# 敏感词分析系统 V1.0

# 设

# 计

# 说

# 明

# 书

目 录

[敏感词分析系统 V1.0 1](#_Toc87352955)

[一、系统简介 1](#_Toc87352961)

[二、系统开发环境 1](#_Toc87352962)

[三、系统运行环境 1](#_Toc87352963)

[3.1硬件环境 1](#_Toc87352964)

[3.2软件环境 1](#_Toc87352965)

[四、系统功能介绍 2](#_Toc87352966)

[4.1系统总体结构设计 2](#_Toc87352967)

[4.2软件系统 2](#_Toc87352968)

[4.3 主界面 2](#_Toc87352969)

[4.4系统管理 3](#_Toc87352971)

[4.4.1用户管理 3](#_Toc87352972)

[4.4.2 角色管理 4](#_Toc87352974)

[4.4.3 权限管理 4](#_Toc87352977)

[4.5数据管理 5](#_Toc87352980)

[4.5.1 文章分析 5](#_Toc87352981)

[4.5.3 公告管理 6](#_Toc87352983)

[4.5.4 敏感词管理 7](#_Toc87352986)

[4.5.4 词云分析 7](#_Toc87352989)

[4．6系统监控 8](#_Toc87352992)

[4.6.1 在线用户 8](#_Toc87352993)

[4.6.2 行为日志 9](#_Toc87352995)

[4.6.3 环境监控 9](#_Toc87352998)

[4.7 登录 10](#_Toc87353002)

[4.7.1 个人账号 11](#_Toc87353003)

[4.8 系统风格 12](#_Toc87353004)

[五、 系统整体运行原理 12](#_Toc87353005)

[5.1 数据显示与管理 12](#_Toc87353006)

[5.2 界面设计需求 13](#_Toc87353007)

[5.3 确定系统实现技术 13](#_Toc87353008)

[六、系统测试 13](#_Toc87353009)

[6.1功能测试 13](#_Toc87353010)

[6.1.2表单测试 14](#_Toc87353011)

[6.2性能测试（包括负载/压力测试） 15](#_Toc87353012)

[6.3用户界面测试/可用性测试 16](#_Toc87353013)

[6.4系统兼容性测试 16](#_Toc87353014)

[6.5安全性测试 17](#_Toc87353015)

[七、系统出错处理设计 17](#_Toc87353016)

[7.1出错信息 17](#_Toc87353017)

[7.2补救措施 18](#_Toc87353018)

[八、 软件维护设计 19](#_Toc87353019)

[8.1影响维护工作的因素 19](#_Toc87353020)

[8.2软件维护的策略 19](#_Toc87353021)

# 一、系统简介

随着互联网的基础设施的逐渐完善，我们可以便捷地使用软件来处理，分析一些文章的敏感词汇，通过已收集的敏感词库与指定的文章进行匹配，筛选出文章中比较敏感的词汇进行修改，降低重复劳动。

# 二、系统开发环境

搭建开发环境需要的软件，如下：

操作环境：Windows 10

开发环境：JDK（Java Development Kit) 1.8+

编程语言：Java，JavaScript，HTML，CSS

集成开发环境：IntelliJ IDEA

数据库服务器：MySQL

Web服务器：Tomcat

# 三、系统运行环境

3.1硬件环境

该系统运行的硬件环境为：台式电脑或者笔记本电脑（PC机），也可以支持在Android系统上使用。

3.2软件环境

通过maven进行的项目构建，后端整体使用Java EE常用开发框架SpringBoot框架。同时，项目资源文件中存放HTML、CSS、JavaScrip文件通过Ajax将获得的数据显现出来。最后购买域名和系统空间，绑定IP地址和域名，配置服务器系统环境，安装MySQL，Tomcat，Nginx，FastDFS等软件运行环境。

# 四、系统功能介绍

4.1系统总体结构设计

敏感词分析系统，主要包含的功能有：登录、首页、用户管理、角色管理、权限管理、公告管理，在线用户，行为日志，环境监控等功能的后台管理平台，系统总体结构图，如图所示。

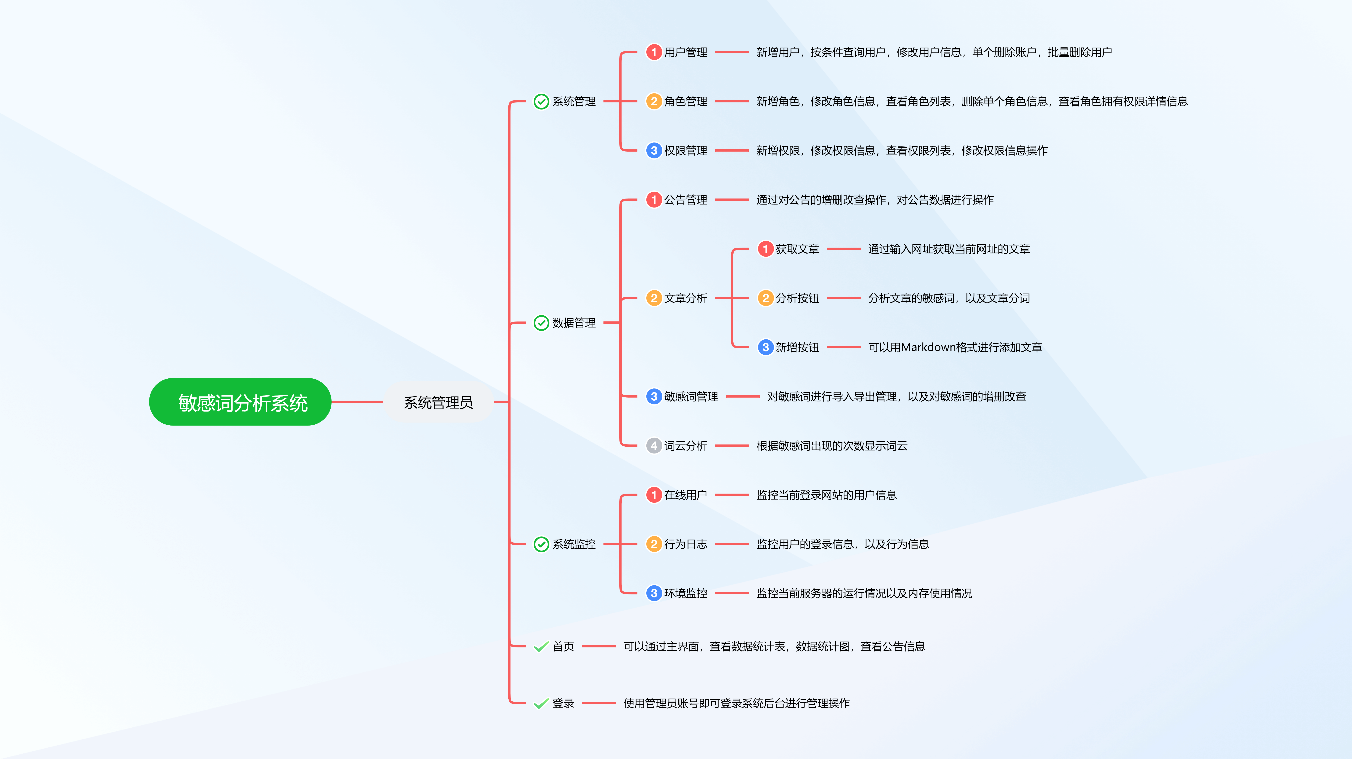


图 4‑1 系统总体结构设计

4.2软件系统

敏感词分析系统平台，登录网页对用户信息进行管理，登录模块的功能是验证用户登录的身份。在系统登录过程中，用户输入用户名密码确认登录后，系统将会在数据库查询该用户信息，判断用户是否存在。如果找到用户信息，并确认用户信息是否有效的，如果有效，系统则提示用户登录成功，进入系统主界面；如果没有查询到该用户信息则提示用户未注册，如果输入的用户密码是错误的，则用户登录的失败并提示用户账号密码错误，并重定向到登录页面。主要用于管理用户提交文章的信息

4.3 主界面

对可以通过首页来查看文章数量，用户数量，发布的公告，敏感词的数量，以及图表来显示文章的记录。

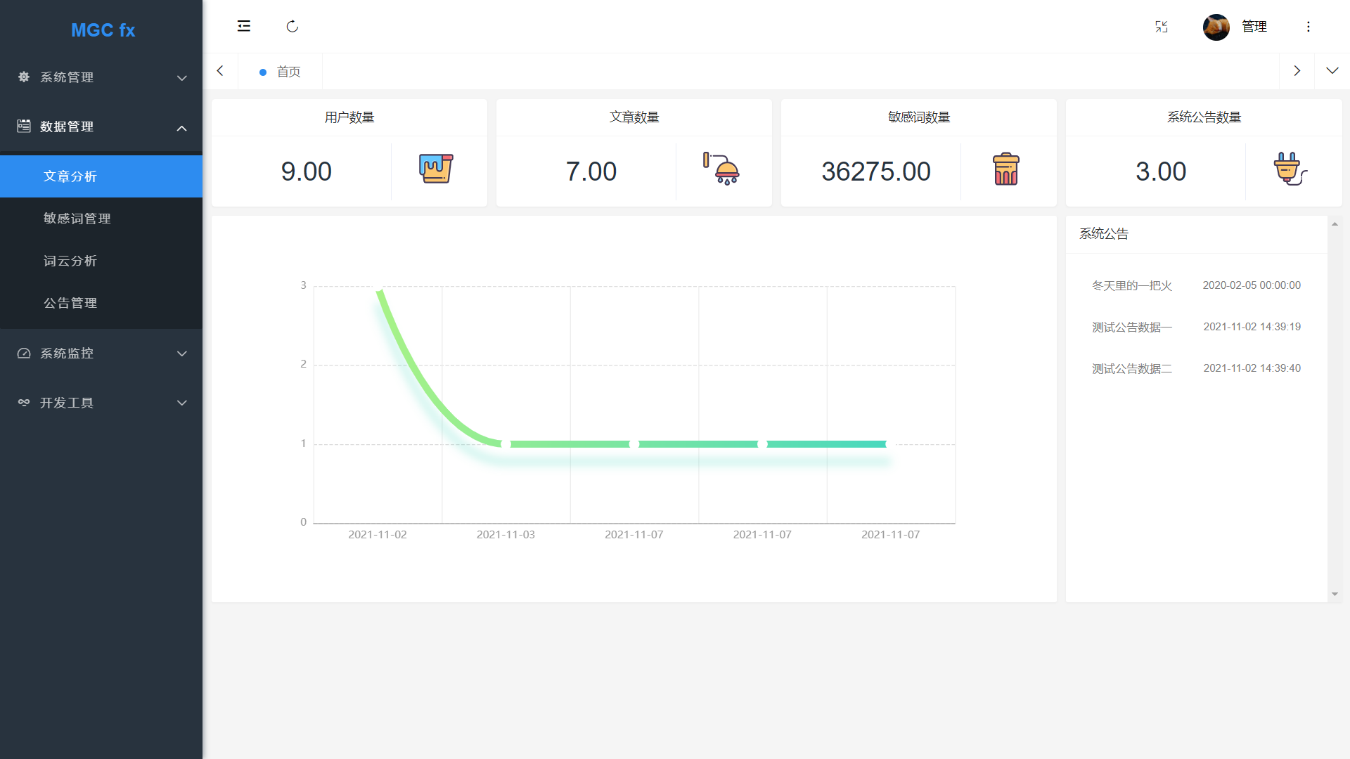


图 4‑2 主界面

4.4系统管理

4.4.1用户管理

对用户的增加，修改，查看，删除，以及条件搜索等操作

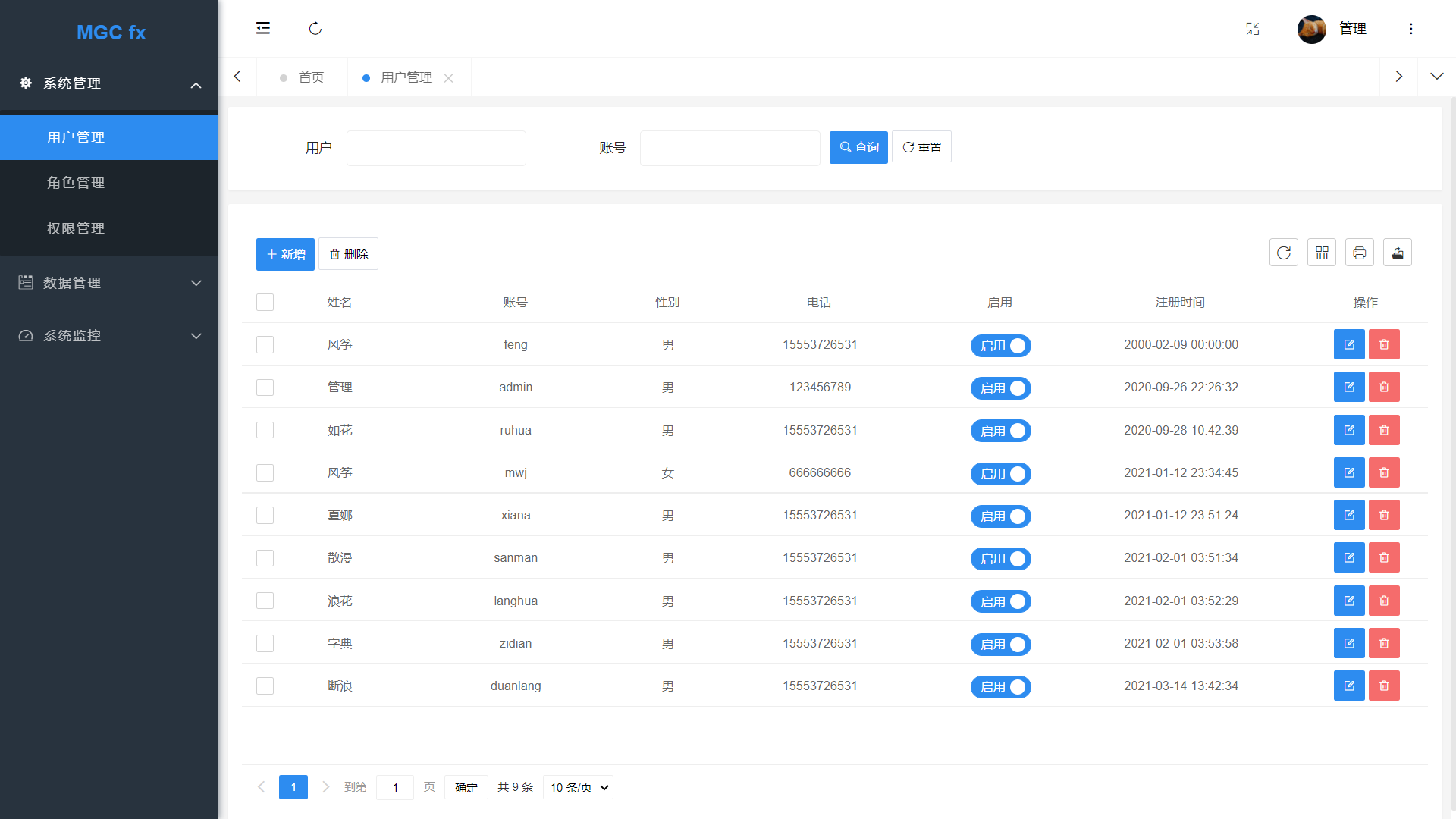
****

图 4‑4-1 用户管理

4.4.2 角色管理

对角色的增加，修改，查看，删除，以及详情查看等操作

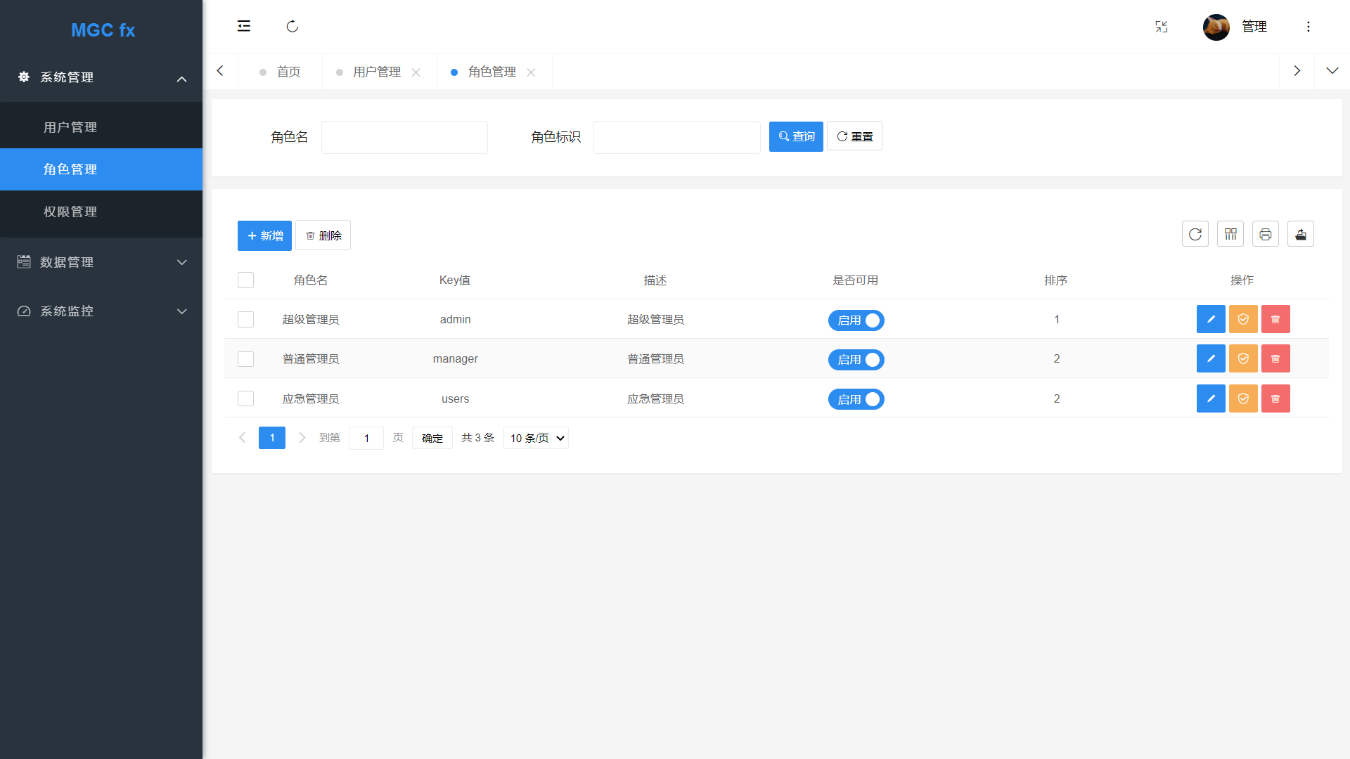


图 4‑4-2 角色管理

4.4.3 权限管理

对权限的增加，修改，查看，删除操作

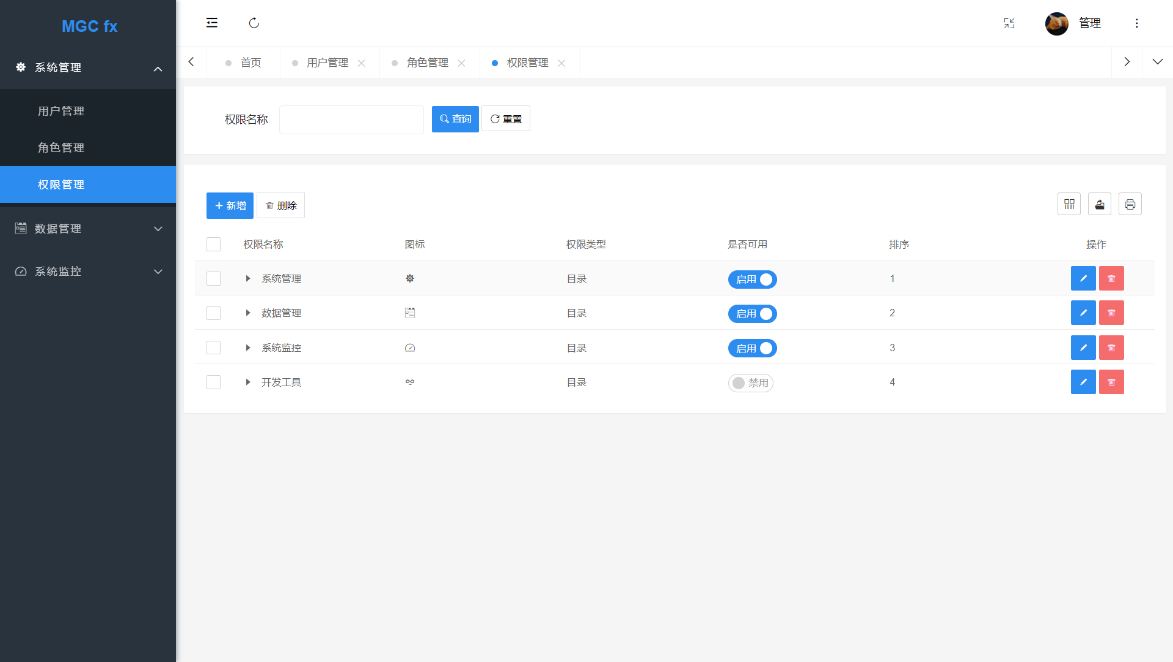


图 4‑4-3 权限管理

4.5数据管理

4.5.1 文章分析

对文章进行管理，点击获取文章按钮输入网址，可以获取文章信息。对文章信息进行增加删除，修改管理，可通过标题，作者信息查询文章信息，支持文章的导出，打印以及敏感词分析功能。

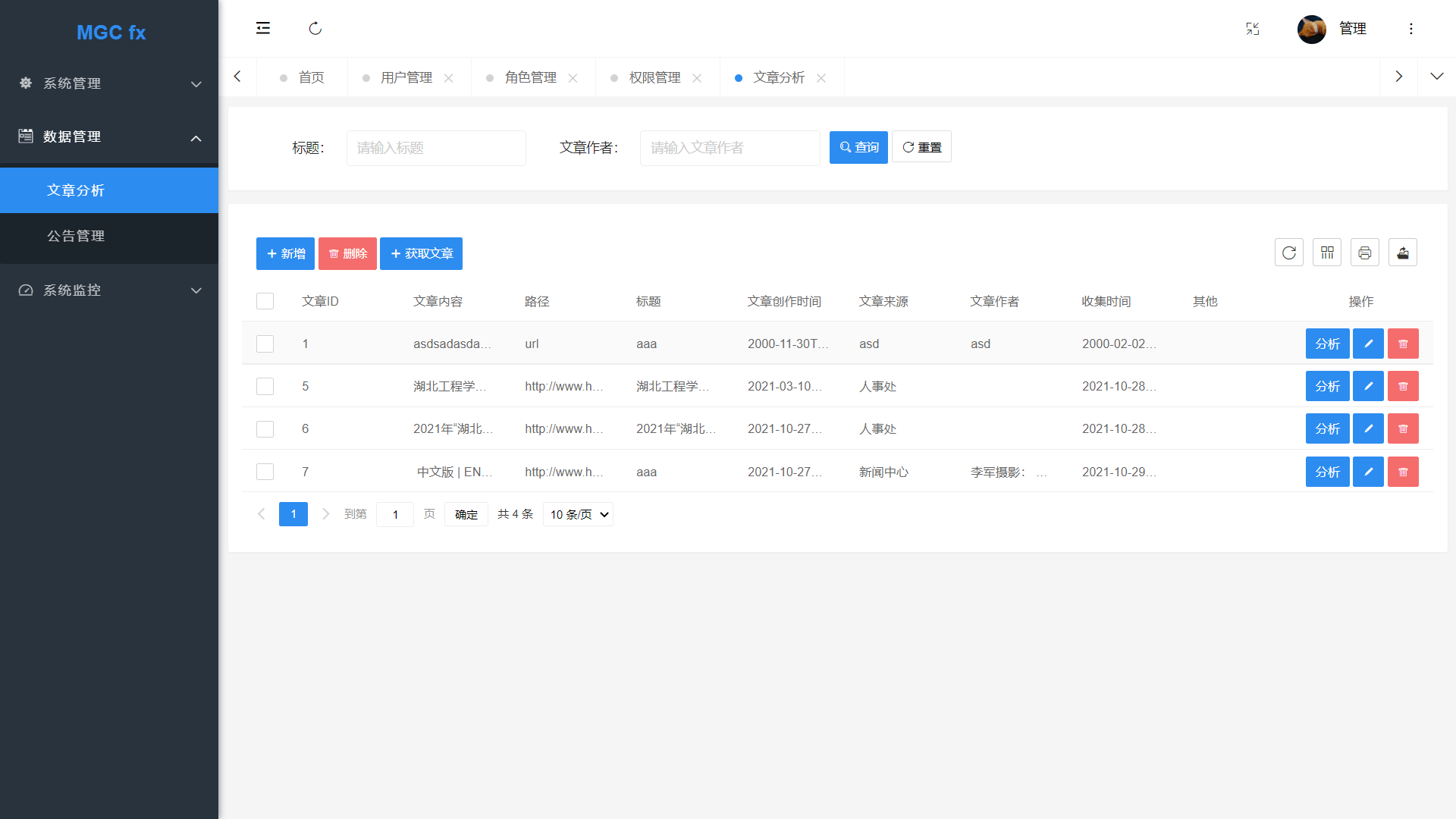


图 4‑5-1 文章分析

新增文章可以支持Markdown格式的文章。

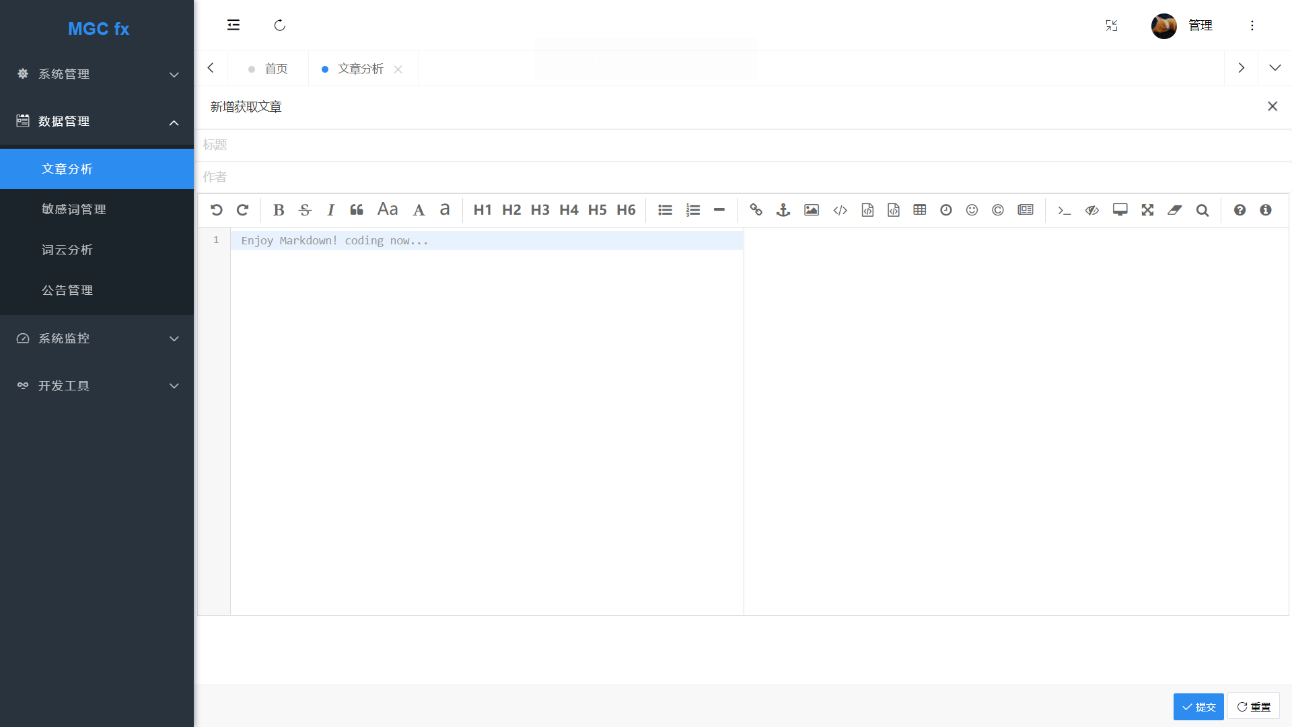


图 4‑5-2 新增文章

文章检测功能，可以对文章的敏感词进行分析，对文章进行分词

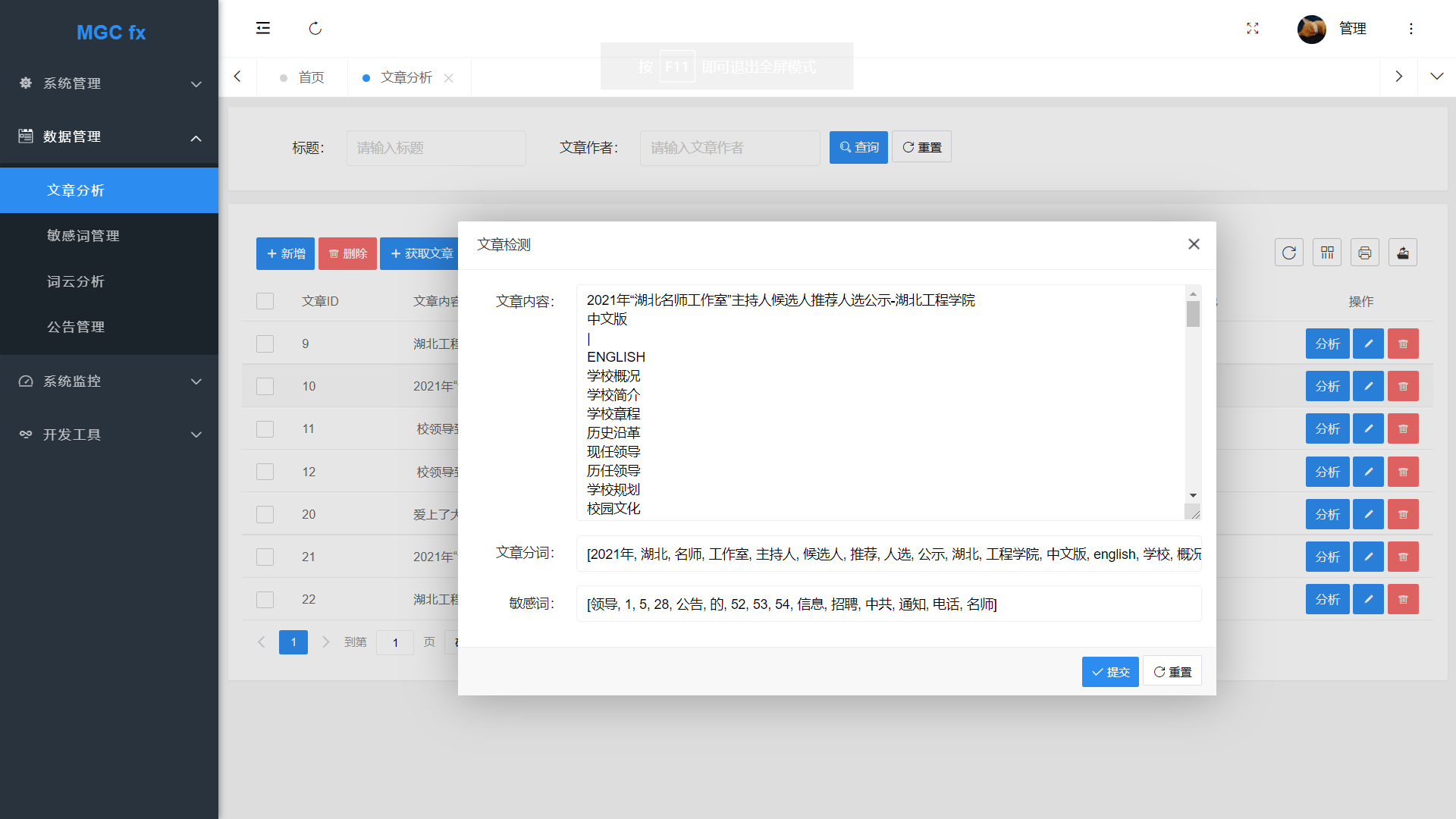


图 4‑5-3 文章检测

4.5.3 公告管理

对主页面的公告栏进行管理，可以查看对公告信息，进行增删改查等操作

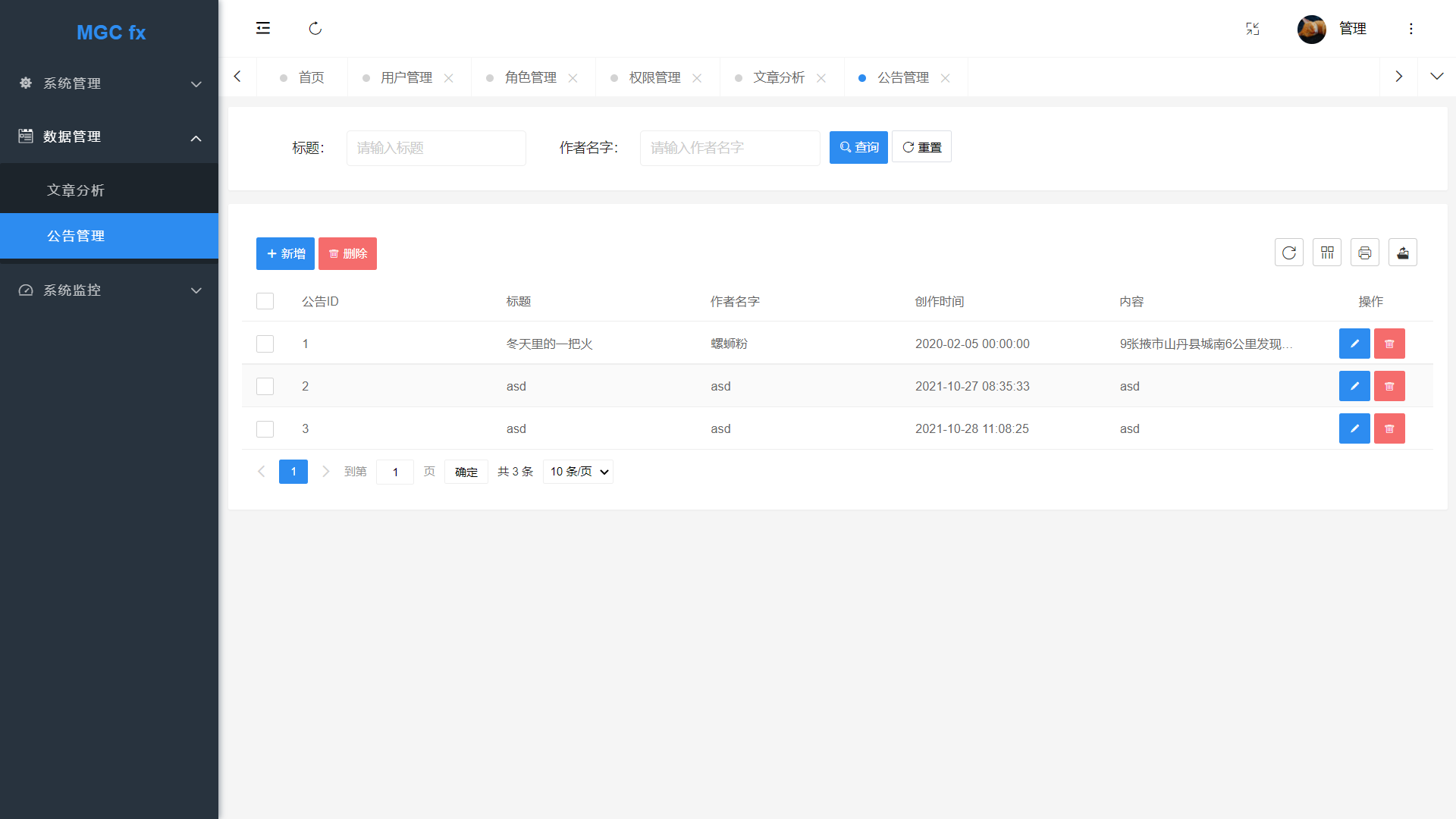


图 4‑5-4 公告管理

4.5.4 敏感词管理

对敏感词进行增删改查操作，导入导出数据功能，对敏感词进行管理。



图 4‑5-4 敏感词管理

4.5.4 词云分析

通过对敏感词的数量产生词云，对敏感词分析。



图 4‑5-4 词云分析

4．6系统监控

4.6.1 在线用户

可以监控此时同时登陆用户的账号信息。

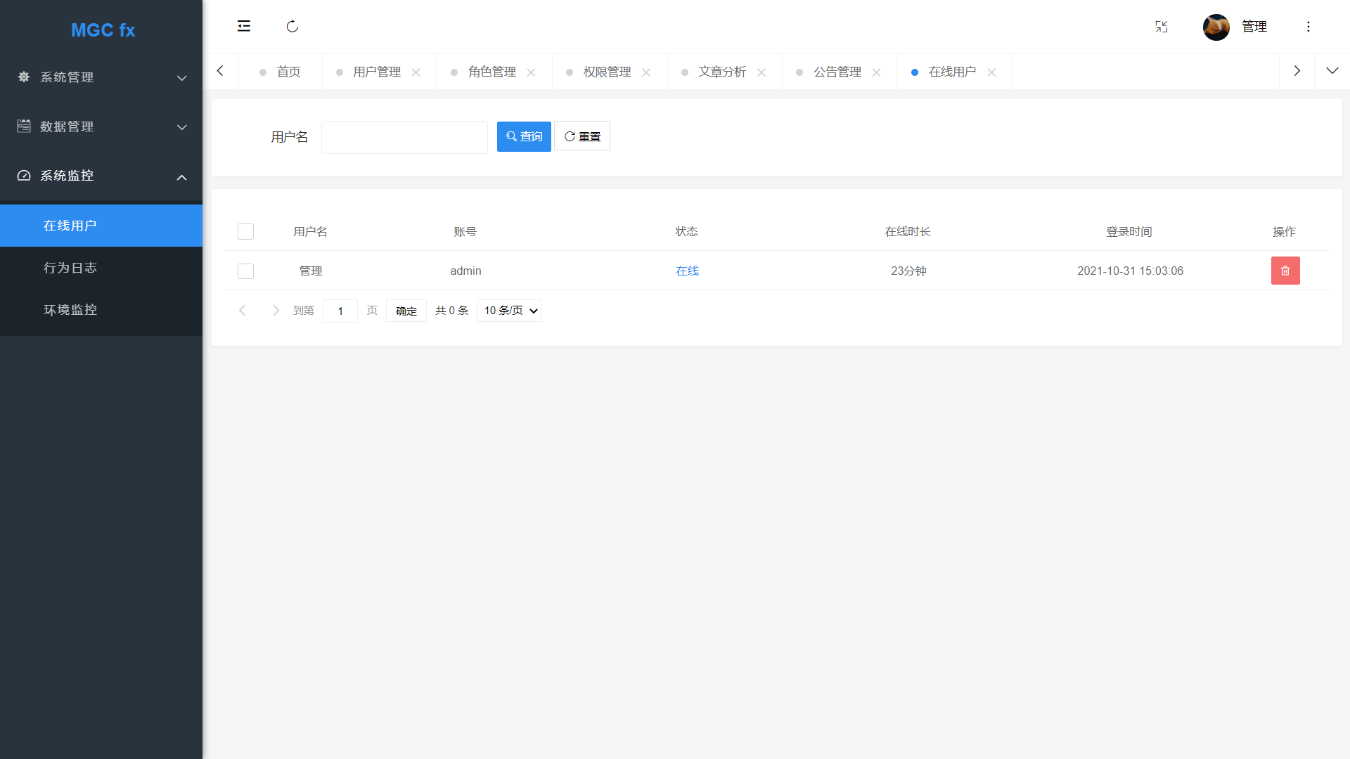


图 4‑6-1在线用户

4.6.2 行为日志

通过行为日志，来查看用户的历史登录信息，以及用户的行为信息。

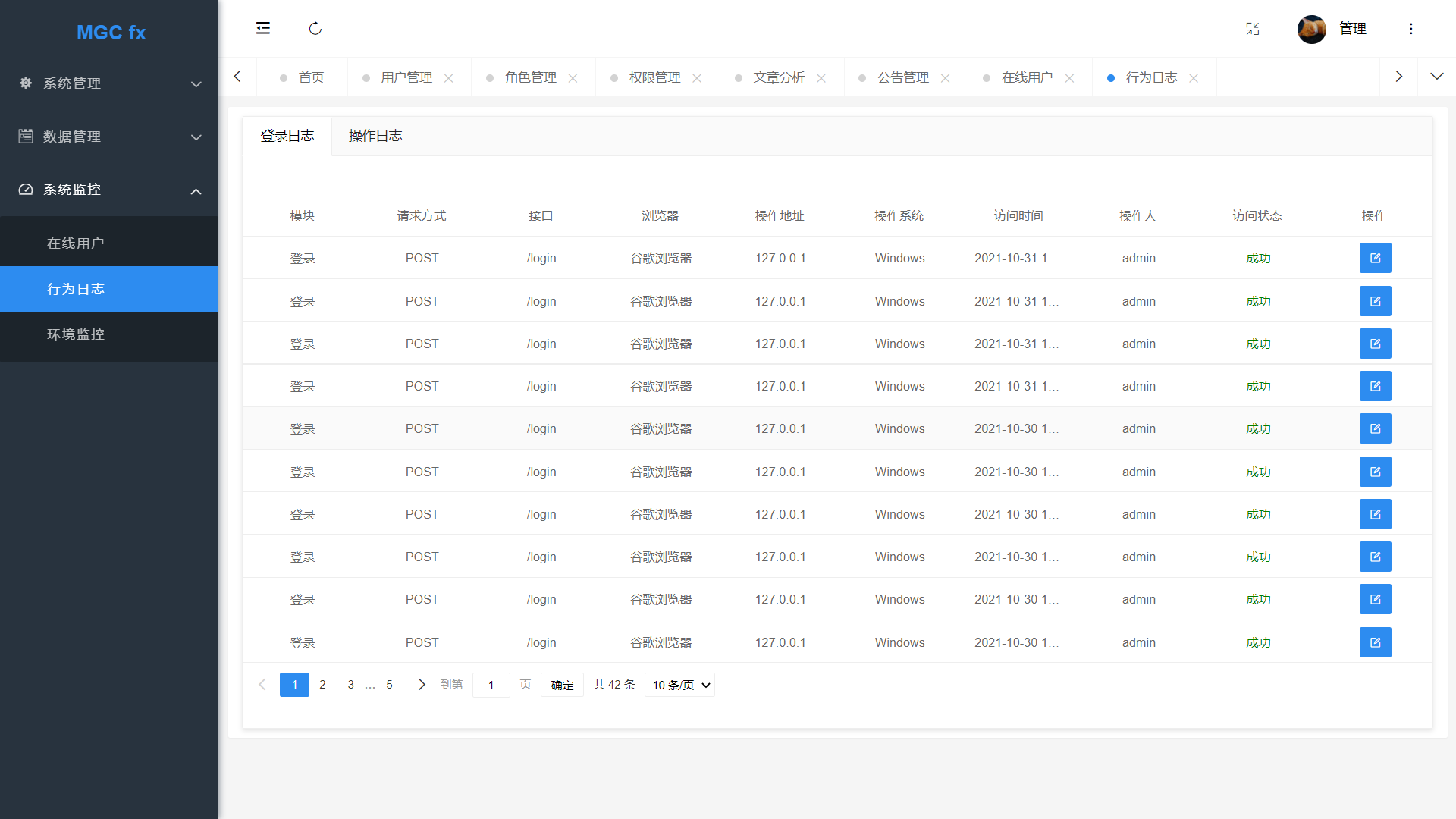


图 4‑6-2行为日志

4.6.3 环境监控

通过环境监控，查看主机信息，内存信息，磁盘信息，可以通过图例，更加直观的查看当前服务器的状态。



图 4‑6-3环境检测

4.7 登录

用户可以进行输入相应的信息后可以进行登录和注册，未注册的用户进行登录会提示“账号或密码不正确”

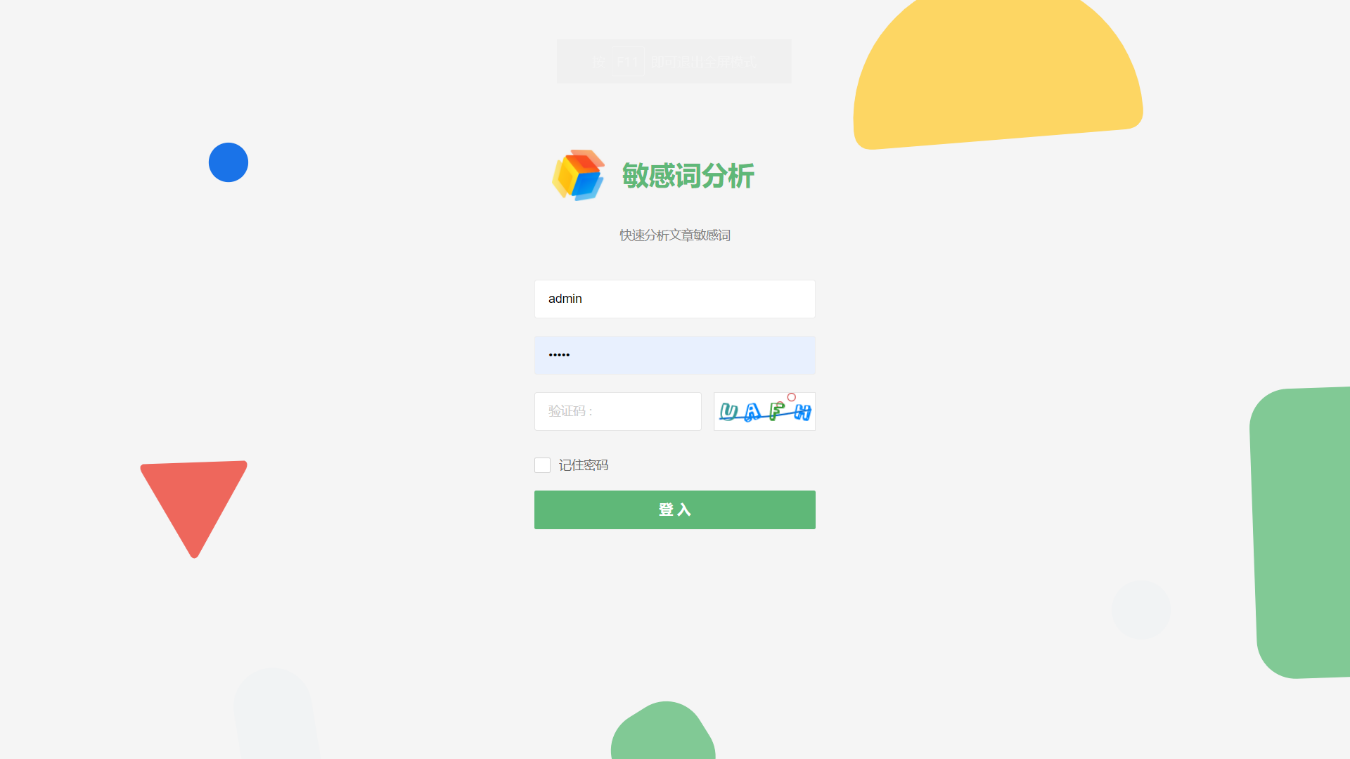


图 4‑7-1 用户登录

## 4.7.1 个人账号

可以修改个人账号信息，以及注销账号。

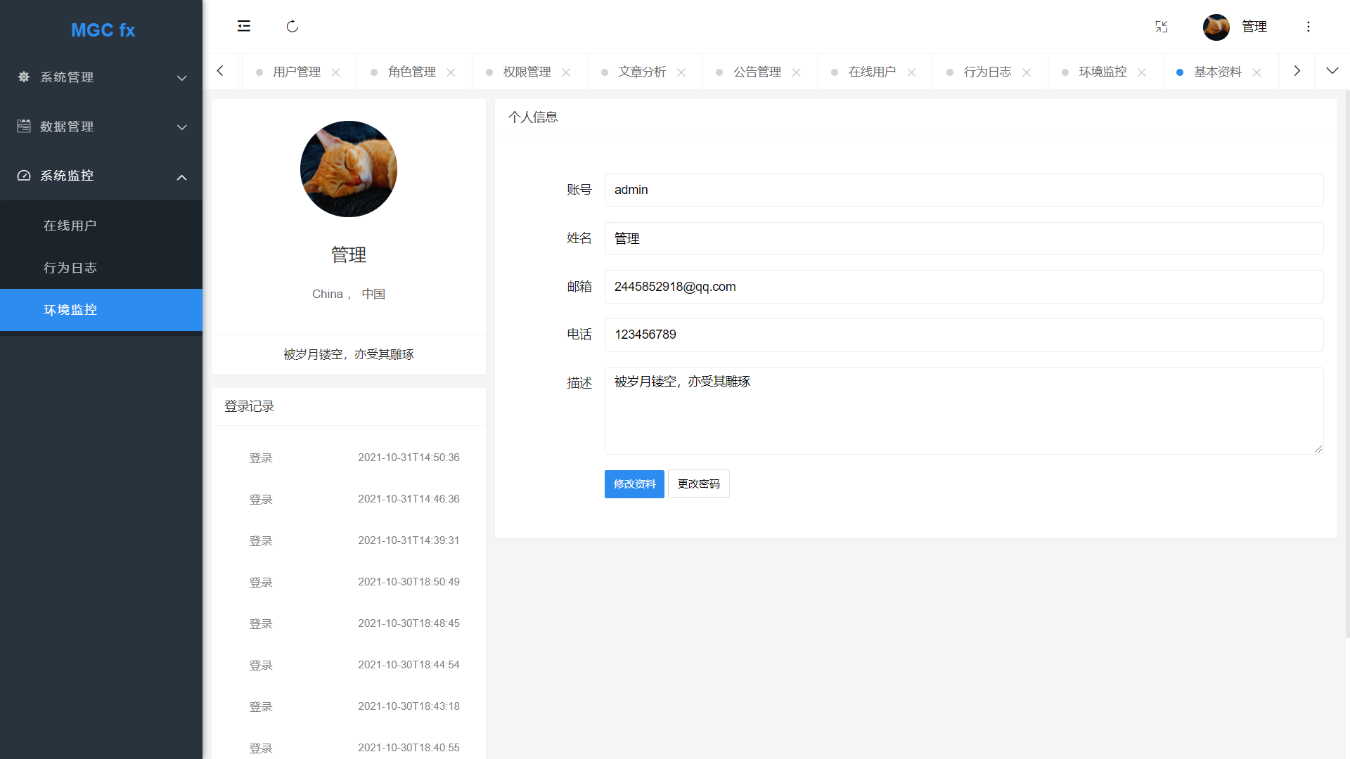


图 4‑7-1 账号信息



图 4‑7-2 注销账号

4.8 系统风格

用户可以通过右上角导航栏的扩展按钮，更改自己喜欢的网页风格。

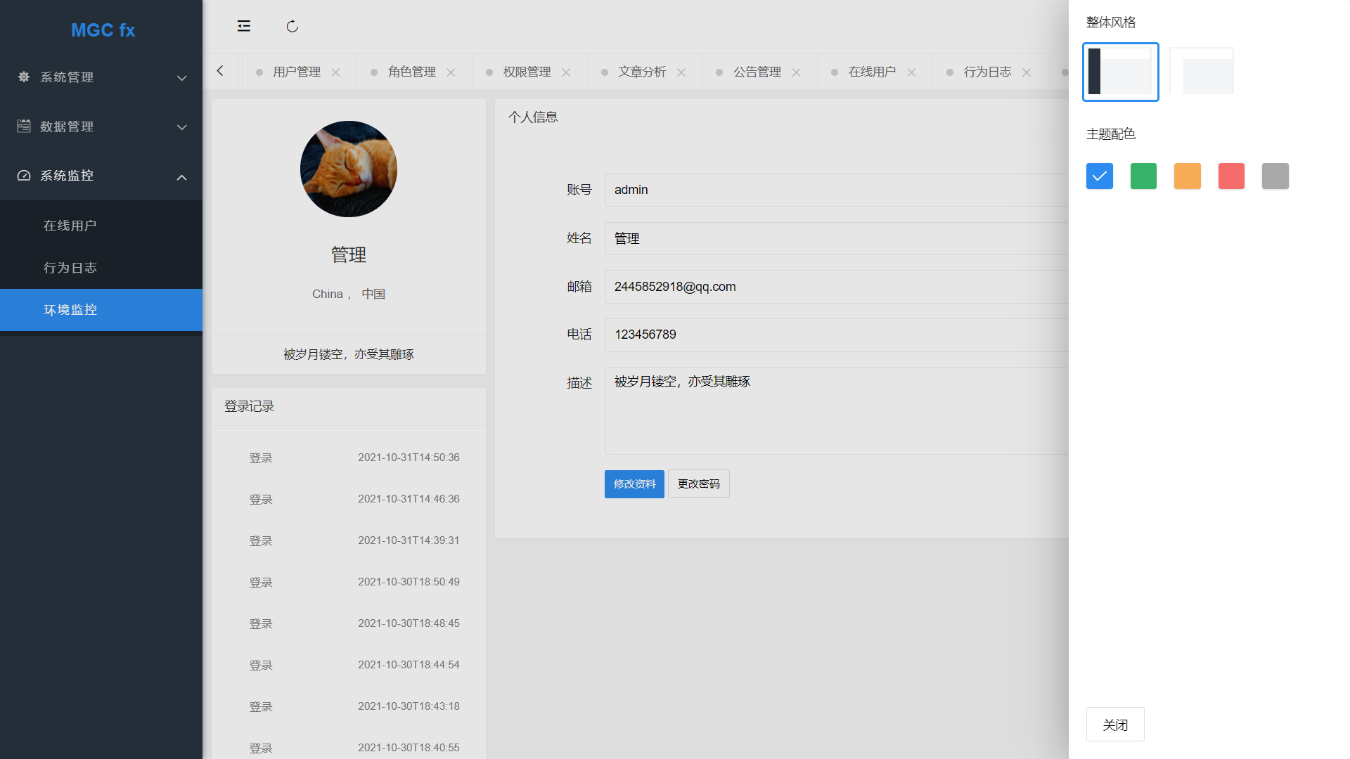


图 4‑8-1 系统风格

# 系统整体运行原理

通过Web页面，将用户上传的文章信息，进行敏感词检测，获取敏感词信息。

## 5.1 数据显示与管理

管理平台，主要用于管理用户上传的文章信息，进行相关的处理，分析，展示，查询，提示。

### 5.1.1 文件导出

需本平台已注册的用户，选择相应文件格式后进行导出操作

### 5.1.2 用户管理

管理员可以对所有已注册的用户信息进行修改、查询、删除等操作

## 5.2 界面设计需求

（1）设计风格大气、专业。

（2）页面上没有与页面无关的内容。

（3）设计界面清晰，操作简单，能及时接受和处理用户相关服务

（4）色彩搭配合适

（5）要有个性

## 5.3 确定系统实现技术

（1）根据系统的目的确定系统的数据结构

（2）根据系统的目的及内容确定系统功能。

（3）确定系统的功能导航中的每个子功能。

（4）确定网页内容的实现方式。

# 六、系统测试

本系统要求通过数据和数据库完整性测试、兼容性测试、功能测试、用户界面测试、性能评测和负载测试等方面的测试。用以检测系统功能是否完善，有无系统漏洞等问题。

## 6.1功能测试

### 6.1.1功能模块测试

功能模块是该数字证书系统的一个主要组成部分，用户可以通过鼠标的点击跳转到相关功能页面进行操作。

功能模块测试可分为以下三个方面：

1. 测试所有功能模块是否按指示的那样确实跳转到了该功能的页面。

注：功能模块测试必须在集成测试阶段完成，也就是说，在整个系统的所有页面开发完成之后进行测试。

a) 测试所有的功能页面是否存在。

b) 保证系统上没有孤立的页面（所谓孤立页面是指没有链接指向该页面， 只有知道正确的URL地址才能访问）

2.系统功能需求

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 功能名称 | 使用人 | 功能描述 | 输入内容 | 输出内容 |
| 1 | 获取文章 | 管理员 | 通过url爬取文章 | 文章网址 | 文章信息更新 |
| 2 | 文章分析 | 管理员 | 对文章进行铭感词分析 | 点击分析按钮 | 敏感词信息 |

3.系统功能测试报告

按照功能点列表内容，结合等价类划分法设计测试用例（输入/输出内容），进行现场测试，记录测试数据，评定测试结果。结果如下表所示：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 功能 | 输入内容 | 期望输出 | 输出内容 | 发现问题 | 结果 | 测试时间 | 测试人 |
| 1 | 获取文章 | 文章网址 | 爬取成功 | 爬取成功 | 未发现 | 正常 | 2021.10.31 | 张三 |
| 2 | 文章分析 | 点击分析按钮 | 查询成功 | 敏感词信息 | 未发现 | 正常 | 2021.10.31 | 张山 |

### 6.1.2表单测试

表单就是一些需要在线显示和填写的表格。表单中有一些标准操作，如确认、保存、提交等，当用户通过表单提交信息的时候，都希望表单能正常工作。 表单测试可以分为5个方面：

1. 用户（包括用户和系统管理员）注册、登录、信息提交等，必须测试提交操作的完整性，以校验提交给服务器的用户信息的准确性。

2. 检查相关联的数据显示是否正确。

3. 如果使用了默认值还需要检验默认值的正确性。

4. 如果表单只能接受指定的某些值，则也要进行测试。例如：只能接受指定的某些值， 测试时可以跳过这些字符，看系统是否会报错。

5. 如系统只接受限定以内的字数，测试时可以测试内外边界值，看系统是否可正确反应。

## 6.2性能测试（包括负载/压力测试）

### 6.2.1系统页面响应速度测试

1.系统响应时间

2.系统响应超时的限制

### 6.2.2系统负载测试

某个时刻同时访问系统的节点数量 通常需要注意以下内容：

1.系统在某时刻最多允许多少个节点同时在线？

2.如果超过了这个数量，会出现什么现象？

3.系统能否处理大量用户同时对同一个页面的请求？

### 6.2.3系统压力测试

（1）重复测试

重复针对系统一个功能反复操作，看系统是否能正常运行。

（2）并发测试

就是在同一时间内执行多个操作。由并发引起的错误只能通过执行多个代码示例才能测出来，测试时间要同时遍历多条代码路径。 通常可以测试以下几个方面：

1. 登录的并发操作

2. 提交的并发操作

3. 对数据库操作的并发操作（例如：多个节点同时从数据库中读出（或向数据库导入）相 同文件、不同文件）

1. 量级增加

要考虑到每个操作中的负载量，操作自身应尽可能给被测软件系统增加压力，即：尽量使单独的操作进行高强度的使用，增加操作的量级。

例如：改变数据的大小、改变时间延迟的长度、输入速度以及输入的变化等。

1. 随机变化

任何压力系统都多多少少具有一些随机性。随机使用前面的压力原则中介绍的无数变化形式，就能够在每次测试运行时应用许多不同的代码路径。

## 6.3用户界面测试/可用性测试

界面测试是从系统用户的角度来看软件，系统能否满足用户的需要，是系统评测的一个重要标准。可以从以下几个方面重点来检查用户界面：

1. 易用性检查：验证系统界面是否易于用户理解、是否方便使用。

2. 正确性检查：检查页面上的表单、按钮、窗体、提示信息、文字拼写等是否正确以及是否存在错别字。

3. 一致性检查：

a) 系统页面的风格是否一致，如字的大小、颜色、字体要相同。

b) 提示信息的表达方式是否一致。

c) 按钮排列顺序是否一致。

d) 按钮跳转页面处理是否一致。

e) 相同字段的名称、长度、类型在不同位置是否一致。

4. 友好性检查：

a) 提示信息是否友好。

b) 执行风险操作时系统是否给出提示信息让用户确认是否继续操作。

c) 系统页面分辨率检查，在各种分辨率浏览系统检查系统界面友好性。

d) 系统页面进行最大化、最小化还原时是否做了相应的处理。

5. 合理性检查：管理员在进行添加、修改、删除、返回等操作后，查看数据的更改情况是否合理。

## 6.4系统兼容性测试

系统兼容性测试内容：

1.跨操作系统测试（例如：Windows 10/Window 7/Linux等）

2.浏览器测试（IE浏览器/Google Chrome浏览器/Firefox浏览器等）

3.分辨率测试（在不同分辨率下系统界面是否会发生错位情况）

## 6.5安全性测试

### 6.5.1登录安全性

现在的管理系统基本采用先注册，后登录的方式。因此，必须通过与后台数据库对比，测试有效和无效的用户名和密码，还需要注意到是否大小写敏感，可以测试多少次的登录限制，是否可以不登录而直接浏览某个页面等。

### 6.5.2日志文件

为了保证电梯安全卫士系统的安全性，需要测试后台相关操作信息是否写进了日志文件。能否查到对系统数据更改的信息，以防操作失误导致系统数据丢失。

### 6.5.3网址设置

将系统的某个功能模块的url地址，然后直接打开页面输入该url地址看其是否能跨过系统的登录模块直接进入该功能页面。

七、系统出错处理设计

数字证书系统出错最严重的就是系统由众多软件模块组成，一环扣一环，其中任何一个模块出现错误将会导致整个系统崩溃，为了从根本上解决这个问题，首先在编写前台和后台代码时就要注意编写的严密性，每次在对模块进行编程时都要设置提示语句来显示每个模块的使用情况，其次，在模块的选择上尽量做到质量达标，以免对用户形成不好的体验。

7.1出错信息

用一览表的方式说明每一种可能出错的情况出现时，系统输出信息的形式、含义及处理的方法。

由于输入信息不符合规范（如：电梯内的参数发送格式不符合规范），称之为软错误。

由于硬件方面的错误（如：网络传输超时、硬件出错等），称之为软错误。

对于一些关键的操作（如：删除操作），应该提供提示确认机制。

对于数据、测试文档，都是要提供相应的保密措施设置。

|  |  |  |
| --- | --- | --- |
| 错误类型 | 子项 | 错误原因 |
| 数据库错误 | 连接 | 连接超时 |
| 连接断开 |
| 数据库本身 | 数据库代码错误 |
| 数据库溢出 |
| TCP连接错误 | 连接 | 连接超时 |
| 连接断开 |
| 其他TCP错误 | Socket自身错误 |
| 系统部分自定义错误 | 权限错误 | 管理员权限故障 |
| 输入错误 | 用户ID错误/为空 |
| 用户密码错误/为空 |
| 验证码错误/为空 |
| 查找错误 | 未查找到符合要求的记录 |
| 链接错误 | 内部链接错误 | 网页链接错误 |
| 文件/图片链接错误 |
| 外部链接错误 | 页面修整或其他错误无法打开 |
| 友情网址维护或其他错误无法打开 |

7.2补救措施

说明故障出现后可能采取的变通措施，主要包括：

1．对于软错误，需要在添加/修改操作中及时对输入数据进行验证，分析错误的类型，并且给出相应的错误提示语句，传送到客户端的浏览器上。

2．对于硬错误，错误类型较少而且比较明确，所以可以在可能出错的地方中输出相应的出错语句，并将程序重置，最后返回输入阶段。

3．后备技术说明准备采用的后备技术，当原始数据丢失时启用的副本的建立和启动的技术，例如周期性的把磁盘信息记录在案。

4．恢复及在启动技术说明将使用的恢复再启动技术，是软件从故障点恢复执行或是软件从头开始重新开始运行的方法。

1. 软件维护设计

8.1影响维护工作的因素

1. 系统大小：系统越大，理解掌握起来越困难，所执行功能越复杂因而需要更多的维护工作量。
2. 硬件质量：硬件质量越好，系统的稳定性更强，系统的寿命更长，所以系统开发之初应该选择质量有保证的硬件模块，最好有可靠的售后服务以保证模块出现故障时可以及时排查和修复。
3. 程序设计语言：语言功能越强，生产程序所需要的指令数越少；语言的功能越弱，实现同样功能所需语句越多，程序就越大。有许多软件使用较老的程序设计语言书写的，程序逻辑复杂而且混乱，没有做到模块化和结构化，直接影响程序的可读性。
4. 系统年龄：老系统随着不断的修改，结构越来越乱；由于维护人员程长更换，程序又变得越来越难以理解。而且许多老系统在当初并未按照软件工程的要求进行开发，因为没有文档或者文档太少，在长期的维护过程中在许多地方与程序实现变得不一致，这样维护时就会遇到很大的困难。
5. 数据库技术的应用：使用数据库可以简单而有效地管理和存储用户中的数据，还可以减少生成用户报表应用软件的维护工作量。

（6）先进的软件开发技术：在软件开发时，若使用能使软件结构计较稳定的分析与设计技术及程序设计技术，如面向对象技术、复用技术等，可以减少大量的工作量。

8.2软件维护的策略

（1）改正性维护：要生成维护100%可靠的软件成本太高，不一定合算。但通过使用新技术可大大提高可靠性，减少进行改正性维护的需要。这些技术包括数据库管理系统、软件开发环境、程序自动生成系统、较高级的语言，应用时尚4种方法可产生更加可靠的代码。此外，利用应用软件包可开发出比完全由用户自己开发的系统可靠性更高的软件；利用结构化开发的软件易于理解和测试；通过防错性程序设计把自检能力引入程序，通过非正常状态的检查提供审查跟踪；通过周期性维护审查，在形成维护问题之前就可确定质量缺陷。

（2）适应性维护：这一类维护不可避免，但可以控制。在配置管理时，把硬件、操作系统和其他相关环境因素的可能变化考虑在内，可以减少某些适应性维护的工作量。把与硬件、操作系统，以及其他外围设备有关的程序归到特定的程序模块中，可把因环境变化为必须修改的程序局部于某些程序模块之中。使用内部程序列表、外部文件以及处理的例行程序包，可为维护时修改程序提供方便。

（3）完善性维护：利用前两类维护中列举的方法，也可以减少这一类维护。特别是数据库管理系统、程序生成器、应用软件包，可减少系统或程序员的维护工作量。此外，建立软件的原型，把他在实际系统开发之前提供给用户。用户通过研究原型，进一步完善他们的功能需求，就可以减少以后完善维护的需要。