4.2.3 FinancialData模块

（1）模块概述

FinancialData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。

FinancialData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。Financial模块对应的接口为展示层与业务逻辑层之间的businesslogicservice.financialblservice.FinancialblService接口与业务逻辑层与数据层之间的dataservice.financialdataservice.FinancialblService接口。为了隔离业务逻辑职责和逻辑控制职责，我们增加了FinancialDataController，这样FinancialDataController会将对账户的数据存储处理委托给FinancialData对象。SalesListPO是作为销售明细的持久化对象被添加到设计模型中去的。AllBillPO是作为经营历程的持久化对象被添加到设计模型中去的。OperatingConditionPO是作为经营情况的持久化对象被添加到设计模型中去的。SaleListManangement存储、管理SaleListPO持久化对象。AllBillManangement存储、管理AllBillPO持久化对象。OperatingConditionManangement存储、管理OperatingConditionPO持久化对象。

FinancialData模块的设计如图

FinancialData模块各个类的职责如表

|  |  |
| --- | --- |
| 模块 | 职责 |
| FinancialDataController | 负责实现财务账单数据，管理所需要的服务 |
| FinancialData | 财务表单管理信息，完成财务表单的查询与添加操作 |
| SaleListManangement | 销售明细领域模型对象，管理销售明细表单的添加与查询 |
| AllBillManangement | 经营情况领域模型对象，管理经营情况表单的添加与查询 |
| OperatingConditionManangement | 经营历程领域模型对象，管理经营历程表单的添加与查询 |

（3）模块内部类的接口规范

FinancialDataController接口规范：

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务 | | | |
| FinancialDataController.getSaleList() | 语法 | | Public SaleListPO[] getSaleList() |
| 前置条件 | | 已创建一个FinancialData领域对象，输入符合规则 |
| 后置条件 | | 调用Financial领域的getSaleList方法 |
| FinancialDataController.getAllBill() | 语法 | | Public AllBillPO[] getAllBill() |
| 前置条件 | | 已创建一个FinancialData领域对象，输入符合规则 |
| 后置条件 | | 调用Financial领域的getAllBill方法 |
| FinancialDataController.getOperatingCondition() | 语法 | | Public OperatingConditionPO[] getOperatingCondition() |
| 前置条件 | | 已创建一个FinancialData领域对象，输入符合规则 |
| 后置条件 | | 调用Financial领域的getOperatingCondition方法 |
| FinancialDataController.addSaleList(SaleListPO po) | 语法 | | Public void addSaleList(SaleListPO po) |
| 前置条件 | | 完成一个进货或销售单据的审批，已创建一个FinancialData领域对象，输入符合规则 |
| 后置条件 | | 调用Financial领域的addSaleList方法 |
| FinancialDataController.addAllBill(AllBillPO po) | 语法 | | Public void addAllBill(AllBillPO po) |
| 前置条件 | | 完成任何单据的审批，已创建一个FinancialData领域对象，输入符合规则 |
| 后置条件 | | 调用Financial领域的addAllBill方法 |
| FinancialDataController.addOperatingCondition(OperatingConditionPO po) | 语法 | | Public void addOperationCondition(OperatingConditionPO po) |
| 前置条件 | | 完成收付款单据的审批，已创建一个FinancialData领域对象，输入符合规则 |
| 后置条件 | | 调用Financial领域的addOperatingCondition方法 |
| FinancialDataController.clear() | 语法 | | Public void clear() |
| 前置条件 | | 进行期初建账之前，已创建一个FinancialData领域对象，输入符合规则 |
| 后置条件 | | 调用Financial领域的clear方法 |
| 需要的服务 | | | |
| FinancialData.getSaleList() | | 获得所有的销售明细 | |
| FinancialData.getAllBill() | | 获得所有的经营历程 | |
| FinancialData.getOperatingCondition() | | 获得所有的经营情况 | |
| FinancialData.addSaleList(SaleListPO po) | | 添加销售明细单 | |
| FinancialData.addAllBill(AllBillPO po) | | 添加经营历程单 | |
| FinancialData.addOperatingCondition(OperatingConditionPO po) | | 添加经营情况单 | |
| FinancialData.clear() | | 清除所有数据 | |

FinancialData接口规范：

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| FinancialData.getSaleList() | 语法 | Public SaleListPO[] getSaleList() |
| 前置条件 | 无 |
| 后置条件 | 返回一个包含所有SaleListPO对象的数组 |
| FinancialData.getAllBill() | 语法 | Public AllBillPO[] getAllBill() |
| 前置条件 | 无 |
| 后置条件 | 返回一个包含所有AllBillPO对象的数组 |
| FinancialData.getOperatingCondition() | 语法 | Public OperatingConditionPO[] getOperatingCondition() |
| 前置条件 | 无 |
| 后置条件 | 返回一个包含所有OperatingConditionPO对象的数组 |
| FinancialData.addSaleList(SaleListPO po) | 语法 | Public void addSaleList(SaleListPO po) |
| 前置条件 | 完成一个进货或销售单据的审批 |
| 后置条件 | 数据库添加一个SaleListPO对象 |
| FinancialData.addAllBill(AllBillPO po) | 语法 | Public void addAllBill(AllBillPO po) |
| 前置条件 | 完成任何单据的审批 |
| 后置条件 | 数据库添加一个AllBillPO对象 |
| FinancialData.addOperatingCondition(OperatingConditionPO po) | 语法 | Public void addOperationCondition(OperatingConditionPO po) |
| 前置条件 | 完成收付款单据的审批 |
| 后置条件 | 数据库添加一个OperatingConditionPO对象 |
| FinancialData.clear() | 语法 | Public void clear() |
| 前置条件 | 进行期初建账之前 |
| 后置条件 | 清楚所有SaleListPO，AllBillPO，OperatingConditionPO对象 |