# 超市(进销存)管理系统

#### 软件设计背景分析：

做为一次JAVA EE 企业级开发实训，本系统更多的是面向开发学习，而不是客户需求。因此，该软件的功能设计在完成基础的实训要求之外，应该更多着重于技术学习与挑战 。使得系统具备一定的可维护性、可重用性、可拓展性。

#### 软件功能需求分析：

###### 商品管理：（必要）

商品增删改查，分类，上下架，保质期问题管理等操作。（可以考虑做个过期提醒---定时邮件通知或者在首页控制台显示一个列表）

###### 人员管理：（必要）

人员增删改查，权限修改（权限和角色一般都在系统管理模块、人员和组织架构可以放一个模块）

###### 库存管理：（必要）

包括商品的入库、出库、库存统计、库存报警等（多仓库、调货功能）

###### 数据分析：（拓展）

主要是对系统产生的销售数据、用户数据、人员数据、库存数据进行分析处理等操作

###### 系统管理：（必要）

对本系统进行一定的自定义操作，例如：功能的禁用、主题修改、设置定时任务等。

###### 用户管理：（拓展）作为系统管理的子模块？

主要是普通用户与会员用户之间的差异管理

###### 权限控制：（必要）作为系统管理的子模块？

以上所有管理操作均涉及权限控制

采购（订单）管理：必要 ----采购流程？工作流设计

# 功能需求

1. 超级管理员
   1. 权限管理
      1. 编辑现有员工的权限
   2. 职员管理
      1. 添加员工
      2. 查询员工
      3. 编辑员工信息
      4. 解雇员工
   3. 查看销售报表\*
      1. 月进货成本、销售额
      2. 成交波浪图
   4. 商品管理
      1. 新设商品
      2. 商品停售
      3. 商品信息编辑
   5. 系统管理（对系统进行一定的自定义操作）
2. 采购员
   1. 采购（订单）管理
      1. 新建采购单
      2. 删除采购单
      3. 编辑采购单
      4. 查询采购单
   2. 提交采购单审核
   3. 通知取货
   4. 查看库存（库存危险自动高亮，信息框）
   5. 处理退货
3. 仓库管理员
   1. 确认进货
   2. 货品上架
   3. （坏旧）货品下架
4. 销售员
   1. 结账
   2. 开会员卡
      1. 预存
      2. 会员卡享9折
      3. 100积分抵购1元
   3. 注销会员卡

# 技术分析

# 类表（字典）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Meaning | Type | Members | | |
| User | 用户 | Class | Long id | 数据库管理 | |
| UserType userType | 用户类型 | |
| Set<Privilege>  privileges | 权限 | |
| String workNum | 工号用作登陆账户 | |
| String password | 密码 | |
| Date hireDate | 雇佣日期 | |
| String name | 姓名 | |
| String identity | 身份证 | |
| Gender gender | 性别 | |
| String iconUrl | 头像路径 | |
| UserStatus userStatus | 用户状态 | |
| UserType | 用户类型 | Enum | MANAGER | 超级管理员 | |
| PURCHASER | 采购员 | |
| KEEPER | 商品管理员 | |
| SALESMAN | 销售员 | |
| Gender | 性别 | Enum | MALE | 男 | |
| FEMALE | 女 | |
| UserStatus | 用户状态 | Enum | LAIDOFF | 被解雇的 | |
| FROZEN | 冻结的 | |
| NORMAL | 正常的 | |
| Privilege | 权限 | Enum | ALL | 所有 | |
| DIAGRAMS | 查看统计记录 | |
| EDIT\_USER | 编辑所有用户信息 | |
| EDIT\_SELF | 编辑个人信息 | |
| EDIT\_COMMODITY | 编辑商品信息 | |
| EDIT\_INDENT | 编辑采货单 | |
| CHECK\_INDENT | 审批采购单 | |
| LOOK\_INDENT | 查看采购单 | |
| CHANGE\_SHELF | 修改货架在售量 | |
| CHANGE\_WAREHOUSE | 修改库存量 | |
| Commodity | 商品 | Class | Long id |  | |
| Long barCode | 条形码-确认同一种商品是唯一的价格 | |
| Date period | 有效期 | |
| String name | 商品名称 | |
| String picUrl | 预览图 | |
| Int amount | 数量 | |
| Int saveStock | 安全库存 | |
| Double price | 售价 | |
| Double cost | 进货价 | |
| String note | 备注，如下架原因（过期、受损、变质、政府强制下架、等等）、保存条件（冷藏、干燥等） | |
| Boolean saleable | 是否可售 | |
| Indent indent | 订单的追溯 | |
| Indent | 货品单 | Class | Long id |  | |
| String indentNum | 订单编号 | |
| List<Commodity> commodities | 商品列表 | |
| Double cost | 订单总费用 | |
| IndentStatus | 状态 | |
| Date createDate | 订单创建日期 | |
| User creator | 订单创建者。采购时是采购员，内部调货时是货品管理员 | |
| User keeper | 货品管理员 | |
| User manager | 管理员 nullable | |
| Boolean isInside | 内部调货，如果为真，统计进货成本的时候就排除在外 | |
| Warehouse fromWarehouse | 调出 nullable | |
| Warehouse toWarehouse | 调入 | |
| String note | 信息，由用户填入，比如缺货少货的原因等，存入是this.note += User.name+获取的信息+”\n”;意思是这个字段记录的可以是多个用户在不同环节写的信息 | |
| IndentStatus | 订单状态 | Enum | INIT | 采购员创建和编辑 | 商品管理员创建编辑 |
| CHECKING | 采购员提交后的状态，管理员审批，此状态可允许修改商品数量? | / |
| APPROVED | 管理员通过后的状态，此状态认为采购员和供应商交易中 | / |
| EXTRACTING | 采购员确认供应商货物后的状态，此状态下认为货物在仓库附近或运输途中，由仓库管理员进行确认 | 商品管理员确认调货操作后的状态，此状态交由收货仓库的商品管理员进行确认 |
| FINISHED | 仓库管理员确认货物进仓后，订单完成 | 收货仓商品管理员确认无误后调货完成 |
| ERROR | 仓库管理员认为货物与订单有出入，订单由采购员处理，处理后可以将订单重新设定为EXTRACTING状态 | 商品管理员可拒收，退回INIT |
| VIPCard | 会员卡 | Class | Long id |  | |
| Long cardNo | 会员卡号 | |
| Int score | 积分 | |
| Static Double discount | 会员卡折扣 | |
| Static int num | 会员人数 | |
| Date openDate | 开卡日 | |
| Deal | 销售单 | Class | Long id |  | |
| VipCard | 会员卡 | |
| User salesman | 经手人 | |
| List<Commodity> commodities | 列表 | |
| Double price | 总价 | |
| Double cost | 总进货价 | |
| Warehouse | 仓库 | Class | Long id |  | |
| String name |  | |
| List<Commodity> badCommodities | 过期品 | |
| List<Commodity> commodities | 存货 | |
| String location | 地理位置 | |
| List<User> keepers | 一个仓库可以有多个货品管理员看守 | |
| Shop | 门店 | Class |  |  | |
|  |  | |