# 快递物流系统(EMDS)软件详细设计描述文档V1.1

目录

**1引言**

**2产品概述**

**3体系结构设计概述**

**4结构视角**

4.1业务逻辑层的分解

4.1.1 financeBL模块

4.1.2 LogisticsBL模块

4.1.3 LogBL模块

4.1.4 ManageBL模块

4.1.5 AdminBL模块

4.1.6 InventoryBL模块

4.1.7 BusinessHallBL模块

4.1.8 DeliveryBL模块

4.1.9 IntermediateBL模块

4.1.10 LoginBL模块

1. **引言**

**1.1编制目的**

本报告详细完成对快递物流系统的详细设计，达到指导后续软件构造的目的，同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

**1.2词汇表**

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| EDMS | 快递物流管理系统 | ExpressDeliveryManagementSystem |
| RMI | 远程接口调用 |  |
| ui | 表示某展示层 |  |
| bl | 表示某业务逻辑层 |  |
| data | 表示某数据层 |  |

**1.3参考资料**

快递物流管理系统用例图，概念类图，状态图，顺序图

快递物流管理系统用例文档V1.1

快递物流系统软件需求规格说明V1.1

项目实践V4

软件工程与计算(卷二)软件开发的技术基础

度量数据文档

系统测试文档

快递物流系统体系结构文档

1. **产品概述**

快递物流管理系统EDMS是为本地民营物流企业，该开发的目标是帮助该本地的民营物流企业处理日常的重点任务，包括揽件送件、物流信息相关操作、财务管理、库存管理、人员管理。

通过快递物流管理系统(EDMS)的应用，期望为该本地的民营物流企业提高工作人员的工作效率、降低库存运营成本、提高用户的满意度，提高收益。

1. **体系结构设计概述**

参考快递物流系统概要设计文档中对体系结构设计的概述

1. **结构视角**

4.1业务逻辑层的分解

业务逻辑层开发包图详细参见软件体系结构文档图3

* + 1. financeBL模块

1. 模块描述

financeBL模块承担的需求参见需求规格说明文档的功能需求以及相关非功能需求

financeBL模块的职责及接口参见软件系统结构描述文档表

1. 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，每一层之间为了增加灵活性，我们会添加接口对应的BLService的接口，业务逻辑层和数据层之间对应的dataService接口，为了隔离业务逻辑层和逻辑控制职责我们增加了对应的BLController 这样各类BLController将会对财务业务逻辑的职责和逻辑控制处理委托给financeBL对象。AccountPO cashregisterPO paymentPO作为财务操作产生的持久化对象被添加到设计模型中去的。

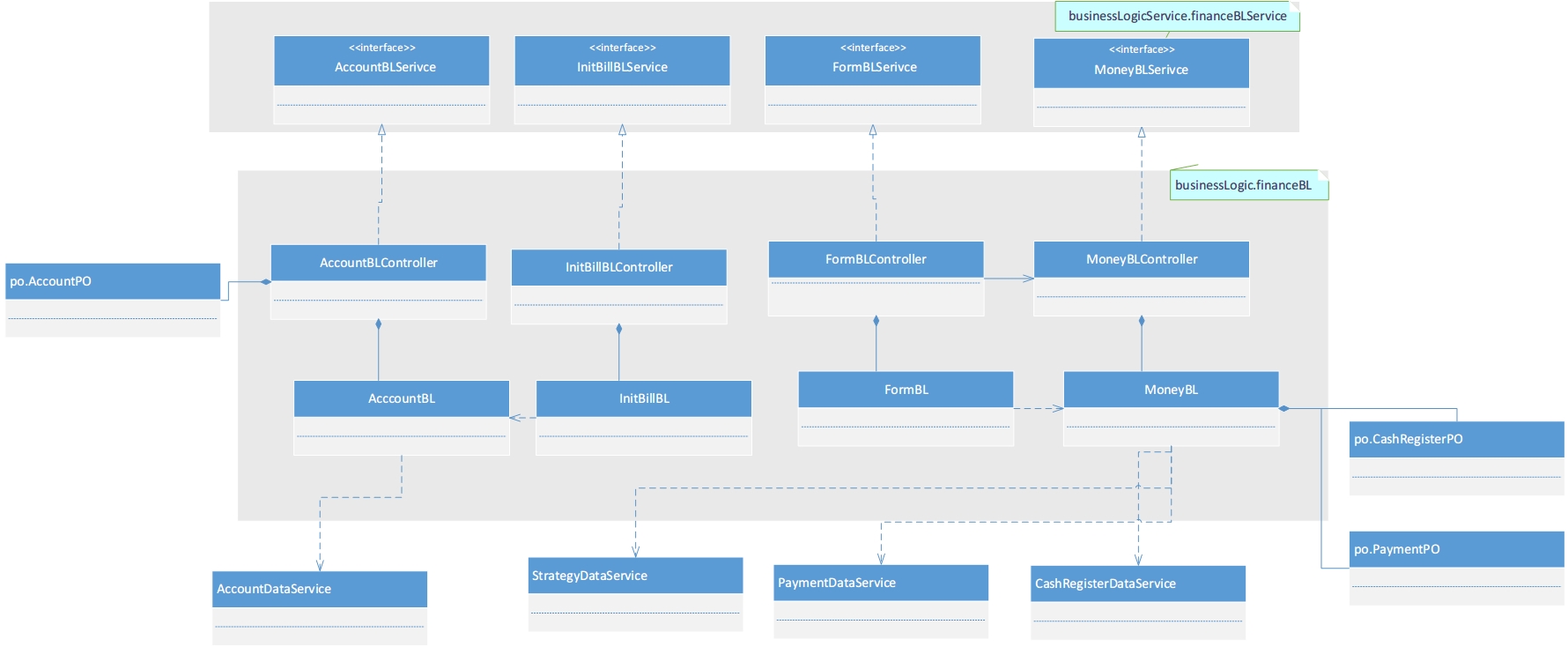


图 financeBL模块的个各类的设计

表 financeBL模块的各个类的职责

|  |  |
| --- | --- |
| **模块** | **职责** |
| loginBLController | 负责实现对应于登陆界面所需要的服务 |
| AccountBLController | 负责实现银行账户管理的所需要的服务 |
| InitBillBLController | 负责实现期初建账以及账本查看的所需要的服务 |
| FormBLController | 负责实现相关报表查看的所需要的服务 |
| MoneyBLController | 负责实现收款单和付款单的相关操作的所需要的服务 |
| logBLController | 负责实现对应的将操作人员的操作记录所需要的服务 |
| AccountBL | 银行账户操作的领域模型对象，拥有一次银行账户的增删改查的账户信息，可以帮助完成银行账户操作界面所需要服务 |
| MoneyBL | 收付款操作的领域模型对象，包括可以进行收款单查看和付款单填写和查看的所需要的服务 |
| FormBill | 报表显示的领域模型对象，可以根据不同的选择生成对应的经营情况表和成本收益表，帮助界面完成所需要的服务 |
| InitBillBL | 期初建账的领域模型对象，可以帮助完成期初建账界面所需要的服务 |

1. 模块内部类的接口规范

AccountBLController 、InitBillBLController、FormBLController、 MoneyBLController和financeController的接口规范如下表

AccountBLController的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| AccountBLController.addAccount | 语法 | public String addAccount(String name, String accountID) |
| 前置条件 | 已经创建一个AccountBL领域对象，并要新建账户，且输入符合规则 |
| 后置条件 | 调用AccountBL领域对象的addAccount方法，并新建账户 |
| AccountBLController.deleteAccount | 语法 | public boolean deleteAccount(AccountVO vo) |
|  | 前置条件 | 已经创建了一个AccountBL领域对象，并要删除账户 |
|  | 后置条件 | 调用AccountBL领域对象的deleteAccount方法 |
| AccountBLController.searchAccount | 语法 | public List<AccountVO> searchAccount(String key) |
|  | 前置条件 | 已经创建一个AccountBL领域对象，并要查找账户 |
|  | 后置条件 | 调用AccountBL领域对象的searchAccount(String key)方法，返回该账户 |
| AccountBLController.modifyAccount | 语法 | public String modifyAccount(AccountVO vo) |
|  | 前置条件 | 已经创建一个AccountBL领域对象，并要修改账户 |
|  | 后置条件 | 调用AccountBL领域对象，对选中的账户进行修改 |
| AccountBLController.getAccounts | 语法 | public List<AccountVO> getAccounts() |
|  | 前置条件 | 已经创建了一个AccountBL领域对象 |
|  | 后置条件 | 调用领域对象的getAccounts方法 获得账户数据持久化对象的集合，然后显示在界面上 |
| **需要的服务（需接口）** | | |
| AccountBL.addAccount | 增加一个银行账户 | |
| AccountBL.deleteAccount | 删除一个银行帐户 | |
| AccountBL.modifyAccount | 修改一个银行账户 | |
| AccountBL.searchAccount | 查账银行账户,通过关键字进行查找 | |
| AccountBL.getAccounts | 获取银行账户信息列表供显示 | |
| LogDataService.add | 加入一条新的操作日志 | |

InitBillBLController的接口规范**还没写？**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| InitBillBLController.initBill | 语法 | public void initBill(String name) |
| 前置条件 | 已经创建一个InitBillBL领域对象,需要初期建账操作 |
| 后置条件 | 传入一个新的账户的名字 然后可以进行一份期初建账 |
| **需要的服务（需接口）** | | |
| initBillBL.initBill | 执行期初建账操作 | |

FormBLController接口的规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| FormBLController.getIncomes | 语法 | public List<IncomeVO> getIncomes() |
| 前置条件 | 已经创建一个FormBL领域对象，想要显示收款单 |
| 后置条件 | 返回与收款单相关的所有信息，并在界面显示 |
| FormBLController.getPayments | 语法 | public List<PaymentVO> getPayments() |
|  | 前置条件 | 已经创建了一个FormBL领域对象，想要显示付款单 |
|  | 后置条件 | 返回与付款单相关的所有信息，并在界面显示 |
| FormBLContoller.exportRunForm | 语法 | public void exportRunForm(Date start,Date end) |
|  | 前置条件 | 已经创建了一个FormBL 领域对象,并要导出某段时期的经营管理表 |
|  | 后置条件 | 导出该时间段的经营管理表到本地 |
| FormBLController.exportCostForm | 语法 | public void exportCostForm() |
|  | 前置条件 | 已经创建了一个FormBL对象 并要导出某段时期的成本收益表 |
|  | 后置条件 | 导出该时间段的成本收益表到本地 |
| FormBLController.getIncomesbyT | 语法 | public List<IncomeVO> getIncomes(Date start, Date end) |
|  | 前置条件 | 已经创建一个FormBL领域对象，想要显示某段时期的收款单，时间输入正确 |
|  | 后置条件 | 返回该段时期与收款单相关的所有信息，并在界面显示 |
| FormBLController.getPaymentsbyT | 语法 | public List<PaymentVO> getPayments(Date start, Date end) |
|  | 前置条件 | 已经创建一个FormBL领域对象，想要显示某段时期的付款单，时间输入正确 |
|  | 后置条件 | 返回该段时期与付款单相关的所有信息，并在界面显示 |
| FormBLController.timeValid | 语法 | public String isValid(String start, String end) |
|  | 前置条件 | 得到时间输入 |
|  | 后置条件 | 判断输入的时间是否正确（符合规范） |
| **需要的服务（需接口）** | | |
| FormBL.exportRunForm | 导出经营管理表 | |
| FormBL.exportCostForm | 导出成本收益表 | |
| FormBL.getIncomes | 获得收款单列表供显示 | |
| FormBL.getPayments | 获得付款单列表供显示 | |
| FormBL.getIncomes | 获得某段时间内收款单列表供查找并显示 | |
| FormBL.getPayments | 获得某段时间内付款单列表供查找并显示 | |
| FormBL.isValid | 检查时间输入是否正确 | |

PaymentBLController接口的规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| PaymentBLController.addPayment | 语法 | public void addPayment(PaymentVO vo) |
|  | 前置条件 | 已经创建一个PaymentBL领域对象，并要建立新付款单，且输入符合规则 |
|  | 后置条件 | 调用PaymentBL领域对象的addPayment方法，并新建账户 |
| PaymentBLController.findByDate | 语法 | public List<PaymentPO> findByDate(Date d) |
|  | 前置条件 | 已经创建一个PaymentBL领域对象，并要查找某时间至今的付款单 |
|  | 后置条件 | 返回该时间到目前为止的所有付款单信息 |
| PaymentBLController.betweenDate | 语法 | public List<PaymentPO> betweenDate(Date start, Date end) |
|  | 前置条件 | 已经创建一个PaymentBL领域对象，并要查找某时间段的付款单 |
|  | 后置条件 | 返回所有符合该时间日期的所有的付款单的信息 |
| PaymentBLController.getTotal | 语法 | public double getSalaryTotal() |
|  | 前置条件 | 已经创建一个PaymentBL领域对象，并要得到付款总额 |
|  | 后置条件 | 计算并返回付款单涉及的总金额 |
| **需要的服务（需接口）** | | |
| PaymentBL.addPayment | 新建一个付款单 | |
| PaymentBL.findByDate | 查找该时间到目前为止所有的付款单 | |
| PaymentBL.betweenDate | 查找该时间段所有的付款单 | |
| PaymentBL.getSalaryTotal | 计算付款单涉及的总金额 | |
| LogBL.add | 加入一条新的操作日志 | |

AccountBL的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| AccountBL.addAccount | 语法 | | public String addAccount(String name, String accountID) |
| 前置条件 | | 财务人员发出增加账户操作，启动一个增加账户的操作 |
| 后置条件 | | 系统增加一个银行账户 |
| AccountBL.deleteAccount | 语法 | | public boolean deleteAccount(AccountPO po) |
|  | 前置条件 | | 财务人员发出删除银行账户的请求，启动一个删除账户的操作 |
|  | 后置条件 | | 系统删除选定的银行账户 |
| AccountBL.modifyAccount | 语法 | | public String modifyAccount(AccountVO vo) |
|  | 前置条件 | | 财务人员发出修改账户的请求，启动一个修改账户的操作 |
|  | 后置条件 | | 系统进行相关账户的修改 |
| AccountBL.searchAccount | 语法 | | public List<AccountVO> searchAccount(String key) |
|  | 前置条件 | | 财务人员发出关键字查找账户的请求，启动查账账户的操作 |
|  | 后置条件 | | 系统进行查找返回符合条件的po列表 |
| AccountBL.getAccounts | 语法 | | public List<AccountVO> getAccounts() |
|  | 前置条件 | | 启动一个账户管理操作回合 |
|  | 后置条件 | | 系统将所有的AcccountPO获取传输到客户端 |
| AccountBL. enter | 语法 | | public String enter(double fare) |
|  | 前置条件 | | 已建立新的收款单，账户得到资金转入 |
|  | 后置条件 | | 系统修改账户余额 |
| AccountBL. pay | 语法 | | public String pay(double fare) |
|  | 前置条件 | | 已建立新的付款单，账户产生资金支出 |
|  | 后置条件 | | 已建立新的收款单，账户得到资金转入 |
| AccountBL. changeDefaultCard | 语法 | | public String changeDefaultCard(AccountVO newAccount) |
|  | 前置条件 | | 已设置默认账户，且存在其他账户 |
|  | 后置条件 | | 系统将目标账户设置为默认账户 |
| **需要的服务（需接口）** | | | |
| AccountDataService.getAccounts | | 从可持久化数据中获得所有的银行账户信息列表 | |
| AccountDataService.add | | 从可持久化数据中增加一个银行账户 | |
| AccountDataService.delete | | 删除一个可持久化数据中的银行帐户 | |
| AccountDataService.modify | | 修改一个可持久化数据中的银行账户 | |
| AccountDataService. search | | 查账银行账户,通过关键字在可持久化数据中进行查找 | |

InitBillBL接口规范**还没写？**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| AccountBL.initBill | 语法 | public Boolean initBill(String name) |
|  | 前置条件 | 财务人员进行期初建账，输入新建的账单的所要依托的银行账户即可（需要存在） |
|  | 后置条件 | 系统初期建账成功，各种单据的数据进行清空成功 |
|  |  |  |
| **需要的服务（需接口）** | | |
| dataFactoryService.InitBill | 将data层的数据更新为期初建账的状态 并保存 | |
| AccountDataService.search | 根据账户名查找对象的账户数据持久化对象 | |
|  |  | |

formBL接口规范

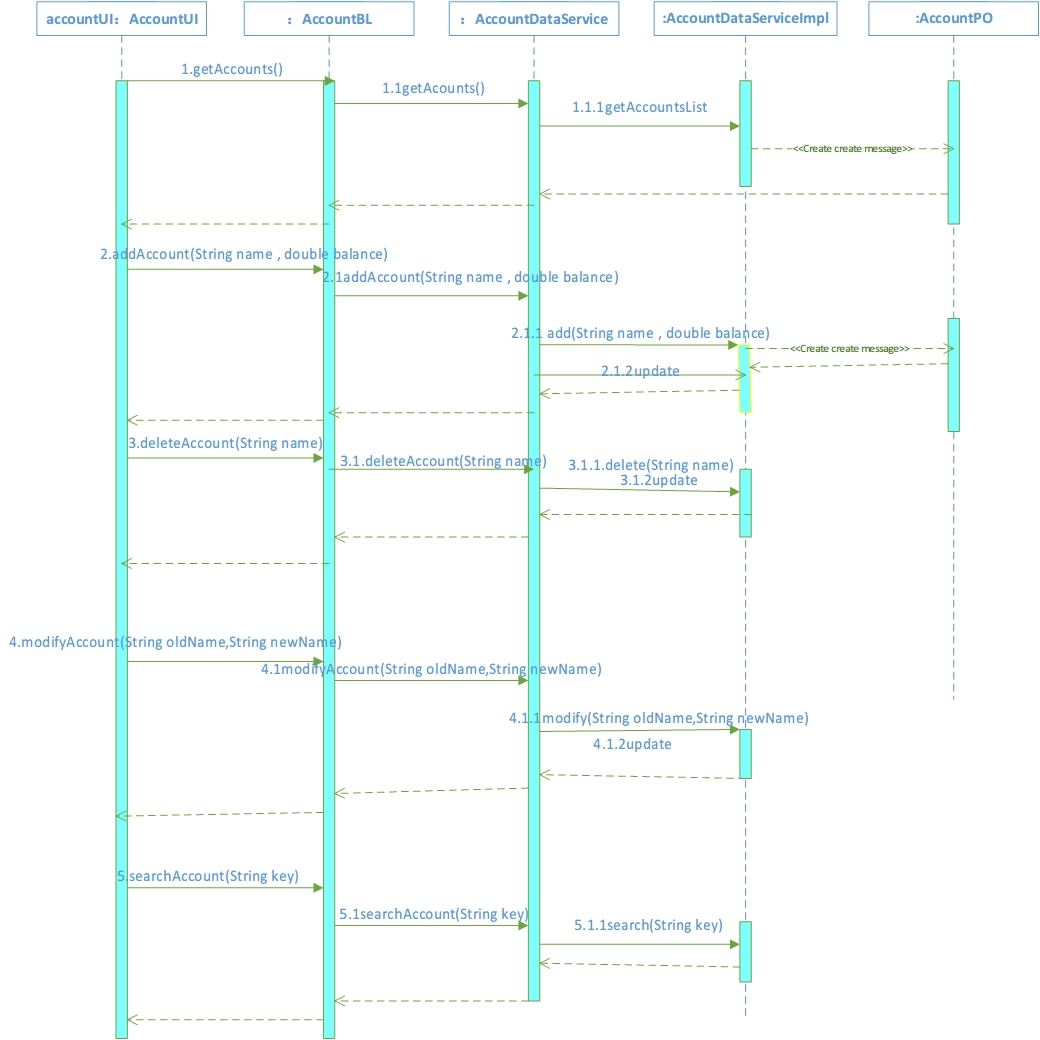
|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| FormBL.runForm | 语法 | public void exportRunForm(Date start, Date end) |
|  | 前置条件 | 财务人员导出经营情况表 输入正确的开始和结束时间 |
|  | 后置条件 | 系统显示一个起始时间的经营情况表 |
| FormBL.costForm | 语法 | public void exportCostForm() |
|  | 前置条件 | 财务人员导出成本收益表 |
|  | 后置条件 | 系统显示成本收益表 |
| FormBL.getIncomes | 语法 | public void getIncomes() |
|  | 前置条件 | 财务人员查看收款单 |
|  | 后置条件 | 系统显示所有的收款单信息 |
| FormBL.getPayments | 语法 | public void getPayments() |
|  | 前置条件 | 财务人员查看付款单 |
|  | 后置条件 | 系统显示所有的付款单信息 |
| FormBL.getIncomesByT | 语法 | public void getIncomes(Date start, Date end) |
|  | 前置条件 | 财务人员查看某段时间的收款单 |
|  | 后置条件 | 系统显示该段时间内所有的收款单信息 |
| FormBL.getPaymentsByT | 语法 | public void getPayments(Date start, Date end) |
|  | 前置条件 | 财务人员查看某段时间的付款单 |
|  | 后置条件 | 系统显示该段时间内所有的付款单信息 |
| FormBL.timeValid | 语法 | public String isValid(String start, String end) |
|  | 前置条件 | 输入时间需要检查时间是否输入正确 |
|  | 后置条件 | 得到时间是否输入正确的结果 |
| FormBL.IncomeList | 语法 | public List<IncomeVO> getIncomeList() |
|  | 前置条件 | 已经存在多个收款单 |
|  | 后置条件 | 建立并返回一个收款单列表 |
| FormBL.PaymentList | 语法 | public List<PaymentVO> getPaymentList() |
|  | 前置条件 | 已经存在多个付款单 |
|  | 后置条件 | 建立并返回一个付款单列表 |
| FormBL.export | 语法 | private void export(File file, HSSFWorkbook data) |
|  | 前置条件 | 文件和工作簿的数据已存在，需要导出excel |
|  | 后置条件 | 将该工作簿数据写入文件中 |
| FormBL.createRunForm | 语法 | private HSSFWorkbook createRunForm() |
|  | 前置条件 | 需要导出经营管理表Excel |
|  | 后置条件 | 将已有经营管理相关数据转为工作簿的数据 |
| FormBL.createCostForm | 语法 | private HSSFWorkbook createCostForm() |
|  | 前置条件 | 需要导出成本收益表Excel |
|  | 后置条件 | 将已有成本收益相关数据转为工作簿的数据 |
| FormBL.incomeTotal | 语法 | private double incomeTotal() |
|  | 前置条件 | 已存在多个收款单 |
|  | 后置条件 | 计算收款单总额 |
| FormBL.paymentTotal | 语法 | private double payTotal() |
|  | 前置条件 | 已存在多个付款单 |
|  | 后置条件 | 计算付款单总额 |
| **需要的服务(需接口)** | | |
| LogDataService.add | 在持久化数据中增加一条操作日志 | |
| CashRegisterBL.getIncomeList | 得到收款单列表 | |
| PaymentBL.getPaymentList | 得到付款单列表 | |
| CashRegisterBL.getIncomeBetweenDate | 得到该段时间内所有的收款单 | |
| PaymentBL.betweenDate | 查找该段时间内所有的付款单 | |

paymentBL接口规范

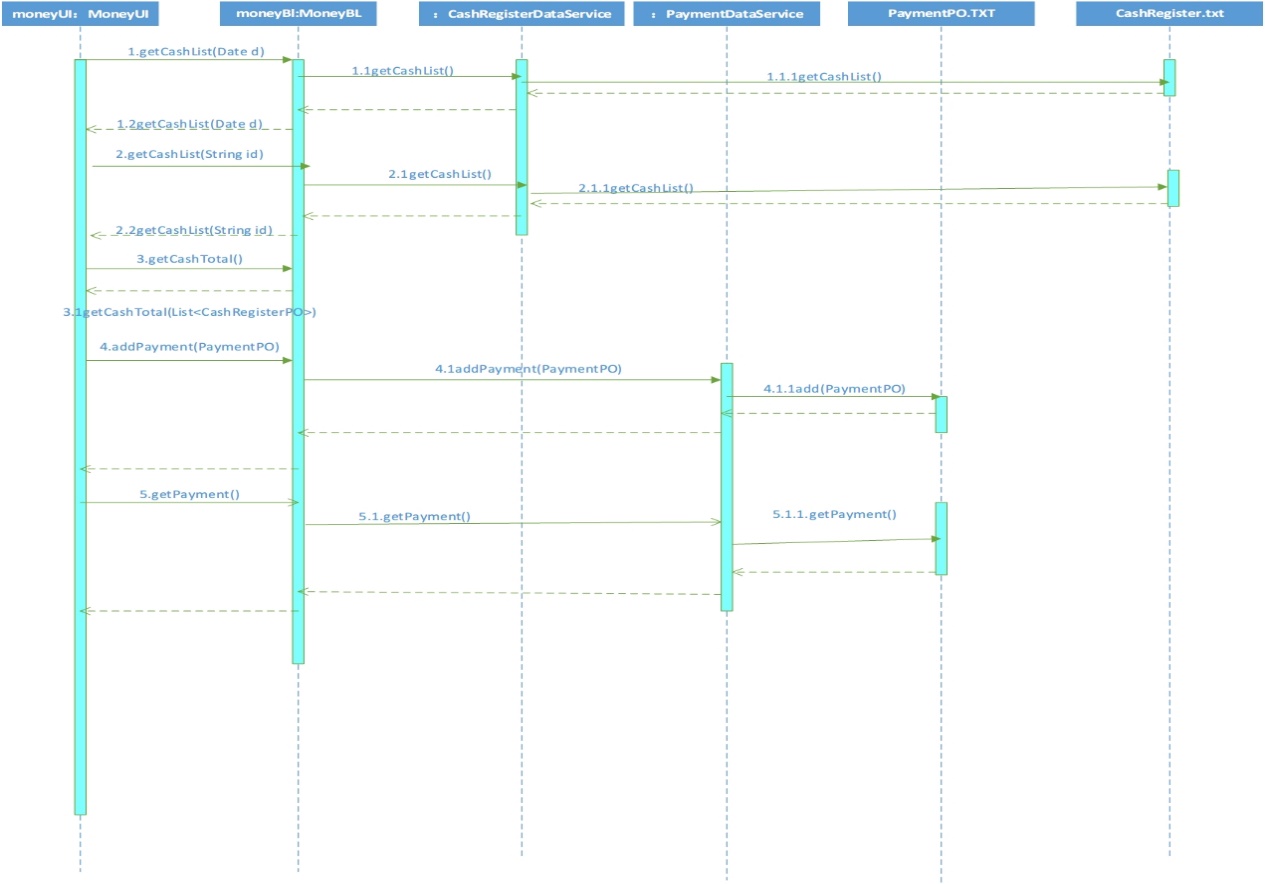
|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| PaymentBL. getPaymentList | 语法 | public List<PaymentPO> getPaymentList() |
|  | 前置条件 | 已存在多个付款单 |
|  | 后置条件 | 系统新建并返回一个付款单列表 |
| PaymentBL. addPayment | 语法 | public void addPayment(PaymentPO po) |
|  | 前置条件 | 财务人员需要新建一个付款单 |
|  | 后置条件 | 系统增加一个付款单 |
| PaymentBL. findByDate | 语法 | public List<PaymentPO> findByDate(Date d) |
|  | 前置条件 | 财务人员查看某时间至今的所有付款单信息 |
|  | 后置条件 | 系统显示该时间至今的所有付款单信息 |
| PaymentBL. betweenDate | 语法 | public List<PaymentPO> betweenDate(Date start,Date end) |
|  | 前置条件 | 财务人员查看某时间段的所有付款单信息 |
|  | 后置条件 | 系统显示该时间段的所有付款单信息 |
| PaymentBL. getAmmount | 语法 | public double getAmmount(List<PaymentPO> paymentList) |
|  | 前置条件 | 已存在一个付款单列表 |
|  | 后置条件 | 系统计算并返回已经支出的付款单总额 |
| PaymentBL. getSalaryTota | 语法 | public double getSalaryTotal() |
|  | 前置条件 | 财务人员将要根据付款单进行付款 |
|  | 后置条件 | 系统计算并返回将要支出的付款单金额 |
| PaymentBL. approve | 语法 | public void approve(PaymentPO po) |
|  | 前置条件 | 付款单未通过审批 |
|  | 后置条件 | 系统将该付款单状态改为已通过审批 |
| PaymentBL. approveAll | 语法 | public void approveAll() |
|  | 前置条件 | 付款单未通过审批 |
|  | 后置条件 | 系统将所有付款单状态改为已通过审批 |
| PaymentBL. getUnapproved | 语法 | public List<PaymentVO> getUnapproved() |
|  | 前置条件 | 存在未通过审批的付款单 |
|  | 后置条件 | 系统返回未通过审批的付款单列表 |
| **需要的服务(需接口)** | | |
| LogDataService.add | 在持久化数据中增加一条操作日志 | |
| PaymentDataService.getList | 获取持久化数据的付款单list | |
| PaymentDataService.add | 在持久化数据中增加一个付款单 | |
| Payment DataService.find | 在持久化数据中查找一个付款单 | |
| PaymentDataService.approval | 在持久化数据中将一个付款单状态改为审批通过 | |
| PaymentDataService.approvalAll | 在持久化数据中将所有付款单状态改为审批通过 | |
| StaffBL. getUsers | 得到所有的员工信息 | |
| SalaryBL. getSalaryPO | 得到一个员工工资PO | |
| SalaryBL.getSalaryByWork | 得到对应员工工资信息 | |

（4）FinanceBL层的动态模型

下图表示了快递物流系统中，当用户发出银行账户管理相关操作，Account相关对象之间的协作

图 银行账户增删改查的顺序图

下图表示了快递物流系统中当用户发出了相关付款单查看新建操作和收款单查看操作，Money相关对象之间的协作

图 首付款单相关操作的顺序图

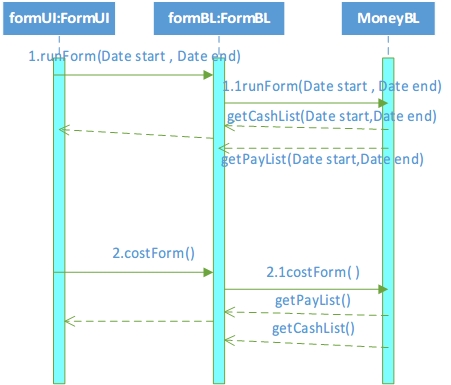
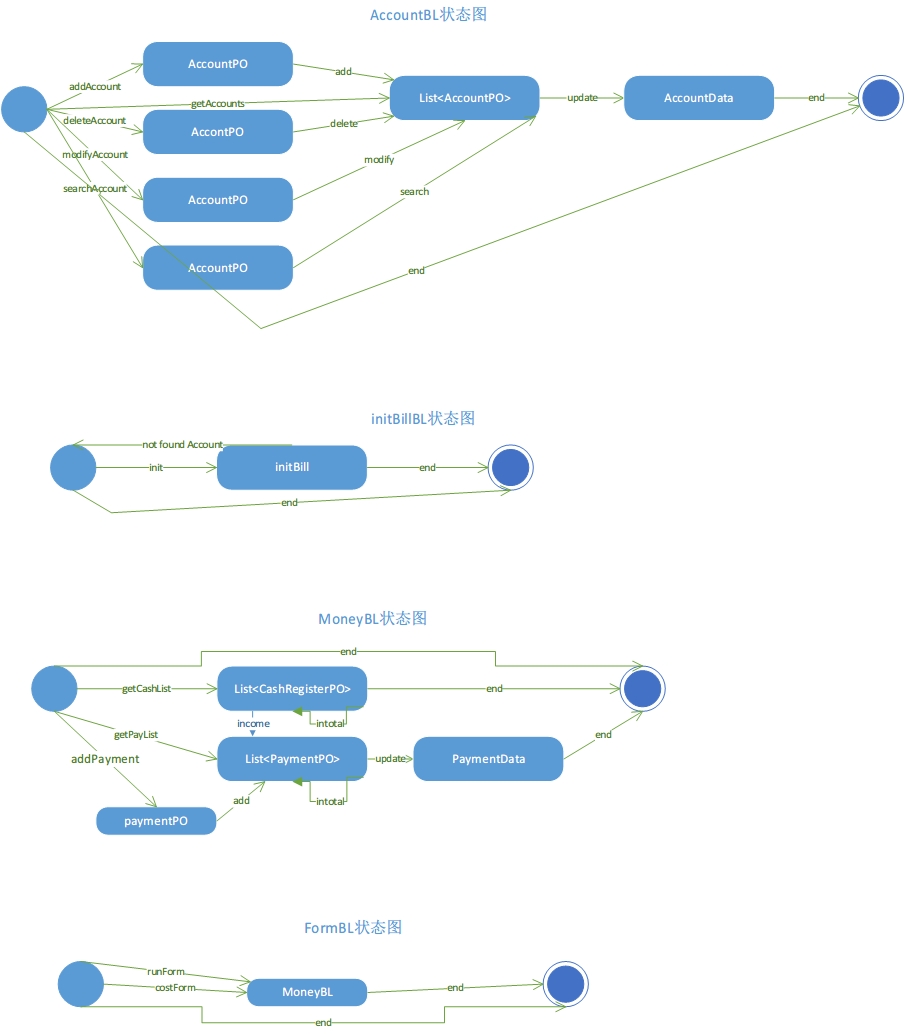
下图表示了快递物流系统中，当用户发出查看报表相关操作，Form相关对象之间的协作

图 报表操作顺序图

FinanceBL包中各BL对象的状态图



4.1.2 LogisticsBL模块

1. 模块概述

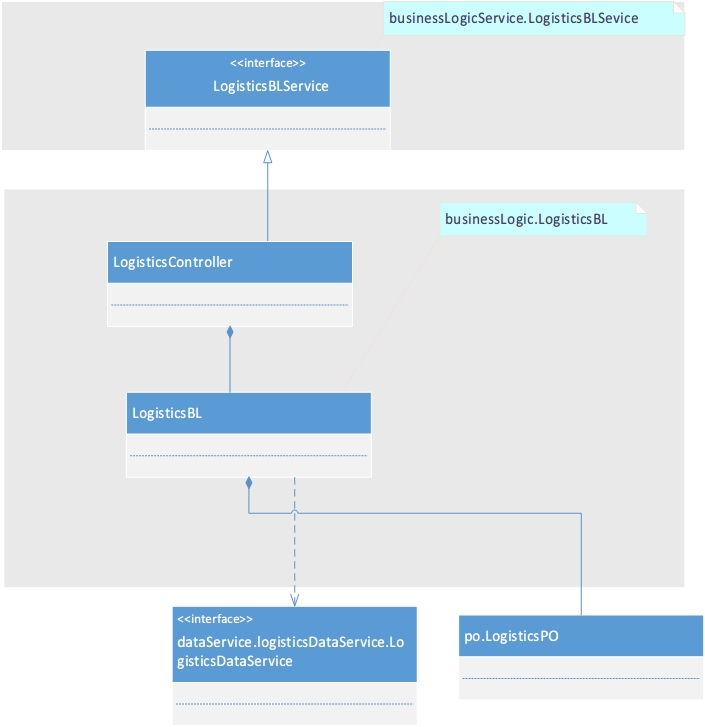
LogisticsBL模块承担的需求参见需求规格说明文档功能需求以及相关非功能需求

LogisticsBL 模块的职责及接口参见软件系统结构描述文档

1. 根据系统结构的设计，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，比如在展示层和业务逻辑层之间我们添加了businessLogiscService.LogisticsBLService.LogisticsBLService接口，业务逻辑层和数据层之间添加了dataService.logisticsDataService.LogisticsDataService接口，LogisticsPO作为快递的物流信息的持久化对象会被添加到设计模型中去。为了隔离业务逻辑职责和逻辑控制职责，我们添加了LogisticsControler,这样LogisticsController会将对物流信息查询的业务逻辑处理委托给LogisticsBL对象。

LogisticsBL模块的设计如图所示

图 logisticsBL模块的各个类设计



LogisticsBL模块的各个类的职责如下表所示

**LogisticsBL模块的个各类的职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| LogisticsController | 负责实现对应于查询、增加、更新、物流信息所需要的服务 |
| LogisticsBL | 查询物流信息的领域模型对象拥有一次查询服务的信息，可以帮助完成物流查询界面所需要的服务 |

1. 模块内部类的接口规范

LogisticsController 和 LogisticsBL的接口规范如下表

LogisticsController接口规范

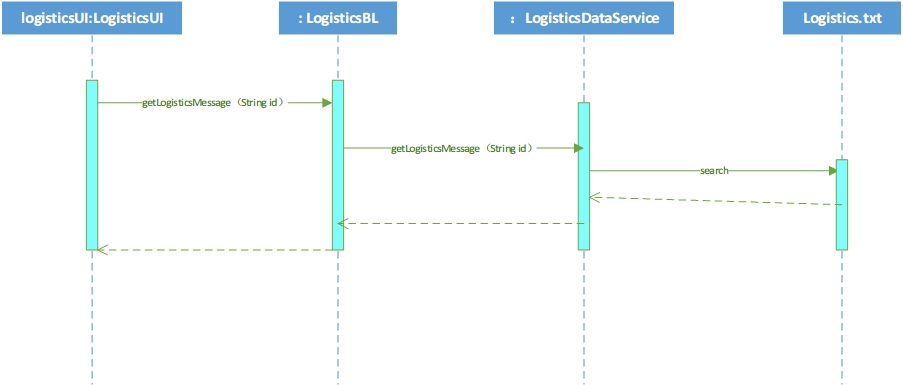
|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| LogisticsController.getLogisticsMessage | 语法 | public LogisticsVO getLogisticsMessage(String id) |
|  | 前置条件 | 拥有一个LogisticsBL领域模型对象，并且输入符合规范 |
|  | 后置条件 | 调用LogisticsBL的getLgisticsMessage(String id)方法,查找对应货物的物流信息 |
| LogisticsController.add | 语法 | public String add(LogisticsPO po) |
|  | 前置条件 | 新建了一个LogisticsBL领域模型对象 |
|  | 后置条件 | 在系统中增加该货物的物流信息po |
| LogisticsController.update | 语法 | public String update(String ID,String message) |
|  | 前置条件 | 拥有一个LogisticsBL领域模型对象，获得新的物流信息 |
|  | 后置条件 | 在系统中更新该货物的物流信息 |
| **需要的服务(需接口)** | | |
| LogisticsBL.getLogisticsMessage | 查找符合输入的快递单号的物流信息PO | |
| logisticsBL.add | 新建一个快递单的物流信息PO | |
| logisticsBL.update | 更新一个快递单的物流信息PO | |

表 LogisticsBL的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| logisticsBL.add | 语法 | public String add(LogisticsPO po) |
|  | 前置条件 | 快递员新建寄件单，需要新建该寄件单对应的物流信息 |
|  | 后置条件 | 系统新建一个物流信息PO |
| logisticsBL.update | 语法 | public String update(String ID,String message) |
|  | 前置条件 | 货物到达新的位置，物流信息更新 |
|  | 后置条件 | 系统更新该货物的物流信息PO |
| LogisticsBL.getLogisticsMessage | 语法 | public LogisticsVO getLogisticsMessage(String id) |
|  | 前置条件 | 客户需要查找物流信息且ID输入正确 |
|  | 后置条件 | 系统根据ID返回对应的物流信息PO |
| **需要的服务(需接口)** | | |
| logisticsDS.getLogisticsList | 可持久化数据中所有的物流信息PO加入一个列表 | |

1. LogisticsBL的动态模型

下表为Logistics想要获知与输入的ID相匹配的物流信息的顺序图

图 查询物流信息的顺序图

由于查询物流信息只有一个界面只要这个界面退出 对应的LogisticsBL对象就会自动的回收

每次查询都需要建立一次Logistics领域模型 因此该领域模型状态图在此处省略。

4.1.3 LogBL模块

（1）模块概述

LogBL模块承担的需求参见需求规格说明文档功能需求以及相关非功能需求

LogBL 模块的职责及接口参见软件系统结构描述文档

（2）整体结构

根绝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了businenssLogicSerivice.logBLService.LogBLService.业务逻辑层和数据层之间添加dataService.logDataService.LogDataDervice。为了隔离业务逻辑职责和逻辑控制职责，我们增加了LogController，这样LogController会将对日志查看的业务逻辑处理委托给LogBL对象。

LogPO 是做为日志记录的持久化对象被添加到设计模型里面去的。

LogBL模块的设计如下图所示。

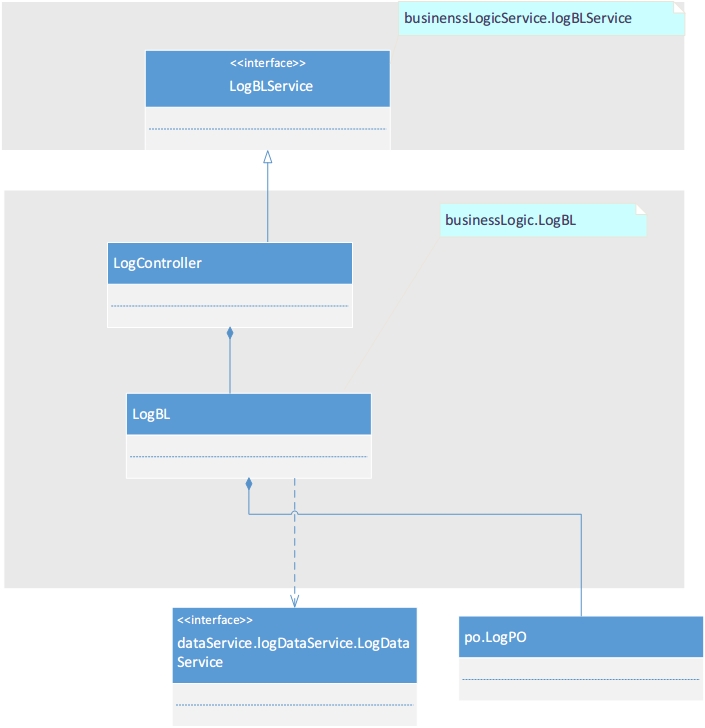


图 LogBL模块的各个类的设计

LogBL模块的各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| LogController | 负责实现对应显示操作日志界面所需要的服务 |
| LogBL | 日志查看的领域模型对象，拥有一次查看日志的所有的相关日至操作信息，可以帮助完成日志查看界面所需要的服务 |

（3）模块内部类的接口规范

LogController 和LogBL的接口规范如下表所示

表 LogController 的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| LogController.getAllList | 语法 | public List<LogVO> getAllLogList() |
|  | 前置条件 | 已经创建了一个LogBL的领域模型对象 |
|  | 后置条件 | 调用该对象的getAllLogList方法,查看所有的操作日志 |
| LogController. getLogListOfDate | 语法 | public List<LogVO> getLogListOfDate(Date d) |
|  | 前置条件 | 已经创建了一个LogBL的领域模型对象 |
|  | 后置条件 | 调用该对象的getLogListOfDate方法,查看该日期至今所有的操作日志 |
| LogController. getLogListOfUser | 语法 | public List<LogVO> getLogListOfUser(String userID) |
|  | 前置条件 | 已经创建了一个LogBL的领域模型对象 |
|  | 后置条件 | 调用该对象的getAllLogList方法,查看该ID对应员工所有的操作日志 |
| LogController. add | 语法 | public String add(LogVO vo) |
|  | 前置条件 | 已经创建了一个LogBL的领域模型对象 |
|  | 后置条件 | 调用该对象的add方法,增加一条操作日志 |
| **需要的服务(需接口)** | | |
| logBL.getAllLogList | 返回所有的日志信息 LogPO的一个列表 | |
| logBL.getLogOfDate |  | |
| logBL.getLogOfUser |  | |
| logBL.add |  | |

表 LogBL的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| LogBL.getLogList | 语法 | public List<LogVO> getAllLogList() |
|  | 前置条件 | 财务人员或者总经理需要查看操作日志 |
|  | 后置条件 | 返回所有操作日志的PO对象进行显示 |
| LogBL. getLogOfDate | 语法 | public List<LogVO> getLogOfDate(Date d) |
|  | 前置条件 | 财务人员或者总经理需要查看某时间至今的操作日志 |
|  | 后置条件 | 返回该时间至今所有的操作日志的PO对象进行显示 |
| LogBL. getLogOfUser | 语法 | public List<LogVO> getLogOfUser(String userID) |
|  | 前置条件 | 财务人员或者总经理需要查看某员工的操作日志 |
|  | 后置条件 | 返回该员工所有的操作日志的PO对象进行显示 |
| LogBL. add | 语法 | public String add(LogVO vo) |
|  | 前置条件 | 员工进行了新的操作 |
|  | 后置条件 | 系统将这一操作加入到操作日志中 |
| **需要的服务(需接口)** | | |
| LogDataService.getLogList | 将可持久化数据中的操作日志PO加入到一个列表 | |
| LogDataService.add | 将一条新的操作日志加入到可持久化数据中 | |

（4）LogBL层的动态模型

下图表示快递物流系统中，当财务人员或者总经理点击查看操作日志时，日志业务逻辑的相关对象之间的协作顺序图



LogBL对象由于只是提供一些数据给显示层，从而可以得出其生命周期只存在于显示操作日志中，因此其状态图在此省略。

1. 利用委托式的控制风格，每个界面需要访问的业务逻辑各自的控制器交给不同的领域对象

4.1.4 manageBL模块

1. 模块描述

manageBL模块承担的需求参见需求规格说明文档的功能需求以及相关非功能需求

manageBL模块的职责及接口参见软件系统结构描述文档表

1. 整体结构

根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了ManageBLService和ManageDataService两个接口。为了隔离业务逻辑职责和逻辑控制职责，我们添加了对应的Controller，这样各类Controller将会将管理的业务逻辑职责和逻辑控制委托给ManageBL对象。UserPO,InstPO,umentPO是作为管理信息需要和产生的持久化对象被添加到设计模型中的。

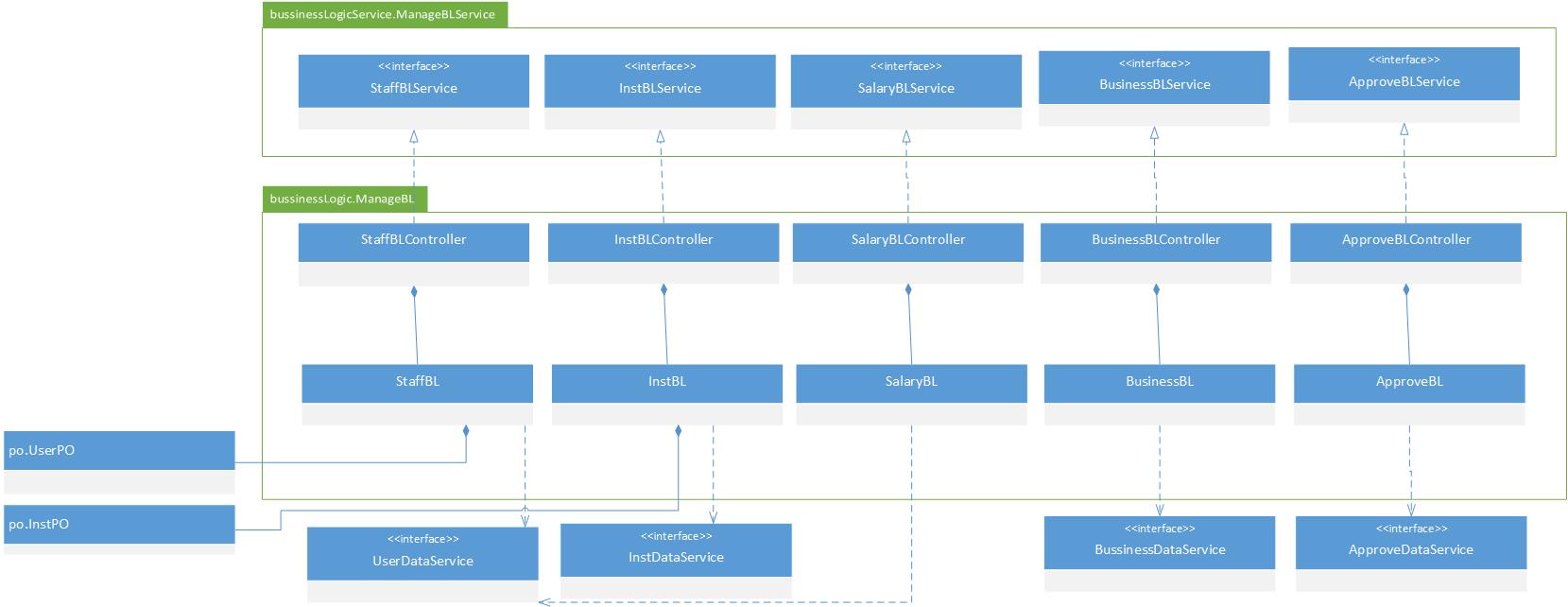


图 manageBL模块的各个类的设计

表 manageBL模块的各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| loginController | 负责实现对应于登陆界面所需要的服务 |
| StaffController | 负责实现人员信息管理的所需要的服务 |
| InstitutionController | 负责实现机构信息管理的所需要的服务 |
| SalaryController | 负责实现薪水策略制定的所需要的服务 |
| StrategyController | 负责实现价格距离常量策略制定的所需要的服务 |
| ApproveController | 负责实现审批单据的相关操作的所需要的服务 |
| logController | 负责实现对应的将操作人员的操作记录所需要的服务 |
| StaffBL | 人员信息管理的领域模型对象，可以帮助完成人员信息管理界面所需要服务 |
| InstitutionBL | 机构信息管理的领域模型对象，可以帮助完成机构信息管理界面所需要服务 |
| SalaryBL | 薪水策略制定的领域模型对象，可以帮助完成薪水策略制定界面所需要服务 |
| StrategyBL | 业务策略制定的领域模型对象，可以帮助完成业务策略制定界面所需要服务 |
| ApproveBL | 审批单据的领域模型对象，可以帮助完成审批单据界面所需要服务 |

1. 模块内部类的接口规范

StaffController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StaffController.getUser | 语法 | public UserVO getUser(String userID) |
|  | 前置条件 | 已经创建一个StaffBL领域对象，并要查看人员信息，且输入符合规则 |
|  | 后置条件 | 调用StaffBL领域对象的getUser方法 |
| StaffController.addUser | 语法 | public String addUser(String id,UserVO vo) |
|  | 前置条件 | 已经创建了一个StaffBL领域对象，并要新增人员信息，且输入符合规则 |
|  | 后置条件 | 调用StaffBL领域对象的addUser方法 |
| StaffController.updateUser | 语法 | public String updateUser(String id,UserVO vo) |
|  | 前置条件 | 已经创建了一个StaffBL领域对象，并要更改人员信息，且输入符合规则 |
|  | 后置条件 | 调用StaffBL领域对象的updateUser方法 |
| StaffController.deleteUser | 语法 | public boolean deleteUser(String id) |
|  | 前置条件 | 已经创建了一个StaffBL领域对象，并要删除人员信息 |
|  | 后置条件 | 调用StaffBL领域对象的deleteUser方法 |
| StaffController.getUsers | 语法 | public List<UserVO> getUsers() |
|  | 前置条件 | 已经创建了一个StaffBL领域对象 |
|  | 后置条件 | 获取所有的人员User的信息列表 |
| StaffController.getUsersOfIns | 语法 | public List<UserVO> getUsersOfIns(String InstitutionID) |
|  | 前置条件 | 已经创建了一个StaffBL领域对象，输入的机构号正确 |
|  | 后置条件 | 获取该机构编号对应的机构的所有的人员User的信息列表 |
| 需要的服务（需接口） | | |
| LogBL.add(Date date,String userID ,String message) | 增加一条log日志 | |
| StaffBL.getUser(String id) | 根据id获取相应人员信息 | |
| StaffBL.addUser(String id,UserVO vo) | 增加一个人员信息 | |
| StaffBL.updateUser(String id,UserVO vo) | 修改一个人员信息 | |
| StaffBL.deleteUser(String id) | 删除一个人员信息 | |
| StaffBL.getUsers() | 获取所有的人员信息 | |
| StaffBL.getUsersOfIns(String InstitutionID) | 根据传入的institutionID后的该机构的User信息 | |

InstitutionController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| InstitutionController.show | 语法 | public List<InstitutionVO> show () |
|  | 前置条件 | 已经创建一个InstitutionBL领域对象，并要查看机构信息 |
|  | 后置条件 | 调用InstitutionBL领域对象的show方法 |
| InstitutionController.addInst | 语法 | public String addIns(InstitutionVO vo) |
|  | 前置条件 | 已经创建了一个InstitutionBL领域对象，并要新增机构信息（营业厅），且输入符合规则 |
|  | 后置条件 | 调用InstitutionBL领域对象的addInst方法 |
| InstitutionController.addInst | 语法 | public String addIns(InstitutionVO vo,InventoryVO ivo) |
|  | 前置条件 | 已经创建了一个InstitutionBL领域对象，并要新增机构信息（中转中心），且输入符合规则 |
|  | 后置条件 | 调用InstitutionBL领域对象的addInst方法 |
| InstitutionController.modify | 语法 | public String modify(String InstitutionID,String newName) |
|  | 前置条件 | 已经创建了一个InstitutionBL领域对象，并要更改机构信息(机构名称)，且输入符合规则 |
|  | 后置条件 | 调用InstitutionBL领域对象的modify方法 |
| InstitutionController.modify | 语法 | public String modify(String InventoryID, InventoryVO vo) |
|  | 前置条件 | 已经创建了一个InstitutionBL领域对象，并要更改机构信息(仓库信息)，且输入符合规则 |
|  | 后置条件 | 调用InstitutionBL领域对象的modify方法 |
| InstitutionController.deleteIns | 语法 | public String deleteIns(InstitutionVO vo) |
|  | 前置条件 | 已经创建了一个InstitutionBL领域对象，并要删机构信息 |
|  | 后置条件 | 调用InstitutionBL领域对象的deleteIns方法 |
| 需要的服务（需接口） | | |
| LogBL.add(Date date,String userID ,String message) | 增加一条log日志 | |
| InstitutionBL.getIns(String id) | 根据id获取相应机构信息 | |
| InstitutionBL.addIns(InstitutionVO vo,InventoryVO ivo) | 增加一个机构信息（中转中心） | |
| InstitutionBL.addIns(InstitutionVO vo) | 增加一个机构信息(营业厅) | |
| InstitutionBL.modify(String inventryID,InventoryVO vo) | 修改机构信息(库存修改) | |
| InstitutionBL. modify(String institutionID,String newName) | 修改一个机构信息(机构名称修改) | |
| InstitutionBL.deleteInst(String id) | 删除一个机构信息 | |

SalaryController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SalaryController.getSalary | 语法 | public SalaryVO getSalary() |
| 前置条件 | 已经创建一个SalaryBL领域对象，并要查看人员薪水策略信息 |
| 后置条件 | 调用SalaryBL领域对象的getSalary方法 |
| SalaryController.modify | 语法 | public boolean updateUserSalary(SalaryVO) |
| 前置条件 | 已经创建了一个SalaryBL领域对象，并要更改人员薪水信息，且输入符合规则 |
| 后置条件 | 调用SalaryBL领域对象的modify方法 |
| 需要的服务（需接口） | | |
| LogBL.add(Date date,String userID ,String message) | 增加一条log日志信息 | |
| SalaryBL.getSalaryPO() | 获取工作人员的薪水信息 | |
| SalaryBL.modify(SalaryVO vo) | 根据修改相应职业人员的薪水信息 | |

StrategyController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StrategyController.getVO | 语法 | public StrategyVO getVO() |
|  | 前置条件 | 已经创建一个StrategyBL领域对象，并要查看当前业务策略信息 |
|  | 后置条件 | 调用StrategyBL领域对象的getVO方法 |
| StrategyController.modify | 语法 | public String modify(PriceConstVO vo) |
|  | 前置条件 | 已经创建了一个StrategyBL领域对象，并要更改业务策略信息，且输入符合规则 |
|  | 后置条件 | 调用StrategyBL领域对象的modify方法 |
| StrategyController.getCities | 语法 | public List<String> getCities() |
|  | 前置条件 | 已经创建了一个StrategyBL领域对象，并要获取目前所有城市的列表 |
|  | 后置条件 | 调用StrategyBL领域对象的getCities方法 |
| StrategyController.getDistance | 语法 | public double getDistance(String city1,String city2) |
|  | 前置条件 | 已经创建了一个StrategyBL领域对象，并要获取两城市的距离 |
|  | 后置条件 | 调用StrategyBL领域对象的getCities方法 |
| 需要的服务（需接口） | | |
| LogBL.add(Date date,String userID ,String message) | 增加一条log日志信息 | |
| StrategyBL.getVO | 获取当前的价格距离常量的信息VO | |
| StrategyBL.modify(PriceConst vo) | 修改当前业务策略信息 | |
| StrategyBL.getCities | 获得当前的距离常量定义中所有的城市信息列表 | |
| StrategyBL.getDistance(Stirng city1,String city2) | 从当前的价格常量信息中查找两个城市的距离 | |

ApproveController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ApproveController.getUnapproveSend/Arrival/Income/Delivery/Entruck/Entry/Shipment/Transfer/Receive | 语法 | public List<ReceiptsVO> getUnApprove\*\*\*\*() |
| 前置条件 | 已经创建一个ApproveBL领域对象，并要查看当前待审批的单据 |
| 后置条件 | 调用ApproveBL领域对象的getUnApproveList方法 |
| ApproveController.setApprovedSend/Arrival/Income/Delivery/Entruck/Entry/Shipment/Transfer/Receive | 语法 | public viod setApproved\*\*\*\*(ReceiptsVO vo) |
| 前置条件 | 已经创建了一个ApproveBL领域对象，并要通过某个单据审批 |
| 后置条件 | 调用ApproveBL领域对象的setApproved方法 |
| ApproveController.setAllApprovedSend/Arrival/Income/Delivery/Entruck/Entry/Shipment/Transfer/Receive | 语法 | public void setAllApproved\*\*\*\*() |
| 前置条件 | 已经创建了一个ApproveBL领域对象，并要通过全部某类型的单据的审批 |
| 后置条件 | 调用ApproveBL领域对象的setAllApproved方法 |
| ApproveController.modifySend/Arrival/Income/Delivery/Entruck/Entry/Shipment/Transfer/Receive | 语法 | public void modify\*\*\*\*(ReceiptsVO vo) |
| 前置条件 | 已经创建了一个ApproveBL领域对象，并要修改单据信息 |
| 后置条件 |  |
| 需要的服务（需接口） | | |
| ApproveBL.getUnApprove \*\*() | 获取某类待审批的单据信息 | |
| ApproveBL.setApproved\*\*(vo) | 通过目标单据的审批 | |
| ApproveBL.setAllApproved\*\*() | 通过某类目标单据的全部审批 | |
| ApproveBL.modify(ReceiptsVOvo) | 修改目标单据的信息 | |

StaffBL的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StaffBL.getUser | 语法 | public UserVO getUser(String userID) |
|  | 前置条件 | 进入人员查看修改界面 |
|  | 后置条件 | 获得所有的人员信息列表 |
| StaffBL.addUser | 语法 | public String addUser(String id,UserVO vo) |
|  | 前置条件 | 提交增加人员的请求，且输入符合规则 |
|  | 后置条件 | 在人员信息中增加一条人员信息 |
| StaffBL.updateUser | 语法 | public String updateUser(String id,UserVO vo) |
|  | 前置条件 | 提交修改人员的信息 |
|  | 后置条件 | 在人员信息的列表中修改该条人员信息 |
| StaffBL.delete | 语法 | public String delete (String id) |
|  | 前置条件 | 删除人员信息 |
|  | 后置条件 | 在人员信息的列表中删除该条人员信息 |
| StaffBL.getUsers | 语法 | public List<UserVO> getUsers() |
|  | 前置条件 | 无 |
|  | 后置条件 | 获取所有的人员User的信息列表 |
| StaffBL.getUsersOfIns | 语法 | public List<UserVO> getUsersOfIns(String InstitutionID) |
|  | 前置条件 | 在按照机构进行人员管理界面，选中某个机构进行人员管理 |
|  | 后置条件 | 获取该机构编号对应的机构的所有的人员User的信息列表 |
| 需要的服务（需接口） | | |
| UserDataService.getUsers() | 获取持久化数据中所有的User信息，返回一个UserPO的列表 | |
| UserDataService.add(UserPO po) | 向持久化数据中加入一条user数据，并更新txt文件 | |
| UserDataService.delete(UserPO po) | 从持久化数据中删除一条User数据并更新txt文件 | |

InstitutionBL的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| InstitutionBL.show | 语法 | public List<InstitutionVO> show () |
|  | 前置条件 | 显示所有的机构 |
|  | 后置条件 | 返回所有机构信息列表 |
| InstitutionBL.showByCity() | 语法 | public List<InstitutionVO> showByCity() |
|  | 前置条件 | 显示所有的机构（按城市） |
|  | 后置条件 | 按城市返回机构信息列表 |
| InstitutionBL.addInst | 语法 | public String addIns(InstitutionVO vo) |
|  | 前置条件 | 增加一个机构(营业厅)，输入符合规则 |
|  | 后置条件 | 在机构信息列表中增加一个机构(营业厅) |
| InstitutionBL.addInst | 语法 | public String addIns(InstitutionVO vo,InventoryVO ivo) |
|  | 前置条件 | 新增机构信息（中转中心），且输入符合规则 |
|  | 后置条件 | 在机构列表中增加一个机构InstitutionPO，而且在库存列表中增加一个InventoryPO |
| InstitutionBL.modify | 语法 | public String modify(String InstitutionID,String newName) |
|  | 前置条件 | 对机构(营业厅)进行修改（名称） |
|  | 后置条件 | 对机构列表中对应机构编号的机构进行名称修改 |
| InstitutionBL.modify | 语法 | public String modify(String InventoryID, InventoryVO vo) |
|  | 前置条件 | 更改机构信息(仓库信息)，且输入符合规则 |
|  | 后置条件 | 将库存列表中对应的机构编号的库存进行修改 |
| InstitutionBL.deleteIns | 语法 | public String deleteIns(InstitutionVO vo) |
|  | 前置条件 | 选定某机构，并删机构信息 |
|  | 后置条件 | 删除该机构并删除该机构的附属产物(人员、库存(if have)) |
| 需要的服务（需接口） | | |
| InstitutitonDataService.getInsList | 从持久化数据中获取所有的机构列表信息 | |
| InstitutitonDataService.add | 在持久化数据中增加一条机构信息 | |
| InstitutionDataService.modify | 在持久化数据中修改一条机构信息 | |
| InstitutitonDataService.delete | 在持久化数据中删除一条机构信息 | |
| InventoryDataService.add | 在持久化数据中增加一条库存信息 | |
| InventoryDataService.delete | 在持久化数据中删除一条库存信息 | |
| InventoryDataService.modify | 在持久化数据中修改一条库存信息 | |
| UserDataService.delete | 在持久化数据中删除一条机构信息 | |
| StrategyBL.getCities | 获取机构的城市列表 | |

(4) manageBL层的动态模型

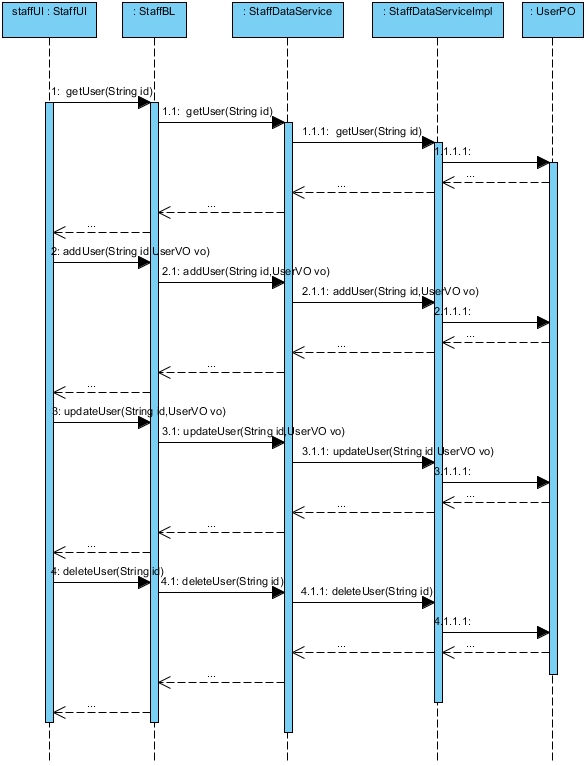


图 人员信息管理的顺序图

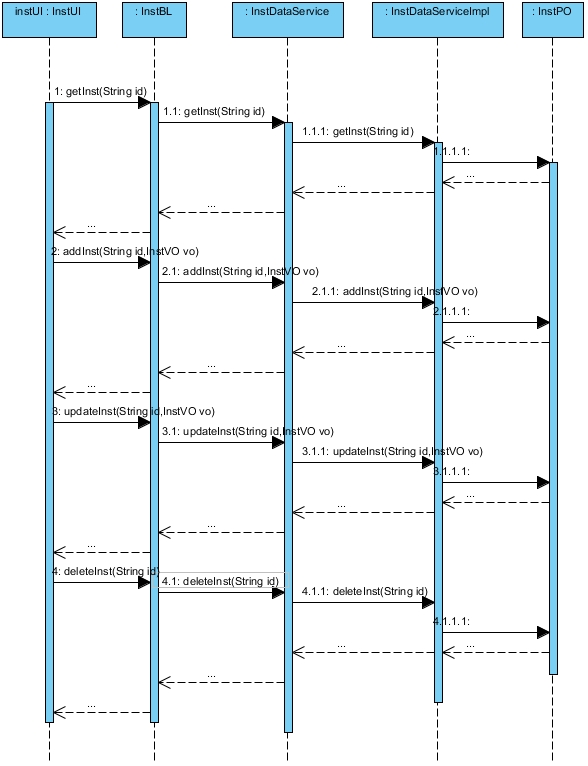


图 机构信息管理的顺序图

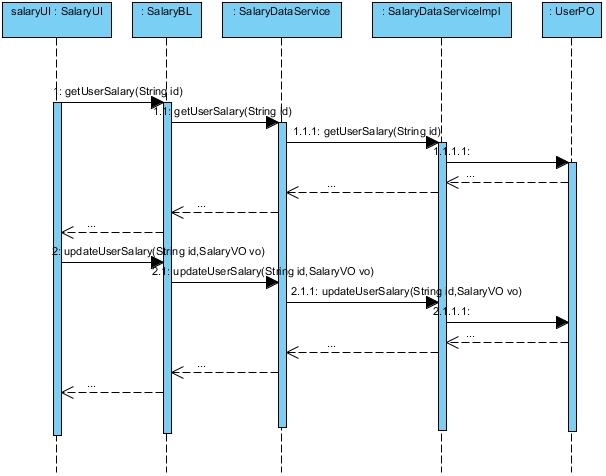


图 薪水策略制定的顺序图

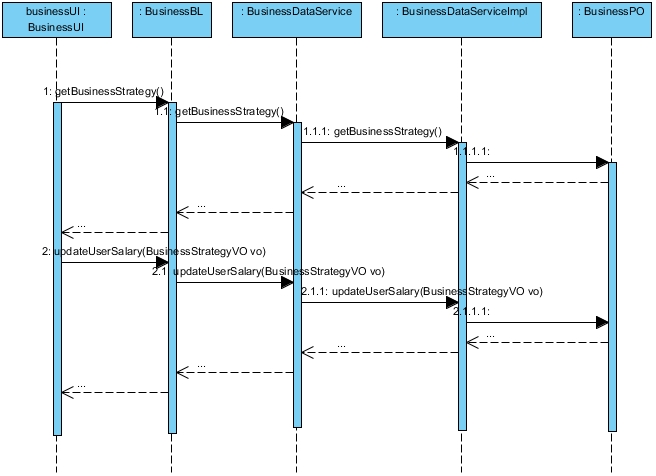


图 业务策略制定的顺序图

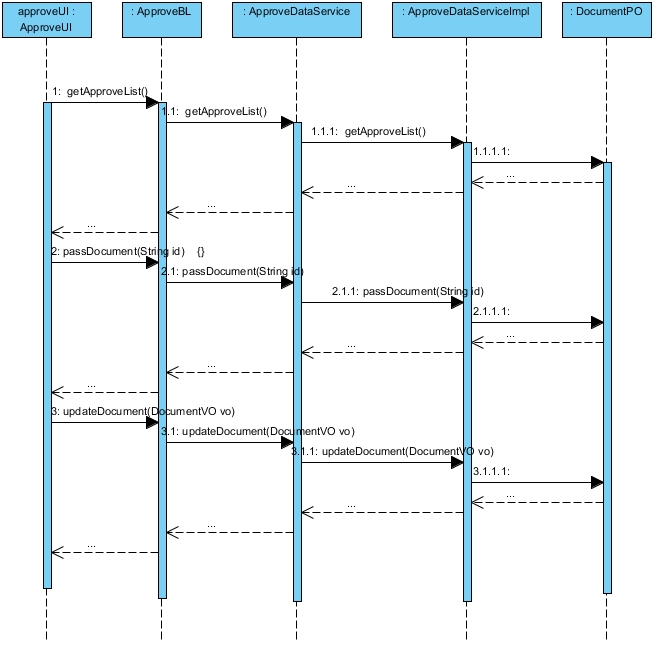


图 审批单据的顺序图

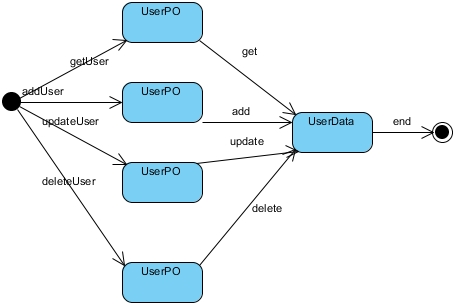


图 人员信息管理的状态图

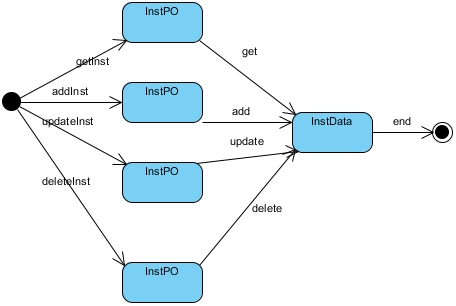


图 机构信息管理的状态图

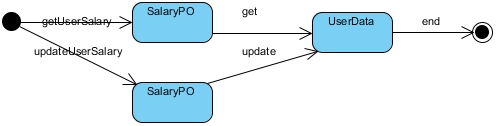


图 薪水策略制定的状态图

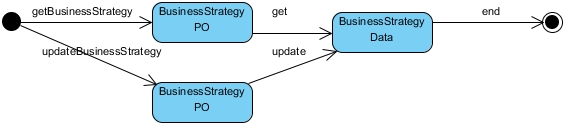


图 业务策略制定的状态图

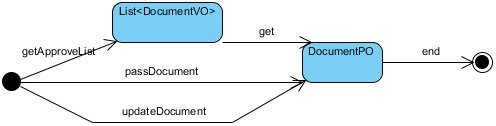


图 审批单据的状态图

* + 1. adminBL模块

(1) 模块描述

adminBL模块承担的需求参见需求规格说明文档的功能需求以及相关非功能需求

adminBL模块的职责及接口参见软件系统结构描述文档表

(2) 整体结构

根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了AdminBLService和AdminDataService两个接口。为了隔离业务逻辑职责和逻辑控制职责，我们添加了对应的Controller，这样各类Controller将会将管理的业务逻辑职责和逻辑控制委托给AdminBL对象。UserPO是作为管理信息需要和产生的持久化对象被添加到设计模型中的。

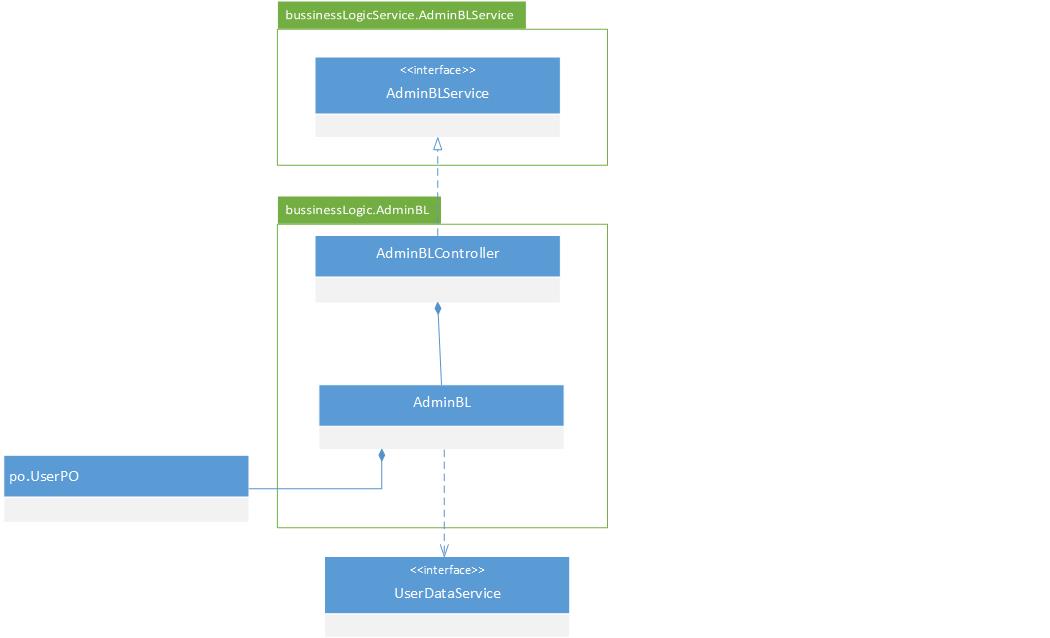


图 adminBL模块的各个类的设计

表 adminBL模块的各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| AdminBLController | 负责实现账户信息管理的所需要的服务 |
| AdminBL | 账户信息管理的领域模型对象，可以帮助完成账户信息管理界面所需要服务 |

(3) 模块内部类的接口规范

AdminController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| AdminController.search | 语法 | public UserVO search(String Key) |
|  | 前置条件 | 已经创建一个AdminBL领域对象，并要查找查看账户信息，且输入符合规则 |
|  | 后置条件 | 调用AdminBL领域对象的search方法 |
| AdminController.modify | 语法 | public String modfy(String identityID,String newPassord) |
|  | 前置条件 | 已经创建了一个AdminBL领域对象，并要修改账户信息，且输入符合规则 |
|  | 后置条件 | 调用AdminBL领域对象的modify方法 |
| AdminController.getUserListByInsID | 语法 | public List<UserVO> getUserListByInsID(String institutionID) |
|  | 前置条件 | 已经创建了一个AdminBL领域对象，并选中某个机构，要查看该机构的人员账户信息 |
|  | 后置条件 | 调用AdminBL领域对象的getUserListByInsID方法 |
| AdminController.getInsList | 语法 | public List<InstitutionVO> getInsList () |
|  | 前置条件 | 已经创建了一个AdminBL领域对象，进入人员账户管理界面 |
|  | 后置条件 | 调用AdminBL领域对象的getInsList方法 |
|  |  |  |
| 需要的服务（需接口） | | |
| LogBL.add(Date date,String userID ,String message) | 增加一条log日志信息 | |
| AdminBL.search(String key) | 按照关键字查找用户 | |
| AdminBL.modify(String id,String newPassword) | 修改某一个人员账户的密码 | |
| AdminBL.getUserListByInsID | 根据机构的编号返回该机构人员信息列表 | |
| AdminBL.getInsList | 返回当前所有的机构列表 | |

(4) adminBL层的动态模型

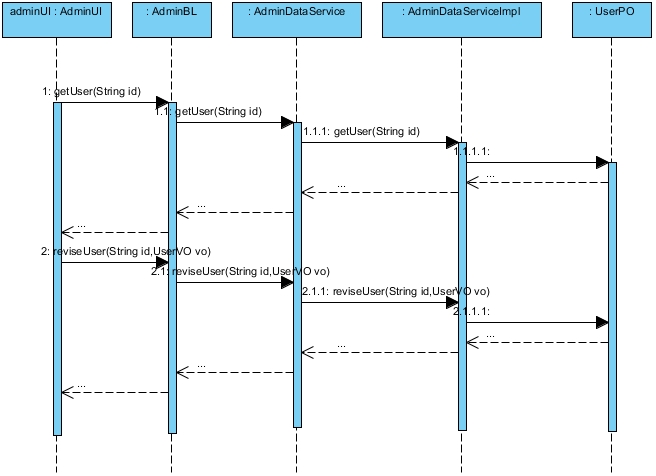


图 账户信息管理的顺序图

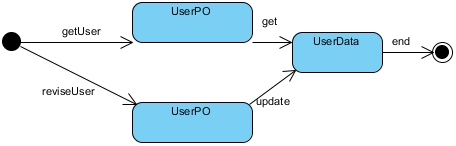


图 账户信息管理的状态图

* + 1. inventoryBL模块

(1) 模块描述

inventoryBL模块承担的需求参见需求规格说明文档的功能需求以及相关非功能需求

inventoryBL模块的职责及接口参见软件系统结构描述文档表

(2) 整体结构

根据体系结构的设计，采用分层风格，将系统分为展示层，业务逻辑层，数据层。每一层之间为了灵活性，添加了接口，以实现针对接口编程，隔离数据传输的职责，降低层与层之间耦合，添加了InventoryBLService和InventoryDataService两个接口。为了隔离业务逻辑职责和逻辑控制职责，我们添加了对应的Controller，这样各类Controller将会将管理的业务逻辑职责和逻辑控制委托给InventoryBL对象。InventoryPO,EntryPO,ExportPO是作为管理信息需要和产生的持久化对象被添加到设计模型中的。

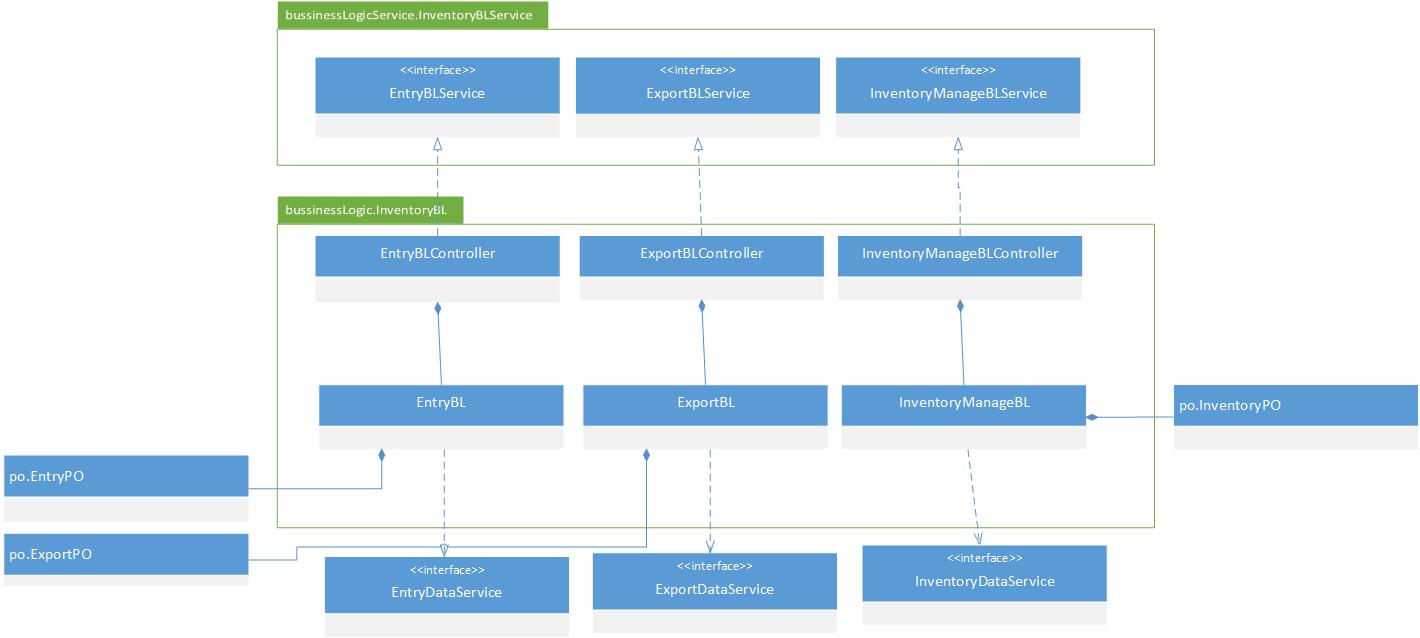


图 inventoryBL模块的各个类的设计

表 inventoryBL模块的各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| EntryController | 负责实现入库管理的所需要的服务 |
| ShipmentController | 负责实现出库管理的所需要的服务 |
| InventoryController | 负责实现库存管理的所需要的服务 |
| EntryBL | 入库管理的领域模型对象，可以帮助完成入库管理界面所需要服务 |
| ShipmentBL | 出库管理的领域模型对象，可以帮助完成出库管理界面所需要服务 |
| InventoryBL | 库存管理的领域模型对象，可以帮助完成库存管理界面所需要服务 |

(3) 模块内部类的接口规范

EntryController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| EntryController.add | 语法 | public String add(EntryVO vo) |
|  | 前置条件 | 已经创建一个EntryBL领域对象，并要提交入库单，且输入符合规则 |
|  | 后置条件 | 调用EntryBL领域对象的add方法 |
| EntryController.delete | 语法 | public String delete(EntryVO vo) |
|  | 前置条件 | 已经创建一个EntryBL领域对象，删除某个入库单 |
|  | 后置条件 | 调用EntryBL领域对象的add方法 |
| EntryController.getEntryList | 语法 | public List<EntryPO> getEntryList(String institutionID) |
|  | 前置条件 | 已经创建一个EntryBL领域对象，并要查看本机构的入库单信息 |
|  | 后置条件 | 调用EntryBL领域对象的getEntryList方法 |
| EntryController.getInstitutionList | 语法 | public List<InstitutionVO> getInstitutionList(String userID) |
|  | 前置条件 | 已经创建一个EntryBL领域对象，填写入库单时在目的机构中调用该方法生成一个到达机构选择下拉框 |
|  | 后置条件 | 调用EntryBL领域对象的getInstitutionList方法 |
| 需要的服务（需接口） | | |
| LogBL.add | 增加一条log日志信息 | |
| EntryBL.add | 在持久化数据中增加一个入库单 | |
| EntryBL.delete | 在持久化数据中删除一个入库单 | |
| EntryBL.getEntryList | 获取持久化数据的入库单list | |
| EntryBL.getInstitutionList | 获取持久化数据的所有机构名称 | |

ShipmentController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ShipmentController.add | 语法 | public String add(ShipmentVO vo) |
|  | 前置条件 | 已经创建一个ShipmentBL领域对象，并要提交出库单，且输入符合规则 |
|  | 后置条件 | 调用ShipmentBL领域对象的add方法 |
| ShipmentController.delete | 语法 | public String delete(ShipmentVO vo) |
|  | 前置条件 | 已经创建一个ShipmentBL领域对象，并要删除出库单 |
|  | 后置条件 | 调用ShipmentBL领域对象的delete方法 |
| ShipmentController.find | 语法 | public ShipmentVO find(String id) |
|  | 前置条件 | 已经创建一个ShipmentBL领域对象，并要按编号查找出库单 |
|  | 后置条件 | 调用ShipmentBL领域对象的find方法 |
| ShipmentController.getInstitutionList | 语法 | public List<InstitutionVO> getInstitutionList(String userID) |
|  | 前置条件 | 已经创建一个ShipmentBL领域对象，并要获得所有机构列表 |
|  | 后置条件 | 调用ShipmentBL领域对象的getInstitutionList方法 |
| ShipmentController.getUnapproved() | 语法 | public List<ShipmentVO> getUnapproved() |
|  | 前置条件 | 已经创建一个ShipmentBL领域对象，并要审批出库单 |
|  | 后置条件 | 调用ShipmentBL领域对象的getUnapproved方法 |
| 需要的服务（需接口） | | |
| LogBL.add | 增加一条log日志信息 | |
| ShipmentBL.add | 在持久化数据中增加一个出库单 | |
| ShipmentBL.delete | 在持久化数据中删除一个出库单 | |
| ShipmentBL.find | 根据编号获取持久化数据的入库单 | |
| ShipmentBL.getInstitutionList | 获取持久化数据的所有机构信息 | |
| ShipmentBL.getunapproved | 获取持久化数据所有未审批的入库单 | |

InventoryController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| InventoryBLController.getInventoryVO | 语法 | public InventoryVO getInventoryVO(String institutionID) |
| 前置条件 | 已经创建一个InventoryBL领域对象，并要查找某机构的库存信息，且输入符合规则 |
| 后置条件 | 调用InventoryBL领域对象的getInventoryVO方法 |
| InventoryBLController.adjust | 语法 | public String adjust(EntryVO vo) |
| 前置条件 | 已经创建了一个InventoryBL领域对象，并要进行库存分区调整，且输入符合规则 |
| 后置条件 | 调用InventoryBL领域对象的adjust方法 |
| InventoryBLController.find | 语法 | public InventoryVO find(String institutionID) |
| 前置条件 | 已经创建了一个InventoryBL领域对象，并要进行库存查看，且输入符合规则 |
| 后置条件 | 调用InventoryBL领域对象的find方法 |
| InventoryBLController.getEntryList | 语法 | public List<EntryVO> getEntryList(String institutionID, InventoryArea area) |
| 前置条件 | 已经创建了一个InventoryBL领域对象，并要查找入库单信息 |
| 后置条件 | 调用InventoryBL领域对象的getEntryList方法 |
| 需要的服务（需接口） | | |
| LogBL.add | 增加一条log日志信息 | |
| InventoryBL.getInventoryVO | 根据机构编号返回库存信息 | |
| InventoryBL.adjust | 调整库存分区 | |
| InventoryBL.find | 获得某机构的库存信息 | |
| InventoryBL.getEntryList | 获得入库单信息 | |

(4) inventoryBL层的动态模型

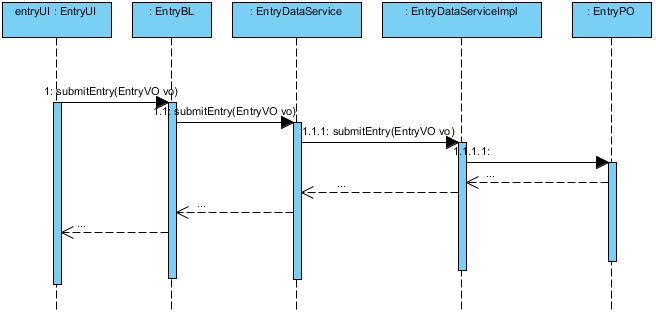


图 入库管理的顺序图

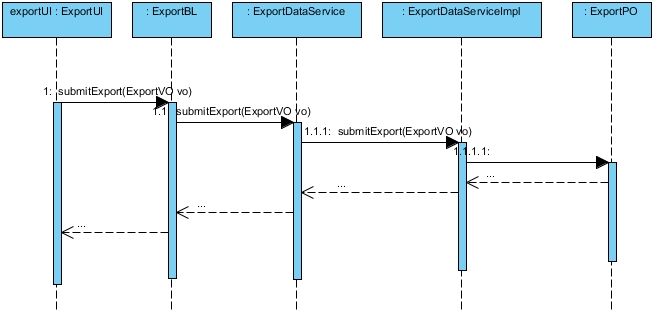


图 出库管理的顺序图

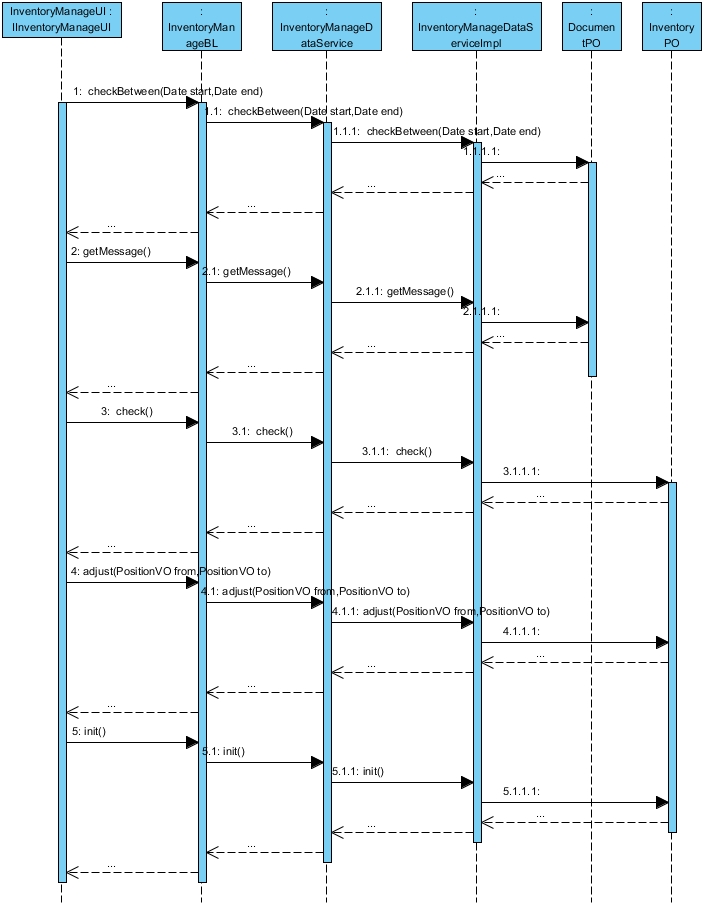


图 库存管理的顺序图

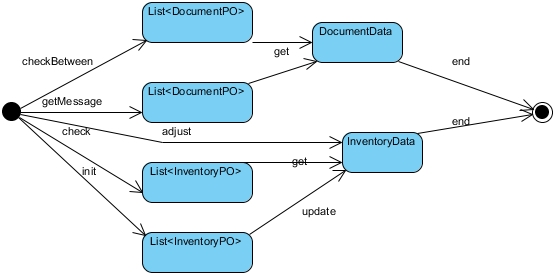
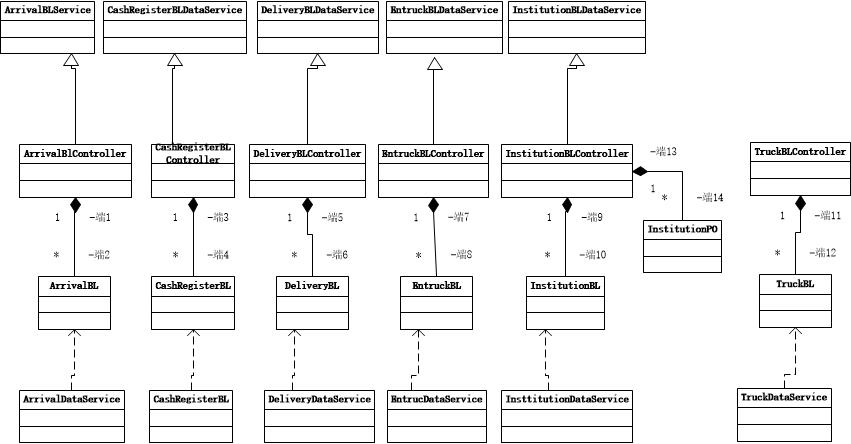


图 库存管理的状态图

**4.1.7businessHallBL模块描述**

businessHallBL模块承担的需求参见需求规格说明文档的功能需求及相关的非功能需求

businessHallBL模块的职责及接口参见软件系统结构描述文档表



businessHallBL 模块各个类的职责如表

|  |  |
| --- | --- |
| 模块 | 职责 |
| ArrivalBLController | 负责实现到达单操作所需要的服务 |
| CashRegisterBLController | 负责实现收款单操作所需要的服务 |
| DeliveryBLController | 负责实现派件单操作所需要的服务 |
| EntruckBLController | 负责实现装车单操作所需要的服务 |
| TruckBLController | 负责实现车辆信息管理所需要的服务 |
| ArrivalBL | 到达单操作的领域模型对象，拥有一次到达单操作所需的信息，可以完成帮助到达单操作界面所需要的任务。 |
| CashRegisterBL | 收款单操作的领域模型对象，拥有一次收款单操作所需的信息，可以完成帮助收款单操作界面所需要的任务。 |
| DeliveryBL | 派件单操作的领域模型对象，拥有一次派件单操作所需的信息，可以完成帮助派件单操作界面所需要的任务。 |
| EntruckBL | 到装车操作的领域模型对象，拥有一次到达单操作所需的信息，可以完成帮助装车单操作界面所需要的任务。 |
| TruckBL | 车辆信息管理的领域模型对象，拥有一次车辆信息管理所需的信息，可以完成帮助到车辆信息界面所需要的任务。 |

模块内部类的接口规范

ArrivalController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ArrivalBLController.add | 语法 | public String add(ArrivalVO arrivalVO) |
| 前置条件 | 已经创建一个ArrivalBL领域对象，并要新建到达单 |
| 后置条件 | 调用ArrivalBL里的add方法 |
| ArrivalBLController.search | 语法 | public ArrivalVO search(String transferID) |
| 前置条件 | 已经创建一个ArrivalBL领域对象，并要查找到达单 |
| 后置条件 | 调用ArrivalBL里的search方法 |
| ArrivalBLController.delete | 语法 | public String delete(ArrivalVO arrivalVO) |
| 前置条件 | 已经创建一个ArrivalBL领域对象，并要删除到达单 |
| 后置条件 | 调用ArrivalBL里的delete方法 |
| ArrivalBLController.modify | 语法 | public String modify(ArrivalVO arrivalVO) |
| 前置条件 | 已经创建一个ArrivalBL领域对象，并要新建到达单 |
| 后置条件 | 调用ArrivalBL里的modify方法 |
| ArrivalBLController.getIDList | 语法 | public List<String> getIDList(String entruckID) |
| 前置条件 | 已经创建一个ArrivalBL领域对象，并要获得所有快递单号的列表 |
| 后置条件 | 调用ArrivalBL里的getIDList方法 |
| ArrivalBLController.getInstitutionName | 语法 | public String getInstitutionName() |
| 前置条件 | 已经创建一个ArrivalBL领域对象，并要获得所有机构名称 |
| 后置条件 | 调用ArrivalBL里的getInstitutionName方法 |
| 需要的服务（需接口） | | |
| ArrivalBL.add | 新建一个到达单 | |
| ArrivalBL.search | 查找一个到达单 | |
| ArrivalBL.delete | 删除一个到达单 | |
| ArrivalBL.modify | 修改一个到达单 | |
| ArrivalBL.getIDList | 根据装车单信息返回对应所有快递单号的列表 | |
| ArrivalBL.getInstitutionName | 返回对应所有机构名称 | |
| LogBL.add | 加入一条新的操作日志 | |

CashRegisterController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CashRegisterController.add | 语法 | public String add(IncomeVO incomeVO) |
| 前置条件 | 已经创建一个CashRegisterBL领域对象，并要新建收款单 |
| 后置条件 | 调用CashRegisterBL里的add方法 |
| CashRegisterController.delete | 语法 | public String delete(CashRegisterVO crVO) |
| 前置条件 | 已经创建一个CashRegisterBL领域对象，并要删除收款单 |
| 后置条件 | 调用CashRegisterBL里的delete方法 |
| CashRegisterController.search | 语法 | public String search(String id) |
| 前置条件 | 已经创建一个CashRegisterBL领域对象，并要查找收款单 |
| 后置条件 | 调用CashRegisterBL里的cancel方法 |
| CashRegisterController.getSendByStaffID | 语法 | public List<SendVO> getSendByStaffID(Date d, String staffID) |
| 前置条件 | 已经创建一个CashRegisterBL领域对象，并要获得寄件单信息 |
| 后置条件 | 调用CashRegisterBL里的getSendByStaffID方法 |
| CashRegisterController.getAmmounts | 语法 | public double getAmmounts(List<SendVO> sendList) |
| 前置条件 | 已经创建一个CashRegisterBL领域对象，并要计算寄件单运费总和 |
| 后置条件 | 调用CashRegisterBL里的getAmmounts方法 |
| CashRegisterController.getIncomeBetweemDate | 语法 | public List<IncomeVO> getIncomeBetweemDate(Date start, Date end) |
| 前置条件 | 已经创建一个CashRegisterBL领域对象，并要获得制定时间段内的收款单信息 |
| 后置条件 | 调用CashRegisterBL里的getIncomeBetweemDate方法 |
| CashRegisterController.getIncomeAmmount | 语法 | public double getIncomeAmmount(List<IncomeVO> incomeList) |
| 前置条件 | 已经创建一个CashRegisterBL领域对象，并要计算收款单收款总和 |
| 后置条件 | 调用CashRegisterBL里的getIncomeAmmount方法 |
| CashRegisterController.getIncomeByIns | 语法 | public List<IncomeVO> getIncomeByIns(String institutionID) |
| 前置条件 | 已经创建一个CashRegisterBL领域对象，并要获得指定机构的所有收款单信息 |
| 后置条件 | 调用CashRegisterBL里的getIncomeByIns方法 |
| CashRegisterController.getIncomeList | 语法 | public List<IncomeVO> getIncomeList() |
| 前置条件 | 已经创建一个CashRegisterBL领域对象，并要获得所有收款单信息 |
| 后置条件 | 调用CashRegisterBL里的getIncomeList方法 |
| 需要的服务（需接口） | | |
| CashRegisterBL.add | 新增一个收款单 | |
| CashRegisterBL.delete | 删除一个收款单 | |
| CashRegisterBL.search | 查找一个收款单 | |
| CashRegisterBL.getSendByStaffID | 查找指定员工提交的所有寄件单 | |
| CashRegisterBL.getAmmounts | 计算寄件单运费总和 | |
| CashRegisterBL.getIncomeBetweemDate | 查找某时间段内提交的收款单 | |
| CashRegisterBL.getIncomeAmmount | 计算收款单收款总额 | |
| CashRegisterBL.getIncomeByIns | 查找指定机构的所有收款单 | |
| CashRegisterBL.getIncomeList | 获取所有收款单 | |
| LogBL.add | 加入一条新的操作日志 | |

DeliveryController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| DeliveryController.add | 语法 | public String add(DeliveryVO deliveryVO) |
| 前置条件 | 已经创建一个DeliveryBL领域对象，需要新建派件单 |
| 后置条件 | 调用DeliveryBL里的add方法 |
| DeliveryController.search | 语法 | public DeliveryVO search(String id) |
| 前置条件 | 已经创建一个DeliveryBL领域对象，需要查找派件单 |
| 后置条件 | 调用DeliveryBL里的search方法 |
| DeliveryController.delete | 语法 | public String delete(DeliveryVO vo) |
| 前置条件 | 已经创建一个DeliveryBL领域对象，需要删除派件单 |
| 后置条件 | 调用DeliveryBL里的delete方法 |
| DeliveryController.getSendMessage | 语法 | public SendVO getSendMessage(String ID) |
| 前置条件 | 已经创建一个DeliveryBL领域对象，需要获得派件单信息 |
| 后置条件 | 调用DeliveryBL里的getSendMessage方法 |
| 需要的服务（需接口） | | |
| DeliveryBL.add | 新增一个派件单 | |
| DeliveryBL.search | 查找一个派件单 | |
| DeliveryBL.delete | 删除一个派件单 | |
| DeliveryBL.getSendMessage | 获得派件单信息 | |
| LogBL.add | 加入一条新的操作日志 | |

EntruckController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| EntruckController.add | 语法 | Publc void add(String ID) |
| 前置条件 | 已经创建一个EntruckBL领域对象需要新建装车单 |
| 后置条件 | 调用EntruckBL里的add方法 |
| EntruckController.cancel | 语法 | Public void cancel(Boolean ancel) |
| 前置条件 | 已经创建一个EntruckBL领域对象并新建装车单，需要取消 |
| 后置条件 | 调用EntruckBL里的cancel方法 |
| EntruckController.show | 语法 | Public EntruckVO show(String ID) |
| 前置条件 | 营业厅业务员需要查看一个装车单 |
| 后置条件 | 调用EntruckBL里的show方法 |
| EntruckController.getTrans | 语法 | Public String getTrans() |
| 前置条件 | 已经创建一个EntruckBL领域对象并新建装车单并需中转单信息 |
| 后置条件 | 调用EntruckBL里的getTrans方法 |
| 需要的服务（需接口） | | |
| EntruckBL.add | 新增装车单 | |
| EntruckBL.cancel | 取消装车单操作 | |
| EntruckBL.show | 查看装车单 | |
| EntruckBL.getTrans | 得到中转单信息 | |

TruckBLController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| TruckBLController.addTruck | 语法 | Public void addTruck(String ID) |
| 前置条件 | 创建TruckBL领域对象，并新增车辆信息 |
| 后置条件 | 调用TruckBL的addTruck方法 |
| TruckBLController.deleteTruck | 语法 | Public void addTruck(String ID) |
| 前置条件 | 创建TruckBL领域对象，并删除车辆信息 |
| 后置条件 | 调用TruckBL的deleteTruck方法 |
| TruckBLController.show | 语法 | Public list<TruckVO> show(String InsID) |
| 前置条件 | 创建TruckBL领域对象，并查看相应机构的车辆信息 |
| 后置条件 | 调用TruckBL的show方法 |
| 需要的服务（需接口） | | |
| TruckBL.addTruck | 新增车辆信息 | |
| TruckBL.deleteTruck | 删除车辆信息 | |
| TruckBL.show | 查看车辆信息 | |

ArrivalBL的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ArrivalBL.add | 语法 | Public void add(String ID) |
| ArrivalBL.cancel | 语法 | Public void cancel(Boolean ancel) |
| ArrivalBL.show | 语法 | Public list<ArrivalVO> Show(String ID) |
| ArrivalBL.updateLogis | 语法 | Public void updateLogis(String ID,String time,String place) |
| 需要的服务（需接口） | | |
| ArrivalDataService.find | 从持久化数据中查找到达单信息 | |
| ArrivalDataService.add | 从持久化数据中新增到达单信息 | |
| LogisticsDataService.update | 从持久化数据中更新物流信息 | |

CashRegisterBL的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| CashRegisterBL.add | 语法 | | Public void add(String ID ) |
| CashRegisterBL.cancel | 语法 | | Public void cancel(Boolean ancel) |
| CashRegisterBL.show | 语法 | | Public list<CashRegisterVO> show(String ID) |
| 需要的服务（需接口） | | | |
| CashRegisterDataService.find | | 从持久化数据中查找收款单 | |
| CashRegisterDataService.add | | 从持久化数据中新增收款单 | |

DeliveryBL的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| DeliveryBL.add | 语法 | Public void add(String ID) |
| DeliveryBL.cancel | 语法 | Public void cancel(Boolean ancel) |
| DeliveryBL.show | 语法 | Public list<DeliveryVO> show(String ID) |
| DeliveryBL.updateLogis | 语法 | Public void update(String ) |
| 需要的服务（需接口） | | |
| DeliveryDataService.add | 从持久化数据中新增派件单 | |
| LogisticsDataService.update | 从持久化数据中更新物流信息 | |

EntruckBL的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| EntruckBL.add | 语法 | Public vod add(String ID) |
| EntruckBL.cancel | 语法 | Public void cancel(Boolean ancel) |
| Entruck.show | 语法 | Public list<EntruckVO> show(String ID) |
| 需要的服务（需接口） | | |
| EntruckDataService.add | 从持久化数据中新增装车单 | |
| EntruckDataService.find | 从持久化数据中查找装车单 | |
| TransferDataService.find | 从持久化数据中查找中转单 | |

InstitutionBL的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| InstitutionBL.addInstitution | 语法 | | Public void addInstitution(String ID) |
| institutionBL.deleteInsitution | 语法 | | Public void deleteInstitution(String ID) |
| InstitutionBL.addStaff | 语法 | | Public void addStaff(String ID,String name) |
| InstitutionBL.deleteStaff | 语法 | | Public void deleteStaff(String ID,String name) |
| InstitutionBL.show | 语法 | | Public InstitutionVO show(String ID) |
| 需要的服务（需接口） | | | |
| InstitutionDataService.addIns | | 从持久化数据中新增机构信息 | |
| InstitutionDataService.deleteIns | | 从持久化数据中删除机构信息 | |
| InstitutionDataService.addStaff | | 从持久化数据中新增人员信息 | |
| InstitutionDataService.deleteStaff | | 从持久化数据中输出人员信息 | |
| InstitutionDataService.updateSaff | | 从持久化数据中更新人员信息 | |
| InstitutionDataService.find | | 从持久化数据中查找机构信息 | |

TruckBL的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| TruckBL.addTruck | 语法 | Public void addTruck(String ID) |
| TruckBL.deleteTruck | 语法 | Public void deleteTruck(String ID ) |
| TruckBL.show | 语法 | Public list<TruckVO> show(String ID) |
| 需要的服务（需接口） | | |
| TruckDataService.add | 从持久化数据中新增车辆信息 | |
| TruckDataService.update | 从持久化数据中更新车辆信息 | |
| TruckDataService.delete | 从持久化数据中输出车辆信息 | |
| TruckDataService.find | 从持久化数据中查找车辆信息 | |

动态模型

（待补充

**4.1.8deliveryBL模块描述**

deliveryBL模块承担的需求参见需求规格说明文档的功能需求及相关的非功能需求

deliveryBL模块的职责及接口参见软件系统结构描述文档表



delivery模块各个类的职责如表

|  |  |
| --- | --- |
| 模块 | 职责 |
| ReceiveBLController | 负责实现对应收件单操作所需要的服务 |
| SendBLController | 负责实现对应寄件单操作所需要的服务 |
| ReceiveBL | 收件单操作的领域模型对象，可以进行完成帮助收件单操作的任务 |
| SendBL | 寄件单操作的领域模型对象，拥有一次寄件单操作所需要的信息，可以进行完成帮助寄件单操作的任务。 |

模块内部类的接口规范

ReceiveBLController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ReceiveBLController.add | 语法 | Public void add(String ID) |
| 前置条件 | 已经创建一个ReceiveBL领域对象并新建收件单 |
| 后置条件 | 调用ReceiveBL的add方法 |
| ReceiveBLController.cancel | 语法 | Public void cancel(Boolean ancel) |
| 前置条件 | 已经创建一个ReceiveBL领域对象并新建收件单，准备取消收件单 |
| 后置条件 | 调用ReceiveBL的cancel方法 |
| ReceiveBLController.show | 语法 | Public ReceiveVO show(String ID) |
| 前置条件 | 已经创建一个ReceiveBL领域对象准备查看收件单 |
| 后置条件 | 调用ReceiveBL的show方法 |
| 需要的服务（需接口） | | |
| ReceiveBL.add | 新增收件单 | |
| ReceiveBL.cancel | 取消收件单操作 | |
| ReceiveBL.show | 查看收件单 | |
| ReceiveBL.updateLogis | 更新物流信息 | |

SendBLController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SendBLController.add | 语法 | Public void add(String ID) |
| 前置条件 | 已经创建一个ReceiveBL领域对象并新建寄件单 |
| 后置条件 | 调用SendBL的add方法 |
| SendBLController.cancel | 语法 | Public void cancel(Boolean ancel) |
| 前置条件 | 已经创建一个sendBL领域对象并新建寄件单,取消寄件单操作 |
| 后置条件 | 调用SendBL的cancel方法 |
| SendBLController.show | 语法 | Public sendVO show(String ID) |
| 前置条件 | 已经创建一个SendBL领域对象并查看寄件单 |
| 后置条件 | 调用SendBL的show方法 |
| 需要的服务（需接口） | | |
| SendBL.add | 新增寄件单 | |
| SendBL.cancel | 取消寄件单操作 | |
| SendBL.Show | 查看寄件单 | |
| SendBL.updateLogis | 更新物流信息 | |

ReceiveBL的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ReceiveBL.add | 语法 | Public void add(String ID) |
| ReceiveBL.cancel | 语法 | Public void cancel(Boolean ancel) |
| ReceiveBL.show | 语法 | Public receiveVO show(String ID) |
| ReceiveBL.updateLogis | 语法 | Public void updateLogis(String ID,String time,String place) |
| 需要的服务（需接口） | | |
| ReceiveDataService.add | 从持久化数据中新增收件单 | |
| ReceiveDataService.find | 从持久化数据中查找收件单 | |
| LogisticsDataService.update | 从持久化数据中更新 | |

SendBL的接口规范

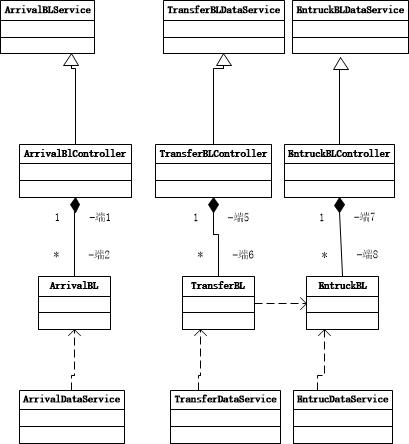
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SendBL.add | 语法 | Public void add(String ID) |
| SendBL.cancel | 语法 | Public void cancel(Boolean ancel) |
| SendBL.Show | 语法 | Public SendVO show(String ID) |
| SendBL.updateLogis | 语法 | Public void updateLogis(String ID,String time,String place) |
| 需要的服务（需接口） | | |
| SendDataService.add | 从持久化数据中新增寄件单信息 | |
| SendDataService.find | 从持久化数据中查找寄件单信息 | |
| LogisticsDataService.add | 从持久化数据中新建物流信息 | |

动态模型

**4.1.9IntermediateBL模块描述**

intermediateBL模块承担的需求参见需求规格说明文档的功能需求及相关的非功能需求

intermediateBL模块的职责及接口参见软件系统结构描述文档表



intermediateBL模块各个类的职责如表

|  |  |
| --- | --- |
| 模块 | 职责 |
| TransferBLController | 负责实现中转单操作所需要的服务 |
| TransferBL | 中转单操作的领域模型对象，拥有一次中转单操作所需的信息，可以帮助完成中转单操作所需的服务 |

模块内部类的接口规范

TransferBLController的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| TransferBLController.add | 语法 | Public void add(String ID) |
| 前置条件 | 已经创建一个TransferBL领域对象，并要新建中转单 |
| 后置条件 | 调用TransferBL里的add方法 |
| TransferBLController.cancel | 语法 | Public void cancel(Boolean ancel) |
| 前置条件 | 已经创建一个TransferBL领域对象并已新建中转单，要取消中转单操作 |
| 后置条件 | 调用TransferBL里的cancel方法 |
| TransferBLController.show | 语法 | Public TransferVO showDco(String ID) |
| 前置条件 | 查看中转单操作 |
| 后置条件 | 调用TransferBL里的show方法 |
| 需要的服务（需接口） | | |
| TransferBL.add | 新增中转单 | |
| TransferBL.cancel | 取消中转单操作 | |
| TransferBL.show | 查看中转单 | |
| TransferBL.updateLoigs | 更新物流信息 | |

动态模型