### 3.3 HotelUI包所对应的Controller

#### 3.3.1 HotelController

图3.3.1-1 HotelController 类图

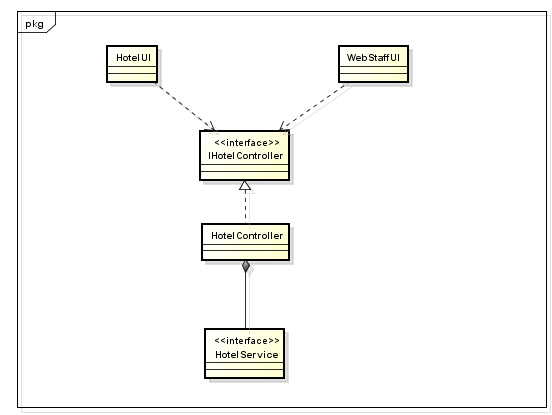


表3.3.1 -2 HotelController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| HotelController.findHotelByID | 语法 | public HotelVO. findHotelByID(int HotelID) | |
| 前置条件 | 已知酒店ID，需要返回该酒店的详细信息 | |
| 后置条件 | 无 | |
| HotelController.updateHotel | 语法 | public ResultMessage updateHotel(Hotel VO hotelvo) | |
| 前置条件 | 已知更新后的信息，需要将其添加到数据库 | |
| 后置条件 | 在数据库中更新酒店信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| HotelService.update(HotelVO) | | | 更新单一持久化序列 |
| HotelService.findByID(int HotelID) | | | 查找单一持久化序列 |

#### 3.3.2 HotelOrderController

图3.3.2-1 HotelOrderController 类图

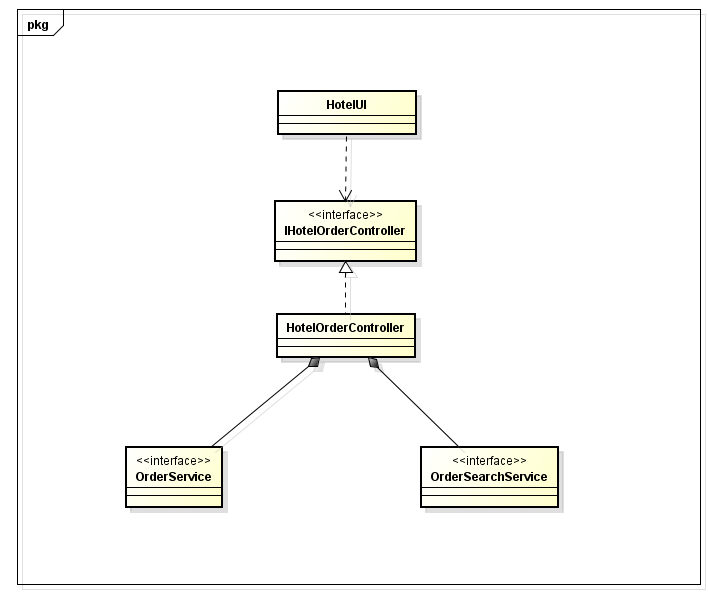


表3.3.2 -2 HotelOrderController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| HotelOrderController.findOrderByHotelAndType | 语法 | Public List<OrderVO> findOrderByHotel-AndType(int HotelID,OrderStatus type) | |
| 前置条件 | 已知酒店ID和订单状态，需要返回该酒店特定状态的所有订单的详细信息 | |
| 后置条件 | 无 | |
| HotelOrderController.findOrderByHotelAndUsername | 语法 | Public List<OrderVO> findOrderByHotel-AndUsername(String username,int HotelID) | |
| 前置条件 | 已知酒店ID和订单所有者用户名，需要返回该用户在此酒店的所有订单的详细信息 | |
| 后置条件 | 无 | |
| HotelOrderController.findOrderByID | 语法 | Public OrderVO findOrderByID(int id) | |
| 前置条件 | 已知订单ID，需要返回该订单的详细信息 | |
| 后置条件 | 无 | |
| HotelOrderController.checkin | 语法 | Public ResultMessage checkout(OrderVO ordervo) | |
| 前置条件 | 已知订单信息，需要进行入住登记 | |
| 后置条件 | 在数据库中更新订单信息 | |
| HotelOrderController.revoke | 语法 | Public ResultMessage revoke(OrderVO or-dervo) | |
| 前置条件 | 已知订单信息，需要进行订单撤销 | |
| 后置条件 | 在数据库中删除订单信息 | |
| HotelOrderController.delayCheckin | 语法 | Public ResultMessage dealayChickin(Order-VO ordervo) | |
| 前置条件 | 已知订单信息，需要进行订单延迟入住 | |
| 后置条件 | 在数据库中更新订单信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| OrderService.checkout(OrderVO ordervo) | | | 更新单一持久化序列 |
| OrderService.checkin(OrderVO ordervo) | | | 更新单一持久化序列 |
| OrderService.revoke (OrderVO ordervo) | | | 更新单一持久化序列 |
| OrderService.delayCheckin(OrderVO ordervo) | | | 更新单一持久化序列 |
| OrderSearchService.findByID(int OrderID) | | | 查找单一持久化序列 |
| OrderSearchService.findByHotelAndType(int HotelID,OrderStatus type) | | | 查找单一持久化序列 |
| OrderSearchService. findByHotelAndUsername (String username,int HotelID) | | | 查找单一持久化序列 |

#### 3.3.3 HotelDiscountController

图3.3.3-1 HotelDiscountController 类图

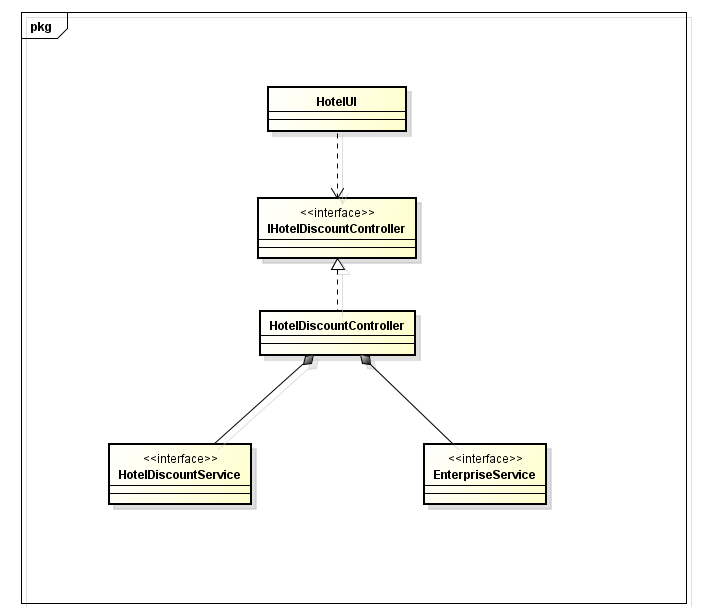


表3.3.3 -2 HotelDiscountController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| HotelDiscountController.findAllByHotelID | 语法 | Public List<HotelDiscountVO> findAllBy-HotelID(int HotelID) | |
| 前置条件 | 已知酒店ID，需要返回该酒店所有优惠策略的详细信息 | |
| 后置条件 | 无 | |
| HotelDiscountController.findNotAddedType | 语法 | Public List<HotelDiscountType> findNotA-ddedType (int HotelID) | |
| 前置条件 | 已知酒店ID，需要返回该酒店尚未制定的优惠策略类型 | |
| 后置条件 | 无 | |
| HotelDiscountController.getAllEnterprises | 语法 | Public List<EnterpriseVO> getAllEnterprises () | |
| 前置条件 | 返回所有企业信息 | |
| 后置条件 | 无 | |
| HotelDiscountController.delete | 语法 | Public ResultMessage delete(int id) | |
| 前置条件 | 已知优惠策略id，需要删除该优惠策略 | |
| 后置条件 | 在数据库中更新优惠策略信息 | |
| HotelDiscountController.add | 语法 | Public ResultMessage add(HotelDiscount-VO hoteldiscountvo) | |
| 前置条件 | 已知优惠策略信息，需要进行优惠策略的添加 | |
| 后置条件 | 在数据库中添加优惠策略信息 | |
| HotelDiscountController.update | 语法 | Public ResultMessage update(HotelDisc-ountVO hoteldiscountvo) | |
| 前置条件 | 已知优惠策略信息，需要修改该优惠策略 | |
| 后置条件 | 在数据库中更新优惠策略信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| HotelDiscountService. add(HotelDiscountVO hot-eldiscountvo) | | | 更新单一持久化序列 |
| HotelDiscountService. update(HotelDiscountVO hoteldiscountvo) | | | 更新单一持久化序列 |
| HotelDiscountService. delete(int id) | | | 更新单一持久化序列 |
| HotelDiscountService. findAllByHotelID(int Hote lID) | | | 查找单一持久化序列 |
| HotelDiscountService. findNotAddedType (int Ho telID) | | | 查找单一持久化序列 |
| EnterpriseService. getAllEnterprises() | | | 查找单一持久化序列 |

### 2.2 Order模块的静态结构与动态行为

#### 2.2.1 OrderSearchService

##### 2.2.1.1 局部模块的职责

（1）模块概述

OrderSearchService模块承担的需求参见需求规格说明文档功能需求及相关的非功能需求。OrderSearchService模块的职责及接口参见软件系统结构描述文档表5.3.2-2

（2）整体结构

OrderSearchService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。OrderSearchService模块的职责及接口参见软件体系结构描述文档。根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了OrderSearchService, OrderDAO两个接口。OrderPO是做为管理信息的持久化对象被添加到设计模型中的。

图2.2.1.1-1 OrderSearchService模块设计

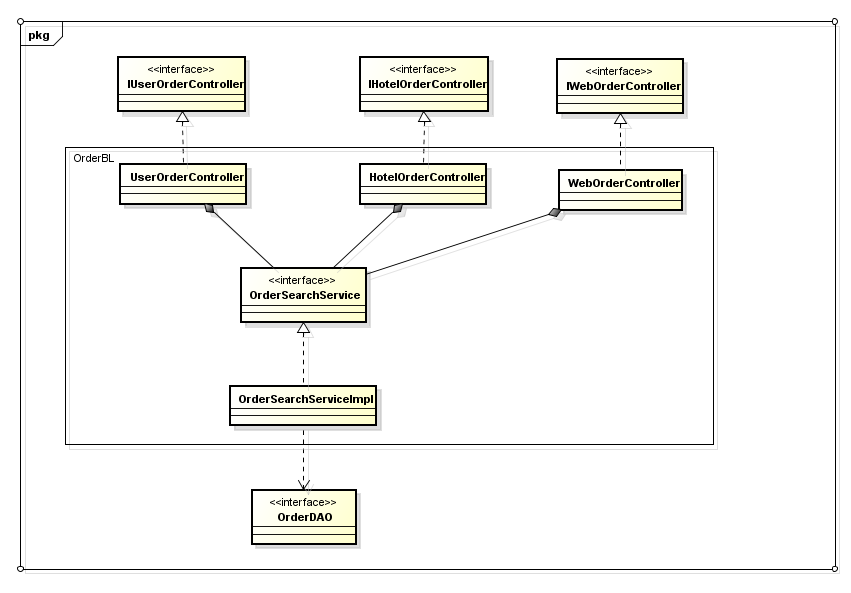


表2.2.1.1-2 OrderSearchServic模块各个类的职责

|  |  |
| --- | --- |
| 类 | 职责 |
| UserOrderController | 负责实现对应用户订单信息界面所需的方法 |
| HotelOrderController | 负责实现对应酒店订单信息界面所需的方法 |
| WebOrderController | 负责实现对应网站订单信息界面所需的方法 |
| OrderSearchServiceImpl | 负责处理有关订单搜索的功能 |

##### 2.2.1.2 局部模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| OrderSearchService.findByID | 语法 | public OrderVO findByID(int id) | |
| 前置条件 | 已知该订单的ID，需要得到该订单的详细信息 | |
| 后置条件 | 返回该订单的详细信息 | |
| OrderSearchService.findByUsernameAndType | 语法 | public List<OrderVO>  findByUsernameAndType(String username,OrderStatus) | |
| 前置条件 | 已知订单的所属用户和订单类型，需要获得符合条件的订单列表 | |
| 后置条件 | 无 | |
| OrderSearchService. findByUsername | 语法 | public List<OrderVO> findByUsername(String username) | |
| 前置条件 | 已知订单的所属用户，需要获得该用户的所有订单 | |
| 后置条件 | 无 | |
| OrderSearchService.findByHotelAndUsername | 语法 | public List<OrderVO> findByHotelAndUsername(int hotelID,String username) | |
| 前置条件 | 已知订单的所属酒店和所属用户，需要获得所有订单 | |
| 后置条件 | 无 | |
| OrderSearchService. findByOrderType | 语法 | public List<OrderVO> findByOrderType(OrderStatus) | |
| 前置条件 | 已知订单的类型，需要获得所有该类型的订单 | |
| 后置条件 | 无 | |
| OrderSearchService. findByHotelAndTime | 语法 | public List<OrderVO> findByHotelAndTime(int hotelID,Date begin,Date end) | |
| 前置条件 | 已知酒店和时间，需要得到符合该条件的所有订单 | |
| 后置条件 | 无 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| OrderDAO. findByID(int ID) | | | 查找单一持久化序列 |
| OrderDAO. findByUsernameAndType(String username,OrderStatus) | | | 查找单一持久化列表 |
| OrderDAO. findByUsername(String username) | | | 查找单一持久化列表 |
| OrderDAO. findByHotelAndUsername(int hotelID,String username | | | 查找单一持久化列表 |
| OrderDAO.findByOrderType(OrderStatus) | | | 查找单一持久化列表 |
| OrderDAO. findByHotelAndTime(int hotelID,Date begin,Date end); | | | 查找单一持久化列表 |

##### 2.2.1.3 局部模块的行为

图2.2.1.3-1 OrderSearchService模块时序图1

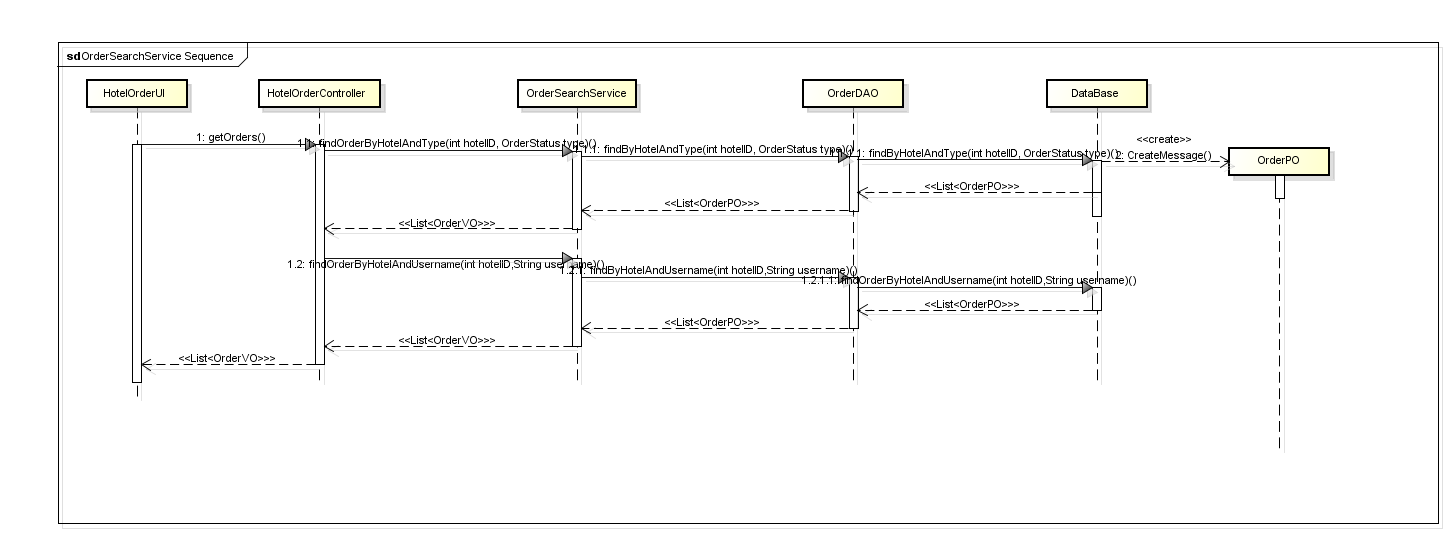


图2.2.1.3-1 表明了酒店管理系统中，当酒店管理人员查询订单信息时，订单业务逻辑处理的相关对象之间的协作。

图2.2.1.3-2 OrderSearchService模块时序图2

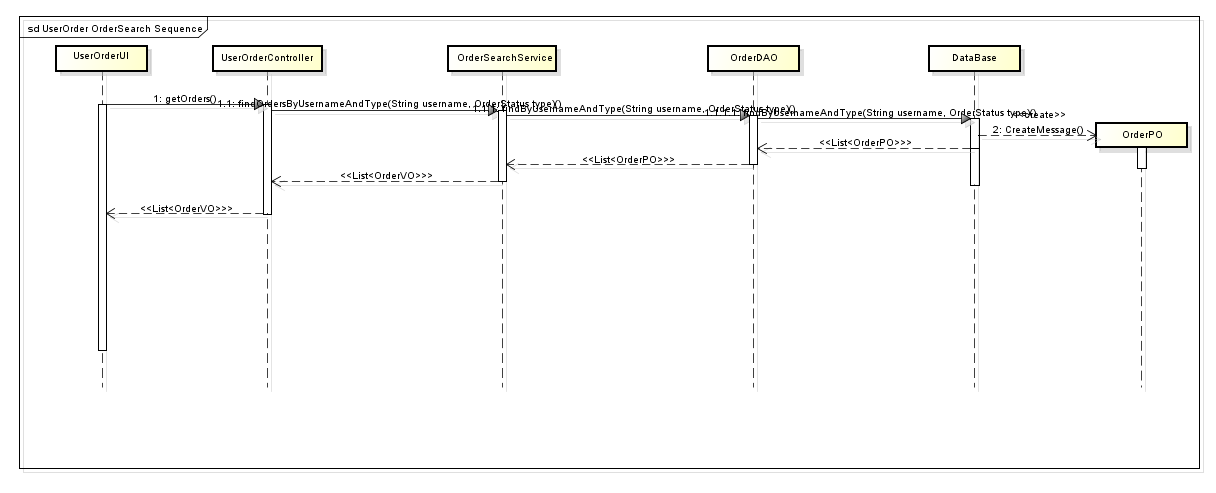


图2.2.1.3-2 表明了酒店管理系统中，当用户查询订单信息时，订单业务逻辑处理的相关对象之间的协作。

图2.2.1.3-3 OrderSearchService模块时序图2

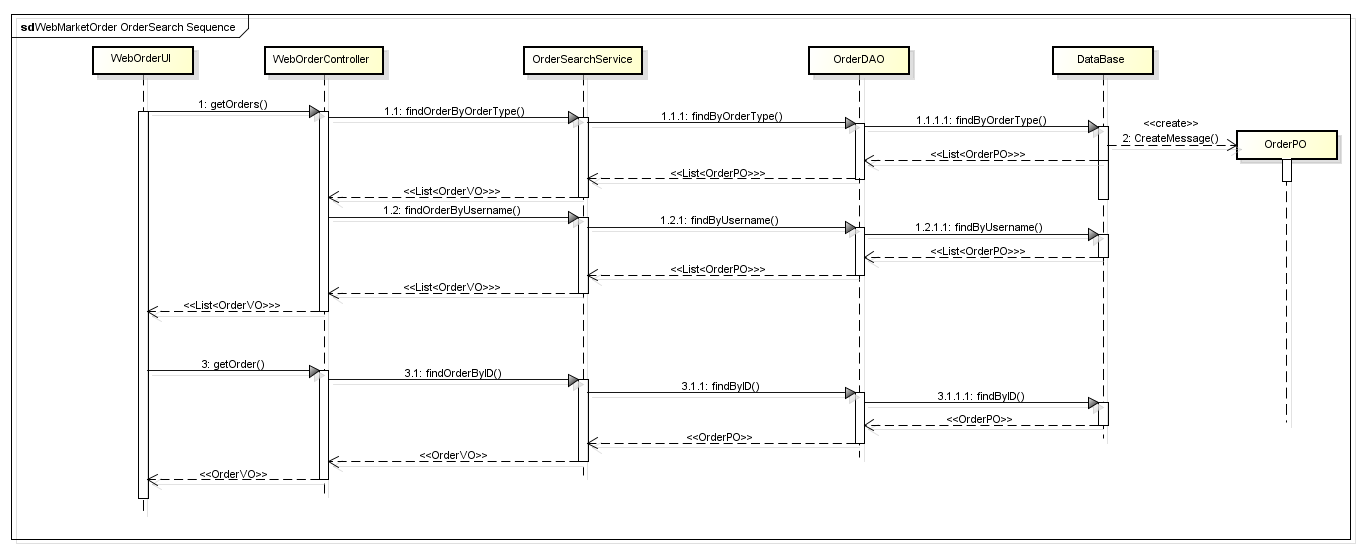
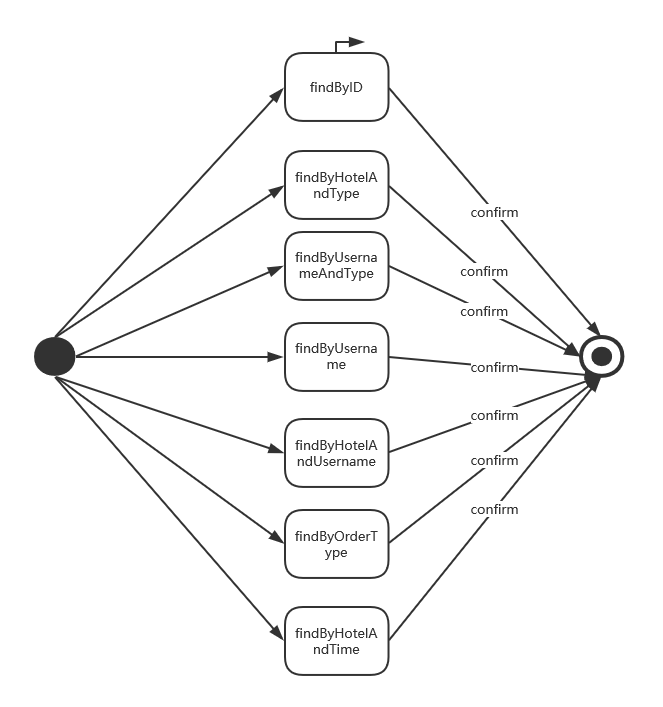


图2.2.1.3-3 表明了酒店管理系统中，当网站营销人员查询订单信息时，订单业务逻辑处理的相关对象之间的协作。

图2.2.1.3-4 OrderSearchService模块状态图



#### 2.2.2 OrderService

##### 2.2.2.1 局部模块的职责

（1）模块概述

OrderService模块承担的需求参见需求规格说明文档功能需求及相关的非功能需求。OrderService模块的职责及接口参见软件系统结构描述文档表5.3.2-2

（2）整体结构

OrderService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。OrderService模块的职责及接口参见软件体系结构描述文档。根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了OrderService, OrderDAO两个接口。OrderPO是做为管理信息的持久化对象被添加到设计模型中的。

图2.2.2.1-1OrderService模块设计

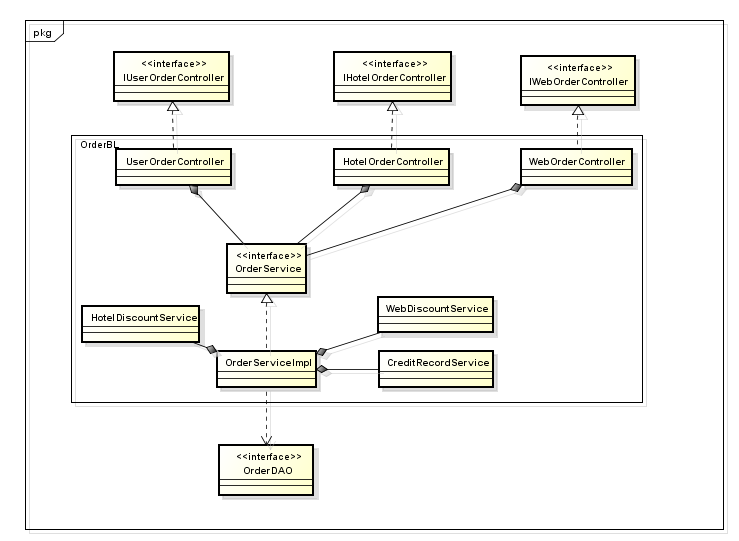


表2.2.2.1-2 OrderService模块各个类的职责

|  |  |
| --- | --- |
| 类 | 职责 |
| UserOrderController | 负责实现对应用户订单信息界面所需的方法 |
| HotelOrderController | 负责实现对应酒店管理界面所需的方法 |
| WebOrderController | 负责实现对应网站管理界面所需的方法 |
| OrderServiceImpl | 负责处理有关订单的功能 |

##### 2.2.2.2 局部模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| OrderService.  placeOrder | 语法 | public OrderVO placeOrder(OrderVO) | |
| 前置条件 | 已知该订单部分信息，需要将订单信息进行完善 | |
| 后置条件 | 更新订单对象的信息 | |
| OrderService.add | 语法 | public ResultMessage add(OrderVO) | |
| 前置条件 | 已知该订单信息，需要将其添加到数据库 | |
| 后置条件 | 在数据库中添加该订单信息 | |
| OrderService.checkin | 语法 | public ResultMessage checkin(OrderVO ) | |
| 前置条件 | 已知订单信息，需要将该订单状态改为已入住 | |
| 后置条件 | 在数据库中修改该订单状态 | |
| OrderService.checkout | 语法 | public ResultMessage checkout(OrderVO ordervo) | |
| 前置条件 | 已知订单信息，需要将该订单的状态改为已退房 | |
| 后置条件 | 在数据库中修改该订单状态 | |
| OrderService.revoke | 语法 | public ResultMessage revoke(OrderVO ) | |
| 前置条件 | 已知订单信息，需要将该订单的状态改为已撤销 | |
| 后置条件 | 在数据库中修改该订单状态 | |
| OrderService.remark | 语法 | public ResultMessage remark(OrderVO ) | |
| 前置条件 | 已知订单信息，需要增加该订单的评价信息 | |
| 后置条件 | 在数据库中修改该订单状态 | |
| OrderService.delayCheckin | 语法 | public ResultMessage delayCheckin(OrderVO ) | |
| 前置条件 | 已知订单信息，需要将该订单的状态改为已入住 | |
| 后置条件 | 在数据库中修改该订单状态 | |
| 需要的服务（需接口） | | | |
| OrderDAO. add(OrderPO) | | | 添加单一持久化列表 |
| OrderDAO. update(OrderPO) | | | 更新单一持久化序列 |
| HotelDiscountService.findAllByHotelID(int hotelID) | | | 返回该酒店的所有优惠策略 |
| CreditRecordService. add(CreditRecordVO) | | | 添加当前用户的一条信用记录 |
| WebDiscountService.findAll() | | | 返回网站的所有优惠策略 |

##### 2.2.2.3 局部模块的行为

图2.2.2.3-1 OrderSearchService模块时序图1

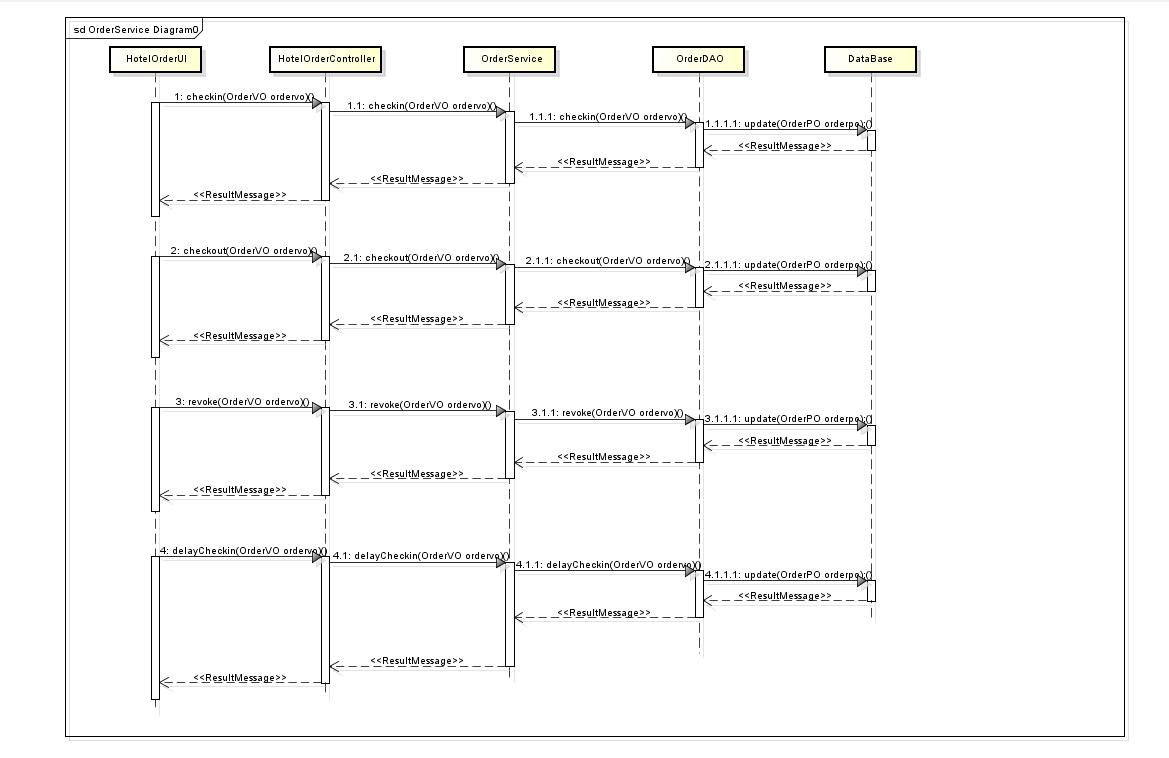


图2.2.2.3-1 表明了酒店管理系统中，当酒店管理人员进行订单管理时，订单业务逻辑处理的相关对象之间的协作。

图2.2.2.3-2 OrderSearchService模块时序图2

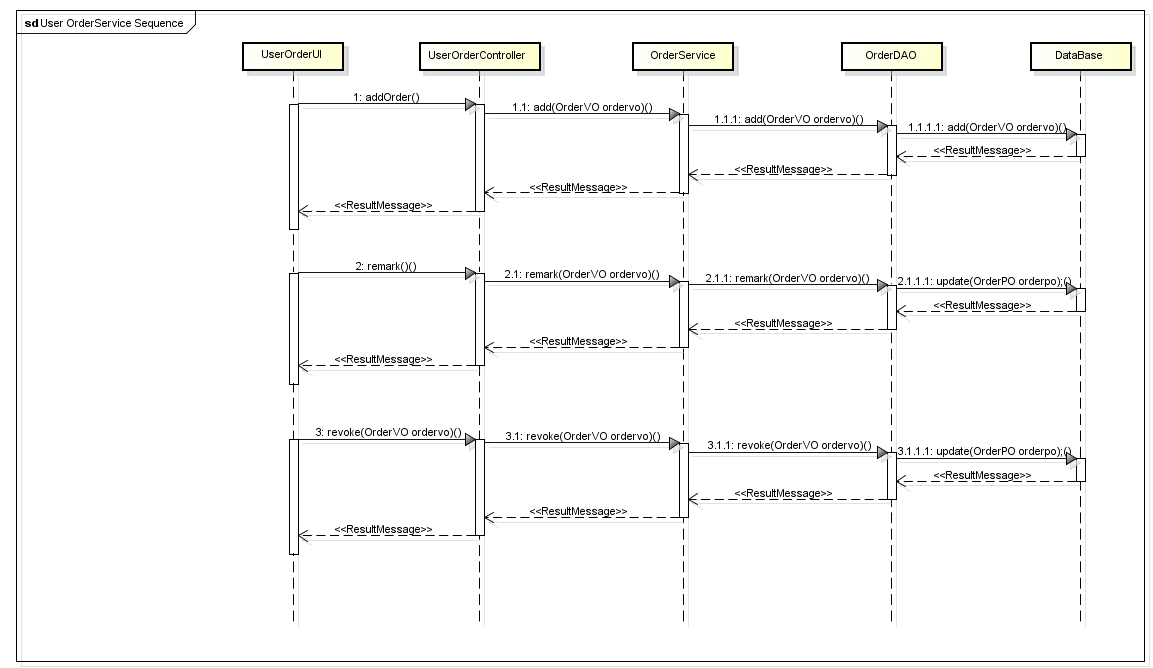


图2.2.2.3-2 表明了酒店管理系统中，当用户进行订单管理时，订单业务逻辑处理的相关对象之间的协作。

图2.2.2.3-3 OrderSearchService模块时序图3

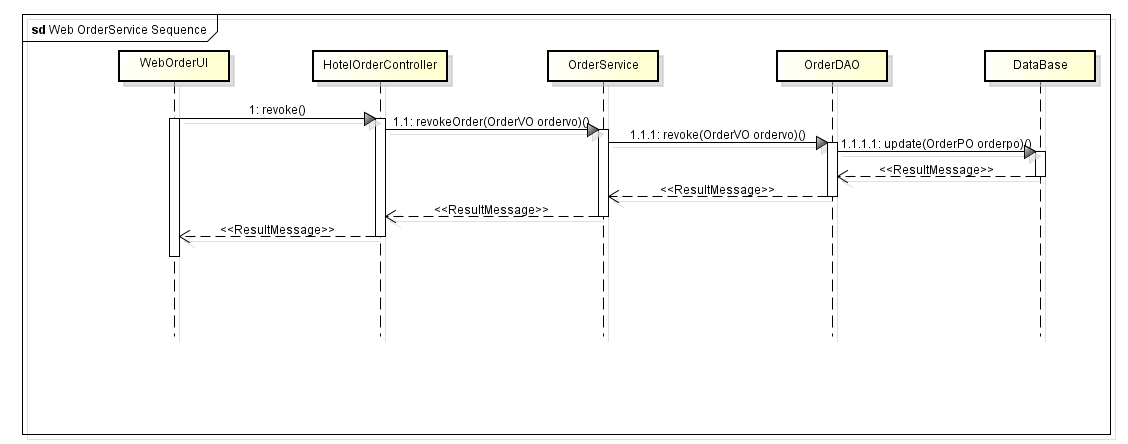
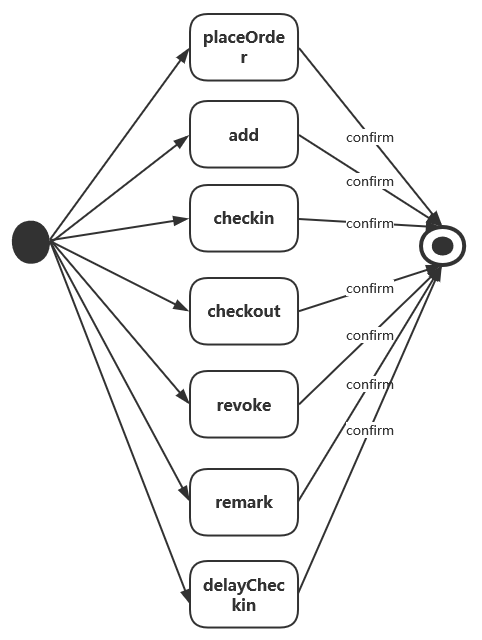


图2.2.2.3-3 表明了酒店管理系统中，当网站营销人员进行订单管理时，订单业务逻辑处理的相关对象之间的协作。

图2.2.1.3-4 OrdeService模块状态图



### 2.4 Room模块的静态结构与动态行为

#### 2.4.1 局部模块的职责

（1）模块概述

Room模块承担的需求参见需求规格说明文档功能需求及相关的非功能需求。Room模块的职责及接口参见软件系统结构描述文档表5.3.2.4

（2）整体结构

Room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。Room模块的职责及接口参见软件体系结构描述文档。根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了RoomService, RoomDAO 两个接口。RoomPO是做为管理信息的持久化对象被添加到设计模型中的。

图2.4.1-1 Room模块设计

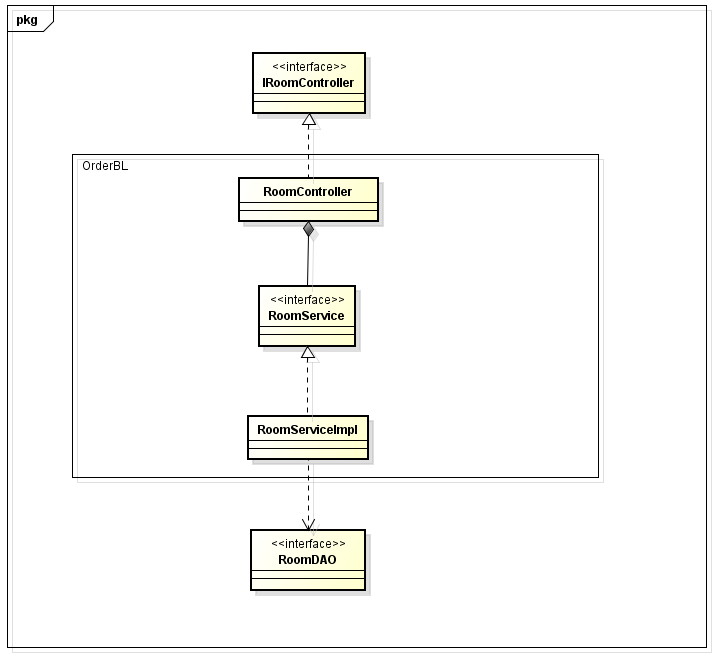


表2.4.1-2 Room模块各个类的职责

|  |  |
| --- | --- |
| 类 | 职责 |
| RoomController | 负责实现对应个人信息界面所需的方法 |
| RoomServiceImpl | 负责处理有关房间信息的功能 |

#### 2.4.2 局部模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| RoomService. findByHotelID | 语法 | public List<RoomVO> findByHotelID (int hotelID) | |
| 前置条件 | 已知酒店，需要房间详细信息列表 | |
| 后置条件 | 无 | |
| RoomService.findAvailableByHotelID | 语法 | public List<RoomVO> findAvailableByHotelID(int hotelID,Date begin,Date end) | |
| 前置条件 | 已知酒店和时间，需要在该时间段内有效的房间信息 | |
| 后置条件 | 无 | |
| RoomService.findNotAddedRoomType | 语法 | List<RoomType> findNotAddedRoomType(int hotelID) | |
| 前置条件 | 已知酒店编号，需要该酒店尚未添加过的房间类型列表 | |
| 后置条件 | 无 | |
| RoomService.update | 语法 | public ResultMessage update(RoomVO) | |
| 前置条件 | 已知房间更新后的信息，需要更新数据库中的信息 | |
| 后置条件 | 在数据库中更新房间信息 | |
| RoomService.add | 语法 | public ResultMessage add(RoomVO) | |
| 前置条件 | 已知房间信息，需要将其添加到数据库中 | |
| 后置条件 | 在数据库中保存新添加的房间信息 | |
| RoomService. findAvailableRoomNum | 语法 | public int findAvailableRoomNum(int hotelID,RoomType type,Date begin,Date end) | |
| 前置条件 | 已知酒店、房间类型和时间，需要可用房间数量 | |
| 后置条件 | 返回可用房间数量 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| RoomDAO. findByHotelAndType(int hotelID,RoomType) | | | 查找单一持久化序列 |
| RoomDAO. findByHotelID(int hotelID) | | | 查找单一持久化列表 |
| RoomDAO. add(RoomPO) | | | 增加单一持久化序列 |
| RoomDAO. update(RoomPO) | | | 更新单一持久化序列 |
| RoomDAO. findAvailableRoomNum (int hotelID,RoomType type,Date begin,Date end) | | | 返回在当前酒店和房间类型下一段时间内可用房间数量 |

#### 2.4.3 局部模块的行为

图2.4.3-1 RoomService模块时序图1

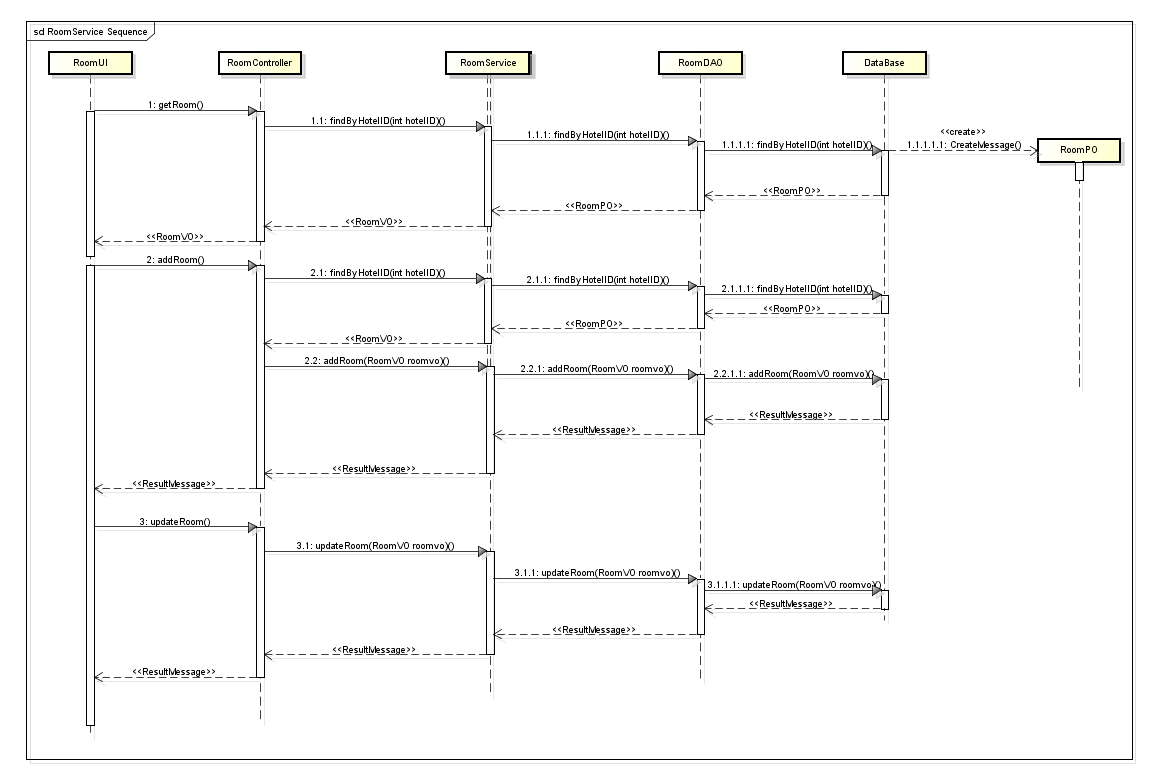


图24. 3-1 表明了酒店管理系统中，当酒店管理人员查询，修改，增加可用房间信息时时，订单业务逻辑处理的相关对象之间的协作。

图2.4.3-2 OrderSearchService模块状态图

