**酒店预订系统**

**详细设计**

**描述文档**

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10月09日

目录

[更新历史 2](#_Toc465100730)

[1 引言 3](#_Toc465100731)

[1.1 编制目的 3](#_Toc465100732)

[1.2 词汇表 3](#_Toc465100733)

[1.3 参考资料 3](#_Toc465100734)

[2 产品概述 3](#_Toc465100735)

[3 体系结构设计概述 4](#_Toc465100736)

[4 结构视角 4](#_Toc465100737)

[4.1 展示层的分解 4](#_Toc465100738)

[4.1.1 UI层设计概要 4](#_Toc465100739)

[4.1.2 UI层内部包的职责与划分 4](#_Toc465100740)

[4.2 业务逻辑层的分解 9](#_Toc465100741)

[4.2.1 Order模块 9](#_Toc465100742)

[4.3 数据层的分解 14](#_Toc465100743)

[4.3.1 DataFactoryService模块 14](#_Toc465100744)

[5 依赖视角 16](#_Toc465100745)

# 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 |

# 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** |
|  |  |

## 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

## 展示层的分解

### UI层设计概要

ui 层主要以人员的性质来分，包括客户、酒店工作人员、网站营销人员、网站管理人员。

ui层的入口是mainframe，显示的是打开系统的login界面，然后通过不同身份的人的账户登录，跳转到各个职位所对应的ui包内，之后职位所对应的各个功能的界面可以最大程度在包内跳转，更好的实现高内聚、低耦合。

### UI层内部包的职责与划分

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 包名 | 类名 | 功能 |
| presentation | Orderui | PersonalOrderScanUI | 初始化客户浏览个人订单的窗口 |
| Orderui | PersonalOrderDetailUI | 初始化客户浏览个人订单详情的窗口 |
| Orderui | PersonalOrderCancelUI | 初始化客户撤销个人订单的窗口 |
| Orderui | HotelOrderScanUI | 初始化酒店工作人员浏览酒店订单的窗口 |
| Orderui | HotelOrderModifyUI | 初始化酒店工作人员修改酒店订单状态的窗口 |
| Orderui | HotelOrderDetailUI | 初始化酒店工作人员浏览酒店订单详情的窗口 |
| Orderui | ComplaintListScanUI | 初始化网站营销人员浏览申诉列表的窗口 |
| Orderui | ComplaintHandleUI | 初始化网站营销人员处理申诉的窗口 |
| Orderui | AbnormalOrderScanUI | 初始化网站营销人员浏览异常订单的窗口 |
| Orderui | AbnormalOrderDetailUI | 初始化网站营销人员浏览异常订单详情的窗口 |

## 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接口。业务逻辑层和数据层之间添加dataservice.orderdataservice.OrderOnUserDataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加OrderOnUserController，这样OrderOnUserController会将对订单的业务逻辑处理委托给OrderOnUser对象。OrderOnUserPO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：

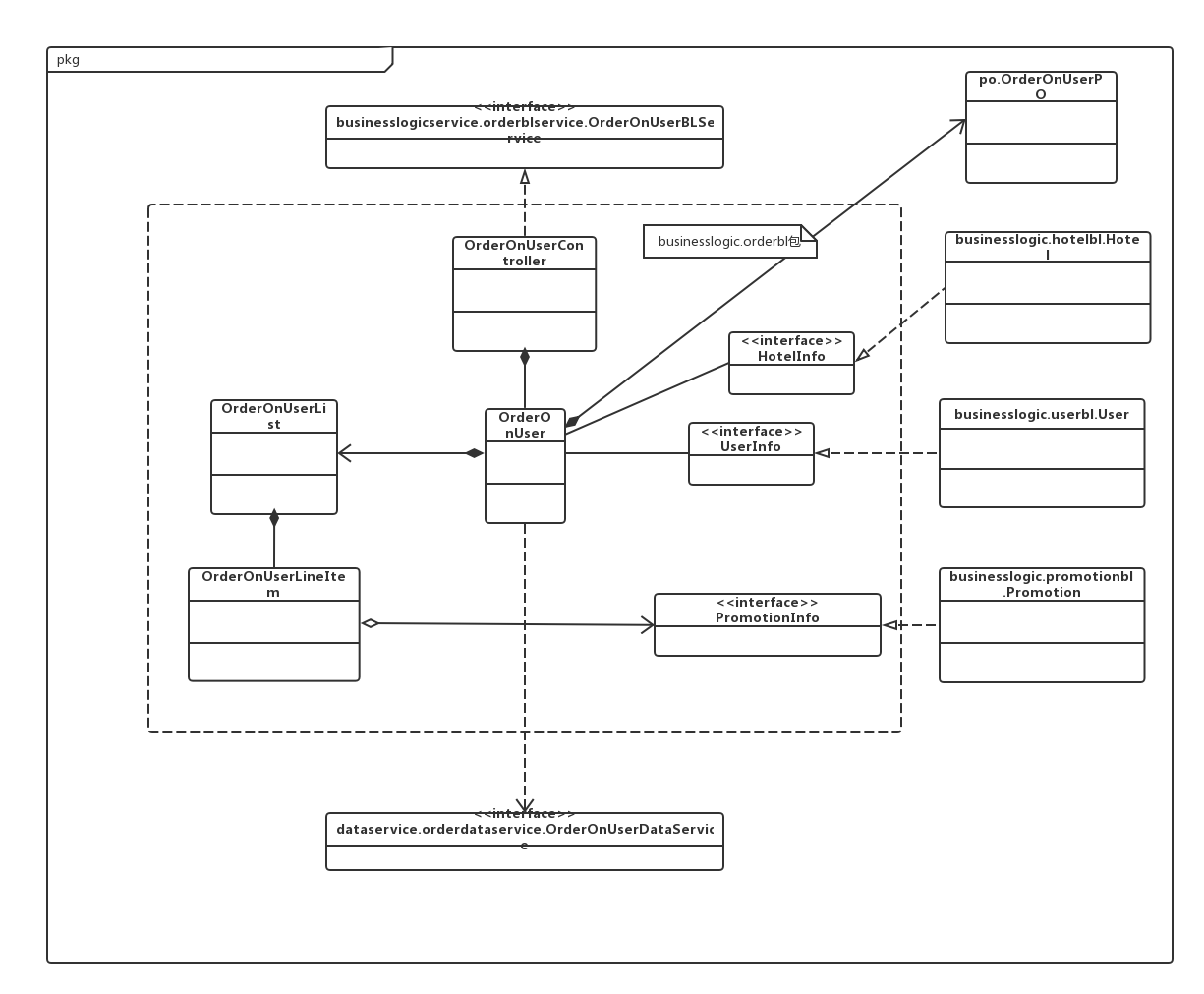


图1-1 Orderbl模块中各个类的设计

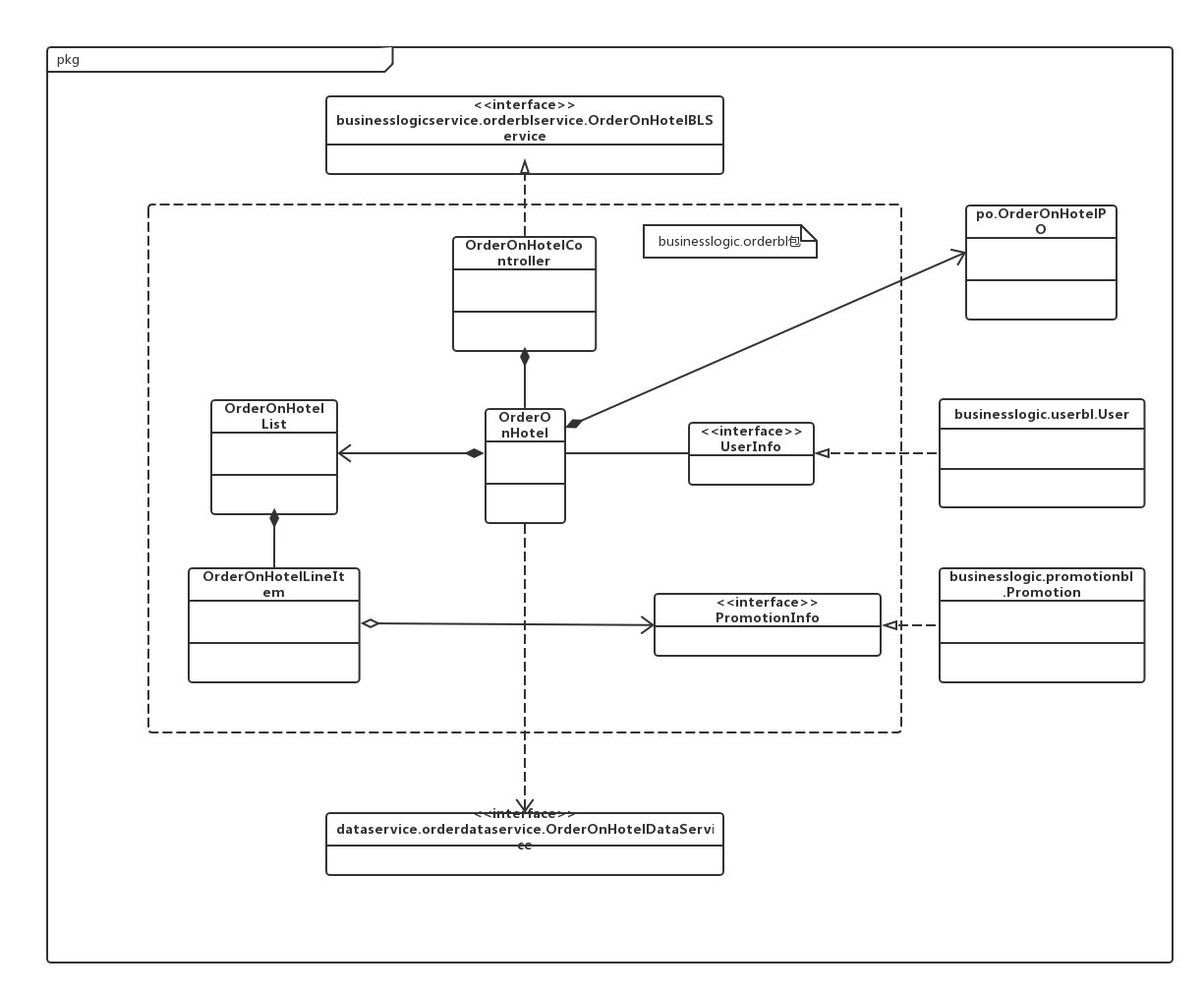


图1-2 Orderbl模块中各个类的设计

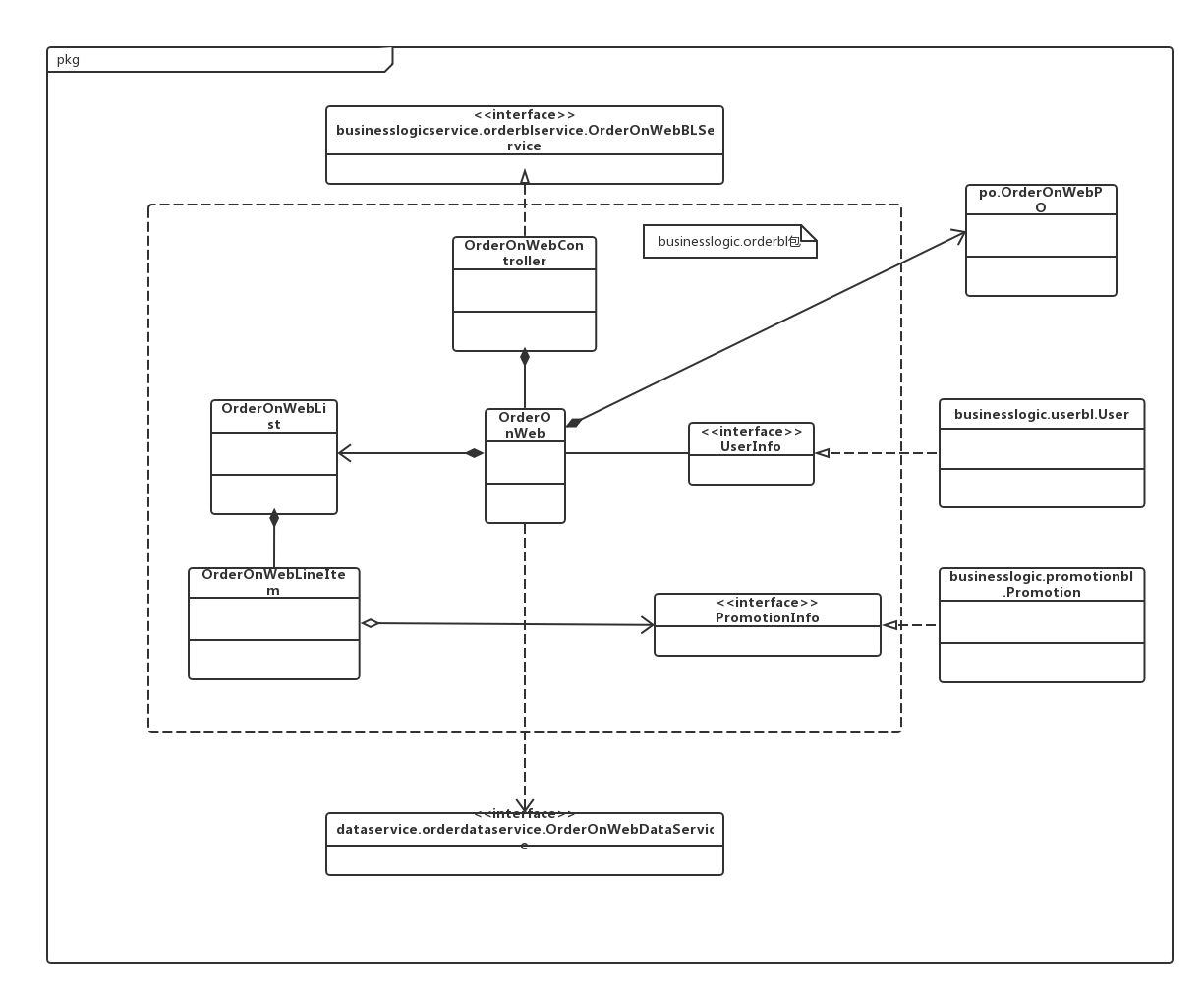


图1-3 Orderbl模块中各个类的设计

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、申诉列表界面、申诉处理界面所需要的服务 |

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<OrderOnUserVO> personalOrderScan(); |
| 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(OrderOnUserVO 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 OrderMsg personalOrderDetail(OrderOnUserVO 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personalOrderDetail方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderOnUser.personalOrderScan() | 查查看个人订单 | |
| OrderOnUser.personalOrderCancel(OrderOnUserVO vo) | 撤销个人订单 | |
| OrderOnUser.personalOrderDetail(OrderOnUserVO vo) | 查看个人订单详情 | |

表2-1-2 OrderOnUser的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderOnUser. personalOrderScan | 语法 | public ArrayList<OrderOnUserVO> personalOrderScan(); |
| 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(OrderOnUserVO 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 ArrayList<OrderOnHotelVO> hotelOrderDetail(OrderOnUserVO orderVO); |
| 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 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | |
| OrderOnUserDataService.  insert(OrderOnUserPO po) | 新增单一持久化对象 | |
| OrderOnUserDataService.  delete(OrderOnUserPO po) | 删除单一持久化对象 | |
| OrderOnUserDataService.  update(OrderOnUserPO po) | 更新单一持久化对象 | |
| OrderOnUserDataService.findByName  (OrderOnUesrPO po) | 根据姓名查找单一持久化对象 | |
| OrderOnUserDataService.findByCondition  (OrderOnUesrPO po) | 根据条件查找单一持久化对象 | |

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<OrderOnHotelVO> hotelOrderScan(); |
| 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(OrderOnHotelVO 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 OrderMsg hotelOrderDetail(OrderOnHotelVO 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hotelOrderDetail方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderOnHotel.hotelOrderScan() | 查看酒店订单 | |
| OrderOnHotel.hotelOrderNodify(OrderOnHotelVO vo) | 修改酒店订单状态 | |
| OrderOnHotel.hotelOrderDetail(OrderOnHotelVO vo) | 查看酒店订单详情 | |

表2-2-2 OrderOnHotel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderOnHotel. hotelOrderScan | 语法 | public ArrayList<OrderOnHotelVO> hotelOrderScan(); |
| 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 personalOrderModify(OrderOnHotelVO 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 ArrayList<OrderOnHotelVO> hotelOrderDetail(OrderOnHotelVO orderVO); |
| 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 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | |
| OrderOnHotelDataService.insert(OrderOnHotelPO po) | 新增单一持久化对象 | |
| OrderOnHotelDataService.delete(OrderOnHotelPO po) | 删除单一持久化对象 | |
| OrderOnHotelDataService.update(OrderOnHotelPO po) | 更新单一持久化对象 | |
| OrderOnHotelDataService.find  (OrderOnHotelPO po) | 查找单一持久化对象 | |
| OrderOnHotelDataService.show(OrderOnHotelPO po) | 显示单一持久化对象 | |

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.complaintListScan | 语法 | public ArrayList<OrderOnWebVO> complaintListScan(); |
| 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complaintListScan方法，持久化更新相关数据 |
| OrderOnWebController.complaintHandle | 语法 | public ResultMsg complaintHandle(OrderOnWebVO orderVO); |
| 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complaintHandle方法，系统持久化更新相关数据 |
| 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(OrderOnWebVO orderVO); |
| 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 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderOnWeb.complaintListScan() | 查看申诉列表 | |
| OrderOnWeb.conplaintHandle(OrderOnWebVO vo) | 处理申诉 | |
| OrderOnWeb.abnormalOrderScan() | 浏览异常订单 | |
| OrderOnWeb.abnormalOrderDetail(OrderOnWebVO vo) | 查看异常订单详情 | |

表2-3-2 OrderOnWeb的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderOnWeb. complaintListScan | 语法 | public ArrayList<OrderOnWebVO> complaintListScan(); |
| 前置条件 | 订单信息符合规则且齐全 |
| 后置条件 | 系统显示申诉列表 |
| OrderOnWeb. complaintHandle | 语法 | public ResultMsg complaintHandle(OrderOnWebVO orderVO); |
| 前置条件 | 订单信息符合规则且齐全 |
| 后置条件 | 系统处理申诉成功 |
| 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 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | |
| OrderOnWebDataService.  insert(OrderOnWebPO po) | 新增单一持久化对象 | |
| OrderOnWebDataService.  delete(OrderOnWebPO po) | 删除单一持久化对象 | |
| OrderOnWebDataService.  update(OrderOnWebPO po) | 更新单一持久化对象 | |
| OrderOnWebDataService.findByName  (OrderOnWebPO po) | 根据姓名查找单一持久化对象 | |
| OrderOnWebDataService.findByCondition(OrderOnUesrPO po) | 根据条件查找单一持久化对象 | |
| OrderOnWebDataService.show(OrderOnWebPO po) | 显示单一持久化对象 | |

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。

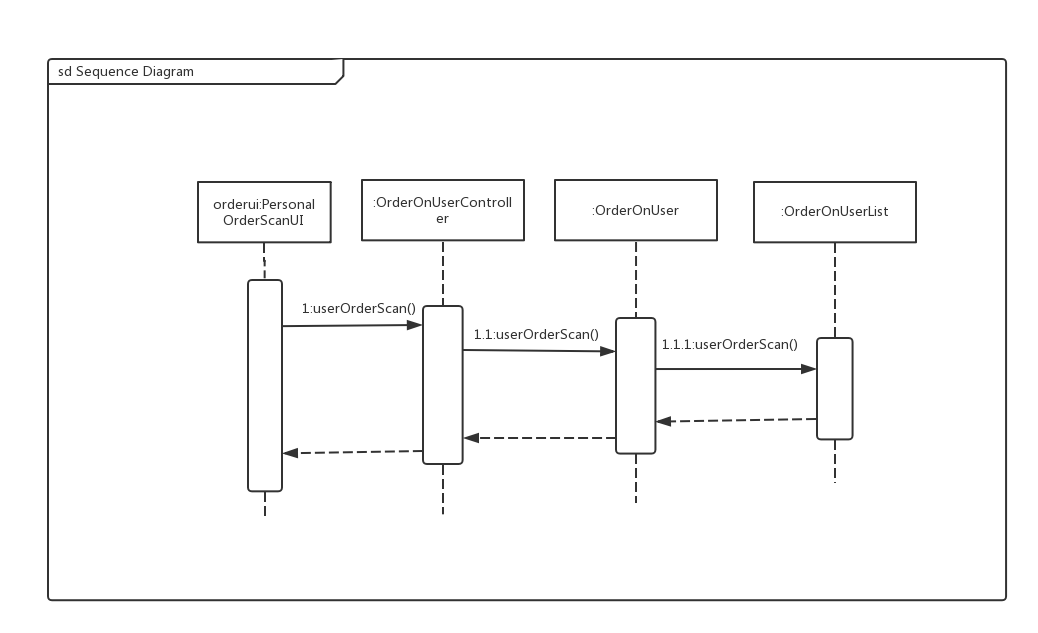


图2-1-1 查看个人订单的顺序图

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。

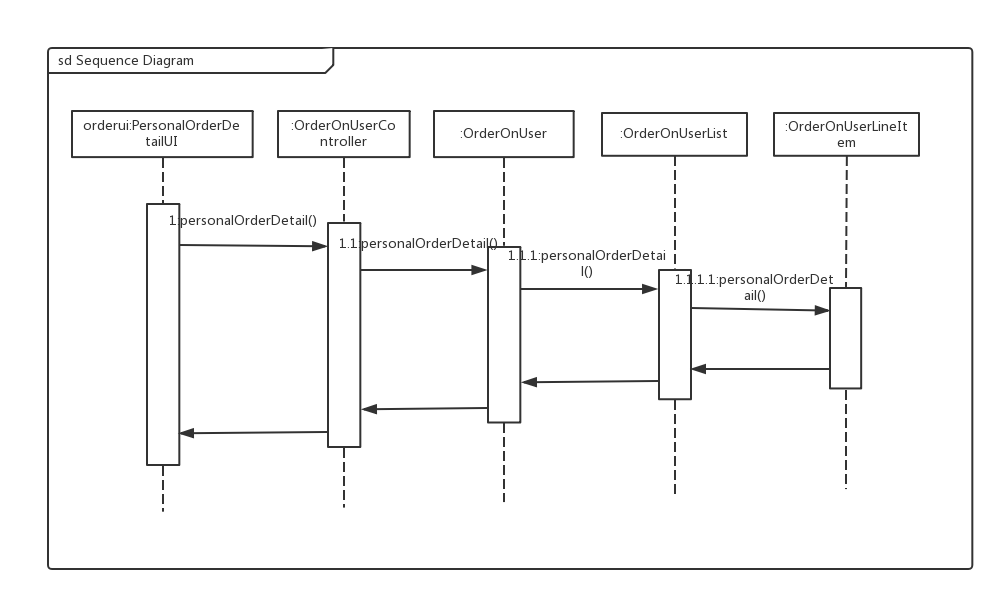


图2-1-2 查看个人订单详情的顺序图

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。

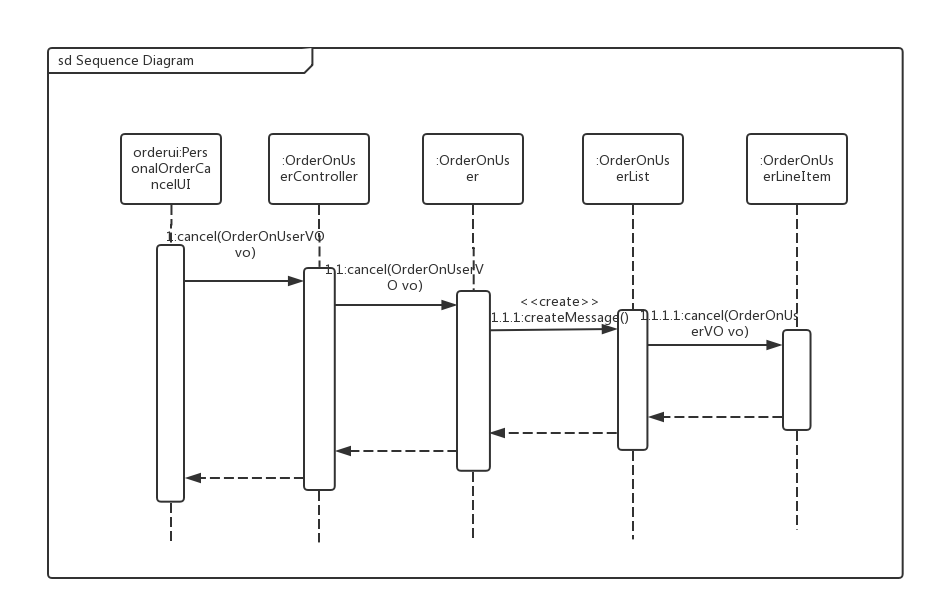


图2-1-3 撤销订单的顺序图

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。

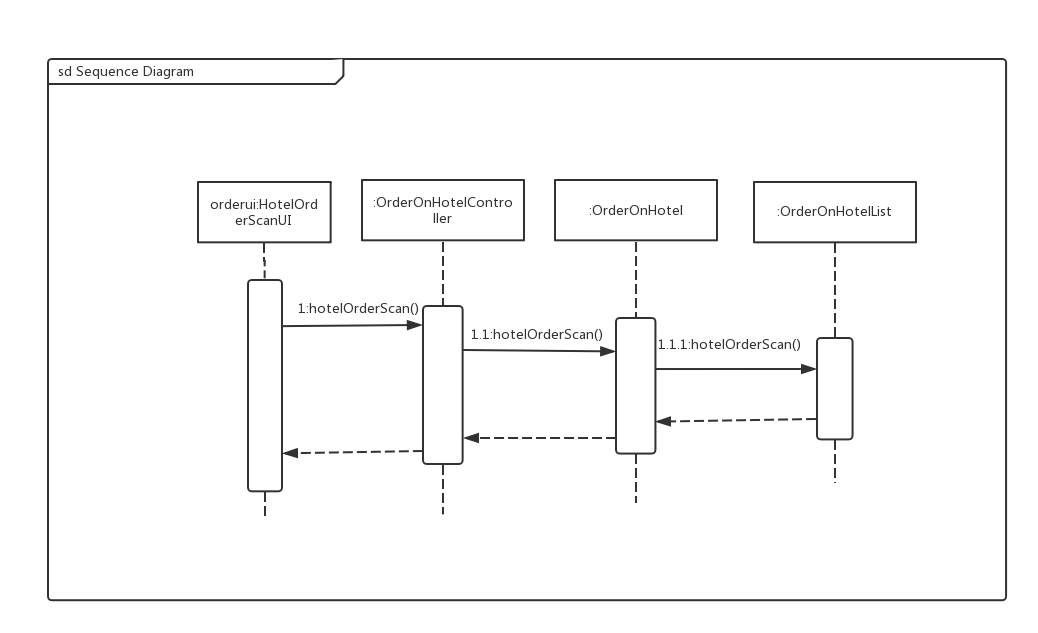


图2-2-1 查看酒店订单信息的顺序图

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。

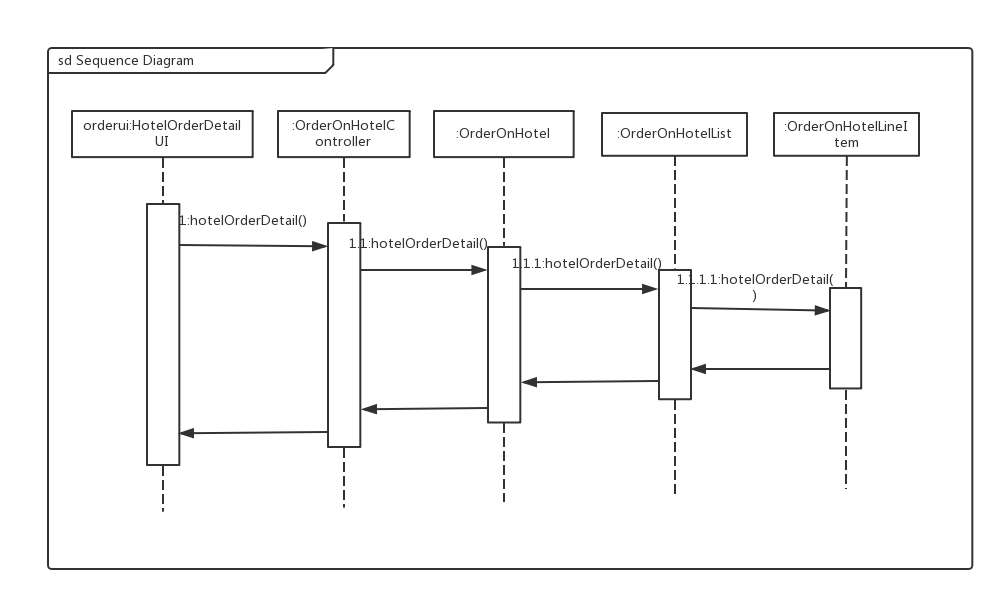


图2-2-2 查看酒店订单详情的顺序图

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。

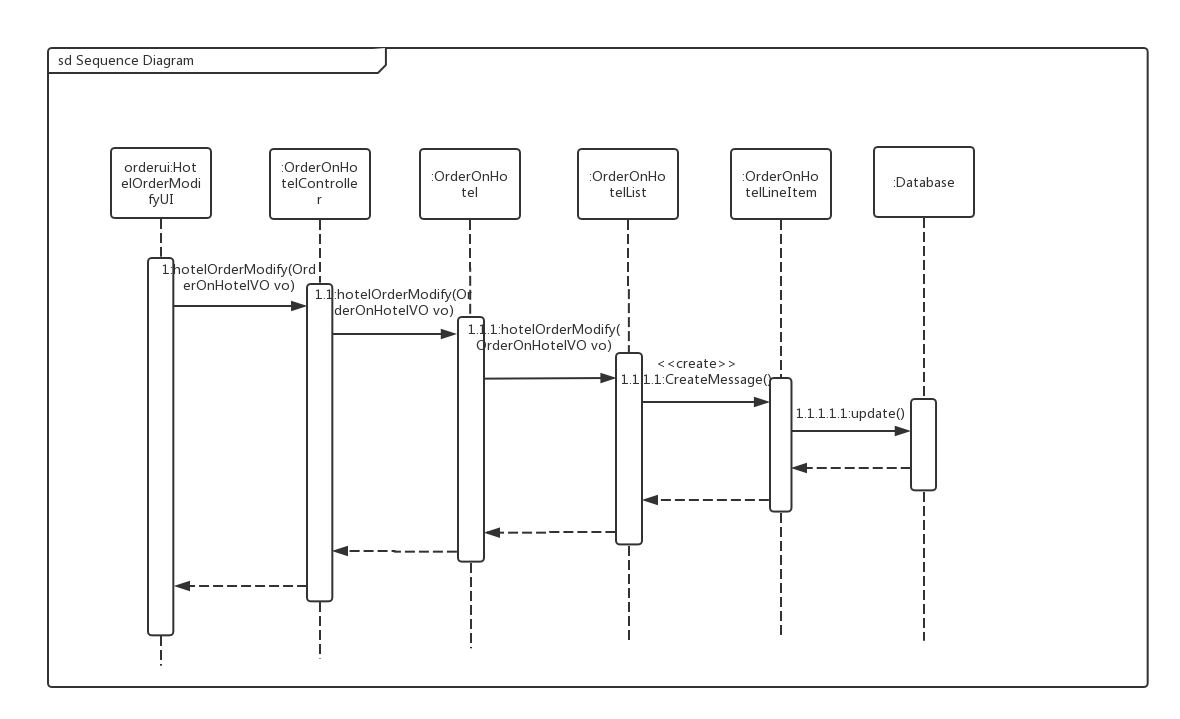


图2-2-3 修改订单状态的顺序图

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。

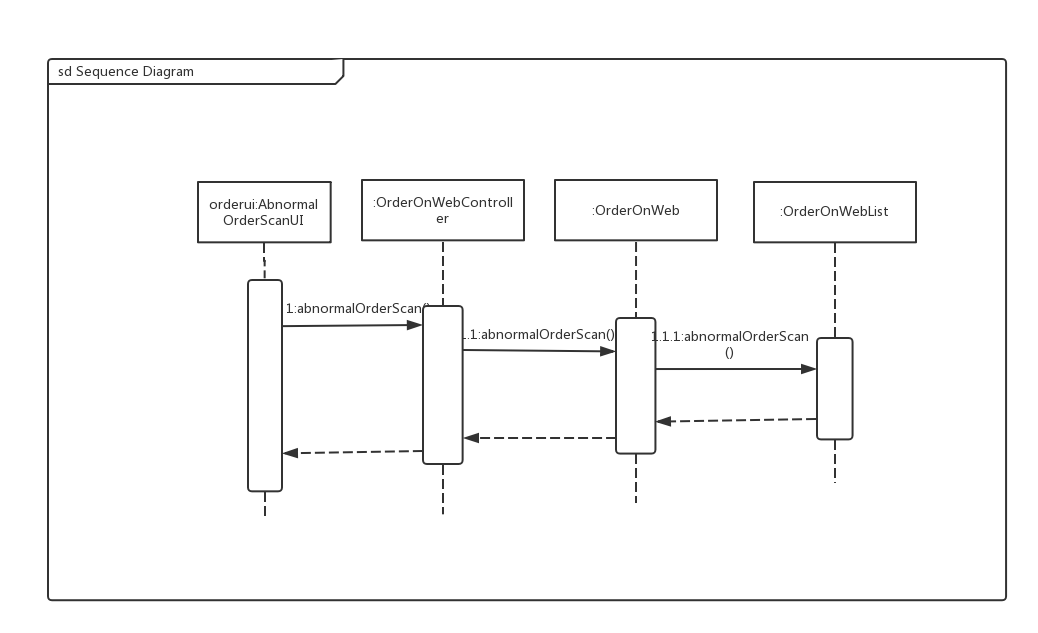


图2-3-1 查看异常订单的顺序图

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。

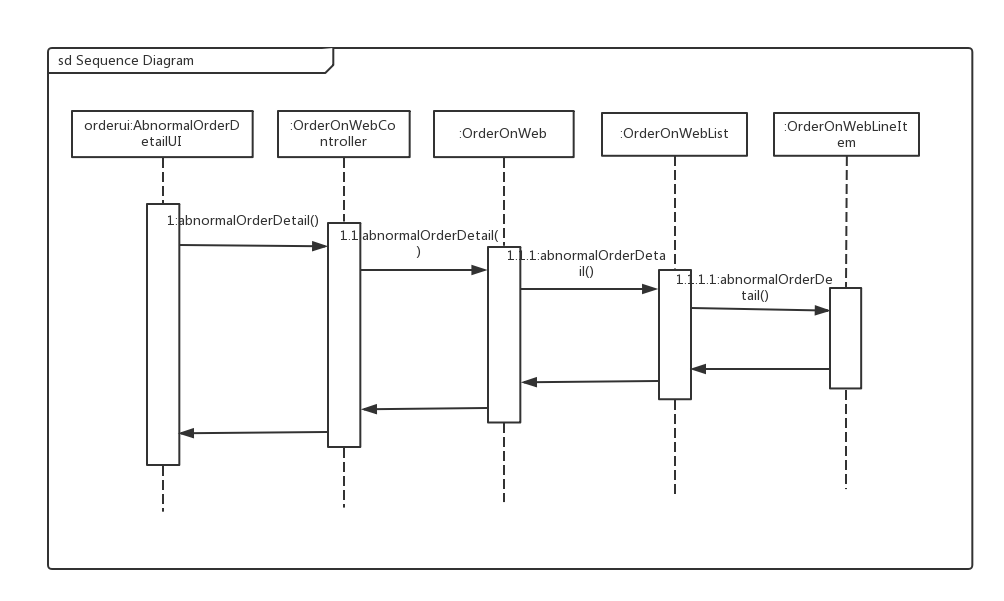


图2-3-2 查看异常订单详情的顺序图

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。

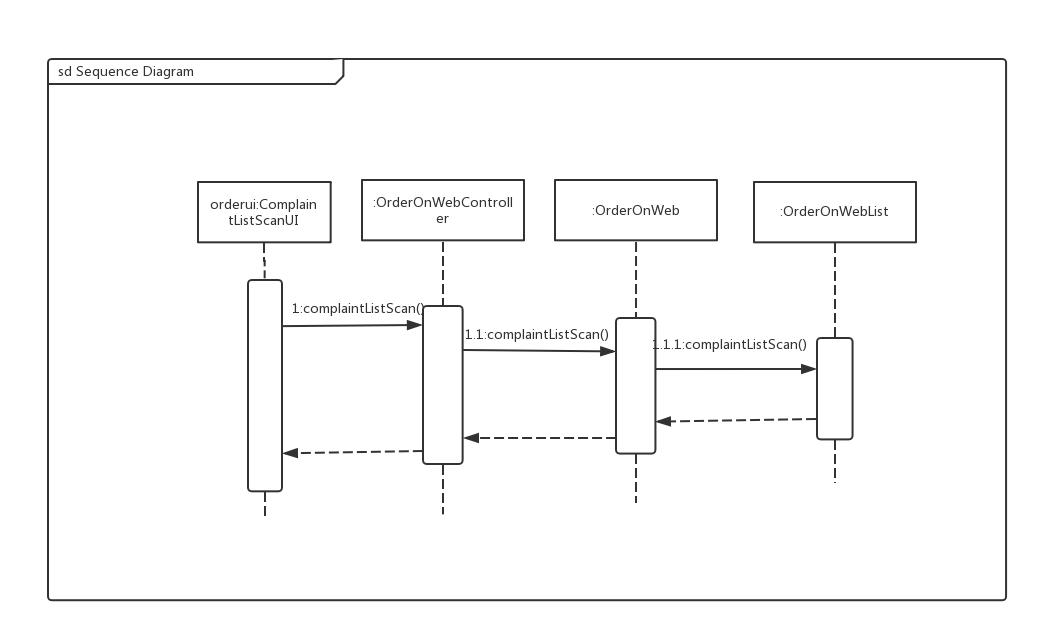


图2-3-3 查看申诉列表的顺序图

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。

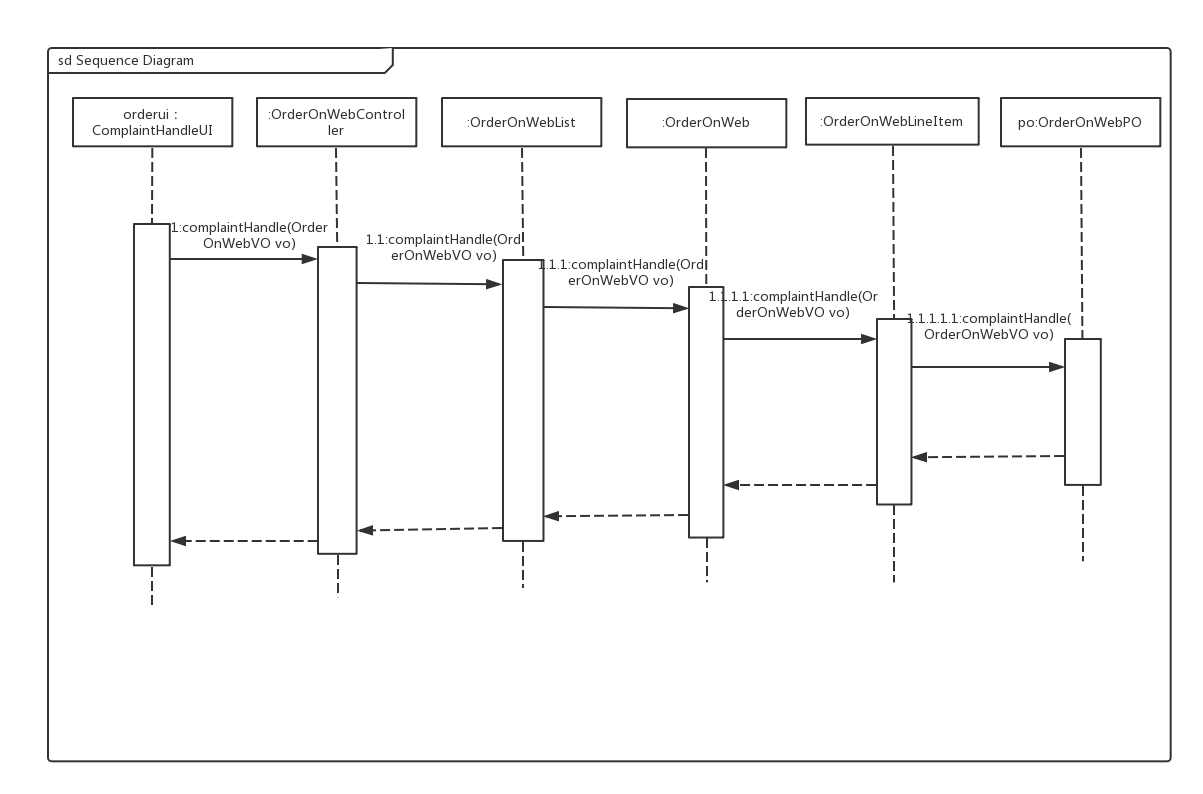


图2-3-4 处理申诉的顺序图

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。

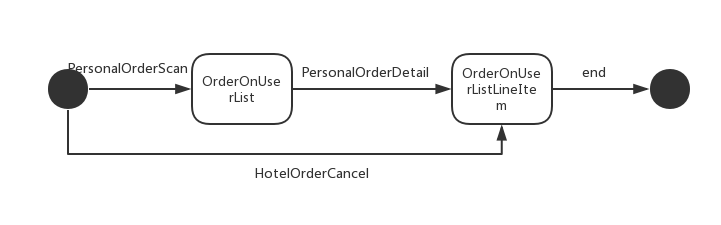


图3-1 OrderOnUser状态图

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。

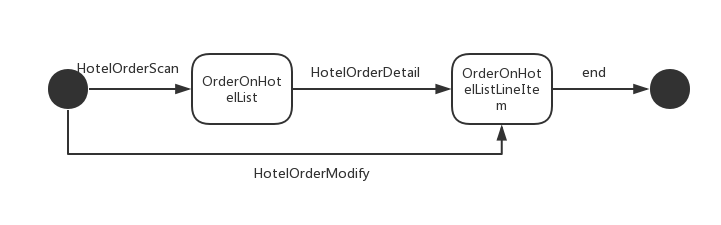


图3-1 OrderOnHotel状态图

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。

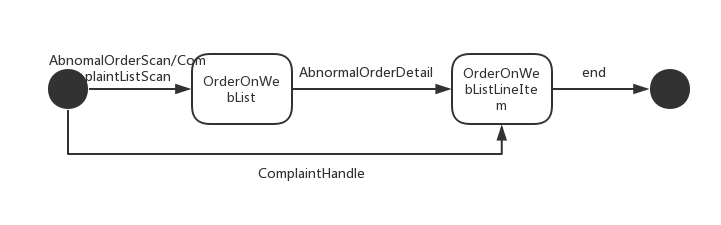


图3-1 OrderOnWeb状态图

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。

## 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) | | |
| OrderOnUserDataService .insert | 语法 | public void insert(OrderOnUserPO 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 |
| OrderOnUserDataService .delete | 语法 | public void delete(OrderOnUserPO 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 |
| OrderOnUserDataService.update | 语法 | public void update(OrderOnUserPO 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 |
| OrderOnUserDataService.findByName | 语法 | public OrderOnUserPO  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OrderOnUserPO结果 |
| OrderOnUserDataService.findByCondition | 语法 | public ArrayList<OrderOnUserPO>  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OrderOnUserPO结果 |
| OrderOnUserDataService.show | 语法 | public  ArrayList<OrderOnUserPO>  show() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的客户订单po |
| OrderOnHotelDataService.insert | 语法 | public void insert(OrderOnHotelPO 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 |
| OrderOnHotelDataService.delete | 语法 | public void delete(OrderOnHotelPO 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 |
| OrderOnHotelDataService.update | 语法 | public void update(OrderOnHotelPO 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 |
| OrderOnHotelDataService.find | 语法 | public OrderOnHotelPO  findByName(String searchInfo) throws  RemoteException |
| 前置条件 | 无 |
| 后置条件 | 进行查找返回相应的OrderOnHotelPO结果 |
| OrderOnHotelDataService.show | 语法 | public  ArrayList<OrderOnHotelPO>  show() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的酒店订单po |
| OrderOnWebDataService.insert | 语法 | public void insert(OrderOnWebPO 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 |
| OrderOnWebDataService.delete | 语法 | public void delete(OrderOnWebPO 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 |
| OrderOnWebDataService.update | 语法 | public void update(OrderOnWebPO 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 |
| OrderOnWebDataService.findByName | 语法 | public OrderOnWebPO  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OrderOnWebPO结果 |
| OrderOnWebDataService.findByCondition | 语法 | public ArrayList<OrderOnWebPO>  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OrderOnWebPO结果 |
| OrderOnWebDataService.show | 语法 | public  ArrayList<OrderOnWebPO>  show() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的异常订单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：

