学 院：南京大学软件学院

团 队：第5组

成 员：陈书玉 陈俐俐 程芸菲 陈悦

完成日期：2016/12/22

酒店预订系统

体系结构设计文档

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **修改日期** | **修改原因** | **版本号** |
| 全体成员 | 2016/10/1 | 逻辑分包 | V0.1 |
| 全体成员 | 2016/10/2 | 物理分包，确定接口 | V0.2 |
| 陈俐俐 | 2016/10/3 | 完成用户界面跳转图 | V0.3 |
| 全体成员 | 2016/10/6 | 完成逻辑层和数据层的接口规范 | V0.4 |
| 全体成员 | 2016/10/7 | 完成文档剩余部分 | V0.5 |
| 陈书玉 | 2016/10/8 | 合并美化文档，并统一接口 | V0.6 |
| 全体成员 | 2016/10/8 | 确定PO，并修改部分接口。User.getUserInfo改成依赖Personnel.getCustomerDetail(String customerID) | V0.7 |
| 陈书玉 | 2016/10/10 | 补充制定酒店策略用例层之间调用的接口的文字与图片部分 | V0.8 |
| 陈书玉 | 2016/10/11 | 完成信息视角的文件格式部分 | V0.9 |
| 陈书玉 | 2016/10/12 | 1. 完成信息视角的数据持久化对象部分 2. 将部分String类改成具体的关键类 | V1.0 |
| 陈书玉 | 2016/10/19 | 在讨论详细设计的时候，回来新增了\_list（）传递的参数为String keyword,新增了数据层接口HotelDao.getBookedHotelList(HotelSearchInfoPO po),把Order中间传递的参数由keyword变为OrderSearchInfoPO po | V1.1 |
| 陈书玉 | 2016/10/20 | 在讨论详细设计的时候，在order模块中新增了接口Order.getHotelDetailInfo和Order.checkAvailableRoomNumber，将获取价格的接口改成Order.next | V1.2 |
| 程芸菲 | 2016/10/20 | 在完成详细设计文档时，修改了部分Personnel模块中的内容，将Personnel.checkUserTelephone与Personnel.checkHotelTelephoen合并，将Personnel部分顾客、酒店工作人员、网站营销人员的查看详细信息、增加、修改与查看各自列表方法合并。 | V1.3 |
| 陈俐俐 | 2016/10/23 | 修改了hotelStrategy和Order模块的前置和后置条件 | V1.4 |
| 陈悦 | 2016/10/24 | 修改了Checkin和AvailableRoom和Credit模块的前置和后置条件 | V1.5 |
| 程芸菲 | 2016/10/24 | 修改了Personnel和Hotel模块的前置和后置条件 | V1.6 |
| 陈书玉 | 2016/10/24 | 修改了User，WebStrategy和Search Hotel的前置和后置条件 | V1.7 |
| 陈俐俐 | 2016/12/18 | 修改了personnel模块和order模块的接口 | V2.0 |
| 程芸菲 | 2016/12/20 | 修改了HotelStrategy和Hotel模块的接口 | V2.1 |
| 陈书玉 | 2016/12/20 | 修改了user，webstrategy和search hotel模块的接口。负责整个文档的更新 | V2.2 |

**目录**

[更新历史 2](#_Toc470044494)

[一、引言 7](#_Toc470044495)

[1.1编制目的 7](#_Toc470044496)

[1.2 词汇表 7](#_Toc470044497)

[1.3参考资料 7](#_Toc470044498)

[二、产品描述 7](#_Toc470044499)

[三、逻辑视角 8](#_Toc470044500)

[四、组合视角 9](#_Toc470044501)

[4.1开发包图 9](#_Toc470044502)

[4.2运行时进程 13](#_Toc470044503)

[4.3物理部署 14](#_Toc470044504)

[五、接口视角 15](#_Toc470044505)

[5.1模块的职责 15](#_Toc470044506)

[5.2用户界面层的分解 17](#_Toc470044507)

[5.2.1用户界面层模块的职责 19](#_Toc470044508)

[5.2.2用户界面模块的接口规范 19](#_Toc470044509)

[5.2.3用户界面模块设计原理 19](#_Toc470044510)

[5.3业务逻辑层的分解 19](#_Toc470044511)

[5.3.1业务逻辑层模块的职责 20](#_Toc470044512)

[5.3.2业务逻辑模块的接口模范 21](#_Toc470044513)

[5.3.2.1 User模块的接口规范 21](#_Toc470044514)

[5.3.2.2 WebStrategy模块的接口规范 22](#_Toc470044515)

[5.3.2.3 HotelStrategy模块的接口规范 23](#_Toc470044516)

[5.3.2.4 Personnel模块的接口规范 24](#_Toc470044517)

[5.3.2.5 SearchHotel模块的接口规范 25](#_Toc470044518)

[5.3.2.6 Checkin模块的接口规范 26](#_Toc470044519)

[5.3.2.7 AvailableRoom模块的接口规范 27](#_Toc470044520)

[5.3.2.8 Credit模块的接口规范 28](#_Toc470044521)

[5.3.2.9 Hotel模块的接口规范 29](#_Toc470044522)

[5.3.2.10 Order模块的接口规范 30](#_Toc470044523)

[5.4.数据层的分解 33](#_Toc470044524)

[5.4.1.数据层模块的职责 33](#_Toc470044525)

[5.4.2.数据层模块的接口规范 34](#_Toc470044526)

[5.4.2.1 UserDao 34](#_Toc470044527)

[5.4.2.2 WebStrategyDao 35](#_Toc470044528)

[5.4.2.3 HotelStrategyDao 35](#_Toc470044529)

[5.4.2.4 PersonnelDao 35](#_Toc470044530)

[5.4.2.5 SearchHotelDao 36](#_Toc470044531)

[5.4.2.6 CheckinDao 37](#_Toc470044532)

[5.4.2.7 AvailableRoomDao 37](#_Toc470044533)

[5.4.2.8 CreditDao 38](#_Toc470044534)

[5.4.2.9 HotelDao 38](#_Toc470044535)

[5.4.2.10 OrderDao 39](#_Toc470044536)

[5.4.2.11 LogDao 40](#_Toc470044537)

[六、信息视角 40](#_Toc470044538)

[6.1数据持久化对象 40](#_Toc470044539)

[6.2文件格式 41](#_Toc470044540)

[6.2.1 user表 41](#_Toc470044541)

[6.2.2 order表 42](#_Toc470044542)

[6.2.3 hotel表 42](#_Toc470044543)

[6.2.4 hotel strategy表 43](#_Toc470044544)

[6.2.5 remark表 43](#_Toc470044545)

[6.2.6 credit表 44](#_Toc470044546)

[6.2.7 checkin表 44](#_Toc470044547)

[6.2.8 availableroom表 45](#_Toc470044548)

# 一、引言

## 1.1编制目的

本报告详细完成对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。

## 1.2 词汇表

|  |  |
| --- | --- |
| **缩写或单词** | **解释** |
| \_ui | 表示某展示层 |
| \_bl | 表示某逻辑层 |
| \_Dao | 表示某数据层 |
| RMI | 表示远程方法调用 |
| \_Str\_ | 表示某具体策略 |

## 1.3参考资料

《酒店预订系统用例文档》

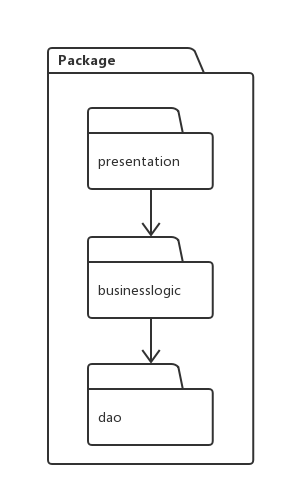
《酒店预订系统需求规格说明文档》

# 二、产品描述

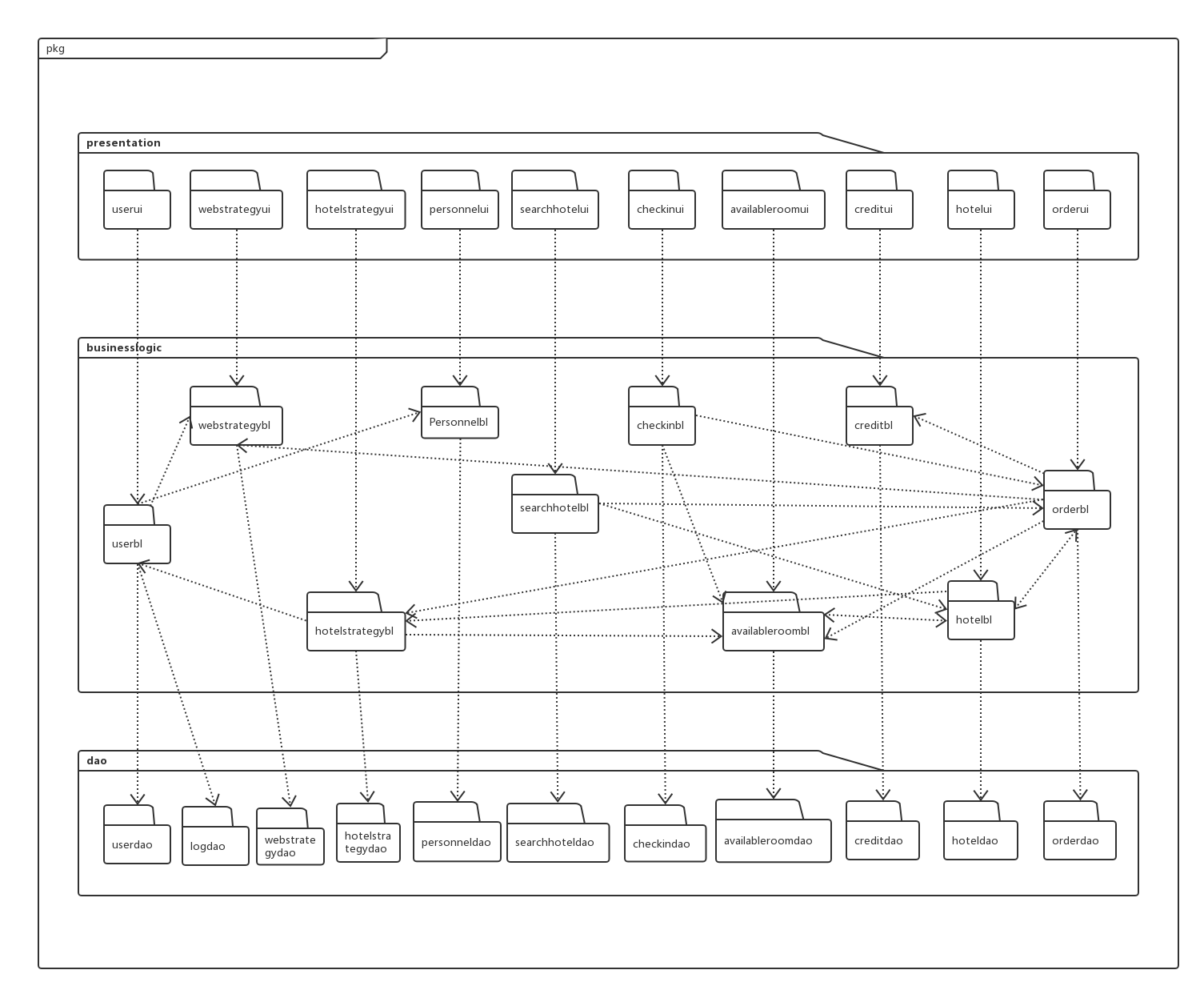
参考《酒店预订系统用例文档》和《酒店预订系统需求规格说明文档》中对产品的概括描述。

# 三、逻辑视角

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**

# 四、组合视角

[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）presentation层与logic层被置于客户端，Dao层被置于服务器端，那么logic层的开发包不可能依赖于Dao层的开发包。使用RMI技术，将Dao层开发包分解为置于客户端的Dao接口包和置于服务器端的Dao层开发包。这样一来，logic层开发包依赖于Dao包，Dao包和Dao层的开发包都依赖于RMI类库包。

2）所有的Dao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.

3）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。

4）此外，presentation层实现时，由mainui包负责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。

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，logic层开发包也依赖于（实现了）logic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Dao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）。

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：

* Order和Hotel:将部分Hotel类抽象接口HotelInfoOrderService置入Order包，这样Hotel单向依赖于Order（实现接口+调用）
* AvailableRoom和Hotel:将部分Hotel类抽象接口HotelInfoAvailService置入AvailableRoom包，这样Hotel单向依赖于AvailableRoom（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。

经过细节改进，最终建立的酒店预订系统开发包设计如表4.1-1，其局部包图如图4.1-1和4.1-2所示：

## 4.1开发包图

酒店预订系统的最终开发包设计如表1所示：

**表1酒店预订系统的最终开发包设计**

|  |  |
| --- | --- |
| **开发（物理）包** | **依赖的其他开发包** |
| loginui | Userui，Personnelui，Orderui，Creditui，Hotelui， AvailableRoomui，Checkinui，HotelStrategyui，WebStrategyui，SearchHotelui ，vo |
| Userui | Userblservice，界面类库包，vo |
| Userblservice |  |
| Userbl | Userblservice ,UserDao,po, WebStrategybl，Personnelbl |
| UserDao | Java RMI,po |
| UserDao | UserDao, po |
| Personnelblservice |  |
| Personnelbl | Personnelblservice, Personnelblservice,po |
| PersonnelDao | Java RMI,po |
| PersonnelDao | UserDao ,po |
| Orderui | Orderblservice，界面类库包，vo |
| Orderblservice |  |
| Orderbl | Orderblservice, OrderDao,po, Availableroombl,Userbl, HotelStrategybl,WebStrategybl, Creditbl |
| OrderDao | Java RMI,po |
| OrderDaoImpl | OrderDao, po |
| Creditui | Creditblservice，界面类库包，vo |
| Creditblservice |  |
| Creditbl | Creditblservice, CreditDao,po,Userbl |
| CreditDao | Java RMI,po |
| CreditDaoImpl | CreditDao, po |
| Hotelui | Hotelblservice，界面类库包，vo |
| Hotelblservice |  |
| Hotelbl | Hotelblservice, HotelDao,po,  HotelStrategybl,Userbl |
| HotelDao | Java RMI,po |
| HotelDaoImpl | HotelDao, po |
| AvailableRoomui | AvailableRoomblservice，界面类库包，vo |
| AvailableRoomblservice |  |
| AvailableRoombl | AvailableRoomblservice, AvailableRoomDao,po, Userbl |
| AvailableRoomDao | Java RMI,po |
| AvailableRoomDaoImpl | AvailableRoomDao, po |
| Checkinui | Checkinblservice，界面类库包，vo |
| Checkinblservice |  |
| Checkinbl | Checkinblservice, CheckiDao,po, Orderbl,Availableroombl,Userbl |
| CheckiDao | Java RMI,po |
| CheckinDaoImpl | CheckiDao, po |
| HotelStrategyui | HotelStrategyblservice，界面类库包，vo |
| HotelStrategyblservice |  |
| HotelStrategybl | HotelStrategyblservice, HotelStrategyDao,po, Userbl,AvailableRoombl |
| HotelStrategyDao | Java RMI,po |
| HotelStrategyDaoImpl | HotelStrategyDao, po |
| WebStrategyui | WebStrategyblservice，界面类库包，vo |
| WebStrategyblservice |  |
| WebStrategybl | WebStrategyblservice, WebStrategyDao,po, |
| WebStrategyDao | Java RMI,po |
| WebStrategyDaoImpl | WebStrategyDao, po |
| SearchHotelui | SearchHotelblservice，界面类库包，vo |
| SearchHotelblservice |  |
| SearchHotelbl | SearchHotelblservice, SearchHotelblservice,po, Hotelbl,Orderbl,Userbl |
| SearchHotelDao | Java RMI,po |
| SearchHotelDaoImpl | SearchHotelDao,po |
| Vo |  |
| Po |  |
| 界面类库包 |  |
| Java RMI |  |
| Datahelper |  |
| Datahelperimpl |  |

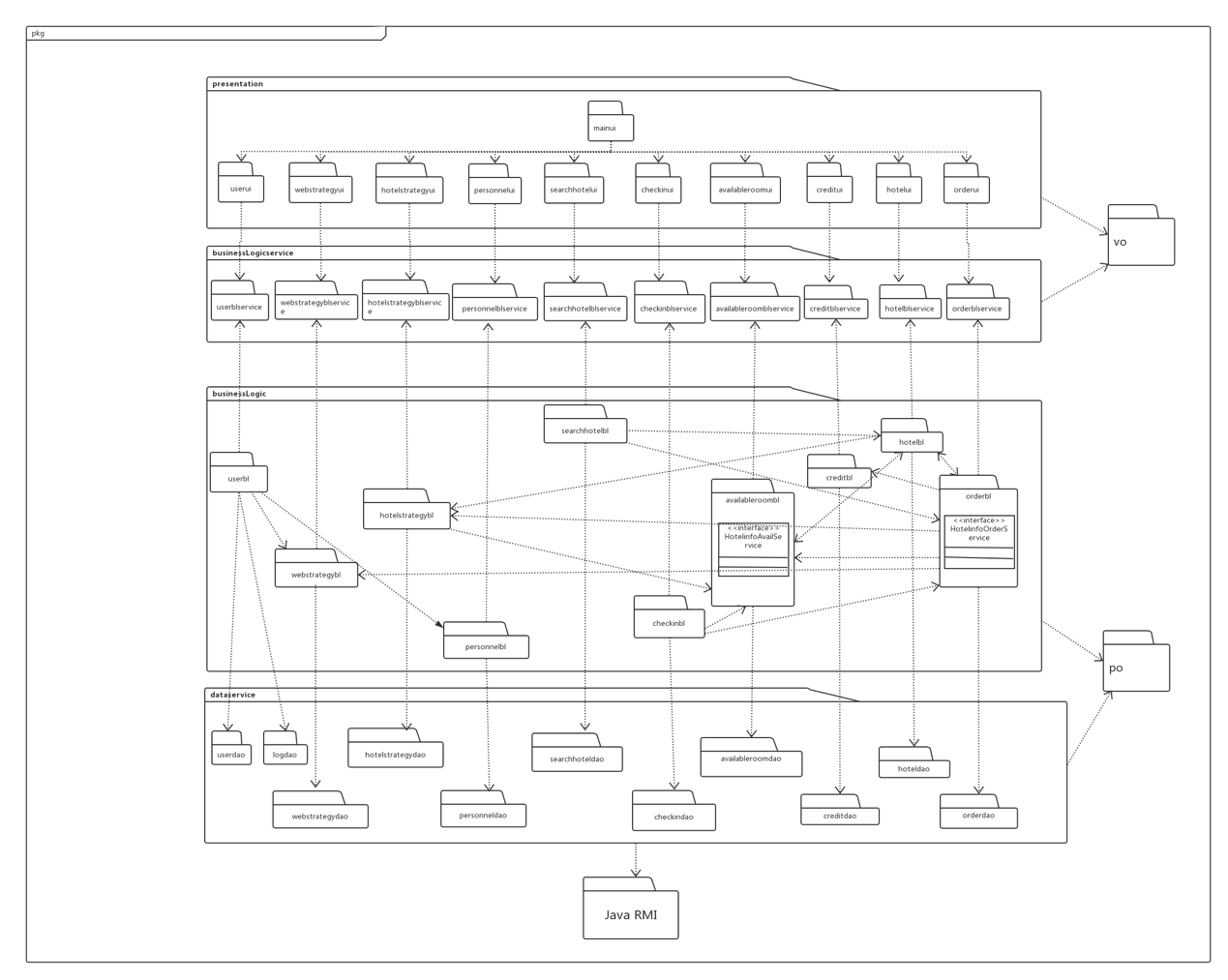


图4.1-1 酒店预订系统客户端开发包图

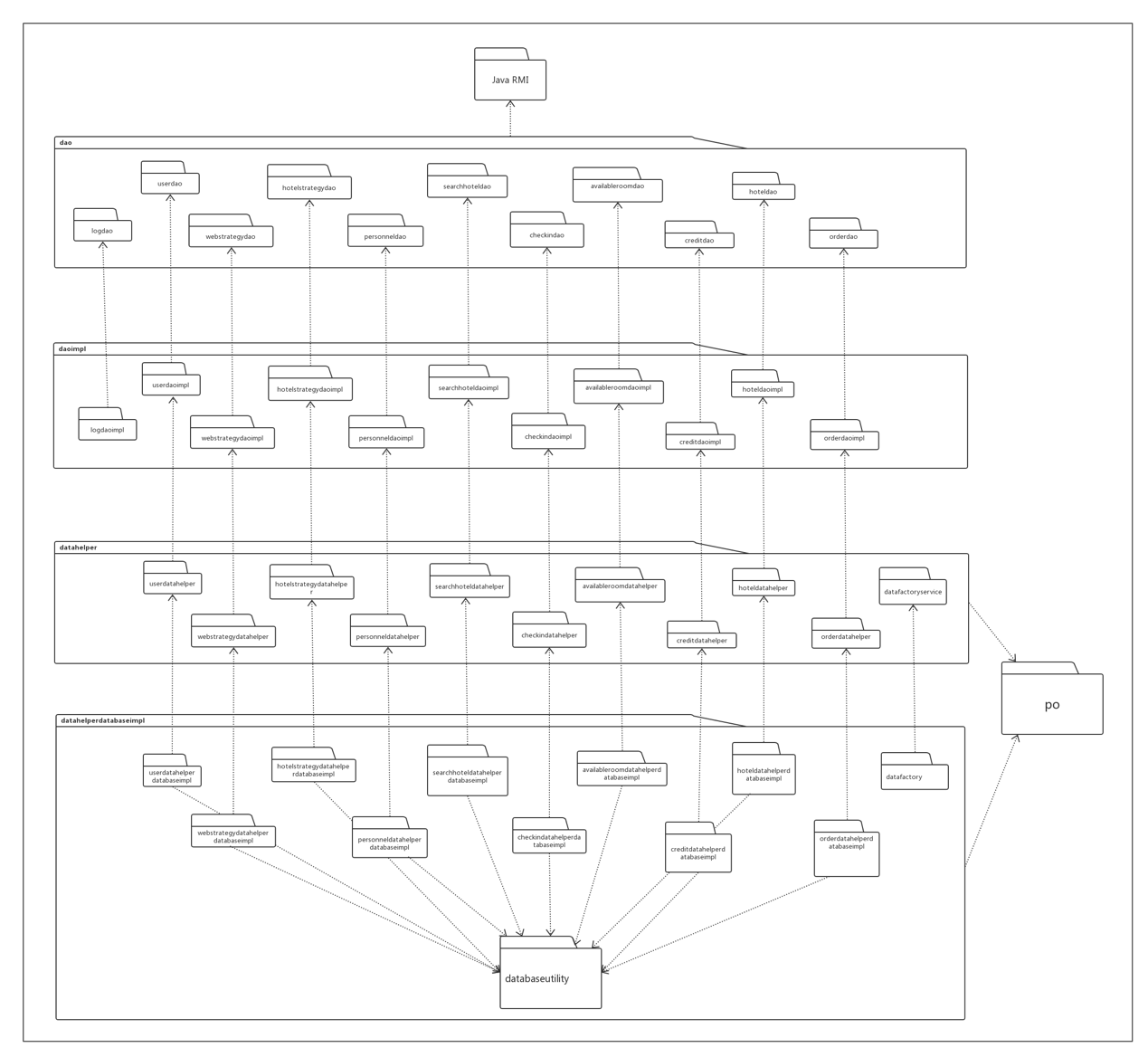


图4.1-2 酒店预订系统服务器端开发包图

## 4.2运行时进程

在酒店预订系统中，会有多个客户端进程和一个服务器端进程，其进程图如图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**

## 4.3物理部署

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。部署图如图6所示。

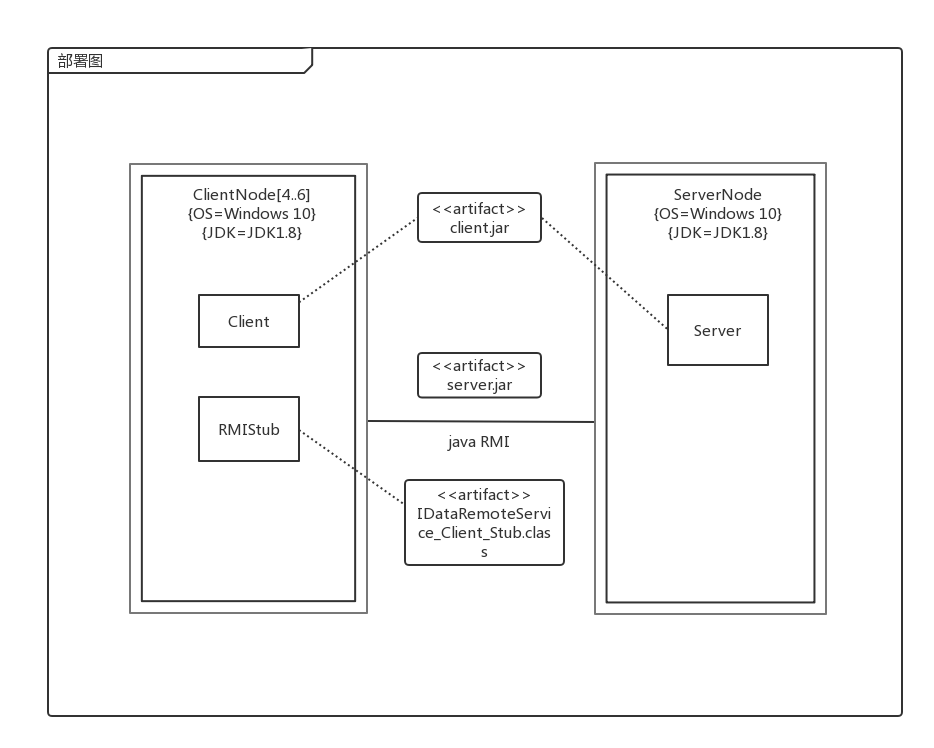


图 6 部署图

# 五、接口视角

## 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端各层的职责分别如表2和表3所示。

**启动模块**

**业务逻辑层**

**展示层**

**网络模块**

**图7 客户端模块视图**

**启动模块**

**网络模块**

**业务逻辑层**

**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 | 基于窗口的酒店预订系统客户端用户界面 |
| 业务逻辑层 | 对于用户界面的输入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4所示。

**表4 层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| **接口** | **服务调用方** | **服务提供方** |
| Userbl  Webstrategybl  Hotelstrategybl  Personnelbl  Searchhotelbl  Checkinbl  Availableroombl  Creditbl  Hotelbl  Orderbl | 客户端展示层 | 客户端业务逻辑层 |
| UserDao  PersonnelDao  OrderDao  CreditDao  HotelDao  AvailableroomDao  CheckinDao  WebstrategyDao  HotelstrategyDao  searchhotelDao  LogDao | 客户端业务逻辑层 | 客户端数据层 |

借用制定酒店促销策略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（需接口），而下层实现这个接口（供接口）。HotelstrategyBLService提供了User界面所需要的所有业务逻辑功能HotelstrategyDao提供了对数据库的增、改、查等操作。这样的实现就大大降低了层与层之间的耦合。

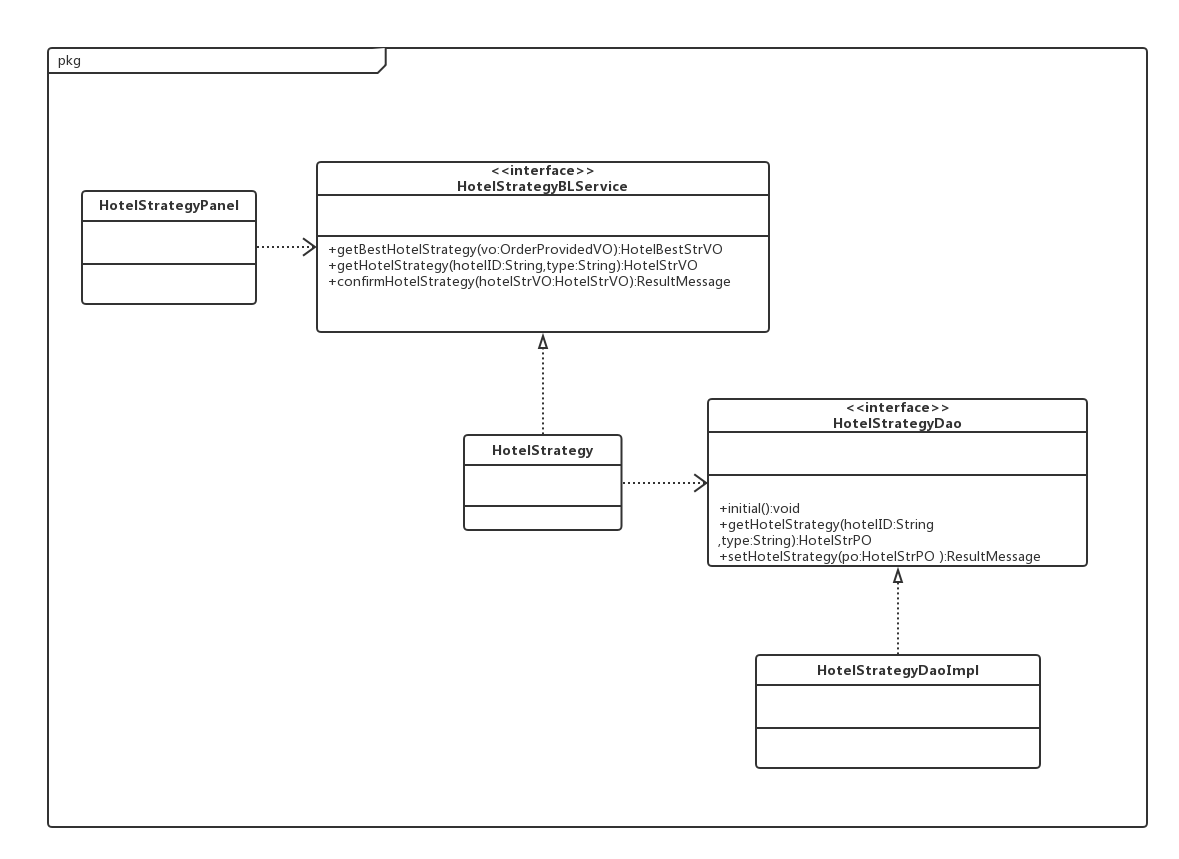


图9制定酒店策略用例层之间调用的接口

## 5.2用户界面层的分解

根据需求，系统存在57个用户界面（加粗的为主界面）：**初始界面**，选择企业或个人注册界面，个人注册界面，企业注册界面，顾客登录界面，**顾客主界面，**修改密码界面，维护个人信息界面，搜索浏览酒店界面，浏览订单列表界面，酒店详细信息界面，生成订单界面，订单详细信息界面，撤销订单界面，评价订单界面，网站管理人员登录界面，**网站管理人员主界面**，选择用户类别界面，网站营销人员列表界面，增加网站营销人员界面，查找网站营销人员界面，修改网站营销人员界面，顾客列表界面，查找顾客界面，修改顾客界面，酒店列表界面，增加酒店界面，增加酒店工作人员界面，查找酒店界面，查看酒店工作人员信息界面，修改酒店工作人员信息界面，网站营销人员登录界面，**网站营销人员主界面**，选择制定策略类型界面，时期阶段折扣界面，会员特定商圈专属折扣策略界面，会员等级折扣策略界面，制定会员等级策略界面，充值信用界面，浏览订单界面，撤销异常订单界面，酒店工作人员登录界面，**酒店工作人员主界面**，维护酒店信息界面，录入客房信息界面，浏览订单列表界面，订单详细信息界面，选择制定策略类型界面，生日特惠策略制定界面，预订三间以上策略制定界面，合作企业策略制定界面，特定节日策略制定界面，线上入住界面，线上退房界面，线下入住界面，线下退房界面，酒店住房记录界面，修改酒店可用房间数界面。

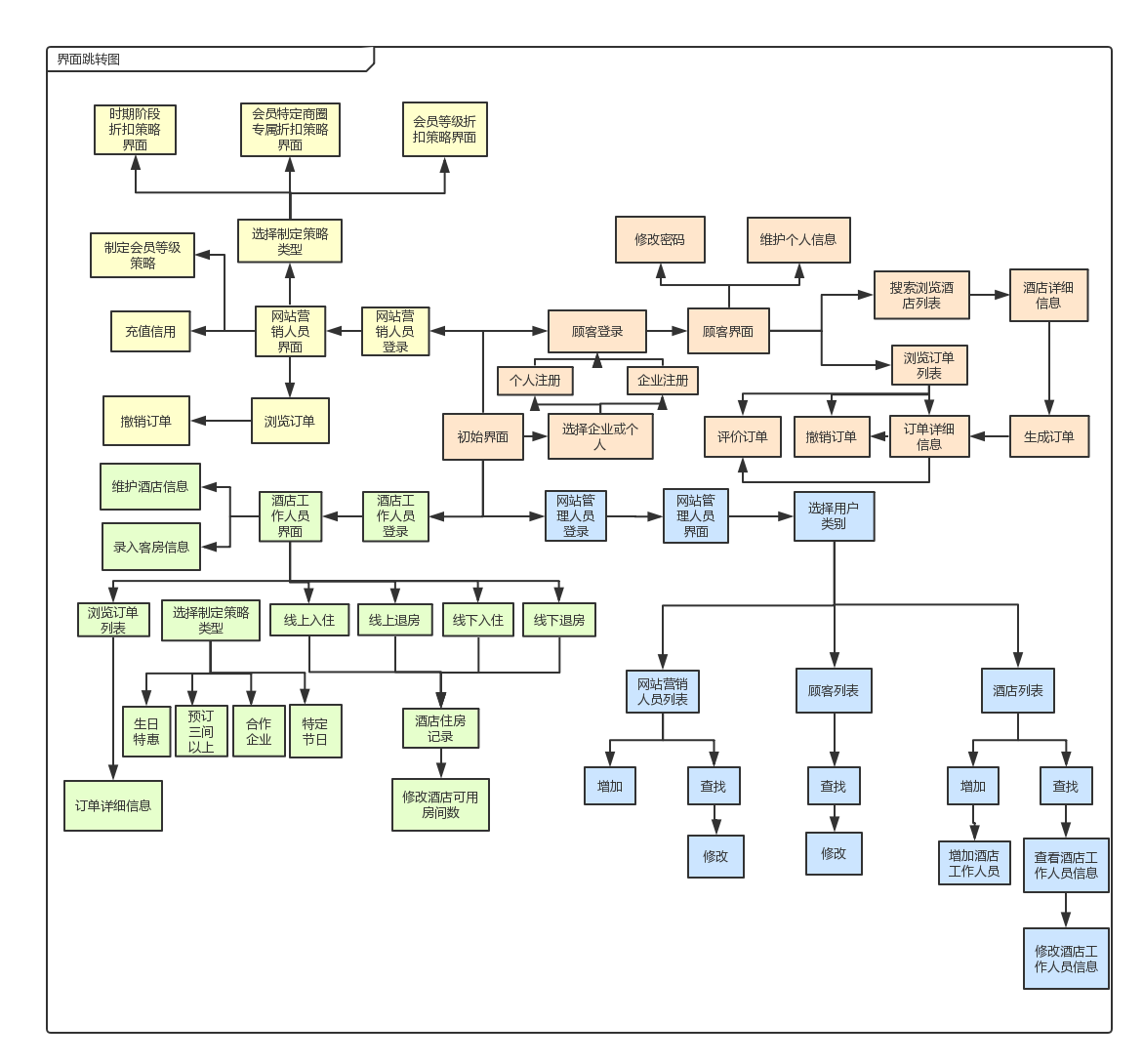


图10酒店预订系统界面跳转图

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

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

### 5.2.2用户界面模块的接口规范

**表6 用户界面层模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| MainFrame | 语法 | init(args:String[]) |
| 前置条件 | 用户触发相应需要显示界面的条件 |
| 后置条件 | 显示Frame并且加载LoginPanel |

**表7 用户界面层模块需要的服务接口**

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务内容 | |
| bl.Personnelservice | 登录界面的业务逻辑接口 | |
| bl.\*blservice | 每个ui模块会有对应的数据逻辑接口 | |

### 5.2.3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。

## 5.3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User对象负责处理登录界面的业务逻辑。业务逻辑层的设计如下图所示：

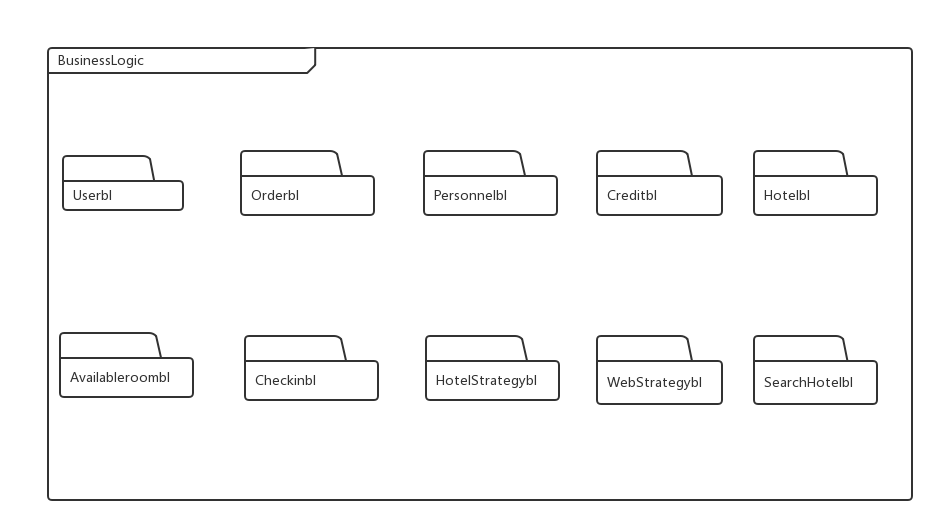


图12 业务逻辑层的设计

### 5.3.1业务逻辑层模块的职责

|  |  |
| --- | --- |
| **模块** | **职责** |
| Userbl | 实现登录与用户账户信息修改的服务，存储当前登录用户身份 |
| Orderbl | 实现生成订单以及对有关订单状态的操作（执行，撤销，评价），浏览查看订单信息（包括搜索） |
| Personnelbl | 实现对用户信息的增改查的管理，注册会员 |
| Creditbl | 实现对顾客信用值的管理 |
| Hotelbl | 实现对酒店基本信息的维护 |
| AvailableRoombl | 实现对可用客房信息的更新，客房信息的录入 |
| Checkinbl | 实现酒店入住退房信息的更新 |
| HotelStrategybl | 实现对酒店营销策略的制定和修改 |
| WebStrategybl | 实现对网站营销策略以及会员等级的制定和修改 |
| SearchHotelbl | 实现对酒店的搜索浏览 |

### 5.3.2业务逻辑模块的接口模范

#### 5.3.2.1 User模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| User.Login | 语法 | Public ResultMessage login(String name ,String password) |
| 前置条件 | 账号、密码符合语法规则 |
| 后置条件 | 无 |
| User. getDetailInfo | 语法 | Public DetailInfoVO getDetailInfo(String userID) |
| 前置条件 | 用户登录 |
| 后置条件 | 无 |
| User. checkTel | 语法 | Public ResultMessage checkTel（String tel） |
| 前置条件 | 无 |
| 后置条件 | 无 |
| User. Confirm | 语法 | Public ResultMessage confirmUserInfo（PersonDetailVO detailInfoVO） |
| 前置条件 | 输入信息格式正确 |
| 后置条件 | 系统更新个人信息 |
| User. checkOldPassword | 语法 | Public ResultMessage checkOldPassword（String password） |
| 前置条件 | 用户应登录 |
| 后置条件 | 无 |
| User. checkNewPassword | 语法 | Public ResultMessage checkNewPassword（String password） |
| 前置条件 | 用户已确认旧密码 |
| 后置条件 | 无 |
| User. confirmPassword | 语法 | Public ResultMessage confirmPassword（PasswordVO passwordVO） |
| 前置条件 | 新密码格式正确 |
| 后置条件 | 若密码一致，更新个人信息。 |
| User. getBasicInfo | 语法 | Public BasicInfoVO getBasicInfo() |
| 前置条件 | 用户已登录 |
| 后置条件 | 无 |
| User. getUserID | 语法 | Public String getUserID (String name) |
| 前置条件 | 用户已登录 |
| 后置条件 | 无 |
| User. getGrade | 语法 | Public int getGrade（String userID） |
| 前置条件 | 用户已登录 |
| 后置条件 | 无 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | | |
| UserDao. getUserPassword | 提供VO返回用户密码信息 | | |
| UserDao. setPassword | 根据提供的VO修改用户信息 | | |
| WebStrategy.getGradeRule | 根据webstr返回顾客会员等级信息 | | |
| Personnel.getPersonDetail | 根据顾客编号返回顾客详细信息 | | |
| Personnel.modifyPerson | 修改顾客个人信息 | | |
| LogDao.logIn | 根据传入的po判断是否用户已经登录过 | | |

#### 5.3.2.2 WebStrategy模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| WebStrategy. getWebBestStrategy | 语法 | Public WebBestStrVO getWebBestStrategy (WebProvidedVO webProvidedVO) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| WebStrategy. getGradeRule | 语法 | Public GradeRuleVO getGradeRule（） |
| 前置条件 | 无 |
| 后置条件 | 无 |
| WebStrategy. getWebStrategy | 语法 | Public WebStrVO getWebStrategy(WebStrategyType type) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| WebStrategy. comfirmGradeRule | 语法 | Public ResultMessage comfirmGradeRule（GradeRuleVO vo） |
| 前置条件 | 输入信息格式正确 |
| 后置条件 | 系统更新会员等级规则 |
| WebStrategy. comfirmWebStrategy | 语法 | Public ResultMessage comfirmWebStrategy（WebStrVO vo） |
| 前置条件 | 输入信息格式正确 |
| 后置条件 | 系统更新网站策略 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | | |
| WebStrategyDao.getGradeRule | 返回会员等级制定规则信息 | | |
| WebStrategyDao.getWebStrategy | 返回网站策略信息 | | |
| WebStrategyDao.setWebStrategy | 系统更新网站策略信息 | | |
| WebStrategyDao.setGradeRuleategy | 系统更新会员等级制定规则信息 | | |

#### 5.3.2.3 HotelStrategy模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| HotelStrategy.getBestHotelStrategy | 语法 | Public HotelBestStrVO getBestHotelStrategy(OrderProvidedVO vo) |
| 前置条件 | 顾客成功下订单 |
| 后置条件 | 无 |
| HotelStrategy.getHotelStrategy | 语法 | Public HotelStrVO getHotelStrategy(String hotelID, HotelStrategyType type) |
| 前置条件 | 酒店制定过优惠策略 |
| 后置条件 | 无 |
| HotelStrategy.confirmHotelStrategy | 语法 | Public ResultMessage confirmHotelStrategy(HotelStrVO hotelStrVO) |
| 前置条件 | 在presentation层已经输入该策略信息并确认 |
| 后置条件 | 系统更新该酒店的策略 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | | |
| HotelStrategyDao.getHotelStrategy | 从数据库中获得该酒店的营销策略 | | |
| HotelStrategyDao. setHotelStrategy | 更改单一化持久对象 | | |
| Availableroom.setBestPrice | 更新酒店房间的最低价格 | | |
| Hotel.getHotelBasicInfo | 获得酒店基本信息 | | |
| Hotel.confirmModifyInfo | 修改酒店基本信息中的合作企业 | | |

#### 5.3.2.4 Personnel模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Personnel.checkUserName | 语法 | Public ResultMessage checkUserName(String userName) |
| 前置条件 | 用户已经输入用户名 |
| 后置条件 | 无 |
| Personnel.checkTel | 语法 | Public ResultMessage checkTel(String Tel) |
| 前置条件 | 用户已经输入联系方式 |
| 后置条件 | 无 |
| Personnel.checkPassword | 语法 | Public ResultMessage checkPassword(String password) |
| 前置条件 | 用户已经输入密码 |
| 后置条件 | 无 |
| Personnel.addPerson | 语法 | Public ResultMessage addPerson(PersonVO personVO) |
| 前置条件 | 顾客已经注册成功或者管理人员已经被识别 |
| 后置条件 | 系统增加用户信息并持久化更新涉及的对象的数据 |
| Personnel.getPersonList | 语法 | Public ArrayList<PersonListVO> getPersonList(String keyWord，UserType useType) |
| 前置条件 | 管理人员已经被识别 |
| 后置条件 | 无 |
| Personnel.getPersonDetail | 语法 | Public PersonDetailVO getPersonDetail(String personID) |
| 前置条件 | 管理人员已经被识别 |
| 后置条件 | 无 |
| Personnel.setPerson | 语法 | Public ResultMessage setPerson (PersonDetailVO personDetailVO) |
| 前置条件 | 输入的信息格式符合规范 |
| 后置条件 | 系统持久化修改并更新涉及的对象的数据 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | | |
| PersonnelDao.addPerson | 添加person单一化持久对象 | | |
| PersonnelDao.getPersonList | 获得person的多个持久化对象 | | |
| PersonnelDao.getPersonDetail | 获得某一person的持久化对象 | | |
| PersonnelDao. setPerson | 更改person单一化持久对象 | | |
| PersonnelDao. checkUserName | 检查用户名是否已存在 | | |

#### 5.3.2.5 SearchHotel模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SearchHotel. getSortedHotelList | 语法 | Public ArrayList<HotelListVO> getSortedHotelList（HotelSearchInfoVO hotelSearchInfoVO） |
| 前置条件 | 用户已登录 |
| 后置条件 | 无 |
| SearchHotel. getCustomerHotelList | 语法 | Public ArrayList<HotelListVO> getCustomerHotelList(String customerID) |
| 前置条件 | 用户已登录 |
| 后置条件 | 无 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | | |
| SearchHotelDao.getHotelList | 获得酒店列表信息 | | |
| SearchHotelDao. getSortedHotelList | 获得排序后的酒店列表信息 | | |
| Order.getOrderList | 获得预订过的订单列表信息 | | |
| Hotel.getHotelDetailInfo | 获得酒店详细信息 | | |
| AvailableRoom.checkAvailableRoomNumber | 检测可用客房数量 | | |

#### 5.3.2.6 Checkin模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Checkin.getOrderInfo | 语法 | public OrderInfoVO getOrderInfo(String orderID) |
| 前置条件 | 订单号符合输入语法要求 |
| 后置条件 | 无 |
| Checkin.getCheckinInfo | 语法 | public CheckinInfoVO getCheckinInfo(String orderID) |
| 前置条件 | 订单号符合输入语法要求 |
| 后置条件 | 无 |
| Checkin.checkIDLength | 语法 | public ResultMessage checkIDLength(String ID) |
| 前置条件 | 输入顾客身份证ID |
| 后置条件 | 无 |
| Checkin.confirmCheckinInfo | 语法 | public ResultMessage confirmCheckinInfo(CheckinInfoVO vo) |
| 前置条件 | 住房信息符合输入语法规范 |
| 后置条件 | 系统持久化新增顾客住房信息并自动更新订单信息。 |
| Checkin.comfirmCheckoutInfo | 语法 | public ResultMessage confirmCheckoutInfo(CheckinInfoVO vo) |
| 前置条件 | 住房信息符合输入语法规范 |
| 后置条件 | 系统持久化修改顾客住房信息并自动更新订单信息和可用客房信息。 |
| Checkin.checkAvailableRoomNumber | 语法 | Public ResultMessage checkAvailableRoomNumber(String hotelID,BedType bedType,String number) |
| 前置条件 | 输入需要变化的可用客房数量 |
| 后置条件 | 无 |
| Checkin.confirmAvailableRoomNumber | 语法 | public ResultMessage confirmAvailableRoomNumber(AvailableRoomNumberVO vo) |
| 前置条件 | 可用客房信息符合输入语法规范 |
| 后置条件 | 系统持久化修改该酒店可用客房信息 |
| Checkin.getAvailableRoomInfo | 语法 | public AvailableRoomInfoVO getAvailableRoomInfo(String hotelID) |
| 前置条件 | 线下入住退房修改可用房间数量 |
| 后置条件 | 无 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | | |
| CheckinDao. getCheckinInfo | 根据订单号获取顾客住房信息 | | |
| CheckinDao. addCheckinInfo | 增加顾客住房信息持久化对象 | | |
| CheckinDao. modifyCheckinInfo | 修改顾客住房信息持久化对象 | | |
| Order.getOrderInfo | 根据订单号获取订单信息 | | |
| Order.setCheckInTime | 根据时间修改订单上实际入住时间的持久化对象 | | |
| Order.setCheckOutTime | 根据时间修改订单上实际退房时间的持久化对象 | | |
| AvailableRoom.setAvailableRoomNumber | 修改可用客房数量的持久化对象 | | |
| AvailableRoom.getAvailableRoomInfo | 根据酒店编号获取可用客房数量 | | |

#### 5.3.2.7 AvailableRoom模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| AvailableRoom. getAvailablerRoomInfo | 语法 | public AvailableRoomInfoVO getAvailableRoomInfo(String hotelID) |
| 前置条件 | 酒店编号符合输入语法要求 |
| 后置条件 | 无 |
| AvailableRoom. confirmAvailableRoomInfo | 语法 | public ResultMessage confirmAvailableRoomInfo (String hotelID ，AvailableRoomInfoVO vo) |
| 前置条件 | 房间信息符合输入语法要求，房间数量为正数 |
| 后置条件 | 系统持久化新增酒店房间信息 |
| AvailableRoom.setAvailableRoomNumber | 语法 | Public ResultMessage setAvailableRoomNumber(String hotelID， AvailableRoomNumberVO vo) |
| 前置条件 | 可用房间数量符合输入语法要求 |
| 后置条件 | 系统持久化更新可用房间信息 |
| AvailableRoom.setBestPrice | 语法 | public ResultMessage setBestPrice (String hotelID,double discount) |
| 前置条件 | 折扣值符合输入语法规范 |
| 后置条件 | 系统持久化更新各种房型的最低价格 |
| AvailableRoom.getRoomPrice | 语法 | public double getRoomPrice (String hotelID,BedType bedType) |
| 前置条件 | 输入的酒店编号和房间类型合法且存在 |
| 后置条件 | 无 |
| AvailableRoom. checkAvailableRoomNumber | 语法 | public ResultMessage checkAvailableRoomNumber (String hotelID ，AvailableRoomNumberVO vo) |
| 前置条件 | 生成订单时，系统自动更新线上可用客房数量 |
| 后置条件 | 无 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | | |
| AvailableRoomDao. getAvailableRoomInfo | 根据酒店编号获取酒店可用客房信息 | | |
| AvailableRoomDao. addAvailableRoomInfo | 增加酒店可用客房信息持久化对象 | | |
| AvailableRoomDao. modifyAvailableRoomInfo | 修改酒店可用客房信息持久化对象 | | |
| AvailableRoomDao. setAvailableRoomNumber | 修改酒店可用客房数量持久化对象 | | |
| AvailableRoomDao. getRoomPrice | 根据酒店编号和房间类型获取房间原始价格 | | |
| AvailableRoomDao.  setBestPrice | 修改可用客房的最低价格 | | |

#### 5.3.2.8 Credit模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Credit. getCreditInfo | 语法 | public CreditInfoVO getCreditInfo () |
| 前置条件 | 顾客编号符合输入语法要求 |
| 后置条件 | 无 |
| Credit. addCredit | 语法 | public ResultMessage addCredit (CreditVO vo) |
| 前置条件 | 无 |
| 后置条件 | 系统持久化新增一条顾客的信用记录并更改顾客的信用值 |
| Credit. cutCredit | 语法 | Public ResultMessage cutCredit (CreditVO vo) |
| 前置条件 | 无 |
| 后置条件 | 系统持久化新增一条顾客的信用记录并更改顾客的信用值 |
| Credit. confirmCreditDeposit | 语法 | public ResultMessage confirmCreditDeposit (double money,String customerId) |
| 前置条件 | 充值金额符合输入语法规范 |
| 后置条件 | 系统持久化新增一条顾客的信用记录并更改顾客的信用值 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | | |
| CreditDao.setCredit | 系统持久化新增一条顾客的信用记录并更改顾客的信用值 | | |
| CreditDao. CreditInfoPO getCreditInfo | 系统返回顾客的信用记录 | | |

#### 5.3.2.9 Hotel模块的接口规范

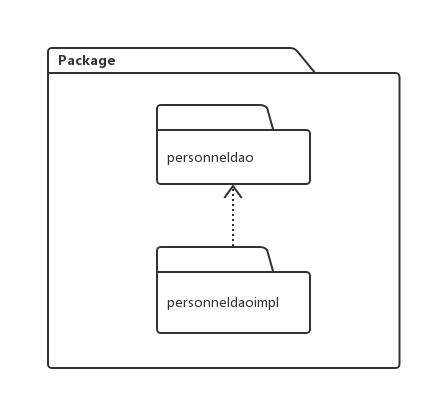
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Hotel.getHotelBasicInfo | 语法 | public HotelBasicInfoVO getHotelBasicInfo(String hotelID) |
| 前置条件 | 酒店工作人员身份已被识别 |
| 后置条件 | 无 |
| Hotel.getHotelDetailInfo | 语法 | public HotelDetailInfoVO getHotelDetailInfo(String hotelID,String customerID) |
| 前置条件 | 顾客已登入 |
| 后置条件 | 无 |
| Hotel.checkTel | 语法 | public ResultMessage checkTel(String telNum) |
| 前置条件 | 联系方式已被输入 |
| 后置条件 | 无 |
| Hotel.confirmModifyInfo | 语法 | public ResultMessage confirmModifyInfo(HotelBasicInfoVO hotelInfovo) |
| 前置条件 | 酒店工作人员身份已经被识别并且联系方式格式符合要求 |
| 后置条件 | 系统更新数据库中的酒店信息 |
| Hotel.getHotelOrderList | 语法 | public ArrayList<HotelOrderVO> getHotelOrder(String hotelID) |
| 前置条件 | 酒店工作人员身份已经被识别 |
| 后置条件 | 无 |
| Hotel.getHotelOrderInfo | 语法 | public HotelOrderInfoVO getHotelOrderInfo(String orderID) |
| 前置条件 | 酒店工作人员身份已经被识别 |
| 后置条件 | 无 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | | |
| HotelDao.getHotelBasicInfo | 获得酒店基础信息 | | |
| HotelDao.setHotelBasicInfo | 更新酒店基础信息 | | |
| HotelDao.addHotelBasicInfo | 初始化酒店基础信息 | | |
| HotelDao.addRemarkInfo | 更新酒店评价信息 | | |
| HotelDao.setBestPrice | 更新酒店的最低价格 | | |
| Availableroom.getAvailableRoomInfo | 获得酒店可用客房信息 | | |
| OrderList.getOrderList | 获得该酒店的订单列表信息 | | |
| SingleOrder.getOrderInfo | 获得该酒店某一订单的详细信息 | | |
| HotelStrategy.getHotelStrategy | 获得酒店优惠策略 | | |

#### 5.3.2.10 Order模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Order.checkUserCredit | 语法 | public ResultMessage checkUserCredit(String CustomerID) |
| 前置条件 | 顾客已选择要预定的酒店 |
| 后置条件 | 若信用值大于0则数据库新增一条订单，若为负，无 |
| Order.checkTelephone | 语法 | public ResultMessage checkTelephone(String telNum) |
| 前置条件 | 顾客已填写联系方式 |
| 后置条件 | 无 |
| Order.checkTime | 语法 | public ResultMessage checkTime(Date time) |
| 前置条件 | 顾客填写预计入住日期 |
| 后置条件 | 无 |
| Order.confirmAddOrder | 语法 | public ResultMessage confirmAddOrder() |
| 前置条件 | 顾客已确认提交订单且系统检查格式无误 |
| 后置条件 | 数据库新增一条订单 |
| Order.getOrderInfo | 语法 | public OrderInfoVO getOrderInfo(String orderID) |
| 前置条件 | orderid符合系统要求 |
| 后置条件 | 无 |
| Order.getOrderList | 语法 | public ArrayList<OrderVO> getOrderList(TypeInfoVO vo) |
| 前置条件 | 用户已选择要浏览的订单类型 |
| 后置条件 | 无 |
| Order.cancelOrderConfirm | 语法 | public ResultMessage cancelOrderConfirm (String orderID) |
| 前置条件 | orderid符合系统要求 |
| 后置条件 | 系统将该订单状态置为已撤销并且更新数据库 |
| Order.remarkOrder | 语法 | public ResultMessage remarkOrder(RemarkVO vo) |
| 前置条件 | 顾客已填写了评价信息 |
| 后置条件 | 更新订单和对应酒店的评价信息 |
| Order.next | 语法 | public StrategyVO next（OrderInfoVO orderinfovo) |
| 前置条件 | 已完成生成订单 |
| 后置条件 | 无 |
| Order.setCheckinTime | 语法 | public ResultMessage setCheckinTime(Date time,String orderID) |
| 前置条件 | 顾客入住 |
| 后置条件 | 更新订单实际入住时间 |
| Order.setCheckoutTime | 语法 | Public ResultMessage setCheckoutTime(Date time,,String orderID) |
| 前置条件 | 顾客退房 |
| 后置条件 | 更新订单实际退房时间 |
| Order.setReturnCredit | 语法 | public ResultMessage setReturnCredit(int creditNum) |
| 前置条件 | 网站营销人员已撤销异常订单 |
| 后置条件 | 增加顾客订单总价全部或二分之一的信用值 |
| Order.getHotelDetailInfo | 语法 | Public HotelDetailInfo getHotelDetailInfo(String hotelID) |
| 前置条件 | 顾客生成订单的时候已选择酒店房间 |
| 后置条件 | 无 |
| Order.checkAvailableRoomNumber | 语法 | public ResultMessage checkAvailableRoomNumber (AvailableRoomNumberVO vo) |
| 前置条件 | 生成订单时，系统已自动更新线上可用客房数量 |
| 后置条件 | 无 |
| 需要的服务（需接口） | | |
| 服务名 | 服务内容 | | |
| OrderDao.getOrderInfo | 获得订单信息 | | |
| OrderDao.addOrder | 向数据库中新增订单 | | |
| OrderDao. getOrderList | 获得订单列表 | | |
| OrderDao. addOrderState | 更新订单状态 | | |
| OrderDao. setOrderRemark | 更新订单的评价信息 | | |
| OrderDao .setCheckintime | 更新订单的实际入住时间 | | |
| OrderDao .setCheckouttime | 更新订单的实际退房时间 | | |
| User.getUserID | 获得该订单所属的顾客信息 | | |
| Availableroom.setAvailableRoomNumber | 更新可用房间数 | | |
| Availableroom.checkAvailableRoomNumber | 检查可用房间数 | | |
| Availableroom.getRoomPrice | 获得酒店房间价格 | | |
| Credit.addCredit | 为顾客增加信用值 | | |
| Credit.cutCredit | 扣除顾客的信用值 | | |
| WebStrategy.getWebBestStrategy | 获得最大折扣的网站优惠策略 | | |
| HotelStrategy.getBestHotelStrategy | 获得最大折扣的酒店优惠策略 | | |

## 5.4.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，PersonnelBusinessService的服务主要有PersonnelDao提供。酒店预定系统 的持久化数据的保存为一种形式：数据库，所示抽象了数据服务。以PersonnelDao为例，数据层模块的具体描述下图所示



**图13 数据层模块的描述**

### 5.4.1.数据层模块的职责

数据层模块职责如下表所示

|  |  |
| --- | --- |
| **模块** | **职责** |
| UserDao | 负责保存用户账户的信息，进行增改查的操作 |
| OrderDao | 负责保存订单信息 |
| PersonnelDao | 负责保存人员机构的信息供读取和修改 |
| CreditDao | 负责保存信用记录 |
| HotelDao | 负责保存酒店信息 |
| AvailableRoomDao | 负责保存酒店可用客房信息 |
| CheckinDao | 负责保存酒店住房信息 |
| HotelStrategyDao | 负责保存酒店营销策略 |
| WebStrategyDao | 负责保存网站营销策略以及会员等级制度信息 |
| SearchHotelDao | 负责酒店列表信息 |
| LogDao | 负责登录登出 |

### 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

#### 5.4.2.1 UserDao

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| UserDao. getUserPassword | 语法 | Public CilentPO getUserPassword（String name） |
| 前置条件 | 数据库存在该用户 |
| 后置条件 | 无 |
| UserDao. setPassword | 语法 | Public ResultMessage setPassword（CilentPO po） |
| 前置条件 | 数据库存在该用户 |
| 后置条件 | 数据库更新该用户密码信息 |

#### 5.4.2.2 WebStrategyDao

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| WebStrategyDao. getGradeRule | 语法 | Public GradeRulePO getGradeRule（） |
| 前置条件 | 无 |
| 后置条件 | 无 |
| WebStrategyDao. getWebStrategy | 语法 | Public WebStrPO getWebStrategy(WebStrategyType type) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| WebStrategyDao. setGradeRule | 语法 | Public ResultMessage setGradeRule （GradeRulePO po） |
| 前置条件 | 信息格式正确 |
| 后置条件 | 数据库更新会员等级规则信息 |
| WebStrategyDao. setWebStrategy | 语法 | Public ResultMessage setWebStrategy（WebStrPO po） |
| 前置条件 | 信息格式正确 |
| 后置条件 | 数据库更新网站策略信息 |

#### 5.4.2.3 HotelStrategyDao

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| HotelStrategyDao. getHotelStrategy | 语法 | Public HotelStrPO getHotelStrategy(String hotelID,HotelStrategyType type) |
| 前置条件 | 数据库中存在该酒店的该策略 |
| 后置条件 | 无 |
| HotelStrategyDao. setHotelStrategy | 语法 | Public ResultMessage setHotelStrategy(HotelStrPO po) |
| 前置条件 | 数据库中存在该酒店策略 |
| 后置条件 | 数据库更新该策略 |

#### 5.4.2.4 PersonnelDao

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| PersonnelDao.addPerson | 语法 | Public ResultMessage addPerson(PersonDetailPO personDetailPO) |
| 前置条件 | 同样的账户在系统中不存在 |
| 后置条件 | 系统永久增加该账户数据 |
| PersonnelDao.getPersonList | 语法 | Public ArrayList<PersonListPO> getPersonList(UserType userType,String username,int userID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| PersonnelDao.getPersonDetail | 语法 | Public PersonDetailPO getPersonDetail(int userID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| PersonnelDao. setPerson | 语法 | Public ResultMessage setPerson (PersonDetailPO personDetailPO) |
| 前置条件 | 该账户存在数据库中 |
| 后置条件 | 修改该账户数据 |
| PersonnelDao. checkUserName | 语法 | public ResultMessage checkUserName(String userName) |
| 前置条件 | 无 |
| 后置条件 | 无 |

#### 5.4.2.5 SearchHotelDao

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| SearchHotelDao.getHotelList | 语法 | Public ArrayList<HotelListPO> getHotelList（HotelSearchInfoPO po） |
| 前置条件 | 无 |
| 后置条件 | 无 |
| SearchHotelDao. getSortedHotelList | 语法 | Public ArrayList<HotelListPO> getSortedHotelList（HotelSortType type） |
| 前置条件 | 无 |
| 后置条件 | 无 |

#### 5.4.2.6 CheckinDao

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| CheckinDao. addCheckinInfo | 语法 | public ResultMessage addCheckinInfo (CheckinInfoPO po) |
| 前置条件 | 无 |
| 后置条件 | 系统新增顾客住房信息CheckinInfoPO |
| CheckinDao. getCheckinInfo | 语法 | public CheckinInfoPO getCheckinInfo (String orderID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| CheckinDao. modifyCheckinInfo | 语法 | public ResultMessage modifyCheckinInfo (CheckinInfoPO po) |
| 前置条件 | 系统中顾客住房信息已存在 |
| 后置条件 | 系统修改顾客住房信息CheckinInfoPO |

#### 5.4.2.7 AvailableRoomDao

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| AvailableRoomDao.getAvailableRoomInfo | 语法 | Public AvailableRoomInfoPO getAvailableRoomInfo (String hotelID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| AvailableRoomDao. addAvailableRoomInfo | 语法 | Public ResultMessage addAvailableRoomInfo (AvailableRoomInfoPO po) |
| 前置条件 | 无 |
| 后置条件 | 系统新增可用客房信息AvailableRoomInfoPO |
| AvailableRoomDao.modifyAvailableRoomInfo | 语法 | Public ResultMessage modifyAvailableRoomInfo (AvailableRoomInfoPO po) |
| 前置条件 | 无 |
| 后置条件 | 系统新增可用客房信息AvailableRoomInfoPO |
| AvailableRoomDao.setAvailableRoomNumber | 语法 | Public ResultMessage setAvailableRoomNumber (AvailableRoomNumberPO po) |
| 前置条件 | 无 |
| 后置条件 | 系统修改可用客房数量信息AvailableRoomInfoPO |
| AvailableRoomDao. getRoomPrice | 语法 | Public double getRoomPrice（String hotelID, BedType bedType） |
| 前置条件 | 无 |
| 后置条件 | 无 |
| AvailableRoomDao. setBestPrice | 语法 | Public ResultMessage setBestPrice(ArrayList<AvailableRoomInfoPO> po) |
| 前置条件 | 无 |
| 后置条件 | 系统修改酒店房间最低价格 |
| AvailableRoomDao.getAvailableRoomNumber | 语法 | public AvailableRoomNumberPO getAvailableRoomNumber(String hotelID, Date date, BedType type) |
| 前置条件 | 无 |
| 后置条件 | 无 |

#### 5.4.2.8 CreditDao

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| CreditDao. setCredit | 语法 | public ResultMessage setCredit (CreditPO po) |
| 前置条件 | 无 |
| 后置条件 | 系统持久化新增一条顾客的信用记录并更改顾客的信用值 |
| CreditDao.  getCreditInfo | 语法 | public CreditInfoPO getCreditInfo(String customerID) |
| 前置条件 | 无 |
| 后置条件 | 无 |

#### 5.4.2.9 HotelDao

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| HotelDao.getHotelBasicInfo | 语法 | public HotelBasicInfoPO getHotelBasicInfo(String hotelID) |
| 前置条件 | 系统中有该酒店 |
| 后置条件 | 无 |
| HotelDao.addRemarkInfo | 语法 | public ResultMessage addRemarkInfo(RemarkPO po) |
| 前置条件 | 系统中有该酒店 |
| 后置条件 | 系统更新该酒店的评价信息 |
| HotelDao.addHotelBasicInfo | 语法 | public ResultMessage setHotelBasicInfo(HotelBasicInfoPO po) |
| 前置条件 | 系统中有该酒店 |
| 后置条件 | 系统增加该酒店的基本信息 |
| HotelDao.set HotelBasicInfo | 语法 | public ResultMessage setHotelBasicInfo(HotelBasicInfoPO po) |
| 前置条件 | 系统中有该酒店 |
| 后置条件 | 系统更新该酒店的基本信息 |
| HotelDao.setBestPrice | 语法 | public ResultMessage setBestPrice(HotelBestPricePO po) |
| 前置条件 | 系统中有该酒店 |
| 后置条件 | 系统更新该酒店的最低价格 |

#### 5.4.2.10 OrderDao

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| OrderDao.getOrderInfo | 语法 | public OrderInfoPO getOrderInfo(String orderID) |
| 前置条件 | 系统中有该订单 |
| 后置条件 | 无 |
| OrderDao.getOrderList | 语法 | public ArrayList<OrderListPO> getOrderList(OrderSearchInfoPO po) |
| 前置条件 | 系统中有该关键字 |
| 后置条件 | 无 |
| OrderDao.addOrderState | 语法 | Public ResultMessage addOrderState(OrderStatePO po) |
| 前置条件 | 系统中有该订单 |
| 后置条件 | 系统更新该订单状态 |
| OrderDao.setOrderRemark | 语法 | public ResultMessage setOrderRemark(RemarkPO po) |
| 前置条件 | 系统中有该订单 |
| 后置条件 | 系统更新该订单评价信息 |
| OrderDao.setCheckintime | 语法 | public ResultMessage setCheckintime(CheckTimePO po) |
| 前置条件 | 系统中有该订单 |
| 后置条件 | 系统更新该订单实际入住时间 |
| OrderDao.setCheckouttime | 语法 | public ResultMessage setCheckouttime(CheckTimePO po) |
| 前置条件 | 系统中有该订单 |
| 后置条件 | 系统更新该订单实际入住时间 |
| OrderDao.addOrder | 语法 | public ResultMessage addOrder(OrderInfoPO po) |
| 前置条件 | 逻辑层已创建一个order对象 |
| 后置条件 | 系统加入该订单 |

#### 5.4.2.11 LogDao

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| LogDao. logIn | 语法 | **Public** ResultMessage logIn(ClientPO po) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| LogDao. logOut | 语法 | **public** ResultMessage logOut(String userName) |
| 前置条件 | 无 |
| 后置条件 | 无 |

# 六、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，如下所示。本系统共包含23个PO类，在此

不一一赘述，只做部分介绍：

GradeRulePO类为会员等级规则，包括信用值。

WebMarketManDetailPO类为网站营销人员信息，包括头像，用户名，联系方式，密码，用户编号，制定过的策略列表等。

HotelWorkerDetailPO类为酒店工作人员细节信息类，包括头像，用户名，联系方式，密码，所在酒店名称，所在酒店编号

持久化对象GradeRulePO类的定义如图14所示：

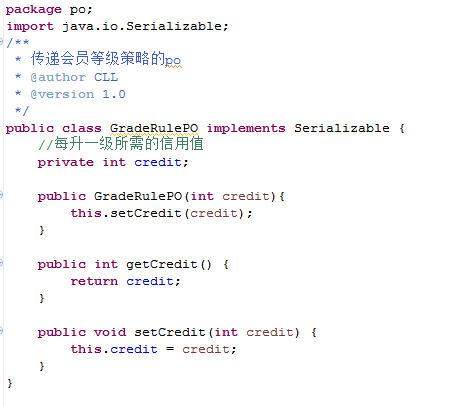


图14 持久化对象GradeRulePO类的定义

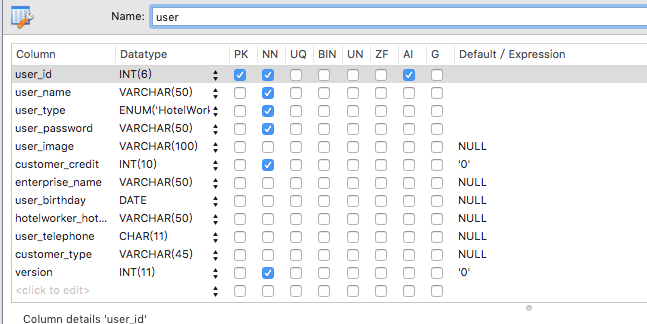
## 6.2文件格式

本系统采用数据库持久化保存一种方式保存文件。

数据库表中包含user表，Order表，Hotel表，Checki表，Hotelstrategy表， Availableroom表， Credit表，remark表。

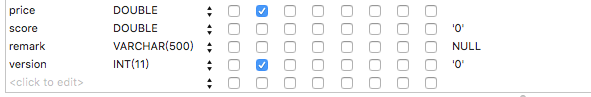
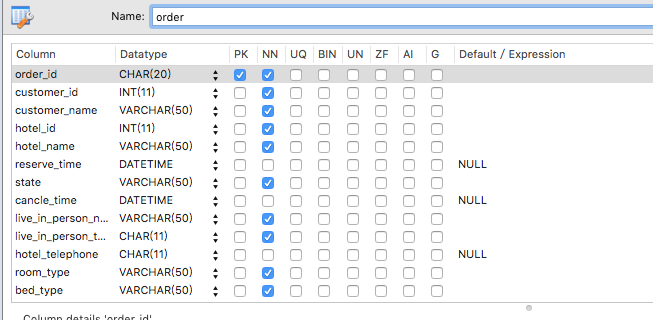
### 6.2.1 user表

user表用于存放用户信息



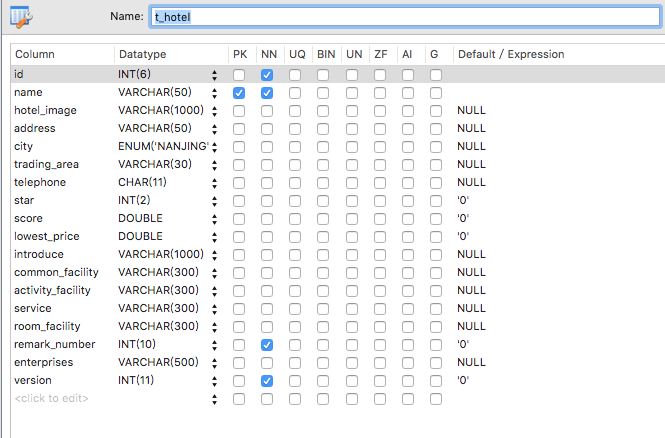
### 6.2.2 order表

order表用于存放订单信息



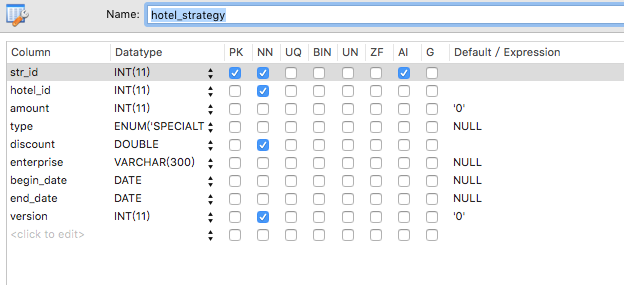
### 6.2.3 hotel表

hotel表用于存放酒店信息



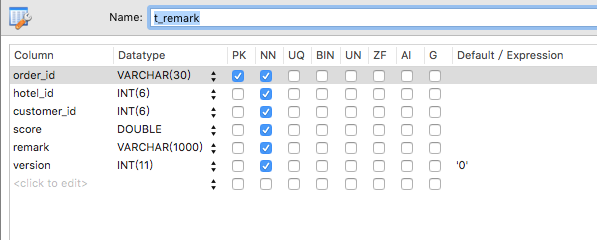
### 6.2.4 hotel strategy表

hotel strategy表 用于存放酒店策略信息



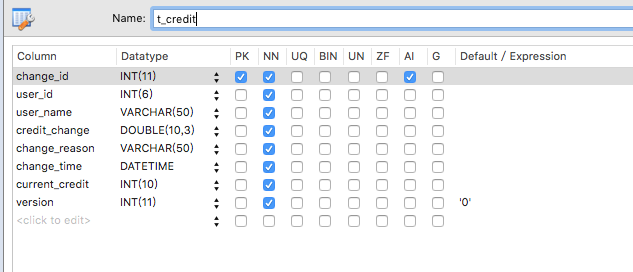
### 6.2.5 remark表

remark表用于存放顾客对酒店的评价信息



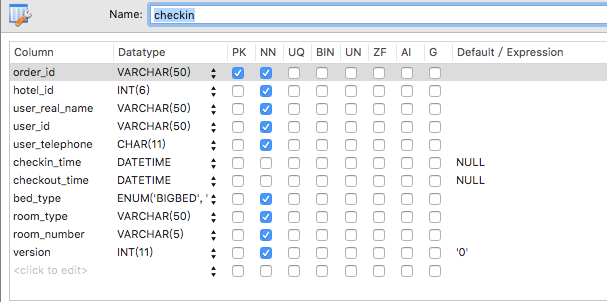
### 6.2.6 credit表

credit表用于存放顾客的信用记录信息



### 6.2.7 checkin表

checkin表用于存放酒店入住信息



### 6.2.8 availableroom表

availableroom表用于存放酒店可用客房信息

