物流管理与服务系统（CMASS）

软件体系结构描述文档

--------------V0.1



**2015.10.21**

**目录**

[1. 引言 3](#_Toc3705)

[1.1 编制目的 3](#_Toc15145)

[1.2 词汇表 3](#_Toc13792)

[1.3 参考资料 3](#_Toc32090)

[2. 产品概述 3](#_Toc4671)

[3. 逻辑视角 3](#_Toc6917)

[4. 组合视角 5](#_Toc19972)

[4.1 开发包图 5](#_Toc8709)

[4.1 运行时进程 9](#_Toc8246)

[4.2 物理部署 9](#_Toc13173)

[5. 接口视角 10](#_Toc6898)

[5.1 模块的职责 10](#_Toc24773)

[5.2 用户界面层的分解 12](#_Toc15005)

[5.2.1 用户界面层模块的职责 13](#_Toc2705)

[5.2.2 用户界面层模块的接口规范 13](#_Toc9197)

[5.2.3 用户界面模块设计原理 13](#_Toc5916)

[5.3 业务逻辑层的分解 14](#_Toc12027)

[5.3.1 业务逻辑层模块的职责 14](#_Toc17808)

[5.3.2 业务逻辑层模块的接口规范 14](#_Toc4074)

[5.4 数据层的分解 20](#_Toc19585)

[5.4.1 数据层模块的职责 21](#_Toc12794)

[5.4.2 数据层模块的接口规范 21](#_Toc9539)

[6. 信息视角 27](#_Toc16380)

[6.1 数据持久化对象 27](#_Toc13062)

# 1. 引言

## 1.1 编制目的

本报告详细完成对快递系统的概要设计，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 |
| **YloadDoc** | 营业厅装车单 |  |
| **YArrivalDoc** | 营业厅到达单 |  |
| **ZArrivalDoc** | 中转中心到达单 |  |
| **TransferDoc** | 中转单 |  |
| **ZloadDoc** | 中转中心装车单 |  |
| **SendDoc** | 寄件单 |  |
| **InWarehouseDoc** | 入库单 |  |
| **OutWarehouseDoc** | 出库单 |  |
| **OverDoc** | 收件单 |  |
| **PayDoc** | 付款单 |  |
| **GatheringDoc** | 收款单 |  |
| **YDeliverDoc** | 派件单 |  |
| **CostInComeDocPO** | 成本收益表 |  |
| **RecordPO** | 仓库记录 |  |
| **CMASSS** | 快递管理与服务系统 |  |

## 1.3 参考资料

1) 快递管理与服务系统（CMASS）用例文档v1.0

2) 快递管理与服务系统（CMASS）软件需求规格说明文档

# 产品概述

CMASS是为一家本地民营物流企业开发的快递管理与服务系统，开发的目标是帮助企业wanchengrichang 的业务流程，包括：

SF1：对输送、保管、包装、流通加工等快递物流活动进行衔接活动，对装卸活动的管理，确定最恰当的装卸方式，合理配置。获得较好的经济效果。

SF2：对运输活动的管理，要求选择技术经济效果最好的运输方式及联运方式，合理确定运输路线，以实现安全、迅速、准时、价廉的要求。

SF3：处理营业厅司机，车辆的管理。

SF4：对快件的运费和到达时间有预估功能。

SF5：处理中转中心或营业厅快递的接收与派件，包括装车管理，装运管理等。

SF6：对仓库的管理，包括仓库货物的出库，入库，报警，分区，盘点等。

SF7：帮助财务人员处理公司财务状况，包括账户管理，制作成本收益，经营情况表等。

SF8：协助总经理的决策。

通过CMASS的应用，期望为民营企业提高员工工作效率，降低成本，减少企业内部交流障碍，吸引顾客，提高利润。

# 逻辑视角

逻辑快递系统中，选择了分层体系结构风格，将系统分为三层（展示层、业务逻辑层、数据层）能够很好地示意整个高层抽象。展示层包括GUI页面的实现，业务逻辑层包括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图所示：

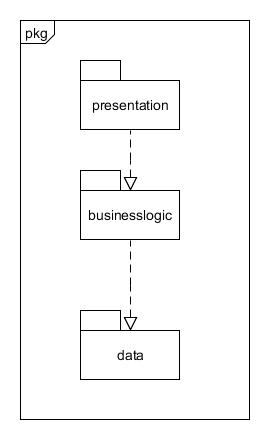


图1 参照体系结构风格的包图表达逻辑视角

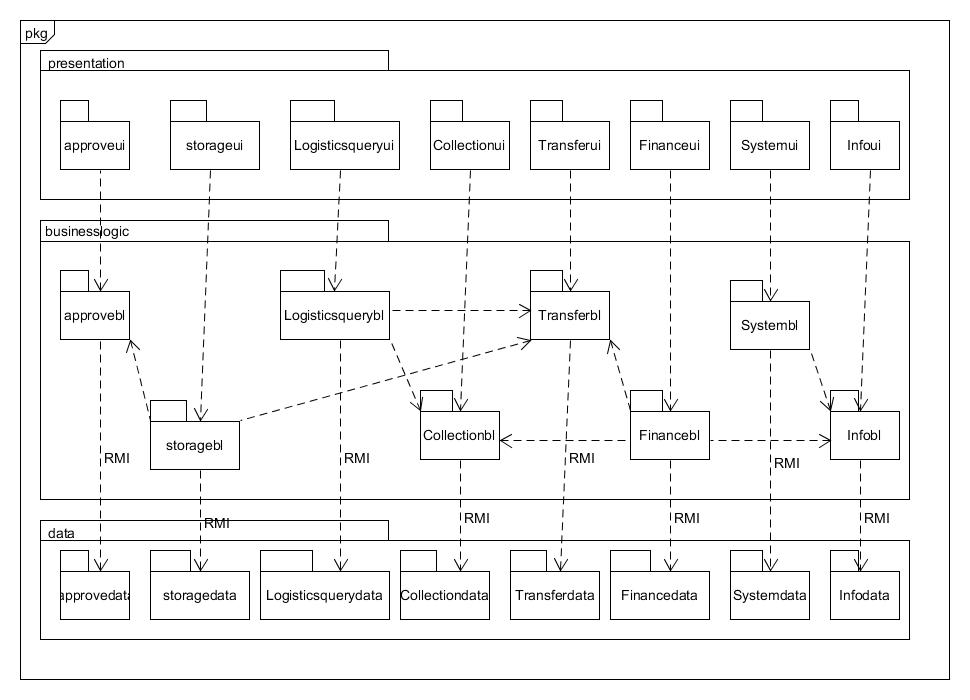


图2 软件体系结构逻辑设计方案

# 组合视角

## 开发包图

快递系统的最终开发包图如表1所示：

**表1 快递系统的最终开发包设计**

|  |  |
| --- | --- |
| 开发（物理）包 | 依赖的其他开发包 |
| mainui | Storageui, Logisticsqueryui, Collectionui, Transferui, Infoui, Financeui, Systemui, Approveui,VO |
| storageui | Storageblservice  界面类库包 |
| Storageblservice |  |
| storagebl | storagedataservice  transferbl  Storageblservice |
| storagedataservice | Java RMI  PO |
| storagedata | Java RMI  Databaseutility  PO |
| Logisticsqueryui | Logisticsqueryblservice  界面类库包 |
| Logisticsqueryblservice |  |
| logisticsquerybl | logisticsquerydataservice  transferbl  collectionbl  Logisticsqueryblservice |
| logisticsquerydataservice | Java RMI  PO |
| logisticsquerydata | Java RMI  databaseutility  PO |
| Collectionui | Collectionblservice  界面类库包 |
| collectionblservice |  |
| collectionbl | Collectiondataservice  Collectionblservice |
| collectiondataservice | Java RMI  PO |
| collectiondata | Java RMI  databaseutility  PO |
| Transferui | Transferblservice  界面类库包 |
| transferblservice |  |
| transferbl | Transferdataservice  Transferblservice |
| transferdataservice | Java RMI  PO |
| transferdata | Java RMI  databaseutility  PO |
| Infoui | Infoblservice  界面类库包 |
| infoblservice |  |
| infobl | Infodataservice  Infoblservice |
| Infodataservice | Java RMI  PO |
| Infodata | Java RMI  databaseutility  PO |
| Financeui | financeblservice  界面类库包 |
| financeblservice |  |
| financebl | financedataservice  collectionbl  infobl,  transferbl  financeblservice |
| financedataservice | Java RMI  PO |
| financedata | Java RMI  databaseutility  PO |
| Systemui | Systemblservice  界面类库包 |
| systemblservice |  |
| systembl | systemdataservice  infobl  systemblservice |
| systemdataservice | Java RMI  PO |
| systemdata | Java RMI  databaseutility  PO |
| Approveui | Approvebl  界面类库包 |
| approveblservice |  |
| approvebl | Approvedata  storagebl  collectionbl  transferbl  financebl  approveblservice |
| approvedataservice | Java RMI  PO |
| approvedata | Java RMI  databaseutility  PO |
| databaseutility |  |
| utilitybl |  |
| VO |  |
| PO |  |
| Java RMI |  |
| 界面类库包 | IO |

快递系统的客户端开发包图如下图所示：

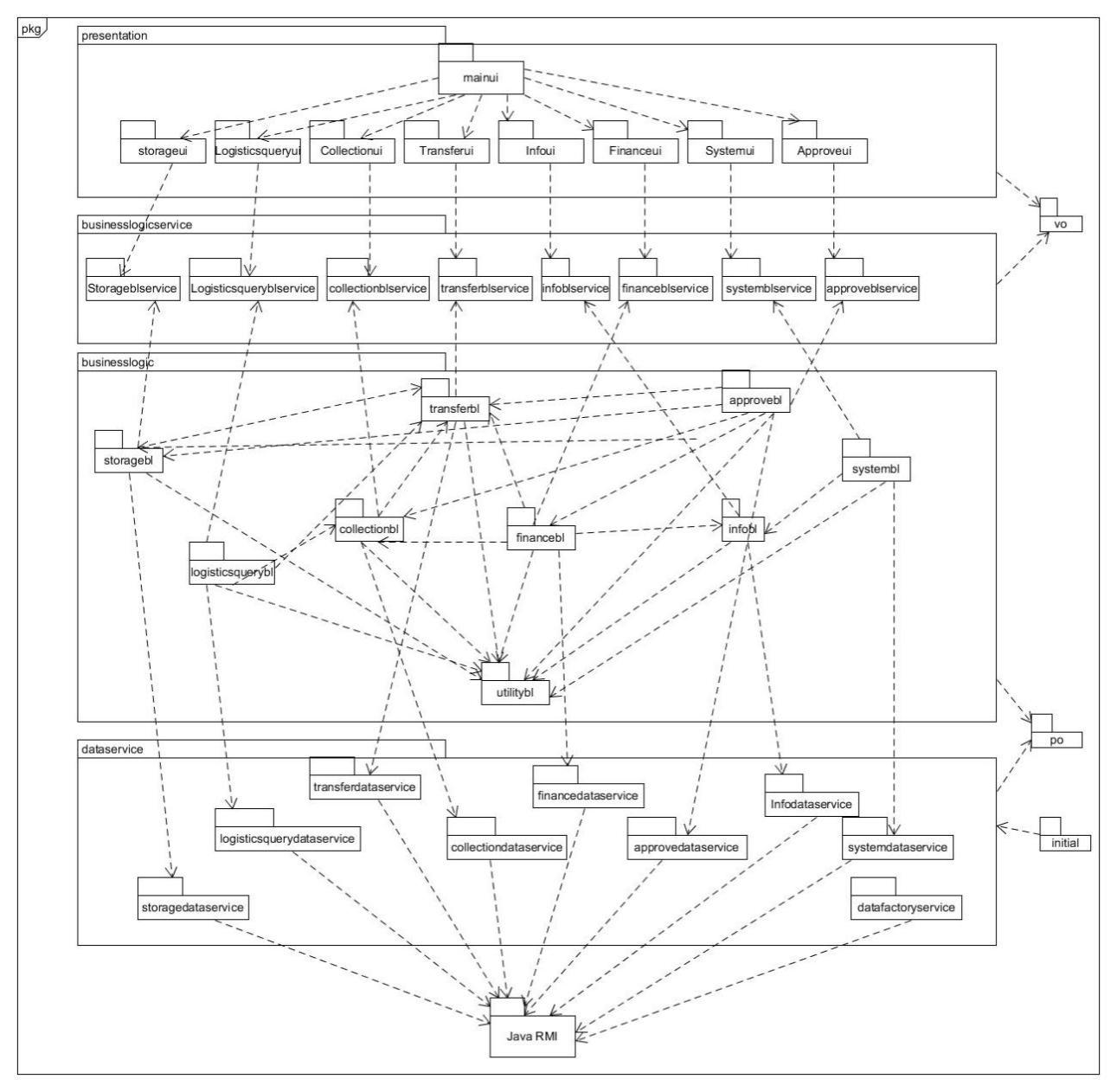


图3 快递系统的客户端开发包图

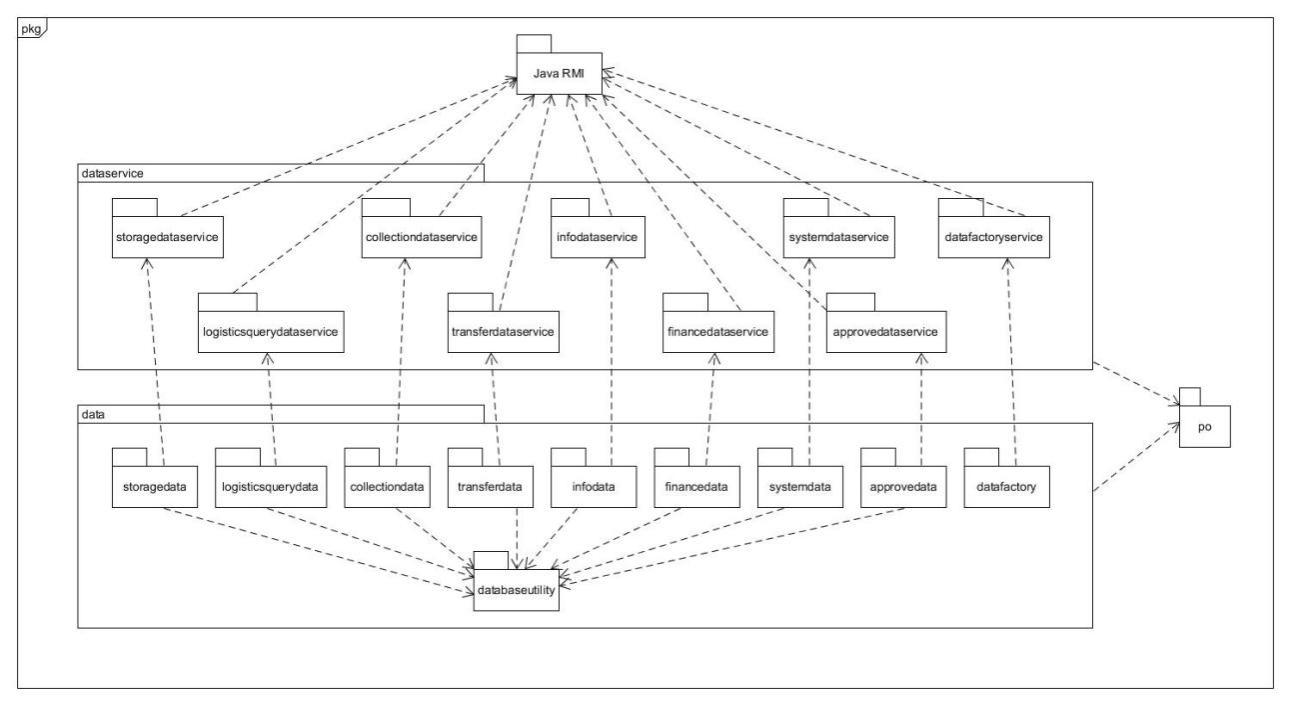


图4 快递系统的服务器开发包图

## 4.1 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进程，其进程图如下图所示。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

## 物理部署

在快递系统中客户端构件是放在客户端机器上，服务器够减是放在服务器机器上。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，不需要在独立部署。部署图如图所示：



图6 部署图

# 5. 接口视角

## 5.1 模块的职责

客户端模块额服务器模块视图如图所示，客户端个层和服务器隔层的职责如表所示：



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 | 基于窗口的工作人员使用界面 |
| 业务逻辑层 | 对于用户界面的输入进行响应并进行对应的逻辑处理 |
| 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四所示。

**表4 层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| Storageblservice  Logisticsqueryblservice  Collectionblservice  Transferblservice  Infoblservice  Financeblservice  approveblservice  systemblservice | 客户端展示层 | 客户端业务逻辑层 |
| Storagedataservice  Logisticsquerydataservice  Collectiondataservice  Transferdataservice  Infobdataservice  Financedataservice  approvedataservice  systemdataservice  datafactoryservice | 客户端业务逻辑层 | 服务器端数据层 |

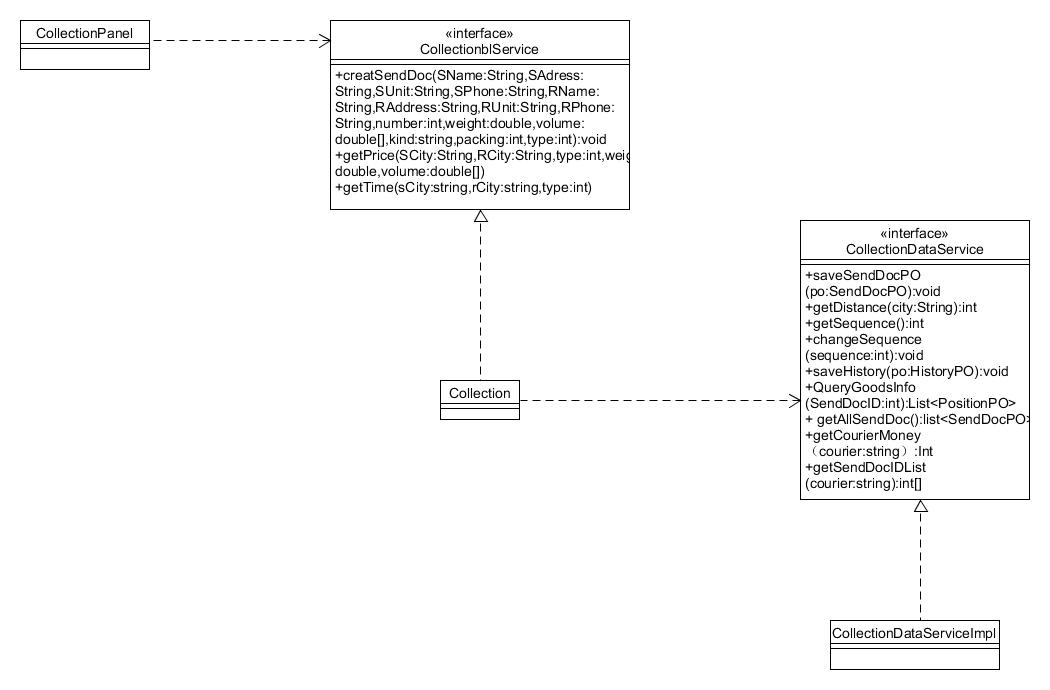


图9 揽件用例层之间调用的接口

## 5.2 用户界面层的分解

根据需求，系统存在38个界面：人员管理、机构管理、机构与人员管理、账户密码权限管理、管理员主界面、查询物流主界面、快递员主界面、揽件、派件与收件、营业厅主界面、车辆装车、接件与派件、收款单、车辆信息管理、司机信息管理、中转中心主界面、交通装运管理、中转中心接收单、仓库管理主界面、库存信息初始化、库区调整、仓库流量、库存盘点、出库、入库、财务人员主界面、成本管理、租金管理、运费管理、工资管理、银行账户管理、总经理主界面、查看统计报表、成本收益表、经营情况表、系统日志查询、审批单据、工资策略管理。界面跳转如图所示：

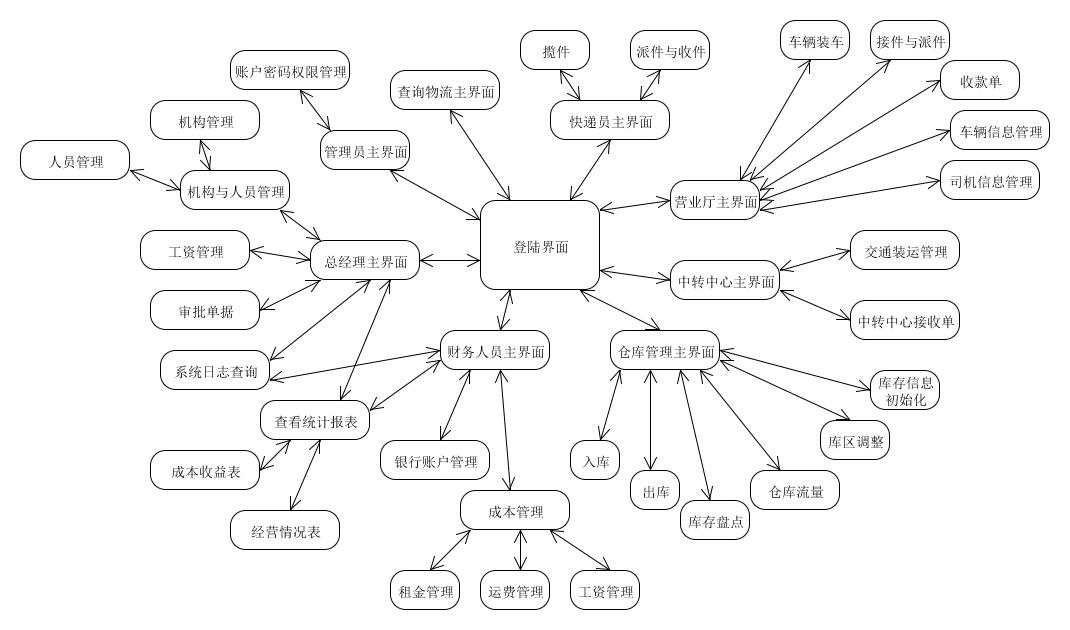


图10 用户界面跳转

服务器端和客户端的用户界面设计接口是一致的，知识具体的页面不一样。用户界面类如下图所示：

### 5.2.1 用户界面层模块的职责

如表5所示为用户界面模块的职责。

**表5 用户界面模块的职责**

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

### 5.2.2 用户界面层模块的接口规范

用户界面模块的接口规范如表6 所示。

表6 用户界面模块的接口规范

|  |  |  |
| --- | --- | --- |
| MainFrame | 语法 | Init(args:Strings[]) |
| 前置条件 | 无 |
| 后置条件 | 显示frame以及LoginPanel |

用户界面层需要的服务接口如表7所示。

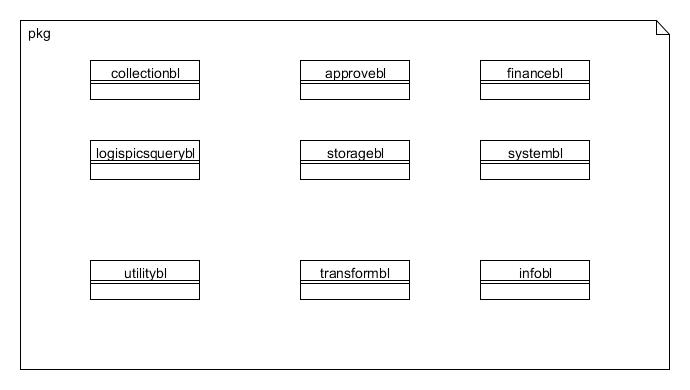
表7 用户界面层模块需要的服务接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| Businesslogicserviceservice.systemblservice | 登录界面的业务逻辑接口 |
| Businesslogicserviceservice.\*blservice | 每一个界面都有一个对应的业务逻辑接口 |

### 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

**表8 业务逻辑层模块的职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| Collectionbl | 负责实现对应与揽件界面所需要的服务 |
| approvebl | 负责实现对应与审判单据界面所需要的服务 |
| financebl | 负责实现对应与成本管理、银行账户管理、查看统计报表、生成收款单（这个是在营业厅主界面下 界面所需要的服务 |
| logispicsquerybl | 负责实现对应与物流查询界面所需要的服务 |
| storagebl | 负责实现对应与出库管理、入库管理、库存信息初始化、库存盘点、库区调整、仓库流量界面所需要的服务 |
| systembl | 负责实现对应系统日志保存与查询、账户密码权限管理 、登陆账户查验界面所需要的服务 |
| utilitybl | 无 |
| transformbl | 负责实现对应与车辆装车管理、交通装运管理、派件与接件、生成中转中心接收单界面所需要的服务 |
| infobl | 负责实现对应与司机信息管理、车辆信息管理、机构与人员管理、工资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 Collectionbl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Collectionbl.QueryGoodsInfo | 语法 | | Public List<PositionPO> QueryGoodsInfo(int SendDocID) |
| 前置条件 | | SendDocID 正确 |
| 后置条件 | | 根据寄件单编号查询物流状态信息，返回一个List<PositionPO> |
| Collectionbl.getAllSendDoc | 语法 | | Public list<SendDocPO> getAllSendDoc() |
| 前置条件 | | 无 |
| 后置条件 | | 得到待审批寄件单 |
| Collectionbl.getCourierMoney(Courier courier) | 语法 | | Public Int getCourierMoney（Courier courier） |
| 前置条件 | | 快递员正确 |
| 后置条件 | | 得到该快递员的收款金额（通过寄件单文件） |
| Collectionbl.getSendDocIDList(Courier courier) | 语法 | | Public list<int> getSendDocIDList(Courier courier) |
| 前置条件 | | 快递员正确 |
| 后置条件 | | 得到快递员的订单条形码号（通过寄件单文件） |
| 需要的服务 | | | |
| 服务名 | | 服务 | |
| CollectionDataService.saveSendDocPO(SendDocPO po) | | 保存寄件单单一持久化对象 | |
| CollectionDataService.getDistance(String city) | | 获取距离数据 | |
| CollectionDataService.getSequence() | | 获取序列号 | |
| CollectionDataService.changeSequence(int sequence) | | 改变最后序列号 | |
| CollectionDataService.saveHistory(HistoryPO po) | | 保存历史数据单一持久化对象 | |

**表10 Transferbl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Transferbl.getAllYDeliverDoc | 语法 | | Public YDeliverDocPO DegetYDeliverDoc() |
| 前置条件 | | 得到派件单 |
| 后置条件 | | 无 |
| Transferbl.getAllYArrivalDoc | 语法 | | Public YArrivalDocPO getAllYArrivalDoc() |
| 前置条件 | | 无 |
| 后置条件 | | 得到待审批到达单 |
| Transferbl.getAllZArrivalDoc | 语法 | | Public ZArrivalDocPO getAllZArrivalDoc() |
| 前置条件 | | 无 |
| 后置条件 | | 得到待审批中转中心到达单 |
| Transferbl.getAllTransferDoc | 语法 | | Public TransferDocPO getAllTransferDoc() |
| 前置条件 | | 无 |
| 后置条件 | | 得到待审批中转单 |
| Transferbl.getAllYloadDoc | 语法 | | Public YloadDocPO getAllYloadDoc() |
| 前置条件 | | 无 |
| 后置条件 | | 得到待审批营业厅装车单 |
| Transferbl.getAllZloadDoc | 语法 | | Public ZloadDocPO getAllZloadDoc() |
| 前置条件 | | 无 |
| 后置条件 | | 得到待审批中转中心装车单 |
| Transferbl.getFreight(int number ) | 语法 | | Public int getFreight(int number ) |
| 前置条件 | | 编号正确 |
| 后置条件 | | 输入装车单编号分别得到运费 |
| 需要的服务 | | | |
| 服务名 | | 服务 | |
| TransferDataService.saveTransferDocPO(TransferDocPO po) | | 保存中转单单一持久化对象 | |
| TransferDataService.getTransferSequence() | | 获取中转单序列号 | |
| TransferDataService.changeTransferSequence(int TransferSequence) | | 改变中转单最后序列号 | |
| TransferDataService.saveZloadDocPO(ZloadDocPO po) | | 保存装车单单一持久化对象 | |
| TransferDataService.getZloadSequence() | | 获取装车单序列号 | |
| TransferDataService.changeZloadSequence(int ZloadSequence) | | 改变装车单最后序列号 | |
| TransferDataService.saveZArrivalDocPO(ZArrivalDocPO po) | | 保存中转中心到达单单一持久化对象 | |
| TransferDataService.getZArrivalSequence() | | 获取中转中心到达单序列号 | |
| TransferDataService.changeZArrivalSequence(int ZArrivalSequence) | | 改变中转中心到达单最后序列号 | |
| TransferDataService.saveYArrivalDocPO(YArrivalDocPO po) | | 保存营业厅到达单单一持久化对象 | |
| TransferDataService.getYArrivalSequence() | | 获取营业厅到达单序列号 | |
| TransferDataService.changeYArrivalSequence(int YArrivalSequence) | | 改变营业厅到达单最后序列号 | |
| TransferDataService.saveYDeliverDocPO(YDeliverDocPO po) | | 保存派件单单一持久化对象 | |
| TransferDataService.getYDeliverSequence() | | 获取派件单序列号 | |
| TransferDataService.changeYDeliverSequence(int YDeliverSequence) | | 改变派件单最后序列号 | |
| TransferDataService.saveOverDocPO(OverDocPO po) | | 保存收件单单一持久化对象 | |
| TransferDataService.getOverSequence() | | 获取收件单序列号 | |
| TransferDataService.changeOverSequence(int OverSequence) | | 改变收件单最后序列号 | |
| TransferDataService.getTransferDocPO(int TransferDocID) | | 根据ID查找中转单单一持久化对象 | |
| TransferDataService.getZloadDocPO(int ZloadDocID) | | 根据ID查找装车单单一持久化对象 | |
| TransferDataService.getZArrivalDocPO(int ZArrivalDocID) | | 根据ID查找中转中心到达单单一持久化对象 | |
| TransferDataService.getYArrivalDocPO(int YArrivalDocID) | | 根据ID查找营业厅到达单单一持久化对象 | |
| TransferDataService.getYDeliverDocPO(int DeliverDocID) | | 根据ID查找派件单单一持久化对象 | |
| TransferDataService.getOverDocPO(int OverDocID) | | 根据ID查找收件单单一持久化对象 | |
| Collectionbl.saveHistory(HistoryPO po) | | 保存历史数据单一持久化对象 | |
|  | |  | |

**表11 Storagebl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| storagebl.getOutWarehouseDoc | 语法 | Public OutWarehouseDocPO getOutWarehouseDoc() |
| 前置条件 | 无 |
| 后置条件 | 得到出库单 |
| Storagebl.getInWarehouseDoc | 得到入库单 | Public InWarehouseDoc getInWarehouseDoc() |
| 前置条件 | 无 |
| 后置条件 | 得到入库单 |
| 需要的服务 | | |
| 服务名 | 服务 | |
| Transferbl.throughTransferDoc() | 遍历本中转中心未出库的中转单，返回一个中转单列表VO对象，用于展示层显示 | |
| Transferbl.find(int TransferDocID) | 根据选中的中转单编号查询该中转单信息，返回一个TransferDocVO对象 | |
| Transferbl.throughZArrivalDoc () | 遍历本中转中心仓库的所有接收单，返回一个接收单列表VO对象，用于展示层展示 | |
| Transferbl.findZArrivalDoc(int ZArrivalDocID ) | 根据入库接收单编号查找其属性信息 | |
| StorageDataService.update(RecordListPO t ) | 根据具体的仓库数据持久化对象更新仓库记录文件(点击更新按钮时调用这个方法) | |
| StorageDataService.deleteRecord | 清空仓库库存记录 | |
| StorageDataService.getRecord() | 返回仓库所有库存商品记录的VO对象 | |
| StorageDataService.Modify(double t) | 接收一个百分比，并检查这个百分比的合理性，将对应的百分比的机动区划分给库存爆满的区 | |
| StorageDataService.updateInWarehouseDoc (InWarehouseDocPO t ) | 更新入库单文件 | |
| StorageDataService.updateOutWarehouseDoc (OutWarehouseDocPO t ) | 更新出库单文件 | |
| StorageDataService.find(int StorageItemID) | 根据一个快递编号返回一个库存对象的引用 | |
| StorageDataService.getPercent(String district) | 得到当前区域（航运区，货运区，汽运区）已经占用的比例 | |

**表12 Logisticsquerybl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Logisticsquerybl.QueryGoodsInfo(int SendDocID) | 语法 | Public PositionVO QueryGoodsInfo(int SendDocID) |
| 前置条件 | 无 |
| 后置条件 | 返回对应快递的物流信息给展示层 |
| 需要的服务 | | |
| 服务名 | 服务 | |
| Collectionbl.QueryGoodsInfo(int SendDocID) | 根据寄件单编号查询物流状态信息，返回一个List<PositionPO> | |

**表13 Infobl模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Infobl.getSalary | 语法 | Public List<SalaryPO> GetSalary()； |
| 前置条件 | 登陆账户已被授权和验证 |
| 后置条件 | 从文件读入工资总额 |
| Infobl.getIncome | 语法 | Public List<IncomePO> GetIncome()； |
| 前置条件 | 登陆账户已被授权和验证 |
| 后置条件 | 从文件中读入总收入 |
| Infobl.getCourierList | 语法 | Public List<CourierPO> GetCourierList(); |
| 前置条件 | 登陆账户已被授权和验证 |
| 后置条件 | 得到本机构快递员列表 |
| Infobl.findStaff | 语法 | Public List<StaffPO> FindStaff(); |
| 前置条件 | 登陆账户已被授权和验证 |
| 后置条件 | 获取未分配账户的人员列表 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| Infodataservice.findDriver | | 获得司机信息 |
| Infodataservice.findCar | | 获得车辆信息 |
| Infodataservice.findSalary | | 获得薪水信息 |
| Infodataservice.findStaff | | 获得员工 |
| Infodataservice.findInstitution | | 获得机构信息 |
| Infodataservice.findDistance | | 获得城市距离信息 |
| Infodataservice.saveDriver(DriverPO po) | | 保存司机信息 |
| Infodataservice.saveCar(CarPO po) | | 保存车辆信息 |
| Infodataservice.saveSalary(SalaryPO po) | | 保存薪水信息 |
| Infodataservice.saveStaff(StaffPO po) | | 保存员工机构信息 |
| Infodataservice.saveInstitution(InstitutionPO po ) | | 保存机构信息 |
| Infodataservice.saveDistance(string city1,string city2,double distance) | | 保存城市距离信息 |

**表14 Systembl的接口规范**

|  |  |
| --- | --- |
| 服务名 | 服务 |
| Systemdataservice.findOperationLog | 获得系统日志 |
| Systemdataservice.findAccount | 查找账户密码权限信息 |
| Infobl.findStaffList | 新建账户时，获取未分配的人员列表（POList） |
| Systemdataservice.saveOperationLog(OperationPO po) | 保存系统日志 |
| Systemdataservice.saveAccount(AccountPO po) | 保存修改的账户密码权限信息 |

**表15 Approvebl的接口规范**

|  |  |
| --- | --- |
| 服务名 | 服务 |
| storagebl.getOutWarehouseDoc(po对象) | 得到出库单 |
| Storagebl.getInWarehouseDoc | 得到入库单 |
| Transferbl.getYDeliverDoc | 得到派件单 |
| Transferbl.getYArrivalDoc | 得到到达单 |
| Transferbl.getZArrivalDoc | 得到中转中心到达单 |
| Transferbl.getTransferDoc | 得到中转单 |
| Transferbl.getYloadDoc | 得到营业厅装车单 |
| Transferbl.getZloadDoc | 得到中转中心装车单 |
| Collectionbl.getSendDoc | 得到寄件单 |
| Financebl.getPayDoc | 得到付款单 |
| Financebl.getGatheringDoc | 得到收款单 |

**表16 Financebl的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Financebl.getPayDoc | 语法 | public PayDoc getPayDoc(); |
|  | 前置条件 | 总经理已经被识别和授权 |
|  | 后置条件 | 总经理得到付款单文件并可修改付款单的PO对象 |
| Financebl.getGatheringDoc | 语法 | Public GatheringDoc getGatheringDoc() |
|  | 前置条件 | 总经理已经被识别和授权 |
|  | 后置条件 | 总经理得到收款单文件并可修改收款单的PO对象 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| Infobl.getSalary | 得到工资的总额 | |
| Transferbl.getFreight(int number ) | 输入装车单编号分别得到运费 | |
| Financedataservice.getRent() | 得到财务人员填写的总租金 | |
| Infodataservice.getBusinessManList | 得到财务人员列表便于选择付款人 | |
| Financedataservice.updateAccountInfo(string name,Account account) | 更新账户（修改账户名称） | |
| Financedataservice.updatePayDoc(PayDoc Po) | 更新付款单文件 | |
| Financedataservice.deleteAccount(string name) | 删除账户 | |
| Financedataservice.addAccount(Account PO) | 增加账户 | |
| Financedataservice.getAccount | 得到公司的银行账户列表 | |
| Infobl.getIncome（） | 得到总收入 | |
| Financedataservice.get PayDoc（Time startTime,Time endTime） | 得到期间付款单列表（PO） | |
| Financedataservice.getGatheringDoc(Time startTime,Time endTime) | 得到期间收款单列表（PO） | |
| Infobl.getCourierList() | 得到本机构快递员列表（POlist） | |
| Collectionbl.getCourierMoney(Courier courier) | 得到该快递员的收款金额（通过寄件单文件） | |
| Collectionbl.getSendDocIDList(Courier courier) | 得到快递员的订单条形码号（通过寄件单文件） | |

## 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，包括对于持久化数据的增删改查。Collection业务逻辑层需要的服务有CollectionDataService接口提供。由于持久化数据的保存主要为序列化文件，所示抽象了数据服务。数据层模块的描述具体如图所示。

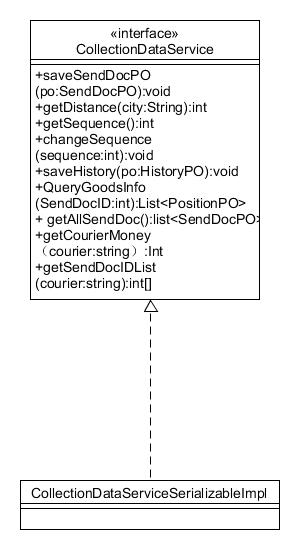


图13 数据层模块的描述

### 5.4.1 数据层模块的职责

**表17 数据层模块的职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| CollectionDataService | 持久化数据库的接口，提供增、删、改、查、集体导出服务 |
| CollectionDataServiceSerializableImpl | 基于序列化文件的持久化数据库的接口，提供增、删、改、查、集体导出服务 |

### 5.4.2 数据层模块的接口规范

**表18 StorageDateService的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务 | | |
| StorageDataService.update(RecordPOList t ) | 语法 | Public update(RecordPOList t) |
| 前置条件 |  |
| 后置条件 | 根据具体的仓库数据持久化对象更新仓库记录文件(点击更新按钮时调用这个方法) |
| StorageDataService.deleteRecord | 语法 | Public void clear() |
| 前置条件 |  |
| 后置条件 | 清空仓库库存记录 |
| StorageDataService.getRecordList() | 语法 | Public getRecordList() |
| 前置条件 |  |
| 后置条件 | 返回仓库所有库存记录对象PO列表 |
| StorageDataService.Modify(double t) | 语法 | Public modify(double t) |
| 前置条件 |  |
| 后置条件 | 接收一个百分比，并检查这个百分比的合理性，将对应的百分比的机动区划分给库存爆满的区 |
| StorageDataService.updateInWarehouseDoc (InWarehouseDocPO t ) | 语法 | public updateInWarehouseDoc (InWarehouseDocPO t ) |
| 前置条件 |  |
| 后置条件 | 根据入库单更新入库单文件 |
| StorageDataService.updateOutWarehouseDoc (OutWarehouseDocPO t ) | 语法 | Public updateOutWarehouseDoc(OutWarehouseDocPO t) |
|  | 前置条件 |  |
|  | 后置条件 | 根据出库单更新入库单文件 |
| StorageDataService.find(int StorageItemID) | 语法 | Public find(int StorageItemID) |
| 前置条件 |  |
| 后置条件 | 根据一个快递编号返回一个库存对象的引用 |
| StorageDataService.getPercent(String district) | 语法 | Public getPercent(String district) |
|  | 前置条件 |  |
|  | 后置条件 | 得到当前区域（航运区，货运区，汽运区）已经占用的比例 |
| storagebl.getOutWarehouseDoc | 语法 | public ArrayList<OutWarehouseDocPO>getOutWarehouseDoc() |
|  | 前置条件 |  |
|  | 后置条件 | 到待审批出库单，返回值是一个PO对象 |
| Storagebl.getInWarehouseDoc | 语法 | Public ArrayList<InWarehouseDocPO> getInWarehouseDoc |
|  | 前置条件 |  |
|  | 后置条件 | 得到待审批入库单，返回一个POlist |

**表19 Infodataservice的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Infodataservice.getBusinessManList | 语法 | Public List<BusinessManPO> getBusinessManList(); |
| 前置条件 | 无 |
| 后置条件 | 得到财务人员列表便于选择付款人 |
| Infodataservice.findDriver | 语法 | Public List<DriverPO> findDriver(); |
| 前置条件 | 无 |
| 后置条件 | 获得司机信息 |
| Infodataservice.findCar | 语法 | Public List<CarPO> findCar(); |
| 前置条件 | 无 |
| 后置条件 | 获得车辆信息 |
| Infodataservice.findSalary | 语法 | Public List<SalaryPO> findSalary(); |
| 前置条件 | 无 |
| 后置条件 | 获得薪水信息 |
| Infodataservice.findStaff | 语法 | Public List<StaffPO> findStaff(); |
| 前置条件 | 无 |
| 后置条件 | 获得员工 |
| Infodataservice.findInstitution | 语法 | Public List<InstitutionPO> findInstitution(); |
| 前置条件 | 无 |
| 后置条件 | 获得机构信息 |
| Infodataservice.findDistance | 语法 | Public List<DistancePO> findDistance(); |
| 前置条件 | 无 |
| 后置条件 | 获得城市距离信息 |
| Infodataservice.saveDriver(DriverPO po) | 语法 | Public void saveDriver(DriverPO po); |
| 前置条件 |  |
| 后置条件 | 增加一个司机PO记录 |
| Infodataservice.saveCar(CarPO po) | 语法 | Public void saveCar(CarPO po); |
| 前置条件 |  |
| 后置条件 | 增加一个车辆PO记录 |
| Infodataservice.saveSalary(SalaryPO po) | 语法 | Public void saveSalary (SalaryPO po); |
| 前置条件 | 无 |
| 后置条件 | 增加一个薪水PO记录 |
| Infodataservice.saveStaff(StaffPO po) | 语法 | Public void saveStaff(StaffPO po); |
| 前置条件 | 无 |
| 后置条件 | 增加一个人员PO记录 |
| Infodataservice.saveInstitution(InstitutionPO po ) | 语法 | Public void saveInstitution (InstitutionPO po); |
| 前置条件 | 无 |
| 后置条件 | 增加一个机构PO记录 |
| Infodataservice.saveDistance(string city1,string city2,double distance) | 语法 | Public void saveDistance(string city1,string city2,double distance)； |
| 前置条件 | 无 |
| 后置条件 | 保存城市距离信息 |

**表20 Systemdataservice的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Systemdataservice.findOperationLog | 语法 | Public List<OperationPO> findOperationLog(); |
| 前置条件 | 无 |
| 后置条件 | 返回OperationPO列表 |
| Systemdataservice.findAccount | 语法 | Public List<AccountPO> findAccount(); |
| 前置条件 | 登录账户已被授权验证 |
| 后置条件 | 返回所有账户的List |
| Systemdataservice.saveOperationLog(OperationPO po) | 语法 | Public void saveOperationLog(OperationPO po); |
| 前置条件 | 文件中不存在相同的PO |
| 后置条件 | 新增一个PO |
| Systemdataservice.saveAccount(AccountPO po) | 语法 | Public void saveAccount(AccountPO po) |
| 前置条件 | 无 |
| 后置条件 | 更新一个po |

**表21 Financedataservice的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Financedataservice.getRent() | 语法 | Public int getRent() |
|  | 前置条件 | 无 |
|  | 后置条件 | 得到财务人员填写的总租金 |
| Financedataservice.updatePayDoc(PayDoc Po) | 语法 | Public void updatePayDoc (PayDoc Po) |
|  | 前置条件 | 无 |
|  | 后置条件 | 更新付款单文件 |
| Financedataservice.updateAccountInfo(string name,Account account) | 语法 | Public void updateAccountInfo(string name，Account account) |
|  | 前置条件 | 无 |
|  | 后置条件 | 修改账户名称 |
| Financedataservice.deleteAccount(string name) | 语法 | Public void deleteAccount(string name) |
|  | 前置条件 | 无 |
|  | 后置条件 | 删除该账户 |
| Financedataservice.addAccount(string name) | 语法 | Public void addAccount(string name) |
|  | 前置条件 | 无 |
|  | 后置条件 | 增加一个账户，其金额设为0 |
| Financedataservice.getAccount | 语法 | Public list<Accoutnt> getAccount() |
|  | 前置条件 | 无 |
|  | 后置条件 | 得到公司的银行账户列表 |
| Financedataservice.getPayDoc（Time startTime,Time endTime） | 语法 | Public list<PayDoc>getPayDOc(Time startTime,Time endTime) |
|  | 前置条件 | 无 |
|  | 后置条件 | 得到期间付款单列表（PO） |
| Financedataservice.getGatheringDoc(Time startTime,Time endTime) | 语法 | Public list<GatheringDoc>(Time startTime,Time endTime) |
|  | 前置条件 | 无 |
|  | 后置条件 | 得到期间收款单列表（PO） |

**表22 CollectionDataServicede 接口规范**

|  |  |  |
| --- | --- | --- |
| CollectionDataService.saveSendDocPO | 语法 | Public void saveSendDocPO(SendDocPO po) |
| 前置条件 | 无 |
| 后置条件 | 保存寄件单单一持久化对象 |
| CollectionDataService.getDistance | 语法 | Public int getDistance(String city) |
| 前置条件 | 无 |
| 后置条件 | 获取距离数据 |
| CollectionDataService.getSequence | 语法 | Public int getSequence() |
| 前置条件 | 无 |
| 后置条件 | 获取序列号 |
| CollectionDataService.changeSequence | 语法 | Public void changeSequence(int sequence) |
| 前置条件 | 无 |
| 后置条件 | 改变最后序列号 |
| CollectionDataService.saveHistory | 语法 | Public void saveHistory(HistoryPO po) |
| 前置条件 | 无 |
| 后置条件 | 保存历史数据单一持久化对象 |
| CollectionDataService.QueryGoodsInfo | 语法 | Public List<PositionPO> QueryGoodsInfo(int SendDocID) |
| 前置条件 | 无 |
| 后置条件 | 根据寄件单编号查询物流状态信息，返回一个List<PositionPO> |
| CollectionDataService.getAllSendDoc | 语法 | Public list<SendDocPO> getAllSendDoc() |
| 前置条件 | 无 |
| 后置条件 | 得到待审批寄件单 |
| CollectionDataService.getCourierMoney | 语法 | Public Int getCourierMoney（Courier courier） |
| 前置条件 | 快递员正确 |
| 后置条件 | 得到该快递员的收款金额（通过寄件单文件） |
| CollectionDataService.getSendDocIDList | 语法 | Public list<int> getSendDocIDList(Courier courier) |
| 前置条件 | 快递员正确 |
| 后置条件 | 得到快递员的订单条形码号（通过寄件单文件） |

**表23 TransferDataService的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| TransferDataService.saveTransferDocPO | 语法 | Public void saveTransferDocPO(TransferDocPO po) | |
| 前置条件 | 无 | |
| 后置条件 | 保存中转单单一持久化对象 | |
| TransferDataService.getTransferSequence | 语法 | Public int getTransferSequence() | |
| 前置条件 | 无 | |
| 后置条件 | 获取中转单序列号 | |
| TransferDataService.changeTransferSequence | 语法 | Public void changeTransferSequence(int TransferSequence) | |
| 前置条件 | 无 | |
| 后置条件 | 改变中转单最后序列号 | |
| TransferDataService.saveZloadDocPO | 语法 | Public void saveZloadDocPO(ZloadDocPO po) | |
| 前置条件 | 无 | |
| 后置条件 | 保存装车单单一持久化对象 | |
| TransferDataService.getZloadSequence | 语法 | Public int getZloadSequence() | |
| 前置条件 | 无 | |
| 后置条件 | 获取装车单序列号 | |
| TransferDataService.changeZloadSequence | 语法 | Public void changeZloadSequence(int ZloadSequence) | |
| 前置条件 | 无 | |
| 后置条件 | 改变装车单最后序列号 | |
| TransferDataService.saveZArrivalDocPO | 语法 | Public void saveZArrivalDocPO(ZArrivalDocPO po) | |
| 前置条件 | 无 | |
| 后置条件 | 保存中转中心到达单单一持久化对象 | |
| TransferDataService.getZArrivalSequence | 语法 | Public int getZArrivalSequence() | |
| 前置条件 | 无 | |
| 后置条件 | 获取中转中心到达单序列号 | |
| TransferDataService.changeZArrivalSequence | 语法 | Public void changeZArrivalSequence(int ZArrivalSequence) | |
| 前置条件 | 无 | |
| 后置条件 | 改变中转中心到达单最后序列号 | |
| TransferDataService.saveYArrivalDocPO | 语法 | Public void saveYArrivalDocPO(YArrivalDocPO po) | |
| 前置条件 | 无 | |
| 后置条件 | 保存营业厅到达单单一持久化对象 | |
| TransferDataService.getYArrivalSequence | 语法 | Public int getYArrivalSequence() | |
| 前置条件 | 无 | |
| 后置条件 | 获取营业厅到达单序列号 | |
| TransferDataService.changeYArrivalSequence | 语法 | Public void changeYArrivalSequence(int YArrivalSequence) | |
| 前置条件 | 无 | |
| 后置条件 | 改变营业厅到达单最后序列号 | |
| TransferDataService.saveYDeliverDocPO | 语法 | Public void saveYDeliverDocPO(YDeliverDocPO po) | |
| 前置条件 | 无 | |
| 后置条件 | 保存派件单单一持久化对象 | |
| TransferDataService.getYDeliverSequence | 语法 | Public int getYDeliverSequence() | |
| 前置条件 | 无 | |
| 后置条件 | 获取派件单序列号 | |
| TransferDataService.changeYDeliverSequence | 语法 | Public void changeYDeliverSequence(int YDeliverSequence) | |
| 前置条件 | 无 | |
| 后置条件 | 改变派件单最后序列号 | |
| TransferDataService.saveOverDocPO | 语法 | Void saveOverDocPO(OverDocPO po) | |
| 前置条件 | 无 | |
| 后置条件 | 保存收件单单一持久化对象 | |
| TransferDataService.getOverSequence | 语法 | Public int getOverSequence() | |
| 前置条件 | 无 | |
| 后置条件 | 获取收件单序列号 | |
| TransferDataService.changeOverSequence | 语法 | Public void changeOverSequence(int OverSequence) | |
| 前置条件 | 无 | |
| 后置条件 | 改变收件单最后序列号 | |
| TransferDataService.getTransferDocPO | 语法 | Public TransferDocPO getTransferDocPO(int TransferDocID) | |
| 前置条件 | 无 | |
| 后置条件 | 根据ID查找中转单单一持久化对象 | |
| TransferDataService.getZloadDocPO | 语法 | Public ZloadDocPO getZloadDocPO(int ZloadDocID) | |
| 前置条件 | 无 | |
| 后置条件 | 根据ID查找装车单单一持久化对象 | |
| TransferDataService.getZArrivalDocPO | 语法 | Public ZArrivalDocPO getZArrivalDocPO(int ZArrivalDocID) | |
| 前置条件 | 无 | |
| 后置条件 | 根据ID查找中转中心到达单单一持久化对象 | |
| TransferDataService.getYArrivalDocPO | 语法 | Public YArrivalDocPO getYArrivalDocPO(int YArrivalDocID) | |
| 前置条件 | 无 | |
| 后置条件 | 根据ID查找营业厅到达单单一持久化对象 | |
| TransferDataService.getYDeliverDocPO | 语法 | Public DeliverDocPO getYDeliverDocPO(int DeliverDocID) | |
| 前置条件 | 无 | |
| 后置条件 | 根据ID查找派件单单一持久化对象 | |
| TransferDataService.getOverDocPO | 语法 | Public OverDocPO getOverDocPO(int OverDocID) | |
| 前置条件 | 无 | |
| 后置条件 | 根据ID查找收件单单一持久化对象 | |
| TransferDataService.getAllYDeliverDoc | 语法 | Public YDeliverDocPO DegetYDeliverDoc() |
| 前置条件 | 无 |
| 后置条件 | 得到待审批派件单 |
| TransferDataService.getAllYArrivalDoc | 语法 | Public YArrivalDocPO getAllYArrivalDoc() |
| 前置条件 | 无 |
| 后置条件 | 得到待审批到达单 |
| TransferDataService.getAllZArrivalDoc | 语法 | Public ZArrivalDocPO getAllZArrivalDoc() |
| 前置条件 | 无 |
| 后置条件 | 得到待审批中转中心到达单 |
| TransferDataService.getAllTransferDoc | 语法 | Public TransferDocPO getAllTransferDoc() |
| 前置条件 | 无 |
| 后置条件 | 得到待审批中转单 |
| TransferDataService.getAllYloadDoc | 语法 | Public YloadDocPO getAllYloadDoc() |
| 前置条件 | 无 |
| 后置条件 | 得到待审批营业厅装车单 |
| TransferDataService.getAllZloadDoc | 语法 | Public ZloadDocPO getAllZloadDoc() |
| 前置条件 | 无 |
| 后置条件 | 得到待审批中转中心装车单 |

# 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。

AccountPO类就是银行账户，包含账户名称、账户余额属性。

BusinessHallPO包含所在城市、本机构类型、机构名称属性。

CarPO包含汽车编号、车牌号、服役时间属性。

CostInComeDocPO包含总收入、总支出、总利润属性。

DriverPO包含驾驶员工号、姓名、出生日期、身份证号、手机号码、性别、行驶证期限属性。

GatheringDocPO包含收款日期、收款金额、快递员姓名、收款项（快递编号）属性。

HistoryPO包含出发地、目的地、运输时间属性。

InstitutionPO包含所在城市、本机构类型、机构名称属性。