### BankManagebl模块

（1）模块概述

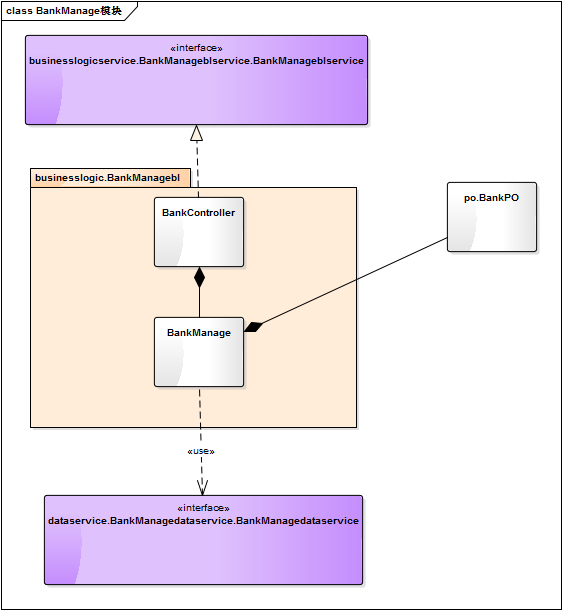
BankManagebl模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

BankManagebl模块的职责及接口参见软件系统结构描述文档表x.

（2）整体结构

Businesslogicservice.BankManageblservice.BankManageblservice为展示层 和业务层之间的接口。 Dataservice.BankManagedataservice.BankManagedataservice为业务层和数 据层之间的接口。BankController负责逻辑控制职责，BankManage负责业务逻辑 职责。BankPO为银行账户的持久化数据对象。

1. BankManagebl模块个各类的设计



1. BankManagebl模块各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| BankManage | 银行账户管理的领域模型对象，拥有银行账户的属性以及对数据库的引用，辅助完成银行账户管理所需要的服务 |
| BankContr  ller | 负责实现银行账户管理所需要的服务 |

（3）模块内部类的接口规范

BankManage和BankController的接口规范如表x到表x所示。

1. BankController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| BankController.add | 语法 | Public void add(BankVo vo) throws Exception |
| 前置条件 | 已创建一个BankManage对象，输入符合规则 |
| 后置条件 | 调用BankManage对象的add方法 |
| BankController.delete | 语法 | Public void delete(BankVo vo) throws Exception |
| 前置条件 | 已创建一个BankManage对象，输入符合规则 |
| 后置条件 | 调用BankManage对象的delete方法 |
| BankController.modify | 语法 | Public void modify(BankVo vo) throws Exception |
| 前置条件 | 已创建一个BankManage对象，输入符合规则 |
| 后置条件 | 调用BankManage的modify方法 |
| BankController.search | 语法 | Public BankVo[] search(String field) throws Exception |
| 前置条件 | 已创建一个BankManage对象，输入符合规则 |
| 后置条件 | 调用BankManage的search方法 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| BankManage.add(BankVPO po) | 添加一个银行账户 |
| BankManage.delete(BankPO po) | 删除一个银行账户 |
| BankManage.modify(BankPO po) | 更新一个银行账户信息 |
| BankManage.search(String field) | 搜索符合条件的结果 |

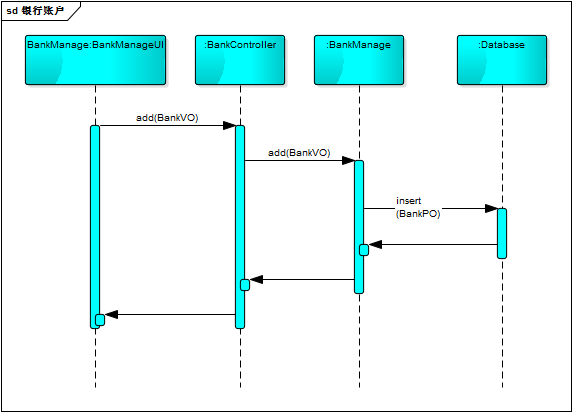
1. BankManage的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| BankManage.add | 语法 | public ResultMessage add(BankPO po)throws Exception |
| 前置条件 | 无 |
| 后置条件 | 新增一个银行账户，刷新列表 |
| BankManage.delete | 语法 | public ResultMessage delete(BankPO po)throws Exception |
| 前置条件 | 已选中列表某账户 |
| 后置条件 | 删除一个银行账户 |
| BankManage.modify | 语法 | public ResultMessage modify(BankPO po)throws Exception |
| 前置条件 | 已选中列表某账户 |
| 后置条件 | 更新一个银行账户的信息 |
| BankManage.search | 语法 | public ResultMessage search(String field)throws Exception |
| 前置条件 | 无 |
| 后置条件 | UI列表中显示符合条件的结果 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| RMIUtility.getImpl(String ) | 得到bank数据库的服务的引用 |
| BankManageDataService.find(String field) | 根据field查找持久化对象 |
| BankManageDataService.delete(BankPo po) | 删除单一持久化对象 |
| BankManageDataService.update(BankPo po) | 更新单一持久化对象 |
| BankManageDataService.insert(BankPo po) | 插入单一持久化对象 |
| BankManageDataService.finds() | 获得数据库所有数据 |

（4）业务逻辑层的动态模型

1. 添加银行账户的顺序图



（5）业务逻辑层的设计原理

利用委托式控制风格，将BankManage的bl层的职责交给BankManage。每个界面需要访问的业务逻辑有BankController控制器委托给上述的领域对象。

### FinancialReceiptbl模块

（1）模块概述

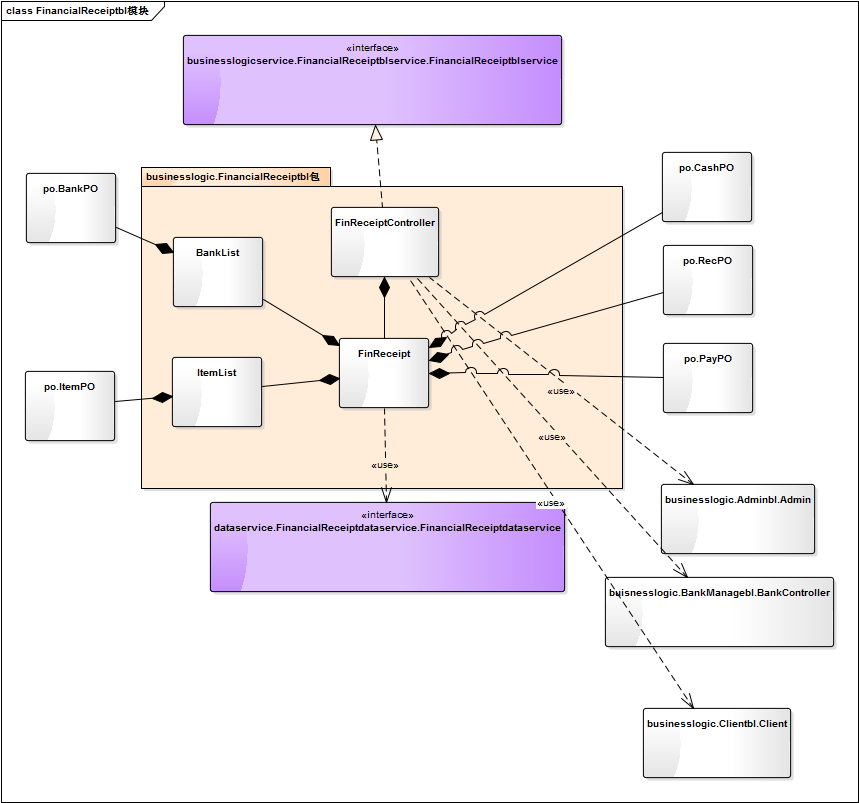
FinancialReceiptbl模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

FinancialReceiptbl模块的职责及接口参见软件系统结构描述文档表10.

（2）整体结构

Businesslogicservice.FinancialReceiptblservice.FinancialReceiptblservice为 展示层和业务层之间的接口。 Dataservice.FinancialReceiptdataservice.FinancialReceiptdataservice为业 务层和数据层之间的接口。FinReceiptController负责逻辑控制职责，FinReceipt负 责业务逻辑职责。BankList作为银行账户的容器类，BankPO为银行账户的持久化数 据对象，这里作为转账的数据对象使用。ItemList作为条目的容器类，ItemPO为条 目的持久化数据对象。CashPO,RecPO,PayPO分别为现金费用单，收款单，付款单的 持久化数据对象。

1. FinancialReceiptbl模块各个类设计



1. FinancialReceiptbl模块各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| FinReceiptControll  er | 负责实现制定财务单据所需要的服务 |
| FinReceipt | 制定财务单据的领域模型对象，拥有财务单据的属性以及对数据库的引用，辅助完成制定财务单据所需要的服务 |
| BankList | 转账列表，拥有转账信息，辅助完成制定付款单，收款单 |
| ItemList | 条目清单，拥有条目信息，辅助完成制定现金费用单 |

（3）模块内部类的接口规范

1. FinReceiptCpmtroller的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| FinancialReceipt.Payment | 语法 | Public void makePayment(PayVo vo) throws Exception |
| 前置条件 | 无 |
| 后置条件 | 生成付款单，提交给总经理 |
| FinancialReceipt.Receive | 语法 | Public void makeReceive(RecVo vo) throws Exception |
| 前置条件 | 无 |
| 后置条件 | 生成收款单，提交给总经理 |
| FinancialReceipt.Cash | 语法 | Public void makeCash(CashVo vo) throws Exception |
| 前置条件 | 无 |
| 后置条件 | 生成现金费用单，提交给总经理 |
| FinancialReceipt.ClearBank | 语法 | public void clearBank(); |
| 前置条件 | 无 |
| 后置条件 | 清空BankList |
| FinancialReceipt.ClearItem | 语法 | public void clearItem() |
| 前置条件 | 无 |
| 后置条件 | 清空ItemList |
| FinancialReceipt.AddBank | 语法 | public double addBank(BkTransPO po) |
| 前置条件 | 发起了制定收款单，付款单的操作 |
| 后置条件 | BankList添加一条转账信息 |
| FinancialReceipt.AddItem | 语法 | public double addItem(ItemPO po) |
| 前置条件 | 发起了制定现金费用单的操作 |
| 后置条件 | ItemList添加一条条目信息 |
| FinancialReceipt.DelBank | 语法 | Public double deleteBank(int i) |
| 前置条件 | 发起了制定收款单，付款单的操作 |
| 后置条件 | BankList删除一条转账信息 |
| FinancialReceipt.DelItem | 语法 | Public double deleteItem(int i) |
| 前置条件 | 发起了制定现金费用单的操作 |
| 后置条件 | ItemList删除一条条目信息 |
| FinancialReceipt.GetBank | 语法 | public String[] getBank() throws Exception; |
| 前置条件 | 无 |
| 后置条件 | 获得可选用的银行账户列表 |
| FinancialReceipt.GetClient | 语法 | public String[] getClient() throws Exception |
| 前置条件 | 无 |
| 后置条件 | 获得可选用的客户列表 |
| FinancialReceipt.GetUser | 语法 | public String getOperator() throws Exception |
| 前置条件 | 无 |
| 后置条件 | 获得当前操作员 |
| FinancialReceipt.GetReceipt | 语法 | public ArrayList<ReceiptPO> getReceipt()throws Exception; |
| 前置条件 | 无 |
| 后置条件 | 获得所有未审批财务类单据 |
| FinancialReceipt.GetReceipt | 语法 | public ArrayList<ReceiptPO> getReceipt(ReceiptConditionVO vo)throws Exception |
| 前置条件 | 无 |
| 后置条件 | 根据筛选条件获得符合条件的财务类单据 |
| FinancialReceipt.CreditCash | 语法 | public void creditCash(CashPO po)throws Exception |
| 前置条件 | 无 |
| 后置条件 | 创建现金费用单红冲单 |
| FinancialReceipt.CreditRec | 语法 | public void creditRec(RecPO po)throws Exception |
| 前置条件 | 无 |
| 后置条件 | 创建收款单红冲单 |
| FinancialReceipt.CreditPay | 语法 | public void creditPay(PayPO po)throws Exception |
| 前置条件 | 无 |
| 后置条件 | 创建付款单红冲单 |
| FinancialReceipt.Update | 语法 | public void update(ReceiptPO[] po)throws Exception |
| 前置条件 | 无 |
| 后置条件 | 批量更新财务类单据的处理状态 |
| FinancialReceipt.Update | 语法 | public void update(ReceiptPO po)throws Exception |
| 前置条件 | 无 |
| 后置条件 | 更新一条财务类单据的处理状态 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| BankList.addBank(BankVO vo) | 添加一条转账信息 |
| ItemList.addItem(int x) | 添加一条条目信息 |
| BankList.deleteBank(BankVO vo) | 删除一条转账信息 |
| ItemList.deleteItem(int x) | 删除一条条目信息 |
| FinReceipt.makeCash(CashVO vo) | 生成现金费用单 |
| FinReceipt.makePayment(PayVo vo) | 生成付款单 |
| FinReceipt.makeReceive(RecVo vo) | 生成收款单 |
| FinReceipt.update(ReceiptPO[] list) | 更新财务类单据信息 |
| FinReceipt.queryReceipt(ReceiptCOndiitonVO vo) | 获得符合条件的财务类单据 |
| BankController.search(null) | 获得银行账户列表 |
| ClientUtilityImpl.queryClient(null) | get客户列表 |
| Admin.getUser() | get当前操作员 |

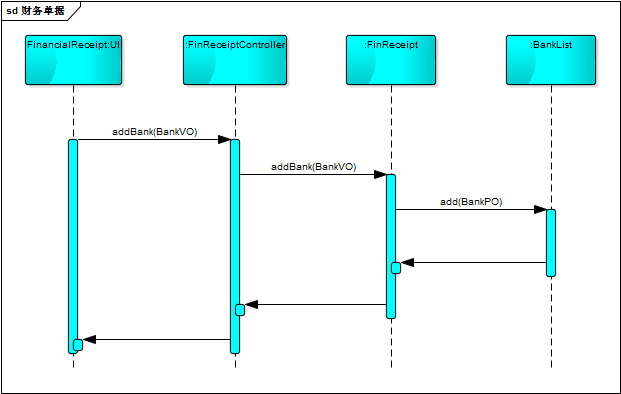
1. FinReceipt的接口规范

|  |  |  |
| --- | --- | --- |
| FinReceipt.update | 语法 | public ResultMessage update(ReceiptPO [] receipt) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 更新财务单据信息 |
| FinReceipt.query | 语法 | public ResultMessage queryReceipt(ReceiptConditionVO condition)throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 查询符合条件的财务单据 |
| FinReceipt.makePayment | 语法 | public ResultMessage makePayMent(PayPO po) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 生成付款单，提交给总经理 |
| FinReceipt.makeReceive | 语法 | public ResultMessage makeReceive(RecPO po)throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 生成收款单，提交给总经理 |
| FinReceipt.makeCash | 语法 | public ResultMessage makeCash(CashPO po)throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 生成现金费用单，提交给总经理 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| RMIUtility.getImpl(String) | 获得FinReceipt数据库服务的引用 |
| FinancialReceiptDataService.addPay(PayPo po) | 添加一个持久化对象PayPo |
| FinancialReceiptDataService.addRec(RecPo po) | 添加一个持久化对象RecPo |
| FinancialReceiptDataService.addCash(CashPo po) | 添加一个持久化对象CashPo |
| FinancialReceiptDataService.update(Receipt  Po[] receipt) | 更新数据库持久化对象信息 |
| FinancialReceiptDataService.queryReceipt(ReceiptConditionVO vo) | 获得符合条件的持久化对象 |

（4）业务逻辑层的动态模型

1. 转账列表中添加银行账户的顺序图



（5）业务逻辑层的设计原理

利用委托式控制风格，将FinancialReceipt的bl层的职责交给FinReceipt来实现，FinReceipt将部分职责委托给BankList,ItemList。每个界面需要访问的业务逻辑有FinReceiptController控制器委托给上述的领域对象。

### RunningTablebl模块

（1）模块概述

RunningTablebl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。

RunningTablebl模块的职责及接口参见软件系统结构描述文档表11.

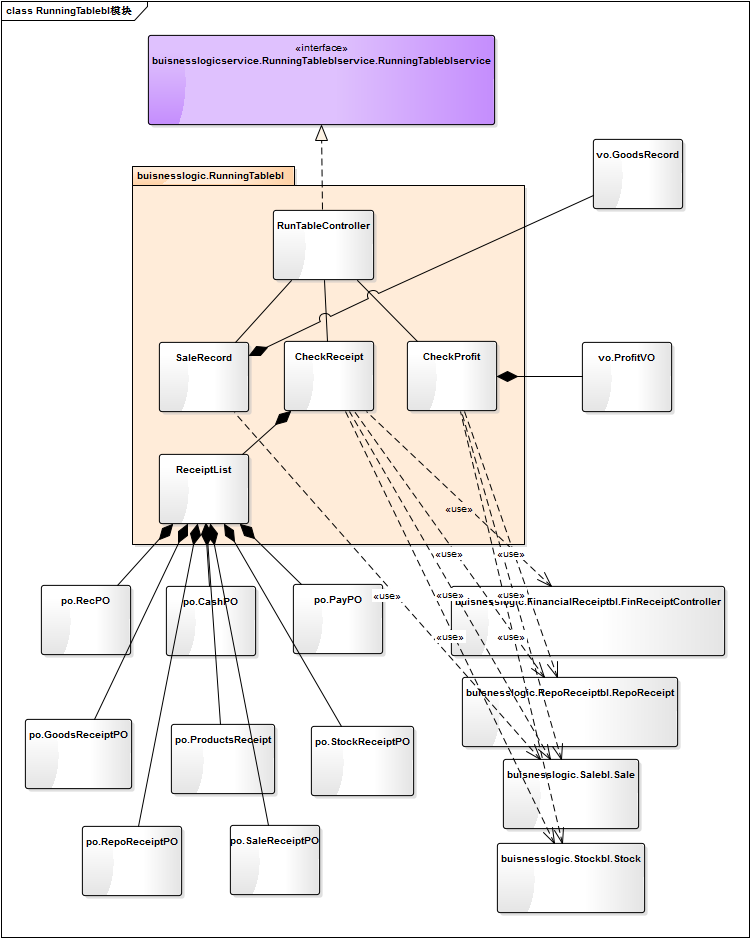
（2）整体结构

Businesslogicservice.RunningTableblservice.RunningTableblservice为展示层 和业务层之间的接口。RunTableController负责逻辑控制职责，业务逻辑职责由 SaleRecord,

CheckReceipt,CheckProfit来共同实现。GoodsRecord为销售记录的临时对象， ProfitVO为利润的临时对象，ReceiptList为所有单据的容器类，负责处理单据显示 的信息。RecPO,CashPO

,PayPO,GoodsRceiptPO,ProductsReceipt,StockReceiptPO,RepoReceiptPO,S aleReceiptPO为各种单据的持久化数据对象。

1. RunningTablebl模块各个类设计



1. RunningTablebl模块各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| RunTableController | 负责实现查看经营表格的所需要的服务 |
| SaleRecord | 销售明细，可以对商品销售明细进行查询，辅助实现查看经营表格的功能 |
| CheckReceipt | 查看单据，可以对所有单据进行查询，并提供红冲功能，辅助实现查看经营表格的功能 |
| CheckProfit | 收支，可以计算一段时间内的收支状况，辅助时间查看经营表格的功能 |
| ReceiptList | 单据列表，可以统计单据信息，进行单据分类，控制不同单据显示的内容，辅助实现查看经营历程表的功能 |

（3）模块内部类的接口规范

1. RunTableController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| RunningTable.SaleTable | 语法 | public GoodsRecordVO[] getSaleTable (SaleConditionVO vo) throws Exception |
| 前置条件 | 无 |
| 后置条件 | 返回符合条件的商品销售记录 |
| RunningTable.ProgressTable | 语法 | public ArrayList<ReceiptPO> getReceipt(ReceiptConditionVO vo)throws Exception |
| 前置条件 | 无 |
| 后置条件 | 返回符合条件的单据 |
| RunningTable.FinanceTable | 语法 | public ProfitVO getProfit(ReceiptConditionVO vo)throws Exception |
| 前置条件 | 无 |
| 后置条件 | 返回符合条件的单据 |
| RunningTable.ProgressTable.CreditNote | 语法 | public void CreditNote(ReceiptPO po)throws Exception |
| 前置条件 | 选中列表中某单据 |
| 后置条件 | 创建红冲单 |
| RunningTable.GetClient | 语法 | public String[] getClient() throws Exception |
| 前置条件 | 无 |
| 后置条件 | 返回可查询的客户列表 |
| RunningTable.GetOperator | 语法 | public String[] getOperator() throws Exception |
| 前置条件 | 无 |
| 后置条件 | 返回可查询的操作员列表 |
| RunningTable.GetGoods | 语法 | public String[] getGoodsList() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回可查询的商品列表 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| SaleRecord.getSaleTable(SaleConditionVO vo) | 根据条件查询销售记录 |
| CheckReceipt.getReceipt(ReceiptConditionVO vo) | 根据条件查询所有单据 |
| CheckProfit.getProfit(ReceiptConditionVO vo) | 根据条件查询经营情况（收入支出利润） |
| CheckReceipt.CreditNote(Receipt PO) | 生成一条红冲单 |
| ClientUtilityImpl.queryClient(null) | 查询可供选择的客户 |
| AdminController.getUser() | 获得当前操作员 |
| GL\_controller.stock\_type() | 查询可供选择的商品 |

1. SaleRecord的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SaleRecord.getSaleTable | 语法 | public GoodsRecordVO[] getSaleTable(SaleConditionVO vo) throws Exception |
| 前置条件 | 无 |
| 后置条件 | 返回符合条件的销售情况 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| SaleUtilityImpl.querySaleRecord(ReceiptConditionVO vo) | 查询销售情况 |

1. CheckReceipt的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CheckReceipt.getReceipt | 语法 | public ArrayList<ReceiptPO> getReceipt(ReceiptConditionVO vo)throws Exception |
| 前置条件 | 无 |
| 后置条件 | 返回符合条件的单据 |
| CheckReceipt.CreditNote | 语法 | public void CreditNote(ReceiptPO vo)throws Exception |
| 前置条件 | 该单据可建立红冲 |
| 后置条件 | 创建红冲单 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| RepoReceiptImpl.sendReceipt(ReceiptConditionVO vo) | 根据条件查询库存类单据 |
| SaleUtilityImpl.queryReceipt(ReceiptConditionVO vo) | 根据条件查询销售类单据 |
| StockUtilityImpl.queryReceipt(ReceiptConditionVO vo) | 根据条件查询进货类单据 |
| FinReceiptController.getReceipt(ReceiptConditionVO vo) | 根据条件查询财务类单据 |
| RepoReceiptBLImpl.produceGoodsReceipt(GoodsReceiptVO vo) | 生成库存赠送单 |
| SaleUtilityImpl.makeReceipt(SaleReceiptPO po) | 生成销售/销售退货单 |
| StockUtilityImpl.makeReceipt(SaleReceiptPO po) | 生成进货/进货退货单 |
| FinReceiptController.creditCash(CashVOvo) | 生成现金费用单 |
| FinReceiptController.creditPay(PayVO vo) | 生成付款单 |
| FinReceiptController.creditRec(RecVO  vo) | 生成收款单 |

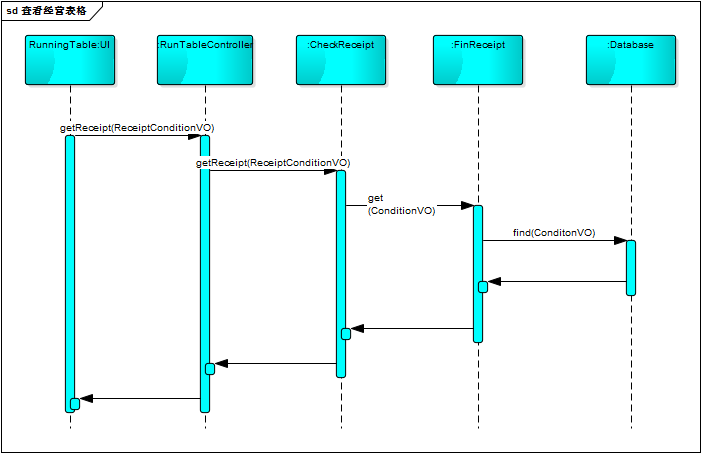
1. CheckProfit的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CheckProfit.getProfit | 语法 | public ProfitVO getProfit(ReceiptConditionVO vo)throws Exception |
| 前置条件 | 无 |
| 后置条件 | 返回收支情况 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| RepoCheckBLImpl.get\_aver\_in() | 获得最近一次库存盘点数据 |
| RepoReceiptBLImpl.sendReceipt(ReceiptConditionVO vo) | 查询符合条件的库存类单据 |
| GL\_controller.iSearch(String field) | 查询商品信息 |
| SaleUtilityImpl.queryReceipt(ReceiptConditionVO vo) | 查询符合条件的销售类单据 |
| StockUtilityImpl.queryReceipt(ReceiptConditionVO vo) | 查询符合条件的进货类单据 |

（4）业务逻辑层的动态模型

1. 查看财务单据的顺序图



(5)业务逻辑层的设计原理

利用委托式控制风格，将RunningTable的bl层的职责交给CheckReceipt,CheckProfit,SaleRecord来实现，CheckReceipt,CheckProfit,SaleRecord将获取单据的职责分配给其他Bl。每个界面需要访问的业务逻辑有RunTableController控制器委托给上述的领域对象。