

# 酒店预订管理系统

## System of Booking Hotel

### 体系结构设计文档

### V 1.5

小组 37    陈统 刚昭 高岳 贺云青

## 更新历史

日期	更新者	更新内容	版本号
2016. 10. 9	全体组员	第一版本初稿	V 0. 1
2016. 10. 11	全体组员	拆分酒店模块，增加评价模块，修改依赖关系，增加部分图	V 0. 2
2016. 10. 14	全体组员	增加了接口规范	V 0. 3
2016. 10. 14	高岳	增加了界面跳转图，业务逻辑层的职责说明，信息视角	V 0. 4
2016. 10. 15	贺云青	增加了数据层的职责说明	V 1. 0
2016. 10. 15	高岳	增加 PO 说明	V 1. 1
2016. 10. 15	陈统	做了一些内容错误的修改	V 1. 2
2016. 10. 15	刚昭	修正 HotelDataService 部分语法的错误返回类型	V 1. 3
2016. 10. 16	陈统	更新数据库表，持久化对象并修正一些接口返回值小错误	V 1. 4
2016. 10. 18	刚昭	修正用户界面层分解部分的错别字	V 1. 5

# 目录

1. 引言 .....	4
1.1 编制目的 .....	4
1.2 词汇表 .....	4
1.3 参考资料 .....	4
2. 产品描述 .....	4
3. 逻辑视角 .....	4
4. 组合视角 .....	6
4.1 开发包图 .....	6
4.2 运行时进程 .....	8
4.3 物理部署 .....	9
5. 接口视角 .....	9
5.1 模块的职责 .....	9
5.2 用户界面层的分解 .....	11
5.2.1 用户界面层模块的职责 .....	12
5.2.2 用户界面模块的接口规范 .....	12
5.2.3 用户界面模块设计原理 .....	12
5.3 业务逻辑层的分解 .....	12
5.3.1 业务逻辑层模块的职责 .....	13
5.3.2 业务逻辑模块的接口模范 .....	13
5.4 数据层的分解 .....	21
5.4.1 数据层模块的职责 .....	21
5.4.2 数据层模块的接口规范 .....	22
6. 信息视角 .....	27
6.1 数据持久化对象 .....	27
6.2 数据库表 .....	28

# 1. 引言

## 1.1 编制目的

本报告详细完成对酒店预订管理系统的概要设计，达到指导详细设计和开发的目的，同时实现测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户编写，是了解系统的导航。

## 1.2 词汇表

缩写或单词	解释
SBH	酒店预订管理系统
user	用户
hotel	酒店
order	订单
promotion	促销策略
comment	评价
level	用户等级
HotelFilter	酒店筛选信息

## 1.3 参考资料

《酒店预订管理系统用例文档》

《酒店预订管理系统需求规格说明文档》

# 2. 产品描述

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

# 3. 逻辑视角

酒店预订管理系统中，选择了分层体系结构的风格，将系统分为 3 层（展示层、业务逻辑层、数据层）能够很好的示意整个高层抽象。展示层包括 GUI 页面

的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如图 1 和图 2 所示。

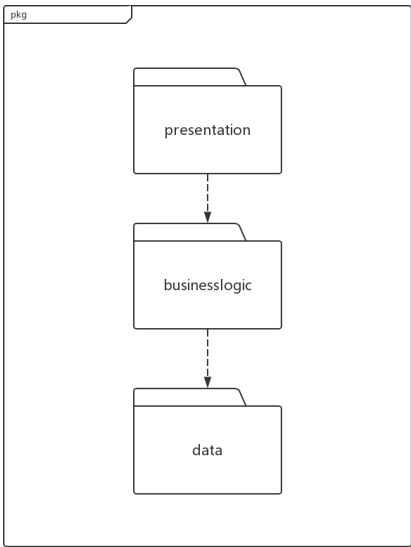


图 1：分层结构

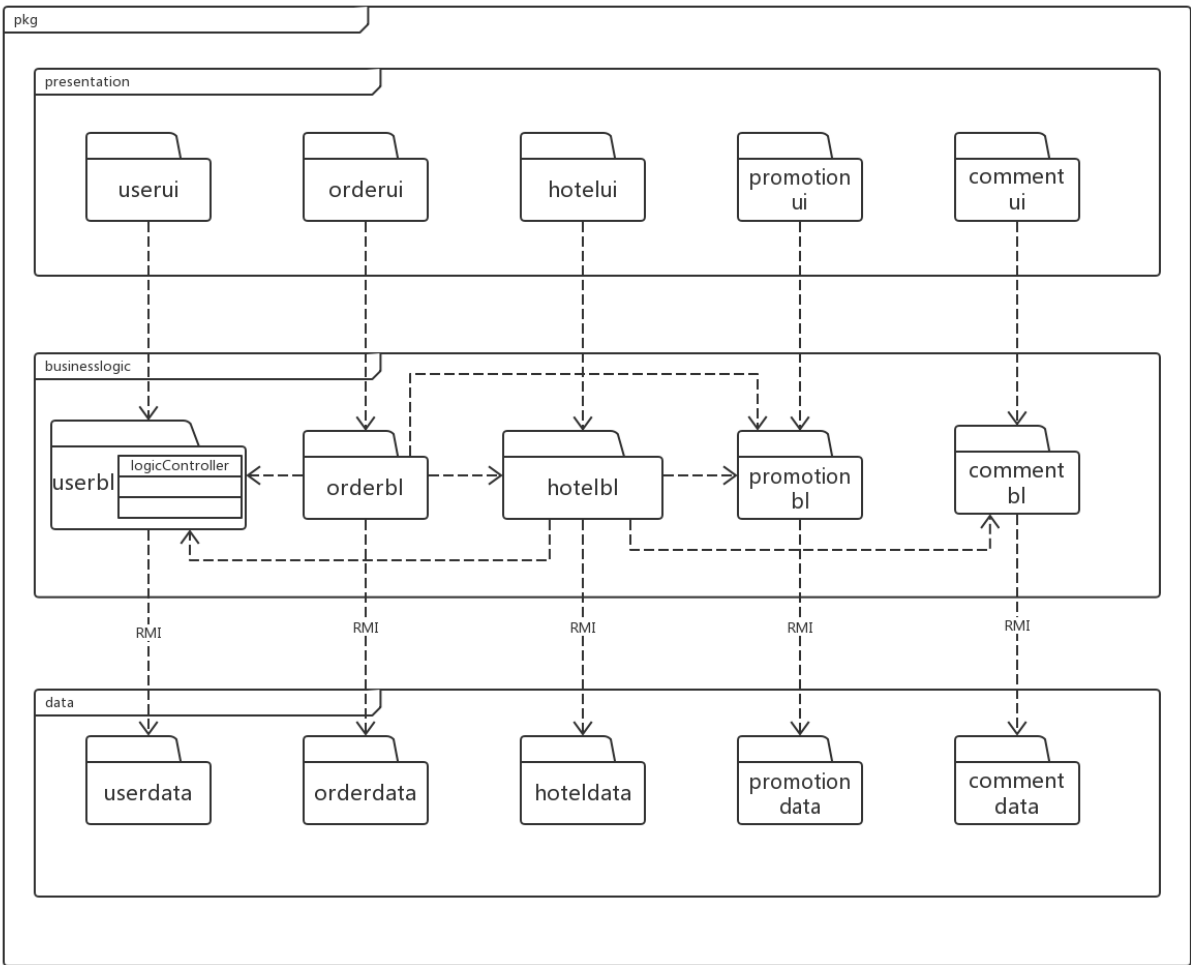


图 2：逻辑设计方案

## 4. 组合视角

### 4.1 开发包图

表 1 酒店预订管理系统的最终开发包设计

开发（物理）包	依赖的其他开发包
mainui	userui, orderui, hotelui, promotionui, vo
userui	userblservice, 界面类库包, vo
userblservice	
userbl	userblservice, userdataservice, po
userdataservice	Java RMI, po
userdata	userdataservice , databaseutility, po
orderui	orderblservice, 界面类库包, vo
orderblservice	
orderbl	orderdataservice, orderblservice, userbl, hotelbl, promotionbl, po
orderdataservice	Java RMI, po
orderdata	orderdataservice , databaseutility, po
hotelui	hotelblservice, 界面类库包, vo
hotelblservice	
hotelbl	hoteldataservice, hotelblservice, commentbl, orderbl, userbl, promotionbl, po
hoteldataservice	Java RMI, po
hoteldata	hoteldataservice , databaseutility, po
commentui	commentblservice, 界面类库包, vo
commentblservice	
commentbl	commentdataservice, commentblservice, po
commentdataservice	Java RMI, po
commentdata	commentdataservice, databaseutility, po
promotioniui	promotionblservice, 界面类库包, vo
promotionblservice	
promotionbl	promotiondataservice, promotionblservice, po
promotiondataservice	Java RMI, po
promotiondata	promotiondataservice , databaseutility, po
vo	
po	
界面类库包	
databasesutility	JDBC
Java RMI	

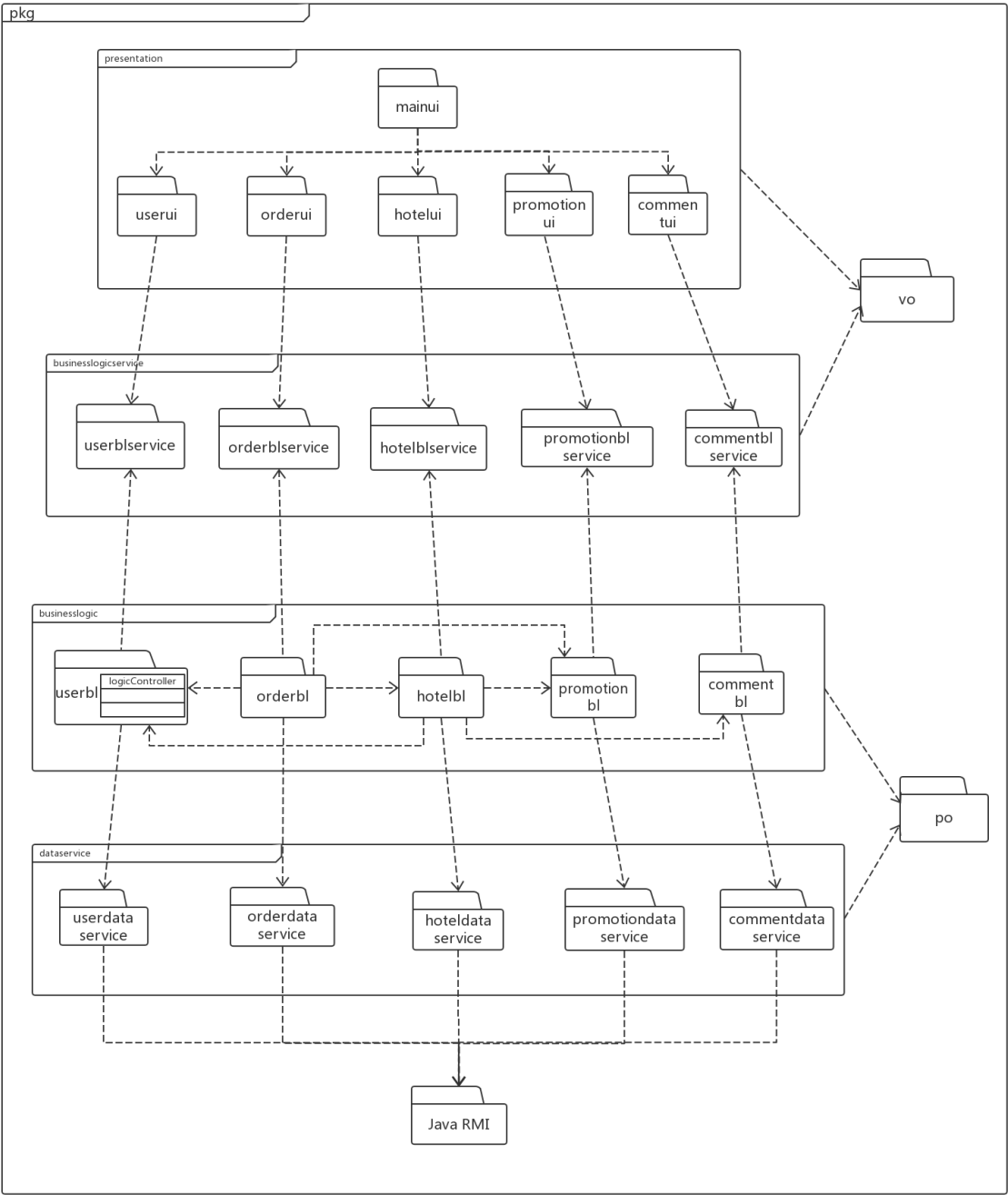


图 3 客户端开发包图

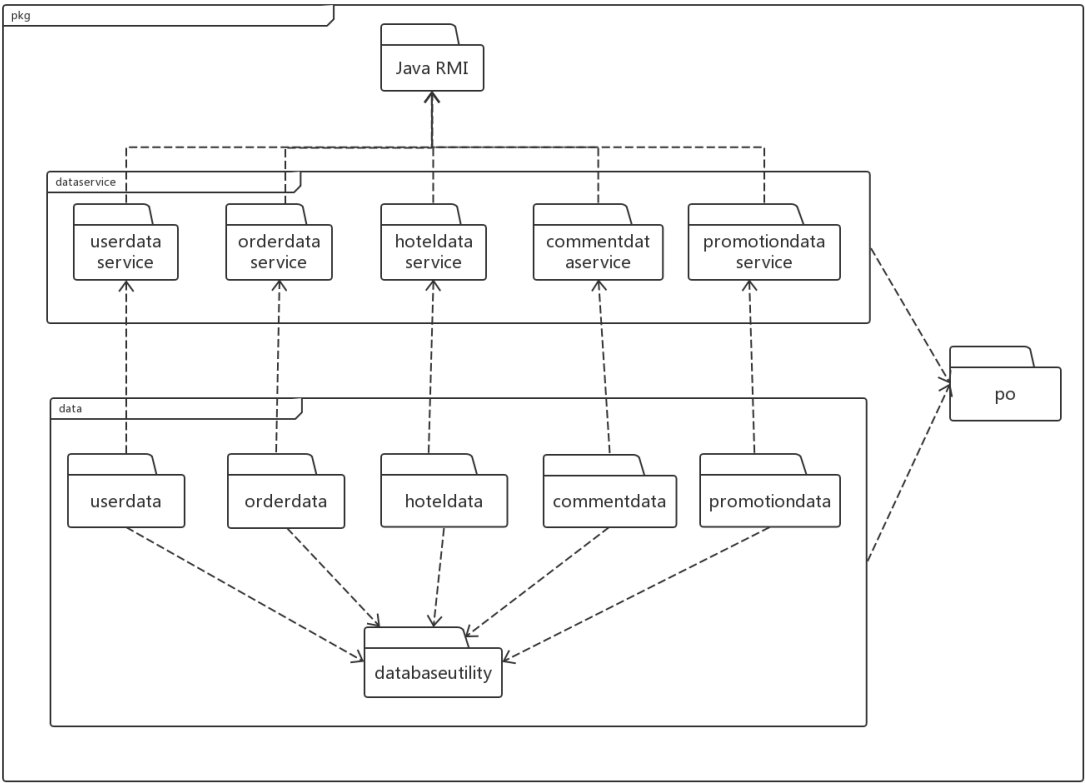


图 4 服务器端开发包图

## 4.2 运行时进程

在酒店预订管理系统中，会有多个客户端进程和一个服务器端进程，其进程图如图 5 所示。结合部署图，客户端进程是在客户端机器上运行，服务器端进程是在服务器端机器上运行。

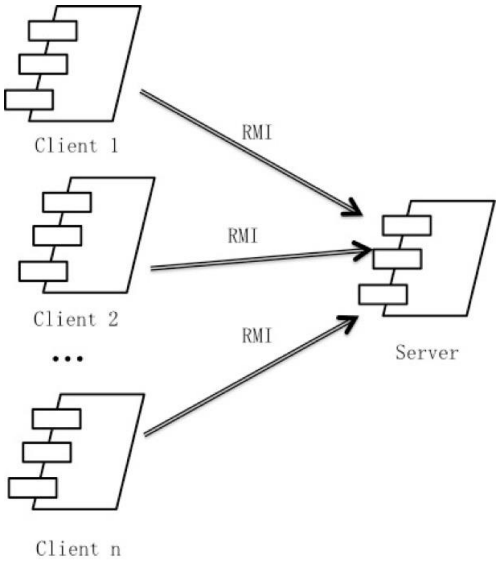


图 5 进程图



### 4.3 物理部署

酒店预订管理系统系统中客户端构件是放在客户端机上，服务器端构件是放在服务器端机器上。部署图如图 6 所示。

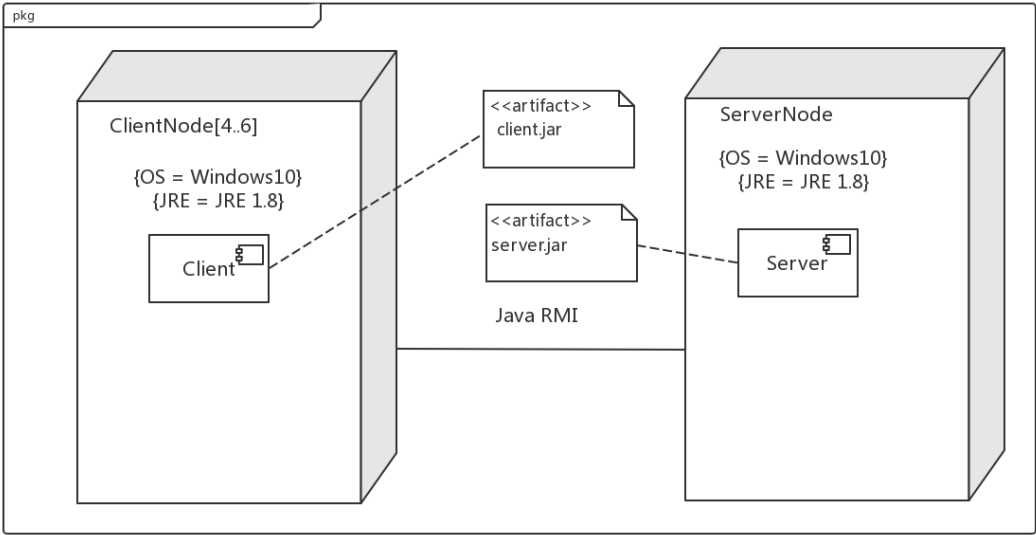


图 6 部署图

## 5. 接口视角

### 5.1 模块的职责

客户端模块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端各层的职责分别如表 2 和表 3 所示。

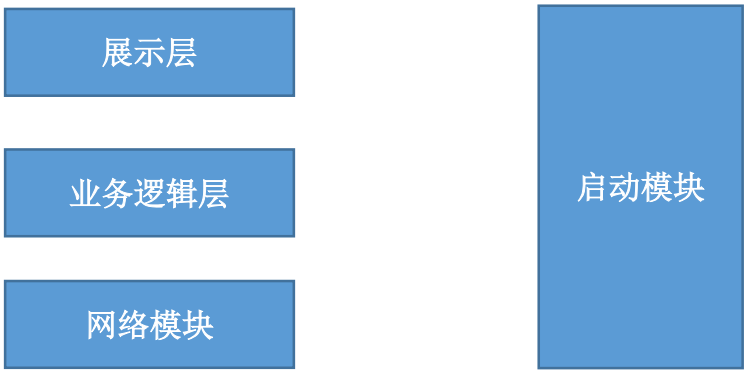


图 7 客户端模块视图

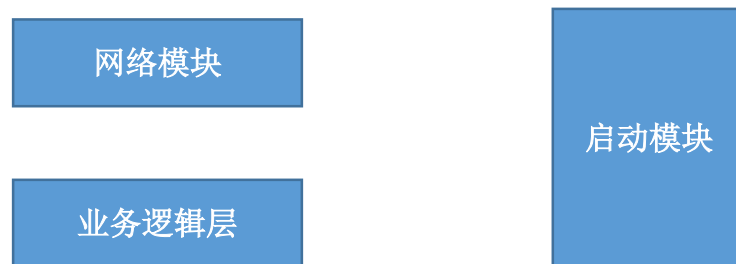


图 8 服务器端模块视图

表 2 客户端各层的职责

层	职责
启动模块	负责初始化网络通信机制，启动用户界面
用户界面层	基于窗口的进销存管理系统客户端用户界面
业务逻辑层	对于用户界面的输入进行响应并进行业务处理逻辑
客户端网络模块	利用 java RMI 机制查找 RMI 服务

表 3 服务器端各层的职责

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

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

表 4 层之间调用的接口

接口	服务调用方	服务提供方
userblservice orderblservice hotelblservice promotionblservice commentblservice	客户端展示层	客户端业务逻辑层
userdataservice orderdataservice hoteldataservice promotiondataservice commentdataservice	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 37 个用户界面（加粗为主界面）：**登录界面**、**客户主界面**、查看客户信息界面、修改客户信息界面、浏览酒店界面、查看信用记录界面、浏览客户未执行订单界面、浏览客户历史订单界面、生成订单界面、酒店详细信息界面、未执行订单详细信息界面、**酒店工作人员主界面**、维护酒店信息界面、管理酒店促销策略界面、执行订单界面、浏览酒店历史订单界面、修改酒店促销策略界面、异常订单详细信息界面、已撤销订单详细信息界面、已执行订单详细信息界面、更新退房信息界面、**网站营销人员主界面**、信用充值界面、修改会员等级制度界面、网站促销策略界面、浏览异常订单界面、修改网站促销策略界面、**网站管理人员主界面**、用户管理界面、新增酒店界面、浏览客户信息界面、浏览网站营销人员信息界面、浏览酒店工作人员信息界面、新增酒店工作人员界面、修改客户信息界面、修改网站营销人员界面、新增网站营销人员界面、新增酒店工作人员界面

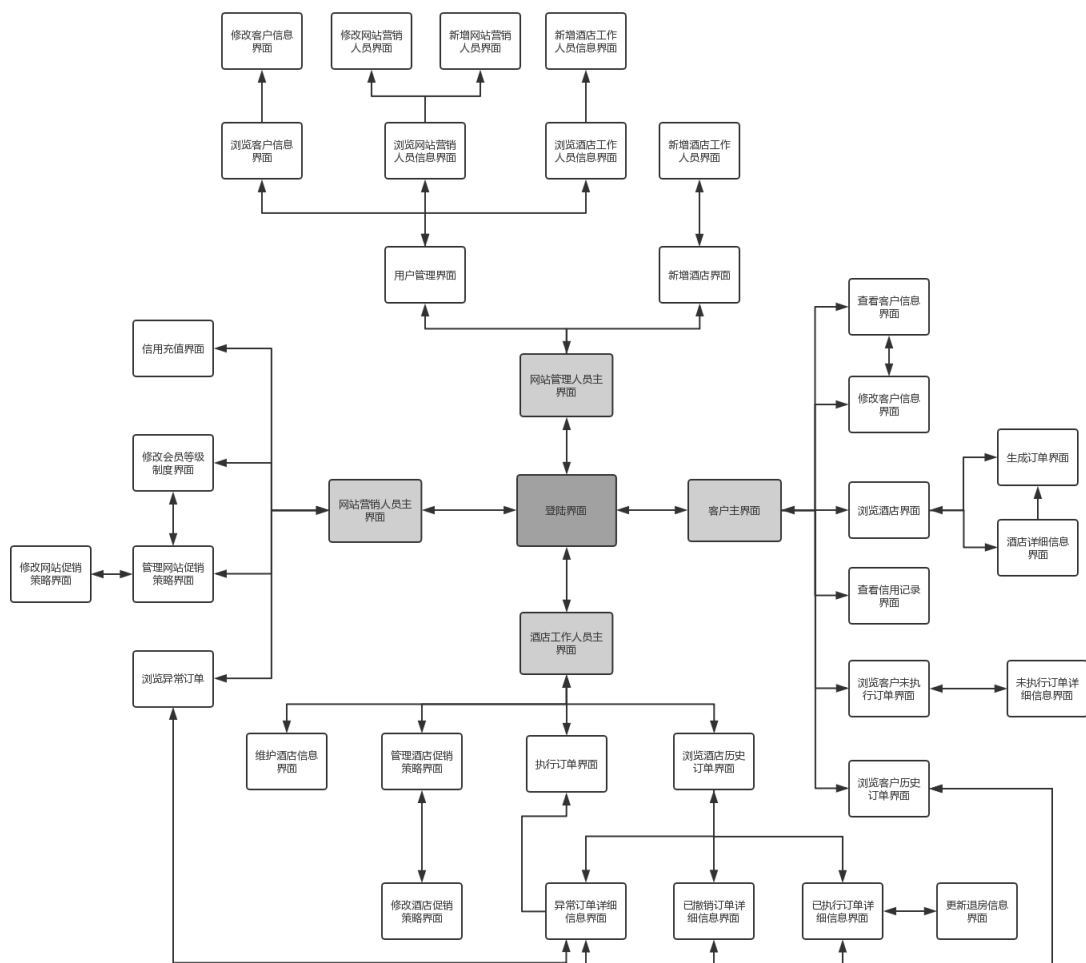


图 9 酒店预订管理系统界面跳转图

### 5.2.1 用户界面层模块的职责

表 5 用户界面层模块的职责

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

### 5.2.2 用户界面模块的接口规范

表 6 用户界面层模块的接口规范

MainFrame	语法	init(args:String[])
	前置条件	用户触发相应需要显示界面的条件
	后置条件	显示 Frame 并且加载 LoginPanel

表 7 用户界面层模块需要的服务接口

服务名	服务
blservice.Userblservice	登录界面的业务逻辑接口
blservice.*blservice	每个 ui 模块会有对应的数据逻辑接口

### 5.2.3 用户界面模块设计原理

用户界面拟用 javaFX 来实现。

### 5.3 业务逻辑层的分解

业务逻辑层包括多个针对界面及业务逻辑处理对象。例如 User 对象负责处理登录界面的业务逻辑。业务逻辑层的设计如下图所示：

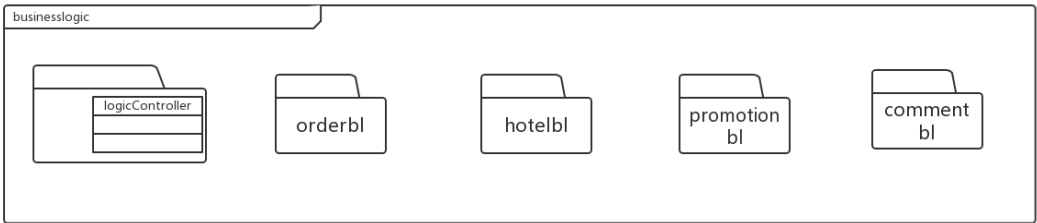


图 10 业务逻辑层的设计

### 5.3.1 业务逻辑层模块的职责

表 8 业务逻辑层模块的职责

模块	职责
userbl	实现登录与对用户账户信息增删改查的服务
orderbl	实现对订单进行增删改查操作的服务
hotelbl	实现对酒店信息进行增改查和对酒店可用客房进行修改的服务
promotionbl	实现对网站促销策略、酒店促销策略和会员登记制度进行制定、修改、增加的服务
commentbl	实现客户对订单进行评价的服务

### 5.3.2 业务逻辑模块的接口模范

#### 5.3.2.1 userbl 模块的接口规范

提供的服务（供接口）		
User.login	语法	<code>public ResultMessage login(String userName, String password)</code>
	前置条件	账号、密码符合语法规则
	后置条件	系统确认账号密码是否正确，若正确，进入主界面并记录登录信息，否则提示错误信息
User.logout	语法	<code>public ResultMessage logout(String ID)</code>
	前置条件	系统已登录
	后置条件	系统退出登录，记录登出信息
User.add	语法	<code>public ResultMessage add(UserVO vo)</code>
	前置条件	账户信息符合规则且齐全
	后置条件	系统创建该账户并持久化增加该账户数据
User.findbyID	语法	<code>public UserVO findbyID(String ID)</code>
	前置条件	账户 ID 符合输入语法要求
	后置条件	系统返回查找到的搜索结果
User.findbyUserName	语法	<code>public UserVO findbyUserName(String userName)</code>

	前置条件	账户名符合输入语法要求
	后置条件	系统返回查找到的搜索结果
User.modifyPassword	语法	public ResultMessage modifyPassword(String userName, String oldPassword, String newPassword)
	前置条件	新密码符合输入语法要求, 旧密码匹配正确
	后置条件	系统修改该账户的密码数据
User.modify	语法	public ResultMessage modify(UserVO vo)
	前置条件	账户信息符合规则且齐全
	后置条件	系统修改该账户并持久化保存该账户数据
User.addCreditRecord	语法	public ResultMessage addCreditRecord (CreditRecordVO vo)
	前置条件	该账户 ID 为客户
	后置条件	系统增加该客户信用记录并持久化保存数据
User.findCreditRecord	语法	public ArrayList<CreditRecordVO> findCreditRecord(String ID)
	前置条件	该账户 ID 为客户且输入符合语法要求
	后置条件	返回该账户所有信用
需要的接口（需接口）		
服务名	服务内容	
UserDataService. add(UserPO po)	根据提供的用户 po 增加持久化对象	
UserDataService. findbyID(String ID)	根据用户 ID 查找持久化对象	
UserDataService. findbyUserName(String userName)	根据 UserName 查找持久化对象	
UserDataService. modify (UserPO po)	修改单一持久化对象	
UserDataService. Check(String userName, String password)	检查账户密码是否正确	
UserDataService.	根据提供的信用记录 po 增加持久化对象	

<b>addCreditRecord (CreditRecordPO po)</b>	
<b>UserDataService. findCreditRecord (String ID)</b>	根据用户 ID 查找所有信用记录的持久化对象
<b>UserDataService.init()</b>	初始化持久化数据库
<b>UserDataService.finish()</b>	结束持久化数据库的使用

### 5.3.2.2 orderbl 模块的接口规范

#### 提供的服务（供接口）

<b>Order.createOrder</b>	语法	<b>public ResultMessage createOrder(OrderVO vo)</b>
	前置条件	orderIfo 包含完整且正确的信息
	后置条件	创建和保存新的未执行订单
<b>Order.cancelOrder</b>	语法	<b>public ResultMessage cancelOrder (String orderId)</b>
	前置条件	存在未执行订单
	后置条件	系统成功/失败撤销一条未执行订单
<b>Order.getOrder</b>	语法	<b>public OrderVO getOrder (String ID)</b>
	前置条件	ID 输入符合规范
	后置条件	返回一个订单的 vo
<b>Order. getAbnormalOrder</b>	语法	<b>public ArrayList&lt;OrderVO&gt; getAbnormalOrder ()</b>
	前置条件	无
	后置条件	返回当日所有异常订单的 vo
<b>Order.getUserOrderList</b>	语法	<b>public ArrayList&lt;OrderVO&gt; getUserOrderList (String userID, OrderType type)</b>
	前置条件	无
	后置条件	系统返回某客户某种订单的订单列表
<b>Order. getHotelOrderList</b>	语法	<b>public ArrayList&lt;OrderVO&gt; getHotelOrderList(String hotelID, OrderType type)</b>
	前置条件	无
	后置条件	系统返回某酒店某种订单的订单列表
<b>Order.executeOrder</b>	语法	<b>public ResultMessage executeOrder</b>

		(String orderID)
	前置条件	存在未执行订单
	后置条件	未执行订单被置为已执行订单，信用记录被更新
Order. cancelAbnormalOrder	语法	public ResultMessage cancelAbnormalOrder (String orderID, Boolean isHalf)
	前置条件	存在异常订单
	后置条件	系统将异常订单更新为已撤销订单，记录撤销时间，并更新信用值和信用记录
需要的接口（需接口）		
服务名	服务内容	
User.addCreditRecord (CreditRecordVO vo)	增加该客户信用记录并持久化保存数据	
User.findbyID (String ID)	返回一个 UserVO	
Promotion. getPromotion (String hotelAddress, String userID)	返回符合该酒店该客户的所有促销策略（包括酒店促销策略与网站促销策略）	
Hotel.getHotelVO(String hotelAddress)	返回一个酒店的 vo	
Hotel.changeRoom( RoomType type, int num)	系统更新酒店的可用客房信息并保存酒店的 po	
OrderDataService. add (OrderPO po)	系统添加新的订单记录	
OrderDataService. getOrderPO(String orderID)	系统返回一条订单信息	
OrderDataService. getUserOrderPO (String userID, OrderType type)	返回某客户某种类全部订单 po	
OrderDataService. getHotelOrderPO (String hotelID, OrderType type)	返回某酒店某种类全部订单 po	
OrderDataService. getAbnormalOrderPO	返回当日全部异常订单 po	



(Date date)	
OrderDataService. modify (OrderPO po)	更新一个 po

### 5.3.2.3 hotelbl 模块的接口规范

提供的服务（供接口）

Hotel.findHotels	语法	public ArrayList<HotelVO> findHotels(HotelFilter filter )
	前置条件	无
	后置条件	系统返回该筛选条件下的所有酒店 vo
Hotel.getHotel	语法	public HotelVO getHotel(String hotelAddress)
	前置条件	无
	后置条件	系统返回一个该酒店的 vo
Hotel.addHotel	语法	public ResultMessage addHotel(HotelVO hotelVO, UserVO userVO)
	前置条件	酒店信息、酒店工作人员信息输入合法
	后置条件	系统增加酒店和酒店工作人员的持久化对象
Hotel.modifyHotel	语法	public ResultMessage modifyHotel(HotelVO vo)
	前置条件	酒店信息输入合法
	后置条件	系统更新酒店 po
Hotel.changeRoom	语法	public ResultMessage changeRoom(RoomType type, int num)
	前置条件	房间类型数量输入合法
	后置条件	系统更新酒店 po
Hotel.getHotelVO	语法	public HotelVO getHotelVO(String hotelAddress)
	前置条件	存在对应酒店
	后置条件	返回一个酒店的 vo
Hotel.getCitys	语法	public ArrayList<CityVO> getCitys()

	前置条件	无
	后置条件	返回系统中所有城市的名称
Hotel.getAreas	语法	<code>public ArrayList&lt;AreaVO&gt; getAreas(CityVO city)</code>
	前置条件	城市名称存在
	后置条件	返回系统中该城市的所有商圈名称
需要的接口（需接口）		
服务名	服务内容	
Promotion. getPromotion (String hotelAddress,String userID)	根据酒店地址和客户 ID 返回促销策略	
HotelDataService. addHotelInfo (HotelPO po)	增加酒店的持久化对象	
HotelDataService. modifyHotelInfo (HotelPO po)	修改酒店的持久化对象	
HotelDataService. getHotelInfo (String hotelAddress)	根据酒店地址返回单一的酒店持久化对象	
HotelDataService. findHotels(HotelFilter filter)	系统返回该筛选条件下的所有酒店 vo	
User.add (UserVO vo)	增加酒店工作人员的持久化对象	
Comment. getComments (String hotelAddress)	根据酒店地址返回该酒店的所有评论的持久化对象	

#### 5.3.2.4 promotionbl 模块的接口规范

提供的服务（供接口）		
Promotion.add	语法	<code>public ResultMessage add(PromotionVO vo)</code>
	前置条件	系统登录账户为酒店工作人员或网站营销人员

	后置条件	系统持久化增加该促销策略数据
Promotion. getPromotion	语法	<code>public ArrayList&lt;PromotionVO&gt; getPromotion (String hotelAddress, String userID)</code>
	前置条件	输入的地址名称、客户 ID 符合文本规则
	后置条件	系统根据酒店地址返回符合该酒店该客户订单的所有促销策略（包括酒店促销策略与网站促销策略）
Promotion. showHotelPromotion	语法	<code>public ArrayList&lt;PromotionVO&gt; showHotelPromotion (String hotelAddress)</code>
	前置条件	输入的地址名称符合文本规则
	后置条件	系统根据酒店地址返回该酒店的所有促销策略
Promotion. showWebsitePromotion	语法	<code>public ArrayList&lt;PromotionVO&gt; showWebsitePromotion()</code>
	前置条件	无
	后置条件	系统返回所有网站促销策略
Promotion.modify	语法	<code>public ResultMessage modify (PromotionVO vo)</code>
	前置条件	填写的内容符合规范且齐全
	后置条件	系统修改该策略并持久化存储该策略数据
Promotion.delete	语法	<code>public ResultMessage delete (PromotionVO vo)</code>
	前置条件	无（删除表明该数据必然是存在的）
	后置条件	系统永久性删除该策略的数据
Promotion.showLevel	语法	<code>public LevelVO showLevel ()</code>
	前置条件	无
	后置条件	系统返回用户等级信息
Promotion.modifyLevel	语法	<code>public ResultMessage modifyLevel (LevelVO vo)</code>
	前置条件	输入的内容符合规范且齐全
	后置条件	系统修改用户等级策略并持久化存储该策略数据
需要的接口（需接口）		

服务名	服务内容
PromotionDataService. add (PromotionPO po)	根据提供的促销策略 po 增加持久化对象
PromotionDataService. modify (PromotionPO po)	修改单一促销策略持久化对象
PromotionDataService. showHotelPromotion (String hotelAddress)	根据 hotelAddress 查找该酒店的所有持久化对象
PromotionDataService. showWebsitePromotion ( )	查找所有网站营销策略
PromotionDataService. delete (PromotionPO po)	删除单一持久化对象
PromotionDataService. showLevel ( )	返回用户等级策略的持久化对象
PromotionDataService. modifyLevel (LevelPO po)	修改单一用户等级策略的持久化对象
PromotionDataService. init()	初始化持久化数据库
PromotionDataService. finish()	结束持久化数据库的使用

### 5.3.2.5 commentbl 模块的接口规范

提供的服务（供接口）

Comment.getComments	语法	public ArrayList<CommentVO> getComment(String hotelAddress)
	前置条件	存在该酒店的评价
	后置条件	返回该酒店的所有评价
Comment.addComment	语法	public ResultMessage addComment(CommentVO vo)
	前置条件	评价输入合法
	后置条件	系统增加该评价的持久化对象
需要的接口（需接口）		
服务名	服务内容	
CommentDataService. find(String hotelAddress)	根据酒店地址返回多个评价的持久化对象	

```
CommentDataService.  
insert(Com  
mentPO po)
```

在数据库中插入单一的评价的持久化对象

## 5.4 数据层的分解

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的增、删、改、查。例如，Userbl 的服务主要由 UserDataService 提供。持久化数据将以数据库的形式被保存。以 UserDataService 为例，数据层模块的具体描述下图所示：

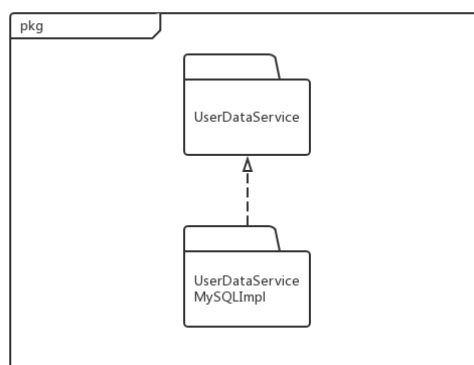


图 11 数据层模块的描述

### 5.4.1 数据层模块的职责

数据层模块职责如下表所示

表 9 数据层模块的职责

模块	职责
Userdataservice	负责保存用户账户的信息，进行增删改查的操作
OrderDataService	负责保存各类订单，供读取修改写入
HotelDataservice	负责保存酒店的信息，供读取修改写入
PromotionDataService	负责保存促销策略，供读取修改写入
CommentDataService	负责保存评价信息，供读取和写入

## 5.4.2 数据层模块的接口规范

### 5.4.2.1 UserDataService 模块的接口规范

提供的服务（供接口）		
UserDataService.add	语法	<code>public ResultMessage add (UserPO po) throws RemoteException</code>
	前置条件	同样的账户在数据库中不存在
	后置条件	在数据库中增加一个 UserPO 记录
UserDataService.findbyID	语法	<code>public UserPO findbyID(String ID) throws RemoteException</code>
	前置条件	无
	后置条件	按 ID 进行查找返回相应的 UserPO 记录
UserDataService. findbyUserName	语法	<code>public UserPO findbyUserName(String userName) throws RemoteException</code>
	前置条件	无
	后置条件	按 UserName 进行查找返回相应的 UserPO 记录
UserDataService.modify	语法	<code>public ResultMessage modify (UserPO po) throws RemoteException</code>
	前置条件	在数据库中存在同样的 PO
	后置条件	更新一个 PO
UserDataService.Check	语法	<code>public ResultMessage Check(String userName, String password) throws RemoteException</code>
	前置条件	账号、密码合法
	后置条件	系统检查是否存在账户账户密码是否对应
UserDataService. addCreditRecord	语法	<code>public ResultMessage addCreditRecord(CreditRecordPO po) throws RemoteException</code>
	前置条件	无
	后置条件	在数据库中增加一个 CreditRecordPO 记录
UserDataService. findCreditRecord	语法	<code>public CreditRecordPO findCreditRecord (String ID) throws RemoteException</code>
	前置条件	无

	后置条件	按 ID 进行查找返回相应的 CreditRecordPO 记录
UserDataService.init	语法	public void init() throws RemoteException
	前置条件	无
	后置条件	初始化持久化数据库
UserDataService.finish	语法	public void finish() throws RemoteException
	前置条件	无
	后置条件	结束持久化数据库的使用

#### 5.4.2.2 orderDataService 模块的接口规范

##### 提供的服务（供接口）

OrderDataService. getOrderPO	语法	public OrderPO getOrderPO(String orderID) throws RemoteException
	前置条件	无
	后置条件	系统返回一条订单信息
OrderDataService. getUserOrderPO	语法	public ArrayList<OrderPO> getUserOrderPO (String userID, OrderType type) throws RemoteException
	前置条件	无
	后置条件	返回某客户某种类全部订单 po
OrderDataService. getHotelOrderPO	语法	public ArrayList<OrderPO> getHotelOrderPO (String hotelID, OrderType type) throws RemoteException
	前置条件	无
	后置条件	返回某酒店某种类全部订单 po
OrderDataService. getAbnormalOrderPO	语法	public ArrayList<OrderPO> getAbnormalOrderPO (Date date) throws RemoteException
	前置条件	无
	后置条件	返回当日全部异常订单 po
OrderDataService. add	语法	public ResultMessage add (OrderPO po) throws RemoteException
	前置条件	该订单在数据库中不存在

	后置条件	系统返回添加结果记录
OrderDataService. modify	语法	public ResultMessage modify (OrderPO po) throws RemoteException
	前置条件	在数据库中存在同样订单号的 po
	后置条件	更新一个 po
OrderDataService.init	语法	public void init() throws RemoteException
	前置条件	无
	后置条件	初始化持久化数据库
OrderDataService.finish	语法	public void finish() throws RemoteException
	前置条件	无
	后置条件	结束持久化数据库的使用

#### 5.4.2.3 HotelDataService

##### 提供的服务（供接口）

HotelDataService.init	语法	public void initial() throws RemoteException
	前置条件	无
	后置条件	初始化持久化数据库
HotelDataService. addHotelInfo	语法	public ResultMessage addHotelInfo(HotelPO po) throws RemoteException
	前置条件	无
	后置条件	在数据库中插入一个 po
HotelDataService. modifyHotelInfo	语法	public ResultMessage modifyHotelInfo(HotelPO po) throws RemoteException
	前置条件	数据库中存在一个相同地址信息的 po
	后置条件	在数据库中更新一个 po
HotelDataService. getHotelInfo	语法	public HotelPO getHotelInfo(String hotelAddress) throws RemoteException
	前置条件	无
	后置条件	返回一个酒店 po



<b>HotelDataService. findHotels</b>	语法	<code>public ArrayList&lt;HotelPO&gt; findHotels(HotelFilter filter) throws RemoteException</code>
	前置条件	无
	后置条件	返回多个酒店 po
<b>HotelDataService.getCitys</b>	语法	<code>public ArrayList&lt;CityPO&gt; getCitys() throws RemoteException</code>
	前置条件	无
	后置条件	在数据库中获得所有 CityPO
<b>HotelDataService.getAreas</b>	语法	<code>public ArrayList&lt;AreaPO&gt; getAreas(CityPO po) throws RemoteException</code>
	前置条件	无
	后置条件	根据 CityPO 返回所有 AreaPO
<b>HotelDataService.finish</b>	语法	<code>public void finish() throws RemoteException</code>
	前置条件	无
	后置条件	结束持久化数据库的使用

#### 5.4.2.4 PromotionDataService 模块的接口规范

提供的服务（供接口）		
<b>PromotionDataService. add</b>	语法	<code>public ResultMessage add (PromotionPO po) throws RemoteException</code>
	前置条件	同样的 PromotionPO 在数据库中不存在
	后置条件	在数据库中增加一个 PromotionPO 记录
<b>PromotionDataService. modify</b>	语法	<code>public ResultMessage modify (PromotionPO po) throws RemoteException</code>
	前置条件	在数据库中存在同样的 PO
	后置条件	更新一个 PO
<b>PromotionDataService. showHotelPromotion</b>	语法	<code>public ArrayList&lt;PromotionPO&gt; showHotelPromotion (String hotelAddress) throws RemoteException</code>
	前置条件	无
	后置条件	按 hotelAddress 进行查找返回该酒店的 PromotionPO 记录
<b>PromotionDataService.</b>	语法	<code>public ArrayList&lt;PromotionPO&gt;</code>

<b>showWebsitePromotion</b>		<b>showWebsitePromotion () throws RemoteException</b>
	前置条件	无
	后置条件	返回网站策略的 PromotionPO 记录
<b>PromotionDataService. delete</b>	语法	<b>public ResultMessage delete (PromotionPO po) throws RemoteException</b>
	前置条件	在数据库中存在同样的 PO
	后置条件	删除一个 PO
<b>PromotionDataService. showLevel</b>	语法	<b>public LevelPO showLevel () throws RemoteException</b>
	前置条件	无
	后置条件	返回 LevelPO 记录
<b>PromotionDataService. modifyLevel</b>	语法	<b>public ResultMessage modifyLevel (LevelPO po) throws RemoteException</b>
	前置条件	在数据库中存在同样的 PO
	后置条件	更新一个 PO
<b>PromotionDataService. init</b>	语法	<b>public void init() throws RemoteException</b>
	前置条件	无
	后置条件	初始化持久化数据库
<b>PromotionDataService. finish</b>	语法	<b>public void finish() throws RemoteException</b>
	前置条件	无
	后置条件	结束持久化数据库的使用

#### 5.4.2.5 CommentDataService

##### 提供的服务（供接口）

<b>CommentDataService. init</b>	语法	<b>public void initial() throws RemoteException</b>
	前置条件	无
	后置条件	初始化持久化数据库
<b>CommentDataService. find</b>	语法	<b>public ArrayList&lt;CommentPO&gt; find(String hotelAddress) throws RemoteException</b>
	前置条件	无

	后置条件	按地址进行查找返回相应的 CommentPO 结果
CommentDataService. insert	语法	public ResultMessage insert(CommentPO po) throws RemoteException
	前置条件	无
	后置条件	在数据库中插入一个 po
CommentDataService. finish	语法	public void finish() throws RemoteException
	前置条件	无
	后置条件	结束持久化数据库的使用

## 6. 信息视角

### 6.1 数据持久化对象

系统的 PO 类就是对应的相关的实体类，在此只做简单介绍：

- UserPO 类为用户信息，包含用户名、用户 ID、密码、用户类型、其他个人信息属性
- OrderPO 类为订单信息，包含开始时间、退房时间、最晚订单执行时间、房间类型及数量、预计入住人数、有无儿童、订单状态、撤销时间（已撤销状态订单）。入住酒店名称、酒店地址、订单号属性
- HotelPO 类为酒店信息，包含地址、所属商圈、简介、设施服务、价格、星级、评分、可用客房信息（可用客房的类型、数量、原始价格）、房间入住信息（房间号、入住时间、预计离开时间）属性
- PromotionPO 类为促销策略，包含折扣额度、适用日期、适用人员、适用条件属性
- LevelPO 类为会员等级制度，包含每一等级所需信用值属性
- Comment PO 类为酒店评价，包含评价人，评价时间，用户打分，具体评价内容
- CreditRecordPO 为信用记录类，包含信用记录的时间，订单号，动作（订单执行、订单异常、订单撤销、充值），信用度变化、信用度结果属性
- CityPO 类为城市信息
- AreaPO 类为商圈信息
- RoomPO 类为一个酒店的房间信息，包含房间类型，数量，价格

持久化用户对象 LevelPO 类的定义如图所示：

```
1 package org.common.po;
2
3 import java.io.Serializable;
4
5
6 public class LevelPO implements Serializable {
7
8     /**
9      *
10     */
11     private static final long serialVersionUID = 2360607519386548756L;
12
13     /**
14      * 等级的总个数
15     */
16     public int levelNum;
17
18     /**
19      * 每一等级需要的信用值
20     */
21     public ArrayList<Integer> credits;
22
23     public LevelPO(int levelNum, ArrayList<Integer> credits){
24         this.levelNum = levelNum;
25         this.credits = credits;
26     }
27
28     public LevelPO(int levelNum){
29         this.levelNum = levelNum;
30     }
31
32     public LevelPO(ArrayList<Integer> credits){
33         this.credits = credits;
34     }
35 }
36
```

## 6.2 数据库表

本系统使用数据库来存储文件，数据库表包含 User 表、Order 表、Hotel 表、Room 表、Promotion 表、Level 表、Comment 表、City 表、Area 表、Credit 表