|  |
| --- |
|  |
| 物流管理系统 |
| 软件体系结构描述文档 |
|  |
| **版本：v3.0 FinalVersion** |
| **2016/1/3** |

|  |
| --- |
|  |

**小组成员： 汤大业**

**谭期友**

**杨三洋**

**王家玮**

**目录及索引**

[1.引言 3）](#引言)

[1.1编制目的](#编制目的)

[1.2参考文献](#参考资料)

[2.产品概述 3）](#产品概述)

[3.逻辑视角 3）](#逻辑视角)

[4.组合视角 4）](#组合视角)

[4.1开发包图](#开发包图)

[4.2运行时进程](#运行时进程)

[4.3物理部署](#物理部署)

[5.接口视角 9）](#接口视角)

[5.1模块的职责](#模块的职责)

[5.2 用户界面分层解](#用户界面的分解)

[5.3业务逻辑层的分解](#业务逻辑层的分解)

[5.4数据层的分解](#数据层的分解)

[6.信息视角 53）](#信息视角)

[6.1数据持久化对象](#数据持久化对象)

[6.2 Txt持久化格式](#Txt持久化格式)

**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概要设计，达到指导详细设计和开发的的目的，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

**1.2参考文献**

1. IEE标准

2.项目实践V4 南京大学软件学院 刘钦 2015

3.软件工程与计算Ⅱ 南京大学软件学院 刘钦 丁二玉 2015

4.物流管理系统用例文档V1.0

5.物流管理系统软件需求规格说明

**2.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中对产品的概括描述

**3.逻辑视角**

物流信息管理系统中，选择了分层体系结构风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好地示意整个高层抽象。展示层包含GUI页面的实现，业务逻辑层包含业务逻辑处理的实现， 数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如下图1和图2所示。

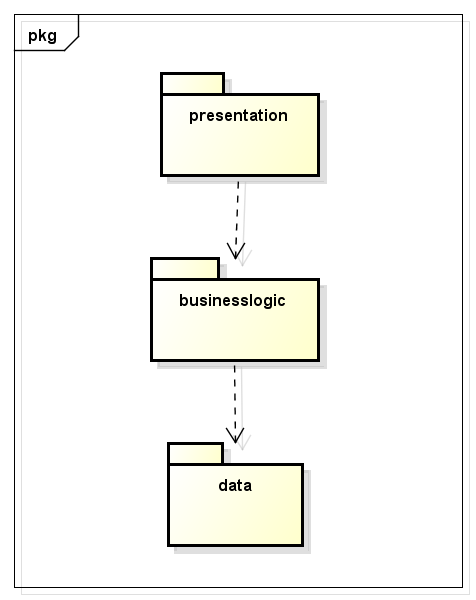
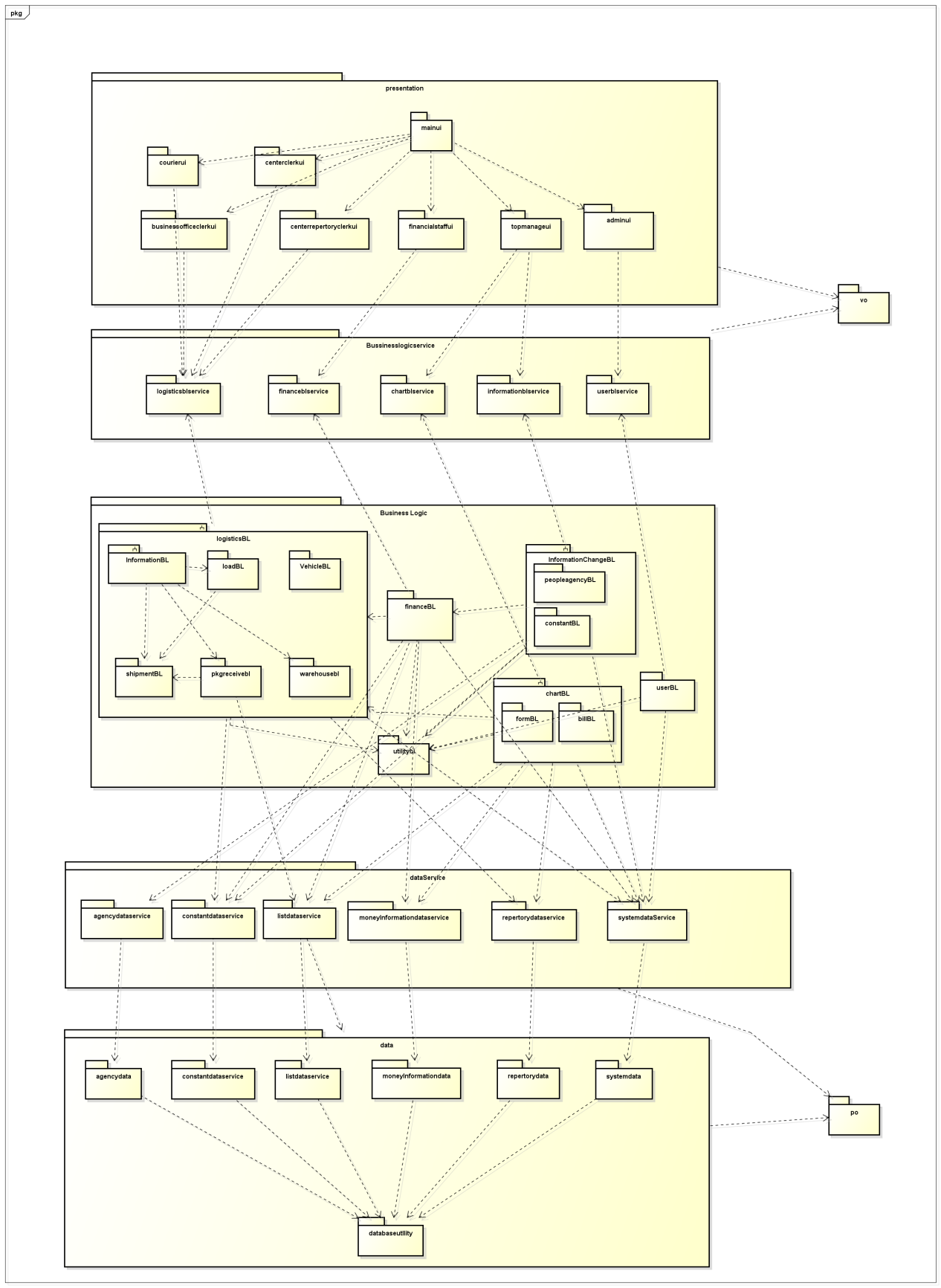


图1 参照体系结构风格的包图表达逻辑视角

图2 软件体系结构逻辑设计方案

**4.组合视角**

**4.1开发包图**

物流管理系统的最终开发包设计如表1所示。

**表1 物流管理系统的最终开发包设计**

|  |  |
| --- | --- |
| 开发（物理）包 | 依赖的其他开发包 |
| Mainui | courierui,topmanageui,finacialstaffui,centerclerkUI,businessclerkui,userui,vo |
| businessclerkui | Logisticsbl,界面类库包,vo |
| courierui | Logisticsbl,界面类库包,vo |
| topmanageui | Logisticsbl,fincialbl,chartbl,systembl,界面类库包 |
| Logisticsservice |  |
| Logisticsbl | Logisticsservice, logisticsdataservice,po |
| Logisticsdataservice | Java RMI,po |
| Logisticsdata | Databaseutility,po, logisticsdataservice |
| Financestaffui | Financeblservice,界面类库包 |
| Financeservice |  |
| Financebl | Financeblservice,financedataservice,po,logisticsbl |
| Financedataservice | Java RMI,po |
| Financedata | Java RMI,po,databaseutility |
| userbl | Userbl,界面类库包 |
| Chartservice |  |
| Chartbl | Chartblservice,chartdataservice,po,logisticbl |
| Chartdataservice | Java RMI,po |
| Chartdata | Java RMI,po,databaseutility |
| Informationchangeui | Informationchangebl,界面类库包 |
| Informationchangeservice |  |
| Informationchangebl | Informationchangeblservice,informationchangedataservice,po,financebl |
| Informationchangedataservice | Java RMI,po |
| Informationchangedata | Java RMI,po,databaseutility |
| Userui | Userblservice，界面类库包 |
| Userservice |  |
| Userbl | UserInterface,UserDataClient,UserPo |
| Userdataservice | Java RMI,po |
| Userdata | Java RMI,po,databaseutility |
| Vo |  |
| Po |  |
| Utilitybl |  |
| 界面类库包 |  |
| Java RMI |  |
| Databaseutility | JDBC |

物流管理系统客户端开发包图如图3所示，服务器端开发包图如图4所示

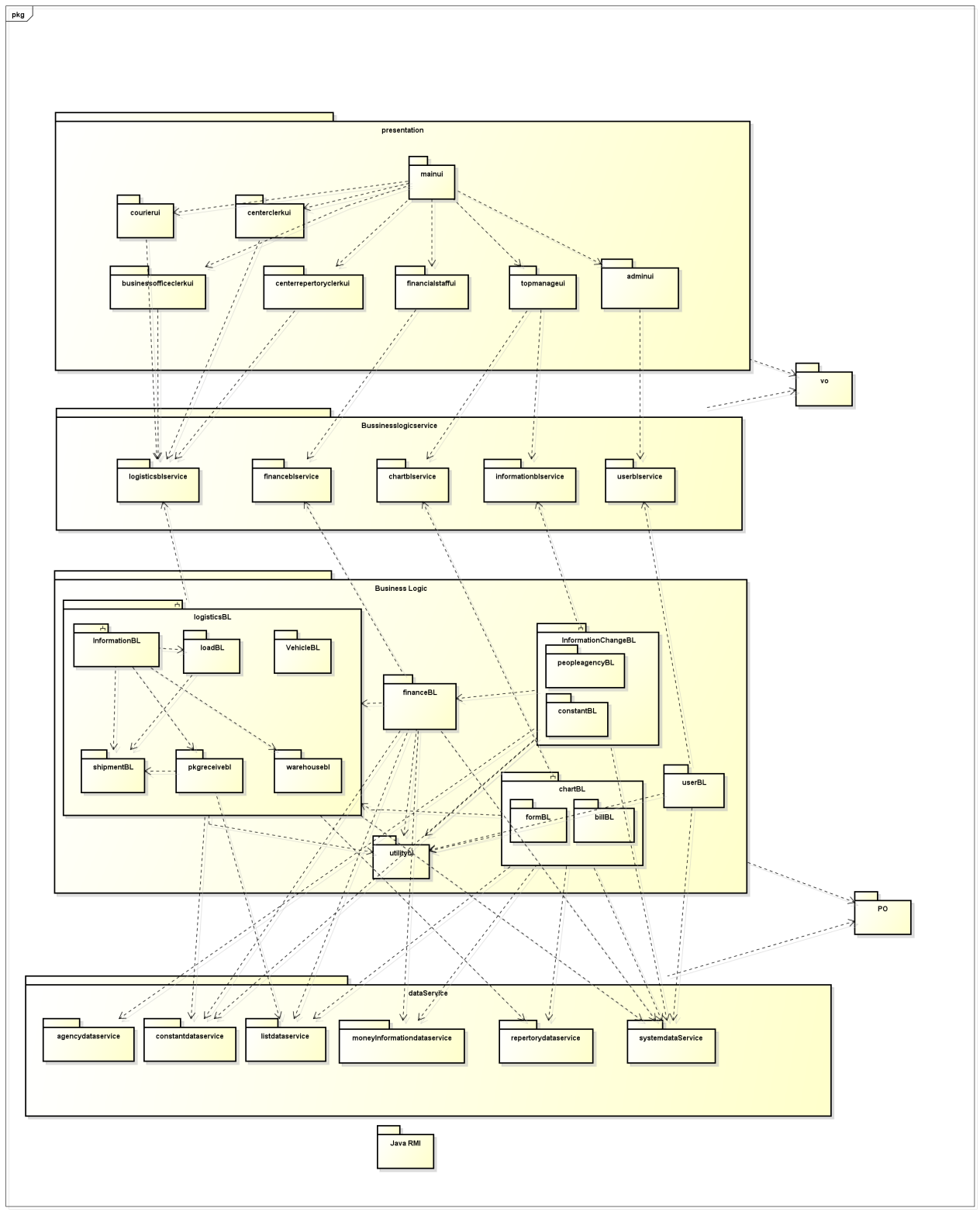


图3 物流管理系统客户端开发包图

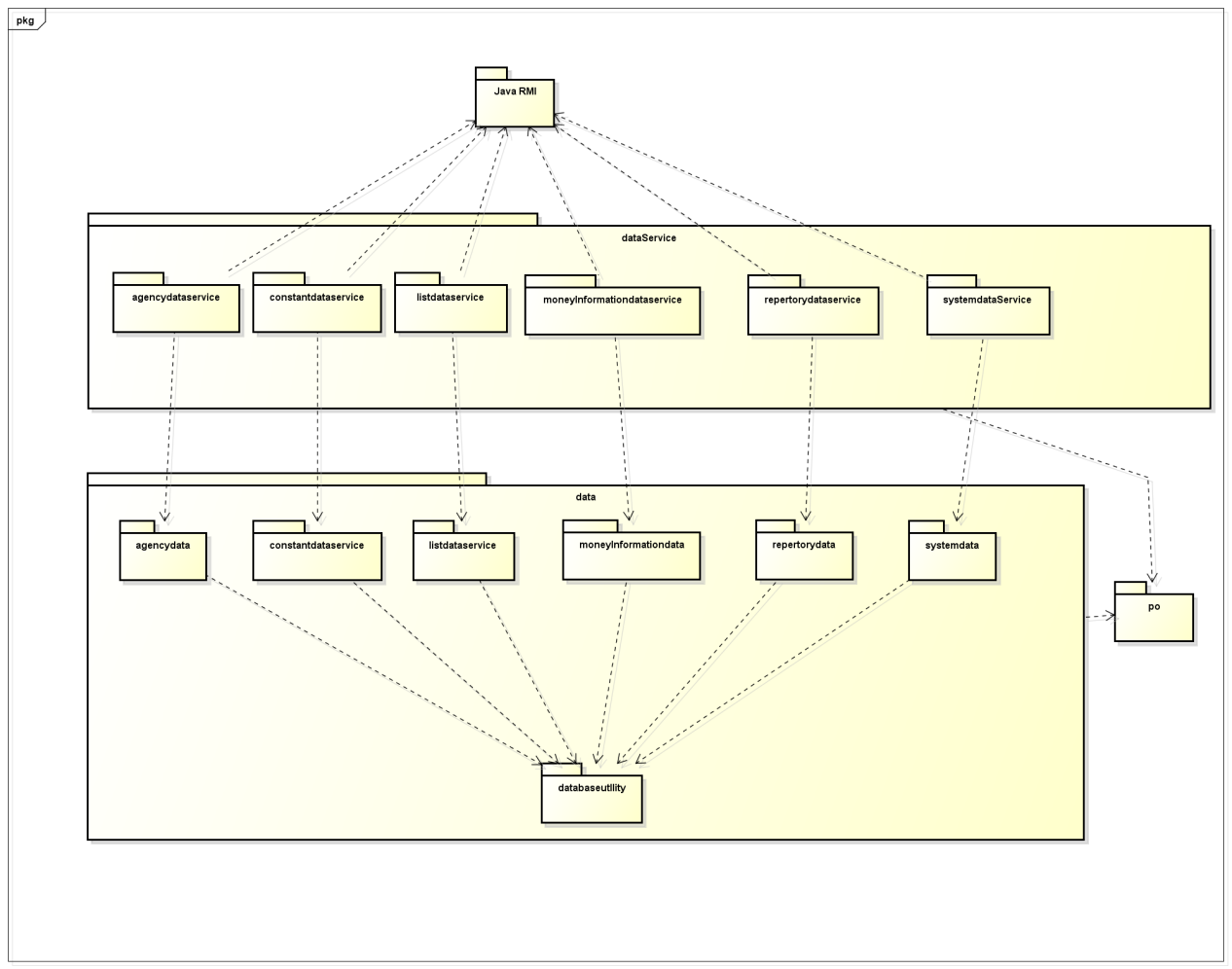


图4 物流管理系统服务器端开发包图

**4.2运行时进程**

在物流管理系统中，会有多个客户端进程和一个服务器端进程，其进程图如图5所示。结合部署图，客户端是在客户端机器上运行，服务器端在服务器端机器上运行。

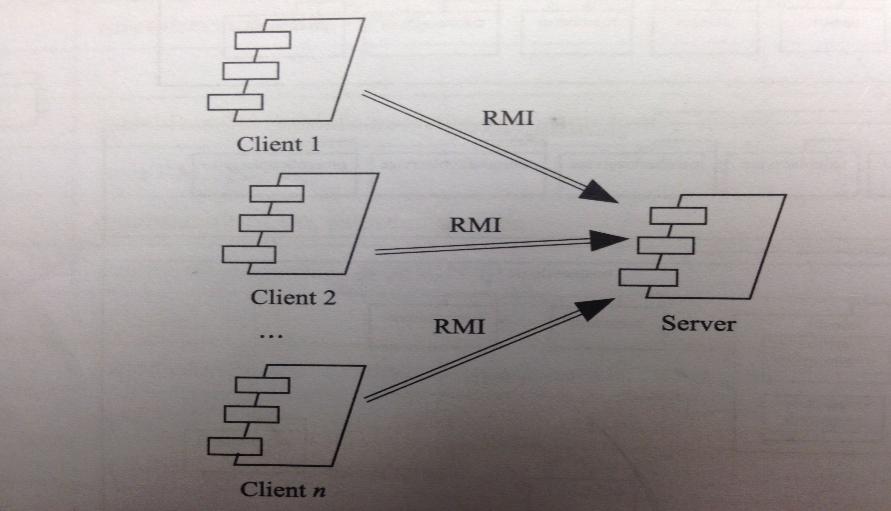


图5 进程图

**4.3物理部署**

物流管理系统中客户端构建是放在客户端机器上，服务器构建是放在服务器端机器上。在客户端节点上，还要部署RMIStub构件。由于Java RMI构件属于JDK6.0的一部分。所以，在系统JDK环境已经设置好的情况下，不需要再独立部署。部署图如图6所示。

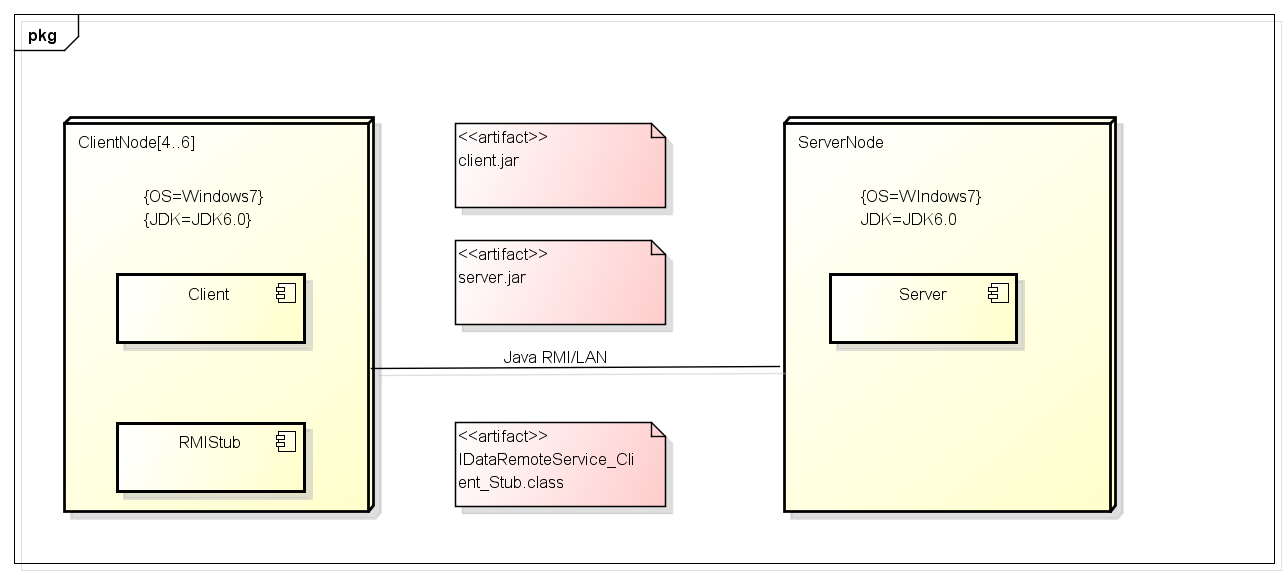


图6 部署图

**5.接口视角**

**5.1模块的职责**

客户端模块和服务器端模块视图分别如图7和图8所示。客户端各层和服务器端各层的职责分别如表2和表3所示。

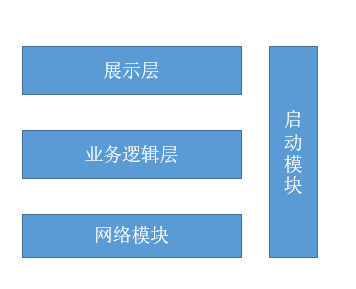


图7 客户端模块视图

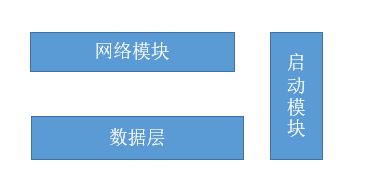


图8 服务器端模块视图

**表2 客户端各层的职责**

|  |  |
| --- | --- |
| **层** | **职责** |
| 启动模块 | 负责初始化网络通信机制，启动用户界面 |
| 用户界面层 | 基于窗口的物流管理系统客户端用户界面 |
| 业务逻辑层 | 对于用户界面的输入进行相应并进行业务处理逻辑 |
| 客户端网络模块 | 利用Java RMI机制查找RMI服务 |

**表3 服务器端各层的职责**

|  |  |
| --- | --- |
| **层** | **职责** |
| 启动模块 | 负责初始化网络通信机制，启动用户界面 |
| 数据层 | 负责数据的持久化及数据访问接口 |
| 服务器端网络模块 | 利用Java RMI机制开启RMI服务，注册RMI服务 |

每一层只是使用下方直接接触的层。层与层之间仅仅是通过接口的调用来完成的。层之间调用的接口如表4所示。

**表4 层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| **接口** | **服务调用方** | **服务提供方** |
| Logisticsblservice  Financeblservice  Chartblservice  Informationblservice  Userblservice | 用户端界面层 | 用户端业务逻辑层 |
| Agencydataservice  Constantinfodataservice  Listdataservice  Moneyinformationdataservice  Repertorydataservice  systemdataservice | 用户端业务逻辑层 | 服务器端数据层 |

**5.2用户界面层的分解**

根据需求，系统存在31个界面。界面跳转如图9所示。

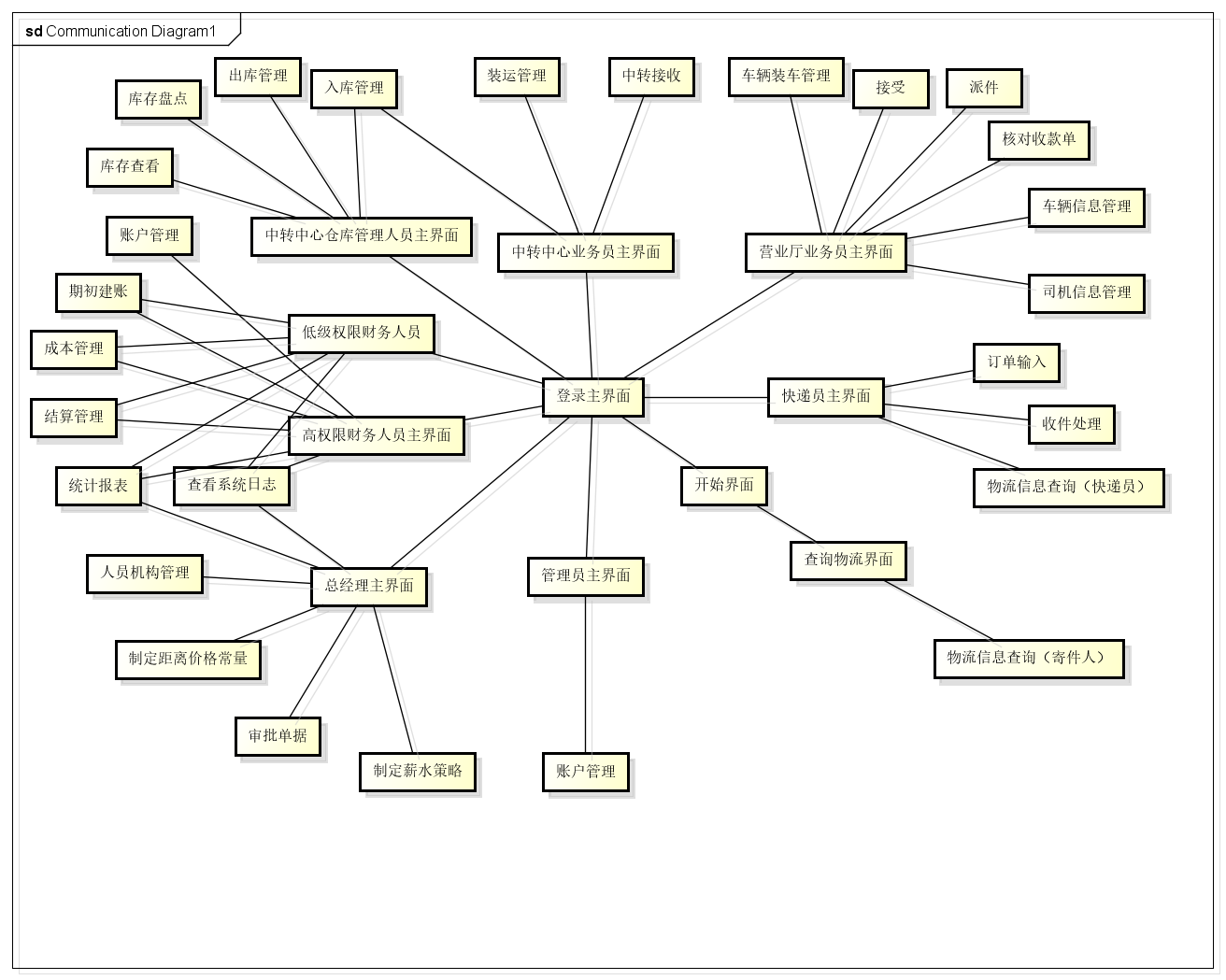


图9 用户界面跳转

用户界面类如图10所示。

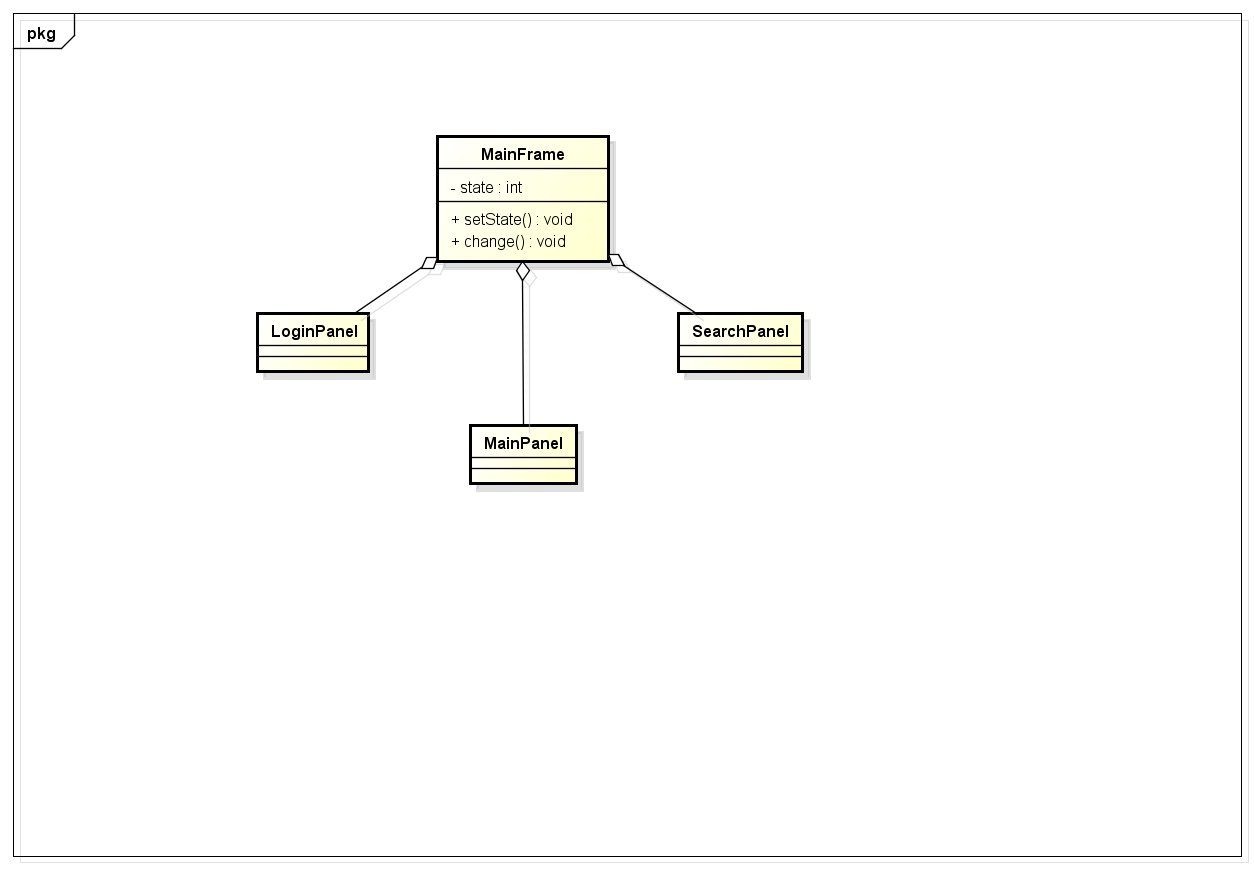


图10 用户界面类

5.2.1用户界面层模块的职责

如表5所示为用户界面层模块的职责。

**表5 用户界面层模块的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| MainFrame | 界面Frame，负责界面的展示和界面跳转 |
| MainPanel | 负责界面的跳转 选择向哪个界面跳转 |
| LoginPanel | 负责界面的登录，进入登录界面 |
| SearchPanle | 负责用户快递信息的查询，进入查询快递信息 |

5.2.2用户界面层模块的接口规范

用户界面层的接口规范如表6所示。

**表6 用户界面层模块的接口规范**

|  |  |  |
| --- | --- | --- |
| MainFrame | **语法** | Init（args:String[]） |
| **前置条件** | 无 |
| **后置条件** | 显示Frame和LoginPanel |
| LoginPanel | **语法** | Login（String admin，String password） |
| **前置条件** | 网络连接正常 |
| **后置条件** | 显示登陆界面 |
| MainPanel | **语法** | Change（） |
| **前置条件** | 点击转换面板按键 |
| **后置条件** | 显示主界面 |
| SearchPanel | **语法** | Search(String barcode) |
| **前置条件** | 订单号格式输入正确 |
| **后置条件** | 显示需要查找的订单信息 |

用户界面层需要的服务接口如表7所示

**表7 用户界面层模块需要的服务接口**

|  |  |
| --- | --- |
| 服务名 | 服务 |
| Busineesslogicservice.\*BLService | 每个界面都有一个相应的业务逻辑接口 |

5.2.3用户界面模块设计原理

用户界面利用Java的Swing和AWT库来实现。

**5.3业务逻辑层的分解**

业务逻辑层包括多个针对界面的业务逻辑处理对象。业务逻辑层的设计如图11所示。

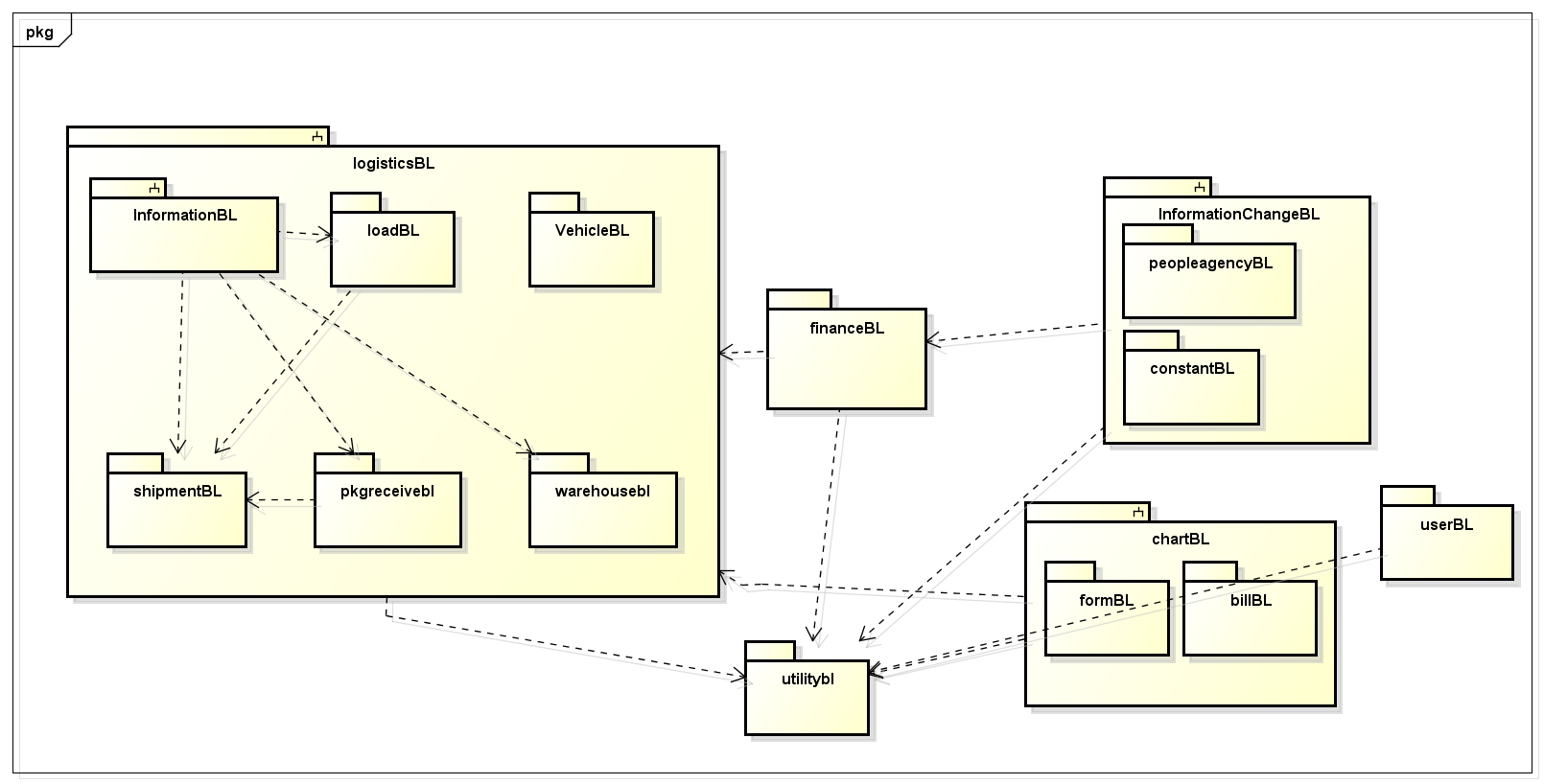
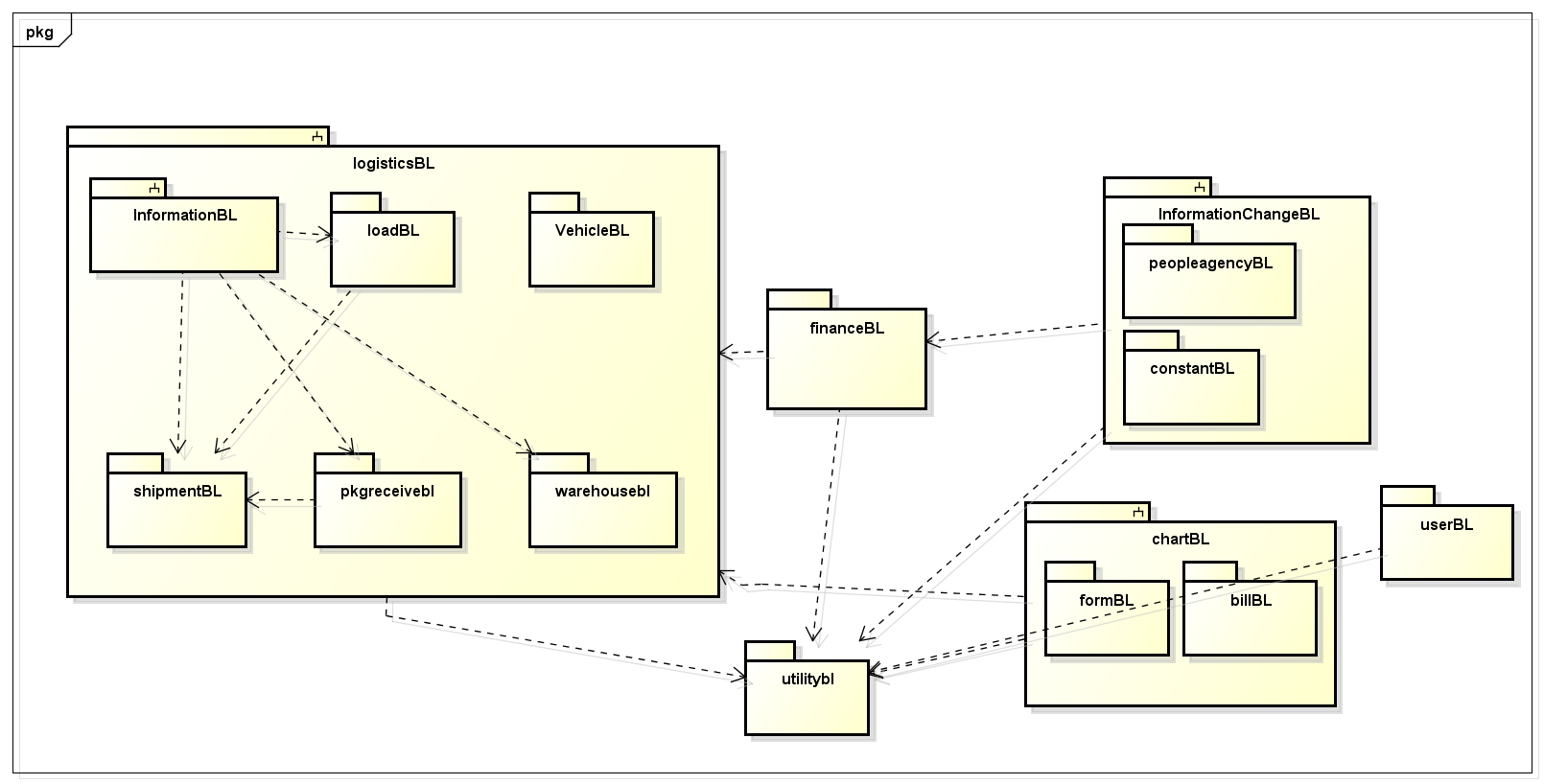


图11 业务逻辑层的设计

5.3.1业务逻辑层模块的职责

业务逻辑层模块的职责如表8所示。

**表8 业务逻辑层模块的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| Logisticsbl | 负责实现与物流系统相关所需要的服务 |
| Userbl | 负责实现对应与管理员登录管理所需要的服务 |
| Financebl | 负责实现与金融任务相关所需要的服务 |
| Informationchangebl | 负责实现与信息修改任务相关所需要的服务 |
| Chartbl | 负责实现与各种表格任务相关所需要的服务 |
| Utilitybl | 负责实现过程中各种各样通用的类的服务 |

5.3.2业务逻辑层模块的接口规范

**表9 Chartbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **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 |
| **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 |

**表10 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 |

**表11 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 |

**表12 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 |

**表13 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 |

**表14 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 |

**表15 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 |

**表16 PeopleAgencybl接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的接口（供接口）** | | |
| init | **语法** | public ResultMessage init(); |
| **前置条件** | 期初建账初始化 |
| **后置条件** | 返回操作反馈 |
| findBaseAccount | **语法** | public ArrayList<BankAccountVO> findBaseAccount(); |
| **前置条件** | 要求查找期初账户信息 |
| **后置条件** | 返回账户信息的集合 |
| findBaseAgency | **语法** | public ArrayList<BaseAgencyVO> findBaseAgency(); |
| **前置条件** | 要求查找期初机构信息 |
| **后置条件** | 返回机构信息的集合 |
| findBaseStaff | **语法** | public ArrayList<StaffVO> findBaseStaff(); |
| **前置条件** | 要求查找期初的员工信息 |
| **后置条件** | 返回员工信息的集合 |
| 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 |
| findBaseTruck | **语法** | public ArrayList<TruckVO> findBaseTruck(); |
| **前置条件** | 要求查找期初的车辆信息 |
| **后置条件** | 返回车辆信息的集合 |
| findBaseRepertory | **语法** | public ArrayList<RepertoryInfoVO> findBaseRepertory(); |
| **前置条件** | 要求查找期初的库存信息 |
| **后置条件** | 返回库存信息的集合 |

|  |  |
| --- | --- |
| **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 |

**表17 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 | |
| **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 | |

|  |  |  |
| --- | --- | --- |
| **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 | |

**表18 ReciveListBL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 | |

**表19 RepertoryManageBL 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 | |

**表20 SearchPkgBLservice 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 | |

**表21 SearchPkgBLservice 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 |

**表21 TransArrivalListBLservice 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 | |

**表22 ShipListBLservice 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 | |

**表23 TruckBLservice 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 | |

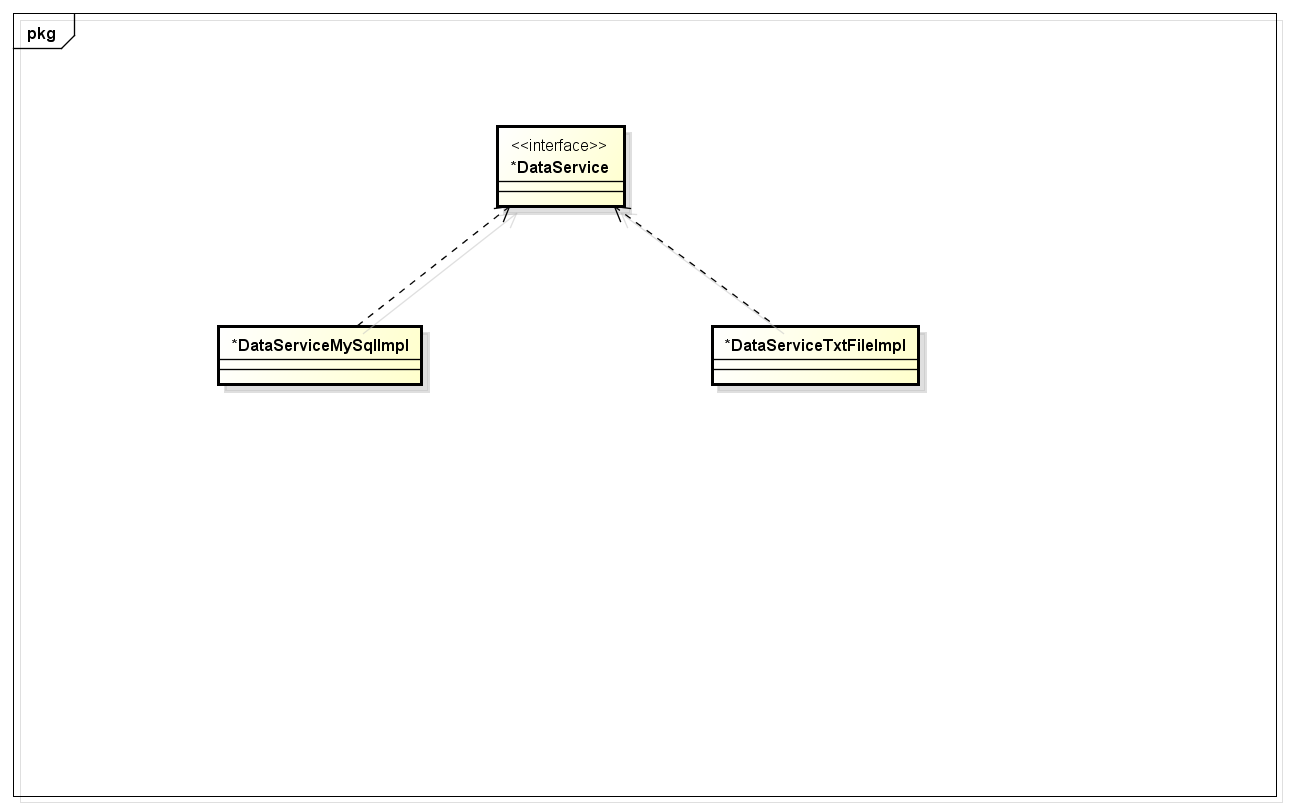
**表24 UserManagebl接口规范**

|  |  |  |
| --- | --- | --- |
| **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 |

**5.4数据层的分解**

数据层主要给业务逻辑层提供数据访问服务，包括对持久化数据的增、删、改、查。由于持久化数据的保存可能存在多种形式：Txt文件、数据库，所以抽象了数据服务。数据层模块的描述具体如图13所示。



**图13 数据层模块的职责**

5.4.1数据层模块的职责

数据层模块的职责如表11所示。

**表25 数据层模块的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| \*DataService | 持久化数据库接口，提供机体载入、集体保存、增、删、改、查服务 |
| \*DataServiceTxtFileImpl | 基于Txt文件的持久化数据库的接口，提供集体载入，集体保存，增、删、改、查服务 |
| \*DataServiceMySqlImpl | 基于序列化文件的持久化数据库的接口，提供集体载入，集体保存，增、删、改、查服务 |

5.4.2数据层模块的接口规范

**表26 数据层模块的接口规范**

**AgencydataService**

|  |  |  |
| --- | --- | --- |
| **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 |

**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 |

**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 |

**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 |

**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 |

**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 |

**6.信息视角**

**6.1数据持久化对象**

·**UserPO**类包含用户的用户名、密码属性。  
**· BankAccountPO**类中包含了银行的账户信息，账号，密码，余额。 ·OrderListPO类包含寄件人姓名、住址、单位、电话、手机；收件人姓名、住址、单位、电话、手机；托运货物信息（原件数、实际重量、体积、内件品名、尺寸）；包装费（纸箱、木箱、快递袋、其它）；费用合计；订单条形码号（10位数）；经济快递、标准快递、特快；种类。  
·**TransShipmentPO**类包含装车日期、本营业厅编号（025城市编码+000鼓楼营业厅）、汽运编号 （营业厅编号+20150921日期+00000编码 、五位数字）、到达地（本地中转中心或者其它营业厅）、车辆代号、监装员、押运员、本次装箱所有订单条形码号）、运费（运费根据出发地和目的地自动生成）  
·**TruckPO**类包含车辆代号（城市编号（电话号码区号南京025）+营业厅编号（000三位数字）+000三位数字）、车牌号（苏A 00000）、服役时间；司机信息（司机编号、营业厅编号、000、姓名、出生日期、身份证号、手机、性别、行驶证期限）  
·**LoadListPO**类包含装车日期，本中转中心中转单编号（区别航运货运汽运），航班号（或者火车车次，汽车车次），出发地，到达地，货柜号（车厢号），监装员，本次装箱所有托运单号，运费  
·**AgencyPO**类包含机构信息，人员内容，机构编号  
·**repertoryInfoPO**类包含位置（区、排、架、位机）及位置上放置的商品。  
·**DriverPO**类包含了司机的司机编号，姓名，出生日期，身份证号，手机号，性别，行驶证期限。  
·**GoodsInfoPO**包含了货物快递编号，货物到达状态，货物出发地  
·**HallArrivalListPO**类包含了到达信息，中转单编号，出发地，到达状态，货物托运条形码  
·**MoneyInformationPO**类包含成本收益日期，当日收入，当日支出，当日利润  
·**MoneyInListPO**类包含了收款日期，收款金额，快递员姓名，条形码  
·**MoneyOutListPO**类包含了日期，金额，收款人，银行账号，付款方式  
·**RepertoryInPO**类包含了快递编号，入库时间，目的地，区号，排号，架号，位号  
·**RepertoryOutPO**类包含了快递编号，出库时间，目的地，装运方式，中转单编号，汽运编号  
·**StaffPO**类包含了姓名，性别，职位，身份证，工作时长，手机，工资  
·**SystemLogPO**系统日志类包含了创建时间，操作者，操作内容  
·**TransArrivalListPO**包含了中转单编号，中转中心编号，货物到达日期，货物状态信息

**·OrderListPO** 包括了所有的订单信息，订单号，发件人，发件人地址，发件人手机，收件人，收件人地址，收件人手机，实际签收人，订单费用，货物体积，货物重量，货物达到路径

**6.2 持久化格式**

**1.AgencyPO**

000001；010001,025002;

**2.BankAccountPO**

张三:000000:1000(账户号，余额)

**3.DeliveringListPO**

2015-10-25: OrderList1; OrderList2; OrderList3（编号）

**4.DistanceChartPO**

String[][] distanceChart

**5.DriverPO**

025000000:张三:1970-1-1:131029197001011356:13500000000:男:2015-10-27

**6.GoodsInfoPO**

201510260925(快递编号):已到达/未到达:南京

**7.HallArrivalListPO**

2015-10-26:0250000002015092100000(装车单/中转单编号):南京:到达/未到达:

OrderList1; OrderList2; OrderList3（编号）

**8.LoadaListPO**

2015-10-26:025000(本营业厅编号):2015092100000(汽运编号):北京:A00000(车辆代号):张三：

张三：OrderList1; OrderList2; OrderList3

**9.MoneyInformationListPO**

2015-10-26:100:50:50

**10.MoneyInListPO**

2015-10-26:100:张三：String1；String2；String3；

**11.MoneyOutListPO**

2015-10-26:100:张三：000000(银行账号)：现金/支付宝(付款方式)

**12.OrderListPO**

张三:南京市栖霞区(客户地址):供电局(客户单位地址):13100000000:2:10(kg):5000(cm^3):玻璃瓶:南京市 鼓楼营业厅至中转中心途中:10:201510260925

**13.PriceChartPO**

10:20:50:18:23:25

**14.ReceiverPO**

李四：2015-10-27:

**15.RepertoryInfoPO**

100:50:200:repertory[2][2][2][2][2][2](库存情况)

**16.RepertoryInPO**

201510260925:201510261224(入库时间):北京:1:1:1:1(区;排;架;位)

**17.RepertoryOutPO**

201510261046:201510262310:北京：火车：2015092100000

**18.StaffPO**

张三:男:快递员：131028197001010101：12(month):13100000000:3000(RMB)

**19.SystemLogPO**

201510260925(创建时间):张三:增加账户信息(操作内容)

**20.TransArrivalListPO**

0250000002015092100000:0250(中转中心编号):2015-10-27:未到达

**21.TransShipmentListPO**

2015-10-26: 0250000002015092100000:F00000(航班号)：南京:北京：20(货柜号):张三：OrderList1；OrderList2；OrderList3（单号）:50

**22.TruckPO**

000:E00000:1(年):

**23.UserPO**

Admin:admin:高/中/低

**时间格式**：2009-11-24 18:09:37

**订单货物轨迹信息格式**：时间+“ ”+地点

**营业厅编号**：025（区号）001（区内编号）

**中转中心编号**：025（区号）000

**汽运编号**：025001（营业厅编号或者中转中心编号）20151111（日期编号）00000（顺序数）

**订单编号**：025001（营业厅编号）20151111（日期）1011（时、分）

**中转单编号**：025000（中转中心编号）20151111（日期）0000000（顺序数）

**司机编号**：025001（营业厅编号）000（顺序数）

**货车编号**：车牌号（7位）

**员工编号**：025（3位区号）+000或001（顺序号）或111（分别表示中转中心，营业厅，快递中心）+000（顺序号，员工编号）

**任何未定义的id**：前六位是营业厅或中转中心编号，后5位是随机编号