

快递物流系统软件体系结构描述文档

V1.0



2015-10-15

第7组

|  |  |  |  |
| --- | --- | --- | --- |
| 变更历史 | | | |
| 版本号 | **更新时间** | **更新人** | **更新摘要** |
| V1.0 | 2015/10/15 | 董本超 |  |
|  |  |  |  |

目录

[1引言 3](#_Toc432692868)

[1.1编制目的 3](#_Toc432692869)

[1.2词汇表 3](#_Toc432692870)

[1.3参考资料 3](#_Toc432692871)

[2产品概述 3](#_Toc432692872)

[3逻辑视角 4](#_Toc432692873)

[4组合视角 7](#_Toc432692874)

[4.1开发包图 7](#_Toc432692875)

[4.2运行时进程 7](#_Toc432692876)

[4.3物理部署 7](#_Toc432692877)

[5接口视角 7](#_Toc432692878)

[5.1模块的职责 7](#_Toc432692879)

[5.2用户界面层的分解 7](#_Toc432692880)

[5.3业务逻辑层的分解 7](#_Toc432692881)

[5.4数据层的分解 7](#_Toc432692882)

[6信息视角 7](#_Toc432692883)

[6.1数据持久化对象 7](#_Toc432692884)

[6.2 Txt持久化格式 7](#_Toc432692885)

[6.3数据库表 7](#_Toc432692886)

# 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** |
|  |  |  |
|  |  |  |

## 1.3参考资料

1.快递物流系统用例文档v1.2

2.项目描述v4

3.需求规格说明文档v1.2

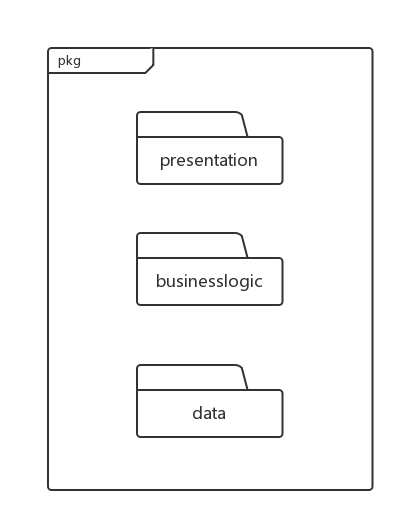
# 2产品概述

快递物流系统是为xxx物流公司开发的新的辅助办公系统，该系统开发的目标是帮助该公司高效处理日益增长的业务，提高员工的工作效率和用户的满意度。

该系统的业务目标为帮助公司保持合理库存，提高用户服务体验，增加业务额，提高公司职工的工作效率，并为总经理的决策做支持。

# 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3层（展示层、业务逻辑层、数据层）能很好的表示整个高层抽象。展示层包含GUI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，其中展示层和业务逻辑层部署在客户端，数据层部署在服务器。分层体系结构的逻辑视角如图1所示。

图1 参考体系结构风格的包图表达逻辑视角

快递物流系统的主要功能有：查询物流信息、收件、派件、车辆信息管理、司机信息管理、营业厅-营业厅转运、营业厅-中转中心转运、中转中心-中转中心转运、快件入库、快件出库、初始化仓库信息、库存查看、库存盘点、收入管理、成本管理、期初建账、银行账户管理、查询系统日志、查询报表、人员管理、机构管理、审批单据、制定薪水策略、制定价格/距离策略、用户账户管理。经过分析发现下列功能可以在设计时合并，因为它们使用的信息和行为相似：

* 查询物流信息🡪物流状态
* 收件、派件、营业厅-营业厅、营业厅-中转中心、中转中心-中转中心转运🡪运输
* 司机、车辆、人员、机构信息管理、银行账户管理、用户账户管理🡪信息管理
* 快件入库、快件出库、初始化仓库信息、库存查看、库存盘点🡪库存
* 收入管理、成本管理、期初建账、查询报表🡪财务
* 查询系统日志🡪日志
* 审批单据🡪单据
* 制定薪水策略、制定价格/距离策略🡪策略

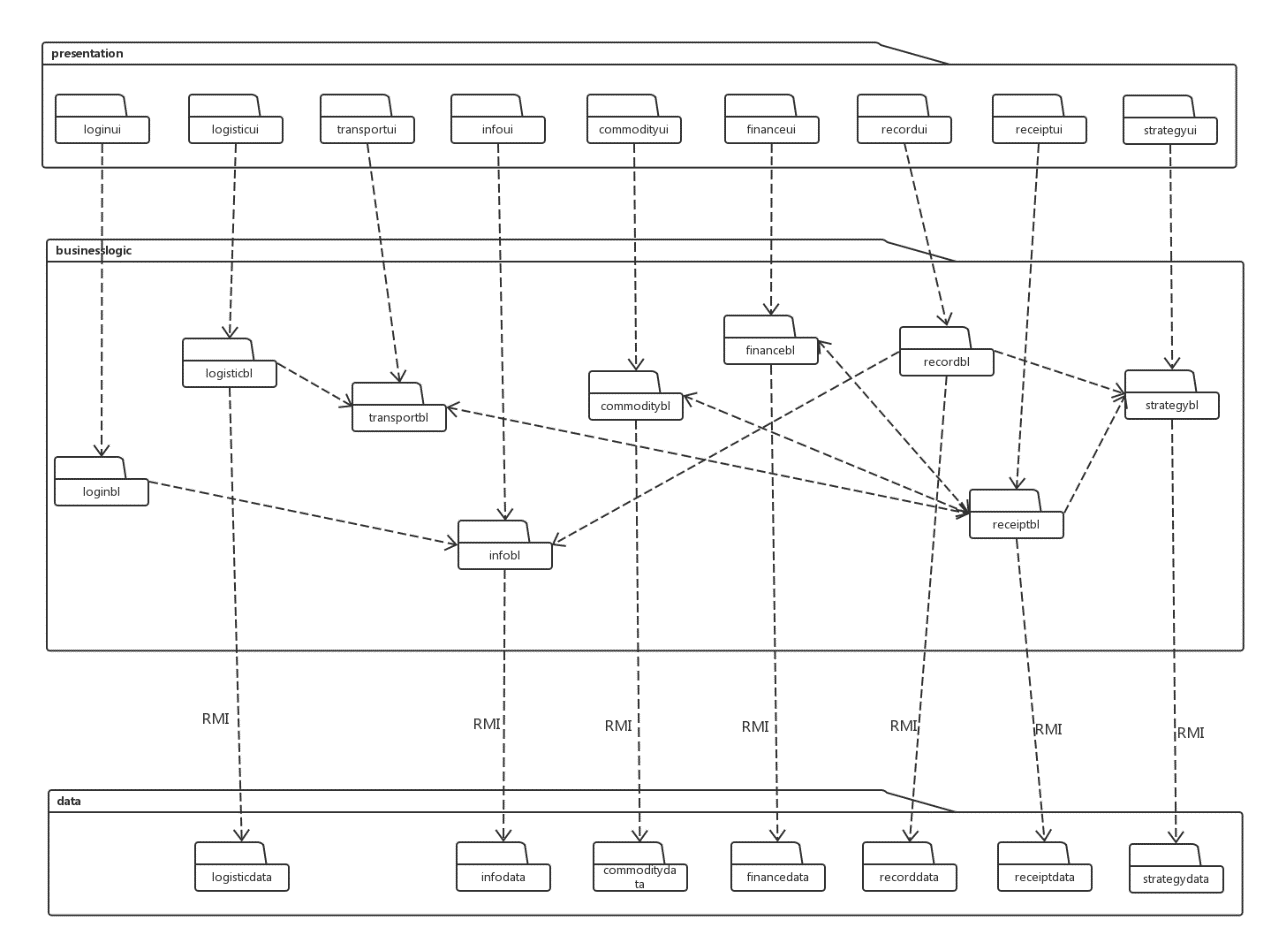
从以上合并后的功能点可以设计出对应的逻辑包：

|  |  |
| --- | --- |
| **功能** | **对应逻辑包** |
| 物流状态 | LogisticsUI, LogisticsBL, LogisticsData |
| 运输 | TransportUI, TransportBL |
| 信息管理 | InfoUI, InfoBL, InfoData |
| 库存 | CommodityUI, CommodityBL, CommodityData |
| 财务 | FinanceUI, FinanceBL, FinanceData |
| 日志 | RecordUI, RecordBL, RecordData |
| 单据 | ReceiptUI, ReceiptBL, ReceiptData |
| 策略 | StrategyUI, StrategyBL, StrategyData |

再加上不同功能之间相互信息使用，上表修改为：

|  |  |
| --- | --- |
| **功能** | **对应逻辑包** |
| 物流状态 | LogisticsUI, LogisticsBL, LogisticsData  TransportBL |
| 运输 | TransportUI, TransportBL  ReceiptBL, ReceiptData |
| 信息管理 | InfoUI, InfoBL, InfoData |
| 库存 | CommodityUI, CommodityBL, CommodityData  ReceiptBL, ReceiptData |
| 财务 | FinanceUI, FinanceBL, FinanceData  ReceiptBL, ReceiptData  StrategyBL, StrategyData |
| 日志 | RecordUI, RecordBL, RecordData  InfoBL, InfoData  StrategyBL, StrategyData |
| 单据 | ReceiptUI, ReceiptBL, ReceiptData  TransportBL  CommodityBL, CommodityData  FinanceBL, FinanceData  StrategyBL, StrategyData |
| 策略 | StrategyUI, StrategyBL, StrategyData |

由上表再加上登录模块（即新增LoginUI, LoginBL）可以生成软件体系结构逻辑设计方案，如图2所示。

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

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** |
| LoginUI | LoginBLService, LogisticUI, TransportUI, InfoUI, CommodityUI, FinanceUI, RecordUI, ReceiptUI, StrategyUI, vo, 界面类库包 |
| LoginBLService | vo |
| LoginBL | LoginBLService, InfoBL, UtilityBL, vo |
| LogisticUI | LogisticBLService, vo, 界面类库包 |
| LogisticBLService | vo |
| LogisticBL | LogisticBLService, LogisticDataService, TransportBL, UtilityBL |
| LogisticDataService | Java RMI, po |
| LogisticData | Java RMI, po, DatabaseUtility |
| TransportUI | TransportBLService, vo, 界面类库包 |
| TransportBLService | vo |
| TransportBL | TransportBLService, UtilityBL |
| InfoUI | InfoBLService, vo, 界面类库包 |
| InfoBLService | vo |
| InfoBL | InfoBLService, InfoDataService, UtilityBL |
| InfoDataService | Java RMI, po |
| InfoData | Java RMI, po, DatabaseUtility |
| CommodityUI | CommodityBLService, vo, 界面类库包 |
| CommodityBLService | vo |
| CommodityBL | CommodityBLService, CommodityDataService, UtilityBL |
| CommodityDataService | Java RMI, po |
| CommodityData | Java RMI, po, DatabaseUtility |
| FinanceUI | FinanceBLService, vo, 界面类库包 |
| FinanceBLService | vo |
| FinanceBL | FinanceBLService, FinanceDataService, StrategyBL, UtilityBL |
| FinanceDataService | Java RMI, po |
| FinanceData | Java RMI, po, DatabaseUtility |
| RecordUI | RecordBLService, vo, 界面类库包 |
| RecordBLService | vo |
| RecordBL | RecordBLService, RecordDataService, InfoBL, StrategyBL, UtilityBL |
| RecordDataService | Java RMI, po |
| RecordData | Java RMI, po, DatabaseUtility |
| ReceiptUI | ReceiptBLService, vo, 界面类库包 |
| ReceiptBLService | vo |
| ReceiptBL | ReceiptBLService, ReceiptDataService, TransportBL, CommodityBL, FinanceBL, StrategyBL, UtilityBL |
| ReceiptDataService | Java RMI, po |
| ReceiptData | Java RMI, po, DatabaseUtility |
| StrategyUI | StrategyBLService, vo, 界面类库包 |
| StrategyBLService | vo |
| StrategyBL | StrategyBLService, StrategyDataService, UtilityBL |
| StrategyDataService | Java RMI, po |
| StrategyData | Java RMI, po, DatabaseUtilityBL |
| vo |  |
| po |  |
| Java RMI |  |
| 界面类库包 |  |
| UtilityBL |  |
| DatabaseUtility | JDBC |

快递物流系统的客户端开发包图如图3所示，服务器端开发包图如图4所示

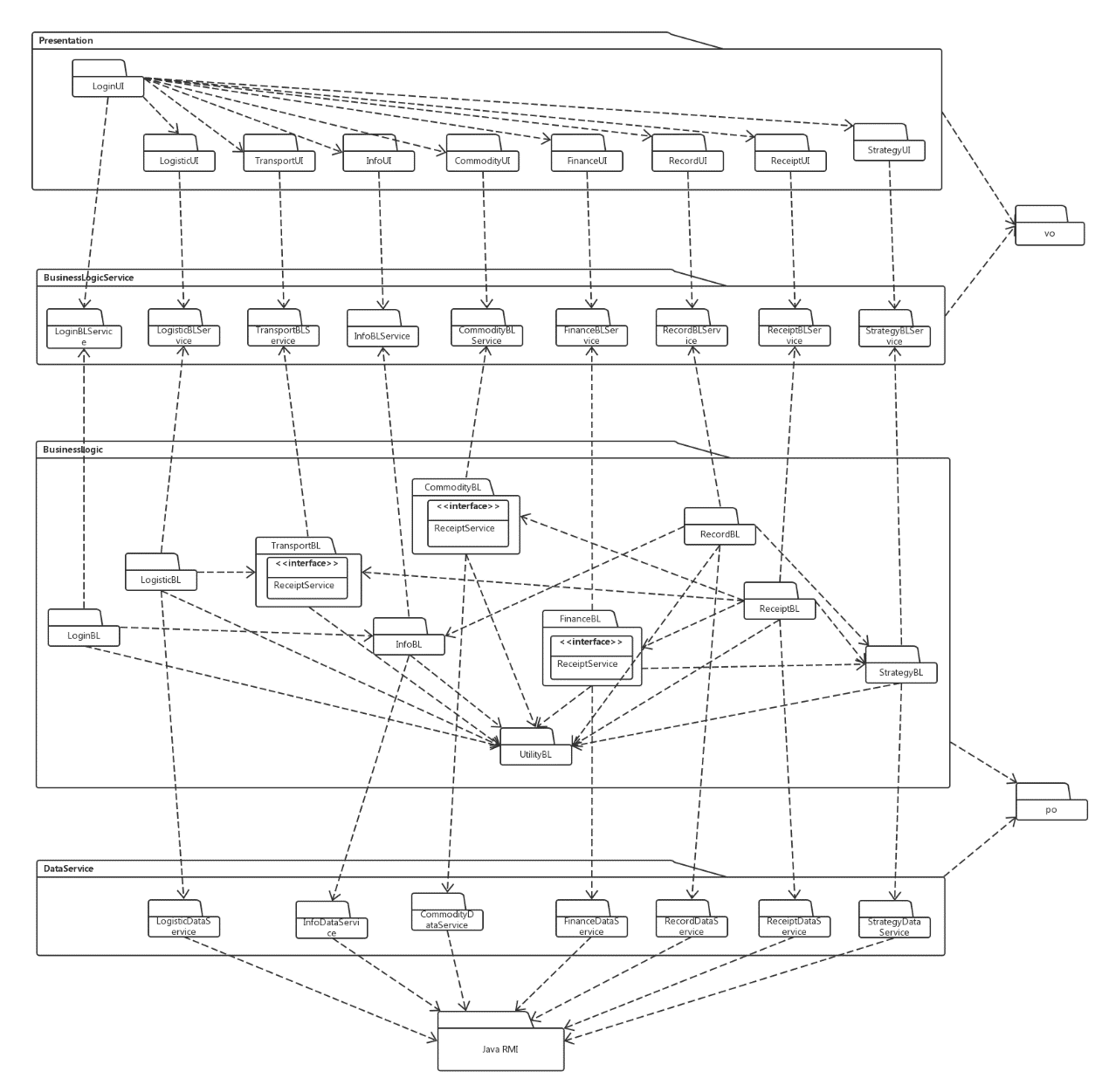


图3 快递物流系统客户端开发包图

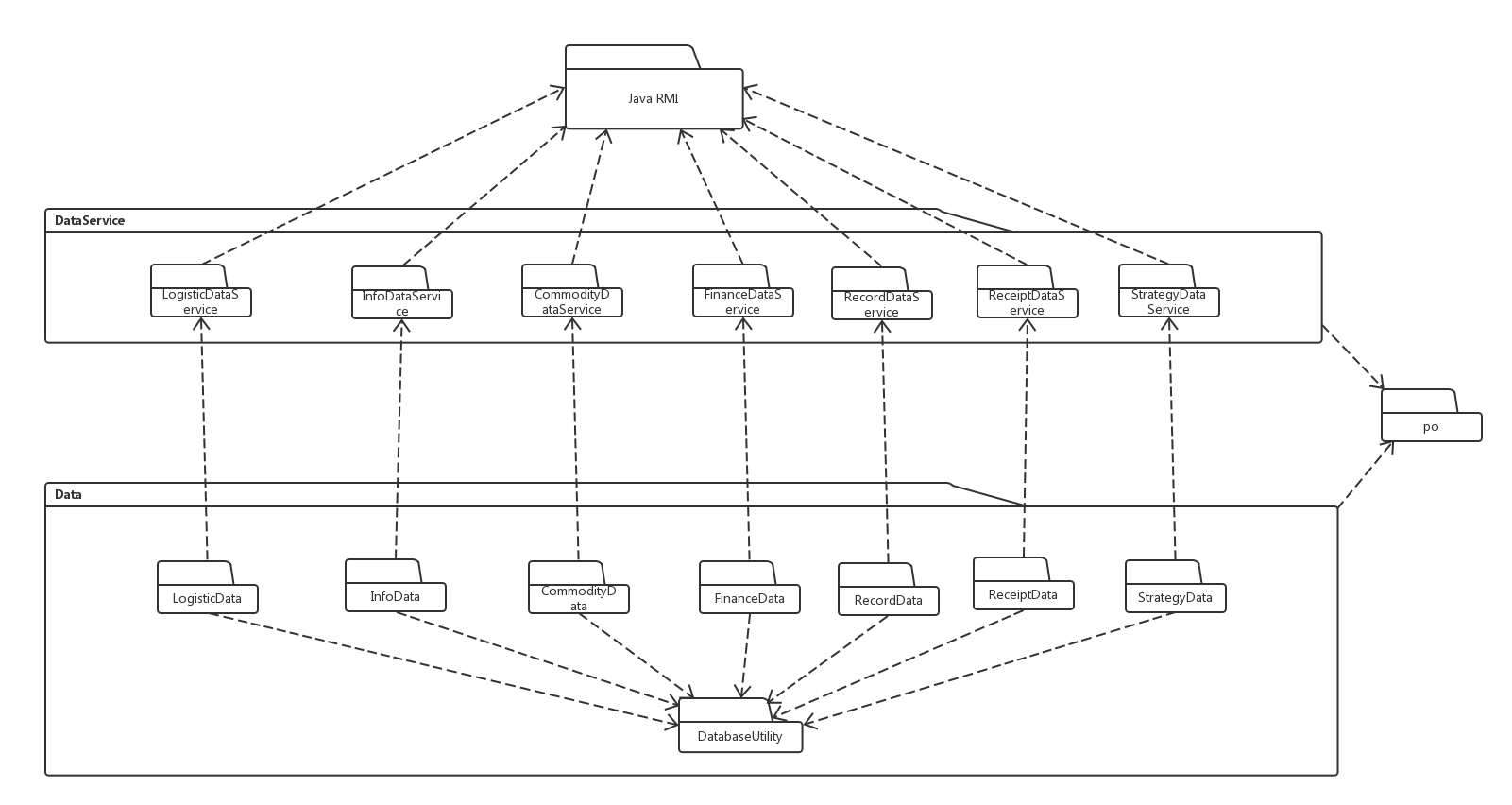
****

图4 快递物流系统服务器端开发包图

## 4.2运行时进程

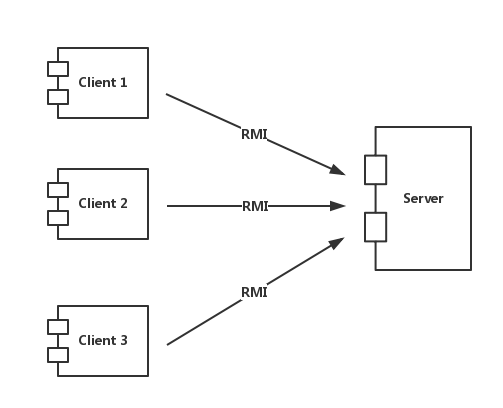
 在快递物流系统中，会有多个客户端进程和一个服务器进程，其进程图如图5所示。结合部署图，客户端进程在客户端机器上运行，服务器进程在服务器端机器上运行。

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的，服务器端构件是放在服务器端机器上的。在客户端节点上，还要部署RMIStub构件。由于Java RMI属于JDK6.0及其以后版本的一部分，所以在系统JDK环境已经设置好的情况下，不需要再独立部署。部署图如图6所示。

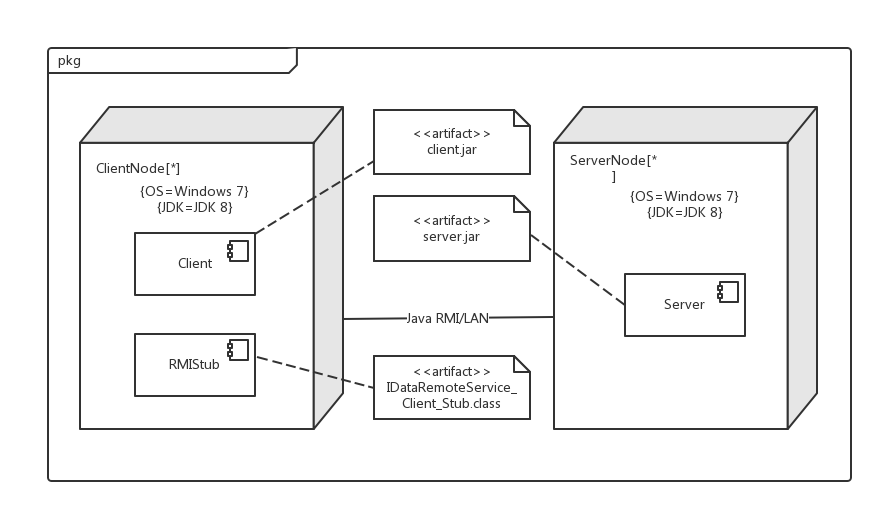
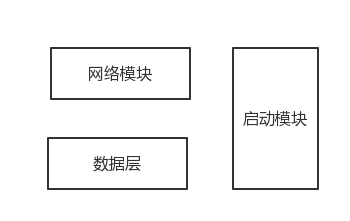
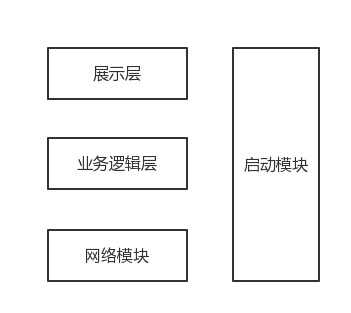


图6 部署图

# 5接口视角

## 5.1模块的职责

 客户端模块和服务器端模块视图分别如图7和图8所示。客户端各层和服务器端各层的职责分别如表2和表3所示。

 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 |
| 展示层 | 基于窗口的快递物流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**

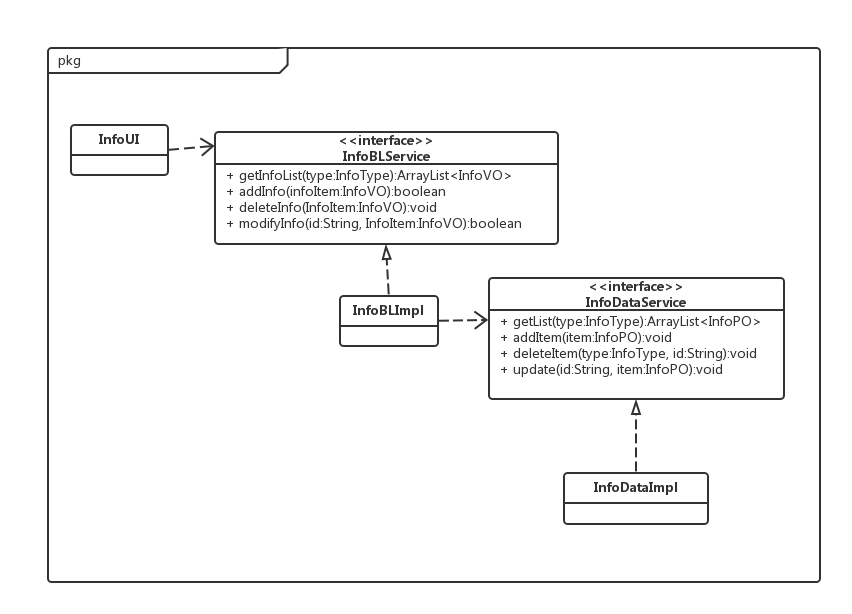
|  |  |
| --- | --- |
| **层** | **职责** |
| 启动模块 | 负责初始化网络通信机制 |
| 数据层 | 负责数据的持久化金数据访问接口 |
| 服务器端网络模块 | 利用Java RMI机制开启RMI服务，注册RMI服务 |

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** |
| LoginBLService  LogisticBLService  TranportBLService  InfoBLService  CommodityBLService  FinanceBLService  RecordBLService  ReceiptBLService  StrategyBLService | 客户端展示层 | 客户端业务逻辑层 |
| LogisticDataService  InfoDataService  CommodityDataService  FinanceDataService  RecordDataService  ReceiptDataService  StrategyDataService | 客户端业务逻辑层 | 服务器端数据层 |

图9-图17展示了各个功能层之间的接口调用。

图12 信息管理用例层之间调用的接口

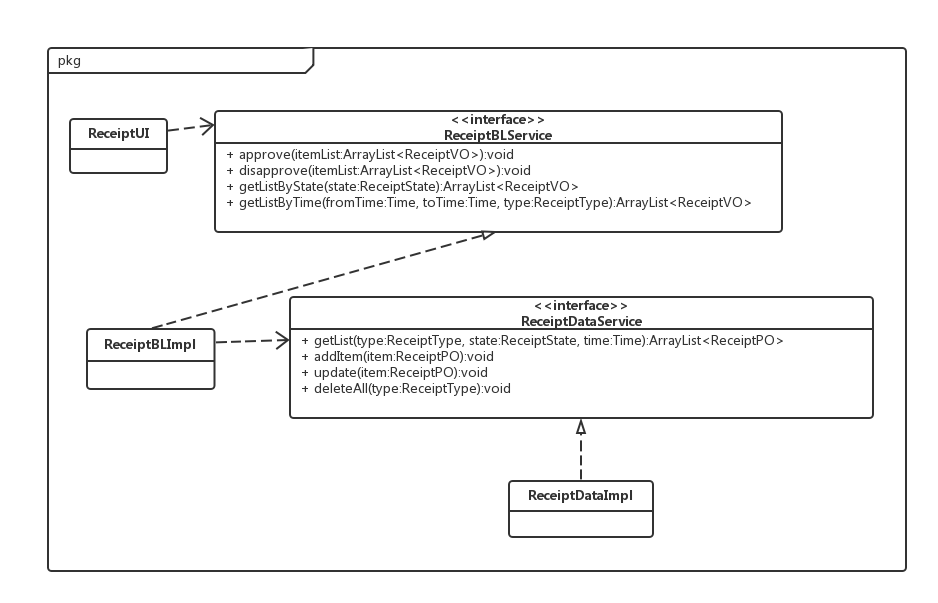


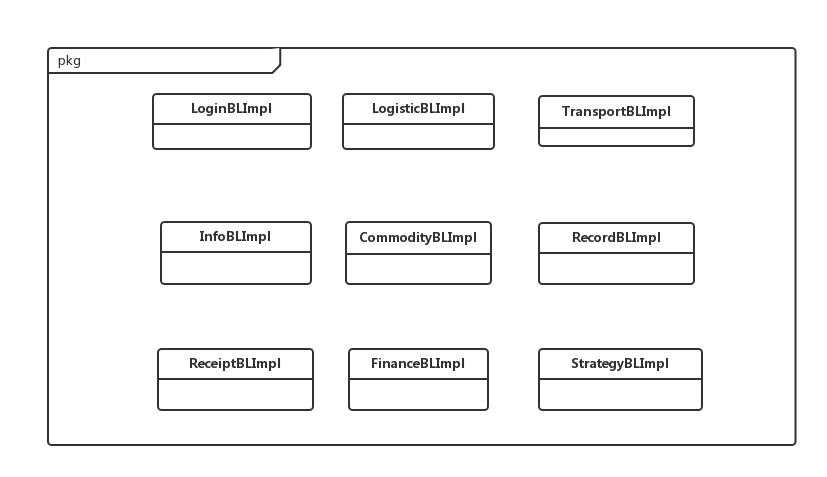
图16 单据用例层间调用的接口

此处为9张图！！！！

## 5.2用户界面层的分解

## 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，LoginBLImpl对象负责处理登录界面的业务逻辑；InfoBLImpl对象负责处理用户账户管理界面的业务逻辑以及LoginBLImpl对象验证密码时请求对应账户密码信息的业务逻辑。业务逻辑层的设计如图20所示。

图20 业务逻辑层的设计

### 5.3.1业务逻辑层模块的职责

业务逻辑层模块的职责如表8所示。

**表8 业务逻辑层模块的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| LoginBL | 负责实现登录界面所需要的服务 |
| LogisticBL | 负责实现查询物流信息界面所需要的服务 |
| TransportBL | 负责实现快递员、营业厅业务员、中转中心业务员在快件收、派、转运过程中对应于系统操作界面的服务，并对LogisticBL模块提供物流信息更新的服务，对ReceiptBL模块提供单据状态改变时向参与者发出通知服务 |
| InfoBL | 负责实现车辆、司机、人员、机构、用户账户、银行账户管理界面所需要的服务，并对LoginBL模块提供验证登录密码的服务，对RecordBL提供信息变更时发出通知的服务 |
| CommodityBL | 负责实现快件入库、出库、初始化仓库信息、库存查看、库存盘点界面所需要的服务，并对ReceiptBL模块提供单据状态改变时向参与者发出通知服务 |
| RecordBL | 负责实现查询系统日志界面所需要的服务 |
| ReceiptBL | 负责实现审批单据界面所需要的服务，并对TransportBL模块、CommodityBL模块、FinanceBL模块提供新建相应单据的服务，尤其对FinanceBL模块中查询报表功能提供查询收款单和入款单的服务 |
| FinanceBL | 负责实现收入管理、成本管理、期初建账、查询报表界面所需要的服务，并对ReceiptBL模块提供单据状态改变时向参与者发出通知服务 |
| StrategyBL | 负责实现制定策略界面所需要的服务，并对FinanceBL模块提供人员工资计算相关服务，对RecordBL模块提供策略变更时发出通知的服务，对ReceiptBl模块提供计算快递费用和运输费用的服务 |
| UtilityBL | 负责业务逻辑层所有模块的初始化和公用操作 |

### 5.3.2业务逻辑层模块的接口规范

此处为表9—表17

**表12 InfoBL模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| InfoBLImpl.getInfoList | 语法 | public ArrayList<InfoVO> getInfoList(InfoType type) |
| 前置条件 | 无 |
| 后置条件 | 在系统中展示司机、车辆、人员、机构、用户账户或银行账户的信息列表 |
| InfoBLImpl.addInfo | 语法 | public boolean addInfo(InfoVO infoItem) |
| 前置条件 | 已经打开相关信息列表 |
| 后置条件 | 添加成功则返回true并更新相关持久化数据并通知更新系统日志，添加失败则返回false |
| InfoBLImpl.deleteInfo | 语法 | public void deleteInfo(InfoVO infoItem) |
| 前置条件 | 已经打开相关信息列表 |
| 后置条件 | 更新相关持久化数据并通知更新系统日志 |
| InfoBLImpl.modifyInfo | 语法 | public boolean modifyInfo(InfoVO infoItem) |
| 前置条件 | 已经打开相关信息列表 |
| 后置条件 | 修改成功则返回true并更新持久化数据并通知更新系统日志，失败则返回false |
| InfoBLImpl.verifyPassword | 语法 | public boolean verifyPassword(String id, String password) |
| 前置条件 | 已输入用户名和密码 |
| 后置条件 | 验证成功则返回true，否则返回false |
| InfoBLImpl.notifyRecord | 语法 | public void notifyRecord(String operation) |
| 前置条件 | 对应的RecordBLImpl已经向InfoBLImpl注册 |
| 后置条件 | 监听对象更新系统日志持久化数据 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| InfoDataService.getList(InfoType type) | 得到type类型信息（如司机信息、机构信息等）的持久化对象列表 | |
| InfoDataService.addItem(InfoPO item) | 向数据库中添加一个持久化对象 | |
| InfoDataService.deleteItem(InfoType type, String id) | 从数据库中删除一个持久化对象 | |
| InfoDataService.update(String id, InfoPO item) | 更新单一持久化对象 | |

**表16 ReceiptBL模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| ReceiptBLImpl.getListByState | 语法 | public ArrayList<ReceiptVO> getListByState(ReceiptState state) |
| 前置条件 | 无 |
| 后置条件 | 返回对应state的单据列表 |
| ReceiptBLImpl.getListByTime | 语法 | public ArrayList<ReceiptVO> getListByTime(Time fromTime, Time toTime, ReceiptType type) |
| 前置条件 | 无 |
| 后置条件 | 返回对应时间段内对应type的单据列表 |
| ReceiptBLImpl.approve | 语法 | public void approve(ArrayList<ReceiptVO> itemList) |
| 前置条件 | 已经打开待审批或未通过单据列表 |
| 后置条件 | 更新对应单据持久化数据，并且调用ReceiptBLImpl.notifyActors接口来通知相应参与者 |
| ReceiptBLImpl.disapprove | 语法 | public void disapprove(ArrayList<ReceiptVO> itemList) |
| 前置条件 | 已经打开待审批单据列表 |
| 后置条件 | 更新对应单据持久化数据，并且调用ReceiptBLImpl.notifyActors接口来通知相应参与者 |
| ReceiptBLImpl.createReceipt | 语法 | public void createReceipt(ReceiptVO item) |
| 前置条件 | 无 |
| 后置条件 | 如果有需要，根据item中的相关信息（如出发地、目的地），根据对应策略完成单据的缺省项（如费用），并更新单据持久化数据 |
| ReceiptBLImpl.notifyActors | 语法 | public void notifyActors(ArrayList<ReceiptVO> itemList) |
| 前置条件 | 对应的TransportBLImpl, CommodityBLImpl, FinanceBLImpl已经向ReceiptBLImpl注册 |
| 后置条件 | 监听对象更新相关单据状态信息 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| ReceiptDataService.getList(ReceiptType type, Time time) | 得到数据库中对应type, state, time的单据列表 | |
| ReceiptDataService.addItem(ReceiptPO item) | 向单据数据库中添加一个持久化对象 | |
| ReceiptDataService.update(ReceiptPO item) | 更新单一持久化对象 | |
| ReceiptDataService.deleteAll(ReceiptType type) | 清空数据库中type类型的所有单据 | |
| StrategyBLImpl. | 此处接口待花蕾定义！！！ | |

## 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：txt文件、序列化文件、数据库等，所以抽象了数据服务。

图21到图27展示了各个数据层模块的描述。

此处为7张图！！！

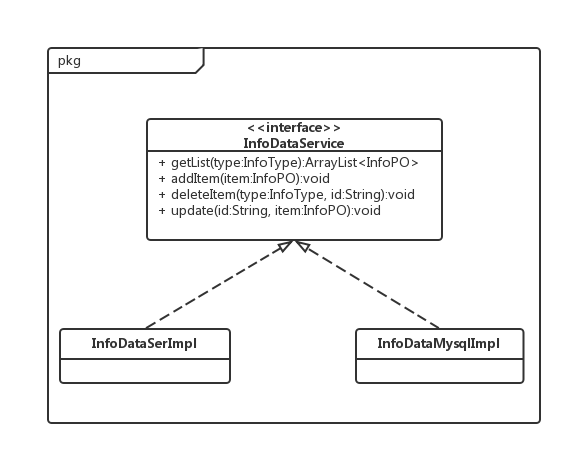


图22 信息管理数据层模块的描述

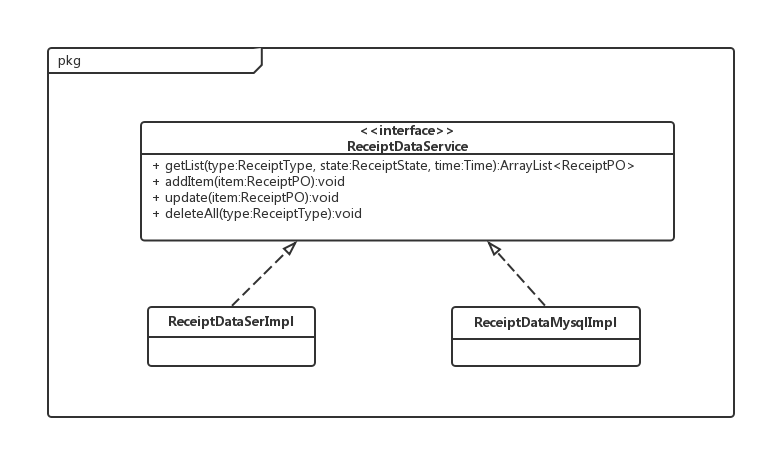


图26 单据数据层模块的描述

### 5.4.1数据层模块的职责

由于各个功能模块对应数据层模块的职责基本相同，故这里只用一张表总结描述，如表18所示。

**表18 数据层模块的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| \*DataService | 持久化数据操作的接口，提供持久化数据的增、删、改、查服务 |
| \*DataTxtImpl | 基于txt文件的持久化数据操作的接口，提供持久化数据的增、删、改、查服务 |
| \*DataSerImpl | 基于序列化文件的持久化数据操作的接口，提供持久化数据的增、删、改、查服务 |
| \*DataMysqlImpl | 基于Mysql数据库持久化数据操作的接口，提供持久化数据的增、删、改、查服务 |

### 5.4.2数据层模块的接口规范

表19到表25为各个功能数据层模块的接口规范。

此处为7张表格！！！！

**表20 信息管理数据层模块的接口规范**

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| InfoDataService.getList | 语法 | public ArrayList<InfoPO> getList(InfoType type) throws RemoteExceptoin; |
| 前置条件 | 无 |
| 后置条件 | 按type返回相应的InfoPO列表 |
| InfoDataService.addItem | 语法 | public void addItem(InfoPO item) throws RemoteException; |
| 前置条件 | 相同id的po在列表中不存在 |
| 后置条件 | 在持久化数据中增加一个po条目 |
| InfoDataService.deleteItem | 语法 | public void deleteItem(InfoType type, String id) throws RemoteException |
| 前置条件 | 持久化数据列表中存在一个相同type和id的po |
| 后置条件 | 删除一个po |
| InfoDataService.update | 语法 | public void update(String id, InfoPO item) throws RemoteException |
| 前置条件 | 在持久化数据列表中存在相同id的po |
| 后置条件 | 更新一个po |

**表24 单据数据层模块的接口规范**

# 6信息视角

## 6.1数据持久化对象

系统的PO就是对应的需要持久化的类，所有的po类的介绍如下所示。

* LogisticPO
* InfoPO类是车辆、司机、人员、机构、用户账户、银行账户的持久化信息父类，其对应的子类包括====================to be continued==================

## 6.2 Txt持久化格式

## 6.3数据库表