



山东大学（威海）

物流仓储管理系统结构化设计

年级： 2019级

专业： 计算机科学与技术（中澳合作）

姓名： 朱胤璘 黄皓 赵玉淋 樊华

指导老师： 王文玉

编写日期：2021 年 04 月 23 日

目录

一、 引言	2
二、 体系结构设计	3
2.1 改善后的数据流图	3
2.2 系统功能模块划分	8
2.3 系统架构图	15
2.4 系统结构图	16
三、 接口设计	21
3.1 人机交互接口设计	21
3.2 模块间接口设计	25
四、 数据设计	28
4.1 调整后的部分实体图	28
4.2 关系模式的建立	30
4.3 数据库中表的实现	32
4.4 建表语句 (Mysql)	41
五、 过程设计	49
5.1 主要功能模块的程序流程图	49
5.2 重要算法的伪代码表述	55

一、引言

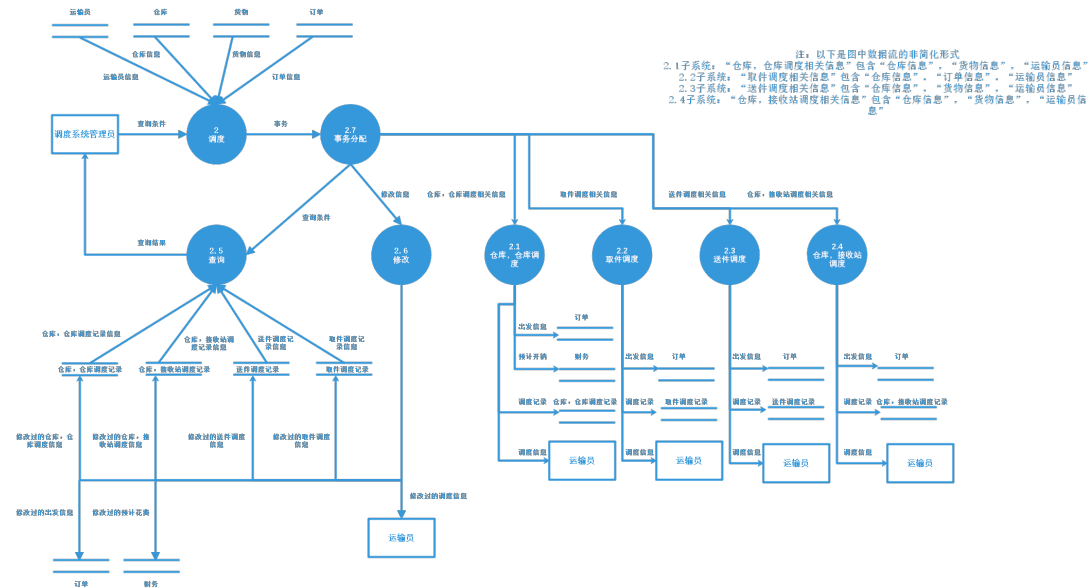
在软件生存期中，软件设计处于需求分析阶段及软件构造（编码）阶段的中间位置。需求分析的主要任务是明确“做什么”，在完成了需求分析之后，就进入了软件设计阶段，它的任务是回答“怎么做”。

结构化设计方法是在模块化、自顶向下逐步细化及结构化程序设计技术基础之上发展起来的。结构化设计方法可以分为两类：一类是根据系统的数据流进行设计，称为面向数据流的设计，或称过程驱动的设计；另一类是根据系统的数据结构进行设计，称为面向数据结构的设计，或称数据驱动的设计。

面向数据流的设计工作与软件需求分析阶段的结构化分析方法相衔接，可以很方便地将用数据流图表示的信息转换成程序结构的设计描述，这一方法还能和编码阶段的“结构化程序设计方法”相适应，成为常用的结构化设计方法。

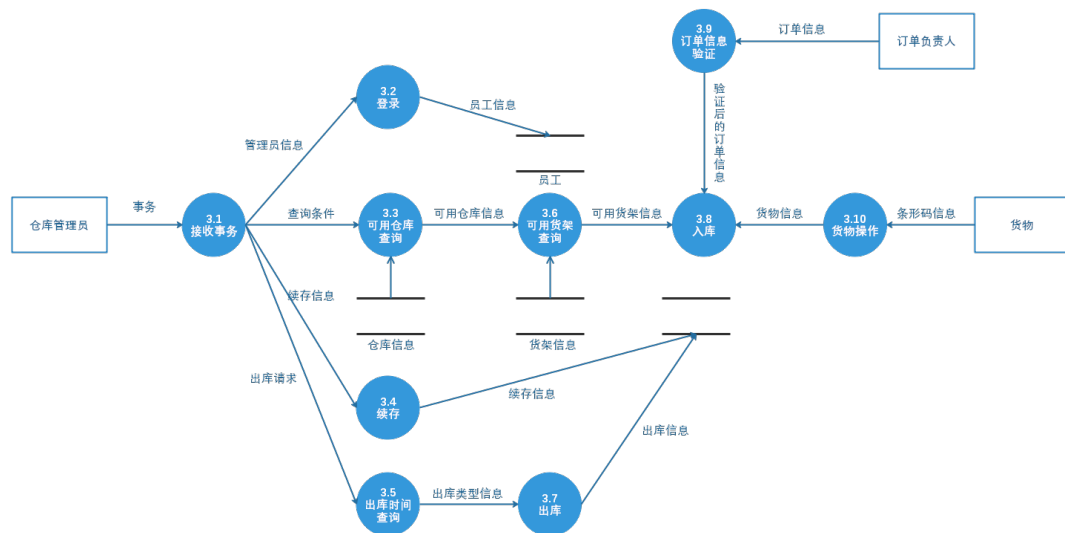
订单管理子系统数据流图经过了一定的改动后，被定义为事务型数据流图。在其数据流图中，共分为发起订单、查询订单、取消订单、编辑收寄人信息、审核订单、查询审核结果事务。

(2) 调度管理子系统



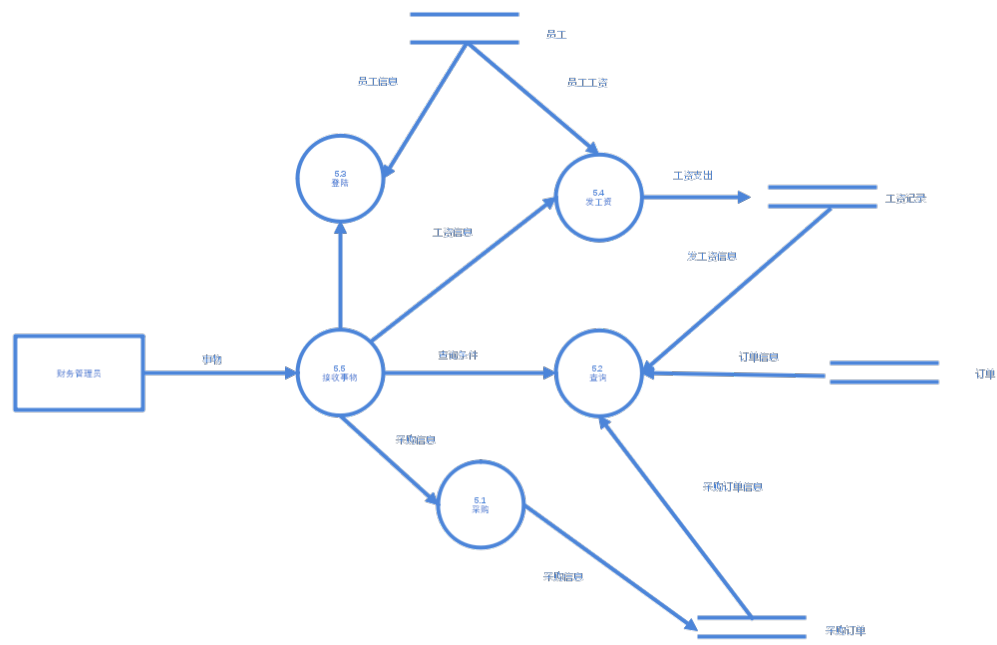
调度子系统数据流图经过了一定的改动后，被定义为事务型数据流图。在其数据流图中，共分为仓库，仓库调度事务，仓库，接收站调度事务，送件调度事务，取件调度事务，管理员查询事务，管理员修改事务。

(3) 仓库管理子系统



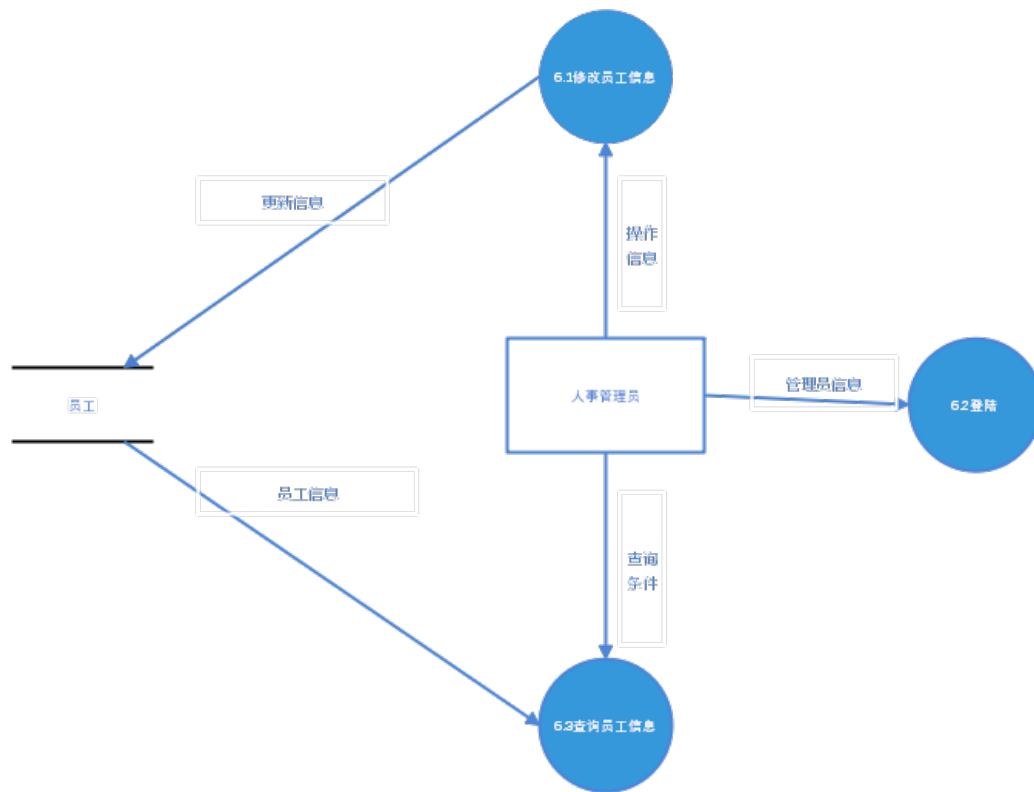
仓库管理子系统经过了一定的改动后，被定义成事务型数据流图。在其数据流图中，共分为登录、可用仓库查询、续存、出库时间查询、可用货架查询、出库、入库、订单信息验证、货物操作事务。

(4) 财务管理子系统



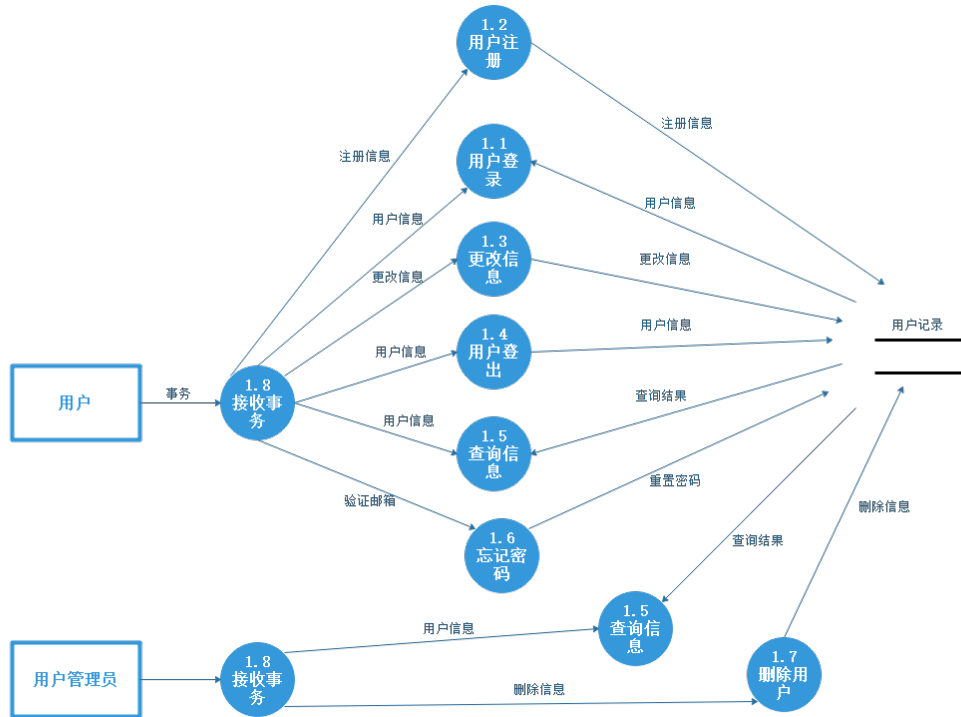
财务管理子系统数据流图经过了一定的改动后，被定义为事务型数据流。在其数据流图中，共分为登陆事务、发工资事务、查询事务和采购事务，订单结算事务

(5) 人事管理子系统



人事管理子系统数据流图经过了一定的改动后，被定义为事务型数据流。在其数据流图中，共分为登陆事务、修改员工表事务 和查询员工信息事务。

(6) 新增子系统：用户管理子系统



用户管理子系统被定义为事务性数据流。在其数据流图中，包括用户注册、用户登录、更改信息、用户登出、查询信息、忘记密码、删除用户事务。

2.2 系统功能模块划分

经过需求分析，我团队最终确定了每个模块所具备的功能。整个项目含有六个子系统，每个子系统功能如下：

(1) 订单管理子系统

①常用收寄人管理

用户在下订单之前需要指定订单的寄件人信息和收件人信息，系统会要求。用户从常用收寄人里选择一项作为寄件人，一项作为收件人。用户可以向常用收寄人信息簿中添加、修改、删除收寄人信息。每一个收寄人信息包括了收寄编号、用户名、收寄人姓名、收寄人电话、收寄人地址。一个收寄可以在不同的订单中被设置为寄件人或收件人

②发起订单

用户登录后，可以发起订单。发起订单需要提供用户的基本信息和订单的基本信息。订单的基本信息包括物品种类、寄件人信息、收件人信息。寄件人信息和收件人信息从常用收寄人信息簿中选择，随后，系统将计算金额并等待用户支付。用户支付成功后，系统自动生成待审核订单。

③审核订单

审核员可以从所有待审核订单中选择货物进行审核。审核员应该检查订单信息与货物一致。如果检查无误，则审核完成，订单状态变更为“生效”。如果审核不通过，则订单状态变更为“终止”。无论检查是否通过，该审核都会被记录到审核结果表中

④取消订单

用户可以对当前正在进行的订单进行撤销操作。对于待审核的订单，撤销后可以全额退费，用户应当自行取回物品。对于生效的订单，如果未运输，取消后可以退掉扣除审核费的剩余费用。如果已经在运输中，则无法取消此订单。

⑤查询订单

用户可以对自己当前正在进行的订单及历史订单的信息进行查询。系统首先列出该用户所有订单的基本信息，用户可以选择其中某一个订单查看详细信息。用户选择一个订单，则向系统发送该订单的订单号，系统会从订单表中查询此订单。管理员可以查询任意用户的所有订单信息。

⑥管理员删除订单

管理员可以从订单表中删除某条订单记录。管理员查询审核结果管理员可以输入订单号，查询相应订单的审核结果。如果订单无审核结果，则返回提示信息。

(2) 调度管理子系统

①运输员取件调度：

用户下的订单经过审核后会发送到取件调度系统。在取件调度系统中每个用户都会在某一个接收站的服务区域之内。取件调度系统会根据用户位置找到能为其提供取件服务的接收站。取件调度系统接着能够派出该接收站的某一名空闲快递员前往接收快递，由于受到地图服务商

接口限制，调度系统只能够每次安排一名快递员取最多 18 个物品(最多提供 18 个途径点的导航服务)

②运输员仓库，接收站调度：

可以在每天的早晚 9 点安排运输员往返仓库和接收站进行货物运输。每个大型存储仓库会有一个接收覆盖面积，该区域内有若干接收站类型的小仓库。调度系统可以指定运输员往返于某些仓库和接收站以及运送哪些货物。

③运输员仓库，仓库调度：

在每一个省份中都会有若干集散地类型的仓库，职能是负责把运输到该省的货物重新分配，运往下面地市中的大型存储仓库。调度系统可以在每天早晚 10 点安排运输员发车，把仓库货物运往目标集散地。调度系统可以按照货物的在仓库的滞留时间安排合理的出仓顺序，尽可能地使货物以最快速度运往用户手中。

④运输员送件调度：

调度系统可以安排快递员把快件送往指定的用户位置，同样是因为地图服务商接口的限制，调度系统可以安排每个快递员最多同时派送前往 18 个不同的地点。

⑤调度记录查询：

调度系统管理员可以查询相关调度记录

⑥调度记录修改：

调度系统管理员可以修改相关调度记录

(3) 仓库管理子系统

①仓库管理员查看当前仓库、货架信息：

仓库管理员通过货物操作扫描三段码获取到当前货物的基本信息及相关的存储要求，验证订单负责人所给出的订单信息后，查看当前空闲的、满足存储要求并且最为合适的仓库、货架，并为货物办理入库。

②仓库管理员办理入库：

验证订单消息后如果未虚假订单则将拒绝货物进入仓库，验证成功后根据订单中货物信息及数量选择合适的仓库、货架，指定对应的仓库编号、货架编号等，用户缴纳一定的存储定金及费用，并且更新数据库中的内容。

③货物出库：

若货物已经逾期，发出警告并向用户提出询问是否退回货物，如果用户选择退回计算退回费用与逾期存放费用，如果用户拒绝退回，则执行货物回收操作，只计算逾期存放费用并且不退回定金，然后计算总存储费用，记录出库时间，更新数据库内容；若货物未逾期，则根据入库时间计算存储费用，并且退回定金后计算总应收金额，记录出库时间，更新数据库中内容。

④货物续存：

若货物存储时间未满足一定期限执行续存，则拒绝执行；若货物已逾期或者将要逾期，则需要明确存储天数，若在指定天数内没有订单提示且货架、仓库可用，则为货物办理续存，并且更新数据库中的内容。

(4) 财务管理子系统

① 工资发放：

财务管理员点击相应按钮，为部分/全部员工发放工资。该功能仅限在每个月 1 日和 2 日两天使用，其余时间财务管理员无发放工资权限。发放工资以后，自动生成账务记录并存入数据库。

②采购信息导入：

采购人员将采购单输入到系统中，系统自动生成账务记录并 存入数据库。

③物流订单和退款订单导入：

物流订单和退款订单导入

对于这两种订单，均可在结算时自动生成一条财务记录并且 保存在数据库。

④账目明细查询：

财务管理员可以根据相应的查询条件查询账目明细并生成 报表。

(5) 人事管理子系统

①员工信息管理：

人力资源人员可以在系统中拥有对员工表的增删改的权限， 可以对其他员工的信息进行统一的管理， 通过这个操作实现人事 调动。

②员工信息查询：

人力资源人员可以在查询系统中通过选择查询条件对我们的 员工信息表进行检索， 筛选出符合条件的员工。

(6) 用户管理子系统

①用户注册

用户在初次使用本系统， 应该首先进行注册。注册包括两类， 一种是注册为普通用户， 另一种为员工用户。注册员工用户需要提供公式分派的员工身份识别码。注册时， 需要提供用户的基本信息， 包括用户名， 密码， 验证邮箱。随后， 系统会向验证邮箱发送一条验证码， 在用户输入正确的验证码后， 系统自动添加新的用户记录。每一个邮箱仅能注册一次。

②修改密码

用户可以修改登录密码。首先输入用户名和旧密码， 在验证成功后， 需要输入新的密码并重复确认。成功修改后， 如果该用户处于登录状态， 系统会强制用户登出。

③忘记密码

用户如果忘记密码，可以通过验证电子邮箱进行密码找回。系统会自动向用户发送一串 16 位验证码，用户验证成功后，可以修改密码。

④用户登录

已注册的用户输入正确的用户名和密码后，系统自动建立与该用户的会话。随后，系统根据用户的类型分配相应的权限，用户可以进行一系列特定的操作。

⑤用户登出

已登录的用户可以登出，系统将自动中断与该用户的会话。

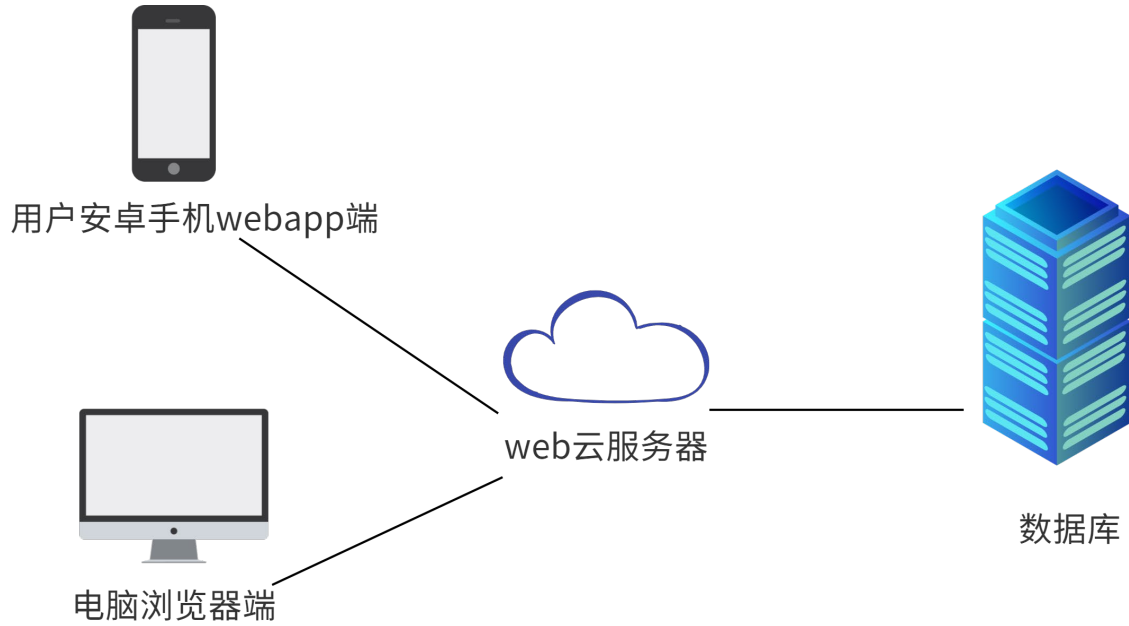
⑥用户修改信息

已登录的用户可以修改个人基本信息。

⑦用户管理员查看、删除用户

用户管理员向系统输入用户名，可以查看特定的用户信息、删除特定的用户信息。

2.3 系统架构图

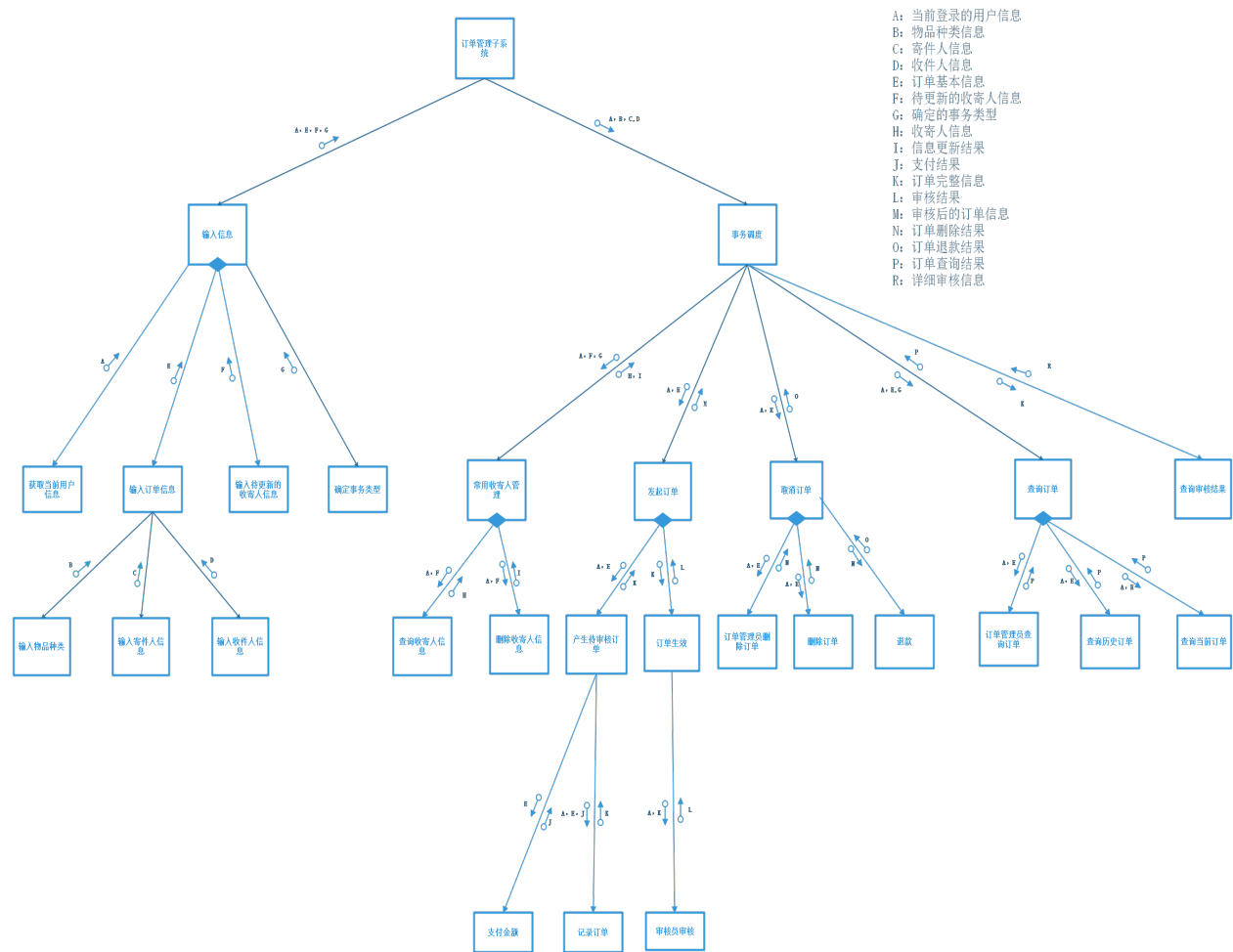


本系统采用 HTML5+JavaScript+Css+Java+MySQL 进行开发。MySQL 是云端数据库的引擎。

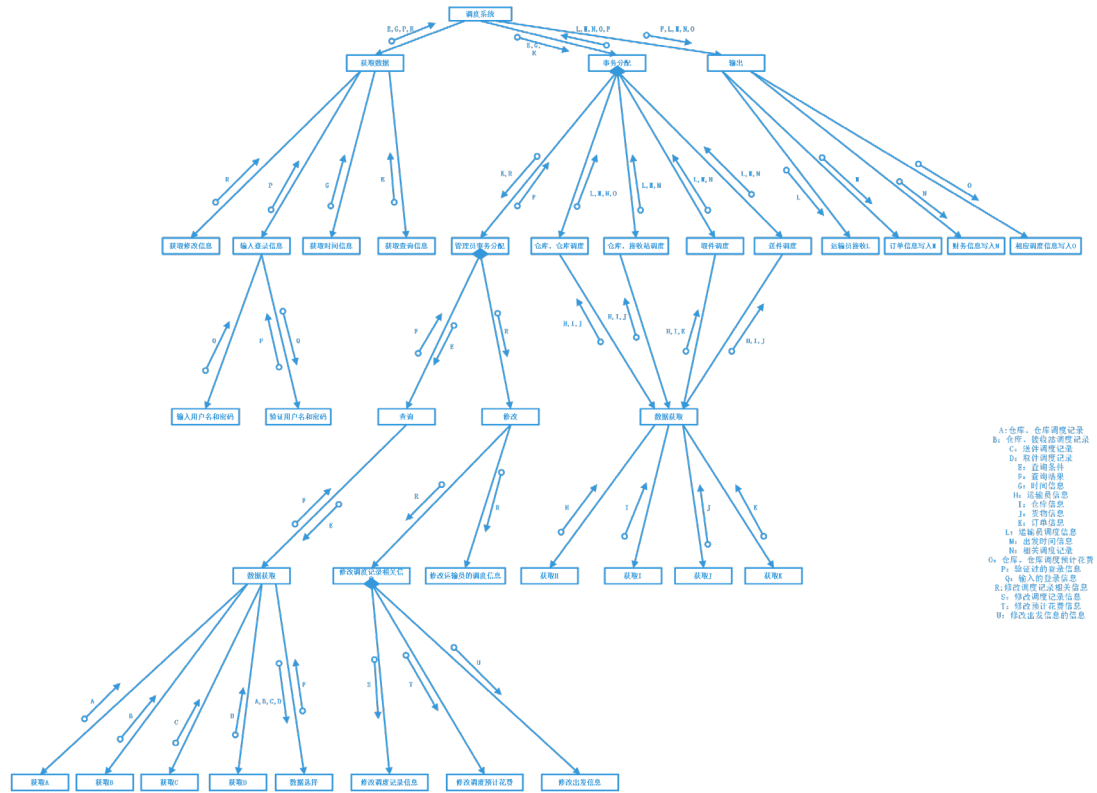
基于 HTML5 开发的 web 页面将会应用于电脑浏览器端，也会经过 webapp 打包应用到安卓手机端。

2.4 系统结构图

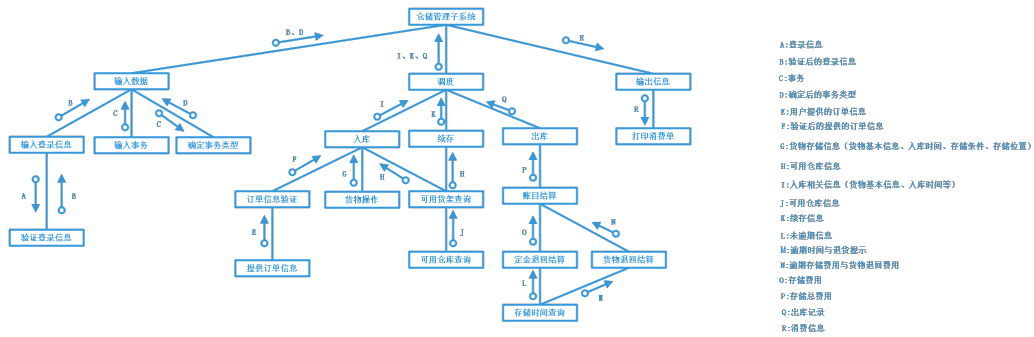
(1) 订单管理子系统



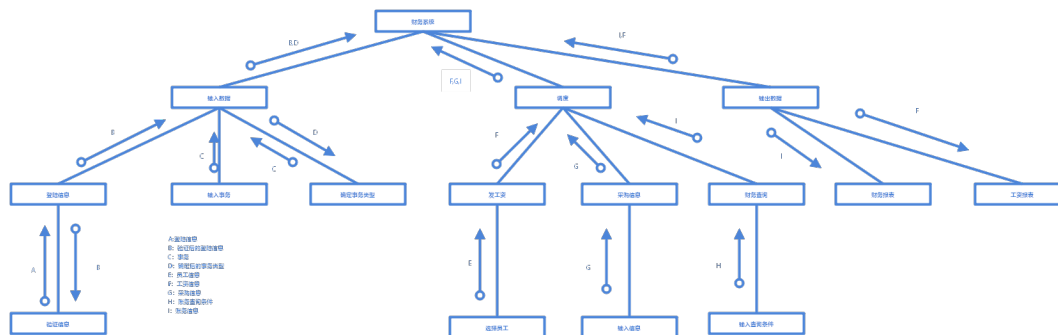
(2) 调度管理子系统



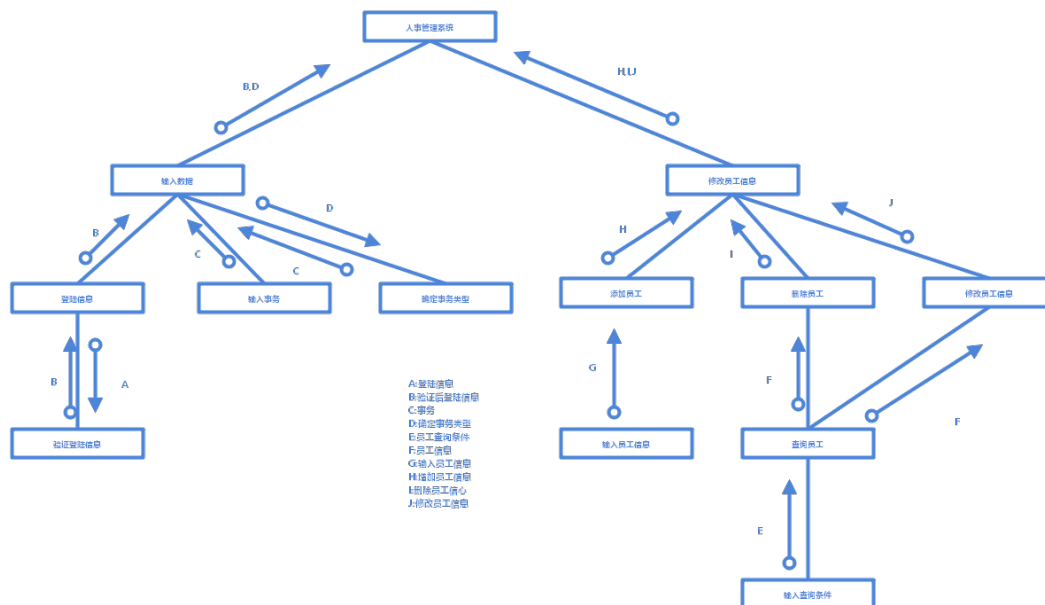
(3) 仓库管理子系统



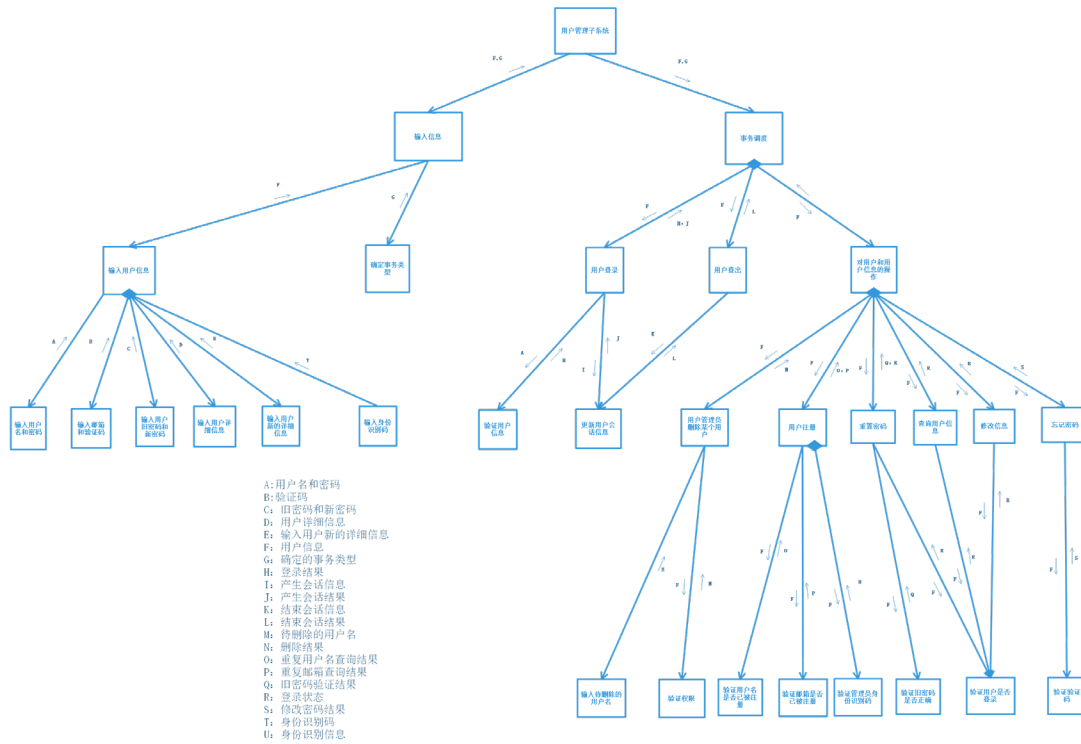
(4) 财务管理子系统



(5) 人事管理子系统

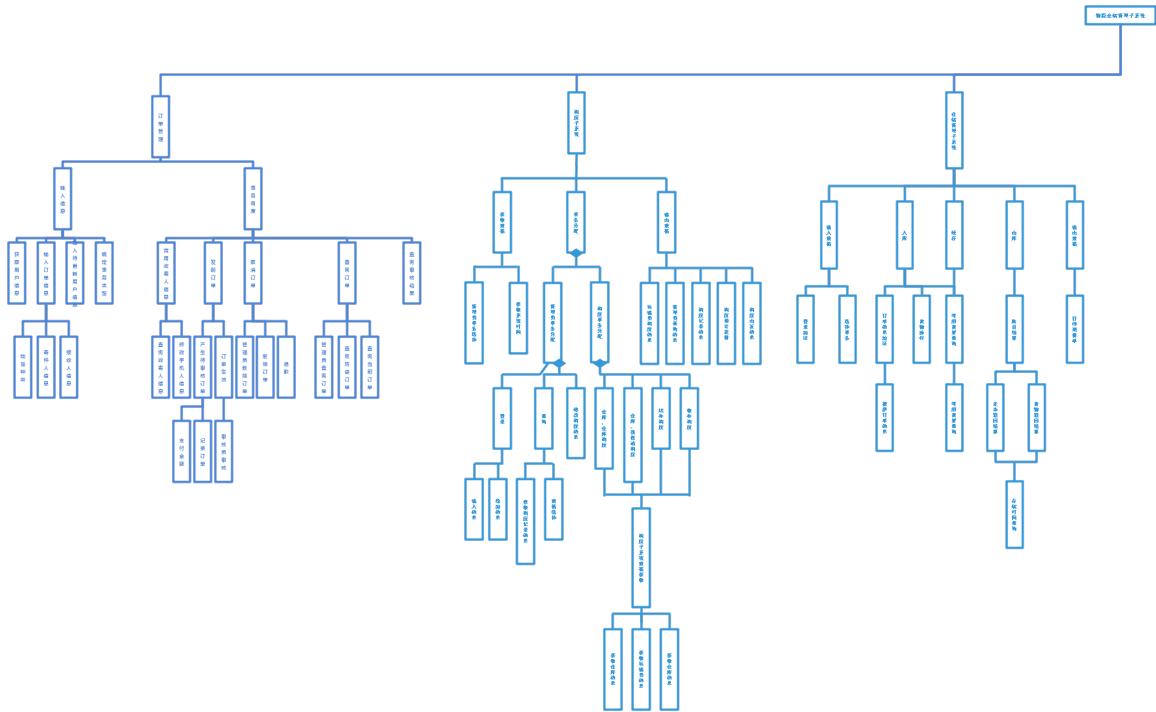


(6) 用户管理子系统

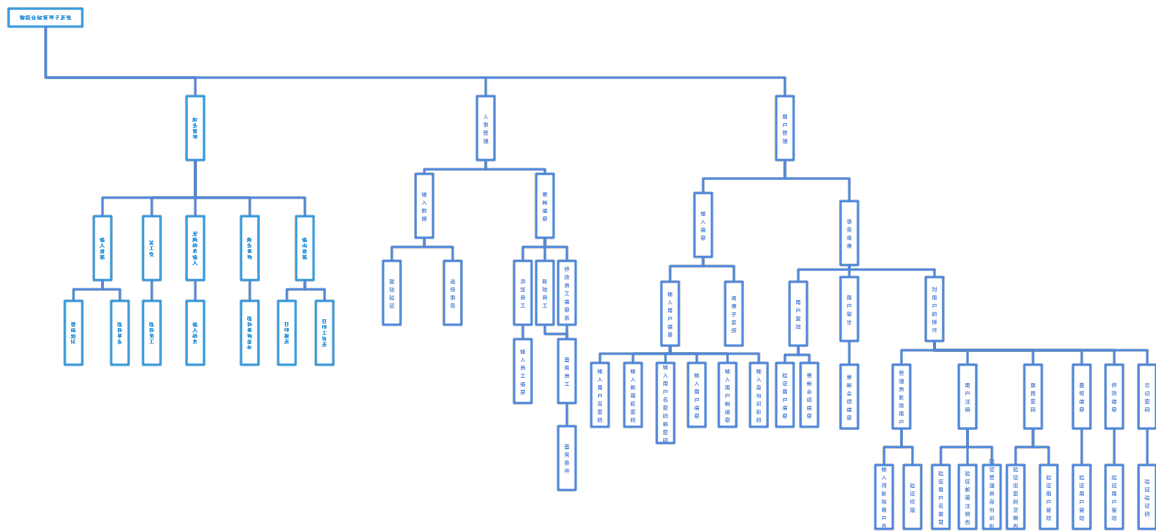


(7) 物流仓储管理系统总体结构图

①左部分



②右部分



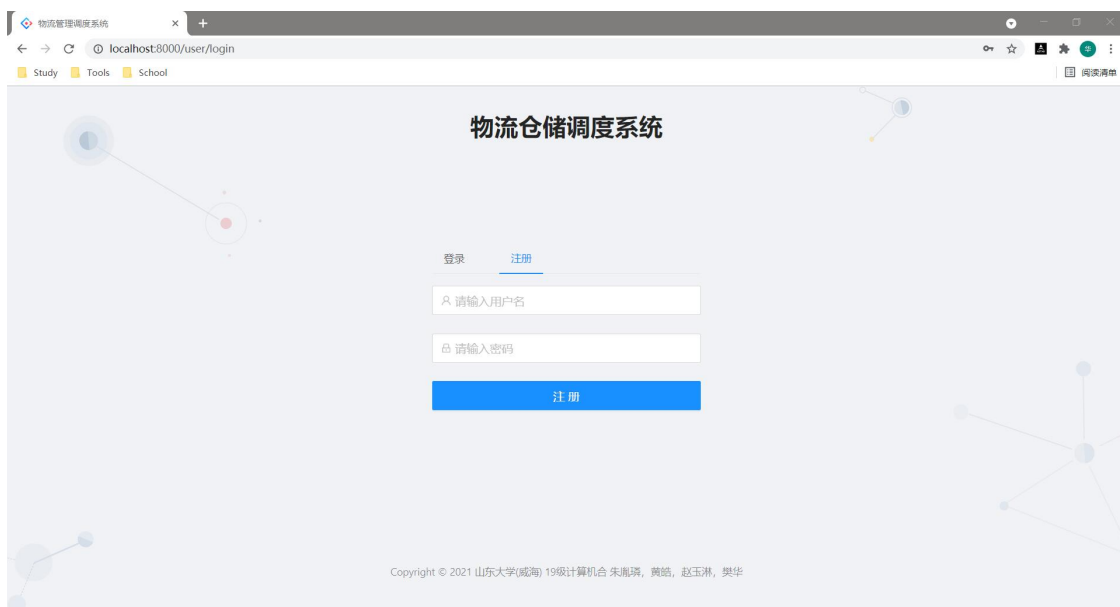
三、接口设计

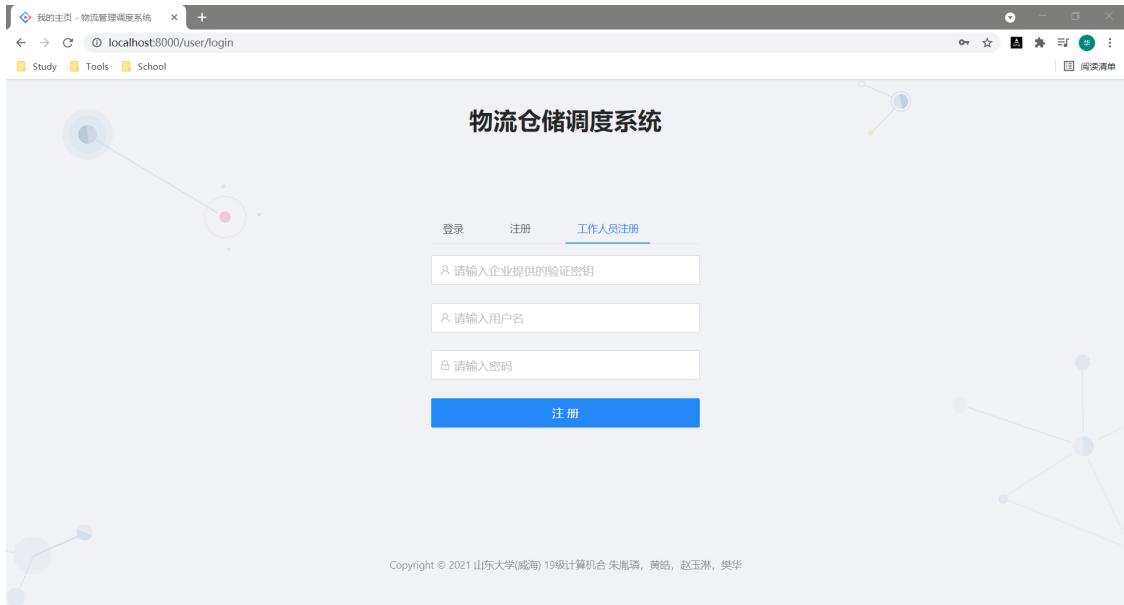
3.1 人机交互接口设计

(1) 登录界面设计

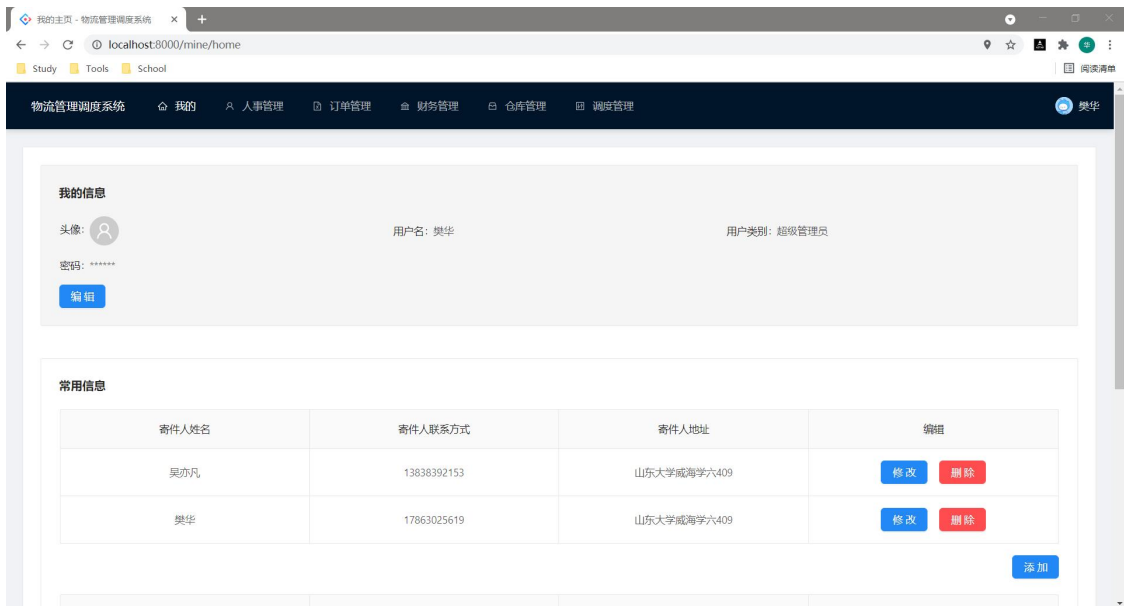


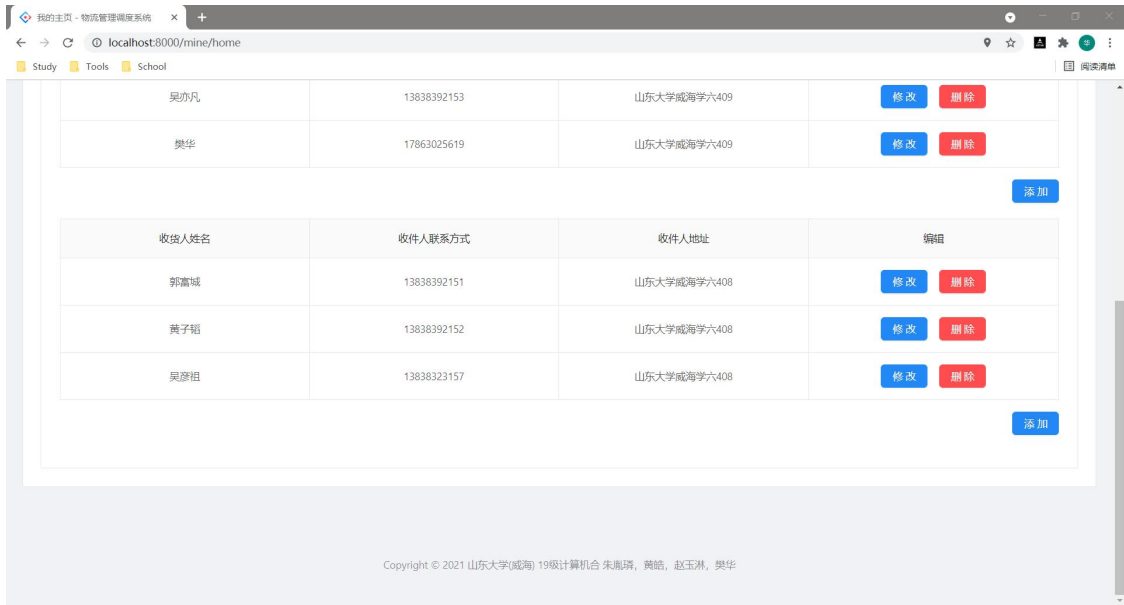
(2) 注册界面设计



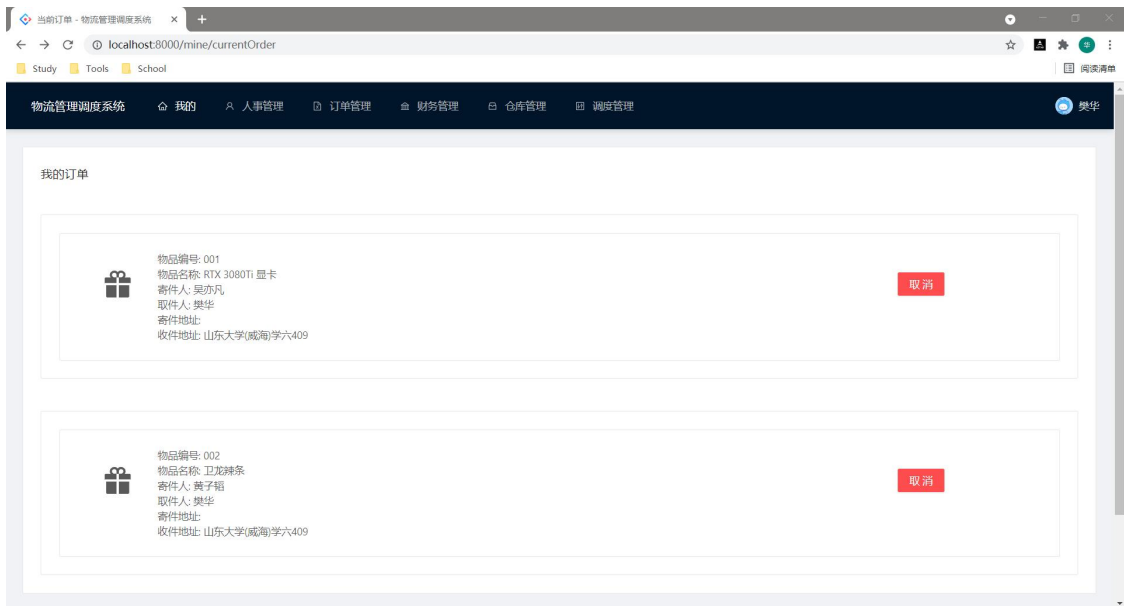


(3) 个人信息界面设计

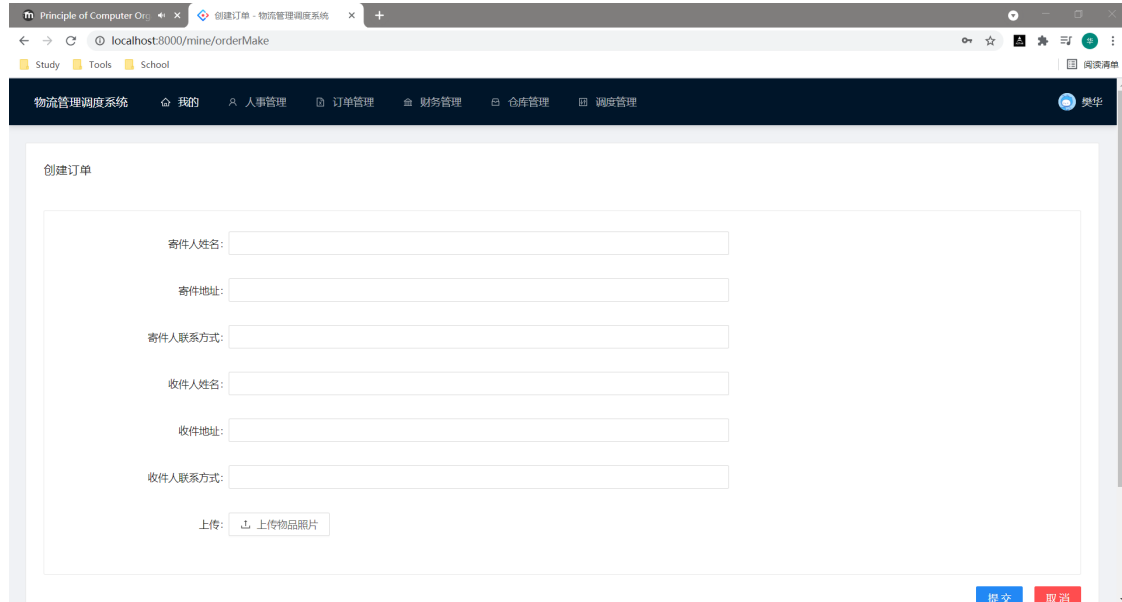




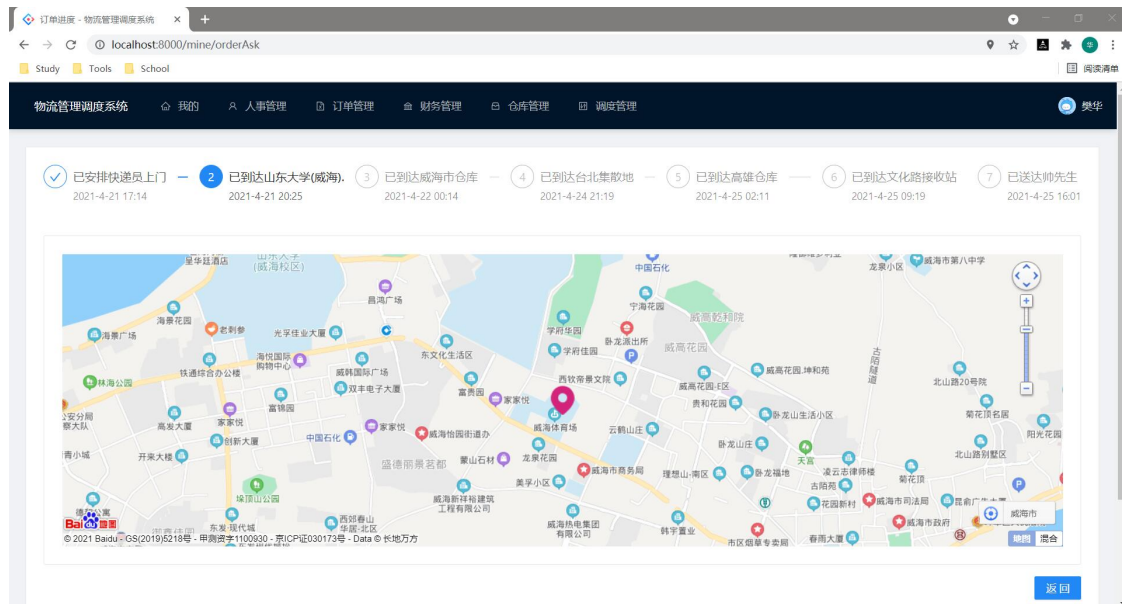
(4) 当前订单宏观查询界面设计



(5) 订单创建界面设计



(6) 正在进行订单具体信息查看界面设计



3.2 模块间接口设计

```
{  
  "transporterArray" : [  
    {  
      "id" : 1,  
      "category" : "快递员",  
      "warehouse_id" : 1,  
      "vehicle" : "电动车",  
      "capacity" : 20,  
      "phone" : "17863025618",  
      "name" : "吴彦祖"  
    },  
    {  
      "id" : 2,  
      "category" : "快递员",  
      "warehouse_id" : 1,  
      "vehicle" : "电动车",  
      "capacity" : 20,  
      "phone" : "13838392152",  
      "name" : "郭富城"  
    },  
    {  
      "id" : 3,  
      "category" : "快递员",  
      "warehouse_id" : 1,  
      "vehicle" : "电动车",  
      "capacity" : 20,
```

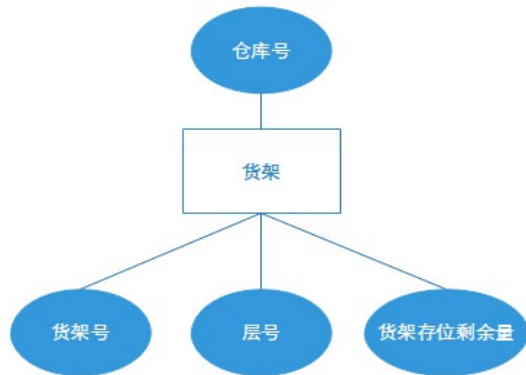
```
        "phone" : "17863025617" ,
        "name" : "黄子韬"
    }
],
"warehouseArray" : [
    {
        "id" : 2
        "position" : {
            "lat" : "117.0034"
            "lng" : "118.1223"
        },
        "category" : "接收站"
    },
    {
        "id" : 1
        "position" : {
            "lat" : "110.0034"
            "lng" : "98.1223"
        },
        "category" : "接收站"
    }
],
"orderArray" : [
    {
        "id" : 12,
        "mailAddress" : "山东大学(威海)409" ,
        "mailMan" : "樊华" ,
```

```
        "mailManPhone" : "17863025619",
        "time" : "2021-4-24 20:50:00",
        "acceptAddress" : "山东大学(威海)408",
        "acceptMan" : "帅先生",
        "acceptManPhone" : "13838392153",
        "target" : "GTX 3080Ti 显卡",
        "volume" : "small"
    },
    {
        "id" : 13,
        "mailAddress" : "山东大学(威海)409",
        "mailMan" : "陈冠希",
        "mailManPhone" : "17863025619",
        "time" : "2021-4-24 20:51:00",
        "acceptAddress" : "山东大学(威海)408",
        "acceptMan" : "帅先生",
        "acceptManPhone" : "13838392153",
        "target" : "卫龙辣条",
        "volume" : "big"
    }
]
}
```

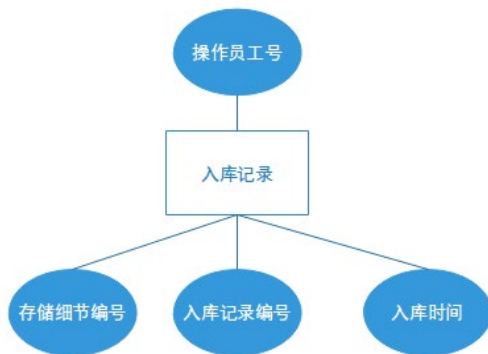
四、数据设计

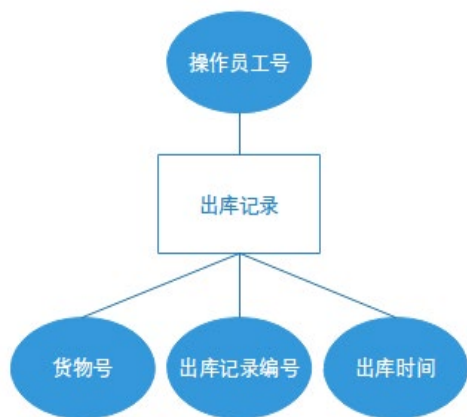
4.1 调整后的部分实体图

①货架-变更属性

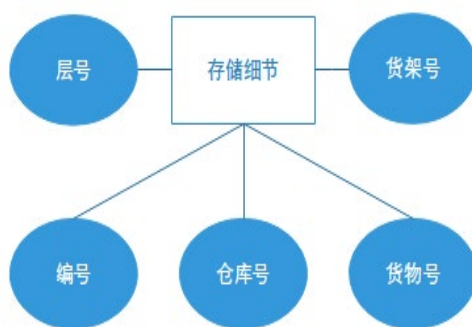


②出入记录变为出库记录、入库记录

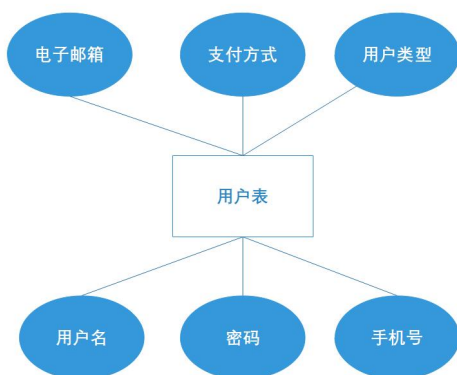




③存储细节实体优化



④用户表优化



4.2 关系模式的建立

(1) 订单管理子系统

①订单表(订单号, 订单日期, 用户名, 物品种类, 物品重量, 订单状态, 寄件信息编号, 收件信息编号, 订单金额)

②收寄信息表(收寄编号, 用户名, 收寄人姓名, 收寄人电话, 收寄人地址)

③审核结果表 (订单号, 审核员用户名, 审核日期, 审核结果, 备注)

(2) 调度管理子系统

①运输员(运输员编号, 运输员类型, 所属仓库编号, 运输载具, 运输能力)

②仓库, 仓库调度记录(调度编号, 仓库编号, 集散地编号, 调度时间, 发车方向, 运输员编号)

③仓库, 仓库调度货物(调度编号, 订单编号)

④仓库, 接收站调度记录(调度编号, 仓库编号, 接收站编号, 调度时间, 发车方向, 运输员编号)

⑤仓库, 接收站调度货物(调度编号, 订单编号)

⑥取件调度记录(调度编号, 接收站编号, 调度时间, 运输员编号)

⑦取件调度货物(调度编号, 订单编号, 预计上门时间)

⑧送件调度记录(调度编号, 接收站编号, 调度时间, 运输员编号)

⑨送件调度货物(调度编号, 订单编号, 预计送达时间)

(3) 仓库管理子系统

①货物表(货物号, 三段码信息, 货物种类)

②仓库表(仓库编号, 仓库存储量)

③货架表(货架号, 仓库号, 层号, 货架存储量)

④存储细节表(编号, 货物号, 仓库号, 货架号, 层号)

⑤入库记录表(记录号, 存储细节编号, 入库时间, 操作员工号)

⑥出库记录表(记录号, 货物号, 出库时间, 操作员工号)

(4) 财务管理子系统

①发放工资记录表(记录编号, 接受员工用户名, 发放金额, 发放时间, 管理员编号)

②采购单表(采购单编号, 采购花费, 采购日期)

③采购物品表(物品编号, 物品名称, 物品类型, 物品价格)

④采购细节表(采购记录编号, 采购单编号, 物品编号, 采购数量)

⑤工资表(用户名, 员工编号, 工资, 银行卡号)

(5) 用户管理子系统

用户表(用户名, 密码, 电子邮箱, 手机号, 支付方式, 用户类型)

4.3 数据库中表的实现

(1) 订单管理子系统

订单表 orders	存放用户的订单信息	
属性名	数据类型	约束
订单号 id	int	primary key, auto_increment
订单日期 orderdate	datetime	not null, default current_timestamp
用户名 username	varchar(20)	foreign key fkorderusername (username) references users(username)
物品种类 goodkind	varchar(20)	not null
物品重量 goodweight	float	check (goodweight > 0)
订单状态 orderstatus	nvarchar(5)	enum('待审核' , '生效' , '已完成')
寄件信息编号 sendid	int	foreign key fkordersendid (sendid) references send_receive (id)
收件信息编号 receiveid	int	foreign key fkordersendid (receiveid) references send_receive (id)
订单金额 price	float	check(price > 0)

收寄信息表 send_receive	存放用户的常用收寄人信息	
属性名	数据类型	约束
收寄编号 id	int	primary key, auto_increment

用户名 username	varchar (20)	foreign key fksrusername (username) references users (username)
收寄人姓名 srname	nvarchar (20)	not null
收寄人电话 srtelephone	datetime	check (srtelephone regexp '^1[0-9]{8}\$')
收寄人地址 srpos	nvarchar (30)	not null

审核结果表 checkresult	存放订单审核结果的细节信息	
属性名	数据类型	约束
审核员用户名 resname	varchar (20)	foreign key (resname) references users (username)
订单号 orderid	int	primary key
审核日期 resdate	datetime	Not null
审核结果 result	nvarchar (5)	enum ('通过' , ' 不通过')
备注 note	nvarchar (50)	

(2) 调度管理子系统

运输员表 transporter	存放运输员信息	
属性名	数据类型	约束
运输员编号 id	int	primary key, auto_increment
运输员类型 category		enum (' 货车司机' , ' 快递员' , ' 其它 司机')
所属仓库编号 warehouse_id	int	参考仓库表的 id 的 foreign key
运输载具 vehicle		enum (' 大货车' , ' 小货车' , ' 电动 车' , ' 其它载具')

运输能力 capacity	int	
姓名 name	nvarchar(10)	not null

仓库，仓库调度记录表 wtw_dispatch	存放仓库和集散地的调度信息	
属性名	数据类型	约束
调度编号 id	int	primary key, auto_increment
仓库编号 warehouse_id	int	参考仓库表的 id 的 foreign key
集散地编号 assign_id	int	参考仓库表的 id 的 foreign key
调度时间 time	datetime	
发车方向 direction	tinyint(1)	
运输员编号 transporter_id	int	参考运输员表的 id 的 foreign key

仓库，仓库调度货物表 wtw_goods	存放仓库和集散地之间运输的货物信息	
属性名	数据类型	约束
调度编号 id	int	和订单编号共同为 primary key
订单编号 order_id	int	和调度编号共同为 primary key

仓库，接收站调度记录表 wtr_dispatch	存放仓库和集散地的调度信息	
--------------------------	---------------	--

属性名	数据类型	约束
调度编号 id	int	primary key, auto_increment
仓库编号 warehouse_id	int	参考仓库表的 id 的 foreign key
接收站编号 receive_id	int	参考仓库表的 id 的 foreign key
调度时间 time	datetime	
发车方向 direction	bit	
运输员编号 transporter_id	int	参考运输员表的 id 的 foreign key

仓库，接收站调度货物表 wtr_goods	存放仓库和接收站之间运输的货物信息	
属性名	数据类型	约束
调度编号 id	int	和订单编号 共同为 primary key
订单编号 order_id	int	和调度编号 共同为 primary key

取件调度记录表 mail_dispatch	存放快递员取件的调度信息	
属性名	数据类型	约束
调度编号 id	int	primary key,

		auto_increment
接收站编号 receive_id	int	参考仓库表的 id 的 foreign key
调度时间 time	datetime	
运输员编号 transporter_id	int	参考运输员表的 id 的 foreign key

取件调度货物表 mail_goods	存放取件时的货物信息	
属性名	数据类型	约束
调度编号 id	int	和订单编号共同为 Primary key
订单编号 order_id	int	和调度编号共同为 Primary key
预计上门时间 time	datetime	

送件调度记录表 accept_dispatch	存放快递员送件的调度信息	
属性名	数据类型	约束
调度编号 id	int	Primary key, auto_increment
接收站编号 receive_id	int	参考仓库表的 id 的 foreign key
调度时间 time	datetime	
运输员编号 transporter_id	int	参考运输员表的 id 的 foreign key

送件调度货物表 accept_goods	存放送件时的货物信息	
属性名	数据类型	约束

调度编号 id	int	和订单编号共同为 Primary key
订单编号 order_id	int	和调度编号共同为 Primary key
预计送达时间 time	datetime	

(3) 仓库管理子系统

货物表 Goods	存放货物信息	
属性名	数据类型	约束
货物号 Goods_Num	Int	PRIMARY KEY AUTO_INCREMENT
三段码信息 TSegmentCode_Info	Varchar (20)	
货物种类 Goods_kinds	Int	

仓库表 warehouse	存放仓库信息	
属性名	数据类型	约束
仓库编号 WareHouse_num	int	PRIMARY KEY AUTO_INCREMENT
仓库存储量 WareHouse_StockNum	Int	

存储细节表 Storage_Details	存储的信息	
属性名	数据类型	约束
编号 Details_Num	Int	PRIMARY KEY AUTO_INCREMENT
仓库号 WareHouse_Num	Int	FOREIGN KEY (WareHouseNum) REFERENCES Warehouse (WareHouseNum)
货物号 Goods_Num	Int	FOREIGN KEY (GoodsNum) REFERENCES Goods (GoodsNum)
货架号 Shelves_Num	Int	FOREIGN KEY (ShelvesNum) REFERENCES Shelves (ShelvesNum)
层号 Floor_Num	Int	

货架表 Shelves	存放货架信息	
属性名	数据类型	约束
货架号 Shelves_Num	Int	PRIMARY KEY AUTO_INCREMENT
仓库号 WareHouse_Num	Int	FOREIGN KEY (WareHouseNum) REFERENCES WareHouse (WareHouseNum)
层号 Floor_Num	Int	
货架存储量 Shelves_StockNum	Int	

入库记录 Warehousing_Record	存放货物入库信息	
属性名	数据类型	约束
记录号 Records_Num	Int	PRIMARY KEY AUTO_INCREMENT,
存储细节编号 Details_Num	Int	FOREIGN KEY (DetailsNum) REFERENCES StorageDetails (Details_Num)
入库时间 WH_time	Datetime	
操作人员号 Operator_Num	Varchar (20)	FOREIGN KEY (Operator_Num) REFERENCES Users (username

出库表 ExWarehousingRecord	存放出库信息	
记录号 ExRecordNum	Int	PRIMARY KEY AUTO_INCREMENT,
货物号 Goods_Num	Int	FOREIGN KEY (GoodsNum) REFERENCES Goods (GoodsNum)
出库时间 ExWHtime	Datetime	
操作人员号 Operator_Num	Int	FOREIGN KEY (Operator_Num)

		REFERENCES Users(username
--	--	---------------------------

(4) 财务管理子系统

发放工资记录表 payoff	存放财务管理员每次发放工资的信息	
属性名	数据类型	约束
记录编号 id	Int	Primary key, auto_increment
管理员编号 paymanid	Int	参考员工表中 id 的 foreign key
发放时间 datetime	Datetime	当前时间
接收员工用户名 username	Varchar (20)	参考员工表中的 username 的 foreign key
发放金额 money	Int	

采购单表	存放采购员采购的账单	
属性名	数据类型	约束
采购单编号 id	Int	primary key, auto_increment
采购日期 buydate	Datetime	当前时间
采购花费 cost	Int	

采购物品表	采购物品的信息	
属性名	数据类型	约束
物品编号 id	Int	primary key, auto_increment
物品名称 name	Varchar (40)	Not null
物品类型 type	Int	
物品价格 price	Int	

采购细节表	采购单中买了什么	
属性名	数据类型	约束
采购记录编号 <i>billrecordid</i>	Int	primary key, auto_increment
采购单编号 <i>bill_id</i>	Int	参考采购单中的 id 的 foreign key
物品编号 <i>thing_id</i>	Int	参考采购物品中的 id 的 foreign key
采购数量 <i>num</i>	Int	Not null

工资表 <i>salary</i>	存放员工和工资	
属性名	数据类型	约束
用户名 <i>workername</i>	Varchar (20)	Primary key 参照 users 表中的 username
员工编号 <i>workerid</i>	Int	参照 users 表中的 id
工资 <i>money</i>	Int	
银行卡号 <i>banknum</i>	Int	Notnull

(5) 用户管理子系统

用户表 <i>users</i>	存放所有普通用户及员工用户的信息	
属性名	数据类型	约束
用户名 <i>username</i>	varchar (20)	primary key
密码 <i>userpassword</i>	varchar (50)	not null
电子邮箱 <i>email</i>	varchar (30)	check(email regexp '^([a-z0-9A-Z]+[- a-z0-9A-Z . _]+@([a-z0-9A-Z]+(-[a-z0-9A-Z]+)?\.)+[a-z]{2,})\$')

手机号 telephone	char(11)	check (telephone regexp '1[0-9]{8}\$')
支付方式 payment	nvarchar(30)	
用户类型 usertype	nvarchar(20)	

4.4 建表语句 (Mysql)

— 订单表

```
create table orders
(
    id int auto_increment primary key,
    orderdate datetime not null default current_timestamp,
    username varchar(20), foreign key fk_order_username (username) references users (username),
    goodkind varchar(20) not null,
    goodweight float , check (goodweight > 0),
    orderstatus nvarchar(5) enum('待审核', '生效', '已完成'),
    sendid int foreign key fk_order_sendid (sendid) references send_receive (id),
    receiveid int foreign key fk_order_receiveid (receiveid) references send_receive (id),
    price float check (price > 0)
)
```

— 收寄信息表

```
create table send_receive
(
    id int auto_increment primary key,
    username varchar(20) foreign key fk_sr_username (username) references users (username),
    srname nvarchar(20) not null,
    srtelephone datetime check (srtelephone regexp '^[1](3[4-9]|47|5[012789]|8[23478])
```

```
[0-9]{8}$'),  
srpos nvarchar(30) not null  
)
```

-- 审核结果表

```
create table checkresult  
(  
  
orderid int primary key,  
resname varchar(20) foreign key (resname) references users(username),  
resdate datetime not null,  
result nvarchar(5) enum('通过','不通过'),  
note nvarchar(50)  
)
```

-- 运输员表

```
create table transporter  
(  
  
    id int auto_increment primary key,  
    category enum('货车司机','快递员','其它司机'),  
    wraehouse_id int references warehouse (id),  
    vehicle enum('大货车','小货车','电动车','其他载具'),  
    capacity int,  
    name nvarchar(10) not null  
)
```

-- 仓库, 仓库调度记录表

```
create table wtw_dispatch  
(  
  
    id int auto_increment primary key,  
    warehouse_id int references warehouse (id),  
    assign_id int references warehouse (id),
```

```
        time datetime,  
        direction tinyint(1),  
        transporter_id int references transporter (id)  
    )
```

-- 仓库, 仓库调度运输的货物表

```
create table wtw_goods  
(  
    id int not null,  
    order_id int not null,  
    primary key(id, order_id)  
)
```

-- 仓库, 接收站调度记录表

```
create table wtr_dispatch  
(  
    id int auto_increment primary key,  
    warehouse_id int references warehouse (id),  
    receive_id int references warehouse (id),  
    time datetime,  
    direction tinyint(1),  
    transporter_id int references transporter (id)  
)
```

-- 仓库, 接收站调度运输的货物表

```
create table wtr_goods  
(  
    id int not null,  
    order_id int not null,  
    primary key(id, order_id)  
)
```

-- 取件调度记录表

```
create table mail_dispatch
(
    id int auto_increment primary key,
    receive_id int references warehouse (id),
    time datetime,
    transporter_id int references transporter (id)
)
```

-- 取件调度运输的货物表

```
create table mail_goods
(
    id int not null,
    order_id int not null,
    time datetime,
    primary key(id,order_id)
)
```

-- 送件调度记录表

```
create table accept_dispatch
(
    id int auto_increment primary key,
    receive_id int references warehouse (id),
    time datetime,
    transporter_id int references transporter (id)
)
```

-- 送件调度运输的货物表

```
create table accept_goods
(
    id int not null,
    order_id int not null,
```

```
        time datetime,  
        primary key(id,order_id)  
    )
```

-- 货物表

```
CREATE TABLE Goods  
(  
    Goods_Num INT PRIMARY KEY AUTO_INCREMENT,  
    TSegmentCode_Info VARCHAR(20),  
    Goods_Kinds INT  
);
```

-- 仓库表

```
CREATE TABLE WareHouse  
(  
    WareHouse_Num INT PRIMARY KEY AUTO_INCREMENT,  
    WareHouse_StockNum INT  
);
```

-- 货架表

```
CREATE TABLE Shelves  
(  
    Shelves_Num INT PRIMARY KEY AUTO_INCREMENT,  
    WareHouse_Num INT,  
    Floor_Num INT,  
    Shelves_StockNum INT,  
    CONSTRAINT Fk_WarehouseNum_2 FOREIGN KEY(WareHouse_Num) REFERENCES WareHous  
e(WareHouse_Num)  
);
```

-- 存储细节表

```
CREATE TABLE Storage_Details
```

```
(
    Details_Num INT PRIMARY KEY AUTO_INCREMENT,
    Warehouse_Num INT,
    Goods_Num INT,
    Shelves_Num INT,
    Floor_Num INT,
    CONSTRAINT Fk_WarehouseNum_1 FOREIGN KEY(Warehouse_Num) REFERENCES Warehouse(Warehouse_Num),
    CONSTRAINT Fk_goodsNum FOREIGN KEY(Goods_Num) REFERENCES Goods(Goods_Num),
    CONSTRAINT Fk_shelvesNum FOREIGN KEY(Shelves_Num) REFERENCES Shelves(Shelves_Num)
);
```

— 入库记录表

```
CREATE TABLE Warehousing_Record
(
    Records_Num INT PRIMARY KEY AUTO_INCREMENT,
    Details_Num INT,
    WH_time DATETIME DEFAULT CURRENT_TIMESTAMP,
    Operator_Num VARCHAR(20),
    CONSTRAINT Fk_storageDetails FOREIGN KEY(Details_Num) REFERENCES Storage_Details(Details_Num),
    CONSTRAINT Fk_operatorNum_1 FOREIGN KEY(Operator_Num) REFERENCES Users(username)
);
```

— 出库记录表

```
CREATE TABLE Ex_Warehousing_Record
(
    Ex_Record_Num INT PRIMARY KEY AUTO_INCREMENT,
    Goods_Num INT,
    Ex_WH_time DATETIME DEFAULT CURRENT_TIMESTAMP,
```

```

        Operator_Num VARCHAR(20),
        CONSTRAINT Fk_storageDetails FOREIGN KEY(Goods_Num) REFERENCES Goods(Goods_
Num),
        CONSTRAINT Fk_operatorNum_2 FOREIGN KEY(Operator_Num) REFERENCES Users(user
name)
);

```

-- 采购单表

```

create table buybill
(
    id int auto_increment primary key,
    buydate datetime default now(),
    money int
);

```

-- 采购物品表

```

create table buything
(
    id int auto_increment primary key,
    name varchar(40) not null,
    type VARCHAR(10) ,
    pricce int
);

```

-- 工资表

```

create table salary
(
    workername varchar(20) PRIMARY KEY FOREIGN key(workername) REFERENCES users(userna
me),
    workerid int FOREIGN KEY(workerid) REFERENCES users(id),
    money int,

```



```
banknum int,
```

```
);
```

-- 采购细节表

```
create table buydetail
```

```
(
```

```
detail_id INT primary KEY
```

```
buybillid int, foreign key(buybillid) references buybill(id),
```

```
buythingid int, foreign key(buythingid) references buything(id),
```

```
num int not null,
```

```
);
```

-- 发工资记录表

```
create table payoff
```

```
(
```

```
id int auto_increment primary key, paymaname varchar(20),
```

```
foreign key(paymaname) references users(username), receiverid varchar(20),
```

```
foreign key(receiverid) references users(username), time datetime default now(),
```

```
should int,
```

```
really int
```

```
);
```

-- 用户表

```
create table users
```

```
(
```

```
username varchar(20) primary key,
```

```
userpassword varchar(50) not null,
```

```
email varchar(30) check(email regexp '^[a-z0-9A-Z]+[-|a-z0-9A-Z . _]+@[a-z0-9A-Z]+(-[a-z0-9A-Z]+)?\\.|[a-z]{2,}$') ,
```

```
telephone char(11) check(telephone regexp '^[1](3[4-9]|47|5[012789]|8[23478])[0-9]{8}$' ),
```

```
payment nvarchar(30),
```

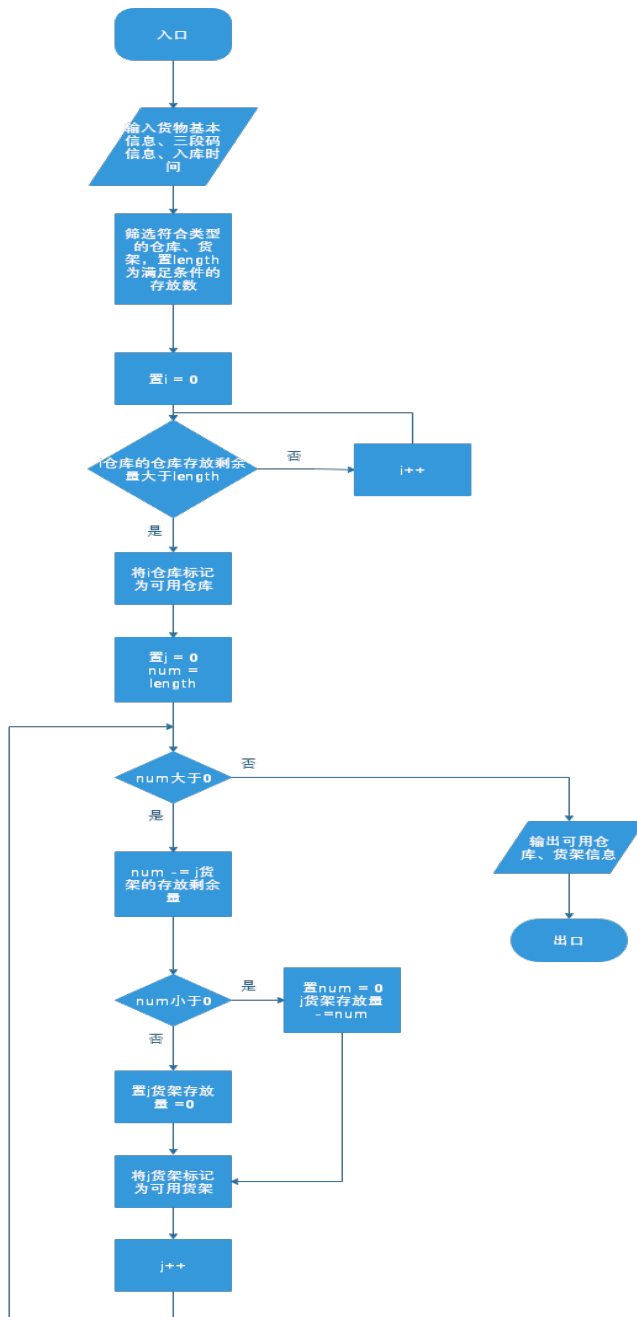
```
usertype nvarhchar(20)
```

```
)
```

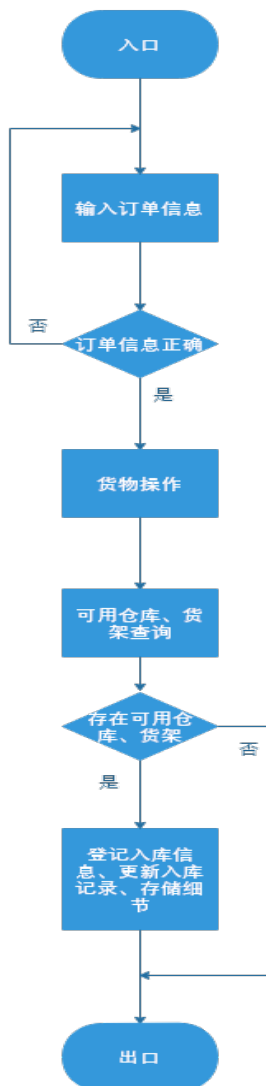
五、过程设计

5.1 主要功能模块的程序流程图

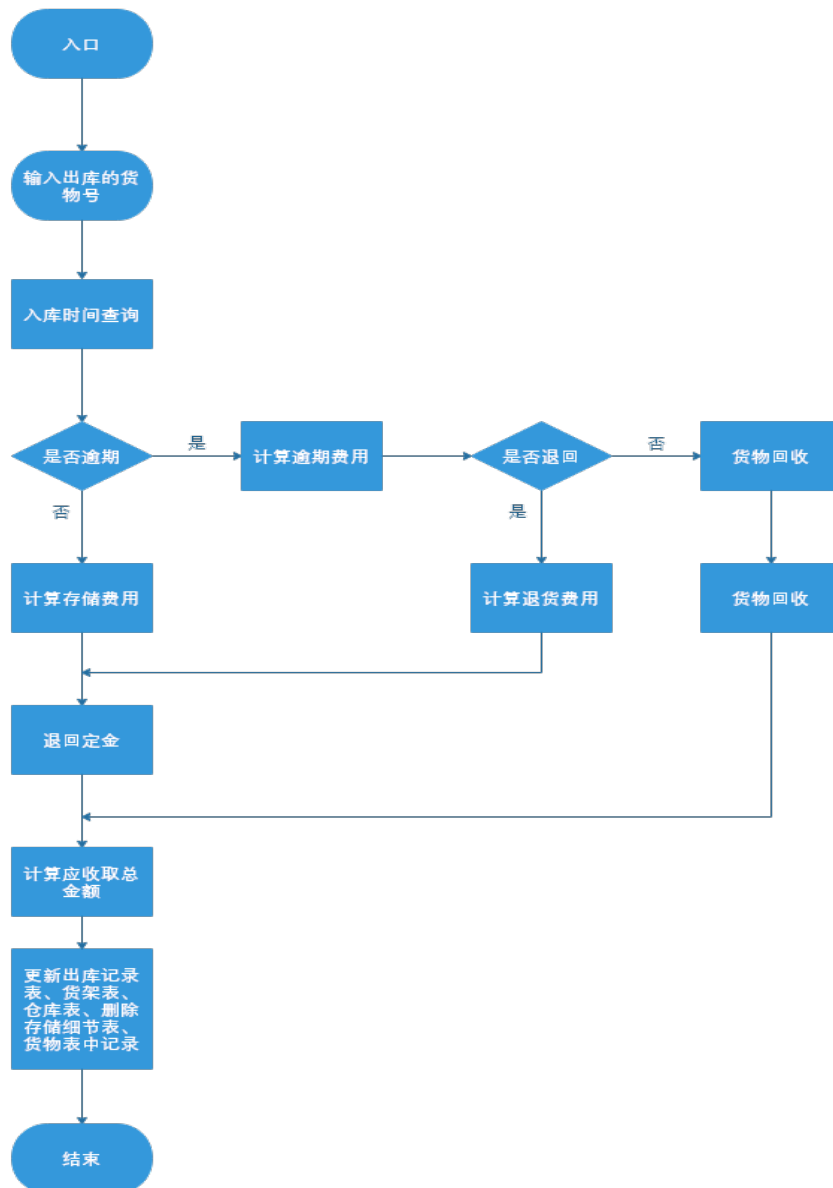
(1) 可用仓库、货架查询



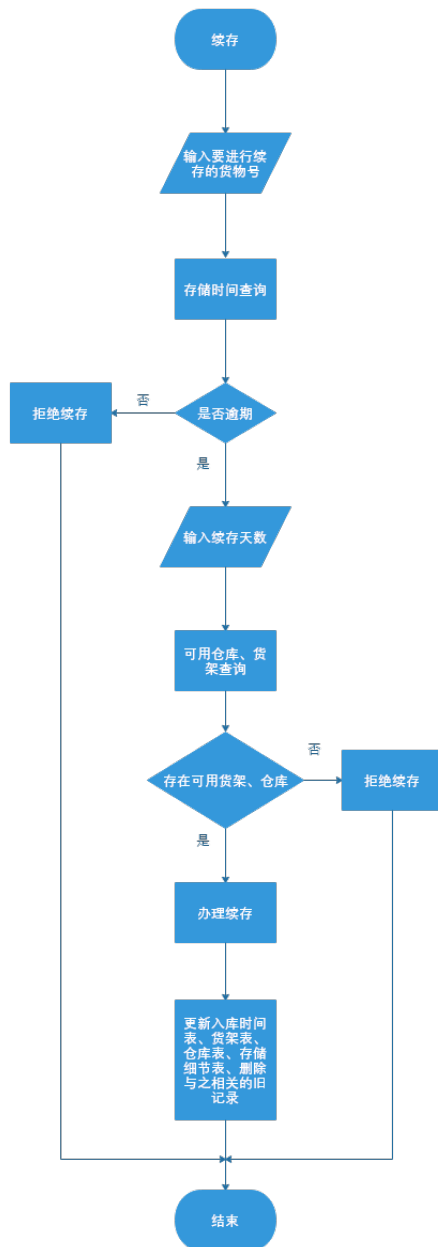
(2) 入库办理



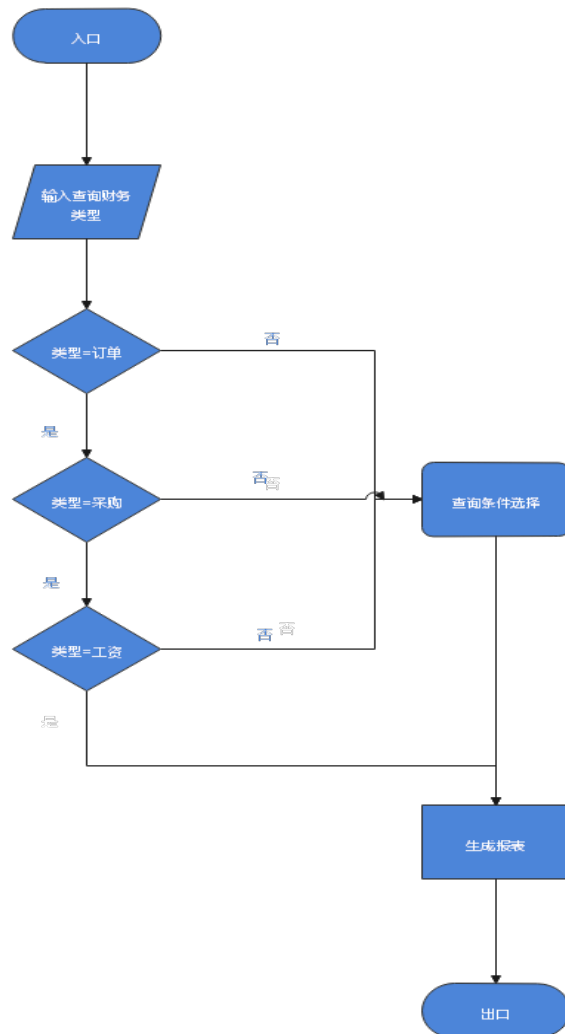
(3) 出库



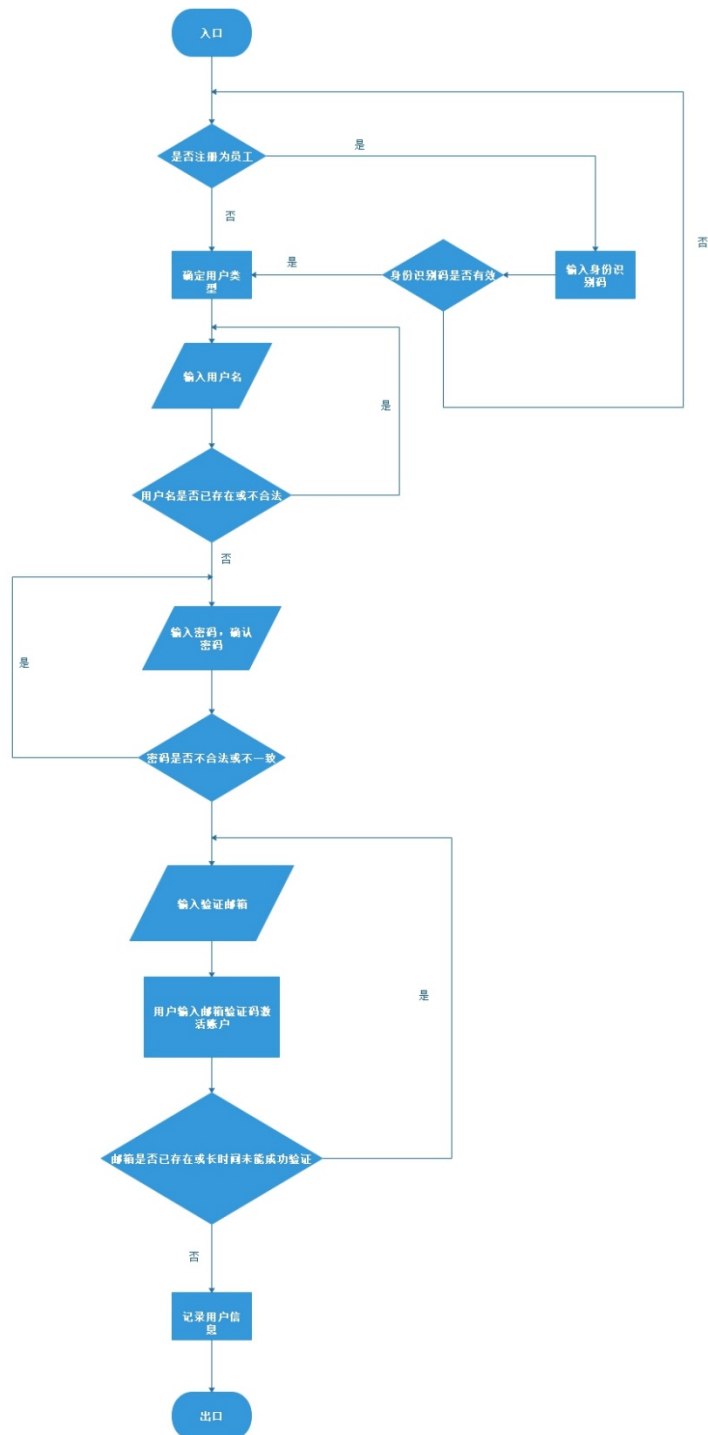
(4) 续存



(5) 财务查询流程



(6) 注册流程



5.2 重要算法的伪代码表述

(1) 取件调度任务分配

```
public void mailDispatch () {  
    List<Transporter> T = 获取运输员信息列表;  
    List<Warehouse> W = 获取仓库信息列表;  
    List<Order> O = 获取订单信息列表;  
  
    //遍历接收站信息列表  
    for (Warehouse i : W) {  
        tempO = O.stream().filter(该订单在当前接收站的服务范围内).collect(Collectors.toList());  
        tempT = O.stream().filter(该接收站的运输员).collect(Collectors.toList());  
  
        //遍历该接收站的运输员  
        for (Transporter j : tempT) {  
            //运输员电动车容量  
            int capacity = j.capacity;  
            //该运输员需要运输的订单中的货物  
            List<Order> thisO = new ArrayList<Order>();  
            //该运输员还可以运输时不断给他分配快递  
            while (capacity > 0) {  
                for (Order k : tempO)  
                    if (该订单没有被安排) {  
                        capacity -= k 的体积;  
                        thisO.add(k);  
                        修改订单状态为已分配;  
                    }  
            }  
            把 thisO 和相关信息写入取件调度运输的货物记录;  
            把 j 和相关信息写入取件调度记录;  
        }  
    }  
}
```



```
        更新订单表中订单状态;
    }
}

//释放内存
T = null;
W = null;
O = null;
}
```

（2）取件调度获取百度地图调度信息

```
public void mailDispatchFrontend() {
    List<mailMapResult> R = 获取前端百度地图分配的结果列表;
    根据列表中的订单编号和预计上门时间更新订单表中的订单状态;
}
```