<<基于Web的进销存仓库管理系统>>

需求分析文档

项目名称 基于Web的进销存仓库管理系统

学生学号 201726702112

学生姓名 励轩

指导教师 邓有莲

完成时间 2020年11月

目录

[1. 引言 3](#_Toc27842)

[1.1编写目的 3](#_Toc14012)

[1.2开发背景 3](#_Toc14944)

[1.3目前研究现状 4](#_Toc23211)

[1.4开发目标 4](#_Toc18520)

[2.系统需求 5](#_Toc25369)

[2.1角色分析 5](#_Toc14635)

[2.2 系统功能及用户需求分析 5](#_Toc3811)

[2.3软、硬件需求 6](#_Toc25890)

[2.4开发语言、开发工具需求 6](#_Toc32419)

[2.5非功能性需求 6](#_Toc13846)

[3.用例分析 7](#_Toc14799)

[3.1总用例分析 7](#_Toc20126)

[3.2数据维护用例分析 8](#_Toc5552)

[3.3用户权限管理分析 10](#_Toc16296)

[4.流程分析 10](#_Toc21795)

[4.1入库 10](#_Toc26557)

[4.2单号生成 11](#_Toc20286)

[4.3入库退货 13](#_Toc12674)

[4.4商品添加 14](#_Toc7250)

[4.5权限分配 14](#_Toc21305)

[4.6授权流程 15](#_Toc1004)

[5.功能模块分析 16](#_Toc32589)

[6.架构设计 17](#_Toc31663)

[参考文献 18](#_Toc13865)

# 引言

## 1.1编写目的

本文档的目的在于明确系统开发的意图,包括系统的目标人群,系统需求以及界定系统的功能范围等,此文档将指导系统的设计和编码,并能够向读者说明有关此系统开发的背景.

## 1.2开发背景

近年来,各种电商平台发展迅速,各种直播带货微商等平台层出不穷,如雨后春笋一般,由此出现了一大批小型规模的店铺.另一方面,仓库管理办公自动化正处于蓬勃发展时期。企业要想在市场竞争中时刻保持强大的竞争力，必须实现现代化的管理。不论是小型电商，还是大型企业，都需要进行仓库管理,这为此类软件的市场需求奠定基础。

另一方面,随着面向对象开发技术的突飞猛进，极大地保障了软件开发的效率和质量,软件开发过程趋于工程化,文档资料趋于标准化,使得软件开发的路线更加清晰,软件维护的成本也在降低,这为技术可行性奠定了基础。

而软硬件、云服务器的价格变得更加亲民,降低了开发运营一款软件的成本.并且软件开发行业也受到了国家的大力扶持和高度重视,相应的免税减税政策已经出台。这为经济可行性奠定了基础。

因此设计和开发一个进销存仓库管理系统已具备较好的可行性。

本次开发的<<基于Web的进销存仓库管理系统>>,其特点是围绕着库存物品的管理角度,制定库存进销存的业务逻辑,将库存物品相关的数据在数据库进行统一管理,完成信息源的集成.

通过本系统,用户得以减少了冗余的操作,从繁琐的仓储工作中解放，能够集中更多的精力去处理机器无法处理的业务,库存量的精度也将更为准确,能够提高库存周转速度,并且减少库存过程中造成的资金损耗.

## 1.3目前研究现状

管理信息系统(MIS)是由人和计算机设备或其他信息处理手段、组成并用于管理信息的系统。其管理的范围覆盖了信息的使用、储存、采集、加工、维护和传递6个方面.

MIS所管理的信息,是一项重要资源,信息决定着企业的管理层能否做出合理的决策.信息的有效管理是所有企业在管理时面临的巨大难题,在这个背景下, 能够对信息进行科学化的管理的MIS顺势迅猛发展,在重视信息的现代化社会中趋于普及化.

而进销存仓库管理系统也是MIS的一项分支,参考了当前同类的企业仓储管理系统后,总结得出目前主流的库存管理体系:仓库管理包括了采购管理,库存管理,统计管理,财务管理,盘点管理等模块.而在这些模块中,库存管理是整个仓库管理中的重中之重.

库存管理大致有以下几项功能需求:

1. 要求能够系统根据库存商品的类别、主要销售地点等因素,将库存商品存储在不同的仓库中来进行库存的规划,如采购、入库、验收等
2. 各地的仓库之间会有库存的流动,需要根据各地仓库的需求来调拨库存,
3. 定期进行库存盘点、统计分析,并生成相应的报表.
4. 规划合理的库存数量以及库存费用,使得企业在库存管理中达到最佳的状态.
5. 加强对库存物资的管控,定期统计出入库情况,根据出入库记录指定相应的库存定量化管理,物资的储备保持充足.

## 1.4开发目标

系统旨在提高仓库管理人员的工作效率,简化以往出入库的繁琐流程以及周期性长的弊端.

本系统可以很好地解决手工录入单据信息时,由于抄写错误等问题导致的数据不准确问题,也简化了查询步骤,使得工作人员无需通过手工来翻阅查找.

通过电子化达到提高生产效率.同时也能够减少库存管理中的漏洞,也能够节约在管理方面的开支,仓库的管理水平将得到显著提高.

针对用户需求将系统划分为多个功能模块,能让用户在相应的功能模块下，通过简单的操作就能管理仓库.

在系统的功能方面,系统的库存信息必须保持最新并且准确无误.

系统还需应具备良好的安全性,各项功能只有拥有对应权限的用户才能进行操作,除此之外,还需要保证数据的安全性.

# 2.系统需求

## 2.1角色分析

系统涉及到出入库管理员、库存管理员、系统维护管理员3种角色,如下表所示

表2.1.1系统角色表

|  |  |
| --- | --- |
| 角色 | 职责 |
| 出入库管理员 | 录入商品的出入库及退货信息 |
| 库存管理员 | 录入库存损耗信息，查看库存报表，维护系统业务数据 |
| 系统维护管理员 | 维护系统运行，用户及用户权限管理 |

## 2.2 系统功能及用户需求分析

本系统针对中小企业日常库存信息进行统一管理.系统将仓库的日常作业划分成进货、出货、退货、存货4项业务功能,以及数据维护、用户管理2项基础功能。

出入库管理员的主要工作是进货、出货、退货的管理.

通过”进货管理”录入商品的进货信息,进货信息包括商品、供应商、进货价格、进货时间等.也可以对进货记录进行退货操作.需要填写退货的数量与价格,其中退货总数不得超过进货数量.用户操作完成后,对应的商品库存将自动更新.

通过”出货管理”页面录入商品的出货信息,出货信息包括商品、顾客、出货价格、出货时间等.也可以对出货记录进行退货操作.需要填写退货的数量与价格,其中退货总数不得超过出货数量.用户操作完成后,对应的商品库存将自动更新.

库存管理员的主要工作是存货以及业务数据的维护.

通过”供应商管理”页面录入名称、地址、联系方式、开户银行等供应商信息,为商品入库提供供应商信息.

通过”顾客管理”页面录入名称、联系方式、付款方式等顾客信息,为商品出库提供顾客信息.

通过”商品管理”页面录入商品名称、供应商、库存、价格、图片等商品信息,为商品出入库提供商品信息.也可以录入库存损耗信息,商品库存将会自动更新.点击”统计分析”按钮将弹出每个月的库存统计报表,包括出库、入库、退货、损耗信息.

系统维护管理员的主要工作是用户管理以及系统数据的维护.

“菜单管理”页面列出了系统所有菜单页面以及页面对应的权限信息,通过添加、修改按钮可以动态地为系统增加菜单页面.

通过“角色管理”页面录入角色信息,系统目前有库存管理员, 出入库管理员, 系统维护员三个角色,每个角色下有着不同的权限,在进行操作时将校验用户是否具有该权限,没有则拒绝操作.

通过”部门管理”录入名称、上级部门等部门信息,为用户所属部门提供部门数据.

通过”用户管理”录入姓名、所属部门、联系方式等用户信息,点击”分配角色”按钮可以授予用户该角色下的所有权限.

在”系统公告”页面录入公告信息,公告将展示在主页面的控制台中.

## 2.3软、硬件需求

系统运行的硬件配置

CPU 2核Intel Xeon E5-2682 v4 及以上

内存 4G DDR4内存及以上

硬盘 40G及以上

系统运行的软件配置

Jre1.8及以上

Redis3.0及以上

Mysql5.7及以上

## 2.4开发语言、开发工具需求

采用Java语言进行开发,使用SpringBoot搭建框架,通过Swagger的API接口进行快速的接口测试.

使用[IntelliJ IDEA](https://www.jetbrains.com/idea/" \t "https://www.baidu.com/_blank)和VsCode完成系统前后端代码的编写.

使用Xshell用于与服务器的远程连接,Navicat用于对数据库的可视化操作,RedisPlus用于对Redis数据的可视化操作.

## 2.5非功能性需求

|  |  |
| --- | --- |
| 需求名称 | 详细要求 |
| 易用性 | 页面简洁明了,易上手 |
| 安全性 | 不同用户具备不同权限，对敏感资料只有经过授权才能访问，并且对非特权用户只能只读访问，特权用户才能修改其中的内容。 |
| 兼容性 | 兼容Chrome浏览器 |
| 可扩充性 | 当系统达到瓶颈的时,能够在不修改代码的情况下提高系统的负载能力 |
| 界面需求 | 系统的登录、主界面风格保持一致，以绿白两种颜色作为主色调。 |
| 完整性 | 对数据进行有效性验证。保证数据的完整性和安全性.包括数据备份、恢复、日志管理等基本功能 |
| 可维护性 | 采用面向对象的分析来对系统功能进行建模.在系统构架上考虑注重可维护性 |

# 3.用例分析

将上述需求转化为系统用例图,用例图展示了系统功能中参与者、用例、边界3者之间的关系.从而更加方便地让人们查看系统中所拥有的功能.

而用例是参与者与系统的一次交互过程,是角色和角色完成的动作的有机结合，形象且直观的表达了项目需求。

## 3.1总用例分析

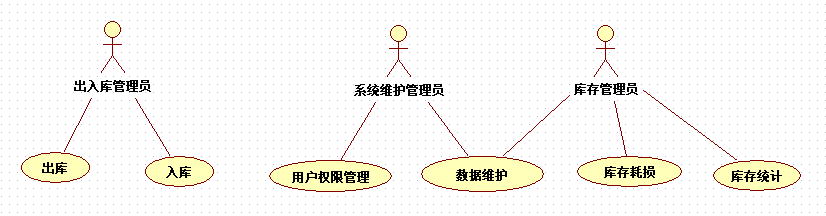


图3.1系统总用例图

依据第2章节的需求分析,本系统包括入库管理员、库存管理员、系统维护管理员3种角色.

将各个角色所需求的系统行为定义总结为系统总用例图,由出库,入库,数据维护,库存损耗,库存统计,用户权限管理6个用例组成.

以总用例图为基础,将角色间的公共行为分解为新用例,再将用例中的变更行为分解为用例之间的联系(拓展,包含,泛化,依赖等).最后编写用例的事件流进行用例描述.

以下展开对系统子用例的分析.

图中以带空心箭头的实线表示泛化关系,以虚箭头外加版型《extend》来表示用例之间的拓展关系

## 3.2数据维护用例分析

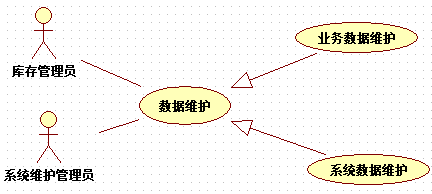


图3.2数据维护用例图

主要参与者为库存管理员和系统维护管理员.

库存管理员负责业务数据的维护,系统维护管理员负责系统数据的维护.

将数据维护用例根据用户的不同行为,细分为业务和系统数据维护2项子用例.

### 3.2.1业务数据维护用例分析



图3.3 业务数据维护用例图

业务数据维护用例与商品、供应商、客户管理之间存在泛化关系.

商品管理用例:商品数据的维护.包括商品信息的录入,商品图片的上传.

供应商管理用例:供应商数据的维护.包括供应商名称,地址等信息的录入.

客户管理用例:客户数据的维护.包括客户名称,联系方式等信息的录入.

表3.1商品管理用例规约

|  |  |  |
| --- | --- | --- |
| 用例名称 | | 商品管理用例 |
| 用例标识号 | | Goods\_01 |
| 参与者 | | 库存管理员 |
| 前置条件 | | 成功登陆本系统,被授予商品管理的权限 |
|  | 参与者动作 | 系统响应 |
| 基本事件流 | 上传商品图片 | 对合法图片进行压缩上传,并刷新页面中的缩略图 |
| 输入商品信息,点击添加按钮 | 录入商品信息至商品表  刷新商品列表网页  提示用户添加成功 |
| 异常事件流 | 商品价格不正确(3位小数,负数等) | 提示价格格式非法  要求用户重新输入 |
| 未输入必填项 | 提示必填项未输入不能提交表单 |
| 上传错误格式图片 | 拒绝上传  弹框提示用户格式非法 |

### 3.2.1系统数据维护用例分析

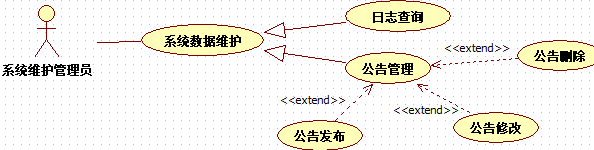


图3.4系统数据维护用例图

系统数据维护用例与日志查询、公告管理之间存在泛化关系.

日志查询用例:日志数据的搜索.

公告管理用例:公告的发布、修改、删除与公告管理用例间存在拓展关系,公告将展示在系统的主页面上.

表3.2发布公告用例规约

|  |  |  |
| --- | --- | --- |
| 用例名称 | | 发布公告用例 |
| 用例标识号 | | Notices\_01 |
| 参与者 | | 系统维护管理员 |
| 前置条件 | | 成功登陆本系统,被授予发布公告的权限 |
|  | 参与者动作 | 系统响应 |
| 基本事件流 | 发布公告 | 发布公告成功,生成公告记录插入数据库  弹窗提示用户发布成功,主页面显示公告内容 |
| 修改公告 | 修改公告成功,修改数据库公告记录  弹窗提示用户修改成功,主页面显示修改后的公告内容 |
| 删除公告 | 修改公告成功,修改数据库公告记录为不可用  弹窗提示用户删除成功,主页面不显示该公告 |

## 3.3用户权限管理分析

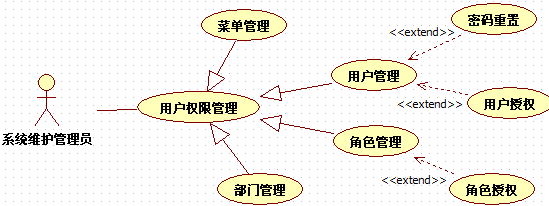


图3.3用户权限管理用例图

用户权限管理用例与菜单、部门、用户、角色管理之间存在泛化关系.

其中密码重置和用户授权与用户管理用例间存在拓展关系.

角色授权与角色管理间存在拓展关系.

表3.3用户管理用例规约

|  |  |  |
| --- | --- | --- |
| 用例名称 | | 用户管理用例 |
| 用例标识号 | | User\_01 |
| 参与者 | | 系统维护管理员 |
| 前置条件 | | 成功登陆本系统,被授予用户管理的权限 |
|  | 参与者动作 | 系统响应 |
| 基本事件流 | 修改用户头像 | 对合法图片进行压缩上传,并刷新页面中的缩略图 |
| 修改用户信息 | 修改用户信息至用户表  刷新用户列表网页  提示用户修改成功 |
| 可选事件流 | 密码重置 | 重置用户密码为系统预设值的默认密码  提示用户密码重置成功 |
| 用户授权 | 弹出授权弹窗  展示该用户下拥有的所有角色信息 |

## 3.4入库用例分析

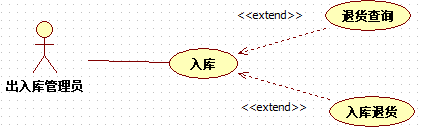


图3.4 入库用例图

出库与出库退货用例、入库与入库用例之间都存在拓展关系.

表3.4入库用例规约

|  |  |  |
| --- | --- | --- |
| 用例名称 | | 入库退货用例 |
| 用例标识号 | | Intport\_01 |
| 参与者 | | 出入库管理员 |
| 前置条件 | | 成功登陆本系统,被授予出入库管理的权限 |
|  | 参与者动作 | 系统响应 |
| 基本事件流 | 入库 | 对入库信息进行校验,入库数量不能为负数,入库价格为正两位小数.  对于非法入库信息,对用户进行错误弹窗  入库信息无误,则入库成功,生成入库单存入数据库 |
| 可选事件流 | 入库退货 | 对入库记录进行退货,退货数量不能大于入库数量,且不能为负数. |
| 退货查询 | 查询入库单下,某个商品的退货信息,包含退货数量,退货日期等. |

# 4.流程分析

## 4.1入库

由出入库管理员进行入货操作,在持有入库权限的情况下,打开商品进货页面,对入库单进行添加或修改操作.

a.点击添加按钮,新建一张进货单,点击后跳转至进货单页面.

新建进货单页面将根据雪花算法,被分配到一个全局唯一的订单号,一张进货单下可以存在多条商品入库记录.

b.点击修改按钮,修改一张进货单,点击后跳转至进货单页面.

修改进货单页面将被分配与原订单一致的订单号.

在进货单页面点击添加按钮,新增一条进货记录,进货记录包含选中的商品,入库数量,入库价格等信息.其中入库的数量为正整数,入库价格为正两位小数,商品为必选项.

填写入库信息后点击提交按钮,进入入库信息的验证,若存在非法输入时入库失败,弹框提示用户存在非法输入.所有信息填写无误时,入库记录添加成功,更新被对应商品的库存信息,最后弹窗提示用户添加入库记录成功.

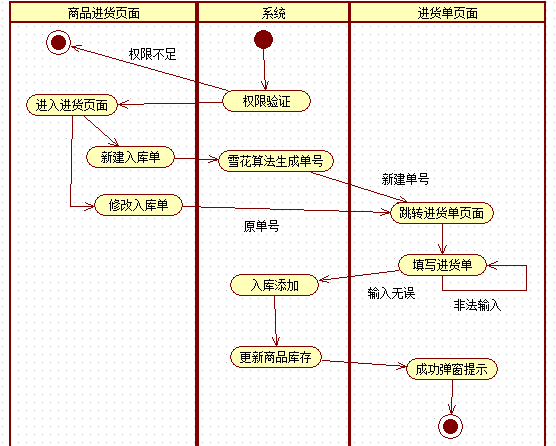


图4.1入库流程图

## 4.2单号生成

在出入库管理员参与入库用例时,会自动生成入库订单号,唯一标识用户本次的入库操作.单号的生成策略借助了雪花算法的思想.

雪花算法是由twitter开源的ID生成算法.由1位符号位,41位时间戳,10位工作机器ID和12位序列号组成,共计64位.相比mysql的自增主键,雪花算法生成的id不容易被爬取,有着更好的安全性.

雪花算法的另一个优点在于,此算法预留了10位的机器id用于自定义,为后续的横向扩容保留余地.10位的工作机器id由数据中心id和工作中心id组成.若后续拓展了数据中心的数量(多个数据库),只需要改变数据中心id;若后续拓展了工作中心的数量(多台服务器),只需要改变工作中心id,在不同服务器上的工作中心id不同,保证了不同服务器生成的ID全局唯一.

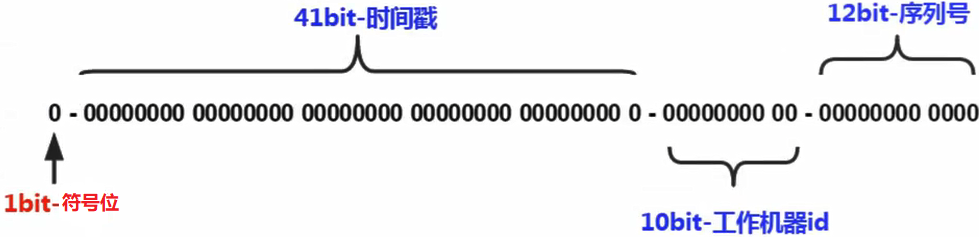


图4.2雪花算法图

由于一台机器上只需要一个ID生成器的实例,以避免生成的ID重复,可以使用懒汉单例模式来对雪花算法进行封装,避免实例被重复创建.

又因为java虚拟机对指令的优化,在多线程环境下容易引发指令重排(初始化对象和分配内存地址执行顺序不确定)而导致对象重复创建,单例模式被打破.可以使用synchronized同步方法体可以实现线程安全.

但同步方法体是对整个方法的约束,会引发性能问题.另一种思路是采用双重检查锁,即重复检查实例是否被创建,只在创建实例代码块进行同步,避免了同步带来的性能问题.

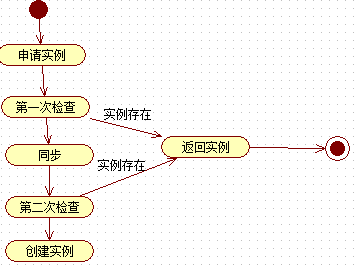


图4.3懒汉单例流程图

具体流程如下:

a.分配静态区内存空间

private volatile static SnowflakeIdWorker snow = null;

b.私有的构造方法,禁止外部创建实例,参数为工作中心id,数据中心id

private SnowflakeIdWorker(long workerId, long datacenterId) {

c.公开获取实例的方法

public static SnowflakeIdWorker getInstance() {

d.第一次检查,实例被创建后不进入方法体

if (snow == null) {

e.同步

synchronized (SnowflakeIdWorker.class) {

f.第二次检查,第一条实例创建了实例,则第二条同步代码块中的线程不再进入创建实例方法体,线程安全

if (snow == null) {

g.创建实例

snow = new SnowflakeIdWorker(workerId,datacenterId);}}}

h.返回实例

return snow;

## 4.3入库退货

由出入库管理员进行入库退货操作,在持有退货权限的情况下,进入进货单页面,对进货单下的入库记录进行退货操作.

点击退货按钮,弹出退货弹窗。输入对应的退货信息,其中退货的数量应小于等于入库单中,该商品对应的入库数量.否则退货失败,对用户进行错误输入提示.

当退货信息输入无误时，退货成功并生成退货记录。

根据退货数量,更新商品库存.

若退货后剩余库存大于预警值上限,则对库存管理员进行库存超额预警。

若退货后剩余库存小于预警值下限,则对库存管理员进行库存不足预警。

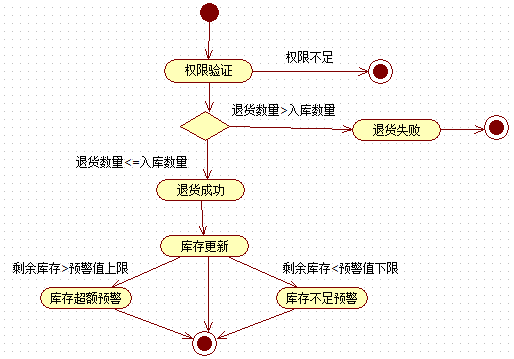


图4.4入库退货流程图

库存预警设计思路:

所有引起库存变动的用例(入库,入库退货,出库,出库退货,库存损耗),都调用了商品服务提供的”更新商品库存”接口,商品库存的改动在一定条件下将触发库存预警.

可见更新商品库存和库存预警是两个高耦合代码,而不同类型的商品库存变动将触发同样的库存预警,将造成库存预警代码冗余.

运用代理模式,可以对更新商品库存接口进行增强,将库存预警从中抽离,封装为库存预警切面织入接口.

使得更新商品库存接口的每次调用,都将自动地执行是否需要进行库存预警的业务.可以降低耦合性.

## 4.4商品添加

由库存管理员进行商品添加操作,在持有商品管理权限的情况下,打开商品管理页面,点击添加按钮,弹出商品添加弹窗。

商品图片的上传是可选项，若不上传图片，则保存默认图片,若上传图片,则对图片格式进行校验,正确格式的图片将被上传至图片存储服务器,根据服务器返回的访问地址来刷新当前商品的缩略图。

编辑商品的供应商,输入名称,描述,包装等基本信息,其中库存量，库存预警值只能输入正整数，商品价格只能输入正数且最大2位小数。在商品信息存在非法输入时，将弹框对用户进行提示。

商品添加成功后,录入商品记录至数据库,关闭商品添加弹窗并刷新商品列表

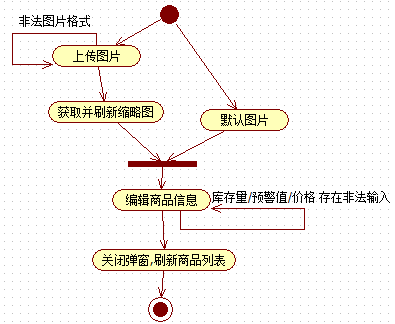


图4.5商品添加流程图

商品图片上传设计思路:

整合fastdfs文件系统,解决商品图片的上传问题.fastdfs由Tracker(注册中心)和Storage(存储节点)组成.

首先,Storage定时向Tracker发送心跳响应,注册节点.

在Client(客户端)请求上传时,首先访问Tracker,由Tracker返回当前可用的Storage地址.

Client向Storage发起真正的上传图片请求,Storage接受请求并将图片压缩后存入磁盘,返回给Client图片的访问路径.

商品图片访问设计思路:

Client根据访问路径发起访问请求,端口为Nginx端口,Nginx首先访问本地缓存中是否存在此图片,若不存在,则访问Storage并将图片内容返回给Client.

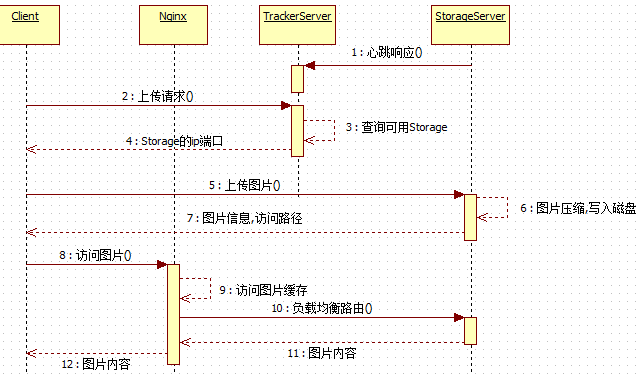


图4.6商品上传访问流程图

## 4.5权限分配

由系统维护管理员进行权限分配操作,在持有用户管理权限的情况下,打开角色管理页面,点击添加按钮,输入角色名称和备注,点击提交.

角色添加成功后,点击分配权限按钮,勾选该角色对应所需的权限数据,点击确认分配按钮完成权限分配.

打开用户管理页面,点击分配角色按钮,勾选该用户对应所需的角色数据,点击确认分配按钮完成授予角色.

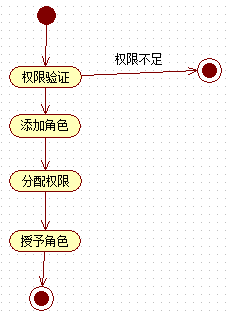


图4.7权限分配流程图

## 4.6授权流程

用户在操作前需要与Shiro的Subject(代理主体)进行交互,由Subject委托给内部的SecurityManager(安全管理器)进行授权操作.

SecurityManager将访问Realm(领域),以获取用户的权限数据.

realm被链式地存储在Realm Chain中,SecurityManager对Realm的轮循访问将采取短路机制,一旦有Realm接受处理并返回授权结果,就不再访问剩余的Realm.

SecurityManager接收并处理授权结果,决定用户是否有权限进行此操作,并将结果返回给Subject.

Subject接收SecurityManager返回的处理结果,若允许则用户操作继续执行,若拒绝则用户操作结束,弹窗提示用户权限不足.

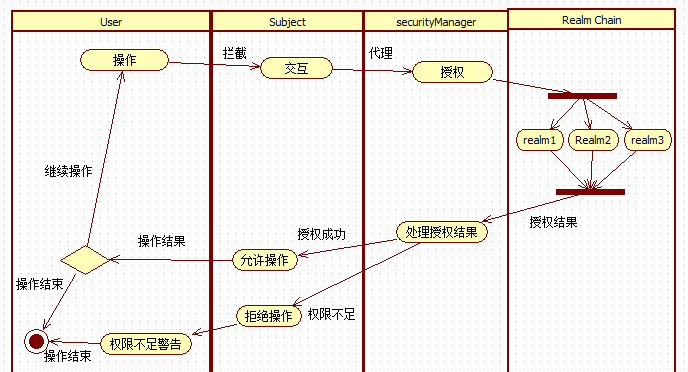


图4.8授权流程图

# 5.功能模块分析

系统包括出入库、库存、用户管理、数据维护4个功能模块

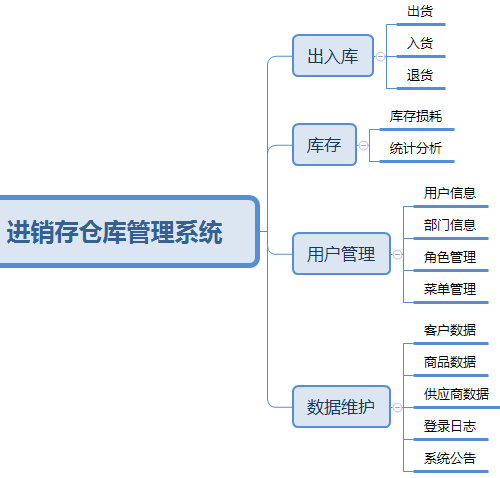


图5.1系统功能模块图

出入库模块:

1. 出货 商品的销售
2. 入货 商品的采购入库
3. 退货 商品的退货

库存模块:

1. 库存损耗 录入一段时间内,商品在库存中的损耗量,修正商品库存
2. 统计分析 对一段时间内的出货入货退货数据进行统计分析

用户管理:

1. 用户信息 维护用户的账号,拥有的角色,所在部门等信息
2. 部门信息 维护部门的基础信息,以及部门之间的层级关系
3. 角色信息 维护角色以及角色下所分配的权限信息
4. 菜单信息 维护系统左侧及顶部导航栏菜单及菜单下的权限信息

数据维护:

1. 客户数据 录入客户信息,包括客户姓名,联系方式等
2. 商品数据 录入商品信息,包括商品图片,商品名称等
3. 供应商数据 录入供应商信息,包括供应商姓名,地址等
4. 登录日志 记录系统的登录人,登录时间等信息
5. 系统公告 发布系统主界面的公告内容

# 6.架构设计

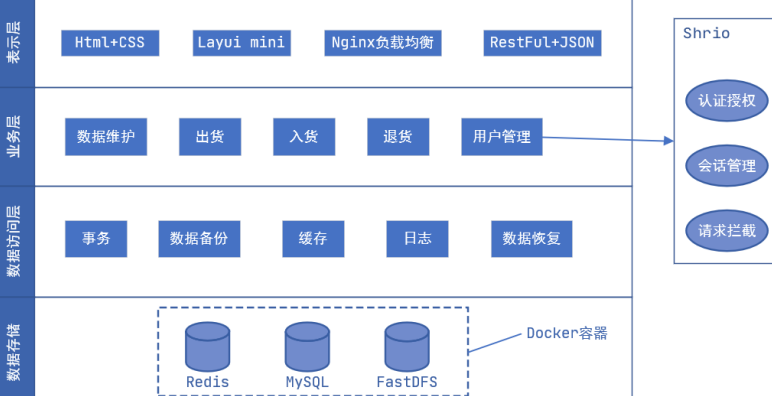


图6.2架构设计图

系统的架构设计如图所示,将系统自顶向下分为表示层,业务层,数据访问层,数据存储层，各层职责如下：

（1）表示层：系统的UI部分,主要负责页面数据的展示以及用户和系统之间的交互,通过html整合layuimini框架进行实现；

（2）业务层：整个系统的业务核心所在,业务层的设计与仓库管理的仓储业务密切相关,例如出入库业务等,所有数据都在业务层加以处理,处理完成后通过接口响应用户在表示层的请求,进行对应的业务处理,访问数据访问层的接口,完成对业务数据的操作,最后将结果返回给用户.

（3）数据访问层：主要负责访问数据库,完成对数据库的Select查询, Insert插入, Update更新, Delete删除操作,数据访问层对数据的操作是.

（4）数据存储层:针对不同数据的存储需求,将数据存储划分为磁盘存储和内存存储,需要持久化的数据存放在磁盘,无需持久化或需要高效访问的数据存储在内存.

本系统的分层架构具有以下优点

1. .扩充性高。每一层只和相邻的上下层进行交互，对于某一层的改变只会影响到它的上下层,并不会影响整个系统,而对于增加额外的一层,也只需要对上下两层进行修改.
2. .可重用。相邻两层之间约定接口进行访问,只要对相邻层提供一样的接口,每一层都可以有替代的实现方法.也就是说,得益于接口的抽象性,对于每一层接口的定义可以有不同的实现方法.
3. .适应抽象程度递增的系统设计.设计者可以将复杂系统依据设计需求,将步骤进行逐层分解,可以降低系统开发的难度.

# 参考文献

[1]殷俊芳，薛红，赵元凤.基于 B/S 模式的仓库管理系统设计[J].北京工商大学学

报,2006(01):27-30

[2] Jens von Berg,Joachim Schmidt,Thomas Wendler.Business process integration for distributed

applications in radiology.. IEEE Trans . 2001

[3]张璐. 库存管理系统设计及功能实现[J]. 数字技术与应用. 2010(04)

[4]盛华.仓库管理器一个库存的分析模式, 浙江物流, 2005, 1(5): 9-11.

[5]栋隆.仓库管理系统的关键 7 步[J].权威论坛.2005(07):103-105

[6] Morris.Collen 等.Hospital Information System Integration. . 2001

[7]胡宜男. 仓库管理的新技术与运用. 物流, 2006, (3): 80-81.

[8]刘冰,王新城.数据库系统概论[M].北京:高等教育出版社.2010

[9]储雪俭.仓库管理系统分析研究[J].物流技术.2006（06）：96-99

[10]吴轶.通用仓库管理系统的设计与实现[J].天津职业院校联合学报.2007,09(05)：36-38

[11]周红.基于 B/S 结构的仓库管理系统[J].电脑知识与技术.2007：602-603