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** |
| 全员 | 2016/10/29 | 初稿，2.10人机交互部分未加入 | v2.0 |
| 叶晓波 | 2016/11/2 | 修改了hotel和hotelStrategy模块的controller的定义和方法 | v2.1 |
|  |  |  |  |

目录

[**1.** **引言** 6](#_Toc465519663)

[**1.1.** **编制目的** 6](#_Toc465519664)

[**1.2.** **词汇表** 6](#_Toc465519665)

[**1.3.** **参考资料** 6](#_Toc465519666)

[**2.** **中层设计** 6](#_Toc465519667)

[**2.1.** **userinfo\_bl模块的静态结构和动态行为** 6](#_Toc465519668)

[2.1.1. userinfo\_bl模块局部模块的职责 6](#_Toc465519669)

[2.1.2. userinfo\_bl模块局部模块的接口规范 7](#_Toc465519670)

[2.1.3. userinfo\_bl模块的行为 13](#_Toc465519671)

[**2.2.** **advicefeedback\_bl模块的静态结构和动态行为** 13](#_Toc465519672)

[2.2.1. advicefeedback\_bl模块局部模块的职责 13](#_Toc465519673)

[2.2.2. advicefeedback\_bl模块局部模块的接口规范 14](#_Toc465519674)

[2.2.3. advicefeedback\_bl模块的行为 17](#_Toc465519675)

[**2.3.** **usermanagement\_bl模块的静态结构和动态行为** 18](#_Toc465519676)

[2.3.1. usermanagement\_bl模块局部模块的职责 18](#_Toc465519677)

[2.3.2. usermanagement\_bl模块局部模块的接口规范 19](#_Toc465519678)

[2.3.3. usermanagement\_bl模块的行为 28](#_Toc465519679)

[**2.4.** **regesiter\_bl模块的静态结构和动态行为** 34](#_Toc465519680)

[2.4.1. regesiter\_bl模块局部模块的职责 34](#_Toc465519681)

[2.4.2. regesiter\_bl模块局部模块的接口规范 35](#_Toc465519682)

[2.4.3. regesiter\_bl模块的行为 36](#_Toc465519683)

[**2.5.** **login\_bl模块的静态结构和动态行为** 37](#_Toc465519684)

[2.5.1. login\_bl模块局部模块的职责 37](#_Toc465519685)

[2.5.2. login\_bl模块局部模块的接口规范 38](#_Toc465519686)

[2.5.3. login\_bl模块的行为 38](#_Toc465519687)

[**2.6.** **order\_bl模块的静态结构和动态行为** 39](#_Toc465519688)

[2.6.1. order\_bl模块局部模块的职责 39](#_Toc465519689)

[2.6.2. order\_bl模块局部模块的接口规范 40](#_Toc465519690)

[2.6.3. order\_bl模块的行为 44](#_Toc465519691)

[**2.7.** **hotel\_bl模块的静态结构和动态行为** 45](#_Toc465519692)

[2.7.1. hotel\_bl模块局部模块的职责 45](#_Toc465519693)

[2.7.2. hotel\_bl模块局部模块的接口规范 46](#_Toc465519694)

[2.7.3. hotel\_bl模块的行为 51](#_Toc465519695)

[**2.8.** **hotelstrategy\_bl模块的静态结构和动态行为** 53](#_Toc465519696)

[2.8.1. hotelStrategy\_bl模块局部模块的职责 53](#_Toc465519697)

[2.8.2. hotelStrategy\_bl模块局部模块的接口规范 54](#_Toc465519698)

[2.8.3. hotelStrategy\_bl模块的行为 58](#_Toc465519699)

[**2.9.** **systemstrategy\_bl模块的静态结构和动态行为** 60](#_Toc465519700)

[2.9.1. systemstrategy\_bl模块局部模块的职责 60](#_Toc465519701)

[2.9.2. systemstrategy\_bl模块局部模块的接口规范 61](#_Toc465519702)

[2.9.3. systemstrategy\_bl模块的行为 66](#_Toc465519703)

[**2.10.** **用户界面层的行为** 67](#_Toc465519704)

[2.10.1. 用户界面层的跳转 67](#_Toc465519705)

[2.10.2. 用户界面层的跳转顺序图示例 67](#_Toc465519706)

1. **引言**
   1. **编制目的**

本报告详细完成对网上酒店客房预订系统的详细设计，达到指导后续软件构造的目的，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. **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 |  |

* 1. **参考资料**

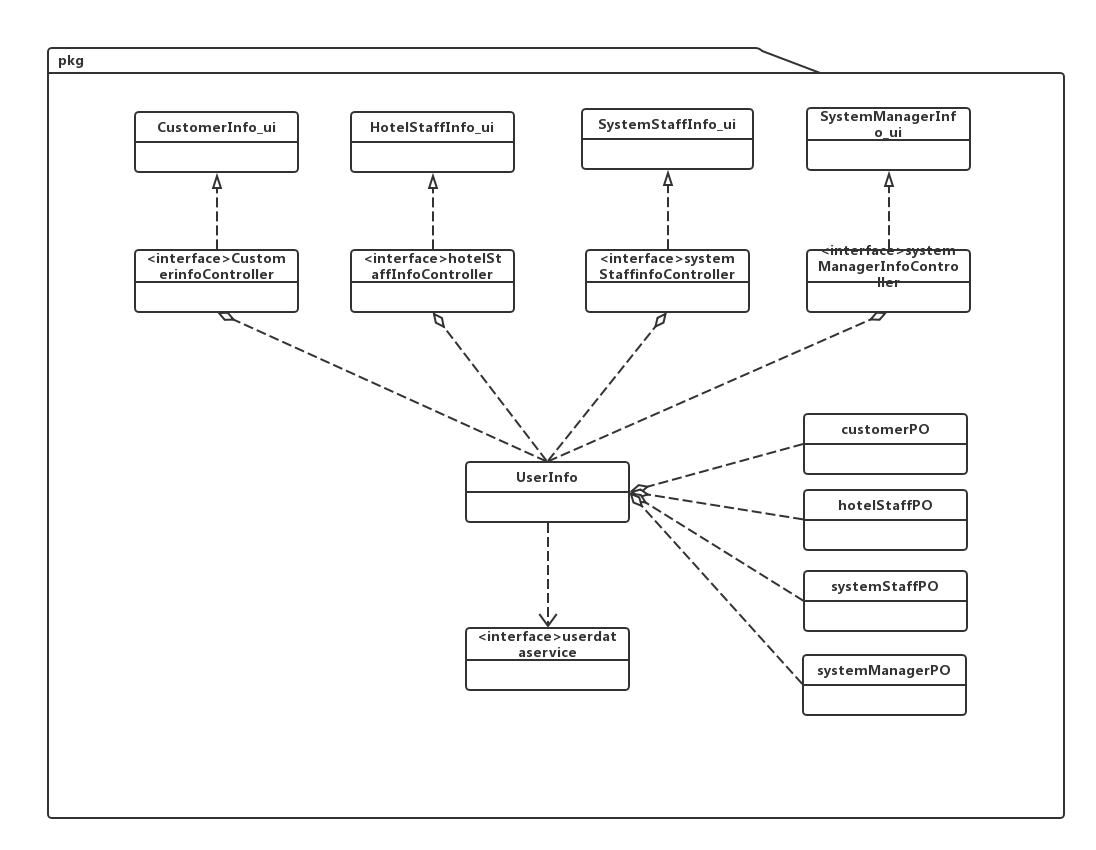
1. IEEE std 1471-2000

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

1. **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

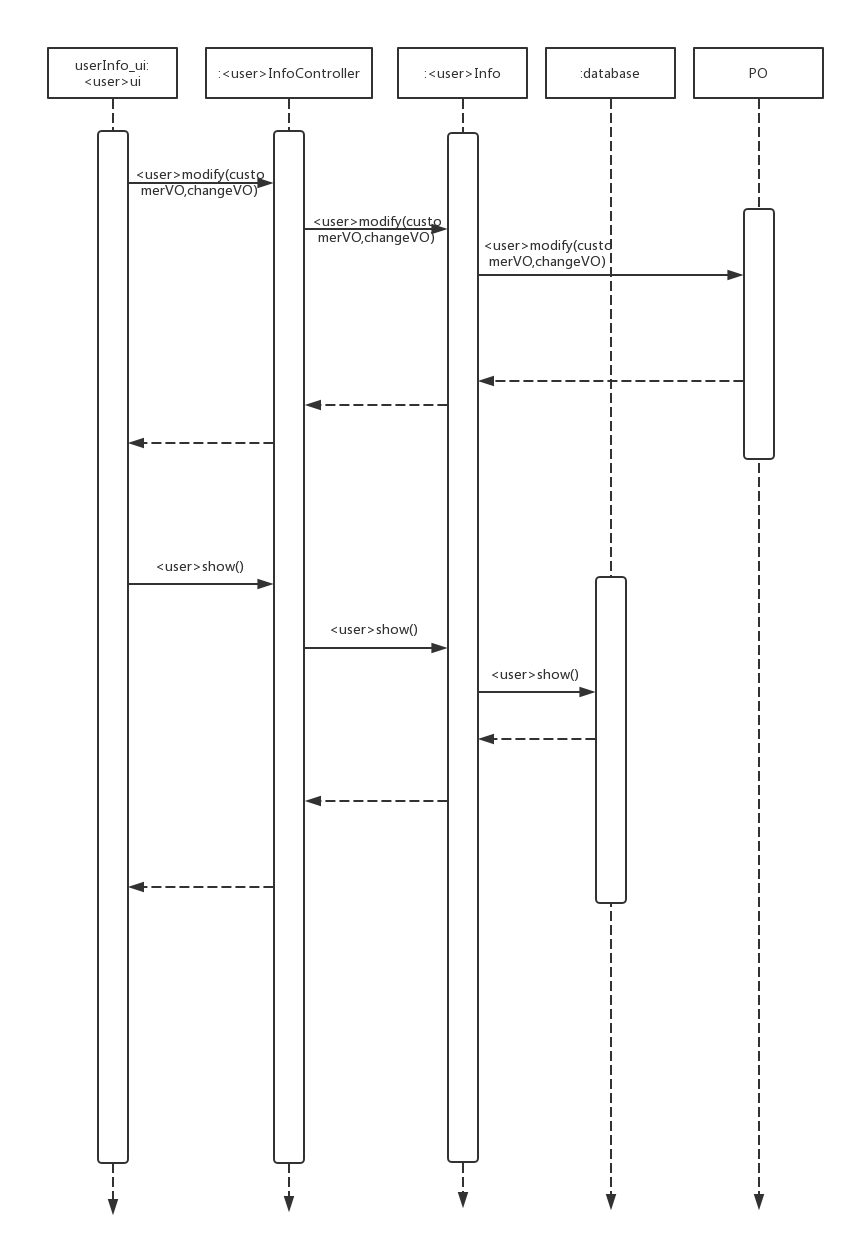
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| systemManagerInfoController. systemManagerInfoModify | 语法 | public ResultMessage  systemManagerInfoModify  (SystemManagerVO systemManager，  SystemManagerInfoChangeVO VO) |
| 前置条件 | 已知用户和对用户信息的修改策略，需要对自己的信息进行更改 |
| 后置条件 | 调用  userInfo. systemManagerInfoModify |
| 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CustomerPO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CustomerPO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hotelStaffPO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hotelStaffPO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SystemStaffPO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systemStaffPO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systemManagementPO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systemManagementPO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 |

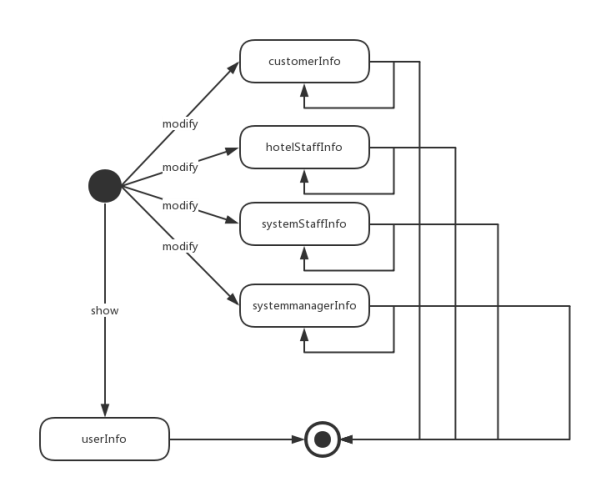
* + 1. userinfo\_bl模块的行为

图2.1.3-1 userinfo\_模块的顺序图



图中<user>可替换四种不同类型的用户，由于顺序流程相似，不用类似的图进行描述。

图2.1.3-2 userInfo\_bl模块的状态图

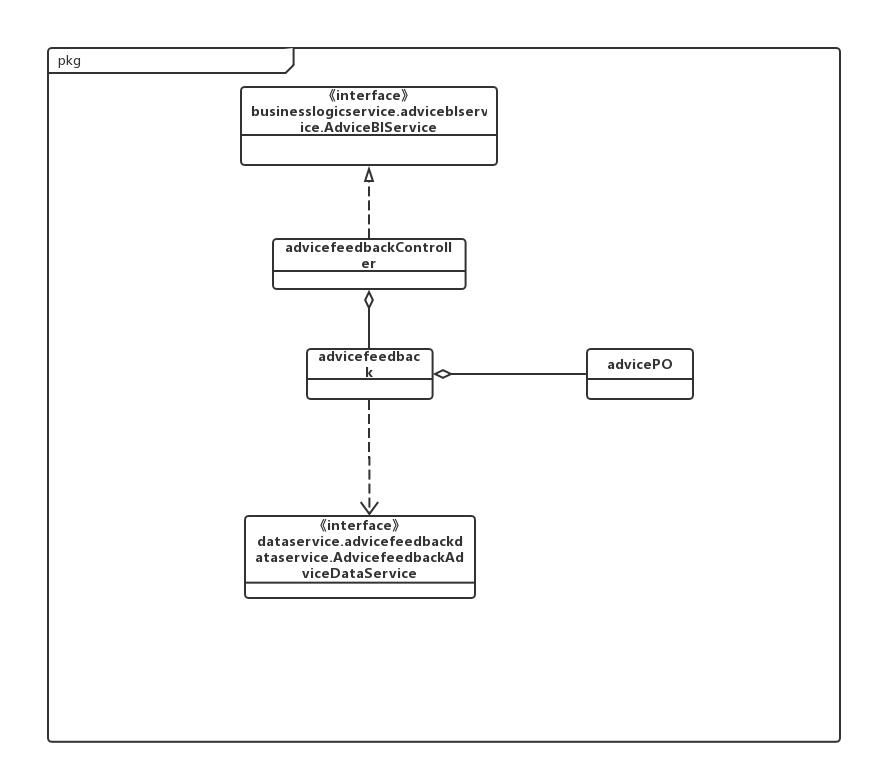


* 1. **advicefeedback\_bl模块的静态结构和动态行为**
     1. advicefeedback\_bl模块局部模块的职责

advicefeedback\_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。advicefeedback \_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advice

feedback\_blservice和advicefeedback\_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adviceFeedBackController和adviceReplyController两个Controller分别对于用户和网站营销人员的view层服务，同时将关于advicefeedback的相关的业务逻辑职责和逻辑控制职责委托给了advicefeedback对象。AdvicePO是作为管理反馈信息的持久化对象被添加到设计模型中的。

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 ResultMessage 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.modify | 语法 | Public ResultMessage 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.getUnProcessedAdvice() | | | 得到未处理的反馈信息 |
| Advicefeedback.getProcessedAdvice() | | | 得到已经处理的反馈信息 |
| 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 adviceReplyController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务(供接口) | | | |
| AdviceReplyController.getProcessedAdvice() | 语法 | Public List<AdvicefeedbackVO>  getProcessedAdvice (String ID) | |
| 前置条件 | 已创建一个advicefeedback领域对象 | |
| 后置条件 | 调用advicefeedback领域对象的getProcessedAdvice()方法 | |
| AdviceReplyController.getUnProcessedAdvice() | 语法 | Public List<AdvicefeedbackVO>  getUnProcessedAdvice (String ID) | |
| 前置条件 | 已创建一个advicefeedback领域对象 | |
| 后置条件 | 调用advicefeedback领域对象的getUnProcessedAdvice()方法 | |
| AdviceReplyController.setStateofAdvice | 语法 | public ResultMessage setStateOfAdvice(string adviceID) | |
| 前置条件 | 已经创建的advicefeedback领域对象，并且知道advicefeedback的id，需要对它的状态进行修改 | |
| 后置条件 | 调用adviceback领域对象的setState方法 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| Advicefeedback.getUnProcessedAdvice | | | 得到所有的未处理的反馈信息 |
| Advicefeedback.setState | | | 修改反馈信息的状态 |
| Advicefeedback.getProcessedAdvice | | | 得到已经处理的反馈信息 |

表2.2.2-3 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

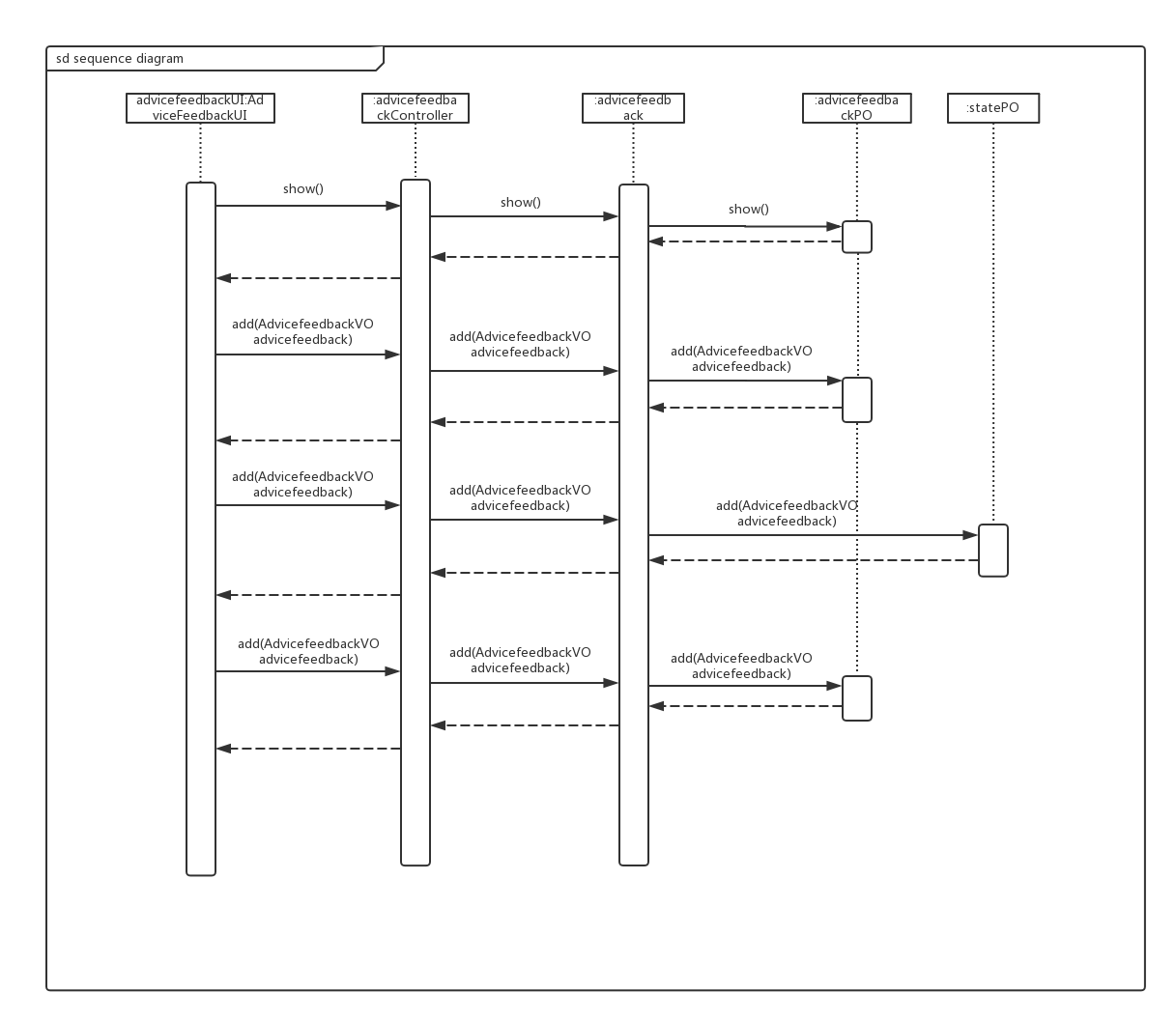
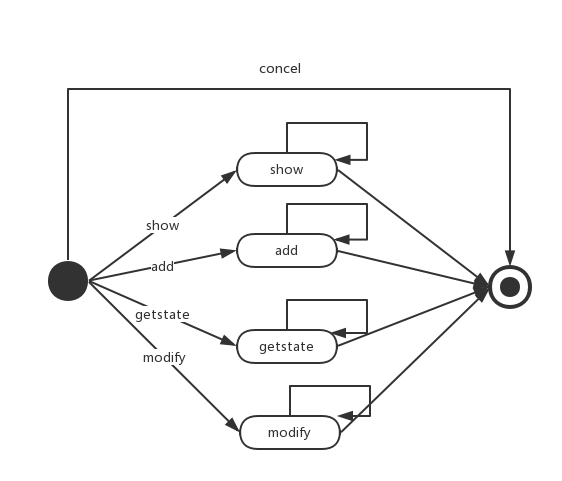


图2.2.3-2 advicefeedback模块状态图

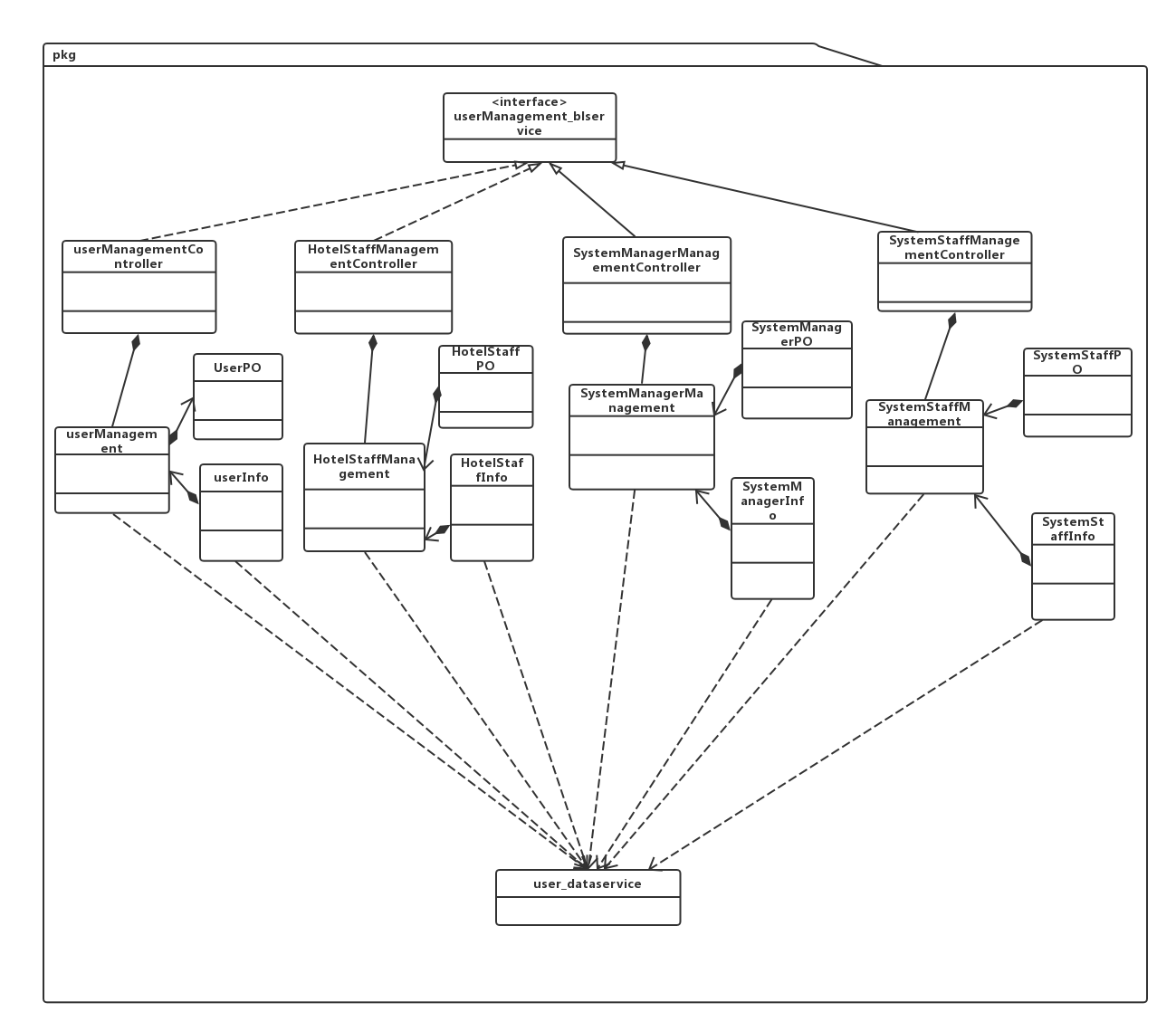


* 1. **usermanagement\_bl模块的静态结构和动态行为**
     1. usermanagement\_bl模块局部模块的职责

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 CustomerManagementController模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| CustomerManagementController.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 |
| CustomerManagementController.modifyUser | 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 | |
| CustomerManagement.modify | | 更改用户的信息 | |
| CustomerManagement.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 HotelManagementController的接口规范

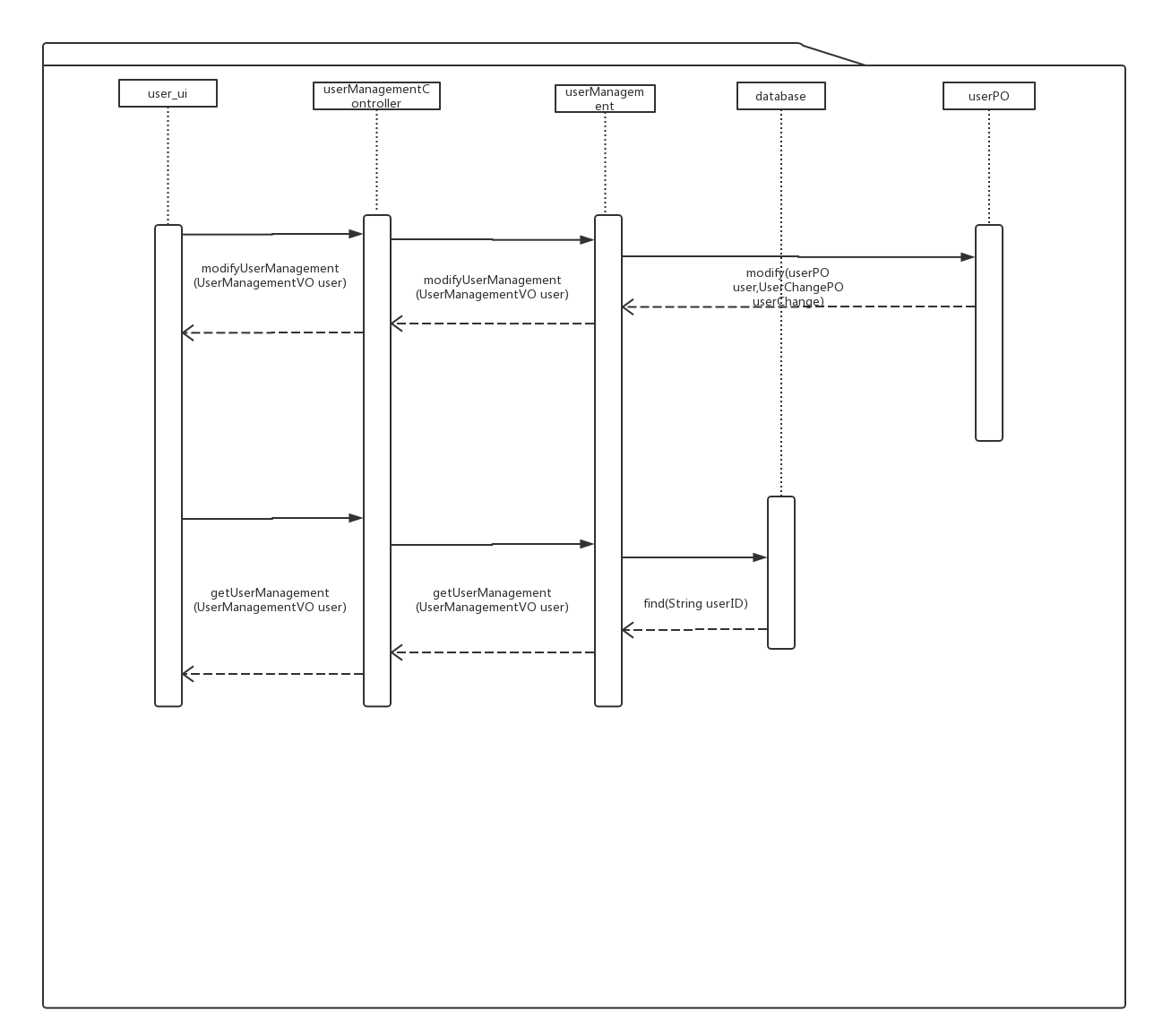
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| HotelManagementController.addNewHotel | 语法 | public ResultMessage addNewHotel(string hotel\_id) |
| 前置条件 | 需要新增一个酒店 |
| 后置条件 | 调用HotelManagement领域的addNewHotel方法 |
| HotelManagementController.addNewStaff | 语法 | public ResultMessage  addNewStaff (UserVO user) |
| 前置条件 | 需要给对应酒店新增一个酒店工作人员 |
| 后置条件 | 调用HotelManagement领域的addNewStaff方法 |
| 服务名 | | 服务 |
| HotelManagement.addNewHotel | | 系统中增加一个酒店 |
| HotelManagement.addNewStaff | | 系统中给对应酒店增加一个酒店工作人员 |

2.3.2-5 UserManagement\_bl的接口规范

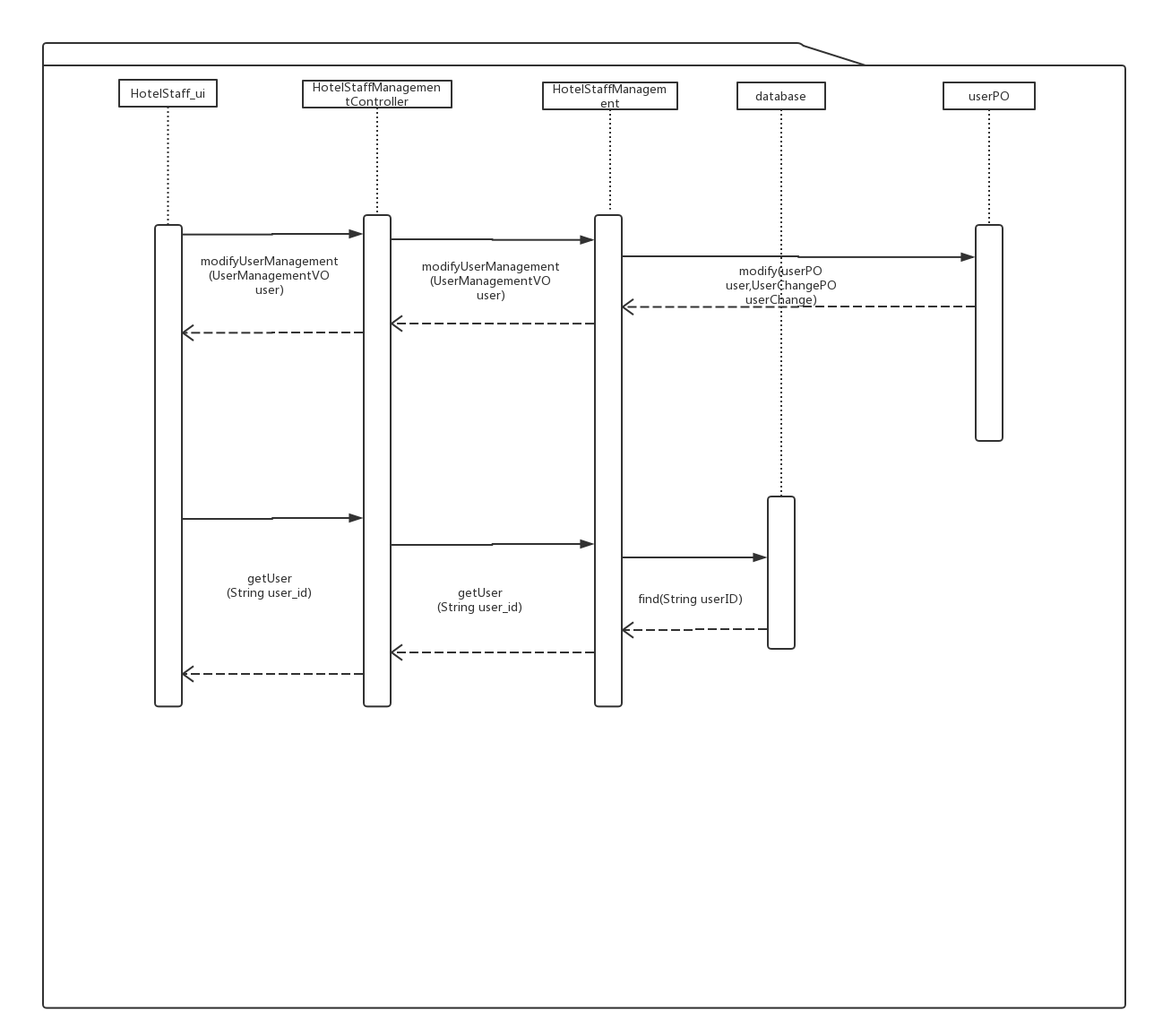
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CustomerManagement.get | 语法 | Public UserVO getUser  (String user\_id) |
| 前置条件 | 已知用户的ID，获取用户 |
| 后置条件 | 返回用户 |
| CustomerManagement.modify |  | Public ResultMessage modifyUserManagement(UserVOuser,UserChangeVO userChange) |
| 前置条件 | 已知用户，需要修改用户的信息 |
| 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 |
| HotelManagementController.addNewHotel | 语法 | public ResultMessage addNewHotel  (string hotel\_id) |
| 前置条件 | 系统中需要增加一个酒店 |
| 后置条件 | 新增一个酒店信息 |
| HotelManagementController.addNewStaff | 语法 | public ResultMessage addNewHotel  (UserVO user) |
| 前置条件 | 系统中需要为对应酒店新增一个工作人员 |
| 后置条件 | 新增一个酒店工作人员 |
| 需要的服务(需接口) | | |
| 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

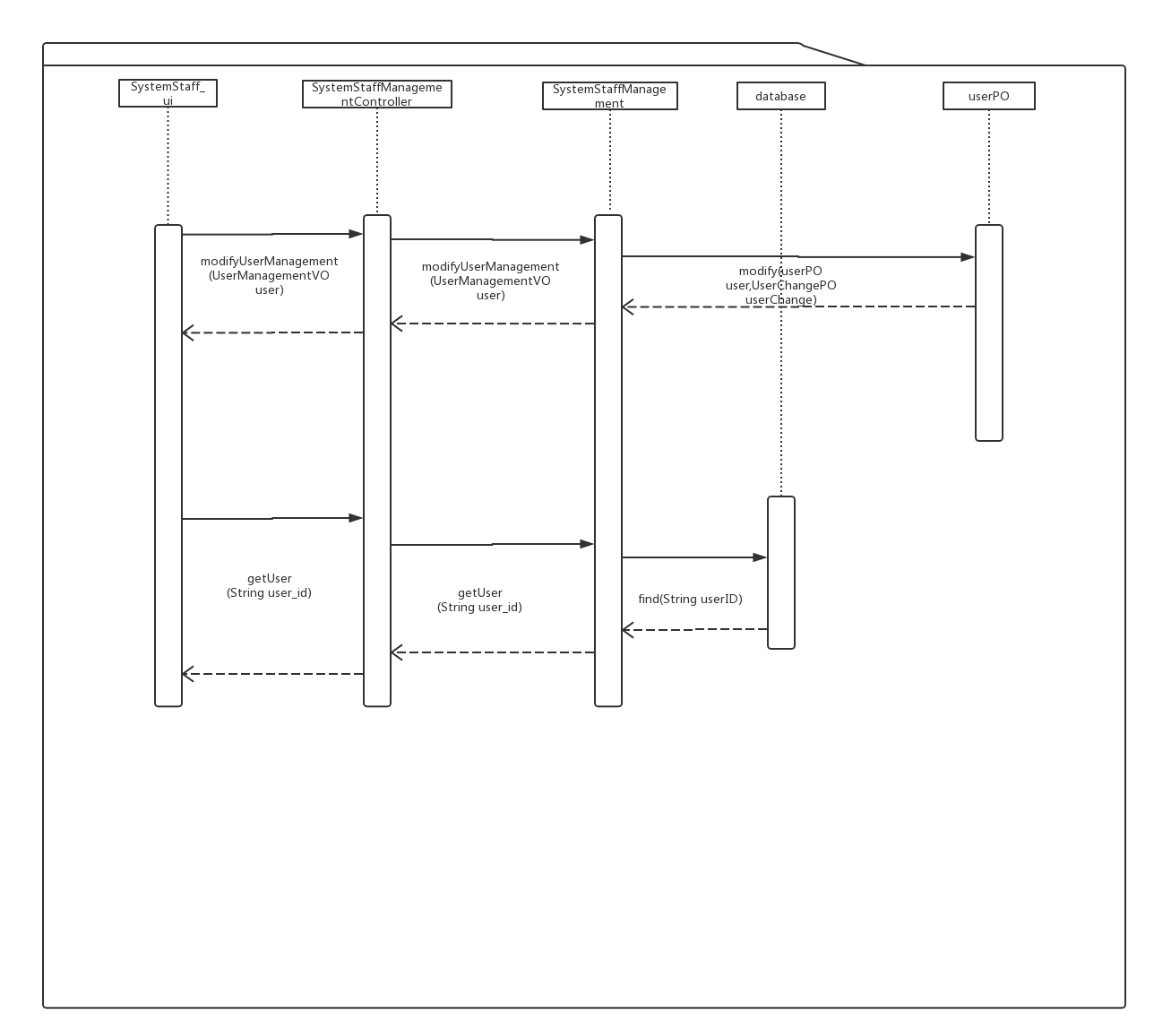
2.3.3-1 userManagement的顺序图



2.3.3-2 HotelstaffManagement的顺序图



2.3.3-3 SystemStaffManagement的顺序图



2.3.3-4 SystemManagerManagement的顺序图

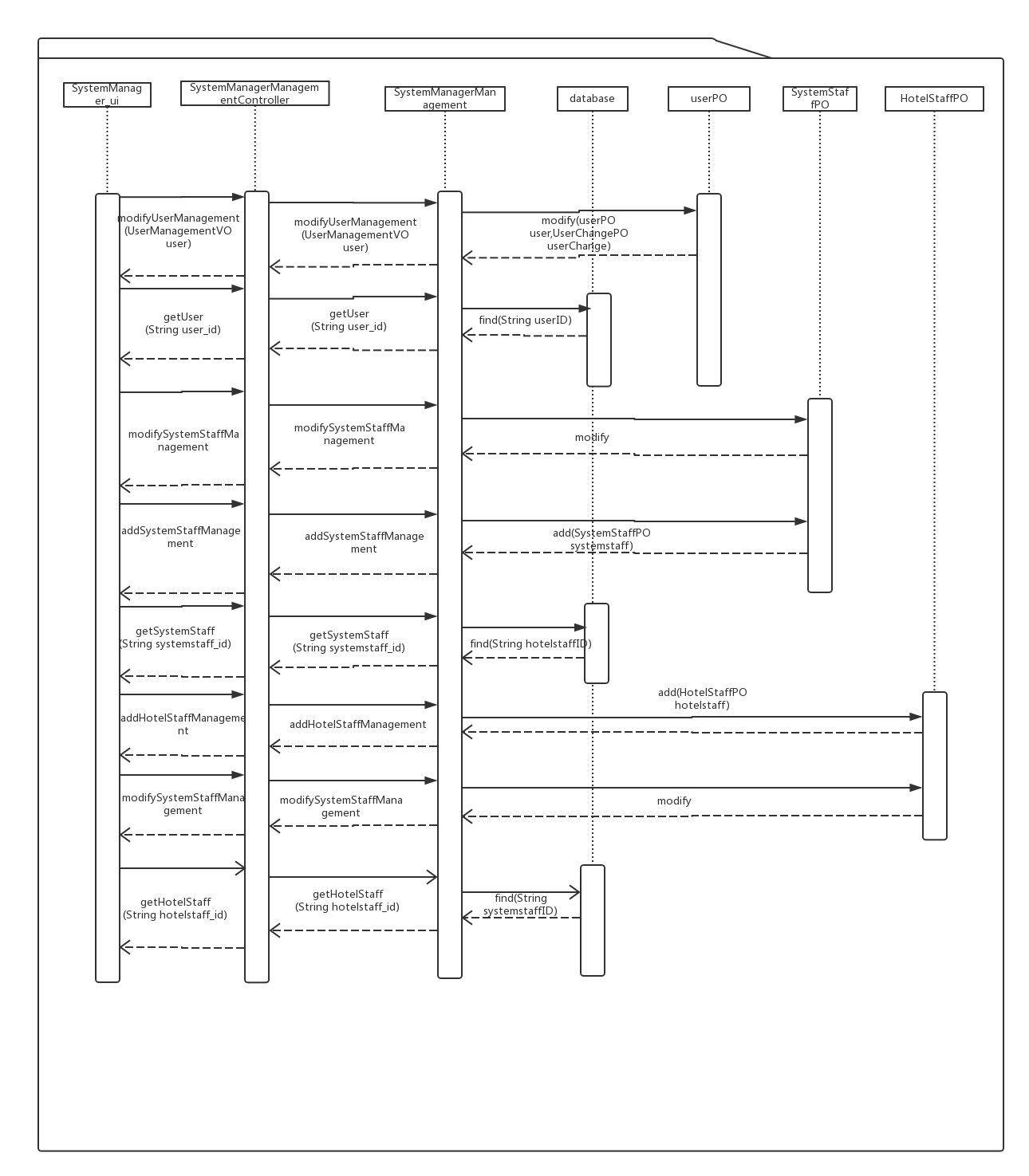


图2.3.3-5 userManagement的状态图



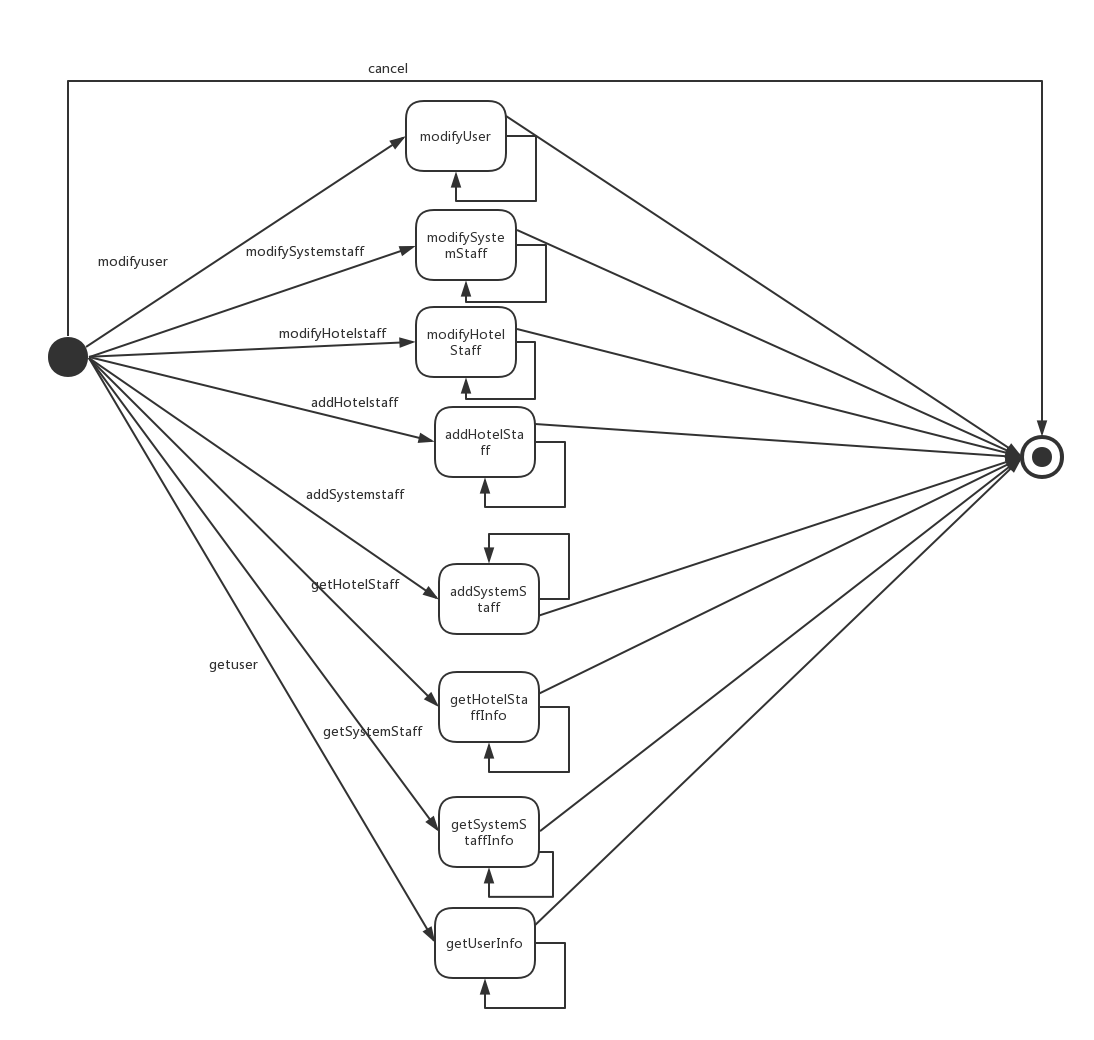
图2.3.3-6 HotelStaffManagement的状态图



图2.3.3-7 SystemStaffManagement的状态图



图2.3.3-7 SystemManagerManagement的状态图



* 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.

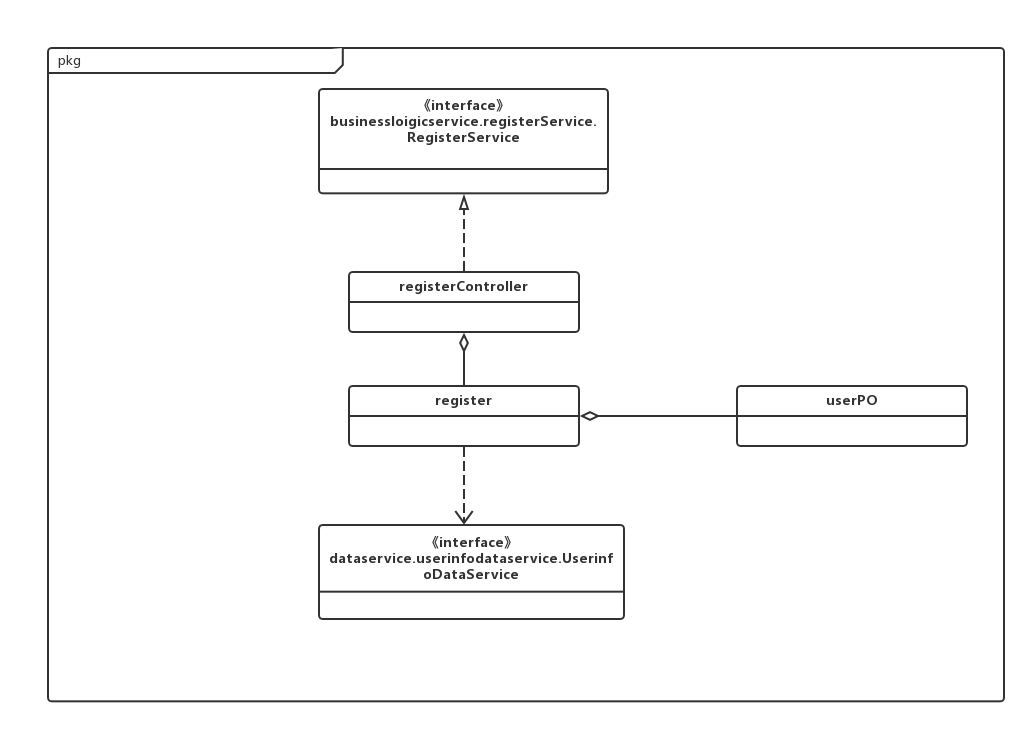


图2.4.1-1 register\_bl模块的各个类的设计

* + 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

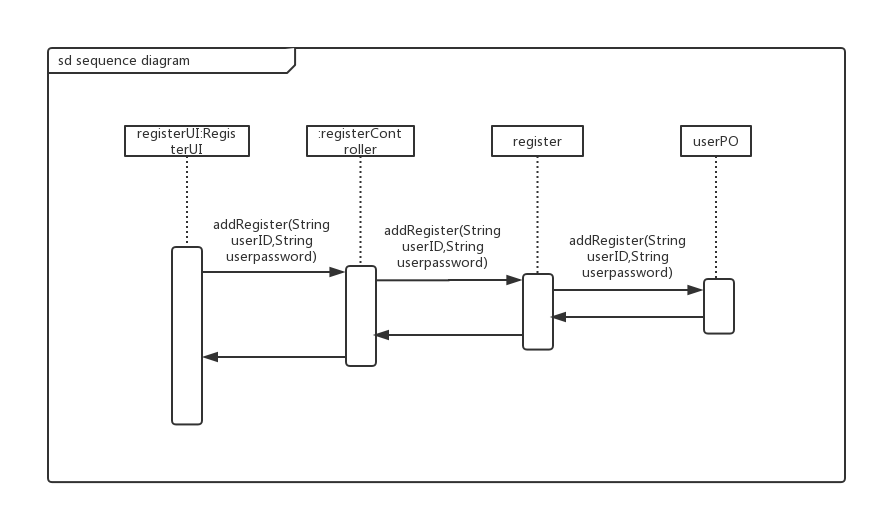


图2.4.3-1 registerBl顺序图

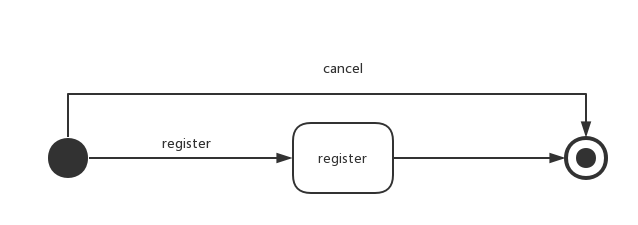


图2.4.3-2register的状态图

* 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

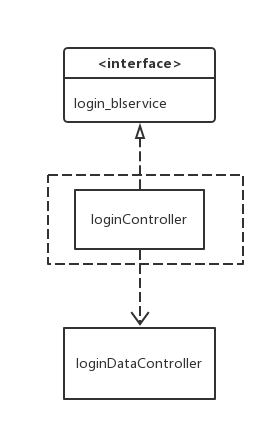


图2.5.1-1 login\_bl模块的各个类的设计

* + 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

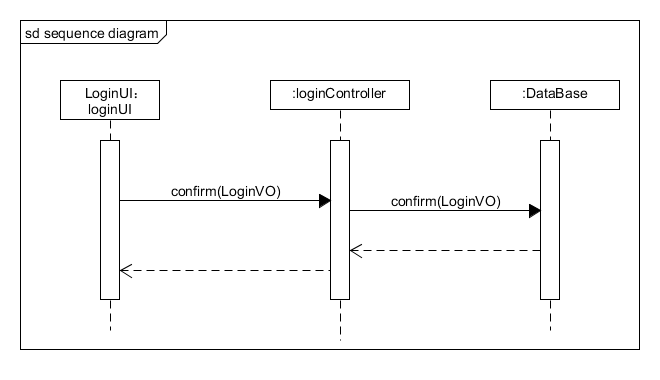


图2.5.3-1 loginBl顺序图

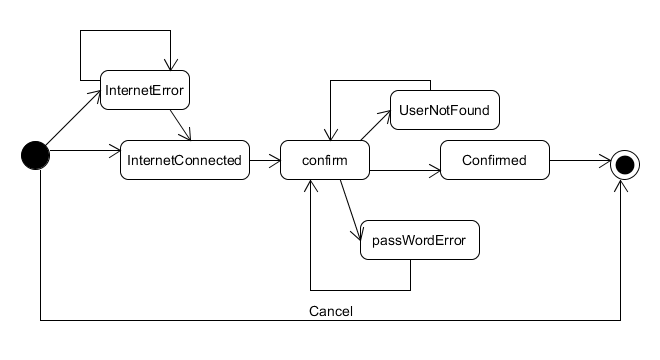


图2.5.3-1 login模块设计

* 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。

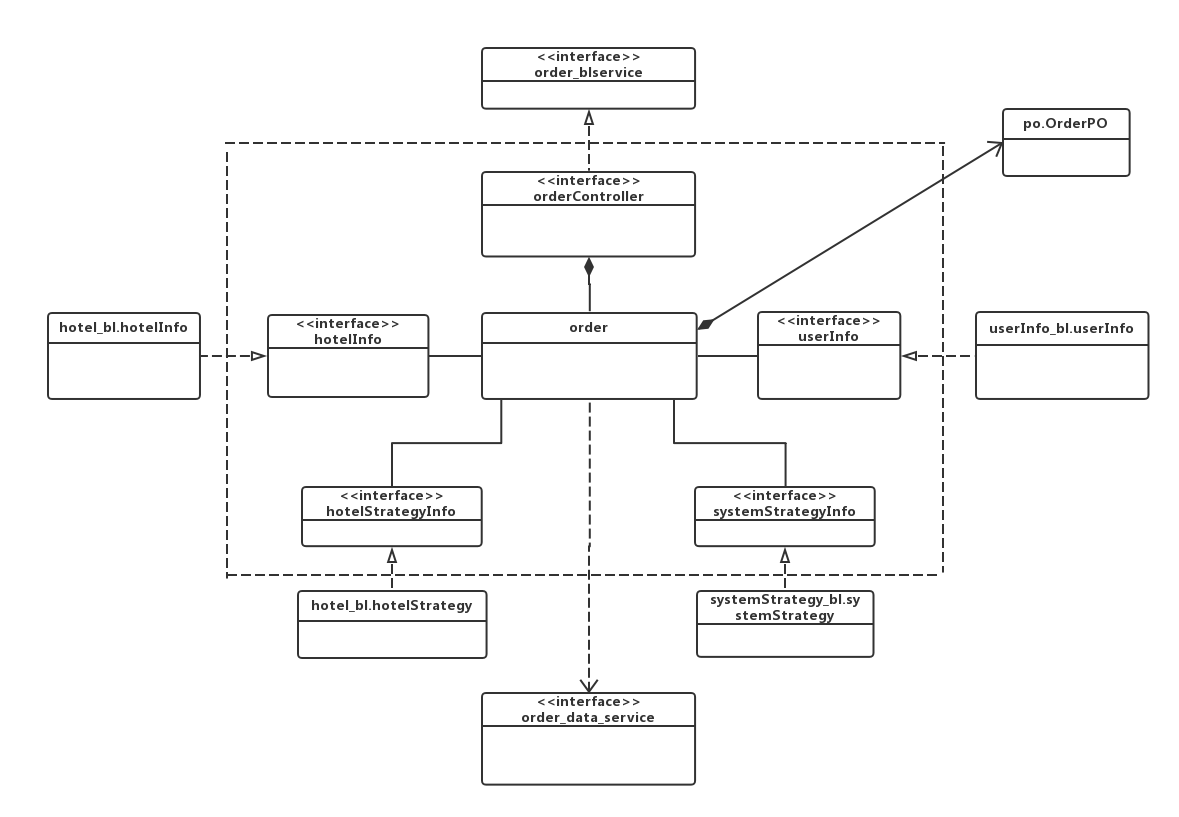


图2.6.1-1 orderbl模块的各个类设计

* + 1. order\_bl模块局部模块的接口规范

orderInfoController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderInfoController.getorderDetail | 语法 | public orderVo getorderDetail(orderID) |
| 前置条件 | 用户已查看订单 |
| 后置条件 | 返回一个订单单一持久化对象 |

orderviewController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderviewController.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 |
| OrderviewController.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 |
| OrderviewController.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 |
| OrderviewController.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 |
| OrderviewController.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 |
| OrderviewController.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 |
| OrderviewController.getUnfinishedOrders | 语法 | Public list<orderVo> getUnfinishedOrders(userID) |
| 前置条件 | 用户已登陆，已查看所有订单信息 |
| 后置条件 | 返回用户所有未执行的订单单一持久化序列列表 |
| OrderviewController.getAbnomalOrders | 语法 | Public list<orderVo> getAbnomalOrders(userID) |
| 前置条件 | 用户已登陆，已查看所有订单信息 |
| 后置条件 | 返回用户所有异常订单单一持久化序列列表 |
| OrderviewController.getHotelOrders | 语法 | Public list<orderVo> getHotelOrders(userID,hotelID) |
| 前置条件 | 用户已查看所有订单信息，选择了一家预订过酒店 |
| 后置条件 | 返回预定过酒店订单持久化单一序列列表 |
| 服务名 | | 服务 |
| OrderviewController.getState | | 获得一个订单的状态 |
| OrderviewController.getOrder | | 根据订单编号获得一个订单的详细信息 |
| OrderviewController.getOrdersofUsers | | 根据用户信息获得该用户的订单信息 |
| OrderviewController.createOrder | | 创建订单 |
| OrderviewController.saveOrder | | 保存订单 |
| OrderviewController.changeState | | 更改订单状态 |
| OrderviewController.getUnfinishedOrders | | 得到用户未完成的订单列表 |
| OrderviewController.getAbnomalOrders | | 得到用户异常的订单 |
| OrderviewController.getHotelOrders | | 得到预订过酒店的订单列表 |

Order\_bl模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Orderservice.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 |
| Orderservice.getorderDetail | 语法 | Public orderVo getorderDetail(orderID) |
| 前置条件 | 用户已查看订单 |
| 后置条件 | 返回一个订单单一持久化对象 |
| Orderservice.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 |
| Orderservice.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 |
| Orderservice.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 |
| Orderservice.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 |
| Orderservice.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 |
| OrderserviceController.getUnfinishedOrders | 语法 | Public list<orderVo> getAbnomalOrders(userID) |
| 前置条件 | 用户已登陆，已查看所有订单信息 |
| 后置条件 | 返回用户所有异常订单单一持久化序列列表 |
| OrderserviceController.getAbnomalOrders | 语法 | Public list<orderVo> getAbnomalOrders(userID) |
| 前置条件 | 用户已登陆，已查看所有订单信息 |
| 后置条件 | 返回用户所有异常订单单一持久化序列列表 |
| Orderservice.getHotelOrders | 语法 | Public list<orderVo> getHotelOrders(userID,hotelID) |
| 前置条件 | 用户已查看所有订单信息，选择了一家预订过酒店 |
| 后置条件 | 返回预定过酒店订单持久化单一序列列表 |
| 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

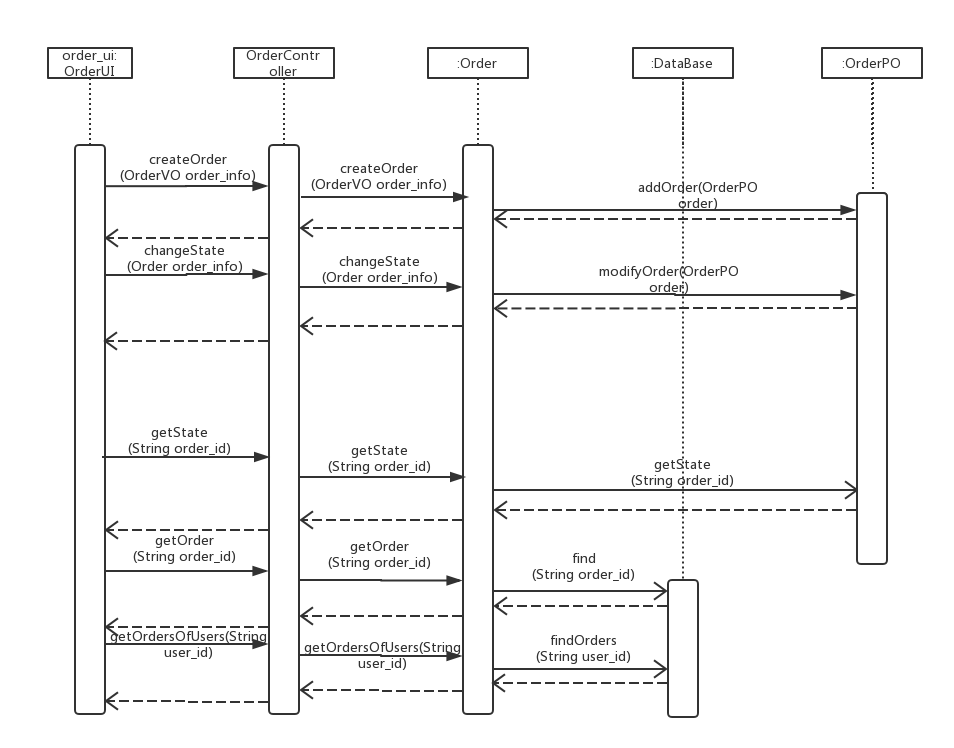


图2.6.3-1订单生成、保存、查找、获取、修改的顺序图

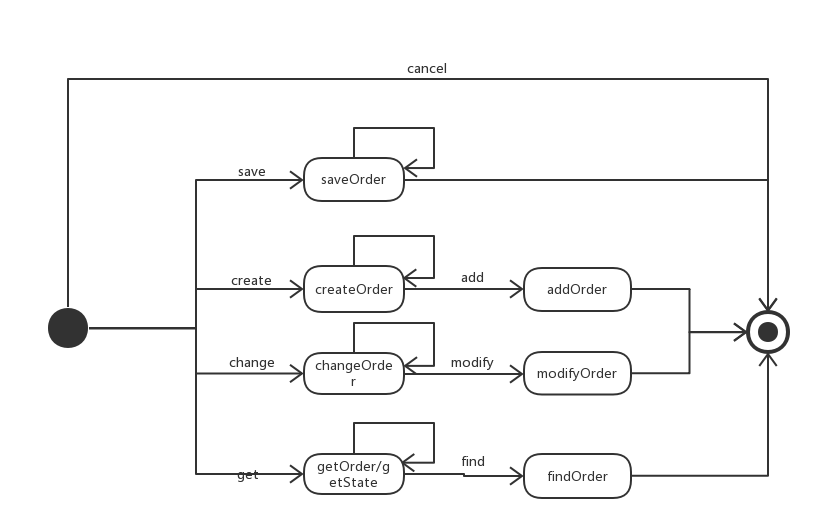


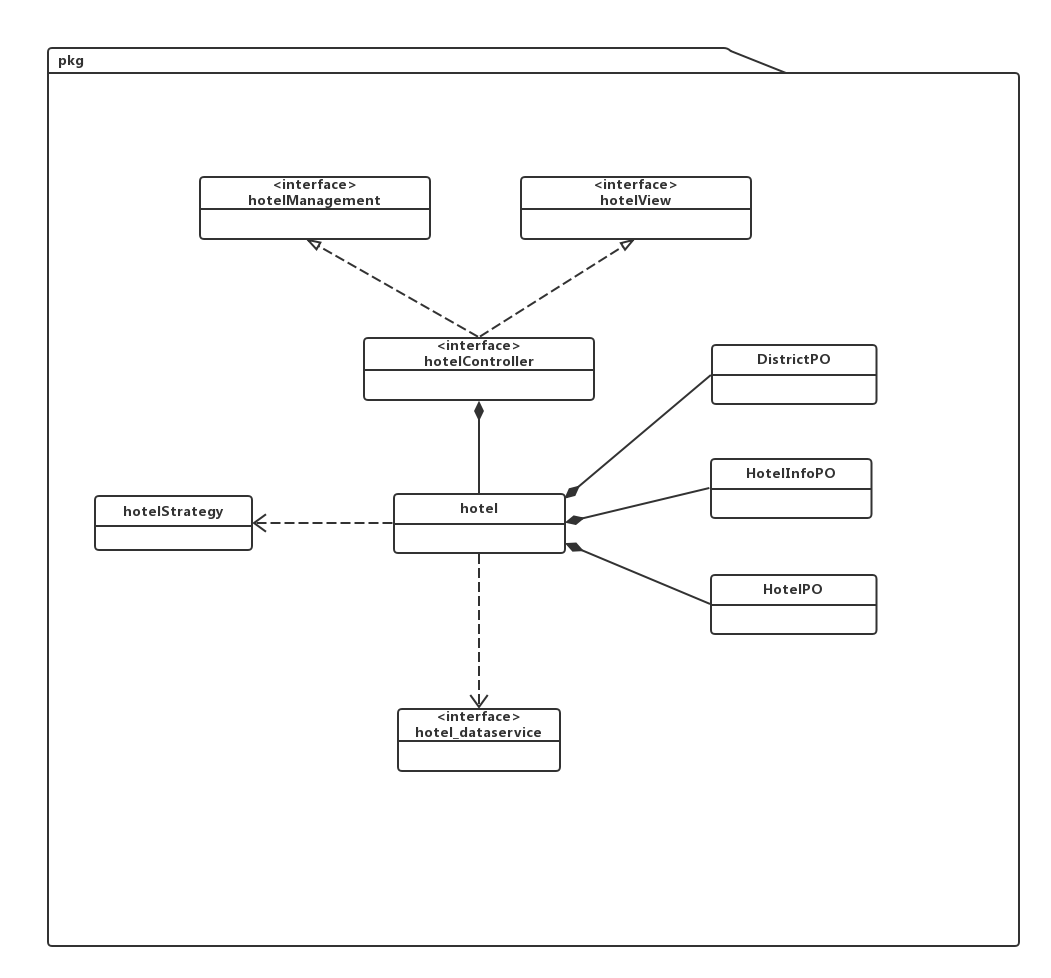
图2.6.3-2出库单生成、保存、查找、获取、修改的状态图

* 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hotelViewController（负责控制客户用户的酒店浏览界面的业务）、hotelRoomManagementController（负责控制酒店工作人员用户的客房管理界面的业务）、AssessmentManagementController（负责控制网站营销人员的评价管理界面的业务）、hotelAssessmentController（负责控制客户用户的酒店评价界面的业务）、hotelManagementController（负责控制网站管理人员的注册酒店界面的业务）、hotelInfoController（负责控制酒店营销人员的维护酒店信息界面的业务）六个类，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等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模块的设计类图：

图2.7.1-1 hotel\_bl模块的设计类图



* + 1. hotel\_bl模块局部模块的接口规范

表2.7.2-1 hotelViewController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| hotelViewController.getHotelInfo | 语法 | public HotelInfoVO  getHotelInfo(String hotelId) |
| 前置条件 | 已知酒店ID，获得酒店信息描述 |
| 后置条件 | 调用hotel.getHotelInfo方法 |
| hotelViewController.showHotelInfo | 语法 | public ResultMessage  showHotelInfo(HotelInfoVOHotelInfo) |
| 前置条件 | 已知酒店信息VO对象，需要在界面层显示 |
| 后置条件 | 在界面层列表中显示这些信息 |
| hotelViewController.getListOfHotelId | 语法 | public String[ ]  getListOfHotelId( ) |
| 前置条件 | 请求所有的hotelId |
| 后置条件 | 调用hotel.getListOfHotelId方法 |
| hotelViewController.getListOfHotel | 语法 | public <List>HotelInfoVO  getListOfHotel(String[] hotelId) |
| 前置条件 | 已知酒店Id列表，获得酒店列表 |
| 后置条件 | 调用  hotelViewController.getHotelInfo |
| hotelViewController.showHotelList | 语法 | public ResultMessage  showHotelList(<List>HotelInfoVO list) |
| 前置条件 | 已知排序过的酒店信息列表，在界面层显示酒店信息列表 |
| 后置条件 | 显示酒店信息列表 |
| hotelViewController.hotelListSort | 语法 | public <List> HotelInfoVO  hotelListSort(SortMethod method,  <List> HotelInfoVO list) |
| 前置条件 | 已知酒店信息列表和列表排序规则，对当前列表进行重排 |
| 后置条件 | 返回当前规下的重排 |
| hotelViewController. getListOfHotelprefer | 语法 | public String[ ]  getListOfHotelPrefer(String userId ) |
| 前置条件 | 已知当前客户Id，返回该客户历史喜好的酒店Id列表 |
| 后置条件 | 调用  hotel.getListOfHotelprefer方法 |
| hotelViewController.bookRoom | 语法 | public ResultMessage bookRoom(String hotelID) |
| 前置条件 | 已知当前酒店id，调用order模块方法生成订单 |
| 后置条件 | 调用  orderFormationController  实现页面跳转 |
| 服务名 | | 服务 |
| hotel.getHotelInfo | | 获得酒店信息 |
| hotel. getListOfHotelId | | 获得酒店id列表 |
| hotel.getListOfHotelprefer | | 获得用户喜好酒店id列表 |

表2.7.2-2 hotelRoomManagementController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| hotelRoomManagementController.roomModify | 语法 | public ResultMessage  roomModify(HotelRoomInfoVO currentInfo) |
| 前置条件 | 已知对于某种类型的房间的更改方案，对该酒店该房间的信息进行更改 |
| 后置条件 | 调用hotel. roomModify方法 |
| hotelRoomManagementController.getHotelRoomInfo | 语法 | public HotelRoomInfoVO  getHotelRoomInfo(String hotelId) |
| 前置条件 | 已知酒店ID，需要获得该酒店当前客房信息 |
| 后置条件 | 调用  hotel.getHotelRoomInfo方法 |
| hotelRoomManagementController.showHotelRoomInfo | 语法 | public ResultMessage  showHotelRoomInfo(HotelRoomInfoVO currentInfo) |
| 前置条件 | 已知当前客房信息，需要在界面层显示这些信息 |
| 后置条件 | 将这些信息显示在界面层 |
| 服务名 | | 服务 |
| hotel. roomModify | | 修改酒店客房信息 |
| hotel.getHotelRoomInfo | | 获得酒店当前客房信息 |

表2.7.2-3 AssessmentManagementController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| AssessmentManagementController.getLabelAssessment | 语法 | public <List>Label  getLabelAssessment(String hotelId) |
| 前置条件 | 已知酒店id，获得当前酒店的标签评价列表 |
| 后置条件 | 调用hotel.  getLabelAssessment |
| AssessmentManagementController.showLabelAssessment | 语法 | public ResultMessage  showLabelAssessment(<List>Label labelList) |
| 前置条件 | 已知酒店标签评价列表，在界面层显示统计过后的酒店标签评价信息 |
| 后置条件 | 调用hotel.  showLabelAssessment |
| AssessmentManagementController.deleteLabelAssessment | 语法 | public ResultMessage  deleteLabelAssessment(Label label) |
| 前置条件 | 已知酒店标签评价信息，删除酒店标签评价信息 |
| 后置条件 | 调用hotel.  deleteLabelAssessment |
| 服务名 | | 服务 |
| hotel.getLabelAssessment | | 获取标签评价 |
| hotel.showLabelAssessment | | 显示统计后的标签评价 |
| hotel.deleteLabelAssessment | | 删除单个标签评价 |

表2.7.2-4 hotelAssessmentController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| hotelAssessmentController.addAssessment | 语法 | public ResultMessage  addAssessment(String hotelID,String assessment) |
| 前置条件 | 已知酒店id和评价信息，需要新增一条酒店文字评价 |
| 后置条件 | 调用hotel. addAssessment方法 |
| hotelAssessmentController.addRankAssessment | 语法 | public ResultMessage  addRankAssessment(String hotelID,Rank rank) |
| 前置条件 | 已知酒店id和对酒店的星级评价，需要新增一条酒店星级评价 |
| 后置条件 | 调用  hotel. addRankAssessment |
| hotelAssessmentController.addLabelAssessment | 语法 | public ResultMessage  addLabelAssessment(String hotelID,<List>Label labelList) |
| 前置条件 | 已知酒店id和对酒店的标签评价list，需要新增一条酒店标签评价 |
| 后置条件 | hotel. addLabelAssessment |
| 服务名 | | 服务 |
| hotel. addAssessment | | 添加文字评价 |
| hotel. addRankAssessment | | 添加星级评价 |
| hotel. addLabelAssessment | | 添加标签评价 |

表2.7.2-5 hotelManagementController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| hotelManagementController.addHotel | 语法 | public String  addHotel(HotelVO hotel) |
| 前置条件 | 已知新酒店信息，需要生成新酒店，并返回新酒店的Id |
| 后置条件 | 调用hotel.addHotel |
| hotelManagementController.addHotelStaff | 语法 | public String  addHotelStaff(String hotelId) |
| 前置条件 | 已知酒店id，需要为其分配新的酒店工作人员，并返回星酒店工作人员ID |
| 后置条件 | 调用hotel. addHotelStaff |
| hotelManagementController.  showAvailableHotelList | 语法 | public ResultMessage  showAvailableHotelList(<List>HotelVO availableHotelList) |
| 前置条件 | 已知当前空闲酒店列表，在界面层显示该列表 |
| 后置条件 | 显示该列表 |
| hotelManagementController.getAvailableHotelList | 语法 | public <List>HotelVO  getAvailableHotelList() |
| 前置条件 | 请求当前空闲酒店列表 |
| 后置条件 | 调用hotel.getListOfHotelId方法和hotel.getHotelInfo方法 |
| 服务名 | | 服务 |
| hotel.getListOfHotelId | | 获得酒店工作列表 |
| hotel.getHotelInfo | | 获得酒店信息 |
| hotel.addHotelStaff | | 分配酒店工作人员 |
| hotel.addHotel | | 增加新酒店对象 |

表2.7.2-6 hotelInfoController类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| hotelInfoController.modifyHotelInfo | 语法 | public ResultMessage  modifyHotelInfo(HotolInfoVO hotelInfo) |
| 前置条件 | 已知对于酒店信息的修改方案，对酒店信息进行修改 |
| 后置条件 | 调用hotel. modifyHotelInfo |
| hotelInfoController.getHotelInfo | 语法 | public HotelInfoVO  getHotelInfo(String hotelId) |
| 前置条件 | 已知酒店ID，获得酒店信息描述 |
| 后置条件 | 调用hotel.getHotelInfo方法 |
| hotelInfoController.showHotelInfo | 语法 | public ResultMessage  showHotelInfo(HotelInfoVOHotelInfo) |
| 前置条件 | 已知酒店信息VO对象，需要在界面层显示 |
| 后置条件 | 在界面层列表中显示这些信息 |
| 服务名 | | 服务 |
| hotel.getHotelInfo | | 获得酒店信息 |
| hotel. modifyHotelInfo | | 修改酒店信息 |

表2.7.2-7 hotel类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| hotel.getHotelInfo | 语法 | public HotelInfoPO  getHotelInfo(String hotelId) |
| 前置条件 | 已知酒店ID，获得酒店信息描述 |
| 后置条件 | 返回对应的HotelInfoPO对象 |
| hotel. modifyHotelInfo | 语法 | public ResultMessage  modifyHotelInfo(HotolInfoPO hotelInfo) |
| 前置条件 | 已知对于酒店信息的修改方案，对酒店信息进行修改 |
| 后置条件 | 在data层修改酒店的详细信息 |
| hotel.getListOfHotelId | 语法 | public String[ ]  getListOfHotelId( ) |
| 前置条件 | 请求所有的hotelId |
| 后置条件 | 从data层调用当前所有酒店的ID，生成列表并返回 |
| hotel.addHotel | 语法 | public String  addHotel(HotelPO hotel) |
| 前置条件 | 已知新酒店信息，需要生成新酒店，并返回新酒店的Id |
| 后置条件 | 在数据层生成新HotePO对象，并返回对应的ID |
| hotel.addHotelStaff | 语法 | public String  addHotelStaff(String hotelId) |
| 前置条件 | 已知酒店id，需要为其分配新的酒店工作人员，并返回新酒店工作人员ID |
| 后置条件 | 在data层生成新工作人员用户，与酒店绑定，并返回工作人员id |
| hotel. addAssessment | 语法 | public ResultMessage  addAssessment(String hotelID,String assessment) |
| 前置条件 | 已知酒店id和评价信息，需要新增一条酒店文字评价 |
| 后置条件 | 在data层生成Assessment对象 |
| hotel. addLabelAssessment | 语法 | public ResultMessage  addLabelAssessment(String hotelID,<List>Label labelList) |
| 前置条件 | 已知酒店id和对酒店的标签评价list，需要新增一条酒店标签评价 |
| 后置条件 | 在data层生成LabelAssessment对象 |
| hotel. addRankAssessment | 语法 | public ResultMessage  addRankAssessment(String hotelID,Rank rank) |
| 前置条件 | 已知酒店id和对酒店的星级评价，需要新增一条酒店星级评价 |
| 后置条件 | 在data层生成RankAssessment对象 |
| hotel. roomModify | 语法 | public ResultMessage  roomModify(HotelRoomInfoPO currentInfo) |
| 前置条件 | 已知对于某种类型的房间的更改方案，对该酒店该房间的信息进行更改 |
| 后置条件 | 在data层修改对应酒店的房间信息 |
| hotel.getHotelRoomInfo | 语法 | public HotelRoomInfoPO  getHotelRoomInfo(String hotelId) |
| 前置条件 | 已知酒店ID，需要获得该酒店当前客房信息 |
| 后置条件 | 从data层返回对应的酒店房间信息 |
| hotel.getLabelAssessment | 语法 | public <List>Label  getLabelAssessment(String hotelId) |
| 前置条件 | 已知酒店id，获得当前酒店的标签评价列表 |
| 后置条件 | 从data层返回Label对象列表 |
| hotel.deleteLabelAssessment | 语法 | public ResultMessage  deleteLabelAssessment(Label label) |
| 前置条件 | 已知酒店标签评价信息，删除酒店标签评价信息 |
| 后置条件 | 在data层删除酒店标签评价信息 |
| hotel.getListOfHotelprefer | 语法 | public String[ ]  getListOfHotelPrefer(String userId ) |
| 前置条件 | 已知当前客户Id，返回该客户历史喜好的酒店Id列表 |
| 后置条件 | 调用userInfo.getHistoryOrder，并返回历史酒店预定信息；若无，返回登录IP附近商圈的酒店信息 |
| 服务名 | | 服务 |
| Hotel\_data.add(HotelPO hotel) | | 新增单一持久化序列 |
| Hotel\_data.delete(String hotelId) | | 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 hotelId) | | 查找列表持久化序列 |
| HotelStrategy. getHotelStrategy(HotelPO hotel) | | 返回酒店促销策略列表 |
| userInfo.getHistoryOrder(String userId) | | 返回客户历史订单信息 |

* + 1. hotel\_bl模块的行为

图2.7.3-1 hotel生成、保存、删除操作的顺序图

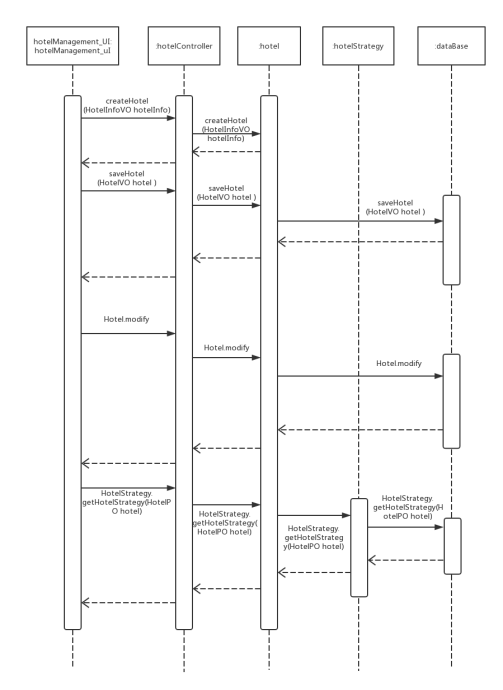
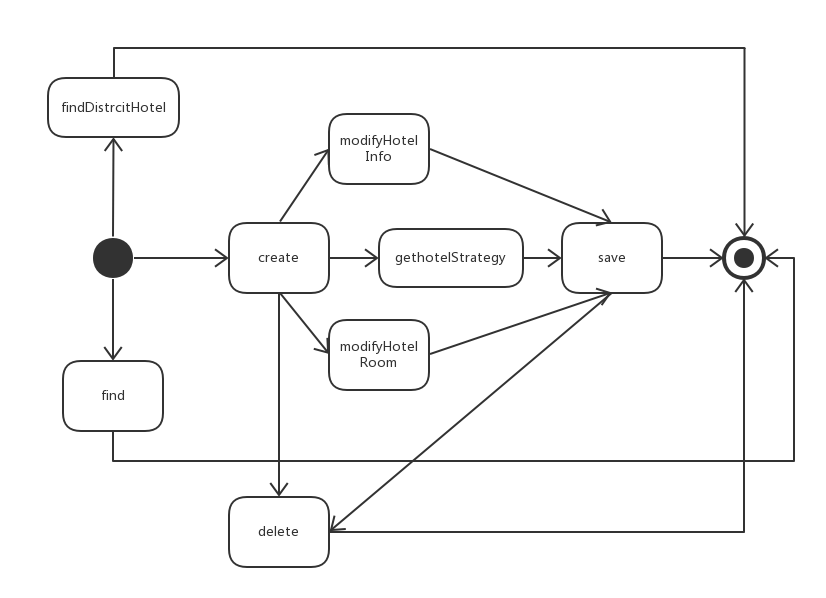


图2.7.3-2 hotel类的状态图



* 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2.8.1-1所示

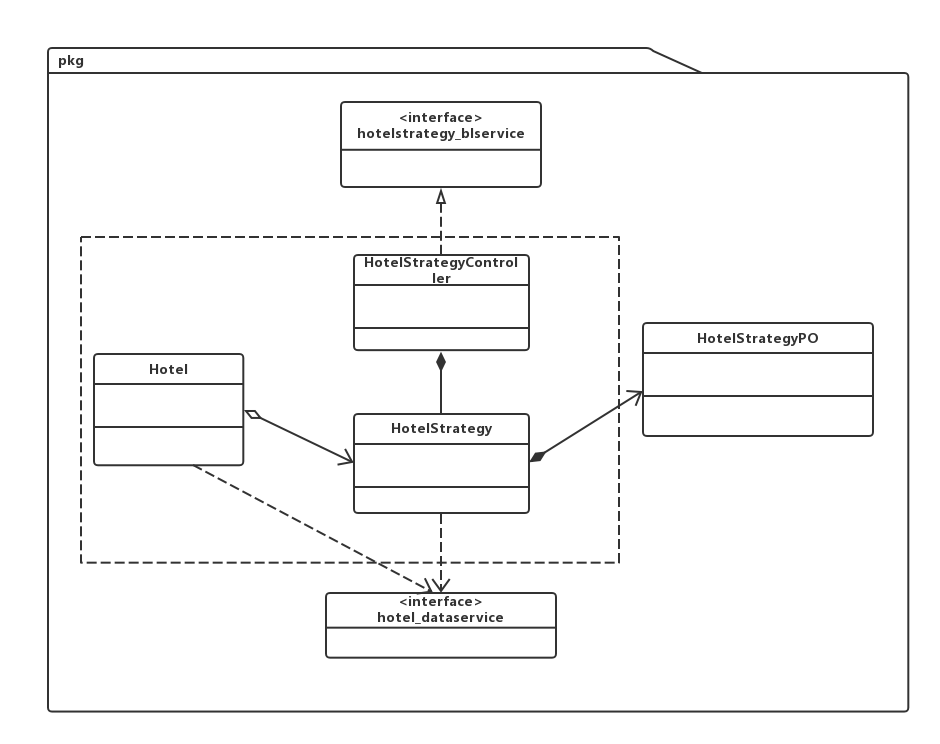


图2.8.1-1 hotelStrategy\_bl模块的各个类设计

* + 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  (HotelStrategyVO 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 | 接受界面的调用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  (String hotelStrategyId) | |
| 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 hotelstrategyId) | |
| 前置条件 | 已知酒店营销策略的ID，需要获得酒店促销策略 | |
| 后置条件 | 调用HotelStrategy领域对象的getHotelStrategy方法 | |
| HotelStrategyController.getListOfHotelStrategys | 语法 | public List<HotelStrategyVO>  getListOfHotelStrategys(String hotelId) | |
| 前置条件 | 已知酒店的Id，需要得到当前酒店所有的促销策略 | |
| 后置条件 | 调用HotelStrategy领域对象的getListOfHotelStrategys方法 | |
| HotelStrategyController.showHotelStrategy | 语法 | public ResultMessage  showHotelStrategy(HotelStrategyVo hotelStrategy) | |
| 前置条件 | 已知酒店促销策略，需要将其在展示层显示具体内容 | |
| 后置条件 | 将hotelStrategy内容在界面层中显示 | |
| 服务名 | | | 服务 |
| HotelStrategy. makeHotelStrategy | | | 制定酒店促销策略 |
| HotelStrategy.modifyHotelStrategy | | | 修改酒店促销策略 |
| HotelStrategydeleteHotelStrategy | | | 删除酒店促销策略 |
| HotelStrategy.getHotelStrategy | | | 获得某个酒店促销策略 |
| HotelStrategy.getListOfHotelStrategys | | | 获得酒店的所有促销策略 |
| HotelStrategy.showHotelStrategy | | | 将单个酒店促销策略在界面层进行展示 |

HotelStrategy\_bl模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| HotelStrategy. makeHotelStrategy | 语法 | public ResultMessage makeHotelStrategy  (HotelStrategyVO 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.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 | 更改酒店的促销策略 | |
| HotelStrategydeleteHotelStrategy | 语法 | public ResultMessage deleteHotelStrategy  (String hotelStrategyId) | |
| 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.getHotelStrategy | 语法 | public HotelstratefyVO getHotelStrategy(String HotelStrategyId) | |
| 前置条件 | 已知酒店营销策略的ID，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 hotelStrategyId) | | | 查找单一持久化序列 |
| Hotel\_data.findListOfstrategy (String hotelId) | | | 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

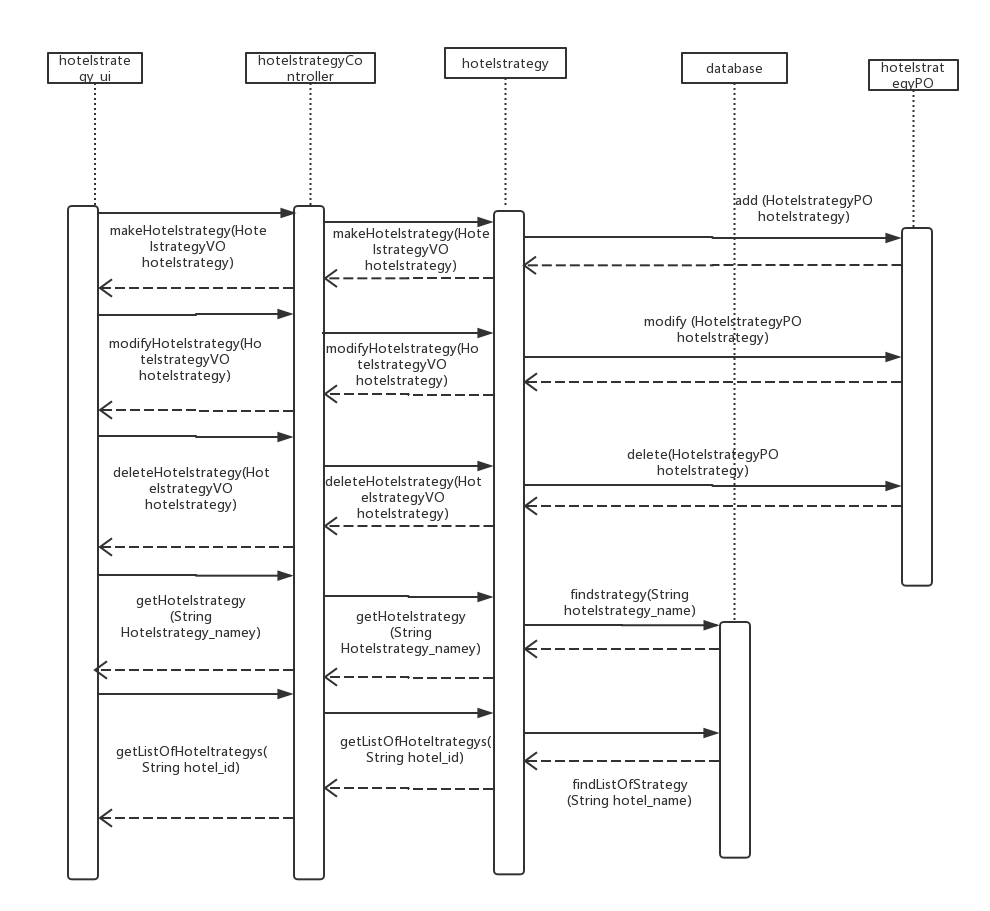
****

图2.8.3-1 对于酒店促销策略的制定、修改和删除的顺序图

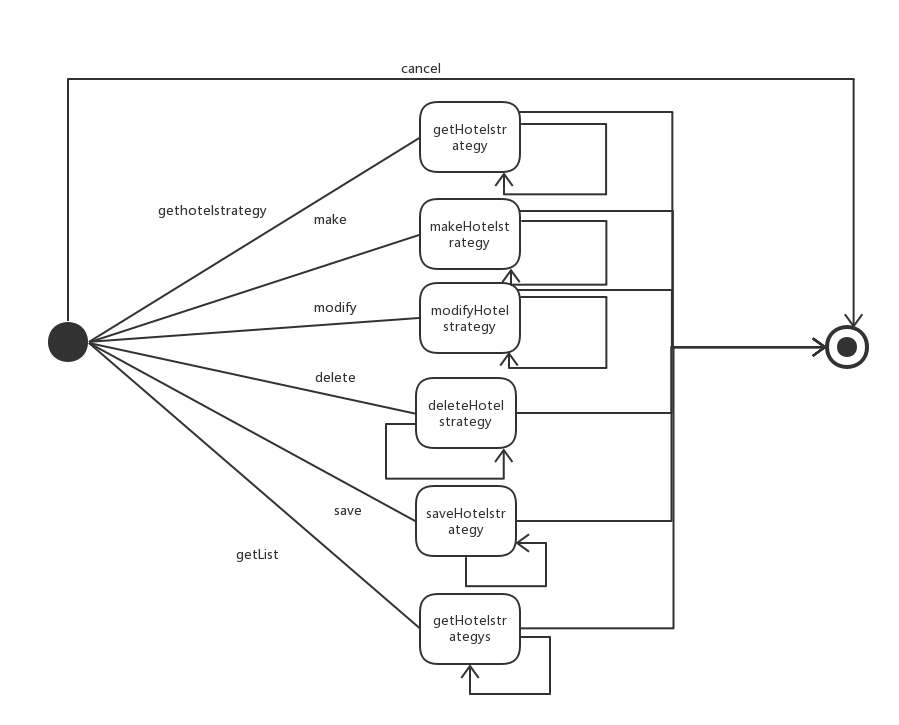


图2.8.3-2酒店促销策略对象的状态图

* 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

systemStrategy\_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。

systemStrategy\_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systemStrategy\_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systemStrategyController，这样systemStrategyController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systemStrategy\_bl对象。systemStrategyPO是作为酒店促销策略的持久化对象添加到设计模型中。

systemStrategybl模块的设计如图2.9.1-1所示

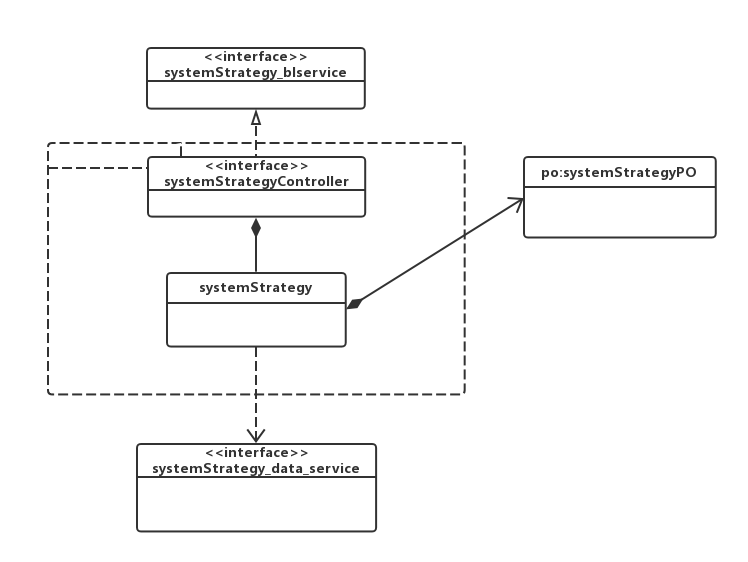


图2.9.1-1 systemStrategybl模块的各个类设计

* + 1. systemstrategy\_bl模块局部模块的接口规范

systemstrategyController类的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| systemStrategyController.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 | 调用systemStrategy领域对象的makeSystemStrategy方法 | |
| systemStrategyController.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 | 调用systemStrategy领域对象getSystemStrategy方法 | |
| systemStrategyController.Save | 语法 | public ResultMessage 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 | 调用systemStrategy领域对象的saveSystemStrategy方法 | |
| systemStrategyController.modify | 语法 | public ResultMessage 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 | 调用systemStrategy领域对象的modifySystemStrategy方法 | |
| systemStrategyController.delete | 语法 | public ResultMessage 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 | 调用systemStrategy领域对象的deleteSystemStrategy方法 | |
| systemStrategyController.getall | 语法 | public List<systemdstrategyVO> getall() | |
| 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 | 调用systemStrategy领域对象的getall方法 | |
| 服务名 | | | 服务 |
| systemstrategy.make | | | 制定一个网站营销策略 |
| systemStrategy.get | | | 获取一个网站营销策略 |
| systemStrategy.Save | | | 保存一个网站营销策略 |
| systemStrategy.modify | | | 修改一个网站营销策略 |
| systemStrategy.delete | | | 删除一个网站营销策略 |
| SystemStrategy.getall | | | 返回所有的已经制定的网站营销策略 |

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 ResultMessage 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 ResultMessage 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 ResultMessage 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.getall | 语法 | public List<systemdstrategyVO> getall() | |
| 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.getall() | | | 查找列表持久化序列 |

* + 1. systemstrategy\_bl模块的行为

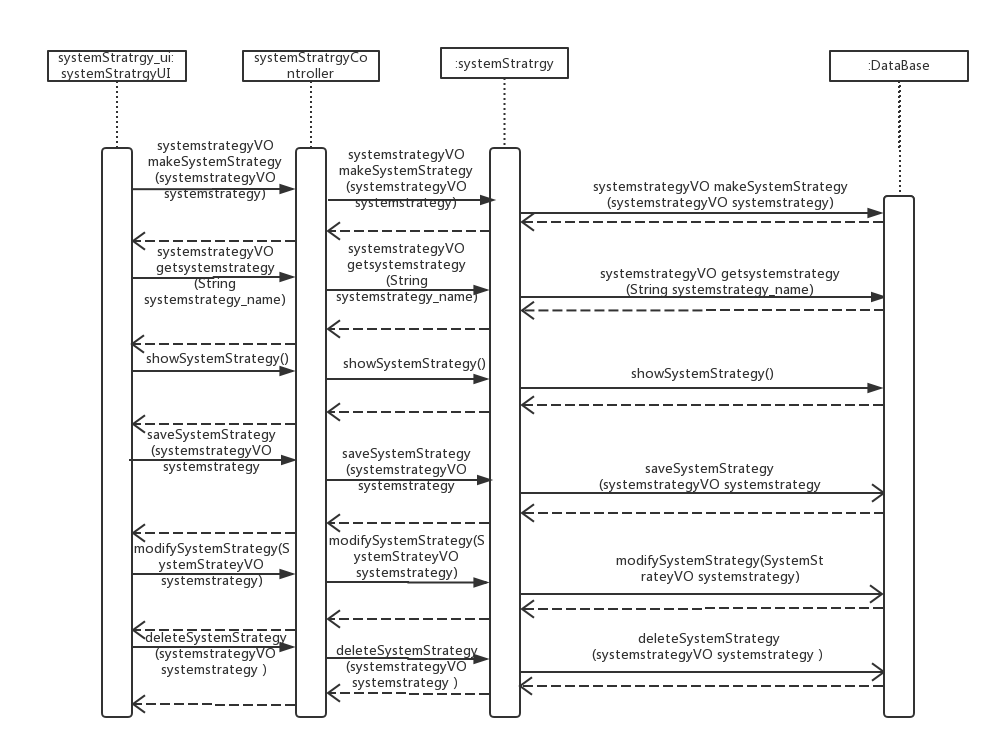


图2.9.3-1 systemStrategy\_bl的顺序图

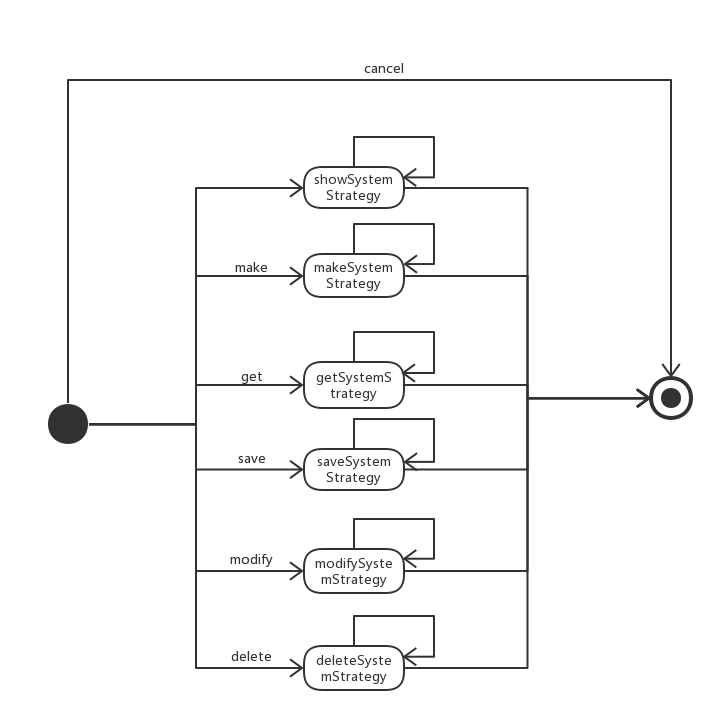


图2.9.3-2 systemStrategy\_bl的状态图

* 1. **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