酒店预订系统

RSH(Reservation System of Hotel)

体系结构设计文档

**V2.0**正式版

南京大学CMBT工作组

2016-10-16

更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 吕丹  沈丽婷  王馨雨  蔚滢璐 | 2016-10-8 | 最初草稿 | V1.0 草稿 |
| 吕丹  沈丽婷  王馨雨  蔚滢璐 | 2016-10-10 | 审核汇总 | V1.1 阶段报告 |
| 吕丹  沈丽婷  王馨雨  蔚滢璐 | 2016-10-12 | 增添用户界面层的分解 | V1.2.0  阶段成果 |
| 吕丹  沈丽婷  王馨雨  蔚滢璐 | 2016-10-14 | 增添信息视角 | v1.2.2.1  阶段成果 |
| 吕丹  沈丽婷  王馨雨  蔚滢璐 | 2016-10-16 | 增添业务逻辑层和数据层的分解 | V1.2.2.2  阶段成果 |
| 吕丹  沈丽婷  王馨雨  蔚滢璐 | 2016-10-16 | 完善接口视角，排版，校对 | V2.0  正式版 |

目录

1. 引言 4
   1. 编制目的 4

1.2 词汇表 4

1.3参考资料 4

1. 产品描述 5
2. 逻辑视角 5
3. 组合视角 6
   1. 开发包图 6
   2. 运行时进程 10
   3. 物理部署 11
4. 接口视角 11
   1. 模块的职责 11
   2. 用户界面层的分解 13
      1. 用户界面层模块的职责 14
      2. 用户界面模块的接口规范 14
      3. 用户界面模块设计原理 15
   3. 业务逻辑层的分解 15
      1. 业务逻辑层模块的职责 16
      2. 业务逻辑模块的接口模范 17
   4. 数据层的分解 49
      1. 数据层模块的职责 50
      2. 数据层模块的接口规范 50
5. 信息视角 76
   1. 数据持久化对象 76
   2. 文件格式 77

**1**、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 |
| Order | 订单 |
| search | 搜索 |
| Hotel | 酒店 |
| webstaff | 网站工作人员 |
| Bl | 业务逻辑 |
| Promotion | 促销 |

**1.3** 参考资料

《软件工程与计算（卷二）软件开发的技术基础》

《酒店预订系统用例文档V1.0》

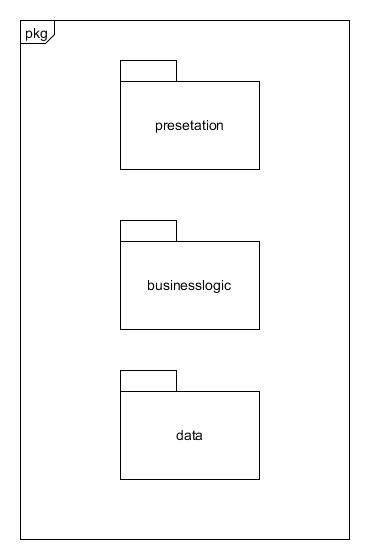
《酒店预订系统需求规格说明文档V1.0》

1. 产品描述

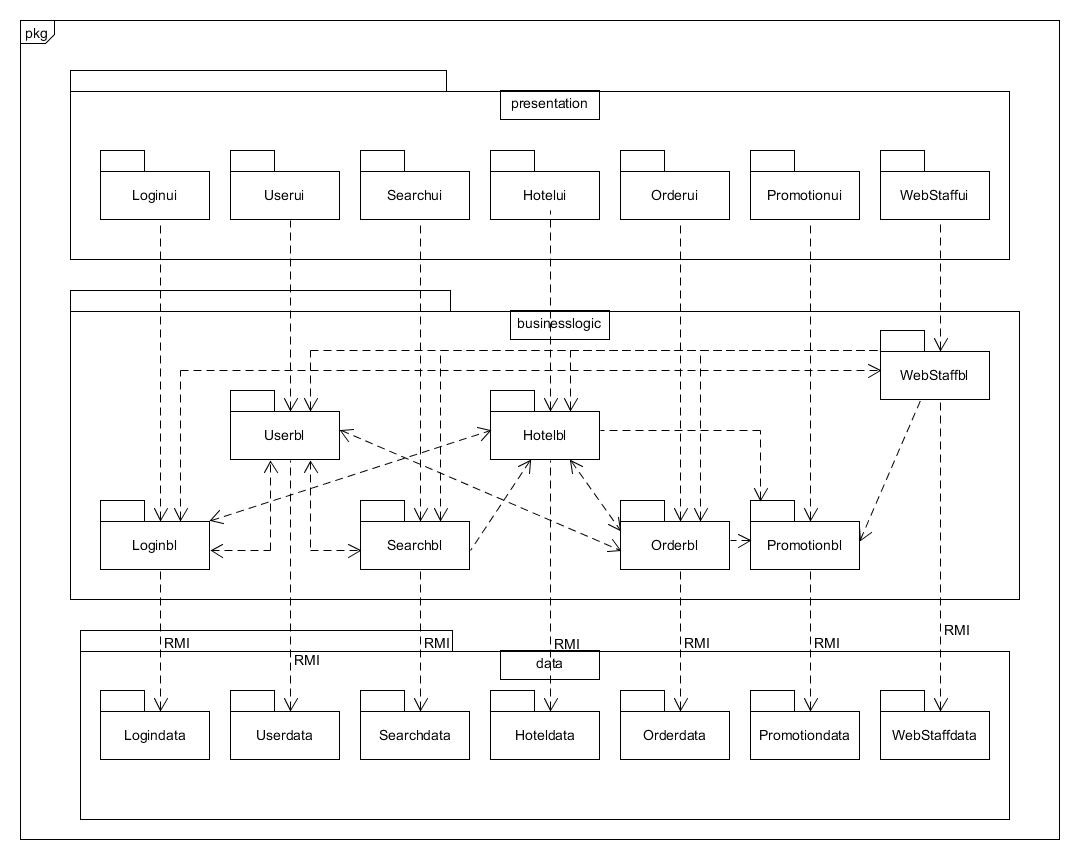
参考《酒店预订管理系统用例文档》和《酒店预订管理系统需求规格说明文档》中对产品的概括描述。

1. 逻辑视角

酒店预订管理系统中，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**：软件体系结构逻辑设计方案

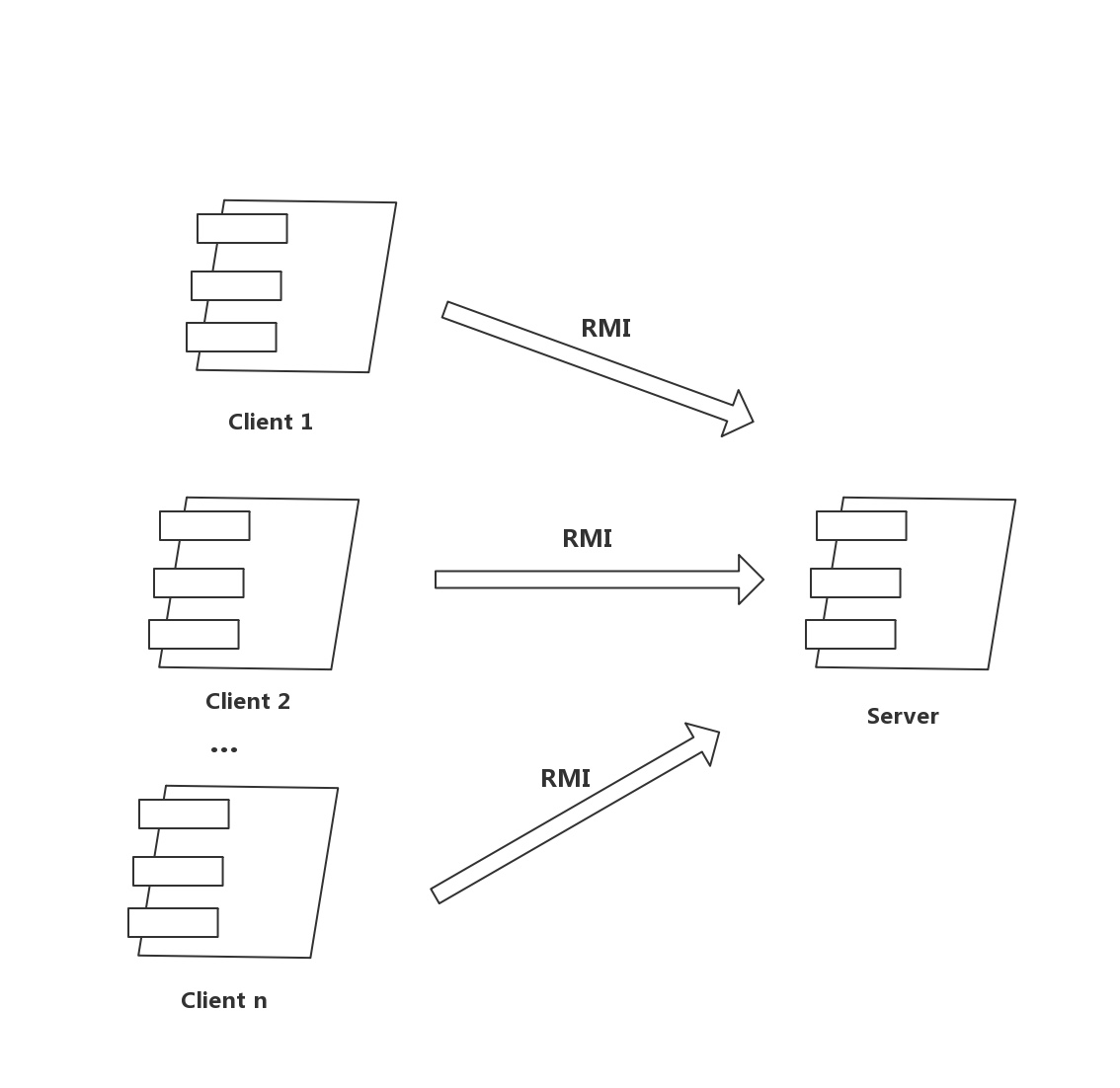
1. 组合视角

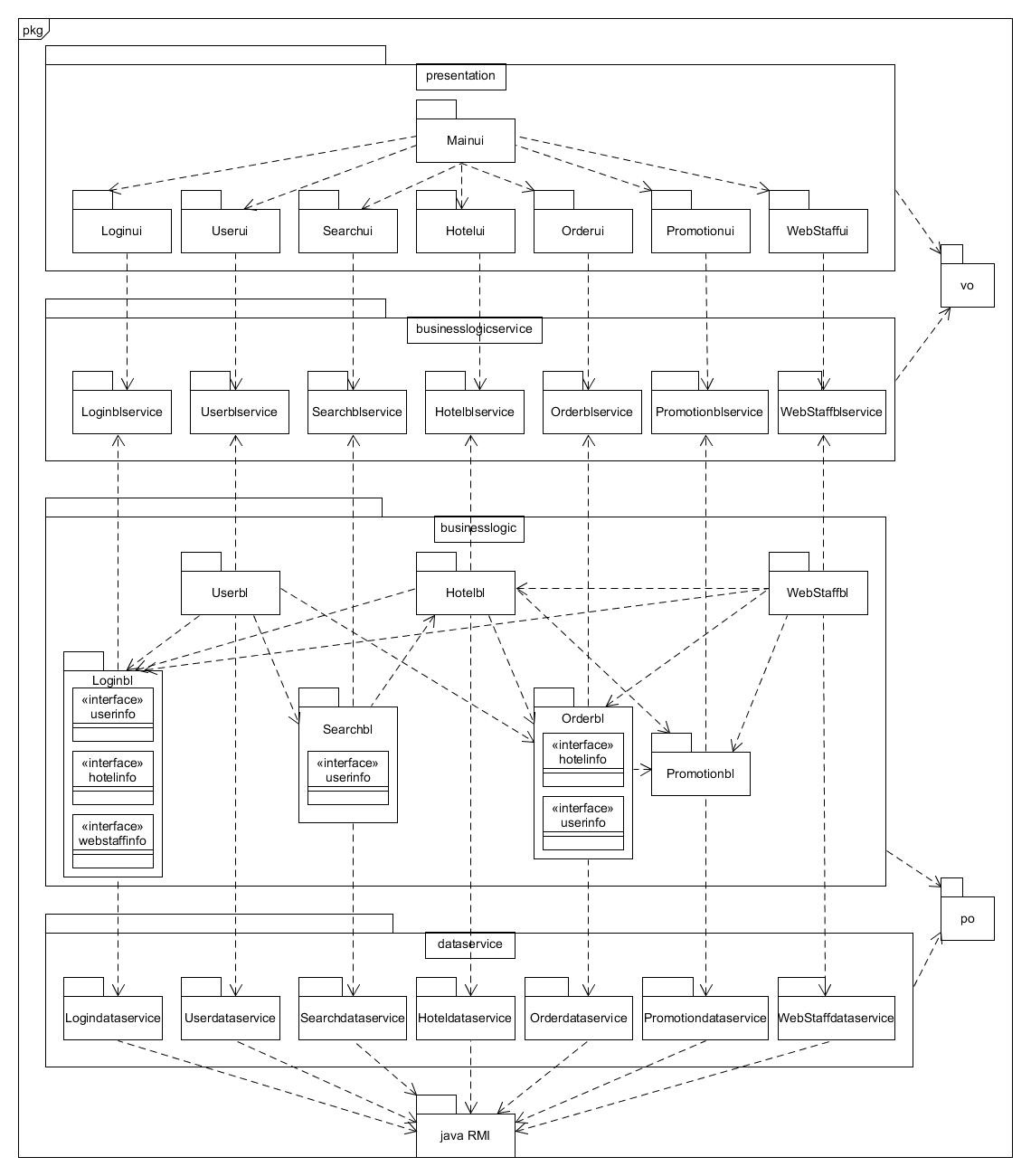
4.1开发包图

酒店预订管理系统的最终开发包设计如表1所示：

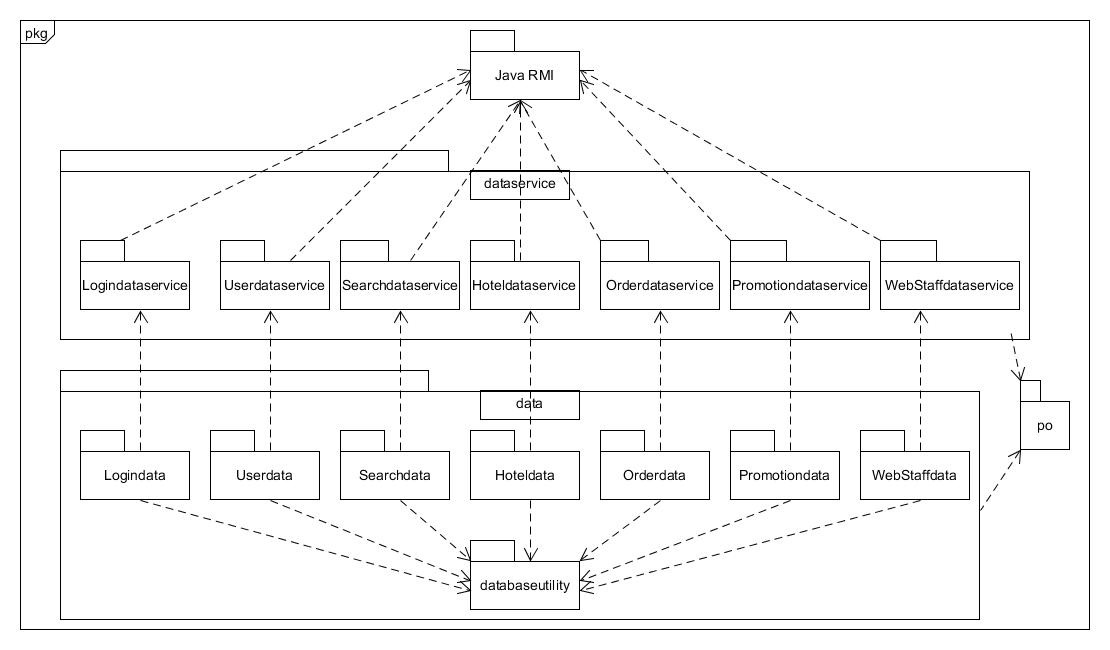
表**1**：酒店预订系统的最终开发包设计

|  |  |
| --- | --- |
| 开发（物理）包 | 依赖的其他开发包 |
| Mainui | Loginui,userui,searchui,hotelui,orderui,promotionui,webstaffui,vo |
| loginui | Loginblservice,界面类库包,vo |
| loginblservice |  |
| loginbl | Loginblservice,logindataservice,po |
| logindataservice | JavaRMI,po |
| logindata | Logindataservice,databaseutility,po |
| userui | Userblservice,界面类库包,vo |
| userblservice |  |
| userbl | Userblservice,userdataservice,po,searchbl,orderbl |
| userdataservice | JavaRMI,po |
| userdata | userdataservice,databaseutility,po |
| Hotelui | hotelblservice,界面类库包,vo |
| hotelblservice |  |
| hotelbl | hotelblservice,hoteldataservice,po,loginbl,searchbl,orderbl, promotionbl |
| hoteldataservice | JavaRMI,po |
| hoteldata | hoteldataservice,databaseutility,po |
| orderui | orderblservice,界面类库包,vo |
| orderblservice |  |
| orderbl | orderblservice,orderdataservice,po,promotionbl |
| orderdataservice | JavaRMI,po |
| orderdata | orderdataservice,databaseutility,po |
| promotionui | promotionblservice,界面类库包,vo |
| promotionblservice |  |
| promotionbl | promotionblservice,promotiondataservice,po |
| promotiondataservice | JavaRMI,po |
| promotiondata | promotiondataservice,databaseutility,po |
| webstaffui | webstaffblservice,界面类库包,vo |
| webstaffblservice |  |
| webstaffbl | webstaffblservice,webstaffdataservice,po,loginbl,orderbl,promotionbl,  userbl |
| webstaffdataservice | JavaRMI,po |
| webstaffdata | webstaffdataservice,databaseutility,po |
| Vo |  |
| po |  |
| utilitybl |  |
| 界面类库包 |  |
| Java RMI |  |
| databaseutility | JDBC |

酒店预订系统客户端开发包图如图3所示，服务器端开发包图如图4所示

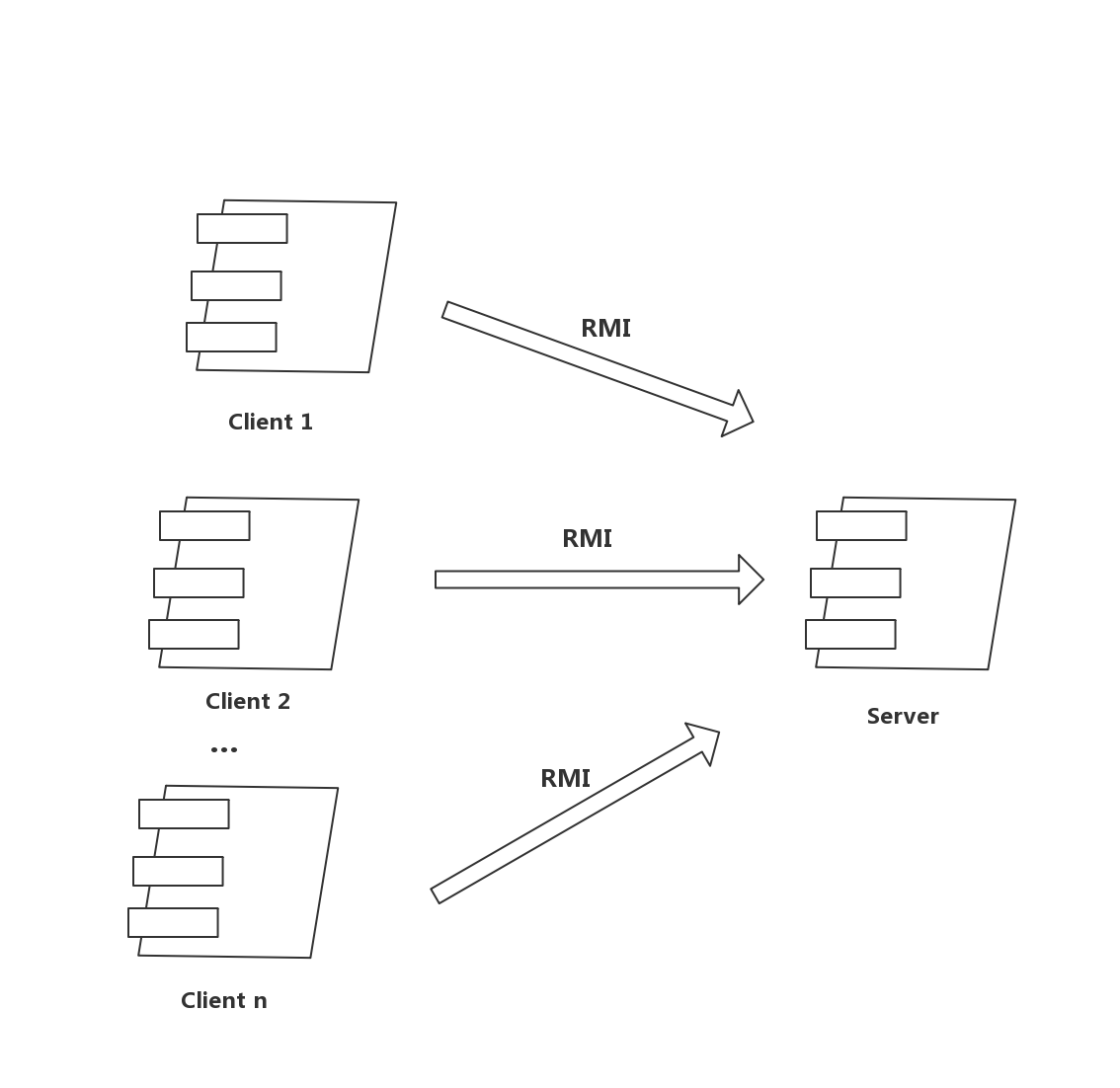


图**3**：客户端开发包图

****

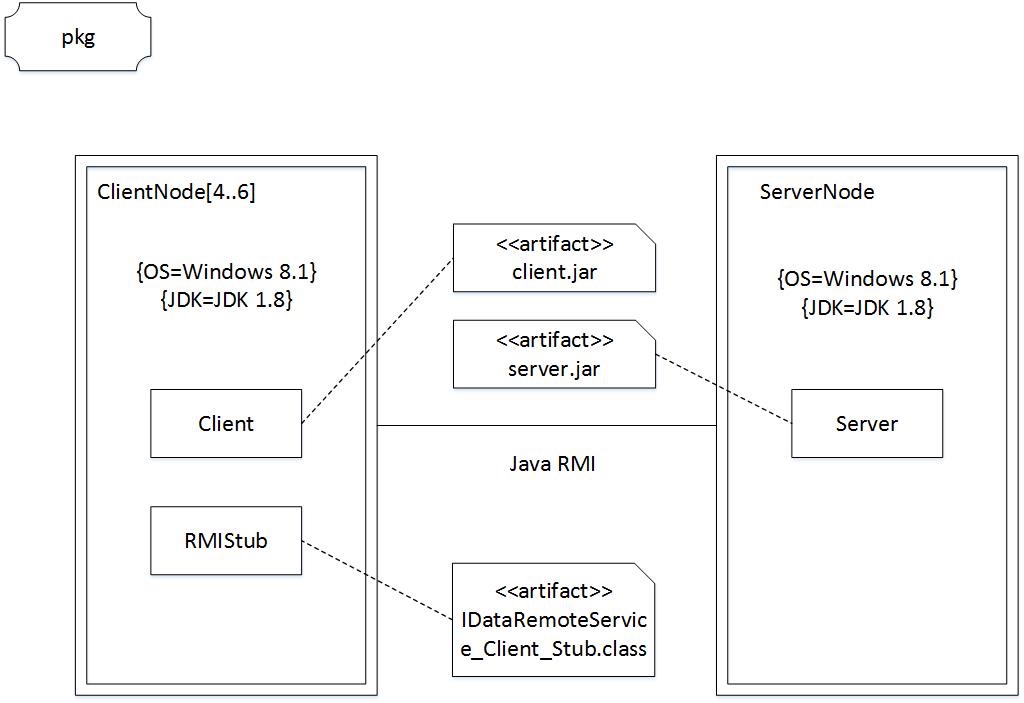
图**4**：服务器端开发包图

4.2运行时进程

在酒店预订管理系统中，会有多个客户端进程和一个服务器端进程，其进程图如图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

酒店预订管理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

**5** 接口视角

5.1 模块的职责

客户端模块视图和服务器端模块视图分别如图7和图8所示。客户端和服务器端各层的职责

分别如表2和表3所示。

启动模块

展示层

业务逻辑层

网络模块

图**7:** 客户端模块式图

启动模块

业务逻辑层

网络模块

图**8:**服务器模块式图

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 |

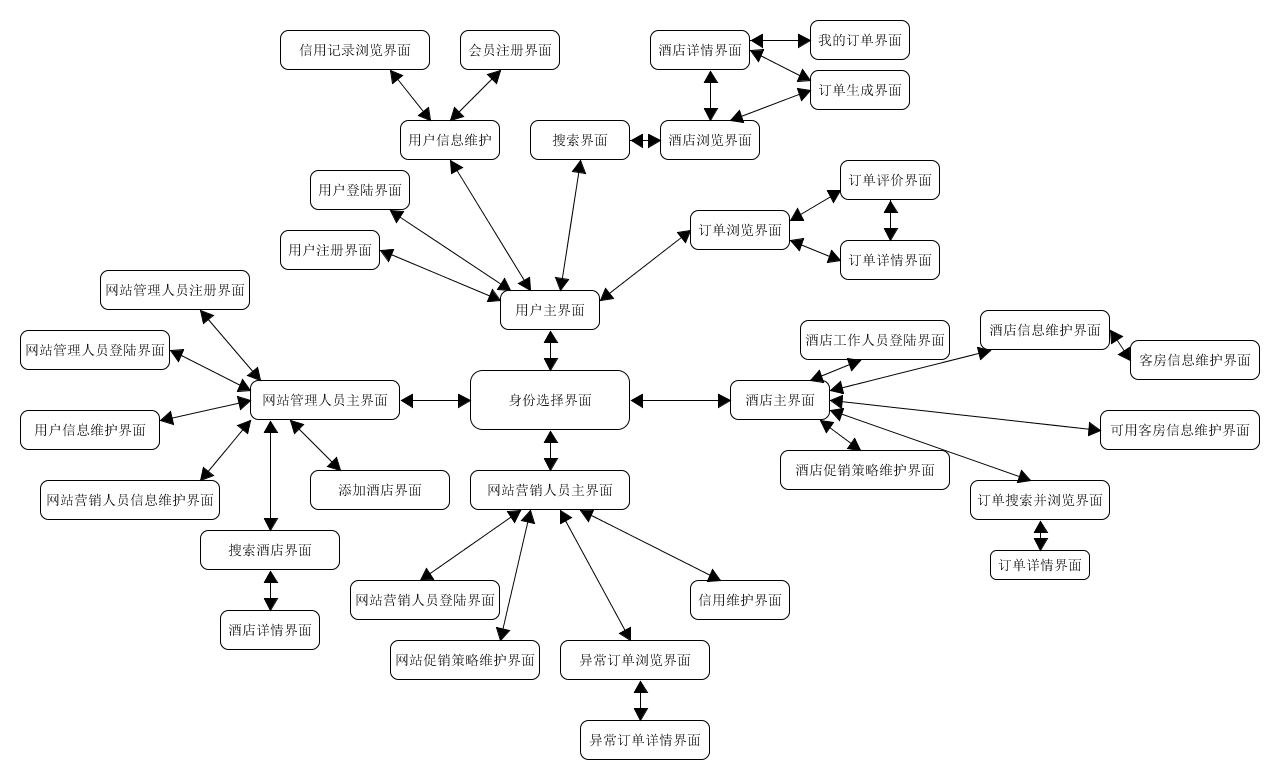
每一层只是使用下方直接接触的层。层与层之间仅仅是通过接口的调用来完成的。层之间调用的接口如表4所示。

表**4** 层之间调用的接口

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| Loginblservice  Userblservice  Hotelblservice  Orderblservice  Promotionblservice  WebStaffblservice | 客户端展示层 | 客户端业务逻辑层 |
| Logindataservice  Userdataservice  Hoteldataservice  Orderdataservice  Promotiondataservice  WebStaffdataservice | 客户端业务逻辑层 | 服务器端数据层 |

5.2 用户界面层的分解

根据需求，系统存在35个用户界面：身份选择界面、网站管理人员注册界面、网站管理人员登录界面、网站营销人员登录界面、酒店工作人员登录界面、用户注册界面、用户登录界面、用户主界面、用户信息维护、会员注册界面、信用记录浏览界面、搜索界面、酒店浏览界面、酒店详情界面、订单生成界面、订单浏览界面、订单详情界面、订单评价界面、酒店主界面、酒店信息维护界面、客房信息维护界面、可用客房信息维护界面、订单搜索并浏览界面、订单详情界面、酒店促销策略维护界面、网站营销人员主界面、网站促销策略维护界面、异常订单浏览界面、异常订单详情界面、信用维护界面、网站管理人员主界面、用户信息维护界面、酒店信息浏览界面、酒店详情界面、网站营销人员信息维护界面

****

图**9:** 酒店预订系统界面跳转图

5.2.1 用户界面层模块的职责

表**5** 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 |

5.2.1 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**

|  |  |  |
| --- | --- | --- |
| 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

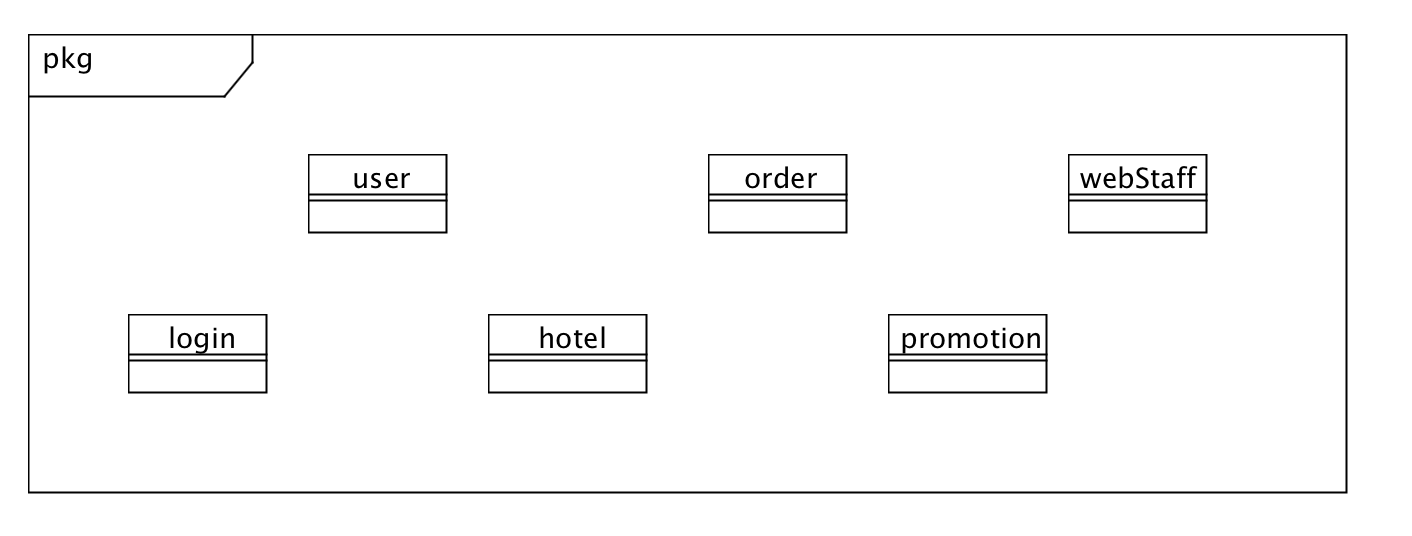
|  |  |
| --- | --- |
| 服务名 | 服务 |
| businesslogicservice.\*blservice | 每个ui模块都有一个相应的业务逻辑接口 |

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。业务逻辑层的设计如下图所示：



图**10**：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 |
| Loginbl | 实现注册账号与登录的服务，为其他包提供注册账号与登陆的接口 |
| Userbl | 实现对用户基本信息的增删改查,为其他包提供操作用户信息的接口 |
| Hotelbl | 实现对酒店自身信息和酒店工作人员信息增删改查，为其他包提供操作酒店信息的接口 |
| Orderbl | 实现对订单信息的增删改查，为其他包提供操作订单信息的接口 |
| Promotionbl | 实现对促销策略的增删改查，为其他包提供操作促销策略信息的接口 |
| WebStaffbl | 实现对网站营销人员信息的增删改查 |

5.3.2 业务逻辑模块的接口模范

5.3.2.1 login模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| login.register | 语法 | public PersonVO register(Role role) |
| 前置条件 | 人员选择身份 |
| 后置条件 | 系统增加该账号的持久化对象 |
| login.check | 语法 | public ResultMessage check(Role role, String id, String password) |
| 前置条件 | 身份确认后，输入的账号、密码符合语法规则 |
| 后置条件 | 系统显示登录验证结果。若账号与密码相匹配，系统更新在线人员信息记录和人员登录记录。 |
| login.logout | 语法 | public ResultMessage logout(Role role, String id) |
| 前置条件 | 人员已登陆并选择注销登录 |
| 后置条件 | 系统更新在线人员信息记录。 |
| 需要的接口（需接口） | | |
| 服务名 | 服务内容 | |
| user.add(UserPO userPO) | 增加单一持久化对象 | |
| webStaff.add(HotelPO hotelPO) | 增加单一持久化对象 | |
| webStaff.add(WebSalesmanPO webSalesmanPO) | 增加单一持久化对象 | |
| webStaff.add(WebManagerPO webManagerPO) | 增加单一持久化对象 | |
| logindataservice.addOnline(OnlinePersonPO onlinePersonPO) | 增加单一持久化对象 | |
| logindataservice.deleteOnline(OnlinePersonPO onlinePersonPO) | 删除单一持久化对象 | |
| logindataservice.addLogin(LoginRecordItemPO loginRecordItemPO) | 增加单一持久化对象 | |
| logindataservice.deleteLogin(LoginRecordItemPO loginRecordItemPO) | 删除单一持久化对象 | |
| logindataservice.checkOnline(String id) | 检查登录冲突 | |
| user.checkPassword(String id, String password) | 验证账号与密码是否匹配 | |
| searchHotel.findID(String id) | 通过酒店账号查找酒店信息 | |
| hotel.checkPassword(String password) | 验证账号与密码是否匹配 | |
| webStaff.checkPassword(Role role, String id, String password) | 验证账号与密码是否匹配 | |

5.3.2.2 user模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| user.add | 语法 | public ResultMessage add(UserPO userPO) |
| 前置条件 | 用户完成注册 |
| 后置条件 | 系统增加该用户持久化对象 |
| user.checkPassword | 语法 | public boolean checkPassword(String id, String password) |
| 前置条件 | 用户输入的账号、密码符合语法规则 |
| 后置条件 | 系统返回验证结果 |
| user.show | 语法 | public UserVO getInfo(String userid) |
| 前置条件 | 该账号存在 |
| 后置条件 | 返回用户值对象 |
| user.update | 语法 | public ResultMessage update(UserVO userVO) |
| 前置条件 | 无 |
| 后置条件 | 系统根据值对象更新该账号的持久化对象 |
| user.setMemberLevel | 语法 | public ResultMessage setMemberLevel(int[][] gradeWithCredit) |
| 前置条件 | 无 |
| 后置条件 | 系统更新所有账号的会员等级 |
| 需要的接口（需接口） | | |
| 服务名 | 服务内容 | |
| userdataservice.add(UserPO userPO) | 增加单一持久化对象 | |
| userdataservice.delete(String id) | 删除账号所对应的单一持久化对象 | |
| userdataservice.update(UserPO) | 更新单一持久化对象 | |
| userdataservice.check(String id) | 查看用户信息 | |
|  |  | |

5.3.2.3 hotel模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Hotel.getHotel | 语法 | Public HotelVO getHotel (); |
| 前置条件 | 该酒店存在 |
| 后置条件 | 返回该酒店的酒店基本信息 |
| Hotel.updateHotel | 语法 | Public Boolean updateHotel (HotelVO vo) |
| 前置条件 | 酒店工作人员输入的信息正确 |
| 后置条件 | 系统更新数据库中该酒店的酒店基本信息 |
| Hotel.getRoomList | 语法 | Public ArrayList<RoomVO> getRoomList(); |
| 前置条件 | 无 |
| 后置条件 | 返回该酒店的客房信息列表 |
| Hotel.updateRoomList | 语法 | Public ResultMessage updateRoomList(ArrayList<RoomVO> roomList); |
| 前置条件 | 酒店工作人员输入的数量和价格正确 |
| 后置条件 | 系统更新数据库中的该酒店的客房信息列表 |
| Hotel.addSpecialRoom | 语法 | Public ResultMessage addSpecialRoom(RoomVO vo); |
| 前置条件 | 无 |
| 后置条件 | 在可用客房数据库中增加该特色可用客房数据 |
| Hotel.deleteSpecialRoom | 语法 | Public ResultMessage deleteSpecialRoom(RoomVO vo) |
| 前置条件 | 无 |
| 后置条件 | 在可用客房数据库中删除该特色可用客房数据 |
| Hotel.getAvailableRoomList | 语法 | Public ArrayList<RoomAvailableVO> getAvailableRoomList(Date date); |
| 前置条件 | 无 |
| 后置条件 | 返回该日期下的可用客房信息列表 |
| Hotel.updateAvailableRoomList | 语法 | Public ResultMessage updateAvailableRoomList(ArrayList<RoomAvailableVO> availableRoomList); |
| 前置条件 | 用户生成订单或退房 |
| 后置条件 | 更新该日期下的可用客房信息列表 |
| Hotel.getRoomType | 语法 | Public ArrayList<String> getRoomType() |
| 前置条件 | 已经从数据库中取得该酒店的可用客房信息列表 |
| 后置条件 | 返回该酒店的所有房间类型 |
| Hotel.checkPassworld | 语法 | Public ResultMessage checkPassword(String password); |
| 前置条件 | Id已存在 |
| 后置条件 | 查找是否存在相应的id，根据输入的password返回登陆验证的结果 |
| Hotel.updateGrade | 语法 | Public ResultMessage updateGrade(double grade); |
| 前置条件 | 无 |
| 后置条件 | 更新数据库中该酒店的评分 |
| Hotel.numOfAvailRoom | 语法 | Public int numOfAvailRoom(String roomType, Date checkIn, Date checkOut); |
| 前置条件 | 该客房类型存在 |
| 后置条件 | 返回这段时间内该客房类型的可用房间数目 |
| Hotel. changeAvailRoom | 语法 | public ResultMessage changeAvailRoom(String roomType, int num,Date checkIn, Date checkOut); |
| 前置条件 | 无 |
| 后置条件 | 更新数据库中该段时间内改客房类型的可用客房数目 |
| SearchHotel.getHotelList | 语法 | Public ArrayList<HotelVO> getHotelList(String address,String businessArea); |
| 前置条件 | 该地址和商圈存在 |
| 后置条件 | 返回符合该地址和商圈的经过综合排序的酒店列表 |
| SearchHotel.Sort | 语法 | Public ArrayList<HotelVO> sort(SortBy sortBy,SortMethod sortM) |
| 前置条件 | 无 |
| 后置条件 | 返回排序后的酒店列表 |
| SearchHotel.Select | 语法 | Public ArrayList<HotelVO> select(SelectConditionVO vo) |
| 前置条件 | 已经从数据库中取得酒店列表 |
| 后置条件 | 返回筛选后的酒店列表 |
| SearchHotel.getOrderStateOfUser | 语法 | Public HotelOrderStateOfUser getOrderStateOfUser(String userID,String hotelID); |
| 前置条件 | 已经从数据库中取得酒店信息列表 |
| 后置条件 | 返回该用户在该酒店的订单状态 |
| SearchHotel.findID | 语法 | Public ResultMessage findID(String id); |
| 前置条件 | Id格式正确 |
| 后置条件 | 从数据库中查找该酒店id，返回查找的结果 |
| ManageHotel.addHotel | 语法 | Public ResultMessage addHotel(String id,String password) |
| 前置条件 | 该id不存在 |
| 后置条件 | 在酒店数据库中增加该酒店数据，在可用客房数据库中增加该酒店的可用客房数据，并进行初始化 |
| ManageHotel.deleteHotel | 语法 | Public ResultMessage deleteHotel(String id) |
| 前置条件 | 该酒店存在 |
| 后置条件 | 在酒店数据库中删除该酒店数据，在可用客房数据库中删除该酒店的可用客房数据 |
| ManageHotel.updateHotelStaff | 语法 | Public ResultMessage updateHotelStaff(String id,String password); |
| 前置条件 | 该id存在 |
| 后置条件 | 更新酒店工作人员信息 |
| ManageHotel.getNewHotelID | 语法 | Public int getNewHotelID(String address); |
| 前置条件 | 无 |
| 后置条件 | 返回符合该地址的酒店数目 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| HotelDataService.getHotel（String id） | 得到根据酒店id得到酒店单一持久化对象 | |
| HotelDataService.getHotelList(); | 得到酒店持久化对象列表 | |
| HotelDataService. updateHotel (HotelPO po) | 更新酒店单一持久化对象 | |
| HotelDataService.insertHotel(HotelPO po) | 插入酒店单一持久化对象 | |
| HotelDataService.deleteHotel(HotelPO po) | 删除酒店单一持久化对象 | |
| HotelDataService.getAvaliableRoomList(String id) | 根据id得到该酒店可用客房持久化对象列表 | |
| HotelDataService.updateAvailableRoomList(ArrayList<RoomAvailablePO> availableRoomList) | 更新可用客房持久化对象列表 | |
| HotelDataService.insertAvailableRoomList(ArrayList<RoomAvailablePO> availableRoomList) | 插入可用客房持久化对象列表 | |
| HotelDataService.insertAvailableRoom(RoomAvailablePO po) | 插入单一可用客房持久化对象 | |
| HotelDataService.deleteAvailableRoomList(ArrayList<RoomAvailablePO> availableRoomList) | 删除可用客房持久化对象列表 | |
| HotelDataService.deleteAvailableRoom(RoomAvailablePO po) | 删除单一可用客房持久化对象 | |
| HotelDataService. getHotelNum(String address) | 根据地址查找符合该地址的酒店数目 | |
| HotelDataService. numOfAvailRoom(String id,String roomType, Date checkIn, Date checkOut); | 在数据库中查找这段时间内该客房类型的可用房间数目 | |
| HotelDataService. changeAvailRoom(String id,String roomType, int num,Date checkIn, Date checkOut); | 更新数据库中该段时间内改客房类型的可用客房数目 | |
| HotelDataService.updateGrade(String id,double grade) | 更新数据库中该酒店的评分 | |
| HotelDataService.getHotelList(String address,String businessArea) | 根据地址和商圈从数据库中查找符合条件的酒店持久化对象 | |
| OrderBl.getOrderStateOfUser(String userID,String hotelID); | 从订单数据库中查找该用户在该酒店的订单状态 | |
| HotelDataService.findID(String id); | 从数据库中查找该酒店id，返回查找的结果 | |

5.3.2.4 order模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Order | 语法 | public order(String id, long userid, bool isMember, date birthday) |
| 前置条件 |  |
| 后置条件 | 生成一个订单对象 |
| Order.addRoom | 语法 | public ResultMessage addRoom(RoomType type, int num, Date checkIn, Date checkOut) |
| 前置条件 | 启动一个生成订单回合 |
| 后置条件 | 在一个生成订单回合中，增加房间预订信息 |
| Order.getTotal | 语法 | public int getTotal() |
| 前置条件 | 已添加用户信息，预订房间信息和预订数量 |
| 后置条件 | 返回已添加房间的总价 |
| Order.getPrice | 语法 | public int getPrice(RoomType) |
| 前置条件 | 已添加用户信息，相应的房间类型和数量 |
| 后置条件 | 返回某条已添加房价的价格 |
| Order.end | 语法 | public ResultMessage end() |
| 前置条件 | 完成一个生成订单回合 |
| 后置条件 | 结束此次生成订单回合，持久化更新涉及的领域类对象的数据 |
| Order.delRoom | 语法 | public ResultMessage delRoom(RoomType type, int num) |
| 前置条件 | 在一个生成订单回合中已添加该种房间预订信息且预订数量不小于要删除的数量 |
| 后置条件 | 将订单中该种房间的数量减少相应的值 |
| Order.getState | 语法 | public OrderState getState(String id) |
| 前置条件 | 已存在相应编号的订单 |
| 后置条件 | 返回该订单的状态 |
| Order.execute | 语法 | public ResultMessage execute(String id) |
| 前置条件 | 已存在相应编号且订单状态为可执行或为超过预计入住时间次日中午12时的订单 |
| 后置条件 | 将订单状态改为已执行，调用方法更新用户信用值 |
| Order.cancel | 语法 | public ResultMessage cancel(String id) |
| 前置条件 | 已存在相应编号且状态为未执行的订单 |
| 后置条件 | 将订单状态改为已撤销，调用方法更新用户信用值 |
| Order.cancelAbnormal | 语法 | public ResultMessage cancelAbnormal(String id) |
| 前置条件 | 已存在相应编号切状态未异常的订单 |
| 后置条件 | 将订单状态改为已撤销，调用方法更新用户信用值 |
| Order.checkAbnormal | 语法 | public ArrayList<Order> checkAbnormal() |
| 前置条件 | 数据库中存在状态为异常的订单 |
| 后置条件 | 返回异常订单列表 |
| Order.search | 语法 | public ArrayList<Order> search(String con) |
| 前置条件 | 已存在符合搜索条件的订单 |
| 后置条件 | 返回符合搜索条件的订单列表 |
| Order.search | 语法 | public ArrayList<Order> search(String userID) |
| 前置条件 | 改用户有预订酒店的记录 |
| 后置条件 | 返回该用户所有预订酒店的记录 |
| Order.getOrderStateOfUser | 语法 | public OrderStateOfUser getOrderStateOfUSer(String userID, String hotelID) |
| 前置条件 | 无 |
| 后置条件 | 返回用户在该酒店曾生成订单的状态 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| OrderDataService.find（String id) | | 根据ID进行查找单一持久化对象 |
| OrderDataService.finds(String field, int value ) | | 根据字段名和值进行查找多个持久化对象 |
| OrderDataService.insert(OrderPO po) | | 插入单一持久化对象 |
| OrderDataService.deletes(OrderPO po) | | 删除单一持久化对象 |
| OrderDataService.update(OrderPO po) | | 更新单一持久化对象 |
| DatabaseFactory.getOrderDatabase | | 得到Order数据库的服务的引用 |
| Hotel.updateGrade(int grade) | | 修改酒店的评分 |
| Hotel.numOfAvailRoom(RoomType type, Date checkIn, Date checkOut) | | 返回酒店特定房间类型在一定时期内的可用客房数量 |
| Hotel.changeAvailRoom(RoomType tye, int num, Date checkIn, Date checkOut) | | 修改酒店特定房间类型在一定时期内的可用客房数量 |
| Promotion.totalDiscount(String id, String date, Boolean IsBirthday, Boolean IsEnterprise, int roomNumber, int consumption)  Promotion.totalDiscount(String date, String id, int memberlevel) | | 返回满足条件的总额特价策略 |
| Promotion.roomDiscount(String id, roomtype type, String date) | | 返回满足条件的房间特价策略 |

5.3.2.5 promotion模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Promotion.show  酒店 网站管理人员 | 语法 | public PromotionList show(String id) |
| 前置条件 | 无 |
| 后置条件 | 系统返回酒店所有促销策略；系统返回所有网站促销策略 |
| Promotion.add  酒店 网站管理人员 | 语法 | public ResultMessage add(String id, int number, String name, condition con, int ratio) |
| 前置条件 | 无 |
| 后置条件 | 系统显示添加结果 |
| Promotion.update  酒店 网站管理人员 | 语法 | public ResultMessage update(String id, int number, String name, condition con, int ratio) |
| 前置条件 | 无 |
| 后置条件 | 系统显示编辑结果 |
| Promotion.del  酒店 网站管理人员 | 语法 | public ResultMessage del(String id,int number) |
| 前置条件 | 无 |
| 后置条件 | 系统显示删除结果 |
| Promotion.totalDiscount  订单 | 语法 | public int totalDiscount(String hotelid, String date, Boolean IsBirthday, Boolean IsEnterprise, int roomNumber, int consumption)  public int totalDiscount(String date, String hotelid, int memberlevel) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| Promotion.roomDiscount  订单 | 语法 | public int roomDiscount(String id, roomtype type, String date) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| Promotion.getName  订单 | 语法 | public String getName(String id, String date, Boolean IsBirthday, Boolean IsEnterprise, int roomNumber, int consumption)  public String getName(String date, String id, int memberLevel) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| Promotiondataservice.find(String hotelid)  Ptomotiondataservice.find() | 根据ID进行查找单一持久化对象;查找网站策略的单一持久化对象 | |
| Promotiondataservice.insert(PromotionPO po) | 插入单一持久化对象 | |
| Promotiondataservice.delete(PromotionPO po) | 删除单一持久化对象 | |
| Promotiondataservice.update(PromotionPO po) | 更新单一持久化对象 | |

5.3.2.6 webStaff模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| WebStaff.checkPassword  登录 | 语法 | public Boolean checkPassword(role r, String id，String password) |
| 前置条件 | 账号密码已经填写并符合规范 |
| 后置条件 | 系统显示登陆结果 |
| WebStaff.addWebSalesman | 语法 | public ResultMessage addWebSalesman(String id, String password) |
| 前置条件 | 无 |
| 后置条件 | 系统显示添加结果 |
| WebStaff.updateWebSalesman | 语法 | public ResultMessage updateWebSalesman(WebSalesmanPO po) |
| 前置条件 | 无 |
| 后置条件 | 系统显示更新结果 |
| WebStaff.delWebSalesman | 语法 | public ResultMessage delWebSalesman(String id) |
| 前置条件 | 无 |
| 后置条件 | 系统显示删除结果 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| Promotion.show() | 显示所有网站促销策略的信息 | |
| Promotion.new(String id, condition con, int ratio) | 增加一条网站促销策略 | |
| Promotion.update(String id, condition con, int ratio) | 编辑一条网站促销策略 | |
| Promotion.del(String id) | 删除一条网站促销策略 | |
| User.show(String userid) | 查看一名用户的信息 | |
| User.update (UserVO vo) | 编辑更新一名用户的信息 | |
| User.setMemberlevel(int[][] gradeWithCredit) | 设置基于信用值的会员等级信息满足条件 | |
| Hotel.deleteHotel(String id) | 删除一家酒店账号 | |
| Order. checkAbnormal() | 查看所有异常订单的信息 | |
| Order.cancelAbnomal(String orderid) | 撤销一个订单 | |
| User.creditRecharge(String userid, int value, Reason r) | 充值一名用户的信用值 | |
| WebStaffdataservice.find(int character, String id) | 根据角色和ID进行查找单一持久化对象 | |
| WebStaffdataservice.insert(WebSalesmanPO po)  WebStaffdataservice.insert(WebManagerPO po) | 插入单一持久化对象 | |
| WebStaffdataservice.delete(WebSalesmanPO po)  WebStaffdataservice.delete(WebManagerPO po) | 删除单一持久化对象 | |
| WebStaffdataservice.update(WebSalesmanPO po)  WebStaffdataservice.update(WebManagerPO po) | 更新单一持久化对象 | |

5.4 数据层的分解

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的增、删、改、查。由于持久化数据的保存可能存在多种形式：Txt文件、序列化文件、数据库等，所示抽象了数据服务。以UserDataService为例，数据层模块的具体描述下图所示：

5.4.1 数据层模块的职责

数据层模块的职责如下表所示：

|  |  |
| --- | --- |
| 模块 | 职责 |
| logindataservice | 负责保存在线人员信息记录和人员登录记录，进行增删改查的操作 |
| userdataservice | 负责保存用户基本信息，进行增删改查的操作 |
| hoteldataservice | 负责保存酒店基本信息，并进行增删改查的操作 |
| orderdataservice | 负责保存订单信息，并进行增删改查的操作 |
| promotiondataservice | 负责保存促销策略信息，并进行增删改查的操作 |
| webStaffdataservice | 负责保存网站工作人员信息，并进行增删改查的操作 |

5.4.2 数据层模块的接口规范

5.4.2.1 LoginDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| logindataservice.add | 语法 | public void add(LoginRecordItemPO loginRecordItemPO) |
| 前置条件 | 无 |
| 后置条件 | 在数据库中增加一个po记录 |
| logindataservice.delete | 语法 | public void detele(LoginRecordItemPO loginRecordItemPO) |
| 前置条件 | 数据库中含有该po |
| 后置条件 | 删除一个po |
| logindataservice.update | 语法 | public void update(LoginRecordItemPO loginRecordItemPO) |
| 前置条件 | 数据库中含有该po |
| 后置条件 | 更新一个po |
| logindataservice.check | 语法 | public LoginRecordItemPO check(String id) |
| 前置条件 | 无 |
| 后置条件 | 按id进行查找并返回LoginPO的结果 |
| logindataservice.checkOnline | 语法 | public boolean checkOnline(String id) |
| 前置条件 | 无 |
| 后置条件 | 按id进行查找并返回boolean的结果 |
| logindataservice.init | 语法 | public void init（） throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 初始化持久化数据库 |
| logindataservice.finish | 语法 | public void finish（） throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |

5.4.2.2 UserDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| userdataservice.add | 语法 | public void add(UserPO userPO) |
| 前置条件 | 同样id的po在数据库中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| userdataservice.delete | 语法 | public void detele(String id) |
| 前置条件 | 数据库中含有该id的po记录 |
| 后置条件 | 在数据库中删除该id的po记录 |
| userdataservice.update | 语法 | public void update(UserPO userPO) |
| 前置条件 | 数据库中含有该id的po记录 |
| 后置条件 | 更新一个po |
| userdataservice.check | 语法 | public UserPO check(String id) |
| 前置条件 | 无 |
| 后置条件 | 按id进行查找并返回相应的UserPO结果 |
| userdataservice.init | 语法 | public void init() throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 初始化持久化数据库 |
| userdataservice.finish | 语法 | public void finish() throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |

5.4.2.3 HotelDataService

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| HotelDataService.getHotel | 语法 | Public HotelPO getHotel(String id) |
| 前置条件 | 该酒店存在 |
| 后置条件 | 返回酒店单一持久化对象 |
| HotelDataService.getHotelList | 语法 | Public ArrayList<HotelPO> getHotelList() |
| 前置条件 | 该酒店列表存在 |
| 后置条件 | 返回酒店持久化对象列表 |
| HotelDataService. updateHotel | 语法 | Public ResultMessage updateHotel(HotelPO po) |
| 前置条件 | 无 |
| 后置条件 | 更新酒店单一持久化对象 |
| HotelDataService.insertHotel | 语法 | Public ResultMessage insertHotel(HotelPO po) |
| 前置条件 | 无 |
| 后置条件 | 插入酒店单一持久化对象 |
| HotelDataService.deleteHotel | 语法 | Public ResultMessage deleteHotel(HotelPO po) |
| 前置条件 | 该酒店存在 |
| 后置条件 | 删除酒店单一持久化对象 |
| HotelDataService.getAvaliableRoomList | 语法 | Public ArrayList<RoomAvailablePO> getAvailableRoomList(String id) |
| 前置条件 | 该客房列表存在 |
| 后置条件 | 返回可用客房持久化对象列表 |
| HotelDataService.updateAvailableRoomList | 语法 | Public ResultMessage updateAvailableRoomList(ArrayList<RoomAvailablePO> availableRoomList) |
| 前置条件 | 无 |
| 后置条件 | 更新可用客房持久化对象列表 |
| HotelDataService.insertAvailableRoomList | 语法 | Public ResultMessage insertAvailableRoomList(ArrayList<RoomAvailablePO> availableRoomList) |
| 前置条件 | 无 |
| 后置条件 | 插入可用客房持久化对象列表 |
| HotelDataService.insertAvailableRoom | 语法 | Public ResultMessage insertAvailableRoom(RoomAvailablePO po) |
| 前置条件 | 无 |
| 后置条件 | 插入单一可用客房持久化对象 |
| HotelDataService.deleteAvailableRoomList | 语法 | Public ResultMessage deleteAvailableRoomList(ArrayList<RoomAvailablePO> availableRoomList) |
| 前置条件 | 该客房列表存在 |
| 后置条件 | 删除可用客房持久化对象列表 |
| HotelDataService. getHotelNum | 语法 | Public int getHotelNum(String address); |
| 前置条件 | 无 |
| 后置条件 | 返回符合该地址的酒店数目 |
| HotelDataService. numOfAvailRoom | 语法 | Public int numOfAvailRoom(String id,String RoomType, Date checkIn, Date checkOut) |
| 前置条件 | 该客房类型存在 |
| 后置条件 | 返回这段时间内该客房类型的可用房间数目 |
| HotelDataService.changeAvailRoom | 语法 | public ResultMessage changeAvailRoom(String id,String RoomType, int num,Date checkIn, Date checkOut); |
| 前置条件 | 无 |
| 后置条件 | 更新数据库中该段时间内改客房类型的可用客房数目 |
| HotelDataService.deleteAvailableRoom | 语法 | Public ResultMessage deleteAvailableRoom(RoomAvailablePO po) |
| 前置条件 | 该客房存在 |
| 后置条件 | 删除单一可用客房持久化对象 |
| HotelDataService.updateGrade | 语法 | Public ResultMessage updateGrade(String id,double grade) |
| 前置条件 | 无 |
| 后置条件 | 更新数据库中该酒店的评分 |
| HotelDataService.findID | 语法 | Public ResultMessage findID(String id); |
| 前置条件 | 该id格式正确 |
| 后置条件 | 从数据库中查找该酒店id，返回查找的结果 |
| HotelDataService.getHotelList | 语法 | Public ArrayList<HotelPO> getHotelList(String address,String businessArea); |
| 前置条件 | 该地址和商圈存在 |
| 后置条件 | 返回符合地址和商圈的酒店持久化对象 |

5.4.2.4 OrderDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderDataService.find | 语法 | public OrderPO find(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的OrderPO结果 |
| OrderDataService.finds(String field, int value ) | 语法 | public ArrayList<OrderPO> finds(String field, int value) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按字段名和值查找返回相应的OrderPO结果 |
| OrderDataService.insert | 语法 | public void insert(OrderPO 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 |
| OrderDataService.delete | 语法 | public void delete(OrderPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 删除一个po |
| OrderDataService.update | 语法 | public void update(OderPO po) throws RemoteExcptioin |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 更新一个po |
| OrderDataService.init | 语法 | public void init() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 初始化持久化数据库 |
| OrderDataService.finish | 语法 | public void finish() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |

5.4.2.5 PromotionDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PromotionDataService.check | 语法 | public PromotionPO check(String id)throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的PromotionPO结果 |
| PromotionDataService.add | 语法 | public void add(PromotionPO po) throws RemoteException |
| 前置条件 | 相同的ID在数据库中不存在 |
| 后置条件 | 在数据库中增加一个po数据 |
| PromotionDataService.update | 语法 | public void update(PromotionPO po) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的po |
| 后置条件 | 更新一个po数据 |
| PromotionDataService.del | 语法 | public void del(PromotionPO po) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的po |
| 后置条件 | 删除一个po数据 |
| PromotionDataService.init | 语法 | public void init()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 初始化持久化数据库 |
| PromotionDataService.finish | 语法 | public void finish()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |

5.4.2.6 WebStaffDataService

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| WebStaffDataService.checkWebSalesman | 语法 | public WebSalesmanPO checkWebSalesman(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的WebSalesmanPO结果 |
| WebStaffDataService.addWebSalesman | 语法 | public void addWebSalesman(WebSalesmanPO po) throws RemoteException |
| 前置条件 | 相同的ID在数据库中不存在 |
| 后置条件 | 在数据库中增加一个po数据 |
| WebStaffDataService.updateWebSalesman | 语法 | public void updateWebSalesman(WebSalesmanPO po) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的po |
| 后置条件 | 更新一个po数据 |
| WebStaffDataService.delWebSalesman | 语法 | public void delWebSalesman(WebSalesmanPO po) throws RemoteException |
| 前置条件 | 在数据库中存在相同ID的po |
| 后置条件 | 删除一个po数据 |
| WebStaffDataService.init | 语法 | public void init()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 初始化持久化数据库 |
| WebStaffDataService.finish | 语法 | public void finish()throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |

**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，在此只做简单介绍。

UserPO类为用户信息，包括ID、密码、昵称以及个人其他信息。

持久化对象xxxx类的定义如图所示：

6.2 文件格式

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

