体系结构文档

物流信息管理系统

更新历史：

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 修改原因 | 版本号 |
| 倪辰皓 | 2015/10/26 | 完成文档 | V1.0 |
| 全体人员 | 2015/11/6 | 修改了模块之间的业务依赖关系 | V1.1 |
| 倪辰皓 | 2015/11/15 | 更新了开发包图 | V1.2 |
| 栾志远、孙婧 | 2015/12/27 | 完善文档 | V1.3 |

目录

[体系结构文档 1](#_Toc433654926)

[1.引言 3](#_Toc433654927)

[1.1编制目的 3](#_Toc433654928)

[2.产品概述 3](#_Toc433654929)

[3.逻辑视角 3](#_Toc433654930)

[4.组合视角 4](#_Toc433654931)

[4.1开发包图 4](#_Toc433654932)

[4.2运行时进程 7](#_Toc433654933)

[4.3物理部署 7](#_Toc433654934)

[5.接口视角 8](#_Toc433654935)

[5.1模块的职责 8](#_Toc433654936)

[5.2 用户界面层的分解 10](#_Toc433654937)

[5.2.1 用户界面层模块的职责 10](#_Toc433654938)

[5.2.2 用户界面层模块的接口规范 12](#_Toc433654939)

[5.2.3 用户界面模块设计原理 12](#_Toc433654940)

[5.3业务逻辑层的分解 12](#_Toc433654941)

[5.3.1业务逻辑层模块的职责 12](#_Toc433654942)

[5.3.2业务逻辑模块的接口规范 12](#_Toc433654943)

[5.4数据层的分解 18](#_Toc433654944)

[5.4.1数据层模块的职责 18](#_Toc433654945)

[5.4.2数据层模块的接口规范 18](#_Toc433654946)

[6.信息视角 21](#_Toc433654947)

[6.1数据持久化对象 21](#_Toc433654948)

[6.2文件格式 22](#_Toc433654949)

# 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目的，同

时实现和开发人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

# 2.产品概述

XXX是一本地民营物流企业。他们希望扩大快运服务通达的片区。随着公司规模扩大，企业业务量，办公场所，员工数都发生增长，该企业设立了新的运营模式，希望尽快适应新的环境，并提高工作效率和用户满意度。

快递物流系统就是为了满足该物流企业新的运营模式的需求而开发的。它包括一个数据集中服务器和多个客户端。数据集中服务器将所有的数据存储起来进行维护。用户通过客户端完成日常任务，客户端与数据集中服务器采取实时通信的方式完成数据交换。

# 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访问。

物流信息管理系统的主要功能有：物流信息查询、订单输入、报价和时间管理、收件信息输入、车辆装车管理、接收与派件、收款单填写、车辆信息管理、司机信息管理、飞机装运管理、火车装运管理、汽车装运管理、中转接收、出库、入库、库区调整、库存信息初始化、库存盘点、结算管理、财务支出、成本收益表查询、经营情况表查询、账户管理、人员与机构管理、审判单据、用户管理

可以合并为：

物流信息查询--信息查询

订单输入、报价和时间管理、订单信息查询--寄件

审判单据--单据审判

车辆装车管理、飞机装运管理、火车装运管理、汽车装运管理、中转接收--装运

收件信息输入、营业厅到达单填写、派件单填写--收件

出库、入库、库区调整、库存信息初始化、库存盘点--库存管理

账户管理、收款单填写、结算管理、财务支出--财务

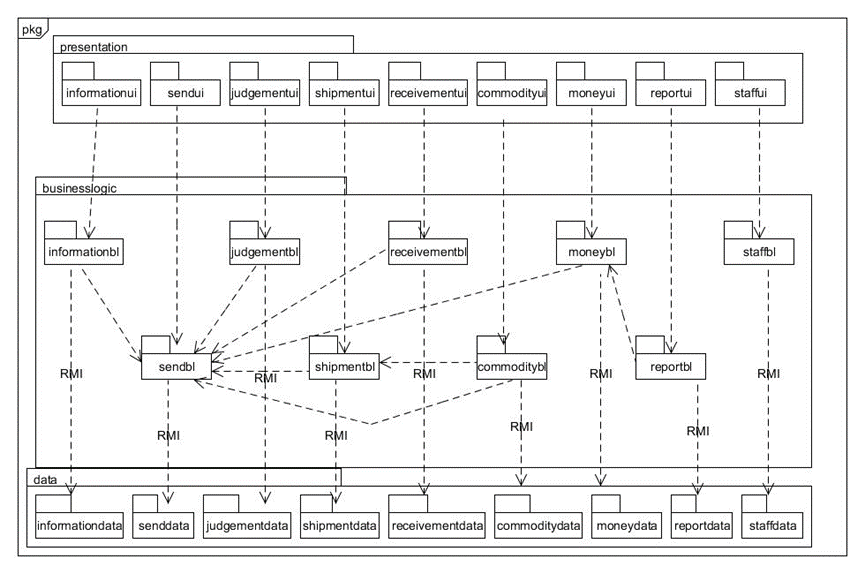
成本收益表查询、经营情况表查询—报表查询

人员管理、机构管理、车辆信息管理、司机信息管理、用户管理--人员管理

**物流信息管理系统概要功能设计**

|  |  |
| --- | --- |
| **功能** | **对应逻辑包** |
| 信息查询 | informationui，informationbl，informationdata  sendbl,senddata  shipmentbl,shipmentdata  receivementbl, receivementdata |
| 寄件 | sendui,sendbl,senddata |
| 单据审判 | judgementui, judgementbl  sendbl,senddata  shipmentbl, shipmentdata  receivementbl, receivementdata  moneybl,moneydata |
| 装运 | shipmentui,shipmentbl,shipmentdata |
| 收件 | receivementui, receivementbl, receivementdata |
| 库存管理 | commodityui, commoditybl, commoditydata |
| 财务 | moneyui,moneybl,moneydata  sendbl,senddata |
| 报表查询 | reportui,reportbl,reportdata  moneybl,moneydata |
| 人员管理 | staffui,staffbl,staffdata |

**软件体系结构设计方案**



# 4.组合视角

## 4.1开发包图

物流系统的最终开发包设计如表1所示

表1物流系统的最终开发包设计

|  |  |
| --- | --- |
| 开发（物理包） | 依赖的其他开发包 |
| mainui | informationui,sendui,judgementui,shipmentui,receivementui,commodityui,moneyui,reportui,staffui,vo |
| informationui | informationblservice,界面库类包 |
| informationblservice |  |
| informationbl | informationblservice,sendbl,po, informationdataservice |
| informationdataservice | Java RMI,po |
| informationdata | informationdataservice, databaseutility |
| sendui | sendblservice,界面类库包 |
| sendblservice |  |
| sendbl | sendblservice,po, senddataservice |
| senddataservice | Java RMI,po |
| senddata | senddataservice,databaseutility |
| judgementui | judgementblservice,界面类库包 |
| judgementblservice |  |
| judgementbl | judgementblservice, sendbl ,po, judgementdataservice |
| judgementdataservice | Java RMI,po |
| judgementdata | judgementdataservice, databaseutility |
| shipmentui | shipmentblservice,界面类库包 |
| shipmentblservice |  |
| shipmentbl | shipmentblservice, sendbl ,po, shipmentdataservice |
| shipmentdataservice | Java RMI,po |
| shipmentdata | shipmentdataservice, databaseutility |
| receivementui | receivementblservice,界面类库包 |
| receivementblservice |  |
| receivementbl | receivementblservice，sendbl ,po, receivementdata |
| receivementdataservice | Java RMI,po |
| receivementdata | receivementdataservice, databaseutility |
| commodityui | commodityblservice,界面类库包 |
| commodityblservice |  |
| commoditybl | commodityblservice,sendbl ,shipmentbl,po,commoditydataservice |
| commoditydataservice | Java RMI,po |
| commoditydata | commoditydataservice, databaseutility |
| moneyui | moneyblservice,界面类库包 |
| moneyblservice |  |
| moneybl | moneyblservice,sendbl ,po, moneydataservice |
| moneydataservice | Java RMI,po |
| moneydata | moneydataservice, databaseutility |
| reportui | reportblservice,界面类库包 |
| reportblservice |  |
| reportbl | reportblservice，moneybl,po, reportdataservice |
| reportdataservice | Java RMI,po |
| reportdata | reportdataservice, databaseutility |
| staffui | staffblservice,界面类库包 |
| staffblservice |  |
| staffbl | staffblservice,po, staffdataservice |
| staffdataservice | Java RMI,po |
| staffdata | staffdataservice, databaseutility |
| vo |  |
| po |  |
| utilitybl |  |
| 界面类库包 |  |
| Java RMI |  |
| databaseutility |  |

物流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。

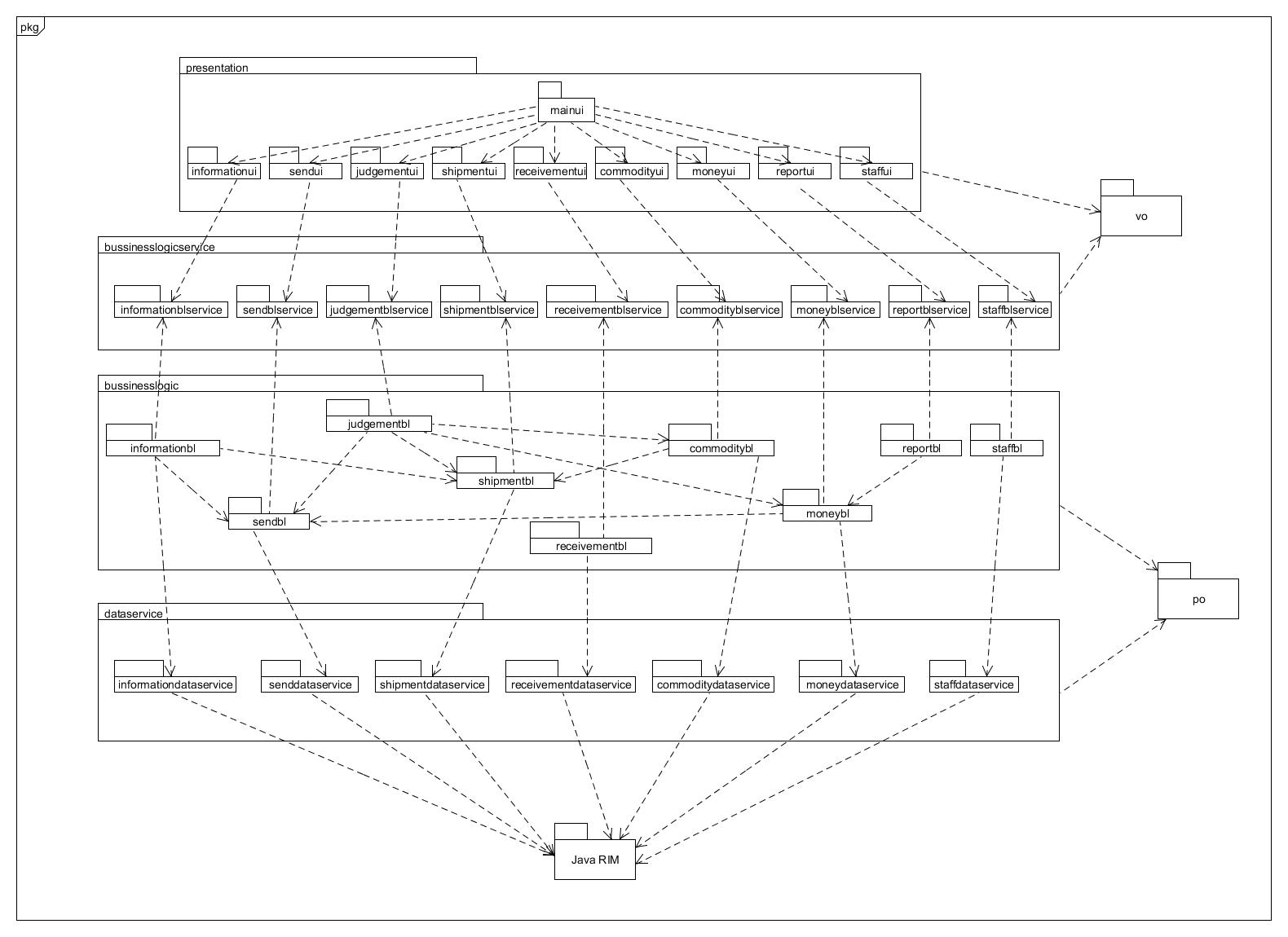


图3 快递系统客户端开发包图

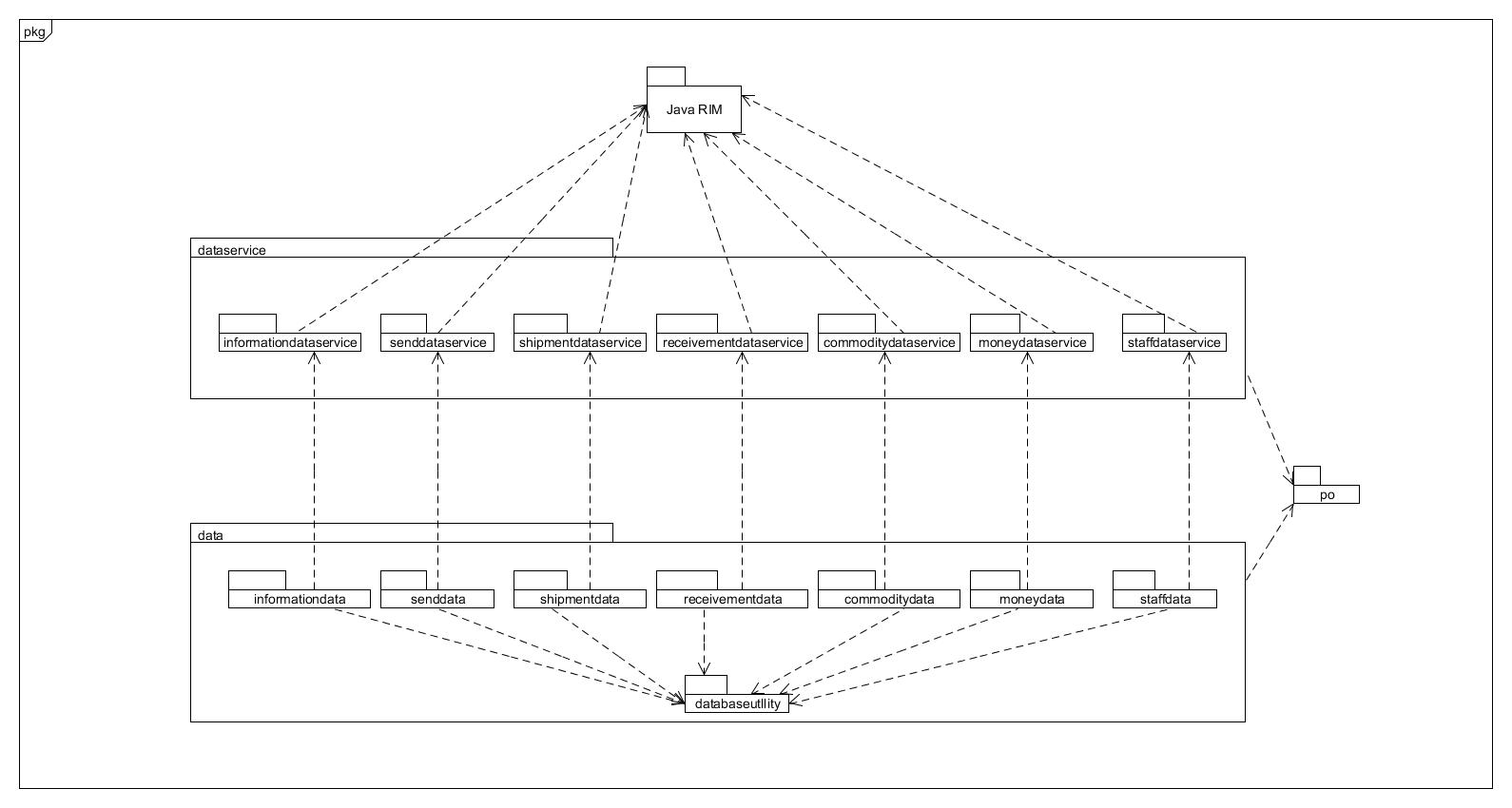


图4 物流系统服务器端开发包图

## 4.2运行时进程

在快递系统中，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，客户端进程是在客户端机器上运行，服务器端进程是在服务器端机器上运行。

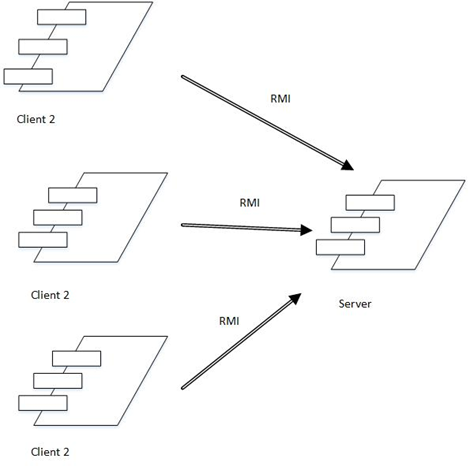


图5 进程图

## 4.3物理部署

物流系统中客户端构件是放在客户端机上，服务器端构件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JDK 1.8的一部分。所以，在系统JDK环境已经设置好的情况下，不需要独立部署。部署图如图6所示。

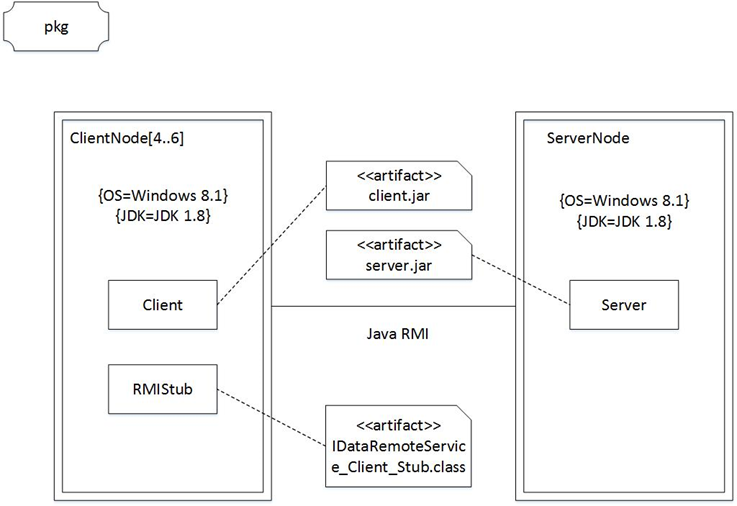


图6 部署图

# 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。

启动模块

展示层

业务逻辑层

网络模块

图7 客户端模块视图

启动模块

网络模块

数据层

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 | 基于窗口的物流公司客户端用户界面 |
| 业务逻辑层 | 对于用户界面的输入进行响应并进行业务处理逻辑 |
| 客户端网络模块 | 利用JavaRMI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JavaRMI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** |

|  |  |  |
| --- | --- | --- |
| informationBLService  sendBLService  judgementBLService  shipmentBLService  receivementBLService  commodityBLService  moneyBLService  reportBLService  staffBLService | 客户端展示层 | 客户端业务逻辑层 |
| informationDataService  sendDataService  judgementDataService  shipmentDataService  receivementDataService  commodityDataService  moneyDataService  reportDataService  staffDataService | 客户端业务逻辑层 | 服务器端数据层 |

以下图9是寄件用例层之间的调用。每一层之间都是由上层依赖了一个接口（需接口），而下层实现这个接口（供接口）。sendblservice提供了send界面所需要的所有业务逻辑功能。senddataservice提供了对数据库的增、删、改、查等操作。这样的实现就大大降低了层与层之间的耦合。

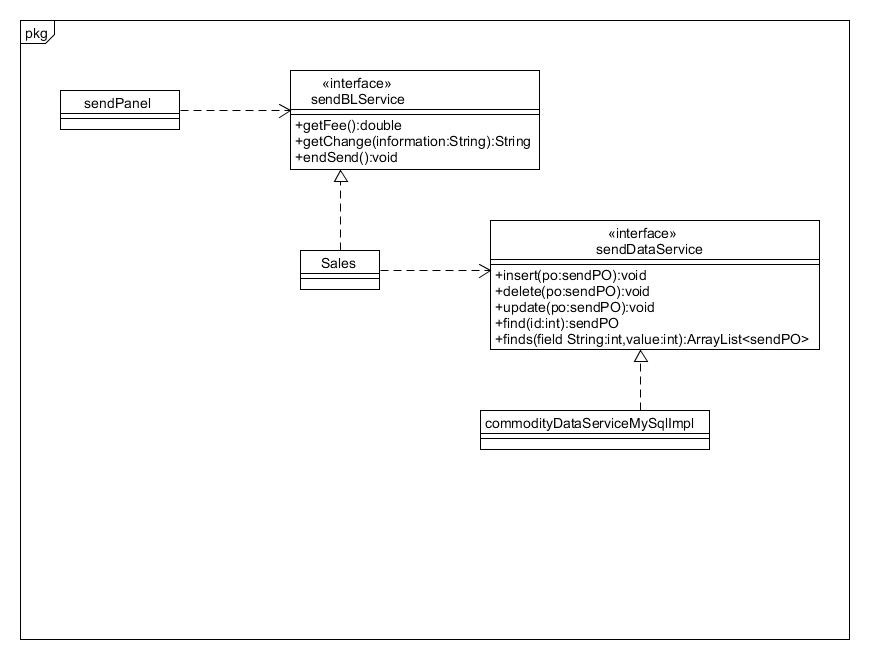


图9 寄件用例层之间调用的接口

## 5.2 用户界面层的分解

根据需求，系统存在26个用户界面：登录界面、寄件人查询界面、物流信息界面、快递员主界面、订单输入界面、报价和时间管理界面、收件信息输入界面、营业厅业务员主界面、装车信息管理界面、车辆信息管理界面、司机信息管理界面、中转中心业务员主界面、货物装运信息管理界面、中转中心仓库管理员主界面、仓库信息管理界面、库存信息初始化界面、财务人员主界面、结算信息管理界面、统计报表界面、账户信息管理界面、总经理主界面、人员和机构信息管理界面、审判单据界面、查看统计分析界面、管理员主界面、调整用户界面。界面跳转如图10所示。

所有的用户界面通过同一个主用户界面类（MainFrame)处理，如图11。

### 5.2.1 用户界面层模块的职责

**用户界面层模块的职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| MainFrame | 界面Frame，负责界面的显示和界面的跳转 |

调整用户界面

管理员主界面

查看统计分析界面

单据审判界面

人员和机构信息管理界面

总经理主界面

账户信息管理界面

统计报表界 面

结算信息管理界面

财务人员主界面

库存信息初始化界面

仓库信息管理界面

中转中心仓库管理员主界面

货物装运信息界面

中转中心业务员主界面

司机信息管理界面

车辆信息管理界面

装车信息管理界面

营业厅业务员主界面

收件信息输入界面

报价和时间管理界面

订单输入

界面

物流信息界面

快递员主界面

寄件人查询

界面

登录界面

图10 界面跳转

pkg

MainFrame

+int():void

图11 用户界面类

### 5.2.2 用户界面层模块的接口规范

**用户界面层模块的接口规范**

|  |  |  |
| --- | --- | --- |
|  | 语法 | int(args:String[]) |
| **MainFrame** | 前置条件 | 无 |
|  | 后置条件 | 显示Frame以及LoginPanel |

**用户界面层模块需要的服务接口**

|  |  |
| --- | --- |
| **服务名** | **服务** |
| businesslogicservice.LoginBLService | 登录界面的业务逻辑接口 |
| Businesslogicservice.\*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

### 5.3.1业务逻辑层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| imformationbl | 负责实现信息查询界面需要的服务 |
| sendbl | 负责实现寄件界面需要的服务 |
| judgementbl | 负责实现单据审判界面需要的服务 |
| shipmentbl | 负责实现装运界面需要的服务 |
| receivementbl | 负责实现收件界面 |
| commoditybl | 负责实现库存管理界面需要的服务 |
| moneybl | 负责实现财务管理界面需要的服务 |
| reportbl | 负责实现报表查询界面需要的服务 |
| staffbl | 负责实现人员管理界面需要的服务 |

### 5.3.2业务逻辑模块的接口规范

**commoditybl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的接口（供接口）** | | | |
| Commodity.inDepot | 语法 | | public StockBill\_In inDepot(StockBill\_In stock\_in) |
| 前置条件 | | 输入合法 |
| 后置条件 | | 加入一个入库单对象信息 |
| Commodity.getUnmarkingBills | 语法 | | public ArrayList<stockBill> getUnmarkingBills () |
| 前置条件 | | 输入合法 |
| 后置条件 | | 加入未审判的单据信息 |
| Commodity. outDepot | 语法 | | public StockBill\_Out outDepot(StockBill\_Out stock\_out) |
| 前置条件 | | 输入合法 |
| 后置条件 | | 加入一个出库单对象信息 |
| Commodity.areaAdjust | 语法 | | public boolean areaAdjust(int[] rows, String type) |
| 前置条件 | | 输入合法 |
| 后置条件 | | 加入库区调整信息 |
| Commodity.init | 语法 | | public boolean init (int[] motorTransport, int[] trainTransport, int[] airTransport, int[] balnkSpace) |
| 前置条件 | | 输入合法 |
| 后置条件 | | 库存信息初始化 |
| Commodity.getStockSnapShot | 语法 | | public StockSnapShotSVO getStockSnapShot() |
| 前置条件 | | 输入合法 |
| 后置条件 | | 得到库存盘点信息 |
| Commodity.export | 语法 | | public boolean export() |
| 前置条件 | | 输入合法 |
| 后置条件 | | 导出库存盘点 |
| **需要的接口（需接口）** | | | |
| 服务名 | | 服务 | |
| Commoditydataservice.insert(StockBill\_In bill) | | 添加单一持久化对象 | |
| Commoditydataservice.delete(String id) | | 删除单一持久化对象 | |
| Commoditydataservice.update(StockBill\_In bill) | | 更新单一持久化对象 | |
| Commoditydataservice.find(String id) | | 根据编号查找相应持久化对象 | |
| Commoditydataservice.getCommoditydata | | 得到commodity数据的服务引用 | |
| Commoditydataservice.init() | | 初始化持久数据 | |

**Imformationbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Information.inquireTransMesg | 语法 | Public PackageVO inquireTransMesg(String id) |
| 前置条件 | 输入符合输入规则 |
| 后置条件 | 返回快递物流信息 |
| Information.refreshMesg | 语法 | Public boolean refreshMesg(String id,String position,String time) |
| 前置条件 | 输入符合规则 |
| 后置条件 | 更新物流信息 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| InformationDataService  .getTransMesg(String id) | 返回物流信息 | |
| InformationDataService  .refreshTransMesg(TransMesgPO po) | 更新物流信息 | |
| Sendbl.inquireSendMesg(String id) | 查询物流信息 | |

**moneybl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供服务） | | | |
| Money.addAccount | 语法 | | public boolean addAccount(ArrayList<Account> account) |
|  | 前置条件 | | 输入合法 |
|  | 后置条件 | | 加入一个账户的信息 |
| Money.deleteAccount | 语法 | | public boolean deleteAccount (ArrayList<String> ID) |
|  | 前置条件 | | 输入合法 |
|  | 后置条件 | | 删除一个账户的信息 |
| Money.changeAccountInfo | 语法 | | public boolean changeAccountInfo (ArrayList<String> ID, ArrayList<String> name) |
|  | 前置条件 | | 输入合法 |
|  | 后置条件 | | 更新一个账户的信息 |
| Money.getAccount | 语法 | | public ArrayList<Account> getAccount (String KeyWord) |
|  | 前置条件 | | 输入合法 |
|  | 后置条件 | | 查看一个账户的信息 |
| Money.getBills | 语法 | | public ReceiveMoneyBill[] getBills (String[] Date, String bussinessHall) |
|  | 前置条件 | | 输入合法 |
|  | 后置条件 | | 得到当天的收款单信息 |
| Money.getTotalMoney | 语法 | | Public double getTotalMoney () |
|  | 前置条件 | | 输入合法 |
|  | 后置条件 | | 得到所查看的收款单总金额 |
| Money.addPaidmentBill | 语法 | | Public boolean addPaidmentBill (ReceiveMoneyBill[] bills) |
|  | 前置条件 | | 输入合法 |
|  | 后置条件 | | 加入一个收款单信息 |
| Money. getPaidmentBill | 语法 | | Public ArrayList<ReceiveMoneyBill> getPaidmentBill (String[] time) |
| 前置条件 | | 输入合法 |
| 后置条件 | | 得到收款单信息 |
| Money.getUnmarkingBills | 语法 | | Public ArrayList<ReceiveMoneyBill> getUnmarkingBills () |
| 前置条件 | | 输入合法 |
| 后置条件 | | 添加一个未审判的单据 |
| Money.addPaymentBill | 语法 | | Public boolean addPaymentBill (ArrayList<PaymentBill>] bills) |
| 前置条件 | | 输入合法 |
| 后置条件 | | 加入一个付款单信息 |
| Money. getPaymentBill | 语法 | | Public ArrayList<PaymentBill> getPaymentBill (String[] endTime) |
| 前置条件 | | 输入合法 |
| 后置条件 | | 得到付款单信息 |
| 需要的服务（需服务） | | | |
| 服务名 | | 服务 | |
| Moneydataservice.insert(Account account) | | 添加单一持久化对象 | |
| Moneydataservice.delete(String id) | | 删除单一持久化对象 | |
| Moneydataservice.update(Account account) | | 更新单一持久化对象 | |
| Moneydataservice.find(String id) | | 根据编号查找相应持久化对象 | |
| Moneydataservice.init() | | 初始化单一持久化对象 | |
| Moneydataservice.insertMember（String id) | | 添加单一持久化账户对象 | |
| Moneydataservice.deleteMember（String id) | | 删除单一持久化账户对象 | |
| Moneydataservice.findMember（String id) | | 查找单一持久化账户对象 | |

**receivementbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| ReceivementController.billFilled | 语法 | public boolean billFilled (ArrayList<ReceiveInformationVO> info) |
| 前置条件 | 已输入收件单信息 |
| 后置条件 | 系统持久化更新收件单信息 |
| ReceivementContoller.distribution | 语法 | Public boolean distribution(ReceivementPO po) |
| 前置条件 | 已输入派件单信息 |
| 后置条件 | 系统持久化更新派件单信息 |
| ReceivementController.receive | 语法 | public boolean receive(ArrivementBill\_Shop arrive) |
| 前置条件 | 已输入营业厅到达单信息 |
| 后置条件 | 系统持久化更新营业厅到达单信息 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| Receivementdataservice.insert(ReceiveBill bill) | 插入单一持久化对象 | |
| receivementdataservice.init() | 初始化持久化数据 | |
| receivementdataservice.delete(String id) | 删除单一持久化对象 | |
| receivementdataservice.update(String id) | 更新单一持久化对象 | |
| receivementdataservice.find(String id) | 查找单一持久化对象 | |

**reportbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| Report.show | 语法 | public VO show() |
| 前置条件 | 输入合法 |
| 后置条件 | 显示一个表对象 |
| Report.export | 语法 | public boolean export() |
| 前置条件 | 输入合法 |
| 后置条件 | 导出一个表对象 |

**Sendbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| Send.inquireSendMesg | 语法 | Public OrderBillPO inquireSendMesg(String id) |
| 前置条件 | 使用者有查看订单信息的权限 |
| 后置条件 | 无 |
| Send.push | 语法 | public boolean push(OrderBillPO order) |
| 前置条件 | 使用者有输入订单的权限 |
| 后置条件 | 系统申请存储订单信息 |
| Send.getUnmarkingBills | 语法 | public ArrayList<OrderBillPO> getUnmarkingBills () |
| 前置条件 | 输入符合规则 |
| 后置条件 | 系统获得未审判的单据 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| Senddata.getSendMesg | 返回单一持久化对象（物流订单信息） | |
| Senddata.getInput | 写入单一持久化对象（订单信息） | |
| Imformationbl.refreshMesg | 写入单一持久对象（物流信息） | |
| SendBill.getTimeAndMoney | 得到订单时间和报价 | |

**staffbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Staff.addNewStaff | 语法 | public boolean addNewStaff (StaffVO[] staffs) |
| 前置条件 | 输入合法 |
| 后置条件 | 添加新员工信息 |
| Staff.deleteStaff | 语法 | public boolean deleteStaff (String[] ID) |
| 前置条件 | 输入合法 |
| 后置条件 | 删除员工信息 |
| Staff.changePosition | 语法 | public boolean changePosition (String[] ID, String[] newPositon, String[] newDepartment) |
| 前置条件 | 输入合法 |
| 后置条件 | 更新员工职位信息 |
| Staff.changesalarymodel | 语法 | public boolean changesalarymodel (String[] ID, String[] salaryModel) |
| 前置条件 | 输入合法 |
| 后置条件 | 更新员工薪水策略 |
| Staff. getStaff | 语法 | public StaffVO getStaff (String ID) |
| 前置条件 | 输入合法 |
| 后置条件 | 查看员工信息 |
| Staff.getAllStaff | 语法 | public ArrayList<StaffVO> getAllStaff (String src) |
| 前置条件 | 输入合法 |
| 后置条件 | 查看所有员工信息 |
| Staff.changeStaff | 语法 | public boolean changeStaff (String ID,String competence, String password, String name, String power) |
| 前置条件 | 输入合法 |
| 后置条件 | 修改员工权限 |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Staff.addNewDriver | 语法 | public boolean addNewDriver (DriverPO po) |
| 前置条件 | 输入合法 |
| 后置条件 | 添加新司机信息 |
| Staff.deleteDriver | 语法 | public boolean deleteDriver(String id) |
| 前置条件 | 输入合法 |
| 后置条件 | 删除司机信息 |
| Staff.ChangeDriverInfo | 语法 | public boolean ChangeDriverInfo (String[] ID, String[] IDCardNumber, String[] birthday, boolean[] isBoy, String[] TelNumber, String[] validate) |
| 前置条件 | 输入合法 |
| 后置条件 | 更新司机信息 |
| Staff.getDriver | 语法 | public DriverPO getDriver (String ID) |
| 前置条件 | 输入合法 |
| 后置条件 | 查看司机信息 |
| Staff.getAllDriver | 语法 | **Public** ArrayList<DriverPO> getAll(String str) |
| 前置条件 | 输入合法 |
| 后置条件 | 查看所有司机信息 |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Staff.addNewTruck | 语法 | public boolean addNewTruck(TruckPO po) |
| 前置条件 | 输入合法 |
| 后置条件 | 添加新车辆信息 |
| Staff.deleteTruck | 语法 | public boolean deleteTruck (String id) |
| 前置条件 | 输入合法 |
| 后置条件 | 删除车辆信息 |
| Staff.ChangeTruckInfo | 语法 | public boolean ChangeTruckInfo (TruckPO po) |
| 前置条件 | 输入合法 |
| 后置条件 | 更新车辆信息 |
| Staff.getTruck | 语法 | public TruckPO getTruck (String id) |
| 前置条件 | 输入合法 |
| 后置条件 | 查看车辆信息 |
| Staff.getAllTruck | 语法 | public ArrayList<TruckPO> getAll() |
| 前置条件 | 输入合法 |
| 后置条件 | 查看所有车辆信息 |

## 5.4数据层的分解

### 5.4.1数据层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| Imformationdata | 提供物流信息查询服务 |
| Senddata | 提供快递信息，报价和时间管理相关服务 |
| Judgementdata | 提供单据查看，审判服务 |
| Shipmentdata | 提供装车管理，装运管理和中转接收的数据存储和查看服务 |
| Receivementdata | 提供收件信息存储，接收与派件信息存储与查询的服务 |
| Commoditydata | 提供库存信息的存储，库存信息变更的存储，库存信息的查询等服务 |
| Moneydata | 提供收款单信息的存储查询，结算管理，财务支出等服务 |
| Staffdata | 提供人员信息、账户信息、机构信息、车辆信息、司机信息、用户管理的增删改查服务 |

### 5.4.2数据层模块的接口规范

**commoditydataservice模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| commoditydataservice.insertIndepot | 语法 | **public** **boolean** insertIndepot (StockBill\_In bill) **throws** RemoteException) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 插入一条po数据 |
| commoditydataservice.deleteIndepot | 语法 | **public** **boolean** deleteIndepot (String id) **throws** RemoteException |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除一条po数据 |
| commoditydataservice.changeIndepot | 语法 | **public** **boolean** changeIndepot(StockBill\_In bill) **throws** RemoteException |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新Po数据 |
| commoditydataservice.getIndepot | 语法 | **public** StockBill\_In getIndepot (String id) **throws** RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回查找相应的结果 |
| commoditydataservice.initIndepot | 语法 | **public** **boolean** initIndepot () **throws** RemoteException |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |
| commoditydataservice.insertOutdepot | 语法 | **public** **boolean** insertOutdepot (StockBill\_Out bill) **throws** RemoteException |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 插入一条po数据 |
| commoditydataservice.deleteOutdepot | 语法 | **public** **boolean** deleteOutdepot (String id) **throws** RemoteException |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除一条po数据 |
| commoditydataservice.changeOutdepot | 语法 | **public** **boolean** changeOutdepot (StockBill\_Out bill) **throws** RemoteException |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新Po数据 |
| commoditydataservice.getOutdepot | 语法 | **public** StockBill\_Out getOutdepot (String id) **throws** RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回查找相应的结果 |
| commoditydataservice.initOutdepot | 语法 | **public** **boolean** initOutdepot () **throws** RemoteException |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

**Informationdata的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的接口（供接口）** | | |
| Informationdataservice.getTransMesg | 语法 | Public ArrayList<TransMesgPO> getTransMesg(String id) |
| 前置条件 | 快递信息已经返回 |
| 后置条件 | 无 |
| Informationdataservice.refreshTransMesg | 语法 | **public** **boolean** refreshTransMesg(TransMesgPO po) |
| 前置条件 | 货物接收成功 |
| 后置条件 | 系统保存货物物流信息 |

**moneydataservicce模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| moneydataservice.insert | 语法 | **public** **boolean** insert(Account account) |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po记录 |
| moneydataservice.delete | 语法 | **public** **boolean** delete(String id) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除一条po数据 |
| moneydataservice.change | 语法 | **public** **boolean** change(Account account) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新一条po数据 |
| moneydataservice.get | 语法 | **public** Account get(String id) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| moneydataservice.init | 语法 | **public** **boolean** init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |
| moneydataservice.getPaid | 语法 | **public** PaymentBill get getPaid (String id) |
| 前置条件 | 无 |
| 后置条件 | 返回付款单id |
| moneydataservie.getExpenseid | 语法 | **public** ReceiveMoneyBill get getExpenseid (String id) |
| 前置条件 | 无 |
| 后置条件 | 返回收款单id |
| moneydataservice. getUnjudged | 语法 | **public** ArrayList<PaymentBill> getUnjudged() **throws** RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回成本收益单id |
| moneydataservice.getMember | 语法 | **public** Account getMember (String id) **throws** RemoteException |
| 前置条件 | 账户已在系统中建立账号 |
| 后置条件 | 返回账户的账号名称 |
| moneydataservice.getAll | 语法 | **public** ArrayList<Account> getAll(String id) **throws** RemoteException |
| 前置条件 | 账户已在系统中进行交易 |
| 后置条件 | 返回查看账户的交易记录（交易金额） |

**receivementdataservice 模块的接口规范**

|  |  |  |
| --- | --- | --- |
|  | 提供的服务 | （供接口） |
| receivementdataserv  ice.get | 语法 | **public** ReceiveBill get(String id) |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| receivementdataservice.delete | 语法 | **public** **boolean** delete(String id) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除po |
| receivementdataservice.update | 语法 | **public** **boolean** update(String id) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| receivementdataservice.init | 语法 | public boolean init() |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |
| receivementdataservice.insert | 语法 | **public** **boolean** insert(ReceiveBill bill) |
| 前置条件 | 同样id的po在数据中不存在 |
| 后置条件 | 在数据库中增加一条po记录 |

**Senddata模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的接口（供接口）** | | |
| Senddataservice.getSendMesg | 语法 | **public** OrderBillPO getSendMesg(String id) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| Senddataservice.getInput | 语法 | **public** **boolean** getInput(OrderBillPO bill) |
| 前置条件 | 使用者有写入的权限 |
| 后置条件 | 系统保存写入的订单信息 |

**staffdataservice 模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StaffDataService.insertStaff | 语法 | **public** **boolean** insertStaff (StaffPO po) **throws** RemoteException |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| StaffDataService.insertDriver | 语法 | **public** **boolean** insert(DriverPO po) **throws** RemoteException |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| StaffDataService.insertTruck | 语法 | **public** **boolean** insert(TruckPO po) **throws** RemoteException |
| 前置条件 | 同样的po在数据中不存在 |
| 后置条件 | 增加一条po数据 |
| StaffDataService.getStaff | 语法 | **public** StaffPO getStaff (String id) **throws** RemoteException |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| StaffDataService.getDriver | 语法 | **public** DriverPO get(String id) **throws** RemoteException |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| StaffDataService.getTruck | 语法 | **public** TruckPO get(String id) **throws** RemoteException |
| 前置条件 | 无 |
| 后置条件 | 查找返回相应的结果 |
| StaffDataService.delete | 语法 | **public** **boolean** delete(String id)**throws** RemoteException |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 删除po |
| StaffDataService.change | 语法 | **public** **boolean** change(StaffPO po) **throws** RemoteException |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po |
| StaffDataService. showStaff | 语法 | **public** ArrayList<StaffPO> getAll(String str)**throws** RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有人员信息 |
| StaffDataService. showDriver | 语法 | **public** ArrayList<DriverPO> getAll(String str) **throws** RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有人员信息 |
| StaffDataService. showTruck | 语法 | **public** ArrayList<TruckPO> getAll() **throws** RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回所有人员信息 |
| StaffDataService.init | 语法 | **public** **boolean** init() **throws** RemoteException |
| 前置条件 | 无 |
| 后置条件 | 初始化持久数据 |

# 6.信息视角

## 6.1数据持久化对象

系统的PO类就是对应的实体类，如下所示。

TransMesgPO类包含订单编号、订单列表、货运状态

OrderBillPO是保存及更新订单数据的类，包含订单编号、订单列表、是否审批通过、收件信息（收件编号、收件人、收件日期）

ReceiveBillPO是保存接受与派件信息的类，包含货物到达信息（到达日期、中转单编号、出发地、货物到达状态（损坏、完整、丢失））、派件信息（到达日期、托运订单条形码号、派送员）

ShipmentBillPO是保存装运信息的类，包含装车单信息、中转单信息、中转中心到达单信息

CommodityPO类包含货物信息（快递编号、入库日期、目的地、区号、排号、架号、位号）

AccountBillPO类包含收款单信息（收款日期、收款金额、收款快递员、对应的所有快递订单条形码号）、付款单信息（付款日期、付款金额、付款人、付款账号、条目（租金（按年收）运费（按次计算）人员工资（按月统计）奖励（一次性））、成本收益表信息（收款单、付款单、总利润）、经营情况表信息（收款单、付款单、起始日期）

StaffBillPO类包含车辆信息、司机信息、人员与机构信息账户信息

UserBillPO类包含用户授权信息

## 6.2文件格式

统一用\*.ser格式，比如orderbill.ser