**酒店预订系统**

**体系结构设计文档**

学 院：南京大学软件学院

成 员：王友运卞纯源赵德宇王雪

完成日期：2016年10月10日

**目录**

[更新历史 3](#_Toc464241856)

[1、引言 4](#_Toc464241857)

[1.1 编制目的 4](#_Toc464241858)

[1.2 词汇表 4](#_Toc464241859)

[1.3 参考资料 5](#_Toc464241860)

[2、产品概述 5](#_Toc464241861)

[3、逻辑视角 6](#_Toc464241862)

[4、组合视角 8](#_Toc464241863)

[4.1 开发包图 9](#_Toc464241864)

[4.2 运行时进程 15](#_Toc464241865)

[4.3 物理部署 16](#_Toc464241866)

[5、接口视角 17](#_Toc464241867)

[5.1 模块的职责 17](#_Toc464241868)

[5.2 用户界面层的分解 20](#_Toc464241869)

[5.2.1 用户界面层模块的职责 22](#_Toc464241870)

[5.2.2用户界面层模块的接口规范 22](#_Toc464241871)

[5.2.3 用户界面模块设计原理 25](#_Toc464241872)

[5.3 业务逻辑层的分解 25](#_Toc464241873)

[5.3.1 业务逻辑层模块的职责 25](#_Toc464241874)

[5.3.2 业务逻辑层模块的接口规范 26](#_Toc464241875)

[5.4 数据层的分解 38](#_Toc464241876)

[5.4.1 数据层模块的职责 39](#_Toc464241877)

[5.4.2 数据层模块的接口规范 40](#_Toc464241878)

[6、信息视角 46](#_Toc464241879)

[6.1 数据持久化对象 46](#_Toc464241880)

[6.2 数据库表 52](#_Toc464241881)

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **修改日期** | **修改原因** | **版本号** |
| 全体人员 | 2016-10-11 | 最初草稿（创建文档模板） | V1.0 |
| 王雪 | 2016-12-19 | 根据需求变化修改文档 | V1.1 |

# 1、引言

## 1.1 编制目的

本报表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，同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

## 1.2词汇表

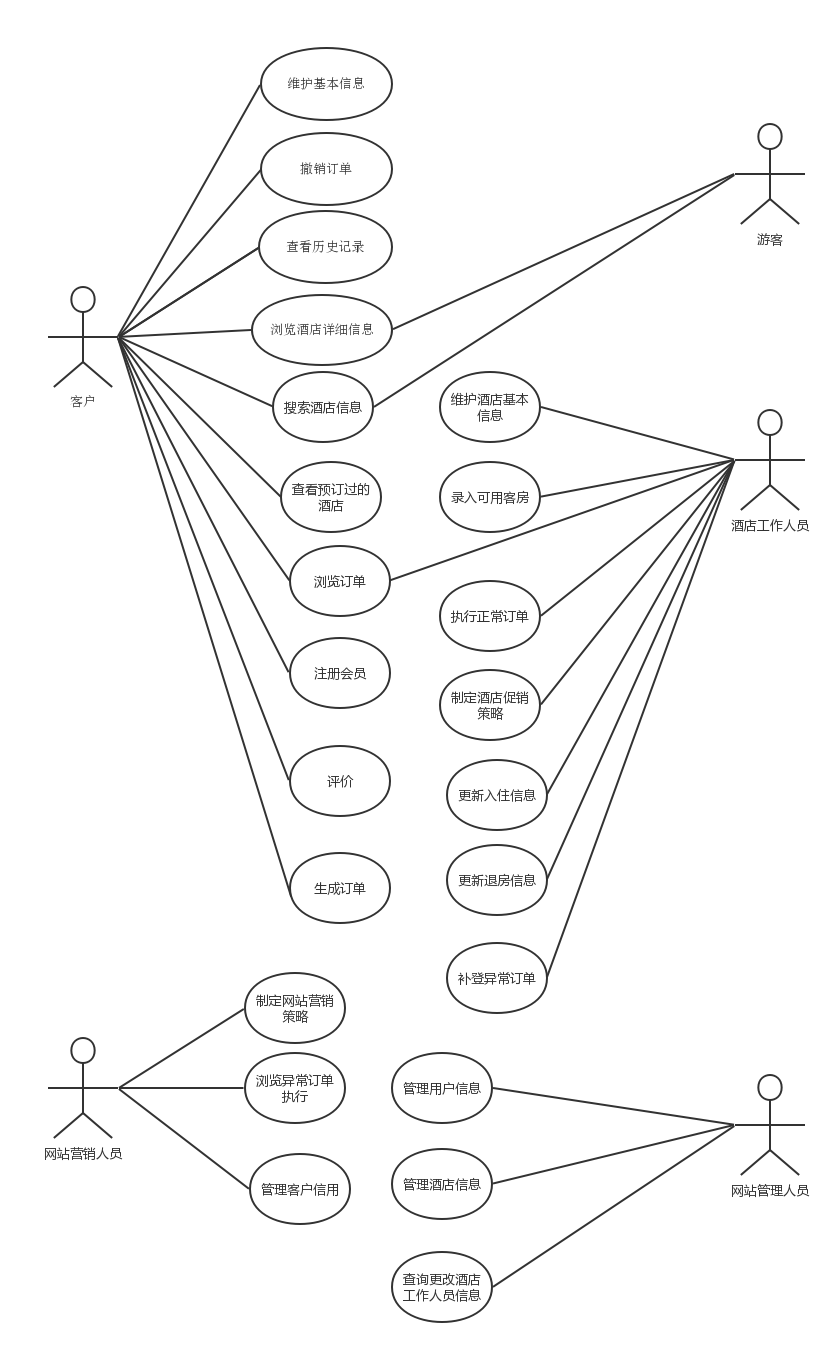
|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| \_ui | 表示某展示层 |  |
| \_bl | 表示某逻辑层 |  |
| \_data | 表示某数据层 |  |
| RMI | 表示远程方法调用 |  |

## 1.3 参考资料

1. IEEE std 1471-2000
2. 丁二玉，刘钦.计算与软件工程（卷二）[M]机械工业出版2012：134—182

# 2、产品概述

参考酒店预订系统用例文档和酒店预订系统软件需求规格说明文档中队产品的概括描述。酒店预订系统主要是应用于网上预定远程酒店订单的在线系统，主要功能见用例图如下。



# 3、逻辑视角

酒店预订系统中，选择了分层体系结构风格，将系统分为3层(展示层、

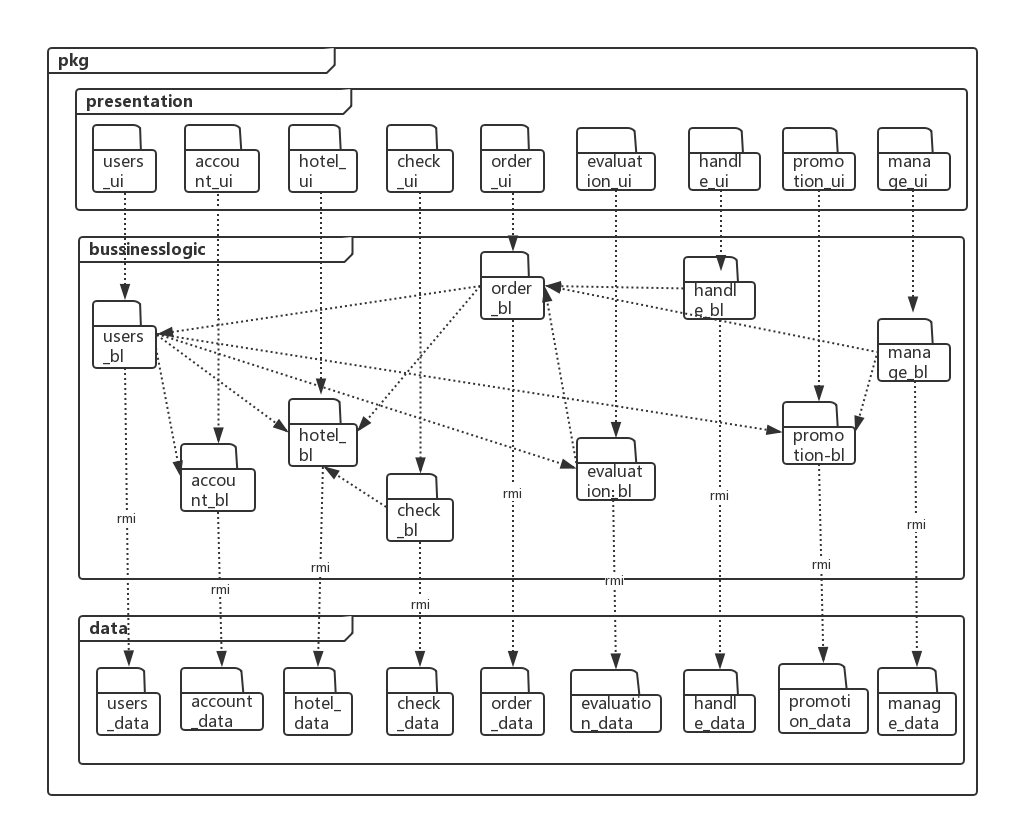
业务逻辑层、数据层)能够很好地示意整个高层抽象。展示层包含GUI页面

的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和

访问。分层体系结构的逻辑视角和逻辑设计方案如图1和图2所示。



图一 参照体系结构风格的包图表达逻辑视角



图二 软件体系结构逻辑设计方案

# 4、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，data层被置于服务器端，那么logic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，logic层开发包依赖于dataservice包，dataservice包和data层的开发包都依赖于RMI类库包。
2. 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databasetility。datavaseutility会依赖于JDBC类库包。
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）。
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）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

表4.1-1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 |
| main\_ui | login\_ui,client\_ui,vistor\_ui,hotelmanager\_ui,webbussiness\_ui,hotellist\_ui,queryorder\_ui,hotelinfo\_ui,VO |
| login\_ui | accout\_blservice,界面类库包,VO |
| client\_ui | credit\_blservice,user\_blservice界面类库包,VO |
| hotelmanager\_ui | room\_blservice界面类库包,VO |
| webmanager\_ui | Promotion\_blservice界面类库包,VO |
| webbussiness\_ui | ordermanagerment\_blservice界面类库包,VO |
| hotellist\_ui | hotel\_blservice界面类库包,VO |
| queryorder\_ui | order\_blservice, evaluation\_blservice |
| hotelinfo\_ui | hotel\_blservice界面类库包,VO |
| credit\_blservice |  |
| credit\_bl | credit\_data\_service,PO |
| user\_blservice |  |
| user\_bl | order\_bl, evaluation\_bl,hotel\_bl  user\_data\_service, PO |
| account\_blservice |  |
| account\_bl | account\_data\_service,PO |
| room\_blservice |  |
| room\_bl | room\_data\_service, PO |
| hotel\_blservice |  |
| hotel\_bl | room\_bl, promotion\_bl,hotel\_data\_service,PO |
| order\_blservice |  |
| order\_bl | credit\_bl, hotel\_bl, evalution\_bl,  promotion\_bl,  order\_data\_service, PO |
| ordermanagement\_blservice |  |
| ordermanagement\_bl | order\_bl,order\_data\_service, PO |
| evaluation\_blservice |  |
| evalution\_bl | evaluation\_data\_service, PO |
| promotion\_blservice |  |
| promotion\_bl | Promotion\_data\_service |
| user\_data | databaseutility,PO |
| user\_data\_serivce | JAVA RMI,PO |
| accout\_data | databaseutility,PO |
| accout\_data\_service | JAVA RMI,PO |
| hotel\_data | databaseutility,PO |
| hotel\_data\_service | JAVA RMI,PO |
| room\_data | databaseutility,PO |
| room\_data\_service | JAVA RMI,PO |
| order\_data | databaseutility,PO |
| order\_data\_  service | JAVA RMI,PO |
| evaluation\_data | databaseutility,PO |
| evaluation\_data\_service | JAVA RMI,PO |
| promotion\_data | databaseutility,PO |
| promotion\_data\_  service | JAVA RMI,PO |
| credit\_data | databaseutility,PO |
| credit\_data\_service | JAVA RMI,PO |
| VO |  |
| PO |  |
| databaseutility | java.JDBC |
| 界面类库包 |  |
| JAVA RMI | java.swing,java.awt,java.2D,java.3D |

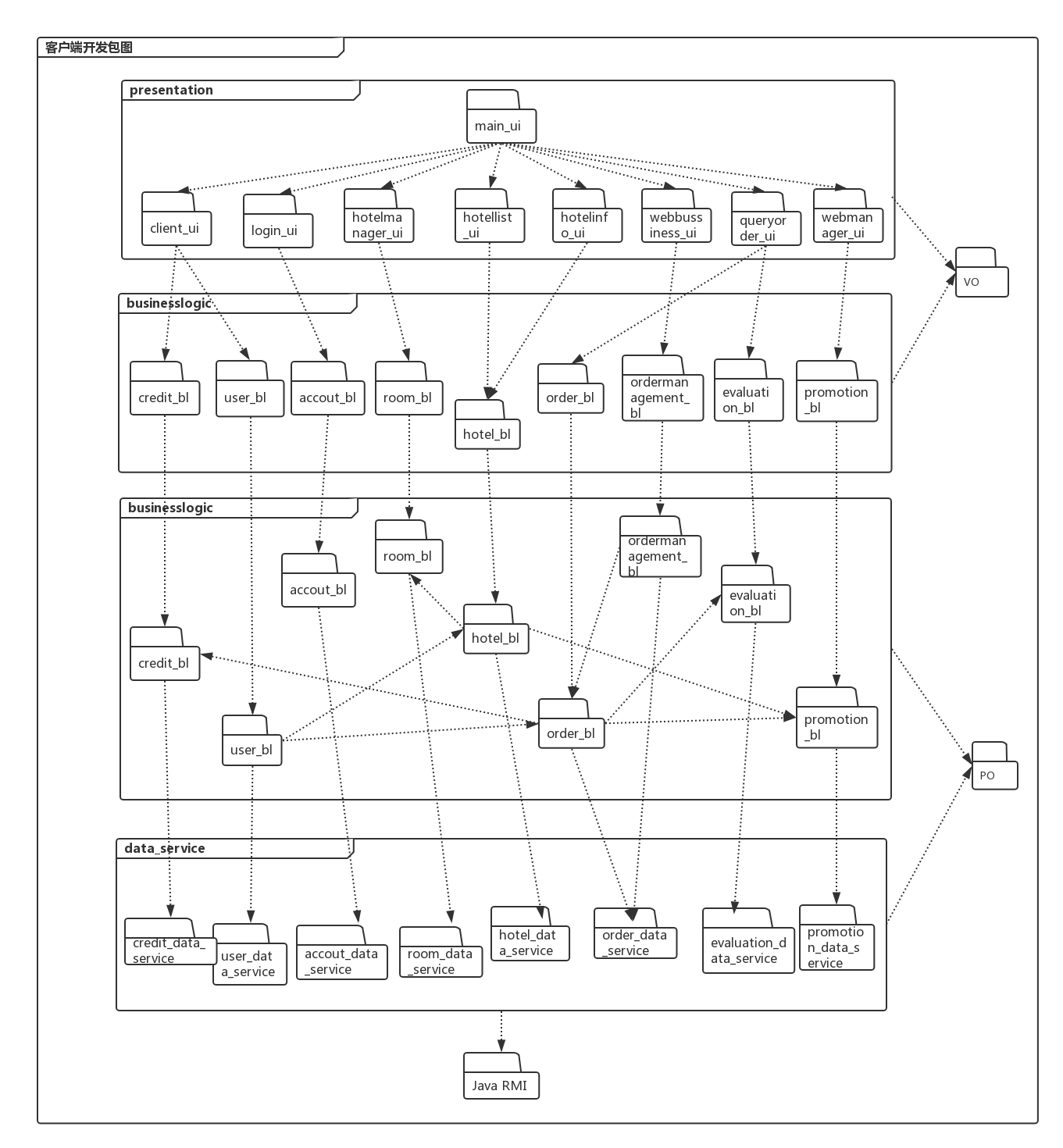


图4.1-1酒店预订系统客户端开发包图

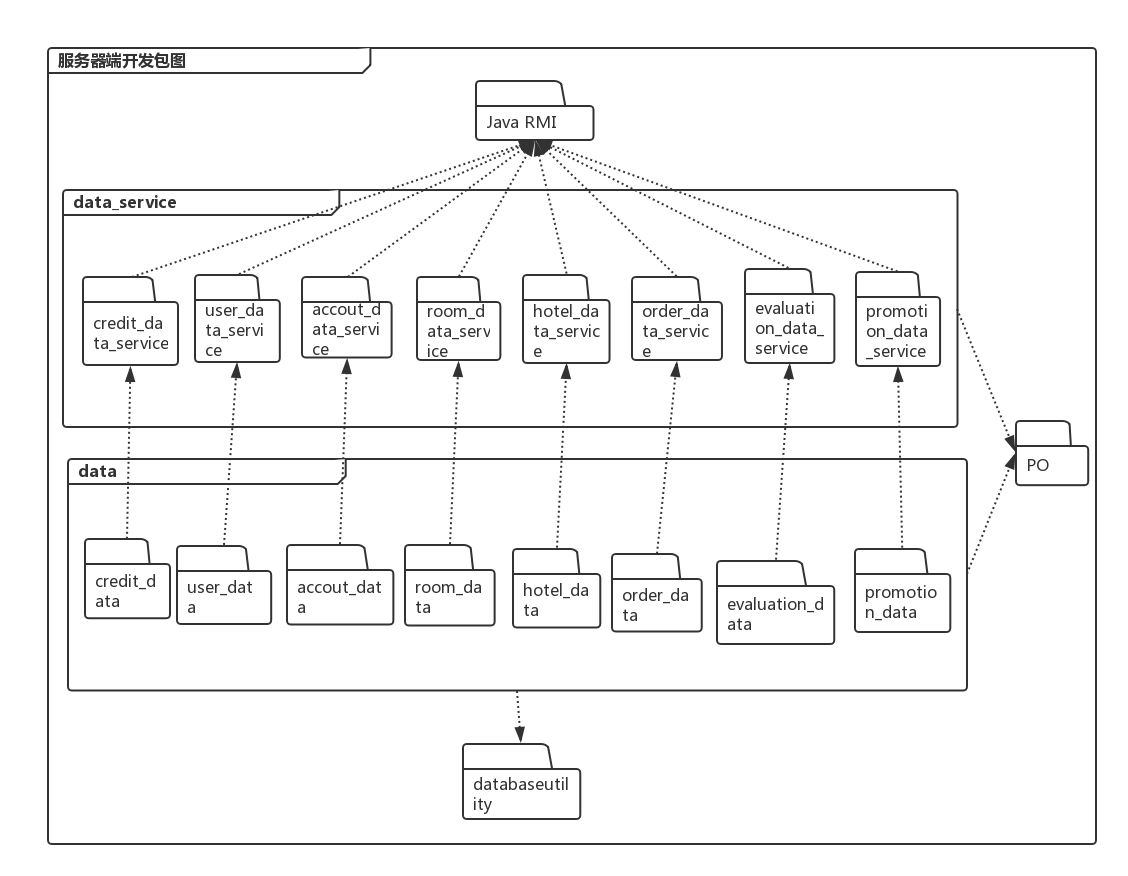


图4.1-2酒店预订系统服务器端开发包图

## 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4.2所示。结合部署图，客户端进程实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4.2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器上，服务器端构件是放在

服务器端机器上。在客户端节点上，还要部署RMIStub构件。由于JavaRMI构件属于JDK6.0的一部分。所以，在系统JDK环境已经设置好的情况下，不需要再独立部署。部署图如图4.3所示。



4.3 部署图

# 5、接口视角

## 5.1模块的职责

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



图5.1-1 客户端模块视图



图5.1-2 服务器端模块视图

表5.1-1 客户端各层的职责

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化网络通信机制，启动用户界面。 |
| 用户界面层 | 基于窗口的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5.1-2 服务器端各层的职责

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化网络通信机制，启动用户界面。 |
| 数据层 | 负责数据的持久化及数据访问接口。 |
| 服务器端网络模块 | 利用Java RMI机制开启RMI服务，注册RMI服务 |

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

表5.1-3 层之间调用的接口

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| credit\_blservice  user\_blservice  accout\_blservice  room\_blservice  hotel\_blservice  order\_blservice  ordermanagement\_blservice  evaluation\_blservice  promotion\_blservice  table\_blservice | 客户端展示层 | 客户端业务逻辑层 |
| credit\_data\_service  user\_data\_service  acccout\_data\_service  hotel\_data\_service  room\_data\_service  order\_data\_service  evaluation\_data\_service  promotion\_data\_service | 客户端业务逻辑层 | 服务器端数据层 |

借用用户管理用例来说明层之间的调用，如图5.1-3所示。每一层之间都是由上层依赖了一个接口（需接口），而下层实现这个接口（供接口）。UserBLService提供了User界面所需要的所有业务逻辑功能

UserDataService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。

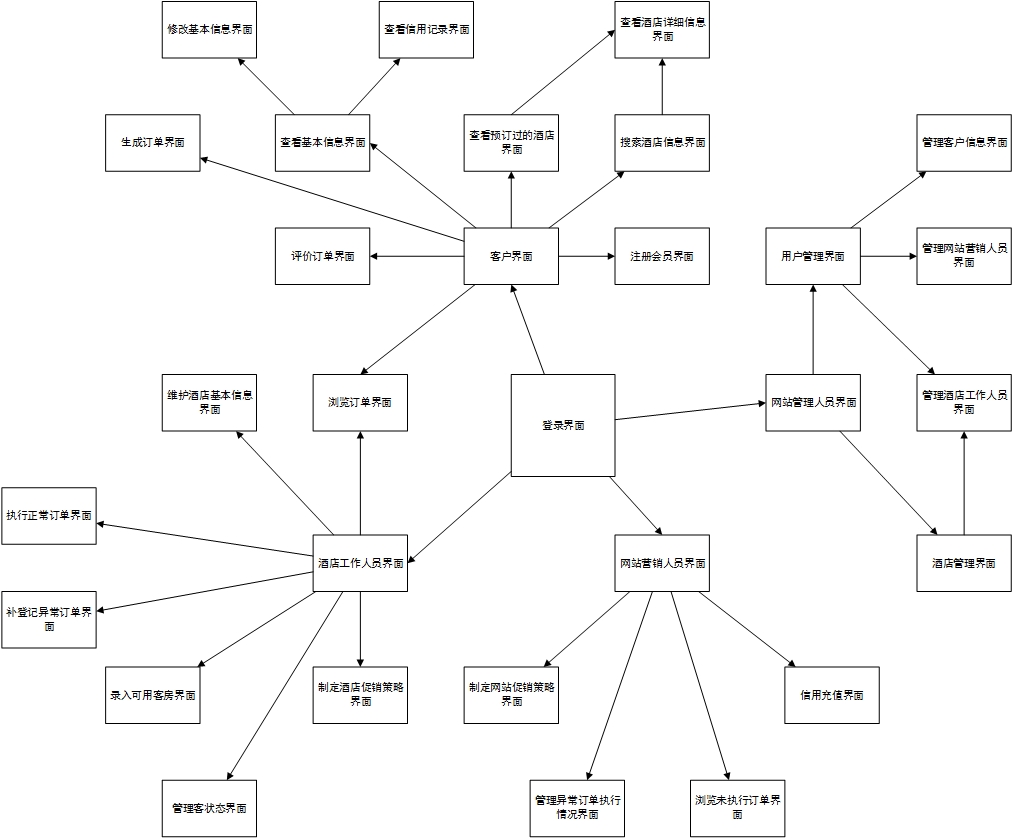


图5.1-3 用户管理用例层之间调用的接口

## 5.2用户界面层的分解

根据需求，系统存在30个用户界面：客户界面，酒店工作人员界面，网站管理人员界面，网站营销人员界面，注册会员界面，搜索酒店信息界面，查看酒店详细信息界面，查看预订过的酒店信息界面，浏览订单界面，生成订单界面，评价订单界面，查看信用记录界面，查看基本信息界面，修改基本信息界面，查看信用记录界面，执行正常订单界面，补登记异常订单界面，维护酒店基本信息界面，录入可用客房信息界面，管理客房状态界面，制定酒店促销策略界面，制定网站促销策略界面，管理异常订单执行情况界面，查看未执行订单界面，信用充值界面，酒店管理界面，用户管理界面，管理客户信息界面，管理酒店工作人员信息界面，管理网站营销人员界面。

界面跳转如图5.2所示



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



图5.2-2 用户界面类

### 5.2.1用户界面层模块的职责

如表5.2.1-1所示为用户界面层模块的职责。

表5.2.1-1 用户界面层模块的职责

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

### 5.2.2用户界面层模块的接口规范

用户界面层模块的接口规范如表5.2.2-1所示。

表5.2.2-1 用户界面层模块的接口规范

|  |  |  |
| --- | --- | --- |
| Main\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示Frame以及LoginPanel |
| Login\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示LoginPanel |
| Webmanager\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示WebManagerPanel |
| Vistor\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示CourierPanel |
| Hotelmanager\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示HotelmanagerPanel |
| Webmanager\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示WebmanagerPanel |
| Webbusiness\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示WebbusinessPanel |
| Hotellist\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示HotellistPanel |
| Queryorder\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示QueryorderPanel |
| Hotelinfo\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示HotelinfoPanel |

用户界面层需要的服务接口如表5.2.2-2所示。

表5.2.2-2 用户界面层模块需要的服务接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| businesslogicservice.credit\_blservice | 提供信用增加和信用扣除 |
| businesslogicservice.user \_blservice | 处理用户信息的接口 |
| businesslogicservice.account\_blservice | 处理登录注册退出登录的接口 |
| businesslogicservice.room\_blservice | 处理酒店客房信息的接口 |
| businesslogicservice.hotel\_blservice | 处理酒店信息的接口（包括搜索酒店和查看酒店） |
| businesslogicservice.order\_blservice | 处理订单信息（浏览、查看订单）提供生成订单的接口 |
| businesslogicservice.ordermanagement\_blservice | 提供处理（包括撤销、执行、补登记）订单接口 |
| Businesslogicservice.evaluation\_blservice | 处理订单评价的接口 |
| businesslogicservice. promotion\_blservice | 处理网站和酒店营销策略的接口 |

### 5.2.3 用户界面模块设计原理

用户界面利用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5.3-1所示。

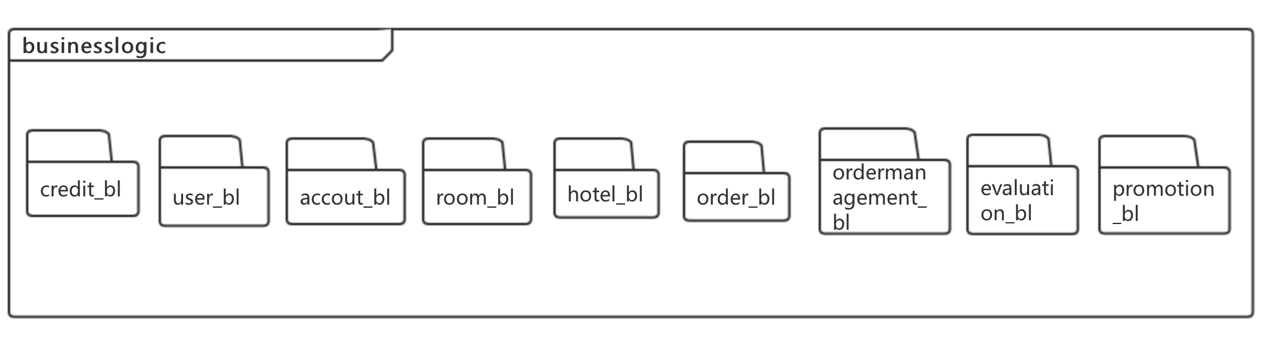


图5.3-1 业务逻辑层

### 5.3.1业务逻辑层模块的职责

业务逻辑层模块的职责如表5.3.1-1所示。

表5.3.1-1 业务逻辑层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| credit\_bl | 提供信用增加和信用扣除 |
| user \_bl | 处理用户信息的接口 |
| account\_bl | 处理登录注册退出登录的接口 |
| room\_bl | 处理酒店客房信息的接口 |
| hotel\_bl | 处理酒店信息的接口（包括搜索酒店和查看酒店） |
| order\_bl | 处理订单信息（浏览、查看订单）提供生成订单的接口 |
| ordermanagement\_bl | 提供处理（包括撤销、执行、补登记）订单接口 |
| evaluation\_bl | 处理订单评价的接口 |
| promotion\_bl | 处理网站和酒店营销策略的接口 |

### 5.3.2业务逻辑层模块的接口规范

表5.3.2-1 credit\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Credit.accountOfUser | 语法 | | public String accountOfUser (int order\_Number , boolean order\_State); |
| 前置条件 | | 已知订单号码和订单状态，需要客户账号 |
| 后置条件 | | 返回客户账号 |
| Credit.numberOfCredit | 语法 | | public CreditVOnumberOfCredit(int order\_Number , boolean order\_State); |
| 前置条件 | | 已知订单号码和订单状态，需要增删的信用值 |
| 后置条件 | | 返回需要增删的信用值 |
| Credit.changCredit | 语法 | | public CreditVO changCredit (CreditVO credit , String userAccount); |
| 前置条件 | | 已知客户账号和需要增删的信用值，需要处理后的信用值 |
| 后置条件 | | 返回当前信用值 |
| Credit.recharge | 语法 | | public CreditVOrecharge(Hashmap user , int money); |
| 前置条件 | | 已知用户账号和充值金额，需要返回当前信用 |
| 后置条件 | | 返回当前信用 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| CreditDataService.add | | 增加新帐号的信用记录 | |
| CreditDataService.modify | | 修改帐号信用信息 | |
| CreditDataService.find | | 查询帐号信用值 | |
| Order\_bl.getOrderInfo | | 获取订单状态 | |

表5.3.2-2 user\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| user.getUserInfo | 语法 | publicUserVO getUserInfo(long id); | |
| 前置条件 | 用户已经登录。  用户选择查看基本信息，调用时提交用户id | |
| 后置条件 | 验证id，返回用户信息 | |
| user.deleteUserInfo | 语法 | publicResultMessage deleteUserInfo(long id, string infoType); | |
|  | 前置条件 | 用户已经登录。  用户删除指定的基本信息，更新时提交删除的信息类型 | |
| 后置条件 | 验证id，返回成功与否和删除完成后信息 | |
| user.updateUserInfo | 语法 | publicResultMessage updateUserInfo(long id, String infoType，String info); | |
| 前置条件 | 用户已经登录。  用户更新指定的基本信息，调用时提交更新的信息 | |
| 后置条件 | 验证id，返回成功与否和更新后的信息表 | |
| user.addUserInfo | 语法 | publicResultMessage addUserInfo(long id, String infoType，String info); | |
| 前置条件 | 用户已经登录。  用户添加指定的基本信息，调用时提交添加的信息 | |
| 后置条件 | 验证id，返回成功与否和添加后的信息表 | |
| user.updateHotelManager | 语法 | publicResultMessage updateHotelManager (HotelManagerVO hotelManager，String hotelId); | |
| 前置条件 | 网站管理人员登录。指定酒店的酒店工作人员需要更改 | |
| 后置条件 | 验证id，更改酒店工作人员信息 | |
| user.changeVIPLevel | 语法 | publicResultMessage changeVIPLevel (int userId); | |
| 前置条件 | 客户会员积分达到要求 | |
| 后置条件 | 验证id，升级会员等级 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| UserDataService. Add | | | 增加一条用户信息 |
| UserDataService. find | | | 查询一条用户信息 |
| UserDataService. modify | | | 修改一条用户信息 |
| UserDataService. Delete | | | 删除一条用户信息 |
| Order\_bl.getOrderList | | | 获取订单列表 |
| Order\_bl.getOrderInfo | | | 查询订单信息 |
| Order\_bl.undoOrder | | | 查询未完成订单信息 |
| Order\_bl.generateOrder | | | 生成订单 |

表5.3.2-3 accout\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Account.register | 语法 | public ResultMessage register(String username , String password1 , String password2) | |
| 前置条件 | 用户注册用户名，密码 | |
| 后置条件 | 返回注册信息（注册成功，失败） | |
| Account.login | 语法 | public ResultMessage login(String username , String password); | |
| 前置条件 | 用户登录，输入账户，密码 | |
| 后置条件 | 返回登录信息（登录成功或密码不符或用户名不存在） | |
| Account.Logout | 语法 | public ResultMessage Logout(String username , String password); | |
| 前置条件 | 用户已经登录，现在退出登录 | |
| 后置条件 | 返回退出登录信息（退出成功） | |
| Account.registerVIP | 语法 | Public UserInfoVO registerVIP(ResultMessage userstate , String username , String VIPtype); | |
| 前置条件 | 用户登录之后，注册会员 | |
| 后置条件 | 返回用户信息（包含会员类型） | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| Account\_data\_service.add | | | 增加一条账户信息 |
| Account\_data\_service.delete | | | 删除一条账户信息 |
| Account\_data\_service.modify | | | 修改一条账户信息 |

表5.3.2-4 room\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| room.getRoomInfo | 语法 | | PublicRoomInfoVOgetRoomInfo(int hotelId, int roomId); |
| 前置条件 | | 利用房间号获取酒店房间信息 |
| 后置条件 | | 返回房间信息（包括房间类型、状态、价格等） |
| room.updateRoomInfo | 语法 | | PublicResultMessageVOupdateRoomInfo(ArrayList<RoomInfoVO>rooms, int hotelId); |
| 前置条件 | | 酒店管理人员登录。客户退房，订单执行完成。或者订单被撤销。 |
| 后置条件 | | 返回成功与否、更新后的酒店房间信息 |
| room.getHotelRooms | 语法 | | Public ArrayList<RoomInfoVO>getHotelRooms(int hotelId); |
| 前置条件 | | 用户登录。获取酒店所有可用客房 |
| 后置条件 | | 返回指定酒店所有可用房间信息（包括房间类型，状态，价格等） |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| Room\_data\_service.returnRoomInfo | | 获取酒店房间信息 | |
| Room\_data\_service.retUpdateRoomInfo | | 更新剩余房间信息 | |
| Room\_data\_service.returnHotelRooms | | 获取酒店剩余房间信息 | |

表5.3.2-5 hotel\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Hotel.searchHotel | 语法 | | Public ArrayList searchHotel(String tradingarea , String level , int pricefloor , int priceceiling) |
| 前置条件 | | 客户按照商圈，酒店级别，价格范围搜索酒店 |
| 后置条件 | | 返回符合要求的酒店列表 |
| Hotel.viewHotel | 语法 | | Public HotelInfoVO viewHotel (String hotelName); |
| 前置条件 | | 用户输入酒店名称查看酒店 |
| 后置条件 | | 返回酒店基本信息（地理位置，价格，简介，评价，实景图片） |
| Hotel.hotelHistory | 语法 | | Public ArrayList hotelHistory(RessultMessage userstate , String username); |
| 前置条件 | | 用户已登录的情况下查看浏览酒店历史记录 |
| 后置条件 | | 返回历史浏览酒店列表 |
| Hotel.addHotel | 语法 | | Public HotelInfoVO addHotel(String hotelid); |
| 前置条件 | | 酒店请求加入网站 |
| 后置条件 | | 返回酒店基本信息（地理位置，价格，简介，评价，实景图片） |
| Hotel.updateHotel | 语法 | | Public HotelInfoVO updateHotel(String hotelid); |
| 前置条件 | | 酒店请求更改酒店信息 |
| 后置条件 | | 返回酒店基本信息（地理位置，价格，简介，评价，实景图片） |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| HotelDataService.add | | 增加新的酒店信息 | |
| HotelDataService.modify | | 修改酒店信息 | |
| HotelDataService.find | | 获取搜索的酒店具体信息 | |
| HotelDataService. Search | | 获取区域内酒店列表 | |
| User\_bl.getUserInfo | | 获取用户基本信息 | |
| Room\_bl.getRoomInfo | | 获取酒店房间信息 | |
| Room\_bl.updateRoomInfo | | 更新剩余房间信息 | |
| Room\_bl.getHotelRooms | | 获取酒店剩余房间信息 | |
| Room\_bl.getRoomInfo | | 判定VIP折扣 | |
| Promotion\_bl.VIPdiscount | | 增加新的酒店信息 | |

表5.3.2-6 order\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| order.getOrderList | 语法 | | Public ArrayList<OrderVO>getOrderList(int userId，inttype); |
| 前置条件 | | 用户已经登录，选择浏览订单（包括已执行，未执行，撤销的订单） |
| 后置条件 | | 返回用户指定类型订单的列表 |
| order.getOrderInfo | 语法 | | Public OrderVOgetOrderInfo (int orderId); |
| 前置条件 | | 用户已经登录，选择查看指定订单详细情况 |
| 后置条件 | | 返回指定订单详细情况 |
| order.undoOrder | 语法 | | Public ResultMessageundoOrder (int orderId); |
| 前置条件 | | 用户已经登录，撤销指定订单 |
| 后置条件 | | 返回撤销结果 |
| order.generateOrder | 语法 | | Public OrderVOgenerateOrder (OrderVO order); |
| 前置条件 | | 用户已经登录，选择酒店下订单。客户信用值大于0 |
| 后置条件 | | 返回生成完成的订单结果 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| OrderDataService.getList | | 获取历史订单列表 | |
| OrderDataService.getOrder | | 获取订单具体信息 | |
| OrderDataService.undoOrder | | 获取未完成订单信息 | |
| OrderDataService.generateOrder | | 生成订单 | |
| Evaluation\_bl.getEvaluationInfo | | 获取评价信息 | |
| Evaluation\_bl.getEvaluations | | 获取评价 | |
| Evaluation\_bl.evaluate | | 生成评价 | |
| Promotion\_bl.judgeVIP | | VIP等级判定 | |
| Promotion\_bl.VIPdiscount | | 判定VIP折扣 | |

表5.3.2-7 ordermanagement\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| orderManagement.undoUnnormalOrder | 语法 | | Public ResultMessage undoUnnormalOrder(int orderId); |
| 前置条件 | | 网站营销人员已经登录，线下的异常订单申诉成功 |
| 后置条件 | | 返回撤销异常订单结果 |
| orderManagement.executeOrder | 语法 | | Public ResultMessage executeOrder (int orderId); |
| 前置条件 | | 客户已经登录，且订单状态正常 |
| 后置条件 | | 返回指定订单执行完成结果 |
| orderManagement.supplyOrder | 语法 | | Public ResultMessagesupplyOrder (int orderId); |
| 前置条件 | | 酒店工作人员已经登录。在订单最晚执行时间后，客户还没有入驻，执行延迟入住 |
| 后置条件 | | 恢复客户信用，更改订单状态 |
| orderManagement. setOrderUnnormal | 语法 | | Public ResultMessagesetOrderUnnormal (int orderId); |
| 前置条件 | | 在订单最晚执行时间后，客户还没有入驻 |
| 后置条件 | | 系统自动设置订单为异常，扣除客户信用 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| Credit\_bl.accountOfUser | | 获取客户的账号 | |
| Credit\_bl.numOfCredit | | 获取订单需要增删掉信用值 | |
| Credit\_bl.changeCredit | | 改变信用值 | |
| Order\_bl.getOrderList | | 按照指定条件获取订单列表 | |
| Order\_bl.getOrderInfo | | 获取指定订单的详细信息 | |
| Order\_bl.undoOrder | | 网站营销人员撤销订单 | |

表5.3.2-8evaluation\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| evaluation.getEvaluationInfo | 语法 | publicevaluationVO getEvaluationInfo (int orderId) | |
| 前置条件 | 用户已登录，且有已评价订单 | |
| 后置条件 | 返回指定订单评价信息 | |
| evaluation.getEvaluations | 语法 | public ArrayList<evaluationVO> getEvaluations (String hotelId) | |
| 前置条件 | 用户已登录，查看酒店评价 | |
| 后置条件 | 返回该酒店所有订单的评价信息 | |
| evaluation.evaluate | 语法 | public ResultMessage evaluate (int orderId) | |
| 前置条件 | 用户已登录，且订单执行完成，提交评价信息 | |
| 后置条件 | 返回对订单的评价结果 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| Evaluation\_data\_service.evaluate | | | 用户在订单执行之后，数据层返回评价信息 |
| Evaluation\_data\_service.delete | | | 数据层提供该酒店所有的评价记录 |
| Evaluation\_data\_service.modify | | | 数据层更改用户对酒店重新的评价 |
| Evaluation\_data\_service.get | | | 数据层返回评价信息 |

表5.3.2-9promotion\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Promotion.justVIP | 语法 | | Public VIPInfoVO justVIP(StateOfLog userstate , String username); |
| 前置条件 | | 用户在登录情况下，下订单会员的折扣 |
| 后置条件 | | 返回用户会员等级及会员类别 |
| Promotion.VIPdiscount | 语法 | | Public double VIPdiscount(StateOfLog userstate , String username , String tradingarea , Date date , int number); |
| 前置条件 | | 用户开始下订单 |
| 后置条件 | | 返回用户最终的折扣 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| Promotion\_data\_service.returnVIPLevel | | 返回客户VIP等级 | |
| Promotion\_data\_service.returnVIPdiscount | | 返回VIP折扣 | |

## 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5.4 数据层模块的描述

### 5.4.1数据层模块的职责

数据层模块的职责如表5.4.1所示。

表5.4.1 数据层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| DataService | 持久化数据库的接口，提供集体载入，集体保存、增、删、改、查服务。 |
| 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5.4.2-1数据层credit\_data模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CreditdataService.add | 语法 | public boolean add(CreditPO credit); |
| 前置条件 | 已知客户id下无信用值记录 |
| 后置条件 | 在数据文件中增加一个信用值记录 |
| CreditdataService.modify | 语法 | public ResultMessage modify(CreditPO credit); |
| 前置条件 | 逻辑层请求修改账户的信用值 |
| 后置条件 | 数据库中修改信用值 |
| CreditdataService.find | 语法 | public ResultMessage find (CreditPO credit); |
| 前置条件 | 逻辑层请求查看账户信用值 |
| 后置条件 | 返回当前客户的信用值 |

表5.4.2-2数据层user\_data模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UserDataService.add | 语法 | publicResultMessaage add(UserPO po) |
| 前置条件 | 同样id的po在数据文件中不存在 |
| 后置条件 | 在数据文件中增加一个po记录 |
| UserDataService. find | 语法 | public UserPO find(Stringid) |
| 前置条件 | 相同id的po在数据文件中已存在 |
| 后置条件 | 返回相应的UserPO结果 |
| UserDataService. modify | 语法 | public ResultMessage  modify(UserPO po) |
| 前置条件 | 相同id的po在数据文件中已存在 |
| 后置条件 | 更新一个po |
| UserDataService. delete | 语法 | public ResultMessage  delete(String user\_id) |
| 前置条件 | 相同id的po在数据文件中已存在 |
| 后置条件 | 删除一个po记录 |

表5.4.2-3数据层account\_data模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Account\_data.add | 语法 | Public Boolean add(AccountPO po) |
| 前置条件 | 逻辑层请求增加一个账户 |
| 后置条件 | 数据库增加账户 |
| Account\_data.delete | 语法 | Public Boolean delete(AccountPO po) |
| 前置条件 | 逻辑层请求删除一个账户 |
| 后置条件 | 数据库删除账户 |
| Account\_data.modify | 语法 | public Boolean modify(AccountPO po) |
| 前置条件 | 逻辑层请求修改一个账户 |
| 后置条件 | 数据库修改账户 |

表5.4.2-4数据层room\_data模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Room.returnRoomInfo | 语法 | public RoomInfoPO returnRoomInfo(int hotelId, int roomId); |
| 前置条件 | 逻辑层请求调用酒店某一客房信息 |
| 后置条件 | 数据层返回客房信息 |
| Room.retUpdateRoomInfo | 语法 | Public ResultMessage retupdateRoomInfo(ArrayList<RoomInfoVO>rooms, int hotelId); |
| 前置条件 | 酒店工作人员更新客房信息 |
| 后置条件 | 数据层返回更新后客房信息 |
| Room.returnHotelRooms | 语法 | Public ArrayList<RoomInfoVO>returnHotelRooms(int hotelId); |
| 前置条件 | 逻辑层请求调用酒店可用客房 |
| 后置条件 | 数据层返回酒店可用客房列表 |

表5.4.2-5数据层hotel\_data模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| HotelDataService.add | 语法 | public boolean  add(hotelPO po) |
| 前置条件 | 逻辑层请求增加一家酒店 |
| 后置条件 | 在数据文件中增加一个hotel记录 |
| HotelDataService.modify | 语法 | public ResultMessage  modify (hotelPO po) |
| 前置条件 | 逻辑层请求修改一家酒店信息 |
| 后置条件 | 在数据文件中修改该酒店信息 |
| HotelDataService.find | 语法 | public ResultMessage  find(hotelPO po) |
| 前置条件 | 逻辑层请求符合条件的酒店信息 |
| 后置条件 | 返回具体酒店信息 |
| ManageDataService. Search | 语法 | public ArrayList  search(String tradingarea , String level , int pricefloor , int priceceiling) |
| 前置条件 | 逻辑层请求符合条件的酒店列表 |
| 后置条件 | 返回符合条件的酒店列表 |

表5.4.2-6数据层order\_data模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderDataService.getList | 语法 | publicArrayList<OrderPO>  getOrderList(int[] orderId) |
| 前置条件 | orderId的po在数据文件中存在 |
| 后置条件 | 返回order表 |
| OrderDataService.getOrder | 语法 | publicOrderPO  getOrder(int orderId) |
| 前置条件 | orderId的po在数据文件中存在 |
| 后置条件 | 返回相应的orderPO结果 |
| OrderDataService.  undoOrder | 语法 | public ResultMessage  undoOrder(int orderId) |
| 前置条件 | 相同id的po在数据文件中已存在 |
| 后置条件 | 更新一个po |
| OrderDataService. generateOrder | 语法 | publicOrderPO  generateOrder(OrderPO order) |
| 前置条件 | 该订单信息在数据文件中不存在 |
| 后置条件 | 添加一个orderpo |

表5.4.2-7数据层evaluation\_data模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Evaluation\_data.evaluate | 语法 | Public Boolean add(EvaluationPO po) |
| 前置条件 | 逻辑层请求增加一条评价 |
| 后置条件 | 数据层增加这条评价 |
| Evaluation\_data.delete | 语法 | Public Boolean delete(EvaluationPO po) |
| 前置条件 | 逻辑层请求删除一条评价 |
| 后置条件 | 数据层删除这条评价 |
| Evaluation\_data.modify | 语法 | Public Boolean modify(EvalutionPO po) |
| 前置条件 | 逻辑层请求修改一条评价 |
| 后置条件 | 数据层修改评价 |
| Evaluation\_data.get | 语法 | Public EvaluationPO get(int order\_id) |
| 前置条件 | 逻辑层请求获得一个订单的评价 |
| 后置条件 | 数据层返回评价 |

表5.4.2-8数据层promotion\_data模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Promotion.returnVIPLevel | 语法 | Public VIPInfoPO returnVIPLevel(String username , String password); |
| 前置条件 | 逻辑层请求调用用户的会员等级 |
| 后置条件 | 数据层返回用户的会员等级 |
| Promotion.returnVIPdiscount | 语法 | Public double returnVIPdiscount(String username , String tradingarea , Date date , int number); |
| 前置条件 | 逻辑层请求数据层返回最终订单折扣 |
| 后置条件 | 数据层返回订单最终的折扣 |

# 6、信息视角

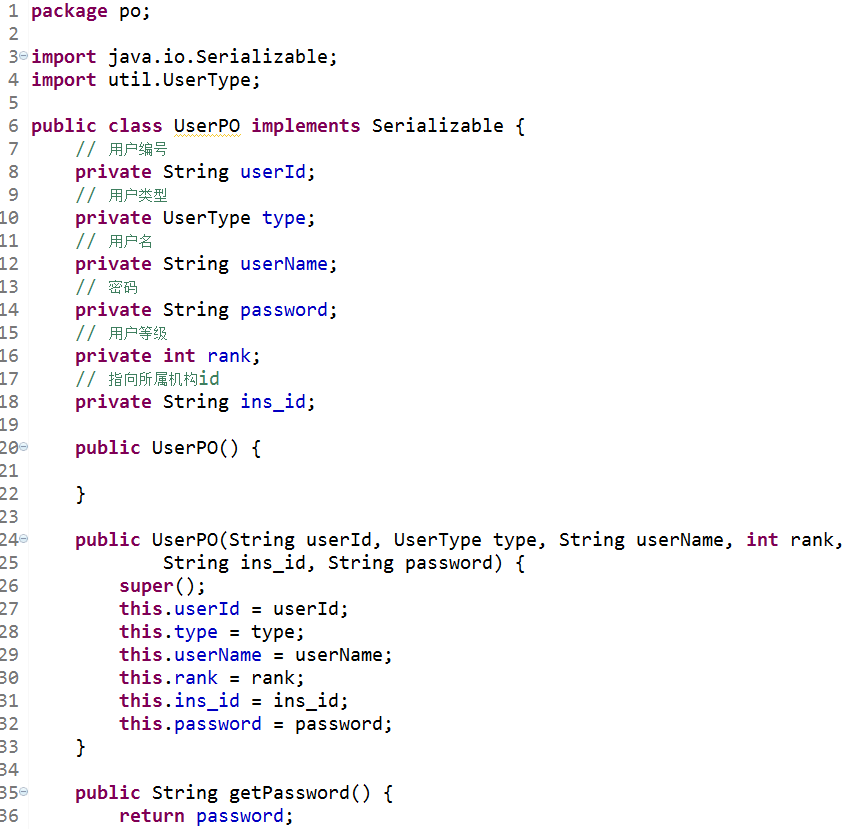
## 6.1 数据持久化对象

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6.1 PO类简单介绍

|  |  |
| --- | --- |
| 类名 | 包含的属性 |
| CreditPO | 信用值；信用记录：充值记录、完成订单信用增加记录、未完成订单扣除信用记录、未完成订单处理恢复信用记录 |
| ClientPO | 客户的信息：名称，联系方式，信用，订单, 预订过的酒店， |
| OrderPO | 订单的信息：订单号， 开始时间，酒店，房间类型及数量，退房时间，最晚订单执行时间，状态，预计入住人数，有无儿童 |
| AccountPO | 账户信息：用户种类、用户ID、密码，会员信息：会员生日、会员企业、会员种类、会员姓名、生日、住址、单位电话、手机 |
| EvaluationPO | 订单评价的信息：订单号、酒店ID、评价信息：星级、评论内容 |
| HotelPO | 酒店的信息：包括酒店名称，酒店所处商圈，酒店级别，酒店价格，酒店的地理位置，酒店实景图，酒店促销折扣，酒店简介；酒店评价信息：客户评价信息、评价星级 |
| RoomPO | 客房的信息：房间类型，房间价格，房间状态，房间号码，房间简介，房间实景图 |
| WebmarketstaffPO | 网站营销人员的信息：账户用户名，账户密码，浏览权限，修改客户信息的权限，真实姓名，联系方式，身份证号 |
| WebmanagementPO | 网站管理人员的信息：用户名，密码，修改信息的权限，真实姓名，联系方式，身份证号 |
| HotelManager | 酒店管理人员的信息：用户名，账户密码，所属酒店，联系方式，真实姓名，身份证号 |

持久化用户对象如UserPO的定义如图6.1，更多定义见原型代码



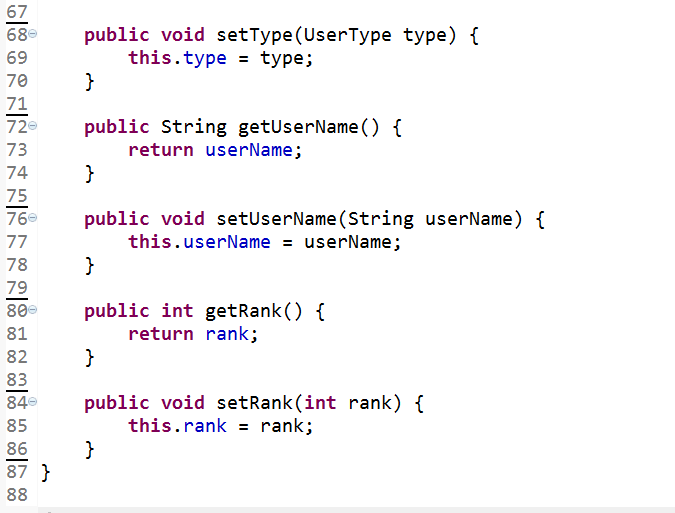
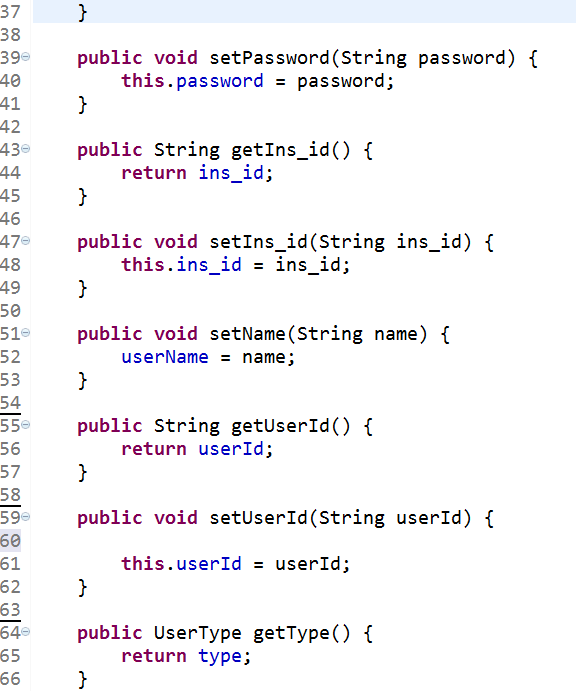


图6.1 持久化用户对象UserPO的定义

## 6.2数据库表

|  |  |
| --- | --- |
| 表名 | 内容 |
| Credit | 客户的信用信息 |
| Client | 客户的信息 |
| Account | 用户的账户信息 |
| Evaluation | 评价 |
| PromotionStrategy | 促销策略 |
| RoomInfo | 客房信息列表 |
| Order | 订单信息 |
| Hotel | 酒店具体信息 |
| HotelManager | 酒店管理人员的信息 |
| WebManager | 网站管理人员的信息 |
| WebBussiness | 网站营销人员的信息 |