网上酒店客房预定系统

详细设计文档

南京大学软件学院

成员：叶晓波、辛志庭、陈步兵、田贵松

日期：2016/10/24

**更新历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **修改日期** | **修改原因** | **版本号** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **引言**
   1. **编制目的**
   2. **词汇表**
   3. **参考资料**
2. **中层设计**
   1. **userinfo\_bl模块的静态结构和动态行为**
      1. userinfo\_bl模块局部模块的职责

userinfo\_bl模块的职责是修改各类用户的个人信息，详见需求规格说明文档，具体的接口规范详见体系结构文档；为了区分不同用户的业务逻辑，并将业务逻辑和逻辑控制分开，我们设计了四种不同用户的controller，负责响应userUI层的不同需求。userPO等PO模块是为了实现持久化数据修改和保存而加入userinfo模块的。userinfo\_bl模块的设计类图如下：

图2.1.1-1 userinfo\_bl 模块的设计类图

* + 1. userinfo\_bl模块局部模块的接口规范

表2.1.2-1 CustomerInfoController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CustomerinfoController. CustomerinfoModify | 语法 | public ResultMessage  CustomerinfoModify  (CustomerVO Customer，  custemorInfoChangeVO VO) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 调用userInfo. CustomerinfoModify |
| CustomerinfoController. CustomerinfoShow | 语法 | public ResultMessage  CustomerinfoShow  (CustomerVO Customer) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 调用userInfo. CustomerinfoShow |
| 服务名 | | 服务 |
| userInfo. CustomerInfoModify | | 修改customerPO单一持久数据 |
| userInfo. CustomerInfoShow | | 返回customerPO单一持久数据 |

表2.1.2-2 hotelStaffInfoController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| hotelStaffInfoController. hotelStaffInfoModify | 语法 | public ResultMessage  hotelStaffInfoModify  (HotelStaffVO hotelStaff，  HotelStaffInfoChangeVO VO) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 调用userInfo. hotelStaffInfoModify |
| HotelStaffinfoController. HotelStaffinfoShow | 语法 | public ResultMessage  HotelStaffinfoShow  (HotelStaffVO HotelStaff) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 调用userInfo. HotelStaffinfoShow |
| 服务名 | | 服务 |
| userInfo. hotelStaffInfoModify | | 修改hotelStaffPO单一持久数据 |
| userInfo. hotelStaffInfoShow | | 返回hotelStaffPO单一持久数据 |

表2.1.2-3 systemStaffInfoController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| systemStaffInfoController. systemStaffInfoModify | 语法 | public ResultMessage  systemStaffInfoModify  (SystemStaffVO systemStaff，  SystemStaffInfoChangeVO VO) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 调用userInfo. systemStaffInfoModify |
| SystemStaffinfoController. SystemStaffinfoShow | 语法 | public ResultMessage  SystemStaffinfoShow  (SystemStaffVO SystemStaff) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 调用userInfo. SystemStaffinfoShow |
| 服务名 | | 服务 |
| userInfo. systemStaffInfoModify | | 修改systemStaffPO单一持久数据 |
| userInfo. systemStaffInfoShow | | 返回systemStaffPO单一持久数据 |

表2.1.2-4 systemManagerInfoController类的接口规范

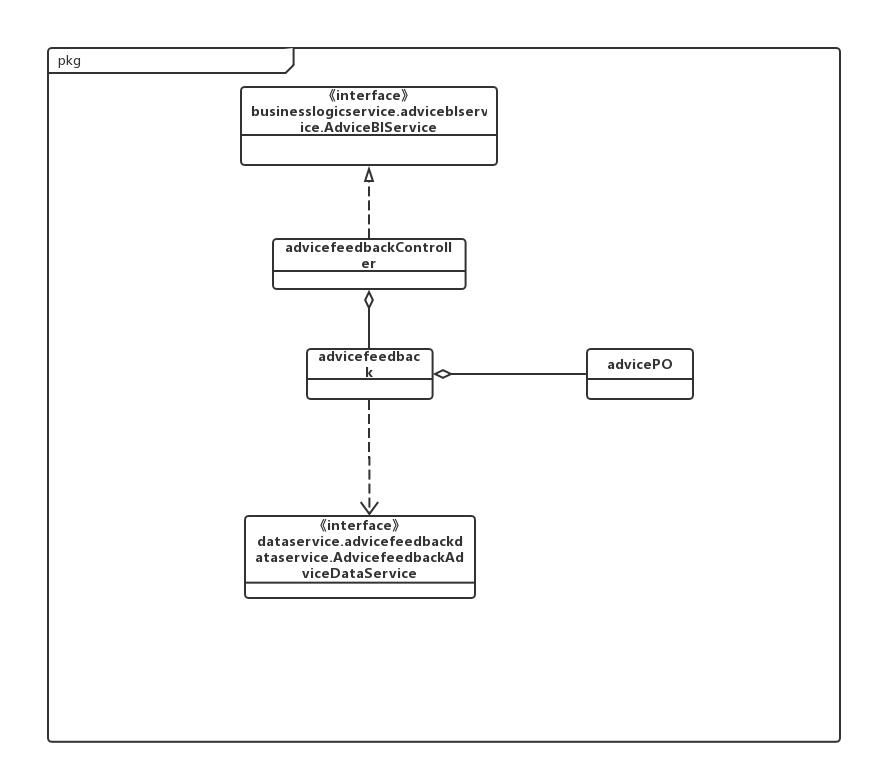
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| systemManagerInfoController. systemManagerInfoModify | 语法 | public ResultMessage  systemManagerInfoModify  (SystemManagerVO systemManager，  SystemManagerInfoChangeVO VO) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 调用  userInfo. systemManagerInfoModify |
| systemManagerInfoController. systemManagerInfoShow | 语法 | public ResultMessage  systemManagerInfoShow  (SystemManagerVO systemManager，  SystemManagerInfoChangeVO VO) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 调用  userInfo. systemManagerInfoShow |
| 服务名 | | 服务 |
| userInfo. systemManagerInfoModify | | 修改systemManagerPO单一持久数据 |
| userInfo. systemManagerInfoShow | | 返回systemManagerPO单一持久数据 |

表2.1.2-5 UserInfo类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CustomerInfo. CustomerInfoModify | 语法 | public ResultMessage  CustomerInfoModify  (CustomerVO Customer，  custemorInfoChangeVO VO) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 修改userPO单一持久数据 |
| CustomerInfo. CustomerInfoShow | 语法 | public ResultMessage  CustomerinfoShow  (CustomerVO Customer) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 显示userPO单一持久数据 |
| hotelStaffInfo. hotelStaffInfoModify | 语法 | public ResultMessage  hotelStaffInfoModify  (HotelStaffVO hotelStaff，  HotelStaffInfoChangeVO VO) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 修改userPO单一持久数据 |
| HotelStaffInfo. HotelStaffinfoShow | 语法 | public ResultMessage  HotelStaffinfoShow  (HotelStaffVO HotelStaff) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 显示userPO单一持久数据 |
| systemStaffInfo. systemStaffInfoModify | 语法 | public ResultMessage  systemStaffInfoModify  (SystemStaffVO systemStaff，  SystemStaffInfoChangeVO VO) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 修改userPO单一持久数据 |
| SystemStaffinfo.SystemStaffinfoShow | 语法 | public ResultMessage  SystemStaffinfoShow  (SystemStaffVO SystemStaff) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 显示userPO单一持久数据 |
| systemManagerInfo. systemManagerInfoModify | 语法 | public ResultMessage  systemManagerInfoModify  (SystemManagerVO systemManager，  SystemManagerInfoChangeVO VO) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 修改userPO单一持久数据 |
| systemManagerInfo. systemManagerInfoShow | 语法 | public ResultMessage  systemManagerInfoShow  (SystemManagerVO systemManager，  SystemManagerInfoChangeVO VO) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 显示userPO单一持久数据 |
| 服务名 | | 服务 |
| userDataService.infoModify | | 修改userPO单一持久数据 |
| userDataService.infoShow | | 显示userPO单一持久数据 |

* + 1. userinfo\_bl模块的行为
  1. **advicefeedback\_bl模块的静态结构和动态行为**
     1. advicefeedback\_bl模块局部模块的职责

图2.2.1-1 transmanage模块类的设计



* + 1. advicefeedback\_bl模块局部模块的接口规范

表2.2.2-1 AdvicefeedbackController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务(供接口) | | | |
| AdvicefeedbackController.show() | 语法 | Public List<AdvicefeedbackVO>  Show() | |
| 前置条件 | 已创建一个advicefeedback领域对象 | |
| 后置条件 | 调用advice feedback领域对象的show()方法 | |
| AdvicefeedbackController.add  (AdvicefeedbackVO advicefeedback) | 语法 | Public result addAdvicefeedback  (AdvicefeedbackVO advicefeedback) | |
| 前置条件 | 已创建一个advicefeedback领域对象 | |
| 后置条件 | 调用advice feedback领域对象的add()方法 | |
| AdvicefeedbackController.getState | 语法 | Public StateVO getStateofAdvice  (AdvicefeedbackVO advicefeedback) | |
| 前置条件 | 已创建一个advicefeedback领域对象,已添加反馈 | |
| 后置条件 | 调用advice feedback领域对象的getState()方法 | |
| AdvicefeedbackController.modify | 语法 | Public result modifyAdvice  (AdvicefeedbackVO advicefeedback) | |
| 前置条件 | 已创建一个advicefeedback领域对象 | |
| 后置条件 | 调用advice feedback领域对象的modify()方法 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| Advicefeedback.show() | | | 得到反馈信息 |
| Advicefeedback.add  (AdvicefeedbackVO advicefeedback) | | | 添加反馈信息 |
| Advicefeedback.getState | | | 得到反馈信息的状态 |
| Advicefeedback.modify | | | 修改反馈信息 |

表2.2.2-2 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 | 发起查看反馈信息 | |
| 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 | 新添反馈信息 | |
| 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 | 新添反馈信息，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 | 已查看反馈信息 | |
| 后置条件 | 更改反馈信息的内容 | |
| 需要的服务(需接口) | | | |
| 服务名 | | | 服务 |
| AdviceFeedbackDataService.modify(AdviceFeedbackPO advicefeedback) | | | 更改单一持久化序列 |
| AdviceFeedbackDataService.add(AdviceFeedbackPO advicefeedback) | | | 新增单一持久化序列 |
| AdviceFeedbackDataService.find(AdviceFeedbackPO advicefeedback) | | | 查找列表持久化序列 |

* + 1. advicefeedback\_bl模块的行为

图2.2.3-1 advicefeedback模块顺序图

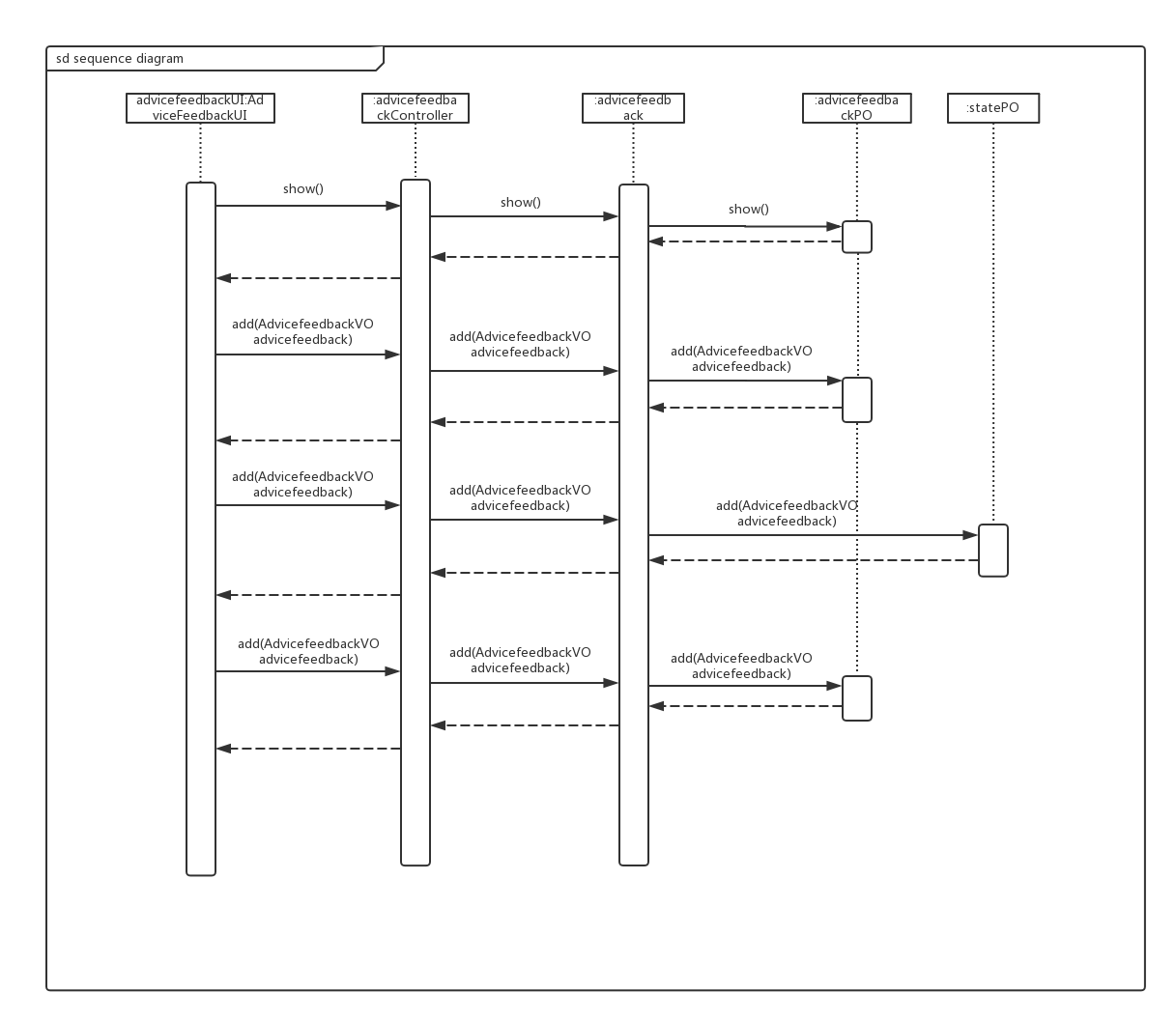
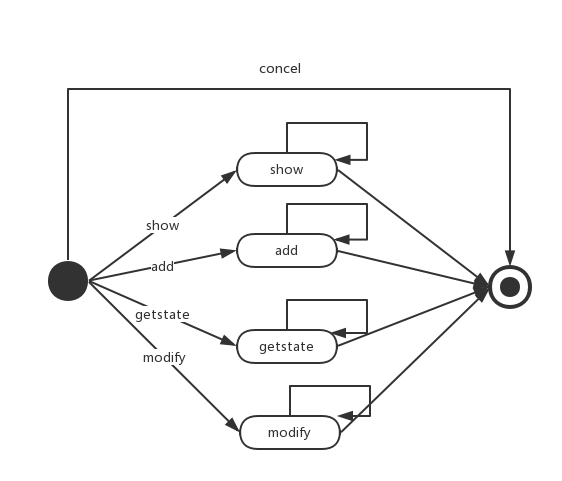


图2.2.3-2 advicefeedback模块状态图

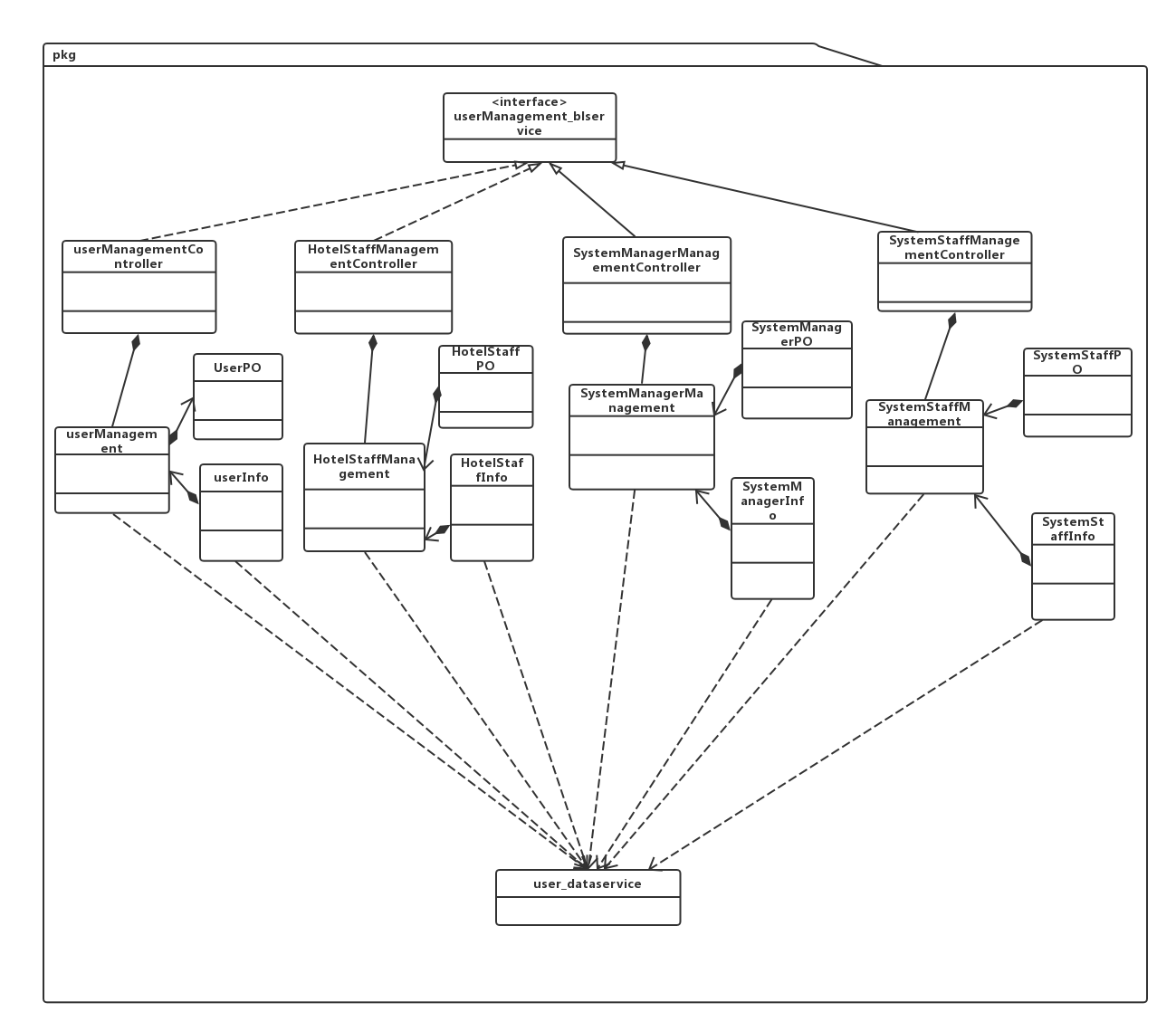


* 1. **usermanagement\_bl模块的静态结构和动态行为**
     1. usermanagement\_bl模块局部模块的职责

UserManagement\_bl模块承担的需求参见需求规格说明文档功能需求以及相关的非功能需求。UserManagement\_bl模块的职责及接口参见软件体系结构描述文档。根据体系结构的设计，采用分层的风格，将系统分为了展示层，业务逻辑层，数据层。每层之间为了灵活性，添加了usermanagement\_blservice和user\_dataservice两个接口。为了隔离业务逻辑层和逻辑控制职责，我们添加了UserManagementCo

ntroller,HotelStaffManagementController,SystemManagerManagementController,SystemStaffManagementController这样四种Controller将会将对于用户管理相关的业务逻辑职责和逻辑控制职责委托给usermanagement\_bl对象。UserPO是作为管理信息的持久化对象被添加到设计模型中的。

UserManagement\_bl模块的设计如图2.3.1-1所示



* + 1. usermanagement\_bl模块局部模块的接口规范

2.3.2-1 UserManagementController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| UserManagementController.addUser | 语法 | | Public ResultMessage addUserManagement  (UserManagementVO user) |
| 前置条件 | | 需要新增一个用户 |
| 后置条件 | | 调用user领域的addUserManagement方法 |
| UserManagementController.getUser | 语法 | | Public UserVO getUser  (String user\_id) |
| 前置条件 | | 已经创建一个user领域，已知user的ID，需要得到用户 |
| 后置条件 | | 调用user领域的getUser方法 |
| UserManagementController.modifyUserManagement | 语法 | | Public ResultMessage modifyUserManagement(UserVOuser，UserChangeVO userChange) |
| 前置条件 | | 已知用户，需要修改用户的信息 |
| 后置条件 | | 调用user领域的modifyUserManagement方法 |
| 服务名 | | 服务 | |
| UserManagement.modify | | 更改用户的信息 | |
| UserManagement.add | | 新增一个用户 | |
| UserManagement.get | | 获得用户的基本信息 | |

2.3.2-2 HotelStaffManagementController模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| HotelStaffManagementController.modifyUser | 语法 | | Public ResultMessage modifyUserManagement  (UserManagementVO user) |
| 前置条件 | | 需要新增一个用户 |
| 后置条件 | | 调用user领域的modifyUserManagement方法 |
| HotelStaffManagementController.getUser | 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 | | 调用user领域的getUser方法 |
| 服务名 | | 服务 | |
| UserManagement.modify | | 更改用户的信息 | |
| UserManagement.get | | 获得用户的基本信息 | |

2.3.2-3 SystemStaffManagementController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| SystemStaffManagementController.modifyUser | 语法 | | Public ResultMessage modifyUserManagement  (UserManagementVO user) |
| 前置条件 | | 需要新增一个用户 |
| 后置条件 | | 调用user领域的modifyUserManagement方法 |
| SystemStaffManagementController.getUser | 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 | | 调用user领域的getUser方法 |
| 服务名 | | 服务 | |
| UserManagement.modify | | 更改用户的信息 | |
| UserManagement.get | | 获得用户的基本信息 | |

2.3.2-4 SystemManagerManagementController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| SystemManagerManagementController.modifyUser | 语法 | | Public ResultMessage modifyUserManagement  (UserManagementVO user) |
| 前置条件 | | 需要新增一个用户 |
| 后置条件 | | 调用user领域的modifyUserManagement方法 |
| SystemManagerManagementController.getUser | 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 | | 调用user领域的getUser方法 |
| SystemManagerManagementController.modifySystemStaff | 语法 | | Public ResultMessage modifySystemStaffManagement  (SystemStaffManagementVO systemstaff,SystemStaffChangeVO systemstaffchange) |
| 前置条件 | | 需要更改一个网站营销人员的信息 |
| 后置条件 | | 调用systemStaff领域的modifySystemStaffManagement方法 |
| SystemManagerManagementController.getSystemStaff | 语法 | | Public SystemStaffVO getSystemStaff  (String systemstaff\_id) |
| 前置条件 | | 已知网站营销人员的ID，获取网站营销人员的信息 |
| 后置条件 | | 调用systemstaff领域的getSystemStaff方法 |
| SystemManagerManagement.addSystemStaff | 语法 | | Public ResultMessage addSystemStaffManagement  (SystemStaffManagementVO systemstaff) |
|  | 前置条件 | | 需要增加一个网站营销人员 |
|  | 后置条件 | | 调用systemStaff领域的addSystemStaff的方法 |
| SystemManagerManagementController.addHotelStaff | 语法 | | Public ResultMessage addHotelStaffManagement  (HotelStaffManagementVO hotelstaff) |
|  | 前置条件 | | 需要增加一个酒店和酒店工作人员 |
|  | 后置条件 | | 调用hotelStaff领域的addHotelStaff方法 |
| 服务名 | | 服务 | |
| UserManagement.modify | | 更改用户的信息 | |
| UserManagement.get | | 获得用户的基本信息 | |
| HotelStaffManagement.add | | 添加酒店和酒店工作人员 | |
| HotelStaffManagement.modify | | 修改酒店工作人员的信息 | |
| HotelStaffManagement.get | | 得到酒店工作人员的信息 | |
| SystemStaffManagement.add | | 添加网站营销人员 | |
| SystemStaffManagement.modify | | 修改网站营销人员的信息 | |
| SystemStaffManagement.get | | 得到网站营销人员的信息 | |

2.3.2-5 UserManagement\_bl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 ResultMessage modifyUserManagement(UserVOuser,UserChangeVO userChange) |
| 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 |
| HotelStaffManagement.modify | 语法 | | Public ResultMessage modifyHotelStaffManagement(HotelStaffVOuser,HotelStaffChangeVO hotelstaffChange) |
| 前置条件 | | 已知酒店工作人员，需要修改酒店工作人员的信息 |
| 后置条件 | | 更改酒店工作人员的信息 |
| SystemStaffManagement.modify | 语法 | | Public ResultMessage modifySystemstaffManagement(SystemStaffVO systemstaff,SystemStaff ChangeVO systemStaffChange) |
| 前置条件 | | 已知网站营销人员，需要修改网站营销人员的信息 |
| 后置条件 | | 更改网站营销人员的信息 |
| HotelStaffManagement.get | 语法 | | Public HotelStaffVO getHotelStaffManagement  (String hotelstaff\_id) |
| 前置条件 | | 已知酒店工作人员的ID，获取酒店工作人员 |
| 后置条件 | | 返回酒店工作人员 |
| SystemStaffManagement.get | 语法 | | Public SystemStaffVO getSystemStaff  (String systemstaff\_id) |
| 前置条件 | | 已知网站营销人员的ID，获取网站营销人员 |
| 后置条件 | | 返回网站营销人员 |
| HotelStaffManagement.add | 语法 | | Public ResultMessage addHotelStaffManagement  (HotelStaffManagementVO hotelstaff) |
|  | 前置条件 | | 需要增加一个酒店和酒店工作人员 |
|  | 后置条件 | | 新增酒店和酒店工作人员 |
| SystemStaffManagement.add | 语法 | | Public ResultMessage addSystemStaffManagement  (SystemStaffManagementVO systemstaff) |
|  | 前置条件 | | 需要增加一个网站营销人员 |
|  | 后置条件 | | 新增网站营销人员 |
| 服务名 | | 服务 | |
| User\_data.modify(userPO user,UserChangePO userChange) | | 更改单一持久序列 | |
| User\_data.find(String userID) | | 查找单一持久化序列 | |
| User\_data.modify(HotelStaffPO hotelstaff,HotelStaffChangePO hotelstaffChange) | | 更改单一持久序列 | |
| User\_data.add(HotelStaffPO hotelstaff) | | 新增大一持久化序列 | |
| User\_data.find(String hotelstaffID) | | 查找单一持久化序列 | |
| User\_data.find(String systemstaffID) | | 查找单一持久化序列 | |
| User\_data.modify(SystemStaffPO systemstaff,SystemStaffChangePO systemstaffChange) | | 更改单一持久序列 | |
| User\_data.add(SystemStaffPO systemstaff) | | 新增大一持久化序列 | |

* + 1. usermanagement\_bl模块的行为
  1. **regesiter\_bl模块的静态结构和动态行为**
     1. regesiter\_bl模块局部模块的职责

registerbl模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

registerbl模块的职责及接口参见软件体系结构描述文档。根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了registerblservice, registerdataservice 两个接口。为了隔离业务逻辑职责和逻辑控制职责，我们添加了registerController，这样registerController 将会将用户管理相关的业务逻辑职责和逻辑控制委托给registerbl 对象。userPO是做为管理信息的持久化对象被添加到设计模型中的。

orderbl模块的设计如图2.3.1-1所示.

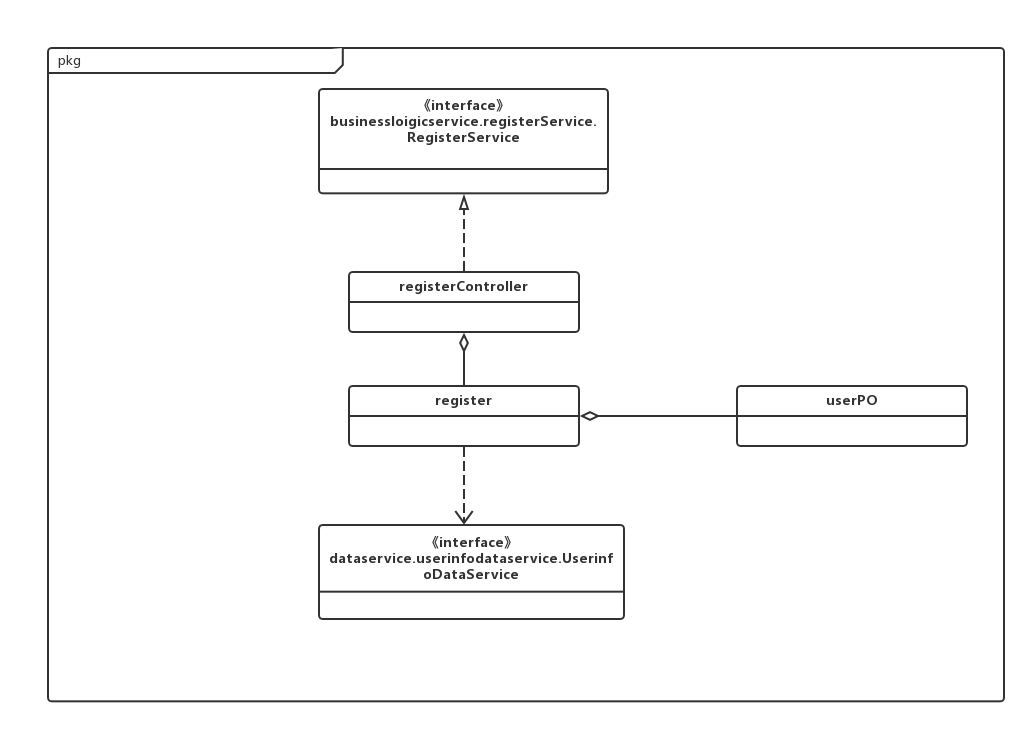


图2.4.1-1 register\_bl模块的各个类的设计

* + 1. regesiter\_bl模块局部模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| RegisterController.add | 语法 | | Public result addRegister(String userID,String userpassword) |
| 前置条件 | | 已创建一个register领域对象，已输入用户ID和密码 |
| 后置条件 | | 调用register领域对象的add方法 |
| 服务名 | | 服务 | |
| register.add(String userID,String userpassword) | | 新增一个持久化序列 | |

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Register.add | 语法 | | Public result addRegister(String userID,String userpassword) |
| 前置条件 | | 已输入用户ID和密码 |
| 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 | |

* + 1. regesiter\_bl模块的行为

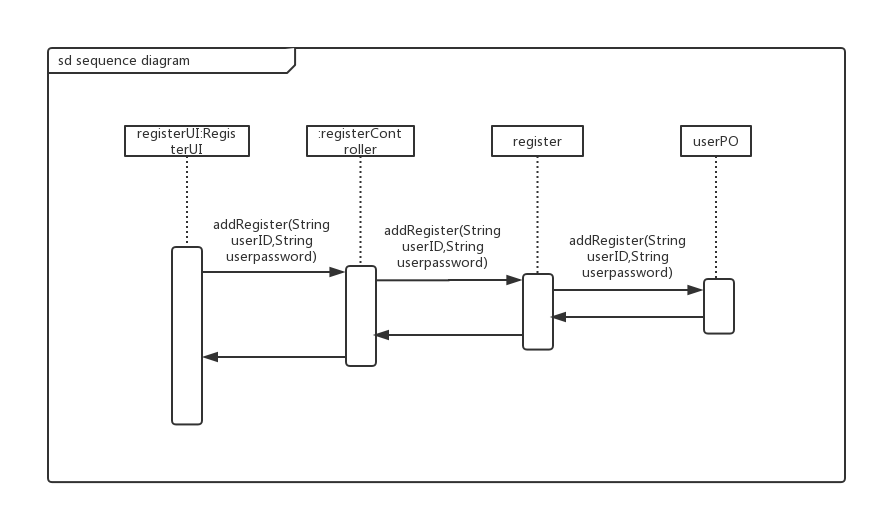


图2.4.3-1 registerBl顺序图

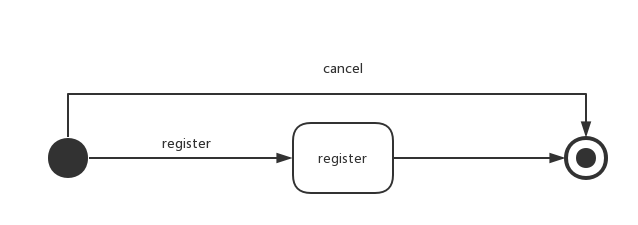


图2.4.3-2register的状态图

* 1. **login\_bl模块的静态结构和动态行为**
     1. login\_bl模块局部模块的职责

login\_bl处理用户的登录操作，若登录失败返回失败原因；若登陆成功，则加载用户的ID、类型作为客户端加载的常量

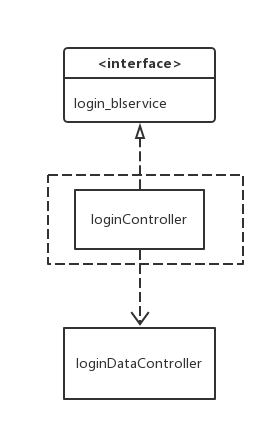


图3.2.1-1 login\_bl模块的各个类的设计

* + 1. login\_bl模块局部模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| loginController.comfirm | 语法 | public ResultMessage comfirm (String user\_id, String user\_password) | |
|  | 启动登录服务，已知用户ID和用户密码 | |
|  | 验证用户ID和密码，返回登录信息（登录是否成功、登录用户类型） | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| loginDataController.confirm | | | 调用数据层方法验证用户信息 |

* + 1. login\_bl模块的行为

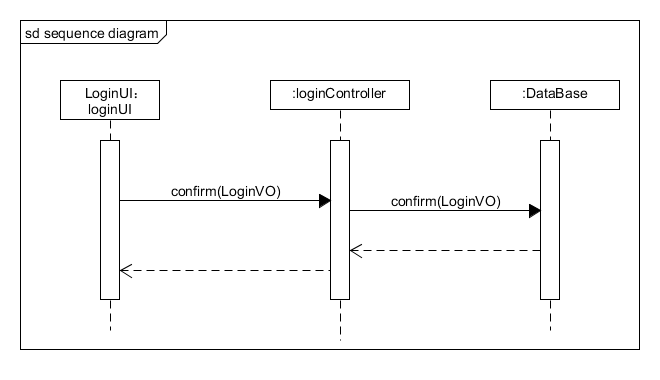


图3.2.3-1 loginBl顺序图

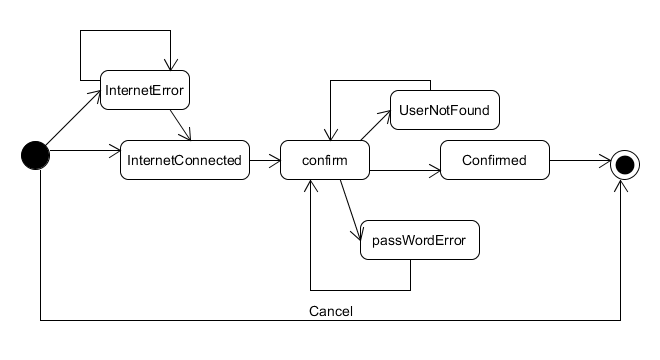


图2.10.2-1 login模块设计

* 1. **order\_bl模块的静态结构和动态行为**
     1. order\_bl模块局部模块的职责

orderbl模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

orderbl模块的职责及接口参见软件体系结构描述文档。根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了orderblservice, orderdataservice 两个接口。为了隔离业务逻辑职责和逻辑控制职责，我们添加了OrderController，这样OrderController 将会将用户管理相关的业务逻辑职责和逻辑控制委托给orderbl 对象。orderPO是做为管理信息的持久化对象被添加到设计模型中的。orderbl模块的设计如图2.7.1-1所示。

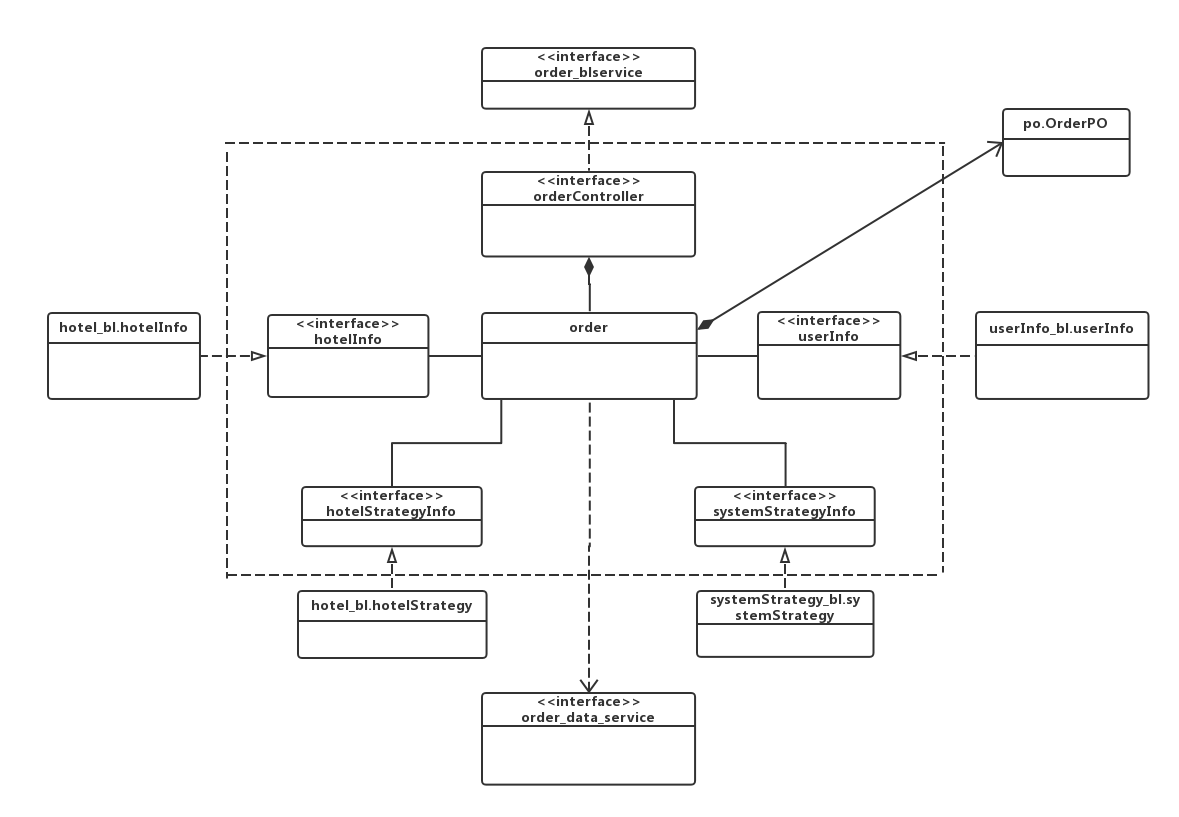


图2.7.1-1 orderbl模块的各个类设计

* + 1. order\_bl模块局部模块的接口规范

orderController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderController.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 |
| OrderController.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 |
| OrderController.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 |
| OrderController.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 |
| OrderController.saveOrder | 语法 | public ResultMessage 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 |
| OrderController.changeState | 语法 | public ResultMessage 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 |
| OrderController.getState | | 获得一个订单的状态 |
| OrderController.getOrder | | 根据订单编号获得一个订单的详细信息 |
| OrderController.getOrdersofUsers | | 根据用户信息获得该用户的订单信息 |
| OrderController.createOrder | | 创建订单 |
| OrderController.saveOrder | | 保存订单 |
| OrderController.changeState | | 更改订单状态 |

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 ResultMessage 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 ResultMessage changeState  (Order order\_info，StateVO state\_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 |

* + 1. order\_bl模块的行为

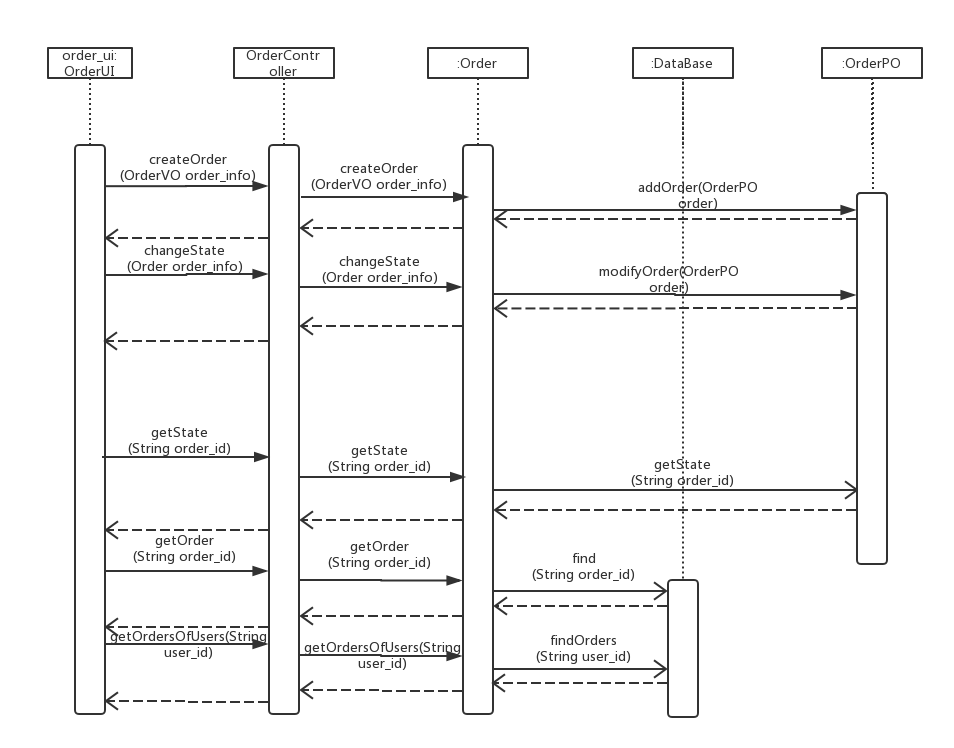


图2.7.3-1订单生成、保存、查找、获取、修改的顺序图

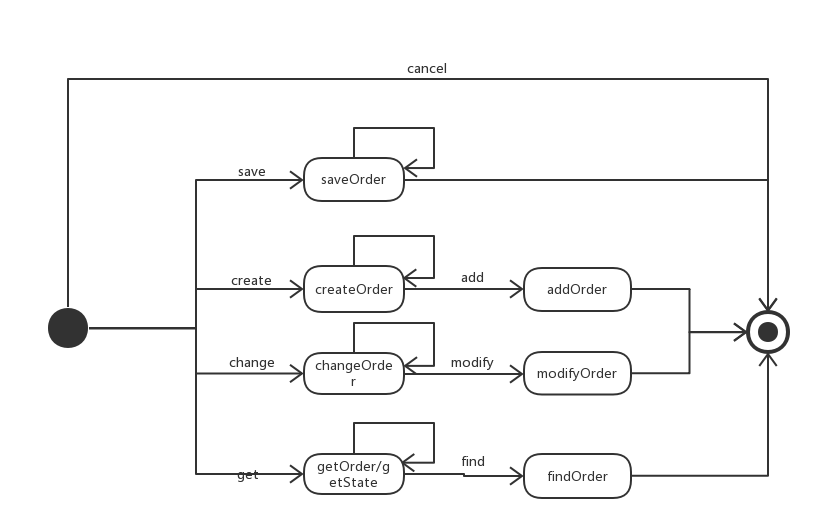


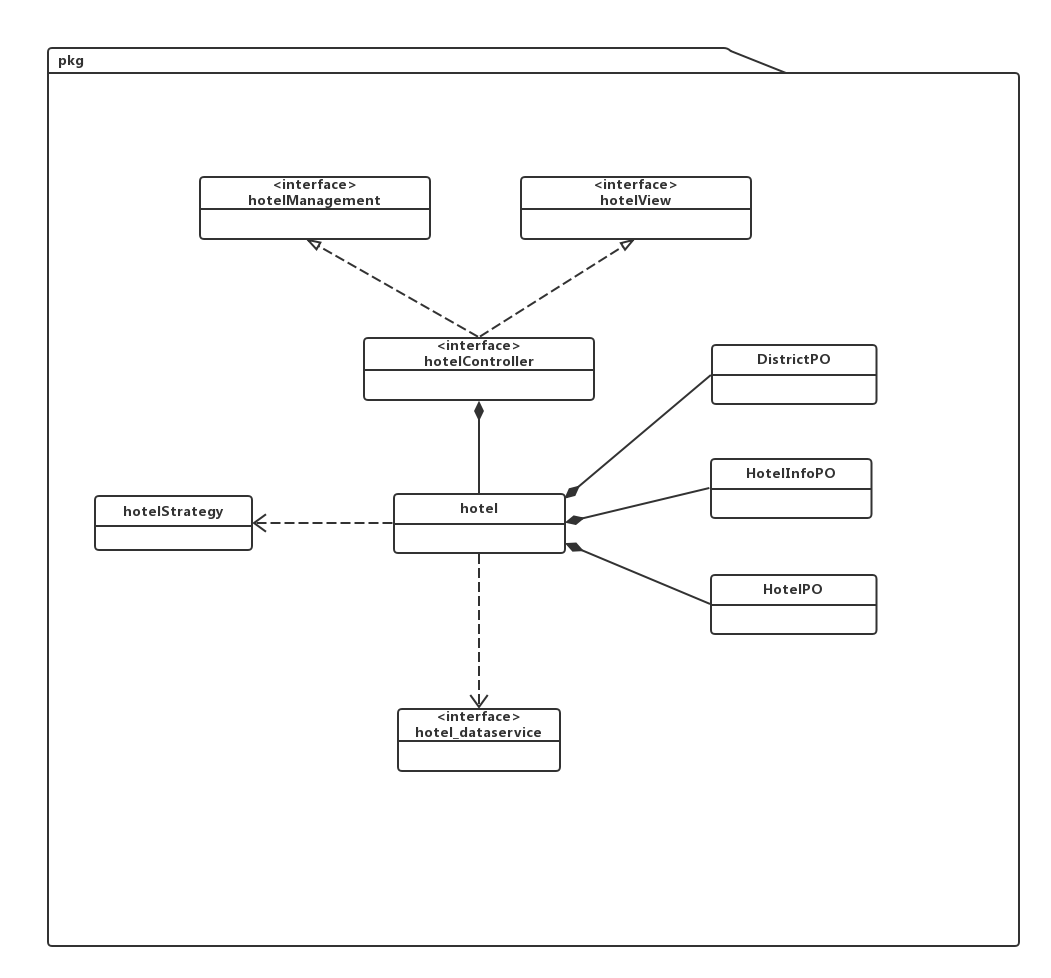
图2.7.3-2出库单生成、保存、查找、获取、修改的状态图

* 1. **hotel\_bl模块的静态结构和动态行为**
     1. hotel\_bl模块局部模块的职责

hotel\_bl模块的职责详见需求规格说明文档，接口的定义详见体系结构文档。采用了委托式的系统控制风格，我们新增了hotelController类，实现业务逻辑和逻辑控制的分离；为了改变依赖关系，我们设置了hotel\_data模块建立与data层持久化数据的交互。HotelPO，HotelInfoPO,DistrictPO是为了实现数据持久化所添加到设计模型之中的。

以下是hotel\_bl的设计类图：

图2.8.1-1 hotel\_bl的设计类图



* + 1. hotel\_bl模块局部模块的接口规范

表2.8.2-1 hotelController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| HotelController.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 |
| HotelController.modifyHotelRoom | 语法 | public ResultMessage  modifyHotelRoom (HotelVO hotel,RoomOperateVO roomOperate) |
| 前置条件 | 已知酒店的信息和对房间的更改操作，需要对于酒店的房间信息进行更改 |
| 后置条件 | 更改酒店的房间信息 |
| HotelController.modifyHotelInfo | 语法 | public ResultMessage modifyHotelInfo(HotelVO hotel,HotelInfoVO hoteInfo) |
| 前置条件 | 已知酒店和对酒店信息的更改方案，需要对酒店的描述信息进行更改 |
| 后置条件 | 更改酒店的信息 |
| HotelController.saveHotel | 语法 | public ResultMessage 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 |
| HotelController.createHotel | 语法 | public HotelVO addHotel  (HotelInfoVO hotelInfo) |
| 前置条件 | 已知酒店的信息，需要生成一个新的酒店 |
| 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VO类 |
| HotelController.deleteHotel | 语法 | public ResultMessage 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 |
| HotelController.showDistrictHotel | 语法 | public List<HotelVO> showDistrictHotel (String districtName) |
| 前置条件 | 已知商圈名称，需要返回该商圈所有酒店的列表 |
| 后置条件 | 返回该商圈所有酒店的列表 |
| HotelController.showHotel | 语法 | public List<HotelVO> showHotel() |
| 前置条件 | 需要得到所有酒店的列表 |
| 后置条件 | 返回所有酒店的列表 |
| 服务名 | | 服务 |
| Hotel.getHotel | | 获得酒店 |
| Hotel.modifyHotelRoom | | 修改酒店房间信息 |
| Hotel.modifyHotelInfo | | 修改酒店描述信息 |
| Hotel.saveHotel | | 持久化保存酒店 |
| Hotel.createHotel | | 生成酒店VO类 |
| Hotel.deleteHotel | | 删除酒店持久化数据 |
| Hotel.showDistrictHotel | | 获得对应商圈酒店列表 |
| Hotel.showHotel | | 获得所有酒店列表 |

表2.8.2-2 hotel类的接口规范

|  |  |  |
| --- | --- | --- |
| 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.modifyHotelRoom | 语法 | public ResultMessage  modifyHotelRoom(HotelVO hotel,RoomOperateVO roomOperate) |
| 前置条件 | 已知酒店的信息和对房间的更改操作，需要对于酒店的房间信息进行更改 |
| 后置条件 | 更改酒店的房间信息 |
| Hotel.modifyHotelInfo | 语法 | public ResultMessage  modifyHotelInfo(HotelVO hotel,HotelInfoVO hoteInfo) |
| 前置条件 | 已知酒店和对酒店信息的更改方案，需要对酒店的描述信息进行更改 |
| 后置条件 | 更改酒店的信息 |
| Hotel.saveHotel | 语法 | public ResultMessage 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.createHotel | 语法 | public HotelVO createHotel  (HotelInfoVO hotelInfo) |
| 前置条件 | 已知酒店的信息，需要生成一个新的酒店 |
| 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VO类 |
| Hotel.deleteHotel | 语法 | public ResultMessage 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.showDistrictHotel | 语法 | public List<HotelVO>  showDistrictHotel  (String districtName) |
| 前置条件 | 已知商圈名称，需要返回该商圈所有酒店的列表 |
| 后置条件 | 返回该商圈所有酒店的列表 |
| Hotel.showHotel | 语法 | public List<HotelVO> showHotel() |
| 前置条件 | 需要得到所有酒店的列表 |
| 后置条件 | 返回所有酒店的列表 |
| 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 |
| HotelStrategy. getHotelStrategy(HotelPO hotel) | | 返回酒店促销策略列表 |

* + 1. hotel\_bl模块的行为

图2.8.3-1 hotel生成、保存、删除操作的顺序图

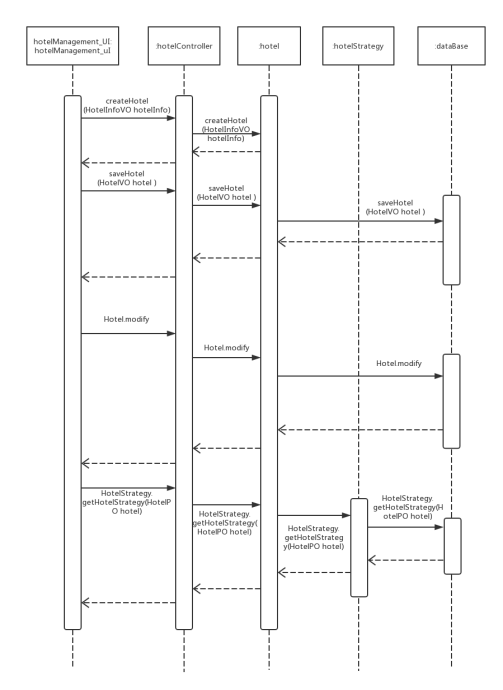
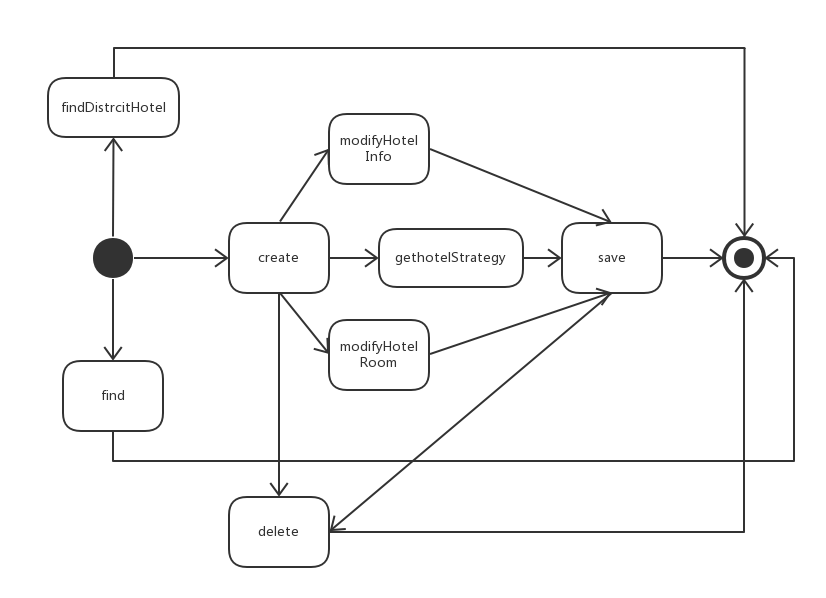


图2.8.3-2 hotel类的状态图



* 1. **hotelstrategy\_bl模块的静态结构和动态行为**
     1. hotelStrategy\_bl模块局部模块的职责

hotelStrategy\_bl模块承担的需求参见需求规格说明文档功能需求以及相关非功能需求。hotelStrategy\_bl模块的职责以及接口参见软件体系结构描述文档。根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了hotelStrategy\_blservice和hotelStrategy\_dataservice两个接口。为了隔离逻辑职责和逻辑控制职责，我们添加了hotelStrategyController，这样hotelStrategyController将会将酒店工作人员需要的逻辑职责和逻辑控制职责委托给hotelStrategy\_bl对象。hotelStrategyPO是作为酒店促销策略的持久化对象添加到设计模型中的。

hotelStrategybl模块的设计如图2.9.1-1所示

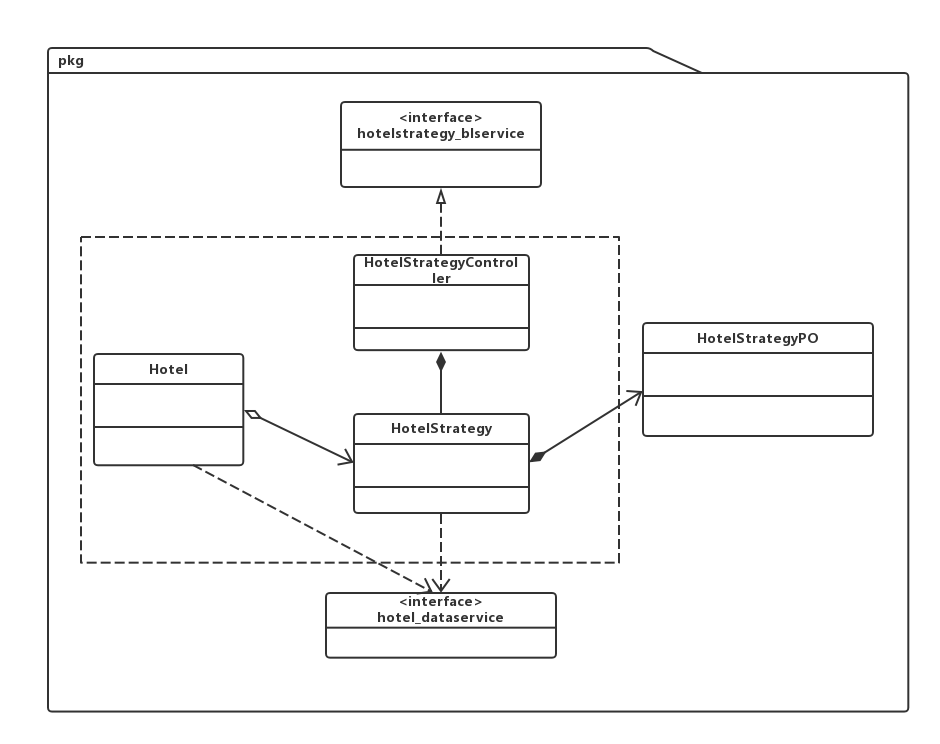


图2.9.1-1 hotelStrategy\_bl模块的各个类设计

* + 1. hotelStrategy\_bl模块局部模块的接口规范

hotelStrategyController类的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| HotelStrategyController. makeHotelStrategy | 语法 | | Public ResultMessage 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 | | 调用HotelStrategy领域对象的makeHotelStrategy方法 |
| HotelStrategyController.modifyHotelStrategy | 语法 | | Public ResultMessage 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 | | 调用HotelStrategy领域对象的modifyHotelStrategy方法 |
| HotelStrategyController.deleteHotelStrategy | 语法 | | Public ResultMessage 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 | | 调用HotelStrategy领域对象的deleteHotelStrategy方法 |
| HotelStrategyController.getHotelStrategy | 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 | | 调用HotelStrategy领域对象的getHotelStrategy方法 |
| HotelStrategyController.getListOfHotelStrategys | 语法 | | Public List<HotelStrategyVO>  getListOfHotelStrategys(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 | | 调用HotelStrategy领域对象的getListOfHotelStrategys方法 |
| 服务名 | | 服务 | |
| HotelStrategy.make | | 制定酒店促销策略 | |
| HotelStrategy.modify | | 修改酒店促销策略 | |
| HotelStrategy.delete | | 删除酒店促销策略 | |
| HotelStrategy.save | | 保存已经制定或者更改的酒店促销策略 | |
| HotelStrategy.get | | 获得某个酒店促销策略 | |
| HotelStrategy.getList | | 获得酒店的所有促销策略 | |

HotelStrategy\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| HotelStrategy.make | 语法 | | Public ResultMessage 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 ResultMessage 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 ResultMessage 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.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.getListOfHotelStrategys | 语法 | | Public List<HotelStrategyVO>  getListOfHotelStrategys(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.findListOfstrategy (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 | |

* + 1. hotelStrategy\_bl模块的行为

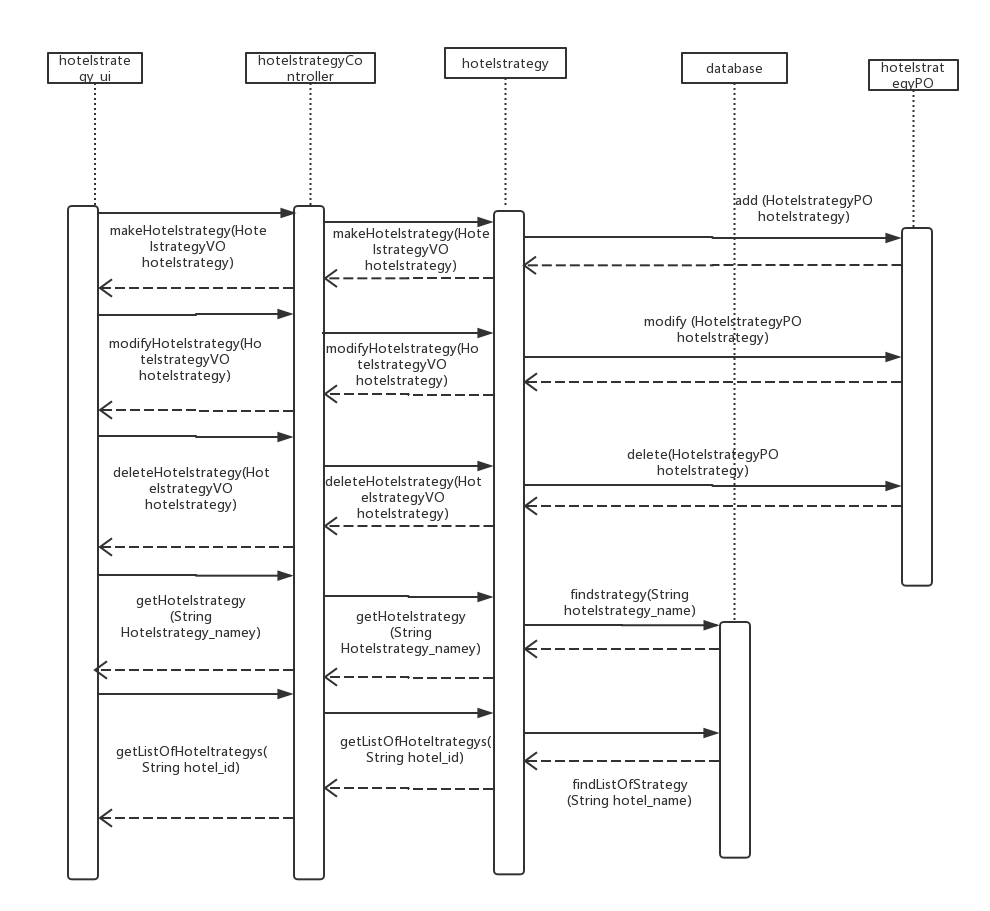
****

图2.9.3-1 对于酒店促销策略的制定、修改和删除的顺序图

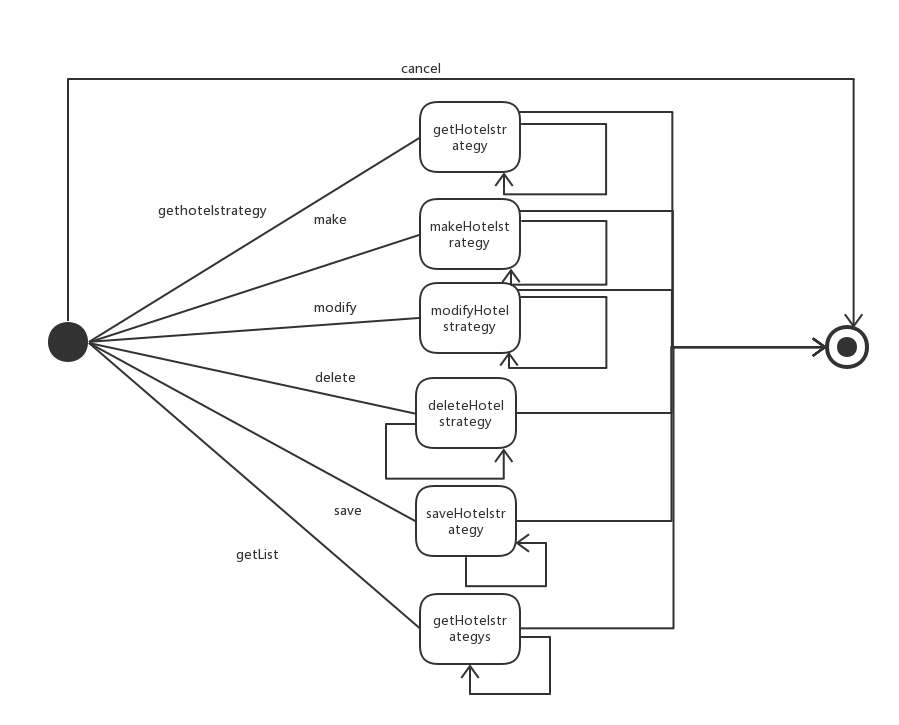


图2.9.3-2酒店促销策略对象的状态图

* 1. **systemstrategy\_bl模块的静态结构和动态行为**
     1. systemstrategy\_bl模块局部模块的职责
     2. systemstrategy\_bl模块局部模块的接口规范
     3. systemstrategy\_bl模块的行为
  2. **用户界面层的行为**
     1. 用户界面层的跳转
     2. 用户界面层的跳转顺序图示例