### hotel模块

1. 模块概述

hotel模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

hotel模块的职责和接口参见软件系统结构描述文档内对该模块的描述。

1. 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。在展示层和业务逻辑层之间添加hotelBLService接口。为了隔离业务逻辑职责和逻辑控制职责，添加hotelBLService，这样hotelBLService会依赖于ExecutedOrderInfo、UnexecutedOrderInfo、AbnormalOrderInfo三个接口实现对订单执行以及退房的业务逻辑，同时hotelBLService将对酒店基本信息维护、客房信息维护、剩余客房信息查看、酒店促销策略维护、线下处理的业务逻辑委托给hotel对象。HotelPO、HotelWorkerPO、RoomInfo、RemainRoomInfo是作为酒店信息的持久性对象被添加到设计模型中去的。

1. 模块内部类的接口规范
2. Controller类的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| HotelController.getHotelInfo | 语法 | Public HotelVO getHotelInfo(String userID) | | | | |
| 前置条件 | 已知酒店工作人员的编号，需要维护酒店基本信息 | | | | |
| 后置条件 | 系统返回该酒店工作人员对应的酒店基本信息 | | | | |
| HotelController.updateHotelInfo | 语法 | Public ResultMessage updateHotelInfo (HotelVO hotelVO) | | | | |
| 前置条件 | 酒店工作人员确认修改酒店基本信息，已知修改后的酒店基本面信息 | | | | |
| 后置条件 | 系统更新并持久化保存酒店基本信息，并提示酒店工作人员保存是否成功 | | | | |
| HotelController.getHotelRoomInfo | 语法 | Public RoomInfoVO getHotelRoomInfo (String userID) | | | | |
| 前置条件 | 已知酒店工作人员的编号，需要维护客房信息 | | | | |
| 后置条件 | 系统返回该酒店工作人员管理的客房信息 | | | | |
| HotelController.updateHotelRoomInfo | 语法 | Public ResultMessage updateHotelRoomInfo (RoomInfoVO RoomInfoVO) | | | | |
| 前置条件 | 酒店工作人员确认修改客房信息，已知修改后的客房信息 | | | | |
| 后置条件 | 系统更新并持久化保存客房信息，并提示酒店工作人员保存是否成功 | | | | |
| HotelController.updateCheckIn | 语法 | Public ResultMessage updateCheckIn (List<String> roomNum,Calendar inTime,Calendar predictedOutTime) | | | | |
| 前置条件 | 酒店工作人员已执行订单，并已知入住房间号的列表，入住时间以及预计离开时间 | | | | |
| 后置条件 | 系统更新并持久化保存入住信息，并提示酒店工作人员是否更新成功 | | | | |
| HotelController.updateCheckOut | 语法 | Public ResultMessage updateCheckOut (Calendar outTime) | | | | |
| 前置条件 | 酒店工作人员对订单办理退房，已知退房时间 | | | | |
| 后置条件 | 系统更新并持久化保存退房信息，并提示酒店工作人员是否更新成功 | | | | |
| HotelController.getRemainRoomInfo | 语法 | Public RemainRoomInfoVO getRemainRoomInfo (String userID) | | | | |
| 前置条件 | 已知酒店工作人员的编号，需要对线下客户的入住和退房进行处理或需要查看剩余客房信息 | | | | |
| 后置条件 | 系统返回该酒店工作人员所对应酒店的剩余客房信息 | | | | |
| 需要的接口（需接口） | | | | | | | |
| 服务名 | **服务内容** | | | | | | |
| Hotel.getHotelInfo | 得到酒店基本信息 | | | | | | |
| Hotel.getRoomInfo | 得到酒店客房信息 | | | | | | |
| Hotel.getRemainRoomInfo | 得到酒店的剩余客房信息 | | | | | | |
| Hotel.updateHotelInfo | 更新酒店基本信息 | | | | |
| Hotel.updateRoomInfo | 更新酒店客房信息 | | | | | | | |
| ExecutedOrderInfo.getExecutedOrder | 得到已执行的订单 | | | | | |
| ExecutedOrderInfo.checkOut | 已执行的订单退房 | | |
| UnExecutedOrderInfo.getUnexecutedOrder | 得到未执行的订单 | |
| UnExecutedOrderInfo.execute | 执行未执行的订单 | |
| AbnormalOrderInfo.getAbnornalOrder | 得到异常的订单 | | | |
| UnExecutedOrderInfo.execute | 执行异常订单 | | | | |

1. Domain类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Hotel.getHotelInfo | | 得到酒店基本信息 |
| Hotel.getRoomInfo | | 得到酒店客房信息 |
| Hotel.getRemainRoomInfo | | 得到酒店的剩余客房信息 |
| Hotel.updateHotelInfo | | 更新酒店基本信息 |
| Hotel.updateRoomInfo | | 更新酒店客房信息 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| HotelDataService.getHotelList(Address VO addressVO) | 得到数据库里的当前商圈、地址信息里的所有酒店列表 | |
| HotelDataService.getHotelDetail(String hotelID) | | 得到数据库里的对应当前酒店编号的详细信息 |
| HotelDataService.addEvaluation(EvaluationPO evaluationPO) | | 在数据库里给对应当前酒店编号的酒店添加评论 |
| HotelDataService.getHotelInfo(String hotelID) | | 从数据库里获得本酒店的酒店基本信息 |
| HotelDataService.setHotelInfo(HotelInfoPO  hotelInfoPO) | | 修改数据库里本酒店酒店基本信息 |
| HotelDataService.getHotelRoomInfo(String hotelID) | | 从数据库里获得本酒店的客房信息 |
| HotelDataService.setHotelRoomInfo (HotelRoomInfoPO hotelRoomInfoPO) | | 修改数据库里本酒店的客房信息 |
| HotelDataService.setCheckInInfo(CheckInInfo checkInInfo) | | 修改数据库里订单的入住信息 |
| HotelDataService.setCheckOutInfo(CheckOutInfo checkOutInfo) | | 修改数据库里该订单的退房时间 |
| HotelDataService.getRemainRoomInfo(String hotelID) | | 获取数据库里该酒店的剩余房间信息 |
| HotelDataService.setRemainRoomInfo  (RemainRoomInfoPO remainRoomInfoPO) | | 修改数据库里该酒店的剩余房间信息 |
| HotelDataService.add(HotelInfoPO hotelInfoPO) | | 在数据库里添加一条就酒店的信息 |
| Promotion.getPromotions(String userID) | | 依据酒店工作人员编号得到数据库里当前酒店的所有策略信息列表 |
| Order.getBookedHotels(String guestID) | 从所有订单中得到该用户预定过的酒店 | |

1. 业务逻辑层的动态模型
2. 业务逻辑层的设计原理

### promotion模块

1. 模块概述

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

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

1. 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。在展示层和业务逻辑层之间添加promotionBLService接口。为了隔离业务逻辑职责和逻辑控制职责，添加promotionController，这样promotionController会将对查看及维护促销策略的业务逻辑处理委托给promotion对象。PromotionPO是作为促销策略的持久性对象被添加到设计模型中去的。

1. 模块内部类的接口规范
2. Controller类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PromotionController.getPromotions | 语法 | Public List<promotionVO> getPromotions (String userID) |
| 前置条件 | 已知用户编号（可以是酒店工作人员，也可以是网站营销人员），需要维护促销策略 |
| 后置条件 | 系统返回当前用户所管理的促销策略的列表 |
| PromotionController.update | 语法 | Public ResultMessage update (List<PromotionVO> list) |
| 前置条件 | 用户（酒店工作人员或网站营销人员）确认更新促销策略，已知当前的促销策略的列表 |
| 后置条件 | 更新并持久化保存当前促销策略，并提示用户更新是否成功 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| Promotion.getPromotions | 得到当前用户所管理的促销策略列表 | |
| Promotion.update | 更新促销策略 | |

1. Domain类的接口规范

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 提供的服务（供接口） | | | | |
| Promotion.getPromotions | 语法 | | Public List<promotionVO> getPromotions (String userID) | |
| 前置条件 | | 已知酒店工作人员或网站营销人员的编号 | |
| 后置条件 | | 返回促销策略列表 | |
| Promotion.update | 语法 | | Public boolean update (String userID) | |
| 前置条件 | | 已知酒店工作人员或网站营销人员的编号 | |
| 后置条件 | | 返回更新是否成功的布尔值 | |
| 需要的接口（需接口） | | | | |
| 服务名 | **服务内容** | | | |
| PromotionDataService.  getPromotions(String userID) | | 根据提供的用户编号，找到并返回数据库中对应用户所管理的策略列表 | |
| PromotionDataService.  setPromotions(String userID,List<promotionPO> list) | | 根据持有一系列promotionPO的list对象修改数据库中对应用户所管理的策略列表 | |

1. 业务逻辑层的动态模型
2. 业务逻辑层的设计原理