

网上酒店预订系统

体系结构设计文档1.0版



2016-10-8

猫尾巴小组

高翔、曾欢、黄韵斐、黄泽鹏

# ·更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 高翔 | 2016-10-10 | 最初草稿 | V0.1 |
| 高翔 | 2016-10-10 | 完善逻辑、开发包图以及最终开发表 | V0.2 |
| 全体成员 | 2015-10-11 | 修改包图、增加用户界面跳转图、完善部分文档 | V0.3 |
| 曾欢 | 2015-10-12 | 设计数据层结构 | V0.46 |
| 黄泽鹏 | 2015-10-13 | 设计客户端结构与接口 | V0.5 |
| 黄韵斐 | 2015-10-14 | 进一步完善客户端接口 | V0.6 |
| 黄泽鹏 | 2015-10-15 | 对类图结构提出异议，并更正 | V0.7 |
| 曾欢 | 2015-10-15 | 从新设计数据层结构 | V0.8 |

# 目录

目录

[·更新历史 1](#_Toc463959686)

[目录 2](#_Toc463959687)

[**1.1.** **编制目的** 3](#_Toc463959688)

[**1.2 词汇表** 3](#_Toc463959689)

[**1.3参考资料** 3](#_Toc463959690)

[**2.** **产品描述** 3](#_Toc463959691)

[**3.** **逻辑视角** 3](#_Toc463959692)

[4](#_Toc463959693)

[**4.** **组合视角** 5](#_Toc463959694)

[**4.1.** **开发包图** 5](#_Toc463959695)

[4.2. 运行时进程 8](#_Toc463959696)

[4.3. 物理部署 8](#_Toc463959697)

[**5.** **接口视角** 9](#_Toc463959698)

[**5.1.** **模块的职责** 9](#_Toc463959699)

[**5.2.** **用户界面层的分解** 12](#_Toc463959700)

[**5.2.1.** **用户界面层模块的职责** 12](#_Toc463959701)

[**5.2.2.** **用户界面模块的接口规范** 13](#_Toc463959702)

[**5.2.3.** **用户界面模块设计原理** 13](#_Toc463959703)

[**5.3.** **业务逻辑层的分解** 13](#_Toc463959704)

[**5.3.1.** **业务逻辑层模块的职责** 14](#_Toc463959705)

[**5.4.** **数据层的分解** 16](#_Toc463959706)

[**5.4.1.** **数据层模块的职责** 16](#_Toc463959707)

[**6.** **信息视角** 16](#_Toc463959708)

[**6.1.** **数据持久化对象** 16](#_Toc463959709)

[**6.2.** **文件格式** 17](#_Toc463959710)

# 1.引言

* 1. **编制目的**

本报告详细完成对网上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** |
| OHRS | 网上酒店预订系统 |
| Od | 订单 |
| FillOd | 成功订单 |
| FailOd | 异常订单 |
| IngOd | 正在进行订单 |
| CancleOd | 撤销订单 |
| Credit | 诚信 |
| Hotel | 酒店 |
| Market | 网站营销 |
| Control | 管理员 |
| Infor | 信息 |
| Impl | 接口 |

**1.3 参考资料**

《网上酒店预订系统用例文档 初始版》

《网上酒店预订系统需求规格说明文档V1.0》

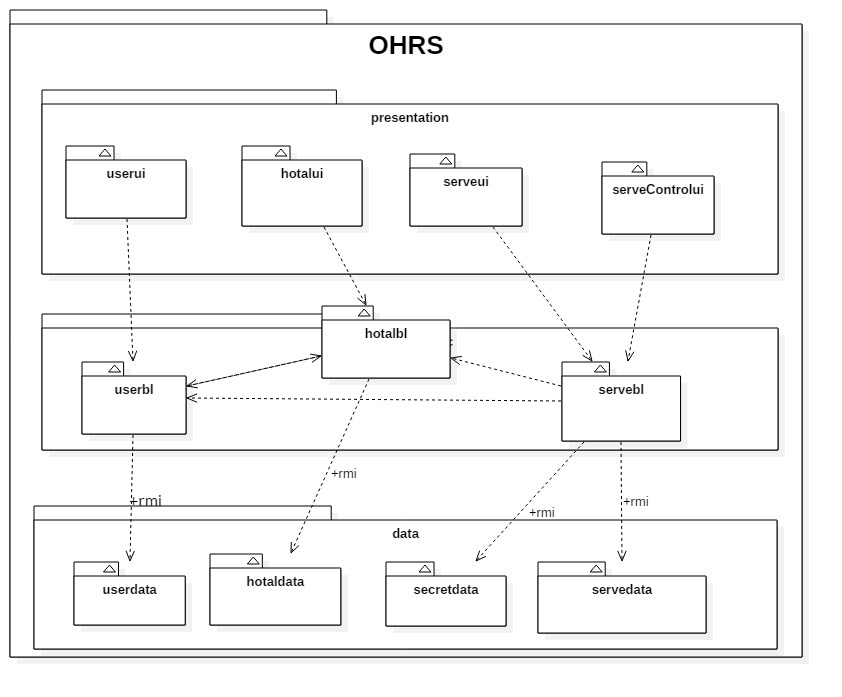
1. **产品描述**

参考《网上酒店预订系统用例文档》和《酒店预订系统规格说明文档》中对产品的概括描述。

1. **逻辑视角**

本系统由于不需要用到web开发，所以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。展示层负责用户交互界面的实现，包括GUI页面。业务逻辑层包含业务逻辑处理的实现，体现在客户端中各种方法的调用。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。

**图一：分层结构**



**图二：逻辑设计方案**

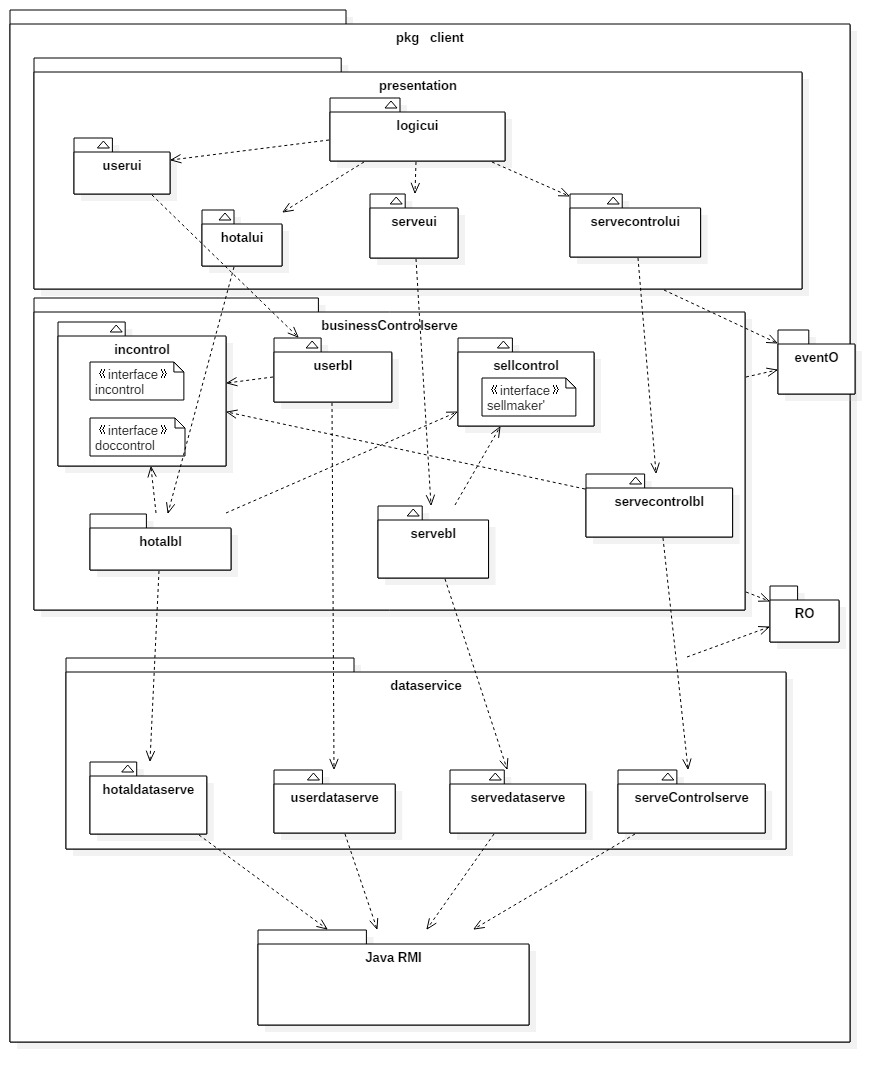
(提示： 逻辑层因为使用rmi的缘故，部分逻辑结构由data包实现)

1. **组合视角**
   1. **开发包图**

网上酒店预订系统的开发包设计如表1所示：

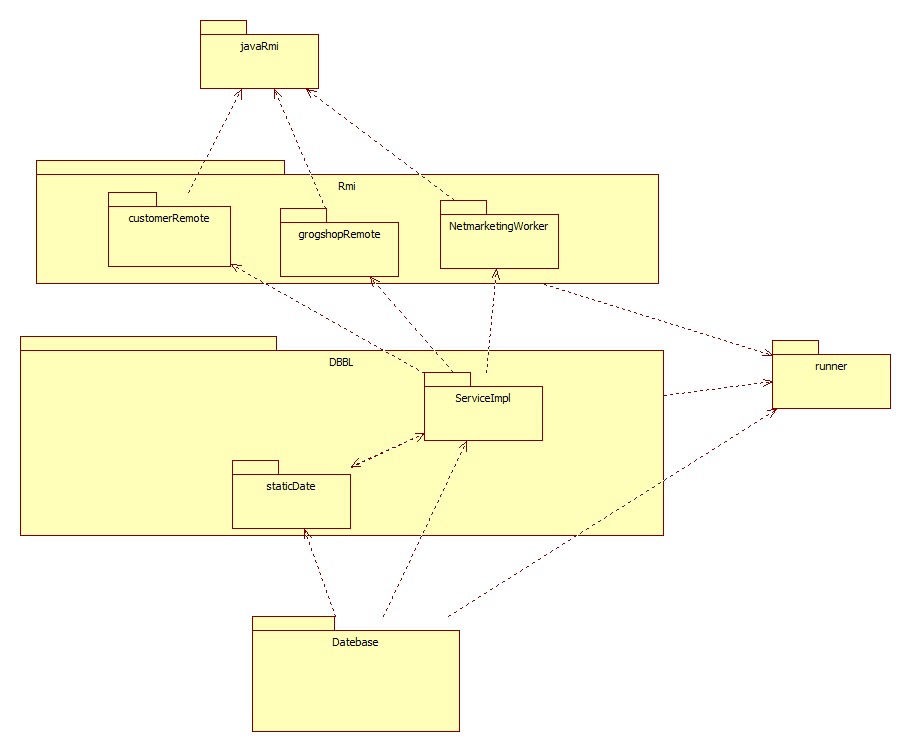
**表1网上酒店管理系统的最终开发包设计**

|  |  |
| --- | --- |
| **开发（物理）包** | **依赖的其他开发包** |
| Datebase | Information , serviceImpl |
| information | Datebase,serviceImpl |
| remote | serviceImpl |
| staticDate | serviceImpl |
| serviceImpl | Ui, remote,datebase,information |
| ui | serviceImpl |
| Frame | (界面的开创者) |
| Frameui | Frame |
| Actioncontrol | Awt.event |
| Userui | Userbl |
| Userbl | Hotalbl、userdata |
| Userdata |  |
| Hotalui | Hotalbl |
| Hotalbl | Userbl、hotaldata |
| Hotaldata |  |
| Serveui | Servebl |
| Serve’controlui | Servebl |
| Servebl | Servedata、userbl、hotalbl |
| Servedata |  |
| Incontrol |  |
| Sellmaker |  |
| Doccontrol |  |



图三：客户端开发包图

（提示：eventO包中是对监听器对象的声明，RO是对网络联通对象的声明）

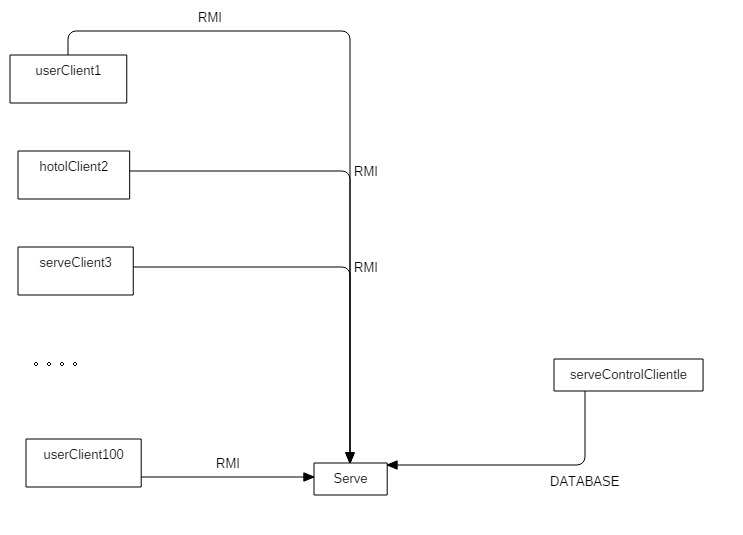


图四： 服务端开发包图

（提示：服务器端包含逻辑结构，封存在DBbl包中）

## 运行时进程

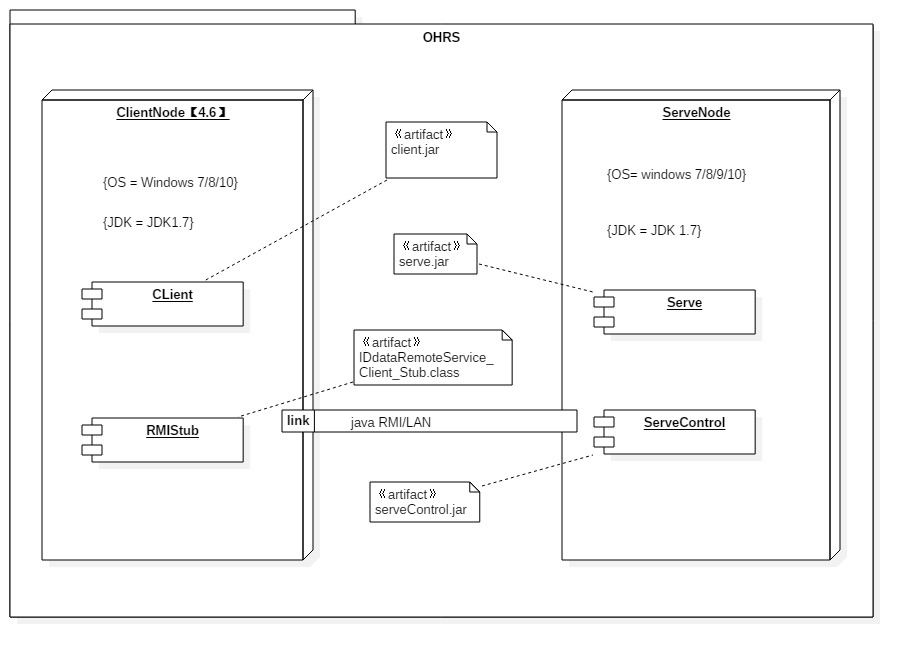
在网上酒店管理系统中，会有多个客户端进程和一个服务器端进程，其进程图如图5所示。结合部署图，客户端进程是在客户端机器上运行，服务器端进程是在服务器端机器上运行，并且服务端进程与网站管理人员进程共存于一台机器，并通过数据库语言进行链接。



**图5 进程图**

## 物理部署

网上酒店预订系统中客户端构件是放在客户端机上，服务器端构件与网站管理人员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Java RMI 构件属于JDK 1.8的一部分。所以，在系统JDK环境已经设置好的情况下，不需要独立部署。部署图如图6所示。



**图 6 部署图**

(提示：最终产品客户端是集成的)

1. **接口视角**
   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。

**启动模块**

**业务逻辑层**

**展示层**

**网络模块**

**图7 客户端模块视图**

**启动模块**

**网络模块**

**业务逻辑层**

**数 据 层**

**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 | 基于窗口的网上酒店预订系统客户端用户界面 |
| 业务逻辑层 | 对于用户界面的输入、行为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 |
| 业务逻辑层 | 进行对数据的算法与整合 |

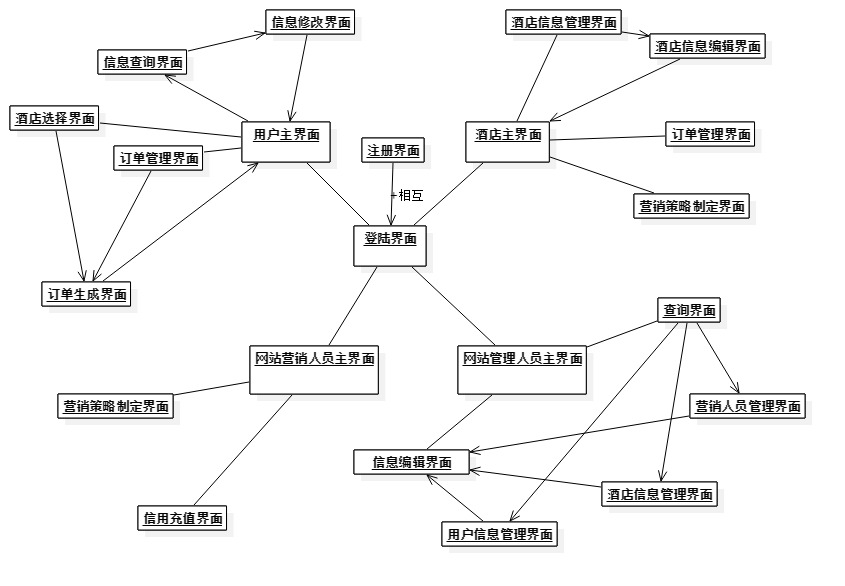
每一层只是使用下方直接接触的层。层与层之间仅仅是通过接口的调用来完成的。层之间调用的接口如表4所示。

**表4 层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| searchhotelActionListener.  actionPerformed  placeorderActionListener2.  actionPerformed  ChangeInformationActionLis  tener.actionPerformed  informationActionListener.  actionPerformed  RegistActionListener2.  actionPerformed  LoginActionListener.  actionPerformed | 客户端展示层 | 客户端业务逻辑层 |
| CustomerRemote.login  CustomerRemote.CreateNewCustomer  CustomerRemote.checkInformation  CustomerRemote.modifyInformation  CustomerRemote.searchGrogsho | 客户端业务逻辑层 | 服务器端数据层 |

* 1. **用户界面层的分解**

根据需求，系统存在23个用户界面（体积大的为主界面）：各个界面之间的关系，以及个界面的名称如下图所示。



**图9 网上酒店预订系统界面跳转图**

* + 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 |

* + 1. **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 |
| bl.accountservice | 登录界面的业务逻辑接口 |
| bl.\*blservice | 每个ui模块会有对应的数据逻辑接口 |

* + 1. **用户界面模块设计原理**

用户界面拟用java的Swing和AWT库来实现。

* 1. **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。比如对登陆对象类型的判断，用以决定跳转位置等的业务逻辑。业务逻辑层的设计如下图所示：

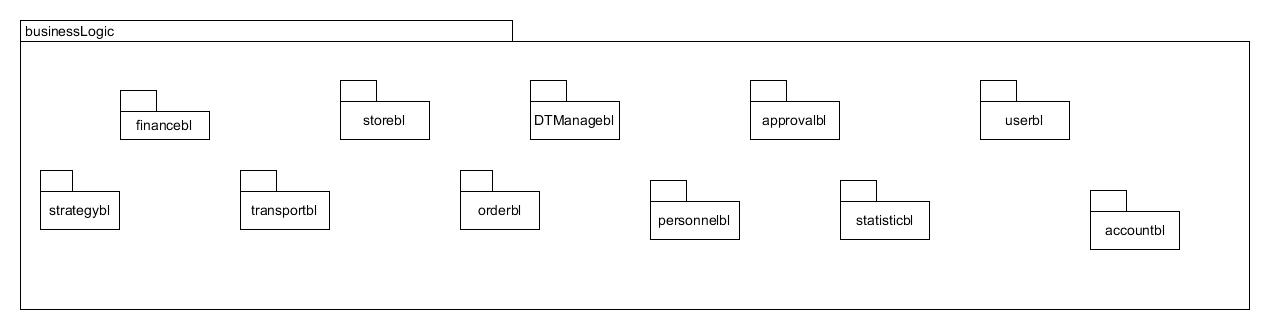


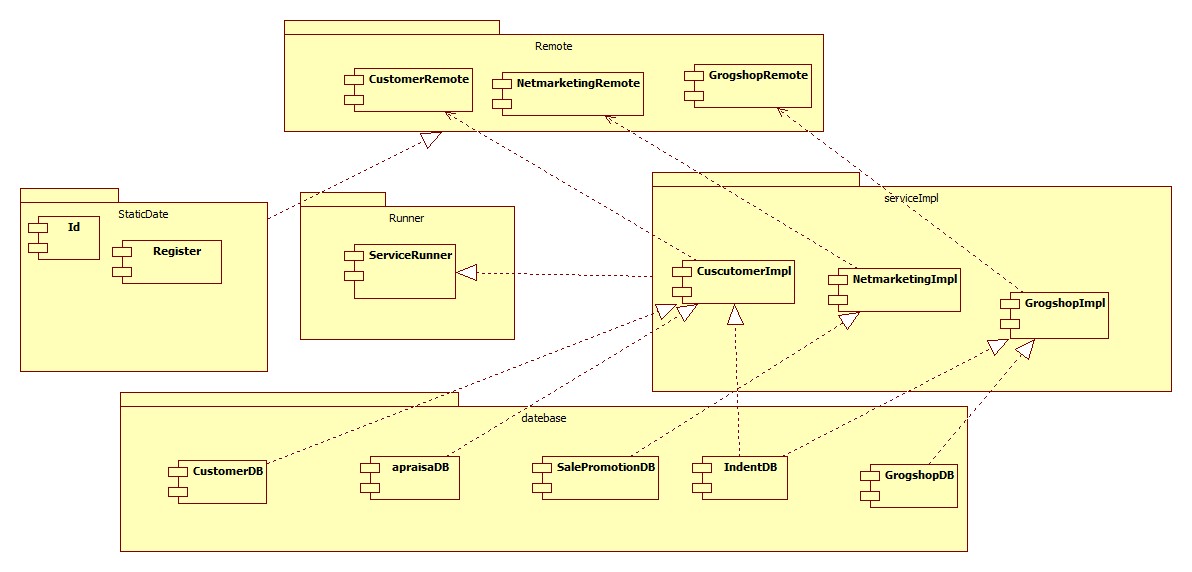
图10 业务逻辑层的设计

* + 1. **业务逻辑层模块的职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| Userbl | 实现登录与个人账户信息修改的服务 |
| Hotelbl | 实现与酒店工作人员有关的账户信息管理的服务 |
| servebl | 实现与网站营销人员有关的逻辑服务 |
| serveConbl  personnelbl | 实现与网站管理人员有关的服务  实现用户的查询与编辑 |
| Orderbl | 实现对信息的编辑与保存 |
| Docbl | 实现对订单的操作 |
| Truebl | 实现对信用值的操作 |

（userbl、hotalbl、servebl、servecontrol归为一类（相似度极高））

**数据层结构细化图：**



### 业务逻辑模块的接口模范

**5.3.2.1 userbl、hotalbl、servebl、servecontrol模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| searchhotelActionListener.  actionPerformed | 语法 | Public void actionPer5.formed (ActionEvent e) |
| 前置条件 | 用户点击“登录”按钮  Ui调用酒店搜寻接口 |
| 后置条件 | Ui调用酒店搜寻接口 |
| placeorderActionListener2.  actionPerformed | 语法 | Public void actionPerformed(ActionEvent e) |
| 前置条件 | 用户点击下订单中的“确定”按钮 |
| 后置条件 | Ui成功调用下订单接口 |
| RegistActionListener2.  actionPerformed | 语法 | Public void actionPerformed(ActionEvent e) |
| 前置条件 | 用户点击“注册”按钮 |
| 后置条件 | Ui成功调用注册接口 |
| LoginActionListener.  actionPerformed | 语法 | Public void actionPerformed(ActionEvent e) |
| 前置条件 | 用户点击“登录”按钮 |
| 后置条件 | Ui成功调用登录接口 |
| informationActionListener.  actionPerformed | 语法 | Public void actionPerformed(ActionEvent e) |
| 前置条件 | 用户点击“基本信息”按钮 |
| 后置条件 | 无 |
| ChangeInformationActionListener.actionPerformed | 语法 | Public void actionPerformed(ActionEvent e) |
| 前置条件 | 用户在编辑信息时点击“确定”按钮 |
| 后置条件 | Ui调用修改基本信息接口 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务内容 |
| Public ArrayList <GrogshopInformation>  searchGrogshop | 根据查询的商圈返回该商圈酒店列表 |
| Public int createIndent | 传入授权码和酒店信息来生成订单 |
| Public String Login(int id,String Password,User\_type type)delete(AccountPO po) | 传入账号和密码以及用户类型，进行登陆判断。  若成功登陆，返回一个授权码  若登陆失败，抛出登陆异常 |
| Public int createNewCustomer  (CustomerInformation susInfor) | 传入一个客户信息数据包来注册，返回注册的账号 |
| Public CustomerInformation  checkInformation(String accredit) | 传入授权码来获取一个客户信息数据包 |
| modifyInformation  (String accredit,  CustomerInformation cuInfor) | 传入授权码和修改后的数据包，返回一个布尔表示修改是否成功 |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| GrogshopRemote. Login | 语法 | public String Login (int id, String password, User\_type type) |
| 前置条件 | 酒店工作人员点击“登录”按钮 |
| 后置条件 | 账号和密码正确，系统返回一个授权码并显示酒店工作人员界面  账号密码错误，系统提示用户名或者密码错误 |
| 需要的服务（需借口） | | |
| 服务名 | 服务 | |
| UserDataService.find | 根据type匹对账号和密码 | |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| GrogshopRemote. CheckInformation | 语法 | public GrogshopInformation GrogshopCheckInformation(String accredit) |
| 前置条件 | 酒店工作人员登录授权后想要查看酒店基本信息 |
| 后置条件 | 系统显示酒店基本信息 |
| 需要的服务（需借口） | | |
| 服务名 | 服务 | |
| HotalDataService.check | 根据酒店查找其基本信息 | |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| GrogshopRemote. checkAllIndent | 语法 | public ArrayList<IndentInformation> checkAllIndent(String accredit,inttype) |
| 前置条件 | 酒店工作人员登录授权后想要查看酒店所有某个类型的订单 |
| 后置条件 | 系统显示酒店所有某个类型的订单 |
| 需要的服务（需借口） | | |
| 服务名 | 服务 | |
| IndentDataService.check | 根据订单类型查找所有某个类型的订单 | |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| GrogshopRemote. offlineCheckIn | 语法 | publicboolean offlineCheckIn(String accredit,Grogshop\_house\_type type,intamount) |
| 前置条件 | 线下直接入住，没有在网上下订单，酒店工作人员输入房间类型，数量 |
| 后置条件 | 系统录入房间类型，数量，并记录当前用户 |
| 需要的服务（需借口） | | |
| 服务名 | 服务 | |

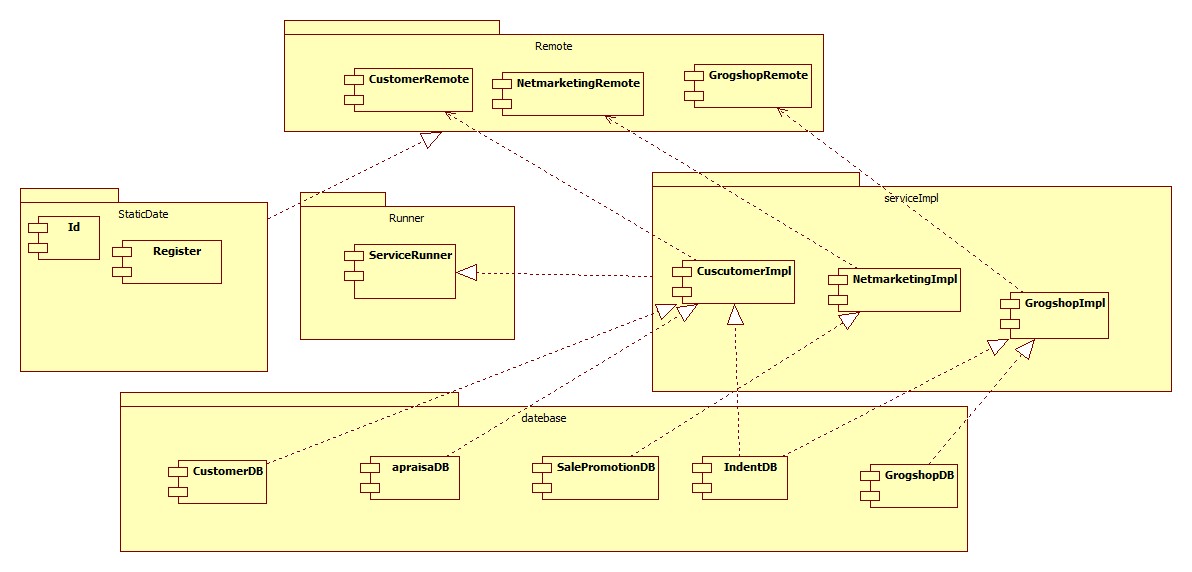
**5.3.2.2 docbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| docbl.getpeopledoc | 语法 | Public ArrayList<PersonVO>  getpeopleByInst(String ID) |
| 前置条件 | 用户选择订单的查看 |
| 后置条件 | 系统返回该用户额订单信息 |
| docbl.getOneDoc | 语法 | Public PersonVO getpeopleByID(String ID) |
| 前置条件 | 用户输入合法 |
| 后置条件 | 返回搜索结果 |
| docbl.getDocBStatic | 语法 | Public ArrayList<PersonVO>  getpeopleByName(String name) |
| 前置条件 | 无 |
| 后置条件 | 返回对应状态的订单 |
| 需要的接口（需接口） | | |
| 服务名 | 服务内容 | |
| docbl.getpeobyInst(String ID) | 根据用户账户返回所有订单 | |
| docbl.getpeobyID(String ID) | 根据ID返回单一订单 | |
| docbl.getDocBStatic (String name) | 根据状态返回订单列表 | |

**5.3.2.3 Truebl模块的借口**

* 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。数据层的实现是通过数据库，数据层中封装了对数据库的操作。



**图11 数据层模块的描述**

* + 1. **数据层模块的职责**

数据层模块职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| customerRemote | 负责处理与登陆有关的数据操作 |
| gropshopRemote | 负责处理与数据查询有关的数据操作 |
| Netmarkingworker | 负责处理营销策略制定有关的操作 |
| StaticData | 负责对数据库的操作 |
| ServiceImpl | 负责数据库的服务 |

### 数据层模块的接口规范

数据层模块的接口规范如下表

**5.4.2.1 CustomRemote的细化**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CustomerRemote.login | 语法 | **public** String Login(**int** id,String password,User\_type type)**throws** RemoteException ,CheckInException |
| 前置条件 | 与服务器建立连接 |
| 后置条件 | 登入到服务器 |
| CustomerRemote.CreateNewCustomer | 语法 | **Public int** createNewCustomer(CustomerInformation cusInfor)**throws** RemoteException ,CheckInException; |
| 前置条件 | 与服务器建立连接 |
| 后置条件 | 创建新账号，并返回 |
| CustomerRemote.checkInformation | 语法 | **Public** CustomerInformation checkInformation(String accredit)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入 |
| 后置条件 | 查看信息 |
| CustomerRemote.modifyInformation | 语法 | **public** **boolean** modifyInformation(String accredit,CustomerInformation cuInfor)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入 |
| 后置条件 | 修改信息，并返回是否成功 |
| CustomerRemote.searchGrogshop | 语法 | **public** ArrayList<GrogshopInformation> searchGrogshop(String accredit,String keyword,String order,**boolean** object)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入 |
| 后置条件 | 得到酒店信息 |
| CustomerRemote.checkAppral | 语法 | **public** ArrayList<Appraisa> checkAppraisal(String accredit,**int** grogshopId)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入 |
| 后置条件 | 评价酒店 |
| CustomerRemote.caculateCost | 语法 | **public** **double** caculateCost(String accredit,IndentInformation indentInfor，boolean vip)**throws** RemoteException,CheckInException |
| 前置条件 | 已经登入并选择订单 |
| 后置条件 | 返回价格 |
| CustomerRemote.createIndent | 语法 | **public** **int** createIndent(String accredit,IndentInformation indentInfor)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入，并确认下订单 |
| 后置条件 | 下订单，并返回订单号，要保存 |
| CustomerRemote.checkCreditLog | 语法 | **public** File checkCreditLog(String accredit)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入 |
| 后置条件 | 返回一个信用记录 |
|  |  |  |
| CustomerRemote.appraiseGrogshop | 语法 | **public** **boolean** appraiseGrogshop(String accredit,Appraisa app)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经下了订单 |
| 后置条件 | 评价酒店完成 |
| CustomerRemote.update | 语法 | **public** Status update(String accredit)**throws** RemoteException,CheckInException; |
| 前置条件 | 与服务器建立连接 |
| 后置条件 | 检查信息更新 |
| CustomerRemote.CreateNewCustomer | 语法 | **Public int** createNewCustomer(CustomerInformation cusInfor)**throws** RemoteException ,CheckInException; |
| 前置条件 | 与服务器建立连接 |
| 后置条件 | 创建新账号，并返回 |
| 需要的服务 | | |
| 服务名 | 服务 | |
| CustomerDB.getCustomerInformationById | 从数据库里读出顾客的所有数据 | |
| CustomerDB.createNewCustomer | 在数据库里插入新顾客数据 | |
| CustomerDB.updateCustomerInformation | 更新数据库里的顾客信息 | |
| Appraisa.getAppraisa | 从数据库里读出酒店的评价 | |
| Appraisa.createNewAppraisa | 在数据库里创建新的评价 | |
| IndentDB.insertNewIndent | 在数据库里插入新的订单 | |

**5.4.2.2** GrogshopRemote接口的细化

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| GrogshopRemote.login | 语法 | **public** String Login(**int** id,String password)**throws** RemoteException,CheckInException; |
| 前置条件 | 与服务器建立连接 |
| 后置条件 | 登入 |
| GrogshopRemote.GrogshopCheckInformation | 语法 | **public** GrogshopInformation GrogshopCheckInformation(String accredit)**throws** RemoteException, SQLException,CheckInException; |
| 前置条件 | 已经登入 |
| 后置条件 | 返回信息 |
| GrogshopRemote.checkAllIndent | 语法 | **public** ArrayList<IndentInformation> checkAllIndent(String accredit,**int** type)**throws** RemoteException; |
| 前置条件 | 已经登入 |
| 后置条件 | 返回所有订单 |
| GrogshopRemote.offlineCheckIn | 语法 | **public** **boolean** offlineCheckIn(String accredit,Grogshop\_house\_type type,**int** amount)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入 |
| 后置条件 | 服务器记录房间少了 |
| GrogshopRemote.offlineCheckOut | 语法 | **public** **boolean** offlineCheckOut(String accredit,Grogshop\_house\_type type,**int** amount)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入 |
| 后置条件 | 服务器更新房间信息 |
| GrogshopRemote.onlineCheckIn | 语法 | **public** **boolean** onlineCheckIn(String accredit,**int** indentId)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入，网上预订顾客入住 |
| 后置条件 | 服务器更细房间信息 |
| GrogshopRemote.onlineCheckout | 语法 | **public** **boolean** onlineCheckOut(String accredit,**int** indentId)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入，网上预订顾客退房 |
| 后置条件 | 服务器更新房间信息 |
| GrogshopRemote.setSalePromotion | 语法 | **public** **boolean** setSalePromotion(String accredit,SalePromotionInformation infor)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入 |
| 后置条件 | 设计新的促销策略 |
| GrogshopRemote.update | 语法 | **public** Status update(String accredit)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入 |
| 后置条件 | 检查信息更新 |
| 需要的服务 | | |
| 服务名 | 服务 | |
| GrogshopDB.getAllGrogshop | 从数据库获得所有符合条件的酒店 | |
| GrogshopDB.getGrogshopById | 依靠酒店id在数据库找到对应的酒店并返回他的信息 | |
| GrogshopDB.insertNewGrogshop | 在数据库里插入新的酒店信息 | |
| SalePromotionDB.UpdateSalePromotion | 在数据库里添加或更新已有的促销策略 | |

**5.4.2.3 NetmarketingWorker 细化**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| NetmarketingWorkerRemote.checkAllIrregularIndent | 语法 | **public** ArrayList<IndentInformation> checkAllIrregularIndent(String accredit,**int** customerId,**int** grogshopId)**throws** RemoteException; |
| 前置条件 | 已经登入 |
| 后置条件 | 检查所有的异常订单 |
| NetmarketingWorkerRemote.removeIrregularIndent | 语法 | **public** **boolean** removeIrregularIndent(String accredit,**int** IndentId)**throws** RemoteException; |
| 前置条件 | 已经登入 |
| 后置条件 | 撤销某个异常订单 |
| NetmarketingWorkerRemote.setSalePromotion | 语法 | **public** **boolean** setSalePromotion(String accredit SalePromotionInformation infor)**throws** RemoteException; |
| 前置条件 | 已经登入 |
| 后置条件 | 制定了新的网站营销策略 |
| NetmarketingWorkerRemote.recharge | 语法 | **public** **boolean** recharge(String accredit,**int** customerId,**int** recoverCredit)**throws** RemoteException; |
| 前置条件 | 已经登入 |
| 后置条件 | 既是信用充值又是信用回复 |
| NetmarketingWorkerRemote. | 语法 | **public** String Login(**int** id,String password)**throws** RemoteException,CheckInException; |
| 前置条件 | 已经登入 |
| 后置条件 | 检查信息更新 |
| NetmarketingWorkerRemote.update | 语法 | **public** Status update(String accredit)**throws** RemoteException; |
| 前置条件 | 已经登入 |
| 后置条件 | 检查信息更新 |
| 需要的服务 | | |
| IndentDB. getIrrugularIndent | 从数据库里获得所有的异常订单 | |
| SalePromotionDB.UpdateSalePromotion | 在数据库里添加或修改促销策略 | |
| IndentDb.updateIndentStatus | 在数据库里修改订单信息 | |

1. **信息视角**
   1. **数据持久化对象**

本系统所有订单用户等数据由数据库承担，无数据持久化对象。

* 1. **文件格式**

本系统依据不同情况分别采用数据库、user文件、hotal文件持久化保存三种方式保存文件