### 4.1.2Financebl模块

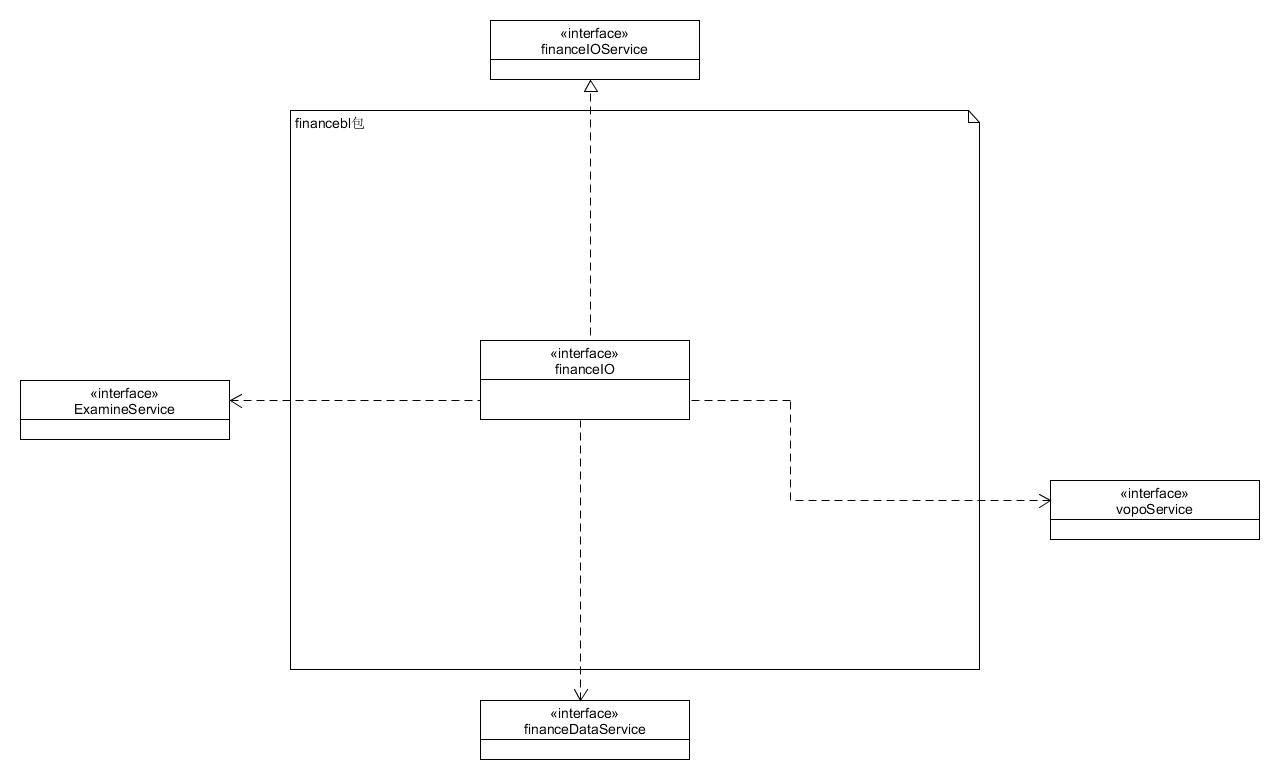
1. 模块概述

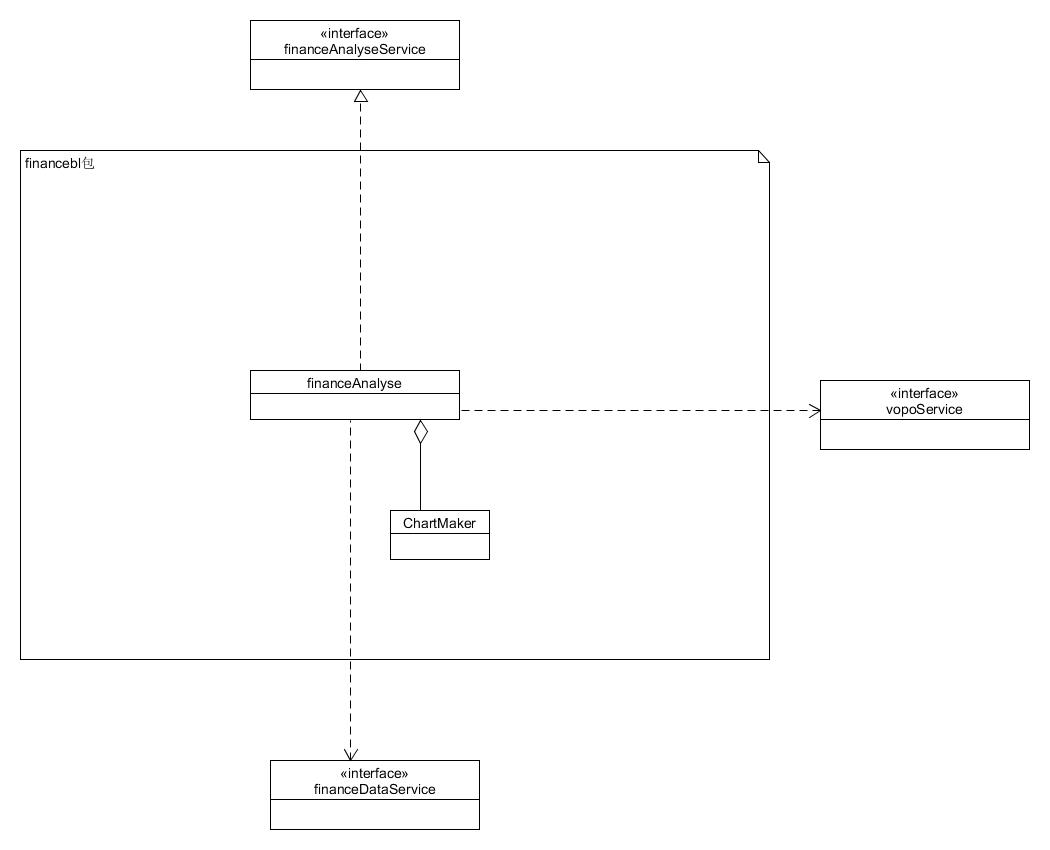
Financebl模块负责实现出入款单以及财务情况统计以及账户管理的服务

1. 整体结构

采用分治思想，将服务分为出入款单、财务分析、账户管理三个主要部分

Financebl模块的设计如图





Financebl模块各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| FinanceblAccount | 负责实现管理账户 |
| FinanceIO | 负责实现新建出入款单 |
| FinanceAnalyse | 负责实现财务分析 |

1. 模块内部类的接口规范

**FinanceblIO的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| FinanceIOService.loadOrder | 语法 | public OperationMessage loadOrder(String orderNumber) |
| 前置条件 | 无 |
| 后置条件 | 若存在对应订单号的OrderPO，载入到此对象中，返回正常信息；否则返回无此订单的提示 |
| FinanceIOService.  getNewRevenueID | 语法 | public String getNewRevenueID  (String date) |
| 前置条件 | 无 |
| 后置条件 | 返回未使用过的收款单的编号 |
| FinanceIOService.  getOrderVO | 语法 | public OrderVO getOrderVO() |
| 前置条件 | 已经调用过loadOrder且成功 |
| 后置条件 | 将加载的OrderPO包装成OrderVO返回 |
| FinanceIOService. checkFormat | 语法 | public List<CheckFormatMessage> checkFormat(RevenueVO rvo，boolean isFinal) |
| 前置条件 | 无 |
| 后置条件 | 检查收款单各项，若有错或不完整，将提示作为CheckFormatMessage作为列表返回；若无误，列表中只包含成功信息 |
| FinanceIOService. submit | 语法 | public OperationMessage submit(RevenueVO rvo) |
| 前置条件 | RevenueVO已经过检查 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| FinanceIOService. loadDraft | 语法 | public RevenueVO loadDraft() |
| 前置条件 | 无 |
| 后置条件 | 若对应序列化文件存在，解序列化成对应的RevenueVO对象返回；否则返回null |
| FinanceIOService. saveDraft | 语法 | public OperationMessage saveDraft(RevenueVO draft) |
| 前置条件 | 输入的RevenueVO不为空 |
| 后置条件 | 将RevenueVO序列化成文件存储，成功后返回成功信息；若之前有同名文件，返回已有草稿这个信息 |
| FinanceIOService. getRevenueVO | 语法 | public RevenueVO getRevenueVO(String date, String hallID) |
| 前置条件 | 两个参数已经经过检查 |
| 后置条件 | 返回对应的收款单PO包装过的VO对象 |
| FinanceIOService. getRevenueVO | 语法 | public RevenueVO getRevenueVO(String revenueID) |
| 前置条件 | 无 |
| 后置条件 | 返回对应的收款单PO包装过的VO对象 |
| FinanceIOService. getRevenueVOs | 语法 | public List<RevenueVO> getRevenueVOs(String startDate, String endDate) |
| 前置条件 | 两个参数已经经过检查 |
| 后置条件 | 返回对应天数的所有的收款单 |
| FinanceIOService. checkFormat | 语法 | public List<CheckFormatMessage> checkFormat(PaymentVO pvo，boolean isFinal) |
| 前置条件 | 输入的PaymentVO不为空 |
| 后置条件 | 检查输入信息是否符合系统要求，若符合，返回的list中只有一个显示正确的CheckFormatMessage, 否则返回能指示出错误的CheckFormatMessage的列表 |
| FinanceIOService. getNewPaymentID | 语法 | public String getNewPaymentID  (String date) |
| 前置条件 | 无 |
| 后置条件 | 返回未使用过的付款单的编号 |
| FinanceIOService. submit | 语法 | public OperationMessage submit(PaymentVO rvo) |
| 前置条件 | PaymentVO已经通过检查 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| FinanceIOService. loadDraft | 语法 | public PaymentVO loadDraft() |
| 前置条件 | 无 |
| 后置条件 | 若对应序列化文件存在，解序列化成对应的PaymentVO对象返回；否则返回null |
| FinanceIOService. saveDraft | 语法 | public OperationMessage saveDraft(PaymentVO draft) |
| 前置条件 | 输入的PaymentVO不为空 |
| 后置条件 | 将PaymentVO序列化成文件存储，成功后返回成功信息；若之前有同名文件，返回已有草稿这个信息 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| FormatCheckService.  checkTransportHallID  (String number) | 检查营业厅编号是否存在 | |
| FormatCheckService.  checkIsNull(String in) | 检查输入是否为空 | |
| FormatCheckService.  checkDate(String date) | 检查日期格式是否正确，是否在系统记录时间内 | |
| FormatCheckService.  checkMoney(String money) | 检查金额格式是否正确 | |
| FinanceFormDataService.  updateRevenuePOs(String staffID) | 将服务器端的staffID对应队列的收款单同步到本地副本 | |
| FinanceFormDataService.  updatePaymentPOs(String staffID) | 将服务器端的staffID对应队列的付款单同步到本地副本 | |

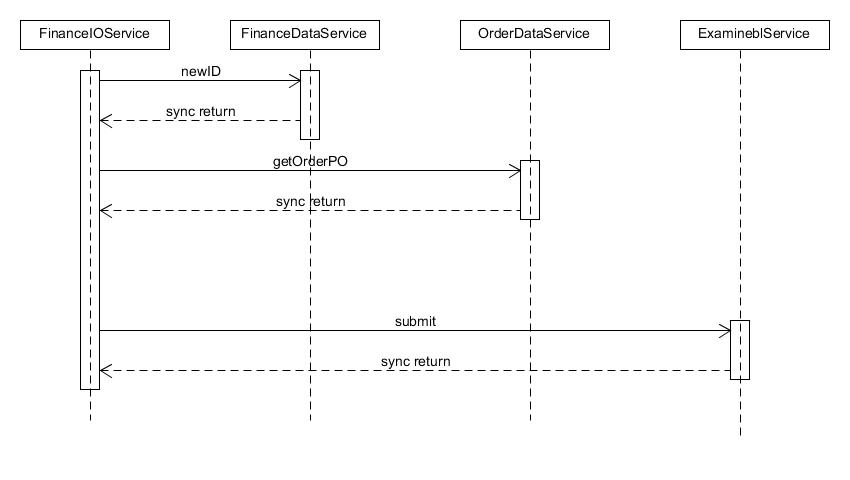
**FinanceAnalyse接口规范**

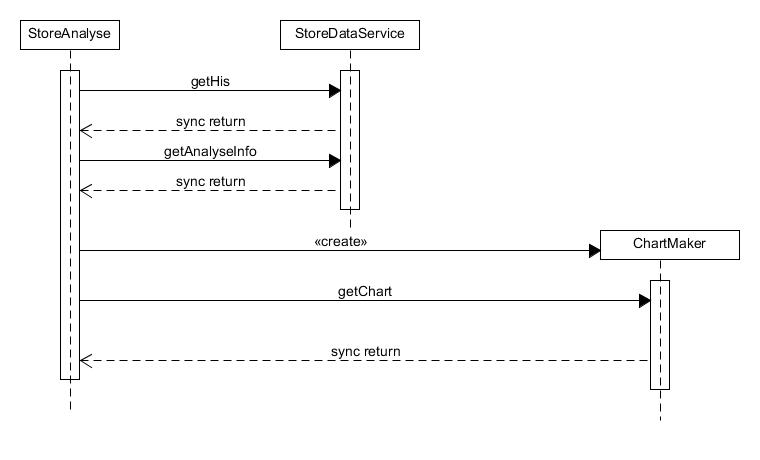
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| FinanceAnalyseService. getHistogram | 语法 | public BaseChartVO getHistogram() |
| 前置条件 | 无 |
| 后置条件 | 将本地的收款单和付款单的数据汇总，生成直方图的VO对象返回 |
| FinanceAnalyseService. getPieChart | 语法 | public PieChartVO getPieChart() |
| 前置条件 | 无 |
| 后置条件 | 将本地的收款单和付款单的数据汇总，生成饼状图的VO对象返回 |
| FinanceAnalyseService. getLineChart | 语法 | public BaseChartVO getLineChart() |
| 前置条件 | 无 |
| 后置条件 | 将本地的收款单和付款单的数据汇总，生成折线图的VO对象返回 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| FinanceFormDataService.  getRevenuePO(String formID) | 根据收款单编号获取付款单的PO | |
| FinanceFormDataService.  getPaymentPO(String formID) | 根据付款单编号获取付款单的PO | |
| BankAccountDataService.  updateAccountOperations  (String staffID) | 将服务器端的staffID对应队列的账号操作同步到本地副本 | |
| BankAccountDataService.  uploadAccountOperations  (String staffID, List<BankAccountOperation> operations) | 将本地副本的队列的新增账号操作同步到服务器端staffID对应的队列中 | |
| BankAccountDataService.  downloadAllAccounts() | 将服务器端所有的账号下载到本地 | |

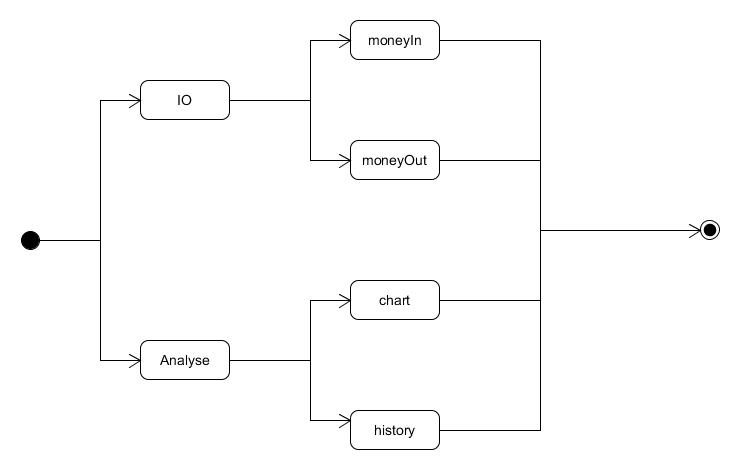
**FinanceManage接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| FinanceManageService. getAllAccounts | 语法 | public List<BankAccountVO> getAllAccounts() |
| 前置条件 | 无 |
| 后置条件 | 返回所有的账号的VO对象的列表 |
| FinanceManageService. filterAccounts | 语法 | public List<BankAccountVO> filterAccounts(List<BankAccountVO> list, String s) |
| 前置条件 | 无 |
| 后置条件 | 用字符串s对list中所有的BankAccountVO的名称筛选，将模糊匹配的对象合成另一个列表返回 |
| FinanceManageService. addAccount | 语法 | public OperationMessage addAccount(BankAccountVO avo) |
| 前置条件 | avo已经通过检查 |
| 后置条件 | 将avo添加到本地副本，此项操作上传到服务端，返回是否成功的信息 |
| FinanceManageService. deleteAccount | 语法 | public OperationMessage deleteAccount(BankAccountVO avo) |
| 前置条件 | avo在本地存在 |
| 后置条件 | 将avo从本地副本删除，此项操作上传到服务端，返回是否成功的信息 |
| FinanceManageService. editAccount | 语法 | public OperationMessage editAccount(BankAccountVO avo, String newName) |
| 前置条件 | avo在本地存在，newName通过检查 |
| 后置条件 | 将本地副本的avo的名称改成新名称，此项操作上传到服务器，返回是否成功的信息 |
| FinanceManageService.  getTradeHistory | 语法 | public List<PaymentVO > getTradeHistory(BankAccountVO avo) |
| 前置条件 | avo 在系统中存在 |
| 后置条件 | 返回与该账号相关的所有的付款单 |
| FinanceManageService.pay | 语法 | public OperationMessage pay(String bankAccID, String amount) |
| 前置条件 | 无 |
| 后置条件 | 将bankAccID对应的账户进行支付操作，返回成功或失败原因相关信息 |
| FinanceManageService.  receive | 语法 | public OperationMessage receive(String bankAccID, String amount) |
| 前置条件 | 无 |
| 后置条件 | 将bankAccID对应的账户进行入账操作，返回成功或失败原因相关信息 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| BankAccountDataService.  downloadAllAccounts() | 将服务器端所有的账号下载到本地 | |
| BankAccountDataService.  getNewBankID() | 获取未使用过的银行账户ID编号 | |
| BankAccountDataService.  checkIsNameUsed  (String name) | 检查银行账户名是否已被占用 | |
| BankAccountDataService.  getBankAccount  (String bankID) | 根据传入的银行账户ID编号获取银行账户PO | |
| OrderDataService.  getOrderPO(String orderID) | 获取指定订单号的订单的数据 | |

1. 业务逻辑层的动态模型







1. 业务逻辑层的设计原理

采用集中式的控制