



---

# 归宿——一个民宿预定平台

---

## 概要设计规约



1851055 汪明杰

1851049 陈中悦

1851231 王立友

1853572 梁 乔

1954098 香宁雨

# 归宿概要设计规约（说明书）

## 目录

1. 引言.....	3
1.1. 概要设计依据.....	3
1.2. 参考资料.....	3
1.3. 假定和约束.....	3
2. 概要设计.....	4
2.1. 系统总体架构设计.....	4
2.2. 系统软件结构设计.....	4
2.3. 接口设计.....	5
2.3.1 订单微服务.....	5
2.3.2 支付微服务.....	7
2.3.3 交易微服务.....	8
2.3.4 登录注册微服务.....	9
2.3.5 管理微服务.....	11
2.3.6 论坛微服务.....	12
2.3.7 房源微服务.....	14
2.3.8 统计微服务.....	15
2.3.9 个人信息微服务.....	15
2.3.10 交易微服务.....	17
2.4. 界面设计.....	18
登录注册子系统.....	18
用户信息子系统.....	19
房源管理系统.....	22
选购子系统.....	23
订单子系统.....	25
论坛子系统.....	26
2.5 数据库设计.....	28
2.5.1 数据库逻辑设计.....	28
2.5.2 数据库物理设计.....	34
2.5.3 数据库设计考虑.....	46
2.6 系统出错处理设计.....	48
2.6.1 出错信息.....	48
2.6.2 补救措施.....	49
2.6.3 系统维护设计.....	50

## 1. 引言

### 1.1. 概要设计依据

- a. 概要设计基本原则;
- b. 归宿-需求分析规约

### 1.2. 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 Pattern》

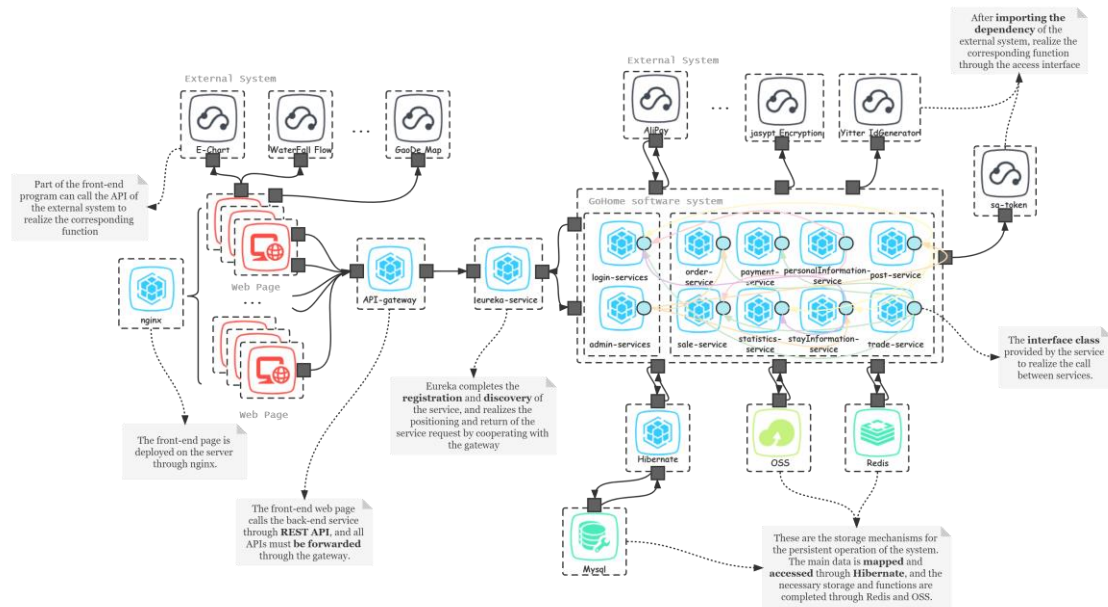
### 1.3. 假定和约束

本项目开发主要受软件工程课程的约束，因此本项目的假定和约束如下所示：

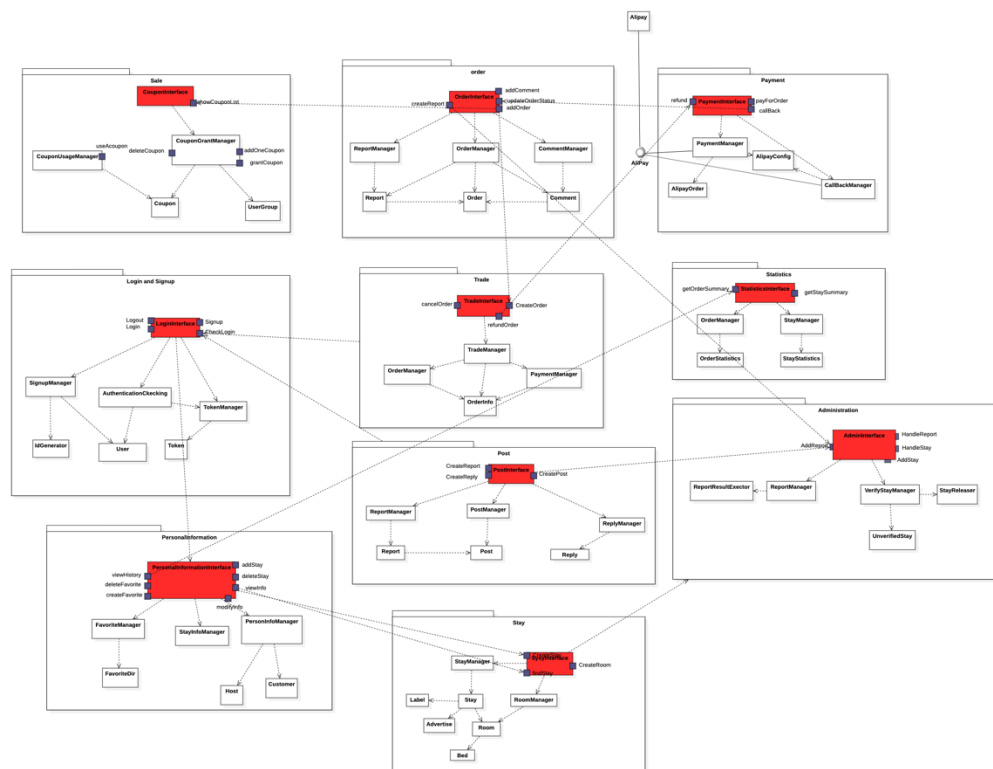
- i. 项目开发期限为 2 个月，时间为 2021 年 11 月~12 月；
- ii. 项目开发无经费，设备条件为 3 台 Windows 操作系统电脑和 1 台 Mac 操作系统电脑以及阿里云云平台等；
- iii. 项目在开发前线下采访了 5 名老师和同学，线上通过问卷调研的方式收集了 217 份问卷，并据此我们制定了用户画像；
- iv. 在交流过程中，我们每三天线下汇报工作进度，同时通过 Github 进行代码协作管理。

## 2. 概要设计

## 2.1. 系统总体架构设计

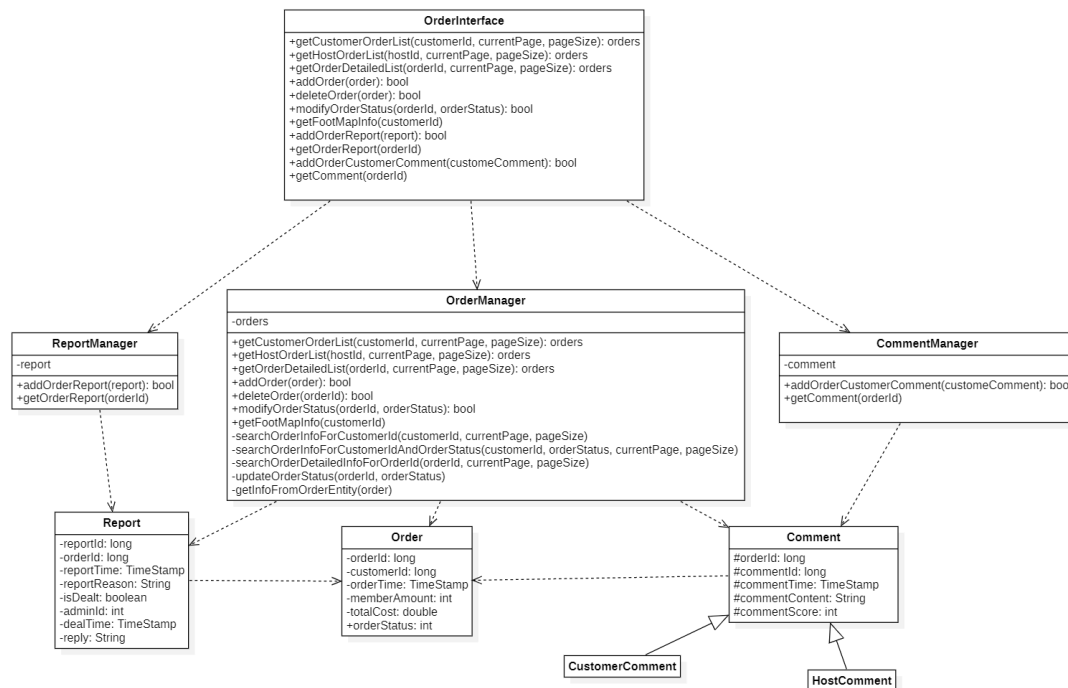


## 2.2. 系统软件结构设计



## 2.3. 接口设计

### 2.3.1 订单微服务



### 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)`

### 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)`

### 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)

## 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)

## 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

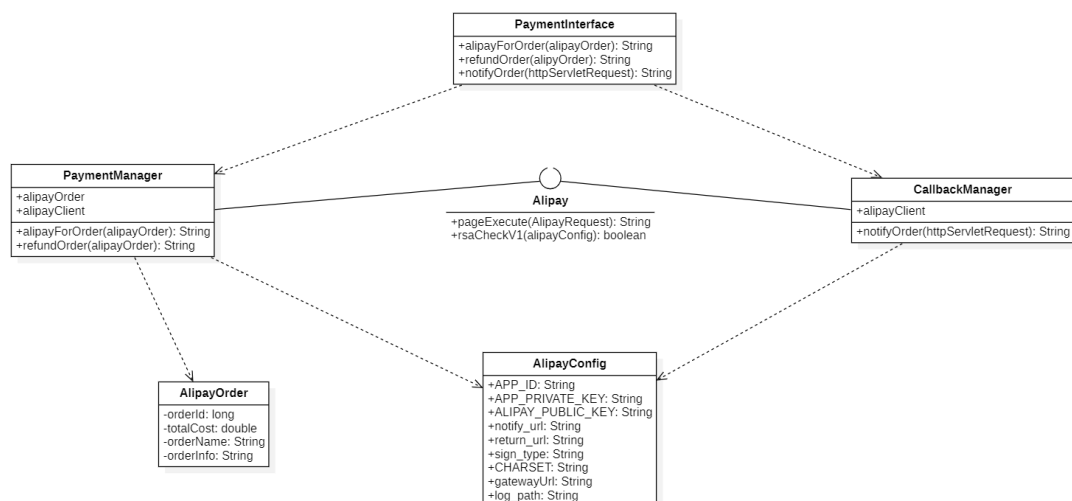
方法签名: `public ResponseEntity getComment(long orderId)`

接口调用: 无外部接口调用

实际调用方法:

`ReportManager.getComment(orderId)`

### 2.3.2 支付微服务



## 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)`

## refundOrder

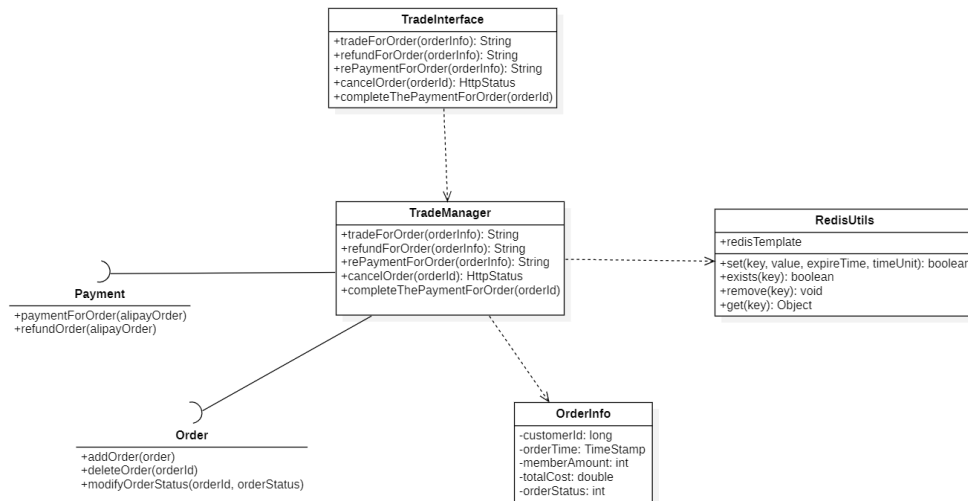
方法签名: `public String refundOrder(AlipayOrder alipayOrder)`

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)

## 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)

### 2.3.3 交易微服务



## 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)

## 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)



## completeThePaymentForOrder

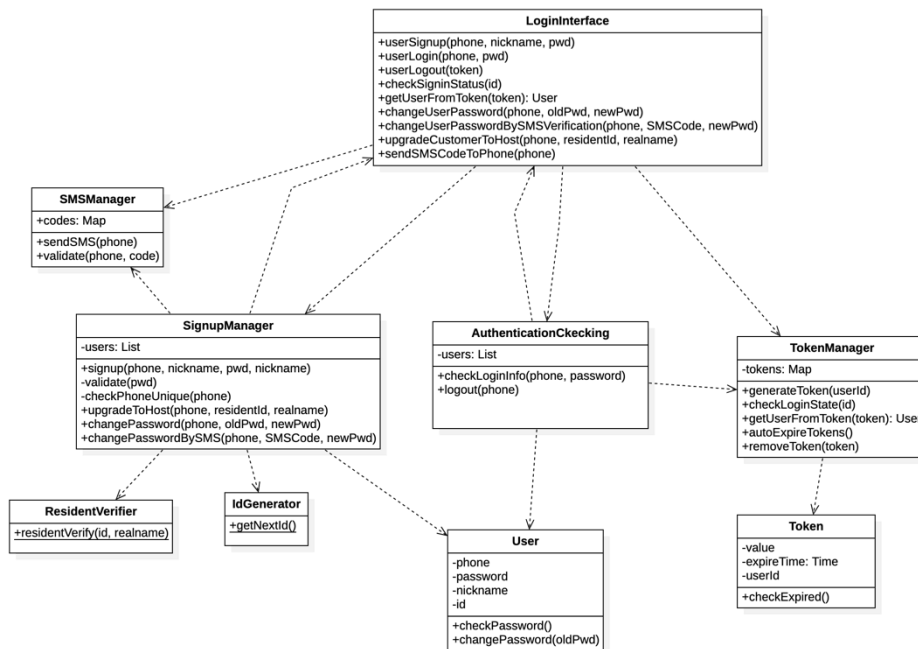
方法签名: `public String completeThePaymentForOrder (long orderId)`

接口调用: 无调用

实际调用方法:

`tradeManger. completeThePaymentForOrder (orderId)`

### 2.3.4 登录注册微服务



## userSignup

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

实际调用方法: `SignupManager.signup`

## userLogin

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

实际调用方法: `SignupManager.signup`

## userLogin

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

实际调用方法: AuthenticationCkecking.checkLoginInfo

## **userLogout**

方法签名: public void userLogout(String token)

实际调用方法: AuthenticationCkecking.logout

## **checkSigninStatus**

方法签名: public Boolean checkSigninStatus(String id)

实际调用方法: TokenManager.checkLoginState

## **getUserFromToken**

方法签名: public User getUserFromToken(String token)

实际调用方法: TokenManager.getUserFromToken

## **changeUserPassword**

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

实际调用方法: SignupManager.changePassword

## **changeUserPasswordBySMSVerification**

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

实际调用方法: SignupManager.changePasswordBySMS

## **upgradeCustomerToHost**

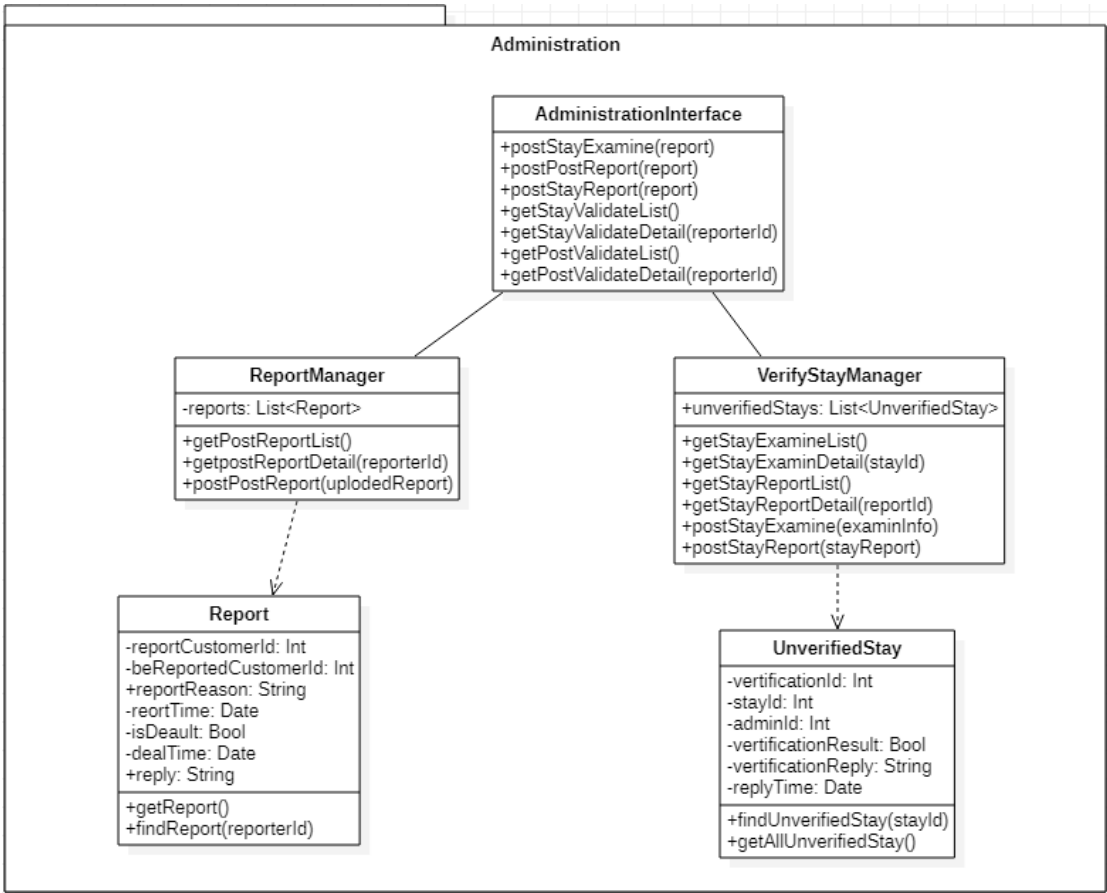
方法签名: public Boolean upgradeCustomerToHost(String phone, String residentId, String realname)

实际调用方法: SignupManager.upgradeToHost

## sendSMSCodeToPhone

方法签名: public Boolean sendSMSCodeToPhone(in phone)  
实际调用方法: SMSManager.sendSMS

### 2.3.5 管理微服务



## 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

## 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

## 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

## getStayValidateList

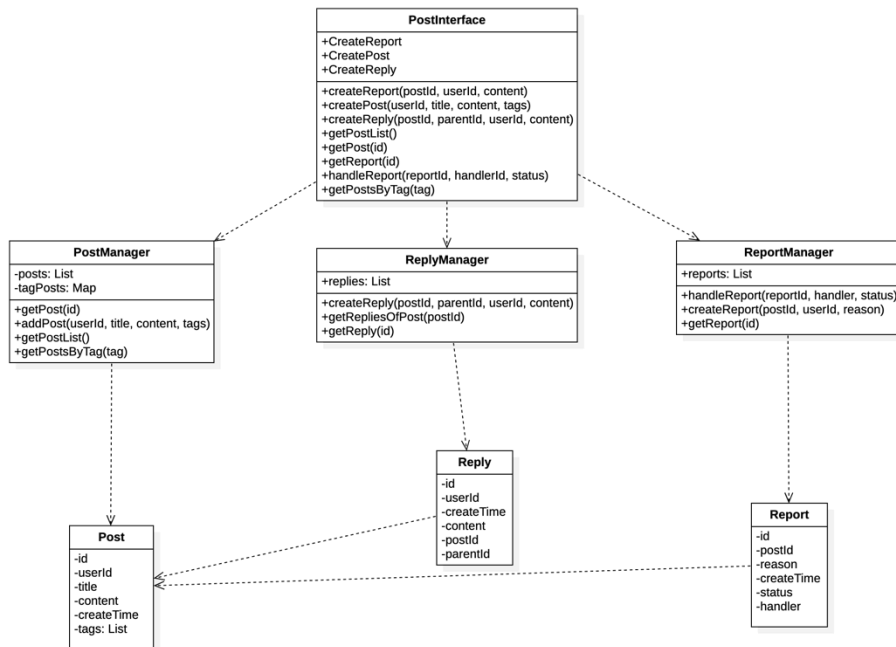
方法签名：public List<UnverifiedStay> getStayValidateList()

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

## 2.3.6 论坛微服务



## createReport

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

实际调用方法: ReportManager.createReport

## **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

## **createReply**

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

实际调用方法: ReplyManager.createReply

## **getPostList**

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

实际调用方法: PostManager.getPostList

## **getPost**

方法签名: public Boolean getPost(Long id)

实际调用方法: PostManager.getPost

## **getReport**

方法签名: public User getReport(Long id)

实际调用方法: ReportManager.getReport

## **handleReport**

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

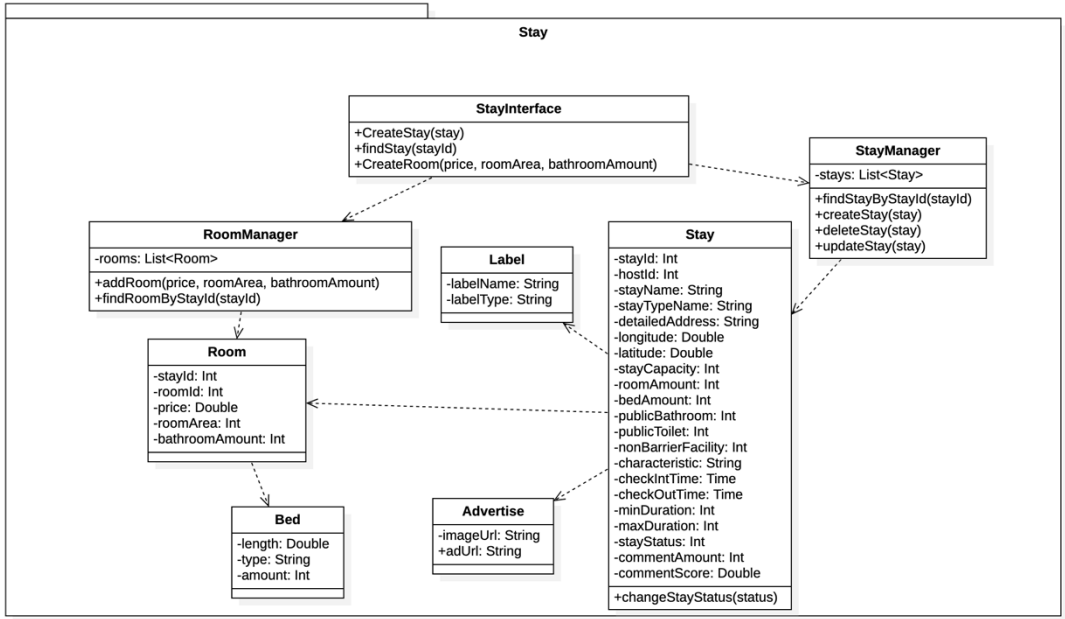
实际调用方法: ReportManager.handleReport

## **getPostsByTag**

方法签名: public Boolean getPostsByTag(String tag)

实际调用方法: PostManager.getPostsByTag

### 2.3.7 房源微服务



#### CreateStay

方法签名： public Stay CreateStay(Stay stay)

实际调用方法： StayManager.createStay

#### findStay

方法签名： public Stay findStay(int stayId)

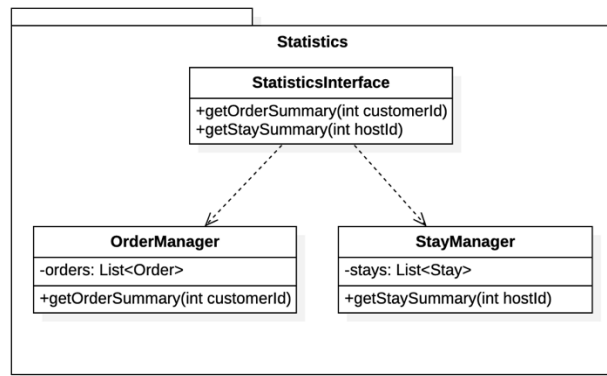
实际调用方法： StayManager.findStayByStayId

#### CreateRoom

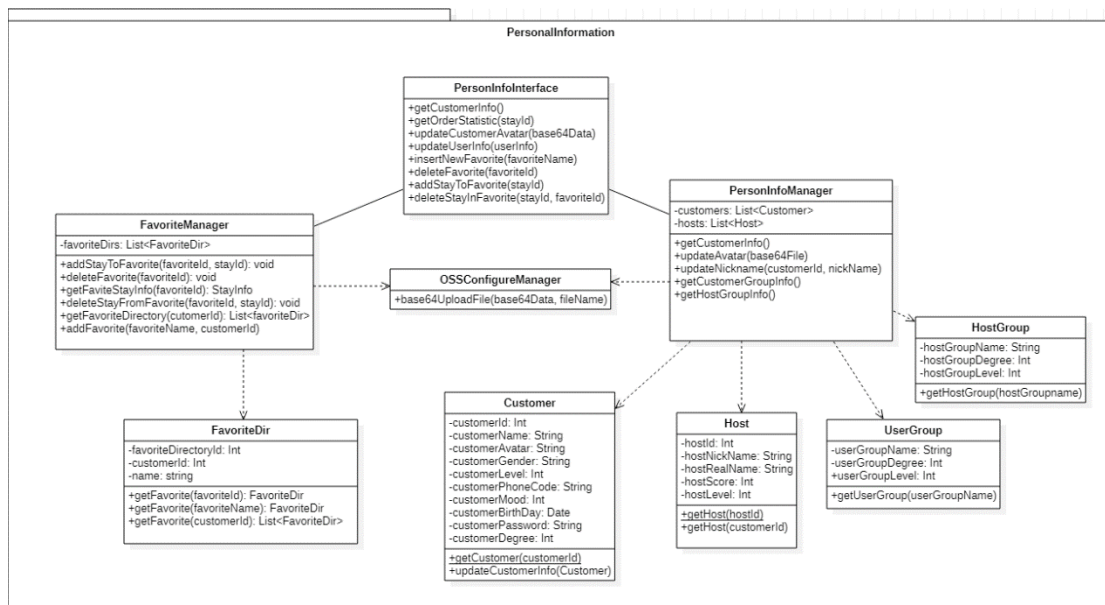
方法签名： public Room CreateRoom(double price, int roomArea, int bathroomAmount)

实际调用方法： RoomManager.addRoom

### 2.3.8 统计微服务



### 2.3.9 个人信息微服务



#### 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

#### 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

## 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

## 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

## 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

## 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

## 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



## addStayToFavorite

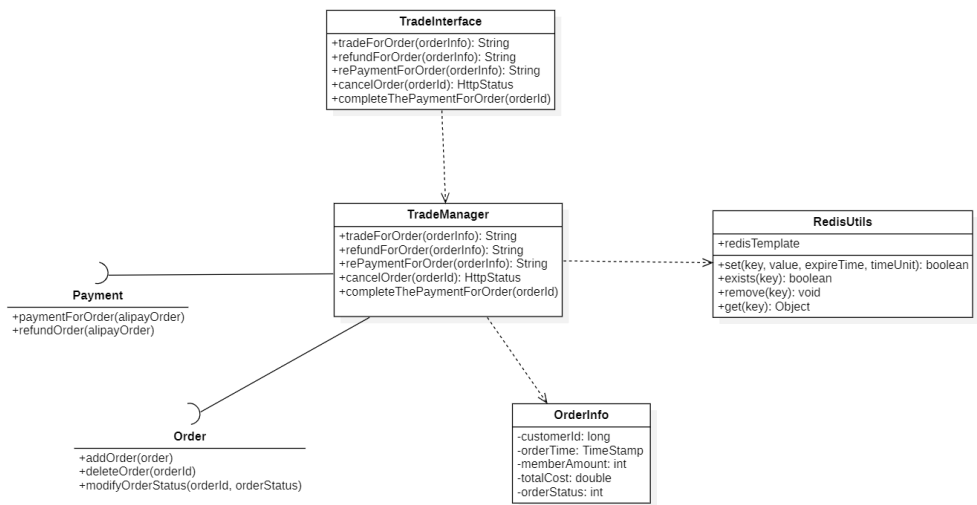
方法签名：public boolean addStaytoFavorite(stayId)

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

### 2.3.10 交易微服务



## 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)

## 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)

## 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)

## 2.4. 界面设计

为了进行交互建模,我们使用了业界最有名的一款专业的快速原型设计软件——

Axure RP。

### 登录注册子系统

#### 1. 登录界面:

The image shows a login interface for a system named '归宿' (Guixu). The interface has a blue header with the system name and a cartoon character. Below the header, the title '登录' (Login) is prominently displayed. A subtitle reads '登录“归宿”，体验更多的精彩世界!' (Login 'Guixu', experience more wonderful world!). There are two tabs: '短信登录' (SMS Login) and '密码登录' (Password Login). The '短信登录' tab is active. Below the tabs, there is a text input field for '手机号' (Mobile Number). Below that is a text input field for '请输入短信验证码' (Please enter SMS verification code). To the right of this field is a blue button labeled '获取验证码' (Get verification code). Below the input fields, there is a checkbox labeled '记住我' (Remember me) and a link '无法验证? 忘记密码?' (Cannot verify? Forgot password?). At the bottom, there are two buttons: a blue '登录' (Login) button and a white '注册' (Register) button.

用户在该界面可以选择“短信登录”或者“密码登录”。

选择“短信登录”，则填写手机号后发送验证码，即可以收到短信验证码。输入所收到的短信验证码，则可以进行登录操作；选择“密码登录”，则需要输入手机验证码和账号密码。

2. 注册界面：

归宿

注册

昵称

密码(6-16个字符组成，区分大小写)

填写常用手机号

请输入短信验证码

获取验证码

☐ 我已同意《归宿用户使用协议》

注册

已有账号，直接登录>

在注册页面，用户需要输入个人昵称、密码和个人手机号。之后，点击“获取验证码”按钮，验证码会被发送到对应的手机上。用户输入正确的验证码，点击同意协议，即可成功注册账号，并跳转到下一界面：

归宿

注册

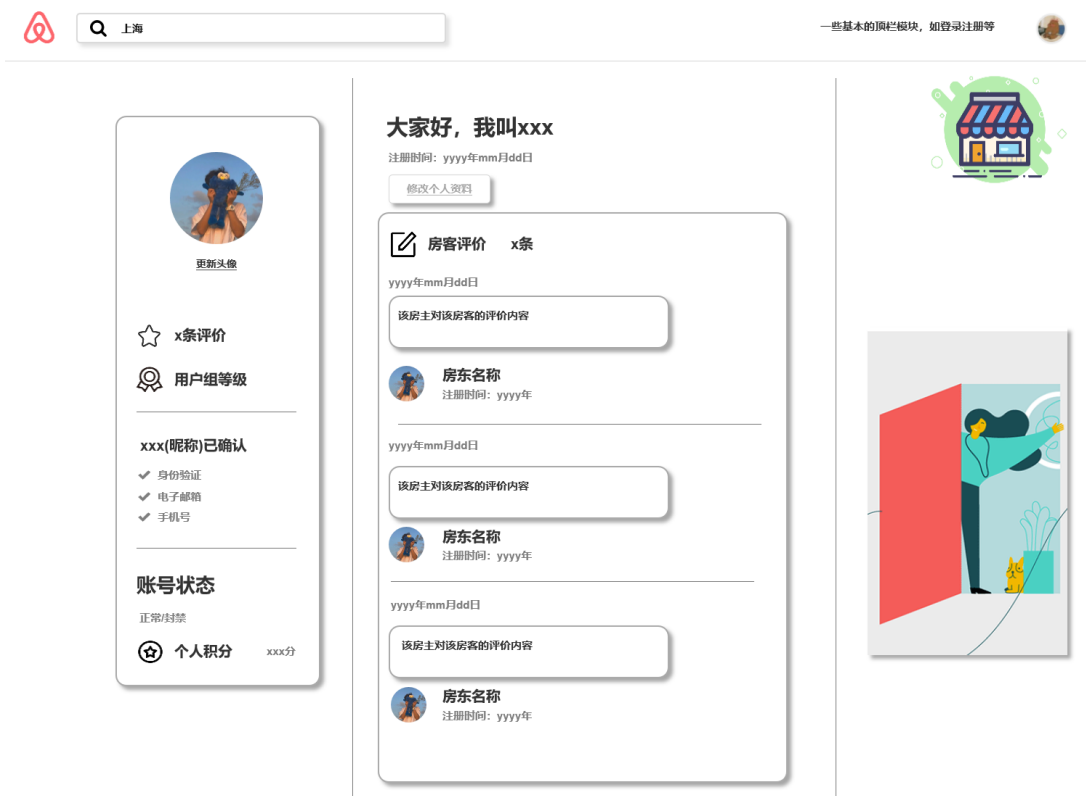
✓

注册成功

页面将在3秒内跳转

用户信息子系统

1. 个人信息界面：



在本界面中，用户可以查看个人的基本信息，包括：头像、昵称和注册时间等。除此之外，房客对该房东的所有评价也可以在该界面看到。

## 2. 收藏夹界面：





### 新的愿望单

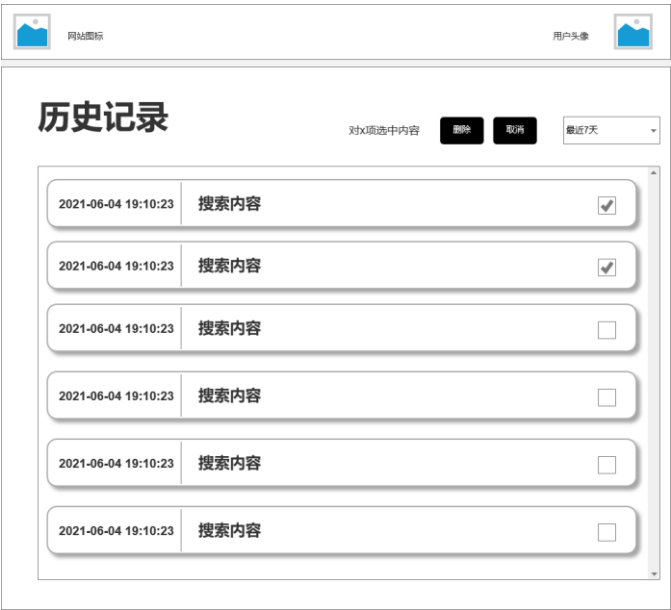
输入名称

不能超过50个字符

创建

用户可以在该界面查看自己的收藏夹，并且可以进行创建收藏夹和删除收藏夹操作。此外，用户可以查看指定收藏夹内的全部房源，并对其进行管理。

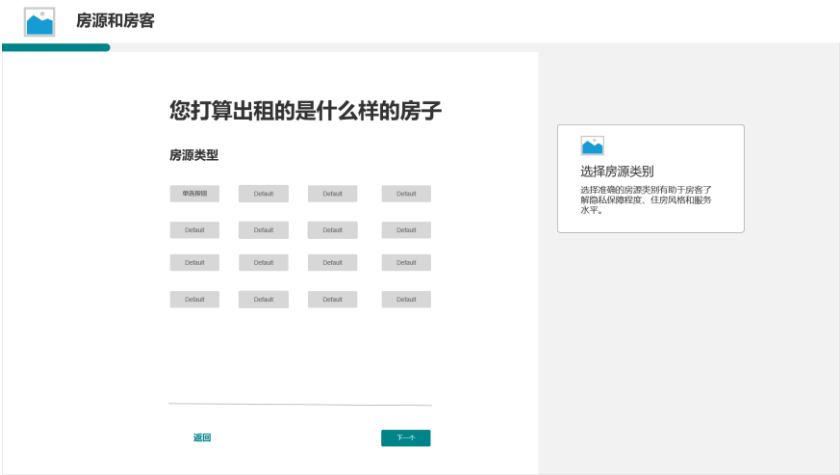
### 3. 查看历史记录:



用户可以在该界面查看自己所浏览过的房源信息，并对历史信息进行删除等操作。

房源管理系统

1. 创建房源：



 房源和房客

您的房源可以住几位房客？

请确保您准备了足够的床位，让所有房客都能舒适入住。

最多可容纳房客人数 

-

0

+

有几个卧室？

下拉菜单

有几张床？

床铺数量 

-

0

+

床型信息

我们为每个卧室提供3种类型，帮助您快速了解床型信息。

卧室0

选择

卧室1

选择

卧室2

选择

返回

下一步

房东在点击“创建新房源”按钮后，即开始创建新的房源。他需要依次选择房源类型，输入卧室数量、床的数量、床型信息，并在最后提交房源申请。

## 选购子系统

### 1. 查看房源界面：



搜索

历史足迹 注册 登录



详情 评价 可订日期 位置 须知

成都市 整租酒店式公寓

【请谈交通】你该去哪儿啊？到二仙桥，什么到二仙桥？我是说你该走哪道？走成华大道，什么成华大道！你这车子能拉吗？只能拉一点点，我是问能不能拉？能，只能拉一点不能拉多了

1间卧室 1张双人床 1个浴室 可住2人

房东昵称  
114514条评价

房东的描述房东的描述房东的描述房东的描述房东的描述房东的描述房东的描述房东的描述房东的描述房东的描述房东的描述房东的描述房东的描述房东的描述房东的描述房东的描述

卧室1

1张 双人床

入住退房 入住时间14:00-23:00, 退房时间0:00

房客评价

★★★★★ 114514条评价

用户1昵称  
2021年5月

你租那么大干什么呀 声音不能小一点？嫌你的态度不好好一点嘛 我为什么不好呢？那我为什么来干嘛？大家晚上好嘛，那你去干嘛嘛 去你！去！你两呢？我给吓唬呢？ 我！你！

用户2昵称  
2021年5月

你租我们不要紧了 我一下我告诉你该去哪儿？到二仙桥！不要，你走哪道嘛，你该124514个走成华大道！！你傻呀，我说你睡哪道了呀，你睡哪道？二仙桥对吧？对，二仙桥要走这个成华大道 我不清楚这个我睡哪道这个车子能不能走？能走就走嘛！嘛！到二仙桥 你睡我们不要乱，冷静一点，我是说你的车子 电瓶车。。，我知道是电瓶车

可订日期

2021年6月							2021年7月						
一	二	三	四	五	六	日	一	二	三	四	五	六	日
		1	2	3	4	5	6	7	8	9	10	11	12
7	8	9	10	11	12	13	14	15	16	17	18	19	20
14	15	16	17	18	19	20	21	22	23	24	25	26	27
21	22	23	24	25	26	27	28	29	30	31			
28	29	30											

房源位置

中国 四川省 成都市成华区 二仙桥

具体位置请见xxxxxx

周边信息

附近：二仙桥、东郊记忆、电子公园

地铁站：理工大学 (约900米)、二仙桥 (约1.2公里)

机场火车站：双流国际机场 (约20公里)



入住须知

房屋守则

本房屋仅供  
婴儿可以入住(2岁以下)  
不允许养宠物和其他活动  
禁止吸烟  
退房儿童(2-12岁)

安全须知

在签订租赁合同期间，我们要求所有租客和租客查看并遵守  
租赁合同中关于安全和消防的相关规定和规定，并遵守相关规定  
并签署一份《安全责任书》  
凡入住者，请务必遵守相关规定

¥23333-¥

114514条评价

入住日期 > 退房日期

人数  
2人

查看可订状态

取消预订



用户可以在该界面查看到所要预订房源的基本信息，包括当前房源的基本信息、可订日期、房客对该房源的评价、房源所处位置等。

2. 支付界面：



用户在选择心仪的房源房间后，即可进行下单操作。在下单界面，用户可以选择要使用的礼券进行折扣，并选择要使用的付款方式。点击“确认并支付”后，即跳转到支付宝支付界面。

订单子系统

1. 订单界面：

## 用户订单

[历史足迹](#)

默认排序

[全部订单](#)

[待支付的订单](#)

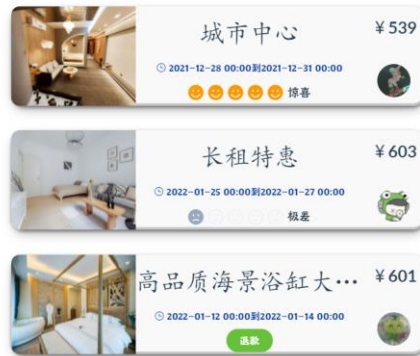
[待进行的订单](#)

[进行中的订单](#)

[待评价的订单](#)

[已完成的订单](#)

[举报中的订单](#)



用户可以在该界面查看所有订单信息，包括待支付订单、待进行订单、进行中订单、待评价订单、已完成订单和举报中的订单

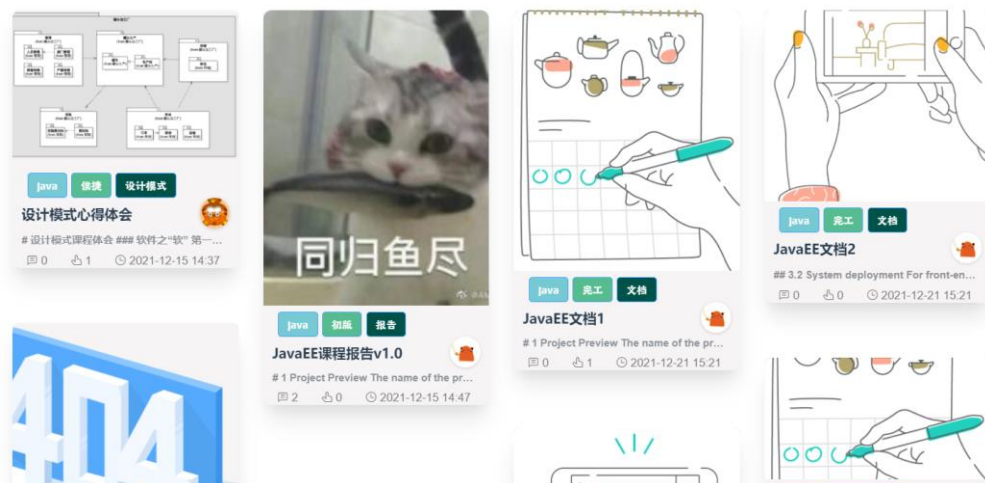
### 2. 评价订单：



针对已经完成的订单，用户可以进行评价操作。

## 论坛子系统

### 1. 广场界面：



用户在该界面可以查看到与指定标签相关的全部帖子的基本信息，包括帖子封面图、标题、内容、评论数、点赞数、发帖时间和发帖人头像等。

## 2. 帖子界面：



用户可以在该界面查看帖子的详细内容，包括标题、全部正文，所有评论。此外，用户还可以进行点赞和回复操作。

2.5 数据库设计

2.5.1 数据库逻辑设计

在数据建模阶段，我们已经设计了每个子系统的类图和序列图，结合需求分析阶段的数据需求分析，我们详细设计了整体系统数据库的逻辑结构：首先分析系统的实体和关系，根据实体和关系构建系统整体 E-R 图。详细说明如下：

2.5.1.1 数据库实体设计

在详细设计数据库的物理存储模型之前，我们首先需要设计数据库的各个实体，列举如下：

<entity>顾客：

实体名称	顾客
标识符	customer
描述	记录顾客的账号及个人信息
主键	顾客 ID
数据项（属性）	顾客 ID，顾客昵称、密码、头像、创建时间、手机号、手机前缀码、邮箱、性别、生日、用户组等级、个人积分、用户封禁状态

<entity>房东：

实体名称	房东
标识符	host
描述	记录房东的账号以及个人信息
主键	房东 ID
数据项（属性）	房东 ID，用户名、密码、头像、邮箱、创建时间、手机号前缀码、手机号、身份证号、真实姓名、性别、信誉积分、封禁状态

<entity>订单：

实体名称	订单
标识符	order
描述	记录顾客订购一次房源的购买记录。用户完成一次付款后，生成一项新的订单实体
主键	订单 ID

数据项（属性）	订单 ID、订单生成时间、订单人数、订单金额
---------	------------------------

**<entity>帖子：**

实体名称	帖子
标识符	post
描述	记录用户（顾客和房东）发布的帖子的基本信息。用户发布一个帖子后生成新的订单实体
主键	帖子 ID
数据项（属性）	帖子 ID、帖子发布时间、帖子内容、帖子主题、帖子点赞数、帖子标签、帖子图片

**<entity>帖子回复：**

实体名称	帖子回复
标识符	reply
描述	记录用户（顾客和房东）对某个帖子的回复的基本信息
主键	回复 ID
数据项（属性）	回复 ID、回复时间、回复内容、点赞数

**<entity>房源：**

实体名称	房源
标识符	stay
描述	记录房东所发布的房源的基本信息。房东发布一个房源后将生成一个新的房源实体
主键	房源 ID
数据项（属性）	房源 ID、房源名称、房源地址、房源经度、房源纬度、房源最大容纳人数、房间数量、床数量、公共卫生间数量、公共浴室数量、无障碍设施数量、房源特色、房源入住时间、房源结束入住时间、最小入住人数、最大入住人数、房源状态、房源数量、房源评价分数

**<entity>房间：**

实体名称	房间
标识符	room
描述	记录房源中的一个房间的基本信息
主键	房间 ID
数据项（属性）	房间 ID、房间价格、房间面积、房间浴室数量、房间图片

**<entity>房源类型：**

实体名称	房源类型
标识符	stay_type

描述	记录各种房源类型的信息，包括房源类型的名字，特色的描述
主键	房源类型名称
数据项（属性）	房源类型名称、房源特色描述

**<entity>床：**

实体名称	床
标识符	bed_type
描述	记录房间内床类型的基本信息
主键	床类型名称
数据项（属性）	床类型名称、床的宽度、床容纳人数

**<entity>房源标签：**

实体名称	房源标签
标识符	label_name
描述	记录房源的类型标签
主键	房源类型标签名称
数据项（属性）	房源类型标签名称、标签分类

**<entity>周边信息：**

实体名称	周边信息
标识符	peripheral
描述	记录房源的周边景点的基本信息。每个房源会根据地理信息同时存储对应的周边景点基本信息
主键	周边信息 ID
数据项（属性）	周边信息 ID、周边信息名称、周边信息类型、周边信息详细地址

**<entity>收藏夹：**

实体名称	收藏夹
标识符	Favorite_directory
描述	记录用户收藏房源的收藏夹信息。用户创建收藏夹时会生成新的收藏夹实体
主键	收藏夹 ID
数据项（属性）	收藏夹 ID、收藏夹名称

**<entity>礼券：**

实体名称	礼券
标识符	coupon
描述	记录用户当前拥有的所有礼券的礼券基本信息。礼券会定时发放给所属特定用户组的用户

主键	礼券 ID
数据项（属性）	礼券 ID、礼券可用开始日期、礼券截止日期

#### <entity>礼券类型：

实体名称	礼券类型
标识符	coupon_type
描述	记录礼券具备的类型信息。一个礼券属于一种特定礼券类型
主键	礼券类型 ID
数据项（属性）	礼券类型 ID、礼券金额、礼券最低使用金额、礼券名称

#### <entity>用户组信息：

实体名称	用户组信息
标识符	customer_group
描述	记录用户组的基本信息。一个用户属于一个特定的用户组类型中
主键	用户组等级 ID
数据项（属性）	用户组等级 ID、用户组等级积分

#### <entity>房东组信息：

实体名称	房东组信息
标识符	host_group
描述	记录房东组的基本信息。一个房东属于一个特定的房东组类型中
主键	房东组等级 ID
数据项（属性）	房东组等级 ID、房东组等级积分

#### <entity>顾客评论

实体名称	顾客评论
标识符	customer_comment
描述	记录顾客对于一个已完成订单的评论信息。用户对已经完成的订单添加评论即生成新的顾客评论实体
主键	顾客评论 ID
数据项（属性）	顾客评论 ID、顾客评论内容、顾客评论分数、顾客评论时间

#### <entity>房东评论

实体名称	房东评论
标识符	host_comment
描述	记录房东对于一个已完成订单的评论信息。房东对已经完成的订单添加评论即生成新的房东评论实体
主键	房东评论 ID
数据项（属性）	房东评论 ID、房东评论内容、房东评论分数、房东评论时间

### <entity>管理员

实体名称	管理员
标识符	administrator
描述	记录管理员的基本信息
主键	管理员 ID
数据项（属性）	管理员 ID、管理员名称、管理员头像、管理员密码、管理员注册时间、管理员手机号

### <entity>订单举报

实体名称	订单举报
标识符	order_report
描述	记录对某个订单的举报信息
主键	订单举报 ID
数据项（属性）	订单举报 ID、订单举报时间、订单举报理由、订单处理结果标识

### 2.5.1.2 数据库关系设计

根据上述实体，我们考虑了每个实体之间的关系，由于实体中有强实体集和弱实体集，因此各实体间的关系也应慎重考虑，为后面物理模型的建立和范式的满足做好准备。下面是数据库中的联系集示意表：

联系集名称	联系类型	描述
房东和房源的拥有关系	一对多	一个房东可以拥有多个房源，而一个房源只能被一个房东
房源与周边地点的临近关系	多对多	一个房源附近有多个周边地点，一个周边地点附近有多个房源
管理员与房源的审核关系	一对多	一个管理员可以审核多个房源，而一个房源只能由一个管理员审核
顾客评论和订单的关系	一对一	每个订单顾客只能评论一次
房东评论和订单的关系	一对一	每个订单房东只能评论一次
房源和房间的关系	一对多	一个房源可以拥有多个房间，而每个房间只属于一个房源
房间和床的关系	多对多	每个房间可以有多种床
房间和房间照片的关系	一对多	每个房间可以拥有多张照片
房源和房源类型的关系	一对多	每个房源只能属于一个房源类型，而每一个房源类型可以有不同的房源
订单和订单举报之间的关系	一对一	每个订单顾客只能举报一次
管理员和订单举报之间的关系	一对多	每个管理员可以受理多个举报，而每个订单举报只能被一个管理员处理



顾客和用户组之间的从属关系	多对一	每个顾客只属于一个用户组，而一个用户组可以有多个顾客
房东和用户组之间的从属关系	多对一	每个房东只属于一个房东组，而一个房东组可以有多个房东
顾客和房源之间的收藏关系	多对多	一个顾客可以收藏多个房源，而一个房源可以被多个顾客收藏
订单和房间之间的契约关系	多对多	一个订单种用户可以选择多个房间；而一个房间又对应多个订单
顾客和礼券之间的拥有关系	一对多	一个顾客可以用于多张礼券，而一张礼券只属于一个用户
礼券和订单的关系	一对一	一张礼券只能被一个订单使用，一个订单只能使用一张礼券
用户和收藏夹的关系	一对多	一个用户可以有多个收藏夹，而一个收藏夹只能属于一个用户
收藏夹和房源的关系	多对多	一个收藏夹可以收藏多个房源，一个房源可以被多个收藏夹收藏
房源和房源标签的关系	多对多	一个房源可以有多个标签，而一个标签可以标记多个房源
顾客和帖子的发布关系	一对多	一个顾客可以发布多个帖子，而一个帖子只能被一个顾客发布
顾客和帖子的点赞关系	多对多	一个顾客可以点赞多个帖子，而一个帖子可以被多个顾客点赞
顾客和帖子回复的拥有关系	一对多	一个顾客可以回复多次，而一个帖子回复只能被一个顾客拥有
顾客和帖子回复的点赞关系	多对多	一个顾客可以点赞多个回复，一个回复可以被多个顾客点赞
帖子回复之间的从属关系	一对多	一个帖子回复可以拥有多个回复，而一个回复只能从属于一个帖子回复
帖子与帖子回复的关系	一对多	一个帖子可以有多个帖子回复，而一个帖子回复只能属于一个帖子
顾客、顾客、管理员的举报关系	多对多对一	一个顾客可以举报多个顾客，一个顾客也可以被多个顾客举报，而一个举报只能被一个管理员受理
礼券类型和用户组之间的分配关系	多对多	一种礼券类型可以被多个用户组分配，而一个用户组可以分配多种礼券类型

### 2.5.1.3 数据库 E-R 图设计

根据上述数据库的实体集和联系集的设计，我们设计出了如下的 E-R 图，其中实体集用矩形标识，标识联系集用双线菱形表示。



性。

- ◆ 对于一对多的联系而言：转换为一个独立的关系模式，也可以与多端对应的关系模式合并。如果转换为一个独立的关系模式，则与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为 N 端实体的码。
- ◆ 对于多对多的联系而言：转换为一个关系模式。与该联系相连的各实体的码为各实体码的组合。
- ◆ 对于三个或三个以上的多元联系而言：转换成一个关系模式。与该多元联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为各实体码的组合。
- ◆ 具有相同键的关系模式可以进行合并

根据以上原则，我们将 E-R 转换为了如下的数据库结构，现将所有的数据库表的名称、标识符、数据项、记录、存储要求、存储类型、索引、存储区域以表格的形式列举如下：

<table>顾客 customer 表

属性标识符	存储类型	存储要求	索引	描述
customer_id	bigint	主键		顾客 id
customer_name	varchar(40)	not null		顾客昵称
customer_password	varchar(100)	not null		顾客密码
customer_phone	char(11)	not null	unique	顾客手机号
customer_email	varchar(64)		unique	顾客邮箱
customer_create_time	timestamp	not null		顾客创建时间
customer_avatar_link	varchar(256)	not null		顾客头像 URL
customer_gender	varchar(1)	not null either f or m		顾客性别

customer_birthday	date		顾客生日
customer_state	decimal(10,0)		顾客账户状态
customer_degree	decimal(10,2)		顾客积分
customer_mood	decimal(5,0)	value between 0 and 7	顾客心情
customer_level	int	foreign_key(customer_group, customer_level)	顾客级别

记录示例：

customer(232363463778373, “root” , 9Q4QCS/7mL5br83l4MurJYkIL6DvOs7, 19121765585,null, +86,2021-12-12 00:00:00, “m” , “1999-12-12” ,2,1)

<table>房东 host 表

属性标识符	存储类型	存储要求	索引	描述
host_id	int	主键		房东 id
host_create_time	timestamp			房东创建时间
host_resident_id	varchar(18)	not null	unique	房东身份证号
host_real_name	varchar(40)	not null		房东真实姓名
customer_score	int	not null		房东积分
host_state	int	not null		房东账户状态
host_level	int	primary_key(host.host_level)	normal	房东等级
customer_id	bigint	not null primary_key(customer_id)	normal	房东对应的顾客 ID

记录示例：

host(2, 2021-12-07 00:00:00, 梁乔, 0, 0, 1, 232364986843205)

<table>房源 stay 表

属性标识符	存储类型	存储要求	索引	描述
stay_id	bigint	主键		房源 id
host_id	int	not null foreign_key(host_id, stay_id)	normal	房东 id
stay_name	varchar(256)	not null		房源名称
stay_type_name	varchar(32)	foreign key(stay_type, stay_type_name)	normal	房源类型名称
detailed_address	varchar(256)	not null		房源详细地址
longitude	decimal(8,4)	not null		房源经度
latitude	decimal(8,4)	not null		房源维度
stay_capacity	int	not null		房源容纳人数
room_amount	int	not null		房源房间数

bed_amount	int	not null	床数
public_toilet	int	not null	公共卫生间数
public_bathroom	int	not null	公共浴室数
non_barrier_facility	int	not null	无障碍设施数
characteristic	varchar(1000)	not null	房源特色描述
check_in_time	time	not null	房源最早入住时间
check_out_time	time	not null	房源最晚入住时间
duration_min	decimal(4,0)	not null	房源最少入住天数
durration_max	decimal(4,0)	not null	房源最大入住天数
stay_status	decimal	not null within 1,2,3,4	房源状态
comment_amount	int	not null	房源评价数
comment_score	decimal(3,2)	not null	房源评分

记录示例：  
stay(234193492865157, 5, “近景区”，“茅屋”，“上海市奉贤区庄行镇浦家宅”， 121.3920, 30.8797, 0, 1, 1, 1, 1, 1, “近景区|近地铁|市中心”, 08:00:00, 23:00:00, 1, 1, 2, 0, 5.0)

<table>帖子 post 表

属性标识符	存储类型	存储要求	索引	描述
post_id	bigint	主键		帖子 id
post_time	timestamp	not null		帖子
post_content	varchar(8000)	not null		帖子内容
post_theme	varcga(40)	not null		帖子主题
reply_count	int	not null		帖子回复数
like_count	int	not null default 0		帖子点赞数
customer_id	bigint	not null foreign_key(custom er.customer_id)	normal	顾客 id

记录示例：  
post(234199482912837, 2021-12-12 12:00:00, “绝对不虚此行啊! ”， “论坛精品帖子集锦”, 8, 2, 232363463778373)

<table>管理员 administrator 表

属性标识符	存储类型	存储要求	索引	描述
admin_id	int	主键		管理员 id
admin_name	varchar(64)	not null		管理员名称
admin_password	varchar(100)	not null		管理员密码

admin_avatr_link	varchar	not null	管理员头像
admin_create_time	timestamp	not null	管理员创建时间
admin_tel	char(11)	not null	管理员手机号

记录示例：

administrator(1, “xny” , “xnyhq” ,  
“https://guisu.oss-cn-shanghai.aliyuncs.com/img/customer\_img.png” , 2021-11-18 13: 51: 31 ,  
1332343254536)

**<table>床 bed 表**

属性标识符	存储类型	存储要求	索引	描述
bed_type	varchar(100)	主键		床类型名称
bed_width	decimal(2,1)	not null		床宽
person_amount	int	not null >0		床容纳人数

记录示例：

bed( “1.4 米宽双人床” , 1.4, 2)

**<table>礼券 coupon 表**

属性标识符	存储类型	存储要求	索引	描述
coupon_id	bigint	主键		礼券 id
coupon_type_id	int	foreign_key(customer.custom er_id)	normal	礼券类型 id
customer_id	bigint	foreign_key(coupon_type.cou pon_type_id)	normal	顾客 id
coupon_start_date	date	not null		礼券有效期开始时间
coupon_end_date	date	not null		礼券有效期结束时间
coupon_status	int	not null		礼券状态

记录示例：

coupon(7, 1, 234190415450181, 2021-12-10, 2021-12-31, 1)

**<table>礼券类型 coupon\_type 表**

属性标识符	存储类型	存储要求	索引	描述
coupon_type_id	int	主键		礼券类型 id
coupon_amont	decimal(8,2)	not null		礼券金额
coupon_limit	decimal(8,2)	not null		礼券使用最低金额
coupon_name	varchar(64)	not null		礼券名称

记录示例：

coupon\_type(1, 20.00, 200.00, “暖心冬游”)

<table>顾客评论 customer\_comment 表

属性标识符	存储类型	存储要求	索引	描述
customer_comment_id	biginy	主键		顾客评论 id
order_id	bigint	not null foreign_key (order.order_id)	normal	订单 id
customer_comment_time	timestamp	not null		顾客评论时间
customer_comment_content	varchar(400)	not null		顾客评论内容
stay_score	int	not null		房源评分

记录示例:

customer\_comment(11, 234225349517381, 2021-12-18 12:12:23, “用户 7 天内未做出评价, 系统默认好评!”, 5)

<table>用户组 customer\_group 表

属性标识符	存储类型	存储要求	索引	描述
customer_level	int	主键		用户组等级数
customer_level_name	varchar(30)	not null		用户组等级名称
customer_level_degree	bigint	not null		用户等级积分

记录示例:

customer\_group(1, “人在旅途”, 0)

<table>用户组礼券 customer\_group\_coupon 表

属性标识符	存储类型	存储要求	索引	描述
customer_level	int	主键 foreign_key(customer_group.customer_level)		用户组等级
coupon_type_id	int	主键 foreign_key(coupon_type.coupon_type_id)	normal	礼券类型 id
coupon_amount	int			礼券数量

记录示例:

(1, 2, 1)

<table>收藏夹 favortite\_firectory 表

属性标识符	存储类型	存储要求	索引	描述
favorite_directory_id	int	主键		收藏夹 id
cusomer_id	bigint	not null foreign_key(c	normal	顾客 id

	ustomer.customer_id)		
name	varchar(50)	not null	顾客名称

记录示例：  
favorite\_directory(10294, 234203956142213)

<table>房东评论 host\_comment 表

属性标识符	存储类型	存储要求	索引	描述
host_comment_id	bigint	主键		房东评论 id
order_id	bigint	not null foreign_key (order.order_id)	normal	订单 id
host_comment_time	timestamp	not null		房东评价时间
host_comment_content	varchar(400)	not null		房东评价内容
customer_score	int	not null		顾客评分

记录示例：  
host\_comment(1, 2, 2021-12-12 20:00:00, “默认好评”, 5.0)

<table>房东组 host\_group 表

属性标识符	存储类型	存储要求	索引	描述
host_level	int	主键		房东组等级数
host_level_name	barchar(32)	not null		房东组名称
host_level_degree	bigint	not null		房东组积分

记录示例：  
host\_group(1, “初学乍练”, 0)

<table>标签 label 表

属性标识符	存储类型	存储要求	索引	描述
label_name	varchar(20)	主键		房源标签名称
label_type	int	not null		房源标签类型

记录示例：  
label(“亲子乐园”, 0)

<table>房源周边 near 表

属性标识符	存储类型	存储要求	索引	描述
peripheral_id	int	主键 foreign_key(peripheral.peripheral_id)		周边地点 id
stay_id	bigint	not null foreign_key(stay.stay_id)	normal	房源 id



distance	decimal(5,2)	not null	周边地点与房源的距离
----------	--------------	----------	------------

记录示例：

near(1231425, 2, 123.00)

**<table>订单 order 表**

属性标识符	存储类型	存储要求	索引	描述
order_id	bigint	主键		订单 id
customer_id	bigint	not null foreign_key(customer.customer_id)	normal	顾客 id
order_time	timestamp			订单创建时间
member_amount	decimal(1,0)	not null		订单人数
total_cost	decimal(15,2)			订单总价
order_status	int	not null		订单状态

记录示例：

order(5349517381, 23419041545018, 2021-12-12 22:45:25, 1, 584.00, 5)

**<table>订单举报 order\_report 表**

属性标识符	存储类型	存储要求	索引	描述
report_id	bigint	主键		订单举报 id
order_id	bigint	not null foreign_key(order.order_id)	normal	订单 id
report_time	timestamp	not null		举报时间
report_reason	varchar(400)	not null		订单举报理由
is_dealt	int	not null		举报是否处理
admin_id	int			管理员 id
deal_time	timestamp			处理时间
reply	varchar(400)			处理回复
reply_flag	int			处理结果

记录示例：

order\_report(3, 237296538423365, 2021-12-21 15:05:06, “虚假宣传，住房和实际完全不符！卫生极其差劲！快点倒闭吧！”, 1, null, null, null, 1)

**<table>房源订单 order\_stay 表**

属性标识符	存储类型	存储要求	索引	描述
order_id	bigint	主键 foreign_key(order.order_id)		订单 id

stay_id	bigint	主键 foreign_key(stay. stay_id)	normal	房源 id
room_id	int	主键 foreign_key(roo m.room_id)	normal	房间 id
start_time	datetime	not null		订单开始时间
end_time	datetime	not null		订单结束时间
money_amount	decimal(15,2)	not null		金额

记录示例:

order\_stay(23422534951738, 234204458795141, 1, 2021-12-14 08:00:00, 2021-12-15 08:00:00, 599.00)

**<table>周边地点 peripheral 表**

属性标识符	存储类型	存储要求	索引	描述
peripheral_id	int	主键		周边地点 id
peripheral_name	varchar(100)	主键	normal	周边地点名称
peripheral_class	varchar(20)	主键	normal	周边地点类型
peripheral_popularity	decimal(10,0)			周边地点热度
detailed_address	varchar(512)			周边地点详细地址

记录示例:

peripheral(10191, “太古里”, “商业区”, 20, 四川省成都市锦江区)

**<table>帖子图片 post\_img 表**

属性标识符	存储类型	存储要求	索引	描述
post_id	bigint	主键 foreign_key(pos t.post_id)		帖子 id
post_img_link	varchar(256)	主键		帖子图片 url

记录示例:

post\_img(234199482912837,  
“http://tongjihohome.oss-cn-shanghai.aliyuncs.com/postImg/232363463778373/23419948372794  
1.png?Expires=4792914010&OSSAccessKeyId=LTAI5t9L46CRZ9pYLUUdSS8b&Signature=FxvS  
w4rAXdqMnwlw7pVZgj45ZQQ%3D” )

**<table>帖子点赞 post\_like 表**

属性标识符	存储类型	存储要求	索引	描述
post_id	bigint	主键 foreign_key(pos t.post_id)		帖子 id
customer_id	bigint	主键 foreign_key(cust omer.customer_id)	normal	顾客 id

记录示例:

post\_like(23419948291283, 232363463778373)

**<table>帖子回复 post\_reply 表**

属性标识符	存储类型	存储要求	索引	描述
reply_id	bigint	主键		回复 id
post_id	bigint	not null foreign_key (post.post_id)	normal	帖子 id
customer_id	bigint	not null foreign_key(c ustome.customer_id)	normal	顾客 id
reply_time	bigint	not null		回复时间
reply_content	varchar(200)	not null		回复内容
reply_like_count	int	not null		回复点赞数
pre_reply_id	bigint	foreign_key(post_repl y.reply_id)	normal	父级回复 id

记录示例:

post\_reply(233172461629509, 2, 233172461629509, 2021-12-07 00:00:00, “haha” , 0, 2)

**<table>帖子回复点赞 post\_reply\_like 表**

属性标识符	存储类型	存储要求	索引	描述
reply_id	bigint	主键 foreign_key(post _reply.reply_id)		回复 id
customer_id	bigint	主键 foreign_key(cust omer.customer_id)	normal	顾客 id

记录示例:

post\_reply\_like(233505290817605, 233505290817602)

**<table>帖子举报 post\_report 表**

属性标识符	存储类型	存储要求	索引	描述
report_customer_id	bigint	主键 foreign_key(cust omer.customer_id)	unique	举报顾客 id
be_reported_customer_id	bigint	主键 foreign_key(cust omer.customer_id)	normal	被举报顾客 id
report_reason	varchar(200)	not null		举报原因描述
report_time	timestamp	not null		举报时间
is_dealt	tinyint(1)	not null		是否处理
admin_id	int			管理员 id
deal_time	timestamp			处理时间
reply	varchar(400)			处理回复
reply_flag	int			处理结果

记录示例：

post\_report(234190415450181, 235889425367109, “看不懂, 发的什么呀? 和你的泥巴小院有关系么?” , 2021-12-21 20:10:32, 1, 1, 2021-12-18 22:29:11, 11, 0)

<table>帖子关联房源 post\_stay 表

属性标识符	存储类型	存储要求	索引	描述
post_id	bigint	主键 foreign_key(pos t.post_id)		帖子 id
stay_id	bigint	主键 foreign_key(stay. stay_id)	normal	房源 id

记录示例：

post\_stay(234190415450181, 234193492865157)

<table>帖子标签 post\_tag 表

属性标识符	存储类型	存储要求	索引	描述
post_id	bigint	主键 foreign_key(pos t.post_id)		帖子 id
post_tag	varchar(20)	主键		帖子标签

记录示例：

post\_tag(2, “近地铁”)

<table>房间 room 表

属性标识符	存储类型	存储要求	索引	描述
stay_id	bigint	主键 foreign_key(stay. stay_id)	normal	房源 id
room_id	int	主键		房间 id
price	decimal(8,2)	not null		房间价格
room_area	decimal(6,2)			房间面积
bathroom_amount	int			浴室数量

记录示例：

room(234193492865157, 1, 299, 30, 1)

<table>房间床 room\_bed 表

属性标识符	存储类型	存储要求	索引	描述
stay_id	bigint	主键 foreign_key(roo m.stay_id)		房源 id
room_id	int	主键 foreign_key(roo m.room_id)		房间 id
bed_type	varchar(100)	主键 foreign_key(bed. bed_type)	normal	床类型

	bed_type)	
bed_amount	int	not null 床数

记录示例：

room\_bed(234193492865157, 1, “1 米宽单人床” , 1)

**<table>房间照片 room\_photo 表**

属性标识符	存储类型	存储要求	索引	描述
stay_id	bigint	主键 foreign_key(roo		房源 id
		m.stay_id)		
room_id	int	主键 foreign_key(roo		房间 id
		m.room_id)		
room_photo_link	varchar(256)	not null		房间照片 url

记录示例：

room\_photo(234193492865157, 1,  
“https://z1.muscache.cn/im/pictures/b481948c-58d4-4e35-8a39-f6701c828710.jpg” )

**<table>房源标签 stay\_label 表**

属性标识符	存储类型	存储要求	索引	描述
stay_id	bigint	主键 foreign_key(stay.		房源 id
		stay_id)		
label_name	varchar(20)	主键 foreign_keylabel.		房源标签名称
		label_name()		

记录示例：

stay\_label(234193492865157, “环境宜人” )

**<table>房源类型 stay\_type 表**

属性标识符	存储类型	存储要求	索引	描述
stay_type_name	varchar(20)	主键 foreign_key(stay.		房源类型名称
		stay_id)		
characteristic	varchar(100)	not null		房源特色描述

记录示例：

stay\_type( “乡村小屋” , “尊贵人的享受，你值得拥有。” )

**<table>verification 审核表**

属性标识符	存储类型	存储要求	索引	描述
verification_id	bigint	主键		审核 id
stay_id	bigint	not null foreign_key(s	normal	房源 id
		tay.stay_id)		
admin_id	int			管理员 id



每一项都是不可分割的原子项。

- ◆ 第二范式 (2NF)。归宿数据库表中的每个实例或记录均可以被唯一地区分。
- ◆ 第三范式 (3NF)。归宿的关系中每个关系都不包含已在其他关系出现的非主关键字信息，任何非主属性不得传递依赖于主属性。

#### 2.5.3.2 存储查询优化

随着系统发布日期的延长，数据库中的数据项将会越来越多，负载也会越来越远大，较慢的查询效率必然会影响整体系统的响应时间。对此，我们对数据库表建立了必要的索引，以加快数据库的查询速度。对此我们制定了以下原则来建立简单索引和组合索引：

- ◆ 默认对表的外键建立索引
- ◆ 对查询中频繁出现的查询条件对应的字段建立索引
- ◆ 对于非必要的字段尽量不建立索引，以避免增加数据库的存储

除此之外，我们还对于某些复杂查询建立视图，以简化查询，方便管理。

#### 2.5.3.3 函数和触发器

由于本系统存在大量紧密关联的业务逻辑，并且需要实时更新，如当用户积分到达一定水平时用户会自动升级；用户会在一定的时间内获得相应数量的礼券；若用户未给出评价，系统会在七天内自动给出默认好评；订单状态也需要实时跟进和维护等等。这些业务逻辑在后端是很难完成的，因此我们相应设计了函数和触发器来自动维护这些功能。

## 2.6 系统出错处理设计

### 2.6.1 出错信息

本系统涉及用户的个人隐私、财产和支付等重要信息，因此对于一些严重的错误建立完备的出错处理机制和系统故障发生后的补救措施是十分重要的。对此我们列举了如下的常见出错信息和预计的系统处理方法。由于某些错误可以避免，因此对于特定的软错误而言，我们直接在前端予以提示和修正。对于一些可能人为造成的错误而言，如误删误改操作而言，我们提供完备的确认机制；对于一些关键信息，如密码等，为防止黑客攻击系统盗用损坏用户信息，系统提相应的保密和加密措施。

以下以一览表的形式说明每种可能出现的软错误和硬错误发生时，系统输出信息的形式，含义及处理方法。

错误类型	错误信息	处理方法	系统输出信息
数据库错误	数据库连接失败	尝试重连	前端输出错误提示信息
	数据库操作失败	事务回滚, 输出错误信息	前端输出更新失败信息
	数据库溢出	事务回滚, 输出错误信息	前端输出更新失败信息
TCP 连接错误	连接超时或断开	尝试重新连接, 特定时间后输出错误信息	尝试失败后输出对应的网络错误信息
系统部分自定义错误	用户名密码错误	不予登录	系统输出输入错误提示信息, 登陆失败
	身份证输入错误	不予注册	系统输出错误的身份证提示信息, 登陆失败
	验证码错误/为空	不予登录, 切换验证码	系统输出验证码错误提示信息, 登陆失败
	修改个人信息错误	不予修改	系统输出不满足条件的错误修改提示信息, 修改失败
	房源信息错误	不予发布房源, 空操作	系统标红不满足输入条件的房源信息, 发布失败, 等待修改



	帖子发布信息错误	不予发布帖子,空操作	系统标红不满足输入条件的帖子信息,发布失败,等待修改
	支付错误	系统回滚事务	系统输出支付失败信息并跳转回待支付界面
链接错误	图片链接错误	抛出异常	系统展示占位图片信息,提示图片展示失败
	外部链接错误	抛出异常	系统提示外部链接错误信息
程序错误	空指针错误	服务停止重启	系统提示内部错误并重启服务
	内部程序错误	服务重启	系统提示内部错误并重启服务
其他错误	服务器故障	无	系统提示服务器故障信息并等待人为修复重启

### 2.6.2 补救措施

由于错误类型不尽相同,错误的原因也各有差异,因此对于不同的错误我们采取不同的变通措施,列举如下:

- ◆ **对于软错误**,如用户输入非法信息,可能会导致后端数据库的操作错误和程序内部发生错误,系统直接在前端的输入或修改操作中对数据本身进行验证,分析错误类型,并给出相应的错误提示语句,从用户输入层面避免输入信息的非法,从而从一定程度上避免因输入非法带来的故障。
- ◆ **对于硬错误**,如数据库连接错误和网络连接错误而言,由于错误类型较少且原因明确,可在前端输出对应的提示信息;而对于程序内部的错误,可以在程序编写阶段设置对应的异常捕获程序和抛出异常语句,在出错时输出相应的错误语句,将服务或程序重启,避免整个业务的故障停滞。维护人员可在输出日志中查看错误信息及时修正。
- ◆ 采取适当的后备技术,如当数据库受到攻击或删库时,通过定期转储数据库,对数据库定期备份避免非法攻击带来的不可逆故障。

- ◆ 制定完备的故障恢复和重启技术，对于因程序内部错误导致的服务故障，采取服务重启并从头开始运行来保证业务逻辑的正常执行。

### 2.6.3 系统维护设计

#### 2.6.3.1 数据库维护

数据库存储了用户的全部信息，涉及到用户的安全和隐私，对于数据库的维护是重中之重。对此我们建立了完备的数据库维护方案，主要包含如下几点：

- ◆ 数据库定期备份。为防止数据库操作失误、服务器宕机、服务器数据丢失带来的严重后果，定期对数据库进行转储备份来避免数据丢失的发生。
- ◆ 定期检查数据库的一致性。数据的不一致可能会导致某些业务功能的失败和财产的流失，因此需要定期对于数据库事务的一致性和数据一致性进行检查，特别是涉及到房源订单数据的一致性检查。
- ◆ 定期删除冗余数据。由于本系统对于房源的删除机制是采用标志位的形式标识房源的删除与否，因此为了防止数据出现过多冗余，管理员需要定期删除一些不再有用的冗余数据。
- ◆ 定期查看操作日志。定期查看操作日志以检查数据库设计可能存在的缺陷，及时修复和更改。

#### 2.6.3.1 后端维护

除数据库外，后端程序的错误和失败也会严重影响整个系统业务的正常运行。因此，需要专门指定维护的监测点和模块来对后端代码进行维护。

- ◆ 异常抛出模块。对于每个微服务的服务实现类，我们均根据特定的错误情形加入相应的 try-catch 块，以保证当错误发生时不会影响到程序的正常运行。
- ◆ 微服务运行检测模块。为了能随时查看各微服务是否正常运行和运行情况，后端使用 Spring Cloud 提供的 Eureka 组件管理各微服务，可在其提供的网页来查看各微服务是否正常运行。
- ◆ 单元测试模块。为保证后端定期维护作出的修改的正确性，每个微服务中的控制类和业务实现类均加入了单元测试模块，方便快速测试和维护。