**物流管理系统**

**详细设计描述文档**

**V3.0**

**南京大学软件学院**

第23组：“梦江文月夜”队

2015-11-15

**文档版本**

|  |  |  |  |
| --- | --- | --- | --- |
| **修改人员** | **日期** | **变更原因** | **版本号** |
| 全体成员 | 2015/11/5 | 初步详细设计，尚有部分未完成 | V0.1 |
| 徐江河 | 2015/11/15 | 增添了关键类图 | V1.0 |
| 徐江河 | 2015/12/9 | 修改了部分类的接口 | V2.0 |
| 汪文藻、吴秦月 | 2015/12/31 | 更新数据层接口 | V3.0 |

目录

1 引言 4

1.1编制目的 4

1.2词汇表 4

1.3参考资料 4

2 产品概述 4

3 体系结构设计概述 4

4 结构视角 5

4.1业务逻辑层的分解 5

4.1.1 logisticBL模块 5

4.1.2 commodityBL模块 18

4.1.3 financeBL模块 25

4.1.4 statisticBL模块 35

4.1.5 infobl模块 52

4.2 数据层的分解 67

4.2.1 数据层模块的职责 68

4.2.2 数据层模块的接口规范 69

5 依赖视角 79

## 1 引言

### 1.1编制目的

本报告详细完成对连锁商店管理系统的详细设计，达到指导后续软件构造的目的，同时实现和测试人员以及用户的沟通。

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

（无）

### 1.3参考资料

1.《软件工程与计算（二）软件开发的技术基础》

2.物流管理系统软件需求规格说明文档V2.0

## 2 产品概述

本物流管理系统为满足XX快递公司的提高用户服务体验、提高员工工作效率、提供决策数据支持、管理库存等需求而设计，能够服务于公司内的所有种类员工与公司客户，为他们提供即时的信息查询、业务操作功能。

本系统运行于Java平台，具有联网功能，采用图形化用户界面，最适合在台式机与笔记本电脑上使用。

## 3 体系结构设计概述

本系统采用分层风格的软件体系，整个系统分为展示层、业务逻辑层与数据层，其中展示层与业务逻辑层存在于软件客户端，在经过身份验证后，可以访问服务器端的数据层，通过RMI的方式查询、操作业务数据。

为了降低系统内部件耦合度，提高内聚度，本系统内建立了businesslogicservice与dataservice两个包，存放各个功能实现类所提供或依赖的功能接口。

## 4 结构视角

### 4.1业务逻辑层的分解

#### 4.1.1 logisticBL模块

（1）模块概述

logisticbl模块负责实现物流过程中各项单据的输入功能。

logisticbl模块的需求可参见需求规格说明文档的功能需求与相关非功能性需求。

logisticbl模块的职责及接口参见软件体系结构设计文档内容。

（2）整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们添加了接口。比如展示层和业务逻辑层之间，我们添加bussinesslogicservice.logisticblservice接口包，内部含有ArrivalNoteOnServiceBLService等7个接口。业务逻辑层和数据层之间添加dataservice.logisticdataservice接口包，内部含有ArrivalNoteOnServiceBLService等7个接口。ArrivalNoteOnServicePO等单据的po对象是作为单据类别的持久化对象被添加到设计模型中去的。

logisticbl模块的设计如图1所示

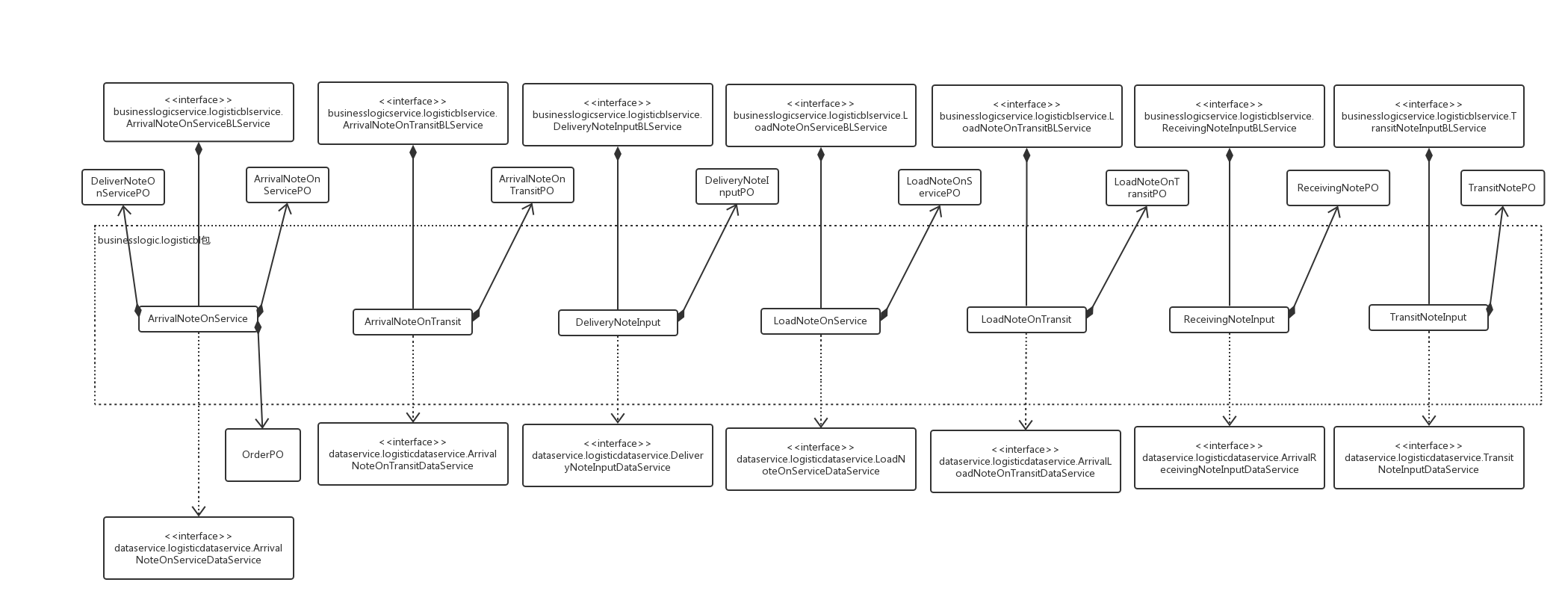


图1 logisticbl模块各个类的设计

logistic模块各个类的职责如表1所示

**表1 logistic模块各个类的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| ArrivalNoteOnService | 处理营业厅到达件的领域模型对象，拥有营业厅到达单和营业厅派送单单据的信息，可以帮助完成营业厅业务员处理到达件所需要的服务 |
| ArrivalNoteOnTransit | 输入中转单的领域模型对象，拥有中转单的信息，可以帮助完成中转中心业务员输入中转单所需要的服务 |
| DeliveryNoteInput | 输入寄件单的领域模型对象，拥有寄件单的信息与货物物流信息，可以帮助完成快递员输入寄件单所需要的服务 |
| LoadNoteOnService | 输入营业厅装车单的领域模型对象，拥有营业厅装车单的信息，可以帮助完成营业厅业务员输入装车单所需要的服务 |
| LoadNoteOnTransit | 输入中转中心装车单的领域模型对象，拥有中转中心装车单的信息，可以帮助完成中转中心业务员输入装车单所需要的服务 |
| ReceivingNoteInput | 输入收件单的领域模型对象，拥有收件单的信息，可以帮助完成快递员输入装收件单所需要的服务 |
| TransitNoteInput | 输入中转中心中转单的领域模型对象，拥有中转单的信息，可以帮助完成中转中心业务员输入中转单所需要的服务 |

（3） 模块内部的接口规范

logistic模块的接口规范如表2所示

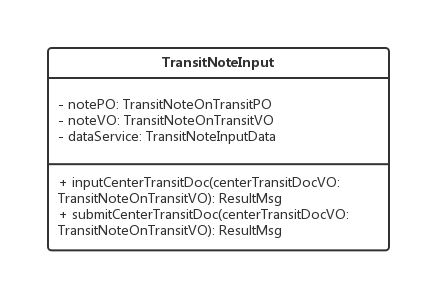
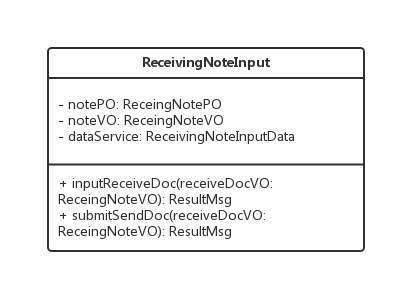
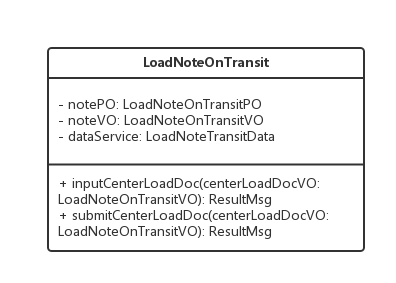
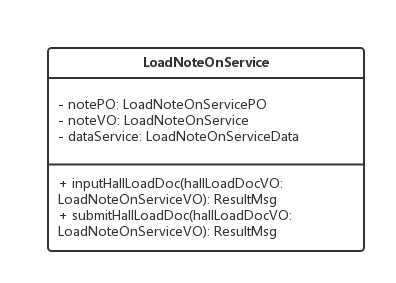
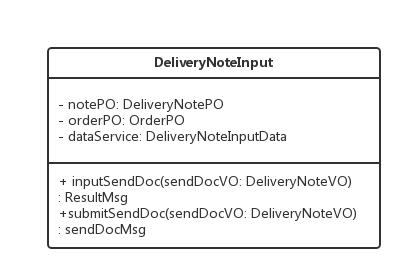
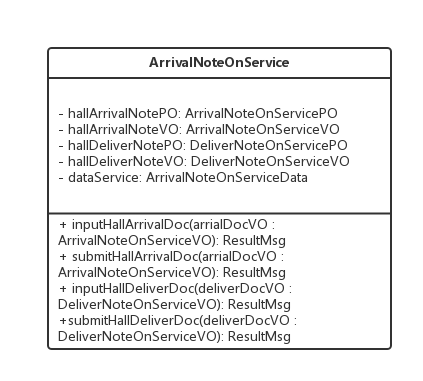
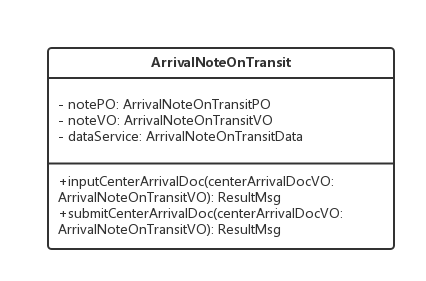
**logisticbl模块提供的的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 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**

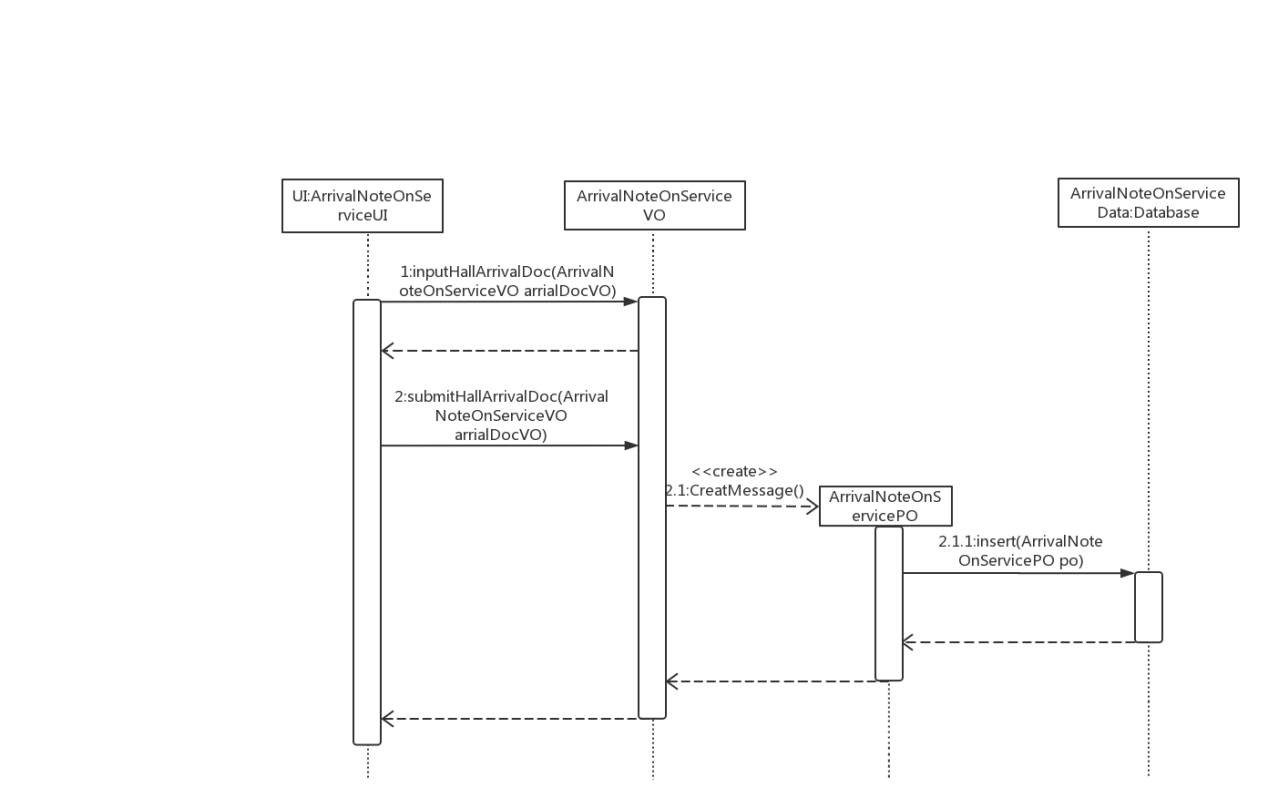
|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| ArrivalNoteOnServiceDataService | public void insert(ArrivalNoteOnServicePO po) throws RemoteException | 在数据库中增加一个po记录 |
| ArrivalNoteOnTransitDataService | public void insert(ArrivalNoteOnTransitPO po) throws RemoteException | 在数据库中增加一个po记录 |
| DeliveryNoteInputDataService | 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 |

**logistic模块的关键类图**

****

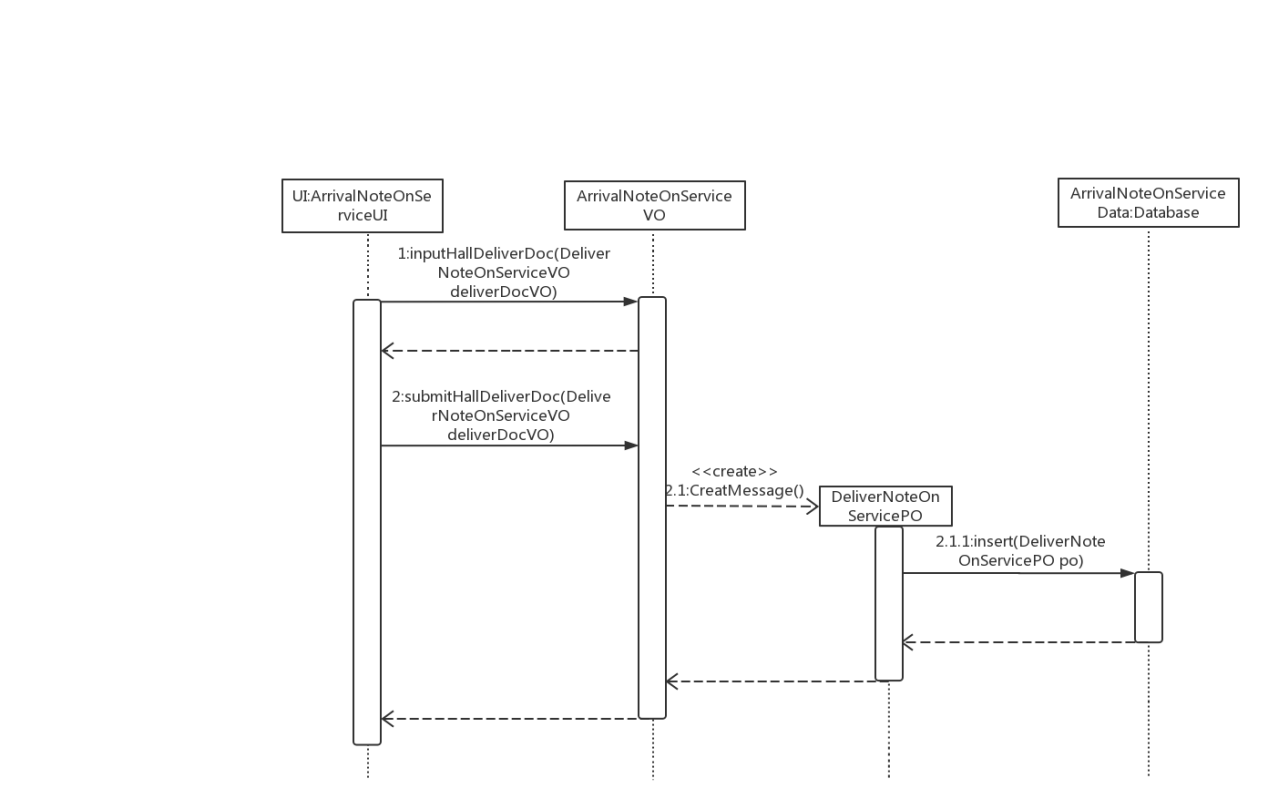
（4）业务逻辑层的动态模型

图1表明了物流管理系统中，当营业厅业务员输入与提交营业厅到达单，处理营业厅到达件业务逻辑的相关对象之间的协作。



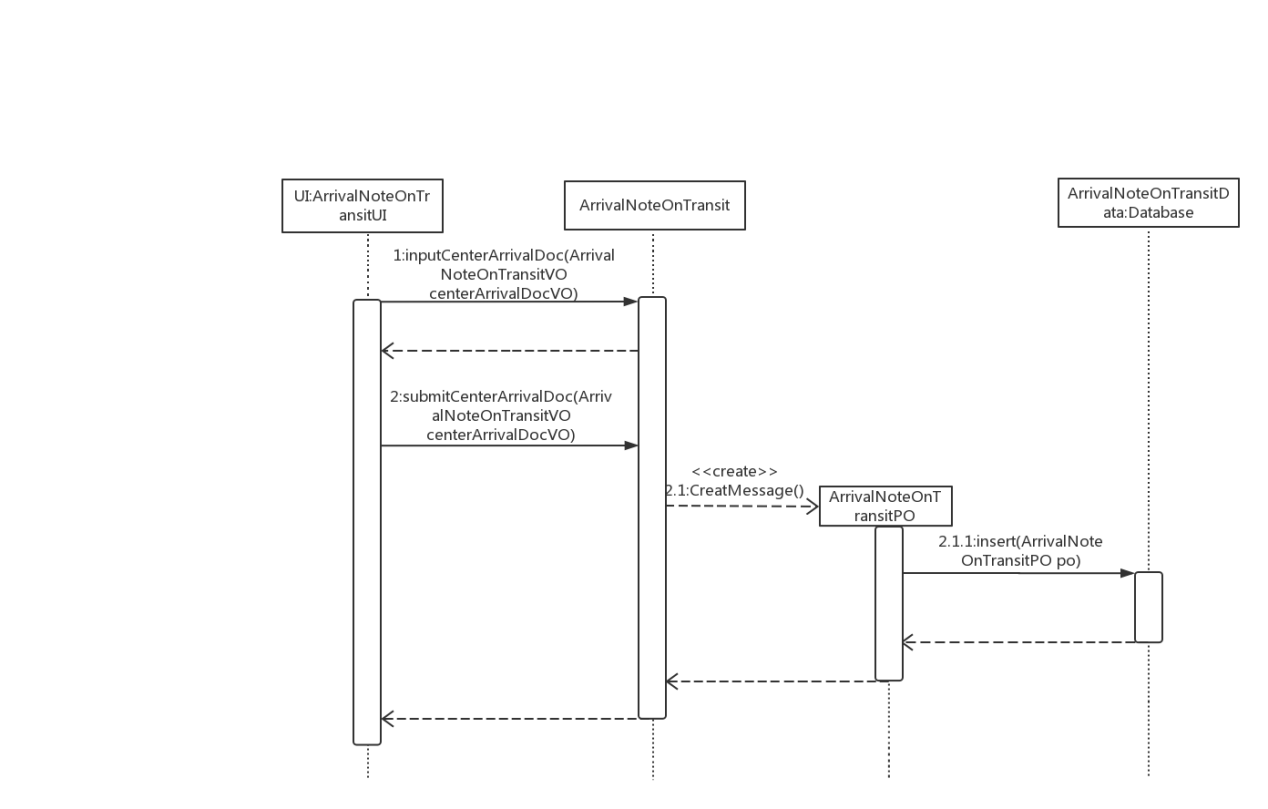
输入营业厅到达单的顺序图

图2表明了物流管理系统中，当营业厅业务员输入与提交营业厅派件单，处理营业厅到达件业务逻辑的相关对象之间的协作。



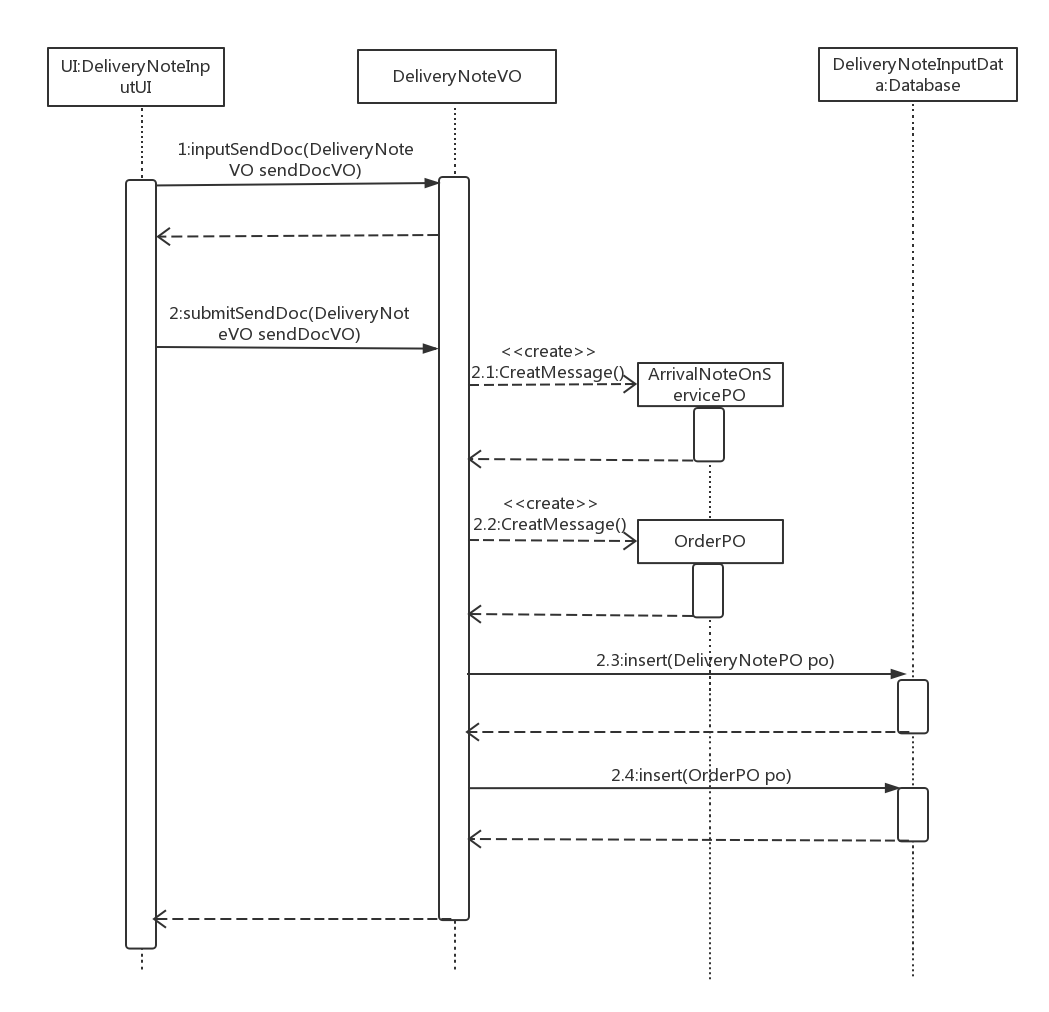
输入营业厅派件单的顺序图

下图表明了物流管理系统中，当中转中心业务员输入与提交中转中心到达单，处理中转中心到达件业务逻辑的相关对象之间的协作。



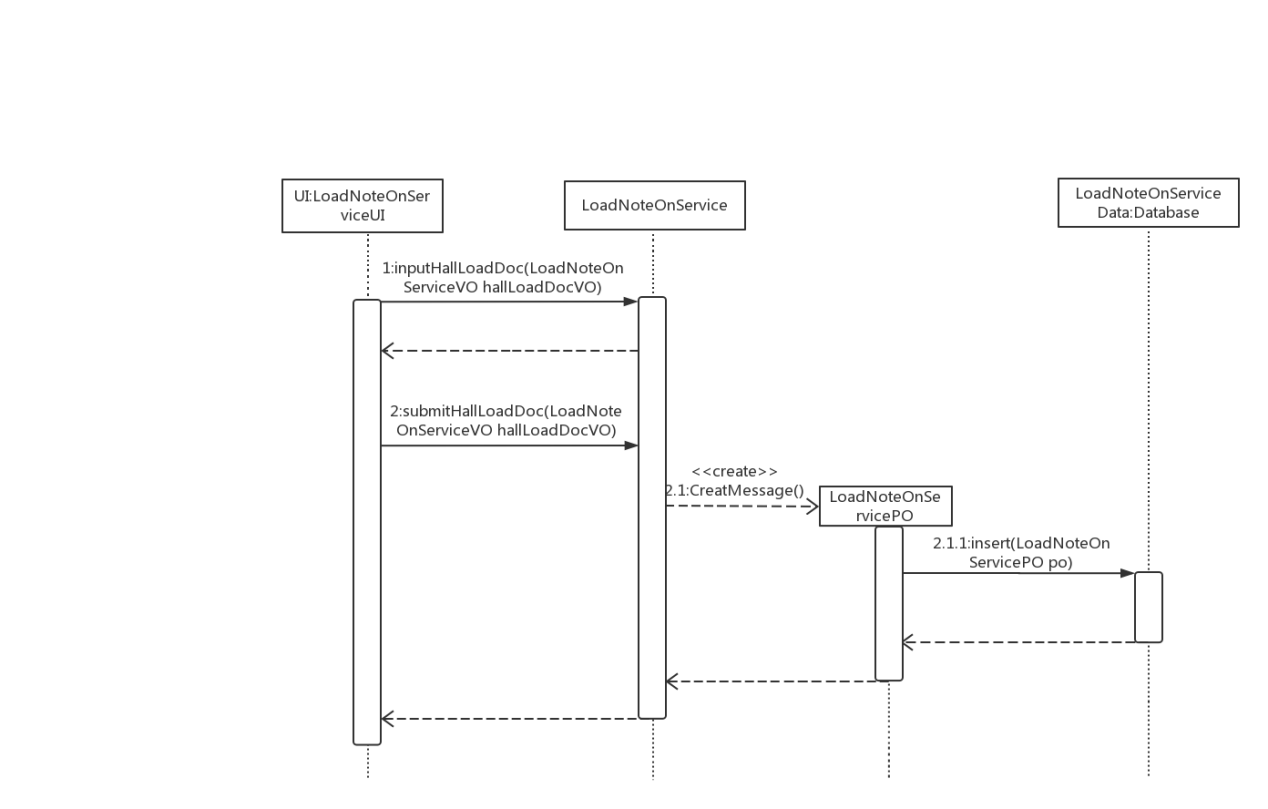
输入中转中心到达单的顺序图

图4表明了物流管理系统中，当营业厅业务员输入与提交寄件单，处理客户寄件业务逻辑的相关对象之间的协作。



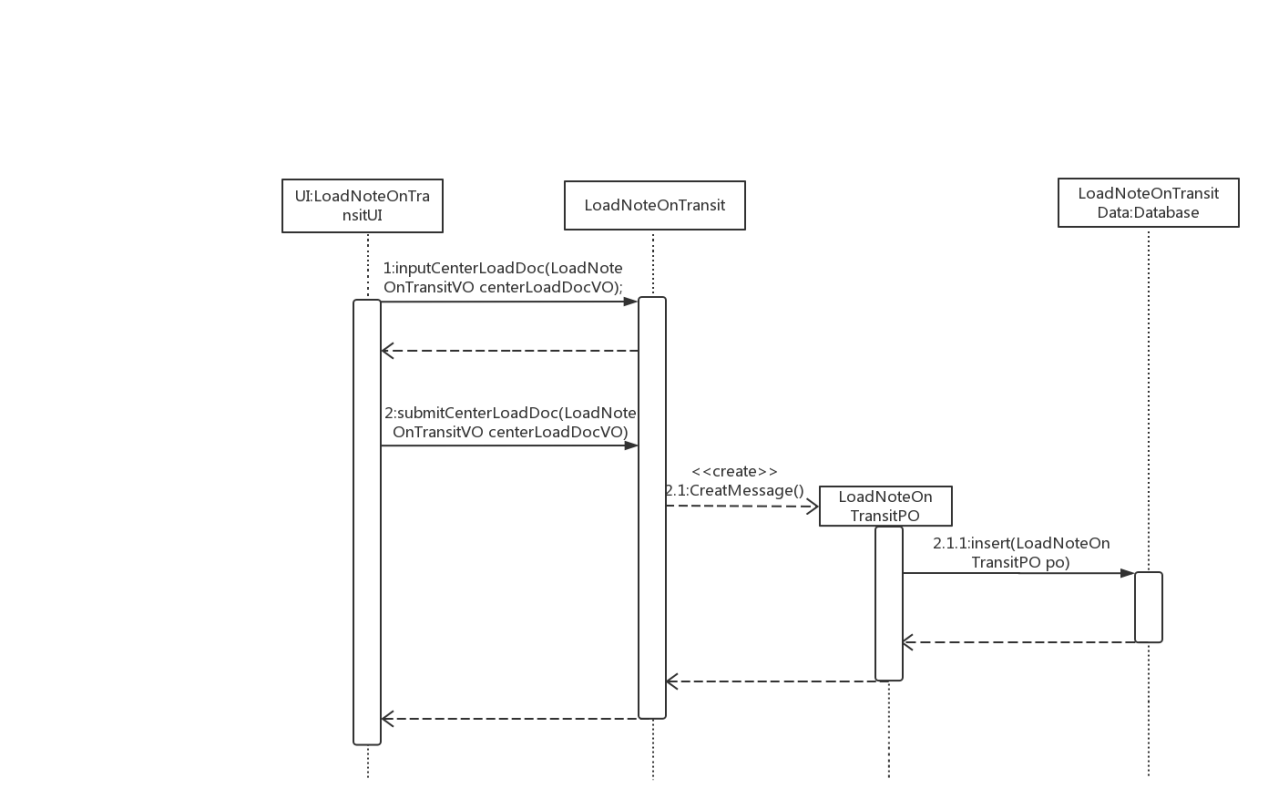
输入寄件单的顺序图

图5表明了物流管理系统中，当营业厅业务员输入与提交营业厅装车单信息，发送营业厅货物业务逻辑的相关对象之间的协作。



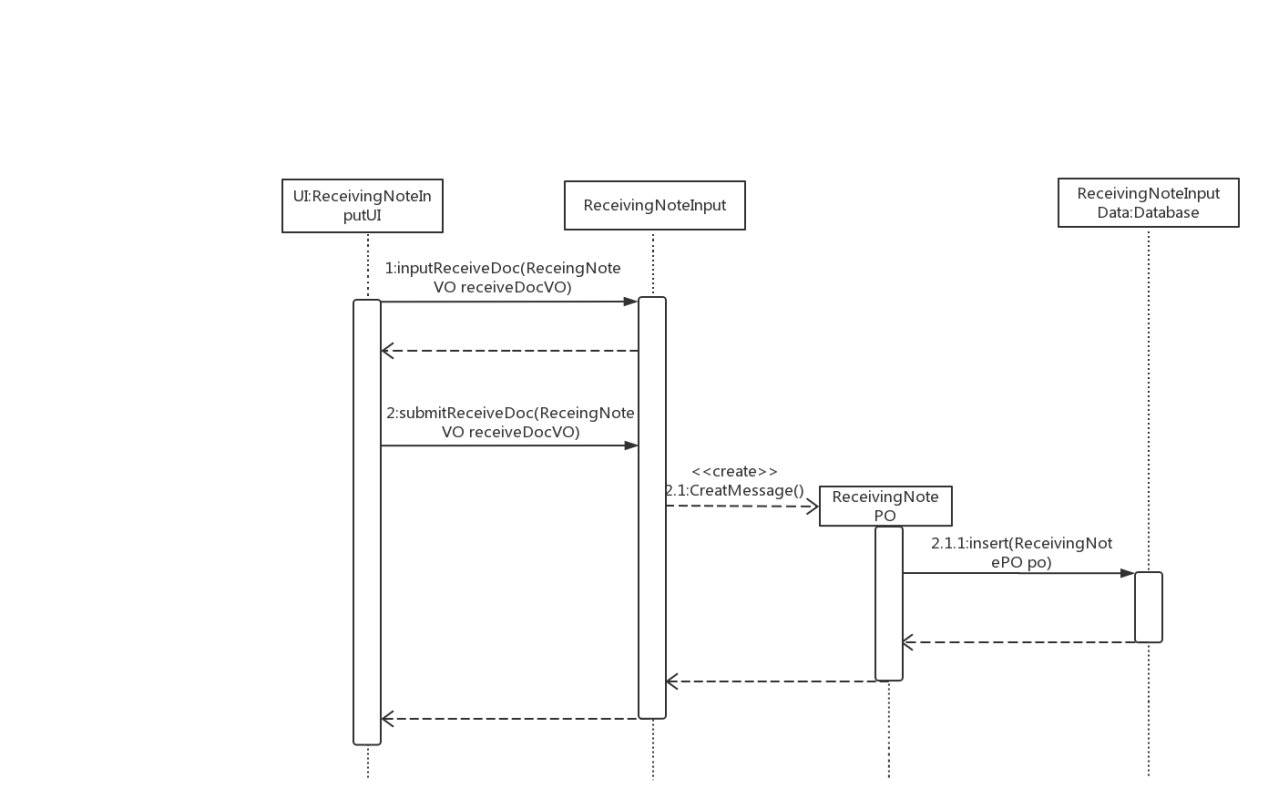
输入营业厅装车单信息的顺序图

图6表明了物流管理系统中，当中转中心业务员输入与提交中转中心装车单信息，发送中转中心货物至本地营业厅业务逻辑的相关对象之间的协作。



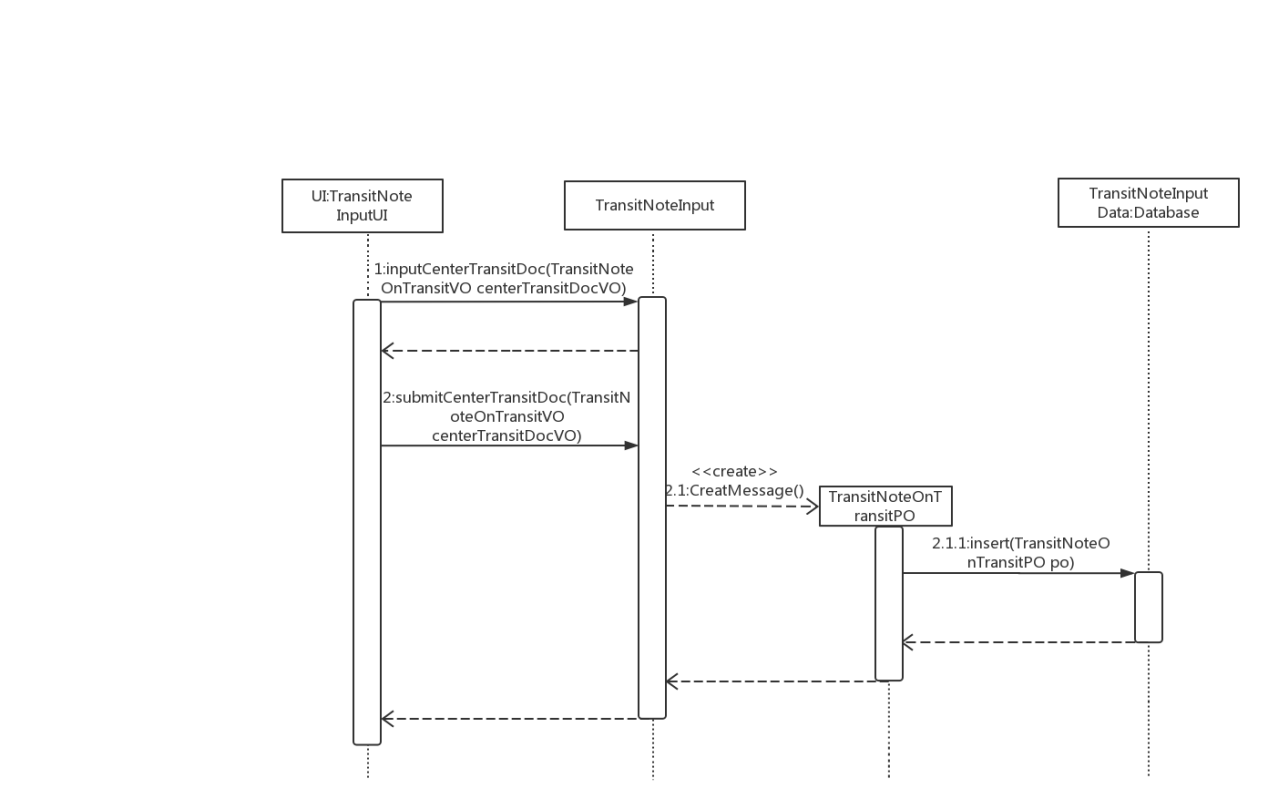
输入中转中心装车单信息的顺序图

图7表明了物流管理系统中，当快递员输入与提交收件单，处理客户收件业务逻辑的相关对象之间的协作。



输入收件单的顺序图

图8表明了物流管理系统中，当中转中心业务员输入与提交中转中心中转单，发送中转中心货物至目的地中转中心业务逻辑的相关对象之间的协作。



输入中转单的顺序图

图1所示的状态图描述了ArrivalNoteOnService对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inputHallArrivalDoc方法被UI调用，ArrivalNoteOnService对象被创建，对象进入单据输入检查格式状态。格式检查通过以后之后UI调用submitHallArrivalDoc方法，向数据层添加单据po持久对象，业务逻辑对象被回收。

inputHallDeliverDoc过程中ArrivalNoteOnService对象的状态包含在此图内。

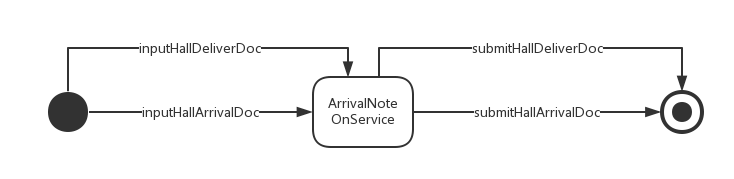
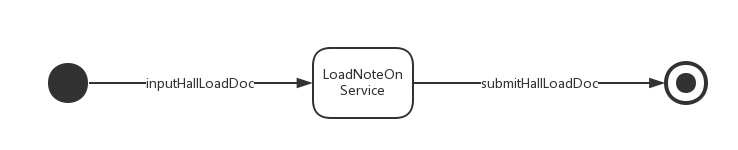


图2所示的状态图描述了ArrivalNoteOnTransit对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inputCenterArrivalDoc方法被UI调用，ArrivalNoteOnTransit对象被创建，对象进入单据输入检查格式状态。格式检查通过以后之后UI调用submitCenterArrivalDoc方法，向数据层添加单据po持久对象，业务逻辑对象被回收。



下图所示的状态图描述了 DeliveryNoteInput对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inputSendDoc方法被UI调用， DeliveryNoteInput对象被创建，对象进入单据输入检查格式状态。格式检查通过以后之后UI调用submitSendDoc方法，向数据层添加单据po持久对象，业务逻辑对象被回收。

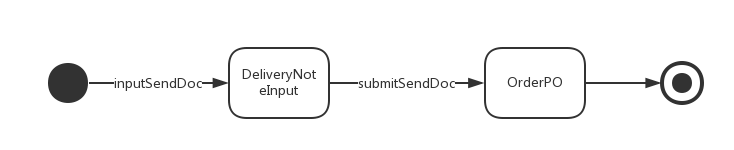


图4所示的状态图描述了LoadNoteOnService对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inputHallLoadDoc方法被UI调用，LoadNoteOnService对象被创建，对象进入单据输入检查格式状态。格式检查通过以后之后UI调用submitHallLoadDoc方法，向数据层添加单据po持久对象，业务逻辑对象被回收。

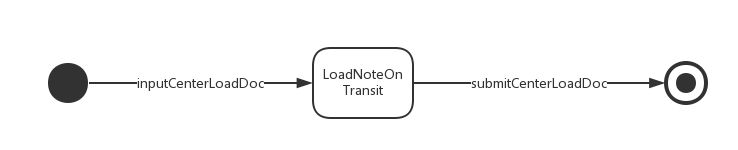


图5所示的状态图描述了LoadNoteOnTransit对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inputCenterLoadDoc方法被UI调用，LoadNoteOnTransit对象被创建，对象进入单据输入检查格式状态。格式检查通过以后之后UI调用submitCenterLoadDoc方法，向数据层添加单据po持久对象，业务逻辑对象被回收。

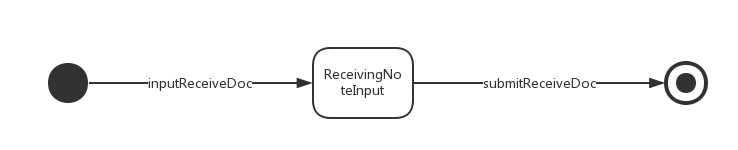


图6所示的状态图描述了ReceivingNoteInput对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inputReceiveDoc方法被UI调用，ReceivingNoteInput对象被创建，对象进入单据输入检查格式状态。格式检查通过以后之后UI调用submitSecdDoc方法，向数据层添加单据po持久对象，业务逻辑对象被回收。

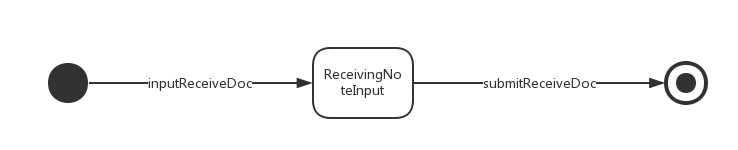
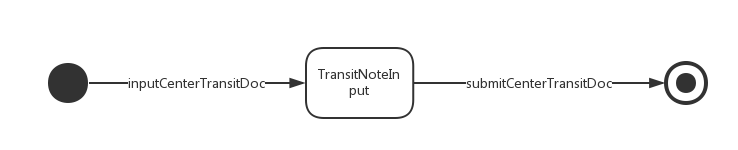


图7所示的状态图描述了TransitNoteInput对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inputCenterTransitDoc方法被UI调用，TransitNoteInput对象被创建，对象进入单据输入检查格式状态。格式检查通过以后之后UI调用submitCenterTransitDoc方法，向数据层添加单据po持久对象，业务逻辑对象被回收。



#### 4.1.2 commodityBL模块

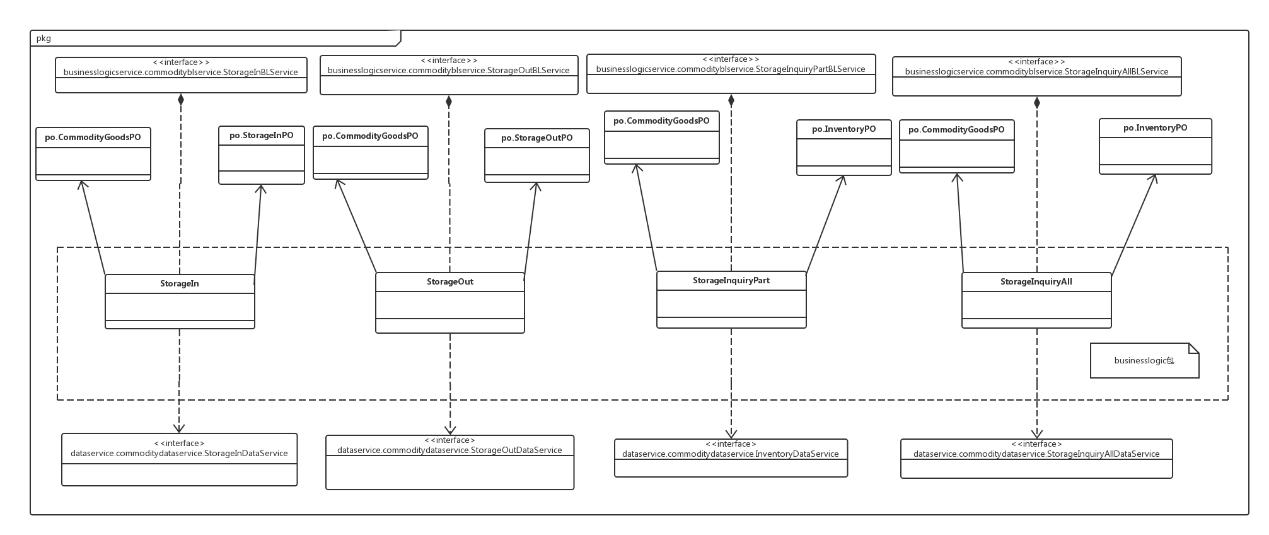
（1）模块概述

commodityBL模块承担的需求参见需求规格说明文档及相关非功能性需求。

commodityBL模块的职责及接口参见软件系统结构描述。

（2）整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层，并在层间添加接口以保证系统灵活性，降低部件耦合度。在本模块中，各类实现businesslogicservice.commodityBLservice包中的各个接口以提供服务，并依赖预设的一些PO与dataservice.commoditydataservice包中的数据库服务接口，完成处理库存数据的功能。



**commodityBL模块各个类的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| StorageIn | 处理入库请求的对象，拥有入库单的信息，能够通过向数据库发出请求完成数据添加任务 |
| StorageOut | 处理出库请求的对象，拥有出库单信息，能够通过向数据库发出请求完成登记出库单的信息的任务 |
| StorageInquiryPart | 处理库存查询的对象，能够向数据库请求特定限制的库存信息，完成库存查看的任务 |
| StorageInquiryAll | 处理库存盘点的对象，能够像数据库请求全部库存信息，完成库存盘点的任务 |

（3）模块内部类的接口规范

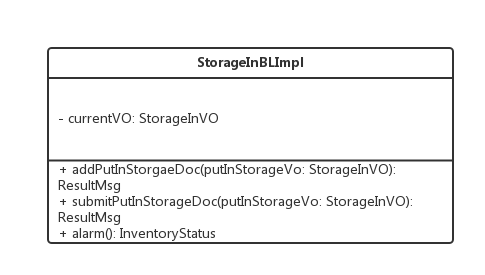
**commoditybl模块的接口规范**

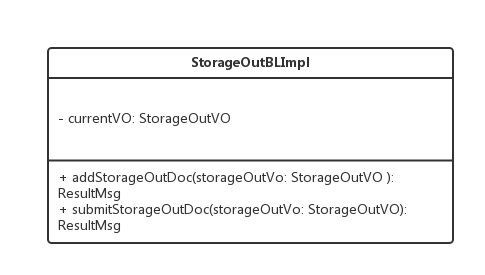
|  |  |  |  |
| --- | --- | --- | --- |
| 接口类 | 方法 | 类型 | 详细信息 |
| StorageInBLService | addPutInStorgaeDoc | 语法 | public ResultMsg addPutInStorgaeDoc(PutInStorageVO putInStorageVo); |
| 前置条件 | 库存管理人员输入入库单信息，选择提交 |
| 后置条件 | 系统检查单据信息格式，反馈检查结果 |
| submitPutInStorageDoc | 语法 | public ResultMsg submitPutInStorageDoc(PutInStorageVO 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 |

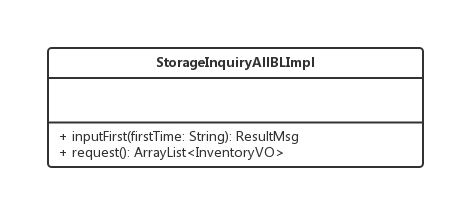
**commodityblservice需要的接口**

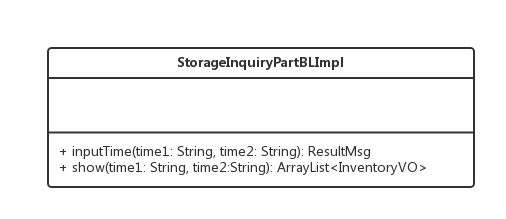
|  |  |  |
| --- | --- | --- |
| 接口名 | 服务名 | 详细信息 |
| InventoryDataService | public InventoryPO findByTime(String sTime,String eTime,String staffID) throws RemoteException; | 按起止时间进行查找返回相应的InventoryPO结果 |
| public InventoryPO findAll(String staffID) throws RemoteException; | 返回截止上次库存盘点所有的库存PO |
| public 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 |

Commodity模块的关键类图



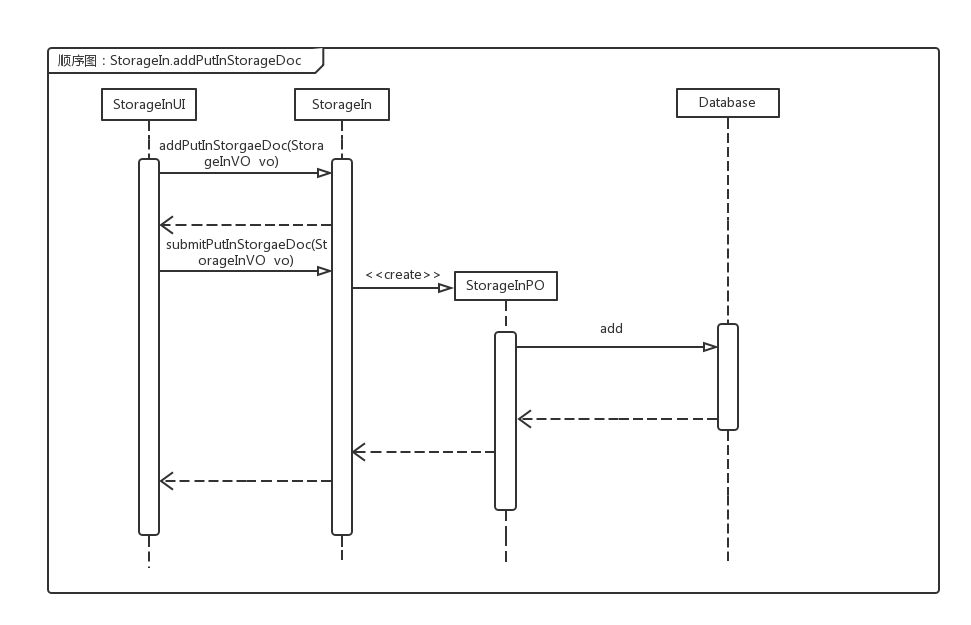




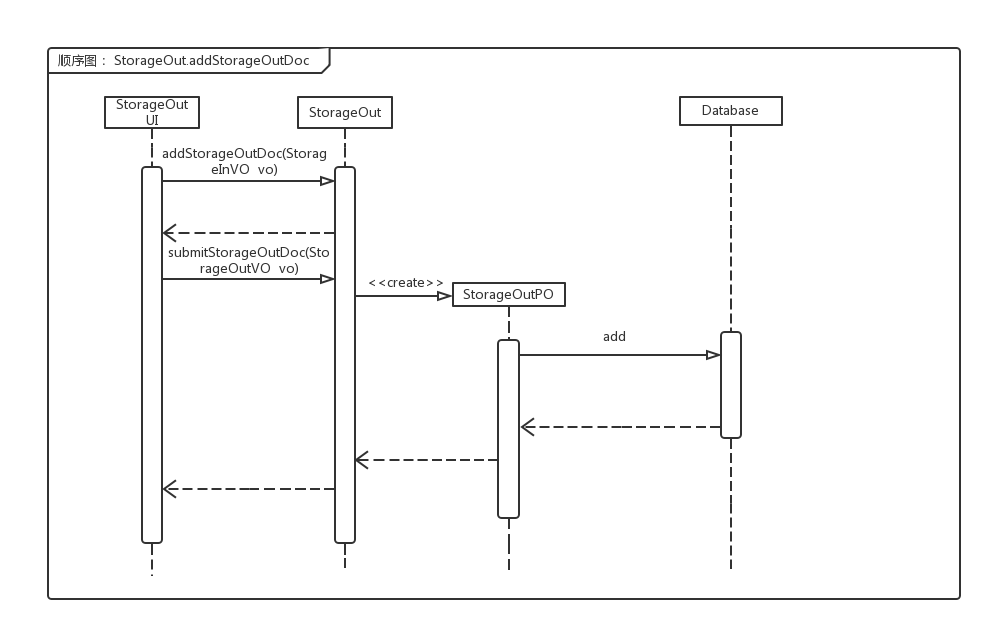


（4）业务逻辑层的动态模型

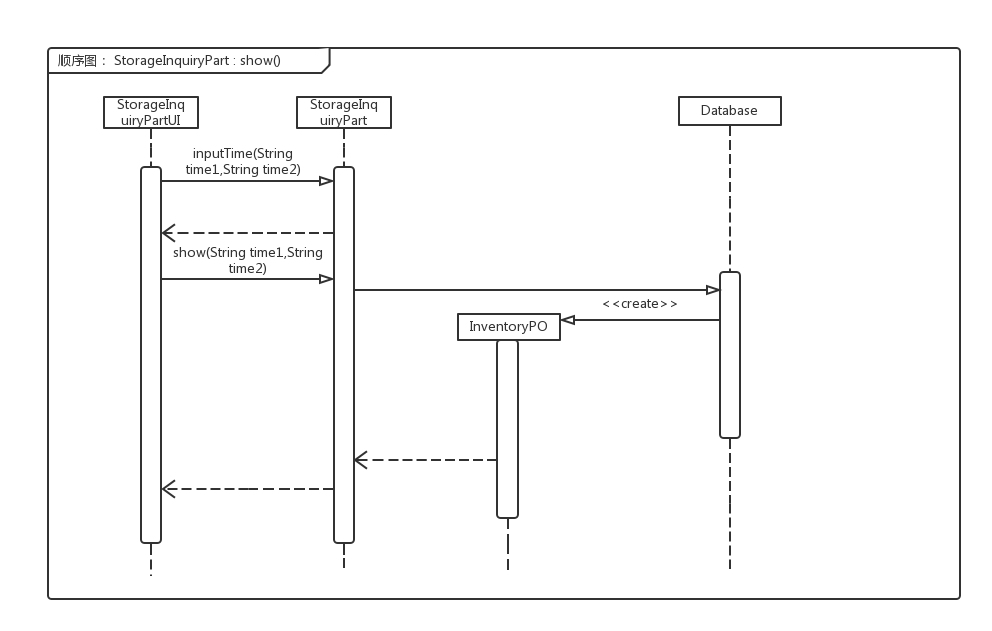
下图为入库处理StorageIn中相关对象之间的协作（数据层省略）。



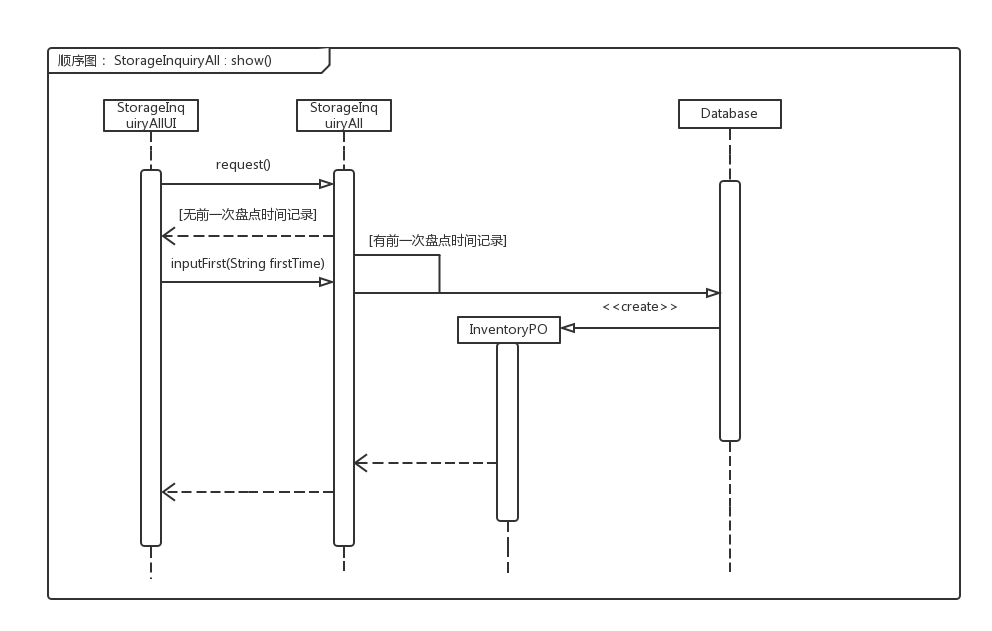
下图为出库处理StorageOut中相关对象之间的协作（数据层省略）。



下图为库存查询StorageInquiryPart中相关对象之间的协作（数据层省略）。



下图为库存盘点StorageInquiryAll中相关对象之间的协作（数据层省略）。



由于commodityBL模块类协作简单、PO较少、交互复杂度极低，故本文档不再给出状态图，状态转移可参见体系结构设计文档。

（5）业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的控制器委托给不同的领域对象。

#### 4.1.3 financeBL模块

（1）模块概述

financebl模块负责实现记录快递收款单、新增付款记录和结算管理的功能。

financebl模块的需求可参见需求规格说明文档的功能需求与相关非功能性需求。

financebl模块的职责及接口参见软件体系结构设计文档内容。

（2）整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们添加了接口。比如展示层和业务逻辑层之间，我们添加bussinesslogicservice. financeblservice接口包，内部含有CreditNoteInputBLService等3个接口。业务逻辑层和数据层之间添加dataservice.financedataservice接口包，内部含有CreditNoteInputDataService等3个接口。CreditNotePO是作为快递收款记录的持久化对象被添加到设计模型中去的。

financebl模块的设计如图1所示

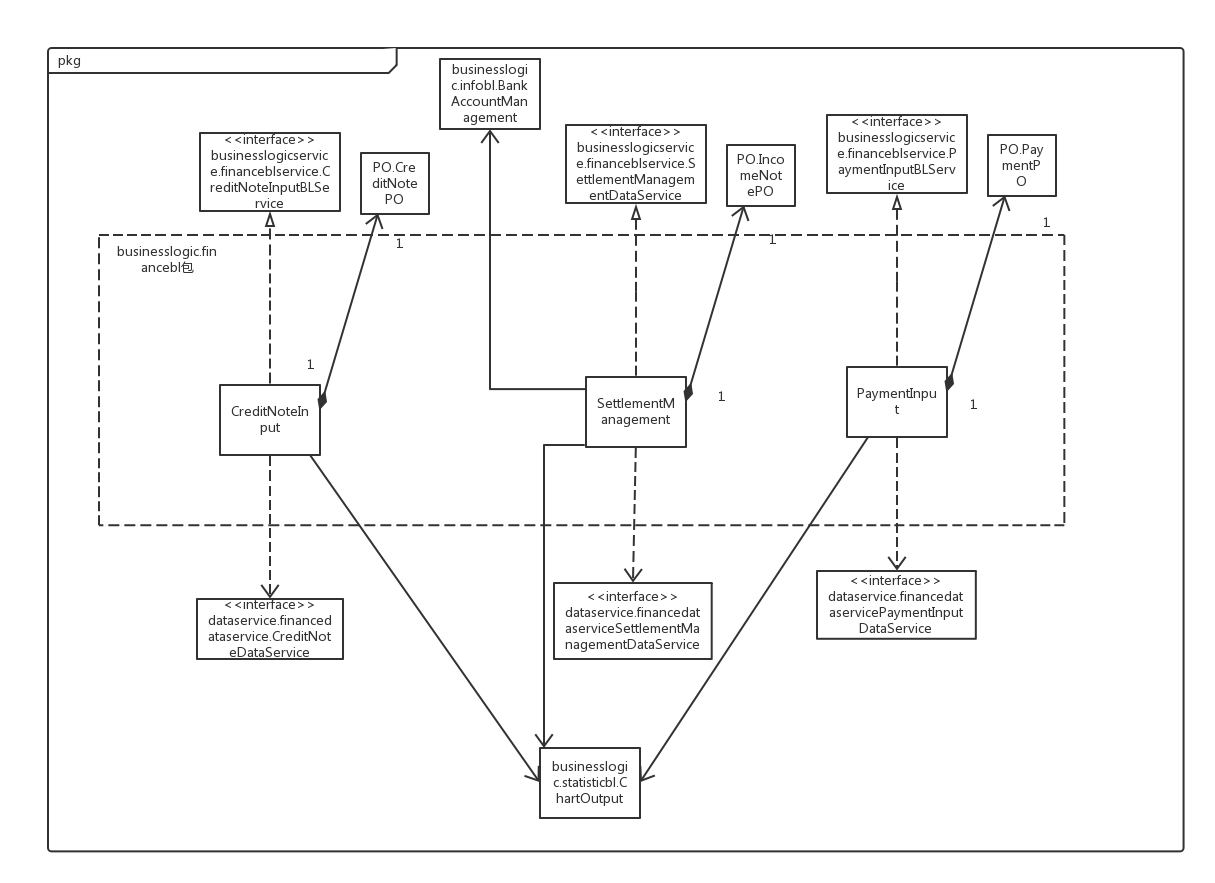


图1 financebl模块各个类的设计

financebl模块各个类的职责如表1所示

**表1**financebl**模块各个类的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| CreditNoteInput | 记录快递收款单的领域模型对象，拥有收款日期、收款金额、收款快递员等信息，可以帮助完成营业厅业务员记录快递收款单所需要的服务 |
| PaymentInput | 付款单的领域模型对象，拥有一个收款日期、收款金额、操作员等信息，可帮助完成创建付款单界面所需要的服务 |
| SettlementManagement | 结算管理的领域模型对象，拥有一个收款项等信息，可帮助完成创建结算管理界面所需要的服务 |

（3）模块内部的接口规范

finance模块的接口规范如表2所示

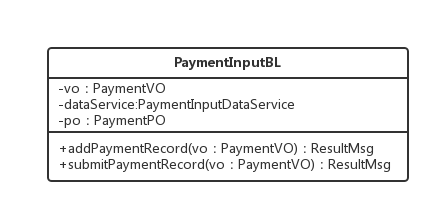
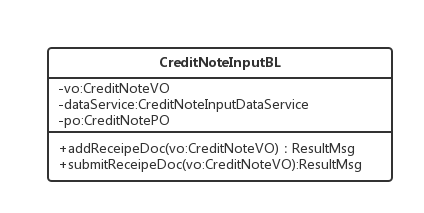
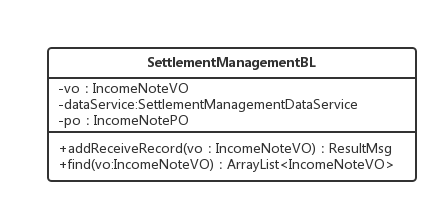
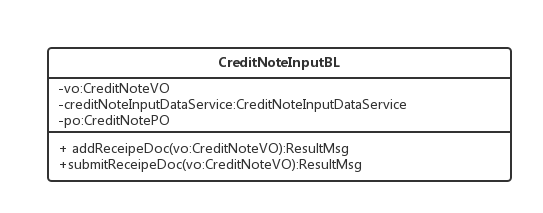
|  |  |  |  |
| --- | --- | --- | --- |
| CreditNoteInputBLService | addReceipeDoc | 语法 | public ResultMsg addReceipeDoc(CreditNoteVO vo); |
| 前置条件 | 财务人员选择新增收款单的功能 |
| 后置条件 | 系统检查收款单信息 |
| submitReceipeDoc | 语法 | public ResultMsg submitReceipeDoc(CreditNoteVO vo) |
| 前置条件 | 收款单格式检查无误 |
| 后置条件 | 系统提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 |

**表2 finance模块的接口规范**

**finance模块的关键类图**



（4）业务逻辑层的动态模型

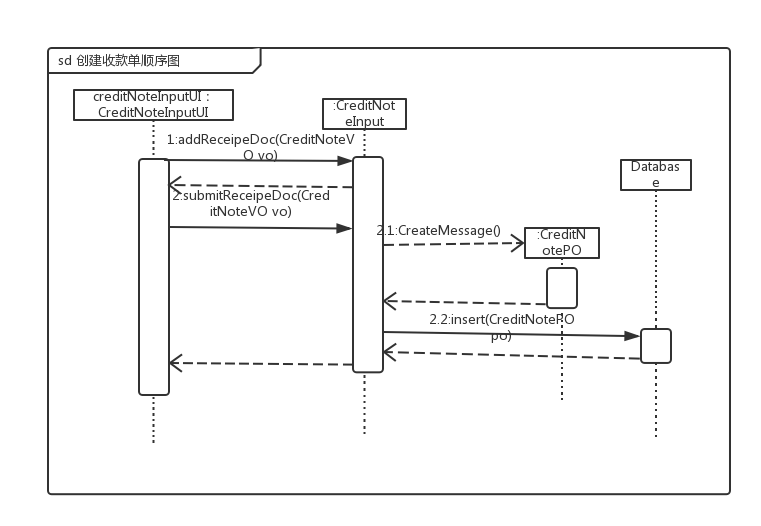
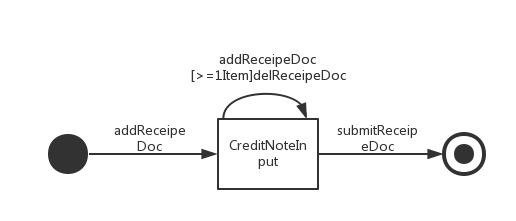
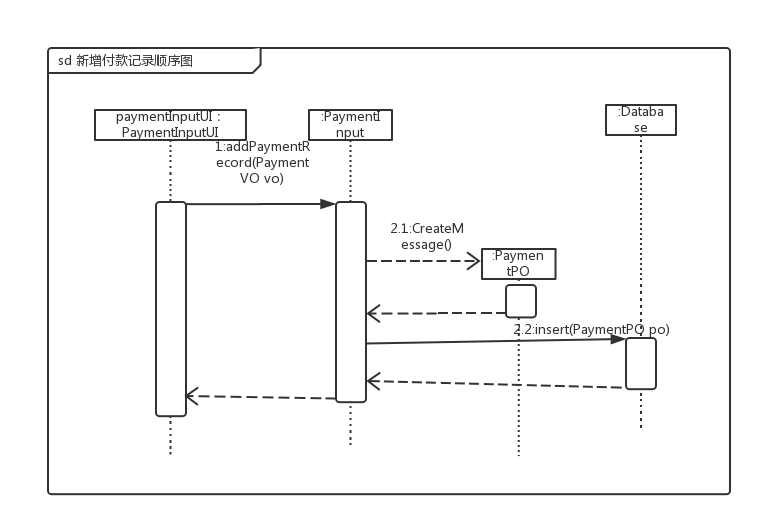
图2表明了物流管理系统中，当营业厅业务员输入快递收款单，记录快递收款单业务逻辑的相关对象之间的协作。

图2 记录快递收款单的顺序图

如下图2所示，CreditNoteInput对象状态图描述了CreditNoteInput对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着addReceipeDoc方法被 UI调用，CreditNoteInput进入CreditNoteInput状态；随着submit方法被调用，CreditNoteInput进入end状态。



下图表明了物流管理系统中，当财务人员输入与提交付款单，记录付款单业务逻辑的相关对象之间的协作。

下图 提交付款单的顺序图

如下图2所示，paymentInput对象状态图描述了paymentInput对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着addPaymentRecord方法被 UI调用，paymentInput进入paymentInput状态；随着submit方法被调用，paymentInput进入end状态。

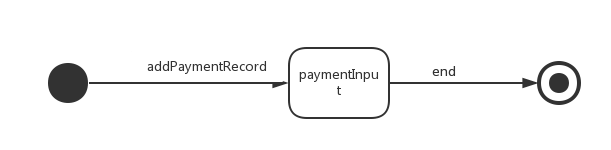


图4表明了物流管理系统中，当财务人员进行结算管理记录新收款项，结算管理单业务逻辑的相关对象之间的协作。

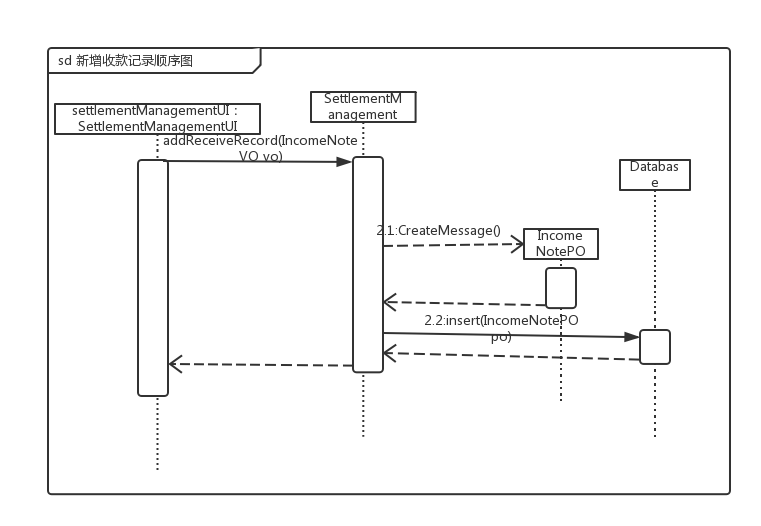


图4记录新收款单的顺序图

如下图2所示，SettlementManagement对象状态图描述了SettlementManagement对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着addReceiveRecord方法被 UI调用，SettlementManagement进入SettlementManagement状态；随着update方法被调用，进入BankAccountManagement状态。

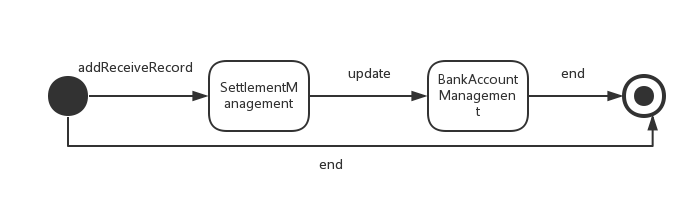
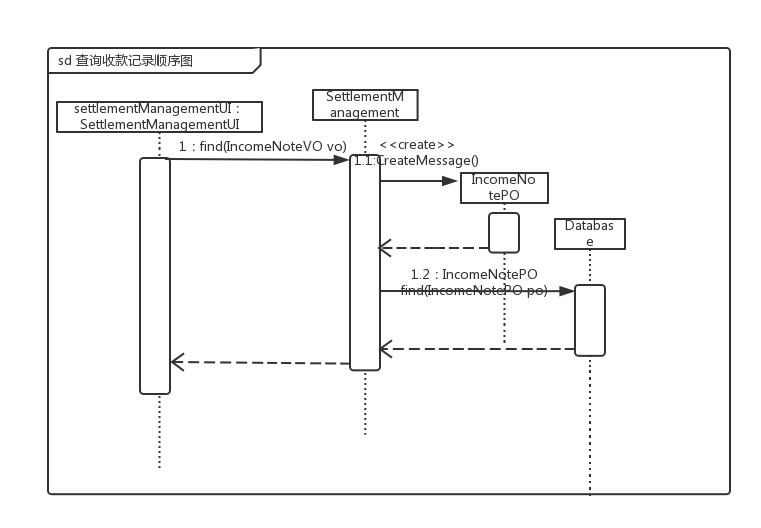
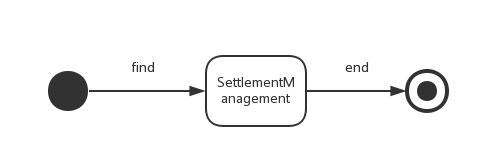
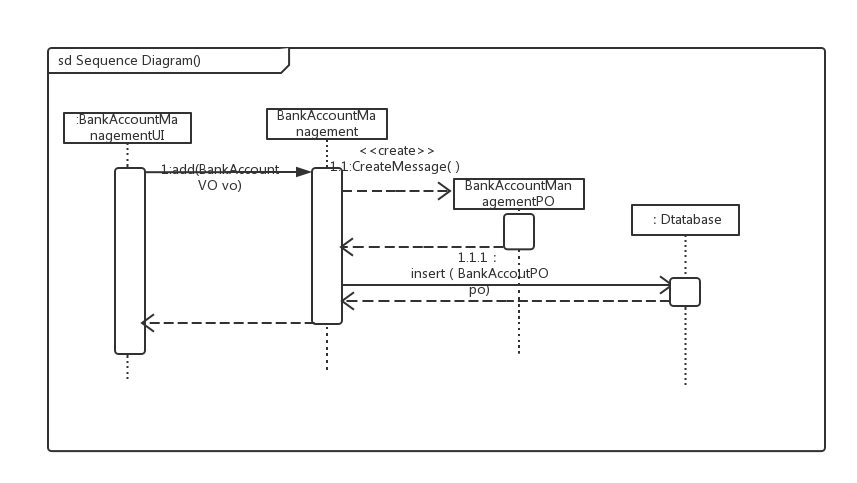


图5表明了物流管理系统中，当财务人员进行结算管理查询收款项，结算管理单业务逻辑的相关对象之间的协作。

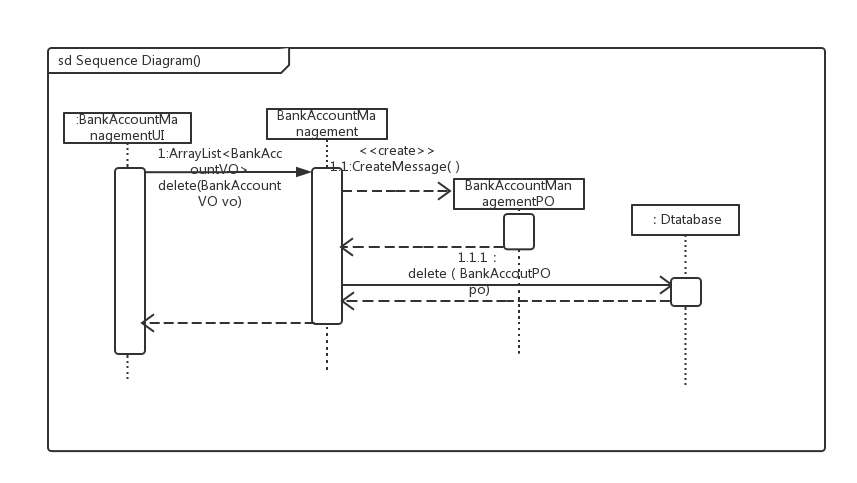
图5查询收款项的顺序图

如下图2所示，settlementManagement对象状态图描述了SettlementManagement对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着find方法被 UI调用，SettlementManagement进入SettlementManagement状态. 

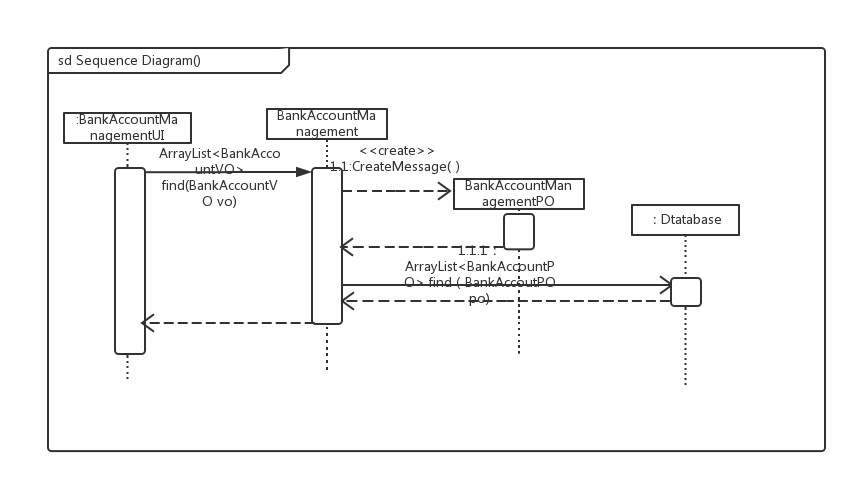
下图表明了物流管理系统中，当财务人员增加银行帐户时，管理银行账户业务逻辑的相关对象之间的协作。



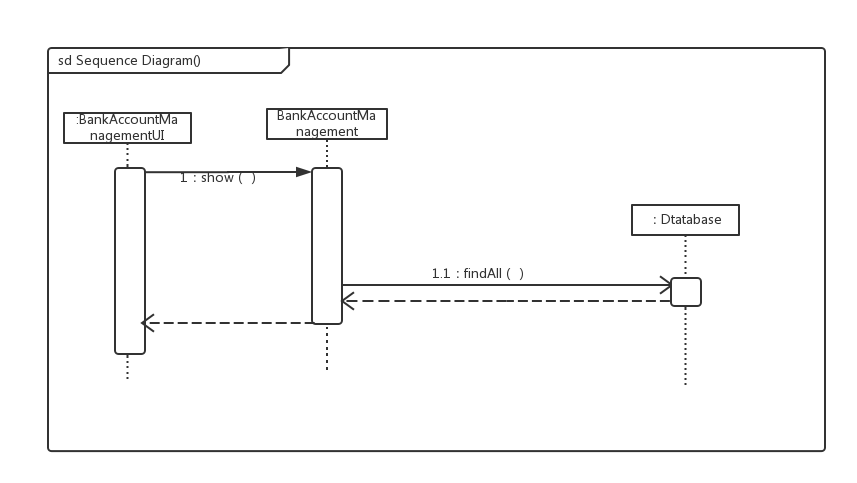
下图表明了物流管理系统中，当财务人员删除银行帐户时，管理银行账户业务逻辑的相关对象之间的协作。



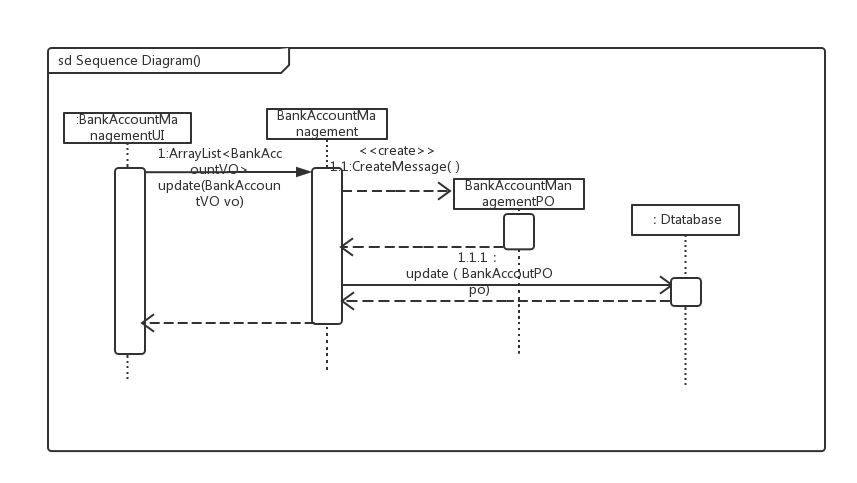
下图表明了物流管理系统中，当财务人员查找银行帐户时，管理银行账户业务逻辑的相关对象之间的协作。



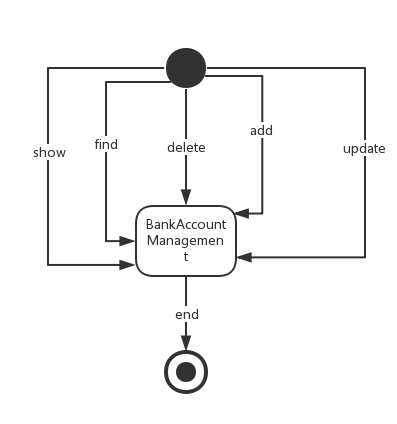
下图表明了物流管理系统中，当财务人员查看所有银行帐户时，管理银行账户业务逻辑的相关对象之间的协作。



下图表明了物流管理系统中，当财务人员修改银行帐户时，管理银行账户业务逻辑的相关对象之间的协作。



如下图2所示，BankAccountManagement对象状态图描述了BankAccountManagement对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着add或show或delete或update或find方法被 UI调用，BankAccountManagement进入BankAccountManagement状态



#### 4.1.4 statisticBL模块

（1）模块概述

statisticbl模块负责实现对业务数据的存储查询更新。

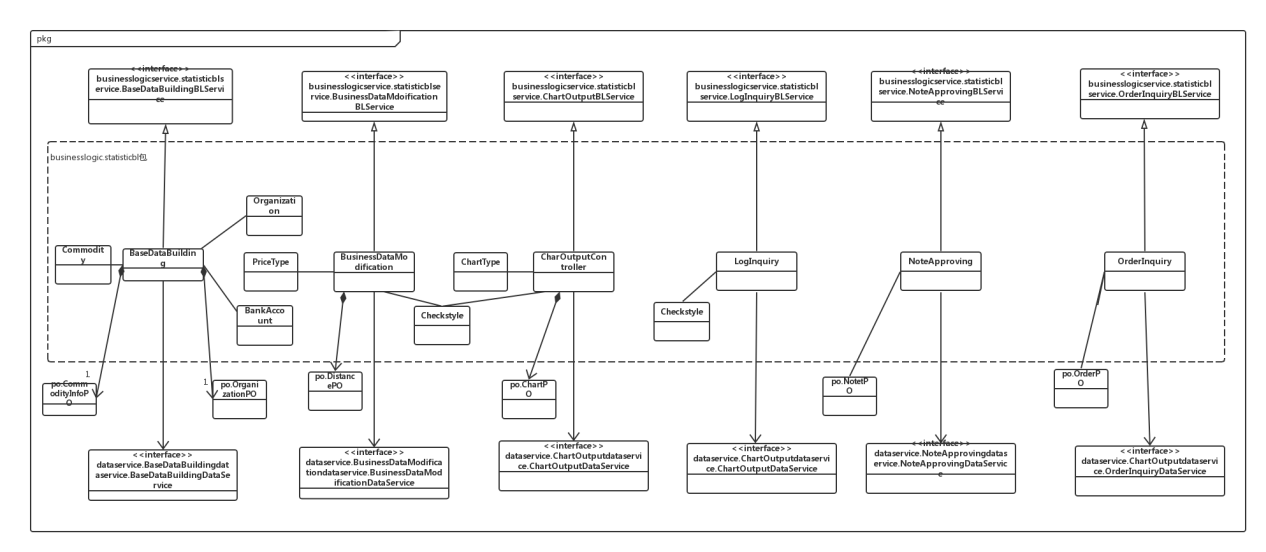
statisticbl模块的需求可参见需求规格说明文档的功能需求与相关非功能性需求。

statisticbl模块的职责及接口参见软件体系结构设计文档内容。

（2）整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们添加了接口。比如展示层和业务逻辑层之间，我们添加bussinesslogicservice.statisticblservice接口包，内部含有BaseDataBuildingBLService等7个接口。业务逻辑层和数据层之间添加dataservice.statisticdataservice接口包，内部含有BaseDataBuildingDataService等7个接口。ChartPO等数据信息的po对象是作为单据类别的持久化对象被添加到设计模型中去的。

Statisticbl模块的设计图如图所示。



statistic模块各个类的职责如表所示

|  |  |
| --- | --- |
| 模块 | 职责 |
| BaseDataBuilding | 期初建账的领域模型对象，拥有期初建账机构、库存、账户的信息，可以帮助完成财务人员期初建账所需要的服务 |
| BusinessDataModification | 修改业务数据的领域模型对象，拥有基础业务数据的信息，可以帮助完成总经理修改业务数据所需要的服务 |
| ChartOutput | 获取统计报表的领域模型对象，拥有统计报表包含经营情况表、成本收益表的信息，可以帮助完成财务人员或总经理获取统计报表所需要的服务 |
| LogInquiry | 查询系统日志的领域模型对象，拥有系统日志的信息，可以帮助完成总经理和财务人员查询系统日志所需要的服务 |
| NoteApproving | 审批单据的领域模型对象，拥有单据审批结果和各种单据的信息，可以帮助完成总经理审批单据所需要的服务 |
| OrderInquiry | 查询订单的领域模型对象，拥有订单物流信息的信息，可以帮助完成客户和快递员查询订单的服务 |

（3）模块内部的接口规范

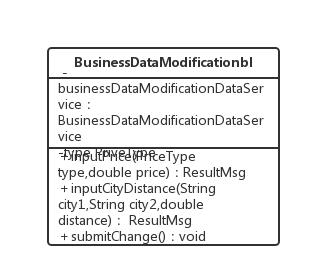
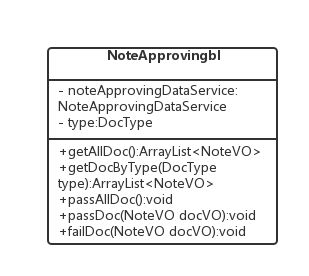
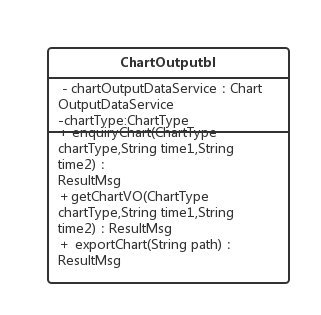
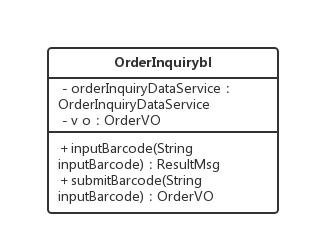
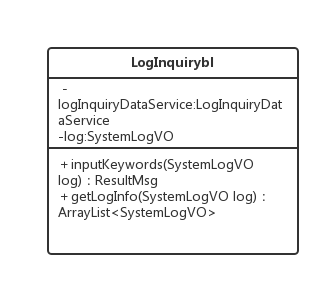
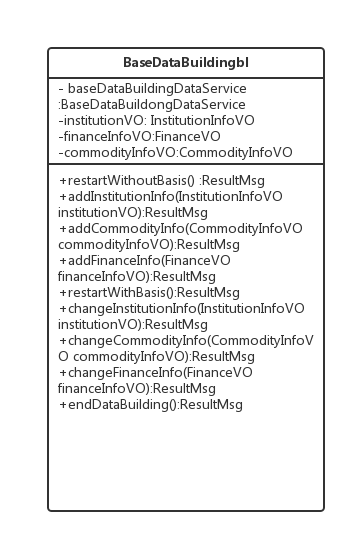
statisticbl模块的接口规范如表2所示

|  |  |  |  |
| --- | --- | --- | --- |
| **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需要的服务（需接口）

|  |  |  |
| --- | --- | --- |
| 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 |

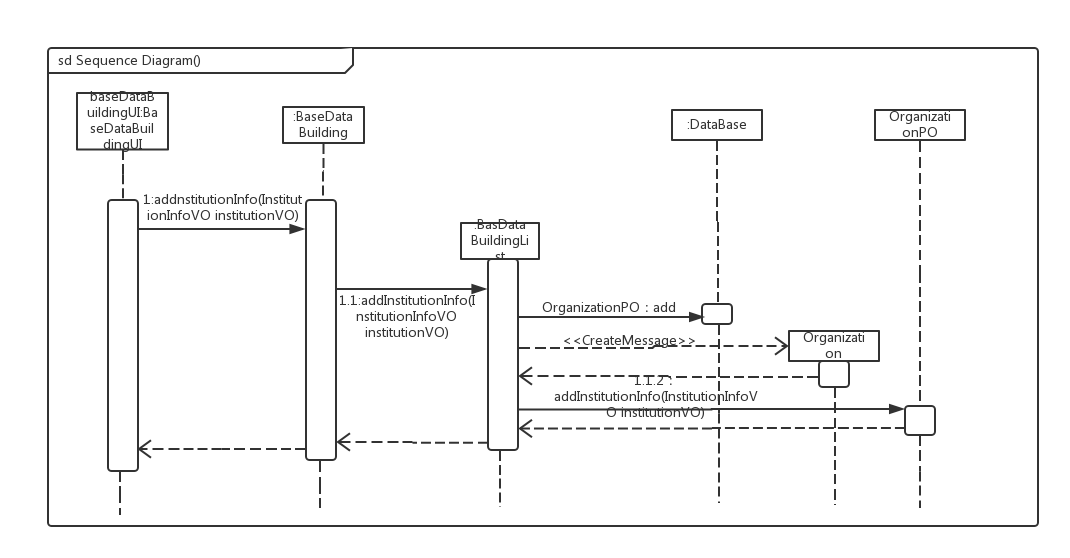
**Statistic模块的关键类图**

****

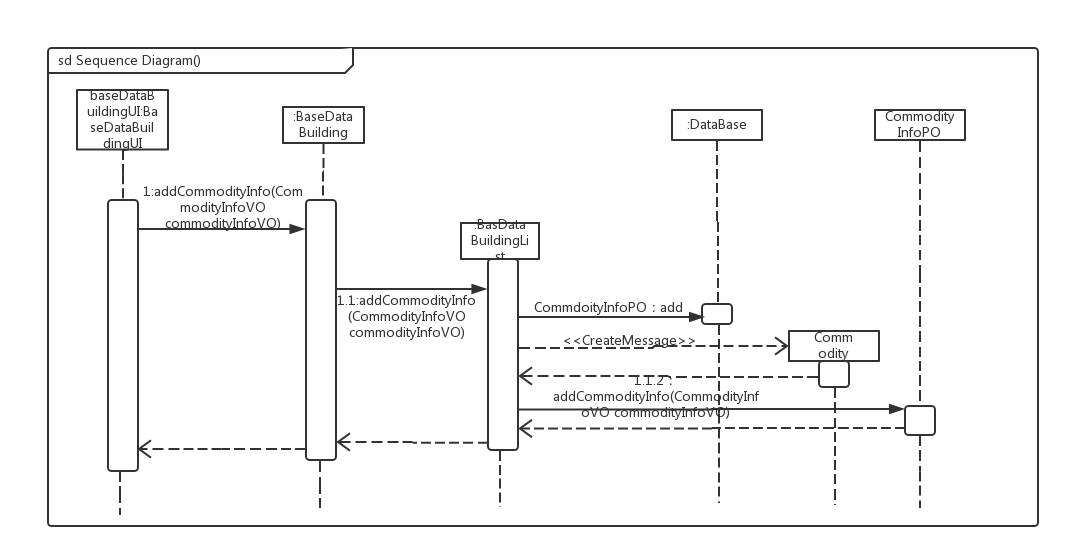
（4）业务逻辑层的动态模型

下列图图表明了物流管理系统中，当财务人员进行期初建账开始新一财年时，期初建账逻辑的相关对象之间的协作

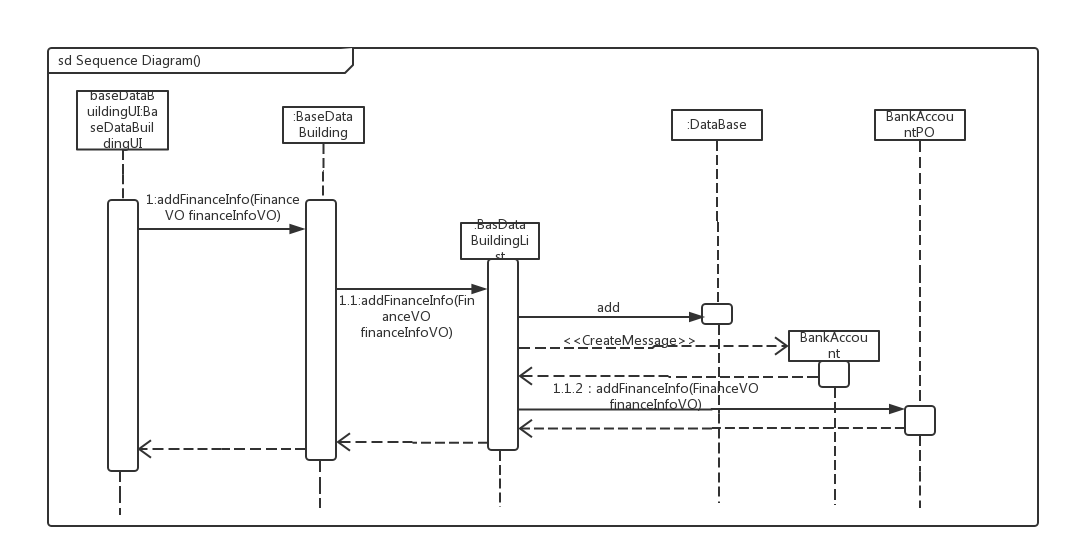
下图为无基础建账输入机构信息的顺序图



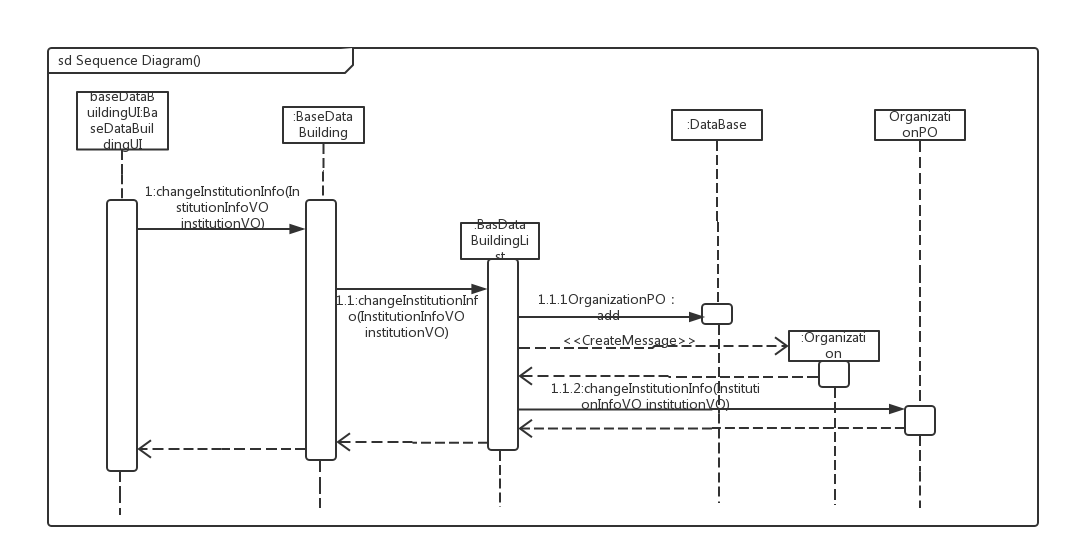
下图为无基础建账增加库存信息的顺序图



下图为无基础建账增加银行账户的顺序图



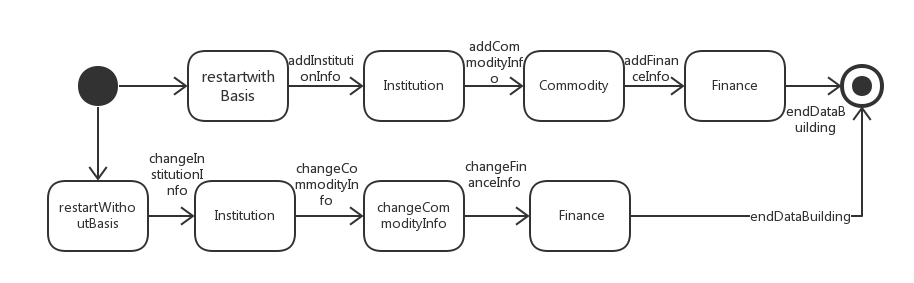
下图为有基础建账修改机构信息顺序图（（图要改）



下图为有基础建账修改库存信息的顺序图（图要改）

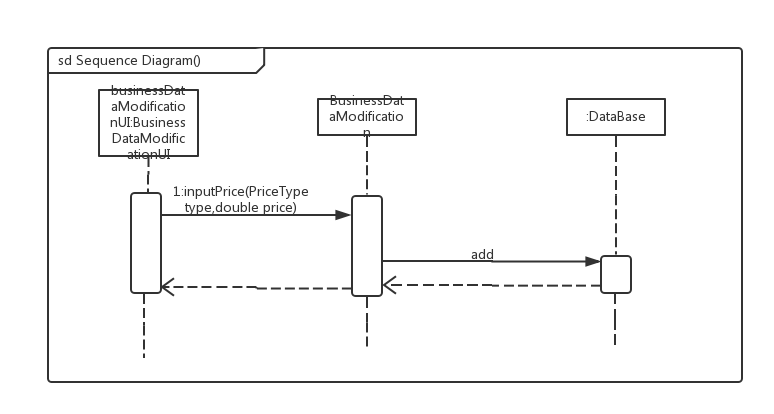
下图为有基础建账修改银行账户信息的顺序图（图要改）

下图所示的状态图描述了BaseDataBuilding对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着restartWithoutBasis的方法被UI调用，BaseDataBuilding对象被创建，对象进入无基础建账状态，UI依次调用addInstitutionInfo、addCommdityInfo、addFinanceInfo方法，对象依次进入增加机构信息状态、增加库存信息状态、增加银行账户状态，增加完毕通过格式检查，向数据层添加单据po持久对象，业务逻辑对象被回收。随着restartWithBasis的方法被UI调用，BaseDataBuilding对象被创建，对象进入有基础建账状态，UI依次调用changeInstitutionInfo、changeCommdityInfo、changeFinanceInfo方法，对象依次进入修改机构信息状态、修改库存信息状态、修改银行账户状态，完毕完毕通过格式检查，向数据层添加单据po持久对象，业务逻辑对象被回收。

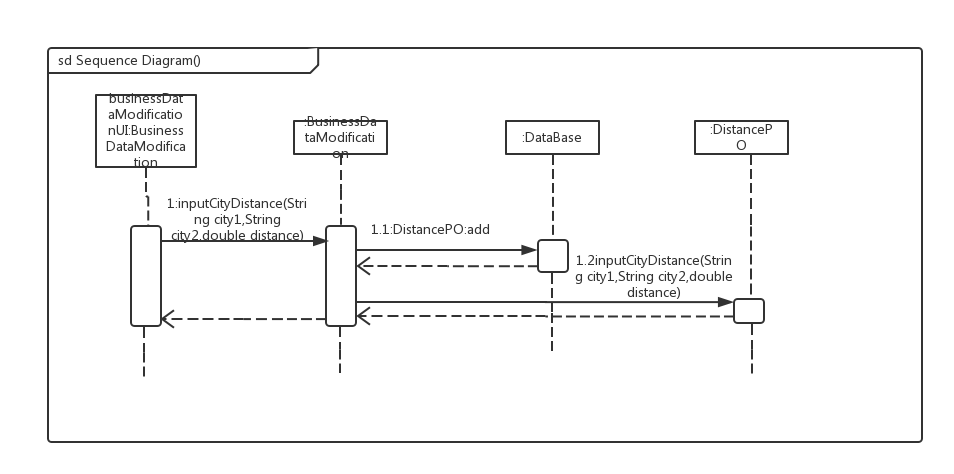


下列图图表明了物流管理系统中，当总经理修改基础业务数据时，修改基础业务数据的相关对象之间的协作

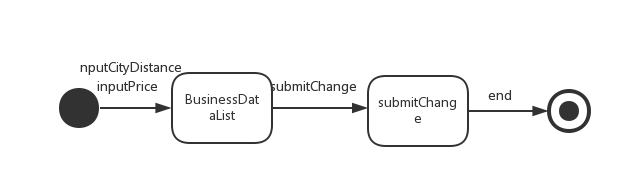
下图为修改价格的顺序图



下图为修改城市距离数据的顺序图

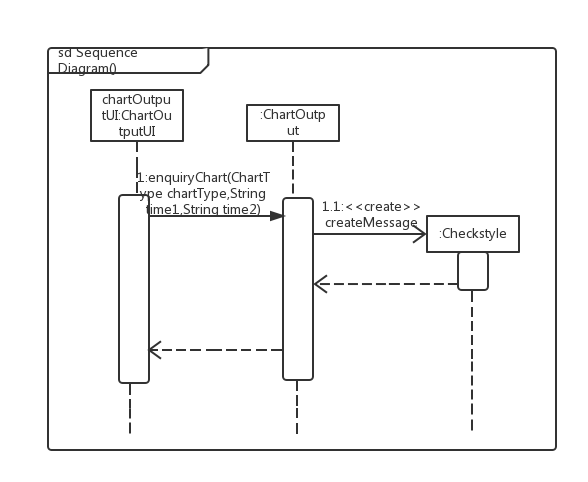


下图所示的状态图描述了BusinessDataModification对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inputPrice的方法被UI调用，BaseDataBuilding对象被创建，对象进入修改价格状态，随着inputCityDistance被UI调用，对象进入修改城市间距离的状态，随着submitChange方法被UI调用，进入格式检查状态，通过格式检查后价格数据添加到数据库，向数据层添加城市距离po持久化对象，业务逻辑对象被回收。

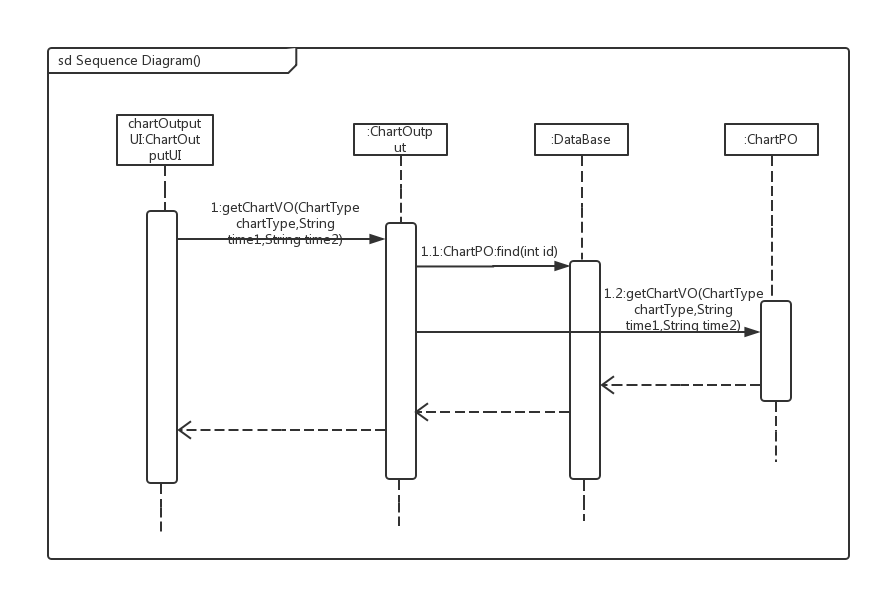


下列图图表明了物流管理系统中，当总经理或财务人员获取统计报表时，获取统计报表的相关对象之间的协作

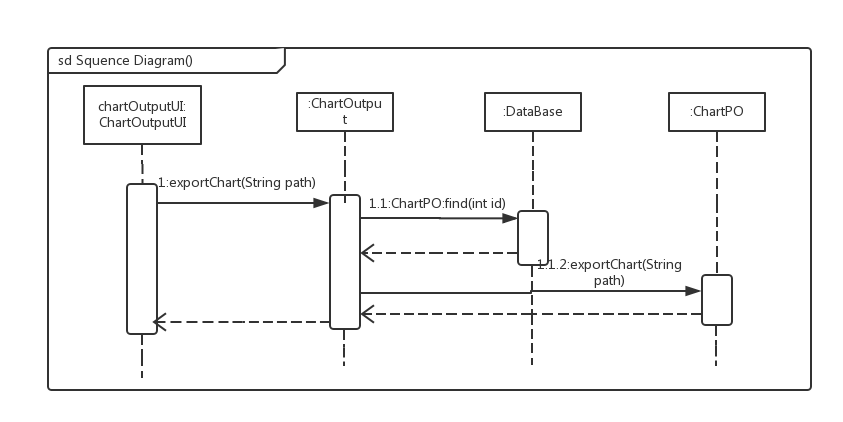
下图为输入查询报表关键字信息的顺序图



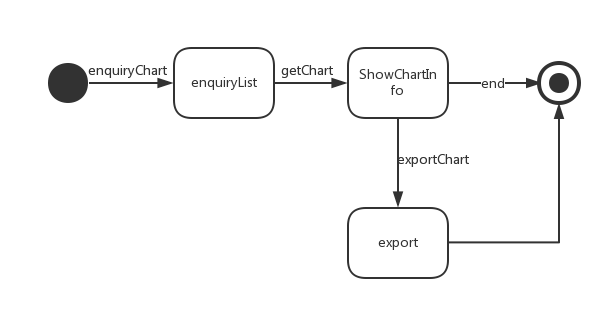
下图为获得查询报表信息的顺序图



下图为导出查询到的报表到本地的顺序图

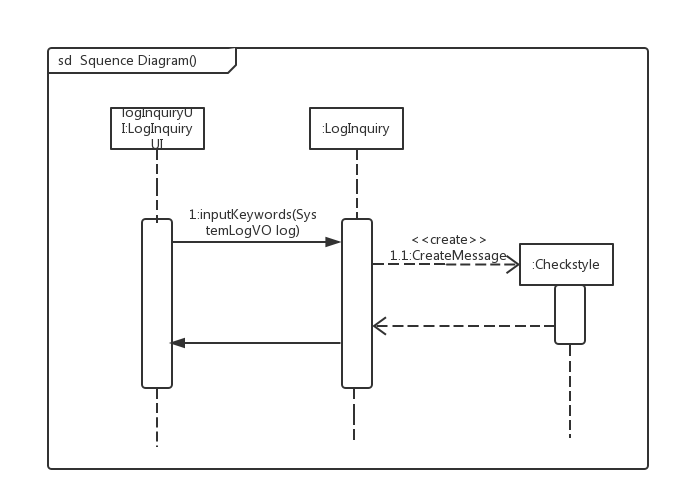


下图所示的状态图描述了ChartOutput对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着enquiryChart的方法被UI调用，ChartOutput对象被创建，对象进入获取关键字状态，检查关键字格式，然后随着getChart方法被UI调用，进入获取符合要求的统计报表的状态，业务逻辑层获取统计报表，展示层获取报表的vo，向用户显示报表信息，业务逻辑对象被回收。

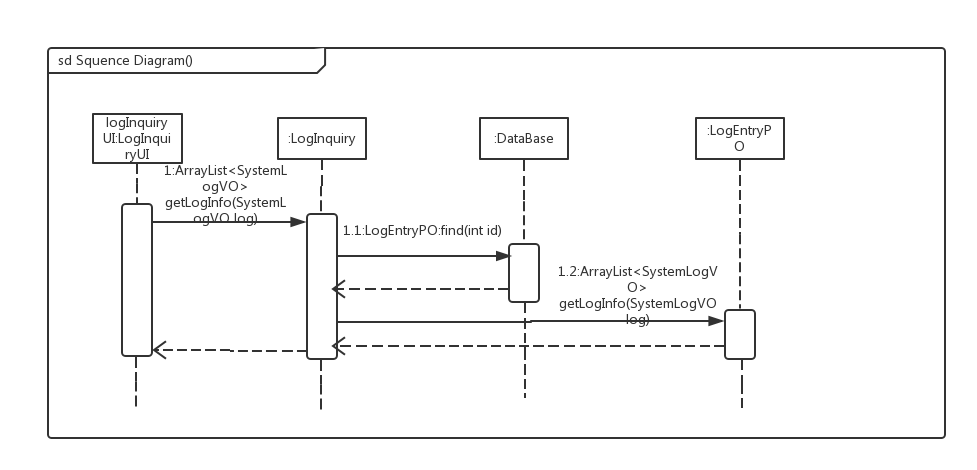


下列图图表明了物流管理系统中，当总经理或财务人员查询系统日志时，查询系统日志的相关对象之间的协作

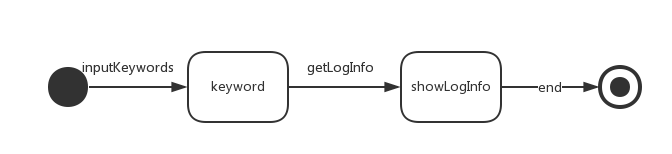
下图为输入查询关键字的顺序图



下图为显示对应关键字的系统日志的顺序图



下图所示的状态图描述了LogInquiry对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inputKeywords的方法被UI调用，LogInquiry对象被创建，LogInquiry对象进入获取关键字状态，检查关键字格式，然后随着getLogInfo方法被UI调用，进入获取符合要求的系统日志的状态，业务逻辑层生成po，数据层根据po信息，查找数据库，将找到的信息封装为po由数据层传如业务逻辑层，业务逻辑层将po转换成vo，传给展示层，向用户展示，业务逻辑对象被回收。

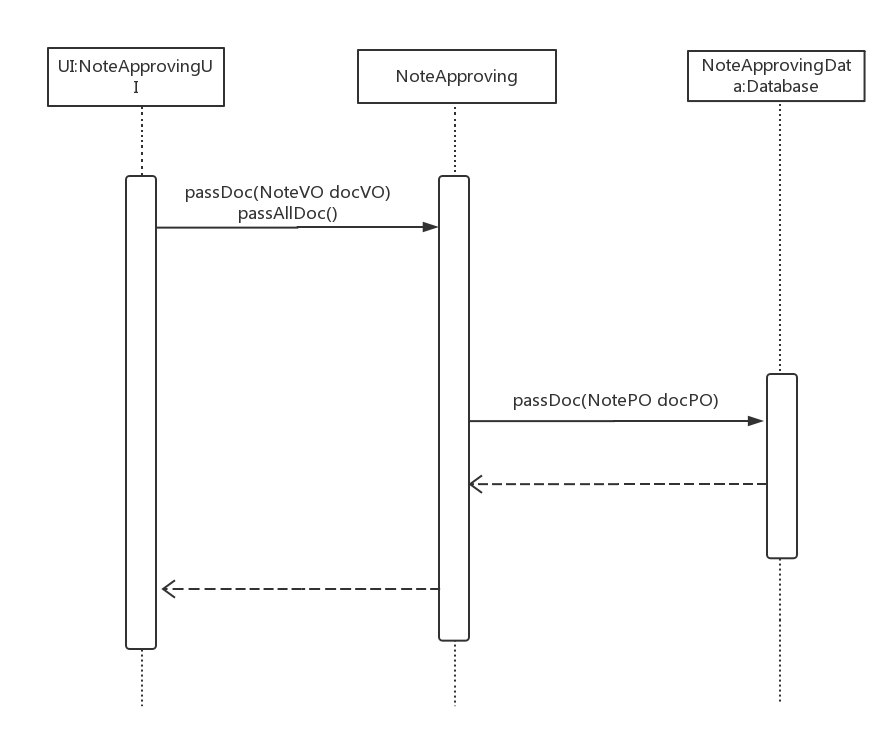


下列图图表明了物流管理系统中，当总经理或财务人员获取统计报表时，获取统计报表的相关对象之间的协作

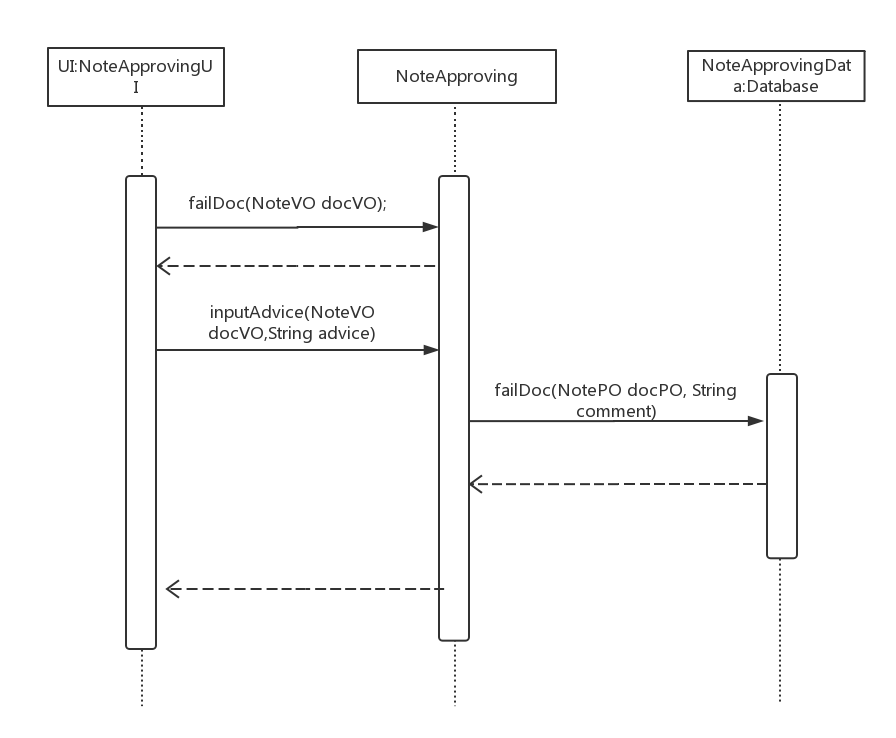
下图是获取单据信息的顺序图

（待插图）

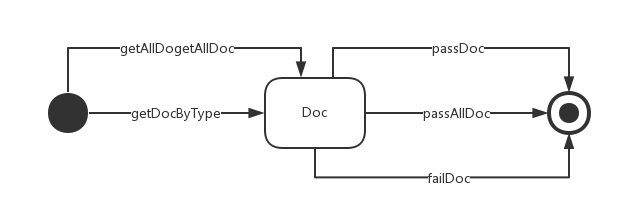
下图是通过单据审批的顺序图（要改）



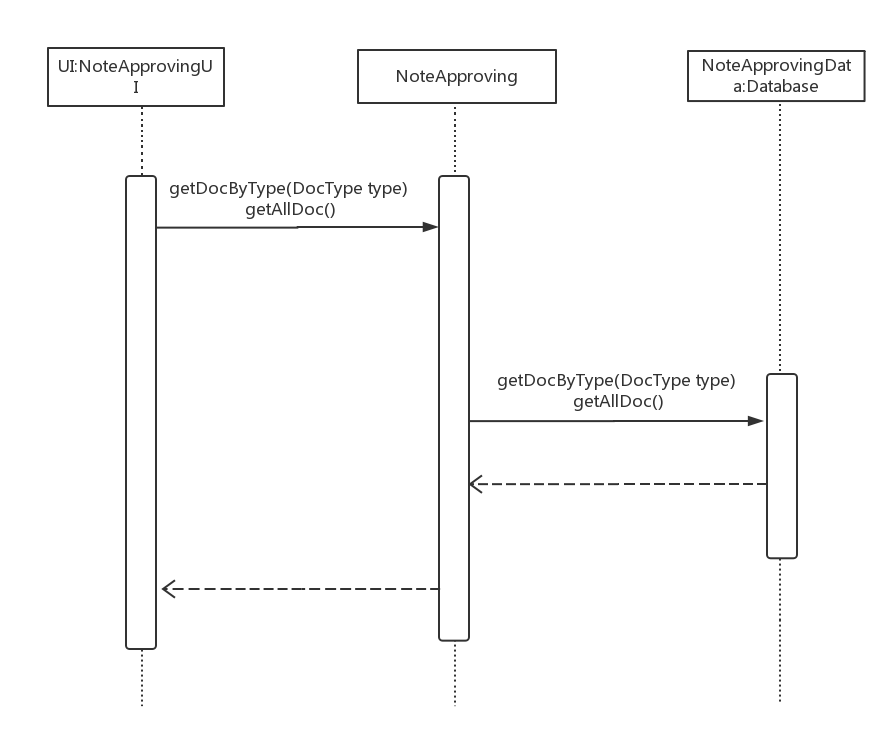
下图是否决单据的顺序图



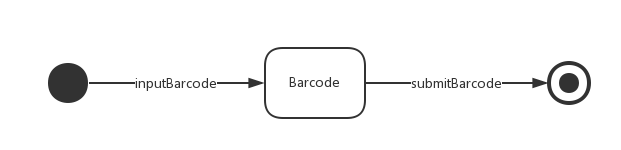
下图所示的状态图描述了NoteApproving对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着getAllDoc或getDocByType的方法被UI调用，NoteApproving对象被创建，NoteApproving对象进入获取单据的状态，然后随着passDoc或passAllDoc()或failDoc方法被UI调用，进入单据审批状态的状态，根据审批结果返回信息，业务逻辑对象被回收。



下列图图表明了物流管理系统中，当客户或快递员查询订单信息时，查询订单信息的相关对象之间的协作



下图所示的状态图描述了OrderInquiry对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inputBarcode的方法被UI调用，OrderInquiry对象被创建，OrderInquiry对象获取条形码的状态，对条形码进行格式检查，然后随着submitBarcode方法被UI调用，进入查询状态的状态，根据查询结果返回信息，业务逻辑对象被回收。



#### 4.1.5 infobl模块

（1）模块概述

infobl模块负责实现管理人员机构等各种信息等功能。

infobl模块的需求可参见需求规格说明文档的功能需求与相关非功能性需求。

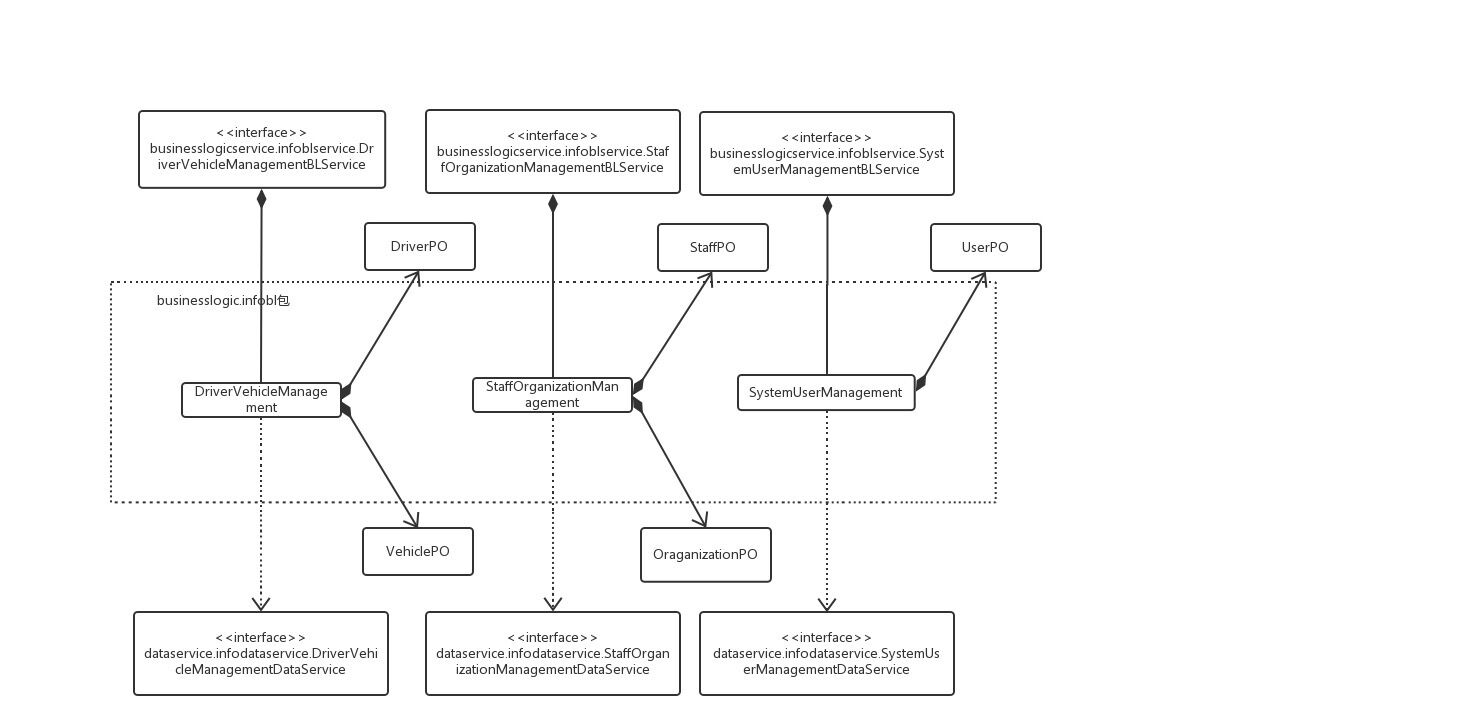
infobl模块的职责及接口参见软件体系结构设计文档内容。

（2）整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们添加了接口。比如展示层和业务逻辑层之间，我们添加bussinesslogicservice.infoblservice接口包，内部含有DriverVehicleManagementBLService等3个接口。业务逻辑层和数据层之间添加dataservice.infodataservice接口包，内部含有DriverVehicleManagementDataService等7个接口。DriverPO,VehiclePO等po对象是作为数据相应类别的持久化对象被添加到设计模型中去的。

infobl模块的设计如图1所示

**图1 infobl模块各个类的设计**



infobl模块各个类的职责如表1所示

**表1 infobl模块各个类的职责**

|  |  |
| --- | --- |
| **模块** | **职责** |
| DriverVehicleManagement | 管理车辆司机信息的领域模型对象，拥有车辆与司机信息，可以帮助完成管理车辆司机信息界面所需要的服务 |
| StaffOrganizationManagement | 管理人员机构信息的领域模型对象，拥有人员与机构信息，可以帮助完成管理人员机构信息界面所需要的服务 |
| SystemUserManagement | 管理系统用户的领域模型对象，拥有系统用户信息，可以帮助完成管理系统用户信息界面所需要的服务 |

（3）模块内部的接口规范

infobl模块的接口规范如表2所示

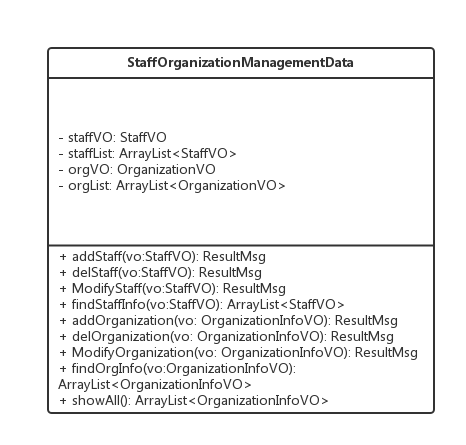
**表2** infobl**模块的接口规范**

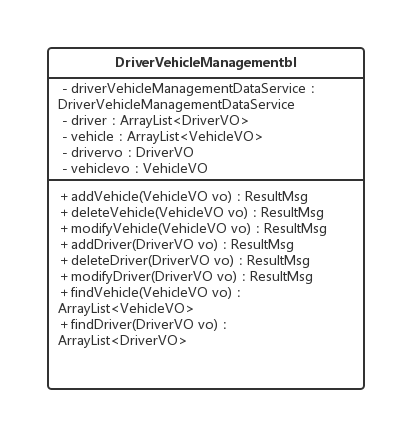
|  |  |  |  |
| --- | --- | --- | --- |
| 接口类 | 方法 | 类型 | 详细信息 |
| DriverVehicleManagementBLService | addVehicle | 语法 | public ResultMsg addVehicle(VehicleVOvo); |
| 前置条件 | 业务员选择新增车辆信息功能 |
| 后置条件 | 系统进行格式检查，返回格式检查结果，系统日志记录 |
| deleteVehicle | 语法 | public ResultMsg deleteVehicle(VehicleVOvo); |
| 前置条件 | 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 |

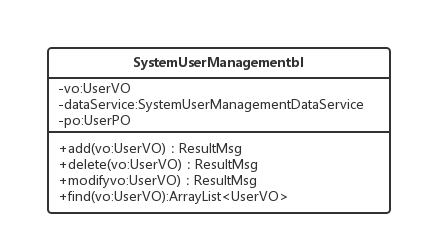
**infobl模块需要的服务**

|  |  |  |
| --- | --- | --- |
| **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 |

**Info模块的关键类图**

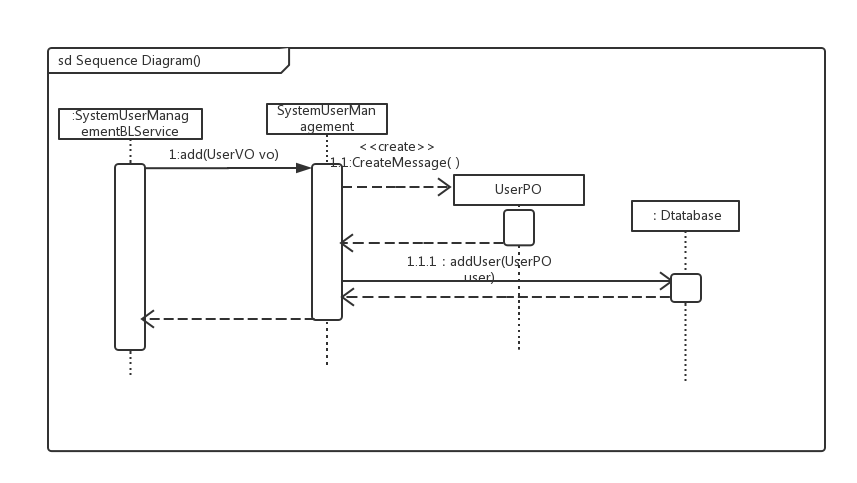
****

****

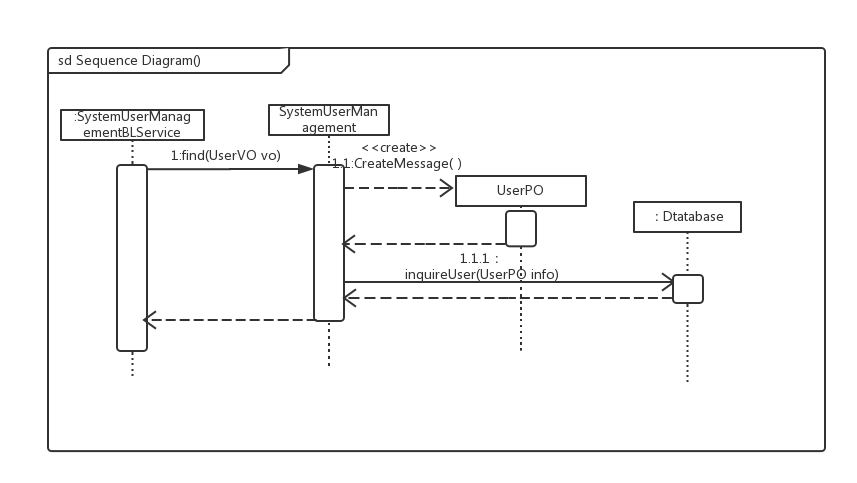
****

（4）业务逻辑层的动态模型

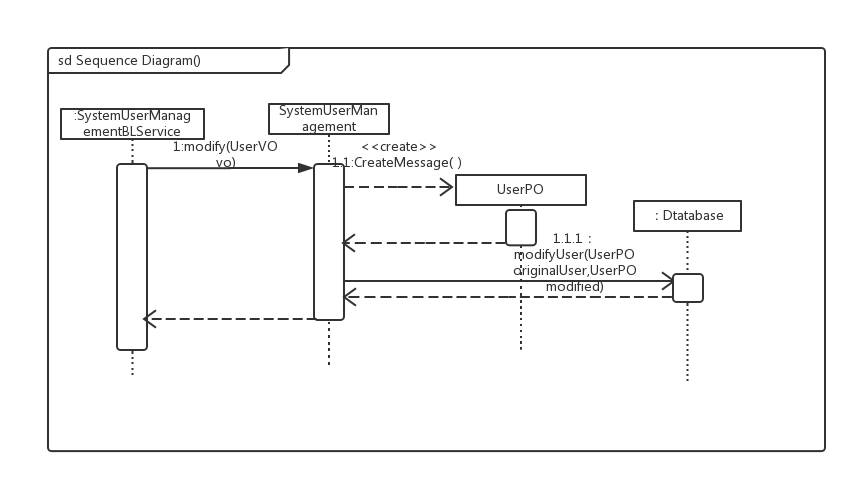
下图表明了物流管理系统中，当系统管理员增加用户帐户时，管理系统用户业务逻辑的相关对象之间的协作。



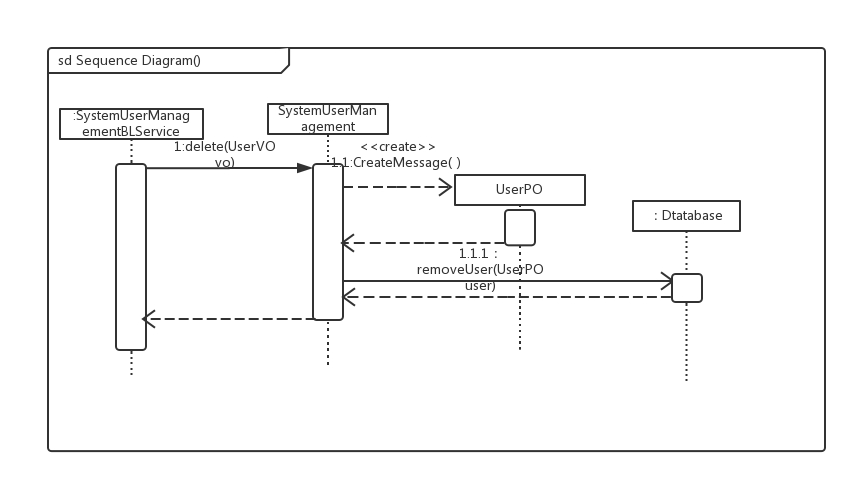
下图表明了物流管理系统中，当系统管理员查找用户帐户时，管理系统用户业务逻辑的相关对象之间的协作。



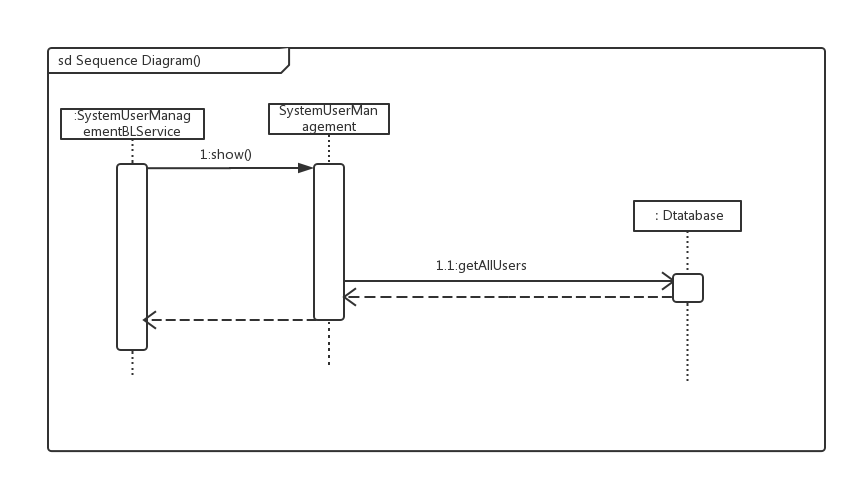
下图表明了物流管理系统中，当系统管理员修改用户帐户时，管理系统用户业务逻辑的相关对象之间的协作。



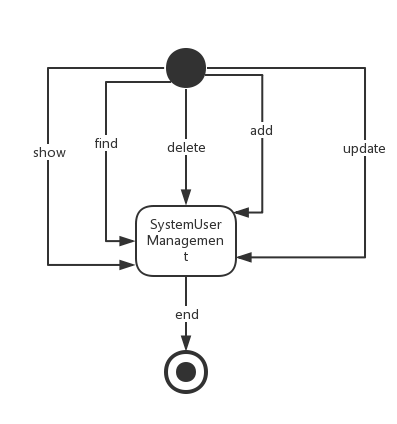
下图表明了物流管理系统中，当系统管理员删除用户帐户时，管理系统用户业务逻辑的相关对象之间的协作。



下图表明了物流管理系统中，当系统管理员查看所有用户帐户时，管理系统用户业务逻辑的相关对象之间的协作。

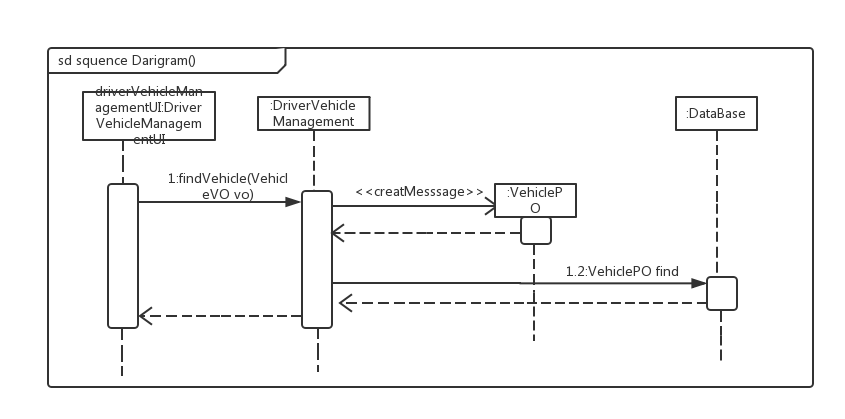


如下图2所示，SystemUserManagement对象状态图描述了SystemUserManagement对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着add或show或delete或update或find方法被 UI调用，SystemUserManagement进入SystemUserManagement状态

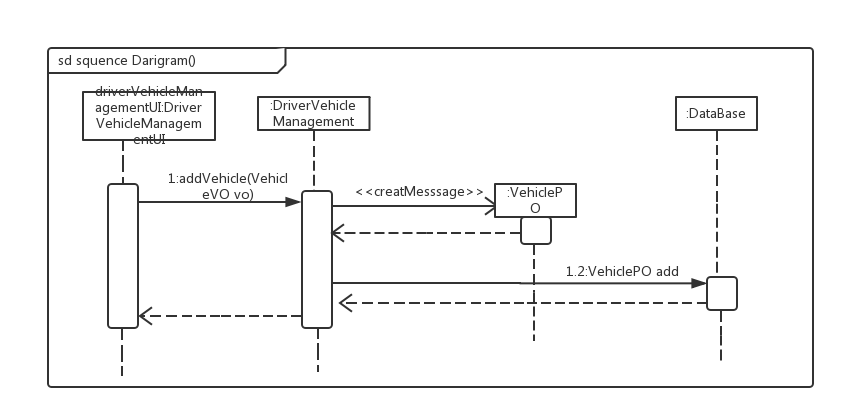


下列图图表明了物流管理系统中，当营业厅业务员进行司机与车辆管理时，管理车辆与司机信息的相关对象之间的协作

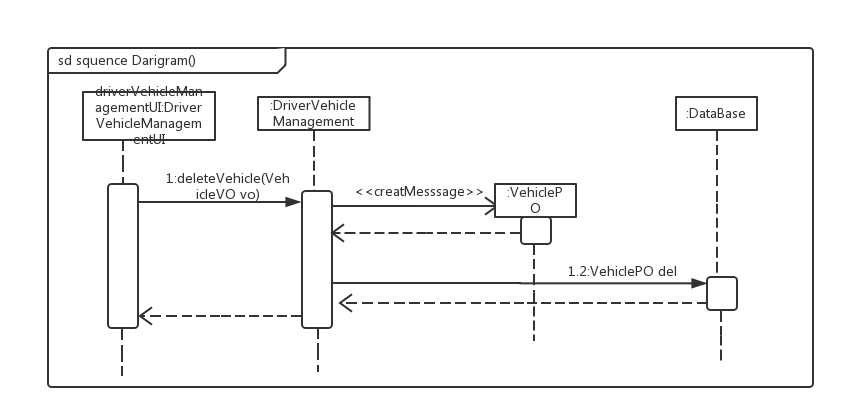
下图为查找车辆信息的顺序图



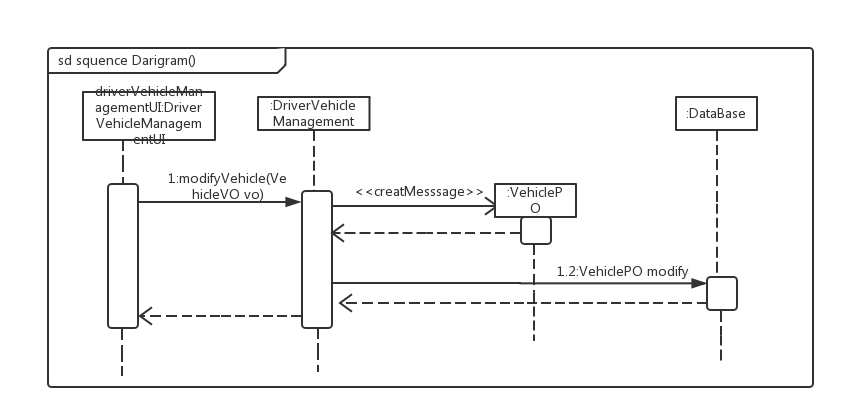
下图为增加车辆信息的顺序图



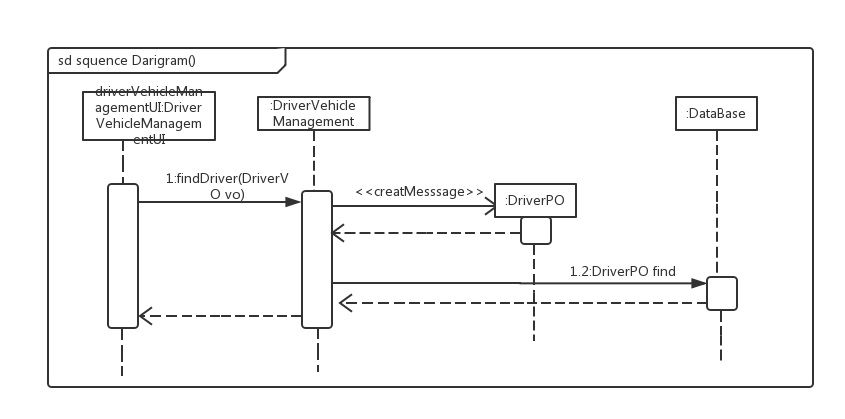
下图为删除车辆信息的顺序图



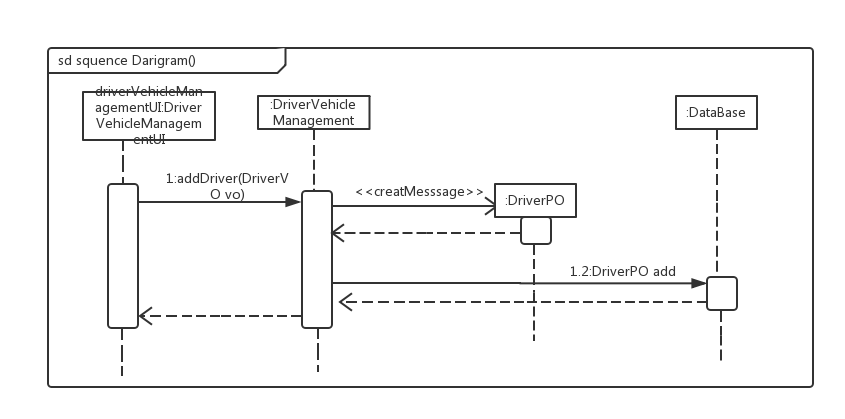
下图为修改车辆信息的顺序图



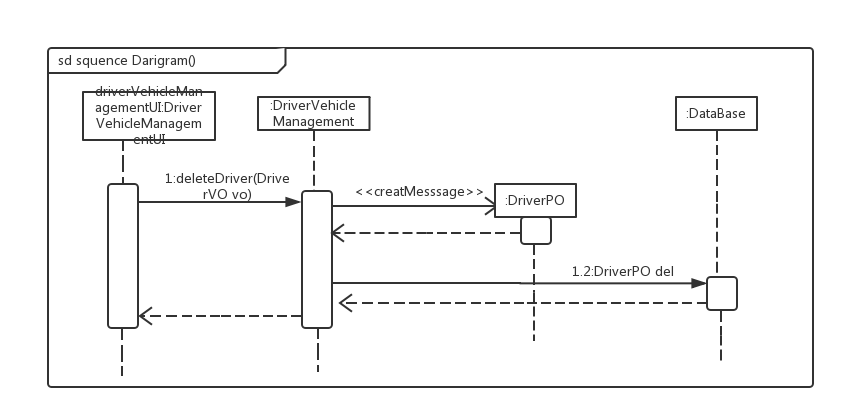
下图为查找司机信息的顺序图



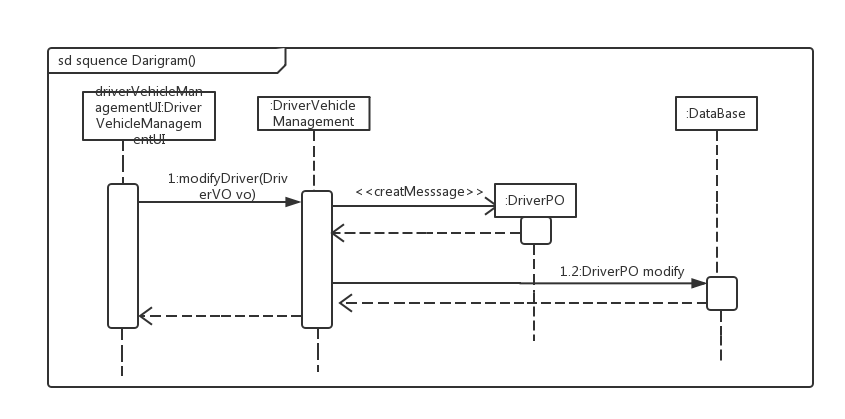
下图为增加司机信息的顺序图



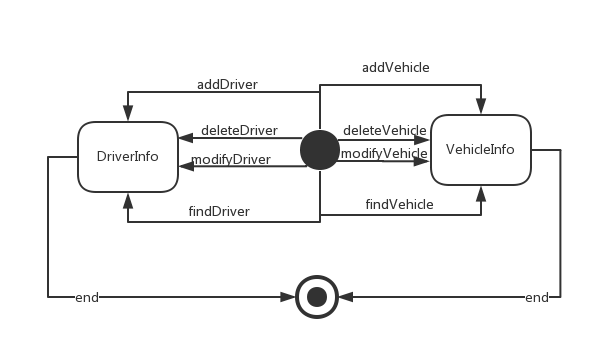
下图为删除司机信息的顺序图



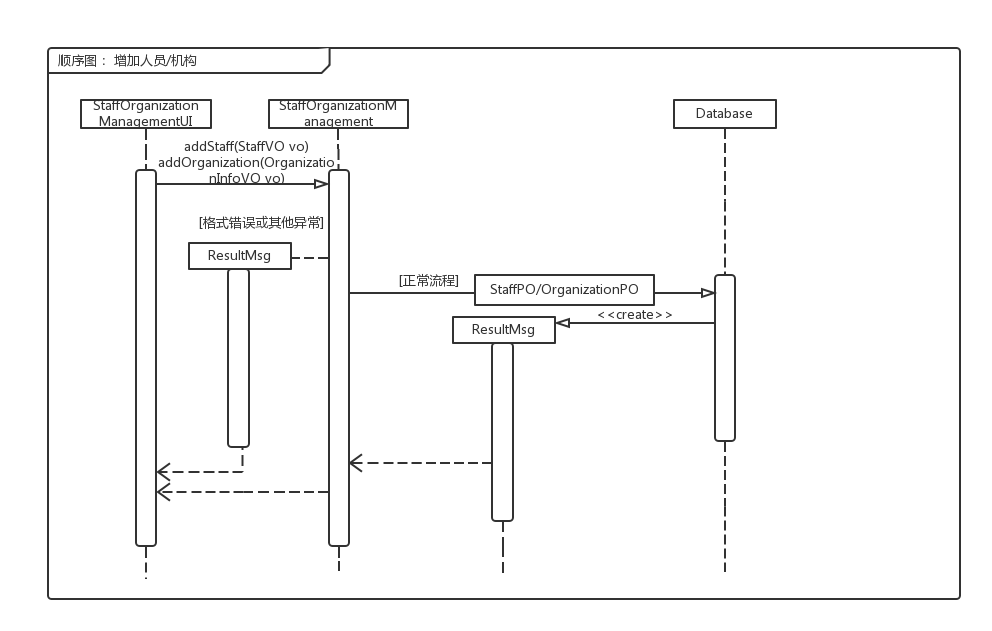
下图为修改司机信息的顺序图



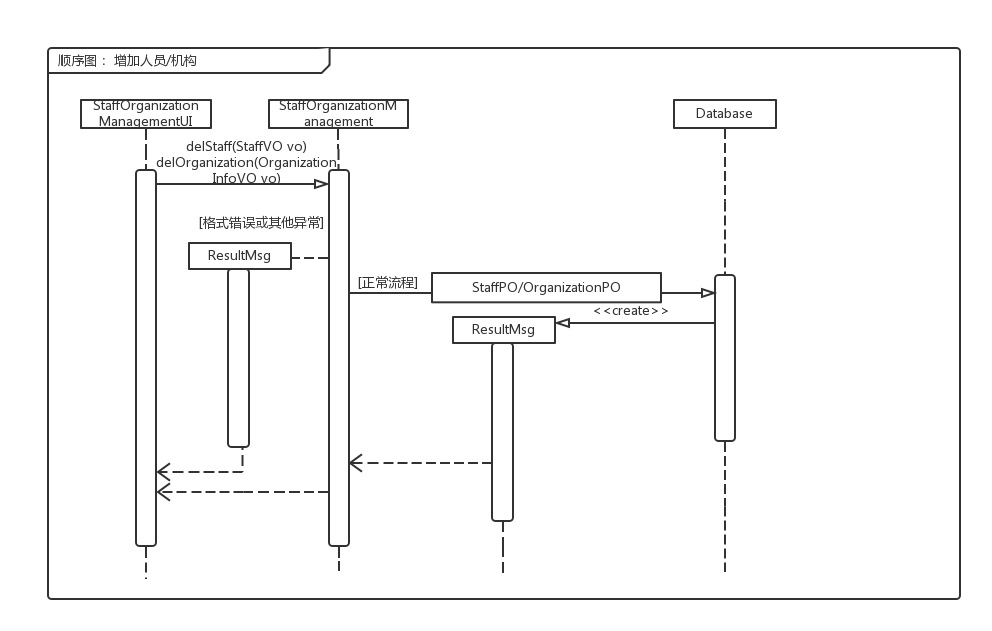
下图所示的状态图描述了DriverVehicleManagement对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着管理车辆与司机信息增删改查的方法被UI调用，DriverVehicleManagement对象被创建，对象进入更新车辆信息或者司机信息状态，经修改信息通过格式检查后，向数据层添加城市距离po持久化对象，业务逻辑对象被回收。



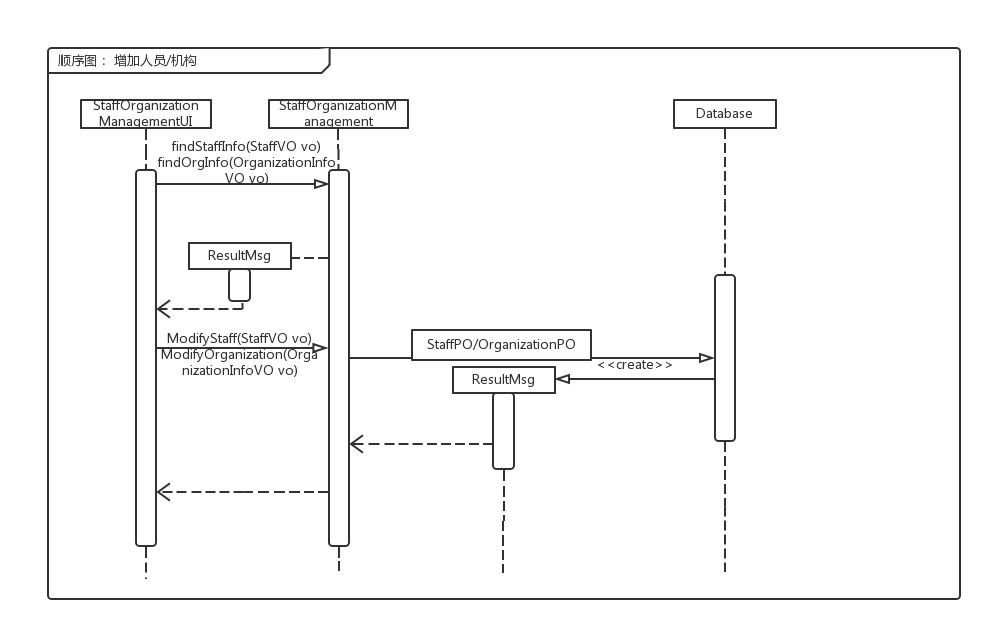
下图为增加人员/机构的类协作顺序图。



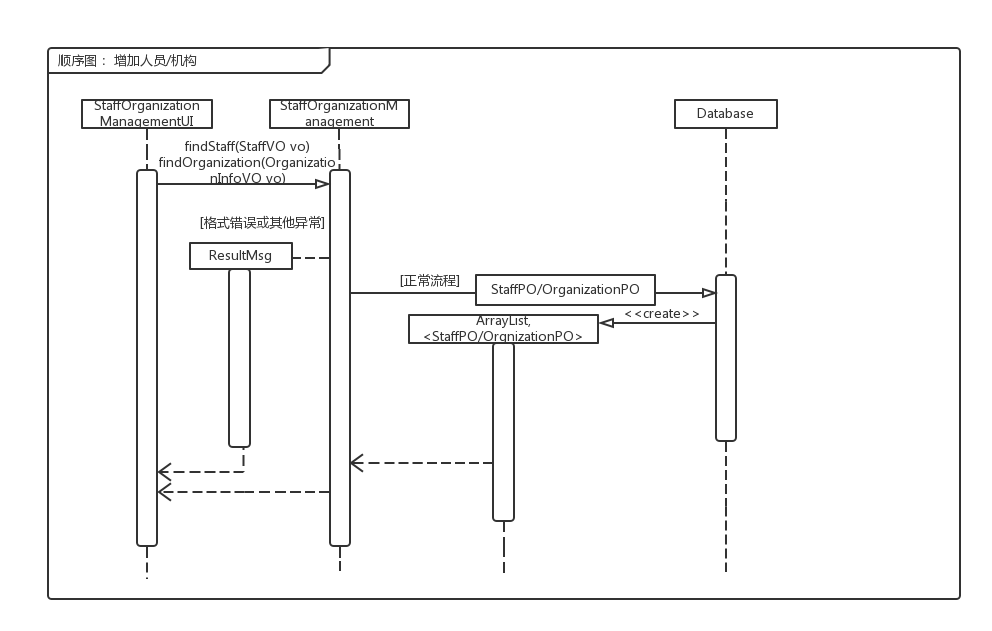
下图为删除人员/机构类协作顺序图。



下图为修改人员/机构信息的类协作顺序图。



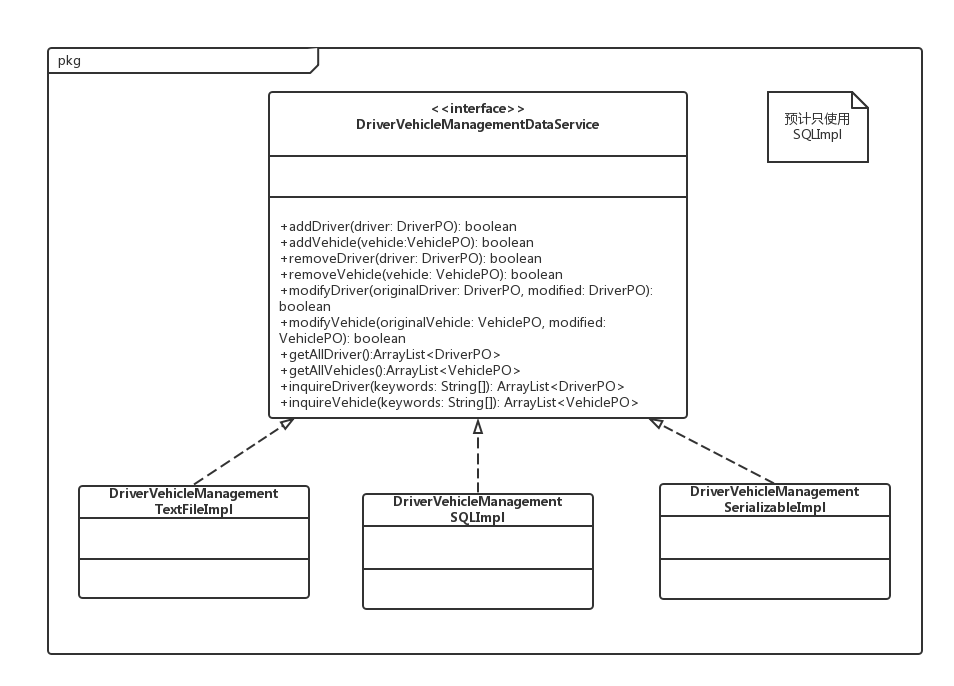
下图为查询人员/机构信息的类协作顺序图。



### 4.2 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所示。



**管理车辆与司机信息数据层模块描述**

#### 4.2.1 数据层模块的职责

数据层模块的职责如表所示

**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 |

#### 4.2.2 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。

**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 |

4.2.3 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

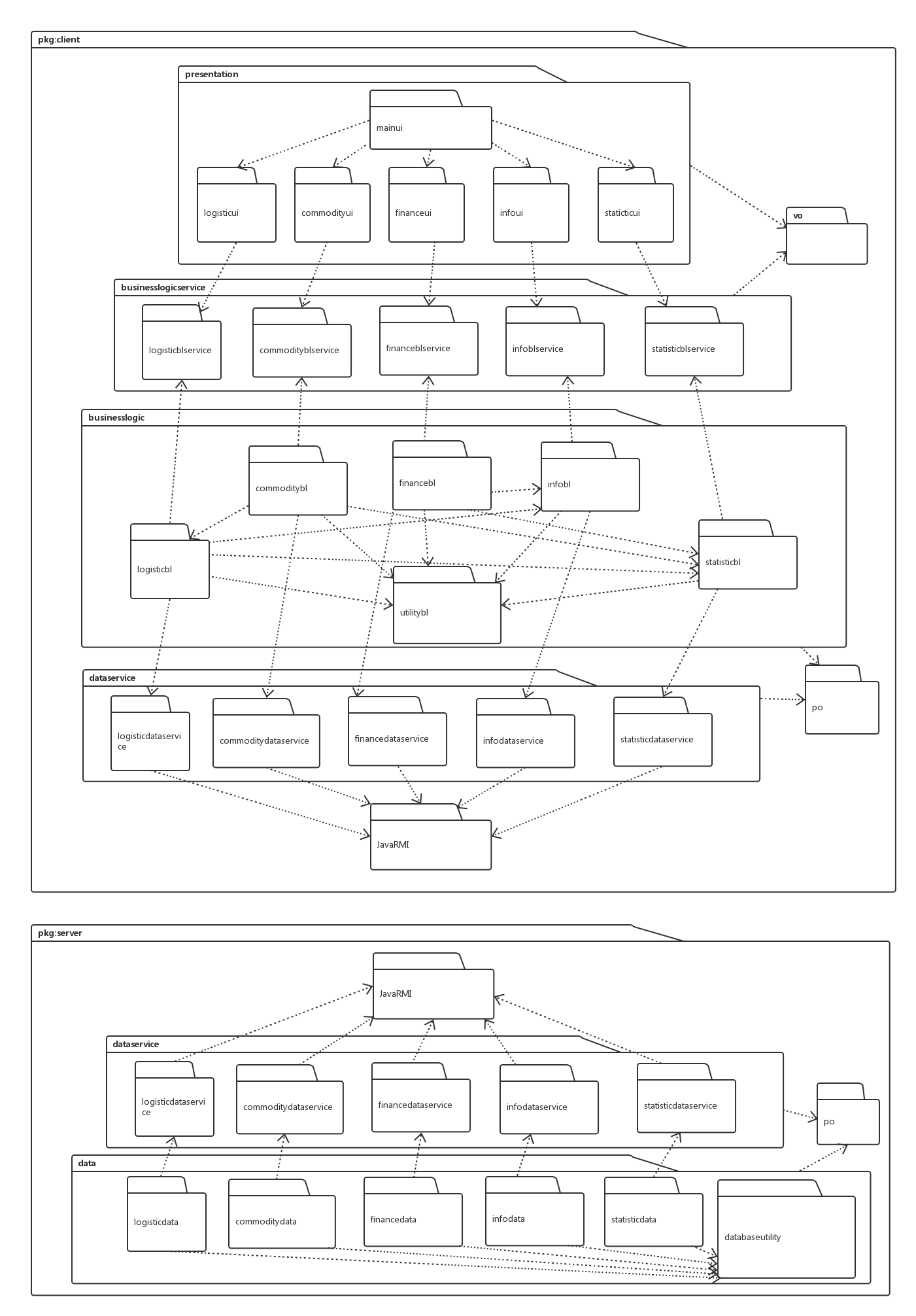
|  |  |  |
| --- | --- | --- |
| 接口名 | 方法 | 详细信息 |
| DatabaseFactory | InventoryDataService getInventoryDataService(); | 从数据层具体实现工厂中获取InventoryDataService 的实现类 |
| PaymentInputDataService getPaymentInputDataService(); | 从数据层具体实现工厂中获取PaymentInputDataService 的实现类 |
| CreditNoteInputDataService getCreditNoteInputDataService(); | 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 |

## 5 依赖视角

下图是客户端和客户端各自的包之间的依赖关系。

****