## 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。

### logIn模块

1. 模块概述

logIn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。

logIn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logInBLService接口。为了隔离业务逻辑职责和逻辑控制职责，添加logInController，这样logInController会将对用户登录和客户注册的业务逻辑处理委托给logIn对象。guestPO、hotelWorkerPO、webMarketer、webManager是作为账户信息的持久性对象被添加到设计模型中去的。



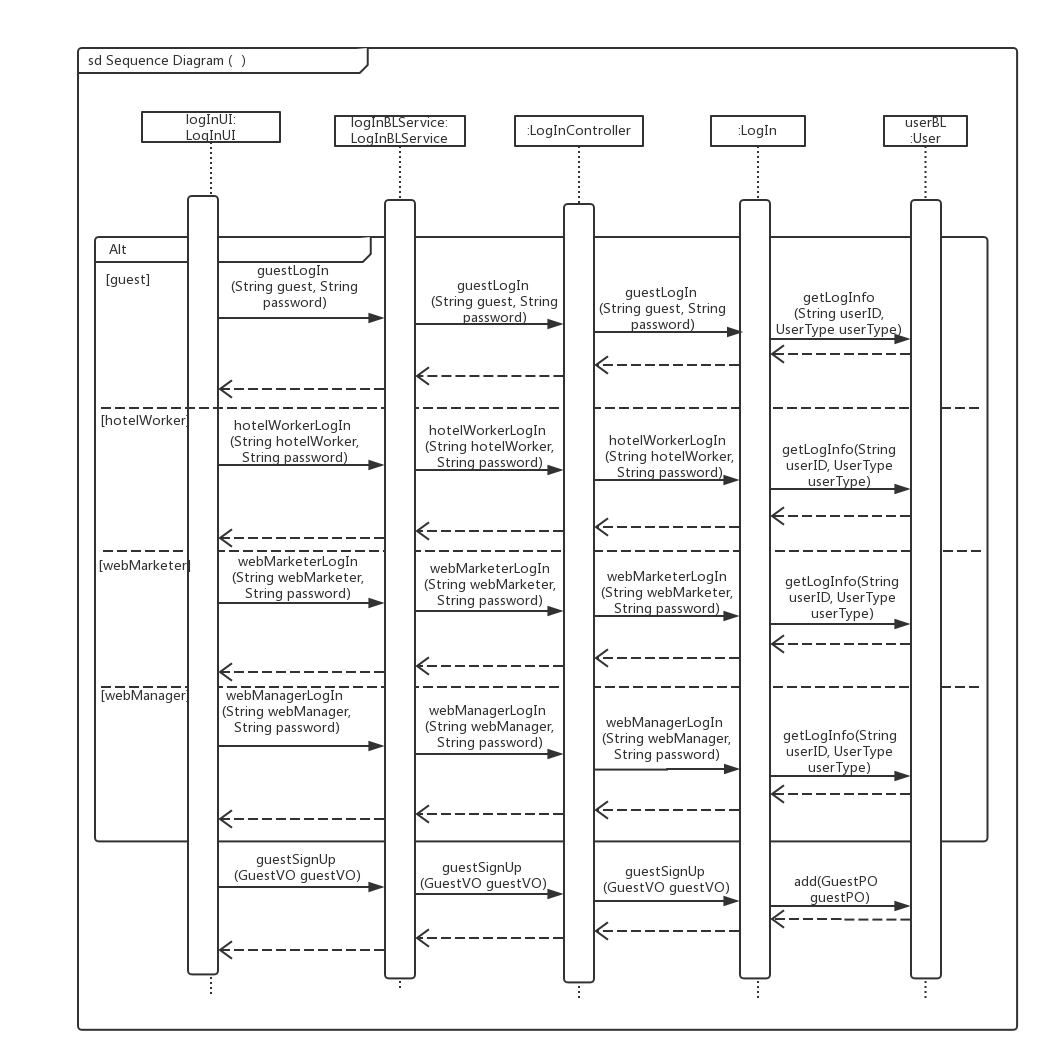
1. 模块内部类的接口规范
2. Controller类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| LogInController.guestSignUp | 语法 | public GuestVO guestSignUp (UserVO guestVO) |
| 前置条件 | 客户信息符合规范要求 |
| 后置条件 | 系统返回是否成功添加客户；是则进入系统的客户主界面，否则提示添加失败 |
| LogInController.logIn | 语法 | public UserType logIn (String userID, String password) |
| 前置条件 | 用户输入的用户编号和密码符合规范 |
| 后置条件 | 用户被认证并进入其相应的主界面 |
| LogInController.logOut | 语法 | public void logOut (String webManager) |
| 前置条件 | 无 |
| 后置条件 | 系统退出，进入登录主界面 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| LogIn.guestSignUp | 客户注册 | |
| LogIn.logIn | 用户登录系统 | |
| LogIn.logOut | 用户登出系统 | |

1. Domain类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| LogIn.guestSignUp | 语法 | public GuestVO guestSignUp (UserVO guestVO) |
| 前置条件 | 客户信息符合规范要求 |
| 后置条件 | 系统返回是否成功添加客户；是则进入系统的客户主界面，否则提示添加失败 |
| LogIn.logIn | 语法 | public UserType logIn (String userID, String password) |
| 前置条件 | 用户输入的用户编号和密码符合规范 |
| 后置条件 | 用户被认证并进入其相应的主界面 |
| LogIn.logOut | 语法 | public void logOut (String webManager) |
| 前置条件 | 无 |
| 后置条件 | 系统退出，进入登录主界面 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| User.add(UserVo newUserVo, UserType userType) | 新增该用户并持久化保存该用户的数据 | |
| User.getLogInfo(String userID, UserType userType) | 返回与该ID对应的user密码 | |

1. 业务逻辑层的动态模型



LogIn顺序图

1. 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。

### order模块

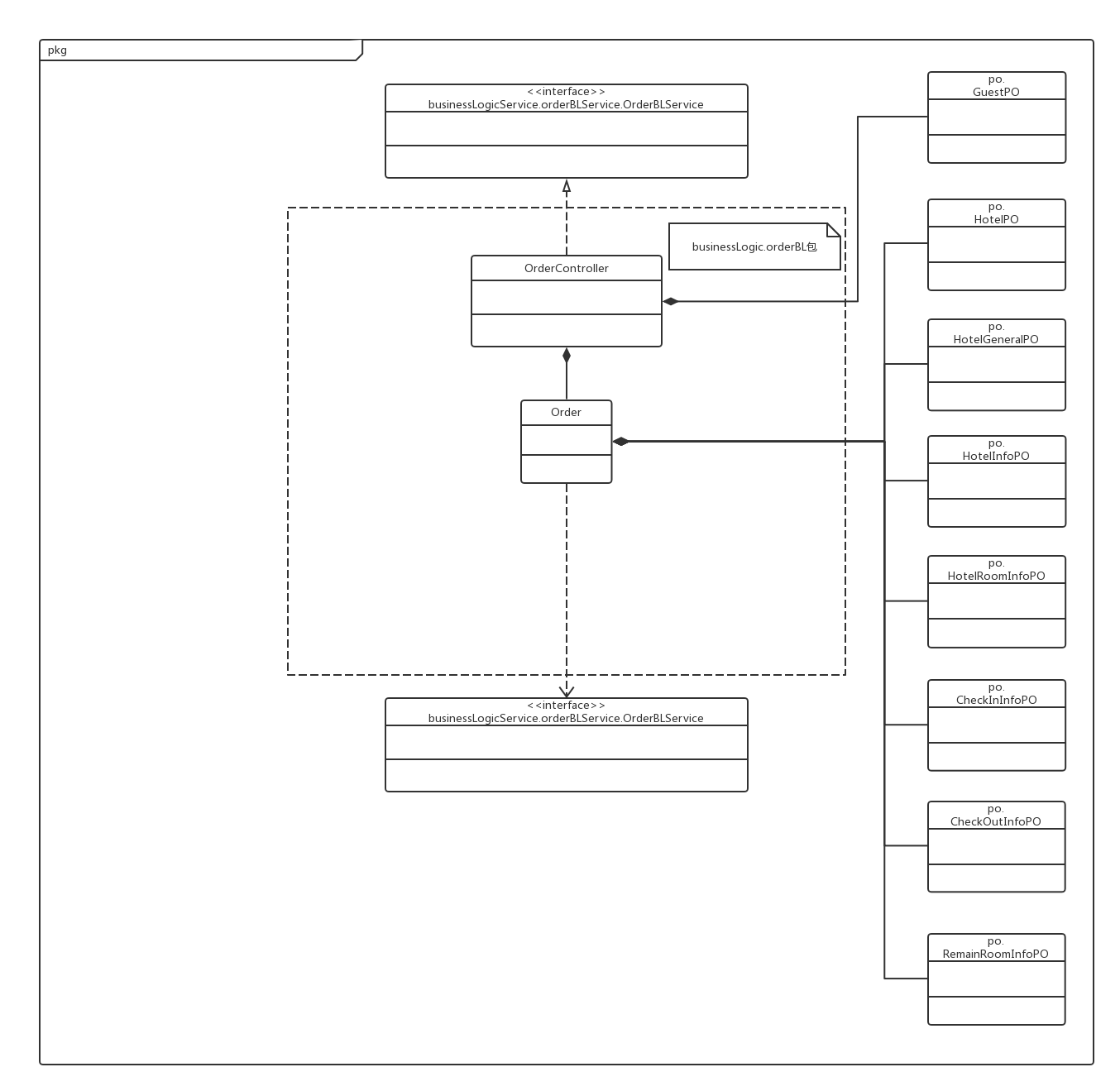
1. 模块概述

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。

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orderBLService接口，在业务逻辑层和数据层之间添加orderDataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加OrderController，这样OrderController会将对用户登录和客户注册的业务逻辑处理委托给Order对象。GuestPO、HotelPO、HotelGeneralPO、HotelInfoPO、HotelRoomInfoPO、CheckInInfoPO、CheckOutInfoPO、RemainRoomInfoPO是作为账户信息的持久性对象被添加到设计模型中去的。



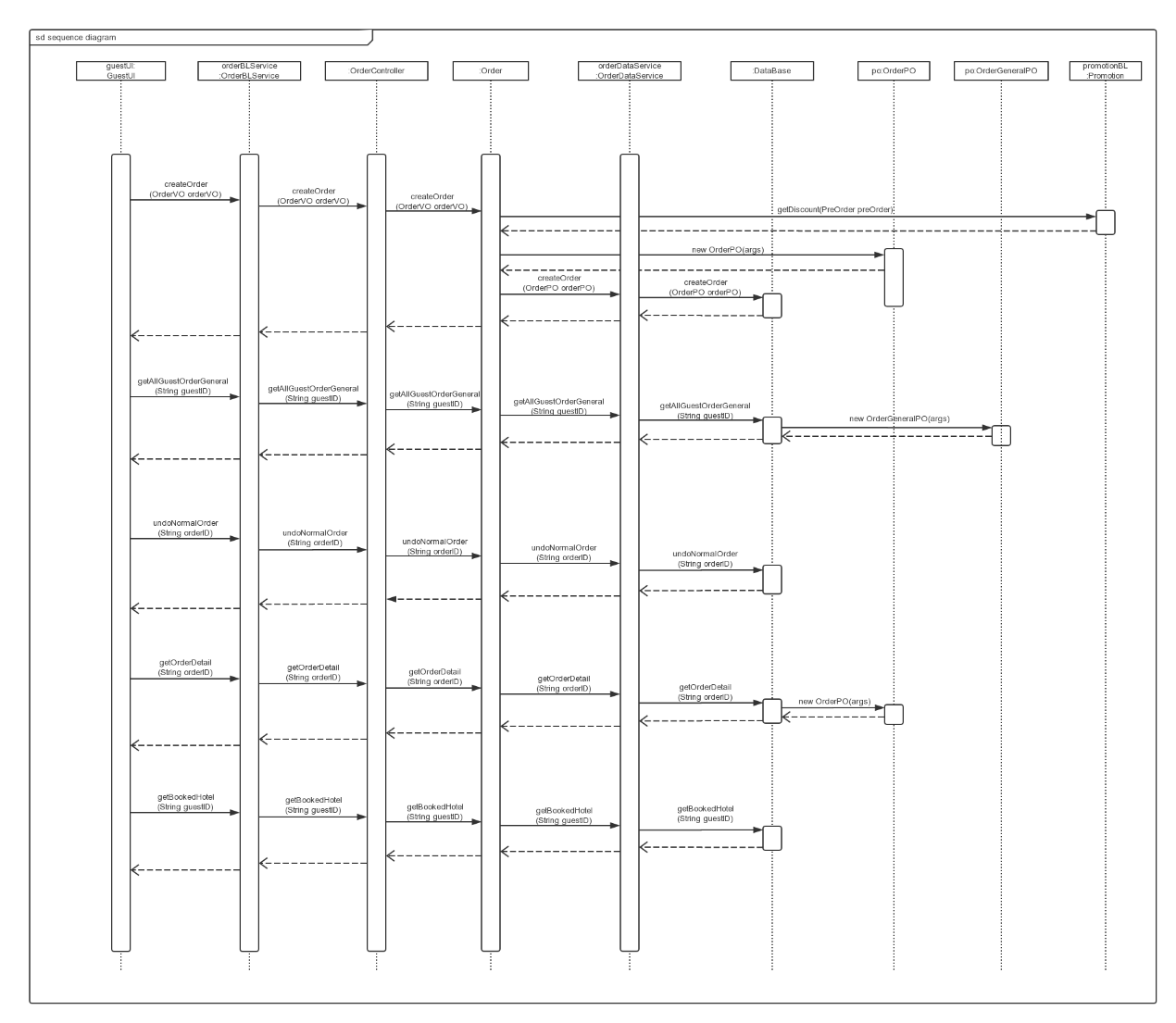
1. 模块内部类的接口规范
2. Controller类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderBLController.getAllOrderGenerals | 语法 | public Iterator<OrderGeneralVO> getAllOrderGenerals(String userID, UserType userType) |
| 前置条件 | 客户／酒店工作人员已经登录 |
| 后置条件 | 无 |
| OrderBLController.getSpecialOrderGenerals | 语法 | public Iterator<OrderGeneralVO> getSpecialOrderGenerals(String userID, UserType userType, OrderState orderState) |
| 前置条件 | 客户／酒店工作人员已经登录 |
| 后置条件 | 无 |
| OrderBLController.getOrderDetail | 语法 | public OrderVO getOrderDetail(String orderID) |
| 前置条件 | 客户／酒店工作人员／网站营销人员已经登录 |
| 后置条件 | 无 |
| OrderBLController.getEvaluations | 语法 | public Iterator<HotelEvaluationVO> getEvaluations(String hotelID) |
| 前置条件 | 客户／酒店工作人员已经登录 |
| 后置条件 | 无 |
| OrderBLController.getCalculatedPrice | 语法 | public int getCalculatedPrice(PreOrderVO preOrderVO) |
| 前置条件 | 客户已经登录，填写订单的内容符合规则且完整 |
| 后置条件 | 系统生成未执行订单并持久化增加该客户的订单数据 |
| OrderBLController.createOrder | 语法 | public ResultMessage createOrder(OrderVO orderVO) |
| 前置条件 | 客户已经登录，填写订单的内容符合规则且完整 |
| 后置条件 | 系统生成未执行订单并持久化增加该客户的订单数据 |
| OrderBLController.undoNormalOrder | 语法 | public ResultMessage undoNormalOrder(String orderID) |
| 前置条件 | 客户已经登录，订单状态为未执行 |
| 后置条件 | 系统更改状态为已执行订单并持久化保存该客户的订单数据 |
| OrderBLController.addEvaluation | 语法 | public ResultMessage addEvaluation(GuestEvaluationVO evaluationVO) |
| 前置条件 | 客户已经登录，订单状态为未执行 |
| 后置条件 | 系统更改状态为已执行订单并持久化保存该客户的订单数据 |
| OrderBLController.getAllGuestCommentOrderGeneral | 语法 | public Iterator<OrderGeneralVO> getAllGuestCommentOrderGeneral(String guestID, boolean hasCommented) |
| 前置条件 | 客户已经登录 |
| 后置条件 | 无 |
| OrderBLController.getMyOrdersOfThisHotel | 语法 | public Iterator<OrderGeneralVO> getMyOrdersOfThisHotel(String guestID, String hotelID) |
| 前置条件 | 客户已经登录 |
| 后置条件 | 无 |
| OrderBLController.updateCheckIn | 语法 | public ResultMessage updateCheckIn(CheckInVO checkInVO) |
| 前置条件 | 酒店工作人员已经登录 |
| 后置条件 | 系统更改此订单状态为已执行订单并持久化保存该客户的订单数据 |
| OrderBLController.updateCheckOut | 语法 | public ResultMessage updateCheckOut(CheckOutVO checkOutVO) |
| 前置条件 | 酒店工作人员已经登录 |
| 后置条件 | 系统更改此订单状态和酒店剩余房间数量，并持久化保存该客户的订单数据 |
| OrderBLController.getAllHotelCheckOutOrderGeneral | 语法 | public Iterator<OrderGeneralVO> getAllHotelCheckOutOrderGeneral(String hotelID, boolean hasCheckOut) |
| 前置条件 | 酒店工作人员已经登录 |
| 后置条件 | 无 |
| OrderBLController.undoAbnormalOrder | 语法 | public ResultMessage undoAbnormalOrder(String orderID, double percent) |
| 前置条件 | 网站营销人员已登录，订单状态为异常 |
| 后置条件 | 系统更改订单状态为已撤销订单并持久化保存该客户的订单数据 |
| OrderBLController.getAllAbnormalOrderGeneral | 语法 | public List<OrderGeneralVO> getAllAbnormalOrderGeneral(LocalDate date) |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 无 |
| OrderBLController.getAllUnexecutedOrderGeneral | 语法 | public List<OrderGeneralVO> getAllUnexecutedOrderGeneral(LocalDate date) |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 无 |
| OrderBLController.getBookedHotel | 语法 | public List<String> getBookedHotels(String guestID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| OrderBLController.getOrderState | 语法 | public OrderState getOrderState(String guestID, String hotelID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| CommonOrder.getAllOrderGenerals(String userID, UserType userType) | 查看客户／酒店工作人员的全部订单 | |
| CommonOrder.getSpecialOrderGenerals(String userID, UserType userType, OrderState orderState) | 查看客户／酒店工作人员某种特定类型（如未执行、已执行、异常、已撤销）的全部订单 | |
| CommonOrder.getOrderDetail(String orderID) | 查看订单详情 | |
| CommonOrder.getEvaluations(String hotelID) | 客户／酒店工作人员查看酒店评价 | |
| GuestOrder.getCalculatedPrice(PreOrderVO preOrderVO) | 通过现有订单条件计算得到的此订单需付价格 | |
| GuestOrder.createOrder (OrderVO OrderVO) | 客户创建订单 | |
| GuestOrder.undoNormalOrder(String orderID) | 客户撤销异常订单 | |
| GuestOrder.addEvaluation(GuestEvaluationVO evaluationVO) | 客户评价订单 | |
| GuestOrder.getAllGuestCommentOrderGeneral(String guestID, boolean hasCommented) | 客户查看本人所有已评价／未评价的订单 | |
| GuestOrder.getMyOrdersOfThisHotel(String guestID, String hotelID) | 查看客户在目标酒店的所有订单记录 | |
| HotelWorkerOrder.updateCheckIn(CheckInVO checkInVO) | 酒店工作人员更新入住信息 | |
| HotelWorkerOrder.updateCheckOut(CheckOutVO checkOutVO) | 酒店工作人员更新入住信息 | |
| HotelWorkerOrder.getAllHotelCheckOutOrderGeneral(String hotelID, boolean hasCheckOut) | 酒店工作人员查看已退房／未退房订单 | |
| WebMarketerOrder.undoAbnormalOrder(String orderID, double percent) | 网站营销人员撤销异常订单 | |
| WebMarketerOrder.getAllAbnormalOrderGeneral(LocalDate date) | 网站营销人员查看当日所有异常订单 | |
| WebMarketerOrder.getAllUnexecutedOrderGeneral(LocalDate date) | 网站营销人员查看当日所有未执行订单 | |
| OrderForHotelModule.getBookedHotel(String guestID) | 查看客户预定过的酒店 | |
| OrderForHotelModule.getOrderState(String guestID, String hotelID) | 查看此客户在此相应酒店预定过的订单状态 | |

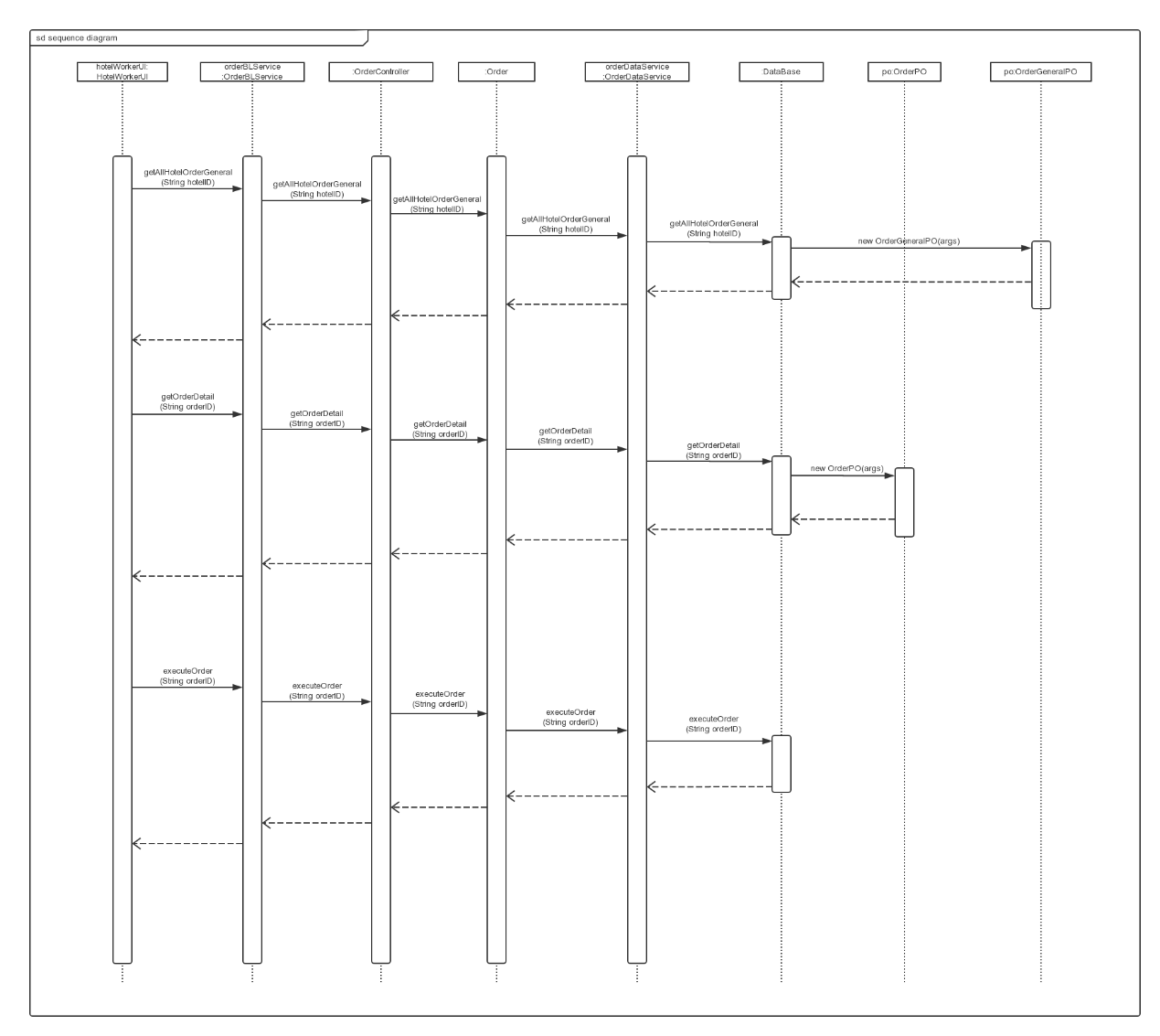
1. Domain类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CommonOrder.getAllOrderGenerals | 语法 | public Iterator<OrderGeneralVO> getAllOrderGenerals(String userID, UserType userType) |
| 前置条件 | 客户／酒店工作人员已经登录 |
| 后置条件 | 无 |
| CommonOrder.getSpecialOrderGenerals | 语法 | public Iterator<OrderGeneralVO> getSpecialOrderGenerals(String userID, UserType userType, OrderState orderState) |
| 前置条件 | 客户／酒店工作人员已经登录 |
| 后置条件 | 无 |
| CommonOrder.getOrderDetail | 语法 | public OrderVO getOrderDetail(String orderID) |
| 前置条件 | 客户／酒店工作人员／网站营销人员已经登录 |
| 后置条件 | 无 |
| CommonOrder.getEvaluations | 语法 | public Iterator<HotelEvaluationVO> getEvaluations(String hotelID) |
| 前置条件 | 客户／酒店工作人员已经登录 |
| 后置条件 | 无 |
| GuestOrder.getCalculatedPrice | 语法 | public int getCalculatedPrice(PreOrderVO preOrderVO) |
| 前置条件 | 客户已经登录，填写订单的内容符合规则且完整 |
| 后置条件 | 系统生成未执行订单并持久化增加该客户的订单数据 |
| GuestOrder.createOrder | 语法 | public ResultMessage createOrder(OrderVO orderVO) |
| 前置条件 | 客户已经登录，填写订单的内容符合规则且完整 |
| 后置条件 | 系统生成未执行订单并持久化增加该客户的订单数据 |
| GuestOrder.undoNormalOrder | 语法 | public ResultMessage undoNormalOrder(String orderID) |
| 前置条件 | 客户已经登录，订单状态为未执行 |
| 后置条件 | 系统更改状态为已执行订单并持久化保存该客户的订单数据 |
| GuestOrder.addEvaluation | 语法 | public ResultMessage addEvaluation(GuestEvaluationVO evaluationVO) |
| 前置条件 | 客户已经登录，订单状态为未执行 |
| 后置条件 | 系统更改状态为已执行订单并持久化保存该客户的订单数据 |
| GuestOrder.getAllGuestCommentOrderGeneral | 语法 | public Iterator<OrderGeneralVO> getAllGuestCommentOrderGeneral(String guestID, boolean hasCommented) |
| 前置条件 | 客户已经登录 |
| 后置条件 | 无 |
| GuestOrder.getMyOrdersOfThisHotel | 语法 | public Iterator<OrderGeneralVO> getMyOrdersOfThisHotel(String guestID, String hotelID) |
| 前置条件 | 客户已经登录 |
| 后置条件 | 无 |
| HotelWorkerOrder.updateCheckIn | 语法 | public ResultMessage updateCheckIn(CheckInVO checkInVO) |
| 前置条件 | 酒店工作人员已经登录 |
| 后置条件 | 系统更改此订单状态为已执行订单并持久化保存该客户的订单数据 |
| HotelWorkerOrder.updateCheckOut | 语法 | public ResultMessage updateCheckOut(CheckOutVO checkOutVO) |
| 前置条件 | 酒店工作人员已经登录 |
| 后置条件 | 系统更改此订单状态和酒店剩余房间数量，并持久化保存该客户的订单数据 |
| HotelWorkerOrder.getAllHotelCheckOutOrderGeneral | 语法 | public Iterator<OrderGeneralVO> getAllHotelCheckOutOrderGeneral(String hotelID, boolean hasCheckOut) |
| 前置条件 | 酒店工作人员已经登录 |
| 后置条件 | 无 |
| WebMarketerOrder.undoAbnormalOrder | 语法 | public ResultMessage undoAbnormalOrder(String orderID, double percent) |
| 前置条件 | 网站营销人员已登录，订单状态为异常 |
| 后置条件 | 系统更改订单状态为已撤销订单并持久化保存该客户的订单数据 |
| WebMarketerOrder.getAllAbnormalOrderGeneral | 语法 | public List<OrderGeneralVO> getAllAbnormalOrderGeneral(LocalDate date) |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 无 |
| WebMarketerOrder.getAllUnexecutedOrderGeneral | 语法 | public List<OrderGeneralVO> getAllUnexecutedOrderGeneral(LocalDate date) |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 无 |
| OrderForHotelModule.getBookedHotel | 语法 | public List<String> getBookedHotels(String guestID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| OrderForHotelModule.getOrderState | 语法 | OrderState getOrderState(String guestID, String hotelID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| OrderDataService.getAllGuestOrderGeneral (String guestID) | 根据客户编号查找客户所有订单概况的持久化对象 | |
| OrderDataService.getAllHotelOrderGeneral (String hotelID) | 根据酒店编号查找酒店所有订单概况的持久化对象 | |
| OrderDataService.getOrderDetail(String orderID) | 根据订单编号查找订单详情的持久化对象 | |
| OrderDataService.getEvaluations(String hotelID) | 根据酒店编号查找此酒店评价信息的持久化对象 | |
| OrderDataService.createOrder(orderPO OrderPO) | 根据提供的po增加订单的持久化对象 | |
| OrderDataService.undoNormalOrder(String orderID) | 根据订单编号修改单一订单持久化对象 | |
| OrderDataService.addEvaluation(GuestEvaluationPO guestEvaluationPO) | 修改单一持久化对象订单的评价信息 | |
| OrderDataService.updateCheckIn(CheckInPO checkInPO) | 修改单一持久化对象订单的入住信息 | |
| OrderDataService.updateCheckOut(CheckOutPO checkOutPO) | 修改单一持久化对象订单的退房信息 | |
| OrderDataService.UndoAbnormalOrder(String orderID, double percent) | 根据订单编号和撤销比例修改单一订单持久化对象 | |
| OrderDataService.getAllAbnormalOrderGeneral (LocalDate date) | 根据时间查找持久化对象订单的集合 | |
| OrderDataService.getAllUnexecutedOrderGeneral (LocalDate date) | 根据时间查找持久化对象订单的集合 | |
| CreditBLService.addCreditRecord(CreditVO creditVO) | 新增信用记录 | |
| HotelInfoOperation.checkIn(String hotelID, RoomType roomType, int roomNum) | 修改酒店剩余房间信息 | |
| HotelInfoOperation.checkOut(String hotelID, RoomType roomType, int roomNum) | 修改酒店剩余房间信息 | |
| HotelInfoOperation.updateRemainRoomNumForUndoOrder(String hotelID, RoomType roomType, int roomNum) | 更新订单撤销后的酒店剩余房间数 | |
| HotelInfoOperation.scoreUpdate(String hotelID,double score) | 更新酒店评分 | |
| DiscountInSpan.getDiscountInSpan(PreOrderVO preOrder) | 根据现有订单状况获取此订单的折扣 | |
| UserBLService.getSingle(String userID) | 获取客户的单一持久化对象 | |

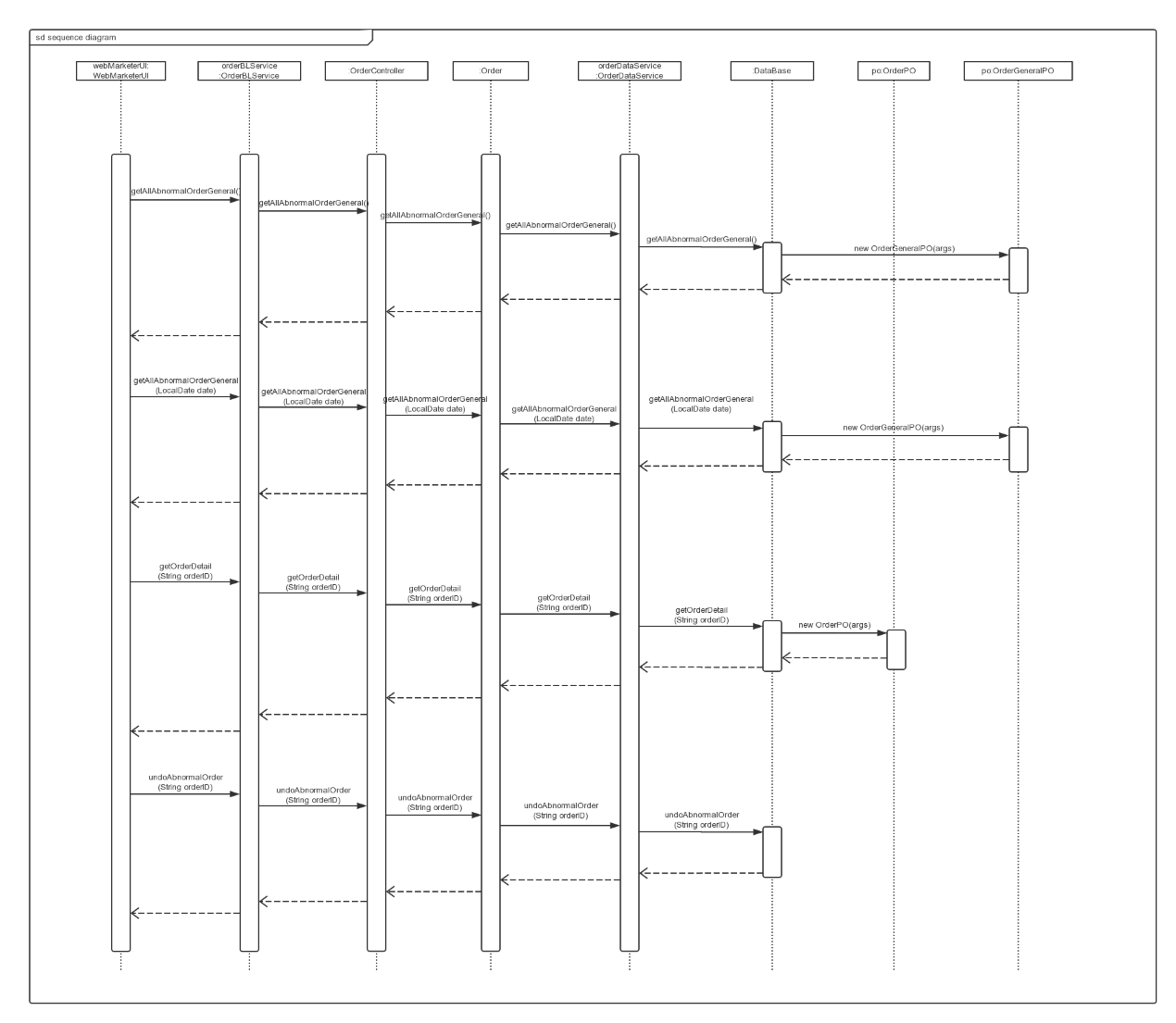
1. 业务逻辑层的动态模型



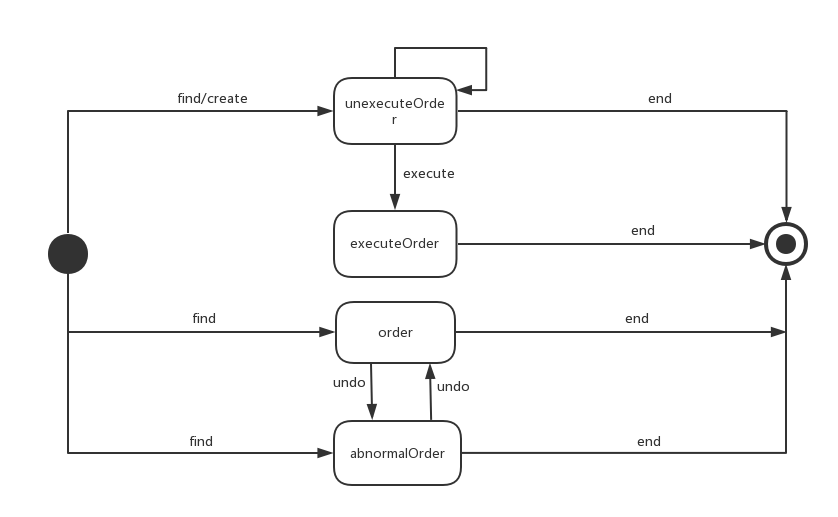
Order面向客户的顺序图



Order面向酒店工作人员的顺序图



Order面向网站营销人员的顺序图



Order状态图

1. 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。

### credit模块

1)模块概述

credit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。

credit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。在展示层和业务逻辑层之间添加creditBLService接口。为了隔离业务逻辑职责和逻辑控制职责，添加creditController，这样creditController会将对用户登录和客户注册的业务逻辑处理委托给credit对象。creditPO是作为账户信息的持久性对象被添加到设计模型中去的。



3)模块内部类的接口规范

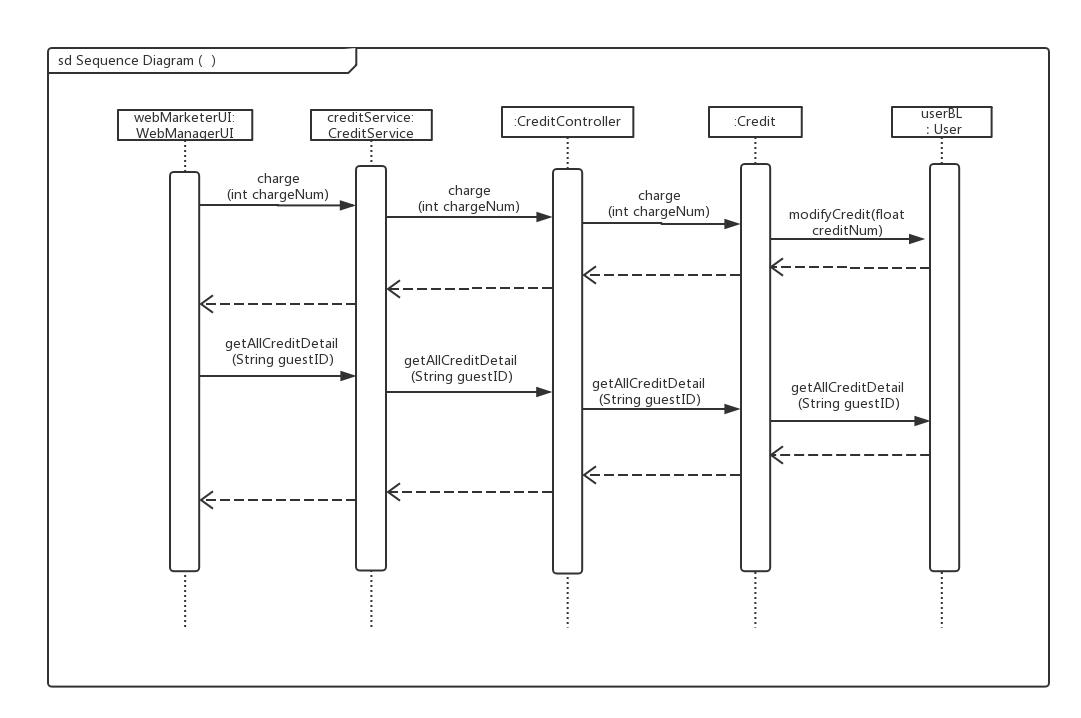
(1) Controller类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CreditController.charge | 语法 | public double charge(String guestID, double creditNum) |
| 前置条件 | 网站营销人员被识别和授权，并线下已知客户编号和欲充值的的信用值 |
| 后置条件 | 系统为客户信用值充值 |
| CreditController.getAllCreditDetail | 语法 | public Iterator<CreditVO> getAllCreditDetail(final String guestID) |
| 前置条件 | 客户已被识别和授权 |
| 后置条件 | 无 |
| CreditController.getCreditOfOneOrder | 语法 | public Iterator<CreditVO> getCreditOfOneOrder(String orderID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| CreditController.addCreditRecord | 语法 | public ResultMessage addCreditRecord(CreditVO creditVO) |
| 前置条件 | 用户已被识别和授权 |
| 后置条件 | 在信用中添加此条记录 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| Credit.charge(String guestID, double creditNum) | 为用户充值信用值 | |
| Credit.getAllCreditDetail(final String guestID) | 根据客户编号获得单个客户的所有信用记录的持久化对象 | |
| Credit.getCreditOfOneOrder(String orderID) | 获得一个订单的所有信用记录持久化对象 | |
| Credit.addCreditRecord(CreditVO creditVO) | 为客户添加信用记录 | |

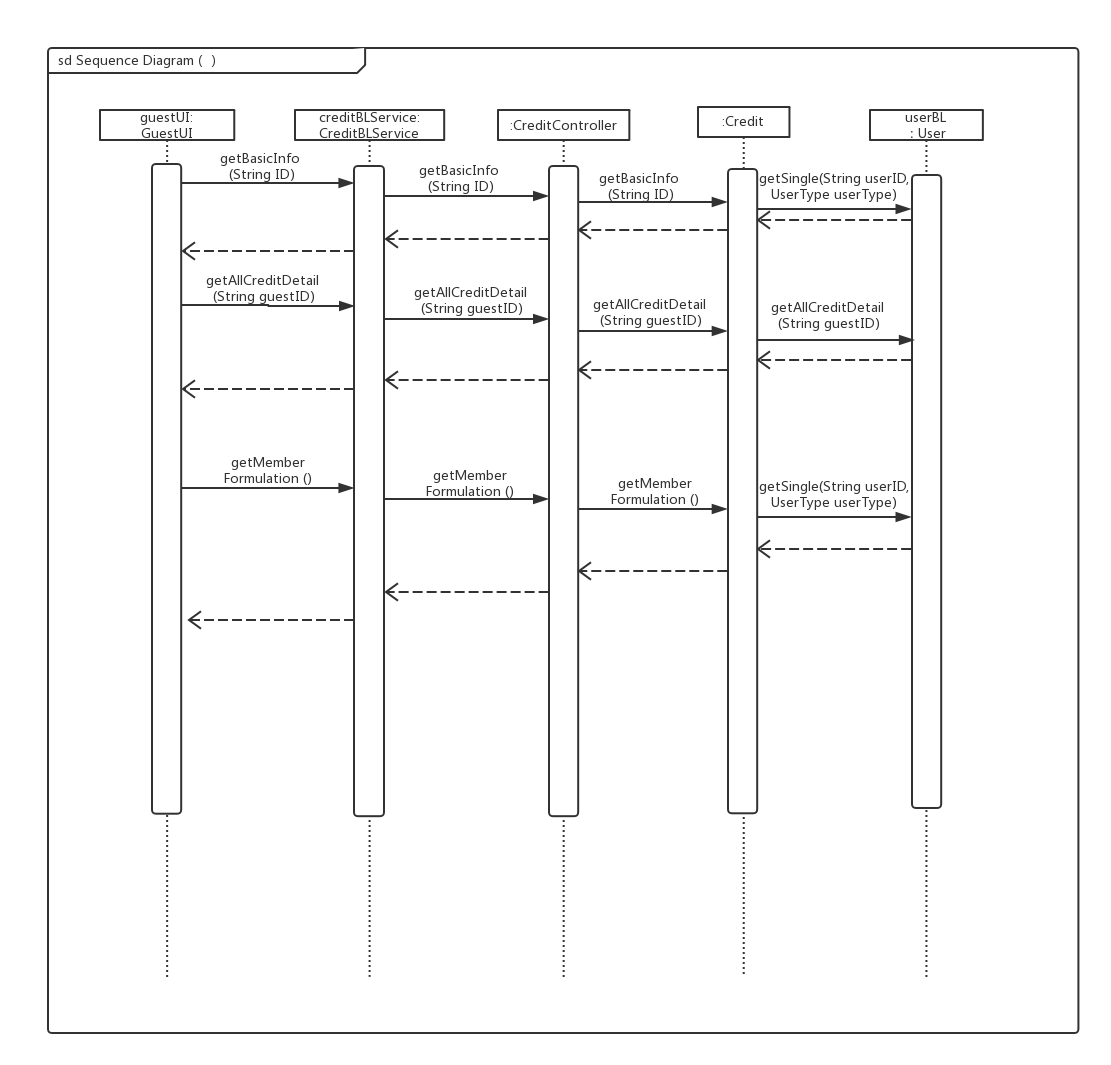
(2)Domain类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Credit.charge | 语法 | public double charge(String guestID, double creditNum) |
| 前置条件 | 网站营销人员被识别和授权，并线下已知客户编号和欲充值的的信用值 |
| 后置条件 | 系统为客户信用值充值 |
| Credit.getAllCreditDetail | 语法 | public Iterator<CreditVO> getAllCreditDetail(final String guestID) |
| 前置条件 | 客户已被识别和授权 |
| 后置条件 | 无 |
| Credit.getCreditOfOneOrder | 语法 | public Iterator<CreditVO> getCreditOfOneOrder(String orderID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| Credit.addCreditRecord | 语法 | public ResultMessage addCreditRecord(CreditVO creditVO) |
| 前置条件 | 用户已被识别和授权 |
| 后置条件 | 在信用中添加此条记录 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| CreditDataService.getAllCreditDetail(final String guestID) | 根据客户编号获得单个客户的所有信用记录的持久化对象 | |
| CreditDataService.getCreditOfOneOrder(String orderID) | 获得一个订单的所有信用记录持久化对象 | |
| CreditDataService.addCredit(CreditVO creditVO) | 为客户添加信用记录 | |
| GuestCreditService.modifyCredit(String guestID, double creditNum) | 修改客户的信用信息 | |
| UserBL.getSingle(String userID) | 根据客户编号查看客户持久化对象 | |

4) 业务逻辑层的动态模型



信用充值顺序图



信用查看顺序图

1. 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文档中的相关描述。

### orderData模块

1)模块概述

orderData模块负责保存订单信息，进行增，改，查的操作。

2)模块内部接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderDataService.createOrder | 语法 | public ResultMessage createOrder (OrderPO order) |
| 前置条件 | 客户已被认证和授权 |
| 后置条件 | 系统生成未执行订单并持久化增加该客户的订单数据 |
| OrderDataService.executeOrder | 语法 | public ResultMessage executeOrder (String orderID) |
| 前置条件 | 酒店工作人员已被认证和授权 |
| 后置条件 | 系统更改状态为已执行订单并持久化保存该客户的订单数据 |
| OrderDataService.undoAbnormalOrder | 语法 | public ResultMessage undoAbnormalOrder (String orderID, double percent) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 系统更改状态为已撤销订单、恢复一定比例的信用值并持久化保存该客户的订单数据 |
| OrderDataService.undoNormalOrder | 语法 | public ResultMessage undoNormalOrder (String orderID) |
| 前置条件 | 客户已被认证和授权，订单状态为未执行 |
| 后置条件 | 系统更改状态为已撤销订单并持久化保存该客户的订单数据 |
| OrderDataService.getOrderDetails | 语法 | public OrderPO getOrderDetail(String orderID) |
| 前置条件 | 用户已被认证和授权 |
| 后置条件 | 无 |
| OrderDataService.getAllGuestOrderGeneral | 语法 | public List<OrderGeneralPO> getAllGuestOrderGeneral (String guestID) |
| 前置条件 | 客户已被认证和授权 |
| 后置条件 | 无 |
| OrderDataService.getAllHotelOrderGeneral | 语法 | public List<OrderGeneralPO> getAllHotelOrderGeneral (String orderID) |
| 前置条件 | 酒店工作人员已被认证和授权 |
| 后置条件 | 无 |
| OrderDataService.getAllAbnormalOrderGeneral | 语法 | public List<OrderGeneralPO> getAllAbnormalOrderGeneral (LocalDate date) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| OrderDataService.getAllUnexecutedOrderGeneral | 语法 | public List<OrderGeneralPO> getAllUnexecutedOrderGeneral (LocalDate date) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| OrderDataService.updateCheckIn | 语法 | ResultMessage updateCheckIn(CheckInPO checkInPO) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| OrderDataService.updateCheckOut | 语法 | ResultMessage updateCheckOut(CheckOutPO checkOutPO) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| OrderDataService.addEvaluation | 语法 | ResultMessage addEvaluation(GuestEvaluationPO guestEvaluationPO) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| OrderDataService.getEvaluations | 语法 | List<HotelEvaluationPO> getEvaluations(String hotelID) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| OrderDataService.getBookedHotels | 语法 | List<String> getBookedHotels(String guestID) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| OrderDataHelper.add(OrderPO orderPO) | 新增订单的持久化对象 | |
| OrderDataService.setState(String orderID, OrderState state) | 根据订单编号修改订单持久化对象的状态 | |
| OrderDataHelper.getSingleOrder(String orderID) | 根据客户编号查找客户单个订单的持久化对象 | |
| OrderDataService.getAllOrderOfGuest(String guestID) | 根据客户编号查找客户的所有订单持久化对象 | |
| OrderDataHelper.getAllOrderOfHotel(String hotelID) | 根据酒店编号查找酒店的所有订单持久化对象 | |
| OrderDataService.getUnexecuted() | 查找所有未执行订单的持久化对象 | |
| OrderDataHelper.getAbnormal() | 查找所有异常订单的持久化对象 | |
| OrderDataService.setCheckIn(String orderID, String roomNumber, LocalDateTime checkInTime, LocalDateTime expectLeaveTime) | 根据订单编号修改单个订单持久化对象的入住信息 | |
| OrderDataHelper.setCheckOut(String orderID, LocalDateTime checkOutTime) | 根据订单编号修改单个订单持久化对象的退房信息 | |
| OrderDataService.setEvaluation(String orderID, double score, String comment) | 根据订单编号修改单个订单持久化对象的评论信息 | |
| OrderDataHelper.setHasCommentBool(String orderID) | 根据订单编号修改单个订单持久化对象的评价特征值 | |

### creditData模块

1)模块概述

orderData模块负责保存订单信息，进行增，改，查的操作。

2)模块内部接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CreditDataService.getAllCreditDetail | 语法 | public List<CreditPO> getAllCreditDetail(String guestID) |
| 前置条件 | 客户已被认证和授权 |
| 后置条件 | 无 |
| CreditDataService.addCredit | 语法 | public ResultMessage addCredit(CreditPO creditPO) |
| 前置条件 | 无 |
| 后置条件 | 添加一条信用记录 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| CreditDataHelper.getAllCreditDetail(String guestID) | 获取该客户的所有订单记录 | |
| CreditDataHelper.addCredit(CreditPO creditPO) | 新增信用记录 | |