|  |  |
| --- | --- |
| 屏幕快照%202016-10-08%2018  **互联网酒店预订系统**  软件体系结构设计模型文档 | 鄢煜民，何慧娴，顾恺嘉，黄潇  [2016-10-08] |

**更新历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **日期** | **变更原因** | **版本号** |
| 全体成员 | 2014/10/12 | 初步版，体系结构初步模型 | V0.1初步版 |
| 全体成员 | 2014/10/20 | 添加接口视角和信息视角 | V0.2 |
| 全体成员 | 2014/10/25 | 根据实际情况更改部分接口的方法 | V0.3 |
| 全体成员 | 2014/10/26 | 评审后的正式版 | V1.0正式版 |

**更新历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **日期** | **变更原因** | **版本号** |
| **鄢煜民** | **2016/10/8** | **创建** | **V1.0** |
| **黄潇** | **2016/10/8** | **修改开发包图：**  **添加Member及其子包**  **添加Member及其子包的接口、逻辑等** | **V1.1** |
| **顾恺嘉** | **2016/10/11** | **更新业务逻辑和数据逻辑的模版** | **V2.0** |
| **黄潇** | **2016/10/13** | **初步完成业务逻辑层的接口规范** | **V2.1** |
| **鄢煜民** | **2016/10/14** | **🙊初步完成数据层模块的接口规范** | **V2.2** |
| **全体成员** | **2016/10/15** | **定义输入规范，txt持久化格式，mysql持久化格式** | **V2.3** |
| **何慧娴** | **2016/10/16** | **整合所有包的逻辑层和数据层接口** | **V3.0** |
| **顾恺嘉** | **2016/12/11** | **修改模块，包的逻辑层数据层** | **V4.0** |
| **顾恺嘉** | **2016/12/16** | **最终修改** | **V5.0** |

**目录**

[1.引言 4](#_Toc20969)

[1.1编制目的 4](#_Toc10270)

[1.2词汇表 4](#_Toc3471)

[1.3参考资料 4](#_Toc19837)

[2.产品描述 4](#_Toc926)

[3. 逻辑视角 5](#_Toc27500)

[4. 组合视角 7](#_Toc4512)

[4.1开发包图 7](#_Toc20894)

[4.2运行时进程 11](#_Toc22402)

[4.3物理部署 12](#_Toc6958)

[5. 接口视角 12](#_Toc6992)

[5.1模块的职责 12](#_Toc6122)

[5.2用户界面层的分解 14](#_Toc20425)

[5.2.1用户界面层模块的职责 16](#_Toc28953)

[5.2.2用户界面层模块的接口规范 16](#_Toc31890)

[5.2.3用户界面模块设计原理 16](#_Toc27994)

[5.3.1业务逻辑层模块的职责 17](#_Toc32024)

[5.3.2业务逻辑层模块的接口规范 18](#_Toc5259)

[5.4.1数据层模块的职责 33](#_Toc29532)

[5.4.2数据层模块的接口规范 36](#_Toc15322)

[6. 信息视角 47](#_Toc30268)

[6.1数据持久化对象 47](#_Toc879)

[6.2 Txt持久化格式 50](#_Toc32509)

**1.引言**

**1.1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 |
| 用户 | 使用该系统预订酒店的用户 |  |
| 系统用户 | 使用该系统的用户 | 包括用户、酒店工作人员、网站管理人员、网站营销人员 |

**1.3参考资料**

1. 互联网酒店预订系统用例文档
2. 互联网酒店预订系统软件需求规格说明文档

**2.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软件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开发的目标是便于人们完成日常的酒店预订，包括酒店的订单管理，用户的订单管理，网站人员的管理等等等等。

通过互联网酒店预订系统的应用，期望为全面展现任一个地域内的酒店信息，帮助顾客进行更好地比较与选择。

1. **逻辑视角**

互联网酒店预订系统中，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3-1和图3-2所示。



图3-1 参照体系结构风格的包图表达逻辑视角

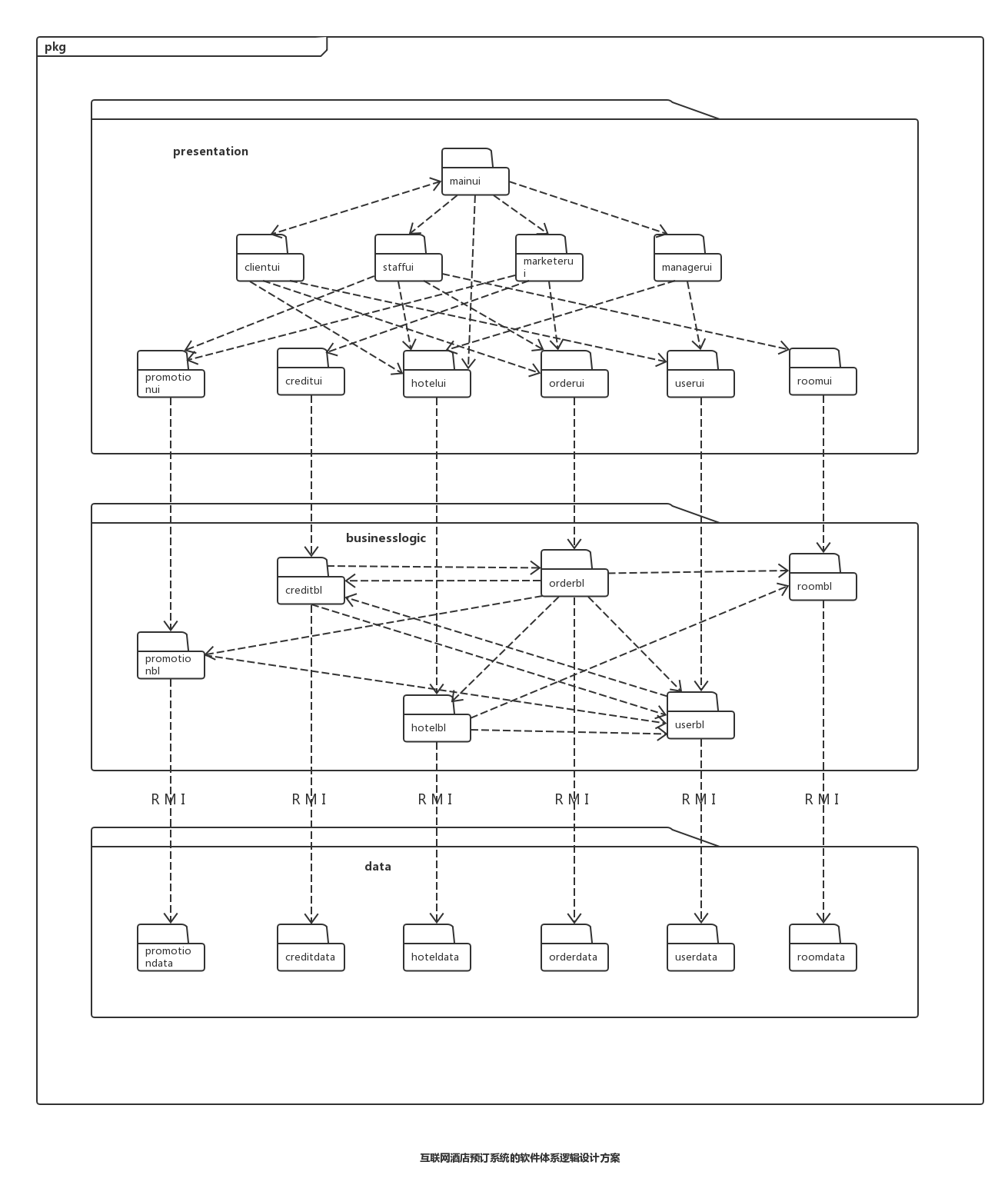


图3-2 软件体系结构逻辑设计方案

1. **组合视角**

**4.1开发包图**

互联网酒店预订系统的最终开发包设计如表4-1-1所示。

**表4-1-1 互联网酒店预定系统系统的最终开发包设计**

|  |  |
| --- | --- |
| **开发（物理）包** | **依赖的其他开发包** |
| mainui | clientui,staffui,marketerui,managerui |
| clientui | userui,hotelui,orderui |
| staffui | promotionui,hotelui,orderui,userui |
| marketerui | Promotionui,orderui |
| managerui | Hotelui,userui |
| promotionui | Promotionblservice,界面类库包,vo |
| promotionblservice |  |
| promotionbl | Promotionblservice,Promotiondataservice,po |
| promotiondataservice | Promotiondata,po,RMI |
| promotiondata | po,RMI |
| creditui | creditblservice,界面类库包,vo |
| creditblservice |  |
| creditbl | creditblservice,creditdataservice,po |
| creditdataservice | creditdata,po,RMI |
| creditdata | po,RMI |
| hotelui | hotelblservice,界面类库包,vo |
| hotelblservice |  |
| hotelbl | hotelblservice,hoteldataservice,po |
| hoteldataservice | hoteldata,po,RMI |
| hoteldata | po,RMI |
| orderui | orderblservice,界面类库包,vo |
| orderblservice |  |
| orderbl | orderblservice,orderdataservice,po |
| orderdataservice | orderdata,po,RMI |
| orderdata | po,RMI |
| roomui | roomblservice,界面类库包,vo |
| roomblservice |  |
| roombl | roomblservice,roomdataservice,po |
| roomdataservice | roomdata,po,RMI |
| roomdata | po,RMI |
| userui | userblservice,界面类库包,vo |
| userblservice |  |
| userbl | userblservice,userdataservice,po |
| userdataservice | userdata,po,RMI |
| userdata | po,RMI |
| vo |  |
| po |  |
| 界面类库包 |  |
| Java RMI |  |
| databaseutility | JDBC |

互联网酒店预订系统客户端开发包图如图4-1-1所示，服务器端开发包图如图4-1-2所示。

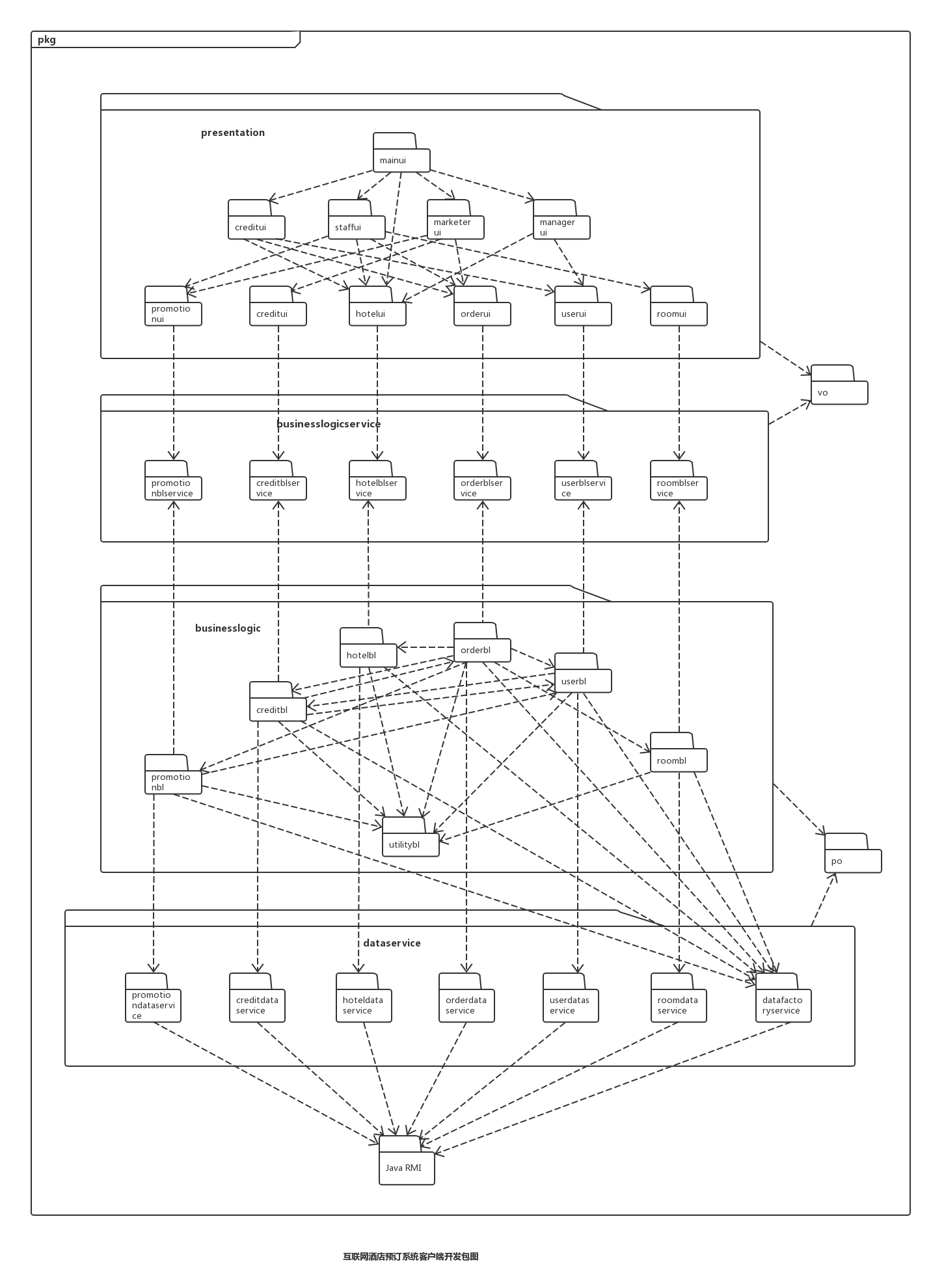


图4-1-1 互联网酒店预订系统客户端开发包图

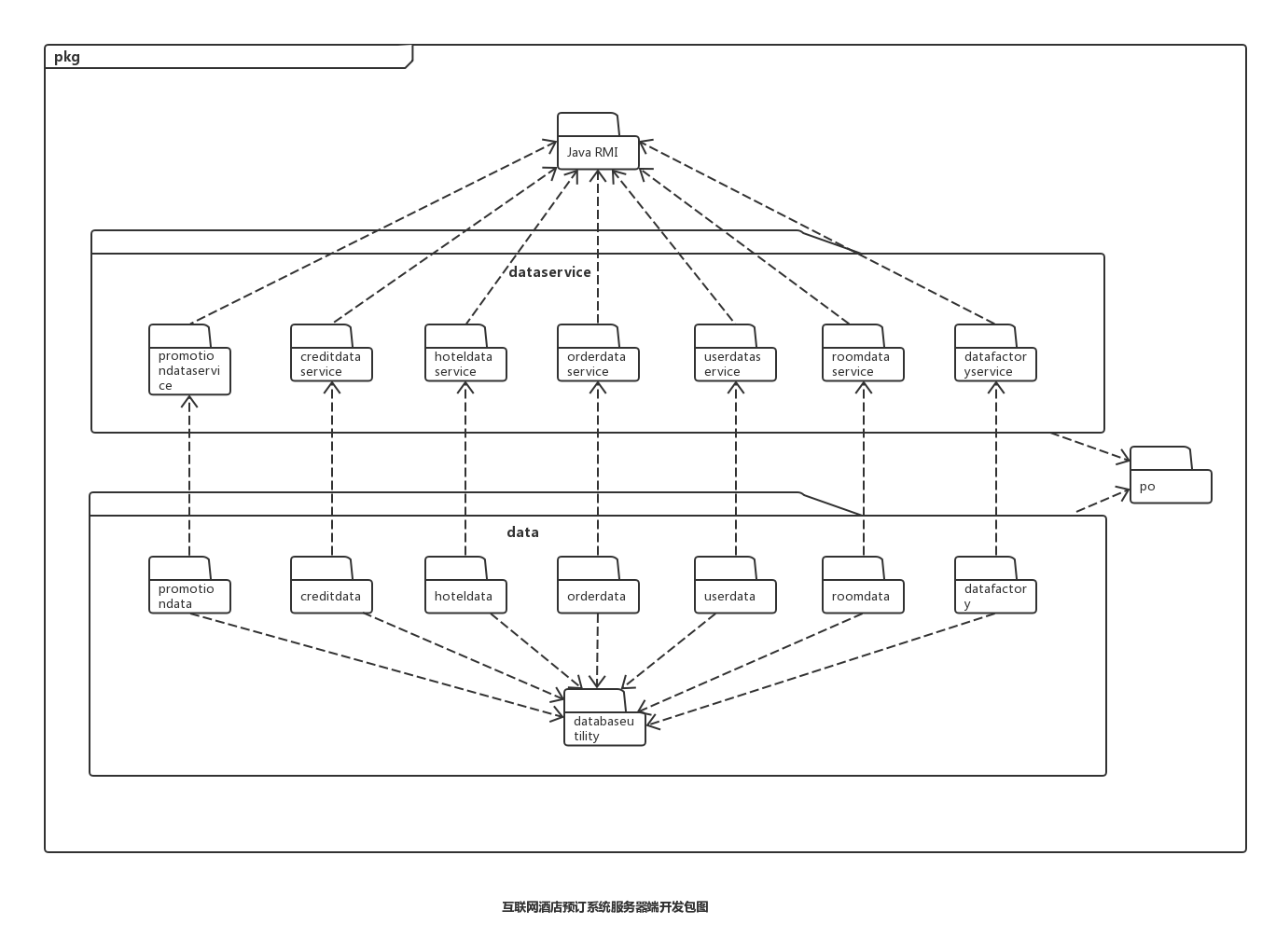


图4-1-2 互联网酒店预订系统服务器端开发包图

**4.2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，其进程图如图4-2-1所示。结合部署图，客户端进程是在客户端机器上运行，服务器端进程在服务器端机器上运行。



图4-2-1 进程图

**4.3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 1.7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4-3-1所示。



图4-3-1 部署图

1. **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。客户端各层和服务端各层的职责分别入表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 | 基于窗口的互联网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，启动服务器 |
| 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** |
| Creditblservice  Hotelblservice  Orderblservice  Promotionblservice  Roomblservice  Userblservice  Clientblservice  Managerblservice  Marketerblservice  Staffblservice | 客户端展示层 | 客户端业务逻辑层 |
| Creditdataservice  Hoteldataservice  Orderdataservice  Promotiondataservice  Roomdataservice  Userdataservice  Clientdataservice  Managerdataservice  Marketerdataservice  Staffbiservice | 客户端业务逻辑层 | 服务端数据层 |

**5.2用户界面层的分解**

根据需求，系统存在23个用户界面：界面跳转如图5-2-1所示。



图5-2-1 用户界面跳转

服务器端和客户端的用户界面设计接口是一致的，只是具体的页面不一样。

用户界面类如图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** |
| Presentation | 总界面模块，负责所有界面的显示和界面的跳转 |
| clientui | client界面模块，负责用户界面的显示和界面的跳转 |
| loginui | login界面模块，负责登陆/注册界面的显示和界面的跳转 |
| manageui | manage界面模块，负责网站管理人员界面的显示和界面的跳转 |
| marketui | market界面模块，负责网站营销人员界面的显示和界面的跳转 |
| staffui | staff界面模块，负责酒店工作人员界面的显示和界面的跳转 |
| utilui | 窗口界面模块，负责检查/窗口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**

|  |  |  |
| --- | --- | --- |
| Loginui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示登陆界面 |
| clientui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示用户界面 |
| staffui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示酒店工作人员界面 |
| marketui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示网站营销人员界面 |
| manageui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示网站管理人员界面 |
| utilui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示辅助界面 |

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.CreditBLService | 管理信用值的业务逻辑接口 |
| businesslogicservice.HotelBLService | 管理酒店的业务逻辑接口 |
| businesslogicservice.OrderBLService | 管理订单的业务逻辑接口 |
| businesslogicservice.PromotionBLService | 管理促销策略的业务逻辑接口 |
| businesslogicservice.RoomBLService | 管理酒店房间的业务逻辑接口 |
| businesslogicservice.ClientBLService | 用户管理的业务逻辑接口 |
| businesslogicservice.ManagerBLService | 网站管理人员管理的业务逻辑接口 |
| businesslogicservice.MarketerBLService | 网站营销人员管理的业务逻辑接口 |
| businesslogicservice.StaffBLService | 酒店工作人员管理的业务逻辑接口 |

5.2.3用户界面模块设计原理

用户界面利用JavaFX的库来实现。

**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。业务逻辑层的设计如图5-3-1所示。



图5-3-1 业务逻辑层的设计

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** |
| creditbl | 负责实现用户信用的录入和扣除所需要的服务 |
| hotelbl | 负责实现酒店信息管理和更新所需要的服务 |
| orderbl | 负责实现用户生成、执行、撤销、查看订单和酒店工作人员查看订单所需要的服务 |
| promotionbl | 负责实现促销策略的制定和实施所需要的服务 |
| userbl | 负责实现用户注册登陆所需要的服务 |
| roombl | 负责实现客房管理界面所需要的服务 |

5.3.2业务逻辑层模块的接口规范

**表5-3-2-1 creditbl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| credit.checkCreditRecord | 语法 | | public ArrayList<CreditVO> checkCreditRecord(String clientID) throws NetException |
| 前置条件 | | 客户查看信用记录 |
| 后置条件 | | 返回此客户信用变化记录列表 |
| credit.creditUpdate | 语法 | | public ResultMessage\_Credit creditUpdate(CreditVO updateVO) |
| 前置条件 | | 客户充值信用，酒店工作人员补登记执行订单  客户取消订单，系统计算减少的信用值 |
| 后置条件 | | 返回信用值更新的执行结果，记录此客户信用值改变记录 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| DataHelperFactory.getDataFactoryHelperInstance().getCreditDatabase() | | 得到Credit数据库的服务的引用 | |
| Creditdataservice. insert (CreditPO po) | | 插入单一持久化对象 | |
| Creditdataservice.find (String clientID) | | 根据客户ID进行查找多个持久化对象 | |

**表5-3-2-2 hotelbl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| Hotel.showHotelInfo | 语法 | public HotelVO showHotelInfo(String hotelID) throws NetException | |
| 前置条件 | 酒店ID正确 | |
| 后置条件 | 系统返回酒店信息 | |
| Hotel.changeHotelInfo | 语法 | public ResultMessage\_Hotel changeHotelInfo (HotelVO vo) | |
| 前置条件 | 酒店工作人员修改酒店信息 | |
| 后置条件 | 系统返回酒店工作人员修改的信息 | |
| Hotel.getHotelsSatisfyCondition | 语法 | public ArrayList<HotelVO> getHotelsSatisfyCondition(SearchCondition sc) throws NetException | |
| 前置条件 | 在数据库中存在相同的搜索条件的酒店 | |
| 后置条件 | 系统返回酒店列表信息 | |
| Hotel.getHotelEvaluation | 语法 | public Iterator<HotelEvaluationVO> getHotelEvalutions(String hotelID) throws NetException | |
| 前置条件 | 在数据库中存在相同ID的酒店 | |
| 后置条件 | 系统返回酒店评论信息 | |
| Hotel. evaluate | 语法 | public ResultMessage\_Hotel evaluate(HotelEvaluationVO vo) | |
| 前置条件 | 用户评价酒店 | |
| 后置条件 | 系统提示评价成功 | |
| **需要的服务（需接口）** | | | |
| 服务名 | | | 服务 |
| DataHelperFactory.getDataFactoryHelperInstance().getHotelDatabase() | | | 得到Hotel数据库的服务的引用 |
| HotelDataService.getHotelInfo (String hotelID) | | | 获取酒店数据 |
| HotelDataService. findHotelByCondition (SearchCondition sc) | | | 获取酒店列表数据 |
| HotelDataService.changeHotelInfo(HotelPO po) | | | 更新HotelPO单一持久化对象 |
| HotelDataService.evaluation (HotelEvaluationPO po) | | | 更新HotelEvaluationPO单一持久化对象 |
| HotelDataService.addHotel(HotelPO po) | | | 添加新酒店 |
| HotelDataService.getHotelEvaluation(String hotelID) | | | 获取酒店评价信息 |

**表5-3-2-3 orderbl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| Order.cancelOrder | 语法 | | public ResultMessage\_Order  cancelOrder(String orderId) |
| 前置条件 | | 用户或网站营销人员已登陆并确认撤销订单 |
| 后置条件 | | 系统返回是否撤销成功 |
| Order.excuteOrder | 语法 | | public ResultMessage\_Order  excuteOrder(String orderId) |
| 前置条件 | | 酒店工作人员确认执行订单 |
| 后置条件 | | 系统返回执行订单操作是否成功 |
| Order.finishOrder | 语法 | | public ResultMessage\_Order finishOrder(String orderID) |
| 前置条件 | | 酒店工作人员完成订单 |
| 后置条件 | | 系统返回完成订单操作是否成功 |
| Order.putUpOrder | 语法 | | public ResultMessage\_Order  putUpOrder(String orderId) |
| 前置条件 | | 酒店工作人员确认补录订单 |
| 后置条件 | | 系统返回订单补录是否成功 |
| Order.appeal | 语法 | | public ResultMessage\_Order appeal(String orderID, boolean all) |
| 前置条件 | | 网站管理人员处理申诉 |
| 后置条件 | | 系统返回处理申诉是否成功 |
| Order.queryOrderById | 语法 | | public OrderVo  queryOrderById(String orderId) |
| 前置条件 | | 订单号输入格式正确 |
| 后置条件 | | 系统查找是否存在该订单，根据输入的订单id返回订单信息 |
| Order.queryOrderByHotel | 语法 | | public ArrayList<OrderVo>  queryOrder(String hotelId,String userId) |
| 前置条件 | | 用户已选择酒店 |
| 后置条件 | | 系统查找是否存在该酒店，根据输入的酒店id返回订单信息 |
| Order.queryRoomOrder | 语法 | | public ArrayList<OrderVo>  queryRoomOrder(String hotelId,  String roomId) |
| 前置条件 | | 酒店工作人员已选择房间 |
| 后置条件 | | 系统查找是否存在该房间，根据输入的订单id返回该房间的预订订单列表 |
| Order.queryUserOrder | 语法 | | public ArrayList<OrderVo>  queryUserOrder(String userId) |
| 前置条件 | | 用户选择查询订单 |
| 后置条件 | | 系统查询是否存在该用户id，并根据id返回该用户的订单信息列表 |
| Order.queryUnexcutedOrder | 语法 | | public ArrayList<OrderVo>  queryUnexcutedOrder() |
| 前置条件 | | 网站营销人员选择查询未执行订单 |
| 后置条件 | | 系统返回未执行的订单列表 |
| Order.queryHotelOrder | 语法 | | public ArrayList<OrderVo>  queryHotelOrder(String hoteIId) |
| 前置条件 | | 酒店工作人员选择查询酒店订单 |
| 后置条件 | | 系统查找是否存在该酒店，根据输入的酒店id返回订单信息 |
| Order.makeOrder | 语法 | | public ResultMessage  makeOrder(OrderVo vo) |
| 前置条件 | | 用户确认订单信息无误 |
| 后置条件 | | 系统自动给该订单生成订单编号并持久化保留该订单信息 |
| Order.queryAbnormalOrder | 语法 | | public ArrayList<OrderVO> queryAbnormalOrder() throws NetException |
| 前置条件 | | 选择查询异常订单 |
| 后置条件 | | 系统返回异常订单列表 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| DataFactoryHelper.getDataFactoryInstance().getOrderDatabase() | | 得到Order数据的服务的引用 | |
| OrderDataService.addOrder(OrderPO po) | | 插入OrderPO单一持久化对象 | |
| OrderDataService.putUpOrder(String orderId) | | 补录该订单 | |
| OrderDataService. appealOrder(String orderID) | | 处理该订单的申诉 | |
| OrderDataService.cancelOrder(String orderID) | | 撤销该订单 | |
| OrderDataService.executeOrder(String orderID) | | 执行该订单 | |
| OrderDataService. finishOrder(String orderID) | | 完成该订单 | |
| OrderDataService.findById (String orderId) | | 得到该orderid的OrderPO持久化对象 | |
| OrderDataService.findByUser(String userId) | | 得到user的所有的OrderPO持久化对象 | |
| OrderDataService.findByRoom(String hotelId,String roomId) | | 得到worker所在酒店下的该Room的所有OrderPO持久化对象 | |
| OrderDataService.findUOByHotel(String hotelId, String userId) | | 得到user在该酒店下的所有的OrderPO持久化对象 | |
| OrderDataService.findHotelOrder(String hotelId) | | 得到worker所在酒店下的所有的OrderPO持久化对象 | |
| OrderDataService.findUnexcutedOrder() | | 得到当天所有未执行订单的OrderPO持久化对象 | |
| OrderDataService.orderEvaluate(String orderID) | | 更新该订单为已评论状态 | |
| OrderDataService. queryAbnormalOrder() | | 查找异常订单 | |

**表5-3-2-4 promotionbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Promotion.addhotelPromotion | 语法 | Public String addhotelPromotion (PromotionVO vo) throws NetException |
| 前置条件 | 输入信息符合输入规则 |
| 后置条件 | 系统保存并更新了当前酒店促销策略 |
| Promotion.addWebPromotion | 语法 | public String addWebPromotion (PromotionVO vo) throws NetException |
| 前置条件 | 输入信息符合输入规则 |
| 后置条件 | 系统保存并更新了网站的促销策略 |
| Promotion.getHotelPromotion | 语法 | public ArrayList<PromotionVO> getHotelPromotion (String hotelID) throws NetException |
| 前置条件 | 无 |
| 后置条件 | 系统根据输入的酒店名称查找并返回酒店促销策略 |
| Promotion.getWebPromotion | 语法 | public ArrayList<PromotionVO> getWebPromotion() throws NetException |
| 前置条件 | 无 |
| 后置条件 | 系统返回网站促销策略 |
| Promotion.getPromotionById | 语法 | public PromotionVO getPromotionById (String promotionID) throws NetException |
| 前置条件 | ID已存在 |
| 后置条件 | 系统返回该ID的促销策略 |
| Promotion.cancel | 语法 | public ResultMessage\_Promotion cancel(String promotionID) |
| 前置条件 | 无 |
| 后置条件 | 系统应该根据输入的促销策略编号查找并取消当前策略 |
| Promotion.getMemberLevel | 语法 | public ArrayList<Integer> getMemberLevel() throws NetException |
| 前置条件 | 无 |
| 后置条件 | 系统返回会员等级划分方案 |
| Promotion.getMemberDiscount | 语法 | publicA rrayList<Double> getMemberDiscount() throws NetException |
| 前置条件 | 无 |
| 后置条件 | 系统返回不同等级会员折扣 |
| Promotion.makeLevel | 语法 | public ResultMessage\_Promotion makeLevel(ArrayList<Integer> level, ArrayList<Double> discount) |
| 前置条件 | 人员制定会员等级 |
| 后置条件 | 系统返回制定会员等级是否成功 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| DataFactory.getPromotionData() | | 得到Promotion的数据的服务的引用 |
| PromotionDataService. addPromotion(PromotionPO po) | | 新增促销策略PromotionPO的单一持久化对象 |
| PromotionDataService. getHotelPromotion(String hotelID) | | 根据酒店信息在数据库中查找对应的酒店，并返回相应的酒店促销策略PromotionPO结果 |
| PromotionDataService. getWebPromotion() | | 在数据库中查找，并返回网站的促销策略PromotionPO结果 |
| PromotionDataService. getPromotionById(String promotionID) | | 在数据库中查找，根据ID返回促销策略 |
| PromotionDataService. deletePromotion(String promotionID) | | 删除数据库中PromotionPO单一持久化对象 |
| promotionDataSercive. getMemberLevel() | | 查看会员等级划分方案 |
| promotionDataSercive.getMemberDiscount() | | 查看不同等级会员折扣 |
| promotionDataSercive.levelMake(ArrayList<Integer> level, ArrayList<Double> discount) | | 制定会员等级制度 |

**表5-3-2-5 roombl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| Room.checkIn | 语法 | | public ResultMessage\_Room checkIn(String hotelId, String roomNumber) |
| 前置条件 | | 酒店工作人员已添加住房所需信息 |
| 后置条件 | | 返回入住客房的执行结果 |
| Room.checkOut | 语法 | | public ResultMessage\_Room checkOut(String hotelId, String roomNumber) |
| 前置条件 | | 酒店工作人员已添加退房所需信息 |
| 后置条件 | | 返回退房的执行结果 |
| Room.importRoom | 语法 | | public ArrayList<String> importRoom(ArrayList<RoomVO> importRoomList) |
| 前置条件 | | 酒店工作人员已添加需要录入客房的信息 |
| 后置条件 | | 返回录入客房的执行结果 |
| Room.getRoomList | 语法 | | public ArrayList<RoomVO> getRoomList(String hotelID) |
| 前置条件 | | 用户查看酒店房间 |
| 后置条件 | | 返回此酒店所有房间信息列表 |
| Room.getOrderRecord | 语法 | | public ArrayList<RoomRecordVO> getOrderReord(String hotelID, String roomNumber) |
| 前置条件 | | 无 |
| 后置条件 | | 返回订单记录 |
| Room.getRoomInfo | 语法 | | public RoomVO getRoomInfo(String hotelId, String roomNumber) throws RemoteException; |
| 前置条件 | | Id输入符合规范 |
| 后置条件 | | 获取房间的信息 |
| **需要的服务（需接口）** | | | |
| 服务名 | | 服务 | |
| DataHelperFactory.getDataFactoryHelperInstance().getRoomDatabase() | | 得到Room数据库的服务的引用 | |
| Roomdataservice.getRoom(String hotelID) | | 获取房间列表 | |
| Roomdataservice.getRoomInfo(String hotelId, String roomNumber) | | 查询单个房间信息 | |
| Roomdataservice.addRoom(RoomPO po) | | 添加RoomPO单一持久化对象 | |
| Roomdataservice.checkIn(String hotelID, String roomNumber) | | 入住 | |
| Roomdataservice.checkOut(String hotelID, String roomNumber) | | 离开 | |
| Roomdataservice. getOrderRecord(String hotelID, String roomNumber) | | 获得预定记录 | |
| Roomdataservice. addRecord(RoomRecordPO po) | | 增加房间预定记录 | |
| Roomdataservice. deleteRecord(String orderID) | | 删除房间预定记录 | |

**表5-3-2-6 userbl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| userbl.Client.regist |  | public ResultMessage\_User regist (ClientRegistVO info) | |
|  | 无 | |
|  | 系统返回用户注册结果 | |
| userbl.Client.getClientInfo | 语法 | public ClientVO getClientInfo (String clientID) throws NetException | |
| 前置条件 | 用户ID正确 | |
| 后置条件 | 系统返回用户信息 | |
| userbl. Client.changeClientInfo | 语法 | public ResultMessage\_User changeClientInfo (ClientInfoChangeVO vo) | |
| 前置条件 | 用户修改个人信息 | |
| 后置条件 | 系统返回修改结果 | |
| userbl. Client.registerMember | 语法 | public ResultMessage\_User registerMember (MemberVO vo) | |
| 前置条件 | 用户注册会员 | |
| 后置条件 | 系统返回注册结果 | |
| userbl.Staff.showData | 语法 | public StaffVO showData (String staffID) | |
| 前置条件 | 酒店工作人员ID正确 | |
| 后置条件 | 系统返回酒店工作人员信息 | |
| userbl. Marketer.showData | 语法 | public MarketerVO showData (String marketerID) | |
| 前置条件 | 网站营销人员ID正确 | |
| 后置条件 | 系统返回网站营销人员信息 | |
| userbl.Manager.getManagerInfo |  | public ManagerVO getManagerInfo() throws NetException | |
|  | 网站管理人员ID与密码匹配 | |
|  | 系统返回网站管理人员信息 | |
| userbl.Manager.changeManagerInfo |  | public ResultMessage\_User changeManagerInfo (ManagerVO vo) | |
|  | 修改网站管理人员信息 | |
|  | 系统返回修改后的信息 | |
| userbl. Manager.showClientData | 语法 | public ClientVO showClientData (String clientID)throws NetException | |
| 前置条件 | 用户ID正确 | |
| 后置条件 | 系统返回用户信息 | |
| userbl. Manager.getClientList | 语法 | public ArrayList<ClientVO> getClientList() throws NetException | |
| 前置条件 | 无 | |
| 后置条件 | 系统返回用户列表 | |
| userbl. Manager.changeClientInfo | 语法 | public ResultMessage\_User changeClientInfo (ClientInfoChangeVO vo) | |
| 前置条件 | 网站管理人员修改用户信息 | |
| 后置条件 | 系统返回修改用户信息结果 | |
| userbl. Manager.showStaffData | 语法 | public StaffVO showStaffData (String staffID) throws NetException | |
| 前置条件 | 酒店工作人员ID正确 | |
| 后置条件 | 系统返回酒店工作人员信息 | |
| userbl. Manager.changeStaffData | 语法 | public ResultMessage\_User changeStaffData (StaffChangeVO vo) | |
| 前置条件 | 网站管理人员修改酒店工作人员信息 | |
| 后置条件 | 系统返回修改后的信息 | |
| userbl. Manager.addStaff | 语法 | public ResultMessage\_User addStaff (StaffVO registVO, String password) | |
| 前置条件 | 网站管理人员添加网站营销人员 | |
| 后置条件 | 系统创建新网站营销人员 | |
| userbl. Manager.getStaffList | 语法 | public ArrayList<StaffVO> getStaffList() throws NetException | |
| 前置条件 | 无 | |
| 后置条件 | 系统返回酒店管理人员列表 | |
| userbl. Manager.showMarketerData | 语法 | public MarketerVO showMarketerData (String marketerID) throws NetException | |
| 前置条件 | 网站营销人员ID正确 | |
| 后置条件 | 系统返回网站营销人员信息 | |
| userbl. Manager.changeMarketerData | 语法 | public ResultMessage\_User changeMarketerData (MarketerVO vo) | |
| 前置条件 | 网站管理人员修改网站营销人员信息 | |
| 后置条件 | 系统返回修改后的信息 | |
| userbl. Manager.addMarketer | 语法 | public ResultMessage\_User addMarketer(MarketerVO vo, String password) | |
| 前置条件 | 网站管理人员添加网站营销人员 | |
| 后置条件 | 系统创建新网站营销人员 | |
| userbl. Manager. getMarketerList | 语法 | public ArrayList<MarketerVO> getMarketerList() throws NetException | |
| 前置条件 | 无 | |
| 后置条件 | 系统返回网站营销人员列表 | |
| userbl. Manager.addHotel | 语法 | public String addHotel(HotelVO vo) throws NetException | |
| 前置条件 | 网站管理人员添加酒店 | |
| 后置条件 | 系统返回添加结果 | |
| **需要的服务（需接口）** | | | |
| 服务名 | | | 服务 |
| DataHelperFactory.getDataFactoryHelperInstance().getClientDatabase() | | | 获取用户数据的服务的引用 |
| DataHelperFactory.getDataFactoryHelperInstance().getStaffDatabase() | | | 获取酒店工作人员数据的服务的引用 |
| DataHelperFactory.getDataFactoryHelperInstance().getMarketerDatabase() | | | 获取网站营销人员数据的服务的引用 |
| DataHelperFactory.getDataFactoryHelperInstance().getManagerDatabase() | | | 获取网站管理人员数据的服务的引用 |
| ControllerFactory.getClient2ManagerInstance() | | | 获取用户数据的服务的引用 |
| ControllerFactory.getStaff2ManagerInstance() | | | 获取酒店工作人员数据的服务的引用 |
| ControllerFactory.getMarketer2ManagerInstance() | | | 获取网站营销人员数据的服务的引用 |
| ControllerFactory.getHotel2ManagerInstance() | | | 获取酒店数据的服务的引用 |
| ClientDataService.regist(ClientRegistPO po) | | | 添加ClientPO单一持久化对象 |
| ClientDataService.queryClient(String clientID) | | | 获取ClientPO内部信息 |
| ClientDataService. updateClientInfo (ClientInfoChangePO changePO) | | | 更新ClientPO单一持久化对象 |
| ClientDataService. registerMember(MemberPO po) | | | 添加MemberPO单一持久化对象 |
| ClientDataService.getClientList() | | | 获取用户列表信息 |
| StaffDataService.findData(String staffID) | | | 获取酒店工作人员数据 |
| StaffDataService.updateData (StaffChangePO po) | | | 更新StaffPO单一持久化对象 |
| StaffDataService.insert (StaffPO po, String password) | | | 添加StaffO单一持久化对象 |
| StaffDataService.getStaffList() | | | 获取酒店工作人员列表信息 |
| MarketerDataService.getMarketerInfo (String marketerID) | | | 获取网站营销人员数据 |
| MarketerDataService.updateData (MarketerPO po) | | | 更新MarketerPO单一持久化对象 |
| MarketerDataService.find(String marketerID, String password) | | | 测试登录信息是否正确 |
| MarketerDataService.insert (MarketerPO po, String password) | | | 添加MarketerPO单一持久化对象 |
| MarketerDataService.getMarketerList() | | | 获取网站营销人员列表信息 |
| ManagerDataService.getManagerInfo() throws RemoteException | | | 获取网站管理人员信息 |
| ManagerDataService. changeManagerInfo (ManagerPO po) | | | 更新ManagerPO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访问服务，包括对于持久化数据的增、删、改、查。各业务逻辑需要的服务由各DataService接口提供。持久化数据的保存采用MySQL数据库进行保存。

以RoomDataService为例，数据层模块的描述具体如图5-4-1所示。



图5-4-1 RoomDataService数据层模块

5.4.1数据层模块的职责

数据层模块的职责如表5-4-1-1 ~ 5-4-1-7所示。

**表5-4-1-2 数据层模块(PromotionDataService)的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| PromotionDataService | 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 |
| PromotionDataServiceTxtFileImpl | 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 |
| PromotionDataServiceMySqlImpl | 基于MySql数据库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 |

**表5-4-1-3 数据层模块(CreditDataService)的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| CreditDataService | 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 |
| CreditDataServiceTxtFileImpl | 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 |
| CreditDataServiceMySqlImpl | 基于MySql数据库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 |

**表5-4-1-4 数据层模块(HotelDataService)的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| HotelDataService | 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 |
| HotelDataServiceTxtFileImpl | 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 |
| HotelDataServiceMySqlImpl | 基于MySql数据库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 |

**表5-4-1-5 数据层模块(OrderDataService)的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| OrderDataService | 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 |
| OrderDataServiceTxtFileImpl | 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 |
| OrderDataServiceMySqlImpl | 基于MySql数据库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 |

**表5-4-1-6 数据层模块(UserDataService)的职责**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块** | | | | **职责** |
| UserDataService | | ClientDataService  MemberData  Service | | 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 |
| StaffDataService | |
| MarketerData  Service | |
| ManagerData  Service | |
| UserDataServiceTxtFileImpl | ClientDataService  MemberData  ServiceTxtFileImpl | | | 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 |
| StaffDataServiceTxtFileImpl | | |
| MarketerData  ServiceTxtFileImpl | | |
| ManagerData  ServiceTxtFileImpl | | |
| UserDataServiceMySqlImpl | | | ClientDataService  MemberData  ServiceMySqlImpl | 基于MySql数据库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 |
| StaffDataServiceMySqlImpl |
| MarketerData  ServiceMySqlImpl |
| ManagerData  ServiceMySqlImpl |

**表5-4-1-7 数据层模块(RoomDataService)的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| RoomDataService | 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 |
| RoomDataServiceTxtFileImpl | 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 |
| RoomDataServiceMySqlImpl | 基于MySql数据库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 |

5.4.2数据层模块的接口规范

数据层模块的接口规范如表5-4-2-1 ~ 5-4-2-7所示。

**表5-4-2-1 数据层模块(PromotionDataService)的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PromotionDataService.addPromotion | 语法 | public String addPromotion (PromotionPO po) throws RemoteException |
| 前置条件 | 输入信息符合输入规则 |
| 后置条件 | 在数据库中增加一个po记录 |
| PromotionDataService. getHotelPromotion | 语法 | public ArrayList<PromotionPO> getHotelPromotion(String hotelID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 从数据库中查找并返回相应的PromotionPO结果 |
| PromotionDataService. getWebPromotion | 语法 | public ArrayList<PromotionPO> getWebPromotion() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 从数据库中查找并返回相应的PromotionPO结果 |
| PromotionDataService. getPromotionById | 语法 | public PromotionPO getPromotionById(String promotionID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 从数据库中查找并返回相应的PromotionPO结果 |
| PromotionDataService.  deletePromotion | 语法 | public ResultMessage\_Promotion deletePromotion (String promotionID) throws RemoteException |
| 前置条件 | 当前策略过期失效 |
| 后置条件 | 在数据库中删除PromotionPO记录 |
| PromotionDataService. getMemberLevel | 语法 | public ArrayList<Integer> getMemberLevel() throws RemoteException |
| 前置条件 | 查看会员等级划分方案 |
| 后置条件 | 从数据库中查找并返回相应的结果 |
| PromotionDataService. getMemberDiscount | 语法 | public ArrayList<Double> getMemberDiscount() throws RemoteException |
| 前置条件 | 查看不同等级会员折扣 |
| 后置条件 | 从数据库中查找并返回相应的结果 |
| PromotionDataService.levelMake | 语法 | public ResultMessage\_Promotion levelMake(ArrayList<Integer> level, ArrayList<Double> discount) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 制定会员等级制度 |

**表5-4-2-3 数据层模块(CreditDataService)的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| creditdataservice.insert | 语法 | public ResultMessage\_Credit insert(CreditPO po) throws NetException |
| 前置条件 | 同样ID的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| creditdataservice.find | 语法 | public ArrayList<CreditPO> find(String clientID) throws NetException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的Credit结果 |

**表5-4-2-4 数据层模块(HotelDataService)的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| HotelDataService.getHotelInfo | 语法 | public HotelPO getHotelInfo(String hotelID) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的HotelPO |
| 后置条件 | 从数据库获取HotelPO内部数据 |
| HotelDataService. findHotelByCondition | 语法 | public ArrayList<HotelPO> findHotelByCondition(SearchCondition sc) throws RemoteException |
| 前置条件 | 在数据库中存在相同的搜索条件的酒店 |
| 后置条件 | 系统返回酒店列表信息 |
| HotelDataService.changeHotelInfo | 语法 | public ResultMessage\_Hotel changeHotelInfo(HotelPO po) throws RemoteException |
| 前置条件 | 在数据库中存在相同HotelID的po |
| 后置条件 | 在数据库中更新一个HotelPO记录 |
| HotelDataService.Evaluation | 语法 | public ResultMessage\_Hotel Evaluation (HotelEvaluationPO po) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的HotelEvaluationPO |
| 后置条件 | 在数据库中更新一个HotelEvaluationPO记录 |
| HotelDataService.addHotel | 语法 | public ResultMessage\_Hotel addHotel(HotelPO po) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 在数据库中生成一个新ID |
| HotelDataService.getHotelEvaluation | 语法 | public ArrayList<HotelEvaluationPO> getHotelEvaluation(String hotelID) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的酒店 |
| 后置条件 | 获取酒店评论列表 |

**表5-4-2-5 数据层模块(OrderDataService)的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| OrderDataService.addOrder | 语法 | public String addOrder(OrderPO po) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 在数据库中增加一个po记录 |
| OrderDataService.putUpOrder | 语法 | public ResultMessage\_Order putUpOrder(String orderID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 补录订单 |
| OrderDataService. appealOrder | 语法 | public ResultMessage\_Order appealOrder(String orderID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 处理申诉订单 |
| OrderDataService.cancelOrder | 语法 | public ResultMessage\_Order cancelOrder(String orderID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 撤销订单 |
| OrderDataService.executeOrder | 语法 | public ResultMessage\_Order executeOrder(String orderID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 执行订单 |
| OrderDataService. finishOrder | 语法 | public ResultMessage\_Order finishOrder(String orderID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 完成订单 |
| OrderDataService.  findById | 语法 | public OrderPO findById(String orderID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 系统返回一个PO记录 |
| OrderDataService.findByUser | 语法 | public ArrayList<OrderPO> findByUser(String clientID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的OrderPO记录 |
| OrderDataService.  findByRoom | 语法 | public ArrayList<OrderPO> findByRoom(String hotelID, String roomNumber) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 系统返回一个或多个PO记录 |
| OrderDataService.findUOByHotel | 语法 | public ArrayList<OrderPO> findUOByHotel(String hotelID, String clientID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回该用户在该酒店所有的  OrderPO记录 |
| OrderDataService.findHotelOrder | 语法 | public ArrayList<OrderPO> findHotelOrder(String hotelID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回该酒店所有的OrderPO记录 |
| OrderDataService.find UnexcutedOrder | 语法 | public ArrayList<OrderPO> findUnexecutedOrder() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回当天所有未执行的OrderPO记录 |
| OrderDataService. orderEvaluate | 语法 | public ResultMessage\_Order orderEvaluate(String orderID) throws RemoteException |
| 前置条件 | 更新订单为已评价状态 |
| 后置条件 | 返回更新订单状态的结果 |
| OrderDataService. findAbnormalOrder | 语法 | public ArrayList<OrderPO> findAbnormalOrder() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回该用户所有的异常订单记录 |

**表5-4-2-6 数据层模块(UserDataService)的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| UserDataService.ClientDataService.regist | 语法 | public ResultMessage\_User regist(ClientRegistPO po) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 在数据库生成一个新的ClientPO |
| UserDataService.ClientDataService. queryClient | 语法 | public ClientPO queryClient(String clientID) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的clientPO |
| 后置条件 | 从数据库获取clientPO内部数据 |
| UserDataService.ClientDataService. updateClientInfo | 语法 | public ResultMessage\_User updateClientInfo(ClientInfoChangePO changePO) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的clientPO |
| 后置条件 | 在数据库中更新一个clientPO记录 |
| UserDataService.ClientDataService.registerMember | 语法 | public ResultMessage\_User registerMember (MemberPO po) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 在数据库中生成一个memberPO记录 |
| UserDataService.ClientDataService. getClientList | 语法 | public ArrayList<ClientPO> getClientList() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 从数据库获取用户列表信息 |
| UserDataService.StaffDataService.findData | 语法 | public StaffPO findData(String staffID) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的StaffPO |
| 后置条件 | 从数据库获取StaffPO内部数据 |
| UserDataService.StaffDataService.updateData | 语法 | public ResultMessage\_User updateData(StaffPO po) throws RemoteException |
| 前置条件 | 在数据库中存在相同staffID的po |
| 后置条件 | 在数据库中更新一个staffpPO记录 |
| UserDataService.StaffDataService.insert | 语法 | public ResultMessage\_User insert(StaffPO po, String password) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 在数据库中生成一个StaffPO记录 |
| UserDataService.StaffDataService.getStaffList | 语法 | public ArrayList<StaffPO> getStaffList() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 从数据库获取酒店工作人员列表信息 |
| UserDataService.MarketerDataService. getMarketerInfo | 语法 | public MarketerPO getMarketerInfo(String marketerID) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的MarketerPO |
| 后置条件 | 从数据库获取MarketerPO内部数据 |
| UserDataService.MarketerDataService.updateData | 语法 | public ResultMessage\_User updateData(MarketerPO po) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的marketerPO |
| 后置条件 | 在数据库中更新一个marketerPO记录 |
| UserDataService.MarketerDataService.find | 语法 | public ResultMessage\_User find(String marketerID, String password) throws RemoteException |
| 前置条件 | 在数据库中存在相同的ID |
| 后置条件 | 密码匹配系统返回许可；密码不匹配系统返回错误 |
| UserDataService.MarketerDataService.insert | 语法 | public ResultMessage\_User insert (MarketerPO po, String password) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 在数据库中生成一个marketerPO记录 |
| UserDataService.MarketerDataService. getMarketerList | 语法 | public ArrayList<MarketerPO> getMarketerList() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 从数据库获取网站营销人员列表信息 |
| UserDataService.ManagerDataService. getManagerInfo | 语法 | public ManagerPO getManagerInfo() throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的ManagerPO |
| 后置条件 | 从数据库获取ManagerPO内部数据 |
| UserDataService.ManagerDataService. changeManagerInfo | 语法 | public ResultMessage\_User changeManagerInfo (ManagerPO po) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的managerPO |
| 后置条件 | 在数据库中更新一个managerPO记录 |

**表5-4-2-7 数据层模块(RoomDataService)的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| Roomdataservice.getRoom | 语法 | public ArrayList<RoomPO> getRoom(String hotelID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的RoomPO结果 |
| Roomdataservice.getRoomInfo | 语法 | public RoomPO getRoomInfo(String hotelID, String roomNumber) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 查询房间信息 |
| Roomdataservice.addRoom | 语法 | public ResultMessage\_Room addRoom(RoomPO po) throws RemoteException |
| 前置条件 | 同样ID的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| Roomdataservice.checkIn | 语法 | public ResultMessage\_Room checkIn(String hotelID, String roomNumber) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 入住 |
| Roomdataservice.checkOut( | 语法 | public ResultMessage\_Room checkOut(String hotelID, String roomNumber) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 离开 |
| Roomdataservice.getOrderRecord | 语法 | public ArrayList<RoomRecordPO> getOrderRecord(String hotelID, String roomNumber) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 获得预定记录 |
| Roomdataservice.addRecord | 语法 | public ResultMessage\_Room addRecord(RoomRecordPO po) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 增加房间预定记录 |
| Roomdataservice.deleteRecord | 语法 | public ResultMessage\_Room deleteRecord(String orderID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 删除房间预定记录 |

1. **信息视角**

**6.1数据持久化对象**

系统的PO类是对应的相关的实体类。系统主要的PO类如下所示：

**OrderPO: 订单PO类**

包含：订单ID，订单状态，用户ID，酒店ID，订单预定房间号列表，使用促销策略列表，订单价格，下单时间，订单执行时间，结束时间（执行完成/撤销），预订入住日期，预定离开日期，实际离开日期，预计入住人数，有无儿童

**CreditPO: 信用值PO类**

包含：用户ID，信用改变时间，信用改变动作，订单号（当改变动作为充值信用时可为空），信用改变值，改变后信用值

**HotelPO: 酒店PO类**

包含：酒店ID，酒店名称，酒店地址，酒店所属地区，酒店所属商圈，酒店星级，酒店评分

**HotelEvaluationPO: 酒店评价PO类**

包含：被评论酒店ID，评论者，评价时间，评分，评论

**MemberPO: 会员PO类**

包含： 用户ID， 用户会员类型， 用户会员等级，用户会员信息

**PromotionPO: 促销策略PO类**

包含： 促销策略ID，促销策略信息，商圈或者酒店ID，促销策略开始时间，促销策略结束时间，促销策略折扣，会员等级数组

**EnterprisePromotionPO: 合作企业促销策略PO类**

包含：促销策略合作企业列表，PromotionPO内部信息

**DistrictPromotionPO: 作用商圈促销策略PO类**

包含：促销策略作用商圈列表，PromotionPO内部信息

**RoomPO: 房间PO类**

包含：所属酒店ID，房间号，房间类型，房间价格，当前状态

**RoomRecordPO: 房间入住信息PO类**

包含：房间所属酒店ID，房间号，订单号，订单记录入住时间，订单预计离开时间

**ClientPO类：用户PO类**

包含：用户ID，用户姓名/名称，用户联系方式，信用值，用户会员类型， 用户会员等级，用户会员信息

**ClientInfoChangePO类：用户修改信息PO类**

包含：用户ID，用户姓名/名称，用户联系方式

**ClientRegistPO类：用户注册PO类**

包含：用户ID，用户姓名/名称，用户联系方式，密码

**StaffPO类：酒店工作人员PO类**

包含：酒店工作人员ID，酒店工作人员姓名，酒店工作人员联系方式，酒店工作人员所在酒店ID

**StaffChangePO类：酒店工作人员信息修改PO类**

包含：酒店工作人员ID，酒店工作人员姓名，酒店工作人员联系方式

**MarketerPO类：网站营销人员PO类**

包含：网站营销人员ID，网站营销人员姓名，网站营销人员联系方式

**ManagerPO类：网站管理人员PO类**

包含：网站管理人员ID，网站管理人员姓名

持久化信用记录对象如CreditPO的定义如图6-1-1所示。

|  |
| --- |
| /\* @author huangxiao  \* 2016年10月14日  \*/  package po;  public class CreditPO {  /\*\*  \* 客户id  \*/  private String clientID;    /\*\*  \* 信用改变时间  \*/  private String changeTime;    /\*\*  \* 信用改变值  \*/  private int changeValue;    /\*\*  \* 改变后信用值  \*/  private int credit;    /\*\*  \* 构造方法  \* @param clientID  \* @param changeTime  \* @param changeValue  \* @param credit  \*/  public CreditPO(String clientID, String changeTime, int changeValue, int credit) {  this.clientID = clientID;  this.changeTime = changeTime;  this.changeValue = changeValue;  this.credit = credit;  }  public String getClientID() {  return clientID;  }  public void setClientID(String clientID) {  this.clientID = clientID;  }  public String getChangeTime() {  return changeTime;  }  public void setChangeTime(String changeTime) {  this.changeTime = changeTime;  }  public int getChangeValue() {  return changeValue;  }  public void setChangeValue(int changeValue) {  this.changeValue = changeValue;  }  public int getCredit() {  return credit;  }  public void setCredit(int credit) {  this.credit = credit;  }    } |

图6-1-1 持久化订单对象CreditPO的定义

**6.2 MySQL持久化格式**

ID规范：  
订单号：   16位  
 1234567890123456

系统用户：  10位

1234567890  
酒店：     8位  
 12345678

促销策略： 9位

123456789

商圈： 6位

123456

信息规范：

姓名/名称：8-16位字符串

administrator

联系方式：手机号码，11位字符串

12345678901

信用值：整型数据

100

密码：     8-16位，仅英文字母加数字

Yym123456789

日期输入规范：

2016/10/16

评分： 1-5整数

2

酒店名称：无特殊字符

和园宾馆

折扣额度： 0-1之间的double值

0.96

时间规范：促销策略开始时间必须小于结束时间

酒店入住时间必须小于离开时间

订单最晚执行时间为订单中预定的入住日期当晚12:00前

**6.3 Txt持久化格式**

各Txt数据保存格式如下所示：

每行属性之间用英文分号“;”分隔，若属性为列表，则列表项间用竖线“|”分隔，列表项内部用英文逗号“,”分隔。

Room.txt

所属酒店id;房间号;房间类型;房间价格;

入住日期,预计退房日期,实际退房日期|

```

入住日期,预计退房日期,实际退房日期;

日期,日期对应房间状态|

```

日期,日期对应房间状态

00000001;1024;单人间;100;

2016/10/14,2016/10/15,2016/10/15|

```

2016/10/16,2016/10/17,2016/10/18;

2016/10/14,Reserved|

2016/10/15,NotReserved|

```

2016/10/17,Occupied

Hotel.txt

酒店id;酒店名称;酒店地址;酒店所属地区;酒店所属商圈;酒店星级;酒店评分

12345678，橘子水晶假日酒店，南京新街口洪武路9号，新街口，五星，4

HotelEvaluation.txt

被评论酒店id;评价客户名称;评价时间;评分;评论

12345678，丁二玉，2016／10/16，4，很好

Client.txt

用户ID，用户姓名/名称，用户联系方式，信用值

0000000001,administrator,12345678901,100

Staff.txt

酒店工作人员ID，酒店工作人员姓名，酒店工作人员所在酒店ID，酒店工作人员所在酒店名称

0000000001,administrator,0000000001,adminhotel

Marketer.txt  
网站营销人员ID，网站营销人员姓名，网站营销人员联系方式

0000000001,administrator,12345678901

member.txt

会员编号；会员基础信息；会员等级

1234567890，1997.01.01，2

promotion.txt

促销策略ID，促销策略信息，商圈／酒店ID，促销策略开始时间，促销策略结束时间，促销策略折扣，会员等级数组

123456789，双十一全网折扣，123456／12345678，2016.11.10，2016.11.12，0.8，｛1，20，40｝

order.txt

订单ID，用户ID，用户名称，用户联系方式，会员类型，订单生成时间，订单完成时间，订单状态，所在酒店，所在房间，订单预订时间段，用户入住时间（订单执行时间），用户离开时间，订单价格，信用值变化

1234567890123456, 1234567890, WoDeMa, 12345678901, MemberType.Ordinary, OrderState.UnExcuted, 12345678, 和园宾馆, 3B346, RoomType.Single, 1,10000 ,1000, 2016/10/14, 2016/10/15, 2016/10/16, 2016/10/14, 2016/10/16, 2016/10/15,2015/10/15,1,false