### logIn模块

1. 模块概述

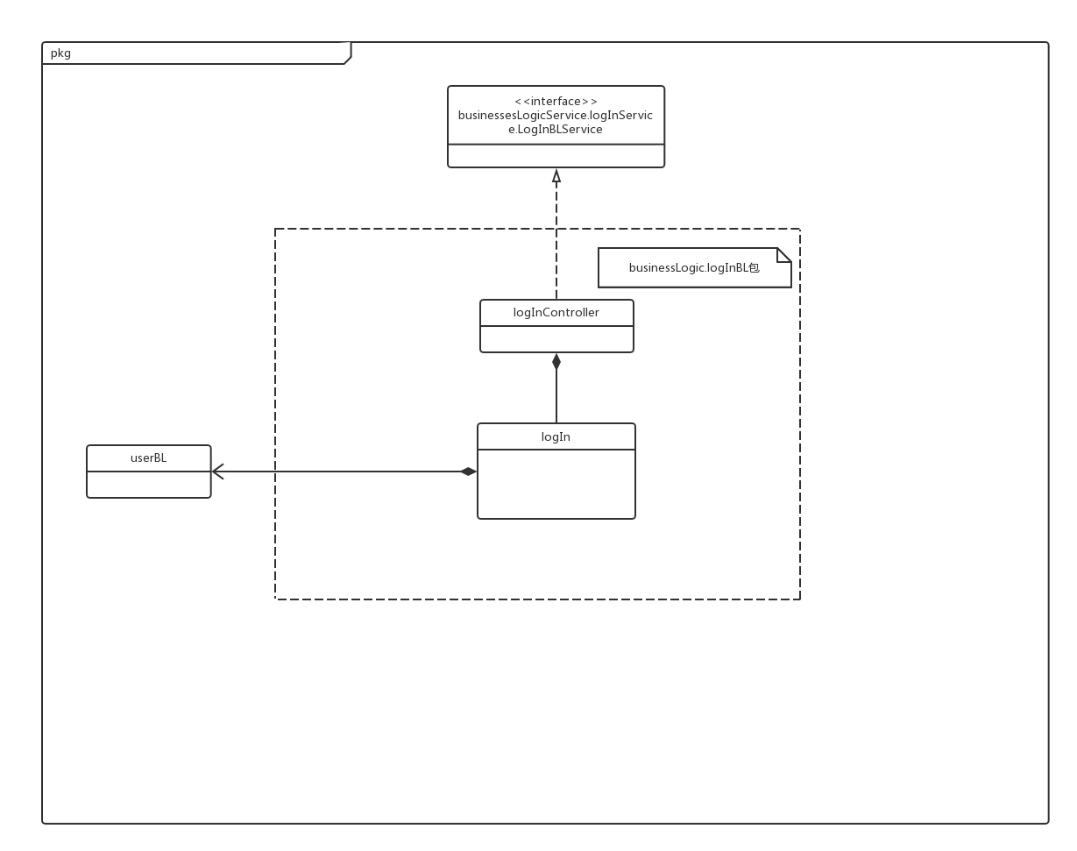
logIn模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

logIn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logInBLService接口。为了隔离业务逻辑职责和逻辑控制职责，添加logInController，这样logInController会将对用户登录和客户注册的业务逻辑处理委托给logIn对象。guestPO、hotelWorkerPO、webMarketer、webManager是作为账户信息的持久性对象被添加到设计模型中去的。

logInbl模块的设计如图所示.



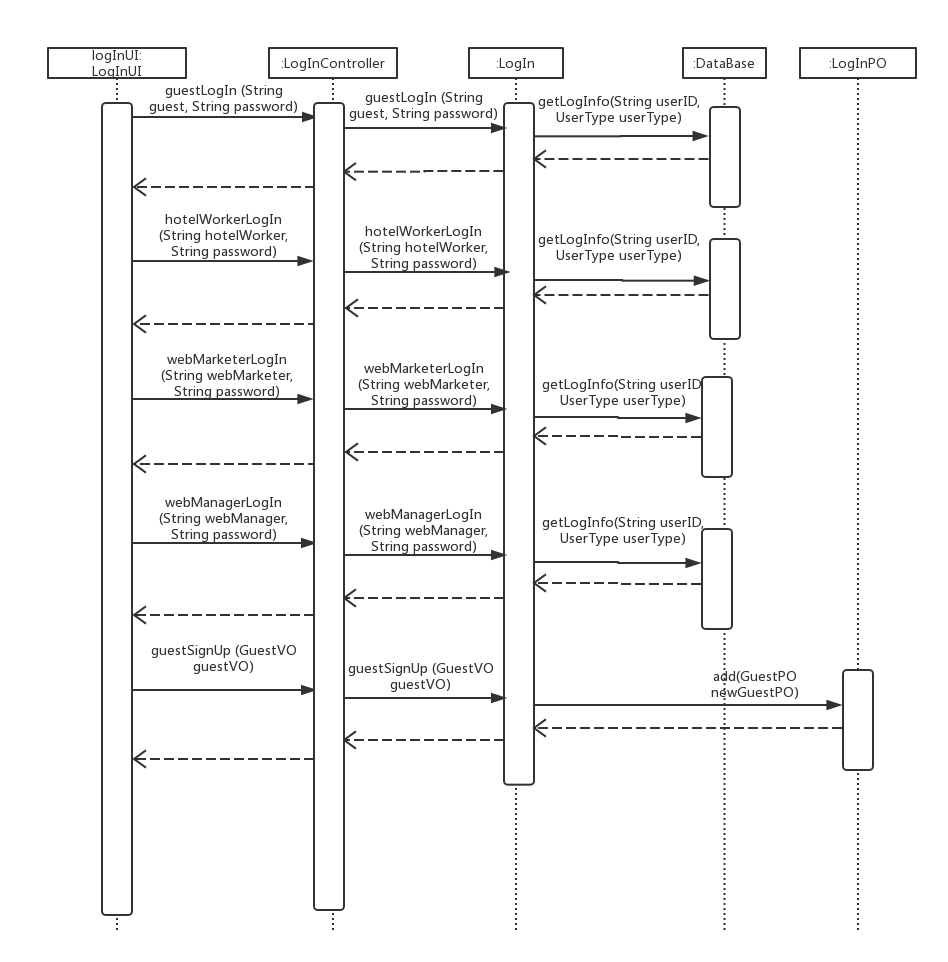
1. 模块内部类的接口规范
2. Controller类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| logInController.guestLogIn | 语法 | Public ResultMessage guestLogIn (String guest, String password) |
| 前置条件 | 客户输入客户编号和密码 |
| 后置条件 | 客户被认证并进入系统客户主界面 |
| logInController.hotelWorkerLogIn | 语法 | Public ResultMessage hotelWorkerLogIn (String hotelWorker, String password) |
| 前置条件 | 酒店工作人员输入编号和密码 |
| 后置条件 | 酒店工作人员被认证并进入系统酒店工作人员主界面 |
| logInController.webMarketerLogIn | 语法 | Public ResultMessage webMarketerLogIn (String webMarketer, String password) |
| 前置条件 | 网站营销人员输入编号和密码 |
| 后置条件 | 网站营销人员被认证并进入系统网站营销人员主界面 |
| logInController.webManagerLogIn | 语法 | Public ResultMessage webManagerLogIn (String webManager, String password) |
| 前置条件 | 网站管理人员输入编号和密码 |
| 后置条件 | 网站管理人员被认证并进入系统网站管理人员主界面 |
| logInController.guestSignUp | 语法 | Public ResultMessage guestSignUp (GuestVO guestVO) |
| 前置条件 | 客户信息符合规范要求 |
| 后置条件 | 系统返回是否成功添加客户；是则进入系统的客户主界面，否则提示添加失败 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| logIn.guestLogIn | 客户登录系统 | |
| logIn.hotelWorkerLogIn | 酒店工作人员登录系统 | |
| logIn.webMarketerLogIn | 网站营销人员登录系统 | |
| logIn.webManagerLogIn | 网站管理人员登录系统 | |
| logIn.guestSignUp | 客户注册 | |

1. Domain类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| logIn.guestLogIn | 语法 | Public ResultMessage guestLogIn (String guest, String password) |
| 前置条件 | 客户输入客户编号和密码 |
| 后置条件 | 客户被认证并进入系统客户主界面 |
| logIn.hotelWorkerLogIn | 语法 | Public ResultMessage hotelWorkerLogIn (String hotelWorker, String password) |
| 前置条件 | 酒店工作人员输入编号和密码 |
| 后置条件 | 酒店工作人员被认证并进入系统酒店工作人员主界面 |
| logIn.webMarketerLogIn | 语法 | Public ResultMessage webMarketerLogIn (String webMarketer, String password) |
| 前置条件 | 网站营销人员输入编号和密码 |
| 后置条件 | 网站营销人员被认证并进入系统网站营销人员主界面 |
| logIn.webManagerLogIn | 语法 | Public ResultMessage webManagerLogIn (String webManager, String password) |
| 前置条件 | 网站管理人员输入编号和密码 |
| 后置条件 | 网站管理人员被认证并进入系统网站管理人员主界面 |
| logIn.guestSignUp | 语法 | Public ResultMessage guestSignUp (GuestVO guestVO) |
| 前置条件 | 客户信息符合规范要求 |
| 后置条件 | 系统返回是否成功添加客户；是则进入系统的客户主界面，否则提示添加失败 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| User.getSingle(String userID,UserType userType) | 返回与该ID对应的user信息 | |
| User.add(UserVo newUserVo) | 新增该用户并持久化保存该用户的数据 | |
| User.getLogInfo(String userID, UserType userType) | 返回与该ID对应的user密码 | |

1. 业务逻辑层的动态模型



用户登录、客户注册顺序图

1. 业务逻辑层的设计原理

### credit模块

1)模块概述

credit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。

credit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。

2)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creditBLService接口。为了隔离业务逻辑职责和逻辑控制职责，添加creditController，这样creditController会将对用户登录和客户注册的业务逻辑处理委托给credit对象。creditPO是作为账户信息的持久性对象被添加到设计模型中去的。

creditbl模块的设计如图所示.



3)模块内部类的接口规范

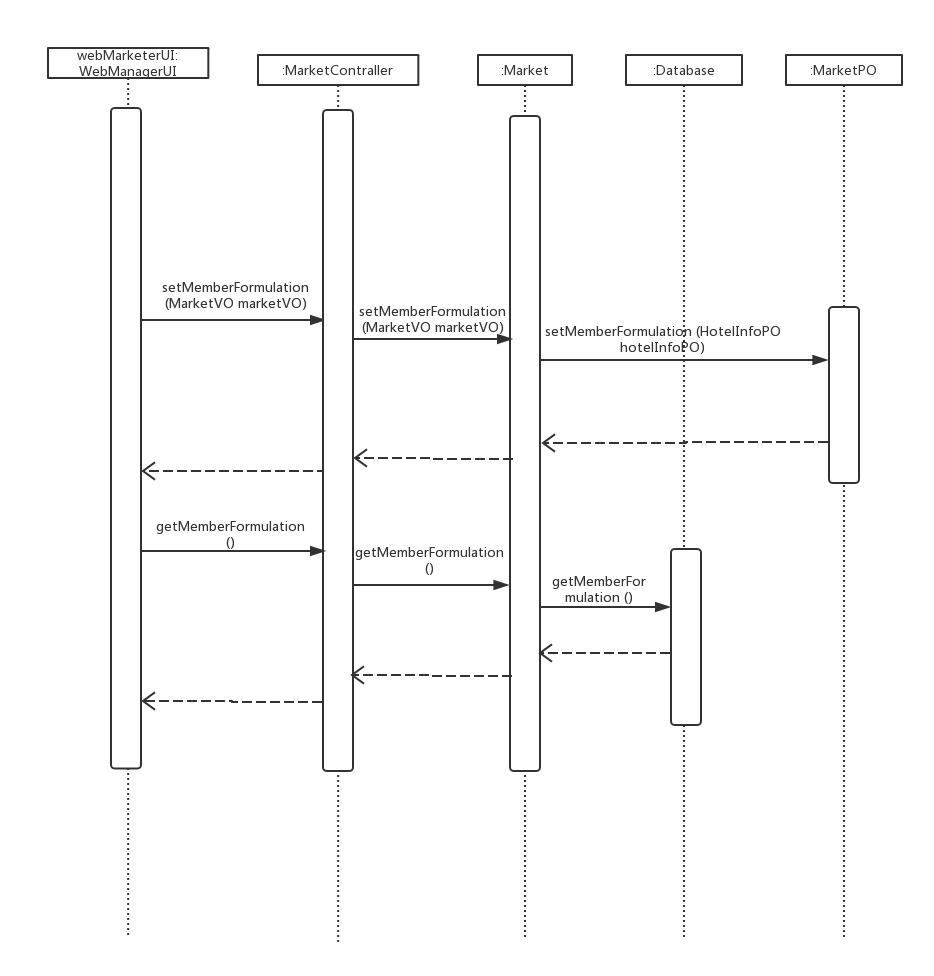
(1) Controller类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| creditController.charge | 语法 | Public ResultMessage charge(int chargeNum) |
| 前置条件 | 网站营销人员为客户信用充值 |
| 后置条件 | 系统为客户信用值充值并返回是否成功 |
| creditController.getBasicInfo | 语法 | Public CreditVO getBasicInfo (String ID) |
| 前置条件 | 用户（酒店工作人员或网站营销人员）确认更新促销策略，已知当前的促销策略的列表 |
| 后置条件 | 系统返回与该ID对应的包含信用值及信用记录的客户个人基本信息 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| credit.charge | 充值信用值 | |
| credit.getBasicInfo | 返回信用值信息 | |

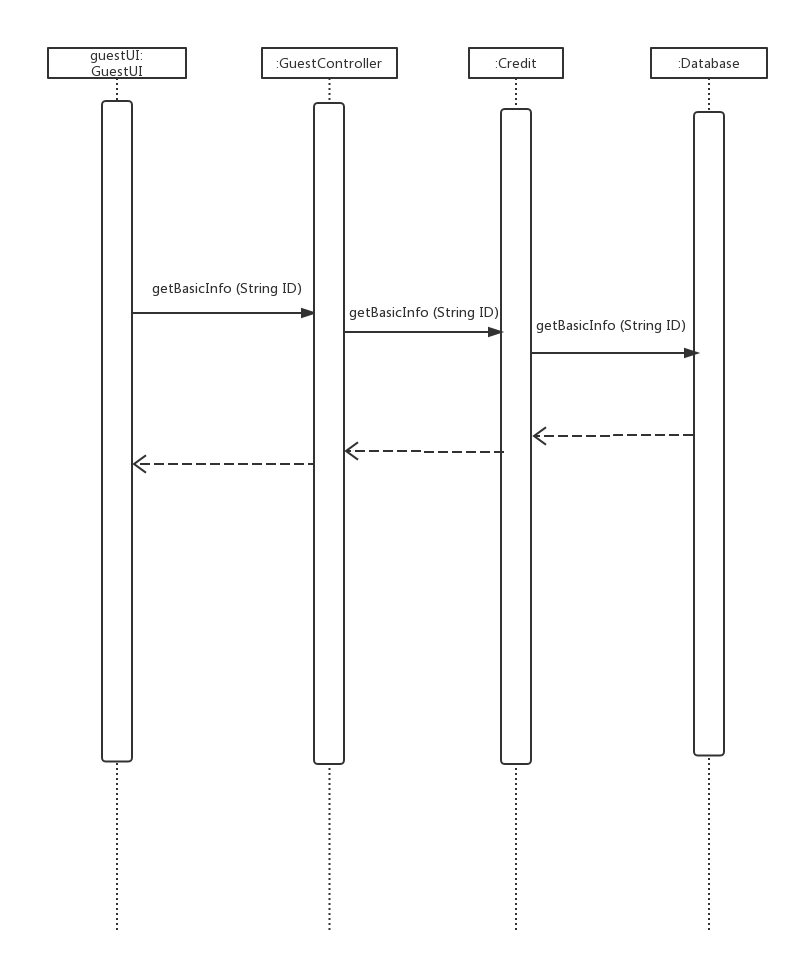
(2)Domain类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| credit.charge | 语法 | Public ResultMessage charge(int chargeNum) |
| 前置条件 | 网站营销人员为客户信用充值 |
| 后置条件 | 系统为客户信用值充值并返回是否成功 |
| credit.getBasicInfo | 语法 | Public CreditVO getBasicInfo (String ID) |
| 前置条件 | 用户（酒店工作人员或网站营销人员）确认更新促销策略，已知当前的促销策略的列表 |
| 后置条件 | 系统返回与该ID对应的包含信用值及信用记录的客户个人基本信息 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| User.getSingle(String userID, UserType userType) | 返回与该ID对应的用户信息 | |
| User.modifyCredit(float creditNum) | 返回客户信用值是否修改成功 | |
| market.getMemberFormulation() | 返回现有的会员等级制度信息 | |
| order.getAllGuestOrderGeneral (String guestID) | 返回与该ID对应的客户所有订单概况 | |

4) 业务逻辑层的动态模型



信用充值顺序图



信用查看顺序图

1. 业务逻辑层的设计原理

### market模块

1)模块概述

market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。

market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。

2)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marketBLService接口。为了隔离业务逻辑职责和逻辑控制职责，添加marketController，这样marketController会将对用户登录和客户注册的业务逻辑处理委托给market对象。marketPO是作为账户信息的持久性对象被添加到设计模型中去的。

marketbl模块的设计如图所示.



3)模块内部类的接口规范

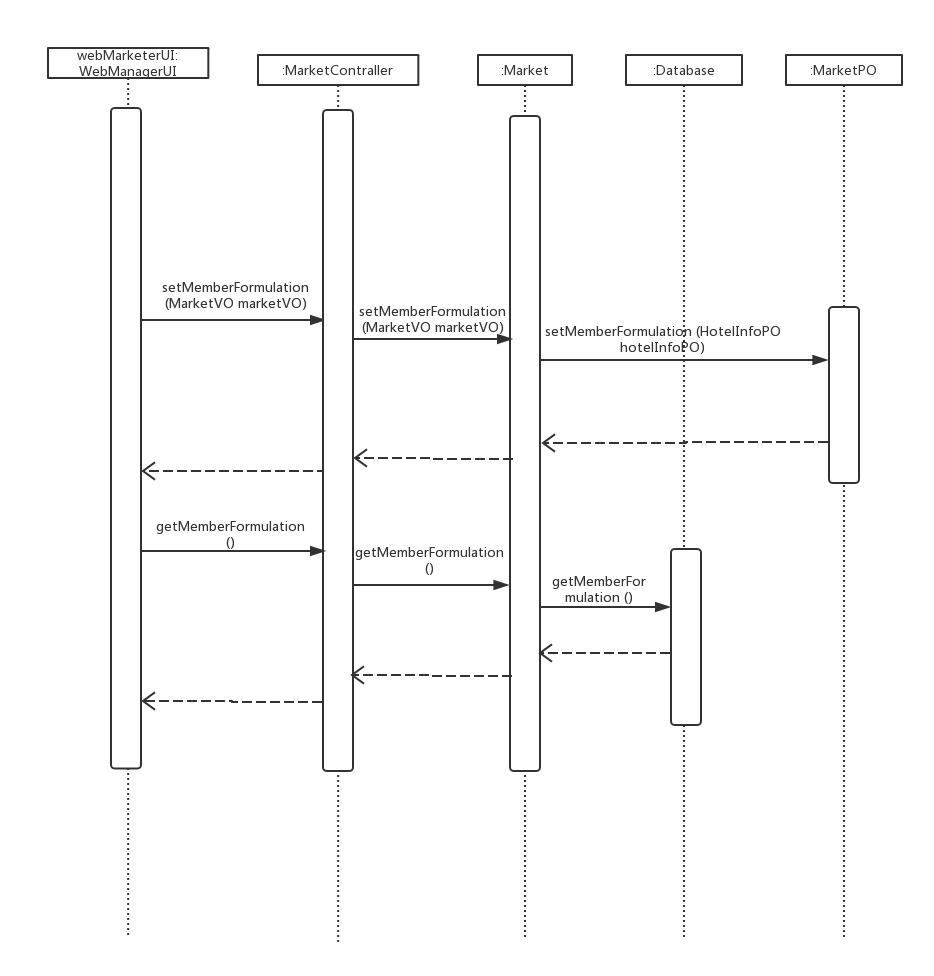
(1) Controller类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| marketController.getMemberFormulation | 语法 | Public MarketVO getMemberFormulation () |
| 前置条件 | 无 |
| 后置条件 | 返回现有的会员等级制度信息 |
| marketController.setMemberFormulation | 语法 | Public ResultMessage setMemberFormulation (MarketVO marketVO) |
| 前置条件 | 网站营销人员输入会员等级制度 |
| 后置条件 | 系统记录此会员等级制度信息 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| market.getMemberFormulation | 返回现有的会员等级制度信息 | |
| market.setMemberFormulation | 修改会员等级制度信息 | |

(2)Domain类的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| market.getMemberFormulation | 语法 | Public MarketVO getMemberFormulation () |
| 前置条件 | 无 |
| 后置条件 | 返回现有的会员等级制度信息 |
| market.setMemberFormulation | 语法 | Public ResultMessage setMemberFormulation (MarketVO marketVO) |
| 前置条件 | 网站营销人员输入会员等级制度 |
| 后置条件 | 系统记录此会员等级制度信息 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| marketDataService.getMemberFormulation | 返回现有的会员等级制度信息 | |
| marketDataService.setMemberFormulation(MarketVO marketVO) | 修改会员等级制度信息持久化对象，系统返回是否成功记录 | |

4) 业务逻辑层的动态模型



会员等级制定与查看顺序图

5) 业务逻辑层的设计原理