|  |
| --- |
| 互联网酒店管理系统  Hotel Property Management System |
| 软件体系结构描述文档 |
| V3.0 |

|  |
| --- |
| 焦立群、金灵益、黎奥邦、李恩铭  2017-1-1 |

更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **日期** | **变更原因** | **版本号** |
| **焦立群** | 2016-10-11 | 最初文档模板 | V0.0 |
| **焦立群** | 2016-10-11 | 添加模块职责，用户界面层的分解，useinforbl模块的接口规范 | V0.1 |
| **李恩铭** | 2016-10-12 | 添加逻辑视角，组合视角，业务逻辑层accountbl模块的接口规范 | V0.2 |
| **金灵益** | 2016-10-12 | 添加数据层的分解，信息视角，业务逻辑层hotelbl模块的接口规范 | V0.3 |
| **焦立群** | 2016-10-12 | 修改了文档的排版 | V0.4 |
| **黎奥邦** | 2016-10-13 | 添加orderbl模块、promotionbl模块的接口规范 | V0.5 |
| **焦立群** | 2016-10-14 | 文档通过评审，将add、delete、find功能添加至accountbl模块接口下，定义了一些名称规范 | V1.0 |
| **焦立群** | 2016-10-15 | 将userinfo模块更改为member模块；在每个业务逻辑模块需接口下新增\*DatabaseServiceMySqlImpl | V1.1 |
| **金灵益** | 2016-10-16 | 删除hotelbl模块update接口，增加showHotelList,showHotelInfo,sortHotel接口，修改getReadyRoom返回类型 | V1.2 |
| **李恩铭** | 2016-10-16 | 在account模块中新增checkMemberName,checkInput,  getNewID,Login,logout | V1.3 |
| **黎奥邦** | 2016-10-16 | 修改了order与promotion在logic层里接口的一些变量和方法 | V1.4 |
| **金灵益** | 2016-10-29 | 撰写详细描述文档时，连带修改了HotelDataService模块接口规范，增加keyFind,getComment接口；修改hotelbl模块showHotelInfo接口规范，修改getPoStrings,增加givePoString；增加orderdataservice的find重载 | V1.5 |
| **焦立群** | 2016-11-13 | 详细描述文档中order模块有较大改动，连带改动体系结构文档 | V1.6 |
| **金灵益** | 2016-11-13 | 详细描述文档中增加HotelRoom、HotelRoomList接口规范，增加HotelList接口内多个方法，对体系结构文档连带修改 | V1.7 |
| **李恩铭** | 2016-12-05 | 在编码阶段对供需接口方法进行了改善设计，由李恩铭整合修改了各个接口的表格 | V1.8 |
| **李恩铭** | 2017-01-01 | 项目完成之前对所有文档进行回顾，进行了全面修正和完善 | V2.0  最终版 |

目录

[更新历史 1](#_Toc464304520)

[1. 引言 3](#_Toc464304521)

[1.1 编制目的 3](#_Toc464304522)

[1.2 词汇表 3](#_Toc464304523)

[1.3 参考资料 3](#_Toc464304524)

[2. 产品概述 3](#_Toc464304525)

[3. 逻辑视角 3](#_Toc464304526)

[4. 组合视角 5](#_Toc464304527)

[4.1 开发包图 5](#_Toc464304528)

[4.2 运行时进程 8](#_Toc464304529)

[4.3 物理部署 9](#_Toc464304530)

[5. 接口视角 10](#_Toc464304531)

[5.1 模块的职责 10](#_Toc464304532)

[5.2 用户界面层的分解 12](#_Toc464304533)

[5.3 业务逻辑层的分解 15](#_Toc464304534)

[5.4 数据层的分解 19](#_Toc464304535)

[6. 信息视角 24](#_Toc464304536)

[6.1 数据持久化对象 24](#_Toc464304537)

[6.2 Txt持久化格式 25](#_Toc464304538)

1. 引言
   1. 编制目的

本报告详细完成对互联网酒店管理系统的概要设计，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。

本报告面向开发人员测试人员及最终用户编写，是了解系统的导航。

* 1. 词汇表

|  |  |  |
| --- | --- | --- |
| **词汇名称** | **词汇含义** | **备注** |
| **HPMS** | 互联网酒店管理系统 | Hotel Property Management System |
| **Order** | 订单 | 无 |
| **Member** | 客户信息 | 企业、普通客户 |
| **Hotel** | 酒店 | 无 |
| **Account** | 账户 | 无 |
| **HotelPromotion** | 酒店促销策略 | 无 |
| **MarketPromotion** | 网站促销策略 | 无 |
| **Credit** | 客户信用值 | 无 |

* 1. 参考资料

互联网酒店管理系统HPMS软件需求规格说明\_V2.0

互联网酒店管理系统HPMS用例文档\_V2.0

1. 产品概述

参考互联网酒店管理系统用例文档和互联网酒店管理系统软件需求规格说明中对产品的概括描述。

1. 逻辑视角

互联网酒店管理系统中，选择了分层体系结构的风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好的示意整个高层抽象。展示层包括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1和图2所示。

图 1参照体系结构风格的包图表达逻辑视角

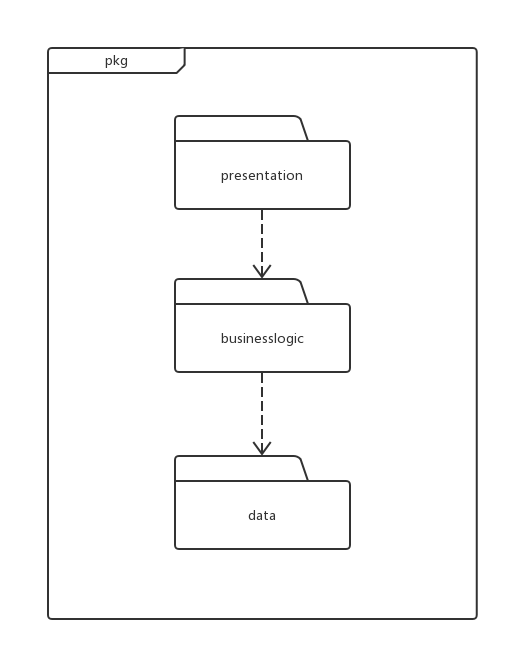
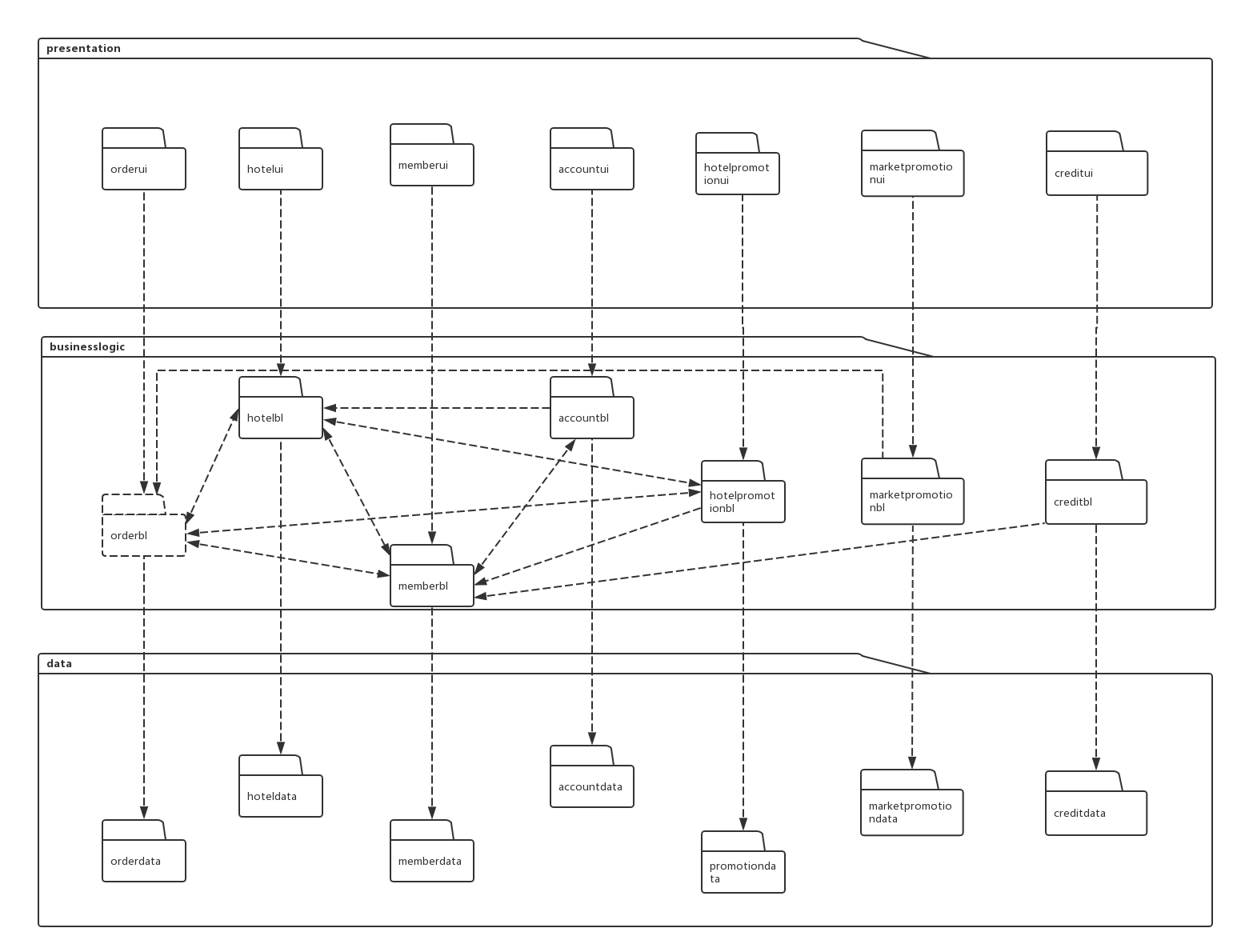


图2 软件体系结构逻辑设计方案

1. 组合视角
   1. 开发包图

互联网酒店管理系统的最终开发包设计如表1所示

表1 互联网酒店管理系统的最终开发包设计

|  |  |
| --- | --- |
| **开发（物理包）** | **依赖的其他开发包** |
| **mainui** | orderui, hotelui, userInfoui, accountui, promotionui |
| **orderui** | orderblservice, 界面类库包 |
| **orderblservice** |  |
| **orderbl** | orderblservice, hotelbl, userInfobl, promotionbl, po |
| **orderdataservice** | Java RMI, po |
| **orderdata** | orderdataservice, databaseutility |
| **hotelui** | hotelblservice, 界面类库包 |
| **hotelblservice** |  |
| **hotelbl** | hotelblservice, orderbl, userInfobl, promotionbl, po |
| **hoteldataservice** | Java RMI, po |
| **hoteldata** | hoteldataservice, databaseutility |
| **memberui** | memberblservice, 界面类库包 |
| **memberblservice** |  |
| **memberbl** | memberblservice, accountbl, orderbl, hotelbl, po |
| **memberdataservice** | Java RMI, po |
| **memberdata** | userInfodataservice, databaseutility |
| **accountui** | accountblservice, 界面类库包 |
| **accountblservice** |  |
| **accountbl** | accountblservice, userInfobl, hotelbl, po |
| **accountdataservice** | Java RMI, po |
| **accountdata** | accountdataservice, databaseutility |
| **promotionui** | promotionblservice, 界面类库包 |
| **promotionblservice** |  |
| **promotionbl** | promotionblservice, orderbl, hotelbl, po |
| **hotelpromotiondataservice** | Java RMI, po |
| **hotelPromotiondata** | promotiondataservice, databaseutility |
| **marketPromotion dataservice** | Java RMI, po |
| **marketpromotiondata** | **marketpromotion**dataservice, databaseutility |
| **creditdataservice** | Java RMI, po |
| **creditdata** | creditdataservice, databaseutility |
| **vo** |  |
| **po** |  |
| **utilitybl** |  |
| **界面类库包** |  |
| **Java RMI** |  |
| **database utility** |  |

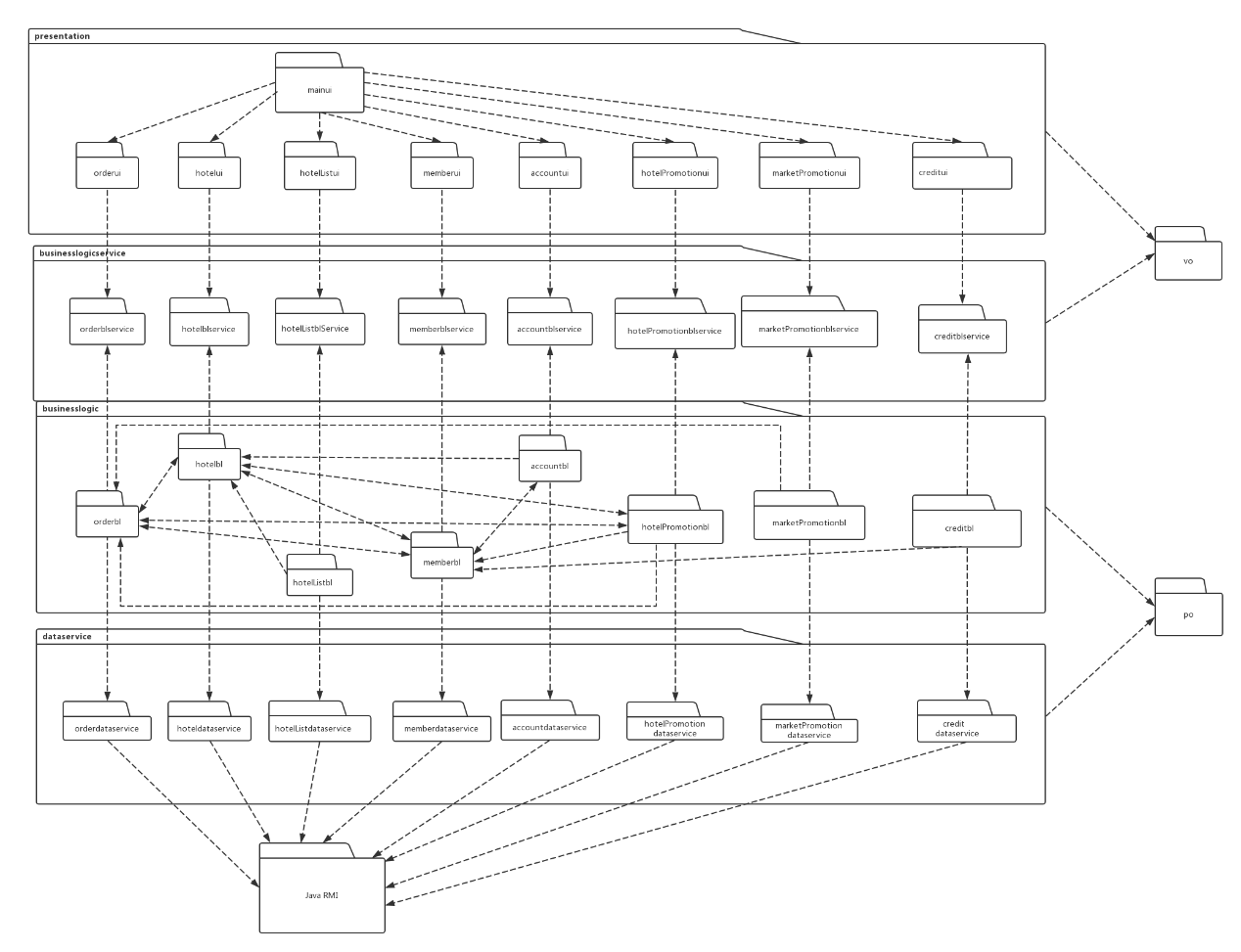


图 3 客户端开发包图

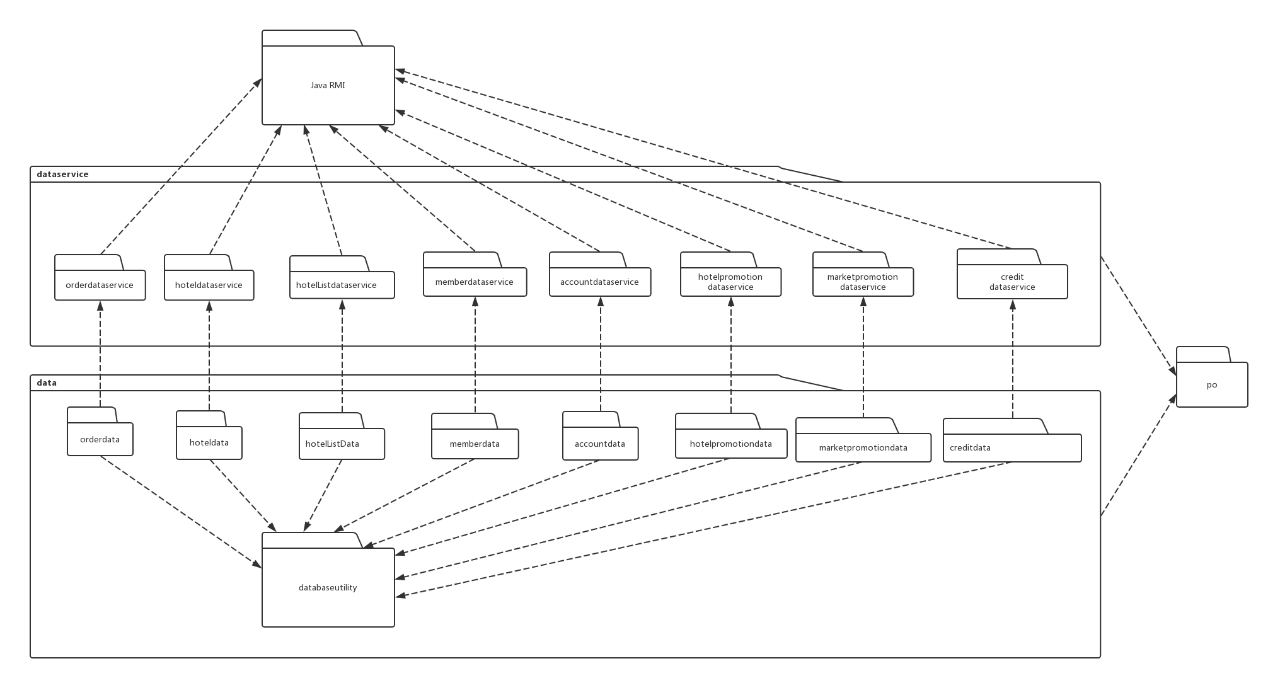


图 4 服务器端开发包图

* 1. 运行时进程

在互联网酒店管理系统中，会有多个客户端进程和一个服务器端进程，其进程图如图5所示。结合部署图，客户端进程是在客户端机器上运行，服务器端进程是在服务器端机器上运行。

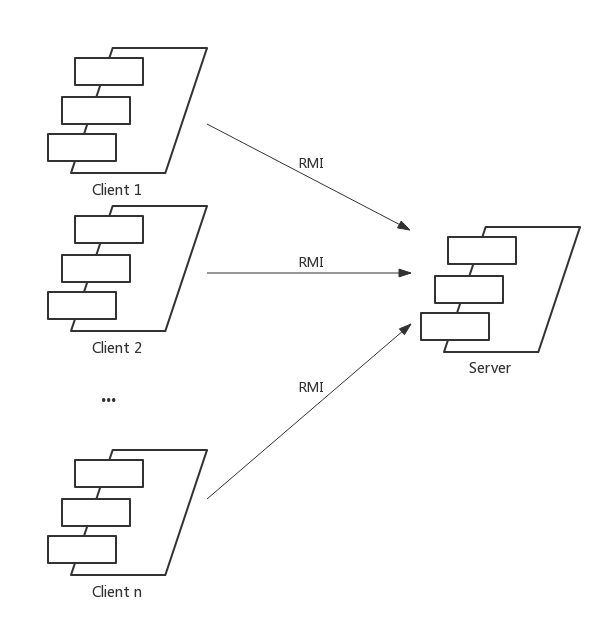


图5 进程图

* 1. 物理部署

互联网酒店管理系统中客户端构件是放在客户端机上，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Java RMI 构件属于JDK 1.8的一部分。所以，在系统JDK环境已经设置好的情况下，不需要独立部署。部署图如图6所示。

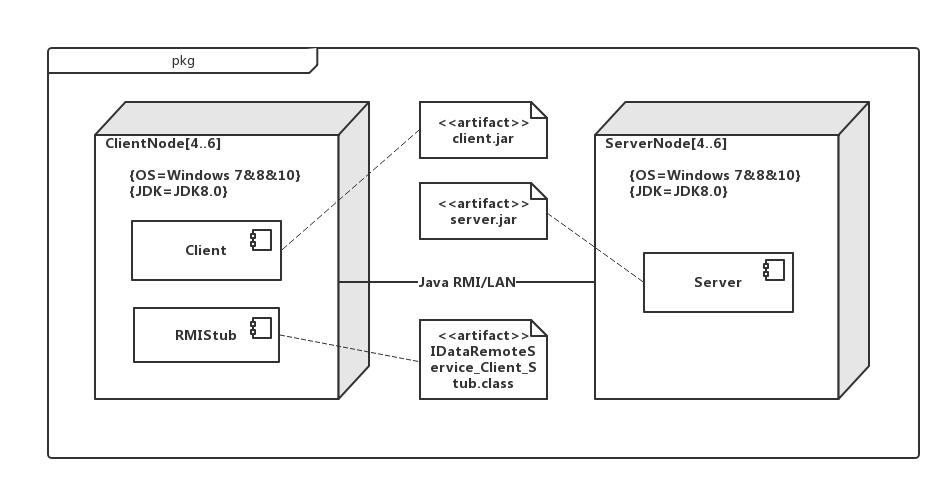


图6 部署图

1. 接口视角
   1. 模块的职责

客户端模块和服务器端模块视图分别如图和图所示。客户端各层和服务器端各层的职责分别如表和表所示。

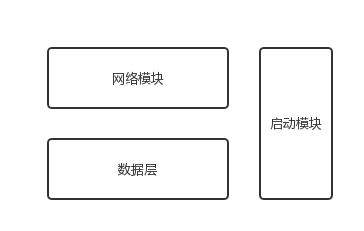
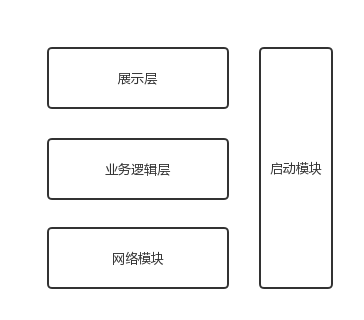


图7 客户端模块视图 图8 服务器端模块视图

表2 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** | 基于窗口的互联网酒店管理系统客户端用户界面 |
| **业务逻辑层** | 对于用户界面的输入进行响应并进行业务处理逻辑 |
| **客户端网络模块** | 利用Java RMI机制查找RMI服务 |

表3 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 |
| **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所示。

表4 层之间调用的接口

|  |  |  |
| --- | --- | --- |
| **接口** | **服务调用方** | **服务提供方** |
| **OrderBLService**  **HotelBLService**  **UserInfoBLService**  **AccountBLService**  **HotelPromotionBLService**  **MarketPromotionBLService**  **CreditBLService** | 客户端展示层 | 客户端业务逻辑层 |
| **OrderDataService**  **HotelDataService**  **UserInfoDataService**  **AccountDataService**  **PromotionDataService**  **MarketPromotionDataService**  **CreditDataService**  **DatabaseFactory** | 客户端业务逻辑层 | 服务器端数据层 |

借用客户用例来说明层之间的调用，如图所示。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）。UserInfoBLService提供了UserInfo界面所需要的所有业务逻辑功能。UserInfoDataService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。

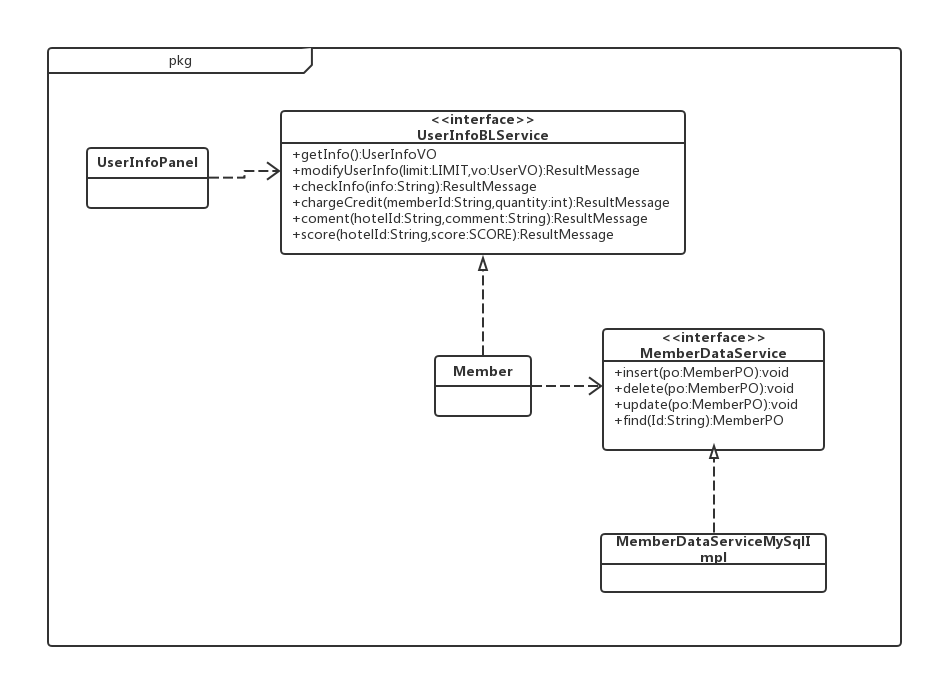


图9 客户用例层之间调用的接口

* 1. 用户界面层的分解

根据需求，系统存在23个用户界面：登陆界面、客户主界面、酒店主界面、网站营销人员主界面、网站管理人员主界面、个人基本信息界面、评价酒店服务界面、订单信息界面、提交订单界面、撤销订单界面、注册会员界面、获取酒店信息界面、酒店基本信息、执行客户订单界面、恢复客户订单界面、更新退房信息界面、更新入住信息界面、制定酒店促销策略界面、制定网站促销策略界面、信用充值界面、撤销异常订单界面、分析未执行订单情况界面、调整用户界面。界面跳转如图所示。

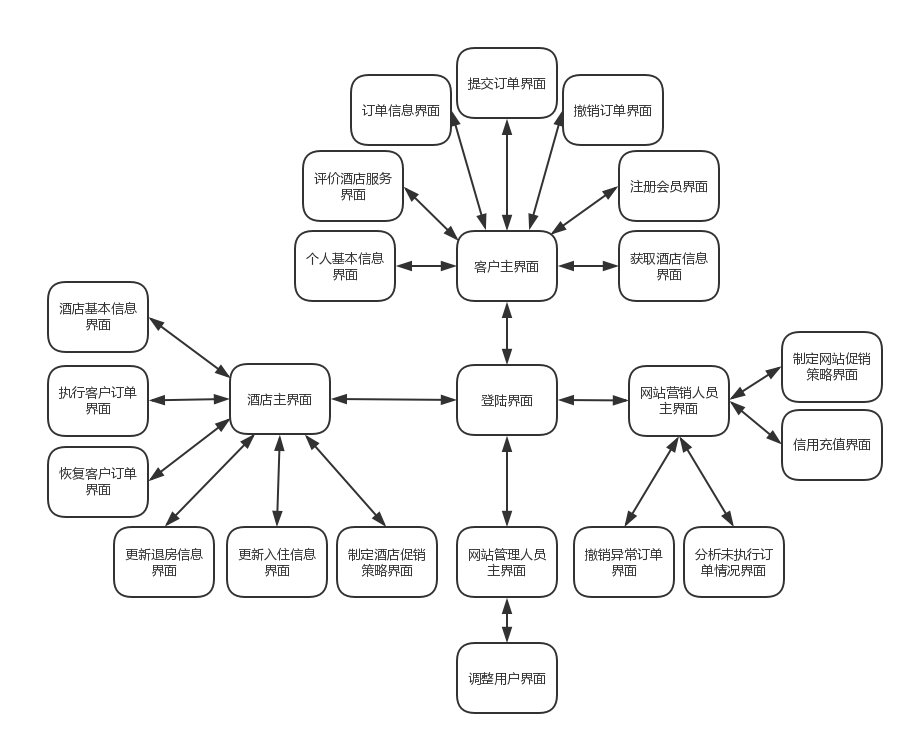


图10 用户界面跳转

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。用户界面类如图所示。

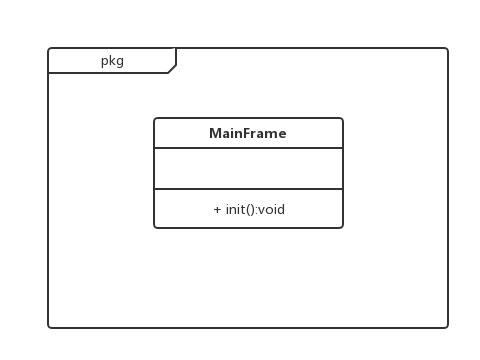


图11 用户界面类

* + 1. **用户界面层模块的职责**

如表所示为用户界面层模块的职责

表5 用户界面层模块的职责

|  |  |
| --- | --- |
| **模块** | **职责** |
| **MainFrame** | 界面Frame，负责界面的显示和界面的跳转 |

* + 1. **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所示。

表6 用户界面层模块的接口规范

|  |  |  |
| --- | --- | --- |
| **MainFrame** | **语法** | **Init(args:String[])** |
| 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所示。

表7 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 |

* + 1. **用户界面模块设计原理**

用户界面利用JavaFX的API来实现

* 1. 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，Order对象负责处理订单界面的业务逻辑。业务逻辑层的设计如图所示。

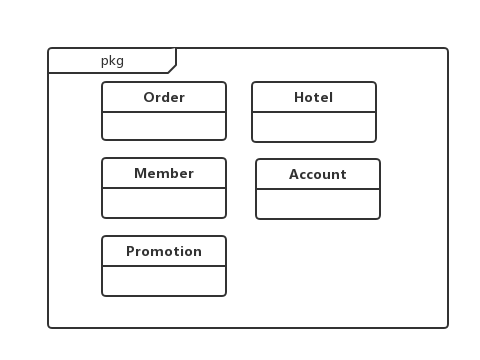


图12 业务逻辑层的设计

* + 1. **业务逻辑层模块的职责**

业务逻辑层模块职责如表所示

表8 业务逻辑层模块的职责

|  |  |
| --- | --- |
| **模块** | **职责** |
| **orderbl** | 负责处理和订单相关的服务 |
| **hotelbl** | 负责处理和酒店退订房相关的服务 |
| **memberbl** | 负责处理和用户信息相关的服务 |
| **accountbl** | 负责处理和用户账号相关的服务 |
| **hotelpromotionbl** | 负责处理和酒店促销策略相关的服务 |
| **marketpromotionbl** | 负责处理和网站促销策略相关的服务 |
| **creditbl** | 负责处理和信用记录相关的服务 |

* + 1. **业务逻辑层模块的接口规范**

orderbl、hotelbl、memberbl、accountbl和hotelpromotionbl、**marketpromotionbl**模块的接口规范如下表所示

表9 **Order**的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| **Order.getOrder** | 语法 | Public OrderVo getOrderVo(String orderId) | |
| 前置条件 | 客户获取订单 | |
| 后置条件 | 无 | |
| **Order.getOrderId** | 语法 | Public String getOrderId(String orderId) | |
| 前置条件 | 客户获取订单编号 | |
| 后置条件 | 无 | |
| **Order.getOrderCondition** | 语法 | Public OrderCondition getOrderCondition(String orderId) | |
| 前置条件 | 客户获取订单状态 | |
| 后置条件 | 无 | |
| **Order.getMemberId** | 语法 | public String getMemberId(String orderId); | |
| 前置条件 | 获取订单上的用户ID | |
| 后置条件 | 无 | |
| **Order.getMemberName** | 语法 | public String getMemberName(String orderId); | |
| 前置条件 | 获取订单上的用户名字 | |
| 后置条件 | 无 | |
| **Order.getCreateTime** | 语法 | public Date getCreateTime(String orderId); | |
| 前置条件 | 获取订单的生成时间 | |
| 后置条件 | 无 | |
| **Order.getCheckInTime** | 语法 | public Date getCheckInTime(String orderId); | |
| 前置条件 | 获取订单的执行时间 | |
| 后置条件 | 无 | |
| **Order.getCheckOutTime** | 语法 | public Date getCheckOutTime(String orderId); | |
| 前置条件 | 获取订单的结束时间 | |
| 后置条件 | 无 | |
| **Order.getHotelName** | 语法 | public String getHotelName(String orderId); | |
| 前置条件 | 获取订单的酒店名称 | |
| 后置条件 | 无 | |
| **Order.getHotelId** | 语法 | public String getHotelId(String orderId); | |
| 前置条件 | 获取订单的酒店ID | |
| 后置条件 | 无 | |
| **Order.getRoomType** | 语法 | public RoomType getRoomType(String orderId); | |
| 前置条件 | 获取订单的房间类型 | |
| 后置条件 | 无 | |
| **Order.getNumOfRoom** | 语法 | public int getNumOfRoom(String orderId); | |
| 前置条件 | 获取订单的房间数量 | |
| 后置条件 | 无 | |
| **Order.getNumOfGuest** | 语法 | public int getNumOfGuest(String orderId); | |
| 前置条件 | 获取订单的客人数量 | |
| 后置条件 | 无 | |
| **Order.getChildExist** | 语法 | public boolean getChildExist(String orderId); | |
| 前置条件 | 获取是否有儿童 | |
| 后置条件 | 无 | |
| **Order.getOriginalPrice** | 语法 | public int getOriginalPrice(String orderId); | |
| 前置条件 | 获取订单房间的原价 | |
| 后置条件 | 无 | |
| **Order.getDiscount** | 语法 | public double getDiscount(String orderId); | |
| 前置条件 | 获取促销策略 | |
| 后置条件 | 无 | |
| **Order.getDiscountedPrice** | 语法 | public int getDiscountedPrice(String orderId); | |
| 前置条件 | 获取折后价格 | |
| 后置条件 | 无 | |
| **Order.getAbnormalOrders** | 语法 | public List<OrderVo> getAbnormalOrders() | |
| 前置条件 | 网站营销人员获取异常订单列表 | |
| 后置条件 | 获取异常订单列表 | |
| **Order.getAllOrdersByHotel** | 语法 | public List<OrderVo> getAllOrdersByHotel  (String hotelId) | |
| 前置条件 | 酒店工作人员获取所有订单 | |
| 后置条件 | 得到酒店的所有订单列表 | |
| **Order.getAllOrdersByMember** | 语法 | public List<OrderVo> getAllOrdersByMember  (String memberId) | |
| 前置条件 | 客户获取所有订单 | |
| 后置条件 | 得到客户的所有订单列表 | |
| **Order.getNotExecutedOrdersByWebsite** | 语法 | public List<OrderVo> getNotExecutedByWebsite() | |
| 前置条件 | 网站营销人员获取未执行订单列表 | |
| 后置条件 | 得到所有未执行订单列表 | |
| **Order.getOrdersInConditionByHotel** | 语法 | public List<OrderVo> get**Orders**InConditionByHotel  (String hotelId, OrderCondition condition) | |
| 前置条件 | 酒店工作人员按订单类型获取订单列表 | |
| 后置条件 | 根据订单类型获取酒店的订单列表 | |
| **Order.getOrdersInConditionByMember** | 语法 | public List<OrderVo> get**Orders**InConditionByMember  (String memberId, OrderCondition condition) | |
| 前置条件 | 客户按订单类型获取订单列表 | |
| 后置条件 | 根据订单类型获取客户的订单列表 | |
| **Order.setToAbnormal** | 语法 | public ResultMessage setToAbnormal() | |
| 前置条件 | 客户在超过最晚订单执行时间未办理入住 | |
| 后置条件 | 将订单状态置为异常，更新客户信用值 | |
| **Order.submit** | 语法 | public ResultMessage submit  (String memberId, String planCheckInTime, String hotelId, int numOfDays, RoomType roomType, int numOfRoom, int numOfGuest, Boolean childExist) | |
| 前置条件 | 客户提交订单 | |
| 后置条件 | 更新订单、客户、酒店房间信息 | |
| **Order.cancel** | 语法 | public ResultMessage cancel(String orderId) | |
| 前置条件 | 客户取消订单 | |
| 后置条件 | 修改订单类型信息 | |
| **Order.checkIn** | 语法 | public ResultMessage checkIn(String orderId) | |
| 前置条件 | 酒店工作人员执行订单 | |
| 后置条件 | 更新订单状态为正在执行，客户信用值 | |
| **Order.abnormalCheckIn** | 语法 | public ResultMessage abnormalCheckIn  (String orderId) | |
| 前置条件 | 订单状态为异常 | |
| 后置条件 | 将订单状态置为正在执行，更新客户信用值 | |
| **Order.checkOut** | 语法 | public ResultMessage checkout(String orderId) | |
| 前置条件 | 酒店工作人员为订单执行退房操作 | |
| 后置条件 | 更新订单状态为已执行 | |
| **Order.revoke** | 语法 | public ResultMessage revoke  (String orderId, double allOrHalf) | |
| 前置条件 | 网站营销人员撤销异常订单 | |
| 后置条件 | 更新订单状态为已撤销，更新客户信用值 | |
| **Order.delete** | 语法 | public ResultMessage delete(String orderId) | |
| 前置条件 | 存在该id的订单 | |
| 后置条件 | 删除订单 | |
| **Order.setToFinished** | 语法 | public ResultMessage setToFinished(String orderId) | |
| 前置条件 | 客户已执行评分操作 | |
| 后置条件 | 更新订单状态为已完成 | |
| **需要的服务（需接口）** | | | |
| 服务名 | | | 服务 |
| **DatabaseFactory.getOrderData()** | | | 得到Order数据库的服务的引用 |
| **Member.chargeCredit(String id, int amount)** | | | 修改客户信用值 |
| **Member.getInfo(String memberId)** | | | 得到客户信息 |
| **Promotion.getPromotion()** | | | 得到促销策略 |
| **Hotel.bookRoom(RoomType type, int number)** | | | 预定房间，更新订单、酒店房间信息 |
| **Hotel.checkoutRoom**  **(RoomType type, int number)** | | | 退房，更新订单、酒店房间信息 |
| **OrderDataService.inset(OrderPo po)** | | | 在数据库中增加一条po记录 |
| **OrderDataService.find(String orderId)** | | | 按ID查找返回相应的OrderPo结果 |
| **OrderDataService.update(OrderPo po)** | | | 更新一个po |
| **OrderDataService.delete(String orderId)** | | | 删除该po |
| **OrderDataService.getAllByHotel(String hotelId)** | | | 得到该酒店的所有订单 |
| **OrderDataService.getAllByMember**  **(String memberId)** | | | 得到该客户的所有订单 |
| **OrderDataService.getNotExecuted()** | | | 得到所有未执行订单 |
| **OrderDataService.getAbnormal()** | | | 得到所有异常订单 |

表10 **Member** 的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| **Member.getInfo** | 语法 | | public MemberVo getInfo(String id) |
| 前置条件 | | Id合法 |
| 后置条件 | | 返回用户vo |
| **Member.checkInfo** | 语法 | | public ResultMessage checkInfo  (String info, InfoType type) |
| 前置条件 | | 无 |
| 后置条件 | | 返回需检查信息的检查结果 |
| **Member. modifyInfo** | 语法 | | public ResultMessage modifyInfo  (MemberVo vo) |
| 前置条件 | | vo合法 |
| 后置条件 | | 返回修改成功/失败信息，持久化更新所涉及的领域对象的数据 |
| **Member.chargeCredit** | 语法 | | public ResultMessage chargeCredit  (String id, int amount) |
| 前置条件 | | id合法 |
| 后置条件 | | 对客户信用进行充值 |
| **Member.getCredit** | 语法 | | Public int getCredit(String id) |
| 前置条件 | | Id合法 |
| 后置条件 | | 得到该账号的信用值 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| **DatabaseFactory.getMemberData()** | | 得到Member数据层的服务的引用 | |
| **MemberDataService.getCredit**  **(String memberId)** | | 根据账号得到该账号的信用值 | |
| **MemberDataService.chargeCredit**  **(String memberId, int amount)** | | 对客户信用进行充值,更新一条po | |
| **MemberDataService.getInfo**  **(String memberId)** | | 根据账号在数据库查找po | |
| **MemberDataService.modifyInfo**  **(MemberPo po)** | | 修改账号信息，更新一条po | |

表11 **Account** 的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| **Account.login** | 语法 | | public AccountType login  (String id, String password) |
| 前置条件 | | 账号密码均正确且账号未登录 |
| 后置条件 | | 将账号状态置为已登录 |
| **Account.logout** | 语法 | | public ResultMessage logout(String id) |
| 前置条件 | | 账号状态为已登录 |
| 后置条件 | | 将账号状态置为未登录 |
| **Account.register** | 语法 | | public String register(String name,  String password, boolean isMember) |
| 前置条件 | | 用户名、密码合法 |
| 后置条件 | | 生成用户账号 |
| **Account.modifyPassword** | 语法 | | public ResultMessage modifyPassword  (String id, String newPassword) |
| 前置条件 | | 该账号存在且新密码合法 |
| 后置条件 | | 修改该账号的密码 |
| **Account.checkInput** | 语法 | | public ResultMessage checkInput(String input) |
| 前置条件 | | 无 |
| 后置条件 | | 检查输入是否合法 |
| **Account.findAccount** | 语法 | | public AccountVo findAccount(String id) |
| 前置条件 | | Id合法 |
| 后置条件 | | 根据id查找账号 |
| **Account.insertAccount** | 语法 | | public String insertAccount(String name,  String password, AccountType type) |
| 前置条件 | | 网站管理人员增加任意类型的用户 |
| 后置条件 | | 生成该用户的账号 |
| **Account.updateAccount** | 语法 | | public ResultMessage updateAccount  (AccountVo vo) |
| 前置条件 | | 用户更新账号信息 |
| 后置条件 | | 更新账号信息 |
| **Account.deleteAccount** | 语法 | | public ResultMessage deleteAccount(String id) |
| 前置条件 | | 网站管理人员删除账号 |
| 后置条件 | | 删除账号 |
| **Account.getAccountTypeById** | 语法 | | public AccountType getAccountTypeById  (String id) |
| 前置条件 | | 无 |
| 后置条件 | | 根据账号得到用户类型 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| **DatabaseFactory.getAccountData()** | | 得到Account数据层的服务的引用 | |
| **AccountDateService.insert(String username, String password, AccountType type)** | | 在数据库中插入一个单一持久化对象 | |
| **AccountDateService.delete(String id)** | | 在数据库中删除单一持久化对象 | |
| **AccountDateService.update(AccountPo po)** | | 在数据库中更新单一持久化对象 | |
| **AccountDateService.find(String id)** | | 根据账号查找单一持久化对象 | |
| **AccountDateService.modifyPassword(String id, String newPassword)** | | 更新账号的密码 | |
| **AccountDateService.login**  **(String id, String password)** | | 将账号置为登录状态 | |
| **AccountDateService.logout(String id)** | | 账将号置为未登录状态 | |

表12 **Hotel**的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| **Hotel.getHotel** | 语法 | public HotelVo getHotel(String hotelId) | |
| 前置条件 | 无 | |
| 后置条件 | 按照酒店编号得到酒店vo | |
| **Hotel.updateBasicInfo** | 语法 | public ResultMessage updateBasicInfo(HotelVo vo) | |
| 前置条件 | 创建一个hotel对象 | |
| 后置条件 | 更新酒店基本信息 | |
| **Hotel.addHotel** | 语法 | public ResultMessage addHotel(String hotelId) | |
| 前置条件 | 不存在该ID的酒店 | |
| 后置条件 | 增加一个酒店 | |
| **Hotel.deleteHotel** | 语法 | public ResultMessage deleteHotel(String hotelId) | |
| 前置条件 | 存在该酒店 | |
| 后置条件 | 删除一个酒店 | |
| **Hotel.initializeRoom** | 语法 | public ResultMessage initializeRoom(String hotelId, RoomType type, int number, int price) | |
| 前置条件 | 输入合法 | |
| 后置条件 | 初始化客房信息 | |
| **Hotel.bookRoom** | 语法 | public ResultMessage bookroom(RoomType type, int number) | |
| 前置条件 | 该类型房间有空余数量 | |
| 后置条件 | 更新客房数量信息 | |
| **Hotel.checkoutRoom** | 语法 | public ResultMessage checkoutRoom(RoomType type, int number) | |
| 前置条件 | 创建一个hotel对象 | |
| 后置条件 | 更新客房数量信息 | |
| **Hotel.getReadyRoom** | 语法 | public int getReadyRoom(RoomType type) | |
| 前置条件 | 创建一个hotel对象 | |
| 后置条件 | 得到该类型房间的空余数量 | |
| **Hotel.comment** | 语法 | public ResultMessage comment(String comment) | |
| 前置条件 | 客户给予文字评价 | |
| 后置条件 | 更新酒店文字评论信息 | |
| **Hotel.givePoStrings** | 语法 | public ResultMessage givePoStrings(String orderId, double poString) | |
| 前置条件 | 客户给予评分 | |
| 后置条件 | 更新酒店评分、评分人数信息，更新订单状态 | |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| **DatabaseFactory.getHotelData** | | | 得到Hotel数据层的服务的引用 |
| **Order.setToFinished(String orderId)** | | | 将订单状态置为已评价状态 |
| **HotelDataService.addHotelPo(HotelPo po)** | | | 添加一个酒店po |
| **HotelDataService.deleteHotelPo(String hotelId)** | | | 删除一个酒店po |
| **HotelDataService.updateHotelList(HotelPo po)** | | | 更新一个酒店po |
| **HotelDataService.find(String id)** | | | 按ID进行查找返回相应的HotelPo结果 |
| **HotelDataService.updateComment(HotelPo po)** | | | 更新酒店文字评论 |
| **HotelDataService.initHotelTypeRoom(String hotelId, RoomType type, int number, int price)** | | | 初始化酒店房间信息 |
| **HotelDataService.getReadyRoom(String hotelId, RoomType type)** | | | 得到某类型房间的空余数量 |
| **HotelDataService.updateOrderedRoom(String hotelId, RoomType type, int number,**  **boolean ischeckI0n)** | | | 更新酒店房间预订数量 |

表13 **HotelPromotion**的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| **HotelPromotion.addActivityStrategy** | 语法 | public ResultMessage addActivityStrategy  (String hotelId, ActivityPromotionVo vo) | |
| 前置条件 | 酒店工作人员增加活动促销策略 | |
| 后置条件 | 增加一条活动促销策略 | |
| **HotelPromotion.upActivityStrategy** | 语法 | public ResultMessage upActivityStrategy  (String hotelId, ActivityPromotionVo vo) | |
| 前置条件 | 酒店工作人员更新活动促销策略 | |
| 后置条件 | 更新一条活动促销策略 | |
| **HotelPromotion.delActivityStrategy** | 语法 | public ResultMessage delActivityStrategy  (String hotelId, ActivityPromotionVo vo) | |
| 前置条件 | 酒店工作人员删除一条活动策略 | |
| 后置条件 | 删除一条活动促销策略 | |
| **HotelPromotion.getCurrentActStrategy** | 语法 | public List<ActivityPromotionVo> getCurrentActStrategy(String hotelId) | |
| 前置条件 | 无 | |
| 后置条件 | 得到当前的有效活动促销策略列表 | |
| **HotelPromotion.upCoopStrategy** | 语法 | public ResultMessage upCoopStrategy  (String hotelId, CompanyProVo vo) | |
| 前置条件 | 酒店工作人员更新合作企业促销策略 | |
| 后置条件 | 更新企业促销策略 | |
| **HotelPromotion.getCoopStrategy** | 语法 | public double getCoopStrategy  (String hotelId, String memberId) | |
| 前置条件 | 无 | |
| 后置条件 | 得到企业在酒店的促销折扣 | |
| **HotelPromotion.upOrderRoomStrategy** | 语法 | public ResultMessage upOrderRoomStrategy  (String hotelId, RoomPromotionVo vo) | |
| 前置条件 | 酒店工作人员更新房间数量预订促销策略 | |
| 后置条件 | 更新房间预订促销策略 | |
| **HotelPromotion.getOrderRoomStrategy** | 语法 | public RoomPromotionVo getOrderRoomStrategy  (String hotelId, int numOfRoom) | |
| 前置条件 | 无 | |
| 后置条件 | 得到酒店房间数量预定促销策略 | |
| **HotelPromotion.upBirthStrategy** | 语法 | public ResultMessage upBirthStrategy  (String hotelId, BirthdayProVo vo) | |
| 前置条件 | 酒店工作人员更新客户生日促销策略 | |
| 后置条件 | 更新客户生日促销策略 | |
| **HotelPromotion.getBirthStrategy** | 语法 | public double getBirthStrategy  (String hotelId, Date birthday) | |
| 前置条件 | 无 | |
| 后置条件 | 得到客户生日当日酒店促销折扣 | |
| **需要的服务（需接口）** | | | |
| 服务名 | | | 服务 |
| **DatabaseFactory.getHotelPromotionData()** | | | 得到HotelPromotion数据层的服务的引用 |
| **HotelPromotionDataService.upActPromotion(String hotelId, ActivityPromotionPo po)** | | | 更新活动策略 |
| **HotelPromotionDataService.delActPromotion(String hotelId, ActivityPromotionPo po)** | | | 删除活动策略 |
| **HotelPromotionDataService.getActProList(String hotelId)** | | | 得到活动促销策略列表 |
| **HotelPromotionDataService.upBirthPromotion(String hotelId, BirthdayProPo po)** | | | 更新客户生日优惠策略 |
| **HotelPromotionDataService.getBirthPromotion(String hotelId)** | | | 得到客户生日优惠策略 |
| **HotelPromotionDataService.updateCooperPro(String hotelId, CompanyProPo po)** | | | 更新合作企业优惠策略 |
| **HotelPromotionDataService.getCooperPro(String hotelId)** | | | 得到合作企业优惠策略 |
| **HotelPromotionDataService.upRoomPromotion(String hotelId, RoomPromotionPo po)** | | | 更新预订房间数量优惠策略 |
| **HotelPromotionDataService.getRoomPromotion(String hotelId)** | | | 得到预订房间数量优惠策略 |

表14 **MarketPromotion**的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| **MarketPromotion.addActivityStrategy** | 语法 | | public ResultMessage addActivityStrategy  (ActivityPromotionVo vo) |
| 前置条件 | | 网站营销人员增加活动促销策略 |
| 后置条件 | | 增加一条活动促销策略 |
| **MarketPromotion.upActivityStrategy** | 语法 | | public ResultMessage upActivityStrategy  (ActivityPromotionVo vo) |
| 前置条件 | | 网站营销人员更新活动促销策略 |
| 后置条件 | | 更新一条活动促销策略 |
| **MarketPromotion.delActivityStrategy** | 语法 | | public ResultMessage delActivityStrategy  (ActivityPromotionVo vo) |
| 前置条件 | | 网站营销人员删除一条活动策略 |
| 后置条件 | | 删除一条活动促销策略 |
| **MarketPromotion.getCurrentActStrategy** | 语法 | | public List<ActivityPromotionVo> getCurrentActStrategy() |
| 前置条件 | | 无 |
| 后置条件 | | 得到当前的有效活动促销策略列表 |
| **MarketPromotion.addBusinessStrategy** | 语法 | | public ResultMessage addBusinessStrategy  (BusinessProVo vo) |
| 前置条件 | | 网站营销人员增加一条商圈策略 |
| 后置条件 | | 增加一条商圈策略 |
| **MarketPromotion.upBusinessStrategy** | 语法 | | public ResultMessage upBusinessStrategy  (BusinessProVo vo) |
| 前置条件 | | 网站营销人员更新一条商圈策略 |
| 后置条件 | | 更新一条商圈策略 |
| **MarketPromotion.delBusinessStrategy** | 语法 | | public ResultMessage delBusinessStrategy  (BusinessProVo vo) |
| 前置条件 | | 网站营销人员删除一个商圈策略 |
| 后置条件 | | 删除一个商圈促销策略 |
| **MarketPromotion.getBusinessStrategy** | 语法 | | public BusinessProVo getBusinessStrategy  (String businessName) |
| 前置条件 | | 无 |
| 后置条件 | | 得到当前商圈专属促销策略 |
| **MarketPromotion.upLevelStrategy** | 语法 | | public ResultMessage upLevelStrategy  (List<LevelVo> list) |
| 前置条件 | | 网站营销人员更新等级促销策略 |
| 后置条件 | | 更新等级促销策略列表 |
| **MarketPromotion.getLevelStrategy** | 语法 | | public double getLevelStrategy(int level) |
| 前置条件 | | 无 |
| 后置条件 | | 得到客户等级对应的折扣 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| **DatabaseFactory.getMarketPromotionData()** | | 得到MarketPromotion数据层的服务的引用 | |
| **MarketPromotionDataService.upActPromotion**  **(ActivityPromotionPo po)** | | 更新活动策略 | |
| **MarketPromotionDataService.delActPromotion**  **(ActivityPromotionPo po)** | | 删除活动策略 | |
| **MarketPromotionDataService.getActProList()** | | 得到活动促销策略列表 | |
| **MarketPromotionDataService.updateLevelRule** | | 更新等级促销策略 | |
| **MarketPromotionDataServicegetLevelList()** | | 得到客户等级促销策略 | |
| **MarketPromotionDataService.updateBusiness**  **(BusinessProPo po)** | | 更新特定商圈促销策略 | |
| **MarketPromotionDataService.deleteBusiness**  **(BusinessProPo po)** | | 删除一条商圈策略 | |
| **MarketPromotionDataService.getBusinessList()** | | 得到特定商圈促销列表 | |
| **MemberDataService.modifyInfo(Member vo)** | | 当等级策略变化，更新所有客户等级 | |

表15 Credit的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| **Credit.addCreditByOrder** | 语法 | | public ResultMessage addCreditByOrder  (String id,int amount,OrderVo vo) |
| 前置条件 | | Id,订单,信用变化合法 |
| 后置条件 | | 更新信用值和信用值记录 |
| **Credit.getCreditRecordList** | 语法 | | public List<CreditRecordVo> get CreditRecordList(String id) |
| 前置条件 | | id合法 |
| 后置条件 | | 得到用户的信用值记录表 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| **DatabaseFactory.getCreditData()** | | 得到Credit数据层的服务的引用 | |
| **CreditDataService.updateCreditRecordData**  **(List<CreditRecordPo> list,String id)** | | 更新用户的信用记录 | |

* 1. 数据层的分解

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的增、删、改、查。数据层模块的描述具体如下图所示。

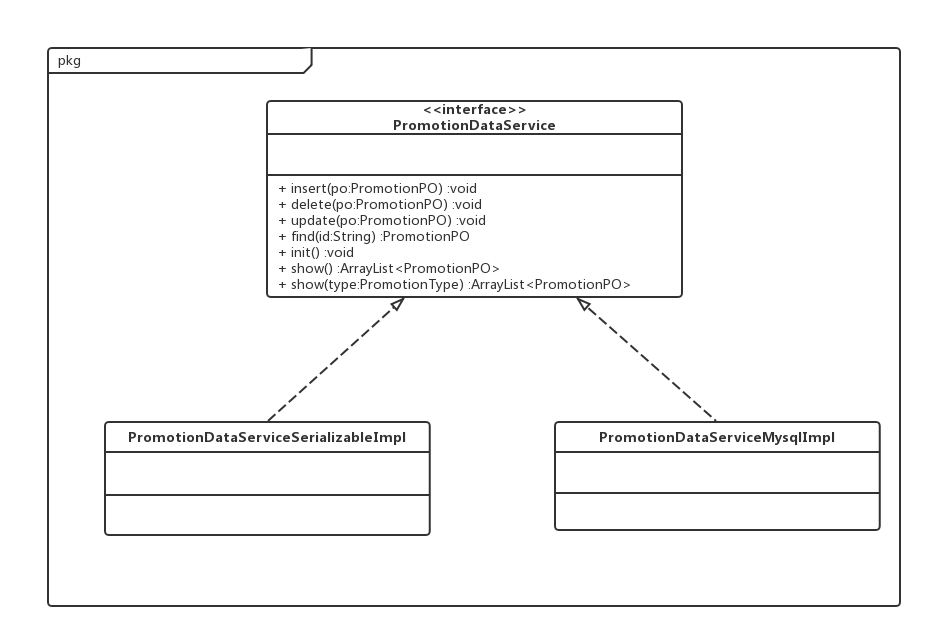


图13 数据层模块的描述

**5.4.1．数据层模块的职责**

数据层模块的职责如下表所示：

表14 数据层模块职责

|  |  |
| --- | --- |
| **模块** | **职责** |
| **OrderDataService** | 提供订单数据的集体载入、保存、增加、查找、修改服务 |
| **HotelDataService** | 提供酒店数据的集体载入、保存、增加、查找、修改、删除服务 |
| **MemberDataService** | 提供用户信息数据集体载入、保存、增加、查找、修改、删除服务 |
| **AccountDataService** | 提供账号数据查找服务 |
| **HotelPromotionDataService** | 提供酒店促销策略数据的集体载入、保存、增加、查找、修改、删除服务 |
| **MarketPromotionDataService** | 提供网站促销策略数据的集体载入、保存、增加、查找、修改、删除服务 |
| **CreditDataService** | 提供信用记录的集体载入、查找服务 |

**5.4.2．数据层模块的接口规范**

数据层模块的接口规范如下表

表15 **OrderDataService接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **OrderDataService.insert** | **语法** | public void insert (OrderPo po) |
| **前置条件** | 相同ID的po在数据库中不存在 |
| **后置条件** | 在数据库中增加一条po记录 |
| **OrderDataService.find** | **语法** | public OrderPo find(String orderId) |
| **前置条件** | 订单号合法 |
| **后置条件** | 按ID查找返回相应的OrderPo结果 |
| **OrderDataService.update** | **语法** | Public void update(OrderPo po) |
| **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 |
| **OrderDataService.delete** | **语法** | public ResultMessage deleteOrder  (String orderId) |
| **前置条件** | 数据库中存在该po |
| **后置条件** | 删除该po |
| **OrderDataService.getAllByHotel** | **语法** | public List<OrderPo> getAllByHotel  (String hotelId) |
| **前置条件** | 酒店账号存在 |
| **后置条件** | 得到该酒店的所有订单 |
| **OrderDataService.getAllByMember** | **语法** | public List<OrderPo> getAllByMember  (String memberId) |
| **前置条件** | 客户账号存在 |
| **后置条件** | 得到该客户的所有订单 |
| **OrderDataService.getNotExecuted** | **语法** | public List<OrderPo> getNotExecuted() |
| **前置条件** | 无 |
| **后置条件** | 得到所有未执行订单 |
| **OrderDataService.getAbnormal** | **语法** | public List<OrderPo> getAbnormal() |
| **前置条件** | 无 |
| **后置条件** | 得到所有异常订单 |

表16 **HotelDataService接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **HotelDataService.insert** | **语法** | public ResultMessage insert(HotelPo po) |
| **前置条件** | 同样ID的po在数据库中不存在 |
| **后置条件** | 在数据库中增加一条po |
| **HotelDataService.updateHotelList** | **语法** | public ResultMessage updateHotelList(HotelPo po) |
| **前置条件** | 在数据库中存在同样ID的po |
| **后置条件** | 更新酒店列表信息 |
| **HotelDataService.delete** | **语法** | public ResultMessage delete(String hotelId) |
| **前置条件** | 在数据库中存在同样ID的po |
| **后置条件** | 删除一个po |
| **HotelDataService.initHotelTypeRoom** | **语法** | public ResultMessage initHotelTypeRoom(String hotelId, RoomType type, int number, int price) |
| **前置条件** | 酒店工作人员录入客房信息 |
| **后置条件** | 初始化房间信息 |
| **HotelDataService.findHotel** | **语法** | public HotelPo findHotel(String hotelId) |
| **前置条件** | 客户查看酒店信息 |
| **后置条件** | 按ID返回相应的HotelPo结果 |
| **HotelDataService.keyFind** | **语法** | public List<HotelPo> keyFind(String info) |
| **前置条件** | 客户输入关键字搜索酒店 |
| **后置条件** | 按关键字进行查找并返回所有符合条件的HotelPo结果 |
| **HotelDataService.getReadyRoom** | **语法** | public int getReadyRoom(String hotelId, RoomType type) |
| **前置条件** | 酒店工作人员查看酒店空余房间数 |
| **后置条件** | 返回酒店某类型房间空余数 |
| **HotelDataService.updateComment** | **语法** | public ResultMessage updateComment(HotelPo po) |
| **前置条件** | 得到客户的文字评价 |
| **后置条件** | 更新酒店文字评论 |
| **HotelDataService.updateOrderedRoom** | **语法** | public ResultMessage updateOrderedRoom(String hotelId, RoomType type, int number,  Boolean isCheckIn) |
| **前置条件** | 客户预定酒店或退房 |
| **后置条件** | 更新酒店房间信息 |
| **HotelDataService.getProvinceList** | **语法** | public List<String> getProvinceList() |
| **前置条件** | 客户通过地址筛选搜索酒店 |
| **后置条件** | 得到所有省份列表 |
| **HotelDataService.getCityList** | **语法** | Public List<String> getCityList  (String province) |
| **前置条件** | 客户已选择一个省份 |
| **后置条件** | 得到该省份所有的城市列表 |
| **HotelDataService.getBusinessList** | **语法** | Public List<String> getBusinessList  (String province, String city) |
| **前置条件** | 客户已选择一个城市 |
| **后置条件** | 得到该城市所有的商圈 |

表17 **MemberDataService接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **MemberDataService.getCredit** | **语法** | public int getCredit(String memberId) |
| **前置条件** | 该客户账号存在 |
| **后置条件** | 得到该客户的信用值 |
| **MemberDataService.chargeCredit** | **语法** | public ResultMessage chargeCredit  (String memberId, int amount) |
| **前置条件** | 该客户账号存在，修改的信用值合法 |
| **后置条件** | 修改客户信用值 |
| **MemberDataService.getInfo** | **语法** | public MemberVo getInfo  (String memberId) |
| **前置条件** | 该客户账号存在 |
| **后置条件** | 得到memberVo |
| **MemberDataService.modifyInfo** | **语法** | public ResultMessage modifyInfo  (MemberPo po) |
| **前置条件** | 数据库中存在该po |
| **后置条件** | 更新一条po |

表18 AccountDataService接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **AccountDataService.insert** | **语法** | public String insert(String username, String password, AccountType type) |
| **前置条件** | 数据库中不存在该用户名，密码合法 |
| **后置条件** | 增加一条po，生成账号 |
| **AccountDataService.update** | **语法** | public ResultMessage update  (AccountPo po) |
| **前置条件** | 数据库中存在该po |
| **后置条件** | 更新一条po |
| **AccountDataService.delete** | **语法** | public ResultMessage delete(String id) |
| **前置条件** | 数据库中存在该po |
| **后置条件** | 删除一条po |
| **AccountDataService.find** | **语法** | public AccountPo find(String id) |
| **前置条件** | 数据库中存在该id的po |
| **后置条件** | 得到一条po |
| **AccountDataService.**  **modifyPassword** | **语法** | public ResultMessage modifyPassword  (String id, String newPassword) |
| **前置条件** | 数据库中存在该id的po，新密码合法 |
| **后置条件** | 更新一条po |
| **AccountDataService.login** | **语法** | public AccountType login  (String id, String password) |
| **前置条件** | 账号和密码正确，账号未在登录状态 |
| **后置条件** | 账号置为登录状态 |
| **AccountDataService.logout** | **语法** | public ResultMessage logout(String id) |
| **前置条件** | 该账号为已登录状态 |
| **后置条件** | 账号置为未登入状态 |

表19 **HotelDataService接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **HotelDataService.insert** | **语法** | public ResultMessage insert(HotelPo po) |
| **前置条件** | 同样ID的po在数据库中不存在 |
| **后置条件** | 在数据库中增加一条po |
| **HotelDataService.updateHotelList** | **语法** | public ResultMessage updateHotelList(HotelPo po) |
| **前置条件** | 在数据库中存在同样ID的po |
| **后置条件** | 更新酒店列表信息 |
| **HotelDataService.delete** | **语法** | public ResultMessage delete(String hotelId) |
| **前置条件** | 在数据库中存在同样ID的po |
| **后置条件** | 删除一个po |
| **HotelDataService.initHotelTypeRoom** | **语法** | public ResultMessage initHotelTypeRoom(String hotelId, RoomType type, int number, int price) |
| **前置条件** | 酒店工作人员录入客房信息 |
| **后置条件** | 初始化房间信息 |
| **HotelDataService.findHotel** | **语法** | public HotelPo findHotel(String hotelId) |
| **前置条件** | 客户查看酒店信息 |
| **后置条件** | 按ID返回相应的HotelPo结果 |
| **HotelDataService.keyFind** | **语法** | public List<HotelPo> keyFind(String info) |
| **前置条件** | 客户输入关键字搜索酒店 |
| **后置条件** | 按关键字进行查找并返回所有符合条件的HotelPo结果 |
| **HotelDataService.getReadyRoom** | **语法** | public int getReadyRoom(String hotelId, RoomType type) |
| **前置条件** | 酒店工作人员查看酒店空余房间数 |
| **后置条件** | 返回酒店某类型房间空余数 |
| **HotelDataService.updateComment** | **语法** | public ResultMessage updateComment(HotelPo po) |
| **前置条件** | 得到客户的文字评价 |
| **后置条件** | 更新酒店文字评论 |
| **HotelDataService.updateOrderedRoom** | **语法** | public ResultMessage updateOrderedRoom(String hotelId, RoomType type, int number,  Boolean isCheckIn) |
| **前置条件** | 客户预定酒店或退房 |
| **后置条件** | 更新酒店房间信息 |
| **HotelDataService.getProvinceList** | **语法** | public List<String> getProvinceList() |
| **前置条件** | 客户通过地址筛选搜索酒店 |
| **后置条件** | 得到所有省份列表 |
| **HotelDataService.getCityList** | **语法** | Public List<String> getCityList  (String province) |
| **前置条件** | 客户已选择一个省份 |
| **后置条件** | 得到该省份所有的城市列表 |
| **HotelDataService.getBusinessList** | **语法** | Public List<String> getBusinessList  (String province, String city) |
| **前置条件** | 客户已选择一个城市 |
| **后置条件** | 得到该城市所有的商圈 |

表20 **MarketDataService接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **MarketPromotionDataService.**  **upActPromotion** | **语法** | public ResultMessage upActPromotion  (ActivityPromotionPo po) |
| **前置条件** | 输入折扣合法，开始日期在结束日期前 |
| **后置条件** | 更新一条po |
| **MarketPromotionDataService.**  **addActPromotion** | **语法** | public ResultMessage addActPromotion  (ActivityPromotionPo po) |
| **前置条件** | 输入折扣合法，开始日期在结束日期前 |
| **后置条件** | 增加一条po |
| **MarketPromotionDataService.**  **delActPromotion** | **语法** | public ResultMessage delActPromotion  (ActivityPromotionPo po) |
| **前置条件** | 数据库中存在该活动促销策略 |
| **后置条件** | 删除一条po |
| **MarketPromotionDataService.**  **getActProList** | **语法** | public List<String> getActProList() |
| **前置条件** | 无 |
| **后置条件** | 得到网站促销活动策略 |
| **MarketPromotionDataService.**  **updateLevelRule** | **语法** | public ResultMessage updateLevelRule  (List<LevelPo> list) |
| **前置条件** | 每个等级所需信用值随着等级递增而增加，折扣合法 |
| **后置条件** | 更新等级促销列表 |
| **MarketPromotionDataService.**  **getLevelList** | **语法** | public List<LevelPo> getLevelList() |
| **前置条件** | 无 |
| **后置条件** | 得到等级促销策略列表 |
| **MarketPromotionDataService.**  **updateBusiness** | **语法** | public ResultMessage updateBusiness  (BusinessProPo po) |
| **前置条件** | 输入折扣合法 |
| **后置条件** | 增加或更新一条po |
| **MarketPromotionDataService.**  **deleteBusiness** | **语法** | public ResultMessage deleteBusiness  (BusinessProPo po) |
| **前置条件** | 数据库中存在该po |
| **后置条件** | 删除一条po |
| **MarketPromotionDataService.**  **getBusinessList** | **语法** | public List<BusinessProPo> getBusinessList() |
| **前置条件** | 无 |
| **后置条件** | 得到有特定促销策略的商圈列表 |

表21 CreditD**ataService接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **CreditDataService.**  **addCreditByOrder** | **语法** | public ResultMessage addCreditByOrder  (String id,int amount,OrderVo vo) |
| **前置条件** | Id,订单号,数额合法 |
| **后置条件** | 增加或更新一条CreditRecordPo |
| **CreditDataService.**  **getCreditRecordList** | **语法** | public List<CreditRecordPo> getCreditRecordList(String id) |
| **前置条件** | id合法且存在 |
| **后置条件** | 得到该id的信用记录列表 |

1. 信息视角
   1. 数据持久化对象

系统的PO类就是对应的相关的实体类，如下所示。

* OrderPO类包括订单的编号（订单号为13位0~9的数字，前8位为当前日期，后5位由系统随机生成，日期格式为：yyyy-mm-dd，因此一天最多可以生成100,000条订单），开始时间，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，儿童数，订单状态（包括已执行、未执行、已撤销、异常），价格（大于等于0、精确到小数点后2位的浮点数），折扣，折扣后总额
* HotelPO类包括酒店名称，地址，所属商圈，联系方式，简介，评分，评价，空余房间数，房间总数
* UserInfoPO类包括客户的编号（格式为一个字母加5位0~9数字，普通客户以“N”开头，企业客户以“E”开头），用户名，密码，分类（普通客户、企业客户），联系方式，信用值，企业名称，生日（日期格式为：yyyy-MM-dd HH:mm:ss）
* AccountPO类包括用户的编号（用户账号统一采用一个字母加5位0~9数字格式，网站管理人员账号以“A”开头，网站营销人员账号以“M”开头，酒店以“H”开头，普通会员账号以“N”开头，企业会员账号以“E”开头），密码
* PromotionPO类是HotelPromotionPO和MarketPromotionPO的父类
* HotelPromotionPO类包括促销类型，开始日期，结束日期（日期格式为：yyyy-MM-dd HH:mm:ss），活动名称，活动折扣；生日折扣；房间数，房间数折扣；企业折扣
* MarketPromotionPO类包括促销类型，开始日期，结束日期（日期格式为：yyyy-MM-dd HH:mm:ss），活动名称，活动折扣，活动简介；商圈名称，商圈等级，商圈折扣；会员等级，每次升级所需的信用值，每个等级对应的折扣
  1. Txt持久化格式

统一用\*.txt格式