快递物流系统结构描述文档

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 |
| ELS | 快递物流系统 | …… |
| …… | …… | …… |

1.3 参考资料

2.产品概述

随着现公司规模扩大，企业业务量、办公场所、员工数都发生增长，为适应新的环境，提高工作效率和用户满意度，该产品被设计出为使用者提供订单查询、信息管理、库存管理等多种功能。

3.逻辑视角

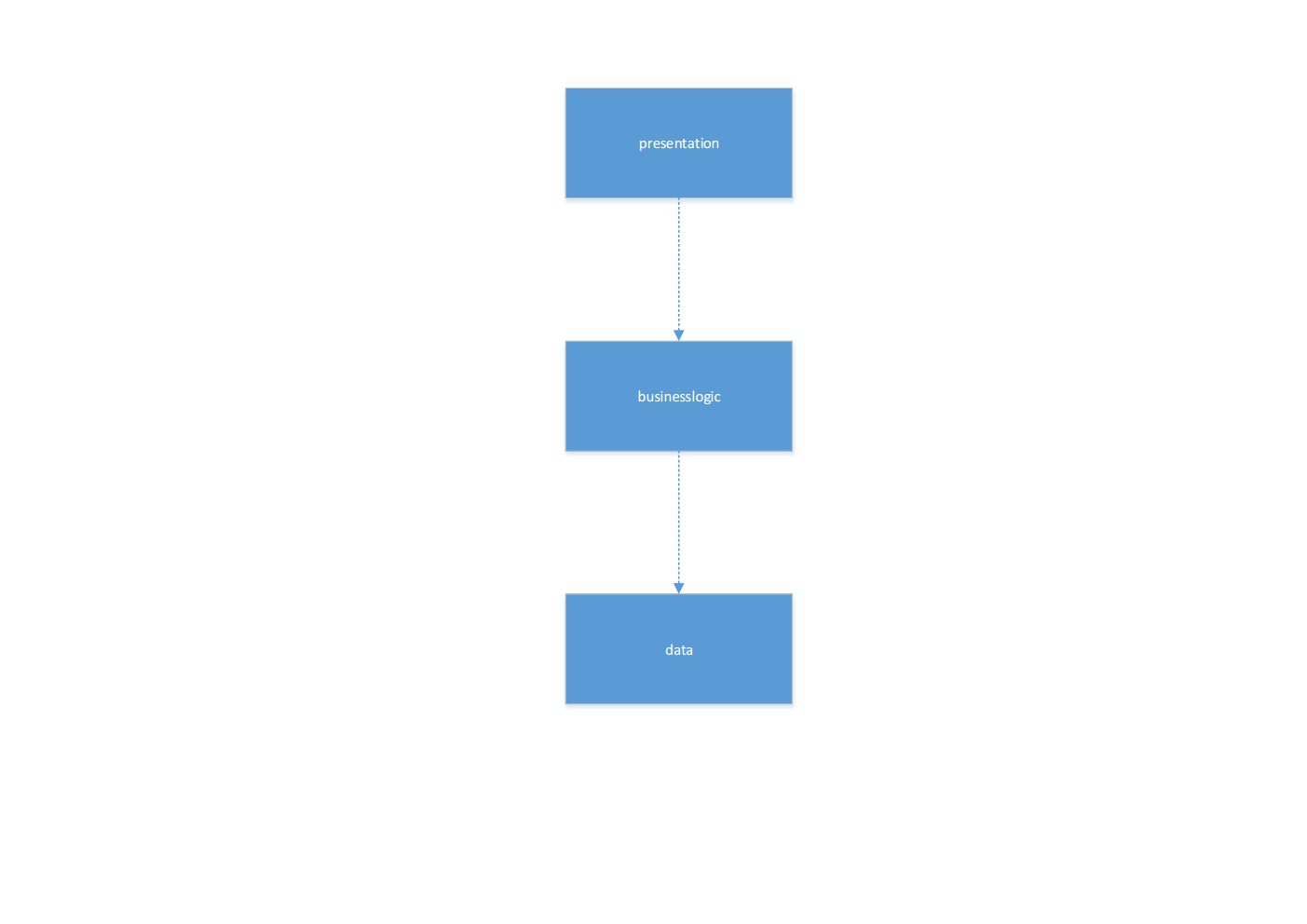
快递物流系统中，选择了分层体系结构，将系统分为三层（展示层、业务逻辑层、数据层）能够很好地展示整个高层抽象。展示层包含GUI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如图1，2所示。

图1 参照体系结构风格的包图表达逻辑视角

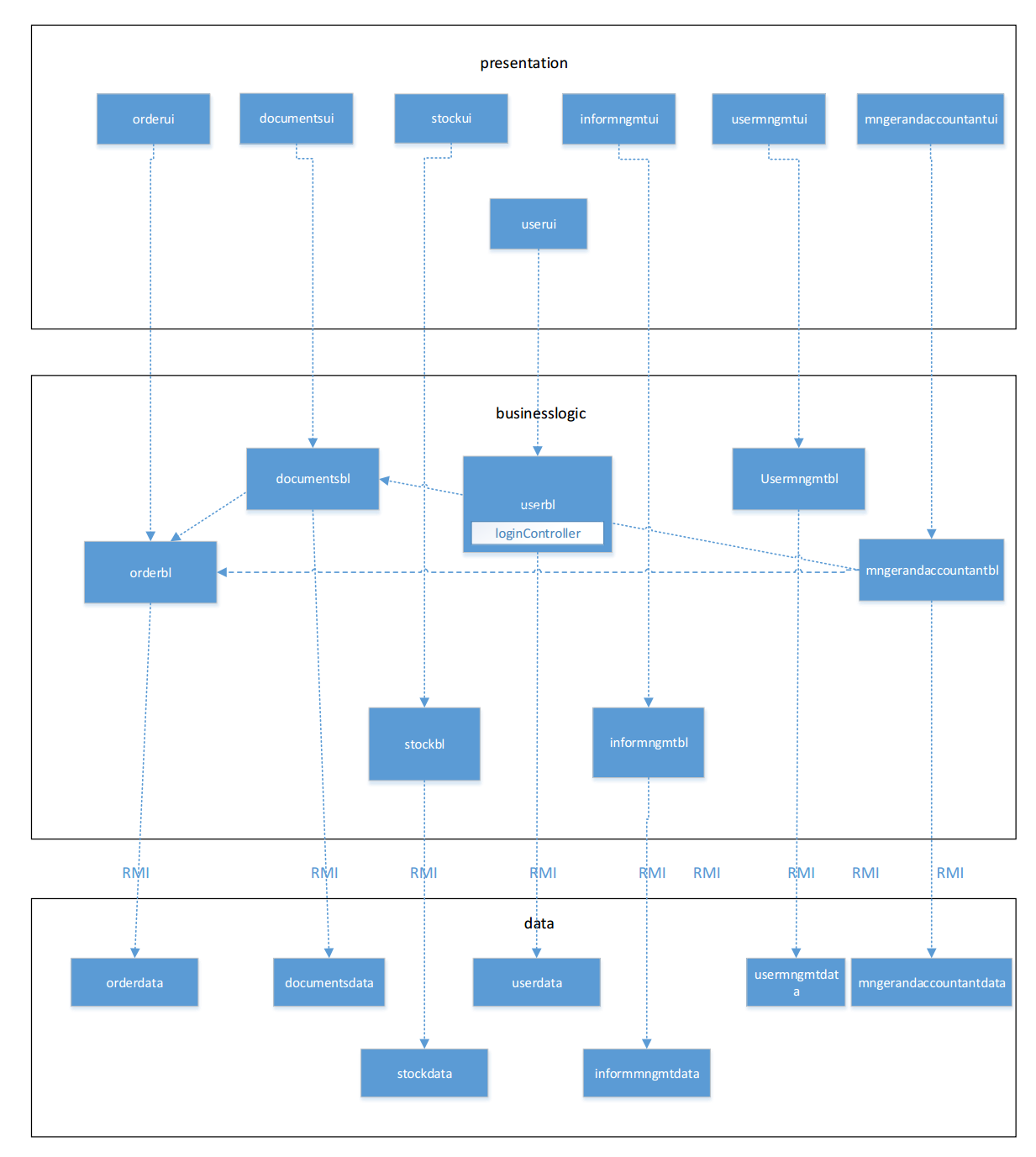


图2 软件体系结构逻辑设计方案

4.组合视角

4.1 开发图包

快递物流系统的最终开发包设计如表1所示。

表1 快递物流系统的最终开发包设计

|  |  |
| --- | --- |
| 开发（物理）包 | 依赖的其他开发包 |
| mainui | orderui, documentsui, stockui, informngmtui, mngerandaccoutantui, usermngmtui,userui |
| orderui | orderblservice, 界面类库包,vo |
| orderblservice |  |
| orderbl | orderblservice ,orderdataservice,po |
| orderdataservice | Java RMI,po |
| orderdata | databaseutility ,po , orderdataservice |
| documentsui | documentsblservice,界面类库包,vo |
| documentsblservice |  |
| documentsbl | documentsblservice, documentsdataservice , po ,orderbl |
| documentsdataservice | Java RMI,po |
| documentsdata | databaseutility ,po, documentsdataservice |
| stockui | stockblservice,界面类库包,vo |
| stockblservice |  |
| stockbl | stockblservice ,stockdataservice,po |
| stockdataservice | Java RMI,po |
| stockdata | databaseutility ,po, stockdataservice |
| informngmtui | informngmtblservice,界面类库包,vo |
| informngmtblservice |  |
| informngmtbl | informngmtblservice, informngmtdataservice,po |
| informngmtdataservice | Java RMI,po |
| informngmtdata | databaseutility ,po, informngmtdataservice |
| usermngmtui | usermngmtblservice,界面类库包,vo |
| usermngmtblservice |  |
| usermngmtbl | usermngmtblservice, usermngmtdataservice,po |
| usermngmtdataservice | Java RMI,po |
| usermngmtdata | databaseutility ,po, usermngmtdataservice |
| userui | userblservice,界面类库包,vo |
| userblservice |  |
| userbl | userblservice, userdataservice,po, |
| userdataservice | Java RMI,po |
| userdata | databaseutility ,po, userdataservice |
| mngerandaccoutantui | mngerandaccoutantblservice，界面类库包,vo |
| mngerandaccoutantblservice |  |
| mngerandaccoutantbl | mngerandaccoutantblservice, mngerandaccoutantdataservice,po, documentsbl, orderbl |
| mngerandaccoutantdataservice | Java RMI,po |
| mngerandaccoutantdata | databaseutility ,po, mngerandaccoutantdataservice |
| po |  |
| utibilitybl |  |
| 界面类库包 |  |
| Java RMI |  |
| databaseutility | JDBC |

快递物流系统客户端开发包图如图3所示，服务端开发包图如图4所示。

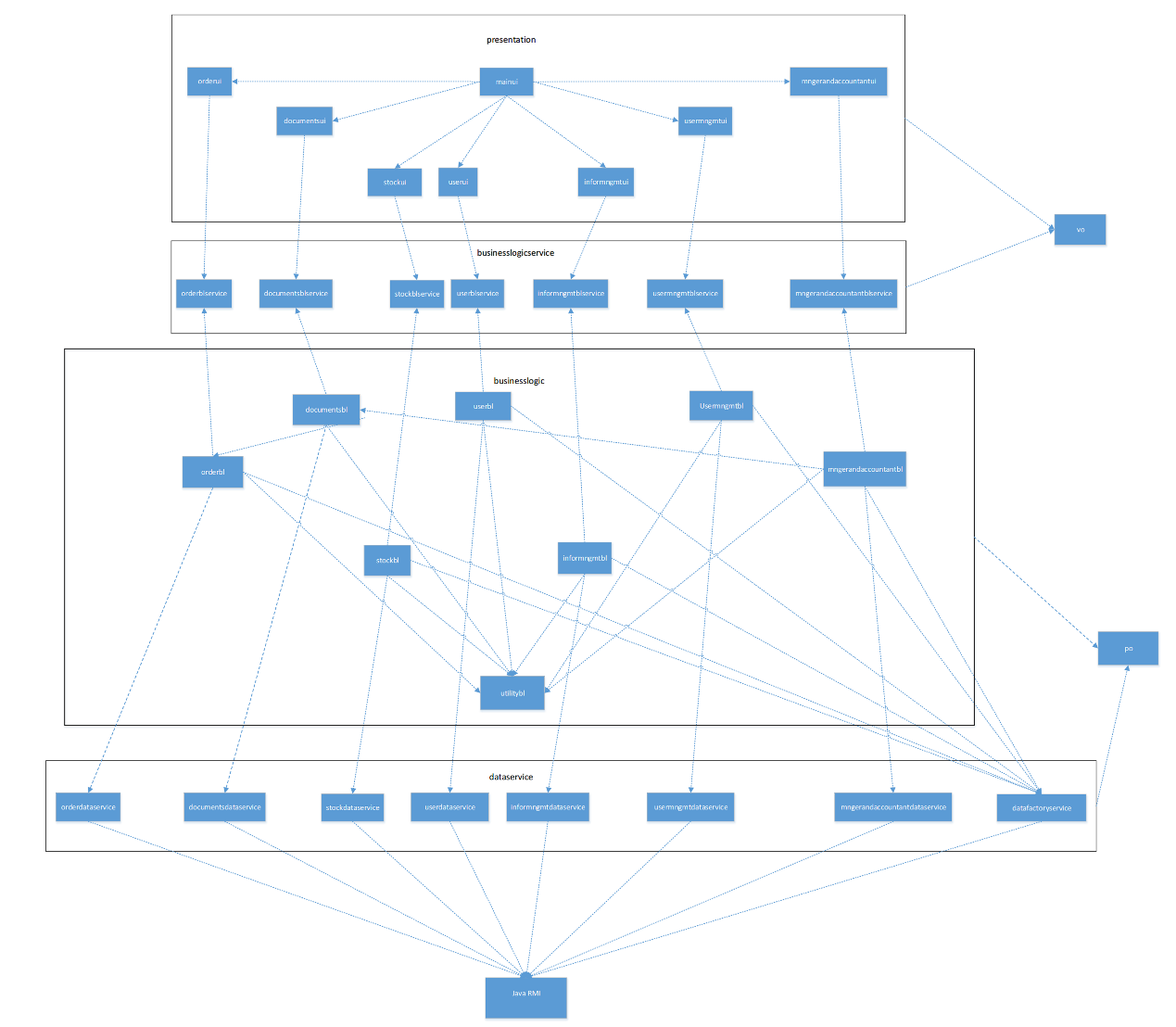


图3 客户端开发包图

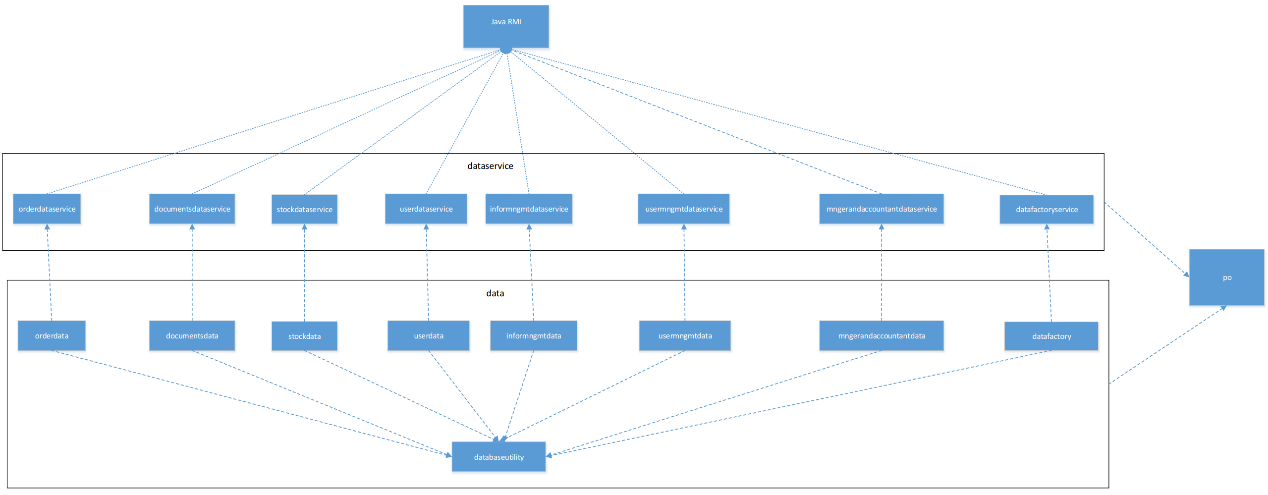


图4 服务器端开发包图

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，其进程图如附图所示。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。

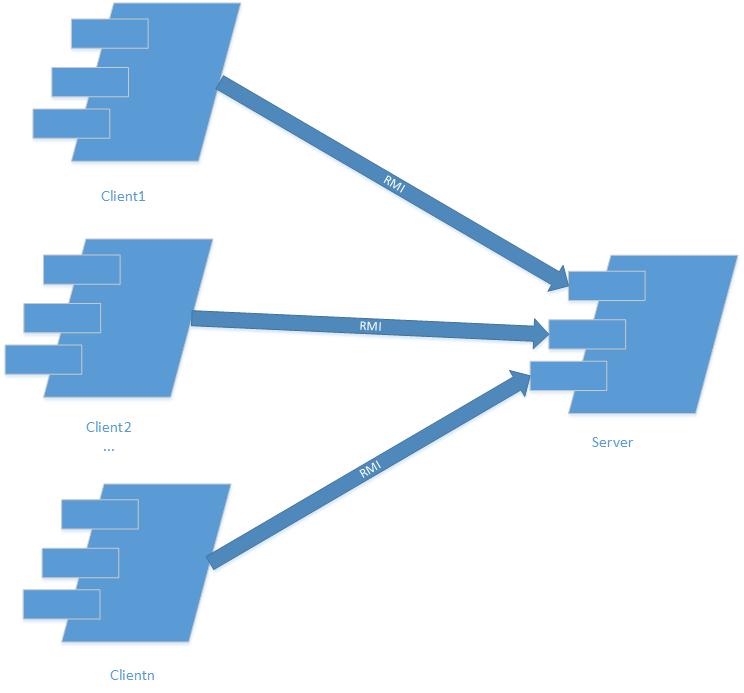


Figure 5 进程图

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。由于Java RMI构件属于JDK 6.0的一部分。所以，在系统JDK环境已经设置好的情况下，不需要再独立部署。部署图如附图所示。

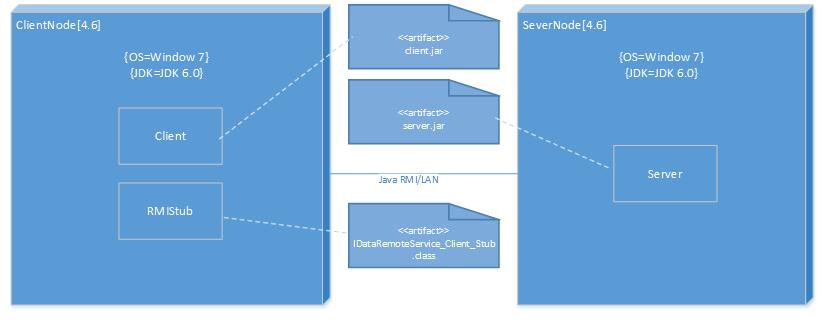
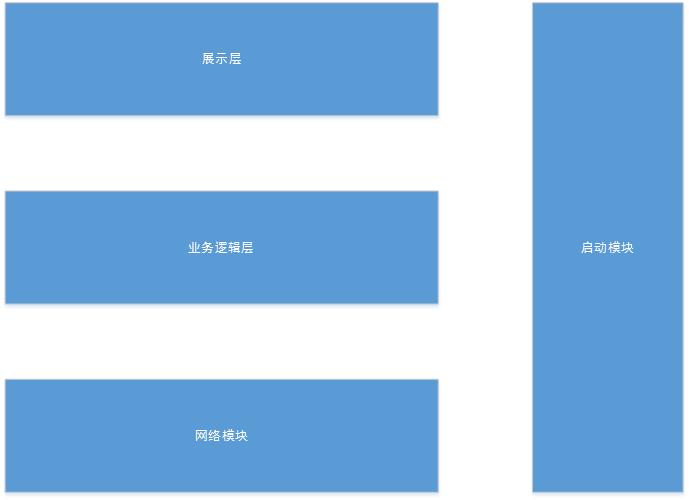


Figure 6 部署图

5接口视角

5.1模块的职责

客户端模块和服务器端模块视图分别如附图所示。客户端各层和服务器端各层的职责分别如表2和表3所示。

 图7 客户端模块视图

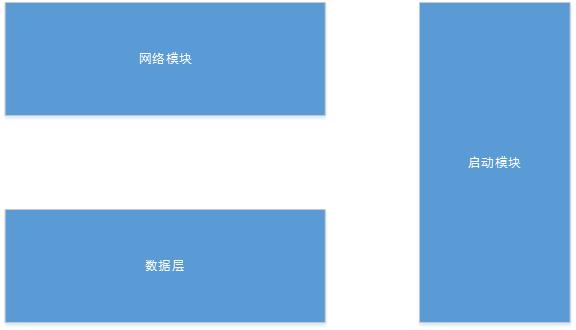
 图8 服务器端模块视图

表2 客户端各层的职责

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化网络通信机制，启动用户界面 |
| 用户界面层 | 基于窗口的快递物流客户端用户界面 |
| 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4所示。

表4 层之间调用的接口

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| documentsblservice  informngmtblservice  usermngmtblservice  mngerandaccountantblservice  orderblservice  stockblservice  userblservice | 客户端展示层 | 客户端业务逻辑层 |
| datafactoryservice  documentsdataservice  informngmtdataservice  mngerandaccountantdataservice  orderdataservice  stockdataservice  userdataservice  usermngmtdataservice | 客户端业务逻辑层 | 服务器端数据层 |

借用单据用例来说明层之间的调用，如附图所示。每一层之间都是由上层依赖了一个接口（需接口），而下层实现这个接口（供接口）。documentsblservice提供了docunments界面所需要的所有业务逻辑功能。Documentsdataservice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。

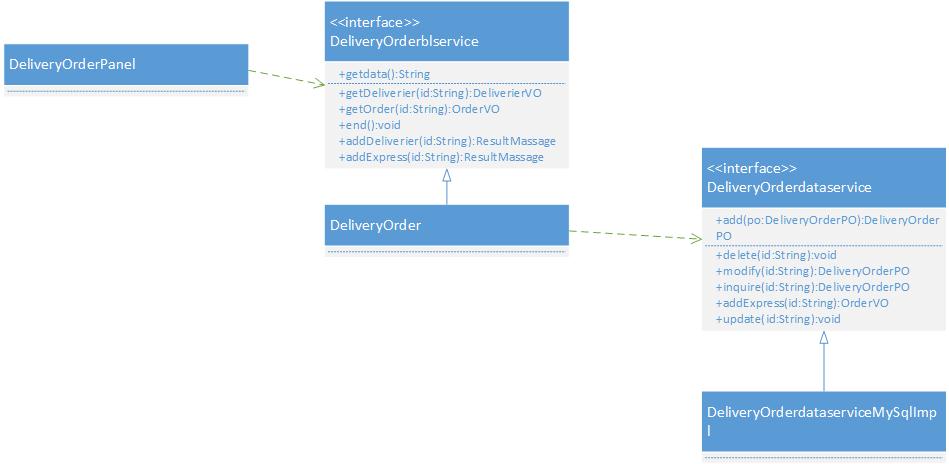


Figure 9 派件单用例层之间调用的接口

5.2 用户界面层的分解

根据需求，系统存在58个界面：登陆界面、营业厅业务员主界面、派件单编辑界面、派件单列表界面、派件单界面、司机信息编辑界面、司机列表界面、司机信息界面、装车单编辑界面、装车单列表界面、装车单界面、到达单编辑界面、到达单列表界面、到达单界面、车辆信息编辑界面、车辆列表界面、车辆信息界面、财务人员主界面、银行账户列表界面、付款单编辑界面、付款单列表界面、付款单界面、收款单编辑界面、收款单列表界面、收款单界面、操作日志界面、成本收益表界面、经营情况表界面、总经理主界面、单据列表界面、单据界面、单据编辑界面、部门下人员列表界面、人员跨部门调动界面、人员信息界面、管理员主界面、用户管理界面、快递员主界面、订单编辑界面、订单界面、订单列表界面、中转中心业务员主界面、中转单编辑界面、中转单列表界面、中转单界面、到达单编辑界面、到达单列表界面、到达单界面、中转中心库存管理人员主界面、入库单编辑界面、入库单界面、出库单编辑界面、出库单界面、调整分区界面、库存快照界面、库存查看界面、寄件人主界面、物流信息查询界面。界面跳转如图所示。

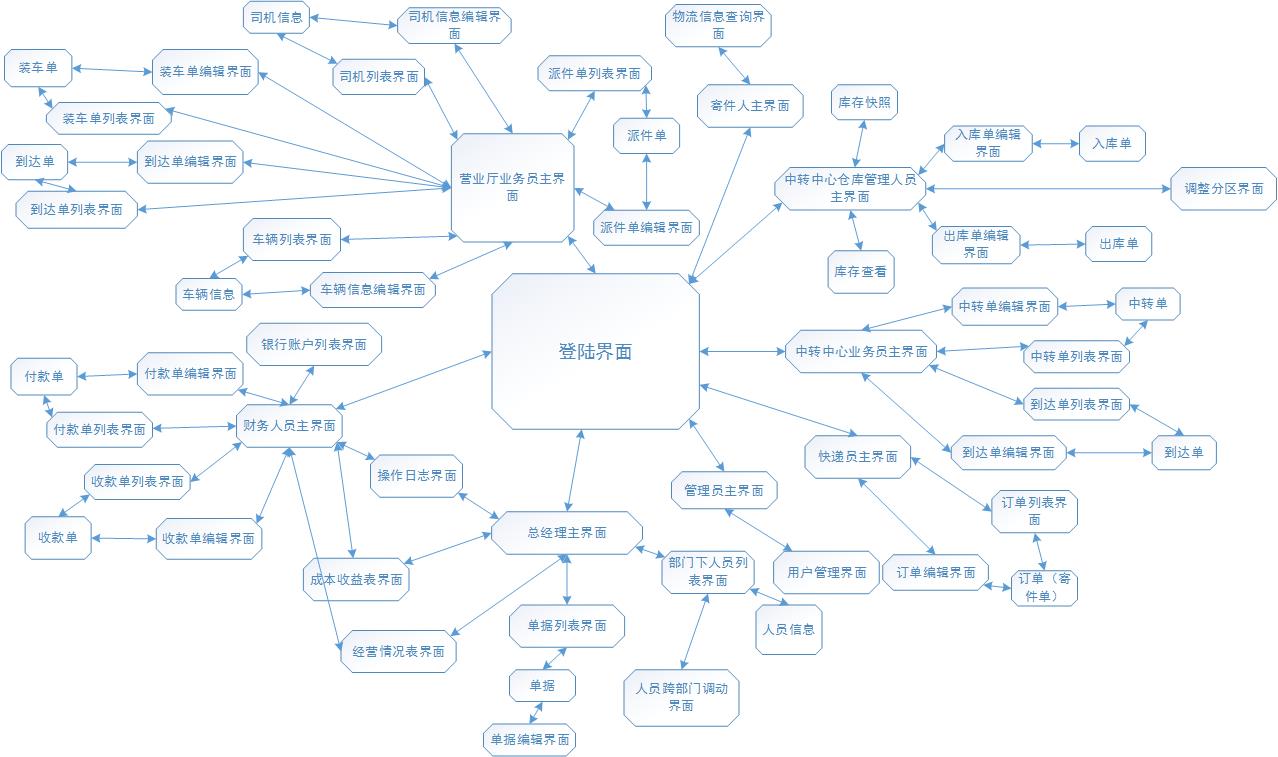


Figure 0 用户界面跳转

pkg

Figure 11 用户界面类

informngmtui

mngerandaccountantui

mainui

main（）

Stockui

documentsui

userui

orderui

usermngmtui

用户界面类如图11所示。

5.2.1 用户界面层模块的职责。

|  |  |
| --- | --- |
| 模块 | 职责 |
| mainui | 整个系统的入口，显示Frame |
| mngerandaccountantui | 负责总经理和财务人员处理某些业务的界面的显示 |
| orderui | 负责订单界面的显示 |
| documentsui | 负责除订单外其他单据界面的显示 |
| informngmtui | 负责信息（车辆、司机等）管理界面的显示 |
| stockui | 负责库存管理界面的显示 |
| userui | 负责登陆界面以及系统用户主界面的显示 |
| usermngmtui | 负责管理员用户管理界面的显示 |
| listui | 负责各种列表界面的显示 |

5.2.2 用户界面层模块的接口规范（取部分作为样例）

用户界面层模块的职责

|  |  |  |
| --- | --- | --- |
| mainui | 语法 | main() |
| 前置 | 无 |
| 后置 | 显示Frame |

用户界面层需要的服务接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| businesslogicservice.userblservice | 登陆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**

业务逻辑层包括针对多个界面的业务逻辑处理对象。例如，User对象负责处理登陆界面和各个用户的主界面的业务逻辑；Order对象负责订单界面的业务逻辑。业务逻辑层的设计如图12所示。

Pkg

orderbl

mngerandaccountantbl

informngmtbl

documentsbl

utilitybl

usermngmtbl

userbl

stockbl

Figure 12 业务逻辑层的设计

5.3.1 业务逻辑层模块的职责

表格 8 业务逻辑层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| documentsbl | 负责实现各类单据所需要的服务 |
| Informngmtbl | 负责实现信息管理所需要的服务 |
| mngerandaccountantbl | 负责实现总经理和财务人员在部分业务中所需要的服务 |
| orderbl | 负责实现与订单有关的服务 |
| stockbl | 负责实现库存管理需要的服务 |
| userbl | 负责实现对应于登陆界面和系统用户主界面所需要的服务 |
| usermngmtbl | 负责实现用户管理所需要的服务 |
| utilitybl | 负责实现初始化和业务逻辑层上下文工作所需要的服务 |
| listbl | 负责实现各类列表所需要的服务 |

5.3.2 业务逻辑层模块的接口规范（\*只列出部分来说明，未列出的可以在原型代码的blservice查看）

Table 9 OutBoundOrderbl模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OutBoundOrder.addExpress | 语法 | public OrderlineitemVO addExpress(String id) |
| 前置条件 | 启动一个新建或修改出库单回合 |
| 后置条件 | 在一个新建或修改出库单回合中，增加出库快递编号 |
| OutBoundOrder.generateId | 语法 | public void generateId() |
| 前置条件 | 已添加所有出库单信息 |
| 后置条件 | 自动计算出该出库单的id |
| OutBoundOrder.generateTime | 语法 | public void generateTime() |
| 前置条件 | 已输入除了出库时间信息以外的所有出库单信息 |
| 后置条件 | 自动计算出生成出库单的日期 |
| OutBoundOrder.add | 语法 | public OutBoundOrderVO add(String id4e, String id4to, String desti) |
| 前置条件 | 库存管理人员已经登陆系统 |
| 后置条件 | 新建一个出库单，更新数据持久化对象 |
| OutBoundOrder.deleteone | 语法 | Public ResultMessage deleteone(String id4obo) |
| 前置条件 | 文件中已经有相关出库单的记录 |
| 后置条件 | 从文件中删除该出库单 |
| OutBoundOrder.deleteMany | 语法 | Public ResultMessage deleteMany(ArrayList<String> id4obo) |
| 前置条件 | 要删除的出库单在持久化数据里有记录 |
| 后置条件 | 删除多个出库单 |
| OutBoundOrder.modify | 语法 | public OutBoundOrderVO modify(String id, String id4e, String id4to, String desti) |
| 前置条件 | 要修改的出库单在持久化数据中有记录 |
| 后置条件 | 修改一个出库单，更新相关信息到VO和PO，把VO信息返回到展示层 |
| OutBoundOrder.inquireA | 语法 | public OutBoundOrderVO inquireA(String id) |
| 前置条件 | 要查询的出库单在持久化数据中有记录 |
| 后置条件 | 根据id查询某出库单,返回出库单VO信息 |
| OutBoundOrder.inquireB | 语法 | public OutBoundOrderVO inquireB(String time) |
| 前置条件 | 库存管理人员已经被授权和登录 |
| 后置条件 | 根据时间查询某出库单,返回出库单VO信息 |
| OutBoundOrder.endOBO | 语法 | public void endOBO() |
| 前置条件 | 业务已经处理完成 |
| 后置条件 | 结束此次与出库单相关的业务，持久化更新涉及的领域对象的数据 |
| 需要的服务（需接口） | | |
| 服务名 | | 服务 |
| OutBoundOrderdataservice. add(String id4obo, String time, String id4e, String id4to, String desti) | | 增加一个出库单，更新变动到PO |
| OutBoundOrderdataservice. deleteone(String id4obo) | | 删除一个出库单,更新PO数据 |
| OutBoundOrderdataservice. deleteMany(ArrayList<String> id4obo) | | 删除多个出库单,更新PO记录 |
| OutBoundOrderdataservice. modify(String id, String id4e, String id4to, String desti) | | 修改一个出库单，更新相关信息到PO |
| OutBoundOrderdataservice. inquireA(String id) | | 根据id查询某出库单,返回出库单PO信息给业务逻辑层 |
| OutBoundOrderdataservice. inquireB(String time) | | 根据时间查询某出库单,返回出库单PO信息 |
| OutBoundOrderdataservice.update() | | 持久化更新涉及的领域对象的数据 |

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。OutBoundOrder业务逻辑需要的服务由OutBoundDataService接口提供。我们的项目用序列化文件保存持久化数据，所示抽象了数据服务。数据层模块的描述具体如图13所示。

OutBoundOrderdata

<<inteface>>

OutBoundOrderdataservice

Add modify

Deleteone inqurieA

deleteMany inquireB

Pkg

Figure 13 数据层模块的描述

5.4.1 数据层模块的职责

数据层模块的职责如表11所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| OutBoundDataService | 持久化数据的接口，提供集体载入、集体保存、增删改查服务 |
| OutBoundData | 基于序列化文件的持久化数据库的接口，提供集体载入、集体保存、增删改查服务 |

5.4.2 数据层模块的接口规范（\*只列出部分来说明，未列出的可以在原型代码的dataservice层查看）

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OutBoundOrderdataservice.add | 语法 | public OutBoundOrderPO add(String id4obo, String time, String id4e, String id4to, String desti) |
| 前置条件 | 库存管理人员已经输入了所有出库单参数值 |
| 后置条件 | 增加一个出库单，更新变动到PO |
| OutBoundOrderdataservice.deleteone | 语法 | public ResultMessage deleteone(String id4obo) |
| 前置条件 | 要删除的单据在PO中有记录 |
| 后置条件 | 删除一个出库单,更新PO数据 |
| OutBoundOrderdataservice.deleteMany | 语法 | public ResultMessage deleteMany(ArrayList<String> id4obo) |
| 前置条件 | 要删除的单据在PO中有记录 |
| 后置条件 | 删除多个出库单,更新PO记录 |
| OutBoundOrderdataservice.modify | 语法 | public OutBoundOrderPO modify(String id, String id4e, String id4to, String desti) |
| 前置条件 | 要修改的出库单在PO中有记录 |
| 后置条件 | 修改一个出库单，更新相关信息到PO |
| OutBoundOrderdataservice.inquireA | 语法 | public OutBoundOrderPO inquireA(String id) |
| 前置条件 | 要查询的单据在PO中有记录 |
| 后置条件 | 根据id查询某出库单,返回出库单PO信息给业务逻辑层 |
| OutBoundOrderdataservice.inquireB | 语法 | public OutBoundOrderlineitemPO inquireB(String time) |
| 前置条件 | 要查询的单据在PO中有记录 |
| 后置条件 | 根据时间查询某出库单,返回出库单PO信息 |
| OutBoundOrderdataservice.update | 语法 | public void update() |
| 前置条件 | 行为或操作已经完成 |
| 后置条件 | 持久化更新涉及的领域对象的数据 |

6信息视角

6.1数据持久化对象

系统的PO类就是对应的相关的实体类，在此只做简单介绍。

* ArrivalOrder4BusiHallPO包含编号、到达日期、中转单编号、出发地、货物到达状态（损坏、完整、丢失）
* ArrivalOrder4TranCenPO包含编号、中转中心编号、到达日期、中转单编号、出发地、货物到达状态（损坏、完整、丢失）
* DeliveryOrderPO包含编号、到达日期、托运订单条形码号、派送员
* BankAccountPO包含名称、金额
* TransferOrderPO包含编号、交通方式、装车日期、本中转中心中转单编号、航班号、出发地、到达地、货柜号、监装员、本次装箱所有托运单号、运费
* ReceivablesOrderPO包含编号、收款日期、收款金额、收款快递员、对应的所有快递订单条形码号
* PaymentOrderPO包含编号、付款日期、付款金额、付款人、付款账号、条目、备注
* LoadingListPO包含编号、装车日期、本营业厅编号、汽运编号、到达地（、车辆代号、监装员、押运员、本次装箱所有订单条形码号）、运费
* VehiclesPO包含车辆代号、车牌号、服役时间、车辆图片
* StaffAndAgencyPO包含人员机构的具体信息
* DriversPO包含司机编号、姓名、出生日期、身份证号、手机、车辆单位、性别、行驶证期限
* StockPO包含快递编号、入库日期、目的地、区号、排号、架号、位号
* OutBoundOrderPO包含快递编号、出库日期、目的地、装运形式、中转单编号或者汽运编号
* AccountantPO包含编号、姓名、出生日期、身份证号、手机、性别
* AdministratorPO包含编号、姓名、出生日期、身份证号、手机、性别
* BusiHallClerkPO包含编号、姓名、出生日期、身份证号、手机、性别
* CourierPO包含编号、姓名、出生日期、身份证号、手机、性别
* GeneralManagerPO包含编号、姓名、出生日期、身份证号、手机、性别
* SenderPO包含编号、姓名、出生日期、身份证号、手机、性别、住址、单位
* StockManagerPO包含编号、姓名、出生日期、身份证号、手机、性别
* TranCenClerkPO包含编号、姓名、出生日期、身份证号、手机、性别
* OrderPO包含订单条形码号、寄件人姓名、住址、单位、电话、手机、收件人姓名、住址、单位、电话、手机、托运货物原件数、实际重量、体积、内件品名、经济快递/标准快递/特快专递、包装费、总费用。
* OperaLogPO包含操作日期、操作人员、操作类型、操作编号

持久化用户对象CourierPO的定义如下所示。

public class CourierPO implements Serializable{

String id;

String name;

String password;

String birthday;

String idcard;

String mobilephone;

String sex;

String sex; public CourierPO (String I, String n, String p, String b, String I, String m, String s){

id=i;

name=n;

password=p;

birthday=b;

idcard=i;

mobilephone=m;

sex=s;

}

public String getName(){

return name;

}

public String getId(){

return id;

}

public String getPassword (){

return password;

}

public String getBirthday (){

return birthday;

}

public String getIdcard(){

return idcard;

}

public String getMobilephone (){

return mobilephone;

}

public String getSex (){

return sex;

}

}

6.2Txt持久化格式

Txt数据保持格式以Order.Txt为例。每行分别对应订单条形码号、寄件人姓名、住址、单位、电话、手机、收件人姓名、住址、单位、电话、手机、托运货物原件数、实际重量、体积、内件品名、经济快递/标准快递/特快专递、包装费、总费用。中间以：隔开。如下所示：

0000000001：张三：\*\*小区：\*\*公司：86\*\*\*：189\*\*\*：李四：\*\*小区：\*\*公司：87\*\*\*：182\*\*\*：3：1kg：2dm^3:李子：经济快递：1：7

0101010111：李四：\*\*小区：\*\*公司：87\*\*\*：182\*\*\*：张三：\*\*小区：\*\*公司：86\*\*\*：189\*\*\*：1：2kg：1m^3:被子：经济快递：5：20

6.3数据库表

数据库中包含ArrivalOrder4BusiHall表、ArrivalOrder4TranCen表、DeliveryOrder表、BankAccount表、TransferOrder表、ReceivablesOrder表、PaymentOrder表、LoadingList表、VehiclesPO表、StaffAndAgency表、Drivers表、Stock表、OutBoundOrder表、User表、Order表、OperaLog表。