



---

# 体系结构描述文档

---

互联网酒店预订系统



2016-10-6

秋刀鱼的鱼

更新历史

| 修改人员 | 日期         | 变更原因        | 版本号  |
|------|------------|-------------|------|
| 马晗蕾  | 2016/10/06 | 建立初稿        | V0.0 |
| 马晗蕾  | 2016/10/09 | 整合成员各自负责的部分 | V1.0 |
| 马晗蕾  | 2016/10/09 | 审核后的修改      | V1.1 |
| 马晗蕾  | 2016/10/10 | 数据库表单的修改    | V1.2 |
|      |            |             |      |
|      |            |             |      |
|      |            |             |      |
|      |            |             |      |

目录

|                |   |
|----------------|---|
| 1.引言 .....     | 3 |
| 1.1编制目的 .....  | 3 |
| 1.2词汇表 .....   | 3 |
| 1.3 参考资料 ..... | 3 |
| 2. 产品描述 .....  | 3 |

|                                       |    |
|---------------------------------------|----|
| 3.逻辑视角 .....                          | 3  |
| 4.组合视角 .....                          | 5  |
| 4.1.开发包图 .....                        | 5  |
| 4.2运行时进程 .....                        | 10 |
| 4.3物理部署 .....                         | 11 |
| 5.接口视角 .....                          | 11 |
| 5.1模块的职责 .....                        | 11 |
| 5.2.用户界面层的分解 .....                    | 12 |
| 5.2.1.用户界面层模块的职责 .....                | 13 |
| 5.2.2.用户界面模块的接口规范 .....               | 13 |
| 5.2.3.用户界面模块设计原理 .....                | 15 |
| 5.3.业务逻辑层的分解 .....                    | 15 |
| 5.3.1.业务逻辑层模块的职责 .....                | 16 |
| 5.3.2.业务逻辑模块的接口规范 .....               | 16 |
| 5.4.数据层的分解 .....                      | 23 |
| 5.4.1.数据层模块的职责 .....                  | 23 |
| 5.4.2.数据层模块的接口规范 .....                | 23 |
| 6.信息视角 .....                          | 27 |
| 6.1数据持久化对象 .....                      | 27 |
| 6.2数据库表 .....                         | 28 |
| 客户信息表clientInfo .....                 | 28 |
| 信用记录表 creditRecord .....              | 28 |
| 订单信息表 orderInfo .....                 | 29 |
| 评价信息表 assessingInfo .....             | 29 |
| 酒店信息表 hotelInfo .....                 | 30 |
| 酒店跟合作企业匹配表 hotelMatchEnterprise ..... | 30 |
| 酒店促销策略信息表 hotelStrategy .....         | 30 |
| 房间信息表 roomInfo .....                  | 31 |
| 网站促销策略信息表 websiteStrategy .....       | 31 |
| 用户信息表userInfo .....                   | 31 |

# 1.引言

## 1.1编制目的

本报告详细完成对互联网酒店预定系统的概要设计，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

| 词汇名称  | 词汇含义     | 备注 |
|-------|----------|----|
| _ui   | 表示某展示层   |    |
| _bl   | 表示某逻辑层   |    |
| _data | 表示某数据层   |    |
| RMI   | 表示远程方法调用 |    |

## 1.3 参考资料

- 1.IEEE std 1471-2000
- 2.丁二玉，刘钦.计算与软件工程（卷二）[M]机械工业出版社2012： 134—182

# 2. 产品描述

互联网酒店管理预定系统用例文档和互联网酒店管理预定系统软件规格说明中对产品的概括描述。

# 3.逻辑视角

互联网酒店管理预定系统中，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。

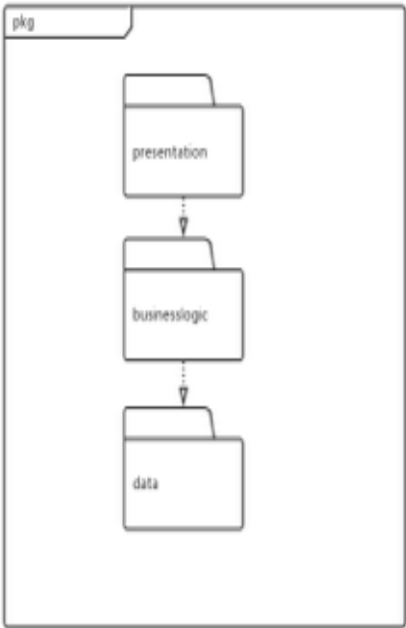


图 1 参照体系结构风格的包图表达逻辑视角

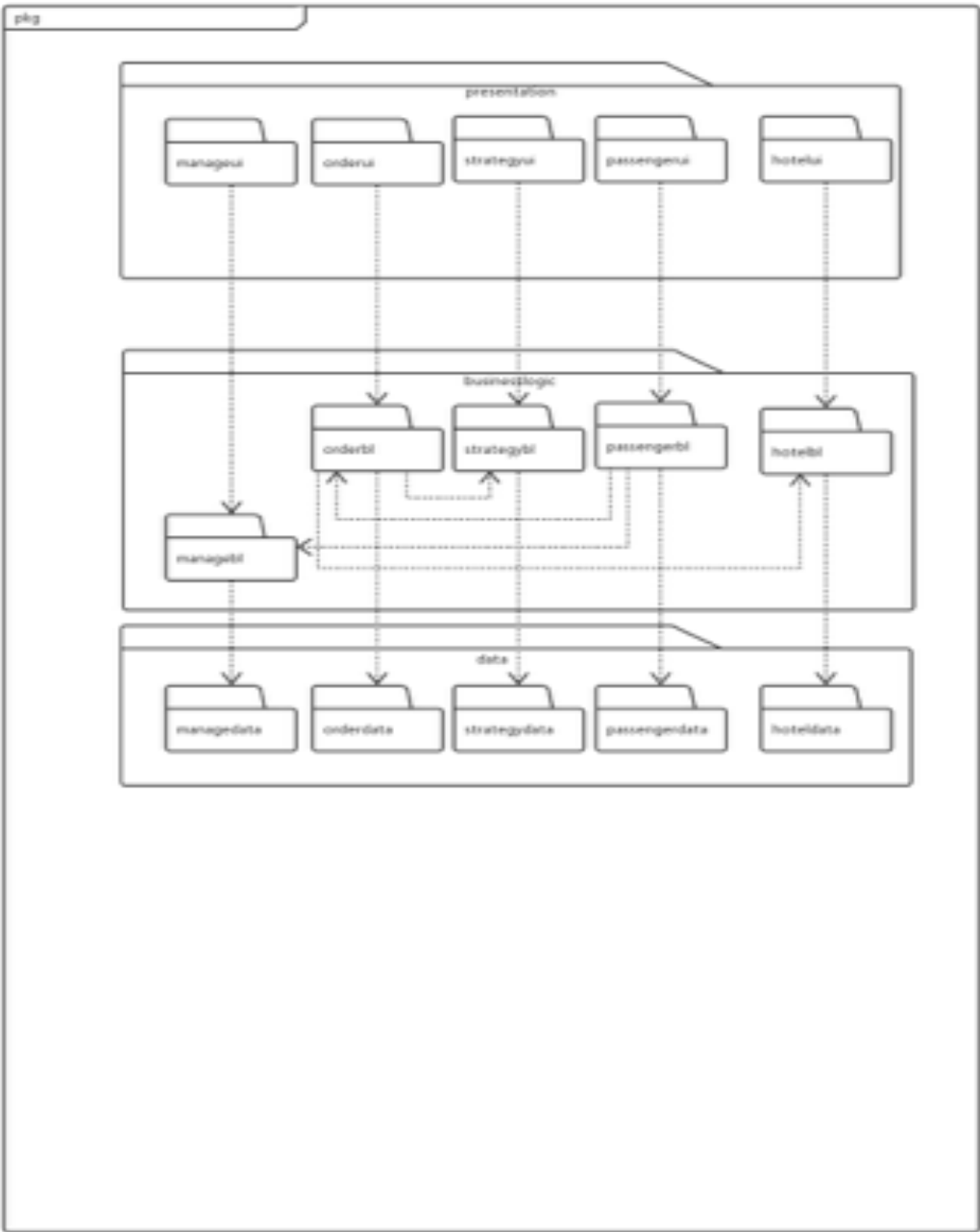


图 2 软件体系结构逻辑设计方案

## 4.组合视角

### 4.1.开发包图

表1 互联网酒店预订系统的最终开发包设计

| 开发（物理包）             | 依赖的其他开发包   |
|---------------------|--|
| mainUI              | loginui<br>manageui<br>orderui<br>strategyui<br>clientui<br>updateui                                   |
| loginui             | 界面类库包<br>loginservice<br>vo  |
| manageui            | 界面类库包<br>manageservice<br>vo   |
| orderui             | 界面类库包<br>orderservice<br>vo  |
| strategyui          | 界面类库包<br>strategy<br>service<br>vo   |
| clientui            | 界面类库包<br>clientservice<br>vo   |
| updateui            | 界面类库包<br>updateservice<br>vo   |
| loginservice        | java RMI<br>vo   |
| manageservice       | java RMI<br>vo   |
| orderservice        | java RMI<br>vo   |
| strategy<br>service | java RMI<br>vo   |
| clientservice       | java RMI<br>vo   |
| updateservice       | java RMI<br>vo   |
| vo                  |  |
| java RMI            | loginservice<br>manageservice<br>orderservice<br>strategy<br>service<br>clientservice<br>updateservice |

|                |  |
|----------------|--|
| loginbl        | loginservice<br>logindao<br>datafactorydao<br>vo             |
| managebl       | manageservice<br>managedao<br>datafactorydao<br>vo           |
| orderbl        | orderservice<br>orderdao<br>datafactorydao<br>creditbl<br>vo |
| strategybl     | strategieservice<br>strategydao<br>datafactorydao<br>vo      |
| clientbl       | clientservice<br>clientdao<br>datafactorydao<br>vo           |
| updatebl       | updateservice<br>updatedao<br>datafactorydao<br>vo           |
| creditbl       | datafactorydao<br><br>vo                                     |
| logindao       | po   |
| managedao      | po   |
| orderdao       | po   |
| strategydao    | po   |
| clientdao      | po   |
| updatedao      | po   |
| creditdao      | po   |
| datafactorydao | po   |
| logindata      | logindao<br>JDBCUtil<br>po                                   |
| managedata     | managedao<br>JDBCUtil<br>po                                  |



|                 |                               |
|-----------------|-------------------------------|
| orderdata       | orderdao<br>JDBCUtil<br>po    |
| strategydata    | strategydao<br>JDBCUtil<br>po |
| clientdata      | clientdao<br>JDBCUtil<br>po   |
| updatedata      | updatedao<br>JDBCUtil<br>po   |
| datafactorydata | datafactorydap<br>po          |
| JDBCUtil        |                               |
| po              |                               |
| 界面类库包           |                               |

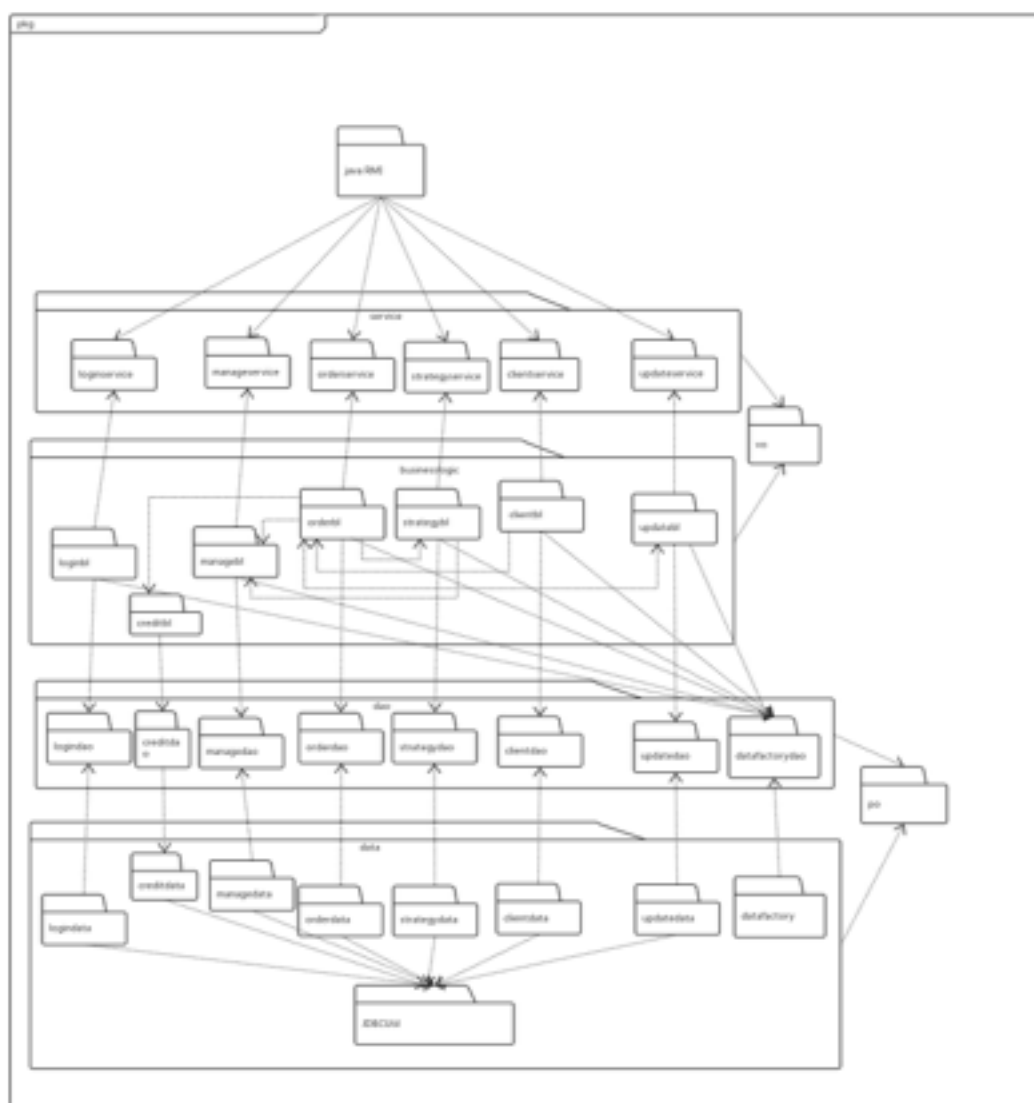


图3服务器端开发包图

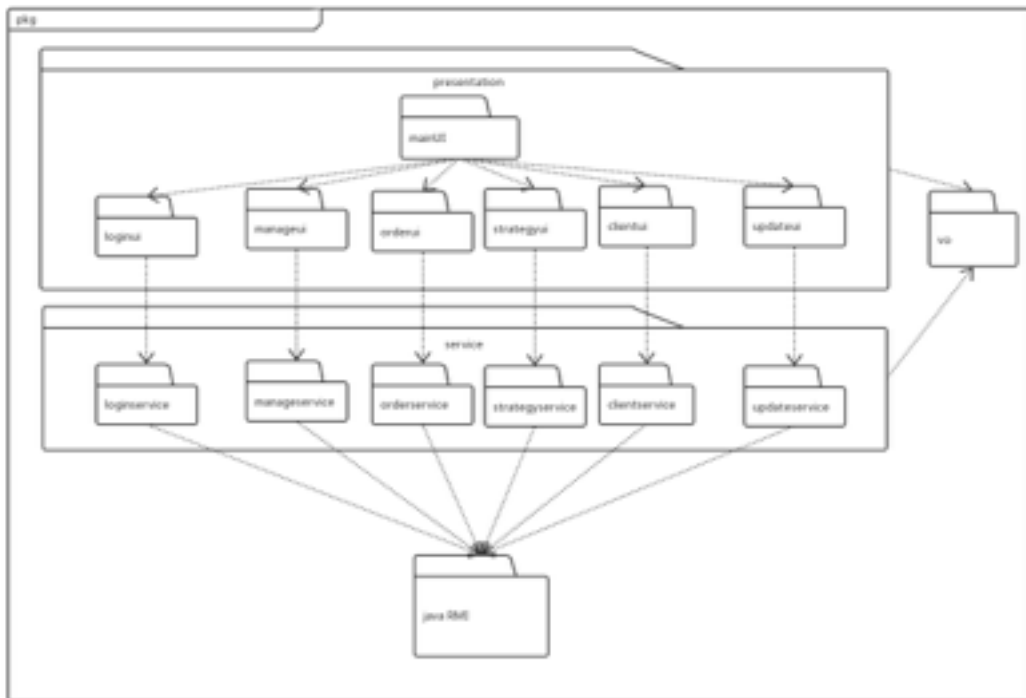


图4客户端开发包图

## 4.2运行时进程

在互联网酒店预订系统中，会有多个客户端进程和一个服务器端进程，其进程图如图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。

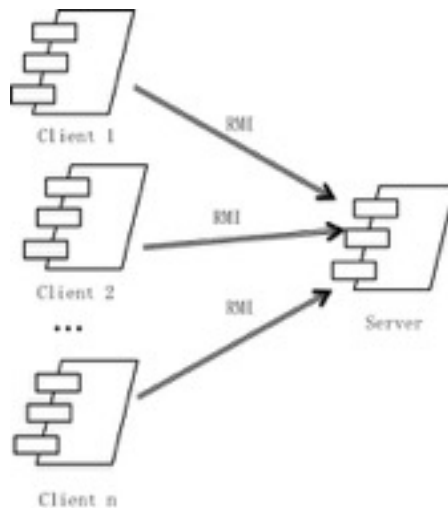


图5进程图

4.3物理部署

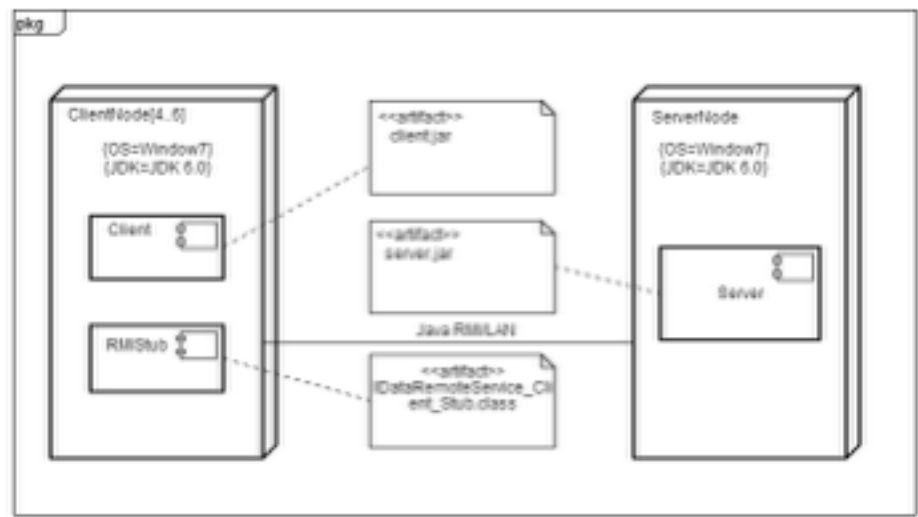


图6部署图

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。

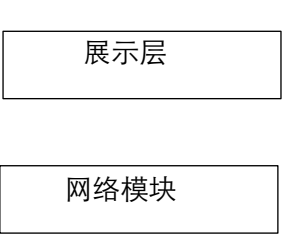


图7 客户端模块视图

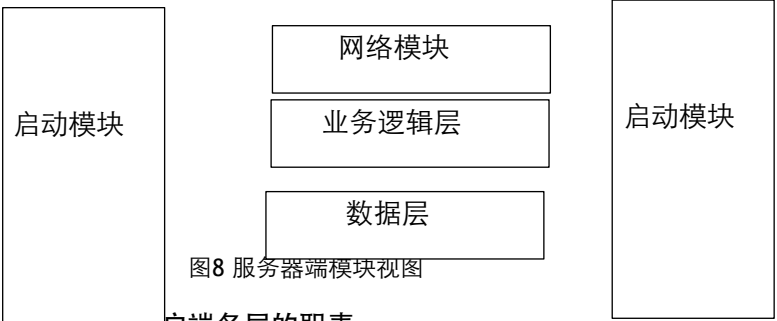


图8 服务器端模块视图

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 |
| 展示层  | 基于窗口的酒店管理系统客户端用户界面 |

|         |                     |
|---------|---------------------|
| 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          |
| 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      |
| 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    |
|---|----------|----------|
| loginService<br>manageService<br>orderService<br>strategyService<br>clientService<br>updateService<br>creditService | 客户端展示层   | 客户端业务逻辑层 |
| loginDao<br>manageDao<br>orderDao<br>strategyDao<br>clientDao<br>updateDao<br>creditDao                             | 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 42个用户界面：

登录界面

客户：

客户主界面，个人基本信息界面，注册会员界面，信用记录界面，客户订单列表界面，查找酒店界面，酒店列表界面，酒店详情界面，预定信息表单界面，订单信息界面，评价酒店界面

酒店：

酒店主界面，酒店基本信息界面，线下订单填写界面，房源信息界面，订单详情界面，酒店已有促销策略界面，添加促销策略编辑界面，酒店订单列表界面，异常订单处理界面，查找订单界面，处理执行订单界面

网站营销人员：

网站营销人员主界面，查找异常订单界面，处理异常订单界面，网站促销策略列表界面，添加促销策略编辑界面，信用充值界面，未执行订单列表界面，修改会员制度界面，会员等级折扣界面

网站管理人员：

网站管理人员主界面，维护酒店界面，酒店用户名输入界面，酒店信息修改界面，添加酒店界面，用户管理界面，客户用户名输入界面，添加网站营销人员界面，客户信息详情界

面，客户信息修改界面。  
界面跳转如图9所示。



图9界面跳转

5.2.1.用户界面层模块的职责

| 模块     | 职责                     |
|--------|------------------------|
| MainUI | 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

loginui模块需要的服务接口

| 服务名                     | 服务   |
|-------------------------|------|
| LoginblService.login    | 用户登陆 |
| LoginblService.register | 用户注册 |

clienui模块需要的服务接口

| 服务名                           | 服务             |
|-------------------------------|----------------|
| ClientblService.submitComment | 对已执行订单中的酒店提交评论 |
| ClientService.getHotelList    | 显示酒店列比阿婆       |
| ClientService.getHotelInfo    | 显示酒店详情         |
| ClientService.checkCredit     | 系统检查信用值        |

### Manageui模块需要的服务接口

| 服务名                                  | 服务                   |
|--------------------------------------|----------------------|
| manageSeverice.recharge              | 根据输入的充值数据对用户信用值进行充值  |
| ManageSeverice.addMarketStaff        | 添加市场营销人员记录           |
| ManageSeverice.addHotel              | 添加酒店记录               |
| ManageSeverice.getHotelInfo          | 根据输入的酒店用户名获得酒店的信息    |
| ManageSeverice.getUserInfo           | 根据输入的用户名和类型获得相应的用户信息 |
| ManageSeverice.changeHotelInfo       | 根据输入的酒店用户名和类型改变酒店记录  |
| ManageSeverice.changeUserInfo        | 根据输入的用户名改变用户记录       |
| ManageSeverice.getMemberType         | 根据输入的用户名获得会员的类型      |
| ManageSeverice.registerCommonMember  | 添加普通会员记录             |
| ManageSeverice.registerCompanyMember | 添加企业会员记录             |

### ordeui模块需要的服务接口

| 服务名                                | 服务                            |
|------------------------------------|-------------------------------|
| orderLogicService.getOrderInfo     | 根据订单编号得到相应类型的订单信息             |
| orderLogicService.changeOrderState | 更改订单的状态                       |
| orderLogicService.changeOrder      | 更改订单的状态以及异常执行情况               |
| orderLogicService.getOrderList     | 根据输入的订单编号或者选择的订单类型得到相应类型的订单列表 |
| orderLogicService.checkSixHour     | 检查客户撤销订单时间距离最晚订单时间是否不足6小时     |
| orderLogicService.withdrawOrder    | 客户订单撤销                        |

|                              |        |
|------------------------------|--------|
| orderLogicService.addAnOrder | 线下订单登记 |
|------------------------------|--------|

strategyui模块需要的服务接口

| 服务名                            | 服务        |
|--------------------------------|-----------|
| StrategyService.getStrategy    | 找到所有的促销策略 |
| StrategyService.addNewStrategy | 添加新的促销策略  |
| StrategyService.removeStrategy | 删除选定的促销策略 |

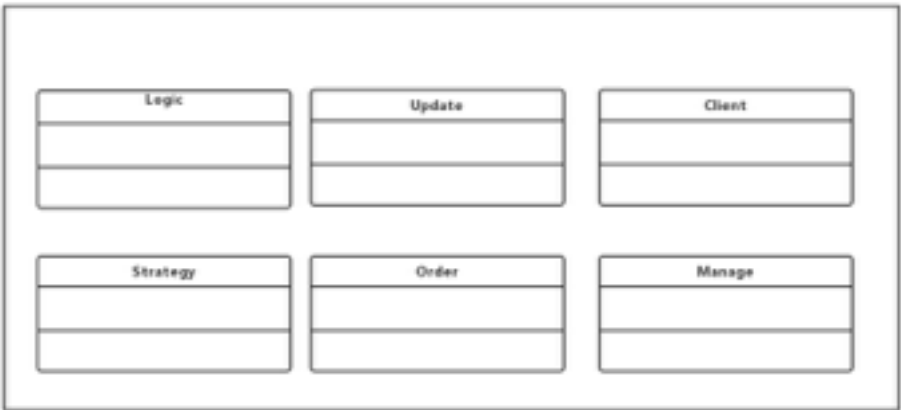
updateui模块需要的服务接口

| 服务名                           | 服务        |
|-------------------------------|-----------|
| UpdateService.getRoomInfo     | 查找相应的房源信息 |
| UpdateService.updateRoomInfo  | 更新房源信息    |
| UpdateService.addOfflineOrder | 生成线下订单    |

5.2.3. 用户界面模块设计原理

用户界面用java的javafx库来实现。

5.3. 业务逻辑层的分解





5.3.1.业务逻辑层模块的职责

| 模块         | 职责                      |
|------------|-------------------------|
| logicbl    | 负责实现商品分类与商品分类管理界面所需要的服务 |
| managebl   | 负责实现网站和酒店工作人员管理界面所需要的服务 |
| orderbl    | 负责实现处理订单界面所需要的服务        |
| strategybl | 负责实现制定促销策略界面所需要的服务      |
| clientbl   | 负责实现客户管理界面所需要的服务        |
| updatebl   | 负责实现系统更新所需要的服务          |

5.3.2业务逻辑模块的接口规范

Loginbl模块的接口规范

| 提供的服务（供接口）                                     |                      |  |
|--|----------------------|--|
| Loginbl.login                                  | 语法                   | public loginMessage login(RegisterVO loginVO)          |
|  | 前置条件                 | PO数据存在   |
|  | 后置条件                 | 返回ID   |
| Loginbl.register                               | 语法                   | public registerMessage register(RegisterVO registerVO) |
|  | 前置条件                 | 无  |
|  | 后置条件                 | 返回注册结果   |
| 需要的服务（需接口）                                     |                      |  |
| 服务名  | 服务                   |  |
| LoginblDao.login (String name,String password) | 根据ID来查找用户，返回查找到的PO对象 |  |
| LoginblDao.register(Client PO client)          | 增加一条PO数据             |  |

clientbl模块的接口规范

| 提供的服务（供接口） |
|------------|
|------------|

|  |                  |   |
|--|------------------|---|
| Clientbl.submitComment   | 语法               | public Information submit(String name,String HotelName,String comment,int star)                     |
|  | 前置条件             | 客户已登陆且评论格式合法  |
|  | 后置条件             | 增加一条VO记录  |
| Clientbl.getHotelList  | 语法               | public HotelNameVO[] getHotelList(String address,String area,String price,String star,String grade) |
|  | 前置条件             | 无   |
|  | 后置条件             | 返回酒店列表  |
| Clientbl.getHotellInfo   | 语法               | public HoteldetailVO getHotellInfo (String hotelName)   |
|  | 前置条件             | 无   |
|  | 后置条件             | 返回酒店详情  |
| Clientbl.checkCredit   | 语法               | public boolean checkCredit(String user)   |
|  | 前置条件             | 用户已登陆   |
|  | 后置条件             | 判断信用值   |
| 需要的服务（需接口）   |                  |   |
| 服务名  | 服务               |   |
| ClientDao.getHotelList(String address,String area,String price,String star,String grade) | 根据搜索条件返回查找到的PO对象 |   |
| Managebl.getHotellInfo(String client,Type type)  | 返回查找到的PO对象       |   |
| CreditDao.showCreditInformation(String clientName)                                       | 返回查找到的PO对象       |   |
| Orderbl.addAnOrder(OrderInfoVO orderInfoVO)  | 增加一个订单           |   |

managebl模块的接口规范

| 提供的服务（供接口）                |      |   |
|---------------------------|------|---|
| Managebl.recharge         | 语法   | Public boolean recharge(String client,int number)           |
|                           | 前置条件 | 输入的用户名存在，数值格式正确   |
|                           | 后置条件 | 根据输入的信息，增加该用户对应的信用值，  |
| Managebl. addMarketStaff  | 语法   | Public Message addMarketStaff (MarketStaffVO marketstaffVO) |
|                           | 前置条件 | 新增网站营销人员信息输入合法  |
|                           | 后置条件 | 根据输入的信息，增加网站营销人员，   |
| Managebl. addHotel        | 语法   | Public Message addHotel(HotelPO hotel)                      |
|                           | 前置条件 | 新增酒店信息输入合法  |
|                           | 后置条件 | 根据输入的信息，添加酒店，   |
| Managebl. getHotelInfo    | 语法   | public HotelVO getHotelInfo(GetVO get)                      |
|                           | 前置条件 | VO内用户名存在，type格式正确。  |
|                           | 后置条件 | 根据输入的用户名和type返回酒店信息   |
| Managebl. getUserInfo     | 语法   | Public UserVO getUserInfo(GetVO get)                        |
|                           | 前置条件 | 输入的用户名存在type格式正确  |
|                           | 后置条件 | 根据输入的用户名和type返回用户信息   |
| Managebl. changeHotelInfo | 语法   | public Message changeHotelInfo (ChangeHotelVO hotel)        |
|                           | 前置条件 | 输入的酒店用户名存在，type格式正确，酒店信息合法                                  |
|                           | 后置条件 | 修改该酒店信息   |
| Managebl. changeUserInfo  | 语法   | public Message changeUserInfo (UserVO user)                 |
|                           | 前置条件 | 输入的用户名存在，用户信息合法   |
|                           | 后置条件 | 修改用户信息  |
| Managebl.getMemberType    | 语法   | public MemberTypeVO getMemberType (String clientName)       |

|   |                    |  |
|---|--------------------|--|
|   | 前置条件               | 输入的用户名存在   |
|   | 后置条件               | 找到会员类型并返回  |
| Managebl.<br>registerCommonMember                   | 语法                 | public Message<br>registerCommonMember(CommonMemberVO commonMember)    |
|   | 前置条件               | 输入的普通会员信息合法  |
|   | 后置条件               | 修改用户的会员信息  |
| Managebl.<br>registerCompanyMember                  | 语法                 | public Message<br>registerCompanyMember(CompanyMemberVO companyMember) |
|   | 前置条件               | 输入的企业会员信息合法  |
|   | 后置条件               | 修改用户的会员信息  |
| Managebl. isPartner                                 | 语法                 | public bool isPartner(String<br>hotelID,String companyname)            |
|   | 前置条件               | 输入的信息合法  |
|   | 后置条件               | 判断是否是合作企业  |
| Managebl. getPrices                                 | 语法                 | public int[] getPrices(String<br>hotelID)                              |
|   | 前置条件               | 输入的信息合法  |
|   | 后置条件               | 返回酒店房间的价钱  |
| 需要的服务（需接口）  |                    |  |
| 服务名   | 服务                 |  |
| ManageDao.<br>addMarketStaff(UserPO<br>marketStaff) | 添加单一持久化对象          |  |
| ManageDao.<br>addHotel(HotelPO hotel)               | 添加单一持久化对象          |  |
| ManageDao.<br>getHotelInfo(String hotel)            | 根据酒店用户名进行查找单一持久化对象 |  |
| ManageDao.<br>getUserInfo(String client)            | 根据用户名进行查找单一持久化对象   |  |
| ManageDao.<br>changeHotelInfo(HotelPO<br>hotel)     | 更新单一持久化对象          |  |
| ManageDao.<br>changeUserInfo(ClientPO<br>clientPO)  | 更新单一持久化对象          |  |
| ManageDao.getMemberType()                           | 根据用户名查找单一持久化对象     |  |

orderbl模块的接口规范

| 提供的服务（供接口）               |      |   |
|--------------------------|------|---|
| Orderbl.getOrderInfo     | 语法   | public OrderInfoVO getOrderInfo (String orderID)  |
|                          | 前置条件 | 用户选择查看订单详细信息  |
|                          | 后置条件 | 根据用户选择的订单类型和用户名，分类出相应的订单信息  |
| Orderbl.changeOrderState | 语法   | public Message changeOrderState (String orderID, OrderType orderType)                       |
|                          | 前置条件 | 用户根据订单编号查找到要修改的订单更改订单类型   |
|                          | 后置条件 | 更改订单状态  |
| Orderbl.changeOrder      | 语法   | public Message changeOrderState (exceptionVO exceptionVO)                                   |
|                          | 前置条件 | 酒店工作人员查找到异常订单，补登记执行情况（即延迟入住）并将此订单类型更改为已执行   |
|                          | 后置条件 | 订单类型更新，客户信用值信息更新  |
| Orderbl.getOrderList     | 语法   | public OrderInfoVO[] getOrderList (String username, UserType userType, OrderType orderType) |
|                          | 前置条件 | 用户选择查看订单  |
|                          | 后置条件 | 根据用户选择的订单类型以及用户类型，将订单列表按照时间倒叙排列   |
| Orderbl.checkSixHour     | 语法   | public boolean checkSixHour (OrderInfoVO orderinfo)   |
|                          | 前置条件 | 客户撤销订单  |
|                          | 后置条件 | 根据将要撤销的订单信息判断此时距离最晚订单执行时间是否满足6小时  |
| Orderbl.addAnOrder       | 语法   | public Message addAnOrder (OrderInfoVO orderInfoVO)   |
|                          | 前置条件 | 传入的参数合法   |
|                          | 后置条件 | 线下订单登记完成  |
| 需要的服务（需接口）               |      |   |
| 服务名                      | 服务   |   |

|  |                  |
|--|------------------|
| OrderDao.getOrderInfo (String orderID)   | 根据订单编号查找持久化对象    |
| OrderDao.updateOrder(OrderPO orderPO)  | 更新订单信息           |
| OrderDao.getOrderList(String username, UserType userType, OrderType orderType) | 根据用户名和订单类型获得订单信息 |
| Strategybl.getBestStrategy (Client client, Type type, String hotelID)          | 比较出最好的优惠         |
| Creditbl.addnewCreditInformation(CreditInformationPO creditInformationPO)      | 增加新的信用记录         |

### strategybl模块的接口规范

| 提供的服务（供接口）  |                |   |
|---|----------------|---|
| Strategybl.getStrategy                              | 语法             | public StrategyVO[] getStrategy(Type type ,String hotelname)          |
|   | 前置条件           | 酒店名称或者网站营销人员类型输入合法  |
|   | 后置条件           | 找到所有已经制定过的促销策略  |
| Strategybl.addNewStrategy                           | 语法             | public Message addNewStrategy(StrategyVO strategyVO)                  |
|   | 前置条件           | 输入的促销策略合法   |
|   | 后置条件           | 持久化更新促销策略   |
| Strategybl.removeStrategy                           | 语法             | public boolean removeStrategy(Type type,StrategyVO strategy)          |
|   | 前置条件           | 输入的促销策略合法   |
|   | 后置条件           | 删除该促销策略，持久化更新涉及的对象的数据   |
| Strategybl.getBestStrategy                          | 语法             | public double getBestStrategy(Client client,Type type,String hotelID) |
|   | 前置条件           | 输入的参数合法   |
|   | 后置条件           | 计算出最好的优惠策略  |
| 需要的服务（需接口）  |                |   |
| 服务名   | 服务             |   |
| StrategyDao.getStrategy(Type type,String hotelname) | 根据名称来寻找相应的促销策略 |   |

|  |           |
|--|-----------|
| StrategyDao.addNewStrategy(StrategyVO strategyVO)            | 添加单一化持久对象 |
| StrategyDao.removeStrategy(Type type, StrategyPO strategyPO) | 删除单一化持久对象 |

### updatebl模块的接口规范

| 提供的服务（供接口）  |                 |  |
|---|-----------------|--|
| Updatebl.getRoomInfo                                | 语法              | public RoomVO getRoomInfo(String hotelmanagerid)         |
|   | 前置条件            | 酒店工作人员用户名正确  |
|   | 后置条件            | 查找到相应的房源信息   |
| Updatebl.updateRoomInfo                             | 语法              | public Message updateRoomInfo(UpdateRoomVO updateRoomVO) |
|   | 前置条件            | 无  |
|   | 后置条件            | 更改房间信息，持久化更新涉及到的数据                                       |
| Updatebl.addOfflineOrder                            | 语法              | public Message addOfflineOrder(OrderVO order)            |
|   | 前置条件            | 订单信息输入合法   |
|   | 后置条件            | 生成线下订单，持久化更新订单信息   |
| 需要的服务（需接口）  |                 |  |
| 服务名   | 服务              |  |
| UpdateDao.getRoomInfo(String hotelmanagerid)        | 根据ID进行查找单一持久化对象 |  |
| UpdateDao.updateRoomInfo(UpdateRoomPO updateRoomPO) | 修改持久化对象         |  |

### creditbl模块需要的服务接口

| 提供的服务（供接口）                    |      |  |
|-------------------------------|------|--|
| Creditbl.addNewCreditInfo     | 语法   | Public boolean addnewCreditInformation (CreditRecord creditRecord)   |
|                               | 前置条件 | 传入的参数合法  |
|                               | 后置条件 | 增加了一条信用记录  |
| Creditbl.getCreditInformation | 语法   | public CreditInformationPO[] getCreditInformation(String clientName) |

|                                |           |            |
|--------------------------------|-----------|------------|
|                                | 前置条件      | PO数据存在     |
|                                | 后置条件      | 返回查找到的PO对象 |
| 需要的服务（需接口）                     |           |            |
| 服务名                            | 服务        |            |
| CreditDao.addNewCreditInfo     | 增加了一条信用记录 |            |
| CreditDao.getCreditInformation | 返回信用记录信息  |            |

## 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。持久化数据保存采用序列化文件

### 5.4.1.数据层模块的职责

| 模块           | 职责                           |
|--------------|------------------------------|
| logindata    | 提供用户登录服务                     |
| managedata   | 提供用户和酒店信息的增删改查服务             |
| orderdata    | 提供订单的查看、撤销，异常订单的处理，正常订单的执行服务 |
| strategydata | 提供促销策略的增删服务                  |
| clientdata   | 提供酒店的查看、预定、评价服务              |
| updatedata   | 提供信息更新服务                     |

### 5.4.2.数据层模块的接口规范

#### Logindata模块的接口规范

|                 |      |   |
|-----------------|------|---|
| 提供的服务（供接口）      |      |   |
| Logindata.login | 语法   | public boolean login(String name,String password) |
|                 | 前置条件 | PO数据存在  |
|                 | 后置条件 | 返回查找到的PO对象  |



|                    |      |  |
|--------------------|------|--|
| Logindata.register | 语法   | public registerMessage register(ClientPO client) |
|                    | 前置条件 | PO数据不存在  |
|                    | 后置条件 | 增加一条PO数据   |

## clientdata模块的接口规范

| 提供的服务（供接口）              |      |   |
|-------------------------|------|---|
| Clientdata.addComment   | 语法   | public boolean addComment(CommentPO commentPO)  |
|                         | 前置条件 | PO不存在   |
|                         | 后置条件 | 新增一条PO记录  |
| Clientdata.getHotelList | 语法   | public HotelPO getHotelList(String address,String area,String price,String star,String grade) |
|                         | 前置条件 | 在数据库中存在同样ID的PO  |
|                         | 后置条件 | 显示酒店列表  |
| Clientdata.addAnOrder   | 语法   | public boolean addAnOrder(OrderPO orderPO)  |
|                         | 前置条件 | PO不存在   |
|                         | 后置条件 | 增加一条PO记录  |

## Creditdata模块的接口规范

| 提供的服务（供接口）                         |      |   |
|------------------------------------|------|---|
| Creditdata.showCreditInformation   | 语法   | public CreditInformationPO[] showCreditInformation(String clientName)                 |
|                                    | 前置条件 | PO数据存在  |
|                                    | 后置条件 | 返回查找到的PO对象  |
| Creditdata.addnewCreditInformation | 语法   | public Information addNewCreditInformation (CreditInformation PO creditInformationPO) |
|                                    | 前置条件 | PO数据不存在   |

|  |      |          |
|--|------|----------|
|  | 后置条件 | 增加一条PO数据 |
|--|------|----------|

### managedata模块的接口规范

| 提供的服务（供接口）                 |      |   |
|----------------------------|------|---|
| Managedata.addMarketStaff  | 语法   | public boolean addMarketStaff(UserPO marketStaff) |
|                            | 前置条件 | 同样的po在数据中不存在                                      |
|                            | 后置条件 | 新增一个po  |
| Managedata.addHotel        | 语法   | public boolean addHotel(HotelPO hotel)            |
|                            | 前置条件 | 同样的po在数据中不存在                                      |
|                            | 后置条件 | 新增一个po  |
| Managedata.getHotelInfo    | 语法   | public HotelPO getHotelInfo(String hotel)         |
|                            | 前置条件 | 无   |
|                            | 后置条件 | 按酒店用户名进行查找返回相应的HotelPO的结果                         |
| Managedata.getUserInfo     | 语法   | public ClientPO getUserInfo(String client)        |
|                            | 前置条件 | 无   |
|                            | 后置条件 | 按用户名和type进行查找返回相应的ClientPO的结果                     |
| Managedata.changeHotelInfo | 语法   | public boolean changeHotelInfo(HotelPO hotel)     |
|                            | 前置条件 | 同样的po在数据中存在                                       |
|                            | 后置条件 | 更新相应的HotelPO                                      |
| Managedata.changeUserInfo  | 语法   | public boolean changeUserInfo(ClientPO clientPO)  |
|                            | 前置条件 | 同样的po在数据中存在                                       |
|                            | 后置条件 | 更新相应的ClientPO                                     |
| Managedata.getMemberType   | 语法   | public MemberInfoPO getMemberType()               |
|                            | 前置条件 | 无   |
|                            | 后置条件 | 查找相应的po并返回  |

strategydata模块的接口规范

| 提供的服务（供接口）                  |      |  |
|-----------------------------|------|--|
| Strategydata.getStrategy    | 语法   | public StrategyPO[] getStrategy(Type type ,String hotelname) |
|                             | 前置条件 | 输入的酒店名称存在  |
|                             | 后置条件 | 返回查找到的PO对象   |
| Strategydata.addNewStrategy | 语法   | public boolean addNewStrategy(StrategyPO strategyPO)         |
|                             | 前置条件 | 添加的促销策略在数据库中不存在  |
|                             | 后置条件 | 增加一条PO记录   |
| Strategydata.removeStrategy | 语法   | public boolean removeStrategy(Type type,StrategyPO strategy) |
|                             | 前置条件 | 输入的促销策略在数据库中已存在  |
|                             | 后置条件 | 删除一条PO记录   |

updatedata模块的接口规范

| 提供的服务（供接口）                |      |  |
|---------------------------|------|--|
| Updatedata.getRoomInfo    | 语法   | public RoomVO getRoomInfo(String hotelmanagerID)         |
|                           | 前置条件 | 无  |
|                           | 后置条件 | 返回查找到的PO记录   |
| Updatedata.updateRoomInfo | 语法   | public boolean updateRoomInfo(UpdateRoomPO updateRoomPO) |
|                           | 前置条件 | po记录存在   |
|                           | 后置条件 | 修改一条PO记录   |

orderdata模块的接口规范

| 提供的服务（供接口）         |      |   |
|--------------------|------|---|
| order.getOrderInfo | 语法   | public OrderPO getOrderInfo(String orderID) |
|                    | 前置条件 | 无   |
|                    | 后置条件 | 返回相应订单编号的订单信息                               |
|                    | 语法   | public boolean updateOrder(OrderPO orderPO) |

|                    |      |   |
|--------------------|------|---|
| order.updateOrder  | 前置条件 | PO记录存在  |
|                    | 后置条件 | 更新此PO包  |
| order.getOrderList | 语法   | public OrderPO[] getOrderList (String username, UserType userType, OrderType orderType) |
|                    | 前置条件 | 无   |
|                    | 后置条件 | 返回相应用户的所有订单   |
| order.addAnOrder   | 语法   | public boolean addAnOrder (OrderPO orderPO)   |
|                    | 前置条件 | 无此PO记录  |
|                    | 后置条件 | 增加此po记录   |

## 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，在此只做简单的介绍：

UserPO类是用户的属性，如账号，信用值，联系方式等属性

OrderPO类包含订单的属性，如订单编号，订单类型，订单时间，预订酒店，预订人，价格等属性

HotelPO类包含酒店的信息，如酒店名称，酒店的地址，酒店的星级，酒店的简介等属性

CommentPO类包含评论的信息，如评论的时间，评论的人，是否匿名，评论内容等属性

StrategyPO类包含促销策略的信息，如促销策略的类型，促销策略的折扣等属性

CreditPO类包含信用记录的信息，包括信用记录变更时间，信用记录变更原因，信用记录变更额度。

CompanyPO类包含合作企业的信息，包括合作企业的名称

RoomPO类包含酒店房间的信息，包含酒店房间的数目，酒店房间的价格

## 6.2数据库表

### 客户信息表clientInfo

| 字段             | 类型      | 长度  | 主键 | 说明                   |
|----------------|---------|-----|----|----------------------|
| name           | varchar | 10  | 是  | 用户名，最大长度为10          |
| type           | char    |     |    | 客户类型：普通用户、普通会员、企业会员  |
| contact        | char    | 11  |    | 联系方式，11位手机号码         |
| birthday       | data    |     |    | 普通会员生日，默认值为1970年1月1日 |
| rank           | int     |     |    | 会员等级                 |
| credit         | int     |     |    | 信用值数值                |
| enterpriseName | varchar | 255 |    | 企业会员的企业名称            |

### 信用记录表 creditRecord

| 字段       | 类型       | 长度 | 主键 | 说明              |
|----------|----------|----|----|-----------------|
| userName | varchar  | 10 |    | 客户用户名           |
| orderId  | char     | 18 | 是  | 订单编号            |
| time     | datetime |    |    | 信用值变化的时间        |
| action   | varchar  | 25 |    | 动作（执行、异常、撤销、充值） |
| change   | int      |    |    | 数值的变化           |

## 订单信息表 orderInfo

| 字段                   | 类型       | 长度  | 主键 | 说明       |
|----------------------|----------|-----|----|----------|
| orderID              | char     | 18  | 是  | 订单编号     |
| userName             | char     | 10  |    | 客户用户名    |
| orderState           | char     |     |    | 订单状态     |
| hotelName            | varchar  | 255 |    | 酒店名称     |
| roomType             | char     |     |    | 房间类型     |
| numberOfRoom         | int      |     |    | 数量       |
| expectedNumberOfUser | int      |     |    | 预计入住人数   |
| hasChild             | boolean  |     |    | 有无儿童     |
| expectedInTime       | datetime |     |    | 预计入住时间   |
| expectedOutTime      | datetime |     |    | 预计退房时间   |
| lastestTime          | datetime |     |    | 最迟订单执行时间 |
| actualInTime         | datetime |     |    | 实际入住时间   |
| actualOutTime        | datetime |     |    | 实际退房时间   |
| exception            | varchar  | 255 |    | 异常情况，可为空 |
| cancelTime           | datetime |     |    | 撤销时间，可为空 |

## 评价信息表 assessingInfo

| 字段        | 类型       | 长度  | 主键 | 说明           |
|-----------|----------|-----|----|--------------|
| orderID   | char     | 18  | 是  | 订单编号         |
| hotelName | varchar  | 255 |    | 酒店名称         |
| time      | datetime |     |    | 评价时间         |
| userName  | char     | 10  |    | 客户用户名        |
| content   | text     |     |    | 评价内容，不少于15个字 |

酒店信息表 hotelInfo

| 字段           | 类型      | 长度  | 主键 | 说明            |
|--------------|---------|-----|----|---------------|
| hotelName    | varchar | 255 |    | 酒店名称          |
| intro        | varchar | 255 |    | 介绍            |
| address      | varchar | 255 |    | 地址            |
| businessArea | varchar | 255 |    | 商圈            |
| service      | varchar | 255 |    | 服务设施          |
| star         | char    |     |    | 星级            |
| staffName    | varchar | 255 | 是  | 酒店工作人员<br>用户名 |
| point        | double  |     |    | 酒店的评价得分       |
| pointNumber  | int     |     |    | 打分人数          |

酒店跟合作企业匹配表 hotelMatchEnterprise

| 字段             | 类型      | 长度  | 主键 | 说明     |
|----------------|---------|-----|----|--------|
| hotelStaffName | varchar | 255 | 是  | 酒店名称   |
| co-enterprise  | varchar | 255 | 是  | 合作企业名称 |

酒店促销策略信息表 hotelStrategy

| 字段             | 类型      | 长度  | 主键 | 说明   |
|----------------|---------|-----|----|------|
| hotelStaffName | varchar | 255 |    | 酒店名称 |
| name           | varchar | 255 |    | 名称   |
| discountValue  | char    | 4   |    | 折扣数值 |

## 房间信息表 roomInfo

| 字段             | 类型      | 长度  | 主键 | 说明   |
|----------------|---------|-----|----|------|
| hotelStaffName | varchar | 255 | 是  | 酒店名称 |
| type           | char    |     | 是  | 房间类型 |
| number         | int     |     |    | 数量   |
| initialPrice   | char    |     |    | 初始价格 |

## 网站促销策略信息表 websiteStrategy

| 字段           | 类型      | 长度  | 主键 | 说明     |
|--------------|---------|-----|----|--------|
| name         | varchar | 255 | 是  | 促销策略名称 |
| period       | varchar | 255 |    | 特殊时期   |
| discount     | char    |     |    | 折扣数值   |
| businessArea | varchar | 255 |    | 商圈     |
| rank         | int     |     |    | 会员等级   |

## 用户信息表userInfo

| 字段       | 类型      | 长度 | 主键 | 说明                           |
|----------|---------|----|----|------------------------------|
| name     | varchar | 10 | 是  | 用户名                          |
| password | varchar | 14 |    | 密码                           |
| type     | char    |    |    | 用户类型：客户、网站管理人员、网站营销人员、酒店工作人员 |

## 会员制度信息表 memberInfo

| 字段   | 类型  | 长度 | 主键 | 说明   |
|------|-----|----|----|------|
| rank | int |    | 是  | 会员等级 |



|               |     |  |  |                |
|---------------|-----|--|--|----------------|
| <b>Credit</b> | int |  |  | 每个等级开始的<br>信用值 |
|---------------|-----|--|--|----------------|