智能分货SAAS系统

系统架构设计

|  |  |
| --- | --- |
| 文档编号 | gyxr-adam-sad-design |
| 当前版本 | V0.9 |
| 作者 | 谢志研 |
| 完成日期 | 2017年8月29日 |

# 一、需求分析

## . 1需求描述

商品价值的影响因素：

生命周期（上市日期、品类、厚薄）

销量

转化率

访客数量

收藏数

加购数

库存

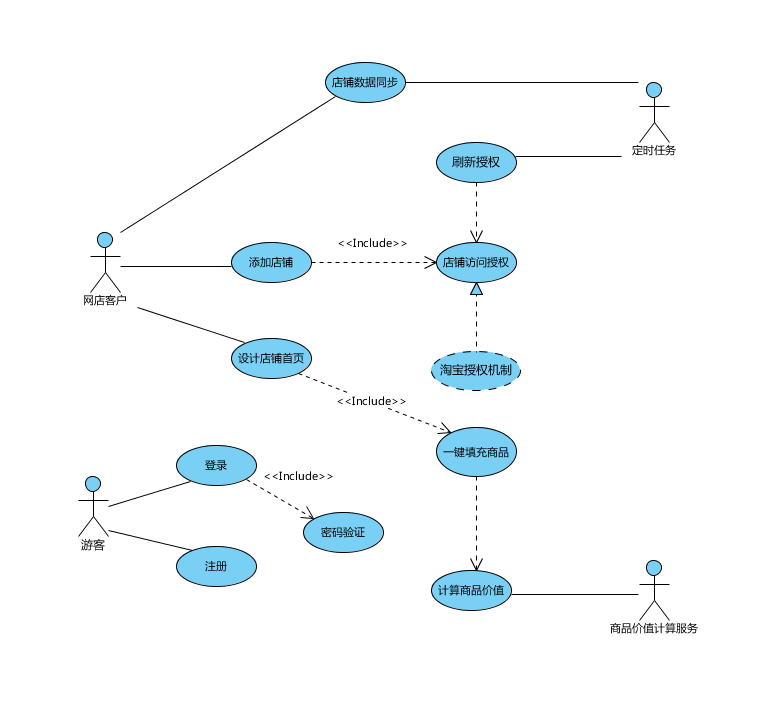
**用户需求**

1. 利用大数据、智能算法辅助优化店铺商品结构。一方面减少人工参与，降低运营成本；另一方面通过算法优化商品结构，提高商品指标，给用户带来经济效益。
2. 支持店铺商品摆放可视化设计（给用户参考）。
3. 支持多用户，用户可以在平台上自主注册。
4. 每个用户允许添加多个店铺。（前期只允许用户创建一个店铺，后期可以根据店铺数量进行收费，也是一种可以考虑的商业模式）
5. 与电商平台数据同步（店铺信息、商品、商品销售数据等），简化用户操作。（需要电商平台接口支持，初期支持淘宝，后期可以扩展到其他电商平台）
6. \*页面、模块、商品通过标签匹配

**其他需求**

1. 和其他子系统（模块）集成
2. 系统方便演化
3. 后台管理

## 1.2 用例图



## 1.3 用例描述

### 1.3.1 新用户注册

在系统中注册个人信息，用来识别用户身份。后期演化的广告、活动、收费等业务模式也会根用户账户进行绑定。

### 1.3.2 用户登录系统

游客提供用户名和密码，系统验证通过后进入系统。

### 1.3.3 添加店铺

用户在系统中添加店铺，对应电商平台上的店铺，在进行店铺首页设计之前需要先添加店铺。一种可能的场景是：

1. 用户点击添加店铺按钮，系统跳转到淘宝授权页面

2. 用户输入用户名，密码进行验证，验证成功之后跳转到回系统

3. 系统后台根据授权成功后返回的access\_token获取淘宝店铺信息

4. 保存店铺信息和授权信息到系统数据库

### 1.3.4 店铺访问授权

用户在添加店铺的时候需要登录电商平台（淘宝）进行验证授权。授权成功之后，系统根据取得的授权信息访问淘宝账号数据，同步到系统中。

### 1.3.5 刷新授权

由于系统授权存在有效期，淘宝一般是72个小，所以需要定期刷新授权保证授权信息有效。系统会在后台定时刷新。

### 1.3.6 设计店铺首页

用户可以在系统中对店铺首页进行可视化设计。设计的重点商品与数据相关性，利用数据来对商品摆放结构优化。用户在进行店铺首页时，利用系统提供的一键填充商品功能，简化用户对于商品摆放结构的选择。用户也可以手动调整商品摆放摆放结构。

### 1.3.7 一键填充商品

客户端根据商品价值获取排序后的商品，填充到页面中。

### 1.3.8 计算商品价值

系统利用大量数据和我们总结的算法，计算每个商品的价值，并且能够根据商品价值对商品进行排序。商品价值由算法定义。

## 1.4 流程

## 1.4.1 用户注册流程

## 

## 1.4.2 店铺首页设计流程

## 

## 1.5 功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 功能名称 | 功能描述 | 优先级 |
| F001 | 用户注册 |  | 高 |
| F002 | 用户登录 |  | 高 |
| F003 | 添加店铺 | 需要跳转到淘宝网授权 | 高 |
| F004 | 店铺列表 |  | 中 |
| F005 | 删除店铺 |  | 低 |
| F006 | 店铺数据定时自动同步 |  | 中 |
| F007 | 店铺数据手动同步 |  | 中 |
| F008 | 店铺首页设计 | 可视化设计用户店铺首页，支持一键填充商品，也支持手动填充商品。 | 高 |
| F009 | 一键填充商品 | 利用商品价值对商品排序，获取指定数量的商品 | 中 |
| F010 | 计算商品价值 | 利用数据和智能算法计算商品的价值，能够能够商品价值对商品进行排序 | 中 |

## 1.6 领域模型

# 

# 二、系统设计

## 2.1 系统逻辑架构

UI 层

API层

业务逻辑层

基础设施层

## 2.2 子系统划分

## 

## 2.3 开发设计

1. 技术选型

|  |  |
| --- | --- |
| 后台开发语言 | Java 8 |
| 后台框架 | Spring Boot |
| 前端开发语言 | Javascript, HTML, CSS |
| 前端框架 | AngularJS |
| 数据库 | MySql |
| 搜索引擎 | ElasticSearch |
| 服务器 | 阿里云ECS |

系统采用前后端分离的B/S架构，前端整体用angularJS框架，样式用bootstrap3。 后台采用Java Spring Boot框架开发。

系统后端服务API采用restful风格, 提高系统的性能、吞吐量和可伸缩性。

## 2.4 逻辑数据库设计

### 

## 2.4 物理数据库设计

* 根据领域模型自动生成物理数据库Schema。（Spring Data JPA）
* 集成liquibase简化数据库变更管理。
* 系统支持多数据库，避免使用函数、触发器、存储过程和视图。
* 数据库管理系统用mysql， 开发时可以用h2等内存数据库。

## 2.5 安全性设计

### 2.5.1 认证和授权

用户认证采用用户名/密码认证和JWT Token两种认证方式。用户登录时进行用户名、密码认证，认证完成之后返回JWT Token，后续接口调用用JWT Token认证。

### 2.5.2 数据安全

在正式环境下采用https协议，提高系统安全性，同时也避免电信运行商广告植入对系统造成影响。

### 2.5.3 用户淘宝账号安全

系统不会保存用户淘宝账号密码，系统只会保存淘宝账号的授权信息，用来进行数据同步。因此用户不需要担心淘宝账号密码泄漏。

## 2.6 外部接口

淘宝用户授权

淘宝开放API

## 2.7 系统部署设计

## 

## 三、详细设计

## 2.1 添加店铺交互过程

### 商品价值的影响因素：

### 生命周期（上市日期、品类、厚薄）

### 销量

### 转化率

### 访客数量

### 收藏数

### 加购数

### 库存

## 

## 2.2 数据同步设计

## 