|  |
| --- |
| 南京大学软件学院 |
| 物流管理系统 |
| 软件详细设计描述文档 |
|  |
| **版本：V2.0** |
| **2015/10/29** |

|  |
| --- |
|  |
|  |

**小组成员： 汤大业**

**谭期友**

**杨三洋**

**王家玮**

**目录及索引**

[1.引言 （2）](#引言)

[1.1编制目的](#编制目的)

[1.2词汇表](#词汇表)

[1.3参考资料](#参考资料)

[2.产品概述 （2）](#产品概述)

[3.体系结构设计概述 （2）](#体系结构描述)

[4.结构视角 （2）](#结构视角)

[5.依赖视角 （76）](#依赖视角)

1. **引言**

**1.1编制目的**

本报告详细完成对物流信息管理系统的详细设计，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、测试人员及最终的用户而编写，是了解系统的导航。

**1.2词汇表**

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| Bl | 业务逻辑(business logic) | 无 |

**1.3参考资料**

[Bin2012]骆斌，丁二玉，刘钦.软件工程与计算（卷二）——软件开发的技术基础.北京：机械工程出版社，2012.12;

物流信息管理系统用例文档

物流信息管理系统需求规格说明

物流信息管理系统体系结构设计文档

1. **产品概述**

参考物流信息管理系统用例文档和物流信息管理系统软件需求规格说明文档中对产品的概括描述。

1. **体系结构设计概述**

参考物流信息管理系统概要设计文档中对体系结构设计的概述

**4.结构视角**

**4.1客户端展示层模块的静态结构和动态职责**

客户端展示层由8个模块组成，分别是mainui、adminui、businessofficeclerkui、centerclerkui、centerrepertoryclerkui、courierui、financialstaffui、topmanagerui。mainui模块负责界面的跳转；adminui模块包含用户管理的界面；businessofficeclerkui模块包含车辆装车管理、接收与派件、核对收款单、车辆信息管理、司机信息管理的界面；centerclerkui模块包含中转接收、装运管理、入库管理的界面；centerrepertoryclerkui模块包含仓库管理的界面；courierui模块包含物流信息查询、寄件、收件的界面；financialstaffhighui模块包含结算管理、成本管理、账户管理、统计报表、期初建账、查看系统日志的界面；financialstafflowui模块包含结算管理、成本管理、统计报表、期初建账、查看系统日志的界面;topmanagerui模块包含统计报表、人员和机构管理、制定薪水策略、审判单据、制定距离价格常量、查看系统日志的界面。

**4.1.1客户端展示层模块局部模块的职责**

客户端展示层模块的职责如下表所示。

**表3 客户端展示层模块的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| mainui | 负责界面的跳转 |
| adminui | 包含用户管理的界面 |
| businessofficeclerkui | 包含车辆装车管理、接收与派件、核对收款单、车辆信息管理、司机信息管理的界面 |
| centerclerkui | 包含中转接收、装运管理、入库管理的界面 |
| centerrepertoryclerkui | 包含仓库管理的界面 |
| courierui | 包含物流信息查询、寄件、收件的界面 |
| Financialstaffhighui | 包含结算管理、成本管理、账户管理、统计报表、期初建账、查看系统日志的界面 |
| topmanagerui | 包含统计报表、人员和机构管理、制定薪水策略、审判单据、制定距离价格常量、查看系统日志的界面 |
| Financialstafflowui | 包含结算管理、成本管理、统计报表、期初建账、查看系统日志界面 |

**4.1.2客户端展示层模块局部模块的接口规范**

**4.1.2.1 mainui模块的接口规范**

mainui模块的接口规范如下表所示。

**表4 mainui模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的接口（供接口）** | | |
| mainui.init | 语法 | public void init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 根据逻辑进行界面跳转。 |

**4.1.2.2** adminui**模块的接口规范**

adminui模块的接口规范如下表所示。

**表5** adminui**模块的接口规范**

|  |  |
| --- | --- |
| **需要的接口（需接口）** | |
| **服务名** | **服务** |
| businesslogicservice.userblservice.UserManageBLService | 用户管理的业务逻辑接口 |

**4.1.2.3** businessofficeclerkui**模块的接口规范**

businessofficeclerkui模块的接口规范如下表所示。

**表6** businessofficeclerkui**模块的接口规范**

|  |  |
| --- | --- |
| **需要的接口（需接口）** | |
| **服务名** | **服务** |
| businesslogicservice.logisticsblservice. LoadBLService | 车辆装车管理的业务逻辑接口 |
| businesslogicservice.logisticsblservice. DeliverAndReceiveBLService | 接收与派件的业务逻辑接口 |
| businesslogicservice.logisticsblservice. | 核对收款单的业务逻辑接口 |
| businesslogicservice.logisticsblservice. TruckManageBLService | 车辆信息管理的业务逻辑接口 |
| businesslogicservice.logisticsblservice. DriverManageBLService | 司机信息管理的业务逻辑接口 |

**4.1.2.4** centerclerkui**模块的接口规范**

centerclerkui模块的接口规范如下表所示。

**表7** centerclerkui**模块的接口规范**

|  |  |
| --- | --- |
| **需要的接口（需接口）** | |
| **服务名** | **服务** |
| businesslogicservice.logisticsblservice.TransferCenterReceiveBLService | 中转接收的业务逻辑接口 |
| businesslogicservice.logisticsblservice.TransShipmentBLService | 装运管理的业务逻辑接口 |
| businesslogicservice.logisticsblservice.RepertoryManageBLService | 入库管理的业务逻辑接口 |

**4.1.2.5** centerrepertoryclerkui**模块的接口规范**

centerrepertoryclerkui模块的接口规范如下表所示。

**表8** centerrepertoryclerkui**模块的接口规范**

|  |  |
| --- | --- |
| **需要的接口（需接口）** | |
| **服务名** | **服务** |
| businesslogicservice.logisticsblservice.RepertoryManageBLService | 仓库管理的业务逻辑接口 |

**4.1.2.6** courierui**模块的接口规范**

courierui模块的接口规范如下表所示。

**表9** courierui**模块的接口规范**

|  |  |
| --- | --- |
| **需要的接口（需接口）** | |
| **服务名** | **服务** |
| businesslogicservice.logisticsblservice.SearchPkgInformationBLService | 物流信息查询的业务逻辑接口 |
| businesslogicservice.logisticsblservice.SendPkgBLService | 寄件的业务逻辑接口 |
| businesslogicservice.logisticsblservice.ReceivePkgBLService | 收件的业务逻辑接口 |

**4.1.2.7** financialstaffui**模块的接口规范**

financialstaffui模块的接口规范如下表所示。

**表10** financialstaffui**模块的接口规范**

|  |  |
| --- | --- |
| **需要的接口（需接口）** | |
| **服务名** | **服务** |
| businesslogicservice.financeblservice.SettlementManageBLService | 结算管理的业务逻辑接口 |
| businesslogicservice.financeblservice.CostManagementBLService | 成本管理的业务逻辑接口 |
| businesslogicservice.financeblservice.AccountBLService | 账户管理的业务逻辑接口 |
| businesslogicservice.chartblservice.FormBLService | 统计报表的业务逻辑接口 |
| businesslogicservice.financeblservice.BaseDataSettingBLService | 期初建账的业务逻辑接口 |
| businesslogicservice.chartblservice.SystemLogBLService | 查看系统日志的业务逻辑接口 |

**4.1.2.8** topmanagerui**模块的接口规范**

topmanagerui模块的接口规范如下表所示。

**表11** topmanagerui**模块的接口规范**

|  |  |
| --- | --- |
| **需要的接口（需接口）** | |
| **服务名** | **服务** |
| businesslogicservice.chartblservice.FormBLService | 统计报表的业务逻辑接口 |
| businesslogicservice.informationchangeblservice.PeopleAgencyBLService | 人员和机构管理的业务逻辑接口 |
| businesslogicservice.informationchangeblservice.PeopleAgencyBLService | 制定薪水策略的业务逻辑接口 |
| businesslogicservice.chartblservice.CheckBLService | 审判单据的业务逻辑接口 |
| businesslogicservice.informationchangeblservice.ConstantManageBLService | 制定距离价格常量的业务逻辑接口 |
| businesslogicservice.chartblservice.SystemLogBLService | 查看系统日志的业务逻辑接口 |

**4.1.3 客户端展示层模块局部模块的设计原理**

客户端展示层模块使用了集中式的控制风格。每个界面都通过mainui模块实现跳转。

客户端展示层模块使用了集中式的控制风格。每一个界面的职责从mainui之中分离并通过mainui模块实现跳转。

**4.2客户端业务逻辑层模块的静态结构和动态职责**

**4.2.1客户端业务逻辑层模块局部模块的职责**

客户端业务逻辑层由20个模块组成，分别是CheckBL，FormBL，SystemLogBL，AccountBL，BaseDataSettingBL，CostManageBL，SettlementManageBL，ConstantManageBL，PeopleAgencyBLService，DeliverAndRecieveBL，DriverManageBL，LoadBL，ReceivePkgBL，RepertoryManage，SearchPkgInformationBL，SendPkgBL，TransferCenterRecieveBL，TransShipmentBL，TruckManageBL，UserManageBL。其中，CheckBL模块负责总经理审批单据，FormBL模块负责获取统计报表，SystemLogBL模块负责获取系统日志，AccountBL模块负责账户的管理，BaseDataSettingBL负责期初的建账，CostManageBL负责成本的管理，SettlementManageBL负责结算管理，ConstantManageBL负责常量的变更，PeopleAgencyBLService负责人员机构的变更，DeliverAndRecieveBL负责营业厅的接收与派件任务，DriverManageBL负责营业厅的司机信息管理任务，LoadBL负责营业厅的装车任务，ReceivePkgBL负责收件人的收件任务，RepertoryManage负责仓库管理任务，SearchPkgInformationBL负责搜索订单信息任务，SendPkgBL负责寄件任务，TransferCenterRecieveBL负责中转中心中转任务，TransShipmentBL负责装运管理任务，TruckManageBL负责营业厅的车辆管理任务，UserManageBL负责用户管理任务。

**4.2.2客户端业务逻辑层模块局部模块的接口规范**

**4.2.2.1Checkbl模块**

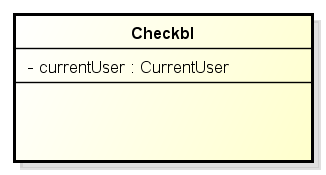
1. **模块概述**

Checkbl模块承担总经理审批单据的功能需求，具体可参见需求规格说明文档功能需求和非功能需求。

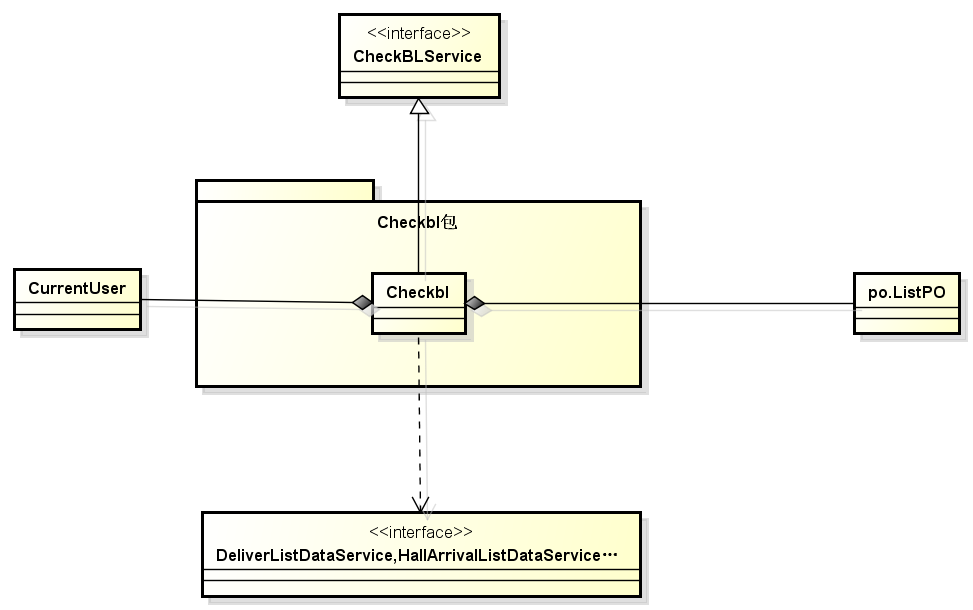
1. **整体结构**

根据体系结构的设计，我们在展示层和业务逻辑层之间添加了CheckBLService接口。在业务逻辑层和数据层之间添加了DeliverListDataService,HallArrivalListDataService,LoadListDataService,OrderListDataService,TransArrivalListDataService,TransShipmentListDataService,MoneyInformationListService,MoneyInListDataService，MoneyOutListDataService，RepertoryInDataService，RepertoryOutDataService，SystemLogDataService接口。

Checkbl类的属性如图所示



Checkbl模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| Checkbl | 实现CheckBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

1. **模块内部类的接口规范**

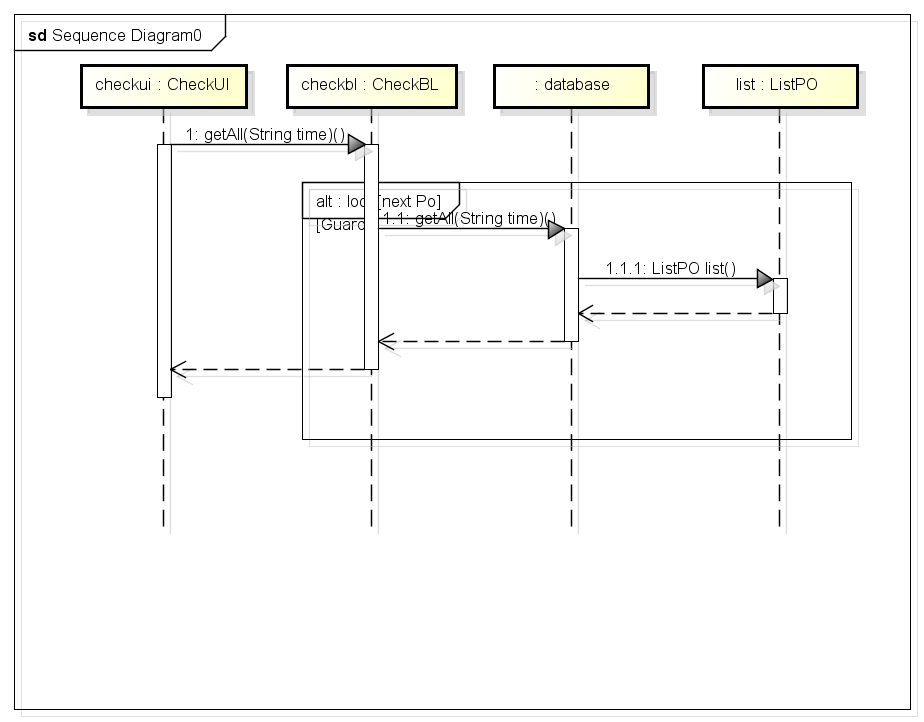
Checkbl的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Checkbl.getAll | 语法 | Public ArrayList<ListVo>getAll(DocType docType,String time) |
| 前置条件 | 总经理选择需要审批的单据类型 |
| 后置条件 | 返回相应的所有单据 |
| Checkbl.passAllDoc | 语法 | Public Boolean passAllDoc(DocType docType,String time) |
| 前置条件 | 总经理审批一个状态下全部单据 |
| 后置条件 | 是否审批成功 |
| Checkbl.passSeceletedDoc | 语法 | Public Boolean passSelectedDoc(DocType docType,ArrayList<ListVO> listVO) |
| 前置条件 | 总经理审批部分单据通过 |
| 后置条件 | 是否审批成功 |
| Checkbl.failSeceletedDoc | 语法 | Public Boolean faliSelectedDoc(DocType doeType,ArrayList<ListVO> listVO) |
| 前置条件 | 总经理审批部分单据 |
| 后置条件 | 是否审批成功 |
| Checkbl.ChangeDoc | 语法 | Public Boolean passSelectedDoc(DocType docType,ArrayList<ListVO> listVO |
| 前置条件 | 总经理修改订单数据 |
| 后置条件 | 修改是否成功 |
| Checkbl.passAllSelected | 语法 | public ResultMessage passAllSelected(DocType docType,ArrayList<String> id) |
| 前置条件 | 总经理输入要通过的订单id |
| 后置条件 | 订单审批为通过 |
| Checkbl.failAllSelected | 语法 | public ResultMessage failAllSelected(DocType docType,ArrayList<String> id) |
| 前置条件 | 总经理输入要未通过的订单id |
| 后置条件 | 订单审批状态为未通过 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| DeliveringListDataService. showAll(String time1,String time2) | 根据时间进行查找多个持久化对象 |
| HallArrivalListDataService. showAll(String time1,String time2) | 根据时间进行查找多个持久化对象 |
| LoadListDataService. showAll(String time1,String time2) | 根据时间进行查找多个持久化对象 |
| OrderListDataService. showAll(String time1,String time2) | 根据时间进行查找多个持久化对象 |
| TransArrivalListDataService. showAll(String time1,String time2) | 根据时间进行查找多个持久化对象 |
| TransShipmentListDataService. showAll(String time1,String time2) | 根据时间进行查找多个持久化对象 |
| MoneyInformationListService. findOnTime(String time1,String time2) | 根据时间进行查找多个持久化对象 |
| MoneyInListDataService. showAll(String start\_day, String end\_day) | 根据时间进行查找多个持久化对象 |
| MoneyOutListDataService. showAll(String start\_day, String end\_day) | 根据时间进行查找多个持久化对象 |
| RepertoryInDataService. showAll(String start\_day, String end\_day) | 根据时间进行查找多个持久化对象 |
| RepertoryOutDataService. showAll(String start\_day, String end\_day) | 根据时间进行查找多个持久化对象 |
| SystemLogDataService. add(SystemLogPO systemLog) | 插入单一持久化对象 |

1. **模块的动态职责**

下图表示在物流信息管理系统中，当总经理输入日期后，审判单据业务逻辑之间相关对象的协作。



**4.2.2.2Formbl模块**

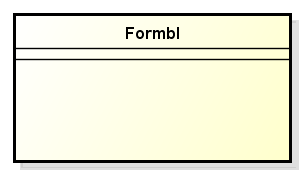
**（1）模块概述**

Formbl模块承担总经理查看统计报表的功能需求，具体可参见需求规格说明文档功能需求和非功能需求。

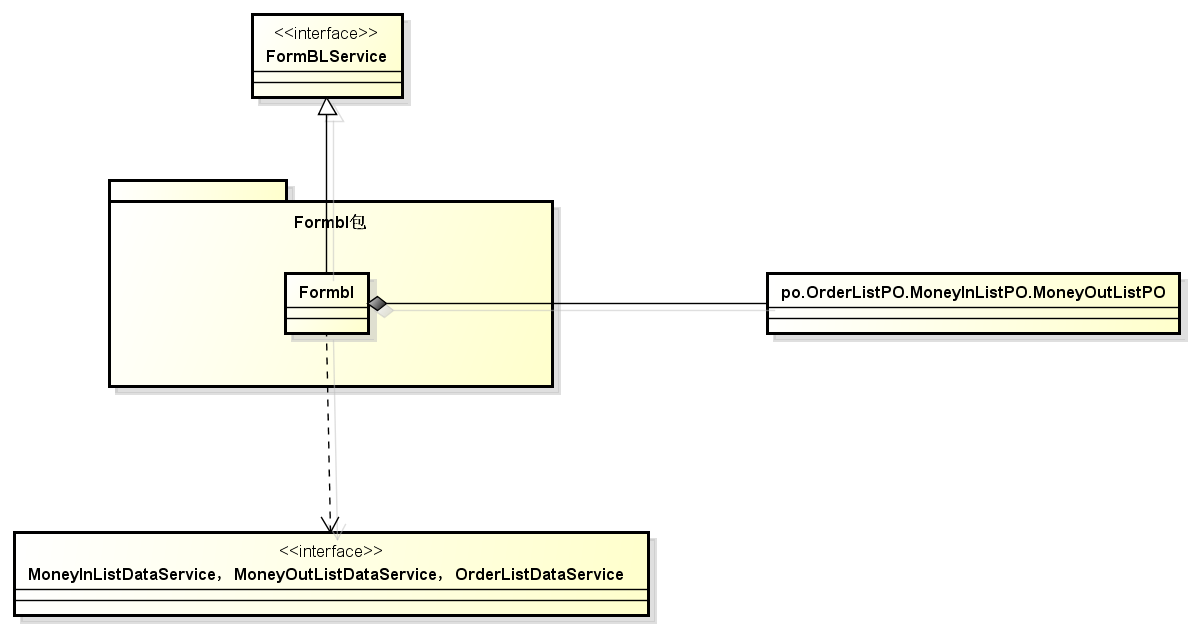
**（2）整体结构**

根据体系结构的设计，我们在展示层和业务逻辑层之间添加了FormBLService接口。在业务逻辑层和数据层之间添加了MoneyInListDataService，MoneyOutListDataService，OrderListDataService接口

Formbl类的属性如图所示



Formbl模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| Formbl | 实现FormBLService的各个方法 |

**（3）模块内部类的接口规范**

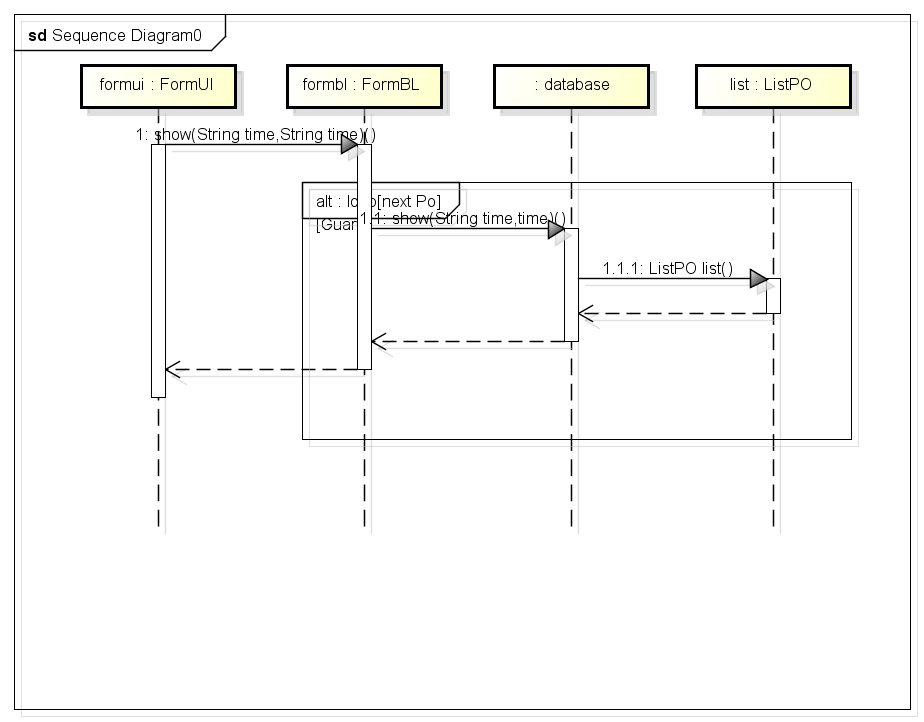
Formbl接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Formbl.getCostandBenefitChart | 语法 | Public ArrayList<MoneyInformationListVo>getCostandBenefitChart (String start\_day, String end\_day) |
| 前置条件 | 财务人员或总经理已经登陆，申请查看成本收益表 |
| 后置条件 | 返回成本收益信息 |
| Formbl.getBusinessCircumstanceChart | 语法 | public ArrayList<? super ListVO> getBusinessCircumstanceChart(String start\_day, String end\_day) |
| 前置条件 | 财务人员或总经理已经登陆，申请查看经营情况表 |
| 后置条件 | 返回经营情况表 |
| Formbl.exportChart | 语法 | public void exportChart(String strat\_day,String end\_day,String path) |
| 前置条件 | 用户已登录，申请打印导出报表 |
| 后置条件 | 导出报表至指定路径 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| MoneyInListDataService. showAll(String start\_day, String end\_day) | 根据时间进行查找多个持久化对象 |
| MoneyOutListDataService. showAll(String start\_day, String end\_day) | 根据时间进行查找多个持久化对象 |
| OrderListDataService. showAll(String time1,String time2) | 根据时间进行查找多个持久化对象 |

1. 模块的动态职责

下图表示在物流信息管理系统中，当总经理输入开始日期和结束日期请求查看成本收益表后，业务逻辑的相关对象之间的协作。



**4.2.2.3SystemLogbl模块**

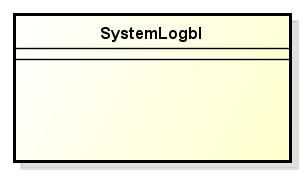
**（1）模块概述**

SystemLogbl模块承担查看系统的功能需求，具体可参见需求规格说明文档功能需求和非功能需求。

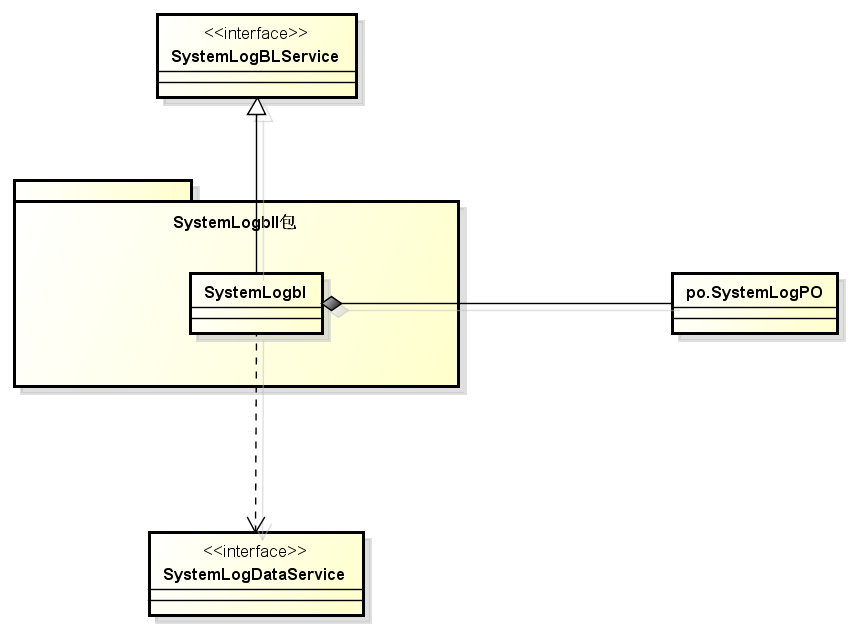
**（2）整体结构**

根据体系结构的设计，我们在展示层和业务逻辑层之间添加了SystemLogBLService接口。在逻辑层和数据层之间添加了SystemLogDataService接口

SystemLogbl类的属性如图所示



SystemLogbl模块的设计如图所示



模块内各个类的职责如下所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| SystemLogbl | 实现SystemLogBLService的各个方法 |

**（3）模块内部类的接口规范**

SystemLogbl接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SystemLogbl.showAll | 语法 | public ArrayList<SystemLogVO> showAll(String time1,String time2) |
| 前置条件 | 用户输入正确格式的日期 |
| 后置条件 | 系统显示匹配日的系统日志查询结果 |
| SystemLogbl.addLogInfo | 语法 | public ResultMessage addLogInfo(SystemLogVO systemLogVO) |
| 前置条件 | 创建新的系统日志 |
| 后置条件 | 返回是否创建信息 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| SystemLogDataService. showAll(String time1,String time2) | 根据时间进行查找多个持久化对象 |
| SystemLogDataService. add(SystemLogPO systemLog) | 插入单一持久化对象 |

**4.2.2.4Accountbl模块**

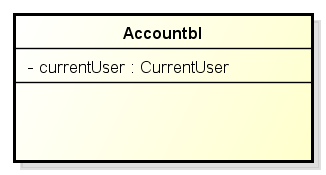
1. **模块概述**

Accountbl模块承担账户管理的功能需求，具体可参见需求规格说明文档功能需求和非功能需求。

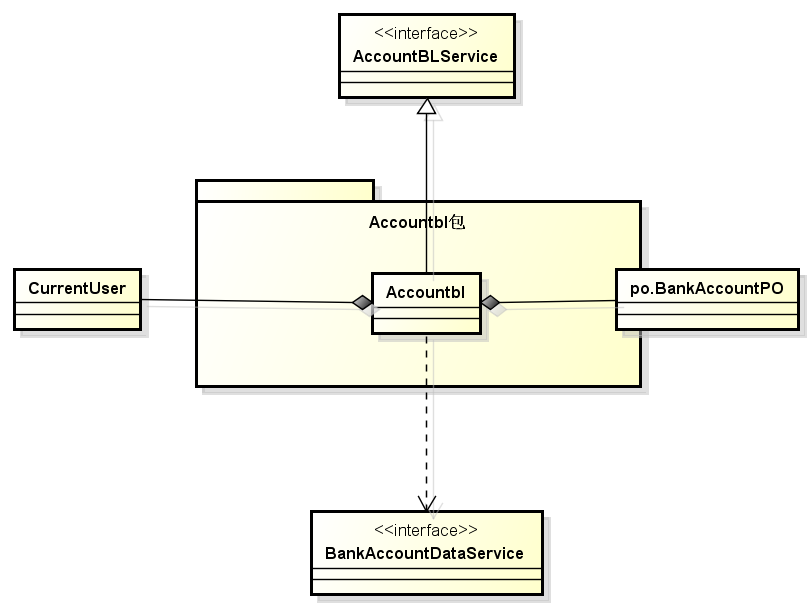
1. **整体结构**

根据体系结构的设计，我们在展示层和业务逻辑层之间添加了AccountBLService接口。在逻辑层和数据层之间添加了BankAccountDataService接口。

Accountbl类的属性如图所示



Accountbl模块的设计如图所示



模块内各个类的职责如下所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| Accountbl | 实现AccountBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块内部类的接口规范**

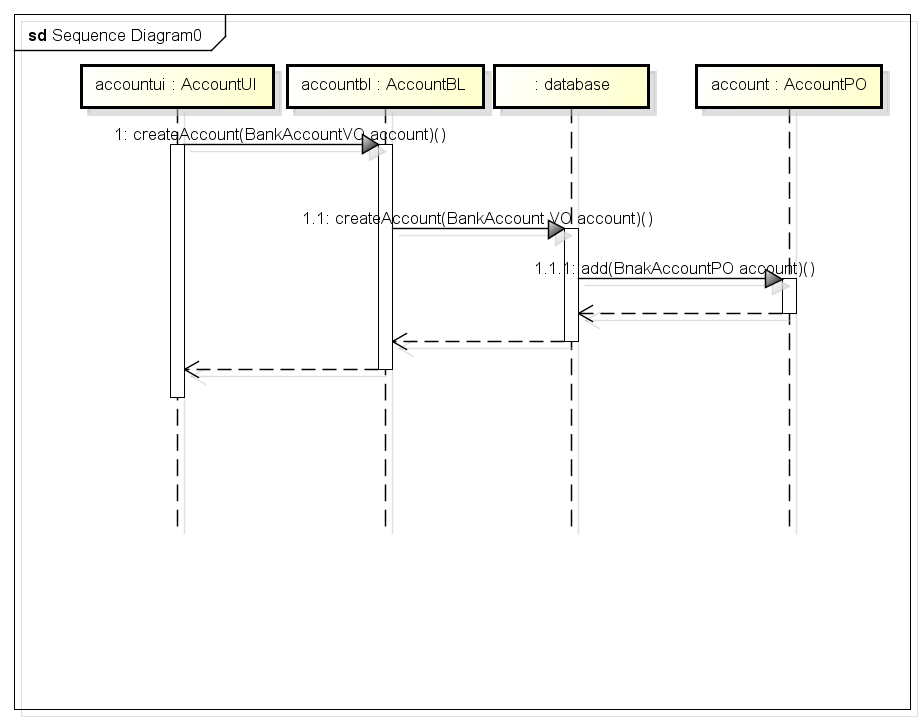
Accountbl接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| Accountbl.createCount | 语法 | public ResultMessage createCount(BankAccountVO countVo) |
| 前置条件 | 财务人员已经登陆 |
| 后置条件 | 增加账户 |
| Accountbl.removeCount | 语法 | public ResultMessage removeCount(String name) |
| 前置条件 | 财务人员已经登陆 |
| 后置条件 | 删除账户 |
| Accountbl.updateCount | 语法 | public ResultMessage updateCount(BankAccountVO countVo) |
| 前置条件 | 财务人员已经登陆 |
| 后置条件 | 更改账户 |
| Accountbl.searchCount | 语法 | public BankAccountVO searchCount(String keyword) |
| 前置条件 | 财务人员已经登陆 |
| 后置条件 | 查询账户 |
| Accountbl.show | 语法 | public ArrayList<BankAccountVO> show() |
| 前置条件 | 请求所有账户信息 |
| 后置条件 | 显示所有账户信息 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| BankAccountDataService. add(BankAccountPO countpo) | 插入单一持久化对象 |
| BankAccountDataService. delete(String name) | 删除单一持久化对象 |
| BankAccountDataService. update(BankAccountPO countpo) | 更新单一持久化对象 |
| BankAccountDataService. find(String keyword) | 根据关键字查找单一持久化对象 |
| BankAccountDataService. showAll() | 显示多个持久化对象 |

**(4)模块的动态职责**

下图表示在物流信息管理系统中，当财务人员输入账户信息，请求增加账户信息时，业务逻辑相关对象的协作。



**4.2.2.5BaseDataSettingbl模块**

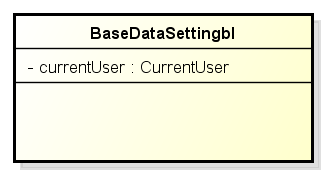
**（1）模块概述**

BaseDataSettingbl模块承担期初建账的功能需求，具体可参见需求规格说明文档功能需求和非功能需求。

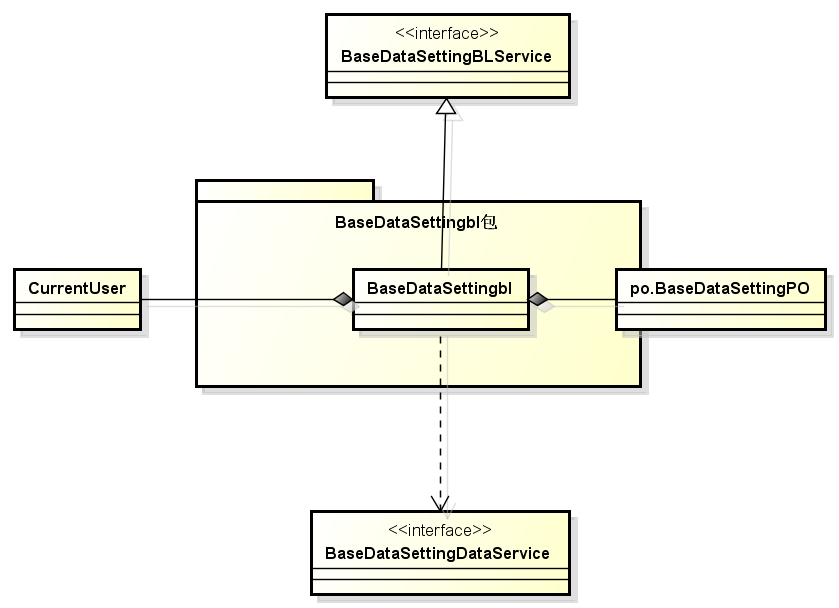
**（2）整体结构**

根据体系结构的设计，我们在展示层和业务逻辑层之间添加了BaseDataSettingBLService接口。在逻辑层和数据层之间添加了BaseDataSettingDataService接口

BaseDataSettingbl类的属性如图所示



BaseDataSettingbl模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| BaseDataSettingbl | 实现BaseDataSettingBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块内部类的接口规范**

BaseDataSettingbl接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| BaseDataSettingbl.restart | 语法 | public ResultMessage restart() |
| 前置条件 | 期初建账，输入全新的机构，库存，财务信息 |
| 后置条件 | 无 |
| BaseDataSettingbl.addInstitutionInfo | 语法 | public ResultMessage addInstitutionInfo(AgencyVO agencyVO) |
| 前置条件 | 财务人员开始期初建账操作，输入机构信息 |
| 后置条件 | 返回操作反馈 |
| BaseDataSettingbl. addCommodityInfo | 语法 | public ResultMessage addCommodityInfo(RepertoryInfoVO repertoryVO) |
| 前置条件 | 期初建账操作，输入库存信息 |
| 后置条件 | 返回操作反馈 |
| BaseDataSettingbl.addTruckInfo | 语法 | public ResultMessage addTruckInfo(TruckVO truck) |
| 前置条件 | 期初建账，输入车辆信息 |
| 后置条件 | 返回操作反馈 |
| BaseDataSettingbl.addFinanceInfo | 语法 | public ResultMessage addFinanceInfo(BankAccountVO bankAccountVO) |
| 前置条件 | 无基础期初建账操作，输入财务信息 |
| 后置条件 | 返回操作反馈 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| BaseDataSettingDataService. addInstitutionInfo(AgencyPO agencyPO) | 插入单一持久化对象 |
| BaseDataSettingDataService. addCommodityInfo(RepertoryInfoPO repertoryPO) | 插入单一持久化对象 |
| BaseDataSettingDataService. addTruckInfo(TruckPO truck) | 插入单一持久化对象 |
| BaseDataSettingDataService. addFinanceInfo(BankAccountPO bankAccountPO) | 插入单一持久化对象 |
| BaseDataSettingDataService. show() | 显示多个持久化对象 |

**4.2.2.6 CostManagementbl模块**

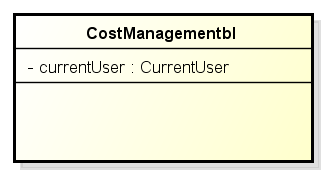
**（1）模块概述**

CostManagementbl模块承担成本管理的功能需求，具体可参见需求规格说明文档功能需求和非功能需求。

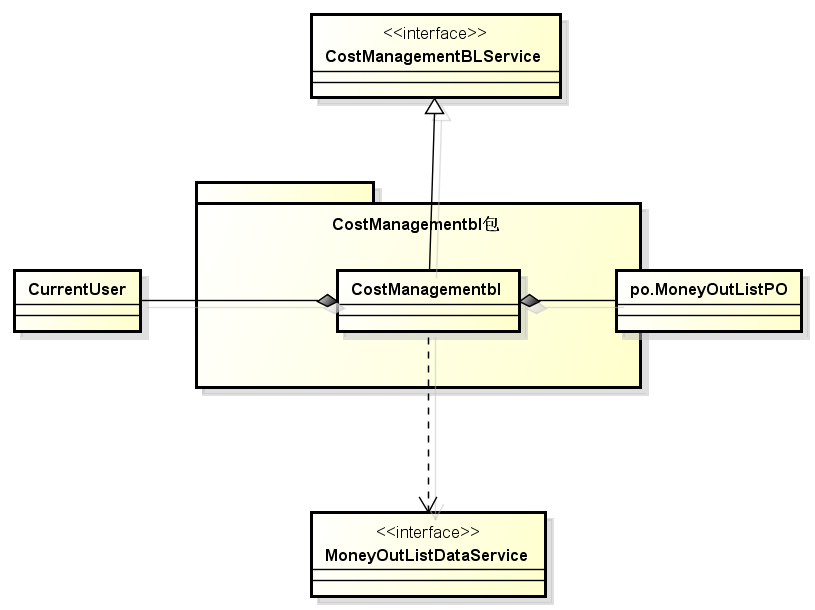
**（2）整体结构**

根据体系结构的设计，我们在展示层和业务逻辑层之间添加了CostManagementBLService接口。在逻辑层和数据层之间添加了MoneyOutListDataService接口

CostManagementbl类的属性如图所示



CostManagementbl的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| CostManagementbl | 实现CostManagementBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块内部类的接口规范**

CostManagementbl接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| CostManagementbl. createMoneyOutlist | 语法 | public ResultMessage createMoneyOutlist(MoneyOutListVO moneyOut) |
| 前置条件 | 财务人员已经登陆 |
| 后置条件 | 新建付款单 |
| CostManagementbl. createMoneyOutlistId | 语法 | public String createMoneyOutListId() |
| 前置条件 | 财务人员要求获得下一个付款单的id |
| 后置条件 | 返回id |
| CostManagementbl.search | 语法 | public ArrayList<MoneyOutListVO> search(String start\_day,String end\_day); |
| 前置条件 | 财务人员输入起始日期和结束日期，要求查找付款单 |
| 后置条件 | 返回付款单的集合 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| MoneyOutListDataService. add(MoneyOutListPO moneyOutList) | 插入单一持久化对象 |

**4.2.2.7SettlementManagebl模块**

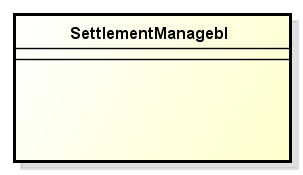
**（1）模块概述**

SettlementManagebl模块承担查看收款单的功能需求，具体可参见需求规格说明文档功能需求和非功能需求。

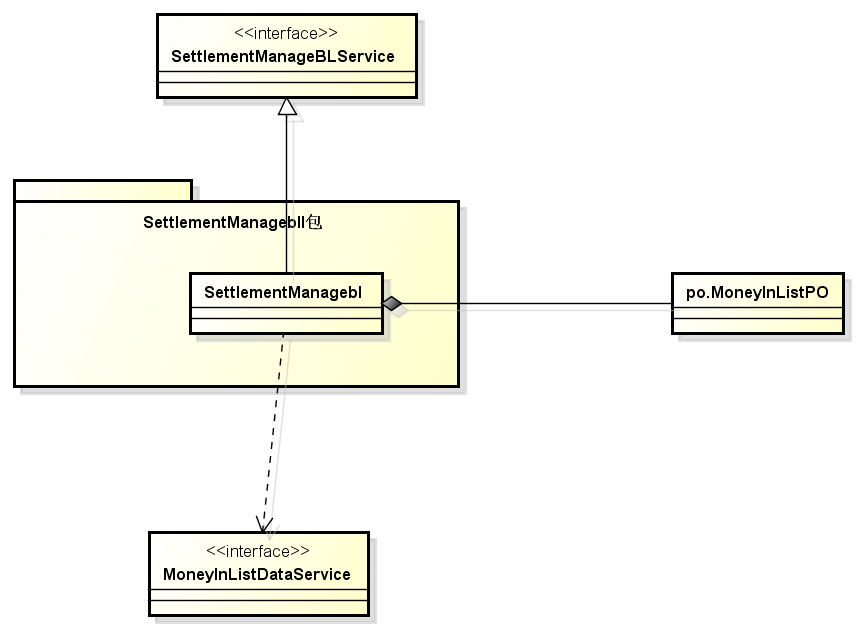
**（2）整体结构**

根据体系结构的设计，我们在展示层和业务逻辑层之间添加了SettlementManageBLService接口。在逻辑层和数据层之间添加了MoneyInListDataService接口

SettlementManagebl类的属性如图所示



SettlementManagebl模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| SettlementManagebl | 实现SettlementManageBLService的各个方法 |

**（3）模块内部类的接口规范**

SettlementManagebl接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| SettlementManagebl. search | 语法 | public ArrayList<MoneyInListVO> search(String start\_day, String end\_day,String hall\_id) |
| 前置条件 | 财务人员已经登陆 |
| 后置条件 | 系统返回付款单的集合 |
| SettlementManagebl.searchbyhall | 语法 | public ArrayList<MoneyInListVO> searchbyhall(String start\_day, String end\_day,String hall\_id) |
| 前置条件 | 财务人员输入起始日期、结束日期和营业厅id |
| 后置条件 | 系统返回收款单的集合 |
| SettlementManagebl.createMoneyInList | 语法 | public ResultMessage createMoneyInList(MoneyInListVO moneyin) |
| 前置条件 | 输入收款单 |
| 后置条件 | 返回操作是否成功 |
| SettlementManagebl.createMoneyInListId | 语法 | public String createMoneyInListId(); |
| 前置条件 | 营业厅业务员要求产生收款单id |
| 后置条件 | 返回相应的收款单id |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| MoneyInListDataService. showAll(String start\_day, String end\_day) | 根据时间进行查找多个持久化对象 |

**4.2.2.8 ConstantManagebl模块**

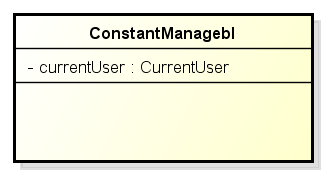
**（1）模块概述**

ConstantManagebl模块承担制定距离价格常量的功能需求，具体可参见需求规格说明文档功能需求和非功能需求。

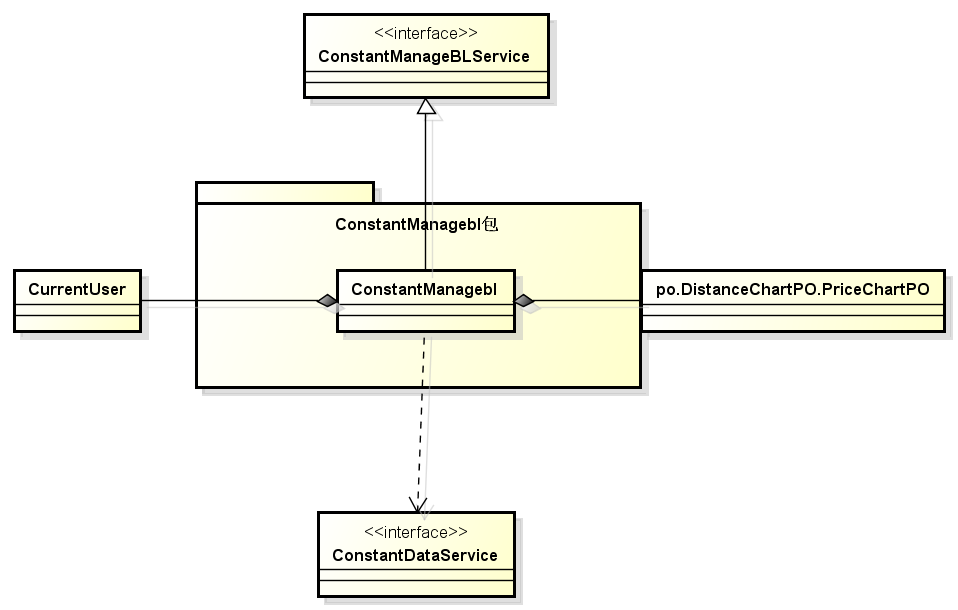
**（2）整体结构**

根据体系结构的设计，我们在展示层和业务逻辑层之间添加了ConstantManageBLService接口。在逻辑层和数据层之间添加了ConstantDataService接口。

ConstantManagebl类的属性如图所示



ConstantManagebl模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| ConstantManagebl | 实现ConstantManageBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块内部类的接口规范**

ConstantManagebl接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| ConstantManagebl.addCity | 语法 | public ResultMessage addCity(String name) |
| 前置条件 | 添加城市 |
| 后置条件 | 返回操作反馈 |
| ConstantManagebl.showDistanceChart | 语法 | public DistanceChartVO showDistanceChart() |
| 前置条件 | 请求城市距离情况表 |
| 后置条件 | 显示城市距离情况表 |
| ConstantManagebl.showPriceChart | 语法 | public DistanceChartVO showDistanceChart() |
| 前置条件 | 请求价格常量情况表 |
| 后置条件 | 显示价格常量情况表 |
| ConstantManagebl.submitDistance | 语法 | public ResultMessage submitDistance(String newDistance,String cityA,String cityB) |
| 前置条件 | 输入城市信息和距离信息 |
| 后置条件 | 修改城市和距离信息 |
| ConstantManagebl.submitPrice | 语法 | public ResultMessage submitPrice(String newPrice,String otherInformation) |
| 前置条件 | 输入价格和类型 |
| 后置条件 | 返回操作反馈 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| ConstantDataService. setDistance(String newDistance,String cityA,String cityB) | 更新单一持久化对象 |
| ConstantDataService. setPrice(String newPrice,String otherInformation) | 更新单一持久化对象 |
| ConstantDataService. addCity(String name) | 插入单一持久化对象 |
| ConstantDataService. showDistanceChart() | 显示多个持久化对象 |
| ConstantDataService. showPriceChart() | 显示多个持久化对象 |

**4.2.2.9PeopleAgencybl模块**

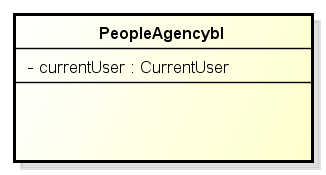
**（1）模块概述**

PeopleAgencybl模块承担人员机构管理的功能需求，具体可参见需求规格说明文档功能需求和非功能需求。

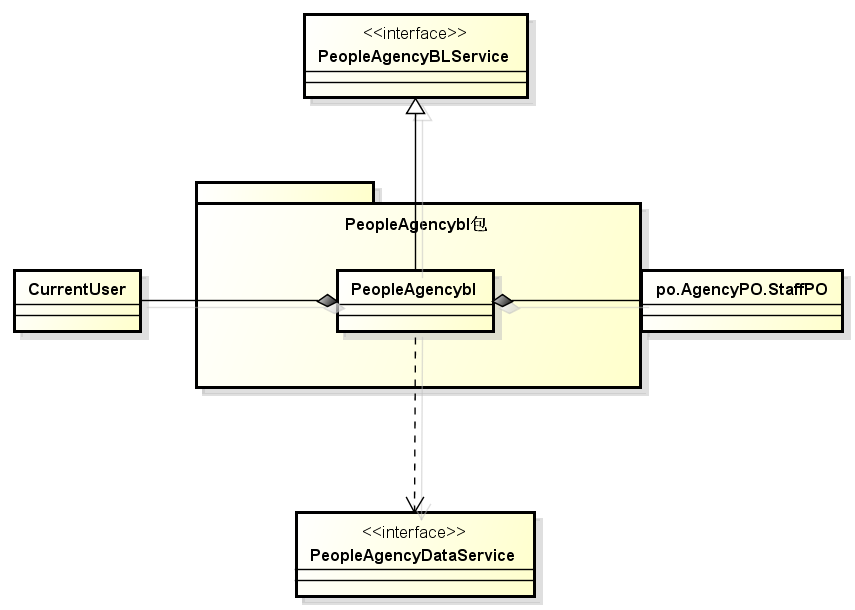
**（2）整体结构**

根据体系结构的设计，我们在展示层和业务逻辑层之间添加了PeopeAgencyBLService接口。在逻辑层和数据层之间添加了PeopleAgencyDataService接口。

PeopleAgencybl类的属性如图所示



PeopleAgencybl模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| PeopleAgencybl | 实现PeopleAgencyBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块内部类的接口规范**

PeopleAgencybl接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| PeopleAgencybl.salaryManage | 语法 | public ResultMessage salaryManage(String userID,String salary) |
| 前置条件 | 要调整的用户编号新的工资方案 |
| 后置条件 | 返回操作反馈 |
| PeopleAgencybl.addAgency | 语法 | public ResultMessage addAgency(AgencyVO agency) |
| 前置条件 | 总经理选择新建机构 |
| 后置条件 | 机构信息是否合理合法 |
| PeopleAgencybl.deleteAgency | 语法 | public ResultMessage deleteAgency(String agencyId) |
| 前置条件 | 总经理选择删除机构 |
| 后置条件 | 返回操作反馈 |
| PeopleAgencybl.updateAgency | 语法 | public ResultMessage updateAgency(AgencyVO agency) |
| 前置条件 | 总经理选择修改机构信息(包括对机构人员的增删改查) |
| 后置条件 | 返回信息是否合理 |
| PeopleAgencybl.showAgency | 语法 | public ArrayList<AgencyVO> showAgency() |
| 前置条件 | 请求显示机构信息 |
| 后置条件 | 显示所有机构信息 |
| PeopleAgencybl.FindAgency | 语法 | public AgencyVO findAgency(String id) |
| 前置条件 | 输入机构编号 |
| 后置条件 | 返回机构信息 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| PeopleAgencyDataService. salaryManage(String userID,String salary) | 更新单一持久化对象 |
| PeopleAgencyDataService. add(AgencyPO agency) | 插入单一持久化对象 |
| PeopleAgencyDataService. delete(AgencyPO agency) | 删除单一持久化对象 |
| PeopleAgencyDataService. update(AgencyPO agency) | 更新单一持久化对象 |
| PeopleAgencyDataService. find(String id) | 根据ID进行查找单一持久化对象 |
| PeopleAgencyDataService. showAll() | 显示多个持久化对象 |

**4.2.2.10** **DeliverAndRecieveBL模块**

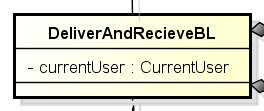
**（1）模块概述**

DeliverAndRecieveBL模块承担了营业厅接收与派件的功能需求，具体参见需求规格说明文档和相关非功能需求。

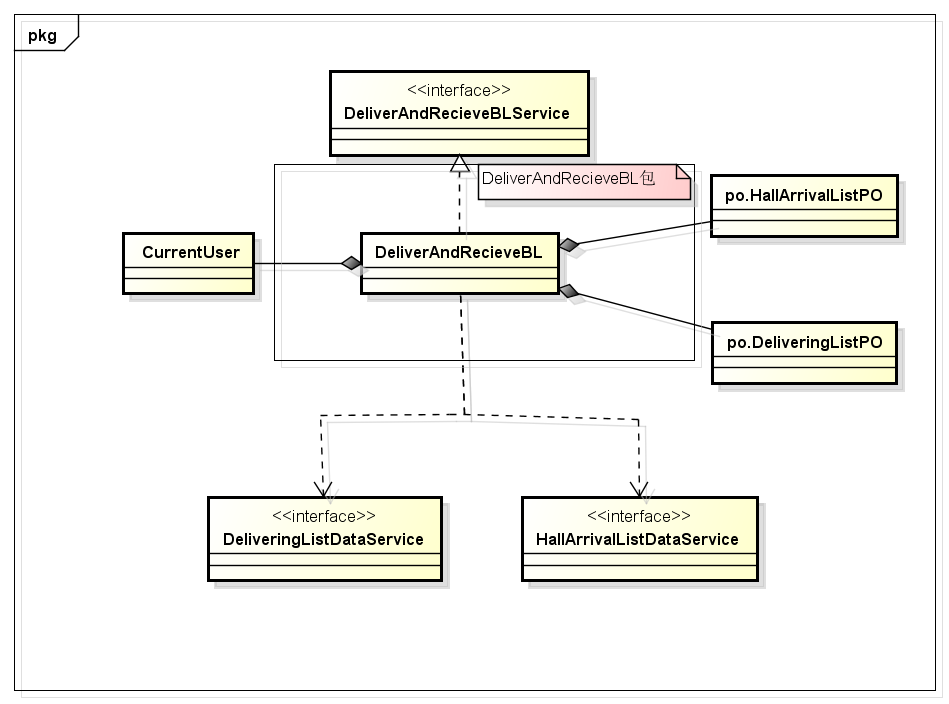
**（2）整体结构**

根据体系结构设计，我们在展示层和业务逻辑层之间添加了DeliveringListBLService接口。在业务逻辑层和数据层之间添加了DeliveringListDataService和HallArrivalListDataService接口。

DeliverAndRecieveBL类的属性如图所示



DeliverAndRecieveBL模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| DeliverAndRecieveBL | 实现DeliverAndRecieveService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

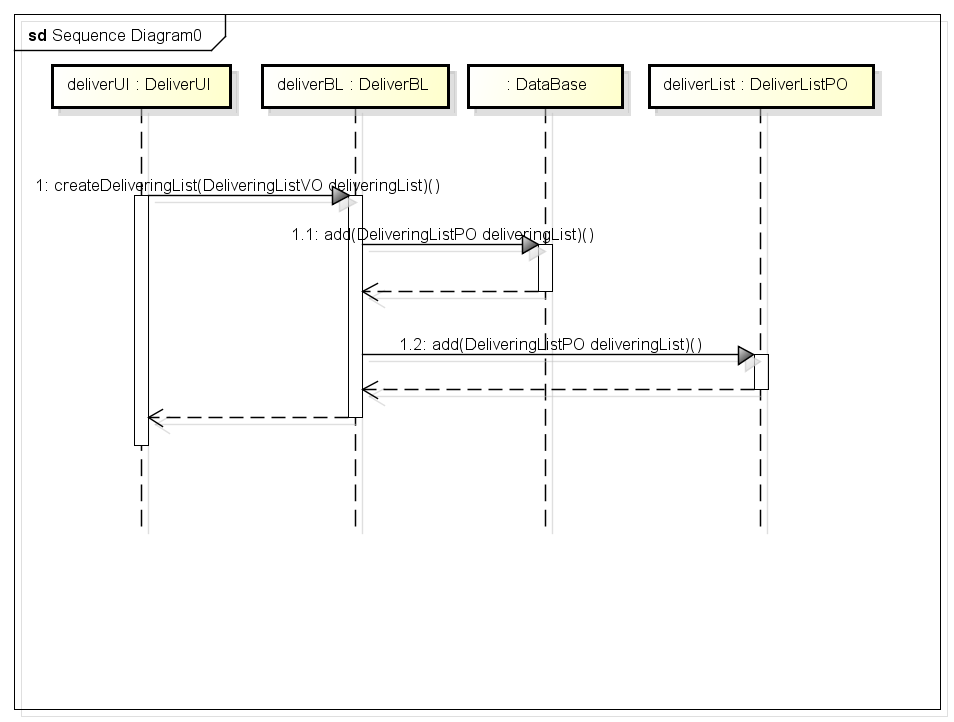
**（3）模块的接口规范（供给接口和需要接口）**

DeliverAndRecieveBL模块的接口规范如下表所示

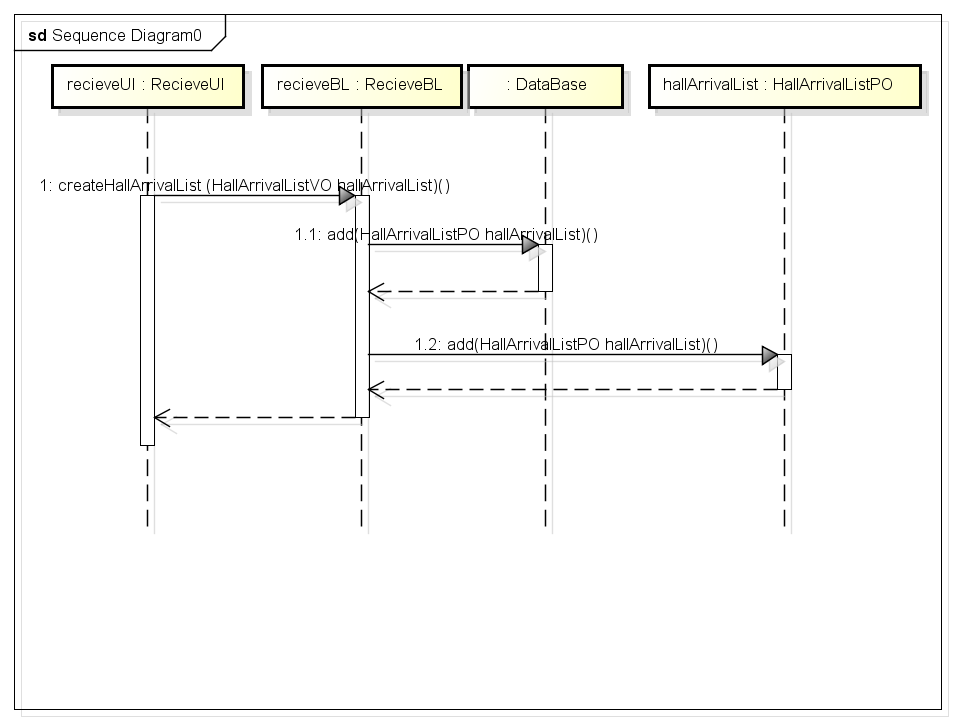
|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| createHallArrivalList | **语法** | Public ResultMessage createHallArrivalList (HallArrivalListVO hallArrivalList) |
| **前置条件** | 输入VO信息格式正确 |
| **后置条件** | 新建一个VO |
| createDeliveringList | **语法** | public ResultMessage createDeliveringList (DeliveringListVO deliveringList) |
| **前置条件** | 输入VO信息格式正确 |
| **后置条件** | 新建一个VO |
| createHallArrivalListId | **语法** | public String createHallArrivalListId(); |
| **前置条件** | 要求获得营业厅到达单id |
| **后置条件** | 系统返回营业厅到达单id |
| createDeliveringListId | **语法** | public String createDeliveringListId(); |
| **前置条件** | 要求获得派件单id |
| **后置条件** | 系统返回派件单id |
| getLoadList | **语法** | public TransArrivalVO getLoadList(String loadlistid) |
| **前置条件** | 输入装车单号来获得上一个位置的货物信息 |
| **后置条件** | 返回TransArrivalVO |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| HallArrivalListDataService.add | 向系统中添加营业厅到达单 | |
| HallArrivalListDataService.delete | 系统中删除营业厅到达单 | |
| HallArrivalListDataService.update | 系统中修改营业厅到达单 | |
| HallArrivalListDataService.find | 系统中寻找营业厅到达单 | |
| DeliveringListDataService.add | 系统中添加派件单 | |
| DeliveringListDataService.delete | 系统中删除派件单 | |
| DeliveringListDataService.update | 系统中修改派件单 | |
| DeliveringListDataService.find | 系统中寻找派件单 | |

**（4）模块的动态职责**

下图表示在物流管理系统中，当营业厅管理员输入派件单相关信息之后，派件业务逻辑处理的相关对象之间的协作。

****

下图表示在物流管理系统中，当营业厅管理员输入接收单相关信息之后，接收业务逻辑处理的相关对象之间的协作。

****

**4.2.2.11** **DriverManageBL模块**

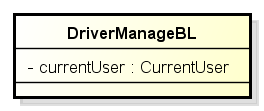
**（1）模块概述**

DriverManageBL模块承担了营业厅司机信息管理的功能需求，具体参见需求规格说明文档和相关非功能需求。

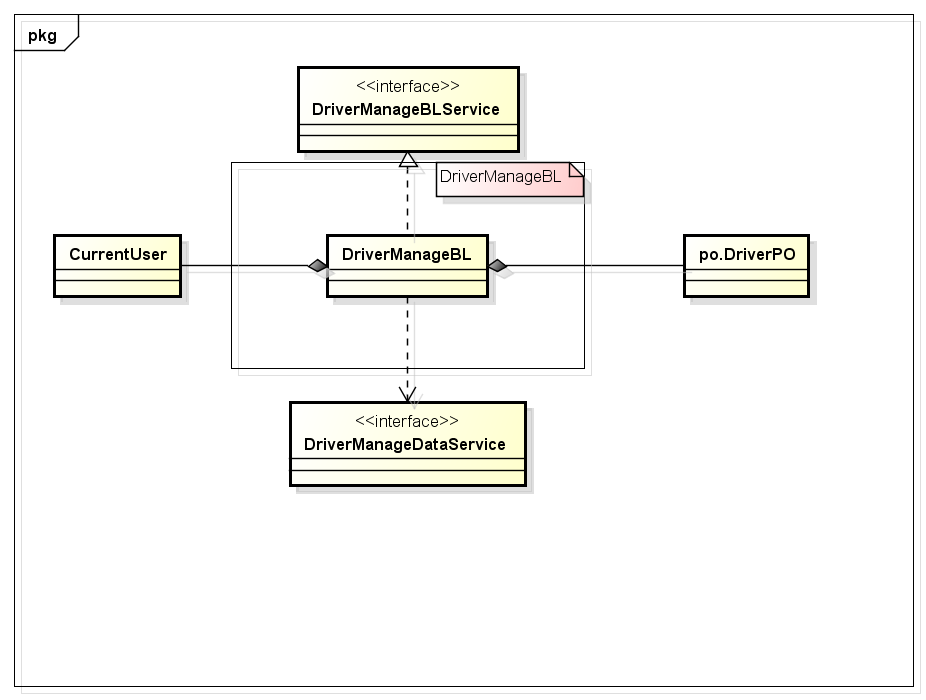
**（2）整体结构**

根据体系结构设计，我们在展示层和业务逻辑层之间添加了DriverManageBLService接口。在业务逻辑层和数据层之间添加了DriverDataService接口。

DriverManageBL类的属性如图所示



DriverManageBL模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| DriverManageBL | 实现DriverManageBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

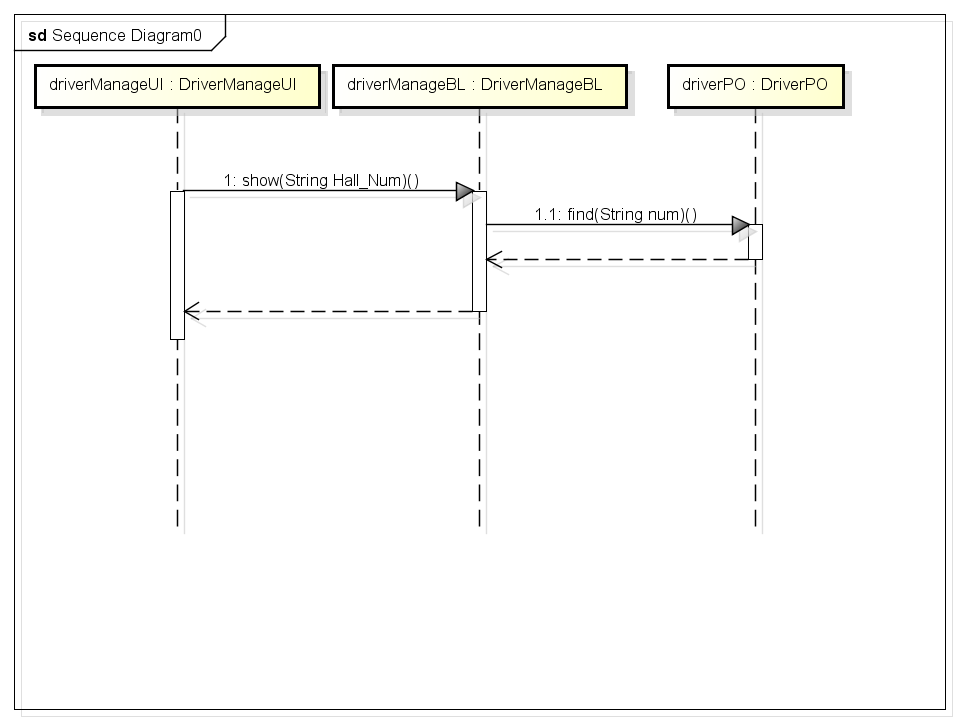
**（3）模块的接口规范**

DriverManageBL模块的接口规范如下表所示

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| show | **语法** | public ArrayList<DriverVO> show(String Hall\_Num) |
| **前置条件** | Name输入合法存在 |
| **后置条件** | 查找该营业厅全部list返回VO |
| update | **语法** | public ResultMessage update(DriverVO driver); |
| **前置条件** | VO修改信息合理 |
| **后置条件** | 更新一个VO |
| delete | **语法** | public ResultMessage delete(DriverVO driver) |
| **前置条件** | VO输入格式正确 |
| **后置条件** | 删除一个司机信息 |
| add | **语法** | public ResultMessage add(DriverVO driver) |
| **前置条件** | VO格式输入正确 |
| **后置条件** | 新建一个VO |
| find | **语法** | public DriverVO find(String num) |
| **前置条件** | Num格式输入正确 |
| **后置条件** | 查找是否存在并返回一个VO |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| DriverDataService.add | 向系统中添加一个司机 | |
| DriverDataService.delete | 系统中删除一个司机 | |
| DriverDataService.update | 系统中修改一个司机 | |
| DriverDataService.find | 系统中寻找一个司机 | |

**（4）模块的动态职责**

下图表示在物流信息管理系统中，营业厅管理员选择显示本营业厅所有司机时，司机管理业务逻辑处理的相关对象之间的协作。



**4.2.2.12** **LoadBL模块**

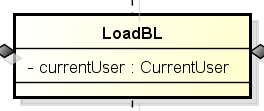
**（1）模块概述**

LoadBL模块承担了营业厅装车的功能需求，具体参见需求规格说明文档和相关非功能需求。

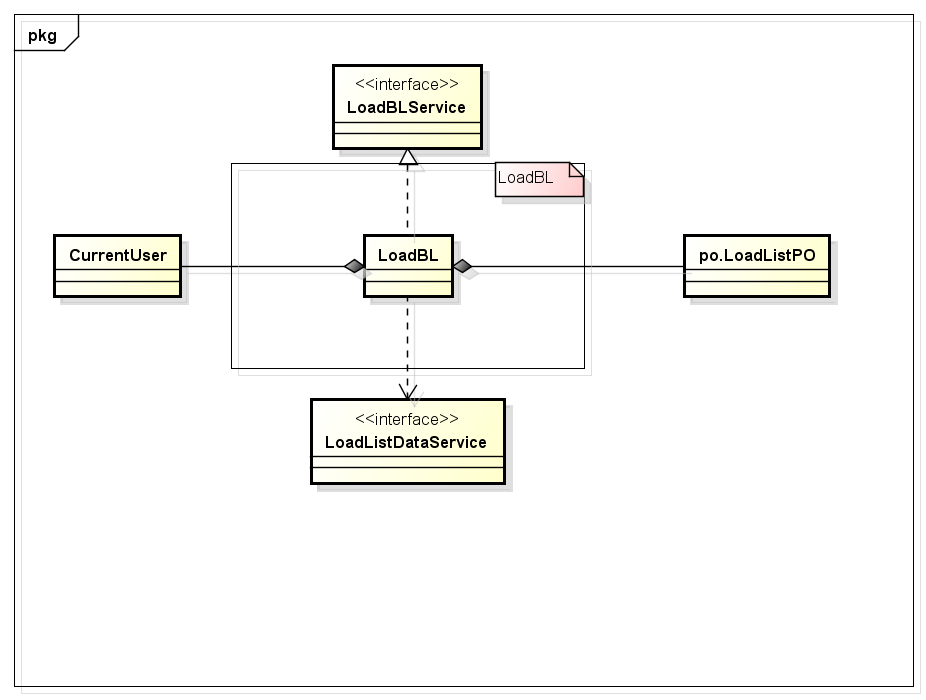
**（2）整体结构**

根据体系结构设计，我们在展示层和业务逻辑层之间添加了LoadBLService接口。在业务逻辑层和数据层之间添加了LoadListDataService接口。

LoadBL类的属性如图所示



LoadBL模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| LaodBL | 实现LoadBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| createLoadListId | **语法** | public String createLoadlistId(); |
| **前置条件** | 要求获得装车单号 |
| **后置条件** | 返回相应机构本次的装车单号 |
| createLoadlist | **语法** | public ResultMessage createLoadlist(LoadListVO loadListVO) |
| **前置条件** | 所有数据录入完成 |
| **后置条件** | 新建一个LoadListVO |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| DriverDataService.add | 向系统中添加一个司机 | |
| DriverDataService.delete | 系统中删除一个司机 | |
| DriverDataService.update | 系统中修改一个司机 | |
| DriverDataService.find | 系统中寻找一个司机 | |

**（4）模块的动态职责**

**4.2.2.13ReceivePkgBL模块**

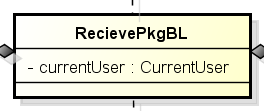
**（1）模块概述**

ReceivePkgBL模块承担了收件人收件的功能需求，具体参见需求规格说明文档和相关非功能需求。

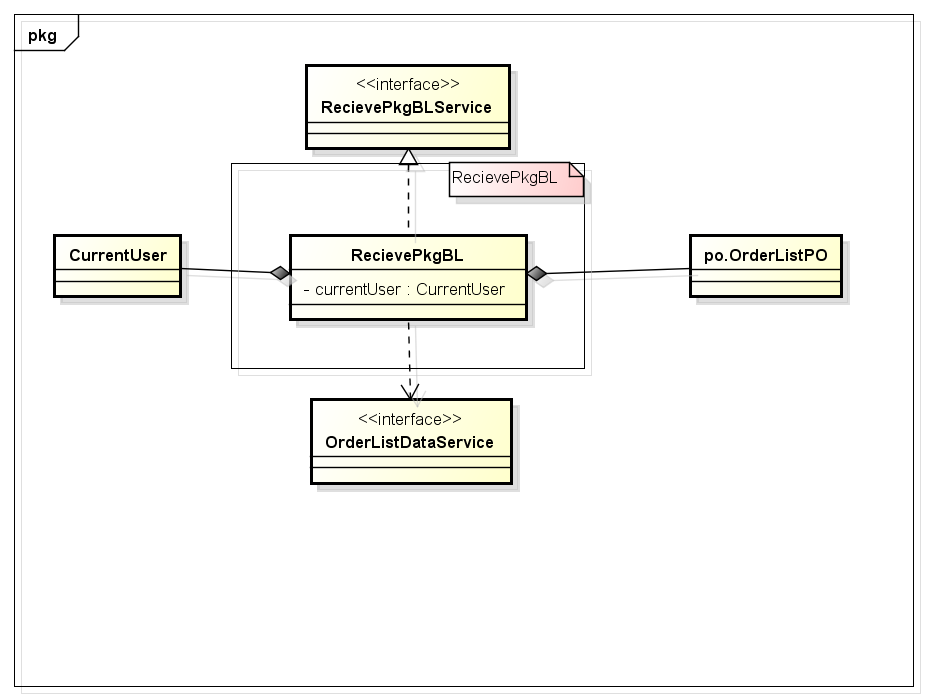
**（2）整体结构**

根据体系结构设计，我们在展示层和业务逻辑层之间添加了ReceivePkgBLService接口。在业务逻辑层和数据层之间添加了OrderListDataService接口。

ReceivePkgBL类的属性如图所示



ReceivePkgBL模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| RecievePkgBL | 实现RecievePkgBL Service的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| findOrderlist | **语法** | public OrderListVO findOrderlist(String orderlistId) |
| **前置条件** | Num输入格式正确 |
| **后置条件** | 查找 若存在该order则返回该list |
| confirmRecieve | **语法** | public ResultMessage confirmRecieve(ReceiverVO receiver) |
| **前置条件** | 所有数据录入完成 |
| **后置条件** | 新建一个LoadListVO |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| OrderListDataService.add | 向系统中添加订单 | |
| OrderListDataService.delete | 系统中删除订单 | |
| OrderListDataService.update | 系统中修改订单 | |
| OrderListDataService.find | 系统中寻找订单 | |

**（4）模块的动态职责**

**4.2.2.14RepertoryManageBL模块**

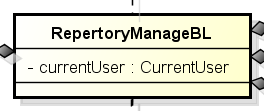
**（1）模块概述**

RepertoryManageBL模块承担了收件人收件的功能需求，具体参见需求规格说明文档和相关非功能需求。

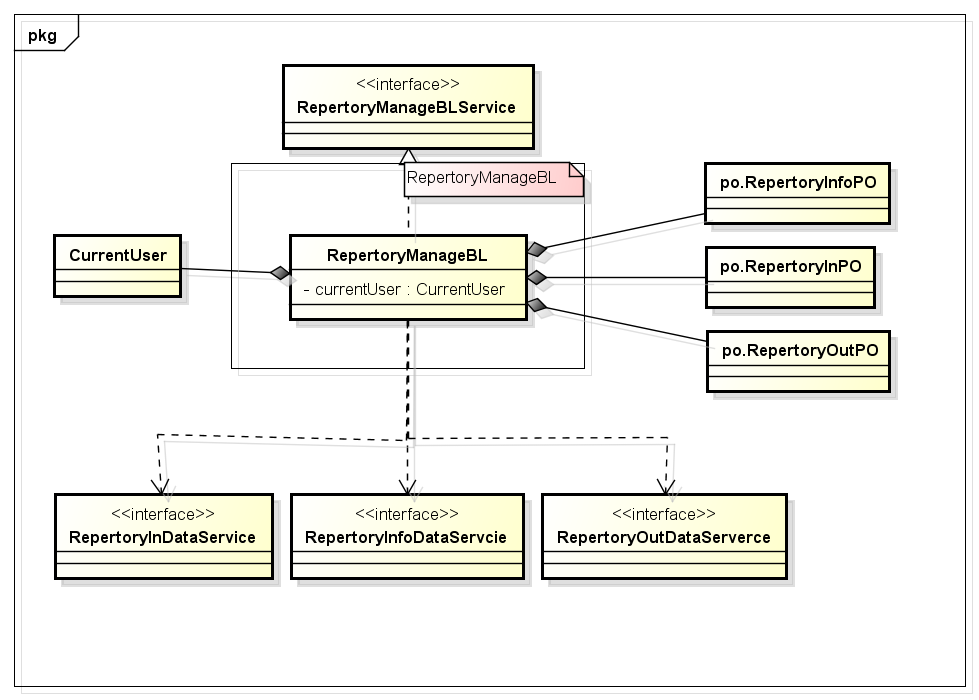
**（2）整体结构**

根据体系结构设计，我们在展示层和业务逻辑层之间添加了RepertoryManageBLService接口。在业务逻辑层和数据层之间添加了RepertoryInfoDataServcie, RepertoryInDataService,RepertoryOutDataServerce接口。

RepertoryManageBL类的属性如图所示



RepertoryManageBL模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| RepertoryManageBL | 实现RepertoryManageBL Service的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| createInputRepertory | **语法** | Public ResultMessage createInputRepertory (RepertoryInVO repertoryIn) |
| **前置条件** | 入库单VO符合语法 |
| **后置条件** | 新建一个VO |
| createOutputRepertory | **语法** | public ResultMessage createOutputRepertory (RepertoryOutVO repertoryOut) |
| **前置条件** | 出库单VO符合语法 |
| **后置条件** | 新建一个VO |
| showRepertory | **语法** | public RepertoryInfoVO showRepertory (String start\_time,String end\_time) |
| **前置条件** | 两个时间点输入合法，可以相同 |
| **后置条件** | 返回一个仓库实体vo |
| repertoryAlarm | **语法** | public ResultMessage repertoryAlarm() |
| **前置条件** | 无 |
| **后置条件** | 返回库存是否报警 |
| repertoryAdjust | **语法** | public ResultMessage repertoryAdjust(String preLoaction,String latLoaction); |
| **前置条件** | 两个位置输入合法 |
| **后置条件** | 先前位置上的订单被放到后来位置上 |
| createRepertoryInId | **语法** | public String createRepertoryInId(); |
| **前置条件** | 要求获得入库单号 |
| **后置条件** | 系统返回响应的入库单号 |
| createRepertoryOutId | **语法** | public String createRepertoryOutId(); |
| **前置条件** | 要求获得出库单号 |
| **后置条件** | 系统返回响应的出库单号 |
| exportRepertoryInformation | **语法** | public void exportRepertoryInformation(String repertoryname,String time,String path); |
| **前置条件** | 要求导出库存信息 |
| **后置条件** | 导出库存信息的Excel表格 |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| RepertoryInDataService.add | 系统中添加入库单 | |
| RepertoryInDataService.delete | 系统中删除入库单 | |
| RepertoryInDataService.update | 系统中修改入库单 | |
| RepertoryInDataService.findOntime | 系统中按照时间寻找入库单 | |
| RepertoryInDataService.findOnID | 系统中按照ID寻找入库单 | |
| RepertoryOutDataService.add | 系统中添加出库单 | |
| RepertoryOutDataService.delete | 系统中删除出库单 | |
| RepertoryOutDataService.update | 系统中修改出库单 | |
| RepertoryOutDataService.findOntime | 系统中按照时间寻找出库单 | |
| RepertoryOutDataService.findOnID | 系统中按照ID寻找出库单 | |
| RepertoryInfoDataService.add | 系统中的仓库添加货物 | |
| RepertoryInfoDataService.delete | 系统中的仓库减少货物 | |

**（4）模块的动态职责**

**4.2.2.15SearchPkgInformationBL模块**

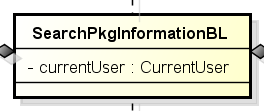
**（1）模块概述**

SearchPkgInformationBL模块承担了寄（收）件人查询订单详情的功能需求，具体参见需求规格说明文档和相关非功能需求。

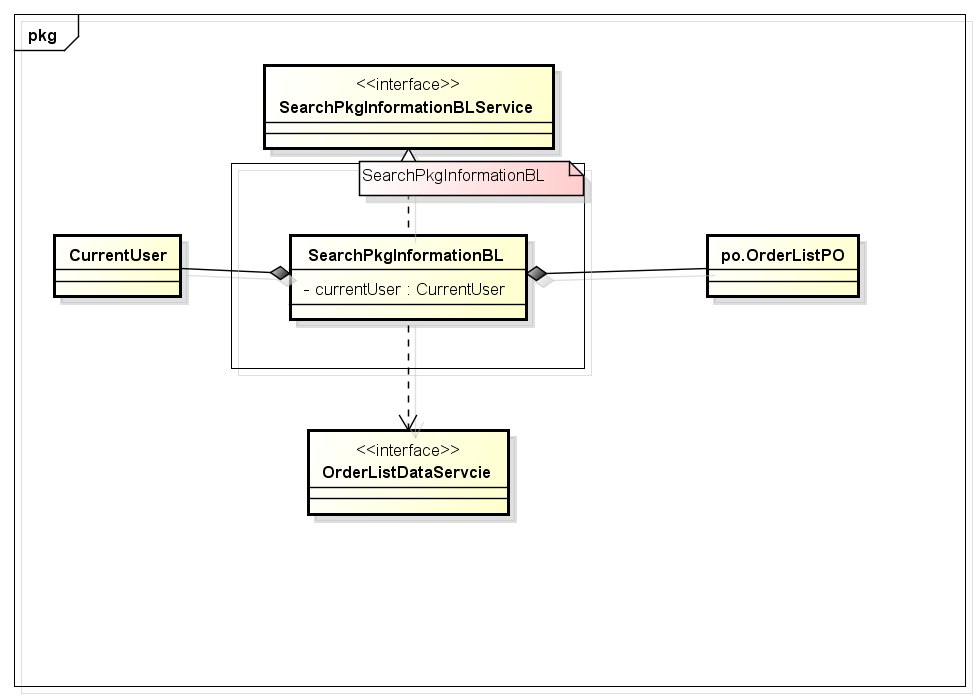
**（2）整体结构**

根据体系结构设计，我们在展示层和业务逻辑层之间添加了SearchPkgInformationBLService接口。在业务逻辑层和数据层之间添加了OrderListDataService接口。

SearchPkgInformationBL类的属性如图所示



SearchPkgInformationBL模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| SearchPkgInformationBL | 实现SearchPkgInformationBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| searchPkgInformation | **语法** | public OrderListVO searchPkgInformation(String orderlistId) |
| **前置条件** | Id输入合法 |
| **后置条件** | 返回一个订单的VO |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| OrderListDataService.add | 向系统中添加订单 | |
| OrderListDataService.delete | 系统中删除订单 | |
| OrderListDataService.update | 系统中修改订单 | |
| OrderListDataService.find | 系统中寻找订单 | |

**（4）模块的动态职责**

**4.2.2.16SendPkgBL模块**

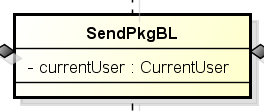
**（1）模块概述**

SendPkgBL模块承担了快递员寄件的功能需求，具体参见需求规格说明文档和相关非功能需求。

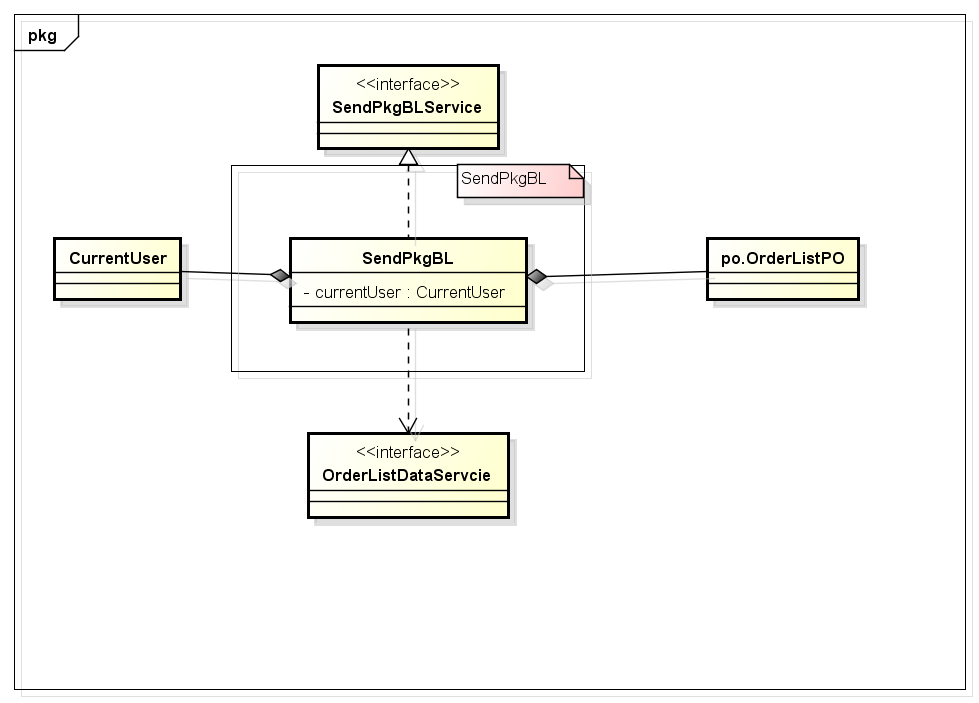
**（2）整体结构**

根据体系结构设计，我们在展示层和业务逻辑层之间添加了SendPkgBLService接口。在业务逻辑层和数据层之间添加了OrderListDataService接口。

SendPkgBL类的属性如图所示



SendPkgBL模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| SendPkgBL | 实现SendPkgBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **createMoneyAndDate** | **语法** | **publicOrderListVO createMoneyAndDate(OrderListVO orderListVO);** |
| **前置条件** | **输入参数** |
| **后置条件** | **返回包含计算的价格和到达日期的vo** |
| **createOrderList** | **语法** | **PublicResultMessage createOrderList(OrderListVO orderListVO);** |
| **前置条件** | **创建快递单** |
| **后置条件** | **返回操作是否成功** |
| **createOrderId** | **语法** | **public String createOrderId();** |
| **前置条件** | **要求得到快递单号** |
| **后置条件** | **系统返回响应的快递单号** |

**（4）模块的动态职责**

**4.2.2.17TransferCenterReceiveBL模块**

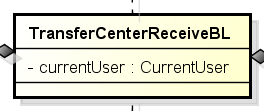
**（1）模块概述**

TransferCenterReceiveBL模块承担了中转中心接收的功能需求，具体参见需求规格说明文档和相关非功能需求。

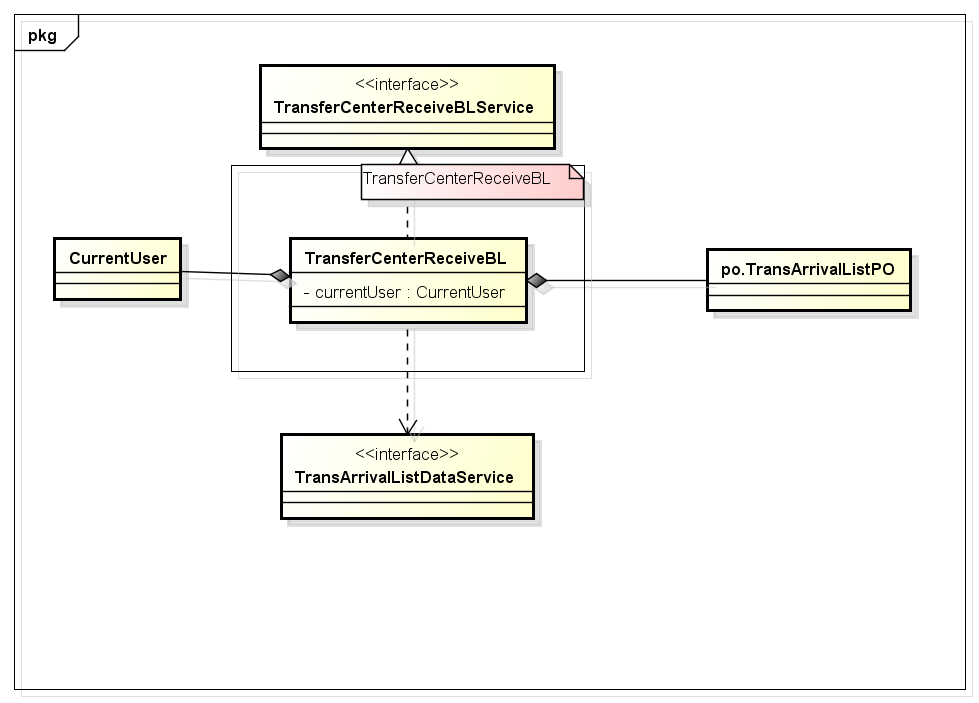
**（2）整体结构**

根据体系结构设计，我们在展示层和业务逻辑层之间添加了TransferCenterReceiveBLService接口。在业务逻辑层和数据层之间添加了TransArrivalListDataService接口。

TransferCenterReceiveBL类的属性如图所示



TransferCenterReceiveBL模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| TransferCenterReceiveBL | 实现TransferCenterReceiveBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| createTransArrivalList | **语法** | public ResultMessage createTransArrivalList(TransArrivalListVO transArrivalList) |
| **前置条件** | 创建中转中心到达单 |
| **后置条件** | 返回系统操作反馈 |
| createTransArrivalListId | **语法** | public String createTransArrivalListId() |
| **前置条件** | 要求得到中转中心到达单单号 |
| **后置条件** | 返回单号 |
| getLoadList | **语法** | public TransArrivalVO getLoadList(String loadlistid) |
| **前置条件** | 输入装车单id，得到所有的货物信息 |
| **后置条件** | 返回所有的货物信息 |
| getTransShipmentList | **语法** | public TransArrivalVO getTransShipmentList(String transshipmentlistid) |
| **前置条件** | 输入中转单id，得到所有的货物信息 |
| **后置条件** | 返回所有的货物信息 |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| OrderListDataService.add | 向系统中添加订单 | |
| OrderListDataService.delete | 系统中删除订单 | |
| OrderListDataService.update | 系统中修改订单 | |
| OrderListDataService.find | 系统中寻找订单 | |

**（4）模块的动态职责**

**4.2.2.18TransShipmentBL模块**

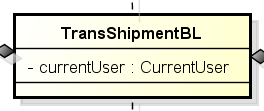
**（1）模块概述**

TransShipmentBL模块承担了中转中心装运的功能需求，具体参见需求规格说明文档和相关非功能需求。

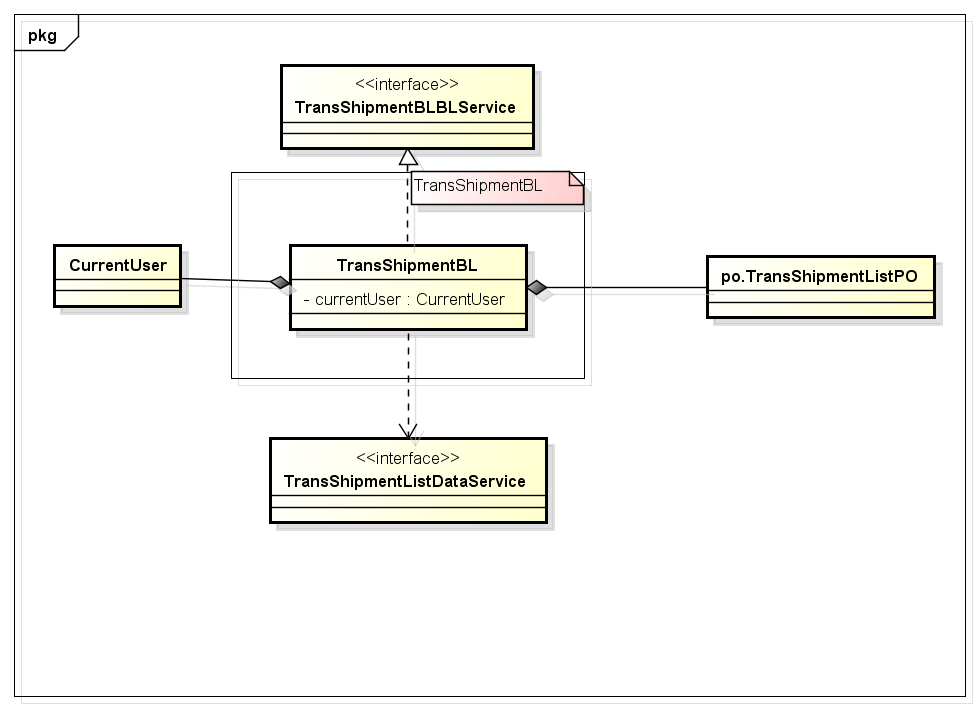
**（2）整体结构**

根据体系结构设计，我们在展示层和业务逻辑层之间添加了TransShipmentBLService接口。在业务逻辑层和数据层之间添加了TransShipmentListDataService接口。

TransShipmentBL类的属性如图所示



TransShipmentBL模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| TransShipmentBL | 实现TransShipmentBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| createShiplist | **语法** | public ResultMessage createShiplist (TransShipmentListVO transShipment); |
| **前置条件** | 开启了装运任务 |
| **后置条件** | 系统判定输入准确无误之后，保存装运单，修改订单的相关属性 |
| createTransShipmentListId | **语法** | public String createTransShipmentListId() |
| **前置条件** | 要求得到中转单id |
| **后置条件** | 系统返回相应的中转单id |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| TransShipmentListDataService.add | 向系统中添加装运单 | |
| TransShipmentListDataService.delete | 系统中删除装运单 | |
| TransShipmentListDataService.update | 系统中修改装运单 | |
| TransShipmentListDataService.find | 系统中寻找装运单 | |

**（4）模块的动态职责**

**4.2.2.19** **TruckManageBL模块**

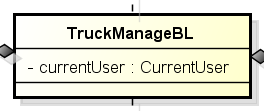
**（1）模块概述**

TruckManageBL模块承担了营业厅司机信息管理的功能需求，具体参见需求规格说明文档和相关非功能需求。

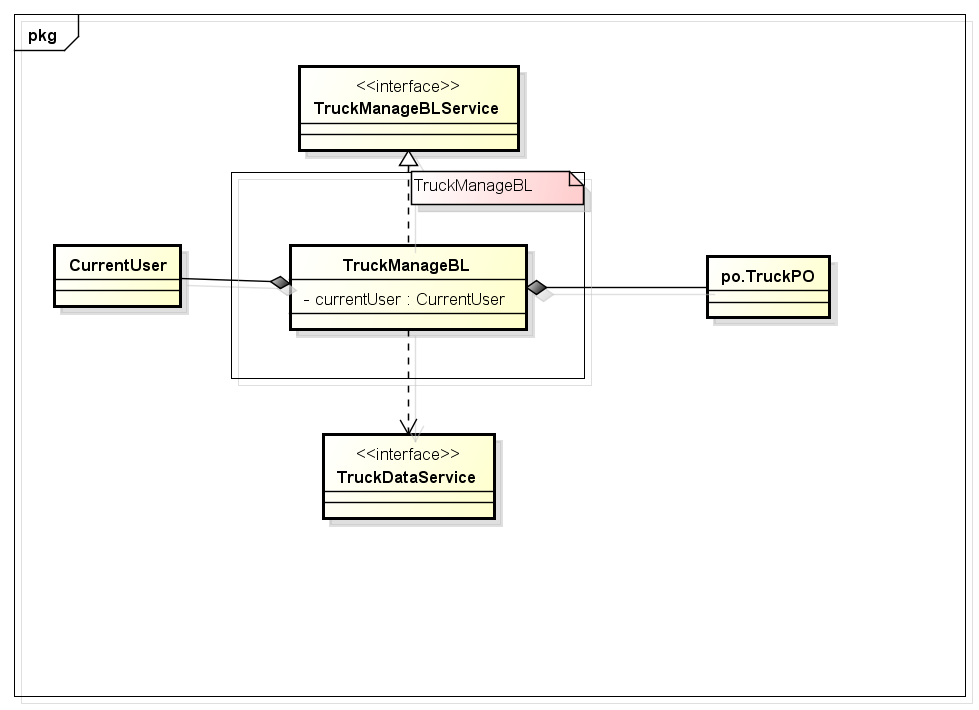
**（2）整体结构**

根据体系结构设计，我们在展示层和业务逻辑层之间添加了TruckManageBL接口。在业务逻辑层和数据层之间添加了TruckDataService接口。

TruckManageBL类的属性如图所示



TruckManageBL模块的设计如图所示



模块内各个类的职责如下表所示

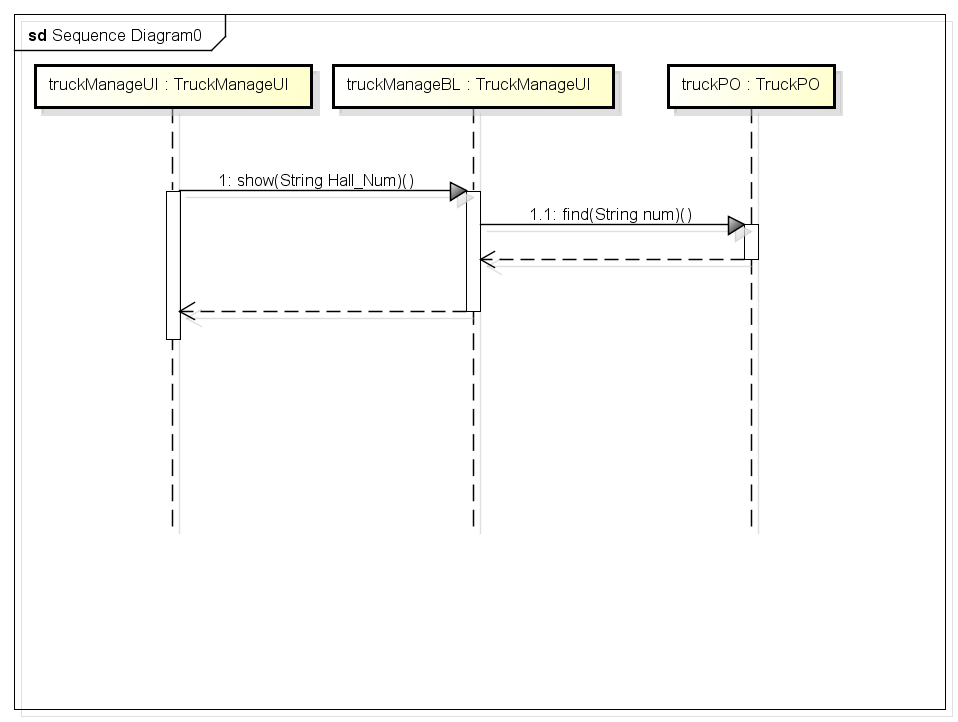
|  |  |
| --- | --- |
| 模块 | 职责 |
| TruckManageBL | 实现TruckManageBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| show | **语法** | public ArrayList<TruckVO> show(String Hall\_Num); |
| **前置条件** | Name输入合法存在 |
| **后置条件** | 查找该营业厅全部list返回VO |
| update | **语法** | public ResultMessage update(TruckVO truck); |
| **前置条件** | VO修改信息合理 |
| **后置条件** | 更新一个VO |
| delete | **语法** | public ResultMessage delete(TruckVO truck) |
| **前置条件** | VO输入格式正确 |
| **后置条件** | 删除一个货车信息 |
| add | **语法** | public ResultMessage add(TruckVO truck) |
| **前置条件** | VO格式输入正确 |
| **后置条件** | 新建一个VO |
| find | **语法** | public TruckVO find(String num) |
| **前置条件** | Num格式输入正确 |
| **后置条件** | 查找是否存在并返回一个VO |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| TruckDataService.add | 向系统中添加一个货车 | |
| TruckDataService.delete | 系统中删除一个货车 | |
| TruckDataService.update | 系统中修改一个货车 | |
| TruckDataService.find | 系统中寻找一个货车 | |

**（4）模块的动态职责**

下图表示在物流信息管理系统中，营业厅管理员选择显示本营业厅所有货车时，货车管理业务逻辑处理的相关对象之间的协作。

****

**4.2.2.20UserManagebl模块**

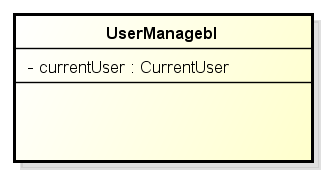
**（1）模块概述**

UserManagebl模块承担用户管理的功能需求，具体可参见需求规格说明文档功能需求和非功能需求。

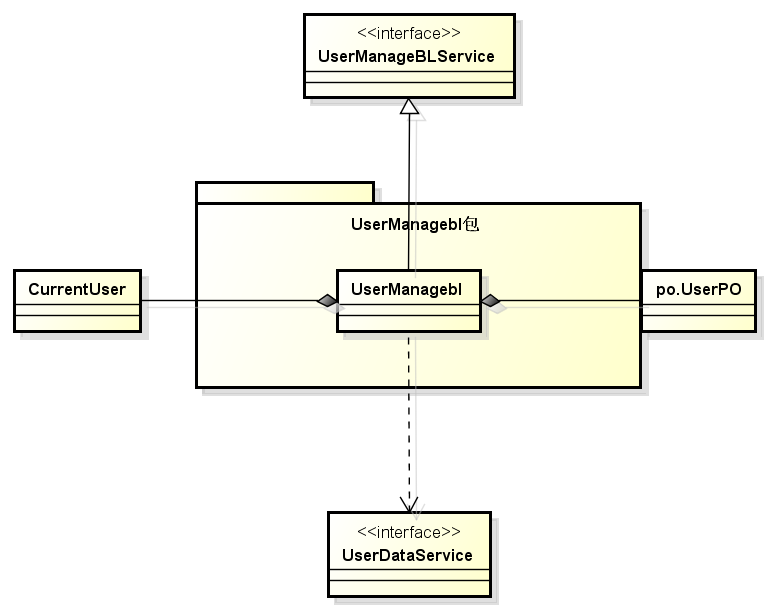
**（2）整体结构**

根据体系结构的设计，我们在展示层和业务逻辑层之间添加了UserManageblService接口。在逻辑层和数据层之间添加了UserDataService接口。

UserManagebl类的属性如下图所示



UserManagebl模块的设计如图所示



模块内各个类的职责如下表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| UserManagebl | 实现UserManageBLService的各个方法 |
| CurrentUser | 创建类时必须确认的使用者对象 |

**（3）模块内部类的接口规范**

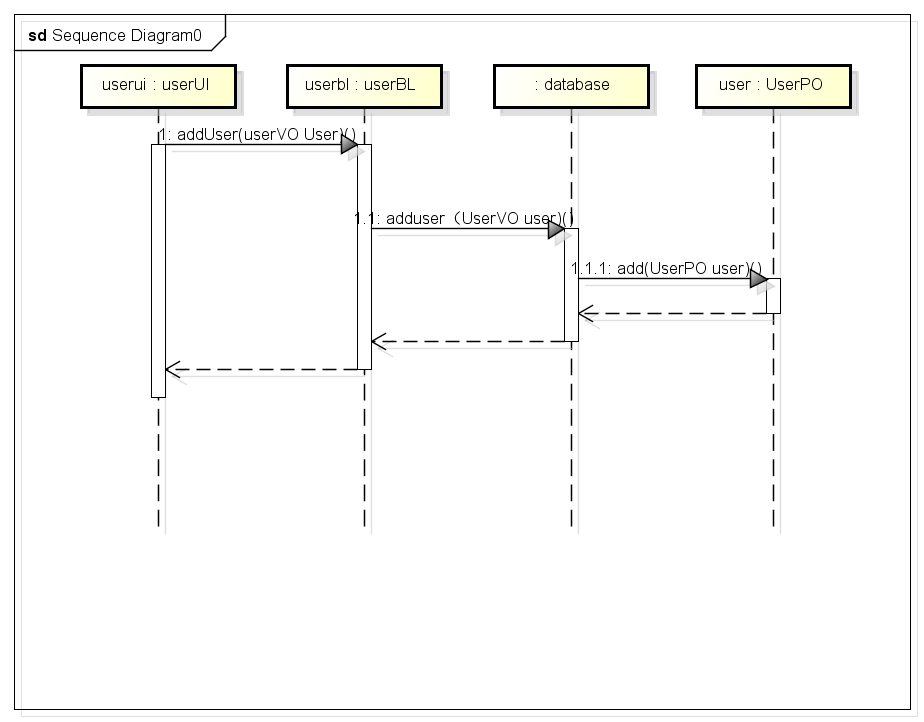
UserManagebl接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| UserManagebl.login | 语法 | public ResultMessage login(String admin,String password) |
| 前置条件 | 输入用户名和密码 |
| 后置条件 | 返回登录信息 |
| UserManagebl.addUser | 语法 | public ResultMessage addUser(UserVO user) |
| 前置条件 | 输入新员工信息 |
| 后置条件 | 返回新员工信息 |
| UserManagebl.removeUser | 语法 | public ResultMessage removeUser(String admin) |
| 前置条件 | 删除员工信息 |
| 后置条件 | 返回操作反馈 |
| UserManagebl.updateUser | 语法 | public ResultMessage updateUser(UserVO user) |
| 前置条件 | 更改员工信息 |
| 后置条件 | 返回操作反馈 |
| UserManagebl.showAll | 语法 | public ArrayList<UserVO> showAll() |
| 前置条件 | 请求查看所有员工信息 |
| 后置条件 | 返回所有员工信息 |
| UserManagebl.finonAdmin | 语法 | public UserVO findonAdmin(String admin) |
| 前置条件 | 输入员工的账号 |
| 后置条件 | 返回员工信息 |
| UserManagebl.findonId | 语法 | public UserVO findonId(String id) |
| 前置条件 | 输入员工的编号 |
| 后置条件 | 返回员工信息 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| UserDataService. add(UserPO user) | 插入单一持久化对象 |
| UserDataService. delete(UserPO user) | 删除单一持久化对象 |
| UserDataService. update(UserPO user) | 更新单一持久化对象 |
| UserDataService. findonId(String id) | 根据ID进行查找单一持久化对象 |
| UserDataService. findonAdmin(String admin) | 根据账号进行查找单一持久化对象 |
| UserDataService. showAll() | 显示多个持久化对象 |

**（4）模块间的动态职责**

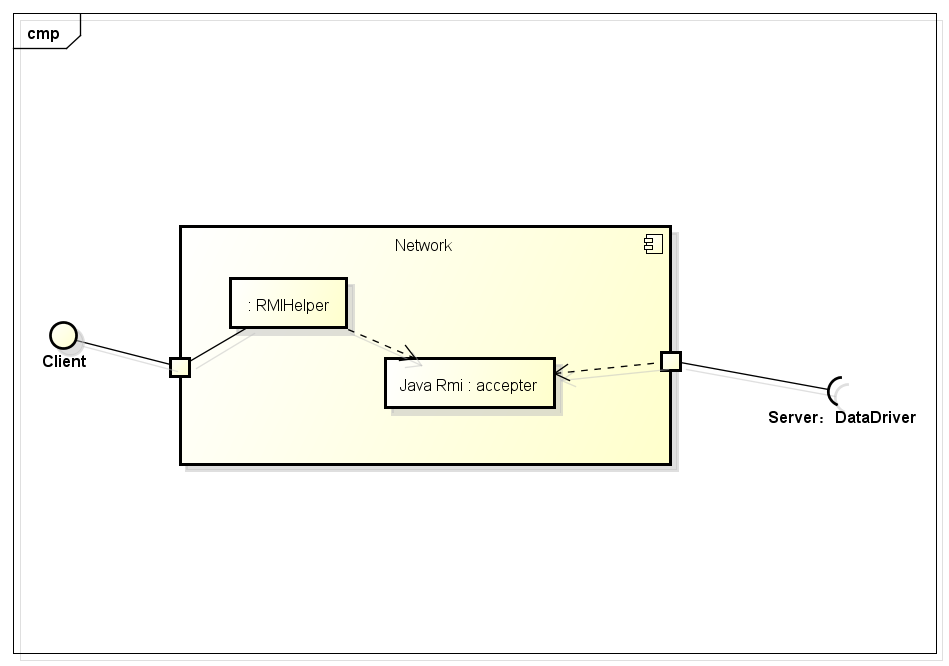
下图表示在物流信息管理系统中，当管理员输入新的账户信息请求新建账户时，业务逻辑的相关对象之间的协作。



**4.3客户端网络模块的静态结构和动态职责**

由于数据是存储在服务器上的，客户端访问数据必须通过网络通信层，具体是由客户端的业务逻辑层和服务器端的数据层通过网络通信层进行通信。如图1所示。

具体网络通信层的实现是通过远程方法调用RMI来实现的。

****

**图1客户端数据通信层模块的描述**

**4.3.1客户端网络模块局部模块的职责**

其实网络通信层的实际运作代码是由RMI API所完成的，在客户端我们只需要部署RMI服务的RMIHelper文件和Java RMI类库即可。如表所示。

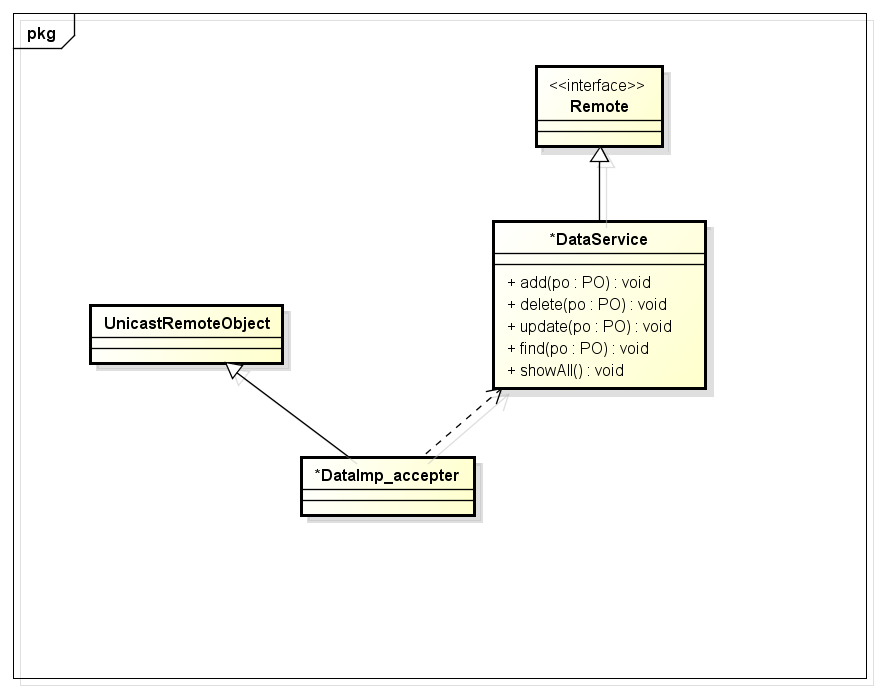
**表33 网络通信层模块的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| RMIHelper | 给业务逻辑层提供RMI服务接口 |
| Java RMI 类库 | 实现RMI机制 |

**4.4服务器端网络模块的静态结构和动态职责**

**4.4.1服务器端网络模块局部模块的职责**

由于数据是存储在服务器上的，客户端访问数据必须通过网络通信层，具体是由客户端的业务逻辑层和服务器端的数据层通过网络通信层进行通信。

具体服务器端模块中只需要包含Java RMI类库即可。但是其数据层实现RMI的类需要符合以下条件：

**图2 服务器数据层实现RMI的类的描述**

**4.4.2服务器端网络模块局部模块的接口规范**

\*DataService 是被业务逻辑层调用的接口。为了实现RMI， \*DataService需要继承 Remote接口，而数据层中的实现这个接口的类\*DataImp\_accepterl都需要继承UnicastRemoteObject类。

以下是服务器网络模块的RMI详细的规范：

**表34服务器网络模块的RMI接口规范**

|  |  |
| --- | --- |
| **类/接口** | **规范** |
| 远程服务接口 | extends Remote，方法 throws RemoteException |
| 远程服务的实现类 | extendsUnicastRemoteObject |

**表35服务器网络模块的RMI接口返回值的规范**

|  |  |
| --- | --- |
| **类/接口** | **规范** |
| 作为远程引用传递 | 该类需要extends Remote，方法 throws RemoteException |
| 通过数据库进行实现 | Implements MySql |

**4.5服务器端数据层模块的静态结构和动态职责**

**4.5.1服务器端数据层模块局部模块的职责**

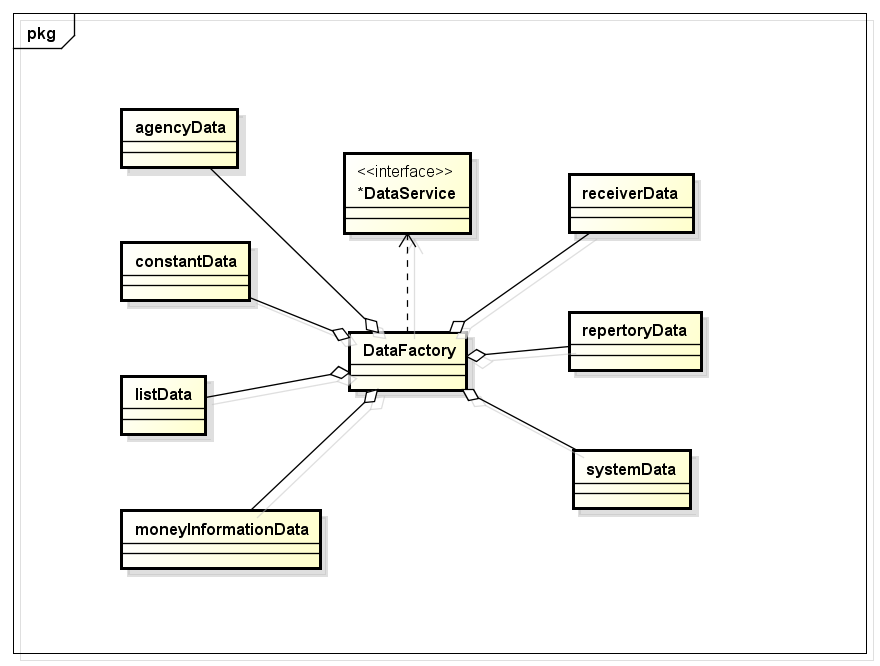
服务器端数据层主要负责数据的持久化及数据访问接口。对于持久化数据的集体载入、集体保存和增、删、改、查。这些服务由DataService接口提供。本系统（物流管理系统）程序的持久化数据的保存以数据库的形式实现。具体如图3所示。

图3 数据层模块的描述

**表36 数据层模块的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| DataFactory | 持久化数据工厂，提供获得不同数据的工厂方法 |
| DataService | 持久化文件的接口，提供集体载入、集体保存、增、删、改、查服务。 |

**4.5.2服务器端数据层模块局部模块的接口规范**

**4.5.2.1 agencyData模块接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| BankAccountDataService.add | 语法 | public boolean add(BankAccountPO countpo)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 添加账户信息 |
| BankAccountDataService.delete | 语法 | public boolean delete(String name)throws RemoteException; |
| 前置条件 | String输入正确 |
| 后置条件 | 删除账户信息 |
| BankAccountDataService.updata | 语法 | public boolean updata(BankAccountPO countpo)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 修改账户信息 |
| BankAccountDataService.find | 语法 | public BankAccountPO find(String keyword)throws RemoteException; |
| 前置条件 | 关键词输入正确 |
| 后置条件 | 根据关键字返回账户信息 |
| BankAccountDataService.showAll | 语法 | public ArrayList<BankAccountPO> showAll()throws RemoteException; |
| 前置条件 | \ |
| 后置条件 | 显示所有账户信息 |
| DriverDataService.add | 语法 | public boolean add(DriverPO driver) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 增加司机信息 |
| DriverDataService.delete | 语法 | public boolean delete(DriverPO driver) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除司机信息 |
| DriverDataService.update | 语法 | public boolean update(DriverPO driver) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新司机信息 |
| DriverDataService.find | 语法 | public DriverPO find(String num) throws RemoteException; |
| 前置条件 | Num格式输入正确 |
| 后置条件 | 找寻司机信息 |
| DriverDataService.showAll | 语法 | public ArrayList<DriverPO> showAll(String Hall\_Num) throws RemoteException; |
| 前置条件 | 营业厅格式输入正确 |
| 后置条件 | 返回营业厅内司机信息PO |
| PeopleAgencyDataService.salaryManage | 语法 | public void salaryManage(String userID,String salary); |
| 前置条件 | UserID与salary格式正确 |
| 后置条件 | 修改一个PO |
| PeopleAgencyDataService.add | 语法 | public void add(AgencyPO agency) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| PeopleAgencyDataService.delete | 语法 | public void delete(AgencyPO agency) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| PeopleAgencyDataService.update | 语法 | public void update(AgencyPO agency) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 修改一个PO |
| PeopleAgencyDataService.find | 语法 | public AgencyPO find(String id) throws RemoteException; |
| 前置条件 | 进货销售人员被成功识别和授权 |
| 后置条件 | 返回一个PO |
| PeopleAgencyDataService.showAll | 语法 | public ArrayList<AgencyPO> showAll() throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 返回所有PO的集合 |
| TruckDataService.add | 语法 | public boolean add(TruckPO driver) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 增加车辆信息 |
| TruckDataService.delete | 语法 | public boolean delete(TruckPO driver) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除车辆信息 |
| TruckDataService.update | 语法 | public boolean update(TruckPO driver) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新车辆信息 |
| TruckDataService.find | 语法 | public TruckPO find(String num) throws RemoteException; |
| 前置条件 | Num格式输入正确 |
| 后置条件 | 找寻车辆信息 |
| TruckDataService.showAll | 语法 | public ArrayList<TruckPO > showAll(String Hall\_Num) throws RemoteException; |
| 前置条件 | 营业厅格式输入正确 |
| 后置条件 | 返回营业厅内所有车辆信息PO |

**4.5.2.2 constantInfo模块接口规范**

|  |  |  |
| --- | --- | --- |
| ConstantDataService.setDistance | 语法 | public boolean setDistance(String newDistance,String cityA,String cityB)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 修改城市距离信息 |
| ConstantDataService.setPrice | 语法 | public boolean setPrice(String newPrice,String otherInformation)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 修改价格常量 |
| ConstantDataService.addCity | 语法 | public boolean addCity(String name)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 新增城市信息 |
| ConstantDataService.showDistanceChart | 语法 | public DistanceChartPO showDistanceChart()throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 显示城市距离表 |
| ConstantDataService.showPriceChart | 语法 | public PriceChartPO showPriceChart()throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 显示价格常量表 |
| ConstantDataService.showAll | 语法 | public ArrayList<TruckPO> showAll(String Hall\_Num); |
| 前置条件 | Hall——num输入正确 |
| 后置条件 | 显示所有司机PO |

**4.5.2.3 receiver模块接口规范**

|  |  |  |
| --- | --- | --- |
| ReceiverDataService. | 语法 | public boolean add(ReceiverPO receiver)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| ReceiverDataService. | 语法 | public ReceiverPO findOnName(String name)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| ReceiverDataService. | 语法 | public ReceiverPO findOnTime(String time)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |

**4.5.2.4 system模块接口规范**

|  |  |  |
| --- | --- | --- |
| SystemLogDataService.showAll | 语法 | public ArrayList<SystemLogPO> showAll(String time1,String time2) throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO列表 |
| SystemLogDataService.add | 语法 | public void add(SystemLogPO systemLog) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |

|  |  |  |
| --- | --- | --- |
| UserDataService.add | 语法 | public void add(UserPO user) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| UserDataService.delete | 语法 | public void delete(UserPO user) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| UserDataService.update | 语法 | public void update(UserPO user) throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新一个PO |
| UserDataService.findonId | 语法 | public UserPO findonId(String id) throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| UserDataService.findonAdmin | 语法 | public UserPO findonAdmin(String admin) throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| UserDataService.showAll | 语法 | public ArrayList<UserPO> showAll() throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 显示所有PO |

**4.5.2.5 repertory模块接口规范**

|  |  |  |
| --- | --- | --- |
| RepertoryInDataService.add | 语法 | public boolean add(RepertoryInPO repertoryIn)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| RepertoryInDataService.delete | 语法 | public boolean delete(RepertoryInPO repertoryIn)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| RepertoryInDataService.update | 语法 | public boolean update(RepertoryInPO repertoryIn)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新一个PO |
| RepertoryInDataService.findOnID | 语法 | public RepertoryInPO findOnID(String id)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| RepertoryInDataService.findOnTime | 语法 | public ArrayList<RepertoryInPO> findOnTime(String time)throws RemoteExcept |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| RepertoryInDataService.showAll | 语法 | public ArrayList<RepertoryInPO> showAll(String start\_day, String end\_day) throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 显示所有PO |
| RepertoryOutDataService.add | 语法 | public boolean add(RepertoryOutPO repertoryOut)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| RepertoryOutDataService.delete | 语法 | public boolean delete(RepertoryOutPOrepertoryout)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| RepertoryOutDataService.update | 语法 | public boolean update(RepertoryOutPOrepertoryout)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新一个PO |
| RepertoryOutDataService.findOnID | 语法 | public RepertoryOutPO findOnID(String id)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| RepertoryOutDataService.findOnTime | 语法 | public ArrayList<RepertoryOutPO> findOnTime(String time)throws RemoteExcept |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| RepertoryOutDataService.showAll | 语法 | public ArrayList<RepertoryOutPO> showAll(String start\_day, String end\_day) throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 显示所有PO |
| RepertoryInfoDataService.add | 语法 | public boolean add(GoodsInfoPO goods,RepertoryInPO repertoryPO); |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 增加一个PO |
| RepertoryInfoDataService.delete | 语法 | public boolean delete(GoodsInfoPO goods,RepertoryOutPO repertoryPO); |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| RepertoryInfoDataService.show | 语法 | public RepertoryInfoPO show(); |
| 前置条件 | 无 |
| 后置条件 | 返回一个PO |

**4.5.2.6 moneyInformation模块接口规范**

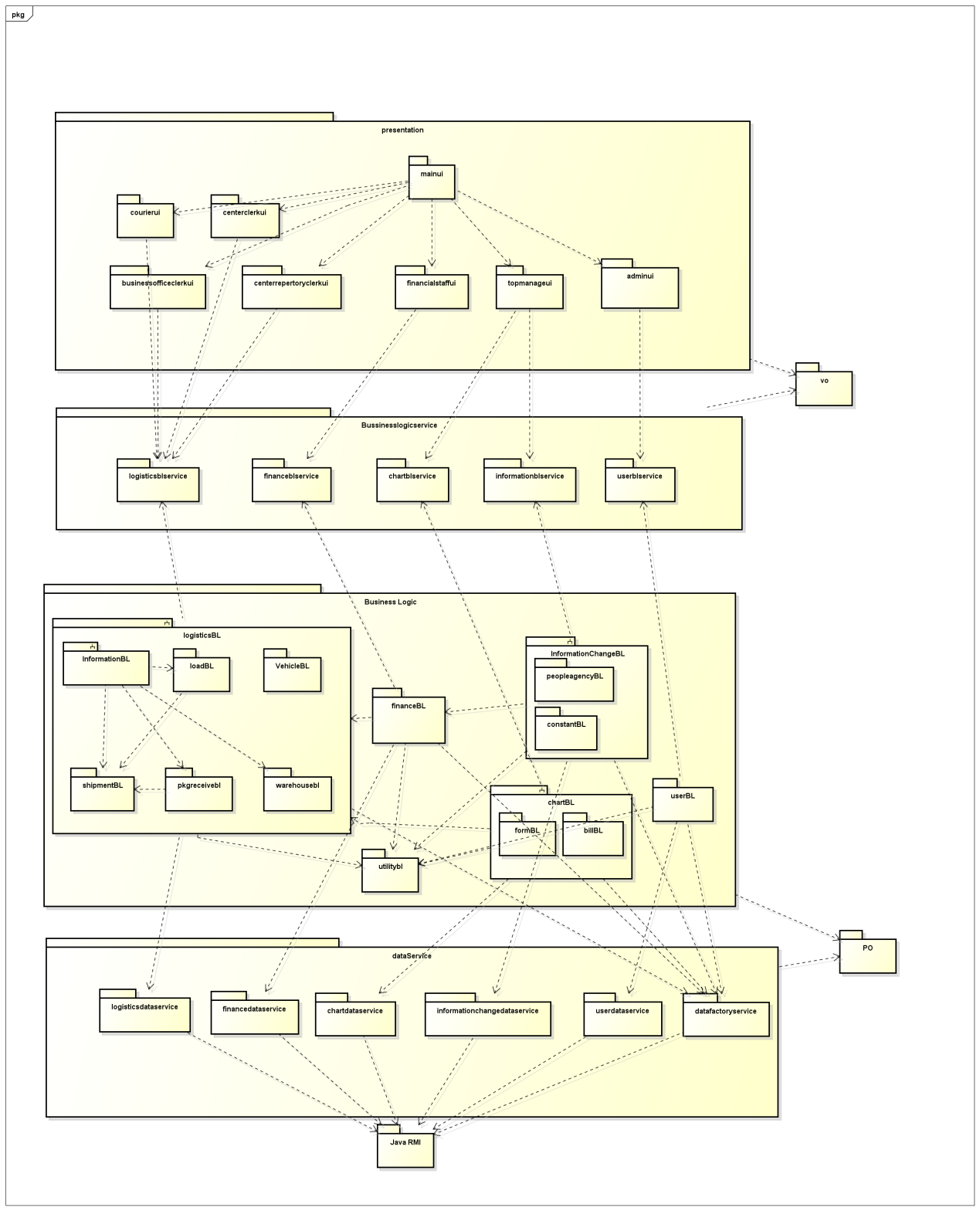
|  |  |  |
| --- | --- | --- |
| BaseDataSettingDataService.addInstitutionInfo | 语法 | public boolean addInstitutionInfo(AgencyPO agencyPO); |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| BaseDataSettingDataService.show | 语法 | public BaseDataSettingPO show(); |
| 前置条件 | 无 |
| 后置条件 | 返回期初信息PO |
| BaseDataSettingDataService.addCommodityInfo | 语法 | public boolean addCommodityInfo(RepertoryInfoPO repertoryPO); |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| BaseDataSettingDataService.addTruckInfo | 语法 | public boolean addTruckInfo(TruckPO truck); |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| BaseDataSettingDataService.addFinanceInfo | 语法 | public boolean addFinanceInfo(BankAccountPO bankAccountPO); |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| MoneyInListDataService.add | 语法 | public boolean add(MoneyInListPO moneyInList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| MoneyInListDataService.delete | 语法 | public boolean delete(MoneyInListPO moneyInList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| MoneyInListDataService.update | 语法 | public boolean update(MoneyInListPO moneyInList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新一个PO |
| MoneyInListDataService.findOnID | 语法 | public MoneyInListPO findOnID(String id)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| MoneyInListDataService.findOnTime | 语法 | public MoneyInListPO findOnTime(String time)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| MoneyInListDataService.showAll | 语法 | public ArrayList<MoneyInListPO> showAll(String start\_day, String end\_day) throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 显示所有PO |
| MoneyOutListDataService.add | 语法 | public boolean add(MoneyOutListPO moneyOutList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| MoneyOutListDataService.delete | 语法 | public boolean delete(MoneyOutListPO moneyOutList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| MoneyOutListDataService.update | 语法 | public boolean update(MoneyOutListPO moneyoutList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新一个PO |
| MoneyOutListDataService.findOnID | 语法 | public MoneyOutListPO findOnID(String id)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| MoneyOutListDataService.findOnTime | 语法 | public MoneyOutListPO findOnTime(String time)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| MoneyOutListDataService.showAll | 语法 | public ArrayList<MoneyOutListPO > showAll(String start\_day, String end\_day) throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 显示所有PO |

**4.5.2.7 list模块接口规范**

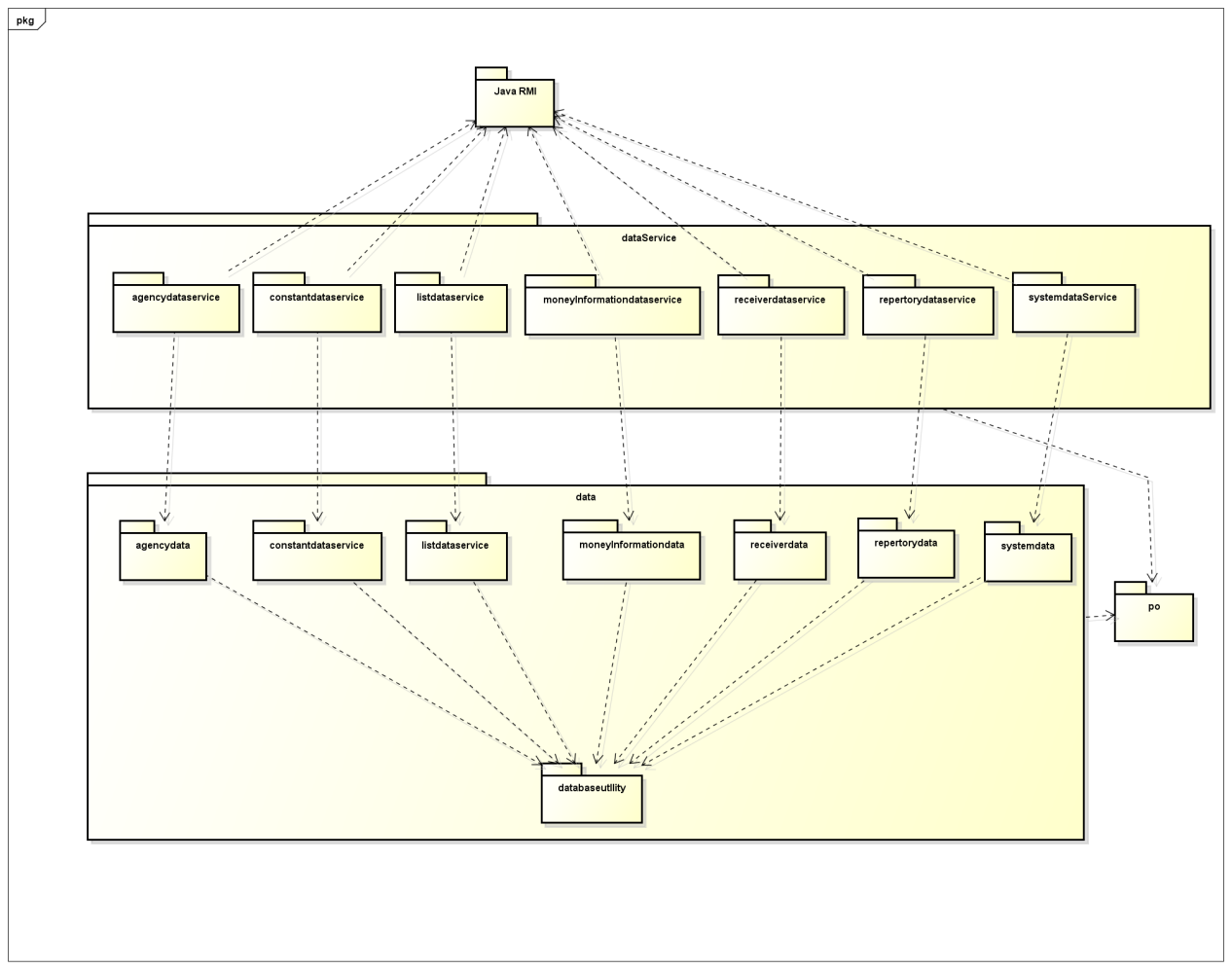
|  |  |  |
| --- | --- | --- |
| DeliveringListDataService.add | 语法 | public boolean add (DeliveringListPO deliveringList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| DeliveringListDataService.delete | 语法 | public boolean delete (DeliveringListPO deliveringList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| DeliveringListDataService.update | 语法 | public boolean update (DeliveringListPO deliveringList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新一个PO |
| DeliveringListDataService.find | 语法 | public DeliveringListPO find(String id)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| DeliveringListDataService.showAll | 语法 | public ArrayList<DeliveringListPO> showAll(String time1,String time2)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO列表 |
| HallArrivalListDataService.add | 语法 | public boolean add (HallArrivalListPO HallArrivalList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| HallArrivalListDataService.delete | 语法 | public boolean delete (HallArrivalListPO HallArrivalList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| HallArrivalListDataService.update | 语法 | public boolean update (HallArrivalListPO HallArrivalList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新一个PO |
| HallArrivalListDataService.find | 语法 | public HallArrivalListPO find(String id)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| HallArrivalListDataService.showAll | 语法 | public ArrayList<HallArrivalListPO > showAll(String time1,String time2)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO列表 |
| LoadListDataService.add | 语法 | public boolean add (LoadListPO LoadList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| LoadListDataService.delete | 语法 | public boolean delete (LoadListPO LoadList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| LoadListDataService.update | 语法 | public boolean update (LoadListPO LoadList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新一个PO |
| LoadListDataService.find | 语法 | public LoadListPO find(String id)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| LoadListDataService.showAll | 语法 | public ArrayList<LoadListPO > showAll(String time1,String time2)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO列表 |
| OrderListDataService.add | 语法 | public boolean add (OrderListPO orderList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| OrderListDataService.delete | 语法 | public boolean delete (OrderListPO orderList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| OrderListDataService.update | 语法 | public boolean update (OrderListPO orderList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新一个PO |
| OrderListDataService.find | 语法 | public OrderListPO find(String id)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| OrderListDataService.showAll | 语法 | public ArrayList<OrderListPO > showAll(String time1,String time2)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO列表 |
| TransArrivalListDataService.add | 语法 | public boolean add (TransArrivalListPO transArrivalList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| TransArrivalListDataService.delete | 语法 | public boolean delete (TransArrivalListPO transArrivalList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| TransArrivalListDataService.update | 语法 | public boolean update (TransArrivalListPO transArrivalList)throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新一个PO |
| TransArrivalListDataService.find | 语法 | public TransArrivalListPO find(String id)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| TransArrivalListDataService.showAll | 语法 | public ArrayList<TransArrivalListPO > showAll(String time1,String time2)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO列表 |
| TransShipmentListDataService.add | 语法 | public boolean add (TransShipmentListPO transShipmentList )throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 新增一个PO |
| TransShipmentListDataService.delete | 语法 | public boolean delete (TransShipmentListPO transShipmentList )throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 删除一个PO |
| TransShipmentListDataService.update | 语法 | public boolean update (TransShipmentListPO transShipmentList )throws RemoteException; |
| 前置条件 | PO格式输入正确 |
| 后置条件 | 更新一个PO |
| TransShipmentListDataService.find | 语法 | public TransShipmentListPO find(String id)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO |
| TransShipmentListDataService.showAll | 语法 | public ArrayList<TransShipmentListPO > showAll(String time1,String time2)throws RemoteException; |
| 前置条件 | String格式输入正确 |
| 后置条件 | 返回一个PO列表 |

**5.依赖视角**

下图分别是客户端和服务器端各自包之间的依赖关系



客户端包图



服务器端包图