软件序列号认证系统设计

# 概述

首先，认证组件的目的是为了防止用户在未得到认证的情况下使用软件，以及阻止用户在认证时间到期后，在未取得新认证的情况下继续使用软件。

其次，我们需要一套认证文件的管理系统，利用该系统，我们可以对这些认证文件进行管理。管理的主要内容包括：对序列号的生成，加密，二次加密，存储。查看序列号的分发情况，跟踪序列号的到期时间以及客户信息，已达到在软件限定时间到期后，提醒客户继续购买使用许可的目的。

最后，考虑到客户软件使用场景的复杂性，序列号验证组件内置在软件客户系统当中，以方便用户在没有网络连接的情况下，也可以使用验证系统进行验证。

# 版本管理

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 修改人 | 修改日期 | 描述 | 审批人 | 审批日期 |
| V1.0 | 刘盾 | 2016-06-06 | 创建 |  |  |
| V2.0 | 刘盾 | 2016-06-12 | 修改：  ①序列号验证系统中添加机器ID验证机制  ②license\_hospital表单中的ID字段删除，设hospital\_number字段为主键  ③改license\_detail表单中的hospital\_number字段为不可空 |  |  |

# 功能设计

系统整体工作流程图：



## 3.1 序列号管理系统（web项目）

### 3.1.1 用户管理系统

用户管理系统的主要作用是限制登陆该序列号管理系统的用户。系统会创建一个默认的admin管理用户，该用户可以凭借登陆密码登陆该系统，并使用用户创建系统添加新的用户。



### 序列号生成和加密模块

1.序列号的生成：

序列号文件为一行字符串，字符串中包含了序列号生成日期，软件许可使用日期，UUUID等其他一些信息合并而构成。

2.序列号的加密

在对原始序列号经过不可逆加密算法加密后(例如MD5算法)，形成一个新的字符串A，在进行进一步的二次混淆算法加密以后，连同原始序列号一同存储在数据库中。



### 3.1.3 序列号管理模块

序列号的管理在web网站上实现：

1.序列号生成

2.序列号的加密

3.原始序列号，加密序列号，生成日期等其他相关信息的储存

4.序列号信息查询

5.序列号信息的导出(脚本)



## 3.2 客户端软件序列号验证系统（接口）

### 3.2.1 接口描述:

1.该接口内置在软件系统的内部，能够被软件系统的其他模块调用；

2.该接口可以访问内置在软件系统内部的数据库

### 3.2.2 接口的主要功能

1.接口初始化的过程中会读取数据库中原始序列号字段的信息，如果获取不到则要求用户输入

2.接口接收的参数信息为用户输入的序列号，系统当前时间。

3.读取到序列号以后会进行同章节3.1.3相同的加密处理，然后与数据库中事先存入的加密后序列号进行比对。

4.序列号匹配成功之后，接口会对用户输入的序列号信息进行分析，获取序列号中设定的软件准用时间，并与系统的当前时间进行比对，如果发现系统时间在限定时间之内，则整个验证过程结束，返回给调用接口的模块一个验证通过的信息。

5.两次验证都成功后，接口将该原始序列号写入数据库。

6.软件再次启动后，如果接口可以从数据库当中获取到序列号，接口不会再次要求用户输入序列号。但是考虑到用户有可能会进入数据库修改序列号信息，则应该将序列号从数据库当中读出，重复4.5.6三个过程，如果在第4步失败，说明序列号被更改，此时应该要求用户重新输入正确的序列号，如果在第5步失败，则说明该序列号已经超期，此时应该要求用户购买新的序列号进行输入。

7.如果序列号使用时间到期，则清空序列号验证表单的内容，用户需要重新购买序列号。

8．软件启动时会从数据库中读取机器的ID号码，如果不存在，则读取机器ID号码并加密写进数据库，每次启动软件时都会读取机器的ID号进行加密后与数据库中已经存在的加密机器ID进行比对，如果匹配失败，说明用户更改了机器，此时应该停止软件并向客户端发送警告信息。

### 3.2.3 客户端序列号验证逻辑流程图：



# 4. 数据表设计

## 4.1 序列号信息表

Table：license\_detail

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段描述 | 字段名称 | 类型 | 长度 | 可空 | 说明 | 默认值 |
| ID | serial\_number\_id | integer | 100 | N | PK,自增 |  |
| 原始序列号 | source\_number | varchar | 255 | N |  |  |
| 生成日期 | create\_day | varchar | 255 | N |  |  |
| 到期日期 | expired\_date | varchar | 255 | N |  |  |
| 加密序列号 | encrypted\_number | varchar | 255 | N |  |  |
| 到期标识 | expired\_flag | integer | 10 | N | 0：未到期1：到期 | 0 |
| 有效天数 | valid\_days | integer | 100 | Y |  |  |
| 医院编号 | hospital\_number | integer | 100 | N |  |  |

## 4.2 序列号验证表

Table:client\_verify

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段描述 | 字段名称 | 类型 | 长度 | 可空 | 说明 | 默认值 |
| 原始序列号 | source\_number | varchar | 255 | Y |  | NULL |
| 加密序列号 | encrypted\_number | varchar | 255 | N |  |  |
| 机器ID | Machine\_id | varchar | 255 | Y |  | NULL |
| 使用时长 |  |  |  |  |  |  |

## 4.3 序列号管理系统用户表

Table:license\_user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段描述 | 字段名称 | 类型 | 长度 | 可空 | 说明 | 默认值 |
| 用户id | user\_id | integer | 100 | N | PK，自增 |  |
| 登录名 | username | varchar | 255 | N | 用于用户登陆，唯一 |  |
| 密码 | password | varchar | 255 | N |  |  |

## 4.4、医院信息表(通过医院编号字段与序列号信息表进行关联)

Table:license\_hospital

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段描述 | 字段名称 | 类型 | 长度 | 可空 | 说明 | 默认值 |
| 医院编号 | hospital\_number | integer | 100 | N | PK，自增 |  |
| 医院名称 | hospital\_name | varchar | 255 | N |  |  |
| 医院联系方式 | hospital\_phone | varchar | 255 | Y |  |  |
| 医院地址 | hospital\_address | varchar | 255 | Y |  |  |

# 5. 项目架构

1.项目基与spring+springMVC+mybatis框架进行开发，采用maven构建项目，采用SVN进行版本控制；

2.数据库采用mysql数据库