**物流管理系统**

**软件体系结构描述文档**

**V3.0**

**南京大学软件学院**

第23组：“梦江文月夜”队

2015-10-25

**更新历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **日期** | **变更原因** | **版本号** |
| 全体成员 | 2015/10/15 | 体系结构的设计初步模型 | V1.0 |
| 全体成员 | 2015/10/24 | 体系结构的设计完整模型，添加接口视角、信息视角 | V2.0 |
| 汪文藻、吴秦月 | 2015/12/31 | 体系结构设计接口更新，数据库表更新 | V3.0 |

目录

[1.引言](1.%E5%BC%95%E8%A8%80) 5

1.1编制目的 5

1.2词汇表 5

1.3参考资料 5

2.产品概述 5

3.逻辑视角 5

4组合视角 8

4.1开发包图 8

4.2运行时进程 11

4.3物理部署 11

5.接口视角 12

5.1模块的职责 12

5.2 用户界面层的分解 15

5.2.1用户界面层模块的职责 17

5.2.2用户界面层模块的接口规范 17

5.3 业务逻辑层的分解 26

5.3.1 业务逻辑层模块的职责 26

5.3.2业务逻辑模块的接口规范 27

5.4数据层的分解 44

5.4.1 数据层模块的职责 45

5.2.4 数据层模块的接口规范 45

6.信息视角 60

6.1数据持久化对象 60

6.2数据库表 61

# 1.引言

## 1.1编制目的

本报告详细完成对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 |
|  |  |  |

## 1.3参考资料

1. 物流管理系统用例文档v3.2
2. 物流管理系统软件需求规格说明文档V2.0

# 2.产品概述

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

# 3.逻辑视角

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



图3-1 参照体系结构风格的包图表达逻辑视角

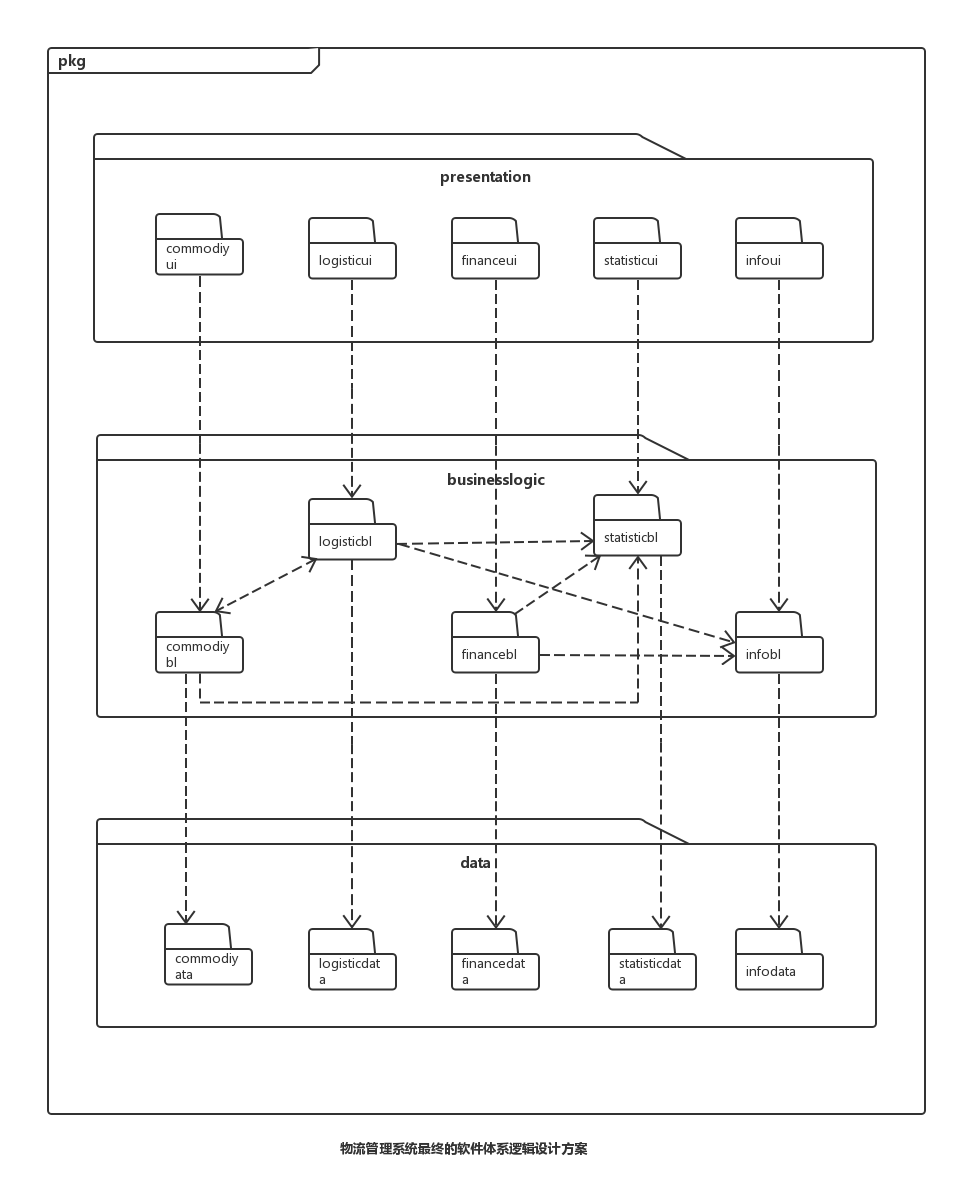
**根据功能性要求获得的初步概要功能设计如下表所示**

|  |  |
| --- | --- |
| **功能** | **对应逻辑包** |
| 物流 | logisticui, logisticbl, logisticdata |
| 库存 | commodityui, commoditybl, commoditydata |
| 财物 | financeui, financebl, financedata |
| 人员机构管理 | infoui, infobl, infodata |
| 数据管理 | statisticui, statisticbl, statisticdata |

**根据不同功能的互相调用情况改进的概要功能设计如下表所示**

|  |  |
| --- | --- |
| **功能** | **对应逻辑包** |
| 物流logistic | logisticui, logisticbl, logisticdata  commoditybl, commoditydata  infobl, infodata  statisticbl, statisticdata |
| 库存commodity | commodityui, commoditybl, commoditydata  logisticbl, logisticdata  statisticbl, statisticdata |
| 财务finance | financeui, financebl, financedata  infoui, infobl, infodata  statisticbl, statisticdata |
| 人员机构管理info | infoui, infobl, infodata |
| 数据管理statistic | statisticui, statisticbl, statisticdata |

**逻辑设计方案和图3-2所示。**



**图3-2 软件体系结构逻辑设计方案**

# 4组合视角

## 4.1开发包图

物流管理系统的初步开发包设计如表4-1-1所示。

**表4-1-1 物流管理系统的初步开发包设计**

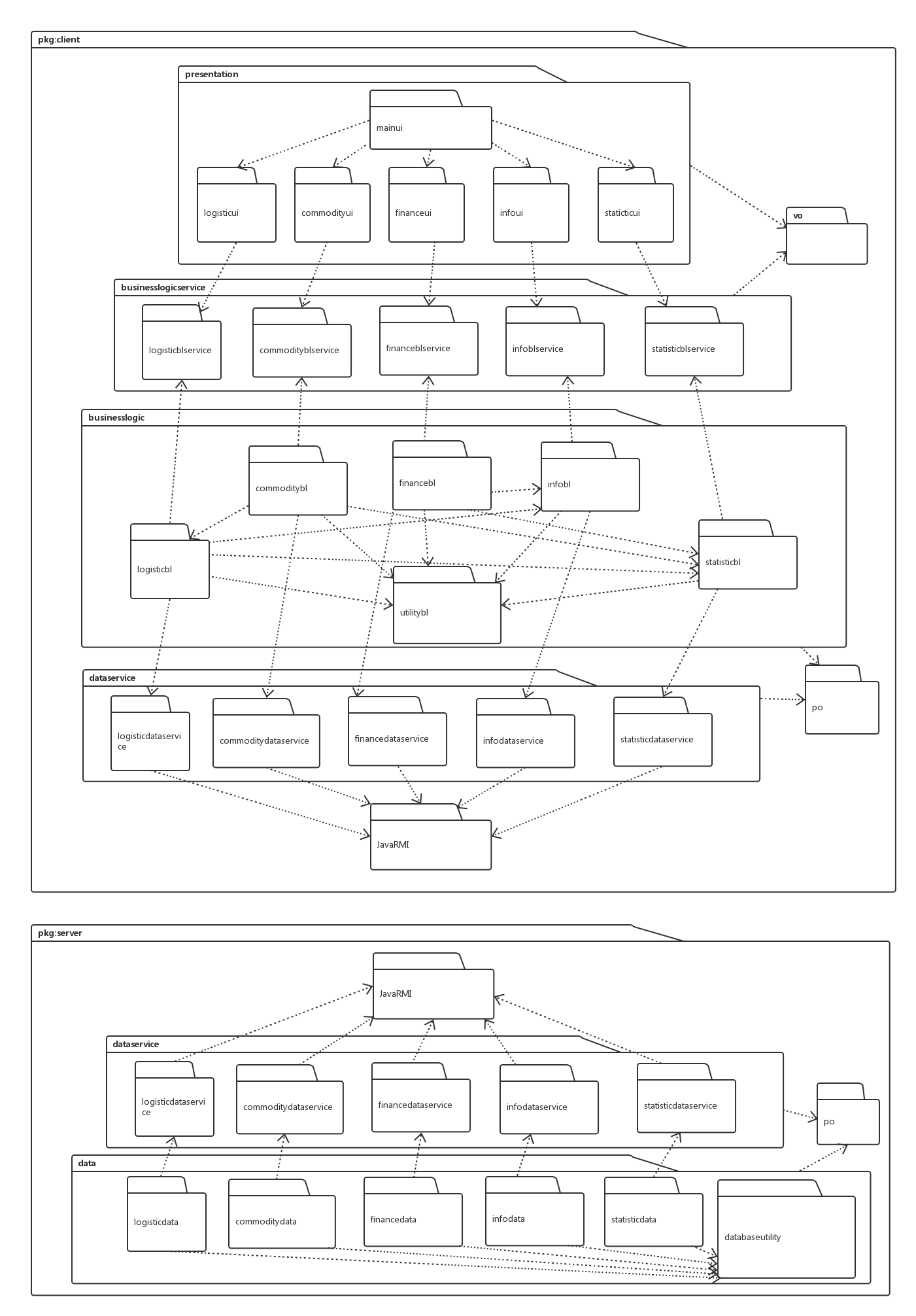
|  |  |  |
| --- | --- | --- |
| **逻辑包** | **物理包（开发包）** | **依赖的其它开发包** |
| logisticui | logisticui | logisticbl |
| logisticbl | logisticbl | logisticdata,commoditybl, infobl ,statisticbl |
| logisticdata | logisticdata |  |
| commodityui | commodityui | commoditybl |
| commoditybl | commoditybl | commoditydata, logisticbl, statisticbl |
| commoditydata | commoditydata |  |
| financeui | financeui | financebl |
| financebl | financebl | financedata, infobl, statisticbl |
| financedata | financedata |  |
| infoui | infoui | infobl |
| infobl | infobl | infodata |
| infodata | infodata |  |
| statisticui | statisticui | statisticbl |
| statisticbl | statisticbl | statisticdata |
| statisticdata | statisticdata |  |

物流管理系统的最终开发包设计如表4-1-2所示。

**表4-1-2 物流管理系统的最终开发包设计**

|  |  |
| --- | --- |
| **开发包（物理包）** | **依赖的其它开发包** |
| Mainui | logisticui,commodityui,financeui,infoui,statisticui |
| Logisticui | logisticblservice,GUI类库包,vo |
| Logisticblservice |  |
| Logisticbl | logisticblservice,logisticdataservice,po,infobl,statisticbl, utilitybl |
| Logisticdataservice | Java RMI,po |
| Logisticdata | logisticdataservice,po,databaseutility |
| Commodityui | commodityblservice,GUI类库包,vo |
| Commodityblservice |  |
| Commoditybl | commodityblservice,commoditydataservice,po,logisticbl, statisticbl,utilitybl |
| Commoditydataservice | Java RMI,po |
| commoditydata | commoditydataservice,po,databaseutility |
| financeui | financeblservice,GUI类库包,vo |
| financeblservice |  |
| financebl | financeblservice,financedataservice,po,infobl,statisticbl, utilitybl |
| financedataservice | Java RMI,po |
| financedata | financedataservice,po,databaseutility |
| infoui | infoblservice,GUI类库包,vo |
| infoblservice |  |
| infobl | infoblservice, infodataservice,po,utilitybl |
| infodataservice | Java RMI,po |
| infodata | infodataservice,po,databaseutility |
| statisticui | statisticblservice,GUI类库包,vo |
| statisticblservice |  |
| statisticbl | statisticblservice, statisticdataservice,po,utilitybl |
| statisticdataservice | Java RMI,po |
| statisticdata | statisticdataservice,po,databaseutility |
| vo |  |
| po |  |
| utilitybl |  |
| GUI类库包 |  |
| Java RMI |  |
| databaseutility | JDBC |

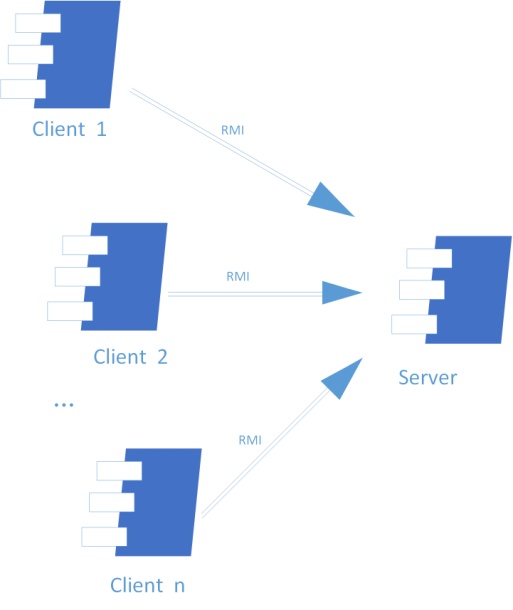
物流管理系统开发包图如图4-1-1所示

****

**图4-1-1物流管理系统的最终开发包设计**

## 4.2运行时进程

在物流管理系统中，会有多个客户端进程和一个服务器端进程，其进程图如图4-2-1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。



**图4-2-1 进程图**

## 4.3物理部署

物流管理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RMI构件。由于Java RMI构件属于JDK 1.7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4-3-1所示。

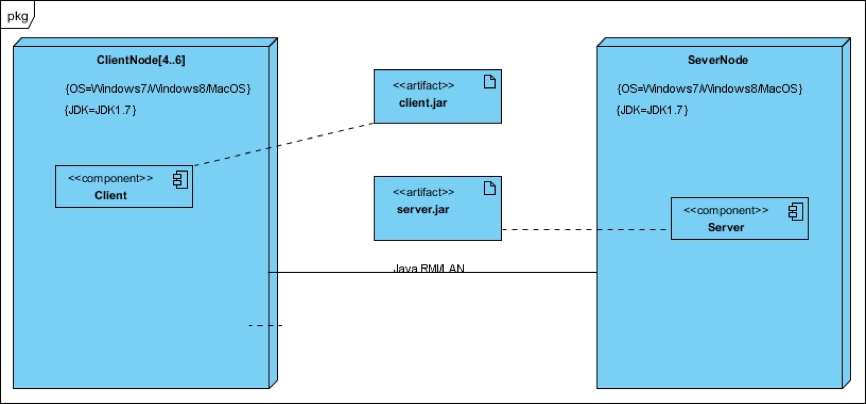
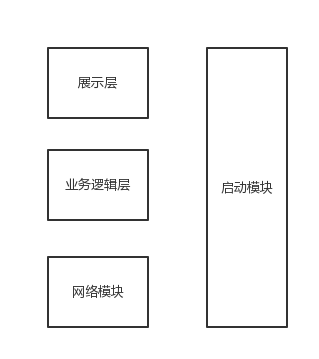


图4-3-1 部署图

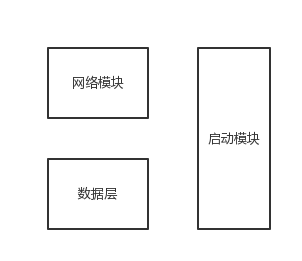
# 5.接口视角

## 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5-1-1和图5-1-2所示。客户端各层和服务端各层的职责分别入表5-1-1和表5-1-2所示。



**图5-1-1 客户端模块视图**



**图5-1-2 服务器端模块视图**

**表5-1-1 客户端各层的职责**

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

**表5-1-2 服务器端各层的职责**

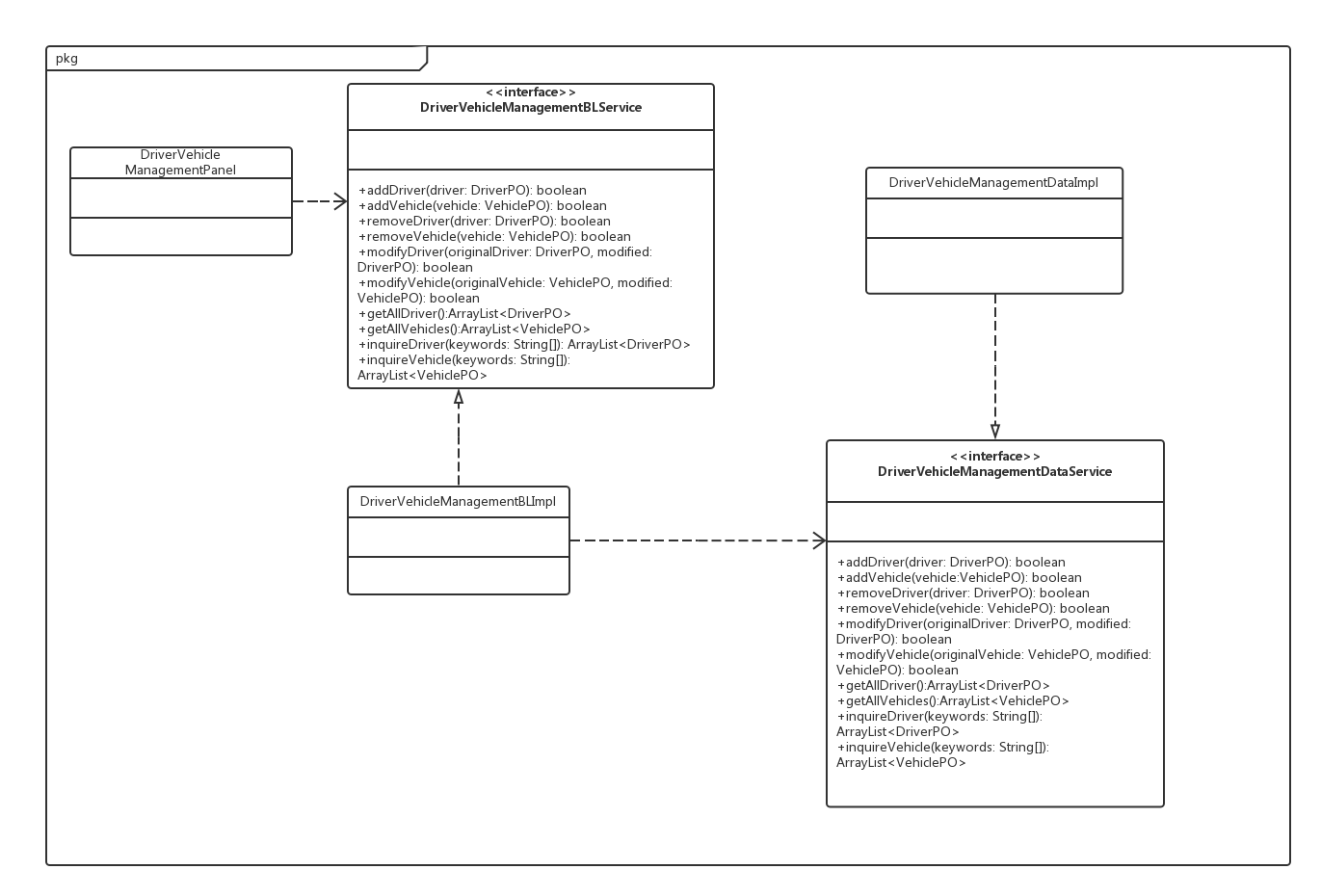
|  |  |
| --- | --- |
| **层** | **职责** |
| 启动模块 | 负责初始化网络通信机制，启动用户界面 |
| 数据层 | 负责数据的持久化及数据访问接口 |
| 服务器端网络模块 | 利用Java RMI机制开启RMI服务，注册RMI服务 |

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5-1-3所示。

**表5-1-3 层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| StorageInBLService  StorageInquiryAllBLService  StorageInquiryPartBLService  StorageOutBLService  BankAccountManagementBLService  CreditNoteInputBLService  PaymentInputBLService  SettlementManagementBLService  DriverVehicleManagementBLService  StaffOrganizationManagementBLService  SystemUserManagementBLService  ArrivalNoteOnServiceBLService  ArrivalNoteOnTransitBLService  DeliveryNoteInputBLService  LoadNoteOnServiceBLService  LoadNoteOnTransitBLService  NoteInputBLService  ReceivingNoteInputBLService  TransitNoteInputBLService  BaseDataBuildingBLService  BusinessDataModificationBLService  ChartOutputBLService  LogInquiryBLService  NoteApprovingBLService | 客户端展示层 | 客户端业务逻辑层 |
| InventoryDataService  StorageInDataService  StorageOutDataService  BankAccountManagementDataService  BaseDataBuildingDataService  CreditNoteInputDataService  PaymentInputDataService  SettlementManagementDataService  DriverVehicleManagementDataService  StaffOrganizationManagementDataService  SystemUserManagementDataService  ArrivalNoteOnServiceDataService  ArrivalNoteOnTransitDataService  DeliveryNoteInputDataService  LoadNoteOnServiceDataService  LoadNoteOnTransitDataService  NoteDataService  ReceivingNoteInputDataService  TransitNoteInputDataService  BusinessDataModificationDataService  ChartOutputDataService  LogInquiryDataService  OrderInquiryDataService | 客户端业务逻辑层 | 服务器端数据层 |

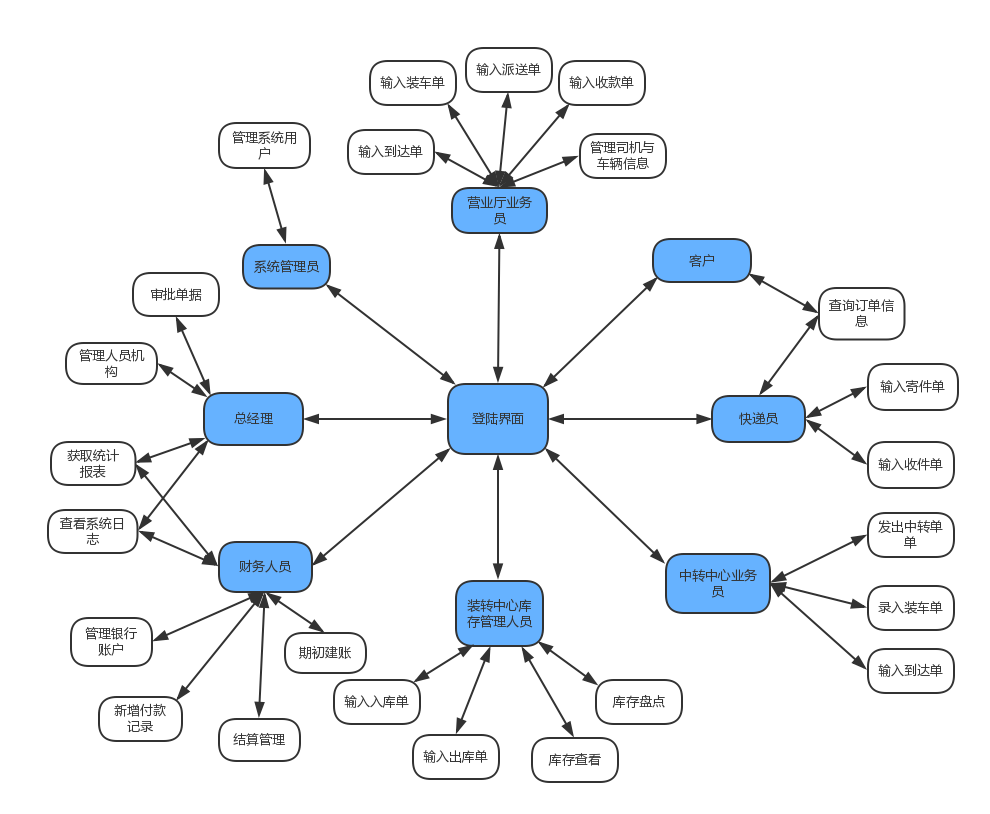
借管理司机与车辆信息用例来说明层之间的调用，如图5-1-3所示。每一层都是由上层依赖了一个接口（需接口），而下层实现这个接口（供接口）。DriverVehicleManagementBLService提供了DriverVehicleManagement界面所需要的所有业务逻辑功能。DriverVehicleManagementDataService提供了对数据库的增、删、改、查等操作。这样的实现就大大降低了层与层之间的耦合。



**图5-1-3管理车辆与司机信息层之间的调用**

## 5.2 用户界面层的分解

根据需求，系统存在33个用户界面：登录界面，客户主界面，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，中转中心库存管理人员主界面，财务人员主界面，总经理主界面，系统管理员主界面，查询订单信息界面，输入营业厅到达单界面，输入营业厅装车单界面，输入派送单界面，管理车辆与司机信息界面，输入寄件单界面，输入收款单界面，输入收款单界面，发出中转单界面，中转中心输入到达单界面，中转中心输入装车单界面，输入入库单界面，输入出库单界面，库存盘点界面，库存查看界面，期初建账界面，结算管理界面，新增付款记录界面，管理银行账户界面，查看系统日志界面，获取统计报表界面，管理人员机构界面，审批单据界面，管理系统用户界面。界面跳转如下图5-2-1所示



**图5-2-1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，只是具体的页面不一样。

用户界面类如图5-2-2所示。



**图5-2-2 用户界面类**

### 5.2.1用户界面层模块的职责

如表5-2-1-1所示为用户界面层模块的职责。

**表5-2-1-1 用户界面层模块的职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| MainFrame | 显示Frame，负责登录界面的显示 |
| logisticui | 负责物流单据输入相关操作界面的显示与跳转 |
| commodityui | 负责库存管理相关操作界面的显示与跳转 |
| financeui | 负责财务管理相关操作界面的显示与跳转 |
| infoui | 负责人员机构信息管理相关操作界面的显示与跳转 |
| statisticui | 负责公司数据管理相关操作界面的显示与跳转 |

### 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5-2-2-1所示。

**表5-2-2-1 用户界面层模块的接口规范**

|  |  |  |
| --- | --- | --- |
| MainFrame | 语法 | init(args:String[]) |
| 前置条件 | 无 |
| 后置条件 | 显示Frame以及LoginPanel |
| Logisticui模块的接口规范 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| LoadCarOrder | 语法 | LoadCarOrder(LogInMsg lim,Service frame)； | |
| 前置条件 | 主界面已加载，营业厅业务员已登录 | |
| 后置条件 | 显示输入营业厅装车单信息的界面panel | |
| MailOrderPanel | 语法 | MailOrderPanel(LogInMsg logInMsg,CourierFrame courierFrame) | |
| 前置条件 | 主界面已加载，快递员已登录 | |
| 后置条件 | 显示输入单据信息的界面panel | |
| ReceiveOrderPanel | 语法 | ReceiveOrderPanel(LogInMsg logInMsg,CourierFrame courierFrame)； | |
| 前置条件 | 主界面已加载，快递员已登录 | |
| 后置条件 | 显示快递员输入收件单的界面panel | |
| TransitNoteInputPanel | 语法 | TransitNoteInputPanel(LogInMsg logInMsg,TransitFrame tf); | |
| 前置条件 | 主界面已加载，中转中心业务员已登录 | |
| 后置条件 | 显示输入中转单信息的界面panel | |
| LoadNoteOnTransitPanel | 语法 | LoadNoteOnTransitPanel( LogInMsg logInMsg,TransitFrame frame); | |
| 前置条件 | 主界面已加载，中转中心业务员已登录 | |
| 后置条件 | 显示输入中转中心装车单信息的界面panel | |
| ArrivalNoteOnTransitPanel | 语法 | ArrivalNoteOnTransitPanel( LogInMsg logInMsg,TransitFrame ft); | |
| 前置条件 | 主界面已加载，中转中心业务员已登录 | |
| 后置条件 | 显示中转中心到达单信息的界面panel | |
| ArrivalOrder | 语法 | ArrivalOrder(LogInMsg lim,Service frame); | |
| 前置条件 | 主界面已加载，营业厅业务员已登录 | |
| 后置条件 | 显示输入营业厅快递收款单信息的界面panel | |
| InquiryPanel | 语法 | InquiryPanel(); | |
| 前置条件 | 主界面已加载，相关用户已登录 | |
| 后置条件 | 显示查询单据信息的界面panel | |
| Car | 语法 | Car(); | |
| 前置条件 | 主界面已加载，相关用户已登录 | |
| 后置条件 | 显示车辆信息的panel | |
| Driver | 语法 | Driver()； | |
| 前置条件 | 主界面已加载，相关用户已登录 | |
| 后置条件 | 显示司机信息的panel | |
| PayMent | 语法 | PayMent(LogInMsg lim,Service frame); | |
| 前置条件 | 主界面已加载，营业厅业务员已登录 | |
| 后置条件 | 显示输入快递收款单信息的界面panel | |
| CourierFrame | 语法 | CourierFrame(LogInMsg logInMsg) | |
| 前置条件 | 相关用户已登录 | |
| 后置条件 | 主界面加载显示 | |
| TransitFrame | 语法 | TransitFrame( LogInMsg logInMsg ) | |
| 前置条件 | 相关用户已登录 | |
| 后置条件 | 主界面加载显示 | |
| Service | 语法 | Service(LogInMsg lim1) | |
| 前置条件 | 相关用户已登录 | |
| 后置条件 | 主界面加载显示 | |
| Financeui模块的接口定义 | | | |
| BankAccountManagePanel | 语法 | BankAccountManagePanel(**boolean** hasBase); | |
| 前置条件 | 主界面已加载，财务人员已登录 | |
| 后置条件 | 显示管理银行账户的界面panel | |
| PaymentInputPanel | 语法 | PaymentInputPanel(LogInMsg logInMsg,financeFrame ff); | |
| 前置条件 | 主界面已加载，财务人员已登录 | |
| 后置条件 | 显示新增付款记录的界面panel | |
| SettlementMangePanel | 语法 | SettlementMangePanel(); | |
| 前置条件 | 主界面已加载，财务人员已登录 | |
| 后置条件 | 显示结算管理的界面panel | |
| BaseDataBuildingPanel | 语法 | BaseDataBuildingPanel( **boolean** hasbase); | |
| 前置条件 | 主界面已加载，财务人员已登录 | |
| 后置条件 | 显示期初建账的界面panel | |
| ChartOutPutPanel | 语法 | ChartOutPutPanel(); | |
| 前置条件 | 主界面已加载，总经理或财务人员已登录 | |
| 后置条件 | 显示获取统计报表的界面panel | |
| Commodityui模块的接口规范 | | | |
| Inorder | 语法 | Inorder(LogInMsg lim,Commodity frame); | |
| 前置条件 | 主界面已加载，库存管理人员已登录 | |
| 后置条件 | 显示入库管理的界面panel | |
| OutOrder | 语法 | OutOrder(LogInMsg lim,Commodity frame); | |
| 前置条件 | 主界面已加载，库存管理人员已登录 | |
| 后置条件 | 显示出库管理的界面panel | |
| PanDian | 语法 | PanDian(JFrame frame); | |
| 前置条件 | 主界面已加载，库存管理人员已登录 | |
| 后置条件 | 显示库存盘点的界面panel | |
| ChaKan | 语法 | ChaKan(LogInMsg lim)); | |
| 前置条件 | 主界面已加载，库存管理人员已登录 | |
| 后置条件 | 显示库存查看的界面panel | |
| Infoui模块的接口规范 | | | |
| StaffManagement | 语法 | StaffManagementPanel(); | |
| 前置条件 | 主界面已加载，总经理已登录 | |
| 后置条件 | 显示管理人员机构信息的界面panel | |
| SystemUser | 语法 | SystemUser(LogInMsg lim); | |
| 前置条件 | 主界面已加载，系统管理员已登录 | |
| 后置条件 | 显示管理系统用户的界面panel | |
| Statisticui模块的接口规范 | | | |
| ModifyBusinessDataPanel | 语法 | | ModifyBusinessDataPanel(); |
| 前置条件 | | 主界面已加载，总经理已登录 |
| 后置条件 | | 显示修改业务数据的界面panel |
| SystemLogPanel | 语法 | | SystemLogPanel(); |
| 前置条件 | | 主界面已加载，总经理或财务人员已登录 |
| 后置条件 | | 显示查询系统日志的界面panel |
| OrderApprove | 语法 | | OrderApprove(); |
| 前置条件 | | 主界面已加载，总经理已登录 |
| 后置条件 | | 显示单据审批的界面panel |

用户界面层需要的接口规范如表5-2-2-2所示

**表5-2-2-2用户界面层需要的规范**

**Part1：logisticbl提供的接口**

|  |  |  |
| --- | --- | --- |
| **接口名** | **方法** | **详细信息** |
| NoteInputBLService | Public ResultMsg inputNote(NoteVO noteVO);; | 业务员要求输入单据信息界面的接口 |
| public ResultMsg submitNote(NoteVO noteVO); | 业务员确认和对确认提交单据界面的接口 |
| ArrivalNoteOnServiceBLService | public ResultMsg inputHallArrivalDoc(ArrivalNoteOnServiceVO arrialDocVO); | 营业厅业务员要求输入到达单 界面的接口 |
| public ResultMsg submitHallArrivalDoc(DeliverNoteOnServiceVO deliverDocVO); | 营业厅业务员核对信息后要求提交单据界面的接口 |
| public ResultMsg inputHallDeliverDoc(DeliverNoteOnServiceVO deliverDocVO); | 营业厅业务员要求输入派件单界面的接口 |
| public ResultMsg submitHallDeliverDoc(DeliverNoteOnServiceVO deliverDocVO); | 营业厅业务员核对信息后要求提交单据界面的接口 |
| ArrivalNoteOnTransitBLService | public ResultMsg inputCenterArrivalDoc(ArrivalNoteOnTransitVO centerArrivalDocVO); | 中转中心业务员要求输入中转中心到达单界面的接口 |
| public ResultMsg submitCenterArrivalDoc(ArrivalNoteOnTransitVO centerArrivalDocVO); | 中转中心业务员核对信息后要求提交单据界面的接口 |
| DeliveryNoteInputBLService | public ResultMsg inputSendDoc(SendingNoteVO sendDocVO); | 快递员要求输入寄件单界面的接口 |
| public sendDocMsg submitSendDoc(SendingNoteVO sendDocVO); | 快递员核对信息后要求提交单据界面的接口 |
| LoadNoteOnServiceBLService | public ResultMsg inputHallLoadDoc(LoadNoteOnServiceVO hallLoadDocVO); | 营业厅业务员要界面的接口求输入营业厅装车单 |
| public ResultMsg submitHallLoadDoc(LoadNoteOnServiceVO hallLoadDocVO); | 营业厅业务员核对信息后要求提交单据界面的接口 |
| LoadNoteOnTransitBLService | public ResultMsg inputCenterLoadDoc(LoadNoteOnTransitVO centerLoadDocVO); | 中转中心业务员要求输入中转中心装车单界面的接口 |
| public ResultMsg submitCenterLoadDoc(LoadNoteOnTransitVO centerLoadDocVO); | 中转中心业务员核对信息后要求提交单据界面的接口 |
| ReceivingNoteInputBLService | public ResultMsg inputReceiveDoc(ReceingNoteVO receiveDocVO); | 快递员要求输入收件单界面的接口 |
| public ResultMsg submitSecdDoc(ReceingNoteVO receiveDocVO); | 快递员核对信息后要求提交单据界面的接口 |
| TransitNoteInputBLService | public ResultMsg inputCenterTransitDoc(TransitNoteOnTransitVO centerTransitDocVO); | 中转中心业务员要求输入中转中心中转单界面的接口 |
| public ResultMsg submitCenterTransitDoc(TransitNoteOnTransitVO centerTransitDocVO); | 中转中心业务员核对信息后要求提交单据界面的接口 |

**Part2：commoditybl模块提供的接口**

|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| StorageInBLService | public ResultMsg addPutInStorgaeDoc(PutInStorageVO putInStorageVo); | 库存管理人员输入入库单信息，选择提交的界面接口 |
| public ResultMsg submitPutInStorageDoc(PutInStorageVO putInStorageVo); | 库存管理人员核对确认无误要求提交入库单信息的界面接口 |
| **public** InventoryStatus alarm(); | 提示库存报警的界面接口 |
| StorageInquiryAllBLService | public ResultMsg inputFirst(String firstTime); | 第一次盘点输入初始时间的界面接口 |
| public ResultMsg request(); | 库存管理人员请求进行库存盘点界面的接口 |
| StorageInquiryPartBLService | public ResultMsg inputTime(String time1,String time2); | 库存管理人员要求进行库存查看输入时间段界面的接口 |
| **public** InventoryVO show(String time1,String time2); | 库存管理人员要求进行库存查看的接口 |
| StorageOutBLService | public ArrayList<StorageOutVO> show(StorageOutVO storageOutVo); | 库存管理人员要求显示出库单信息界面的接口 |
| public ResultMsg addStorageOutDoc(StorageOutVO storageOutVo); | 库存管理人员要求生成出库单界面的接口 |
| public ResultMsg submitStorageOutDoc(StorageOutVO storageOutVo); | 库存管理人员确认生成出库单界面的接口 |
| StorageMapExportService | **public** ResultMsg exportTo(InventoryVO data,String path); | 库存管理人员导出库存快照的接口 |

**Part3： financebl模块提供的接口**

|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| CreditNoteInputBLService | **public** **double** getOrderMoney(String barcode) **throws** ElementNotFoundException; | 财务人员获取单件快件的金额 |
| public ResultMsg addReceipeDoc(ReceipeVO vo); | 财务人员选择新增收款单的功能界面的接口 |
| public ResultMsg submitReceipeDoc(ReceipeVO vo); | 财务人员确认新增收款单的功能界面的接口 |
| PaymentInputBLService | public ResultMsg addPaymentRecord(PaymentRecordVO vo); | 财务人员选择新增付款记录界面的接口 |
| public ResultMsg submitPaymentRecord(PaymentRecordVO vo); | 财务人员核对无误，要求提交单据界面的接口 |
| SettlementManagementBLService | public ResultMsg addReceiveRecord(ReceiveRecordVO vo); | 财务人员要求新增收款记录界面的接口 |
| public ResultMsg submitReceiveRecord(ReceiveRecordVO vo); | 财务人员核对无误，要求提交单据界面的接口 |
| **public** ArrayList<IncomeNoteVO> find(IncomeNoteVO vo); | 财务人员查询收款单界面的接口 |

**Part4：infobl模块的接口规范**

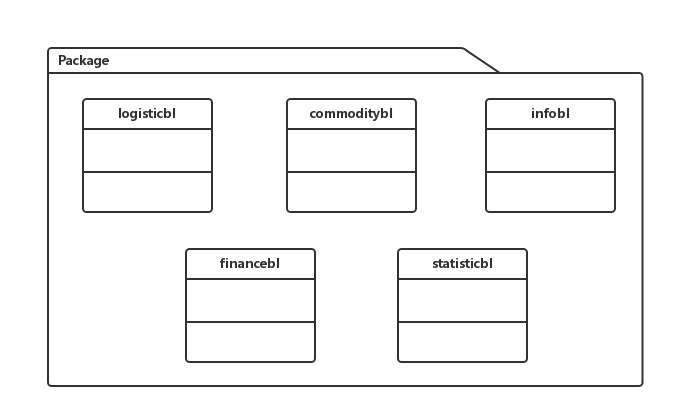
|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| DriverVehicleManagementBLService | public ResultMsg addVehicle(VehicleInfoVO vo); | 业务员选择新增车辆信息功能界面的接口 |
| public ResultMsg deleteVehicle(VehicleInfoVO vo); | 业务员选择删除车辆信息功能界面的接口 |
| public ResultMsg modifyVehicle(VehicleInfoVO vo); | 业务员选择修改车辆信息功能界面的接口 |
| public ResultMsg addVehicle(DriverInfoVO vo); | 业务员选择新增司机信息功能界面的接口 |
| public ResultMsg deleteDriver(DriverInfoVO vo); | 业务员选择删除司机信息功能界面的接口 |
| public ResultMsg modifyDriver(DriverInfoVO vo); | 业务员选择修改司机信息功能界面的接口 |
| public ArrayList<VehicleInfoVO> findVehicle(String keyword); | 业务员选择查询车辆信息界面的接口 |
| public ArrayList<DriverInfoVO> findDriver(String keyword); | 业务员选择查询司机信息界面的接口 |
| StaffOrganizationManagementBLService | public ResultMsg addStaff(StaffInfoVO vo); | 业务员选择新增人员信息界面的接口 |
| public ResultMsg delStaff(StaffInfoVO vo); | 业务员选择删除人员信息界面的接口 |
| public ResultMsg ModifyStaff(StaffInfoVO vo); | 业务员选择修改人员信息界面的接口 |
| **public** ArrayList<StaffVO> findStaffInfo(StaffVO vo); | 业务员选择查找人员信息界面的接口 |
| public ResultMsg addOrganization(OrganizationInfoVO vo); | 业务员选择新增机构信息界面的接口 |
| public ResultMsg delOrganization(OrganizationInfoVO vo); | 业务员选择删除机构信息界面的接口 |
| public ResultMsg modifyOrganization(OrganizationInfoVO vo); | 业务员选择修改机构信息界面的接口 |
| **public** ArrayList<OrganizationInfoVO> findOrgInfo(OrganizationInfoVO vo); | 业务员选择查找机构信息界面的接口 |
| public ArrayList<OrganizationInfoVO> showAll(); | 业务员要求查看机构所有界面的接口信息 |
| SystemUserManagementBLService | public ResultMsg add(UserVO vo); | 管理员选择新增用户信息界面的接口功能 |
| public ResultMsg delete(UserVO vo); | 管理员选择删除用户信界面的接口息功能 |
| public ResultMsg modify(UserVO vo); | 管理员选择修改用户信界面的接口息功能 |
| public ArrayList<UserVO> find(String userNum); | 管理员输入账号查找用界面的接口户信息 |
| public ResultMsg login(String userID,String password); | 用户要求登录系统界面的接口 |
| BankAccountManagementBLService | **public** ArrayList<BankAccountVO> find(BankAccountVO vo); | 财务人员要求查询账户界面的接口 |
| public ResultMsg add(AccountVO vo); | 财务人员新增查询账户界面的接口 |
| public ResultMsg delete(AccountVO vo); | 财务人员删除查询账户界面的接口 |
| public ResultMsg update(AccountVO vo); | 财务人员修改查询账户界面的接口 |
| public ArrayList<AccountVO> show(); | 财务人员查看所有查询账户界面的接口界面的接口 |

**Part5: statisticbl模块提供的接口**

|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| BaseDataBuildingBLService | **public** ResultMsg restartWithoutBasis(); | 财务人员要求进行无基界面的接口础建账操作 |
| public ResultMsg restartWithBasis(); | 财务人员要求进行以上年为基础的建账操作界面的接口 |
| **public** ResultMsg endDataBuilding(); | 财务人员结束期初建账界面的接口 |
| BusinessDataModificationBLService | ArrayList<PriceVO> getAllPrices(); | 总经理要求获取所有价格数据的接口 |
| ResultMsg inputPrice(PriceType type,**double** price); | 总经理要求修改价格常量界面的接口 |
| ResultMsg submitPrice(PriceType type, **double** price); | 总经理提交修改的城市之间距离信息的界面的接口 |
| ArrayList<DistanceVO> getAllDistanceInfo(); | 总经理要求获得所有距离数据的接口 |
| public ResultMsg inputCityDistance(String city1,String city2,double distance); | 总经理要求修改城市之间距离界面的接口 |
| ResultMsg submitCityDistance(String city1, String city2, **double** distance); | 总经理提交修改的城市之间距离信息的接口 |
| ChartOutputBLService | public ResultMsg enquiryChart(ChartType chartType,String time1,String time2); | 用户发起查看报表操作界面的接口 |
| **public** ChartVO getChartVO(ChartType chartType,String time1,String time2); | 用户获取起止时间内的相应类型的报表信息界面的接口 |
| public ResultMsg exportChart(String path); | 用户发起导出报表操作请求界面的接口 |
| LogInquiryBLService | **public** ResultMsg inputKeywords(SystemLogVO log); | 用户选择输入关键字查找日志界面的接口 |
| **public** ArrayList<SystemLogVO> getLogInfo(SystemLogVO log); | 用户获取对应关键字的系统日志界面的接口 |
| NoteApprovingBLService | **public** ArrayList<ApproveNote> getNotes() **throws** Exception; | 总经理审批所有单据的操作界面的接口 |
| **public** ResultMsg pushResults(ArrayList<ApproveNote> results) **throws** Exception; | 审批结果发布的接口（搭配getNotes使用的接受审批结果的接口，将会把审批结果上传） |

## 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，logisticbl对象负责处理营业厅、中转中心物流信息处理界面的业务逻辑；commoditybl对象负责处理库存管理界面的业务逻辑。业务逻辑层的设计如图5-3-1所示。



**图5-3-1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5-3-1-1所示。

**表5-3-1-1 业务逻辑层模块的职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| Logisticbl | 负责实现对应物流过程的界面所需要的服务 |
| Commoditybl | 负责实现对应库存管理界面所需要的服务 |
| Infobl | 负责实现对应公司人员机构信息管理界面所需要的服务 |
| Financebl | 负责实现对应财务操作界面所需要的服务 |
| Statisticbl | 负责实现对应管理公司数据界面所需要的服务 |

### 5.3.2业务逻辑模块的接口规范

业务逻辑层的接口规范如下表所示

**logisticbl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 接口类 | 方法 | 类型 | 详细信息 |
| ArrivalNoteOnServiceBLService | inputHallArrivalDoc | 语法 | public ResultMsg inputHallArrivalDoc(ArrivalNoteOnServiceVO arrialDocVO); |
| 前置条件 | 营业厅业务员要求输入到达单 |
| 后置条件 | 系统检查单据信息的格式，反馈检查结果 |
| submitHallArrivalDoc | 语法 | public ResultMsg submitHallArrivalDoc(ArrivalNoteOnServiceVO arrialDocVO)); |
| 前置条件 | 营业厅业务员核对信息后要求提交单据 |
| 后置条件 | 系统提示单据信息录入成功，并􏰁提交单据给总经理审批 |
| inputHallDeliverDoc | 语法 | public ResultMsg inputHallDeliverDoc(DeliverNoteOnServiceVO deliverDocVO); |
| 前置条件 | 营业厅业务员要求输入派件单 |
| 后置条件 | 系统检查单据信息的格式，反馈检查结果 |
| submitHallDeliverDoc | 语法 | public ResultMsg submitHallDeliverDoc(DeliverNoteOnServiceVO deliverDocVO); |
| 前置条件 | 营业厅业务员核对信息后要求提交单据 |
| 后置条件 | 系统提示单据信息录入成功，并提交单据给总经理审批 |
| ArrivalNoteOnTransitBLService | inputCenterArrivalDoc | 语法 | public ResultMsg inputCenterArrivalDoc(ArrivalNoteOnTransitVO centerArrivalDocVO); |
| 前置条件 | 中转中心业务员要求输入中转中心到达单 |
| 后置条件 | 系统检查单据信息的格式，反馈检查结果 |
| submitCenterArrivalDoc | 语法 | public ResultMsg submitCenterArrivalDoc(ArrivalNoteOnTransitVO centerArrivalDocVO); |
| 前置条件 | 中转中心业务员核对信息后要求提交单据 |
| 后置条件 | 系统提示单据信息录入成功，并提交单据给总经理审批 |
| DeliveryNoteInputBLService | inputSendDoc | 语法 | public ResultMsg inputSendDoc(DeliveryNoteVO sendDocVO); |
| 前置条件 | 快递员要求输入寄件单 |
| 后置条件 | 系统检查单据信息的格式，反馈检查结果 |
| submitSendDoc | 语法 | public sendDocMsg submitSendDoc(DeliveryNoteVO sendDocVO); |
| 前置条件 | 快递员核对信息后要求提交单据 |
| 后置条件 | 系统提示单据信息录入成功，显示寄件报价、预计到达日期,并提交单据给总经理审批 |
| LoadNoteOnServiceBLService | inputHallLoadDoc | 语法 | public ResultMsg inputHallLoadDoc(LoadNoteOnServiceVO hallLoadDocVO); |
| 前置条件 | 营业厅业务员要求输入营业厅装车单 |
| 后置条件 | 系统检查单据信息的格式，反馈检查结果 |
| submitHallLoadDoc | 语法 | public ResultMsg submitHallLoadDoc(LoadNoteOnServiceVO hallLoadDocVO); |
| 前置条件 | 营业厅业务员核对信息后要求提交单据 |
| 后置条件 | 系统提示单据信息录入成功，并提交单据给总经理审批 |
| LoadNoteOnTransitBLService | inputCenterLoadDoc | 语法 | public ResultMsg inputCenterLoadDoc(LoadNoteOnTransitVO centerLoadDocVO); |
| 前置条件 | 中转中心业务员要求输入中转中心装车单 |
| 后置条件 | 系统检查单据信息的格式，反馈检查结果 |
| submitCenterLoadDoc | 语法 | public ResultMsg submitCenterLoadDoc(LoadNoteOnTransitVO centerLoadDocVO); |
| 前置条件 | 中转中心业务员核对信息后要求提交单据 |
| 后置条件 | 系统提示单据信息录入成功，并提交单据给总经理审批 |
| ReceivingNoteInputBLService | inputReceiveDoc | 语法 | public ResultMsg inputReceiveDoc(ReceingNoteVO receiveDocVO); |
| 前置条件 | 快递员要求输入收件单 |
| 后置条件 | 系统检查单据信息的格式，反馈检查结果 |
| submitReceiveDoc | 语法 | public ResultMsg submitSecdDoc(ReceingNoteVO receiveDocVO); |
| 前置条件 | 快递员核对信息后要求提交单据 |
| 后置条件 | 系统提示单据信息录入成功，并提交单据给总经理审批 |
| TransitNoteInputBLService | inputCenterTransitDoc | 语法 | public ResultMsg inputCenterTransitDoc(TransitNoteOnTransitVO centerTransitDocVO); |
| 前置条件 | 中转中心业务员要求输入中转中心中转单 |
| 后置条件 | 系统检查单据信息的格式，反馈检查结果 |
| submitCenterTransitDoc | 语法 | public ResultMsg submitCenterTransitDoc(TransitNoteOnTransitVO centerTransitDocVO); |
| 前置条件 | 中转中心业务员核对信息后要求提交单据 |
| 后置条件 | 系统提示单据信息录入成功，并提交单据给总经理审批 |

**logisticbl需要的接口**

|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| ArrivalNoteOnServiceDataService | public void insert(ArrivalNoteOnServicePO po) throws RemoteException | 在数据库中增加一个po记录 |
| ArrivalNoteOnTransitDataService | public void insert(ArrivalNoteOnTransitPO po) throws RemoteException | 在数据库中增加一个po记录 |
| DeliveryNoteInputDataService | public void insert(DeliveryNotePOpo) throws RemoteException | 在数据库中增加一个po记录 |
| public void insert(OrderPO po) throws RemoteException | 在数据库中增加一个货物历史轨迹po记录 |
| LoadNoteOnServiceDataService | public void insert(LoadNoteOnServicePO po) throws RemoteException | 在数据库中增加一个po记录 |
| LoadNoteOnTransitDataService | public void insert(LoadNoteOnTransitPO po) throws RemoteException | 在数据库中增加一个po记录 |
| NoteDataService | public void insert(NotePO po) throws RemoteException | 在数据库中增加一个po记录 |
| ReceivingNoteInputDataService | public void insert(ReceivingNotePOpo) throws RemoteException | 在数据库中增加一个po记录 |
| TransitNoteInputDataService | public void insert(TransitNotePOpo) throws RemoteException | 在数据库中增加一个po记录 |
| OrderInquiryDataService | public OrderVO findOrder(String barcode)  throws RemoteException, ElementNotFoundException | Data返回对应单据的OrderVO |
| boolean insertOrderPO(String barCode, String orderInfo, double price) throws RemoteException; | 系统增加相应的订单信息 |

**commoditybl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 接口类 | 方法 | 类型 | 详细信息 |
| StorageInBLService | addPutInStorgaeDoc | 语法 | public ResultMsg addPutInStorgaeDoc(PutInStorageVO putInStorageVo); |
| 前置条件 | 库存管理人员输入入库单信息，选择提交 |
| 后置条件 | 系统检查单据信息格式，反馈检查结果 |
| submitPutInStorageDoc | 语法 | public ResultMsg submitPutInStorageDoc(PutInStorageVO putInStorageVo); |
| 前置条件 | 入库单格式信息没有错误 |
| 后置条件 | 系统反馈提交结果 |
| alarm | 语法 | public InventoryStatus alarm(); |
| 前置条件 | 库存超过最大警戒线 |
| 后置条件 | 系统发出提示 |
| StorageInquiryAllBLService | inputFirst | 语法 | public ResultMsg inputFirst(String firstTime); |
| 前置条件 | 本次盘点为第一次盘点 |
| 后置条件 | 系统检查时间格式是否正确，逻辑是否正确 |
| request | 语法 | public InventoryVO request () throws Exception; |
| 前置条件 | 系统允许进行本日盘点 |
| 后置条件 | 若确认盘点，则系统在当前时间生成新的盘点时间点，统计这一时间段的库存信息并显示 |
| StorageInquiryPartBLService | inputTime | 语法 | public ResultMsg inputTime(String time1,String time2); |
| 前置条件 | 库存管理人员要求进行库存查看 |
| 后置条件 | 系统在库存管理人员输入时间选择提交后，对结果进行格式检查 |
| show | 语法 | public InventoryVO show(String time1,String time2); |
| 前置条件 | 库存管理人员输入了正确的起止时间 |
| 后置条件 | 系统显示相应时间段内的库存信息 |
| StorageOutBLService | addStorageOutDoc | 语法 | public ResultMsg addStorageOutDoc(StorageOutVO storageOutVo); |
| 前置条件 | 库存管理人员要求生成出库单 |
| 后置条件 | 系统检查输入信息格式，并给出反馈 |
| submitStorageOutDoc | 语法 | public ResultMsg submitStorageOutDoc(StorageOutVO storageOutVo); |
| 前置条件 | 出库单信息格式无误 |
| 后置条件 | 系统提示单据信息录入成功，并提交单据给总经理审批 |
| StorageMapExportService | exportTo | 语法 | **public** ResultMsg exportTo(InventoryVO data,String path); |
| 前置条件 | 库存盘点获取库存信息 |
| 后置条件 | 系统提示库存快照生成成功 |

**commodityblservice需要的接口**

|  |  |  |
| --- | --- | --- |
| 接口名 | 服务名 | 详细信息 |
| InventoryDataService | public InventoryPO findByTime(String sTime,String eTime,String staffID) throws RemoteException; | 按起止时间进行查找返回相应的InventoryPO结果 |
| public InventoryPO findAll(String staffID) throws RemoteException; | 返回截止上次库存盘点所有的库存PO |
| public boolean setRecentTime(String date, String staffID) throws RemoteException | 设定首次库存盘点日期 |
| StorageInDataService | public boolean insert(StorageInPO po,String staffID) throws RemoteException | 在数据库中增加一个入库单po记录，返回值为操作是否成功 |
| StorageOutDataService | public boolean insert(StorageOutPO po,String staffID) throws RemoteException | 在数据库中增加一个出库单po记录，返回值为操作是否成功 |

**financebl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| CreditNoteInputBLService | addReceipeDoc | 语法 | public ResultMsg addReceipeDoc(CreditNoteVO vo); |
| 前置条件 | 财务人员选择新增收款单的功能 |
| 后置条件 | 系统检查收款单信息 |
| submitReceipeDoc | 语法 | public ResultMsg submitReceipeDoc(CreditNoteVO vo) |
| 前置条件 | 收款单格式检查无误 |
| 后置条件 | 系统提示单据信息录入成功，并提交单据给总经理审批 |
| getOrderMoney | 语法 | public double getOrderMoney(String barcode) throws ElementNotFoundException |
| 前置条件 | 条形码字符串经过了格式检查 |
| 后置条件 | 返回此订单的金额 |
| PaymentInputBLService | addPaymentRecord | 语法 | public ResultMsg addPaymentRecord(PaymentVO vo); |
| 前置条件 | 财务人员选择新增付款记录的功能 |
| 后置条件 | 系统检查输入信息的格式，要求财务人员核对确认 |
| submitPaymentRecord | 语法 | public ResultMsg submitPaymentRecord(PaymentVO vo); |
| 前置条件 | 付款单经过格式检验，财务人员核对无误，要求提交单据 |
| 后置条件 | 系统提示单据信息录入成功，并提交单据给总经理审批 |
| SettlementManagementBLService | addReceiveRecorder | 语法 | public ResultMsg addReceiveRecord(IncomeNoteVO vo); |
| 前置条件 | 收款单对象数据经过了格式检查 |
| 后置条件 | 系统向数据库中添加收款单记录 |
| Find | 语法 | public ArrayList<IncomeNoteVO> find(IncomeNoteVO vo); |
| 前置条件 | 输入关键字格式正确 |
| 后置条件 | 显示查询收款记录结果，系统日志记录 |

**financebl模块需要的接口**

|  |  |  |
| --- | --- | --- |
| **接口名** | **服务名** | **详细信息** |
| CreditNoteInputDataService | public boolean addCreditNote(CreditNotePO note, String staffID)  throws RemoteException,FailToPassApprovingException; | Data向数据库中添加记录 |
| public double getOrderMoney(String barcode) throws RemoteException; | 返回查询的条目金额 |
| PaymentInputDataService | public boolean addPayment(PaymentPO payment, String staffID)  throws RemoteException, FailToPassApprovingException; | Data向数据库中添加记录 |
| SettlementManagementDataService | public boolean addIncomeNote(IncomeNotePO note) throws RemoteException; | Data层增加收款单记录，修改对应账户信息 |
| public ArrayList<IncomeNotePO> findAll()  throws RemoteException; | 获取所有收款单信息 |

**infobl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 接口类 | 方法 | 类型 | 详细信息 |
| DriverVehicleManagementBLService | addVehicle | 语法 | public ResultMsg addVehicle(VehicleVOvo); |
| 前置条件 | 业务员选择新增车辆信息功能 |
| 后置条件 | 系统进行格式检查，返回格式检查结果，系统日志记录 |
| deleteVehicle | 语法 | public ResultMsg deleteVehicle(VehicleVOvo); |
| 前置条件 | 业务员选择删除车辆信息功能 |
| 后置条件 | 系统返回删除结果，系统日志记录 |
| modifyVehicle | 语法 | public ResultMsg modifyVehicle(VehicleVOvo); |
| 前置条件 | 业务员选择修改车辆信息功能 |
| 后置条件 | 系统返回修改结果，系统日志记录 |
| addVehicle | 语法 | public ResultMsg addVehicle(VehicleVOvo); |
| 前置条件 | 业务员选择新增司机信息功能 |
| 后置条件 | 系统进行格式检查，返回格式检查结果，系统日志记录 |
| deleteDriver | 语法 | public ResultMsg deleteDriver(DriverVOvo); |
| 前置条件 | 业务员选择删除司机信息功能 |
| 后置条件 | 系统返回删除结果，系统日志记录 |
| modifyDriver | 语法 | public ResultMsg modifyDriver(DriverVOvo); |
| 前置条件 | 业务员选择修改司机信息功能 |
| 后置条件 | 系统返回删除结果，系统日志记录 |
| findVehicle | 语法 | public ArrayList<VehicleVO> findVehicle(VehicleVO vo); |
| 前置条件 | 业务员选择查询车辆信息 |
| 后置条件 | 系统检查关键字格式，若无误显示查询到的信息 |
| findDriver | 语法 | public ArrayList<DriverVO> findDriver(DriverVOvo); |
| 前置条件 | 业务员选择查询司机信息 |
| 后置条件 | 系统检查关键字格式，若无误显示查询到的信息 |
| StaffOrganizationManagementBLService | addStaff | 语法 | public ResultMsg addStaff(StaffVO vo); |
| 前置条件 | 业务员选择新增人员信息 |
| 后置条件 | 更新人员信息，系统日志记录 |
| delStaff | 语法 | public ResultMsg delStaff(StaffVO vo); |
| 前置条件 | 业务员选择删除人员信息 |
| 后置条件 | 更新人员信息，系统日志记录 |
| ModifyStaff | 语法 | public ResultMsg ModifyStaff(StaffVO vo); |
| 前置条件 | 业务员选择修改人员信息 |
| 后置条件 | 更新人员信息，系统日志记录 |
| findStaffInfo | 语法 | public ArrayList<StaffVO> findStaffInfo(StaffVO vo); |
| 前置条件 | 业务员选择查找人员信息 |
| 后置条件 | 显示查找到的信息，系统日志记录 |
| addOrganization | 语法 | public ResultMsg addOrganization(OrganizationInfoVO vo); |
| 前置条件 | 业务员选择新增机构信息 |
| 后置条件 | 更新机构信息，系统日志记录 |
| delOrganization | 语法 | public ResultMsg delOrganization(OrganizationInfoVO vo); |
| 前置条件 | 业务员选择删除机构信息 |
| 后置条件 | 更新机构信息，系统日志记录 |
| modifyOrganization | 语法 | public ResultMsg delOrganization(OrganizationInfoVO vo); |
| 前置条件 | 业务员选择修改机构信息 |
| 后置条件 | 更新机构信息，系统日志记录 |
| findOrgInfo | 语法 | public ArrayList<OrganizationInfoVO> findOrgInfo(OrganizationInfoVO vo); |
| 前置条件 | 业务员选择查找机构信息 |
| 后置条件 | 显示查找到的信息，系统日志记录 |
| showAll | 语法 | public ArrayList<OrganizationInfoVO> showAll(); |
| 前置条件 | 业务员要求查看机构所有信息 |
| 后置条件 | 显示所有机构人员信息 |
| SystemUserManagementBLService | add | 语法 | public ResultMsg add(UserVO vo); |
| 前置条件 | 管理员选择新增用户信息功能 |
| 后置条件 | 系统反馈新增结果，更新用户信息，系统日志记录 |
| delete | 语法 | public ResultMsg delete(UserVO vo); |
| 前置条件 | 管理员选择删除用户信息功能 |
| 后置条件 | 系统反馈删除结果，更新用户信息，系统日志记录 |
| modify | 语法 | public ResultMsg modify(UserVO vo); |
| 前置条件 | 管理员选择修改用户信息功能 |
| 后置条件 | 系统反馈修改结果，更新用户信息，系统日志记录 |
| find | 语法 | public ArrayList<UserVO> find(UserVO vo); |
| 前置条件 | 管理员输入账号无误 |
| 后置条件 | 系统显示查找到的信息，系统日志记录 |
| logIn | 语法 | public LogInMsg login(String userID,String password); |
| 前置条件 | 用户要求登录系统，输入相应的用户名与密码 |
| 后置条件 | 系统查询相应的用户是否存在，密码是否正确，返回登录结果信息 |
| BankAccountManagementBLService | Add | 语法 | public ResultMsg add(BankAccountVO vo); |
| 前置条件 | Vo中的数据经过了格式检查 |
| 后置条件 | 向数据中添加对应银行账户信息，返回操作是否成功提示 |
| Delete | 语法 | public ResultMsg delete(BankAccountVO vo); |
| 前置条件 | Vo对应的银行账户记录在系统中已存在 |
| 后置条件 | 从数据中删除对应银行账户信息，返回操作是否成功提示 |
| Update | 语法 | public ResultMsg update(BankAccountVO vo); |
| 前置条件 | Vo对应的银行账户信息在系统数据中已存在 |
| 后置条件 | 修改对应银行账户信息（只可修改名称），返回操作是否成功 |
| Find | 语法 | public ArrayList<BankAccountVO> find(BankAccountVO vo); |
| 前置条件 | Vo中包含查询关键词 |
| 后置条件 | 返回包含查询关键词的银行账户信息 |
| show | 语法 | public ArrayList<BankAccountVO> show(); |
| 前置条件 | 财务人员已登录 |
| 后置条件 | 返回所有银行账户信息 |

**infobl模块需要的服务**

|  |  |  |
| --- | --- | --- |
| **接口名** | **服务名** | **详细信息** |
| DriverVehicleManagementDataService | public boolean addDriver(DriverPO driver) | Data向数据库中添加对应项 |
| public boolean addVehicle(VehiclePO vehicle) | Data向数据库中添加对应项 |
| public boolean removeDriver(DriverPO driver) | Data删除对应记录 |
| public boolean removeVehicle(VehiclePO vehicle) | Data删除对应记录 |
| public boolean modifyDriver(DriverPO originalDriver,DriverPO modified) | Data将originalDriver替换为modified |
| public boolean modifyVehicle(VehiclePO originalVehicle,VehiclePO modified) | Data将originalVehicle数据替换为modified |
| public ArrayList<DriverPO> getAllDriver() | Data返回所有司机数据 |
| public ArrayList<VehiclePO> getAllVehicles() | Data返回所有车辆数据 |
| public ArrayList<DriverPO> inquireDriver(String[] keywords) | Data返回所有符合司机数据 |
| public ArrayList<VehiclePO> inquireVehicle(String[] keywords) | Data返回所有符合车辆记录 |
| StaffOrganizationManagementDataService | public boolean addStaff(StaffPO staff) | Data向数据库中添加对应项 |
| public boolean addOrganization(OrganizationPO org) | Data向数据库中添加对应项 |
| public boolean removeStaff(StaffPO staff) | Data删除对应记录 |
| public boolean removeOrganization(OrganizationPO org) | Data删除对应记录 |
| public boolean modifyStaff(StaffPO original,StaffPO modified)n | Data将original替换为modified |
| public boolean modifyOrganization(OrganizationPO original,OrganizationPO modified) | Data将original数据替换为modified |
| public ArrayList<StaffPO> getAllStaff() | Data返回所有员工数据 |
| public ArrayList<OrganizationPO> getAllOrganizations() | Data返回所有机构数据 |
| public StaffPO findStaff(StaffPO po) | Data返回符合员工数据 |
| public OrganizationPO findOrganization(OrganizationPO po) | Data返回符合机构记录 |
| SystemUserManagementDataService | public boolean addUser(UserPO po) | Data向数据库中添加对应记录 |
| public boolean removeUser(UserPO user) | Data从数据库中删除对应条目 |
| public boolean modifyUser(UserPO originalUser,UserPO modified) | Data将originalUser替换为modified |
| public UserPO inquireUser(UserPO userInfo) | Data查询对应的用户并返回其UserPO |
| public ArrayList<UserPO> getAllUsers() | Data返回所有的UserPO |

**statisticbl模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| **接口名** | **方法** | **详细信息** | |
| BaseDataBuildingBLService | restartWithoutBasis | 语法 | public ResultMsg restartWithoutBasis（） |
| 前置条件 | 财务人员要求进行无基础建账操作 |
| 后置条件 | 财务人员输入账目信息 |
| restartWithBasis | 语法 | public ResultMsg restartWithBasis(); |
| 前置条件 | 财务人员要求进行以上年为基础的建账操作 |
| 后置条件 | 财务人员输入账目信息 |
| endDataBuilding | 语法 | public ResultMsg endDataBuilding(); |
| 前置条件 | 财务人员建账操作完成 |
| 后置条件 | 系统提示建账结束，系统使用新的账目信息，生成待查副本，记录日志 |
| BusinessDataModificationBLService | inputPrice | 语法 | public ResultMsg inputPrice(PriceType type,double price); |
| 前置条件 | 总经理要求修改价格信息，选择价格类型输入新的价格 |
| 后置条件 | 系统检查价格的格式正确性，如果错误给出相应提示 |
| inputCityDistance | 语法 | public ResultMsg inputCityDistance(String city1,String city2,double distance); |
| 前置条件 | 总经理要求修改城市之间距离 |
| 后置条件 | 系统检查城市以及距离的格式和逻辑正确性，如果错误则给出相应提示 |
| submitPrice | 语法 | public ResultMsg submitPrice(PriceType type, double price); |
| 前置条件 | 总经理核对新的业务数据，要求提交 |
| 后置条件 | 系统更新业务数据，只新的业务中采用新的数据 |
| submitCityDistance | 语法 | public ResultMsg submitCityDistance(String city1, String city2, double distance); |
| 前置条件 | 总经理核对新的业务数据，要求提交 |
| 后置条件 | 系统更新业务数据，只新的业务中采用新的数据 |
| getAllPrices | 语法 | PriceVO getAllPrices(); |
| 前置条件 | 总经理已被识别身份，要求修改系统常量信息 |
| 后置条件 | 显示系统中所有的价格信息 |
| getAllDistanceInfo | 语法 | DistanceVO getAllDistanceInfo(); |
| 前置条件 | 总经理已被识别身份，要求修改系统常量信息 |
| 后置条件 | 显示系统中所有的城市距离信息 |
| ChartOutputBLService | enquiryChart | 语法 | public ResultMsg enquiryChart(ChartType chartType,String time1,String time2); |
| 前置条件 | 用户发起查看报表操作 |
| 后置条件 | 系统请求用户输入统计数据的起止时间 |
| getChartVO | 语法 | public ChartVO getChartVO(ChartType chartType,String time1,String time2); |
| 前置条件 | 用户选择了报表类型并输入了正确的起止时间 |
| 后置条件 | 系统显示起止时间内的相应类型的报表信息 |
| exportChart | 语法 | public ResultMsg exportChart(String path); |
| 前置条件 | 用户发起导出报表操作请求 |
| 后置条件 | 系统向指定的位置导出相应报表 |
| LogInquiryBLService | inputKeywords | 语法 | public ResultMsg inputKeywords(String keywords); |
| 前置条件 | 用户选择查看系统日志功能 |
| 后置条件 | 系统在查询者输入关键词信息后，对信息的格式进行检查，如果错误给出相应反馈 |
| getLogInfo | 语法 | public ArrayList<SystemLogVO> getLogInfo(String keywords); |
| 前置条件 | 用户输入正确格式的关键字 |
| 后置条件 | 系统显示所有匹配日的系统志查询结果 |
| NoteApprovingBLService | getNotes | 语法 | public ArrayList<ApproveNote> getNotes() |
| 前置条件 | 总经理开始审批单据的操作 |
| 后置条件 | 返回所有未审批的单据信息 |
| pushResults | 语法 | public ResultMsg pushResults(ArrayList<ApproveNote> results) |
| 前置条件 | 总经理完成审批单据 |
| 后置条件 | 系统提示总经理单据审批成功，记录系统日志，并提示业务员单据已被审批 |
| OrderInquiryBLService | inputBarcode | 语法 | public ResultMsg inputBarcode(String inputBarcode); |
| 前置条件 | 用户输入订单条形码，要求查询单据信息 |
| 后置条件 | 返回格式检查结果 |
| submitBarcode | 语法 | public OrderVO submitBarcode(String inputBarcode); |
| 前置条件 | 用户确认查询单据 |
| 后置条件 | 返回单据的货运状态与历史轨迹 |

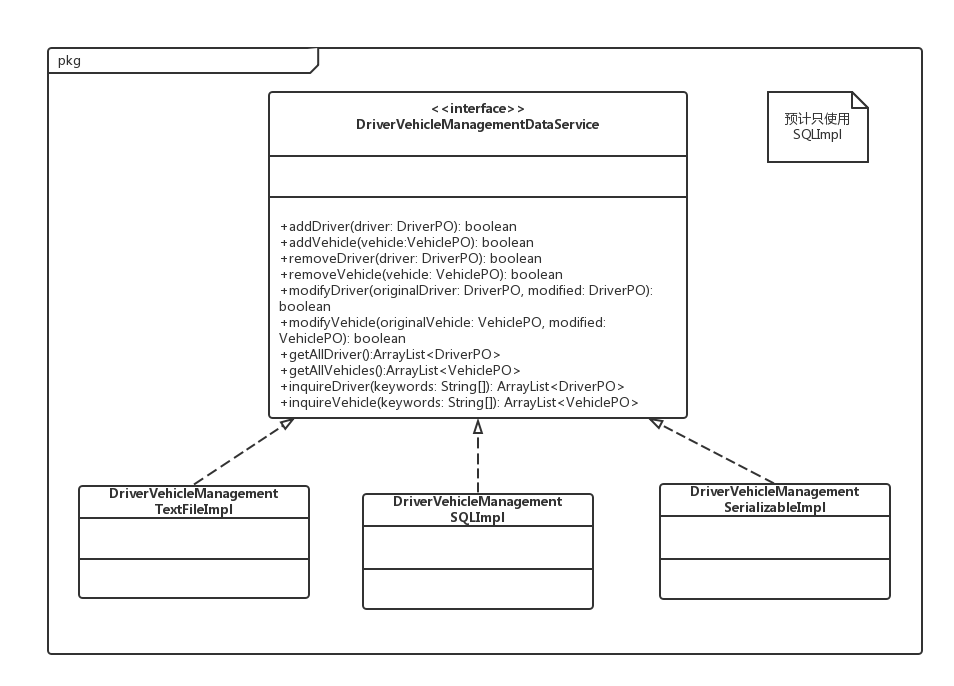
**statisticbl模块需要的服务**

|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| BaseDataBuildingDataService | public boolean startBaseDataBuilding(boolean withBase) throws RemoteException | 系统进入建账状态，其他客户端不可访问服务器 |
| public boolean endBaseDataBuilding() throws RemoteException | 系统结束期初建账，开始使用新的账目信息 |
| BusinessDataModificationDataService | public boolean setDistance(double  distance,String city1,String city2)  throws RemoteException,ElementNotFoundException | Data将city1与city2之间的距离数据改为distance |
| public boolean setPrice(String name,double newValue)  throws RemoteException,ElementNotFoundException | Data将name表示的价格条目数值改为newValue |
| public boolean addCity(String name)  throws RemoteException,InterruptWithExistedElementException | Data向数据库中添加name表示的城市记录 |
| public ArrayList<String> getAllCities()  throws RemoteException | Data返回所有城市名 |
| public double getDistance(String city1,String city2)  throws RemoteException,ElementNotFoundException | Data返回对应距离数值 |
| public double getPrice(String name)  throws RemoteException,ElementNotFoundException | Data返回对应的价格数值 |
| ChartOutputDataService | public BusinessStateChartPO getBusinessStateChart(BusinessStateChartPO po) throws RemoteException | Data返回封装了详细数据的ChartPO |
| public CostAndProfitChartPO getCostAndProfitChart(CostAndProfitChartPO po) throws RemoteException | Data返回封装了详细数据的ChartPO |
| LogInquiryDataService | public ArrayList<LogEntryPO> findLogEntries(String time,ArrayList<String> keyword)  throws RemoteException | Data层根据时间、关键词查询Log，返回相关条目 |
| OrderInquiryDataService | public OrderVO findOrder(String barcode)  throws RemoteException, ElementNotFoundException | Data返回对应单据的OrderVO |
| boolean insertOrderPO(String barCode, String orderInfo, double price) throws RemoteException; | 系统增加相应的订单信息 |

## 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。

DriverVehicleManagementbl业务逻辑需要的服务主要有DriverVehicleManageDataService接口提供。由于持久化数据的保存可能存在多种形式：Txt文件、序列化文件、数据库等，所示抽象了数据服务。数据层模块的描述具体如图5-4-1所示。



**图5-4-1管理车辆与司机信息数据层模块描述**

### 5.4.1 数据层模块的职责

数据层模块的职责如表5-4-1-1所示

**表5-4-1-1数据层模块的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| logisticDataService | 持久化物流数据库的接口，提供集体载入、集体保存、增、删、改、查服务 |
| commditydataservice | 持久化库存数据库的接口，提供集体载入、集体保存、增、删、改、查服务 |
| infoDataService | 持久化人员机构信息数据库的接口，提供集体载入、集体保存、增、删、改、查服务 |
| finceDataService | 持久化财务信息数据库的接口，提供集体载入、集体保存、增、删、改、查服务 |
| satisticDataService | 持久化业务数据数据库的接口，提供集体载入、集体保存、增、删、改、查服务 |
| Datebase | 连接数据库 |

### 5.2.4 数据层模块的接口规范

数据层模块的接口规范如下列表所示。

数据层提供的接口。

**logisticData提供的接口**

|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| ArrivalNoteOnServiceDataService | public ResultMsg insertArrivalNote(ArrivalNoteOnServicePO po)  **throws** RemoteException, ElementNotFoundException ,InterruptWithExistedElementException; | 在数据库中增加一个po记录 |
| ResultMsg insertDeliverNote(DeliverNoteOnServicePO po)  **throws** RemoteException, ElementNotFoundException,InterruptWithExistedElementException; | 在数据库中增加一个po记录 |
| ArrivalNoteOnTransitDataService | ResultMsg insert(ArrivalNoteOnTransitPO po)  **throws** RemoteException, ElementNotFoundException,InterruptWithExistedElementException; | 在数据库中增加一个po记录 |
| DeliveryNoteInputDataService | SendDocMsg insert(DeliveryNotePO po)  **throws** RemoteException, ElementNotFoundException, InterruptWithExistedElementException; | 在数据库中增加一个po记录 |
| LoadNoteOnServiceDataService | ResultMsg insert(LoadNoteOnServicePO po)  **throws** RemoteException, ElementNotFoundException, InterruptWithExistedElementException; | 在数据库中增加一个po记录 |
| LoadNoteOnTransitDataService | ResultMsg insert(LoadNoteOnTransitPO po)  **throws** RemoteException, ElementNotFoundException, InterruptWithExistedElementException; | 在数据库中增加一个po记录 |
| NoteDataService | public void insert(NotePO po) throws RemoteException | 在数据库中增加一个po记录 |
| ReceivingNoteInputDataService | ResultMsg insert(ReceivingNotePO po)  **throws** RemoteException, ElementNotFoundException, InterruptWithExistedElementException; | 在数据库中增加一个po记录 |
| TransitNoteInputDataService | ResultMsg insert(TransitNotePO po)  **throws** RemoteException, ElementNotFoundException, InterruptWithExistedElementException; | 在数据库中增加一个po记录 |
| OrderInquiryDataService | public OrderVO findOrder(String barcode)  throws RemoteException, ElementNotFoundException | Data返回对应单据的OrderVO |
| boolean insertOrderPO(String barCode, String orderInfo, double price) throws RemoteException; | 系统增加相应的订单信息 |

**financedataservice模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **接口名** | **服务名** | **详细信息** |
| CreditNoteInputDataService | **public** **boolean** addCreditNote(CreditNotePO note, String staffID)  **throws** RemoteException,FailToPassApprovingException; | Data向数据库中添加记录 |
| **public** **double** getOrderMoney(String barcode) **throws** RemoteException; | 返回查询的条目金额 |
| PaymentInputDataService | **public** **boolean** addPayment(PaymentPO payment, String staffID)  **throws** RemoteException, FailToPassApprovingException; | Data向数据库中添加记录 |
| SettlementManagementDataService | **public** **boolean** addIncomeNote(IncomeNotePO note, String staffID)  **throws** RemoteException; | Data层增加收款单记录，修改对应账户信息 |
| public ArrayList<IncomeNotePO> findAll()  throws RemoteException; | 获取所有收款单信息 |

**commoditydataservice模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 接口名 | 服务名 | 详细信息 |
| InventoryDataService | public InventoryPO findByTime(String sTime,String eTime,String staffID) throws RemoteException; | 按起止时间进行查找返回相应的InventoryPO结果 |
| public InventoryPO findAll(String staffID) throws RemoteException; | 返回截止上次库存盘点所有的库存PO |
| public boolean setRecentTime(String date, String staffID) throws RemoteException | 设定首次库存盘点日期 |
| StorageInDataService | public boolean insert(StorageInPO po,String staffID) throws RemoteException | 在数据库中增加一个入库单po记录，返回值为操作是否成功 |
| StorageOutDataService | public boolean insert(StorageOutPO po,String staffID) throws RemoteException | 在数据库中增加一个出库单po记录，返回值为操作是否成功 |

**infodataservice模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 接口名 | 服务名 | 详细信息 |
| BankAccountManagementDataService | **public** **boolean** addBankAccount(BankAccountPO account)  **throws** RemoteException, InterruptWithExistedElementException, SQLException; | Data向数据库中添加对应PO信息 |
| **public** **boolean** removeBankAccount(BankAccountPO account)  **throws** RemoteException, ElementNotFoundException, SQLException; | Data删除对应PO记录 |
| public ArrayList<BankAccountPO> findBankAccount(BankAccountPO accountInfo)  throws RemoteException, ElementNotFoundException, SQLException; | Data返回符合信息的BankAccountPO |
| public ArrayList<BankAccountPO> getAllAccounts()  throws RemoteException, ElementNotFoundException, SQLException; | Data返回所有BankAccountPO记录 |
| **public** **boolean** modifyBankAccount(BankAccountPO account)  **throws** RemoteException, ElementNotFoundException,InterruptWithExistedElementException, SQLException; | Data将originalAccount 替换为modifiedAccount |
| StaffOrganizationManagementDataService | **public** **boolean** addStaff(StaffPO staff)  **throws** RemoteException, InterruptWithExistedElementException; | 在数据库中增加一个入库单po记录，返回值为操作是否成功 |
| public boolean removeStaff(StaffPO staff)  throws RemoteException, ElementNotFoundException; | 在数据库中删除一个记录 |
| public boolean removeOrganization(OrganizationPO org)  throws RemoteException, ElementNotFoundException; | 在数据库中删除一个记录 |
| **public** **boolean** modifyStaff(StaffPO staff)  **throws** RemoteException; | 在数据库中修改意向数据 |
| public boolean modifyOrganization(OrganizationPO org)  throws RemoteException; | 在数据库中修改意向数据 |
| public ArrayList<StaffPO> findStaff(StaffPO info)  throws RemoteException, ElementNotFoundException; | 从数据库中查找一个记录 |
| **public** ArrayList<OrganizationPO> findOrganization(OrganizationPO info)  **throws** RemoteException, ElementNotFoundException; | 从数据库中查找一个记录 |
| public ArrayList<StaffPO> getAllStaff()  throws RemoteException; | 从数据库中获取所有人员信息 |
| public ArrayList<OrganizationPO> getAllOrganizations()  throws RemoteException; | 从数据库中获取所有机构信息 |
| **public** **boolean** addOrganization(OrganizationPO org)  **throws** RemoteException, InterruptWithExistedElementException; | 在数据库中增加一个入库单po记录，返回值为操作是否成功 |
| DriverVehicleManagementDataService | **public** **boolean** addDriver(DriverPO driver)  **throws** RemoteException,InterruptWithExistedElementException, SQLException; | 向数据库中添加对应司机PO项 |
| **public** **boolean** addVehicle(VehiclePO vehicle)  **throws** RemoteException,InterruptWithExistedElementException, SQLException, IOException; | Data向数据库中添加对应汽车PO项 |
| **public** **boolean** removeDriver(DriverPO driver)  **throws** RemoteException,ElementNotFoundException, SQLException; | Data删除对应司机PO记录 |
| **public** **boolean** removeVehicle(VehiclePO vehicle)  **throws** RemoteException,ElementNotFoundException, SQLException, IOException; | Data删除对应汽车PO记录 |
| **public** **boolean** modifyDriver(DriverPO driver)  **throws** RemoteException,ElementNotFoundException,InterruptWithExistedElementException, SQLException; | Data将originalDriver替换为modified |
| **public** **boolean** modifyVehicle(VehiclePO originalVehicle)  **throws** RemoteException,ElementNotFoundException,InterruptWithExistedElementException, SQLException, IOException; | Data将originalVehicle数据替换为modified |
| **public** ArrayList<DriverPO> getAllDriver()  **throws** RemoteException, SQLException; | Data返回所有司机数据 |
| **public** ArrayList<VehiclePO> getAllVehicles()  **throws** RemoteException, SQLException, IOException; | Data返回所有车辆数据 |
| **public** ArrayList<DriverPO> inquireDriver(DriverPO keywords)  **throws** RemoteException,ElementNotFoundException, SQLException; | Data返回所有符合关键字的司机数据 |
| **public** ArrayList<VehiclePO> inquireVehicle(VehiclePO keywords)  **throws** RemoteException, SQLException, IOException; | Data返回所有符合关键字的车辆数据 |
| SystemUserManagementDataService | public boolean addUser(UserPO user)  throws RemoteException, InterruptWithExistedElementException; | Data向数据库中添加对应UserPO记录 |
| public boolean removeUser(UserPO user)  throws RemoteException, ElementNotFoundException; | Data从数据库中删除对应UserPO条目 |
| public boolean modifyUser(UserPO originalUser,UserPO modified)  throws RemoteException, ElementNotFoundException, InterruptWithExistedElementException; | Data查询对应的用户并返回其UserPO |
| public ArrayList<UserPO> inquireUser(UserPO info)  throws RemoteException, ElementNotFoundException; | Data查询对应的用户并返回其UserPO |
| public ArrayList<UserPO> getAllUsers()  throws RemoteException, ElementNotFoundException; | Data返回所有的UserPO |
| public LogInMsg logIn(String account,String password) throws RemoteException; | 登录接口 |

**StatisticData提供的接口**

|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| BaseDataBuildingDataService | public boolean startBaseDataBuilding(boolean withBase) throws RemoteException | 系统进入建账状态，其他客户端不可访问服务器 |
| public boolean endBaseDataBuilding() throws RemoteException | 系统结束期初建账，开始使用新的账目信息 |
| BusinessDataModificationDataService | public boolean setDistance(double  distance,String city1,String city2)  throws RemoteException,ElementNotFoundException | Data将city1与city2之间的距离数据改为distance |
| public boolean setPrice(String name,double newValue)  throws RemoteException,ElementNotFoundException | Data将name表示的价格条目数值改为newValue |
| public boolean addCity(String name)  throws RemoteException,InterruptWithExistedElementException | Data向数据库中添加name表示的城市记录 |
| public ArrayList<String> getAllCities()  throws RemoteException | Data返回所有城市名 |
| public double getDistance(String city1,String city2)  throws RemoteException,ElementNotFoundException | Data返回对应距离数值 |
| public double getPrice(String name)  throws RemoteException,ElementNotFoundException | Data返回对应的价格数值 |
| ChartOutputDataService | public BusinessStateChartPO getBusinessStateChart(BusinessStateChartPO po) throws RemoteException | Data返回封装了详细数据的ChartPO |
| public CostAndProfitChartPO getCostAndProfitChart(CostAndProfitChartPO po) throws RemoteException | Data返回封装了详细数据的ChartPO |
| LogInquiryDataService | public ArrayList<LogEntryPO> findLogEntries(String time,ArrayList<String> keyword)  throws RemoteException | Data层根据时间、关键词查询Log，返回相关条目 |
| OrderInquiryDataService | public OrderVO findOrder(String barcode)  throws RemoteException, ElementNotFoundException | Data返回对应单据的OrderVO |
| boolean insertOrderPO(String barCode, String orderInfo, double price) throws RemoteException; | 系统增加相应的订单信息 |

5.5 数据库模块的接口规范

ConnectionPool提供的接口

|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| ConnectionPool | public static ConnectionPool getInstance() | 获取连接池的单例 |
| public Connection getConnection(String poolname, String caller) | 从连接池中获取新的连接 |
| public void freeConnection(Connection connection) | 返回一个数据库连接到池中，再由线程返回连接池 |
| public void release() | 释放所有连接,创建新池 |

ConnectionManager提供的接口

|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| ConnectionManager | public static boolean releaseConnection(Connection connection, Statement statement, ResultSet resultSet) | 数据层的实现类返还连接 |
| public static Connection getConnection() | 数据层的实现类获取新的数据库连接 |

UsingConnection提供的接口

|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| UsingConnection | public String getCaller() | 获取当前被使用的数据库连接的使用者 |
| public long getUsingTime() | 获取当前被使用的数据库连接的使用时间 |

DatabaseFactory

|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| DatabaseFactory | InventoryDataService getInventoryDataService(); | 从数据层具体实现工厂中获取InventoryDataService 的实现类 |
| PaymentInputDataService getPaymentInputDataService(); | 从数据层具体实现工厂中获取PaymentInputDataService 的实现类 |
| CreditNoteInputDataService getCreditNoteInputDataService(); | 从数据层具体实现工厂中获取CreditNoteInputDataService 的实现类 |
| StorageOutDataService getStorageOutDataService(); | 从数据层具体实现工厂中获取StorageOutDataService 的实现类 |
| StorageInDataService getStorageInDataService(); | 从数据层具体实现工厂中获取StorageInDataService 的实现类 |
| SettlementManagementDataService getSettlementManagementDataService(); | 从数据层具体实现工厂中获取SettlementManagementDataService 的实现类 |
| BankAccountManagementDataService getBankAccountManagementDataService(); | 从数据层具体实现工厂中获取BankAccountManagementDataService 的实现类 |
| DriverVehicleManagementDataService getDriverVehicleManagementDataService(); | 从数据层具体实现工厂中获取DriverVehicleManagementDataService 的实现类 |
| StaffOrganizationManagementDataService getStaffOrganizationManagementDataService(); | 从数据层具体实现工厂中获取StaffOrganizationManagementDataService 的实现类 |
| SystemUserManagementDataService getSystemUserManagementDataService(); | 从数据层具体实现工厂中获取SystemUserManagementDataService 的实现类 |
| ArrivalNoteOnServiceDataService getArrivalNoteOnServiceDataService(); | 从数据层具体实现工厂中获取ArrivalNoteOnServiceDataService 的实现类 |
| ArrivalNoteOnTransitDataService getArrivalNoteOnTransitDataService(); | 从数据层具体实现工厂中获取ArrivalNoteOnTransitDataService 的实现类 |
| DeliveryNoteInputDataService getDeliveryNoteInputDataService(); | 从数据层具体实现工厂中获取DeliveryNoteInputDataService 的实现类 |
| LoadNoteOnServiceDataService getLoadNoteOnServiceDataService(); | 从数据层具体实现工厂中获取LoadNoteOnServiceDataService 的实现类 |
| LoadNoteOnTransitDataService getLoadNoteOnTransitDataService(); | 从数据层具体实现工厂中获取LoadNoteOnTransitDataService 的实现类 |
| ReceivingNoteInputDataService getReceivingNoteInputDataService(); | 从数据层具体实现工厂中获取ReceivingNoteInputDataService 的实现类 |
| TransitNoteInputDataService getTransitNoteInputDataService(); | 从数据层具体实现工厂中获取TransitNoteInputDataService 的实现类 |
| BaseDataBuildingDataService getBaseDataBuildingDataService(); | 从数据层具体实现工厂中获取BaseDataBuildingDataService 的实现类 |
| BusinessDataModificationDataService getBusinessDataModificationDataService(); | 从数据层具体实现工厂中获取BusinessDataModificationDataService 的实现类 |
| ChartOutputDataService getChartOutputDataService(); | 从数据层具体实现工厂中获取ChartOutputDataService 的实现类 |
| LogInquiryDataService getLogInquiryDataService(); | 从数据层具体实现工厂中获取LogInquiryDataService 的实现类 |
| NoteApprovingDataService getNoteApprovingDataService(); | 从数据层具体实现工厂中获取NoteApprovingDataService 的实现类 |
| OrderInquiryDataService getOrderInquiryDataService(); | 从数据层具体实现工厂中获取OrderInquiryDataService 的实现类 |
| LogInsertDataService getLogInsertDataService(); | 从数据层具体实现工厂中获取LogInsertDataService 的实现类 |

DatabaseConfig提供的接口

|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| DatabaseConfig | public HashMap<String, String> getParams() | 根据关键字获取数据库的配置信息,配置信息共有dbDriver,dbLocation,dbUnicode,dbEncoding,dbUser,dbPassword 6条信息 |

# 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。

* ArrivalNoteOnServicePO类包含货物到达日期、中转单编号、出发地、货物到达状态
* ArrivalNoteOnTransitPO类包含中转中心编号、货物到达日期、中转单编号、出发地、 货物到达状态。
* BankAccountPO类包含账号、账户名、账户余额。
* ChartPO类包含报表的类型、查询内容开始时间、查询内容结束时间。
* CreditNotePO类包含收款日期、金额、收款快递员姓名
* DeliveryNoteInputPO类包含客户姓名、客户地址、客户单位地址、客户电话（手机）号码、寄件件数、货物实际重量、货物体积、货物内件品名、快递种类、货物条形码、包装费用
* DriverPO类包含司机编号、营业厅编号、姓名、出生日期、身份证号、手机号、性别、行驶证期限
* IncomeNotePO类包含收款日期、收款单位、收款人、交款营业厅、金额、地点、收款银行账户。
* InventoryPO类包含入库数量、出库数量、库存金额、货物位置信息。
* LoadNoteOnServicePO类包含本营业厅编号、汽运编号、目的地、车辆代号、监装员信息、押运员信息、货物条形码。
* LoadNoteOnTransitPO类包含装车日期、本中转中心汽运编号、目的地（营业厅）、车辆代号、监装员信息、押运员信息、装箱所有货物条形码。
* OrderPO类包含条形码、货运状态、历史轨迹。
* LogEntryPO类包含操作发生时间、操作内容记录。
* OrganizationPO类包含机构类型、机构编码、机构名称、机构员工信息
* PaymentPO类包含日期、付款人姓名、银行账号、金额、付款类型、备注。
* ReceivingNoteInputPO类包含类包含收件单号、收件人姓名、收件时间。
* StaffPO类包含姓名、单位、身份证号、工资、手机号、职位、工作时间。
* StorageInPO类包含日期、货物信息。
* StorageOutPO类包含日期、货物信息，包括条形码及其位置信息、
* TransitNoteInputPO类包含装车日期、中转单编号、航班号、出发地、到达地、货柜号、监装员、装箱所有托运单号、运费。
* UserPO类包含账号、密码、权限。
* DistancePO类包含城市名称及城市间距离。
* VehiclePO类包含汽运编号、开始使用时间、所属营业厅编号、车辆图片。
* BusinessStateChartPO类为ChartPO的子类，经营情况表。
* CostAndProfitChartPO类为ChartPO的子类，成本收益表。

注：具体PO的定义，参见项目代码PO包。

## 6.2数据库表

**数据库表**

6.2数据库表

数据库表

数据库表中包BankAccount表 、Chart表 、CreditNote表 、Driver 表 、IncomeNote表 、 LogEntry表 、 Order表 、 Organization 表 、 Payment表 、 Staff表 、 User表 、 Vehicle表 、 ArrivalNoteOnService表 、ArrivalNoteOnTransit表 、 DeliveryNoteInput 表 、 Inventory表 、LoadNoteOnService表 、 LoadNoteOnTransit表 、 ReceivingNoteInput表 、 StorageIn 表 、 StorageOut表 、 TransitNoteInput表 。如下所示：

6.2数据库表

数据库表

6.2数据库表

数据库表

数据库表中包bankaccounts表 、businessprice表 、creditnotes表 、distance表、drivers 表 、goodsorder表、incomenotes表 、inoutinfo表 、note\_arrival\_on\_service表 、 note\_arrival\_on\_transit表 、note\_delivery表 、note\_delivery\_on\_service表 、note\_load\_on\_service表 、note\_load\_on\_transit表 、note\_receive\_note表 、note\_transit表 、 organization表 、paymentnotes表 、recentinquirytime表 、 staff表 、 storageinnote 表 、 storageoutnote 表 、systemlog表 、 user表 、vehicles表。如下所示：

bankaccounts表

name；number；balance

李明；0123456789；10

businessprice表

price；type

5；paperbox

creditnotes表

date；money；name；submitter；isPassed；advice；id

2015-11-11；10；李明;张三；2；；0001

distance表

city1；city2；distance；id

南京；北京；900；1

drivers表

staffID；licenseDate

025001002；2015-10-10

goodsorder表

barcode；stateOfTransport；history；money；isInStorage；warehouseID

0000000011； 完整；2015-12-24 18:38:34,货物已被快递员 收件的快递员 签收;；27.630434782608695； ；

incomenotes表

date；receiveOrg；receiveStaff；payer；money；account；isPassed；advice；id；submitter

2015-12-20；南京总部财务部；财务人员6；南京浦口营业厅；100；收款账户6；2；；1231111；

inoutinfo表

orderID；warehouseID；isIN；Date；areaCode；rowNumber；frameNumber；placeNumber；Destination

1234567890；南京第一仓库；1；2015-12-08；航空区；0001；0001；0001；北京

note\_arrival\_on\_service表

date；arrivalKind；TransferNumber； from；barcodes；isPassed；advice

2015-12-09；'中转到达'；025000199601290000009；南极；1357908642；2；

note\_arrival\_on\_transit表

TransferNumber；centerNumber；date；departurePlace；barcodes；isPassed；advice

025000201512120000080；0251000；2015-12-08；南京；0000000003；2；

note\_delivery表

senderName；sendAddress；senderTeleNumber；receiverName；receiverAddress；reveiverTeleNum；name；goodsNumber；weight；volume；category；packPrice；barcode；isPassed；advice

王二狗；江苏省南京市栖霞区南京大学仙林校区；150052120000；张三；上海市第三中学语文组；19883490000；书；10；10；2；特快专递；Bag；0000000001；2；

note\_delivery\_on\_service表

id；date；barcodes；deliveryMan；isPassed；advice

0000000001；2015-12-07；0000000003；张三；2；

note\_load\_on\_service表

date；hallNumber；transpotationNumber；destination；carNumber；guardMan；supercargoMan；barcodes；transferPrice ；isPassed；advice

2014-12-12；0251000；02510002012121200000；南京；025000000；李明；张三；0000000002；；2；

note\_load\_on\_transit表

date；hallNumber；transpotationNumber；destination；carNumber；guardMan；supercargoMan；barcodes； isPassed；advice；

2014-12-12；025010222；025000201509090000002；北京；025000001；张三；黎明；0000000002；2；

note\_receive\_note表

barcode；receiveCustomer；time； isPassed；advice；

0000000001；黎明；2015-12-24 18:40；2；

note\_transit表

中转中心编号；货物到达日期；中转单编号；出发地；货物到达状态;状态

025000；2015-11-11；025000201510120000003；北京；完整;草稿状态

organization表

客户姓名；客户地址；客户单位地址；客户电话（手机）号码；寄件件数；货物实际重量；货物体积；货物内件品名；快递种类；包装费用；货物条形码;状态

张三；南京市仙林区南京大学；南京大学；0111-3344556；1； 2； 0.01；书；经济快递；10； 1234567890；1234567890;草稿状态

paymentnotes表

出库数量；入库数量；时间段内货物金额；货物位置信息

3；3；100；航空区0002排0002架0002位

recentinquirytime表

本营业厅编号；汽运编号；目的地；车辆代号；监装员信息；押运员信息；货物条形码;状态

0251000；02510002015092100000；北京；025000000；李明；张三；1234567890;草稿状态

staff表

装车日期；本中转中心汽运编号；目的地（营业厅）；车辆代号；监装员信息；押运员信息；装箱所有货物条形码;状态

2015-09-09；025000201509091234567；六合营业厅；025010222；吴老二；铁三胖

；1234567890、1234567891、1234567892;草稿状态

storageinnote表

收件单号；收件人姓名；收件时间;状态

1234567890；张三；2015-09-09;草稿状态

storageoutnote表

日期；条形码；区号；目的地；排号；架号；位号;状态

2015-11-11；0123456789；航空区；南京；0002；0002；0002;草稿状态

systemlog表

日期；快递编号；目的地；装运形式；中转单编号;状态

2015-11-11；0123456789；南京；铁路；025000201510120000003;草稿状态

user表

装车日期；中转单编号；出发地；到达地；货柜号；监装员；装箱所有托运单号；运费；中转单种；类航班号；状态

20015-10-12；025000201510120000003；南京；北京；001；李明；0123456789，1000000000， 1000000001，1000000002；100；飞机；MF8190;草稿状态

李明；\*\*\*\*\*\*(db自带的不可逆加密算法，存储的是加密字符串)；总经理

vehicles表

条形码；区号；目的地；排号；架号；位号

1234567890；航空区；南京；0002；0002；0002