### 4.2.6financebl模块

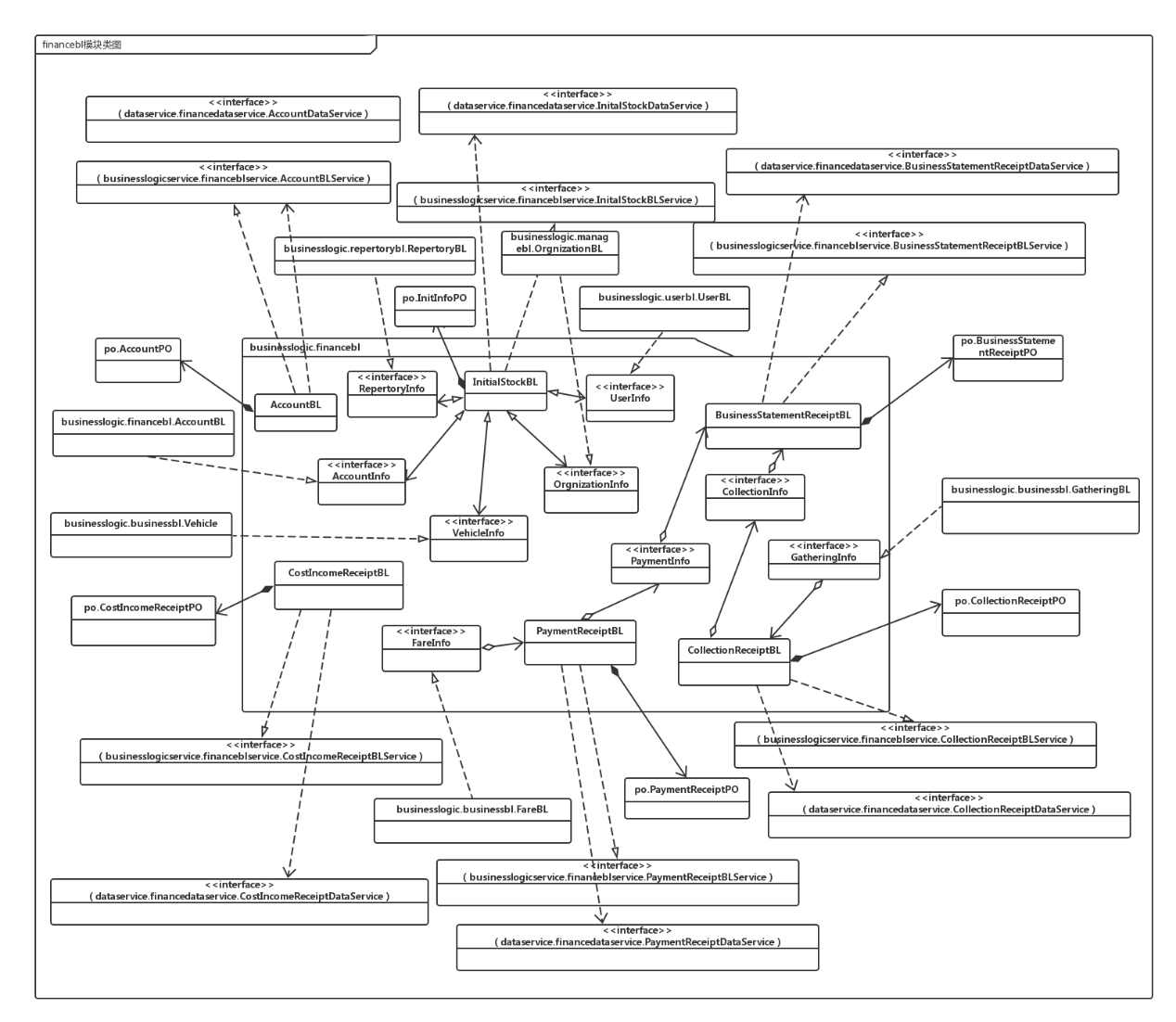
#### 4.2.6.1模块概述

financebl模块承担的需求参见ELS需求规格说明文档功能需求及相关非功能性需求。

financebl模块的职责及接口参见ELS体系结构设计文档。

#### 4.2.6.2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。比如展示层和业务逻辑层之间，我们添加businesslogicservice.financeblservice.CollectionReceiptBLService，businesslogicservice.financeblservice.PaymentReceiptBLService，businesslogicservice.financeblservice.CostIncomeReceiptBLService，businesslogicservice.financeblservice.BusinessStatementReceiptBLService，businesslogicservice.financeblservice.InitalStockBLService和businesslogicservice.financeblservice.AccountBLService接口。因为业务逻辑处理较少，根据专家模式没有添加controller类。业务逻辑层和数据层之间添加dataservice.financedataservice. CollectionReceiptDataService，dataservice.financedataservice. PaymentReceiptDataService，dataservice.financedataservice. CostIncomeReceiptDataService，dataservice.financedataservice. BusinessStatementReceiptDataService，dataservice.financedataservice. InitalStockDataService，dataservice.financedataservice. AccountDataService接口。CollectionReceiptPO,PaymentReceiptPO,BusinessStatementReceiptPO，CostIncomeReceiptPO，AccountPO和InitInfoPO是作为财务管理的持久化对象被添加到设计模型中去的。



**图6-1 financebl模块各个类的设计**

Financebl各个类的职责如表6-1所示

**表6-1 financebl模块各个类的职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| CollectionReceiptBL | 入款单的领域模型对象，拥有一个入款单所持有的编号，操作员，入款金额等信息，可帮助完成结算管理界面所需要的服务。 |
| PaymentReceiptBL | 付款单的领域模型对象，拥有一个付款单所持有的编号，操作员，付款金额等信息，可帮助完成新建付款单界面所需要的服务。 |
| CostIncomeReceiptBL | 成本收益表的领域模型对象，拥有一个成本收益表所持有的编号，操作员，收入，支出金额等信息，可帮助完成新建成本收益表界面所需要的服务。 |
| BusinessStatementReceiptBL | 经营情况表的领域模型对象，拥有一个经营情况表所持有的入款记录，付款记录，编号等信息，可以帮助完成经营情况表界面所需要的服务。 |
| InitialStockBL | 期初建账的领域模型对象，拥有一个套账所持有的人员、机构、车辆、库存和账户等信息，可以帮助完成期初建账界面所需要的服务。 |
| AccountBL | 账户的领域模型对象，拥有账户的名称、金额等信息，可以帮助完成账户管理界面所需要的服务。 |

#### 4.2.6.3模块内部类的接口规范

**表6-2 CollectionReceiptBL的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| CollectionReceiptBL .creatCollection | 语法 | | public int creatCollection (CollectionReceiptVO vo); |
| 前置条件 | | 用户要求新建入款单并输入有效的信息 |
| 后置条件 | | 系统存储入款单信息 |
| CollectionReceiptBL .getCollection | 语法 | | Public CollectionReceiptVO getCollection (String s); |
| 前置条件 | | 用户具有查看入款单的权限 |
| 后置条件 | | 系统显示符合条件的入款单 |
| CollectionReceiptBL .getAllCollection | 语法 | | public ArrayList<CollectionReceiptVO> getAllCollection(); |
| 前置条件 | | 用户有查看入款单的权限 |
| 后置条件 | | 系统显示所有入款单列表 |
| CollectionReceiptBL .getGatheringByTime | 语法 | | public ArrayList< GatheringReceiptVO > getGatheringByTime(String Time); |
| 前置条件 | | 用户输入日期并要求查看该日期的所有营业厅的收款单信息 |
| 后置条件 | | 系统显示该日期的所有收款单列表 |
| CollectionReceiptBL .getGatheringByHall | 语法 | | public ArrayList< GatheringReceiptVO > getGatheringByHall(String hallID); |
| 前置条件 | | 用户输入营业厅编号并要求查看该营业厅的所有收款单信息 |
| 后置条件 | | 系统显示该营业厅的所有收款单列表 |
| CollectionReceiptBL .getGatheringByBoth | 语法 | | public ArrayList<GatheringReceiptVO> getGatheingByBoth(String time,String hallID); |
| 前置条件 | | 用户输入营业厅编号和日期并要求查看符合条件的收款单信息 |
| 后置条件 | | 系统显示该日期该营业厅的收款单列表 |
| CollectionReceiptBL .getTotalMoney | 语法 | | public double getTotalMoney (ArrayList<GatheringReceiptVO> vo); |
| 前置条件 | | 用户要求合计金额 |
| 后置条件 | | 系统返回合计后的金额 |
| CollectionReceiptBL .getCollectionListID | 语法 | | public String getCollectionListID(); |
| 前置条件 | | 用户制定入款单时需要根据日期自动生成入款单编号 |
| 后置条件 | | 系统显示自动生成的入款单ID |
| CollectionReceiptBL .getUnapprovedCollectionReceipt | 语法 | | public ArrayList<CollectionReceiptVO> getUnapprovedCollectionReceipt(); |
| 前置条件 | | 总经理要求获取未审批的入款单 |
| 后置条件 | | 系统返回所有未审批的入款单 |
| CollectionReceiptBL .saveSubmittedCollectionReceiptInfo | 语法 | | public int saveSubmittedCollectionReceiptInfo (CollectionReceiptVO vo); |
| 前置条件 | | 总经理完成单据审批后返回了一条审批后的入款单 |
| 后置条件 | | 系统自动保存该审批后的入款单 |
| CollectionReceiptBL .excute | 语法 | | public int excute(CollectionReceiptVO vo); |
| 前置条件 | | 财务人员成功创建了一张入款单 |
| 后置条件 | | 系统保存账户变化后的结果 |
| 需要的服务（供接口） | | | |
| 服务名 | | 服务 | |
| DataFactory.getCollectionReceiptData | | 得到Collection数据的引用 | |
| FinanceDataService.creatCollection(CollectionReceiptPO po) | | 插入单一持久化对象CollectionReceiptPO | |
| FinanceDataService.getAllCollection() | | 得到持久化对象CollectionReceiptPO的ArrayList | |
| FinanceDataService.getCollection\_right(String beginTime,String endTime) | | 按时间区间获取持久化对象CollectionReceiptPO | |
| FinanceDataService.getUnapprovedCollectionReceipt() | | 获取所有符合未审批状态的持久化对象CollectionReceiptPO | |
| FinanceDataService. saveSubmittedCollectionReceiptInfo(CollectionReceiptPO po) | | 存储单一持久化对象CollectionReceiptPO | |

**表6-3 PaymentReceiptBL的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| PaymentReceiptBL.createPayment | 语法 | | public int creatPayment(PaymentReceiptVO vo); |
| 前置条件 | | 用户要求新建付款单并输入有效信息 |
| 后置条件 | | 系统存储付款单信息 |
| PaymentReceiptBL .getPaymentReceipt | 语法 | | public PaymentReceiptVO getPaymentReceipt(String s); |
| 前置条件 | | 用户有查看付款单的权限并输入有效时间 |
| 后置条件 | | 系统显示符合时间条件的付款单列表 |
| PaymentReceiptBL .getAllPayment | 语法 | | public ArrayList<PaymentReceiptVO> getAllPayment(); |
| 前置条件 | | 用户有查看付款单的权限 |
| 后置条件 | | 系统显示所有付款单列表 |
| PaymentReceiptBL .getPaymentReceiptListID | 语法 | | public String getPaymentReceiptListID(); |
| 前置条件 | | 用户制定付款单时需要根据日期来自动生成付款单编号 |
| 后置条件 | | 系统显示自动生成的付款单ID |
| PaymentReceiptBL .getSalary | 语法 | | public double getSalary(String time); |
| 前置条件 | | 用户有新建付款单的权限并输入有效时间 |
| 后置条件 | | 系统计算该时间内薪水总额，显示并存储 |
| PaymentReceiptBL .getFare | 语法 | | public double getFare(String time); |
| 前置条件 | | 用户有新建付款单的权限并输入有效时间 |
| 后置条件 | | 系统计算该时间内运费总额，显示并存储 |
| PaymentReceiptBL .getRent | 语法 | | public double getRent(String time); |
| 前置条件 | | 用户有新建付款单的权限并输入有效时间 |
| 后置条件 | | 系统计算该时间内租金总额，显示并存储 |
| PaymentReceiptBL .getUnapprovedPaymentReceipt | 语法 | | public ArrayList<PaymentReceiptVO> getUnapprovedPaymentReceipt(); |
| 前置条件 | | 总经理要求获取所有未审批的付款单 |
| 后置条件 | | 系统返回所有未审批的收款单 |
| PaymentReceiptBL .saveSubmittedPaymentReceiptInfo | 语法 | | Public int  saveSubmittedPaymentReceiptInfo (PaymentReceiptVO vo); |
| 前置条件 | | 总经理完成单据审批后返回了一条审批后的付款单 |
| 后置条件 | | 系统自动保存该审批后的付款单 |
| PaymentReceiptBL. excute | 语法 | | public int excute(PaymentReceiptVO vo); |
| 前置条件 | | 财务人员成功创建了一张付款单 |
| 后置条件 | | 系统保存账户金额的变化 |
| 需要的服务（供接口） | | | |
| 服务名 | | 服务 | |
| DataFactory.getPaymentReceiptData | | 得到payment的数据引用 | |
| FinanceDataService. creatPaymentReceipt(PaymentReceiptPO po) | | 插入单一持久化对象 PaymentReceiptPO | |
| FinanceDataService. getAllPaymentReceipt() | | 得到持久化对象 PaymentReceiptPO的ArrayList | |
| FinanceDataService. getPayment\_right(String beginTime,String endTime) | | 按时间区间获取持久化对象PaymentReceiptPO | |
| FinanceDataService. getUnapprovedPaymentReceipt() | | 获取所有符合未审批状态的持久化对象PaymentReceiptPO | |
| FinanceDataService. saveSubmittedPaymentReceiptInfo(PaymentReceiptPO po) | | 存储单一持久化对象PaymentReceiptPO | |

**表6-4 CostIncomeReceiptBL的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| CostIncomeReceiptBL.createCostIncomeList | 语法 | | public int creatCostIncomeList(CostIncomeReceiptVO vo); |
| 前置条件 | | 用户要求新建成本收益表并输入有效信息 |
| 后置条件 | | 向系统中输入新的成本收益表信息 |
| CostIncomeReceiptBL .getCostIncmeList | 语法 | | public CostIncomeReceiptVO getCostIncomeList(String s); |
| 前置条件 | | 用户有查看成本收益表的权限 |
| 后置条件 | | 系统返回符合条件的成本收益表 |
| CostIncomeReceiptBL .getCost | 语法 | | public double getCost(); |
| 前置条件 | | 用户有查看收款单的权限 |
| 后置条件 | | 系统返回所有付款单金额总和 |
| CostIncomeReceiptBL .getIncome | 语法 | | public double getIncome(CollectionReceiptVO vo); |
| 前置条件 | | 用户有查看入款单的权限 |
| 后置条件 | | 系统返回所有入款单金额总和 |
| CostIncomeReceiptBL .getProfit | 语法 | | public double getProfit(double income,double cost); |
| 前置条件 | | 已获取付款单金额之和,入款单金额之和 |
| 后置条件 | | 系统返回利润（即入款单付款单金额之差） |
| CostIncomeReceiptBL .getCostIncomeListID | 语法 | | public String getCostIncomeListID(); |
| 前置条件 | | 用户制定成本收益表时需要根据今天已经制定的成本收益表来自动生成编号 |
| 需要的服务（供接口） | | | |
| 服务名 | | 服务 | |
| DataFactory.getCostIncomeData | | 得到CostIncomeReceipt的引用 | |
| FinanceDataService. creatCostIncomeList(CostIncomeReceiptPO po) | | 插入单一持久化对象 CostIncomeReceiptPO | |
| publicCostIncomeReceiptPOgetCostIncomeReceipt(String time) | | 获取特定时间的成本收益表对象 | |

**表6-5 BusinessStatementReceiptBL的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| BusinessStatementReceiptBL.showBSL | 语法 | public BusinessstatementReceiptVO show(String beginTime,String endTime); |
| 前置条件 | 用户输入起始日期并要求查看经营情况表 |
| 后置条件 | 系统显示该段时间内所有入款单和付款单信息 |
| BusinessStatementReceiptBL .export | 语法 | public int export(BusinessstatementReceiptVO vo); |
| 前置条件 | 用户要求导出经营情况表的excel |
| 后置条件 | 系统生成经营情况表的excel |

**表6-6 InitialStockBL的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| InitialStockBL.initInfo | 语法 | | public int initInfo(InitInfoVO vo,String time); |
| 前置条件 | | 用户要求期初建账 |
| 后置条件 | | 系统初始化一套账 |
| InitialStockBL.getInitInfo | 语法 | | public InitInfoVO getInitInfo(String time); |
| 前置条件 | | 用户要求查看某个时间的期初信息 |
| 后置条件 | | 返回期初信息并显示 |
| InitialStockBL.getAllInitInfo | 语法 | | Public ArrayList<InitInfoVO> getAllInitInfo(); |
| 前置条件 | | 用户具有查看期初信息的权限 |
| 后置条件 | | 系统返回所有期初信息 |
| 需要的服务（供接口） | | | |
| 服务名 | | 服务 | |
| DataFactory.getInitialStockData(); | | 得到InitInfo数据的引用 | |
| FinanceDataService.initInfo(InitInfoPO po,String time) | | 在数据库中添加一套账的初始化信息（InitInfoPO） | |
| FinanceDataService.getInitInfo(String time) | | 查找符合时间条件的期初信息并返回 | |
| FinanceDataService.getAllInitInfo() | | 返回并输出所有的持久化对象InitInfoPO | |

**图6-7 AccountBL的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| AccountBL.addAccount | 语法 | | public int addAccount(AccountVO vo); |
| 前置条件 | | 用户要求新建账户 |
| 后置条件 | | 在账户列表中添加该账户信息 |
| AccountBL.deleteAccount | 语法 | | public int deleteAccount(AccountVO vo); |
| 前置条件 | | 用户要求删除账户 |
| 后置条件 | | 从账户列表中删除该账户信息 |
| AccountBL.modifyAccount | 语法 | | public int modifyAccount(AccountVO vo,String name) |
| 前置条件 | | 用户要求修改账户并选中需修改的账户 |
| 后置条件 | | 在账户列表中修改该账户信息 |
| AccountBL.findByName | 语法 | | public AccountVO findByName(String name); |
| 前置条件 | | 用户输入账户名称并查找 |
| 后置条件 | | 查找是否存在该名称的账户并返回该账户的信息 |
| AccountBL.findByKeyword | 语法 | | public ArrayList<AccountVO> findByKeyword(String s); |
| 前置条件 | | 用户输入关键词并查询 |
| 后置条件 | | 查找是否存在含有该关键字的账户并返回含有该关键字的账户的信息 |
| AccountBL.showAll | 语法 | | public ArrayList<AccountVO> showAll(); |
| 前置条件 | | 用户要求查看所有的账户 |
| 后置条件 | | 返回所有账户列表 |
| 需要的服务（供接口） | | | |
| 服务名 | | 服务 | |
| DataFactory.getAccountData | | 得到Account数据的引用 | |
| FinanceDataService.addAccount(AccountPO po) | | 插入单一持久化对象AccountPO | |
| FinanceDataService.deleteAccount(AccountPO po) | | 删除单一持久化对象AccountPO | |
| FinanceDataService.modifyAccount(AccountPO po,String name) | | 修改单一持久化对象AccountPO | |
| FinanceDataService.findByName(String name) | | 根据输入的名称查找单一持久化对象AccountPO | |
| FinanceDataService. findByKeyword(String s) | | 根据输入的关键字查找持久化对象AccountPO | |
| FinanceDataService.showAll() | | 输出所有持久化对象AccountPO | |
| FinanceDataService.addMoney (String name, double m) | | 修改持久化对象AccountPO的金额 | |
| FinanceDataService.delMoney (String name, double m) | | 修改持久化对象AccountPO的金额 | |

**图6-8 LogDiaryBL的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| LogDiary.addLogDiary | 语法 | | public int addLogDiary(LogDiaryVO vo,String time) |
| 前置条件 | | 用户新增一条系统日志信息 |
| 后置条件 | | 系统保存该系统日志信息 |
| LogDiary.getLogDiaryVO | 语法 | | public ArrayList<LogDiaryVO> getLogDiaryVO(String time) |
| 前置条件 | | 用户要求获取固定时间的所有系统日志 信息并输入有效时间 |
| 后置条件 | | 系统返回该时间的所有系统日志信息 |
| LogDiary.getAllLogDiaryVOs | 语法 | | public ArrayList<LogDiaryVO> getAllLogDiaryVOs(); |
| 前置条件 | | 用户要求获取所有系统日志信息 |
| 后置条件 | | 系统返回所有的系统日志信息 |
| 需要的服务（供接口） | | | |
| 服务名 | | 服务 | |
| DataFactory.getLogDiaryData(); | | 得到LogDiary数据的引用 | |
| public int addLogDiary(LogDiaryPO po,String time) | | 存储单一持久化对象 LogDiaryPO | |
| publicArrayList<LogDiaryPO>getLogDiaryPO(String time) | | 获取符合时间条件的持久化对象 LogDiaryPO | |
| public ArrayList<LogDiaryPO> getAllLogDiaryPOs() | | 获取所有的持久化对象 LogDiaryPO | |