功能说明: 获取当前登录的用户的收藏夹列表

接口调用:

- 需要调用系统内的 `FavoriteManager` 的 `getFavoriteDirectory` 方法

4.5.9.5.6 insertNewFavorite

方法签名: `public boolean insertNewFavorite(String favoriteName)`

功能说明: 根据收藏夹名称在当前登录用户下创建一个新的收藏夹

接口调用:

- 需要调用系统内的 `FavoriteManager` 的 `addFavorite` 方法

4.5.9.5.7 deleteFavorite

方法签名: `public boolean deleteFavorite(Int favoriteId)`

功能说明: 删除指定的收藏夹

接口调用:

- 需要调用系统内的 `FavoriteManager` 类的 `deleteFavorite` 方法

4.5.9.5.8 addStayToFavorite

方法签名: `public boolean addStaytoFavorite(stayId)`

功能说明: 获取房东某个房源的订单数据, 用于前端的展示, 返回的是前端需要展示的数据类型的 DTO 类

接口调用:

- 需要调用系统内的 `addStayToFavorite` 方法

4.6 管理微服务

该微服务一共包括两个实体类：Report,UnverifiedStay 用于管理非法帖子用户的举报、房源的审核等。同时，在该微服务中包含两个控制类完成对用户举报信息的审核和回复、待审核房源信息的处理，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：

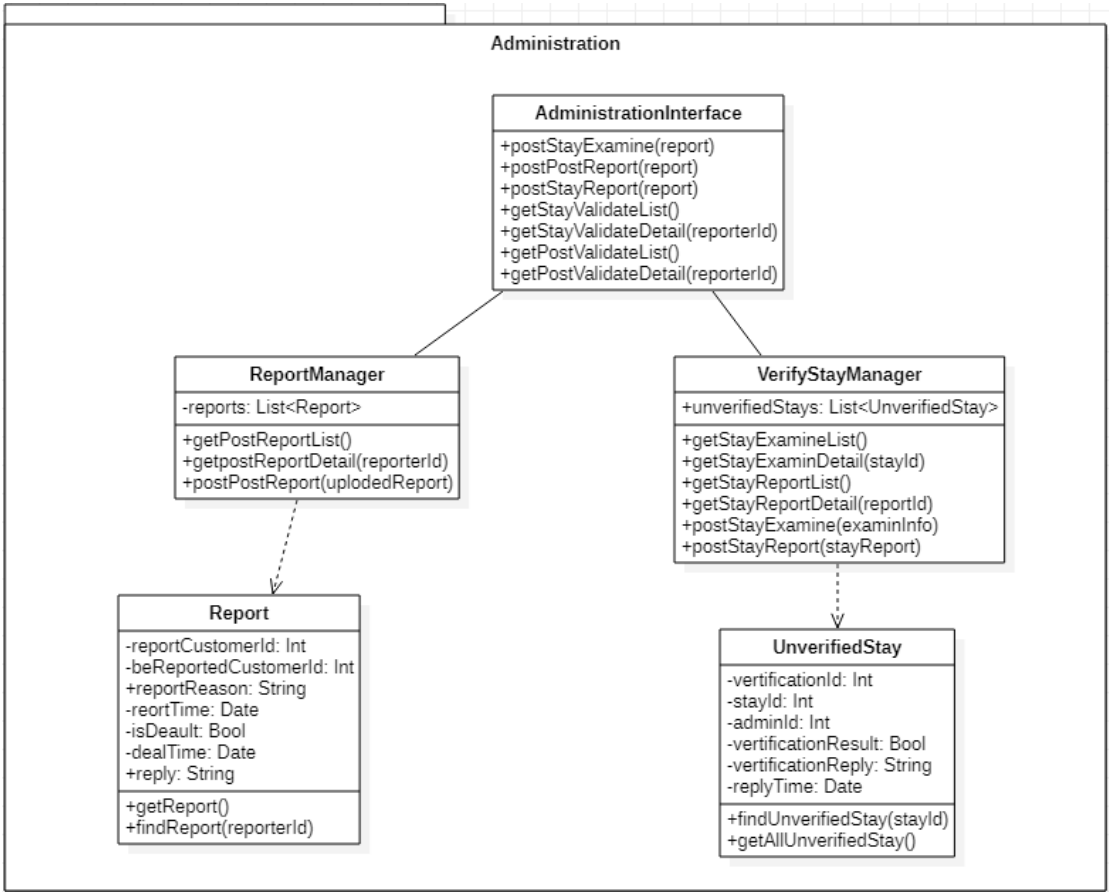


图 12 管理微服务详细类设计

接下来将会详细介绍这几个类的描述。

4.6.1 Report 类

4.6.1.1 功能

管理模块中的 Report 类主要负责管理和维护用户举报的信息，完成举报信息的处理和回复。

4.6.1.2 性能

保证每个方法的执行时间在 100ms 以内。

4.6.1.3 输入项

查询用户举报信息所需的具体的举报信息 id

4.6.1.4 输出项

输出查询到的举报信息或举报信息列表。

4.6.1.5 属性描述

属性名	类型	说明
reportCustomerId	Int	举报信息举报方的用户 id
beReportedCustomerId	Int	举报信息被举报方的用户 id
reportReason	String	举报信息中的举报原因
reportTime	Date	举报时间
isDeault	Boolean	该举报信息当前是否已经处理
dealTime	Date	该举报信息处理的时间，若还未处理则为空值
reply	String	举报信息的回复内容

表格 94 Report 类属性表

4.6.1.6 方法详细描述

4.6.1.6.1 getReport

方法签名：public static List<Report> getReport()

功能说明：获取当前所有的举报信息列表

详细说明：

返回类型	List<Report>
返回值	当前所有待处理的举报信息列表
方法限定符	public static

表格 95 getReport 方法说明

4.6.1.6.2 findReport

方法签名：public Report findReport(Int reporterId, Int beReportedId)

功能说明:根据举报人和被举报人 id 查询某一个举报信息

详细说明：

返回类型	Report	
返回值	举报信息	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
reportId	Int	举报方的用户 id
beReportedId	Int	被举报方的用户 id

表格 96 findReport 方法说明

4.6.2 UnverifiedStay 类

4.6.2.1 功能

管理模块中的 Unverified 类主要负责维护未审核的房源信息。

4.6.2.2 性能

保证每个方法的执行时间在 100ms 内。

4.6.2.3 输入项

查询未审核房源所需的房源 id

4.6.2.4 输出项

输出查询到的待审核信息或待举报信息列表。

4.6.2.5 属性描述

属性名	类型	说明
verificationId	Int	审核信息的 id
stayId	Int	待审核的房源 id
adminId	Int	处理该信息的管理员 id
verificationResult	Bool	该房源是否审核通过
verificationReply	String	审核回复
replyTime	Date	审核回复时间

表格 97 UnverifiedStay 类属性表

4.6.2.6 方法详细描述

4.6.2.6.1 findUnverifiedStay

方法签名: public static UnverifiedStay findUnverifiedStay(Int stayId)

功能说明: 获取对应房源 id 的待审核信息

详细说明:

返回类型	UnverifiedStay
返回值	待处理的审核信息
方法限定符	public static

表格 98 findUnverifiedStay 方法说明

4.6.2.6.2 getAllUnverifiedStay

方法签名: public static getAllUnverifiedStay()

功能说明: 获取当前待处理的所有未审核房源的审核信息

详细说明:

返回类型	Report	
返回值	举报信息	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
reportId	Int	举报方的用户 id
beReportedId	Int	被举报方的用户 id

表格 99 findReport 方法说明

4.6.2.7 类内类外方法调用状态图

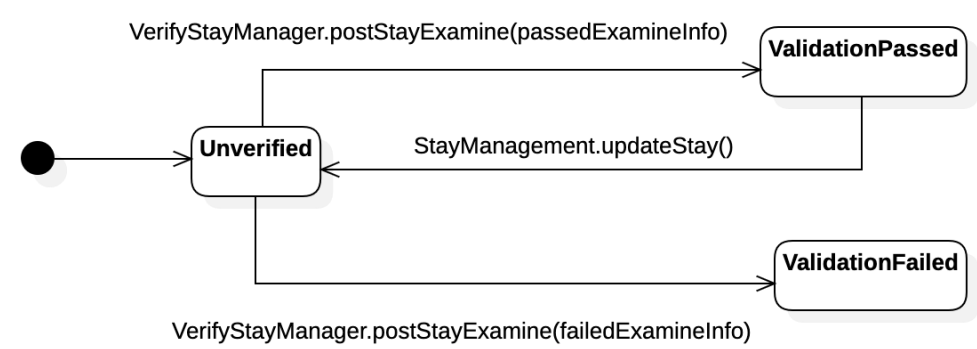


图 13 UnverifiedStay 类方法调用状态图

4.6.3 ReportManager 类

4.6.3.1 功能

管理模块中的 ReportManager 负责管理和维护举报信息。

4.6.3.2 性能

保证每个方法的执行时间在 100ms 内。

4.6.3.3 输入项

获取待处理的举报信息时，输入项为举报信息 Id；更新举报信息时，输入项为更新后

的举报信息 Report

4.6.3.4 输出项

输出项为举报信息实体

4.6.3.5 属性描述

属性名	类型	说明
reports	List<Report>	维护的举报信息的列表

表格 100 ReportManager 类属性表

4.6.3.6 方法详细描述

4.6.3.6.1 getPostReportDetail

方法签名: public Report getPostReportDetail(Int reportedId, Int beReportedId)

功能说明: 获取某一举报信息的详细内容

详细说明:

返回类型	Report	
返回值	举报信息	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
reportedId	Int	举报方的用户 id
beReportedId	Int	被举报方的用户 id

表格 101 getPostReportDetail 方法说明

4.6.3.6.2 postPostReport

方法签名: public void postPostReport(uploadedReport)

功能说明: 上传处理好的举报信息

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）		
uploadedReport	Report	要更新的举报信息

4.6.4 AdministrationInterface 类

4.6.4.1 功能

管理模块中的 AdministrationInterface 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房源信息和房源举报信息及用户举报信息，修改相应的信息并上传。

4.6.4.2 性能

保证每个方法的执行时间在 1000ms 内。

4.6.4.3 输入项

处理相关的审核信息和举报信息时，需要输入房源 id，举报 id,审核信息实体和举报信息实体。

4.6.4.4 输出项

查询相关的房源审核信息和举报信息时，输出相应的房源审核信息和举报信息。

4.6.4.5 方法详细描述

4.6.4.5.1 postStayExamine

方法签名：public Boolean postStayExamine(UnverifiedStay report)

功能说明：更新管理员完成审核的房源待审核审核内容

接口调用：

- 需要调用系统内的 VerifyStayManager 的 postStayExamine 方法

4.6.4.5.2 postPostReport

方法签名: `public Boolean postPostReport(Report report)`

功能说明: 更新管理员完成的帖子举报处理信息

接口调用:

- 需要调用系统内的 ReportManager 的 postPostReport 方法

4.6.4.5.3 postStayReport

方法签名: `public Boolean postStayReport(Report report)`

功能说明: 更新管理员完成审核房源举报信息内容

接口调用:

- 需要调用系统内的 VerifyStayManager 的 postStayReport 方法

4.6.4.5.4 getStayValidateList

方法签名: `public List<UnverifiedStay> getStayValidateList()`

功能说明: 获取管理员当前待处理的所有待审核房源信息列表

接口调用:

- 需要调用系统内的 VerifyStayManager 的 getStayExamineList() 方法

4.7 促销微服务

该微服务一共包括两个实体类: Coupon 和 CouponType 用于管理每个用户拥有的礼券, 在不同订单的付款阶段可以使用的礼券, 以及管理和维护系统的所有礼券类型。同时, 在该微服务中包含两个控制类完成对用户礼券信息的维护, 使用和更新及对礼券类型的增加和删除, 同时使用边界类 SaleInterface 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:

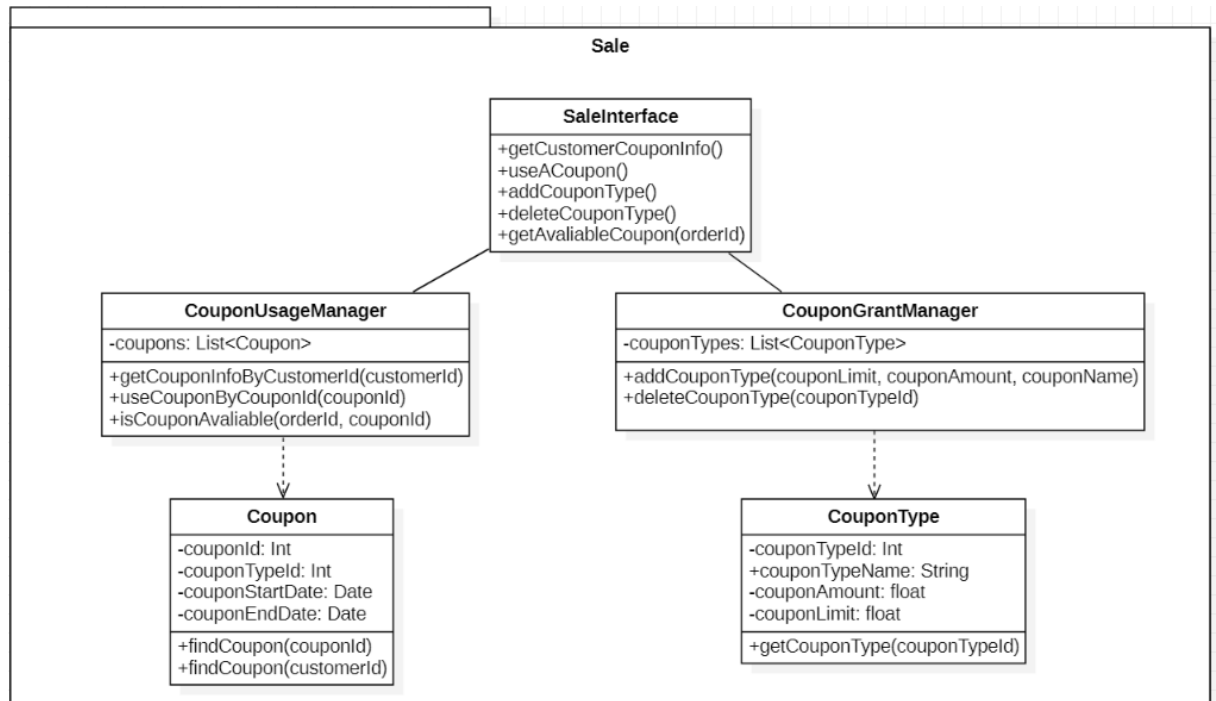


图 14 促销微服务详细类设计

接下来将会详细介绍这几个类的描述。

4.7.1 Coupon 类

4.7.1.1 功能

促销模块中的 **Coupon** 类主要维护了某个特定的顾客拥有的礼券的基本信息。

4.7.1.2 性能

无性能要求。

4.7.1.3 输入项

对于查询获取某个顾客的礼券而言，输入项为礼券 id 和顾客 id。

4.7.1.4 输出项

输出查询到的礼券信息或礼券信息列表。

4.7.1.5 属性描述

属性名	类型	说明
couponId	Int	礼券 id
couponTypeId	Int	礼券类型 id
couponStartDate	Date	礼券可用的起始日期
couponEndDate	Date	礼券可用的结束日期

表格 103 Coupon 类属性表

4.7.1.6 方法详细描述

4.7.1.6.1 findCoupon

方法签名: public static Coupon findCoupon(Int couponId)

功能说明: 获取指定礼券 id 的礼券实体

详细说明:

返回类型	Coupon
返回值	查询到的礼券信息实体
方法限定符	public static

表格 104 getReport 方法说明

4.7.1.6.2 findCoupon(override)

方法签名: public static List<Coupon> findCoupon(Int customerId)

功能说明: 查询指定顾客的所有礼券信息的礼券实体列表。

详细说明:

返回类型	List<Coupon>	
返回值	礼券信息列表	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
customerId	Int	顾客 id

表格 105 findCoupon(override) 方法说明

4.7.1.6.3 deleteCoupon

方法签名：public static void deleteCoupon(Int couponId)

功能说明：删除指定礼券 id 的礼券

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）		
couponId	Int	礼券 id

4.7.1.7 类内类外调用状态图

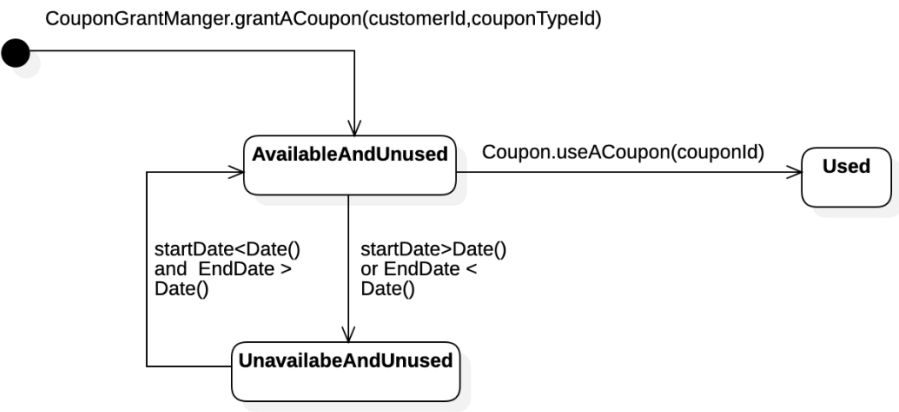


图 15 Coupon 类方法调用状态图

4.7.1.8 限制条件及出错处理

- 礼券的可用起始日期和结束日期均需满足日期格式，且结束日期一定晚于礼券的可用开始日期。

4.7.2 CouponType 类

4.7.2.1 功能

促销模块中的 CouponType 主要维护了某个特定的礼券类型信息。

4.7.2.2 性能

无性能要求。

4.7.2.3 输入项

对于新增和删除某个礼券信息而言，输入项为礼券类型 id。

4.7.2.4 输出项

无输出项。

4.7.2.5 属性描述

属性名	类型	说明
couponTypeId	Int	礼券类型 id
couponTypeName	String	礼券类型名称
couponAmount	float	礼券金额
couponLimit	float	礼券最低使用金额

表格 106 CouponType 类属性表

4.7.2.6 方法详细描述

4.7.2.6.1 addCouponType

方法签名：public static void addCouponType(CouponType couponType)

功能说明：增加新的礼券类型

详细说明：

返回类型	void
返回值	无
方法限定符	public static
参数	couponType

表格 107 addCouponType 方法说明

4.7.2.6.2 deleteCouponType

方法签名：public static void deleteCouponType(Int couponTypeId)

功能说明：删除指定的礼券类型

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）		
couponTypeId	Int	礼券 id

表格 108 deleteCouponType 方法说明

4.7.2.7 限制条件及出错处理

- 礼券类型的金额必须小于最低可用金额，且均大于 0.

4.7.3 CouponUsageManager 类

4.7.3.1 功能

促销模块中的 CouponUsageManager 主要负责处理与顾客使用礼券和获取顾客礼券信息的业务逻辑。

4.7.3.2 性能

每个方法的响应时间为 50ms 内

4.7.3.3 输入项

- 获取某位顾客的礼券信息时，输入为顾客 id
- 使用某个礼券时，输入为礼券 id
- 判断某个礼券是否可用时，输入为礼券 id 和订单 id

4.7.3.4 输出项

当获取某位顾客的礼券信息列表

4.7.3.5 属性描述

属性名	类型	说明
coupons	List<Coupon>	礼券列表

表格 109 CouponUsageMangager 类属性表

4.7.3.6 方法详细描述

4.7.3.6.1 getCouponInfoByCustomerId

方法签名：public List<Coupon> getCouponInfoByCustomerId(Int customerId)

功能说明：获取已登录的用户的礼券信息列表

详细说明：

返回类型	List<Coupon>
返回值	礼券信息列表
方法限定符	public
参数	用户 id

表格 110 getCouponInfoByCustomerId 方法说明

4.7.3.6.2 useCouponByCouponId

方法签名：public float useCouponByCouponId(Int couponId)

功能说明：使用指定的礼券

详细说明：

返回类型	float
------	-------

返回值	礼券金额	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
couponId	Int	礼券 id

表格 111 useCouponByCouponId 方法说明

4.7.4 CouponGrantManager 类

4.7.4.1 功能

促销模块中的 CouponGrantManager 主要负责管理和维护系统的所有礼券类型。可以增加或者删除礼券类型，同时也负责管理礼券的定期分发相关的内部逻辑处理。

4.7.4.2 性能

每个方法的响应时间为 50ms 内

4.7.4.3 输入项

- 新增一个礼券类型时，输入项为礼券金额，礼券名称，礼券的最低使用金额
- 删除某个礼券类型时，输入为礼券类型 id
- 为某位顾客分发礼券时，输入为顾客 id 和礼券类型 id

4.7.4.4 输出项

无

4.7.4.5 属性描述

属性名	类型	说明
couponTypes	List<CouponType>	礼券类型列表

表格 112 CouponGrantMangager 类属性表

4.7.4.6 方法详细描述

4.7.4.6.1 addCouponType

方法签名：public void addCouponType(float couponLimit, float couponAmount, String couponName)

功能说明：增加某一个礼券类型

接口调用：

系统内 CouponType 的 addCouponType 方法

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）		
couponLimit	float	礼券最低使用金额
couponAmount	float	礼券金额
couponName	String	礼券名称

表格 113 addCouponType 方法说明

4.7.4.6.2 grantACoupon

方法签名：public void grantACoupon(Int customerId, Int couponTypeId)

功能说明：为指定的顾客分发一个礼券

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）		
customerId	Int	顾客 id
couponTypeId	Int	礼券类型 id

表格 114 grantACoupon 方法说明

4.7.5 SaleInterface 类

4.7.5.1 功能

促销模块中的 SaleInterface 类为其他微服务提供可供调用的接口和外部系统（如 web 端）的调用接口，包含获取顾客礼券信息的功能，使用礼券，管理礼券信息和获取某个订单可用礼券的功能。

4.7.5.2 性能

每个方法的响应时间为 500ms 内

4.7.5.3 输入项

- 获取某个订单的可用礼券信息列表时，输入项为订单 id

4.7.5.4 输出项

礼券信息列表

4.7.5.5 方法详细描述

4.7.5.5.1 getCustomerCouponInfo

方法签名：public List<Coupon> getCustomerCouponInfo()

功能说明：获取当前顾客的礼券信息列表

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
- 调用系统内 CouponUsageManager 中的 getCouponInfoByCustomerId 方法

4.7.5.5.2 useACoupon

方法签名：public float useACoupon(Int couponId)

功能说明：使用指定的礼券

接口调用：

- 调用系统内 CouponUsageManager 中的 useAcouponByCouponId 的方法

4.7.5.5.3 addCouponType

方法签名：public void addCouponType(CouponType couponType)

功能说明：添加一个新的礼券类型

接口调用：

- 调用系统内 CouponGrantManager 中的 addCouponType 的方法

4.7.5.5.4 getAvailableCoupon

方法签名：public List<Coupon> getAvailable(Int orderId)

功能说明：获取当前订单 id 可用的礼券类型

接口调用：

- 调用系统内 CouponUsageManager 中的 getCouponInfoByCustomerId 的方法
- 调用系统内 CouponUsageManager 中的 isCouponAvailable 的方法

4.7.5.6 接口方法详细描述

4.7.5.6.1 getOrderSummary

方法签名：public String getOrderSummary(int customerId)

实际调用方法：OrderManager.getOrderSummary

4.7.5.6.2 getStaySummary

方法签名：public String getStaySummary(int hostId)

实际调用方法：StayManager.getStaySummary

4.8 订单微服务

订单微服务主要负责订单、订单评论以及订单举报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 HostComment 类。除此之外，负责订单举报信息为 Report 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：

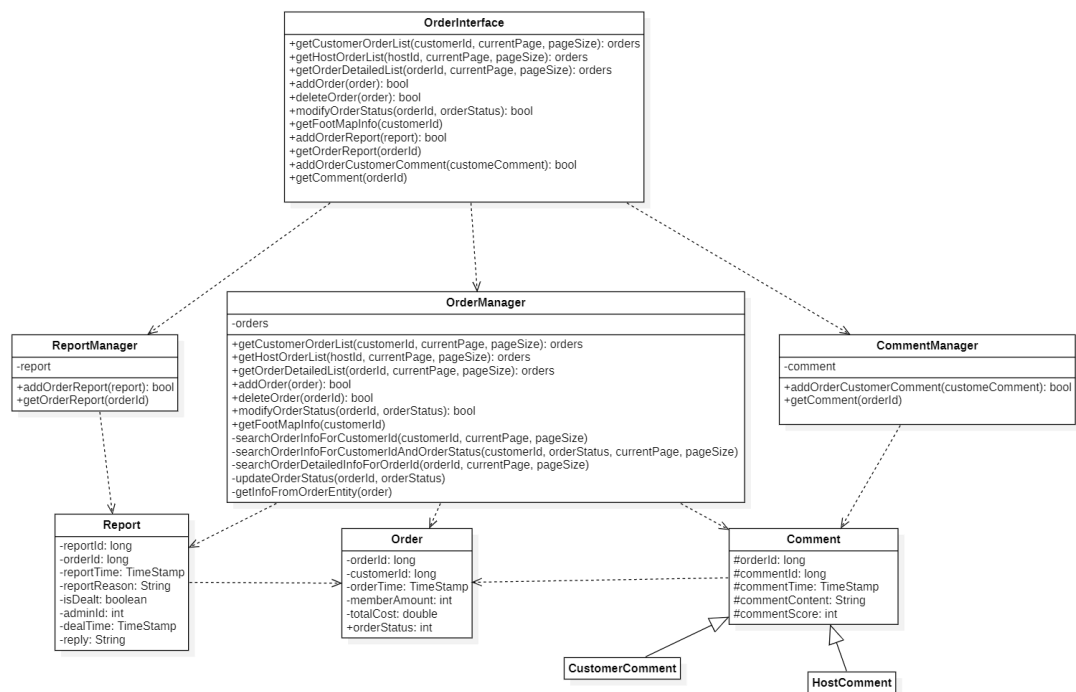


图 16 订单微服务详细类设计

接下来将会详细介绍这几个类的描述。

4.8.1 Order 类

4.8.1.1 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。

4.8.1.2 性能

无约束

4.8.1.3 输入项

1. 输入详细的订单下单时间、下单用户标识符、订单成员数量、订单总价格、订单状态

完成对订单的初始化操作。

- 2. 输入订单标识符与订单状态值进行订单状态的修改。
- 3. 输入订单标识符实现相应订单的查找

4.8.1.4 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. 接受订单标识符与订单状态值的输入，输出修改订单状态完成标志
- 3. 接受订单标识符输入，返回该标识符的订单信息

4.8.1.5 属性描述

属性名	类型	说明
orderId	long	订单的标识符,由系统自动随机生成来唯一标识订单，订单初始化后不允许修改标识符
customerId	long	下单用户的标识符,参照于顾客表的用户标识符，表示下单的用户信息。
orderTime	TimeStamp	下单时间,由系统自动生成表示订单初始化时间信息
memberAmount	int	订单人数，默认值为 1
totalCost	double	订单价格，表示当前订单接受礼券后的总花费，非空
orderStatus	int	订单状态，状态值从 1~6 之间，初始化时默认为 1，由系统自动初始化设置。

表格 115 Order 类属性表

4.8.1.6 方法详细描述

4.8.1.6.1 Order

方法签名：public Order(long customerId, int memberAmount, 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

memberAmount	int	待创建订单的人数
totalCost	double	待创建订单的总花费金额

4.8.1.7 类内、类外方法调用状态图

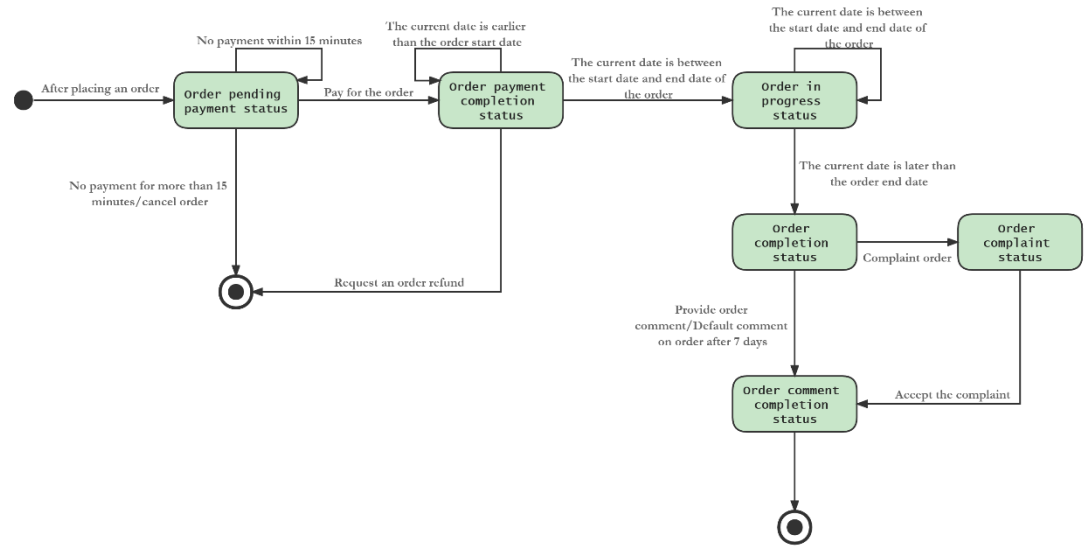


图 2 订单类状态变化图

4.8.1.8 限制条件及出错处理

- 对于不存在的订单查询任务要返回 NullPointerException 空值错误提示

4.8.2 Comment 类

4.8.2.1 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

4.8.2.2 性能

无约束

4.8.2.3 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

4.8.2.4 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

4.8.2.5 属性描述

属性名	类型	说明
commentId	long	订单评论的标识符,由系统自动随机生成来唯一标识订单评论,订单评论初始化后不允许修改标识符
orderId	long	订单的标识符,参照于 Order 表的订单标识符表示订单评论的来源
commentTime	TimeStamp	评论时间,由系统自动生成表示订单评论初始化时间信息
commentContent	String	订单评论的内容,最大长度不超过 400
commentScore	int	订单分数,表示用户对订单的满意打分,默认值为 5,取值范围为 1~5 之间的整数

表格 2 Comment 类属性表

4.8.2.6 方法详细描述

4.8.2.6.1 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	待创建订单评论的指向订单标识符，不能为空且参照 Order 表
orderContent	String	待创建订单评论信息的文本内容，最大长度不超过 400
orderScore	int	待创建订单评论信息的分数，其中取值范围为 1~5，默认值为 5

4.8.2.7 类内、类外方法调用状态图

该类无状态变化过程。

4.8.2.8 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 NullPointerException 空值错误提示

4.8.3 Report 类

4.8.3.1 功能

订单微服务中的 Report 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。

4.8.3.2 性能

无约束

4.8.3.3 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。

4.8.3.4 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

4.8.3.5 属性描述

属性名	类型	说明
reportId	long	订单投诉信息的标识符,由系统自动随机生成来唯一标识订单投诉信息,订单投诉信息初始化后不允许修改标识符
orderId	long	订单的标识符,参照于 Order 表的订单标识符表示订单评论的来源
reportTime	TimeStamp	投诉时间,由系统自动生成表示订单投诉信息初始化时间信息
reportReason	String	订单投诉原因,表示订单投诉的理由,最大长度不超过 200
isDealt	boolean	订单投诉处理标志,表示是否处理了该订单投诉信息
adminId	int	处理订单投诉的管理员标识符,参照于管理员 admin 表
dealTime	TimeStamp	处理订单投诉过程的时间,由系统自动根据当前时间生成
reply	String	处理订单投诉的回复内容,由系统管理员填写对投诉结果的说明,长度不超过 200

表格 3 Report 类属性表

4.8.3.6 方法详细描述

4.8.3.6.1 Report

方法签名: public Report(long orderId, String reportReason)
功能说明: Report 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	待创建订单投诉信息的指向订单标识符，不能为空且参照 Order 表
reportReason	String	待创建订单投诉信息的文本内容，最大长度

		不超过 200
--	--	---------

4.8.3.6.2 dealReport

方法签名：public void dealReport(long reportId, int adminId, String reply)

功能说明：处理存在的订单举报信息，添加处理的订单举报处理结果

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）		
reportId	long	待处理订单投诉信息的标识符，不能为空且参照于 report 表
adminId	int	待处理订单投诉信息的处理人员标识符信息，不能为空，且参照于 admin 表
reply	String	待处理订单投诉信息的回复内容，长度不超过 200，默认值为空文本内容

4.8.3.7 类内、类外方法调用状态图

该类无状态变化过程。

4.8.3.8 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 NullPointerException 空值错误提示
- 对于已经处理好的投诉信息，如果再次调用处理举报函数会提示重复订单投诉的错误提示

4.8.4 OrderManager 类

4.8.4.1 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。

4.8.4.2 性能

无约束

4.8.4.3 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

4.8.4.4 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

4.8.4.5 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,用来存储符合筛选条件的订单信息或者对输入信息切片后得到的订单信息内容。数据结构格式标准参照于 Order 类

表格 4 OrderManager 类属性表

4.8.4.6 方法详细描述

4.8.4.6.1 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，非空且参照于 customer 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，取值不小于 0

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。

4.8.4.6.2 getHostOrderList

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

功能说明：根据输入的房主标识符分页筛选该房主相关的房源全部订单信息

详细说明：

返回类型	List<Order>	
返回值	筛选出的该房主房源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）		
hostId	long	房主标识符，非空且参照于 host 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，取值不小于 0

实现过程：

该函数首先是根据筛选该房主的全部房源信息，然后根据房源标识符直接筛选该房主的全部订单信息，并且根据分页条件进行部分数据返回。

4.8.4.6.3 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。

4.8.4.6.4 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。

4.8.4.6.5 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 1~6，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。

4.8.4.6.6 getFootMapInfo

方法签名：public FootMapInfo getFootMapInfo(long customerId)

功能说明：返回该用户的订单足迹信息，即该用户订单的省市信息

详细说明：

返回类型	FootMapInfo(Data Dictionary)	
返回值	该用户的订单足迹信息	
方法限定符	Public	
参数列表（参数名 / 类型 / 说明）		
customerId	long	查找足迹信息的用户标识符，非空，参照于customer表

实现过程：首先判断 customerId 指向的用户是否存在，如果存在则筛选该用户的全部订单，并且根据正则匹配的原则筛选订单列表与省市信息的对应关系，包装成相应的 FootMapInfo 数据字典，并返回给上层，反之，如果用户不存在，则提示“用户不存在”的报错信息。

4.8.4.6.7 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”。

4.8.4.6.8 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 NullPointerException 的错误信息提示。

4.8.4.7 类内、类外方法调用状态图

该类无状态变化过程。

4.8.4.8 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 NullPointerException 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

4.8.5 ReportManager 类

4.8.5.1 功能

订单微服务中的 ReportManager 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。

4.8.5.2 性能

无约束

4.8.5.3 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

4.8.5.4 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

4.8.5.5 属性描述

属性名	类型	说明
report	Report	reportManager 维护的一个订单举报信息实体

表格 3 ReportManager 类属性表

4.8.5.6 方法详细描述

4.8.5.6.1 addOrderReport

方法签名: public boolean addOrderReport(Report report)

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）		
report	Report	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”。

4.8.5.6.2 getOrderReport

方法签名: public Report getOrderReport(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	Report	
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举报信息, 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”。

4.8.5.7 类内、类外方法调用状态图

该类无状态变化过程。

4.8.5.8 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

4.8.6 CommentManager 类

4.8.6.1 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，添加房东评论等内容，或者根据订单编号返回评论信息。

4.8.6.2 性能

无约束

4.8.6.3 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

4.8.6.4 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

4.8.6.5 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
---------	---------	------------------------------

表格 4 CommentManager 类属性表

4.8.6.6 方法详细描述

4.8.6.6.1 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”。

4.8.6.6.2 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”。

4.8.6.7 类内、类外方法调用状态图

该类无状态变化过程。

4.8.6.8 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

4.8.7 OrderInterface 类

4.8.7.1 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。

4.8.7.2 性能

无约束

4.8.7.3 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、订单举报信息的业务逻辑处理。

4.8.7.4 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

4.8.7.5 属性描述

接口类中不维护属性

4.8.7.6 方法详细描述

4.8.7.6.1 `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)`

4.8.7.6.2 `getHostOrderList`

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

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

实际调用方法:

`OrderManager.getHostOrderList(hostId,orderStatus,currentPage,pageSize)`

4.8.7.6.3 `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)`

4.8.7.6.4 `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)`

4.8.7.6.5 `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)`

4.8.7.6.6 getFootMapInfo

方法签名: `public ResponseEntity getFootMapInfo()`

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

实际调用方法:

`OrderManager.getFootMapInfo(customerId)`

4.8.7.6.7 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)`

4.8.7.6.8 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)`

4.8.7.6.9 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)`

4.8.7.6.10 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)`

4.8.7.7 类内、类外方法调用状态图

该类无状态变化过程。

4.8.7.8 限制条件及出错处理

- HttpStatus 401 表示未登录状态，应提示用户“您需要先登录系统”的信息

4.9 支付微服务

支付微服务主要负责联系阿里支付接口实现该系统中的订单支付、退款功能，为上层微服务提供底层的支付功能的实现。在该微服务中主要存在 1 个实体类，负责为支付功能实现提供基本的订单信息支持。除此之外，存在 1 个配置类，通过配置密钥与远端阿里支付接口建立连接。在此之上，通过 2 个控制类 Manager 分别管理订单付款与退款功能以及回调的实现，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：

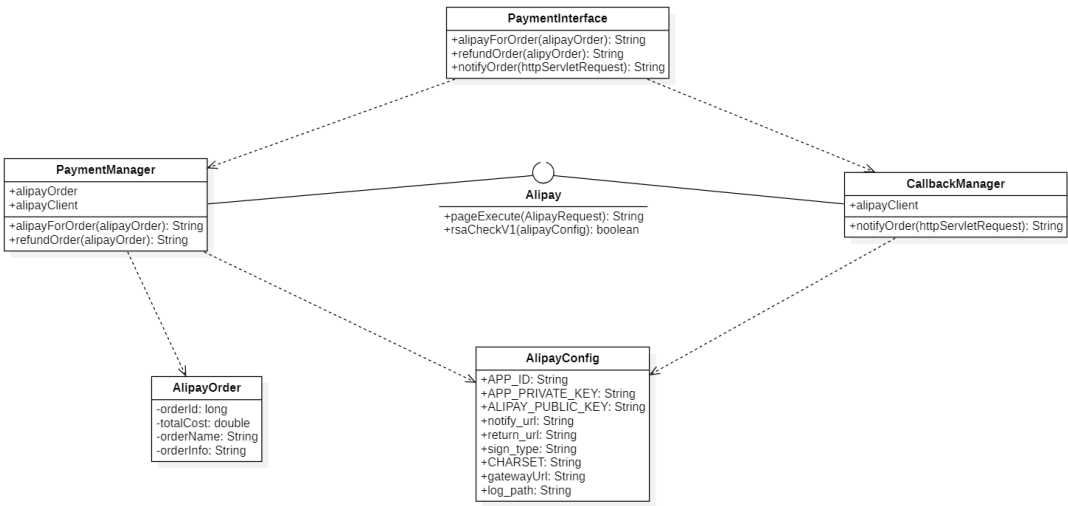


图 17 支付微服务详细类设计

接下来将会详细介绍这几个类的描述。

4.9.1 AlipayOrder 类

4.9.1.1 功能

该类中主要包含了阿里支付接口必要的订单信息，通过对上层对订单信息的提取与构造，实现对必要订单信息的持久化。

4.9.1.2 性能

无约束

4.9.1.3 输入项

1. 输入订单的标识符，订单总花费金额，订单名称与订单的简要信息实现对该类的初始化

4.9.1.4 输出项

1. 接受输入订单的标识符，订单总花费金额，订单名称与订单的简要信息，输出对应初始化操作的标志。

4.9.1.5 属性描述

属性名	类型	说明
orderId	long	订单的标识符信息，唯一标识订单实体，且非空
totalCost	double	订单的总花费金额信息，非空
orderName	String	订单的名称，非空，长度不超过 50
orderInfo	String	订单的主要信息，非空，长度不超过 100

表格 5 AlipayOrder 类属性表

4.9.1.6 方法详细描述

4.9.1.6.1 AlipayOrder

方法签名：public AlipayOrder(long orderId, double totalCost, String orderName, String orderInfo)

功能说明：该类的构造函数，对 alipay 订单信息进行初始化

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 / 类型 / 说明）		
orderId	long	订单的标识符信息，唯一标识订单实体，且非空
totalCost	double	订单的总花费金额信息，非空
orderName	String	订单的名称，非空，长度不超过 50
orderInfo	String	订单的主要信息，非空，长度不超过 100

实现过程：

该函数为基本的构造函数，通过在接受相应的参数传递后，即可完成对 AlipayOrder 对象的构建，如果参数超过长度，则返回用户“订单信息长度超过限制值”的错误信息。

4.9.1.7 类内、类外方法调用状态图

该类无状态变化过程。

4.9.1.8 限制条件及出错处理

- 对于参数超过限制长度的情况，返回用户“订单信息长度超过限制值”的错误信息
- 对于传递参数为空的情况，返回用户“参数值为空”的错误信息

4.9.2 AlipayConfig 类

4.9.2.1 功能

该类中为与 Aplipy 建立连接的配置类，包含了基本的密钥信息、回调通知地址、验签方式等基本静态信息。

4.9.2.2 性能

无约束

4.9.2.3 输入项

该类为配置类，不接受任何形式的输入

4.9.2.4 输出项

该类为配置类，无相关调用方法进行输出

4.9.2.5 属性描述

属性名	类型	说明
-----	----	----

APP_ID	String	应用 Id,该字段对应了支付宝账号
APP_PRIVATE_KEY	String	商户私钥，为 PKCS8 格式 RSA2 私钥
APP_PUBLIC_KEY	String	支付宝公钥
notify_url	String	服务器异步通知地址路径
return_url	String	页面跳转同步通知页面路径
sign_type	String	签名方式
CHARSET	String	字符编码格式
gatewayUrl	String	支付宝网关，利用沙盒环境下的网关地址
log_path	String	日志输出地址

表格 5 AlipayConfig 类属性表

4.9.2.6 方法详细描述

该类的属性字段全部为静态值，无内部方法进行调用与逻辑处理

4.9.2.7 类内、类外方法调用状态图

该类无状态变化过程。

4.9.2.8 限制条件及出错处理

无

4.9.3 PaymentManager 类

4.9.3.1 功能

该类主要负责接受订单信息完成对订单支付与退款操作。

4.9.3.2 性能

无约束

4.9.3.3 输入项

1. 输入完整的 AlipayOrder 信息实现对该订单的付款与退款功能

4.9.3.4 输出项

1. 接受完整的 AlipayOrder 信息的输入，输出支付成功后返回的表单字符串
2. 接受完整的 AlipayOrder 信息的输入，输出退款成功后的通知字符串信息

4.9.3.5 属性描述

属性名	类型	说明
alipayOrder	Alipayorder	待支付或待退款的订单信息
alipayClient	AlipayClient	基于 Alipay 接口实现对请求的提交的客户端实体

表格 5 PaymentManager 类属性表

4.9.3.6 方法详细描述

4.9.3.6.1 alipayForOrder

方法签名：public String alipayForOrder(alipayOrder)

功能说明：实现对订单的支付提交功能，并且返回支付页面的表单。

详细说明：

返回类型	String	
返回值	完整包含 Alipay 提供的支付界面的表单信息	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
alipayOrder	AlipayOrder	待提交支付申请的订单详细信息

实现过程：

该函数实现对支付信息的提交。首先获得参数中的 alipayOrder 信息，将该对象转化为 Json 格式信息，通过封装为 AlipayTradePagePayRequest 类型，由 alipayClient 进行提交得到相应的表单返回值。

4.9.3.6.2 refundOrder

方法签名：public String refundOrder(alipayOrder)

功能说明：实现对订单的退款提交功能，并且返回退款操作标志。

详细说明：

返回类型	String	
返回值	退款操作标志信息	

方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
alipayOrder	AlipayOrder	待提交退款申请的订单详细信息

实现过程：

该函数实现对支付信息的提交。首先获得参数中的 alipayOrder 信息，将该对象转化为 Json 格式信息，通过封装为 AlipayTradeRefundRequest 类型，由 alipayClient 进行提交得到相应的操作标志。

4.9.3.7 类内、类外方法调用状态图

该类无状态变化过程。

4.9.3.8 限制条件及出错处理

- 对于参数为空的情况，返回错误信息“输入参数为空”
- 对于接口调用失误的情况，返回 AlipayApiException 异常信息

4.9.4 CallbackManager 类

4.9.4.1 功能

该类主要负责支付成功后的回调操作与相关业务逻辑的处理。

4.9.4.2 性能

无约束

4.9.4.3 输入项

1. 输入来自 Alipay 远端传来的 HttpServletRequest 信息，执行相关的回调函数处理。

4.9.4.4 输出项

1. 接受来自远端的 Http 请求，返回经过验签与业务逻辑处理后的结果，以字符串的形式返回

4.9.4.5 属性描述

属性名	类型	说明
alipayClient	AlipayClient	基于 Alipay 接口实现对请求的提交的客户端实体

表格 5 CallbackManager 类属性表

4.9.4.6 方法详细描述

4.9.4.6.1 notifyOrder

方法签名：public String notifyOrder(httpServletRequest)

功能说明：在支付成功后，接受阿里支付远端发送的 http 请求，在验签通过后执行相应业务逻辑处理，并返回字符串形式的处理结果。

详细说明：

返回类型	String	
返回值	验签后的处理结果，” success” 或者” failure” 传给阿里支付远端	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
HttpServletRequest	HttpServletRequest	由阿里支付远端传来的包含订单信息的网络请求

实现过程：

该函数为被动调用过程，在用户成功支付后，阿里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，如修改订单状态，最后返回验签结果值。

4.9.4.7 类内、类外方法调用状态图

该类无状态变化过程。

4.9.4.8 限制条件及出错处理

无

4.9.5 PaymentInterface 类

4.9.5.1 功能

该类为支付微服务对外功能的总接口体现,支付微服务通过整合阿里支付提供的功能整合并暴露给其他微服务使用,并完成相应业务逻辑的处理。

4.9.5.2 性能

无约束

4.9.5.3 输入项

1. 接受 alipayOrder 信息与来自网络的 Http 请求实现服务的调用

4.9.5.4 输出项

1. 返回支付界面表单信息与相应接口调用的结果

4.9.5.5 属性描述

接口类不维护属性

4.9.5.6 方法详细描述

4.9.5.6.1 alipayForOrder

方法签名: public String alipayForOrder(AlipayOrder alipayOrder)

接口调用: alipayClient.excutePage(tradeRequest) 通过阿里客户端执行相应的提交

实际调用方法:

paymentManger.alipayForOrder(alipayOrder)

4.9.5.6.2 refundOrder

方法签名: public String refundOrder(AlipayOrder alipayOrder)

接口调用: alipayClient.excutePage(tradeRequest) 通过阿里客户端执行相应的提交

实际调用方法:

paymentManger.refundOrder(alipayOrder)

4.9.5.6.3 notifyOrder

方法签名: public String notifyOrder(httpServletRequest)

接口调用: alipayClient.rsaCheckV1(alipayConfig) 通过阿里客户端执行验签

实际调用方法:

paymentManger.notifyOrder(httpServletRequest)

4.9.5.7 类内、类外方法调用状态图

该类无状态变化过程。

4.9.5.8 限制条件及出错处理

无

4.10 交易微服务

交易微服务主要负责整合订单与支付微服务提供的功能接口实现订单业务流程的完整实现。在该微服务中主要存在 1 个实体类,负责接受来自前端表单返回的订单信息,将该信息经过分片可以分别调用订单与支付微服务的接口实现相应的业务功能。由于该微服务中集成了 Redis 中间件服务,通过 Utils 实现对相应基础功能的调用。该微服务中存在一个控制类负责整合对外部接口功能的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。交易设计类图如下:

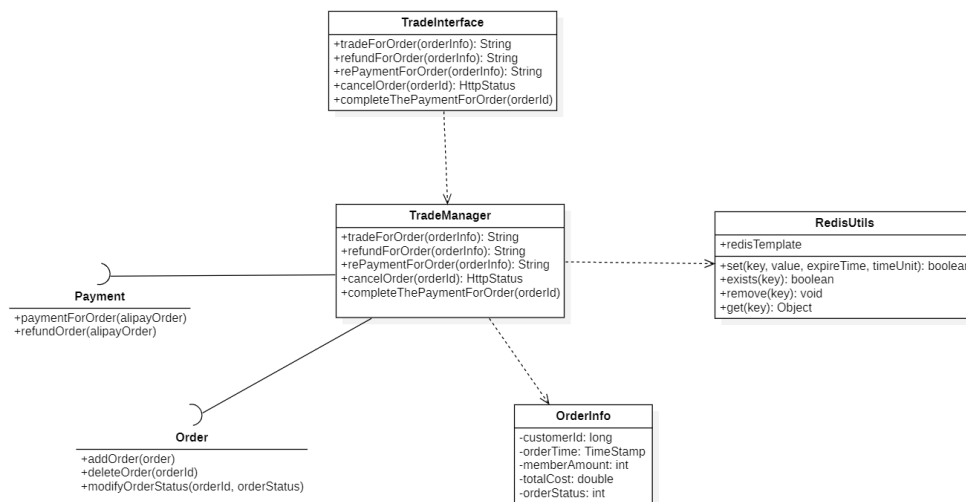


图 18 交易微服务详细类设计

接下来将会详细介绍这几个类的描述。

4.10.1 OrderInfo 类

4.10.1.1 功能

该实体类接受来自前端的表单订单数据, 存储后将该信息分别为调用订单与支付的接口实现提供数据支持。

4.10.1.2 性能

无约束

4.10.1.3 输入项

1. 接受顾客标识符、订单人数、订单总花费、订单状态为该类进行初始化构造

4.10.1.4 输出项

1. 接受初始化参数后, 输出构造操作标志

4.10.1.5 属性描述

属性名	类型	说明
customerId	long	下单用户的标识符，参照于 customer 表
orderTime	TimeStamp	下单时间
memberAmount	int	订单人数
totalCost	double	订单总消耗费用
orderStatus	int	订单状态

表格 5 OrderInfo 类属性表

4.10.1.6 方法详细描述

4.10.1.6.1 OrderInfo

方法签名：public OrderInfo(long customerId, int memberAmount, double totalCost)

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 / 类型 / 说明）		
customerId	long	下单用户的标识符，参照于 customer 表
memberAmount	int	订单人数
totalCost	double	订单总消耗费用

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，如果参数有错误数据，则返回用户“输入参数有误”的错误信息。

4.10.1.7 类内、类外方法调用状态图

该类无状态变化过程。

4.10.1.8 限制条件及出错处理

1. 接收参数有错误，返回用户“输入参数有误”错误信息

4.10.2 RedisUtils 类

4.10.2.1 功能

该类为操作 Redis 中间件的工具包类，实现对 Redis 的基本增删改查操作

4.10.2.2 性能

无约束

4.10.2.3 输入项

1. 接受 Redis 中的键、对应的值、失效时间实现插入数据的操作
2. 接受键实现判断是否值在 Redis
3. 接受键实现对 Redis 中数据的删除操作

4.10.2.4 输出项

1. 在插入数据过程中，接受键与值输出操作的标志

4.10.2.5 属性描述

属性名	类型	说明
redisTemplate	RedisTemplate	操作 RedisTemplate 的工具类

表格 5 RedisUtils 类属性表

4.10.2.6 方法详细描述

4.10.2.6.1 set

方法签名：public boolean set(key, value, expireTime, timeUnit)

功能说明：将对应的键值对插入到 Redis 中，并设置相应的失效时间

详细说明：

返回类型	boolean
------	---------

返回值	插入操作的执行标志	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
key	String	插入到 Redis 中的键信息
value	orderInfo	插入到 Redis 中的值信息
expireTime	int	失效时间
timeUnit	TimeUnit	失效时间单位

4.10.2.6.2 remove

方法签名：public boolean remove(key)

功能说明：在 Redis 中查询相应键的内容，并且将该键对应的记录删除

详细说明：

返回类型	boolean	
返回值	删除操作执行的标志	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
key	String	删除到 Redis 中的键信息

4.10.2.7 类内、类外方法调用状态图

该类无状态变化过程。

4.10.2.8 限制条件及出错处理

无

4.10.3 TradeManager 类

4.10.3.1 功能

该类控制整个交易的过程执行与对外部接口的调用，实现整体订单的复杂业务流程

4.10.3.2 性能

无约束

4.10.3.3 输入项

1. 接受表单订单信息实现对业务逻辑的处理

4.10.3.4 输出项

1. 输出经过操作后的执行状态

4.10.3.5 属性描述

无

4.10.3.6 方法详细描述

4.10.3.6.1 tradeForOrder

方法签名：public String tradeForOrder(OrderInfo orderInfo)

功能说明：创建订单，该过程需要利用订单微服务插入信息的订单，并且调用支付微服务的接口提供支付能力。

接口调用：

1. paymentForOrder(alipayOrder) 实现支付能力的调用
2. addOrder(order) 实现对新订单的插入

详细说明：

返回类型	String	
返回值	支付页面表单信息	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
orderInfo	OrderInfo	输入的订单信息

实现过程：

调用该函数实现初始的订单创建过程。首先调用订单微服务将抽取的订单信息插入到数据库中，然后再生成对应的 AlipayOrder 调用支付微服务的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。

4.10.3.6.2 refundForOrder

方法签名：public String refundForOrder(OrderInfo orderInfo)

功能说明：实现对订单退款的业务逻辑。首先需要调用支付微服务中的退款接口，在退

款成功后，通过调用订单微服务的删除订单接口实现对订单的移除。

接口调用：

1. refundOrder(alipayOrder) 实现退款功能的调用
2. deleteOrder(orderId) 实现对订单的删除

详细说明：

返回类型	String	
返回值	退款操作的标志	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
orderInfo	OrderInfo	待退款的提交订单信息

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删除订单功能，将该订单从数据库中移除，完成整体的退款流程。

4.10.3.6.3 completeThePaymentForOrder

方法签名：public String completeThePaymentForOrder (long orderId)

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：

1. modifyOrderStatus(orderId, orderStatus) 实现对订单状态的修改

详细说明：

返回类型	String	
返回值	操作完成标志	
方法限定符	public	
参数列表（参数名 / 类型 / 说明）		
orderId	OrderId	待确定支付完成的订单标识符

实现过程：

该函数实现完成订单支付确认业务流程，首先函数会通过调用订单微服务完成对订单状态的修改，然后在此基础上调用 redisUtils 类中的 remove 方法移除存储在 Redis 中的相应订单信息。

4.10.3.7 类内、类外方法调用状态图

该类无状态变化过程。

4.10.3.8 限制条件及出错处理

无

4.10.4 TradeInterface 类

4.10.4.1 功能

该类为交易微服务业务功能的整体对外体现,通过暴露接口实现对订单交易过程的控制

4.10.4.2 性能

无约束

4.10.4.3 输入项

1. 接受表单订单信息实现对业务逻辑的处理

4.10.4.4 输出项

1. 输出经过操作后的执行状态

4.10.4.5 属性描述

无

4.10.4.6 方法详细描述

4.10.4.6.1 tradeForOrder

方法签名: public String tradeForOrder(OrderInfo orderInfo)

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

实际调用方法:

tradeManger. tradeForOrder(orderInfo)

4.10.4.6.2 refundForOrder

方法签名: public String refundForOrder(OrderInfo orderInfo)

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

实际调用方法:

tradeManger.refundForOrder(orderInfo)

4.10.4.6.3 completeThePaymentForOrder

方法签名：public String completeThePaymentForOrder (long orderId)

接口调用：无调用

实际调用方法:

tradeManger. completeThePaymentForOrder (orderId)

4.10.4.7 类内、类外方法调用状态图

该类无状态变化过程。

4.10.4.8 限制条件及出错处理

无

5 索引

表格 1 User 类属性表	7
表格 2 checkPassword 方法说明	8
表格 3 changePassword 方法说明	8
表格 4 SignupManager 类属性表	9
表格 5 signup 方法说明	1 0
表格 6 validate 方法说明	1 0
表格 7 checkPhoneUnique 方法说明	1 1
表格 8 upgradeToHost 方法说明	1 1
表格 9 changePassword 方法说明	1 2
表格 10 changePasswordBySMS 方法说明	1 2
表格 11 SMSManager 类属性表	1 3
表格 12 sendSMS 方法说明	1 4
表格 13 validate 方法说明	1 4
表格 14 residentVerify 方法说明	1 5
表格 15 getNextId 方法说明	1 6
表格 16 AuthenticationCkecking 类属性表	1 7
表格 17 sendSMS 方法说明	1 7
表格 18 validate 方法说明	1 8
表格 19 User 类属性表	1 9
表格 20 checkExpired 方法说明	1 9
表格 21 TokenManager 类属性表	2 0
表格 22 generateToken 方法说明	2 1
表格 23 checkLoginState 方法说明	2 1
表格 24 getUserFromToken 方法说明	2 2
表格 25 autoExpireTokens 方法说明	2 2
表格 26 removeToken 方法说明	2 2
表格 27 Post 类属性表	2 6
表格 28 PostManager 类属性表	2 7
表格 29 getPost 方法说明	2 7
表格 30 addPost 方法说明	2 8
表格 31 getPostList 方法说明	2 8
表格 32 autoExpireTokens 方法说明	2 9
表格 33 Post 类属性表	3 0
表格 34 PostManager 类属性表	3 1
表格 35 createReply 方法说明	3 1
表格 36 getRepliesOfPost 方法说明	3 2
表格 37 getReply 方法说明	3 2
表格 38 Report 类属性表	3 3
表格 39 ReportManager 类属性表	3 4
表格 40 createReport 方法说明	3 5
表格 41 handleReport 方法说明	3 5

表格 42	getReport 方法说明	3 6
表格 43	Bed 类属性表	3 9
表格 44	Bed 方法说明	3 9
表格 45	Room 类属性表	4 0
表格 46	Room 方法说明	4 1
表格 47	Label 类属性表	4 2
表格 48	Label 方法说明	4 2
表格 49	Advertise 类属性表	4 3
表格 50	Advertise 方法说明	4 4
表格 51	Stay 类属性表	4 6
表格 52	changeStayStatus 方法说明	4 6
表格 53	RoomManager 类属性表	4 8
表格 54	addRoom 方法说明	4 8
表格 55	findRoomByStayId 方法说明	4 9
表格 56	deleteStay 方法说明	4 9
表格 57	updateStay 方法说明	4 9
表格 58	StayManager 类属性表	5 0
表格 59	findStayByStayId 方法说明	5 1
表格 60	createStay 方法说明	5 1
表格 61	deleteStay 方法说明	5 1
表格 62	updateStay 方法说明	5 2
表格 63	StayManager 类属性表	5 4
表格 64	getStaySummary 方法说明	5 4
表格 65	OrderManager 类属性表	5 5
表格 66	getOrderSummary 方法说明	5 5
表格 67	Customer 类属性表	5 8
表格 68	getCustomerId 方法说明	5 8
表格 69	updateCustomerInfo 方法说明	5 8
表格 70	Host 类属性表	5 9
表格 71	getHostId 方法说明	6 0
表格 72	favoriteDir 类属性表	6 1
表格 73	getFavorite 方法说明	6 1
表格 74	getFavorite 方法说明	6 2
表格 75	getFavorite 方法说明	6 2
表格 76	UserGroup 类属性表	6 3
表格 77	getUserGroup 方法说明	6 3
表格 78	HostGroup 类属性表	6 4
表格 79	getHostGroup 方法说明	6 5
表格 80	FavoriteManager 类属性表	6 6
表格 81	addStayToFavorite 方法说明	6 6
表格 82	deleteFavorite 方法说明	6 7
表格 83	deleteStayFromFavorite 方法说明	6 7
表格 84	getFavoriteDirectory 方法说明	6 7

表格 85 getFavoriteStayInfo 方法说明	6 8
表格 86 addFavorite 方法说明	6 8
表格 87 PersonInfoManager 类属性表	6 9
表格 88 checkPassword 方法说明	6 9
表格 89 updateAvatar 方法说明	7 0
表格 90 updateUserInfo 方法说明	7 0
表格 91 getCustomerGroupInfo 方法说明	7 0
表格 92 updateHostInfo 方法说明	7 1
表格 93 base64UploadFile 方法说明	7 2
表格 94 Report 类属性表	7 6
表格 95 getReport 方法说明	7 7
表格 96 findReport 方法说明	7 7
表格 97 UnverifiedStay 类属性表	7 8
表格 98 findUnverifiedStay 方法说明	7 8
表格 99 findReport 方法说明	7 9
表格 100 ReportManager 类属性表	8 0
表格 101 getPostReportDetail 方法说明	8 0
表格 102 postPostReport 方法说明	8 1
表格 103 Coupon 类属性表	8 4
表格 104 getReport 方法说明	8 4
表格 105 findCoupon(override)方法说明	8 4
表格 106 CouponType 类属性表	8 6
表格 107 addCouponType 方法说明	8 7
表格 108 deleteCouponType 方法说明	8 7
表格 109 CouponUsageMangager 类属性表	8 8
表格 110 getCouponInfoByCustomerId 方法说明	8 8
表格 111 useCouponByCouponId 方法说明	8 9
表格 112 CouponGrantMangager 类属性表	8 9
表格 113 addCouponType 方法说明	9 0
表格 114 grantACoupon 方法说明	9 0
表格 115 Order 类属性表	9 4

图 1 归宿平台系统软件结构图	5
图 2 归宿平台微服务划分调用图	5
图 3 登陆注册微服务详细类设计	6
图 4 User 类方法调用状态图	8
图 5 Token 类方法调用状态图	1 9
图 6 论坛微服务详细类设计	2 5
图 7 Report 类方法调用状态图	3 3
图 8 房源管理微服务详细类设计	3 8
图 9 房源状态图	4 7
图 10 数据统计微服务详细类设计	5 3

图 11 用户信息微服务详细类设计	5 6
图 12 管理微服务详细类设计	7 5
图 13 UnverifiedStay 类方法调用状态图	7 9
图 14 促销微服务详细类设计	8 3
图 15 Coupon 类方法调用状态图	8 5
图 16 订单微服务详细类设计	9 3
图 17 支付微服务详细类设计	1 1 1
图 18 交易微服务详细类设计	1 2 0