

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** |
| **曾虎双** | **2016.11.04** | **修改roombl和strategybl的设计类图和顺序图** | **V0.1** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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。

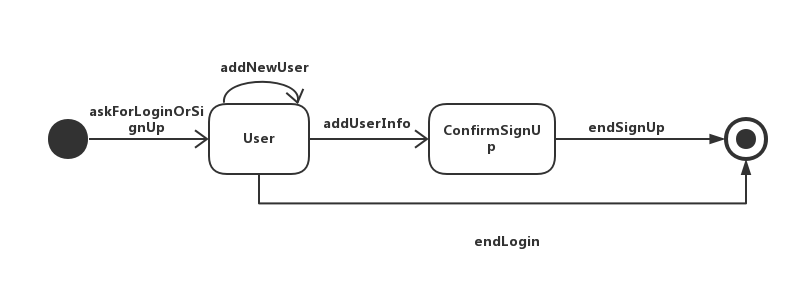
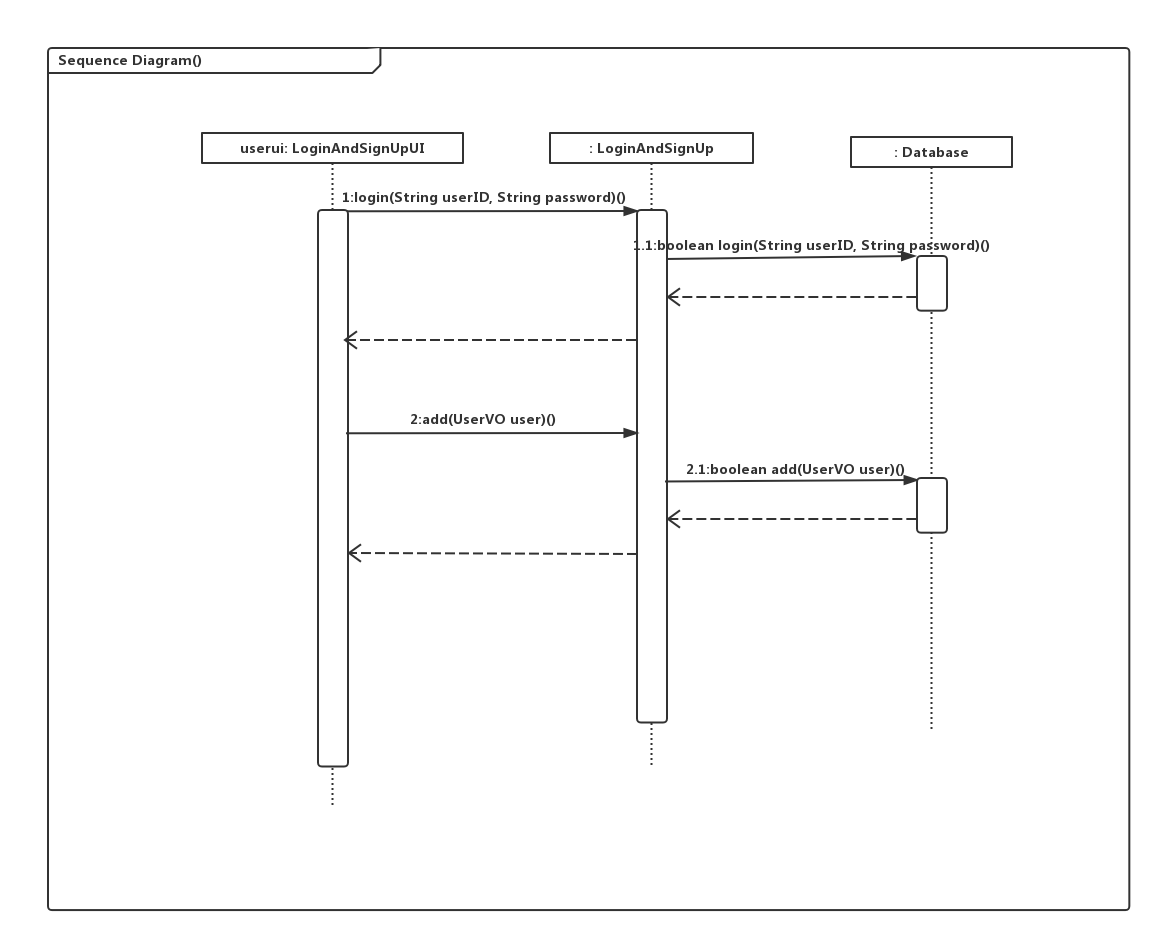
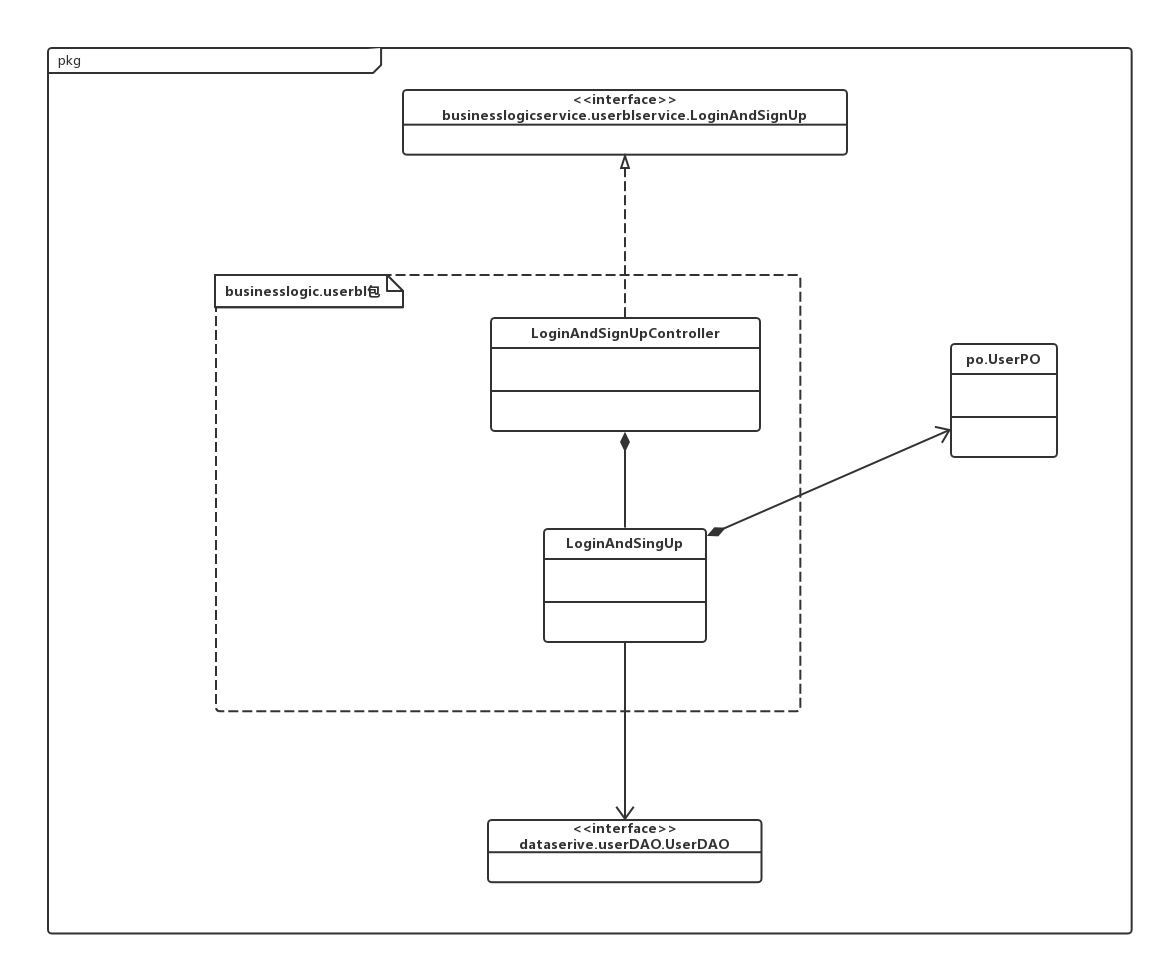
1. **产品概述**

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。

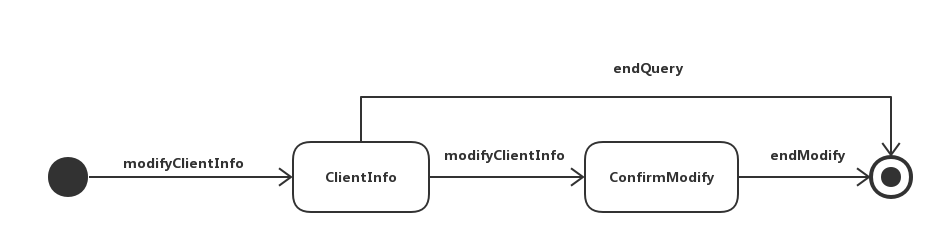
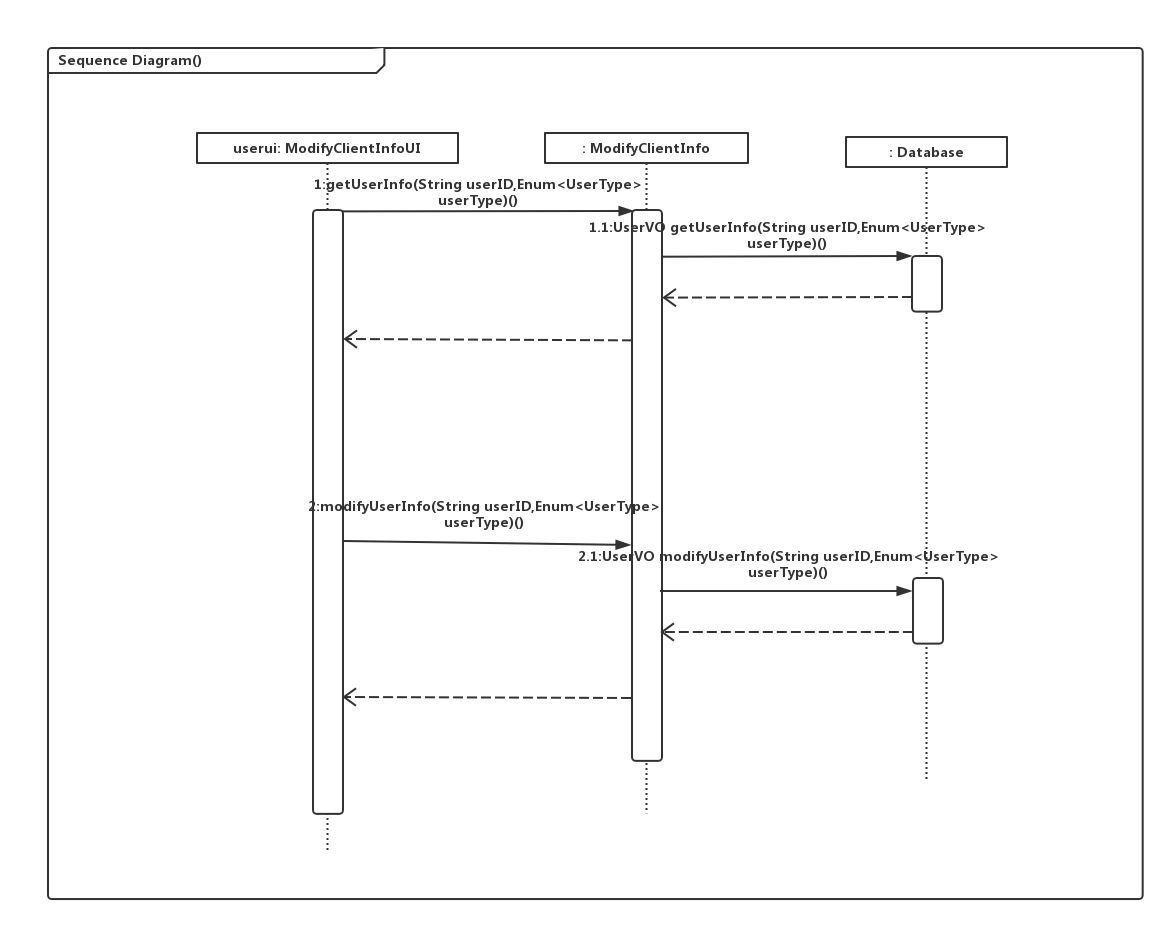
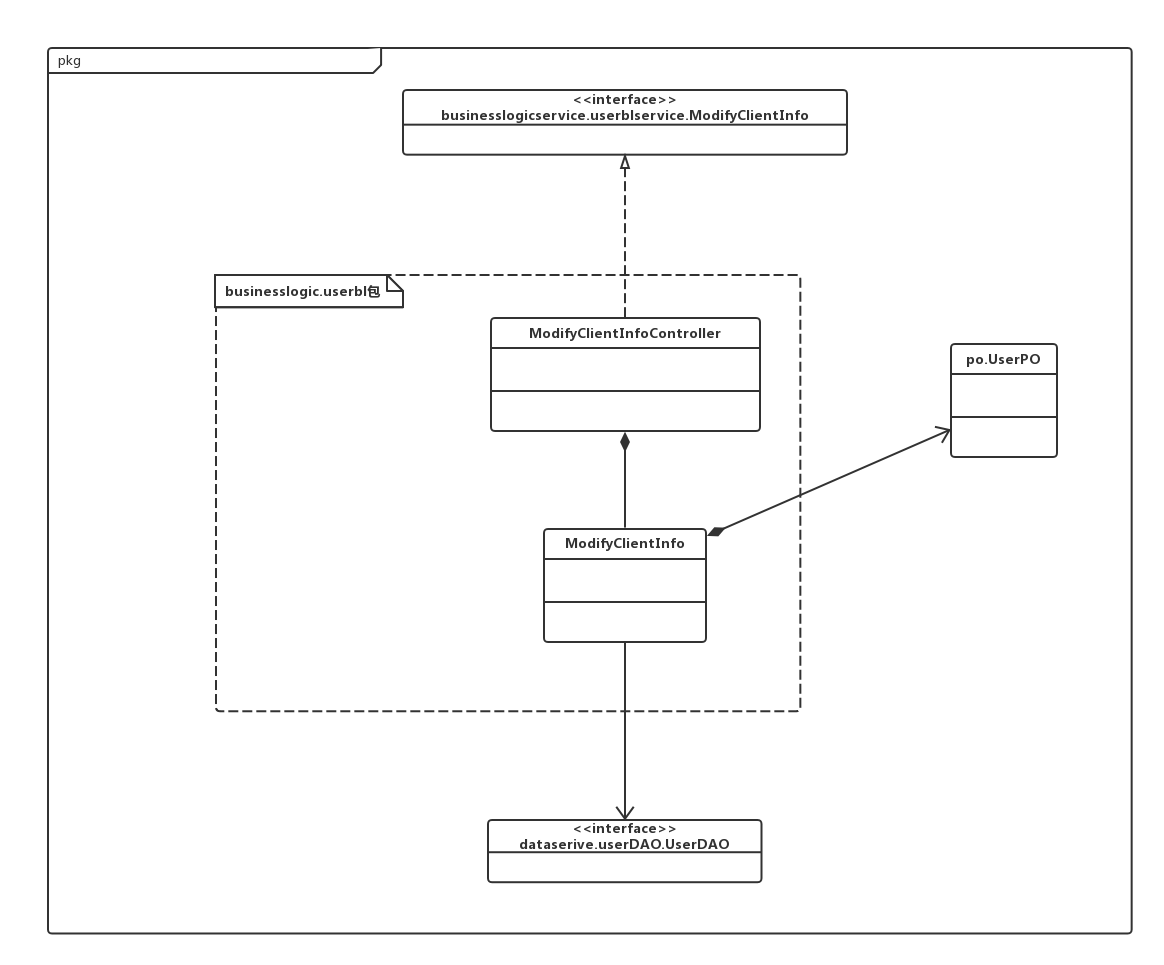
1. **体系结构设计概述**

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。

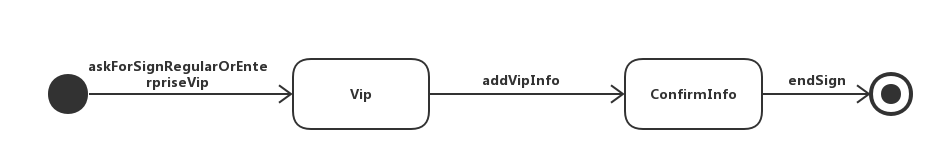
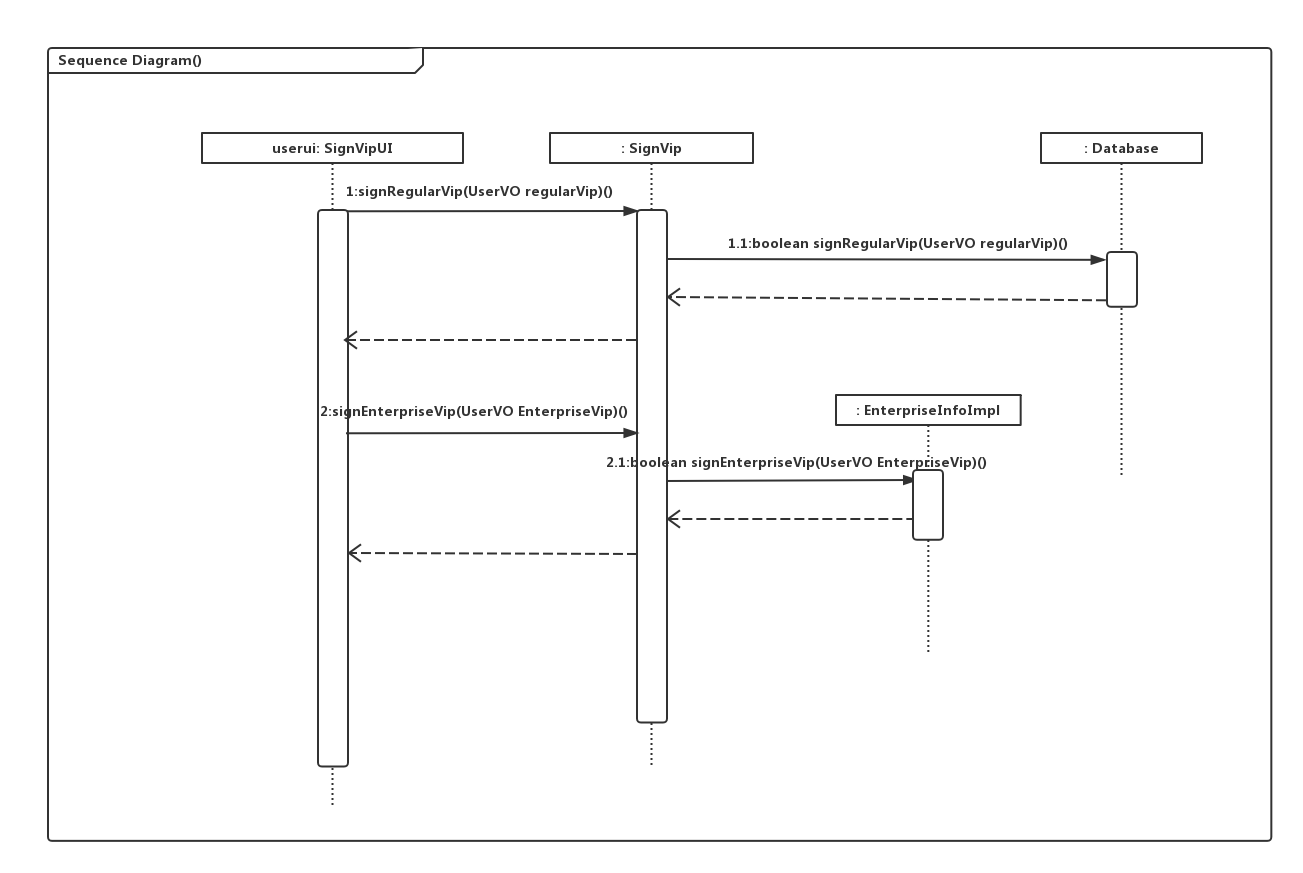
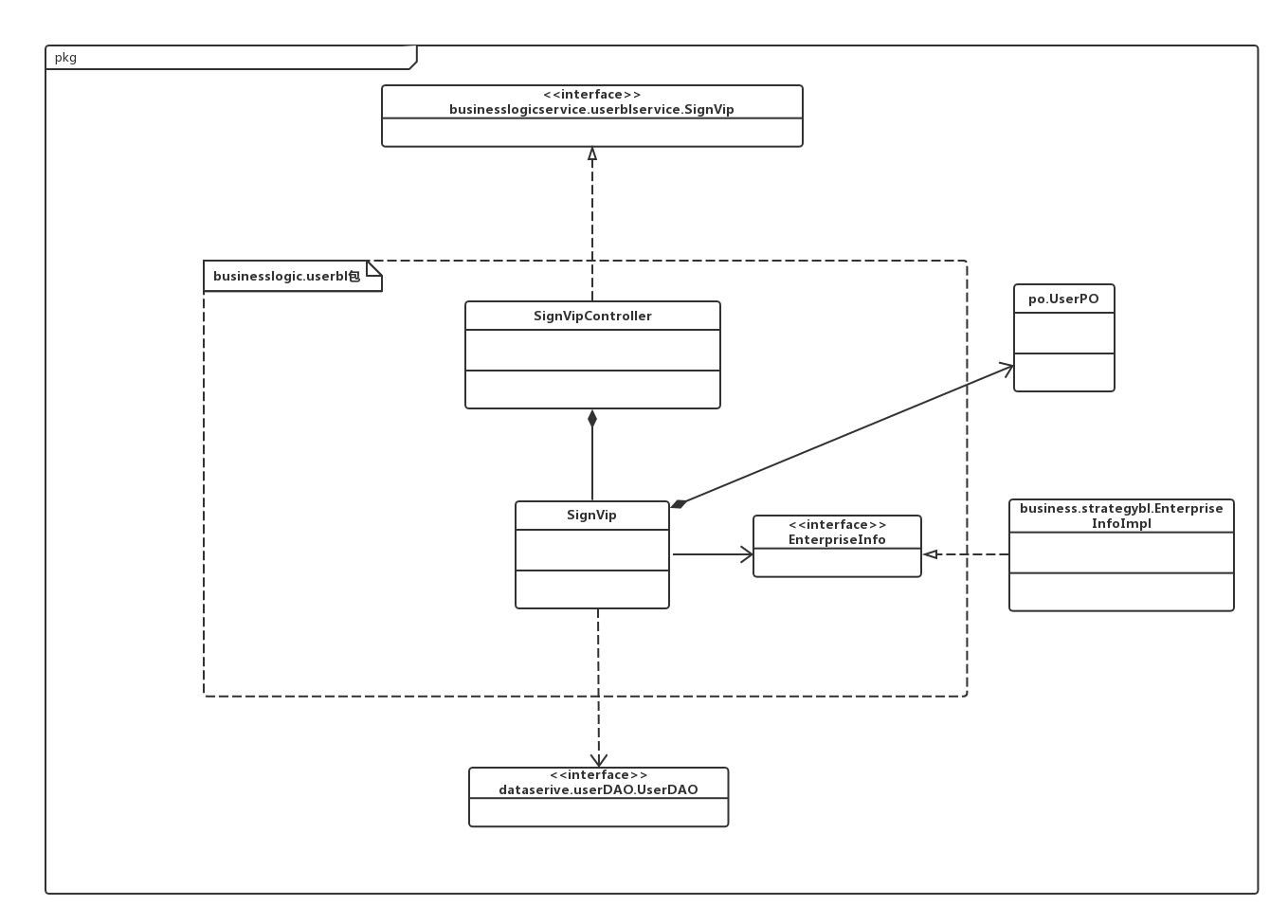
1. **结构视角**
   1. **业务逻辑层的分解**
      1. **userbl模块**
         1. **LoginAndSignUp**

****

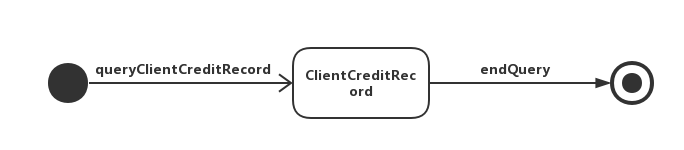
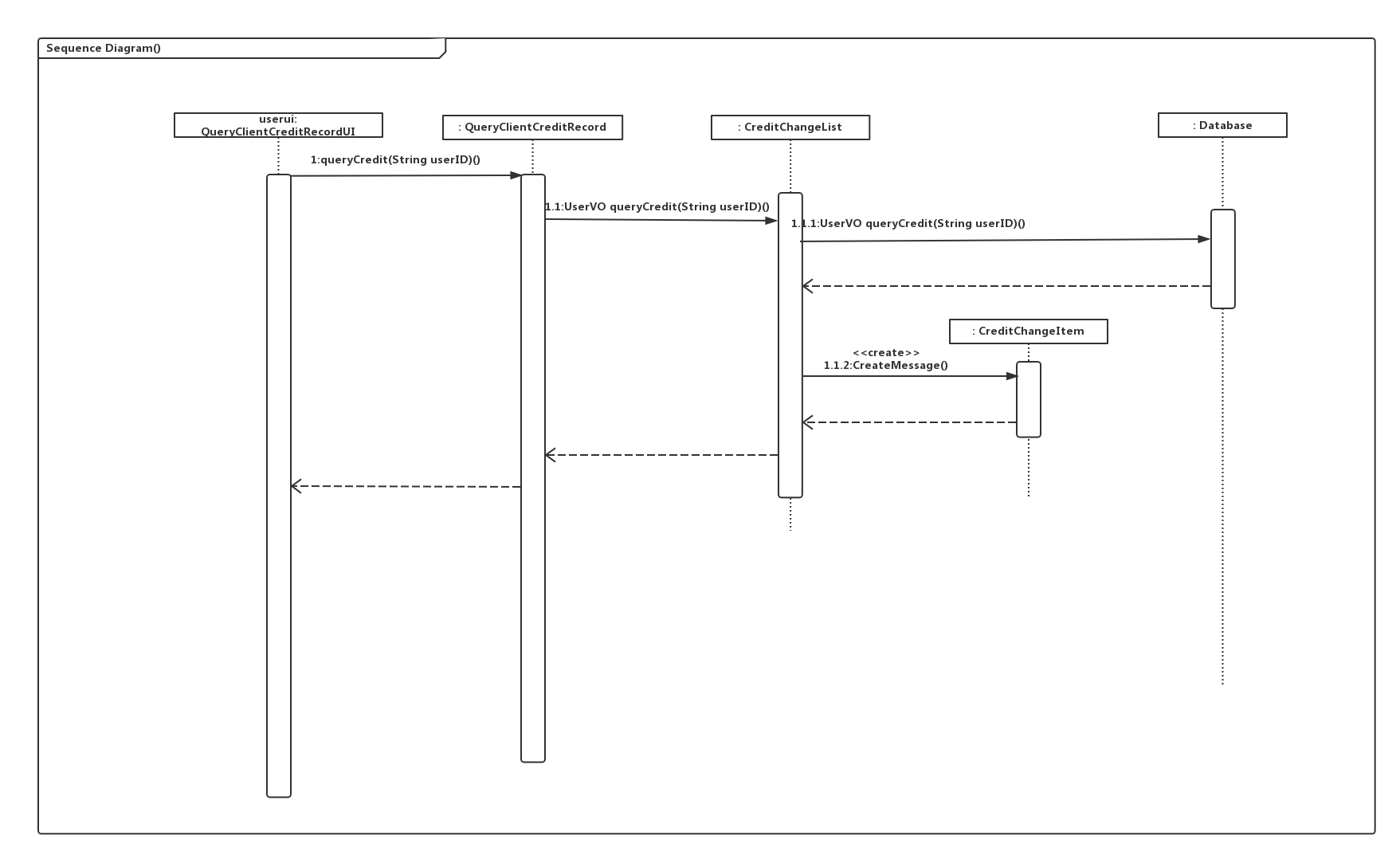
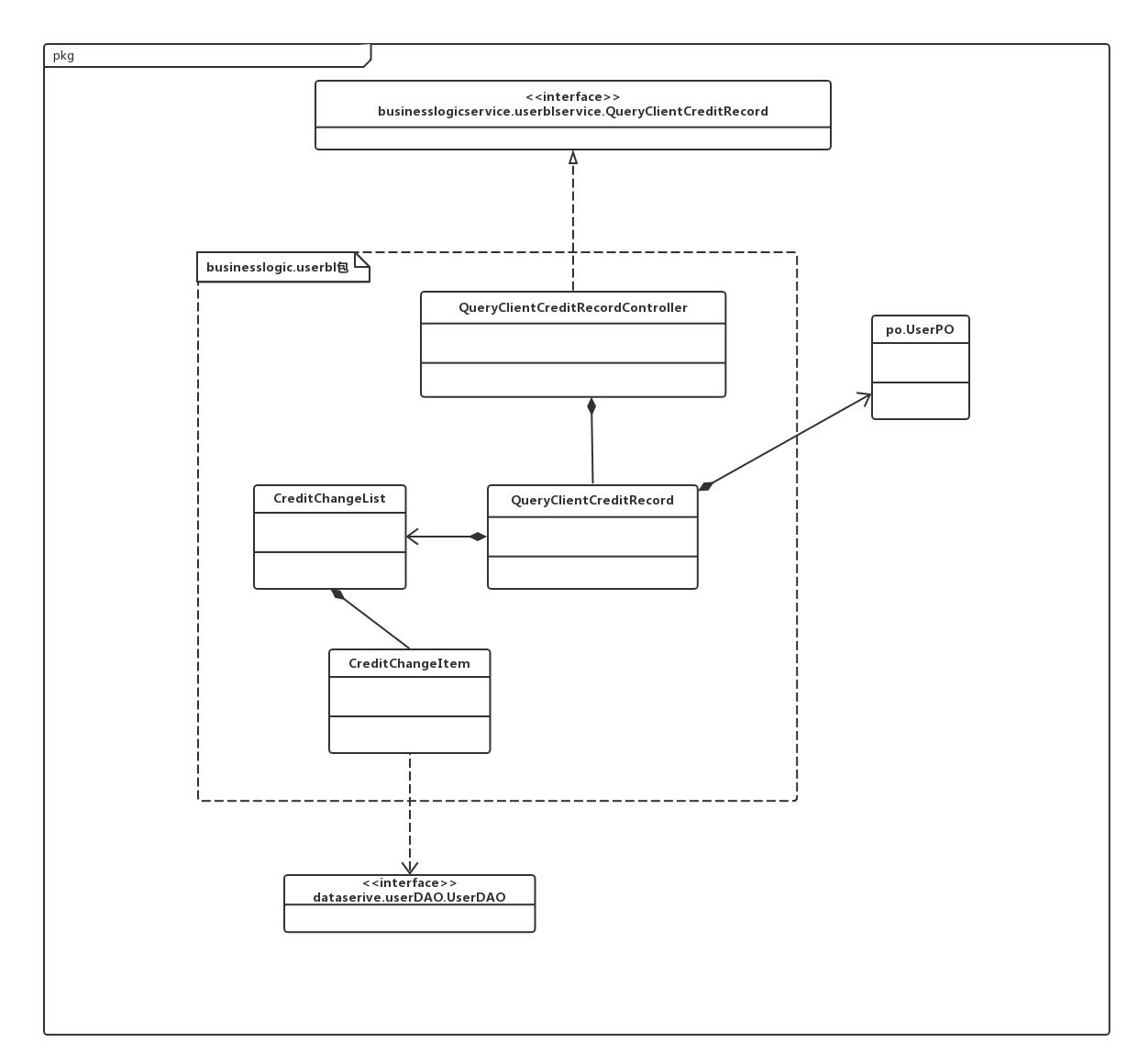
* + - 1. **ModifyClientInfo**

****

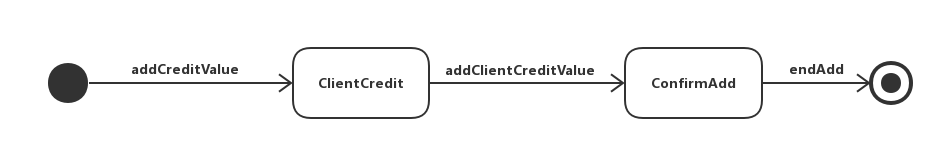
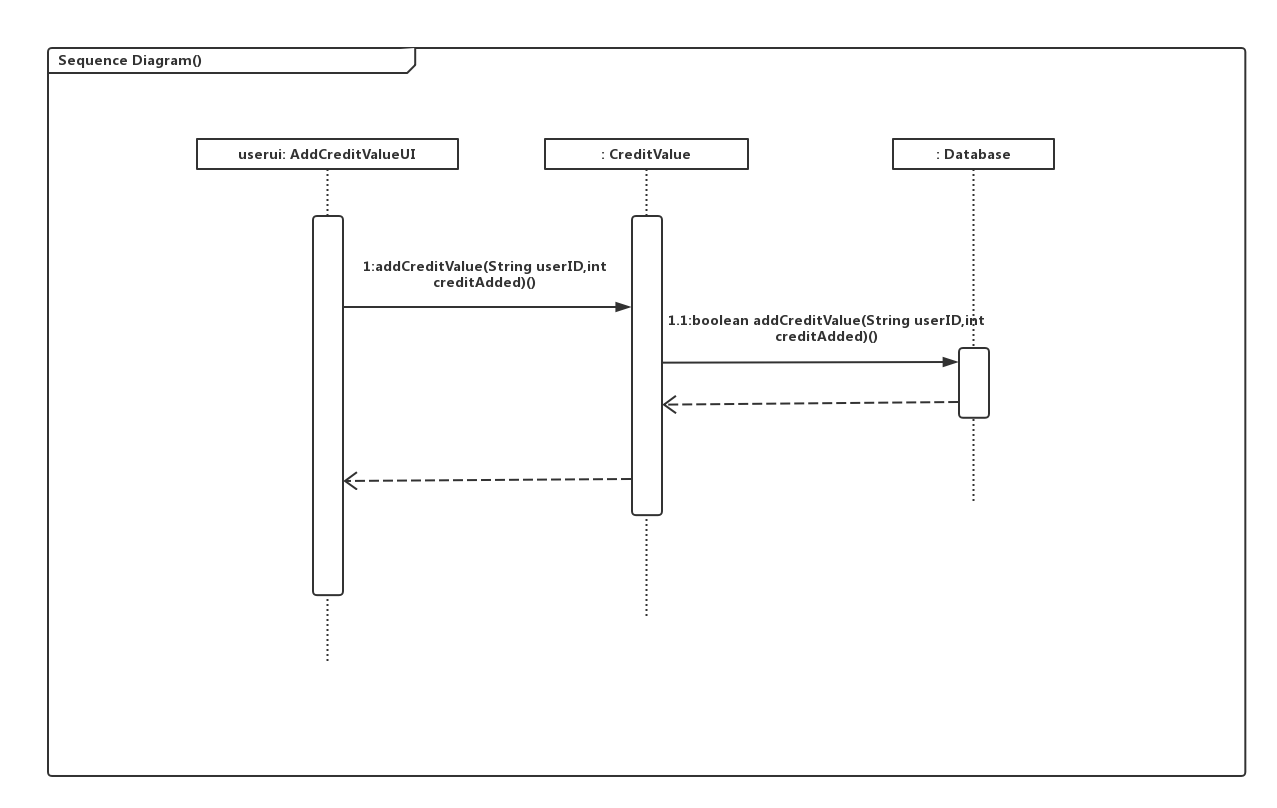
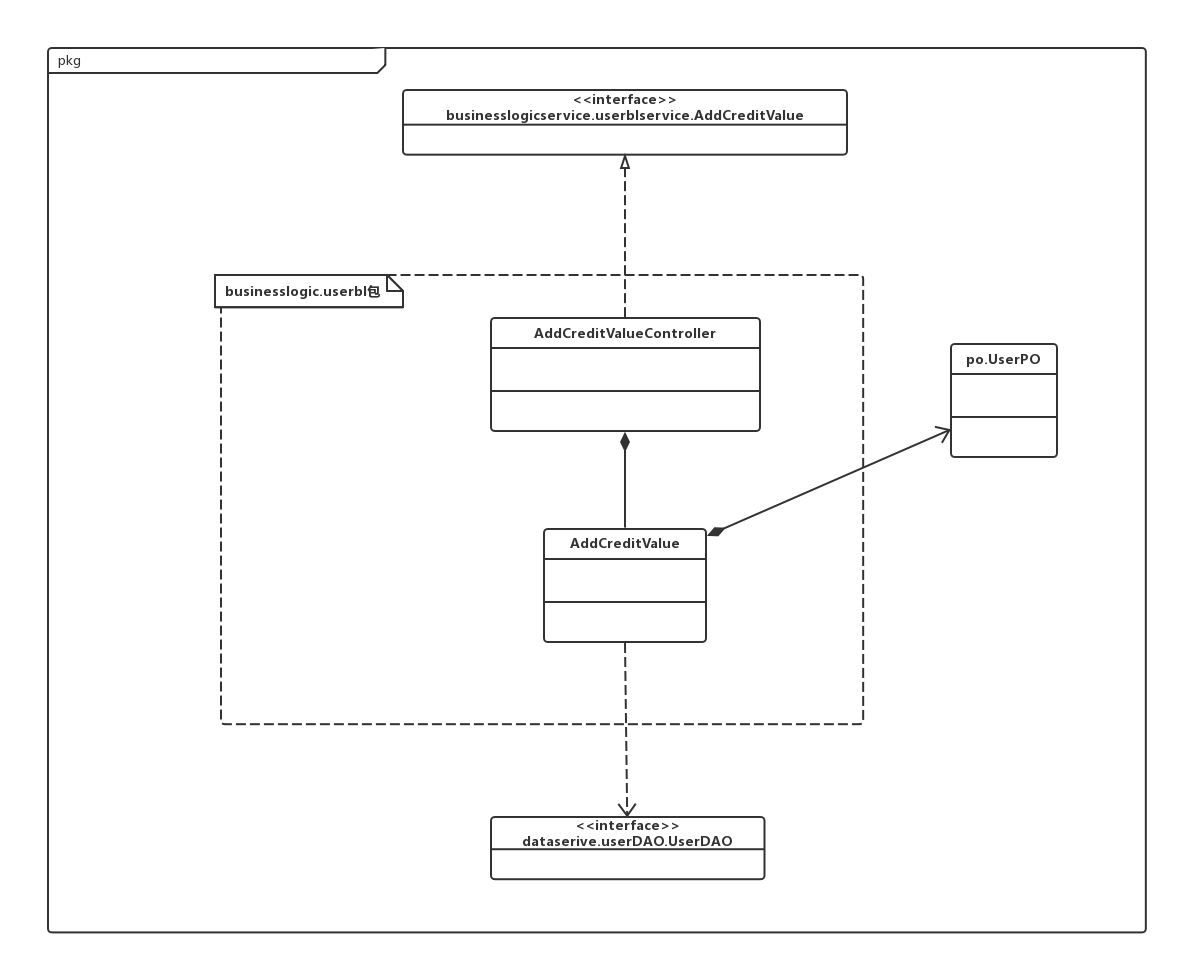
* + - 1. **SignVip**

****

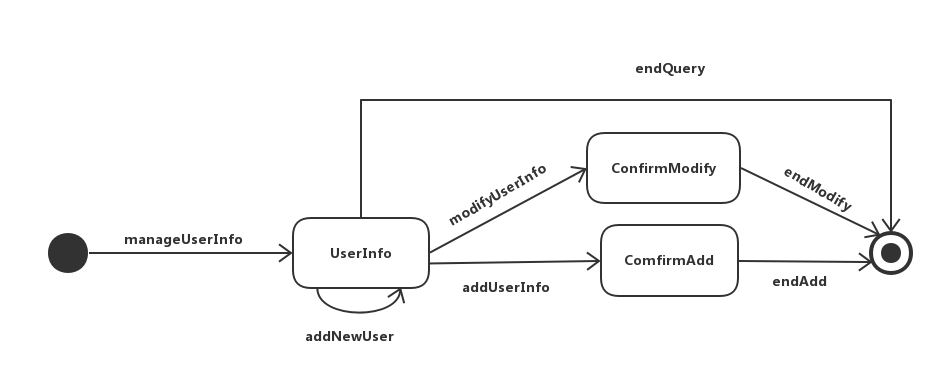
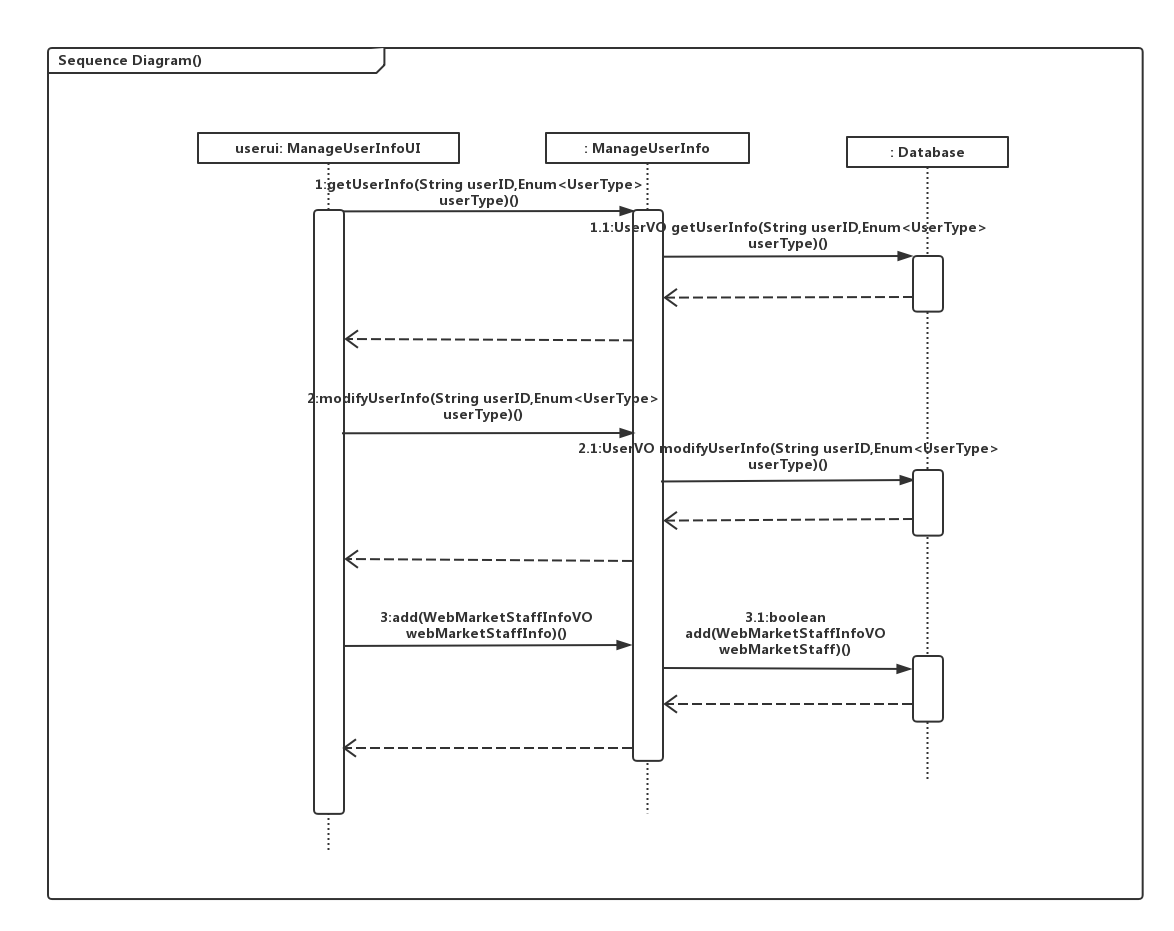
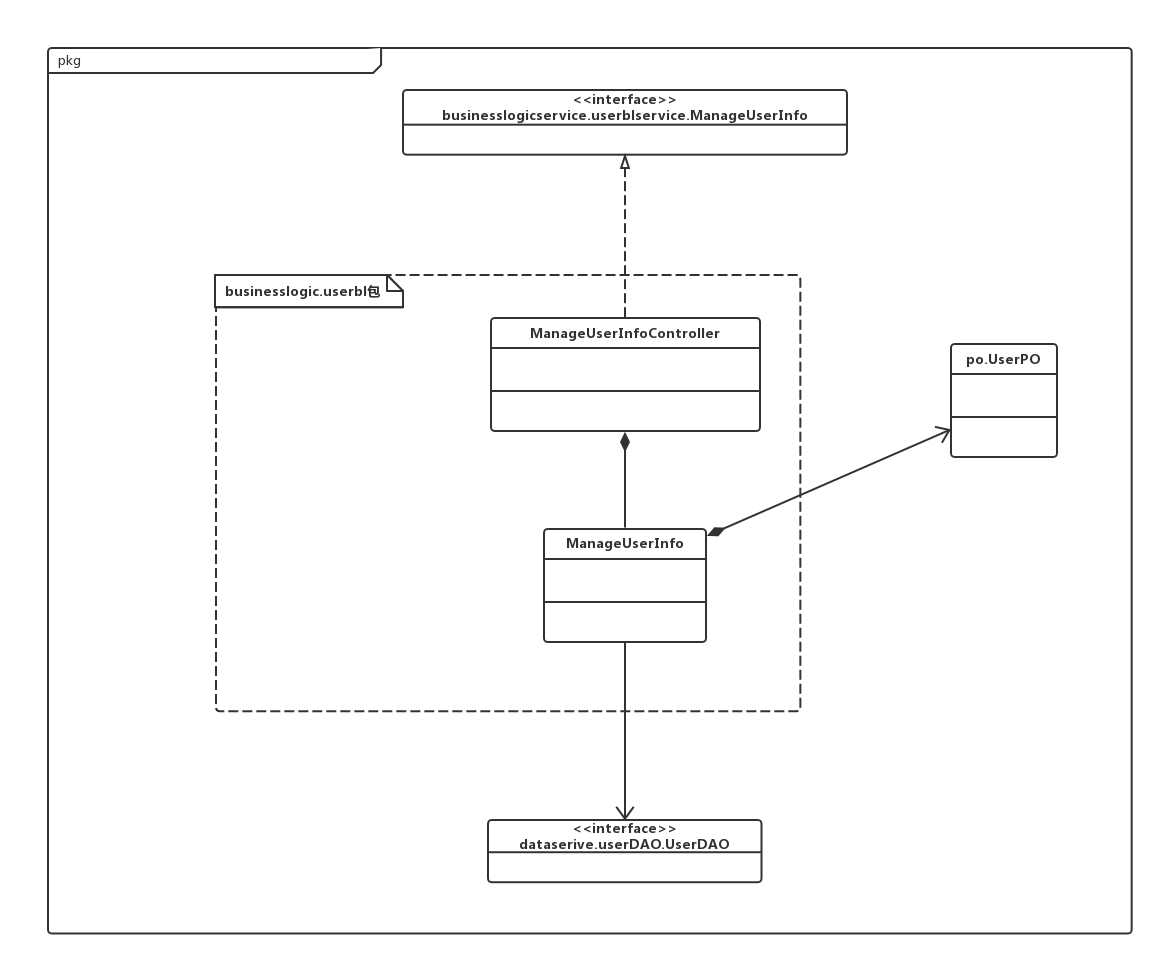
* + - 1. **QueryClientCreditRecord**

****

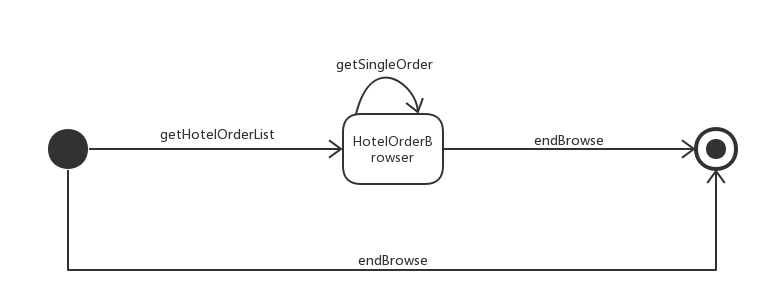
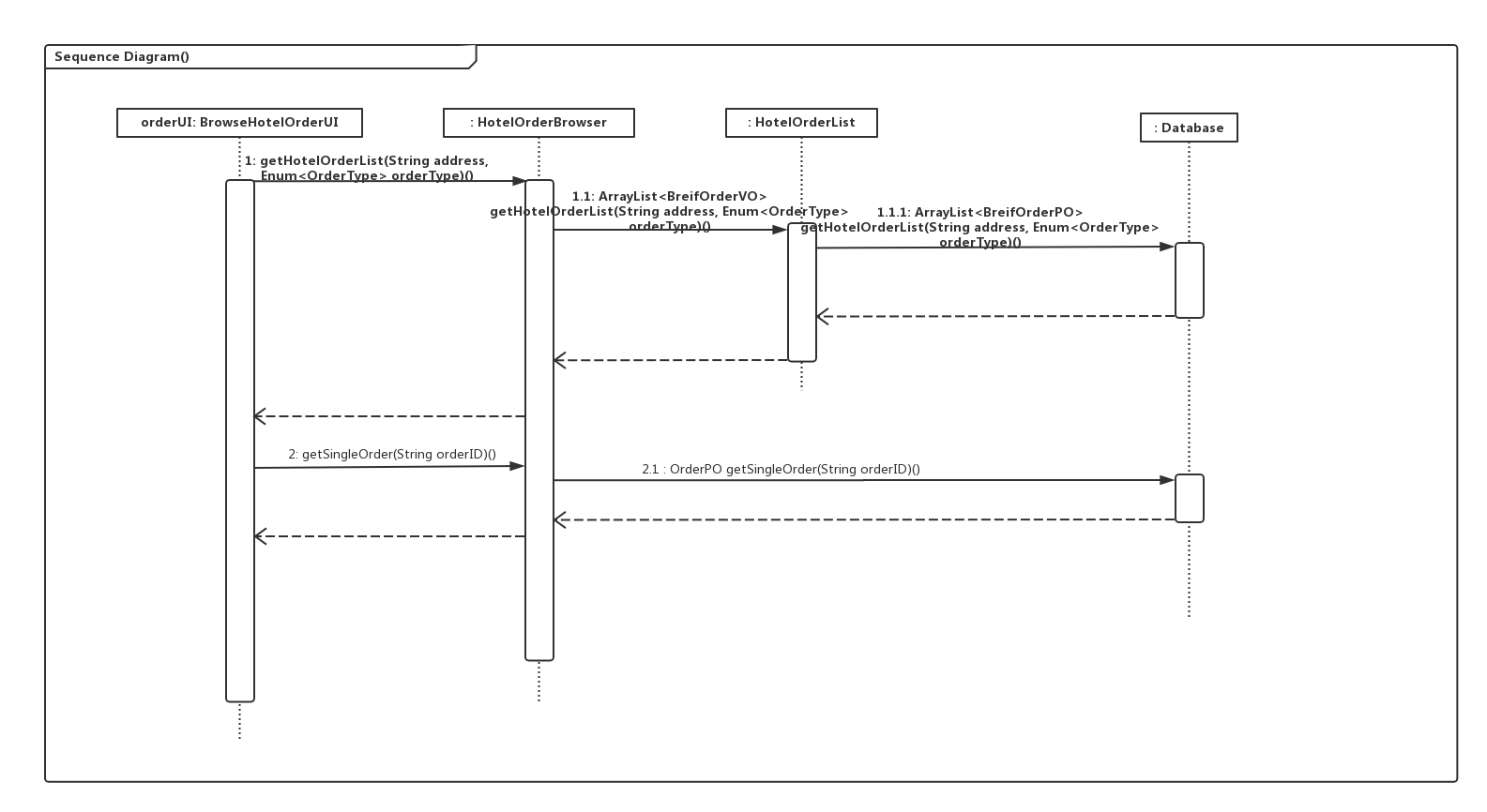
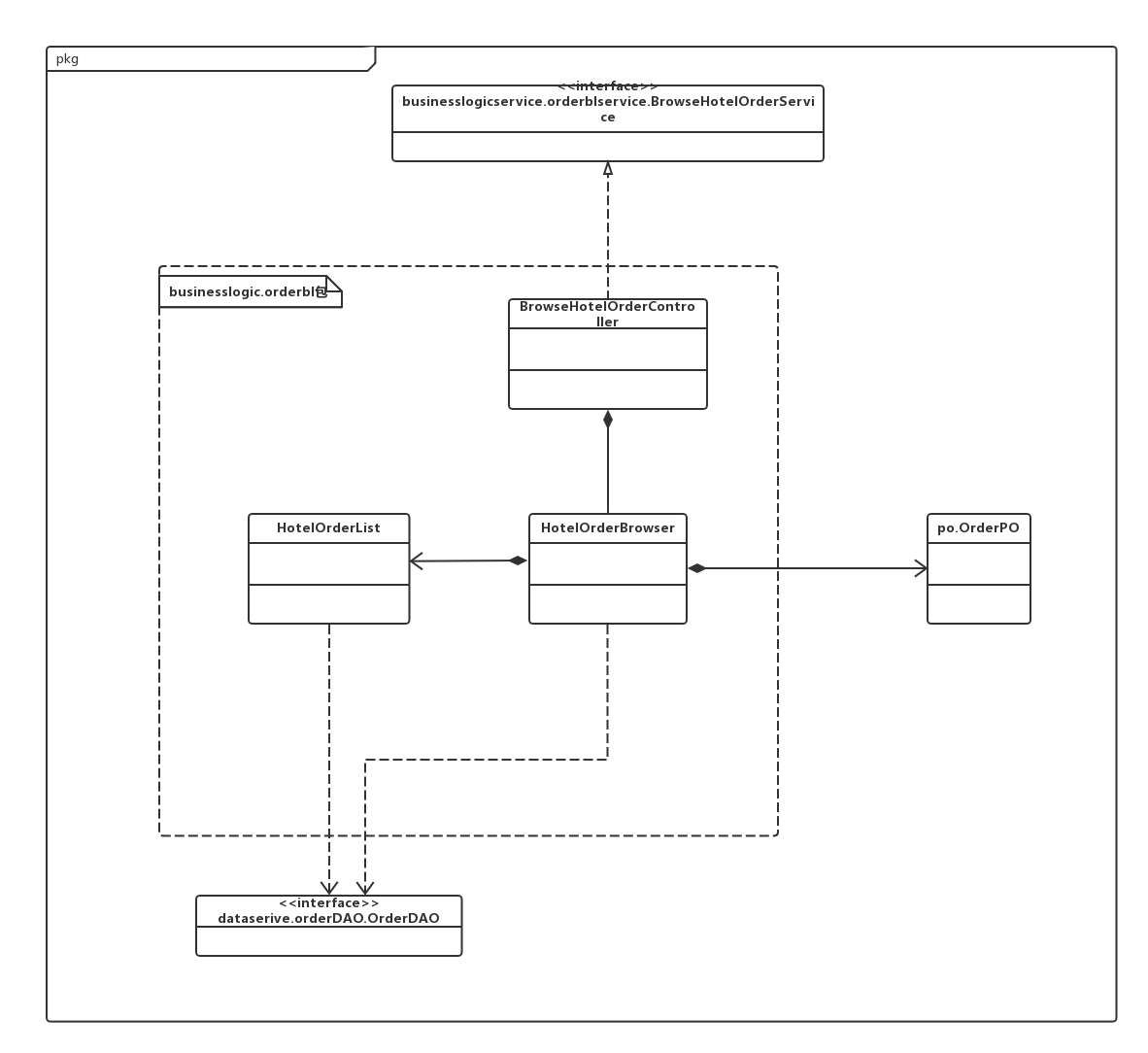
* + - 1. **AddCreditValue**

****

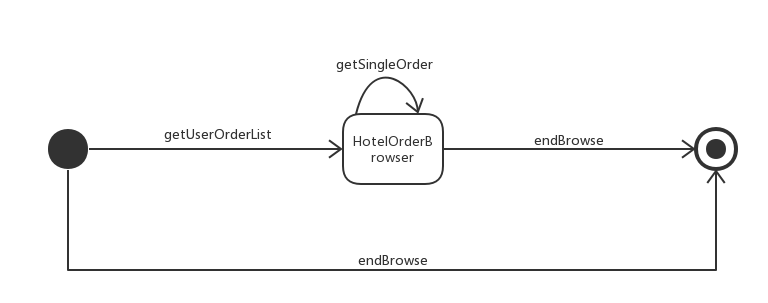
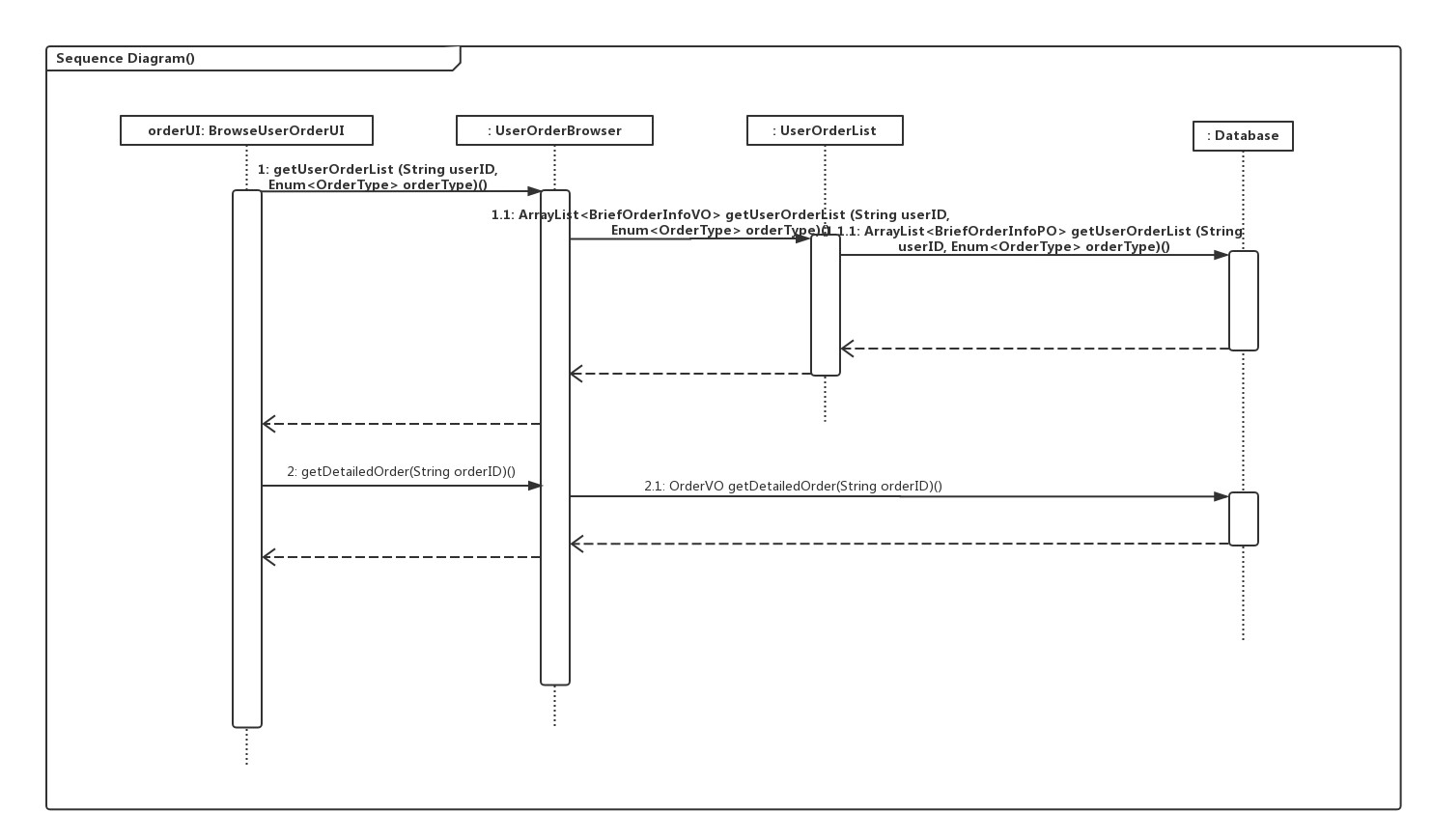
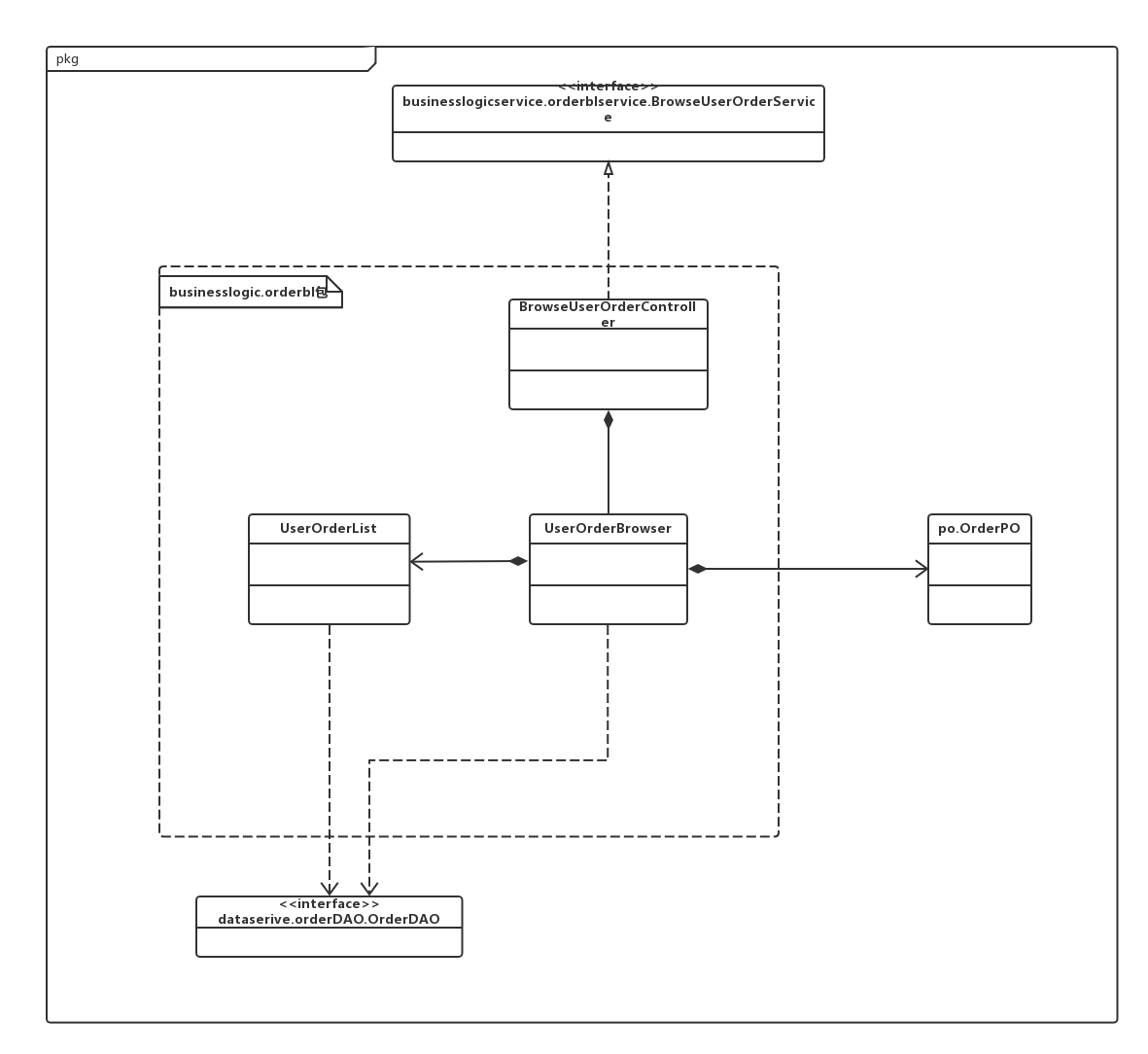
* + - 1. **ManageUserInfo**

****

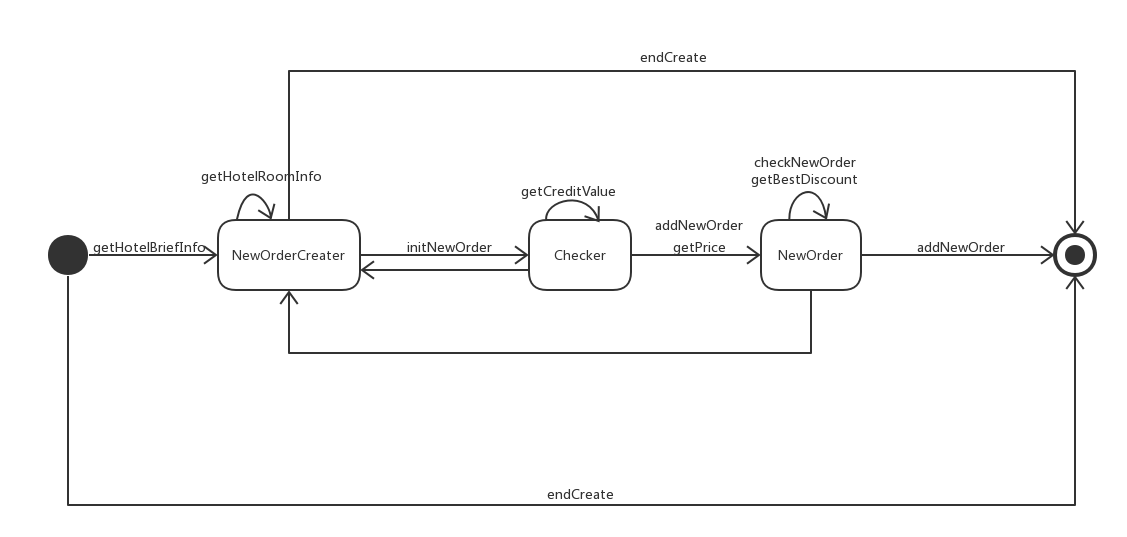
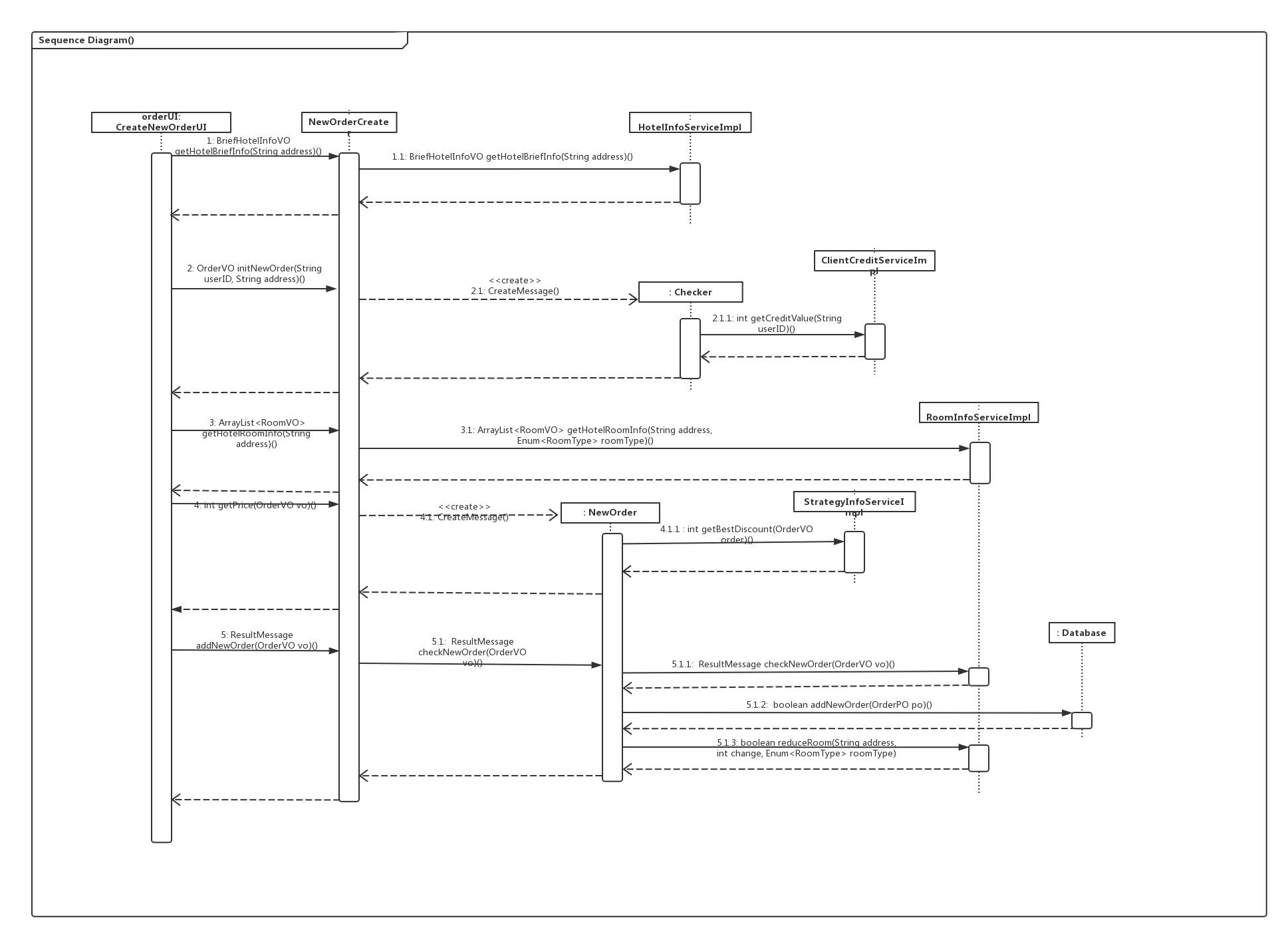
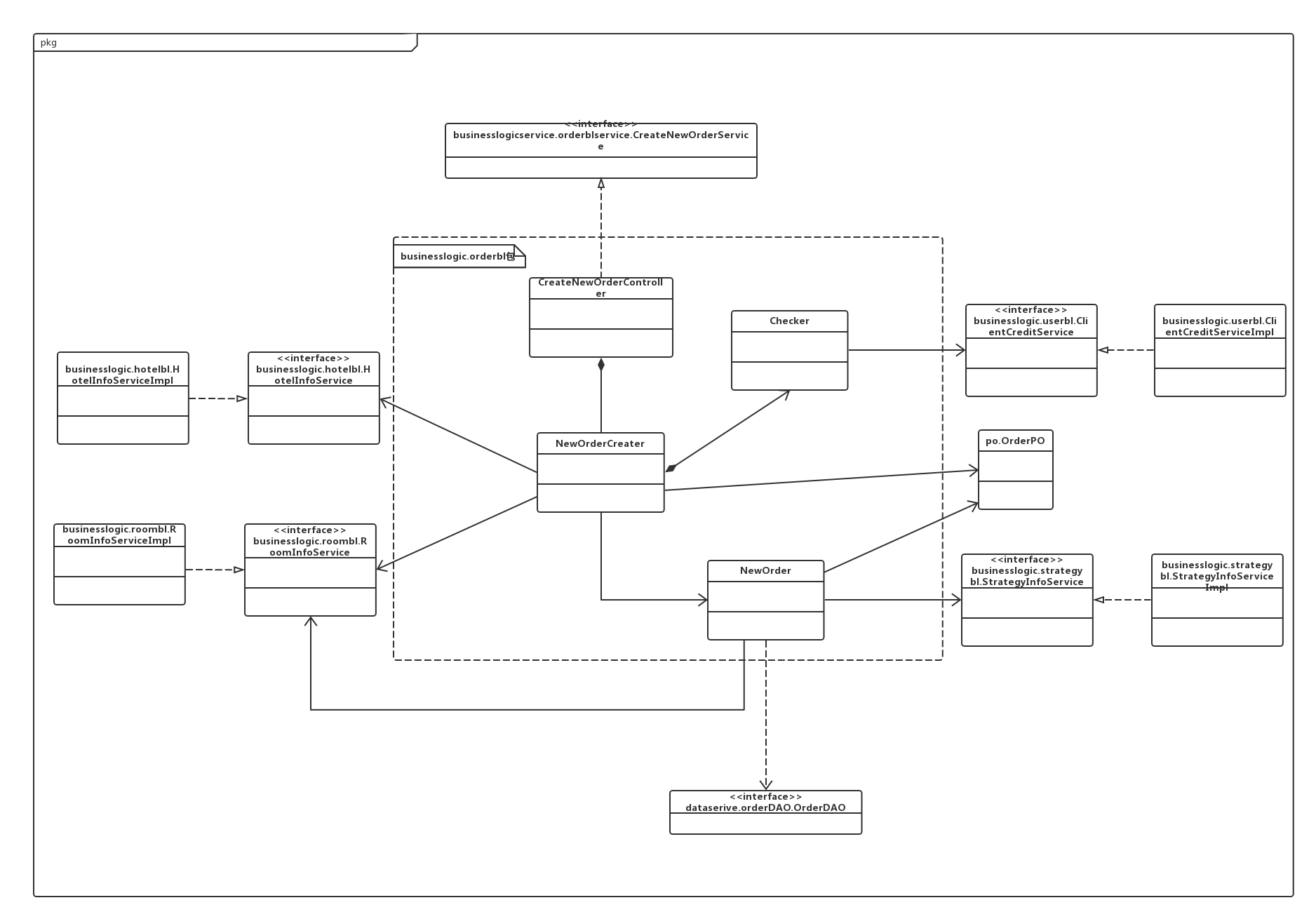
* + 1. **orderbl模块**
       1. **BrowseHotelOrder**

****

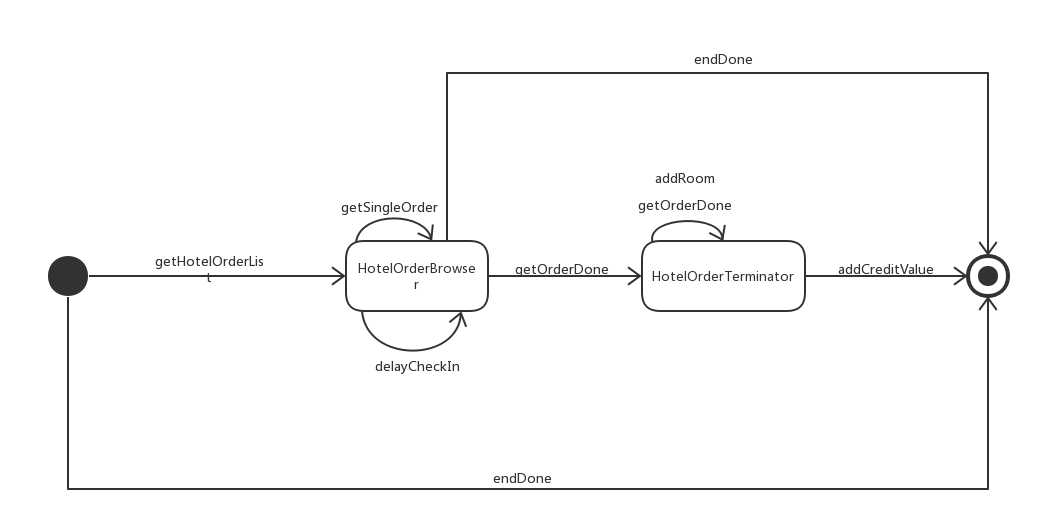
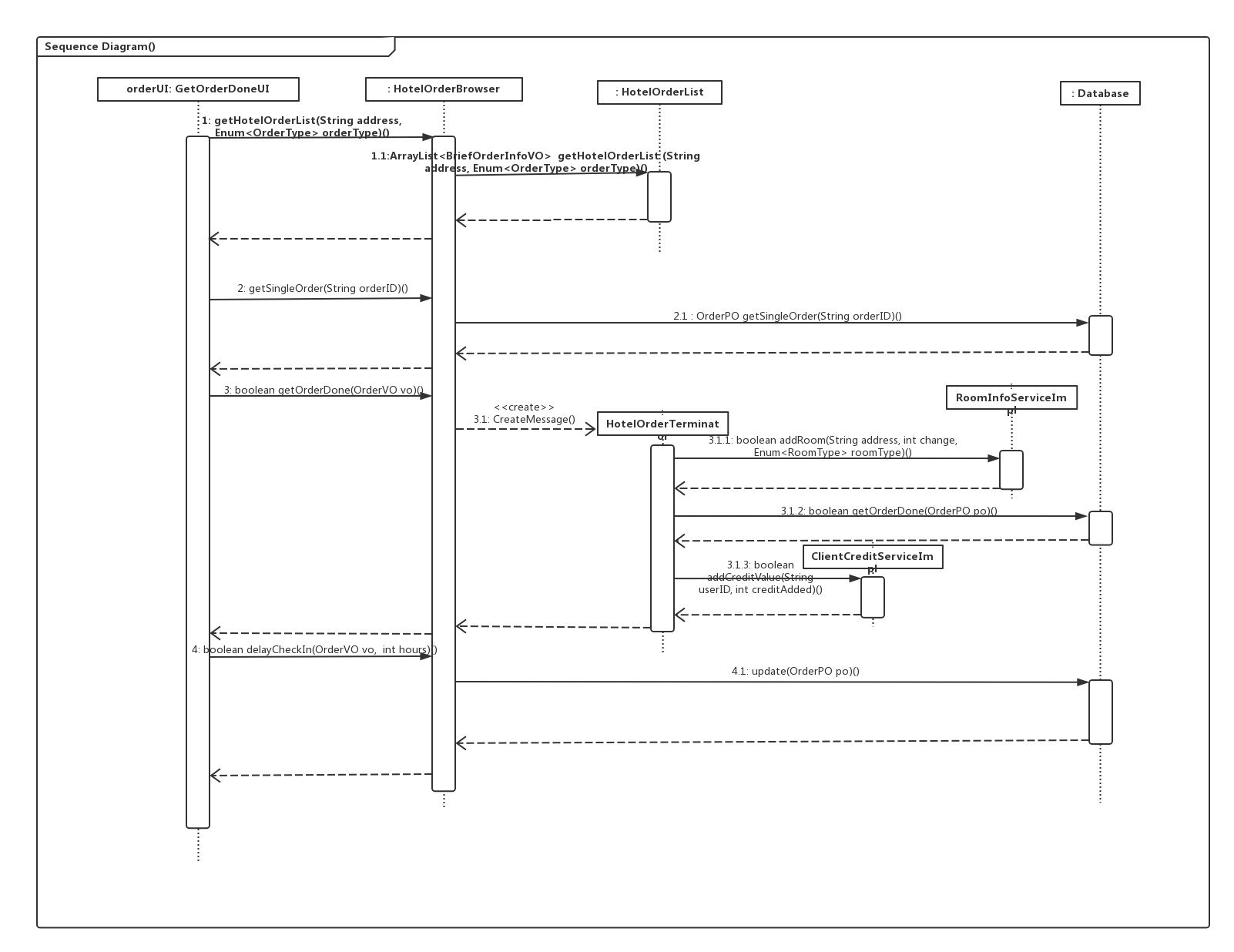
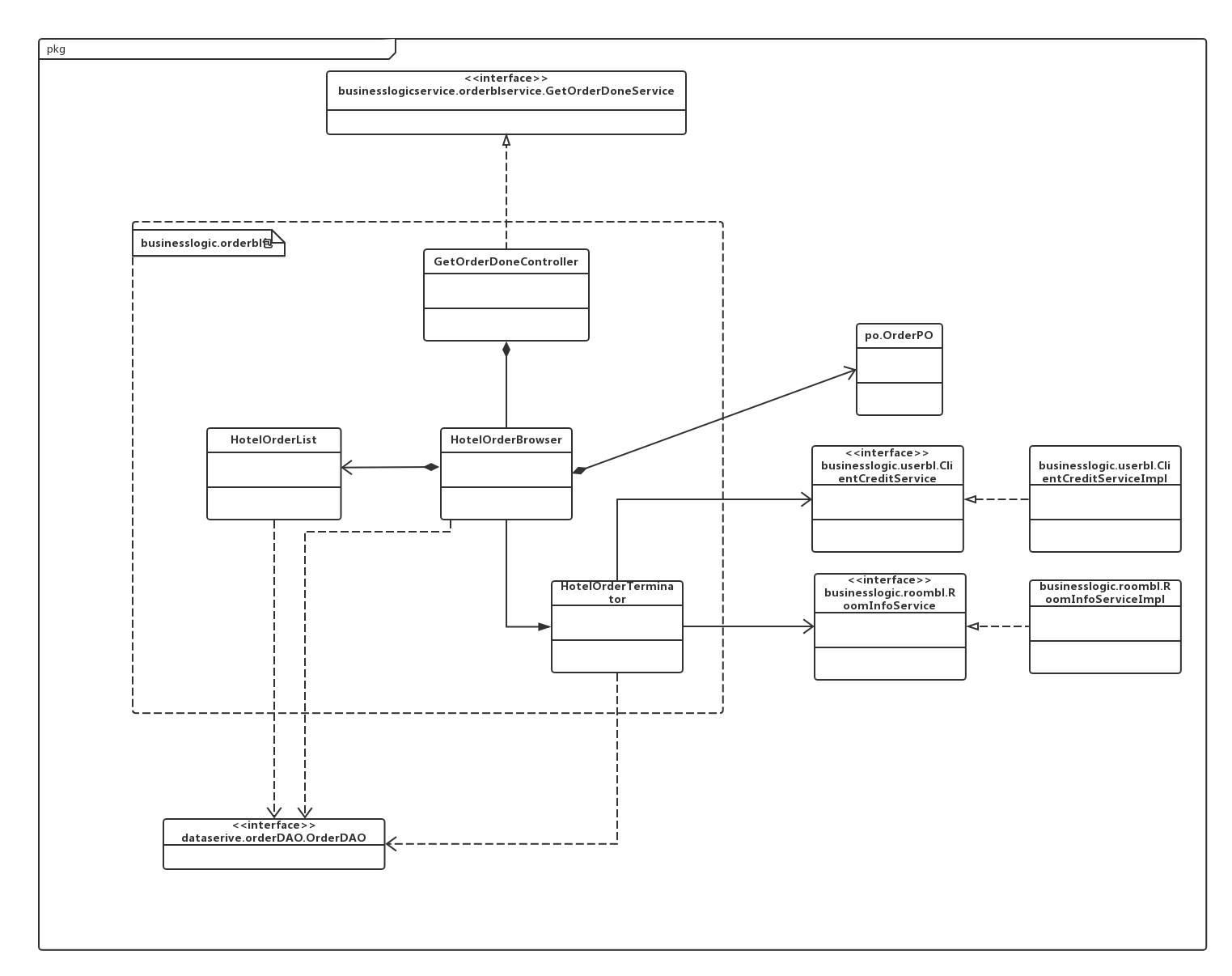
* + - 1. **BrowseUserOrder**

****

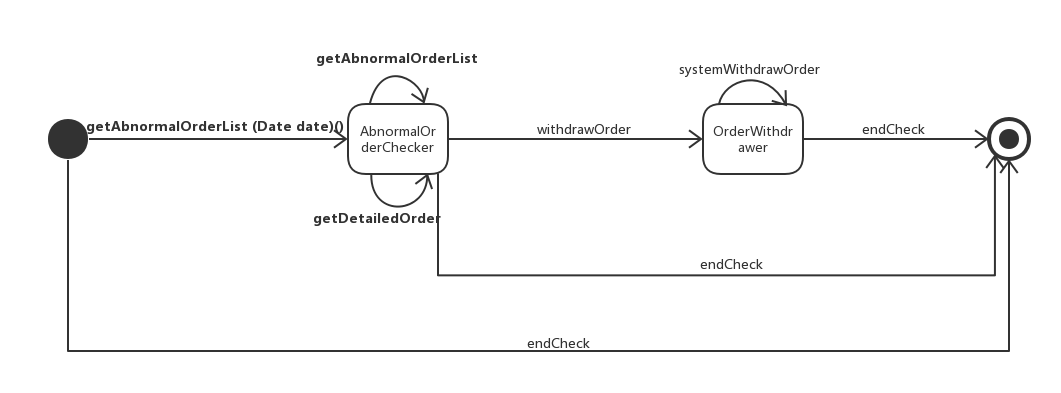
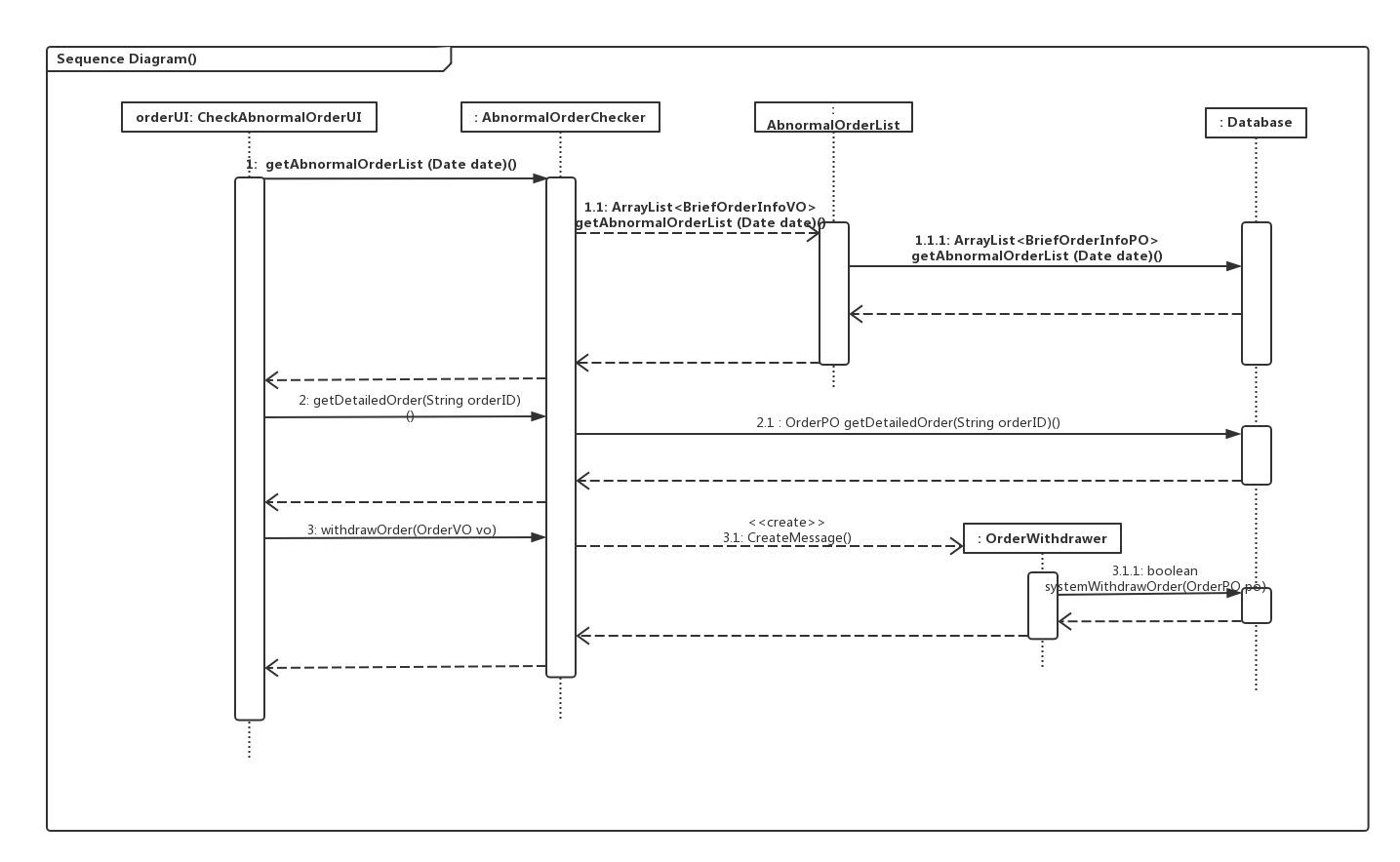
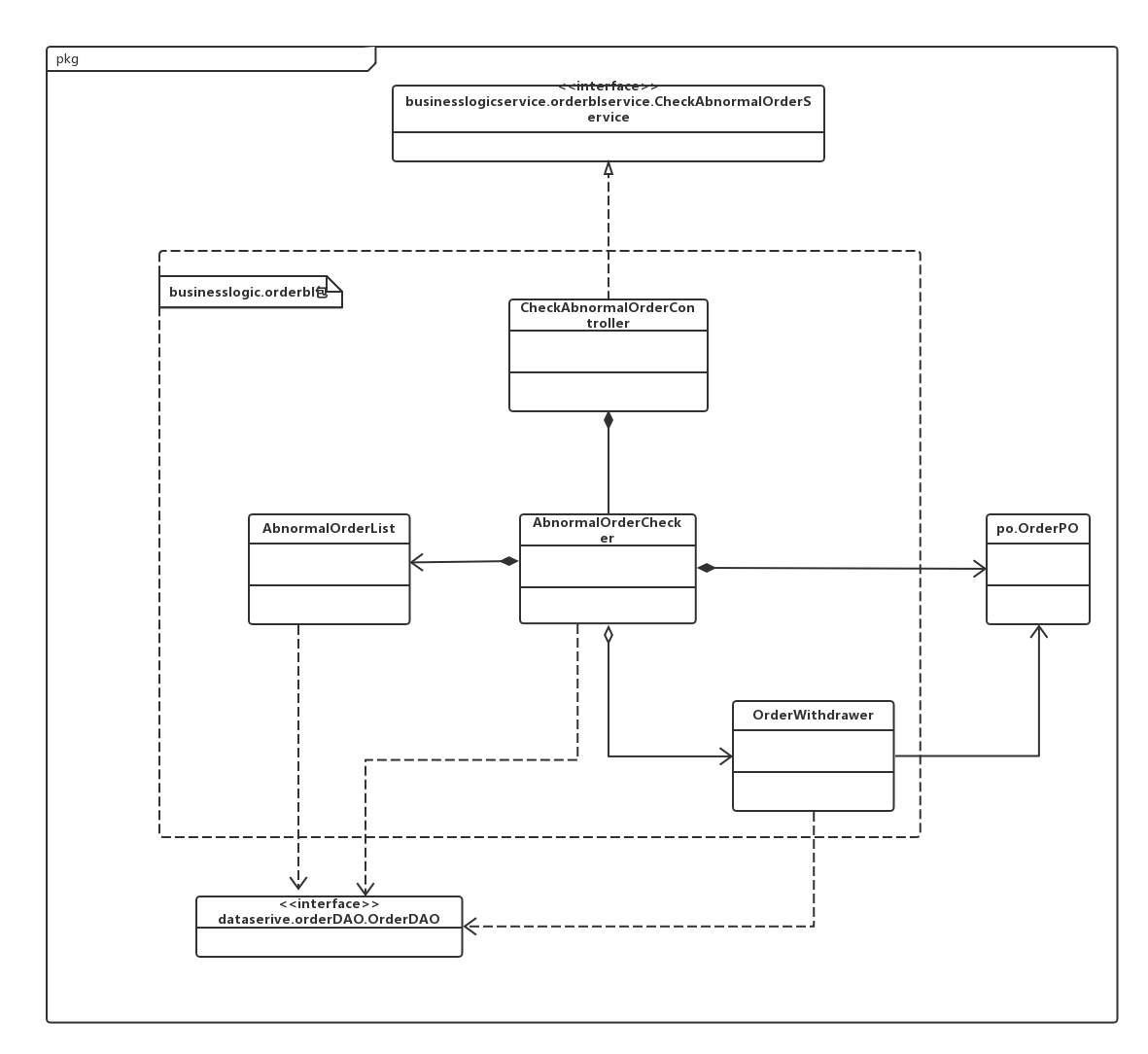
* + - 1. **CreateNewOrder**

****

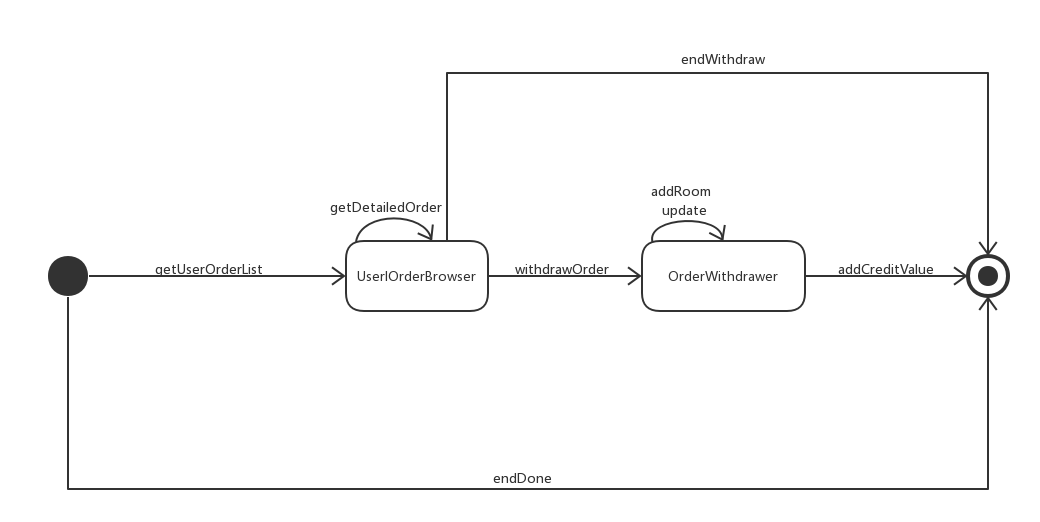
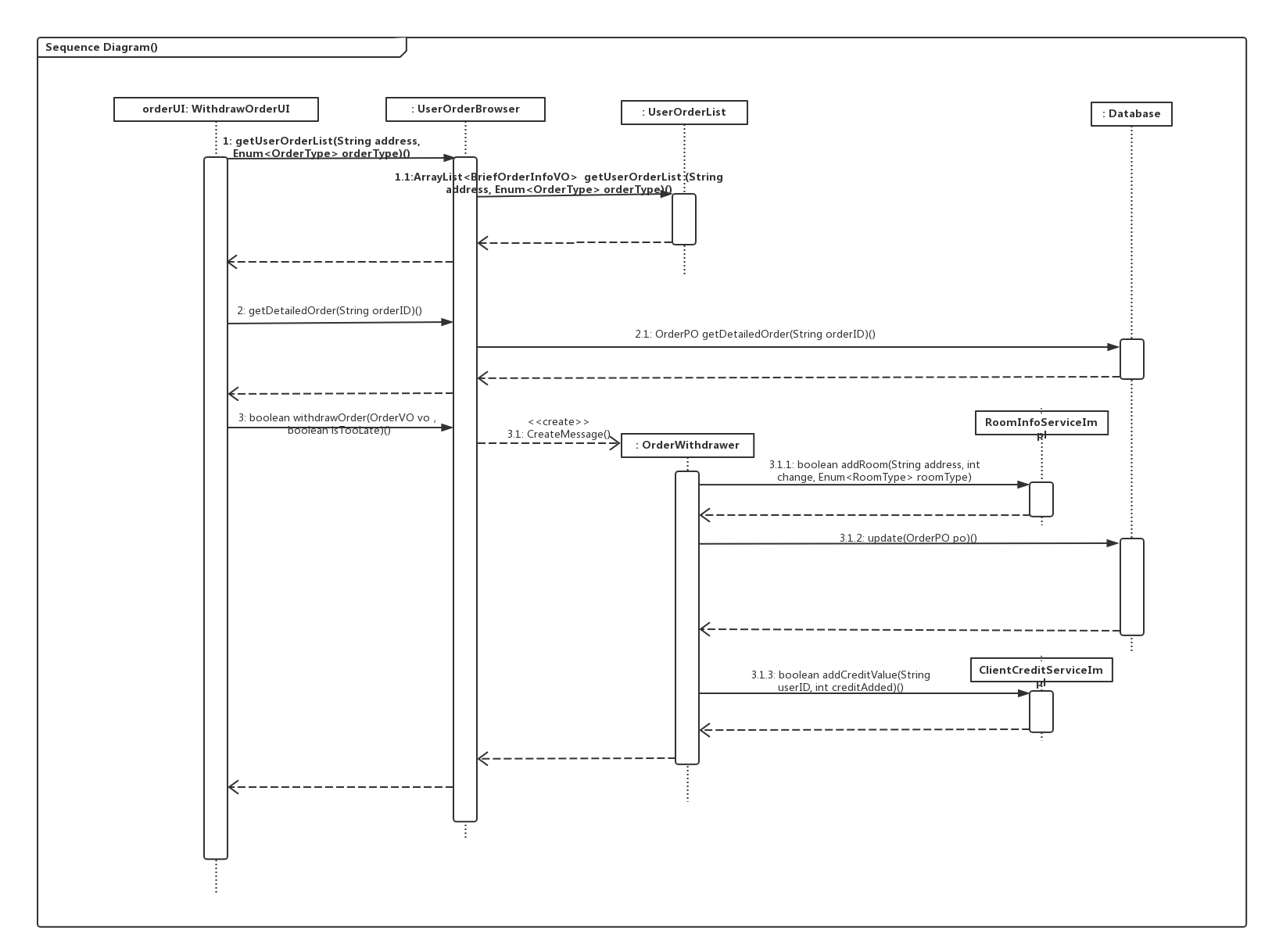
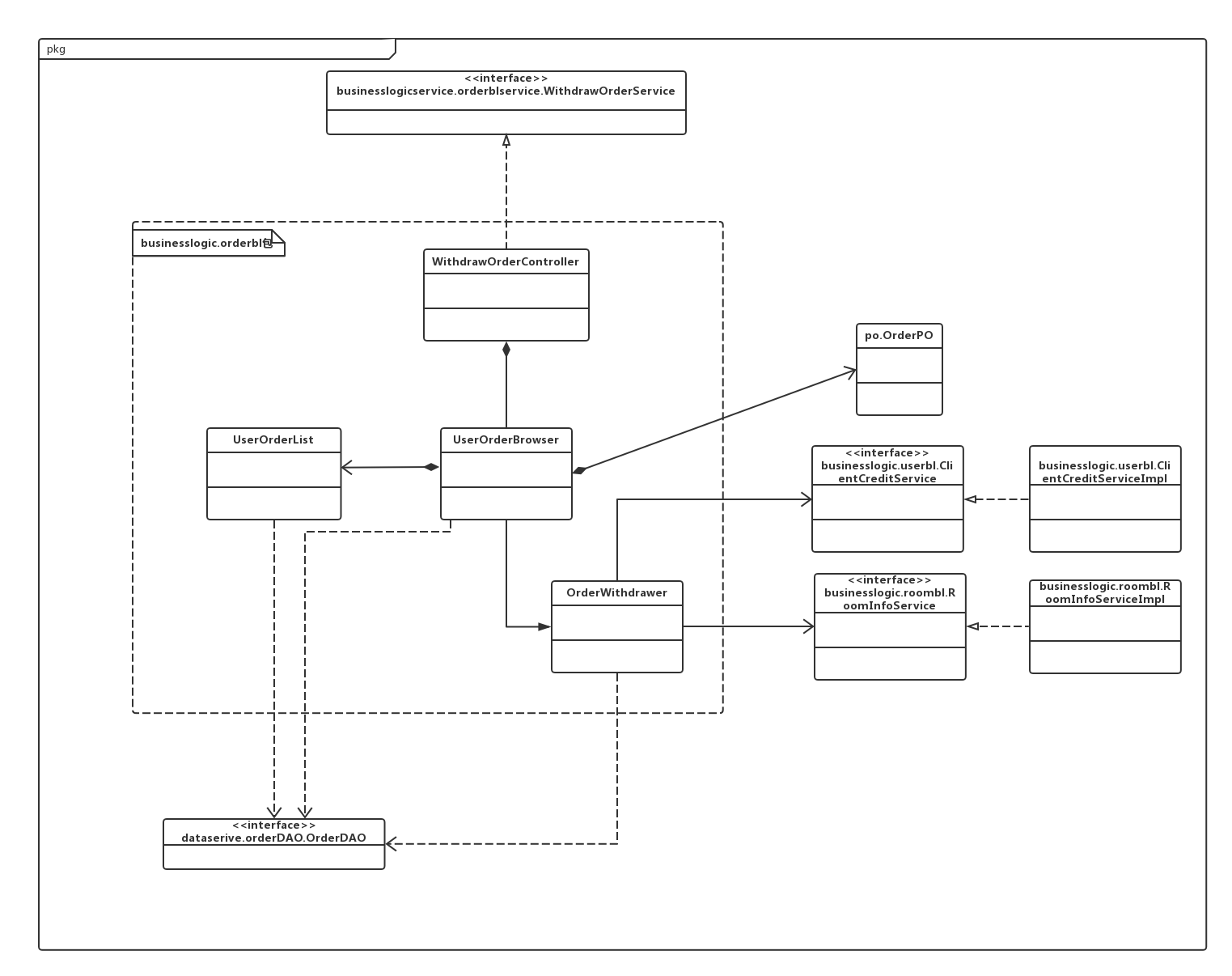
* + - 1. **GetOrderDone**

****

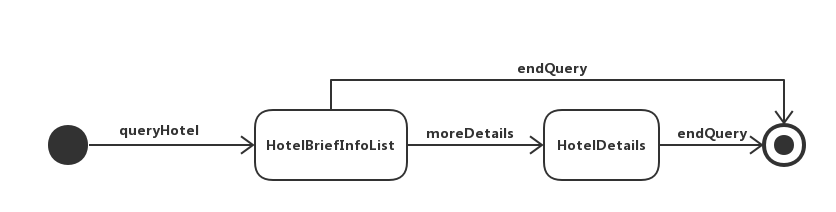
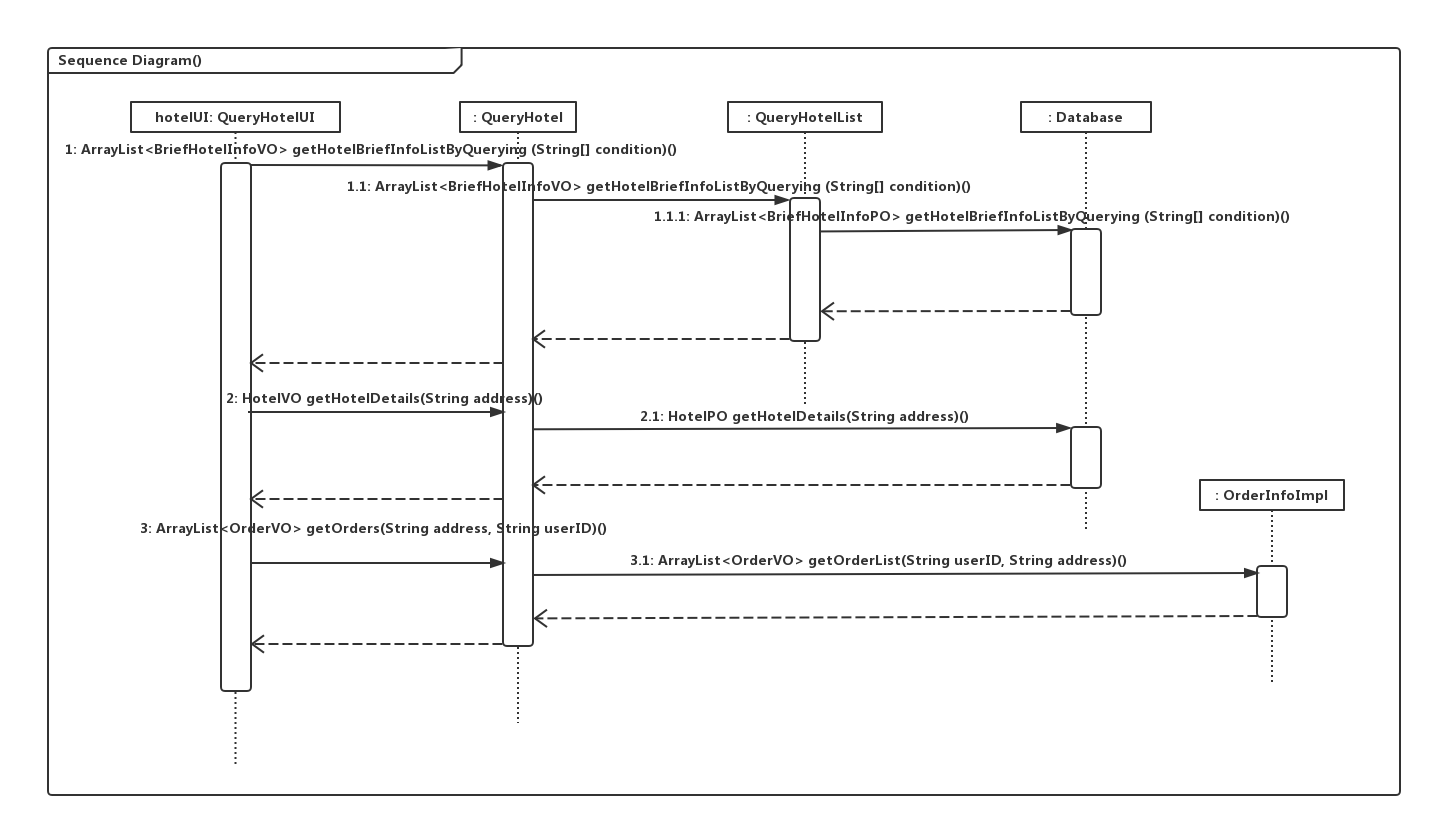
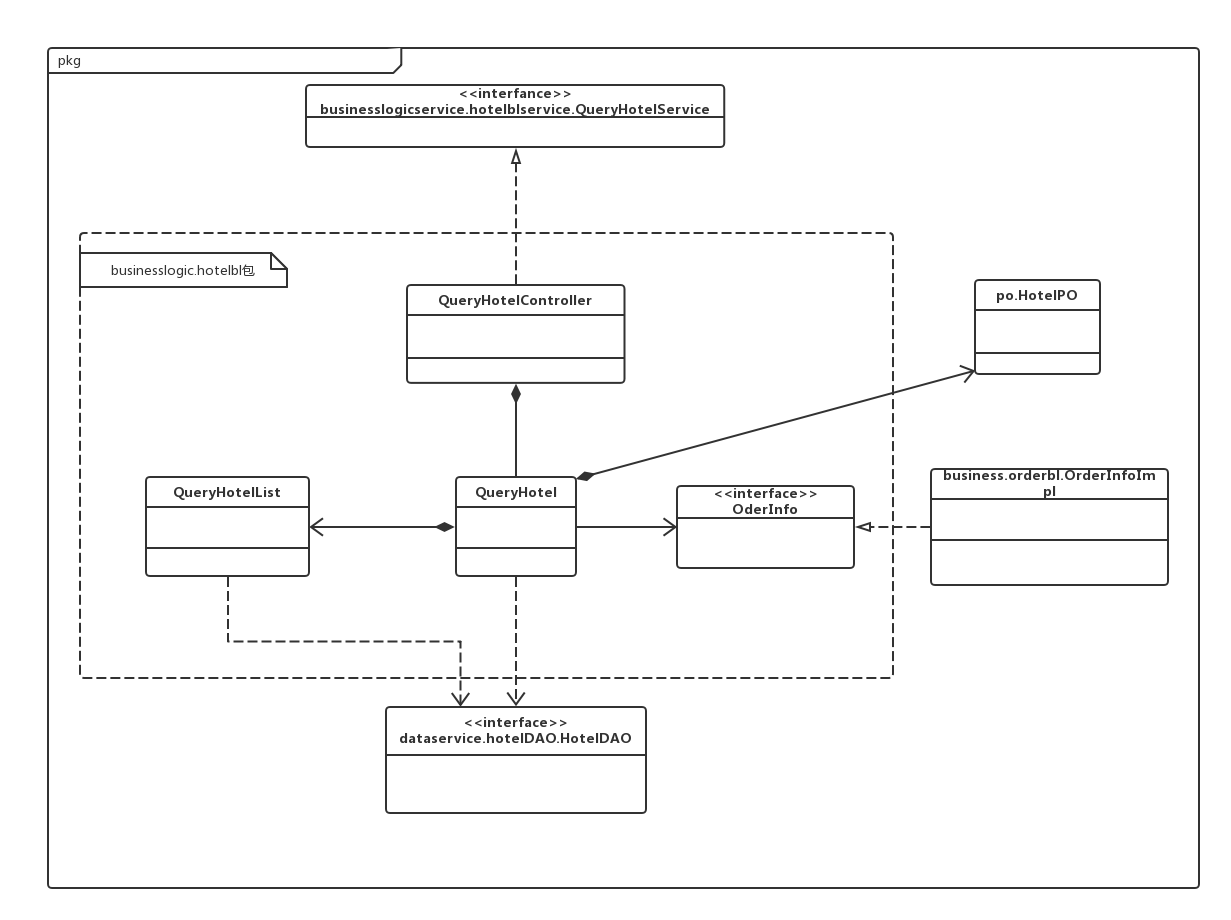
* + - 1. **CheckAbnormalOrder**

****

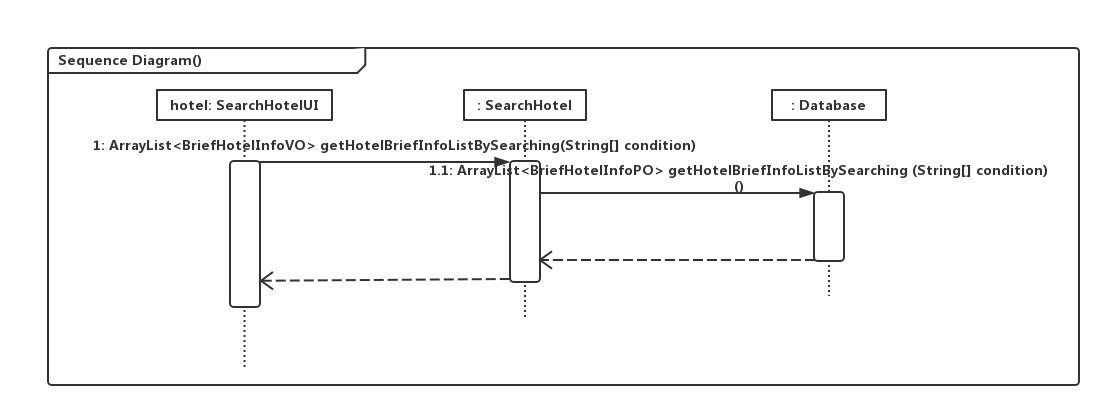
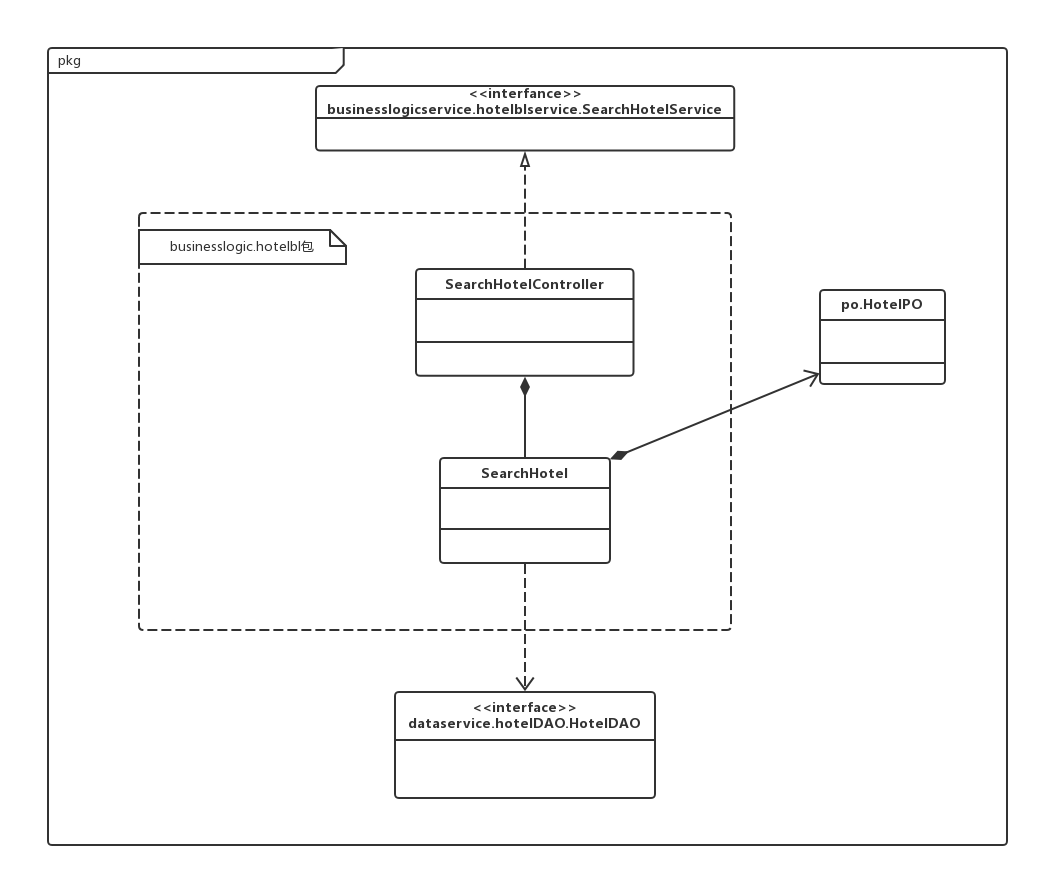
* + - 1. **WithDrawOrder**

****

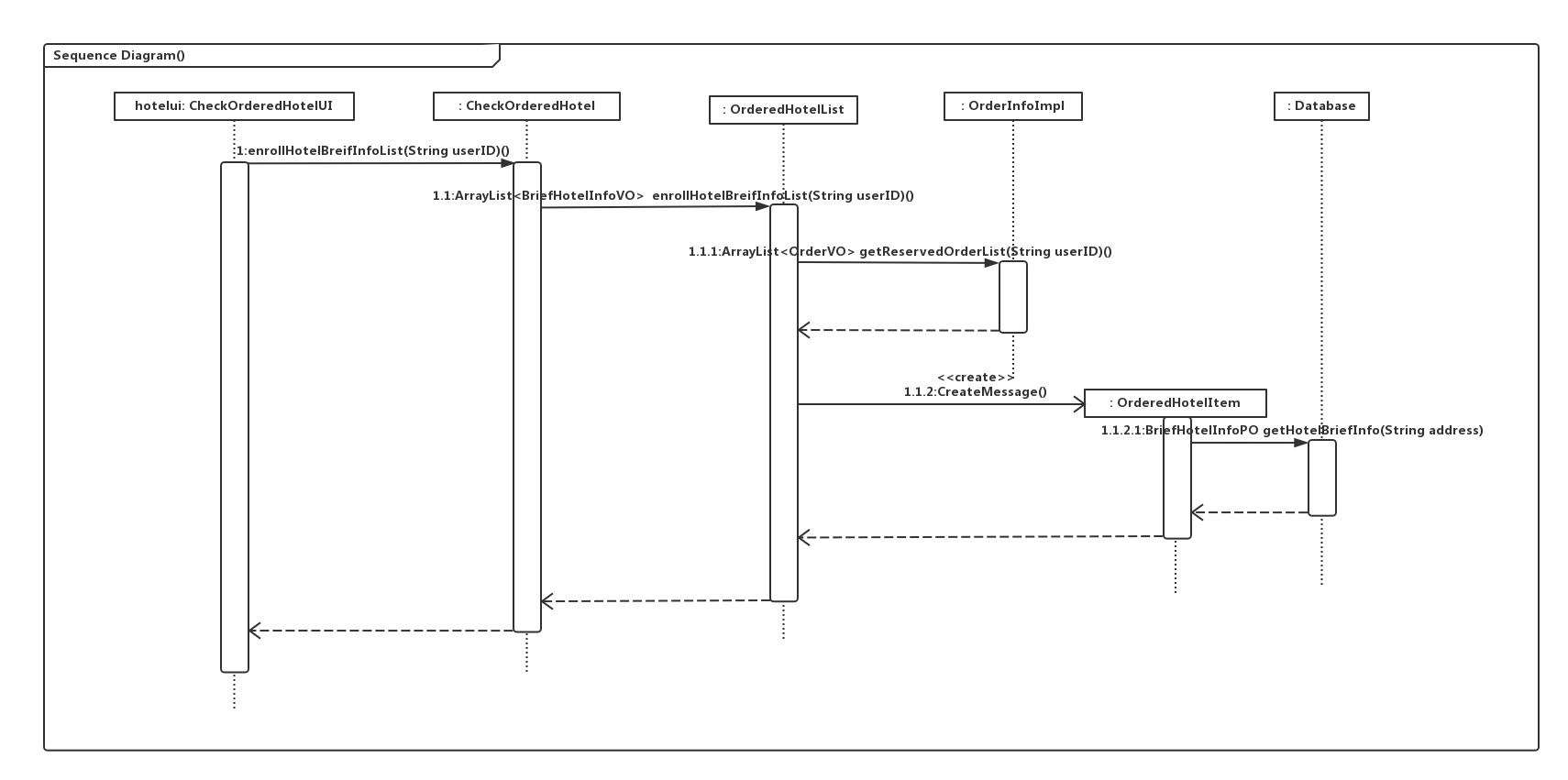
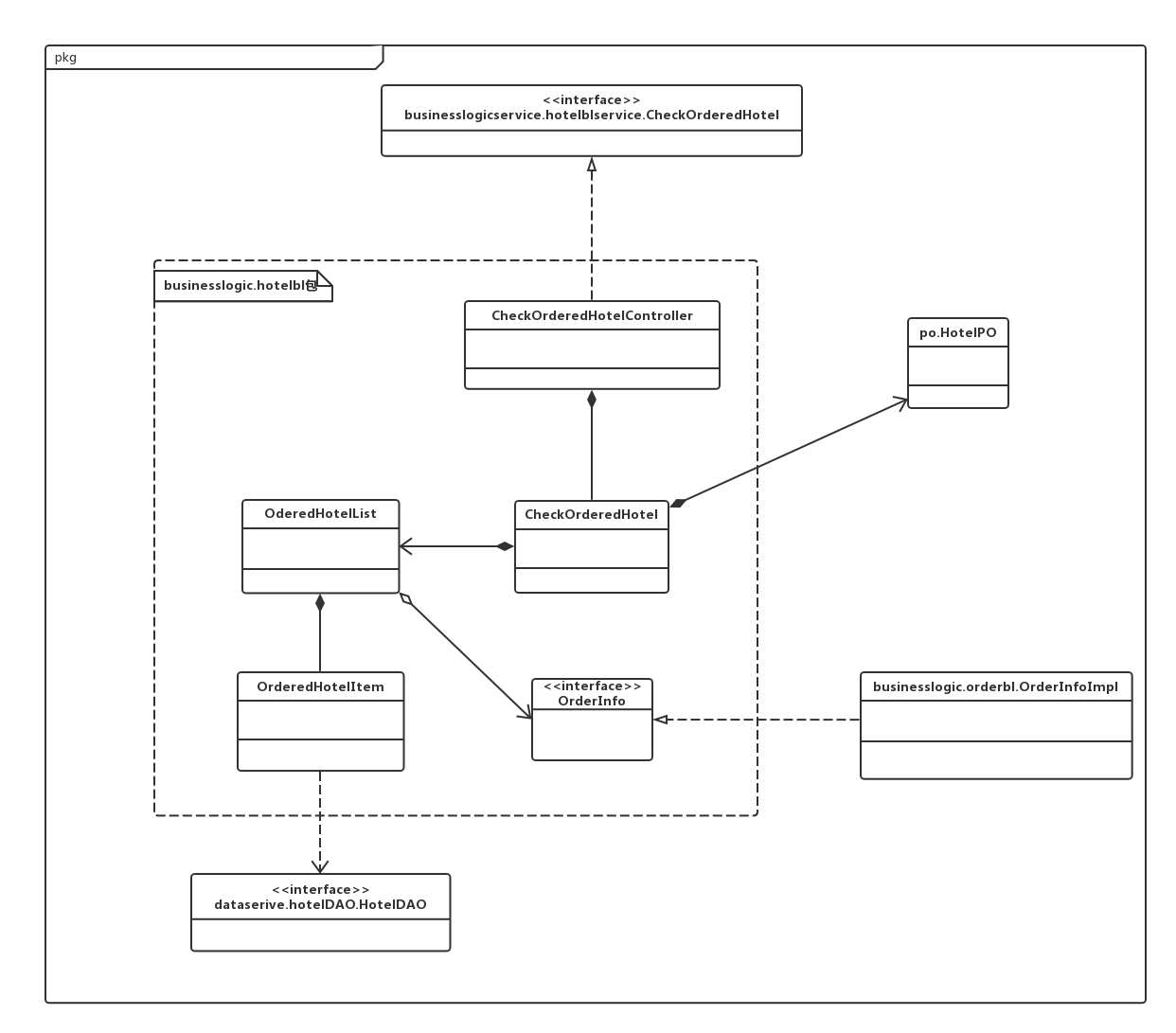
* + 1. **hotelbl模块**
       1. **QueryHotel**

****

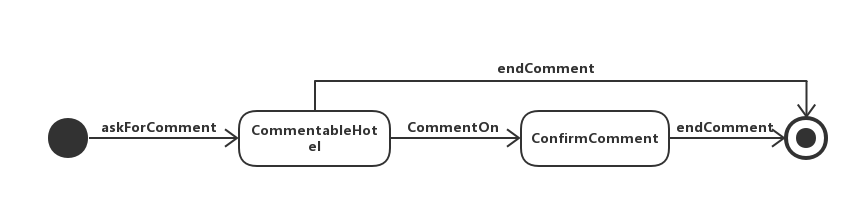
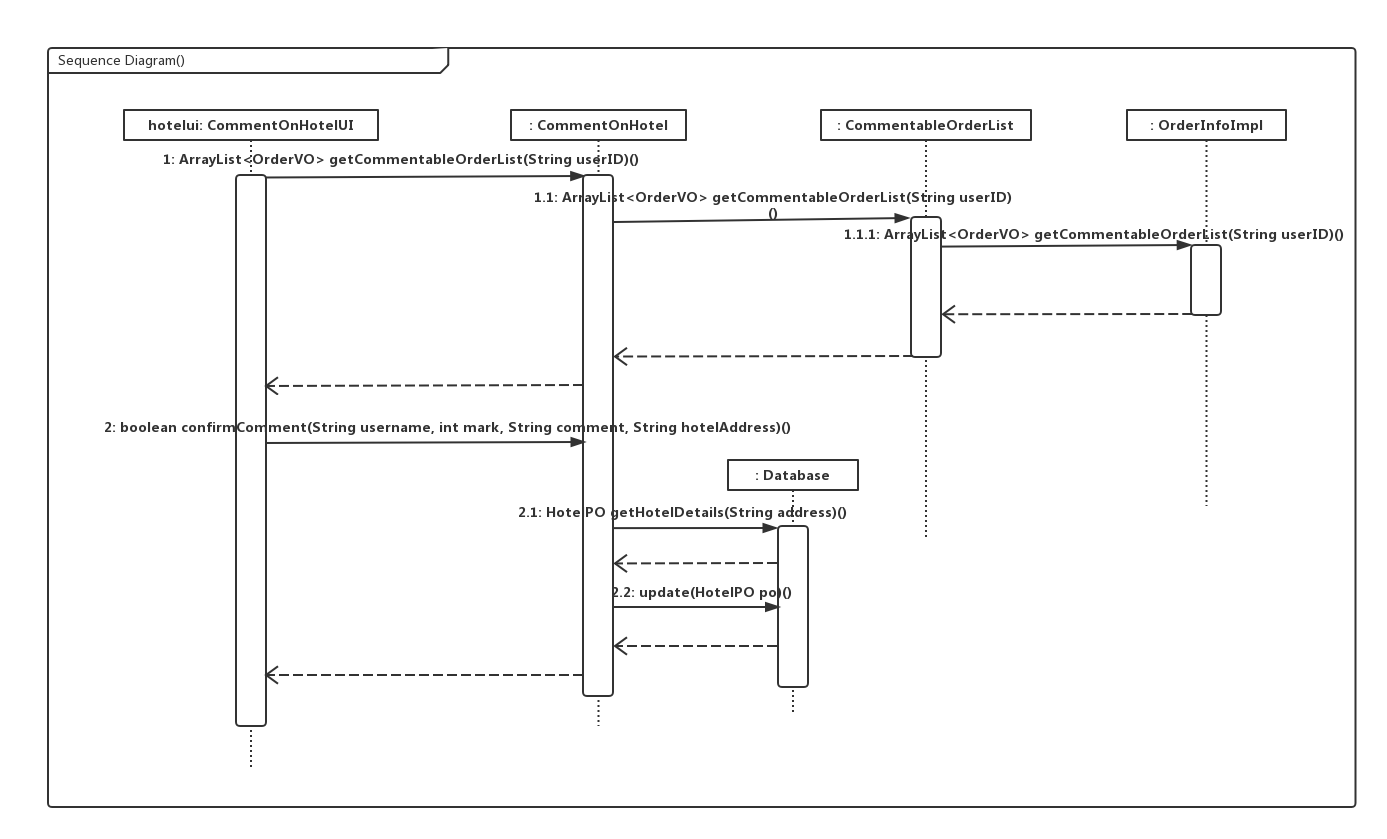
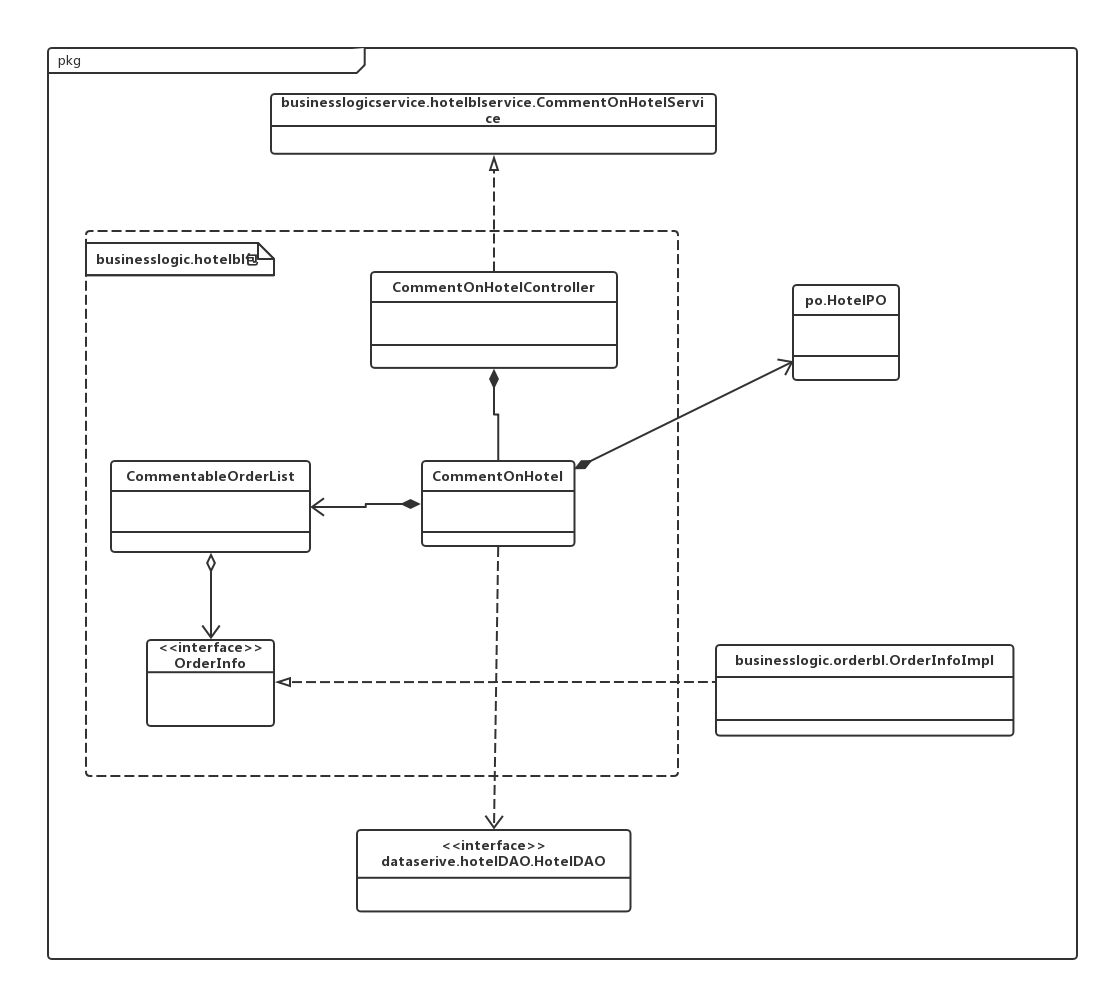
* + - 1. **SearchHotel**

****

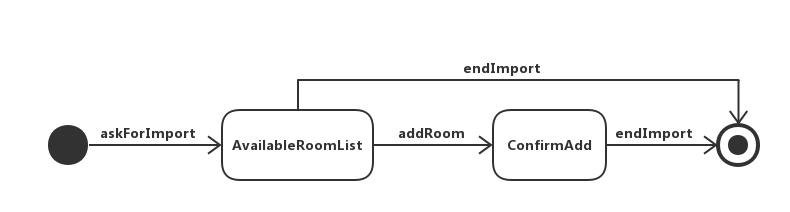
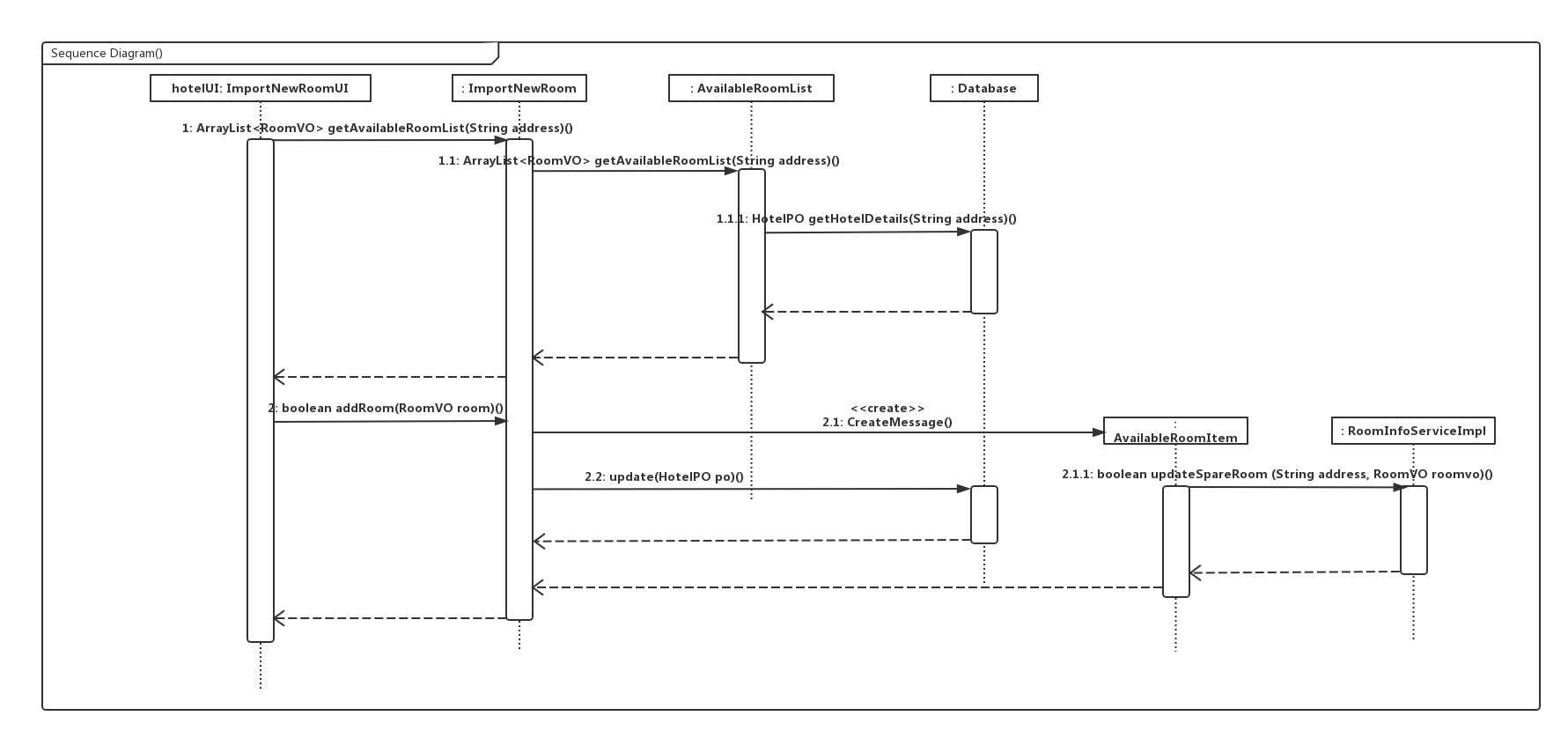
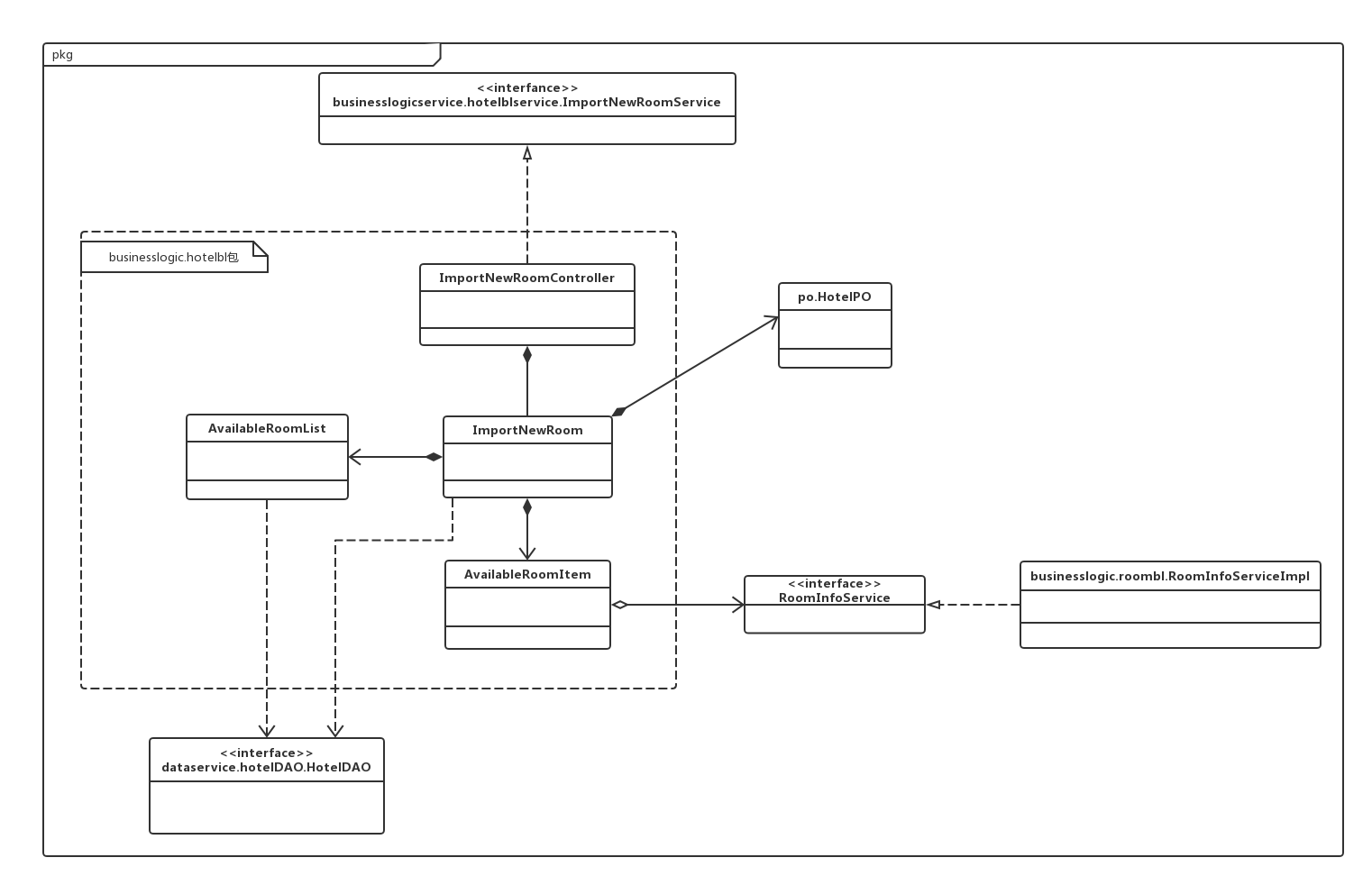
* + - 1. **CheckOrderedHotel**

****

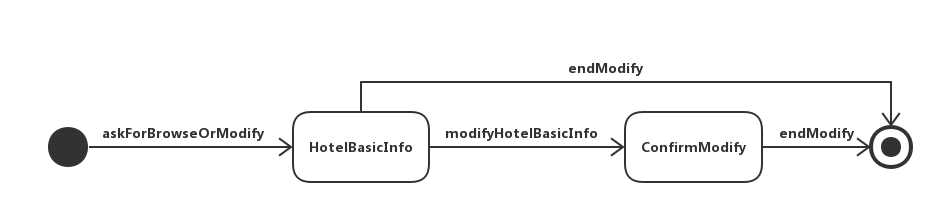
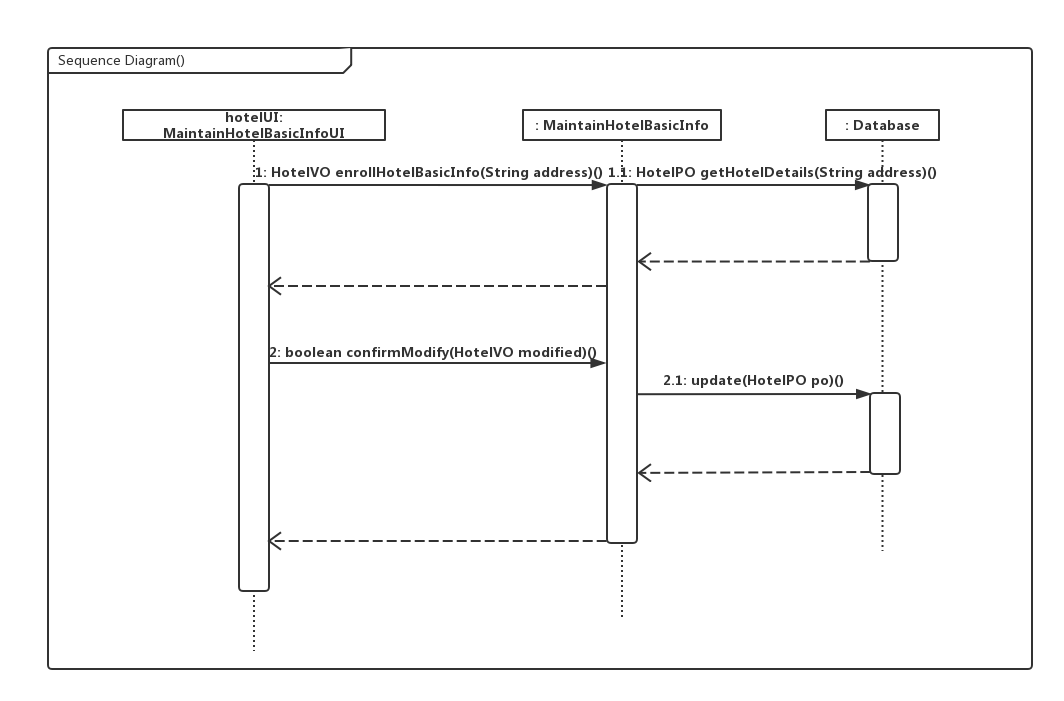
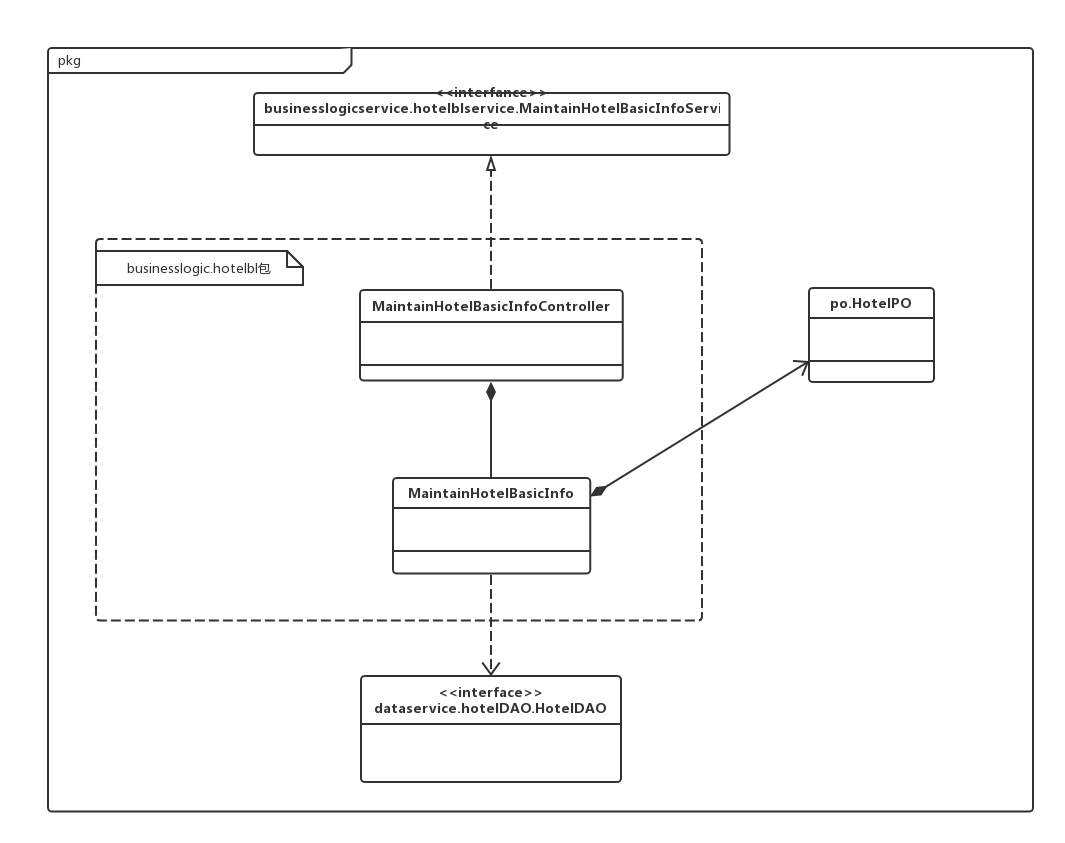
* + - 1. **CommentOnHotel**

****

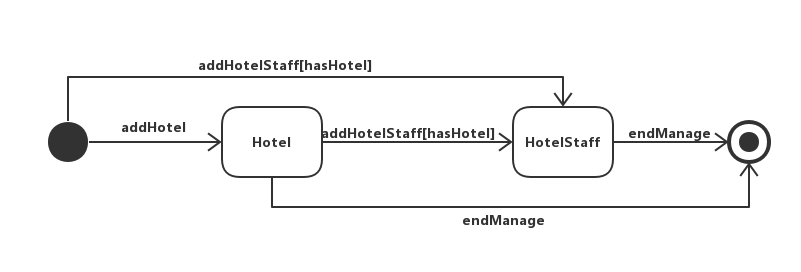
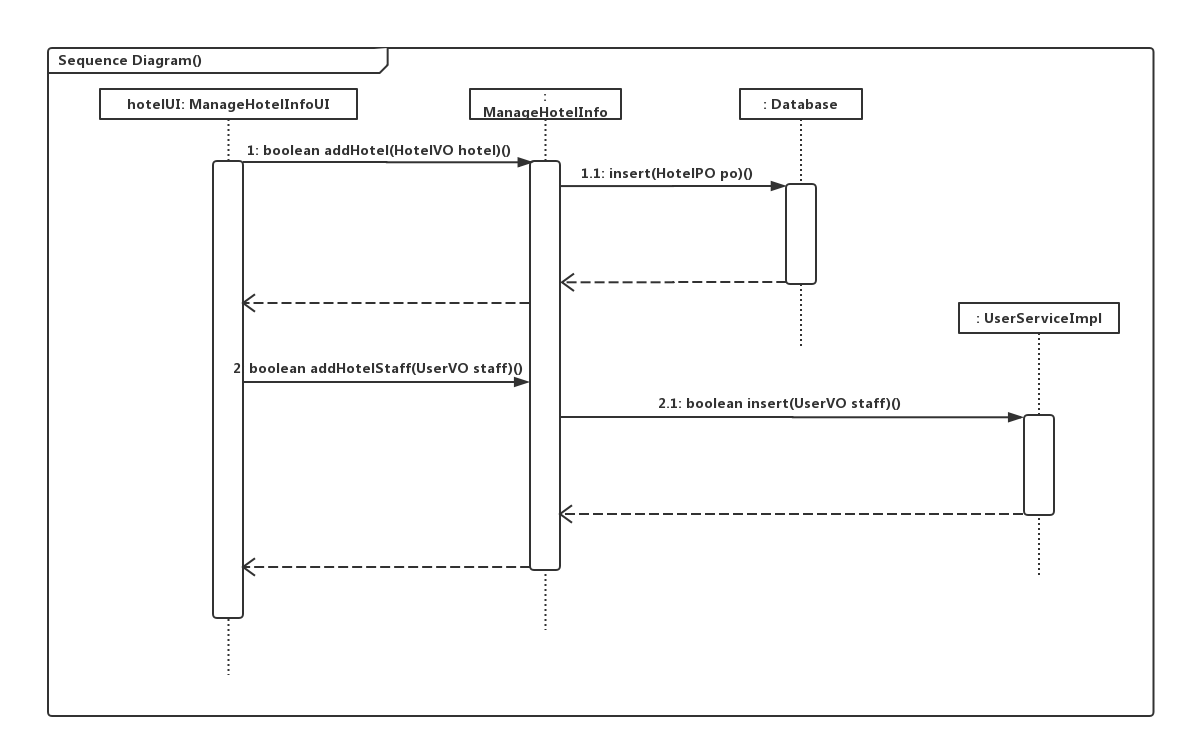
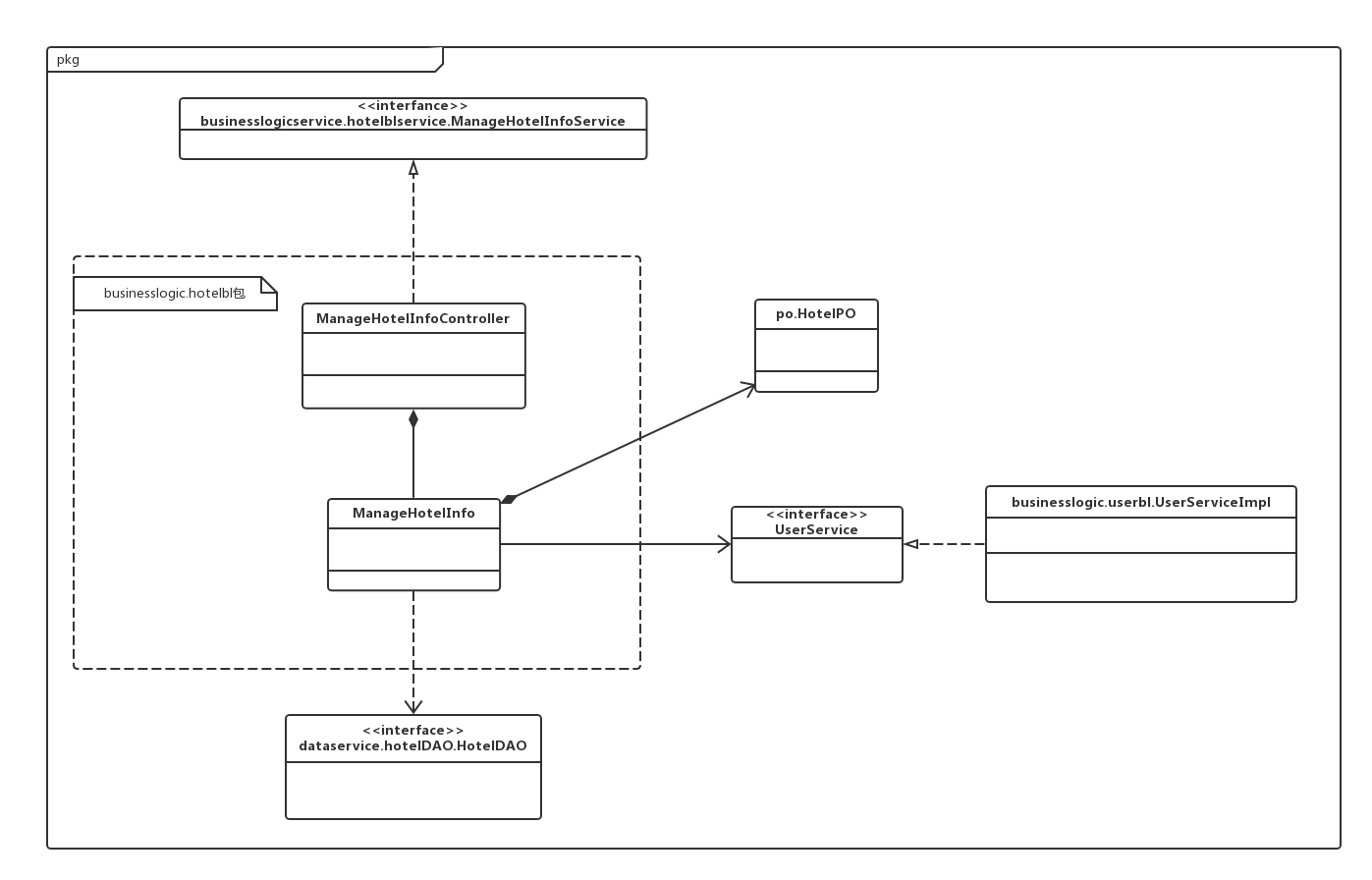
* + - 1. **ImportNewRoom**

****

* + - 1. **MaintainHotelBasicInfo**



* + - 1. **ManageHotelInfo**



* + 1. **roombl模块**

1. **模块概述**

roombl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。

roombl模块的职责及接口参见软件体系结构文档。

1. **整体结构**

展示层和业务逻辑层之间，我们添加

businesslogicservice.roomblservice.BrowseSpareRoomService接口、businesslogicservice.roomblservice.UpdateCheckInService接口、businesslogicservice.roomblservice. UpdateCheckOutService接口。业务逻辑层和数据层之间添加dataservice.roomDAO.RoomDAO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BroseSpareRoomServiceImpl, UpdateCheckInServiceImpl, UpdateCheckOutServiceImpl类。RoomPO、CheckInOutPO是作为销售记录的持久化对象被添加到涉及模型中去的。SpareRoomList和SpareRoomItem的添加是SpareRoom的容器类、CheckInList和CheckInItem的添加是CheckIn的容器类、CheckOutList和CheckOutItem的添加是CheckOut的容器类。RoomInfoService都是根据依赖倒置原则，为了消除循环依赖而产生的接口。

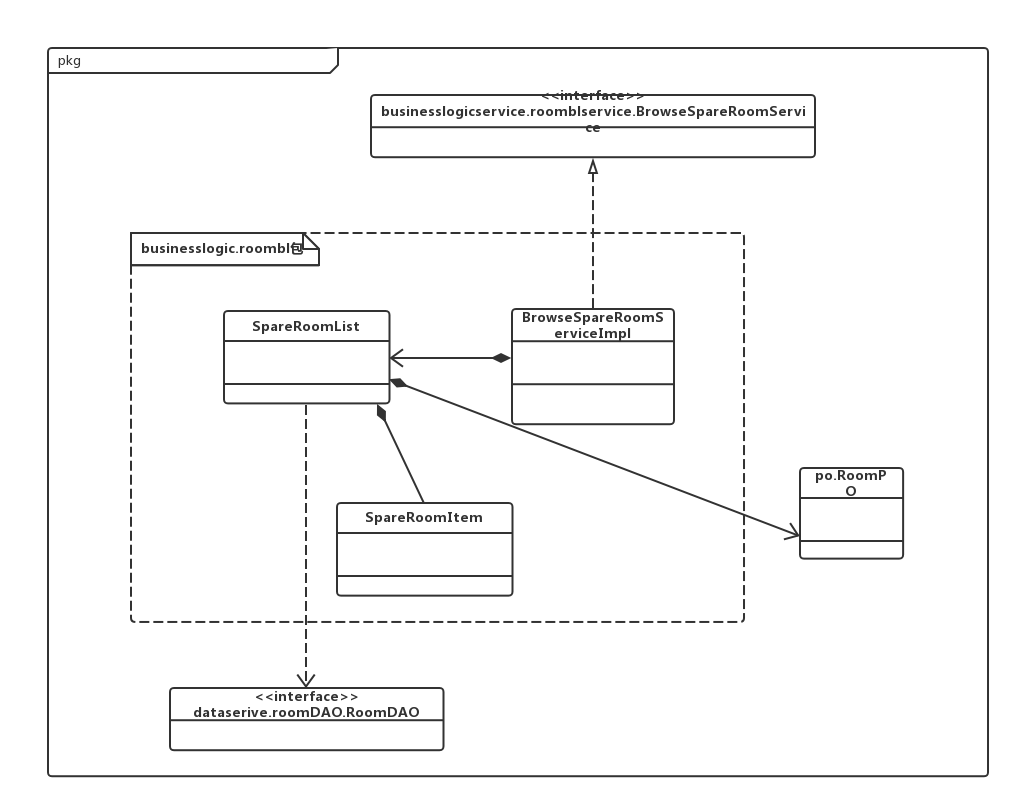
roombl模块各个类的职责如下表所示。

roombl模块各个类的职责

|  |  |
| --- | --- |
| **类名** | **职责** |
| browseSpareRoom.BrowseSpareRoomServiceImpl | 负责实现对应于浏览空房所需要的服务 |
| updateCheckIn.UpdateCheckInServiceImpl | 负责实现对应于更新入住信息所需要实现的服务 |
| updateCheckOut.UpdateCheckOutServiceImpl | 负责实现对应于更新退房信息所需要实现的服务 |
| browseSpareRoom.SpareRoomList | 空房列表对象，拥有空房对象引用，可以提供空房对象的列表 |
| browseSpareRoom.SpareRoomItem | 空房对象，拥有空房类型，价格，数量，可以提供空房信息 |
| updateCheckIn.CheckInList | 入住信息列表对象，拥有入住信息对象引用，可以提供入住信息列表，提供搜索，增加，修改，删除，验证入住信息的服务 |
| updateCheckIn.CheckInItem | 入住信息对象，拥有房间类型，数量，入住时间和预计离开时间，帮助完成增加，修改，删除，验证入住信息的服务 |
| updateCheckOut.CheckOutList | 退房信息列表对象，拥有退房信息对象引用，可以提供退房信息列表，提供搜索，增加，修改，删除，验证退房信息的服务 |
| updateCheckOut.CheckOutItem | 退房信息对象，拥有房间类型，数量，实际离开时间，帮助完成增加，修改，删除，验证退房信息的服务 |
| RoomInfoServiceImpl | 给同层的其他模块提供有关空房的服务 |

* + - 1. **BrowseSpareRoom**

browseSpareRoom模块的设计如下图所示。



BrowseSpareRoomServiceImpl, SpareRoomList和SpareRoomItem的接口规范如下表所示。

BrowseSpareRoomServiceImpl的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| BrowseSpareRoomServiceImpl.getRoomInfoList | 语法 | public ArrayList<RoomVO> getRoomInfoList (String address) |
| 前置条件 | 已创建一个SpareRoomList对象 |
| 后置条件 | 调用SpareRoomList对象的getRoomInfoList方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| SpareRoomList.getRoomInfoList(String address) | 提供空房列表 | |
| SpareRoomItem.toVO() | 把SpareRoomItem转换成RoomVO类型 | |

SpareRoomList的接口规范

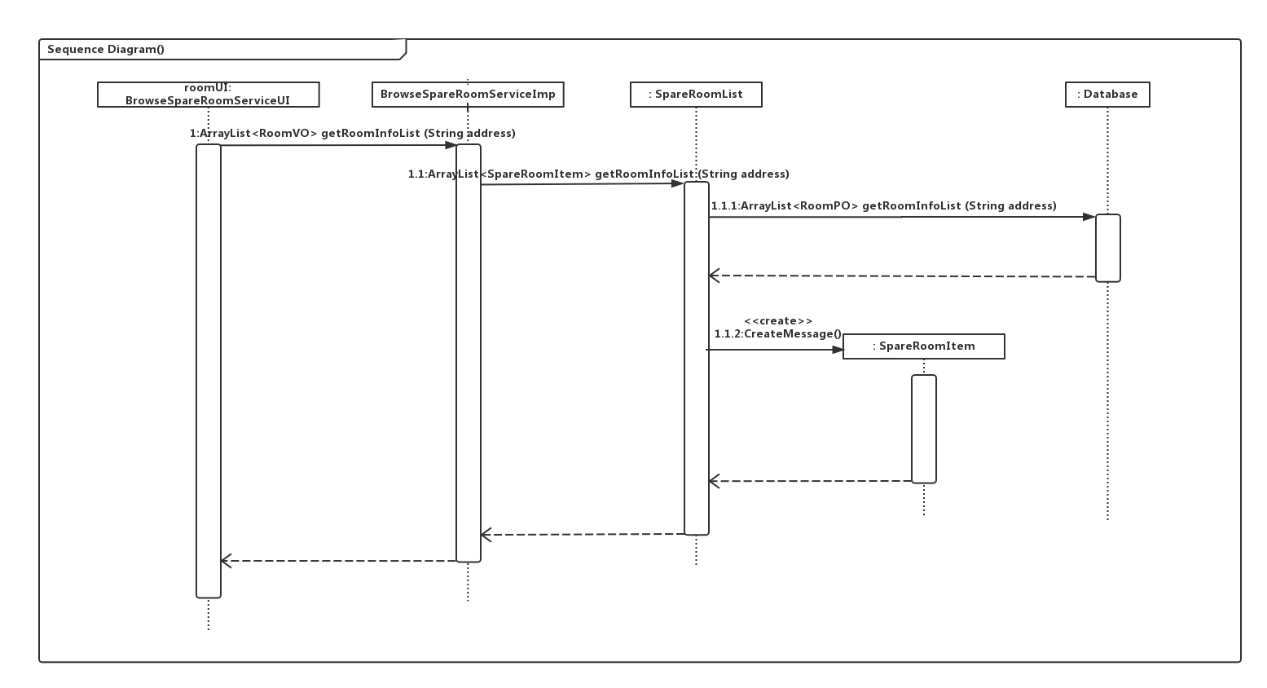
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SpareRoomList.getRoomInfoList | 语法 | public ArrayList<SpareRoomItem> getRoomInfoList (String address) |
| 前置条件 | 启动一个得到空房列表回合 |
| 后置条件 | 在一个得到空房列表回合中，返回空房列表 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| RoomDAO. getSpareRoomInfoList(String address) | 得到空房列表持久化对象 | |

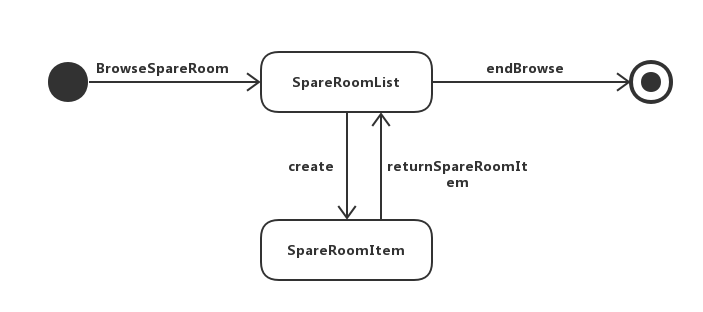
SpareRoomItem的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SpareRoomItem.toVO | 语法 | public RoomVO toVO() |
| 前置条件 | 无 |
| 后置条件 | 返回空房RoomVO |

BrowseSpareRoom的动态模型

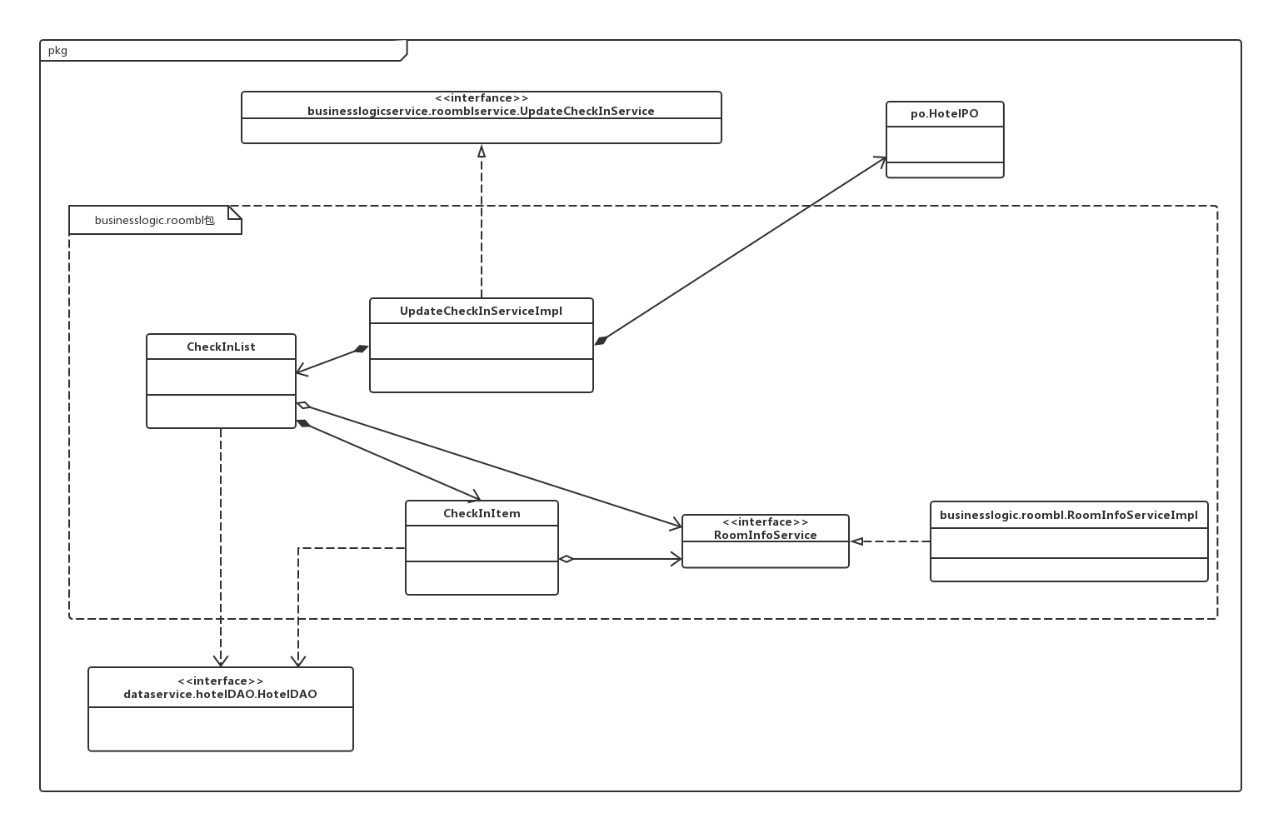
BrowseSpareRoom的顺序图如下图所示。

 BrowseSpareRoom的状态图如下图所示。



* + - 1. **UpdateCheckIn**

UpdateCheckIn模块的设计如下图所示。



UpdateCheckInServiceImpl, CheckInList和CheckInItem的接口规范如下表所示。

UpdateCheckInServiceImpl的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UpdateCheckInServiceImpl. getCheckInList | 语法 | public ArrayList<RoomVO> getCheckInList(String address) |
| 前置条件 | 已创建一个CheckInList对象 |
| 后置条件 | 调用CheckInList对象的getRoomInfoList方法 |
| UpdateCheckInServiceImpl. searchCheckInInfo | 语法 | public ArrayList<RoomVO> searchCheckInInfo(String address ,Date time) |
| 前置条件 | 已创建一个CheckInList对象 |
| 后置条件 | 调用CheckInList对象的searchCheckInInfo方法 |
| UpdateCheckInServiceImpl. searchCheckInInfo | 语法 | public ArrayList<RoomVO> searchCheckInInfo(String address ,Enum<RoomType> roomType) |
| 前置条件 | 已创建一个CheckInList对象 |
| 后置条件 | 调用CheckInList对象的searchCheckInInfo方法 |
| UpdateCheckInServiceImpl. addCheckIn | 语法 | public boolean addCheckIn(String address, RoomVO checkIn) |
| 前置条件 | 已创建一个CheckInList对象 |
| 后置条件 | 调用CheckInList对象的addCheckIn方法 |
| UpdateCheckInServiceImpl. modifyCheckIn | 语法 | public boolean modifyCheckIn(String address, RoomVO checkIn) |
| 前置条件 | 已创建一个CheckInList对象 |
| 后置条件 | 调用CheckInList对象的modifyCheckIn方法 |
| UpdateCheckInServiceImpl.delCheckIn | 语法 | public boolean delCheckIn(String address, RoomVO checkIn) |
| 前置条件 | 已创建一个CheckInList对象 |
| 后置条件 | 调用CheckInList对象的delCheckIn方法 |
| UpdateCheckInServiceImpl.validCheckIn | 语法 | public boolean validCheckIn(String address, RoomVO checkIn) |
| 前置条件 | 已创建一个CheckInList对象 |
| 后置条件 | 调用CheckInList对象的validCheckIn方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| CheckInList.getCheckInList( String address) | 提供入住信息列表 | |
| CheckInItem.toVO() | 把SpareRoomItem转换成RoomVO类型 | |
| CheckInList. searchCheckInInfo | 提供根据入住时间搜索得到的入住信息列表 | |
| CheckInList. searchCheckInInfo | 提供根据房间类型搜索得到的入住信息列表 | |
| CheckInList. addCheckIn | 提供增加入住信息的服务 | |
| CheckInList. modifyCheckIn | 提供修改入住信息的服务 | |
| CheckInList. delCheckIn | 提供删除入住信息的服务 | |
| CheckInList. validCheckIn | 提供验证入住信息的服务 | |

CheckInList的接口规范

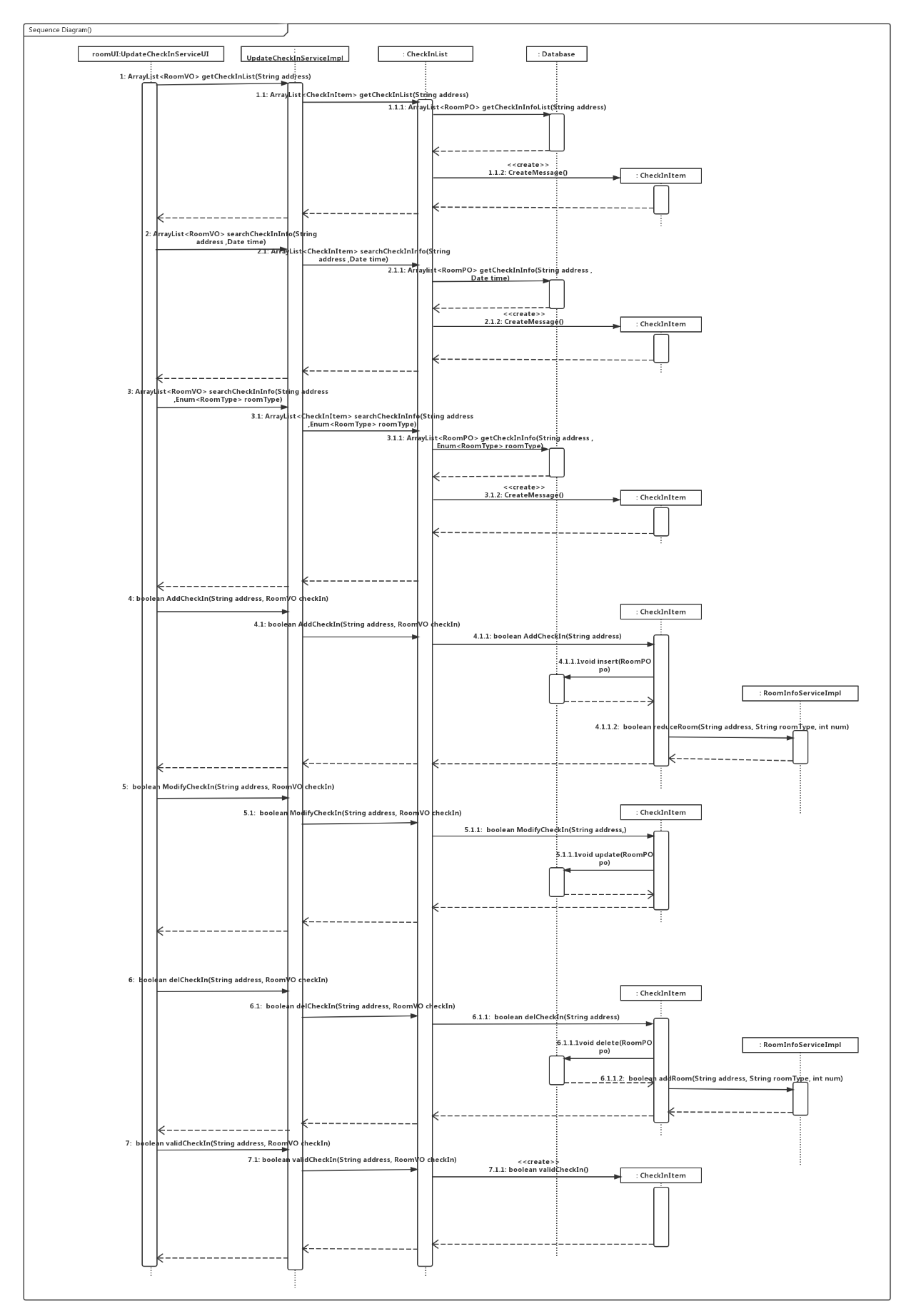
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CheckInList.getCheckInList | 语法 | public ArrayList<CheckInItem> getCheckInList(String address) |
| 前置条件 | 启动一个得到入住信息列表回合 |
| 后置条件 | 在一个得到入住信息列表回合中，返回入住信息列表 |
| CheckInList. searchCheckInInfo | 语法 | public ArrayList<CheckInItem> searchCheckInInfo(String address ,Date time) |
| 前置条件 | 启动一个搜索入住信息回合 |
| 后置条件 | 在一个搜索入住信息回合中，返回搜索到的入住信息列表 |
| CheckInList. searchCheckInInfo | 语法 | public ArrayList<CheckInItem> searchCheckInInfo(String address ,Enum<RoomType> roomType) |
| 前置条件 | 启动一个搜索入住信息回合 |
| 后置条件 | 在一个搜索入住信息回合中，返回搜索到的入住信息列表 |
| CheckInList. addCheckIn | 语法 | public boolean addCheckIn(String address, RoomVO checkIn) |
| 前置条件 | 已创建一个CheckInItem对象 |
| 后置条件 | 调用CheckInItem对象的addCheckIn方法 |
| CheckInList. modifyCheckIn | 语法 | public boolean modifyCheckIn(String address, RoomVO checkIn) |
| 前置条件 | 已创建一个CheckInItem对象 |
| 后置条件 | 调用CheckInItem对象的modifyCheckIn方法 |
| CheckInList. delCheckIn | 语法 | public boolean delCheckIn(String address, RoomVO checkIn) |
| 前置条件 | 已创建一个CheckInItem对象 |
| 后置条件 | 调用CheckInItem对象的delCheckIn方法 |
| CheckInList. validCheckIn | 语法 | public boolean validCheckIn(String address, RoomVO checkIn) |
| 前置条件 | 已创建一个CheckInItem对象 |
| 后置条件 | 调用CheckInItem对象的validCheckIn方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| CheckInItem.toVO() | 把SpareRoomItem转换成RoomVO类型 | |
| RoomDAO. getCheckInInfo(String address , Date time) | 得到对应入住时间的入住信息列表 | |
| RoomDAO. getCheckInInfo(String address ,Enum<RoomType> roomType) | 得到对应房间类型的入住信息列表 | |
| CheckInItem.addCheckIn | 提供增加入住信息的服务 | |
| CheckInItem. modifyCheckIn | 提供修改入住信息的服务 | |
| CheckInItem. delCheckIn | 提供删除入住信息的服务 | |
| CheckInItem. validCheckIn | 提供验证入住信息的服务 | |

CheckInItem的接口规范

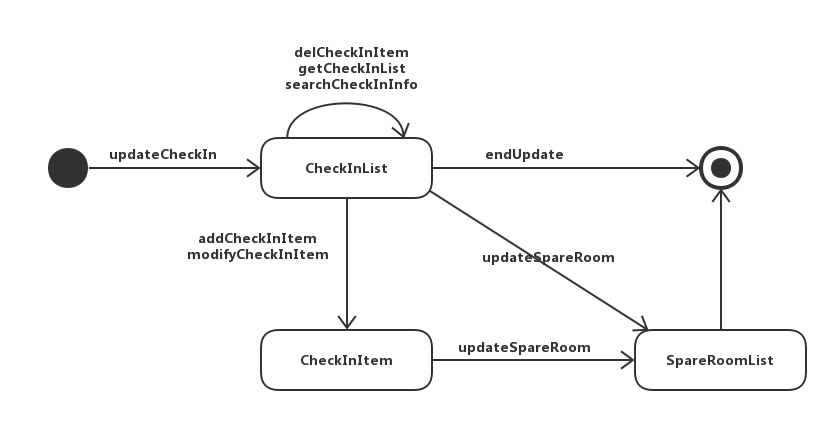
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CheckInItem.toVO | 语法 | public RoomVO toVO() |
| 前置条件 | 无 |
| 后置条件 | 返回入住信息CheckInOutVO |
| CheckInItem.addCheckIn | 语法 | boolean addCheckIn(String address) |
| 前置条件 | 启动一个增加入住信息回合 |
| 后置条件 | 返回是否增加成功 |
| CheckInItem. modifyCheckIn | 语法 | boolean modifyCheckIn(String address) |
| 前置条件 | 启动一个修改入住信息回合 |
| 后置条件 | 返回是否修改成功 |
| CheckInItem. delCheckIn | 语法 | boolean delCheckIn(String address) |
| 前置条件 | 启动一个删除入住信息回合 |
| 后置条件 | 返回是否删除成功 |
| CheckInItem. validCheckIn | 语法 | boolean validCheckIn(String address) |
| 前置条件 | 启动一个验证入住信息回合 |
| 后置条件 | 返回是否验证成功 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| RoomDAO.insert(RoomPO po) | 插入单一持久化对象 | |
| RoomDAO.update(RoomPO po) | 更新单一持久化对象 | |
| RoomDAO.delete(RoomPO po) | 删除单一持久化对象 | |

UpdateCheckIn的动态模型

UpdateCheckIn的顺序图如下图所示。

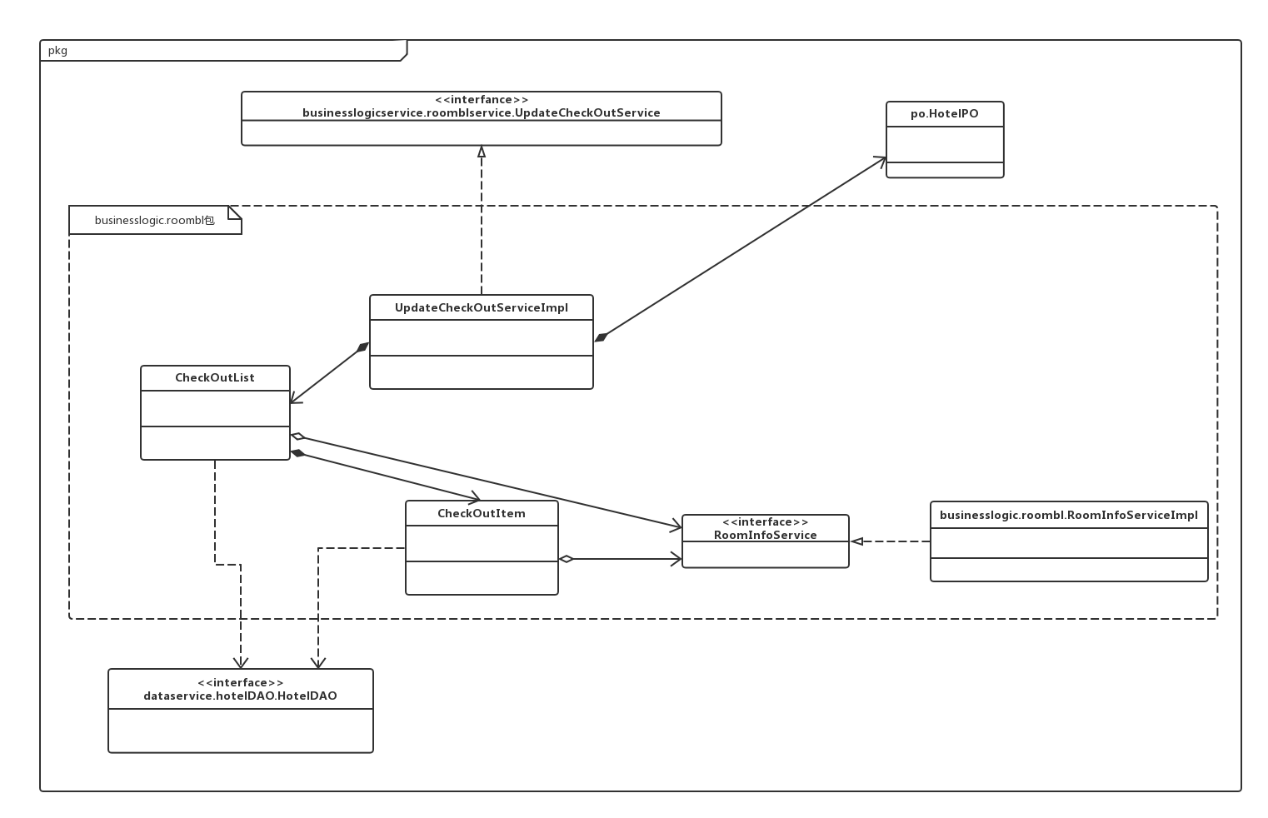


UpdateCheckIn的状态图如下图所示。



* + - 1. **UpdateCheckOut**

UpdateCheckOut模块的设计如下图所示。



UpdateCheckOutServiceImpl, CheckOutList和CheckOutItem的接口规范如下表所示。

UpdateCheckOutServiceImpl的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UpdateCheckOutServiceImpl. getCheckOutList | 语法 | public ArrayList<RoomVO> getCheckOutList(String address) |
| 前置条件 | 已创建一个CheckOutList对象 |
| 后置条件 | 调用CheckOutList对象的getRoomInfoList方法 |
| UpdateCheckOutServiceImpl. searchCheckOutInfo | 语法 | public ArrayList<RoomVO> searchCheckOutInfo(String address ,Date time) |
| 前置条件 | 已创建一个CheckOutList对象 |
| 后置条件 | 调用CheckOutList对象的searchCheckOutInfo方法 |
| UpdateCheckOutServiceImpl. searchCheckOutInfo | 语法 | public ArrayList<RoomVO> searchCheckOutInfo(String address ,Enum<RoomType> roomType) |
| 前置条件 | 已创建一个CheckOutList对象 |
| 后置条件 | 调用CheckOutList对象的searchCheckOutInfo方法 |
| UpdateCheckOutServiceImpl. addCheckOut | 语法 | public boolean addCheckOut(String address, RoomVO CheckOut) |
| 前置条件 | 已创建一个CheckOutList对象 |
| 后置条件 | 调用CheckOutList对象的addCheckOut方法 |
| UpdateCheckOutServiceImpl. modifyCheckOut | 语法 | public boolean modifyCheckOut(String address, RoomVO CheckOut) |
| 前置条件 | 已创建一个CheckOutList对象 |
| 后置条件 | 调用CheckOutList对象的modifyCheckOut方法 |
| UpdateCheckOutServiceImpl.delCheckOut | 语法 | public boolean delCheckOut(String address, RoomVO CheckOut) |
| 前置条件 | 已创建一个CheckOutList对象 |
| 后置条件 | 调用CheckOutList对象的delCheckOut方法 |
| UpdateCheckOutServiceImpl.validCheckOut | 语法 | public boolean validCheckOut(String address, RoomVO CheckOut) |
| 前置条件 | 已创建一个CheckOutList对象 |
| 后置条件 | 调用CheckOutList对象的validCheckOut方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| CheckOutList.getCheckOutList( String address) | 提供退房信息列表 | |
| CheckOutItem.toVO() | 把SpareRoomItem转换成RoomVO类型 | |
| CheckOutList. searchCheckOutInfo | 提供根据实际离开时间搜索得到的退房信息列表 | |
| CheckOutList. searchCheckOutInfo | 提供根据房间类型搜索得到的退房信息列表 | |
| CheckOutList. addCheckOut | 提供增加退房信息的服务 | |
| CheckOutList. modifyCheckOut | 提供修改退房信息的服务 | |
| CheckOutList. delCheckOut | 提供删除退房信息的服务 | |
| CheckOutList. validCheckOut | 提供验证退房信息的服务 | |

CheckOutList的接口规范

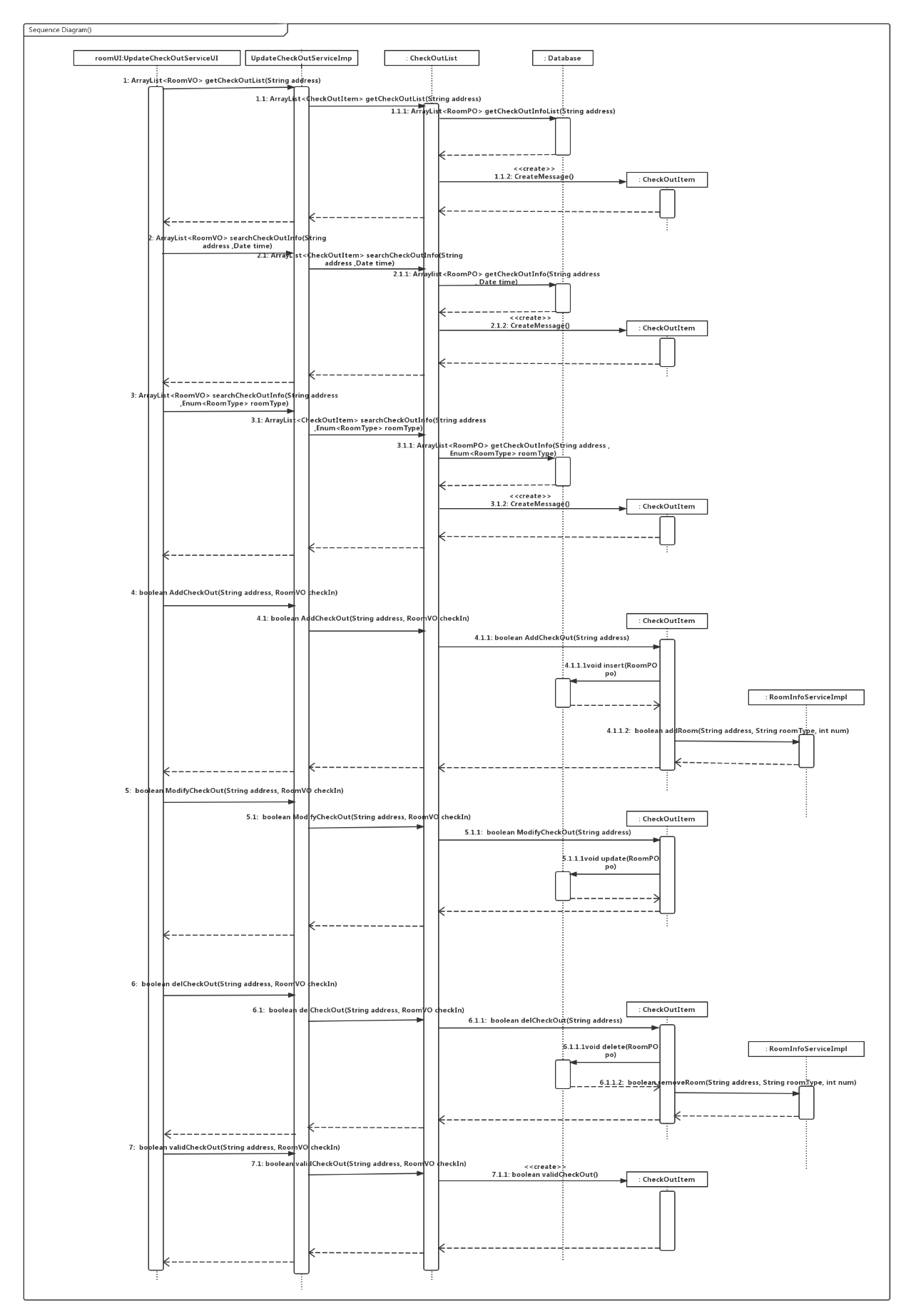
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CheckOutList.getCheckOutList | 语法 | public ArrayList<CheckOutItem> getCheckOutList(String address) |
| 前置条件 | 启动一个得到退房信息列表回合 |
| 后置条件 | 在一个得到退房信息列表回合中，返回退房信息列表 |
| CheckOutList. searchCheckOutInfo | 语法 | public ArrayList<CheckOutItem> searchCheckOutInfo(String address ,Date time) |
| 前置条件 | 启动一个搜索退房信息回合 |
| 后置条件 | 在一个搜索退房信息回合中，返回搜索到的退房信息列表 |
| CheckOutList. searchCheckOutInfo | 语法 | public ArrayList<CheckOutItem> searchCheckOutInfo(String address ,Enum<RoomType> roomType) |
| 前置条件 | 启动一个搜索退房信息回合 |
| 后置条件 | 在一个搜索退房信息回合中，返回搜索到的退房信息列表 |
| CheckOutList. addCheckOut | 语法 | public boolean addCheckOut(String address, RoomVO CheckOut) |
| 前置条件 | 已创建一个CheckOutItem对象 |
| 后置条件 | 调用CheckOutItem对象的addCheckOut方法 |
| CheckOutList. modifyCheckOut | 语法 | public boolean modifyCheckOut(String address, RoomVO CheckOut) |
| 前置条件 | 已创建一个CheckOutItem对象 |
| 后置条件 | 调用CheckOutItem对象的modifyCheckOut方法 |
| CheckOutList. delCheckOut | 语法 | public boolean delCheckOut(String address, RoomVO CheckOut) |
| 前置条件 | 已创建一个CheckOutItem对象 |
| 后置条件 | 调用CheckOutItem对象的delCheckOut方法 |
| CheckOutList. validCheckOut | 语法 | public boolean validCheckOut(String address, RoomVO CheckOut) |
| 前置条件 | 已创建一个CheckOutItem对象 |
| 后置条件 | 调用CheckOutItem对象的validCheckOut方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| CheckOutItem.toVO() | 把SpareRoomItem转换成RoomVO类型 | |
| RoomDAO. getCheckOutInfo(String address , Date time) | 得到对应退房时间的退房信息列表 | |
| RoomDAO. getCheckOutInfo(String address ,Enum<RoomType> roomType) | 得到对应房间类型的退房信息列表 | |
| CheckOutItem.addCheckOut | 提供增加退房信息的服务 | |
| CheckOutItem. modifyCheckOut | 提供修改退房信息的服务 | |
| CheckOutItem. delCheckOut | 提供删除退房信息的服务 | |
| CheckOutItem. validCheckOut | 提供验证退房信息的服务 | |

CheckOutItem的接口规范

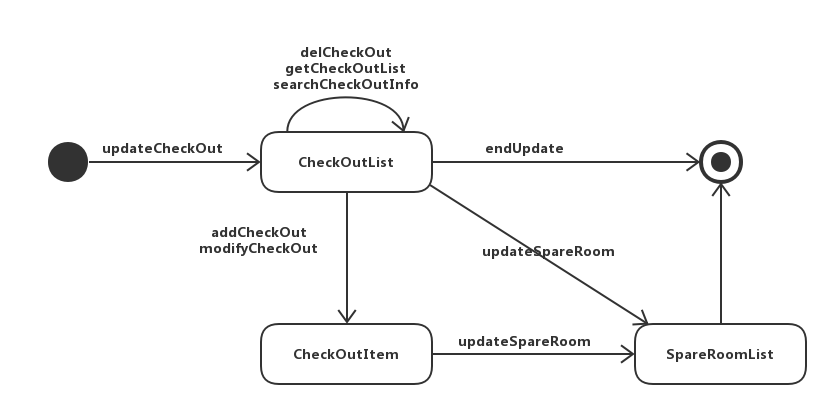
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CheckOutItem.toVO | 语法 | public RoomVO toVO() |
| 前置条件 | 无 |
| 后置条件 | 返回退房信息CheckInOutVO |
| CheckOutItem.addCheckOut | 语法 | boolean addCheckOut(String address) |
| 前置条件 | 启动一个增加退房信息回合 |
| 后置条件 | 返回是否增加成功 |
| CheckOutItem. modifyCheckOut | 语法 | boolean modifyCheckOut(String address) |
| 前置条件 | 启动一个修改退房信息回合 |
| 后置条件 | 返回是否修改成功 |
| CheckOutItem. delCheckOut | 语法 | boolean delCheckOut(String address) |
| 前置条件 | 启动一个删除退房信息回合 |
| 后置条件 | 返回是否删除成功 |
| CheckOutItem. validCheckOut | 语法 | boolean validCheckOut(String address) |
| 前置条件 | 启动一个验证退房信息回合 |
| 后置条件 | 返回是否验证成功 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| RoomDAO.insert(RoomPO po) | 插入单一持久化对象 | |
| RoomDAO.update(RoomPO po) | 更新单一持久化对象 | |
| RoomDAO.delete(RoomPO po) | 删除单一持久化对象 | |

UpdateCheckOut的动态模型

UpdateCheckOut的顺序图如下图所示。



UpdateCheckOut的状态图如下图所示。



* + 1. **strategybl模块**

**（1）模块概述**

strategybl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。

strategybl模块的职责及接口参见软件体系结构文档。

**（2）整体结构**

展示层和业务逻辑层之间，我们添加businesslogicservice. strategyblservice.UpdateStrategyService接口。业务逻辑层和数据层之间添加dataservice.strategyDAO.StrategyDAO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UpdateStrategyServiceImpl类。StrategyPO是作为销售记录的持久化对象被添加到涉及模型中去的。StrategyList和StrategyItem的添加是Strategy的容器类StrategyInfoService都是根据依赖倒置原则，为了消除循环依赖而产生的接口。

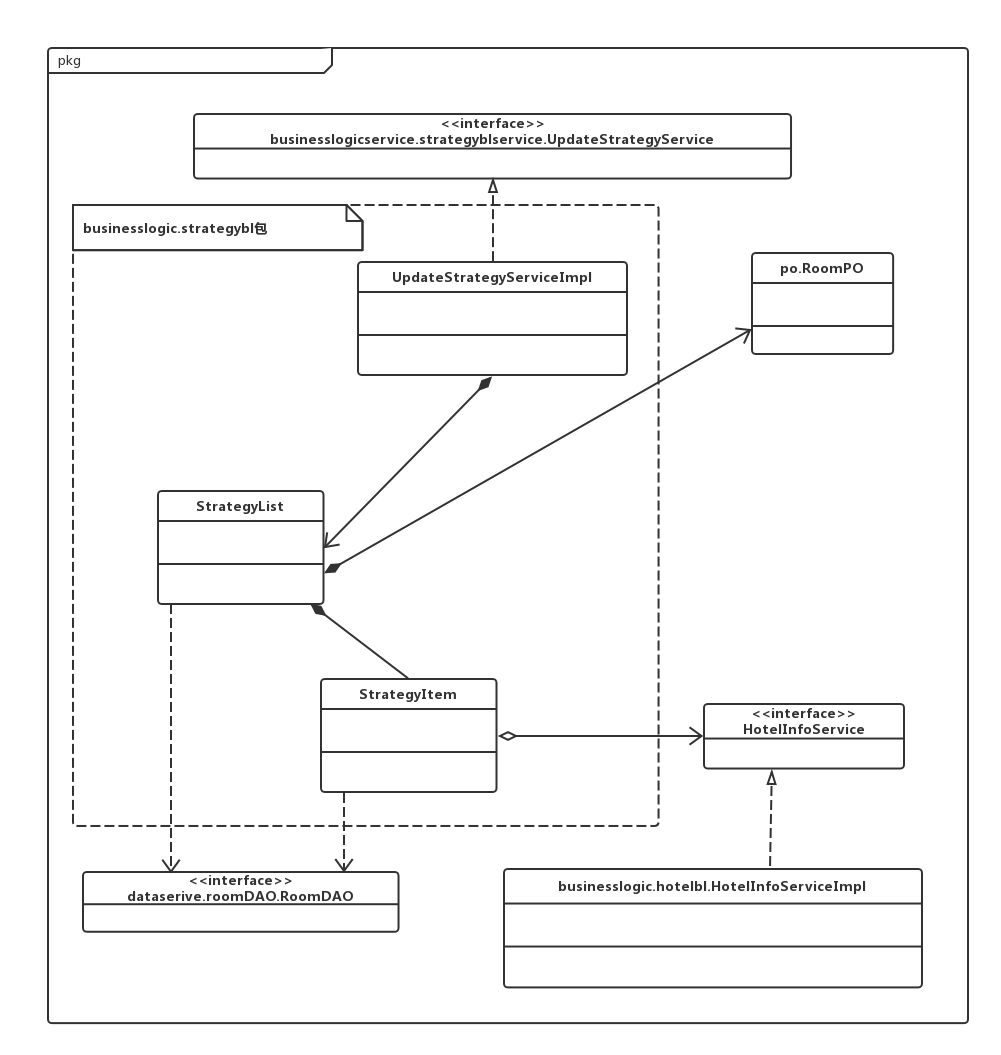
strategybl模块各个类的职责如下表所示。

Strategybl模块各个类的职责

|  |  |
| --- | --- |
| **类名** | **职责** |
| updateStrategy.UpdateStrategyServiceImpl | 负责实现对应于更新促销和营销策略所需要实现的服务 |
| updateStrategy.StrategyList | 策略列表对象，拥有策略对象引用，可以提供某种策略类型的策略列表，提供搜索，增加，修改，删除，验证策略信息的服务 |
| updateStrategy.StrategyItem | 策略对象，拥有酒店地址，折扣类型，折扣名称，折扣百分比，帮助完成增加，修改，删除，验证策略信息的服务 |
| StrategyInfoServiceImpl | 为同层的其他模块提供有关策略信息的服务 |

**4.1.5.1** **UpdateStrategy**

UpdateStrategy模块的设计如下图所示。



UpdateStrategyServiceImpl, StrategyList和StrategyItem的接口规范如下表所示。

UpdateStrategyServiceImpl的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UpdateStrategyServiceImpl. getStrategyList | 语法 | public ArrayList<RoomVO> getStrategyList(String address , Enum<StrategyType> StrategyType) |
| 前置条件 | 已创建一个StrategyList对象 |
| 后置条件 | 调用StrategyList对象的getStrategyList方法 |
| UpdateStrategyServiceImpl. getStrategyInfo | 语法 | public ArrayList<RoomVO> getStrategyInfo(String address , String name) |
| 前置条件 | 已创建一个StrategyList对象 |
| 后置条件 | 调用StrategyList对象的getStrategyInfo方法 |
| UpdateStrategyServiceImpl. add | 语法 | public boolean add(String address, StrategyVO strategy) |
| 前置条件 | 已创建一个StrategyList对象 |
| 后置条件 | 调用StrategyList对象的add方法 |
| UpdateStrategyServiceImpl. modify | 语法 | public boolean add(String address, StrategyVO strategy) |
| 前置条件 | 已创建一个StrategyList对象 |
| 后置条件 | 调用StrategyList对象的modify方法 |
| UpdateStrategyServiceImpl.delete | 语法 | public boolean delete (String address,StrategyVO Strategy) |
| 前置条件 | 已创建一个StrategyList对象 |
| 后置条件 | 调用StrategyList对象的del方法 |
| UpdateStrategyServiceImpl.valid | 语法 | public boolean valid (String address, StrategyVO Strategy) |
| 前置条件 | 已创建一个StrategyList对象 |
| 后置条件 | 调用StrategyList对象的valid方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| StrategyList.getStrategyList( String address , Enum<StrategyType> strategyType) | 提供某种策略类型的策略列表 | |
| StrategyItem.toVO() | 把StrategyItem转换成StrategyVO类型 | |
| StrategyList.getStrategyInfo(String address, String name) | 提供根据折扣名称搜索到的策略 | |
| StrategyList. add | 提供增加策略的服务 | |
| StrategyList. modify | 提供修改策略的服务 | |
| StrategyList. delete | 提供删除策略的服务 | |
| StrategyList. valid | 提供验证策略的服务 | |

StrategyList的接口规范

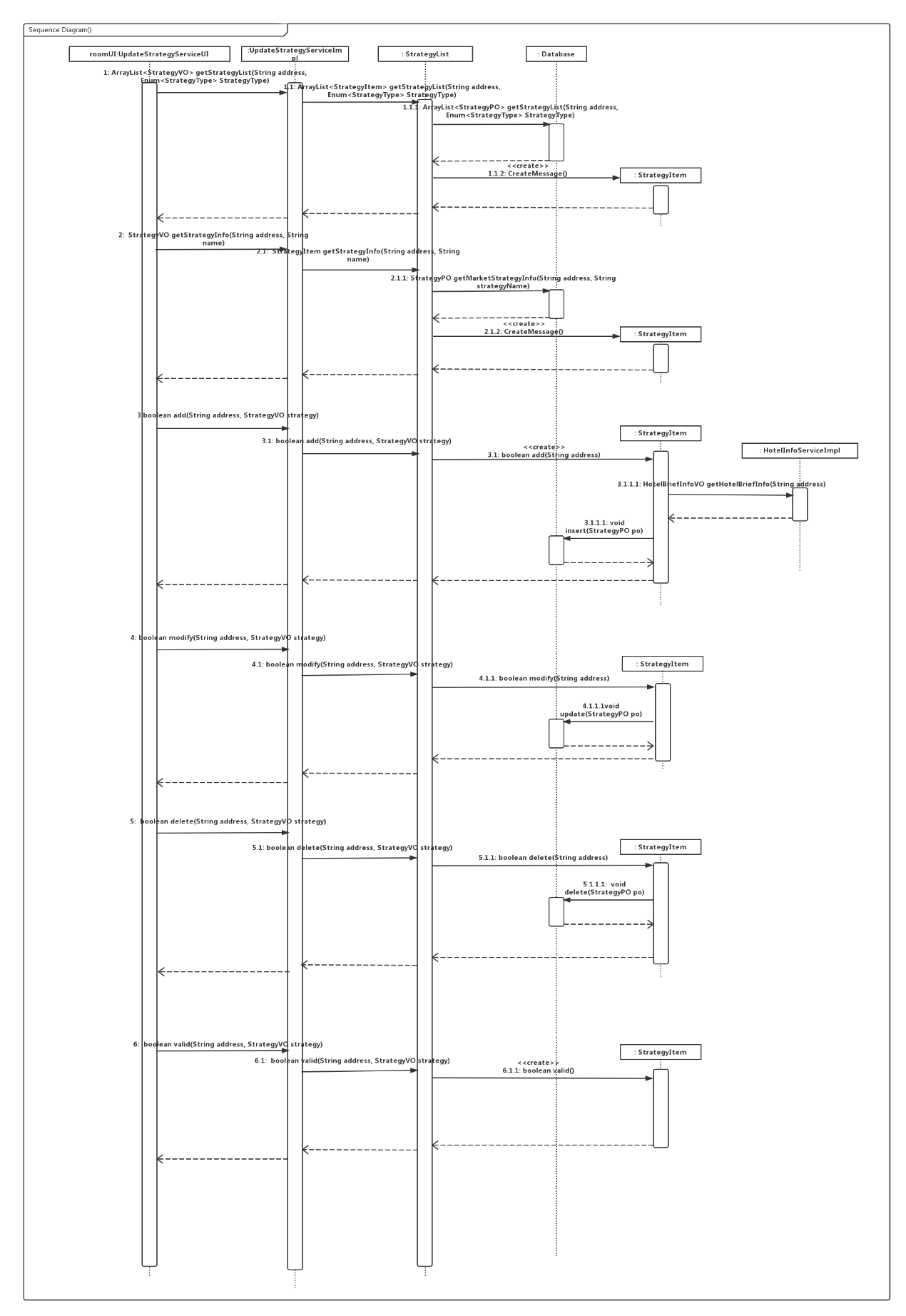
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StrategyList.getStrategyList | 语法 | public ArrayList<StrategyItem> getStrategyList(String address, Enum<StrategyType> strategyType) |
| 前置条件 | 启动一个得到策略列表回合 |
| 后置条件 | 在一个得到策略列表回合中，返回策略列表 |
| StrategyList. getStrategyInfo | 语法 | public StrategyItem getStrategyInfo(String address, String name) |
| 前置条件 | 启动一个搜索策略回合 |
| 后置条件 | 在一个搜索策略回合中，返回搜索到的策略 |
| StrategyList. add | 语法 | public boolean add(String address, StrategyVO strategyVO) |
| 前置条件 | 已创建一个StrategyItem对象 |
| 后置条件 | 调用StrategyItem对象的add方法 |
| StrategyList. modify | 语法 | public boolean modify(String address, StrategyVO strategyVO) |
| 前置条件 | 已创建一个StrategyItem对象 |
| 后置条件 | 调用StrategyItem对象的modify方法 |
| StrategyList. delete | 语法 | public Boolean delete(String address, StrategyVO strategyVO) |
| 前置条件 | 已创建一个StrategyItem对象 |
| 后置条件 | 调用StrategyItem对象的delete方法 |
| StrategyList. valid | 语法 | public boolean valid(String address, StrategyVO strategyVO) |
| 前置条件 | 已创建一个StrategyItem对象 |
| 后置条件 | 调用StrategyItem对象的valid方法 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| StrategyItem.toVO() | 把SpareRoomItem转换成RoomVO类型 | |
| StrategyDAO.getMarketStrategyInfo(String address, String strategyName) | 得到对应策略名称的策略信息 | |
| StrategyItem.add | 提供增加策略的服务 | |
| StrategyItem. modify | 提供修改策略的服务 | |
| StrategyItem. del | 提供删除策略的服务 | |
| StrategyItem. valid | 提供验证策略的服务 | |

StrategyItem的接口规范

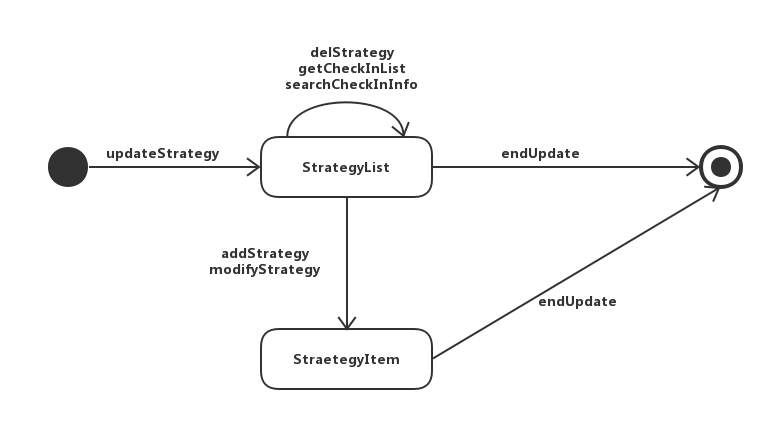
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StrategyItem.toVO | 语法 | public RoomVO toVO() |
| 前置条件 | 无 |
| 后置条件 | 返回策略信息StrategyVO |
| StrategyItem.add | 语法 | boolean add(String address) |
| 前置条件 | 启动一个增加策略回合 |
| 后置条件 | 返回是否增加成功 |
| StrategyItem. modify | 语法 | boolean modify(String address) |
| 前置条件 | 启动一个修改策略回合 |
| 后置条件 | 返回是否修改成功 |
| StrategyItem. del | 语法 | boolean delete(String address) |
| 前置条件 | 启动一个删除策略回合 |
| 后置条件 | 返回是否删除成功 |
| StrategyItem. valid | 语法 | boolean valid(String address) |
| 前置条件 | 启动一个验证策略回合 |
| 后置条件 | 返回是否验证成功 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| StrategyDAO.insert(StrategyPO po) | 插入单一持久化对象 | |
| StrategyDAO.update(StrategyPO po) | 更新单一持久化对象 | |
| StrategyDAO.delete(StrategyPO po) | 删除单一持久化对象 | |

UpdateStrategy的动态模型

UpdateStrategy的顺序图如下图所示。



UpdateStrategy的状态图如下图所示。



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。

