### user模块

1. 模块概述

user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。

user模块的职责和接口参见软件系统结构描述文档内对该模块的描述。

1. 整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。在展示层和业务逻辑层之间添加userBLService接口。为了隔离业务逻辑职责和逻辑控制职责，添加userController，这样userController会将对用户信息的增改查和酒店添加的业务逻辑处理委托给user对象。guestPO、hotelWorkerPO、webMarketerPO、webManagerPO是作为用户信息的持久性对象被添加到设计模型中去的。



图1—user模块设计类图

1. 模块内部类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UserController.add | 语法 | Public ResultMessage add (String userID,UserVO userVO) |
| 前置条件 | 用户输入用户编号， 输入的用户信息符合信息规范 |
| 后置条件 | 系统返回是否成功添加用户；是则提示添加成功，否则提示添加失败原因 |
| UserController.modify | 语法 | Public ResultMessage modify (String userID,UserVO userVO) |
| 前置条件 | 用户输入用户编号， 输入的用户信息符合信息规范 |
| 后置条件 | 系统返回是否成功修改用户信息；是则提示修改成功，否则提示修改失败原因 |
| UserController.getSingle | 语法 | Public UserVO getSingle (String userID) |
| 前置条件 | 用户输入用户编号 |
| 后置条件 | 系统返回与用户编号对应的用户信息 |
| UserController.addHotel | 语法 | Public ResultMessage addHotel(String hotalID, HotelVO hotelVO) |
| 前置条件 | 网站管理人员输入酒店编号和符合信息规范的酒店信息 |
| 后置条件 | 系统返回是否成功添加酒店信息；是则提示添加成功，否则提示添加失败原因 |
| UserController.modifyCredit | 语法 | Public ResultMessage modifyCredit (String guestID ,float creditNum) |
| 前置条件 | 网站营销人员输入客户编号和更改后的信用值 |
| 后置条件 | 系统返回是否成功修改信用值；是则提示修改成功，否则提示修改失败原因 |
| UserController.getAll | 语法 | Public List<UserVO> getAll(UserType userType) |
| 前置条件 | 用户输入用户类型 |
| 后置条件 | 系统返回该用户类型的所有用户的信息 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| UserController.add | 系统添加用户信息 | |
| UserController.modify | 系统修改用户信息 | |
| UserController.getSingle | 系统返回单一用户的信息 | |
| UserController.addHotel | 系统添加酒店信息 | |
| UserController.getAll | 系统返回指定用户类型的所有用户的信息 | |
| UserController.modifyCredit | 系统修改指定客户的信用值 | |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| User.getAll | 语法 | Public List<UserVO> getAll(UserType userType) |
| 前置条件 | 无 |
| 后置条件 | 系统返回指定用户类型的所有用户信息 |
| User.getSingle | 语法 | Public UserVO getSingle(String userID，UserType userType) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 系统返回与该ID对应的user信息 |
| User.add | 语法 | Public ResultMessage add(UserVO newUserVO) |
| 前置条件 | 输入的userID不存在，password和所有信息符合规范 |
| 后置条件 | 系统新增该用户并持久化保存该用户的数据 |
| User.modify | 语法 | Public ResultMessage modify(UserVO userVO) |
| 前置条件 | 用户信息符合规则且齐全 |
| 后置条件 | 系统修改该用户信息并持久化保存该用户数据 |
| User.addHotel | 语法 | Public ResultMessage addHotel(HotelVO newHotelVO, String hotelID) |
| 前置条件 | 输入的hotelID不存在,所有酒店信息符合规范 |
| 后置条件 | 系统新增该酒店并持久化保存该酒店的数据 |
| User.modifyCredit | 语法 | Public ResultMessage modifyCredit (float creditNum) |
| 前置条件 | 网站营销人员为客户信用充值 |
| 后置条件 | 系统修改该客户信用值并持久化保存该用户数据 |
| User.getLoginInfo | 语法 | Public ResultMessage getLoginInfo(String userID, UserType userType) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 系统返回与该用户ID对应的密码  如果ID不存在，则系统返回该ID对应用户不存在 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| guestDataService.  add(GuestPO newGuestPO) | 根据提供的newGuestPO增加客户持久化对象 | |
| guestDataService.  getSingle (String guestID) | 根据userID查找客户持久化对象 | |
| guestDataService.modify  (GuestPO guestPO) | 修改单一客户持久化对象 | |
| hotelWorkerDataService  add(HotelWorkerPO newHotelWorkerPO) | 根据提供的newHotelWorkerPO增加酒店工作人员持久化对象 | |
| hotelWorkerDataService  getSingle (String hotelWorkerID) | 根据userID查找酒店工作人员持久化对象 | |
| hotelWorkerDataService.modify  (HotelPO hotelPO) | 修改单一酒店工作人员持久化对象 | |
| hotelWorkerDataService.init() | 清空所有酒店工作人员持久化对象 | |
| webMarketerDataService.  add(WebMarketerPO newWebMarketerPO) | 根据提供的newWebMarkterPO增加网站营销人员持久化对象 | |
| webMarkterDataService.  getSingle (String webMarketerID) | 根据userID查找网站营销人员持久化对象 | |
| webMarkterDataService.modify(WebMarketerPO webMarketerPO) | 修改单一网站营销人员持久化对象 | |
| webMarketerDataService.init() | 清空所有网站营销人员持久化对象 | |
| webManagerDataService.  add(WebManagerPO newWebManagerPO) | 根据提供的newWebManagerPO增加网站管理人员持久化对象 | |
| webManagerDataService.  getSingle (String webManagerID) | 根据userID查找网站管理人员持久化对象 | |
| webManagerDataService.modify  (WebManagerPO webManagerPO) | 修改单一网站管理人员持久化对象 | |
| webManagerDataService.init() | 清空所有网站管理人员持久化对象 | |
| Hotel.add | 新增该酒店并持久化保存该酒店的数据 | |

1. 业务逻辑层的动态模型

User顺序图

1：guestUI

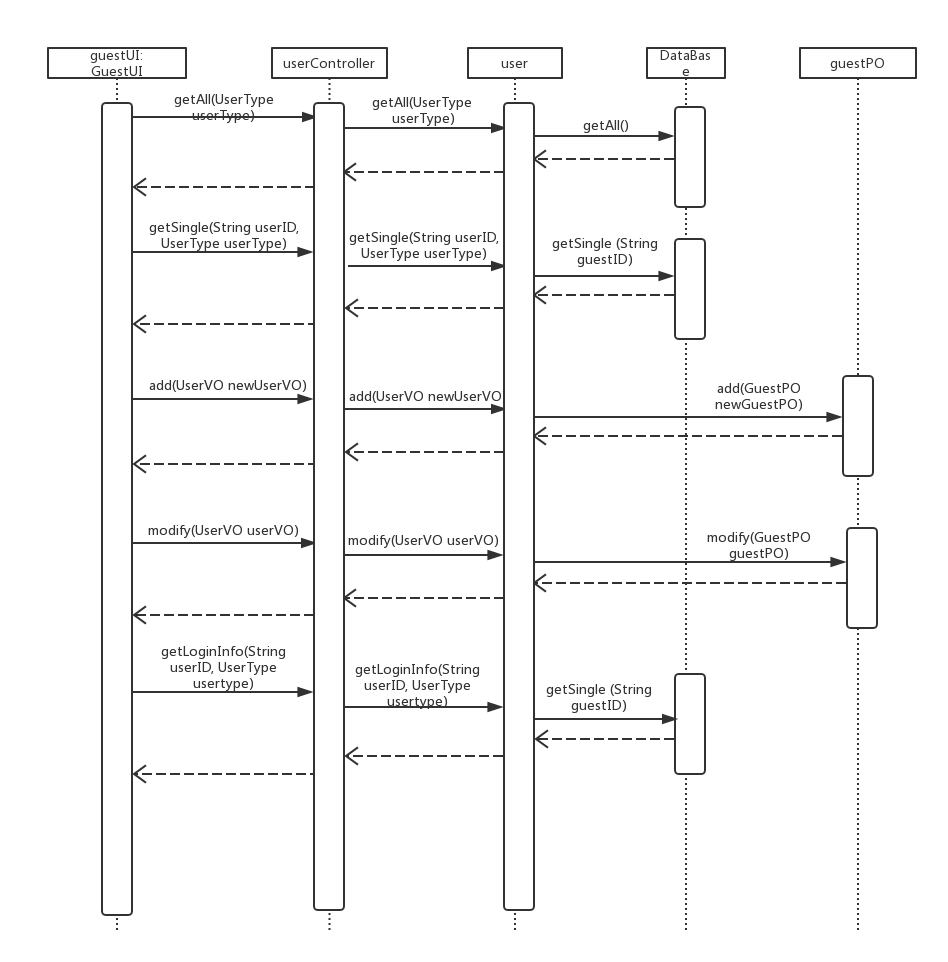


图2—user模块设计顺序图（guest）

2:webManagerUI

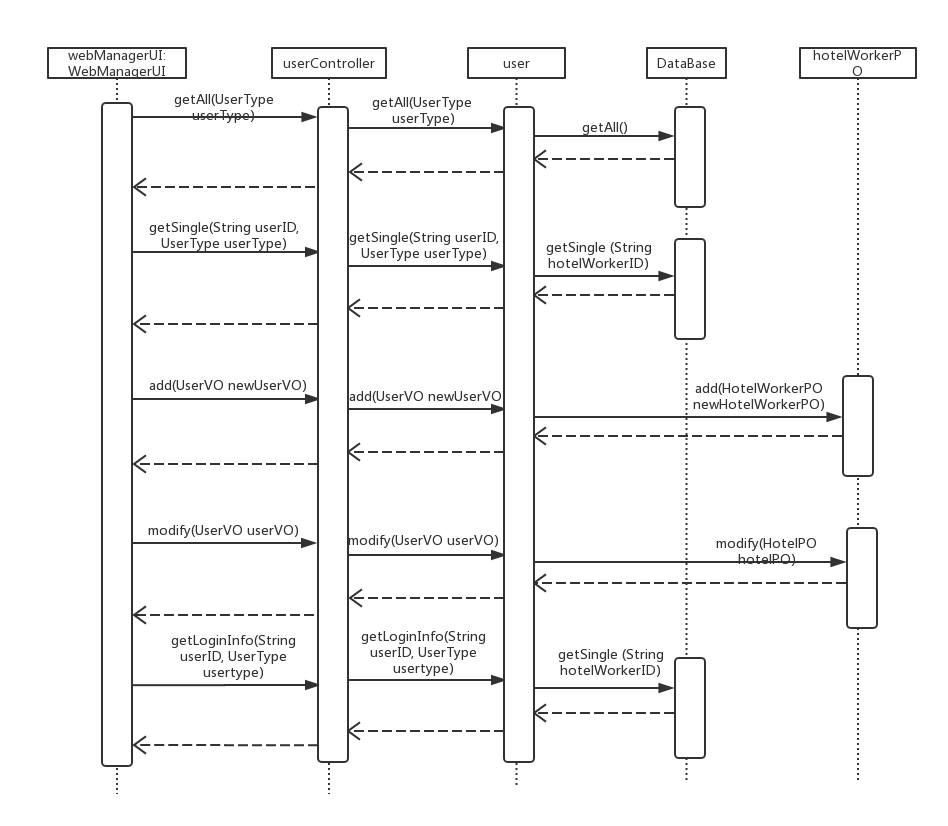


图3—user模块设计顺序图（hotelWorker）

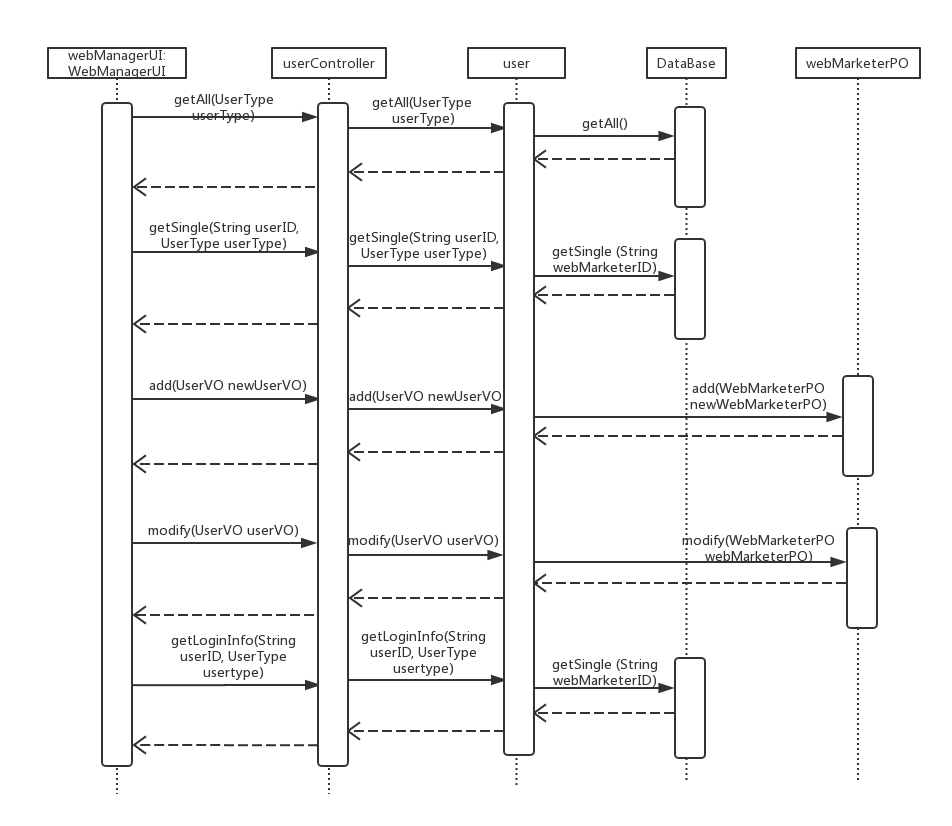


图4—user模块设计顺序图（webMarketer）

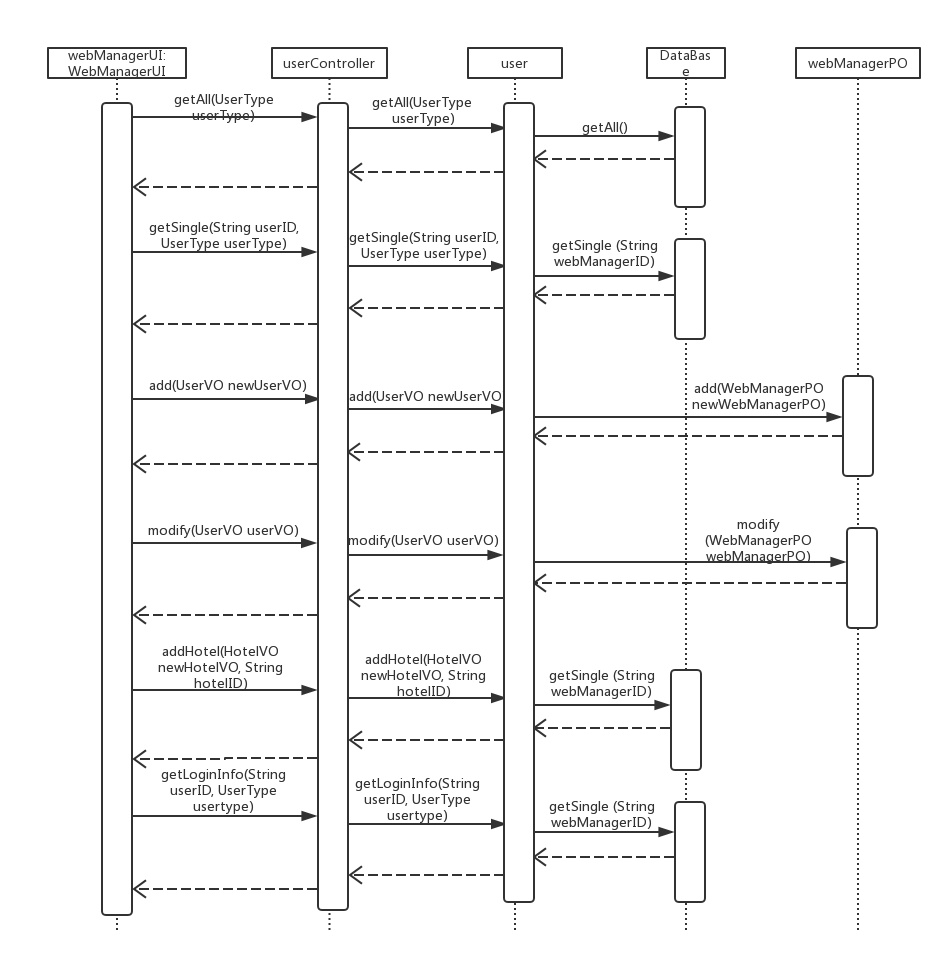


图5—user模块设计顺序图（webManager）



图6—user模块设计状态图

1. 业务逻辑层的设计原理