4.1.1 Accountbl模块

（1）模块概述

Accountbl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。

Accountbl模块的职责及借款参见软件体系结构描述文档。

（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。Account模块对应的接口为展示层与业务逻辑层之间的businesslogicservice.accountblservice.AccountblService接口与业务逻辑层与数据层之间的dataservice.accountdataservice.AccountblService接口。为了隔离业务逻辑职责和逻辑控制职责，我们增加了AccountController，这样AccountController会将对账户的业务逻辑处理委托给Account对象。AccountPO是作为银行账户的持久化对象被添加到设计模型中去的。ReceiptPO是作为收款单的持久化对象被添加到设计模型中去的。PaymentPO是作为付款单的持久化对象被添加到设计模型中去的。而Account保存简单的账户信息，Receipt为收款的转账列表中的一项，ReceiptList负责管理Receipt，并计算收款总额。Payment为付款的条目清单中的一项。PaymentList负责管理Payment，并计算付款总额。SystemlogInfo、FinancialInfo、SalesInfo、InvoiceInfo都是根据依赖倒置原则，为了消除循环依赖而产生的接口。

Accountbl模块的设计如图

Accountbl模块各个类的职责如表

|  |  |
| --- | --- |
| 模块 | 职责 |
| AccountController | 负责实现账户界面所需要的服务 |
| Account | 收付款以账户管理领域模型对象，拥有一次收款、付款以及账户等信息，可以帮助完成账户界面所需要的服务 |
| ReceiptList | 管理此次操作中的收款的转账列表，完成计算收款总额操作 |
| Receipt | 收款操作中的转账列表的一项 |
| PaymentList | 管理此次操作中的付款的条目清单，完成计算付款总额的操作 |
| Payment | 付款操作中的条目清单的一项 |

（3）模块内部类的接口规范

AccountController接口规范：

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| AccountController.checkAccount() | 语法 | Public ResultMessage checkAccount() |
| 前置条件 | 已创建一个Account领域对象 |
| 后置条件 | 调用Account领域对象的checkAccount方法 |
| AccountController.addAccount | 语法 | Public ResultMessage addAccount(String name, int money) |
| 前置条件 | 已创建一个Account领域对象，输入符合规则 |
| 后置条件 | 调用Account领域对象addAccount的方法 |
| AccountController.delAccount | 语法 | Public ResultMessage delAccount(String name) |
| 前置条件 | 已创建一个Account领域对象，输入符合规则 |
| 后置条件 | 调用Account领域对象delAccount的方法 |
| AccountController.updateAccount | 语法 | Public ResultMessage updateAccount(String name1, String name2) |
| 前置条件 | 已创建一个Account领域对象，输入符合规则 |
| 后置条件 | 调用Account领域对象updateAccount的方法 |
| AccountController.searchAccount | 语法 | Public ResultMessage searchAccount(String word) |
| 前置条件 | 已创建一个Account领域对象 |
| 后置条件 | 调用Account领域对象searchAccount的方法 |
| AccountController.addReceipt | 语法 | Public ResultMessage addReceipt(String customer\_name, String clerk, String account\_name, int money, int all\_money, String ps) |
| 前置条件 | 已创建一个Account领域对象，输入符合规则 |
| 后置条件 | 调用Account领域对象addReceipt的方法 |
| AccountController.addPayment | 语法 | Public ResultMessage addPayment(String customer\_name, String clerk, String account\_name, int money, int all\_money, String ps) |
| 前置条件 | 已创建一个Account领域对象，输入符合规则 |
| 后置条件 | 调用Account领域对象addPayment的方法 |
| 需要的服务 | | |
| Account.checkAccount() | | 获得当前账户信息 |
| Account.addAccount (String name, int money) | | 添加账户 |
| Account.delAccount(String name) | | 删除账户 |
| Account.updateAccount(String name1, String name2) | | 更新账户 |
| Account.searchAccount(String word) | | 根据word查找账户 |
| Account.addReceipt(String customer\_name, String clerk, String account\_name, int money, int all\_money, String ps) | | 添加收款单 |
| Account.addPayment(String customer\_name, String clerk, String account\_name, int money, int all\_money, String ps) | | 添加付款单 |

Account接口规范：

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| Account.checkAccount() | 语法 | Public ResultMessage checkAccount() |
| 前置条件 | 管理人员为总经理或财务经理 |
| 后置条件 | 返回账户信息 |
| Account.addAccount | 语法 | Public ResultMessage addAccount(String name, int money) |
| 前置条件 | 需要添加账户时，管理人员为总经理或财务经理 |
| 后置条件 | 返回是否添加成功 |
| Account.delAccount | 语法 | Public ResultMessage delAccount(String name) |
| 前置条件 | 需要删除账户时，管理人员为总经理或财务经理 |
| 后置条件 | 返回是否删除账户成功 |
| Account.updateAccount | 语法 | Public ResultMessage updateAccount(String name1, String name2) |
| 前置条件 | 需要更新账户信息时，管理人员为总经理或财务经理 |
| 后置条件 | 完成账户信息修改，返回是否成功修改 |
| Account.searchAccount | 语法 | Public ResultMessage searchAccount(String word) |
| 前置条件 | 操作员是总经理或财务经理 |
| 后置条件 | 返回查找结果 |
| Account.addReceipt | 语法 | Public ResultMessage addReceipt(String customer\_name, String clerk, String account\_name, int money, int all\_money, String ps) |
| 前置条件 | 收到货款时，输入的money合法 |
| 后置条件 | 完成收款单，交由总经理审批 |
| Account.addPayment | 语法 | Public ResultMessage addPayment(String customer\_name, String clerk, String account\_name, int money, int all\_money, String ps) |
| 前置条件 | 完成付款时，输入的money合法 |
| 后置条件 | 完成付款单，交由总经理审批 |
| 需要的服务 | | |
| AccountDataService.getAccount() | | 获得当前账户信息 |
| AccountDataService.addAccount(AccountPO po) | | 添加账户AccountPO对象 |
| AccountDataService.delAccount(AccountPO po) | | 删除账户AccountPO对象 |
| AccountDataService.updateAccount(AccountPO po1, AccountPO po2) | | 更新账户AccountPO对象 |
| AccountDataService.findAccount(String name) | | 根据name查找账户对象 |
| AccountDataService.getAllAccount() | | 获得所有的账户对象 |
| AccountDataService.addReceipt(ReceiptPO po) | | 添加收款单ReceiptPO对象 |
| AccountDataService.addPayment(PaymentPO po) | | 添加付款单PaymentPO对象 |
| AccountDataService.getAllReceipt() | | 获得所有的收款单对象 |
| AccountDataService.getAllPayment() | | 获得所有的付款单对象 |
| AccountDataService.clear | | 清除所有对象 |

（4）业务逻辑层的动态模型