4.2.4 vipbl模块

1、模块概述

Vip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。

Vip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VipBLService接口。在业务逻辑层和数据层之间添加VipDataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加VipController，这样VipController会将对会员信息管理的业务逻辑处理委托给Vip对象。VipPO是作为账户信息的持久性对象被添加到设计模型中去的。

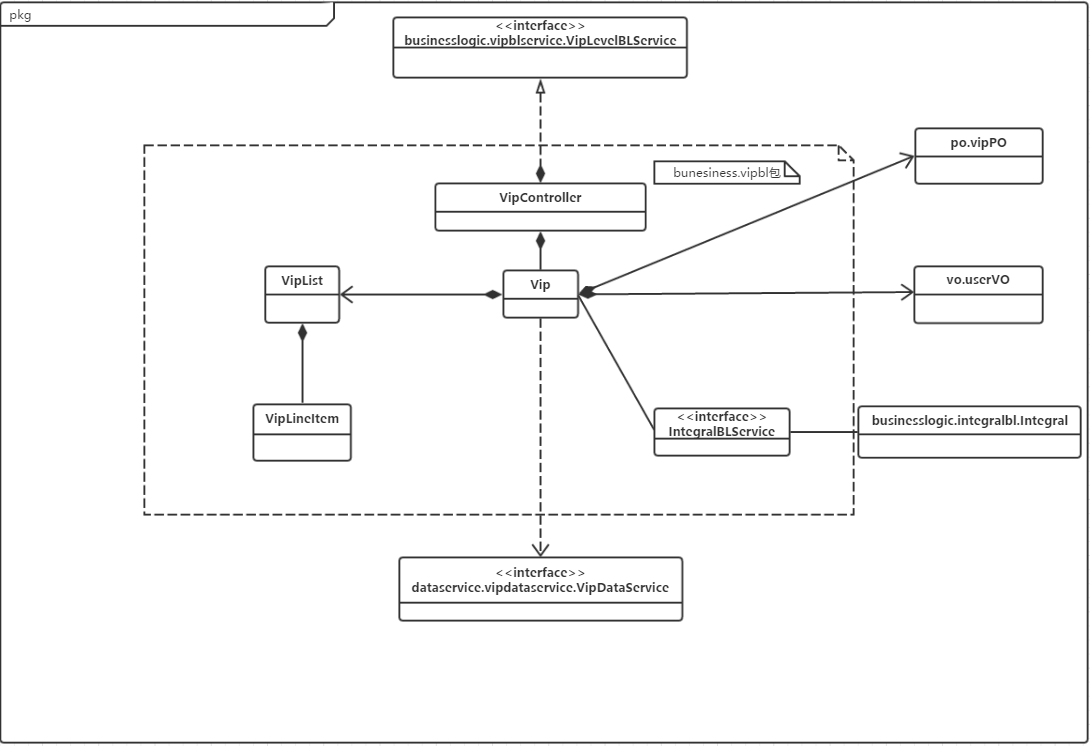


表4.2.4.1 vipbl模块各个类的设计

vipbl模块各个类的职责如表4.2.4.2所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| VipControl | 负责实现会员部分所需要的服务 |
| Vip | 会员的领域模型对象，拥有会员等级划分的积分要求信息，可以解决用户当前信用值对应的会员等级问题 |
| VipList | 负责保存会员等级对应信用值的相关信息 |
| VipLineItem | 会员等级对应信用值的条目信息 |

3、模块内部类的接口规范

**VipControl类的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| VipControl.searchLevelNeed | 语法 | public ArrayList<vipVO> searchLevelNeed() |
| 前置条件 | 已经创建一个vip领域对象，且输入格式符合规范 |
| 后置条件 | 调用vip类的searchLevelNeed方法，系统自动返回返回各个会员等级需要的信用值 |
| VipControl.searchLevel | 语法 | public int searchLevel(userVO user) |
| 前置条件 | 已经创建一个vip领域对象，且输入格式符合规范 |
| 后置条件 | 调用vip类的searchLevel方法，系统自动返回当前用户对应的会员等级 |
| VipControl.changeLevelNeed | 语法 | public void changeLevelNeed(vipVO vip) |
| 前置条件 | 已经创建一个vip领域对象，且输入格式符合规范 |
| 后置条件 | 调用vip类的changeLevelNeed方法，系统自动修改会员等级要求，并且持久化更新vipVO |
| VipControl. addLevel | 语法 | public void addLevel (vipVO vip) |
| 前置条件 | 已经创建一个vip领域对象，且输入格式符合规范 |
| 后置条件 | 调用vip类的addLevel方法，系统自动添加一个会员等级，并且持久化更新vipVO |
| VipControl. subLevel | 语法 | public void subLevel (vipVO vip) |
| 前置条件 | 已经创建一个vip领域对象，且输入格式符合规范 |
| 后置条件 | 调用vip类的subLevel方法，系统自动删除一个会员等级，并且持久化更新vipVO |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| Vip.searchLevelNeed | 获得会员等级和信用值的对应关系 | |
| Vip.searchLevel | 获得对应的会员等级 | |
| Vip.changeLevelNeed | 修改会员等级和信用值至今的对应关系 | |
| Vip.addLevel | 增加一个会员等级 | |
| Vip.subLevel | 删除一个会员等级 | |

**Vipbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Vip.searchLevelNeed | 语法 | public ArrayList<vipVO> searchLevelNeed() |
| 前置条件 | 用户要求查看会员等级需要的信用值 |
| 后置条件 | 系统返回各个会员等级需要的信用值 |
| Vip.searchLevel | 语法 | public int searchLevel(userVO user) |
| 前置条件 | 用户获得当前积分对应的会员等级 |
| 后置条件 | 系统根据用户的信息，反馈当前用户的会员等级 |
| Vip.changeLevelNeed | 语法 | public void changeLevelNeed(vipVO vip) |
| 前置条件 | 用户修改相应会员等级需要的信用值 |
| 后置条件 | 系统将当前等级需要的信用值修改为integral |
| Vip.addLevel | 语法 | public void addLevel (vipVO vip) |
| 前置条件 | 用户增加相应会员等级需要的信用值 |
| 后置条件 | 系统将当前增加会员等级 |
| Vip.subLevel | 语法 | public void subLevel (vipVO vip) |
| 前置条件 | 用户删除相应会员等级需要的信用值 |
| 后置条件 | 系统删除一个会员等级 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| Integral.getIntegral | 获取对应客户的信用值 |
| VipDataService.insert | 添加一个持久化的会员信息 |
| VipDataService.delete | 删除一个持久化的会员信息 |
| VipDataService.update | 更新一个持久化的会员信息 |
| VipDataService.findByLevel | 按照等级找到一个持久化的会员信息 |
| VipDataService.Show | 展示所有的持久化的会员信息 |

4.业务逻辑层的动态模型

在会员业务中，分为会员等级需要的信用值查看，用户当前会员等级，修改会员等级需要的信用值功能。图4.2.4.3表示了得到会员等级和信用值对应关系的相关对象之间的协作。

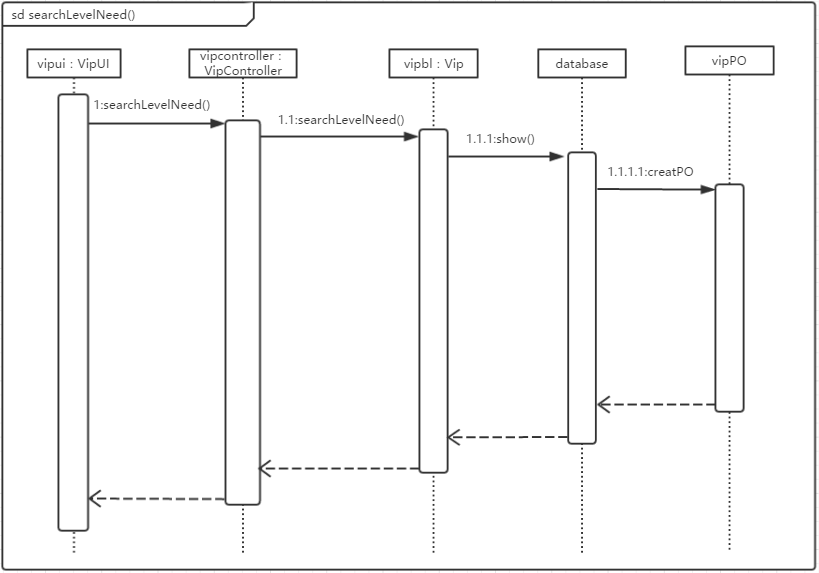


图4.2.4.3查询会员等级和信用值对应关系

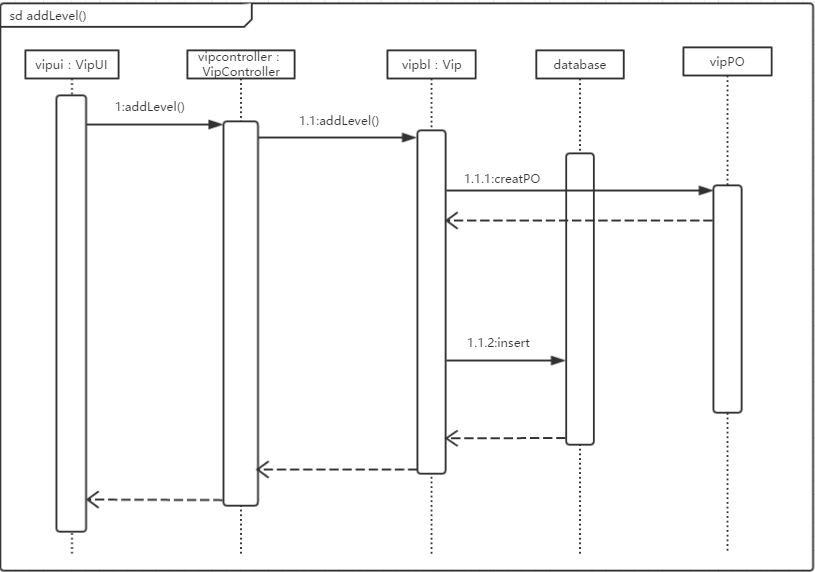
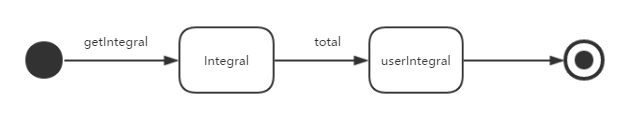


图4.2.4.3增加一个会员等级

下图所示的状态图描述了Vip对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。



4.2.5 integralbl模块

1、模块概述

Integral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。

Integral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IntegralBLService接口。在业务逻辑层和数据层之间添加IntegralDataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加IntegralController，这样IntegralController会将对信用值操作的业务逻辑处理委托给Integral对象。IntegralPO是作为信用值的持久性对象被添加到设计模型中去的。

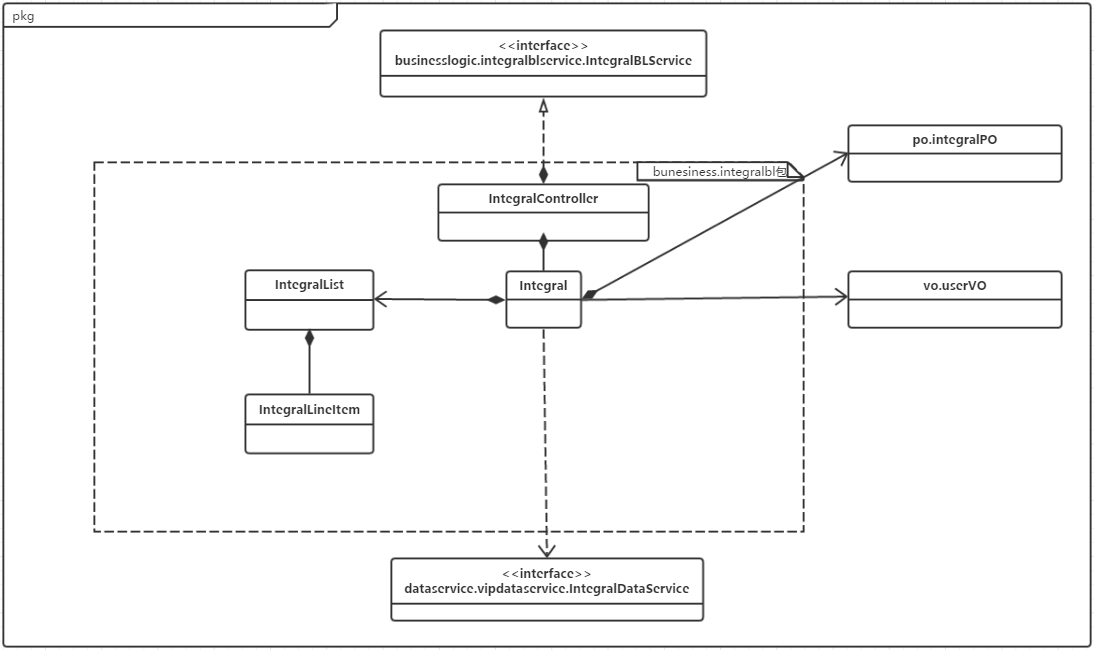


表4.2.5.1 integralbl模块各个类的设计

integralbl模块各个类的职责如表4.2.5.2所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| IntegralControl | 负责实现信用值部分所需要的服务 |
| Integral | 信用值的领域模型对象，拥有对应用户的信用值信息，可以解决用户信用值变化，修改问题 |
| IntegralList | 负责保存信用值的相关信息 |
| IntegralLineItem | 信用值的条目信息 |

3、模块内部类的接口规范

**IntegralControl类的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| IntegralControl.getIntegral | 语法 | public IntegralVO getIntegral(userVO client) |
| 前置条件 | 已经创建一个integral领域对象，且输入格式符合规范 |
| 后置条件 | 调用integral类的getIntegral方法，系统返回当前用户信用值 |
| IntegralControl.addIntegral | 语法 | public void addIntegral(userVO client，int value) |
| 前置条件 | 已经创建一个integral领域对象，且输入格式符合规范 |
| 后置条件 | 调用integral类的addIntegral方法，系统将要增加的信用值增加到该用户身上，并且持久化更新integralVO |
| IntegralControl.subIntegral | 语法 | public void subIntegral(userVO client，int value) |
| 前置条件 | 已经创建一个integral领域对象，且输入格式符合规范 |
| 后置条件 | 调用integral类的subIntegral方法，系统将要减少的信用值从该用户减少，并且持久化更新integralVO |
| IntegralControl.changeIntegral | 语法 | public void changeIntegral(userVO client，int value) |
| 前置条件 | 已经创建一个integral领域对象，且输入格式符合规范 |
| 后置条件 | 调用integral类的changeIntegral方法，系统该用户信用值修改为规定数值，并且持久化更新integralVO |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| Integral.getIntegral(userVO client) | 得到信用值单一化持久对象 | |
| Integral.addIntegral(userVO client，int value) | 增加信用值数额 | |
| Integral.subIntegral(userVO client，int value) | 减少信用值数额 | |
| Integral.changeIntegral(userVO client，int value) | 修改信用值数额 | |

**Integralbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Integral.getIntegral | 语法 | public IntegralVO getIntegral(userVO client) |
| 前置条件 | 用户要获得当前的信用值 |
| 后置条件 | 系统获取当前用户的信用值，返回回来 |
| Integral.addIntegral | 语法 | public void addIntegral(userVO client，int value) |
| 前置条件 | 用户或系统要对当前客户的信用值进行增加 |
| 后置条件 | 系统将增加的数值增加到当前客户的信用值上 |
| Integral.subIntegral | 语法 | public void subIntegral(userVO client，int value) |
| 前置条件 | 用户或系统要对当前客户的信用值进行减少 |
| 后置条件 | 系统将增加的数值从当前客户的信用值减少 |
| Integral.changeIntegral | 语法 | public void changeIntegral(userVO client，int value) |
| 前置条件 | 用户或系统要对当前客户的信用值进行修改 |
| 后置条件 | 系统将用户信用值修改为当前数值 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| IntegralDataService.insert | 增加一个持久化的信用值 |
| IntegralDataService.delete | 删除一个持久化的信用值 |
| IntegralDataService.update | 更新一个持久化的信用值 |
| IntegralDataService.findByName | 根据名字找到一个持久化的信用值 |
| IntegralDataService.Show | 展示所有的持久化的信用值 |

4.业务逻辑层的动态模型

在信用值业务中，分为信用值修改，信用值查看，信用值增加功能。图4.2.5.3表示了信用值修改的相关对象之间的协作。

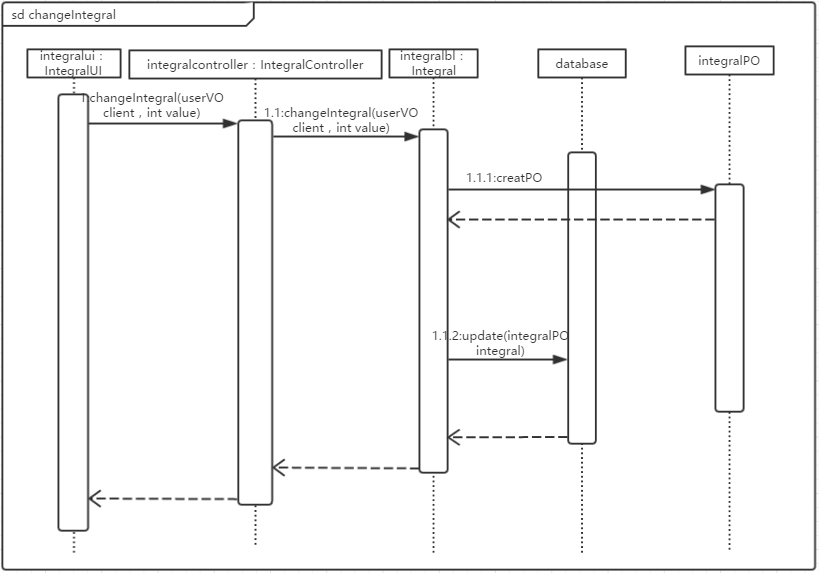
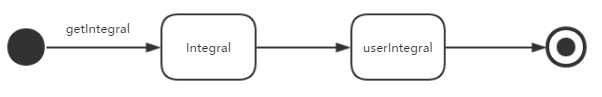


图4.2.5.3信用值修改

下图所示的状态图描述了Integral对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。



4.2.6 promotionblbl模块

1、模块概述

Promotion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。

Promotion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PromotionBLService接口。在业务逻辑层和数据层之间添加PromotionDataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加PromotionHotelController，PromotionWebController，PromotionValueController，这样PromotionHotelController会将对信用值操作的业务逻辑处理委托给PromotionHotel对象，PromotionWebController会将对信用值操作的业务逻辑处理委托给PromotionWeb对象，PromotionValueController会将对信用值操作的业务逻辑处理委托给PromotionValue对象。PromotionPO是作为促销策略的持久性对象被添加到设计模型中去的。

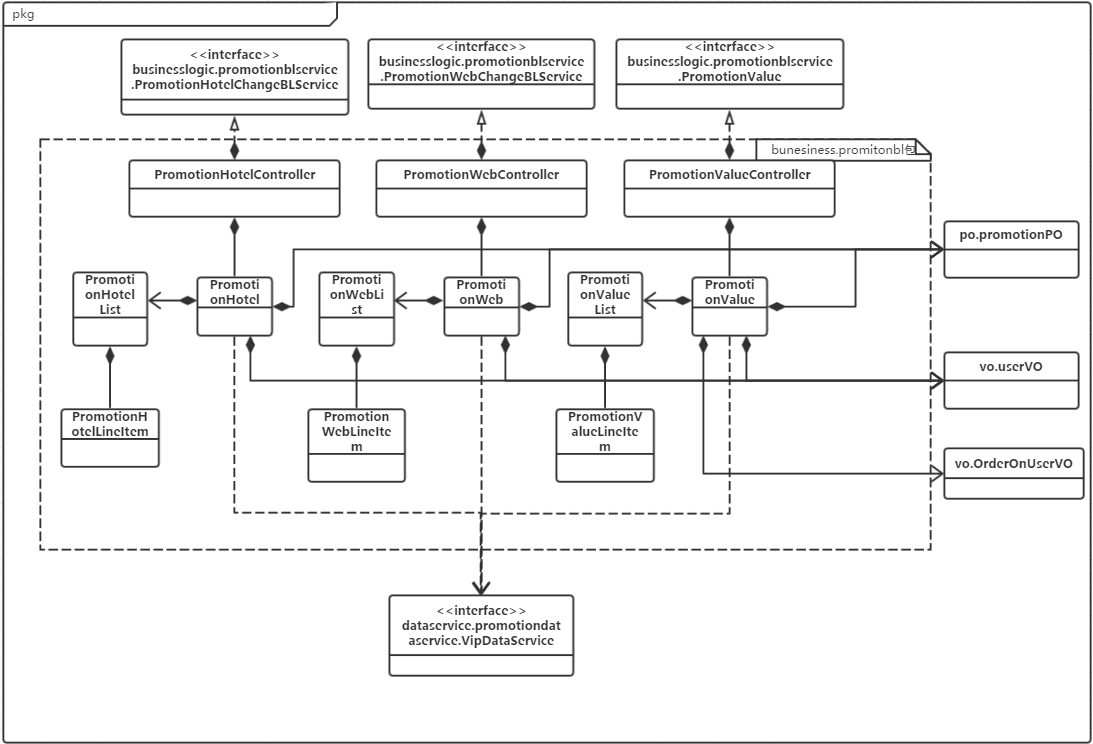


表4.2.6.1 promotionbl模块各个类的设计

promotionbl模块各个类的职责如表4.2.6.2所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| promotionHotelControl | 负责实现酒店促销策略部分需要的服务 |
| PromotionHotel | 酒店促销策略的领域模型对象，拥有策略适用用户种类，促销策略降价比例，促销策略有效时间等信息，可以解决酒店促销策略的修改，增加，删除问题。 |
| PromotionHotelList | 负责保存酒店促销策略的相关信息 |
| PromotionHotelLineItem | 酒店促销策略的条目信息 |
| promotionWebControl | 负责实现网站促销策略部分需要的服务 |
| PromotionWeb | 网站促销策略的领域模型对象，拥有策略适用用户种类，促销策略降价比例，促销策略有效时间等信息，可以解决网站促销策略的修改，增加，删除问题。 |
| PromotionWebList | 负责保存网站促销策略的相关信息 |
| PromotionWebLineItem | 网站促销策略的条目信息 |
| promotionValueControl | 负责实现订单经过促销策略之后的订单价值部分所需要的服务 |
| PromotionValue | 订单价值计算的领域模型对象，拥有各种促销策略适用对象信息，能够解决促销策略匹配问题 |
| PromotionValueList | 负责保存订单价值计算的相关信息 |
| PromotionValueLineItem | 订单价值计算的条目信息 |

3、模块内部类的接口规范

**promotionHotelControl类的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PromotionHoelControl.changeBirthCut | 语法 | public void changeBirthCut(promotionPO po) |
| 前置条件 | 已经创建一个PromotionHoel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHoel类的changeBirthCut方法，系统将生日促销策略进行修改，并且持久化更新promotionPO |
| PromotionHoelControl.changeOverCut | 语法 | public void changeOverCut(promotionPO po) |
| 前置条件 | 已经创建一个PromotionHoel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHoel类的changeOverCut方法，系统将三间以上促销策略进行修改，并且持久化更新promotionPO |
| PromotionHoelControl.changeJoinCut | 语法 | public void changeJoinCut (promotionPO po) |
| 前置条件 | 已经创建一个PromotionHoel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHoel类的changeJoinCut方法，系统将合作企业促销策略进行修改，并且持久化更新promotionPO |
| PromotionHoelControl.changeHotelCustomCut | 语法 | public void changeHotelCustomCut (promotionPO po) |
| 前置条件 | 已经创建一个PromotionHoel领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionHoel类的changeHotelCustomCut方法，系统将自定义促销策略进行修改，并且持久化更新promotionPO |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| PromotionHoel.changeBirthCut | 修改会员生日优惠政策 | |
| PromotionHoel.changeOverCut | 修改三间以上优惠政策 | |
| PromotionHoel.changeJoinCut | 修改合作企业优惠政策 | |
| PromotionHoel.changeHotelCustomCut | 修改酒店自定义优惠政策 | |

**promotionWebControl类的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PromotionWebControl.changeLevelCut | 语法 | public void changeLevelCut(promotionPO po) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的changeLevelCut方法，系统将会员等级优惠策略进行修改，并且持久化更新promotionPO |
| PromotionWebControl.changeCircleCut | 语法 | public void changeCircleCut(promotionPO po) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的changeCircleCut方法，系统将特定商圈优惠策略进行修改，并且持久化更新promotionPO |
| PromotionWebControl.changeWebCustomCut | 语法 | public void changeWebCustomCut(promotionPO po) |
| 前置条件 | 已经创建一个PromotionWeb领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionWeb类的changeWebCustomCut方法，系统将自定义优惠策略进行修改，并且持久化更新promotionPO |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| PromotionWeb.changeLevelCut | 修改会员等级优惠策略 | |
| PromotionWeb.changeCircleCut | 修改会员特定商圈优惠策略 | |
| PromotionWeb.changeWebCustomCut | 修改网站自定义优惠策略 | |

**promotionValueControl类的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PromotionValueControl.getValue | 语法 | public promotionVO getValue(userVO user, OrderOnUserVO orderVO) |
| 前置条件 | 已经创建一个PromotionValue领域对象，且输入格式符合规范 |
| 后置条件 | 调用PromotionValue类的getValue方法，系统自动计算订单价值，并且持久化更新promotionPO |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| PromotionValue.getValue | 获得该订单经过打折之后的信息 | |

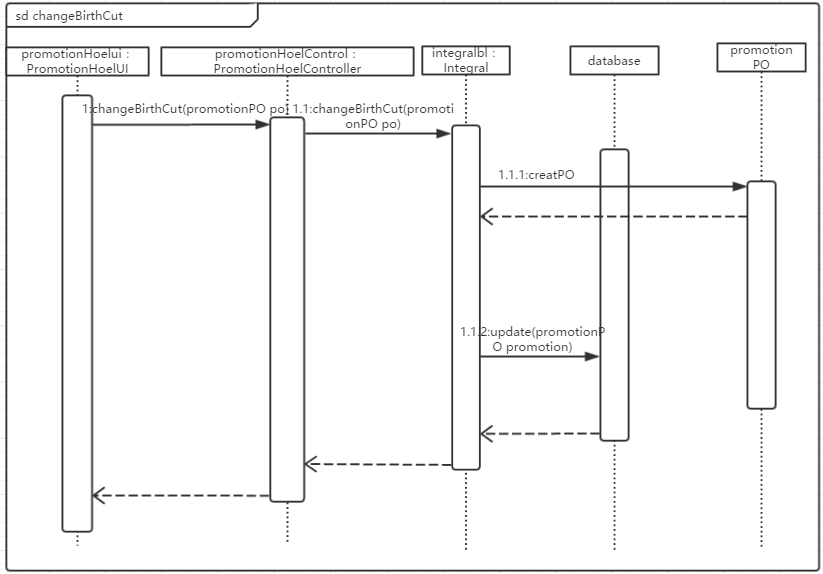
**promotionbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PromotionHoel.changeBirthCut | 语法 | public void changeBirthCut(promotionPO po) |
| 前置条件 | 酒店工作人员要修改酒店会员生日促销策略 |
| 后置条件 | 系统将酒店会员生日促销策略降价比例修改为ratio |
| PromotionHoel.changeOverCut | 语法 | public void changeOverCut(promotionPO po) |
| 前置条件 | 酒店工作人员要修改酒店三间以上优惠策略 |
| 后置条件 | 系统将酒店三间以上促销策略降价比例修改为ratio |
| PromotionHoel.changeJoinCut | 语法 | public void changeJoinCut (promotionPO po) |
| 前置条件 | 酒店工作人员要修改酒店合作企业客户优惠策略 |
| 后置条件 | 系统将酒店合作企业促销策略降价比例修改为ratio |
| PromotionHoel.changeHotelCustomCut | 语法 | public void changeHotelCustomCut (promotionPO po) |
| 前置条件 | 酒店工作人员要修改酒店自定义优惠策略 |
| 后置条件 | 系统将酒店自定义优惠策略修改为当前设定 |
| PromotionWeb.changeLevelCut | 语法 | public void changeLevelCut(promotionPO po) |
| 前置条件 | 网站营销人员修改网站会员等级优惠 |
| 后置条件 | 系统将网站会员等级优惠修改为ratio |
| PromotionWeb.changeCircleCut | 语法 | public void changeCircleCut(promotionPO po) |
| 前置条件 | 网站营销人员修改网站vip商圈专属折扣优惠 |
| 后置条件 | 系统将网站vip商圈专属优惠折扣修改为ratio |
| PromotionWeb.changeWebCustomCut | 语法 | public void changeWebCustomCut(promotionPO po) |
| 前置条件 | 网站营销人员修改网站自定义促销策略 |
| 后置条件 | 系统将网站自定义促销策略修改为当前设定 |
| PromotionValue.getValue | 语法 | public promotionVO getValue(userVO user, OrderOnUserVO orderVO) |
| 前置条件 | 订单要利用促销策略计算订单价值 |
| 后置条件 | 系统返回订单经过促销策略打折后的价值 |

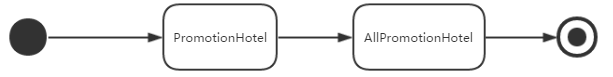
|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| PromotionDataService.insert | 怎加一个促销策略 |
| PromotionDataService.delete | 删除一个促销策略 |
| PromotionDataService.update | 更新一个促销策略 |
| PromotionDataService.findByName | 根据促销策略种类找到一个促销策略 |
| PromotionDataService.Show | 展示所有的促销策略 |

4.业务逻辑层的动态模型

在促销策略业务中，分为酒店促销策略修改，网站促销策略修改，订单价值计算功能。图4.2.6.3表示了酒店促销会员生日策略修改的相关对象之间的协作。



下图所示的状态图描述了PromotionHotel对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。



4.3.4 VipDataService模块

（1）模块概述

VipDataService模块负责保存会员的相关信息，供读取、保存、修改。

（2）模块内部接口规范

|  |  |  |
| --- | --- | --- |
| 需要的服务（需接口） | | |
| VipDataService.insert | 语法 | public void insert(vipPO vip) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中不存在 |
| 后置条件 | 在数据库中增加一个 po 记录 |
| VipDataService.delete | 语法 | public void delete(vipPO vip) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中删除一个 po 记录 |
| VipDataService.update | 语法 | public void update(vipPO vip) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中更新一个 po 记录 |
| VipDataService.findByLevel | 语法 | public vipPO findByLevel(int level) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按等级返回相应的vipPO结果 |
| VipDataService.Show | 语法 | public ArrayList<vipPO> show() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存 PO |

4.3.5 IntegralDataService模块

（1）模块概述

IntegralDataService模块负责保存信用值的相关信息，供读取、保存、修改。

（2）模块内部接口规范

|  |  |  |
| --- | --- | --- |
| 需要的服务（需接口） | | |
| IntegralDataService.insert | 语法 | public void insert(integralPO integral) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中不存在 |
| 后置条件 | 在数据库中增加一个 po 记录 |
| IntegralDataService.delete | 语法 | public void delete(integralPO integral) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中删除一个 po 记录 |
| IntegralDataService.update | 语法 | public void update(integralPO integral) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中更新一个 po 记录 |
| IntegralDataService.findByName | 语法 | public integralPO findByLevel(String name) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按等级返回相应的integralPO结果 |
| IntegralDataService.Show | 语法 | public ArrayList<integralPO> show() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存 PO |

4.3.6 PromotionDataService模块

（1）模块概述

PromotionDataService模块负责保存促销策略的相关信息，供读取、保存、修改。

（2）模块内部接口规范

|  |  |  |
| --- | --- | --- |
| 需要的服务（需接口） | | |
| PromotionDataService.insert | 语法 | public void insert(promotionPO promotion) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中不存在 |
| 后置条件 | 在数据库中增加一个 po 记录 |
| PromotionDataService.delete | 语法 | public void delete(promotionPO promotion) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中删除一个 po 记录 |
| PromotionDataService.update | 语法 | public void update(promotionPO promotion) throws RemoteException |
| 前置条件 | 同样 ID 的 po 在 Mapper 中存在 |
| 后置条件 | 在数据库中更新一个 po 记录 |
| PromotionDataService.findByName | 语法 | public promotionPO findByLevel(String name) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按等级返回相应的promotionPO结果 |
| PromotionDataService.Show | 语法 | public ArrayList<promotionPO> show() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有的库存 PO |