## 5、接口视角

### 5.1 模块的职责

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 |
| UserService  OrderService  HotelService  LocationService  CommCircleService  RoomService  AvailableRoomService  StaffService  CreditRecordService  HotelDiscountService  EnterpriseService  WebDiscountService  OfflineRecordService | 客户端展示层 | 客户端业务逻辑层 |
| UserDAO  OrderDAO  HotelDAO  LocationDAO  CommCircleDAO AvailableRoomDAO  RoomDAO  StaffDAO  CreditRecordDAO  HotelDiscountDAO  EnterpriseDAO  WebDiscountDAO  OfflineRecordDAO | 客户端业务逻辑层 | 服务器端数据层 |

借用用户管理用例来说明层之间的调用，如图5.1-3所示。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）。UserBLService提供了User界面所需要的所有业务逻辑功能UserDataService提供了对数据库的增、删、改、查等操作。这样的实现就大大降低了层与层之间的耦合。

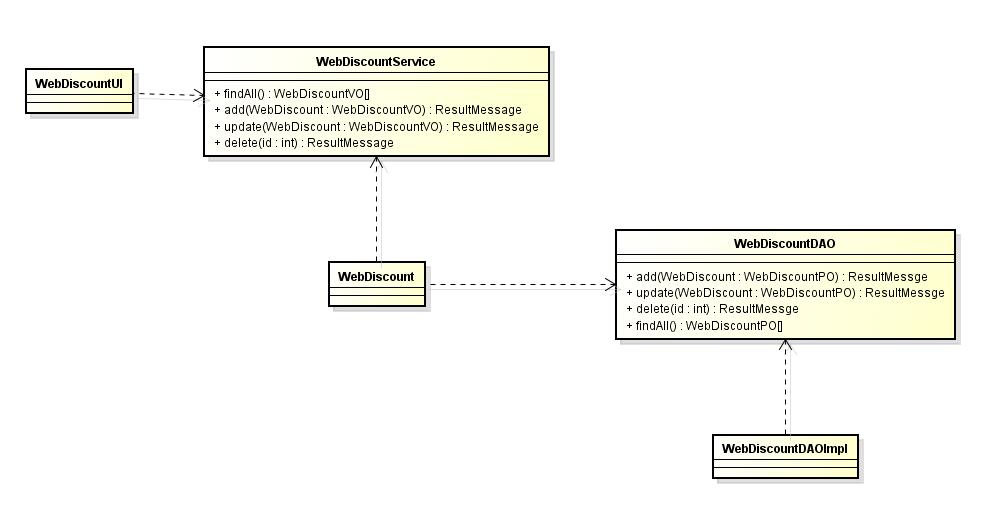


图5.1-3 网站促销策略用例层之间调用的接口

### 5**.2 用户界面层的分解**

根据需求，系统存在28个用户界面：注册登录界面，用户个人中心界面，个人信息界面，订单信息界面，用户详细订单界面，酒店信息界面，信用记录界面，酒店查询界面，酒店信息展示界面，下单界面，酒店管理中心界面，酒店信息维护界面，酒店订单管理界面，酒店详细订单界面，录入客房界面，酒店促销策略界面，线下入住记录界面，网站营销人员界面，网站促销策略界面，异常订单管理界面，信用充值界面，网站信息管理界面，用户信息管理界面，用户信息界面，酒店工作人员界面，网站营销人员界面，酒店添加界面，酒店工作人员添加界面。

界面跳转如图5.2所示。

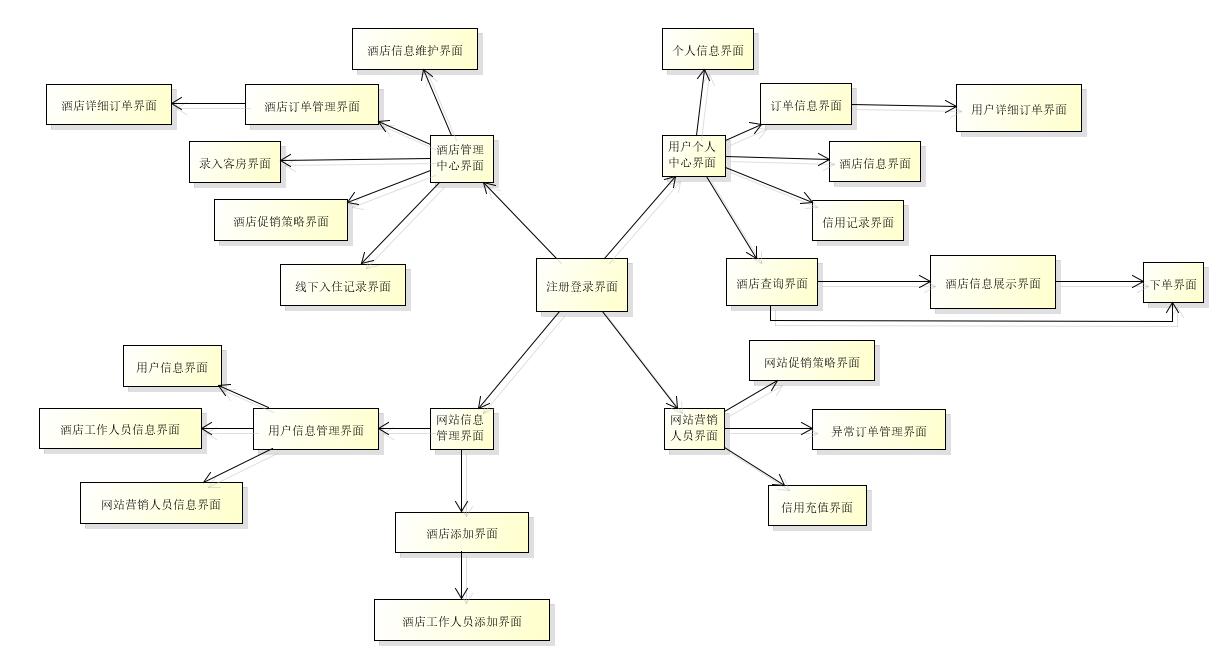


图5.2 用户界面跳转

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。

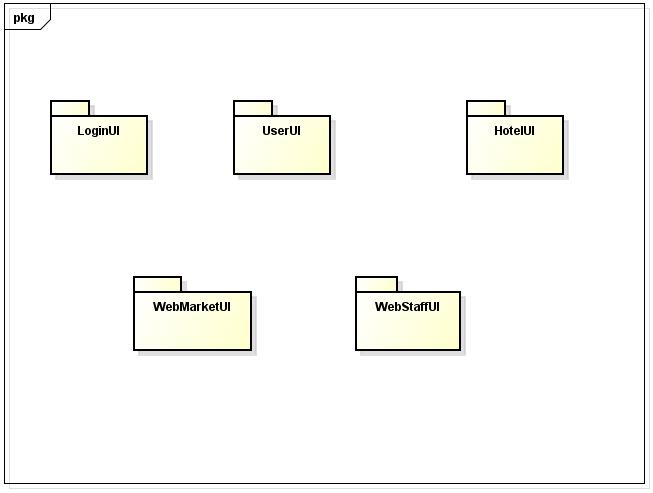


图5.2-2 用户界面类

#### 5.2.1 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 |
| LoginUI | 登录界面包，负责显示登录界面和注册界面 |
| UserUI | 用户界面包，负责实现用户预订酒店和管理自己账号的功能 |
| HotelUI | 酒店工作人员界面包，负责实现酒店管理的功能 |
| WebMarketUI | 网站营销人员界面包，负责实现网站营销的功能 |
| WebStaffUI | 网站管理人员界面包，负责实现网站管理的功能 |

#### 5.2.2用户界面层模块的接口规范

用户界面层模块的接口规范如表5.2.2-1所示。

表5.2.2-1 LoginUI模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| LoginUI.LoginUI.login | 语法 | Public ResultMessage login |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 登录成功后显示登录人员对应的功能界面。 |
| LoginUI.RegisterUI.register | 语法 | Public ResultMessage register |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 注册成功后显示个人信息管理界面 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| LoginController.loginUser(String username, String password) | | 用户进行登录 |
| LoginController.loginStaff(String username, String password) | | 酒店工作人员或网站营销人员进行登录 |
| LoginController.register(UserVO uservo) | | 用户进行注册 |

表5.2.2-2 UserUI模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UserUI.UserUI.getUser | 语法 | Public ResultMessage getUser |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示用户个人信息 |
| UserUI.UserUI.updateUser | 语法 | Public ResultMessage updateUser |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示更新后的用户个人信息 |
| UserUI.UserOrderUI.getOrders | 语法 | Public void getOrders |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示用户订单列表 |
| UserUI.UserOrderUI.remark | 语法 | Public void remark |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示用户订单列表 |
| UserUI.UserOrderUI.revoke | 语法 | Public void revoke |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示用户订单列表 |
| UserUI. OrderDetailUI.getOrder | 语法 | Public void getOrder |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示用户详细订单信息 |
| UserUI.UserHotelUI.getOrderedHotelAndOrder | 语法 | Public void getOrderedHotelAndOrder |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示用户曾预订过的酒店信息 |
| UserUI.CreditRecordUI.getCreditRecords | 语法 | Public void getCreditRecords |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示用户信用记录界面 |
| UserUI.PlaceOrderUI.addOrder | 语法 | Public void addOrder |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示酒店查询界面或酒店信息展示界面 |
| UserUI.HotelSearchUI.getLocations | 语法 | Public void getLocations |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示所有可选城市 |
| UserUI.HotelSearchUI.getCommerCircles | 语法 | Public void getCommerCircles |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示所有可选商圈 |
| UserUI.HotelSearchUI.getHotels | 语法 | Public void getHotels |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示经过筛选后的酒店列表 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| UserController.updateUser(UserVO uservo) | | 更新修改后的用户信息 |
| UserController.findUserByUsername(String username) | | 显示对应用户名的用户信息 |
| UserController.findCreditRecordByUsername(String username) | | 显示用户信用记录 |
| UserOrderController.findOrderByUsernameAndType(String username, OrderStatus type) | | 显示对应用户名和订单类型的订单列表 |
| UserOrderController.placeOrder(OrderVO ordervo) | | 对订单进行处理，为下单做准备 |
| UserOrderController.addOrder(OrderVO ordervo) | | 用户下单 |
| UserOrderController.remark(OrderVO ordervo) | | 用户评价订单 |
| UserOrderController.revoke(OrderVO ordervo) | | 用户撤销订单 |
| UserOrderController.addCreditRecord(CreditRecordVO creditrecordvo) | | 添加信用记录 |
| UserHotelController.findOrderByHotelAndUsername(int hotelID, String username) | | 显示对应酒店id和用户名的订单列表 |
| UserHotelController.getRoomDetail(int hotelID) | | 显示对应酒店的房间列表 |
| UserHotelController.findOrderedHotelAndOrder(String username) | | 显示用户曾预订过的酒店及其订单列表 |
| UserHotelController.findHotels(int loc, int circle, Date begin, Date end) | | 显示满足查询筛选条件的酒店列表 |
| UserHotelController.orderHotels(OrderRule rule, Boolean isDecrease) | | 显示根据排序规则排序后的酒店列表 |
| UserHotelController.findAllLocations() | | 显示可选城市列表 |
| UserHotelController.findCircleByLoc(int locID) | | 显示可选商圈列表 |

表5.2.2-3 HoteUI模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| HotelUI.HotelUI.getHotel | 语法 | Public void getHotel |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示酒店信息 |
| HotelUI.HotelUI.updateHotel | 语法 | Public void updateHotel |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示更新后的酒店信息 |
| HotelUI.HotelOrderUI.checkin | 语法 | Public void checkin |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示更新后的酒店订单详细信息 |
| HotelUI.HotelOrderUI.checkout | 语法 | Public void checkout |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示更新后的酒店订单详细信息 |
| HotelUI.HotelOrderUI.delayCheckin | 语法 | Public void delayCheckin |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示更新后的酒店订单详细信息 |
| HotelUI.RoomUI.getRoom | 语法 | Public void getRoom |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示房间信息 |
| HotelUI.RoomUI.addRoom | 语法 | Public void addRoom |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示房间信息 |
| HotelUI.RoomUI.updateRoom | 语法 | Public void updateRoom |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示更新后的房间信息 |
| HotelUI.HotelDiscountUI.getHotelDiscounts | 语法 | Public void getHotelDiscounts |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示酒店促销策略列表 |
| HotelUI.HotelDiscountUI.addHotelDiscount | 语法 | Public void addHotelDiscount |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示酒店促销策略列表 |
| HotelUI.HotelDiscountUI.updateHotelDiscount | 语法 | Public void updateHotelDiscount |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示酒店促销策略列表 |
| HotelUI.HotelDiscountUI.deleteHotelDiscount | 语法 | Public void deleteHotelDiscount |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示酒店促销策略列表 |
| HotelUI.HotelDiscountUI.getEnterprises | 语法 | Public void getEnterprises |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示可选的企业列表 |
| HotelUI.HotelDiscountUI.getHotelDiscountType | 语法 | Public void getHotelDiscountType |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示可选的酒店促销策略类型 |
| HotelUI.OfflineRecordUI.getOfflineRecord | 语法 | Public void getOfflineRecord |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示酒店线下入住记录 |
| HotelUI.OfflineRecordUI.addOfflineRecord | 语法 | Public void addOfflineRecord |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示酒店线下入住记录 |
| HotelUI.OfflineRecordUI.offlineCheckin | 语法 | Public void offlineCheckin |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示酒店线下入住记录 |
| HotelUI.OfflineRecordUI.offlineCheckout | 语法 | Public void offlineCheckout |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示酒店线下入住记录 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| HotelController.findHotelByID(int HotelID) | | 显示对应id的酒店信息 |
| HotelController.updateHotel(HotelVO hotelvo) | | 更新修改后的酒店信息 |
| HotelOrderController.findOrderByHotelAndType(int HotelID, OrderStatus type) | | 显示对应酒店id和状态的订单列表 |
| HotelOrderController.findOrderByHotelAndUsername(String username, int HotelID) | | 显示对应酒店id和用户名的订单列表 |
| HotelOrderController.findOrderByID(int id) | | 显示对应订单id的订单信息 |
| HotelOrderController.checkout(OrderVo ordervo) | | 根据退房信息修改相应的订单信息 |
| HotelOrderController.revoke(OrderVO ordervo) | | 撤销订单 |
| HotelOrderController.delayCheckin(OrderVO ordervo) | | 修改异常订单状态为已执行 |
| HotelDiscountController.findAllByHotelID(int HotelID) | | 显示酒店促销策略列表 |
| HotelDiscountController.findNotAddedType(int HotelID) | | 显示可选的酒店促销策略类型 |
| HotelDiscountController.getAllEnterprises() | | 显示可选的企业列表 |
| HotelDiscountController.delete(int id) | | 删除酒店促销策略 |
| HotelDiscountController.add(HotelDiscountVO hoteldiscountvo) | | 添加酒店促销策略 |
| HotelDiscountController.update(HotelDiscountVO hoteldiscountvo) | | 更新修改后的酒店促销策略 |

表5.2.2-4 WebMarketUI模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| WebMarketUI.WebDiscountUI.getWebDiscount | 语法 | Public void getWebDiscount |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示网站促销策略列表 |
| WebMarketUI.WebDiscountUI.addWebDiscount | 语法 | Public void addWebDiscount |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示网站促销策略列表 |
| WebMarketUI.WebDiscountUI.updateWebDiscount | 语法 | Public void updateWebDiscount |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示网站促销策略列表 |
| WebMarketUI.WebDiscountUI.deleteWebDiscount | 语法 | Public void deleteWebDiscount |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示网站促销策略列表 |
| WebMarketUI.WebOrderUI.getOrder | 语法 | Public void getOrder |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示网站异常订单列表 |
| WebMarketUI.WebOrderUI.revoke | 语法 | Public void revoke |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示网站异常订单列表 |
| WebMarketUI.WebOrderUI.getUser | 语法 | Public void getUser |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示对应用户名的用户的异常订单列表 |
| WebMarketUI.CreditChargeUI.addCreditRecord | 语法 | Public void addCreditRecord |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示网站信用充值界面 |
| WebMarketUI.CreditChargeUI.getUser | 语法 | Public void getUser |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 在网站信用充值界面显示对应用户名的用户信息 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| WebDiscountController.findAll() | | 显示网站促销策略列表 |
| WebDiscountController.add(WebDiscountVO webdiscountvo) | | 添加网站促销策略 |
| WebDiscountController.update(WebDiscountVO webdiscountvo) | | 更新修改后的网站促销策略 |
| WebDiscountController.delete(int id) | | 删除网站促销策略 |
| WebOrderController.findOrderByOrderType(OrderStatus status) | | 显示网站异常订单列表 |
| WebOrderController.findOrderByID(int ID) | | 显示对应id的订单信息 |
| WebOrderController.revokeOrder(OrderVO ordervo) | | 撤销订单 |
| WebOrderController.findOrderByUsername(String username) | | 显示对应用户名的订单列表 |
| WebCreditController.findUserByUsername(String username) | | 显示对应用户名的用户信息 |
| WebCreditController.addCreditRecord(CreditRecordVO creditrecordvo) | | 增加信用记录 |

表5.2.2-5 WebStaffUI模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| WebStaffUI.WebUserUI.getUser | 语法 | Public void getUser |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示对应用户名的网站用户的信息 |
| WebStaffUI.WebUserUI.UpdateUser | 语法 | Public void updateUser |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示更新后的网站用户的信息 |
| WebStaffUI.WebStaffUI.getStaff | 语法 | Public void getStaff |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示对应的网站酒店工作人员的信息 |
| WebStaffUI.WebStaffUI.UpdateStaff | 语法 | Public void updateStaff |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示更新后的网站酒店工作人员的信息 |
| WebStaffUI.WebMarketUI.getStaff | 语法 | Public void getStaff |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示对应的网站营销人员的信息 |
| WebStaffUI.WebMarketUI.UpdateStaff | 语法 | Public void updateStaff |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示更新后的网站营销人员的信息 |
| WebStaffUI.WebStaffUI.addStaff | 语法 | Public void addStaff |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示新增的网站营销人员的信息 |
| WebStaffUI.HotelAddUI.addHotel | 语法 | Public void addHotel |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示酒店添加界面 |
| WebStaffUI.HotelStaffAddUI.addStaff | 语法 | Public void addStaff |
| 前置条件 | 内部Init()方法调用完成之后 |
| 后置条件 | 显示酒店工作人员添加界面 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| WebUserController.findUserByUsername(String username) | | 显示对应用户名的用户信息 |
| WebUserController.updateUser(UserVO uservo) | | 更新修改后的用户信息 |
| WebStaffController.addHotel(HotelVO hotelvo) | | 添加酒店信息 |
| WebStaffController.addStaff(StaffVO staffvo) | | 添加酒店工作人员信息或网站营销人员信息 |
| WebStaffController.updateStaff(StaffVO staffvo) | | 更新修改后的酒店工作人员信息或网站营销人员信息 |
| WebStaffController.findStaffByUsername(String username) | | 显示对应用户名的酒店工作人员信息或网站营销人员信息 |
| WebStaffController.findStaffByHotelName(String hotelName) | | 显示对应酒店名称的酒店工作人员信息 |

**5.2.3 用户界面模块设计原理**

用户界面利用Java的Swing和AWT库来实现。