4.2.5 AccountUI模块

（1）模块描述

AccountUI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。

AccountUI模块的职责及接口参加软件系统体系结构描述文档5.2.2表*AccountUI模块的接口规范。*

（2）整体结构

根据体系结构的设计，我们将系统分为界面层、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。比如展示层和逻辑层之间，我们添加LogicService.accountLogic接口。AccountManPanel用于封装账户管理的职责，CountPanel和CountInfoPanel分别用于新建账和查看期初信息。CountVO和AccountVO分别作为账户和账的持久化对象被添加到设计模型中。

AccountUI模块的设计如图所示



AccountUI模块各个类的设计

AccountUI模块各个类的职责如下表所示

**AccountUI各个类的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| FunctionFrame | 负责由用户功能主界面到具体功能界面的跳转 |
| AccountManPanel | 负责账户管理的面板 |
| CountPanel | 负责新建账的面板 |
| CountInfoPanel | 负责期初信息查看的面板 |

（3）模块内部类的接口规范

AccountManPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| AccountManPanel.search | 语法 | Public ResultMessage search(String keywords) |
| 前置条件 | 用户输入关键词并确认搜索 |
| 后置条件 | 获得符合关键词的账户信息列表 |
| AccountManPanel.showAll | 语法 | Public ResultMessage showAll() |
| 前置条件 | 用户确认搜索账户 |
| 后置条件 | 显示符合要求的账户信息列表 |
| AccountManPanel.addAccount | 语法 | Public Boolean addAccount(AccountVO account) |
| 前置条件 | 用户选择新增账户并输入新增账户信息 |
| 后置条件 | 显示新增是否成功 |
| AccountManPanel.deleteAccount | 语法 | Public Boolean deleteAccount(String id) |
| 前置条件 | 用户选择删除选中的账户 |
| 后置条件 | 显示删除是否成功 |
| AccountManPanel.modifyAccount | 语法 | Public Boolean modifyAccount(AccountVO account) |
| 前置条件 | 用户选择修改选中的账户信息 |
| 后置条件 | 显示修改是否成功 |
| AccountManPanel.show | 语法 | Public Boolean show(String id) |
| 前置条件 | 用户在列表中选中了一个账户 |
| 后置条件 | 显示该账户详细信息 |
| AccountManPanel.end | 语法 | Public Boolean end() |
| 前置条件 | 用户关闭账户管理 |
| 后置条件 | 系统返回上一层 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| AccountLogicSer. addAccount(accountPO po) | 根据po新增账户对象 |
| AccountLogicSer.deleteAccount(String id) | 根据id删除账户对象 |
| AccountLogicSer.reviseAccount(accountPO po) | 根据po修改对应账户信息 |
| AccountLogicSer.accountList() | 查看账户列表界面的逻辑接口 |
| AccountLogicSer.checkAccount() | 查看某一账户信息界面的逻辑接口 |
| AccountLogicSer.searchAccount(String keywords) | 检索账户界面的逻辑接口 |

CountPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| CountPanel.search | 语法 | Public ResultMessage search(String keywords) |
| 前置条件 | 用户输入关键词并确认搜索 |
| 后置条件 | 获得符合关键词的账列表 |
| CountPanel showAll | 语法 | Public ResultMessage showAll() |
| 前置条件 | 用户确认搜索账信息 |
| 后置条件 | 显示符合要求的账列表 |
| CountPanel.addCount | 语法 | Public Boolean addAccount(CountVO count) |
| 前置条件 | 用户选择新增账并输入新增账信息 |
| 后置条件 | 显示新增是否成功 |
| CountPanel.deleteCount | 语法 | Public Boolean deleteCount(String id) |
| 前置条件 | 用户选择删除选中的账 |
| 后置条件 | 显示删除是否成功 |
| CountPanel.modifyCount | 语法 | Public Boolean modifyCount(CountVO count) |
| 前置条件 | 用户选择修改选中的账信息 |
| 后置条件 | 显示修改是否成功 |
| CountPanel.show | 语法 | Public Boolean show(String id) |
| 前置条件 | 用户在列表中选中了一个账 |
| 后置条件 | 显示该套账详细信息 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| CountLogicSer. addCount(accountPO po) | 根据po新增账对象 |
| CountLogicSer.deleteCcount(String id) | 根据id删除账对象 |
| CountLogicSer.reviseCount(accountPO po) | 根据po修改对应账信息 |
| CountLogicSer.CountList() | 查看账的列表界面的逻辑接口 |
| CountLogicSer.checkCount() | 查看某一账信息界面的逻辑接口 |
| CountLogicSer.searchCcount(String keywords) | 检索账的界面的逻辑接口 |

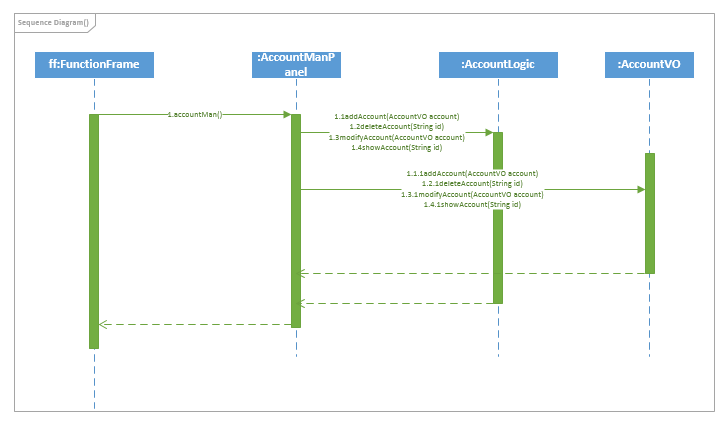
CountInfoPanel的模块接口

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| CountInfoPanel.showWorker | 语法 | Public ResultMessage showWorker(String org,String id) |
| 前置条件 | 用户输入机构名称，人员姓名 |
| 后置条件 | 显示人员信息 |
| CountInfoPanel.showVan | 语法 | Public ResultMessage showVan(String org,String id) |
| 前置条件 | 用户输入机构名称，车辆代号 |
| 后置条件 | 显示车辆信息 |
| CountInfoPanel.showGoodsInfo | 语法 | Public ResultMessage showGoodsInfo(String stock,String id) |
| 前置条件 | 用户输入中转中心仓库，订单条码号 |
| 后置条件 | 显示货物信息 |
| CountInfoPanel.showAccount | 语法 | Public ResultMessage showAccount(String id) |
| 前置条件 | 用户输入账户名称 |
| 后置条件 | 显示账户余额 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| AccountLogicSer.checkInitInfo() | 查看期初信息界面的逻辑接口 |

下图为使用账户管理功能时，账户管理界面处理相关对象之间的协作。



如下图所示的状态图描述了当用户使用账户管理功能时，系统的状态序列、引起转移的事件，以及因状态转移而伴随的动作呢。伴随着login方法被调用，界面跳转至FunctionFrame界面，用户选择账户管理功能后转至AccountManPanel。



AccountManPanel的状态图

（5）界面层的设计原理

同CheckUI模块。

4.2.6 CityUI模块

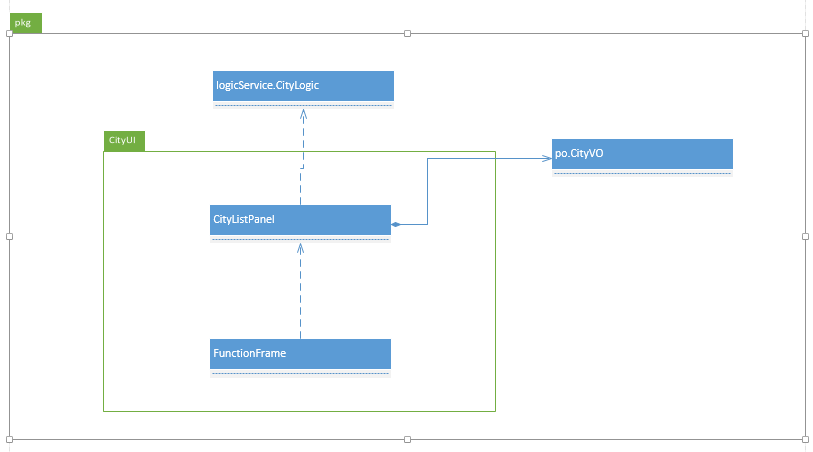
（1）模块描述

参见*4.2.5* *AccountUI模块*

（2）整体结构

在界面层和逻辑层之间添加logicService.cityLogic接口以增加灵活性。CityListPanel用于显示城市信息列表以及管理城市信息，CityVO作为持久化对象被添加到设计模型中去。

CityUI模块的设计如下图所示



CityUI模块类的职责参见*4.2.5* *AccountUI模块。*

（3）（4）（5）参见*4.2.5* *AccountUI模块 （3）AccountManPanel、（4）、（5）。*