网上酒店客房预订系统

体系结构设计文档

南京大学软件学院

xxx小组

成员：叶晓波、田贵松、辛志庭、陈步兵

更新历史:

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **修改日期** | **修改原因** | **版本号** |
| 全员 | 2016/10/16 | 初版 | v1.0 |
|  |  |  |  |

目录

[目录 3](#_Toc464424138)

[**1.** **引言** 5](#_Toc464424139)

[**1.1.** **编制目的** 5](#_Toc464424140)

[**1.2.** **参考资料** 6](#_Toc464424141)

[**2.** **产品概述** 6](#_Toc464424142)

[**3.** **逻辑视角** 6](#_Toc464424143)

[**4.** **组合视角** 8](#_Toc464424144)

[**4.1.** **开发包图** 9](#_Toc464424145)

[**4.2.** **运行时进程** 14](#_Toc464424146)

[**4.3.** **物理部署** 14](#_Toc464424147)

[**5.** **接口视角** 15](#_Toc464424148)

[**5.1.** **模块职责** 15](#_Toc464424149)

[**5.2.** **用户界面层的分解** 18](#_Toc464424150)

[**5.2.1.** **用户界面层模块的职责** 20](#_Toc464424151)

[**5.2.2.** **用户界面层模块的接口规范** 20](#_Toc464424152)

[**5.2.3.** **用户界面模块设计原理** 23](#_Toc464424153)

[**5.3.** **业务逻辑层** 23](#_Toc464424154)

[**5.3.1.** **业务逻辑层模块的职责** 24](#_Toc464424155)

[**5.3.2.** **业务逻辑层模块的接口规范** 24](#_Toc464424156)

[**5.4.** **数据层的分解** 36](#_Toc464424157)

[**5.4.1.** **数据层模块的职责** 36](#_Toc464424158)

[**5.4.2.** **数据层模块的接口规范** 37](#_Toc464424159)

[**6.** **信息视角** 43](#_Toc464424160)

[**6.1.** **数据持久化对象** 43](#_Toc464424161)

[**7.** **原型代码** 44](#_Toc464424162)

1. **引言**
   1. **编制目的**

本报表详细完成对网上酒店客房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、测试人员及最终用户而编写，是了解系统的导航。

* 1. **参考资料**

暂无

1. **产品概述**

参见用例文档和需求规格说明文档。用例图展示如下：

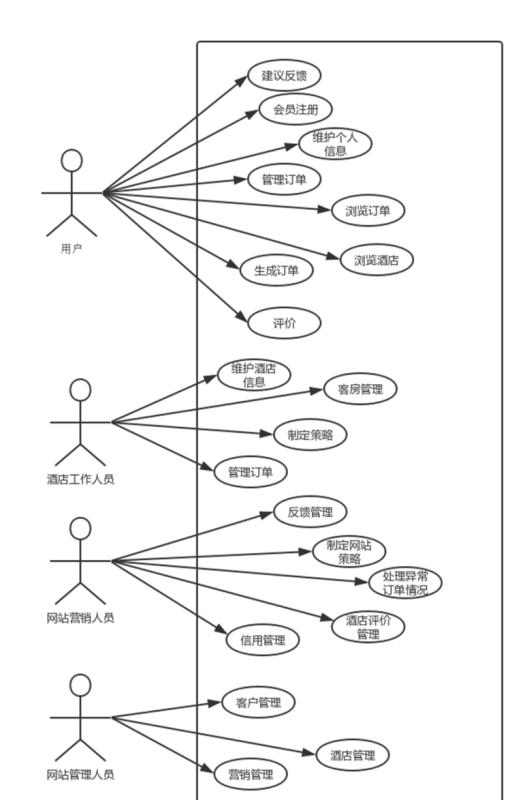


图2：预订系统用例图

1. **逻辑视角**

本系统采用分层体系结构，将系统分为三层（展示层presentation，业务逻辑层businesslogical和数据层data）。展示层包括GUI界面的实现，业务逻辑层包括业务实现的逻辑关系，数据层包括数据的持有，维护和数据库链接。分层体系结构的逻辑包图表达如下：

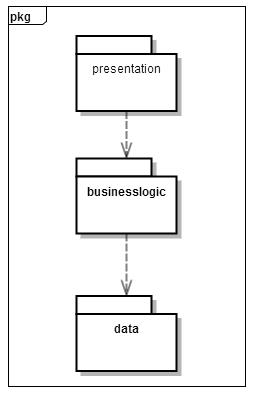


图3.1-1包图表达，逻辑视角

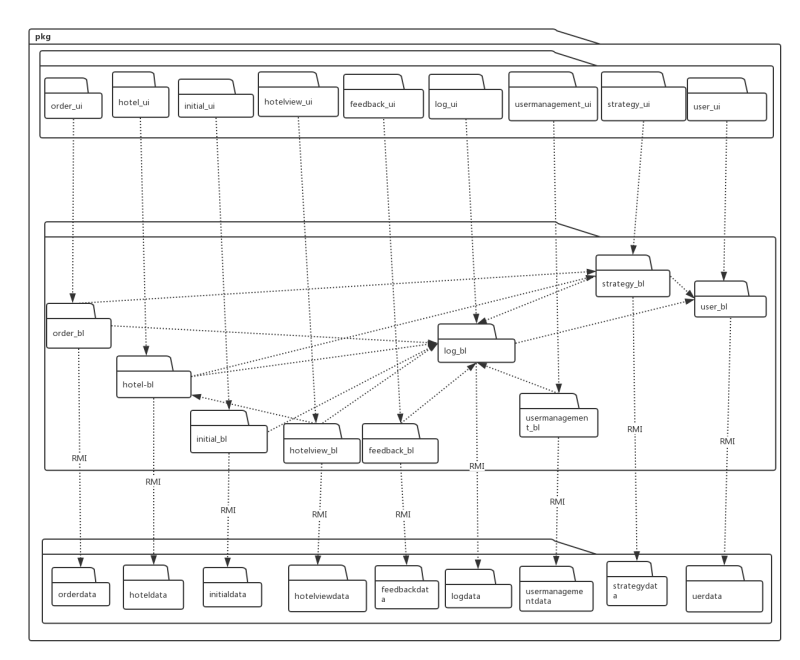


图3.1-2网上酒店客房预订：系统逻辑设计方案

1. **组合视角**

在设计的过程中，小组成员做出了如下的考虑和决策：

1.将presentation层和bl层置于客户端，将data层放在服务器，这样可以最大程度地减少RMI接口的数量。bl层不再直接依赖data层，而是通过RMI技术调用data层方法；

2.所有的data层包都要依赖databaseutility包，databaseutility包依赖JDBC类库包；

3.所有的presentation层开发包会依赖于图形界面类库包；

4.presentation层实现时，通过mainui包来实现整个页面的跳转和链接；

5.利用底层建立接口包的方法，使高层不再直接依赖低层；

6.在传递复杂数据时，对象声明独立为开发包VO和PO；

7.暂时不存在循环依赖；

8.考虑四种用户共用客户端，通过login识别进入不同页面，详细信息参见下文5.1；

经过讨论最终的网上酒店客房预订系统的开发包设计如下。

* 1. **开发包图**

表4.1-1 快递物流系统的最终开发包设计

|  |  |
| --- | --- |
| 开发（物理）包 | 依赖的其他开发包 |
| main\_ui | 依赖于所有的ui包 |
| userinfo\_ui | userinfo\_blservice |
| advicefeedback\_ui | advicefeedback\_blservice |
| usermanagement\_ui | usermanagement\_blservice |
| register\_ui | register\_blservice |
| login\_ui | login\_blservice |
| orderformation\_ui | orderformation\_blservice |
| ordermanagement\_ui | ordermanagement\_blservice |
| hotelview\_ui | hotelview\_blservice |
| hotelmanagement\_ui | gotelmanagement\_blservice |
| hotelstrategy\_ui | hotelstrategy\_blservcie |
| systemstrategy\_ui | systemstrategy\_blservice |
| userinfo\_blservice |  |
| advicefeedback\_blservice |  |
| usermanagement\_blservice |  |
| register\_blservice |  |
| login\_blservice |  |
| orderformation\_blservice |  |
| ordermanagement\_blservice |  |
| hotelview\_blservice |  |
| hotelmanagement\_blservice |  |
| hotelstrategy\_blservice |  |
| systemstrategy\_blsercice |  |
| userinfo\_bl | userinfo\_blservice, userinfo\_dataservice,  login\_bl |
| advicefeedback\_bl | advicefeedback\_blservice,  advicefeedback\_dataservice,  login\_bl |
| usermanagement\_bl | usermanagement\_blservice,  uerinfo\_dataservice,userinfo\_bl,  hotel\_bl |
| regesiter\_bl | regesiter\_blservice,  userinfo\_dataservice  login\_bl |
| login\_bl | login\_blservice,  userinfo\_dataservice |
| order\_bl | orderformation\_blservice,  ordermanagement\_blservice,  order\_dataservice,  userinfo\_bl,  systemstrategy\_bl,  hotelstrategy\_bl,  hotel\_bl |
| hotel\_bl | hotelview\_blservice,  hotelmanagement\_blsevice,  hotel\_dataservice,  login\_bl,  hotelstratregy\_bl |
| hotelstrategy\_bl | hotelstrategy\_blservice,  hotel\_dataservice |
| systemstrategy\_bl | systemstrategy\_blservice,  systemstrategy\_dataservice  userinfo\_bl |
| userinfo\_dataservice |  |
| advicefeedback\_dataservice |  |
| order\_dataservice |  |
| hotel\_dataservice |  |
| systemstrategy\_dataservice |  |
| userinfo\_data | userinfo\_dataservice |
| advicefeedback\_data | advicefeedback\_dataservice |
| order\_data | order\_dataservice |
| hotel\_data | hotel\_dataservice |
| systemstrategy\_data | systemstrategy\_dataservice |
| PO | 所有的ui和blservice均依赖于VO |
| VO | 所有的data和dataservice均依赖于PO |
| 界面类库包 | 所有的ui均依赖于界面类库包 |
| javaRMIjava.swing,java.awt,java.2D,java.3D | 服务器端和客户端之间传递依赖于RMI |

所对应的物理包设计图如下：

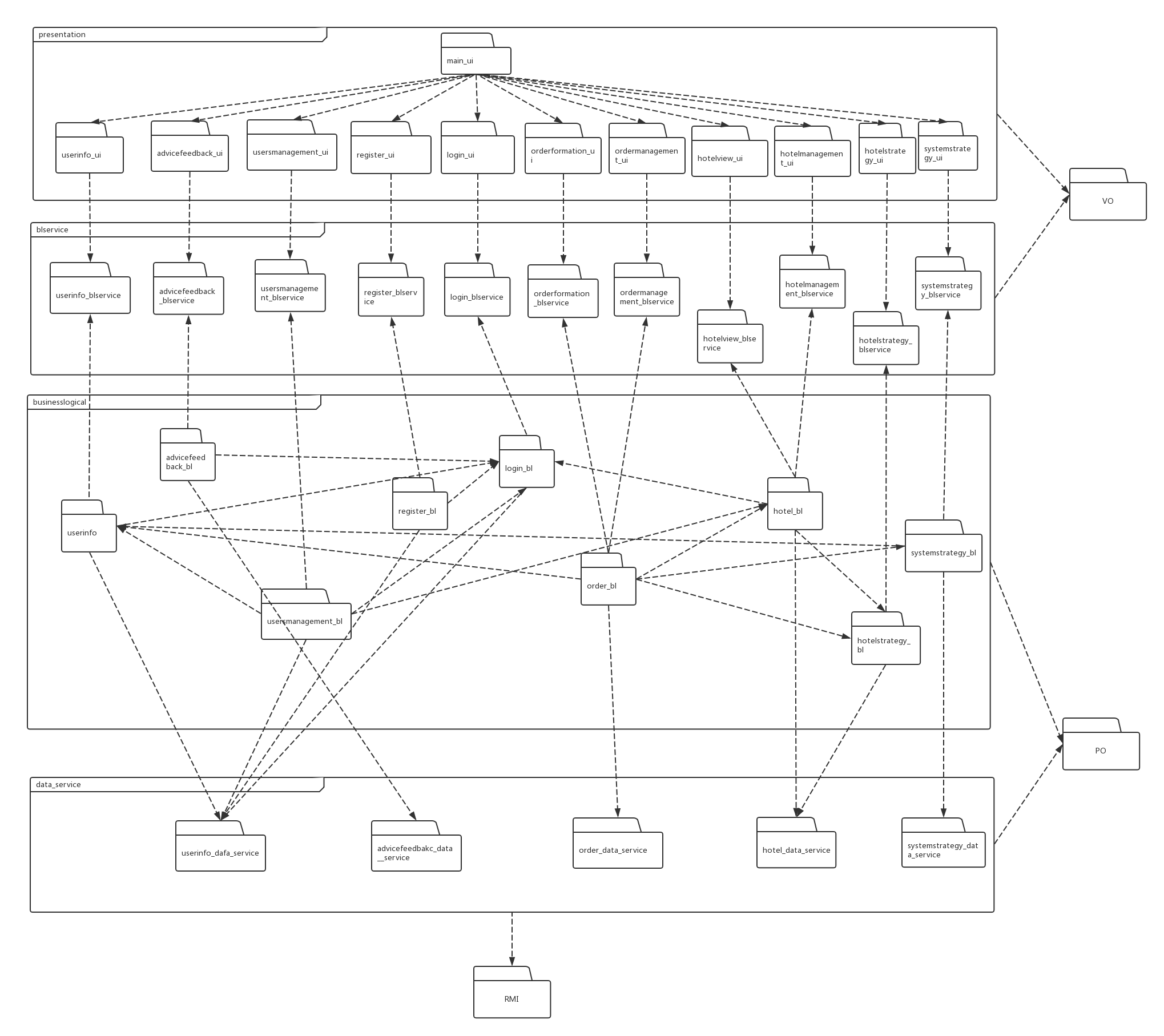


图4.1-1客户端物理包结构设计图

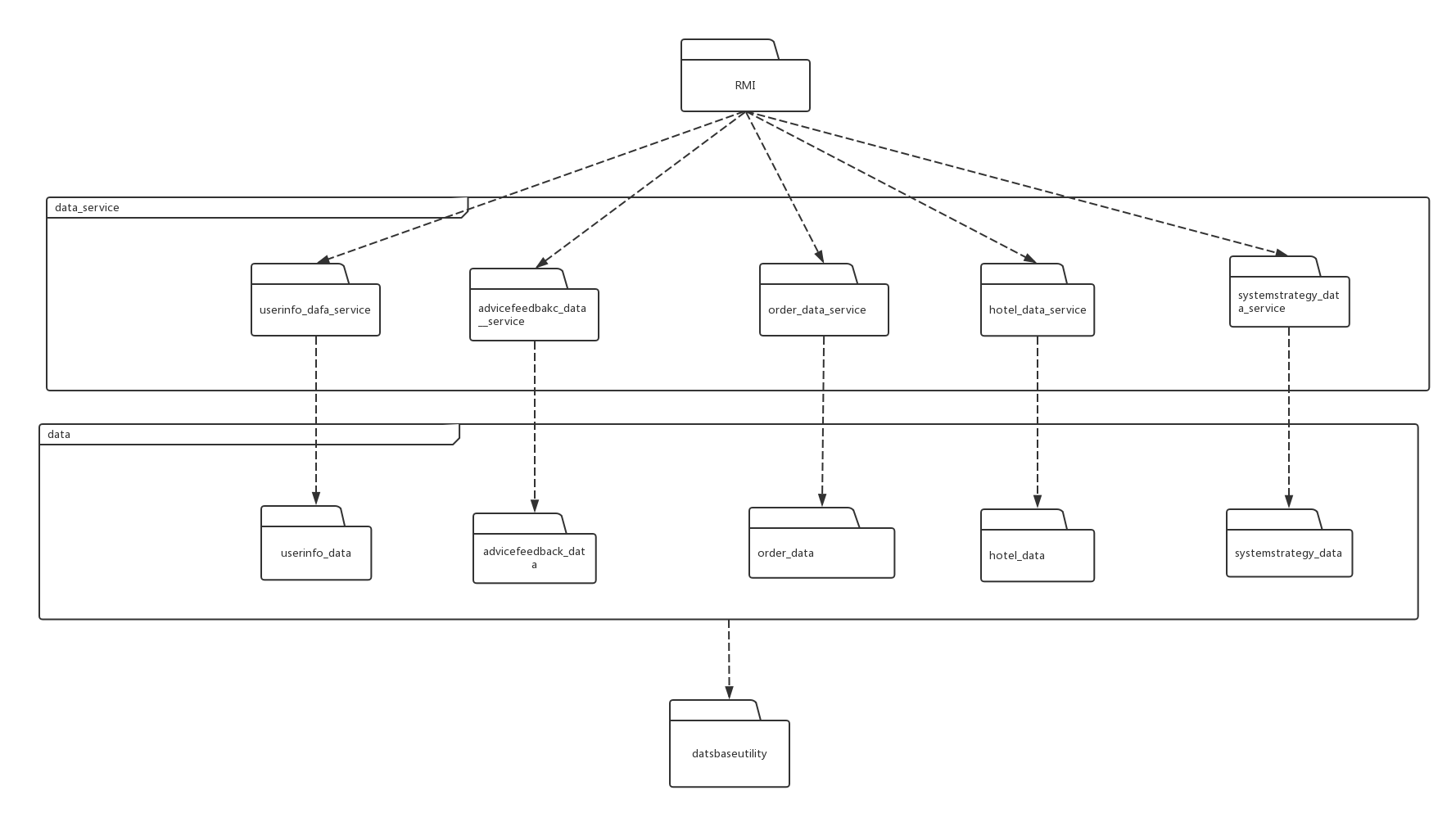


图4.1-2服务器端物理包结构设计图

* 1. **运行时进程**

多个客户端和一个服务器程序同时运行的时候，进程图如下（客户端程序在客户端机器上运行，服务器程序在服务器端机器上运行）：

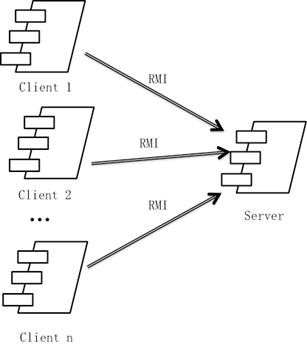


图4.2-1：预订系统的进程图

* 1. **物理部署**

客户端构件是放在客户端机器上，服务器端构件是放在服务器端机器上。在客户端节点上，还要部署RMIStub构件。由于JavaRMI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，不需要再独立部署。部署图如图4.3所示。

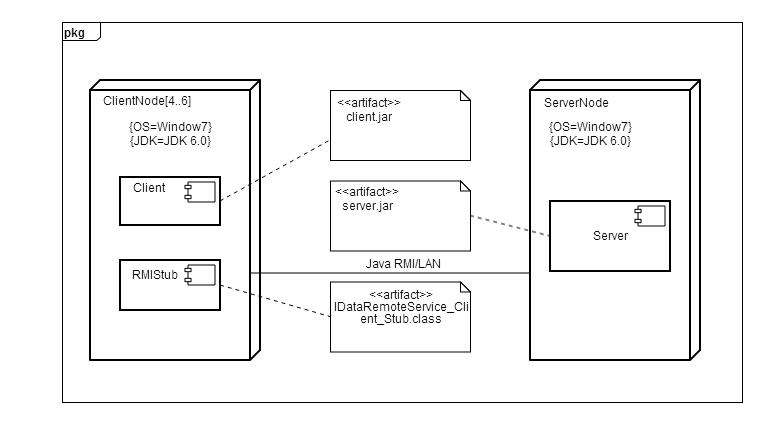


图4.3预订系统部署图

注：图示为win7系统，本系统实际使用的是win10系统

1. **接口视角**
   1. **模块职责**

客户端模块和服务器端模块视图分别如图5.1-1和图5.1-2所示。客户端各层和服务器端各层的职责分别如表5.1-1和表5.1-2所示。

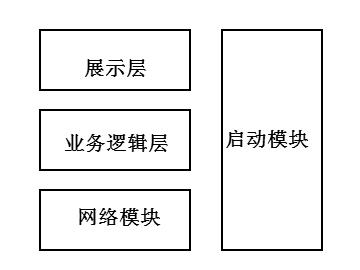


图5.1-1 客户端模块视图

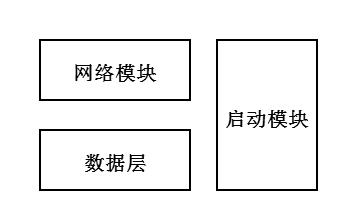


图5.1-2 服务器端模块视图

表5.1-1 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 | 基于窗口的进销存系统客户端用户界面。 |
| 业务逻辑层 | 对于用户界面的输入响应和业务处理逻辑。 |
| 客户端网络模块 | 利用Java RMI机制查找RMI服务,检测网络连接状态，进行断线重连 |

表5.1-2 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5.1-3所示。

表5.1-3 层之间调用的接口

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| userinfo\_blservice | 客户端界面展示层 | 客户端业务逻辑层 |
| advicefeedback\_blservice |
| usermanagement\_blservice |
| register\_blservice |
| login\_blservice |
| orderformation\_blservice |
| ordermanagement\_blservice |
| hotelview\_blservice |
| hotelmanagement\_blservice |
| hotelstrategy\_blservice |
| systemstrategy\_blsercice |
| userinfo\_dataservice | 客户端业务逻辑层 | 服务器端数据层 |
| advicefeedback\_dataservice |
| login\_dataservice |
| order\_dataservice |
| hotel\_dataservice |
| systemstrategy\_dataservice |

以uesrinfo为例，每一层之间都是由上层依赖了一个接口（需接口），而下层实现这个接口（供接口）。userinfo\_ui提供客户端所需要的界面，通过userinfo\_blservice来获得所需要的用户信息；userinfo\_bl向其他bl包调用必要的数据，通过user\_dataservice获取数据库数据，然后向userinfo\_blservice提供这些数据；user\_data层负责数据库的增删改查操作，并向user\_dataservice提供这些数据。

* 1. **用户界面层的分解**

根据需求，系统存在39个用户界面：登录界面，注册界面，客户界面，维护基本信息界面，查看订单界面，客户正常订单界面，客户未执行订单界面，客户异常和已撤销订单界面，预订过酒店界面，浏览酒店界面，酒店信息显示界面，酒店预订界面，生成订单界面，评价界面，注册会员界面，网站营销人员界面，信用充值界面，撤销异常订单界面，浏览异常订单界面，制定网站促销策略界面，酒店管理人员界面，录入可用客房界面，制定酒店促销策略界面，更新房间信息界面，退房信息界面，浏览订单界面，未执行订单界面，正常订单界面，异常与撤销订单界面，执行订单界面，网站管理人员界面，用户管理界面，客户管理界面，酒店工作人员管理界面，网站营销人员管理界面，添加酒店界面，录入酒店信息界面，添加酒店工作人员界面。

界面跳转如图5.2所示。

服务器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5.2-2所示。

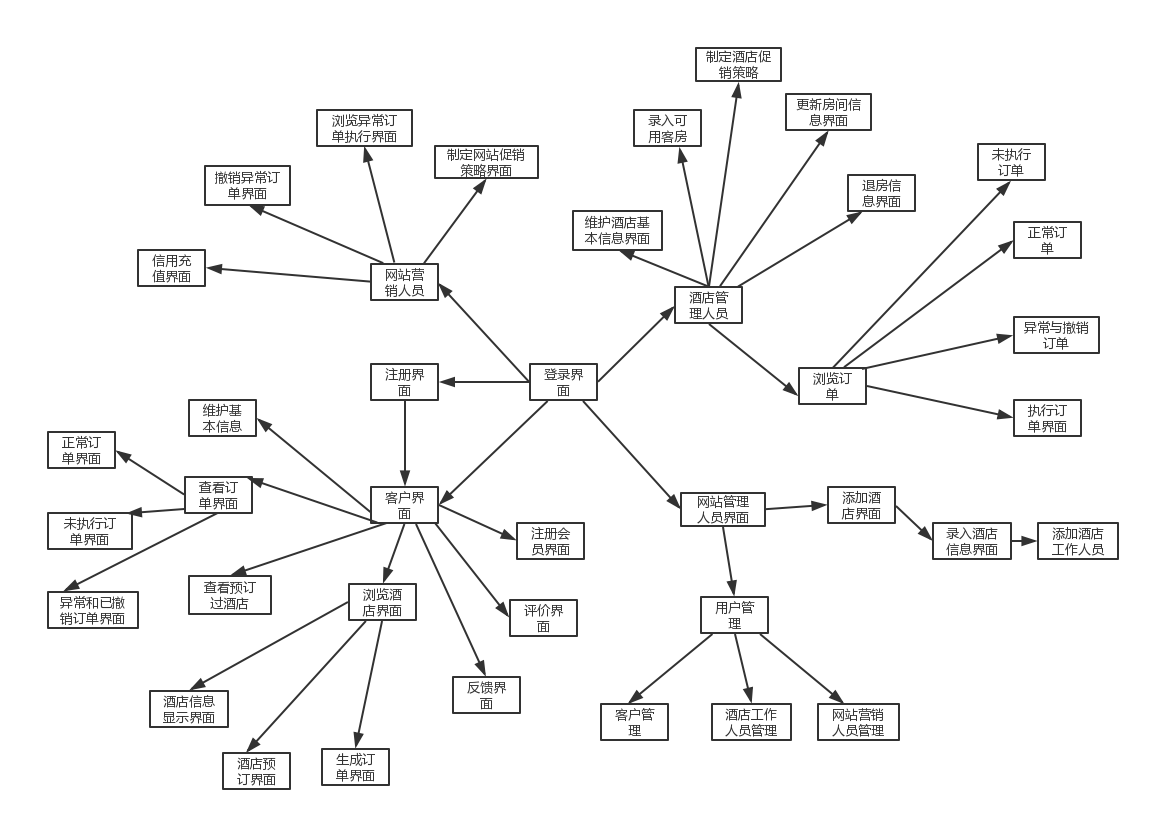


图5.2-2 用户界面类

* + 1. **用户界面层模块的职责**

如表5.2.1-1所示为用户界面层模块的职责。

表5.2.1-1 用户界面层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| main\_ui | 初始界面，提供主面板 |

* + 1. **用户界面层模块的接口规范**

用户界面层模块的接口规范如表5.2.2-1所示。

表5.2.2-1 用户界面层模块的接口规范

|  |  |  |
| --- | --- | --- |
| orderview\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示orderlistPanel |
| Ordermanagement\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示ordernamagementPanel |
| Login\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示LoginPanel |
| hotelview\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示hotellistPanel |
| advicefeedback\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示advicefeedbackPanel |
| Usermanagement\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示UsermanagementPanel |
| hotelstrategy\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示hotelstrategyPanel |
| userinfo\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示userinfoPanel |
| Systemstrategy\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示systemstrategyPanel |
| Hotelmanagement\_ui | 语法 | Init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示hotelmanagementPanel |
| register\_ui | 语法 | Init(args：String[]） |
| 前置条件 | 无 |
| 后置条件 | 显示reigsterPanel |

用户界面层需要的服务接口如表5.2.2-2所示。

表5.2.2-2 用户界面层模块需要的服务接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| businesslogicservice.login\_blservice | 登陆界面的业务逻辑接口 |
| businesslogicservice.orderview\_blservice | 提供客户处理与订单查看，管理的接口 |
| businesslogicservice.ordermanagement\_blservice | 提供酒店工作人员查看，处理酒店订单的接口 |
| businesslogicservice.hotelview\_blservice | 客户浏览酒店，搜索酒店各种条件操作的接口。 |
| Businesslogicservice.feedback\_blservice | 负责处理与客户评价操作相关的服务 |
| businesslogicservice.usermanagement\_blservice | 负责处理客户，酒店工作人员，网站营销人员信息管理相关的接口 |
| businesslogicservice.hotelstrategy\_blservice | 负责处理与酒店促销策略相关的接口 |
| Businesslogicservice.systemstrategy\_blservice | 提供网站营销人员促销策略制定管理业务逻辑的接口 |
| businesslogicservice.user\_blservice | 负责处理客户登录后界面的业务逻辑 |
| businesslogicservice.hotelmanagement\_blservice | 提供酒店工作人员对酒店客房，信息管理的接口 |
| businesslogicservice.LoginBLService | 登陆界面的业务逻辑接口 |
| businesslogicservice.\*BLService | 每个界面都有一个相应的业务逻辑接口 |

* + 1. **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。

* 1. **业务逻辑层**

业务逻辑层包括多个针对界面的业务逻辑处理对象。例如，login\_bl对象负责处理登录界面的业务逻辑；usermanagement\_bl对象负责管理用户的业务逻辑等等。业务逻辑层的设计如图5.3-1所示

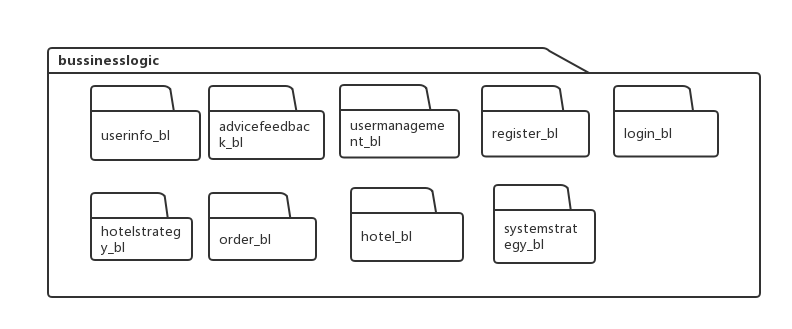


图 5.3-1业务逻辑层的设计

* + 1. **业务逻辑层模块的职责**

业务逻辑层各个模块的职责如表5.3.1-1所示。

表 5.3.1-1 业务逻辑层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| Userinfo\_bl | 负责处理用户对自己个人信息的维护的服务 |
| Advicefeedback\_bl | 负责处理用户和酒店工作人员对于网站的反馈和网站营销人员对于反馈的处理相关的服务 |
| Usermanagement\_bl | 提供管理人员的用户管理功能的服务 |
| Register\_bl | 负责与用户注册相关的服务 |
| Login\_bl | 负责提供验证用户登录的服务 |
| Hotelstrategy\_bl | 负责处理和酒店促销策略相关的服务 |
| Order\_bl | 负责提供与订单相关信息的服务 |
| Hotel\_bl | 负责处理与酒店相关的服务 |
| Systemstrategy\_bl | 负责处理与网站营销策略相关的服务 |

* + 1. **业务逻辑层模块的接口规范**

5.3.2-1 userinfo\_bl模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Userinfo.modify | 语法 | public result modifyUserinfo  (UserinfoVO userinfo) |
| 前置条件 | 已知用户，需要对自己的信息进行更改 |
| 后置条件 | 更改用户的个人信息 |
| 服务名 | | 服务 |
| User\_data.modify  （UserinfoPO userinfo） | | 修改单一持久化序列 |

5.3.2-2 advicefeedbackbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务(供接口) | | | |
| Advicefeedback.show() | 语法 | | Public List<AdvicefeedbackVO>  Show() |
| 前置条件 | | 需要知道所有的反馈信息 |
| 后置条件 | | 返回所有的反馈信息 |
| Advicefeedback.add  (AdvicefeedbackVO advicefeedback) | 语法 | | Public result addAdvicefeedback  (AdvicefeedbackVO advicefeedback) |
| 前置条件 | | 已知反馈信息，需要将反馈信息添加到反馈列表中 |
| 后置条件 | | 将反馈信息添加到反馈信息列表中 |
| Advicefeedback.getState | 语法 | | Public StateVO getStateofAdvice  (AdvicefeedbackVO advicefeedback) |
| 前置条件 | | 已知反馈信息，获得反馈信息的状态 |
| 后置条件 | | 返回反馈信息的状态 |
| Advicefeedback.modify | 语法 | | Public result modifyAdvice  (AdvicefeedbackVO advicefeedback) |
| 前置条件 | | 已知反馈信息，需要对反馈信息进行修改 |
| 后置条件 | | 更改反馈信息的内容 |
| 服务名 | | 服务 | |
| AdviceFeedback\_data.modify(AdviceFeedbackPO advicefeedback) | | 更改单一持久化序列 | |
| AdviceFeedback\_data.add(AdviceFeedbackPO advicefeedback) | | 新增单一持久化序列 | |
| AdviceFeedback\_data.find(AdviceFeedbackPO advicefeedback) | | 查找列表持久化序列 | |

5.3.2-3 usermanagementbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| UserManagement.add | 语法 | | Public result addUserManagement  (UserManagementVO user) |
| 前置条件 | | 需要新增一个用户 |
| 后置条件 | | 新增一个用户 |
| UserManagement.get | 语法 | | Public UserVO getUser  (String user\_id) |
| 前置条件 | | 已知用户的ID，获取用户 |
| 后置条件 | | 返回用户 |
| UserManagement.modify | 语法 | | Public result modifyUserManagement(UserVOuser) |
| 前置条件 | | 已知用户，需要修改用户的信息 |
| 后置条件 | | 更改用户的信息 |
| 服务名 | | 服务 | |
| User\_data.modify | | 更改单一持久序列 | |
| User\_data.add | | 新增大一持久化序列 | |
| User\_data.find | | 查找单一持久化序列 | |

5.3.2-4 registerbl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Register.add | 语法 | | Public result addRegister(String userID,String userpassword) |
| 前置条件 | | 已知用户ID和密码，需要注册一个用户 |
| 后置条件 | | 数据库中新增一个UserPO对象 |
| 服务名 | | 服务 | |
| User\_data.add(String userID,String userpassword) | | 新增一个持久化序列 | |

5.3.2-5 login\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Login.login | 语法 | | Public logVO login  (String user\_id,  String user\_password) |
| 前置条件 | | 启动登录服务，已知用户ID和用户密码 |
| 后置条件 | | 验证用户ID和密码，返回登录信息（登录是否成功、登录权限） |
| 服务名 | | 服务 | |
| User\_data.find(String user\_name) | | 查找单一持久化序列 | |

表5.3.2-6 Hotelstrategy\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Hotelstrategy.make | 语法 | | Public result makeHotelStrategy  (HotelStrategy hotelstrategy) |
| 前置条件 | | 已知酒店促销策略，将酒店促销策略添加到酒店促销策略列表 |
| 后置条件 | | 添加到酒店促销策略列表 |
| Hotelstrategy.modify | 语法 | | Public result modifyHotelstrategy  (HotelstrategyVO hotelstrategy) |
| 前置条件 | | 已知酒店促销策略，需要对酒店促销策略 |
| 后置条件 | | 更改酒店的促销策略 |
| Hotelstrategy.delete | 语法 | | Public result deleteHotelstrategy  (HotelstrategyVO hotelstrategy) |
| 前置条件 | | 已知酒店促销策略，删除酒店促销策略 |
| 后置条件 | | 删除酒店促销策略 |
| Hotelstrategy.save | 语法 | | Public result saveHotelStrategy  (HotelstrategyVO hotelstrategy) |
| 前置条件 | | 已知酒店促销策略，需要生成持久化序列 |
| 后置条件 | | 生成酒店促销策略的单一持久化序列 |
| Hotelstrategy.get | 语法 | | Public HotelstratefyVO getHotelStrategy(String Hotelstrategy\_name) |
| 前置条件 | | 已知酒店营销策略的名称，需要得到酒店促销策略 |
| 后置条件 | | 返回酒店促销策略 |
| Hotelstrategy.getList | 语法 | | Public List<HotelstrategyVO>  getList(String hotel\_id) |
| 前置条件 | | 已知酒店的ID，需要得到酒店的促销策略 |
| 后置条件 | | 返回酒店的促销策略 |
| 服务名 | | 服务 | |
| Hotel\_data.findstrategy(String hotelstrategy\_name) | | 查找单一持久化序列 | |
| Hotel\_data.findListstrategy (String hotel\_name) | | 查找列表持久化序列 | |
| Hotel\_data.add (HotelstrategyPO hotelstrategy) | | 新增单一持久化序列 | |
| Hotel\_data.modify (HotelstrategyPO hotelstrategy) | | 更改单一持久化序列 | |
| Hotel\_data.delete(HotelstrategyPO hotelstrategy) | | 删除单一持久化序列 | |

表 5.3.2-7 order\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Order.getState | 语法 | | Public StateVO getState  (String order\_id) |
| 前置条件 | | 已知订单号，需要获得订单的执行状态 |
| 后置条件 | | 返回订单的执行状态 |
| Order.getOrder | 语法 | | Public StateVO getOrder  (String order\_id) |
| 前置条件 | | 已知订单号，需要获得对应的订单 |
| 后置条件 | | 返回对应的订单 |
| Order.getOrdersofUsers | 语法 | | Public List<OrderVO> getOrdersOfUsers(String user\_id) |
| 前置条件 | | 已知用户的ID,需要获取用户的订单信息 |
| 后置条件 | | 返回用户的订单信息 |
| Order.createOrder | 语法 | | Public OrderVO createOrder  (OrderVO order\_info) |
| 前置条件 | | 已知所有的订单信息，需要生成订单 |
| 后置条件 | | 根据订单类给出报价和时间信息，返回订单类 |
| Order.saveOrder | 语法 | | Public Result saveOrder  (Order order\_info) |
| 前置条件 | | 已知订单，需要将订单持久化保存 |
| 后置条件 | | 持久化保存订单的信息 |
| Order.changeState | 语法 | | Public result changeState  (Order order\_info) |
| 前置条件 | | 已知订单，需要对于订单的状态进行修改 |
| 后置条件 | | 更改订单的状态信息 |
| 服务名 | | 服务 | |
| Order\_data.add(OrderPO order) | | 新增单一持久化序列 | |
| Order\_data.find(String order\_id) | | 查找单一持久化序列 | |
| Order\_data.modify(OrderPO order) | | 改变单一持久化序列 | |
| Order\_data.findOrders (String user\_id) | | 查找列表持久化序列 | |

5.3.2-8 hotel\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Hotel.getHotel | 语法 | | Public HotelVO getHotel  (String hotel\_name) |
| 前置条件 | | 已知酒店的名称，查询酒店的信息 |
| 后置条件 | | 返回酒店 |
| Hotel.modifyHotel | 语法 | | Public result modifyHotel  (HotelVO hotel) |
| 前置条件 | | 已知酒店的信息，需要对于酒店的信息进行更改 |
| 后置条件 | | 更改酒店的房间信息 |
| Hotel.saveHotel | 语法 | | Public result saveHotel  (HotelVO hotel ) |
| 前置条件 | | 已知酒店，需要将酒店持久化保存 |
| 后置条件 | | 持久化保存酒店信息 |
| Hotel.addHotel | 语法 | | Public HotelVO addHotel  (HotelVO hotel) |
| 前置条件 | | 已知酒店，需要添加酒店的信息 |
| 后置条件 | | 根据酒店信息给出的房间数量、类型信息，返回酒店的PO类 |
| Hotel.deleteHotel | 语法 | | Public result deleteHotel  (HotelVO hotel) |
| 前置条件 | | 已知酒店，需要将酒店删除 |
| 后置条件 | | 根据酒店的信息，将酒店的PO类删除 |
| Hotel.showHotel | 语法 | | Public List<HotelVO> showHotel() |
| 前置条件 | | 需要得到所有酒店的信息 |
| 后置条件 | | 返回所有酒店的信息 |
| 服务名 | | 服务 | |
| Hotel\_data.add(HotelPO hotel) | | 新增单一持久化序列 | |
| Hotel\_data.delete(HotelPO hotel) | | 删除单一持久化序列 | |
| Hotel\_data.modify(HotelPO hotel) | | 改变单一持久化序列 | |
| Hotel\_data.show() | | 返回列表持久化序列 | |
| Hotel\_data.find(String hotel\_name) | | 查找列表持久化序列 | |

5.3.2-9 systemstrategy\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| systemstrategy.make | 语法 | Public systemstrategyVO makeSystemStrategy  (systemstrategyVO systemstrategy) | |
| 前置条件 | 已知网站营销策略，需要生成网站营销策略 | |
| 后置条件 | 根据网站营销策略给出的信息返回PO类 | |
| Systemstrategy.get | 语法 | Public systemstrategyVO getsystemstrategy  (String systemstrategy\_name) | |
| 前置条件 | 已知网站营销策略的名称，需要知道具体的网站营销策略的信息 | |
| 后置条件 | 返回网站营销策略名称对应的网站营销策略 | |
| systemstrategy.Save | 语法 | Public result saveSystemStrategy  (systemstrategyVO systemstrategy) | |
| 前置条件 | 已知网站营销策略，需要将网站营销策略持久化保存 | |
| 后置条件 | 持久化保存网站营销策略的信息 | |
| systemstrategy.modify | 语法 | Public result modifySystemStrategy(SystemStrateyVO systemstrategy) | |
| 前置条件 | 已知网站营销策略，需要对于网站营销策略 | |
| 后置条件 | 更改网站营销策略信息 | |
| systemstrategy.delete | 语法 | Public result deleteSystemStrategy  (systemstrategyVO systemstrategy) | |
| 前置条件 | 已知网站营销策略，需要将已经制定的网站营销策略删除 | |
| 后置条件 | 删除网站营销策略 | |
| Systemstrategy.show | 语法 | Public List<systemdstrategyVO> showSystemStrategy() | |
| 前置条件 | 需要知道所有已经指定的网站营销策略 | |
| 后置条件 | 返回所有的网站营销策略 | |
| 服务名 | | | 服务 |
| Systemstrategy\_data.find(String systemstrategy\_name) | | | 查找单一持久化序列 |
| Systemstrategy\_data.modify(systemstrategyPO systemstrategy) | | | 改变单一持久化序列 |
| systemstrategy\_data.delete | | | 删除单一持久化序列 |
| Systemstrategy\_data.add(systemstrategyPO systemstrategy) | | | 新增单一持久化序列 |
| Systemstrategy\_data.show() | | | 查找列表持久化序列 |

* 1. **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、Txt文件、数据库等。

* + 1. **数据层模块的职责**

数据层模块的职责如下表所示：

|  |  |
| --- | --- |
| 模块 | 职责 |
| UserData | 提供客户数据集体载入、保存、增删改查服务 |
| AdviceFeedbackData | 提供意见反馈数据的增改查服务 |
| OrderData | 提供订单数据集体载入、保存、增删改查服务 |
| HotelData | 提供酒店信息的集体载入、保存、增删改查和酒店销售策略的载入、保存和增删改查服务以及对酒店客房信息的管理服务 |
| SystemStrategyData | 提供网站销售策略集体载入保存和增删改查功能 |

* + 1. **数据层模块的接口规范**

表5.4.2-1数据层UserData模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| User\_data.add | 语法 | Public ResultMessaage add(Use String userID,String userpassword) |
| 前置条件 | 同样的id在po数据中不存在 |
| 后置条件 | 数在数据文件中增加一个po记录 |
| User\_data.find | 语法 | Public UserPO find(String id) |
| 前置条件 | 无 |
| 后置条件 | 返回相应的UserPO结果；如果id不存在，则返回null |
| User\_data.modify | 语法 | public ResultMessage modify(UserPO userInfo) |
| 前置条件 | 相同id的po在数据文件中已存在 |
| 后置条件 | 更细一个po |
| User\_data.delete | 语法 | public ResultMessage delete(String id) |
| 前置条件 | 相同id的po在数据文件中已存在 |
| 后置条件 | 删除一个po记录 |
| User\_data.modify | 语法 | public ResultMessage modify(UserPO userInfo) |
| 前置条件 | 相同id的po在数据文件中已存在 |
| 后置条件 | 更新一个po |
| User\_data. getHistory | 语法 | Public List<HistoryPO> getHistory(string user\_id); |
| 前置条件 | 逻辑层请求获得客户的历史订房信息 |
| 后置条件 | 返回历史订房信息 |

表5.4.2-2数据层AdviceFeedbackData模块的接口规范

|  |  |  |
| --- | --- | --- |
| AdviceFeedback\_data.add(AdviceFeedbackPO advicefeedbackPO) | 语法 | public ResultMessage add(AdviceFeedbackPO advicefeedbackPO |
| 前置条件 | 同样ID的po在数据文件中不存在 |
| 后置条件 | 在数据文件中增加一个PO记录 |
| AdviceFeedback\_data.find(string adviceID) | 语法 | public AdviceFeedbackPO find(string adviceID) |
| 前置条件 | 无 |
| 后置条件 | 返回一个相应的AdviceFeedbackPO结果；如果对应的不存在，则返回null |
| AdviceFeedback\_data.show(string adviceID) | 语法 | public ArrayList<AdviceFeedbackPO>show(string adviceID) |
| 前置条件 | 无 |
| 后置条件 | 显示相应的反馈信息数据 |
| AdviceFeedback\_data.modify(AdviceFeedbackPO advicefeedback) | 语法 | public ResultMessage  modify(AdviceFeedbackPO advicefeedback) |
| 前置条件 | 相同id的po在数据文件已存在 |
| 后置条件 | 更新一个po |

表5.4.2-3数据层OrderData模块的接口规范

|  |  |  |
| --- | --- | --- |
| Order\_data.add | 语法 | public ResultMessage add(OrderPO order) |
| 前置条件 | 相同id的po在数据文件中不存在 |
| 后置条件 | 在数据文件中增加一条新的order信息 |
| Order\_data.modify | 语法 | public ResultMessage modify(OrderPO order) |
| 前置条件 | 相同id的po在数据文件中已存在 |
| 后置条件 | 更新一个po |
| Order\_data.find | 语法 | public OrderPO find(string userID) |
| 前置条件 | 无 |
| 后置条件 | 返回与对应id的OrderPO,如果对应user的id不存在，返回Null |
| Order\_data.addException | 语法 | public ResultMessage addException(OrderPO order) |
| 前置条件 | 相同id的order在数据文件中不存在 |
| 后置条件 | 数据文件中添加一条ExceptionOrder记录 |
| Order\_data.addExcuted | 语法 | public ResultMessage addE[xecuted](http://www.baidu.com/link?url=ShK40S5saIJW8LZDThVvZYFjKS9PQM7sktcLBu9JWVK57cHU_E1m_TDqMMLN1ytQx8P9vJxKhcVdnXp4g2C2imDgaGhRt0ntQPjZS4YpA7S) (OrderPO order) |
| 前置条件 | 相同id的order在数据文件中不存在 |
| 后置条件 | 数据文件中添加一条ExcutedOrder记录 |
| Order\_data.addRevocation | 语法 | public ResultMessage addRevocation (OrderPO order) |
| 前置条件 | 相同id的order在数据文件中不存在 |
| 后置条件 | 数据文件中添加一条RevocationOrder记录 |
| Order\_data.findException | 语法 | public ExceptionOrderPO findException(String order\_id) |
| 前置条件 | 无 |
| 后置条件 | 返回与对应id的ExceptionOrder,如果order\_id不存在，返回Null |
| Order\_data.findExcuted | 语法 | public ExcutedOrderPO findExcuted(String order\_id) |
| 前置条件 | 无 |
| 后置条件 | 返回与对应id的ExcutedOrder,如果order\_id不存在，返回Null |
| Order\_data.findRevocation | 语法 | public RevocationOrderPO findRevocation (String order\_id) |
| 前置条件 | 无 |
| 后置条件 | 返回与对应id的RevocationOrder,如果order\_id不存在，返回Null |
| Order\_data.CreatBill | 语法 | public ResultMessage creatBill(OrderPO order) |
| 前置条件 | 执行订单后逻辑层请求生成和打印bill |
| 后置条件 | 返回由订单和账单信息生成的bill信息 |

表5.4.2-4数据层HotelData模块的接口规范

|  |  |  |
| --- | --- | --- |
| Hotel\_data.add | 语法 | Public ResultMessage add(HotelPO hotel) |
| 前置条件 | 同样id的po在数据文件中不存在 |
| 后置条件 | 在数据文件中增加一个新的PO记录 |
| Hotel\_data.delete | 语法 | Public ResultMessage delete(string hotel\_id) |
| 前置条件 | 同样id的po在数据文件中已存在 |
| 后置条件 | 删除一个PO记录 |
| Hotel\_data.Modify | 语法 | Public ResultMessage Modify(HotelPO hotel) |
| 前置条件 | 同样id的po在数据文件中已存在 |
| 后置条件 | 更新数据文件中相应的PO |
| Hotel\_data.find | 语法 | Public HotelPO find(String hotel\_name) |
| 前置条件 | 无 |
| 后置条件 | 如果hotel\_name存在,返回与对应id的RevocationOrder,如果hotel\_name不存在，返回Null |
| Hotel\_data.show | 语法 | public ArrayList<HotelPO>show(string hotel\_name) |
| 前置条件 | 无 |
| 后置条件 | 显示相应的hotel信息 |
| Hotel\_data.addStrategy | 语法 | public ResultMessaage addHotelStrategy(HotelStrategyPO po) |
| 前置条件 | 相同的HotelStrategyPO的po在数据文件中不存在 |
| 后置条件 | 在数据文件中增加一个po数据 |
| Hotel\_data.deleteStrategy | 语法 | public ResultMessaage deleteHotelStrategy(HotelStrategyPO po) |
| 前置条件 | 相同的HotelStrategyPO的po在数据文件中已存在 |
| 后置条件 | 删除一个po记录 |
| Hotel\_data.modifyStrategy | 语法 | public ResultMessaage modifyHotelStrategy(HotelStrategyPO po |
| 前置条件 | 相同的HotelStrategyPO的po在数据文件中已存在 |
| 后置条件 | 更新一个po记录 |
| Hotel\_data.findStrategy | 语法 | public ResultMessaage findStrategy(String hotelstrategy\_name) |
| 前置条件 | 无 |
| 后置条件 | 返回一个hotelSrategy的po数据 |
| Hotel\_data.findListStrategy | 语法 | public ResultMessaage findListStrategy(HotelstrategyPO hotelstrategy) |
| 前置条件 | 无 |
| 后置条件 | 返回一个HotelstrategyList的po数据 |
| Hotel\_data.addRoom | 语法 | Public ResultMessage addRoom(RoomPO po) |
| 前置条件 | 相同的RoomPO的po在数据文件中不存在 |
| 后置条件 | 在数据文件中增加一个新的RoomPO记录 |
| Hotel\_data.ModifyRoom | 语法 | Public ResultMessage modifyRoom (RoomPO room) |
| 前置条件 | 同样id的po在数据文件中已存在 |
| 后置条件 | 更新一个po |
| Hotel\_data.deleteRoom | 语法 | Public ResultMessage deleteRoom(RoomPO room) |
| 前置条件 | 同样id的po在数据文件中已存在 |
| 后置条件 | 删除一个po |

表5.4.2-5数据层SystemStrategyData模块的接口规范

|  |  |  |
| --- | --- | --- |
| SystemStrategy\_data.add | 语法 | public ResultMessaage add(StrategyPO po) |
| 前置条件 | 相同的StrategyPO的po在数据文件中不存在 |
| 后置条件 | 在数据文件中增加一个po数据 |
| SystemStrategy\_data.delete | 语法 | public ResultMessaage delete (StrategyPO po) |
| 前置条件 | 相同的StrategyPO的po在数据文件中已存在 |
| 后置条件 | 删除一个po记录 |
| SystemStrategy\_data.modify | 语法 | public ResultMessaage modify(StrategyPO po |
| 前置条件 | 相同的StrategyPO的po在数据文件中已存在 |
| 后置条件 | 更新一个po记录 |
| SystemStrategy\_data.find | 语法 | public ResultMessaage find(String systemstrategy\_name) |
| 前置条件 | 无 |
| 后置条件 | 如果存在相应的strategy，则返回systemStrategy数据；如果没有，则返回null |

1. **信息视角**
   1. **数据持久化对象**

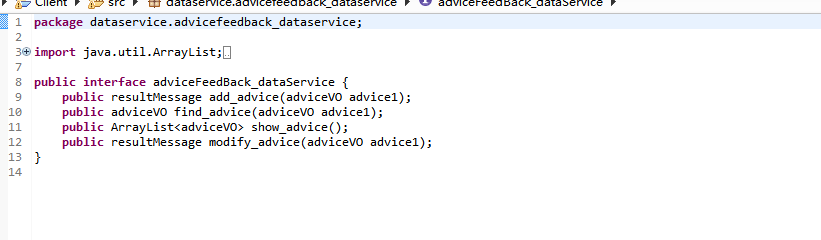
系统的PO类就是对应的相关的实体类，在此只做简单的介绍，如表6.1所示：

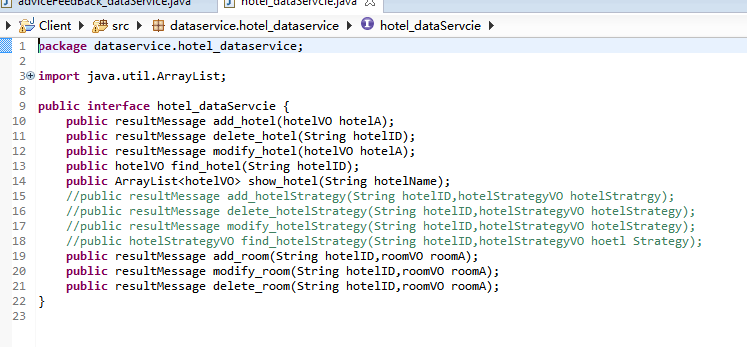
|  |  |
| --- | --- |
| 类名 | 包含的属性 |
| userPO | userID,userName,userPassword,  userRegisterTime,userType,  userCondition |
| orderPO | orderDate,orderHotelName,  orderFinalPrice,orderOriginPrice,  orderSystemStrategy,  orderCondition,orderUserID |
| hotelPO | hotelName,hotelID,hotelStaffID,hotelInfo,hotelStar,hotelAssessment,hotelOrderList |
| hotelstrategyPO | hotelstrategyInfo |
| systemstrategyPO | systemstrategyNormal,  systemstrategySpecial,  systemstrategyMember |
| advicePO | adviceInfo,adviceSenderID  adviceReplyID,adviceReply |

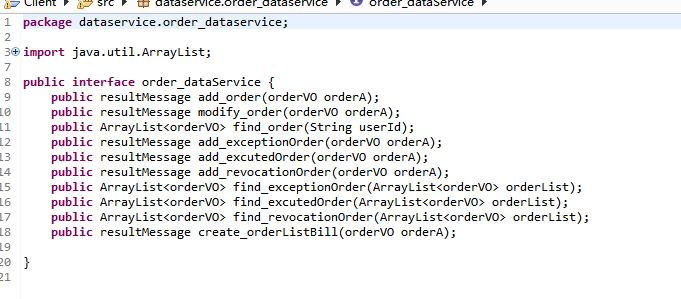
1. **原型代码**

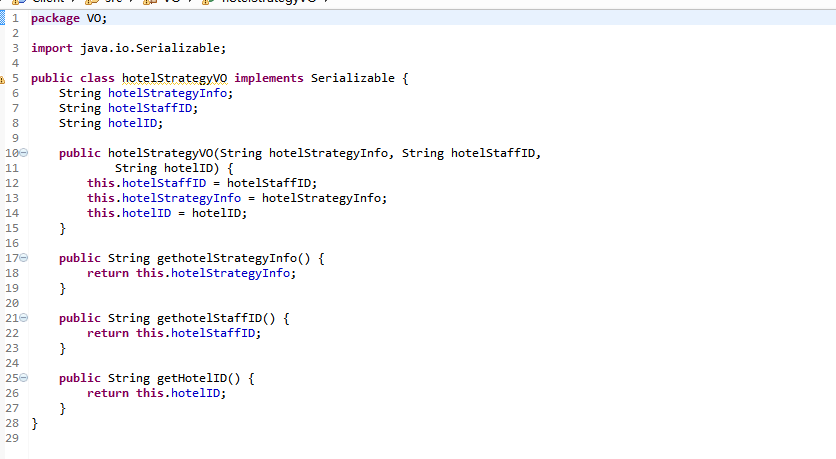
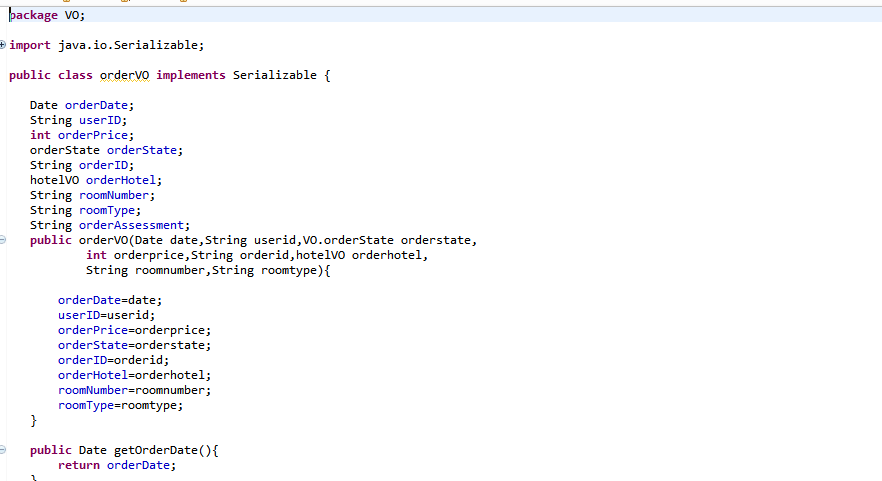
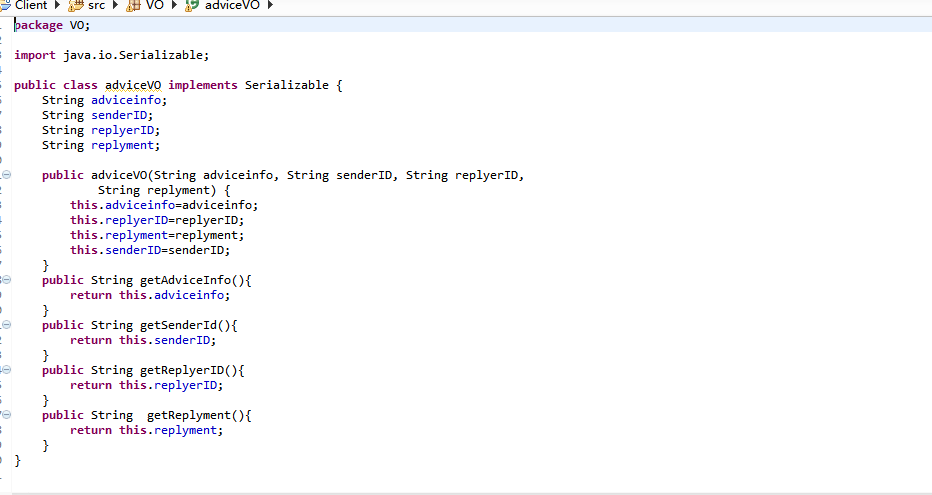
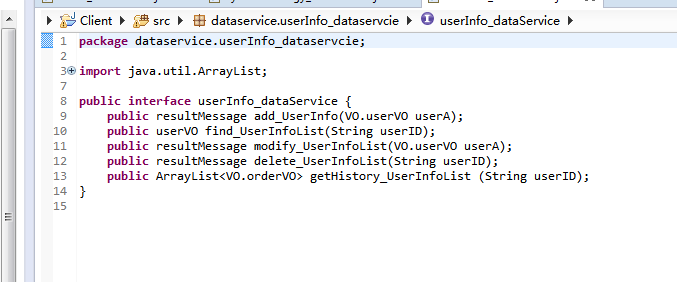
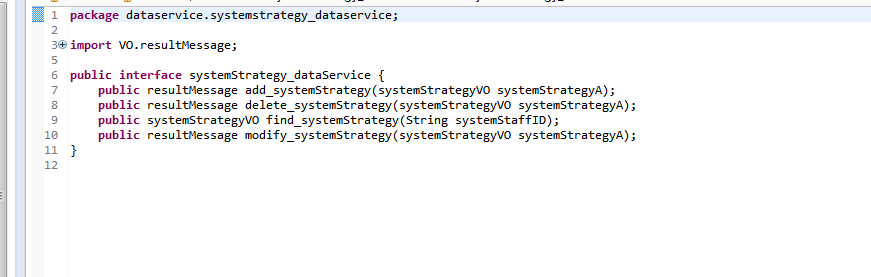
各文件原型代码详见截图

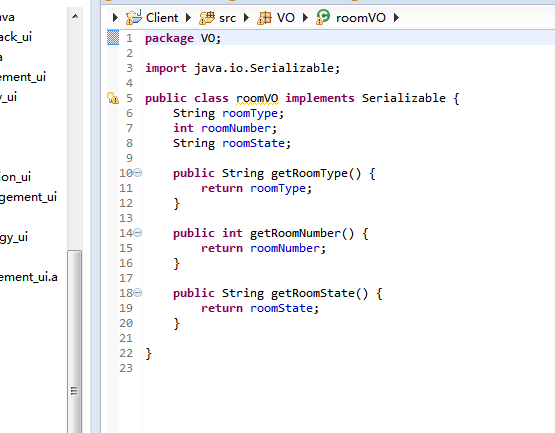
桩程序：

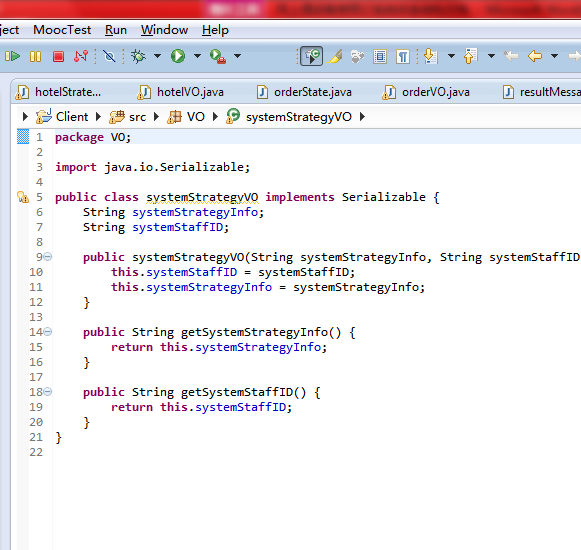


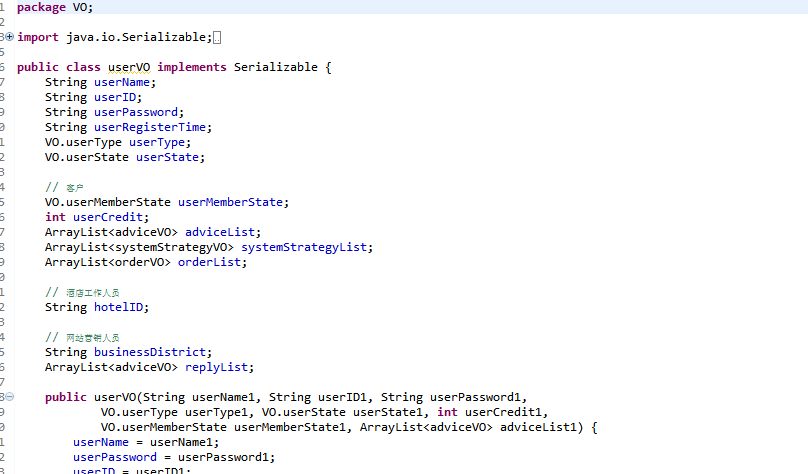






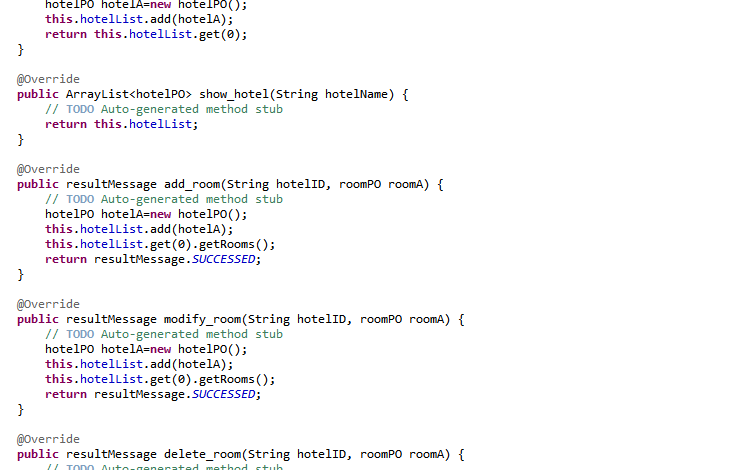


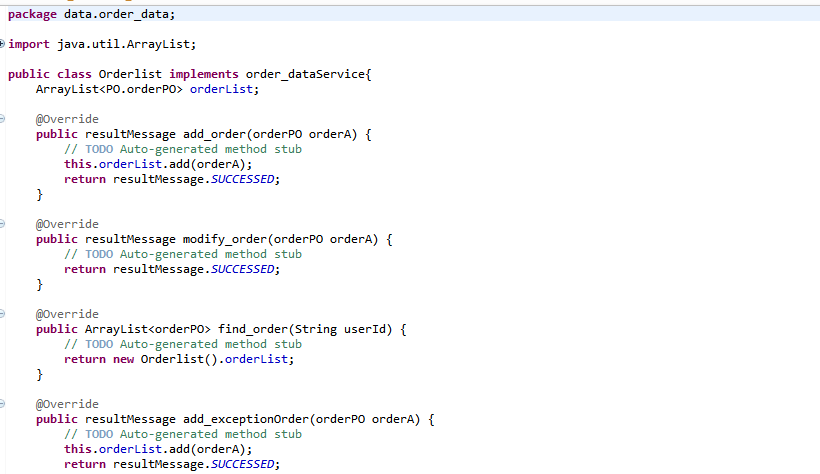




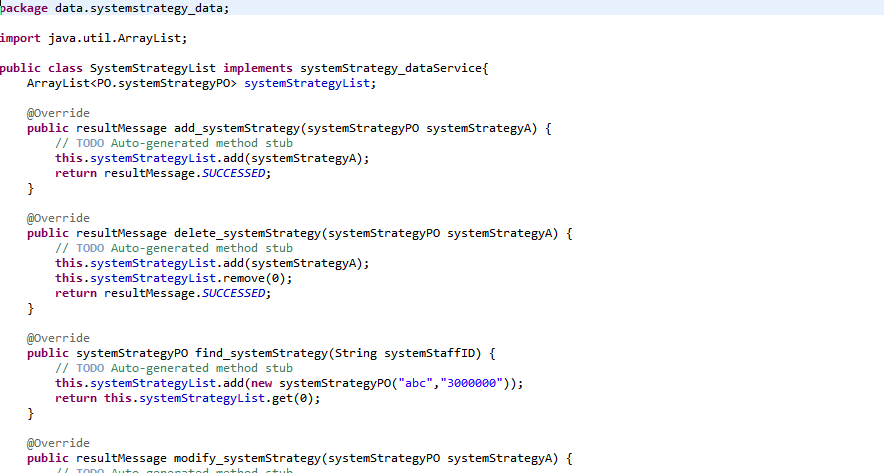














驱动程序：

