

# 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	3
1.引言	4
1.1 编制目的	4
1.2 词汇表	
1.3 参考资料	
2.产品概述	
2-) 月日194.火亡	4
3.逻辑视角	4
. (P. A. ) P. A.	
4.组合视角	6
4.1 开发包图	6
4.2 运行时进程	9
4.3 物理部署	10
5.接口视角	10
5.1 模块的职责	10
5.2 用户界面层的分解	
5.2.1 用户界面层模块的职责	13
5.2.2 用户界面层模块的接口规范	13
5.2.3 用户界面模块设计原理	13
5.3 业务逻辑层的分解	14
5.3.1 业务逻辑层模块的职责	14
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	
5.3.2.2 orderbl 模块的接口规范	
5.3.2.3 hotelbl 模块的接口规范	
5.3.2.4 roombl 的模块接口规范	
5. 3. 2. 5 strategybl 的模块接口规范	
5.4 数据层的分解	
5.4.1 数据层模块的职责	
5.4.2 数据层模块的接口规范	35
6.信息视角	43
6.1 数据持久化对象	
6.2 持久化格式	
6.3 数据库表	44

## 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</ordervo>	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。

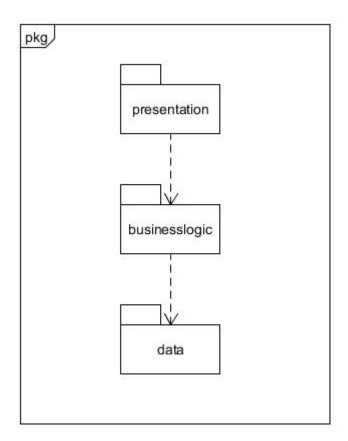
## 2. 产品概述

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。

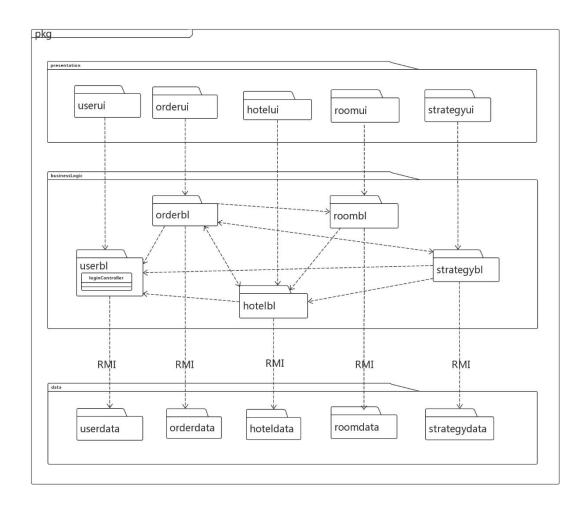
## 3. 逻辑视角

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



## 3.2 软件体系结构逻辑设计方案



## 4. 组合视角

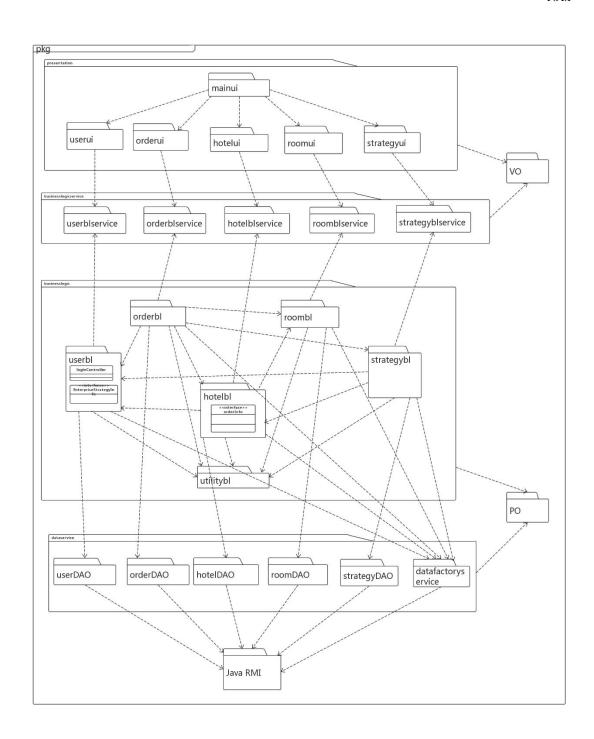
### 4.1 开发包图

酒店预定系统最终开发包设计

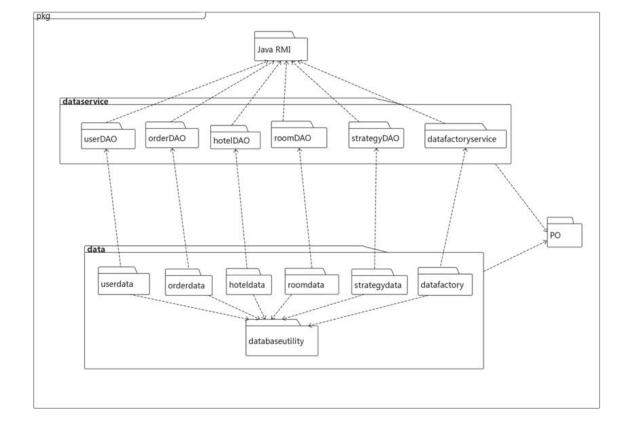
开发(物理)包	依赖的其他开发包
mainui	userui,orderui,hotelui,roomui,strategyui,vo
userui	userblservice,界面类库包
userblservice	
userbl	userblservice, userDao, po, vo
userDao	Java RMI,po
userdata	userDao ,Java RMI,po ,databaseutility
orderui	orderblservice,界面类库包
orderblservice	

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

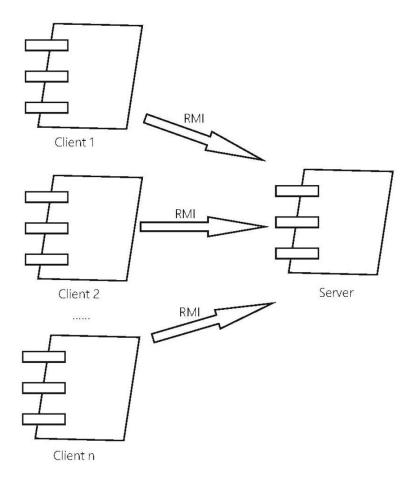


酒店预订系统服务器端开发包图



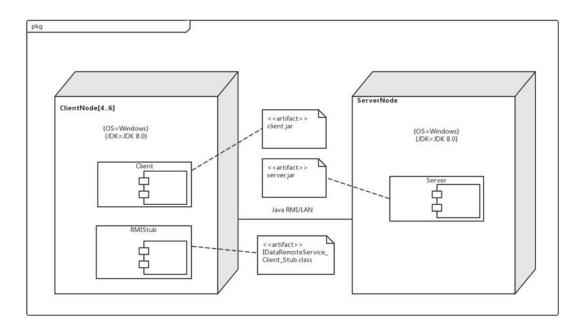
## 4.2运行时进程

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

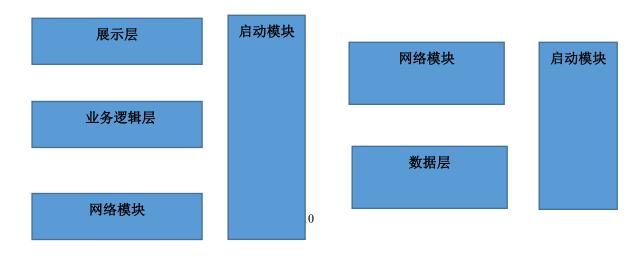
酒店预订系统中客户构件是放在客户端机器上,服务器端构件时放在服务器端机器上。在客户端节点上,还要部署\*\*构件。由于 Java RMI 构件属于 JDk6.0 的一部分。所以,在系统 JDk 环境已经设置好的情况下,不要再独立部署。



## 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。



#### 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

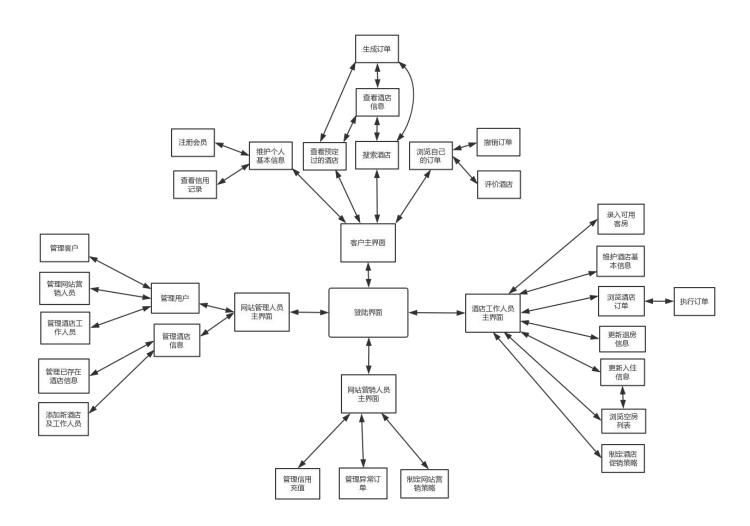
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
层	职责
启动模块	负责初始化网络通信机制,启动用户界面
数据层	负责数据的持久化及数据访问接口
服务器端网络模块	利用 Java Socket 机制实现数据通讯

#### 层之间调用的接口

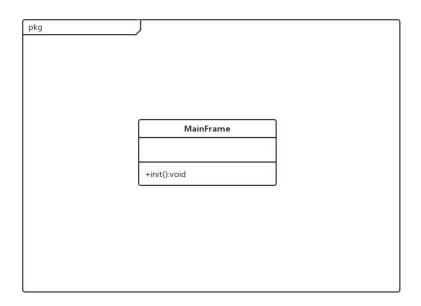
接口	服务调用方	服务提供方
OrderBLService		
UserBLService		
HotelBLService	客户端展示层	客户端业务逻辑层
RoomBLService		
StrategyBLService		
OrderDAO		
UserDAO		
HotelDAO		
RoomDAO	客户端业务逻辑层	服务器端数据层
StrategyDAO		
DatabaseFactory		

#### 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

	语法	Init(args:String[])
MainFrame	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

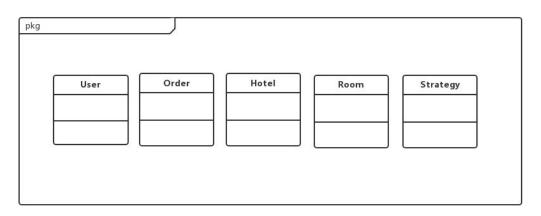
服务名	服务
businessloicservice.LoginBLService	登录界面的业务逻辑接口
Businesslogcservice.*BLService	每个界面都有一个相应的业务逻辑接口

#### 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)
LoginAndSignUpServic	语法	public boolean login(String userID, String password)
e.login	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
LoginAndSignUpServic	语法	public boolean add(UserVO user)
e.add	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
ModifyClientInfoServic e.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
ModifyClientInfoServic	语法	public ClientInfoVO getClientInfo(String userID)
e.getClientInfo	前置条 件	客户查看用户信息
	后置条 件	返回该客户信息
ManageUserInfoServic	语法	public boolean add(UserVO user)
e.add	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
ManageUserInfoServic	语法	public HotelStaffInfoVO getHotelStaffInfo(String userID)
e.getHotelStaffInfo	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
ManageUserInfoServic e.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.signEnt	语法	public boolean signEnterpriseVip(EnterpriseVO
erpriseVip		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.signRe	语法	public boolean signRegularVip(VipInfoVO regularVip)
gularVip	前置条 件	客户申请企业会员
	后置条 件	返回是否申请成功
QueryClientCreditReco	语法	public ArrayList <clientcreditrecordvo></clientcreditrecordvo>
rdService.queryCreditR		queryCreditRecord(String userID)
ecord	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.getCre	语法	public int getCreditValue(String userID)
ditValue	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.change	语法	public boolean changeCreditValue(String userID,int
CreditValue		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

	7		
VerifyEnterpriseMembe	语法	public boolean verifyEnterpriseMember(String	
r.verifyEnterpriseMemb		enterpriseName, String securityCode)	
er	前置条 件	EnterpriseID 和 securityCode 符合输入规范	
	后置条	返回是否注册成功	
	件		
VipInfo.getRegularVipI	语法	public VipInfoVO getRegularVipInfo(String userID)	
nfo	前置条	userID 符合输入规范	
	件		
	后置条	返回普通会员的信息	
	件		
VipInfo.getEnterpriseVi	语法	public VipInfoVO getEnterpriseVipInfo(String userID)	
pInfo	前置条	userID 符合输入规范	
	件		
	后置条	返回企业会员的信息	
	件		
服务名	服务		
DatabaseFactory.getUse	得到 user 数据库的服务的引用		
rDatabase	14-4 XXVIII   HAMINAN HA 41/14		
EnterpriseInfo.verifyEn	验证企业会员注册的信息		
terpriseMember(String			
hotelName, String			
enterpriseName, String			
securityCode)			
userDAO.getUserInfo(	得到 ID	对应的用户信息	
String userID)			
userDAO.getClientInfo		-1 4L	
(String userID)	得到 ID 对应的客户信息		
userDAO.getHotelStaff	组AUD 对应的演成了 <i>作</i> L P 产自		
Info(String userID)	得到 ID 对应的酒店工作人员信息		
userDAO.queryCredit	得到 ID	对应的用户信用值及信用值变化记录	
(String userID)			
UserDAO.getCreditVal	得到 ID	对应的用户的当前信用值	
ue(String userID)			
userDAO.insertUser(Us	任)片	- 快力 (ル) - (中)	
erPO userPO)	烟八甲-	-持久化对象	
userDAO.insertClient(	任)员	- 快力ル社会	
ClientInfoPO client)	1出八牛	·持久化对象 	
userDAO.insertHotelSt			
aff(HotelStaffInfoPO	插入单一	一持久化对象	
hotelStaff)			

userDAO.updateUser(U serPO modified)	更新单一持久化对象
userDAO.updateClient( ClientInfoPO modified)	更新单一持久化对象
userDAO.signRegularV ip(RegularVipPO regularVip)	插入单一持久化对象
userDAO. signEnterpriseVip(Enter priseVipPO enterpriseVip)	插入单一持久化对象
userDAO. getRegularVipInfo(Stri ng userID)	得到 ID 对应的普通会员信息
userDAO. getEnterpriseVipInfo(St ring 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)		
	语法	<pre>public ArrayList<ordervo> getAllOrders(String userID);</ordervo></pre>
OrderInfo.getAllOrders	前置条	ID 有效
	件	
	后置条	返回对应客户的所有订单的 VO, 否则返回 null
	件	
	语法	public boolean isReserved(String userID, String address);
OrderInfo.isReserved	前置条	ID 和 address 有效
	件	
	后置条	返回该酒店是否被该用户预定过
	件	
	语法	public ArrayList <ordervo> getCommentableOrderList</ordervo>
OrderInfo.getComment		(String userID);
ableOrderList	前置条	ID 有效
	件	
	后置条	返回当前用户可评价酒店对应的订单列表
	件	

	语法	public ArrayList <ordervo> getOrderList(String userID,</ordervo>
OrderInfo.getOrderList	ши	String address);
Orderinio.getOrderEist	前置条	ID 和 address 有效
	件	ID An addices A XX
	后置条	返回该用户在该酒店的所有订单
	件	<b>应回该用)在该相归的</b> 拥有4平
OderInfo.getReservedO	语法	public ArrayList <ordervo> getReservedOrderList(String</ordervo>
rderList	旧亿	•
TUCILIST	光型々	userID);
	前置条 件	userID 符合输入规范
		· 运回济田 白庇 左耳 执 经 · □ 樹 经 和 目 偿 江 苗
	后置条	返回该用户所有已执行、已撤销和异常订单
	件	
OderInfo.	语法	public boolean setOrderCommented(String orderID);
setOrderCommented	前置条	orderID 有效
	件	
	后置条	将对应的订单置为已评价订单,若置成功则返回 true,
	件	反之则返回 false
	语法	public OrderVO getDetailedOrder(String orderID);
Order.getDetailedOrder	前置条	客户请求展开订单详情
	件	
	后置条	检测订单号是否存在,如果存在,则返回对应订单的 VO,
	件	否则返回 null
	语法	public OrderVO getSingleOrder(String address, String
Order.getSingleOrder		orderID);
	前置条	酒店工作人员输入订单号
	件	
	后置条	检测该订单号是否存在,如果存在,检测是否是该酒店
	件	的订单,若是则返回对应订单的 VO,否则返回 null
	语法	public ArrayList <brieforderinfovo> getHotelOrderList</brieforderinfovo>
Order.getHotelOrderLis		(String address, Enum <ordertype> orderType);</ordertype>
t	前置条	酒店工作人员启动浏览订单流程
	件	
	后置条	按 orderType 的值进行查找并按生成时间顺序返回相应的
	件	订单列表
	语法	public ArrayList <brieforderinfovo> getUserOrderList</brieforderinfovo>
Order.getUserOrderList		(String userID, Enum <ordertype> orderType);</ordertype>
	前置条	客户启动浏览订单流程
	件	
	后置条	按 orderType 的值进行查找并按生成时间顺序返回相应的
	件	订单列表
	语法	public ArrayList <brieforderinfovo></brieforderinfovo>
Order.getAbnormalOrd		getAbnormalOrderList (Date date);
erList	前置条	网站营销人员发起浏览每日异常订单情况操作
	144 12 741	

	件	
	后置条	返回所查询日期当天所有异常订单列表
	件	
	语法	public OrderVO initNewOrder(String userID, String
Order.initNewOrder		address);
	前置条	客户确认即将生成订单的酒店信息
	件	
	后置条	初始化订单信息,返回一个被初始化的 OrderVO
	件	
	语法	public ResultMessage checkNewOrder(OrderVO vo);
Order.checkNewOrder	前置条	客户确认生成订单
	件	
	后置条	检查订单要求是否可以得到满足,格式是否正确,并返
	件	回结果
Ondered Discost	语法	public ResultMessage addNewOrder(OrderVO vo);
Order.addNewOrder	前置条	订单要求可以得到满足
	件 后置条	增加新的订单
	一 件 件	增加利的 () <del>中</del>
	语法	public int getPrice(OrderVO vo);
Order.getPrice	前置条	用户已填写房间类型和房间数量
Order.gett fice	件	用/ 口块司房间大空仰房间数里
	后置条	计算并返回订单价格
	件	
	语法	public int getOriginalPrice(String hotelAddress, RoomType
Order.getOriginalPrice		roomType);
	前置条	用户选择生成订单
	件	
	后置条	返回对应酒店各个房型对应的原始价格
	件	
	语法	public boolean delayCheckIn (OrderVO vo, int hours);
Order.delayCheckIn	前置条	酒店工作人员确定延迟入住
	件	
	后置条	延迟最晚入住时间,返回是否成功
	件	
0.1.	语法	public boolean getOrderDone(OrderVO vo);
Order.getOrderDone	前置条	酒店工作人员确定执行订单
	件	エボルギンサルナエコルにマメーン・ハウトはたとマン
	后置条	更新当前订单状态为已执行订单,并为客户增加与订单
	件	金额等值的信用值
Order.withdrawOrder	语法	public boolean withdrawOrder(OrderVO vo);  克 白 海 江 始 将 江 角
Order.witndrawOrder	前置条件	客户确认撤销订单
	TT	

:			
	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.systemWithdraw	语法	public boolean systemWithdrawOrder(OrderVO vo, boolean isRecoverHalfCredit);	
Order	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	
	语法	public Time getLatestDoneTime(OrderVO vo);	
Order.getLatestDoneTi me	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.getUserOrd erList(String userID, Enum <ordertype> orderType)</ordertype>	根据 ord	erType 的值得到当前用户的对应订单持久化对象列表	
orderDAO.getUserOrde	返回该客户在该酒店的所有订单列表		
rsByHotel(String			
userID, String address)			
OrderDAO.getUserAll	返回对应客户的所有订单的 PO,否则返回 null		
Orders(String userID)			
OrderDAO.insertOrder( OrderPO po)	插入单一	-持久化对象	
OrderDAO.deleteOrder (OrderPO po)	删除单-	·持久化对象	
OrderDAO.updateOrde	更新单-	-持久化对象	
r(OrderPO po)			
OrderDAO.getHotelOr		rderType 的值进行查找并按时间顺序返回该酒店的相应订	
derList(String address,	单列表		
Enum <ordertype></ordertype>			
orderType) OrderDAO.getAllAbno	<b>但到本</b> 语	1日期当于所有县党订单列事	
rmalList(Date date)	得到查询日期当天所有异常订单列表		
OrderDAO.getSingleOr	检测li玄洲	盾店的订单号是否存在,如果存在,则返回对应订单的 VO,	
der(String address,	否则返回		
String orderID)	H 7176		
OrderDAO.getDetailed	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	
Order(String orderID)		返回null	
OrderDAO.isReserved(	返回该酒	原店是否被该用户预定过	
	ł		

String userID, String	
address);	返回当前用户可评价酒店对应的订单列表
OrderDAO.getCommen	
tableOrders (String	
userID);	<u> </u>
OrderDAO.	将对应的订单置为已评价订单
setOrderCommented(St	
ring orderID)	
RoomInfoService.getA	获取当前酒店 roomType 房型的空房数量
vailableRoomNum(Stri	
ng address,	
Enum <roomtype></roomtype>	
roomType)	
RoomInfoService.isTim	查询用户预定时间段有无用户要求的房型剩余
eAvailable (String	
address,	
Enum <roomtype></roomtype>	
roomType, Date	
beginDate, Date	
finishDate)	
RoomInfoService.addS	更新(增加)当前酒店被预定房型的可用房间数据
pareRoom(String	
address, int change,	
Enum <roomtype></roomtype>	
roomType)	
RoomInfoService.reduc	更新(减少)当前酒店被预定房型的可用房间数据
eSpareRoom(String	
address, int change,	
Enum <roomtype></roomtype>	
roomType)	
RoomInfoService.check	检查该订单中的信息当前能否满足
Order(OrderVO vo)	
HotelInfoService.getHo	得到当前酒店的简要信息
telBriefInfo(String	
address)	
HotelInfoService.getHo	得到当前酒店的详细信息
telDetails(String	
address)	
HotelInfoService.	得到对应酒店各个房型对应
getOriginalPrice(String	
hotelAddress,	
RoomType roomType)	
ClientCreditInfo.getCre	得到当前登陆客户的信用值
ditValue(String userID)	

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

## 5.3.2.3 hotelbl 模块的接口规范

		提供的服务(供接口)
CheckOrderedHotelSeri vice.enrollHotelBriefInf	语法	public ArrayList <hotelbriefinfovo> enrollHotelBriefInfoList (String userID) throws RemoteException;</hotelbriefinfovo>
oList	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.get HotelBriefInfoListBySe	语法	<pre>public ArrayList<orderedhotelinfovo> getHotelBriefInfoListBySearching (String[] condition) throws RemoteException;</orderedhotelinfovo></pre>
arching	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	<pre>public ArrayList<businessdistrictpo> getBusinessDistrictList(String city) throws RemoteException;</businessdistrictpo></pre>
	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.get HotelBriefInfoListByQ	语法	public ArrayList <orderedhotelinfovo> getHotelBriefInfoListByQuerying (String[] condition) throws RemoteException;</orderedhotelinfovo>
uerying	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.getH	语法	public HotelVO getHotelDetails(String address) throws RemoteException;
otelDetails	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.get	语法	public ArrayList <ordervo> getOrders(String address, String userID) throws RemoteException;</ordervo>
Orders	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
CommentOnHotelServi	语法	public ArrayList <ordervo> getCommentableOrderList(String userID) throws</ordervo>
ce.getCommentableOrd		RemoteException;
erList	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
CommentOnHotelServi ce.confirmComment	语法	public boolean confirmComment(String username, float mark, String comment, String hotelAddress, String orderID) throws RemoteException;
co.commiconmicit	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
MaintainHotelBasicInfo Service.enrollHotelBasi	语法	public HotelVO enrollHotelBasicInfo(String address) throws RemoteException;
cInfo	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
MaintainHotelBasicInfo	语法	public boolean confirmModify(HotelVO modified) throws RemoteException;
Service.confirmModify	前置条 件	modified 符合输入规范
	后置条 件	返回是否修改成功
ImportNewRoomServic	语法	public ArrayList <roomvo> getAvailableRoomList(String address) throws RemoteException;</roomvo>
e.getAvailableRoomLis t	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	
	语法	public boolean addRoom(RoomVO room) throws
ImportNewRoomServic		RemoteException;
e.addRoom	前置条 件	room 符合输入规范
	后置条	返回是否录入成功
	件	
	语法	public boolean addHotel(HotelVO hotel) throws
ManageHotelInfoServic		RemoteException;
e.addHotel	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
	语法	public boolean addHotelStaff(UserVO staff) throws
ManageHotelInfoServic		RemoteException;
e.addHotelStaff	前置条	staff 符合输入规范
	件	
	后置条 件	返回是否添加酒店工作人员成功
ManagallatalInfaCania	件 语法	public HotalVO gotHotalInfo(String hotalAddrags) throws
ManageHotelInfoServic e. getHotelInfo	<b>归</b> 亿	public HotelVO getHotelInfo(String hotelAddress) throws RemoteException;
c. geniotennio	前置条	hotelAddress 不为空
	件	noter tadiess 1 /3 1.
	后置条	无
	件	
	语法	public HotelBriefInfoVO getHotelBriefInfo(String address)
HotelInfoService.getHo		throws RemoteException;
telBriefInfo	前置条 件	无
	后置条 件	返回该酒店的简要信息
HotelInfoService.	语法	public HotelVO getHotelDetails(String address) throws RemoteException;
getHotelDetails	前置条	address 符合输入规范
	件	
	后置条	返回在该地址处的酒店的详细信息
	件	II. A Lived Division
HotelInfoService.	语法	public ArrayList <businessdistrictpo> getBusinessDistrictList(String city) throws</businessdistrictpo>
getBusinessDistrictList		RemoteException;
	前置条	city 不为空
	件	
	后置条	无
	件	

getRoomPrice 前置条 件	HotelInfoService.	语法	public int getRoomPrice(String hotelAddress, RoomType	
###	getRoomPrice		roomType) throws RemoteException;	
信置条   作		前置条	hotelAddress 不为空	
服务名 OrderInfo getAllOrders (String userID) HoteIDAO getHoteIBri efInfo(String address) OderInfo.isReserved(St ring ID, String address) HoteIDAO.getHoteIBri efInfo(IstBySearching(String[I] condition) HoteIDAO.getHoteIBri efInfoListBySearching(String[I] condition) HoteIDAO.getHoteIBri efInfoListByQuerying(String[I] condition) HoteIDAO.getHoteIDet alist(String address) OrderInfo getCorderList(String userID) OrderInfo getComment ableOrderList(String userID) OrderInfo getReserved OrderList (String userID) HoteIDAO.update((Hot eIPO hoteI) HoteIDAO.ind(String address)  OrderInfo getReserved OrderList(String userID)  OrderInfo getReserved OrderList(String userID)  HoteIDAO.ind(String address)  DatabaseFactory.getHot el Exting the Able OrderList (String userID)  OrderInfo getReserved OrderList(String userID)  HoteIDAO.ind(String address)  ### Address at ### Able Order Able OrderList(String userID)  ### Address at ### Able Order Able OrderList(String userID)  ### Address at ### Able Order Able O		件		
服务名		后置条	无	
RB タ名 OrderInfo.getAllOrders (String userID) HotelDAO.getHotelBri efInfo(String address) OderInfo.isReserved(St ring ID, String address) HotelDAO.getHotelBri efInfoListBySearching(String[] condition) HotelDAO.getHotelBri efInfoListByQuerying(String[] condition) HotelDAO.getHotelBri efInfoListByQuerying(String[] condition) HotelDAO.getHotelDet ails(String address) OrderInfo.getCorderList(String userID) OrderInfo.getComment ableCorderList(String userID) OrderInfo.getReserved OrderList (String userID)  OrderInfo.getReserved OrderList (String userID) HotelDAO.update(Hotel PO hotel) HotelDAO.find(String address)  OrderInfo.getComment ableCorderList(String userID)  OrderInfo.getReserved OrderList (String userID)  OrderInfo.getComment orderList (String userID)  Order		件		
OrderInfo.getAllOrders (String userID) HoteIDAO getHotelBri efInfo(String address) OderInfo.isReserved(St ring ID, String address) HoteIDAO getHotelBri efInfo(String address) HoteIDAO getHotelBri efInfoListBySearching(String[] condition) HoteIDAO.getHotelBri efInfoListByQuerying(String[] condition) HoteIDAO.getHotelDet ails(String address)  OrderInfo.getCorderList(String userID)  OrderInfo.getComment ableOrderList(String userID)  HoteIDAO.update(Hot eIPO hotel) HoteIDAO.ind(String address)  Fayliam IP Fix (String userID)  HoteIDAO.ind(String address)  OrderInfo.getComment ableOrderList(String userID)  HoteIDAO.ind(String address)  Fayliam IP Fix (String userID)  Fayl			需要的服务 (需接口)	
(String userID) HotelDAO.getHotelBri efInfo(String address) OderInfo.isReserved(St ring ID, String address) HotelDAO.getHotelBri efInfoListBySearching(String[] condition) HotelDAO.getHotelBri efInfoListByQuerying(String[] condition) HotelDAO.getHotelDet ails(String address) OrderInfo.getConderList(String address) OrderInfo.getConderList(String address) OrderInfo.getConderList(String address) OrderInfo.getConderList(String address) OrderInfo.getComment ableOrderList(String userID) OrderInfo.getReserved OrderList(String userID) HotelDAO.update((Hot eIPO hotel) HotelDAO.insert(Hotel PO hotel) HotelDAO.insert(Hotel PO hotel) UserPL insert(UserPO staff) DatabaseFactory.getHot eIDatabase roombl.updateSpareRo  QdayiemEne Agia Ene Agia En	服务名		服务	
HotelDAO.getHotelBri efInfo(String address) OderInfo.isReserved(St ring IID, String address) HotelDAO.getHotelBri efInfoListBySearching(String[] condition) HotelDAO.getHotelBri efInfoListByQuerying(String[] condition) HotelDAO.getHotelDet ails(String address) OrderInfo.getCorderList(String underss) OrderInfo.getComment ableOrderList(String userID) OrderInfo.getReserved OrderList (String userID) HotelDAO.update((Hot eIPO hotel) HotelDAO.insert(Hotel PO hotel) HotelDAO.insert(Hotel PO hotel)  HotelDAO.insert(UserPO staff) DatabaseFactory.getHotel String logister and staff) DatabaseFactory.getHotel page in Fig. 12 pt and staff) DatabaseFactory.getHotel page in Fig. 12 pt and staff)  DatabaseFactory.getHotel page in Fig. 12 pt and staff	_	得到当前	<b>前用户所有订单</b>	
efInfo(String address) OderInfo.isReserved(String ID, String address) HotelDAO.getHotelBri efInfoListBySearching(String[] condition) HotelDAO.getHotelBri efInfoListByQuerying(String[] condition) HotelDAO.getHotelDet ails(String address) OrderInfo.getOrderList(String ID, String ID, String ID, String ID, String address) OrderInfo.getComment ableOrderList(String userID) OrderInfo.getReserved OrderList (String userID) HotelDAO.update((Hot eIPO hotel) HotelDAO.insert(Hotel PO hotel) HotelDAO.insert(Hotel PO hotel) Userbl.insert(UserPO staff) DatabaseFactory.getHotel Polatabase roombl.updateSpareRo  OderInfo.getReserved (Paylish Address)  Agilian Address at paylish Able Polatabase roombl.updateSpareRo  DatabaseFactory.getHotel Polatabase roombl.updateSpareRo  PaylismarEaway (Paylismarking)  R## address at paylish Able Polatabase roombl.updateSpareRo  PaylismarEaway (Paylismarking)  R## address at paylish Able Polatabase roombl.updateSpareRo  Paylismarking Adviss (Paylismarking)  R## address at paylish Able Polatabase roombl.updateSpareRo  Paylismarking Advissory (Paylismarking)  R## address at paylismarking Advissory (Paylismarking)  R##				
OderInfo.isReserved(String ID, String address)  HoteIDAO.getHotelBri efInfoListBySearching(String[] condition)  HoteIDAO.getHotelBri efInfoListByQuerying(String[] condition)  HoteIDAO.getHotelBri efInfoListByQuerying(String[] condition)  HoteIDAO.getHoteIDet ails(String address)  OrderInfo.getOrderList(String address)  OrderInfo.getComment ableOrderList(String userID)  OrderInfo.getReserved OrderList (String userID)  HoteIDAO.update(Hot eIPO hotel)  HoteIDAO.insert(Hotel PO hotel)  UserbL.insert(UserPO staff)  DatabaseFactory.getHot eIDadaser noombl.updateSpareRo  District in the part of the part o	The state of the s	得到单-	一的酒店简要信息持久化对象	
ring ID, String address)  HotelDAO.getHotelBri efInfoListBySearching(String[] condition)  HotelDAO.getHotelBri efInfoListByQuerying(String[] condition)  HotelDAO.getHotelDet ails(String address)  OrderInfo.getOrderList(String userID)  OrderInfo.getReserved OrderList (String userID)  HotelDAO.update((Hot eIPO hotel)  HotelDAO.insert(Hotel Po hotel)  userbL.insert(UserPO staft)  DatabaseFactory.getHotelDet address (Pay Hotel & Warfer App Hotel & W		VD 7:112-12		
HotelDAO.getHotelBri efInfoListBySearching(String[] condition)  HotelDAO.getHotelBri efInfoListByQuerying(String[] condition)  HotelDAO.getHotelDri efInfoListByQuerying(String[] condition)  HotelDAO.getHotelDet ails(String address)  OrderInfo.getCorderList(String userID)  OrderInfo.getComment ableOrderList(String userID)  OrderInfo.getReserved OrderList (String userID)  HotelDAO.update((Hot eIPO hotel))  HotelDAO.find(String address)  HotelDAO.insert(Hotel PO hotel)  userbLinsert(UserPO staff)  DatabaseFactory.getHot eIDatabase roombl.updateSpareRo  更新酒店空房信息	` `	得到该准	自占是	
effnfoListBySearching(String[] condition)  HotelDAO getHotelBri effnfoListByQuerying(String[] condition)  HotelDAO.getHotelDet ails(String address)  OrderInfo.getOrderList(String address)  OrderInfo.getComment ableOrderList(String userID)  OrderInfo.getReserved OrderList (String userID)  HotelDAO.update((Hot eIPO hotel))  HotelDAO.find(String address)  ### ### ### ### ### ### ### ### ### #		40 40 4da #	5. <b>从</b> 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.	
String[] condition) HotelDAO.getHotelBri efInfoListByQuerying(String[] condition) HotelDAO.getHotelDet ails(String address) OrderInfo.getOrderList(String ID, String address) OrderInfo.getComment ableOrderList(String userID) OrderInfo.getReserved OrderList (String userID)  ### ### ### ### ### ### ### ### ### #	ŭ	<b>伙</b> / 据	《余件侍到酒店间安信息持久化刈家列衣	
HotelDAO.getHotelBri efInfoListByQuerying(String[] condition)  HotelDAO.getHotelDet ails(String address) OrderInfo.getOrderList(String ID, String address) OrderInfo.getComment ableOrderList(String userID) OrderInfo.getReserved OrderList (String userID)  ### ### ### ### ### ### ### ### ### #	•			
efInfoListByQuerying(String[] condition)  HotelDAO.getHotelDet ails(String address)  OrderInfo.getOrderList(String address)  OrderInfo.getComment ableOrderList(String userID)  OrderInfo.getReserved OrderList (String userID)  OrderInfo.getReserved OrderList (String userID)  ### ### ### ### ### ### ### ### ### #		担根本毛久从组到满定络而厚自杜九儿动免到丰		
String[] condition) HotelDAO.getHotelDet ails(String address) OrderInfo.getOrderList( String ID, String address) OrderInfo.getComment ableOrderList(String userID) OrderInfo.getReserved OrderList (String userID)  HotelDAO.update((Hot elPO hotel) HotelDAO.insert(Hotel PO hotel) userbl.insert(UserPO staff) DatabaseFactory.getHot elDatabase roombl.updateSpareRo  # 到当前用户可评价酒店对应的订单列表  # 到当前用户所有已执行、已撤销和异常订单  # 提 address 查找单一持久化对象  # 提 address 查找单一持久化对象  # 表 表 表 表 表 表 表 表 表 表 表 表 表 表 表 表 表 表	_	恨掂笡有条件得到酒店间安信息持久化对象列表		
HotelDAO.getHotelDet ails(String address) OrderInfo.getOrderList(String ID, String ID, String address) OrderInfo.getComment ableOrderList(String userID) OrderInfo.getReserved OrderList (String userID)  HotelDAO.update((Hot elPO hotel) HotelDAO.insert(Hotel PO hotel)  HotelDAO.insert(Hotel PO hotel)  Extra daddress at ## A hotelPO 对象  A ## Address at ## A hotelPO 对象  Extra daddress at ## A hotelPO 对象  Extra daddress at ## A hotelPO 对象  A ## Address at ## A hotelPO 对象  Extra daddress at ## A hotelPO 和 hotelPO hotelPO 和 hotelPO hotelPO hotelPO hotelPO hotelPO hotelPO hote				
ails(String address) OrderInfo.getOrderList( String ID, String address) OrderInfo.getComment ableOrderList(String userID) OrderInfo.getReserved OrderList (String userID)  HotelDAO.update((Hot elPO hotel) HotelDAO.insert(Hotel PO hotel)  HotelDAO.insert(Hotel PO hotel)  Examinating address)  HotelDAO.insert(Hotel PO hotel)  Again Address 查找单一持久化对象  Examinating address)  HotelDAO.insert(Hotel PO hotel)  Examinating address 在数据库中插入 hotelPO 对象  Examinating address 在数据库中插入 UserPO 对象  Staff)  DatabaseFactory.getHot elDatabase  Foombl.updateSpareRo  Examinating address 在数据库的服务的引用  Examinating address addr		得到该酒店详细信息的持久化对象		
OrderInfo.getOrderList( String ID, String address)  OrderInfo.getComment ableOrderList(String userID)  OrderInfo.getReserved OrderList (String userID)  OrderInfo.getReserved OrderList (String userID)  HotelDAO.update((Hot elPO hotel)  HotelDAO.find(String address)  HotelDAO.insert(Hotel PO hotel)  userbl.insert(UserPO staff)  DatabaseFactory.getHot elDatabase roombl.updateSpareRo  更新酒店空房信息	_	102101		
String ID, String address)  OrderInfo.getComment ableOrderList(String userID)  OrderInfo.getReserved OrderList (String userID)  HotelDAO.update((Hot elPO hotel)  HotelDAO.find(String address)  HotelDAO.insert(Hotel PO hotel)  userbl.insert(UserPO staff)  DatabaseFactory.getHot elDatabase roombl.updateSpareRo  更新酒店空房信息	`	得到当前	7用户在该酒店的所有订单	
日本の	,			
ableOrderList(String userID)  OrderInfo.getReserved OrderList (String userID)  HotelDAO.update((Hot elPO hotel)  HotelDAO.find(String address)  HotelDAO.insert(Hotel PO hotel)  Example 4	address)			
userID) OrderInfo.getReserved OrderList(String userID) HotelDAO.update((Hot elPO hotel)  HotelDAO.find(String address) HotelDAO.insert(Hotel PO hotel)  userbl.insert(UserPO staff) DatabaseFactory.getHot elDatabase roombl.updateSpareRo  更新曾一持久化对象  程据 address 查找单一持久化对象  在数据库中插入 hotelPO 对象  在数据库中插入 UserPO 对象  是数据库中插入 UserPO 对象  是数据库中插入 UserPO 对象  是数据库的服务的引用	OrderInfo.getComment	得到当前	7用户可评价酒店对应的订单列表	
OrderInfo.getReserved OrderList(String userID) HotelDAO.update((Hot elPO hotel)  HotelDAO.find(String address)  HotelDAO.insert(Hotel PO hotel)  userbl.insert(UserPO staff)  DatabaseFactory.getHot elDatabase roombl.updateSpareRo  # 到当前用户所有已执行、已撤销和异常订单	ableOrderList(String			
OrderList(String userID)  HotelDAO.update((Hot elPO hotel)  HotelDAO.find(String address)  HotelDAO.insert(Hotel PO hotel)  Example 1	userID)			
HotelDAO.update((Hot elPO hotel)  HotelDAO.find(String address)  HotelDAO.insert(Hotel PO hotel)  HotelDAO.insert(Hotel PO hotel)  userbl.insert(UserPO staff)  DatabaseFactory.getHot elDatabase roombl.updateSpareRo  更新单一持久化对象  根据 address 查找单一持久化对象  在数据库中插入 hotelPO 对象  在数据库中插入 UserPO 对象  手数据库中插入 UserPO 对象  更新酒店空房信息	_	得到当前	7用户所有已执行、已撤销和异常订单	
HotelDAO.update((Hot elPO hotel)  HotelDAO.find(String address)  HotelDAO.insert(Hotel PO hotel)  Userbl.insert(UserPO staff)  DatabaseFactory.getHot elDatabase  roombl.updateSpareRo  更新单一持久化对象  根据 address 查找单一持久化对象  在数据库中插入 hotelPO 对象  在数据库中插入 UserPO 对象  手数据库中插入 UserPO 对象  是数据库的服务的引用				
elPO hotel)  HotelDAO.find(String address)  HotelDAO.insert(Hotel PO 对象  PO hotel)  userbl.insert(UserPO 在数据库中插入 UserPO 对象 taff)  DatabaseFactory.getHot elDatabase roombl.updateSpareRo  更新酒店空房信息			LL 1 0 - 1 0	
HotelDAO.find(String address)  HotelDAO.insert(Hotel PO hotel)  userbl.insert(UserPO 在数据库中插入 UserPO 对象 taff)  DatabaseFactory.getHot elDatabase roombl.updateSpareRo  根据 address 查找单一持久化对象  在数据库中插入 hotelPO 对象  在数据库中插入 UserPO 对象  更新酒店空房信息		更新单-	· 持久化对象	
address) HotelDAO.insert(Hotel PO 对象 PO hotel)  userbl.insert(UserPO 在数据库中插入 UserPO 对象 staff)  DatabaseFactory.getHot elDatabase roombl.updateSpareRo  更新酒店空房信息	, , , , , , , , , , , , , , , , , , ,	担护 11	wasa 本批的一块力化对色	
HotelDAO.insert(Hotel PO hotel)  userbl.insert(UserPO 在数据库中插入 UserPO 对象	, ,	恨据 address		
PO hotel) userbl.insert(UserPO	<u></u>	左粉据庞山插 λ hotalDO 对角		
userbl.insert(UserPO	· ·	在剱h/牛中抽八 noteiro 对家		
staff) DatabaseFactory.getHot elDatabase roombl.updateSpareRo 更新酒店空房信息	,	在数据度	E中插入UserPO 对象	
DatabaseFactory.getHot elDatabase roombl.updateSpareRo 更新酒店空房信息	` ·	11 3X VII / I	- 1 1H1/ C 00011 O 1/1 3/	
elDatabase roombl.updateSpareRo 更新酒店空房信息		得到 Hot	el 数据库的服务的引用	
roombl.updateSpareRo 更新酒店空房信息		, , , , , , , , ,		
	roombl.updateSpareRo	更新酒店	5空房信息	
OHI(KOOHI V O TOOHIVO)	om(RoomVO roomvo)			

## 5.3.2.4 roombl 的模块接口规范

	提供的	的服务(供接口)			
RoomInfoService.getSpare	语法	public ArrayList <roomvo> getRoomInfoList (String</roomvo>			
RoomInfoList		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.getAvail	语法	public int getAvailableRoomNum(String address,			
bleRoomNum		Enum <roomtype> roomType, Date day)</roomtype>			
	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.isTimeA	语法	public boolean isTimeAvailable (String addresss,			
vailable (Enum roomType,		Enum <roomtype> roomType, OrderTime time)</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.checkOr	语法	public ResultMessage checkOrder(OrderVO vo)			
der	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.updateS	语法	public boolean updateSpareRoom (String address,			
pareRoom		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.reduceS	语法	public boolean reduceRoom(String address, String			
pareRoom		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.AddSpa	语法	public boolean addRoom(String address, String			
reRoom		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	语法	public ArrayList <roomvo> getRoomInfoList (String</roomvo>			
.getRoomInfoList		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.get	语法	public ArrayList <roomvo> getCheckInList(String</roomvo>			
CheckInList		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.sea	语法	public RoomVO searchCheckInInfo(String			
rchCheckInInfo	ии	address ,Date time)			
Tenencerimino	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.sea	语法	public RoomVO searchCheckInInfo (String			
rchCheckInInfo	ии	address ,Enum <roomtype> 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.Ad	语法	public boolean AddCheckIn(String address, RoomVO			
dCheckIn	714	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.val	语法	public boolean validCheckIn(String address,			
idCheckIn		RoomVO checkIn)			
	前置条件	无			
	后置条件	返回 checkIn 的信息是否符合规范			
UpdateCheckOutService.g	语法	public ArrayList <roomvo> getCheckOutList(String</roomvo>			
etCheckOutList		address)			
	前置条件	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.s	语法	public RoomPO getCheckOutInfo(String address,			
earchCheckOutInfo		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.s	语法	public RoomPO getCheckOutInfo(String address,			
earchCheckOutInfo		Enum <roomtype> 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.a	语法	public boolean AddCheckOut(String address,			
ddCheckOut		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.v	语法	public boolean validCheckIn(String address,			
alidCheckOut	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.getSpareRoom	得到空房信息持久化对象列表				
InfoList(String address)					
RoomDAO.getSpareRoom 得到对应房间类型的空房信息持久化对象					

oList(String address)  RoomDAO.getCheckInInf o(String address, String time)  RoomDAO.getCheckInInf o(String address, String roomType)  RoomDAO.getCheckOutI nfoList(String address)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String roomType)  RoomDAO.updateRoom( RoomPO po)	得到入住信息持久化对象列表 得到对应入住时间的入住信息持久化对象 得到对应房间类型的入住信息持久化对象
RoomDAO.getCheckInInf oList(String address) RoomDAO.getCheckInInf o(String address, String time) RoomDAO.getCheckInInf o(String address, String roomType) RoomDAO.getCheckOutI nfoList(String address) RoomDAO.getCheckOutI nfo(String address, String time) RoomDAO.getCheckOutI nfo(String address, String time) RoomDAO.getCheckOutI nfo(String address, String time) RoomDAO.getCheckOutI nfo(String address, String roomType) RoomDAO.updateRoom( RoomPO po)	得到对应入住时间的入住信息持久化对象
oList(String address)  RoomDAO.getCheckInInf o(String address, String time)  RoomDAO.getCheckInInf o(String address, String roomType)  RoomDAO.getCheckOutI nfoList(String address)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String roomType)  RoomDAO.updateRoom( RoomPO po)	得到对应入住时间的入住信息持久化对象
RoomDAO.getCheckInInf o(String address, String time)  RoomDAO.getCheckInInf o(String address, String roomType)  RoomDAO.getCheckOutI nfoList(String address)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String roomType)  RoomDAO.updateRoom( RoomPO po)	
o(String address, String time)  RoomDAO.getCheckInInf o(String address, String roomType)  RoomDAO.getCheckOutI nfoList(String address)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String roomType)  RoomDAO.updateRoom(RoomPO po)	
time)  RoomDAO.getCheckInInf o(String address, String roomType)  RoomDAO.getCheckOutI nfoList(String address)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String roomType)  RoomDAO.updateRoom( RoomPO po)	得到对应房间类型的入住信息持久化对象
RoomDAO.getCheckInInf o(String address, String roomType)  RoomDAO.getCheckOutI nfoList(String address)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String roomType)  RoomDAO.updateRoom( RoomPO po)	得到对应房间类型的入住信息持久化对象
o(String address, String roomType)  RoomDAO.getCheckOutI nfoList(String address)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String roomType)  RoomDAO.updateRoom(RoomPO po)	得到对应房间类型的人任信息持久化对象
roomType) RoomDAO.getCheckOutI nfoList(String address) RoomDAO.getCheckOutI nfo(String address, String time) RoomDAO.getCheckOutI nfo(String address, String roomType) RoomDAO.updateRoom( RoomPO po)	
RoomDAO.getCheckOutI nfoList(String address) RoomDAO.getCheckOutI nfo(String address, String time) RoomDAO.getCheckOutI nfo(String address, String roomType) RoomDAO.updateRoom( RoomPO po)	
nfoList(String address)  RoomDAO.getCheckOutI nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String roomType)  RoomDAO.updateRoom( RoomPO po)	
RoomDAO.getCheckOutI nfo(String address, String time) RoomDAO.getCheckOutI nfo(String address, String roomType) RoomDAO.updateRoom( RoomPO po)	得到退房信息持久化对象列表
nfo(String address, String time)  RoomDAO.getCheckOutI nfo(String address, String roomType)  RoomDAO.updateRoom( RoomPO po)	
time)  RoomDAO.getCheckOutI nfo(String address, String roomType)  RoomDAO.updateRoom( RoomPO po)	得到对应实际离开时间的退房信息持久化对象
RoomDAO.getCheckOutI nfo(String address, String roomType) RoomDAO.updateRoom( RoomPO po)	
nfo(String address, String roomType)  RoomDAO.updateRoom( RoomPO po)	
roomType) RoomDAO.updateRoom( RoomPO po)	得到对应房间类型的退房信息持久化对象
RoomDAO.updateRoom( RoomPO po)	
RoomPO po)	
	更新单一持久化对象
RoomDAO.insertRoom(R	在数据库插入 Room PO 对象
oomPO po)	
RoomDAO.	在数据库插入 CheckInPO 对象
insertCheckIn(CheckInPO	
po)	
RoomDAO.	在数据库插入 CheckOutPO 对象
insertCheckOut(CheckOut	
PO po)	
FactoryService.getRoomD	用工厂创建一个 RoomDAO 对象
AO()	
Factory.createStrategyInfo	用工厂创建一个 StrategyInfoService 对象
Service()	
Factory.createRoomInfoSe	用工厂创建一个 RoomInfoService 对象
rvice()	
Factory.createAvailableRo	用工厂创建一个 AvailableRoomService 对象
omService()	
StrategyInfoServiceisRig	—————————————————————————————————————
htName(address)	/ 4-21 184/ - 124   D 14 / C   1 1 1 / N
	7.47.11.2.1.4 11.7.2 11.3.7.1
etHotelDetails(address)	得到某个酒店的基本信息
StrategyInfoServiceisRig htName(address)	判断输入的名称是否正确

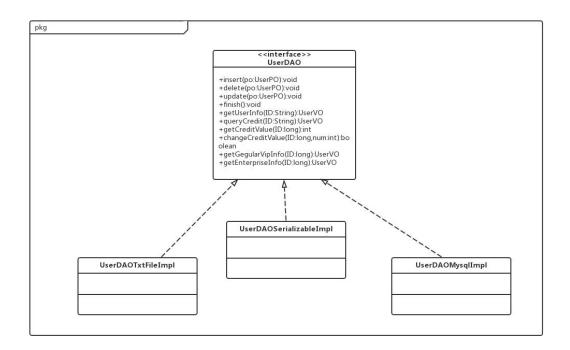
## 5.3.2.5 strategybl 的模块接口规范

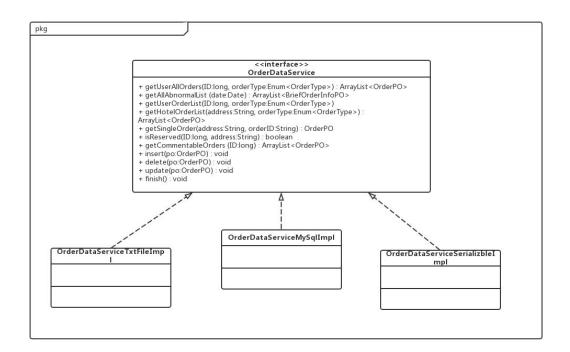
	提供的	的服务(供接口)
StrategyInfoService.getAv	语法	public String .getAvailblePromotionName (OrderVO
ailble PromotionName		order)
	前置条件	无
	后置条件	返回某订单能享受的唯一酒店促销策略折扣名称,
		如果没有,则返回 null
StrategyInfoService.getAv	语法	public String getAvailbleMarketStrategyName
ailbleMarketStrategyName		(OrderVO order)
	前置条件	无
	后置条件	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.	语法	public int getBestDiscount(OrderVO order)
getBestDiscount	前置条件	无
	后置条件	返回某订单的最终的折扣百分比
StrategyInfoService.isRigh	语法	public boolean isRightName(String name)
tName	前置条件	无
	后置条件	判断一个名称格式是否正确
UpdateStrategyService.get	语法	public ArrayList <strategyvo> getStrategyList(String</strategyvo>
StrategyList		address, Enum <strategytype> 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.get	语法	public StrategyVO getStrategyInfo(String address,
StrategyInfo		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.ad	语法	public boolean add(String address, StrategyVO
d		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.mo	语法	public boolean modify(String address, StrategyVO
dify		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.del	语法	public boolean delete(String address, StrategyVO
ele		strategy)
	前置条件	该 StrategyVO 已存在
	后置条件	返回是否删除成功
UpdateStrategyService.Val	语法	public boolean Valid(String address, StrategyVO
id		strategy)
	前置条件	无
	后置条件	返回该策略信息是否符合规范
VerifyEnterpriseVipImpl.v	语法	public boolean verifyEnterpriseMember(String
erifyEnterpriseMember		hotelName, String enterpriseName, String

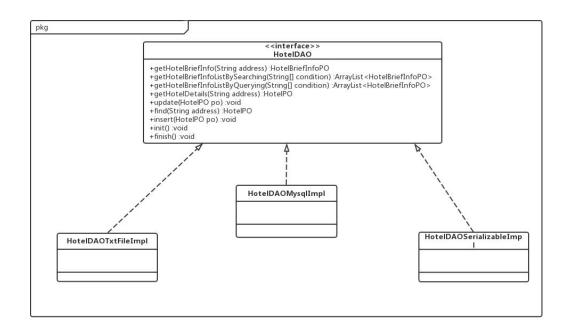
		securityCode)	
	前置条件	securityCode 输入符合规范	
	后置条件	返回验证是否通过	
UpdateStraegyService.veri	语法 public boolean verifyTradeArea(String city,		
fyTradeArea	.,,,,	StrategyVO strategyVO)	
-,	前置条件	用户制定或修改商圈折扣	
	后置条件	验证特定商圈会员专属折扣的商圈名称在某城市	
	// · · · · · · · · · · · · · · · · · ·	是否存在	
UpdateStraegyService.get	语法	public ArrayList <businessdistrictpo></businessdistrictpo>	
BusinessDistrictList		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	· 约服务(需接口)	
服务名			
StrategyDAO.getStrategy	得到策略折	扣持久化对象列表	
List(String address, String			
StrategyType)			
StrategyDAO.getStrategyI	得到对应折扣名称的折扣策略持久化对象		
nfo(String address, String			
strategyName)			
StrategyDAO.updateStrate	更新单一持久化对象		
gy(StrategyPO po)			
StrategyDAO.insertStrateg	在数据库插入 StrategyPO 对象		
y(StrategyPO po)			
StrategyDAO.deleteStrate	在数据库删除某 StrategyPO 对象		
gy(StrategyPO po)			
StrategyDAO.EnterpriseM	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		
ember(String			
enterpriseName, String			
securityCode)	ᄱᆉᄮ	VIII elemente	
HotelInfoService.	得到某个城	市的商圈列表	
getBusinessDistrictList(Str			
ing city)	但不少至宁力	ン¥/m ├ 白	
hotelInfoService.getHotel	得到酒店的详细信息		
Details(String address)	但如此落入日尺百		
User.getRegularVipInfo(St	得到普通会员信息		
ring ID)	2000年100日	品/de 自	
User.getEnterpriseVipInfo( String ID)	得到企业会员信息		
FactoryService.getStrategy	从丁厂纸净	StrategyDAO 对象	
DAO()	从工厂新建 StrategyDAO 对象		
FactoryService.createHote	从丁厂新建	一个 HotelInfoSeivice 对象	
IInfoService()		110000000000 /11/200	
11110001 1100()			

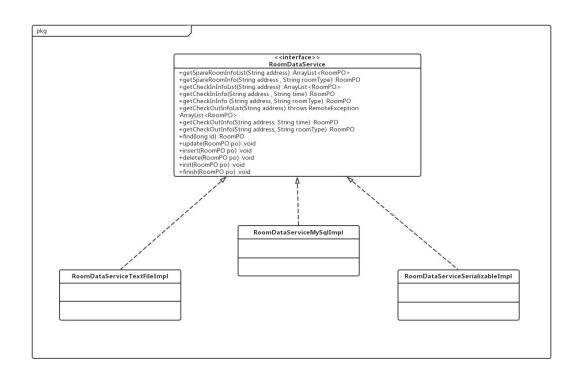
#### 5.4 数据层的分解

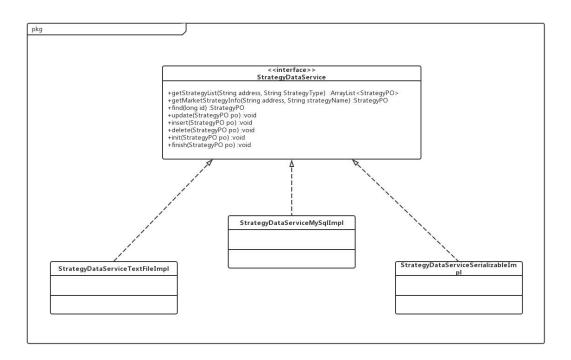
数据层主要给业务逻辑层提供数据访问服务,包括对持久化数据的增、删、改、查。User 业务逻辑需要的服务由 userDAO 接口提供,Order 业务逻辑需要的服务由 orderDAO 接口提供,Hotel 业务逻辑需要的服务由 hotelDAO 接口提供,Room 业务逻辑需要的服务由 roomDAO 接口提供,strategy 业务逻辑需要的服务由 strategyDAO 接口提供。由于持久化数据的保存可能存在多种形式:Txt 文件、序列化文件等,所示抽象了数据服务。数据层模块的描述具体如下图所示。











### 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
vsanDAOCanializahlaFilaInanl	基于序列化文件的持久化数据接口,提供集体
userDAOSerializableFileImpl	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
usciDAOMysqiiiipi	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
r	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 持久化数据的接口,提供集体载入、集体保存、
hotelDAO	持久化致据的接口, 提供集体软入、集体保存、 增、改、查服务
	基于 Txt 文件的持久化数据的接口,提供集体
hotelDAOTxtFileImpl	载入、集体保存、增、改、查服务
	基于序列化文件的持久化数据接口,提供集体
hotelDAOSerializableFileImpl	载入、集体保存、增、改、查服务
	基于 Mysql 数据库的持久化数据接口,提供集
hotelDAOMysqlImpl	体载入、集体保存、增、改、查服务
	持久化数据的接口,提供集体载入、集体保存、
roomDAO	增、改、查服务
roomDAOTxtFileImpl	基于 Txt 文件的持久化数据的接口,提供集体
100mDAOTAU Hemipi	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
Tooling/Toolerunzuoler nemipi	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
- J-1 F	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 基工 R + 全体 + 体 + 体 + 体 + 体 + 体 + 体 + 体 + 体 +
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 其子序列化文件的特点化数据接口。提供集体
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
	基于 Mysql 数据库的持久化数据接口,提供集
strategyDAOMysqlImpl	基丁 Mysqi 数据库的符入化数据接口,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	<b>共的服务</b>	(供接口			
userDAO.getUserInfo	语法	public	UserPO	getUserInfo(String	userID)	throws
	旧仏	Remote	Exception	l		

	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.getHotelStaffIn fo	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	publicArrayList <creditrecordpo>queryCreditRecord(StringuserID)throwsRemoteException;</creditrecordpo>
	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
	语法	public void insertUser(UserPO po) throws RemoteException
userDAO.insertUser	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			
	语法	public void updateUser(UserPO po) throws RemoteException			
userDAO.updateUser	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			
	语法	public boolean updateClient(ClientInfoPO clientInfoPO, String oldUserID) throws RemoteException;			
userDAO.updateClient	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			
	语法	public boolean signRegularVip(RegularVipPO regularVipPO) throws RemoteException;			
userDAO.signRegularVip	前置条件	regularVipPO 不为空			
	后置条件	在会员中增加一条记录			
	语法	public boolean signEnterpriseVip(EnterpriseVipPO enterpriseVipPO) throws RemoteException;			
userDAO. signEnterpriseVip	前置条件	enterpriseVipPO 不为空			
	后置条件	在会员中增加一条记录			
	语法	public RegularVipPO getRegularVipInfo(String userID) throws RemoteException;			
userDAO. getRegularVipInfo	前置条件	userID 不为空			
	后置条件	无			
	语法	public EnterpriseVipPO getEnterpriseVipInfo(String userID) throws RemoteException;			
userDAO. getEnterpriseVipInfo	前置条件	userID 不为空			
	后置条件	无			
userDAO.	语法	public boolean updateRegularVipInfo(RegularVipPO regularVipPO) throws RemoteException;			
updateRegularVipInfo	前置条件	regularVipPO 不为空			

后置条件	更新对应的会员信息
------	-----------

	提住	<b>共的服务(供接口)</b>
		public ArrayList <brieforderinfopo></brieforderinfopo>
	语法	getUserOrderList(String userID, Enum <ordertype></ordertype>
orderDAO.getUserOrder		orderType) throws RemoteException; 无
List	前置条件	)u
	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)
	语法	public ArrayList <orderpo> getUserAllOrders(String</orderpo>
1 540 41 410 1		userID) throws RemoteException;
orderDAO.getUserAllOrd ers	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
	语法	public ArrayList <orderpo> getUserOrdersByHotel</orderpo>
1 540 44 0 1		(String userID, String address) throws
orderDAO.getUserOrders ByHotel	前置条件	RemoteException; ID 有效, address 有效
Byrioter	加重次日	
	后置条件	返回该用户在该酒店所有订单的列表
	语法	public OrderPO getSingleOrder(String address, String
orderDAO.getSingleOrde		orderID) throws RemoteException; 无
r	前置条件	<i>/</i> L
	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
	语法	public boolean isReserved(String userID, String address) throws RemoteException;
orderDAO.isReserved	前置条件	无
	后置条件	返回该酒店是否被该用户预定过
orderDAO.	语法	public ArrayList< OrderPO > getCommentableOrders (String userID) throws RemoteException;
getCommentableOrders	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
	语法	public void insertOrder (OrderPO po) throws RemoteException;
OrderDAO.insertOrder	前置条件	无
	后置条件	在数据库中增加一个 po 记录
	语法	public void deleteOrder (OrderPO po) throws RemoteException;
OrderDAO.deleteOrder	前置条件	无
	后置条件	删除一个 po
	语法	public void updateOrder (OrderPO po) throws RemoteException;
OrderDAO.updateOrder	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
	语法	public ArrayList <brieforderinfopo> getAllAbnormalList (Date date) throws RemoteException;</brieforderinfopo>
OrderDAO.getAllAbnor malList	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)
	语法	public ArrayList <brieforderinfopo> getHotelOrderList(String address, Enum<ordertype> orderType) throws RemoteException</ordertype></brieforderinfopo>
OrderDAO.getHotelOrder List	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
	语法	public boolean setOrderCommented(String orderID) throws RemoteException;
OrderDAO.setOrderCom mented	前置条件	orderID 不为空
	后置条件	无
	语法	public OrderPO getDetailedOrder(String orderID) throws RemoteException;
OrderDAO.getDetailedOr der	前置条件	orderID 不能为空
	后置条件	无

提供的服务(供接口)					
	语法	public HotelBriefInfoPO getHotelBriefInfo(String address) throws RemoteException;			
HotelDAO.getHotelBriefInfo	前置条件	无			
	后置条件	按 address 进行查找返回相应的 HotelBriefInfoPO 结果			
HotelDAO.getHotelBriefI	语法	public ArrayList <briefhotelinfopo> getHotelBriefInfoListBySearching (String[] condition, ArrayList<brieforderinfopo> orderedHotelList) throws RemoteException;</brieforderinfopo></briefhotelinfopo>			
nfoListBySearching	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.getHotelBriefI	语法	public ArrayList <hotelbriefinfopo> getHotelBriefInfoListByQuerying (String[] condition, ArrayList<brieforderinfopo> orderedHotelList) throws RemoteException;</brieforderinfopo></hotelbriefinfopo>			
nfoListByQuerying	前置条件	无			
	后置条件	按照 condition 进行查找返回相应的所有 HotelBriefInfoPO			
	语法	public HotelPO getHotelDetails(String address) throws RemoteException;			
HotelDAO.getHotelDetail s	前置条件	无			
	后置条件	根据 address 进行查找返回相应的 HotelPO			
	语法	public void updateHotel(HotelPO po) throws RemoteException;			
HotelDAO.updateHotel	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			
	语法	public void insertHotel(HotelPO po) throws RemoteException;			
HotelDAO.insertHotel	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Ⅰ	的服务(供接口)
RoomDAO.getSpareRoom	语法	public ArrayList <roompo></roompo>
InfoList		getSpareRoomInfoList(String address, Date day)
		throws RemoteException;
	前置条件	无
	后置条件	返回空房列表
RoomDAO.getSpareRoom	语法	public RoomPO getSpareRoomInfo(String address,
Info		Enum <roomtype> roomType, Date day) throws</roomtype>
		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.getCheckInInf	语法	public ArrayList <roompo></roompo>
oList		getCheckInInfoList(String address) throws
		RemoteException
	前置条件	无
	后置条件	返回入住信息列表
RoomDAO.getCheckInInf	语法	public ArrayList <roompo> getCheckInInfo(String</roompo>
0		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.getCheckInInf	语法	public RoomPO getCheckInInfo (String address,
0		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.getCheckOutI	语法	public ArrayList <roompo></roompo>
nfoList		getCheckOutInfoList(String address) throws
		RemoteException
	前置条件	无
	后置条件	返回退房信息列表
RoomDAO.getCheckOutI	语法	public ArrayList <roompo> getCheckOutInfo(String</roompo>
nfo		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.getCheckOutI	语法	public RoomPO getCheckOutInfo(String address,
nfo		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.insertCheckOu	语法	public void insertCheckOut(CheckOutPO po) throws
t		RemoteException;
	前置条件	无
	后置条件	在离店表里插入一条记录
	i	的服务(供接口)
StrategyDAO.getStrategy	语法	public ArrayList <strategypo> getStrategyList(String</strategypo>
List		address, Enum <strategytype> strategyType) throws</strategytype>
		RemoteException;
	前置条件	无
	后置条件	返回某种策略折扣列表
StrategyDAO.getStrategyI	语法	public StrategyPO getStrategyInfo(String address,
nfo		Enum <strategytype> strategyType, String</strategytype>
		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.updateStrate	语法	public boolean updateStrategy(StrategyPO po) throws
gy		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.insertStrateg	语法	public void insertStrategy(StrategyPO po) throws
y		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.deleteStrate	语法	public void deleteStrategy(StrategyPO po) throws
gy		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	语法	public boolean verifyEnterpriseMember(String
verifyEnterpriseMember		enterpriseName, String securityCode) throws
		RemoteException;
	前置条件	无
	后置条件	无
	š	·

# 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: 2222222222: 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	Туре	Nu11	Кеу	Default	Extra
userID	char (20)	NO	PRI	NULL	
password	bigint(20)	YES		NULL	
te1Num	int (11)	YES		NULL	
creditValue	int(11)	YES		NULL	
isMember	char (16)	YES		NULL	

#### commomMember 表:

Field Type	Nu11	Key	Default	Extra	
------------	------	-----	---------	-------	--

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

#### creditRecord 表:

Field	Туре	Nu11	Кеу	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	Туре	Nu11	Кеу	Default	Extra
userID	char (20)	YES	MUL	NULL	
enterpriseID	char (20)	YES		NULL	

## hotelStaff 表:

Field	Туре	Nu11	Кеу	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	Туре	Nu11	Key	Default	Extra
userID	char (20)	NO	PRI	NULL	
password	char(16)	YES		NULL	
te1Num	bigint(20)	YES		NULL	

#### webMarketStaff 表:

Field	Туре	Nu11	Кеу	Default	Extra
userID	char (20)	NO	PRI	NULL	
password	char (16)	YES		NULL	
telNum	bigint(20)	YES		NULL	

## hotel 表:

Field	Туре	Nu11	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	Туре	Nu11	Key	Default	Extra
hotelAddress	char (50)	YES	MUL	NULL	
clientName	char (20)	YES		NULL	
comment	char (200)	YES		NULL	

## roomTypeAndPrice表:

Field	Туре	Nu11	Кеу	Default	Extra
address	char (50)	YES	MUL	NULL	
roomType	int (11)	YES		NULL	
price	int(11)	NO		NULL	

## OrderInfo表:

Field	Туре	Nu11	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	Туре	Nu11	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	Туре	Nu11	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	Туре	Nu11	Кеу	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	Туре	Nu11	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	Туре	Nu11	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	Туре	Nu11	Кеу	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	Туре	Nu11	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	