目录

[更新历史 2](#_Toc464379329)

[一、 引言 4](#_Toc464379330)

[(一) 编制目的 4](#_Toc464379331)

[(二) 词汇表 4](#_Toc464379332)

[(三) 参考资料 5](#_Toc464379333)

[二、 产品描述 6](#_Toc464379334)

[三、 逻辑视角 7](#_Toc464379335)

[四、 组合视角 9](#_Toc464379336)

[(一) 开发包图 9](#_Toc464379337)

[(二) 运行时进程 14](#_Toc464379338)

[(三) 物理部署 15](#_Toc464379339)

[五、 接口视角 16](#_Toc464379340)

[(一) 模块的职责 16](#_Toc464379341)

[1. 客户端模块视图 16](#_Toc464379342)

[2. 服务器端模块视图 16](#_Toc464379343)

[3. 客户端各层的职责 16](#_Toc464379344)

[4. 服务器端各层的职责 17](#_Toc464379345)

[5. 各层联系 17](#_Toc464379346)

[(二) 用户界面层的分解 18](#_Toc464379347)

[1. 用户界面层模块的职责 19](#_Toc464379348)

[2. 用户界面模块的接口规范 19](#_Toc464379349)

[3. 用户界面模块设计原理 21](#_Toc464379350)

[(三) 业务逻辑层的分解 21](#_Toc464379351)

[1. 业务逻辑层模块的职责 21](#_Toc464379352)

[2. 业务逻辑模块的接口模范 22](#_Toc464379353)

[(四) 数据层的分解 42](#_Toc464379354)

[1. 数据层模块的职责 42](#_Toc464379355)

[2. 数据层模块的接口规范 42](#_Toc464379356)

[六、 信息视角 54](#_Toc464379357)

[(一) 数据持久化对象 54](#_Toc464379358)

[(二) 文件格式 55](#_Toc464379359)

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **修改日期** | **修改原因** | **版本号** |
| 冯俊杰 | 2016/10/4 | 最初草稿 |  |
| 全体成员 | 2016/10/10 | 新增逻辑分包 |  |
| 全体成员 | 2016/10/11 | 新增物理分包 |  |
| 冯俊杰 | 2016/10/12 | 修改物理分包 |  |
| 龚尘淼 | 2016/10/12 | 修改逻辑分包 |  |
| 龚尘淼 | 2016/10/12 | 新增promotion、hotel接口规范 |  |
| 董金玉 | 2016/10/12 | 新增member、user接口规范 |  |
| 冯俊杰 | 2016/10/12 | 新增logIn、credit、market接口规范 |  |
| 高源 | 2016/10/12 | 新增order接口规范 |  |
| 董金玉 | 2016/10/13 | member、user接口规范评审后第一次更新 |  |
| 冯俊杰 | 2016/10/13 | logIn、credit、market接口规范评审后第一次更新 |  |
| 高源 | 2016/10/13 | order接口规范评审后第一次更新 |  |
| 冯俊杰 | 2016/10/13 | credit、market接口规范评审后第二次更新 |  |
| 冯俊杰 | 2016/10/13 | 修改物理分包逻辑层依赖关系 |  |
| 龚尘淼 | 2016/10/14 | 修改逻辑图和用户界面跳转图 |  |
| 龚尘淼 | 2016/10/14 | promotion、hotel接口规范评审后第一次更新 |  |
| 冯俊杰 | 2016/10/14 | 修改逻辑分包 |  |
| 高源 | 2016/10/14 | 新增进程图 |  |
| 龚尘淼 | 2016/10/14 | promotion、hotel接口规范评审后第二次更新 |  |
| 董金玉 | 2016/10/14 | member、user接口规范评审后第二次更新 |  |
| 高源 | 2016/10/15 | order接口规范评审后第二次更新 |  |
| 冯俊杰 | 2016/10/15 | 整合、合并文档 | V0.1 |
| 龚尘淼 | 2016/10/14 | promotion、hotel接口规范评审后第三次更新 | V0.2 |
| 高源 | 2016/10/15 | 新增数据持久化对象PO | V0.3 |
|  |  |  |  |

# 引言

## 编制目的

本报告详细完成对互联网酒店预订系统的概要设计，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。

## 词汇表

|  |  |
| --- | --- |
| **缩写或单词** | **解释** |
| HRS | 互联网酒店预订系统 |
| order | 订单 |
| hotel | 酒店 |
| guest | 客户 |
| hotelWorker | 酒店工作人员 |
| webMarketer | 网站营销人员 |
| webManager | 网站管理人员 |
| signUp | 注册 |
| logIn | 登录 |
| UI | 界面 |
| BL | 逻辑 |
| Data | 数据 |
| promotion | 促销策略 |
| credit | 信用值 |
| member | 会员 |
| market | 营销 |
|  |  |
|  |  |
|  |  |
|  |  |

## 参考资料

1. 互联网酒店预订系统用例文档
2. 互联网酒店预订系统需求规格说明文档
3. 丁二玉，刘钦.计算与软件工程（卷二）[M]机械工业出版2012：134—182
4. IEEE std 1471-2000

# 产品描述

参考《互联网酒店预订系统用例文档》和《互联网酒店预订系统需求规格说明文档》中对产品的概括描述。互联网酒店预订系统主要功能见用例图如下。



# 逻辑视角

互联网酒店预订系统中，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：分层结构**



**图2：逻辑设计方案**

# 组合视角

## 开发包图

[软工2 166]与抽象的逻辑设计相比，实现物理设计要考虑更多的实现细节，这些细节有：

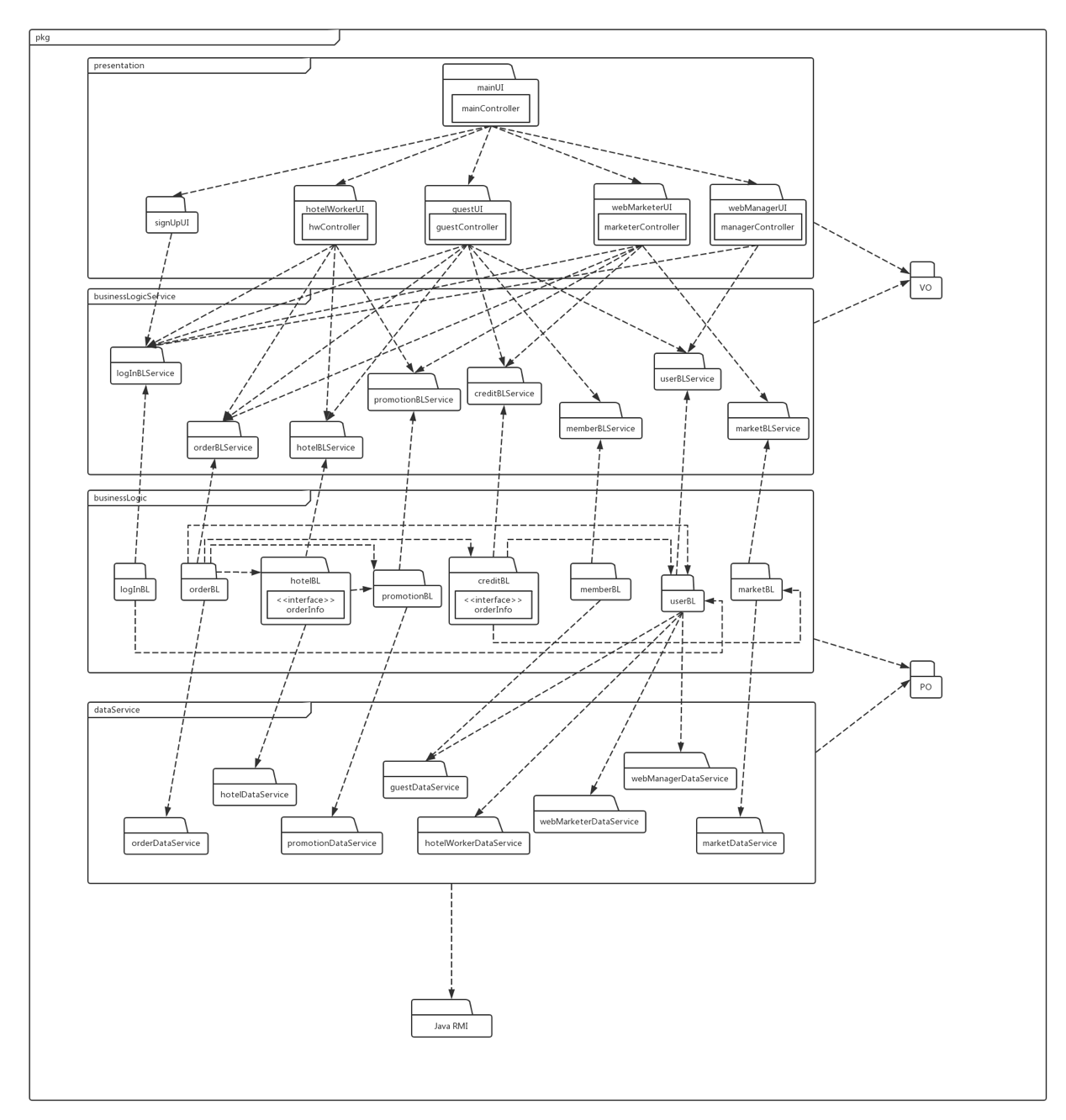
1. 所有的presentation层开发包都需要使用图形类型建立界面，都要依赖于图形界面类库包。
2. 在presentation层实现时，由mainUI包负责不同参与者在整个页面之间的跳转逻辑。其他各包负责各自页面自身的功能，并由各自的controller负责自身内部页面之间的跳转逻辑。
3. presentation层与businessLogic层被置于客户端，data层被置于服务器端，那么businessLogic层的开发包不可能依赖于data层的开发包。使用RMI技术，将data层开发包分解为置于客户端的dataService接口包和置于服务器端的data层开发包。这样一来，businessLogic层开发包依赖于dataService包，dataService包依赖于RMI类库包。
4. 所有的data层开发包都需要进行数据持久化（例如读写数据库、读写文件等），所以它们会有一些重复代码，可以将重复代码独立为新的开发包，然后所有的data层开发包都依赖于dataBaseUtility。dataBaseUtility会依赖于JDBC类库包。
5. 在分层风格的典型设计中，不希望高层直接依赖于低层，而是为低层建立接口包，实现依赖倒置原则，所以应该调整为：各presentation层开发包（调用）依赖于logic层接口包businessLogicService包，businessLogic层开发包也依赖于（实现了）businessLogic层接口包businessLogicService包。
6. 在分层风格的典型设计中，presentation层与logic层之间、logic层与data层之间可能会传递复杂数据对象，那么相邻两层都需要使用数据对象声明，所以需要将数据对象声明独立为开发包——VO包和PO包。其中VO包负责在presentation层与logic层之间传递，PO包负责在logic层与data层之间传递。
7. 使用依赖倒置原则消除包的循环依赖现象，将循环依赖变为单向依赖：
8. orderBL和hotelBL:将部分orderBL类抽象接口orderInfo置入hotelBL包，这样orderBL单向依赖于hotelBL（实现接口+调用）
9. orderBL和creditBL:将部分orderBL类抽象接口orderInfo置入creditBL包，这样orderBL单向依赖于hotelBL（实现接口+调用）

8）在logic层中，初始化和业务逻辑层上下文的工作被分配到utility包中。

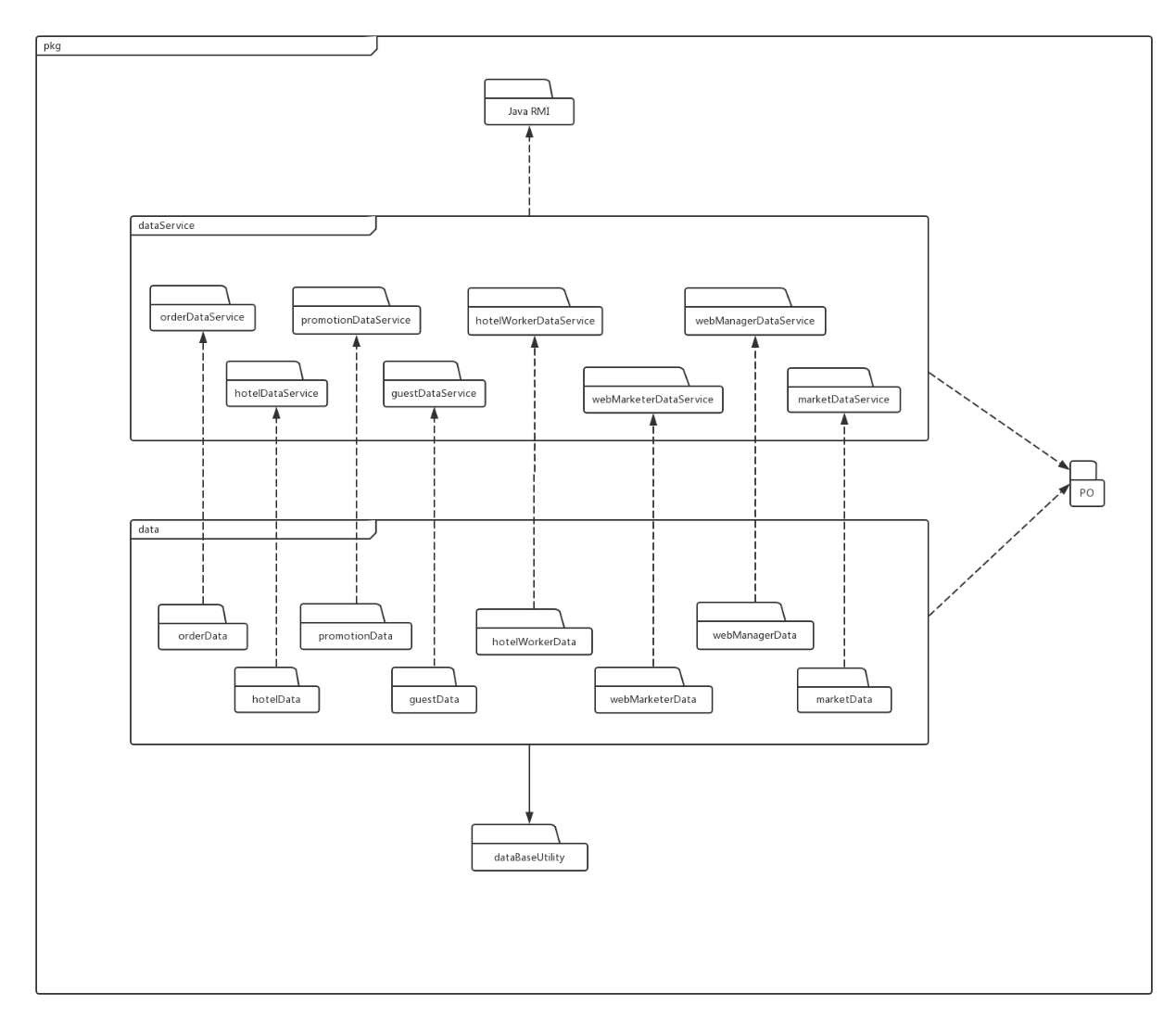
界面层依据参与者分为main、注册界面和4个参与者主界面；逻辑层主要依据逻辑分包的结果，并添加logInBL，负责用户的登录和客户的注册；数据层进行整合合并，依照参与者、订单、酒店的实际情况进行分类。

|  |  |
| --- | --- |
| 开发（物理）包 | 依赖的其他包 |
| mainUI | signUpUI, guestUI, hotelWorkerUI, webMarketerUI, webManagerUI, 界面类库包 |
| signUpUI | logInBLService, 界面类库包, VO |
| guestUI | logInBLService, orderBLService, hotelBLService, creditBLService, memberBLService, UserBLService, 界面类库包, VO |
| hotelWorkerUI | logInBLService, orderBLService, hotelBLService, promotionBLService, 界面类库包,VO |
| webMarketerUI | logInBLService, orderBLService, promotionBLService, creditBLService, marketBLService, 界面类库包,VO |
| webManagerUI | logInBLService, UserBLService, 界面类库包,VO |
| logInBLService | VO |
| orderBLService | VO |
| hotelBLService | VO |
| promotionBLService | VO |
| creditBLService | VO |
| memberBLService | VO |
| marketBLService | VO |
| userBLService | VO |
| logInBL | logInBLService, userBL, PO |
| orderBL | orderBLService, orderDataService, hotelBL, promotionBL, creditBL, userBL, PO |
| hotelBL | hotelBLService, hotelDataService, orderBL, promotionBL, PO |
| promotionBL | promotionBLService, promotionDataService, PO |
| creditBL | creditBLService, orderBL, userBL, marketBL, PO |
| memberBL | memberBLService, guestDataService, PO |
| marketBL | marketBLService, marketDataService, PO |
| userBL | userBLService, guestDataService, hotelWorkerDataService, webMarketerDataService, webManagerDataService, hotelBL, PO |
| orderDataService | Java RMI, PO |
| hotelDataService | Java RMI, PO |
| guestDataService | Java RMI, PO |
| hotelWorkerDataService | Java RMI, PO |
| webMarketerDataService | Java RMI, PO |
| webManagerDataService | Java RMI, PO |
| promotionDataService | Java RMI, PO |
| marketDataService | Java RMI, PO |
| orderData | orderDataService, PO |
| hotelData | hotelDataService, PO |
| guestData | guestDataService, PO |
| hotelWorkerData | hotelWorkerDataService, PO |
| webMarketerData | webMarketerDataService, PO |
| webManagerData | webManagerDataService, PO |
| promotionData | promotionDataService, PO |
| marketData | marketDataService, PO |
| VO |  |
| PO |  |
| dataBaseUtility | java.JDBC |
| 界面类库包 |  |
| JAVA RMI |  |
| java.swing,java.awt,java.2D |  |

下图为互联网酒店预订系统客户端开发包图：



下图为互联网酒店预订系统服务端开发包图：



## 运行时进程

在互联网酒店预订系统中，会有多个客户端进程和一个服务器端进程，其进程图如下图所示。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。数据库的自动更新（如过了订单的最晚入住时间会将订单置为异常）会在服务器端自动进行，作为另一个进程。



## 物理部署

互联网酒店预订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。部署图如下图所示。



# 接口视角

## 模块的职责

### 客户端模块视图



### 服务器端模块视图



### 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 |

### 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 |

### 各层联系

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下图所示。

|  |  |  |
| --- | --- | --- |
| **接口** | **服务调用方** | **服务提供方** |
| logInBLService  orderBLService  hotelBLService  promotionBLService  creditBLService  memberBLService  marketBLService  userBLService | 客户端  展示层  （presentation） | 客户端  业务逻辑层  （businessLogic） |
| orderDataService  hotelDataService  guestDataService  hotelWorkerDataService  webMarketerDataService  webManagerDataService  promotionDataService  marketDataService | 客户端  业务逻辑层  （businessLogic） | 服务器端  数据层  （data） |

## 用户界面层的分解

根据需求，系统存在37个用户界面（加粗为主界面）： **登录界面**、**注册界面**、**客户主界面**、个人基本信息界面、个人基本信息维护界面、会员注册界面、未评价订单界面、订单评价界面、个人订单概况列表界面、个人订单详情界面、个人订单撤销界面、商圈地址选定界面、酒店概况列表界面、订单生成界面、酒店搜索界面、酒店详情界面、会员注册界面、**酒店工作人员界面**、酒店信息界面、酒店信息维护界面、客房信息维护界面、酒店促销策略界面、订单列表界面、订单详情界面、剩余客房信息界面、线下客户入住和退房处理界面、**网站营销人员界面**、用户编号录入界面、信用充值界面、异常订单概况列表界面、异常订单详情界面、异常订单撤销界面、网站促销策略界面、会员等级制定界面、**网站管理人员界面**、会员信息界面、会员信息维护界面、酒店信息添加界面。

界面跳转如下图所示。



### 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 |

### 用户界面模块的接口规范

1. **用户界面层模块的接口规范**

|  |  |  |
| --- | --- | --- |
| mainUI | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示Frame |
| logInUI | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示LoginPanel |
| guestUI | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示guestPanel |
| hotelWorkerUI | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示hotelWorkerPanel |
| webMarketerUI | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示webMarketerPanel |
| webManagerUI | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示webManagerPanel |

1. **用户界面层模块需要的服务接口**

|  |  |
| --- | --- |
| 服务名 | 服务 |
| businessLogicService.logInBLService | 登录界面和客户注册的业务逻辑接口 |
| businessLogicService.orderBLService | 负责处理与订单相关的接口 |
| businessLogicService.hotelBLService | 负责处理与酒店相关的接口 |
| businessLogicService.promotionBLService | 负责处理与促销策略相关的接口 |
| businessLogicService.creditBLService | 负责处理与信用值相关的接口 |
| businessLogicService.memberBLService | 负责处理与会员相关的接口 |
| businessLogicService.userBLService | 负责处理与所有用户信息相关的接口 |
| businessLogicService.marketBLService | 负责处理与会员等级制度信息相关的接口 |

### 用户界面模块设计原理

用户界面拟用java的Swing和AWT库来实现。

## 业务逻辑层的分解

业务逻辑层包括多个针对界面及业务逻辑处理对象。例如logIn对象负责处理所有用户（客户、酒店工作人员、网站营销人员、网站管理人员）的登录和客户的注册的业务逻辑。业务逻辑层的设计如下图所示：



### 业务逻辑层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| logInBL | 负责提供所有用户的登录和客户的注册的服务 |
| orderBL | 负责处理与订单相关的服务 |
| hotelBL | 负责处理与酒店相关的服务 |
| promotionBL | 负责处理与促销策略相关的服务 |
| creditBL | 负责处理与信用值相关的服务 |
| memberBL | 负责处理与会员相关的服务 |
| userBL | 负责处理与所有用户信息相关的服务 |
| marketBL | 负责处理与会员等级制度信息相关的服务 |

### 业务逻辑模块的接口模范

1. **logIn模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| logIn.guestLogIn | 语法 | Public Boolean guestLogIn (String guest, String password) |
| 前置条件 | 客户输入客户编号和密码 |
| 后置条件 | 客户被认证并进入系统客户主界面 |
| logIn.hotelWorkerLogIn | 语法 | Public Boolean hotelWorkerLogIn (String hotelWorker, String password) |
| 前置条件 | 酒店工作人员输入编号和密码 |
| 后置条件 | 酒店工作人员被认证并进入系统酒店工作人员主界面 |
| logIn.webMarketerLogIn | 语法 | Public Boolean webMarketerLogIn (String webMarketer, String password) |
| 前置条件 | 网站营销人员输入编号和密码 |
| 后置条件 | 网站营销人员被认证并进入系统网站营销人员主界面 |
| logIn.webManagerLogIn | 语法 | Public Boolean webManagerLogIn (String webManager, String password) |
| 前置条件 | 网站管理人员输入编号和密码 |
| 后置条件 | 网站管理人员被认证并进入系统网站管理人员主界面 |
| logIn.guestSignUp | 语法 | Public ResultMessage guestSignUp (GuestVO 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 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| User.getSingle(String userID,UserType userType) | 返回与该ID对应的user信息 | |
| User.add(UserVo newUserVo) | 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. **order模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Order.createOrder | 语法 | Public ResultMessage createOrder(orderVO OrderVO) |
| 前置条件 | 客户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 |
| Order.executeOrder | 语法 | Public ResultMessage executeOrder (String orderID) |
| 前置条件 | 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 |
| Order.undoAbnormalOrder | 语法 | Public ResultMessage undoAbnormalOrder(String orderID) |
| 前置条件 | 订单状态为异常 |
| 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 |
| Order.undoNormalOrder | 语法 | Public ResultMessage undoNormalOrder(String orderID) |
| 前置条件 | 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 |
| Order.getOrderDetail | 语法 | Public OrderVO getOrderDetail(String orderID) |
| 前置条件 | 知道订单号，需要知道订单内容 |
| 后置条件 | 系统返回该订单号的订单 |
| Order.getAllGuestOrderGeneral | 语法 | Public List<OrderGeneralVO> getAllGuestOrderGeneral (String guestID) |
| 前置条件 | 已经登录，需要知道的订单号 |
| 后置条件 | 系统返回查找到的订单结果 |
| Order.getAllHotelOrderGeneral | 语法 | Public List<OrderGeneralVO> getAllHotelOrderGeneral (String hotelID) |
| 前置条件 | 酒店工作人员已经登录 |
| 后置条件 | 系统返回该酒店所有订单 |
| Order.getBookedHotel | 语法 | Public List<HotelID> getBookedHotel (String guestID) |
| 前置条件 | 客户需要知道所有住过酒店的信息 |
| 后置条件 | 系统返回所有该客户已住过酒店的ID列表 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| OrderDataService.createOrder(orderVO OrderVO) | 根据提供的po增加持久化对象 | |
| OrderDataService.executeOrder(String orderID) | 修改单一持久化对象 | |
| OrderDataService.getOrderDetail(String orderID) | 根据ID查找持久化对象 | |
| OrderDataService.getAllHotelOrderGeneral(String hotelID) | 根据ID查找持久化对象 | |
| OrderDataService.getAllGuestOrderGeneral(String guestID) | 根据ID查找持久化对象 | |
| OrderDataService.undoNormalOrder(String orderID) | 修改单一持久化对象 | |
| OrderDataService.UndoAbnormalOrder(String orderID) | 修改单一持久化对象 | |
| Hotel.getHotelInfo(String userID) | 获得单一对象 | |
| User.getSingle(String userID,UserType userType) | 获得单一客户的持久化对象信息 | |
| Promotion.getDiscount(PreOrder preOrder) | 获得单一对象 | |

1. **hotel模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| hotel.getHotelList | 语法 | Public List<hotelGeneralVO> getHotelList(AddressVO addressVO) |
| 前置条件 | 已知地址和商圈的信息，需要浏览商圈内所有酒店的概况 |
| 后置条件 | 系统根据地址和商圈返回符合要求的酒店信息列表 |
| hotel.getSortedHotels | 语法 | Public List<hotelGeneralVO> getSortedHotels(SortStrategy sortStrategy) |
| 前置条件 | 已知该商圈的所有酒店列表以及排序的规则（价格，星级，评分），需要对当前酒店列表按照特定的规则排序 |
| 后置条件 | 系统返回以排序规则排好序的酒店列表 |
| hotel.getBookedHotels | 语法 | Public List<hotelGeneralVO> getBookedHotels(String userID) |
| 前置条件 | 已知该商圈的所有酒店列表以及客户的编号，需要查看该客户已经预定过的酒店 |
| 后置条件 | 系统根据客户的编号，返回该客户已经预定过的酒店列表 |
| hotel.getHotelDetail | 语法 | Public hotelVO  getHotelDetail(String hotelID) |
| 前置条件 | 客户需要查看酒店的详细信息，已知酒店的编号 |
| 后置条件 | 系统返回该酒店的详细信息 |
| hotel.getUncommentedHotels | 语法 | Public List<hotelGeneralVO> getUncommentedHotels(String userID) |
| 前置条件 | 客户查看已执行订单并要评价还未评价的酒店，已知客户的编号 |
| 后置条件 | 返回客户所有已执行订单的酒店列表 |
| hotel.updateEvaluation | 语法 | Public ResultMessage update  Evaluation(EvaluationVO evaluationVO, float mark,String comment) |
| 前置条件 | 已知客户当前评价的酒店信息表以及客户选定的评分等级和输入的评论，客户确认更新评论 |
| 后置条件 | 系统更新并持久化保存当前酒店评价信息表，并更新酒店评分值，系统提示客户评论是否成功 |
| hotel.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 |
| hotel.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 |
| hotel.getHotelRoomInfo | 语法 | Public hotelRoomVO 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 |
| hotel.updateHotelRoomInfo | 语法 | Public ResultMessage updateHotelRoomInfo (hotelRoomVO hotelRoomVO) |
| 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 |
| hotel.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 |
| hotel.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 |
| hotel.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 |
| hotel.updateRemainRoomInfo | 语法 | Public ResultMessage updateRemainRoomInfo (Operation operation, Map<RoomType, int> roomInfo) |
| 前置条件 | 在线下客户入住和退房处理中, 酒店工作人员确认选择的操作及输入的房型和该房型的数量或者酒店工作人员已执行订单以及为已执行订单退房 |
| 后置条件 | 系统更新并持久化保存酒店剩余客房信息, 并提示酒店工作人员是否保存成功 |
| hotel.add | 语法 | Public add (HotelInfo hotelInfo) |
| 前置条件 | 网站管理人员确认添加酒店, 已知添加的酒店信息 |
| 后置条件 | 系统更新并持久化保存添加的酒店信息 |
| 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. **promotion模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 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 |
| promotion.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 |
| promotion.getDiscount | 语法 | Public float getDiscout(PreOrder preOrder) |
| 前置条件 | 客户在填写订单时，订单需要知道当前订单所能使用的最大折扣率，已知该预定酒店的编号，商圈和地址信息，以及该客户是否为会员 |
| 后置条件 | 系统返回当前订单所能使用的最大折扣率 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| promotionDataService.  getPromotions(String userID) | 得到数据库中对应用户所管理的策略列表 | |
| promotionDataService.  setPromotions(String userID,List<promotionPO> list) | 修改数据库中对应用户所管理的策略列表 | |

1. **credit模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| credit.charge | 语法 | Public ResultMessage charge(int chargeNum) |
| 前置条件 | 网站营销人员为客户信用充值 |
| 后置条件 | 系统为客户信用值充值并返回是否成功 |
| credit.getBasicInfo | 语法 | Public CreditVO getBasicInfo (String ID) |
| 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 | 系统返回与该ID对应的包含信用值及信用记录的客户个人基本信息 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| User.getSingle(String userID, UserType userType) | 返回与该ID对应的用户信息 | |
| User.modifyCredit(float creditNum) | 返回客户信用值是否修改成功 | |
| market.getMemberFormulation() | 返回现有的会员等级制度信息 | |
| order.getAllGuestOrderGeneral (String guestID) | 返回与该ID对应的客户所有订单概况 | |

1. **member模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Memeber.add | 语法 | Public ResultMessage add(MemberVO memberVO) |
| 前置条件 | 会员信息符合规则且齐全 |
| 后置条件 | 系统根据已指定会员类型持久化为该客户增加会员数据 |
| Member.modify | 语法 | Public ResultMessage modify(MemberVO memberVO) |
| 前置条件 | 会员信息符合规则且齐全 |
| 后置条件 | 系统修改该会员并持久化保存该客户的会员数据 |
| Member.getMemberInfo | 语法 | Public MemberVO getMemberInfo(String userID,MemberType memberType) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 系统返回与该userID对应的指定会员类型 |
| Member.isMember | 语法 | public boolean isMember(String userID, MemberType memberType) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 返回该userID的用户是否为会员 |
| Member.getMemberType | 语法 | public MemberType getMemberType(String userID, MemberType memberType) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 系统返回该userID的会员类型 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| GuestDataService.ModifyMember(MemberPO memberPO) | 根据提供的memberPO修改单一持久化客户对象的信用值 | |
| GuestDataService.  getSingle (String userID) | 根据ID查找客户持久化对象 | |

1. **user模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| User.getAll | 语法 | Public List<UserVO> getAll(UserType userType) |
| 前置条件 | 无 |
| 后置条件 | 系统返回指定用户类型的所有用户信息 |
| User.getSingle | 语法 | Public UserVO getSingle(String userID，UserType userType) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 系统返回与该ID对应的user信息 |
| User.add | 语法 | Public ResultMessage add(UserVO newUserVO) |
| 前置条件 | 输入的userID不存在，password和所有信息符合规范 |
| 后置条件 | 系统新增该用户并持久化保存该用户的数据 |
| User.addHotel | 语法 | Public ResultMessage addHotel(HotelVO newHotelVO, String hotelID) |
| 前置条件 | 输入的hotelID不存在,所有酒店信息符合规范 |
| 后置条件 | 系统新增该酒店并持久化保存该酒店的数据 |
| User.modify | 语法 | Public ResultMessage modify(UserVO userVO) |
| 前置条件 | 用户信息符合规则且齐全 |
| 后置条件 | 系统修改该用户信息并持久化保存该用户数据 |
| User.modifyCredit | 语法 | Public ResultMessage modifyCredit (float creditNum) |
| 前置条件 | 网站营销人员为客户信用充值 |
| 后置条件 | 系统修改该客户信用值并持久化保存该用户数据 |
| User.getLoginInfo | 语法 | Public ResultMessage getLoginInfo(String userID, UserType userType) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 系统返回与该用户ID对应的密码  如果ID不存在，则系统返回该ID对应用户不存在 |
|  | | |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| guestDataService.  add(GuestPO newGuestPO) | 根据提供的newGuestPO增加客户持久化对象 | |
| guestDataService.  getSingle (String guestID) | 根据userID查找客户持久化对象 | |
| guestDataService.modify  (GuestPO guestPO) | 修改单一客户持久化对象 | |
| guestDataService.init() | 清空所有客户持久化对象 | |
| hotelWorkerDataService.  add(HotelWorkerPO newHotelWorkerPO) | 根据提供的newHotelWorkerPO增加酒店工作人员持久化对象 | |
| hotelWorkerDataService.  getSingle (String hotelWorkerID) | 根据userID查找酒店工作人员持久化对象 | |
| hotelWorkerDataService.modify  (HotelPO hotelPO) | 修改单一酒店工作人员持久化对象 | |
| hotelWorkerDataService.init() | 清空所有酒店工作人员持久化对象 | |
| webMarketerDataService.  add(WebMarketerPO newWebMarketerPO) | 根据提供的newWebMarkterPO增加网站营销人员持久化对象 | |
| webMarkterDataService.  getSingle (String webMarketerID) | 根据userID查找网站营销人员持久化对象 | |
| webMarkterDataService.modify(WebMarketerPO webMarketerPO) | 修改单一网站营销人员持久化对象 | |
| webMarketerDataService.init() | 清空所有网站营销人员持久化对象 | |
| webManagerDataService.  add(WebManagerPO newWebManagerPO) | 根据提供的newWebManagerPO增加网站管理人员持久化对象 | |
| webManagerDataService.  getSingle (String webManagerID) | 根据userID查找网站管理人员持久化对象 | |
| webManagerDataService.modify  (WebManagerPO webManagerPO) | 修改单一网站管理人员持久化对象 | |
| webManagerDataService.init() | 清空所有网站管理人员持久化对象 | |
| Hotel.add | 新增该酒店并持久化保存该酒店的数据 | |

1. **market模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| market.getMemberFormulation | 语法 | Public MarketVO getMemberFormulation () |
| 前置条件 | 无 |
| 后置条件 | 返回现有的会员等级制度信息 |
| market.setMemberFormulation | 语法 | Public ResultMessage setMemberFormulation (MarketVO marketVO) |
| 前置条件 | 网站营销人员输入会员等级制度 |
| 后置条件 | 系统记录此会员等级制度信息 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| marketDataService.getMemberFormulation | 返回现有的会员等级制度信息 | |
| marketDataService.setMemberFormulation(MarketVO marketVO) | 修改会员等级制度信息持久化对象，系统返回是否成功记录 | |

## 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。例如，AccountBusinessService的服务主要由AccountDataService提供。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。以AccountDataService为例，数据层模块的具体描述下图所示：

### 数据层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| orderData | 负责保存订单信息，进行增、改、查的操作 |
| hotelData | 负责保存酒店信息，进行增、改、查的操作 |
| promotionData | 负责保存促销策略信息，进行增、删、改、查的操作 |
| marketData | 负责保存会员等级制度信息，进行改、查的操作 |
| guestData | 负责保存客户信息，进行增、改、查的操作 |
| hotelWorkerData | 负责保存酒店工作人员信息，进行增、改、查的操作 |
| webMarketerData | 负责保存网站营销人员信息，进行改、查的操作 |
| webManagerData | 负责保存网站管理人员信息，进行改、查的操作 |

### 数据层模块的接口规范

1. **orderData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| orderData.createOrder | 语法 | Public ResultMessage createOrder (OrderPO order) |
| 前置条件 | 不存在相同的orderId |
| 后置条件 | 数据库中添加一条订单数据 |
| orderData.getOrderDetails | 语法 | Public OrderPO getOrderDetail (String orderID) |
| 前置条件 | 无 |
| 后置条件 | 系统返回对应id的OrderPO,如果id不存在，返回Null |
| orderData.getAllGuestOrderGeneral | 语法 | Public List<OrderGeneralPO> getAllGuestOrderGeneral (GuestID GuestID) |
| 前置条件 | 无 |
| 后置条件 | 系统返回对应id的OrderPO,如果id不存在，返回Null |
| orderData. getAllHotelOrderGeneral | 语法 | Public List<OrderGeneralPO> getAllHotelOrderGeneral (GuestID GuestID) |
| 前置条件 | 无 |
| 后置条件 | 系统返回对应id的OrderPO,如果id不存在，返回Null |
| orderData. executeOrder | 语法 | Public ResultMessage executeOrder (String orderID) |
| 前置条件 | 存在相同的orderId且订单状态为未执行 |
| 后置条件 | 数据库中更改未执行订单的状态为已执行 |
| orderData. undoAbnormalOrder | 语法 | Public ResultMessage undoAbnormalOrder (String orderID) |
| 前置条件 | 存在相同的orderId且订单状态为异常 |
| 后置条件 | 数据库中更改并保存订单的状态数据 |
| orderData. undoNormalOrder | 语法 | Public ResultMessage undoNormalOrder (String orderID) |
| 前置条件 | 存在相同的orderId且订单状态为未执行 |
| 后置条件 | 数据库中更改并保存订单的状态 |

1. **hotelData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| hotelData.getHotelList | 语法 | Public List<HotelGeneralPO> getHotelList(AddressPO addressPO) |
| 前置条件 | 已知商圈和地址信息 |
| 后置条件 | 系统在数据库中查询并返回符合该商圈和地址信息的所有酒店列表 |
| hotelData.getHotelDetail | 语法 | Public HotelPO getHotelDetail(String hotelID) |
| 前置条件 | 已知客户点击查看的酒店编号 |
| 后置条件 | 返回保存酒店详细信息的HotelPO对象 |
| hotelData.addEvaluation | 语法 | Public boolean addEvaluation (EvaluationPO evaluationPO) |
| 前置条件 | 已知评论信息的对象evaluationPO以及被评论的酒店编号 |
| 后置条件 | 系统向数据库中添加evaluationPO，并返回添加是否成功的布尔值 |
| hotelData.getHotelInfo | 语法 | Public HotelInfoPO getHotelInfo (String hotelID) |
| 前置条件 | 已知酒店编号，逻辑层想获取酒店的基本信息 |
| 后置条件 | 返回一个保存有该酒店基本信息的HotelInfoPO对象 |
| hotelData.setHotelInfo | 语法 | Public boolean  setHotelInfo(HotelInfoPO hotelInfoPO) |
| 前置条件 | 已知保存有酒店基本信息的HotelInfoPO对象，逻辑层请求修改该酒店的基本信息 |
| 后置条件 | 系统在数据库中更新该HotelInfoPO对象，并返回一个修改是否成功的布尔值 |
| hotelData.getHotelRoomInfo | 语法 | Public HotelRoomInfoPO getHotelRoomInfo(String hotelID) |
| 前置条件 | 已知酒店编号，逻辑层请求查看该酒店的客房信息 |
| 后置条件 | 系统返回一个保存有酒店客房信息的HotelRoomInfoPO的对象 |
| hotelData.setHotelRoomInfo | 语法 | Public boolean setHotelRoomInfo  (HotelRoomInfoPO hotelRoomInfoPO) |
| 前置条件 | 已知一个保存有酒店客房信息的HotelRoomInfoPO对象，逻辑层请求更新数据库的酒店客房信息 |
| 后置条件 | 系统更新数据库里的HotelRoomInfoPO对象，并返回是否修改成功的布尔值 |
| hotelData.setCheckInInfo | 语法 | Public boolean setCheckInInfo (CheckInInfo checkInInfo) |
| 前置条件 | 已知保存有入住的信息的对象，该订单已被执行，需要修改数据库中的入住信息 |
| 后置条件 | 系统修改数据库中的入住信息，并返回一个修改是否成功的布尔值 |
| hotelData.setCheckOutInfo | 语法 | Public boolean setCheckOutInfo (CheckOutInfo checkOutInfo) |
| 前置条件 | 已知保存有退房信息的对象，该客户已办理退房，需要更新数据库里的退房信息 |
| 后置条件 | 系统更新数据库里的退房信息，并返回更新是否成功的布尔值 |
| hotelData.getRemainRoomInfo | 语法 | Public RemainRoomInfoPO getRemainRoomInfo(String hotelID) |
| 前置条件 | 已知酒店的编号，逻辑层请求当前酒店剩余客房的信息 |
| 后置条件 | 系统返回一个保存剩余客房信息的RemainRoomInfoPO对象 |
| hotelData.setRemainRoomInfo | 语法 | Public boolean setRemainRoomInfo  (RemainRoomInfoPO remainRoomInfoPO) |
| 前置条件 | 已知保存有已被修改剩余客房信息的RemainRoomInfo的对象以及当前酒店的编号，逻辑层请求修改此酒店的剩余客房信息 |
| 后置条件 | 系统更新数据库里此酒店的剩余客房信息，并返回一个是否修改成功的布尔值 |
| hotelData.add(HotelInfoPO hotelInfoPO) | 语法 | Public boolean add (HotelInfoPO hotelInfoPO) |
| 前置条件 | 已知需要被添加酒店的HotelInfoPO对象 |
| 后置条件 | 系统向数据库里添加此HotelInfoPO对象，并返回一个添加是否成功的布尔值 |

1. **promotionData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| promotionData.getPromotions | 语法 | Public List<PromotionPO> getPromotions(String userID) |
| 前置条件 | 已知用户编号，逻辑层请求获得该用户（酒店工作人员、网站营销人员）所管理的促销策略 |
| 后置条件 | 返回该用户所管理的促销策略 |
| promotionData.setPromotions | 语法 | Public boolean setPromotions(String userID,List<PromotionPO> list) |
| 前置条件 | 已知用户编号和被修改的促销策略列表，逻辑层请求更新数据库中的策略列表 |
| 后置条件 | 系统更新数据库中的策略列表，并返回一个是否修改成功的布尔值 |

1. **marketData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| marketData.getMemberFormulation | 语法 | Public MarketPO getMemberFormulation() |
| 前置条件 | 无 |
| 后置条件 | 返回现有的会员等级制度信息 |
| marketData.setMemberFormulation | 语法 | Public MessageResult setMemberFormulation (HotelInfoPO hotelInfoPO) |
| 前置条件 | 网站营销人员输入会员等级制度信息 |
| 后置条件 | 系统记录此会员等级制度信息 |

1. **guestData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| guestData.modifyMember | 语法 | Public ResultMessage modifyMember(MemberPO memberPO) |
| 前置条件 | 无 |
| 后置条件 | 根据提供的memberPO修改单一持久化客户对象的信用值 |
| guestData.getSingle | 语法 | Public ResultMessage getSingle (String guestID) |
| 前置条件 | guestID.Length=客户的ID长度10 |
| 后置条件 | 系统返回与该ID对应的客户信息 |
| guestData.add | 语法 | Public ResultMessage add(GuestPO newGuestPO) |
| 前置条件 | 输入的guestID不存在，password和所有客户信息符合规范 |
| 后置条件 | 系统新增该客户并持久化保存该客户的数据 |
| guestData.modify | 语法 | Public ResultMessage modify  (GuestPO guestPO) |
| 前置条件 | 客户信息符合规则且齐全 |
| 后置条件 | 系统修改该用户信息并持久化保存该用户数据 |
| guestData.init | 语法 | Public ResultMessage init() |
| 前置条件 | 无 |
| 后置条件 | 清空所有客户持久化对象 |

1. **hotelWorker模块的接口规范**

|  |  |  |
| --- | --- | --- |
| hotelWorkerData.getSingle | 语法 | Public ResultMessage getSingle (String hotelWorkerID) |
| 前置条件 | hotelWorkerID.Length=酒店工作人员的ID长度8 |
| 后置条件 | 系统返回与该ID对应的酒店工作人员信息 |
| hotelWorkerData.add | 语法 | Public ResultMessage add(HotelWorkerPO newHotelWorkerPO) |
| 前置条件 | 输入的hotelWorkerID不存在，password和所有酒店工作人员信息符合规范 |
| 后置条件 | 系统新增该酒店工作人员并持久化保存该酒店工作人员的数据 |
| hotelWorkerData.modify | 语法 | Public ResultMessage modify  (HotelWorkerPO hotelWorkerPO) |
| 前置条件 | 酒店工作人员信息符合规则且齐全 |
| 后置条件 | 系统修改该酒店工作人员信息并持久化保存该酒店工作人员数据 |
| hotelWorkerData.init | 语法 | Public ResultMessage init() |
| 前置条件 | 无 |
| 后置条件 | 清空所有酒店工作人员持久化对象 |

1. **webMarketer模块的接口规范**

|  |  |  |
| --- | --- | --- |
| webMarketerData.getSingle | 语法 | Public ResultMessage getSingle (String webMarketID) |
| 前置条件 | webMarketID.Length=网站营销人员的ID长度6 |
| 后置条件 | 系统返回与该ID对应的网站营销人员信息 |
| webMarketerData.add | 语法 | Public ResultMessage add(WebMarketerPO newWebMarketerPO) |
| 前置条件 | 输入的webMarketerID不存在，password和所有网站营销人员信息符合规范 |
| 后置条件 | 系统新增该网站营销人员并持久化保存该网站营销人员的数据 |
| webMarketerData.modify | 语法 | Public ResultMessage modify  (WebMarketerPO webMarketerPO) |
| 前置条件 | 网站营销人员信息符合规则且齐全 |
| 后置条件 | 系统修改该网站营销人员信息并持久化保存该网站营销人员数据 |
| webMarketerData.init | 语法 | Public ResultMessage init() |
| 前置条件 | 无 |
| 后置条件 | 清空所有网站营销人员持久化对象 |

1. **webManager模块的接口规范**

|  |  |  |
| --- | --- | --- |
| webManagerData.getSingle | 语法 | Public ResultMessage getSingle (String webManagerID) |
| 前置条件 | webManagerID.Length=网站管理人员的ID长度4 |
| 后置条件 | 系统返回与该ID对应的网站管理人员信息 |
| webManagerData.add | 语法 | Public ResultMessage add(WebManagerPO newWebManagerPO) |
| 前置条件 | 输入的webManagerID不存在，password和所有网站管理人员信息符合规范 |
| 后置条件 | 系统新增该网站管理人员并持久化保存该网站管理人员的数据 |
| webManagerData.modify | 语法 | Public ResultMessage modify  (WebManagerPO webManagerPO) |
| 前置条件 | 网站管理人员信息符合规则且齐全 |
| 后置条件 | 系统修改该网站管理人员信息并持久化保存该网站管理人员数据 |

# 信息视角

## 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。

|  |  |
| --- | --- |
| 类名 | 包含的属性 |
| MemberPO | 会员等级、生日（普通会员）、企业名称（企业会员） |
| GuestPO | MemberPO＋姓名、昵称、用户编号、密码、联系方式，信用值 |
| CreditPO | GuestPO＋信用记录（时间、订单号、动作、信用度变化、信用度结果） |
| HotelWorkerPO | 用户编号、密码、所属酒店 |
| WebMarketPO | 用户编号、密码 |
| WebManagerPO | 用户编号、密码 |
|  |  |
| OrderGeneralPO | 酒店名、地址、价格、入住日期和退房日期、订单状态 |
| OrderPO | 订单编号、酒店名、地址、价格、生成时间、入住时间和退房时间、最晚订单执行时间、订单状态、房间类型、房间数、预计入住人数、入住人姓名、联系方式、特别要求（如有无儿童）、房间号，入住时间、预计离开时间、实际离开时间 |
| CheckInPO | 房间号，入住时间、预计离开时间 |
| CheckOutPO | 实际离开时间 |
|  |  |
| HotelGeneralPO | 酒店名称、所属城市商圈、星级、评分、最低房价 |
| HotelPO | 酒店编号、酒店名称、所属城市商圈、详细地址、星级、评分、简介、设施服务 |
| RoomInfoPO | 类型、数量、原始价格 |
| RemainRoomInfoPO | 类型、数量、原始价格 |
|  |  |
| MarketPO | 等级名称、等级信用度、等级优惠 |
|  |  |
| PromotionPO | 促销策略、折扣率、折扣酒店、开始日期和结束日期 |

## 文件格式

本系统依据不同情况分别采用数据库、序列化、txt文件持久化保存三种方式保存文件