License管理系统设计文档

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订人** | **修订日期** | **修订描述** |
| v0.1 | 邓恺 | 20190307 | 初稿 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目 录

[1. 系统背景 1](#_Toc331595690)

[2. 系统目的 1](#_Toc339547459)

[3. 系统功能 1](#_Toc942867981)

[3.1. 注册管理 1](#_Toc1817194949)

[3.2. 系统管理 1](#_Toc269995928)

[4. 系统结构 1](#_Toc555579030)

[5. 主要数据结构 1](#_Toc1135394105)

[5.1. 序列号 1](#_Toc55834780)

[5.2. 产品 2](#_Toc242578317)

[5.3. 注册码 2](#_Toc1116780566)

[6. 数据字典 2](#_Toc1131996382)

[6.1. 字典规范 2](#_Toc442725709)

[6.2. 基本配置 3](#_Toc1485476527)

[6.2.1. error（错误信息对照） 3](#_Toc1160808262)

[6.3. 系统配置 3](#_Toc384155859)

[6.3.1. parameter（系统配置） 3](#_Toc1880055231)

[6.4. 日志管理 3](#_Toc1319732846)

[7. 接口设计 3](#_Toc144850284)

[7.1. 设计要求 3](#_Toc1843411729)

[7.2. 接口示例 4](#_Toc382366683)

[7.3. 接口定义 4](#_Toc861264898)

[7.3.1. 返回值结构 4](#_Toc1174692453)

[8. 性能要求 4](#_Toc1650997118)

[9. 兼容性要求 4](#_Toc496694745)

[10. 易用性要求 5](#_Toc819862517)

[11. 安全性要求 5](#_Toc1424568032)

[12. 国际化 5](#_Toc1150378000)

# 系统背景

# 系统目的

License系统用于公司内部，管理登记产品的注册码。

# 系统功能

## 注册管理

* 产品管理：公司产品线管理，每款产品可设置序列号前缀
* 注册码管理：根据产品序列号、客户名称、注册类型（正式/试用）、注册日期、有效期生成客户产品的注册码；注册码过期后，可根据登记信息重新生成新注册码；提供注册文件下载，发送给客户自行注册。

## 系统管理

* 用户管理：增删系统用户；修改密码
* 数据备份：自动/手动备份；数据恢复
* 操作日志：包括用户登录在内的系统操作日志。

# 系统结构

前端：VUE

后端：Flask

数据库：mysql

# 主要数据结构

## 序列号

产品序列号用于唯一标识产品，使用字母缩写+数字形式，长度为10字节。现有产品缩写如下：

DC——数据中心可视化

RB——ROBO

如果是测试产品，应在产品缩写后增加TEST，序列号总长不变。

DC00000001：数据中心可视化的正式产品00000001号

RBTEST0001：ROBO测试产品0001号

## 产品

——产品名称

——产品序列号前缀

——当前最大序号：初始值为1，增加一个注册码应自动加1

—— 备注

## 注册码

生成注册码应提供的信息包括：

——产品序列号：应根据产品类型和注册类型自动生成

——客户名称

——注册类型：正式/试用

——注册日期：注册码有效的起始日期，默认为生成当天的日期，可修改

——有效期：默认60天

——产品独有属性

DC：最大机柜数

# 数据字典

## 字典规范

1. 所有名称全部使用小写字符；
2. 名称尽量使用完整英文单词，超过8个字符后进行缩写（应符合一般缩写规范，方便理解）；
3. 多个单词间使用下划线“\_”进行连接；
4. 使用组合时，多个相近名称应保持主语在前，如：开始时间（time\_ini）和结束时间（time\_fin），服务器地址（ip\_server）和客户端地址（ip\_client）；
5. 字符串长度应使用2的整数倍，16、32、64、128；
6. 默认数值单位：长度为厘米（cm），重量为千克（kg），时间为秒（sec），功率为瓦（w）；
7. 部分通用名称定义：

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **缩写** | **类型** | **含义** |
| id |  | int | 编号，自增主键 |

## 基本配置

### error（错误信息对照）

|  |  |  |
| --- | --- | --- |
| **列名** | **类型** | **说明** |
| code | int | 主键，错误代码 |
| strcode | char(32) | 错误消息缩写 |
| message | char(255) | 错误消息 |

注：strcode和message通过后台服务收集，code在产品发布后确定

## 系统配置

### parameter（系统配置）

|  |  |  |
| --- | --- | --- |
| 列名 | 类型 | 说明 |
| id | int | 主键 |
| object | char(16) | 对象 |
| attrib | char(64) | 属性 |
| vtype | char(16) | 值类型(python)，如：int, str, list |
| value | char(64) | 值 |

## 日志管理

# 接口设计

## 设计要求

所有接口遵循以下设计要求：

1. URI根地址：/v1/
2. 资源访问请求包括CURD操作（Create/Update/Read/Delete），对应的HTTP方法如下：

GET —— 查询资源（Read）

POST —— 创建资源（Create）

PUT —— 修改资源（Update）

DELETE —— 删除资源（Delete）

1. 路径形式：URL根地址+/一级菜单名/功能名/资源名称
2. 所有参数（除url地址中的参数）和返回值严格遵守json格式规范

## 接口示例

1. 新增用户：POST /v1/users
2. 查询用户：GET /v1/users/<id>
3. 修改用户：PUT /v1/users/<id>
4. 删除用户：DELETE /v1/users/<id>
5. 搜索用户：GET /v1/users?name=<name>

## 接口定义

### 返回值结构

* 正常

{“code”：0，“strcode”：“succ”，“message”：“succ”，”data”：{......}}

* 错误

{“code”：-1，“strcode”：“succ”，“message”：“wrong password”}

code —— 返回结果错误编号，0为成功，其他为错误

data —— 结构由具体接口决定，没有返回数据时无此参数

以下所有接口的返回值均按照上述结构。

# 性能要求

# 兼容性要求

浏览器兼容：IE10，Firefox 56，Chrome 60及以上版本。

# 易用性要求

# 安全性要求

# 国际化

页面支持多语言版本。