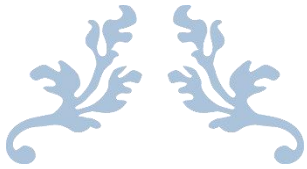


F.A.F 酒店预订系统

体系结构文档

2017-1-1

NJU F.A.F 万年杰 田原 曾虎双 陆茹茹

V1.8

目录

更新历史

更新人员	日期	变更原因	版本号
万年杰	2016.10.11	草稿	V0.0
田原	2016.10.14	客户端开发包图中的 strategybl 去掉原有的 orderInfo 接口，对调 roombl 与 hotelbl 的依赖关系	V1.0
田原	2016.10.20	在 userbl 中增加之前被忽略的“验证企业会员”这一接口，该接口由持有合作企业信息的 strategy 模块来实现； 修改之前的数据库表	V1.1
田原	2016.10.22	在 hotelbl 中的 OrderInfo 接口中增加 public ArrayList<OrderVO> getReservedOrderList(String userID)这一接口，来得到某用户所有已执行、已撤销和异常订单	V1.2
曾虎双	2016．11.04	修改 updateCheckIn, updateCheckOut, 和 UpdateStrategy 的接口规范，把 room type 和 strategyType 的类型改成枚举类，把 time 的类型从 string 改成 Date	V1.3
陆茹茹	2016.11.05	修改包名、重新分包，部分类名也有变动；将 userID 统一改成 String 型，telNum 也改成 String 型；将网站管理人员的 delete 权限删除，并修改 add 能够增加的类型	V1.4
陆茹茹	2017.01.01	修改 user 部分	V1.5
万年杰	2017.01.01	修改 order 部分	V1.6
田原	2017.01.01	修改 hotel 部分	V1.7
曾虎双	2017.01.01	修改 room、strategy 部分	V1.8

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，同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员和最终用户而编写，是了解系统的导航。

1.2 词汇表

词汇名称	词汇含义	备注

1.3 参考资料

参考酒店预订系统用例文档和软件需求规格说明文档。

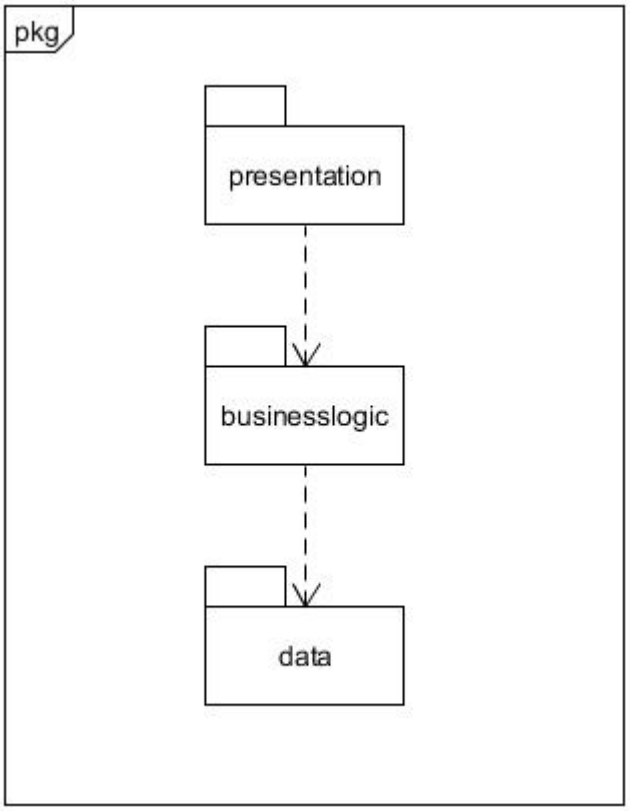
2. 产品概述

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书中对产品的概括描述。

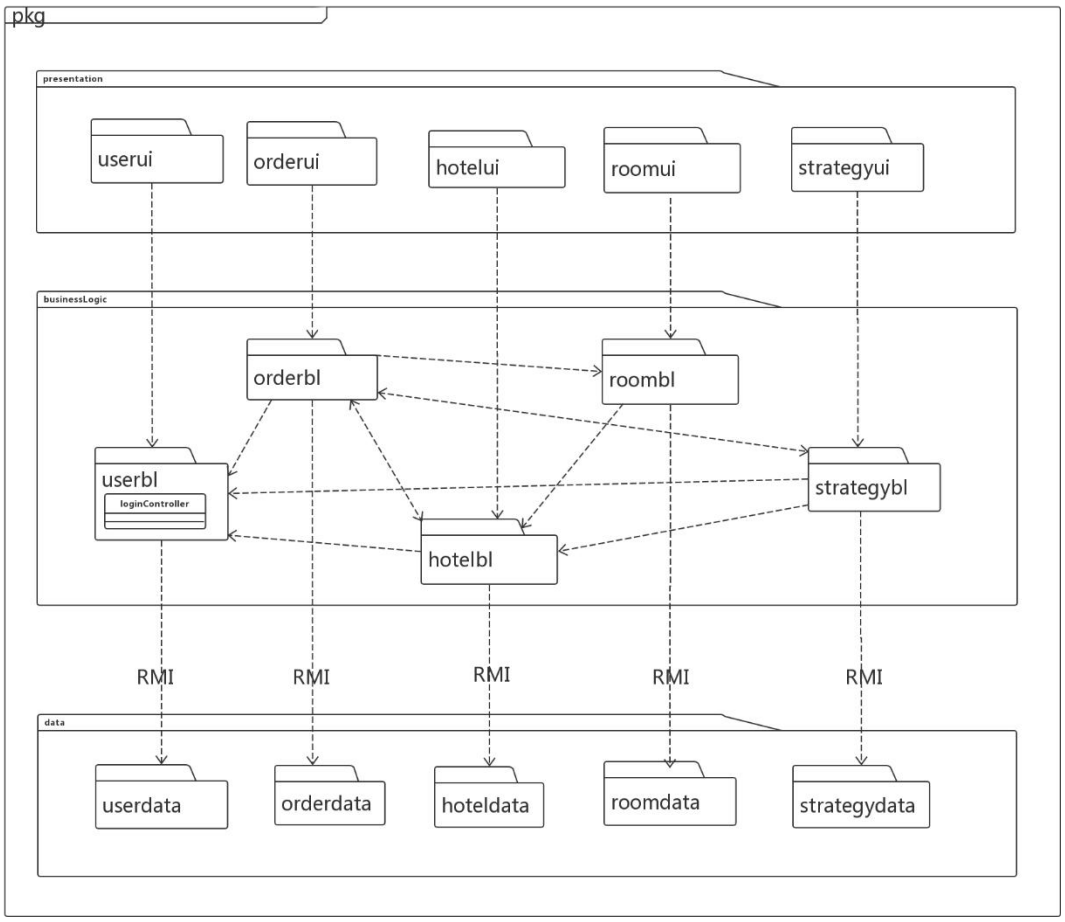
3. 逻辑视角

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。

3.1 参照体系结构风格的包图表达逻辑视角


pkg
presentation
businesslogic
data

3.2 软件体系结构逻辑设计方案



4. 组合视角

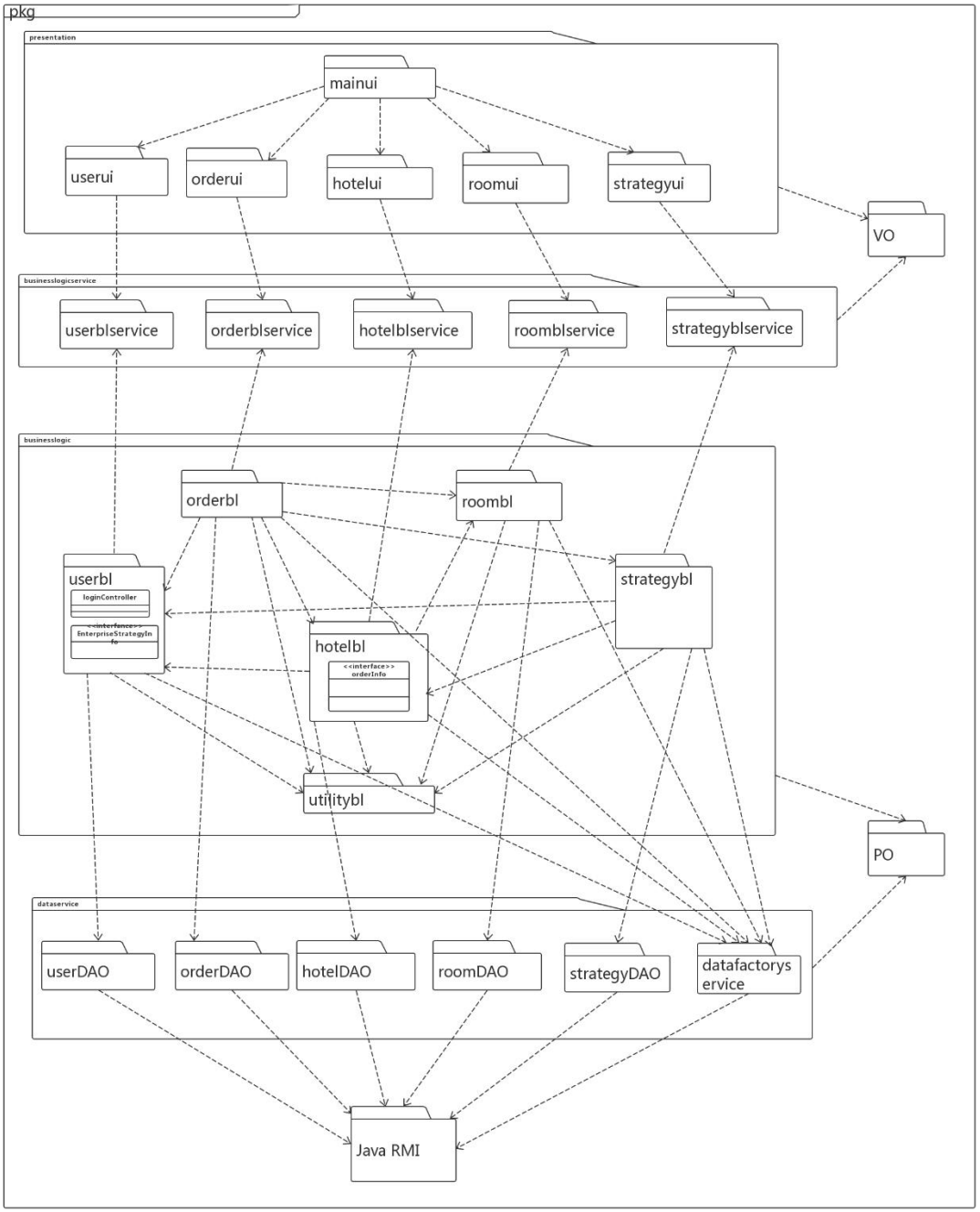
4.1 开发包图

酒店预定系统最终开发包设计

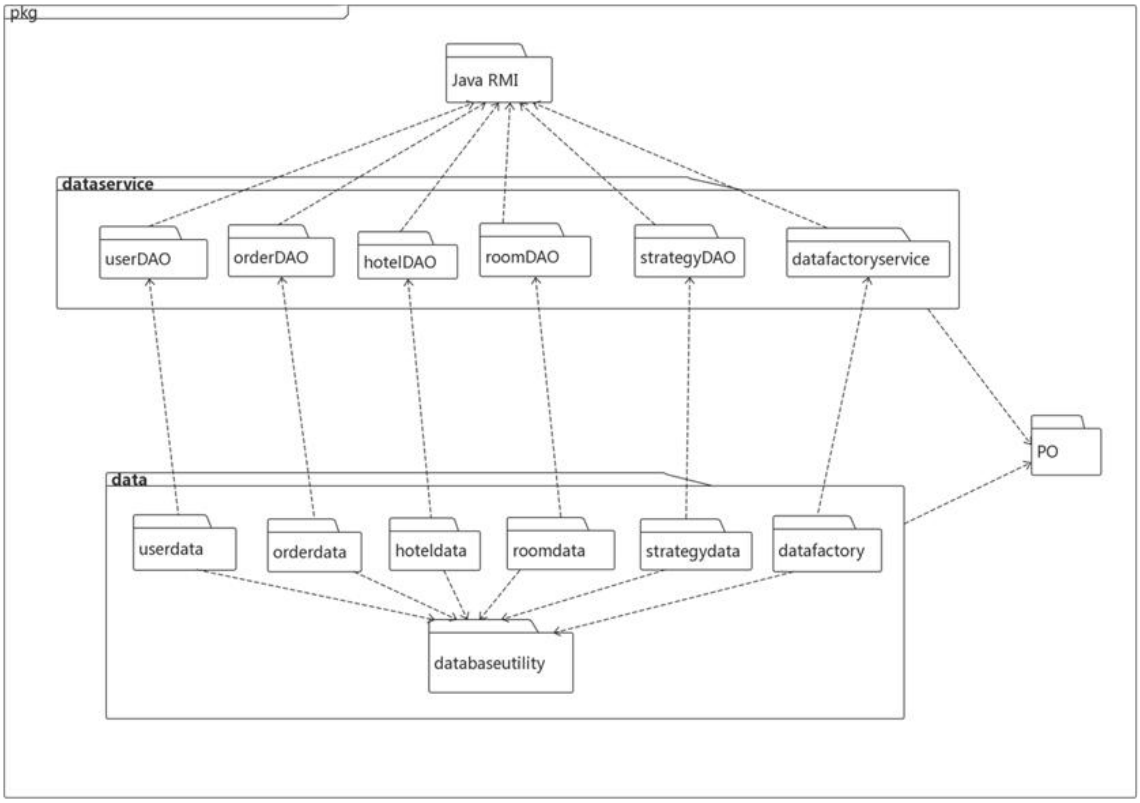
开发（物理）包	依赖的其他开发包
mainui	userui,orderui,hotelui,roomui,strategyui,vo
userui	userblservice,界面类库包
userblservice	
userbl	userblservice , userDao ,po, vo
userDao	Java RMI,po
userdata	userDao ,Java RMI,po ,databaseutility
orderui	orderblservice,界面类库包
orderblservice	

orderbl	orderblservice , orderDao ,po,vo, userbl, hotelbl, roombl, strategybl
orderDao	Java RMI,po
orderdata	orderDao ,Java RMI,po, databaseutility
hotelui	hotelblservice,界面类库包
hotelblservice	
hotelbl	hotelblservice , hotelDao ,po,vo,userbl
hotelDao	Java RMI,po
hoteldata	hotelDao ,Java RMI,po, databaseutility
roomui	roomblservice,界面类库包
roomblservice	
roombl	roomblservice , roomDao ,po,vo,hotelbl
roomDao	Java RMI,po
roomdata	roomDao ,Java RMI,po, databaseutility
strategyui	strategyblservice,界面类库包
strategyblservice	
strategybl	strategyblservice , strategyDao ,po,vo, userbl, hotelbl
strategyDao	Java RMI,po
strategydata	strategyDao ,Java RMI,po, databaseutility
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

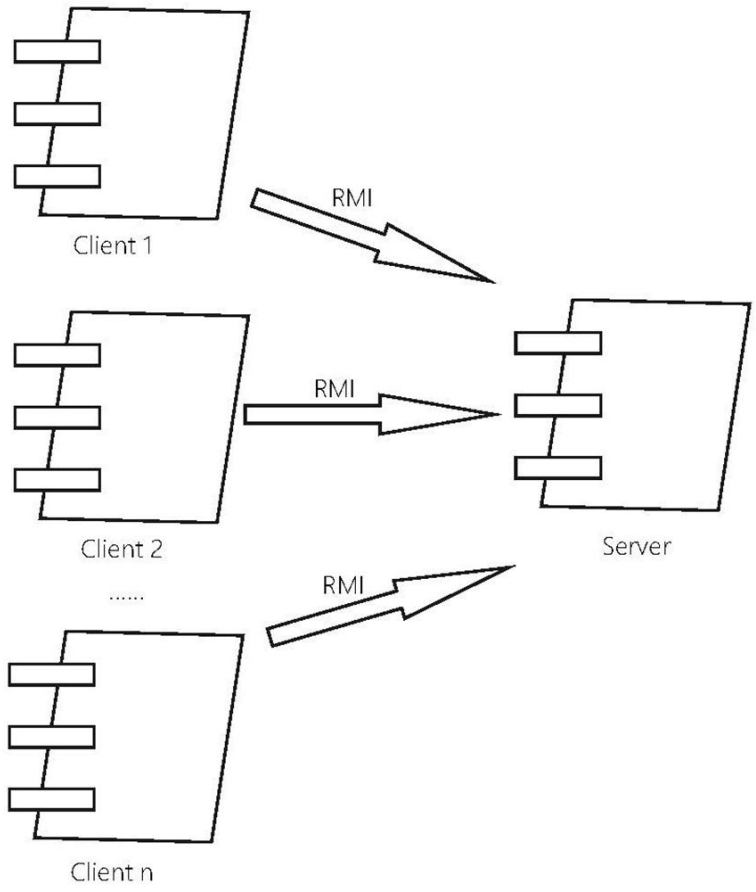


酒店预订系统服务器端开发包图



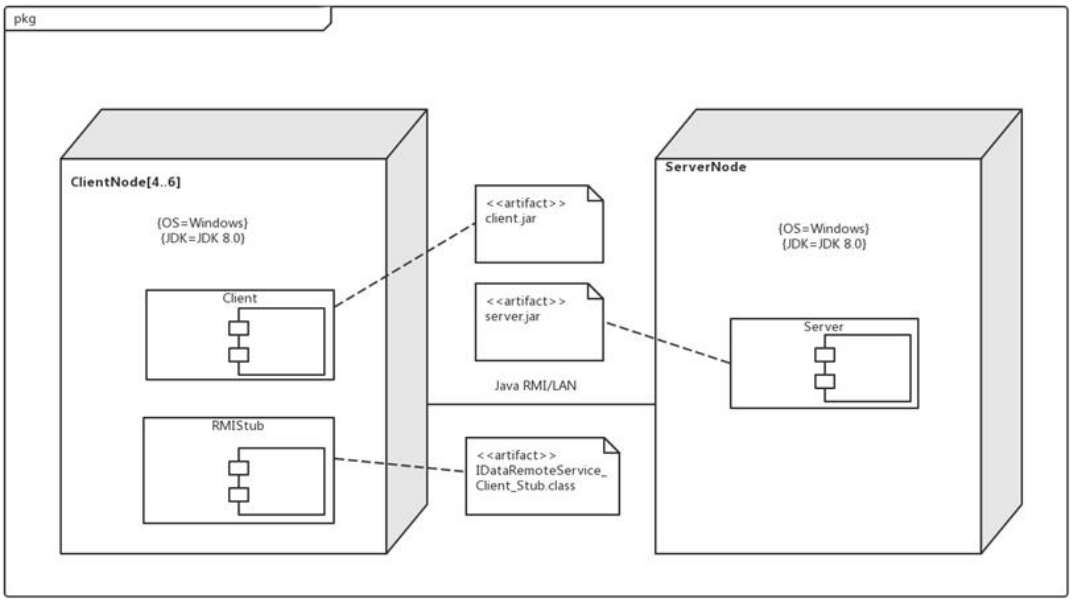
4.2 运行时进程

在酒店预订系统中，会有多个客户端进程和一个服务器端进程，其进程图如图 5 所示。结合部署图，客户端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。



4.3 物理部署

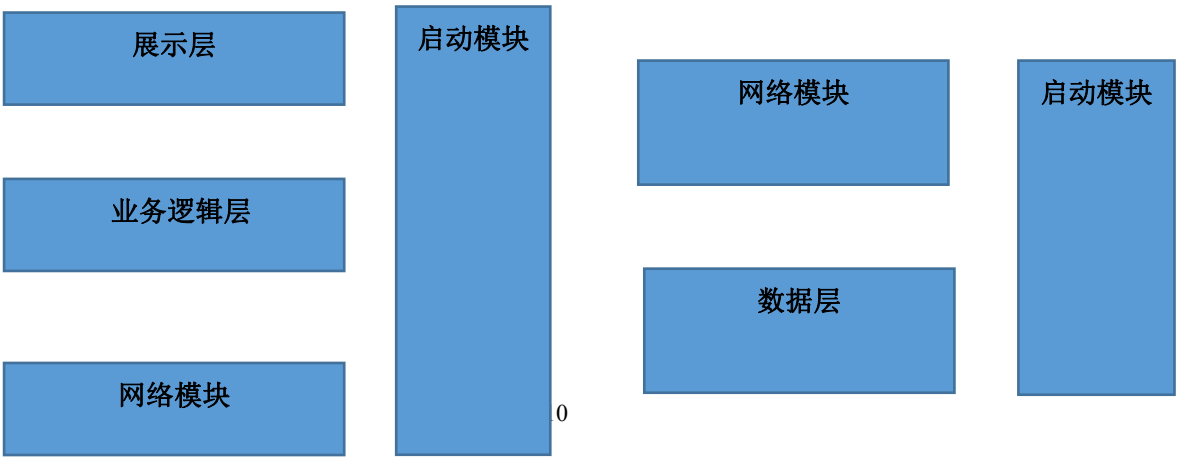
酒店预订系统中客户构件是放在客户端机器上，服务器端构件时放在服务器端机器上。在客户端节点上，还要部署**构件。由于 Java RMI 构件属于 JDk6.0 的一部分。所以，在系统 JDk 环境已经设置好的情况下，不要再独立部署。



5. 接口视角

5.1 模块的职责

客户端模块和服务器模块视图分别如图 7 和图 8 所示。客户端各层和服务器各层的职责分别如表 2 和表 3 所示。



0

客户端模块视图　　服务器模块图

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	基于窗口的快递物流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 Socket 机制实现数据通讯

服务器端各层的职责

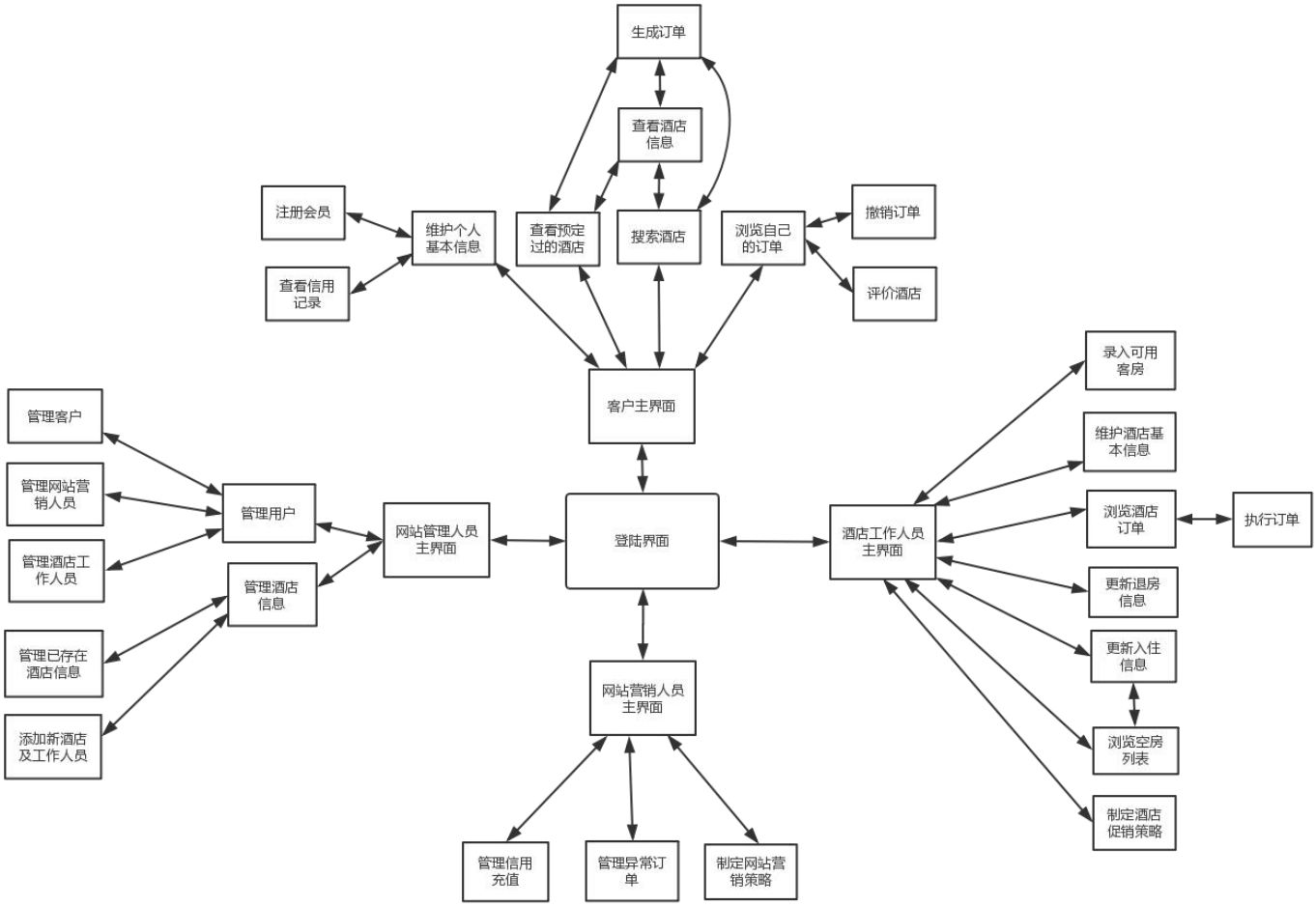
层	职责
启动模块	负责初始化网络通信机制，启动用户界面
数据层	负责数据的持久化及数据访问接口
服务器端网络模块	利用 Java Socket 机制实现数据通讯

层之间调用的接口

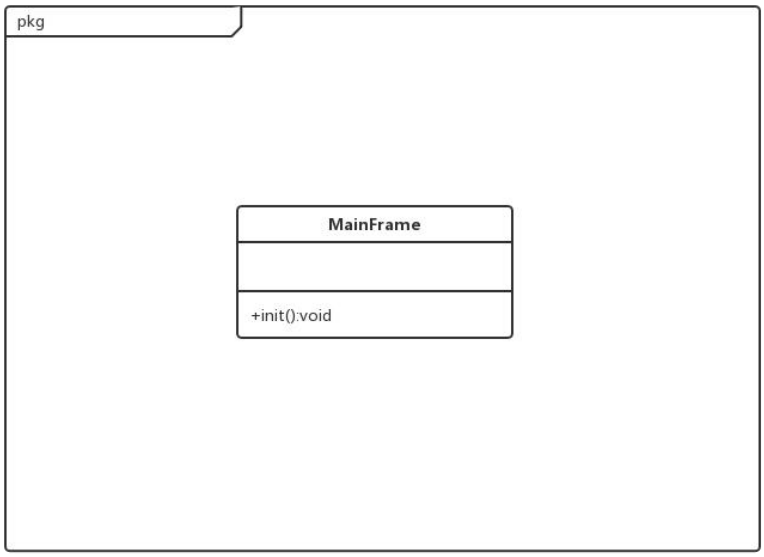
接口	服务调用方	服务提供方
OrderBLService UserBLService HotelBLService RoomBLService StrategyBLService	客户端展示层	客户端业务逻辑层
OrderDAO UserDAO HotelDAO RoomDAO StrategyDAO DatabaseFactory	客户端业务逻辑层	服务器端数据层

5.2 用户界面层的分解

根据需求，系统存在 33 个用户界面：登录界面、客户主界面、酒店工作人员主界面、网站营销人员主界面、网站管理人员主界面、维护个人基本信息界面、注册会员界面、查看信用记录界面、查看预定过的酒店界面、搜索酒店界面、查看酒店信息界面、生成订单界面、浏览自己的订单界面、撤销订单界面、评价酒店界面、录入可用客房界面、维护酒店基本信息界面、浏览酒店订单界面、执行订单界面、更新退房信息界面、更新入住信息界面、浏览空房列表界面、指定酒店促销策略界面、指定网站营销策略界面、管理信用充值界面、管理异常订单界面、管理用户界面、管理客户界面、管理网站营销人员界面、管理酒店工作人员界面、管理酒店信息界面、管理已存在酒店信息界面、添加新酒店及工作人员界面。



服务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。用户界面类如图所示。



5.2.1 用户界面层模块的职责

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

5.2.2 用户界面层模块的接口规范

用户界面层模块的接口规范

MainFrame	语法	Init(args:String[])
	前置条件	无
	后置条件	显示 Frame 以及 LoginPanel

用户界面层模块需要的服务接口

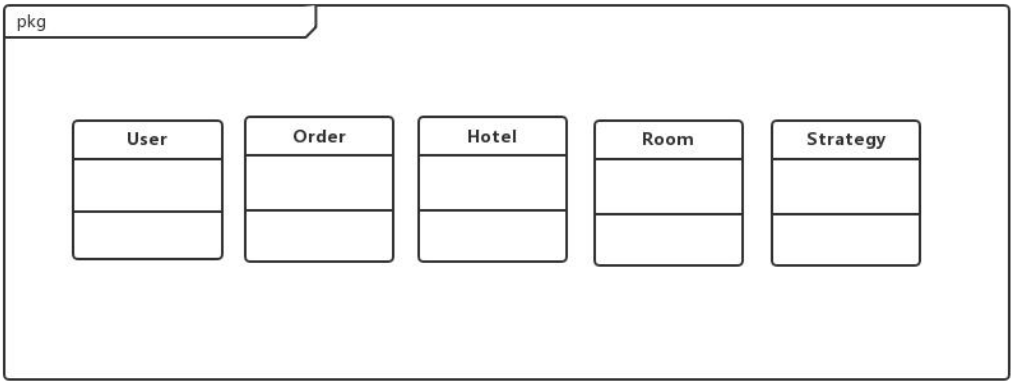
服务名	服务
businessloicservice.LoginBLService	登录界面的业务逻辑接口
Businesslogcservice.*BLService	每个界面都有一个相应的业务逻辑接口

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

模块	职责
userbl	负责实现与用户相关的服务
orderbl	负责实现与订单信息及处理相关的服务
hotelbl	负责实现酒店界面所需要的服务
roombl	实现与更新空房信息，更新入住和退房信息相关的服务
strategybl	实现与更新酒店促销策略和网站营销策略相关的服务

5.3.2 业务逻辑层模块的接口规范

5.3.2.1 userbl 模块的接口规范

提供的服务（供接口）		
LoginAndSignUpService.login	语法	public boolean login(String userID, String password)
	前置条件	ID 和 password 符合输入规范
	后置条件	查找是否存在相应的 ID，根据输入的 password 返回输入的结果
LoginAndSignUpService.add	语法	public boolean add(UserVO user)
	前置条件	用户注册账号
	后置条件	返回是否注册成功，若成功则返回 true，否则返回 false
ModifyClientInfoService.modifyUserInfo	语法	public boolean modifyUserInfo(UserVO user, String oldUserID)
	前置条件	客户修改用户信息
	后置条件	返回是否修改成功
ModifyClientInfoService.getClientInfo	语法	public ClientInfoVO getClientInfo(String userID)
	前置条件	客户查看用户信息
	后置条件	返回该客户信息
ManageUserInfoService.add	语法	public boolean add(UserVO user)
	前置条件	网站管理人员添加网站营销人员
	后置条件	返回是否添加成功
ManageUserInfoService.getHotelStaffInfo	语法	public HotelStaffInfoVO getHotelStaffInfo(String userID)
	前置条件	网站管理人员得到酒店工作人员信息
	后置条件	返回酒店工作人员信息
ManageUserInfoService.modifyUserInfo	语法	public boolean modifyUserInfo(UserVO user, String oldUserID)
	前置条件	网站管理人员修改用户信息
	后置条件	返回是否修改成功
ManageUserInfoServic	语法	public UserVO getUserInfo(String userID)

e.getUserInfo	前置条件	网站管理人员获取用户信息
	后置条件	返回该客户信息
SignVipService.signEnterpriseVip	语法	public boolean signEnterpriseVip(EnterpriseVO enterpriseVip)
	前置条件	客户申请企业会员
	后置条件	返回是否申请成功
SignVipService.signRegularVip	语法	public boolean signRegularVip(VipInfoVO regularVip)
	前置条件	客户申请企业会员
	后置条件	返回是否申请成功
QueryClientCreditRecordService.queryCreditRecord	语法	public ArrayList<ClientCreditRecordVO> queryCreditRecord(String userID)
	前置条件	userID 符合输入规范
	后置条件	返回该客户的信用记录列表
AddCreditValueService.addCreditValue	语法	public UserVO addCreditValue(String userID,int creditAdded)
	前置条件	userID 和 creditAdded 符合输入规范
	后置条件	返回是否充值成功
UserInfo.insert	语法	public boolean insert(HotelStaaaInfoVO staff)
	前置条件	网站管理人员添加酒店工作人员账号
	后置条件	返回是否添加成功
ClientCreditInfo.getCreditValue	语法	public int getCreditValue(String userID)
	前置条件	ID 符合输入规范
	后置条件	返回客户的当前信用值
ClientCreditInfo.changeCreditValue	语法	public boolean changeCreditValue(String userID,int num,String orderID, ActionType actionType)
	前置条件	ID 和 num 符合输入规范
	后置条件	返回是否更改信用值成功

VerifyEnterpriseMember.verifyEnterpriseMember	语法	public boolean verifyEnterpriseMember(String enterpriseName, String securityCode)
	前置条件	EnterpriseID 和 securityCode 符合输入规范
	后置条件	返回是否注册成功
VipInfo.getRegularVipInfo	语法	public VipInfoVO getRegularVipInfo(String userID)
	前置条件	userID 符合输入规范
	后置条件	返回普通会员的信息
VipInfo.getEnterpriseVipInfo	语法	public VipInfoVO getEnterpriseVipInfo(String userID)
	前置条件	userID 符合输入规范
	后置条件	返回企业会员的信息
需要的服务（需接口）		
服务名	服务	
DatabaseFactory.getUserDatabase	得到 user 数据库的服务的引用	
EnterpriseInfo.verifyEnterpriseMember(String hotelName, String enterpriseName, String securityCode)	验证企业会员注册的信息	
userDAO.getUserInfo(String userID)	得到 ID 对应的用户信息	
userDAO.getClientInfo(String userID)	得到 ID 对应的客户信息	
userDAO.getHotelStaffInfo(String userID)	得到 ID 对应的酒店工作人员信息	
userDAO.queryCredit（String userID)	得到 ID 对应的用户信用值及信用值变化记录	
UserDAO.getCreditValue(String userID)	得到 ID 对应的用户的当前信用值	
userDAO.insertUser(UserPO userPO)	插入单一持久化对象	
userDAO.insertClient(ClientInfoPO client)	插入单一持久化对象	
userDAO.insertHotelStaff(HotelStaffInfoPO hotelStaff)	插入单一持久化对象	

userDAO.updateUser(UserPO modified)	更新单一持久化对象
userDAO.updateClient(ClientInfoPO modified)	更新单一持久化对象
userDAO.signRegularVip(RegularVipPO regularVip)	插入单一持久化对象
userDAO. signEnterpriseVip(EnterpriseVipPO enterpriseVip)	插入单一持久化对象
userDAO. getRegularVipInfo(String userID)	得到 ID 对应的普通会员信息
userDAO. getEnterpriseVipInfo(String userID)	得到 ID 对应的企业会员信息
userDAO. updateRegularVipInfo(RegularVipPO modified)	更新单一持久化对象

5.3.2.2 orderbl 模块的接口规范

提供的服务（供接口）		
OrderInfo.getAllOrders	语法	public ArrayList<OrderVO> getAllOrders(String userID);
	前置条件	ID 有效
	后置条件	返回对应客户的所有订单的 VO，否则返回 null
OrderInfo.isReserved	语法	public boolean isReserved(String userID, String address);
	前置条件	ID 和 address 有效
	后置条件	返回该酒店是否被该用户预定过
OrderInfo.getCommentableOrderList	语法	public ArrayList<OrderVO> getCommentableOrderList (String userID);
	前置条件	ID 有效
	后置条件	返回当前用户可评价酒店对应的订单列表

OrderInfo.getOrderList	语法	public ArrayList<OrderVO> getOrderList(String userID, String address);
	前置条件	ID 和 address 有效
	后置条件	返回该用户在该酒店的所有订单
OderInfo.getReservedOrderList	语法	public ArrayList<OrderVO> getReservedOrderList(String userID);
	前置条件	userID 符合输入规范
	后置条件	返回该用户所有已执行、已撤销和异常订单
OderInfo. setOrderCommented	语法	public boolean setOrderCommented(String orderID);
	前置条件	orderID 有效
	后置条件	将对应的订单置为已评价订单，若置成功则返回 true，反之则返回 false
Order.getDetailedOrder	语法	public OrderVO getDetailedOrder(String orderID);
	前置条件	客户请求展开订单详情
	后置条件	检测订单号是否存在，如果存在，则返回对应订单的 VO，否则返回 null
Order.getSingleOrder	语法	public OrderVO getSingleOrder(String address, String orderID);
	前置条件	酒店工作人员输入订单号
	后置条件	检测该订单号是否存在，如果存在，检测是否是该酒店的订单，若是则返回对应订单的 VO，否则返回 null
Order.getHotelOrderList	语法	public ArrayList<BriefOrderInfoVO> getHotelOrderList (String address, Enum<OrderType> orderType);
	前置条件	酒店工作人员启动浏览订单流程
	后置条件	按 orderType 的值进行查找并按生成时间顺序返回相应的订单列表
Order.getUserOrderList	语法	public ArrayList<BriefOrderInfoVO> getUserOrderList (String userID, Enum<OrderType> orderType);
	前置条件	客户启动浏览订单流程
	后置条件	按 orderType 的值进行查找并按生成时间顺序返回相应的订单列表
Order.getAbnormalOrderList	语法	public ArrayList<BriefOrderInfoVO> getAbnormalOrderList (Date date);
	前置条	网站营销人员发起浏览每日异常订单情况操作

	件	
	后置条件	返回所查询日期当天所有异常订单列表
Order.initNewOrder	语法	public OrderVO initNewOrder(String userID, String address);
	前置条件	客户确认即将生成订单的酒店信息
	后置条件	初始化订单信息，返回一个被初始化的 OrderVO
Order.checkNewOrder	语法	public ResultMessage checkNewOrder(OrderVO vo);
	前置条件	客户确认生成订单
	后置条件	检查订单要求是否可以得到满足，格式是否正确，并返回结果
Order.addNewOrder	语法	public ResultMessage addNewOrder(OrderVO vo);
	前置条件	订单要求可以得到满足
	后置条件	增加新的订单
Order.getPrice	语法	public int getPrice(OrderVO vo);
	前置条件	用户已填写房间类型和房间数量
	后置条件	计算并返回订单价格
Order.getOriginalPrice	语法	public int getOriginalPrice(String hotelAddress, RoomType roomType);
	前置条件	用户选择生成订单
	后置条件	返回对应酒店各个房型对应的原始价格
Order.delayCheckIn	语法	public boolean delayCheckIn (OrderVO vo, int hours);
	前置条件	酒店工作人员确定延迟入住
	后置条件	延迟最晚入住时间，返回是否成功
Order.getOrderDone	语法	public boolean getOrderDone(OrderVO vo);
	前置条件	酒店工作人员确定执行订单
	后置条件	更新当前订单状态为已执行订单，并为客户增加与订单金额等值的信用值
Order.withdrawOrder	语法	public boolean withdrawOrder(OrderVO vo);
	前置条件	客户确认撤销订单

	后置条件	（扣除信用值），系统将订单置为撤销状态，记录撤销时间，更新空房列表
Order.systemWithdrawOrder	语法	public boolean systemWithdrawOrder(OrderVO vo, boolean isRecoverHalfCredit);
	前置条件	网站营销人员确认撤销订单
	后置条件	系统撤销此异常订单并将其状态置为已撤销、记录撤销时间，并提示选择恢复此客户信用值的全部或一半
Order.getLatestDoneTime	语法	public Time getLatestDoneTime(OrderVO vo);
	前置条件	客户发起撤销订单动作
	后置条件	返回当前订单的最晚执行时间
需要的服务（需接口）		
服务名	服务	
OrderDAO.getUserOrderList(String userID, Enum<OrderType> orderType)	根据 orderType 的值得到当前用户的对应订单持久化对象列表	
orderDAO.getUserOrdersByHotel(String userID, String address)	返回该客户在该酒店的所有订单列表	
OrderDAO.getUserAllOrders(String userID)	返回对应客户的所有订单的 PO，否则返回 null	
OrderDAO.insertOrder(OrderPO po)	插入单一持久化对象	
OrderDAO.deleteOrder(OrderPO po)	删除单一持久化对象	
OrderDAO.updateOrder(OrderPO po)	更新单一持久化对象	
OrderDAO.getHotelOrderList(String address, Enum<OrderType> orderType)	得到按 orderType 的值进行查找并按时间顺序返回该酒店的相应订单列表	
OrderDAO.getAllAbnormalList(Date date)	得到查询日期当天所有异常订单列表	
OrderDAO.getSingleOrder(String address, String orderID)	检测该酒店的订单号是否存在，如果存在，则返回对应订单的 VO，否则返回 null	
OrderDAO.getDetailedOrder(String orderID)	检测该订单号是否存在，如果存在，则返回对应订单的 VO，否则返回 null	
OrderDAO.isReserved(	返回该酒店是否被该用户预定过	

String userID, String address);	
OrderDAO.getCommentableOrders (String userID);	返回当前用户可评价酒店对应的订单列表
OrderDAO. setOrderCommented(String orderID)	将对应的订单置为已评价订单
RoomInfoService.getAvailableRoomNum(String address, Enum<RoomType> roomType)	获取当前酒店 roomType 房型的空房数量
RoomInfoService.isTimeAvailable (String address, Enum<RoomType> roomType, Date beginDate, Date finishDate)	查询用户预定时间段有无用户要求的房型剩余
RoomInfoService.addSpareRoom(String address, int change, Enum<RoomType> roomType)	更新（增加）当前酒店被预定房型的可用房间数据
RoomInfoService.reduceSpareRoom(String address , int change, Enum<RoomType> roomType)	更新（减少）当前酒店被预定房型的可用房间数据
RoomInfoService.checkOrder(OrderVO vo)	检查该订单中的信息当前能否满足
HotelInfoService.getHotelBriefInfo(String address)	得到当前酒店的简要信息
HotelInfoService.getHotelDetails(String address)	得到当前酒店的详细信息
HotelInfoService. getOriginalPrice(String hotelAddress, RoomType roomType)	得到对应酒店各个房型对应
ClientCreditInfo.getCreditValue(String userID)	得到当前登陆客户的信用值

ClientCreditInfo.changeCreditValue(String userID, int num)	根据 num(可正可负)的值，增加或减少当前客户的信用值
StrategyInfoService.getAvailblePromotionName(OrderVO vo)	得到当前最优酒店优惠策略的名称
StrategyInfoService.getAvailbleMarketStrategyName(OrderVO vo)	得到当前最优网站优惠策略的名称
StrategyInfoService.getBestDiscount(OrderVO vo)	得到当前最终的最优的优惠折扣

5.3.2.3 hotelbl 模块的接口规范

提供的服务（供接口）		
CheckOrderedHotelSerivice.enrollHotelBriefInfoList	语法	public ArrayList<HotelBriefInfoVO> enrollHotelBriefInfoList (String userID) throws RemoteException;
	前置条件	启动一个浏览预订过的酒店请求
	后置条件	返回当前用户所有预订过的酒店简要信息列表
SearchHotelService.getHotelBriefInfoListBySearching	语法	public ArrayList<OrderedHotelInfoVO> getHotelBriefInfoListBySearching (String[] condition) throws RemoteException;
	前置条件	condition 符合输入规范
	后置条件	查找是否存在符合该输入条件的酒店，如果存在则返回所有满足条件的酒店简要信息列表
SearchHotelService.getBusinessDistrictList	语法	public ArrayList<BusinessDistrictPO> getBusinessDistrictList(String city) throws RemoteException;
	前置条件	city 不为空
	后置条件	返回该城市所有的商圈列表
QueryHotelService.getHotelBriefInfoListByQuerying	语法	public ArrayList<OrderedHotelInfoVO> getHotelBriefInfoListByQuerying (String[] condition) throws RemoteException;
	前置条件	condition 符合输入规范
	后置条	查找是否存在符合该输入条件的酒店，如果存在则返回

	件	所有满足条件的酒店简要信息列表
QueryHotelSevice.getHotelDetails	语法	public HotelVO getHotelDetails(String address) throws RemoteException;
	前置条件	address 符合输入规范
	后置条件	返回在该地址处的酒店的详细信息
QueryHotelService.getOrders	语法	public ArrayList<OrderVO> getOrders(String address, String userID) throws RemoteException;
	前置条件	address 和 ID 符合输入规范
	后置条件	返回该用户在该酒店的所有订单
CommentOnHotelService.getCommentableOrderList	语法	public ArrayList<OrderVO> getCommentableOrderList(String userID) throws RemoteException;
	前置条件	启动一个评价酒店请求
	后置条件	返回可以评价的酒店对应的已执行订单列表
CommentOnHotelService.confirmComment	语法	public boolean confirmComment(String username, float mark, String comment, String hotelAddress, String orderID) throws RemoteException;
	前置条件	mark 和 comment 符合输入规范
	后置条件	返回是否评论成功
MaintainHotelBasicInfoService.enrollHotelBasicInfo	语法	public HotelVO enrollHotelBasicInfo(String address) throws RemoteException;
	前置条件	启动一个修改或浏览酒店的基本信息的请求
	后置条件	返回该酒店的基本信息
MaintainHotelBasicInfoService.confirmModify	语法	public boolean confirmModify(HotelVO modified) throws RemoteException;
	前置条件	modified 符合输入规范
	后置条件	返回是否修改成功
ImportNewRoomService.getAvailableRoomList	语法	public ArrayList<RoomVO> getAvailableRoomList(String address) throws RemoteException;
	前置条件	启动一个查看酒店可用客房或者录入可用客房的请求
	后置条	返回该酒店的可用客房列表

	件	
ImportNewRoomService.addRoom	语法	public boolean addRoom(RoomVO room) throws RemoteException;
	前置条件	room 符合输入规范
	后置条件	返回是否录入成功
ManageHotelInfoService.addHotel	语法	public boolean addHotel(HotelVO hotel) throws RemoteException;
	前置条件	hotel 符合输入规范
	后置条件	返回是否添加酒店成功
ManageHotelInfoService.addHotelStaff	语法	public boolean addHotelStaff(UserVO staff) throws RemoteException;
	前置条件	staff 符合输入规范
	后置条件	返回是否添加酒店工作人员成功
ManageHotelInfoService. getHotelInfo	语法	public HotelVO getHotelInfo(String hotelAddress) throws RemoteException;
	前置条件	hotelAddress 不为空
	后置条件	无
HotelInfoService.getHotelBriefInfo	语法	public HotelBriefInfoVO getHotelBriefInfo(String address) throws RemoteException;
	前置条件	无
	后置条件	返回该酒店的简要信息
HotelInfoService. getHotelDetails	语法	public HotelVO getHotelDetails(String address) throws RemoteException;
	前置条件	address 符合输入规范
	后置条件	返回在该地址处的酒店的详细信息
HotelInfoService. getBusinessDistrictList	语法	public ArrayList<BusinessDistrictPO> getBusinessDistrictList(String city) throws RemoteException;
	前置条件	city 不为空
	后置条件	无

HotelInfoService.getRoomPrice	语法	public int getRoomPrice(String hotelAddress, RoomType roomType) throws RemoteException;
	前置条件	hotelAddress 不为空
	后置条件	无
需要的服务（需接口）		
服务名	服务	
OrderInfo.getAllOrders(String userID)	得到当前用户所有订单	
HotelDAO.getHotelBriefInfo(String address)	得到单一的酒店简要信息持久化对象	
OderInfo.isReserved(String ID, String address)	得到该酒店是否被该用户预定过	
HotelDAO.getHotelBriefInfoListBySearching(String[] condition)	根据搜索条件得到酒店简要信息持久化对象列表	
HotelDAO.getHotelBriefInfoListByQuerying(String[] condition)	根据查看条件得到酒店简要信息持久化对象列表	
HotelDAO.getHotelDetails(String address)	得到该酒店详细信息的持久化对象	
OrderInfo.getOrderList(String ID, String address)	得到当前用户在该酒店的所有订单	
OrderInfo.getCommentableOrderList(String userID)	得到当前用户可评价酒店对应的订单列表	
OrderInfo.getReservedOrderList（String userID）	得到当前用户所有已执行、已撤销和异常订单	
HotelDAO.update((HotelPO hotel)	更新单一持久化对象	
HotelDAO.find(String address)	根据 address 查找单一持久化对象	
HotelDAO.insert(HotelPO hotel)	在数据库中插入 hotelPO 对象	
userbl.insert(UserPO staff)	在数据库中插入 UserPO 对象	
DatabaseFactory.getHotelDatabase	得到 Hotel 数据库的服务的引用	
roombl.updateSpareRoom(RoomVO roomvo)	更新酒店空房信息	

5.3.2.4 roombl 的模块接口规范

提供的服务（供接口）		
RoomInfoService.getSpareRoomInfoList	语法	public ArrayList<RoomVO> getRoomInfoList (String address)
	前置条件	启动一个查看空房列表的请求
	后置条件	返回空房列表
RoomInfoService.getAvailableRoomNum	语法	public int getAvailableRoomNum(String address, Enum<RoomType> roomType, Date day)
	前置条件	无
	后置条件	返回对应房间类型和日期的空房列表
RoomInfoService.isTimeAvailable (Enum roomType, OrderTime time)	语法	public boolean isTimeAvailable (String addresss, Enum<RoomType> roomType, OrderTime time)
	前置条件	无
	后置条件	返回用户预定时间段有无用户要求的房型剩余
RoomInfoService.checkOrder	语法	public ResultMessage checkOrder(OrderVO vo)
	前置条件	用户生成一个订单后
	后置条件	返回该订单在对应酒店能否得到满足
RoomInfoService.updateSpareRoom	语法	public boolean updateSpareRoom (String address, RoomVO roomvo)
	前置条件	每次更新可用客房后
	后置条件	返回是否更新空房信息成功
RoomInfoService.reduceSpareRoom	语法	public boolean reduceRoom(String address, String roomType, int num)
	前置条件	（1）用户生成订单时 （2）线下入住并更新入住信息时 （3）异常订单通过酒店工作人员手动延迟入住变为已执行订单时
	后置条件	返回是否减少空房成功
RoomInfoService.AddSpareRoom	语法	public boolean addRoom(String address, String roomType, int num)
	前置条件	（1）更新退房信息时 （2）当正常订单变成异常订单时 （3）用户撤销订单时
	后置条件	返回是否增加空房成功
BrowseSpareRoomService.getRoomInfoList	语法	public ArrayList<RoomVO> getRoomInfoList (String address)
	前置条件	启动一个浏览空房信息的请求
	后置条件	返回空房信息列表
UpdateCheckInService.getCheckInList	语法	public ArrayList<RoomVO> getCheckInList(String address)

	前置条件	启动一个更新入住信息请求
	后置条件	返回入住信息列表
UpdateCheckInService.searchCheckInInfo	语法	public RoomVO searchCheckInInfo(String address ,Date time)
	前置条件	无
	后置条件	按入住时间进行查找返回相应的入住信息
UpdateCheckInService.searchCheckInInfo	语法	public RoomVO searchCheckInInfo (String address ,Enum<RoomType> roomType)
	前置条件	无
	后置条件	按房间类型进行查找返回相应的入住信息
UpdateCheckInService.AddCheckIn	语法	public boolean AddCheckIn(String address, RoomVO checkIn)
	前置条件	启动增加入住信息请求，checkIn 符合规范
	后置条件	返回是否更新成功
UpdateCheckInService.validCheckIn	语法	public boolean validCheckIn(String address, RoomVO checkIn)
	前置条件	无
	后置条件	返回 checkIn 的信息是否符合规范
UpdateCheckOutService.getCheckOutList	语法	public ArrayList<RoomVO> getCheckOutList(String address)
	前置条件	启动一个更新退房信息请求
	后置条件	返回退房信息列表
UpdateCheckOutService.searchCheckOutInfo	语法	public RoomPO getCheckOutInfo(String address, Date time)
	前置条件	无
	后置条件	按实际离开时间进行查找返回相应的退房信息
UpdateCheckOutService.searchCheckOutInfo	语法	public RoomPO getCheckOutInfo(String address, Enum<RoomType> roomType)
	前置条件	无
	后置条件	按房间类型进行查找返回相应的退房信息
UpdateCheckOutService.addCheckOut	语法	public boolean AddCheckOut(String address, RoomVO roomvo)
	前置条件	启动增加退房信息请求，checkOut 符合规范
	后置条件	返回是否更新成功
UpdateCheckOutService.validCheckOut	语法	public boolean validCheckIn(String address, RoomVO checkOut)
	前置条件	无
	后置条件	返回 checkOut 的信息是否符合规范
需要的服务（需接口）		
服务名	服务	
RoomDAO.getSpareRoomInfoList(String address)	得到空房信息持久化对象列表	
RoomDAO.getSpareRoom	得到对应房间类型的空房信息持久化对象	

Info(String address, String roomType)	
RoomDAO.getCheckInInfoList(String address)	得到入住信息持久化对象列表
RoomDAO.getCheckInInfo(String address, String time)	得到对应入住时间的入住信息持久化对象
RoomDAO.getCheckInInfo(String address, String roomType)	得到对应房间类型的入住信息持久化对象
RoomDAO.getCheckOutInfoList(String address)	得到退房信息持久化对象列表
RoomDAO.getCheckOutInfo(String address, String time)	得到对应实际离开时间的退房信息持久化对象
RoomDAO.getCheckOutInfo(String address, String roomType)	得到对应房间类型的退房信息持久化对象
RoomDAO.updateRoom(RoomPO po)	更新单一持久化对象
RoomDAO.insertRoom(RoomPO po)	在数据库插入 Room PO 对象
RoomDAO. insertCheckIn(CheckInPO po)	在数据库插入 CheckInPO 对象
RoomDAO. insertCheckOut(CheckOutPO po)	在数据库插入 CheckOutPO 对象
FactoryService.getRoomDAO()	用工厂创建一个 RoomDAO 对象
Factory.createStrategyInfoService()	用工厂创建一个 StrategyInfoService 对象
Factory.createRoomInfoService()	用工厂创建一个 RoomInfoService 对象
Factory.createAvailableRoomService()	用工厂创建一个 AvailableRoomService 对象
StrategyInfoService. .isRightName(address)	判断输入的名称是否正确
AvailableRoomService. .getHotelDetails(address)	得到某个酒店的基本信息

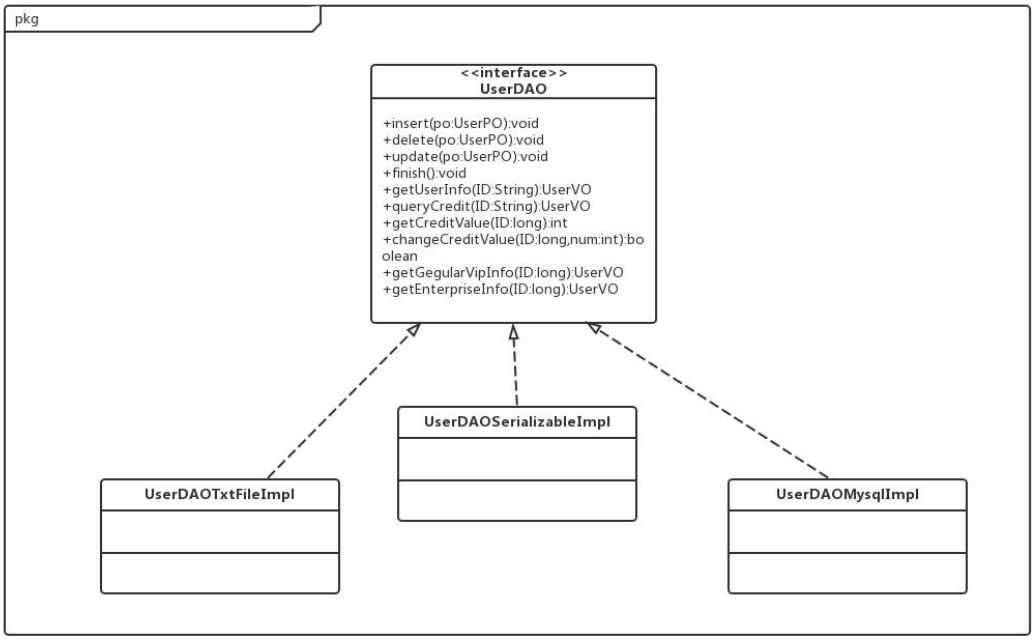
5.3.2.5 strategybl 的模块接口规范

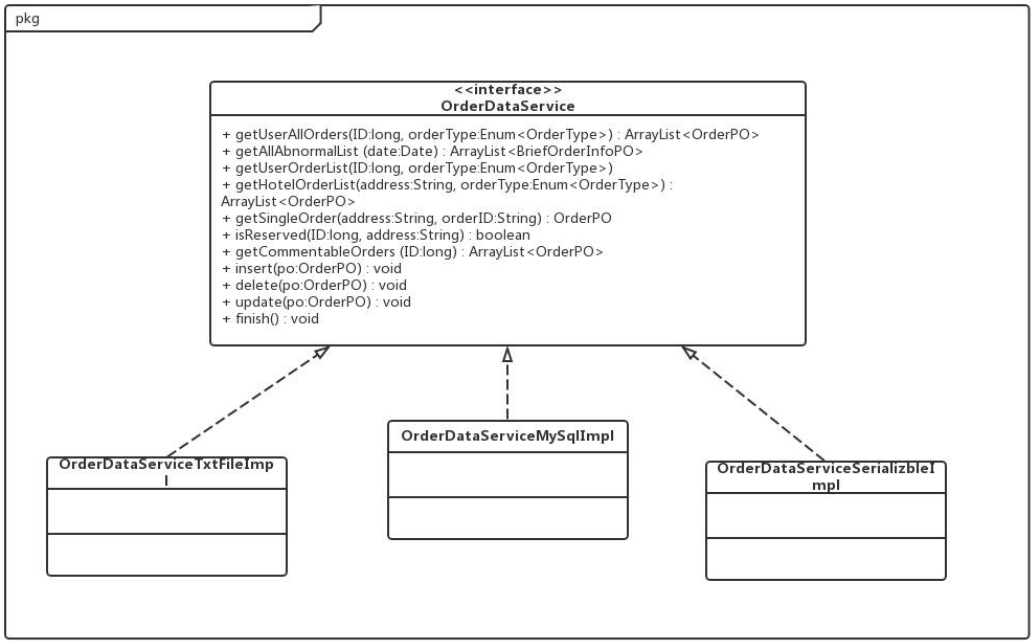
提供的服务（供接口）		
StrategyInfoService.getAvailble PromotionName	语法	public String .getAvailblePromotionName (OrderVO order)
	前置条件	无
	后置条件	返回某订单能享受的唯一酒店促销策略折扣名称，如果没有，则返回 null
StrategyInfoService.getAvailbleMarketStrategyName	语法	public String getAvailbleMarketStrategyName (OrderVO order)
	前置条件	无
	后置条件	返回某订单能享受的唯一网站营销策略折扣名称，如果没有，则返回 null
StrategyInfoService. getBestDiscount	语法	public int getBestDiscount(OrderVO order)
	前置条件	无
	后置条件	返回某订单的最终的折扣百分比
StrategyInfoService.isRightName	语法	public boolean isRightName(String name)
	前置条件	无
	后置条件	判断一个名称格式是否正确
UpdateStrategyService.getStrategyList	语法	public ArrayList<StrategyVO> getStrategyList(String address, Enum<StrategyType> StrategyType)
	前置条件	启动一个制定策略的请求
	后置条件	返回某种策略类型的折扣列表
UpdateStrategyService.getStrategyInfo	语法	public StrategyVO getStrategyInfo(String address, String name)
	前置条件	无
	后置条件	返回某个折扣名称的具体折扣信息
UpdateStrategyService.add	语法	public boolean add(String address, StrategyVO strategy)
	前置条件	该 StrategyVO 符合规范，通过 Valid 方法
	后置条件	返回是否制定策略成功
UpdateStrategyService.modify	语法	public boolean modify(String address, StrategyVO strategy)
	前置条件	该 StrategyVO 符合规范，通过 Valid 方法
	后置条件	返回是否修改策略信息成功
UpdateStrategyService.delele	语法	public boolean delete(String address, StrategyVO strategy)
	前置条件	该 StrategyVO 已存在
	后置条件	返回是否删除成功
UpdateStrategyService.Valid	语法	public boolean Valid(String address, StrategyVO strategy)
	前置条件	无
	后置条件	返回该策略信息是否符合规范
VerifyEnterpriseVipImpl.verifyEnterpriseMember	语法	public boolean verifyEnterpriseMember(String hotelName, String enterpriseName, String

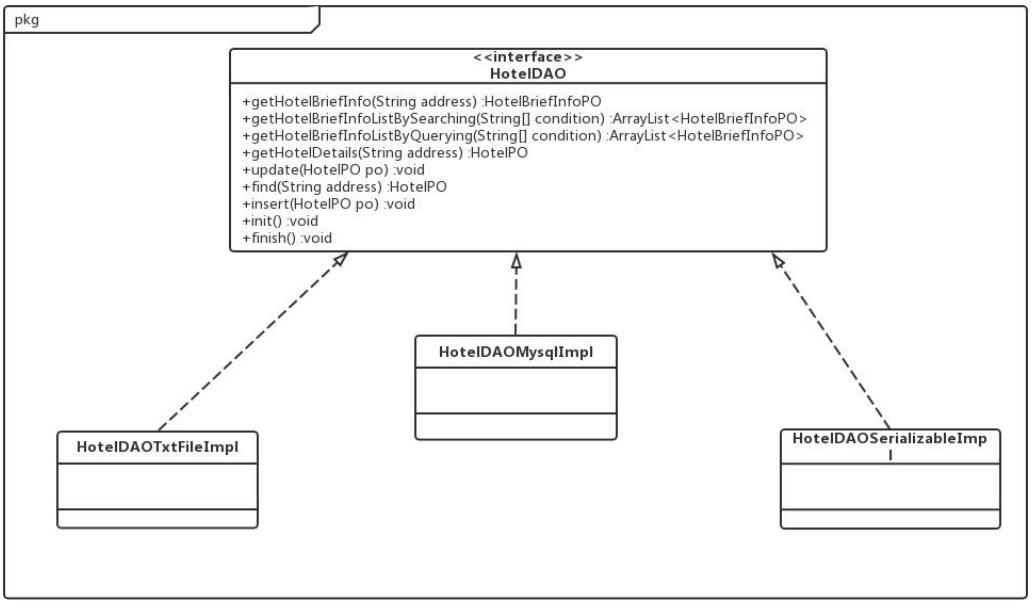
		securityCode)
	前置条件	securityCode 输入符合规范
	后置条件	返回验证是否通过
UpdateStraegyService.verifyTradeArea	语法	public boolean verifyTradeArea(String city, StrategyVO strategyVO)
	前置条件	用户制定或修改商圈折扣
	后置条件	验证特定商圈会员专属折扣的商圈名称在某城市是否存在
UpdateStraegyService.getBusinessDistrictList	语法	public ArrayList<BusinessDistrictPO> getBusinessDistrictList(String city)
	前置条件	用户选择了商圈所在的城市
	后置条件	返回某个城市的商圈列表
需要的服务（需接口）		
服务名	服务	
StrategyDAO.getStrategyList(String address, String StrategyType)	得到策略折扣持久化对象列表	
StrategyDAO.getStrategyInfo(String address, String strategyName)	得到对应折扣名称的折扣策略持久化对象	
StrategyDAO.updateStrategy(StrategyPO po)	更新单一持久化对象	
StrategyDAO.insertStrategy(StrategyPO po)	在数据库插入 StrategyPO 对象	
StrategyDAO.deleteStrategy(StrategyPO po)	在数据库删除某 StrategyPO 对象	
StrategyDAO.EnterpriseMember(String enterpriseName, String securityCode)	通过验证合作企业名称和验证码是否匹配来验证企业会员	
HotelInfoService. getBusinessDistrictList(String city)	得到某个城市的商圈列表	
hotelInfoService.getHotelDetails(String address)	得到酒店的详细信息	
User.getRegularVipInfo(String ID)	得到普通会员信息	
User.getEnterpriseVipInfo(String ID)	得到企业会员信息	
FactoryService.getStrategyDAO()	从工厂新建 StrategyDAO 对象	
FactoryService.createHotelInfoService()	从工厂新建一个 HotelInfoSeivice 对象	

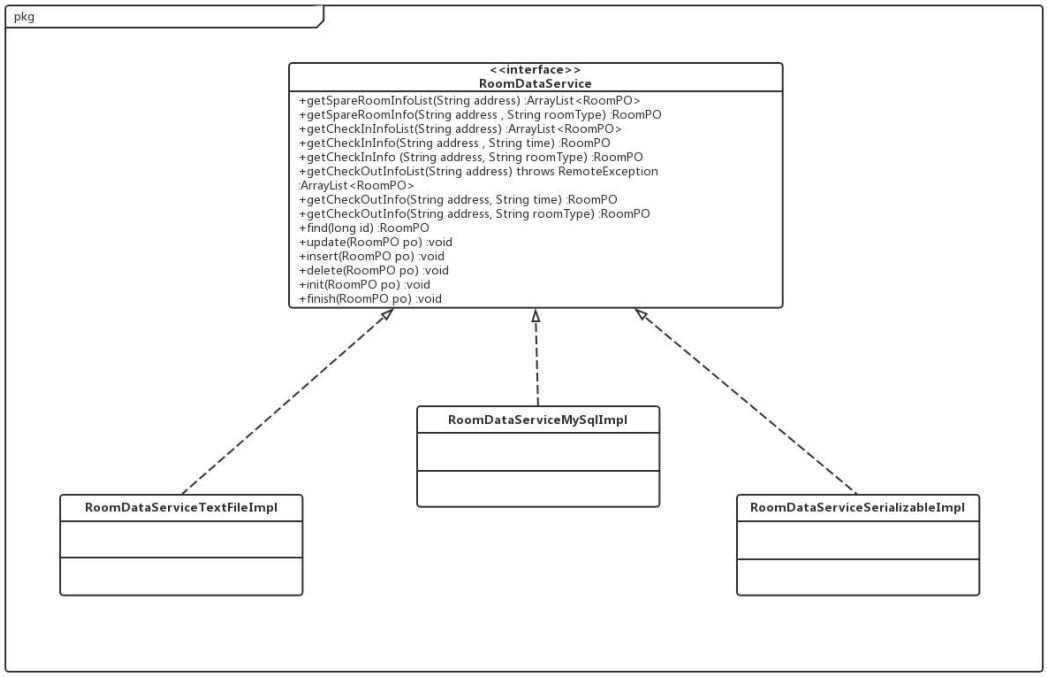
5.4 数据层的分解

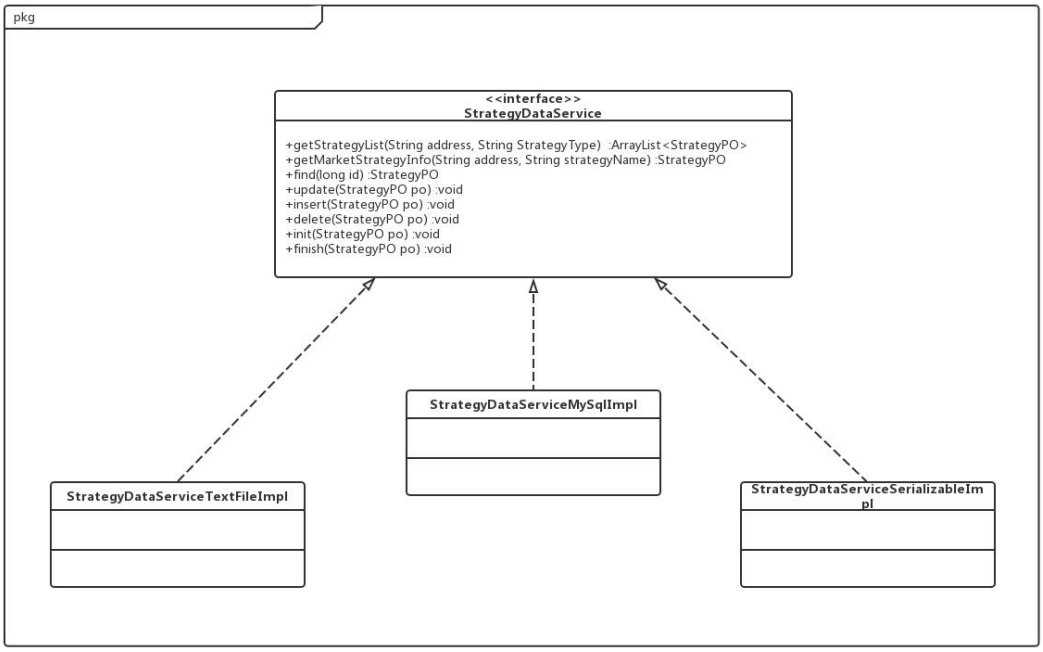
数据层主要给业务逻辑层提供数据访问服务，包括对持久化数据的增、删、改、查。User 业务逻辑需要的服务由 userDAO 接口提供,Order 业务逻辑需要的服务由 orderDAO 接口提供,Hotel 业务逻辑需要的服务由 hotelDAO 接口提供,Room 业务逻辑需要的服务由 roomDAO 接口提供,strategy 业务逻辑需要的服务由 strategyDAO 接口提供。由于持久化数据的保存可能存在多种形式：Txt 文件、序列化文件等，所示抽象了数据服务。数据层模块的描述具体如下图所示。




pkg
<<interface>>
OrderDataService
+ getUserAllOrders(ID:long, orderType:Enum<OrderType>) : ArrayList<OrderPO>
+ getAllAbnormalList (date:Date) : ArrayList<BriefOrderInfoPO>
+ getUserOrderList(ID:long, orderType:Enum<OrderType>)
+ getHotelOrderList(address:String, orderType:Enum<OrderType>) :
ArrayList<OrderPO>
+ getSingleOrder(address:String, orderID:String) : OrderPO
+ isReserved(ID:long, address:String) : boolean
+ getCommentableOrders (ID:long) : ArrayList<OrderPO>
+ insert(po:OrderPO) : void
+ delete(po:OrderPO) : void
+ update(po:OrderPO) : void
+ finish() : void
OrderDataServiceTxtFileImp
OrderDataServiceMySqlImpl
OrderDataServiceSerializbleI
mpl


pkg
<<interface>>
HotelDAO
+getHotelBriefInfo(String address) :HotelBriefInfoPO
+getHotelBriefInfoListBySearching(String[] condition) :ArrayList<HotelBriefInfoPO>
+getHotelBriefInfoListByQuerying(String[] condition) :ArrayList<HotelBriefInfoPO>
+getHotelDetails(String address) :HotelPO
+update(HotelPO po) :void
+find(String address) :HotelPO
+insert(HotelPO po) :void
+init() :void
+finish() :void
HotelDAOTxtFileImpl
HotelDAOMysqlImpl
HotelDAOSerializableImp
l





5.4.1 数据层模块的职责

数据层模块的职责

模块	职责
userDAO	持久化数据的接口，提供集体载入、集体保存、增、改、查服务
userDAOTxtFileImpl	基于 Txt 文件的持久化数据的接口，提供集体

	载入、集体保存、增、改、查服务
userDAOSerializableFileImpl	基于序列化文件的持久化数据接口，提供集体载入、集体保存、增、改、查服务
userDAOMysqlImpl	基于 Mysql 数据库的持久化数据接口，提供集体载入、集体保存、增、改、查服务
orderDAO	持久化数据的接口，提供集体载入、集体保存、增、改、查服务
orderDAOTxtFileImpl	基于 Txt 文件的持久化数据的接口，提供集体载入、集体保存、增、改、查服务
orderDAOSerializableFileImpl	基于序列化文件的持久化数据接口，提供集体载入、集体保存、增、改、查服务
orderDAOMysqlImpl	基于 Mysql 数据库的持久化数据接口，提供集体载入、集体保存、增、改、查服务
hotelDAO	持久化数据的接口，提供集体载入、集体保存、增、改、查服务
hotelDAOTxtFileImpl	基于 Txt 文件的持久化数据的接口，提供集体载入、集体保存、增、改、查服务
hotelDAOSerializableFileImpl	基于序列化文件的持久化数据接口，提供集体载入、集体保存、增、改、查服务
hotelDAOMysqlImpl	基于 Mysql 数据库的持久化数据接口，提供集体载入、集体保存、增、改、查服务
roomDAO	持久化数据的接口，提供集体载入、集体保存、增、改、查服务
roomDAOTxtFileImpl	基于 Txt 文件的持久化数据的接口，提供集体载入、集体保存、增、改、查服务
roomDAOSerializableFileImpl	基于序列化文件的持久化数据接口，提供集体载入、集体保存、增、改、查服务
roomDAOMysqlImpl	基于 Mysql 数据库的持久化数据接口，提供集体载入、集体保存、增、改、查服务
strategyDAO	持久化数据的接口，提供集体载入、集体保存、增、改、查服务
strategyDAOTxtFileImpl	基于 Txt 文件的持久化数据的接口，提供集体载入、集体保存、增、改、查服务
strategyDAOSerializableFileImpl	基于序列化文件的持久化数据接口，提供集体载入、集体保存、增、改、查服务
strategyDAOMysqlImpl	基于 Mysql 数据库的持久化数据接口，提供集体载入、集体保存、增、改、查服务

5.4.2 数据层模块的接口规范

提供的服务（供接口）		
userDAO.getUserInfo	语法	public UserPO getUserInfo(String userID) throws RemoteException

	前置条件	无
	后置条件	查找是否存在相应的 ID，根据输入的 password 返回输入的结果
userDAO.getClientInfo	语法	public ClientInfoPO getClientInfo(String userID) throws RemoteException;
	前置条件	userID 不为空
	后置条件	无
userDAO.getHotelStaffInfo	语法	public HotelStaffInfoPO getHotelStaffInfo(String userID) throws RemoteException;
	前置条件	userID 不为空
	后置条件	无
userDAO. queryCreditRecord	语法	public ArrayList<CreditRecordPO> queryCreditRecord(String userID) throws RemoteException;
	前置条件	无
	后置条件	查找是否存在相应的 ID，存在则返回相应的 PO
userDAO.getCreditValue	语法	public int getCreditValue(String userID) throws RemoteException
	前置条件	无
	后置条件	查找是否存在相应的 ID，存在则返回相应的信用值
userDAO.insertUser	语法	public void insertUser(UserPO po) throws RemoteException
	前置条件	在数据库中不存在同样 ID 的 po
	后置条件	在数据库中增加一个 po 记录
userDAO.insertClient	语法	public boolean insertClient(ClientInfoPO clientInfoPO) throws RemoteException;
	前置条件	在数据库中不存在同样 ID 的 po
	后置条件	在数据库中增加一个 po 记录
userDAO.insertHotelStaff	语法	public boolean insertHotelStaff(HotelStaffInfoPO hotelStaffInfoPO) throws RemoteException;

	前置条件	在数据库中不存在同样 ID 的 po
	后置条件	在数据库中增加一个 po 记录
userDAO.updateUser	语法	public void updateUser(UserPO po) throws RemoteException
	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
userDAO.updateClient	语法	public boolean updateClient(ClientInfoPO clientInfoPO, String oldUserID) throws RemoteException;
	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
userDAO.signRegularVip	语法	public boolean signRegularVip(RegularVipPO regularVipPO) throws RemoteException;
	前置条件	regularVipPO 不为空
	后置条件	在会员中增加一条记录
userDAO. signEnterpriseVip	语法	public boolean signEnterpriseVip(EnterpriseVipPO enterpriseVipPO) throws RemoteException;
	前置条件	enterpriseVipPO 不为空
	后置条件	在会员中增加一条记录
userDAO. getRegularVipInfo	语法	public RegularVipPO getRegularVipInfo(String userID) throws RemoteException;
	前置条件	userID 不为空
	后置条件	无
userDAO. getEnterpriseVipInfo	语法	public EnterpriseVipPO getEnterpriseVipInfo(String userID) throws RemoteException;
	前置条件	userID 不为空
	后置条件	无
userDAO. updateRegularVipInfo	语法	public boolean updateRegularVipInfo(RegularVipPO regularVipPO) throws RemoteException;
	前置条件	regularVipPO 不为空

	后置条件	更新对应的会员信息

提供的服务（供接口）		
orderDAO.getUserOrderList	语法	public ArrayList<BriefOrderInfoPO> getUserOrderList(String userID, Enum<OrderType> orderType) throws RemoteException;
	前置条件	无
	后置条件	按 orderType 的值进行查找返回相应的 OrderPO 列表（全部订单，未执行订单，已执行订单，异常订单和已撤销订单 4 种）
orderDAO.getUserAllOrders	语法	public ArrayList<OrderPO> getUserAllOrders(String userID) throws RemoteException;
	前置条件	ID 有效
	后置条件	返回对应客户的所有订单的 PO，否则返回 null
orderDAO.getUserOrdersByHotel	语法	public ArrayList<OrderPO> getUserOrdersByHotel (String userID, String address) throws RemoteException;
	前置条件	ID 有效, address 有效
	后置条件	返回该用户在该酒店所有订单的列表
orderDAO.getSingleOrder	语法	public OrderPO getSingleOrder(String address, String orderID) throws RemoteException;
	前置条件	无
	后置条件	检测该酒店的订单号是否存在，如果存在，则返回对应订单的 VO，否则返回 null
orderDAO.isReserved	语法	public boolean isReserved(String userID, String address) throws RemoteException;
	前置条件	无
	后置条件	返回该酒店是否被该用户预定过
orderDAO. getCommentableOrders	语法	public ArrayList< OrderPO > getCommentableOrders (String userID) throws RemoteException;
	前置条件	无

	后置条件	返回当前用户可评价酒店对应的订单列表
OrderDAO.insertOrder	语法	public void insertOrder (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
OrderDAO.deleteOrder	语法	public void deleteOrder (OrderPO po) throws RemoteException;
	前置条件	无
	后置条件	删除一个 po
OrderDAO.updateOrder	语法	public void updateOrder (OrderPO po) throws RemoteException;
	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
OrderDAO.getAllAbnormalList	语法	public ArrayList<BriefOrderInfoPO> getAllAbnormalList (Date date) throws RemoteException;
	前置条件	无
	后置条件	返回所查询日期当天所有异常订单列表，若无异常订单，则返回 null（数据库中所有的）
OrderDAO.getHotelOrderList	语法	public ArrayList<BriefOrderInfoPO> getHotelOrderList(String address, Enum<OrderType> orderType) throws RemoteException
	前置条件	无
	后置条件	按 orderType 的值进行查找并按生成时间顺序返回该酒店的相应的订单列表，若无对应类型，则返回 null
OrderDAO.setOrderCommented	语法	public boolean setOrderCommented(String orderID) throws RemoteException;
	前置条件	orderID 不为空
	后置条件	无
OrderDAO.getDetailedOrder	语法	public OrderPO getDetailedOrder(String orderID) throws RemoteException;
	前置条件	orderID 不能为空
	后置条件	无

提供的服务（供接口）		
HotelDAO.getHotelBriefInfo	语法	public HotelBriefInfoPO getHotelBriefInfo(String address) throws RemoteException;
	前置条件	无
	后置条件	按 address 进行查找返回相应的 HotelBriefInfoPO 结果
HotelDAO.getHotelBriefInfoListBySearching	语法	public ArrayList<BriefHotelInfoPO> getHotelBriefInfoListBySearching (String[] condition, ArrayList<BriefOrderInfoPO> orderedHotelList) throws RemoteException;
	前置条件	无
	后置条件	按照 condition 进行查找返回相应的所有 HotelBriefInfoPO
HotelDAO.getHotelBriefInfoListByQuerying	语法	public ArrayList<HotelBriefInfoPO> getHotelBriefInfoListByQuerying (String[] condition, ArrayList<BriefOrderInfoPO> orderedHotelList) throws RemoteException;
	前置条件	无
	后置条件	按照 condition 进行查找返回相应的所有 HotelBriefInfoPO
HotelDAO.getHotelDetails	语法	public HotelPO getHotelDetails(String address) throws RemoteException;
	前置条件	无
	后置条件	根据 address 进行查找返回相应的 HotelPO
HotelDAO.updateHotel	语法	public void updateHotel(HotelPO 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
HotelDAO.insertHotel	语法	public void insertHotel(HotelPO 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
	后置条件	结束持久化数据存储的使用

提供的服务（供接口）		
RoomDAO.getSpareRoomInfoList	语法	public ArrayList<RoomPO> getSpareRoomInfoList(String address, Date day) throws RemoteException;
	前置条件	无
	后置条件	返回空房列表
RoomDAO.getSpareRoomInfo	语法	public RoomPO getSpareRoomInfo(String address , Enum<RoomType> roomType, Date day) throws RemoteException;
	前置条件	无
	后置条件	按房间类型进行查找返回相应的空房信息
RoomDAO.getCheckInInfoList	语法	public ArrayList<RoomPO> getCheckInInfoList(String address) throws RemoteException
	前置条件	无
	后置条件	返回入住信息列表
RoomDAO.getCheckInInfo	语法	public ArrayList<RoomPO> getCheckInInfo(String address , Date startTime, Date endTime) throws RemoteException;
	前置条件	无
	后置条件	按入住时间进行查找返回相应的入住信息
RoomDAO.getCheckInInfo	语法	public RoomPO getCheckInInfo (String address, String roomType) throws RemoteException
	前置条件	无
	后置条件	按房间类型进行查找返回相应的入住信息
RoomDAO.getCheckOutInfoList	语法	public ArrayList<RoomPO> getCheckOutInfoList(String address) throws RemoteException
	前置条件	无
	后置条件	返回退房信息列表
RoomDAO.getCheckOutInfo	语法	public ArrayList<RoomPO> getCheckOutInfo(String address, Date startTime, Date endTime) throws RemoteException;
	前置条件	无
	后置条件	按实际离开时间进行查找返回相应的退房信息
RoomDAO.getCheckOutInfo	语法	public RoomPO getCheckOutInfo(String address, String roomType) throws RemoteException
	前置条件	无
	后置条件	按房间类型进行查找返回相应的退房信息
RoomDAO.updateRoom	语法	public void updateRoom(RoomPO po) throws RemoteException

	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
RoomDAO.insertRoom	语法	public void insertRoom(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
RoomDAO.insertCheckIn	语法	public void insertCheckIn(CheckInPO po) throws RemoteException;
	前置条件	无
	后置条件	在入住表里插入一条记录
RoomDAO.insertCheckOut	语法	public void insertCheckOut(CheckOutPO po) throws RemoteException;
	前置条件	无
	后置条件	在离店表里插入一条记录
提供的服务（供接口）		
StrategyDAO.getStrategyList	语法	public ArrayList<StrategyPO> getStrategyList(String address, Enum<StrategyType> strategyType) throws RemoteException;
	前置条件	无
	后置条件	返回某种策略折扣列表
StrategyDAO.getStrategyInfo	语法	public StrategyPO getStrategyInfo(String address, Enum<StrategyType> strategyType, String strategyName) throws RemoteException;
	前置条件	无
	后置条件	按 ID 进行查找返回相应的策略折扣
StrategyDAO.updateStrategy	语法	public boolean updateStrategy(StrategyPO po) throws RemoteException;
	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
StrategyDAO.insertStrategy	语法	public void insertStrategy(StrategyPO 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
StrategyDAO.deleteStrategy	语法	public void deleteStrategy(StrategyPO 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
StrategyDAO.. verifyEnterpriseMember	语法	public boolean verifyEnterpriseMember(String enterpriseName, String securityCode) throws RemoteException;
	前置条件	无
	后置条件	无

6. 信息视角

6.1 数据持久化对象

系统的 PO 类就是对应的实体类，在此只做简单的介绍。

- UserPO 类包含用户 ID、密码、联系方式、信用变化记录、用户类型属性
- ClientInfoPO 类包含用户 ID、密码、联系方式、信用变化记录、信用值属性
- HotelStaffInfoPO 类包含用户 ID、密码、联系方式、信用变化记录、企业名称属性
- WebMarketStaffPO 类包含用户 ID、密码、联系方式、信用变化记录属性
- OrderPO 类包含客户 ID、订单 ID、酒店名称、酒店地址、开始时间、退房时间、客房类型、客房数量、总价、订单生成时间、最晚订单执行时间、预计入住人数、有无儿童、是否享受折扣、订单状态、评价状态属性
- BriefOrderInfoPO 类包含客户 ID、订单 ID、酒店名称、酒店地址、开始时间、退房时间、客房类型、客房数量、总价属性
- HotelPO 类继承 HotelPO 类，还包含酒店的简介、设施服务、房间类型与原始价格，评价
- HotelBriefInfoPO 类包含酒店的名称、所属商圈、地址、星级、评分
- RoomPO 类是保存房间信息的类，包括酒店地址，房间类型，房间信息，房间价格属性
- CheckInOutPO 类是保存入住退房信息的类，继承 RoomPO 类，还包括入住时间，预计离开时间，实际离开时间属性
- StrategyPO 类是保存折扣策略信息的类，包括酒店地址，折扣类型，折扣名称，折扣百分比，最小房间数，合作企业名称，验证码，开始时间，结束时间，会员等级，商圈名称属性

持久化用户对象 HotelPO 的定义如下所示。

```
public class HotelPO extends HotelBriefInfoPO {

    private String briefIntroduction;
    private String facilityAndService;
    private HashMap<RoomType, Integer> roomTypeAndPrice;
    private HashMap<String, String> comments;

    public String getBriefIntroduction() {
        return briefIntroduction;
    }
    public void setBriefIntroduction(String briefIntroduction) {
        this.briefIntroduction = briefIntroduction;
    }
    public String getFacilityAndService() {
        return facilityAndService;
    }
    public void setFacilityAndService(String facilityAndService) {
```

```
        this.facilityAndService = facilityAndService;
    }
    public HashMap<RoomType, Integer> getRoomTypeAndPrice() {
        return roomTypeAndPrice;
    }
    public void setRoomTypeAndPrice(HashMap<RoomType, Integer>
roomTypeAndPrice) {
        this.roomTypeAndPrice = roomTypeAndPrice;
    }
    public HashMap<String, String> getComments() {
        return comments;
    }
    public void setComments(HashMap<String, String> comments) {
        this.comments = comments;
    }
}
```

6.2 持久化格式

Txt 数据保持格以 User.txt 为例。每行分别对应 ID、密码、联系方式、用户类型。中间用“:”隔开。如下所示：
123：qweq123：11111111111：客户
456：vdddfvvd：22222222222：网站营销人员

6.3 数据库表

数据库中包含 client 表、commomMember 表、creditRecord 表、enterpriseMember 表、hotelStaff 表、webManageStaff 表、webMarketStaff 表、hotel 表、comments 表、roomTypeAndPrice 表、OrderInfo 表、room 表、checkInOut 表、birthdayPromotion 表、cooperativeEnterprise 表、MultiRoomPromotionandMemberRankMarket 表、specificTimePromotion 表、vipTradeAreaMarket 表

client 表:

Field	Type	Null	Key	Default	Extra
userID	char(20)	NO	PRI	NULL	
password	bigint(20)	YES		NULL	
telNum	int(11)	YES		NULL	
creditValue	int(11)	YES		NULL	
isMember	char(16)	YES		NULL	

commomMember 表:

Field	Type	Null	Key	Default	Extra

userID	char(20)	YES	MUL	NULL	
birth	date	YES		NULL	

creditRecord 表:

Field	Type	Null	Key	Default	Extra
userID	char(20)	YES	MUL	NULL	
changeTime	date	YES		NULL	
orderID	int(11)	YES		NULL	
action	int(11)	YES		NULL	
process	int(11)	YES		NULL	
creditResult	int(11)	YES		NULL	

enterpriseMember 表:

Field	Type	Null	Key	Default	Extra
userID	char(20)	YES	MUL	NULL	
enterpriseID	char(20)	YES		NULL	

hotelStaff 表:

Field	Type	Null	Key	Default	Extra
enterpriseID	char(20)	NO	PRI	NULL	
userID	char(20)	YES		NULL	
password	char(16)	YES		NULL	
telNum	bigint(20)	YES		NULL	

webManageStaff 表:

Field	Type	Null	Key	Default	Extra
userID	char(20)	NO	PRI	NULL	
password	char(16)	YES		NULL	
telNum	bigint(20)	YES		NULL	

webMarketStaff 表:

Field	Type	Null	Key	Default	Extra
userID	char(20)	NO	PRI	NULL	
password	char(16)	YES		NULL	
telNum	bigint(20)	YES		NULL	

hotel 表:

Field	Type	Null	Key	Default	Extra
hotelName	char(20)	NO		NULL	
businessDistrict	char(20)	NO		NULL	
hotelAddress	char(50)	NO	PRI	NULL	
starLevel	int(11)	YES		NULL	
mark	float(2,1)	YES		NULL	
briefIntroduction	char(200)	YES		NULL	
facilityAndService	char(100)	YES		NULL	

comments 表:

Field	Type	Null	Key	Default	Extra
hotelAddress	char(50)	YES	MUL	NULL	
clientName	char(20)	YES		NULL	
comment	char(200)	YES		NULL	

roomTypeAndPrice 表:

Field	Type	Null	Key	Default	Extra
address	char(50)	YES	MUL	NULL	
roomType	int(11)	YES		NULL	
price	int(11)	NO		NULL	

OrderInfo 表:

Field	Type	Null	Key	Default	Extra
orderID	int(11)	NO	PRI	NULL	auto_increment
userID	char(20)	NO		NULL	
hotelName	char(20)	NO		NULL	
hotelAddress	char(20)	NO		NULL	
beginDate	date	YES		NULL	
finishDate	date	YES		NULL	
roomType	int(11)	YES		NULL	
num	int(11)	YES		NULL	
totalPrice	int(11)	YES		NULL	
orderProduceTime	datetime	YES		NULL	
lastedOrderDoneTime	datetime	YES		NULL	
numOfPerson	int(11)	YES		NULL	
isChildren	int(11)	YES		NULL	
isOnSale	int(11)	YES		NULL	
orderState	int(11)	YES		NULL	
isCommented	int(11)	YES		NULL	

room 表:

Field	Type	Null	Key	Default	Extra
hotelAddress	char(50)	YES	MUL	NULL	
roomType	int(11)	YES		NULL	
roomNum	int(11)	YES		NULL	
roomPrice	int(11)	YES		NULL	

checkInOut 表:

Field	Type	Null	Key	Default	Extra
hotelAddress	char(50)	YES	MUL	NULL	
roomType	int(11)	YES		NULL	
roomNum	int(11)	YES		NULL	
roomPrice	int(11)	YES		NULL	
checkInTime	datetime	YES		NULL	
expDepartTime	datetime	YES		NULL	
actDepartTime	datetime	YES		NULL	

birthdayPromotion 表:

Field	Type	Null	Key	Default	Extra
hotelAddress	char(50)	YES	MUL	NULL	
strategyName	char(20)	YES		NULL	
discount	float	YES		NULL	
strategyType	int(11)	YES		NULL	

cooperativeEnterprise 表:

Field	Type	Null	Key	Default	Extra
hotelName	char(20)	YES		NULL	
enterpriseName	char(20)	YES		NULL	
strategyType	int(11)	YES		NULL	
strategyName	char(20)	YES		NULL	
securityCode	char(8)	YES		NULL	
discount	float	YES		NULL	

MultiRoomPromotionandMemberRankMarket 表:

Field	Type	Null	Key	Default	Extra
hotelAddress	char(50)	YES	MUL	NULL	
strategyName	char(20)	YES		NULL	
discount	float	YES		NULL	

strategyType	int(11)	YES		NULL	
minRoomNumOrVipRank	int(11)	YES		NULL	

specificTimePromotion 表:

Field	Type	Null	Key	Default	Extra
hotelAddress	char(50)	YES	MUL	NULL	
discount	float	YES		NULL	
strategyType	int(11)	YES		NULL	
strategyName	char(20)	YES		NULL	
startTime	date	YES		NULL	
endTime	date	YES		NULL	

vipTradeAreaMarket 表:

Field	Type	Null	Key	Default	Extra
hotelAddress	char(50)	YES	MUL	NULL	
discount	float	YES		NULL	
strategyType	int(11)	YES		NULL	
strategyName	char(20)	YES		NULL	
vipRank	int(11)	YES		NULL	
tradeArea	char(20)	YES		NULL	