二、中层设计之逻辑层设计

### 2.5 Promotion模块的静态结构与动态行为

#### 2.5.1 HotelDiscountService

#### 2.5.1.1 局部模块的职责

（1）模块概述

HotelDiscountService模块承担的需求参见需求规格说明文档功能需求及相关的非功能需求。HotelDiscountService模块的职责及接口参见软件系统结构描述文档表5.3.2-7

（2）整体结构

HotelDiscountService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。HotelDiscountService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HotelDiscountService, HotelDiscountDAO 两个接口。HotelDiscountPO是做为管理信息的持久化对象被添加到设计模型中的。

图2.5.1.1-1 HotelDiscountService模块设计

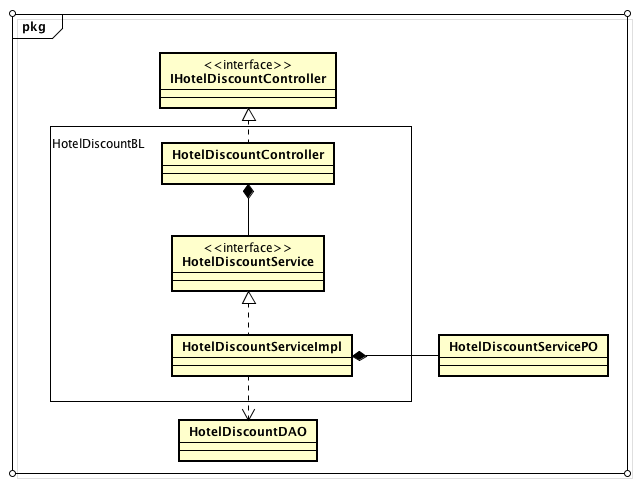


表2.5.1.1-2 HotelDiscountService模块各个类的职责

|  |  |
| --- | --- |
| 类 | 职责 |
| HotelDiscountController | 负责实现对应酒店折扣界面所需的方法 |
| HotelDiscountServiceImpl | 负责处理有关酒店折扣的功能 |

#### 2.5.1.2 局部模块的接口规范

表2.5.1.2 -1 HotelDiscountService的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| HotelDiscountService. add | 语法 | public ResultMessage add (HotelDiscountVO) | |
| 前置条件 | 已知酒店促销策略信息，需要将其添加到数据库 | |
| 后置条件 | 返回添加的结果信息 | |
| HotelDiscountService.update | 语法 | public ResultMessage update(HotelDiscountVO) | |
| 前置条件 | 已知更新后的酒店促销策略信息，需要更新数据库 | |
| 后置条件 | 返回更新的结果信息 | |
| HotelDiscountService.delete | 语法 | public ResultMessage delete(int id) | |
| 前置条件 | 已知酒店优惠策略的编号，需要将其删除 | |
| 后置条件 | 返回删除的结果信息 | |
| HotelDiscountService.findAllByHotelID | 语法 | public List<HotelDiscountVO> findAllByHotelID(int hotelID) | |
| 前置条件 | 已知酒店编号，需要该酒店所有的优惠策略 | |
| 后置条件 | 返回该酒店的所有优惠策略 | |
| HotelDiscountService.findNotAddedType | 语法 | List<HotelDiscountType> findNotAddedType(int hotelID); | |
| 前置条件 | 已知酒店编号，需要返回该酒店尚未添加过的酒店促销策略类型 | |
| 后置条件 | 无 | |
| HotelDiscountService.createAllStrategies | 语法 | public List<HotelDiscount> createAllStrategies(int hotelID); | |
| 前置条件 | 已知酒店编号，需要返回该酒店的所有策略领域对象 | |
| 后置条件 | 无 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| HotelDiscountDAO.add (HotelDiscountPO) | | | 添加单一持久化序列 |
| HotelDiscountDAO.update(HotelDiscountPO) | | | 更新单一持久化序列 |
| HotelDiscount.delete(int id) | | | 删除单一持久化序列 |
| HotelDiscount.findAllByHotelID(int hotelID); | | | 查询单一持久化列表 |

#### 2.5.1.3 局部模块的行为

图2.5.1.3-1 HotelDiscountService模块时序图

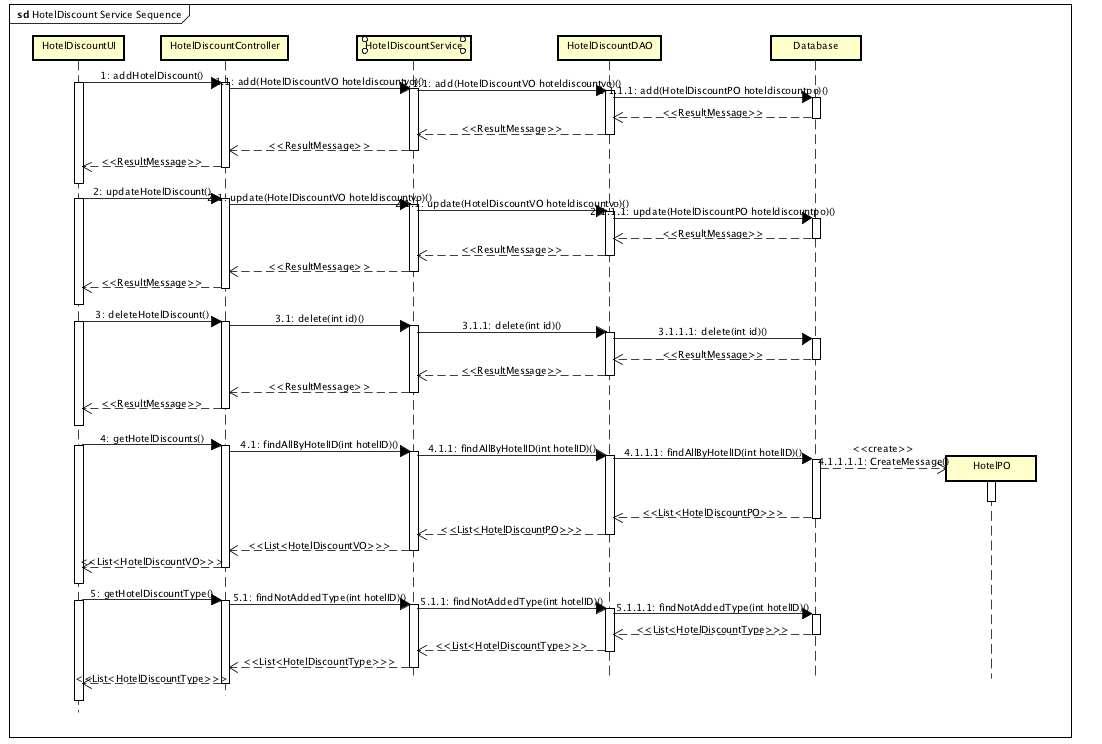
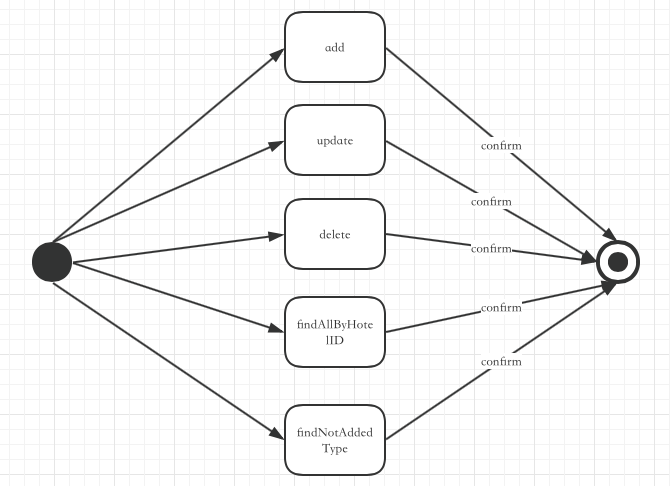


图2.5.1.3-1 表明了酒店管理系统中，当酒店工作人员添加、更新、删除、查看酒店折扣时，酒店折扣业务逻辑处理的相关对象之间的协作。

图2.5.1.3-2 HotelDiscountService模块状态图



#### 2.5.2 WebDiscountService

#### 2.5.2.1 局部模块的职责

（1）模块概述

WebDiscountService模块承担的需求参见需求规格说明文档功能需求及相关的非功能需求。WebDiscountService模块的职责及接口参见软件系统结构描述文档表5.3.2-9

（2）整体结构

WebDiscountService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。WebDiscountService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WebDiscountService, WebDiscountDAO 两个接口。WebDiscountPO是做为管理信息的持久化对象被添加到设计模型中的。

图2.5.2.1-1 WebDiscountService模块设计

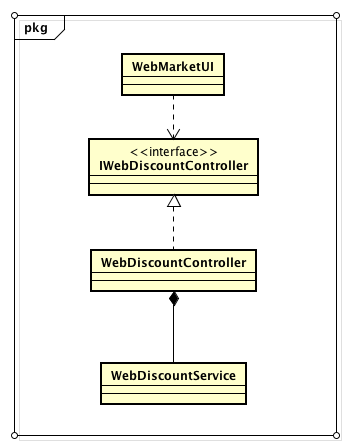


表2.5.2.1-2 WebDiscountService模块各个类的职责

|  |  |
| --- | --- |
| 类 | 职责 |
| WebDiscountController | 负责实现对应网站折扣界面所需的方法 |
| WebDiscountServiceImpl | 负责处理有关网站折扣的功能 |

#### 2.5.2.2 局部模块的接口规范

表2.5.2.2 -1 WebDiscountService的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| WebDiscountService.findAll | 语法 | public List<WebDiscountVO> findAll() | |
| 前置条件 | 无，需要网站所有促销策略信息 | |
| 后置条件 | 返回网站所有促销策略信息 | |
| WebDiscountService.add | 语法 | public ResultMessage add (WebDiscountVO) | |
| 前置条件 | 已知网站促销策略信息，需要将其添加到数据库 | |
| 后置条件 | 返回添加的结果信息 | |
| WebDiscountService. update | 语法 | public ResultMessage update(WebDiscountVO) | |
| 前置条件 | 已知更新后的网站促销策略信息，需要将数据库中的信息进行更新 | |
| 后置条件 | 返回更新的结果信息 | |
| WebDiscount.delete | 语法 | public ResultMessage delete(int id) | |
| 前置条件 | 已知网站促销策略编号，需要将其从数据库中删除 | |
| 后置条件 | 返回删除的结果信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| WebDiscountDAO. add (WebDiscountPO) | | | 添加单一持久化序列 |
| WebDiscountDAO.update | | | 更新单一持久化序列 |
| WebDiscount. delete | | | 删除单一持久化序列 |
| WebDiscount.findAll | | | 查找单一持久化列表 |

#### 2.5.2.3 局部模块的行为

图2.5.2.3-1 WebDiscountService模块时序图

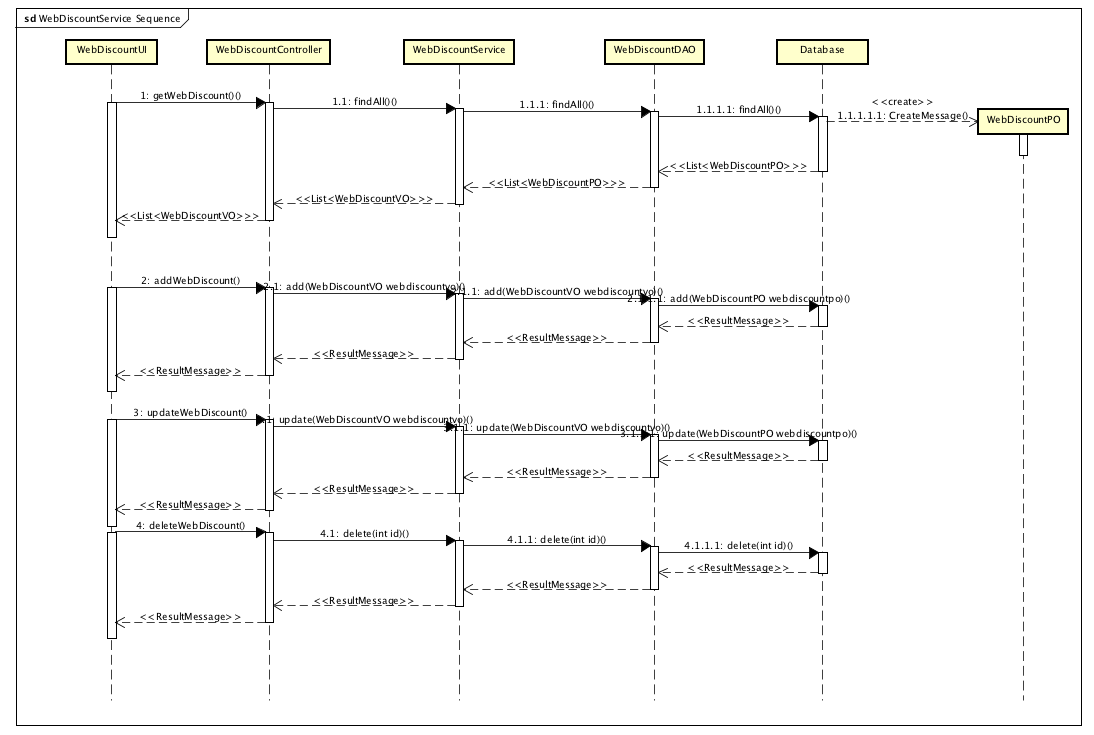
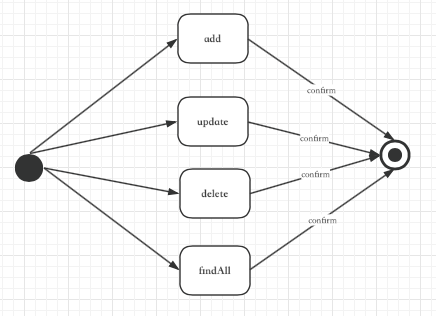


图2.5.2.3-1 表明了酒店管理系统中，当网站营销人员查看、添加、更新、删除网站折扣时，网站折扣业务逻辑处理的相关对象之间的协作。

图2.5.2.3-2 WebDiscountService模块状态图



#### 2.5.2 EnterpriseService

#### 2.5.2.1 局部模块的职责

（1）模块概述

EnterpriseService模块承担的需求参见需求规格说明文档功能需求及相关的非功能需求。EnterpriseService模块的职责及接口参见软件系统结构描述文档表5.3.2-7

（2）整体结构

EnterpriseService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。EnterpriseService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EnterpriseService, EnterpriseDAO 两个接口。EnterprisePO是做为管理信息的持久化对象被添加到设计模型中的。

图2.5.3.1-1 EnterpriseService模块设计

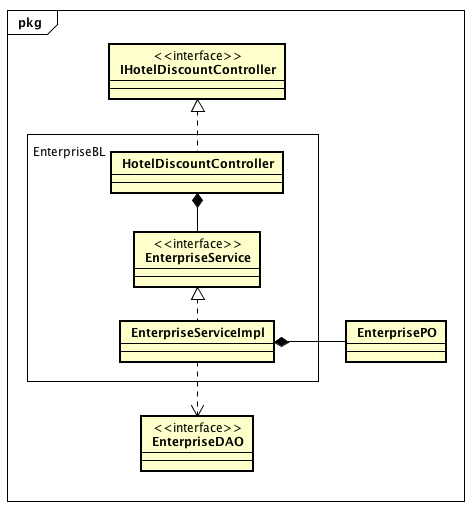


表2.5.3.1-2 EnterpriseService模块各个类的职责

|  |  |
| --- | --- |
| 类 | 职责 |
| HotelDiscountController | 负责实现对应酒店促销策略界面，与合作企业有关的方法 |
| EnterpriseServiceImpl | 负责处理有关合作企业的功能 |

#### 2.5.3.2 局部模块的接口规范

表2.5.3.2 -1 EnterpriseService的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| EnterpriseService.getAllEnterprises | 语法 | public List<EnterpriseVO> getAllEnterprises() | |
| 前置条件 | 无，需要所有的企业信息 | |
| 后置条件 | 无 | |
| EnterpriseService.add | 语法 | public ResultMessage add(EnterpriseVO ); | |
| 前置条件 | 已知企业信息，需要将其添加到数据库 | |
| 后置条件 | 在数据库中保存新添加的酒店信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| EnterpriseDAO.findAll(); | | | 返回企业信息列表 |
| EnterpriseDAO.add(EnterprisePO ) | | | 新增单一持久化序列 |

#### 2.5.3.3 局部模块的行为

图2.5.3.3-1 EnterpriseService模块时序图

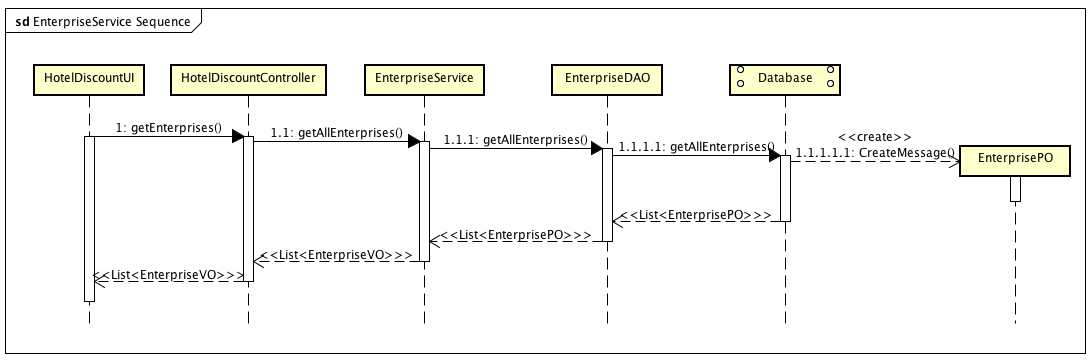
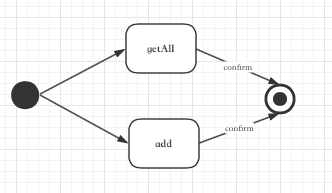


图2.5.3.3-1 表明了酒店管理系统中，当酒店工作人员查看合作企业列表时，合作企业业务逻辑处理的相关对象之间的协作。

图2.5.3.3-2 EnterpriseService模块状态图



### 2.6 CreditRecord模块的静态结构与动态行为

#### 2.6.1 局部模块的职责

（1）模块概述

CreditRecord模块承担的需求参见需求规格说明文档功能需求及相关的非功能需求。CreditRecord模块的职责及接口参见软件系统结构描述文档表5.3.2-6

（2）整体结构

CreditRecord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。CreditRecord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CreditRecordService, CreditRecordDAO 两个接口。CreditRecordPO是做为管理信息的持久化对象被添加到设计模型中的。

图2.6.1-1 CreditRecord模块设计

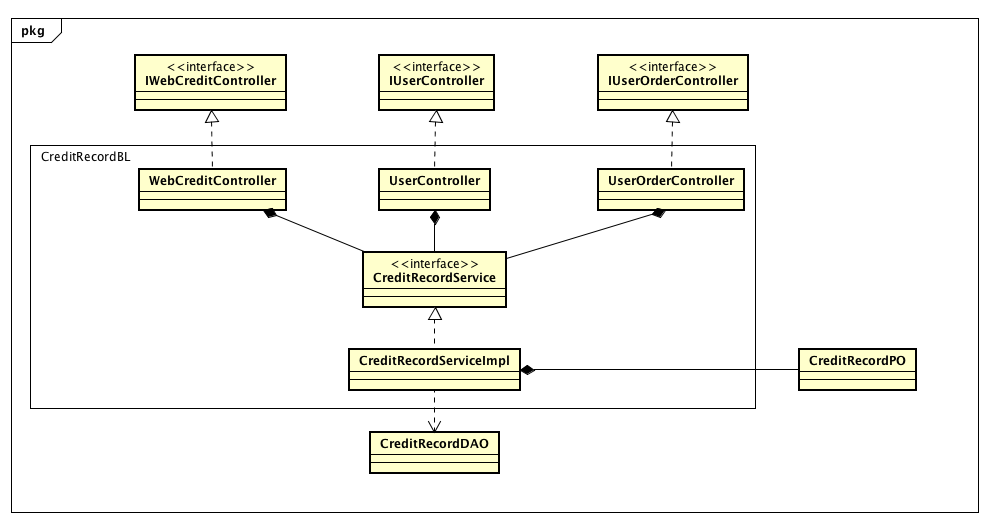


表2.6.1-2 CreditRecord模块各个类的职责

|  |  |
| --- | --- |
| 类 | 职责 |
| WebCreditController | 负责实现对应信用充值界面所需的方法 |
| UserController | 负责实现对应个人信息界面所需的方法 |
| UserOrderController | 负责实现对应订单信息界面所需的方法 |
| CreditRecordServiceImpl | 负责处理有关信用记录的功能 |

#### 2.6.2 局部模块的接口规范

表2.6.2-1CreditRecordService的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| CreditRecordService. findByUsername | 语法 | public List<CreditRecordVO> findByUsername(String username) | |
| 前置条件 | 已知用户名，需要该用户的所有信用记录 | |
| 后置条件 | 返回该用户的所有信用记录 | |
| CreditRecordService.add | 语法 | public ResultMessage add(CreditRecordVO) | |
| 前置条件 | 已知信用记录的信息，需要将其添加到数据库 | |
| 后置条件 | 返回添加的结果信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| CreditRecordDAO. findByUsername(String username) | | | 查找单一持久化列表 |
| CreditRecordDAO. add(CreditRecordPO) | | | 增加单一持久化序列 |
| UserService. findByUsername(String username) | | | 查找用户的原始信用值 |
| UserService. update(UserVO) | | | 更新用户的信用值信息 |

#### 2.6.3 局部模块的行为

图2.6.3-1 CreditRecord模块时序图1

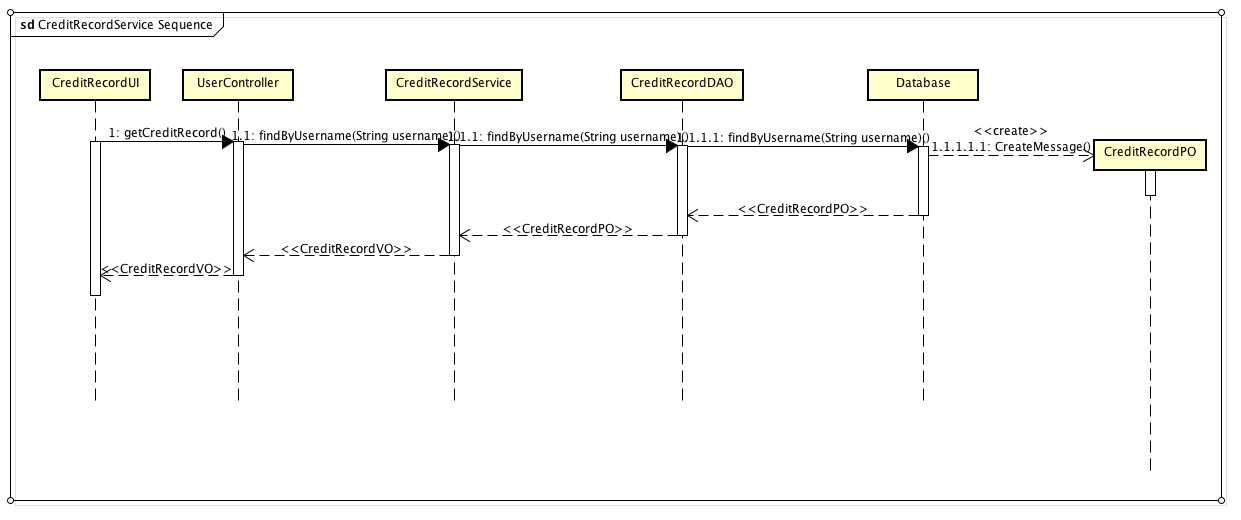


图2.6.3-1 表明了酒店管理系统中，当用户查询信用记录时，信用记录业务逻辑处理的相关对象之间的协作。

图2.6.3-2 CreditRecord模块时序图2

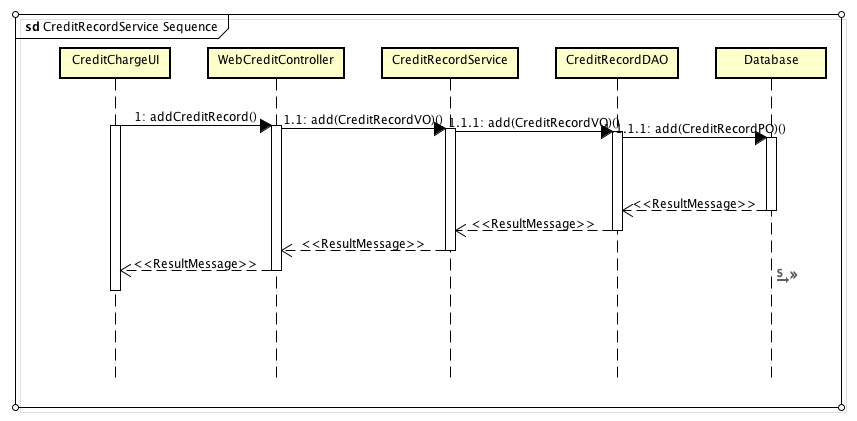
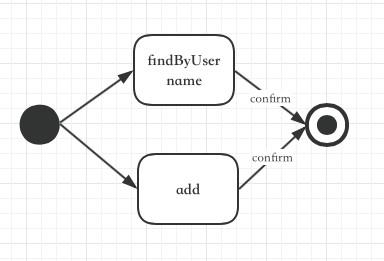


图2.6.3-2 表明了酒店管理系统中，当网站营销人员为用户信用充值时，信用记录业务逻辑处理的相关对象之间的协作。

图2.6.3-3 CreditRecord模块状态图



## 三、中层设计之控制器设计

### 3.4 WebMarketUI包所对应的Controller

#### 3.4.1 WebDiscountController

图3.4.1-1 WebDiscountController 类图

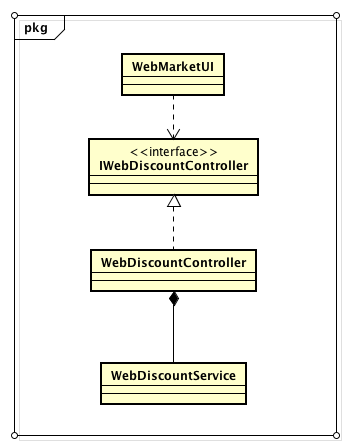


表3.4.1-2 WebDiscountController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| WebDiscountController.findAll | 语法 | Public List<WebDiscountVO> findAll() | |
| 前置条件 | 需要返回所有WebDiscount列表 | |
| 后置条件 | 无 | |
| WebDiscountController.add | 语法 | Public ResultMessage add (WebDiscountVO webdiscountvo) | |
| 前置条件 | 已知网站折扣，需要将其添加到数据库中 | |
| 后置条件 | 在数据库中保存新增的网站折扣信息 | |
| WebDiscountController.update | 语法 | Public ResultMessage update (WebDiscountVO webdiscountvo) | |
| 前置条件 | 已知网站折扣，需要将其在数据库中更新 | |
| 后置条件 | 在数据库中更新网站折扣信息 | |
| WebDiscountController.delete | 语法 | Public ResultMessage delete(int id) | |
| 前置条件 | 已知网站折扣ID，需要将其删除 | |
| 后置条件 | 在数据库中删除此网站折扣 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| WebDiscountService.findAll() | | | 查找单一持久化序列 |
| WebDiscountService.add(WebDiscountVO webdiscountvo) | | | 添加单一持久化序列 |
| WebDiscountService.update(WebDiscountVO webdiscountvo) | | | 更新单一持久化序列 |
| WebDiscountService.delete() | | | 删除单一持久化序列 |

#### 3.4.2 WebOrderController

图3.4.2-1 WebOrderController 类图

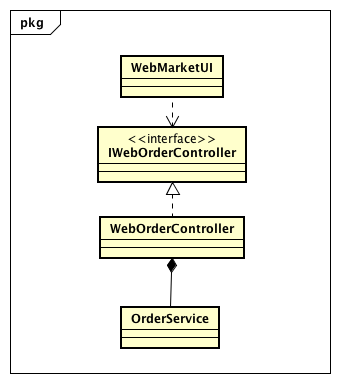


表3.4.2 -2 WebOrderController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| WebOrderController.  findOrderByOrderType | 语法 | Public List<OrderVO> findOrderByOrderType(OrderStatus status) | |
| 前置条件 | 已知订单状态，需要返回符合条件的订单列表 | |
| 后置条件 | 无 | |
| WebOrderController.  findOrderByID | 语法 | Public OrderVO findOrderByID(int ID) | |
| 前置条件 | 已知订单ID,需要返回该订单信息 | |
| 后置条件 | 无 | |
| WebOrderController.  revokeOrder | 语法 | Public ResultMessage revokeOrder(OrderVO ordervo) | |
| 前置条件 | 已知订单记录，需要撤销该订单 | |
| 后置条件 | 在数据库中删除此订单记录 | |
| WebOrderController.  findOrderByUsername | 语法 | Public List<OrderVO> findOrderByUsername(String username) | |
| 前置条件 | 已知username，需要返回对应的订单列表 | |
| 后置条件 | 无 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| OrderService.revoke(OrderVO ordervo) | | | 删除单一持久化序列 |
| OrderService.findByID(int ID) | | | 查找单一持久化序列 |
| OrderService.findByUsername(String username) | | | 查找单一持久化序列 |
| OrderService.findByOrderType(OrderStatus status) | | | 查找单一持久化序列 |

#### 3.4.3 WebCreditController

图3.4.3-1 WebCreditController 类图

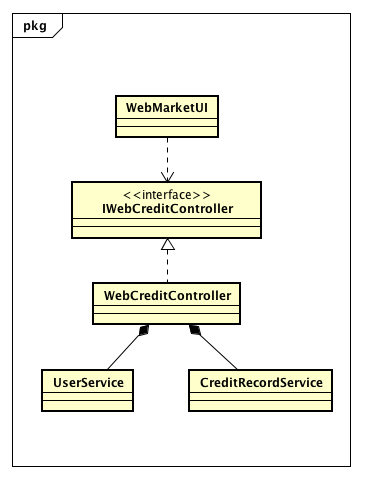


表3.4.3 -2 WebCreditController的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| WebCreditController.  findUserByUsername | 语法 | Public userVO findUserByUsername(String username) | |
| 前置条件 | 已知username，需要返回相关用户信息 | |
| 后置条件 | 无 | |
| WebCreditController.  addCreditRecord | 语法 | Public ResultMessage addCreditRecord(CreditRecordVO creditrecordvo) | |
| 前置条件 | 已知信用记录，需要将其添加到数据库 | |
| 后置条件 | 在数据库中保存新增的信用记录 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| CreditRecordService.findByUsername(String username) | | | 查找单一持久化序列 |
| CreditRecordService.add(CreditRecordVO creditrecordvo) | | | 添加单一持久化序列 |
| UserService.findByUsername(String username) | | | 查找单一持久化序列 |