# 快递物流管理系统（EDMS）软件体系结构描述文档

目录

[快递物流管理系统（EDMS）软件体系结构描述文档 1](#_Toc433658797)

[引言 2](#_Toc433658797)

[产品概述 2](#_Toc433658797)

[逻辑视角 2](#_Toc433658797)

[组合视图 4](#_Toc433658797)

[接口视角 10](#_Toc433658797)

[信息视角 10](#_Toc433658797)

**1引言**

**1.1编制目的**

本报告详细完成对快递物流管理系统的概要设计，达到指导详细设计和开发的目的，同时实现和测试人员及用户的沟通。

本报告是面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

**1.2词汇表**

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| EMDS | 快递物流管理系统 |  |
| RMI | 远程接口调用 |  |
| OS | 操作系统 |  |
| JDK | Java 语言的软件开发工具包 |  |

**1.3参考资料**

快递物流管理系统用例图，概念类图，状态图，顺序图

快递物流管理系统用例文档V1.1

快递物流系统软件需求规格说明V1.1

项目实践V3

软件工程与计算(卷二)软件开发的技术基础

度量数据文档

系统测试文档

**2产品概述**

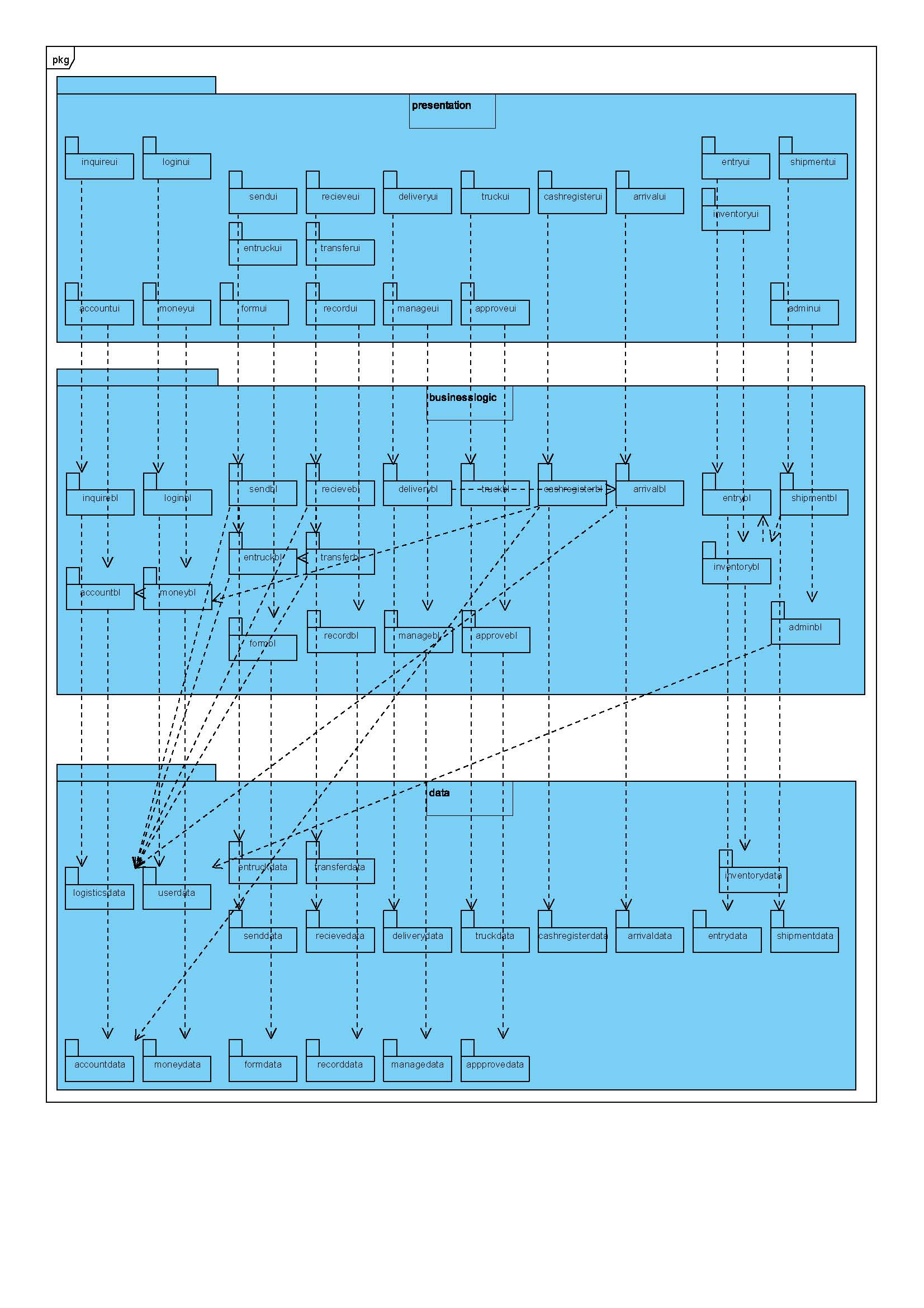
本系统是为本地某公司开发的适应其日常业务的一个快递物流系统，针对不同的人员的不同业务实现各自对应的功能，本系统旨在提高用户体验度，提高用户的体验满意度，提高公司的业务工作的效率，从而达到帮助该公司增加收益，易于扩张的目的。

**3逻辑视角**

快递物流管理系统中，选择了分层体系结构风格，将系统分为了三层（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，业务逻辑包含业务逻辑的实现，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.组合视图**

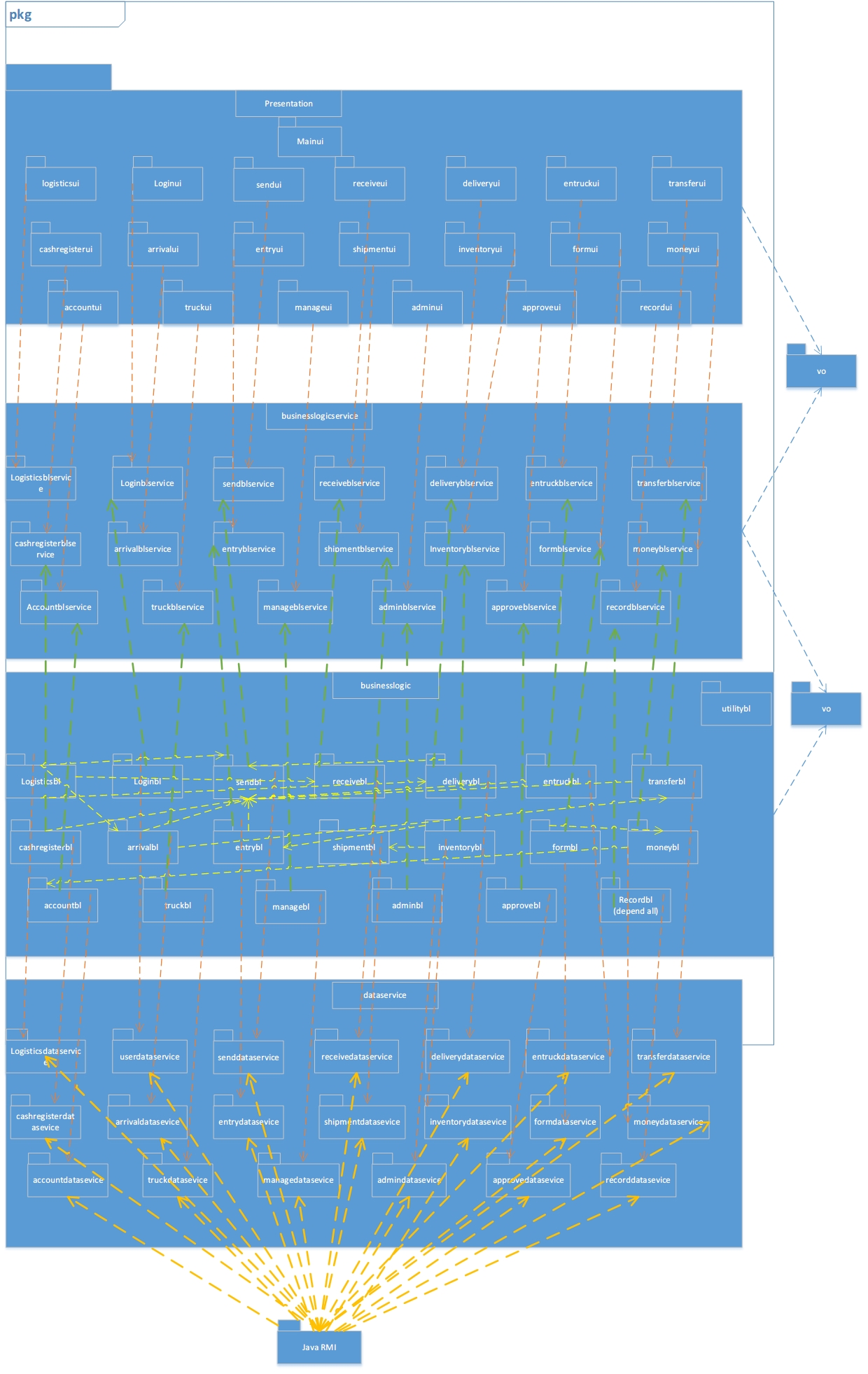
**4.1开发包图**

快递物流系统的最终开发包设计如表1所示

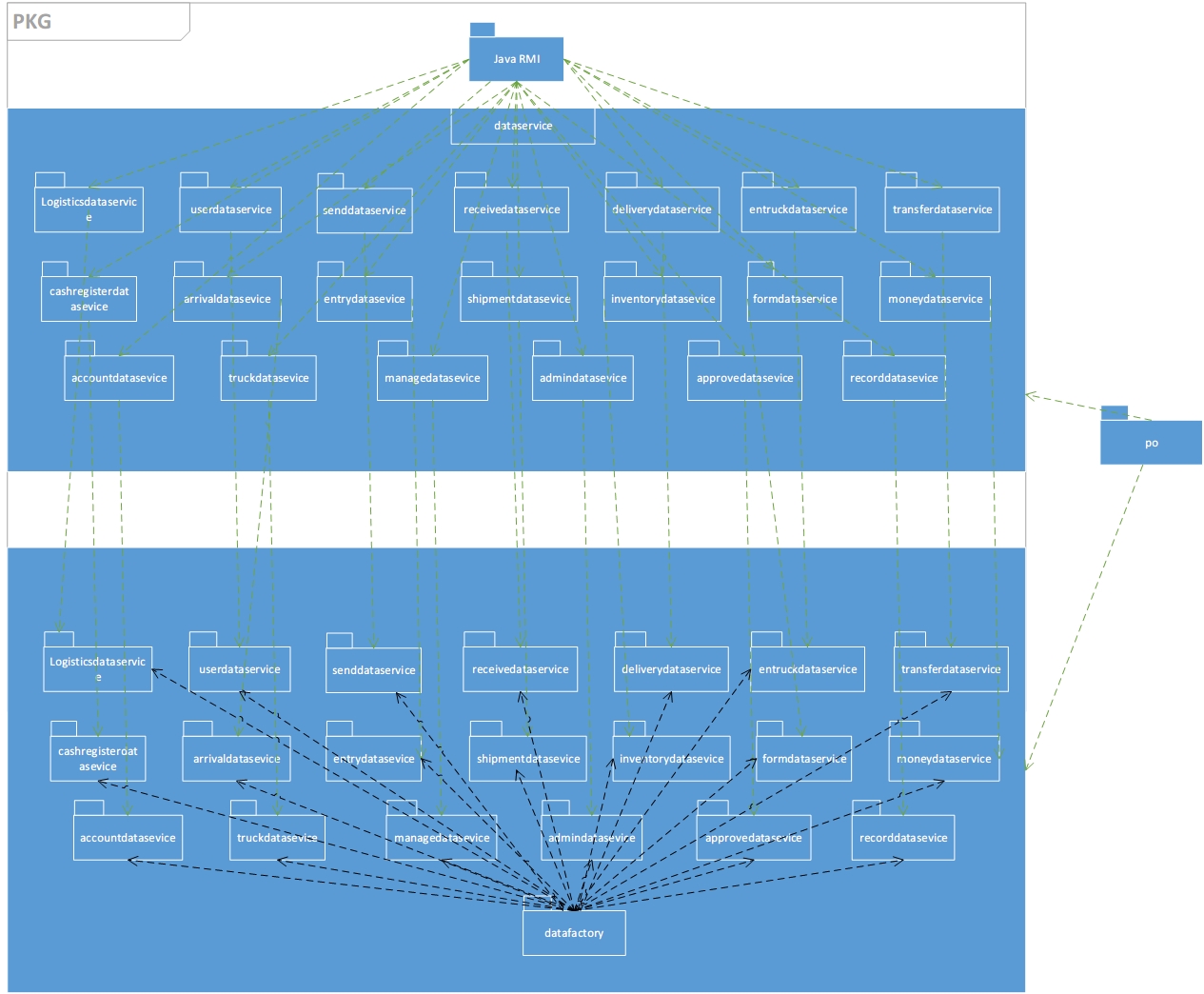
**表1 快递物流系统的最终开发包设计**

|  |  |
| --- | --- |
| **开发（物理）包** | **依赖的其他开发包** |
| mainui | AdminUI,BusinessHallUI,FinanceUI, IntermediateUI ,InventoryUI  LoginUI, LogisticsUI ,LogUI ,ManageUI,界面类库包 |
| AdminUI | vo,界面类库包 |
| BusinessHallUI | vo,ArrivalDataService,CashRegisterDataService,DeliveryDataService,DriverDataService,EntuckDataService,TruckDataService,界面类库包 |
| DeliveryManUI | vo,DeceiveDataService,界面类库包 |
| FinanceUI | Vo,PaymentDataService |
| IntermediateUI | datafactory,po |
| TransferUI | Vo,界面类库包 |
| InventoryUI | Vo,EntryDataService,ShipmentDataService,界面类库包 |
| LoginUI | AdminUI, BusinessHallUI, DeliveryManUI, FinanceUI,界面类库包 |
| LogisticsUI | Vo,LogisticsDataService,界面类库包 |
| LogUI | Vo,FinanceUI,ManageUI,界面类库包 |
| ManageUI | Vo,LogUI,SalaryDataService,界面类库包 |
| adminBL | Po,vo,ManageDataService,logDataService,Java RMI |
| businessHallBL | Po,vo,ManageBL,logBL,LogisticsBL,intermidateDataService,DeliveryBL,businesshallDataService,logDataService,financeDataService,Java RMI |
| deliveryBL | Po,vo,manageBL,logBL,deliveryDataService,ManageDataService,logDataService,Java RMI |
| financeBL | Po,vo,businesshallBL,logBL,financeDataService,logDataService,JAVA RMI |
| intermediateBL | Vo,po,manegeBL,logBL,inventoryBL,logisticsBL,businesshallDataService,intermediateDataService,logDataService,Java RMI |
| inventoryBL | Po,vo,inventoryDataService,Java RMI |
| logBL | Po,vo,logDataService,Java RMI |
| loginBL | Po,vo,businesshallBL,manageDataService,logDataService,Java RMI |
| logisticsBL | Po,vo, businesshallDataService,logisticsDataService,Java RMI |
| manageBL | Po,vo,businesshallBL,financeBL,deliveryBL,logBL,manageDataService,inventoryDataService,Java RMI |
| businesshallDataService | data,po |
| deliveryDataService | data,po |
| financeDataService | data,po |
| intermidateDataService | data,po |
| inventoryDataService | data,po |
| logDataService | data,po |
| logisticsDataService | data,po |
| manageDataService | data,po |

物流管理系统客户端开发包如图3所示，服务器开发包图如图4所示



**图3 物流管理系统客户端开发包**

**图4 物流快递系统服务器端开发包**

**4.2运行时进程**

在物流快递系统中，会有多个客户端进程和一个服务器进程，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，服务器端进程在服务器端机器上运行。

RMI

Client 1

RMI

Client 2

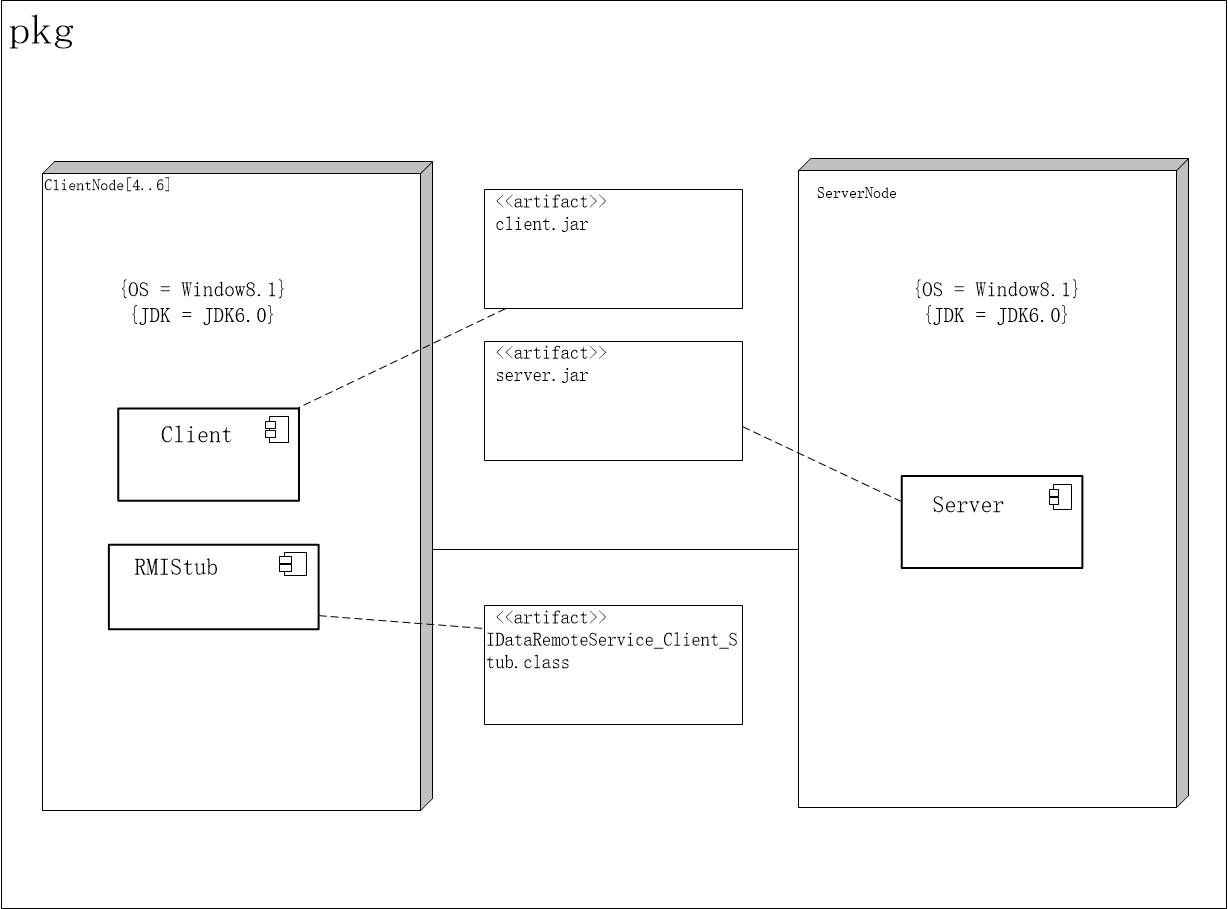
… RMI Server

Client n

**图5 进程图**

**4.3物理部署**

物流快递系统中客户端构件是放在客户端机器上，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JavaRMI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

网络模块

数据层

启动模块

**图7 客户端模块视图 图8 服务器模块视图**

**表2 客户端各层职责**

|  |  |
| --- | --- |
| **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 | 基于窗口的物流快递客户端用户界面 |
| 业务逻辑层 | 对于用户界面的输入进行响应并进行业务逻辑处理 |
| 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。层之间用的接口如表4所示。

**表4 层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| **接口** | **服务器调用方** | **服务器提供方** |
| AdminBLService  ArrivalBLService  CashRegisterBLService  DeliveryBLService  DriverBLService  EntruckBLService  TruckBLService  ReceiveBLService  SendBLService  AccountBLService  FormBLService  MoneyBLService  PaymentBLService  TransferBLService  EntryBLService  InventoryBLService  ShipmentBLService  LogBLService  LoginBLService  InquireBLService  LogisticsBLService  ApproveBLService  InstitutionBLService  SalaryBLService  StaffBLService  StrategyBLService | 客户端展示层 | 客户端业务逻辑层 |
| UserDataService  LogisticsDataService  SendDataService  ReceiveDataService  DeliveryDataService  EntruckDataService  TransferDataService  ArrivalDataService  TruckDataService  EntryDataService  ShipmentDataService  InventoryDataService  AccountDataService  PaymentDataService  DriverDataService  IncomeDataService  PriceConstDataService  SalaryDataService  LogDataService  InstitutionDataService  DataFactory | 客户端业务逻辑层 | 服务器端数据层 |

借用用例来说明层之间的调用，如图9所示。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）。LoginBLService提供了Login界面所需要的所有业务逻辑功能。AccountDataService提供了对TXT文件的增、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 寄件单操作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30个用户界面：登录界面、游客界面、快递员主界面、营业厅业务员主界面、中转中心业务员主界面、中转中心仓库管理人员主界面、财务人员主界面、总经理主界面、管理员主界面、收件单界面、营业厅到达单界面、营业厅装车单界面、车辆司机信息管理界面、派件单界面、收款单界面、中转中心到达单界面、中转中心装车单界面、中转单界面、入库界面、出库界面、库存管理界面、账户信息管理界面、人员机构管理界面、薪水策略制定界面、业务策略制定界面、审批单据界面、银行账户管理界面、收付款管理界面、经营管理界面、操作日志查询界面。界面跳转如图10所示。

车辆司机信息管理

装车单单

收件单界面

账户信息管理

到达单

人员机构管理

派件单

游客界面

快递员界面

收款单

管理员界面

审批单据

营业厅业务员界面

到达单

登录界面

总经理界面

中转中心业务员界面

装车单

业务策略制定

财务人员界面

仓库管理人员界面

中转单

薪水策略制定

银行账户管理

操作日志查询

经营管理

收付款管理

库存管理界面

出库界面

入库界面

**图10 用户界面跳转**

服务器端和客户端的用户界面设计接口是一致的，只是具体的界面不一样。用户界面类如图11所示。

pkg

MainFrame

+init():void

1. **图11 用户界面类**

**图11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，负责登录后界面的显示和界面的跳转 |
| LoginFrame | 登录Frame，负责登录时根据账号权限进行界面跳转 |

**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如表6所示。

**表6 用户界面层模块的接口规范**

|  |  |  |
| --- | --- | --- |
| LoginFrame | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示登录界面 |
| MainFrame | 语法 | setView(Jpanel panel) |
| 前置条件 | 接收到界面跳转指令 |
| 后置条件 | 进行界面跳转和界面显示 |

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** |
| 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**

业务逻辑层包括多个针对界面的业务逻辑处理对象。例如，User对象负责处理登录界面的业务逻辑。业务逻辑层的设计如图12所示

**图12 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 | 数据 |
| LogisticsBL | 负责实现与快递物流信息查询所需要的服务 |
| LoginBL | 负责实现与登录界面所需要的服务 |
| SendBL | 负责实现与寄件单界面所需要的服务 |
| ReceiveBL | 负责实现与收件单界面所需要的服务 |
| DeliveryBL | 负责实现与派件单界面所需要的服务 |
| EntruckBL | 负责实现与装车单界面所需要的服务 |
| TransferBL | 负责实现与中转单界面所需要的服务 |
| CashregisterBL | 负责实现与收款单界面所需要的服务 |
| ArrivalBL | 负责实现与到达单界面所需要的服务 |
| EntryBL | 负责实现与入库界面所需要的服务 |
| ShipmentBL | 负责实现与出库界面所需要的服务 |
| InventoryBL | 负责实现库存管理界面所需要的服务 |
| FormBL | 负责实现与查看报表界面所需要的服务 |
| InitBillBL | 负责实现与期初建账界面所需要的服务 |
| PaymentBL | 负责实现与付款单界面所需要的服务 |
| MoneyBL | 负责实现与查看收款单和新建付款单所需要的服务 |
| AccountBL | 负责实现与银行账户操作所需要的服务 |
| TruckBL | 负责实现与车辆信息管理界面所需要的服务 |
| ManageBL | 负责实现与总经理进行人员策略管理界面所需要的服务 |
| StrategyBL | 负责实现与总经理进行业务策略制定界面所需要的服务 |
| SalaryBL | 负责实现与总经理进行薪水策略制定界面所需要的服务 |
| AdminBL | 负责实现与管理员界面所需要的服务 |
| ApproveBL | 负责实现与总经理审批单据所需要的服务 |
| InstitutionBL | 负责实现与总经理进行机构管理界面所需要的服务 |
| LogBL | 负责实现与查看操作日志界面所需要的服务 |
| DriverBL | 负责实现与司机信息管理界面所需要的服务 |

**5.3.2业务逻辑层模块的接口规范**

**表9 logisticsbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Logistics.add** | **语法** | **Public List<LogisticsVO> getLogisticsList(String id) throws RemoteException;** |
| **前置条件** | **快递单号符合输入规则** |
| **后置条件** | **开始记录新的快递单的物流信息** |
| **Logistics.update** | **语法** | **Public String update(String id，String message)** |
|  | **前置条件** | **需要物流信息更新时** |
|  | **后置条件** | **根据传入的快递单号更新物流信息** |
| **Logistics.search** | **语法** | **public logisticsVO find(String id)** |
|  | **前置条件** | **无** |
|  | **后置条件** | **返回一个对应该编号的快递单的物流信息的VO** |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| **LogisticsDataService.find String id)** | **按ID进行查找返回相应的LogisticsPO结果** | |
| **LogisticsDataService.update(LogisticsPO po)** | **更新一个logisticsPO** | |
| **LogisticsDataService.getLogisticsList** | **将数据层的物流信息的列表获取到** | |

**表10 loginbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Login.login** | **语法** | **public boolean login(String id,String password) throws RemoteException;** |
| **前置条件** | **password 和 id 符合输入规则** |
| **后置条件** | **查找是否存在相应的User，根据输入的password返回登录的验证结果** |
| **需要的服务（需接口）** | | |
| 服务名 | | 服务 |
| **LogDataSerivice.add(LogPO po)** | | **增加一条log信息，并更新对应的TXT文件** |
| **UserDataService.find(long id)** | | **按ID进行查找返回相应的UserPO结果** |

**表11 sendbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **send.add** | 语法 | **public String add(SendVO VO) throws RemoteException ;** |
| 前置条件 | **启动一次寄件单填写** |
| 后置条件 | **在一次寄件单填写回合中，增加货物信息** |
| **send.getFare** | 语法 | **public double getFare(String departure, String destination, String packType, String TransferType);** |
| 前置条件 | **已填写寄件单的相关的填写** |
| 后置条件 | **根据相关信息生成运费** |
| **send.delete** | 语法 | **public String delete();** |
| 前置条件 | **目标寄件单存在** |
| 后置条件 | **删除目标寄件单，更新TXT文件** |
| **send.search** | 语法 | **public SendVO search(SendVO sVO);** |
| 前置条件 | **无** |
| 后置条件 | **返回在VO列表中查找结果** |
| **send.modify** | 语法 | **public void modify(SendPO Vo)** |
| 前置条件 | **修改了选中的寄件单** |
| 后置条件 | **修改数据层的数据并更新txt文件** |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| **ManageDataService.returnStrategy ()** | **查找返回相应的ManageStrategyPO结果,获取价格常量** | |
| **SendDataService.delete(SendPO po)** | **删除一个SendPO，并更新对应的txt文件** | |
| **SendDataService.modify (SendPO po)** | **修改一个SendPO，并更新对应的txt文件** | |
| **SendDataService.find (String id)** | **根据快递的10位单号来查找对应的寄件单** | |
| **SendDataService.add (SendPO po)** | **新增一个SendPO，并更新对应的txt文件** | |
| **LogisticsDataService.add(LogisticsPO po)** | **增加一个LogisticsPO** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息** | |

**表12 receivebl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **receive.add** | **语法** | **public String add(ReceiveVO rVO);** |
| **前置条件** | **启动一次收件单填写** |
| **后置条件** | **在一次收件单填写回合中，填写收件单信息** |
| **receive.delete** | **语法** | **public String delete(ReceiveVO rVO);** |
| **前置条件** | **目标收件单存在** |
| **后置条件** | **删除目标收件单，更新TXT文件** |
| **receive.search** | **语法** | **public ReceiveVO search(String id);** |
| **前置条件** | **无** |
| **后置条件** | **返回在VO列表中查找结果** |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| **LogisticsDataService.delete**  **(ReceivePO po)** | **删除一个ReceivePO，并更新对应的txt文件** | |
| **LogisticsDataService.find**  **(String id)** | **根据id查找一个ReceivePO** | |
| **LogisticsDataService.modify**  **(ReceivePO po)** | **修改一个ReceivePO，并更新对应的txt文件** | |
| **LogisticsDataService.add**  **(ReceivePO po)** | **新增一个ReceivePO，并更新对应的txt文件** | |
| **LogisticsDataService.update (LogisticsPO po)** | **更新一个LogisticsPO，并更新对应的txt文件** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表13 deliverybl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Delivery.add** | **语法** | **public String add(DeliveryVO VO) throws RemoteException;** |
| **前置条件** | **启动一次派件单填写** |
| **后置条件** | **在一次派件单填写回合中，填写派件单信息** |
| **Delivery.delete** | **语法** | **public String delete(DeliveryVO VO) throws RemoteException;** |
| **前置条件** | **目标派件单存在** |
| **后置条件** | **删除目标派件单，更新TXT文件** |
| **Delivery.search** | **语法** | **public DeliveryVO search(String id) throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **返回在VO列表中查找结果** |
| **Delivery.modify** | **语法** | **public void modify(DeliveryVO dVO) throws RemoteException;** |
| **前置条件** | **修改一张派件单** |
| **后置条件** | **修改派件单信息，并更新txt文件** |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| **DliveryDataService.add(deliveryPO po)** | **新增一个deliveryPO，并更新对应的txt文件** | |
| **DliveryDataService.delete(deliveryPO po)** | **删除一个deliveryPO，并更新对应的txt文件** | |
| **DliveryDataService.modify(deliveryPO po)** | **修改一个deliveryPO，并更新对应的txt文件** | |
| **DliveryDataService.find(String id)** | **根据单号ID查找一个deliveryPO** | |
| **LogisticsDataService.update (LogisticsPO po)** | **更新一个LogisticsPO，并更新对应的txt文件** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表14 entruckbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Entruck.add** | 语法 | **public String add(EntruckVO eVO) throws RemoteException;** |
| 前置条件 | **启动一次装车单填写** |
| 后置条件 | **在一次装车单填写回合中，填写收件单信息** |
| **Entruck.delete** | 语法 | **public String delete(EntruckVO eVO) throws RemoteException;** |
| 前置条件 | **目标装车单存在** |
| 后置条件 | **删除目标装车单，更新TXT文件** |
| **Entruck.search** | 语法 | **public EntruckVO search(String id) throws RemoteException;** |
| 前置条件 | **无** |
| 后置条件 | **返回在VO列表中查找结果** |
| **Entruck.modify** | 语法 | **public EntruckVO search(String id) throws RemoteException;** |
| 前置条件 | **无** |
| 后置条件 | **返回在VO列表中查找结果** |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| **EntruckDataService.add(entruckPO po)** | **新增一个entruckPO，并更新对应的txt文件** | |
| **EntruckDataService.modify(entruckPO po)** | **修改一个entruckPO，并更新对应的txt文件** | |
| **EntruckDataService.delete(entruckPO po)** | **删除一个entruckPO，并更新对应的txt文件** | |
| **EntruckDataService.find(entruckPO po)** | **查找一个entruckPO** | |
| **LogisticsDataService.update (LogisticsPO po)** | **更新一个LogisticsPO，并更新对应的txt文件** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表15 transferbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Transfer.add** | 语法 | **Public void add(String id)** |
| 前置条件 | **启动一次中转单填写** |
| 后置条件 | **在一次中转单填写回合中，填写中转单信息** |
| **Transfer.delete** | 语法 | **Public void delete(TransferVO po)** |
| 前置条件 | **目标中转单存在** |
| 后置条件 | **删除一条中转单信息，更新对应的txt文件** |
| **Tansfer.modify** | 语法 | **public void modify(TranferPO PO) throws RemoteException;** |
| 前置条件 | **目标中转单存在** |
| 后置条件 | **修改目标中转单，更新对应的TXT文件** |
| **Transfer.search** | 语法 | **public EntruckPO search(String id) throws RemoteException;** |
| 前置条件 | **无** |
| 后置条件 | **返回在VO列表中查找结果** |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| **LogisticsDataService.add(transferPO po)** | **新增一个transferPO，并更新对应的txt文件** | |
| **LogisticsDataService.delete**  **(transferPO po)** | **删除一个transferPO，并更新对应的txt文件** | |
| **LogisticsDataService.modify(transferPO po)** | **修改一个transferPO，并更新对应的txt文件** | |
| **LogisticsDataService.add(String transID)** | **根据中转单的单号，查找一个transferPO** | |
| **LogisticsDataService.update (LogisticsPO po)** | **更新一个LogisticsPO，并更新对应的txt文件** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表16 cashregisterbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Cashregister.add** | **语法** | **public String add(CashRegisterVO crVO) throws RemoteException;** |
| **前置条件** | **启动一次收款单操作** |
| **后置条件** | **在一次收款单操作中，填写收款单信息** |
| **Cashregister.delete** | **语法** | **public void delete(CashRegisterVO crVO)** |
| **前置条件** | **目标收款单存在** |
| **后置条件** | **删除目标收款单，更新TXT文件** |
| **Cashregister.find** | **语法** | **public List<CashRegisterVO> find(Date d)** |
| **前置条件** | **无** |
| **后置条件** | **根据日期查找返回一个收款单的列表** |
| **CashRegister.find** | **语法** | **public List<CashRegisterVO> find(String id)** |
| **前置条件** | **无** |
| **后置条件** | **根据机构编号查找返回一个收款单的列表** |
| **Cashregister.find** | **语法** | **public List<CashRegisterVO> find(Date start,Date end)** |
| **前置条件** | **无** |
| **后置条件** | **根据日期区间查找返回一个收款单的列表** |
| **CashRegister.find** | **语法** | **public List<CashRegisterVO> find()** |
| **前置条件** | **无** |
| **后置条件** | **返回截止到当前日期的所有的收款单列表** |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| **CashregisterDataService.add(cashregisterPO po)** | **新增一个cashregisterPO** | |
| **CashRegisterDataService.getCashList** | **获取所有收款单的列表** | |
| **CashregisterDataService.add(cashregisterPO po)** | **删除一个cashregisterPO** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息** | |

**表17 arrivalbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Arrival.add** | **语法** | **public String add(ArrivalVO arrivalVO) throws RemoteException, ClientInitException** |
| **前置条件** | **开始一个到达单操作** |
| **后置条件** | **在一次到达单操作中，填写到达单信息** |
| **Arrival.search** | **语法** | **public ArrivalVO search(String transferID) throws RemoteException** |
| **前置条件** | **无** |
| **后置条件** | **根据ID返回一个ArrivalVO** |
| **Arrival.delete** | **语法** | **public String delete(ArrivalVO arrivalVO) throws RemoteException** |
| **前置条件** | **目标到达单存在** |
| **后置条件** | **删除目标到达单，更新TXT文件** |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| **ArrivalDataService.add (arrivalPO po)** | **新增一个arrivalPO** | |
| **ArrivalDataService.delete (arrivalPO po)** | **删除一个arrivalPO** | |
| **ArrivalDataService.modify (arrivalPO po)** | **修改一个arrivalPO** | |
| **ArrivalDataService.find (String id)** | **根据快递单号id返回一个到达单** | |
| **LogisticsDataService.update (logisticaPO po)** | **更新一个logisticsPO** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**18 entrybl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Entry.add** | **语法** | **public String add(EntryVO vo) throws RemoteException** |
| **前置条件** | **启动一次入库单填写** |
| **后置条件** | **在一次入库单填写回合中，填写入库单信息** |
| **Entry.delete** | **语法** | **public String delete(EntryVO vo) throws RemoteException** |
| **前置条件** | **目标入库单存在** |
| **后置条件** | **删除目标入库单，更新TXT文件** |
| **Entry.find** | **语法** | **public List<EntryPO> getEntryList(String institutionID) throws RemoteException** |
| **前置条件** | **无** |
| **后置条件** | **根据ID查找EntryPO存入list中，返回该list** |
| **Entry.modify** | **语法** | **public void modify（EntryPO）** |
| **前置条件** | **需要进行库存调整** |
| **后置条件** | **修改对应的入库单的数据** |
|  | | |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| **EntryDataService.add** | **新增一个entryPO，并更新对应的txt文件** | |
| **EntryDataService.delete** | **删除一个entryPO，并更新对应的txt文件** | |
| **EntryDataService.modify** | **更新一个entryPO，并更新对应的txt文件** | |
| **EntryDataService.getEntryList** | **获取所有的入库单列表** | |
| **Logistics.update** | **更新logistics** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表19 shipmentbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Shipment.add** | **语法** | **public String add(ShipmentVO vo) throws RemoteException;** |
| **前置条件** | **启动一次出库单操作** |
| **后置条件** | **在一次出库单填写回合中，填写出库单信息** |
| **Shipment.delete** | **语法** | **public String delete(ShipmentVO vo) throws RemoteException;** |
| **前置条件** | **目标出库单存在** |
| **后置条件** | **删除目标出库单，更新TXT文件** |
| **Shipment.find** | **语法** | **public ShipmentVO find(String id) throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **根据ID返回查找结果** |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| **ShipmentDataService.add** | **新增一个shipmentPO，并更新对应的txt文件** | |
| **ShipmentDataService.delete** | **删除一个shipmentPO，并更新对应的txt文件** | |
| **ShipmentDataService.getShipmentList** | **获取所有的出库单的列表** | |
| **Logistics.update** | **更新logistics** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表20 inventorybl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Inventory.adjust** | **语法** | **public String adjust(EntryVO vo) throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **为入库的货物重新分配位置** |
| **Inventory.getInventory** | **语法** | **public InventoryVO getInventory(String institutionID);** |
| **前置条件** | **无** |
| **后置条件** | **根据机构ID返回查找仓库结果** |
| **Inventory.setBusy** | **语法** | **public InventoryVO setBusy(EntryPO po)throws RemoteException** |
| **前置条件** | **填写一张入库单成功** |
| **后置条件** |  |
| **Inventory.modify** | **语法** | **public String modify(InventoryPO po)throws RemoteException** |
| **前置条件** | **总经理要修改库存的信息** |
| **后置条件** | **库存的信息修改，并更新** |
| **Inventory.setFree** | **语法** | **public String setFree(ShipmentPO po)throws RemoteException** |
| **前置条件** | **快递已经出库** |
| **后置条件** | **设置状态为空闲** |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| **inventoryDataService.modify** | **更新一个inventoryPO** | |
| **inventoryDataService.add** | **增加一个inventoryPO** | |
| **inventoryDataService.delete** | **删除一个inventoryPO** | |
| **inventoryDataService.getList** | **获得所有InventoryPO的列表** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表21 formbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Form.income** | **语法** | **public List<IncomeVO> getIncomes();** |
| **前置条件** | **目标收款单存在** |
| **后置条件** | **将目标收款单放入list，返回该list** |
| **Form.payment** | **语法** | **public List<PaymentVO> getPayments();** |
| **前置条件** | **目标付款单存在** |
| **后置条件** | **将目标付款单放入list，返回该list** |
| **Form.findIncome** | **语法** | **public List<IncomeVO> getIncomes(Date start, Date end);** |
| **前置条件** | **时间设置正确** |
| **后置条件** | **将该段时间内收款单放入list，返回该list** |
| **Form.findPayment** | **语法** | **public List<PaymentVO> getPayments(Date start, Date end);** |
| **前置条件** | **时间设置正确** |
| **后置条件** | **将该段时间内付款单放入list，返回该list** |
| **Form.checkDate** | **语法** | **public String isValid(String start, String end);** |
| **前置条件** | **无** |
| **后置条件** | **返回时间设置是否正确** |
| **Form.exportCost** | **语法** | **public void exportCostForm();** |
| **前置条件** | **无** |
| **后置条件** | **导出支出表** |
| **Form.exportRun** | **语法** | **public void exportRunForm(Date start, Date end);** |
| **前置条件** | **无** |
| **后置条件** | **导出经营状况表** |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| **Cashregister.getCashList** | **多种重载方式得到cashregisterVO的list，详见cashregister的接口说明** | |
| **payment.getPayList** | **多种重载方式得到paymentVO的list，详见cashregister的接口说明** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表23 accountb**l**模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Account.add** | **语法** | **public String addAccount(String name, String accountID)throws RemoteException ;** |
| **前置条件** | **启动一次账户添加** |
| **后置条件** | **在一次账户添加操作中，填写账户信息** |
| **Account.delete** | **语法** | **public boolean deleteAccount(AccountVO vo)throws RemoteException ;** |
|  | **前置条件** | **目标账户存在** |
|  | **后置条件** | **删除该账户，更新TXT文件** |
| **Account.search** | **语法** | **public List<AccountVO> searchAccount(String key) throws RemoteException;** |
|  | **前置条件** | **无** |
|  | **后置条件** | **根据string返回查找结果** |
| **Account.modify** | **语法** | **public String modifyAccount(AccountVO vo) throws RemoteException ;** |
|  | **前置条件** | **目标账户存在** |
|  | **后置条件** | **修改该账户，更新TXT文件** |
| **Account.getAccounts** | **语法** | **public List<AccountVO> getAccounts();** |
|  | **前置条件** | **无** |
|  | **后置条件** | **显示所有的账户** |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| **accountDataService.getAccounts** | **得到所有的银行账户信息的列表** | |
| **accountDataService.delete** | **删除一个银行账户，并更新对应的txt文件** | |
| **accountDataService.add** | **增加一个accountPO，并更新对应的txt文件** | |
| **accountDataService.modify** | **更新一个accountPO，并更新对应的txt文件** | |
| **accountDataService.check** | **根据ID索引返回一个accountPO** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表24 truckbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Truck.find** | **语法** | **public TruckVO find(String id);** |
| **前置条件** | **无** |
| **后置条件** | **根据ID返回查找结果** |
| **Truck.add** | **语法** | **public String add(TruckVO vo);** |
| **前置条件** | **启动一次车辆信息填写** |
| **后置条件** | **在车辆信息填写操作中，填写车辆信息** |
| **Truck.delete** | **语法** | **public String delete(TruckVO vo);** |
| **前置条件** | **目标车辆信息存在** |
| **后置条件** | **删除此条车辆信息，更新TXT文件** |
| **Truck.modify** | **语法** | **public String modify(TruckVO vo);** |
| **前置条件** | **目标车辆信息存在** |
| **后置条件** | **修改该车辆信息，更新TXT文件** |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| **truckDataService.add** | **增加一条truckPO** | |
| **truckDataService.delete** | **删除一条truckPO** | |
| **truckDataService.modify** | **修改一条truckPO** | |
| **truckDataService.find** | **根据ID查找返回一个truckPO** | |
| **truckDataService.getList** | **从数据层获取全部的truckPO的列表** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表25 driverbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Drive.find** | **语法** | **public DriverVO find(String id);** |
| **前置条件** | **无** |
| **后置条件** | **根据ID返回查找结果** |
| **Driver.add** | **语法** | **public String addDriver(DriverVO vo);** |
|  | **前置条件** | **启动一次司机信息填写** |
|  | **后置条件** | **在司机信息填写操作中，填写司机信息** |
| **Driver.delete** | **语法** | **public String delete(DriverVO vo);** |
|  | **前置条件** | **目标司机信息存在** |
|  | **后置条件** | **删除此条司机信息，更新TXT文件** |
| **Driver.modify** | **语法** | **public String modify(DriverVO vo);** |
|  | **前置条件** | **目标司机信息存在** |
|  | **后置条件** | **修改该司机信息，更新TXT文件** |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| **DriverDataService.getList** | **从数据底层获取司机的信息po列表** | |
| **DriverDataService.find** | **根据ID查找对应司机信息po** | |
| **DriverDataService.add** | **数据层增加一个driverPO，更新txt文件** | |
| **DriverDataService.delete** | **数据层删除传入的对应的driverPO，更新txt文件** | |
| **DriverDataService.modify** | **数据层修改对应的driverPO的相关信息，并更新txt文件** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |
|  |  | |

**表26 adminbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Admin.search** | 语法 | **public List<UserVO> search(String Key);** |
| 前置条件 | **增加一条用户信息** |
| 后置条件 | **根据传入信息增加用户信息** |
| **Admin.modify** | 语法 | **public String modify(String identityID,String newPassWord) throws RemoteException;** |
|  | 前置条件 | **删除一条用户信息** |
|  | 后置条件 | **根据索引信息删除此条用户信息** |
| **Admin.getUser** | 语法 | **public List<UserVO> getUserListByInsID(String institutionID);** |
|  | 前置条件 | **更新一条用户信息** |
|  | 后置条件 | **根据索引信息更新此条用户信息** |
| **Admin.getInstitution** | **语法** | **public List<InstitutionVO> getInsList();** |
|  | **前置条件** | **结束一次用户信息更改操作** |
|  | **后置条件** | **结束此次用户信息更改操作，持久化更新涉及领域的对象的数据** |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| **UserDataService.update** | **根据信息更新一个UserPO** | |
| **UserDataService.find** | **根据ID查找返回一个userPO** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表27 approvebl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Approve.unarrival** | **语法** | **public List<ArrivalVO> getUnapprovedArrival()** |
| **前置条件** | **无** |
| **后置条件** | **得到所有未通过审批的到达单** |
| **Approve.arrival** | **语法** | **public void setApprovedArrival(ArrivalVO vo)** |
| **前置条件** | **无** |
| **后置条件** | **通过当前到达单** |
| **Approve.arrivalAll** | **语法** | **public void setAllApprovedArrival()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过所有到达单** |
| **Approve.unincome** | **语法** | **public List<IncomeVO> getUnapprovedIncome()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **得到所有未通过审批的收款单** |
| **Approve.income** | **语法** | **public void setApprovedIncome(IncomeVO vo)** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过当前收款单** |
| **Approve.incomeAll** | **语法** | **public void setAllApprovedIncome()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过所有收款单** |
| **Approve.undelivery** | **语法** | **public List<DeliveryVO> getUnapprovedDelivery()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **得到所有未通过审批的派件单** |
| **Approve.delivery** | **语法** | **public void setApprovedDelivery(DeliveryVO vo)** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过当前派件单** |
| **Approve.deliveryAll** | **语法** | **public void setAllApprovedDelivery()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过所有派件单** |
| **Approve.unentry** | **语法** | **public List<EntryVO> getUnapproveEntry()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **得到所有未通过审批的入库单** |
| **Approve.enrty** | **语法** | **public void setApprovedEntry(EntryVO vo)** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过当前入库单** |
| **Approve.entryAll** | **语法** | **public void setAllApprovedEntry()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过所有入库单** |
| **Approve.unpayment** | **语法** | **public List<PaymentVO> getUnapprovePayment()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **得到所有未通过审批的付款单** |
| **Approve.payment** | **语法** | **public void setApprovedPayment(PaymentVO vo)** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过当前付款单** |
| **Approve.paymentAll** | **语法** | **public void setAllApprovedPayment()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过所有付款单** |
| **Approve.unreceive** | **语法** | **public List<ReceiveVO> getUnapproveReceive()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **得到所有未通过审批的收件单** |
| **Approve.receive** | **语法** | **public void setApprovedReceive(ReceiveVO vo)** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过当前收件单** |
| **Approve.receiveAll** | **语法** | **public void setAllApprovedReceive()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过所有收件单** |
| **Approve.unshipment** | **语法** | **public List<ShipmentVO> getUnapproveShipment()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **得到所有未通过审批的收款单** |
| **Approve.shipment** | **语法** | **public void setApprovedShipment(ShipmentVO vo)** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过当前收款单** |
| **Approve.shipmentAll** | **语法** | **public void setAllApprovedShipment()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过所有收款单** |
| **Approve.untransfer** | **语法** | **public List<TransferVO> getUnapproveTransfer(){** |
|  | **前置条件** | **无** |
|  | **后置条件** | **得到所有未通过审批的中转单** |
| **Approve.transfer** | **语法** | **public void setApprovedTransfer(TransferVO vo)** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过当前中转单** |
| **Approve.transferAll** | **语法** | **public void setAllApprovedTransfer()** |
|  | **前置条件** | **无** |
|  | **后置条件** | **通过所有中转单** |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |
| **TransferDataService.getTransList** | **获取所有的中转单po的list** | |
| **TransferDataService.approve** | **审批某张中转单** | |
| **TransferDataService.approveall** | **审批所有的中转单** | |
| **PaymentDataService.getTransList** | **获取所有的付款单po的list** | |
| **PaymentDataService.approve** | **审批某张付款单** | |
| **PaymentDataService.approveall** | **审批所有的付款单** | |
| **IncomeDataService.getTransList** | **获取所有的收款单po的list** | |
| **IncomeDataService.approve** | **审批某张收款单** | |
| **IncomeDataService.approveall** | **审批所有的收款单** | |
| **SendDataService.getTransList** | **获取所有的寄件单po的list** | |
| **SendDataService.approve** | **审批某张寄件单** | |
| **SendDataService.approveall** | **审批所有的寄件单** | |
| **EntruckDataService.getTransList** | **获取所有的装车单po的list** | |
| **EntruckDataService.approve** | **审批某张装车单** | |
| **EntruckDataService.approveall** | **审批所有的装车单** | |
| **ArrivalDataService.getTransList** | **获取所有的到达单po的list** | |
| **ArrivalDataService.approve** | **审批某张到达单** | |
| **ArrivalDataService.approveall** | **审批所有的到达单** | |
| **DeliveryDataService.getTransList** | **获取所有的派件单po的list** | |
| **DeliveryDataService.approve** | **审批某张派件单** | |
| **DeliveryDataService.approveall** | **审批所有的派件单** | |
| **ReceiveDataService.getTransList** | **获取所有的收件单po的list** | |
| **ReceiveDataService.approve** | **审批某张收件单** | |
| **ReceiveDataService.approveall** | **审批所有的收件单** | |
| **EntryDataService.getTransList** | **获取所有的入库单po的list** | |
| **EntryDataService.approve** | **审批某张入库单** | |
| **EntryDataService.approveall** | **审批所有的入库单** | |
| **ShipemntDataService.getTransList** | **获取所有的出库单po的list** | |
| **ShipmentDataService.approve** | **审批某张出库单** | |
| **ShipmentDataService.approveall** | **审批所有的出库单** | |

**表28 logbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Log.check** | **语法** | **public List<LogVO> getAllLogList();** |
| **前置条件** | **无** |
| **后置条件** | **显示操作日志** |
| **Log.date** | **语法** | **public List<LogVO> getLogListOfDate(Date d);** |
|  | **前置条件** | **时间设置正确** |
|  | **后置条件** | **显示该段时间内所有的日志** |
| **Log.user** | **语法** | **public List<LogVO> getLogListOfUser(String userID);** |
|  | **前置表情** | **UserID 输入正确** |
|  | **后置条件** | **显示该员工的操作日志** |
| **Log.add** | **语法** | **public String add(LogVO vo) throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **添加一条操作日志** |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| **LogDataService.getLogList** | **获取所有的log信息的列表** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表29 paymentbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Payment.add** | **语法** | **public void addPayment(PaymentVO vo)throws RemoteException;** |
| **前置条件** | **启动一次付款单填写** |
| **后置条件** | **在付款单填写操作中，填写付款单信息** |
| **Payment.findByDate** | **语法** | **public List<PaymentPO> findByDate(Date d)throws RemoteException;** |
|  | **前置条件** | **时间设置正确** |
|  | **后置条件** | **显示改天付款单信息** |
| **Payment.betweenDate** | **语法** | **public List<PaymentPO> betweenDate(Date start, Date end)throws RemoteException;** |
|  | **前置表情** | **时间设置正确** |
|  | **后置条件** | **显示该时间段付款单信息** |
| **Payment.delete** | **语法** | **public void addPayment(PaymentVO vo)throws RemoteException;** |
|  | **前置条件** | **启动一次付款单填写** |
|  | **后置条件** | **在付款单填写操作中，填写付款单信息** |
| **Payment.modify** | **语法** | **public List<PaymentPO> findByDate(Date d)throws RemoteException;** |
|  | **语法** | **public void addPayment(PaymentVO vo)throws RemoteException;** |
|  | **前置条件** | **启动一次付款单填写** |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| **PaymentDataService.add** | **增加一个付款单信息paymentPO，并更新对应的txt文件** | |
| **PaymentDataService.delete** | **删除一个付款单信息paymentPO，并更新对应的txt文件** | |
| **PaymentDataService.modify** | **修改一个付款单信息paymentPO，并更新对应的txt文件** | |
| **PaymentDataService.getPayList** | **获取所有的付款单信息paymentPO的列表** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表30 institutionbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Institution.show** | 语法 | **public List<InstitutionVO> show();** |
| 前置条件 | **无** |
| 后置条件 | **显示所有机构** |
| **Institution.add** | **语法** | **public String addIns(InstitutionVO insVO) throws RemoteException;** |
|  | **前置条件** | **启动机构增加** |
|  | **后置条件** | **增加机构，更新TXT文件** |
| **Institution.copy** | **语法** | **public String addIns(InstitutionVO vo,InventoryVO ivo) throws RemoteException;** |
|  | **前置表情** | **启动中转中心对应仓库增加** |
|  | **后置条件** | **增加一个仓库，更新TXT文件** |
| **Institution.delete** | **语法** | **public String deleteIns(InstitutionVO insVO) throws RemoteException;** |
|  | **前置条件** | **目标机构存在** |
|  | **后置条件** | **删除该机构，更新TXT文件** |
| **Institution.modify** | **语法** | **public String modify(String id,String newName) throws RemoteException;** |
|  | **前置条件** | **目标机构存在** |
|  | **后置条件** | **修改该机构信息，更新TXT文件** |
| **Institution.modifyInv** | **语法** | **public String modify(String inventoryID,InventoryVO vo) throws RemoteException;** |
|  | **前置条件** | **目标仓库存在** |
|  | **后置条件** | **修改该仓库信息，更新TXT文件** |
| **Institution.searchIns** | **语法** | **public InstitutionVO searchInstitution(String institutionID);** |
|  | **前置条件** | **无** |
|  | **后置条件** | **根据ID返回机构查找结果** |
| **Institution.searchInv** | **语法** | **public InventoryVO searchInventory(String inventoryID) throws RemoteException;** |
|  | **前置条件** | **无** |
|  | **后置条件** | **根据ID返回仓库查找结果** |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| **InstitutionDataService.add** | **增加一个机构institutionPO 并更新对应的txt文件** | |
| **InstitutionDataService.delete** | **删除一个机构institutionPO 并更新对应的txt文件** | |
| **InstitutionDataService.modify** | **修改一个机构institutionPO 并更新对应的txt文件** | |
| **InstitutionDataService.getInsList** | **得到所有的insitituionPO的列表** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表31 salarybl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Salary.getSalary** | 语法 | **public SalaryVO getSalary() throws RemoteException;** |
| 前置条件 | **无** |
| 后置条件 | **得到目标SalaryVO** |
| **Salary.modify** | **语法** | **public String modify(SalaryVO vo) throws RemoteException;** |
|  | **前置条件** | **无** |
|  | **后置条件** | **修改工资信息，更新TXT文件** |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| **SalaryDataService.update** | **当工资策略发生修改的时候进行修改salaryPO，并对对应的txt文件进行修改** | |
| **SalaryDataService.getSalary** | **从数据层获取工资制定的策略salaryPO** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表32 staffbl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Staff.getUser** | 语法 | **public UserVO getUser(String id) throws RemoteException;** |
| 前置条件 | **Id输入正确** |
| 后置条件 | **根据ID查找员工信息** |
| **Staff.add** | **语法** | **public String addUser(String id,UserVO vo) throws RemoteException;** |
|  | **前置条件** | **Id输入正确** |
|  | **后置条件** | **增加一个员工信息，更新TXT文件** |
| **Staff.update** | **语法** | **public String updateUser(String userID, UserVO vo) throws RemoteException;** |
|  | **前置条件** | **ID输入正确** |
|  | **后置条件** | **修改一个员工信息，更新TXT文件** |
| **Staff.delete** | **语法** | **public String deleteUser(String id) throws RemoteException;** |
| **前置条件** | **ID输入正确。目标员工信息存在** |
| **后置条件** | **删除一个员工信息，更新TXT文件** |
| **Staff.** | **语法** | **public List<UserVO> getUsers();** |
| **前置条件** | **无** |
| **后置条件** | **得到所有的员工信息** |
| **Staff.ins** | **语法** | **public List<UserVO> getUsersOfIns(String InstitutionID) throws RemoteException;** |
| **前置条件** | **ID输入正确** |
| **后置条件** | **得到该机构所有的员工信息** |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| **UserDataService.add** | **增加一个userPO，并更新对应的txt文件** | |
| **UserDataService.find** | **根据人员的ID查找一个userPO，并更新对应的txt文件** | |
| **UserDataService.delete** | **删除一个userPO，并更新对应的txt文件** | |
| **UserDataService.modify** | **修改一个userPO，并更新对应的txt文件** | |
| **UserDataService.getUsers** | **返回所有的userPO的一个List** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**表33 strategybl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **Strategy.getVO** | **语法** | **public PriceConstVO getVO();** |
| **前置条件** | **无** |
| **后置条件** | **从数据层得到价格常量** |
| **Strategy.modify** | **语法** | **public String modify(PriceConstVO vo) throws RemoteException;** |
| **前置条件** | **目标价格常量存在** |
| **后置条件** | **修改该价格常量，更新TXT文件** |
| **Strategy.city** | **语法** | **public List<String> getCities();** |
| **前置表情** | **无** |
| **后置条件** | **得到业务涉及的所有城市** |
| **Strategy.diatance** | **语法** | **public double getDistance(String city1,String city2);** |
| **前置条件** | **无** |
| **后置条件** | **得到业务涉及城市之间的距离** |
| **需要的服务（需接口）** | | |
| **服务名** | **服务** | |
| **PriceConstDataService.update** | **当价格距离常量策略发生修改的时候进行修改priceConstPO，并对对应的txt文件进行修改** | |
| **PriceConstDataService.getSalary** | **从数据层获取价格距离常量的策略priceConstPO** | |
| **LogDataService.add(LogPO po)** | **增加一条LogPo信息，并更新对应的txt文件** | |

**5.4数据层的分解**

数据层主要是给业务逻辑层提供数据访问服务，包括对于持久化数据的增、删、改、删、查。业务逻辑需要的服务由DataService接口提供。持久化数据的保存主要使用TXT文件保存。

**5.4.1数据层模块的职责**

数据层模块的职责如表所示

**表34 数据层模块的职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| UserDataService | 用户ID密码TXT文件的接口，提供增、删、改、查服务。 |
| LogisticsDataService | 物流信息持久化TXT文件的接口，提供增、删、查服务。 |
| SendDataService | 寄件单的持久化TXT文件接口，提供增、删、查服务。 |
| ReceiveDataService | 收件单的持久化TXT文件接口，提供增、删、查服务。 |
| DeliveryDataService | 派件单的持久化TXT文件接口，提供增、删、查服务。 |
| EntruckDataService | 装车单的持久化TXT文件接口，提供增、删、查服务。 |
| TransferDataService | 中转单的持久化TXT文件接口，提供增、删、查服务。 |
| ArrivalDataService | 到达单的持久化TXT文件接口，提供增、删、查服务。 |
| TruckDataService | 车辆信息的持久化TXT文件接口，提供增、删、改、查服务。 |
| EntryDataService | 入库单的持久化TXT文件接口，提供增、删、查服务。 |
| ShipmentDataService | 出库单的持久化TXT文件接口，提供增、删、查服务。 |
| InventoryDataService | 库存信息的持久化TXT文件接口，提供增、删、查服务。 |
| AccountDataService | 银行账户的持久化TXT文件接口，提供增、删、改、查服务。 |
| PaymentDataService | 付款单的持久化TXT文件接口，提供增、删、查服务。 |
| DriverDataService | 司机信息的持久化TXT文件接口，提供增、删、改、查服务。 |
| IncomeDataService | 收款单的持久化TXT文件接口，提供增、删、查服务。 |
| PriceConstDataService | 价格常量的持久化TXT文件接口，提供增、删、改、查服务。 |
| SalaryDataService | 薪水策略制定的持久化TXT文件接口，提供增、删、改、查服务。 |
| LogDataService | 操作日志的持久化数据库接口，提供增、查服务。 |
| InstitutionDataService | 人员机构管理的持久化数据库接口，提供增、删、改、查服务。 |

**表35 数据层模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| **UserDataService.find** | 语法 | **Public UserPO find(String identityID) throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **按identityID进行查找返回相应的UserPO结果** |
| **UserDataService.delete** | 语法 | **Public void delete(UserPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **UserDataService.modify** | 语法 | **Public void modify(UserPO po,String password)throws RemoteException;** |
| **前置条件** | **从TXT文件读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **UserDataService.init** | 语法 | **Public void init()throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **初始化持久化TXT文件** |
| **UserDataService.add** | 语法 | **Public void delete(UserPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件读取的po列表不存在相应的po** |
| **后置条件** | **增加一个po** |
| **LogisticsDataService.modify** | 语法 | **Public void update(String id,String message)throws RemoteException;** |
| **前置条件** | **从TXT文件读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **LogisticsDataService.add** | **语法** | **Public void add(LogisticsPO po)throws RemoteException** |
| **前置条件** | **从TXT文件读取的po列表不存在相应的po** |
| **后置条件** | **增加一个po** |
| **SendDataService.add** | 语法 | **Public void add(SendPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件读取的po列表不存在相同ID的po** |
| **后置条件** | **新增一个po** |
| **SendDataService.delete** | 语法 | **Public void delete(SendPO po)throws RemoteException;** |
| 前置条件 | **从TXT文件读取的po列表存在相应的po** |
| 后置条件 | **删除一个po** |
| **SendDataService.modify** | 语法 | **Public SendPO modify(SendPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **ReceiveDataService.add** | 语法 | **Public void add(ReceivePO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **新增一个po** |
| **ReceiveDataService.delete** | 语法 | **Public void delete(ReceivePO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **ReceiveDataService.find** | **语法** | **Public ReceivePO find(String id)throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **按ID进行查找返回相应的ReceivePO结果** |
| **ReceiveDataService.modify** | **语法** | **Public void modify(ReceivePO po)throws RemoteException;** |
|  |  |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **EntruckDataService.add** | 语法 | **Public void add(EntruckPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表中不存在相应的po** |
| **后置条件** | **新增一个po** |
| **EntruckDataService.delete** | 语法 | **Public void delete(EntruckPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **EntruckDataService.find** | 语法 | **Public EntruckPO find(String id)throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **按ID进行查找返回相应的EntruckPO结果** |
| **EntruckDataService.modify** | 语法 | **Public void modify(EntruckPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **TransferDataService.add** | **语法** | **Public void add(TransferPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **新增一个po** |
| **TransferDataService.delete** | 语法 | **Public void delete(TransferPO po)throws RemoteException;** |
|  |  |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **TransferDataService.find** | 语法 | **Public TransferPO find(String id)throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **按ID进行查找返回相应的TransferPO结果** |
| **TransferDataService.modify** | **语法** | **Public void modify(TransferPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **ArrivalDataService.add** | 语法 | **Public void add(ArrivalPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **新增一个po** |
| **ArrivalDataService.delete** | 语法 | **Public void delete(ArrivalPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **ArrivalDataService.modify** | 语法 | **Public ArrivalPO modify(ArrivalPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **TrucktDataService.add** | 语法 | **Public void add(TruckPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **新增一个po** |
| **TruckDataService.delete** | 语法 | **Public void delete(TruckPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **TruckDataService.modify** | 语法 | **Public void modify(TruckPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **TruckDataService.find** | 语法 | **Public TruckPO find(String id)throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **按ID进行查找返回相应的TransferPO结果** |
| **EntryDataService.add** | 语法 | **Public void add(EntryPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **新增一个po** |
| **EntryDataService.delete** | 语法 | **Public void delete(EntryPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **EntryDataService.modify** | 语法 | **Public void modify(EntryPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **EntryDataService.find** | 语法 | **Public EntryPO find(String id)throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **按ID进行查找返回相应的EntryPO结果** |
| **ShipmentDataService.add** | 语法 | **Public void add(ShipmentPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **新增一个po** |
| **ShipmentDataService.delete** | 语法 | **Public void delete(ShipmentPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **ShipmentDataService.modify** | **语法** | **Public void modify(ShipmentPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **ShipmentDataService.find** | 语法 | **Public ShipmentPO find(String id)throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **按ID进行查找返回相应的ShipmentPO结果** |
| **InventoryDataService.add** | 语法 | **Public void update(InventoryPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **增加一个po** |
| **InventoryDataService.modify** | **语法** | **public void modify(InventoryPO po) throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **InventoryDataService.delete** | **语法** | **public void delete(String InstitutionID) throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应ID的po** |
| **后置条件** | **删除一个po** |
| **AccountDataService.add** | 语法 | **Public void add (AccouintPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **增加一个po** |
| **AccountDataService.modify** | 语法 | **Public void modifyupdate(AccouintPO po,String name)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **AccountDataService.delete** | 语法 | **Public void delete (AccouintPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **AccountDataService.search** | **语法** | **public List<AccountPO> search(String key)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **存入list中并返回list** |
| **DeliveryDataService.add** | **语法** | **public void add(DeliveryPO po) throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **增加一个po** |
| **DeliveryDataService.delete** | **语法** | **public void delete(DeliveryPO po) throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **DeliveryDataService.modify** | **语法** | **public void modify(DeliveryPO po) throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **DeliveryDataService.find** | **语法** | **public DeliveryPO find(String id) throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **按ID进行查找返回相应的DeliveryPO结果** |
| **DriverDataService.add** | **语法** | **public void add(DriverPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **增加一个po** |
| **DriverDataService.delete** | **语法** | **public void delete(DriverPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **DriverDataService.modify** | **语法** | **public void Modify(DriverPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **DriverDataService.find** | **语法** | **public DriverPO find(String id)throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **按ID进行查找返回相应的DriverPO结果** |
| **IncomeDataService.add** | **语法** | **public void add(IncomePO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **增加一个po** |
| **IncomeDataService.delete** | **语法** | **public void delete(IncomePO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **IncomeDataService.modify** | **语法** | **public void modify(IncomePO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **IncomeDataService.find** | **语法** | **public List<IncomePO> getListByDate(Date d)throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **根据输入时间 返回符合该时间的IncomePO列表** |
| **PaymengDataService.add** | **语法** | **public void add(PaymentPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **增加一个po** |
| **PaymengDataService.delete** | **语法** | **public void delete(PaymentPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **PaymengDataService.modify** | **语法** | **public void modify(PaymentPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **PaymengDataService.find** | **语法** | **public List<PaymentPO> find(Date d)throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **根据输入时间 返回符合该时间的PaymentPO列表** |
| **LogDataService.add** | **语法** | **public void add(LogPO po) throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **增加一个po** |
| **LogDataService.clear** | **语法** | **public void clear() throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **清空LogPO** |
| **LogDataService.check** | **语法** | **public List<LogPO> getLogList()throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **返回LogPO列表** |
| **InstitutionDataService.add** | **语法** | **public void add(InstitutionPO po) throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表不存在相应的po** |
| **后置条件** | **增加一个po** |
| **InstitutionDataService.delete** | **语法** | **public void delete(String institutionID) throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **删除一个po** |
| **InstitutionDataService.modify** | **语法** | **public void modify(String id,String name)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **InstitutionDataService.find** | **语法** | **public List<InstitutionPO> getInsList() throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **返回InstitutionPO列表** |
| **PriceconstDataService.update** | **语法** | **public void update(PriceConstPO po)throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **PriceconstDataService.check** | **语法** | **public PriceConstPO getPriceConst()throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **返回PriceConstPO列表** |
| **SalaryDataService.update** | **语法** | **public SalaryPO getSalaryPO()throws RemoteException;** |
| **前置条件** | **从TXT文件中读取的po列表存在相应的po** |
| **后置条件** | **更新一个po** |
| **SalaryDataService.check** | **语法** | **public SalaryPO getSalaryPO()throws RemoteException;** |
| **前置条件** | **无** |
| **后置条件** | **返回SalaryPO列表** |

**6信息视角**

**6.1数据持久化对象**

系统的PO类就是对应的相关的实体类，在此只做简单的介绍，详见代码中的对应的相关的po类，属性在类中都有指出，而且有相备注注释。

**6.2 Txt持久化格式**

Txt数据格式 是将一个数据的相关信息 分布在一行上 每项信息都用 ” ” 隔开，二级信息用“;”间隔开