**自然堂产品可视化设计文档**

目录

[1 引言 1](#_Toc511683505)

[1.1 概述 1](#_Toc511683506)

[1.2 编写目的 2](#_Toc511683507)

[1.3 术语和缩略语 2](#_Toc511683508)

[1.4 参考资料 2](#_Toc511683509)

[1.5 开发环境 2](#_Toc511683510)

[2 设计 2](#_Toc511683511)

[2.1 系统描述 2](#_Toc511683512)

[2.1.1 需求描述 2](#_Toc511683513)

[2.1.2 需求分析 2](#_Toc511683514)

[2.1.3 系统结构 3](#_Toc511683515)

[2.2 总体设计 3](#_Toc511683516)

[2.2.1 设计目标 3](#_Toc511683517)

[2.2.2 处理流程 4](#_Toc511683518)

[2.2.3 时序图 5](#_Toc511683519)

[2.2.4 接口设计 7](#_Toc511683520)

[3 数据库结构 9](#_Toc511683521)

# 引言

## 概述

为扫描自然堂产品花盒二维码的消费者提供更加赏心悦目的可视化、可声化产品说明，并引导消费者关注品牌微信，提升消费者与品牌的互动与粘性。

## 编写目的

本文档编写目的是说明程序模块设计考虑，包括需求描述、程序流程、数据表设计等，为系统的后期维护提供基础，留待系统设计人员、系统开发人员、项目维护人员和项目测试人员传阅。

## 术语和缩略语

Spring Boot、Mysql、MyBatis、Tomcat、Sl4j、html、jquery、ajax

## 参考资料

防伪查询接口说明.docx

自然堂+第一阶段产品可视化项目执行方案（技术）ver1.pptx

自然堂可视化可声化合同技术附件.doc

阿里巴巴Java开发手册v1.1.0.pdf

## 开发环境

Eclipse集成环境、windows10系统、viso制图、word文档、centos服务器

# 设计

## 系统描述

### 需求描述

前台：基于手机网页、微信公众号等，实现对消费者产品的可声化、可视化的页面展示，并对接防伪码查询系统，提供消费者防伪查询。

后台：实现产品可声化、可视化的后台页面编辑与预览。

接口：实现与《防伪码查询系统》的对接

### 需求分析

根据需求整理，将项目划分如下四个部分进行开发：

1. 对接自然堂产品信息查询接口和产品防伪校验接口。
2. 增加后台管理功能，实现对产品的动态发布，上新、下架、编辑功能。
3. 后台管理主界面实现产品的分页查询和模糊查询功能。
4. 对于第三方接口调用异常情况，进行邮件通知。

### 系统结构

图1、系统结构图

## 总体设计

### 设计目标

#### 安全性

对外公开指定的产品查询接口和防伪验证接口。

后台管理加入身份校验功能，未登录禁止访问后台页面。

#### 性能设计

为实现快速响应，采用mysql数据库，后期数据量不断增大可采用分表分区方案进行扩展。

当并发量大的时候，可采用nginx分流，动态扩展服务进行多套部署，缓解服务压力。

#### 技术结构

前端采用html+css+jquery+ajax技术相结合。

后台采用spring boot + mybatis框架进行开发，即SSM框架结构。

服务器采用tomcat，数据库采用Mysql数据库。

### 处理流程

#### 校验流程：

1. 客户端通过扫码携带产品码请求后台，后台处理请求，将信息加密，调用自然堂接口查询产品名称信息，然后携带产品名称参数访问本地数据库进行商品信息查询，并返回给客户端。
2. 客户端进行产品校验，前端传递商品码给后台服务，后台携带商品防伪信息请求自然堂防伪校验接口进行校验并给客户端返回相应结果。

#### 管理流程

1. 后台登录后，首页分页展示所有产品总体信息。左上角支持模糊查询。
2. 点击产品上新，管理员选择上传产品相关图片、文字、视频、音频等信息，后台进行持久化处理。
3. 点击产品编辑，后台对数据进行查询展示，管理员选择修改相关产品信息，提交后台，服务进行信息更新持久化。
4. 点击产品删除，后台产品清除该产品所有信息，不可恢复。

### 时序图



图2、产品查询和防伪校验时序图



图3、后台管理系统时序图

### 接口设计

#### 外部接口

/\*\*

\* 查询商品信息接口

\*

\* @return

\*/

@RequestMapping(value = "/product\_info")

public ProcessResult getProductInfo(String param)

/\*\*

\* 校验商品是否为正品接口

\*

\* @return

\*/

@RequestMapping(value = "/auth\_product")

public ProcessResult authProductInfo(HttpServletRequest request, String code)

#### 内部接口

/\*\*

\* 发送邮件接口

\*

\* @param from

\* 发送者

\* @param to

\* 接受者

\* @param subject

\* 主题

\* @param content

\* 内容

\*/

void sendMail(String from, String to, String subject, String content);

/\*\*

\* 查询商品信息业务逻辑

\*

\* @param productName

\* 产品名称

\* @return

\*/

ProductInfo queryProductInfo(String productName);

/\*\*

\* 存储产品信息

\*

\* @param productInfo

\* 产品信息

\* @return

\*/

int saveProductInfo(ProductInfo productInfo);

/\*\*

\*

\* 根据名称删除记录

\*

\* @param productName

\* 记录名称

\* @param path

\* 信息路径

\* @return

\*/

int deleteProductInfoByName(String productName, String path);

/\*\*

\* 查询所有产品

\*

\* @return

\*/

PageInfo<ProductInfo> listAllProductInfo(int startPage, int pageSize);

/\*\*

\* 模糊查询产品信息，没有的话返回默认值

\*

\* @param productName

\* 产品名称

\* @param seriesName

\* 系列名称

\* @param startPage

\* 起始页码

\* @param pageSize

\* 每页数目

\* @return

\*/

PageInfo<ProductInfo> listProductInfo(String productName, String seriesName, int startPage, int pageSize);

/\*\*

\* 查询系列数目

\*

\* @return

\*/

List<String> listSeriesInfo();

/\*\*

\* 更新产品信息

\*

\* @param productInfo

\* 更改后的产品信息

\* @param oldName

\* 旧名称

\* @return

\*/

int updateProductInfo(ProductInfo productInfo, String oldName);

/\*\*

\* 根据名称批量删除产品

\*

\* @param nameList

\* 名称列表

\* @return

\*/

int delteProductList(List nameList);

# 数据库结构

表1商品信息表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| Id | Bigint | 每条记录的随机唯一id |
| Product\_name | Varchar | 产品名称 |
| Pruduct\_series | Varchar | 产品系列 |
| Product\_pic | Varchar | 产品图片 |
| Product\_desc | Varchar | 产品描述 |
| Audio\_url | Varchar | 音频地址 |
| Video\_url | Varchar | 视频地址 |
| Video\_pic | Vahchar | 视频封面图片 |
| Code\_pic | Varcahr | 防伪图片 |
| TemplateId | Int | 模板id |

表2管理员信息表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| User\_name | Varchar | 用户名 |
| Password | Varchar | 密码 |