­

南京大学软件学院

2015秋 软件工程 快递系统项目实践

**体系结构设计文档**

程翔 吴嘉荣

王嘉琛 孟鑫

目录 Contents

1. 引言

1.1 编制目的

2. 产品描述

3. 逻辑视角

4. 组合视角

4.1 开发包图

4.2 运行时进程

4.3 物理部署

5. 接口视角

5.1 模块的职责

5.2 用户界面层的分解

5.2.1 用户界面层模块的职责

5.2.2 用户界面模块的接口规范

5.2.3 用户界面模块设计原理

5.3 业务逻辑层的分解

5.3.1 业务逻辑层模块的职责

5.3.2 业务逻辑模块的接口模范

目录 Contents

5.4 数据层的分解

5.4.1 数据层模块的职责

5.4.2 数据层模块的接口规范

6. 信息视角

6.1 数据持久化对象

6.2 文件格式

7. 界面视角

更新历史

|  |  |  |
| --- | --- | --- |
| 姓名 | 时间 | 更新细节 |
| 程翔 | 2015.10.21 | 创建文档 |
| 孟鑫 | 2015.10.26 | 添加内容 |
| 孟鑫 | 2015.11.6 | 添加人机交互设计部分 |

# 引言

## 1.1编制目的

本报告详细完成对快递物流系统的概要设计，达到指导详细设计和开发的目的，同时实现测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户编写，是了解系统的导航。

## 1.2参考文献

1、IEEE标准

2、《2333小组用例文档》、《2333小组需求规格说明文档》

3、《软件体系结构设计目标（IEEE标准）》

4、《项目实践V4》 刘钦

# 产品描述

快递物流系统是为本地快递物流企业开发的业务系统，开发目标是帮助该企业处理日常的重点业务，包括收发快递、包裹转运、库存管理和财务统计等。

通过该快递物流系统的使用，期望帮助该快递物流企业保持合理库存，提高用户服务体验，增加业务额，提高财务人员工作效率，并且为经理的决策做支持。

# 3.逻辑视角

快递物流系统中，选择了分层体系结构的风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好的示意整个高层抽象。展示层包括GUI页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案下面两图所示。

图 1参照体系结构风格的包图表达逻辑视角

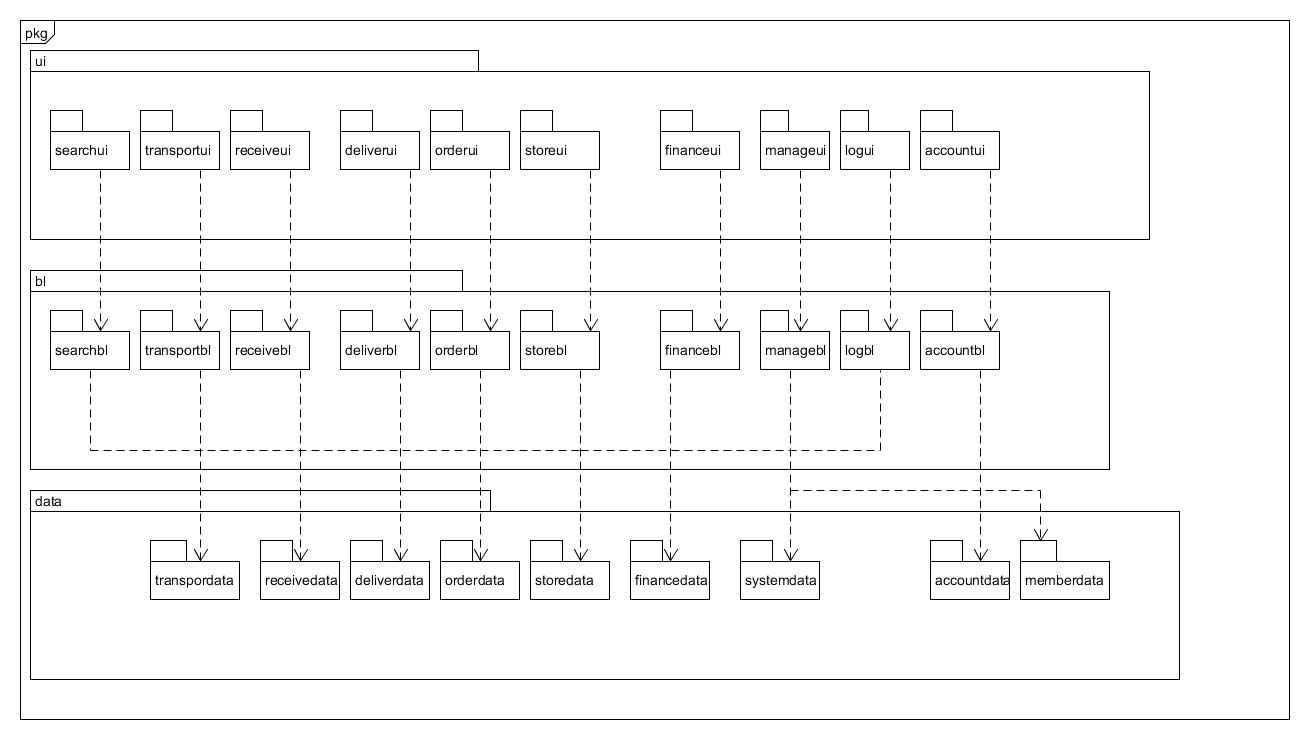


图 2 软件体系结构逻辑设计方案

# 4.组合视角

## 4.1开发包图

快递物流系统的最终开发包设计如表1所示

**表1 快递物流系统的最终开发包设计**

|  |  |
| --- | --- |
| 开发（物理）包 | 依赖的其他开发包 |
| mainui | searchui, transportui, receiveui, deliverui, orderui, storeui, financeui, manangeui, configurationui, accountui, logui, examineui,vo |
| searchui | searchblService,界面类库包,vo |
| searchblService | / |
| searchbl | searchblService ,transpotDataService, receiveDataService, deliverDataService, orderDataService, storeDataService,po |
| transportui | transportblService,界面类库包,vo |
| transportblService | / |
| transportbl | transportblService , transportDataService ,formatCheckService, examinebl,po |
| transportDataService | JAVA RMI,po |
| transportdata | transportDataService, databaseutility,po |
| receiveui | receiveblService,界面类库包,vo |
| receiveblService | / |
| receivebl | receiveblService , receiveDataService ,formatCheckService, examinebl,po |
| receiveDataService | JAVA RMI,po |
| receivedata | receiveDataService, databaseutility,po |
| deliverui | deliverblService,界面类库包,vo |
| deliverblService | / |
| deliverbl | deliverblService , deliverDataService ,formatCheckService, examinebl,po |
| deliverDataService | JAVA RMI,po |
| deliverdata | deliverDataService, databaseutility,po |
| orderui | orderblService,界面类库包,vo |
| orderblService | / |
| orderbl | orderblService , orderDataService ,formatCheckService, examinebl,po |
| orderDataService | JAVA RMI,po |
| orderdata | orderDataService, databaseutility,po |
| storeui | storeblService,界面类库包,vo |
| storeblService | / |
| storebl | storeblService, storeDataService, formatCheckService, examinebl,po |
| storeDataService | JAVA RMI,po |
| storedata | storeDataService, databaseutility,po |
| financeui | financeblService,界面类库包,vo |
| financeblService | / |
| financebl | financeblService, financeDataService, formatCheckService, examinebl,po |
| financeDataService | JAVA RMI,po |
| financedata | financeDataService, databaseutility,po |
| manangeui | manangeblService,界面类库包,vo |
| manangeblService | / |
| manangebl | manangeblService, companyDataService, memberDataService, po |
| companyDataService | JAVA RMI,po |
| companydata | companyDataService, databaseutility,po |
| memberDataService | JAVA RMI,po |
| memberdata | memberDataService, databaseutility,po |
| configurationui | configurationblService, 界面类库包,vo |
| configurationblService | / |
| configurationbl | configurationblService, configurationDataService,po |
| configurationDataService | JAVA RMI,po |
| configurationdata | configurationDataService, databaseutility,po |
| accountui | accountblService, 界面类库包,vo |
| accountblService | / |
| accountbl | accountblService, accountDataService, charRemindService,po |
| accountDataService | JAVA RMI,po |
| accountdata | accountDataService, databaseutility,po |
| logui | logService, 界面类库包,vo |
| logService | / |
| logbl | logService, transpotDataService, receiveDataService, deliverDataService, orderDataService, storeDataService, financeDataService ,po |
| examineui | examineblService, 界面类库包,vo |
| examineblService | JAVA RMI,po |
| examinebl | JAVA RMI, examineblImpl ,po |
| examineblImpl | transportDataService, receiveDataService, deliverDataService, orderDataService, storeDataService, financeDataService,po |
| formatCheckService | formatCheckImpl |
| formatCheckImpl | transpotDataService, receiveDataService, deliverDataService, orderDataService, storeDataService, financeDataService ,po |
| chatRemindService | JAVA RMI,po |
| chatRemindImpl |  |
| vo |  |
| po |  |
| utilitybl |  |
| 界面类库包 |  |
| JAV RMI |  |
| databaseutility |  |



图 3 客户端开发包图



图 4 服务器端开发包图

## 4.2运行时进程

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

## 4.3物理部署

快递物流系统中客户端构件是放在客户端机上，服务器端构件是放在服务器端机器上。在客户端节点上，还要部署RMIStub构件。由于Java RMI 构件属于JDK 1.8的一部分。所以，在系统JDK环境已经设置好的情况下，不需要独立部署。部署图如下图所示。

图 5 进程图



图 6 部署图

# 5.接口视角

## 5.1模块的职责

客户端模块和服务器模块视图分别如下两图所示。客户端各层和服务器端各层的职责分别如表2和表3所示。

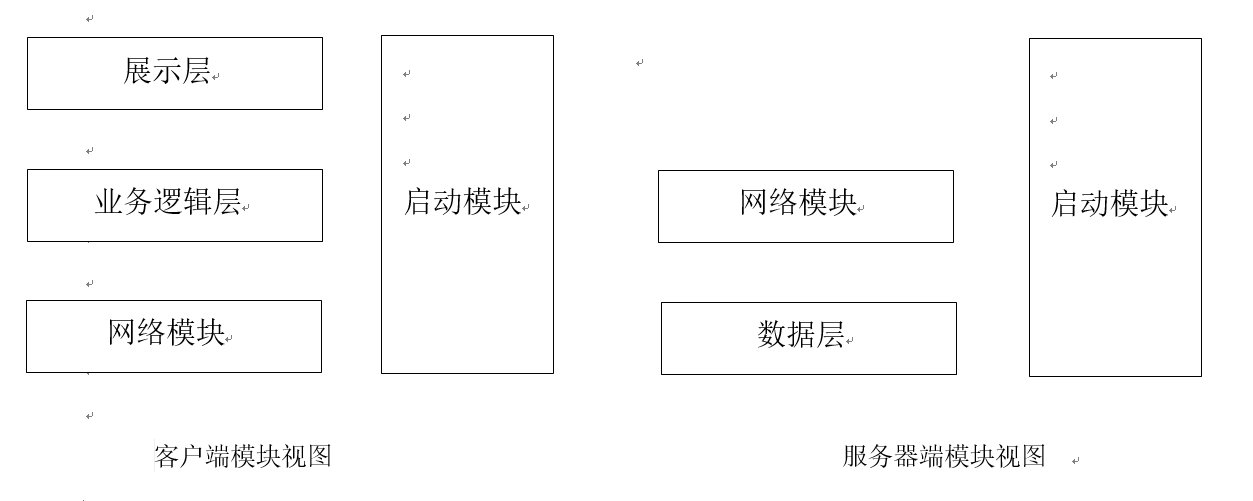


图7 客户端模块视图 图8 服务器端模块视图

**表2 客户端各层的职责**

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责初始化网络通信机制，启动用户界面 |
| 用户界面层 | 基于窗口的进销存管理系统客户端用户界面 |
| 业务逻辑层 | 对于用户界面的输入进行响应并进行业务处理逻辑 |
| 客户端网络模块 | 利用java RMI机制查找RMI服务 |

**表3 服务器端各层的职责**

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

每一层只是使用下方直接接触的层。层与层之间仅仅是通过接口的调用来完成的。层之间调用的接口如表4所示。

**表4 层之间调用的接口**

|  |  |  |
| --- | --- | --- |
| 接口 | 服务调用方 | 服务提供方 |
| commodityblservice  inventoryblservice  clientblservice  purchaseblservice  saleblservice  recordblservice  accountblservice  fundblservice  accountaintblservice  approvalblservice  promotionblservice  userblservice | 客户端展示层 | 客户端业务逻辑层 |
| commoditydataservice  inventorydataservice  clientdataservice  purchasedataservice  saledataservice  recorddataservice  accountdataservice  funddataservice  accountaintdataservice  approvaldataservice  promotiondataservice  userdataservice | 客户端业务逻辑层 | 服务器端数据层 |

借用销售用例来说明层之间的调用，如图9所示。每一层之间都是由上一层依赖了一个接口(需接口)，而下层实现这个接口(供接口)。orderblservice提供了order界面所需要的所有业务逻辑功能。orderdataservice提供了对数据库的增、删、改、查等操作。这样的实现就大大降低了层与层之间的耦合。

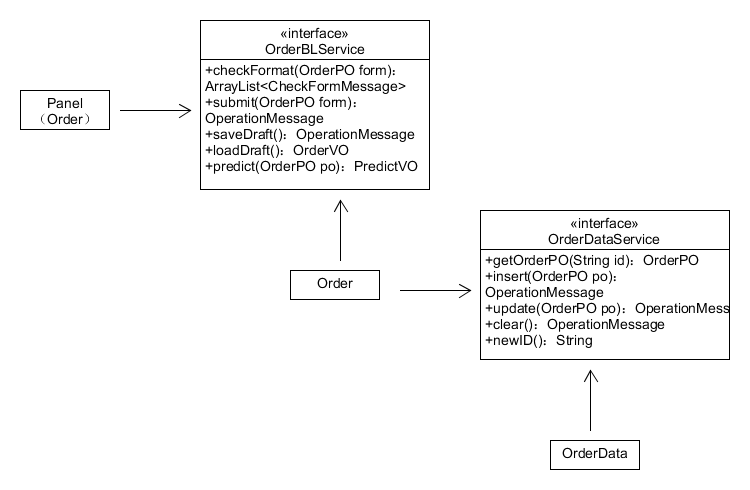


图9 输入新订单用例层之间调用的接口

## 5.2用户界面层的分解

系统存在8类44个用户界面，他们分为：查询物流信息界面；

快递员界面（快递员主界面，查询物流信息界面，输入新订单界面，查看派送单界面，输入收件信息界面）；

营业厅业务员界面（营业厅业务员主界面，填写到达单界面，填写装车单界面，填写派件单界面，填写收款单界面，管理车辆信息界面，管理司机信息界面）；

财务会计界面（财务会计主界面，查看收款单界面，新建付款单界面，期初建账界面，查询期初信息界面，查看财务报表界面，银行账户管理界面）；

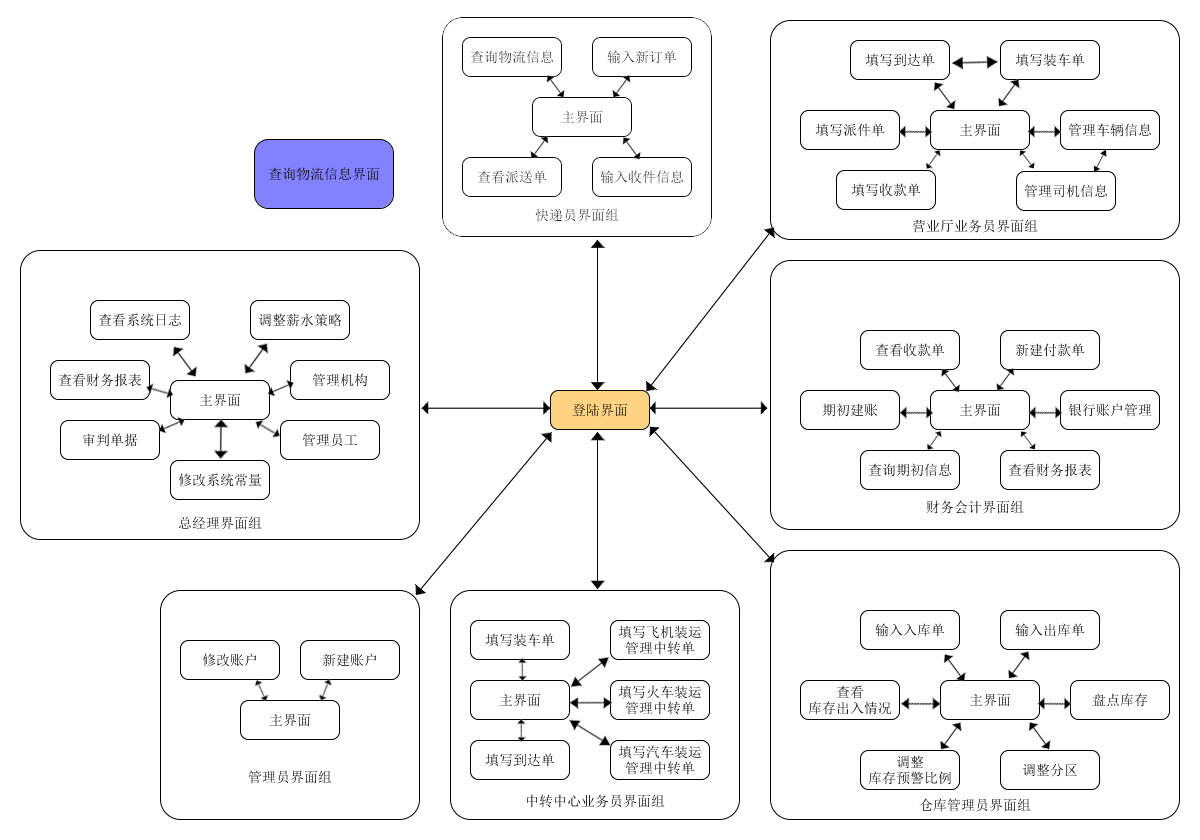
仓库管理员界面（仓库管理员主界面，输入入库单界面，输入出库单界面，盘点库存界面，调整分区界面，调整库存预警比例界面，查看库存出入情况界面）；

中转中心业务员界面（中转中心业务员主界面，填写装车单界面，填写到达单界面，填写飞机装运管理中转单界面，填写火车装运管理中转单界面，填写汽车装运管理中转单界面）；

管理员界面（管理员主界面，修改账户界面（包括查看和删除功能），新建账户界面）；

总经理界面（总经理主界面，查看系统日志界面，调整薪水策略界面，管理机构界面，管理员工界面，修改系统常量界面，审判单据界面，查看财务报表界面）

\*注：其中每个用户的主界面几乎相同，在此只给出快递员的主界面作为样例，总经理和财务人员的“查看财务报表界面”和“查看系统日志界面”几乎相同，只在财务人员处给出，不再赘述



其中所有用户界面通过同一个类处理，即主用户界面类（MainFrame）



### 5.2.1.用户界面层模块的职责

MainFrame主要用于负责界面的显示和界面之间的跳转工作。

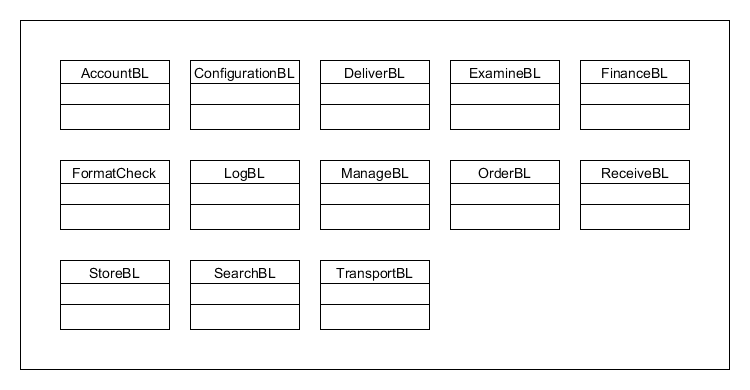
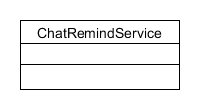
### 5.2.2.用户界面模块的接口规范

|  |  |  |
| --- | --- | --- |
| MaiFrame | 语法 | init(args:String[]) |
| 前置条件 | 用户触发相应需要显示界面的条件 |
| 后置条件 | 显示Frame并且加载Panel |

### 5.2.3.用户界面模块设计原理

用户界面你用java的Swing和AWT库来实现。

## 5.3.业务逻辑层的分解



### 5.3.1业务逻辑层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| accountbl | 负责实现账户管理和登陆所需要的服务 |
| configurationbl | 负责实现查询、修改系统常量所需要的服务 |
| deliverbl | 负责实现输入派件单所需要的服务 |
| examinebl | 负责实现审批单据所需要的服务 |
| financebl | 负责实现财务管理所需要的服务 |
| formatcheckbl | 负责实现检查表单格式所需要的服务 |
| logbl | 负责实现查询系统日志所需要的服务 |
| managebl | 负责实现总经理职责（除审批单据）所需要的服务 |
| orderbl | 负责实现输入新订单所需要的服务 |
| receivebl | 负责实现输入到达单所需要的服务 |
| storebl | 负责实现库存管理所需要的服务 |
| searchbl | 负责实现查询物流信息所需要的服务 |
| transportbl | 负责实现输入装车单/中转单所需要的服务 |
| initialbl | 负责实现期初建账所需要的服务 |
| chatremindservice | 负责实现消息提醒所需要的服务 |

**accountbl包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Account.getAccountVOs | 语法 | public ArrayList<AccountVO> getAccountVOs() |
| 前置条件 | 管理员登陆成功 |
| 后置条件 | 显示所有账户信息 |
| AccountBL.getAccountVO | 语法 | public getAccountVO(String accountID) |
| 前置条件 | 管理员输入用户名并且已经通过检查 |
| 后置条件 | 显示该用户名对应的用户信息 |
| Account.addAccount | 语法 | public OperationMessage addAccount (AccountPO po) |
| 前置条件 | 管理员确认删除该用户信息 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| AccountBL.deleteAccount | 语法 | public OperationMessage deleteAccount (AccountPO po) |
| 前置条件 | 管理员确认删除该用户信息 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| Account. modifyAccount | 语法 | public OperationMessage modifyAccount (AccountPO po) |
| 前置条件 | 管理员确认完成修改该用户信息 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| Account.checkAccountat | 语法 | public OperationMessage  checkAccountat (String id , String password) |
| 前置条件 | 用户输入账号和密码并确认 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| AccountDataService.find(String name) | 查找name对应的AccountPO 并返回 | |
| AccountDataService.  insert(AccountPO po) | 插入单一持久化对象 | |
| AccountDataService.  delete(AccountPO po) | 删除单一持久化对象 | |
| AccountDataService.  update(AccountPO po) | 更新单一持久化对象 | |
| AccountDataService.checkID | 检查ID是否存在 | |
| AccountDataService.checkAccount(String ID , String password) | 检查ID对应的密码是否是password | |
| VOService.getUserVO(String ID) | 返回ID对应的UserVO | |

**configurationbl包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| configurationbl.getCityDistance | 语法 | public ArrayList<CityDistanceVO> getCityDistance() |
| 前置条件 | 无 |
| 后置条件 | 返回城市间距离列表 |
| configurationbl.modifyCityDistance | 语法 | public OperationMessage modifyCityDistance(CityDistanceVO after) |
| 前置条件 | after为新修改的城市距离数据；已经获取所有城市距离的列表 |
| 后置条件 | 若修改成功则返回成功；否则返回失败原因，系统提示修改失败原因 |
| configurationbl.getSalaryStrategy | 语法 | public ArrayList<SalaryStrategyVO> getSalaryStrategy() |
| 前置条件 | 无 |
| 后置条件 | 获取员工薪水策略列表 |
| configurationbl.modifySalaryStrategy | 语法 | public OperationMessage modifySalaryStrategy(ArrayList<SalaryStrategyVO> after) |
| 前置条件 | after为所有修改后的数据列表 |
| 后置条件 | 若修改成功返回成功；否则返回失败原因 |
| configurationbl.getPack | 语法 | public PackVO getPack() |
| 前置条件 | 无 |
| 后置条件 | 返回包装费用的信息 |
| configurationbl.modifyPack | 语法 | public OperationMessage modifyPack(PackVO after) |
| 前置条件 | after为修改后的包装价格；已经获取所有的包装费用信息 |
| 后置条件 | 若修改成功，返回成功；否则，返回失败原因 |
| configurationbl.getPrice | 语法 | public PriceVO getPrice() |
| 前置条件 | 无 |
| 后置条件 | 返回当前普通快递的价格 |
| configurationbl.modifyPrice | 语法 | public OperationMessage modifyPrice(PriceVO after) |
| 前置条件 | after为修改后的运费价格；已经获取运费价格信息 |
| 后置条件 | 若修改成功，返回成功；否则，返回失败原因 |
| configurationbl.getProportion | 语法 | public ProportionPO getProportion() |
| 前置条件 | 无 |
| 后置条件 | 获取不同快递收费比例 |
| configurationbl.modifyProportion | 语法 | public OperationMessage modifyProportion(ProportionVO after) |
| 前置条件 | after为修改后的比例信息；已经获取价格比例 |
| 后置条件 | 若修改成功，返回成功；否则返回失败原因 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| configurationDataService.  getCityDistance() | 获取城市距离信息 | |
| configurationDataService.  modifyCityDistance(CityDistancePO after) | 修改城市距离信息 | |
| configurationDataService.  getSalaryStrategy() | 获取员工薪水策略 | |
| configurationDataService.  modifySalaryStrategy(SalaryPOafter) | 修改员工薪水策略 | |
| configurationDataService.  getPack() | 获取包装费用 | |
| configurationDataService.  modifyPack(PackPO after) | 修改包装费用 | |
| configurationDataService.  getPrice() | 获取快递收费价格 | |
| configurationDataService.  modifyPrice(PricePO after) | 修改快递收费价格，并重新计算其它价格 | |
| configurationDataService.  getProportion() | 获取不同快递收费比例 | |
| configurationDataService.  modifyProportion(ProportionPO after) | 修改不同快递收费比例，并重新计算其它价格 | |

**deliverbl包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| DeliverBL.checkFormat | 语法 | public ArrayList< CheckFormMessage > checkFormat(DeliverVO form,boolean isFinal) |
| 前置条件 | 输入一个派件单VO对象，isFinal是否为最终提交 |
| 后置条件 | 检查输入信息是否符合系统要求，若符合，返回的list中只有一个显示正确的CheckFormMessage, 否则返回能指示出错误的CheckFormMessage的列表 |
| DeliverBL.submit | 语法 | public OperationMessage submit(DeliverVO form) |
| 前置条件 | 参数form作为被提交的到达单VO对象，且已经通过检查 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| 前置条件 | 添加相应的数据到派件单VO中 |
| 后置条件 | 返回派件单VO |
| DeliverBL. saveDraft | 语法 | public OperationMessage save Draft (DeliverVO form) |
| 前置条件 | 派件单没有被提交，选择保存，form 表单信息 |
| 后置条件 | 返回保存是否成功的对象 |
| DeliverBL.loadDraft | 语法 | public DeliverVO loadDraft () |
| 前置条件 | 系统中存在DeliverVO草稿 |
| 后置条件 | 返回相应的DeliverVO对象 |
| DeliverBL.getOrderVO | 语法 | public OrderVO getOrderVO(String orderID) |
| 前置条件 | 传入订单号orderID |
| 后置条件 | 返回该ID对应的VO对象 |
| 需要的服务（需接口） | | |
| FormatCheckService.checkOrderID(String number) | 检查订单号码是否存在 | |
| FormatCheckService.checkPreDate(String date) | 检查预计到达日期填写是否有误 | |
| FormatCheckService.checkName(String name) | 检查派件员信息填写是否有误 | |
| DeliverDataService.newID | 获得一个新的到达单ID | |
| ExamineService.  submit(FormPO po) | 提交表格审批 | |
| DeliverDataService.insert | 插入单据 | |
| DeliverDataService.delete | 删除单据 | |
| DeliverDataService.update | 更新单据 | |
| OrderDataService.getOrderPO | 获得订单PO | |

**financebl包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| RevenueBLService.loadOrder | 语法 | public OperationMessage loadOrder(String orderNumber) |
| 前置条件 | 无 |
| 后置条件 | 若存在对应订单号的OrderPO，载入到此对象中，返回正常信息；否则返回无此订单的提示 |
| RevenueBLService.  getNewRevenueID | 语法 | public String getNewRevenueID  (String date) |
| 前置条件 | 无 |
| 后置条件 | 返回未使用过的收款单的编号 |
| RevenueBLService.  getOrderVO | 语法 | public OrderVO getOrderVO() |
| 前置条件 | 已经调用过loadOrder且成功 |
| 后置条件 | 将加载的OrderPO包装成OrderVO返回 |
| RevenueBLService. checkFormat | 语法 | public List<CheckFormatMessage> checkFormat(RevenueVO rvo，boolean isFinal) |
| 前置条件 | 无 |
| 后置条件 | 检查收款单各项，若有错或不完整，将提示作为CheckFormatMessage作为列表返回；若无误，列表中只包含成功信息 |
| RevenueBLService. submit | 语法 | public OperationMessage submit(RevenueVO rvo) |
| 前置条件 | RevenueVO已经过检查 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| RevenueBLService. loadDraft | 语法 | public RevenueVO loadDraft() |
| 前置条件 | 无 |
| 后置条件 | 若对应序列化文件存在，解序列化成对应的RevenueVO对象返回；否则返回null |
| RevenueBLService. saveDraft | 语法 | public OperationMessage saveDraft(RevenueVO draft) |
| 前置条件 | 输入的RevenueVO不为空 |
| 后置条件 | 将RevenueVO序列化成文件存储，成功后返回成功信息；若之前有同名文件，返回已有草稿这个信息 |
| RevenueBLService. getRevenueVO | 语法 | public RevenueVO getRevenueVO(String date, String hallID) |
| 前置条件 | 两个参数已经经过检查 |
| 后置条件 | 返回对应的收款单PO包装过的VO对象 |
| RevenueBLService. getRevenueVO | 语法 | public RevenueVO getRevenueVO(String revenueID) |
| 前置条件 | 无 |
| 后置条件 | 返回对应的收款单PO包装过的VO对象 |
| RevenueBLService. getRevenueVOs | 语法 | public List<RevenueVO> getRevenueVOs(String startDate, String endDate) |
| 前置条件 | 两个参数已经经过检查 |
| 后置条件 | 返回对应天数的所有的收款单 |
| PaymentBLService. checkFormat | 语法 | public List<CheckFormatMessage> checkFormat(PaymentVO pvo，boolean isFinal) |
| 前置条件 | 输入的PaymentVO不为空 |
| 后置条件 | 检查输入信息是否符合系统要求，若符合，返回的list中只有一个显示正确的CheckFormatMessage, 否则返回能指示出错误的CheckFormatMessage的列表 |
| PaymentBLService. getNewPaymentID | 语法 | public String getNewPaymentID  (String date) |
| 前置条件 | 无 |
| 后置条件 | 返回未使用过的付款单的编号 |
| PaymentBLService. submit | 语法 | public OperationMessage submit(PaymentVO rvo) |
| 前置条件 | PaymentVO已经通过检查 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| PaymentBLService. loadDraft | 语法 | public PaymentVO loadDraft() |
| 前置条件 | 无 |
| 后置条件 | 若对应序列化文件存在，解序列化成对应的PaymentVO对象返回；否则返回null |
| PaymentBLService. saveDraft | 语法 | public OperationMessage saveDraft(PaymentVO draft) |
| 前置条件 | 输入的PaymentVO不为空 |
| 后置条件 | 将PaymentVO序列化成文件存储，成功后返回成功信息；若之前有同名文件，返回已有草稿这个信息 |
| PaymentBLService. getPaymentVO | 语法 | public PaymentVO getPaymentVO(String paymentID) |
| 前置条件 | 无 |
| 后置条件 | 返回对应的付款单PO包装过的VO对象 |
| PaymentBLService. getPaymentVOs | 语法 | public List<PaymentVO> getPaymentVOs(String startDate, String endDate) |
| 前置条件 | 两个参数已经经过检查 |
| 后置条件 | 返回对应天数的所有的付款单 |
| FinanceChartBLService. getHistogram | 语法 | public BaseChartVO getHistogram() |
| 前置条件 | 无 |
| 后置条件 | 将本地的收款单和付款单的数据汇总，生成直方图的VO对象返回 |
| FinanceChartBService. getPieChart | 语法 | public PieChartVO getPieChart() |
| 前置条件 | 无 |
| 后置条件 | 将本地的收款单和付款单的数据汇总，生成饼状图的VO对象返回 |
| FinanceChartBLService. getLineChart | 语法 | public BaseChartVO getLineChart() |
| 前置条件 | 无 |
| 后置条件 | 将本地的收款单和付款单的数据汇总，生成折线图的VO对象返回 |
| BankAccountBLService. getAllAccounts | 语法 | public List<BankAccountVO> getAllAccounts() |
| 前置条件 | 无 |
| 后置条件 | 返回所有的账号的VO对象的列表 |
| BankAccountBLService. filterAccounts | 语法 | public List<BankAccountVO> filterAccounts(List<BankAccountVO> list, String s) |
| 前置条件 | 无 |
| 后置条件 | 用字符串s对list中所有的BankAccountVO的名称筛选，将模糊匹配的对象合成另一个列表返回 |
| BankAccountBLService. addAccount | 语法 | public OperationMessage addAccount(BankAccountVO avo) |
| 前置条件 | avo已经通过检查 |
| 后置条件 | 将avo添加到本地副本，此项操作上传到服务端，返回是否成功的信息 |
| BankAccountBLService. deleteAccount | 语法 | public OperationMessage deleteAccount(BankAccountVO avo) |
| 前置条件 | avo在本地存在 |
| 后置条件 | 将avo从本地副本删除，此项操作上传到服务端，返回是否成功的信息 |
| BankAccountBLService. editAccount | 语法 | public OperationMessage editAccount(BankAccountVO avo, String newName) |
| 前置条件 | avo在本地存在，newName通过检查 |
| 后置条件 | 将本地副本的avo的名称改成新名称，此项操作上传到服务器，返回是否成功的信息 |
| BankAccountBLService.  getTradeHistory | 语法 | public List<PaymentVO > getTradeHistory(BankAccountVO avo) |
| 前置条件 | avo 在系统中存在 |
| 后置条件 | 返回与该账号相关的所有的付款单 |
| BankAccountBLService.pay | 语法 | public OperationMessage pay(String bankAccID, String amount) |
| 前置条件 | 无 |
| 后置条件 | 将bankAccID对应的账户进行支付操作，返回成功或失败原因相关信息 |
| BankAccountBLService.  receive | 语法 | public OperationMessage receive(String bankAccID, String amount) |
| 前置条件 | 无 |
| 后置条件 | 将bankAccID对应的账户进行入账操作，返回成功或失败原因相关信息 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| FormatCheckService.  checkTransportHallID  (String number) | 检查营业厅编号是否存在 | |
| FormatCheckService.  checkIsNull(String in) | 检查输入是否为空 | |
| FormatCheckService.  checkDate(String date) | 检查日期格式是否正确，是否在系统记录时间内 | |
| FormatCheckService.  checkMoney(String money) | 检查金额格式是否正确 | |
| FinanceFormDataService.  updateRevenuePOs(String staffID) | 将服务器端的staffID对应队列的收款单同步到本地副本 | |
| FinanceFormDataService.  updatePaymentPOs(String staffID) | 将服务器端的staffID对应队列的付款单同步到本地副本 | |
| FinanceFormDataService.  downloadAllPaymentPOs() | 将服务器端所有的付款单下载到本地 | |
| FinanceFormDataService.  downloadAllRevenuePOs() | 将服务器端所有的收款单下载到本地 | |
| FinanceFormDataService.  getNewRevenueID  (String date, String hallID) | 得到未使用过的收款单编号 | |
| FinanceFormDataService.  getNewPaymentID  (String date) | 得到未使用过的付款单编号 | |
| FinanceFormDataService.  getRevenuePO(String formID) | 根据收款单编号获取付款单的PO | |
| FinanceFormDataService.  getPaymentPO(String formID) | 根据付款单编号获取付款单的PO | |
| BankAccountDataService.  updateAccountOperations  (String staffID) | 将服务器端的staffID对应队列的账号操作同步到本地副本 | |
| BankAccountDataService.  uploadAccountOperations  (String staffID, List<BankAccountOperation> operations) | 将本地副本的队列的新增账号操作同步到服务器端staffID对应的队列中 | |
| BankAccountDataService.  downloadAllAccounts() | 将服务器端所有的账号下载到本地 | |
| BankAccountDataService.  getNewBankID() | 获取未使用过的银行账户ID编号 | |
| BankAccountDataService.  checkIsNameUsed  (String name) | 检查银行账户名是否已被占用 | |
| BankAccountDataService.  getBankAccount  (String bankID) | 根据传入的银行账户ID编号获取银行账户PO | |
| OrderDataService.  getOrderPO(String orderID) | 获取指定订单号的订单的数据 | |

**initbl包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| InitializationBLService.  getAllAccounts | 语法 | public List<BankAccountVO> getAllAccounts() |
| 前置条件 | 无 |
| 后置条件 | 返回所有的银行账户的VO |
| InitializationBLService.  filterAccounts | 语法 | public List<BankAccountVO> filterAccounts  (List<BankAccountVO> list, String s) |
| 前置条件 | 无 |
| 后置条件 | 将参数的s作为模糊搜索条件，对传进的账户的列表匹配，返回模糊匹配的各个账户组成的列表 |
| InitializationBLService.  addAccount | 语法 | public OperationMessage addAccount(BankAccountVO bavo) |
| 前置条件 | bavo经过检查 |
| 后置条件 | 将该对象增加到InitialDataVO对象里 |
| InitializationBLService.  deleteAccount | 语法 | public OperationMessage deleteAccount(BankAccountVO avo) |
| 前置条件 | 无 |
| 后置条件 | 将该对象从InitialDataVO对象里删除 |
| InitializationBLService.  editAccount | 语法 | public OperationMessage editAccount(BankAccountVO avo, String newName) |
| 前置条件 | 无 |
| 后置条件 | 将InitialDataVO对象里对应的账户修改 |
| InitializationBLService.  getAllStoreModels | 语法 | public List<StoreModel> getAllStoreModels() |
| 前置条件 | 无 |
| 后置条件 | 返回所有仓库模型组成的列表 |
| InitializationBLService.  selectStoreModel | 语法 | public OperationMessage selectStoreModel(StoreModel model) |
| 前置条件 | 无 |
| 后置条件 | 将传入的参数的模型作为操作模型 |
| InitializationBLService.  reducePartition | 语法 | public OperationMessage reducePartition  (StoreAreaCode area, int number) |
| 前置条件 | 已经选定有操作的仓库 |
| 后置条件 | 将指定分区的货架数减少number个 |
| InitializationBLService.  expandPartition | 语法 | public OperationMessage expandPartition(StoreAreaCode area, int number) |
| 前置条件 | 已经选定有操作的仓库 |
| 后置条件 | 将指定分区的货架数增加number个 |
| InitializationBLService.  deleteRow | 语法 | public OperationMessage deleteRow(StoreAreaCode area, int rowNum, boolean confirmed) |
| 前置条件 | 已经选定有操作的仓库 |
| 后置条件 | 指定分区减少rowNum个排 |
| InitializationBLService.  addRow | 语法 | public OperationMessage addRow(StoreAreaCode area, int initCapacity) |
| 前置条件 | 已经选定有操作的仓库 |
| 后置条件 | 将指定分区增加一个排，上面有initCapacity个货架 |
| InitializationBLService.  adjustRow | 语法 | public OperationMessage adjustRow(StoreAreaCode area, int rowNum, int newCapacity, boolean confirmed) |
| 前置条件 | 已经选定有操作的仓库 |
| 后置条件 | 将选定分区的选定排的货架数改为新的 |
| InitializationBLService.  getAllCars | 语法 | public List<CarVO> getAllCars() |
| 前置条件 | 无 |
| 后置条件 | 返回所有车辆的VO对象组成的列表 |
| InitializationBLService.  filterCarsByHall | 语法 | public List<CarVO> filterCarsByHall(String hallID) |
| 前置条件 | 无 |
| 后置条件 | 将符合营业厅编号的车辆组成列表返回 |
| InitializationBLService.  addCar | 语法 | public OperationMessage addCar(CarVO car) |
| 前置条件 | car经过检查 |
| 后置条件 | InitialDataVO里新增一个CarVO对象 |
| InitializationBLService.  modifyCar | 语法 | public OperationMessage modifyCar(CarVO car) |
| 前置条件 | car经过检查 |
| 后置条件 | 将InitialDataVO里的指定的CarVO对象修改成新的 |
| InitializationBLService.  deleteCar | 语法 | public OperationMessage deleteCar(CarVO car) |
| 前置条件 | 无 |
| 后置条件 | InitialDataVO里的指定的CarVO对象删除 |
| InitializationBLService.  getAllStaffs | 语法 | public List<StaffVO> getAllStaffs() |
| 前置条件 | 无 |
| 后置条件 | 返回所有员工对象的VO组成的列表 |
| InitializationBLService.  filterStaffsByHall | 语法 | public List<StaffVO> filterStaffsByHall(String hallID) |
| 前置条件 | 无 |
| 后置条件 | 将与传入的营业厅编号匹配的员工的VO组成列表返回 |
| InitializationBLService.  modifyStaff | 语法 | public OperationMessage modifyStaff(StaffVO after) |
| 前置条件 | after经过检查 |
| 后置条件 | 将InitialDataVO对象里的对应的StaffVO对象换成新的对象 |
| InitializationBLService.  addStaff | 语法 | public OperationMessage addStaff(StaffVO staff) |
| 前置条件 | staff经过检查 |
| 后置条件 | InitialDataVO对象里增加新的StaffVO对象 |
| InitializationBLService.  deleteStaff | 语法 | public OperationMessage deleteStaff(StaffVO staff) |
| 前置条件 | 无 |
| 后置条件 | InitialDataVO对象里删除指定的StaffVO对象 |
| InitializationBLService.  getAllCenters | 语法 | public List<CenterVO> getAllCenters(); |
| 前置条件 | 无 |
| 后置条件 | 返回所有的中转中心的VO对象组成的列表 |
| InitializationBLService.  filterCentersByNumber | 语法 | public List<CenterVO> filterCentersByNumber(String number) |
| 前置条件 | number中只包含数字 |
| 后置条件 | 将中转中心编号的前面的数字是传入的参数的营业厅的VO对象组成列表返回 |
| InitializationBLService.  addCenter | 语法 | public OperationMessage addCenter(CenterVO center) |
| 前置条件 | center经过检查 |
| 后置条件 | InitialDataVO里增加指定的CenterVO对象 |
| InitializationBLService.  deleteCenter | 语法 | public OperationMessage deleteCenter(CenterVO center) |
| 前置条件 | 无 |
| 后置条件 | InitialDataVO里删除指定的CenterVO对象 |
| InitializationBLService.  modifyCenter | 语法 | public OperationMessage modifyCenter(CenterVO center) |
| 前置条件 | center经过检查 |
| 后置条件 | InitialDataVO里将指定的CenterVO对象换成新的 |
| InitializationBLService.  getAllHalls | 语法 | public List<HallVO> getAllHalls() |
| 前置条件 | 无 |
| 后置条件 | 将所有的营业厅的对象的VO列表返回 |
| InitializationBLService.  filterHallsByNumber | 语法 | public List<HallVO> filterHallsByNumber(String number) |
| 前置条件 | number里都是数字 |
| 后置条件 | 将营业厅编号的前面的数字是传入的参数的营业厅的VO对象组成列表返回 |
| InitializationBLService.  addHall | 语法 | public OperationMessage addHall(HallVO hall) |
| 前置条件 | hall对象经过检查 |
| 后置条件 | InitialDataVO里增加新的HallVO对象 |
| InitializationBLService.  deleteHall | 语法 | public OperationMessage deleteHall(HallVO hall) |
| 前置条件 | 无 |
| 后置条件 | InitialDataVO里删除指定的HallVO对象 |
| InitializationBLService.  modifyHall | 语法 | public OperationMessage modifyHall(HallVO hall) |
| 前置条件 | hall对象经过检查 |
| 后置条件 | InitialDataVO里指定的HallVO对象替换成新的 |
| InitializationBLService.  getInitialDataVO | 语法 | public InitialDataVO getInitialDataVO(String version) |
| 前置条件 | 无 |
| 后置条件 | 返回对应的帐的版本号对应的InitialDataVO |
| InitializationBLService.  requestInitData | 语法 | public OperationMessage requestInitData() |
| 前置条件 | 无 |
| 后置条件 | 要求系统进入“建账中”状态，建账人为使用此方法的登陆的账号 |
| InitializationBLService.  uploadInitialData | 语法 | public OperationMessage uploadInitialData() |
| 前置条件 | 系统进入“建账中”状态 |
| 后置条件 | 将InitialDataVO包装成PO后上传服务器，开始期初建账 |
| InitializationBLService.  abortInitData | 语法 | public OperationMessage abortInitData() |
| 前置条件 | 系统进入“建账中”状态 |
| 后置条件 | 要求系统使用之前的帐，进入正常状态 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| BankAccountBLService.  getAllAccounts | 获取所有账户 | |
| BankAccountBLService.  filterAccounts | 模糊筛选查找账户 | |
| BankAccountBLService.  addAccount | 增加账户 | |
| BankAccountBLService.  deleteAccount | 删除账号 | |
| BankAccountBLService.  editAccount | 修改账户 | |
| StoreModelBLService.  reducePartition | 减少一个分区的货架数 | |
| StoreModelBLService.  expandPartition | 增加一个分区的货架数 | |
| StoreModelBLService.  deleteRow | 删除一排货架 | |
| StoreModelBLService.  addRow | 增加一排货架 | |
| StoreModelBLService.  adjustRow | 修改一排货架的货架数 | |
| ManageblCarService.  getCar | 获取所有车辆 | |
| ManageblCarService.  addCar | 新增一车辆 | |
| ManageblCarService.  modifyCar | 修改一车辆 | |
| ManageblCarService.  deleteCar | 删除一车辆 | |
| ManageblCenterService  getCenter | 获取所有中转中心 | |
| ManageblCenterService  addCenter | 新增一中转中心 | |
| ManageblCenterService  deleteCenter | 修改一中转中心 | |
| ManageblCenterService  modifyCenter | 删除一中转中心 | |
| ManageblHallService  getHall | 获取所有营业厅 | |
| ManageblHallService  addHall | 新增一营业厅 | |
| ManageblHallService  modifyHall | 修改一营业厅 | |
| ManageblHallService  deleteHall | 删除一营业厅 | |
| Mana getStaff  geblStaffService | 获取所有员工 | |
| ManageblStaffService  modifyStaff | 新增一员工 | |
| ManageblStaffService  addStaff | 修改一员工 | |
| ManageblStaffService  dismissStaff | 删除一员工 | |
| InitialDataService  getInitialDataPO | 获取期初信息 | |
| InitialDataService  requestInitData | 要求进入期初建账中的状态 | |
| InitialDataService  uploadInitialData | 上传期初建账模型 | |
| InitialDataService  abortInitData | 要求终止期初建账中的状态 | |

**logbl包接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| logbl.history | 语法 | public ArrayList<FormVO> getHistory() |
| 前置条件 | 无 |
| 后置条件 | 返回所有审批过的单据的展示层数据 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| transportDataService.  getTransportHistory() | 获取中转单据历史 | |
| receiveDataService.  getReceiveHistory() | 获取接收单据历史 | |
| deliverDataService.  getDeliverHistory() | 获取派送单据历史 | |
| orderDataService.  getOrderHistory() | 获取寄件单据历史 | |
| storeDataService.  getStoreHistory() | 获取库存单据历史 | |
| financeDataService.  getFinanceHistory() | 获取账目单据历史 | |

**managebl包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| manageblService.getStaff | 语法 | public ArrayList<StaffVO> getStaff(StaffTypeEnum StaffTypeEnum) |
| 前置条件 | StaffTypeEnum为员工类型的枚举 |
| 后置条件 | 根据StaffTypeEnum获取相应的所有员工信息 |
| manageblService.modifyStaff | 语法 | public OperationMessage modifyStaff(StaffPO after) |
| 前置条件 | after为修改后的信息 |
| 后置条件 | 若修改成功返回成功，否则失败原因 |
| manageblService.addStaff | 语法 | public OperationMessage addStaff(StaffPO staff) |
| 前置条件 | Staff为新员工的信息;已经获取所有员工信息 |
| 后置条件 | 若新建成功则返回成功；否则，返回失败原因 |
| manageblService.dismissStaff | 语法 | public OperationMessage dismissStaff(StaffPO Staff) |
| 前置条件 | Staff为待解雇员工信息；已经获取员工信息列表 |
| 后置条件 | 若删除成功则返回成功；否则返回失败原因 |
| manageblService.searchStaff | 语法 | public StaffPO searchStaff(StaffPO staff) |
| 前置条件 | staff为被索对象信息，不能为null |
| 后置条件 | 若查找成功，返回信息；否则返回null |
| manageblService.getCar | 语法 | public ArrayList<CarVO> getCar(HallPO itself) |
| 前置条件 | itself为发出请求的营业厅信息 |
| 后置条件 | 获取车辆信息列表 |
| manageblService.addCar | 语法 | public OperationMessage addCar(CarPO Car) |
| 前置条件 | Car为新加车辆的信息；已经获取当前营业厅车辆列表 |
| 后置条件 | 若新建成功返回成功；否则返回失败原因 |
| manageblService.modifyCar | 语法 | public OperationMessage modifyCar(CarPO after) |
| 前置条件 | after为修改后的车辆信息；已经获取当前营业厅车辆列表 |
| 后置条件 | 若修改成功返回成功；否则返回失败原因 |
| manageblService.deleteCar | 语法 | public OperationMessage deleteCar(CarPO Car) |
| 前置条件 | Car为要删除车辆的信息 |
| 后置条件 | 若删除成功，返回成功；否则，返回失败原因 |
| manageblService.searchCar | 语法 | public CarPO searchCar(CarPO car) |
| 前置条件 | car为被索对象信息，不能为null |
| 后置条件 | 若查找成功，返回信息；否则返回null |
| manageblService.getHall | 语法 | public ArrayList<HallVO> getHall() |
| 前置条件 | 无 |
| 后置条件 | 返回公司所有营业厅的信息 |
| manageblService.addHall | 语法 | public OperationMessage addHall(HallPO Hall) |
| 前置条件 | newHall为新营业厅的信息；已经获取所有的营业厅信息 |
| 后置条件 | 若新建成功，返回成功；否则，返回失败原因 |
| manageblService.deleteHall | 语法 | public OperationMessage deleteHall(HallPO hall) |
| 前置条件 | hall为待删除营业厅信息；已经获取所有的营业厅信息 |
| 后置条件 | 若删除成功返回成功，否则返回失败原因 |
| manageblService.modifyHall | 语法 | public OperationMessage modifyHall(HallPO after) |
| 前置条件 | after为修改后的营业厅信息 |
| 后置条件 | 若修改成功，返回成功；否则返回失败原因 |
| manageblService.searchHall | 语法 | public HallPO searchHall(HallPO hall) |
| 前置条件 | hall为被索营业厅信息，不能为null |
| 后置条件 | 若检索成功，返回营业厅信息；否则返回null |
| manageblService.getCenter | 语法 | public ArrayList<CenterVO> getCenter() |
| 前置条件 | 无 |
| 后置条件 | 获取所有中转中心的信息 |
| manageblService.addCenter | 语法 | public OperationMessage addCenter(CenterPO center) |
| 前置条件 | center为新建中转中心的信息 |
| 后置条件 | 若新建成功，返回成功；否则返回失败原因 |
| manageblService.deleteCenter | 语法 | public OperationMessage deleteCenter(CenterPO center) |
| 前置条件 | center为待删除中转中心信息 |
| 后置条件 | 若删除成功，返回成功，否则返回失败原因 |
| manageblService.modifyCenter | 语法 | public OperationMessage modifyCenter(CenterPO center) |
| 前置条件 | center为待修改中转中心信息 |
| 后置条件 | 若修改成功，返回成功；否则，返回失败原因 |
| manageblService.searchCenter | 语法 | public CenterPO searchCenter(CenterPO center) |
| 前置条件 | center为被索中转中心信息，不能为null |
| 后置条件 | 若检索成功，返回中转中心信息；否则返回null |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| memberDataService.  getStaff(StaffTypeEnum StaffTypeEnum) | 获取指定员工信息 | |
| memberDataService  modifyStaff(StaffPO after) | 修改员工信息 | |
| memberDataService.  dismissStaffBat(InstitutionPO institute) | 按照机构解雇员工 | |
| memberDataService.  addStaff(StaffPO staff) | 添加员工信息 | |
| memberDataService.  dismissStaff(StaffPO staff) | 解雇员工 | |
| conpanyDataService.  getHall() | 获取营业厅信息 | |
| conpanyDataService.  modifyHall(HallPO after) | 修改营业厅信息 | |
| conpanyDataService.  addHall(HallPO hall) | 新建营业厅 | |
| conpanyDataService.  deleteHall(HallPO hall) | 删除营业厅 | |
| conpanyDataService.  getCenter() | 获取中转中心信息 | |
| conpanyDataService.  modifyCenter (CenterPO after) | 修改中转中心信息 | |
| conpanyDataService.  addCenter (CenterPO center) | 新建中转中心 | |
| conpanyDataService.  deleteCenter (CenterPO center) | 删除中转中心 | |
| conpanyDataService.  getCar() | 获取车辆信息 | |
| conpanyDataService.  modifyCar(CarPO car) | 修改车辆信息 | |
| conpanyDataService.  addCar(CarPO car) | 添加车辆信息 | |
| conpanyDataService.  deleteCar(CarPO car) | 删除车辆信息 | |

**orderbl包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderBL.checkFormat | 语法 | Public List < CheckFormMessage > checkFormat(String OrderID) |
| 前置条件 | 输入的OrderID不为空 |
| 后置条件 | 检查输入信息是否符合系统要求，若符合，返回的list中只有一个显示正确的CheckFormMessage，否则返回能指出错误的CheckFormMessage的列表 |
| OrderBL.submit | 语法 | Public OperationMessage submit(OrderPO form) |
| 前置条件 | OrderPO已通过检查 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| OrderBL.saveDraft | 语法 | public OperationMessage saveDraft(OrderPO draft) |
| 前置条件 | 输入的OrderPO不为空 |
| 后置条件 | 将OrderPO序列化成文件存储 |
| OrderBL.loadDraft | 语法 | public OrderPO loadDraft() |
| 前置条件 | 有对应的序列化文件存在 |
| 后置条件 | 将序列化文件解序列化成对应的OrderPO对象返回 |
| OrderBL.predict | 语法 | Public PredictPO predict(OrderPO po) |
| 前置条件 | OrderPO已通过检查 |
| 后置条件 | 返回经计算得出的运费和预计到达时间 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| ExamineService.  submit(FormPO po) | 提交表格审批 | |
| OrderDataService.insert | 插入新订单 | |
| OrderDataService.delete | 删除订单信息 | |
| OrderDataService.update | 更新订单 | |
| OrderDataService.newID | 得到新的订单号 | |

**receivebl包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ReceiveBL.checkFormat | 语法 | public List< CheckFormMessage > checkFormat(ReceiveVO form,boolean isFinal) |
| 前置条件 | 输入一个到达单VO对象，isFinal是否为最终提交 |
| 后置条件 | 检查输入信息是否符合系统要求，若符合，返回的list中只有一个显示正确的CheckFormMessage, 否则返回能指示出错误的CheckFormMessage的列表 |
| ReceiveBL.submit | 语法 | public OperationMessage submit(ReceiveVO form) |
| 前置条件 | 参数form作为被提交的到达单VO对象，且已经通过检查 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| ReceiveBL. saveDraft | 语法 | Public　OperationMessage saveDraft (ReceiveVO form) |
| 前置条件 | 到达单没有被提交，选择保存，form 表单信息 |
| 后置条件 | 返回保存是否成功的OperationMessage |
| ReceiveBL.loadDraft | 语法 | public ReceiveVO loadDraft () |
| 前置条件 | 系统中存在ReceiveVO草稿 |
| 后置条件 | 返回相应的ReceiveVO对象 |
| ReceiveBL.getOrderVO | 语法 | public OrderVO getOrderVO(String orderID) |
| 前置条件 | 传入订单号orderID |
| 后置条件 | 返回该ID对应的VO对象 |
| ReceiveBL.getTransitVO | 语法 | public TransitVO get TransitVO () |
| 前置条件 | 已经载入了一个OrderPO对象 |
| 后置条件 | 返回当前订单对应最新中转单的VO对象 |
| 需要的服务（需接口） | | |
| FormatCheckService.checkOrderID(String number) | 检查订单号码是否存在 | |
| FormatCheckService.checkDate(String date) | 检查日期是否符合格式，是否符合系统要求 | |
| FormatCheckService.checkLocation(String location,String local) | 检查地点是否有误 | |
| FormatCheckService.checkTransitID(String number) | 检查中转单号码是否存在 | |
| ReceiveDataService.newID | 获得一个新的到达单ID | |
| ExamineService.  submit(FormPO po) | 提交表格审批 | |
| ReceiveDataService.insert | 插入单据 | |
| ReceiveDataService.delete | 删除单据 | |
| ReceiveDataService.update | 更新单据 | |
| OrderDataService.getOrderPO | 获得订单PO | |
| TransportDataService.getTransportPO | 获得中转单PO | |

**searchbl包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| SearchBL.checkFormat | 语法 | public List<CheckFormMessage> checkFormat(String OrderID) |
| 前置条件 | 输入一个订单号String 字符串 |
| 后置条件 | 检查输入信息是否符合系统要求，若符合，调用integrate整理物流信息并返回，否则返回能指出错误的CheckFormMessage的列表 |
| SearchBL. integrate | 语法 | Public LogisticsVO integrate (String OrderID) |
| 前置条件 | 输入的订单号经检查无误 |
| 后置条件 | 输出经过整理按时间排列的货运状态和历史轨迹 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| FormatCheckService.checkOrderID(String OrderID) | 检查订单号码是否存在 | |
| TransportDataService.getTransportPO(String OrderID) | 获取所需中转单/装车单信息 | |
| ReceiveDataService.getReveivePO(String OrderID) | 获取所需到达单信息 | |
| DeliverDataService. getDeliverPO(String OrderID) | 获取所需派件单信息 | |
| OrderDataService.getOrderPO(String OrderID) | 获取所需订单信息 | |
| StoreDataService.getStoreOutPO (String OrderID) | 获取所需出库单信息 | |
| StoreDataService.getStoreInPO (String OrderID) | 获取所需入库单信息 | |

**storebl包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StoreInBLService. getAvailableLocation | 语法 | public StoreLocation getAvailableLocation (StoreAreaCode area) |
| 前置条件 | 输入的区号正确 |
| 后置条件 | 若有空的位置返回其中一个，若无，返回null |
| StoreInBLService. getNewStoreInID | 语法 | public String getNewStoreInID  (String date) |
| 前置条件 | 输入的date格式正确 |
| 后置条件 | 返回当天未使用的入库单编号 |
| StoreInBLService.loadOrder | 语法 | public OperationMessage loadOrder(String orderNumber) |
| 前置条件 | 无 |
| 后置条件 | 将对应的OrderPO对象存储在本对象里，返回是否载入成功的信息 |
| StoreInBLService.getOrderVO | 语法 | public OrderVO getOrderVO() |
| 前置条件 | 已经执行过loadOrder且成功 |
| 后置条件 | 返回OrderPO对象包装过的OrderVO对象 |
| StoreInBLService.checkFormat | 语法 | public List<CheckFormMessage> checkFormat(StoreInVO form, boolean isFinal) |
| 前置条件 | 输入的StoreInVO不为null |
| 后置条件 | 检查输入信息是否符合系统要求，若符合，返回的list中只有一个显示正确的CheckFormMessage, 否则返回能指示出错误的CheckFormMessage的列表 |
| StoreInBLService.submit | 语法 | public OperationMessage submit(StoreInVO form) |
| 前置条件 | StoreInVO已经通过检查 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| StoreInBLService.saveDraft | 语法 | public OperationMessage saveDraft(StoreInVO draft) |
| 前置条件 | 输入的StoreInVO不为null |
| 后置条件 | 将StoreOutVO序列化成文件存储，成功后返回成功信息；若之前有同名文件，返回已有草稿这个信息 |
| StoreInBLService.loadDraft | 语法 | public StoreInVO loadDraft() |
| 前置条件 | 无 |
| 后置条件 | 若对应序列化文件存在，解序列化成对应的StoreInVO对象返回；否则返回null |
| StoreInBLService. clearLocalBuffer | 语法 | public OperationMessage clearLocalBuffer() |
| 前置条件 | 无 |
| 后置条件 | 清除本地的入库单数据的缓存 |
| StoreOutBLService. getNewStoreOutID | 语法 | public String getNewStoreOutID  (String date) |
| 前置条件 | date格式正确 |
| 后置条件 | 返回当天未使用过的出库单编号 |
| StoreOutBLService.  loadOrder | 语法 | public OperationMessage loadOrder(String orderNumber) |
| 前置条件 | 无 |
| 后置条件 | 将对应的OrderPO对象和StoreInPO对象存储在本对象里，返回是否载入成功的信息 |
| StoreOutBLService.  getOrderVO | 语法 | public OrderVO getOrderVO() |
| 前置条件 | 已经执行过loadOrder且成功 |
| 后置条件 | 返回OrderPO对象包装过的OrderVO对象 |
| StoreOutBLService.  getTransportVO | 语法 | public TransitVO getTransportVO() |
| 前置条件 | 已经执行过loadOrder且成功 |
| 后置条件 | 根据OrderPO里的最新的中转单/装车单编号找到中转单/装车单PO，保存到此对象，并且返回包装过的VO |
| StoreOutBLService.  checkFormat | 语法 | public List< CheckFormMessage > checkFormat(StoreOutVO form, boolean isFinal) |
| 前置条件 | 输入的StoreOutVO不为空 |
| 后置条件 | 检查输入信息是否符合系统要求，若符合，返回的list中只有一个显示正确的CheckFormMessage, 否则返回能指示出错误的CheckFormMessage的列表 |
| StoreOutBLService.  submit | 语法 | public OperationMessage submit(StoreOutVO form) |
| 前置条件 | StoreOutVO已经通过检查 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| StoreOutBLService.  saveDraft | 语法 | public OperationMessage saveDraft(StoreOutVO draft) |
| 前置条件 | 输入的StoreOutVO不为空 |
| 后置条件 | 将StoreOutVO序列化成文件存储，成功后返回成功信息；若之前有同名文件，返回已有草稿这个信息 |
| StoreOutBLService.  loadDraft | 语法 | public StoreOutVO loadDraft() |
| 前置条件 | 无 |
| 后置条件 | 若对应序列化文件存在，解序列化成对应的StoreInVO对象返回；否则返回null |
| StoreOutBLService.  clearLocalBuffer | 语法 | public OperationMessage clearLocalBuffer() |
| 前置条件 | 无 |
| 后置条件 | 清除本地的出库单数据的缓存 |
| StoreIOBLService. checkFormat | 语法 | public List<CheckFormMessage> checkFormat(String inDate, String inTime, String outDate, String outTime) |
| 前置条件 | 无 |
| 后置条件 | 检查输入信息是否符合系统要求，若符合，返回的list中只有一个显示正确的CheckFormMessage, 否则返回能指示出错误的CheckFormMessage的列表 |
| StoreIOBLService. getGoodsInfo | 语法 | public List<GoodsVO> getGoodsInfo(String inDate, String inTime, String outDate, String outTime) |
| 前置条件 | 起始时间经过检查 |
| 后置条件 | 将该时间段内的出库单和入库单信息整合成GoodsVO对象的列表，存储在此对象里，返回引用 |
| StoreIOBLService. filtGoods | 语法 | public List<GoodsVO> filterGoods(String orderNumber) |
| 前置条件 | 无 |
| 后置条件 | 将符合输入的订单号的出库单和入库单的信息整合成GoodsVO对象的列表，返回引用 |
| StockTackBLService. getStockTack | 语法 | public StockTackVO getStockTack() |
| 前置条件 | 无 |
| 后置条件 | 将当时的仓库的存储情况整合成StockTackVO对象，存储在本对象里，返回引用 |
| StockTackBLService. filtLocation | 语法 | public StockTackVO filtLocation(StoreAreaCode area, int row, int shelf, int position) |
| 前置条件 | row，shelf，position等为负数时表示忽略匹配该项 |
| 后置条件 | 将与传入的位置匹配的位置的信息整合成StockTackVO返回 |
| StockTackBLService. getOrder | 语法 | public OrderVO getOrder(String orderNumber) |
| 前置条件 | 无 |
| 后置条件 | 返回与传入订单号匹配的OrderPO对象包装过的VO对象 |
| StoreTackBLService. getStoreInVO | 语法 | public StoreInVO getStoreInVO(String orderNumber) |
| 前置条件 | 无 |
| 后置条件 | 返回与传入订单号匹配的StoreInVO对象 |
| StoreTackBLService. makeExcel | 语法 | public OperationMessage makeExcel(String path) |
| 前置条件 | 无 |
| 后置条件 | 将此时的库存盘点信息生成excel文件保存在传入的路径，返回成功信息；若路径不存在或其他错误，返回错误提醒信息 |
| StoreModelService. setWarningLine | 语法 | public OperationMessage setWarningLine(double percent) |
| 前置条件 | 无 |
| 后置条件 | 若percent范围在0-100内，更新警戒线，返回正常信息；否则返回错误提示 |
| StoreModelService. addObserver | 语法 | public void addObserver(Observer o) |
| 前置条件 | StoreModel为Observable，有UI为Observer |
| 后置条件 | 将传入的Observer对象注册到此对象上，当警报情况发生改变时通知 |
| StoreModelService. reducePartition | 语法 | public OperationMessage reducePartition(StoreAreaCode area, int number) |
| 前置条件 | 无 |
| 后置条件 | 若number输入合理，将area的货架数减少number个，机动区货架数增加number个，保存，返回正常信息；否则返回提示错误原因 |
| StoreModelService. expandPartition | 语法 | public OperationMessage expandPartition(StoreAreaCode area, int number) |
| 前置条件 | 无 |
| 后置条件 | 若number输入合理，将area的货架数增加number个，机动区货架数减少number个，保存，返回正常信息；否则返回提示错误原因 |
| StoreModelService.deleteRow | 语法 | public OperationMessage deleteRow(StoreAreaCode area, int rowNum，boolean confirmed) |
| 前置条件 | 区号和排号都正确 |
| 后置条件 | confirmed为false，若该排无货物，将此排删除，返回成功信息；若该排有货物，请求确认，confirmed为真再调用；若confirmed为true，将该排删除，返回成功信息 |
| StoreModelService.addRow | 语法 | public OperationMessage addRow(StoreAreaCode area, int initCapacity) |
| 前置条件 | 区号存在 |
| 后置条件 | 将对应架子数的排添加到对应的区 |
| StoreModelService.adjustRow | 语法 | public OperationMessage adjustRow(StoreAreaCode area, int rowNum, int newCapacity, boolean confirmed) |
| 前置条件 | 区号，排号存在 |
| 后置条件 | confirmed为false，若该排有货物架子多于新输入架子数，请求确认，confirmed为真再调用；若该排有货物架子少于新输入架子数，排的架子数调整为新架子数，返回成功信息；若该排有货物，若confirmed为true，排的架子数调整为新架子数（可删去任意架子），返回成功信息 |
| StoreModelService. clearLocalBuffer | 语法 | public OperationMessage clearLocalBuffer() |
| 前置条件 | 无 |
| 后置条件 | 清楚本地的StoreModel缓存 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| OrderDataService.  getOrderPO(String orderID) | 获取指定订单号的订单的数据 | |
| TransportDataService.  getTransportPO(String id) | 获取指定编号的装车单/中转单的数据 | |
| FormatCheckService.  checkDate(String date) | 检查日期是否符合格式，是否符合系统要求 | |
| FormatCheckService.  checkTime(String time) | 检查时间是否符合格式，是否符合系统要求 | |
| FormatCheckService.  checkMoney(String money) | 检查金额格式是否正确 | |
| FormatCheckService.  checkIsNull(String in) | 检查输入是否为空 | |
| FormatCheckService.  checkTransitID  (String number) | 检查中转单编号是否存在 | |
| FormatCheckService.  checkTruckLoadID  (String number) | 检查装车单编号是否存在 | |
| FormatCheckService.  checkOrderID(String number) | 检查订单号是否存在 | |
| ExamineService.  submit(FormPO form) | 提交表格审批 | |
| StoreFormDataService.  updateStoreInPOs  (String staffID) | 更新服务端上该中转中心的入库单的信息 | |
| StoreFormDataService.  downloadAllStoreInPOs  (String staffID) | 下载服务端上该中转中心所有的入库单 | |
| StoreFormDataService.  updateStoreOutPOs  (String staffID) | 更新服务端上该中转中心的出库单的信息 | |
| StoreFormDataService.  downloadAllStoreOutPOs  (String staffID) | 下载服务端上该中转中心所有的出库单 | |
| StoreFormDataService.  getStoreInPO (String id) | 根据单据编号取得入库单的PO | |
| StoreFormDataService.  getStoreOutPO(String id) | 根据单据编号取得出库单的PO | |
| StoreDataService.  updateModelOperations  (String centerID, String staffID) | 更新服务端上对该中转中心仓库模型的操作 | |
| StoreDataService.  uploadModelOperations  (String centerID,  String staffID,  List<StoreModelOperation>) | 上传到服务端本地staffID对该中转中心仓库模型的操作 | |

**transportbl包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| TransportHallBL. checkFormat | 语法 | public List< CheckFormMessage > checkFormat(LoadPO form, boolean isFinal) |
| 前置条件 | 输入的LoadPO不为空,isFinal判断是否为提交表格的操作 |
| 后置条件 | 检查输入信息是否符合系统要求，若符合，返回的list中只有一个显示正确的CheckFormMessage, 否则返回能指示出错误的CheckFormMessage的列表 |
| TransportHallBL.submit | 语法 | public OperationMessage submit(LoadPO form) |
| 前置条件 | LoadPO已经通过检查 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| TransportHallBL.saveDraft | 语法 | public OperationMessage saveDraft(LoadPO draft) |
| 前置条件 | 输入的LoadPO不为空 |
| 后置条件 | 将LoadPO序列化成文件存储 |
| TransportHallBL.loadDraft | 语法 | public LoadPO loadDraft() |
| 前置条件 | 有对应的序列化文件存在 |
| 后置条件 | 将序列化文件解序列化成对应的LoadPO对象返回 |
| TransportCenterBL. checkFormat | 语法 | public List<OperationMessage> checkFormat(CenterPO form) |
| 前置条件 | 输入的CenterPO不为空 |
| 后置条件 | 检查输入信息是否符合系统要求，若符合，返回的list中只有一个显示正确的OperationMessage, 否则返回能指示出错误的OperationMessage的列表 |
| TransportCenterBL.submit | 语法 | public OperationMessage submit(CenterPO form) |
| 前置条件 | CenterPO已经通过检查 |
| 后置条件 | 若网络连接正常，返回正常的OperationMessage，否则返回相应的提示信息 |
| TransportCenterBL.saveDraft | 语法 | public OperationMessage saveDraft(CenterPO draft) |
| 前置条件 | 输入的CenterPO不为空 |
| 后置条件 | 将CenterPO序列化成文件存储 |
| TransportCenterBL.loadDraft | 语法 | public CenterPO loadDraft() |
| 前置条件 | 有对应的序列化文件存在 |
| 后置条件 | 将序列化文件解序列化成对应的CenterPO对象返回 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| ExamineService.  submit(FormPO po) | 提交表格审批 | |
| TransportDataService.insert | 插入新中转单/装车单 | |
| TransportDataService.delete | 删除中转单/装车单信息 | |
| TransportDataService.update | 更新中转单/装车单信息 | |
| TransportDataService.newID | 得到新的中转单/装车单号 | |

## 5.4.数据层的分解

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的增、删、改、查。持久化数据保存采用序列化文件

### 5.4.1数据层模块的职责

数据层模块职责如下表所示

**数据层模块职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| AccountDataService | 提供账户信息的增加、删除、修改、保存、查找操作 |
| CompanyDataService | 提供营业厅、中转中心、车辆信息的增加、删除、修改、保存、查找操作 |
| ConfigurationDataService | 提供系统常量的查看、修改、保存操作 |
| DeliverDataService | 提供派件单的增加、删除、修改、保存、查找操作 |
| FinanceDataService | 提供银行账户的管理，以及收款单和付款单的的添加和查询，财务状况的可视化 |
| MemberDataService | 提供员工信息的增加、删除、修改、保存、查找操作 |
| OrderDataServer | 提供新订单信息的增加、删除、修改、保存、查找操作 |
| ReceiveDataService | 提供到达单信息的增加、删除、修改、保存、查找操作 |
| StoreDataService | 提供仓库的管理，入库单和出库单的增加、删除、修改、保存、查找操作 |
| TransportDataService | 提供装车单/中转单信息的增加、删除、修改、保存、查找操作 |

### 5.4.2.数据层模块的接口规范

数据层模块的接口规范如下表

**数据层模块的接口规范**

**AccountData包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| AccountDataService. getAccountPO | 语法 | public AccountPO getAccountPO (String accountID) |
| 前置条件 | AccountID在数据中存在 |
| 后置条件 | 查找返回相应的AccountPO |
| AccountDataService.getAccountPOs | 语法 | public AccountPO getAccountPOs() |
| 前置条件 | 无 |
| 后置条件 | 返回所有的AccountPO |
| AccountDataService.insert | 语法 | Public OperationMessage insert  (AccountPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| AccountDataService.delete | 语法 | public OperationMessage delete  (String name) |
| 前置条件 | 同样的po在数据中存在,传入name |
| 后置条件 | 删除po，返回结果信息 |
| AccountDataService.update | 语法 | public OperationMessage update  (AccountPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| AccountDataService.checkID | 语法 | public Boolean checkID(String accountID) |
| 前置条件 | 无 |
| 后置条件 | 检查ID是否存在于数据中，若是，则返回true，否则返回false |
| AccountDataService.checkAccount | 语法 | Public boolean checkAccount(String id , String password) |
| 前置条件 | CheckID返回true |
| 后置条件 | 若ID与password匹配，则返回true，否则，返回false |
| OrderDataService.newAccountID | 语法 | public String newAccountID() |
| 前置条件 | 无 |
| 后置条件 | 返回一个新的账户名 |

**CompanyData包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| companyDataService.getHall | 语法 | public ArrayList<HallPO> getHall() |
| 前置条件 | 无 |
| 后置条件 | 返回公司所有营业厅的信息 |
| companyDataService.newHallID | 语法 | public String newHallID(String centerID) |
| 前置条件 | 上属中转中心编号 |
| 后置条件 | 返回一个合法的新编号 |
| companyDataService.addHall | 语法 | public OperationMessage addHall(HallPO Hall) |
| 前置条件 | Hall为新营业厅的信息 |
| 后置条件 | 若新建成功，返回成功；否则，返回失败原因 |
| companyDataService.deleteHall | 语法 | public OperationMessage deleteHall(HallPO hall) |
| 前置条件 | hall为待删除营业厅信息；已经获取所有的营业厅信息 |
| 后置条件 | 若删除成功返回成功，否则返回失败原因 |
| companyDataService.modifyHall | 语法 | public OperationMessage modifyHall(HallPO hall) |
| 前置条件 | hall为修改后的营业厅信息 |
| 后置条件 | 若修改成功，返回成功；否则返回失败原因 |
| companyDataService.getCenter | 语法 | public ArrayList<CenterPO> getCenter() |
| 前置条件 | 无 |
| 后置条件 | 获取所有中转中心的信息 |
| companyDataService.addCenter | 语法 | public OperationMessage addCenter(CenterPO center) |
| 前置条件 | center为新建中转中心的信息 |
| 后置条件 | 若新建成功，返回成功；否则返回失败原因 |
| companyDataService.deleteCenter | 语法 | public OperationMessage deleteCenter(CenterPO center) |
| 前置条件 | center为待删除中转中心信息 |
| 后置条件 | 若删除成功，返回成功，否则返回失败原因 |
| companyDataService.modifyCenter | 语法 | public OperationMessage modifyCenter(CenterPO center) |
| 前置条件 | center为待修改中转中心信息 |
| 后置条件 | 若修改成功，返回成功；否则，返回失败原因 |
| companyDataService.getCar | 语法 | public ArrayList<CarPO> getCar() |
| 前置条件 | 无 |
| 后置条件 | 返回公司车辆信息 |
| companyDataService.newCarID | 语法 | public String newCarID() |
| 前置条件 | 无 |
| 后置条件 | 返回一个合法的新编号 |
| companyDataService.addCar | 语法 | public OperationMessage addCar(CarPO car) |
| 前置条件 | car为新增车辆的信息 |
| 后置条件 | 若新建成功则返回成功，否则返回失败原因 |
| companyDataService.deleteCar | 语法 | public OperationMessage deleteCar(CarPO car) |
| 前置条件 | car为待删除的车辆信息 |
| 后置条件 | 若删除成功返回成功，否则返回失败原因 |
| companyDataService.modifyCar | 语法 | public OperationMessage modifyCar(CarPO car) |
| 前置条件 | car为修改后的车辆信息 |
| 后置条件 | 若修改成功则返回成功；否则返回失败原因 |

**ConfigurationData包接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| configurationDataService.getCityDistance | 语法 | public ArrayList<CityDistancePO> getCityDistance() |
| 前置条件 | 无 |
| 后置条件 | 返回城市距离信息 |
| configurationDataService.modifyCityDistance | 语法 | public OperationMessage modifyCityDistance(CityDistancePO after) |
| 前置条件 | after为修改后的信息 |
| 后置条件 | 若修改成功返回成功，否则失败原因 |
| configurationDataService.getSalaryStrategy | 语法 | public ArrayList<SalaryStrategyPO> getSalaryStrategy () |
| 前置条件 | 无 |
| 后置条件 | 返回员工的薪水策略信息 |
| configurationDataService.modifySalaryStrategy | 语法 | public OperationMessage modifySalaryStrategy (SalaryStrategyPO SalaryStrategy) |
| 前置条件 | SalaryStrategy为修改的薪水策略 |
| 后置条件 | 若修改成功则返回成功；否则返回失败原因 |
| configurationDataService.getPack | 语法 | public ArrayList<PackPO>  getPack() |
| 前置条件 | 无 |
| 后置条件 | 返回包装费用信息 |
| configurationDataService.modifyPack | 语法 | public OperationMessage modifyPack(PackPO pack) |
| 前置条件 | pack为修改后的包装信息 |
| 后置条件 | 若修改成功返回成功；否则返回失败原因 |
| configurationDataService.getPrice | 语法 | public PricePO getPrice() |
| 前置条件 | 无 |
| 后置条件 | 返回当前快递价格 |
| configurationDataService.modifyPrice | 语法 | public OperationMessage modifyPrice(PricePO price) |
| 前置条件 | price为修改后价格信息 |
| 后置条件 | 若修改成功返回成功；否则返回失败原因 |
| configurationDataService.getProportion | 语法 | public ProportionPO getProportion () |
| 前置条件 | 无 |
| 后置条件 | 返回不同快递收费比例 |
| configurationDataService.modifyProportion | 语法 | public OperationMessage modifyProportion(ProportionPO proportion) |
| 前置条件 | proportion为修改后的比例 |
| 后置条件 | 若修改成功，返回成功；否则返回失败原因 |
| configurationDataService. getInstitutionDistance | 语法 | public Object getInstitutionDistance() |
| 前置条件 | 无 |
| 后置条件 | 返回各个机构的距离信息 |
| configurationDataService. newInstitutionDistanceSearch | 语法 | public Object[] newInstitutionDistanceSearch(String ID) |
| 前置条件 | 传入新建机构的编号 |
| 后置条件 | 返回需要填写的距离信息Sea |
| configurationDataService. newInstitutionDistanceInsert | 语法 | public OperationMessage newInstitutionDistanceInsert(String ID,Object[] ob) |
| 前置条件 | ID为新机构编号，ob为相关距离信息 |
| 后置条件 | 返回操作结果 |
| configurationDataService. modifyInstitutionDistance | 语法 | public OperationMessage modifyInstitutionDistance(String ID,Object ob) |
| 前置条件 | ID为修改的机构编号 |
| 后置条件 | ob为修改内容 |

**DeliverData包接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| DeliverDataService.insert | 语法 | public OperationMessage insert  (DeliverPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| DeliverDataService.getDeliverPO | 语法 | public DeliverPO getDeliverPO(String id) |
| 前置条件 | 参数为派件单号 |
| 后置条件 | 查找返回相应的结果 |
| DeliverDataService.delete | 语法 | public OperationMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| DeliverDataService.update | 语法 | public OperationMessage update  (DeliverPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| DeliverDataService.init | 语法 | public OperationMessage init(String id) |
| 前置条件 | 到达单号 |
| 后置条件 | 初始化持久数据(清空单) |
| DeliverDataService.newID | 语法 | public String newID() |
| 前置条件 | 无 |
| 后置条件 | 返回一个新的派件单ID |
| DeliverDataService. clear | 语法 | public OperationMessage clear () |
| 前置条件 | 无 |
| 后置条件 | 清空DeliverPO数据 |
| DeliverDataService. show | 语法 | public ArrayList< DeliverPO >  show () |
| 前置条件 | 无 |
| 后置条件 | 返回所有派件单PO数据 |

**FinanceData包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| FinanceFormDataService. updateRevenuePOs | 语法 | public LinkedList<RevenuePO>  updateRevenuePOs(String staffID) |
| 前置条件 | 无 |
| 后置条件 | 返回服务端staffID对应的队列中的收款单数据的对象 |
| FinanceFormDataService. updatePaymentPOs | 语法 | public LinkedList <PaymentPO>  updatePaymentPOs(String staffID) |
| 前置条件 | 无 |
| 后置条件 | 返回服务端staffID对应的队列中的付款单数据的对象 |
| FinanceFormDataService. downloadAllPaymentPOs | 语法 | public LinkedList <PaymentPO>  downloadAllPaymentPOs() |
| 前置条件 | 无 |
| 后置条件 | 将所有收款单的数据下载到本地 |
| FinanceFormDataService. downloadAllRevenuePOs | 语法 | public LinkedList <RevenuePO>  downloadAllRevenuePOs() |
| 前置条件 | 无 |
| 后置条件 | 将所有付款单的数据下载到本地 |
| FinanceFormDataService. | 语法 | publicRevenuePO getRevenuePO(String formID) |
| 前置条件 | 无 |
| 后置条件 | 返回传入的收款单编号对应的RevenuePO |
| FinanceFormDataService. getPaymentPO | 语法 | public PaymentPO getPaymentPO(String formID) |
| 前置条件 | 无 |
| 后置条件 | 返回传入的入款单编号对应的PaymentPO |
| FinanceFormDataService. getNewRevenueID | 语法 | public String getNewRevenueID  (String date, String hallID) |
| 前置条件 | date和hallID格式正确 |
| 后置条件 | 返回新的未使用过的收款单ID |
| FinanceFormDataService. getNewPaymentID | 语法 | public String getNewPaymentID  (String date) |
| 前置条件 | date格式正确 |
| 后置条件 | 返回新的未使用过的收款单ID |
| BankAccountDataService. | 语法 | public BankAccountPO getBankAccount(String bankID); |
| 前置条件 | 无 |
| 后置条件 | 返回传入的银行账号id对应的银行账户 |
| BankAccountDataService. updateAccountOperations | 语法 | public List<BankAccountOperation>  updateAccountOperations(String staffID) |
| 前置条件 | 无 |
| 后置条件 | 返回服务端staffID对应的队列中未同步的对账户进行的操作 |
| BankAccountDataService. uploadAccountOperations | 语法 | public OperationMessage  uploadAccountOperations  (String staffID, LinkedList<BankAccountOperation> operations) |
| 前置条件 | 无 |
| 后置条件 | 将本地队列中未上传的对账户的操作上传到服务端 |
| BankAccountDataService. downloadAllAccounts | 语法 | public List<BankAccountPO >  downloadAllAccounts() |
| 前置条件 | 无 |
| 后置条件 | 下载所有的账户的数据 |
| BankAccountDataService. getNewBankID | 语法 | public String getNewBankID() |
| 前置条件 | 无 |
| 后置条件 | 返回未使用过的银行账户ID号 |
| BankAccountDataService.  checkIsNameUsed | 语法 | public boolean  checkIsNameUsed(String name) |
| 前置条件 | 无 |
| 后置条件 | 检查传入的账户名是否已经在使用 |

**MemberData包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| memberDataService.getStaff | 语法 | public ArrayList<StaffVO> getStaff(StaffTypeEnum StaffTypeEnum) |
| 前置条件 | StaffTypeEnum为员工类型的枚举 |
| 后置条件 | 根据StaffTypeEnum获取相应的所有员工信息 |
| memberDataService.modifyStaff | 语法 | public OperationMessage modifyStaff(StaffPO after) |
| 前置条件 | after为修改后的信息 |
| 后置条件 | 若修改成功返回成功，否则失败原因 |
| memberDataService.addStaff | 语法 | public OperationMessage addStaff(StaffPO staff) |
| 前置条件 | Staff为新员工的信息;已经获取所有员工信息 |
| 后置条件 | 若新建成功则返回成功；否则，返回失败原因 |
| memberDataService.dismissStaff | 语法 | public OperationMessage dismissStaff(StaffPO Staff) |
| 前置条件 | Staff为待解雇员工信息；已经获取员工信息列表 |
| 后置条件 | 若删除成功则返回成功；否则返回失败原因 |
| memberDataService.newStaffID | 语法 | public String newStaffID(StaffTypeEnum type) |
| 前置条件 | type为新员工的工种 |
| 后置条件 | 返回合法的员工ID |

**OrderData包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderDataService.getOrderPO | 语法 | public OrderPO getOrderPO(String id) |
| 前置条件 | 参数为快递单号 |
| 后置条件 | 查找返回相应的结果 |
| OrderDataService.insert | 语法 | public OperationMessage insert  (OrderPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| OrderDataService.delete | 语法 | public OperationMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| OrderDataService.update | 语法 | public OperationMessage update  (OrderPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| OrderDataService.newID | 语法 | public String newID() |
| 前置条件 | 无 |
| 后置条件 | 返回一个新的订单号 |
| OrderDataService. clear | 语法 | public void clear () |
| 前置条件 | 无 |
| 后置条件 | 清空OrderPO数据 |

**ReceiveDataService接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ReceiveDataService.insert | 语法 | public OperationMessage insert  (ReceivePO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| ReceiveDataService.getReceivePO | 语法 | public ReceivePO gerReceivePO(String id) |
| 前置条件 | 参数为到达单号 |
| 后置条件 | 查找返回相应的结果 |
| ReceiveDataService.delete | 语法 | public OperationMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| ReceiveDataService.update | 语法 | public OperationMessage update  (ReceivePO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| ReceiveDataService.newID | 语法 | public String newID() |
| 前置条件 | 无 |
| 后置条件 | 返回一个新的到达单ID |
| ReceiveDataService. clear | 语法 | public OperationMessage clear () |
| 前置条件 | 无 |
| 后置条件 | 清空ReceivePO数据 |
| ReceiveDataService. show | 语法 | public ArrayList< ReceivePO >  show () |
| 前置条件 | 无 |
| 后置条件 | 返回所有到达单PO数据 |
| ReceiveDataService.init | 语法 | public OperationMessage init(String id) |
| 前置条件 | 到达单号 |
| 后置条件 | 初始化持久数据(清空单) |

**StoreData包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StoreFormDataService.  getStoreInPO | 语法 | public StoreInPO getStoreInPO  (String id) |
| 前置条件 | 无 |
| 后置条件 | 返回id对应的入库单的数据 |
| StoreFormDataService.  getStoreOutPO | 语法 | public StoreOutPO getStoreOutPO  (String id) |
| 前置条件 | 无 |
| 后置条件 | 返回id对应的入库单的数据 |
| StoreFormDataService. updateStoreInPOs | 语法 | public List<StoreInPO>  updateStoreInPOs(String staffID) |
| 前置条件 | 无 |
| 后置条件 | 返回staffID对应的服务器队列中的所有的入库单数据 |
| StoreFormDataService. downloadAllStoreInPOs | 语法 | public List<StoreInPO> downloadAllStoreInPOs  (String centerID) |
| 前置条件 | 无 |
| 后置条件 | 返回centerID对应的所有的入库单 |
| StoreFormDataService. updateStoreOutPOs | 语法 | public List<StoreOutPO> updateStoreOutPOs  (String centerID) |
| 前置条件 | 无 |
| 后置条件 | 返回staffID对应的服务器队列中的所有的出库单数据 |
| StoreFormDataService. downloadAllStoreOutPOs | 语法 | public List<StoreOutPO> downloadAllStoreOutPOs  (String staffID) |
| 前置条件 | 无 |
| 后置条件 | 返回centerID对应的所有的出库单 |
| StoreModelDataService. updateModelOperations | 语法 | public List<StoreModelOperation>  updateModelOperations  (String centerID, String staffID) |
| 前置条件 | 无 |
| 后置条件 | 返回对应staffID队列里的未更新的对仓库做的操作 |
| StoreModelDataService. uploadModelOperations | 语法 | public OperationMessage  uploadModelOperations  (String centerID, String staffID,  List<StoreModelOperation> operations) |
| 前置条件 | 无 |
| 后置条件 | 将对仓库做的操作上传到服务器 |
| StoreModelDataService. downloadStoreModel | 语法 | public StoreModel  downloadStoreModel  (String centerID) |
| 前置条件 | 无 |
| 后置条件 | 下载仓库的数据 |

**TransportData包的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| TransportDataService.insert | 语法 | public OperationMessage insert  (TransportPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| TransportDataService.getTransportPO | 语法 | public TransportPO getTransportPO (String id) |
| 前置条件 | 参数为快递单号 |
| 后置条件 | 查找返回相应的结果 |
| TransportDataService.delete | 语法 | public OperationMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| TransportDataService.update | 语法 | public OperationMessage update  (TransportPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| TransportDataService.newID | 语法 | public String newID() |
| 前置条件 | 无 |
| 后置条件 | 返回一个新的装车单/中转单编号 |
| TransportDataService. clear | 语法 | public void clear () |
| 前置条件 | 无 |
| 后置条件 | 清空TransportPO数据 |

# 6.信息视角

## 6.1.数据持久化对象

系统的PO、VO类就是对应的相关的实体类，如下所示：

**数据结构：AccountPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | ID | 用户名 |
| String | password | 密码 |
| AuthorityEnum | authority | 用户权限 |

AuthorityEnum: HAVE , DONT\_HAVE

**数据结构：BankAccountPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | bankID | 该账户的ID(不可修改) |
| String | accountName | 银行账户名（可以修改） |
| String | balance | d+.dd形式的字符串，指示余额，单位元 |

**数据结构：BankAccountOperation**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| BankAccountOperationType | operationType | 操作类型 |
| String | bankID | 账户标识，用于寻找账户 |
| String | para\_name | 操作参数：名字 |
| String | para\_balance | 操作参数：余额 |

**数据结构：BankAccountOperationType**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| enum | ADD\_ACC | 增加账户 |
| enum | DELETE\_ACC | 删除账户 |
| enum | EDIT\_ACC\_NAME | 修改账户名称 |
| enum | PAY | 修改账户余额，支出 |
| enum | RECEIVE | 修改账户余额，汇入 |

**数据结构：CarPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| boolean | free | 是否可用 |
| int | carID | 车辆代号 |
| int | engineID | 发动机号 |
| int | nameID | 车辆号 |
| int | chassisID | 底盘号 |
| calendar | buyTime | 购买时间 |
| calendar | useTime | 服役时间 |
| imageicon | imag | 车辆图片 |

**数据结构：CenterPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| string | centerID | 中转中心编号 |
| string | city | 所属城市 |
| ArrayList<StaffPO> | storeman | 下属库存管理员 |
| ArrayList<StaffPO> | counterman | 下属业务员 |

**数据结构：CityDistancePO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| ConfigurationEnum | ID | 用于标识这是城市间距离 |
| string | city1 | 一个城市 |
| string | city2 | 另一个城市 |
| int | distance | 距离 |

**数据结构：DeliverPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | orderID | 订单号 |
| String | data | 派送日期 |
| String | postman | 派件员 |

**数据结构：FormPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| FormEnum | formType | 用于标识表单类型 |
| FormStateEnum | state | 用于表示表单状态 |
| string | formID | 表单编号 |

FormEnum：

FormStateEnum：DRAFT、SUBMIT、PASS

**数据结构：HallPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| string | hallID | 营业厅编号 |
| string | city | 所属城市 |
| string | area | 所属区/县 |
| ArrayList<StaffPO> | driver | 下属司机信息 |
| ArrayList<StaffPO> | deliver | 下属快递员 |
| ArrayList<StaffPO> | counterman | 下属业务员 |
| String | nearCenter | 最近的中转中心编号 |

**数据结构：InitialDataPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | version | 该帐对应的版本id |
| String | dbName | 该帐对应的数据库的名字 |
| List<StoreModel> | storeModels | 仓库数据列表 |
| List<BankAccountPO> | bankAccounts | 银行账户数据列表 |
| List<CarPO> | cars | 车辆数据列表 |
| List<StaffPO> | staffs | 员工数据列表 |
| List<HallPO> | halls | 营业厅数据列表 |
| List<CenterPO> | centers | 中转中心数据列表 |

**数据结构：OrderPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | nameFrom | 寄件人姓名 |
| String | nameTo | 收件人姓名 |
| String | Location | 寄件人地址 |
| String | Local | 收件人地址 |
| String | UnitFrom | 寄件人单位 |
| String | UnitTo | 收件人单位 |
| String | PhoneNumFrom | 寄件人手机号码 |
| String | PhoneNumTo | 收件人手机号码 |
| String | telNumFrom | 寄件人电话号码 |
| String | telNumTo | 收件人电话号码 |
| String | goodsNum | 货物内件数 |
| String | GoodsName | 货物内件名 |
| String | Weight | 货物重量 |
| String | Volume | 货物体积 |
| String | Money | 包装费 |
| TypeEnum | Standard | 快递标准 |

TypeEnum :SLOW , NORMAL , FAST

**数据结构：PackPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| ConfigurationEnum | ID | 用于标识这是一个包装费的PO |
| Map< PackEnum type, double price> | packPrice | 包装与费用对应 |

PackEnum:PAPER;WOOD;PACKAGE;OTHER

**数据结构：PaymentPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | date | yyyy-mm-dd格式的字符串，指示日期 |
| String | amount | d+.dd形式的字符串，指示金额，单位元 |
| String | payerAccID | 付款账户ID |
| String | payerName | 付款人姓名 |
| String | payerAccount | 付款账户名 |
| String | receiverAccID | 收款账户ID |
| String | receiverName | 收款人姓名 |
| String | receiverAccount | 收款账户名 |
| String | item | 条目 |
| String | note | 备注，可以空 |

**数据结构：PricePO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| ConfigurationEnum | ID | 用于标识这是快递价格 |
| Map<DeliveryType type, int price> | price | 快递价格  可修改的只有USUAL型的 |

DeliveryType:USUAL;FAST;ECONOMIC

**数据结构：ProportionPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| ConfigurationEnum | ID | 用于标识这是价格比例 |
| Map<DeliveryType type, int proportion> | proportion | 价格比例 |

**数据结构：ReceivePO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | orderID | 订单号 |
| String | transitID | 中转单号 |
| String | data | 到达日期 |
| String | depature | 出发地 |
| StateEnum | state | 货物到达状态 |

StateEnum : Damage,Complete,Lose

**数据结构：RevenuePO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | date | yyyy-mm-dd格式的字符串，指示日期 |
| String | amount | d+.dd形式的字符串，指示金额，单位元 |
| String | deliverName | 只包含中英文和空格，指示快递员名字 |
| String | hallID | 营业厅编号 |
| String | orderID | 订单号 |

**数据结构：SalaryStrategyPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| ConfigurationEnum | ID | 用于标识这是一个薪水策略 |
| StaffTypeEnum | staff | 标识员工类型 |
| int | base | 基本工资 |
| int | commission | 计件提成 |
| int | bonus | 特殊奖金 |

**数据结构：StaffPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| StaffTypeEnum | staff | 用于标识员工工种 |
| string | ID | 员工编号 |
| string | name | 姓名 |
| int | age | 年龄 |
| string | personID | 身份证号 |
| string | sex | 性别 |
| string | love | 情感状况 |

**数据结构：StoreInPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | orderID | 快递编号 |
| String | date | 入库日期 |
| String | destination | 目的地 |
| StoreLocation | location | 库存位置 |

**数据结构：StoreOutPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | orderID | 快递编号 |
| String | date | 出库日期 |
| String | destination | 目的地 |
| TransportationEnum | transportation | 装运方式 |
| String | transID | 中转单编号或快递单编号 |

**数据结构：LogisticsVO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | nameFrom | 寄件人姓名 |
| String | nameTo | 收件人姓名 |
| ArrayList<String> | location | 历史到达地列表 |
| ArrayList<String> | time | 历史到达时间列表 |
| String | local | 现在所在地/状态 |
| String | phoneNumTo | 收件人手机号码 |
| String | goodsNum | 货物件数 |
| String | goodsName | 货物名称 |
| TypeEnum | type | 快递类型 |

**数据结构：LoadPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | loadDate | 装车日期 |
| String | loadID | 本中转中心汽运编号 |
| String | placeTo | 到达地（营业厅） |
| String | truckID | 车辆代号 |
| String | peopleSee | 监装员 |
| String | peopletransport | 押运员 |
| ArrayList<String> | IDs | 本次装箱所有订单条形码号 |
| int | expense | 运费 |

**数据结构：CenterPO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | LoadDate | 装车日期 |
| String | LoadID | 本中转中心汽运编号 |
| String | placeFrom | 出发地 |
| String | placeTo | 到达地 |
| String | TransID | 航班号/火车列次号/汽车编号 |
| String | peopleSee | 监装员 |
| String | shelfNum | 货柜号 |
| ArrayList<String> | IDs | 本次装箱所有订单条形码号/托运单号 |

**数据结构：BaseChartVO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | title | 图表标题 |
| String | mainXType | 主X轴的类型描述 |
| String | mainYType | 主Y轴的类型描述 |
| double | ySpan | y轴的分刻度 |
| double | ybegin | y轴的起始刻度 |
| double | yend | y轴的终止刻度 |
| String[] | x1Classes | 主x轴的类型分类 |
| String[] | x2Classes | 对比项的分类 |
| double[][] | values | 第一个选取对比项，第二个选取x轴位置 |

**数据结构：GoodsVO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| OrderVO | orderVO | 货物对应的订单的VO的引用 |
| StoreInVO | storeInVO | 货物对应的入库单的VO的引用 |
| StoreOutVO | storeOutVO | 货物对应的出库单的VO的引用 |

**数据结构：PieChartVO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | title | 饼状图的标题 |
| String | valueType | 描述比较的项目的类型，比如支出，收入 |
| Map<String, Double> | originMap | 饼状图的每个项的名字及原始数值 |
| Map<String,Double> | ratioMap | 饼状图的每个项的名字及最终比例 |

**数据结构：StockTackVO**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| String | date | 日期 |
| String | id | 该日期的序号 |
| List<StoreLocation> | locations | 筛选出的符合的货架位置 |
| List<String> | filters | 已经添加的筛选条件 |

**其他数据类型：**

**数据结构：Transportation**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| enum | AIR | 空运 |
| enum | RAIL | 铁路运输 |
| enum | ROAD | 公路运输 |

**数据结构：StoreAreaCode**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| enum | AIR | 空运区 |
| enum | RAIL | 铁路运输区 |
| enum | ROAD | 公路运输区 |
| enum | FLEX | 机动区 |

**数据结构：StoreLocation**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| StoreAreaCode | area | 分区号 |
| int | row | 排号，负数代表全选 |
| int | shelf | 货架号，负数代表全选 |
| int | position | 位号，负数代表全选 |
| String | orderID | 货物订单号，若为null表示无货物 |

**数据结构：StoreArea**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| List<StoreRow> | rows | 存储该区的排的列表 |
| StoreAreaCode | areaID | 该区的名称 |

**数据结构：StoreRow**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| List<StoreShelf> | shelves | 存储该排的货架的列表 |
| int | rowID | 该排的序号 |

**数据结构：StoreShelf**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| List<StoreLocation> | locations | 存储该货架的StoreLocation的列表 |
| int | shelfID | 该货架的序号 |

**数据结构：StoreModel**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| Map<StoreAreaCode, StoreArea> | areasMap | 主键为区号，值为对应的区的对象 |
| double | warningLine | 警戒线 |
| Map<StoreAreaCode,Boolean> | warningStates | 每个区是否为警戒状态 |

**数据结构：StoreModelOperation**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| StoreModelOperationType | operationType | 提示操作类型 |
| StoreLocation | location | 提示操作的对象 |
| int | size | 操作的数值大小 |

**数据结构：StoreModelOperationType**

|  |  |  |
| --- | --- | --- |
| 类型 | 名称 | 描述 |
| enum | ADD\_ROW | 增加排 |
| enum | DELETE\_ROW | 减少排 |
| enum | EDIT\_ROW | 修改排的架子数 |
| enum | REDUCE\_PART | 分区货架移出 |
| enum | EXPAND\_PART | 分区货架移入 |

## 6.2文件格式

统一用\*.2333格式，比如client.2333

## 7用户界面层的分解

系统存在8类44个用户界面，他们分为：

查询物流信息界面；

快递员界面（快递员主界面，查询物流信息界面，输入新订单界面，查看派送单界面，输入收件信息界面）；

营业厅业务员界面（营业厅业务员主界面，填写到达单界面，填写装车单界面，填写派件单界面，填写收款单界面，管理车辆信息界面，管理司机信息界面）；

财务会计界面（财务会计主界面，查看收款单界面，新建付款单界面，期初建账界面，查询期初信息界面，查看财务报表界面，银行账户管理界面）；

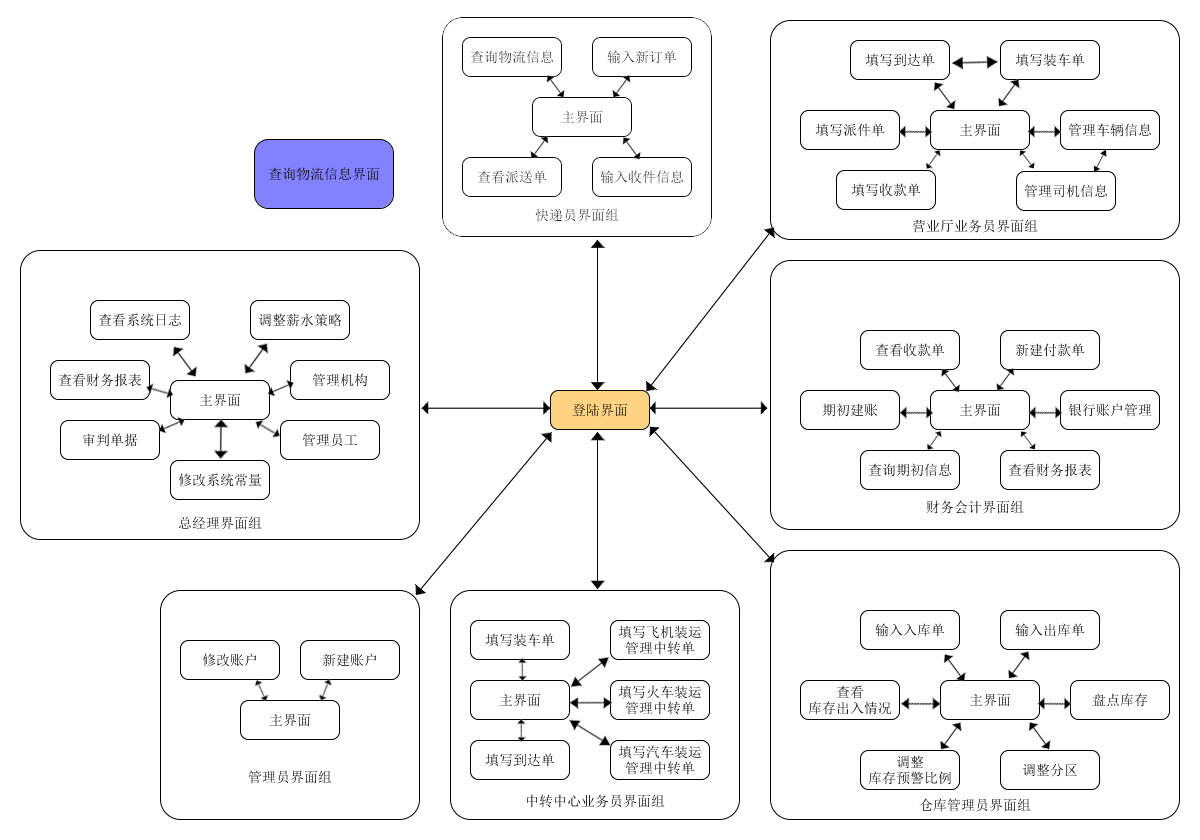
仓库管理员界面（仓库管理员主界面，输入入库单界面，输入出库单界面，盘点库存界面，调整分区界面，调整库存预警比例界面，查看库存出入情况界面）；

中转中心业务员界面（中转中心业务员主界面，填写装车单界面，填写到达单界面，填写飞机装运管理中转单界面，填写火车装运管理中转单界面，填写汽车装运管理中转单界面）；

管理员界面（管理员主界面，修改账户界面（包括查看和删除功能），新建账户界面）；

总经理界面（总经理主界面，查看系统日志界面，调整薪水策略界面，管理机构界面，管理员工界面，修改系统常量界面，审判单据界面，查看财务报表界面）

\*注：其中每个用户的主界面几乎相同，在此只给出快递员的主界面作为样例，总经理和财务人员的“查看财务报表界面”和“查看系统日志界面”几乎相同，只在财务人员处给出，不再赘述



## 登录界面

* + 1. 导航设计

针对登录任务，可以设计下列独立界面或组件

用户登录任务：进行用户登录的主要内容独立界面

用户名与密码输入：两个独立组件

输入错误提示：独立组件

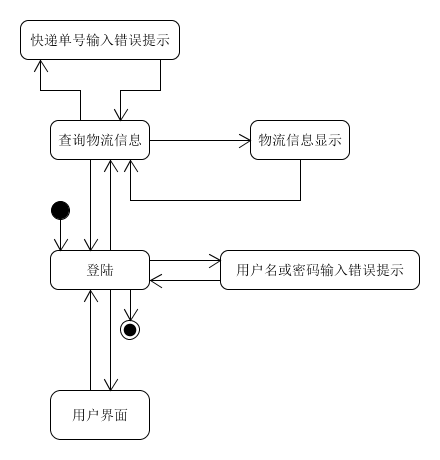
查询物流信息任务：进行物流信息查询的内容独立界面

快递单号输入：独立组件

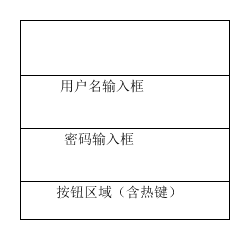
快递单号输入错误提示：独立组件

物流信息显示：独立组件

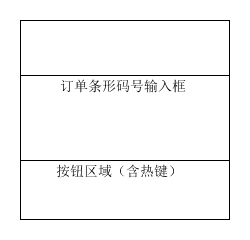
此部分内容结构如下：



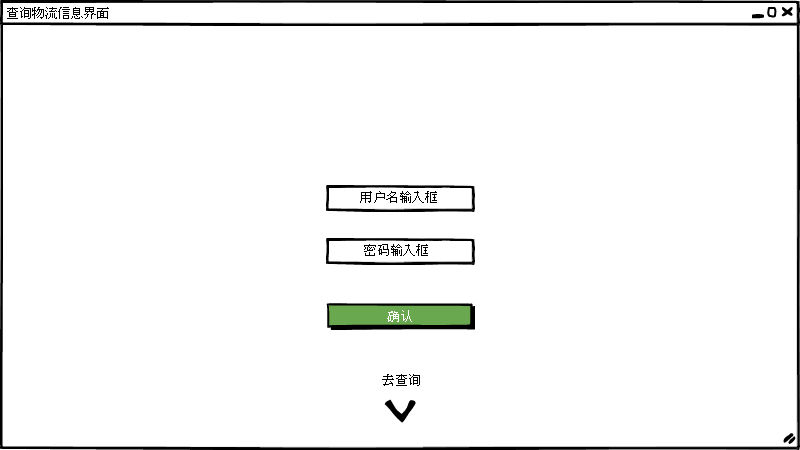
* + 1. 界面设计与原型化



登陆界面设计



查询物流信息界面设计



登陆界面原型



查询物流信息界面原型

* 1. **管理账户信息界面**
     1. 导航设计

针对管理员管理账户信息任务，可以设计下列独立界面或组件

用户名输入：为了搜索某个用户信息，独立组件

用户名输入错误提示：独立界面

删除用户：独立按钮组件

删除失败提示：独立界面

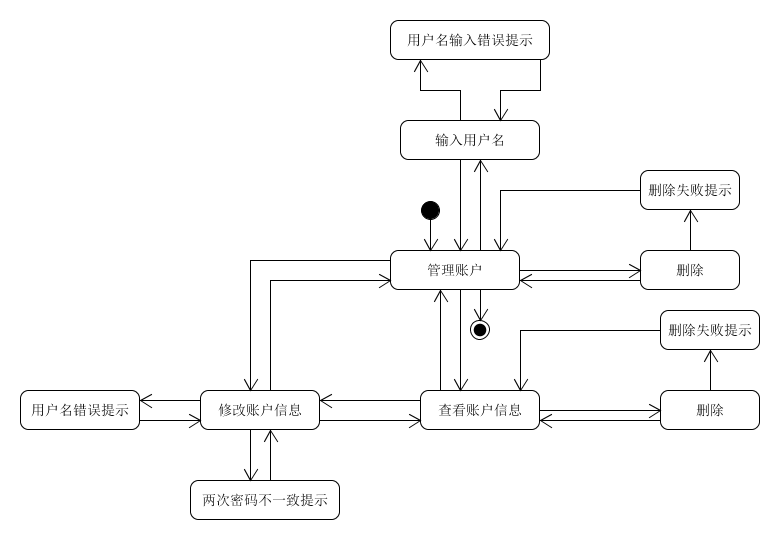
查看账户信息：双击表格中的账户信息，独立按钮组件

修改账户信息：独立按钮组件

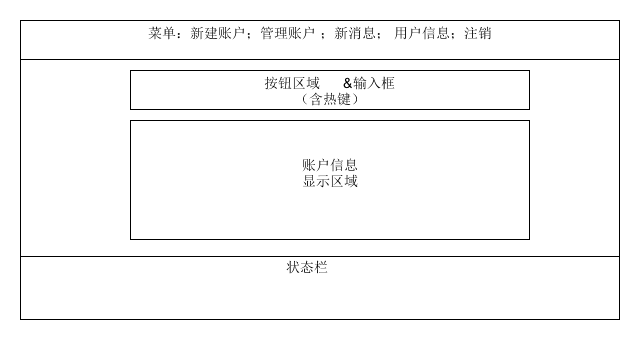
用户名错误提示：独立界面

两次密码不一致提示：独立界面

此部分内容结构如下：



* + 1. 界面设计与原型化



管理账户信息界面设计



管理账户信息原型化主页面

* 1. **管理员新建账户界面**
     1. 导航设计

针对管理员新建账户任务，可以设计下列独立界面或组件

快递单号输入：独立组件

快递单号输入错误提示：独立界面

清空：独立按钮组件

清空失败提示：独立界面

保存草稿：独立按钮组件

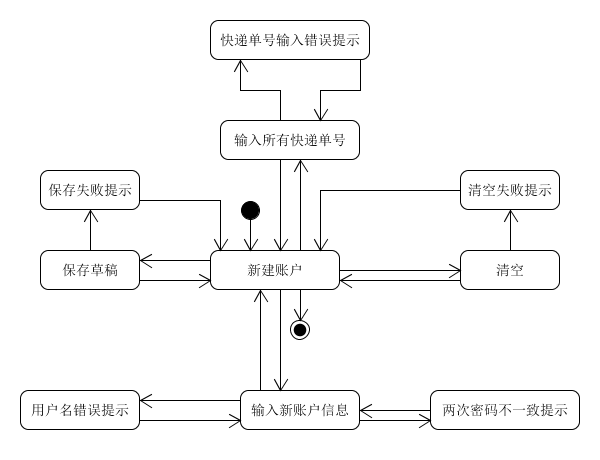
保存失败提示：独立界面

输入新账户信息：独立组件

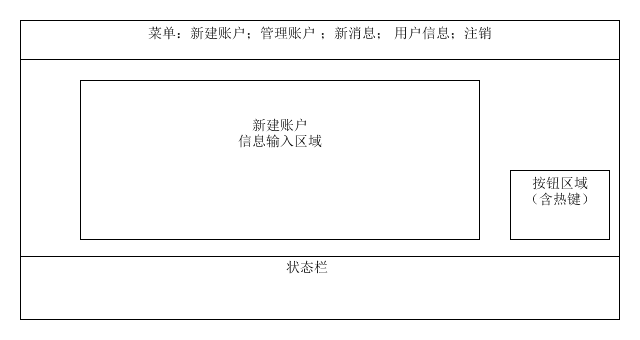
用户名错误提示：独立界面

两次密码不一致提示：独立界面

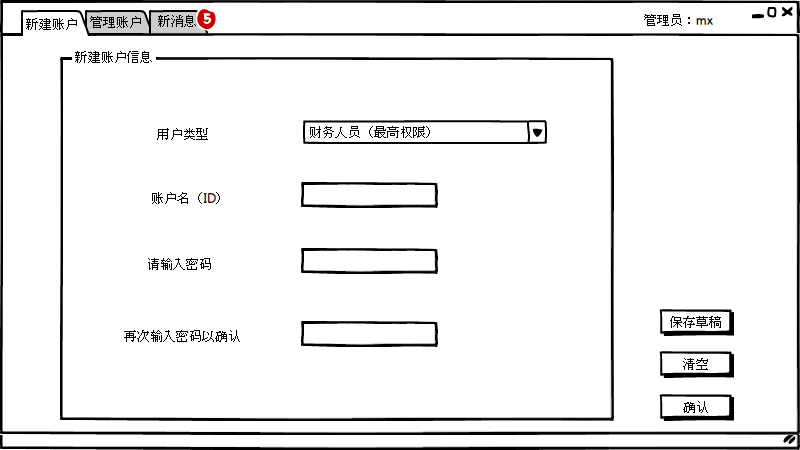
此部分内容结构如下：



* + 1. 界面设计与原型化



管理员新建账户界面设计



管理员新建账户原型化主页面

* 1. **中转中心业务员填写到达单界面**
     1. 导航设计

针对中转中心业务员填写到达单任务，可以设计下列独立界面或组件

查询快递信息：独输入想要查询的快递单号，一个独立组件

快递单号输入错误提示：独立界面

快递信息显示：显示填写到达单需要的快递信息

清空：清空所填信息，一个独立组件

清空失败提示：一个独立界面

保存草稿：一个独立组件

保存失败提示：一个独立界面

输入到达单信息：一个独立组件

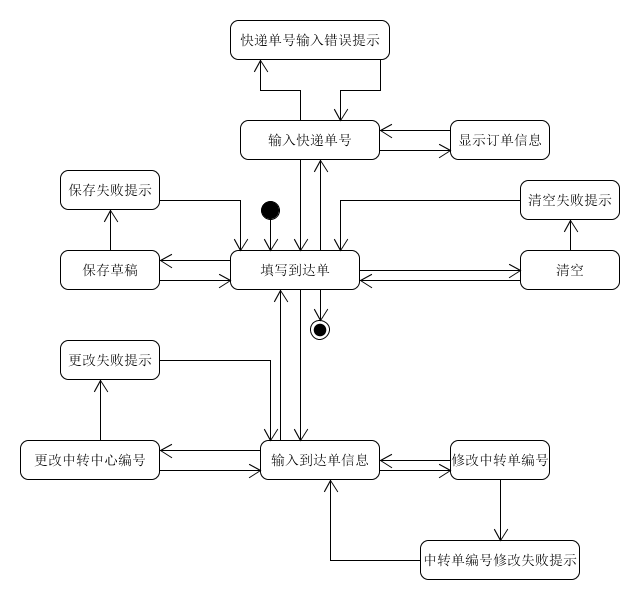
更改中转单编号：一个独立组件

更改失败提示：一个独立界面

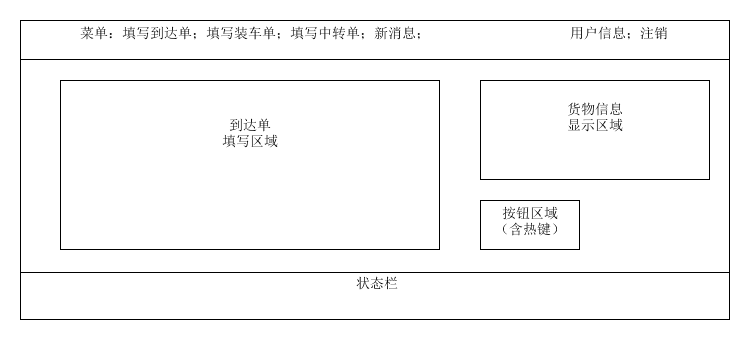
更改到达收件时间：一个独立组件

更改失败提示：一个独立界面

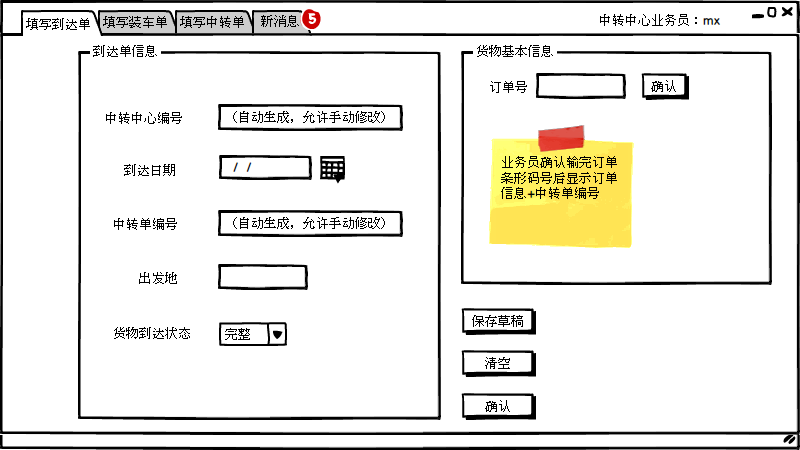
此部分内容结构如下：



* + 1. 界面设计与原型化



中转中心业务员填写到达单界面设计



中转中心业务员填写到达单原型化主页面

* 1. **中转中心业务员填写中转单界面**
     1. 导航设计

针对中转中心业务员填写中转单任务，可以设计下列独立界面或组件

输入所有快递单号：输入本次装车所有的快递单号，一个独立组件

快递单号输入错误提示：一个独立界面

清空：清空所填信息，一个独立按钮组件

清空失败提示：一个独立界面

保存草稿：一个独立按钮组件

保存失败提示：一个独立界面

输入中转单单信息：一个独立组件

输入飞机装运管理中转单信息：一个独立按钮组件

输入火车装运管理中转单信息：一个独立按钮组件

输入汽车装运管理中转单信息：一个独立按钮组件

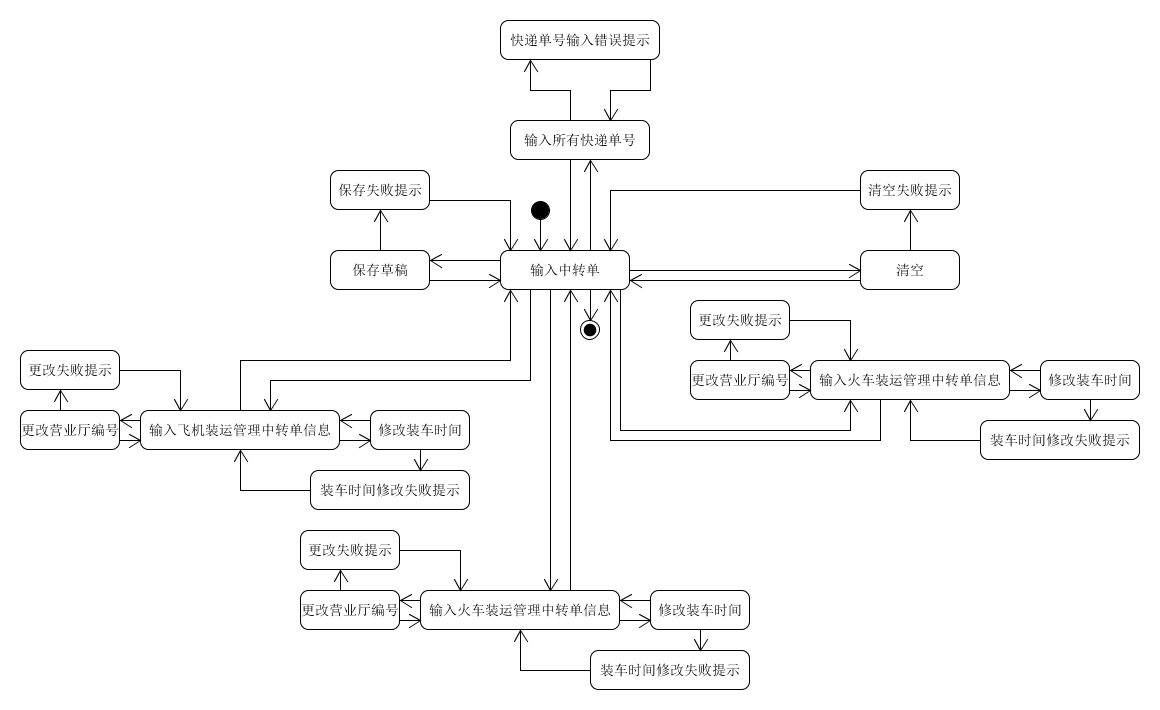
更改营业厅编号：一个独立组件

更改失败提示：一个独立界面

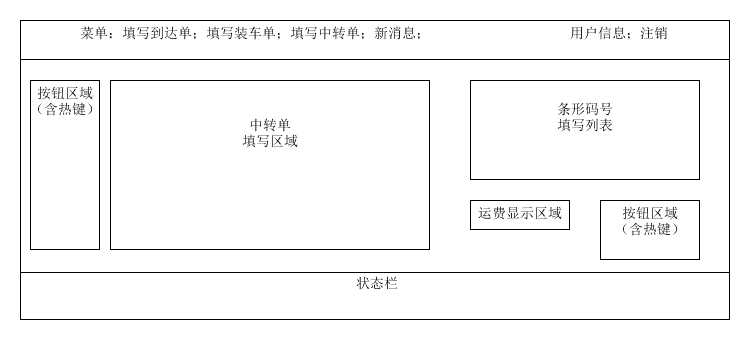
更改装车时间：一个独立组件

更改失败提示：一个独立界面

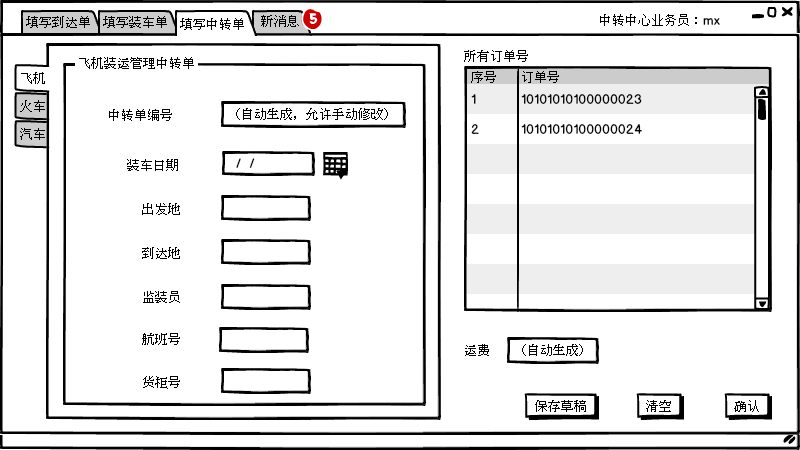
此部分内容结构如下：



* + 1. 界面设计与原型化



中转中心业务员填写中转单界面设计



中转中心业务员填写中转单原型化主页面

* 1. **中转中心业务员填写装车单界面**
     1. 导航设计

针对中转中心业务员填写装车单任务，可以设计下列独立界面或组件

输入所有快递单号：输入本次装车所有的快递单号，一个独立组件

快递单号输入错误提示：一个独立界面

清空：清空所填信息，一个独立组件

清空失败提示：一个独立界面

保存草稿：一个独立组件

保存失败提示：一个独立界面

输入装车单信息：一个独立组件

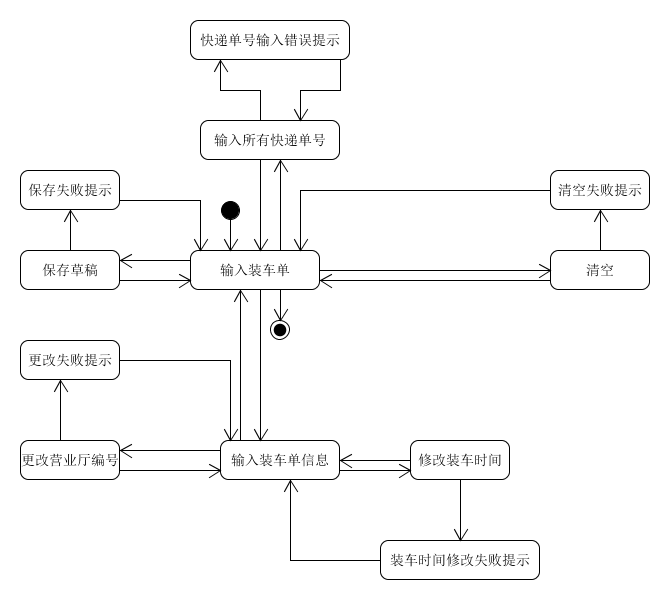
更改营业厅编号：一个独立组件

更改失败提示：一个独立界面

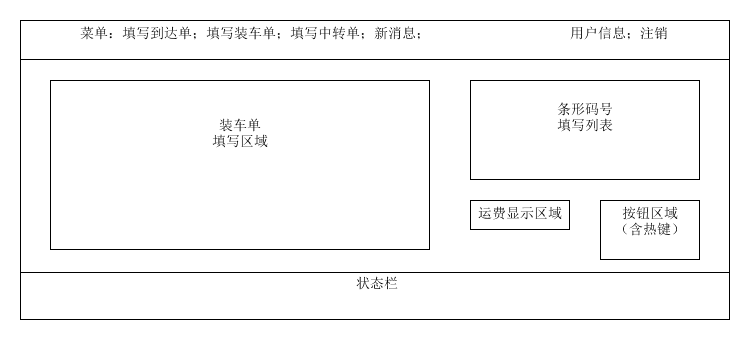
更改装车时间：一个独立组件

更改失败提示：一个独立界面

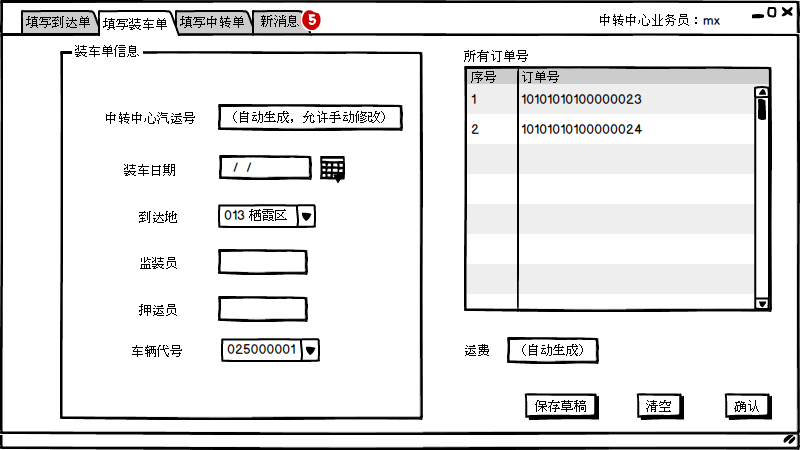
此部分内容结构如下：



* + 1. 界面设计与原型化



中转中心业务员填写装车单界面设计



中转中心业务员填写装车单原型化主页面

* 1. **查看财务报表主界面**
     1. 导航设计

针对财务人员和总经理查看财务报表任务，可以设计下列独立界面或组件

查看财务报表：财务人员查看财务报表的独立界面

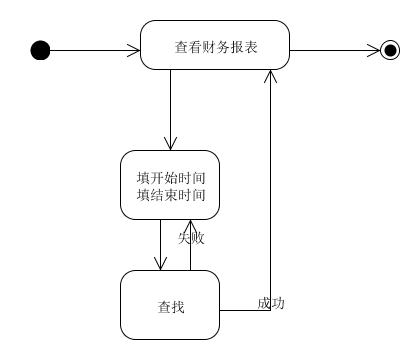
起始时间选择：选择查看财务报表的起始时间，两个独立组件

查看：刷新财务报表的显示，一个独立按钮组件

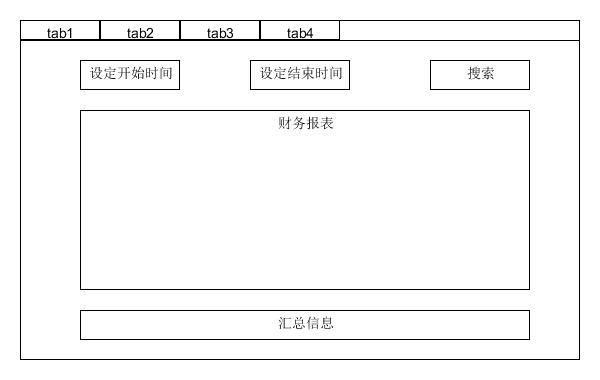
显示表格：显示财务报表的信息，一个独立组件

汇总：显示财务报表的信息的汇总，一个独立组件

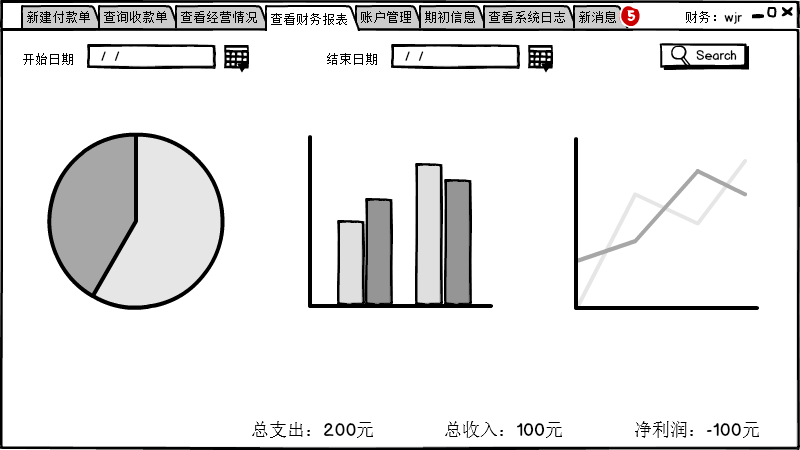
此部分内容结构如下：



* + 1. 界面设计与原型化



查看财务报表界面设计



查看财务报表原型化主页面

* 1. **查看经营情况主界面**
     1. 导航设计

针对财务人员和总经理查看经营情况任务，可以设计下列独立界面或组件

查看经营情况：财务人员查看经营情况的独立界面

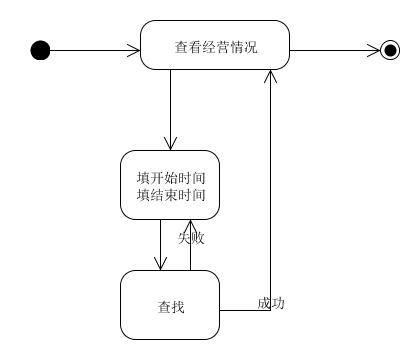
起始时间选择：选择查看经营情况的起始时间，两个独立组件

查看：刷新经营情况的显示，一个独立按钮组件

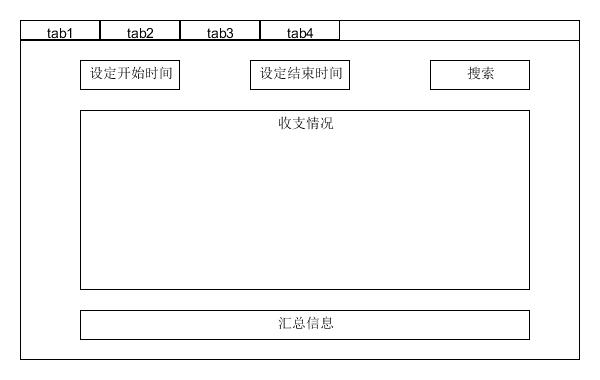
显示表格：显示经营情况的信息，一个独立组件

汇总：显示经营情况的信息的汇总，一个独立组件

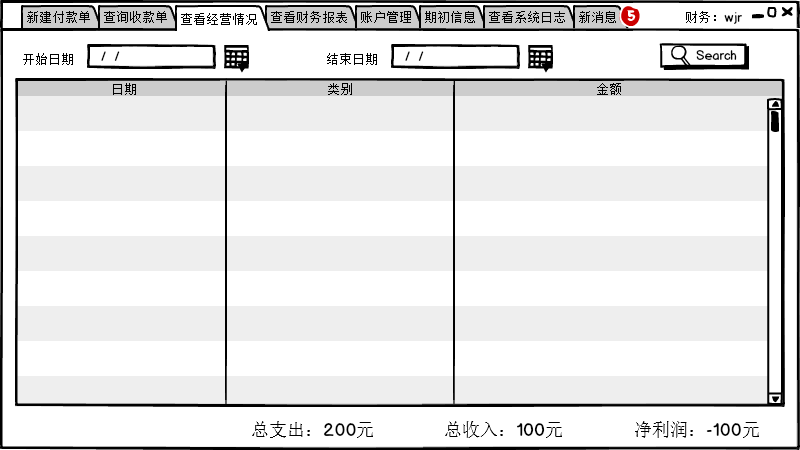
此部分内容结构如下：



* + 1. 界面设计与原型化



查看经营情况界面设计



查看经营情况原型化主页面

* 1. **查看期初信息主界面**
     1. 导航设计

针对财务人员查看期初信息任务，可以设计下列独立界面或组件

查看期初信息：财务人员查看期初信息的独立界面

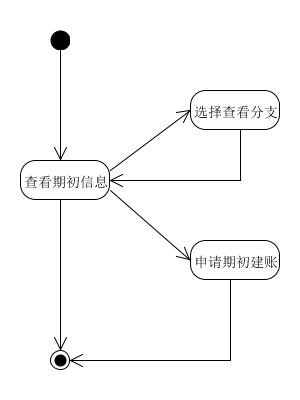
系统状态：显示系统当前状态，一个独立组件

分支选择：选择要查看的分支信息，一个独立组件

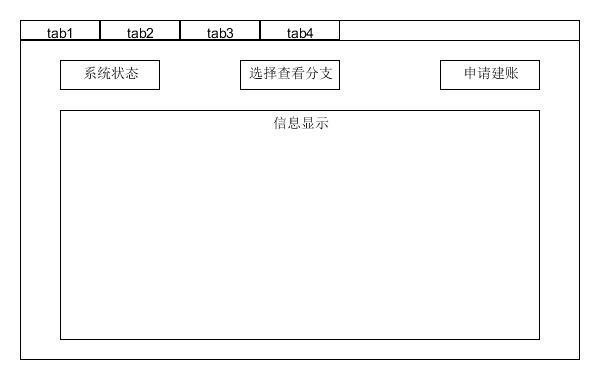
申请建账：发起期初建账的请求，一个独立按钮组件

显示表格：显示选择的分支的期初信息，一个独立组件

此部分内容结构如下：



* + 1. 界面设计与原型化



查看期初信息界面设计



查看期初信息原型化主页面

* 1. **查询收款单主界面**
     1. 导航设计

针对财务人员查询收款单任务，可以设计下列独立界面或组件

查询收款单：财务人员查询收款单的独立界面

营业厅：填写营业厅号，并提示输入的错误信息，一个独立组件

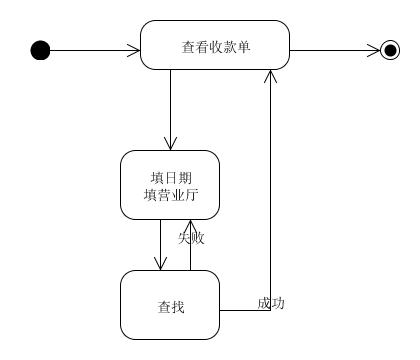
付款日期：选择查询的收款单的日期，一个独立组件

搜索：刷新收款单的显示，一个独立按钮组件

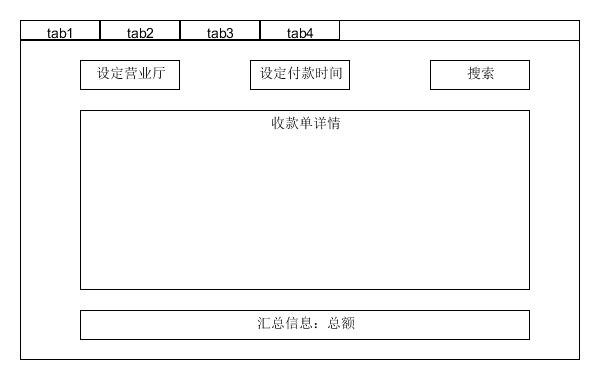
显示列表：显示收款单的信息，一个独立组件

汇总：显示金额的汇总，一个独立组件

此部分内容结构如下：



* + 1. 界面设计与原型化



查询收款单界面设计



查询收款单原型化主页面

* 1. **查看系统日志主界面**
     1. 导航设计

针对财务人员和总经理查看系统日志任务，可以设计下列独立界面或组件

查看系统日志：财务人员查看系统日志的独立界面

关键字：输入日志相关信息，一个独立组件

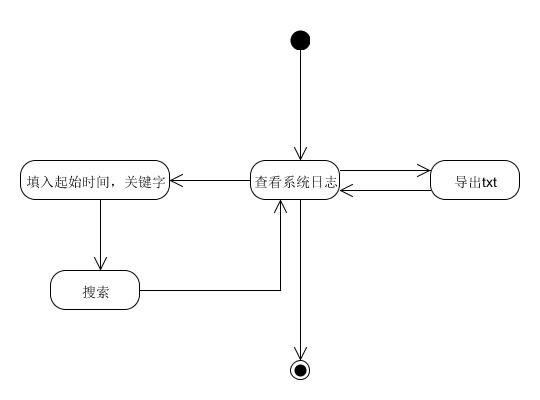
起始时间选择：选择查看系统日志的起始时间，两个独立组件

查看：刷新系统日志的显示，一个独立按钮组件

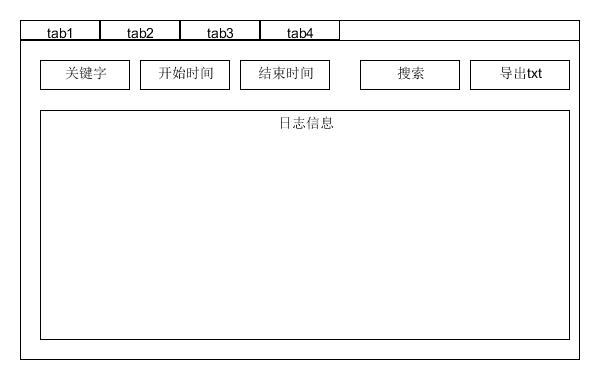
导出txt：导出txt文件，一个独立按钮组件

显示表格：显示系统日志的信息，一个独立组件

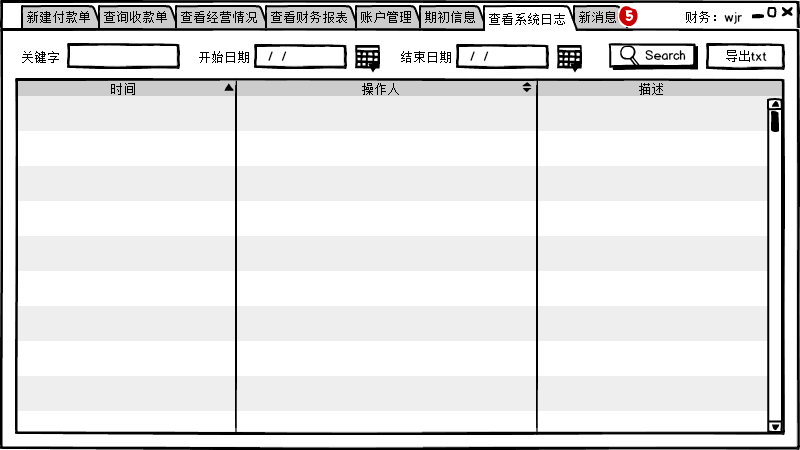
此部分内容结构如下：



* + 1. 界面设计与原型化



查看系统日志界面设计



查看系统日志原型化主页面

* 1. **新建付款单主界面**
     1. 导航设计

针对财务人员新建付款单任务，可以设计下列独立界面或组件

新建付款单：财务人员新建付款单的独立界面

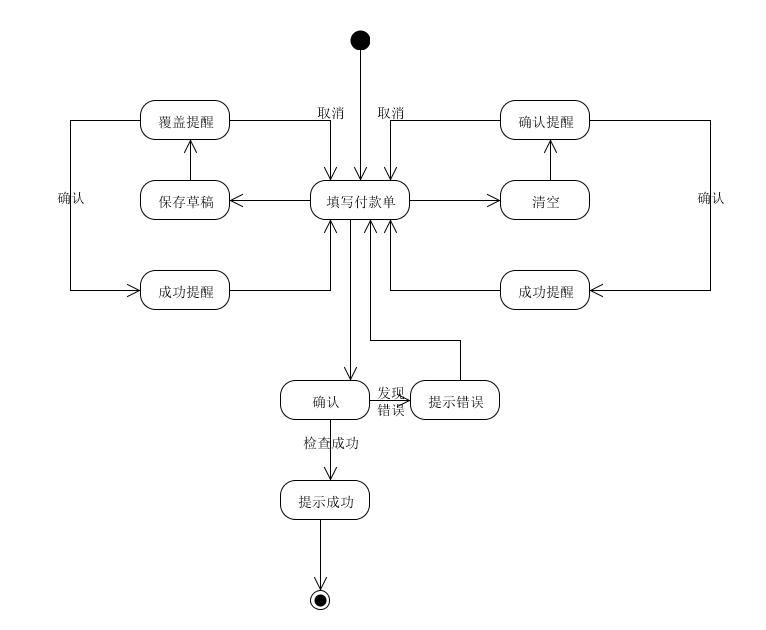
付款人信息，收款人信息，款项信息：填写付款单相关信息，并提示输入的错误信息，三个独立组件

保存草稿：发起草稿保存的独立按钮组件

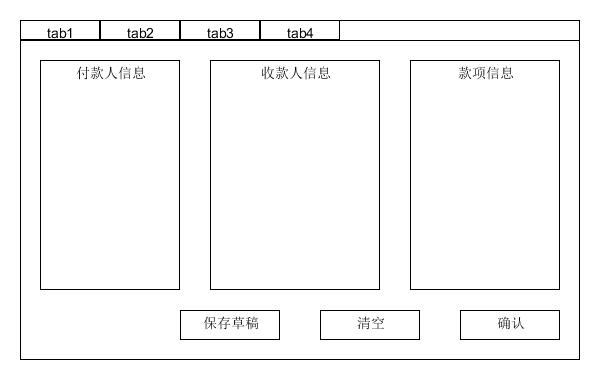
清空：清空所有输入的独立按钮组件

确认：进行入库单提交的独立按钮组件

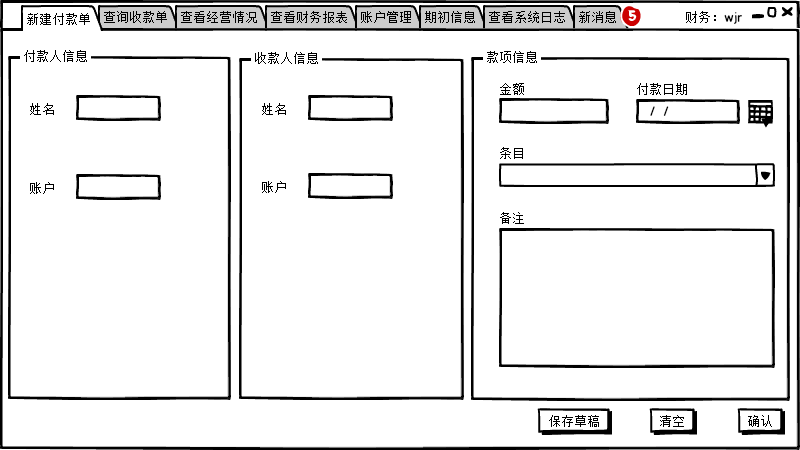
此部分内容结构如下：



* + 1. 界面设计与原型化



新建付款单界面设计



新建付款单原型化主页面

* 1. **期初建账主界面**
     1. 导航设计

针对财务人员期初建账任务，可以设计下列独立界面或组件

期初建账：财务人员期初建账的独立界面

系统状态：显示系统当前的状态，一个独立组件

仓库管理：进入仓库管理子界面的独立按钮组件

账户管理：进入账户管理子界面的独立按钮组件

人员管理：进入人员管理子界面的独立按钮组件

车辆管理：进入车辆管理子界面的独立按钮组件

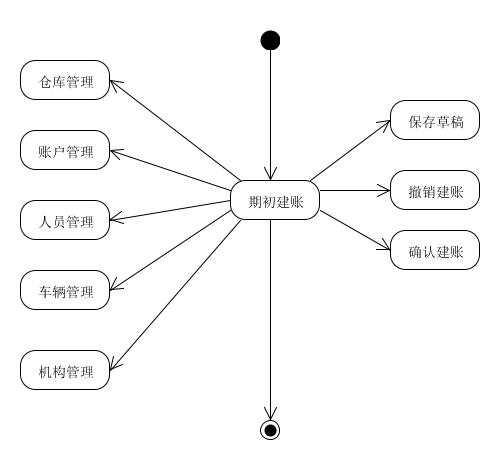
机构管理：进入机构管理子界面的独立按钮组件

保存草稿：发起草稿保存的独立按钮组件

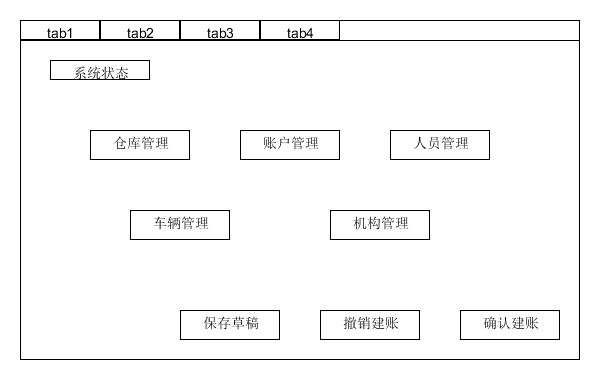
撤销建账：撤销建账的独立按钮组件

确认建账：进行期初建账提交的独立按钮组件

此部分内容结构如下：



* + 1. 界面设计与原型化



期初建账界面设计



期初建账原型化主页面

* 1. **账户（银行）管理主界面**
     1. 导航设计

针对财务人员管理账户的任务，可以设计下列独立界面或组件

管理账户界面：财务人员管理账户的独立界面

账户名输入：输入账户名相关的信息，一个独立组件

查看：刷新账户列表的显示，一个独立按钮组件

账户列表：显示账户名和余额的列表，一个独立组件

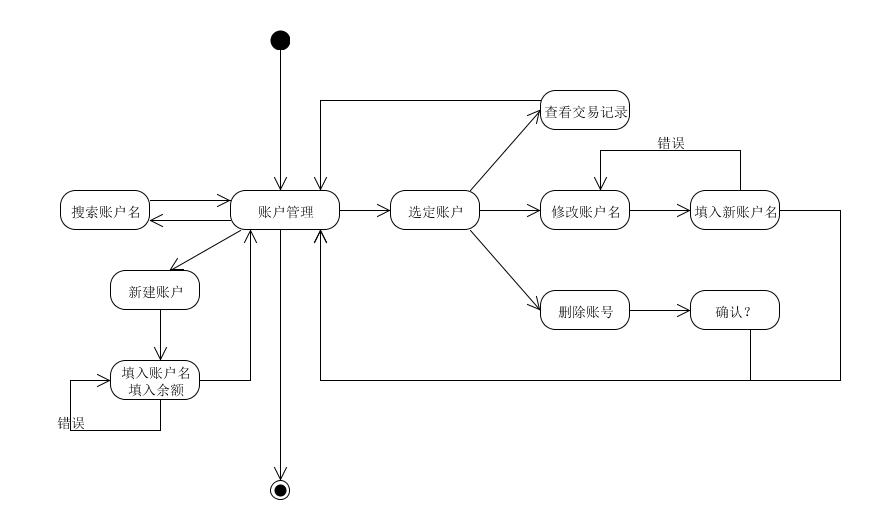
新建：发起新建账户的请求，一个独立按钮组件

删除：发起删除账户的请求，一个独立按钮组件

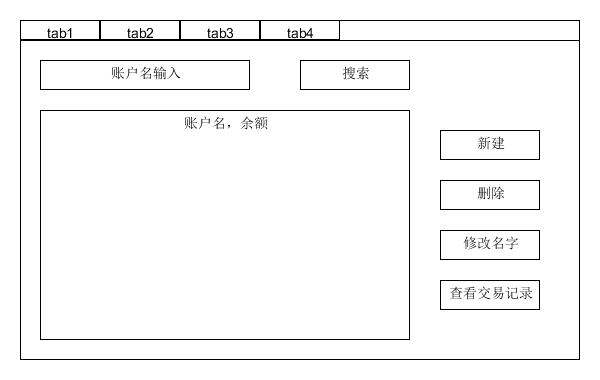
修改名字：发起修改账户名字的请求，一个独立按钮组件

查看交易记录：发起查看账户交易记录的请求，一个独立按钮组件

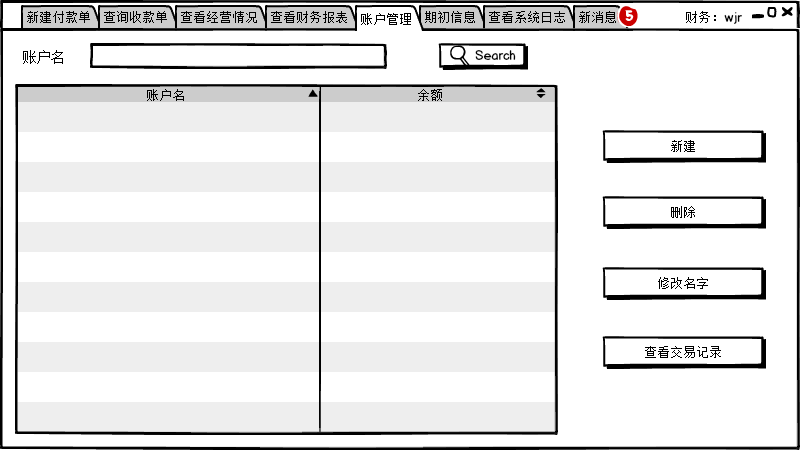
此部分内容结构如下：



* + 1. 界面设计与原型化



管理银行账户界面设计



管理银行账户原型化主页面

* 1. **营业厅业务员管理车辆司机信息界面**
     1. 导航设计

针对营业厅业务员管理车辆司机信息任务，可以设计下列独立界面或组件

查询车辆信息：独输入想要查询的汽车编号，一个独立组件

汽车编号输入错误提示：一个独立界面

车辆信息显示：显示车辆信息，一个独立组件

删除：删除所选车辆/司机信息，一个独立按钮组件

删除失败提示：一个独立界面

修改：一个独立按钮组件

修改失败提示：一个独立界面

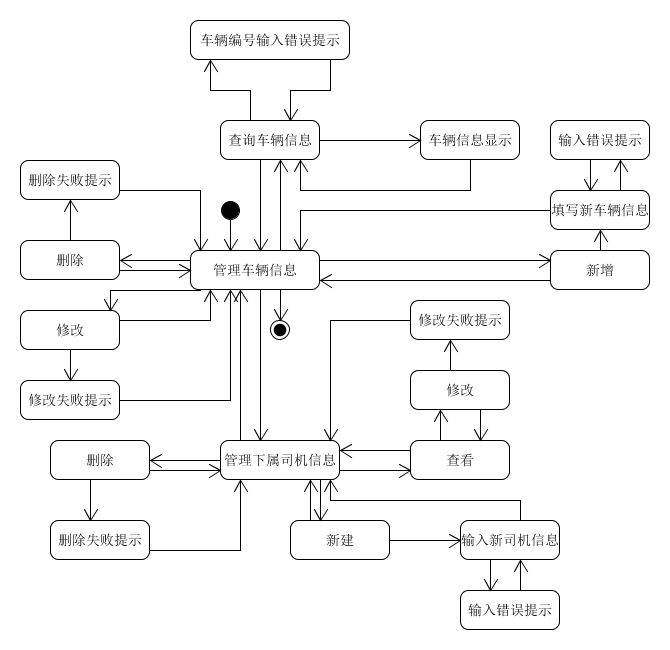
查看：双击信息即可一个独立按钮组件

新增：新增车辆/司机信息，一个独立按钮组件

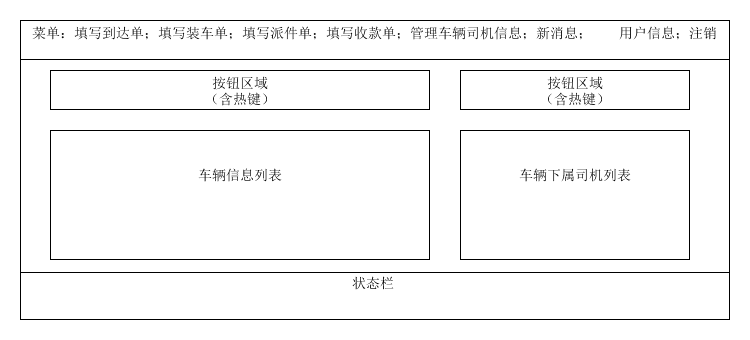
输入新信息：输入想要新增的车辆/司机信息，一个独立组件

输入错误提示：新增信息检查有误提示，一个独立界面

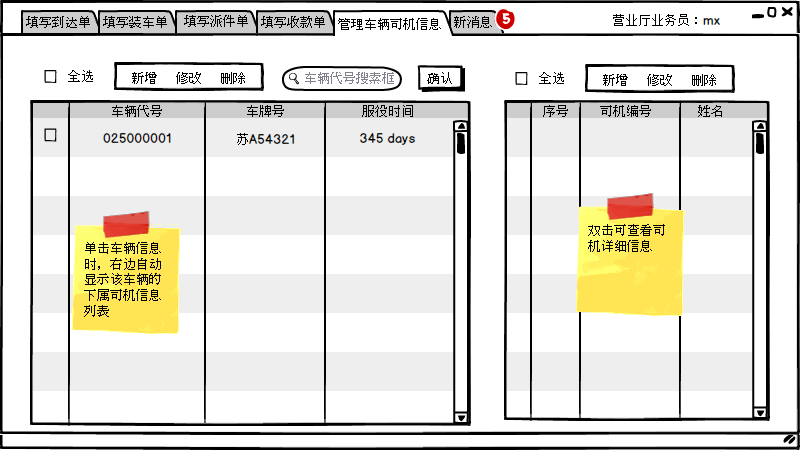
此部分内容结构如下：



* + 1. 界面设计与原型化



营业厅业务员管理车辆司机信息界面设计



营业厅业务员管理车辆司机信息原型化主页面

* 1. **营业厅业务员填写收款单界面**
     1. 导航设计

针对营业厅业务员填写收款单任务，可以设计下列独立界面或组件

输入收款信息：一个独立组件

清空：清空所填信息，一个独立组件

清空失败提示：一个独立界面

保存草稿：一个独立组件

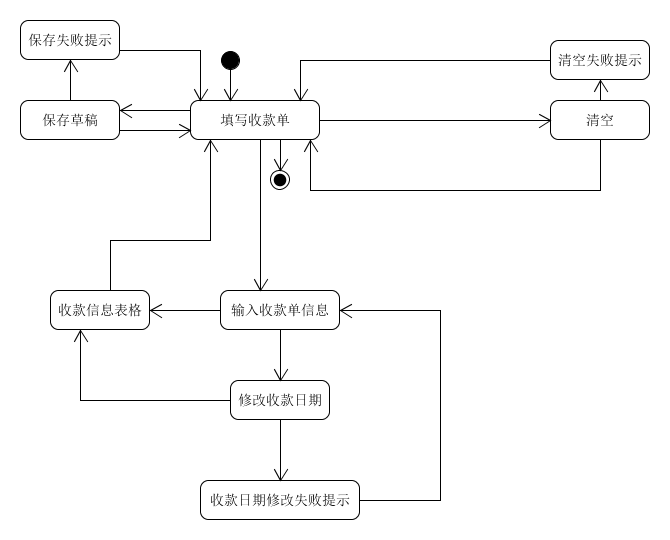
保存失败提示：一个独立界面

修改收款日期：一个独立组件

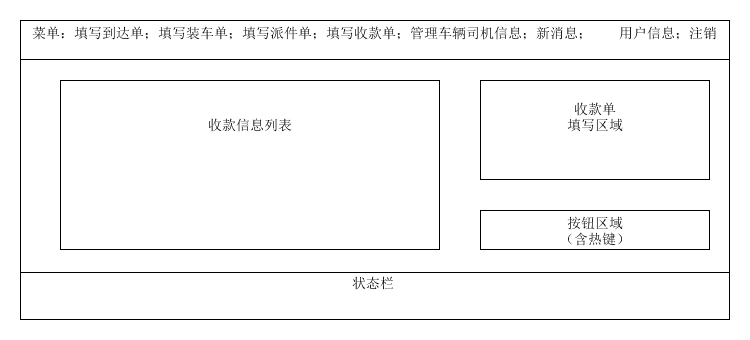
修改失败提示：一个独立界面

收款信息表格：显示已填入的收款信息，一个独立组件

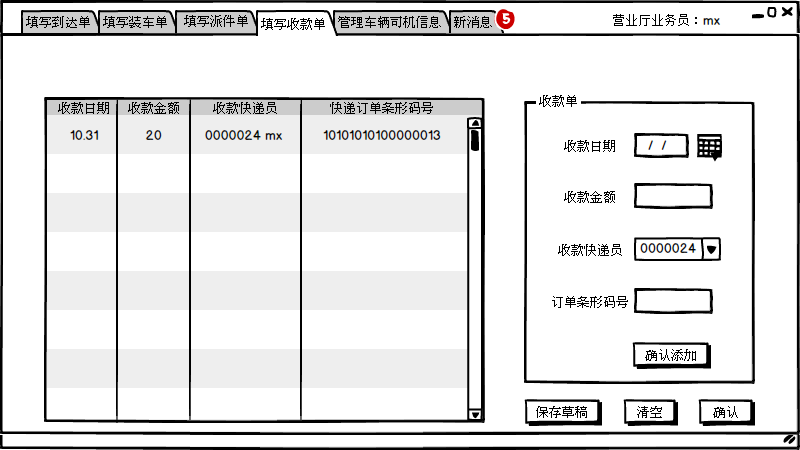
此部分内容结构如下：



* + 1. 界面设计与原型化



营业厅业务员填写收款单界面设计



营业厅业务员填写收款单原型化主页面

* 1. **营业厅业务员填写到达单界面**
     1. 导航设计

针对营业厅业务员填写到达单任务，可以设计下列独立界面或组件

查询快递信息：独输入想要查询的快递单号，一个独立组件

快递单号输入错误提示：独立界面

快递信息显示：显示填写到达单需要的快递信息

清空：清空所填信息，一个独立组件

清空失败提示：一个独立界面

保存草稿：一个独立组件

保存失败提示：一个独立界面

输入到达单信息：一个独立组件

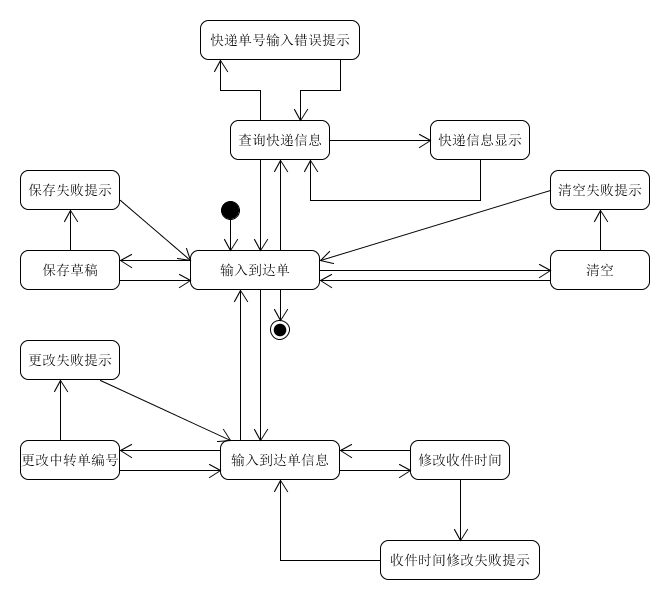
更改中转单编号：一个独立组件

更改失败提示：一个独立界面

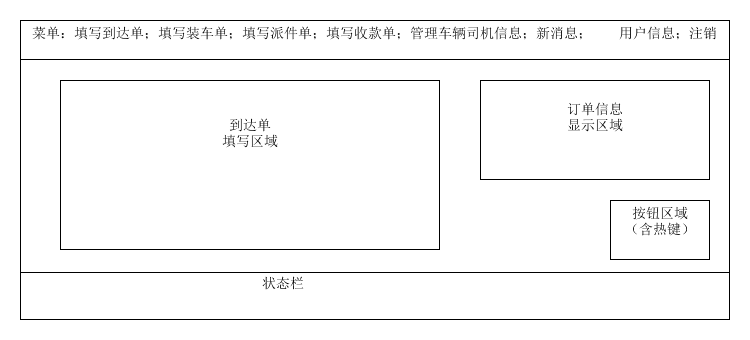
更改到达收件时间：一个独立组件

更改失败提示：一个独立界面

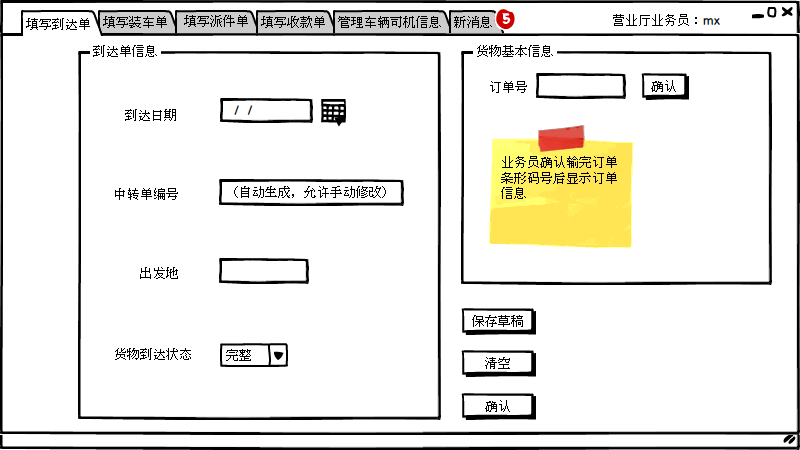
此部分内容结构如下：



* + 1. 界面设计与原型化



营业厅业务员填写到达单界面设计



营业厅业务员填写到达单原型化主页面

* 1. **营业厅业务员填写派件单界面**
     1. 导航设计

针对营业厅业务员填写派件单任务，可以设计下列独立界面或组件

查询快递信息：独输入想要查询的快递单号，一个独立组件

快递单号输入错误提示：一个独立界面

快递信息显示：显示填写派件单需要的快递信息，一个独立组件

清空：清空所填信息，一个独立组件

清空失败提示：一个独立界面

保存草稿：一个独立组件

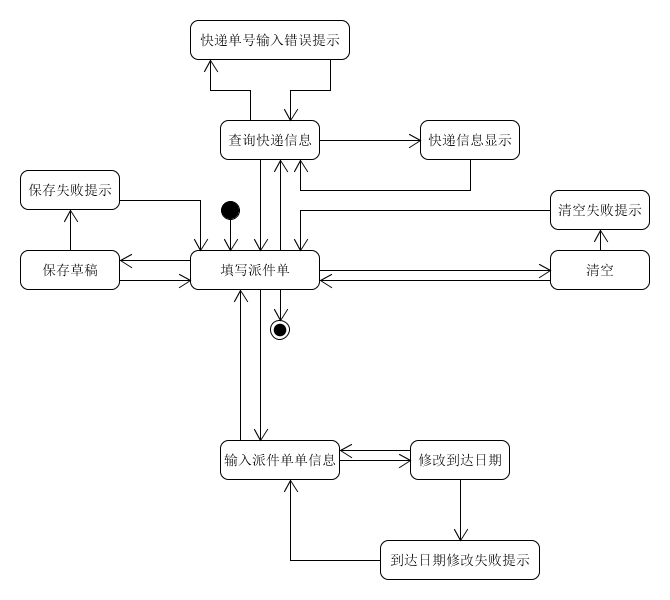
保存失败提示：一个独立界面

输入派件单信息：一个独立组件

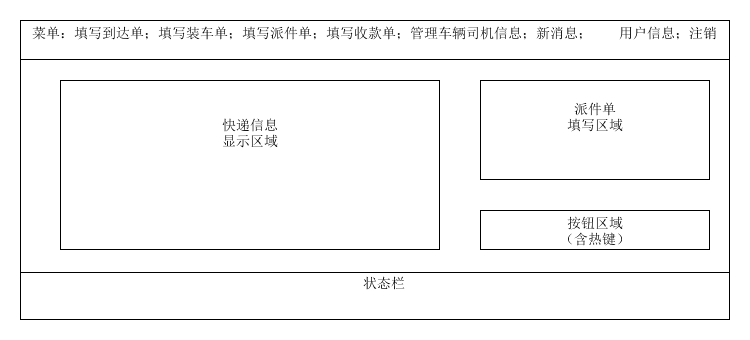
更改到达日期：一个独立组件

更改失败提示：一个独立界面

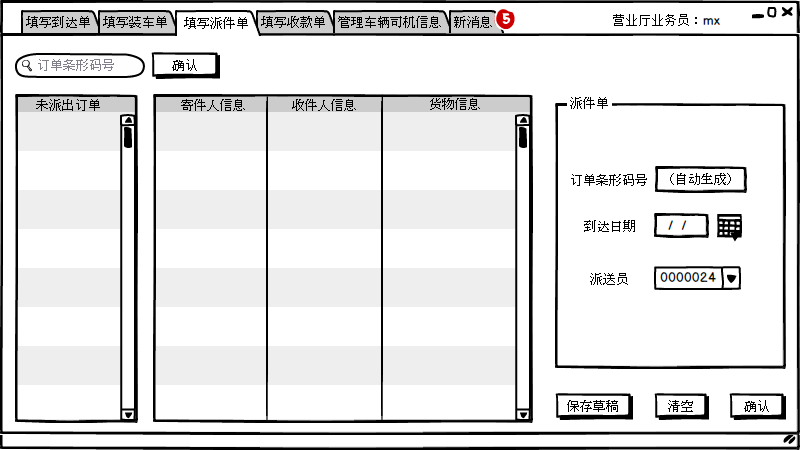
此部分内容结构如下：



* + 1. 界面设计与原型化



营业厅业务员填写派件单界面设计



营业厅业务员填写派件单原型化主页面

* 1. **营业厅业务员填写装车单界面**
     1. 导航设计

针对营业厅业务员填写装车单任务，可以设计下列独立界面或组件

查询快递信息：独输入想要查询的快递单号，一个独立组件

快递单号输入错误提示：一个独立界面

快递信息显示：显示填写到达单需要的快递信息，一个独立组件

清空：清空所填信息，一个独立组件

清空失败提示：一个独立界面

保存草稿：一个独立组件

保存失败提示：一个独立界面

输入到达单信息：一个独立组件

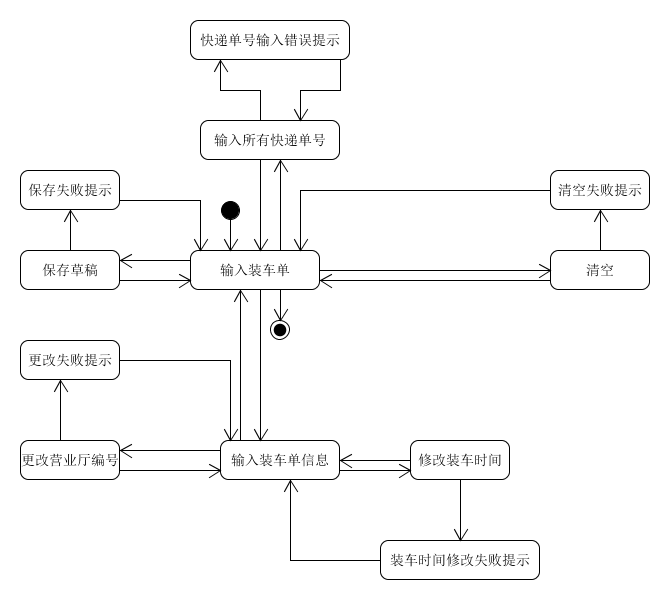
更改营业厅编号：一个独立组件

更改失败提示：一个独立界面

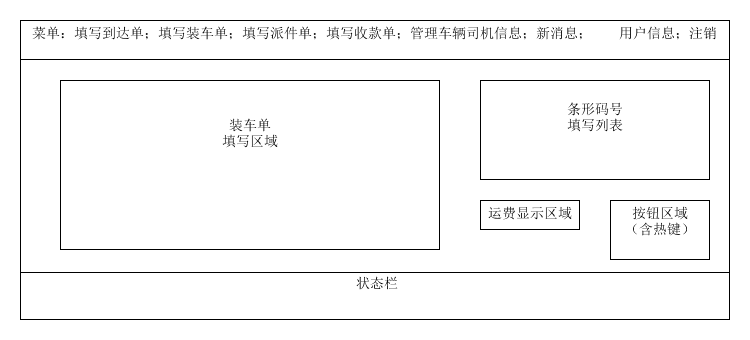
更改装车时间：一个独立组件

更改失败提示：一个独立界面

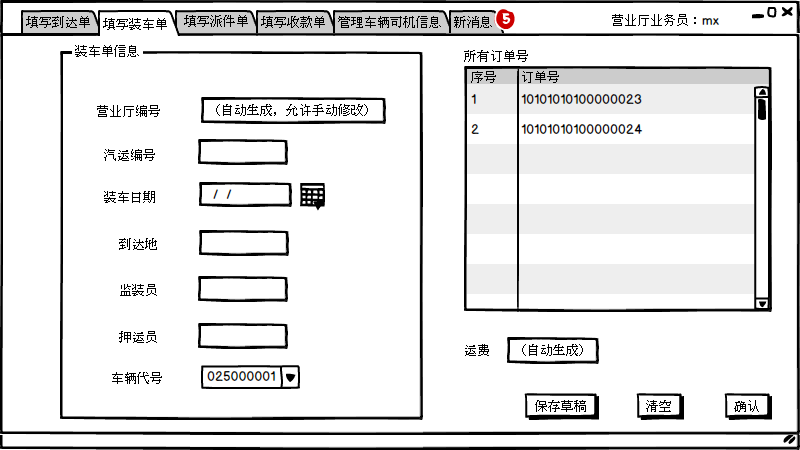
此部分内容结构如下：



* + 1. 界面设计与原型化



营业厅业务员填写装车单界面设计



营业厅业务员填写装车单原型化主页面

## 总经理管理机构界面

* + 1. 导航设计

针对总经理管理机构任务，可以设计下列独立界面或组件

查询机构信息：一个独立按钮组件

修改机构信息：一个独立按钮组件

删除机构：一个独立按钮组件

删除失败提示：一个独立界面

清空：清空所填内容，一个独立按钮组件

清空失败提示：一个独立组件

保存草稿：临时保存现在所填的内容，一个独立按钮组件

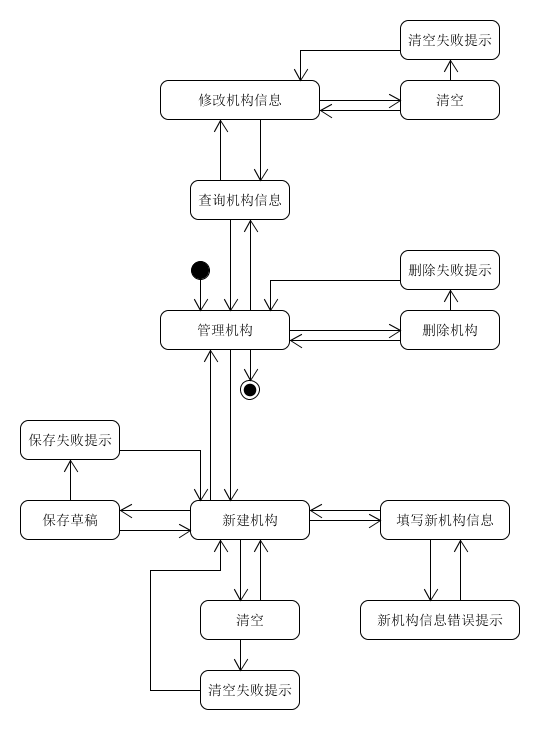
保存失败提示：一个独立界面

新建机构：一个独立按钮组件

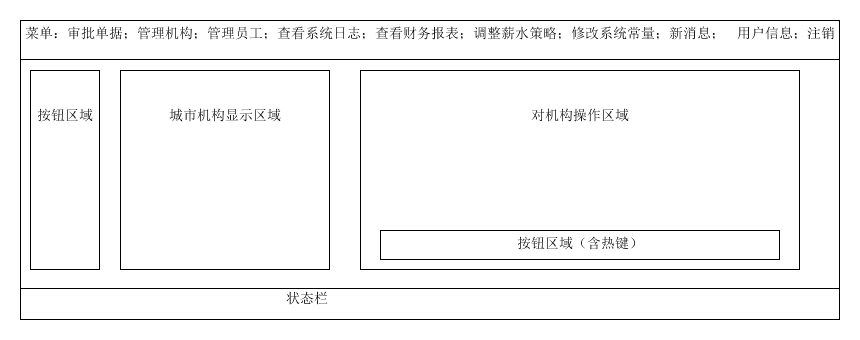
填写新机构信息：一个独立组件

信息错误提示：当填写的新机构信息检测有误时予以提醒，一个独立界面

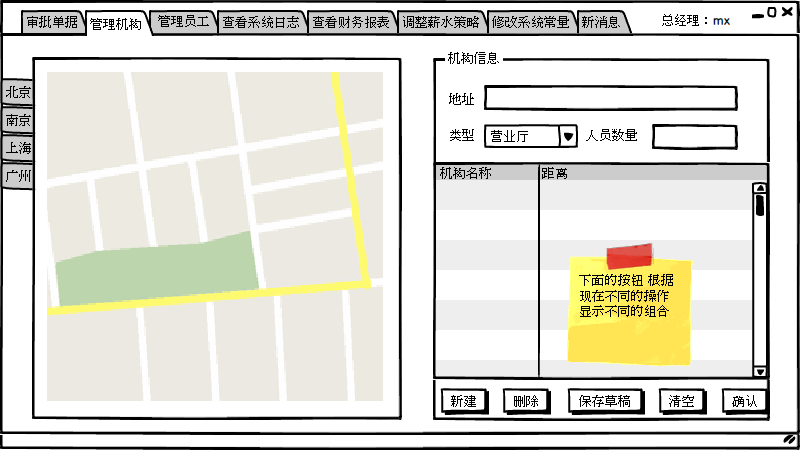
此部分内容结构如下：



* + 1. 界面设计与原型化



总经理管理机构界面设计



总经理管理机构原型化界面

## 总经理管理员工界面

* + 1. 导航设计

针对总经理管理员工任务，可以设计下列独立界面或组件

查询员工信息：一个独立按钮组件

修改员工信息：一个独立按钮组件

删除员工：一个独立按钮组件

删除失败提示：一个独立界面

清空：清空所填内容，一个独立按钮组件

清空失败提示：一个独立组件

保存草稿：临时保存现在所填的内容，一个独立按钮组件

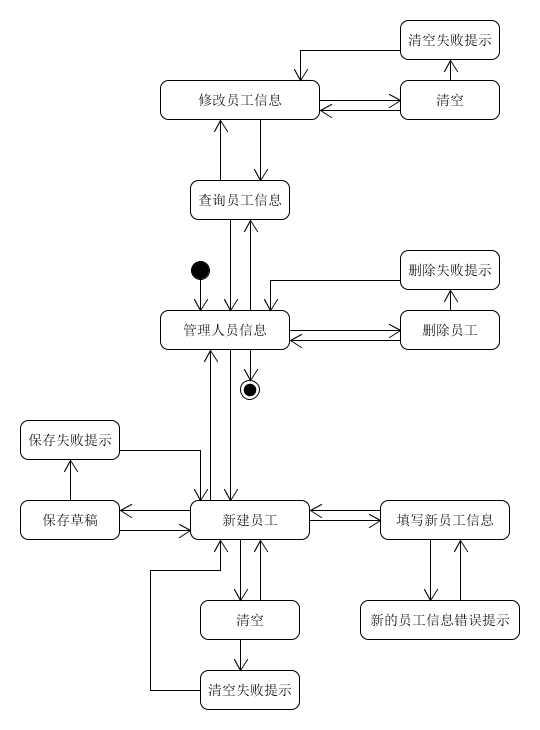
保存失败提示：一个独立界面

新建员工：一个独立按钮组件

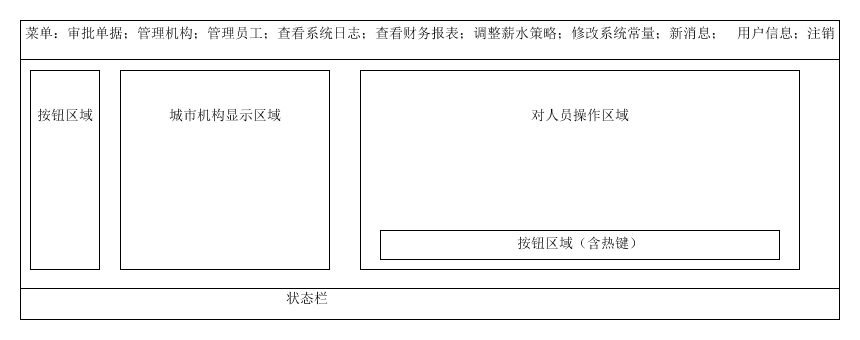
填写新员工信息：一个独立组件

信息错误提示：当填写的新员工信息检测有误时予以提醒，一个独立界面

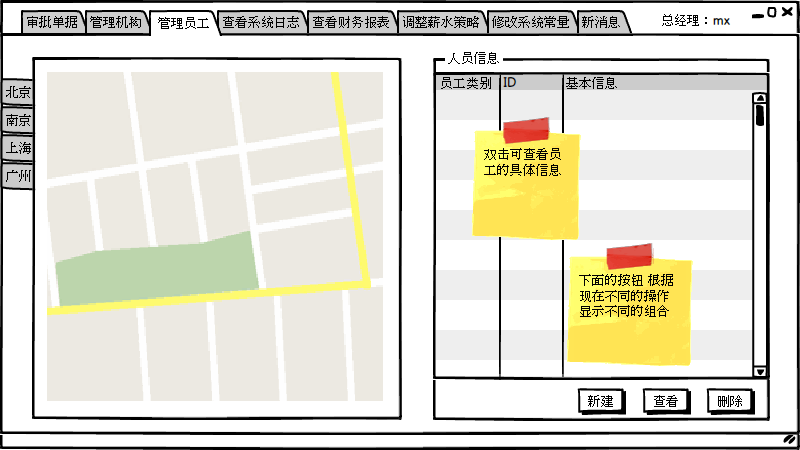
此部分内容结构如下：



* + 1. 界面设计与原型化



总经理管理员工界面设计



总经理管理员工原型化界面

## 总经理审批单据界面

* + 1. 导航设计

针对总经理审批单据任务，可以设计下列独立界面或组件

审批：双击单据表格中的信息，一个独立按钮组件

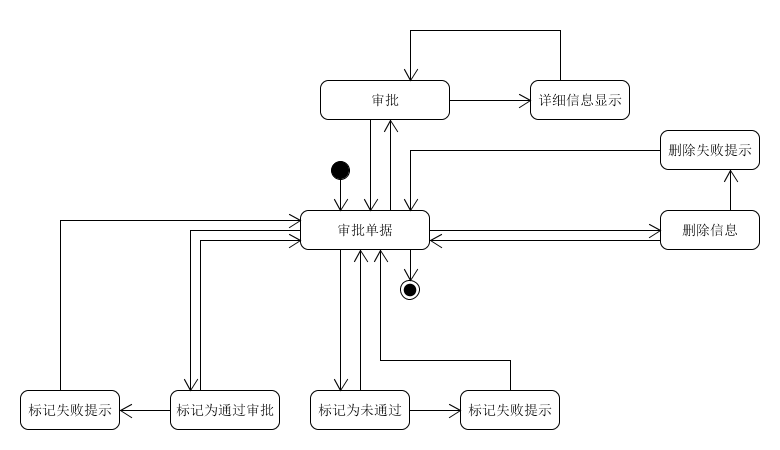
删除信息：删除选中的信息：一个独立按钮组件

删除失败提示：一个独立界面

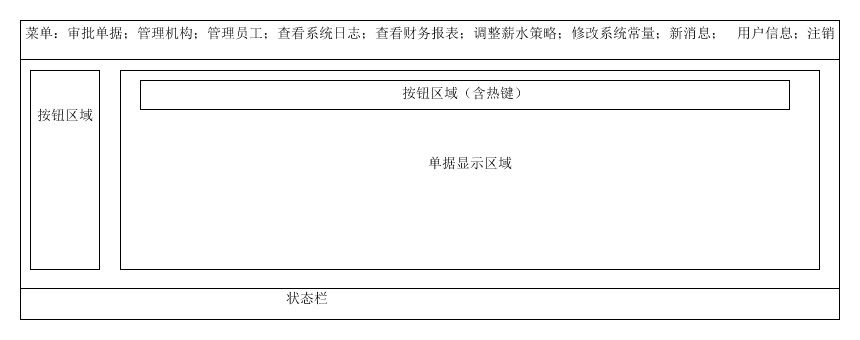
标记为未通过：将选中的单据标记为未通过审批，一个独立按钮组件

标记为已通过：将选中的单据标记为通过审批，一个独立按钮组件

此部分内容结构如下：



* + 1. 界面设计与原型化



总经理审批单据界面设计



总经理审批单据原型化界面

## 总经理调整薪水策略界面

* + 1. 导航设计

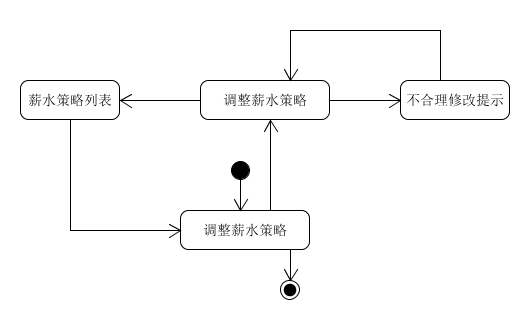
针对总经理调整薪水策略任务，可以设计下列独立界面或组件

调整薪水策略：输入新的薪水策略，一个独立组件

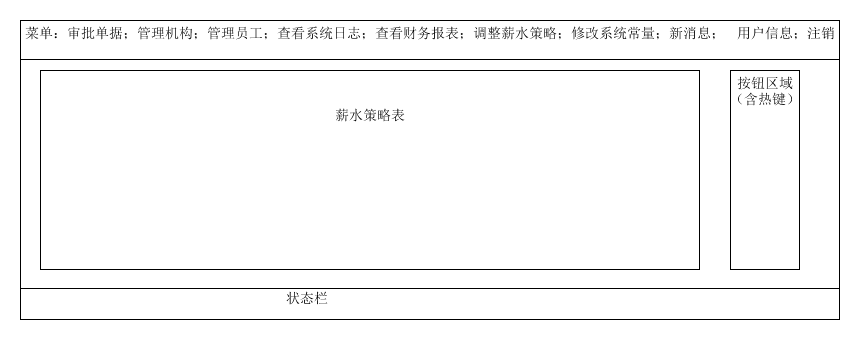
薪水策略列表：调整之后的薪水策略列表，一个独立组件

不合理修改提示：对不合理的薪水策略修改进行提示，一个独立界面

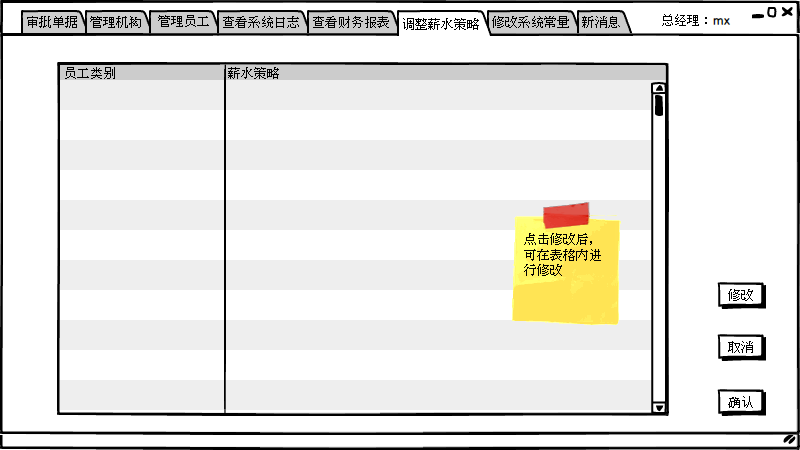
此部分内容结构如下：



* + 1. 界面设计与原型化



总经理调整薪水策略界面设计



总经理调整薪水策略原型化界面

## 总经理修改系统常量界面

* + 1. 导航设计

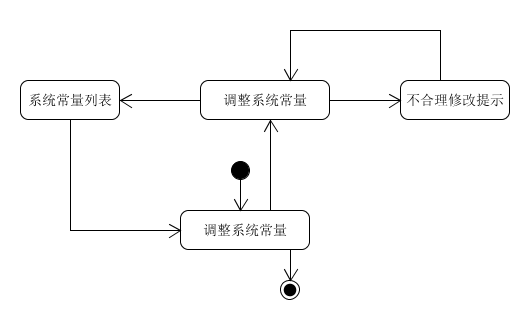
针对总经理修改系统常量任务，可以设计下列独立界面或组件

修改系统常量：输入新的系统常量，一个独立组件

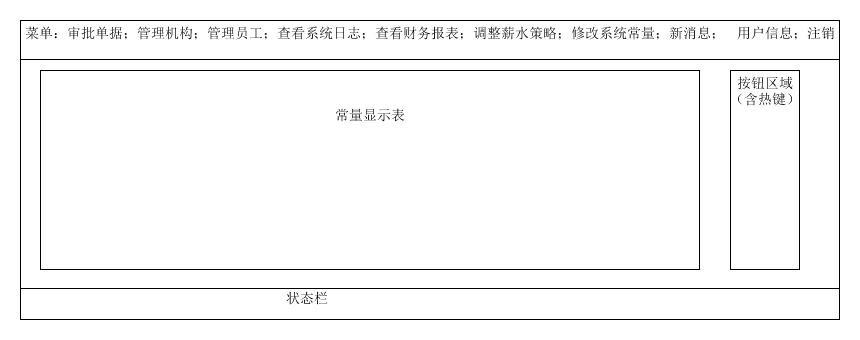
系统常量列表：调整之后的系统常量列表，一个独立组件

不合理修改提示：对不合理的系统常量修改进行提示，一个独立界面

此部分内容结构如下：



* + 1. 界面设计与原型化



总经理修改系统常量界面设计



总经理修改系统常量原型化界面

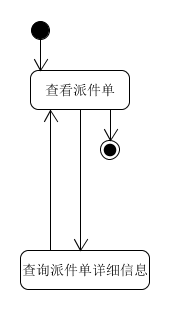
## 快递员查看派件单界面

* + 1. 导航设计

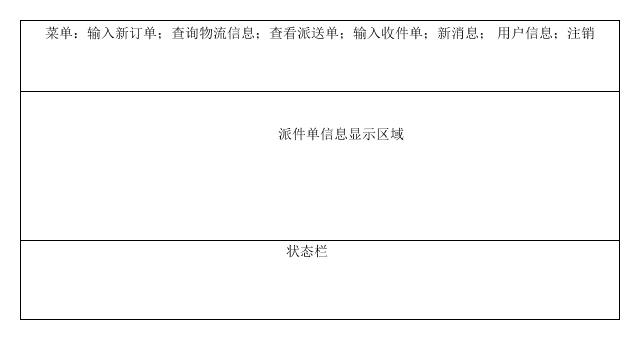
针对快递员查看派件单任务，可以设计下列独立界面或组件

查询派件单详细信息：响应后显示派件单详细信息，独立按钮组件

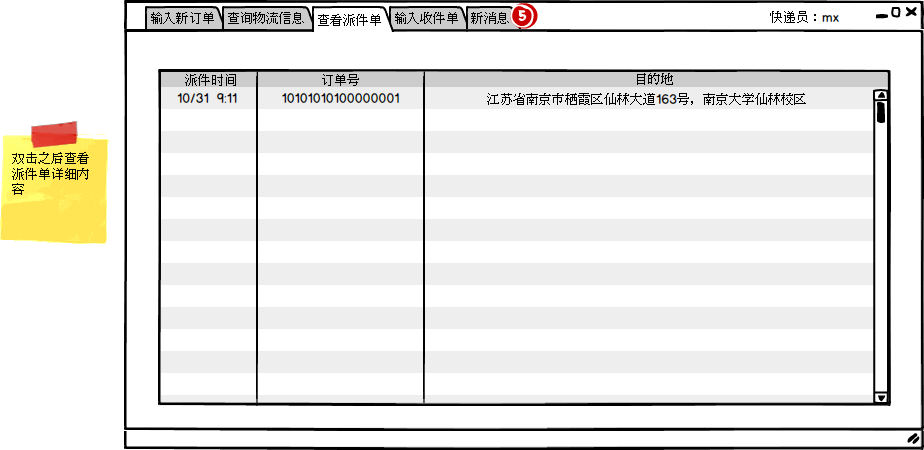
此部分内容结构如下：



* + 1. 界面设计与原型化



快递员查看派件单界面设计



快递员查看派件单原型化界面

## 快递员查询物流信息界面

* + 1. 导航设计

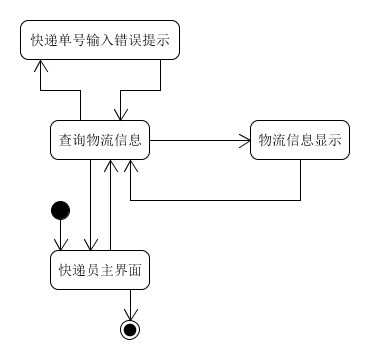
针对库存管理人员选择工作内容任务，可以设计下列独立界面或组件

此部分内容结构如下：

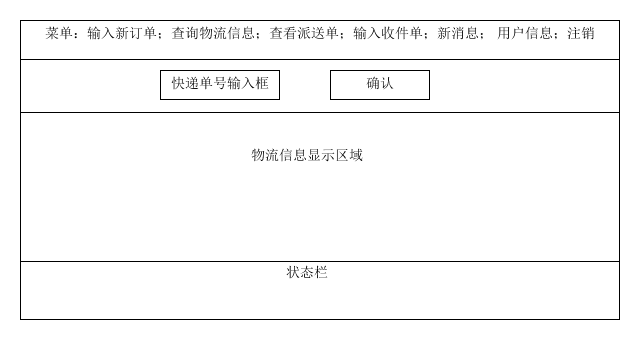
快递单号输入：独立组件

快递单号输入错误提示：独立组件

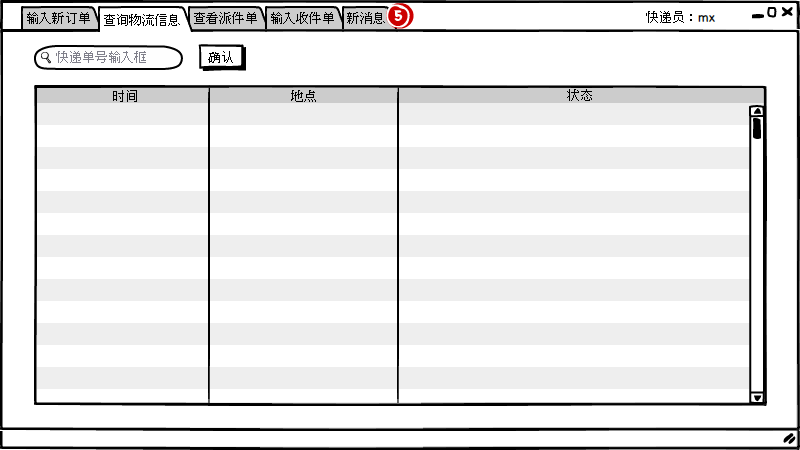
物流信息显示：独立组件



* + 1. 界面设计与原型化



快递员查询物流信息主界面设计



快递员查询物流信息原型化主页面

* 1. **快递员主界面**
     1. 导航设计

针对快递员登陆后查看新信息，可以设计下列独立界面或组件

查询信息详细内容：双击信息，一个独立按钮组件

详细信息显示：查询新消息详细内容的独立界面

删除信息：删除选中的信息，一个独立按钮组件

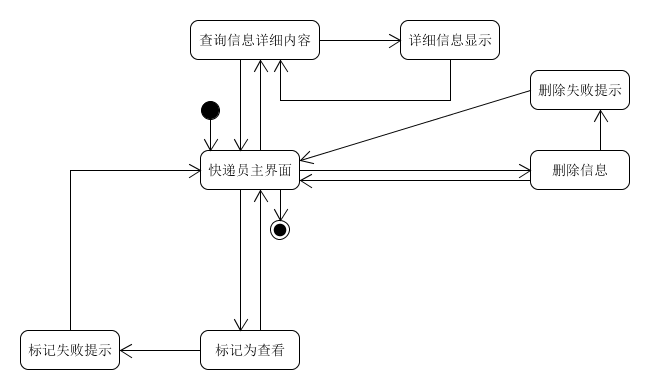
删除失败提示：一个独立组件

标记为已查看：将选中的信息标记为已查看，一个独立按钮组件

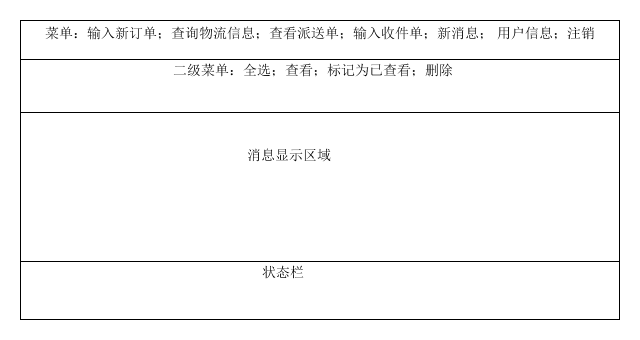
标记失败提示：一个独立组件

\*注：每个用户登陆后的主界面都是自己的消息提醒，内容大致相同，后面的用户主界面就不再做单独说明

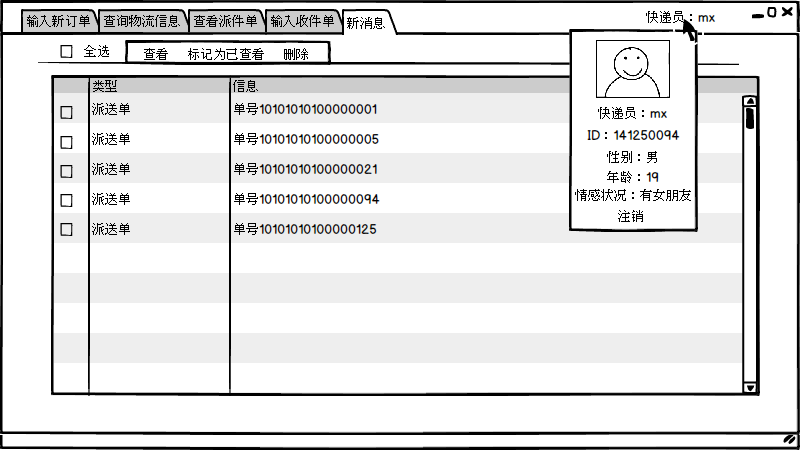
此部分内容结构如下：



* + 1. 界面设计与原型化



快递员主界面界面设计



快递员主界面原型化界面

## 快递员输入收件信息界面

* + 1. 导航设计

针对快递员输入收件信息任务，可以设计下列独立界面或组件

搜索快递单号：独立组件

快递单号输入错误提示：独立界面

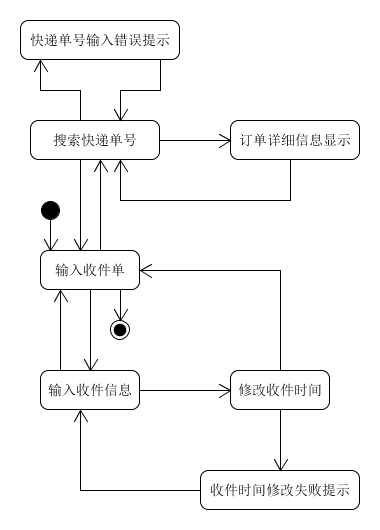
订单详细信息显示：独立组件

输入收件信息：独立组件

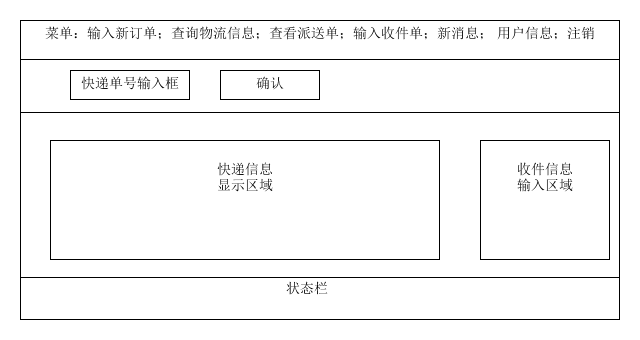
修改收件时间：独立组件，建立在输入收件信息界面上

修改收件时间失败提示：独立界面

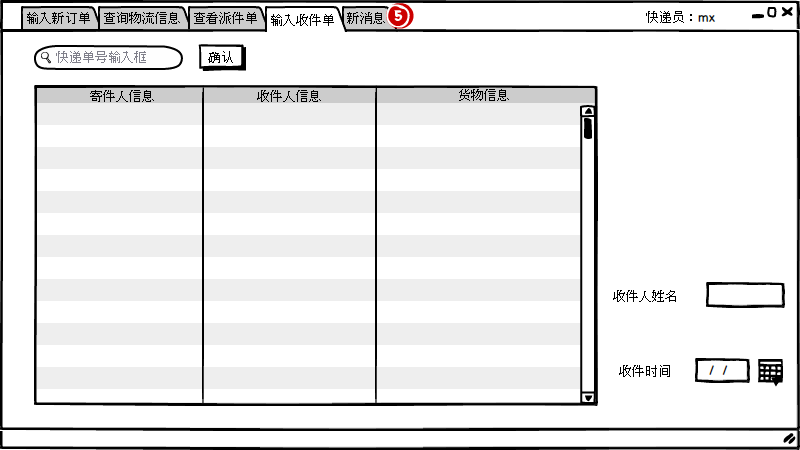
此部分内容结构如下：



* + 1. 界面设计与原型化



快递员输入收件信息界面设计



快递员输入收件信息原型化界面

## 填写新订单界面

* + 1. 导航设计

针对快递员填写新订单的任务，可以设计下列独立界面或组件

此部分内容结构如下：

填写订单信息：一个独立组件

清空：清空所填内容，一个独立按钮组件

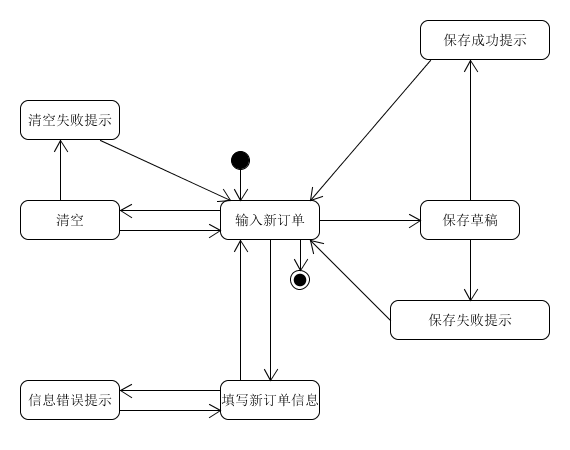
清空失败提示：一个独立组件

保存草稿：临时保存现在所填的内容，一个独立按钮组件

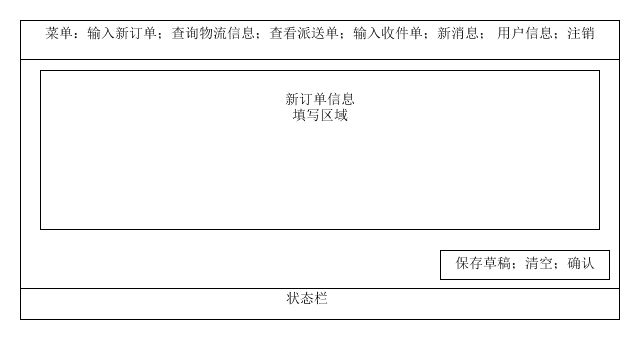
保存成功提示：一个独立组件

保存失败提示：一个独立组件

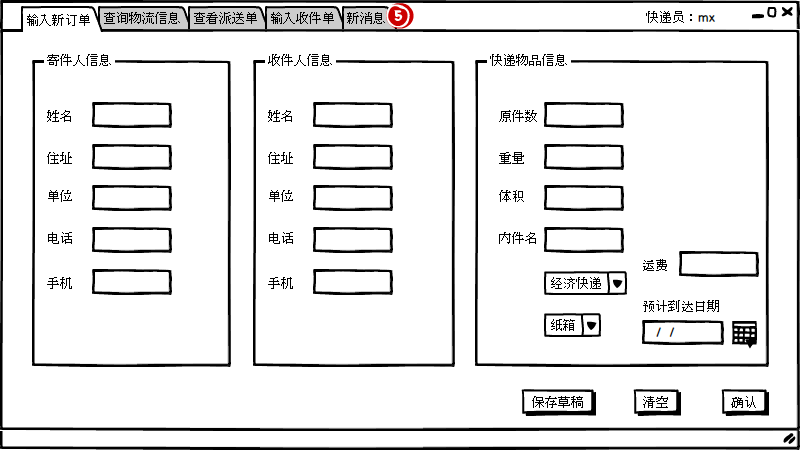
信息错误提示：当填写的新订单内容经检测有误时予以提醒，一个独立组件



* + 1. 界面设计与原型化



快递员输入新订单界面设计



快递员输入新订单原型化主页面

* 1. **新建出库单主界面**
     1. 导航设计

针对库存管理人员新建出库单任务，可以设计下列独立界面或组件

新建出库单：进行新建出库单的主要内容独立界面

出库信息和装运：要填写的内容，并负责提示输入错误，两个独立组件

订单预览：显示输入的订单号对应的订单的信息，一个独立组件

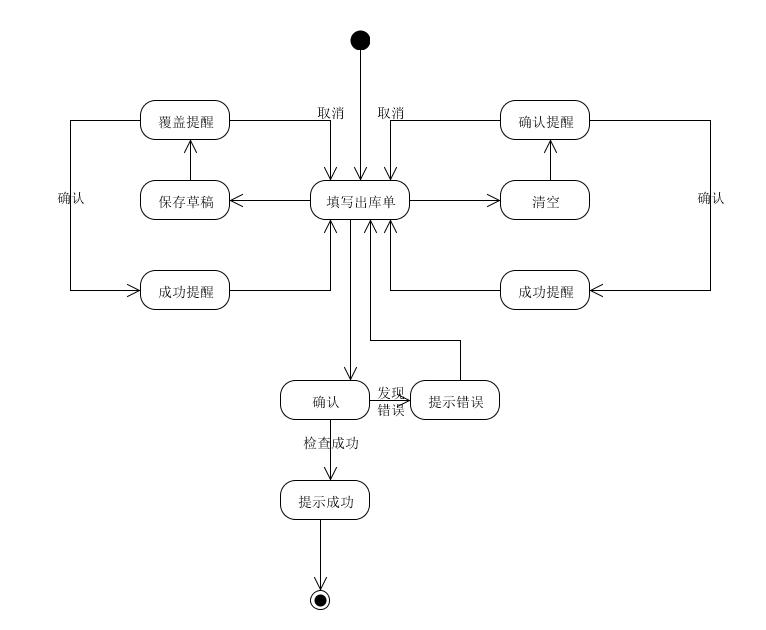
中转单/装车单预览：显示输入的单号对应的单的信息，一个独立组件

保存草稿：发起草稿保存的独立按钮组件

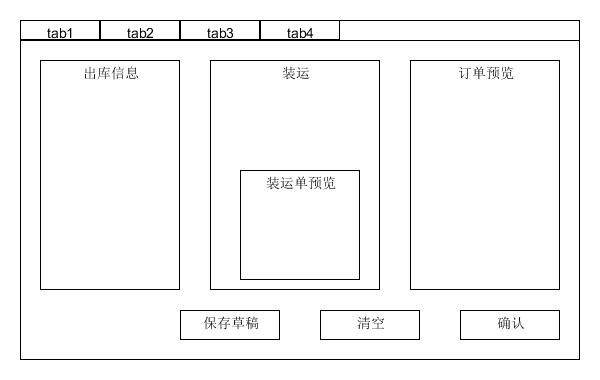
清空：清空所有输入的独立按钮组件

确认：进行入库单提交的独立按钮组件

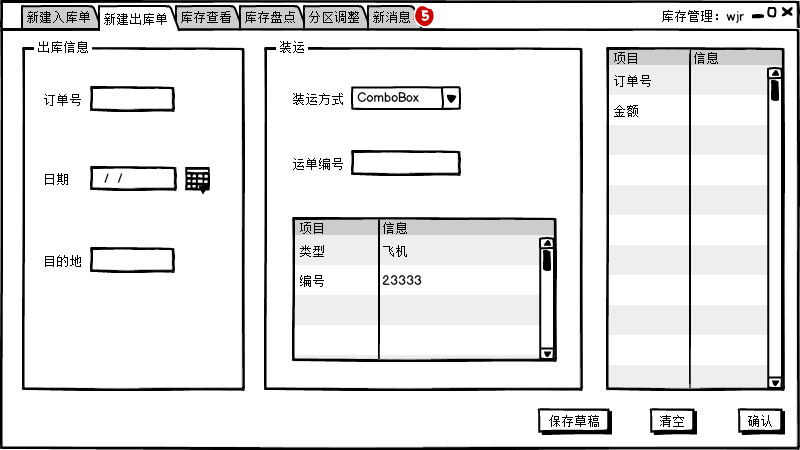
此部分内容结构如下：



* + 1. 界面设计与原型化



新建出库单界面设计



新建出库单原型化主页面

* 1. **查看出入库情况主界面**
     1. 导航设计

针对库存管理人员查看出入库情况任务，可以设计下列独立界面或组件

出入库查看：库存管理人员查看出入库情况的独立界面

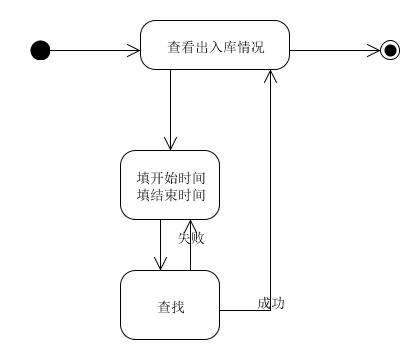
起始时间选择：选择查看出入库情况的起始时间，两个独立组件

查看：刷新出入库情况的显示，一个独立按钮组件

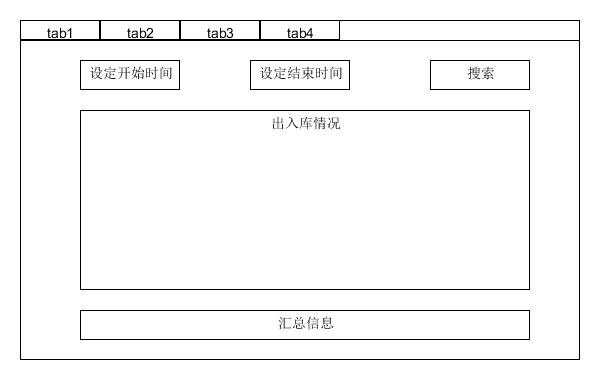
显示表格：显示出入库情况的信息，一个独立组件

汇总：显示入库情况的信息的汇总，一个独立组件

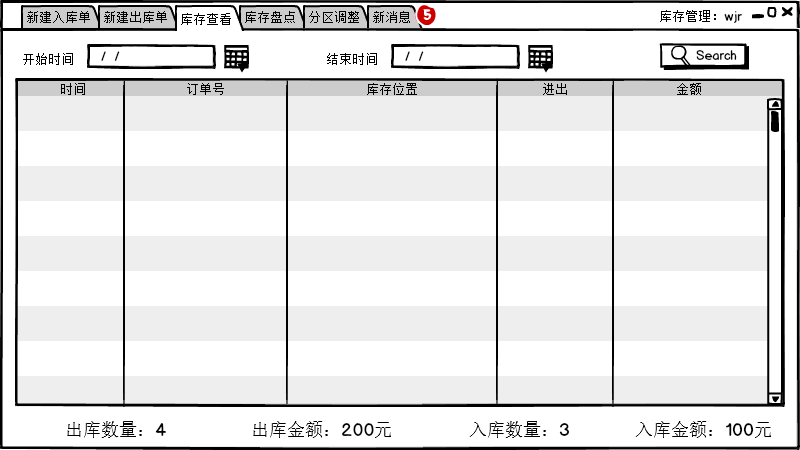
此部分内容结构如下：



* + 1. 界面设计与原型化



查看出入库情况界面设计



查看出入库情况原型化主页面

* 1. **分区调整主界面**
     1. 导航设计

针对库存管理人员调整分区和警戒比例任务，可以设计下列独立界面或组件

分区调整界面：库存管理人员进行分区和警戒比例调整的独立界面

分区选择：选择仓库中的分区，一个独立组件

排信息概览：显示选中分区的各排的信息，一个独立组件

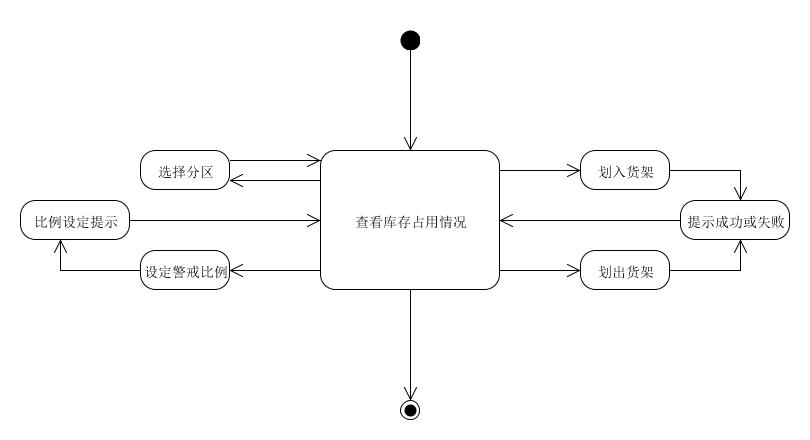
分区信息概览：显示选中分区的信息，一个独立组件

划出货架：发起货架划出请求，一个独立按钮组件

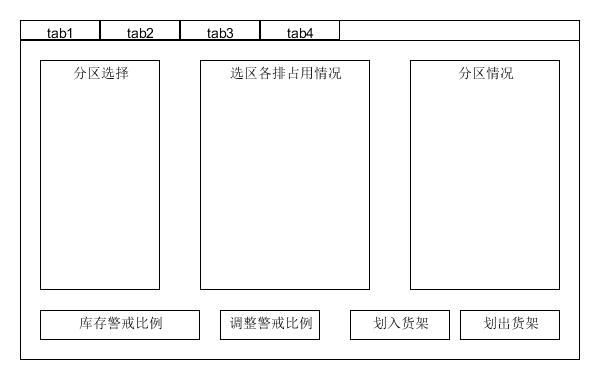
划入货架：发起货架划入请求，一个独立按钮组件

警戒比例：显示警戒比例，可以重新键入后调整，一个独立组件

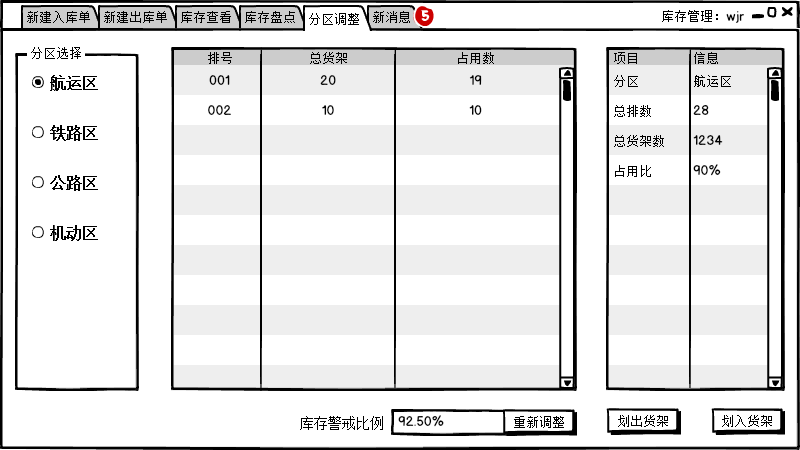
此部分内容结构如下：



* + 1. 界面设计与原型化



调整分区界面设计



分区调整原型化主页面

* 1. **库存盘点主界面**
     1. 导航设计

针对库存管理人员库存盘点任务，可以设计下列独立界面或组件

库存盘点界面：库存管理人员进行库存盘点的独立界面

位置选择：选择仓库位置的类文件查看器的独立组件

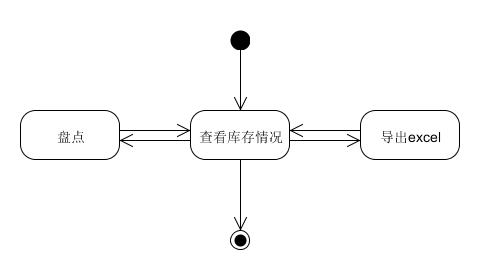
信息概览：显示选中位置的信息，一个独立按钮组件

重新盘点：刷新仓库信息，一个独立按钮组件

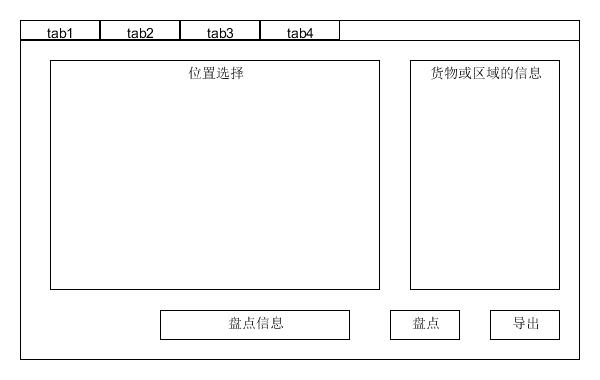
导出excel：导出excel文件，一个独立按钮组件

盘点信息：显示上次盘点的日期和序号，一个独立组件

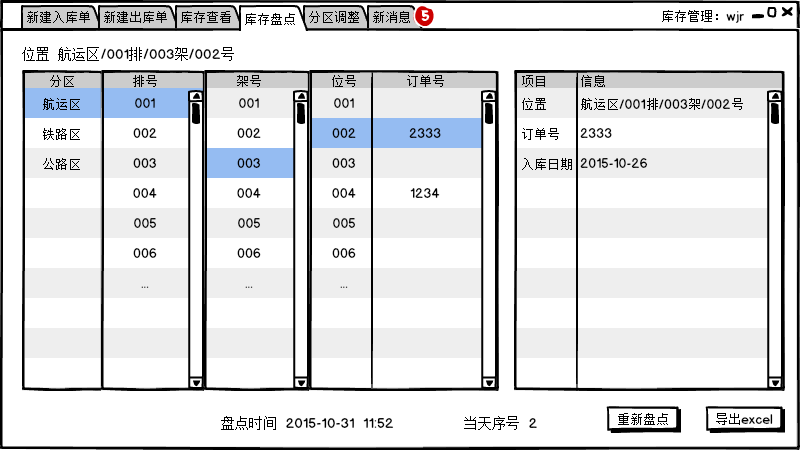
此部分内容结构如下：



* + 1. 界面设计与原型化



盘点库存界面设计



盘点库存原型化主页面

* 1. **新建入库单主界面**
     1. 导航设计

针对库存管理人员新建入库单任务，可以设计下列独立界面或组件

新建入库单：进行新建入库单的主要内容独立界面

入库信息和库存位置：要填写的内容，并负责提示输入错误，两个独立组件

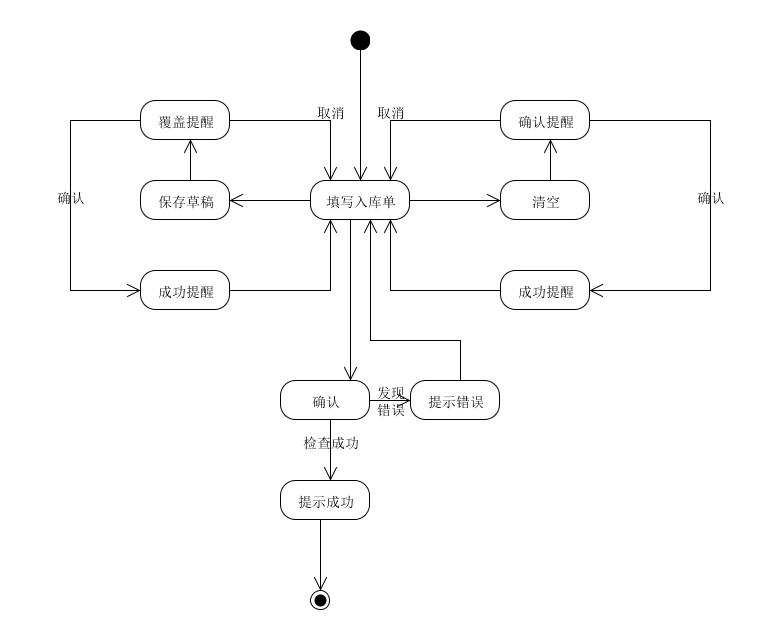
订单预览：显示输入的订单号对应的订单的信息，一个独立组件

保存草稿：发起草稿保存的独立按钮组件

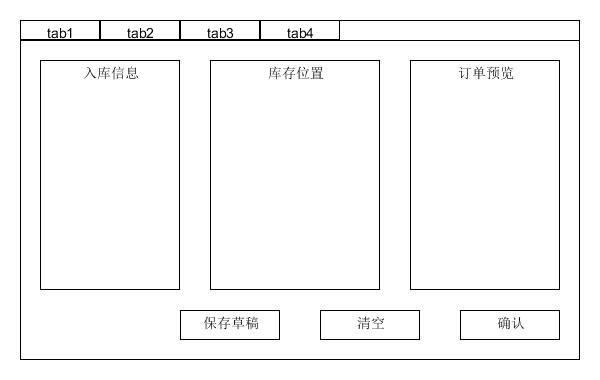
清空：清空所有输入的独立按钮组件

确认：进行入库单提交的独立按钮组件

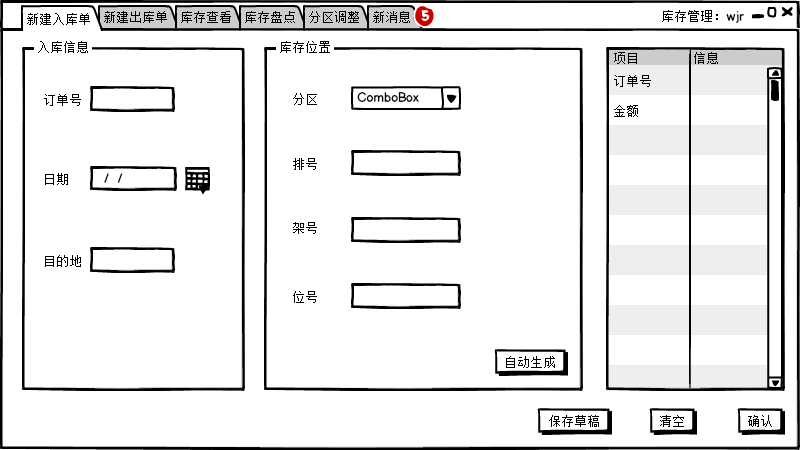
此部分内容结构如下：



* + 1. 界面设计与原型化



新建入库单界面设计



新建入库单原型化主页面