|  |
| --- |
| B401 |
| 快递物流系统 |
| 软件体系结构描述文档 |

|  |
| --- |
| 2015-10-20 |

目录

[1.引言 2](#_Toc433562361)

[1.1编制目的 2](#_Toc433562362)

[1.2词汇表 2](#_Toc433562363)

[1.3参考资料 2](#_Toc433562364)

[2.产品概述 2](#_Toc433562365)

[3.逻辑视角 2](#_Toc433562366)

[4.组合视角 4](#_Toc433562367)

[4.1开发包图 4](#_Toc433562368)

[4.2运行时进程 6](#_Toc433562369)

[4.3物理部署 7](#_Toc433562370)

[5.接口视角 7](#_Toc433562371)

[5.1模块的职责 7](#_Toc433562372)

[5.2用户界面层的分解 9](#_Toc433562373)

[5.3业务逻辑层的分解 10](#_Toc433562374)

[5.3.1业务逻辑层模块的职责 11](#_Toc433562375)

[5.3.2业务逻辑层模块的接口规范 11](#_Toc433562376)

[5.4数据层的分解 18](#_Toc433562377)

[5.4.1数据层模块的职责 19](#_Toc433562378)

[5.4.2数据层模块的接口规范 19](#_Toc433562379)

[6. 信息视角 23](#_Toc433562380)

[6.1数据持久化对象 23](#_Toc433562381)

[6.2Txt持久化格式 23](#_Toc433562382)

[6.3数据库表 23](#_Toc433562383)

[数据库中包含User表、Commodity表、Bills 23](#_Toc433562384)

# 1.引言

## 1.1编制目的

本报告详细完成对快递物流系统的概要设计，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** |
| LCMS | 物流公司管理系统 |  |
|  |  |  |

## 1.3参考资料

# 2.产品概述

参考快递物流系统用例文档和快递物流系统软件需求规格说明中对产品的概括描述。

# 3.逻辑视角

快递物流系统中，选择了分层体系结构风格，将系统分为三层，能够很好地示意整个高层抽象。展示层包含GUI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如下图1和图2所示。

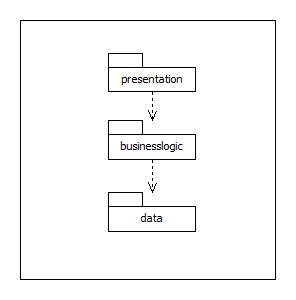


图1 参照体系结构风格的包图表达逻辑视角

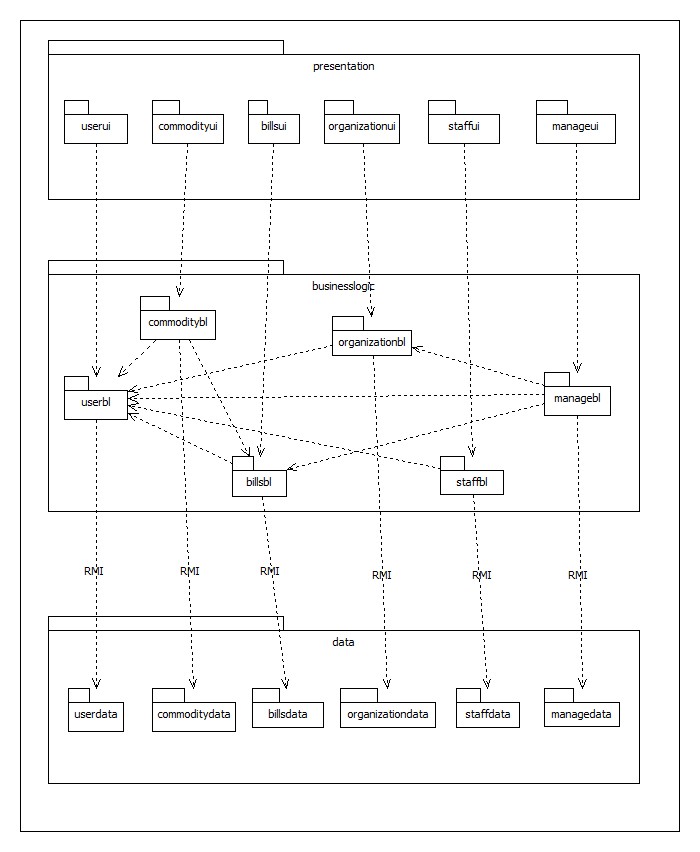


图2 软件体系结构逻辑设计方案

# 4.组合视角

## 4.1开发包图

物流管理系统的最终开发包设计如表1所示。

|  |  |
| --- | --- |
| 开发（物理）包 | 依赖的其他开发包 |
| mainui | userui,commodityui,billsui,organizationui,staffui,manageui,vo |
| userui | userblservice,界面类库包 |
| userblservice |  |
| userbl | userblservice,userdataservice,userPO |
| userdataservice | Java RMI,po |
| userdata | RMI,po,databaseutility |
| commodityui | commodityblservice,界面类库包，vo |
| commodityblservice |  |
| commoditybl | commodityblservice,commoditydataserveice,po,userbl,billsbl |
| commoditydataservice | Java RMI,po |
| commoditydata | RMI,po,databaseutility |
| billsui | billsblservice,界面类库包,vo |
| billsblservice |  |
| billsbl | billsblservice,billsdataservice,po,userbl |
| billsdataservice | Java RMI,po |
| billsda | RMI,po,databaseutility |
| organizationui | organizationblservice,界面类库包,vo |
| organizationblservice |  |
| organizationbl | organizationblservice,organizationdataservice,po,userbl |
| organizationdataservice | Java RMI,po |
| organizationdata | RMI,po,databaseutility |
| staffui | staffblservice,界面类库包,vo |
| staffblservice |  |
| staffbl | staffblservice,staffdataservice,po,userbl |
| staffdataservice | Java RMI,po |
| staffdata | RMI,po,databaseutility |
| manageui | manageblservice,界面类库包,vo |
| manageblservice |  |
| managebl | manageblservice,managedataservice,po,userbl,organizationbl,billsbl |
| managedataservice | Java RMI,po |
| managedata | RMI,po,databaseutility |
| vo |  |
| po |  |
| utilitybl |  |
| 界面类库包 |  |
| Java RMI |  |
| databaseutility |  |

物流管理系统客户端开发包如图3所示，服务器端开发包如图4所示。

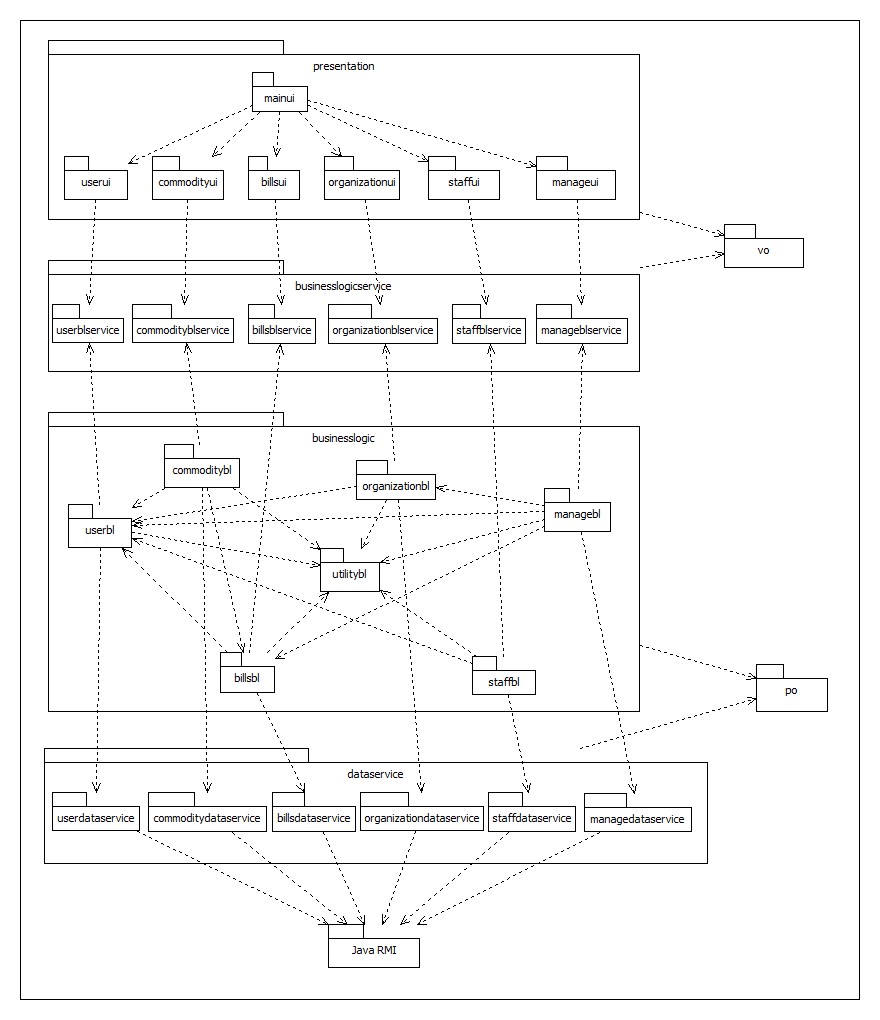


图3 客户端开发包图

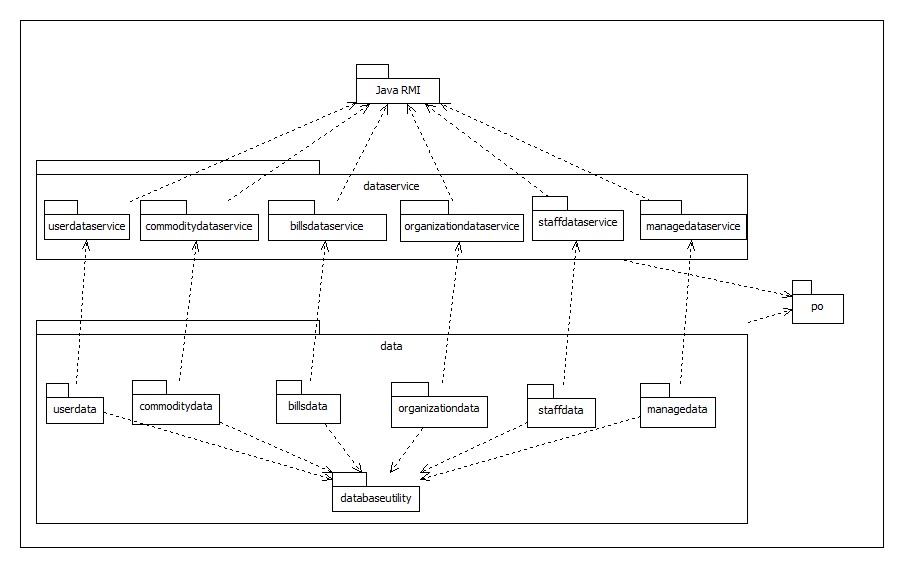


图4 服务器端开发包图

## 4.2运行时进程

在快递物流系统中，会有多个客户端进程和一个服务器端进程，其进程如下图所示。结合部署图，客户端进程是在客户端机器上运行，服务器端进程在服务器端机器上运行。

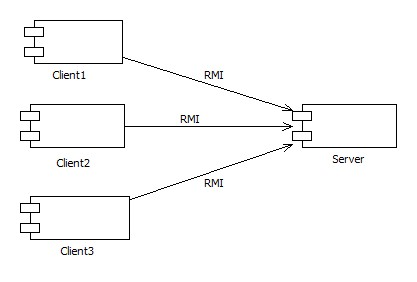
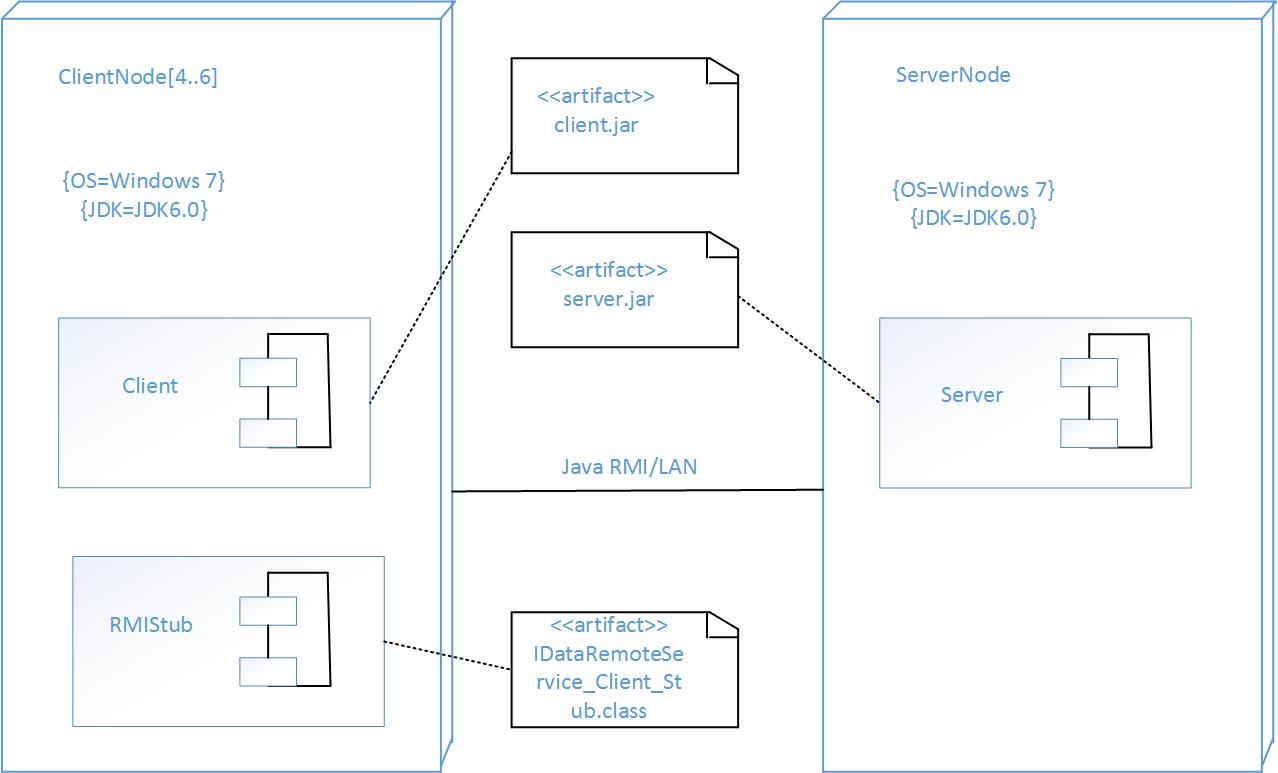


图5 进程图

## 4.3物理部署

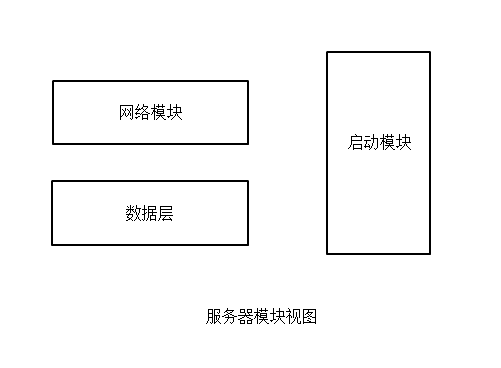
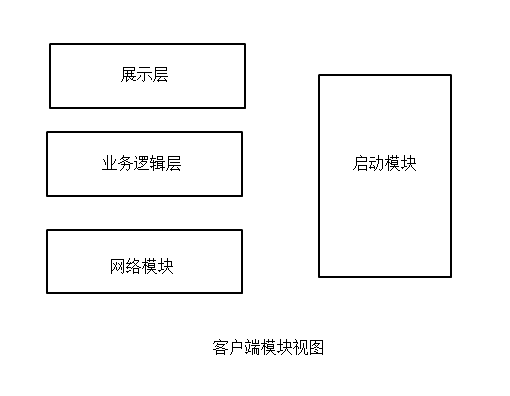
快递物流系统中客户端构件是放在客户端机器上，服务器端构件是放在服务器端机器上。在客户端节点上，还要部署RMIStub构件。



# 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如图所示。客户端各层和服务器端各层的职责如表所示。



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 | 基于窗口的快递物流公司客户端用户界面 |
| 业务逻辑层 | 对于用户界面的输入进行响应并进行业务处理逻辑 |
| 客户端网络模块 | 利用Java RMI机制查找RMI服务 |

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。层与层间调用的接口如下表所示。

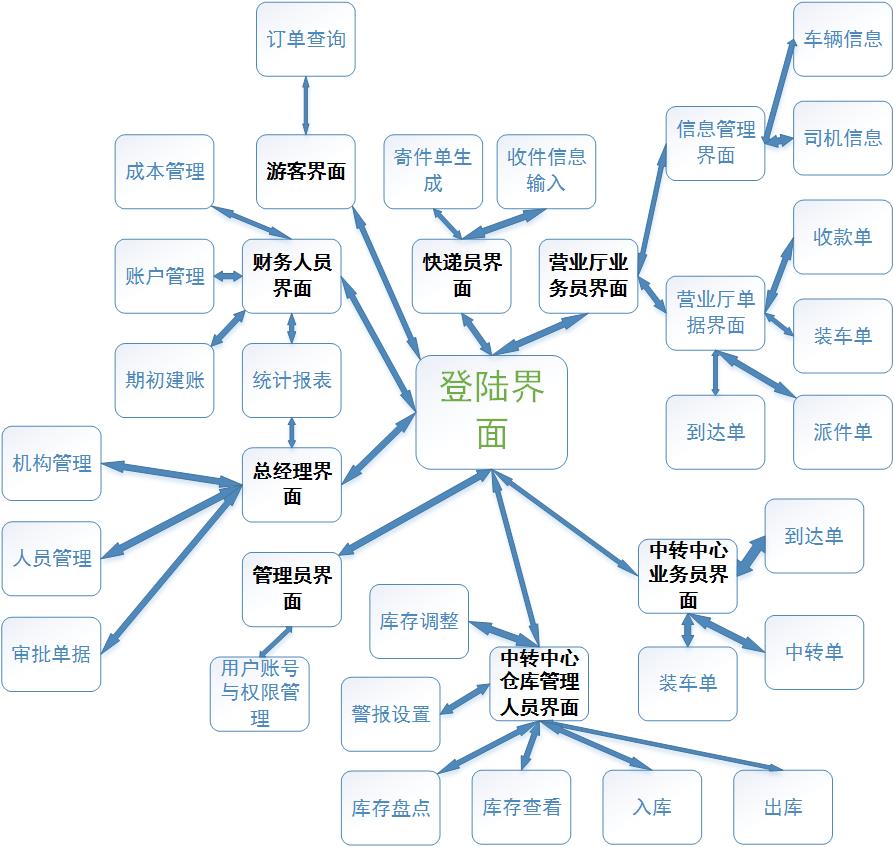
层之间调用的接口

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
|  | 客户端展示层 | 客户端业务逻辑层 |
|  | 客户端业务逻辑层 | 服务器端数据层 |

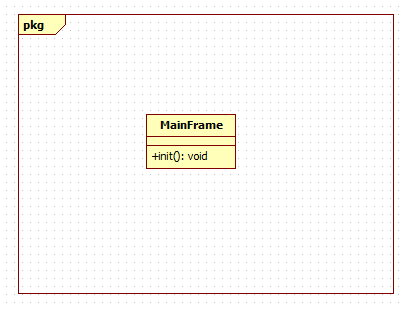
借用销售用例来说明层之间的调用，如下图所示。每一层之间都是由上层依赖了一个接口（需接口），而下层实现了这个借口（供接口）。

## 5.2用户界面层的分解

根据需求，系统界面跳转如下图所示。



服务器端和客户端的用户界面设计是一致的，只是具体的页面不一样。用户界面类如图所示。



5.2.1用户界面层模块的职责

如下表所示为用户界面层模块的职责

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

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下表所示

|  |  |  |
| --- | --- | --- |
| MainFrame | 语法 | Init（args:String[]） |
| 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下表所示

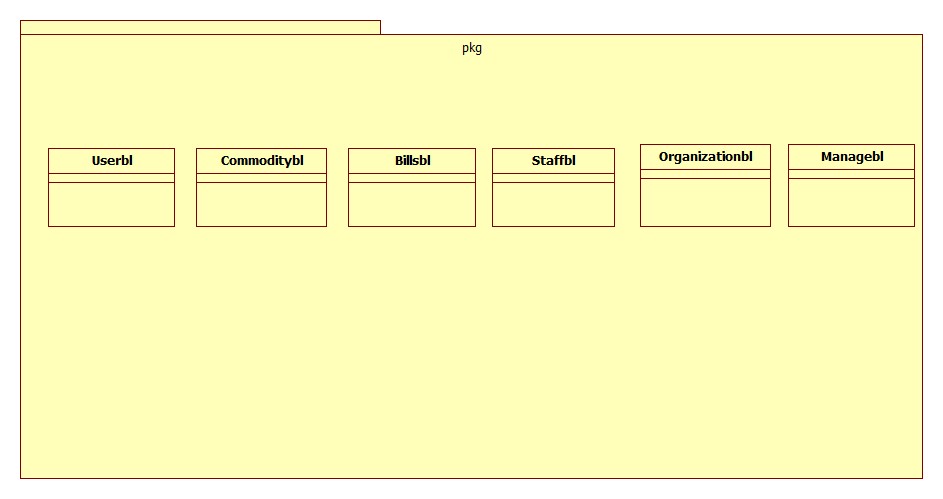
|  |  |
| --- | --- |
| 服务名 | 服务 |
| Businesslogicservice.LoginBLService | 登录界面的业务逻辑接口 |
| Businesslogicservice.\*BLService | 每个界面都有一个相应的业务逻辑接口 |

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。业务逻辑层的设计如下图所示。



### 5.3.1业务逻辑层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| Userbl | 负责实现对应与登录界面所需要的服务 |
| Commoditybl | 负责实现库存界面所需要的服务 |
| Billsbl | 负责实现单据界面所需要的服务 |
| Organizationbl | 负责实现机构界面所需要的服务 |
| Staff | 负责实现人员管理界面所需的服务 |
| Managebl | 负责实现经营参数、财务人员和总经理活动界面所需要的服务 |

### 5.3.2业务逻辑层模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| Userbl模块的接口规范 | | | |
| 提供的服务（供接口） | | | |
| User.login | 语法 | Public ResultMessage login(long id,String password); | |
| 前置条件 | Password符合输入规范 | |
| 后置条件 | 查找是否存在相应的User，根据输入的password返回登录验证的结果 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| UserDataService.insert(UserPO po) | | | 在数据库中插入UserPO对象 |
|  | | |  |

|  |  |  |
| --- | --- | --- |
| Commoditybl模块的接口规范 | | |
| 提供的服务（供接口） | | |
| Commodity.startCheckCommodity | 语法 | Public boolean startCheckCommodity(myDate time); |
| 前置条件 | 启动库存盘点 |
| 后置条件 | 冻结入库操作，返回盘点结果 |
| Commodity.checkCommodityInf | 语法 | Public ComVO checkCommodityInf(); |
| 前置条件 | 启动过startCheckCommodity并且获得盘点结果 |
| 后置条件 | 显示仓库所有库存货物信息（快递编号、入库日期、目的地、区号、排号、架号、位号），返回相应结果 |
| Commodity.examineCommodity | 语法 | Public int[] examineCommodity(myDate timestart,myDate timeend); |
| 前置条件 | 启动库存查看 |
| 后置条件 | 系统显示两个时间之间的入库数量、金额、出库数量、金额。返回相应结果 |
| Commodity.settingAlert | 语法 | Public boolean settingAlert(double alertLine); |
| 前置条件 | 启动警报设置 |
| 后置条件 | 系统更改警戒线数值，并且在入库操作后库存超过警戒线时发出警报并且返回相应结果 |
|  |  |  |
| Commodity.in | 语法 | Public boolean Commodityin(long Sending) |
| 前置条件 | 入库单生成 |
| 后置条件 | 数据库保存托运单号，并返回操作结果 |
| Commodity.out | 语法 | Public boolean Commodityout(long Sending) |
| 前置条件 | 出库单生成 |
| 后置条件 | 数据库删除托运单号，并返回操作结果 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| CommodityDataService.addCom(ComPO po) | 在数据库中加入ComPO对象 | |
| CommodityDataService.delCom(long GoodsNum) | 在数据库中删除ComPO对象 | |
| CommodityDataService.findCom(long GoodsNum) | 在数据库中查找COmPO对象 | |

|  |  |  |  |
| --- | --- | --- | --- |
| Billsbl模块的接口规范 | | | |
| 提供的服务（供接口） | | | |
| Bills.newCenterArriveBill | 语法 | Public ResultMessage newCenterArriveBill(String CenterNum,String BillNum,myDate date,String State); | |
| 前置条件 | 启动中转中心到达单生成 | |
| 后置条件 | 系统将生成的单据添加到数据库 | |
| Bills.newCenterFreightBill | 语法 | Public ResultMessage newCenterFreightBill(String BillNum,myDate date,String ContainerNum,String Means,String Name,String StartPlace,String EndPlace,long[] GoodsNum); | |
| 前置条件 | 启动中转中心中转单生成 | |
| 后置条件 | 系统将生成的单据添加到数据库 | |
| Bills.newCenterEntruckBill | 语法 | Public ResultMessage newCenterEntruckBill(String TravelNum,myDate date,String TrafficNum,String HallNum,String Name,long[] GoodsNum); | |
| 前置条件 | 启动中转中心装车单生成 | |
| 后置条件 | 系统将生成的单据添加到数据库 | |
| Bills.newInstorageBill | 语法 | Public ResultMessage newInstorageBill(long GoodsNum, myDate date,String Place,String LocationNum); | |
| 前置条件 | 启动入库单生成 | |
| 后置条件 | 系统将生成的单据添加到数据库 | |
| Bills.newOutstorageBill | 语法 | Public ResultMessage newOutstorageBill(long GoodsNum, myDate date,String FreightNum); | |
| 前置条件 | 启动出库单生成 | |
| 后置条件 | 系统将生成的单据添加到数据库 | |
| Bills.newHallArrivingBill | 语法 | Public ResultMessage newHallArrivingBill(myDate date,String FreightNum,String StartPlace,String State); | |
| 前置条件 | 启动营业厅到达单生成 | |
| 后置条件 | 系统将生成的单据添加到数据库 | |
| Bills.newHallDispatchBill | 语法 | Public ResultMessage newHallDispatchBill(myDate date,long GoodsNum,String name); | |
| 前置条件 | 启动营业厅派件单生成 | |
| 后置条件 | 系统将生成的单据添加到数据库 | |
| Bills.newHallColletcionBill | 语法 | Public ResultMessage newHallColletcionBill(myDate date,int Cash,String name,long[] GoodsNum); | |
| 前置条件 | 启动营业厅收款单生成 | |
| 后置条件 | 系统将生成的单据添加到数据库 | |
| Bills.newHallLoadingBill | 语法 | Public ResultMessage newHallLoadingBill(myDate date,String HallNum,String TravelNum,String TrafficNum,String EndPlace,String name,long[] GoodsNum); | |
| 前置条件 | 启动营业厅装车单生成 | |
| 后置条件 | 系统将生成的单据添加到数据库 | |
| Bills.newSendingBill | 语法 | Public ResultMessage newSendingBill(myDate date,long GoodsNum,String name,String StartPlace,String EndPlace,String Location,BOXSTYPE boxstype,SENDSTYPE sendstype,double price);s | |
| 前置条件 | 启动寄件单生成 | |
| 后置条件 | 系统将生成的单据添加到数据库 | |
| Bills.newPayingBill | 语法 | Public ResultMessage newPayingBill(myDate date,long Cash,String name,String PayNum,String[] Item); | |
| 前置条件 | 启动付款单生成 | |
| 后置条件 | 系统将生成的单据添加到数据库 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| BillsDataService.addBills(BillsPO po) | | | 在数据库中插入BillsPO对象 |
| BillsDataService.findBills(int BillNum) | | | 根据BillsPO对象编号查找对象 |
| BillsDataService.deleteBills(int BillNum) | | | 删除BillsPO对象 |
| BillsDataService.updateBills(int BillNum,BillsPO po) | | | 更新BillsPO对象 |

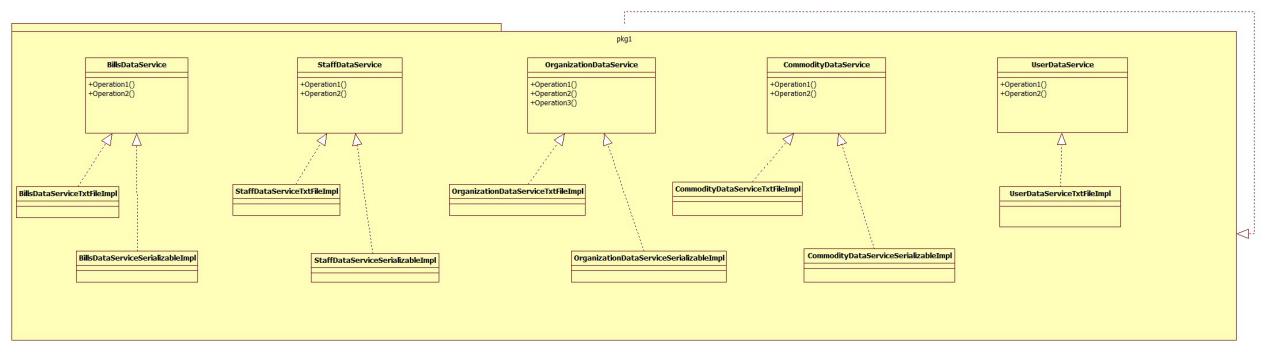
|  |  |  |  |
| --- | --- | --- | --- |
| Staffbl模块的接口规范 | | | |
| 提供的服务（供接口） | | | |
| Staff.newStaff | 语法 | | public ResultMessage newStaff(StaffVO vo) |
| 前置条件 | | 用户已输入新员工的各项信息，并点击了新增员工 |
| 后置条件 | | 在数据库中加入StaffPO对象 |
| Staff.deleteStaff | 语法 | | public ResultMessage deleteStaff(StaffVO vo) |
| 前置条件 | | 用户选择删除某员工 |
| 后置条件 | | 数据库中删除该员工 |
| Staff.updateStaff | 语法 | | Public ResultMessage updateStaff(StaffVO vo) |
| 前置条件 | | 用户输入了更新的数值并保存 |
| 后置条件 | | 数据库更新与之对应的StaffPO |
| Staff.getAllStaff | 语法 | | Public ArrayList<StaffVO> getAllStaff() |
| 前置条件 | | 无 |
| 后置条件 | | 把所有员工返回给界面 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| StaffDataService.add(StaffPO po) | | 新增员工 | |
| StaffDataService.remove(StaffPO po) | | 删除员工 | |
| StaffDataService.update(StaffPO po) | | 更改员工信息 | |
| StaffDataService.find(long id) | | 根据id查找某一员工 | |
| StaffDataService.getAll() | | 获取全部员工 | |

|  |  |  |
| --- | --- | --- |
| Organizationbl模块的接口规范 | | |
| 提供的服务（供接口） | | |
| Organizationbl.AddHallbl | 语法 | Public boolean AddHallbl(String cityName,int HallNum,double[] distanceToOtherCities); |
| 前置条件 | 启动新增营业厅 |
| 后置条件 | 数据层将营业厅添加到数据库中，返回操作结果 |
| Organizationbl.AddCenterbl | 语法 | Public boolean AddCenterbl(int city,int CenterNum,String name); |
| 前置条件 | 启动新增城市 |
| 后置条件 | 数据层将中转中心（城市）添加到数据库中，返回操作结果 |
| Organizationbl.DelHallbl | 语法 | Public boolean DelHallbl(int HallNum); |
| 前置条件 | 启动删除营业厅 |
| 后置条件 | 数据层将营业厅从数据库中删除，返回操作结果 |
| Organizationbl.DelCenterbl | 语法 | Public boolean DelCenterbl(int city); |
| 前置条件 | 启动删除城市 |
| 后置条件 | Organizationbl.DelHallbl也会在本接口之后被自动调用，数据层将中转中心（城市）从数据库中删除，返回操作结果 |
| Organizationbl.HallAddCarbl | 语法 | Public boolean HallAddCarbl(int carNum,int engineNum,int chassisNum,myDate buyTime,myDate serveTime,String number); |
| 前置条件 | 启动增加车辆信息 |
| 后置条件 | 数据层将车辆信息添加到数据库中，返回操作结果 |
| Organizationbl.HallAddDriverbl | 语法 | Public boolean HallAddDriverbl(int driverNum,String name,myDate birthTime,String ID,boolean gender,String phoneNum,int licensePeriod); |
| 前置条件 | 启动增加司机信息 |
| 后置条件 | 数据层将司机信息添加到数据库中，返回操作结果 |
| Organizationbl.HallChaCarbl | 语法 | Public CarPO HallChaCarbl(int carNum); |
| 前置条件 | 启动修改车辆信息 |
| 后置条件 | Organizationbl.HallAddCarbl和Organizationbl.HallDelCarbl也会在本接口之后被调用，数据层将车辆信息更新后添加到数据库中，返回操作结果 |
| Organizationbl.HallChaDriverbl | 语法 | Public DriverPO HallChaDriverbl(int driverNum); |
| 前置条件 | 启动修改司机信息 |
| 后置条件 | Organizationbl.HallAddDriverbl和Organizationbl.HallDelDriverbl也会在本接口之后被调用，数据层将司机信息更新后添加到数据库中，返回操作 |
| Organizationbl.HallDelCarbl | 语法 | Public boolean HallDelCarbl(int carNum); |
| 前置条件 | 启动删除车辆信息 |
| 后置条件 | 数据层将车辆信息从数据库中删除，返回操作结果 |
| Organizationbl.HallDelDiverbl | 语法 | Public boolean HallDelDriverbl(int driverNum); |
| 前置条件 | 启动删除司机信息 |
| 后置条件 | 数据层将司机信息从数据库中删除，返回操作结果 |
| Organizationbl.ManagerSetCitiesDistance | 语法 | Public boolean ManagerSetCitiesDistance(double[] distance); |
| 前置条件 | 启动设置距离 |
| 后置条件 | 数据层将将新的城市与其他城市的距离添加到数据库，返回操作结果 |
| Organizationbl.ManagerSetStdFee | 语法 | Public boolean ManagerSetStdFee(double stdfee); |
| 前置条件 | 启动设置标准运费 |
| 后置条件 | 数据层将将新的标准运费添加到数据库，返回操作结果 |
| Organizationbl.ManagerSetVehFee | 语法 | Public boolean ManagerSetVehFee(int veh,int fee); |
| 前置条件 | 启动设置载具费用 |
| 后置条件 | 数据层将新的载具费用添加到数据库，返回操作结果 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrganizationDataService.addHall(HallPO po) | 在数据库里插入HallPO对象 | |
| OrganizationDataService.delHall(int HallNum) | 在数据库里删除HallPO对象 | |
| OrganizationDataService.addCenter(CenterPO po) | 在数据库里插入CenterPO对象 | |
| OrganizationDataService.delCenter(int CenterNum) | 在数据库里删除CenterPO对象 | |
| OrganizationDataService.findCenter(int CenterNum) | 在数据库里查找CenterPO对象 | |
| OrganizationDataService.findHall(int HallNum) | 在数据库里查找HallPO对象 | |
| OrganizationDataService.addCar(CarPO po) | 在数据库里插入CarPO对象 | |
| OrganizationDataService.delCar(int CarNum) | 在数据库里删除CarPO对象 | |
| OrganizationDataService.findCar(int CarNum) | 在数据库里寻找CarPO对象 | |
| OrganizationDataService.addDriver(DriverPO po) | 在数据库里插入driverPO对象 | |
| OrganizationDataService.delDriver(int driverNum) | 在数据库里删除driverPO对象 | |
| OrganizationDataService.findDriver(int driverNum) | 在数据库里查找driverPO对象 | |

|  |  |  |
| --- | --- | --- |
| Managebl模块的接口规范 | | |
| 提供的服务（供接口） | | |
| Manage.Beginningaccount.new | 语法 | Public benningAccount newBenningAccount(String organization,int people,int car,int storage,String accountname,int balance); |
| 前置条件 | 财务人员输入机构、人员、车辆、库存、 银行账户信名称，余额。 |
| 后置条件 | 验证数据无错误后，记录当前账单 |
| Manage.Beginningaccount.initialize | 语法 | Public void accountInitilize(BeginaccountPO account); |
| 前置条件 | 期初账单被创建 |
| 后置条件 | 将账单中数据作为系统启动初始状态 |
| Manage.Beginningaccount.delete | 语法 | Public void accountDelete(BeginAccountPO account); |
| 前置条件 | 财务人员选择删除账单 |
| 后置条件 | 删除此账单，系统初始状态清零 |
| Manage.accountManagement.new | 语法 | Public BankAccountPO newaccount(String name); |
| 前置条件 | 财务人员输入有效的账户名称 |
| 后置条件 | 根据系统收款单和付款单计算该账户余额 |
| Manage.accountManagement.del | 语法 | Public void delAccount(BankAccountPO account); |
| 前置条件 | 财务人员选择删除账号 |
| 后置条件 | 系统删除账号 |
| Manage.accountManagement.change | 语法 | Public void changeAccount(BankAccountPO account,String newname); |
| 前置条件 | 财务人员输入新的账号名 |
| 后置条件 | 系统修改账号名 |
| Manage.accountManagement.search | 语法 | Public BankAccountPO[] accountSearch(String namepart); |
| 前置条件 | 财务人员输入关键字 |
| 后置条件 | 系统查找账户，若有，则记录，若没有，则提示没找到 |
| Manage.accountManagement.check | 语法 | Public boolean checkAccount(UserPO user) |
| 前置条件 | 财务人员选择账户管理 |
| 后置条件 | 若财务人员有最高权限，进入账户管理；若没有，则拒绝进入账户管理 |
| Manage.incometable.calculate | 语法 | Public void calculate(); |
| 前置条件 | 用户选择成本收益表 |
| 后置条件 | 无 |
| Manage.incometable.output | 语法 | Public void incometableOutput(); |
| 前置条件 | 用户选择导出成本收益表 |
| 后置条件 | 无 |
| Manage.statustable.search | 语法 | Public void statustableSearch(myDate date1,myDate date2); |
| 前置条件 | 用户选择开始日期和结束日期 |
| 后置条件 | 系统搜索所有符合条件的条目并列出 |
| Manage.statustable.output | 语法 | Public void statustableOutput(); |
| 前置条件 | 用户选择导出经营情况表 |
| 后置条件 | 无 |
| Manage.incomemanagement.select | 语法 | Public void incomeSelect(myDate date,int hallNumber) |
| 前置条件 | 用户输入日期,营业厅 |
| 后置条件 | 系统查找制定日期该营业厅的所有收款单 |
| Manage.incomemanagement.add | 语法 | Public void incomeAdd(incomebill bill) |
| 前置条件 | 营业厅收款 |
| 后置条件 | 无 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| ManageDataServive.beginningaccount.get | 获取期初账单 | |
| ManageDataService.beginningaccount.new | 存储新建期初账单数据，修改系统初始数据 | |
| ManageDataService. beginningaccount.delete | 从数据中删除账单数据，修改系统初始数据 | |
| ManageDataService.accountmanagement.add | 添加新的账户数据 | |
| ManageDataService. accountmanagement.delete | 删除账户数据 | |
| ManageDataService. accountmanagement.getall | 获取所有的账户 | |
| ManageDataService. accountmanagement.search | 搜索含有关键字的账户 | |
| ManageDataService.incomemanagement.select | 搜索所有符合条件的付款单 | |
| ManageDataService.Incometable.calculate | 计算收入支出收益 | |
| ManageDataService. Incometable.output | 输出成本收益表 | |
| ManageDataService. statustable.output | 输出经营情况表 | |
| ManageDataService. statustable.search | 搜索符合时间条件内的收款单 付款单 | |

## 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。由于持久化数据的保存可能存在多种形式，如txt文件、序列化文件、数据库，所示抽象了数据服务。数据层模块的描述具体如下图所示。



### 5.4.1数据层模块的职责

|  |  |
| --- | --- |
| 数据层模块的职责 | |
| 模块 | 职责 |
| BillsDataService | 持久化账单的数据库的接口，提供集体载入、保存以及增删改查服务 |
| BillsDataServiceTxtFileImpl | 基于txt文件的接口 |
| BillsDataServiceSerializableImpl | 基于序列化文件的接口 |
| StaffDataService | 持久化员工的数据库的接口，提供集体载入、保存以及增删改查服务 |
| StaffDataServiceTxtFileImpl | 基于txt文件的接口 |
| StaffDataServiceSerializableImpl | 基于序列化文件的接口 |
| OrganizationDataService | 持久化机构的数据库的接口，提供集体载入、保存以及增删改查服务 |
| OrganizationDataServiceTxtFileImpl | 基于txt文件的接口 |
| OrganizationDataServiceSerializableImpl | 基于序列化文件的接口 |
| CommodityDataService | 持久化库存的数据库的接口，提供集体载入、保存以及增删改查服务 |
| CommodityDataServiceTxtFileImpl | 基于txt文件的接口 |
| CommodityDataServiceSerializableImpl | 基于序列化文件的接口 |
| UserDataService | 持久化登录信息的数据库的接口，提供集体载入、保存以及增删改查服务 |
| UserDataServiceTxtFileImpl | 基于txt文件的接口 |

### 5.4.2数据层模块的接口规范

|  |  |  |
| --- | --- | --- |
| 数据层模块的接口规范 | | |
| 提供的服务（供接口） | | |
| BillsDataService.addBills | 语法 | Public ResultMessage addBills(BillsPO po) |
| 前置条件 | 同样单据编号的单据在BillsData中不存在 |
| 后置条件 | 在BillsData中添加一个PO |
| BillsDataService.findBills | 语法 | Public ResultMessage findBills(int BillNum) |
| 前置条件 | 无 |
| 后置条件 | 返回找到的PO |
| BillsDataService.deleteBills | 语法 | Public ResultMessage deleteBills(int BillNum) |
| 前置条件 | 在BillsData中存在该编号 |
| 后置条件 | 从BillsData中删除PO |
| BillsDataService.updateBills | 语法 | Public ResultMessage updateBills(int BillNum,BillsPO po) |
| 前置条件 | 在BillsData中存在该编号 |
| 后置条件 | 删除掉原先的PO，添加修改后的PO |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StaffDataService.add | 语法 | Public ResultMessage add(StaffPO po) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库新增 |
| StaffDataService.remove | 语法 | Public ResultMessage remove(StaffPO po) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库删除此PO |
| StaffDataService.update | 语法 | Public ResultMessage update(StaffPO po) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库更新此PO |
| StaffDataService.find | 语法 | Public StaffPO find(long id) |
| 前置条件 | BL层调用 |
| 后置条件 | 返回此PO给BL层 |
| StaffDataService.getAll | 语法 | Public ArrayList<StaffPO> getAll() |
| 前置条件 | BL层调用 |
| 后置条件 | 返回此ArrayList给BL层 |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrganizationDataService.addHall | 语法 | Public ResultMessage addHall(HallPO po) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库新增 |
| OrganizationDataService.delHall | 语法 | Public ResultMessage delHall(HallPO po) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库删除此PO |
| OrganizationDataService.addCenter | 语法 | Public ResultMessage addCenter(CenterPO po) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库新增 |
| OrganizationDataService.delCenter | 语法 | Public ResultMessage delCenter(int CenterNum) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库删除此PO |
| OrganizationDataService.findHall | 语法 | Public HallPO findHall(int HallNum) |
| 前置条件 | BL层调用 |
| 后置条件 | 返回此PO |
| OrganizationDataService.findCenter | 语法 | Public CenterPO findCenter(int CenterNum) |
| 前置条件 | BL层调用 |
| 后置条件 | 返回此PO |
| OrganizationDataService.addCar | 语法 | Public ResultMessage addCar(CarPO po) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库新增 |
| OrganizationDataService.addDriver | 语法 | Public ResultMessage addDriver(DriverPO po) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库新增 |
| OrganizationDataService.delCar | 语法 | Public ResultMessage delCar(CarPO po) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库删除此PO |
| OrganizationDataService.delDriver | 语法 | Public ResultMessage delDriver(DriverPO po) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库删除此PO |
| OrganizationDataService.findHall | 语法 | Public CarPO findHall(int CarNum) |
| 前置条件 | BL层调用 |
| 后置条件 | 返回此PO |
| OrganizationDataService.findHall | 语法 | Public DriverPO findHall(int driverNum) |
| 前置条件 | BL层调用 |
| 后置条件 | 返回此PO |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ManageDataServive.beginningaccount.get | 语法 | Public BeginAccountPO getBeginningDS(); |
| 前置条件 | 财务人员打开期初建账，已有期初账单 |
| 后置条件 | 显示期初账单 |
| ManageDataService.beginningaccount.new | 语法 | Public void newBeginAccountDS(BeginaccountPO account); |
| 前置条件 | 期初账单新建 |
| 后置条件 | 初始化期初数据 |
| ManageDataService. beginningaccount.delete | 语法 | Public void delBeginAccountDS(BeginAccountPO account); |
| 前置条件 | 期初账单被从列表中删除 |
| 后置条件 | 初始化期初数据 |
| ManageDataService.accountmanagement.add | 语法 | Public void addBankAccountDS(String name); |
| 前置条件 | 新建账号 |
| 后置条件 | 无 |
| ManageDataService. accountmanagement.delete | 语法 | Public void deleteBankAccountDS(String name); |
| 前置条件 | 账号被从列表中删除 |
| 后置条件 | 无 |
| ManageDataService. accountmanagement.getall | 语法 | Public void getAllBankAccountDS(); |
| 前置条件 | 有账户数据，用户打开账户管理或者搜索账户管理 |
| 后置条件 | 无 |
| ManageDataService. accountmanagement.search | 语法 | Public BankAccountPO[] searchBankAccountDS (String name); |
| 前置条件 | 用户输入查找关键字 |
| 后置条件 | 将符合条件的结果返回逻辑层 |
| ManageDataService.incomemanagement.select | 语法 | Public incomecollectionbill[] incomeSelectDS(myDate date,int hallnumber); |
| 前置条件 | 用户输入时间，营业厅 |
| 后置条件 | 将符合条件的结果返回逻辑层 |
| ManageDataService.Incometable.calculate | 语法 | Public IncomeTableData calculateDS(); |
| 前置条件 | 用户选择成本收益表 |
| 后置条件 | 将数据返回逻辑层 |
| ManageDataService. Incometable.output | 语法 | Public void incomeTableOutputDS(IncomeTableData data); |
| 前置条件 | 用户选择导出成本收益表 |
| 后置条件 | 系统输出成本收益表 |
| ManageDataService. statustable.output | 语法 | Public void statusTableOutputDS(); |
| 前置条件 | 用户选择导出经营情况 |
| 后置条件 | 系统输出经营情况表 |
| ManageDataService. statustable.search | 语法 | Public void statustableSearch(myDate date1,myDate date2); |
| 前置条件 | 用户选择开始日期和结束日期 |
| 后置条件 | 系统搜索所有符合条件的条目并列出 |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CommodityDataService.addCom | 语法 | Public ResultMessage add(ComPO po) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库新增 |
| CommodityDataService.delCom | 语法 | Public ResultMessage del(ComPO po) |
| 前置条件 | BL层调用 |
| 后置条件 | 数据库中删除此PO |
| CommodityDataService.findCom | 语法 | Public ComPO find(long GoodsNum) |
| 前置条件 | BL层调用 |
| 后置条件 | 返回此PO |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UserDataService.findUser | 语法 | Public ResultMessage insert(UserPO po) |
| 前置条件 | 启动登录 |
| 后置条件 | 返回登录消息 |

# 信息视角

## 6.1数据持久化对象

系统的PO类就是对应相关的实体类，在此只做简单的介绍。

UserPO类包含用户的用户名、密码属性。

CommodityPO类包含库存托运货物的编号、入库日期、目的地、区号、排号、架号、位号属性。

BillsPO类包含各种单据的单号、单据生成日期以及其他特殊属性。

StaffPO类包括职员的个人信息、工资、岗位。

## 6.2Txt持久化格式

Txt数据保存格式以User.txt为例。每行分别对应用户名、密码，如下所示：

141250001 141250001

161250001 woshiniyeye

171250001 hiahiahiahia

## 6.3数据库表

## 数据库中包含User表、Commodity表、Bills