酒店管理系统

HMS (Hotel Management System)

体系结构设计文档

V3.3

钛4+小组 (Titanium Quartet)

蒋文荟 李佩瑶 李珍鸿 李一然

2016/10/16

目录

[更新历史 4](#_Toc465309581)

[一、 引言 5](#_Toc465309582)

[1.1编制目的 5](#_Toc465309583)

[1.2词汇表 5](#_Toc465309584)

[二，产品概述 5](#_Toc465309585)

[三，逻辑视角 6](#_Toc465309586)

[四，组合视角 7](#_Toc465309587)

[4.1开发包图 7](#_Toc465309588)

[4.2 运行时进程 12](#_Toc465309589)

[4.3 物理部署 13](#_Toc465309590)

[五、接口模块 13](#_Toc465309591)

[5.1 模块的职责 13](#_Toc465309592)

[5.2 用户界面层的分解 16](#_Toc465309593)

[5.2.1 用户界面层模块的职责 19](#_Toc465309594)

[5.2.2 用户界面层的接口规范 19](#_Toc465309595)

[5.2.3 用户界面模块设计原理 22](#_Toc465309596)

[5.3业务逻辑层的分解 22](#_Toc465309597)

[5.3.1 业务逻辑层的职责 22](#_Toc465309598)

[5.3.2业务逻辑层的接口规范 23](#_Toc465309599)

[5.4数据层的分解 40](#_Toc465309600)

[5.4.1 数据层模块的职责 40](#_Toc465309601)

[5.4.2 数据层模块的接口规范 41](#_Toc465309602)

[6、信息视角 49](#_Toc465309603)

[6.1 数据持久化对象 49](#_Toc465309604)

[6.2 数据库表 52](#_Toc465309605)

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 李一然 | 2016/10/10 | 编写逻辑模型示意图和客户端及服务器包图 | V1.0 |
| 李珍鸿 | 2016/10/10 | 增加开发包设计表格 | V1.1 |
| 李佩瑶 | 2016/10/10 | 增加界面跳转示意图 | V1.2 |
| 李珍鸿 | 2016/10/13 | 编写部分逻辑层模块的接口规范 | V2.0 |
| 李佩瑶 | 2016/10/13 | 编写部分逻辑层模块的接口规范 | V2.1 |
| 蒋文荟 | 2016/10/14 | 增加数据库表 | V2.2 |
| 李一然 | 2016/10/14 | 修改逻辑层模块的接口规范 | V2.3 |
| 李珍鸿 | 2016/10/14 | 编写引言、产品概述、逻辑视角、组合视角及用户界面层的分解 | V3.0 |
| 蒋文荟 | 2016/10/15 | 修改数据库表，增加数据层的分解 | V3.1 |
| 李珍鸿 | 2016/10/16 | 合并各个部分 | V3.2 |
| 李一然 | 2016/10/16 | 校对 | V3.3 |

# 引言

## 1.1编制目的

本报告详细完成对酒店管理系统的概要设计，达到知道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，测试人员以及最终用户而编写，是了解系统的导航

## 1.2词汇表

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| HMS | 酒店管理系统 |  |
| ui | 用户界面 |  |
| bl | 业务逻辑 |  |
| RMI | 远程方法调用 |  |
| PO | 持久化对象 |  |
| VO | 值对象 |  |

# 二，产品概述

参考酒店管理系统用例文档和酒店管理系统软件需求规格说明中对产品的描述。

# 三，逻辑视角

酒店管理系统中，选择了分层体系结构风格，将系统分为3层（界面层，业务逻辑层，数据层）能够很好地展示整个高层抽象。界面层包含图形用户界面的实现，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一和图二所示：



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

# 四，组合视角

## 4.1开发包图

酒店管理系统的最终开发包设计如表1所示。

表1 酒店管理系统的最终开发包设计

|  |  |
| --- | --- |
| 开发包 | 依赖的其他开发包 |
| mainui | usermainui, personalui, createorderui, strategyui, hotelinfoui, searchhotelui, manageroomui, manageorderui, managehotelui, manageuserui |
| usermainui | usermainblservice , vo, 界面类库包 |
| personalui | personalblservice, vo, 界面类库包 |
| createorderui | createorderblservice, vo, 界面类库包 |
| strategyui | strategyblservice, vo, 界面类库包 |
| hotelinfoui | hotelinfoblservice, vo, 界面类库包 |
| searchhetelui | searchhetelblservice, vo, 界面类库包 |
| manageroomui | manageroomblservice, vo, 界面类库包 |
| manageorderui | manageorderblservice, vo, 界面类库包 |
| managehotelui | managehotelblservice, vo, 界面类库包 |
| manageuserui | manageuserblservice, vo, 界面类库包 |
| usermainblservice |  |
| usermainbl | userdataservice, po |
| personalblservice |  |
| personalbl | userdataservice, creditdataservice, po |
| createorderblservice |  |
| createorderbl | orderdataservice, roomdataservice, creditdataservice, strategydataservice, po |
| strategyblservice |  |
| strategybl | mainuserbl, strategydataservice, locationdataservice, po |
| Hotelinfoblservice |  |
| hotelinfobl | manageorderbl , roomdataservice, strategydataservice, hotelinfodataservice, reviewdataservice, po |
| searchhotelblservice |  |
| searchhotelbl | manageorderbl, roomdataservice, reviewdataservice, locationdataservice, hotelinfodataservice, po |
| manageroomblservice |  |
| manageroombl | mainuserbl, orderdataservice, roomdataservice, po |
| manageorderblservice |  |
| manageorderbl | mainuserbl, orderdataservice, roomdataservice, creditdataservice, po |
| managehotelblservice |  |
| managehotelbl | mainuserbl, userdataservice, hotelinfodataservice, po |
| manageuserblservice |  |
| mangeuserbl | creditdataservice, userdataservice |
| orderdataservice | Java RMI, po |
| orderdata | databaseutility, po |
| roomdataservice | Java RMI, po |
| roomdata | databaseutility, po |
| userdataservice | Java RMI, po |
| userdata | databaseutility, po |
| creditdataservice | Java RMI, po |
| creditdata | databaseutility, po |
| strategydataservice | Java RMI, po |
| strategydata | databaseutility, po |
| hotelinfodataservice | Java RMI, po |
| hoteldatainfo | databaseity, po |
| reviewdataservice | Java RMI, po |
| reviewdata | databaseutility, po |
| Locationdataservice | Java RMI, po |
| locationdata | databaseutility, po |
| po |  |
| vo |  |
| 界面类库包 |  |
| Java RMI |  |
| databaseutility | JDBC |

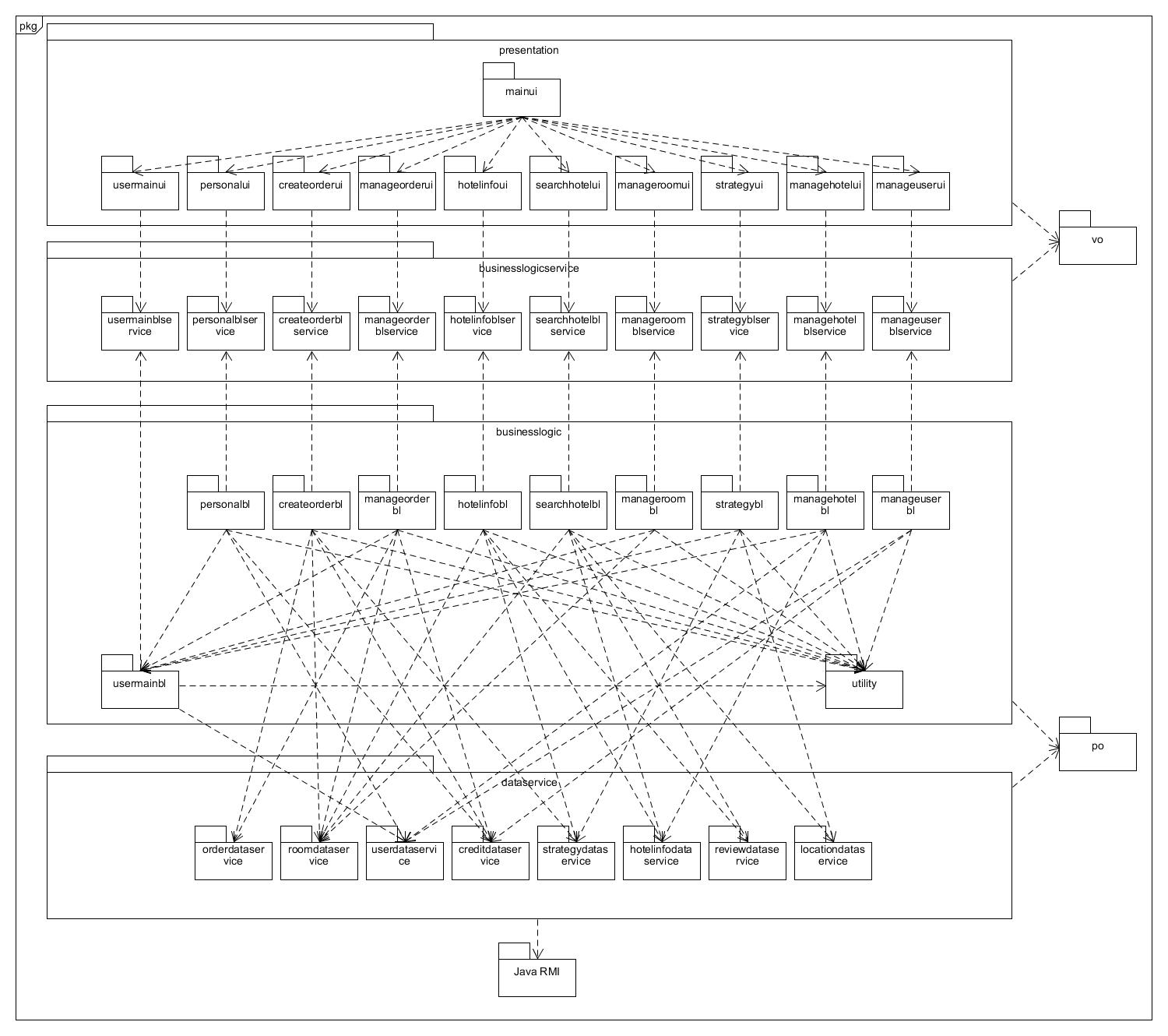


图3 酒店管理系统客户端开发包图



图4 酒店管理系统服务器端开发包图

## 4.2 运行时进程

在酒店管理系统中，会有多个客户端进程和一个服务器端进程，其进程图如图五所示，结合部署图，客户端进程是在客户端机器上运行，服务器进程在服务器端机器上运行。

图5 进程图

## 4.3 物理部署

酒店管理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java RMI构件属于JDK 8.0的一部分。所以，在系统JDK环境已经设计好的情况下，不需要再独立部署。部署图如图6所示。



图6 部署图

# 五、接口模块

## 5.1 模块的职责

客户端模块和服务器模块视图分别如图7和图8所示。客户端各层和服务器各层的职责分别如表2和表3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 | 基于窗口的进销存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,检测网络连接状态，进行断线重连 |

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。层间调用的接口如表四所示

表4 层之间调用的接口

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| usermainblservice  personalblservice  createorderblservice  strategyblservice  hotelinfoblservice  searchhotelblservice  manageroomblservice  manageorderblservice  managehotelblservice  manageuserblservice | 客户端展示层 | 客户端业务逻辑层 |
| orderdataservice  roomdataservice  userdataservice  creditdataservice  strategydataservice  hotelinfodataservice  reviewdataservice  locationdataservice | 客户端业务逻辑层 | 服务器端数据层 |

借用查看酒店详情用例来说明层之间的调用，如图9所示。每一层之间都是由上层依赖了一个接口，而下层实现这个接口。HotelInfoBLService提供了酒店详情界面所需要的所有业务逻辑功能；HotelInfoDataService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 查看酒店详情用例各层之间调用的接口

## 5.2 用户界面层的分解

根据需求，系统存在45个用户界面。它们分别是：修改酒店工作人员信息界面，修改客户信息界面，修改网站营销人员信息界面，查看酒店工作人员信息界面，查看用户信息界面，查看网站营销人员信息界面，查找酒店工作人员界面，查找用户界面，查找网站营销人员界面，网站管理人员界面，添加酒店及其工作人员界面，登录界面，客户注册界面，酒店工作人员界面，查看酒店促销策略界面，查看可用客房界面，线下入住界面，修改酒店信息界面，执行订单界面，更新退房信息界面，浏览酒店订单界面，查看酒店基本信息界面，修改酒店基本信息界面，网站营销人员界面，查看未执行订单界面，撤销异常订单界面，查看网站促销策略界面，修改促销策略界面，会员折扣制度界面，商圈促销策略界面，特殊日期促销界面，信用充值界面，客户信息核对界面，查找客户界面，信用记录界面，客户界面，维护用户信息界面，注册会员界面，搜索酒店信息界面，酒店详情界面，生成订单界面，查看预定过的酒店界面，撤销未执行订单界面，浏览客户订单界面。

界面跳转如图10所示：

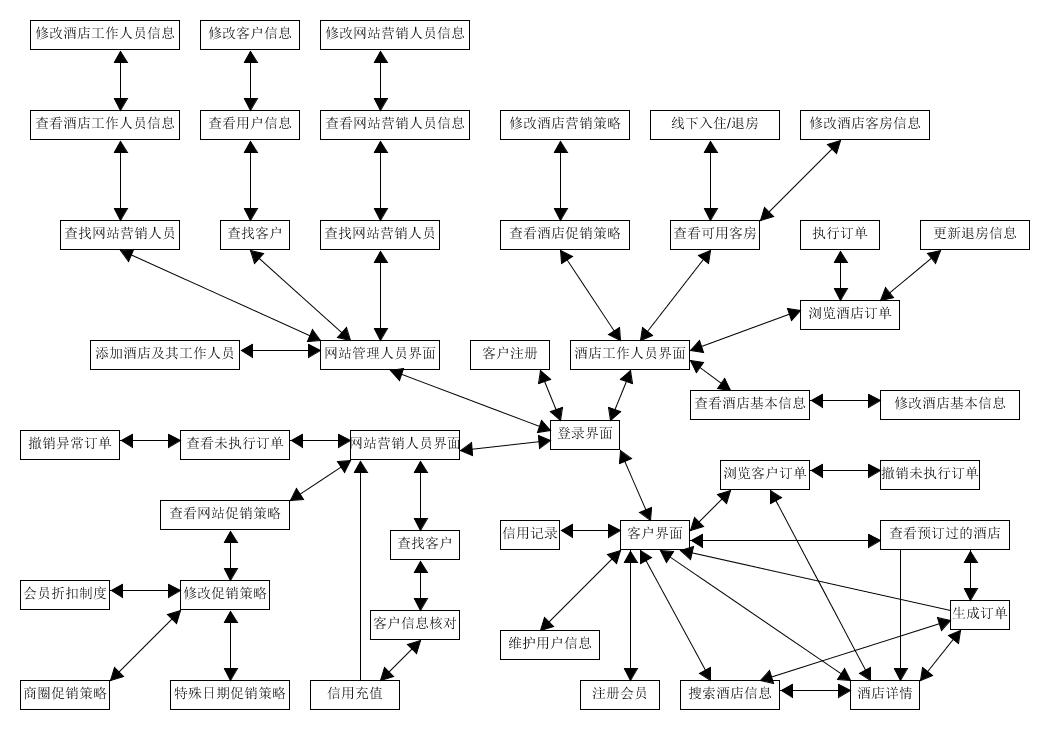


图10 用户界面跳转示意图

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11所示：



图11 用户界面类

### 5.2.1 用户界面层模块的职责

用户界面层模块的职责如表5所示：

表5 用户界面层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| HMSApplication | 用于启动用户界面并管理各个界面场景的初始化 |
| Controller | 用于控制界面与用户的交互 |

### 5.2.2 用户界面层的接口规范

用户界面层模块接口规范如表6所示

表6 用户界面层模块接口规范

|  |  |  |
| --- | --- | --- |
| usermainui | 语法 | loadScene() |
| 前置条件 | 无 |
| 后置条件 | 显示MainScene |
| personalui | 语法 | loadScene() |
| 前置条件 | 无 |
| 后置条件 | 显示用户信息界面 |
| createorderui | 语法 | loadScene() |
| 前置条件 | 无 |
| 后置条件 | 显示createorderScene |
| manageorderui | 语法 | loadScene() |
| 前置条件 | 无 |
| 后置条件 | 显示manageorderScene |
| hotelinfoui | 语法 | loadScene() |
| 前置条件 | 无 |
| 后置条件 | 显示hotelinfoScene |
| searchhotelui | 语法 | loadScene() |
| 前置条件 | 无 |
| 后置条件 | 显示searchhotelScene |
| managerooui | 语法 | loadScene() |
| 前置条件 | 无 |
| 后置条件 | 显示manageroomScene |
| strategyui | 语法 | loadScene() |
| 前置条件 | 无 |
| 后置条件 | 显示strategyScene |
| managehotelui | 语法 | loadScene() |
| 前置条件 | 无 |
| 后置条件 | 显示managehotelScene |
| manageuser\_ui | 语法 | loadScene() |
| 前置条件 | 无 |
| 后置条件 | 显示manageuserScene |

用户界面层模块需要的服务接口如表7所示：

表7 用户界面层模块需要的服务接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| businesslogicservice.usermain\_blservice | 登陆界面的业务逻辑接口 |
| businesslogicservice.personal \_blservice | 处理用户信息的接口 |
| businesslogicservice.createorder\_blservice | 处理与生成订单相关的接口 |
| businesslogicservice.hotelinfoblservice | 处理各种酒店信息的接口 |
| businesslogicservice.manageorderblservice | 提供管理订单的接口 |
| businesslogicservice.strategyblservice | 提供促销策略的接口。 |
| Businesslogicservice.manageroomblservice | 负责管理房间的服务的接口 |
| businesslogicservice.manageuserblservice | 负责管理用户的接口 |
| businesslogicservice.managehotelblservice | 负责管理酒店的服务 |
| businesslogicservice.searchhotelblservice | 提供搜索酒店的接口 |
| businesslogicservice.managehotelblservice | 提供管理酒店的接口 |

### 5.2.3 用户界面模块设计原理

用户界面利用JavaFX界面库来实现。

## 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，User对象负责处理登陆界面的业务逻辑；User对象负责处理管理用户的业务逻辑。业务逻辑层的设计如图12所示。

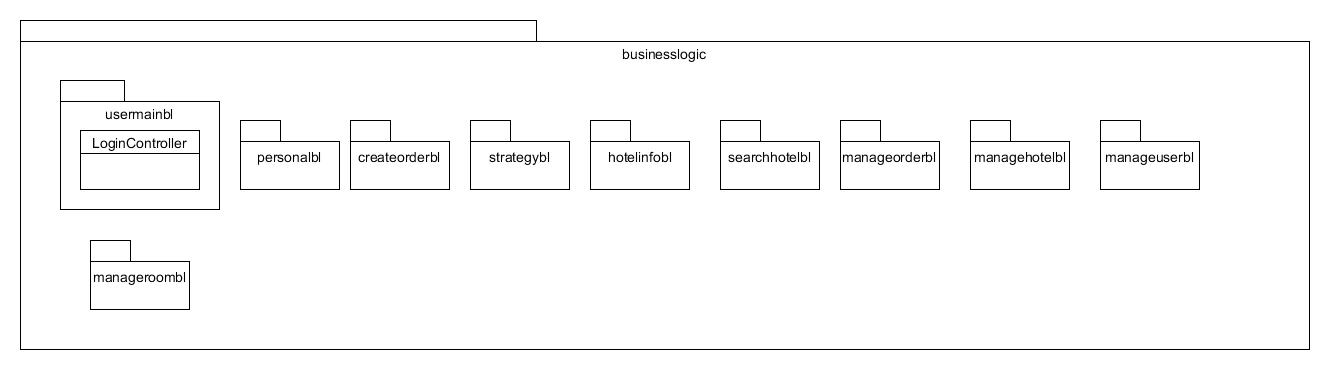


图12 业务逻辑层设计包图

### 5.3.1 业务逻辑层的职责

表8 业务逻辑层各模块及职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| usermainbl | 提供验证用户登录的服务 |
| personalbl | 负责查看用户信息的服务 |
| createorderbl | 负责处理生成订单的服务 |
| managehotelbl | 负责管理酒店的服务 |
| Hotelinfobl | 负责处理酒店信息相关的服务 |
| manageorderbl | 负责管理订单服务。 |
| strategybl | 提供与促销策略相关服务。 |
| manageuserbl | 负责管理用户的服务 |
| manageroombl | 负责管理房间的服务 |
| searchhotelbl | 负责搜索酒店的服务 |

### 5.3.2业务逻辑层的接口规范

表9 usermainbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Usermain.login | 语法 | public UserVO login(long userID, String password); | |
| 前置条件 | Password符合输入规则 | |
| 后置条件 | 查找是否有相应的User，根据输入的密码返回登录验证的结果及相应的用户信息值对象 | |
| Usermain.logout | 语法 | public ResultMessage logout(long userID); | |
| 前置条件 | 已经登录 | |
| 后置条件 | 返回操作结果，相应用户退出登录状态 | |
| Usermain.signUp | 语法 | public ResultMessage signUp(String username, String password); | |
| 前置条件 | 输入的用户名和密码符合规则 | |
| 后置条件 | 保存此客户的用户名和密码 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| DatabaseFacory.getUserDatabase | | | 得到User数据库服务的引用 |
| UserDataService.insert(UserPO user) | | | 在数据库中插入UserPO对象 |

表10 personalbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Personal.getCreditRecord | 语法 | public List<CreditVO> getCreditRecord (long userID) | |
| 前置条件 | 已经登录为客户 | |
| 后置条件 | 返回当前客户的信用记录列表 | |
| Personal.memberSignIn | 语法 | public ResultMessage memberSignIn(PersonalVO personal) | |
| 前置条件 | 已经登录为客户 | |
| 后置条件 | 返回操作结果，保存会员信息 | |
| Personal.manageInfo | 语法 | public ResultMessage manageInfo (PersonalVO personal) | |
| 前置条件 | 已经登录为网站管理人员 | |
| 后置条件 | 返回操作结果，若成功则系统保存修改后的用户信息 | |
| Personal.getPersonal | 语法 | public PersonalVO getPersonal (long userID) | |
| 前置条件 | 已经登录 | |
| 后置条件 | 返回相应用户的信息 | |
| Personal.modifyPersonal | 语法 | public ResultMessage modifyPersonal (PersonalVO personal) | |
| 前置条件 | 已经进入用户信息界面 | |
| 后置条件 | 返回操作结果，若成功则系统保存修改后的相应用户个人信息 | |
| Personal.getHotelList | 语法 | public List<HotelVO> getHotelList (long userID) | |
| 前置条件 | 已经登录为客户 | |
| 后置条件 | 返回当前客户预订过的酒店列表 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| UserDataService.getUserInfo(long userID) | | | 根据用户编号进行查找多个持久化对象 |
| UserDataService.update(PersonalPO personal) | | | 更新单一持久化对象 |
| OrderDataService.getHotelList(long userID) | | | 根据用户编号进行查找多个持久化对象 |
| HotelInfoDataService.briefHotelInfo(long hotelID) | | | 根据酒店编号进行查找单个持久化对象 |
| CreditDataService.getRecord(long userID) | | | 根据用户编号进行查找多个持久化对象 |
| UserDataService.insert(PersonalPO personal) | | | 插入单一持久化对象 |
| DatabaseFactory.getPersonalDatabase | | | 得到Personal数据库的服务的引用 |
| DatabaseFactory.getCreditDatabase | | | 得到Credit数据库的服务的引用 |
| DatabaseFactory.getOrderDatabase | | | 得到Order数据库的服务的引用 |
| DatabaseFactory.getHotelDatabase | | | 得到Hotel数据库的服务的引用 |

表11 createorderbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| CreateOrder.selectRoomType | 语法 | public ResultMessage selectRoomType (long hotelID, int roomTypeID); | |
| 前置条件 | 已经登录为客户，已经选定订房起止日期 | |
| 后置条件 | 系统返回操作结果，若成功则系统保存相应的酒店编号及客房类型编号 | |
| CreateOrder.orderInfo | 语法 | public ResultMessage orderInfo (OrderInfoVO orderInfo); | |
| 前置条件 | 已经登录为客户，已经选定客房类型 | |
| 后置条件 | 系统返回操作结果，若成功则系统保存正在生成的订单的最晚订单执行时间、入住人姓名及酒店编号 | |
| CreateOrder.endCreateOrder | 语法 | public void endCreateOrder() | |
| 前置条件 | 已支付 | |
| 后置条件 | 系统处理订单信息，并改变相应客房状态 | |
| CreateOrder.getOptimal | 语法 | public OrderStrategyVO getOptimal() | |
| 前置条件 | 客户已完成订单信息填写 | |
| 后置条件 | 返回此订单的最优折扣策略信息及相应价格 | |
| CreateOrder.otherStrategy | 语法 | public List<OrderStrategyVO> otherStrategy () | |
| 前置条件 | 客户已完成订单信息填写 | |
| 后置条件 | 返回此订单可使用的策略信息及相应价格 | |
| CreateOrder.selectStrategy | 语法 | public ResultMessage selectStrategy (long strategyID); | |
| 前置条件 | 已完成订单信息填写 | |
| 后置条件 | 返回操作结果，若成功则系统保存此订单的信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| OrderDataService.insert(OrderPO order) | | | 插入单一持久化对象 |
| HotelInfoDataService.getRoomType(long hotelID, int roomTypeID) | | | 根据酒店编号和客房类型编号进行查找单一持久化对象 |
| StrategyDataService.searchByHotel(long hotelID) | | | 根据酒店编号进行查找多个持久化对象 |
| UserDataService.getCreditBalance(long userID) | | | 根据用户编号进行查找单一持久化对象 |
| DatabaseFactory.getOrderDatabase | | | 得到Order的数据库的服务的引用 |
| DatabaseFactory.getPersonalDatabase | | | 得到Personal的数据库的服务的引用 |
| DatabaseFactory.getStrategyDatabase | | | 得到Strategy的数据库的服务的引用 |

表12 strategybl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Strategy.addStrategy | 语法 | public ResultMessage addStrategy(StrategyVO); | |
| 前置条件 | 网站人员已经点击一类网站营销策略 | |
| 后置条件 | 返回操作结果，若成功则系统添加相应的营销策略 | |
| Strategy.deleteStrategy | 语法 | public ResultMessage deleteStrategy (long strategyID) | |
| 前置条件 | 网站营销人员或者酒店管理人员已经登录 | |
| 后置条件 | 返回操作结果，若成功则系统删除编号对应的营销策略 | |
| Strategy.getStrategy | 语法 | public List<StrategyVO> getStrategy(long userID) | |
| 前置条件 | 已登录为网站营销人员或酒店工作人员 | |
| 后置条件 | 根据用户编号返回相应营销策略列表 | |
| Strategy.modifyStrategy | 语法 | public ResultMessage changeStrategy(StrategyVO); | |
| 前置条件 | 酒店工作人员或网站营销人员对当前已有的营销策略进行了修改 | |
| 后置条件 | 返回操作结果，若成功则系统保存修改后的相应营销策略 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| StrategyDataService.insert(StrategyPO strategy) | | | 插入单一持久化对象 |
| StrategyDataService.delete(long strategyID) | | | 删除单一持久化对象 |
| StrategyDataService.update(StrategyPO strategy) | | | 更新单一持久化对象 |
| DatabaseFactory.getStrategyDatabase | | | 得到strategy的数据库的服务的引用 |

表13 5hotelinfo模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| HotelInfo.getHotelDetails | 语法 | public HotelDetailsVO getHotelDetails (long hotelID); | |
| 前置条件 | 已经登录为客户且需要显示酒店详情页面 | |
| 后置条件 | 返回该酒店的详细信息 | |
| HotelInfo.availableRoomType | 语法 | public List<RoomTypeVO> availableRoomType (Calendar start, Calendar end, long hotelID); | |
| 前置条件 | 已经登录为客户 | |
| 后置条件 | 返回满足该条件的所有可用房型 | |
| HotelInfo.reviewHotel | 语法 | public ResultMessage reviewHotel(ReviewVO review) | |
| 前置条件 | 已经登录为客户且评论超过三个字符 | |
| 后置条件 | 返回操作结果，若成功则系统保存相应的评价 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| HotelInfoDataService.searchHotelDetials(long hotelID) | | | 根据酒店编号进行查找单一持久化对象 |
| HotelInfoDataService.updateRate(long HotelID, int newRate) | | | 更新单一持久化对象 |
| HotelInfoDataService.getRoomTypes(long HotelID) | | | 根据酒店编号进行查找多个持久化对象 |
| ReviewDataService.search(long hotelID) | | | 根据酒店编号进行查找多个持久化对象 |
| ReviewDataService.insert(ReviewPO review) | | | 插入单一持久化对象 |
| DatabaseFactory.getOrderDatabase | | | 得到Order的数据库的服务的引用 |
| DatabaseFactory.getRoomDatabase | | | 得到Room的数据库的服务的引用 |
| DatabaseFactory.getReviewDatabase | | | 得到Review的数据库的服务的引用 |
| DatabaseFactory.getHotelInfoDatabase | | | 得到HotelInfo的数据库的服务的引用 |
| DatabaseFecory.getStrategyDatabase | | | 得到Strategy的数据库的服务的引用 |

表14 searchhotelbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| SearchHotel. getHotelList | 语法 | public List<HotelVO> getHotelList (int cityID, int districtID); | |
| 前置条件 | 已经登录为客户 | |
| 后置条件 | 返回根据城市和商圈编号查找到的酒店列表 | |
| SearchHotel. filter | 语法 | public List<HotelVO> filter (HotelFilterVO filter); | |
| 前置条件 | 已经登录为客户且显示商圈的酒店列表 | |
| 后置条件 | 根据筛选条件查找符合要求的酒店列表并返回 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| HotelInfoDataService.getHotelList(int cityID, int districtID); | | | 根据城市及商圈编号进行查找多个持久化对象 |
| RoomDataService.isAvailable(Calendar start, Calendar end, int roomTypeID, int numOfRoom) | | | 根据客房类型及特定日期预订状态进行查找多个持久化对象 |
| OrderDataService.hasBeenOrdered(long hotelID, long userID) | | | 根据酒店编号及订房客户的用户名进行查找多个持久化对象 |
| DatabaseFactory. getHotelInfo | | | 得到对HotelInfo数据库的服务的引用 |
| DatabaseFactory. getRoom | | | 得到对Room数据库的服务的引用 |
| DatabaseFactory. getOrder | | | 得到对Order数据库的服务的引用 |
| DatabaseFactory. getLocation | | | 得到对Location数据库的服务的引用 |

表15 manageroombl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| ManageRoom. getRoomList | 语法 | public List<RoomVO> getRoomList (long hotelID); | |
| 前置条件 | 已经登录为酒店工作人员 | |
| 后置条件 | 返回相应酒店所有可用客房的列表 | |
| ManageRoom. modifyRoomInfo | 语法 | public ResultMessage modifyRoomInfo (RoomVO room); | |
| 前置条件 | 已经登录为酒店工作人员 | |
| 后置条件 | 返回操作结果，若成功则系统保存修改后的相应房间信息 | |
| ManageRoom. addRoom | 语法 | public ResultMessage addRoom (RoomVO room); | |
| 前置条件 | 已经登录为酒店工作人员 | |
| 后置条件 | 返回操作结果，若成功则系统添加相应的房间信息 | |
| ManageRoom. deleteRoom | 语法 | public ResultMessage deleteRoom (int roomID); | |
| 前置条件 | 已经登录为酒店工作人员 | |
| 后置条件 | 返回操作结果，若成功则系统删除相应的房间信息 | |
| ManageRoom. checkIn | 语法 | public ResultMessage checkIn (int roomID); | |
| 前置条件 | 已经登录为酒店工作人员 | |
| 后置条件 | 返回操作结果，若成功则系统将相应的客房设置为已入住状态 | |
| ManageRoom. checkOut | 语法 | public ResultMessage checkOut (int roomID); | |
| 前置条件 | 已经登录为酒店工作人员 | |
| 后置条件 | 返回操作结果，若成功则系统将相应的客房设置为空闲状态 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| DatabaseFactory. getRoom | | | 获得对Room数据库的服务的引用 |
| RoomDataService. update (RoomPO room) | | | 更新单一持久化对象 |
| RoomDataService. insert (RoomPO room) | | | 插入单一持久化对象 |
| RoomDataService. delete (int roomID) | | | 删除单一持久化对象 |
| RoomDataService. checkIn (int roomID) | | | 更新单一持久化对象 |
| RoomDataService. checkOut (int roomID) | | | 更新单一持久化对象 |

表16 manageorderbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| ManageOrder. clientOrderList | 语法 | public List<OrderVO> clientOrderList (char orderState, long userID); | |
| 前置条件 | 已经登录为客户 | |
| 后置条件 | 返回客户该状态的订单列表 | |
| ManageOrder. getOrderList | 语法 | public List<OrderVO> hotelOrderList (long hotelID, OrderFilterVO orderFilter, int page); | |
| 前置条件 | 已经登录为酒店工作人员或网站营销人员 | |
| 后置条件 | 返回全网站或酒店符合筛选条件的相应页码的订单列表 | |
| ManageOrder. clientCancelOrder | 语法 | public ResultMessage clientCancelOrder (String orderID); | |
| 前置条件 | 已经登录为客户 | |
| 后置条件 | 返回操作结果，若成功则系统撤销相应的订单 | |
| ManageOrder. marketerCancelOrder | 语法 | public ResultMessage marketerCancelOrder (String orderID, CreditRestore restore); | |
| 前置条件 | 已经登录为网站营销人员 | |
| 后置条件 | 返回操作结果，若成功则系统撤销订单并相应地恢复信用值 | |
| ManageOrder. clientOrderList | 语法 | public List<OrderVO> clientOrderList (long userID, long hotelID); | |
| 前置条件 | 已经登录为网站营销人员 | |
| 后置条件 | 返回该用户在该酒店的所有订单列表 | |
| ManageOrder. checkIn | 语法 | public ResultMessage checkIn(String roomNum, Time leaveTime, String orderID); | |
| 前置条件 | 已经登录为酒店工作人员 | |
| 后置条件 | 返回操作结果，若成功则系统修改相应订单及客房的状态 | |
| ManageOrder. checkOut | 语法 | public ResultMessage checkOut(int orderID); | |
| 前置条件 | 已经登录为酒店工作人员 | |
| 后置条件 | 返回操作结果，若成功则系统修改相应订单及客房的状态 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| OrderDataService. update(OrderPO order) | | | 更新单一持久化对象 |
| OrderDataService. getOrderPage(long hotelID, OrderFilterVO orderFilter, int page) | | | 根据筛选条件进行查找多个持久化对象 |
| RoomDataService. checkIn (int roomID) | | | 更新单一持久化对象 |
| RoomDataService. checkOut (int roomID) | | | 更新单一持久化对象 |
| DatabaseFactory. getOrder | | | 获得对Order数据库的服务的引用 |
| DatabaseFactory. getRoom | | | 获得对Room数据库的服务的引用 |
| DatabaseFactory. getCredit | | | 获得对Credit数据库服务的引用 |

表17 managehotelbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Managehotel. getHotelInfo | 语法 | public HotelInfoVO getHotelInfo(long userID); | |
| 前置条件 | 已经登录为酒店管理人员 | |
| 后置条件 | 返回酒店信息 | |
| Managehotel. modifyHotelInfo | 语法 | public ResultMessage modifyHotelInfo(HotelInfoVO hotelInfo); | |
| 前置条件 | 已经登录为酒店管理人员，且对酒店信息做出了修改 | |
| 后置条件 | 返回操作结果，若成功则系统保存酒店信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| DatabaseFactory. getHotelInfo | | | 获得对HotelInfo数据库的服务的引用 |
| DatabaseFactory. getUser | | | 获得对User数据库的服务的引用 |
| HotelInfoDataService.update(HotelInfoPO hotelInfo) | | | 修改单一持久化对象 |
| HotelInfoDataService.searchHotelDetails(long hotelID) | | | 根据酒店编号进行查找单一持久化对象 |

表18 manageuserbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Manageuser. searchUser | 语法 | public List<UserPO> searchUser(String username, String realName); | |
| 前置条件 | 已经登录为网站管理人员 | |
| 后置条件 | 返回用户信息列表 | |
| Manageuser. searchHotelStaff | 语法 | public List<HotelStaffPO> searchHotelStaff(int cityID, int districtID); | |
| 前置条件 | 已经登录为网站管理人员 | |
| 后置条件 | 返回符合要求的酒店工作人员信息列表 | |
| Manageuser. getUser | 语法 | public UserPO getUser(long userID); | |
| 前置条件 | 已经登录为网站管理人员 | |
| 后置条件 | 返回该用户的信息 | |
| Manageuser. getUserList | 语法 | public ArrayList<UserPO> getUserList(); | |
| 前置条件 | 无 | |
| 后置条件 | 返回客户名单列表 | |
| Manageuser. sortUserList | 语法 | public ArrayList<UserPO> sortUserList(String way); | |
| 前置条件 | 营销人员对用户排序，已知排序的方式 | |
| 后置条件 | 返回按照要求排序的客户名单列表 | |
| Manageuser. credit | 语法 | public ResultMessage credit(double money, long userID); | |
| 前置条件 | 已登录为网站营销人员 | |
| 后置条件 | 返回充值结果信息，若成功更新客户信用值 | |
| Manageuser. addHotel | 语法 | public ResultMessage addHotel(String hotlName); | |
| 前置条件 | 添加酒店，已知酒店名称 | |
| 后置条件 | 返回添加结果信息，若成功添加酒店数据 | |
| Manageuser. addHotelStaff | 语法 | public ResultMessage add(String username, String password); | |
| 前置条件 | 添加酒店工作人员，已知酒店编号 | |
| 后置条件 | 返回添加结果信息，若成功增加酒店工作人员 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| DatabaseFactory. getUser | | | 获得对User数据库的服务的引用 |
| DatabaseFactory. getCredit | | | 获得对Credit数据库服务的引用 |
| HotelDataService. insert(HotelDetailsPO hotel) | | | 添加单一持久化对象 |
| UserDataService. insert(PersonalPO personal) | | | 插入单一持久化对象 |
| UserDataService. searchClient(String username, String realName) | | | 根据用户名及订房客户的真实姓名进行查找多个持久化对象 |
| UserDataService. getUserInfo(long userID) | | | 根据用户编号进行查找单一持久化对象 |
| UserDataService. update(PersonalPO personal) | | | 更新单一持久化对象 |

## 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。User业务逻辑需要的服务由UserDataService接口提供。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。数据层模块的具体描述如图5.4所示。

* 

图12 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19所示。

表19 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。 |
| DataServiceTxtImpl | 基于TXT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 |
| DataServiceDatabaseImpl | 基于数据库的持久化数据的接口，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20 数据层orderdata模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderDataService.getHotelList | 语法 | public List<OrderPO> getHotelList(long userID) |
| 前置条件 | 无 |
| 后置条件 | 显示客户的酒店订单列表 |
| OrderDataService.insert | 语法 | public ResultMessage insert(OrderPO order) |
| 前置条件 | 无 |
| 后置条件 | 增加一个酒店订单 |
| OrderDataService. update | 语法 | public ResultMessage update(OrderPO order) |
| 前置条件 | 在数据库中存在同样ID的PO |
| 后置条件 | 更新酒店订单 |
| OrderDataService.hasBeenOrdered | 语法 | public boolean hasBeenOrdered(long hotelID, long userID) |
| 前置条件 | 无 |
| 后置条件 | 返回是否存在该用户在相应酒店的订单记录 |
| OrderDataService. getOrderPage | 语法 | public List<OrderPO> getOrderPage (long hotelID, OrderFilterVO orderFilter, int page) |
| 前置条件 | 无 |
| 后置条件 | 返回相应页码的符合筛选条件的订单列表 |

表21数据层roomdata模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| RoomDataService. update | 语法 | public ResultMessage update(RoomPO room) |
| 前置条件 | 在数据库中存在同样ID的PO |
| 后置条件 | 更新客房信息 |
| RoomDataService. insert | 语法 | public ResultMessage insert(RoomPO room) |
| 前置条件 | 无 |
| 后置条件 | 增加一条客房信息 |
| RoomDataService. delete | 语法 | public ResultMessage delete(int roomID) |
| 前置条件 | 在数据库中存在同样ID的PO |
| 后置条件 | 删除一条客房信息 |
| RoomDataService.isAvailable | 语法 | public boolean isAvailable(Calendar start, Calendar end, int roomTypeID, int numOfRoom) |
| 前置条件 | 无 |
| 后置条件 | 返回是否存在足够在给定日期范围内可预订的给定类型客房 |
| RoomDataService.checkIn | 语法 | public ResultMessage checkIn (int roomID) |
| 前置条件 | 该客房目前为空闲状态 |
| 后置条件 | 修改该客房状态为已入住 |
| RoomDataService.checkOut | 语法 | public ResultMessage checkOut (int roomID) |
| 前置条件 | 该客房目前为已入住状态 |
| 后置条件 | 修改该客房状态为空闲 |

表22 数据层userdata模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UserDataService.insert | 语法 | public ResultMessage insert(UserPO user) |
| 前置条件 | 无 |
| 后置条件 | 在数据库中添加一个用户 |
| UserDataService.getUserInfo | 语法 | public PersonalPO getUserInfo(long userID) |
| 前置条件 | 无 |
| 后置条件 | 返回客户信息 |
| UserDataService.update | 语法 | public ResultMessage update(PersonalPO personal) |
| 前置条件 | 在数据库中存在同样ID的PO |
| 后置条件 | 更新用户信息 |
| UserDataService.insert | 语法 | public ResultMessage insert(PersonalPO personal) |
| 前置条件 | 无 |
| 后置条件 | 增加一条用户信息 |
| UserDataService. searchClient | 语法 | public List<UserPO> searchClient(String username, String realName) |
| 前置条件 | 无 |
| 后置条件 | 返回包含相应用户名和真名的用户列表 |
| UserDataService.getCreditBalance | 语法 | public CreditPO getCreditBalance(long userID) |
| 前置条件 | 无 |
| 后置条件 | 返回当前信用值 |

表23 数据creditdata模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CreditDataService.getRecord | 语法 | public List<CreditPO> getRecord(long userID) |
| 前置条件 | 无 |
| 后置条件 | 根据用户编号返回信用记录 |

表24数据层strategydata模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StrategyDataService.searchByHotel | 语法 | public List<StrategyPO > searchByHotel(long hotelID) |
| 前置条件 | 无 |
| 后置条件 | 根据酒店编号返回促销策略列表 |
| StrategyDataService.insert | 语法 | public ResultMessage insert(StrategyPO strategy) |
| 前置条件 | 无 |
| 后置条件 | 插入单一持久化对象 |
| StrategyDataService.delete | 语法 | public ResultMessage delete(long strategyID) |
| 前置条件 | 在数据库中存在同样ID的PO |
| 后置条件 | 删除该促销策略 |
| StrategyDataService.update | 语法 | public ResultMessage update(StrategyPO strategy) |
| 前置条件 | 在数据库中存在同样ID的PO |
| 后置条件 | 更新某条促销策略 |

表25 数据层hotelinfodata模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| HotelInfoDataService.briefHotelInfo | 语法 | public HotelPO briefHotelInfo (long hotelID) |
| 前置条件 | 无 |
| 后置条件 | 返回相应的酒店简要信息 |
| HotelInfoDataService.searchHotelDetails | 语法 | public HotelDetailsPO searchHotelDetails (long hotelID) |
| 前置条件 | 无 |
| 后置条件 | 返回相应的酒店详细信息 |
| HotelInfoDataService.updateRate | 语法 | public HotelPO updateRate(long HotelID, int newRate) |
| 前置条件 | 数据库中存在与hotelID相对应的酒店 |
| 后置条件 | 更新酒店平均评分 |
| HotelInfoDataService. insert | 语法 | public ResultMessage insert(HotelDetailsPO hotel) |
| 前置条件 | 无 |
| 后置条件 | 添加一条酒店信息 |
| HotelInfoDataService.update | 语法 | public ResultMessage update(HotelInfoPO hotelInfo) |
| 前置条件 | 数据库中存在ID相同的HotelInfoPO |
| 后置条件 | 更新酒店信息 |
| HotelInfoDataService.getRoomType | 语法 | public RoomTypePO getRoomType(long hotelID, int roomTypeID) |
| 前置条件 | 无 |
| 后置条件 | 返回相应的客房类型信息 |
| HotelInfoDataService.getRoomTypeList | 语法 | public List<RoomTypePO> getRoomTypeList(long HotelID) |
| 前置条件 | 无 |
| 后置条件 | 返回相应酒店的客房类型列表 |
| HotelInfoDataService.getHotelList | 语法 | Public HotelPO getHotelList(int cityID, int districtID); |
| 前置条件 | 无 |
| 后置条件 | 返回相应城市商圈的酒店列表 |

表26 数据层reviewdata模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ReviewDataService.search | 语法 | public reviewPO search(long hotelID) |
| 前置条件 | 无 |
| 后置条件 | 根据酒店编号返回酒店的评价 |
| ReviewDataService.insert | 语法 | public ResultMessage insert(ReviewPO review) |
| 前置条件 | 无 |
| 后置条件 | 添加一条评价 |

# 6、信息视角

## 6.1 数据持久化对象

系统的PO类就是对应的相关的实体类，在此只做简单的介绍，如表6.1所示：

表27 PO类简单介绍

|  |  |
| --- | --- |
| 类名 | 包含的属性 |
| UserPO | 用户ID，用户类型，用户密码，用户真实名字 |
| PersonalPO | 用户ID，用户密码，用户类型，用户真实名字，会员等级（是否会员），信用值，生日，公司名称 |
| CreditPO | 信用变化类型，信用值变化值，信用值余额，相关订单号 |
| HotelPO | 酒店编号，酒店名称，星级，平均评分，城市名称，商圈名称，最低价格 |
| OrderPO | 订单编号，最晚订单执行时间，房间数量，入住人数，有无儿童，入住人真实姓名 |
| StrategyPO | 策略编号，策略介绍，酒店编号，折扣比例 |
| HotelDetailsPO | 酒店编号，酒店名称，星级，平均评分，酒店地址，城市名称，商圈名称，酒店介绍，服务设施介绍，评论列表 |
| HotelInfoPO | 酒店编号，酒店名称，星级，酒店地址，酒店介绍，服务设施介绍 |
| ReviewPO | 酒店编号，评分，评论内容，用户编号，用户名，评价时间 |
| RoomPO | 客房编号，实际房号，客房类型编号，客房状态 |
| RoomTypePO | 客房类型编号，客房类型名称，客房类型介绍，价格 |
| OrderFilterPO | 订单编号，订单状态，订单生成日期，入住日期，客户姓名，用户名 |

持久化对象如UserPO的定义如下图，更多定义见common.jar中源码。





## 6.2 数据库表

表28 数据库表

|  |  |  |
| --- | --- | --- |
| 表名 | 内容 | 字段设计 |
| client | 客户 |  |
| hotelworker | 酒店工作人员 |  |
| marketer | 网站营销人员 |  |
| administrator | 网站管理人员 |  |
| user | 所有用户 |  |
| orderlist | 订单列表 |  |
| order | 订单 |  |
| hotel | 酒店信息 |  |
| strategylist | 策略列表 |  |
| strategy | 策略 |  |
| availableroom | 可用客房信息 |  |
| Businessdistrict | 商圈 |  |
| city | 城市 |  |
| credit | 信用记录 |  |
| roomtype | 客房类型 |  |
| review | 用户评论 |  |