

F.A.F酒店预订系统

详细设计文档



2016-10-26

NJU F.A.F 万年杰 田原 曾虎双 陆茹茹

V1.0

[**更新历史 3**](#_Toc7488)

[**1. 引言 4**](#_Toc28025)

[**1.1 编制目的** 4](#_Toc16228)

[**1.2 词汇表 4**](#_Toc19022)

[**1.3 参考资料 4**](#_Toc3649)

[**2. 产品概述 4**](#_Toc8259)

[**3. 体系结构设计概述 4**](#_Toc27871)

[**4. 结构视角 4**](#_Toc27279)

[**4.1 业务逻辑层的分解 4**](#_Toc17572)

[**4.1.1 userbl模块** 4](#_Toc999)

[4.1.1.1 LoginAndSignUp 4](#_Toc18633)

[4.1.1.2 ModifyClientInfo 6](#_Toc28813)

[4.1.1.3 SignVip 7](#_Toc23221)

[4.1.1.4 QueryClientCreditRecord 9](#_Toc27090)

[4.1.1.5 AddCreditValue 10](#_Toc15853)

[4.1.1.6 ManageUserInfo 11](#_Toc16057)

[**4.1.2 orderbl模块 13**](#_Toc179)

[4.1.2.1 BrowseHotelOrder 13](#_Toc31385)

[4.1.2.2 BrowseUserOrder 14](#_Toc26264)

[4.1.2.3 CreateNewOrder 16](#_Toc2550)

[4.1.2.4 GetOrderDone 17](#_Toc19585)

[4.1.2.5 CheckAbnormalOrder 19](#_Toc5856)

[4.1.2.6 WithDrawOrder 20](#_Toc31218)

[**4.1.3 hotelbl模块 22**](#_Toc24991)

[4.1.3.1 QueryHotel 22](#_Toc12702)

[4.1.3.2 SearchHotel 23](#_Toc16250)

[4.1.3.3 CheckOrderedHotel 24](#_Toc9101)

[4.1.3.4 CommentOnHotel 26](#_Toc30924)

[4.1.3.5 ImportNewRoom 27](#_Toc13655)

[4.1.3.6 MaintainHotelBasicInfo 28](#_Toc18642)

[4.1.3.7 ManageHotelInfo 29](#_Toc31039)

[**4.1.4 roombl模块 30**](#_Toc5329)

[4.1.4.1 BrowseSpareRoom 30](#_Toc13115)

[4.1.4.2 UpdateCheckIn 32](#_Toc23945)

[4.1.4.3 UpdateCheckOut 34](#_Toc813)

[**4.1.5 strategybl模块 36**](#_Toc17672)

[4.1.5.1 UpdateStrategy 36](#_Toc17872)

[**5. 依赖视角 39**](#_Toc5216)

**更新历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **更新人员** | **日期** | **变更原因** | **版本号** |
| **陆茹茹** | **2016.10.29** | **草稿** | **V0.0** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **引言**
   1. **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。

* 1. **词汇表**

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
|  |  |  |

* 1. **参考资料**

参考酒店预订系统用例文档和体系结构文档。

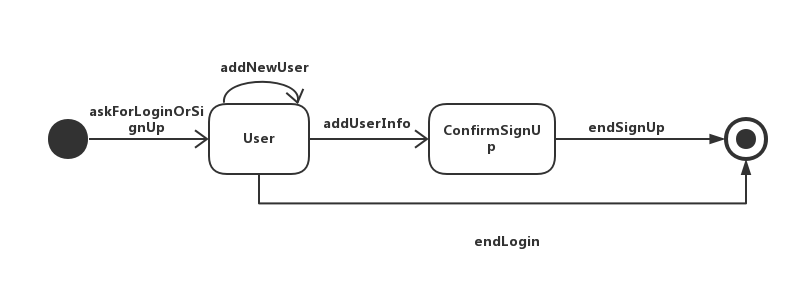
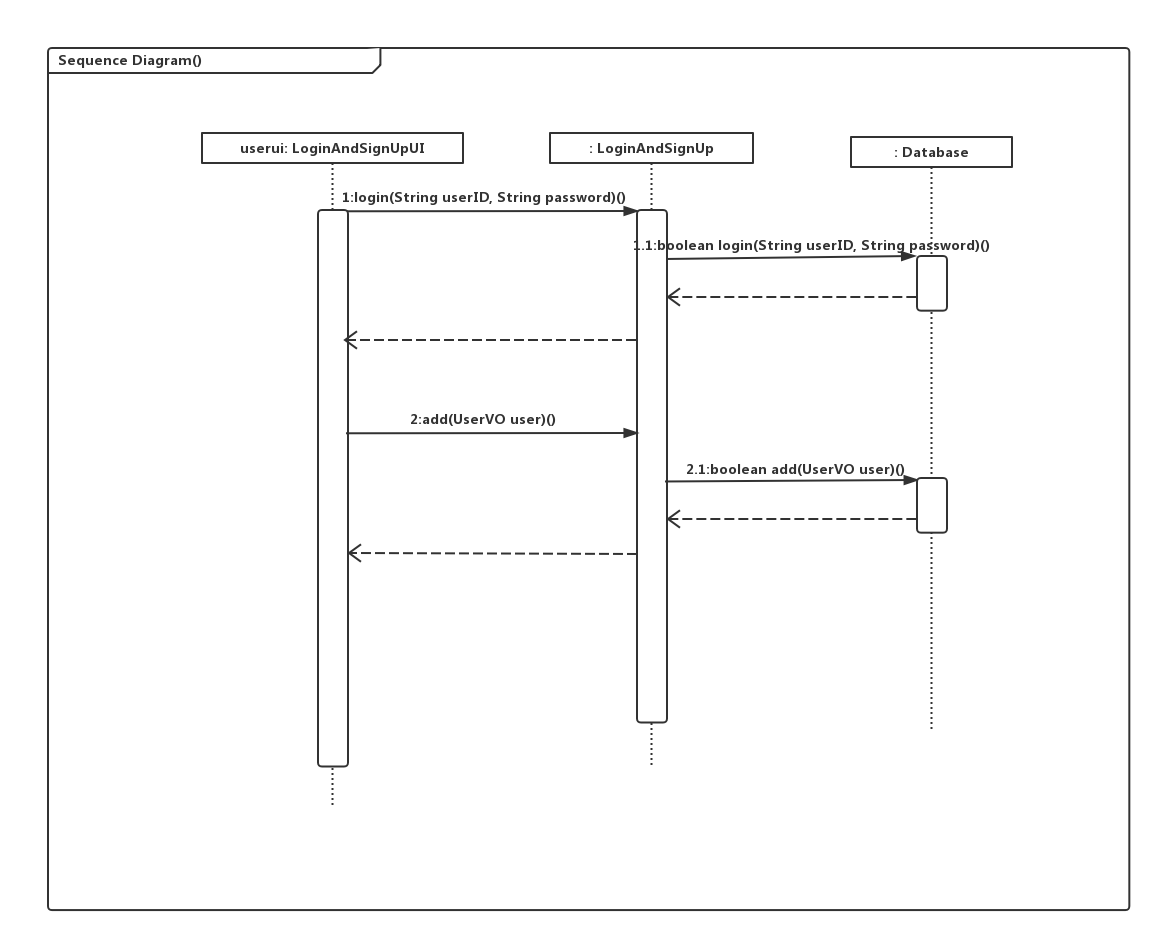
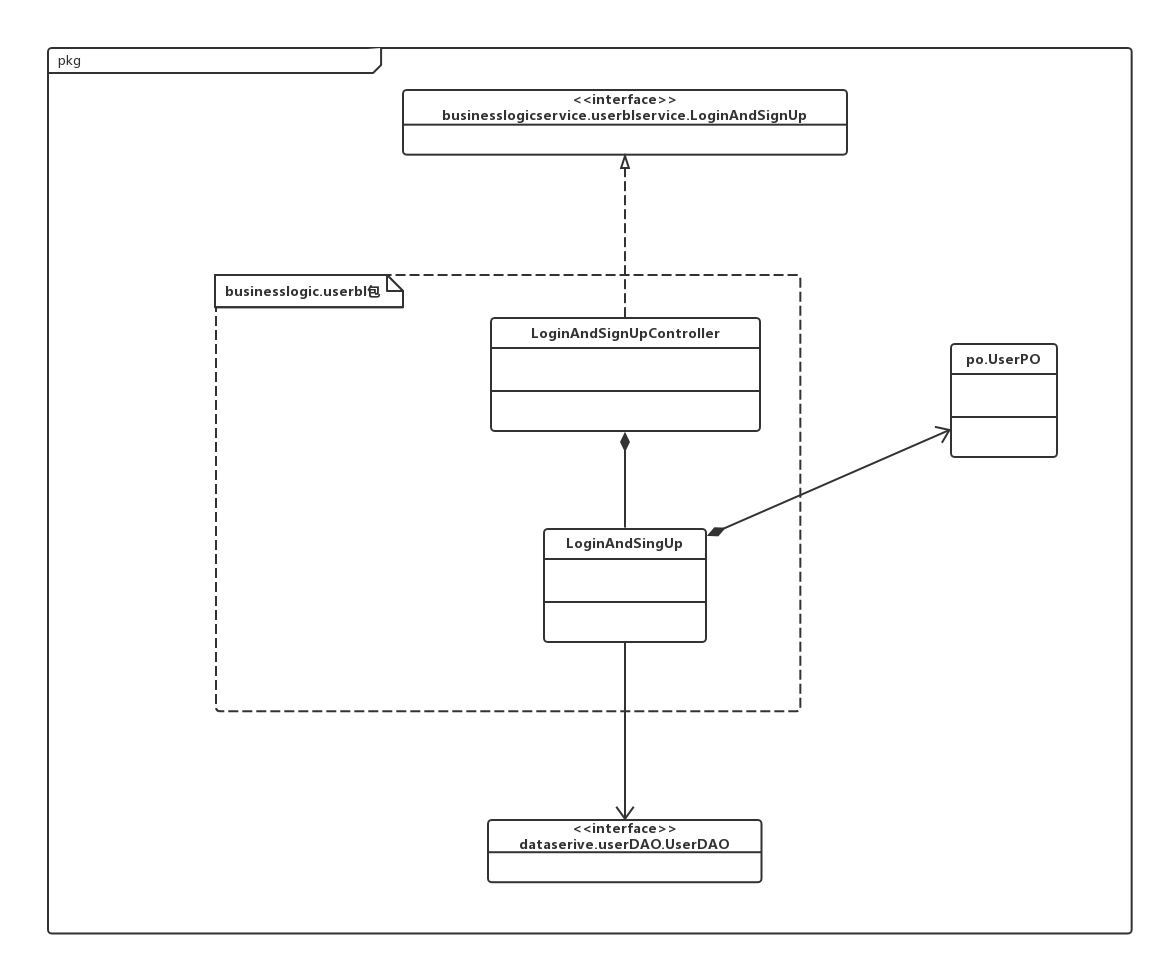
1. **产品概述**

参考酒店预订系统用例文档和软件需求规格说明文档中对产品的概括描述。

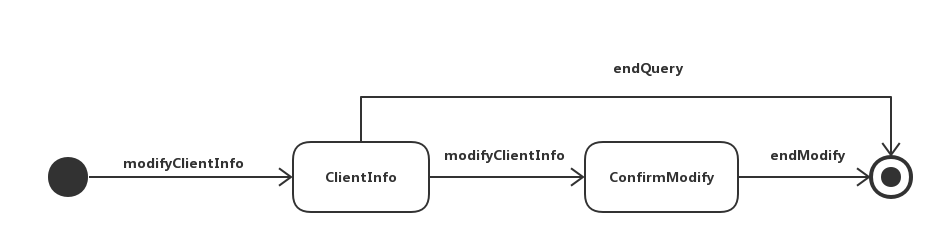
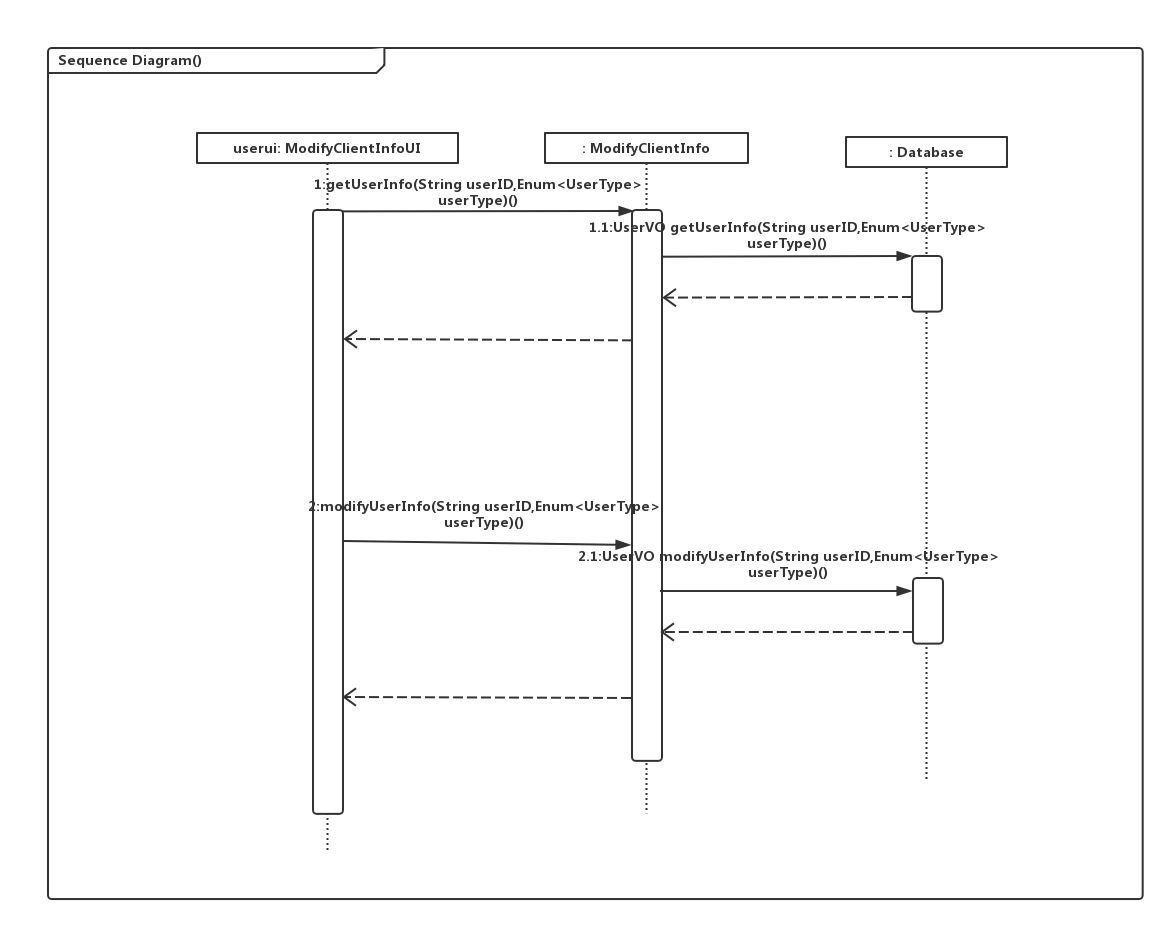
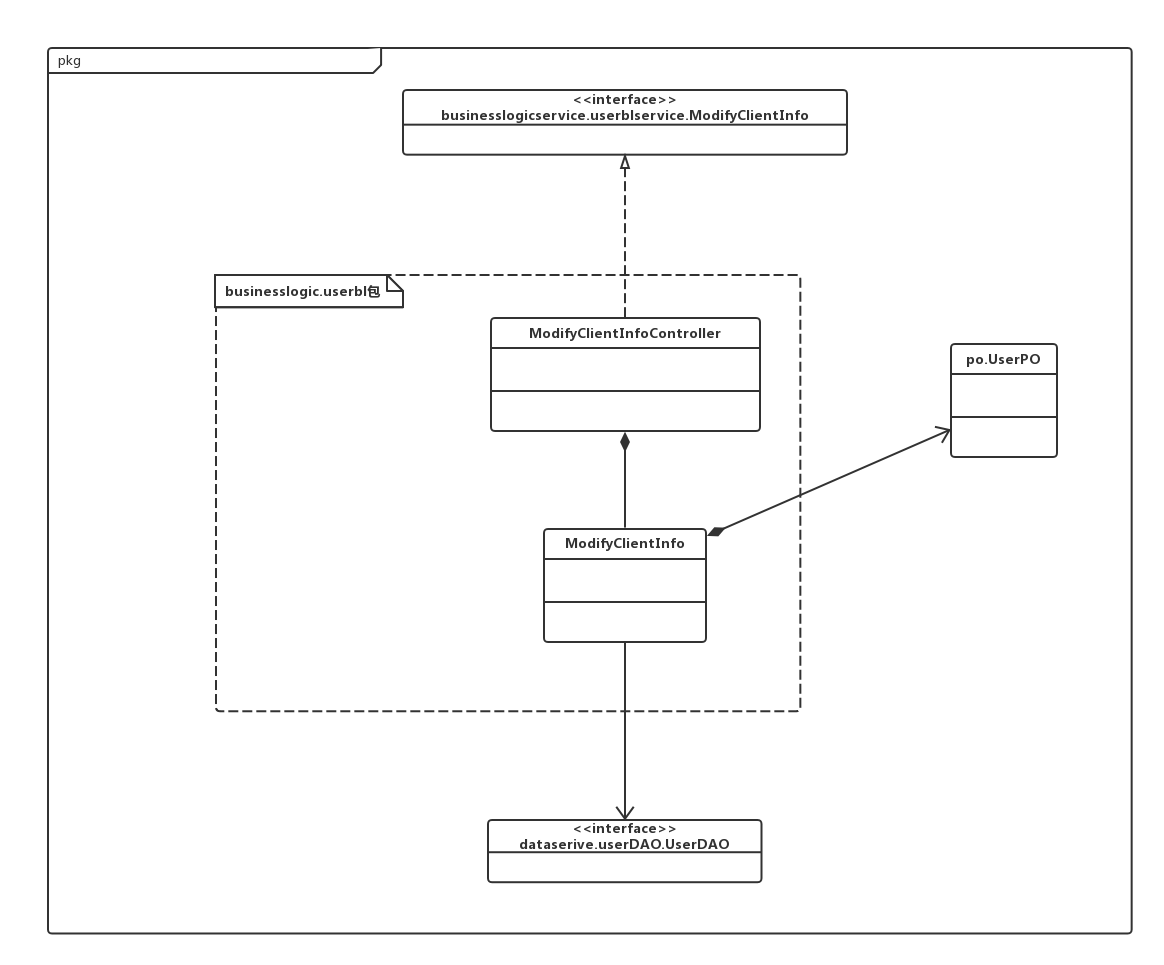
1. **体系结构设计概述**

参考酒店预订系统体系结构文档中对体系结构设计的概述。

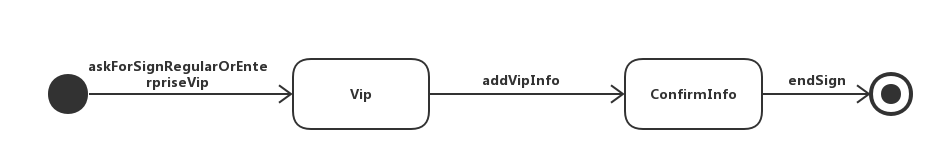
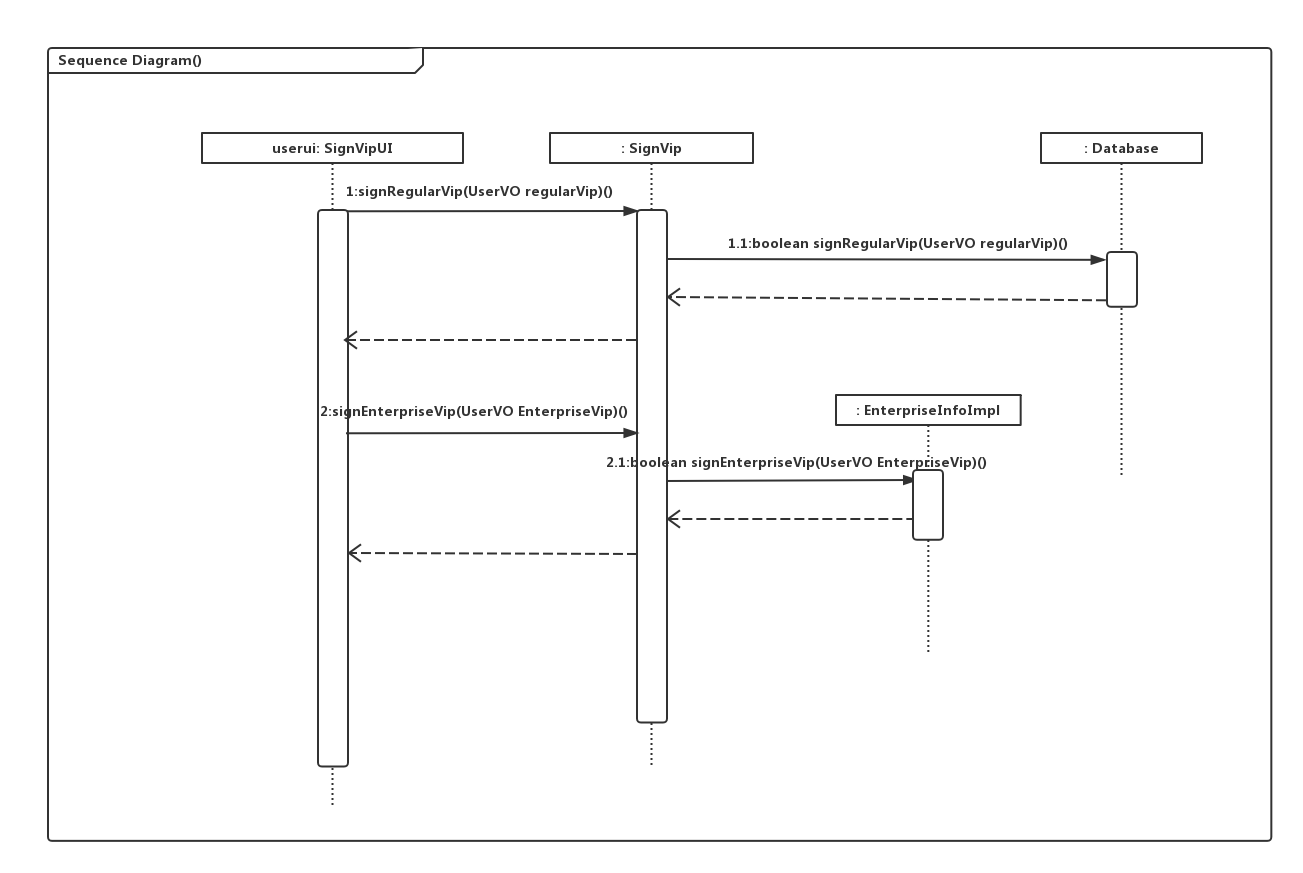
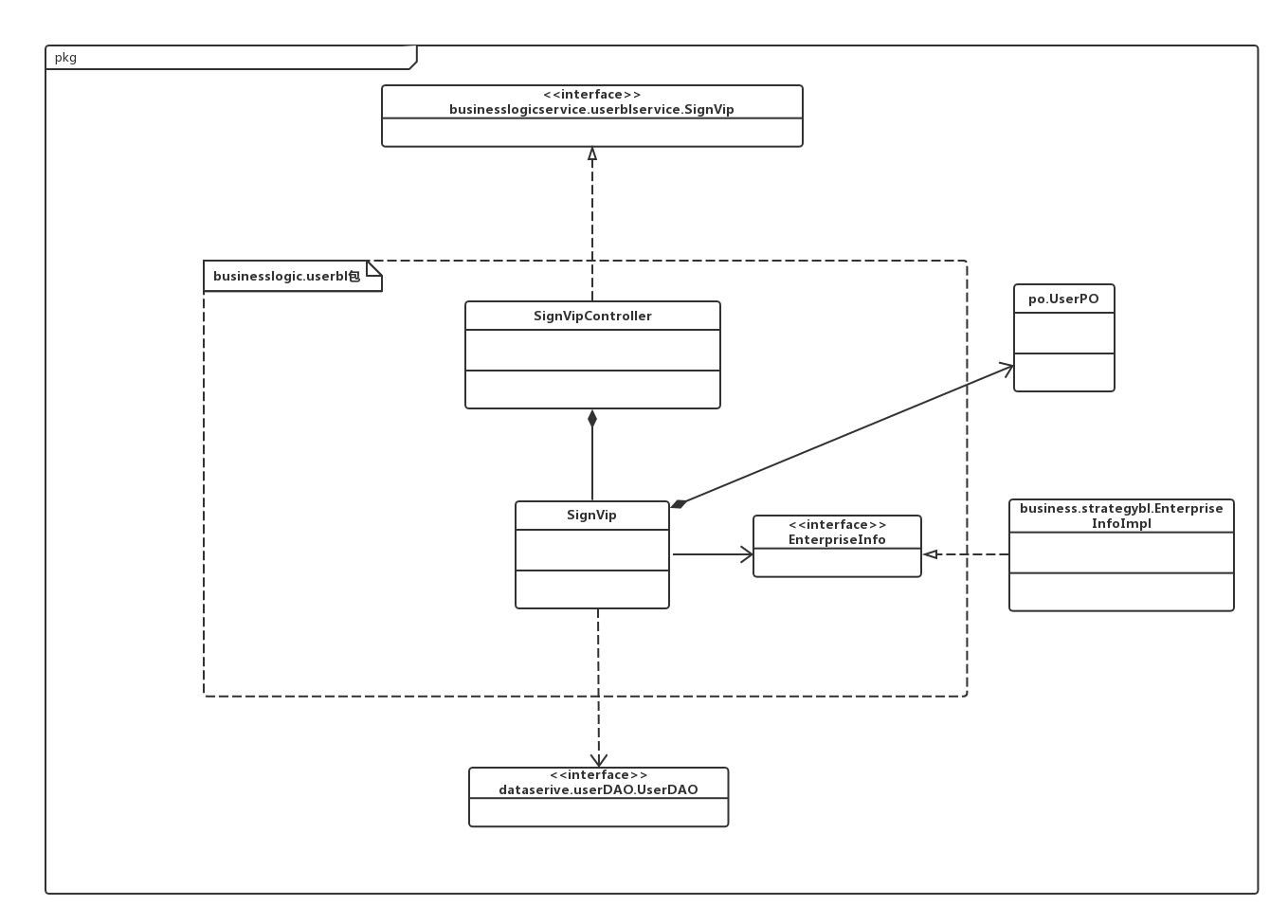
1. **结构视角**
   1. **业务逻辑层的分解**
      1. **userbl模块**
         1. **LoginAndSignUp**

****

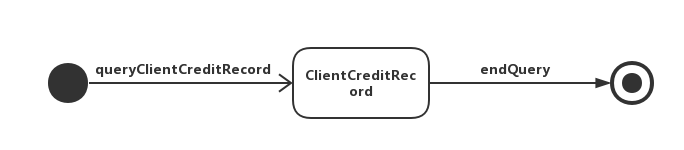
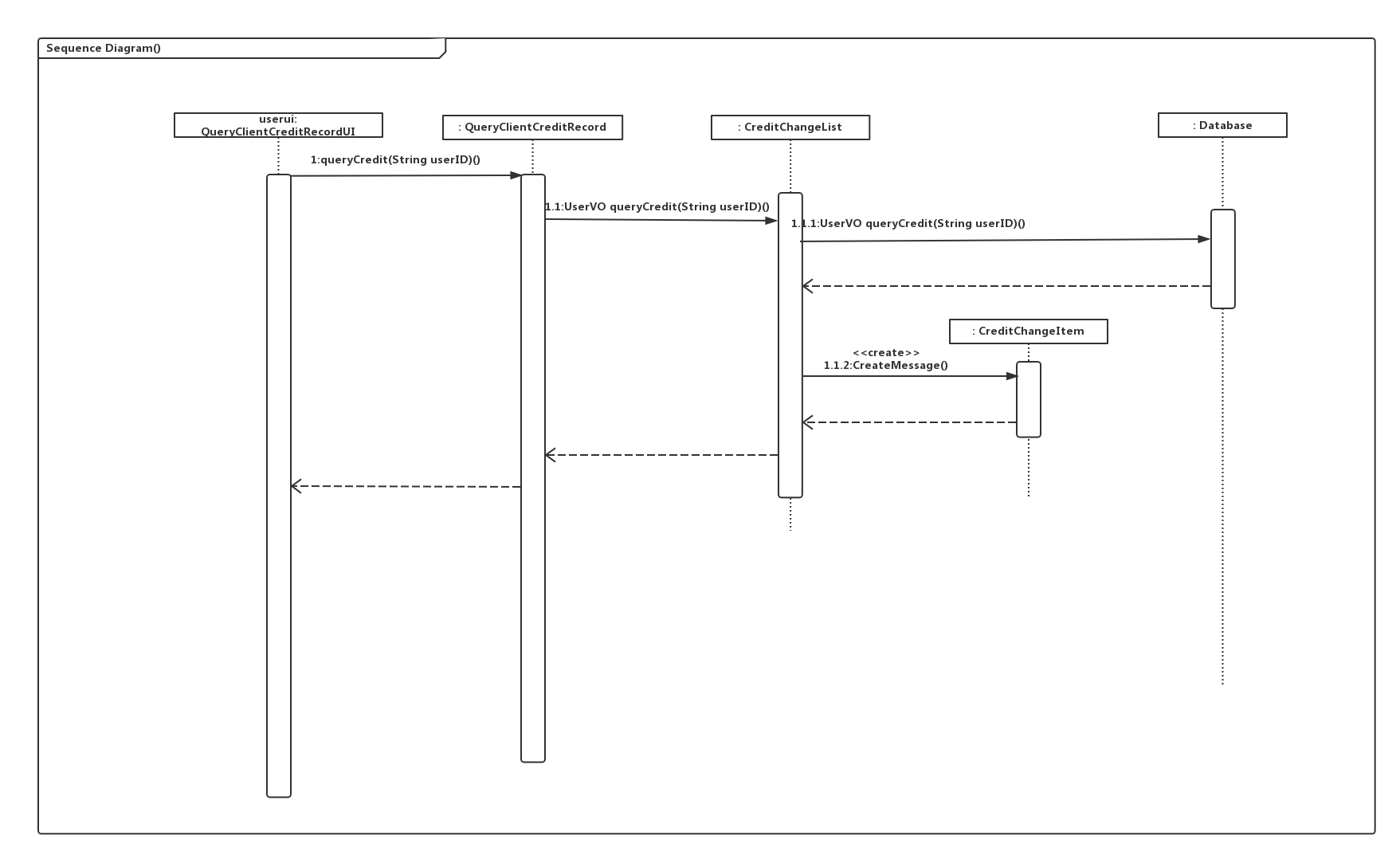
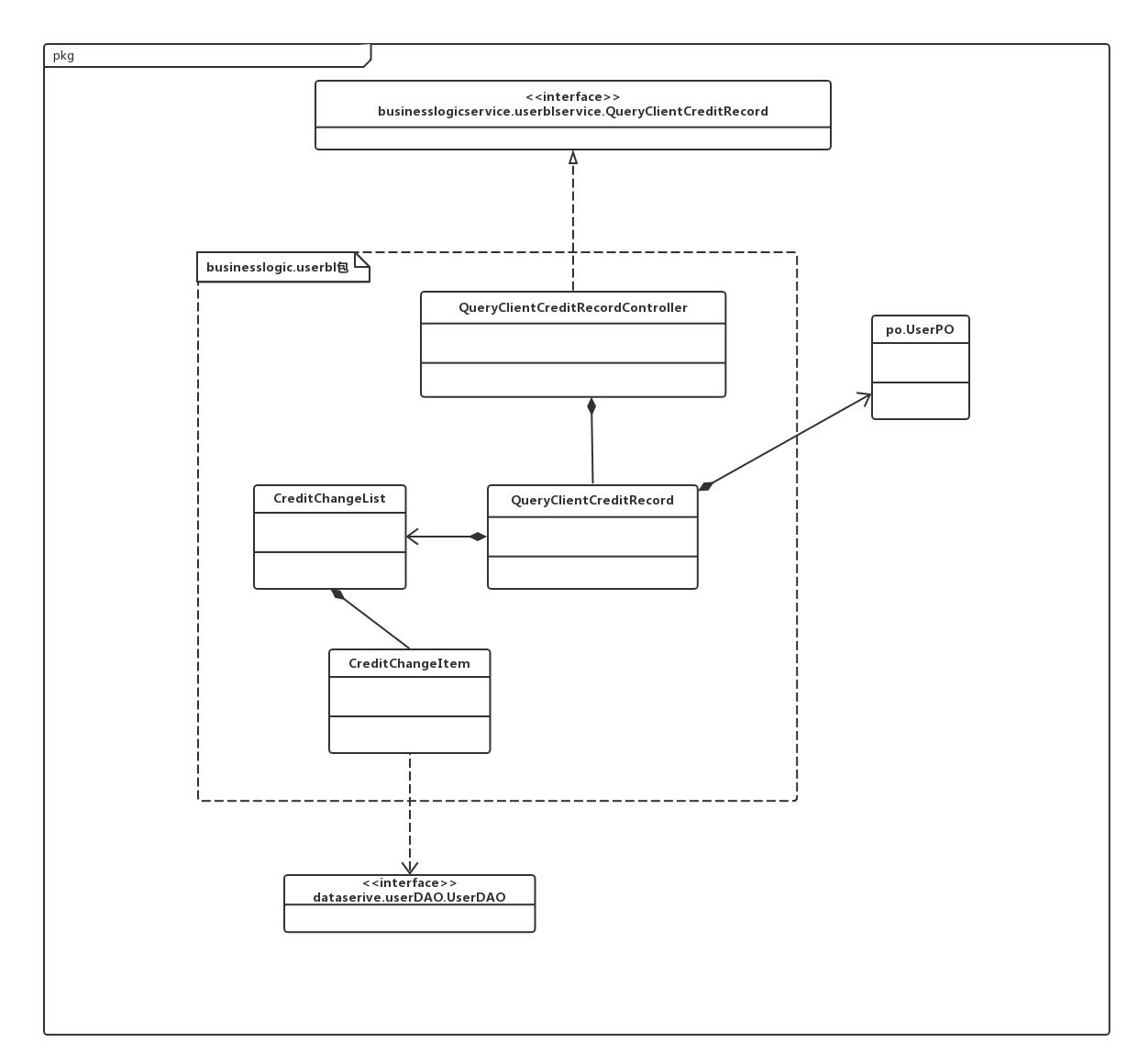
* + - 1. **ModifyClientInfo**

****

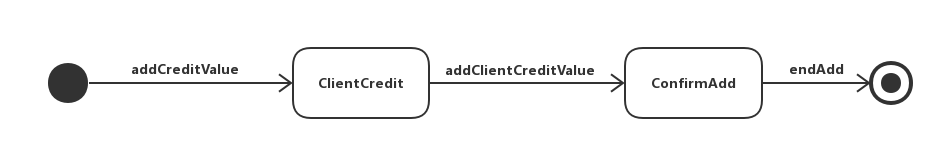
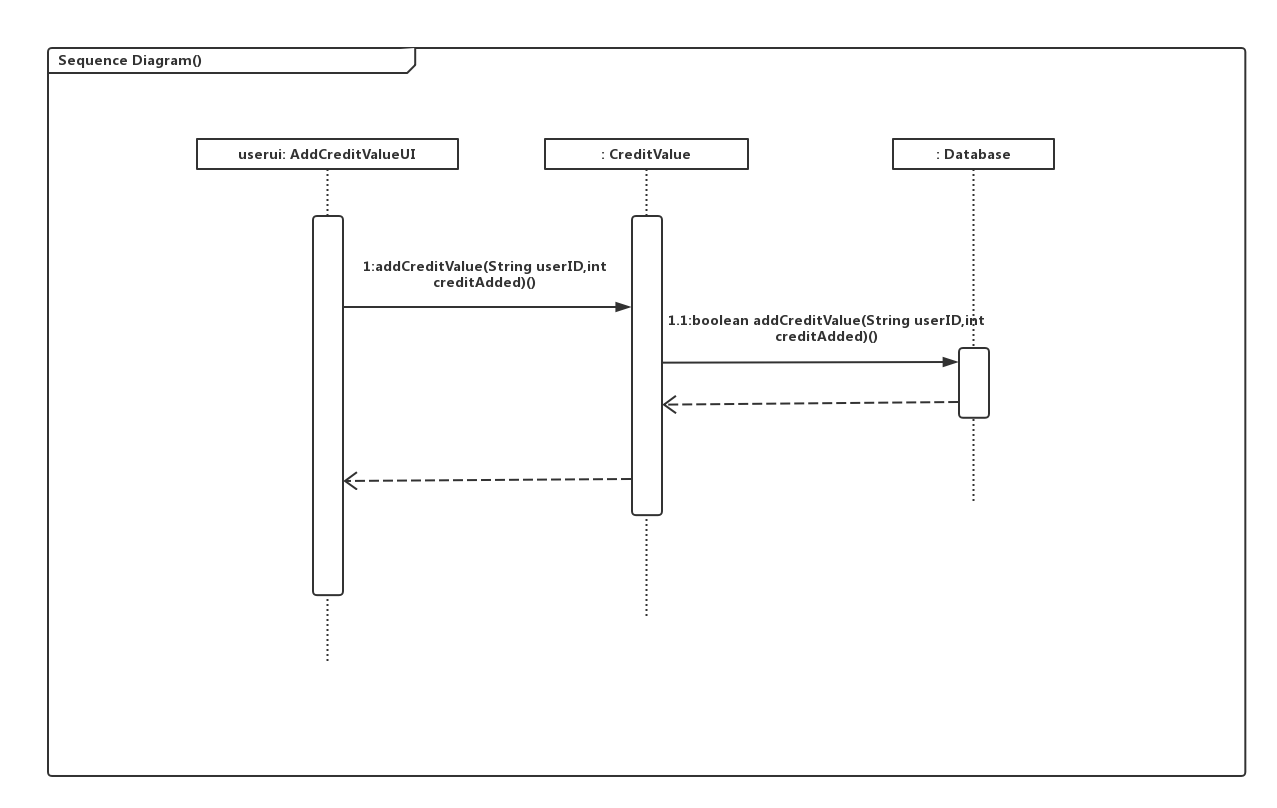
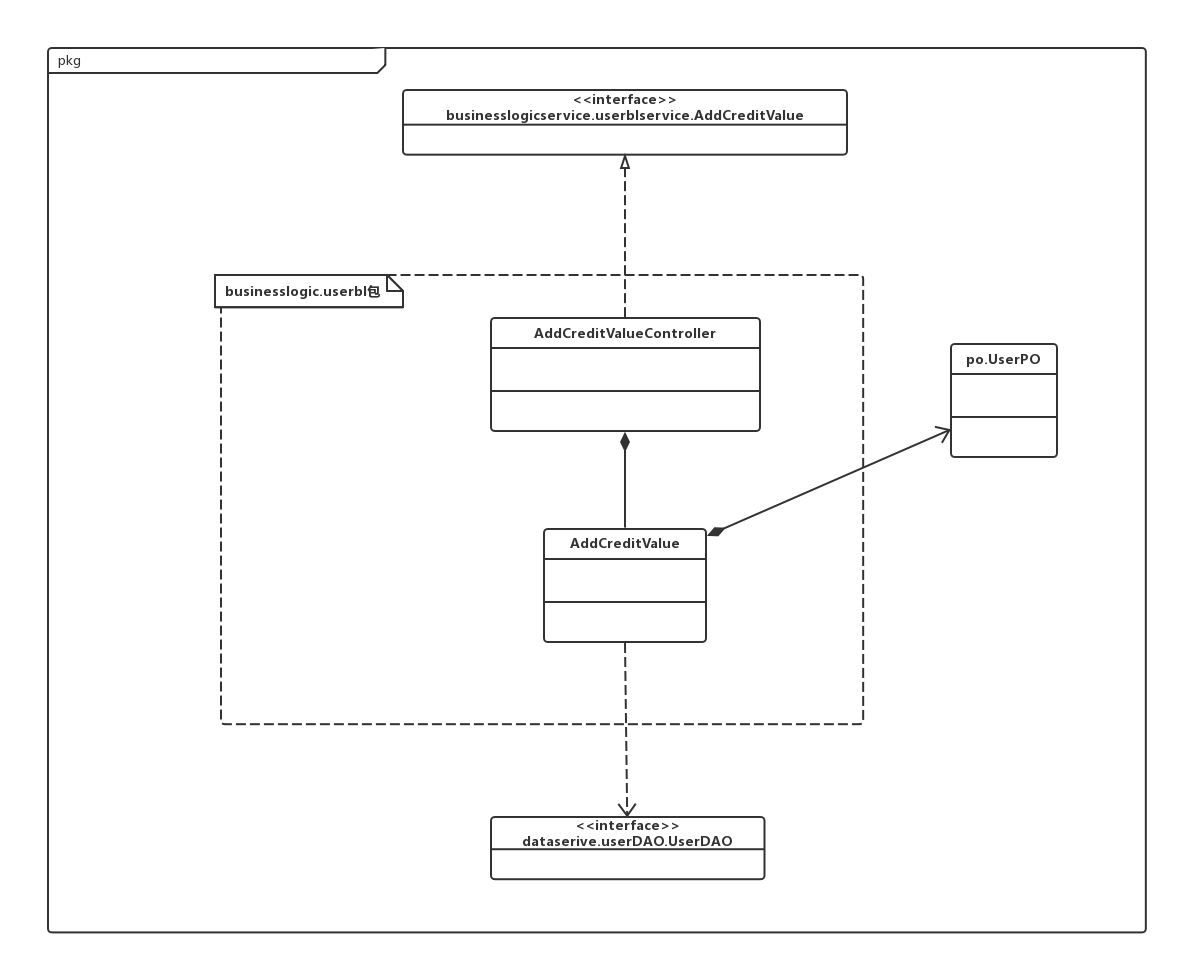
* + - 1. **SignVip**

****

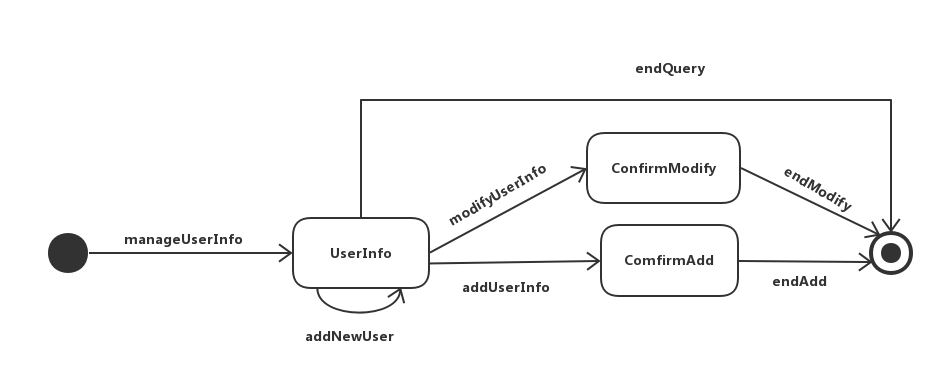
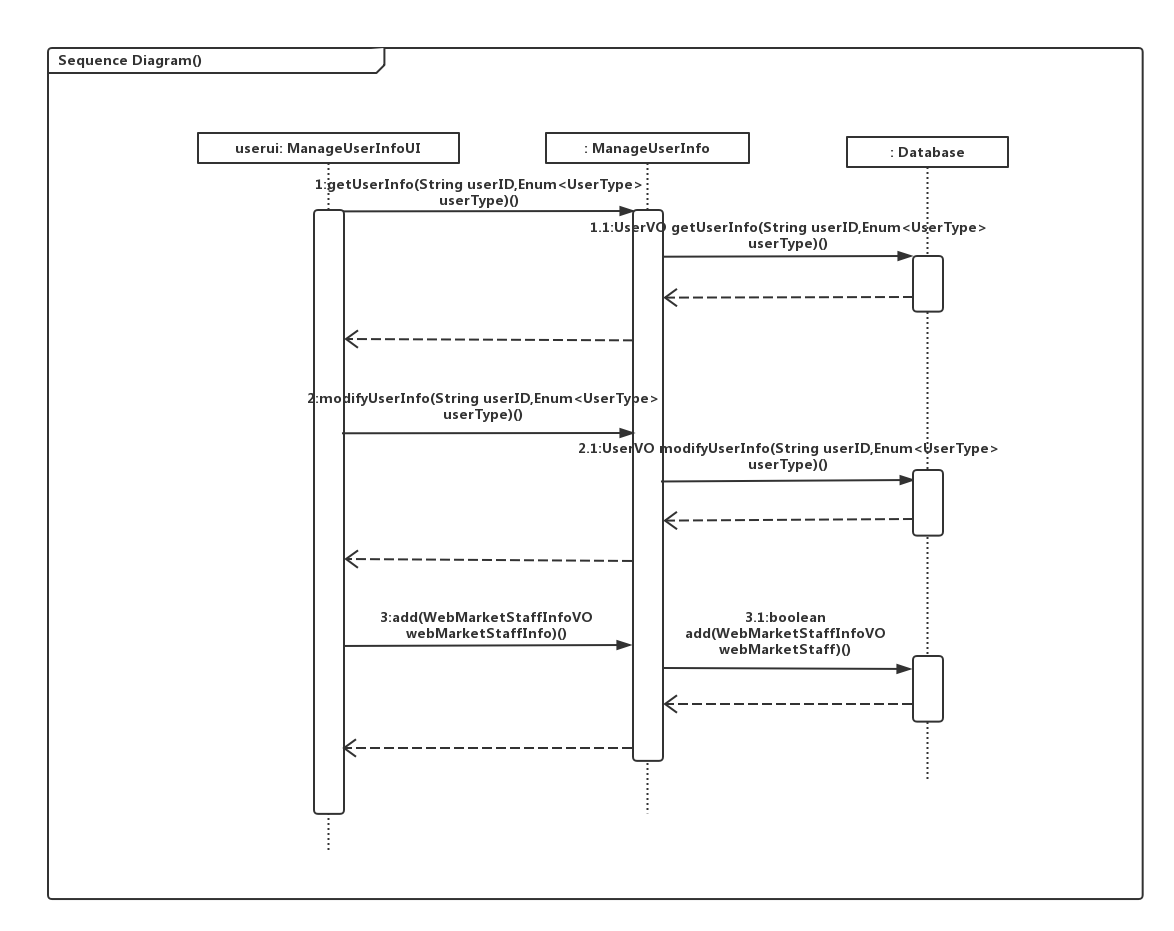
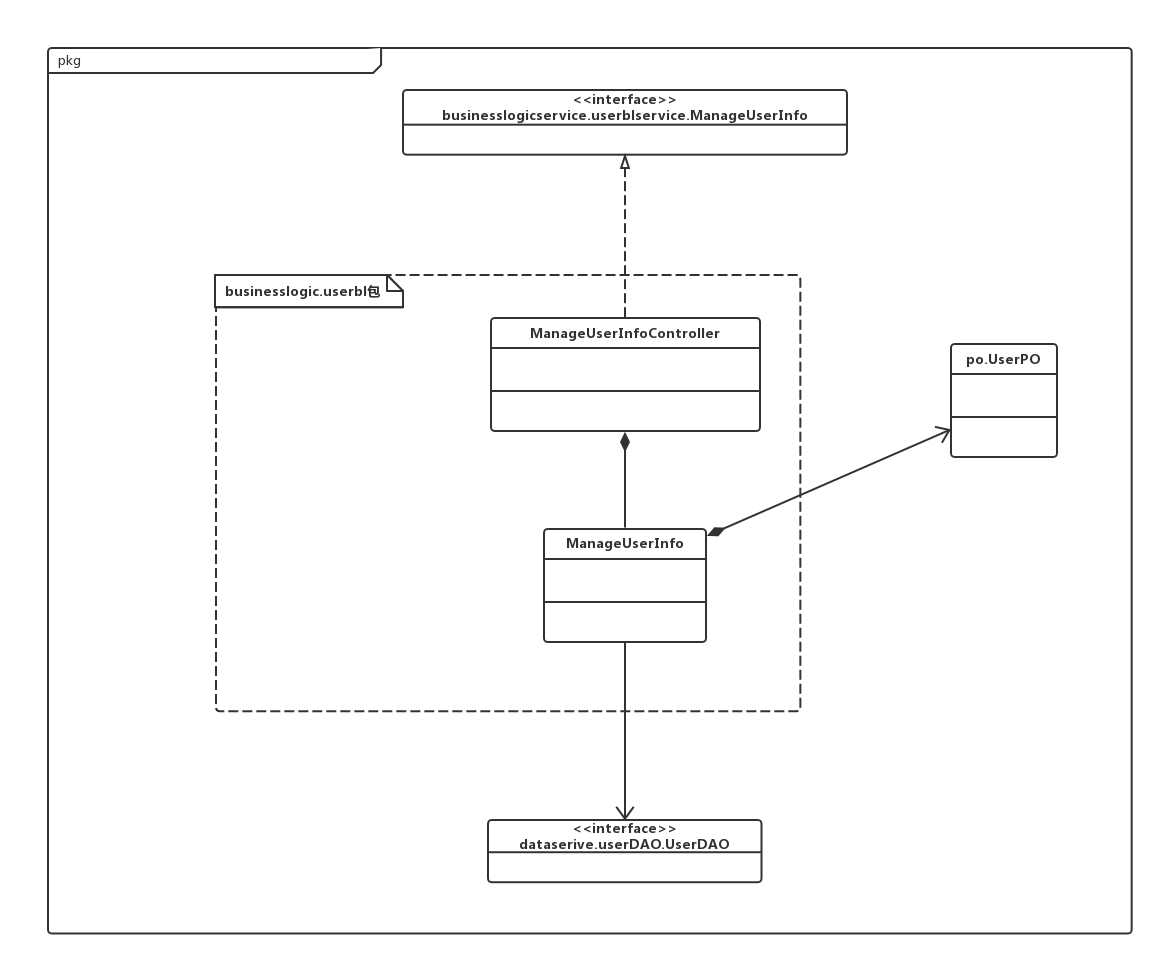
* + - 1. **QueryClientCreditRecord**

****

* + - 1. **AddCreditValue**

****

* + - 1. **ManageUserInfo**

****

* + 1. **orderbl模块**

**（1）模块概述**

orderbl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。

orderbl模块的职责及接口参见软件体系结构描述文档5.3.2.2

**（2）整体结构**

展示层和业务逻辑层之间，我们添加businesslogicservice.orderblservice. BrowseHotelOrderService接口、businesslogicservice.orderblservice. BrowseUserOrderService接口、businesslogicservice.orderblservice. CheckAbnormalOrderService接口、businesslogicservice.orderblservice. CreateNewOrderService接口、businesslogicservice.orderblservice. GetOrderDoneService接口、businesslogicservice.orderblservice. WithdrawOrderService接口。业务逻辑层和数据层之间添加dataservice.orderDAO.OrderDAO接口。由于本模块涉及到的逻辑控制职责较为简单，将其与业务逻辑职责合并形成BrowseHotelOrderServiceImpl、BrowseUserOrderServiceImpl、CheckAbnormalOrderServiceImpl、CreateNewOrderServiceImpl、GetOrderDoneServiceImpl、WithdrawOrderServiceImpl类。OrderPO、BriefOrderInfoPO是作为订单信息的持久化对象被添加到涉及模型中去的。HotelOrderList的添加是HotelOrder的容器类、UserOrderList的添加是UserOrder的容器类、AbnormalOrderList是AbnormalOrder的容器类。

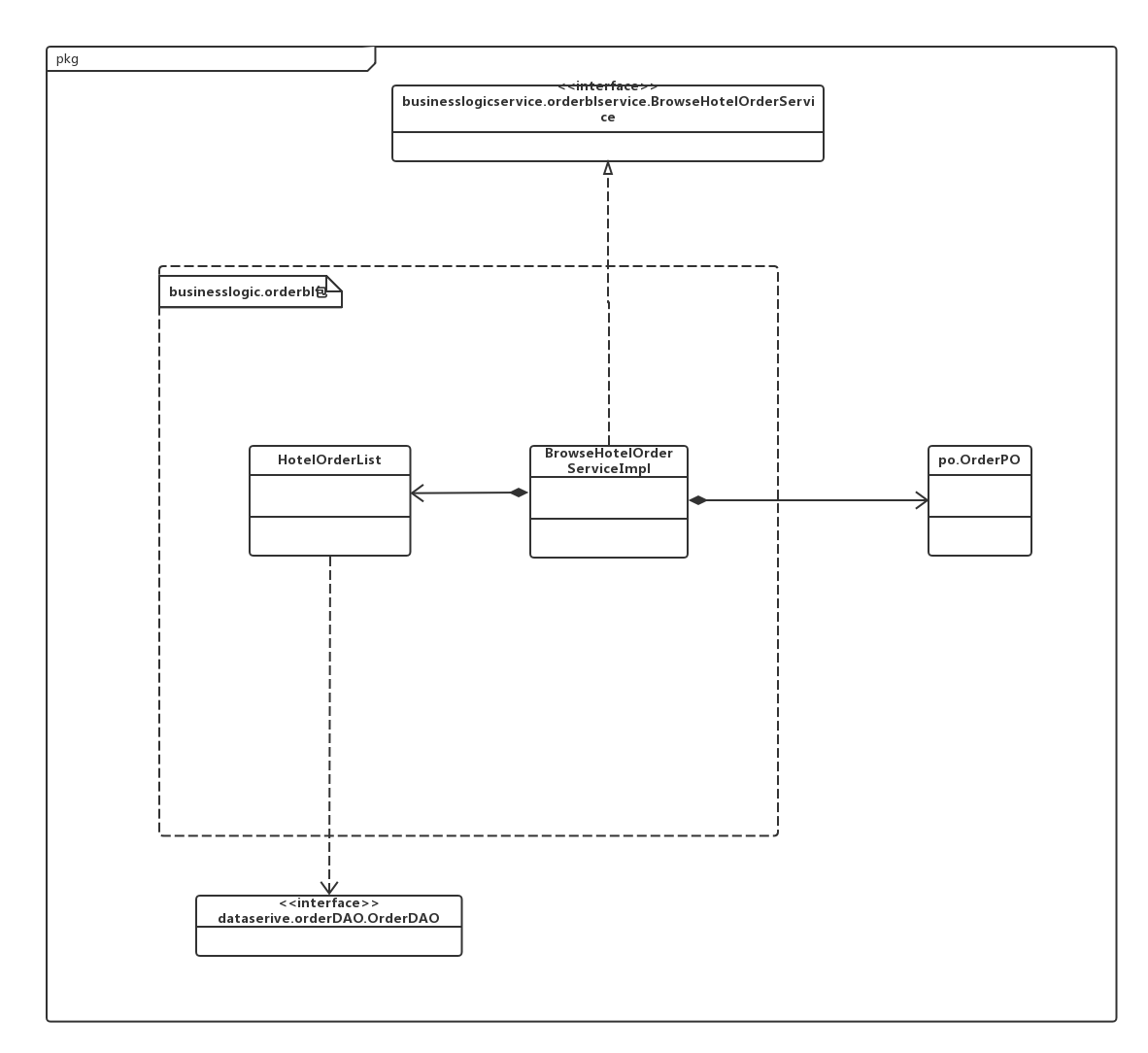
**hotelbl模块各个类的职责如下表所示**

1. **hoetlbl模块各个类的职责**

|  |  |
| --- | --- |
| **类名** | **职责** |
| browseHotelOrder.  BrowseHotelOrderServiceImpl | **负责实现对于浏览酒店订单界面所需要的服务** |
| browseHotelOrder.  HotelOrderList | 酒店订单列表对象，拥有OrderDAO对象引用，提供查看、搜索订单的服务 |
| browseUserOrder.  BrowseUserOrderServiceImpl | **负责实现对于浏览用户订单界面所需要的服务** |
| browseUserOrder.  UserOrderList | 客户订单列表对象，拥有OrderDAO对象引用，提供查看、搜索订单的服务 |
| checkAbnormalOrder.  CheckAbnormalOrderServiceImpl | **负责实现对于处理异常订单界面所需要的服务** |
| checkAbnormalOrder.  AbnormalOrderList | 异常订单列表对象，拥有OrderDAO对象引用，提供查看、搜索订单的服务 |
| checkAbnormalOrder.  SystemOrderWithdrawer | **负责完成网站营销人员撤销订单操作的各种后置条件，拥有OrderDAO、ClientCreditInfo、RoomInfoService对象的引用** |
| createNewOrder.  CreateNewOrderServiceImpl | **负责实现对于生成订单界面所需要的服务** |
| createNewOrder.  Checker | **负责检查某客户是否可以生成订单以及某订单在当前条件下能否被生成** |
| createNewOrder.  NewOrder | **负责初始化订单，计算订单价格，创建订单，拥有OrderDAO的引用，可以向数据库中增加新的订单信息** |
| getOrderDone.  GetOrderDoneServiceImpl | **负责实现对于执行订单界面所需要的服务** |
| getOrderDone.  OrderTerminator | **负责完成执行订单操作的各种后置条件，拥有OrderDAO、ClientCreditInfo、RoomInfoService对象的引用** |
| withdrawOrder.  WithdrawOrderServiceImp | **负责实现对于撤回订单界面所需要的服务** |
| withdrawOrder.  OrderWithdrawer | **负责完成客户撤销订单操作的各种后置条件，拥有OrderDAO、ClientCreditInfo、RoomInfoService对象的引用** |

* + - 1. **BrowseHotelOrder**

**BrowseHotelOrder的设计如下图所示。**



**接口规范：**

BrowseHotelOrderServiceImpl和HotelOrderList的接口规范如下表所示

**BrowseHotelOrderServiceImpl的接口规范**

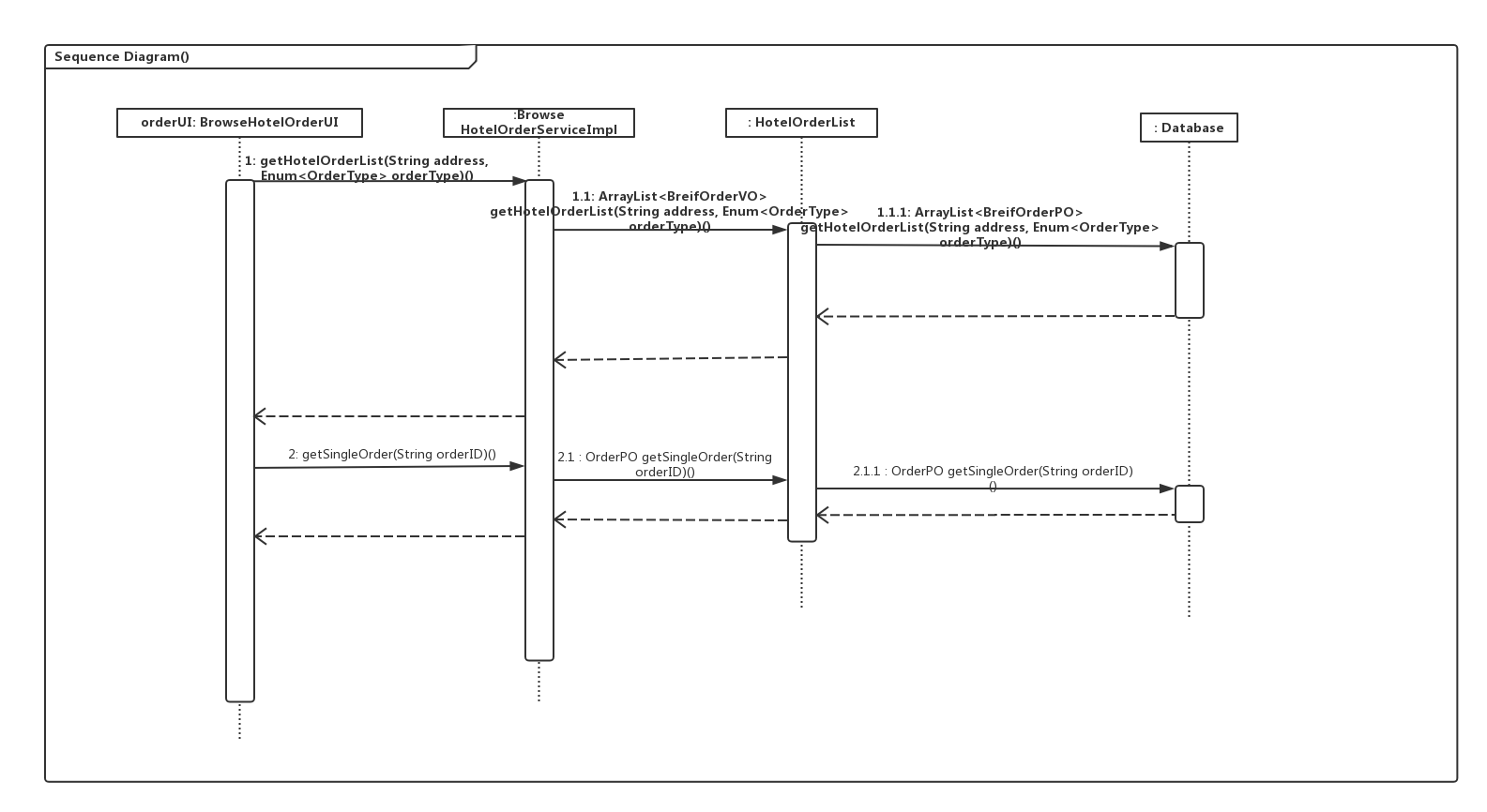
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| BrowseHotelOrderServiceImpl.getHotelOrderList | 语法 | public ArrayList<BriefOrderInfoVO> getHotelOrderList (String address, Enum<OrderType> orderType); |
| 前置条件 | 酒店工作人员启动浏览酒店订单流程，，且一个HotelOrderList对象已经被创建 |
| 后置条件 | 调用HotelOrderList的getHotelOrderList方法 |
| BrowseHotelOrderServiceImpl.getSingleOrder | 语法 | public OrderVO getSingleOrder(String address, String orderID); |
| 前置条件 | 酒店工作人员启动浏览酒店订单流程，，且一个HotelOrderList对象已经被创建 |
| 后置条件 | 调用HotelOrderList的getSingleOrder方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| HotelOrderList.getHotelOrderList(String address, Enum<OrderType> orderType); | 得到该酒店所有订单的列表 | |
| HotelOrderList.getSingleOrder(String address, String orderID); | 检测该订单号是否存在，检测该订单号对应的订单是否属于该酒店，得到该订单号对应的订单详情 | |

**HotelOrderList的接口规范**

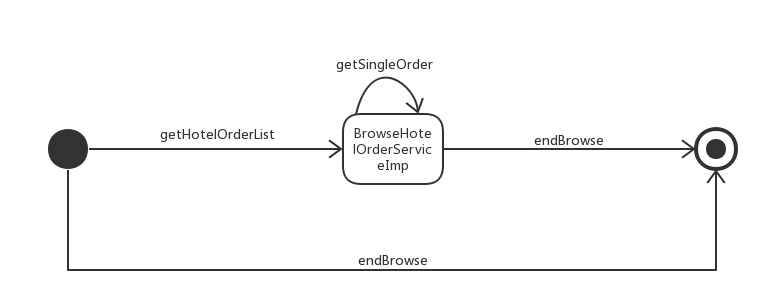
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| HotelOrderList.getHotelOrderList | 语法 | public ArrayList<BriefOrderInfoVO> getHotelOrderList (String address, Enum<OrderType> orderType); |
| 前置条件 | 酒店工作人员启动浏览酒店订单流程 |
| 后置条件 | 按orderType的值进行查找并按生成时间顺序返回相应的订单列表 |
| HotelOrderList.getSingleOrder | 语法 | public OrderVO getSingleOrder(String address, String orderID); |
| 前置条件 | 酒店工作人员已经输入订单号 |
| 后置条件 | 检测该订单号是否存在，如果存在，检测是否是该酒店的订单，若是则返回对应订单的VO，否则返回null |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderDAO.getHotelOrderList(String address, Enum<OrderType> orderType) | 得到按orderType的值进行查找并按时间顺序返回该酒店的相应订单列表 | |
| OrderDAO.getSingleOrder(String address, String orderID) | 检测该酒店的订单号是否存在，如果存在，则返回对应订单的VO，否则返回null | |

**业务逻辑层的动态模型**

**BrowseHotelOrder的顺序图如下图所示。**

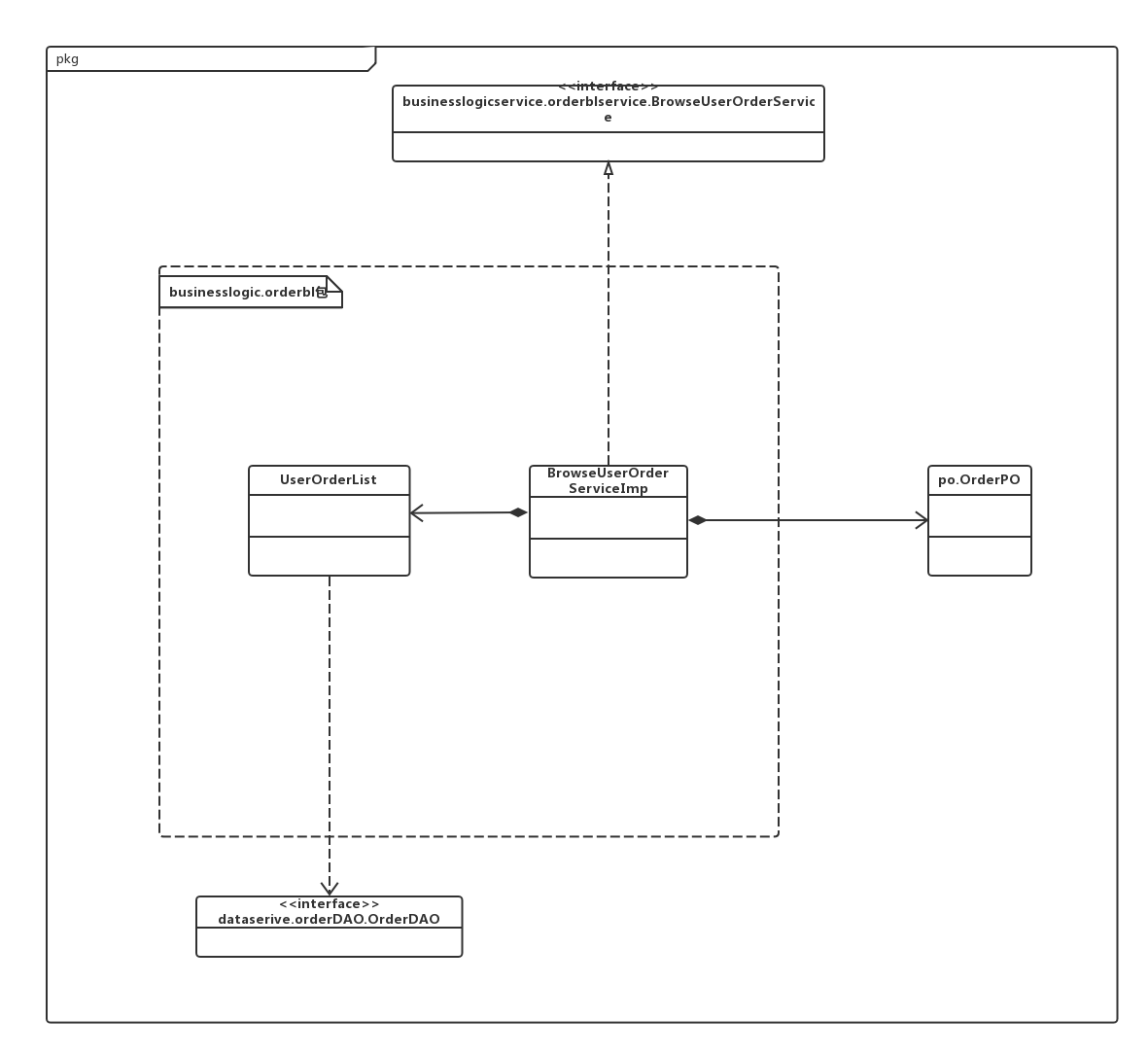
****

**BrowseHotelOrder的状态图如下图所示。**

****

* + - 1. **BrowseUserOrder**

**BrowseUserOrder的设计如下图所示。**



**接口规范：**

BrowseUserOrderServiceImpl和UserOrderList的接口规范如下表所示

**BrowseUserOrderServiceImpl的接口规范**

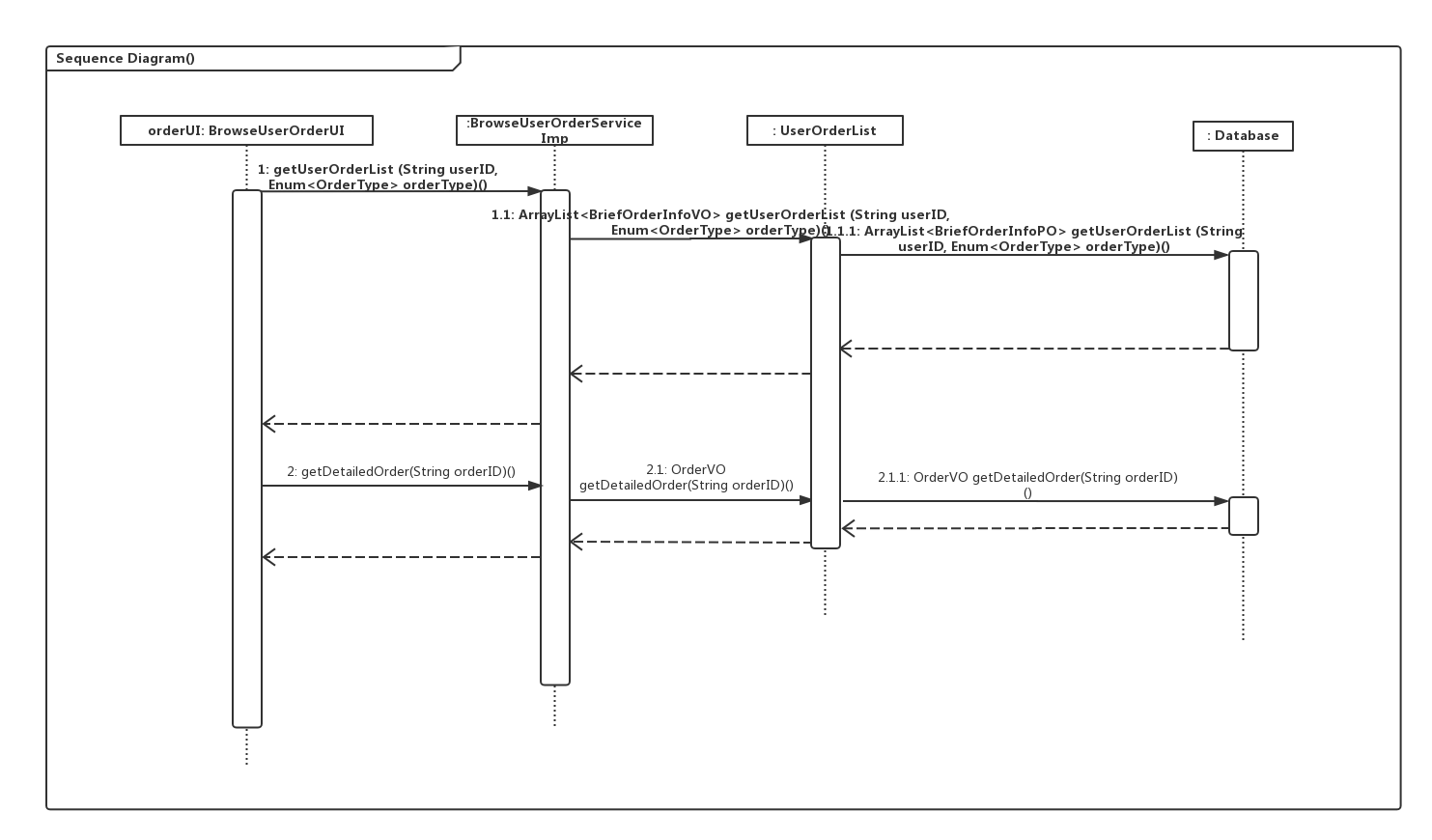
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| BrowseUserOrderServiceImpl.getUserOrderList | 语法 | public ArrayList<BriefOrderInfoVO> getUserOrderList (String userID, Enum<OrderType> orderType); |
| 前置条件 | 客户启动浏览订单流程，，且一个UserOrderList对象已经被创建 |
| 后置条件 | 调用UserOrderList的getUserOrderList方法 |
| BrowseUserOrderServiceImpl.getDetailedOrder | 语法 | public OrderVO getDetailedOrder (String orderID); |
| 前置条件 | 客户启动浏览订单流程，，且一个UserOrderList对象已经被创建 |
| 后置条件 | 调用UserOrderList的getDetailedOrder方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| UserOrderList.getUserOrderList(String userID, Enum<OrderType> orderType); | 得到该客户所有订单的列表 | |
| UserOrderList.getDetailedOrder(String orderID); | 检测该订单号是否存在，得到该订单号对应的订单详情 | |

**UserOrderList的接口规范**

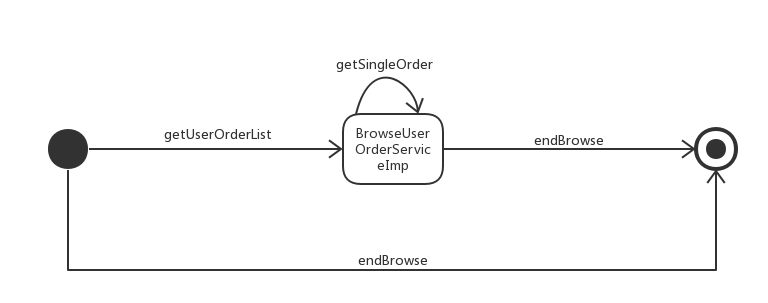
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UserOrderList.getUserOrderList | 语法 | public ArrayList<BriefOrderInfoVO> getUserOrderList (String userID, Enum<OrderType> orderType); |
| 前置条件 | 客户启动浏览订单流程 |
| 后置条件 | 按orderType的值进行查找并按生成时间顺序返回相应的订单列表 |
| Order.getDetailedOrder | 语法 | public OrderVO getDetailedOrder(String orderID); |
| 前置条件 | 客户请求展开订单详情 |
| 后置条件 | 检测订单号是否存在，如果存在，则返回对应订单的VO，否则返回null |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderDAO.getUserOrderList(String userID, Enum<OrderType> orderType) | 根据orderType的值得到当前用户的对应订单持久化对象列表 | |
| OrderDAO.getDetailedOrder(String orderID) | 检测该订单号是否存在，如果存在，则返回对应订单的VO，否则返回null | |

**业务逻辑层的动态模型**

**BrowseUserOrder的顺序图如下图所示。**

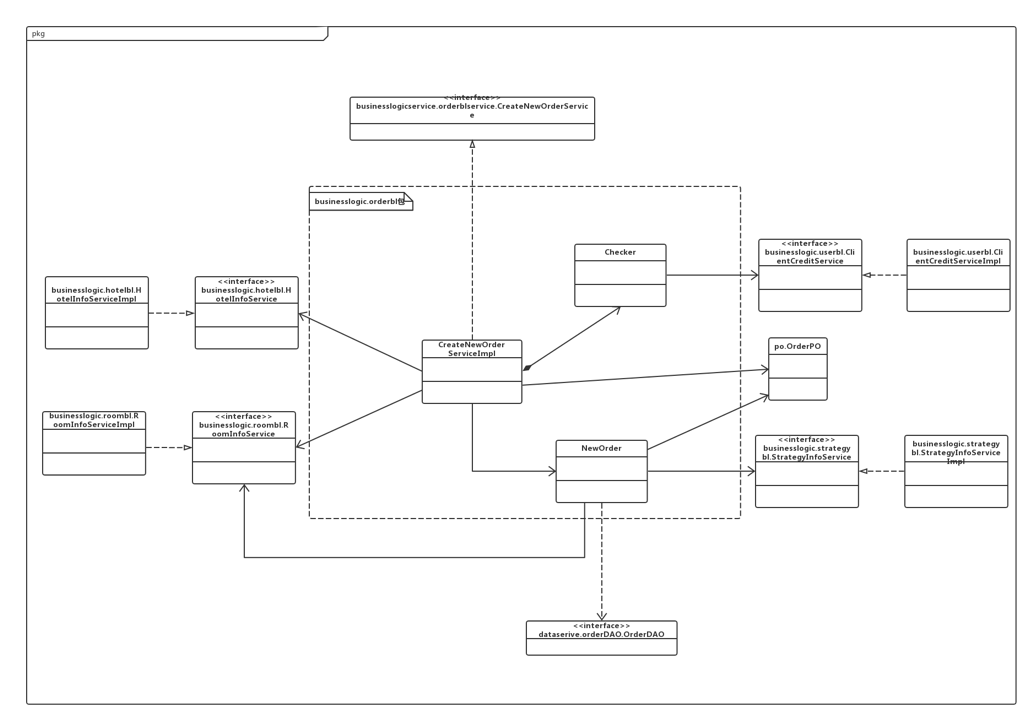
****

**BrowseUserOrder的状态图如下图所示。**

****

* + - 1. **CreateNewOrder**

**CreateNewOrder的设计如下图所示。**



**接口规范：**

CreateNewOrderServiceImpl、NewOrder和Checker的接口规范如下表所示

**CreateNewOrderServiceImpl的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CreateNewOrderServiceImpl.initNewOrder | 语法 | public OrderVO initNewOrder(String userID, String address); |
| 前置条件 | 客户确认即将生成订单的酒店信息，且一个Checker和一个NewOrder对象均被创建 |
| 后置条件 | 调用Checker的canUserCreateNewOrder方法，调用NewOrder的initNewOrder方法 |
| CreateNewOrderServiceImpl. getAvailableRoomNum | 语法 | public int getAvailableRoomNum(String address, Enum<RoomType> roomType); |
| 前置条件 | 客户已经选择房型，且一个RoomInfoService 实现类已经被创建 |
| 后置条件 | 调用RoomInfoService的getAvailableRoomNum方法 |
| CreateNewOrderServiceImpl. checkNewOrder | 语法 | public boolean checkNewOrder( OrderVO vo); |
| 前置条件 | 客户确认生成订单，且一个Checker已被创建 |
| 后置条件 | 调用checker的checkNewOrder方法 |
| CreateNewOrderServiceImpl. getPrice | 语法 | public int getPrice( OrderVO vo); |
| 前置条件 | Checker已经检查过可以生成订单，且一个NewORder对象已被创建 |
| 后置条件 | 调用NewOrder的getPrice方法 |
| CreateNewOrderServiceImpl. addNewOrder | 语法 | public boolean addNewOrder( OrderVO vo); |
| 前置条件 | Checker已经检查过可以生成订单，且一个NewORder对象已被创建 |
| 后置条件 | 调用NewOrder的addNewOrder方法 |
|  | | |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| Checker.canUserCreateNewOrder(String userID) | 判断该客户能否生成订单 | |
| Checker. checkNewOrder(OrderVO vo) | 检查该订单在当前时刻能否被生成 | |
| NewOrder.initNewOrder(String userID, String address) | 初始化订单信息，返回一个被初始化的OrderVO | |
| NewOrder.getPrice(OrderVO vo) | 计算并返回订单价格 | |
| NewOrder.addNewOrder(OrderVO vo) | 增加新的订单 | |
| RoomInfoService.getAvailableRoomNum(String address, Enum<RoomType> roomType) | 获取当前酒店roomType房型的空房数量 | |

**Checker的接口规范**

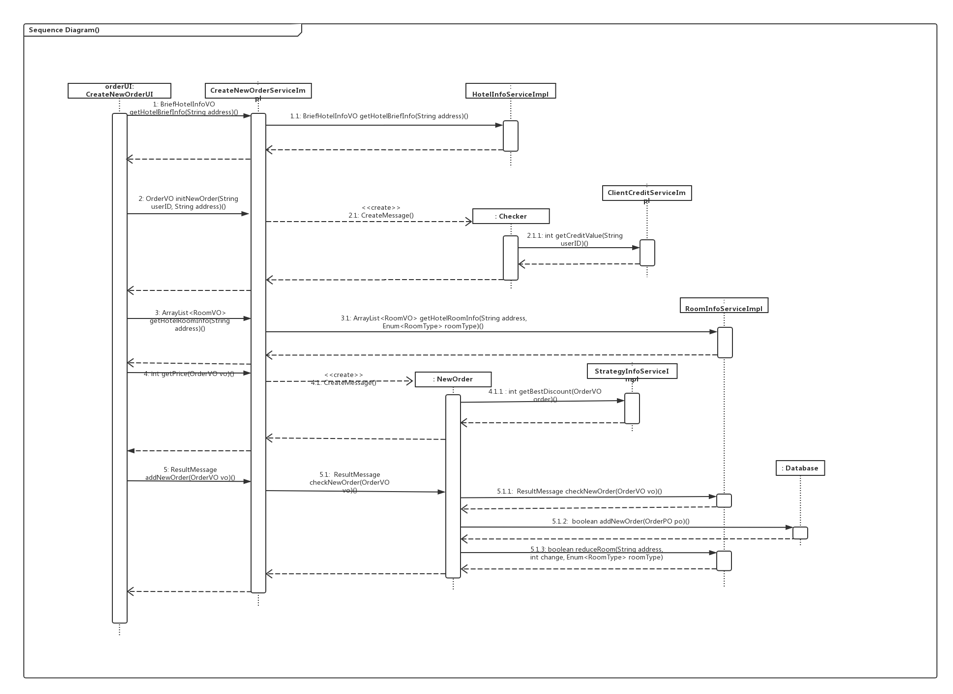
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Checker.canUserCreateNewOrder | 语法 | public boolean canUserCreateNewOrder(String userID) |
| 前置条件 | 客户启动浏览订单流程 |
| 后置条件 | 判断该客户能否生成订单，返回一个布尔值 |
| Checker. checkNewOrder | 语法 | public ResultMessage checkNewOrder(OrderVO vo) |
| 前置条件 | 客户确认生成订单 |
| 后置条件 | 检查该订单在当前时刻能否被生成，返回一个布尔值 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| ClientCreditInfo. getCreditValue(String userID); | 检查该订单在当前时刻能否被生成，返回一个布尔值 | |
| RoomInfoService.checkNewOrder(OrderVO vo) | 检查该订单在当前时刻能否被生成，返回一个布尔值 | |

**NewOrder的接口规范**

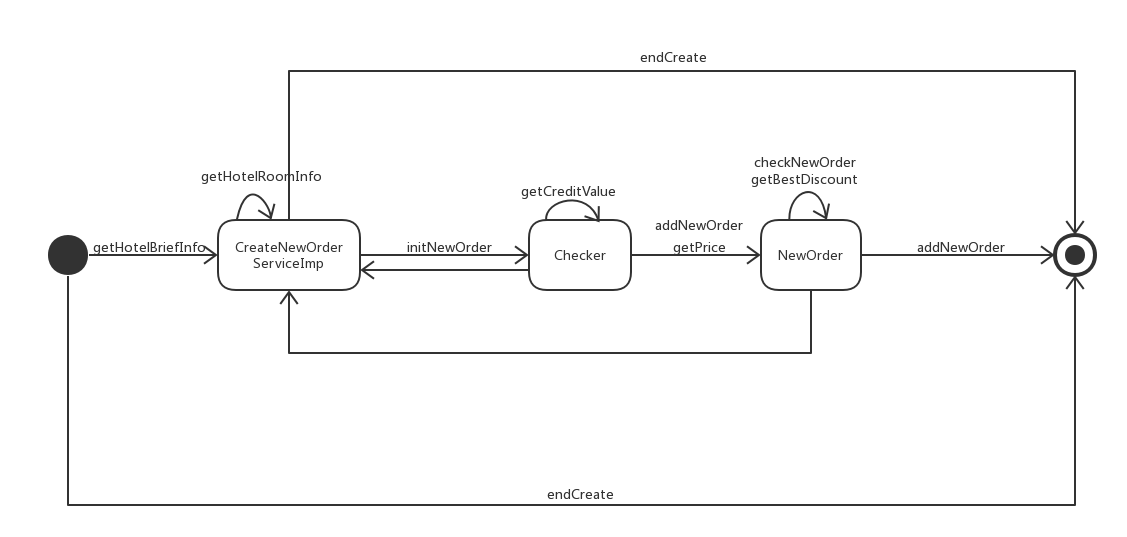
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| NewOrder.initNewOrder | 语法 | public OrderVO initNewOrder(String userID, String address); |
| 前置条件 | 客户选择生成订单 |
| 后置条件 | 初始化订单信息，返回一个被初始化的OrderVO |
| NewOrder.getPrice | 语法 | public int getPrice(OrderVO vo); |
| 前置条件 | 客户确认生成订单 |
| 后置条件 | 计算并返回订单价格 |
| NewOrder.addNewOrder(OrderVO vo) | 语法 | public boolean addNewOrder(OrderVO vo); |
| 前置条件 | 客户确认生成订单 |
| 后置条件 | 增加新的订单 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderDAO.insert(OrderPO po) | 插入单一持久化对象 | |
| OrderDAO.finish() | 结束持久化数据库的使用 | |
| DatabaseFactory.getOrderDatabase | 得到Order数据库的服务的引用 | |
| Strategy.getBestDiscount(OrderVO vo) | 得到当前最终的最优的优惠折扣 | |

**业务逻辑层的动态模型**

**CreateNewOrder的顺序图如下图所示。**

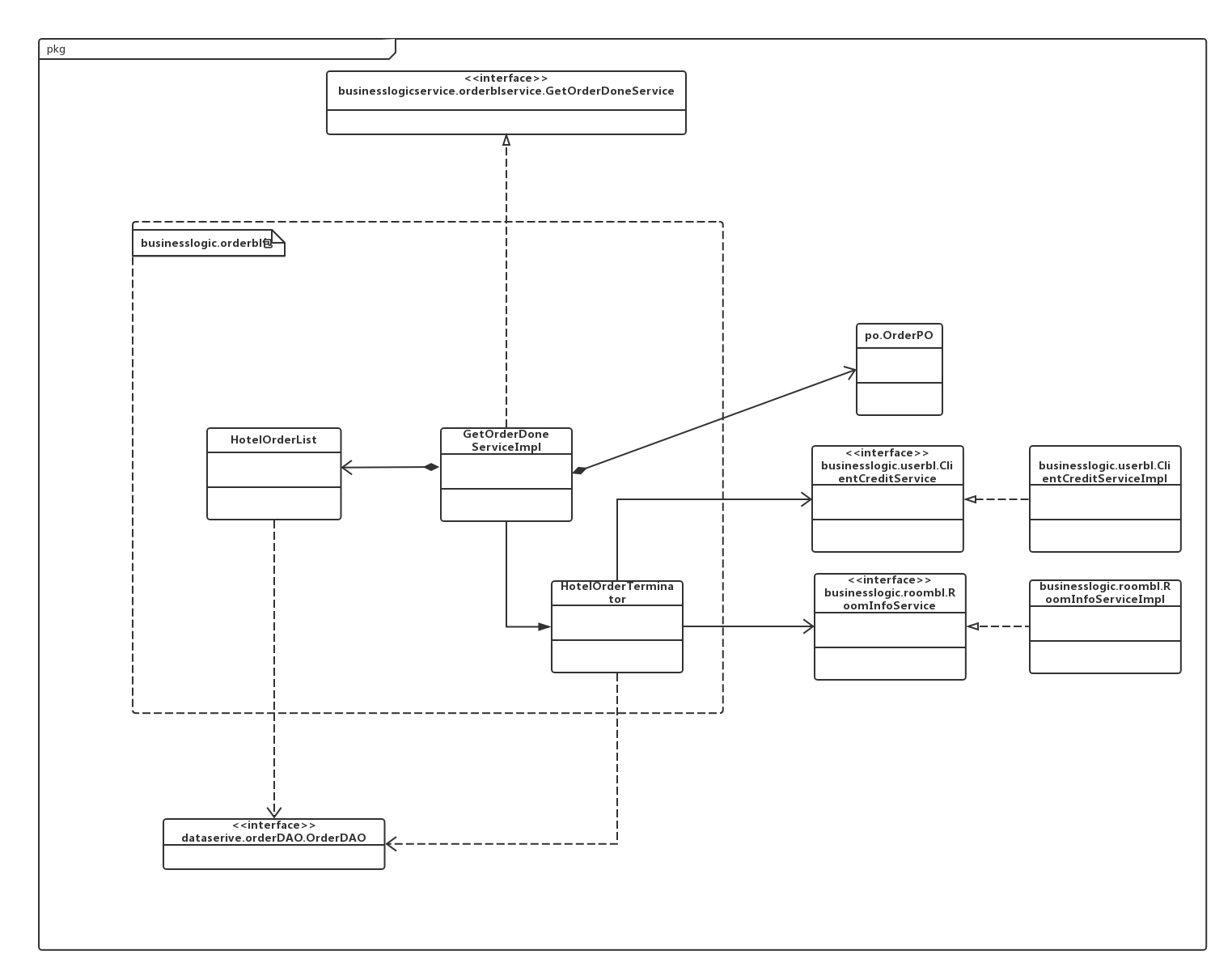
****

**CreateNewOrder的状态图如下图所示。**

****

* + - 1. **GetOrderDone**

**GetOrderDone的设计如下图所示。**



**接口规范：**

GetOrderDoneServiceImpl和OrderTerminator的接口规范如下表所示

**GetOrderDoneServiceImpl的接口规范**

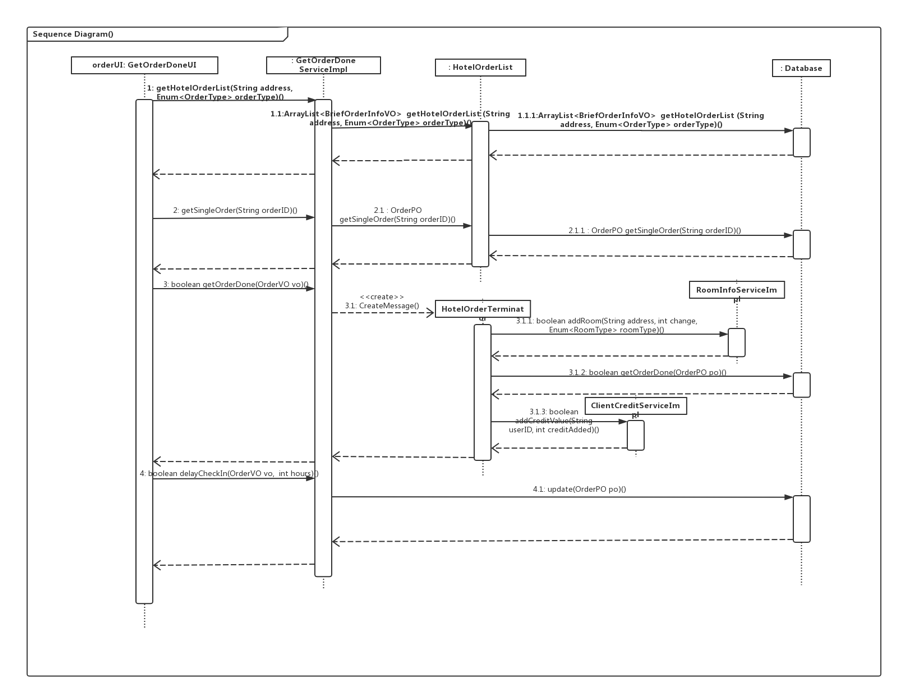
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| GetOrderDoneServiceImpl.getHotelNotDoneOrderList | 语法 | public ArrayList<BriefOrderInfoVO> getHotelNotDoneOrderList(String address); |
| 前置条件 | 酒店工作人员启动执行酒店订单流程，且一个HotelOrderList对象已经被创建 |
| 后置条件 | 调用HotelOrderList的getHotelOrderList方法 |
| GetOrderDoneServiceImpl. getSingleOrder | 语法 | public OrderVO getSingleOrder(String address, String orderID); |
| 前置条件 | 酒店工作人员启动执行酒店订单流程，且一个HotelOrderList对象已经被创建 |
| 后置条件 | 调用HotelOrderList的getSingleOrder方法 |
| GetOrderDoneServiceImpl. getOrderDone | 语法 | public boolean getOrderDone(OrderVO vo); |
| 前置条件 | 酒店工作人员选择执行酒店订单，且一个OrderTerminator对象已经被创建 |
| 后置条件 | 调用OrderTerminator的getOrderDone方法 |
| GetOrderDoneServiceImpl. delayCheckIn | 语法 | public boolean delayCheckIn (OrderVO vo); |
| 前置条件 | 酒店工作人员选择延迟入住，且一个OrderTerminator对象已经被创建 |
| 后置条件 | 调用OrderTerminator的delayCheckIn方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| HotelOrderList.getHotelOrderList(String address, Enum<OrderType> orderType); | 得到该酒店对应类型订单的列表（此处为未执行订单） | |
| HotelOrderList.getSingleOrder(String address, String orderID); | 检测该订单号是否存在，检测该订单号对应的订单是否属于该酒店，得到该订单号对应的订单详情 | |
| OrderTerminator.getOrderDone(OrderVO vo) | 将该订单改为已执行状态，记录入住时间，然后为该客户增加与订单价值等额的信用值, 并增加可用客房 | |
| OrderTerminator.delayCheckIn(OrderVO vo) | 将该订单置为已执行订单，记录入住时间，恢复扣除的信用值, 并增加可用客房 | |

**OrderTerminator的接口规范**

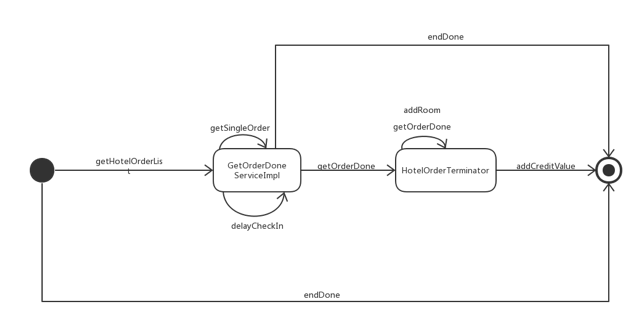
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| GetOrderDoneServiceImpl. getOrderDone | 语法 | public boolean getOrderDone(OrderVO vo); |
| 前置条件 | 酒店工作人员选择执行酒店订单 |
| 后置条件 | 将该订单改为已执行状态，记录入住时间，然后为该客户增加与订单价值等额的信用值, 并增加可用客房 |
| GetOrderDoneServiceImpl. delayCheckIn | 语法 | public boolean delayCheckIn (OrderVO vo); |
| 前置条件 | 酒店工作人员选择延迟入住 |
| 后置条件 | 将该订单置为已执行订单，记录入住时间，恢复扣除的信用值, 并增加可用客房 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderDAO.update(OrderPO po) | 更新单一持久化对象 | |
| OrderDAO.finish() | 结束持久化数据库的使用 | |
| DatabaseFactory.getOrderDatabase | 得到Order数据库的服务的引用 | |
| RoomInfoService.addSpareRoom(String address, int change, Enum<RoomType> roomType) | 更新（增加）当前酒店被预定房型的可用房间数据 | |
| ClientCreditInfo.changeCreditValue(String userID, int num) | 根据num(可正可负)的值，增加或减少当前客户的信用值 | |

**业务逻辑层的动态模型**

**GetOrderDone的顺序图如下图所示。**

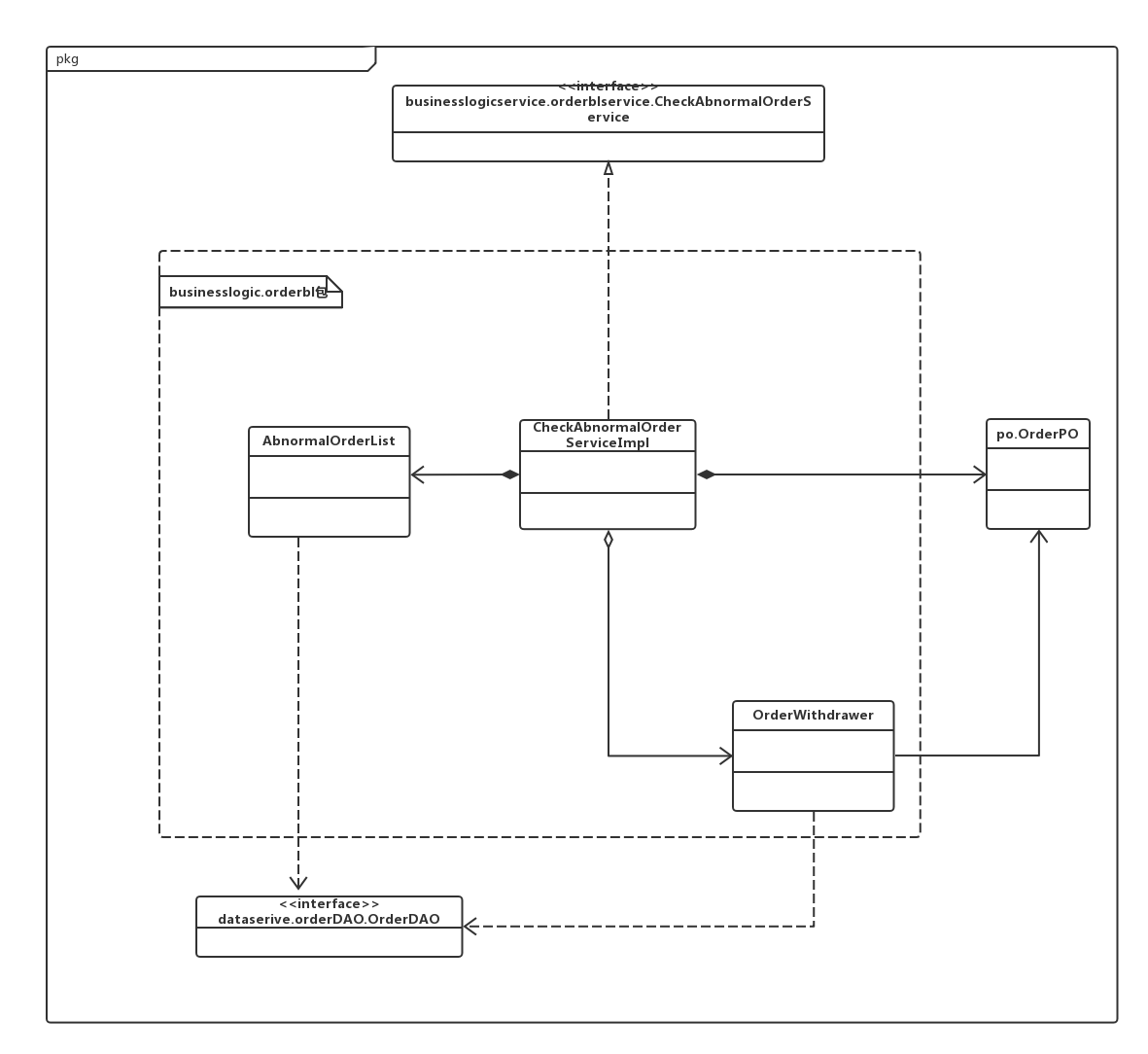
****

**GetOrderDone的状态图如下图所示。**

****

* + - 1. **CheckAbnormalOrder**

**CheckAbnormalOrder的设计如下图所示。**



**接口规范：**

CheckAbnormalOrderServiceImpl、AbnormalOrderList和SystemOrderWithdrawer的接口规范如下表所示

**CheckAbnormalOrderServiceImpl的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CheckAbnormalOrderServiceImpl.getAbnormalOrderList | 语法 | public ArrayList<BriefOrderInfoVO> getAbnormalOrderList(Date date); |
| 前置条件 | 网站营销人员启动浏览每日异常订单流程，且一个AbnormalOrderList对象已经被创建 |
| 后置条件 | 调用AbnormalOrderList的getAbnormalOrderList方法 |
| CheckAbnormalOrderServiceImpl.getDetailedOrder | 语法 | public OrderVO getDetailedOrder (String orderID); |
| 前置条件 | 网站营销人员请求浏览订单详情，且一个AbnormalOrderList对象已经被创建 |
| 后置条件 | 调用AbnormalOrderList的getDetailedOrder方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| AbnormalOrderList. getAbnormalOrderList(Date date) | 得到该日期所有异常订单的列表 | |
| AbnormalOrderList.getDetailedOrder(String orderID) | 检测该订单号是否存在，得到该订单号对应的订单详情 | |

**AbnormalOrderList的接口规范**

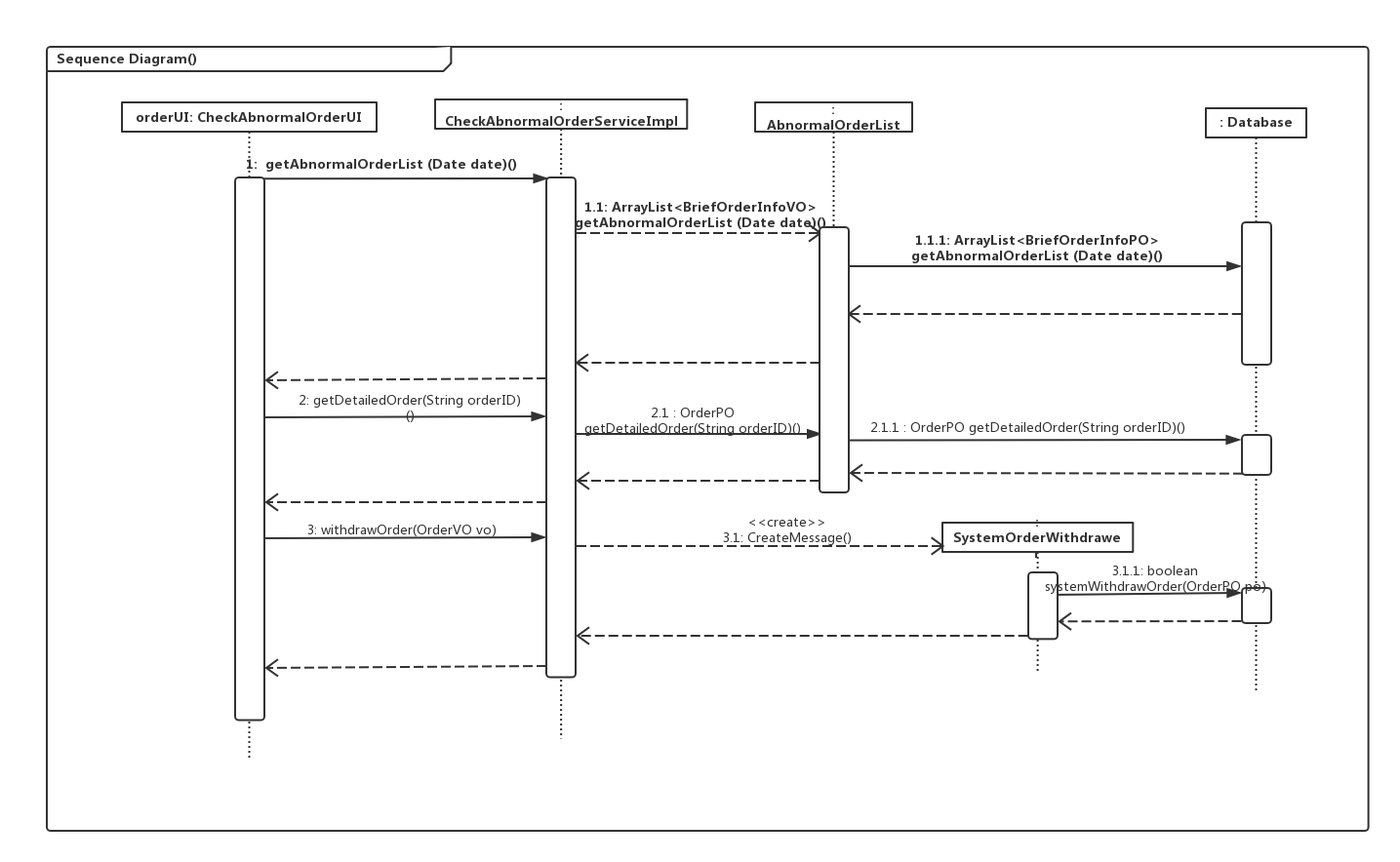
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| AbnormalOrderList. getAbnormalOrderList | 语法 | public ArrayList<BriefOrderInfoVO> getAbnormalOrderList (Date date); |
| 前置条件 | 网站营销人员启动浏览每日异常订单流程 |
| 后置条件 | 得到改日期所有异常订单的列表 |
| AbnormalOrderList.getDetailedOrder | 语法 | public OrderVO getDetailedOrder(String orderID); |
| 前置条件 | 网站营销人员请求浏览订单详情 |
| 后置条件 | 检测订单号是否存在，如果存在，则返回对应订单的VO，否则返回null |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderDAO.getAllAbnormalList (Date date) | 得到所查询日期当天所有异常订单的列表 | |
| OrderDAO.getDetailedOrder(String orderID) | 检测该订单号是否存在，如果存在，则返回对应订单的VO，否则返回null | |

**SystemOrderWithdrawer的接口规范**

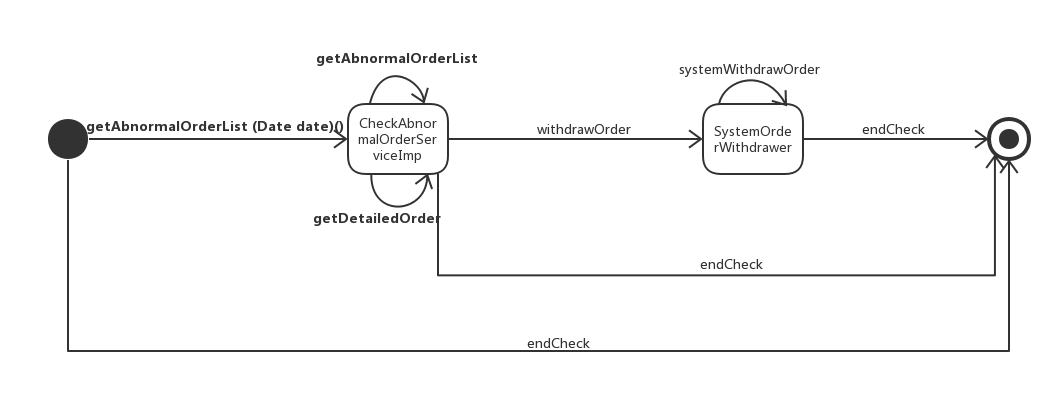
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SystemOrderWithdrawer.systemWithdrawOrder(OrderVO vo, boolean isRecoverHalfCredit) | 语法 | public boolean systemWithdrawOrder(OrderVO vo, boolean isRecoverHalfCredit); |
| 前置条件 | 网站营销人员确认撤销订单，且已经选择了恢复客户已经扣除的信用值的全部或一半 |
| 后置条件 | 撤销此异常订单并将其状态置为已撤销、记录撤销时间，恢复此客户信用值的全部或一半 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderDAO.update(OrderPO po) | 更新单一持久化对象 | |
| OrderDAO.finish() | 结束持久化数据库的使用 | |
| DatabaseFactory.getOrderDatabase | 得到Order数据库的服务的引用 | |
| RoomInfoService.addSpareRoom(String address, int change, Enum<RoomType> roomType) | 更新（增加）当前酒店被预定房型的可用房间数据 | |
| ClientCreditInfo.changeCreditValue(String userID, int num) | 根据num(可正可负)的值，增加或减少当前客户的信用值 | |

**业务逻辑层的动态模型**

**CheckAbnormalOrder的顺序图如下图所示。**

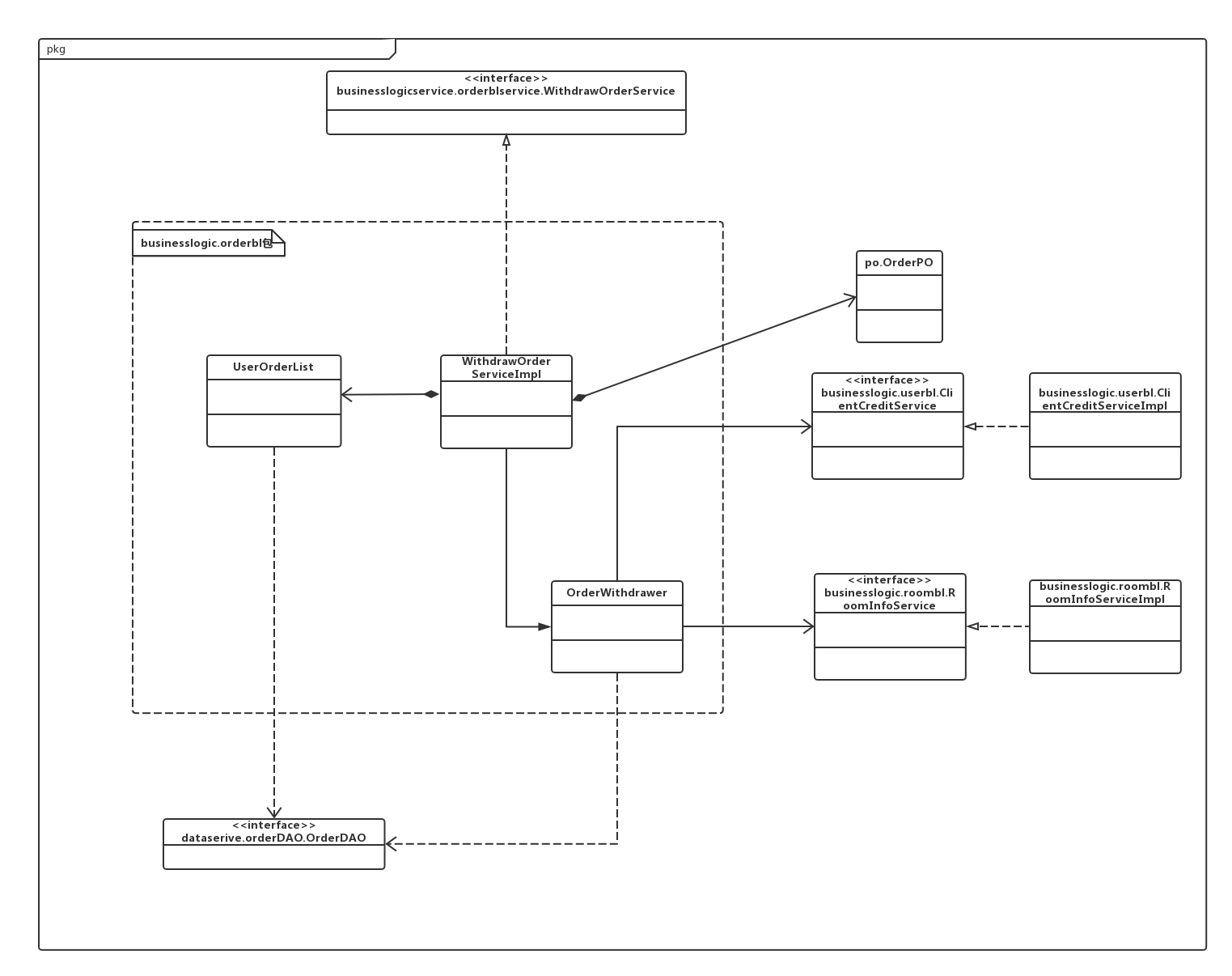
****

**BrowseUserOrder的状态图如下图所示。**

****

* + - 1. **WithdrawOrder**

**WithdrawOrder的设计如下图所示。**



**接口规范：**

WithdrawOrderServiceImpl和OrderWithdrawer的接口规范如下表所示

**WithdrawOrderServiceImpl的接口规范**

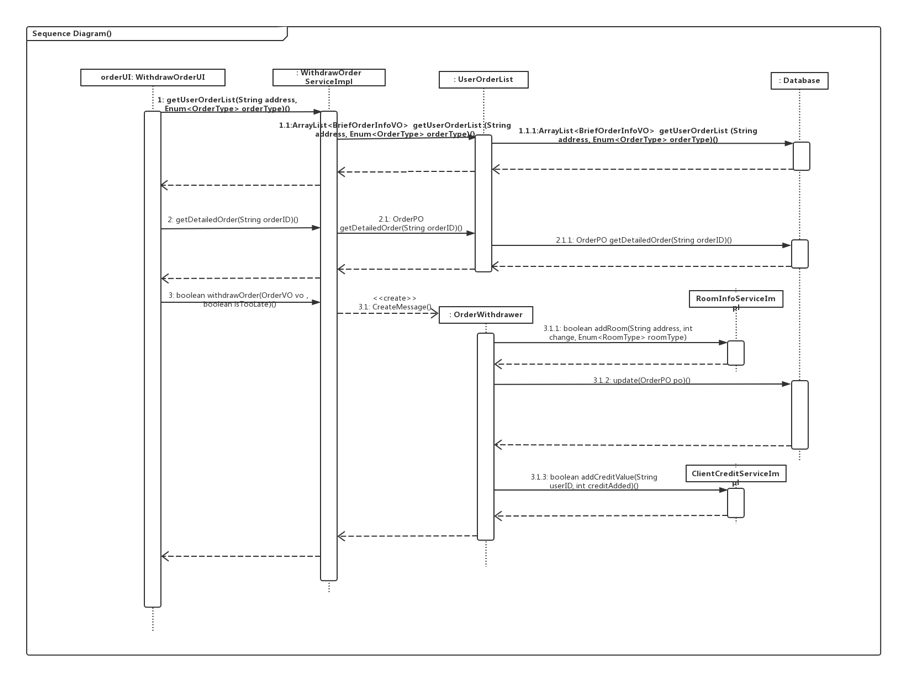
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| WithdrawOrderServiceImpl.getUserOrderList | 语法 | public ArrayList<BriefOrderInfoVO> getUserOrderList (String userID, Enum<OrderType> orderType); |
| 前置条件 | 客户启动浏览订单流程，，且一个UserOrderList对象已经被创建 |
| 后置条件 | 调用UserOrderList的getUserOrderList方法 |
| WithdrawOrderServiceImpl.getDetailedOrder | 语法 | public OrderVO getDetailedOrder (String orderID); |
| 前置条件 | 客户启动浏览订单流程，，且一个UserOrderList对象已经被创建 |
| 后置条件 | 调用UserOrderList的getDetailedOrder方法 |
| WithdrawOrderServiceImpl. withdrawOrder | 语法 | public boolean withdrawOrder(OrderVO vo, boolean isTooLate); |
| 前置条件 | 客户确认撤销订单，且一个OrderWithdrawer对象已经被创建 |
| 后置条件 | 调用OrderWithdrawer的withdrawOrder方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| UserOrderList.getUserOrderList(String userID, Enum<OrderType> orderType); | 得到该客户所有订单的列表 | |
| UserOrderList.getSingleOrder(String orderID); | 检测该订单号是否存在，得到该订单号对应的订单详情 | |
| OrderWithdrawer. withdrawOrder(OrderVO vo, boolean isTooLate) | 更改订单信息，置为已撤销状态，记录撤销时间, 根据isTooLate更改客户信用值 | |

**OrderWithdrawer的接口规范**

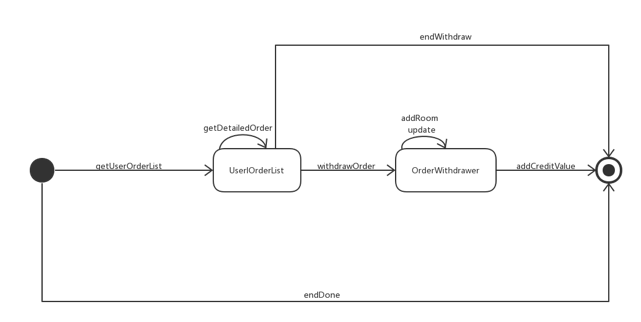
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderWithdrawer. withdrawOrder | 语法 | public boolean withdrawOrder(OrderVO vo, boolean isTooLate); |
| 前置条件 | 客户确认撤销订单 |
| 后置条件 | 更改订单信息，置为已撤销状态，记录撤销时间, 根据isTooLate更改客户信用值 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderDAO.update(OrderPO po) | 更新单一持久化对象 | |
| OrderDAO.finish() | 结束持久化数据库的使用 | |
| DatabaseFactory.getOrderDatabase | 得到Order数据库的服务的引用 | |
| RoomInfoService.addSpareRoom(String address, int change, Enum<RoomType> roomType) | 更新（增加）当前酒店被预定房型的可用房间数据 | |
| ClientCreditInfo.changeCreditValue(String userID, int num) | 根据num(可正可负)的值，增加或减少当前客户的信用值 | |

**业务逻辑层的动态模型**

**OrderWithdrawer的顺序图如下图所示。**

****

**OrderWithdrawer的状态图如下图所示。**

****

* + 1. **hotelbl模块**

**（1）模块概述**

hotelbl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公能需求。

hotelbl模块的职责及接口参见软件体系结构描述文档5.3.2.3

**（2）整体结构**

展示层和业务逻辑层之间，我们添加businesslogicservice.hotelblservice. CheckOrderedHotelService接口、businesslogicservice.hotelblservice. CommentOnHotelService接口、businesslogicservice.hotelblservice. ImportNewRoomService接口、businesslogicservice.hotelblservice. MaintainHotelBasicInfoService接口、businesslogicservice.hotelblservice. ManageHotelInfoService接口、businesslogicservice.hotelblservice. QueryHotelService接口、businesslogicservice.hotelblservice. SearchHotelService接口。业务逻辑层和数据层之间添加dataservice.hotelDAO.HotelDAO接口。由于本模块涉及到的逻辑控制职责较为简单，将其与业务逻辑职责合并形成CheckOrderedHotelServiceImpl、CommentOnHotelServiceImpl、ImportNewRoomServiceImpl、MaintainHotelBasicInfoServiceImpl、ManageHotelInfoServiceImpl、QueryHotelServiceImpl、SearchHotelServiceImpl类。HotelPO、BriefHotelInfoPO是作为销售记录的持久化对象被添加到涉及模型中去的。OrderedHotelList和OrderedHotelItem的添加是OrderedHotelInfo的容器类、CommentableHotelList的添加是CommentableHotel的容器类、AvailableRoomList和AvailableRoomItem的添加是AvailableRoom的容器类、QueryHotelList的添加是QueryHotel的容器类。HotelInfoService、OrderInfo都是根据依赖倒置原则，为了消除循环依赖而产生的接口。

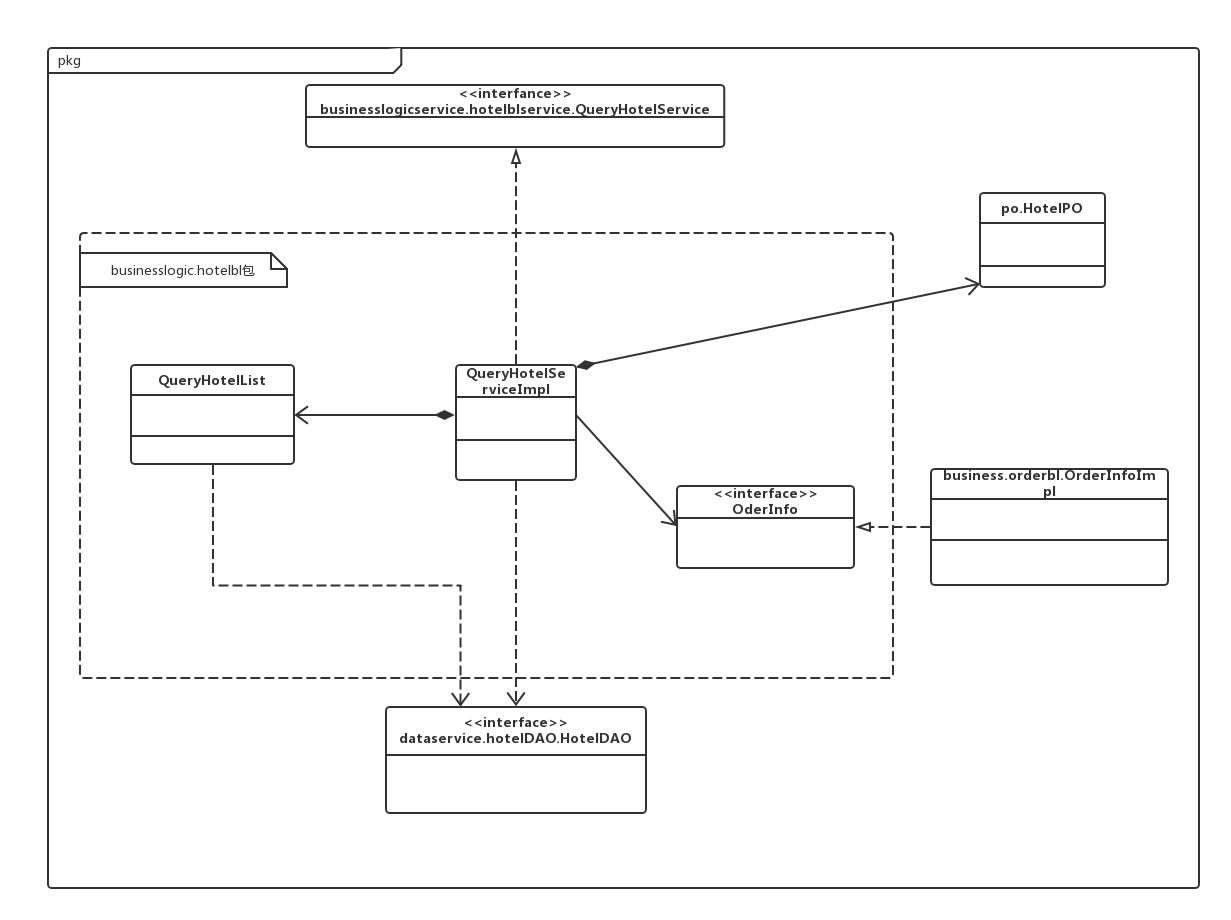
**hotelbl模块各个类的职责如下表所示**

**hoetlbl模块各个类的职责**

|  |  |
| --- | --- |
| **类名** | **职责** |
| checkOrderedHotel. CheckOrderedHotelServiceImpl | **负责实现对于浏览预定过的酒店界面所需要的服务** |
| checkOrderedHotel. OrderedHotelList |  |
| checkOrderedHotel. OrderedHotelItem |  |
| commentOnHotel. CommentOnHotelServiceImpl |  |
| commentOnHotel. CommentableOrderList |  |
| importNewRoom. ImportNewRoomServiceImpl |  |
| importNewRoom. AvailableRoomList |  |
| importNewRoom. AvailableRoomItem |  |
| maintainHotelBasicInfo. MaintainHotelBasicInfoServiceImpl |  |
| manageHotelInfo. ManageHotelInfoServiceImpl |  |
| queryHotel. QueryHotelList |  |
| queryHotel. QueryHotelServiceImpl |  |
| searchHotel. SearchHotelServiceImpl |  |

* + - 1. **QueryHotel**

**QueryHotel的设计如下图所示。**



**接口规范：**

QueryHotelServiceImpl和QueryHotelList的接口规范如下表所示

**QueryHotelServiceImpl的接口规范**

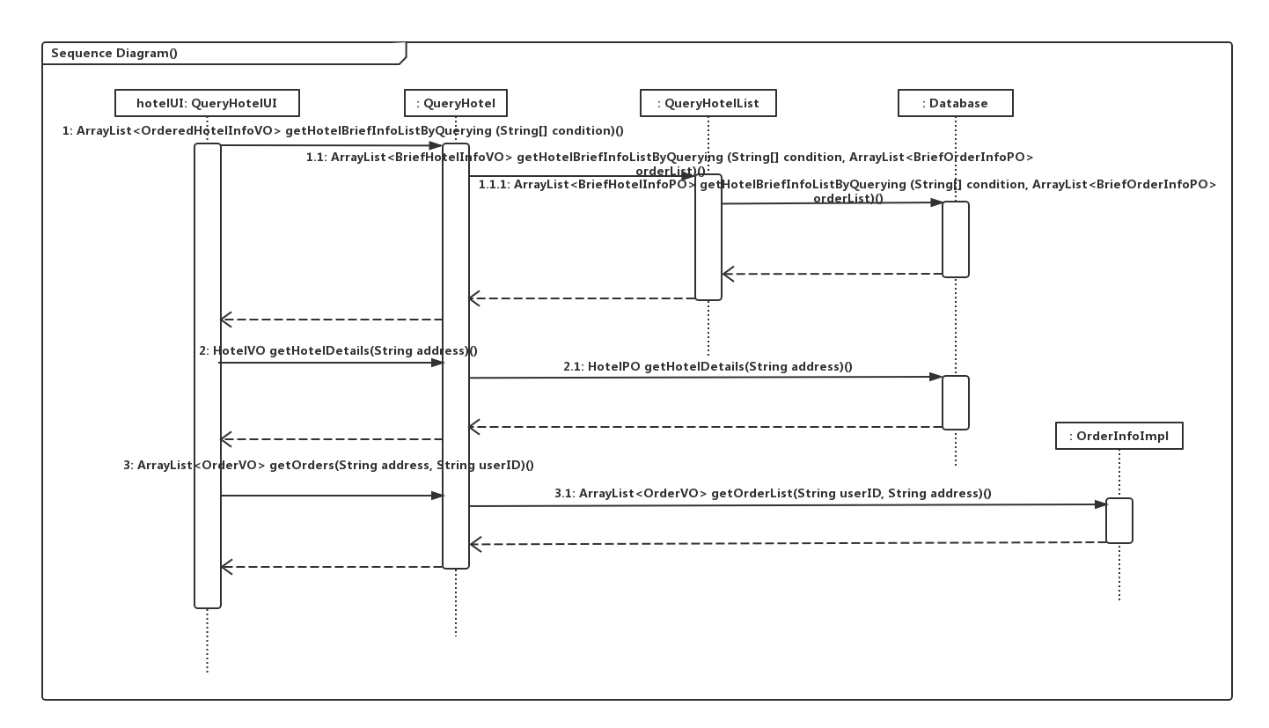
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| QueryHotelServiceImpl.getHotelBriefInfoListByQuerying | 语法 | public ArrayList<OrderedHotelInfoVO> getHotelBriefInfoListByQuerying(String[] condition) |
| 前置条件 |  |
| 后置条件 |  |
| QueryHotelServiceImpl. getHotelDetails | 语法 | public HotelVO getHotelDetails(String address) |
| 前置条件 |  |
| 后置条件 |  |
| QueryHotelServiceImpl.getOrders | 语法 | public ArrayList<OrderVO> getOrders(String address, String userID) |
| 前置条件 |  |
| 后置条件 |  |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderInfo.getOrderList(String userID, String address) | 得到某用户在某酒店所有订单的列表 | |
| OrderInfo.getReservedOrderList(String userID) | 得到某用户的所有已执行、已撤销和异常订单 | |
| QueryHotelList.getHotelBriefInfoListByQuerying(String[] condition, ArrayList<BriefOrderInfoPO> orderedHotelList) | 获得符合条件的酒店列表 | |
| HotelDAO.getHotelDetails(String address) | 获取酒店详情 | |

**QueryHotelList的接口规范**

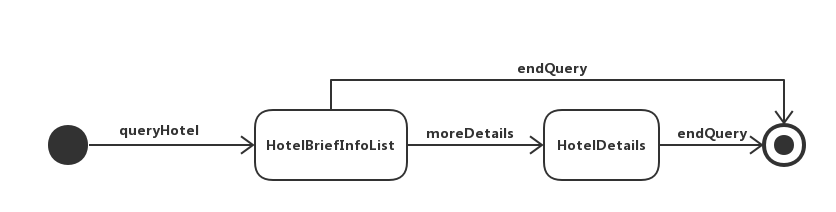
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| QueryHotelList. getHotelBriefInfoListByQuerying | 语法 | public ArrayList<BriefHotelInfoPO> getHotelBriefInfoListByQuerying(String[] condition, ArrayList<BriefOrderInfoPO> orderedHotelList) |
| 前置条件 |  |
| 后置条件 |  |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| HotelDAO.getHotelBriefInfoListByQuerying(String[] condition, ArrayList<BriefOrderInfoPO> orderedHotelList) | 通过查看获取酒店基本信息列表 | |

**业务逻辑层的动态模型**

**QueryHotel的顺序图如下图所示。**

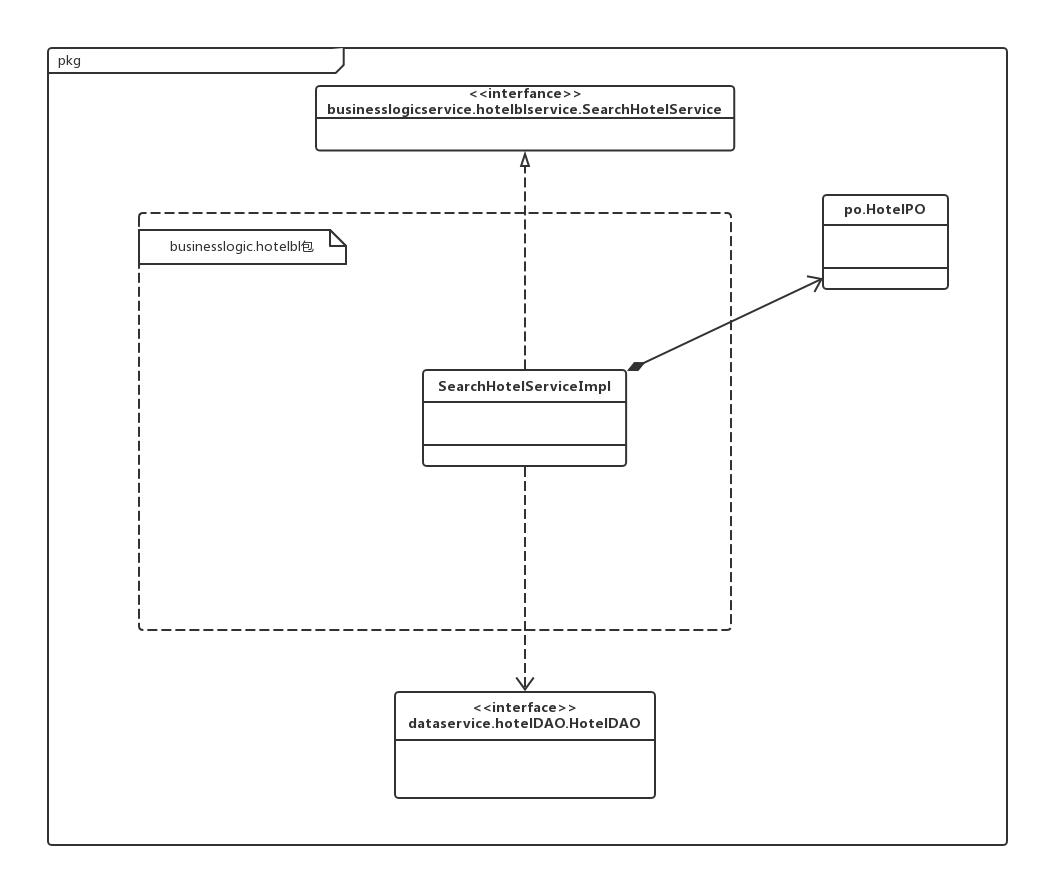
****

**QueryHotel的状态图如下图所示。**

****

* + - 1. **SearchHotel**

**SearchHotel的设计如下图所示。**

**接口规范：**

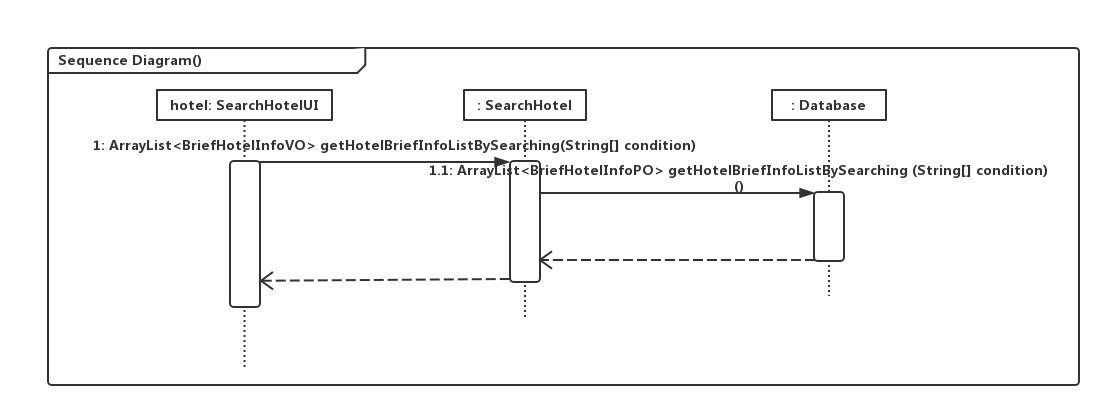
SearchHotelServiceImpl的接口规范如下表所示

**SearchHotelServiceImpl的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SearchHotelServiceImpl.getHotelBriefInfoListBySearching | 语法 | public ArrayList<OrderedHotelInfoVO> getHotelBriefInfoListByQuerying(String[] condition) |
| 前置条件 |  |
| 后置条件 |  |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| HotelDAO.getHotelBriefInfoListBySearching(String[] condition) | 搜索符合输入条件的酒店 | |

**业务逻辑层的动态模型**

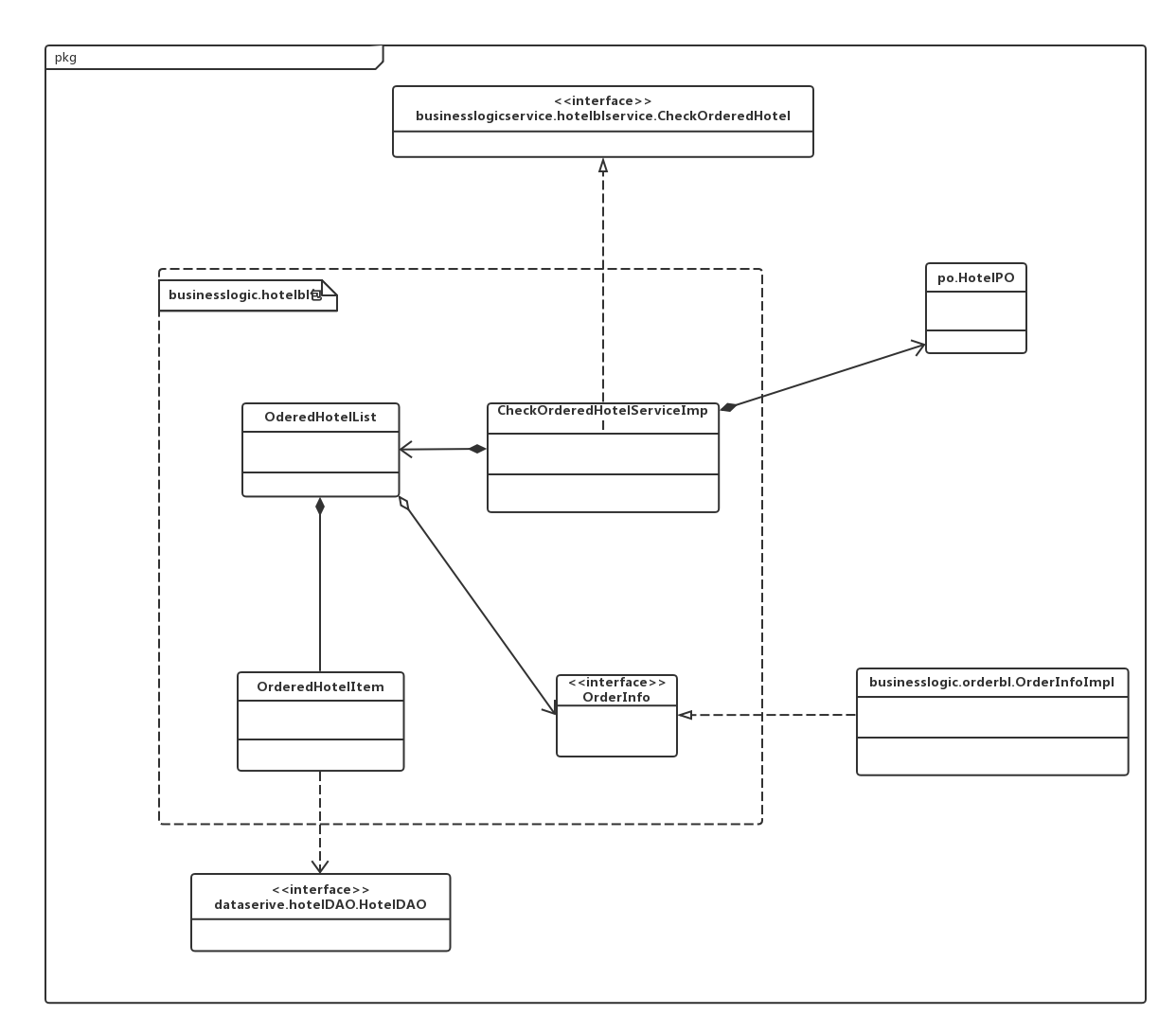
**SearchHotel的顺序图如下图所示。**

**SearchHotel的状态图如下图所示。**

****

* + - 1. **CheckOrderedHotel**

**CheckOrderedHotel的设计如下图所示。**

**接口规范：**

CheckOrderedHotelServiceImpl、OrderedHotelList、OrderedHotelItem的接口规范如下表所示

**CheckOrderedHotelServiceImpl的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CheckOrderedHotelServiceImpl.enrollHotelBreifInfoList | 语法 | public ArrayList<OrderedHotelInfoVO> enrollHotelBreifInfoList(String userID) |
| 前置条件 |  |
| 后置条件 |  |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderedHotelList.enrollHotelBreifInfoList() | 获得预定过的酒店的简要信息列表 | |

**OrderedHotelList的接口规范**

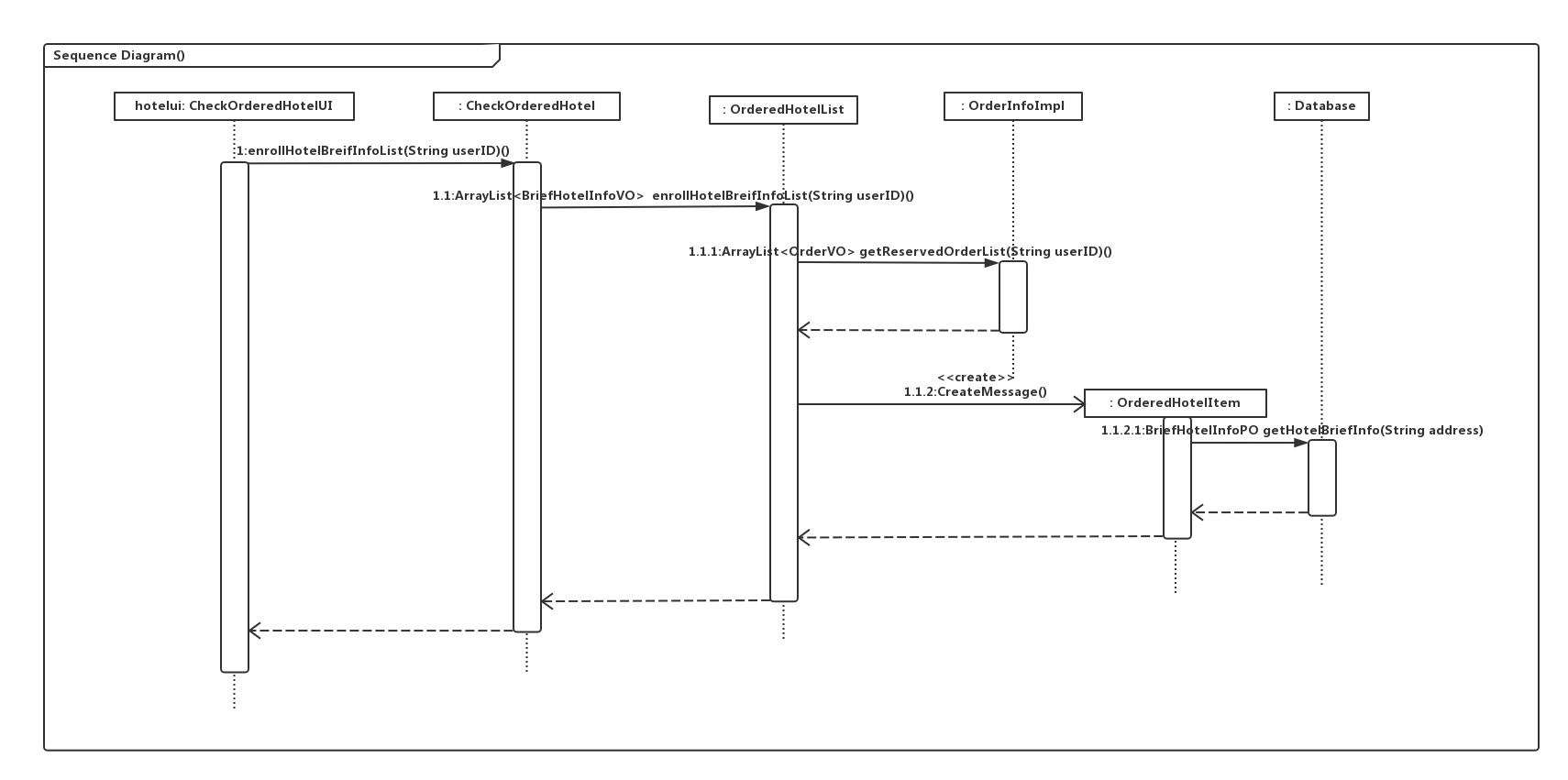
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderedHotelList. enrollHotelBreifInfoList | 语法 | public ArrayList<OrderedHotelInfoVO> enrollHotelBreifInfoList() |
| 前置条件 |  |
| 后置条件 |  |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderInfo.getReservedOrderList(String userID) | 得到某用户的所有已执行、已撤销和异常订单 | |
| OrderedHotelItem.getbriefHotelInfo() | 得到相应酒店的信息 | |

**OrderedHotelItem的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderedHotelItem. getbriefHotelInfo | 语法 | public BriefHotelInfoVO getbriefHotelInfo() |
| 前置条件 |  |
| 后置条件 |  |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| HotelDAO.getHotelBriefInfo(String address) | 获取酒店的基本信息 | |

**业务逻辑层的动态模型**

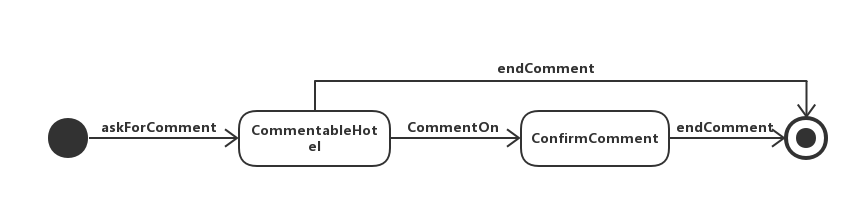
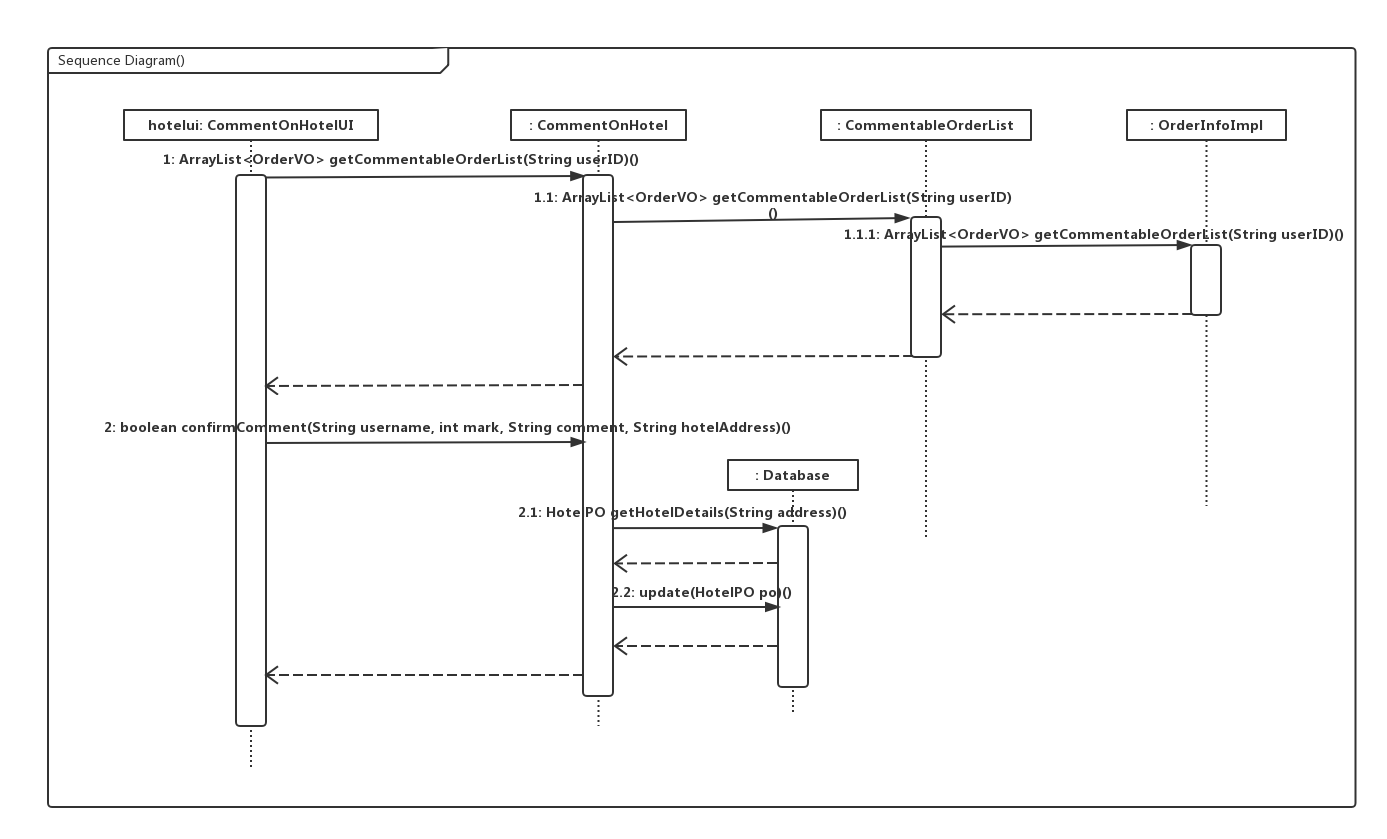
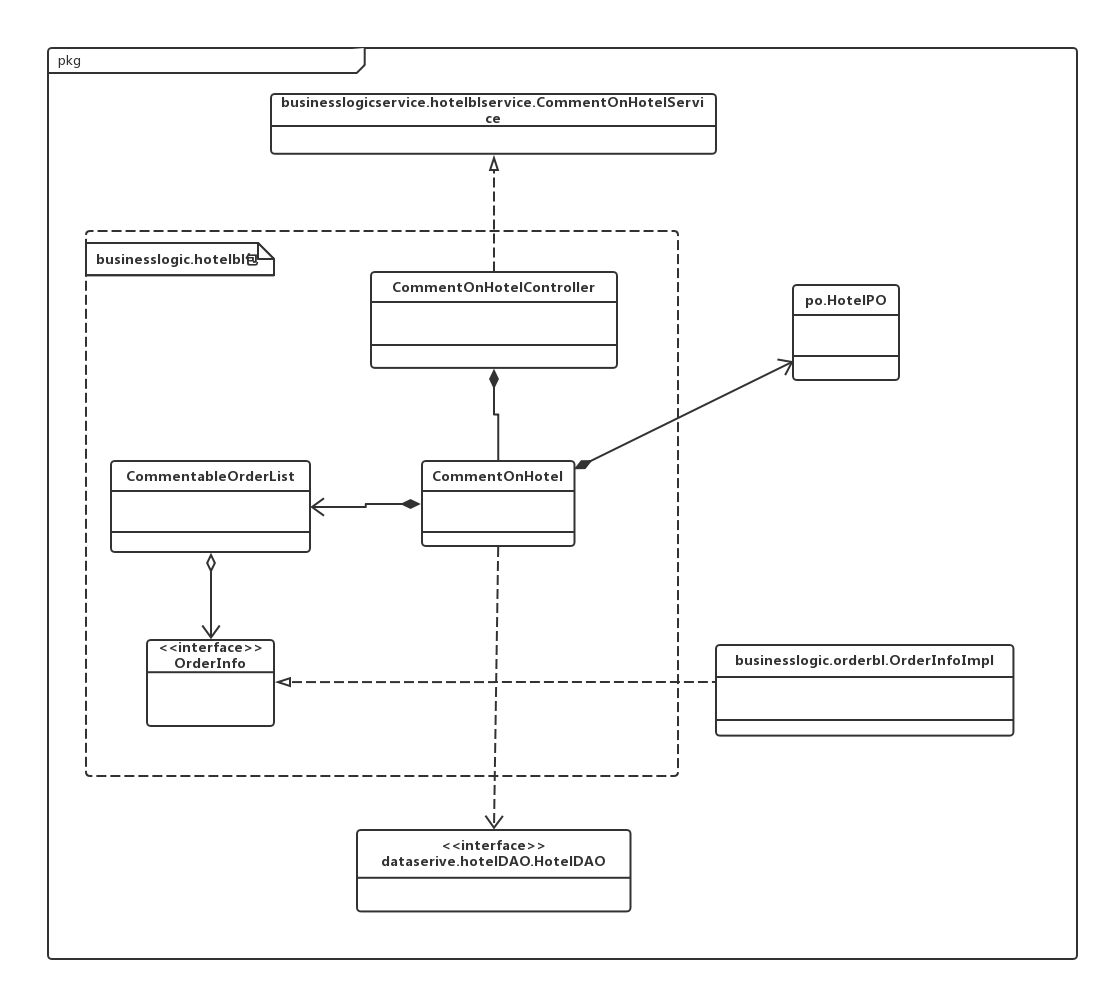
**CheckOrderedHotel的顺序图如下图所示。**

****

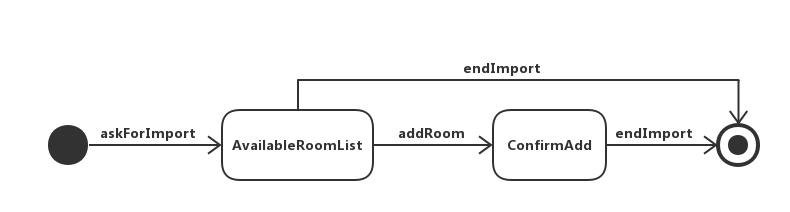
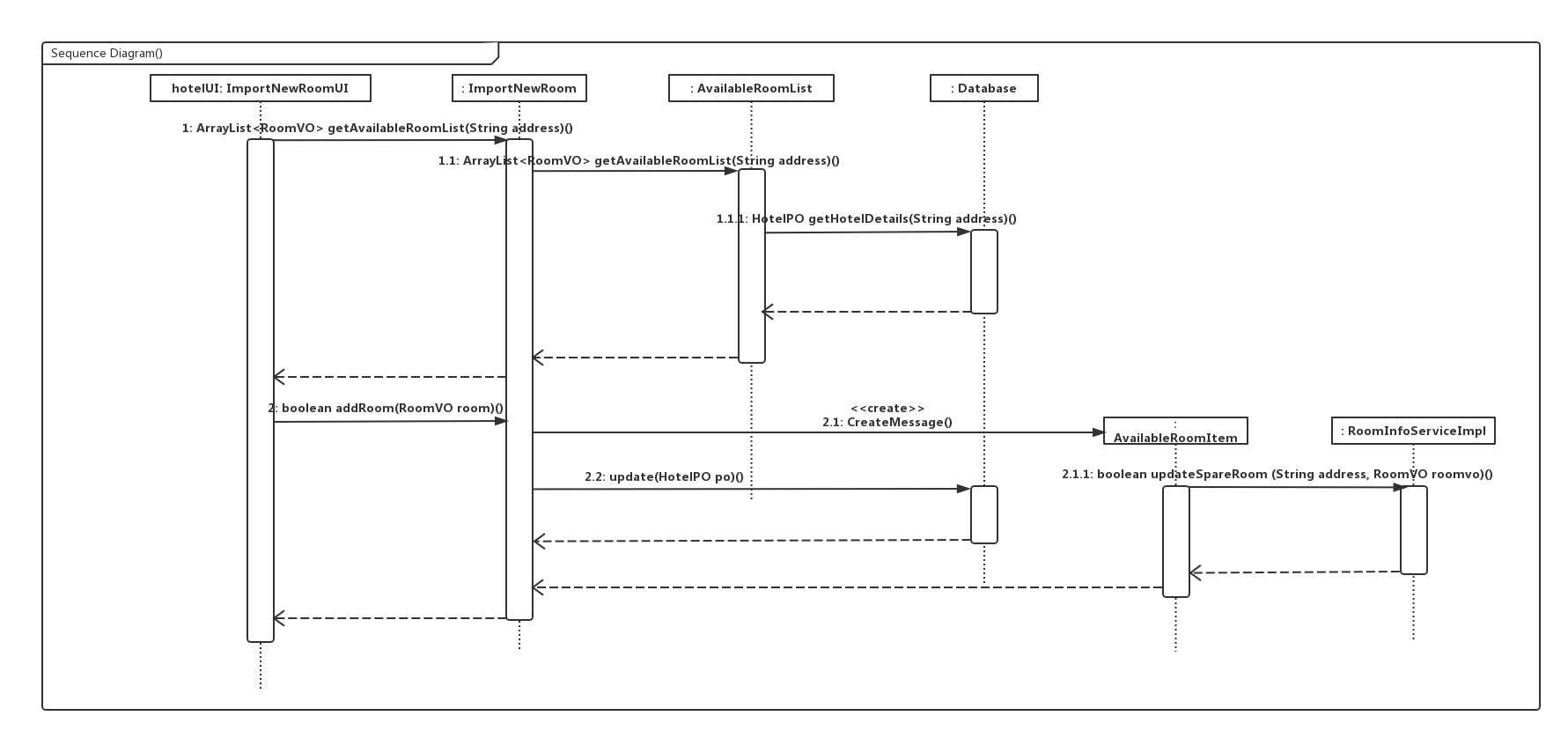
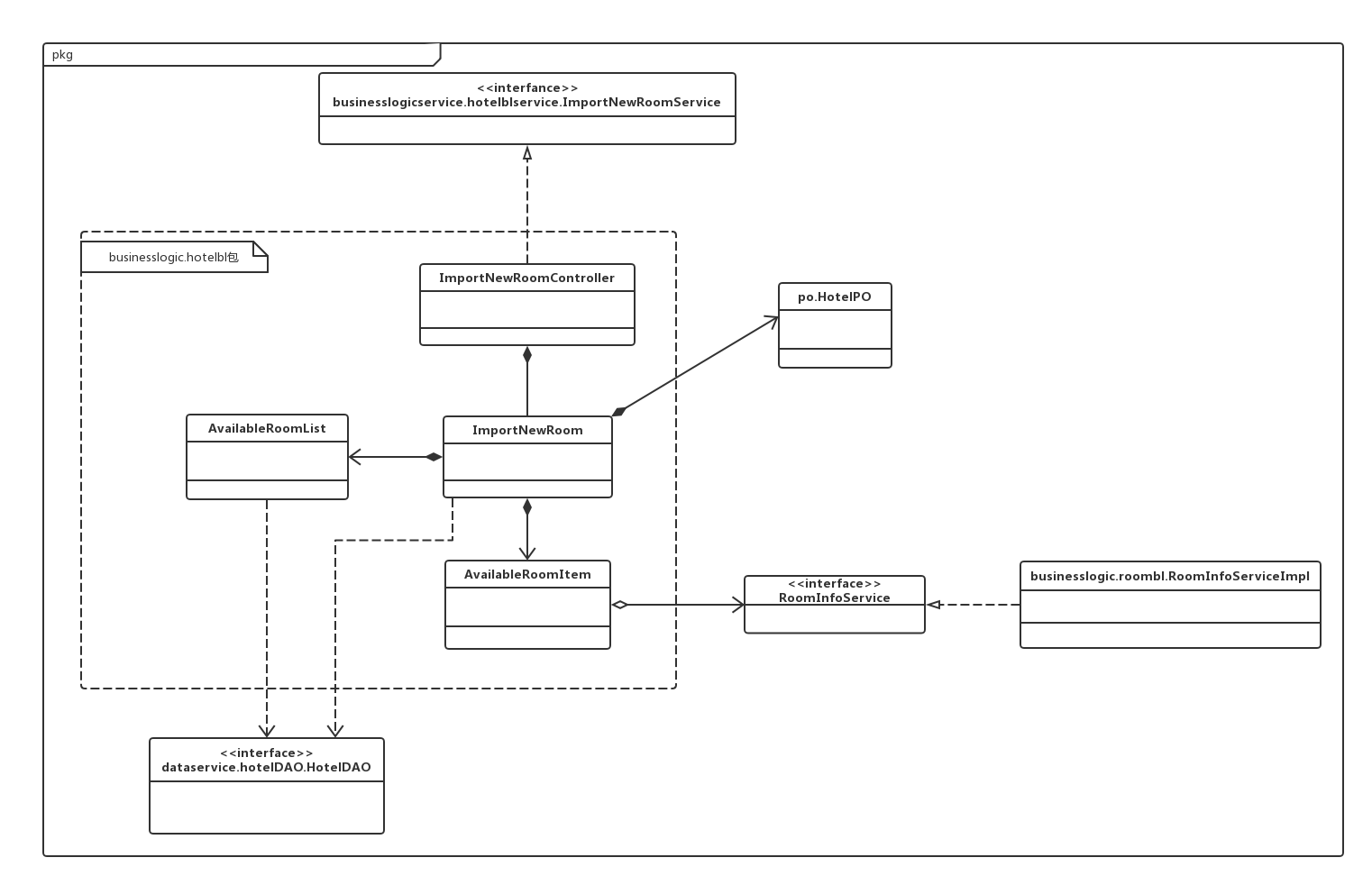
**CheckOrderedHotel的状态图如下图所示。**

****

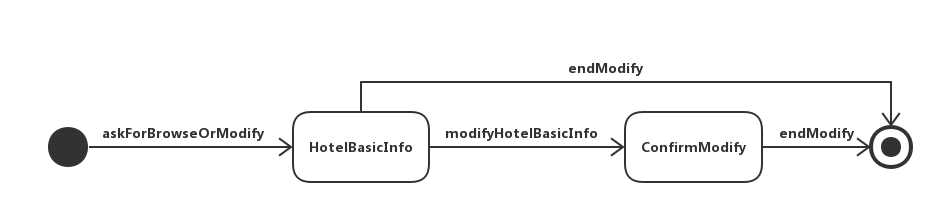
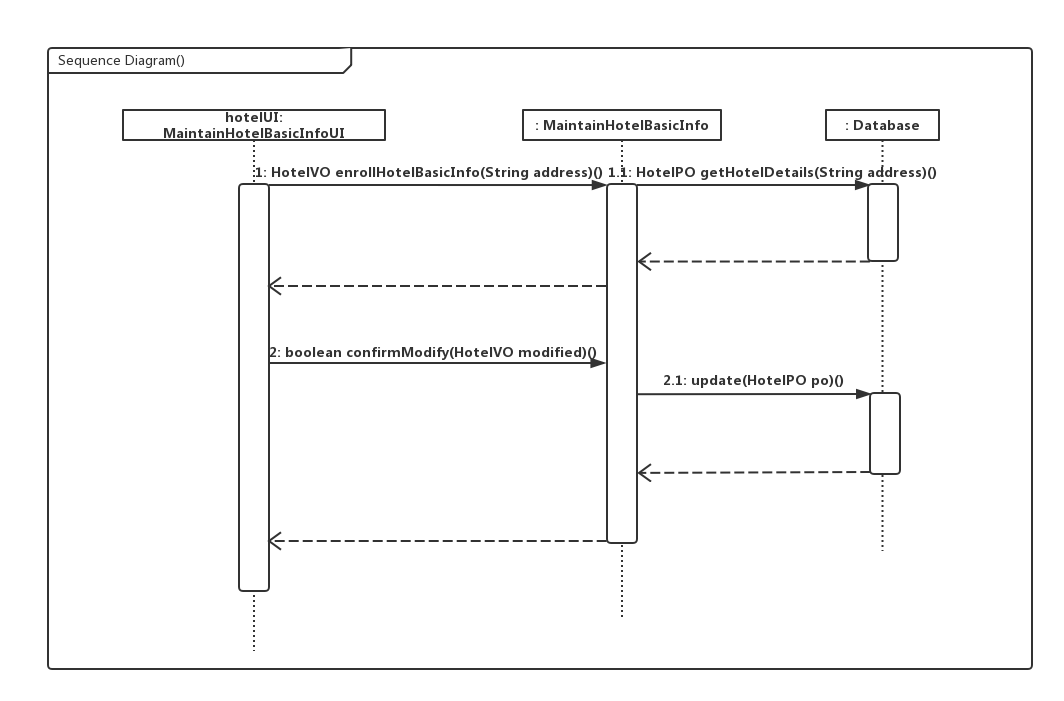
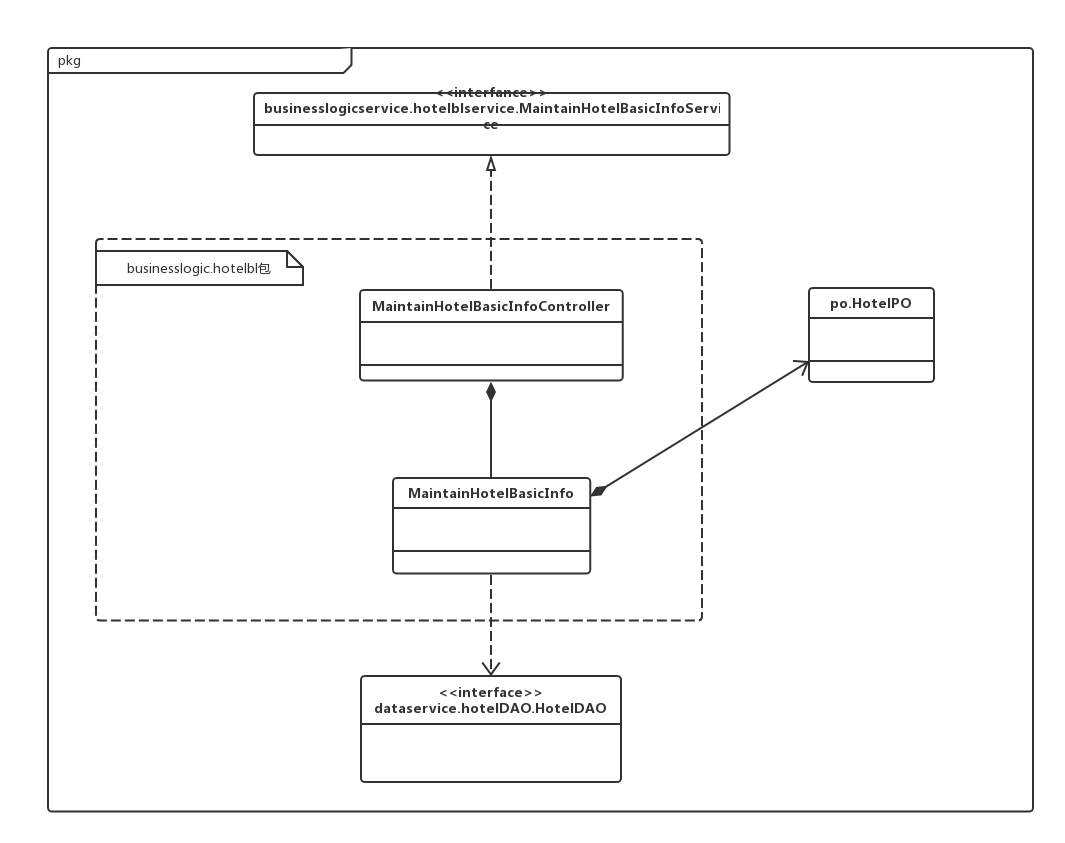
* + - 1. **CommentOnHotel**

****

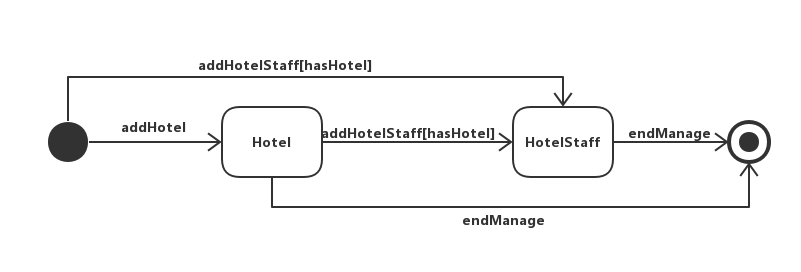
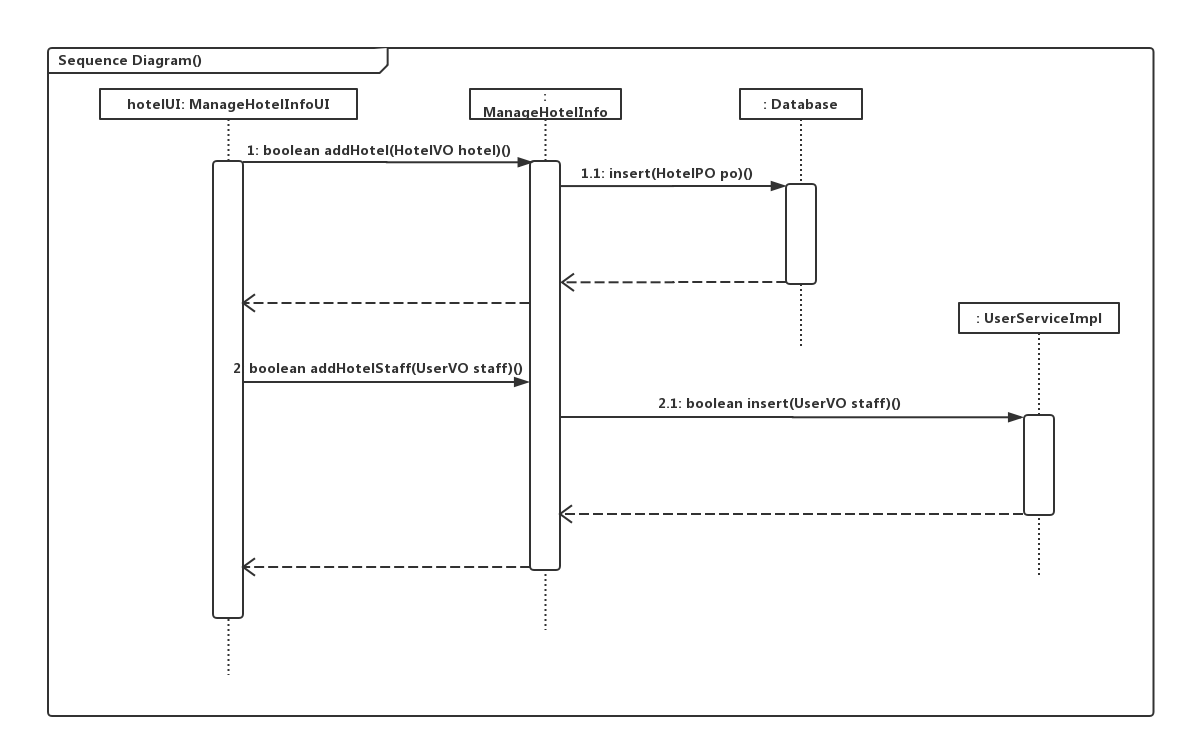
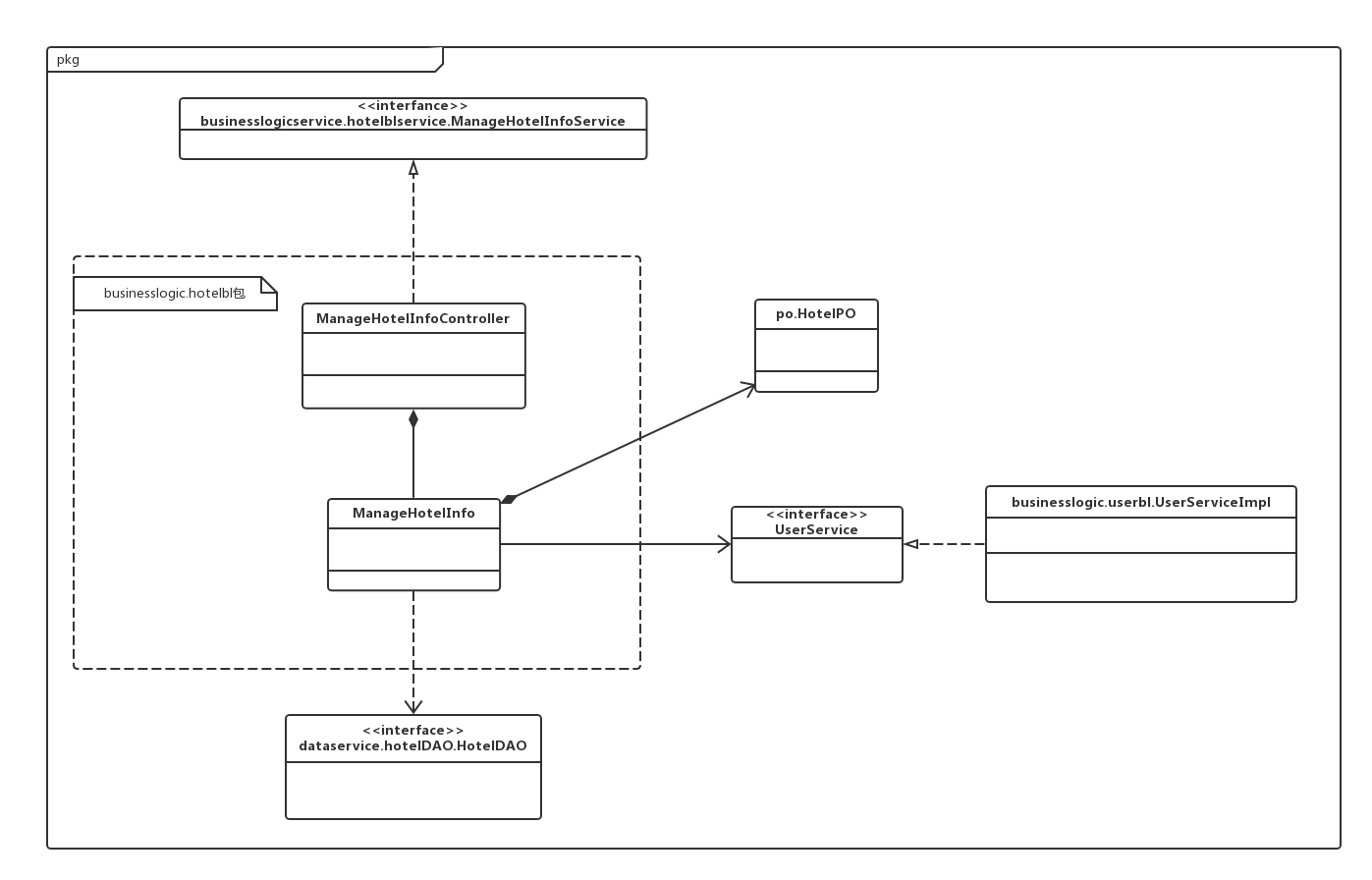
* + - 1. **ImportNewRoom**

****

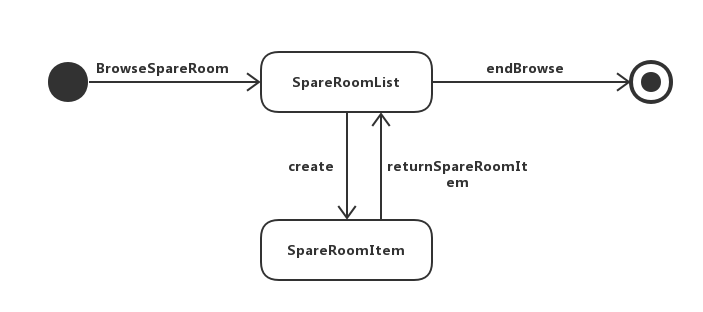
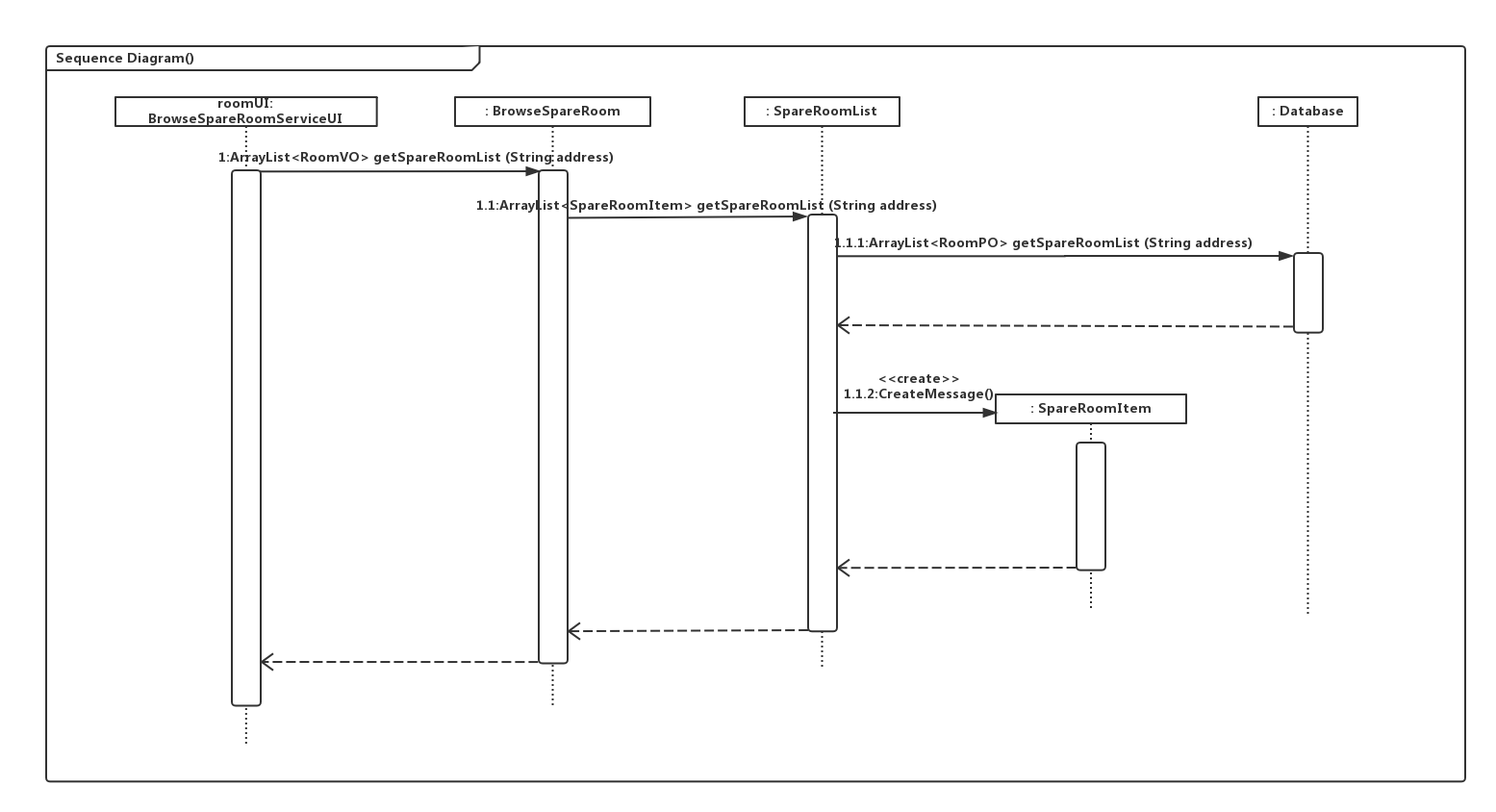
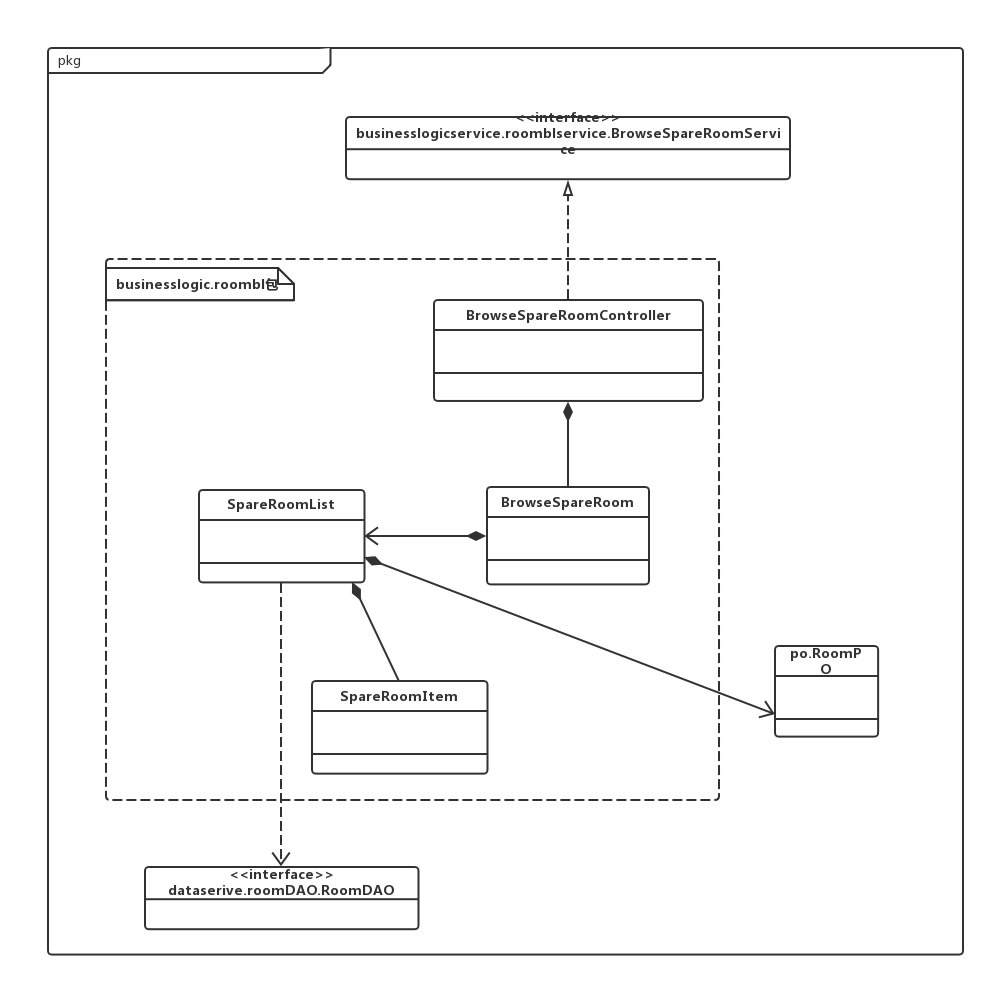
* + - 1. **MaintainHotelBasicInfo**



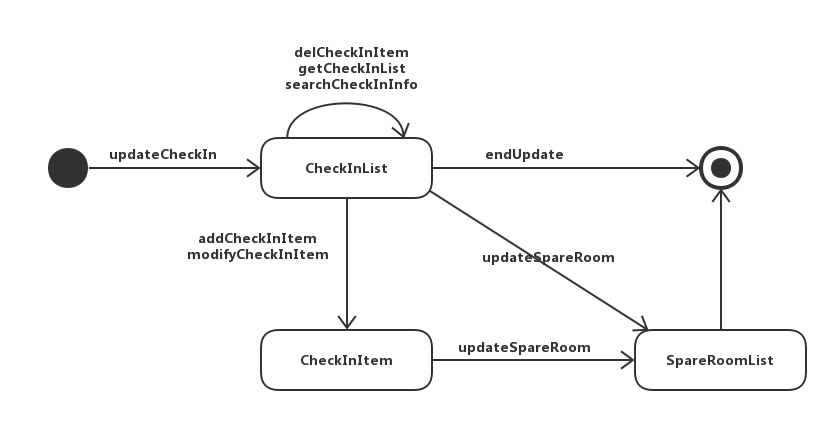
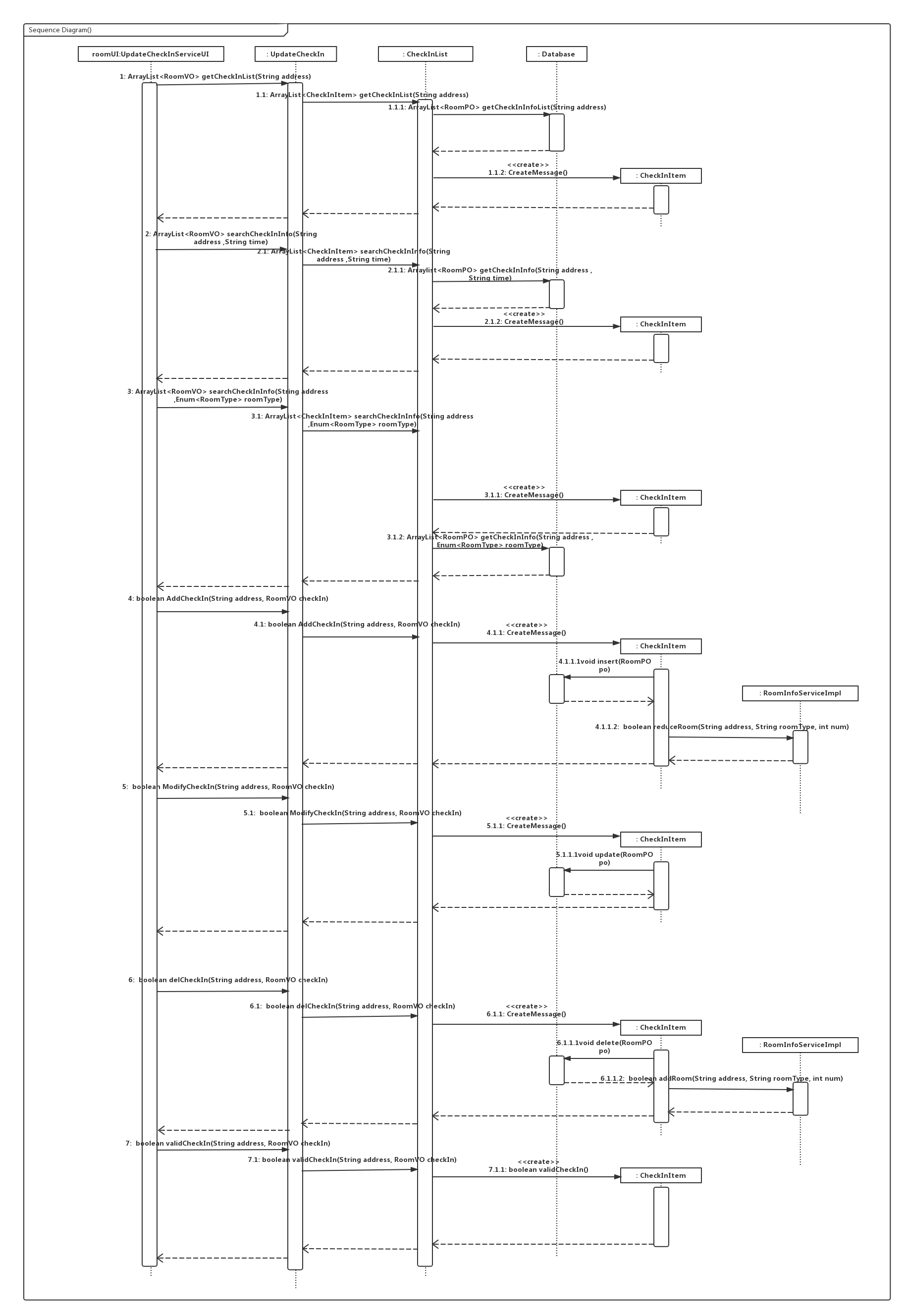
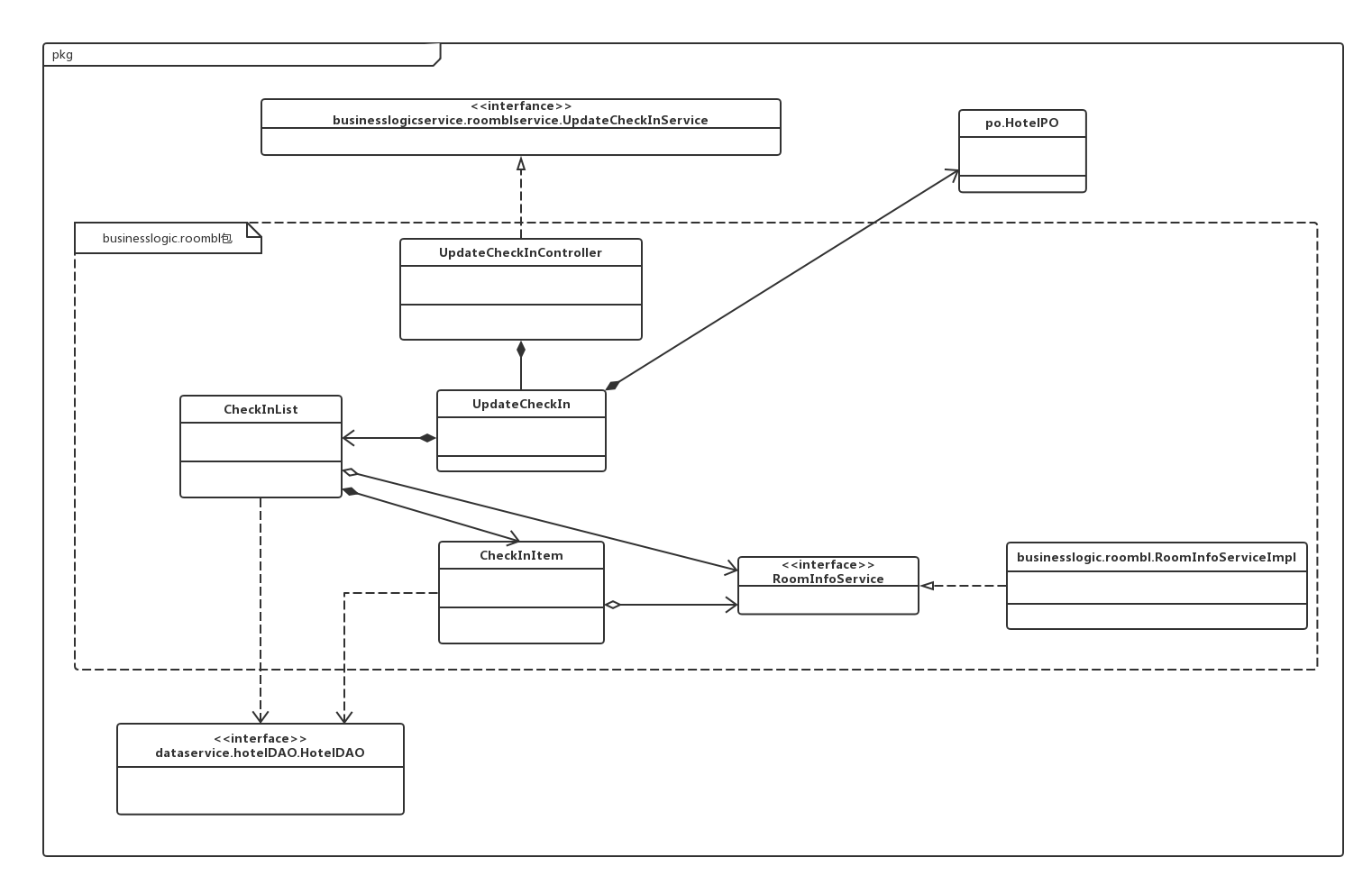
* + - 1. **ManageHotelInfo**



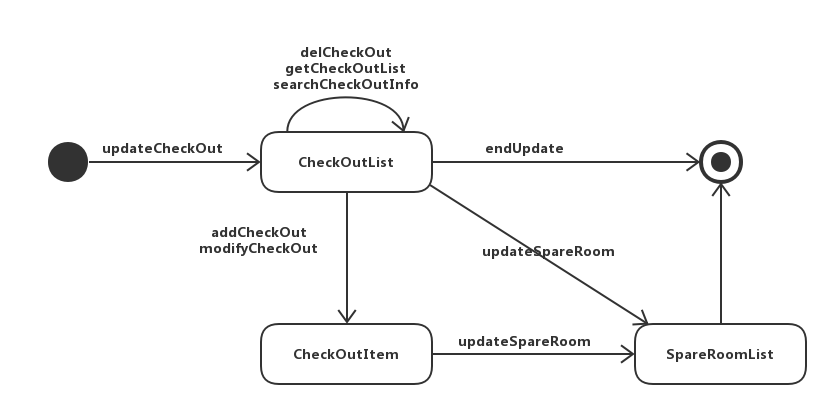
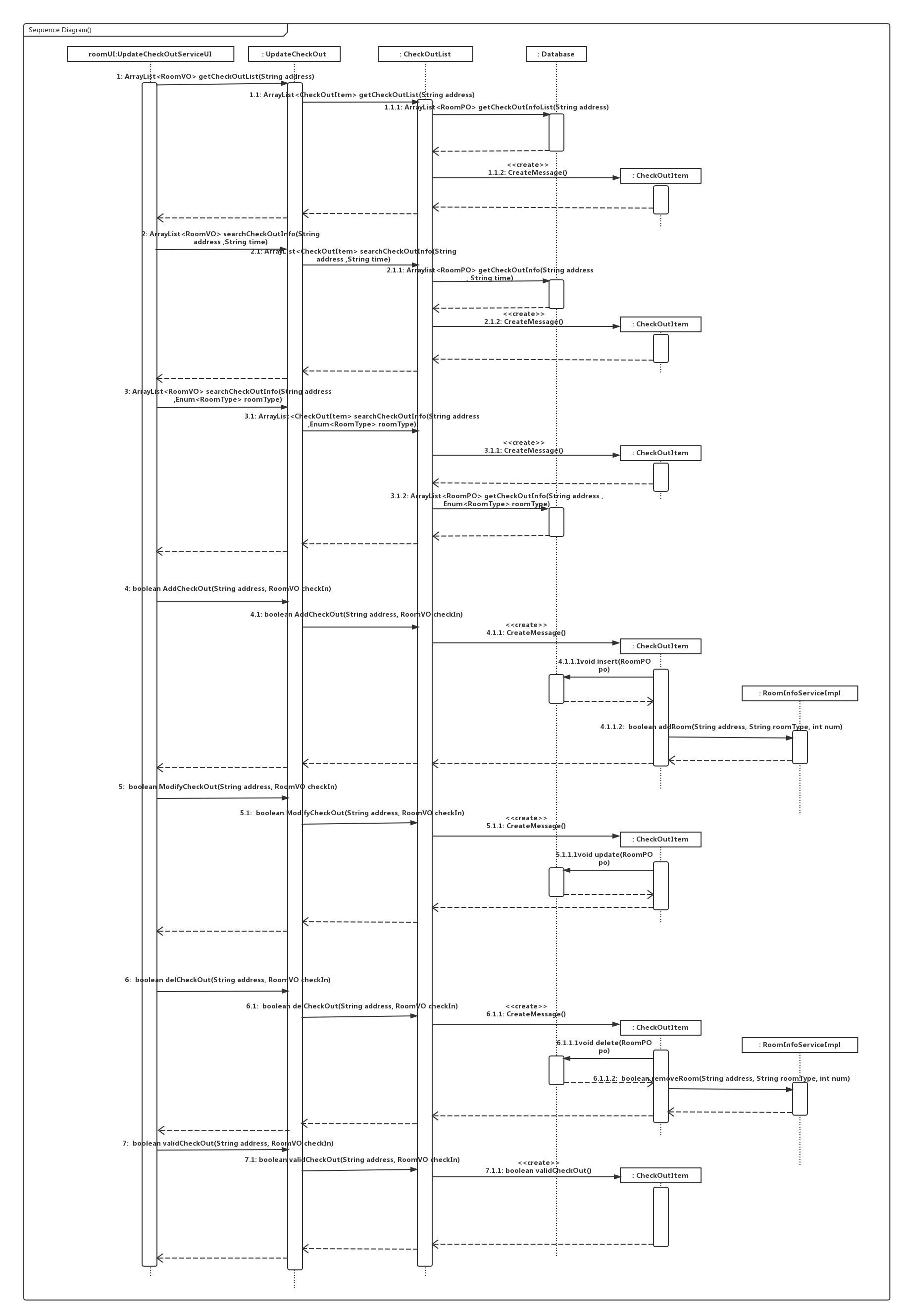
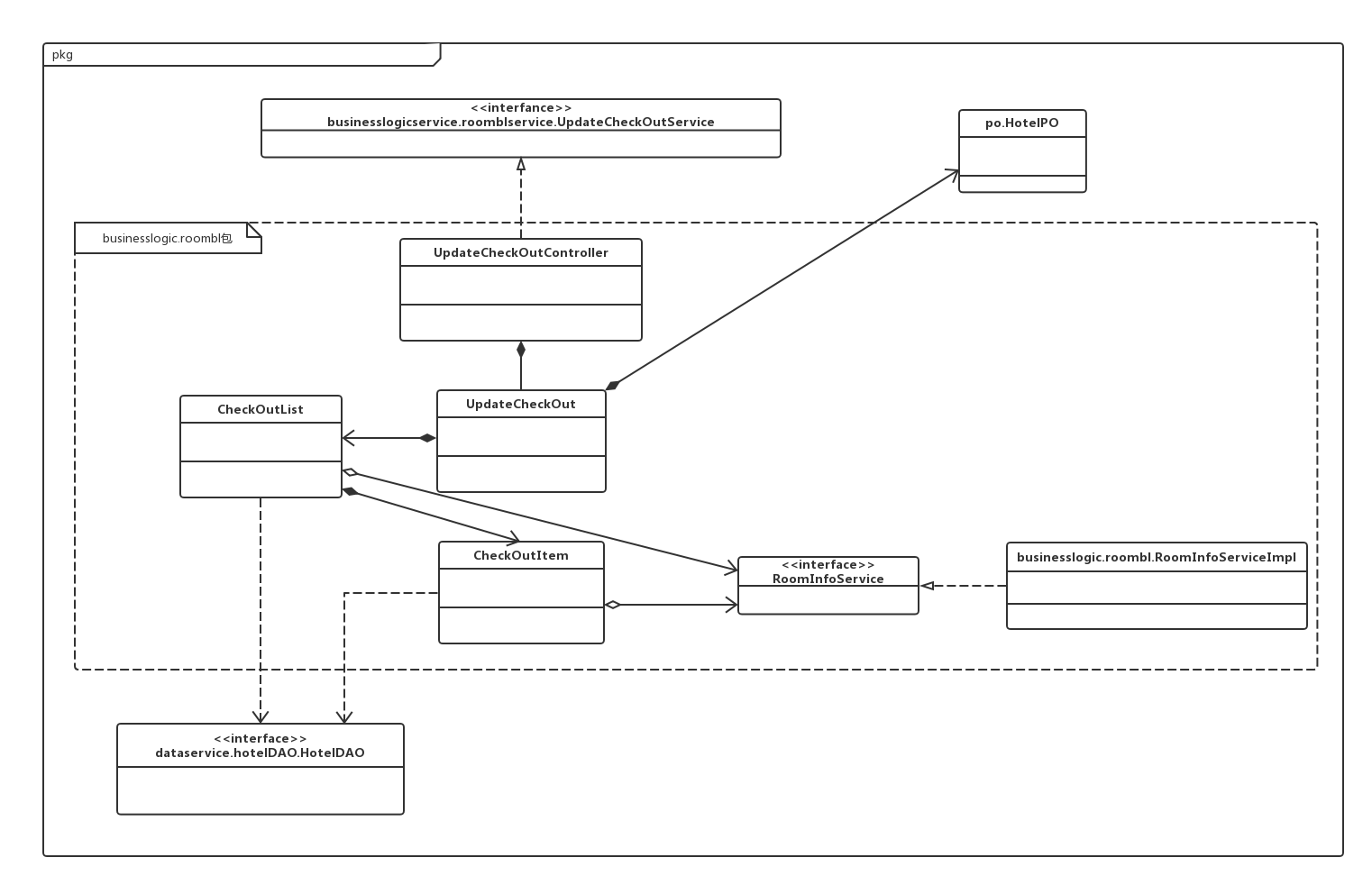
* + 1. **roombl模块**
       1. **BrowseSpareRoom**



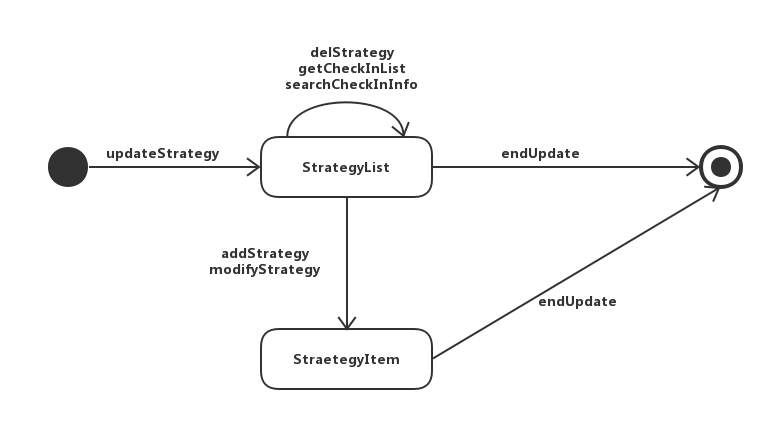
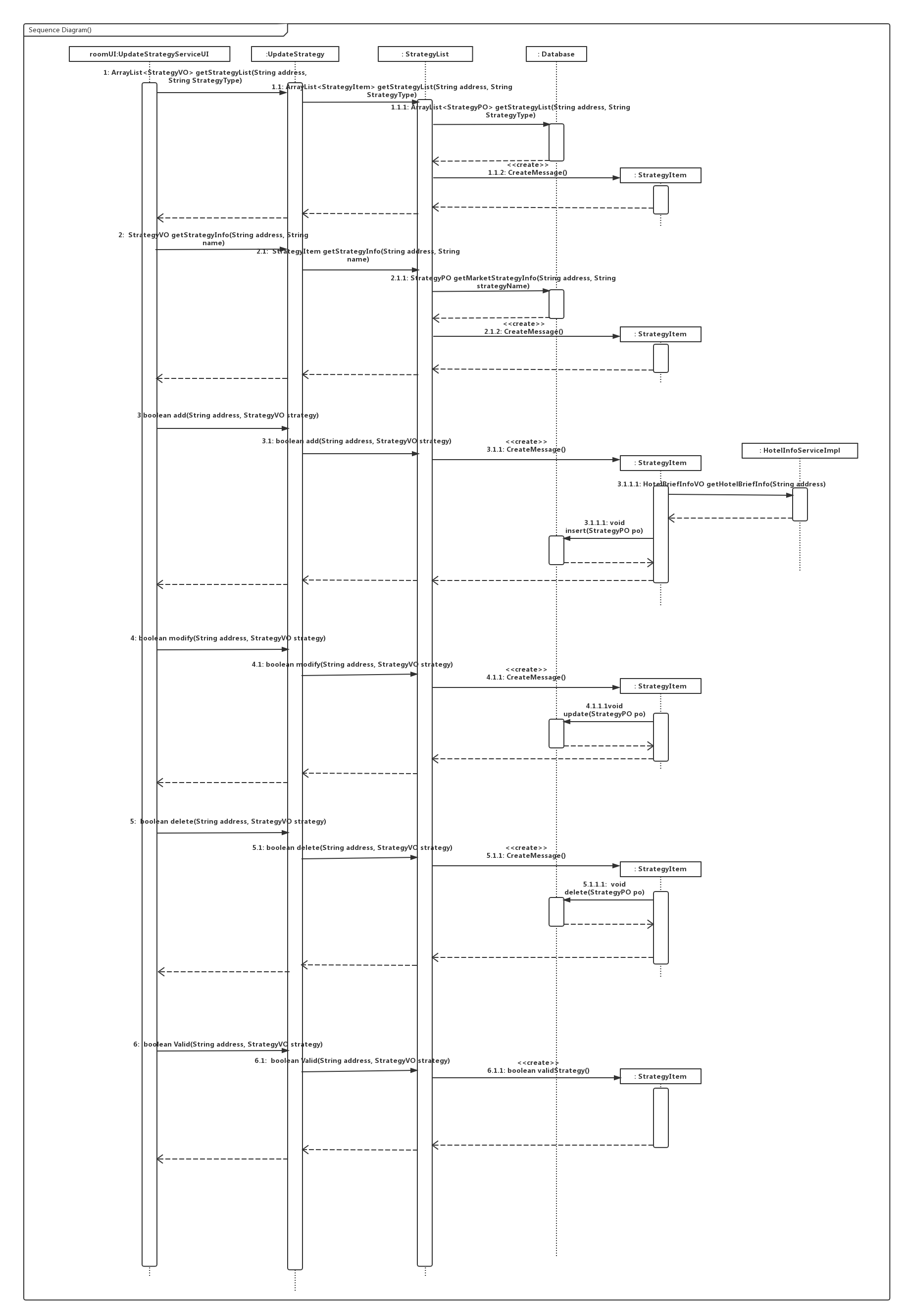
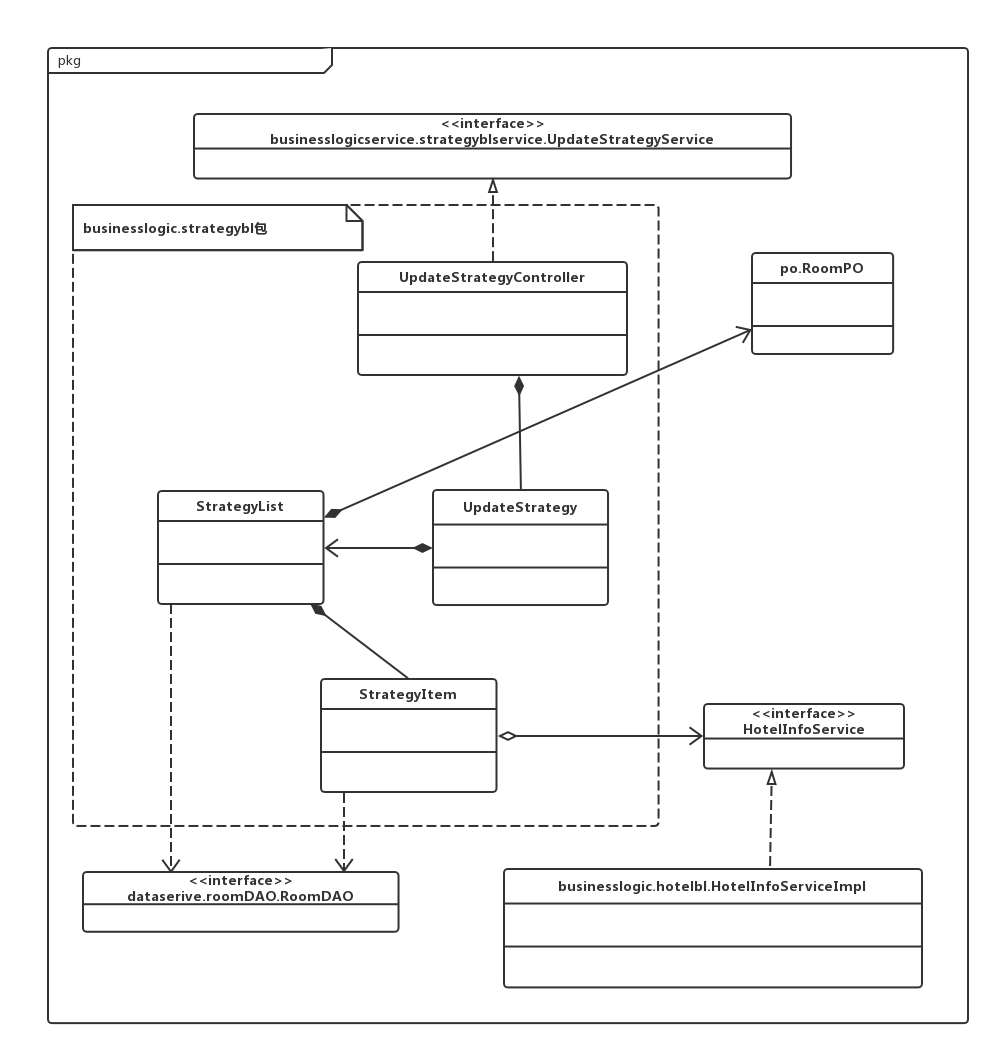
* + - 1. **UpdateCheckIn**



* + - 1. **UpdateCheckOut**



* + 1. **strategybl模块**
       1. **UpdateStrategy**



1. **依赖视角**

下图是客户端和服务器端各自的包之间的依赖关系。

