Promotionbl模块

1. 模块概述

Promotionbl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。

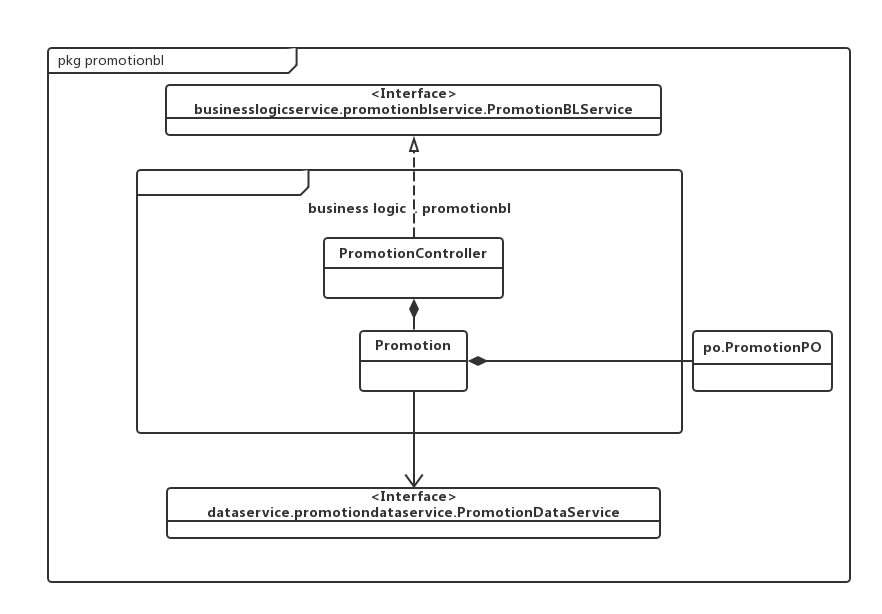
Promotionbl模块的职责及接口参见软件系统结构描述文档表5-3-2-1。

Promotionbl模块承担促销策略方面的功能，包括制定酒店的促销策略，制定网站的促销策略，修改促销策略等。

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businesslogicservice.promotionblservice.PromotionBLService接口。在业务逻辑层和数据层之间添加dataservice.promotiondataservice. PromotionDataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加PromotionController，这样PromotionController会将对促销策略的业务逻辑处理委托给Promotion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。

Promotionbl模块的设计如图所示。



图：promotionbl模块各个类的设计

Promotionbl模块各个类的职责如表所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| PromotionController | 负责实现促销策略管理所需要的服务 |
| Promotion | 促销策略管理的领域模型对象，拥有促销策略管理所持有的促销策略信息，可以帮助完成促销策略管理需要的服务。 |

1. 模块内部类的接口规范

**promotioncontroller模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| PromotionController.hotelAdd | 语法 | | public ResultMessage hotelAdd (PromotionVO vo) |
| 前置条件 | | 需创建一个Promotion领域对象，并且分类名称符合输入规则 |
| 后置条件 | | 调用Promotion领域对象的hotelAdd方法 |
| PromotionController.webAdd | 语法 | | public ResultMessage webAdd (PromotionVO vo) |
| 前置条件 | | 需创建一个Promotion领域对象，并且分类名称符合输入规则 |
| 后置条件 | | 调用Promotion领域对象的webAdd方法 |
| PromotionController.webLook | 语法 | | public ArrayList<PromotionVO> webLook () |
| 前置条件 | | 已创建一个Promotion领域对象，并且分类名称符合输入规则 |
| 后置条件 | | 调用Promotion领域对象的webLook方法 |
| PromotionController.hotelLook | 语法 | | public ArrayList<PromotionVO> hotelLook (String hotelID) |
| 前置条件 | | 已创建一个Promotion领域对象，并且分类名称符合输入规则 |
| 后置条件 | | 调用Promotion领域对象的hotelLook方法 |
| PromotionController.cancel | 语法 | | public ResultMessage cancel(String promotionID) |
| 前置条件 | | 已创建一个Promotion领域对象，并且分类名称符合输入规则 |
| 后置条件 | | 调用Promotion领域对象的cancel方法 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| Promotion.hotelAdd | | 增加一个酒店促销策略对象 | |
| Promotion.webAdd | | 增加一个网站促销策略对象 | |
| Promotion.webLook | | 查看一个网站促销策略对象 | |
| Promotion.hotelLook | | 查看一个酒店促销策略对象 | |
| Promotion.cancel | | 取消一条促销策略对象 | |

**promotion模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Promotion.hotelAdd | 语法 | public ResultMessage hotelAdd (PromotionVO vo) |
| 前置条件 | 输入信息符合输入规则 |
| 后置条件 | 系统保存并更新了当前酒店促销策略 |
| Promotion.webAdd | 语法 | public ResultMessage webAdd (PromotionVO vo) |
| 前置条件 | 输入信息符合输入规则 |
| 后置条件 | 系统保存并更新了网站的促销策略 |
| Promotion.webLook | 语法 | public ArrayList<PromotionVO> webLook () |
| 前置条件 | 无 |
| 后置条件 | 系统返回网站促销策略 |
| Promotion.hotelLook | 语法 | public ArrayList<PromotionVO> hotelLook (String hotelID) |
| 前置条件 | 无 |
| 后置条件 | 系统根据输入的酒店名称查找并返回酒店促销策略 |
| Promotion.cancel | 语法 | public ResultMessage cancel(String promotionID) |
| 前置条件 | 无 |
| 后置条件 | 系统应该根据输入的促销策略编号查找并取消当前策略 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| DataFactory.getPromotionData() | | 得到Promotion的数据的服务的引用 |
| PromotionDataService.hotelAdd(PromotionPO po) | | 新增酒店促销策略PromotionPO的单一持久化对象 |
| PromotionDataService.hotelLook(String name,PromotionPO po) | | 根据酒店信息在数据库中查找对应的酒店，并返回相应的酒店促销策略PromotionPO结果 |
| PromotionDataService.webAdd(PromotionPO po) | | 新增网站促销策略PromotionPO的单一持久化对象 |
| PromotionDataService.webLook(String name,PromotionPO po) | | 在数据库中查找，并返回网站的促销策略PromotionPO结果 |
| PromotionDataService.levelMake (ArrayList<Integer> levelList) | | 更新酒店会员等级MemberPO单一持久化对象 |
| PromotionDataService.cancel(String promotionID) | | 删除数据库中PromotionPO单一持久化对象 |

1. 业务逻辑层的动态模型

在促销策略管理业务中，分为新增酒店促销策略，新增网站促销策略，查看酒店促销策略，查看网站促销策略，取消制定促销策略等业务。

图 表明了制定酒店促销策略业务逻辑处理的相关对象之间的协作。

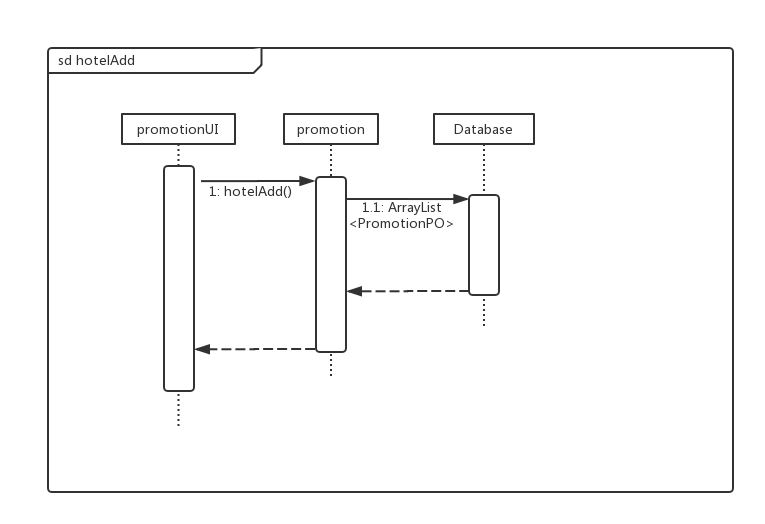
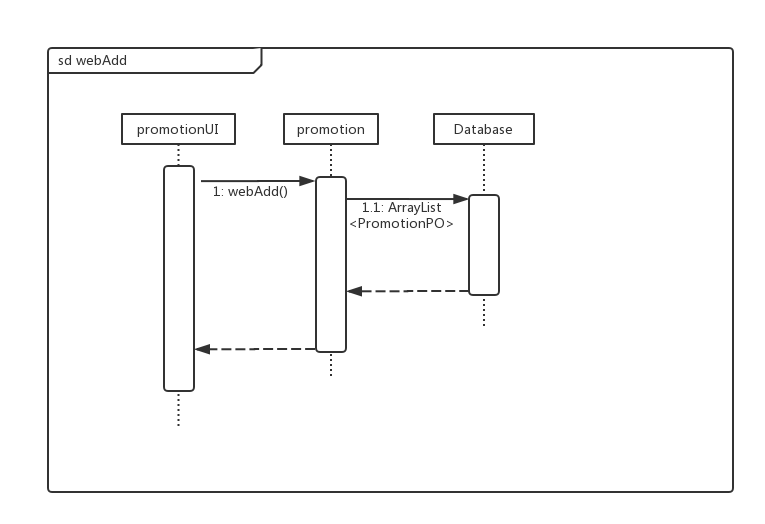
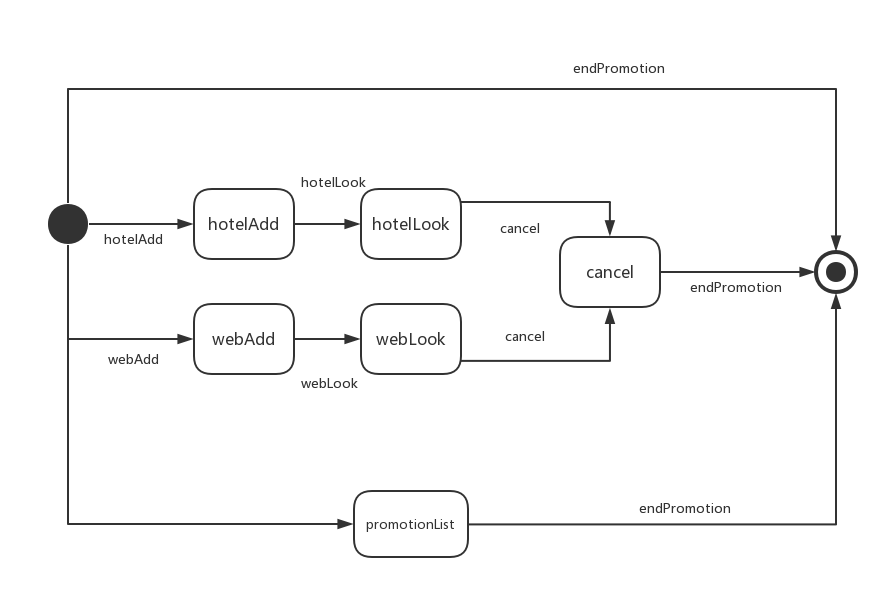


图 表明了制定网站促销策略业务逻辑处理的相关对象之间的协作。



所示的状态图描述了promotion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。

hotelAdd被UI调用进入hoteladd状态的promotion，之后通过hotelLook，promotion进入查看状态，之后通过cancel进入取消状态，最后通过endPromotion方法，进入结束状态；如果webAdd方法被UI调用，Promotion进入webAdd状态，之后通过webLook方法，进入webLook状态，之后通过cancel进入取消状态，最后通过endPromotion方法，进入结束状态；



MemberPromotionbl模块

（1）模块概述

MemberPromotionbl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。

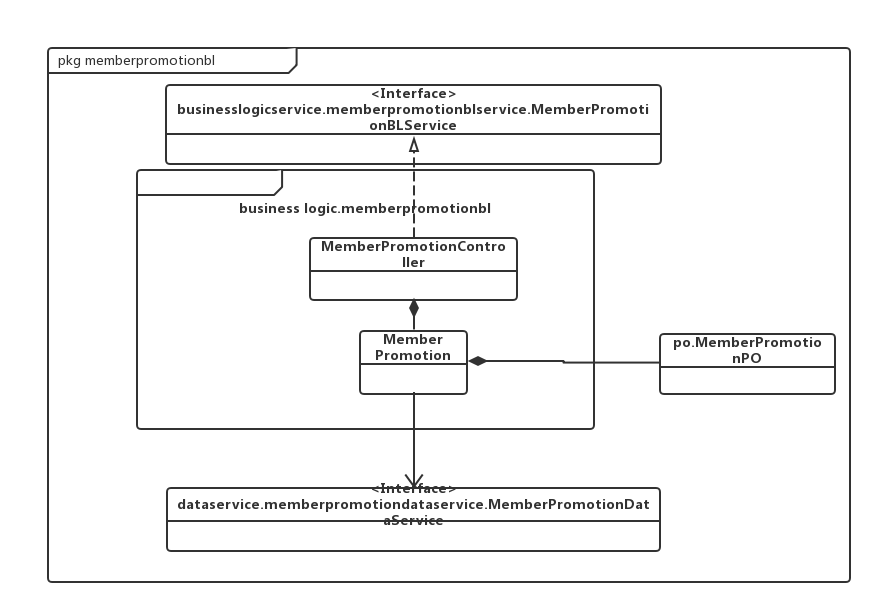
MemberPromotionbl模块的职责及接口参见软件系统结构描述文档表5-3-2-2。

MemberPromotionbl模块承担会员促销策略方面的功能，包括制定会员等级信息、制定会员折扣等。

（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businesslogicservice.memberpromotionblservice. MemberPromotionBLService接口。在业务逻辑层和数据层之间添加dataservice.memberpromotiondataservice. MemberPromotionDataService接口。为了隔离业务逻辑职责和逻辑控制职责，添加MemberPromotionController，这样MemberPromotionController会将对会员促销策略的业务逻辑处理委托给MemberPromotion对象。MemberPromotionPO是作为促销策略的持久性对象被添加到设计模型中去的。

MemberPromotionbl模块的设计如图所示。



（3）模块内部类的接口规范

**memberpromotioncontroller模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| MemberPromotionController.levelMake | 语法 | | public ArrayList <Integer > levelMake (levelInfo level) |
| 前置条件 | | 需创建一个MemberPromotion领域对象 |
| 后置条件 | | 调用Promotion领域对象的levelMake方法 |
| MemberPromotion.getLevel | 语法 | | Public void getLevel(levelInfo level) |
| 前置条件 | | 已创建一个MemberPromotion领域对象，并且分类名称符合输入规则 |
| 后置条件 | | 调用MemberPromotion领域对象的getLevel方法 |
| MemberPromotion.getDiscount | 语法 | | Public void getLevel(levelInfo level) |
| 前置条件 | | 已创建一个MemberPromotion领域对象，并且分类名称符合输入规则 |
| 后置条件 | | 调用MemberPromotion领域对象的getDiscount方法 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| MemberPromotionController.levelMake | | 制定会员等级 | |
| MemberPromotion.getLevel | | 获取会员等级信息 | |
| MemberPromotion.getDiscount | | 获取会员折扣信息 | |

**MemberPromotion模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| MemberPromotion.levelMake | 语法 | | public ArrayList <Integer > levelMake (levelInfo level) |
| 前置条件 | | 无 |
| 后置条件 | | 系统根据信用值划分会员等级 |
| MemberPromotion.getLevel | 语法 | | Public void getLevel(levelInfo level) |
| 前置条件 | | 无 |
| 后置条件 | | 系统获取输入的等级信息 |
| MemberPromotion.getDiscount | 语法 | | Public void getDiscount (levelInfo level) |
| 前置条件 | | 会员等级在折扣范围内 |
| 后置条件 | | 系统获取输入的等级信息，找到对应的折扣，返回折扣信息 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| DataFactory.getMemberPromotionData() | | 得到MemberPromotion的数据的服务的引用 | |
| MemberPromotionDataService.  levelMake(levelInfo level) | | 得到会员信用值信息，制定会员等级 | |
| MemberPromotion.getLevel(levelInfo level) | | 得到会员等级信息 | |
| MemberPromotion.getDiscount(levelInfo level) | | 根据会员等级信息得到会员折扣 | |

（4）业务逻辑层的动态模型

在会员促销策略管理业务中，分为制定会员等级，获取会员折扣信息，获取会员等级信息等业务。

所示的状态图描述了MemberPromotion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。

levelMake被UI调用进入make状态的memberpromotion，之后通过getLevel，memberpromotion进入查看状态，之后通过endMember Promotion方法，进入结束状态；如果getDiscount方法被UI调用，memberPromotion进入discount状态，之后通过endPromotion方法，进入结束状态；

