**酒店预订系统**

**体系结构设计文档**

V3.1

软工小组：刘港欢、段梦洋、刘钦、曹利航

2016-10-16

# 目录目录有链接、还有导航的哟~

[更新历史](#30j0zll)·······················································································································································3

[1.引言](#1fob9te)···························································································································································4

[1.1编制目的](#3znysh7)······································································································································4

[1.2词汇表](#2et92p0)···········································································································································4

[1.3参考资料](#tyjcwt)······································································································································4

1. [产品概述](#3dy6vkm)···················································································································································4
2. [逻辑视角](#1t3h5sf)···················································································································································5
3. [组合视角](#4d34og8)···················································································································································7

[4.1开发包图](#2s8eyo1)······································································································································7

[4.2运行时进程](#17dp8vu)···································································································································10

[4.3物理部署](#3rdcrjn)······································································································································11

[4.4构件图](#26in1rg)··········································································································································12

1. [接口视角](#lnxbz9)···················································································································································13

[5.1模块的职责](#35nkun2)··································································································································13

[5.2用户界面层的分解](#1ksv4uv)·····················································································································16

[5.3业务逻辑层的分解](#3j2qqm3)······················································································································18

[5.4数据层的分解](#2xcytpi)·······························································································································28

[5.4.1数据层模块的职责](#1ci93xb)··········································································································28

[5.4.2 数据层模块的接口规范](#3whwml4)································································································29

5.5模块的关键类图···························································································································34

[6.信息视角](#2bn6wsx)····················································································································································38

[6.1数据持久化对象](#qsh70q)··························································································································38

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 刘港欢 | 10.8 | 初稿：增加逻辑包图和开发包图 | V1.0 |
| 刘港欢 | 10.11 | 增加order的业务逻辑层和数据层接口规范 | V2.0 |
| 曹利航 | 10.12 | 增加manager的业务逻辑层和数据层接口规范 | V2.1 |
| 段梦洋 | 10.13 | 增加Customer和promotion的业务逻辑层和数据层接口规范 | V2.2 |
| 刘钦 | 10.13 | 增加Hotel的业务逻辑层和数据层接口规范 | V2.3 |
| 刘港欢 | 10.14 | 修改Order逻辑层接口，将获取四种不同类型的订单合为一个接口 | V2.4 |
| 曹利航 | 10.15 | 增加页面跳转和关键类图 | V3.0 |
| 段梦洋 | 10.16 | 增加信息视角 | V3.1 |
| 刘钦 | 2016-11-13 | 从hotel中分出room部分 | V4.0 |
| 段梦洋 | 2016-11-13 | 从customer中分出credit部分 | V4.1 |
| 刘港欢 | 2016-12-24 | 优化order逻辑部分——将不同类型订单的执行整合 | v4.2 |
| 刘钦 | 2016-12-29 | promotion逻辑部分重新设计 | v4.3 |

# 1.引言

## 1.1编制目的

本报告详细完成对酒店预定系统的概要设计，达到指导详细设计和开发的目的，同时实现和测试人员及用户的沟通

## 1.2词汇表

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| \_ui | 表示某展示层 |  |
| \_bl | 表示某逻辑层 |  |
| \_data | 表示某数据层 |  |

## 1.3参考资料

1.IEEE std 1471-2000

2.《用例文档v2.0》

3.丁二玉，刘钦.计算与软件工程（卷二）[M]机械工业出版社，2012：134—182

4.Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Micheael Stal.Pattern-Oriented Software Architecture Volume 1: A system of Patterns [M]机械工业出版社,2003

5.Christopher　Fox.软件工程设计导论：过程、原理与模式（UML2.0版）[M]清华大学出版社，2007

# 2.产品概述

酒店预订系统是一个在线酒店预订系统，旨在提供各种功能来简化顾客，酒店工作人员，网站营销人员，网站管理人员操作流程的应用程序。既减少了酒店工作人员，网站营销人员和网站管理人员管理酒店，营销策略和网站时机械，重复的工作，也为顾客在预订酒店时提供了更方便，友好的体验，从而全面提高了工作效率和用户的满意程度。

# 3.逻辑视角

酒店预定系统中，选择了分层体系结构风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好地示意整个高层抽象。展示层包含GUI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如图3-1和如图3-2所示。

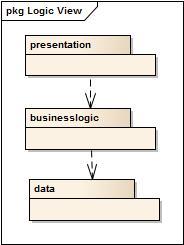


图3-1 参照体系结构风格的包图表达逻辑视角

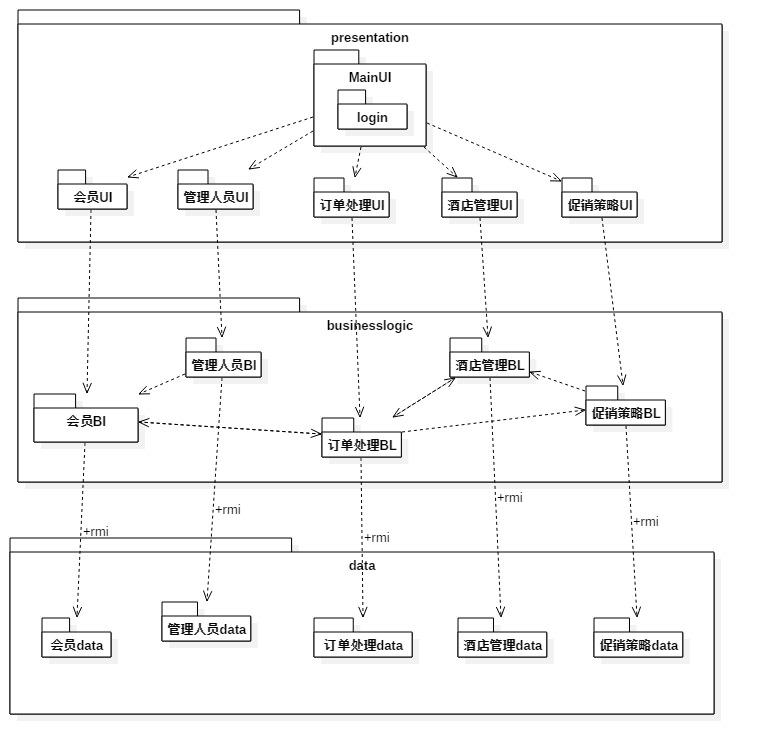


图3-2 软件体系结构逻辑设计方案

# 4.组合视角

## 4.1开发包图

酒店预订系统的最终开发包设计如表4.1-1

表4.1-1 酒店预订系统的最终开发包设计

|  |  |
| --- | --- |
| 开发包（物理） | 依赖的其他开发包 |
| mainui | customerui,promotionui,managerui,hotelui,orderui,roomui,creditui |
| promotionui | promotionblservice,界面类库包，vo |
| promotionblservice | utilitybl |
| promotionbl | promotionblservice,promotiondataservice,po,utilitybl |
| promotiondataservice | JavaRMI, po |
| promotiondata | databaseutility, po,promotiondataservice |
| customerui | cusomerbLservice, 界面类库包，vo |
| customerblservice | utilitybl |
| customerbl | customerbLservice,customerdataservice,po,salebl,utilitybl |
| customerdataservice | JavaRMI, po |
| customerdata | databaseutility, po,customerdataservice |
| managerui | managerbLservice, 界面类库包，vo |
| managerblservice | utilitybl |
| managerbl | managerblservice,managerdataservice,po,hotelbl,utilitybl |
| managerdataservice | JavaRMI, po |
| managerdata | databaseutility, po,hoteldataservice |
| hotelui | managerblservice, 界面类库包，vo |
| hotelblservice | utilitybl |
| hotelbl | hotelblservice,hoteldataservice,po,utilitybl |
| hoteldataservice | JavaRMI, po |
| hoteldata | databaseutility, po,hoteldataservice |
| orderui | orderblservice, 界面类库包，vo |
| orderblservice | utilitybl |
| orderbl | orderblservice,orderdataservice,po,salebl,utilitybl |
| orderdataservice | JavaRMI, po |
| orderdata | databaseutility, po, orderdataservice |
| creditui | creditblservice,界面类库包,vo |
| creditblservice | utilitybl |
| creditbl | creditblservice, creditdataservice, po, utilitybl |
| creditdataservice | JavaRMI, po |
| creditdata | databaseutility, po, creditdataservice |
| roomui | roomblservice,界面类库包,vo |
| roomblservice | utilitybl |
| roombl | roomblservice,roomdataservice,po,utilitybl |
| roomdataservice | JavaRMI, po |
| roomdata | databaseutility, po, roomdataservice |
| vo |  |
| po |  |
| utilitybl |  |
| 界面类库包 |  |
| Java RMI |  |
| databaseutility | JDBC |

酒店预订系统客户端开发包图如图4.1-1所示,服务器端开发包图如图4.1-2所示

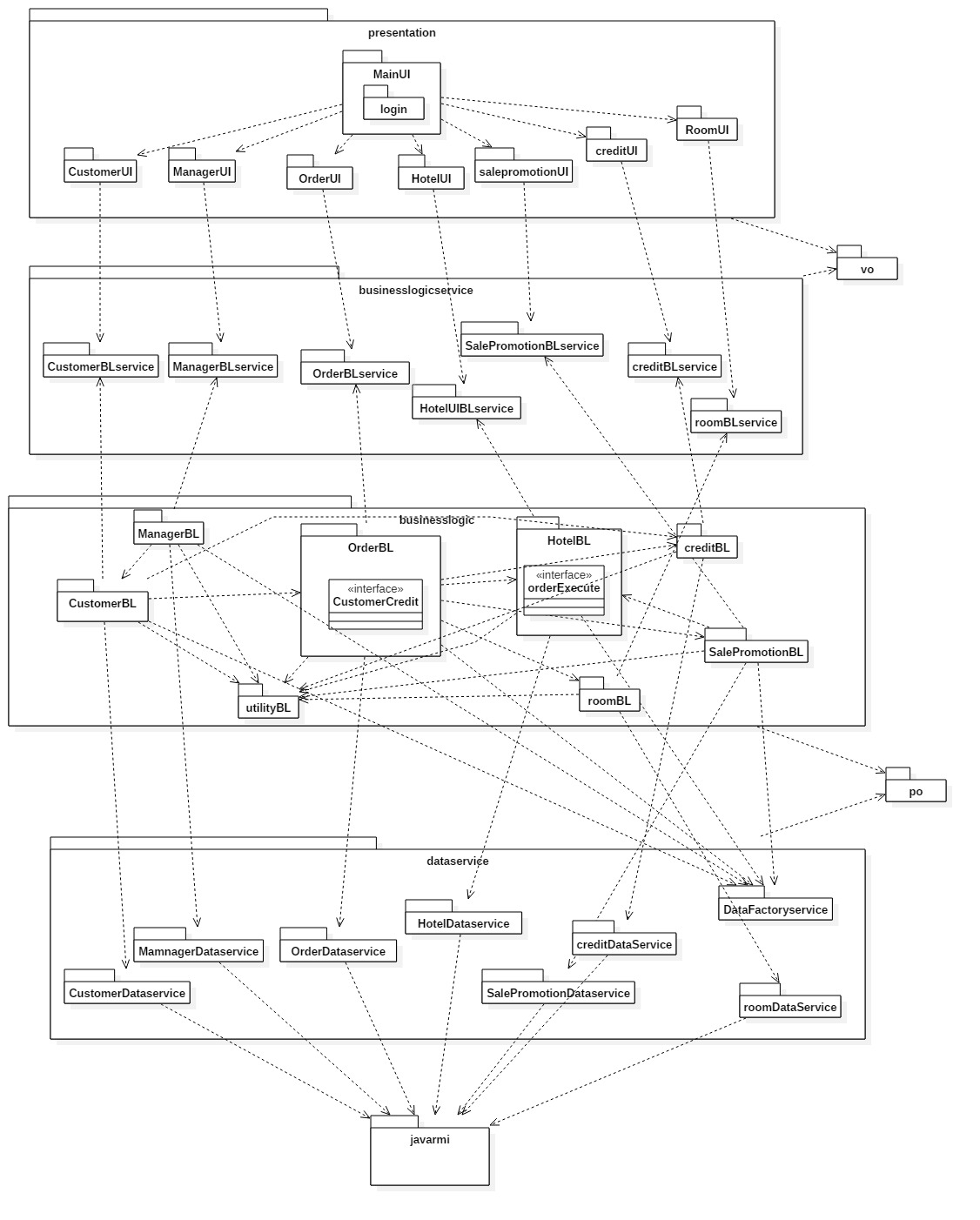


图4.1-1 酒店预订系统客户端开发包图

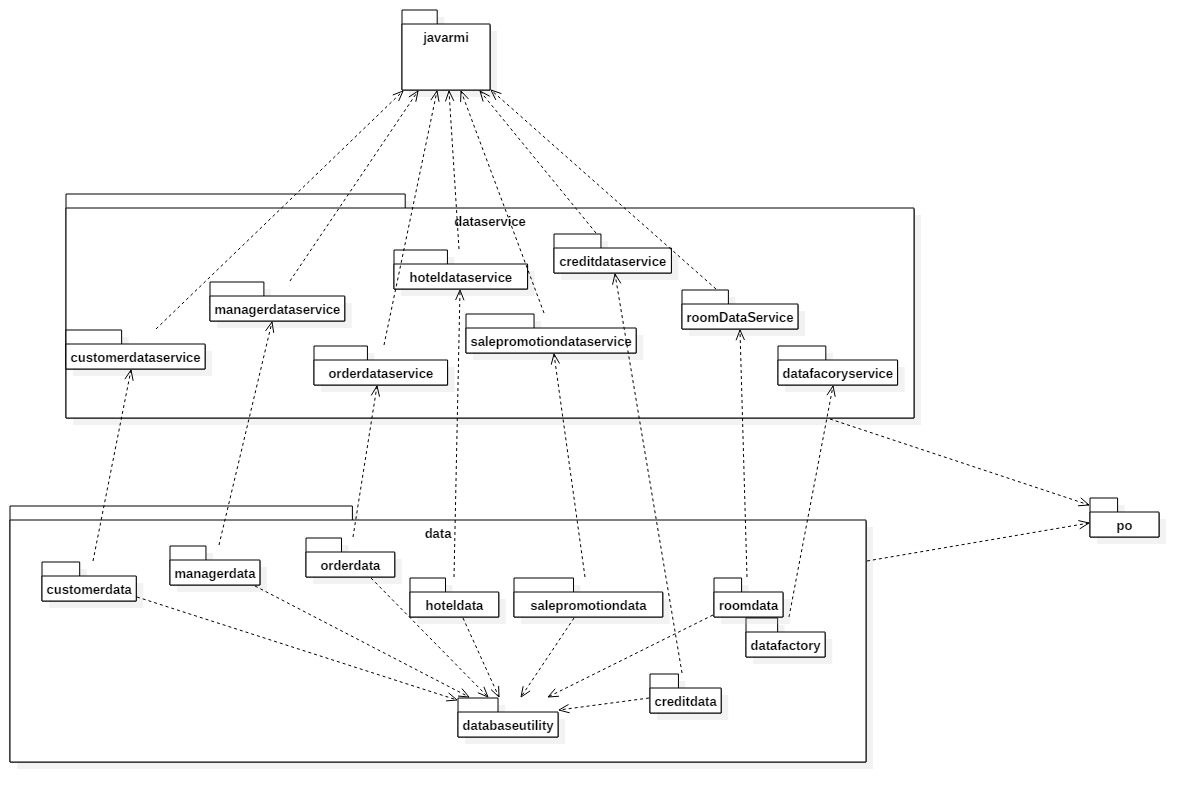


图4.1-2 酒店预订系统服务器端开发包图

## 4.2运行时进程

在CDLL酒店预订系统中,会有多个客户端进程和一个服务器端进程,其进程图如图4.2所示。结合部署图,客户端进程是在客户端机器上运行,服务器端进程在服务器端机器上运行。

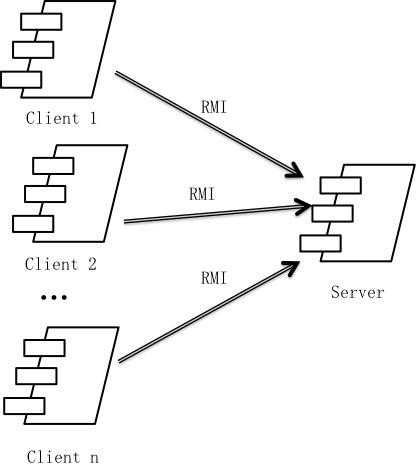


图4.2 进程图

## 4.3物理部署

酒店预定系统中客户端构件是放在客户端机器上，服务器端构件是放在服务器端机器上。在客户端节点上，还要部署RMIStub构件。由于JavaRMI构件属于JDK1.8的一部分。所以，在系统JDK环境已经设置好的情况下，不需要再独立部署。部署图如图4.3所示。

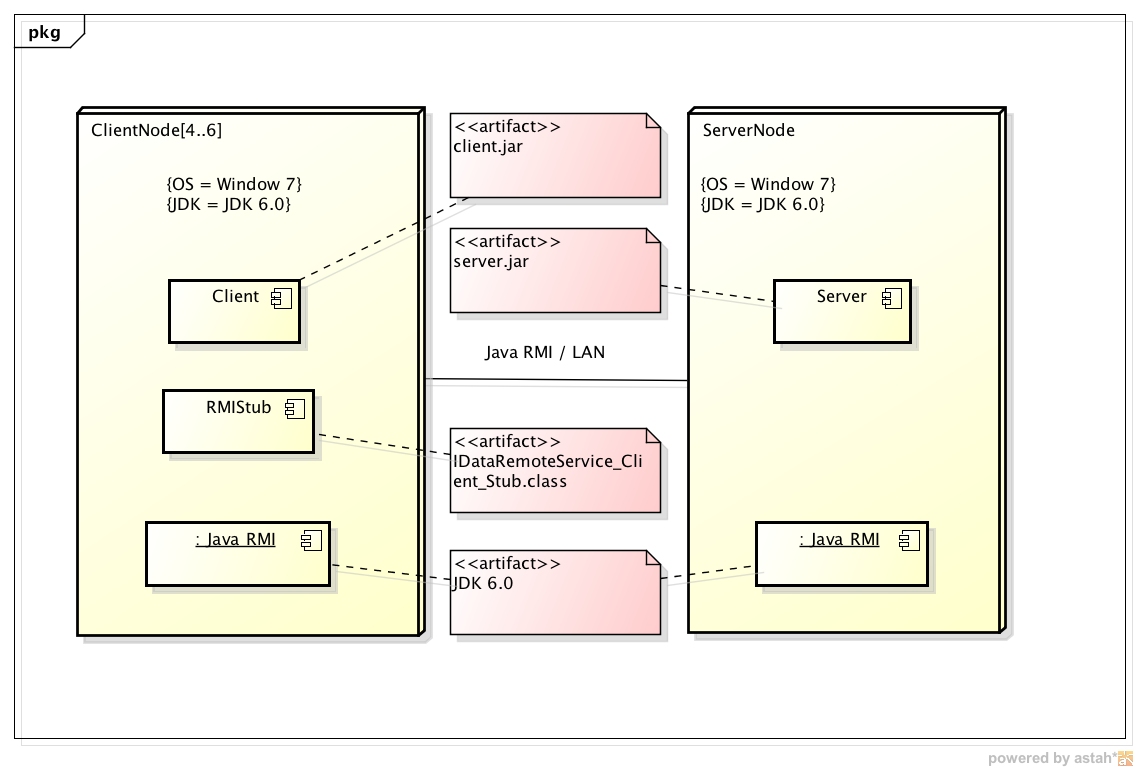


图4.3 部署图

## 4.4构件图

酒店预订系统有3个顶级构件：客户端、服务器端、以及负责连接两端的网络构件。从体系结构的角度，客户端和服务器端是构件（Component）；而网络构件是连接件（Connector）。每个顶层构件还可以进一步再细分出部分。

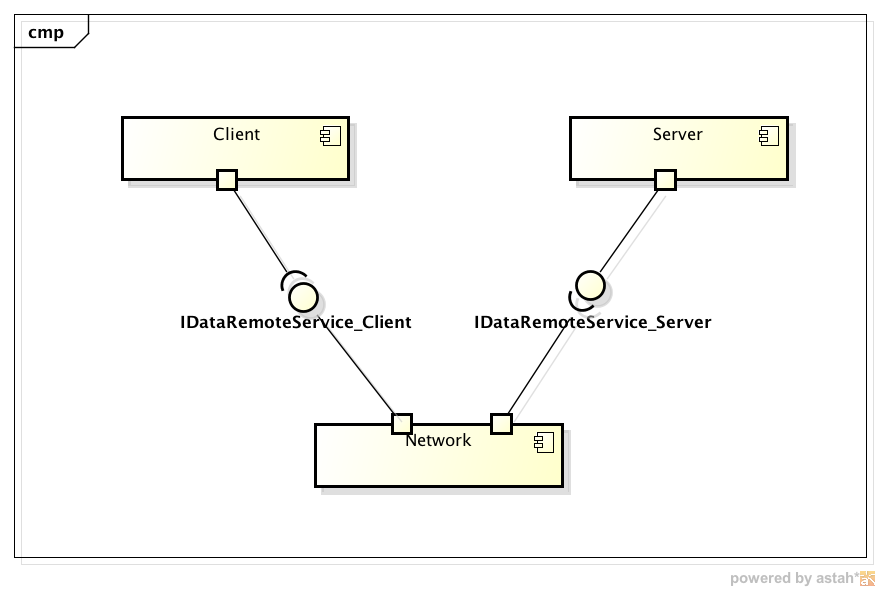


图4.4-1 0层构件图

1层构件图主要进一步细化了0层构件。这时候，需要考虑到逻辑视角的映射，通过1层构件图将逻辑视角的分层映射到0层构件图中的构件中去。逻辑视角的各层可能会分别部署在不同的物理机器上。

酒店预订系统界面层、逻辑层放在客户端，数据层是放在服务器端的。因为不同的用户（会员、酒店工作人员、网站营销人员、网站工作人员）会在客户端和服务器端分别完成相应的不同操作，所以客户端和服务器端都有各自的展示层和业务逻辑层。不同的是，客户端需要通过网络构件连接数据层的IDataRemoteService接口，而服务器端则是业务逻辑层直接访问IDataService接口。对于Network构建，我们也可以细化。由于使用了RMI技术，所以客户端需要有Stub类。Stub类和服务器端的需接口都共同依赖与Java RMI构件。

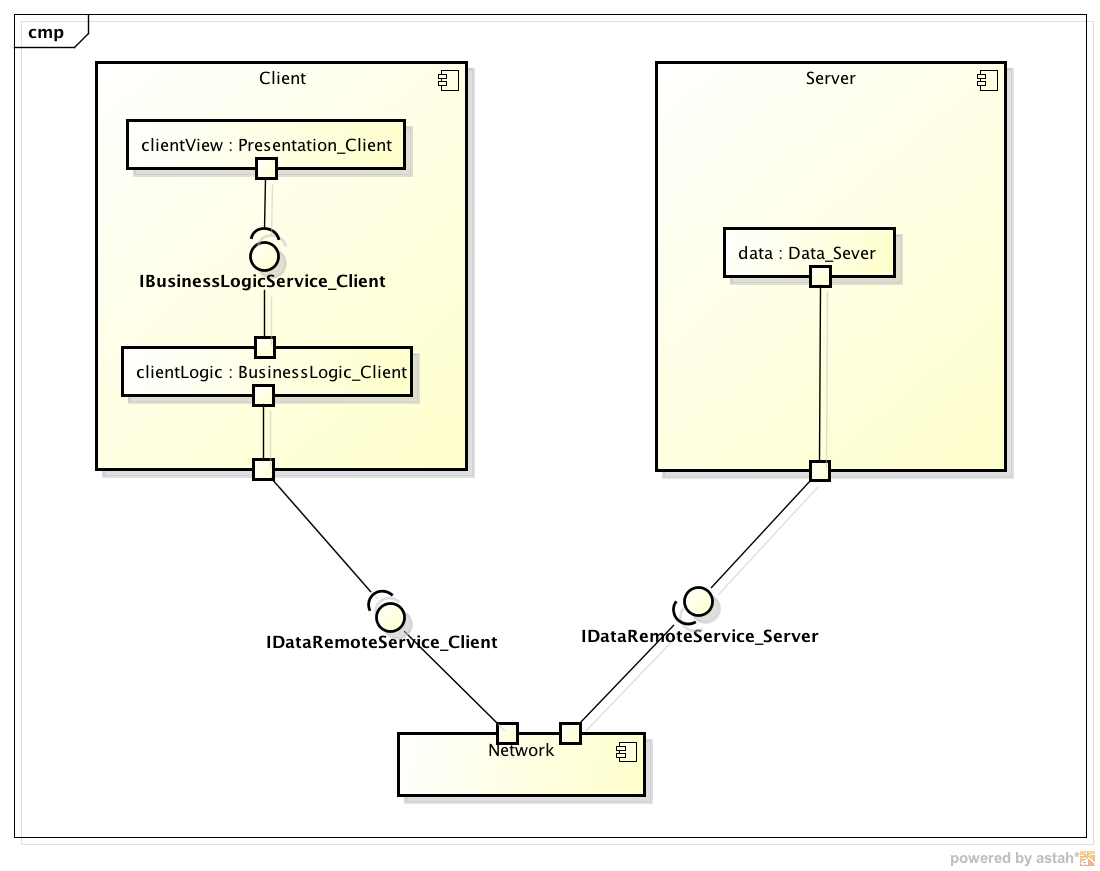


图4.4-2 1层构件图

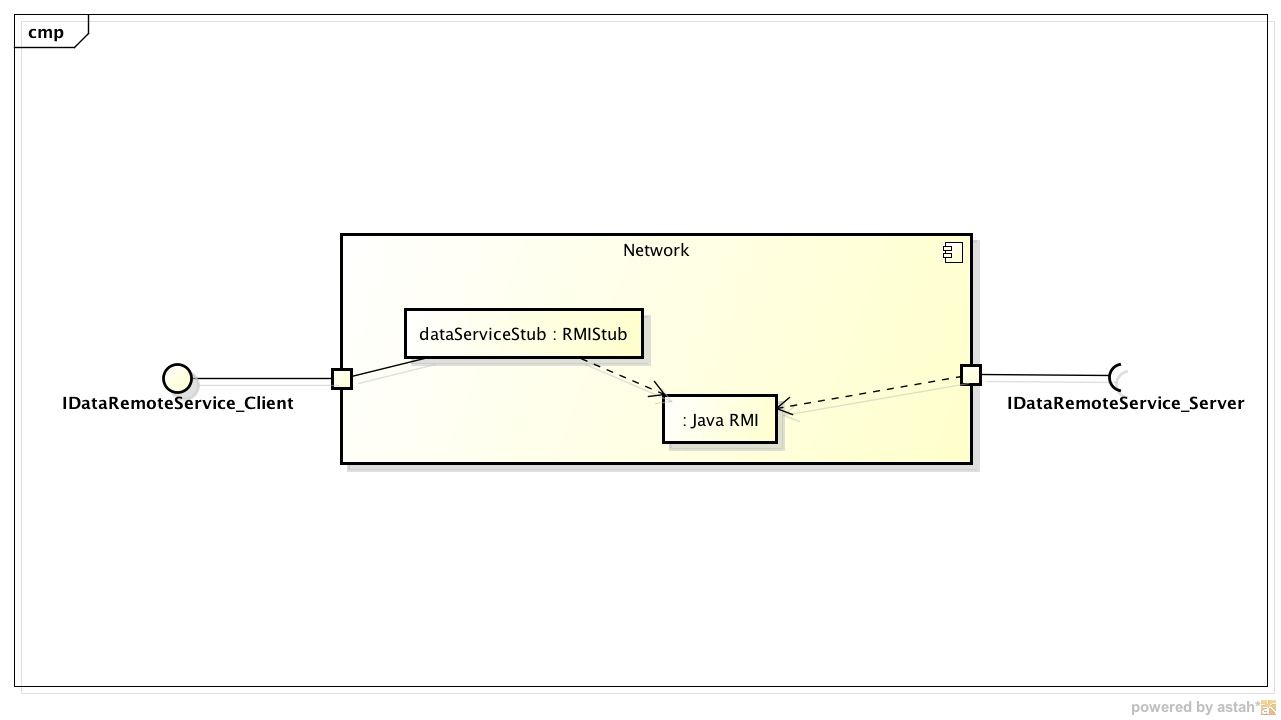


图4.4-3 Network构件图

# 5.接口视角

## 5.1模块的职责

客户端模块和服务器端模块试图分别如图5.1-1和图5.1-2所示。客户端各层和服务器端各层的职责分别如表5.1-1和表5.1-2所示。

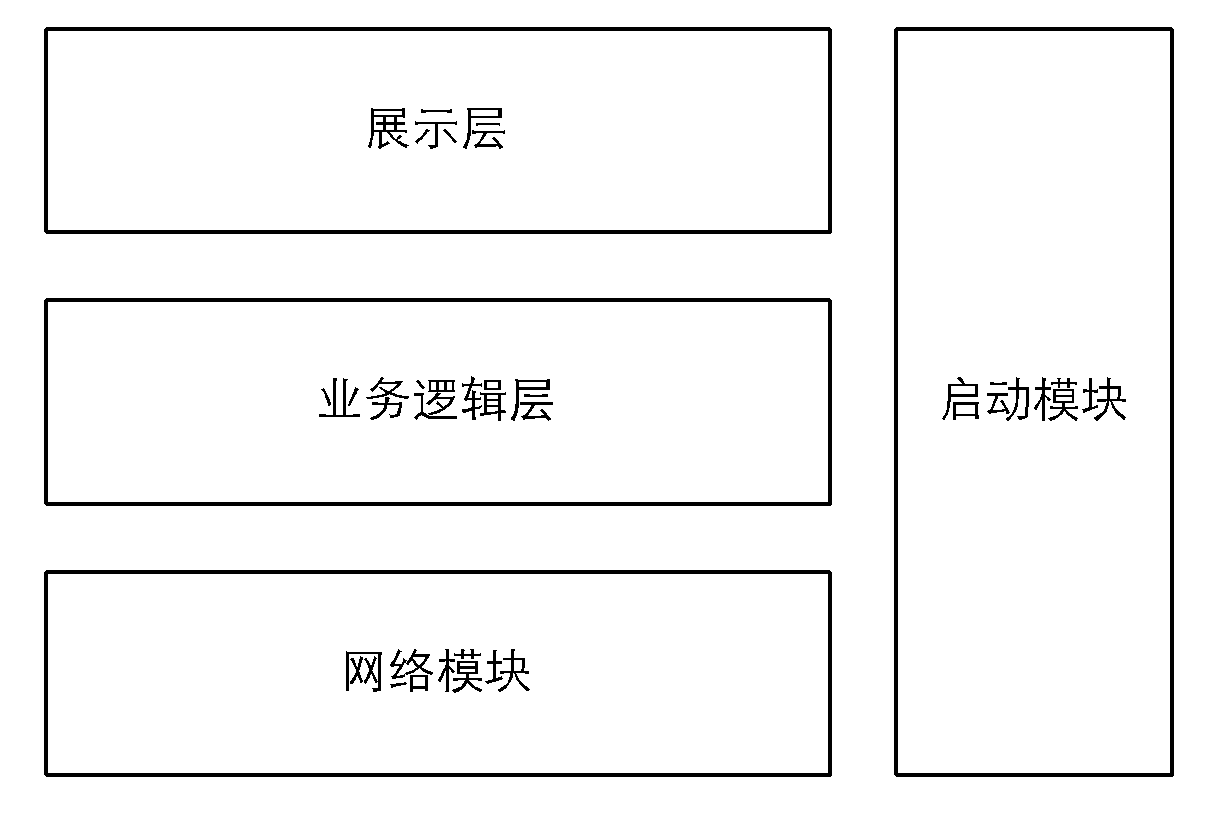


图5.1-1 客户端模块视图

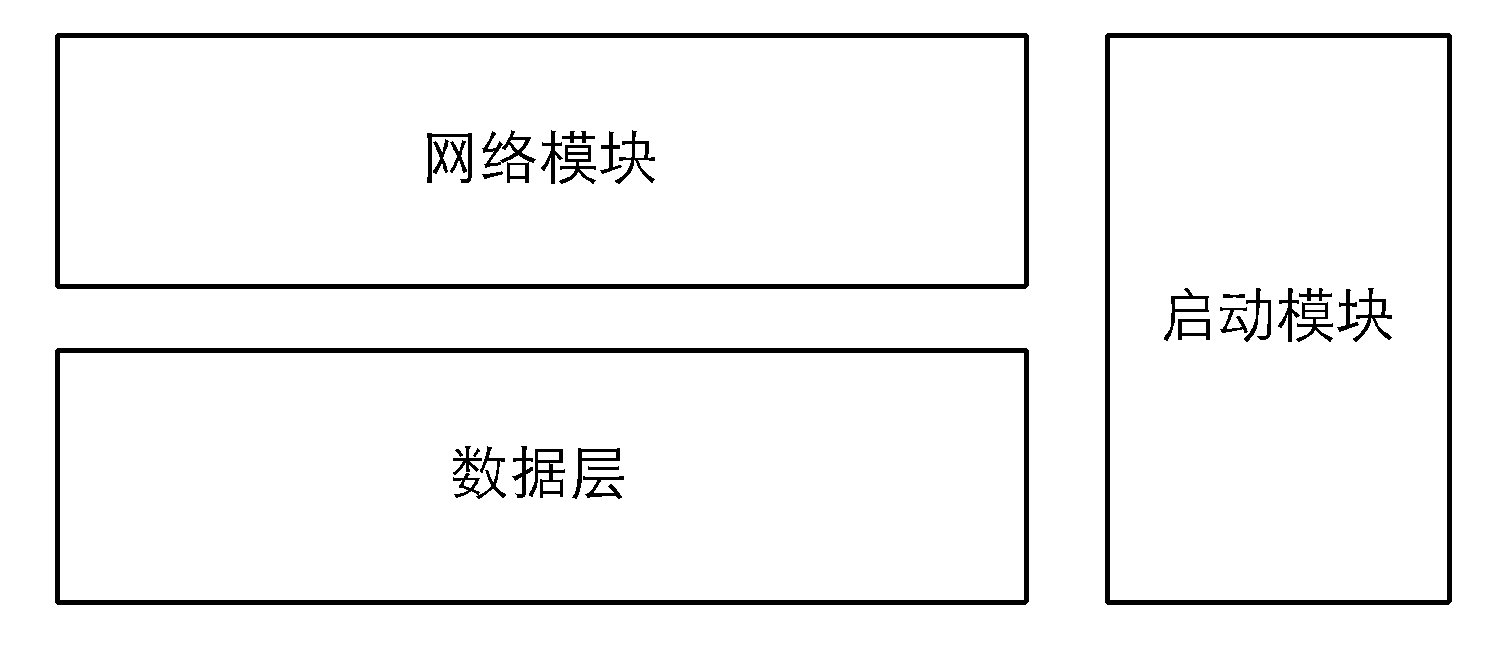


图5.1-2 服务器端模块视图

表5.1-1 客户端各层的职责

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化网络通信机制，启动用户界面。 |
| 用户界面层 | 基于窗口的酒店预订系统客户端用户界面。 |
| 业务逻辑层 | 对于用户界面的输入响应和业务处理逻辑。 |
| 客户端网络模块 | 利用Java RMI机制查找RMI服务 |

表5.1-2 服务器端各层的职责

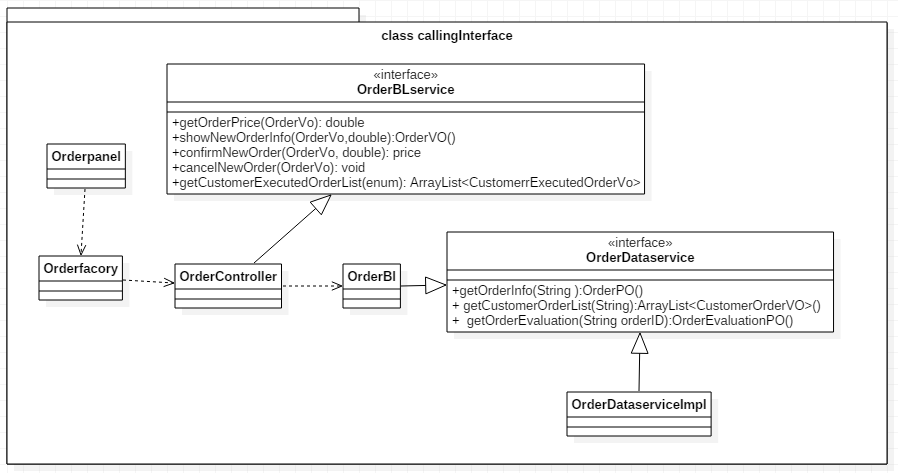
|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化网络通信机制，启动用户界面。 |
| 业务逻辑层 | 对于用户界面的输入响应和业务处理逻辑。 |
| 数据层 | 负责数据的持久化及数据访问接口。 |
| 服务器端网络模块 | 利用Java RMI机制开启RMI服务，注册RMI服务 |

每一层只是使用下方直接接触的层。层与层之间仅仅是通过接口的调用来完成的。层之间调用的接口如表5.1-3所示。

表5.1-3 层之间调用的接口

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| CustomerBLService  ManagerBLService  OrderBLService  PromotionBLService  HotelBLService  RoomBLService  CreditBLService | 客户端展示层 | 客户端业务逻辑层 |
| CustomerDataService  ManagerDataService  OrderDataService  PromotionDataService  HotelDataLService  CreditDataService  RoomDataService  DataFactory | 客户端业务逻辑层 | 服务器端数据层 |

借用订单处理用例来说明层之间的调用，如图5.1-3所示。每一层之间都是由上层依赖了一个接口（需接口），而下层实现这个接口（供接口）。OrderBLService提供了框架界面所需要的所有业务逻辑功能。OrderDataService提供了对数据库的增、删、改、查等操作。这样的实现就大大降低了层与层之间的耦合。



## 5.2用户界面层的分解

根据需求，系统存在39个用户界面：会员登陆界面、管理人员登录界面、会员主界面、酒店工作人员主界面、网站营销人员主界面、网站管理人员主界面、会员注册界面、查看个人信息界面、修改个人信息界面、查看信用记录界面、查看历史订单界面、浏览和撤销未执行订单界面、查看已执行订单正常订单界面、查看订单详情界面、生成评价界面、搜索酒店界面、获取酒店列表界面、查看住过的酒店界面、查看酒店详情界面、生成订单界面、管理可用客房界面、更新入住信息界面、更新退房信息界面、查看酒店订单界面、订单执行界面、查看酒店促销策略界面、制定酒店促销策略界面、查看酒店基本信息界面、修改酒店基本信息界面、查看网站促销策略界面、制定网站促销策略界面、浏览未执行订单和撤销异常订单界面、充值会员信用值界面、制定等级制度界面、添加网站营销人员界面、查看用户信息界面、修改用户信息界面、添加酒店界面、添加酒店工作人员界面。界面跳转如图5.2-1所示。

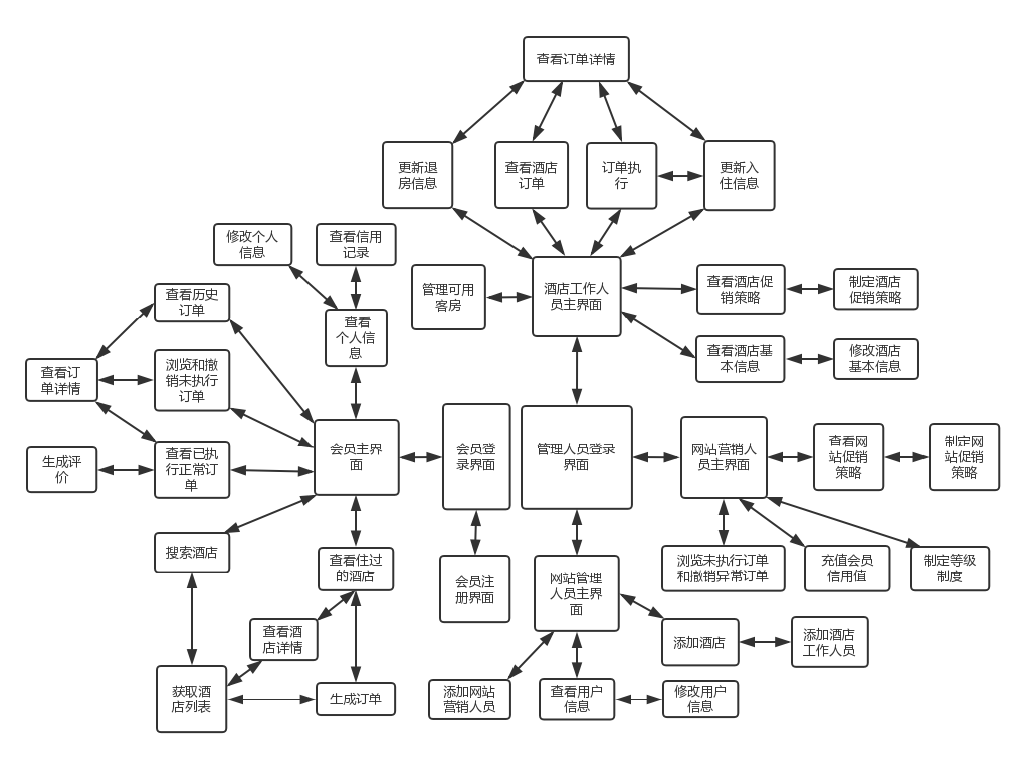


图5.2-1 用户界面跳转

服务器端和客户端的用户界面设计接口是一致的，只是具体的页面不一样。用户界面类如图5.2-2所示。

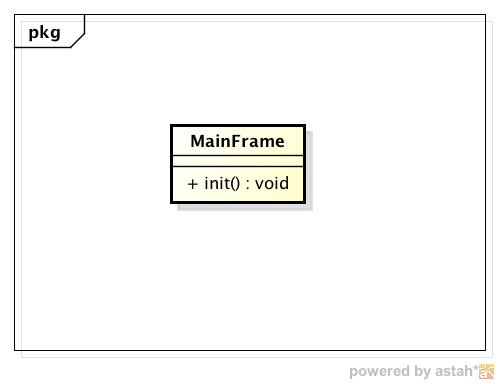


图5.2-2 用户界面类

### 5.2.1用户界面层模块的职责

如表5.2.1-1所示为用户界面层模块的职责。

表5.2.1-1 用户界面层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| MainAPP | 界面Main入口，负责界面的跳转和显示 |

### 5.2.2用户界面层模块的接口规范

用户界面层模块的接口规范如表5.2.2-1所示。

表5.2.2-1 用户界面层模块的接口规范

|  |  |  |
| --- | --- | --- |
| MainFrame | 语法 | Init(args[]) |
| 前置条件 | 无 |
| 后置条件 | 显示Frame以及LoginPanel |

用户界面层需要的服务接口如表5.2.2-2所示。

表5.2.2-2 用户界面层模块需要的服务接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| businesslogicservice.LoginBLService | 登陆界面的业务逻辑接口 |
| businesslogicservice.\*BLService | 每个界面都有一个相应的业务逻辑接口 |

### 5.2.3用户界面模块设计原理

用户界面利用JavaFX8来实现。

## 5.3业务逻辑层的分解

业务逻辑层包括多个针对界面的业务逻辑处理对象。例如:Customer对象负责处理会员相关的逻辑，Order对象负责处理订单处理的逻辑。业务逻辑层的设计如图5.3-1所示。

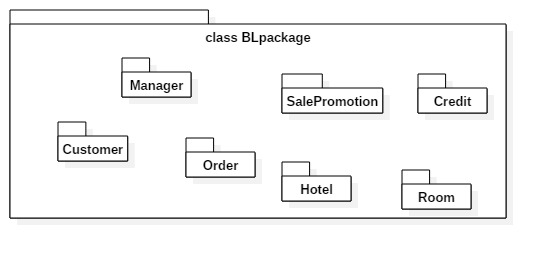


图5.3-1 业务逻辑层

### 5.3.1业务逻辑层模块的职责

业务逻辑层模块的职责如表5.3.1-1所示。

表5.3.1-1 业务逻辑层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| CustomerBL | 负责实现对应与登陆界面所需要的服务 |
| ManagerBL | 负责实现管理人员界面所需要的服务 |
| OrderBL | 负责实现订单处理界面所需要的服务 |
| HotelBL | 负责实现酒店管理界面所需要的服务 |
| PromotionBL | 负责实现促销策略界面所需要的服务 |
| CreditBL | 负责实现信用值改变和查询的服务 |
| RoomBL | 负责实现可用客房的录入和查询的服务 |

### 5.3.2业务逻辑层模块的接口规范

表5.3.2-1orderbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Order.newOrder | 语法 | public ResultMessage newOrder(OrderVO ovo); | |
| 前置条件 | 用户输入订单信息后，系统确定订单可以生成，即有足够的空房，并且信用值大于0 | |
| 后置条件 | 显示订单信息（包括价格），提示用户确定 | |
| Order.getOrderVOList | 语法 | public Map<Integer,OrderVO> getOrderVOList(); | |
| 前置条件 | 用户或酒店工作人员要查看自己的订单列表 | |
| 后置条件 | 系统显示订单列表 | |
| Order.getOrderEvaluationVOlist | 语法 | public Map<Integer,OrderEvaluationVO> getOrderEvaluationVOList(); | |
| 前置条件 | 用户或者酒店工作人员要查看自己的订单评价列表 | |
| 后置条件 | 系统显示订单评价列表 | |
| Order.getSpecificCustomerOrderList | 语法 | public Map<Integer,OrderVO> getSpecificCustomerOrderList(String customerID); | |
| 前置条件 | 启动查看某酒店特定用户的订单列表 | |
| 后置条件 | 显示订单列表，或启动查看特定类型订单列表的任务 | |
| Order.getSpecificHotelOrderList | 语法 | public Map<Integer,OrderVO> getSpecificHotelOrderList(String hotelName); | |
| 前置条件 | 启动查看某用户特定酒店下的订单列表任务 | |
| 后置条件 | 显示用户特定酒店订单列表或启动查看特定类型订单列表的任务 | |
| Order.getOrderInfo | 语法 | public OrderVO getOrderInfo(int orderID); | |
| 前置条件 | 启动查看特定订单信息的任务 | |
| 后置条件 | 显示订单信息 | |
| Order.getOrderEvaluationByID | 语法 | public OrderEvaluationVO getOrderEvaluationByID(int OrderID); | |
| 前置条件 | 启动查看特定订单评价的任务 | |
| 后置条件 | 显示订单评价详情 | |
| Order.evaluateOrder | 语法 | public ResultMessage evaluateOrder(OrderEvaluationVO oevo); | |
| 前置条件 | 启动订单评价 | |
| 后置条件 | 数据层更新订单评价 | |
| Order.executeOrder | 语法 | public boolean executeOrder(int orderID); | |
| 前置条件 | 启动执行订单 | |
| 后置条件 | 修改用户的信用值，数据层更新订单信息和用户信用值 | |
| Order.cancelOrder | 语法 | public ResultMessage cancelOrder(int orderID); | |
| 前置条件 | 启动取消订单任务 | |
| 后置条件 | 修改用户信用值，数据层更新订单信息和信用值 | |
| Order.updateCheckinInfo | 语法 | public boolean updateCheckinInfo(int orderID, String time); | |
| 前置条件 | 执行订单 | |
| 后置条件 | 数据层更新订单信息 | |
| Order.updateCheckoutInfo | 语法 | public boolean updateCheckoutInfo(int orderID, String time); | |
| 前置条件 | 退房 | |
| 后置条件 | 数据层更新订单信息 | |
| Order.getOrderListType | 语法 | public Map<Integer,OrderVO> getOrderListByType(OrderType type); | |
| 前置条件 | 获取特定类型的订单列表 | |
| 后置条件 | 系统显示订单列表 | |
| Order.getSpecificOrderListByType | 语法 | public Map<Integer,OrderVO> getSpecificOrderListByType(OrderType type, String name); | |
| 前置条件 | 获取特定酒店或者用户某种类型的订单 | |
| 后置条件 | 系统显示订单列表 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| CreditBLService.getNumCredit(String userName) | | | 获取会员信用值 |
| RoomBLservice.getAvailableRoomList(String hotelName, String startDate, String endDate) | | | 获取酒店可用客房列表 |
| CustomerBLService.updateCustomerCreditInfo(String userName, CreditVO creditVO, OrderVO orderVO) | | | 更新用户信用值 |
| OrderDataService.updateOrderEvaluation(OrderEvaluationPO oepo | | | 更新订单评价 |
| OrderDataService.updateOrder(OrderPO opo); | | | 更新订单信息 |
| OrderDataService.deleteOrder(OrderPO opo); | | | 取消订单 |
| CreditBLService.getNumCredit(String userName); | | | 获取用户信用值 |
| CreditBLService.updateCustomerCreditInfo(String userName, CreditVO creditVO, OrderVO orderVO)) | | | 更新用户信用值 |
| OrderDataService.insertOrder(OrderPO opo) | | | 在数据层中增加订单 |
| OrderDataService.getCustomerOrderList(String customerID) | | | 获取用户订单列表 |
| OrderDataService.getCustomerEvaluation(String CustommerID) | | | 获取用户评价列表 |
| OrderDataService.getHotelEvalvation(String hotel) | | | 获取酒店评价列表 |
| OrderDataService.getHotelOrderList(String hotel) | | | 获取酒店订单列表 |

表5.3.2-2 customerbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Customer.login | 语法 | public ResultMessage login(String userName, String password); | |
| 前置条件 | password符合输入规则 | |
| 后置条件 | 系统查找是否存在相应的Customer，根据输入的password返回登陆验证的结果 | |
| Customer.addCustomer | 语法 | public ResultMessage addCustomer(CustomerVO vo);; | |
| 前置条件 | 会员注册 | |
| 后置条件 | 系统将会员信息导入，创建相关会员 | |
| Customer.ChangetCustomernfo | 语法 | public ResultMessage changeCustomerInfo(CustomerVO vo); | |
| 前置条件 | 会员或网站管理人员输入了正确的修改的信息，网站管理人员修改用户信息 | |
| 后置条件 | 系统修改相关会员信息 | |
| Customer.getCustomerInfo | 语法 | public CustomerVO getCustomerInfo(String userName); | |
| 前置条件 | 获取会员基本信息 | |
| 后置条件 | 系统显示会员基本信息 | |
| Customer.successLog | 语法 | public CustomerVO successLog(String userName); | |
| 前置条件 | 成功登陆 | |
| 后置条件 | 返回用户信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| CustomerDataService.find(String userName, String password) | | | 根据会员ID和密码进行查找单一持久化对象 |
| CustomerDataService.insert(CustomerPO po) | | | 在数据库中插入CustomerPO对象 |
| CustomerDataService.update(CustomerPO po) | | | 在数据库中更新CustomerPO对象 |
| CustomerDataService.getCustomer(String userName) | | | 根据会员ID得到会员持久化对象 |

表5.3.2-3 promotionbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Promotion.getHotelPromotion | 语法 | public HotelPromotionVO getHotelPromotion(HotelPromotionType hotelPromotionType, String hotelName); | |
| 前置条件 | 无 | |
| 后置条件 | 系统返回一种酒店促销策略 | |
| Promotion.getWebPromotion | 语法 | public WebPromotionVO getWebPromotion(WebPromotionType webPromotionType); | |
| 前置条件 | 无 | |
| 后置条件 | 系统返回一种网站促销策略 | |
| Promotion.updateHotelPromotion | 语法 | public ResultMessage updateHotelPromotion(HotelPromotionType hotelPromotionType, HotelPromotionVO hpvo); | |
| 前置条件 | 无 | |
| 后置条件 | 系统更新一种酒店促销策略 | |
| Promotion.updateWebPromotion | 语法 | public ResultMessage updateWebPromotion(WebPromotionType webPromotionType, WebPromotionVO wpvo); | |
| 前置条件 | 无 | |
| 后置条件 | 系统更新一种网站促销策略 | |
| Promotion.calculateLevel | 语法 | public int calculateLevel(double credit); | |
| 前置条件 | 无 | |
| 后置条件 | 系统返回会员等级 | |
| Promotion.calculatePrice | 语法 | public OrderVO calculatePrice(OrderVO orderVO); | |
| 前置条件 | 填写了完整的订单信息 | |
| 后置条件 | 系统返回计算价格后的OrderVO | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| PromotionDataService.findHotelPromotion(HotelPromotionType hotelPromotionType, String hotelName) | | | 在数据库中得到一种酒店促销策略 |
| PromotionDataService.updateHotelPromotion(HotelPromotionType hotelPromotionType, HotelPromotionPO hppo) | | | 在数据库中更新一种酒店促销策略 |
| PromotionDataService.findWebPromotion(WebPromotionType webPromotionType) | | | 在数据库中得到一种网站促销策略 |
| PromotionDataService.updateWebPromotion(WebPromotionType webPromotionType, WebPromotionPO wppo) | | | 在数据库中更新一种网站促销策略 |
| CustomerBLService.getCustomerInfo(String customerID) | | | 得到会员信息 |
| HotelBLService.getHotelInfo(String hotelName) | | | 得到酒店信息 |

表5.3.2-4 hotelbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Hotel.getHotelInfo | 语法 | public HotelInfoVo getHotelInfo(String hotelName) | |
| 前置条件 | 无 | |
| 后置条件 | 返回某一酒店的详细信息 | |
| Hotel.modifyHotelInfo | 语法 | public ResultMessage modifyHotelInfo(HotelInfoVo hivo) | |
| 前置条件 | 输入了正确的酒店修改信息 | |
| 后置条件 | 修改酒店信息，返回修改结果 | |
| Hotel.getHistoryHotelList | 语法 | public ArrayList<HotelInfoVo>getHistoryList(String customerID) | |
| 前置条件 | 无 | |
| 后置条件 | 返回历史酒店列表 | |
| Hotel.getHotelList | 语法 | public ArrayList<HotelInfoVo>getHotelList (String customerID, SearchInfoVo sivo) | |
| 前置条件 | 输入了正确的搜索条件 | |
| 后置条件 | 返回酒店列表 | |
| Hotel.addHotel | 语法 | public ResultMessage addHotel(HotelInfoVO hivo) | |
| 前置条件 | 无 | |
| 后置条件 | 添加一个酒店 | |
| Hotel.sort | 语法 | public ArrayList<HotelInfoVO> sort(ArrayList<HotelInfoVO> hotelList, SortType sortType, SortWay sortWay) | |
| 前置条件 | 无 | |
| 后置条件 | 返回排序后的酒店列表 | |
| Hotel.updateHotelScore | 语法 | public ResultMessage updateHotelScore(String hotelName, double score); | |
| 前置条件 | 用户评价订单 | |
| 后置条件 | 更新酒店评分 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| HotelDataService.findHotelInfo（String hotelName） | | | 根据酒店名称查找单个酒店详细信息 |
| HotelDataService.updateHotelInfo（HotelInfoPO hipo） | | | 更新酒店详细信息 |
| HotelDataService.findHotels() | | | 获取所有酒店列表 |
| HotelDataService.insertHotel(HotelInfoPO hipo) | | | 根据酒店信息添加酒店 |
| HotelDataService.updateHotelScore(String hotelName, double score) | | | 更新酒店评分 |
| Order.getOrderVOList() | | | 得到用户所有订单列表 |

表5.3.2-5 managerbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Manager.addManager | 语法 | public ResultMessage addManager ( ManagerVo mvo) | |
| 前置条件 | 管理人员信息合法 | |
| 后置条件 | 系统创建相应管理人员，更新相应信息 | |
| Manager.changeManagerInfo | 语法 | public ResultMessage changemanagerInfo ( ManagerVo mvo) | |
| 前置条件 | 管理人员信息合法 | |
| 后置条件 | 系统更新其信息 | |
| Manager.login | 语法 | publicResultMessage login (ManagerType type，String username, String password) | |
| 前置条件 | Password符合输入规则 | |
| 后置条件 | 系统查找是否存在对应用户，并根据password返回验证结果 | |
| Manager.getManagerInfo | 语法 | public ManagerVO getManagerInfo(ManagerType managerType, String username); | |
| 前置条件 | 无 | |
| 后置条件 | 根据用户类型和用户名返回ManagerVO | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| ManagerDataService.findManagerInfo(ManagerType managerType, String username) | | | 在数据库查找指定的管理人员信息 |
| ManagerDataService.insertManager(ManagerPo po) | | | 在数据库中插入新的管理人员 |
| ManagerDataService.updateManager(ManagerPo po) | | | 在数据库中更新管理人员信息 |
| ManagerDataService.check(ManagerType managerType, String username, String password) throws RemoteException | | | 管理人员登录校验 |

表5.3.2-6 creditbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Credit.getCustomerCreditInfo | 语法 | public List<CreditVO> getCustomerCreditInfo(String userName); | |
| 前置条件 | 启动查看信用值变化 | |
| 后置条件 | 系统显示信用值变化记录 | |
| Credit.getNumCredit | 语法 | public double getNumCredit(String userName); | |
| 前置条件 | 启动查看信用值任务 | |
| 后置条件 | 系统显示信用值 | |
| Credit.increaseCredit | 语法 | public void increaseCredit(String userName, int creditChange); | |
| 前置条件 | 启动信用充值任务 | |
| 后置条件 | 系统更新信用值变化 | |
| Credit.updateCustomerCreditInfo | 语法 | public ResultMessage updateCustomerCreditInfo(String userName, CreditVO creditVO, OrderVO orderVO); | |
| 前置条件 | 启动更新信用值信息任务（订单执行、订单异常、订单撤销） | |
| 后置条件 | 数据层更新信用值 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| CreditDataService.getCustomerCredits(String userName) | | | 获取用户信用值变化的记录 |
| CreditDataService.insert(CreditInfoPO creditInfoPO) | | | 在数据库中插入新的信用值记录 |

表5.3.2-7 roombl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Room.getAvailableRoomList | 语法 | public ArrayList<AvailableRoomVO> getAvailableRoomList(String hotelName, String startDate, String endDate); | |
| 前置条件 | 无 | |
| 后置条件 | 系统显示可用客房列表 | |
| Room.addAvailableRooms | 语法 | public ResultMessage addAvailableRooms(AvailableRoomVO arvo); | |
| 前置条件 | 无 | |
| 后置条件 | 系统增加可用客房数量 | |
| Room.deleteAvailableRooms | 语法 | public ResultMessage deleteAvailableRooms(AvailableRoomVO arvo); | |
| 前置条件 | 无 | |
| 后置条件 | 系统减少可用客房数量 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| RoomDataService.findAvailableRooms(String hotelName) | | | 从数据库得到某酒店可用客房 |
| RoomDataService.updateAvailableRooms(AvailableRoomPO arpo) | | | 在数据库中更新可用客房 |
| RoomDataService.insertAvailableRooms(AvailableRoomPO arpo) | | | 在数据库中插入可用客房 |
| Order.getOrderVOList() | | | 得到酒店所有订单列表 |

## 5.4数据层的分解

数据层主要给业务逻辑层提供数据防伪服务，包括对于持久化数据的增、删、改、查。由于持久化数据的保存可能存在多种形式：Txt文件、序列化文件、数据库等，所示抽象了数据服务。数据层模块的具体描述如图5.4所示。

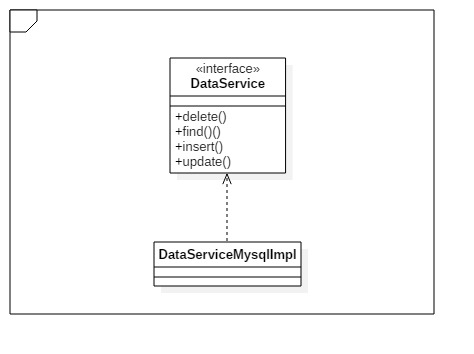


图5.4 数据层模块的描述

### 5.4.1数据层模块的职责

数据层模块的职责如表5.4.1所示。

表5.4.1 数据层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| DataService | 持久化数据库的接口，提供集体载入，集体保存、增、删、改、查服务。 |
| DataServiceSerializableImpl | 基于序列化文件的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务 |
| DataServiceMysqlImpl | 基于mysql的持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务 |

### 5.4.2数据层模块的接口规范

表5.4.2-1 数据层Order模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderDataService.getOrderInfo(int orderID) | 语法 | public OrderPO getOrderInfo(int orderID) throws RemoteException; |
| 前置条件 | 启动查看订单详情 |
| 后置条件 | 按ID进行查找返回相应的OrderPO结果 |
| OrderDataService.getCustomerOrderList(String customerID) | 语法 | public ArrayList<CustomerOrderVO> getCustomerOrderList(String customerID) |
| 前置条件 | 无 |
| 后置条件 | 显示用户订单列表 |
| OrderDataService.getHotelOrderList(String hotel) | 语法 | public Map<Integer, OrderPO> getHotelOrderList(String hotel) throws RemoteException; |
| 前置条件 |  |
| 后置条件 | 显示酒店订单列表 |
| OrderDataService.getUnexcutedOrderList | 语法 | public ArrayList<OrderPO> getUnexcutedOrderList() throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 获取未执行订单列表 |
| OrderDataService.getAbnormalOrderList() | 语法 | public ArrayList<OrderPO> getAbnormalOrderList() throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 获取一场订单列表 |
| OrderDataService.getOrderEvaluation(int orderID) | 语法 | public OrderEvaluationPO getOrderEvaluation(int orderID) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 获取订单评价PO |
| OrderDataService.updateOrderEvaluation(OrderEvaluationPO oepo) | 语法 | public Map<Integer, OrderEvaluationPO> updateOrderEvaluation(OrderEvaluationPO oepo)throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 更新订单评价 |
| OrderDataService.getHotelEvalvation(String hotel) | 语法 | public Map<Integer, OrderEvaluationPO> getHotelEvalvation(String hotel) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 获取酒店评价列表 |
| OrderDataService.getCustomerEvaluation(String CustommerID) | 语法 | public Map<Integer, OrderEvaluationPO> getCustomerEvaluation(String CustommerID) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 获取用户评价 |
| OrderDataService.insertOrder(OrderPO opo) | 语法 | public Map<Integer, OrderPO> insertOrder(OrderPO opo) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 增加一个订单PO |
| OrderDataService.deleteOrder(OrderPO opo) | 语法 | public Map<Integer, OrderPO> deleteOrder(OrderPO opo) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 删除一个订单 |
| OrderDataService.updateOrder(OrderPO opo) | 语法 | public Map<Integer, OrderPO> updateOrder(OrderPO opo) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 更新一个订单 |
|  |  |  |

表5.4.2-2 数据层Customer模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CustomerDataService.find(String userName, String password) | 语法 | public ResultMessage find(String userName, String password) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的CustomerPO结果 |
| CustomerDataService.insert(CustomerPO po) | 语法 | Public void insert(CustomerPO po ) throws RemoteException; |
| 前置条件 | 同样ID的po在CustomerDataService中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| CustomerDataService.update(CustomerPO po) | 语法 | Public void update(CustomerPO po ) throws RemoteException; |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 更新一个po |
| CustomerDataService.getCustomer(String userName) | 语法 | Public CustomerPO getCustomer(String userName ) throws RemoteException; |
| 前置条件 | 在数据库中存在同样ID的PO |
| 后置条件 | 按ID进行查找返回PO对象 |

表5.4.2-3 数据层promotion模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PromotionDataService.findHotelPromotion | 语法 | public HotelPromotionPO findHotelPromotion(HotelPromotionType hotelPromotionType, String hotelName) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 在数据库中查找某一酒店促销策略 |
| PromotionDataService.updateHotelPromotion | 语法 | public boolean updateHotelPromotion(HotelPromotionType hotelPromotionType, HotelPromotionPO hppo) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 在数据库中更新某一酒店促销策略 |
| PromotionDataService.findWebPromotion | 语法 | public WebPromotionPO findWebPromotion(WebPromotionType webPromotionType) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 在数据库中查找某一网站促销策略 |
| PromotionDataService.updateWebPromotion | 语法 | public boolean updateWebPromotion(WebPromotionType webPromotionType, WebPromotionPO wppo) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 在数据库中更新某一网站促销策略 |

表5.4.2-4 数据层hotel模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| HotelDataService.findHotelInfo | 语法 | public HotelInfoPO findHotelInfo (String hotelName) |
| 前置条件 | 无 |
| 后置条件 | 根据酒店名称查找并返回相应的HotelInfoPO结果 |
| HotelDataService.updateHotelInfo | 语法 | public void updateHotelInfo (HotelInfoPO hipo) |
| 前置条件 | 在数据库中存在同样ID的HotelInfoPO |
| 后置条件 | 更新一个HotelInfoPO |
| HotelDataService.findHotels | 语法 | public ArrayList<HotelInfoPO>findHotels (SearchInfoPO sipo) |
| 前置条件 | 无 |
| 后置条件 | 根据搜索信息查找并返回多个HotelInfoPO结果 |
| HotelDataService.insertHotel | 语法 | public boolean insertHotel(HotelInfoPO hipo) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 在数据库中插入一个酒店 |
| HotelDataService.updateHotelScore | 语法 | public boolean updateHotelScore(String hotelName, double score) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 更新酒店评分 |

表5.4.2-5 数据层manager模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ManagerDataService.check | 语法 | public ResultMessage check(ManagerType managerType, String username, String password) |
| 前置条件 | 无 |
| 后置条件 | 按管理人员类型和username进行查找,若找到则检查其password，返回相应的ResultMessage结果 |
| ManagerDataService.insert | 语法 | public ResultMessage insert(ManagerType type, ManagerPo po) |
| 前置条件 | 同样ID、type的ManagerPo对象在数据库中不存在 |
| 后置条件 | 在数据库中添加新的ManagerPo对象 |
| ManagerDataService.update | 语法 | public ResultMessage update(ManagerType type, ManagerPo po); |
| 前置条件 | 需要更新数据库中的某个ManagerPo对象 |
| 后置条件 | 在数据库中更新对应ManagerPo对象 |
| ManagerDataService.findManagerInfo | 语法 | public ManagerPO findManagerInfo(ManagerType managerType, String username) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 按managerType和username进行查找返回相应的ManagerPO结果 |

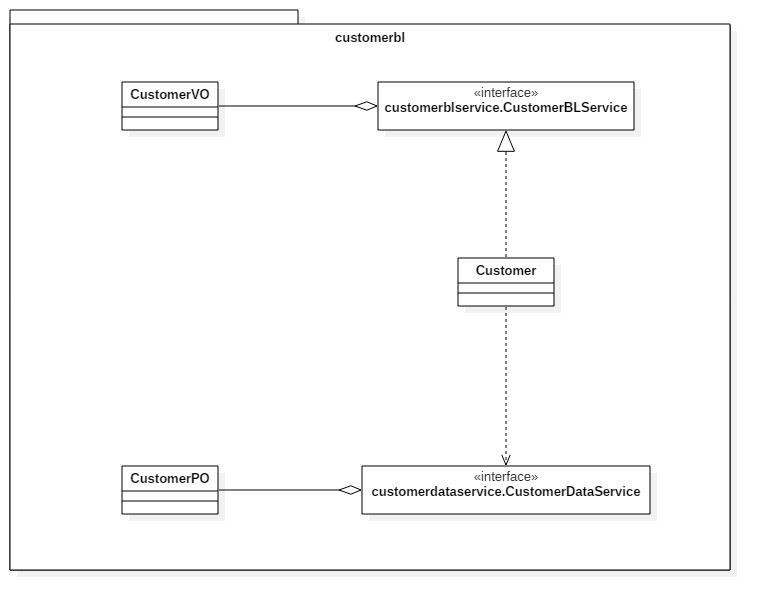
表5.4.2-6 数据层Credit模块的接口规范

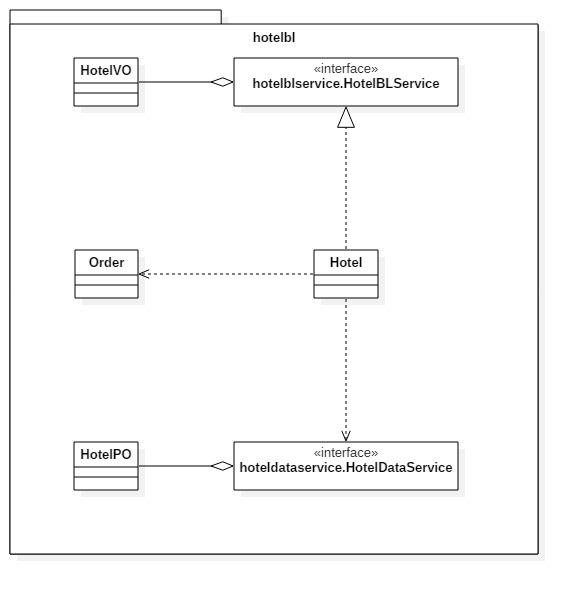
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CreditDataService.getCustomerCredits(String userName) | 语法 | public ArrayList<CreditInfoPO> getCustomerCredits(String userName)throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 获取信用值变化记录 |
| CreditDataService.insert(CreditInfoPO creditInfoPO) | 语法 | public ResultMessage insert(CreditInfoPO creditInfoPO)throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 插入一条信用值变化记录 |

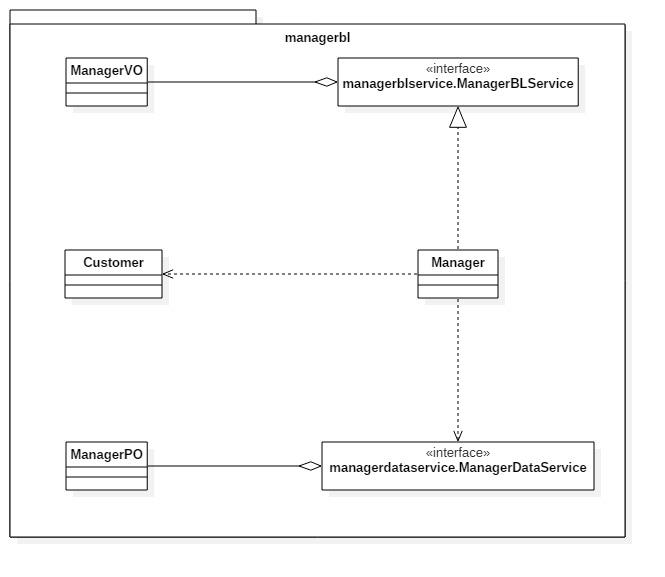
表5.4.2-7 数据层Room模块的接口规范

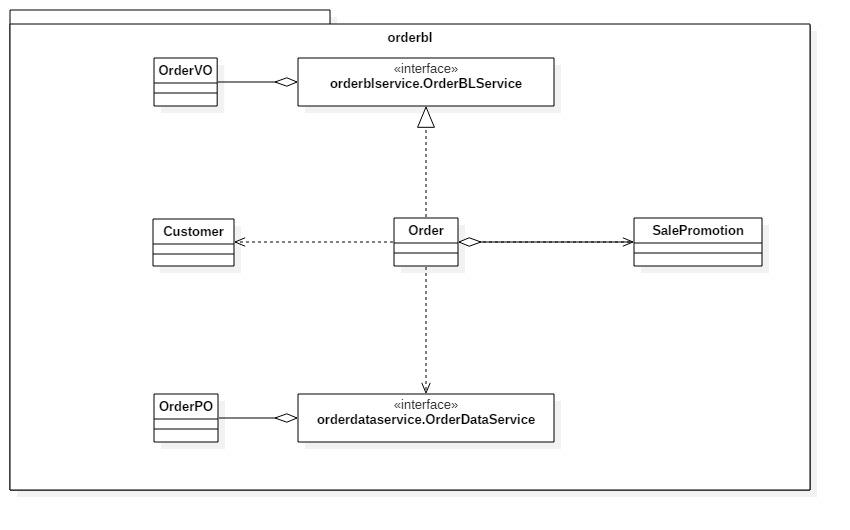
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| RoomDataService.findAvailableRooms | 语法 | public ArrayList<AvailableRoomPO> findAvailableRooms(String hotelName) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 获取可用客房列表 |
| RoomDataService.updateAvailableRooms | 语法 | public boolean updateAvailableRooms(AvailableRoomPO arpo) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 更新可用客房 |
| RoomDataService.insertAvailableRooms | 语法 | public boolean insertAvailableRooms(AvailableRoomPO arpo) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 插入可用客房 |

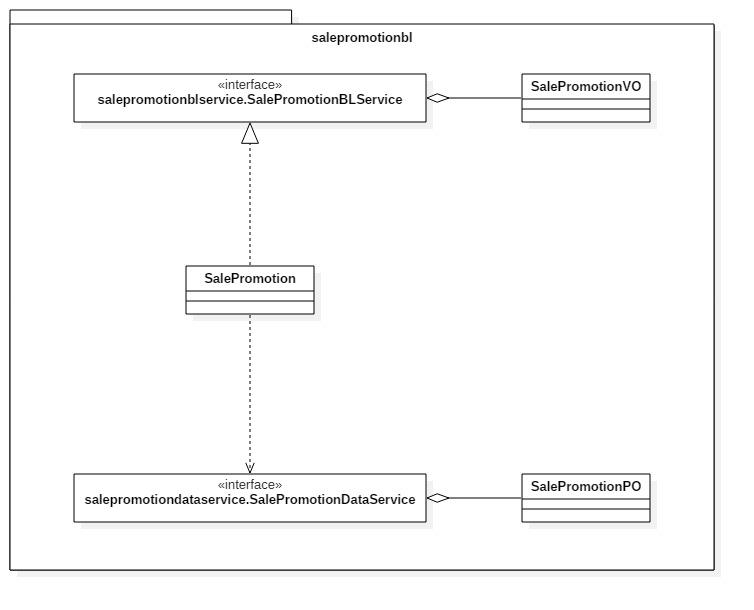
## 5.5模块的关键类图











# 6．信息视角

## 6.1数据持久化对象

|  |  |
| --- | --- |
| 类名 | 包含的属性 |
| AvailableRoomPO | 可用客房类型，数量，价格 |
| CustomerPO | 用户ID，用户名，用户密码，信用值（CreditInfoPO） |
| CreditInfoPO | 信用值 |
| HotelInfoPO | 酒店名，地址，所属商圈，酒店简介，星级，酒店工作人员（ManagerPO），设施服务，合作企业 |
| ManagerPO | 管理人员类型，用户ID，用户名，用户密码 |
| OrderPO | 订单编号，下单时间，订购者ID，订单状态，订单价值，撤销时间，入住时间，退房时间，最晚订单执行时间，房间类型，房间数量，预计入住人数，有无儿童， |
| OrderEvaluationPO | 订单ID，评分，评价 |
| PromotionPO | 促销类型，开始时间，结束时间，有效商圈，有效酒店，折扣力度，有效等级，有效合作企业，至少预定的房间数 |
| SearchInfoPO | 会员编号，所在城市，所属商圈，酒店名称,星级,评分下限，评分上限，房间类型，原始价格下限，原始价格上限，入住时间，退房时间，房间数量 |