**酒店预订系统**

**详细设计描述文档**

**V3.0**

学院：南京大学软件学院

成员：曹畅 顾逸飞 黄凯文 唐鑫

完成日期：2016年12月25日

目录

[更新历史 3](#_Toc466228622)

[1 引言 4](#_Toc466228623)

[1.1 编制目的 4](#_Toc466228624)

[1.2 词汇表 4](#_Toc466228625)

[1.3 参考资料 4](#_Toc466228626)

[2 产品概述 4](#_Toc466228627)

[3 体系结构设计概述 5](#_Toc466228628)

[4 结构视角 5](#_Toc466228629)

[4.1 业务逻辑层的分解 5](#_Toc466228630)

[4.1.1 Order模块 5](#_Toc466228631)

[4.1.2 User模块 19](#_Toc466228632)

[4.1.3 Hotel模块 33](#_Toc466228633)

[4.1.4 Vip模块 47](#_Toc466228634)

[4.1.5 Integral模块 51](#_Toc466228635)

[4.1.6 Promotionbl模块 55](#_Toc466228636)

[4.2 数据层的分解 60](#_Toc466228637)

[4.2.1 OrderDataService模块 60](#_Toc466228638)

[4.2.2 UserDataService模块 63](#_Toc466228639)

[4.2.3 HotelDataService模块 66](#_Toc466228640)

[4.2.4 VipDataService模块 68](#_Toc466228641)

[4.2.5 IntegralDataService模块 69](#_Toc466228642)

[4.2.6 PromotionDataService模块 70](#_Toc466228643)

[5 依赖视角 70](#_Toc466228644)

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 全体人员 | 2016/10/29 | 详细设计初步模型 | V1.0 |
| 顾逸飞 | 2016/11/6 | 在第二次作业基础上修改 | V1.1 |
| 全体人员 | 2016/12/20 | 开发过程中不断同步更新文档，开发工作完成，文档定格 | V2.0 |
| 曹畅 | 2016/12/25 | 根据最终代码修改用户模块 | V2.1 |
| 唐鑫 | 2016/12/25 | 开发完成，整理核对文档 | V3.0 |

# 引言

## 编制目的

本报告详细完成对酒店预订系统的详细设计，达到指导后续软件构造的目的，同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

## 词汇表

|  |  |
| --- | --- |
| **缩写或单词** | **解释** |
| Order | 订单 |
| Credit | 信用值 |
| Promotion | 促销策略 |
| User | 用户 |
| Hotel | 酒店 |
| Vip | 会员 |

## 参考资料

《酒店预订系统用例文档V2.0》

《酒店预订系统需求规格说明文档V2.0》

《酒店预订系统软件体系结构描述文档V2.0》

# 产品概述

参考酒店预订系统用例文档及酒店预订系统需求规格说明文档对产品的概括描述。

随着工作和生活节奏的加快，人们的差旅出行需要迅速提升，但是传统上的线下酒店预订模式有着很多不方便之处。客户无法全面了解当地的酒店情况，选择范围大大受限，难以选择到最适合的酒店，酒店担心恶意预订，顾客担心酒店不符合预期，酒店要及时掌握预订顾客的行程变化也会比较麻烦。

为了方便人们出行中的酒店预订，一家创业公司决定开发一个互联网酒店预订系统，该系统能够全面展现任一个地域内的酒店信息，帮助顾客进行更好地比较与选择，执行信用累计制度，防范恶意预订，进行房客评价，反映酒店与描述的相符程度，让酒店可以及时了解到所有预订订单的变化动态，并且能够通过网站推广，吸引酒店和顾客的入驻。它包括一个数据集中服务器和多个客户端。数据集中服务器将所有的数据储存起来进行维护。用户通过客户端完成日常任务，客户端与数据集中服务器才是实时通信的方式完成数据交换。

# 体系结构设计概述

参考酒店预订系统软件体系结构描述文档。

主要采用分层的架构方式，采用分布式的部署方式，将数据存储在服务器端，客户端

通过RMI方式调用获取数据。

# 结构视角

## 业务逻辑层的分解

业务逻辑层的开发包图请查看体系结构设计文档。

### Order模块

1、模块概述：

Order模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

Order模块的职责和接口参见软件系统结构描述文档内对该模块的描述。

2、整体结构：

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们添加businesslogicservice.orderblservice.OrderOnUserBLService, businesslogicservice.orderblservice.OrderOnHotelBLService, businesslogicservice.orderblservice. OrderOnWebBLService接口。业务逻辑层和数据层之间添加dataservice.orderdataservice. OrderDataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加OrderOnUserController，OrderOnHotelController，OrderOnWebController这样OrderOnUserController，OrderOnHotelController，OrderOnWebController会将对订单的业务逻辑处理委托给OrderOnUser，OrderOnHotel，OrderOnWeb对象。OrderPO是作为账户信息的持久性对象被添加到设计模型中去的。

Orderbl模块的设计如图1-1、图1-2、图1-3：

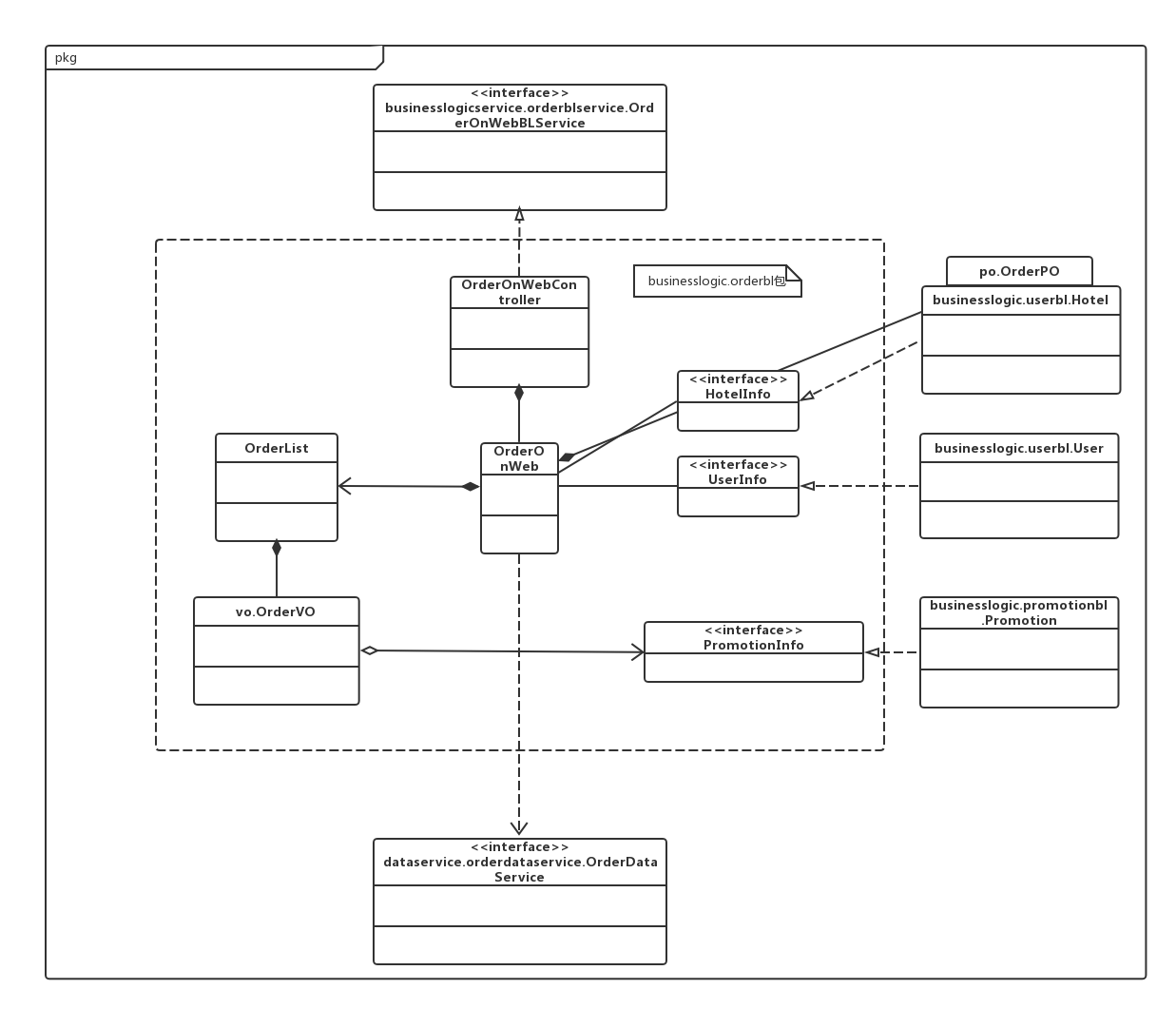


图1-1 Orderbl模块中各个类的设计

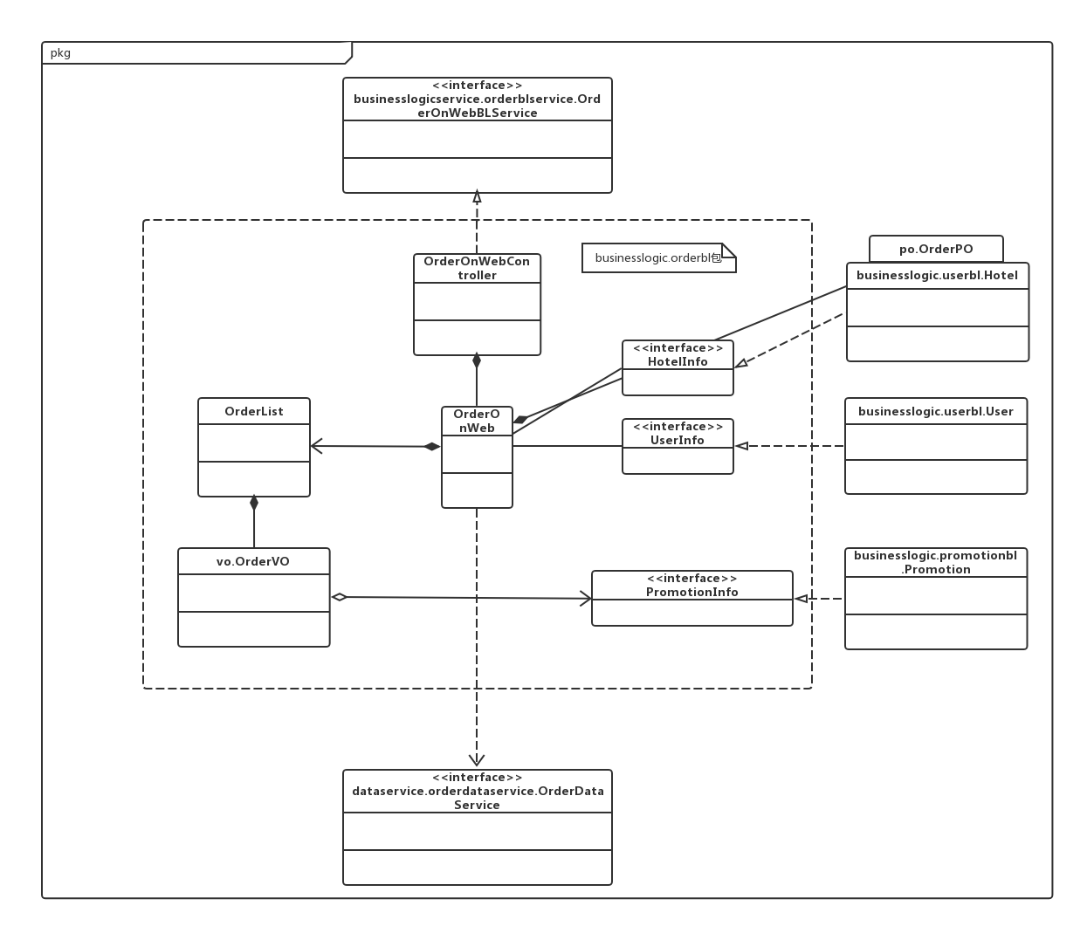


图1-2 Orderbl模块中各个类的设计

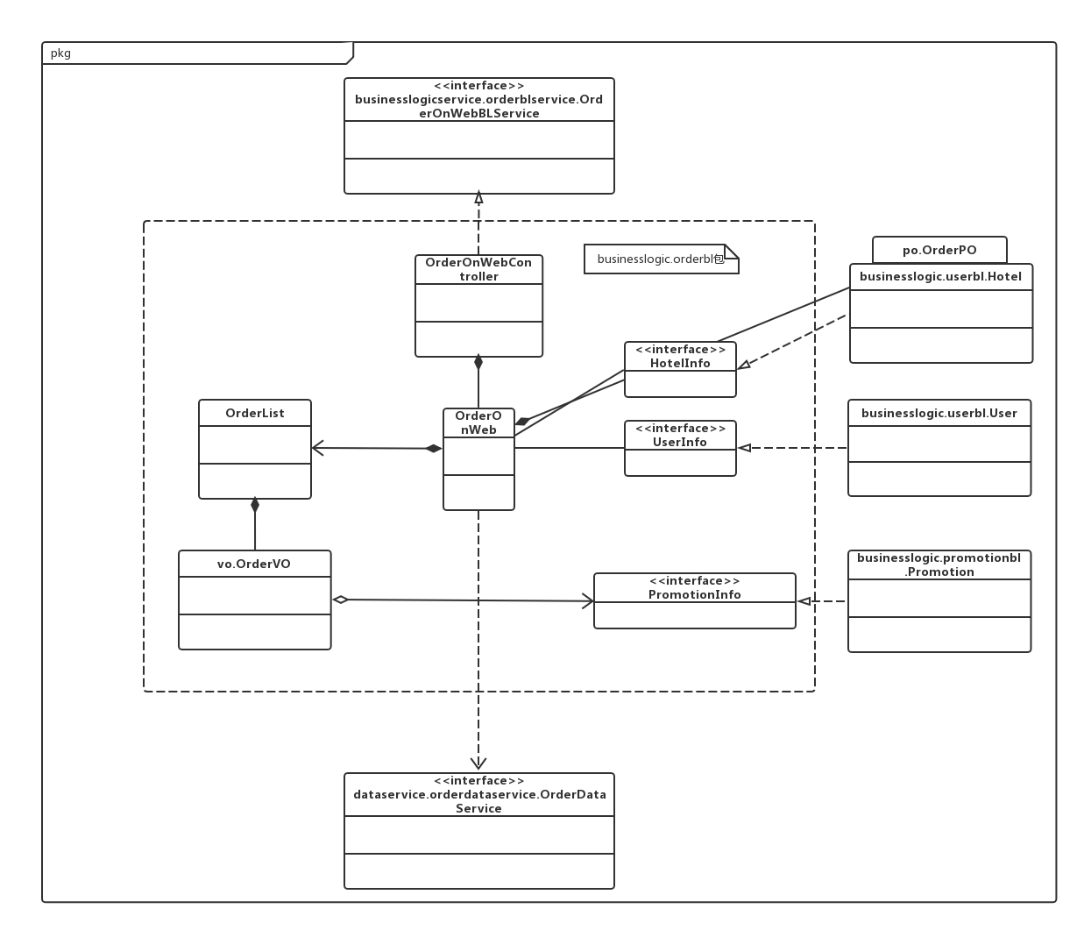


图1-3 Orderbl模块中各个类的设计

Orderbl模块各个类的职责如表1所示。

表1 Orderbl模块各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| OrderOnUserController | 负责实现对应于个人订单界面所需要的服务 |
| OrderOnHotelController | 负责实现对应于酒店订单界面所需要的服务 |
| OrderOnWebController | 负责实现对应于网站订单界面所需要的服务 |
| OrderOnUser | 个人订单对象，拥有个人订单的信息，可以帮助完成个人订单界面、个人订单详情界面、个人订单撤销界面所需要的服务 |
| OrderOnHotel | 酒店订单对象，拥有酒店订单的信息，可以帮助完成酒店订单列表界面、酒店订单详情界面、酒店订单状态修改界面所需要的服务 |
| OrderOnWeb | 网站订单对象，拥有网站订单的信息，可以帮助完成异常订单界面、异常订单详情界面、撤销异常订单界面所需要的服务 |

3、模块内部类的接口规范

OrderOnUserController和OrderOnUser的接口规范如表2-1-1和表2-1-2所示。

表2-1-1 OrderOnUserController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderOnUserController.personalOrderScan | 语法 | public ArrayList<OrderVO> personalOrderScan(String ID); |
| 前置条件 | 已经创建一个OrderOnUser领域对象 |
| 后置条件 | 调用OrderOnUser领域对象的personalOrderScan方法，持久化更新相关数据 |
| OrderOnUserController.personalOrderCancel | 语法 | public ResultMsg personalOrderCancel(OrderVO orderVO); |
| 前置条件 | 已经创建一个OrderOnUser领域对象 |
| 后置条件 | 调用OrderOnUser领域对象的personalOrderCancel方法，系统持久化更新相关数据 |
| OrderOnUserController.personalOrderDetail | 语法 | public OrderVO personalOrderDetail(StringID); |
| 前置条件 | 已经创建一个OrderOnUser领域对象 |
| 后置条件 | 调用OrderOnUser领域对象的personalOrderDetail方法 |
| OrderOnUserController.createOrder | 语法 | public ResultMsg createOrder(OrderVO orderVO) |
| 前置条件 | 已经创建一个orderVO对象 |
| 后置条件 | 调用OrderOnUser领域对象的createOrder方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderOnUser.personalOrderScan(String ID) | 查查看个人订单 | |
| OrderOnUser.personalOrderCancel(OrderVO vo) | 撤销个人订单 | |
| OrderOnUser.personalOrderDetail(String ID) | 查看个人订单详情 | |
| OrderOnUserController.createOrder（OrderVO order） | 创建个人订单 | |

表2-1-2 OrderOnUser的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderOnUser. personalOrderScan | 语法 | public ArrayList<OrderVO> personalOrderScan(String ID); |
| 前置条件 | 订单信息符合规则且齐全 |
| 后置条件 | 系统显示该订单 |
| OrderOnUser. personalOrderCancel | 语法 | public ResultMsg personalOrderCancel(OrderVO orderVO); |
| 前置条件 | 订单信息符合规则且齐全 |
| 后置条件 | 系统提示撤销订单成功 |
| OrderOnUser. personalOrderDetail | 语法 | public OrderVO hotelOrderDetail(String ID); |
| 前置条件 | 账户信息符合规则且齐全 |
| 后置条件 | 系统显示个人订单详情 |
| OrderOnUser.createOrder | 语法 | public ResultMsg createOrder(OrderVO orderVO) |
| 前置条件 | order信息符合规范且完整 |
| 后置条件 | 系统返回是否成功添加 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | |
| OrderDataService.  insert(OrderPO po) | 新增单一持久化对象 | |
| OrderDataService.  update(OrderPO po) | 更新单一持久化对象 | |
| OrderDataService.findByUserID  (String userID) | 根据用户ID查找单一持久化对象 | |
| OrderDataService.findByOrderID  (String userID) | 根据订单ID查找单一持久化对象集合 | |

OrderOnHotelController和OrderOnHotel的接口规范如表2-2-1和表2-2-2所示。

表2-2-1 OrderOnHotelController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderOnHotelController.hotelOrderScan | 语法 | public ArrayList<OrderVO> hotelOrderScan(String ID); |
| 前置条件 | 已经创建一个OrderOnHotel领域对象 |
| 后置条件 | 调用OrderOnHotel领域对象的hotelOrderScan方法，持久化更新相关数据 |
| OrderOnHotelController.hotelOrderModify | 语法 | public ResultMsg hotelOrderModify(OrderVO orderVO); |
| 前置条件 | 已经创建一个OrderOnHotel领域对象 |
| 后置条件 | 调用OrderOnHotel领域对象的hotelOrderModify方法，系统持久化更新相关数据 |
| OrderOnHotelController.hotelOrderDetail | 语法 | public OrderVO hotelOrderDetail(String ID); |
| 前置条件 | 已经有要调用的orderID |
| 后置条件 | 调用OrderOnHotel领域对象的hotelOrderDetail方法 |
| OrderOnHotelController.update | 语法 | public void update(OrderPO po); |
| 前置条件 | 已经有要更新的OrderPO |
| 后置条件 | 调用OrderOnHotel领域对象的update方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderOnHotel.hotelOrderScan(String ID) | 查看酒店订单 | |
| OrderOnHotel.hotelOrderModify(OrderVO vo) | 修改酒店订单状态 | |
| OrderOnHotel.hotelOrderDetail(String ID) | 查看酒店订单详情 | |
| OrderOnHotel.update(OrderPO po) | 更新订单详情 | |

表2-2-2 OrderOnHotel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderOnHotel. hotelOrderScan | 语法 | public ArrayList<OrderVO> hotelOrderScan(String ID); |
| 前置条件 | 订单信息符合规则且齐全 |
| 后置条件 | 系统显示该订单 |
| OrderOnHotel. hotelOrderModify | 语法 | public ResultMsg hotelOrderModify(OrderVO orderVO) |
| 前置条件 | 订单信息符合规则且齐全 |
| 后置条件 | 系统提示修改订单成功 |
| OrderOnHotel. hotelOrderDetail | 语法 | public OrderVO hotelOrderDetail(String ID) |
| 前置条件 | 账户信息符合规则且齐全 |
| 后置条件 | 系统显示酒店订单详情 |
| OrderOnHotel.update | 语法 | public void update(OrderPO po) |
| 前置条件 | 账户信息符合规则且齐全 |
| 后置条件 | 系统显示更新成功 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | |
| OrderDataService.  update(OrderPO po) | 更新单一持久化对象 | |
| OrderDataService.findByOrderID  (String userID) | 根据订单ID查找单一持久化对象集合 | |
| OrderDataService.findByHotelID  (String hotelID) | 根据酒店ID查找单一持久化对象集合 | |

OrderOnWebController和OrderOnWeb的接口规范如表2-3-1和表2-3-2所示。

表2-3-1 OrderOnWebController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderOnWebController.abnormalOrderScan | 语法 | public ArrayList<OrderOnWebVO> abnormalOrderScan(); |
| 前置条件 | 已经创建一个OrderOnWeb领域对象 |
| 后置条件 | 调用OrderOnWeb领域对象的abnormalOrderScan方法 |
| OrderOnWebController.abnormalOrderDetail | 语法 | public ArrayList<OrderOnWebVO> abnormalOrderDetail(String ID); |
| 前置条件 | 已经创建一个OrderOnWeb领域对象 |
| 后置条件 | 调用OrderOnWeb领域对象的abnormalOrderDetail方法 |
| OrderOnWebController.dayUnexOrde | 语法 | public ArrayList<OrderVO> dayUnexOrder(String today) |
| 前置条件 | 已经创建一个OrderOnWeb领域对象 |
| 后置条件 | 调用OrderOnWeb领域对象的dayUnexOrde方法 |
| OrderOnWebController.abnormalOrderCancel | 语法 | public ResultMsg abnormalOrderCancel(String ID,boolean b) |
| 前置条件 | 已经创建一个OrderOnWeb领域对象 |
| 后置条件 | 调用OrderOnWeb领域对象的abnormalOrderCancel方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderOnWeb.abnormalOrderScan() | 浏览异常订单 | |
| OrderOnWeb.abnormalOrderDetail(String ID) | 查看异常订单详情 | |
| OrderOnWeb. dayUnexOrder(String today) | 查看当日未执行订单 | |
| OrderOnWeb. AbnormalOrderCancel(String ID,boolean b) | 撤销异常订单 | |

表2-3-2 OrderOnWeb的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderOnWeb. abnormalOrderScan | 语法 | public ArrayList<OrderOnWebVO> abnormalOrderScan(); |
| 前置条件 | 账户信息符合规则且齐全 |
| 后置条件 | 系统显示异常订单 |
| OrderOnWeb. abnormalOrderDetail | 语法 | public OrderMsg abnormalOrderDetail(OrderOnWebVO orderVO); |
| 前置条件 | 账户信息符合规则且齐全 |
| 后置条件 | 系统显示异常订单详情 |
| OrderOnWebController.dayUnexOrde | 语法 | public ArrayList<OrderVO> dayUnexOrder(String today) |
| 前置条件 | 拥有时间信息 |
| 后置条件 | 调用OrderOnWeb领域对象的dayUnexOrde方法 |
| OrderOnWebController.abnormalOrderCancel | 语法 | public ResultMsg abnormalOrderCancel（String ID,Boolean b） |
| 前置条件 | 系统显示异常订单 |
| 后置条件 | 调用OrderOnWeb领域对象的abnormalOrderCancel方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | |
| OrderDataService.  update(OrderPO po) | 更新单一持久化对象 | |
| OrderDataService.showList() | 显示所有的单一持久化对象集合 | |

4、业务逻辑层的动态模型

图2-1-1表明了在酒店预订系统中，客户查看个人订单中相关对象之间的协作。

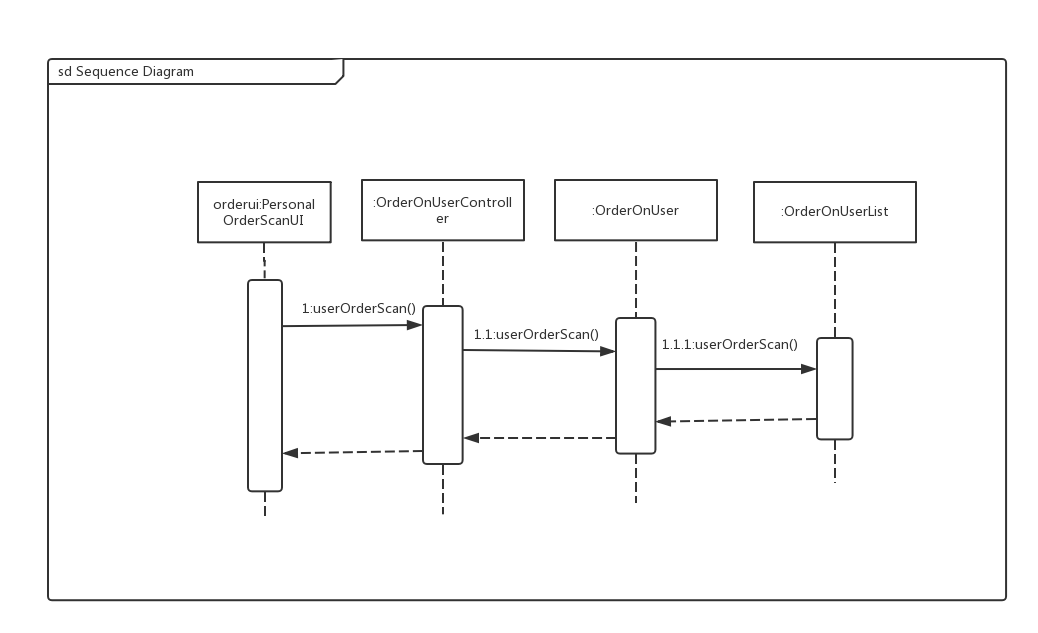


图2-1-1 查看个人订单的顺序图

图2-1-2表明了在酒店预订系统中，客户查看个人订单详情中相关对象之间的协作。

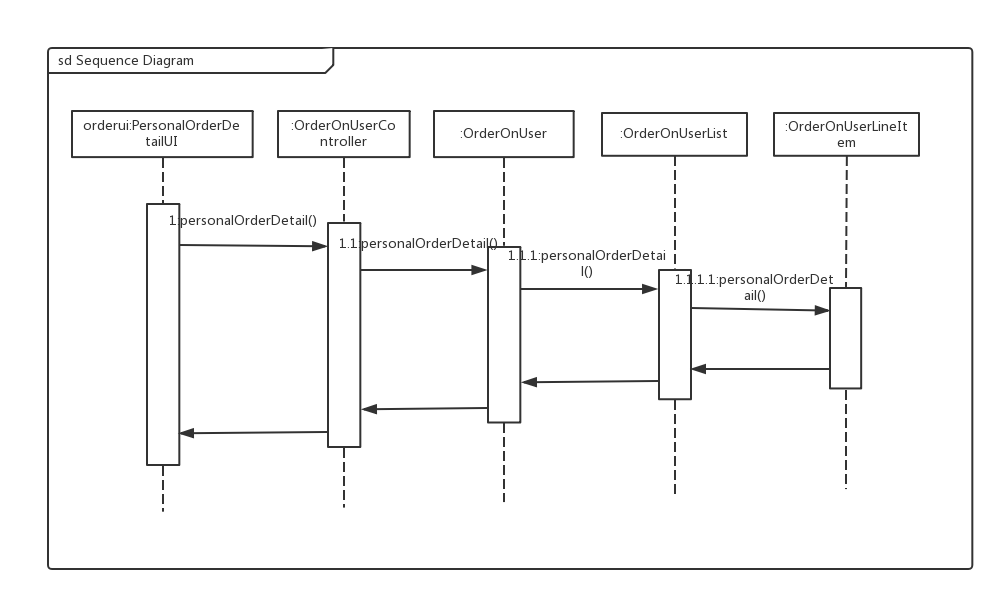


图2-1-2 查看个人订单详情的顺序图

图2-1-3表明了在酒店预订系统中，客户撤销订单中相关对象之间的协作。

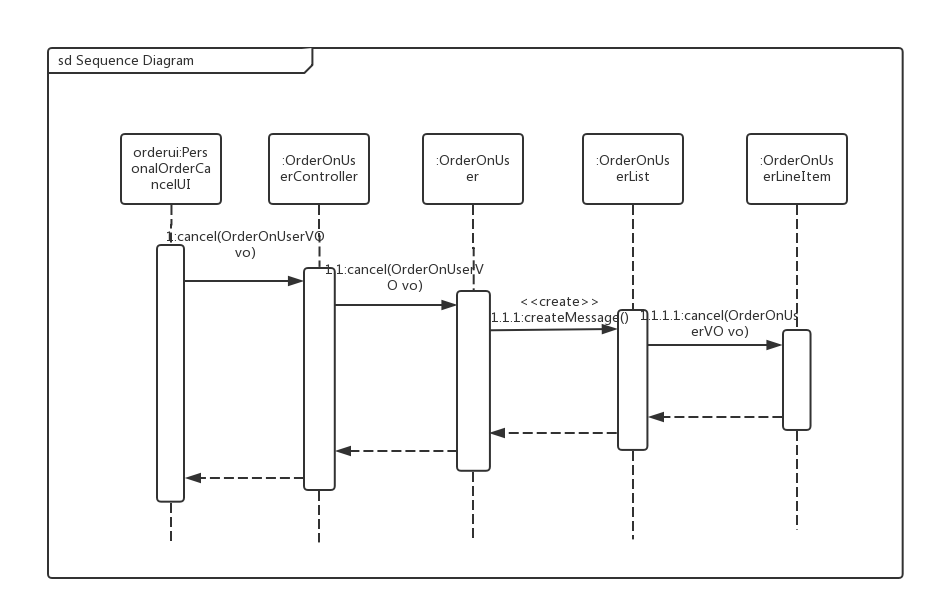


图2-1-3 撤销订单的顺序图

图2-2-1表明了在酒店预订系统中，酒店工作人员查看酒店订单信息中相关对象之间的协作。

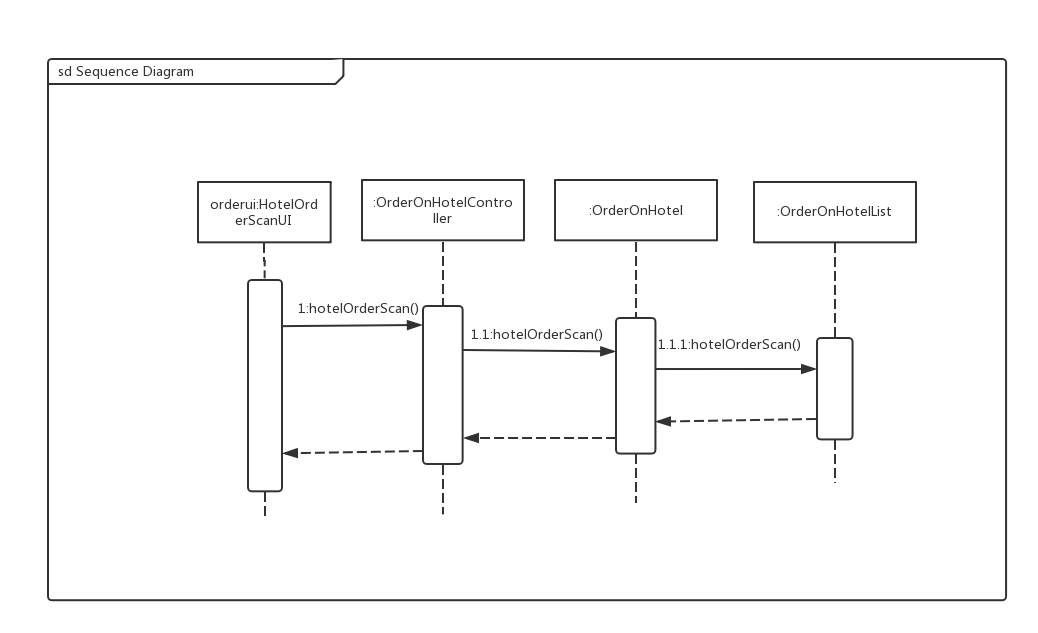


图2-2-1 查看酒店订单信息的顺序图

图2-2-2表明了在酒店预订系统中，酒店工作人员查看酒店订单详情中相关对象之间的协作。

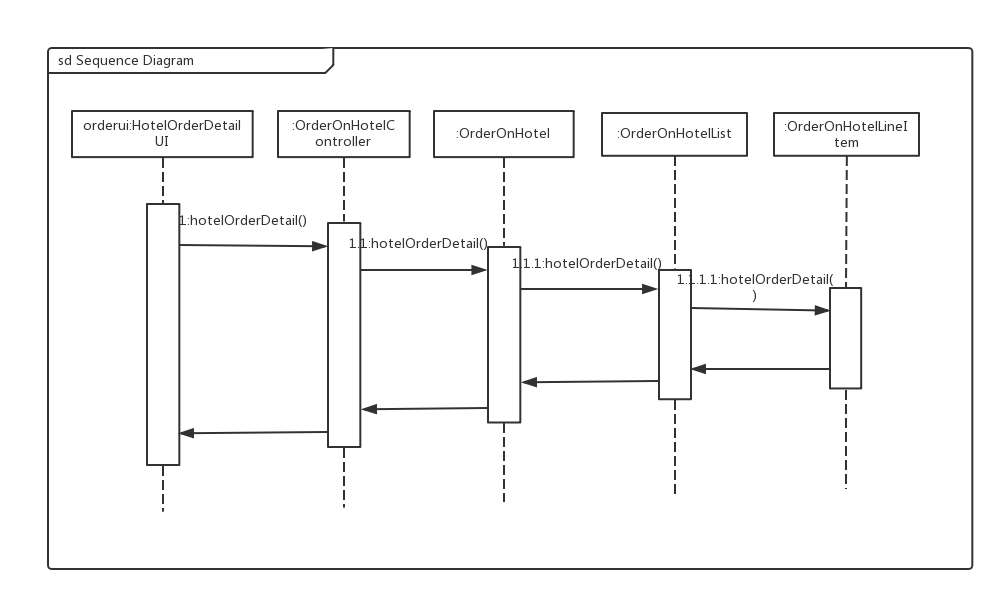


图2-2-2 查看酒店订单详情的顺序图

图2-2-3表明了在酒店预订系统中，酒店工作人员修改订单状态中相关对象之间的协作。

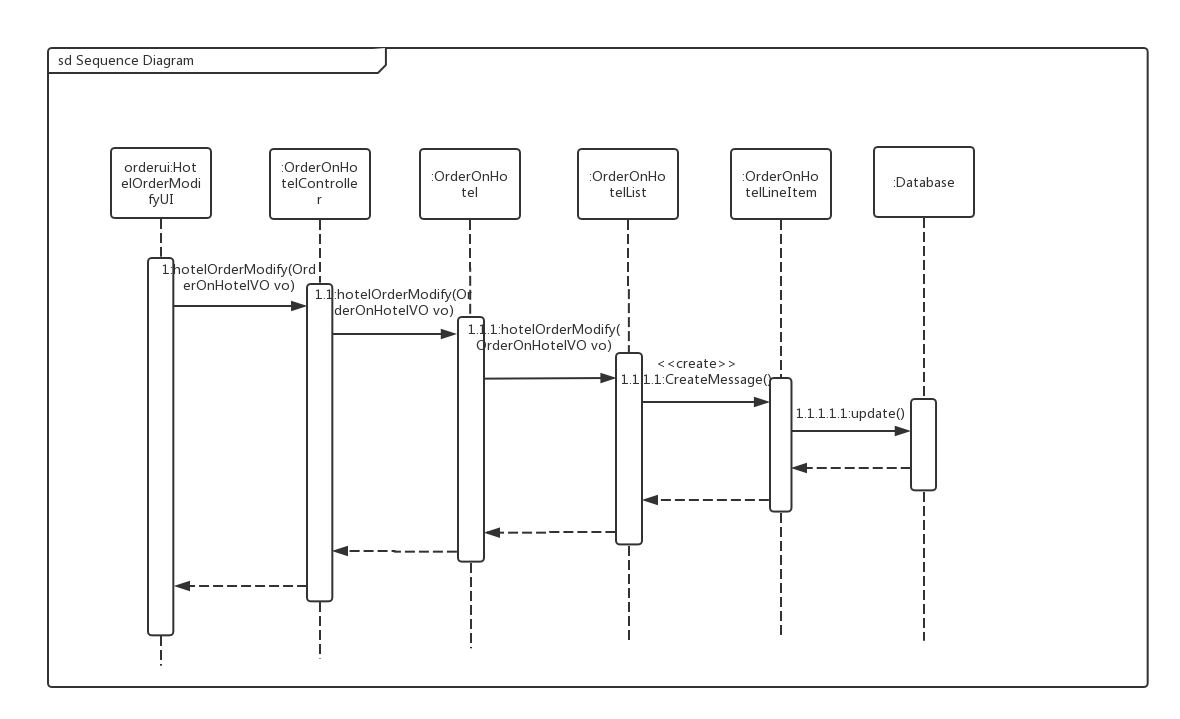


图2-2-3 修改订单状态的顺序图

图2-3-1表明了在酒店预订系统中，网站营销人员查看异常订单中相关对象之间的协作。

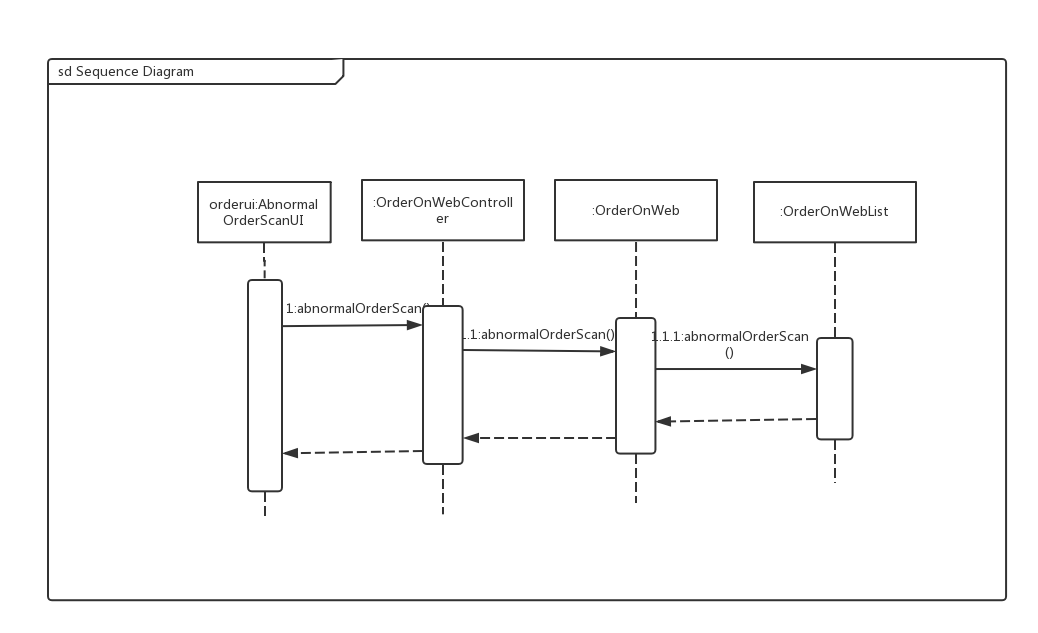


图2-3-1 查看异常订单的顺序图

图2-3-2表明了在酒店预订系统中，网站营销人员查看异常订单详情中相关对象之间的协作。

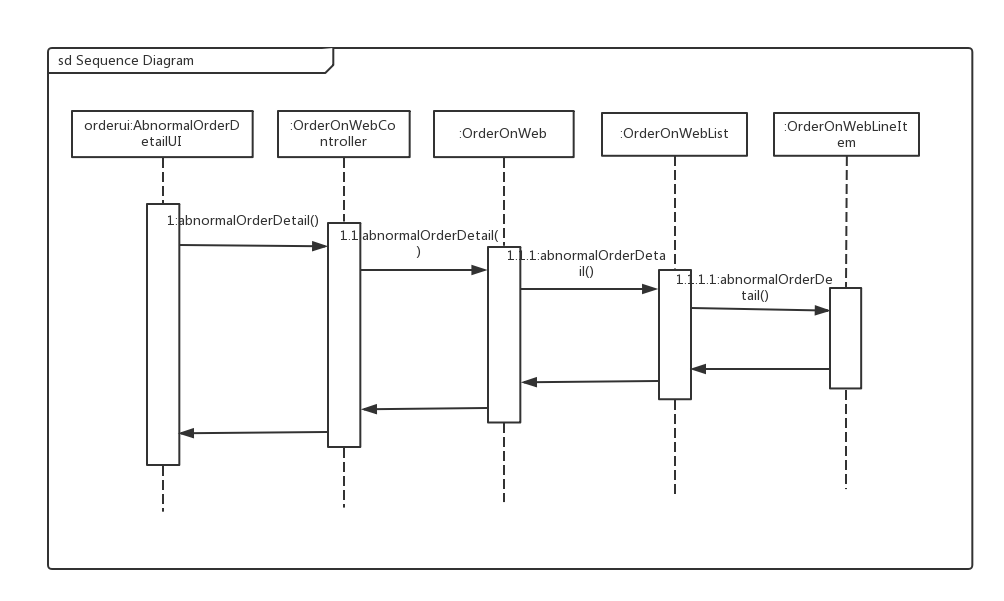


图2-3-2 查看异常订单详情的顺序图

图2-3-3表明了在酒店预订系统中，网站营销人员查看申诉列表中相关对象之间的协作。

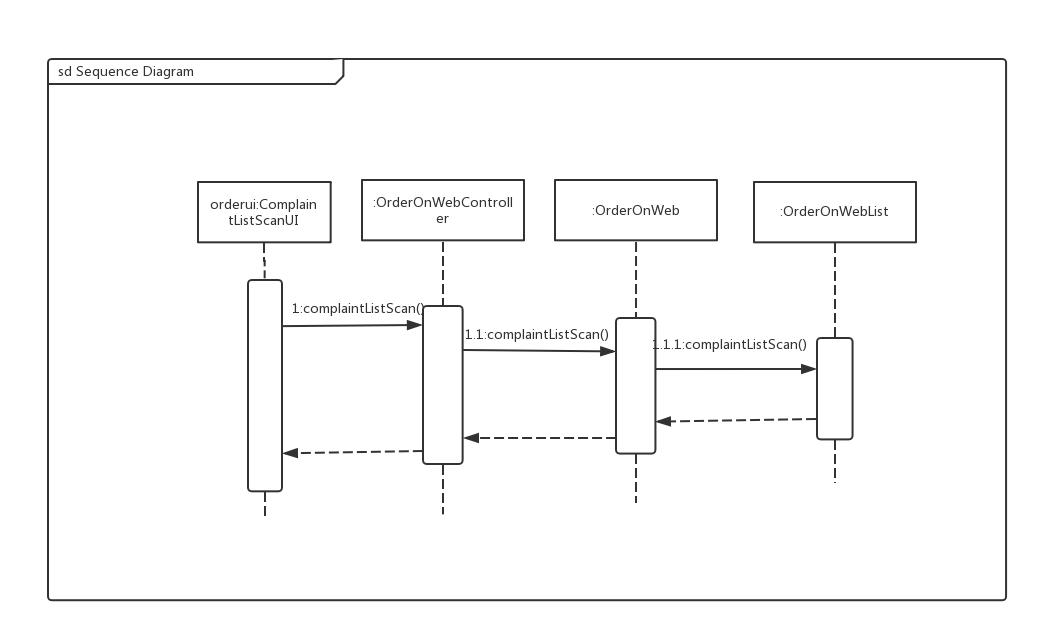


图2-3-3 查看申诉列表的顺序图

图2-3-4表明了在酒店预订系统中，网站营销人员处理申诉中相关对象之间的协作。

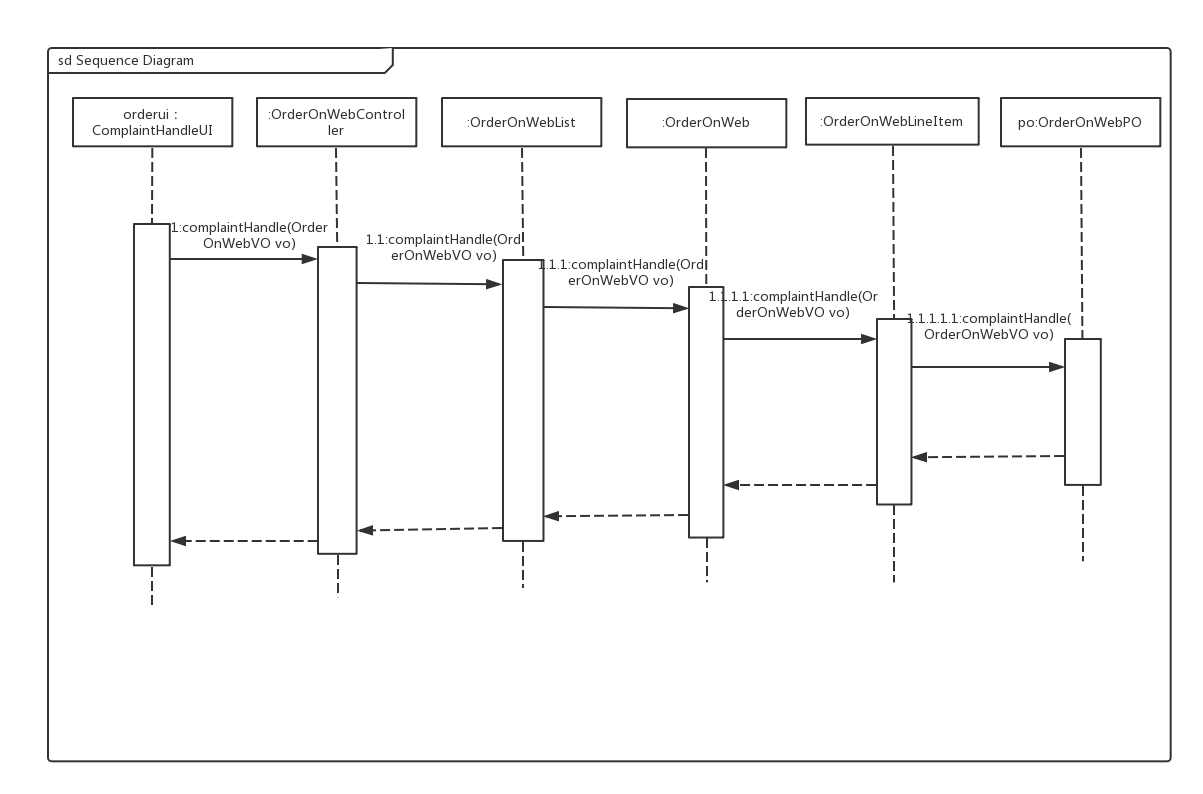


图2-3-4 处理申诉的顺序图

下图描述了OrderOnUser对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。

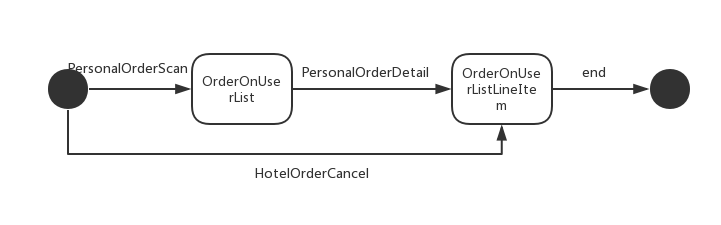


图3-1 OrderOnUser状态图

下图描述了OrderOnHotel对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。

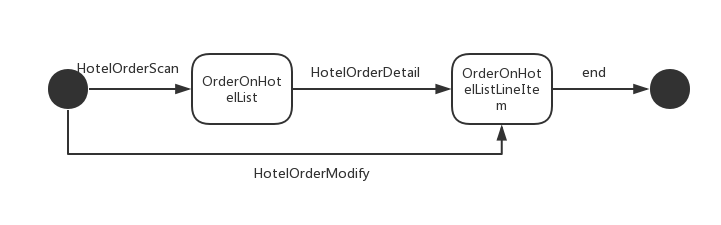


图3-1 OrderOnHotel状态图

下图描述了OrderOnWeb对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。

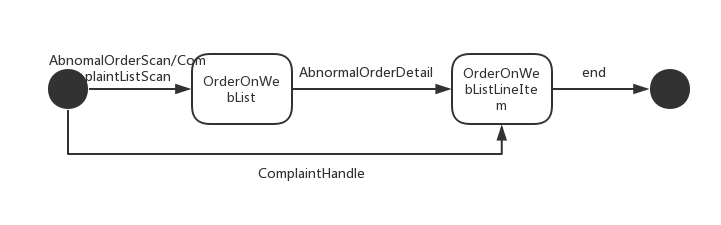


图3-1 OrderOnWeb状态图

5、业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的控制器委托给不同的领域对象。

### User模块

1. 模块概述

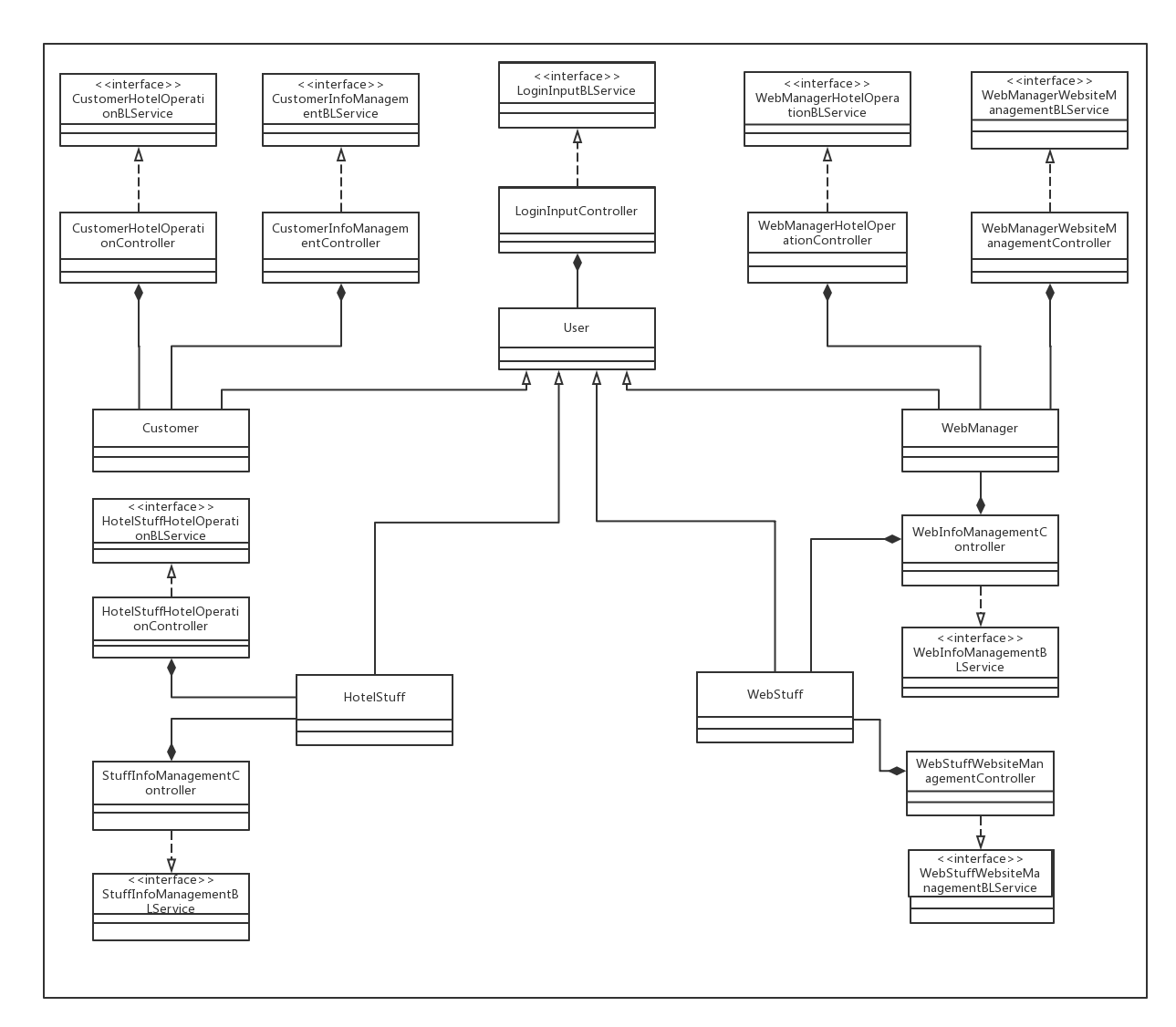
user模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

user模块的职责和接口参见软件系统结构描述文档内对该模块的描述。

1. 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。在展示层和业务逻辑层之间添加UserblService接口。在业务逻辑层和数据层之间添加UserdataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加LoginInputController、CustomerHotelOperationController等9个Controller。这些Controller会将对商品管理的业务逻辑处理委托给相应的用户对象。UserInfoPO、CustomerInfoPO、StuffInfoPO是作为账户信息的持久性对象被添加到设计模型中去的。

Userbl模块的设计如图



Userbl模块中各个类的设计

|  |  |
| --- | --- |
| 类 | 职责 |
| LoginInputController | 负责将来自界面的登录信息交给User类处理 |
| User | 用户实体类，也是客户和酒店管理人员的父类，处理登录信息以及基本信息管理。 |
| CustomerHotelOperationController | 负责将来自界面的客户酒店操作交给Customer类处理 |
| CustomerIndividualManagementController | 负责将来自界面的客户个人信息操作交给Customer类处理 |
| Customer | 客户对象，拥有客户信息，可以帮助完成客户界面所需要的服务 |
| StuffInfoManagementController | 负责将来自界面的酒店工作人员个人信息操作交给HotelStuff类处理 |
| HotelStuffHotelOperationController | 负责将来自界面的酒店工作人员酒店操作交给HotelStuff类处理 |
| HotelStuff | 酒店管理人员对象，拥有酒店管理人员信息和对应酒店部分信息，可以帮助完成酒店管理界面所需要的服务。 |
| WebInfoManagementController | 负责将来自界面的网站营销人员和网站管理人员个人信息操作交给WebStuff类和WebManager类处理 |
| WebStuffWebsiteManagementController | 将来自界面的网站营销人员对网站的操作交给WebStuff类处理 |
| WebStuff | 网站营销人员对象，拥有网站营销人员信息，有，有管理网站营销策略和网站订单信息的权限 |
| WebManagerHotelOperationController | 将来自界面的网站管理人员对酒店的操作交给WebManager类处理 |
| WebManagerWebsiteManagementController | 将来自界面的网站管理人员对网站的操作交给WebManager类处理 |
| WebManager | 网站管理人员对象，又有网站管理人员信息，可以访问网站和管理网站所有用户及酒店的信息。 |

1. 模块内部类的接口规范

LoginInputController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| LoginInputController.LogIn | 语法 | Public boolean LogIn(LoginInputVO vo); |
| 前置条件 | 已创建过一个用户对象。 |
| 后置条件 | 通过访问用户信息数据库检查账号和密码是否匹配，设置当前用户。 |
| LoginInputController.LogOut | 语法 | Public boolean LogOut(); |
| 前置条件 | 程序的当前用户不为空。 |
| 后置条件 | 程序的当前用户为空。 |
| LoginInputController.Register | 语法 | Public String Register(CustomerInfoVO vo); |
| 前置条件 | 输入信息合法。 |
| 后置条件 | 将新的用户信息加入到数据库中。 |
| LoginInputController.getType | 语法 | Public UserInfoVO IndividualBaseInfoInquiry (String userid); |
| 前置条件 | 无 |
| 后置条件 | 根据用户ID返回不同种类的用户信息。 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| User.getin | 客户登录 |
| User.getOut | 客户登出 |
| User.register | 新客户注册 |
| User.getType | 得到用户类型 |

User的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| User.LogIn | 语法 | public boolean LogIn(String id,String password); |
| 前置条件 | 已创建过一个用户对象。 |
| 后置条件 | 通过访问用户信息数据库检查账号和密码是否匹配，设置当前用户。 |
| User.LogOut | 语法 | Public boolean LogOut(boolean ok); |
| 前置条件 | 当前有已经登录的用户 |
| 后置条件 | 用户登出 |
| User.Register | 语法 | Public String Register(CustomerInfoVO vo); |
| 前置条件 | 无 |
| 后置条件 | 生成新用户账号 |
| User.getType | 语法 | Public UserType getType(String userid); |
| 前置条件 | 传入的用户ID以1、2、3、4其中之一开头 |
| 后置条件 | 无 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| UserManagementDataService.GetLoginInfo | 得到用户登录信息 |
| UserManagementDataService.AddCustomer | 增加客户 |
| MD5Util.md5Encode | 密码加密 |

CustomerHotelOperationController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CustomerHotelOperationController.HotelSearch | 语法 | public ArrayList<HotelInfoVO> HotelSearch(RoomInfoVO vo1,HotelInfoVO vo2,String userid); |
| 前置条件 | 已经输入地址和商圈。 |
| 后置条件 | 返回符合条件的酒店信息列表。 |
| CustomerHotelOperationController.HotelDetail | 语法 | Public HotelInfoVO hotelDetail(String hotelid,String userid); |
| 前置条件 | 无 |
| 后置条件 | 显示酒店详细信息 |
| CustomerHotelOperationController.getHotelOrder | 语法 | Public ArrayList<OrderVO> getHotelOrder(String hotelid,String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户在当前酒店的所有订单 |
| CustomerHotelOperationController.sortHotel | 语法 | Public ArrayList<HotelInfoVO> sortHotel(ArrayList<HotelInfoVO> before,Sort type); |
| 前置条件 | 无 |
| 后置条件 | 返回按照特定顺序排好的酒店信息列表 |
| CustomerHotelOperationController.OrderCreat | 语法 | Public ResultMsg OrderCreat(String hotelid,OrderVO vo); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 订单信息更新，酒店房间信息更新，客户个人信息更新 |
| CustomerHotelOperationController.HotelEvaluate | 语法 | Public void HotelEvaluate(HotelEvaluateVO vo); |
| 前置条件 | 当前酒店对应的订单状态为已执行 |
| 后置条件 | 更新酒店信息 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| Customer.HotelSearch | 搜索酒店 |
| Customerr.HotelDetail | 查看酒店详细信息 |
| Customer.getHotelOrder | 查看特定用户特定酒店的订单 |
| Customer.sortHotel | 按照特定顺序排序酒店 |
| Customer.OrderCreat | 创建订单 |
| Customer.HotelEvaluate | 评价酒店 |

CustomerInfoManagementController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CustomerInfoManagementController.IndividualBaseInfoInquiry | 语法 | public CustomerInfoVO IndividualBaseInfoINquiey(String userid); |
| 前置条件 | 当前用户存在 |
| 后置条件 | 返回用户个人信息 |
| CustomerInfoManagementController.IndividualBaseInfoModification | 语法 | Public IndividualBaseInfoModification(String userid,CustomerInfoVO vo); |
| 前置条件 | 当前用户存在且修改信息在授权之内 |
| 后置条件 | 更新数据库 |
| CustomerInfoManagementController.IndividualOrderInquiry | 语法 | Public ArrayList<OrderVO> IndividualOrderInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有订单 |
| CustomerInfoManagementController.UnfinishedOrderInquiey | 语法 | Public ArrayList<OrderVO> UnfinishedOrderInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有未执行订单 |
| CustomerInfoManagementController.finishedOrderInquiey | 语法 | Public ArrayList<OrderVO> finishedOrderInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有已执行订单 |
| CustomerInfoManagementController.CancleOrderInquiey | 语法 | Public ArrayList<OrderVO> CancleOrderInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有撤销订单 |
| CustomerInfoManagementController.AbnormalOrderInquiey | 语法 | Public ArrayList<OrderVO> AbnormalOrderInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有异常订单 |
| CustomerInfoManagementController.UnfinishedHotelInquiey | 语法 | Public ArrayList<HotelInfoVO> UnfinishedHotelInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有未执行订单对应的酒店 |
| CustomerInfoManagementController.finishedHotelInquiey | 语法 | Public ArrayList<HotelInfoVO> finishedHotelInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有已执行订单对应的酒店 |
| CustomerInfoManagementController.CancleHotelInquiey | 语法 | Public ArrayList<HotelInfoVO> CancleHotelInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有撤销订单对应的酒店 |
| CustomerInfoManagementController.AbnormalHotelInquiey | 语法 | Public ArrayList<HotelInfoVO> AbnormalHotelInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有异常订单对应的酒店 |
| CustomerInfoManagementController.IndividualCreditInquiry | 语法 | Public int IndividualCreditInquory(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户信用值 |
| CustomerInfoManagementController.IndividualCreditRecore | 语法 | Public ArrayList<CreditVO> IndividualCreditRecord(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户信用记录 |
| CustomerInfoManagementController.PersonalOrderCancle | 语法 | Public ResultMsg PersonalOrderCancle(OrderVO vo); |
| 前置条件 | 用户已经登录，订单存在且未执行 |
| 后置条件 | 更新用户信息、酒店信息和订单信息 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| Customer.IndividualBaseInfoInquiry | 客户查询个人基本信息 |
| Customer.IndividualBaseInfoModification | 客户修改个人基本信息 |
| Customer.IndividualOrderInquiry | 查看用户个人订单 |
| Customer.SpecialOrderInquiry | 查看用户某一类型的个人订单 |
| Customer.IndividualHotelInquiry | 查看用户预定过的酒店 |
| Customer.SpecialOrderInquiry | 查看用户预定过的某一订单类型类型对应的酒店 |
| Customer.IndividualCreditInquiry | 查看用户信用值 |
| Customer.IndividualCreditRecord | 查看用户信用记录 |
| Customer.PersonalOrderCancle | 撤销订单 |

Customer的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Customer.HotelSearch | 语法 | public ArrayList<HotelInfoVO> HotelSearch(RoomInfoVO vo1,HotelInfoVO vo2,String userid); |
| 前置条件 | 已经输入地址和商圈。 |
| 后置条件 | 返回符合条件的酒店信息列表。 |
| Customer.HotelDetail | 语法 | Public HotelInfoVO hotelDetail(String hotelid,String userid); |
| 前置条件 | 无 |
| 后置条件 | 显示酒店详细信息 |
| Customer.getHotelOrder | 语法 | Public ArrayList<OrderVO> getHotelOrder(String hotelid,String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户在当前酒店的所有订单 |
| Customer.sortHotel | 语法 | Public ArrayList<HotelInfoVO> sortHotel(ArrayList<HotelInfoVO> before,Sort type); |
| 前置条件 | 无 |
| 后置条件 | 返回按照特定顺序排好的酒店信息列表 |
| Customer.OrderCreat | 语法 | Public ResultMsg OrderCreat(String hotelid,OrderVO vo); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 订单信息更新，酒店房间信息更新，客户个人信息更新 |
| Customer.HotelEvaluate | 语法 | Public void HotelEvaluate(HotelEvaluateVO vo); |
| 前置条件 | 当前酒店对应的订单状态为已执行 |
| 后置条件 | 更新酒店信息 |
| Customer.IndividualBaseInfoInquiry | 语法 | public CustomerInfoVO IndividualBaseInfoINquiey(String userid); |
| 前置条件 | 当前用户存在 |
| 后置条件 | 返回用户个人信息 |
| Customer.IndividualBaseInfoModification | 语法 | Public IndividualBaseInfoModification(String userid,CustomerInfoVO vo); |
| 前置条件 | 当前用户存在且修改信息在授权之内 |
| 后置条件 | 更新数据库 |
| Customer.IndividualOrderInquiry | 语法 | Public ArrayList<OrderVO> IndividualOrderInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有订单 |
| Customer.UnfinishedOrderInquiey | 语法 | Public ArrayList<OrderVO> UnfinishedOrderInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有未执行订单 |
| Customer.finishedOrderInquiey | 语法 | Public ArrayList<OrderVO> finishedOrderInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有已执行订单 |
| Customer.CancleOrderInquiey | 语法 | Public ArrayList<OrderVO> CancleOrderInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有撤销订单 |
| Customer.AbnormalOrderInquiey | 语法 | Public ArrayList<OrderVO> AbnormalOrderInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有异常订单 |
| Customer.UnfinishedHotelInquiey | 语法 | Public ArrayList<HotelInfoVO> UnfinishedHotelInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有未执行订单对应的酒店 |
| Customer.finishedHotelInquiey | 语法 | Public ArrayList<HotelInfoVO> finishedHotelInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有已执行订单对应的酒店 |
| Customer.CancleHotelInquiey | 语法 | Public ArrayList<HotelInfoVO> CancleHotelInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有撤销订单对应的酒店 |
| Customer.AbnormalHotelInquiey | 语法 | Public ArrayList<HotelInfoVO> AbnormalHotelInquiry(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户的所有异常订单对应的酒店 |
| Customer.IndividualCreditInquiry | 语法 | Public int IndividualCreditInquory(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户信用值 |
| Customer.IndividualCreditRecore | 语法 | Public ArrayList<CreditVO> IndividualCreditRecord(String userid); |
| 前置条件 | 用户已经登录 |
| 后置条件 | 返回当前用户信用记录 |
| Customer.PersonalOrderCancle | 语法 | Public ResultMsg PersonalOrderCancle(OrderVO vo); |
| 前置条件 | 用户已经登录，订单存在且未执行 |
| 后置条件 | 更新用户信息、酒店信息和订单信息 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| HotelSearchBLService.selectHotel | 筛选酒店 |
| HotelReserveBLService.reserveHotel | 预订酒店 |
| OrderOnUserBLService.creatorder | 创建订单 |
| CustomerManagementDataService.addCustomerHotel | 增加个人预定过的酒店 |
| HotelEvaluateBLService.inputEvaluate | 评价酒店 |
| CustomerManagementDataService.GetCustomerInfo | 获取客户个人信息 |
| CustomerManagementDataService.SetCustomerInfo | 修改客户个人信息 |
| CreditBLService.getCredit | 获取客户信用值 |
| CreditBLService.getCreditList | 获取用户信用记录 |
| OrderOnUserBLService。personalOrderScan | 获得用户的订单信息 |
| CustomerManagementDataService.GetCustomerrHotel | 获得用户预订过的酒店信息 |
| OrderOnUserBLService.PersonalOrderCancle | 撤销订单 |

HotelStuffHotelOperationController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| HotelStuffHotelOperationController.HotelInformationInquiry | 语法 | Public HotelInfoVO HotelInformationInquiry(String hotelid); |
| 前置条件 | 被查询酒店已经存在 |
| 后置条件 | 返回酒店信息 |
| HotelStuffHotelOperationController.HotelInformationModification | 语法 | Public HotelInfoVO HotelInformationModification(HotelInfoVO vo,String userid); |
| 前置条件 | 用户ID以2开头且酒店信息存在 |
| 后置条件 | 更新酒店信息数据库 |
| HotelStuffHotelOperationController.SetRoomInfo | 语法 | Public ResultMsg SetRoomInfo(ArrayList<RoomInfoVO> rooms); |
| 前置条件 | 房间对应的酒店已存在 |
| 后置条件 | 更新酒店房间信息 |
| HotelStuffHotelOperationController.HotelStrategyManage | 语法 | Public boolean HotelStrategyManage(PromotionHotelVO vo); |
| 前置条件 | 对应的酒店已存在 |
| 后置条件 | 更新数据库中的酒店信息 |
| HotelStuffHotelOperationController.HotelStrategyDelete | 语法 | Public boolean HotelStrategyDelete(PromotionHotelVO vo); |
| 前置条件 | 对应的酒店已存在，对应的促销策略已存在 |
| 后置条件 | 更新数据库中的酒店信息，更新数据库中的促销策略信息 |
| HotelStuffHotelOperationController.HotelPromotionInquiry | 语法 | Public ArrayList<PromotionHotelVO> HotelPromotionInquiry(PromotionHotelVO); |
| 前置条件 | 对应的酒店已存在 |
| 后置条件 | 返回酒店促销策略 |
| HotelStuffHotelOperationController.HotelStrategeAdd | 语法 | Public boolean HotelStrategyAdd(PromotionHotelVO); |
| 前置条件 | 对应的酒店已存在 |
| 后置条件 | 更新酒店信息，更新促销策略信息 |
| HotelStuffHotelOperationController.OrderStateUpdate | 语法 | Public ResultMsg OrderStateUpdate(String userid,OrderVO vo); |
| 前置条件 | 对应订单存在且用户办理入住手续 |
| 后置条件 | 更新订单信息和用户信息 |
| HotelStuffHotelOperationController.OrderScan | 语法 | Public ArrayList<OrderVO> OrderScan(String hotelid); |
| 前置条件 | 已经创建一个酒店对象，并且程序当前用户为该酒店的管理人员 |
| 后置条件 | 返回酒店的订单信息列表 |
| HotelStuffHotelOperationController.SpecialOrderScan | 语法 | Public ArrayList<OrderVO> SpecialOrderScan(String hotelid,OrderState state); |
| 前置条件 | 已经创建一个酒店对象，并且程序当前用户为该酒店的管理人员 |
| 后置条件 | 返回酒店的某一种订单信息列表 |
| HotelStuffHotelOperationController.UpdateRoomState | 语法 | Public ResultMsg UpdateRoomState(RoomInfoVO vo); |
| 前置条件 | 操作酒店存在 |
| 后置条件 | 更新酒店信息 |
| HotelStuffHotelOperationController.TypeInRoom | 语法 | Public ResultMsg TypeInRoom(RoomInfoVO vo); |
| 前置条件 | 操作酒店存在 |
| 后置条件 | 更新酒店信息 |
| HotelStuffHotelOperationController.HotelRoomList | 语法 | Public ArryaList<RoomInfoVO> HotelRoomList(String hotelid); |
| 前置条件 | 操作酒店存在 |
| 后置条件 | 返回酒店房间信息 |
| HotelStuffHotelOperationController.setAbnormalOrder | 语法 | Public ResultMsg setAbnormalOrder(String hotelid,String orderid); |
| 前置条件 | 操作酒店存在，订单是异常订单 |
| 后置条件 | 更新订单信息和用户信息 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务内容 |
| HotelStuff.HotelInformationInquiry | 查看酒店信息 |
| HotelStuff.HotelInformationModification | 修改酒店信息 |
| HotelStuff.SetRoomInfo | 设置房间信息 |
| HotelStuffr.HotelStrategyManage | 管理酒店促销策略 |
| HotelStuff.HotelStrategyDelete | 删除酒店促销策略 |
| HotelStuff.HotelPromotionInquiry | 查看酒店促销策略 |
| HotelStuff.HotelStrategeAdd | 添加酒店促销策略 |
| HotelStuff.OrderStateUpdate | 更新订单状态 |
| HotelStuff.OrderScan | 浏览酒店订单 |
| HotelStuff.SpecialOrderScan | 浏览酒店某一类型订单 |
| HotelStuff.UpdateRoomState | 更新房间状态 |
| HotelStuff.TypeInRoom | 录入可用房间 |
| HotelStuff.HotelRoomList | 查看酒店房间列表 |
| HotelStuff.setAbnormalOrder | 处理本地异常订单 |

StuffInfoManagementController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| StuffInfoManagementController.IndividualBaseInfoInquiry | 语法 | Public SruffInfoVO IndividualBaseInfoInquiry(String userid); |
| 前置条件 | 酒店管理人员已经登录 |
| 后置条件 | 返回酒店管理人员信息 |
| StuffInfoManagementController.IndividualBaseInfoModification | 语法 | Public ResultMsg IndividualBaseInfoModification(String userid,StuffInfoVO vo); |
| 前置条件 | 酒店管理人员已经登录 |
| 后置条件 | 更新酒店管理人员信息 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务内容 |
| HotelStuff.IndividualBaseInfoInquiry | 查看酒店管理人员信息 |
| HotelStuff.IndividualBaseInfoModification | 修改酒店管理人员信息 |

HotelStuff的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| StuffInfoManagementController.IndividualBaseInfoInquiry | 语法 | Public SruffInfoVO IndividualBaseInfoInquiry(String userid); |
| 前置条件 | 酒店管理人员已经登录 |
| 后置条件 | 返回酒店管理人员信息 |
| StuffInfoManagementController.IndividualBaseInfoModification | 语法 | Public ResultMsg IndividualBaseInfoModification(String userid,StuffInfoVO vo); |
| 前置条件 | 酒店管理人员已经登录 |
| 后置条件 | 更新酒店管理人员信息 |
| HotelStuffHotelOperationController.HotelInformationInquiry | 语法 | Public HotelInfoVO HotelInformationInquiry(String hotelid); |
| 前置条件 | 被查询酒店已经存在 |
| 后置条件 | 返回酒店信息 |
| HotelStuffHotelOperationController.HotelInformationModification | 语法 | Public HotelInfoVO HotelInformationModification(HotelInfoVO vo,String userid); |
| 前置条件 | 用户ID以2开头且酒店信息存在 |
| 后置条件 | 更新酒店信息数据库 |
| HotelStuffHotelOperationController.SetRoomInfo | 语法 | Public ResultMsg SetRoomInfo(ArrayList<RoomInfoVO> rooms); |
| 前置条件 | 房间对应的酒店已存在 |
| 后置条件 | 更新酒店房间信息 |
| HotelStuffHotelOperationController.HotelStrategyManage | 语法 | Public boolean HotelStrategyManage(PromotionHotelVO vo); |
| 前置条件 | 对应的酒店已存在 |
| 后置条件 | 更新数据库中的酒店信息 |
| HotelStuffHotelOperationController.HotelStrategyDelete | 语法 | Public boolean HotelStrategyDelete(PromotionHotelVO vo); |
| 前置条件 | 对应的酒店已存在，对应的促销策略已存在 |
| 后置条件 | 更新数据库中的酒店信息，更新数据库中的促销策略信息 |
| HotelStuffHotelOperationController.HotelPromotionInquiry | 语法 | Public ArrayList<PromotionHotelVO> HotelPromotionInquiry(PromotionHotelVO); |
| 前置条件 | 对应的酒店已存在 |
| 后置条件 | 返回酒店促销策略 |
| HotelStuffHotelOperationController.HotelStrategeAdd | 语法 | Public boolean HotelStrategyAdd(PromotionHotelVO); |
| 前置条件 | 对应的酒店已存在 |
| 后置条件 | 更新酒店信息，更新促销策略信息 |
| HotelStuffHotelOperationController.OrderStateUpdate | 语法 | Public ResultMsg OrderStateUpdate(String userid,OrderVO vo); |
| 前置条件 | 对应订单存在且用户办理入住手续 |
| 后置条件 | 更新订单信息和用户信息 |
| HotelStuffHotelOperationController.OrderScan | 语法 | Public ArrayList<OrderVO> OrderScan(String hotelid); |
| 前置条件 | 已经创建一个酒店对象，并且程序当前用户为该酒店的管理人员 |
| 后置条件 | 返回酒店的订单信息列表 |
| HotelStuffHotelOperationController.SpecialOrderScan | 语法 | Public ArrayList<OrderVO> SpecialOrderScan(String hotelid,OrderState state); |
| 前置条件 | 已经创建一个酒店对象，并且程序当前用户为该酒店的管理人员 |
| 后置条件 | 返回酒店的某一种订单信息列表 |
| HotelStuffHotelOperationController.UpdateRoomState | 语法 | Public ResultMsg UpdateRoomState(RoomInfoVO vo); |
| 前置条件 | 操作酒店存在 |
| 后置条件 | 更新酒店信息 |
| HotelStuffHotelOperationController.TypeInRoom | 语法 | Public ResultMsg TypeInRoom(RoomInfoVO vo); |
| 前置条件 | 操作酒店存在 |
| 后置条件 | 更新酒店信息 |
| HotelStuffHotelOperationController.HotelRoomList | 语法 | Public ArryaList<RoomInfoVO> HotelRoomList(String hotelid); |
| 前置条件 | 操作酒店存在 |
| 后置条件 | 返回酒店房间信息 |
| HotelStuffHotelOperationController.setAbnormalOrder | 语法 | Public ResultMsg setAbnormalOrder(String hotelid,String orderid); |
| 前置条件 | 操作酒店存在，订单是异常订单 |
| 后置条件 | 更新订单信息和用户信息 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务内容 |
| HotelInfoCheckBLService.checkHotelInfo | 查看酒店信息 |
| HotelInfoMaintainBLService.inputHotelInfo | 修改酒店信息 |
| UserManagementDataService.IndividualBaseInfoInquiry | 查看酒店管理人员信息 |
| UserManagementDataService.IndividualBaseInfoModification | 修改酒店管理人员信息 |
| PromotionHotelBLService.changeBirthCut | 查看酒店生日促销策略 |
| PromotionHotelBLService.changeOverCut | 查看酒店满减促销策略 |
| PromotionHotelBLService.changeJoin | 查看酒店合作促销策略 |
| PromotionHotelBLService.changeHotelCustomCut | 查看酒店普通促销策略 |
| PromotionHotelBLService.deleteBirthCut | 删除酒店生日促销策略 |
| PromotionHotelBLService.deleteOverCut | 删除酒店满减促销策略 |
| PromotionHotelBLService.deleteJoin | 删除酒店合作促销策略 |
| PromotionHotelBLService.deleteCustomCut | 删除酒店普通促销策略 |
| PromotionHotelBLService.addBirthCut | 增加酒店生日促销策略 |
| PromotionHotelBLService.addOverCut | 增加酒店满减促销策略 |
| PromotionHotelBLService.addJoin | 增加酒店合作促销策略 |
| PromotionHotelBLService.addCustomCut | 增加酒店普通促销策略 |
| RoomAddBLService.updateRoom | 更新房间信息 |
| RoomAddBLService.addRoom | 添加房间 |
| RoomAddBLService.hotelRoomSearch | 查看酒店房间 |
| RoomAddBLService.hotelRoomMod | 修改酒店房间 |

WebInfoManagementController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| WebInfoManagementController.IndividualBaeInfoInquiry | 语法 | Public UserInfoVO IndividualBaseInfoInquiry(String userid); |
| 前置条件 | 已经创建网站营销人员或网站管理人员对象并且被查询对象就是程序的当前用户 |
| 后置条件 | 返回网站营销人员或网站管理人员的基本信息 |
| WebInfoManagementController.IndividualBaseInfoModification | 语法 | Public ResultMsg IndividualBaseInfoModification(String userid,UserInfoVO vo); |
| 前置条件 | 营销人员或网站管理人员已经登录 |
| 后置条件 | 更新营销人员或网站管理人员信息 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务内容 |
| WebStuff.IndividualBaseInfoInquory | 查看网站营销人员基本信息 |
| WebStuff.IndividualBaseInfoModifiation | 修改网站营销人员基本信息 |
| WebManager.IndividualBaseInfoInquory | 查看网站管理人员基本信息 |
| WebManager.IndividualBaseInfoModifiation | 修改网站管理人员基本信息 |

WebStuffWebManagementController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| WebStuffWebManagementController.WebsiteStrageCreat | 语法 | Public boolean WebStrategeCreat(PrimotionWebVO vo); |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 更新促销策略数据 |
| WebStuffWebManagementController.WebsiteStrageMod | 语法 | Public boolean WebStrategeMod(PrimotionWebVO vo); |
| 前置条件 | 网站营销人员已经登录且被修改促销策略存在 |
| 后置条件 | 更新促销策略数据 |
| WebStuffWebManagementController.WebsiteStrageDelete | 语法 | Public boolean WebStrategeDelete(PrimotionWebVO vo); |
| 前置条件 | 网站营销人员已经登录且被删除促销策略存在 |
| 后置条件 | 更新促销策略数据 |
| WebStuffWebManagementController.AbnormalOrderScan | 语法 | Public ArrayList<OrderVO> AbnormalOrderScan(); |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 返回异常订单列表 |
| WebStuffWebManagementController.UserCreditModification | 语法 | Public ResultMsg UserCreditModification(Stringnuserid,int n); |
| 前置条件 | 网站营销人员已经登录且被修改用户存在 |
| 后置条件 | 更新用户信息和信用值信息 |
| WebStuffWebManagementController.WebSiteStrategyInquiry | 语法 | Public ArrayList<PromotionWebVO> WebSiteStrategyInquiry(PromotionWebVO vo); |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 返回网站促销策略列表 |
| WebStuffWebManagementController.UserCreditInquiry | 语法 | Public int UserCreditInquiry(String userid); |
| 前置条件 | 网站营销人员已经登录且被查询用户存在 |
| 后置条件 | 返回用户信用值 |
| WebStuffWebManagementController.DayUnexOrder | 语法 | Public ArrayList<OrderVO> DayUnexOrder(String today); |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 返回未执行订单 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务内容 |
| WebStuff.WebsiteStrageCreat | 创建网站促销策略 |
| WebStuff.WebsiteStrageMod | 修改网站促销策略 |
| WebStuff.WebsiteStrageDelete | 删除网站促销策略 |
| WebStuff.AbnormalOrderScan | 浏览异常订单 |
| WebStuff.UserCreditModification | 修改用户信用 |
| WebStuff.WebSiteStrategyInquiry | 查询网站促销策略 |
| WebStuff.UserCreditInquiry | 查询用户信用 |
| WebStuff.DayUnexOrder | 查看未执行订单 |

WebStuff的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| WeStuff.IndividualBaeInfoInquiry | 语法 | Public UserInfoVO IndividualBaseInfoInquiry(String userid); |
| 前置条件 | 已经创建网站营销人员或网站管理人员对象并且被查询对象就是程序的当前用户 |
| 后置条件 | 返回网站营销人员或网站管理人员的基本信息 |
| WebStuff.IndividualBaseInfoModification | 语法 | Public ResultMsg IndividualBaseInfoModification(String userid,UserInfoVO vo); |
| 前置条件 | 营销人员或网站管理人员已经登录 |
| 后置条件 | 更新营销人员或网站管理人员信息 |
| WebStuff.WebsiteStrageCreat | 语法 | Public boolean WebStrategeCreat(PrimotionWebVO vo); |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 更新促销策略数据 |
| WebStuff.WebsiteStrageMod | 语法 | Public boolean WebStrategeMod(PrimotionWebVO vo); |
| 前置条件 | 网站营销人员已经登录且被修改促销策略存在 |
| 后置条件 | 更新促销策略数据 |
| WebStuff.WebsiteStrageDelete | 语法 | Public boolean WebStrategeDelete(PrimotionWebVO vo); |
| 前置条件 | 网站营销人员已经登录且被删除促销策略存在 |
| 后置条件 | 更新促销策略数据 |
| WebStuff.AbnormalOrderScan | 语法 | Public ArrayList<OrderVO> AbnormalOrderScan(); |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 返回异常订单列表 |
| WebStuff.UserCreditModification | 语法 | Public ResultMsg UserCreditModification(Stringnuserid,int n); |
| 前置条件 | 网站营销人员已经登录且被修改用户存在 |
| 后置条件 | 更新用户信息和信用值信息 |
| WebStuff.WebSiteStrategyInquiry | 语法 | Public ArrayList<PromotionWebVO> WebSiteStrategyInquiry(PromotionWebVO vo); |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 返回网站促销策略列表 |
| WebStuff.UserCreditInquiry | 语法 | Public int UserCreditInquiry(String userid); |
| 前置条件 | 网站营销人员已经登录且被查询用户存在 |
| 后置条件 | 返回用户信用值 |
| WebStuff.DayUnexOrder | 语法 | Public ArrayList<OrderVO> DayUnexOrder(String today); |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 返回未执行订单 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| UserManagementDataService.GetWebStuffInfo | 获得网站营销人员基本信息 |
| UserManagementDataService.SetWebStuffInfo | 修改营销人员基本信息 |
| PromotionWebBLService.addLevelCut | 增加促销策略 |
| PromotionWebBLService.addCircleCut | 增加促销策略 |
| PromotionWebBLService.addCustomCut | 增加促销策略 |
| PromotionWebBLService.changeLevelCut | 修改促销策略 |
| PromotionWebBLService.changeCircleCut | 修改促销策略 |
| PromotionWebBLService.changeCustomCut | 修改促销策略 |
| OrderOnWebBLService.abnormalOrderScan | 浏览异常订单 |
| CreditBLService.addCredit | 增加用户信用值 |
| PromotionWebBLService.getWebPromotion | 查看网站促销策略 |
| CreditBLService.getCredit | 查看用户信用值 |
| OrderOnWebBLService.dayUnexOrder | 查看未执行订单 |
| PromotionWebBLService.deleteLevelCut | 删除促销策略 |
| PromotionWebBLService.deleteCircleCut | 删除促销策略 |
| PromotionWebBLService.deleteCustomCut | 删除促销策略 |

WebManagerHotelOperationController接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| WebManagerHotelOperationController.HotelAdd | 语法 | Public String HotelAdd(HotelInfoVO vo); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 更新酒店信息并返回新酒店账号 |
| WebManagerHotelOperationController.HotelScan | 语法 | Public ArrayList<HotelInfoVO> HotelScan(); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 返回所有酒店信息列表 |
| WebManagerHotelOperationController.StuffAdd | 语法 | Public String StuffAdd(StuffInfoVO vo); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 更新酒店信息并返回新酒店管理人员账号 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| WebManager.HotelAdd | 添加酒店 |
| WebManager.HotelScan | 浏览酒店 |
| WebManager.StuffAdd | 添加酒店工作人员 |

WebManagerWebsiteManagementController接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| WebManagerWebsiteManagementController.UserInfomationInquiry | 语法 | Public UserInfoVO UserInfomationInquiry(String userid); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 更新酒店信息并返回新酒店账号 |
| WebManagerWebsiteManagementController.UserInfomationModification | 语法 | Public ResultMsg UserInfomationModification(String userid, UserInfoVO vo); |
| 前置条件 | 网站管理人员已经登录且被修改用户存在 |
| 后置条件 | 更新用户信息 |
| WebManagerWebsiteManagementController.AddWebStuff | 语法 | Public String AddWebStuff(UserInfoVO vo); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 更新用户信息并返回新网站管理人员账号 |
| WebManagerWebsiteManagementController.WebStuffScan | 语法 | ArrayList<UserInfoVO> WebStuffScan(); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 返回网站营销人员信息列表 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| WebManager.UserInfomationInquiry | 查询用户信息 |
| WebManager.UserInfomationModification | 修改用户信息 |
| WebManager.AddWebStuff | 添加网站营销人员 |
| WebManager.WebStuffScan | 浏览网站营销人员列表 |

WebManager接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| WebManager.HotelAdd | 语法 | Public String HotelAdd(HotelInfoVO vo); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 更新酒店信息并返回新酒店账号 |
| WebManager.HotelScan | 语法 | Public ArrayList<HotelInfoVO> HotelScan(); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 返回所有酒店信息列表 |
| WebManager.StuffAdd | 语法 | Public String StuffAdd(StuffInfoVO vo); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 更新酒店信息并返回新酒店管理人员账号 |
| WebManager.UserInfomationInquiry | 语法 | Public UserInfoVO UserInfomationInquiry(String userid); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 更新酒店信息并返回新酒店账号 |
| WebManager.UserInfomationModification | 语法 | Public ResultMsg UserInfomationModification(String userid, UserInfoVO vo); |
| 前置条件 | 网站管理人员已经登录且被修改用户存在 |
| 后置条件 | 更新用户信息 |
| WebManager.AddWebStuff | 语法 | Public String AddWebStuff(UserInfoVO vo); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 更新用户信息并返回新网站管理人员账号 |
| WebManager.WebStuffScan | 语法 | ArrayList<UserInfoVO> WebStuffScan(); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 返回网站营销人员信息列表 |
| WebManager.IndividualBaseInfoInquiry | 语法 | Public UserInfoVO IndividualBaseInfoInquiry(String userid); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 返回网站管理人员个人信息 |
| WebManager.IndividualBaseInfoModification | 语法 | Public ResultMsg IndividualBaseInfoModification(String userid,UserInfoVO vo); |
| 前置条件 | 网站管理人员已经登录 |
| 后置条件 | 更新网站管理人员信息 |

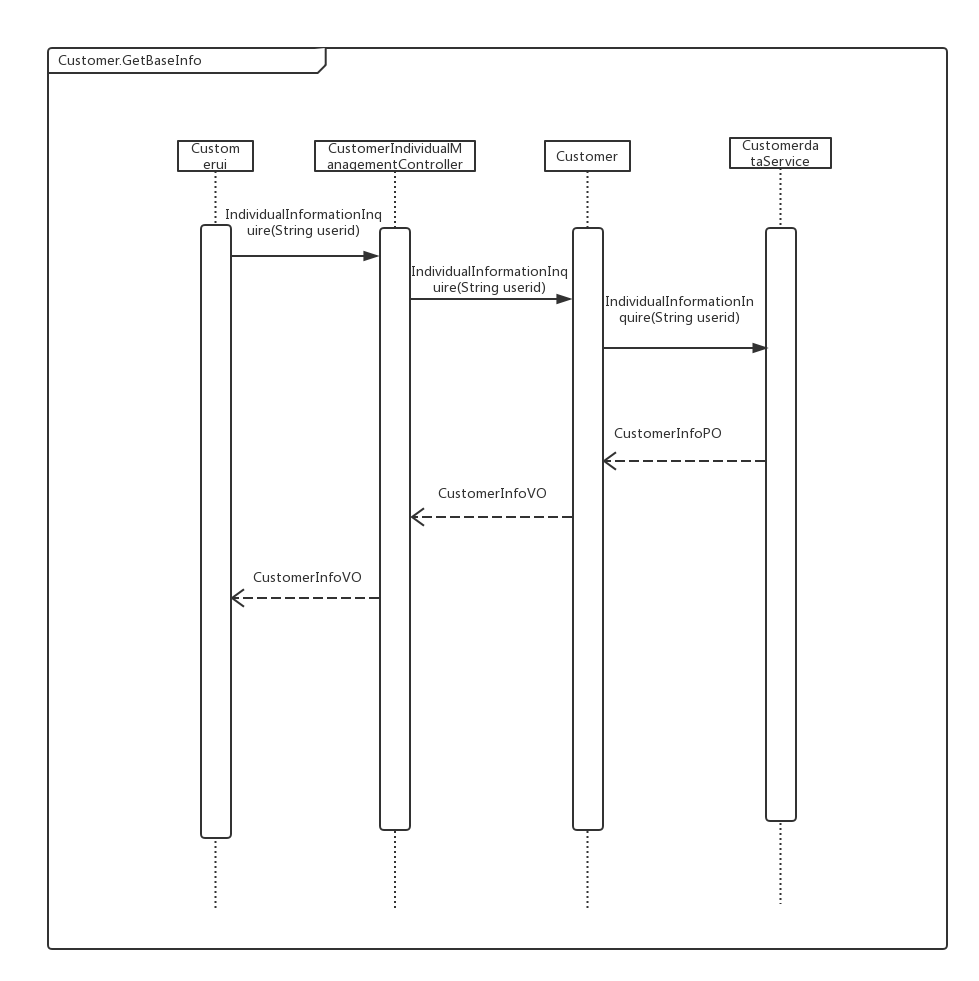
|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| UserManagementDataService.GetWebManagerInfo | 获取网站管理人员的信息 |
| UserManagementDataService.SetWebManagerInfo | 修改网站管理人员的信息 |
| HotelInfoMaintainBLService.addHotel | 再见酒店 |
| UserManagementDataService.HotelStuffScan | 查看酒店管理人员 |
| Md5Util.md5encode | 加密 |
| UserManagementDataService.AddHotelStuff | 添加酒店管理人员 |
| HotelInfoCheckBLService.hotelScan | 浏览酒店 |
| UserManagementDataService.GetWebStuffInfo | 获取网站营销人员信息 |
| UserManagementDataService.GetHotelStuffInfo | 获取酒店管理人员信息 |
| UserManagementDataService.GetCustomerInfo | 获取客户信息 |
| UserManagementDataService.SetWebStuffInfo | 修改网站营销人员信息 |
| UserManagementDataService.SetHotelStuffInfo | 修改酒店管理人员信息 |
| UserManagementDataService.SetCustomerInfo | 修改客户信息 |
| UserManagementDataService.AddWebStuff | 添加网站营销人员 |
| UserManagementDataService.WebStuffScan | 浏览网站营销人员列表 |

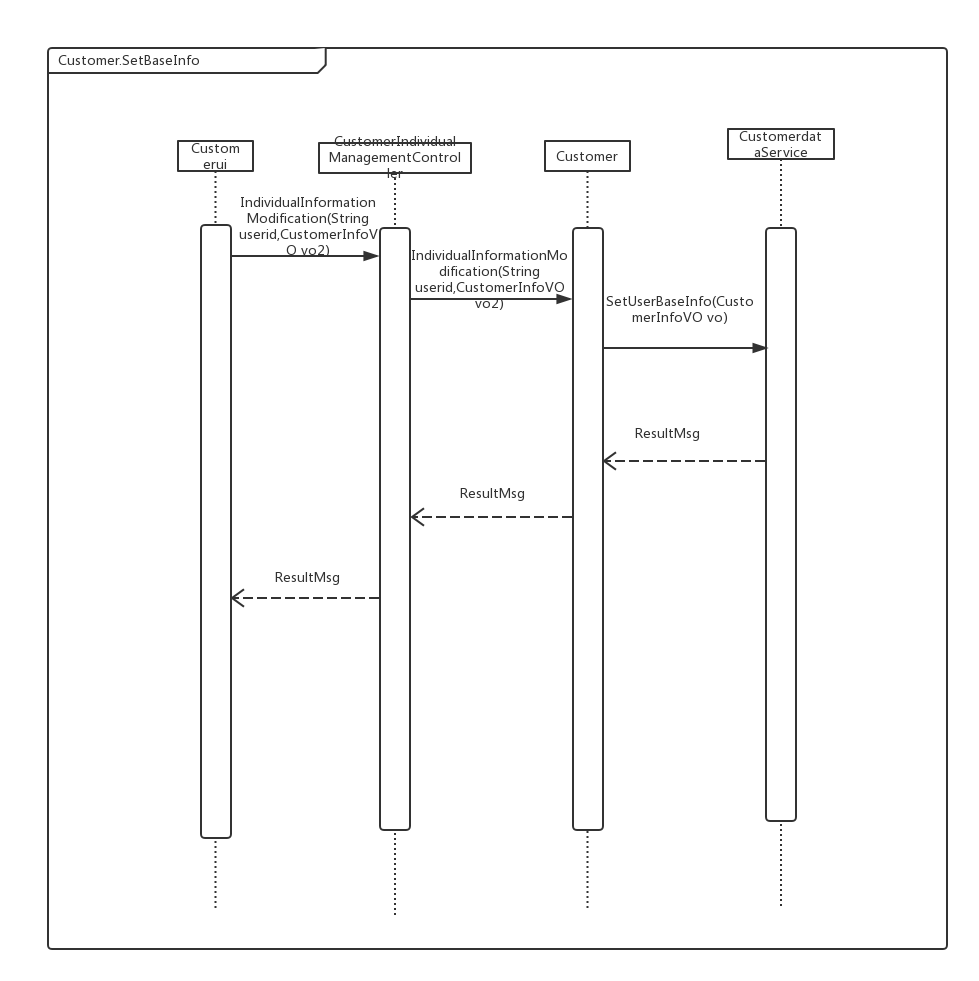
Md5Util接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| Md5Util.md5Encode | 语法 | Public String md5Encode(String s); |
| 前置条件 | 无 |
| 后置条件 | 返回加密字符串 |

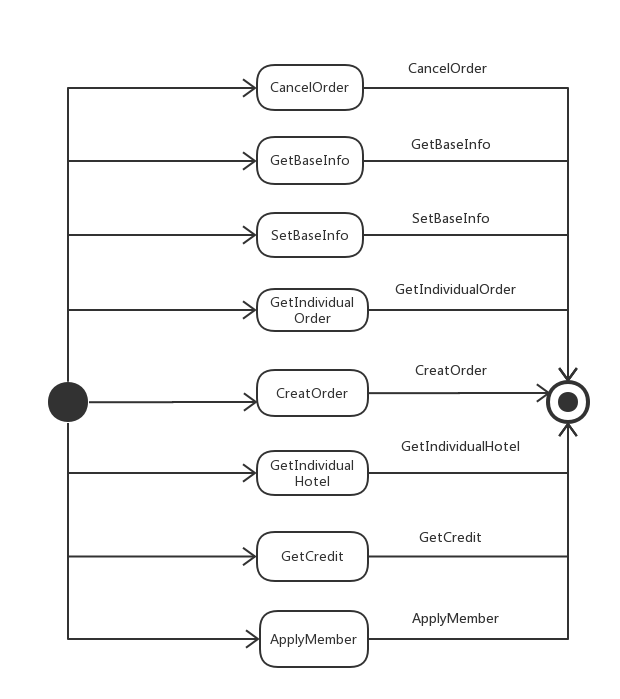
1. 业务逻辑层的动态模型

在用户模块中，查看个人信息、修改个人信息的顺序图如下图所示：





下图描述了Customer对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。



5.业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的控制器委托给不同的领域对象。

### Hotel模块

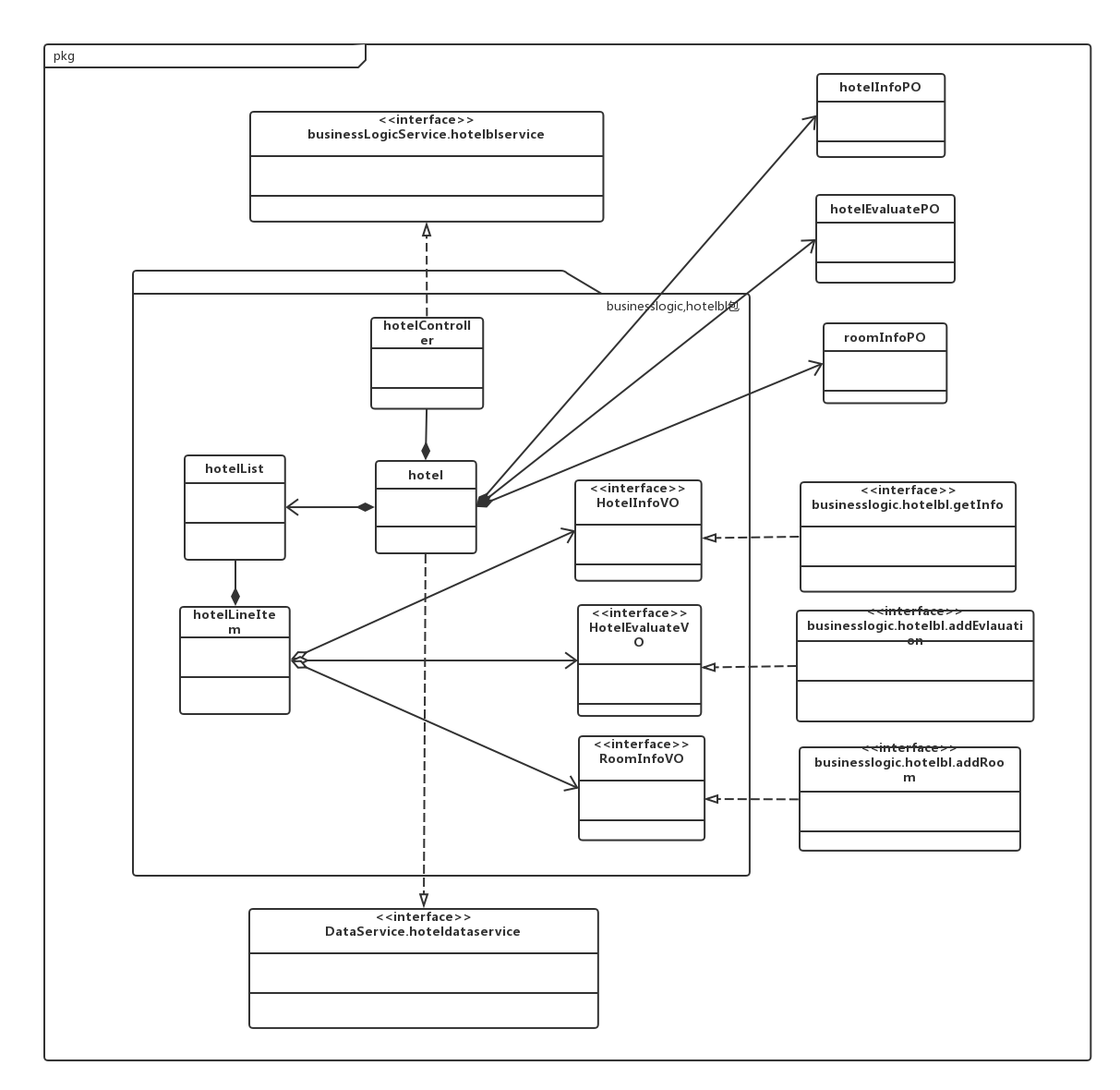
1、模块概述：

**hotel**模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

**hotel**模块的职责和接口参见软件系统结构描述文档内对该模块的描述。

2、整体结构：

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。在展示层和业务逻辑层之间添加HotelService接口。在业务逻辑层和数据层之间添加HotelDataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加HotelController，这样HotelController会将对酒店管理的业务逻辑处理委托给Hotel对象。HotelInfoPO、HotelEvaluatePO、RoomInfoPO是作为酒店信息的持久性对象被添加到设计模型中去的。



Hotelbl模块各个类的职责如表4-1-1-1所示。

**hotelbl模块各个类的职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| HotelController | 负责对应于酒店管理所需要的服务 |
| Hotel | 酒店管理的领域模型对象，拥有酒店的信息，可以帮助完成酒店管理界面所需要的服务 |

3、模块内部类的接口规范

Controller和Hotel的接口规范如下所示

**Controller的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| HotelSearchController.selectCondition | 语法 | public HotelMsg selectCondition(HotelInfoVO hotelConditionVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的searchHotel方法 |
| HotelSearchController.show | 语法 | public HotelMsg show(HotelInfoVO hotelListInfoVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的show方法 |
| HotelInfoCheckController.checkHotelInfo | 语法 | public HotelMsg checkHotelInfo(HotelInfoVO selectHotelVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的checkHotelInfo方法，系统返回酒店信息 |
| HotelReservedCheckController.checkReserved | 语法 | public HotelMsg checkReserved( HotelInfoVO reservedHotelVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的checkReserved方法，系统返回判断结果 |
| HotelReserveController.reserveHotel | 语法 | public HotelMsg reserveHotel(HotelInfoVO reserveHotelVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的reserveHotel方法 |
| HotelReserveController.createOrder | 语法 | public HotelMsg createOrder(HotelInfoVO orderInfoVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的createOrder方法，系统生成表单存储 |
| HotelEvaluateController.inputEvaluate | 语法 | public HotelMsg inputEvaluate(HotelEvaluateVO evaluateInfoVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的inputEvaluate方法，系统生成表单存储 |
| HotelEvaluateController.checkOrder | 语法 | public HotelMsg checkOrder(HotelEvaluateVO orderListVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的checkOrder方法，系统生成表单存储 |
| HotelEvaluateController.completeEvaluation | 语法 | public HotelMsg completeEvaluation( HotelEvaluateVO completeEvaluationVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的completeEvaluation方法，系统生成表单存储 |
| HotelInfoMaintainController.inputHotelInfo | 语法 | public HotelMsg inputHotelInfo(HotelInfoVO hotelInfoVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的inputHotelInfo方法，系统生成表单存储 |
| HotelInfoMaintainController.submitHotelInfo | 语法 | public HotelMsg submitInfo(HotelInfoVO hotelInfoVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的submitHotelInfo方法，系统生成表单存储 |
| RoomAddController.addRoom | 语法 | public HotelMsg addRoom(RoomInfoVO roomInfoVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的addRoom方法，系统生成表单存储 |
| RoomAddController.updateRoom | 语法 | public HotelMsg updateRoom(RoomInfoVO roomInfoVO) |
| 前置条件 | 已经创建一个hotel领域对象，并且输入符合语法规则 |
| 后置条件 | 调用Hotel类的updateRoom方法，系统生成表单存储 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| HotelSearch.createSearch (HotelInfoVO vo) | 生成搜索信息 |
| HotelInfoCheck.getInfo(HotelInfoVO vo) | 获取酒店信息 |
| HotelReservedCheck.getReservedInfo(HotelInfoVO vo) | 获取已预订酒店信息 |
| HotelReserve.addReservation(HotelInfoVO vo) | 生成预定 |
| HotelReserve.createOrder(HotelInfoVO vo) | 生成订单 |
| HotelEvaluate.addEvaluation(HotelEvaluateVO vo) | 添加评论信息 |
| HotelInfoMaintain.correctHotelInfo(HotelInfoVO vo) | 修改酒店信息 |
| RoomAdd.addRoom(RoomInfoVO vo) | 添加客房信息 |

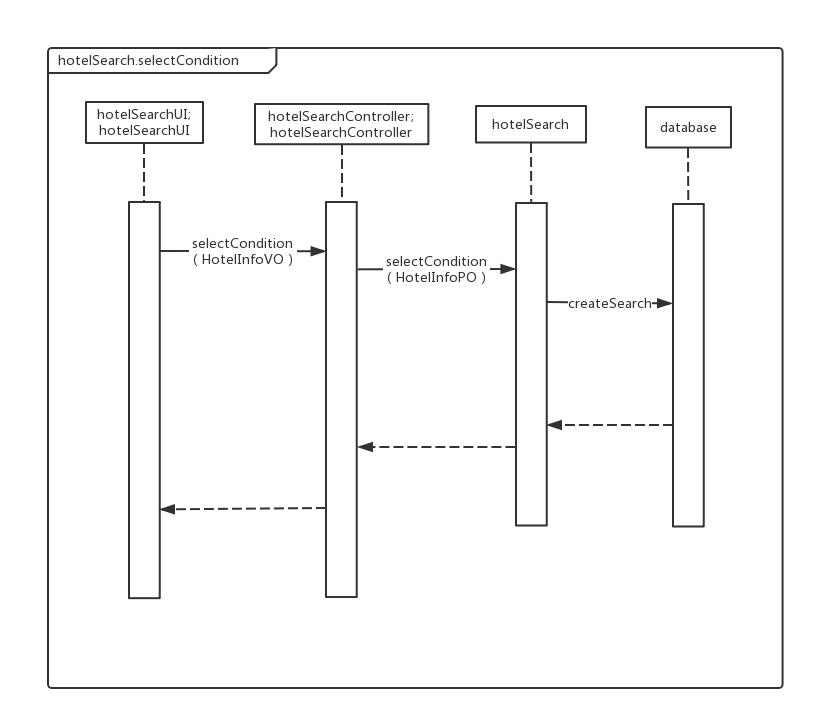
**hotelbl模块的接口规范**

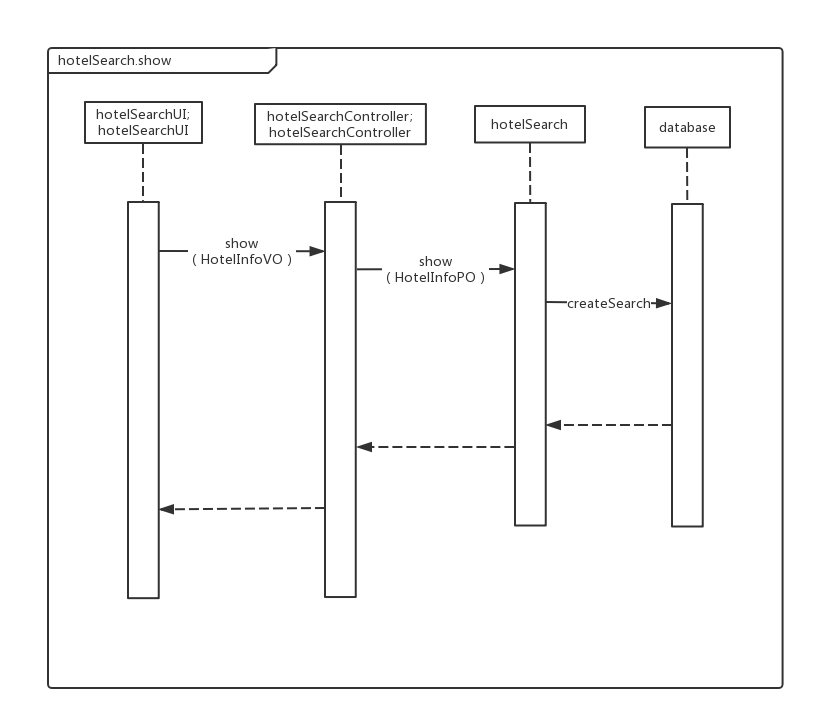
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| HotelSearch.selectCondition | 语法 | public HotelMsg selectCondition(HotelInfoVO hotelConditionVO) |
| 前置条件 | 客户选择酒店搜索条件，选择搜索酒店 |
| 后置条件 | 系统检查酒店条件，反馈检查结果 |
| HotelSearch.show | 语法 | public HotelMsg show(HotelInfoVO hotelListInfoVO) |
| 前置条件 | 酒店搜索条件符合 |
| 后置条件 | 系统反馈搜索结果 |
| HotelInfoCheck.checkHotelInfo | 语法 | public HotelMsg checkHotelInfo(HotelInfoVO selectHotelVO) |
| 前置条件 | 客户选择酒店，请求查看酒店信息 |
| 后置条件 | 系统反馈酒店信息 |
| HotelReservedCheck.checkReserved | 语法 | public HotelMsg checkReserved( HotelInfoVO reservedHotelVO) |
| 前置条件 | 客户请求查看预订过的酒店 |
| 后置条件 | 系统检查订单，反馈检查结果 |
| HotelReserve.reserveHotel | 语法 | public HotelMsg reserveHotel(HotelInfoVO reserveHotelVO) |
| 前置条件 | 客户选择酒店，提交预定请求 |
| 后置条件 | 系统检查客户信息与酒店客房信息，反馈检查结果 |
| HotelReserve.createOrder | 语法 | public HotelMsg createOrder(HotelInfoVO orderInfoVO) |
| 前置条件 | 检查结果符合要求 |
| 后置条件 | 系统生成预定订单，并反馈预定成功提示 |
| HotelEvaluate.inputEvaluate | 语法 | public HotelMsg inputEvaluate(HotelEvaluateVO evaluateInfoVO) |
| 前置条件 | 客户请求对酒店进行评价，并输入评价信息 |
| 后置条件 | 系统检查评价信息格式，反馈检查结果 |
| HotelEvaluate.checkOrder | 语法 | public HotelMsg checkOrder(HotelEvaluateVO orderListVO) |
| 前置条件 | 评价信息符合格式 |
| 后置条件 | 系统检查酒店是否在已执行订单中，并反馈检查结果 |
| HotelEvaluate.completeEvaluation | 语法 | public HotelMsg completeEvaluation( HotelEvaluateVO completeEvaluationVO) |
| 前置条件 | 检查结果酒店在已执行订单中 |
| 后置条件 | 系统更新评价信息，并提示评价成功 |
| HotelInfoMaintain.inputHotelInfo | 语法 | public HotelMsg inputHotelInfo(HotelInfoVO hotelInfoVO) |
| 前置条件 | 酒店工作人员输入维护酒店基本信息 |
| 后置条件 | 系统检查信息格式，并反馈检查结果 |
| HotelInfoMaintain.submitInfo | 语法 | public HotelMsg submitInfo(HotelInfoVO hotelInfoVO) |
| 前置条件 | 酒店基本信息符合格式 |
| 后置条件 | 系统更新信息，并反馈维护成功 |
| RoomAdd.addRoom | 语法 | public HotelMsg addRoom(RoomInfoVO roomInfoVO) |
| 前置条件 | 酒店工作人员添加可用客房 |
| 后置条件 | 系统检查客房信息是否符合要求，并反馈检查结果 |
| RoomAdd.updateRoom | 语法 | public HotelMsg updateRoom(RoomInfoVO roomInfoVO) |
| 前置条件 | 客房信息符合要求 |
| 后置条件 | 系统更新客房信息，并提示操作完成 |

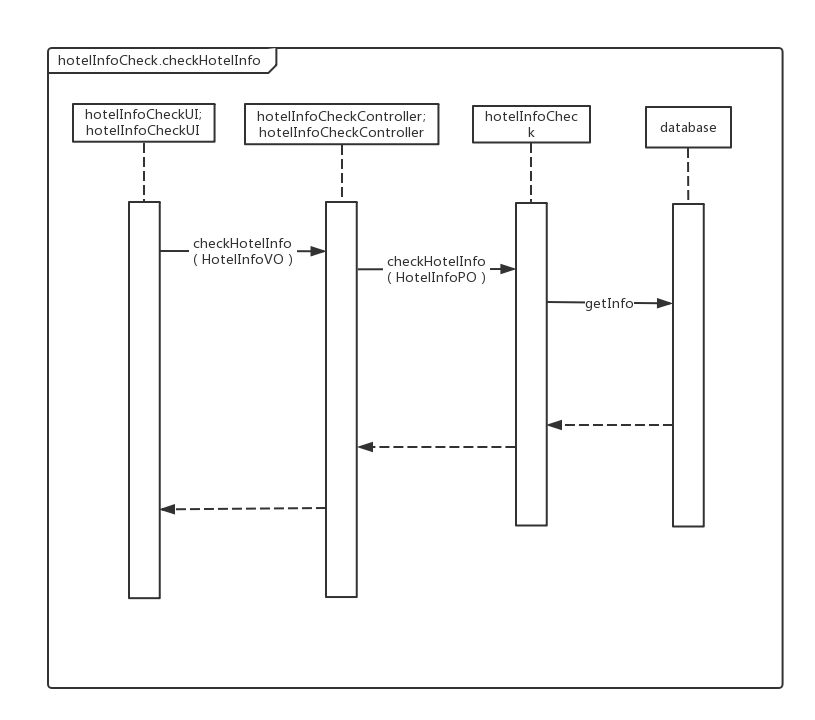
|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| HotelInfoDataService.insert(HotelInfoPO po) | 在数据库中新增一个酒店信息 |
| HotelInfoDataService.delete(HotelInfoPO po) | 在数据库中删除一个酒店信息 |
| HotelInfoDataService.update(HotelInfoPO po) | 在数据库中更新一个酒店信息 |
| HotelInfoDataService.find(HotelInfoPO po) | 在数据库中查找一个酒店信息 |
| HotelEvaluateDataService.insert(HotelEvaluatePO po) | 在数据库中新增一个酒店评价信息 |
| HotelEvaluateDataService.delete(HotelEvaluatePO po) | 在数据库中删除一个酒店评价信息 |
| HotelEvaluateDataService.find(HotelEvaluatePO po) | 在数据库中查找一个酒店评价信息 |
| RoomInfoDataService.insert(RoomInfoPO po) | 在数据库中新增一个可用客房信息 |
| RoomInfoDataService.delete(RoomInfoPO po) | 在数据库中删除一个可用客房信息 |
| RoomInfoDataService.update(RoomInfoPO po) | 在数据库中更新一个可用客房信息 |
| RoomInfoDataService.find(RoomInfoPO po) | 在数据库中查找一个可用客房信息 |

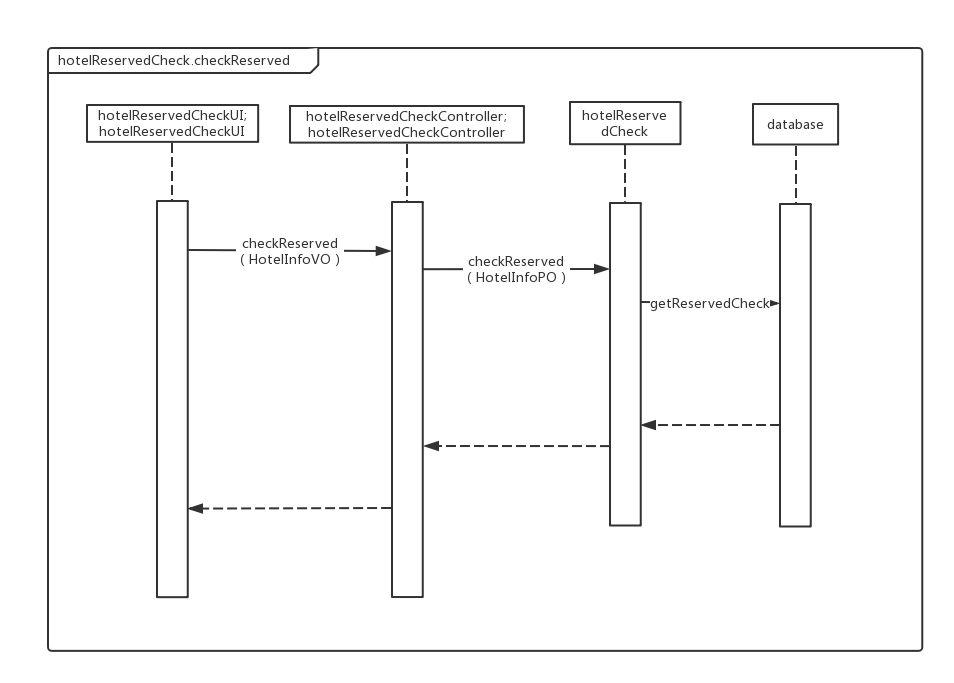
4、业务逻辑层的动态模型

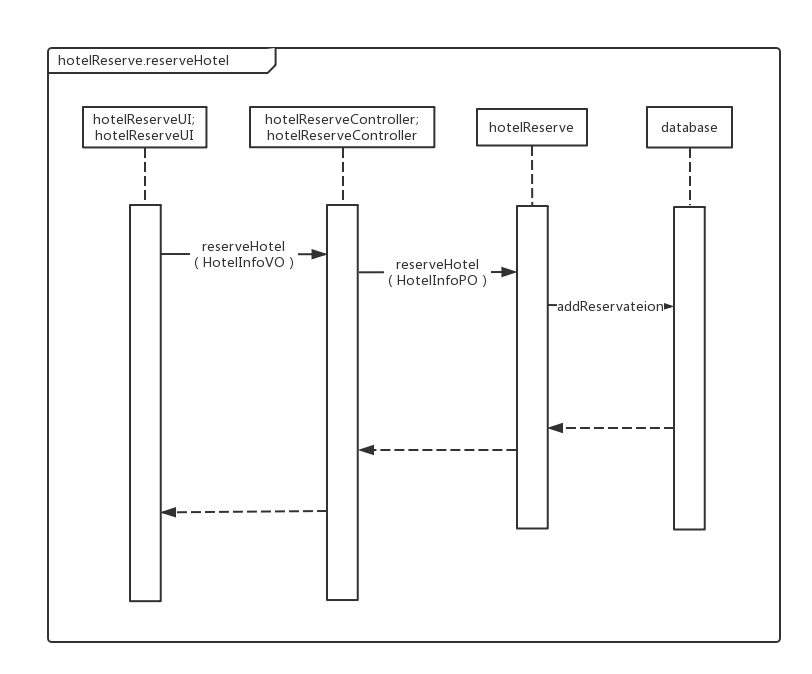
在酒店信息管理中，业务分为搜索酒店、查看酒店、预定酒店、评价酒店、维护酒店基本信息、添加客房，其顺序图如下图所示：

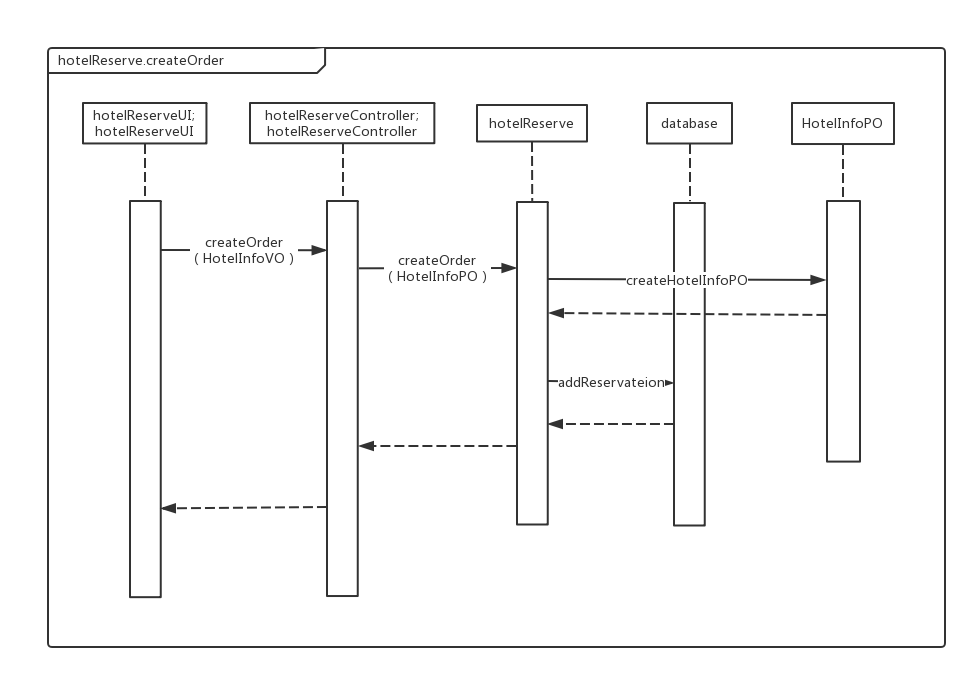


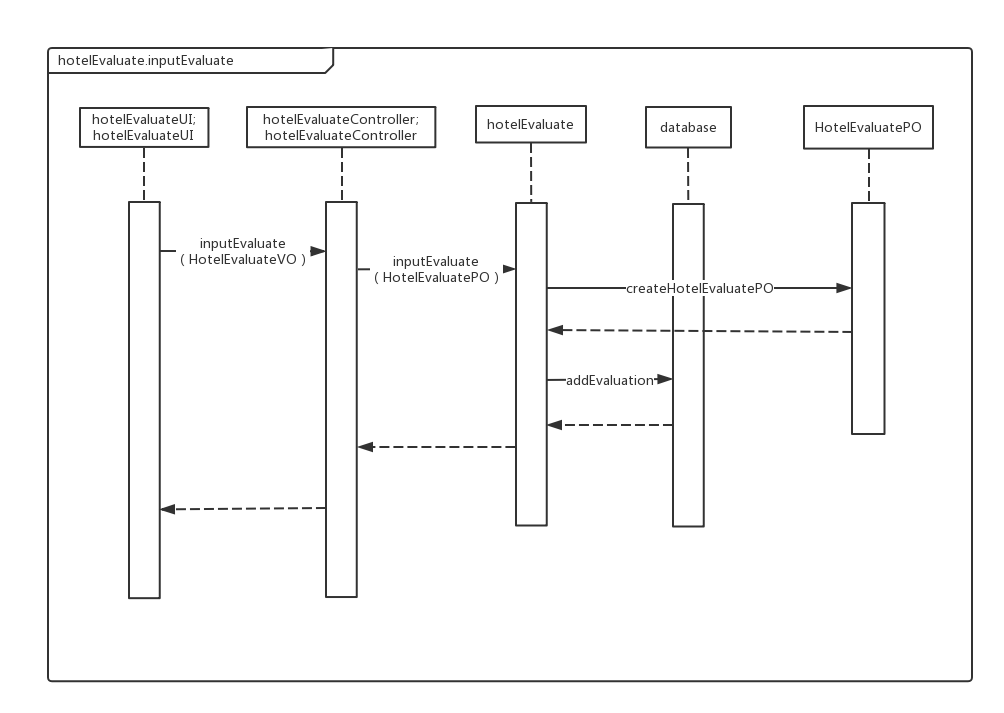


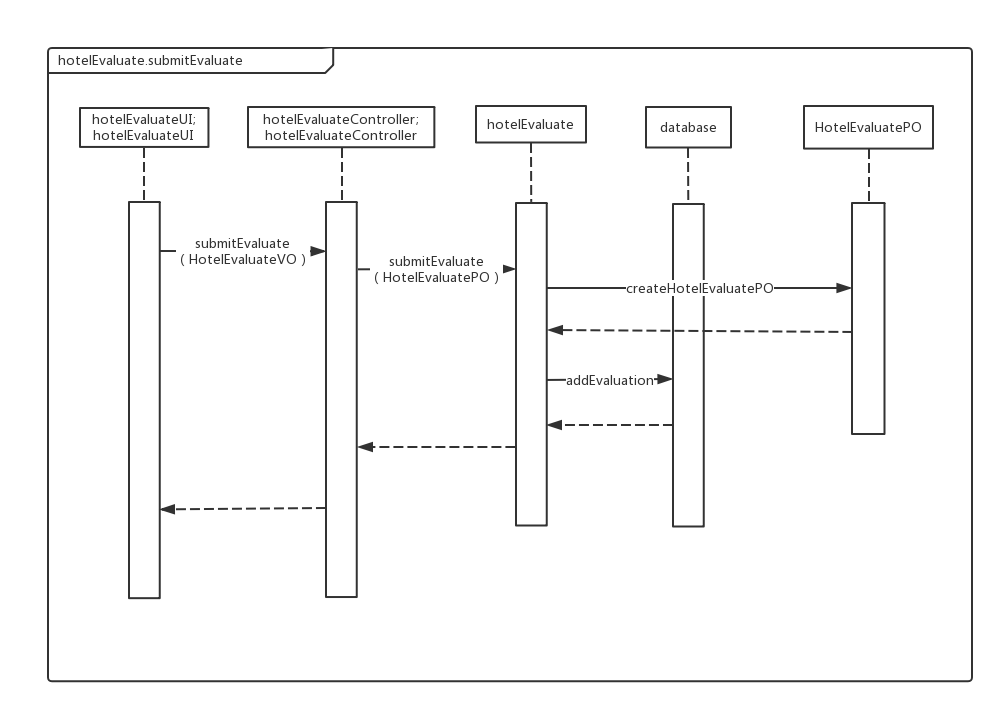


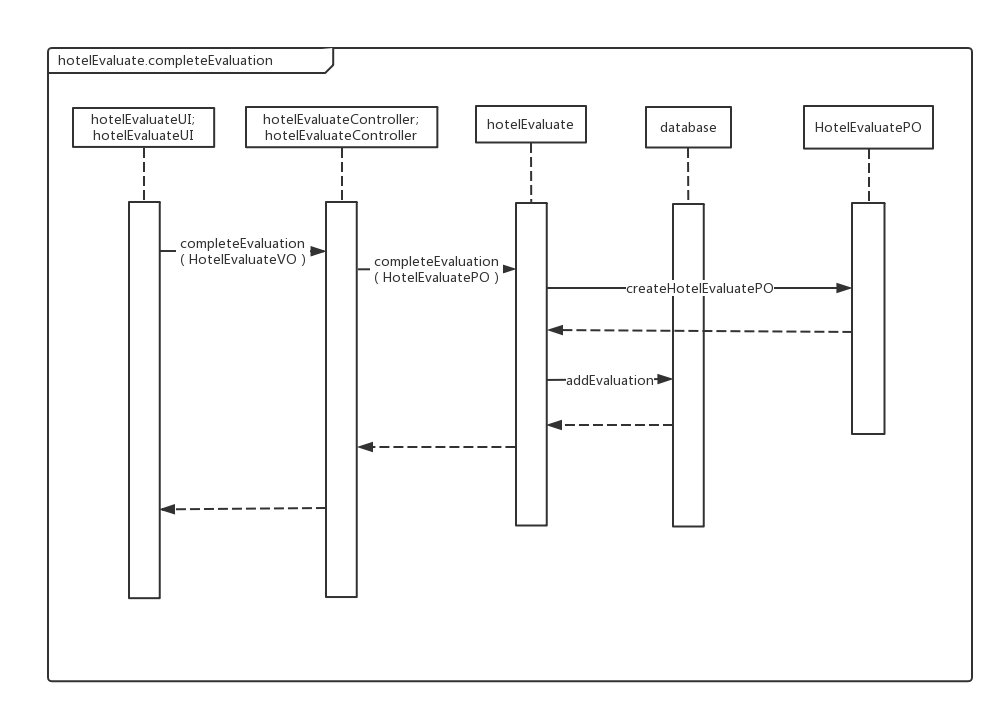


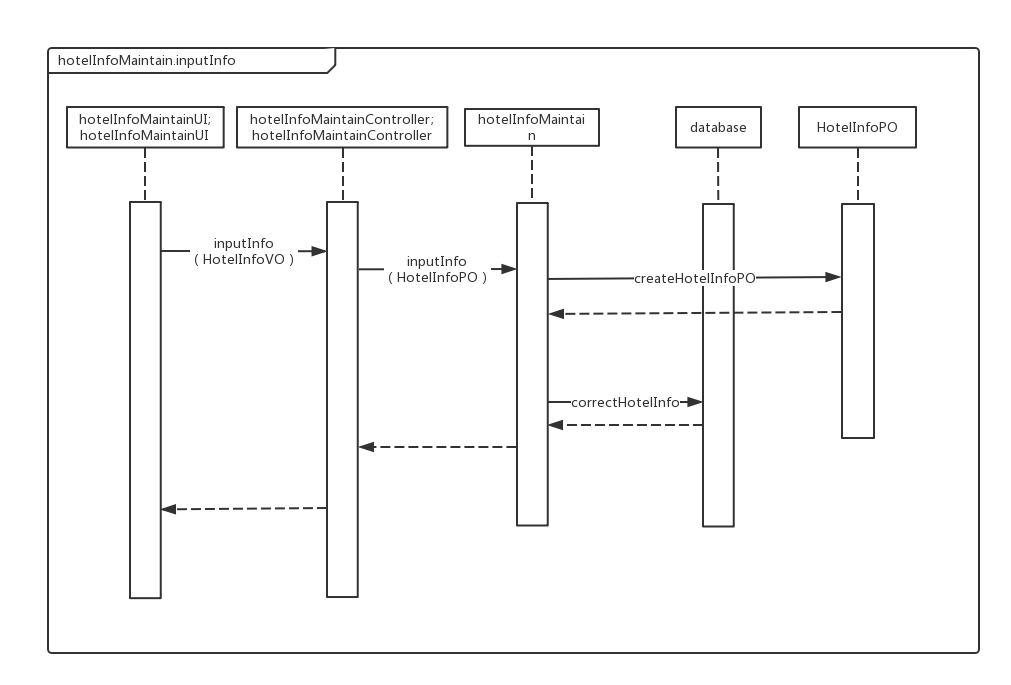


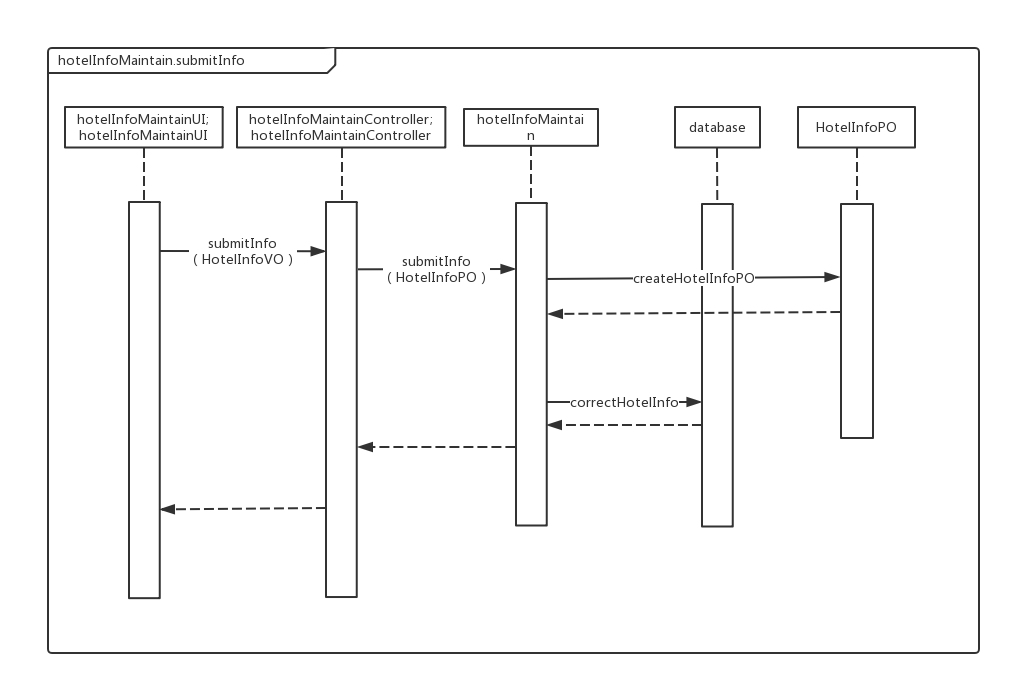


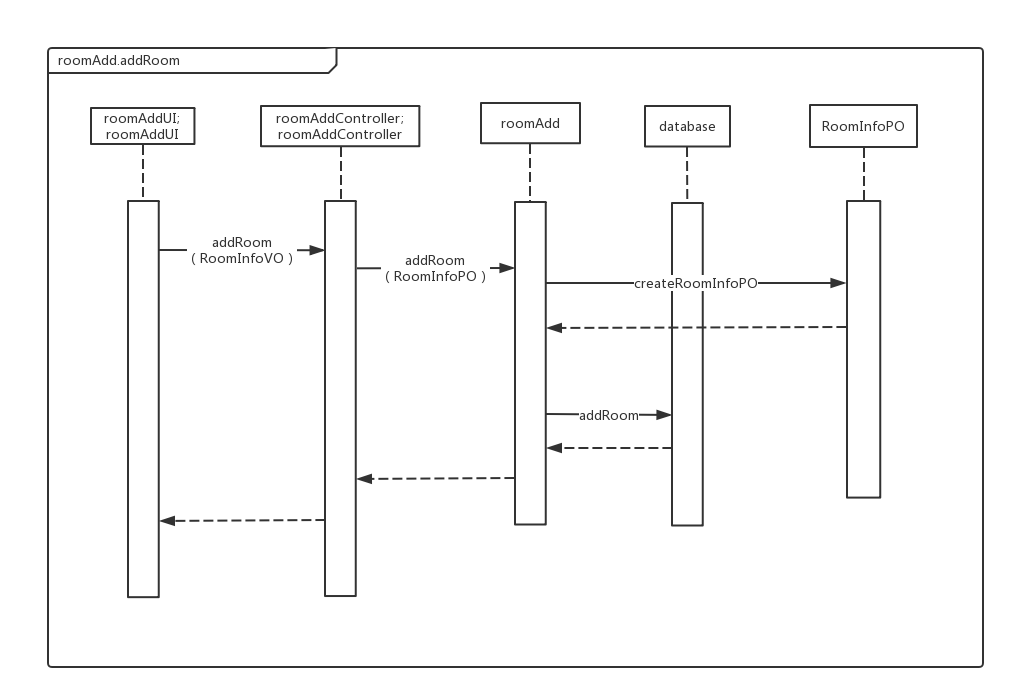


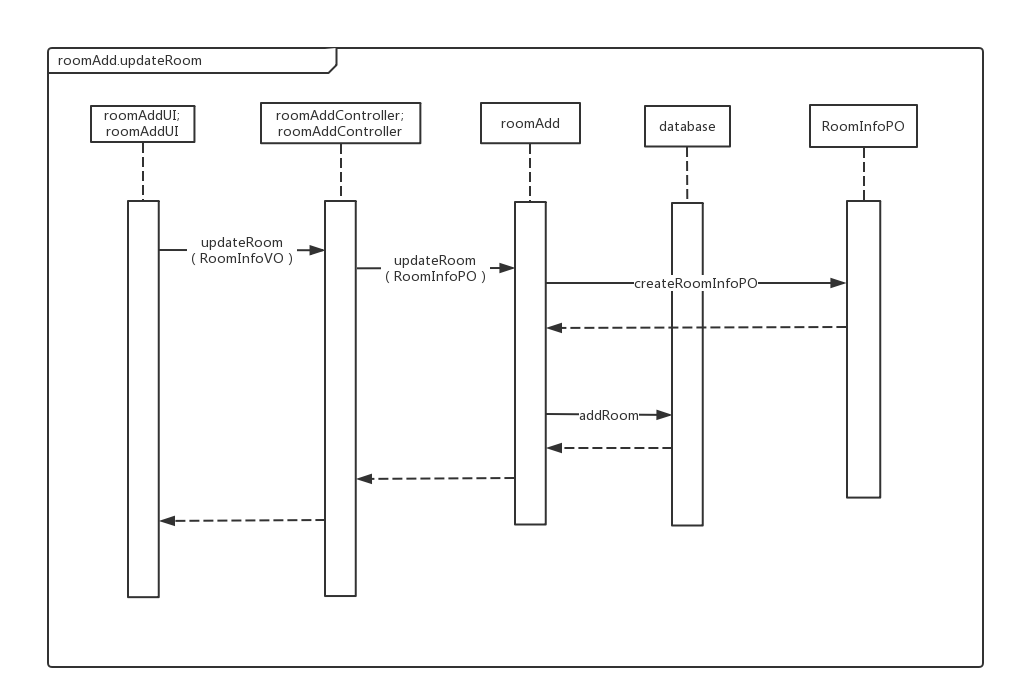




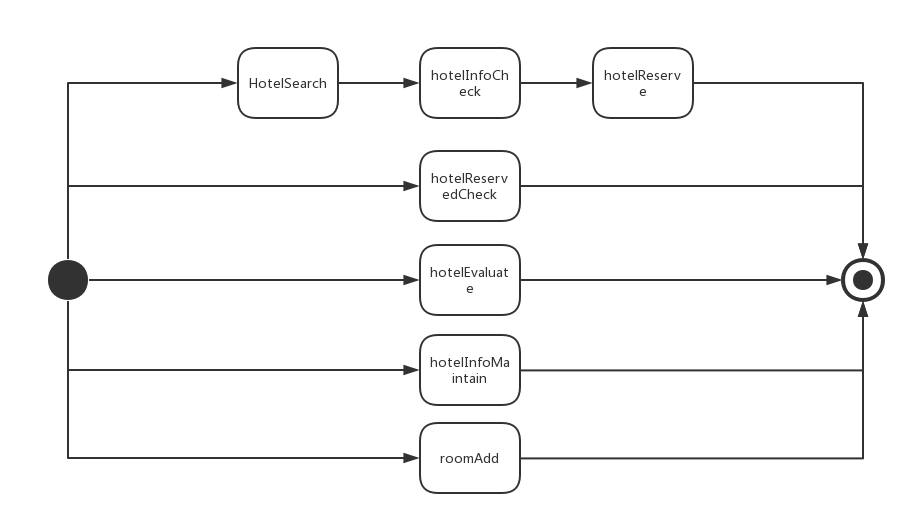








下图描述了hotel对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。



5.业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的控制器委托给不同的领域对象。

### Vip模块

1、模块概述

Vip模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

Vip模块的职责和接口参见软件系统结构描述文档内对该模块的描述。

2、整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。在展示层和业务逻辑层之间添加VipBLService接口。在业务逻辑层和数据层之间添加VipDataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加VipController，这样VipController会将对会员信息管理的业务逻辑处理委托给Vip对象。CommonVipPO和BusinessVipPO是作为账户信息的持久性对象被添加到设计模型中去的。

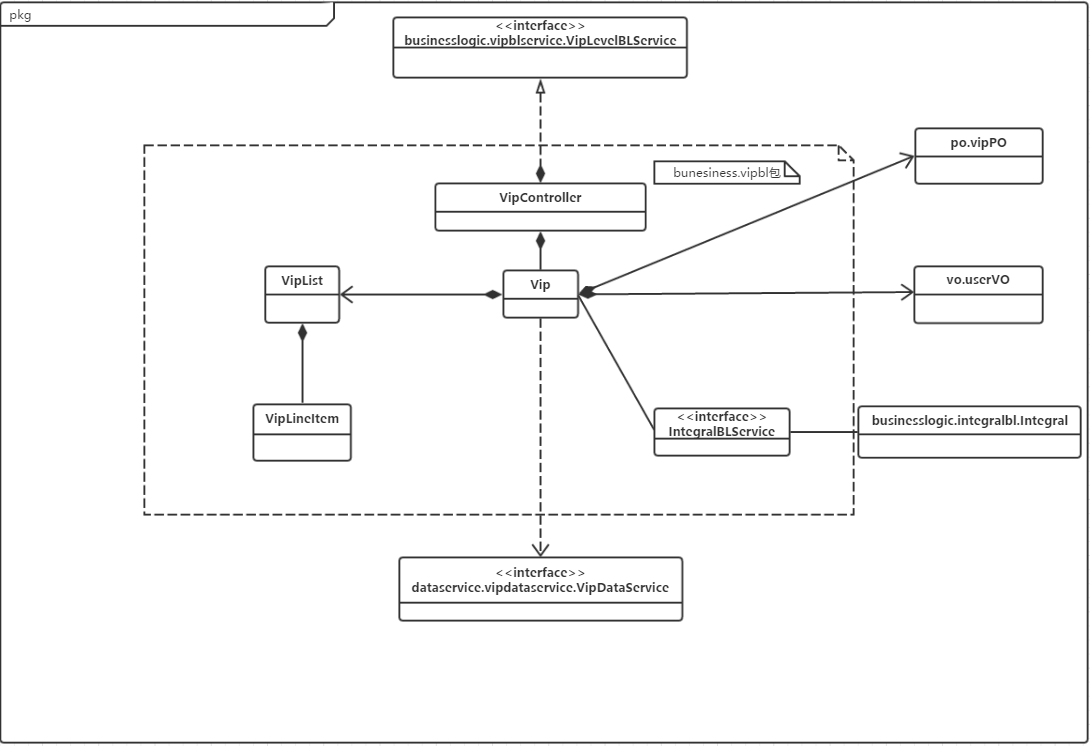


表4.2.4.1 vipbl模块各个类的设计

vipbl模块各个类的职责如表4.2.4.2所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| VipController | 负责实现会员部分所需要的服务 |
| Vip | 会员的领域模型对象，拥有会员等级划分的积分要求信息，可以解决用户当前信用值对应的会员等级问题 |

3、模块内部类的接口规范

**VipControl类的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| VipController.searchLevel | 语法 | public int searchLevel(CustomerInfoVO user) |
| 前置条件 | 已经创建一个vip领域对象，且输入格式符合规范 |
| 后置条件 | 调用vip类的searchLevel方法，系统自动返回当前用户对应的会员等级 |
| VipController.registerVip | 语法 | public ResultMsg registerVip(String userID,VipType type,String str) |
| 前置条件 | 已经创建一个customer领域对象，且输入格式符合规范 |
| 后置条件 | 调用vip类的registerVip方法，系统自动添加会员，并且持久化更新数据 |
| VipController.createLevellSystem | 语法 | public ResultMsg createLevelSystem(LevelSystemVO levelSystemVO) |
| 前置条件 | 已经创建一个vip领域对象，且输入格式符合规范 |
| 后置条件 | 调用vip类的createLevelSystem方法，系统自动添加一个会员等级，并且持久化更新vipVO |
| VipController.findByUserIDC | 语法 | public CommonVipPO findByUserIDC(String userID) |
| 前置条件 | 已经创建一个vip领域对象，且输入格式符合规范 |
| 后置条件 | 调用vip类的findByUserIDC方法，系统自动查找一个会员等级，并且持久化更新vipVO |
| VipController.findByUserIDB | 语法 | public BusinessVipPO findByUserIDB(String userID) |
| 前置条件 | 已经创建一个vip领域对象，且输入格式符合规范 |
| 后置条件 | 调用vip类的findByUserIDB方法，系统自动查找一个会员等级，并且持久化更新vipVO |
| VipController.updateC | 语法 | public ResultMsg updateC(CommonVipPO po) |
| 前置条件 | 已经创建一个vip领域对象，且输入格式符合规范 |
| 后置条件 | 调用vip类的updateC方法，系统自动更新一个会员等级，并且持久化更新vipVO |
| VipController.updateB | 语法 | public ResultMsg updateB(BusinessVipPO po) |
| 前置条件 | 已经创建一个vip领域对象，且输入格式符合规范 |
| 后置条件 | 调用vip类的updateB方法，系统自动更新一个会员等级，并且持久化更新vipVO |
|  |  |  |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| Vip.searchLevel | 获得对应的会员等级 | |
| Vip.createLevelSystem | 增加一个会员等级 | |
| Vip.findByUserIDB | 查找一个企业会员 | |
| Vip.findByUserIDC | 查找一个普通会员 | |
| Vip.updateB | 更新一个企业会员 | |
| Vip.updateC | 更新一个普通会员 | |

**Vipbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Vip.searchLevel | 语法 | public int searchLevel(userVO user) |
| 前置条件 | 用户获得当前积分对应的会员等级 |
| 后置条件 | 系统根据用户的信息，反馈当前用户的会员等级 |
| Vip.registerVip | 语法 | public ResultMsg registerVip（String userID,VipType type,String str） |
| 前置条件 | 用户注册会员 |
| 后置条件 | 系统根据用户的信息，反馈注册成功 |
| Vip.createLevelSystem | 语法 | public ResultMsg createLevelSystem(LevelSystemVO vo) |
| 前置条件 | 用户增加相应会员等级需要的信用值 |
| 后置条件 | 系统将添加相应等级需要的信用值 |
| Vip.findByUserIDC | 语法 | public ComonVipPO findByUserIDC（String userID） |
| 前置条件 | 用户id存在 |
| 后置条件 | 系统返回相应用户信息 |
| Vip.findByUserIDB | 语法 | Public BusinessVipPO findByUserIDB（String userID） |
| 前置条件 | 用户id存在 |
| 后置条件 | 系统返回相应用户信息 |
| Vip.updateC | 语法 | public ResultMsg updateC(CommonVipPO po) |
| 前置条件 | 用户存在 |
| 后置条件 | 系统更新相应会员信息 |
| Vip.updateB | 语法 | public ResultMsg updateB(BusinessVipPO po) |
| 前置条件 | 用户存在 |
| 后置条件 | 系统更新相应会员信息 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| VipDataService.getLevelSystemPO | 展示所有的等级信息 |
| VipDataService.insertB | 添加一个持久化的会员信息 |
| VipDataService.insertC | 添加一个持久化的会员信息 |
| VipDataService.findByUserB | 按照id找到一个持久化的会员信息 |
| VipDataService.findByUserC | 按照id找到一个持久化的会员信息 |
| VipDataService.findByUserB | 按照id找到一个持久化的会员信息 |
| VipDataService.findL | 按照等级信息找到一个持久化的等级信息 |
| VipDataService.insertL | 插入一个持久化的等级信息 |

4.业务逻辑层的动态模型

在会员业务中，分为会员等级需要的信用值查看，用户当前会员等级，修改会员等级需要的信用值功能。图4.2.4.3表示了得到会员等级和信用值对应关系的相关对象之间的协作。

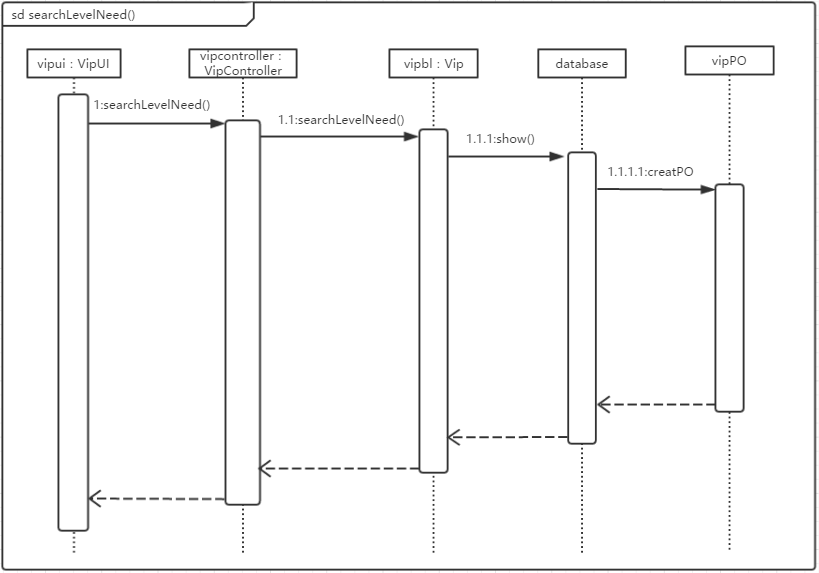


图4.2.4.3查询会员等级和信用值对应关系

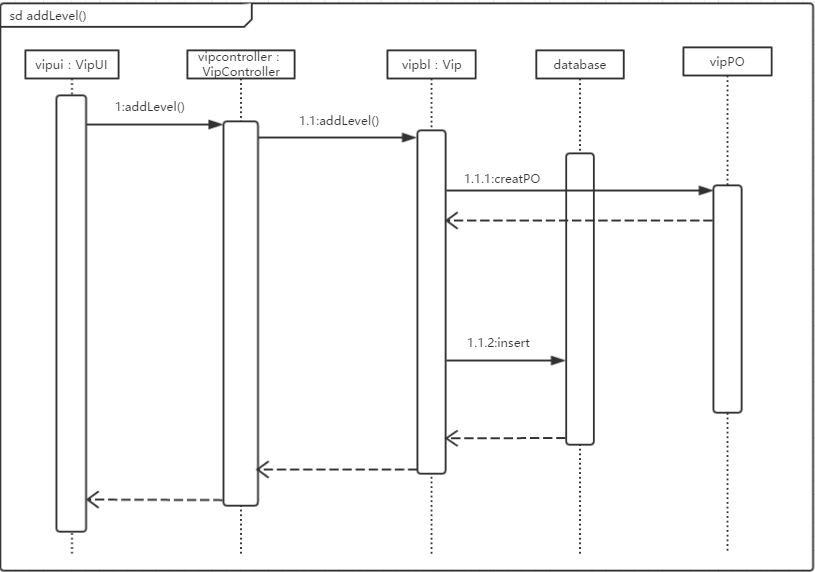
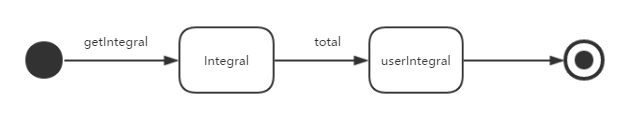


图4.2.4.3增加一个会员等级

下图所示的状态图描述了Vip对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。



5.业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的控制器委托给不同的领域对象。

### Credit模块

1、模块概述

Credit模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

Credit模块的职责和接口参见软件系统结构描述文档内对该模块的描述。

2、整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。在展示层和业务逻辑层之间添加CreditBLService接口。在业务逻辑层和数据层之间添加CreditDataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加CreditController，这样CreditController会将对信用值操作的业务逻辑处理委托给Credit对象。CreditPO是作为信用值的持久性对象被添加到设计模型中去的。

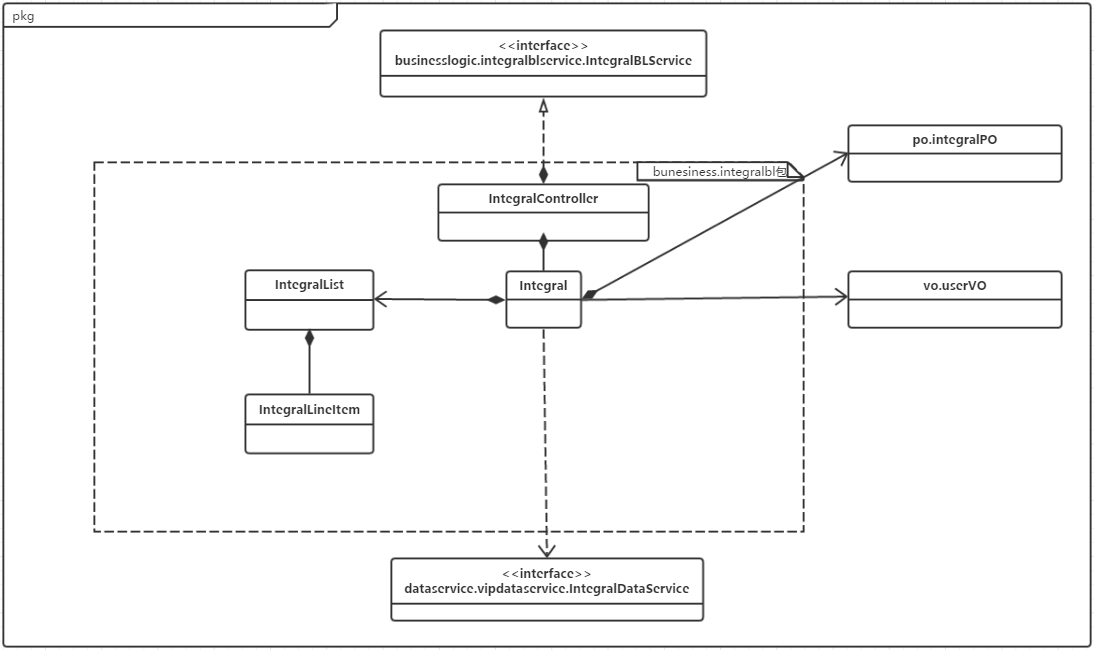


表4.2.5.1 creditbl模块各个类的设计

creditbl模块各个类的职责如表4.2.5.2所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| CreditController | 负责实现信用值部分所需要的服务 |
| Credit | 信用值的领域模型对象，拥有对应用户的信用值信息，可以解决用户信用值变化，修改问题 |

3、模块内部类的接口规范

**CreditControl类的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CreditController.getCredit | 语法 | public CreditVO getCredit(userVO client) |
| 前置条件 | 已经创建一个credit领域对象，且输入格式符合规范 |
| 后置条件 | 调用credit类的getCredit方法，系统返回当前用户信用值 |
| CreditController.addCredit | 语法 | public void addCredit(userVO client，int value) |
| 前置条件 | 已经创建一个credit领域对象，且输入格式符合规范 |
| 后置条件 | 调用credit类的addCredit方法，系统将要增加的信用值增加到该用户身上，并且持久化更新creditVO |
| CreditController.subCredit | 语法 | public void subCredit(userVO client，int value) |
| 前置条件 | 已经创建一个credit领域对象，且输入格式符合规范 |
| 后置条件 | 调用credit类的subCredit方法，系统将要减少的信用值从该用户减少，并且持久化更新creditVO |
| CreditController.changeCredit | 语法 | public void changeCredit(userVO client，int value) |
| 前置条件 | 已经创建一个credit领域对象，且输入格式符合规范 |
| 后置条件 | 调用credit类的changeCredit方法，系统该用户信用值修改为规定数值，并且持久化更新creditVO |
| CreditController.getCreditList | 语法 | Public ArrayList<CreditVO> getCreditList(String userID) |
| 前置条件 | 已经创建一个credit领域对象，且输入格式符合规范 |
| 后置条件 | 调用credit类的getCreditList方法，系统返回当前用户信用值 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| Credit.getCredit(userVO client) | 得到信用值单一化持久对象 | |
| Credit.addCredit(userVO client，int value) | 增加信用值数额 | |
| Credit.subCredit(userVO client，int value) | 减少信用值数额 | |
| Credit.changeCredit(userVO client，int value) | 修改信用值数额 | |
| Credit.getCreditList(String userID) | 得到信用值单一化持久对象列表 | |

**Creditbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Credit.getCredit | 语法 | public CreditVO getCredit(userVO client) |
| 前置条件 | 用户要获得当前的信用值 |
| 后置条件 | 系统获取当前用户的信用值，返回回来 |
| Credit.addCredit | 语法 | public void addCredit(userVO client，int value) |
| 前置条件 | 用户或系统要对当前客户的信用值进行增加 |
| 后置条件 | 系统将增加的数值增加到当前客户的信用值上 |
| Credit.subCredit | 语法 | public void subCredit(userVO client，int value) |
| 前置条件 | 用户或系统要对当前客户的信用值进行减少 |
| 后置条件 | 系统将增加的数值从当前客户的信用值减少 |
| Credit.changeCredit | 语法 | public void changeCredit(userVO client，int value) |
| 前置条件 | 用户或系统要对当前客户的信用值进行修改 |
| 后置条件 | 系统将用户信用值修改为当前数值 |
| Credit.getCreditList | 语法 | public ArrayList<CreditVO> getCreditList(String userID) |
| 前置条件 | 用户要获得当前的信用值列表 |
| 后置条件 | 系统获取当前用户的信用值列表，返回回来 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| CreditDataService.getListByUserID | 得到一个持久化的信用值列表 |

4.业务逻辑层的动态模型

在信用值业务中，分为信用值修改，信用值查看，信用值增加功能。图4.2.5.3表示了信用值修改的相关对象之间的协作。

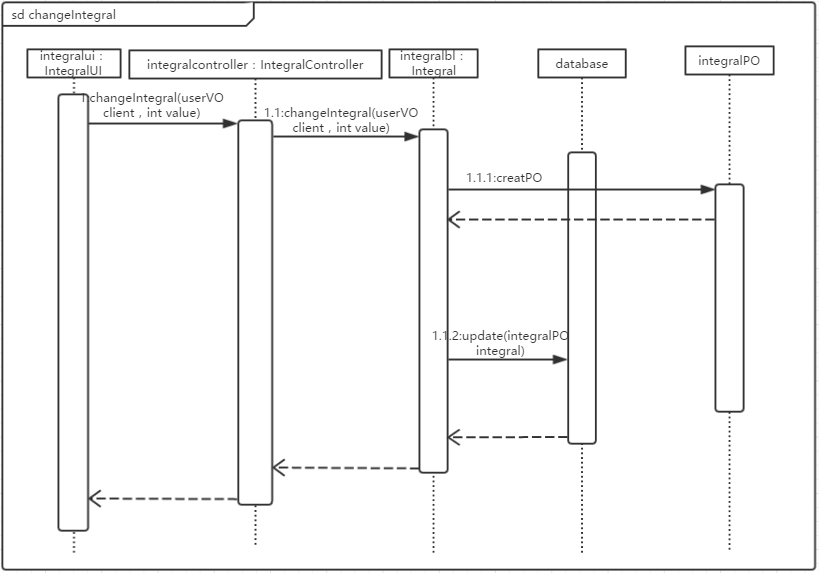
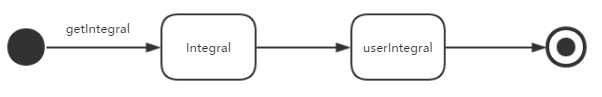


图4.2.5.3信用值修改

下图所示的状态图描述了Credit对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。



5.业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的控制器委托给不同的领域对象。

### Promotionbl模块

1、模块概述

Promotion模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

Promotion模块的职责和接口参见软件系统结构描述文档内对该模块的描述。

2、整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。在展示层和业务逻辑层之间添加PromotionBLService接口。在业务逻辑层和数据层之间添加PromotionDataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加PromotionHotelController，PromotionWebController，PromotionValueController，这样PromotionHotelController会将对信用值操作的业务逻辑处理委托给PromotionHotel对象，PromotionWebController会将对信用值操作的业务逻辑处理委托给PromotionWeb对象，PromotionValueController会将对信用值操作的业务逻辑处理委托给PromotionValue对象。PromotionPO是作为促销策略的持久性对象被添加到设计模型中去的。

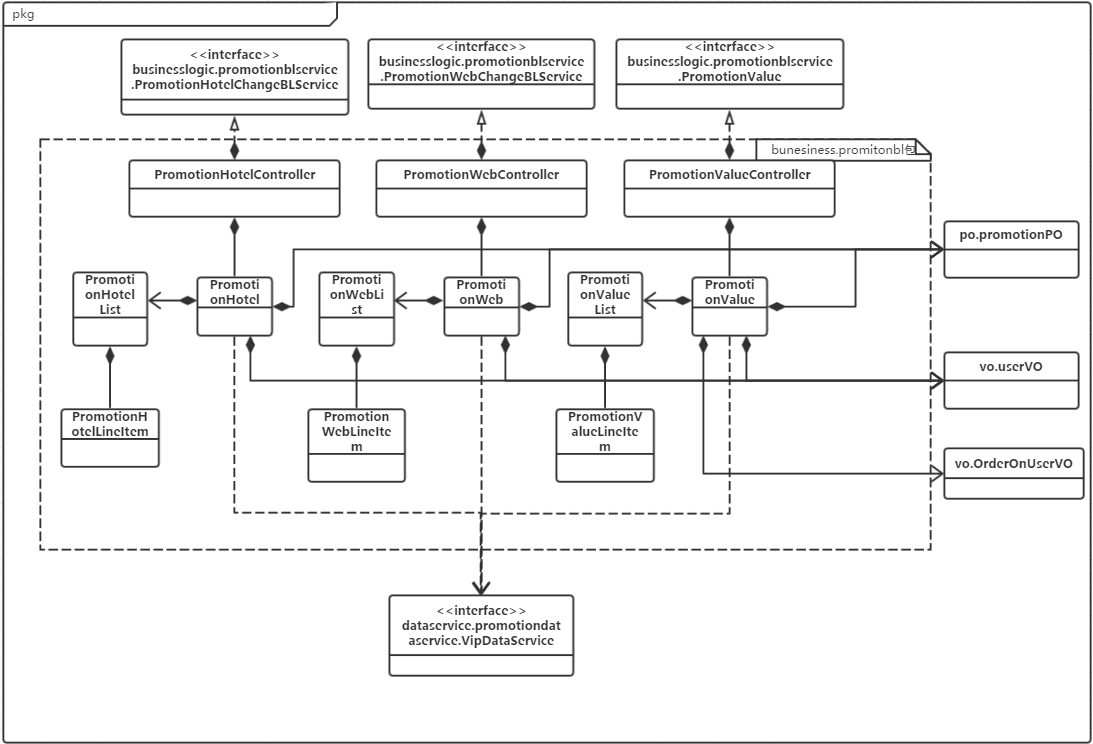


表4.2.6.1 promotionbl模块各个类的设计

promotionbl模块各个类的职责如表4.2.6.2所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| promotionHotelController | 负责实现酒店促销策略部分需要的服务 |
| PromotionHotel | 酒店促销策略的领域模型对象，拥有策略适用用户种类，促销策略降价比例，促销策略有效时间等信息，可以解决酒店促销策略的修改，增加，删除问题。 |
| promotionWebController | 负责实现网站促销策略部分需要的服务 |
| PromotionWeb | 网站促销策略的领域模型对象，拥有策略适用用户种类，促销策略降价比例，促销策略有效时间等信息，可以解决网站促销策略的修改，增加，删除问题。 |
| promotionValueController | 负责实现订单经过促销策略之后的订单价值部分所需要的服务 |
| PromotionValue | 订单价值计算的领域模型对象，拥有各种促销策略适用对象信息，能够解决促销策略匹配问题 |

3、模块内部类的接口规范

**promotionHotelControl类的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PromotionHotelController.changeBirthCut | 语法 | public void changeBirthCut(int level,double ratio,String hotelID) |
| 前置条件 | 已经创建一个PromotionHotel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的changeBirthCut方法，系统将生日促销策略进行修改，并且持久化更新promotionPO |
| PromotionHotelController.changeOverCut | 语法 | public void changeOverCut(int number,double ratio,String hotelID) |
| 前置条件 | 已经创建一个PromotionHotel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的changeOverCut方法，系统将三间以上促销策略进行修改，并且持久化更新promotionPO |
| PromotionHotelController.changeJoinCut | 语法 | public void changeJoinCut (String businessName,double ratio,String hotelID) |
| 前置条件 | 已经创建一个PromotionHotel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的changeJoinCut方法，系统将合作企业促销策略进行修改，并且持久化更新promotionPO |
| PromotionHotelController.changeHotelCustomCut | 语法 | public void changeHotelCustomCut (String timebegin,String timeover,double ratio,String hotelID) |
| 前置条件 | 已经创建一个PromotionHotel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的changeHotelCustomCut方法，系统将自定义促销策略进行修改，并且持久化更新promotionPO |
| PromotionHotelController.addBirthCut | 语法 | public void addBirthCut(int level,double ratio,String hotelID) |
| 前置条件 | 已经创建一个PromotionHotel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的addBirthCut方法，系统将生日促销策略进行添加，并且持久化更新promotionPO |
| PromotionHotelController.addOverCut | 语法 | public void addOverCut(int number,double ratio,String hotelID) |
| 前置条件 | 已经创建一个PromotionHotel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的addOverCut方法，系统将三间以上促销策略进行添加，并且持久化更新promotionPO |
| PromotionHotelController.addJoinCut | 语法 | public void addJoinCut (String businessName,double ratio,String hotelID) |
| 前置条件 | 已经创建一个PromotionHotel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的addJoinCut方法，系统将合作企业促销策略进行添加，并且持久化更新promotionPO |
| PromotionHotelController.addHotelCustomCut | 语法 | public void addHotelCustomCut (String timebegin,String timeover,double ratio,String hotelID) |
| 前置条件 | 已经创建一个PromotionHotel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的addHotelCustomCut方法，系统将自定义促销策略进行添加，并且持久化更新promotionPO |
| PromotionHotelController.deleteBirthCut | 语法 | public void deleteBirthCut(int level,String hotelID) |
| 前置条件 | 已经创建一个PromotionHotel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的deleteBirthCut方法，系统将生日促销策略进行删除，并且持久化更新promotionPO |
| PromotionHotelController.deleteOverCut | 语法 | public void deleteOverCut(int number,String hotelID) |
| 前置条件 | 已经创建一个PromotionHotel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的deleteOverCut方法，系统将三间以上促销策略进行删除，并且持久化更新promotionPO |
| PromotionHotelController.deleteJoinCut | 语法 | public void deleteJoinCut (String businessName,String hotelID) |
| 前置条件 | 已经创建一个PromotionHtoel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的deleteJoinCut方法，系统将合作企业促销策略进行删除，并且持久化更新promotionPO |
| PromotionHotelController.deleteHotelCustomCut | 语法 | public void deleteHotelCustomCut (String timebegin,String timeover,String hotelID) |
| 前置条件 | 已经创建一个PromotionHotel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的deleteHotelCustomCut方法，系统将自定义促销策略进行删除，并且持久化更新promotionPO |
| PromotionHotelController.getHotelPromotion(PromotionType,String hotelID) | 语法 | public ArrayList<PromotionHotelVO> getHotelPromotion(PromotionHotelType type,String hotelID) |
| 前置条件 | 已经创建一个PromotionHotel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHotel类的getHotelPromotion方法，系统将获得促销策略列表 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| PromotionHotel.changeBirthCut | 修改会员生日优惠政策 | |
| PromotionHotel.changeOverCut | 修改三间以上优惠政策 | |
| PromotionHotel.changeJoinCut | 修改合作企业优惠政策 | |
| PromotionHotel.changeHotelCustomCut | 修改酒店自定义优惠政策 | |
| PromotionHotel.addBirthCut | 添加会员生日优惠政策 | |
| PromotionHotel.addOverCut | 添加三间以上优惠政策 | |
| PromotionHotel.addJoinCut | 添加合作企业优惠政策 | |
| PromotionHotel.addHotelCustomCut | 添加酒店自定义优惠政策 | |
| PromotionHotel.deleteBirthCut | 删除会员生日优惠政策 | |
| PromotionHotel.deleteOverCut | 删除三间以上优惠政策 | |
| PromotionHotel.deleteJoinCut | 删除合作企业优惠政策 | |
| PromotionHotel.deleteHotelCustomCut | 删除酒店自定义优惠政策 | |
| PromotionHotel.getHotelPromotion | 获得促销策略列表 | |

**promotionWebControl类的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PromotionWebController.changeLevelCut | 语法 | public void changeLevelCut(int level,double ratio) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的changeLevelCut方法，系统将会员等级优惠策略进行修改，并且持久化更新promotionPO |
| PromotionWebController.changeCircleCut | 语法 | public void changeCircleCut(Area location,double ratio) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的changeCircleCut方法，系统将特定商圈优惠策略进行修改，并且持久化更新promotionPO |
| PromotionWebController.changeWebCustomCut | 语法 | public void changeWebCustomCut(String timeBegin,String timeOver,double ratio) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的changeWebCustomCut方法，系统将自定义优惠策略进行修改，并且持久化更新promotionPO |
| PromotionWebController.deleteLevelCut | 语法 | public void deleteLevelCut(int level) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的deleteLevelCut方法，系统将会员等级优惠策略进行删除，并且持久化更新promotionPO |
| PromotionWebController.deleteCircleCut | 语法 | public void deleteCircleCut(Area location) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的deleteCircleCut方法，系统将特定商圈优惠策略进行删除，并且持久化更新promotionPO |
| PromotionWebController.deleteWebCustomCut | 语法 | public void deleteWebCustomCut(String timeBegin,String timeOver) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的deleteWebCustomCut方法，系统将自定义优惠策略进行删除，并且持久化更新promotionPO |
| PromotionWebController.addLevelCut | 语法 | public void addLevelCut(int level,double ratio) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的addLevelCut方法，系统将会员等级优惠策略进行添加，并且持久化更新promotionPO |
| PromotionWebController.addCircleCut | 语法 | public void addCircleCut(Area location,double ratio) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的addCircleCut方法，系统将特定商圈优惠策略进行添加，并且持久化更新promotionPO |
| PromotionWebController.addWebCustomCut | 语法 | public void addWebCustomCut(String timeBegin,String timeOver,double ratio) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的addWebCustomCut方法，系统将自定义优惠策略进行添加，并且持久化更新promotionPO |
| PromotionWebController.getWebPromotion(PromotionType,String hotelID) | 语法 | public ArrayList<PromotionWebVO> getWebPromotion(PromotionWebType type) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的getWebPromotion方法，系统将获得促销策略列表 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| PromotionWeb.changeLevelCut | 修改会员等级优惠策略 | |
| PromotionWeb.changeCircleCut | 修改会员特定商圈优惠策略 | |
| PromotionWeb.changeWebCustomCut | 修改网站自定义优惠策略 | |
| PromotionWeb.changeLevelCut | 删除会员等级优惠策略 | |
| PromotionWeb.changeCircleCut | 删除会员特定商圈优惠策略 | |
| PromotionWeb.changeWebCustomCut | 删除网站自定义优惠策略 | |
| PromotionWeb.changeLevelCut | 添加会员等级优惠策略 | |
| PromotionWeb.changeCircleCut | 添加会员特定商圈优惠策略 | |
| PromotionWeb.changeWebCustomCut | 添加网站自定义优惠策略 | |
| PromotionWeb.getWebPromotion | 获得促销策略列表 | |

**promotionValueControl类的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PromotionValueControl.getValue | 语法 | public promotionVO getValue(CustimerInfoVO user, OrderVO orderVO,PromotionHotelType type) |
| 前置条件 | 已经创建一个PromotionValue领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionValue类的getValue方法，系统自动计算订单价值，并且持久化更新promotionPO |
| PromotionValueControl.getValue | 语法 | public promotionVO getValue(CustimerInfoVO user, OrderVO orderVO,PromotionWebType type) |
| 前置条件 | 已经创建一个PromotionValue领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionValue类的getValue方法，系统自动计算订单价值，并且持久化更新promotionPO |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| PromotionValue.getValue | 获得该订单经过打折之后的信息 | |

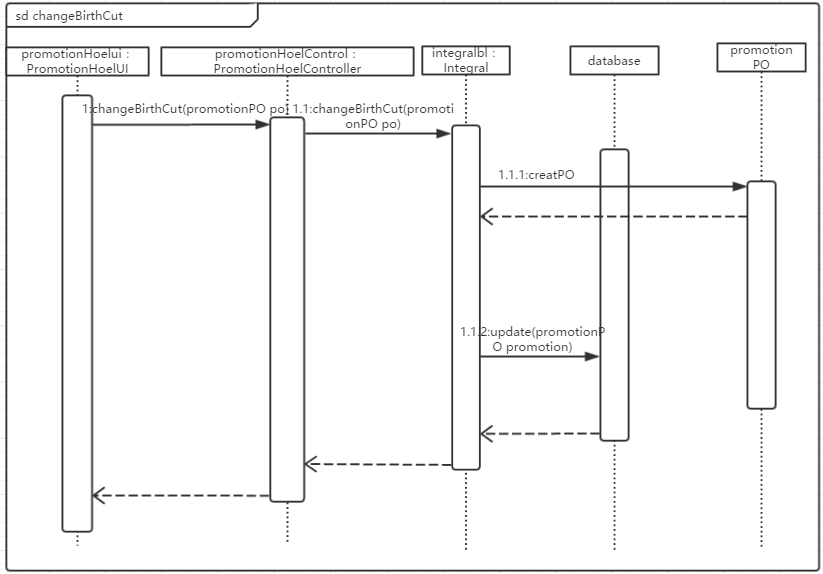
**promotionbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PromotionHotel.changeBirthCut | 语法 | public void changeBirthCut(promotionPO po) |
| 前置条件 | 酒店工作人员要修改酒店会员生日促销策略 |
| 后置条件 | 系统将酒店会员生日促销策略降价比例修改为ratio |
| PromotionHotel.changeOverCut | 语法 | public void changeOverCut(promotionPO po) |
| 前置条件 | 酒店工作人员要修改酒店三间以上优惠策略 |
| 后置条件 | 系统将酒店三间以上促销策略降价比例修改为ratio |
| PromotionHotel.changeJoinCut | 语法 | public void changeJoinCut (promotionPO po) |
| 前置条件 | 酒店工作人员要修改酒店合作企业客户优惠策略 |
| 后置条件 | 系统将酒店合作企业促销策略降价比例修改为ratio |
| PromotionHotel.changeHotelCustomCut | 语法 | public void changeHotelCustomCut (promotionPO po) |
| 前置条件 | 酒店工作人员要修改酒店自定义优惠策略 |
| 后置条件 | 系统将酒店自定义优惠策略修改为当前设定 |
| PromotionWeb.changeLevelCut | 语法 | public void changeLevelCut(promotionPO po) |
| 前置条件 | 网站营销人员修改网站会员等级优惠 |
| 后置条件 | 系统将网站会员等级优惠修改为ratio |
| PromotionWeb.changeCircleCut | 语法 | public void changeCircleCut(promotionPO po) |
| 前置条件 | 网站营销人员修改网站vip商圈专属折扣优惠 |
| 后置条件 | 系统将网站vip商圈专属优惠折扣修改为ratio |
| PromotionWeb.changeWebCustomCut | 语法 | public void changeWebCustomCut(promotionPO po) |
| 前置条件 | 网站营销人员修改网站自定义促销策略 |
| 后置条件 | 系统将网站自定义促销策略修改为当前设定 |
| PromotionHotel.addBirthCut | 语法 | public void addBirthCut(promotionPO po) |
| 前置条件 | 酒店工作人员要添加酒店会员生日促销策略 |
| 后置条件 | 系统将酒店会员生日促销策略降价比例添加为ratio |
| PromotionHotel.addOverCut | 语法 | public void addOverCut(promotionPO po) |
| 前置条件 | 酒店工作人员要添加酒店三间以上优惠策略 |
| 后置条件 | 系统将酒店三间以上促销策略降价比例添加为ratio |
| PromotionHotel.addJoinCut | 语法 | public void addJoinCut (promotionPO po) |
| 前置条件 | 酒店工作人员要添加酒店合作企业客户优惠策略 |
| 后置条件 | 系统将酒店合作企业促销策略降价比例添加为ratio |
| PromotionHotel.addHotelCustomCut | 语法 | public void addHotelCustomCut (promotionPO po) |
| 前置条件 | 酒店工作人员要添加酒店自定义优惠策略 |
| 后置条件 | 系统将酒店自定义优惠策略添加为当前设定 |
| PromotionWeb.addLevelCut | 语法 | public void addLevelCut(promotionPO po) |
| 前置条件 | 网站营销人员添加网站会员等级优惠 |
| 后置条件 | 系统将网站会员等级优惠添加为ratio |
| PromotionWeb.addCircleCut | 语法 | public void addCircleCut(promotionPO po) |
| 前置条件 | 网站营销人员添加网站vip商圈专属折扣优惠 |
| 后置条件 | 系统将网站vip商圈专属优惠折扣添加为ratio |
| PromotionWeb.addWebCustomCut | 语法 | public void addWebCustomCut(promotionPO po) |
| 前置条件 | 网站营销人员添加网站自定义促销策略 |
| 后置条件 | 系统将网站自定义促销策略添加为当前设定 |
| PromotionHotel.deleteBirthCut | 语法 | public void deleteBirthCut(promotionPO po) |
| 前置条件 | 酒店工作人员要删除酒店会员生日促销策略 |
| 后置条件 | 系统将酒店会员生日促销策略降价比例删除为ratio |
| PromotionHotel.deleteOverCut | 语法 | public void deleteOverCut(promotionPO po) |
| 前置条件 | 酒店工作人员要删除酒店三间以上优惠策略 |
| 后置条件 | 系统将酒店三间以上促销策略降价比例删除为ratio |
| PromotionHotel.deleteJoinCut | 语法 | public void deleteJoinCut (promotionPO po) |
| 前置条件 | 酒店工作人员要删除酒店合作企业客户优惠策略 |
| 后置条件 | 系统将酒店合作企业促销策略降价比例删除为ratio |
| PromotionHotel.deleteHotelCustomCut | 语法 | public void deleteHotelCustomCut (promotionPO po) |
| 前置条件 | 酒店工作人员要删除酒店自定义优惠策略 |
| 后置条件 | 系统将酒店自定义优惠策略删除为当前设定 |
| PromotionWeb.deleteLevelCut | 语法 | public void deleteLevelCut(promotionPO po) |
| 前置条件 | 网站营销人员删除网站会员等级优惠 |
| 后置条件 | 系统将网站会员等级优惠删除为ratio |
| PromotionWeb.deleteCircleCut | 语法 | public void deleteCircleCut(promotionPO po) |
| 前置条件 | 网站营销人员删除网站vip商圈专属折扣优惠 |
| 后置条件 | 系统将网站vip商圈专属优惠折扣删除为ratio |
| PromotionWeb.deleteWebCustomCut | 语法 | public void deleteWebCustomCut(promotionPO po) |
| 前置条件 | 网站营销人员删除网站自定义促销策略 |
| 后置条件 | 系统将网站自定义促销策略删除为当前设定 |
| PromotionValue.getValue | 语法 | public promotionVO getValue(userVO user, OrderOnUserVO orderVO,PromotionHotelType type) |
| 前置条件 | 订单要利用促销策略计算订单价值 |
| 后置条件 | 系统返回订单经过促销策略打折后的价值 |
| PromotionValue.getValue | 语法 | public promotionVO getValue(userVO user, OrderOnUserVO orderVO,PromotionWebType type) |
| 前置条件 | 订单要利用促销策略计算订单价值 |
| 后置条件 | 系统返回订单经过促销策略打折后的价值 |

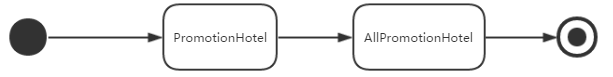
|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| PromotionHotelDataService.insert | 增加一个促销策略 |
| PromotionHotelDataService.deleteBirthCut | 删除一个生日特惠促销策略 |
| PromotionHotelDataService.deleteOverCut | 删除一个三间及以上预订特惠促销策略 |
| PromotionHotelDataService.deleteJoin | 删除一个合作企业客户促销策略 |
| PromotionHotelDataService.deleteHotelCustomCut | 删除一个特定期间折扣促销策略 |
| PromotionHotelDataService.update | 更新一个促销策略 |
| PromotionHotelDataService.findByType | 根据促销策略种类找到一个促销策略 |
| PromotionWebDataService.insert | 增加一个促销策略 |
| PromotionWebDataService.deleteLevelCut | 删除一个会员等级促销策略 |
| PromotionWebDataService.deleteCircleCut | 删除一个会员特定商圈促销策略 |
| PromotionWebDataService.deleteWebCustomCut | 删除一个特定期间折扣促销策略 |
| PromotionWebDataService.update | 更新一个促销策略 |
| PromotionWebDataService.findByType | 根据促销策略种类找到一个促销策略 |

4.业务逻辑层的动态模型

在促销策略业务中，分为酒店促销策略修改，网站促销策略修改，订单价值计算功能。图4.2.6.3表示了酒店促销会员生日策略修改的相关对象之间的协作。



下图所示的状态图描述了PromotionHotel对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。



5.业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的控制器委托给不同的领域对象。

## 数据层的分解

数据层模块的分解详见需求规格说明文档中的相关描述

### OrderDataService模块

1、模块概述

OrderDataService模块负责订单的相关信息，供读取、保存、修改。

模块内部接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务(供接口) | | |
| OrderDataService .insert | 语法 | public void insert(OrderPO po) throws RemoteException |
| 前置条件 | 相同的po在数据文件中不存在 |
| 后置条件 | 新增一个po |
| OrderDataService .delete | 语法 | public void delete(OrderPO po) throws RemoteException |
| 前置条件 | 相同的po在数据文件中已存在 |
| 后置条件 | 删除一个po |
| OrderDataService.update | 语法 | public void update(OrderPO po) throws RemoteException |
| 前置条件 | 相同的po在数据文件中已存在 |
| 后置条件 | 更新一个po |
| OrderDataService.findByOrderID | 语法 | public OrdePO  findByOrderID(String ID) throws  RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按名字进行查找返回相应的OrderPO结果 |
| OrderDataService.findByUserID | 语法 | public ArrayList<OrderPO>  findByUserID(String ID) throws  RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按条件进行查找返回相应的OrderPO结果 |
| OrderDataService.findByHotelID | 语法 | public ArrayList<OrderPO>  findByHotelID(String ID) throws  RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按条件进行查找返回相应的OrderPO结果 |
| OrderDataService.showList | 语法 | public  ArrayList<OrderPO>  showList() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的客户订单po |

### UserDataService模块

1. 模块概述

UserDataService建立逻辑和数据库之间的联系，获取和更改数据信息。

1. 模块内部接口规范

serdataservice模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| Userdataservice.GetLoginInfo | 语法 | Public String GetLoginInfo(UserManagementPo po); |
| 前置条件 | 数据库中有输入ID的登录信息。 |
| 后置条件 | 返回对应账号的密码。 |
| Userdataservice.AddUser | 语法 | Public String AddUser(UserManagementPo po); |
| 前置条件 | 无。 |
| 后置条件 | 返回新注册用户的userID。 |
| Userdataservice.GetUserBaseInfo | 语法 | Public UserVo GetUserBaseInfo(UserManagementPo po); |
| 前置条件 | 输入的用户ID在数据库中已有注册。 |
| 后置条件 | 返回对应用户的个人信息。 |
| Userdataservice.SetUserBaseInfo | 语法 | Public boolean SetUserBaseInfo(UserManagementPo po); |
| 前置条件 | 输入的用户ID存在且VO中信息完整。 |
| 后置条件 | 修改用户信息数据。 |
| Userdataservice.AddMembers | 语法 | Public boolean AddMembers(CustomerManagementDataServicePo po); |
| 前置条件 | 输入的userID在数据库中存在。 |
| 后置条件 | 将相应用户信息中的是否为会员一栏修改为是。 |
| Userdataservice.GetCustomerOrders | 语法 | Public ArrayList<OrderVo> GetCustomerOrders(CustomerManagementDataServicePo po); |
| 前置条件 | 输入的userID存在。 |
| 后置条件 | 返回相应用户ID对应的订单列表。 |
| Userdataservice.UpDateCustomerOrders | 语法 | Public boolean UpdateCustomerOrders(CustomerManagementDataServicePo po); |
| 前置条件 | 输入的userID存在且OrderID对应的订单状态为可修改。 |
| 后置条件 | 修改相应订单状态。 |
| Userdataservice.GetCustomerrHotel | 语法 | Public ArrayList<HotelVo> GetCustomerHotel(CustomerManagementDataServicePo po); |
| 前置条件 | 输入的userID存在。 |
| 后置条件 | 返回对应用户的酒店信息数组。 |
| Userdataservice.SetCustomerHotel | 语法 | Public boolean SetCustomerHotel (CustomerManagementDataServicePo po); |
| 前置条件 | 输入的userID和hotelID存在。 |
| 后置条件 | 系统更新相应用户的酒店信息。 |
| Userdataservice.SetCustomerCredit | 语法 | Public boolean SetCustomerCredit(CustomerManagementDataServicePo po); |
| 前置条件 | 输入的userID存在且输入的信用度在合理范围内。 |
| 后置条件 | 更新相应用户的信用度信息和信用记录。 |

### HotelDataService模块

（1）模块概述

HotelDataService模块负责保存酒店的信息，进行增删改查的操作。

（2）模块内部接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| HotelInfoDataService.Insert | 语法 | public void insert(HotelInfoPO po) throws RemoteException |
| 前置条件 | 相同的po在数据文件中不存在 |
| 后置条件 | 新增一个po |
| HotelInfoDataService.Delete | 语法 | public void delete(HotelInfoPO po) throws RemoteException |
| 前置条件 | 相同的po在数据文件中已存在 |
| 后置条件 | 删除一个po |
| HotelInfoDataService.Update | 语法 | public void update(HotelInfoPO po) throws RemoteException |
| 前置条件 | 相同的po在数据文件中已存在 |
| 后置条件 | 更新一个po |
| HotelInfoDataService.FindByName | 语法 | public HotelInfoPO  findByName(String name) throws  RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按名字进行查找返回相应的HotelInfoPO结果 |
| HotelInfoDataService.FindByCondition | 语法 | public HotelInfoPO  findByCondition(String condition) throws  RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按条件进行查找返回相应的HotelInfoPO结果 |
| HotelInfoDataService.Show | 语法 | public  ArrayList<HotelInfoPO>  show() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存po |
| HotelEvaluateDataService.Insert | 语法 | public void insert(HotelEvaluatePO po) throws RemoteException |
| 前置条件 | 相同的po在数据文件中不存在 |
| 后置条件 | 新增一个po |
| HotelEvaluateDataService.Delete | 语法 | public void delete(HotelEvaluatePO po) throws RemoteException |
| 前置条件 | 相同的po在数据文件中已存在 |
| 后置条件 | 删除一个po |
| HotelEvaluateDataService.Find | 语法 | public HotelEvaluatePO  findByName(String searchInfo) throws  RemoteException |
| 前置条件 | 无 |
| 后置条件 | 进行查找返回相应的HotelEvaluatePO结果 |
| HotelEvaluateDataService.Show | 语法 | public  ArrayList<HotelEvaluatePO>  show() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存po |
| HotelEvaluateDataService.Insert | 语法 | public void insert(RoomInfoPO po) throws RemoteException |
| 前置条件 | 相同的po在数据文件中不存在 |
| 后置条件 | 新增一个po |
| HotelEvaluateDataService.Delete | 语法 | public void delete(RoomInfoPO po) throws RemoteException |
| 前置条件 | 相同的po在数据文件中已存在 |
| 后置条件 | 删除一个po |
| HotelEvaluateDataService.Update | 语法 | public void update(RoomInfoPO po) throws RemoteException |
| 前置条件 | 相同的po在数据文件中已存在 |
| 后置条件 | 更新一个po |
| HotelEvaluateDataService.FindByName | 语法 | public RoomInfoPO  findByName(String name) throws  RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按名字进行查找返回相应的RoomInfoPO结果 |
| HotelEvaluateDataService.FindByCondition | 语法 | public RoomInfoPO  findByCondition(String condition) throws  RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按条件进行查找返回相应的RoomInfoPO结果 |
| HotelEvaluateDataService.Show | 语法 | public  ArrayList<RoomInfoPO>  show() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存po |

### VipDataService模块

（1）模块概述

VipDataService模块负责保存会员的相关信息，供读取、保存、修改。

（2）模块内部接口规范

|  |  |  |
| --- | --- | --- |
| 需要的服务（需接口） | | |
| VipDataService.insertC | 语法 | public void insertC(CommonVipPO vip) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中不存在 |
| 后置条件 | 在数据库中增加一个 po 记录 |
| VipDataService.insertL | 语法 | public void insertL(LevelSystemPO level) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中不存在 |
| 后置条件 | 在数据库中增加一个 po 记录 |
| VipDataService.insertB | 语法 | public void insertB(BusinessVipPO vip) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中不存在 |
| 后置条件 | 在数据库中增加一个 po 记录 |
| VipDataService.delete | 语法 | public void delete(vipPO vip) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中删除一个 po 记录 |
| VipDataService.update | 语法 | public void update(LevelSystemPO level) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中更新一个 po 记录 |
| VipDataService.updateB | 语法 | public void updateB(BusinessVipPO vip) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中更新一个 po 记录 |
| VipDataService.updateC | 语法 | public void updateC(CommonVipPO vip) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中更新一个 po 记录 |
| VipDataService.ShowB | 语法 | public ArrayList<BusinessVipPO> showB() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存 PO |
| VipDataService.getLevelSystemPO | 语法 | public ArrayList<LevelSystemPO> getLevelSystemPO throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按等级返回相应的结果 |
| VipDataService.findL | 语法 | public LevelSystemPO findL(int level) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按等级返回相应的PO结果 |
| VipDataService.findByUserIDC | 语法 | public CommonVipPO findByUserIDC(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按等级返回相应的vipPO结果 |
| VipDataService.findByUserIDB | 语法 | public BusinessVipPO findByUserIDB(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按等级返回相应的vipPO结果 |
| VipDataService.ShowC | 语法 | public ArrayList<CommonVipPO> showB() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存 PO |

### CreditDataService模块

（1）模块概述

CreditDataService模块负责保存信用值的相关信息，供读取、保存、修改。

（2）模块内部接口规范

|  |  |  |
| --- | --- | --- |
| 需要的服务（需接口） | | |
| CreditDataService.insert | 语法 | public void insert(creditPO credit) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中不存在 |
| 后置条件 | 在数据库中增加一个 po 记录 |
| CreditDataService.getListByUserID | 语法 | public ArrayList<creditPO> getListByUserID() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存 PO |
| CreditDataService.getListByOrderID | 语法 | public ArrayList<creditPO> getListByOrderID() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存 PO |
| CreditDataService.Show | 语法 | public ArrayList<creditPO> show() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存 PO |

### PromotionDataService模块

（1）模块概述

PromotionDataService模块负责保存促销策略的相关信息，供读取、保存、修改。

（2）模块内部接口规范

|  |  |  |
| --- | --- | --- |
| 需要的服务（需接口） | | |
| PromotionHotelDataService.insert | 语法 | public void insert(PromotionWebPO promotion) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中不存在 |
| 后置条件 | 在数据库中增加一个 po 记录 |
| PromotionHotelDataService.deleteBirthCut | 语法 | public void deleteBirthCut(int level,String hotelID) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中删除一个 po 记录 |
| PromotionHotelDataService.deleteHotelCustomCut | 语法 | public void deleteHotelCustomCut(String begin,String over,String hotelID) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中删除一个 po 记录 |
| PromotionHotelDataService.deleteJoin | 语法 | public void deleteJoin(String name,String hotelID) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中删除一个 po 记录 |
| PromotionHotelDataService.deleteOverCut | 语法 | public void deleteBirthCut(int num,String hotelID) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中删除一个 po 记录 |
| PromotionHotelDataService.update | 语法 | public ResultMsg update(PromotionHotelPO promotion) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中更新一个 po 记录 |
| PromotionHotelDataService.findByType | 语法 | public ArrayList<PromotionHotelPO> findByType throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按等级返回相应的promotionPO结果 |
| PromotionHotelDataService.Show | 语法 | public ArrayList<PromotionHotelPO> show() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存 PO |
| PromotionWebDataService.insert | 语法 | public void insert(PromotionWebPO promotion) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中不存在 |
| 后置条件 | 在数据库中增加一个 po 记录 |
| PromotionWebDataService.deleteLevelCut | 语法 | public void deleteBirthCut(int level) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中删除一个 po 记录 |
| PromotionWebDataService.deleteWebCustomCut | 语法 | public void deleteWebCustomCut(String begin,String over) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中删除一个 po 记录 |
| PromotionWebDataService.deleteCircleCut | 语法 | public void deleteBirthCut(Area area) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中删除一个 po 记录 |
| PromotionWebDataService.update | 语法 | public ResultMsg update(PromotionWebPO promotion) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中更新一个 po 记录 |
| PromotionWebDataService.findByType | 语法 | public ArrayList<PromotionWebPO> findByType throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按等级返回相应的promotionPO结果 |
| PromotionWebDataService.Show | 语法 | public ArrayList<PromotionWebPO> show() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存 PO |

# 依赖视角

下图分别为客户端和服务器端各自的包之间的联系：

