**快递物流系统**

**(Express Logistics System)**

**软件详细设计模型**



**学 院：南京大学软件学院**

**团 队：伪装者W小组**

**成 员：张词校 张家盛 王丽莉 魏彦淑**

**完成日期：2015年11月16日**

目录

[更新历史 2](#_Toc435398625)

[1. 业务逻辑层的分解 2](#_Toc435398626)

[1.1 expressbl模块 2](#_Toc435398627)

[1.1.1 模块整体结构 2](#_Toc435398628)

[1.1.2 业务逻辑层的动态模型 3](#_Toc435398629)

[1.2 businessbl模块 8](#_Toc435398630)

[1.2.1 模块整体结构 8](#_Toc435398631)

[1.2.2业务逻辑层的动态模型 9](#_Toc435398632)

[1.3 intermediatebl模块 17](#_Toc435398633)

[1.3.1 模块整体结构 17](#_Toc435398634)

[1.3.2业务逻辑层的动态模型 19](#_Toc435398635)

[1.4 receiptbl模块 25](#_Toc435398636)

[1.4.1整体结构 25](#_Toc435398637)

[1.4.2 业务逻辑层的动态模型 26](#_Toc435398638)

[1.5 repertorybl模块 28](#_Toc435398639)

[1.5.1 模块整体结构 28](#_Toc435398640)

[1.5.2 业务逻辑层的动态模型 29](#_Toc435398641)

[1.6 financebl模块 33](#_Toc435398642)

[1.6.1 模块整体结构 33](#_Toc435398643)

[1.6.2 业务逻辑层的动态模型 35](#_Toc435398644)

[1.7 managebl模块 42](#_Toc435398645)

[1.7.1 模块整体结构 42](#_Toc435398646)

[1.7.2 业务逻辑层的动态模型 43](#_Toc435398647)

[1.8 userbl模块 49](#_Toc435398648)

[1.8.1 模块整体结构 49](#_Toc435398649)

[1.8.2 业务逻辑层的动态模型 49](#_Toc435398650)

[2. 依赖视角 51](#_Toc435398651)

# 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 全体成员 | 2015.11.13 | 完成初稿 | V0.1 |
| 张家盛 | 2015.11.15 | 对初稿进行汇总完善 | V1.0 |

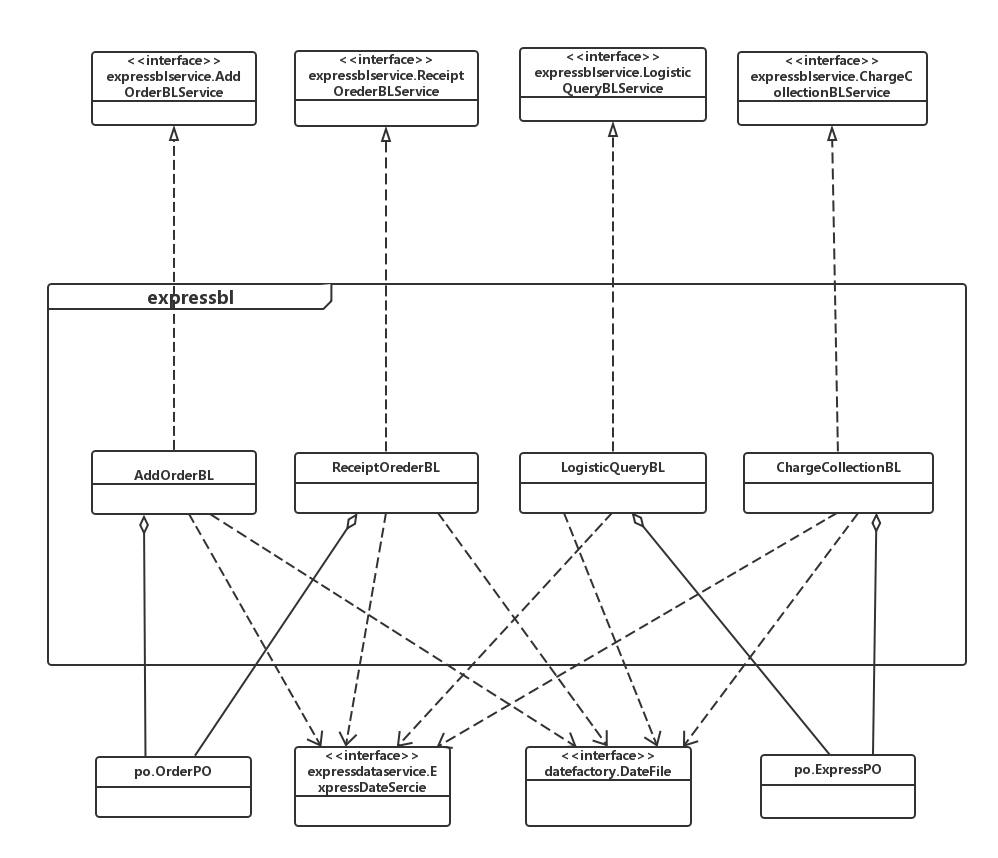
# 业务逻辑层的分解

## 1.1 expressbl模块

### 1.1.1 模块整体结构

expressbl模块实现了展示层与业务逻辑层之间的expressblservice接口以及业务逻辑层和数据层之间的expressdataservice接口。提供了查询订单的calculate和addOrder方法,收件信息输入的getOrderInfo和receiptOrder方法,收费信息汇总的getChargeInfo和chargeCollection方法以及查询订单的query方法。查询订单，收件信息输入，收费信息汇总和查询订单用例分别对应类LogisticQueryBL,ReceiptOrderBL,ChargeCollectionBL和AddOrderBL类，

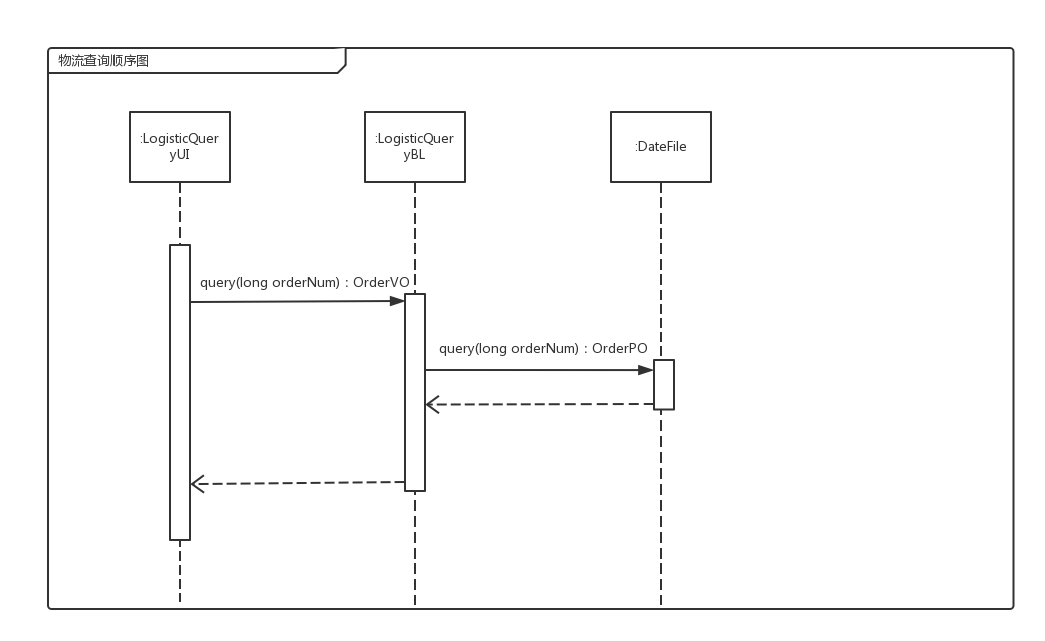
expressbl模块的设计如图1-1所示



**图1-1 expressbl模块各个类的设计**

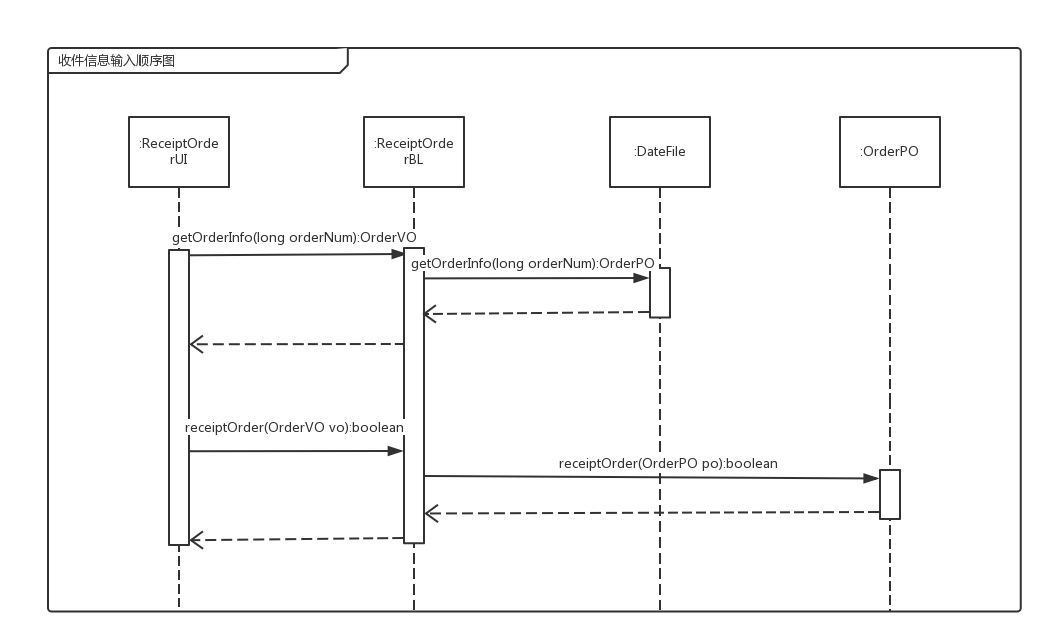
### 业务逻辑层的动态模型

图1-2为LogisticQueryBL领域对象想要新增订单时的顺序图



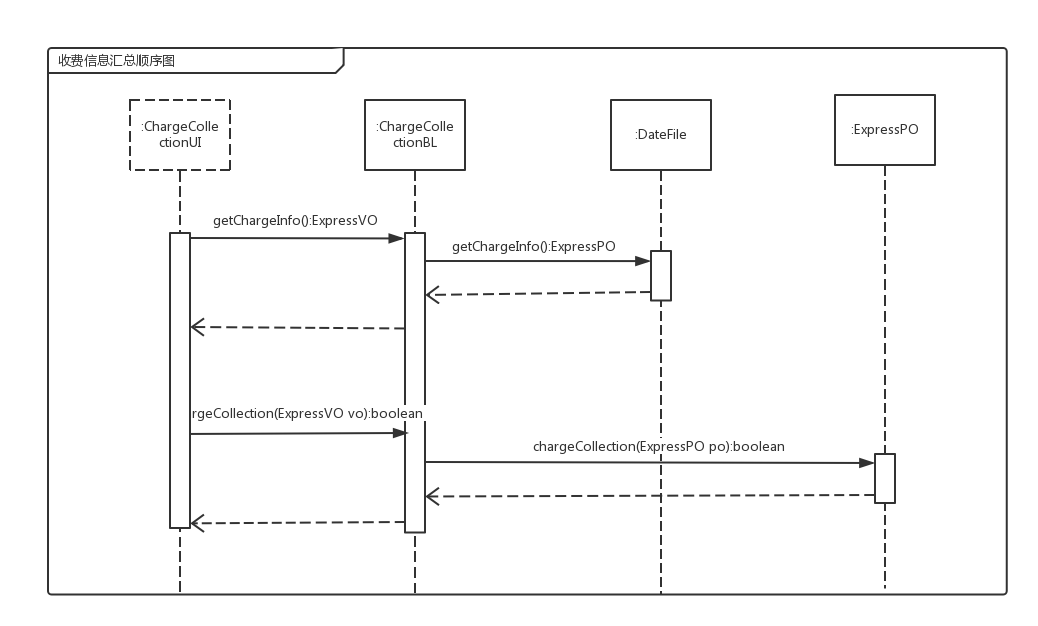
**图1-2物流查询的顺序图**

图1-3为ReceiptOrderBL领域对象想要进行收件信息输入时的顺序图



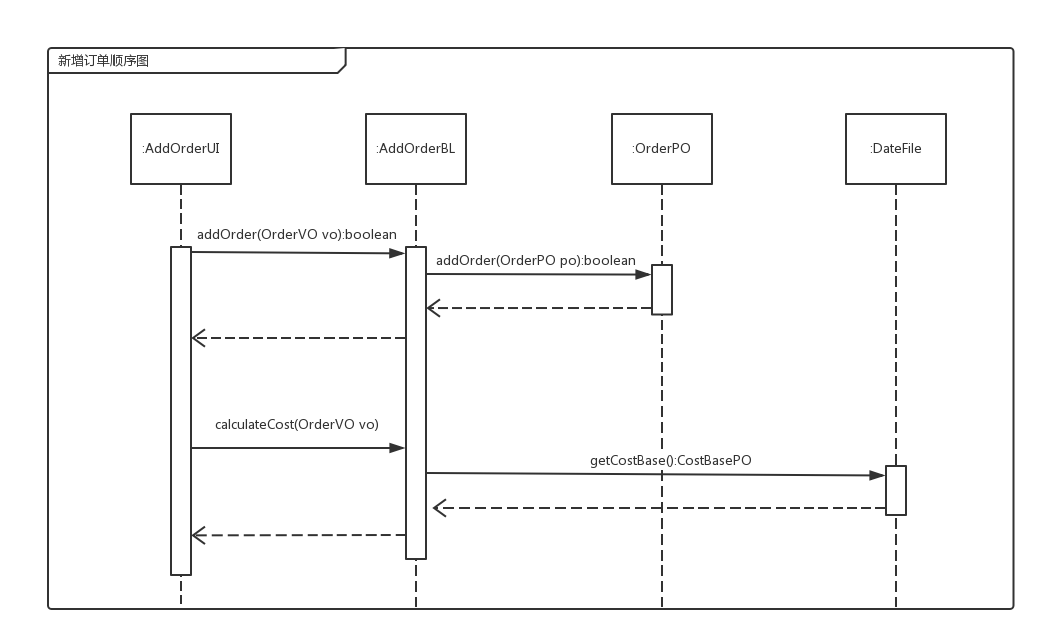
**图1-3 收件信息输入的顺序图**

图1-4为ChargeCollectionBL领域对象想要进行收费信息汇总时的顺序图



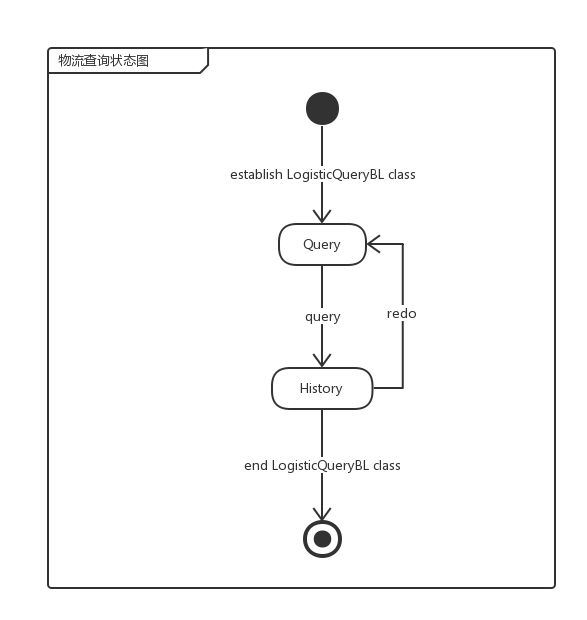
**图1-4 收费信息汇总的顺序图**

图1-5为AddOrderBL领域对象想要新增订单时的顺序图



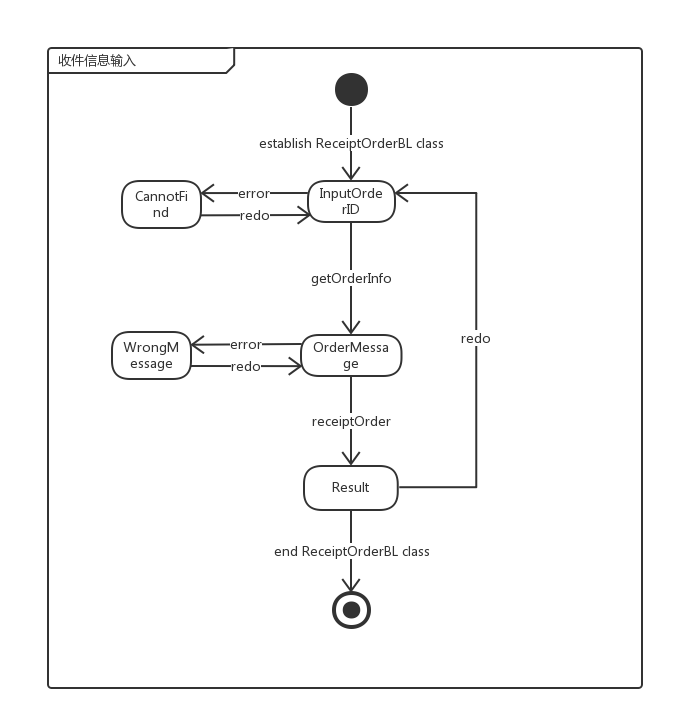
**图1-5新增订单的顺序图**

如图1-6所示，AddOrderBL对象状态图描述了AddOrderBL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establishAddOrderBL方法被 UI调用，AddOrderBL根据不同的指令进入不同状态。



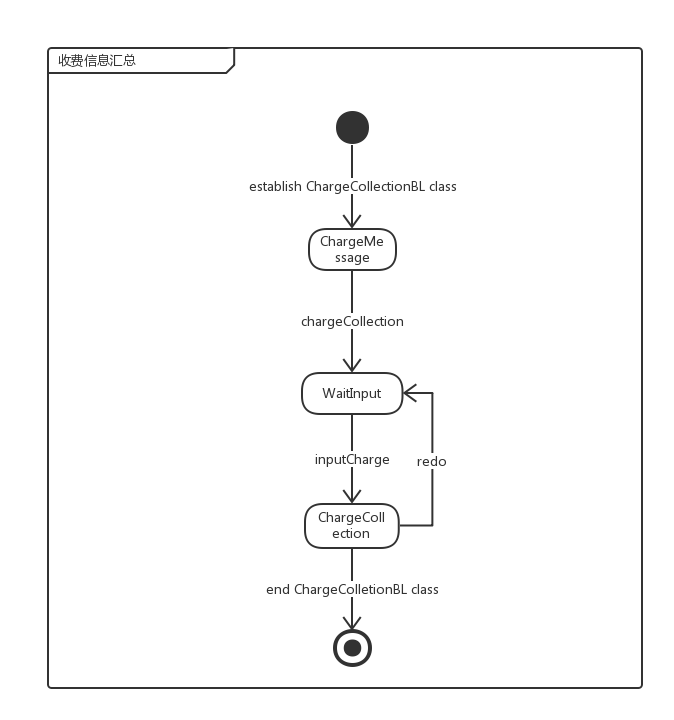
**图1-6 物流查询的状态图**

如图1-7所示，ReceiptOrderBL对象状态图描述了ReceiptOrderBL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establishReceiptOrderBL方法被 UI调用，ReceiptOrderBL根据不同的指令进入不同状态。



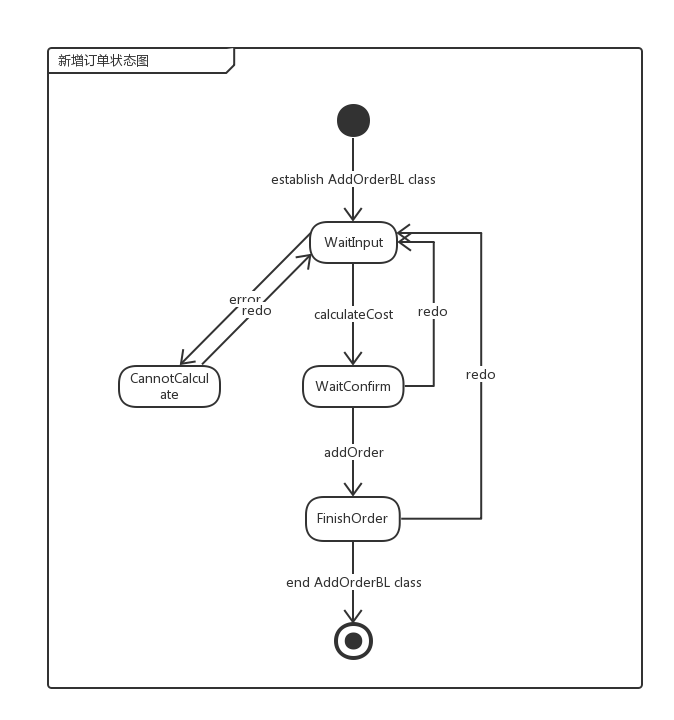
**图1-7 收件信息输入的状态图**

如图1-8所示，ChargeCollectionBL对象状态图描述了ChargeCollectionBL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establishChargeCollectionBL方法被 UI调用，ChargeCollectionBL根据不同的指令进入不同状态。



**图1-8 收费信息输入的状态图**

如图1-9所示，AddOrderBL对象状态图描述了AddOrderBL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establishAddOrderBL方法被 UI调用，AddOrderBL根据不同的指令进入不同状态。

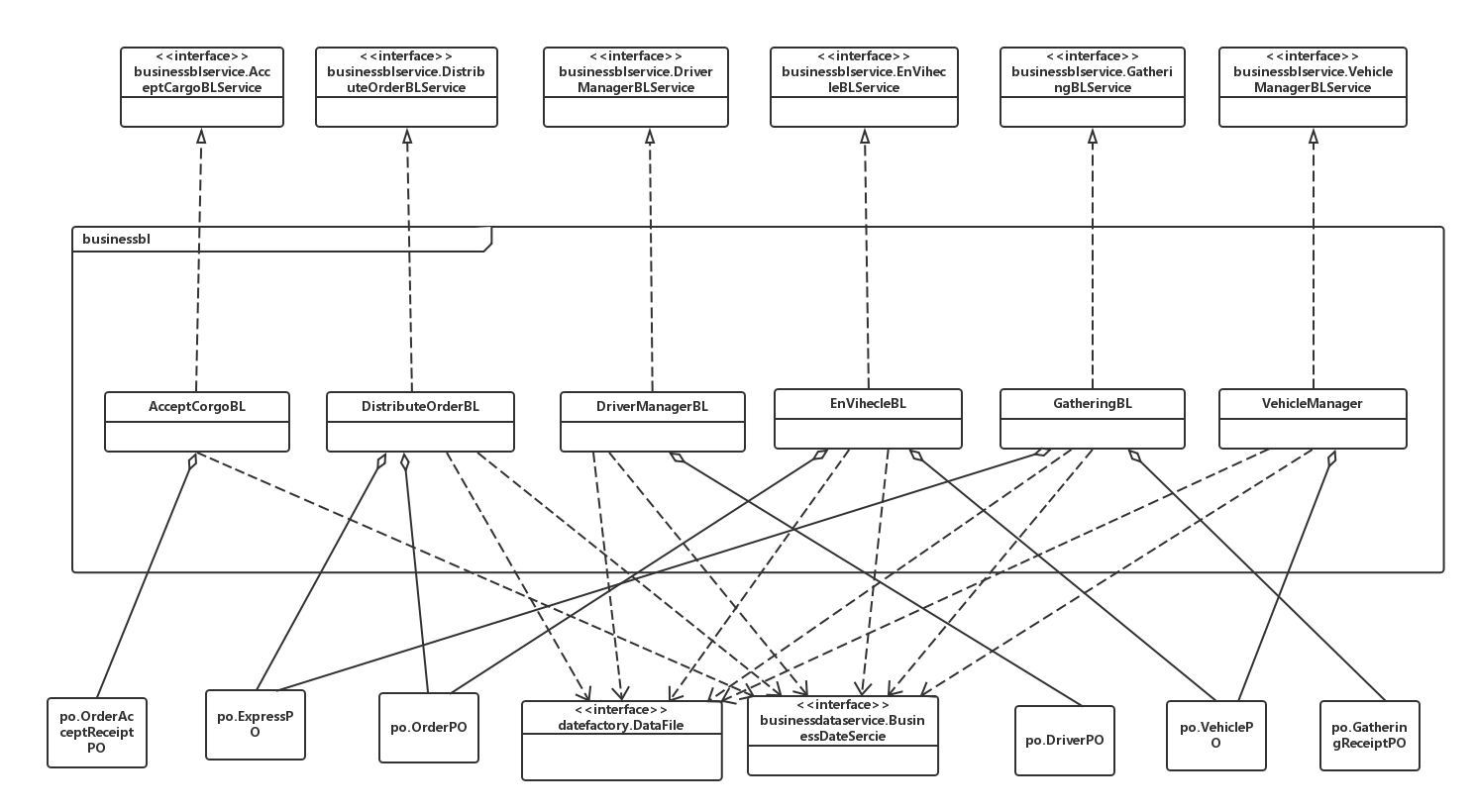


**图1-9 新增订单的状态图**

## 1.2 businessbl模块

### 1.2.1 模块整体结构

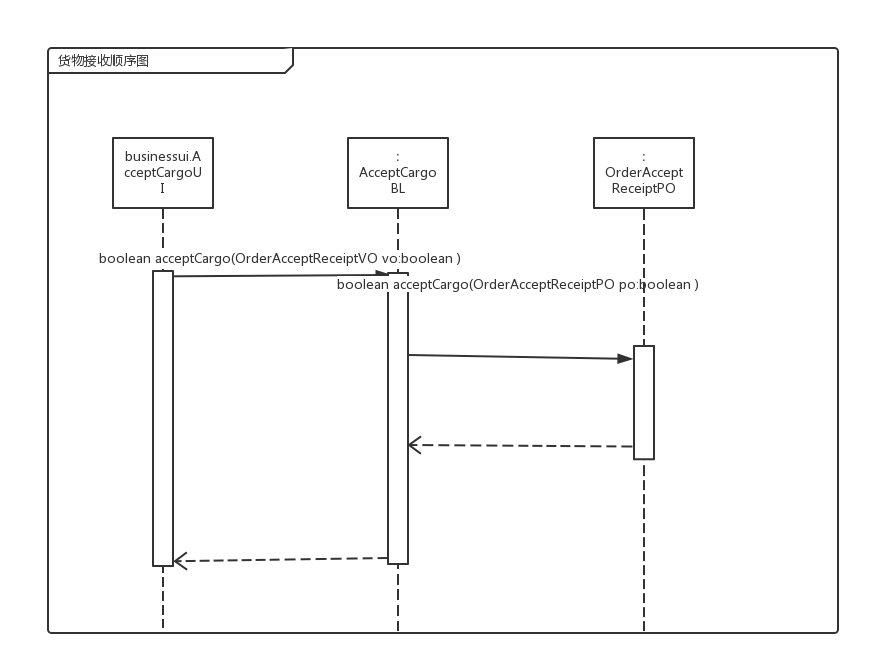
businessbl模块实现了展示层和业务逻辑层之间的businessblservice接口以及业务逻辑层和数据层之间的businessdataservice接口。提供了接收货物的acceptCargo方法，装车管理的autoTruckLoading方法，司机信息管理的addDriver，deleteDriver，modifyDriver方法，车辆信息管理的addVehicle，deleteVehicle，modifyVehicle方法，派件的distributeOrder方法以及收款汇总的gathering方法，接收货物，装车管理，司机信息管理，车辆信息管理，派件，收款汇总用例分别对应类AcceptCargoBL，EnVehicleBL，DriverManagerBL，VehicleManagerBL，DistributeOrderBL，GatheringBL类。

businessbl模块的设计如图2-1所示

**图2-1 businessbl模块各个类的设计**

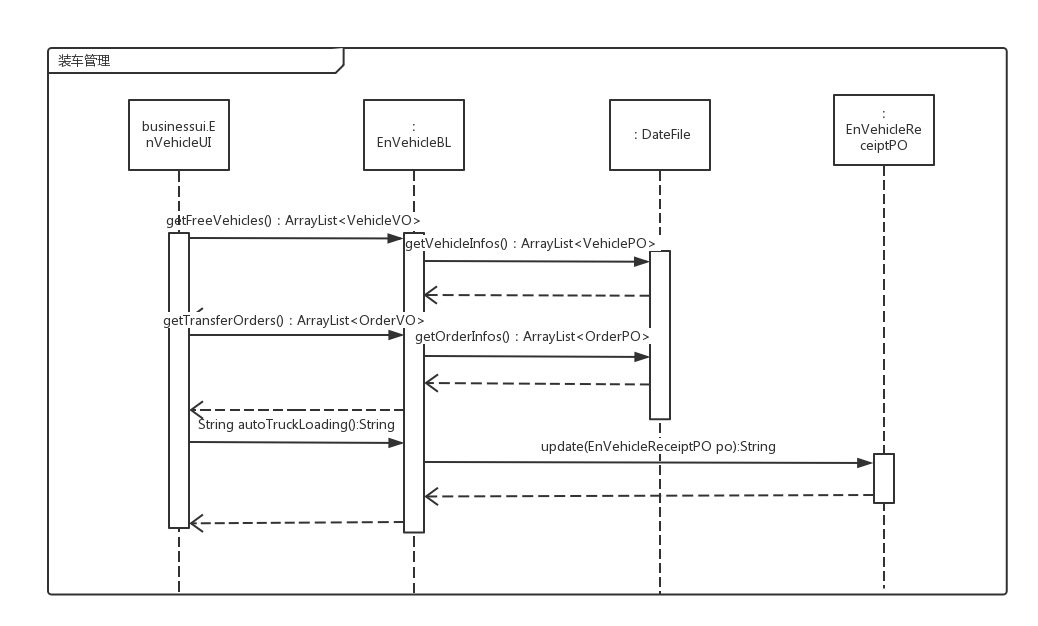
### 1.2.2业务逻辑层的动态模型

图2-2为 AcceptCargoBL领域对象想要进行货物接收时的顺序图

****

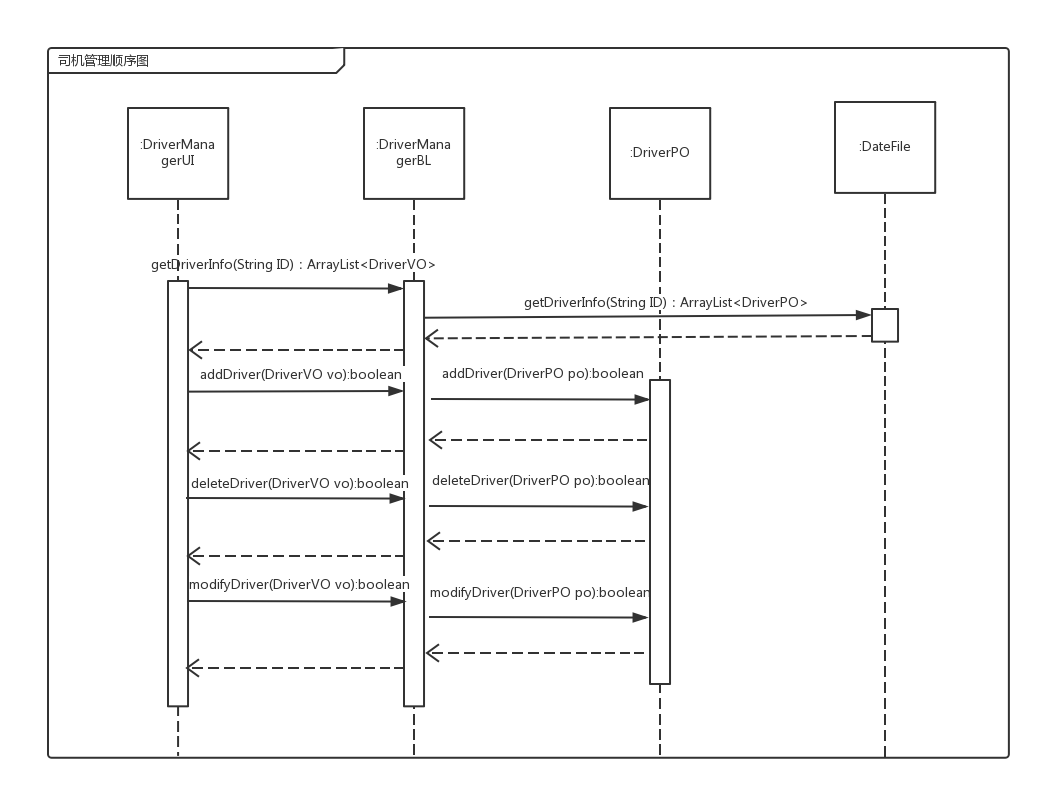
**图2-2 货物接收的顺序图**

图2-3为 EnVehicleBL领域对象想要进行装车管理时的顺序图



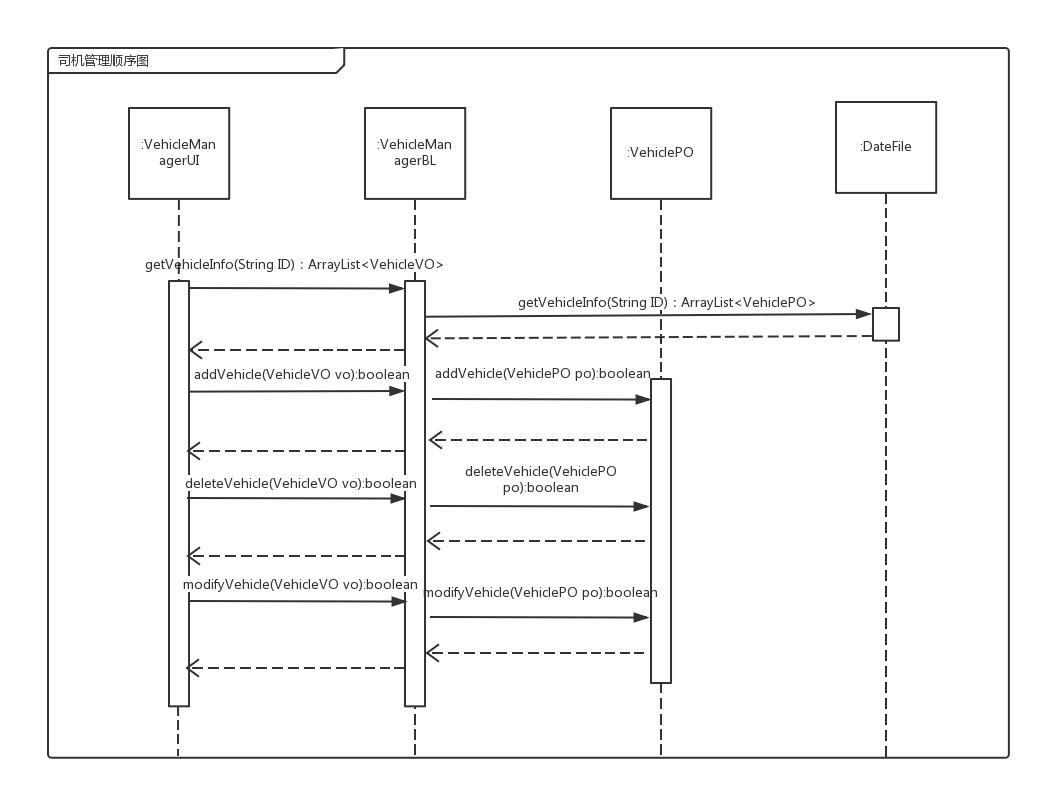
**图2-3 装车管理的顺序图**

图2-4为 DriverManagerBL领域对象想要进行司机信息管理时的顺序图



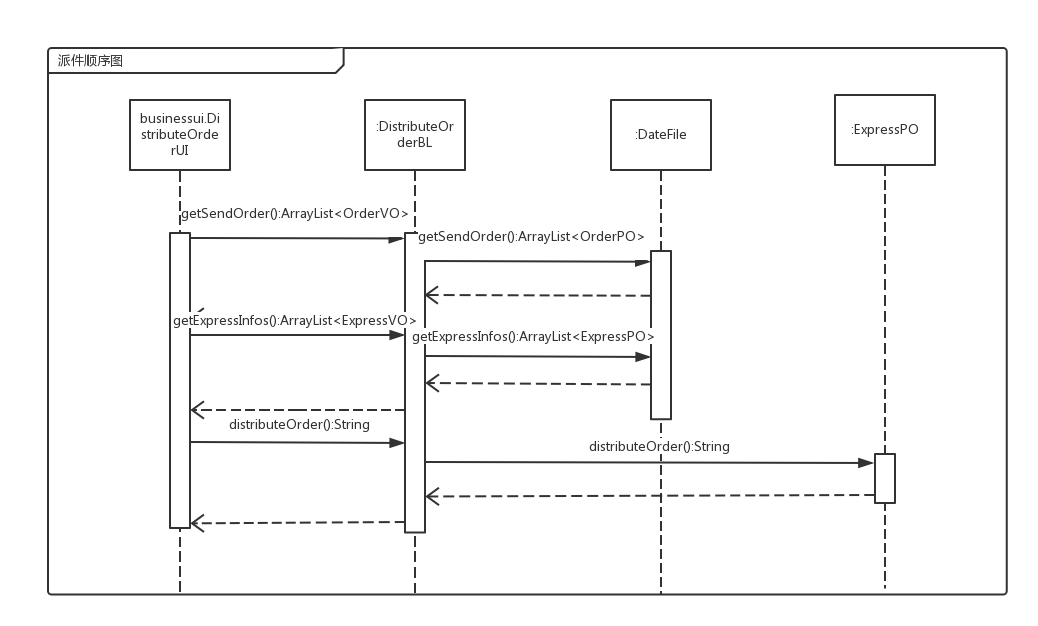
**图2-4 司机信息管理的顺序图**

图2-5为 VehicleManagerBL领域对象想要车辆信息管理时的顺序图



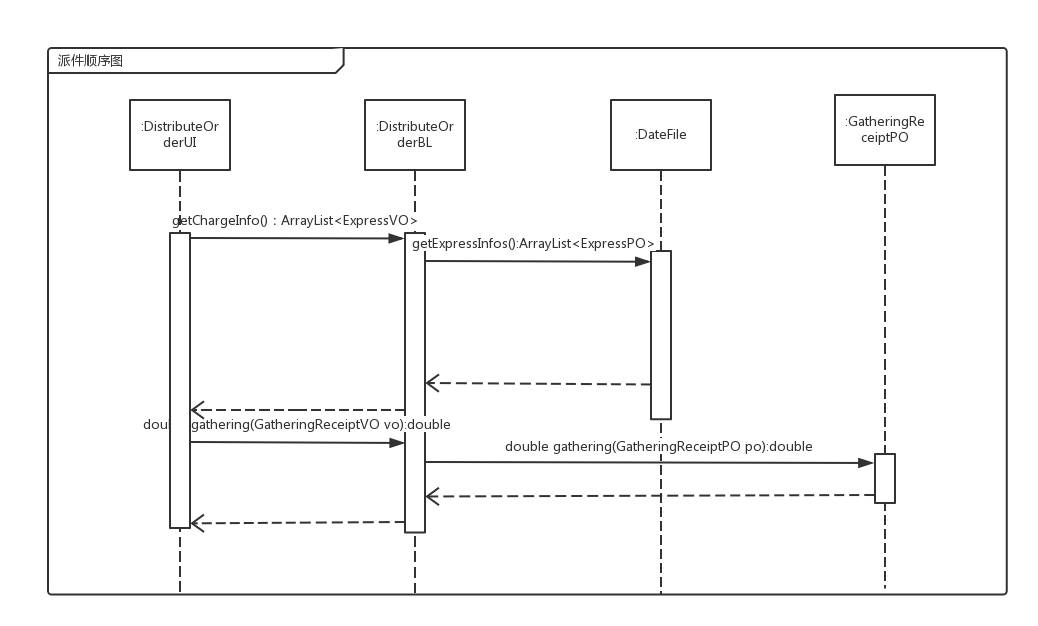
**图2-5 车辆信息管理的顺序图**

图2-6为 DistributeOrderBL领域对象想要派件时的顺序图



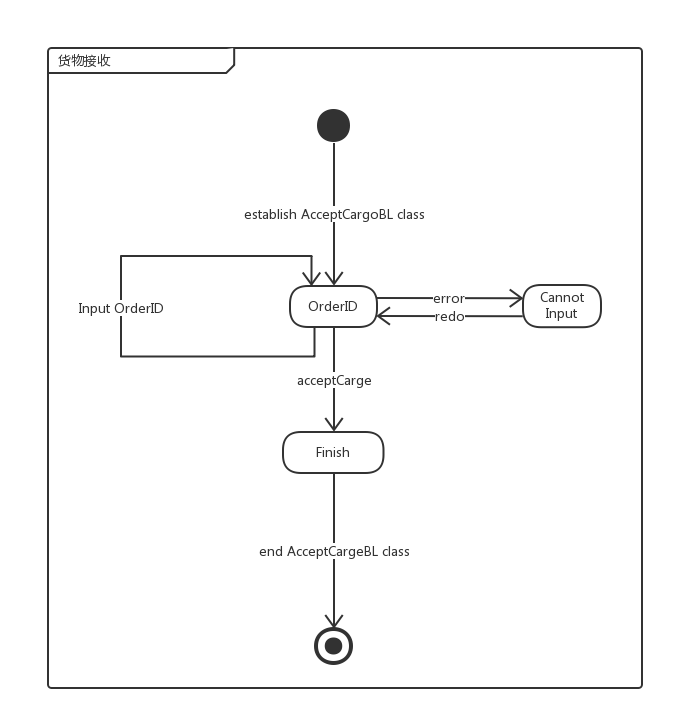
**图2-6 派件的顺序图**

图2-7为 GatheringBL领域对象想要进行收款汇总时的顺序图



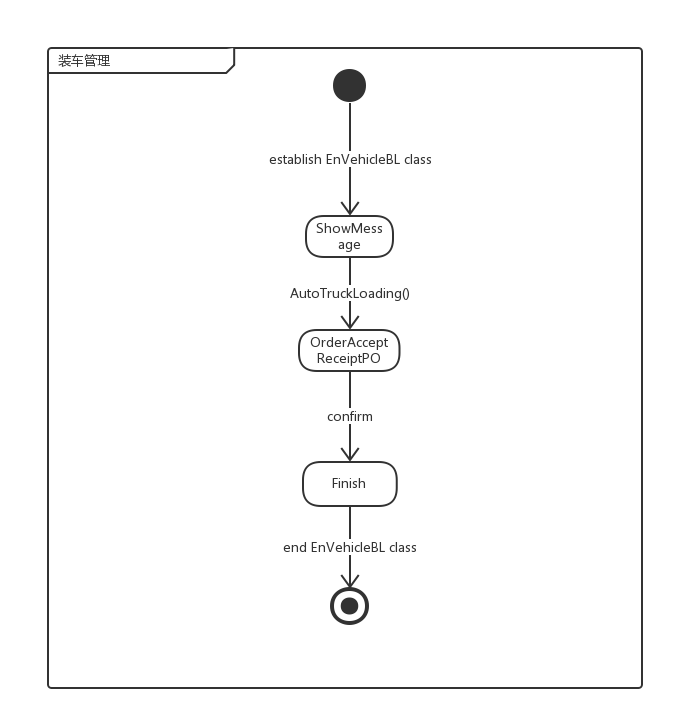
**图2-7 收款汇总的顺序图**

如图2-8所示，AcceptCargoBL对象状态图描述了AcceptCargoBL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establishAcceptCargoBL方法被 UI调用，AcceptCargoBL根据不同的指令进入不同状态。

****

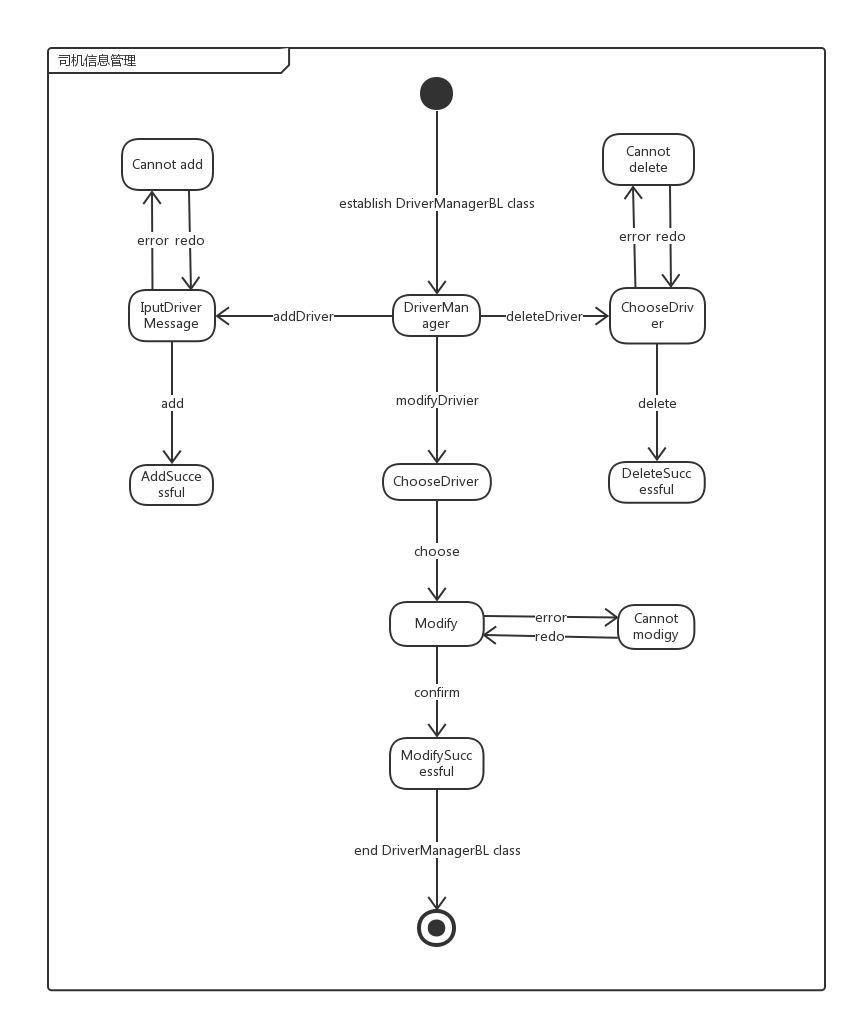
**图2-8 接收货物的状态图**

如图2-9所示，EnVehicleBL对象状态图描述了EnVehicleBL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establishEnVehicleBL方法被 UI调用，EnVehicleBL根据不同的指令进入不同状态。



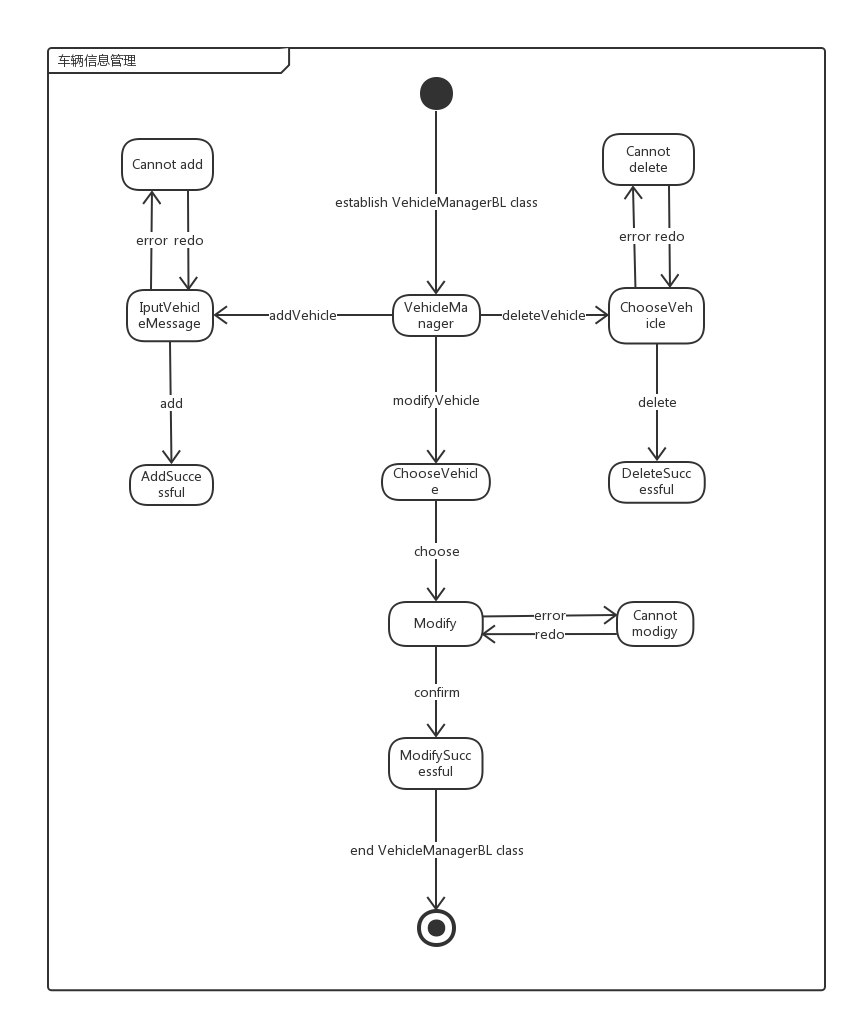
**图2-9 装车管理的状态图**

如图2-10所示，DriverManagerBL对象状态图描述了DriverManagerBL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establishDriverManagerBL方法被 UI调用，DriverManagerBL根据不同的指令进入不同状态。



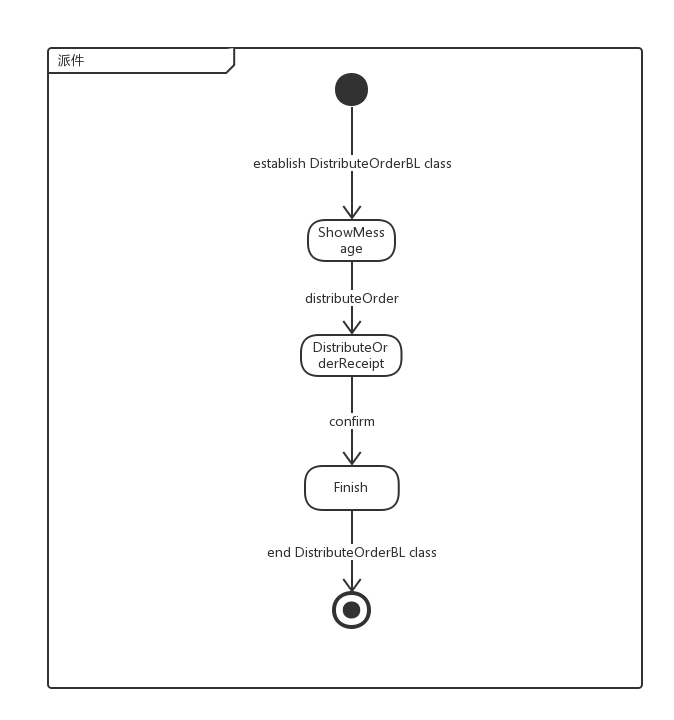
**图2-10 司机信息管理的状态图**

如图2-11所示，VehicleManagerBL对象状态图描述了VehicleManagerBL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establishVehicleManagerBL方法被 UI调用，VehicleManagerBL根据不同的指令进入不同状态。



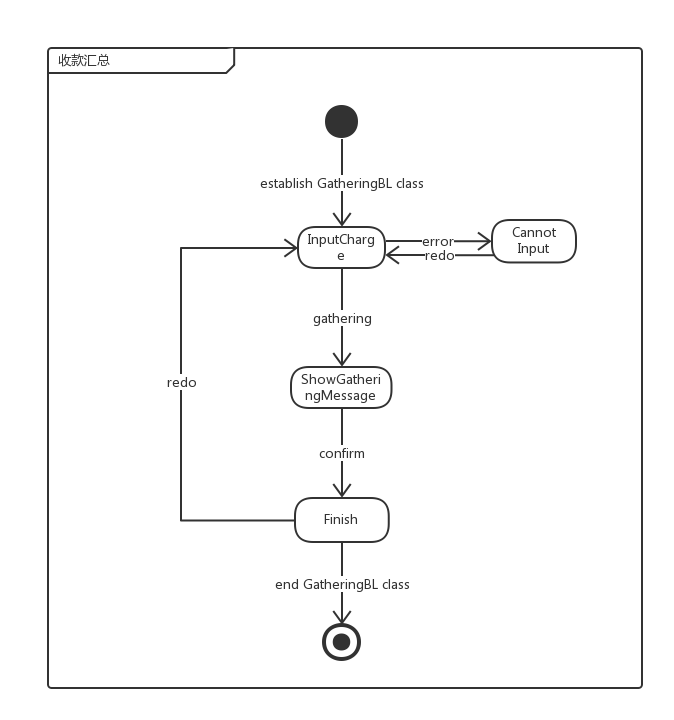
**图2-11 车辆信息管理的状态图**

如图2-12所示，DistributeOrderBL对象状态图描述了DistributeOrderBL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establishDistributeOrderBL方法被 UI调用，DistributeOrderBL根据不同的指令进入不同状态。



**图2-12 派件的状态图**

如图2-13所示，GatheringBL对象状态图描述了GatheringBL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establishGatheringBL方法被 UI调用，GatheringBL根据不同的指令进入不同状态。

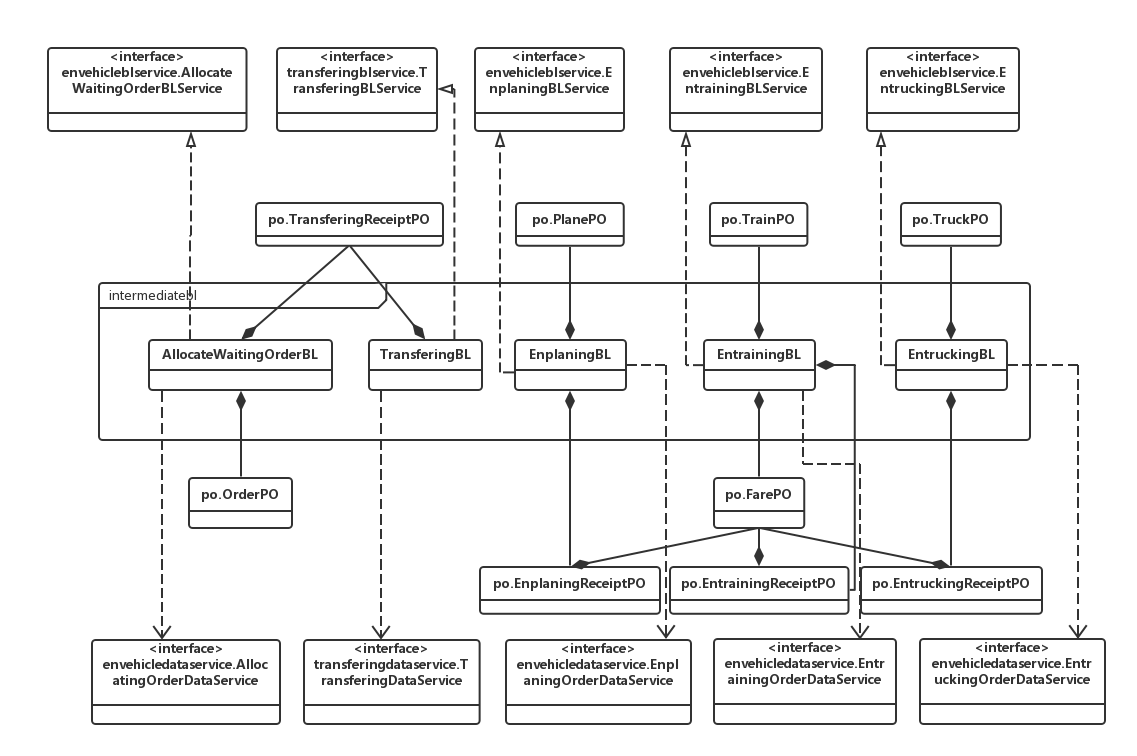
****

**图2-13 收款汇总的状态图**

## 1.3 intermediatebl模块

### 1.3.1 模块整体结构

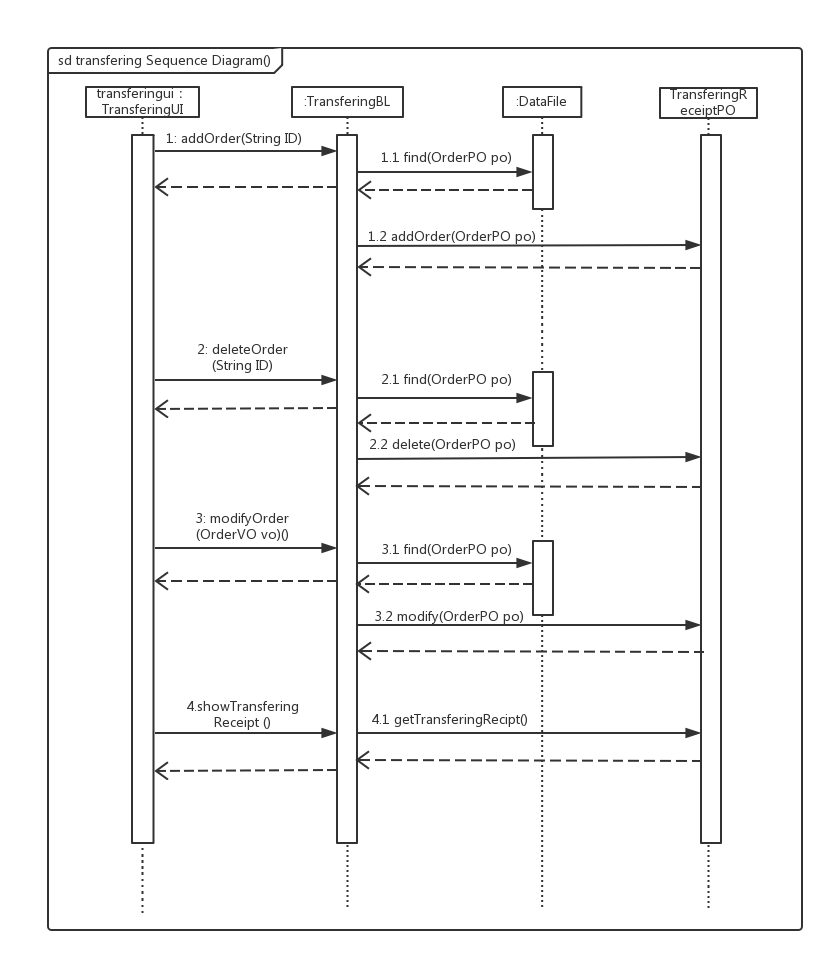
根据体系结构的设计，我们将系统分为展示层，业务逻辑层，数据层。每一层之间为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们添加businesslogicservice.intermediateblservice.envehicleblservice.EnplaningBLService,businesslogicservice.intermediateblservice.envehicleblservice.EntrainingBLService,businesslogicservice.intermediateblservice.envehicleblservice.EntruckingBLService,businesslogicservice.intermediateblserv,ce.envehicleblservice.AllocateBLService,businesslogicservice.intermediateblservice.transferingblservice.TransferingBLService接口。业务逻辑层和数据层之间添加dataservice.intermediatedataservice.envehicledataservice.EnplaningDataService,dataservice.intermediatedataservice.envehicledataservice.EntrainingDataService,dataservice.intermediatedataservice.envehicledataservice.EntruckingDataService,dataservice.intermediatedataservice.envehicledataservice.AllocateDataService,dataservice.intermediatedataservice.transferingdataservice.TransferingDataService接口。TransferingReceiptPO是作为中转中心到达单，PlanePO是作为飞机，TrainPO是作为火车，TruckPO是作为汽车，OrderPO是作为订单，FarePO是作为运费，EnplaningReceiptPO，EntrainingReceiptPO，EntruckingReceiptPO是作为装车单，以上PO分别作为持久化对象被添加到 设计模型中去的。Intermediatebl模块的设计如图3-1所示。



**图3-1 intermediatebl模块各个类的设计**

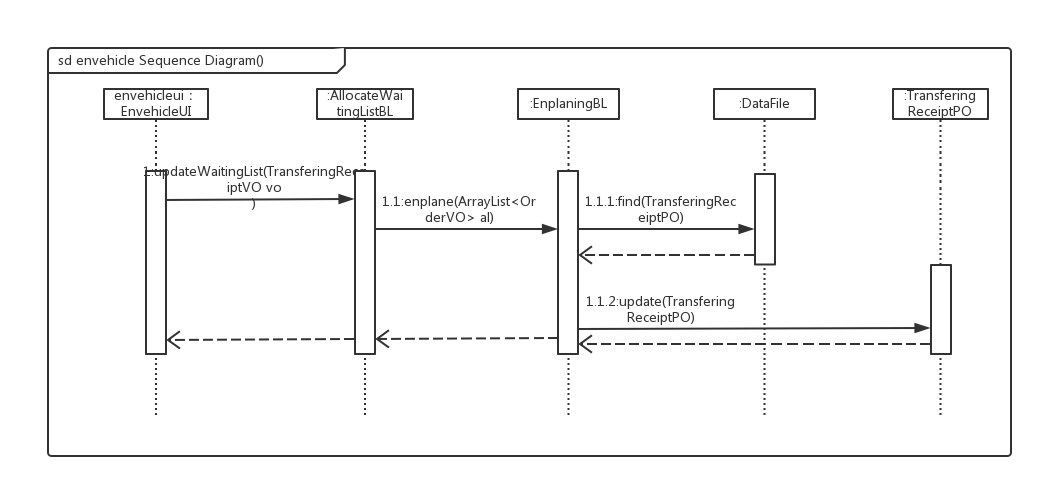
### 1.3.2业务逻辑层的动态模型

图3-2表明了ELS快递物流系统中，当中转中心业务员选择中转接收功能并输入新增中转中心到达单上的底单之后，中转接收业务逻辑处理的相关对象之间的协作。

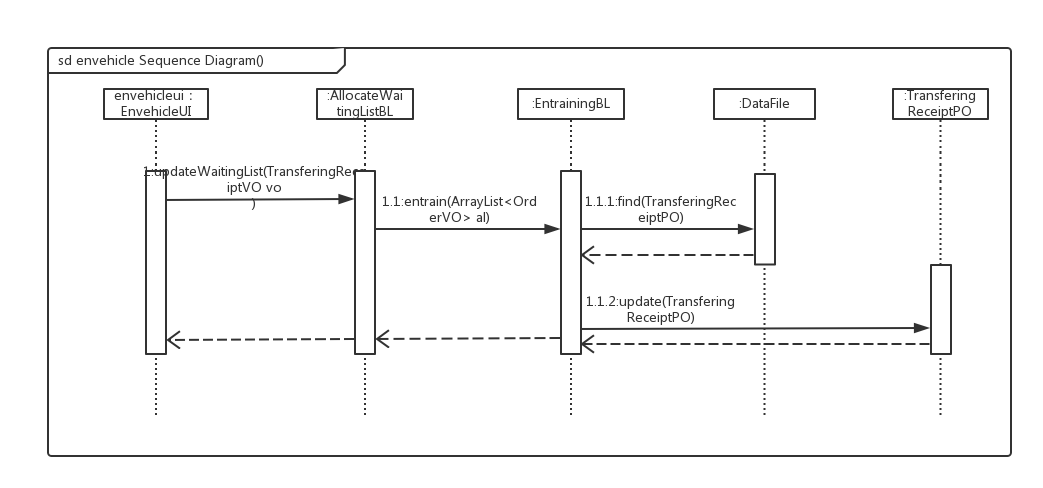


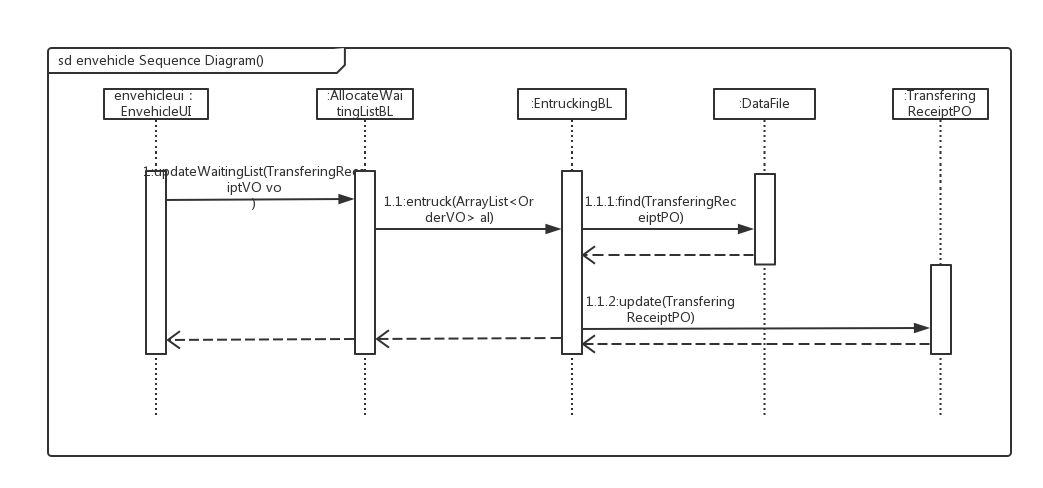
**图3-2 中转接收的顺序图**

图3-3，3-4，3-5表明了ELS快递物流系统中，当用户选择飞机，火车，汽车装运管理并更新待装车订单列表之后，飞机，火车，汽车装运管理业务逻辑处理的相关对象之间的协作。

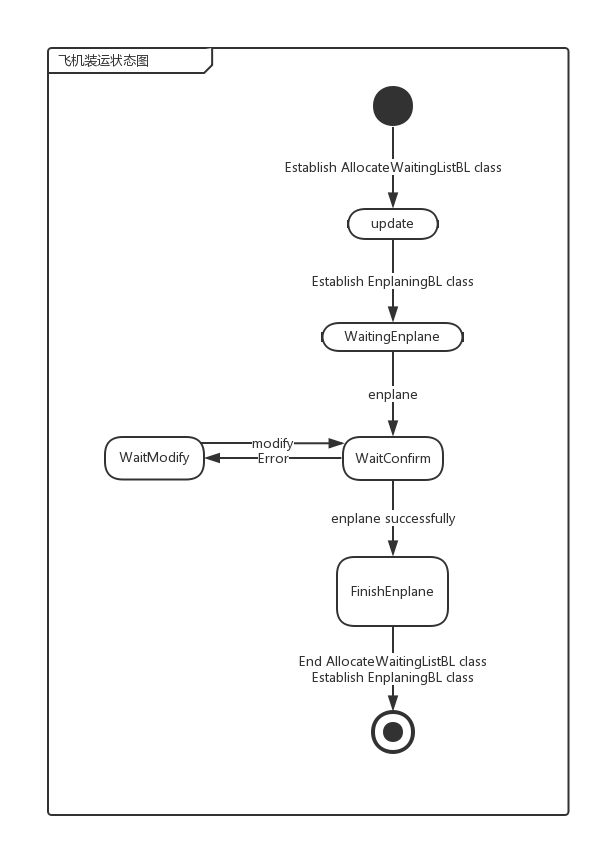


**图3-3 飞机装运管理的顺序图**

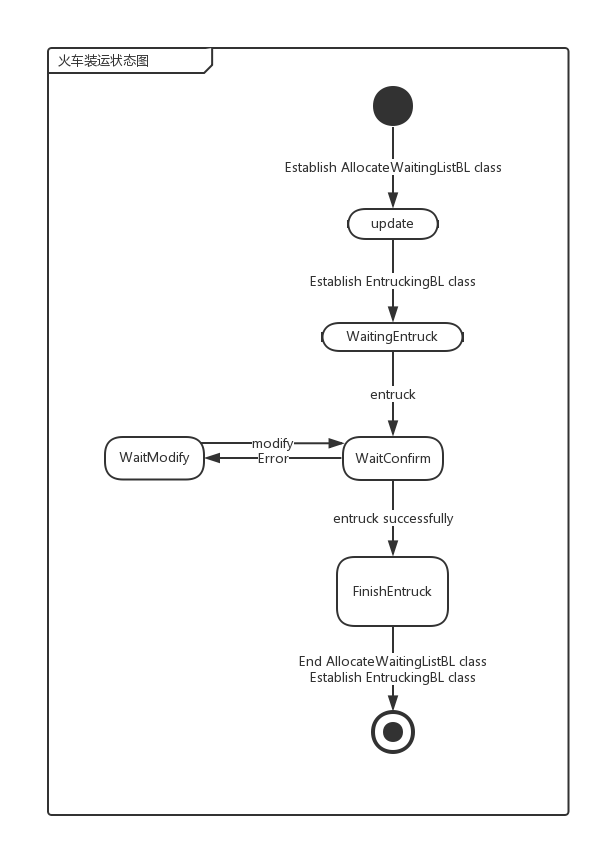
**图3-4 火车装运管理的顺序图**

**图3-5 汽车装运管理的顺序图**

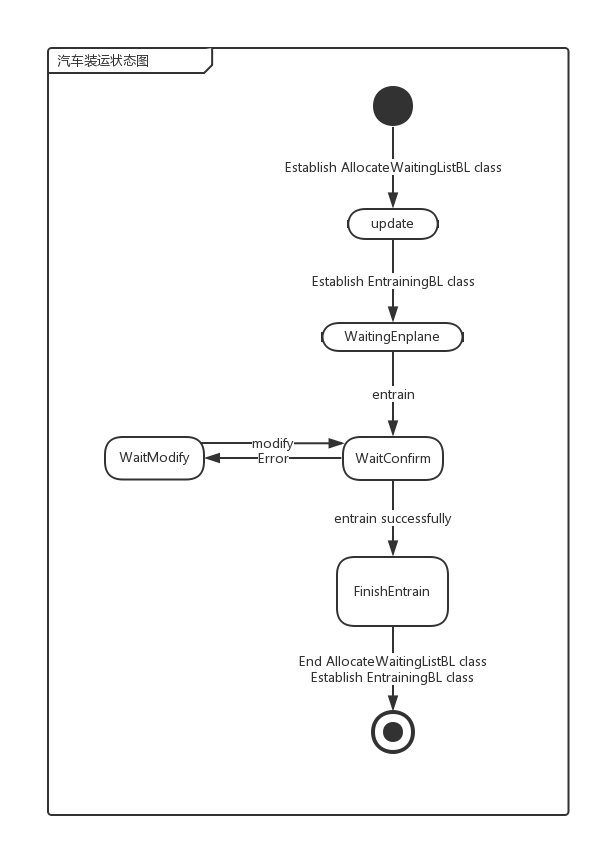
如图3-6，3-7，3-8所示，装运管理对象状态图分别描述了Enplaning，Entraining，Entrucking对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着establishAllocateWaitingList方法以及establishEnplaning，establishEntraining，establishEntrucking被 UI调用，Enplaning，Entraining，Entrucking根据不同的指令进入不同状态。

****

**图3-6 飞机装运管理的状态图**

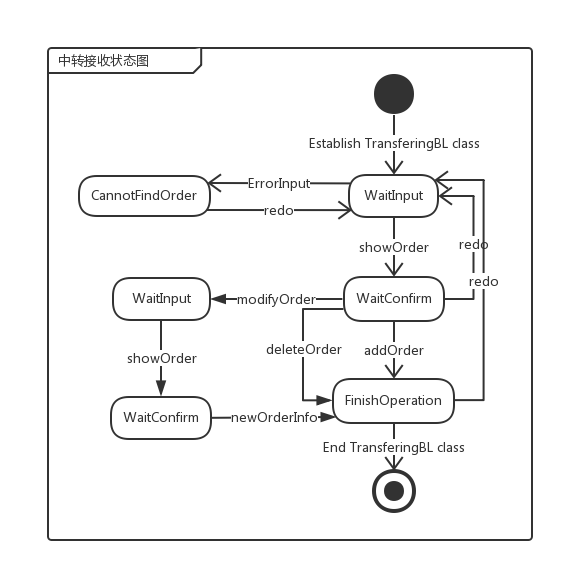
****

**图3-7 火车装运管理的状态图**

****

**图3-8 汽车装运管理的状态图**

如图3-9所示，中转接收对象状态图描述了Transfering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establishTransfering方法被 UI调用，establishTransfering根据不同的指令进入不同状态。

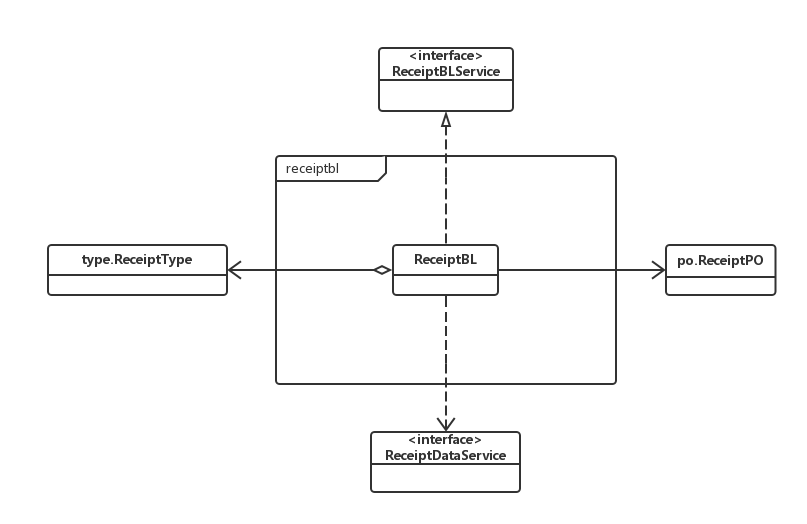


**图3-9 中转接收的状态图**

## 1.4 receiptbl模块

### 1.4.1整体结构

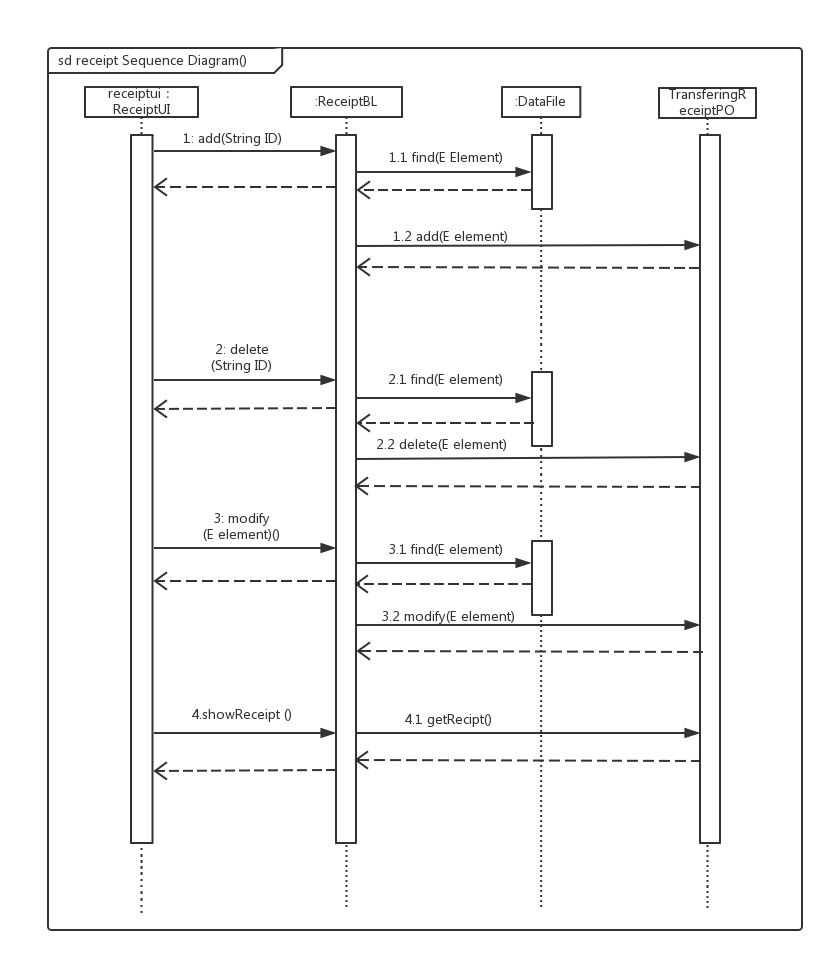
根据体系结构的设计，我们将系统分为展示层，业务逻辑层，数据层。每一层之间为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们添加receiptblservice.ReceiptBLService接口。业务逻辑层和数据层之间添加receiptdataservice.ReceiptDataService接口。ReceiptPO是作为所有单据父类的持久化对象被添加到设计模型中去的。ReceiptType是作为存放所有ReceiptPO当前状态的枚举被添加到设计模型中去的。ReceiptBL模块的设计如图4-1所示。



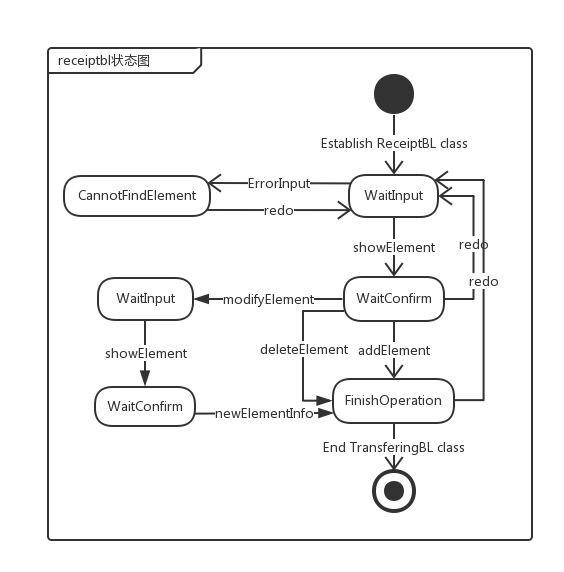
**图4-1 receiptbl模块各个类的设计**

### 1.4.2 业务逻辑层的动态模型

图4-2表明了ELS快递物流系统中，当选择单据增删改查的基本操作之后，单据业务逻辑处理的相关对象之间的协作。

**图4-2 单据增删改查的顺序图**

如图4-3所示，单据增删改查对象状态图描述了Receipt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establishReceipt方法被 UI调用，establishReceipt根据不同的指令进入不同状态。

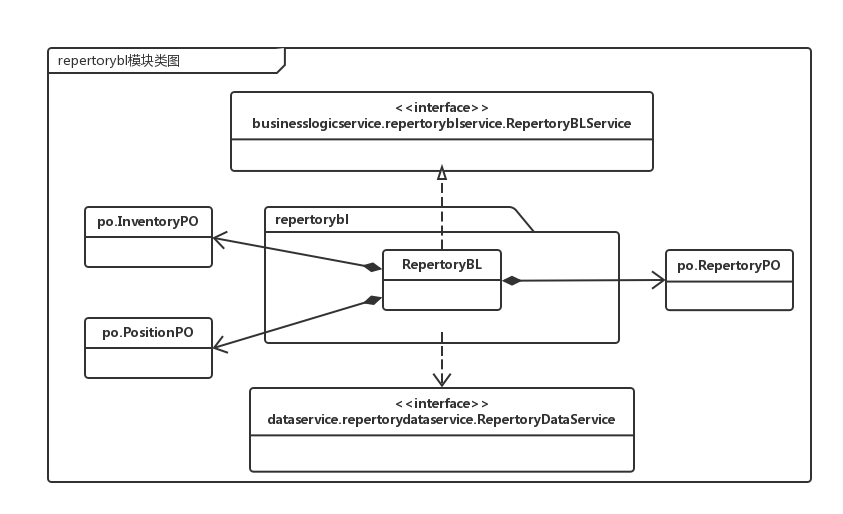


**图4-3 单据增删改查的状态图**

## 1.5 repertorybl模块

### 1.5.1 模块整体结构

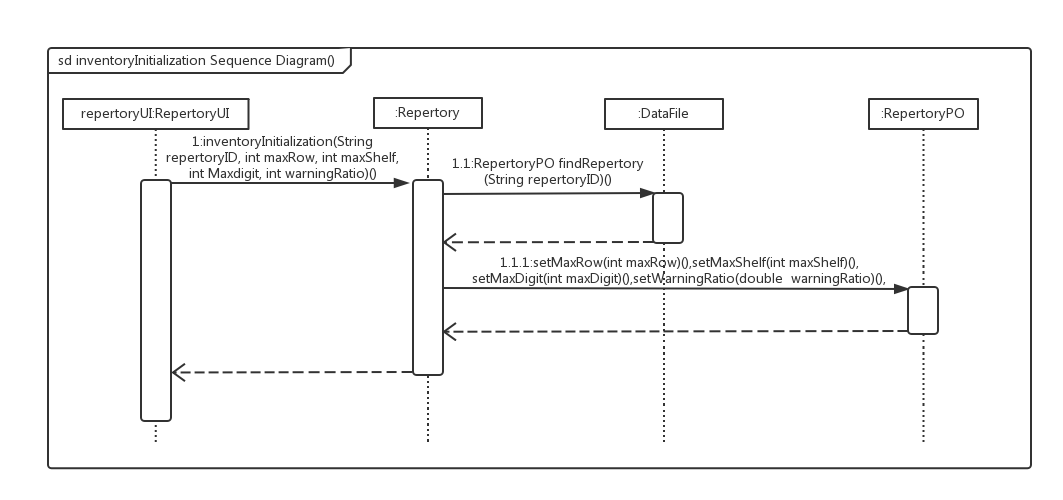
根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们添加businesslogicservice.repertoryblservice.RepertoryBLService接口。业务逻辑层和数据层之间添加dataservice.repertorydataservice.RepertoryDataService接口。因为业务逻辑处理较少，根据专家模式没有添加Controller类。RepertoryPO等PO对象是作为仓库管理员管理记录的持久化对象被添加到设计模型中去的。

repertorybl模块的设计如图5-1所示。

**图5-1 repertorybl模块各个类的设计**

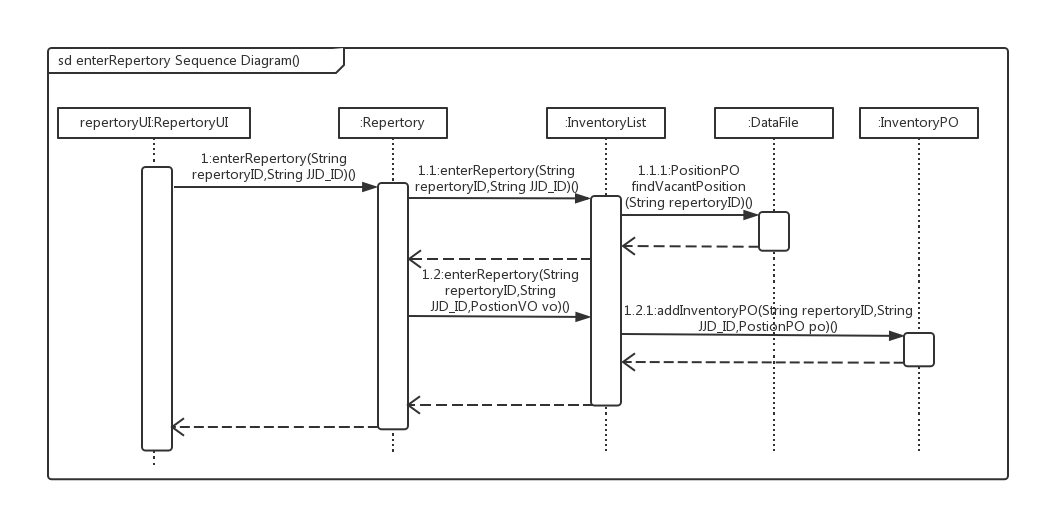
### 1.5.2 业务逻辑层的动态模型

图5-2表明了快递物流系统中，当仓库管理员开始库存信息初始化任务后，仓库管理业务逻辑处理的相关对象的协作。



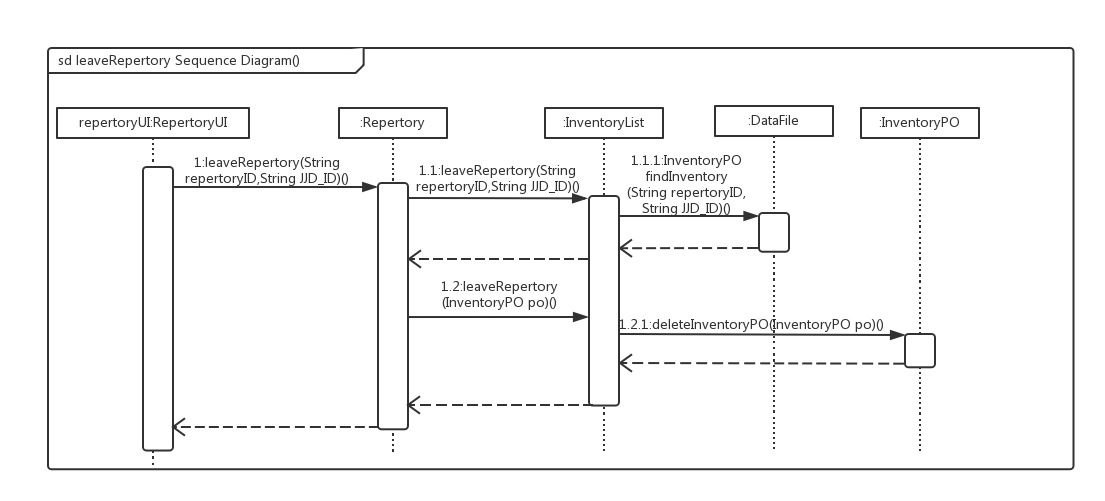
**图5-2 库存信息初始化的顺序图**

图5-3表明了快递物流系统中，当仓库管理员开始入库任务后，仓库管理业务逻辑处理的相关对象的协作。



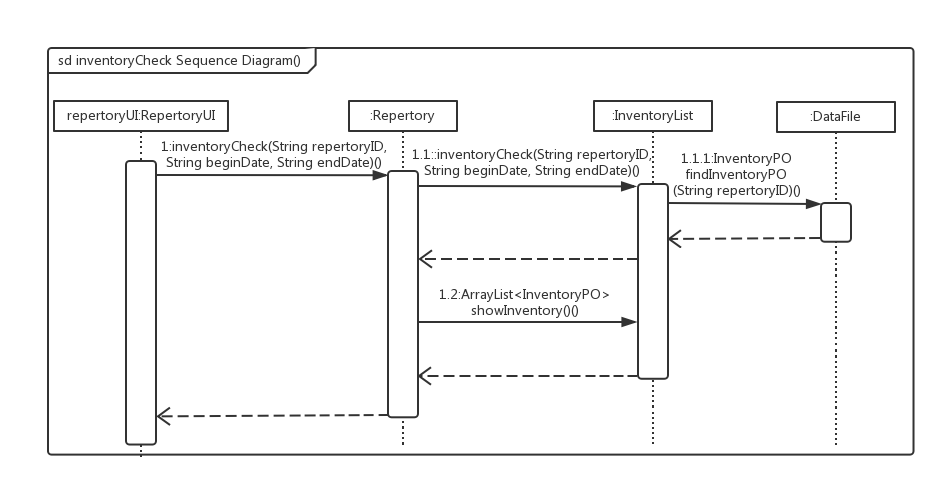
**图5-3 入库的顺序图**

图5-4表明了快递物流系统中，当仓库管理员开始出库任务后，仓库管理业务逻辑处理的相关对象的协作。



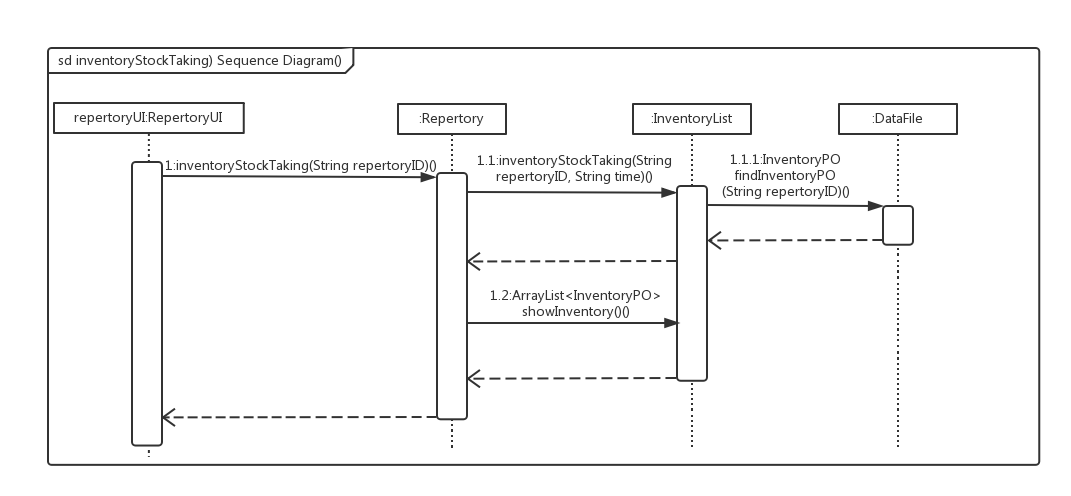
**图5-4 出库的顺序图**

图5-5表明了快递物流系统中，当仓库管理员开始库存查看任务后，仓库管理业务逻辑处理的相关对象的协作。



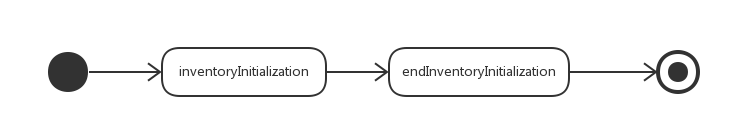
**图5-5 库存查看的顺序图**

图5-6表明了快递物流系统中，当仓库管理员开始库存盘点任务后，仓库管理业务逻辑处理的相关对象的协作。

****

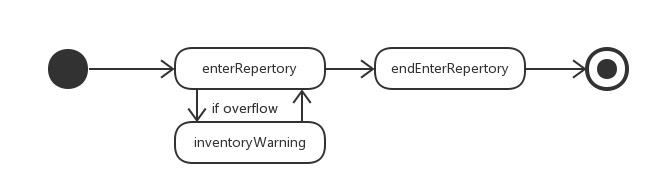
**图5-6 库存盘点的顺序图**

如图5-7所示的状态图描述了RepertoryBL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inventoryInitialization方法被UI调用，RepertoryBL进入inventoryInitialization状态；仓库管理员输入最大容量和警戒比例后，结束库存信息初始化进入endInventoryInitialization状态。



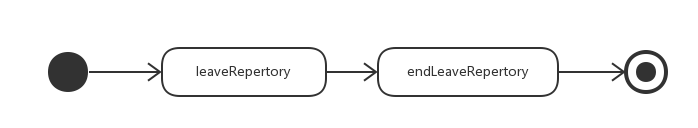
**图5-7 库存信息初始化的状态图**

如图5-8所示的状态图描述了RepertoryBL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enterRepertory方法被UI调用，RepertoryBL进入enterRepertory状态；当仓库管理员选择分区后，系统根据被选择分区是否超过警戒比例进入inventoryWarning状态要求再次确认入库，或者endEnterRepertory状态完成入库操作。



**图5-8 入库的状态图**

如图5-9所示的状态图描述了RepertoryBL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leaveRepertory方法被UI调用，RepertoryBL进入leaveRepertory状态；仓库管理员确认后，最后进入endEnterRepertory状态完成出库操作。



**图5-9 出库的状态图**

如图5-10所示的状态图描述了RepertoryBL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inventoryCheck方法被UI调用，RepertoryBL进入inventoryCheck状态；最后通过endInventoryCheck方法确定查看结束，进入endShow状态。



**图5-10 库存查看的状态图**

如图5-11所示的状态图描述了RepertoryBL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inventoryStockTaking方法被UI调用，RepertoryBL进入inventoryStockTaking状态；最后通过endInventoryStockTaking方法确定查看结束，进入endShow状态。

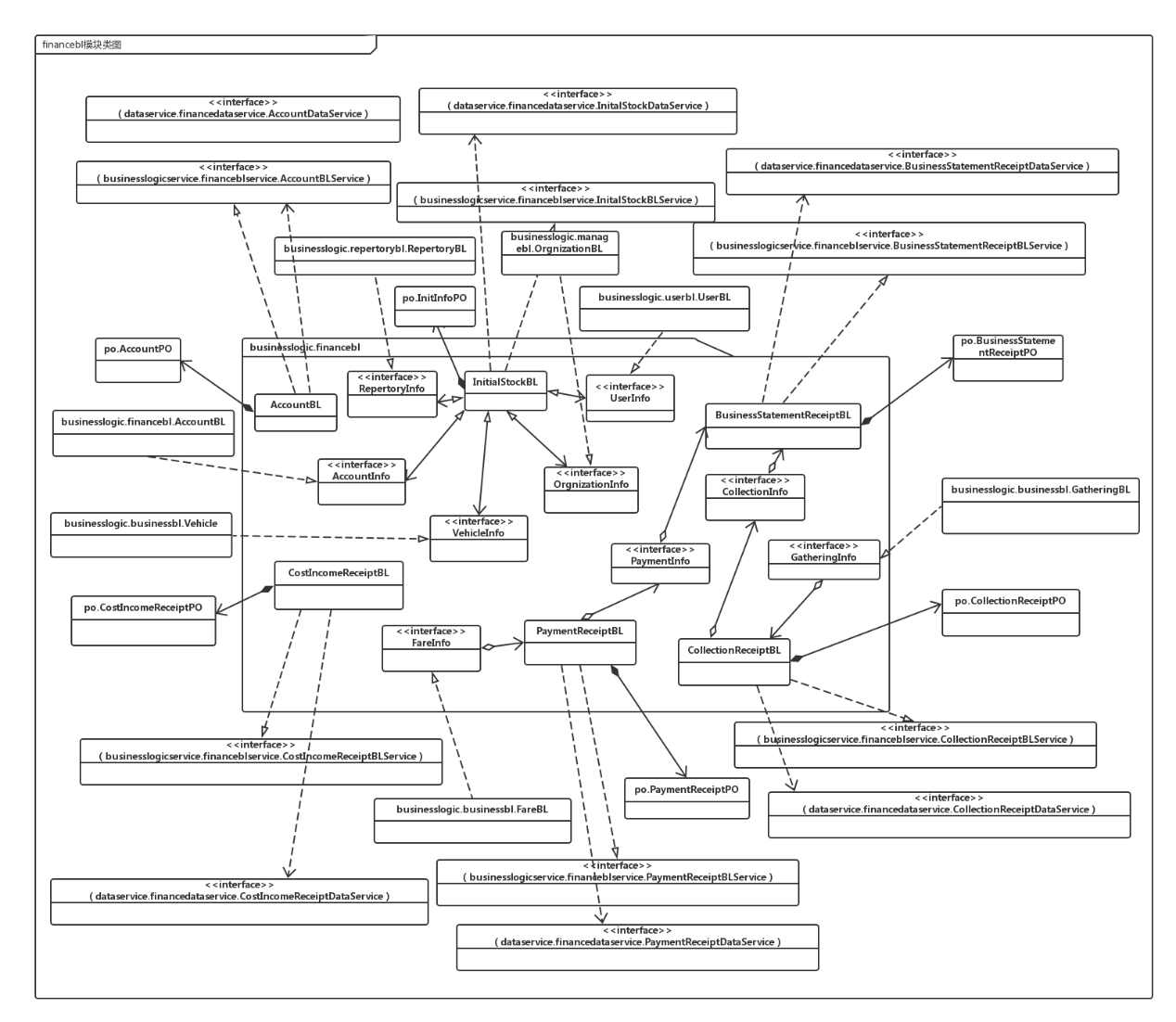
****

**图5-11 库存盘点的状态图**

## 1.6 financebl模块

### 1.6.1 模块整体结构

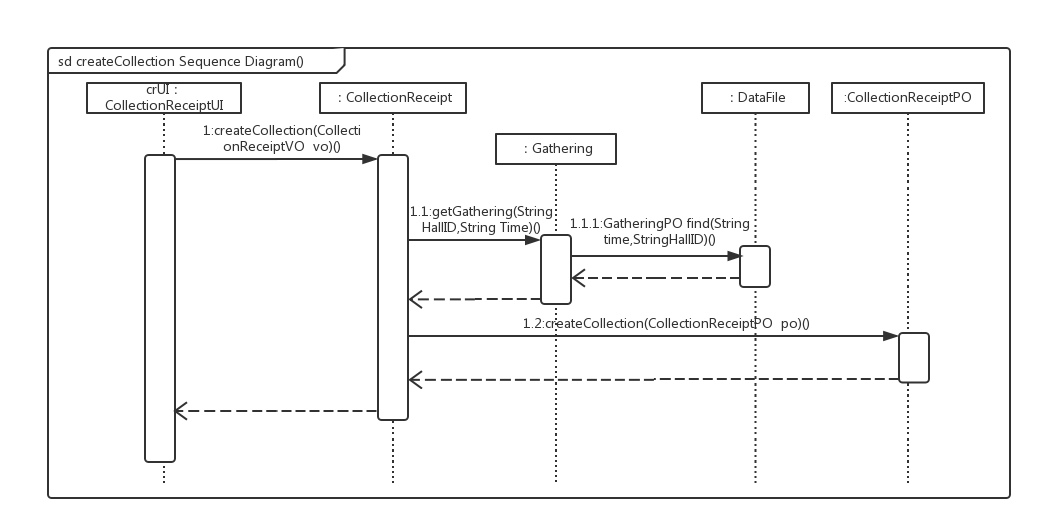
根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们添加businesslogicservice.financeblservice.CollectionReceiptBLService，businesslogicservice.financeblservice.PaymentReceiptBLService，businesslogicservice.financeblservice.CostIncomeReceiptBLService，businesslogicservice.financeblservice.BusinessStatementReceiptBLService，businesslogicservice.financeblservice.InitalStockBLService和businesslogicservice.financeblservice.AccountBLService接口。因为业务逻辑处理较少，根据专家模式没有添加controller类。业务逻辑层和数据层之间添加dataservice.financedataservice. CollectionReceiptDataService，dataservice.financedataservice. PaymentReceiptDataService，dataservice.financedataservice. CostIncomeReceiptDataService，dataservice.financedataservice. BusinessStatementReceiptDataService，dataservice.financedataservice. InitalStockDataService，dataservice.financedataservice. AccountDataService接口。CollectionReceiptPO,PaymentReceiptPO,BusinessStatementReceiptPO，CostIncomeReceiptPO，AccountPO和InitInfoPO是作为财务管理的持久化对象被添加到设计模型中去的。



**图6-1 financebl模块各个类的设计**

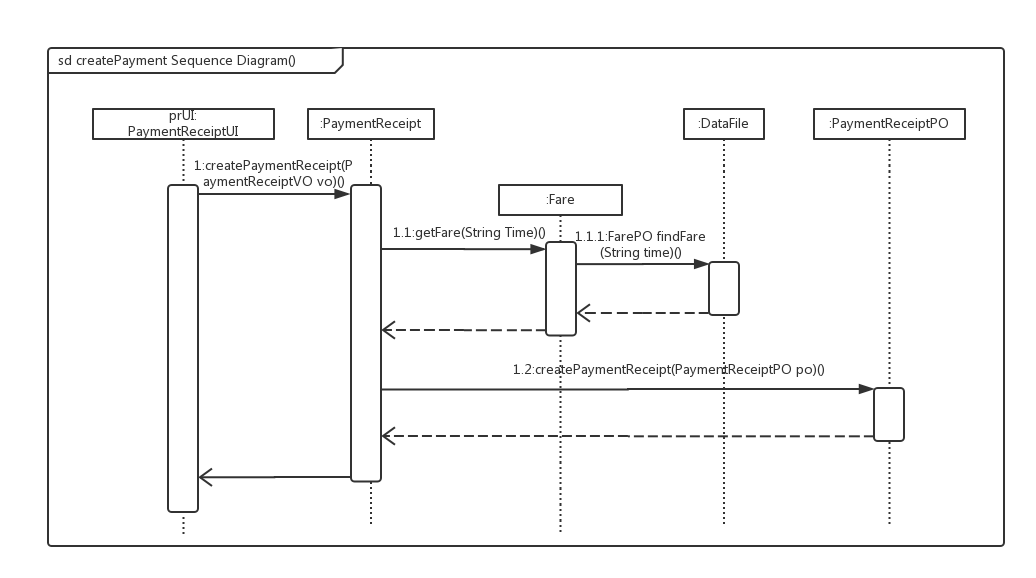
### 1.6.2 业务逻辑层的动态模型

图6-2表示了快递物流系统中，当财务人员进行结算管理时，业务逻辑层处理的相关对象之间的协作。



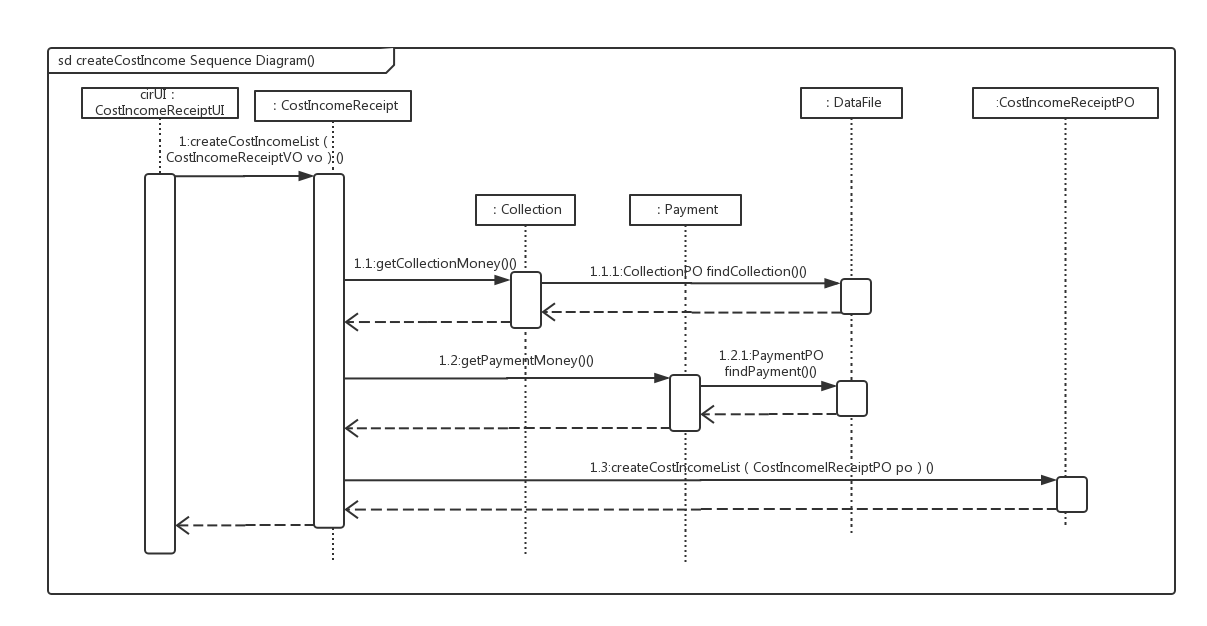
**图6-2 结算管理的顺序图**

图6-3表示了快递物流系统中，当财务人员进行新建付款单时，业务逻辑层处理的相关对象之间的协作。



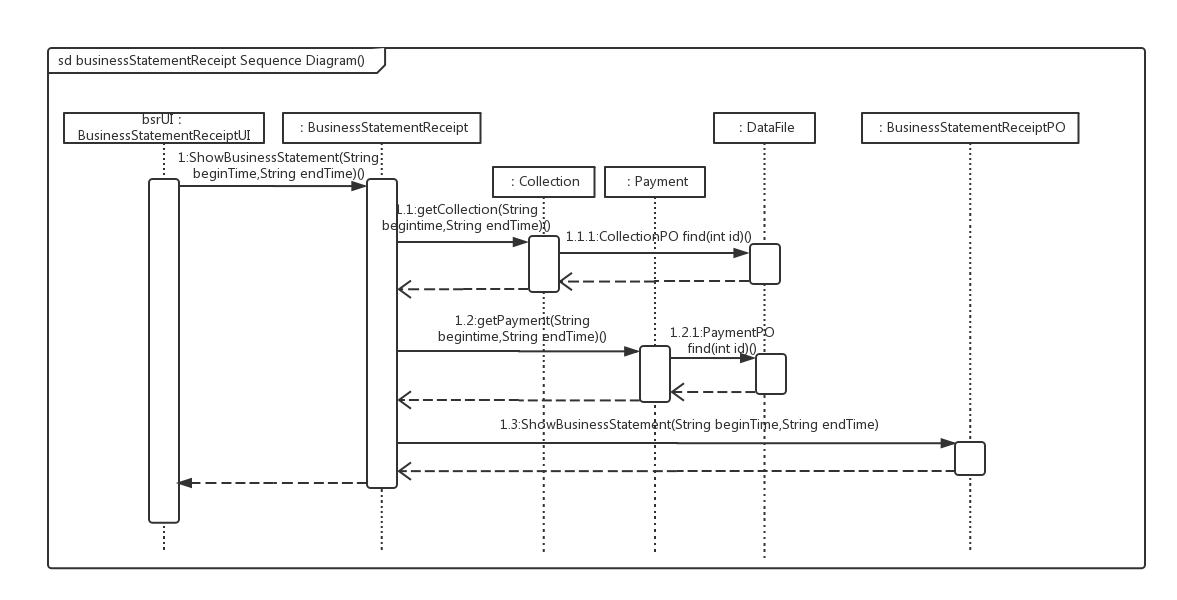
**图6-3 新建付款单的顺序图**

图6-4表示了快递物流系统中，当财务人员进行新建成本收益表时，业务逻辑层处理的相关对象之间的协作。



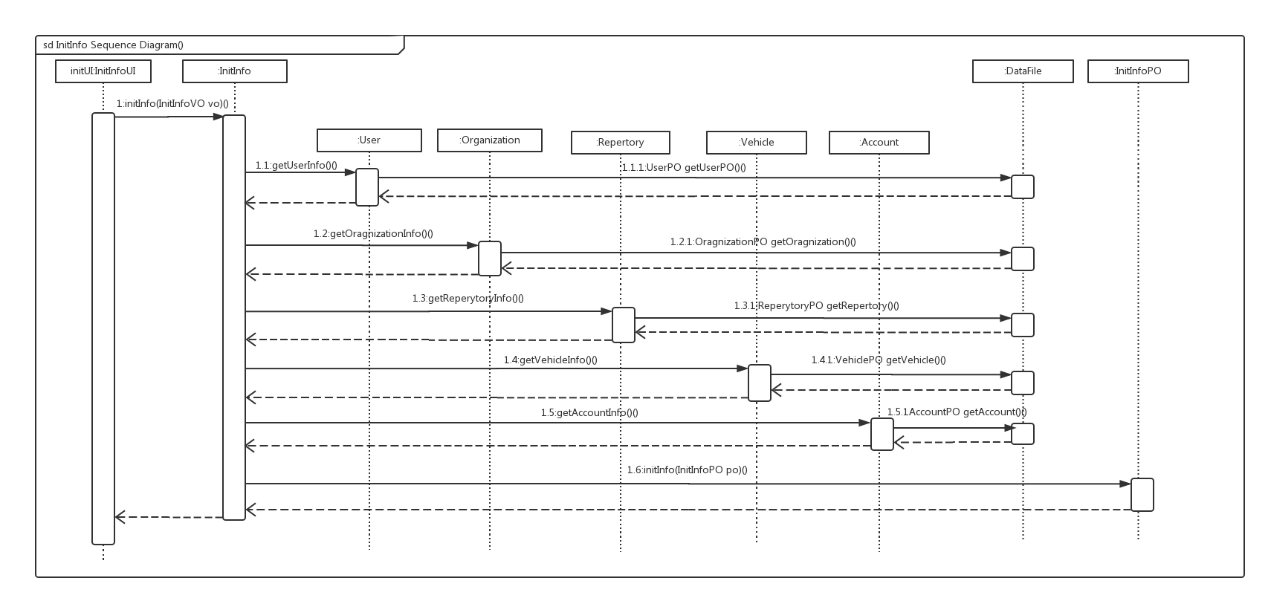
**图6-4 新建成本收益表的顺序图**

图6-5表示了快递物流系统中，当财务人员进行查看经营情况表时，业务逻辑层处理的相关对象之间的协作。



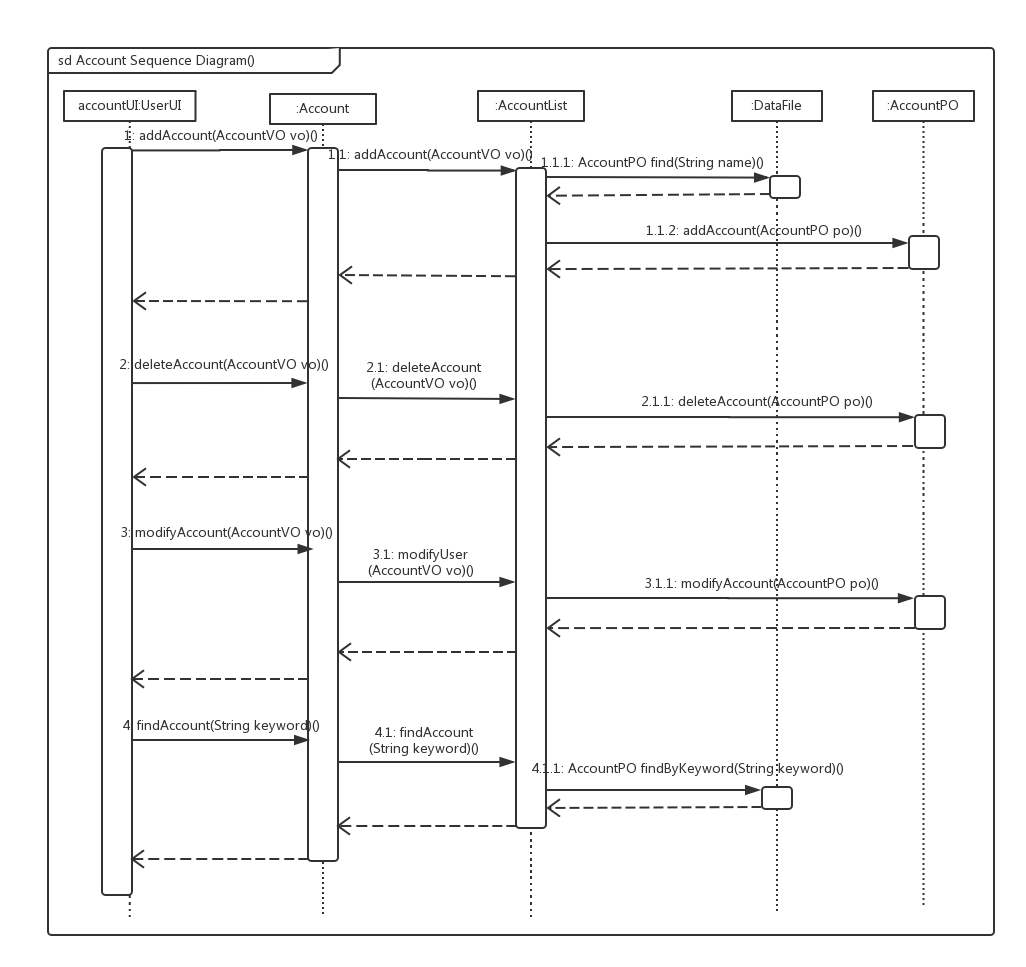
**图6-5 查看经营情况表的顺序图**

图6-6表示了快递物流系统中，当财务人员进行期初建账时，业务逻辑层处理的相关对象之间的协作。



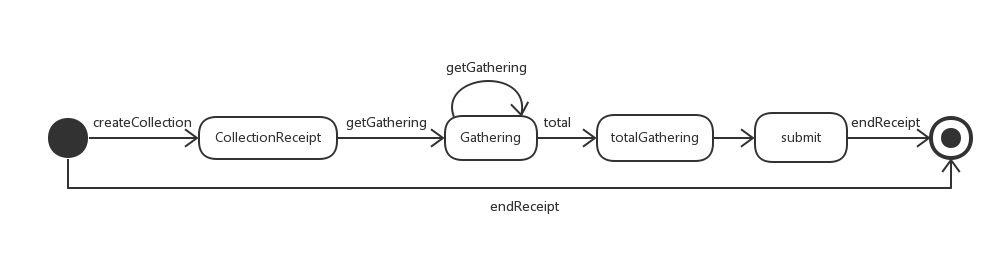
**图6-6 期初建账的顺序图**

图6-7表示了快递物流系统中，当财务人员进行账户管理时，业务逻辑层处理的相关对象之间的协作。



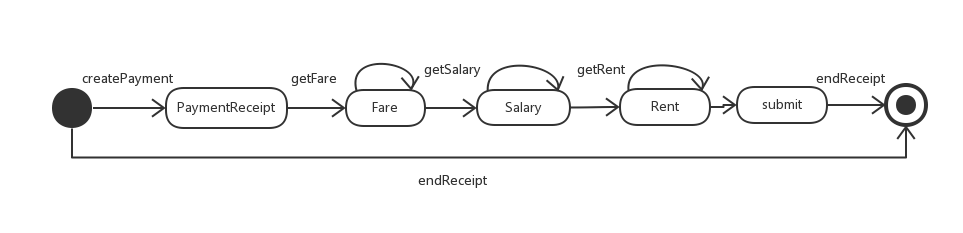
**图6-7 账户管理的顺序图**

如图6-8所示，CollectionReceipt对象状态图描述了CollectionReceipt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createCollection方法被 UI调用，进入CollectionReceipt状态；之后通过获取收款单进入Gathering状态。随后，通过计算收款单金额之和进入totalGathering状态；随着submit方法被UI调用，CollectionReceipt进入endReceipt状态。



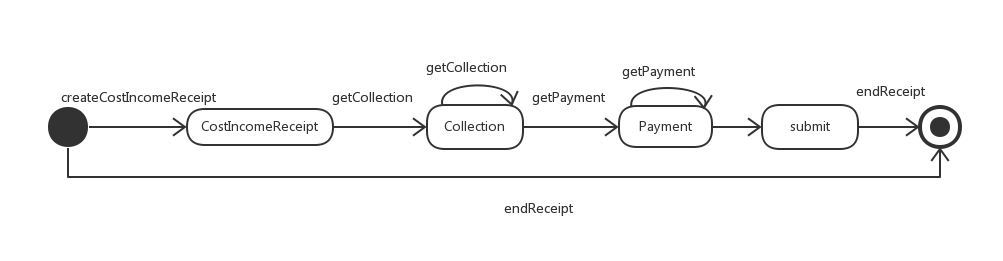
**图6-8 结算管理的状态图**

如图6-9所示，PaymentReceipt对象状态图描述了PaymentReceipt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createPayment方法被 UI调用，进入PaymentReceipt状态；之后通过获取运费单进入Fare状态，通过输入薪水进入Salary状态，通过输入租金进入Rent状态；随着submit方法被UI调用，PaymentReceipt进入endReceipt状态。



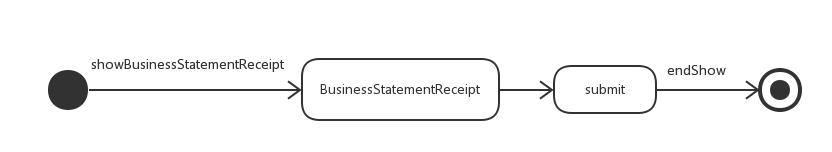
**图6-9 新建付款单的状态图**

如图6-10所示，CostIncomeReceipt对象状态图描述了CostIncomeReceipt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createCostIncome方法被 UI调用，进入CostIncomeReceipt状态；之后通过获取入款单进入Collection状态，通过获取付款单进入Payment状态；随着submit方法被UI调用，CostIncomeReceipt进入endReceipt状态。

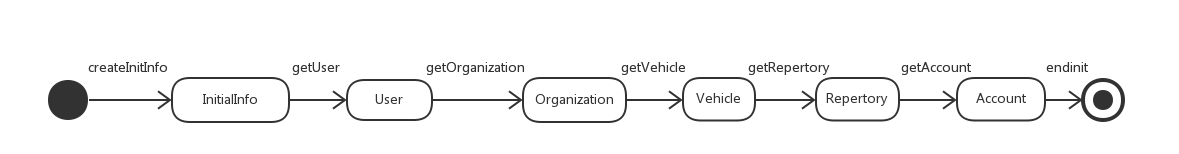


**图6-10 新建成本收益表的状态图**

如图6-11所示，BusinessStatementReceipt对象状态图描述了BusinessStatementReceipt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showBusinessStatementReceipt被 UI调用，进入BusinessStatementReceipt状态；随着submit方法被UI调用，BusinessStatementReceipt进入endShow状态。

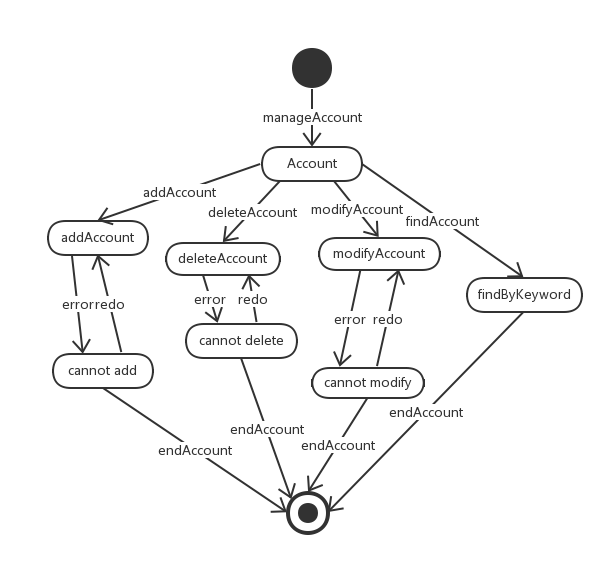


**图6-11 查看经营情况表的状态图**

如图6-12所示, InitInfo对象状态图描述了InitInfo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createInitInfo方法被 UI调用，进入InitialInfo状态；之后通过添加用户信息进入User状态，通过添加机构信息进入Organization状态，通过添加车辆信息进入Vehicle状态，通过添加库存信息进入Repertory状态，通过添加账户信息进入Account状态；最后进入endInit状态。

**图6-12 期初建账的状态图**

如图6-13所示，Account对象状态图描述了Account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manageAccount方法被 UI调用，Account进入Account状态；之后通过添加账户进入addAccount状态，通过删除账户进入deleteAccount状态，通过修改账户进入modifyAccount状态，通过查询账户进入findByKeyword状态；最后进入endAccount状态。

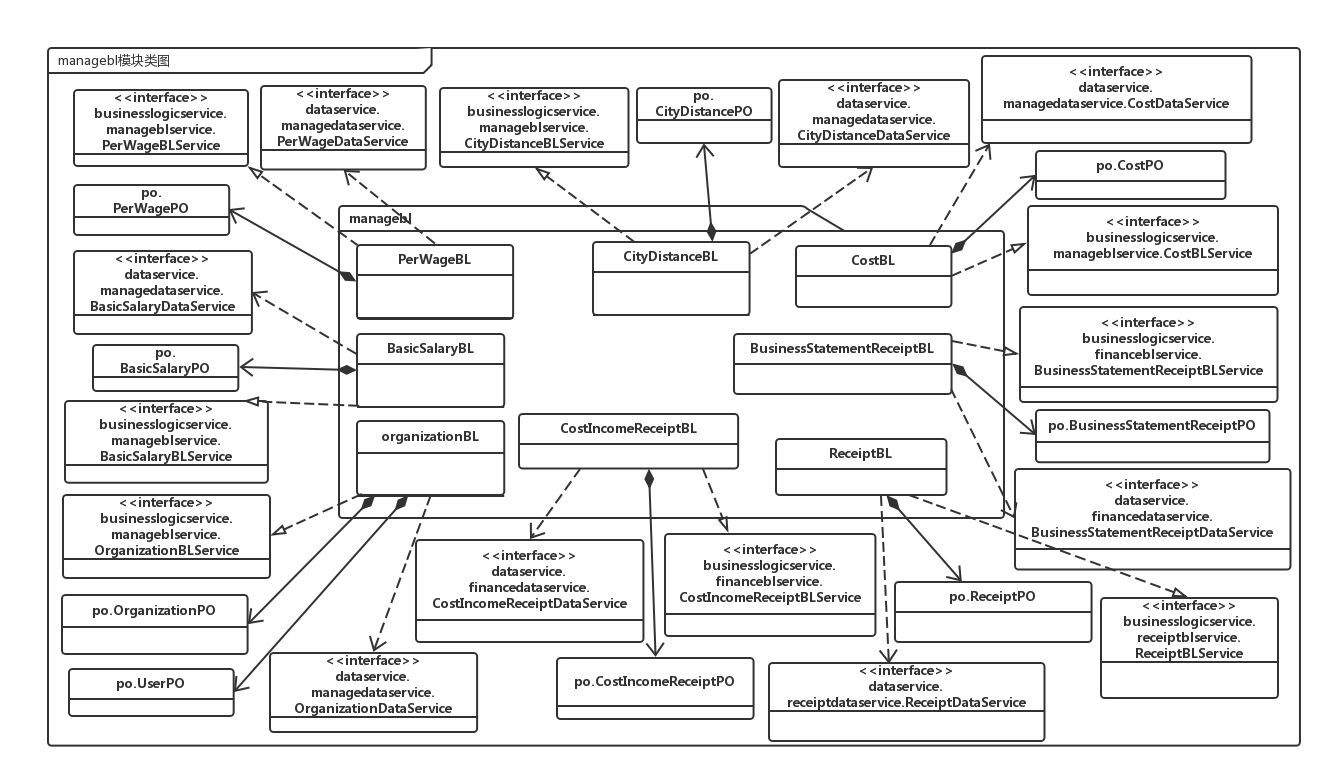


**图6-13 账户管理的状态图**

## 1.7 managebl模块

### 1.7.1 模块整体结构

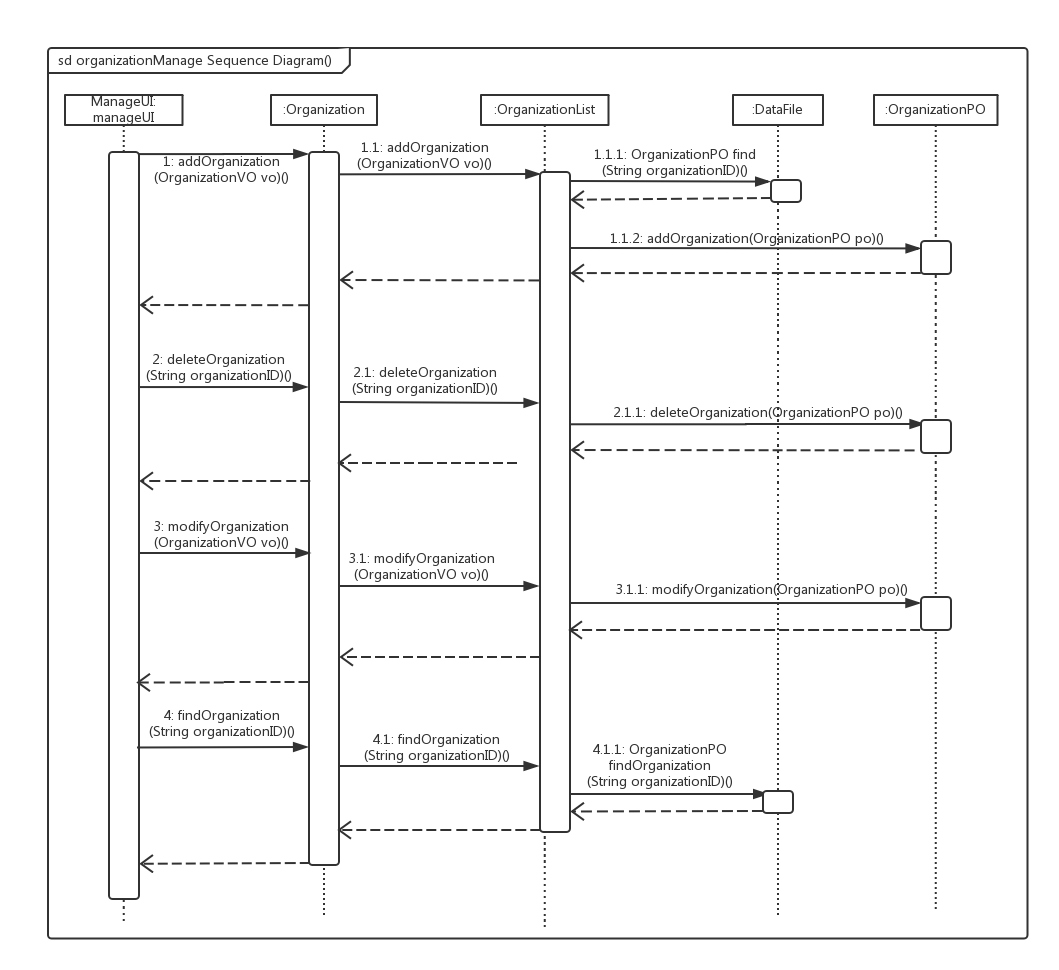
根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们添加businesslogicservice.manageblservice.OrganizationBLService、businesslogicservice.manageblservice.BasicSalaryBLService、businesslogicservice.manageblservice.CityDistanceBLService等接口。业务逻辑层和数据层之间添加dataservice.managedataservice.OrganizationDataService、dataservice.managedataservice.BasicSalaryDataService、ataservice.managedataservice.CityDistanceDataService等接口。因为业务逻辑处理较少，根据专家模式没有添加Controller类。OrganizationPO等PO对象是作为总经理管理记录的持久化对象被添加到设计模型中去的。

managebl模块的设计如图7-1所示。

**图7-1 managebl模块各个类的设计**

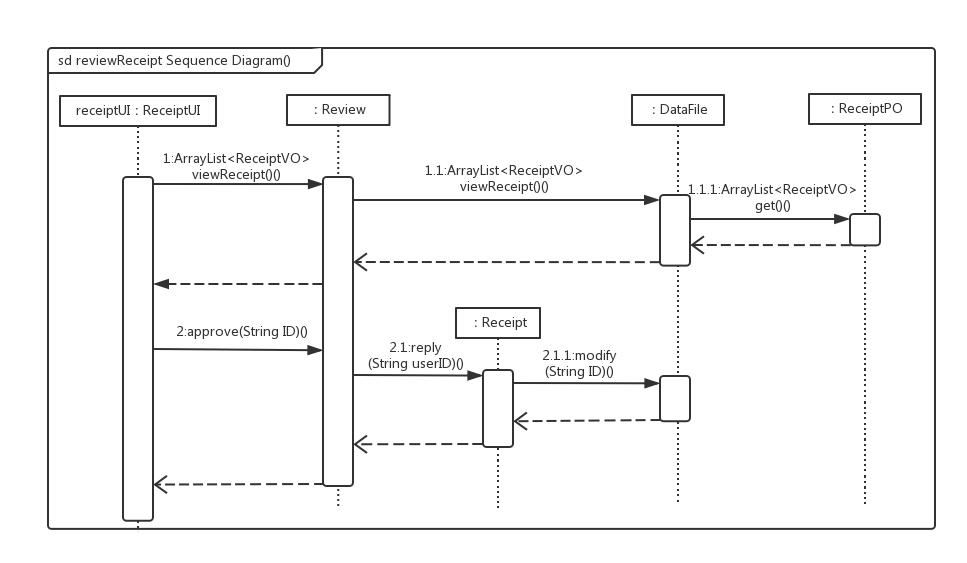
### 1.7.2 业务逻辑层的动态模型

图7-2表明了快递物流系统中，当总经理开始机构管理任务后，机构管理业务逻辑处理的相关对象的协作。



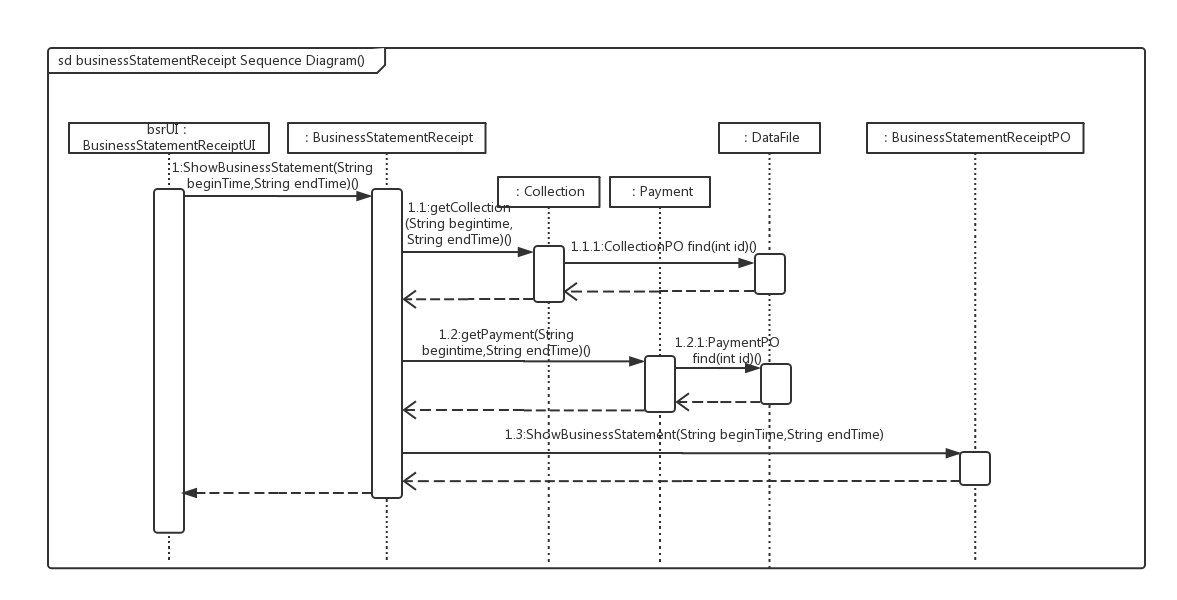
**图7-2 机构管理的顺序图**

图7-3表明了快递物流系统中，当总经理开始单据审批任务后，单据审批业务逻辑处理的相关对象的协作。



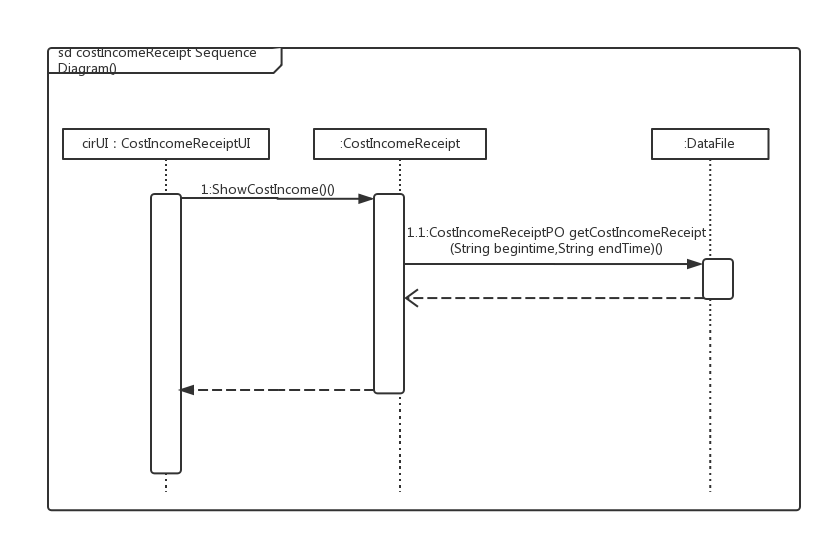
**图7-3 单据审批的顺序图**

图7-4表明了快递物流系统中，当总经理开始查看经营情况表任务后，经营情况表业务逻辑处理的相关对象的协作。



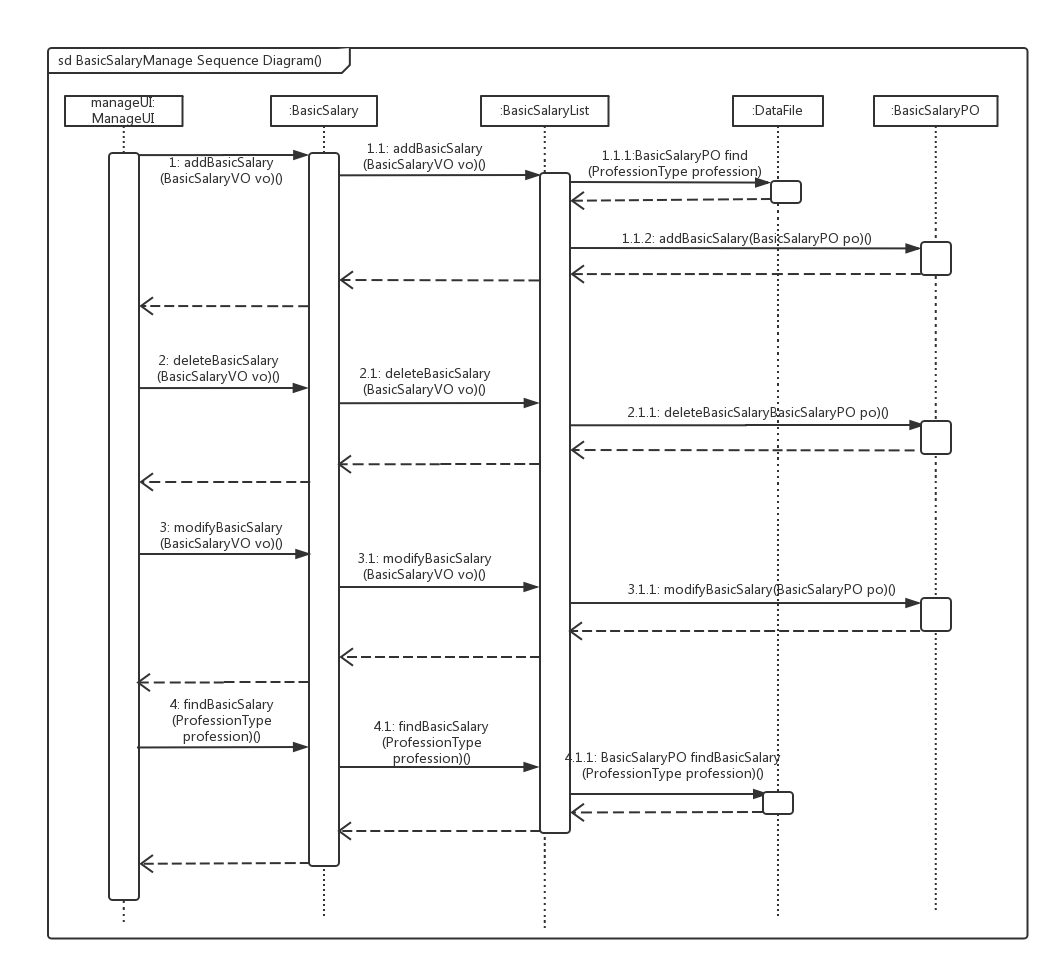
**图7-4 查看经营情况表的顺序图**

图7-5表明了快递物流系统中，当总经理开始查看成本收益表任务后，成本收益表业务逻辑处理的相关对象的协作。



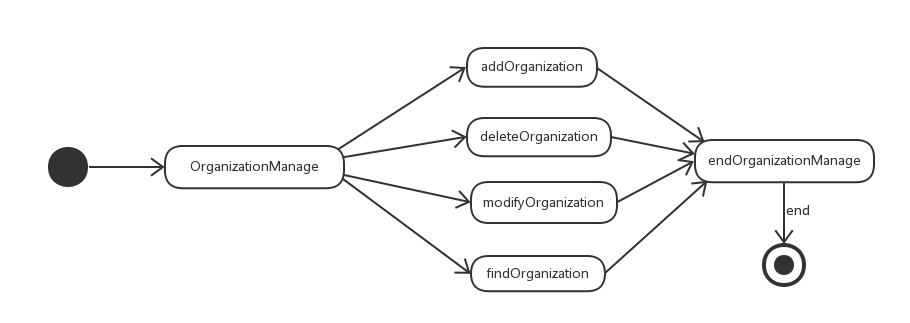
**图7-5 查看成本收益表的顺序图**

图7-6表明了快递物流系统中，当总经理开始查看基础数据设置任务后，基础数据设置业务逻辑处理的相关对象的协作（以基础月薪管理为例）。

****

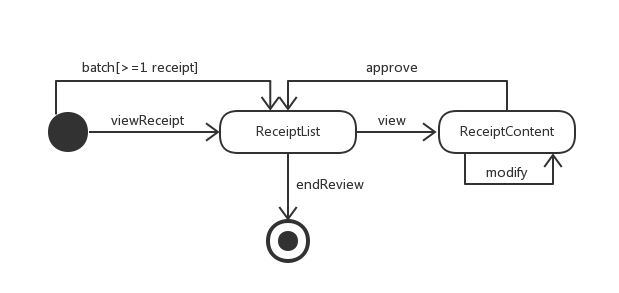
**图7-6 基础月薪管理的顺序图**

如图7-7所示的状态图描述了OrganizationBL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OrganizationManage方法被UI调用，OrganizationBL进入OrganizationManage状态；之后可以选择添加机构进入addOrganization状态，也可以选择删除机构进入deleteOrganization状态，同理modifyOrganization状态和findOrganization状态。最后结束机构管理进入endOrganizationManage状态。



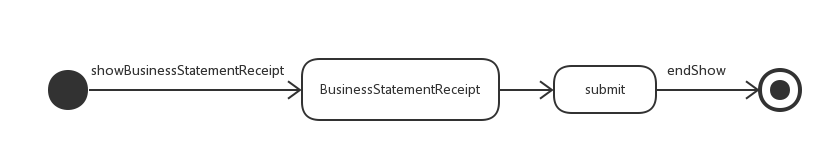
**图7-7 机构管理的状态图**

如图7-8所示的状态图描述了ReceiptBL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viewReceipt方法被UI调用，ReceiptBL进入ReceiptList状态；之后可以选择详细查看进入ReceiptContent状态，也可以选择批量审批进入ReceiptList状态。最后结束单据审批进入endReview状态。



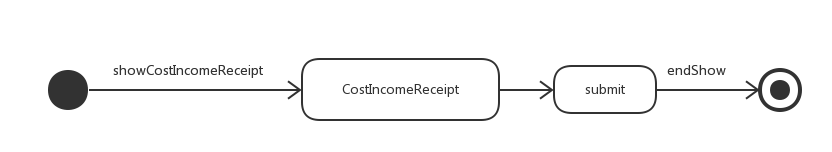
**图7-8 单据审批的状态图**

如图7-9所示的状态图描述了BusinessStatementReceiptBL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showBusinessStatementReceipt方法被UI调用，BusinessStatementReceiptBL进入BusinessStatementReceipt状态；最后通过submit方法确定查看结束，进入endShow状态。



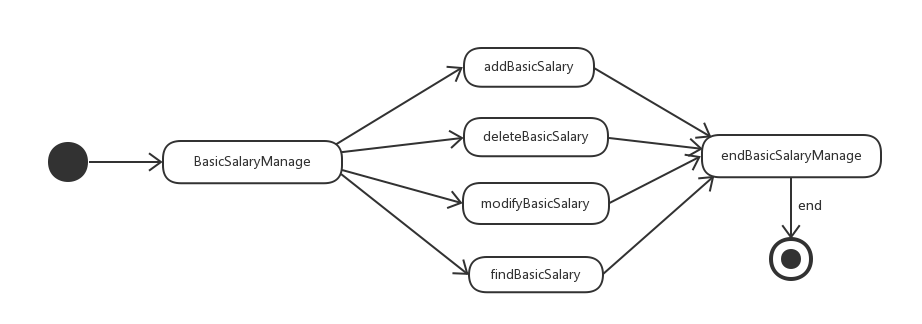
**图7-9 查看经营情况表的状态图**

如图7-10所示的状态图描述了CostIncomeReceiptBL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showCostIncomeReceipt方法被UI调用，CostIncomeReceiptBL进入CostIncomeReceipt状态；最后通过submit方法确定查看结束，进入endShow状态。



**图7-10 查看成本收益表的状态图**

如图7-7所示的状态图描述了BasicSalaryBL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BasicSalaryManage方法被UI调用，BasicSalaryBL进入BasicSalaryManage状态；之后可以选择添加基础月薪进入addBasicSalary状态，也可以选择删除基础月薪进入deleteBasicSalary状态，同理modifyBasicSalary状态和findBasicSalary状态。最后结束基础月薪管理进入endBasicSalaryManage状态。

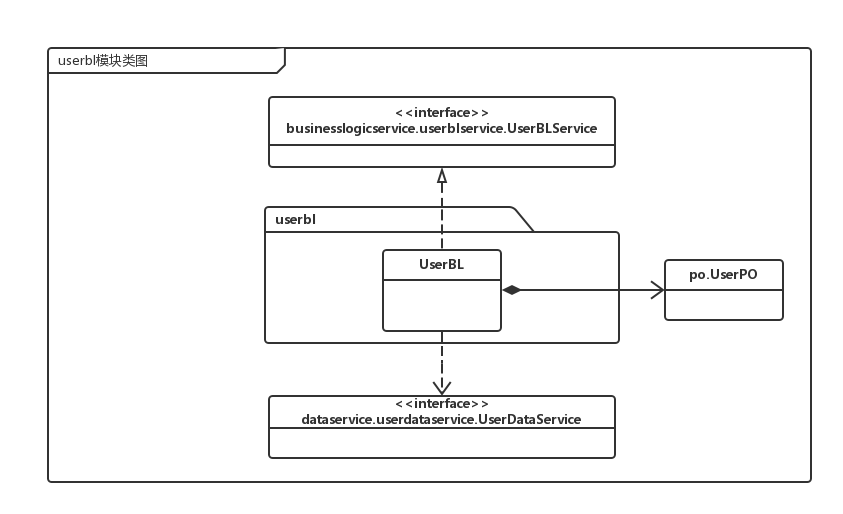
****

**图7-11 基础月薪管理的状态图**

## 1.8 userbl模块

### 1.8.1 模块整体结构

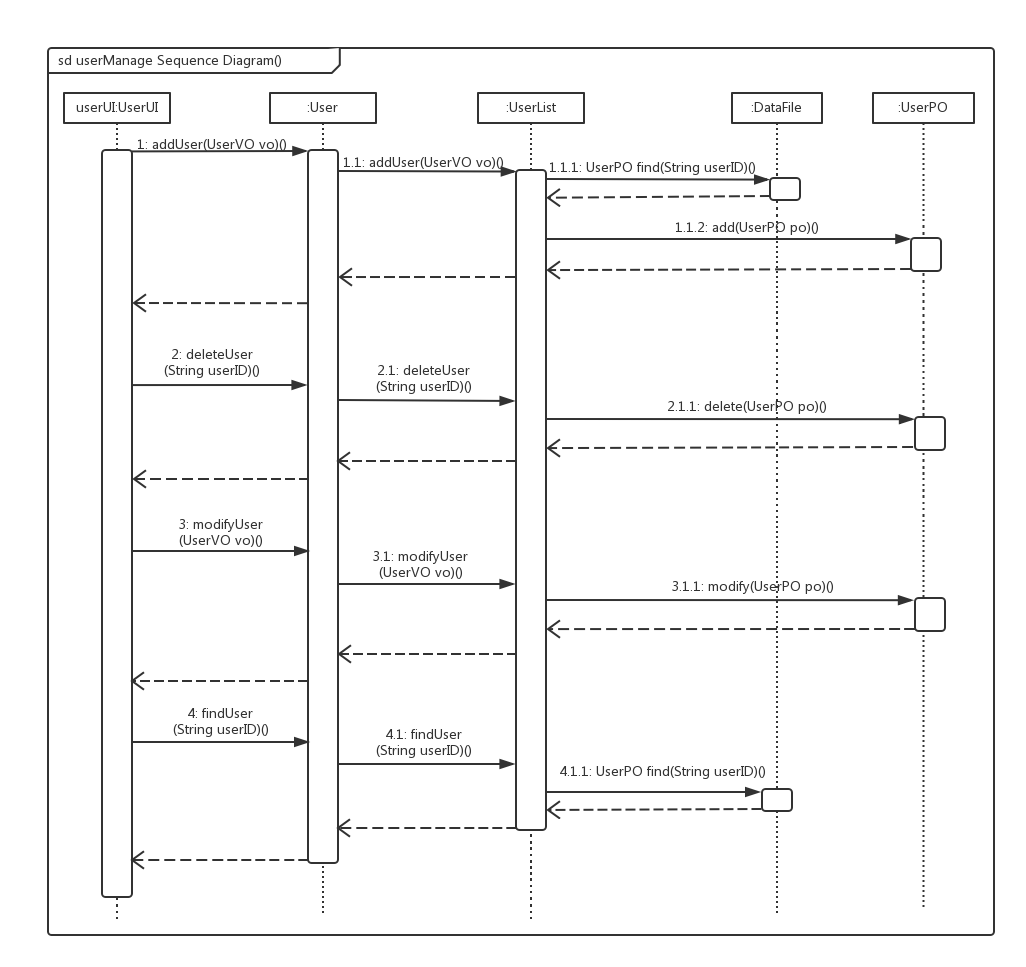
根据体系结构的设计，我们将系统分为展示层，业务逻辑层，数据层。每一层之间为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们添加userblservice.UserBLService接口。业务逻辑层和数据层之间添加uesrdataservice.UserDataService接口。ReceiptPO是作为所有单据父类的持久化对象被添加到设计模型中去的。ReceiptBL模块的设计如图8-1所示。



**图8-1 userbl模块各个类的设计**

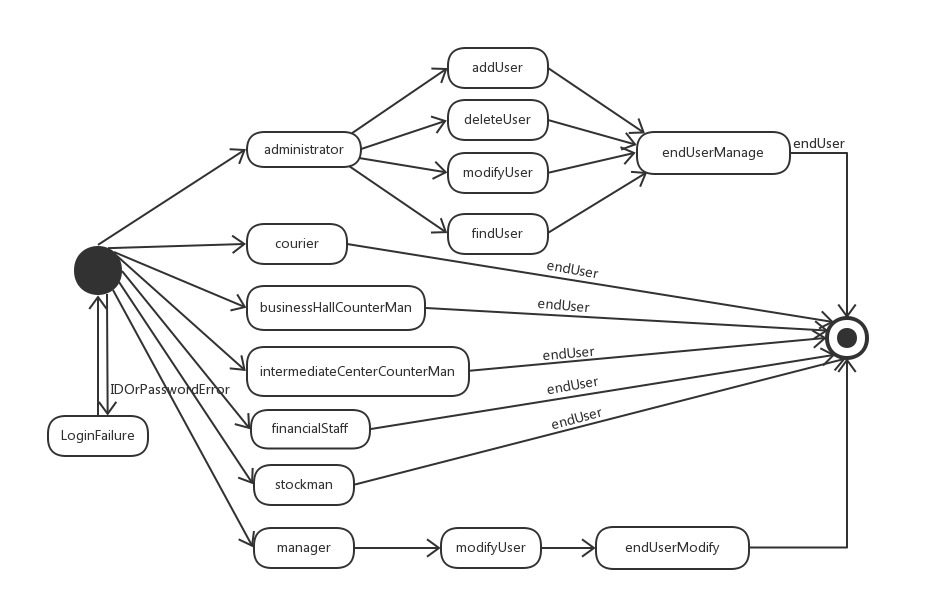
### 1.8.2 业务逻辑层的动态模型

图8-2表明了ELS快递物流系统中，当用户选择登录或管理员选择用户增删改查的基本操作之后，用户业务逻辑处理的相关对象之间的协作。



**图8-2 用户登录及管理员增删改查的顺序图**

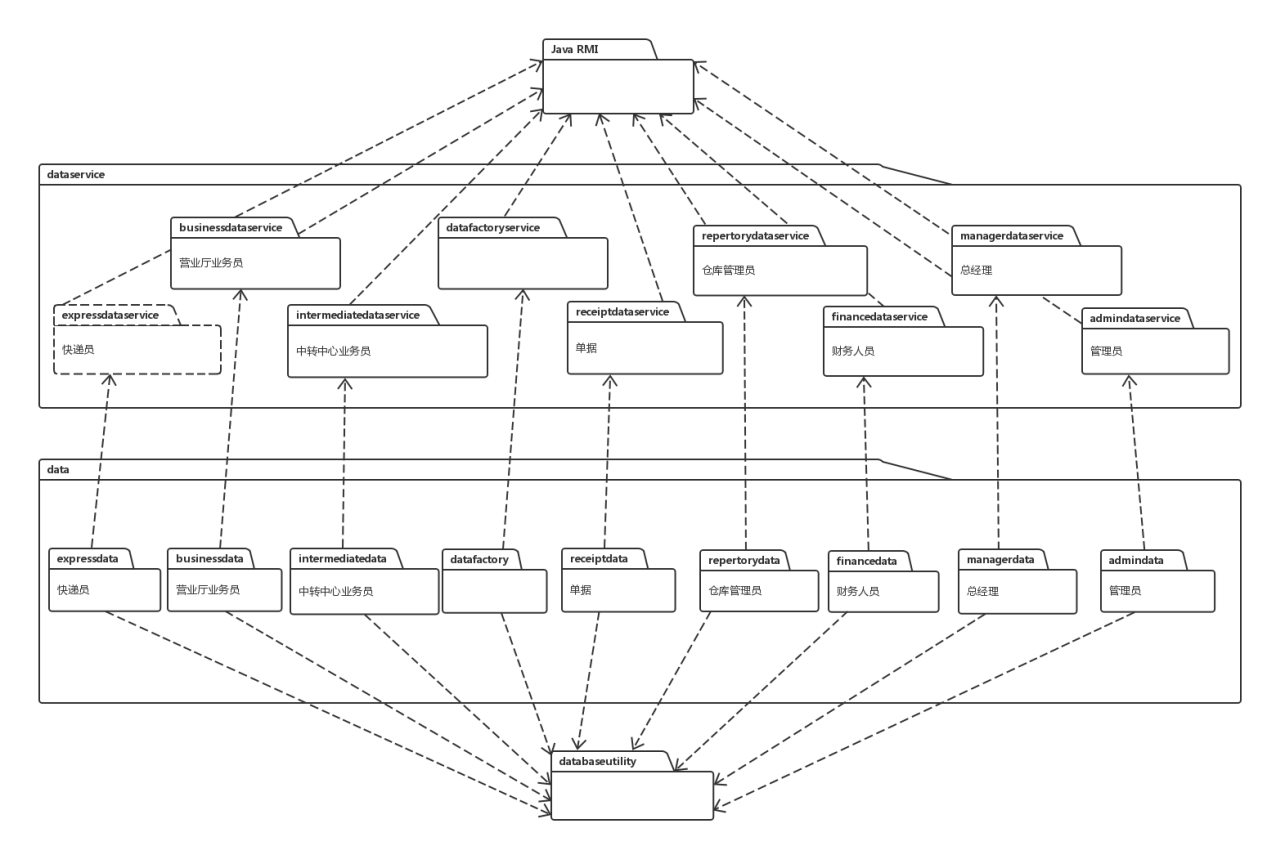
如图8-3所示，用户登录及管理员增删改查对象状态图描述了User对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着establishUser方法被 UI调用，establishUser根据不同的指令进入不同状态。



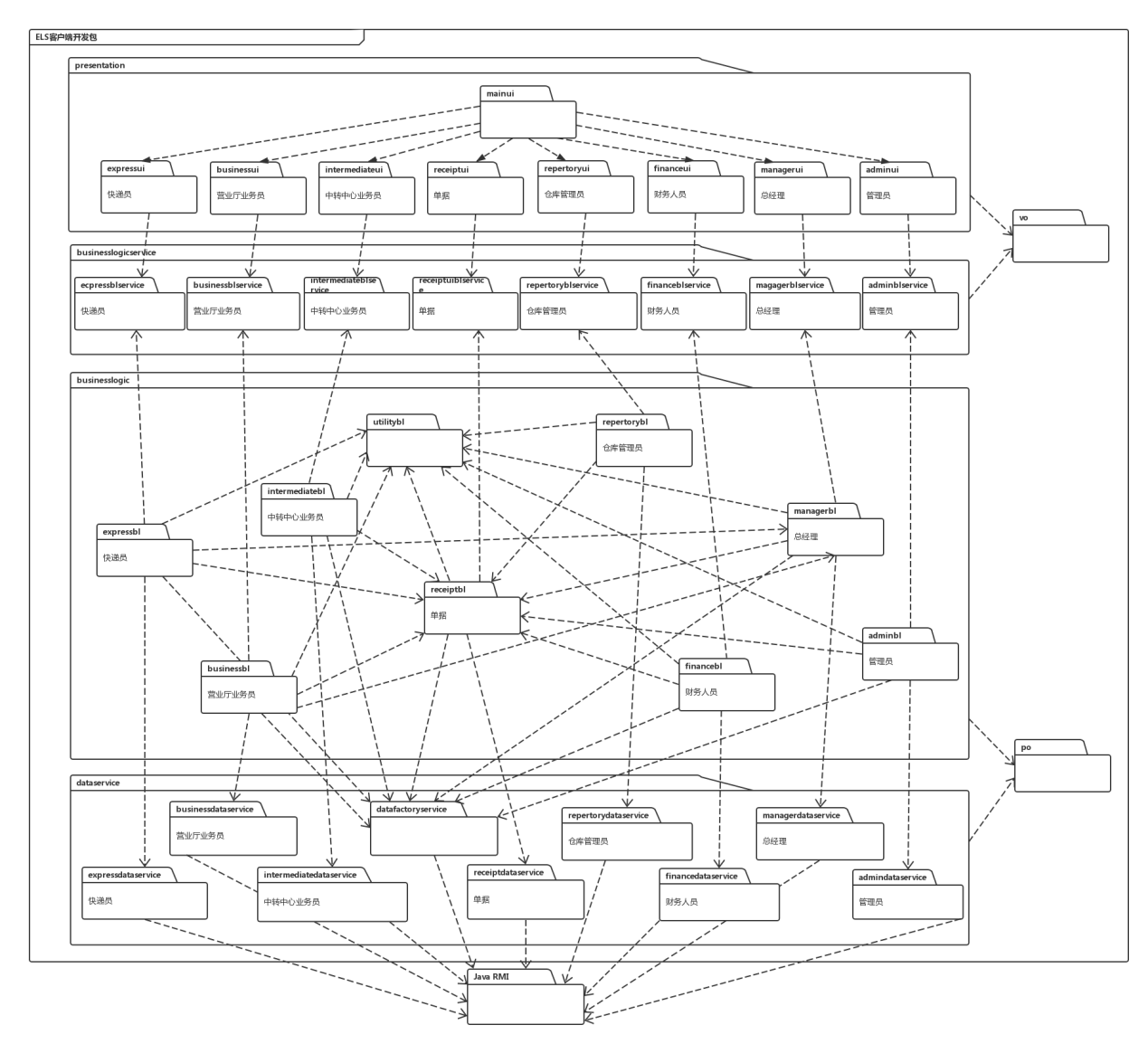
**图8-3 用户登录及管理员增删改查的状态图**

# 依赖视角

图1,2是服务器端和客户端各自的包之间的依赖关系。



**图1 ELS服务器端开发包图**



**图2 ELS客户器端开发包图**