* 1. 界面层分解
     1. orderNewUI模块

1. 模块概述

orderNewUI模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

orderNewUI模块的职责及接口参见软件系统结构描述文档5.2.2表orderNewUI模块的接口规范。

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LogicService. OrderNewLogic接口。OrderVO是各种单据作为单据显示的对象被添加到设计模型中去的。 ExpressPanel，ReceivePanel，PaymentPanel，HallLoadingPanel，IncomePanel，RecipientPanel，SendPanel，ArrivePanel，CenterLoadingPanel，TransitPanel，StockInPanel，StockOutPanel用于显示对应的单据信息。PriceCount接口用于更新价格信息。

OrderNewUI模块的设计如图所示



OrderNewUI模块各个类的职责如下表所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| ExpressPanel | 负责显示快递单填写界面 |
| ReceivePanel | 负责显示收件单填写界面 |
| PaymentPanel | 负责显示付款单填写界面 |
| HallLoadingPanel | 负责显示营业厅装车单填写界面 |
| IncomePanel | 负责显示收款单填写界面 |
| RecipientPanel | 负责显示接收单填写界面 |
| SendPanel | 负责显示派件单填写界面 |
| ArrivePanel | 负责显示到达单填写界面 |
| CenterLoadingPanel | 负责显示中转中心装车填写界面 |
| TransitPanel | 负责显示中转单填写界面 |
| StockInPanel | 负责显示入库单填写界面 |
| StockOutPanel | 负责显示出库单填写界面 |
| PriceCount | 负责计算价格的接口 |

1. 模块内部接口规范

ArrivePanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ArrivePanel(JFrame fr,String name,String orgId,String userId,String org) | 语法 | public ArrivePanel(JFrame fr,String name,String orgId,String userId,String org) |
| 前置条件 | 用户选择生成到达单 |
| 后置条件 | 显示生成到达单界面 |
| ArrivePanel(JFrame fr,ArriveVO vo) | 语法 | public ArrivePanel(JFrame fr,ArriveVO vo) |
| 前置条件 | 用户选择查看到达单 |
| 后置条件 | 显示查看到达单界面 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| ArriveOrderLSer.create(ArriveVO order) | 负责提供新建单据的逻辑接口 | |
| ArriveOrderLSer.getArriveOrders(String org) | 负责提供获取可接受中转单和装车单的逻辑接口 | |
| ArriveOrderLSer.getLocalHallsAndAllCenter(String orgId) | 负责提供获取营业厅和中转中心的逻辑接口 | |

CenterLoadingPanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CenterLoadingPanel(JFrame fr,String name,String userId,String org,String orgId) | 语法 | public CenterLoadingPanel(JFrame fr,String name,String userId,String org,String orgId) |
| 前置条件 | 用户选择生成中转中心接收单 |
| 后置条件 | 显示生成中转中心接收单界面 |
| CenterLoadingPanel(JFrame fr,CenterLoadingVO vo) | 语法 | public CenterLoadingPanel(JFrame fr,CenterLoadingVO vo) |
| 前置条件 | 用户选择查看中转中心接收单 |
| 后置条件 | 显示查看中转中心接收单界面 |
| CenterLoadingPanel.getPrice() | 语法 | public void getPrice() |
| 前置条件 | 用户选择目的地，输入订单 |
| 后置条件 | 显示运费 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| CenterLoadingOrderLSer.create(CenterLoadingVO order) | 负责提供新建单据的逻辑接口 | |
| CenterLoadingOrderLSer.getHalls | 负责获取营业厅选项的逻辑接口 | |
| CenterLoadingOrderLSer.getPrice(String org1,String org2,List<String> expressList) | 负责获取装车单的费用 | |
| CenterLoadingOrderLSer.getNextId(String orgId) | 负责获取装车单的编号 | |
| CenterLoadingOrderLSer.getExpressList(String orgId) | 负责获取此装车单可装的快递编号 | |

ExpressPanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ExpressPanel(JFrame fr,String name,String userId,String orgId) | 语法 | ExpressPanel(JFrame fr,String name,String userId,String orgId) |
| 前置条件 | 用户选择生成快递单 |
| 后置条件 | 显示生成快递单界面 |
| ExpressPanel(JFrame fr,ExpressVO vo) | 语法 | ExpressPanel(JFrame fr,ExpressVO vo) |
| 前置条件 | 用户选择查看快递单 |
| 后置条件 | 显示查看快递单界面 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| ExpressOrderLSer.create(ExpressVO order) | 负责提供新建单据的逻辑接口 | |
| ExpressOrderLSer.getPrice(String city1,String city2,Express expressKind,Parse pakKind,double weight) | 负责提供获取运费的逻辑接口 | |
| ExpressOrderLSer.getTime(String city1,String city2) | 负责提供获取所需时间的逻辑接口 | |

HallLoadingPanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| HallLoadingPanel(JFrame fr,String userName,String userId,String orgId,String org) | 语法 | public HallLoadingPanel(JFrame fr,String userName,String userId,String orgId,String org) |
| 前置条件 | 用户选择生成营业厅装车单 |
| 后置条件 | 显示生成营业厅装车单界面 |
| HallLoadingPanel(JFrame fr,HallLoadingVO vo) | 语法 | public HallLoadingPanel(JFrame fr,HallLoadingVO vo) |
| 前置条件 | 用户选择查看营业厅装车单 |
| 后置条件 | 显示查看营业厅装车单界面 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| HallLoadingOrderLSer.create(HallLoadingVO order) | 负责提供新建单据的逻辑接口 | |
| HallLoadingOrderLSer. getVans(String org) | 负责获取车辆选项的逻辑接口 | |
| HallLoadingOrderLSer. getPrice(String org1,String org2,List<String> expressList) | 负责获取运费的逻辑接口 | |
| HallLoadingOrderLSer. getLocalHallsAndAllCenter(String orgId) | 负责获取本市营业厅或中转中心的逻辑接口 | |
| HallLoadingOrderLSer. getNextId(String orgId) | 负责获取装车单编号的逻辑接口 | |
| HallLoadingOrderLSer. isLegal(String id) | 负责检查订单是否合法的逻辑接口 | |
| HallLoadingOrderLSer. getUnloadExpresses(String orgId) | 负责获取待撞车订单的逻辑接口 | |

IncomePanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| IncomePanel(JFrame fr,String orgId,String name,String userId) | 语法 | public IncomePanel(JFrame fr,String orgId,String name,String userId) |
| 前置条件 | 用户选择生成收款单 |
| 后置条件 | 将用户填写的VO数据传入逻辑层 |
| IncomePanel(JFrame fr,IncomeVO vo) | 语法 | public IncomePanel(JFrame fr,IncomeVO vo) |
| 前置条件 | 用户选择查看收款单 |
| 后置条件 | 显示对应的收款单 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| IncomeOrderLSer.Create(IncomeVO order) | 负责提供新建单据的逻辑接口 | |
| IncomeOrderLSer.getCouriers(String id) | 负责获取快递员选项的逻辑接口 | |
| IncomeOrderLSer.getAccount() | 负责获取账户选项的逻辑接口 | |
| IncomeOrderLSer.getNextId(String orgId) | 负责获取入款单编号的逻辑接口 | |

PaymentPanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PaymentPanel(JFrame fr,String name,String userId) | 语法 | public PaymentPanel(JFrame fr,String name,String userId) |
| 前置条件 | 用户选择生成付款单 |
| 后置条件 | 将用户填写的VO数据传入逻辑层 |
| PaymentPanel(JFrame fr,PaymentVO vo) | 语法 | public PaymentPanel(JFrame fr,PaymentVO vo) |
| 前置条件 | 用户查看生成付款单 |
| 后置条件 | 显示对应入款单信息 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| PaymentOrderLSer.Create(PaymentVO order) | 负责提供新建单据的逻辑接口 | |
| PaymentOrderLSer. getAccount() | 负责获取账户的逻辑接口 | |
| PaymentOrderLSer. getNextId() | 负责获取付款单编号的逻辑接口 | |

ReceivePanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ReceivePanel(JFrame fr,String org,String name) | 语法 | ReceivePanel(JFrame fr,String org,String name) |
| 前置条件 | 用户选择收件信息输入 |
| 后置条件 | 显示收件信息输入界面 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| ReceiveOrderLSer.create(ReceiveVO receive,String org,String name) | 负责提供收件信息传递的逻辑接口 | |
| ReceiveOrderLSer.getExpress(String id) | 负责获取订单的逻辑接口 | |

RecipientPanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| RecipientPanel(JFrame fr,String name,String orgId,String userId,String org) | 语法 | public RecipientPanel(JFrame fr,String name,String orgId,String userId,String org) |
| 前置条件 | 用户选择生成接收单 |
| 后置条件 | 显示接收单生成界面 |
| RecipientPanel(JFrame fr,RecipientVO vo) | 语法 | public RecipientPanel(JFrame fr,RecipientVO vo) |
| 前置条件 | 用户选择查看接收单 |
| 后置条件 | 显示对应接收单 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| RecipientOrderLSer.Create(RecipientVO order) | 负责提供新建单据的逻辑接口 | |
| RecipientOrderLSer.getLocalHallAndAllCenter(String orgId) | 负责获取出发地的逻辑接口 | |
| RecipientOrderLSer.isLoadValid(String id) | 负责检查中转单是否合法的逻辑接口 | |
| RecipientOrderLSer.expressList(String id) | 负责获取中转单中订单列表的逻辑接口 | |
| RecipientOrderLSer.getNextId(String orgId) | 负责获取接收单编号的逻辑接口 | |
| RecipientOrderLSer.getComingLoadingOrder(String org) | 负责获取可接受的中转单的逻辑接口 | |

SendPanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SendPanel(JFrame fr,String name,String orgId,String userId,LinkedList<String> express) | 语法 | public SendPanel(JFrame fr,String name,String orgId,String userId,LinkedList<String> express) |
| 前置条件 | 用户选择生成派件单或派件单生成后还有快递未派件 |
| 后置条件 | 显示派件单生成界面 |
| SendPanel(JFrame fr,SendVO vo) | 语法 | public SendPanel(JFrame fr,SendVO vo) |
| 前置条件 | 用户选择查看派件单 |
| 后置条件 | 显示派件单 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| SendOrderLSer.create(SendVO order) | 负责提供新建单据的逻辑接口 | |
| SendOrderLSer.getCouriers(String id) | 负责获取快递员的逻辑接口 | |
| SendOrderLSer.getNextId(String orgId) | 负责获取派件单编号的逻辑接口 | |
| SendOrderLSer.getGoingExpress(String orgId) | 负责获取可派快递单编号的逻辑接口 | |

StockInPanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockInPanel(JFrame fr,String name,String orgId,String userId) | 语法 | public StockInPanel(JFrame fr,String name,String orgId,String userId) |
| 前置条件 | 用户选择生成入库单 |
| 后置条件 | 显示入库单生成界面 |
| StockInPanel(JFrame fr,StockInVO vo) | 语法 | public StockInPanel(JFrame fr,StockInVO vo) |
| 前置条件 | 用户选择查看入库单 |
| 后置条件 | 显示入库单 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| StockInOrderLSer.Create(StockInVO order) | 负责提供新建单据的逻辑接口 | |
| StockInOrderLSer. getShelves(String orgId) | 负责获取架的逻辑接口 | |
| StockInOrderLSer. getOrgs() | 负责获取目的地的逻辑接口 | |
| StockInOrderLSer. isExpressValid(String id) | 负责检查订单是否合法的逻辑接口 | |
| StockInOrderLSer. getNextId(String orgId) | 负责获取入库单编号的逻辑接口 | |
| StockInOrderLSer. getComingExpresses(String orgId) | 负责获取可入库的快递编号的逻辑接口 | |
| StockInOrderLSer.getRow(String shelf) | 负责获取架的可用排的逻辑接口 | |
| StockInOrderLSer. getPlace(String shelf,int row) | 负责获取排的可用位置的逻辑接口 | |

StockOutPanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockOutPanel(JFrame fr,String name,String orgId,String userId,String org) | 语法 | public StockOutPanel(JFrame fr,String name,String orgId,String userId,String org) |
| 前置条件 | 用户选择生成出库单 |
| 后置条件 | 显示出库单生成界面 |
| StockOutPanel(JFrame fr,StockOutVO vo) | 语法 | public StockOutPanel(JFrame fr,StockOutVO vo) |
| 前置条件 | 用户选择查看出库单 |
| 后置条件 | 显示出库单 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| StockOutOrderLSer.create(StockOutVO order) | 负责提供新建单据的逻辑接口 | |
| StockOutOrderLSer.getOrgs() | 负责提供获取目的地的逻辑接口 | |
| StockOutOrderLSer.isExpressValid(String id) | 负责检查订单是否合法的逻辑接口 | |
| StockOutOrderLSer.isTransitValid(String id) | 负责检查中转单是否合法的逻辑接口 | |
| StockOutOrderLSer.isConveyValid(String id) | 负责检查装车单是否合法的逻辑接口 | |
| StockOutOrderLSer.getNextId(String orgId) | 负责获取出库单编号的逻辑接口 | |
| StockOutOrderLSer.getGoingOrders(String org,String orgId) | 负责获取可出库的中转单和装车单编号的逻辑接口 | |
| StockOutOrderLSer.getGoingExpress(String orderId,String orgId) | 负责获取中转单和装车单中可出库的订单的编号的逻辑接口 | |

TransitPanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| TransitPanel(JFrame fr,String org,String name,String userId,String orgId) | 语法 | public TransitPanel(JFrame fr,String org,String name,String userId,String orgId) |
| 前置条件 | 用户选择生成中转单 |
| 后置条件 | 显示中转单生成界面 |
| TransitPanel(JFrame fr,TransitVO vo) | 语法 | public TransitPanel(JFrame fr,TransitVO vo) |
| 前置条件 | 用户选择查看中转单 |
| 后置条件 | 显示中转单 |
| TransitPanel.getPrice() | 语法 | public void getPrice() |
| 前置条件 | 用户修改中转单包含的订单 |
| 后置条件 | 显示运费 |
| 需要的接口（需接口） | | |
| 服务名 | 服务 | |
| TransitOrderLSer.create(TransitVO order) | 负责提供新建单据的逻辑接口 | |
| TransitOrderLSer.getCenters() | 负责提供获取目的地的逻辑接口 | |
| TransitOrderLSer.isExpressValid(List<String> expressList) | 负责检查订单是否合法的逻辑接口 | |
| TransitOrderLSer.getPrice(String org1,String org2,int tran,List<String> expressList) | 负责提供获取价格的逻辑接口 | |
| TransitOrderLSer.getNextId(String orgId) | 负责提供获取中转单编号的逻辑接口 | |
| TransitOrderLSer.getGoingExpresses(String orgId) | 负责提供获取可以中转的订单的编号的逻辑接口 | |

1. 界面层的动态模型

下图表明了系统中，当用户填写了接收单时，接受单新建逻辑处理的相关对象间的协作。其余单据动态模型类似如图。



1. 界面层的设计原理

见4.3.1.（5）

* + 1. transitInfoUI模块

1. 模块概述

transitInfoUI模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。transitInfoUI模块的职责及接口参见软件系统结构描述文档5.2.2表transitInfoUI模块的接口规范。

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LogicService. transitInfoLogic接口。StockCheckNowPanel，StockCheckPanel，StockWarningPanel，StockDividePanel用于显示对应的单据信息。FunctionFrame负责由功能主界面到具体界面的跳转。

TransitInfoUI模块的设计如图所示



TransitInfoUI模块各个类的职责如下表所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| ExpressInfoInqPanel | 显示订单信息查询界面 |

|  |  |
| --- | --- |
| TransitInfoInqPanel | 显示物流信息查询界面 |

1. 模块内部接口规范

TransitInfoUI的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| TransitInfoInqPanel(MainFrame fr) | 语法 | public TransitInfoInqPanel(MainFrame fr) |
| 前置条件 | 用户选择查看物流信息 |
| 后置条件 | 显示物流信息查询界面 |
| 需要的接口（需接口） | | |

|  |  |
| --- | --- |
| CourierLSer.enterBarcodeCounrier(String  Barcode) | 查询物流信息界面的逻辑接口 |

ExpressInfoInqPanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ExpressInfoInqPanel(JFrame fr) | 语法 | public ExpressInfoInqPanel(JFrame fr) |
| 前置条件 | 用户选择查看订单信息 |
| 后置条件 | 显示订单信息查询界面 |
| 需要的接口（需接口） | | |

|  |  |
| --- | --- |
| CustomerLogicSer.enterBarcodeCustomer(String  Barcode) | 查询物流轨迹和货运状态界面的逻辑接口 |

1. 界面层的动态模型

下图表明了系统中，当用户进行物流信息查询时界面间的跳转。



1. 界面层的设计原理

见4.3.1.（5）

* + 1. workOrgManUI模块

1. 模块描述

workOrgManUI模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

workOrgManUI模块的职责及接口参加软件系统体系结构描述文档5.2.2表workOrgManUI模块的接口规范

1. 整体结构

为了增加灵活性，我们在界面层和逻辑层之间添加logicService.WorkOrgManLogicService接口。VanListPanel DriverListPanel WorkListPanel WorkWageListPanel OrgListPanel反别作为车辆信息列表、司机信息列表、工作人员信息列表、人员薪水信息列表、机构信息列表的容器类，VanInfoPanel DriverInfoPanel WageDraftPanel用于车辆、司机、薪水详细信息查看及编辑，VanVO WorkVO DriverVO WageVO作为界面值对象。VanPO WorkPO DriverPO WagePO作为持久化对象保存。

WorkOrgManUI模块的设计如图所示



WorkOrgManUI各个类的职责如下表所示

WorkOrgManUI模块各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| VanInfoPanel | 负责显示车辆详细信息的面板类 |
| VanListPanel | 负责车辆信息的列表显示以及车辆信息管理操作的面板类 |
| DriverInfoPanel | 负责显示司机详细信息的面板类 |
| DriverListPanel | 负责司机信息的列表显示以及司机信息管理操作的面板类 |
| OrgListPanel | 负责机构信息的列表显示以及机构信息管理操作的面板类 |
| WageDraftPanel | 负责薪水制定操作的面板类 |
| WorkListPanel | 负责人员信息的列表显示以及人员信息管理操作的面板类 |
| WorkWageListPanel | 负责薪水信息的列表显示以及薪水信息管理操作的面板类 |

1. 模块内部类的接口规范

VanInfoPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| VanInfoPanel(InfoFrame fr,VanListPanel father,VanVO vanvo,String orgId) | 语法 | Public VanInfoPanel(InfoFrame fr,VanListPanel father,VanVO vanvo,String orgId) |
| 前置条件 | 用户打开车辆详细信息界面 |
| 后置条件 | 显示该车辆具体信息 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| VanListPanel.confirmed(VanVO van) | 负责提供增加或修改车辆的逻辑接口 |

VanListPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| VanListPanel(FunctionFrame fr,String orgId,String name) | 语法 | Public VanListPanel(FunctionFrame fr,String orgId,String name) |
| 前置条件 | 用户打开车辆列表界面 |
| 后置条件 | 显示车辆列表界面 |
| VanListPanel.confirmed(VanVO van) | 语法 | Result confirmed(VanVO van) |
| 前置条件 | 用户新增车辆或修改车辆信息 |
| 后置条件 | 新增或修改车辆 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| VanManLSer.addVan(VanVO van,String name) | 负责提供增加车辆的逻辑接口 |
| VanManLSer.deleteVan(String id,String name) | 负责提供删除车辆的逻辑接口 |
| VanManLSer.reviseVan(VanVO van,String name) | 负责提供修改车辆信息的逻辑接口 |
| VanManLSer.searchVan(String keyword,String hallId) | 负责提供车辆信息列表的逻辑接口 |
| VanManLSer.getNextId(String hallId) | 负责获取下一个车辆编号的逻辑接口 |

WageDraftPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| WageDraftPanel(JFrame fr,WorkWageListPanel pa) | 语法 | Public WageDraftPanel(JFrame fr,WorkWageListPanel pa) |
| 前置条件 | 用户打开薪水制定界面 |
| 后置条件 | 显示薪水制定界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| WorkWageListPanel.setWage(Wage kind,int money,int commission) | 负责设置人员薪水的接口 |

WorkWageListPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| WorkWageListPanel(FunctionFrame fr,String name) | 语法 | Public WorkWageListPanel(FunctionFrame fr,String name) |
| 前置条件 | 用户打开人员薪水列表界面 |
| 后置条件 | 显示人员薪水列表界面 |
| WorkWageListPanel.setWage(Wage kind,int money,int commission) | 语法 | Result setWage(Wage kind,int money,int commission) |
| 前置条件 | 用户制定人员薪水 |
| 后置条件 | 设置人员薪水 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| WorkOrgManSer. searchWorks(String keyword) | 负责获取人员薪水列表信息的逻辑接口 |
| WorkOrgManSer.setWage(List<WageVO> wageList,int[] index,String name) | 设置薪水的逻辑接口 |

WorkListPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| WorkListPanel(FunctionFrame fr,String name) | 语法 | Public WorkListPanel(FunctionFrame fr,String name) |
| 前置条件 | 用户打开人员列表界面 |
| 后置条件 | 显示人员列表界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| WorkManLSer.addWork（WorkVO work,String name） | 负责提供新建人员的逻辑接口 |
| WorkManLSer.deleteWork(int index,String name) | 负责提供删除人员的逻辑接口 |
| WorkManLSer.reviseWork(WorkVO work,int index,String name) | 负责提供修改人员的逻辑接口 |
| WorkManLSer. orgList() | 负责获取机构列表的逻辑接口 |
| WorkManLSer.searchWork(String keywords) | 负责提供搜索特定人员的逻辑接口 |

DriverInfoPanel的接口规范参见VanInfoPanel接口规范；

DriverListPanel的接口规范参见VanListPanel的接口规范；

OrgListPanel的接口规范参见WorkListPanel的接口规范。

1. 界面层的动态模型

下图表明了在快递物流系统中，当用户使用车辆信息管理工能时，车辆管理界面处理的相关对象之间的协作。



如下图所示的状态图描述了用户在使用车辆管理功能时系统的状态序列、引起转移的事件，以及因为状态转移而伴随的动作。用户进入由功能主界面进入VanListPanel;之后通过新增或者查看已有车辆信息进入VanInfoPanel。



1. 界面层的设计原理

见4.3.1.（5）

* + 1. cityUI模块

1. 模块描述

cityUI模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

cityUI模块的职责及接口参加软件系统体系结构描述文档5.2.2表cityUI模块的接口规范

1. 整体结构

在界面层和逻辑层之间添加logicService.cityLogic logicService.constantLogic接口以增加灵活性。CityListPanel用于显示城市信息列表以及管理城市信息，ConstantPanel用于常量制定，CityVO ConstantVO作为界面值对象被添加到设计模型中去。

CityUI模块的设计如下图所示



CityUI模块类的职责

|  |  |
| --- | --- |
| CityListPanel | 负责城市管理职责的界面 |
| ConstantPanel | 负责常量信息制定职责的界面 |

1. 模块内部的接口规范

CityPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| CityListPanel(JFrame fr,String name) | 语法 | Public CityListPanel(JFrame fr,String name) |
| 前置条件 | 用户打开城市管理界面 |
| 后置条件 | 显示城市管理界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| CityLSer. addCity(CityVO city,String name) | 添加城市的逻辑接口 |
| CityLSer. reviseCity(CityVO city,int index,String name) | 修改城市信息的逻辑接口 |
| CityLSer. deleteCity(String cityName,String name) | 删除城市的逻辑接口 |
| CityLSer. reviseCities(CitiesVO cities,String name) | 修改城市间距离和价格的逻辑接口 |
| CityLSer. citiesList() | 获取城市距离与价格列表的逻辑接口 |
| CityLSer.cityList() | 获取城市列表的逻辑接口 |

ConstantPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| ConstantPanel(JFrame fr,String name) | 语法 | Public ConstantPanel(JFrame fr,String name) |
| 前置条件 | 用户打开常量制定界面 |
| 后置条件 | 显示常量制定界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| ConstantLSer.setConstant(ConstantVO constant,String name) | 制定常量的逻辑接口 |
| ConstantLSer. getConstant() | 获取常量信息的逻辑接口 |

（4）（5）参见4.2.5 AccountUI模块

* + 1. accountUI模块

1. 模块描述

accountUI模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

accountUI模块的职责及接口参加软件系统体系结构描述文档5.2.2表accountUI模块的接口规范。

1. 整体结构

根据体系结构的设计，我们将系统分为界面层、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。比如展示层和逻辑层之间，我们添加LogicService.accountLogic接口。AccountManPanel用于封装账户管理的职责，CountPanel和CountInfoPanel分别用于新建账和查看期初信息。CountVO和AccountVO作为分别作为账户和账的界面对象被添加到设计模型中。

AccountUI模块的设计如图所示



AccountUI模块各个类的设计

AccountUI模块各个类的职责如下表所示

AccountUI各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| AccountManPanel | 负责账户管理的面板 |
| CountPanel | 负责新建账的面板 |
| CountInfoPanel | 负责期初信息查看的面板 |

1. 模块内部类的接口规范

AccountManPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| AccountManPanel(JFrame fr,String name) | 语法 | Public AccountManPanel(JFrame fr,String name) |
| 前置条件 | 用户打开账户管理界面 |
| 后置条件 | 显示账户管理界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| AccountLSer. addAccount(accountVO vo,String opName) | 根据vo新增账户对象 |
| AccountLcSer.deleteAccount(int index,String opName) | 根据index删除账户对象 |
| AccountLSer.reviseAccount(accountVO vo,int index,String opName) | 根据vo修改对应账户信息 |
| AccountLSer.searchAccount(String keywords) | 检索账户界面的逻辑接口 |

CountPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| CountPanel(FunctionFrame fr,String name) | 语法 | Public CountPanel(FunctionFrame fr,String name) |
| 前置条件 | 用户打开账管理界面 |
| 后置条件 | 显示账管理界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| CountLSer.newCount(String name) | 新增账对象 |
| CountLSer.deleteCcount(int index,String name) | 根据index删除账对象 |
| CountLSer.getCount() | 获得账信息 |

CountInfoPanel的模块接口

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| CountInfoPanel(CountVO count,JFrame fr) | 语法 | Public CountInfoPanel(CountVO count,JFrame fr) |
| 前置条件 | 用户打开期初详细信息界面 |
| 后置条件 | 显示期初详细信息界面 |

1. 界面层的动态模型

下图为使用账户管理功能时，账户管理界面处理相关对象之间的协作。



如下图所示的状态图描述了当用户使用账户管理功能时，系统的状态序列、引起转移的事件，以及因状态转移而伴随的动作呢。伴随着login方法被调用，界面跳转至FunctionFrame界面，用户选择账户管理功能后转至AccountManPanel。



AccountManPanel的状态图

1. 界面层的设计原理

见4.3.1.（5）

* + 1. stockUI模块

1. 模块概述

stockUI模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。stockUI模块的职责及接口参见软件系统结构描述文档5.2.2表stockUI模块的接口规范。

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LogicService. stockLogic接口。StockVO是作为库存记录的持久化对象，ShelfVO是作为货架记录的持久化对象被添加到设计模型中去的。StockCheckNowPanel，StockCheckPanel，StockWarningPanel，StockDividePanel用于显示对应的单据信息。FunctionFrame负责由功能主界面到具体界面的跳转。

stockUI模块的设计如图所示



stockUI模块各个类的职责如下表所示。

|  |  |
| --- | --- |
| 模块 | 职责 |

|  |  |
| --- | --- |
| StockCheckNowPanel | 库存查看界面，负责显示库存当前信息 |
| StockCheckPanel | 库存查看界面，负责显示库存信息 |
| StockWarningPanel | 库存警告界面，负责显示库存警告信息界面 |
| StockDividePanel | 库存分区界面，负责处理库存中架信息的管理操作 |

1. 模块内部接口规范

StockCheckPanel模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockCheckPanel(JFrame fr,String orgId) | 语法 | public StockCheckPanel(JFrame fr,String orgId) |
| 前置条件 | 用户选择库存查看 |
| 后置条件 | 显示时间段内库存信息 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| StockCheckLSer.orderStock(Calendar start,Calendar end,String id) | 负责提供查看出入库服务 |

StockCheckNowPanel模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockCheckNowPanel(JFrame fr,String orgId) | 语法 | public StockCheckNowPanel(JFrame fr,String orgId) |
| 前置条件 | 用户选择盘点 |
| 后置条件 | 显示库存盘点界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| StockCheckNowLSer.CheckNow(String id) | 负责提供库存盘点服务 |
| StockCheckNowLSer.outputExcel(String location,String name) | 负责提供导出库存情况为Excel的接口 |

StockWarningPanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockWarningPanel(JFrame fr,String orgId,String name) | 语法 | public StockWarningPanel(JFrame fr,String orgId,String name) |
| 前置条件 | 用户打开警戒线修改界面 |
| 后置条件 | 显示原始警戒线值 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| StockWarningLSer.setWarningsetWarning(ArrayList<Integer> warnings,String id,String name) | 负责设置警戒线 |
| StockWarningLSer.getWarning(String id) | 负责获得警戒线 |

StockDividePanel模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockDividePanel(JFrame fr,String orgId,String name) | 语法 | public StockDividePanel(JFrame fr,String orgId,String name) |
| 前置条件 | 用户输入架的编号，选择搜索 |
| 后置条件 | 显示相应架的分区信息 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |

|  |  |
| --- | --- |
| StockDivideLSer. searchShelf(String id,String keyword) | 负责提供查找架的接口 |
| StockDivideLSer.addShelf (ShelfVO shelf,String name,String orgId) | 负责提供添加架的接口 |
| StockDivideLSer.deleteShelf (String id,String name) | 负责提供删除架的接口 |
| StockDivideLSer.reviseShelf (ShelfVO shelf,String name,String orgId) | 负责提供修改架的接口 |
| StockDivideLSer.getNextId(String orgId) | 负责获取新增架子编号的接口 |

1. 界面层的动态模型

下图表明了系统中，当用户进行库存报警时界面间的跳转。



库存报警的状态图



库存查看的状态图



1. 界面层的设计原理

见4.3.1.（5）

* + 1. checkUI模块

1. 模块概述

checkUI模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

checkUI模块的职责及接口参加软件系统体系结构描述文档5.2.2表checkUI模块的接口规范

1. 整体结构

根据体系结构的设计，我们将系统分为界面层、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。比如界面层和逻辑层之间，我们添加LogicService.checkLogic接口。IncomeVO PaymentVO DiaryVO BusiCircumVO作为界面对象，被添加到设计模型中去，DiaryListPanel保有显示日志记录信息的职责，CostBenePanel用于显示成本收益，IncomeListPanel用于显示收款信息，BusiCircumPabel保有查看收款单和入款单信息的职责，FunctionFrame负责由功能主界面到具体界面的跳转。

CheckUI模块的设计如图所示



CheckUI模块各个类的职责如下表所示。

CheckUI模块各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| FunctionFrame | 负责由用户功能主界面到具体功能界面的跳转 |
| CollectFormPanel | 负责显示成本收益表的面板 |
| DiaryListPanel | 负责显示日志记录的面板 |
| IncomeListPanel | 负责显示收款记录的面板 |
| StatementSheetPanel | 负责显示收款单和付款单信息列表的面板 |

1. 模块内部类的接口规范

CostBenePanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| CostBenePanel(JFrame fr) | 语法 | public CostBenePanel(JFrame fr) |
| 前置条件 | 打开成本收益界面 |
| 后置条件 | 显示成本收益界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| CostBeneLSer. getCostBene() | 获取成本收益 |

DiaryListPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| DiaryListPanel(JFrame fr) | 语法 | Public DiaryListPanel(JFrame fr) |
| 前置条件 | 用户打开查看日志记录界面 |
| 后置条件 | 显示日志记录 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| DiaryLSer.seeDiary(Calendar start,Calendar end) | 根据开始和结束日期获得相应的日志记录 |

IncomeListPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| IncomeListPanel(JFrame fr) | 语法 | Public IncomeListPanel(JFrame fr) |
| 前置条件 | 用户打开收款记录界面 |
| 后置条件 | 显示收款记录界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| IncomeListLSer.getHall() | 获得营业厅列表 |
| IncomeListLSer. total(ArrayList<IncomeVO> income) | 合计收款 |
| IncomeListLSer. seeIncomeList(Calendar date,String id) | 根据日期或营业厅获得所有的收款信息组成的列表 |

BusiCircumPanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| BusiCircumPanel(JFrame fr) | 语法 | Public BusiCircumPanel(JFrame fr) |
| 前置条件 | 用户打开经营情况表界面 |
| 后置条件 | 显示经营情况表界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| BusiCircumLSer. getBusiCircum(Calendar start,Calendar end) | 根据开始和结束日期获得相应的收款单和付款单信息列表。 |
| BusiCircumLSer.outputExcel(String name,String location) | 根据地址和名称导出表格 |

1. 界面层的动态模型

下图为快递物流系统中，当用户选择查看经营情况表功能时，界面层相关对象之间的协作。



下图所示的状态图描述了用户在使用查看经营情况表功能时系统界面的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着login方法被调用界面跳转至FunctionFrame,选择查看经营情况表转至StatementSheetPane，由经营情况表界面可选择跳转至IncomePanel或者PaymentPanel。



1. 界面层的设计原理

见4.3.1.（5）

* + 1. orderApproveUI模块

1. 模块描述

OrderApproveUI模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

OrderApproveUI模块的职责及接口参加软件系统体系结构描述文档5.2.2表OrderApproveUII模块的接口规范

1. 整体结构

在界面层和逻辑层之间添加OrderApproveLSer OrderResubmitLSer接口来增加灵活性。OrderApprovePanel保有查看单据列表以及审批单据的职责,ApproveTable作为OrderApprovePanel的辅助类，OrderRevisePanel用来显示审批未通过的单据，OrderVO以及各种单据的VO作为界面数据对象被添加到设计模型中。

OrderApproveUI模块的设计如下图所示



OrderApproveUI模块的类的职责如下表所示。

OrderApproveUI模块各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| FunctionFrame | 负责由功能主界面到具体功能界面的跳转 |
| OrderApprovePanel | 负责单据审批的面板 |
| ApproveTable | 单据审批的辅助类 |
| OrderRevisePanel | 负责显示未通过审批的单据的面板 |

1. 模块内部类的接口规范

OrderApprovePanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| OrderApprovePanel (FunctionFrame fr,String name) | 语法 | Public OrderApprovePanel(FunctionFrame fr,String name) |
| 前置条件 | 用户打开单据审批界面 |
| 后置条件 | 显示单据审批界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| orderApproveLSer.askExamine() | 提供待审批的单据信息列表 |
| orderApproveLSer.examine(boolean approve,ArrayList<Integer> indexs) | 审批单据并更新单据审批状态 |
| orderApproveLSer.chooseOrder(String id,Order kind) | 根据kinde和id查找对应的单据并提供单据的详细信息 |

ApproveTable的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| ApproveTable() | 语法 | public ApproveTable(DefaultTableModel tableModel) |
| 前置条件 | 用户打开单据审批界面 |
| 后置条件 | 显示待审批单据的表格 |

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| ApproveTable. getTableHeader() | 语法 | public JTableHeader getTableHeader() |
| 前置条件 | 无 |
| 后置条件 | 获取表头 |

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| ApproveTable. isCellEditable() | 语法 | public boolean isCellEditable(int row, int column) |
| 前置条件 | 无 |
| 后置条件 | 返回是否可编辑 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| orderApproveLSer.askExamine() | 提供待审批的单据信息列表 |
| orderApproveLSer. examine (boolean approve,ArrayList<Integer> indexs) | 审批单据并更新单据审批状态 |
| orderApproveLSer.chooseOrder(String id,Order kind) | 根据kinde和id查找对应的单据并提供单据的详细信息 |

OrderRevisePanel的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| OrderRevisePanel | 语法 | Public OrderRevisePanel(JFrame fr) |
| 前置条件 | 用户打开单据重提交界面 |
| 后置条件 | 显示未通过审批待修改的单据 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| orderResubmitLSer. getUnpassed(String id) | 获取未通过审批的单据数量 |
| orderResubmitLSer. getOrdersDisplay() | 获得未通过审批的单据列表 |
| orderResubmitLSer.getOrder(int index) | 显示某单据详细信息 |

1. 界面层的动态模型

下图表明了在快递物流系统中，当用户使用单据审批功能时，单据审批界面处理的相关对象之间的协作。



下图所示的状态图描述了用户在使用单据审批功能时系统的状态序列、引起转移的事件，以及因转移而伴随的动作。随着用户登陆并选择单据审批功能，系统跳转至OrderApprovePanel, 之后可以在本界面进行全选和审批单据操作，也可以通过查看详细操作跳转至指定的单据信息界面，即\*orderPanel。



1. 界面层的设计原理

见4.3.1.（5）

* + 1. UserPanel模块

1. 模块概述

UserPanel模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

UserPanel模块的职责及接口参见软件系统体系结构描述文档5.2.2表*User*Panel*模块的接口规范*

1. 整体结构

在界面层和逻辑层之间，我们添加logicService.UserService接口以增强灵活性，各类用户的UserPanel保有让用户选择功能并跳转至相应界面的职责。

UserPanel模块的设计如图所示



UserPanel模块各个类的职责如下表所示

UserPanel模块各个类的职责

|  |  |
| --- | --- |
| AdministerPanel | 负责由管理员功能界面向具体功能界面的跳转 |
| BussinessHallPanel | 负责由营业厅业务员功能主界面向具体功能界面的跳转 |
| CourierPanel | 负责由快递员功能主界面向具体功能界面的跳转 |
| FinancePanel | 负责由普通财务人员功能主界面向具体功能界面的跳转 |
| FinanceSuperPanel | 负责由高级财务人员功能主界面向具体功能界面的跳转 |
| ManagerPanel | 负责由总经理功能主界面向具体功能界面的跳转 |
| StockManagePanel | 负责由仓库管理员功能主界面向具体功能界面的跳转 |
| TransitCenterPanel | 负责由中转中心业务员功能主界面向具体功能界面的跳转 |

1. 模块内部类的接口规范

BusinessHallPanel模块的接口规范

|  |
| --- |
| 提供的服务（供接口） |

|  |  |  |
| --- | --- | --- |
| BusinessHallPanel | 语法 | public BusinessHallPanel(UserVO user,JFrame fr) |
| 前置条件 | 营业厅业务员登陆 |
| 后置条件 | 显示营业厅业务员功能界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |

|  |  |
| --- | --- |
| LoginLSer.login(String id,String password) | 负责提供登录的逻辑接口 |

其他用户类的接口规范参照AdministerPanel的接口规范，依据用户功能的不同（参照界面图）变动方法名。

1. 界面的动态模型

下图为营业厅业务员进入系统后，界面层相关类之间的协作。



如下图所示的状态图描述了营业厅业务员由登陆至进入具体功能界面的状态序列，引起转移的事件以及由转移伴随的动作。



1. 界面层的设计原理

利用分散式分格，每个界面自主实现各自的界面职责。

1.1.10 Utility模块

1. 模块概述

Utility模块承担的需求为部分非法性检测，操作成功提示与出错报警

1. 整体结构
2. 模块内部类的接口规范