****

**本科生毕业论文（设计）**

|  |  |
| --- | --- |
| 题 目： | 博客的设计与开发 |
| 院 系： | 电气与计算机工程学院 |
| 专 业： | 计算机科学与技术 |
| 学生姓名： | 曾榆洲 |
| 学 号： | 2020170210 |
| 指导教师： | 朱海峰 |

二〇年二四五月

**论文原创性声明**

本人郑重声明:所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究作出重要贡献的个人和集体（及其成果），均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

本人签名：

日期：20××年5月××日

**论文授权使用声明**

本人完全了解广州南方学院有关保留、使用学位论文的规定，即:学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版;有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆、院系资料室被查阅;有权将学位论文的内容编入有关数据库进行检索;可以采用复印、缩印或其他方法保存学位论文;可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

保密论文保密期满后，适用本声明。

本人签名： 指导教师签名：

日期：20××年5月××日

# 摘 要

本论文介绍了智谷博客系统的设计与实现，采用了Java技术栈中的Spring Boot框架作为后端支撑，结合Vue.js作为前端框架，以及Thymeleaf和Element UI等关键技术。智谷博客旨在提供一个功能完善、用户友好的博客平台，用户可以轻松创建个人博客、发布文章、管理，实现与其他用户的交流互动。本文详细介绍了系统的架构设计、技术选型、功能实现以及性能优化等方面，旨在展示如何利用Java技术栈构建稳定、高效的Web应用程序。通过对智谷博客系统的设计与实现，为相关领域的研究与实践提供了有益的参考。

关键词：智谷博客、Java技术、Spring Boot、Vue.js、Thymeleaf、Element UI、Web应用程序、博客系统

# ABSTRACT

This paper introduces the design and implementation of the Zhigu blog system, using the Spring Boot framework in the Java technology stack as back-end support, combined with Vue.js as the front-end framework, and key technologies such as Thymeleaf and Element UI. Zhigu Blog aims to provide a fully functional and user-friendly blog platform. Users can easily create personal blogs, publish articles, manage comments, and achieve communication and interaction with other users. This article introduces the system's architectural design, technology selection, function implementation, and performance optimization in detail, aiming to show how to use the Java technology stack to build stable and efficient Web applications. Through the design and implementation of the Zhigu blog system, it provides a useful reference for research and practice in related fields.

**Key words:** Zhigu Blog, Java technology, Spring Boot, Vue.js, Thymeleaf, Element UI, Web application, blog system

# 目 录

[摘 要 I](#_Toc14303)

[ABSTRACT II](#_Toc3976)

[目 录 III](#_Toc2468)

[第1章 绪论 1](#_Toc1974)

[1.1研究背景 1](#_Toc14651)

[1.2研究意义 1](#_Toc3760)

[1.3研究现状 2](#_Toc18613)

[1.4开发环境与工具 2](#_Toc16933)

[1.4.1 IDEA 2](#_Toc6988)

[1.4.2 Navicat 2](#_Toc19367)

[1.4.3 java版本 3](#_Toc8854)

[1.4.4 MySQL数据库 3](#_Toc21369)

[第2章 关键技术分析 4](#_Toc5805)

[2.1 SpringBoot技术 4](#_Toc9897)

[2.2 Vue.js技术 4](#_Toc6236)

[2.3 Java技术 5](#_Toc4866)

[2.4Thymeleaf技术 6](#_Toc21952)

[2.5 Element UI 6](#_Toc20389)

[2.5 jQuery 7](#_Toc19723)

[第3章 系统分析与设计 9](#_Toc16195)

[3.1可行性分析 9](#_Toc26189)

[3.1.1技术可行性： 9](#_Toc335)

[3.1.2经济可行性： 9](#_Toc9309)

[3.2系统需求分析 9](#_Toc12708)

[3.2.1 智谷博客前台功能需求分析 9](#_Toc28267)

[3.2.1 智谷博客后台功能需求分析 10](#_Toc29297)

[3.2系统总体设计 10](#_Toc16063)

[3.3数据库设计 11](#_Toc31335)

[3.3.1数据库概念模型设计 11](#_Toc10404)

[3.3.2数据库表设计 14](#_Toc29057)

[第4章 系统实现 17](#_Toc30365)

[4.1登录注册功能 17](#_Toc30941)

[4.2首页 19](#_Toc27476)

[4.3问答功能 21](#_Toc24185)

[4.3资讯功能 23](#_Toc17119)

[4.4退出功能 24](#_Toc17370)

[4.5数据统计管理功能 25](#_Toc16210)

[4.6用户管理功能 28](#_Toc30435)

[4.7分类管理功能 33](#_Toc8884)

[4.8博客管理功能 36](#_Toc30310)

[4.9问答管理功能 40](#_Toc18898)

[4.10资讯管理 42](#_Toc10291)

[4.11系统日志功能 44](#_Toc30350)

[第5章 系统测试 46](#_Toc13493)

[5.1概述 46](#_Toc3466)

[5.2测试目标 46](#_Toc23642)

[5.3测试环境 46](#_Toc27246)

[5.4测试用例 46](#_Toc32090)

[2.4.8致谢 50](#_Toc18196)

[2.4.9参考文献 50](#_Toc29848)

[2.4.10附录 51](#_Toc21857)

[2.4.11页眉 51](#_Toc6839)

[致 谢 53](#_Toc26772)

[参考文献 54](#_Toc26258)

[附 录 55](#_Toc12874)

# 

# 第1章 绪论

## 1.1研究背景

博客系统作为一种网络信息传播工具，在互联网时代扮演着越来越重要的角色。它不仅是个人和组织展示自我、分享知识和观点的平台，也是用户获取信息、进行交流互动的重要渠道之一。随着互联网的快速发展，传统的博客形式逐渐演化为多样化、功能丰富的博客系统，以满足不断增长的用户需求。

然而，现有的博客系统在面临着一些挑战。首先，用户对于博客的体验要求越来越高，他们希望能够在博客上获得更加丰富、动态的内容，并能够与其他用户进行交流和互动。其次，随着移动互联网的普及，博客系统需要具备良好的响应性能，能够适应不同终端设备的浏览需求。此外，博客系统还需要考虑到安全性、可扩展性等方面的问题，以保障用户信息和系统稳定运行。

为了应对这些挑战，许多研究者和开发者开始探索新的博客系统设计和实现方案。结合现代化的技术和框架，如Java技术、Spring Boot、Vue.js等，可以为博客系统带来更强大的功能和更好的用户体验。通过对博客系统的研究和优化，可以不断提升其性能、安全性和用户友好性，满足用户日益增长的需求。

因此，本研究旨在深入探讨博客系统的设计与实现，利用先进的技术手段构建一款功能强大、性能优越的博客平台。通过对博客系统的研究与实践，可以为相关领域的技术开发和应用提供有益的参考和借鉴。

## 1.2研究意义

博客系统作为互联网信息传播的重要工具，在当今数字化时代扮演着关键角色。对博客系统的深入研究具有重要的理论和实践意义。

首先，博客系统的研究有助于深入了解现代网络应用程序的设计与实现原理。通过分析博客系统的架构、功能模块和技术实现，可以探索其中涉及的前沿技术和方法，为网络应用程序的进一步发展提供理论指导和技术支持。

其次，博客系统的研究可以促进相关技术的创新与应用。博客系统涉及到诸多技术领域，包括后端开发、前端设计、数据库管理等，通过对其进行深入研究和探索，可以推动这些技术的发展和应用，促进互联网技术的创新与进步。

再者，博客系统的优化和改进对于提升用户体验和信息传播效率具有重要意义。随着互联网用户的不断增加，用户对于博客系统的功能和性能提出了更高的要求，研究如何优化博客系统的功能和性能，可以提高用户的使用体验，促进信息的传播和交流。

综上所述，研究博客系统具有重要的理论和实践意义，对于促进网络应用程序的发展和推动信息社会的进步具有积极的作用。

## 1.3研究现状

在当今数字化时代，博客系统已经成为网络信息传播的重要载体之一，吸引了广泛的关注和研究。目前，关于博客系统的研究主要集中在以下几个方面：

1.技术实现： 许多研究者致力于探索和优化博客系统的技术实现，包括后端开发、前端设计、数据库管理等方面。其中，后端框架如Spring Boot等的应用，以及前端框架如Vue.js、React等的运用，受到了广泛关注。

2.用户体验： 随着用户对博客系统的要求不断提高，研究者开始关注如何提升博客系统的用户体验。通过优化界面设计、改善页面加载速度、增强交互性等方式，致力于提供更好的用户体验。

3.内容管理： 博客系统的内容管理是研究的重要方向之一。研究者通过分析用户行为、挖掘用户需求，探索如何优化博客内容的撰写、发布、管理和推荐，提高内容的质量和吸引力。

4.社交互动： 社交互动是博客系统的核心功能之一，也是当前研究的热点之一。研究者致力于探索如何促进用户之间的交流和互动，增强用户粘性和参与度，提升博客系统的社交价值。

5.安全性与隐私保护： 随着网络安全问题的日益突出，研究者开始关注博客系统的安全性和隐私保护问题。他们通过加强身份认证、加密通信、数据备份等方式，保障用户信息的安全和隐私。

6.综上所述，当前博客系统研究的主要趋势是技术实现的优化、用户体验的提升、内容管理的创新、社交互动的加强以及安全性与隐私保护的强化。未来，随着互联网技术的不断发展和用户需求的不断变化，博客系统的研究仍将是一个持续关注的热点领域。

## 1.4开发环境与工具

### 1.4.1 IDEA

DEA是IntelliJ IDEA的简称，是一款集成开发环境，被广泛认可为Java编程语言开发的优秀工具。它在智能代码助手、代码自动提示、重构、JavaEE支持、版本工具集成、单元测试、代码分析等方面具有卓越的功能。

### 1.4.2 Navicat

Navicat是一款数据库管理工具，支持多种数据库，包括MySQL、PostgreSQL、Oracle等。它具有直观的界面和丰富的功能，能够快速方便地进行数据库的管理、查询和操作，提高开发人员的工作效率。

### 1.4.3 java版本

Java版本： 本系统使用的Java版本为jdk1.8.0，即Java Development Kit 8。Java 8引入了许多新特性和改进，包括Lambda表达式、Stream API、新的日期和时间API等，提升了Java语言的功能和性能。

### 1.4.4 MySQL数据库

MySQL是一种开源的关系型数据库管理系统，被广泛应用于各种Web应用程序和企业级系统中。它具有高性能、可靠性高、易用性好等特点，适合于中小型应用系统的开发和部署。

# **第2章** 关键技术分析

## 2.1 SpringBoot技术

Spring Boot是一种用于构建基于Spring框架的应用程序的快速开发框架。它简化了Spring应用程序的配置和部署过程，使开发者能够更快速、更便捷地创建独立的、生产级别的Spring应用程序。以下是Spring Boot技术的主要特点和优势：

1.简化配置： Spring Boot采用约定大于配置的理念，提供了自动配置的功能，减少了开发者的配置工作量。大部分情况下，Spring Boot能够根据应用程序的依赖和环境自动配置所需的组件。

2.内嵌容器： Spring Boot集成了常用的内嵌式Web容器，如Tomcat、Jetty等，使得应用程序可以打包成一个可执行的JAR文件，不再需要依赖外部的Web服务器。

3.自动化构建： Spring Boot支持自动化构建工具，如Maven、Gradle等，能够自动管理项目的依赖和构建过程，简化了项目的构建和管理。

4.健康检查： Spring Boot提供了健康检查、监控和管理端点，可以通过HTTP接口实时查看应用程序的运行状态、性能指标等，便于进行故障排查和性能优化。

5.集成测试： Spring Boot提供了丰富的测试支持，包括单元测试、集成测试等，能够帮助开发者编写高质量的测试代码，确保应用程序的稳定性和可靠性。

6.生态系统： Spring Boot拥有庞大的生态系统，有大量的第三方库和插件可以与之集成，如Spring Security、Spring Data等，能够满足各种不同需求的应用场景。

总之，Spring Boot是一种简化Spring应用程序开发的利器，它通过简化配置、内嵌容器、自动化构建等功能，使得开发者能够更加快速、高效地构建现代化的Java应用程序。

## 2.2 Vue.js技术

Vue.js（通常简称为Vue）是一种流行的开源JavaScript框架，用于构建用户界面和单页应用程序（SPA）。以下是Vue.js技术的主要特点和优势：

1.轻量级： Vue.js是一款轻量级的框架，文件体积小，易于学习和上手。它采用了渐进式的设计思路，可以逐步引入到现有项目中，也可以作为单独的前端项目使用。

2.响应式： Vue.js采用了响应式数据绑定的机制，能够自动追踪数据的变化，并在数据发生变化时更新相应的视图。这种响应式的数据流机制使得开发者能够更加方便地管理和控制数据。

3.组件化： Vue.js采用了组件化的设计思路，将页面拆分成多个独立、可复用的组件。每个组件都有自己的状态、视图和行为，能够极大地提高代码的复用性和可维护性。

4.虚拟DOM： Vue.js采用了虚拟DOM技术，将页面的状态保存在内存中的虚拟DOM树中，通过比较虚拟DOM树的差异来更新实际的DOM结构，从而提高页面的渲染性能和效率。

5.生态系统： Vue.js拥有丰富的生态系统，包括Vue Router用于路由管理、Vuex用于状态管理、Vue CLI用于项目脚手架搭建等。同时，还有大量的第三方库和插件可以与Vue.js集成，为开发者提供更丰富的功能和工具。

6.社区支持： Vue.js拥有一个活跃的开发者社区，有着广泛的用户群和丰富的资源。开发者可以在社区中获取到大量的教程、文档、示例代码等，快速解决问题和学习新知识。

综上所述，Vue.js是一种功能强大、易于学习、灵活可靠的前端框架，适用于构建各种类型的现代化Web应用程序。它的响应式数据绑定、组件化设计、虚拟DOM技术等特点，使得开发者能够更加高效地构建出优质的用户界面和交互体验。

## 2.3 Java技术

Java技术是一种广泛应用于软件开发领域的高级编程语言和平台。以下是Java技术的主要特点和优势：

1.跨平台性： Java程序可以在不同的操作系统上运行，只需在目标平台上安装对应的Java虚拟机（JVM），使得Java具有很好的跨平台性。

2.面向对象： Java是一种面向对象的编程语言，支持类、继承、封装、多态等面向对象的特性，能够更好地组织和管理程序的结构和功能。

3.健壮性： Java具有良好的内存管理和异常处理机制，能够有效地防止程序的崩溃和内存泄漏等问题，提高了程序的健壮性和稳定性。

4.安全性： Java提供了安全管理机制，包括类加载器、安全管理器等，能够限制程序的访问权限，防止恶意代码的执行，提高了程序的安全性。

5.多线程支持： Java提供了多线程编程的支持，能够方便地创建和管理多个线程，实现并发执行和异步处理，提高了程序的性能和响应速度。

6.丰富的类库： Java拥有丰富的类库和工具包，涵盖了各种领域和功能，包括网络编程、图形界面、数据库连接、安全认证等，能够大大提高开发效率。

综上所述，Java技术具有跨平台性、面向对象、健壮性、安全性、多线程支持、丰富的类库等诸多优势，是一种非常适合于开发各种类型应用程序的编程语言。

## 2.4Thymeleaf技术

Thymeleaf是一种流行的模板引擎，用于在Web应用程序中生成动态的HTML内容。以下是Thymeleaf技术的主要特点和优势：

1.自然模板语法： Thymeleaf采用了自然的模板语法，能够让模板文件看起来像普通的HTML文件，具有良好的可读性和可维护性。开发者可以直接在HTML文件中使用Thymeleaf的模板语法，无需额外学习新的语法规则。

2.与HTML兼容： Thymeleaf模板可以直接在浏览器中打开并查看，而不需要进行预编译或其他特殊处理。这使得开发者可以直观地理解模板文件的结构和内容，并快速进行开发和调试。

3.模板片段： Thymeleaf支持模板片段的重用和组合，能够将重复的HTML代码抽取成片段，并在不同的页面中进行复用。这种模板的分离和组合方式，有助于提高代码的复用性和可维护性。

4.动态数据绑定： Thymeleaf支持动态数据绑定，能够将后端的数据模型与前端的HTML模板进行关联，实现动态数据的展示和更新。开发者可以通过简单的模板语法，在HTML模板中插入变量、表达式等，实现数据的动态渲染。

5.强大的功能扩展： Thymeleaf提供了丰富的标准和自定义功能扩展，包括条件判断、循环遍历、表单处理、国际化支持等，能够满足各种不同需求的模板处理场景。

6.与Spring集成： Thymeleaf与Spring框架无缝集成，可以直接在Spring应用程序中使用，并且能够充分利用Spring提供的各种特性和功能，如数据绑定、表单处理、国际化支持等。

总之，Thymeleaf是一种简单、灵活、功能强大的模板引擎，适用于构建各种类型的Web应用程序。它的自然模板语法、与HTML的兼容性、动态数据绑定等特点，使得开发者能够更加轻松地创建和维护动态的HTML内容，并提供良好的用户体验。

## 2.5 Element UI

Element UI是一套基于Vue.js的开源组件库，旨在提供丰富的用户界面组件和样式，以帮助开发者快速构建漂亮、功能丰富的Web应用程序。以下是Element UI的主要特点和优势：

1.丰富的组件库： Element UI提供了丰富的UI组件，涵盖了常见的按钮、表单、表格、弹窗、菜单、导航等各种元素，以及特殊功能组件如日期选择器、分页器、树形控件等，满足了各种不同场景和需求的开发。

2.易于使用： Element UI的组件设计简洁清晰，API易于理解和使用，开发者无需深入了解底层实现细节，即可快速上手使用各种组件，提高了开发效率。

3.样式丰富： Element UI提供了多种主题样式，包括默认主题、饿了么主题、自定义主题等，开发者可以根据自己的需求选择合适的样式风格，并且可以自定义主题以满足特定的设计要求。

4.响应式设计： Element UI的组件库采用了响应式设计，能够自适应不同的屏幕尺寸和设备类型，保证了用户在不同设备上的良好体验。

5.文档和示例丰富： Element UI提供了详细的官方文档和示例代码，涵盖了组件的使用方法、API文档、主题定制等内容，帮助开发者快速了解和使用组件。

6.活跃的社区支持： Element UI拥有一个活跃的开发者社区，有着广泛的用户群和丰富的资源，开发者可以在社区中获取到各种问题的解答、技术交流和资源分享。

## 2.5 jQuery

jQuery是一个流行的JavaScript库，设计用于简化HTML文档的遍历、操作、事件处理以及动画效果的库。以下是jQuery的主要特点和优势：

1.简化DOM操作： jQuery提供了简洁而强大的API，使得开发者可以更轻松地进行DOM元素的选取、遍历和操作，无需编写繁琐的原生JavaScript代码。

2.跨浏览器兼容性： jQuery解决了不同浏览器之间的兼容性问题，提供了统一的接口和行为，确保在各种现代浏览器中的稳定运行。

3.事件处理： jQuery提供了丰富的事件处理方法，使得开发者可以轻松地为DOM元素添加事件监听器，并进行事件的绑定、触发和处理，实现交互式的用户界面。

4.动画效果： jQuery支持各种动画效果和过渡效果，如淡入淡出、滑动、展开收起等，能够为页面添加丰富的视觉效果，提升用户体验。

5.AJAX支持： jQuery封装了常用的AJAX请求方法，使得开发者可以轻松地进行异步数据请求和处理，实现无需刷新页面的动态内容更新。

6.插件扩展： jQuery拥有庞大的插件生态系统，开发者可以通过使用各种插件来扩展jQuery的功能，满足不同项目的需求，提高开发效率。

7.轻量级： jQuery的代码量相对较小，压缩后文件体积较小，加载速度快，适合用于移动端和前端性能要求较高的项目。

jQuery是一个功能强大、易于学习和使用的JavaScript库，能够大大简化Web开发过程，提高开发效率，并为开发者提供丰富的工具和资源来构建交互式、动态的Web应用程序。

# **第3章** 系统分析与设计

## 3.1可行性分析

### 3.1.1技术可行性：

采用的技术都是当前流行的主流技术，具有较高的技术可行性。Java技术作为一种成熟稳定的后端开发语言，能够提供稳定高效的后端服务；Spring Boot框架简化了Java应用程序的开发和部署过程；Vue.js技术作为一种流行的前端框架，能够提供灵活高效的前端开发方式；Thymeleaf和Element UI提供了丰富的模板引擎和UI组件，能够实现动态的页面渲染和交互式用户界面。

### 3.1.2经济可行性：

1.低成本投入： 开发博客系统所需的软件系统和相关组件都可免费获取，开发成本相对较低。使用开源的开发工具和环境，如IDEA、Navicat、JDK、MySQL等，无需额外购买授权或付费许可费用。

2.收益来源多样： 博客系统可以通过多种方式获得收益，包括但不限于广告收入、会员付费、增值服务等。通过合作广告商进行广告投放，提供高级会员服务或增值功能，都可以为博客系统带来稳定的收益来源。

3.市场需求广泛： 随着互联网的普及和用户对个人信息分享的需求增加，博客系统具有广泛的市场需求。无论是个人博客、企业博客还是技术博客，都有一定的用户群体和市场空间，具有较好的市场前景。

4.运营成本可控： 由于博客系统的运营主要是在线服务和内容管理，相比传统实体业务，运营成本较低且可控。主要包括服务器托管费用、网站维护费用和运营人员成本等，可以通过合理管理和优化运营成本，提高盈利能力。

5.持续稳定收益： 博客系统作为一种长期稳定的网络应用程序，具有持续稳定的收益潜力。一旦建立了稳定的用户群体和内容积累，系统可以持续提供服务并获得持续稳定的收益，具有较好的投资回报率。

## 3.2系统需求分析

在博客系统的需求分析中，我们需要考虑前台用户的功能需求和后台管理员的管理需求。

### 3.2.1 智谷博客前台功能需求分析

1.登录注册功能： 用户可以进行账号的登录和注册操作，通过注册创建新的账号，通过登录进入系统。管理员可以通过首页登录进入系统的后台管理。

2.博客首页功能： 用户进入系统后直接进入博客首页，可以分页查看博文信息，包括博客的标题、摘要、发布时间、博客主图等。同时，用户可以浏览博客分类和推荐的博客资源。

3.问答功能： 用户可以在问答模块中提问和回答问题，参与社区的交流和讨论，对其他用户提出的问题进行回答和解答。

4.资讯功能： 用户可以在资讯模块中查看系统推荐的新闻、科技、健康等热门话题的资讯信息，了解最新的社会动态和行业资讯。

5.推荐功能： 在推荐板块，用户可以查看博主推荐的博客资源，并跳转到相应的路径进行浏览和阅读。

6.点赞功能：用户可以在浏览博文内容时，进行点赞操作。

### 3.2.1 智谷博客后台功能需求分析

1.首页模块： 后台管理首页通过ECharts技术展示系统的用户数量和博客浏览量等统计信息，帮助管理员了解系统的运行情况。

2.用户管理模块： 管理员可以对系统的注册用户进行管理，包括查看用户列表、删除用户、修改用户信息等操作，保证系统的用户管理功能正常运行。

3.分类管理模块： 管理员可以对博客的分类进行管理，包括添加新的分类、删除已有分类、修改分类信息等操作，确保博客的分类体系清晰和完整。

4.博客管理模块： 管理员可以对系统的博客进行统一管理，包括添加新的博客、编辑已有博客、删除博客、查询博客等操作，保证博客内容的质量和完整性。

5.问答管理模块： 管理员可以对问答内容进行统一管理，包括查看问答列表、查询问答内容、删除问答信息等操作，确保问答内容的规范和及时性。

6.资讯管理模块： 管理员可以对博客网站推荐的资讯进行统一管理，包括添加新的资讯、编辑已有资讯、删除资讯、查询资讯等操作，保证资讯内容的更新和准确性。

7.管理员个人信息模块：管理员可以对自己的个人信息进行修改，修改内容包括用户名、昵称、邮箱、电话、年龄、地区。

8.系统日志模块：管理员可以对系统的进行监控，可以了解用户的行为习惯和偏好，优化系统界面和功能设计，提升用户体验和满意度。

## 3.2系统总体设计

系统的架构图如图3-1所示。

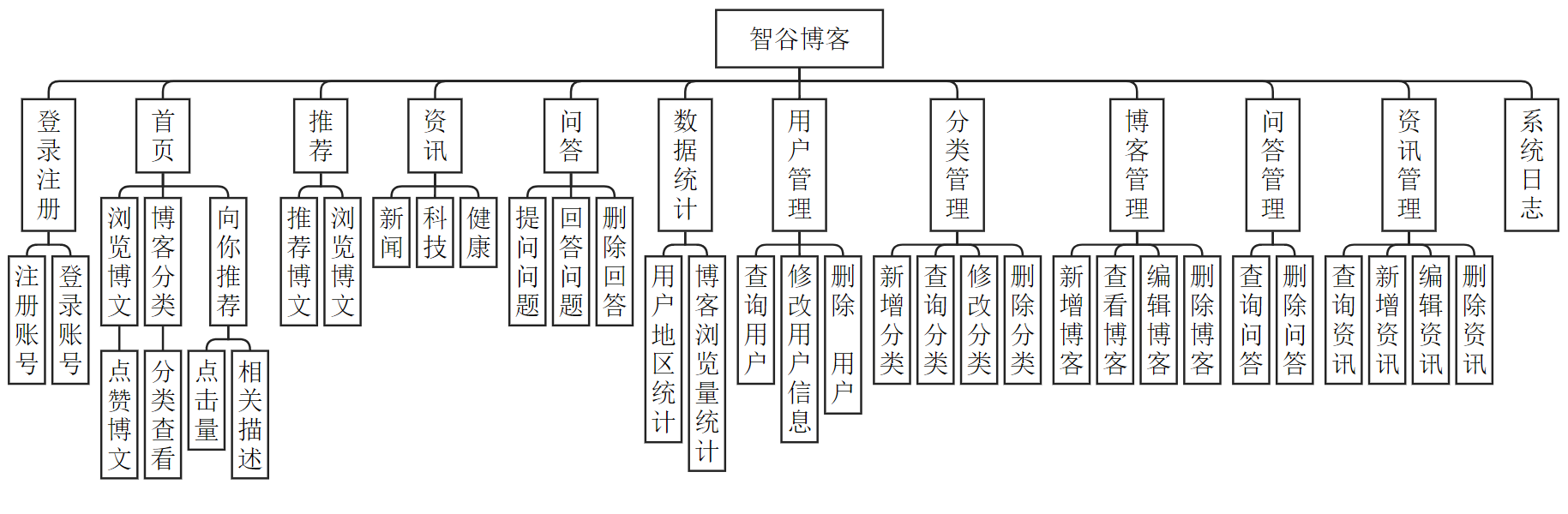


图3-1 系统总体架构图

## 3.3数据库设计

### 3.3.1数据库概念模型设计

本系统共有6个E-R实体属性图，分别为博客的实体属性图、博客分类的实体属性图、留言的实体属性图、用户的实体属性图、资源的实体属性图下面进行详细介绍：

1.博客的实体属性图如图3-2所示。

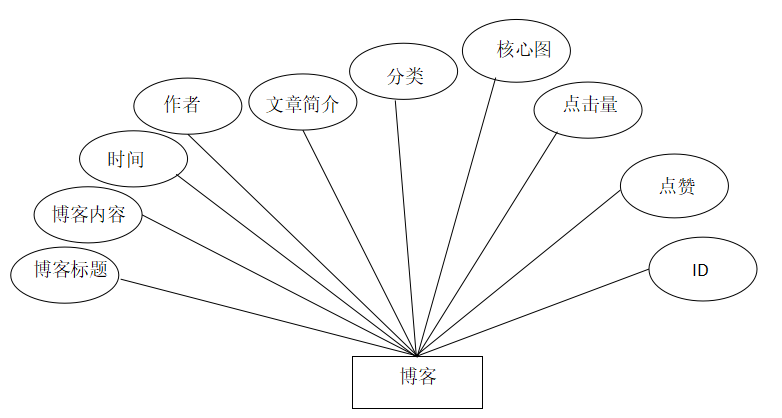


图3-2 博客实体属性图

2.博客的实体属性图如图3-3所示。

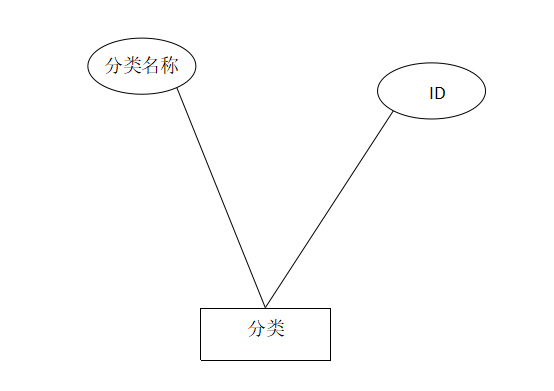


图3-3 博课分类实体属性图

3.资讯的实体属性图如图3-4所示。

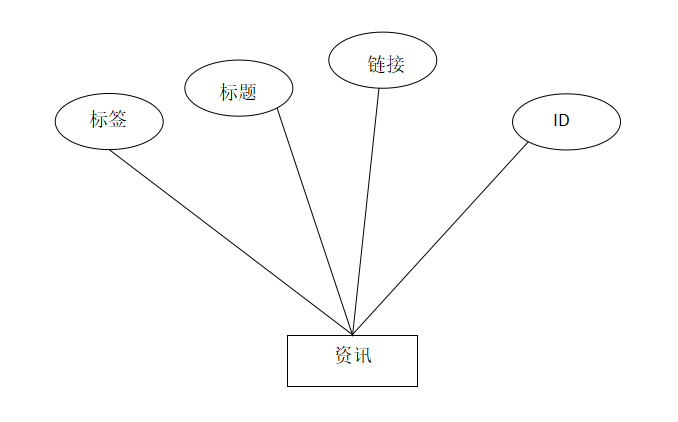


图3-4 资讯分类实体属性图

4.用户的实体属性图如图3-5所示。

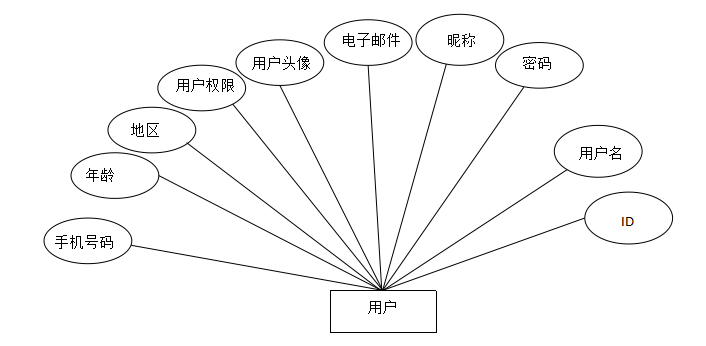


图3-5 用户实体属性图

5.问答实体属性图如图3-6所示。

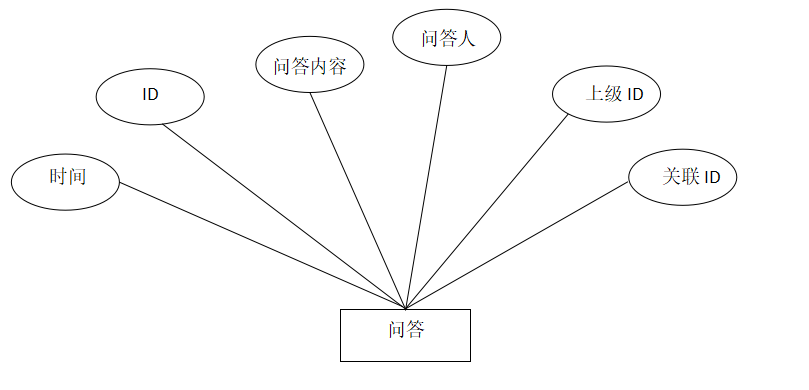


图3-6 问答实体属性图

6.系统日志实体属性图如图3-7所示。

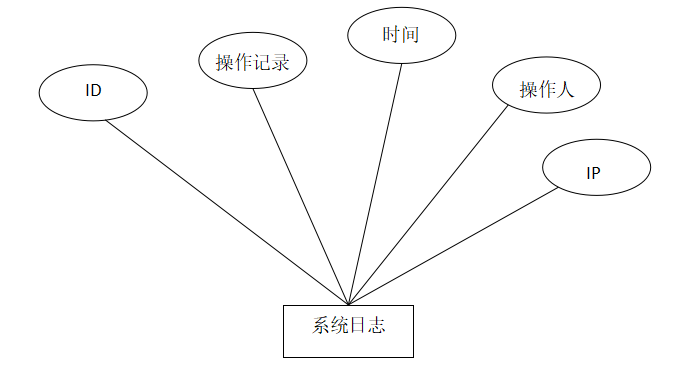


图3-7 系统日志实体属性图

### 3.3.2数据库表设计

表3-1 博客表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 名称 | 数据类型 | 非空 | 主键 |
| 1 | id | 序号 | bigint | 是 | 是 |
| 2 | title | 标题 | varchar(255) | 是 | 否 |
| 3 | content | 内容 | longtext | 是 | 否 |
| 4 | time | 时间 | varchar(255) | 是 | 否 |
| 5 | author | 作者 | varchar(255) | 是 | 否 |
| 6 | img | 核心图 | varchar(255) | 是 | 否 |
| 7 | description | 文章简介 | varchar(255) | 是 | 否 |
| 8 | tag | 分类 | varchar(255) | 是 | 否 |
| 9 | click | 点赞 | int | 是 | 否 |
| 10 | view | 访问量 | int | 是 | 否 |

表3-2 分类表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 名称 | 数据类型 | 非空 | 主键 |
| 1 | id | 主键 | bigint | 是 | 是 |
| 2 | name | 分类名称 | varchar(255) | 是 | 否 |

表3-3 资讯表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 名称 | 数据类型 | 非空 | 主键 |
| 1 | id | 主键 | bigint | 是 | 是 |
| 2 | name | 标题 | varchar | 是 | 否 |
| 3 | url | 链接 | varchar | 是 | 否 |
| 4 | type | 标签 | varchar | 是 | 否 |

表3-4 用户表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 名称 | 数据类型 | 非空 | 主键 |
| 1 | id | 主键 | bigint | 是 | 是 |
| 2 | username | 用户名 | varchar(255) | 是 | 否 |
| 3 | password | 密码 | varchar(255) | 是 | 否 |
| 4 | nick\_name | 昵称 | varchar(255) | 是 | 否 |
| 5 | email | 电子邮件 | varchar(255) | 是 | 否 |
| 6 | avatar | 头像 | varchar(255) | 是 | 否 |
| 7 | role | 权限 | varchar(255) | 是 | 否 |
| 8 | adress | 地址 | varchar(255) | 是 | 否 |
| 9 | age | 年龄 | int | 是 | 否 |
| 10 | phone | 手机号码 | varchar(255) | 是 | 否 |

表3-5 问答表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 名称 | 数据类型 | 非空 | 主键 |
| 1 | id | 主键 | bigint | 是 | 是 |
| 2 | content | 问答内容 | text | 是 | 否 |
| 3 | username | 问答人 | varchar(255) | 是 | 否 |
| 4 | time | 时间 | varchar(255) | 是 | 否 |
| 5 | parent\_id | 上级id | varchar(255) | 是 | 否 |
| 6 | foreign\_id | 关联id | varchar(255) | 是 | 否 |

表3-6 系统日志表结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 名称 | 数据类型 | 非空 | 主键 |
| 1 | id | 主键 | bigint | 是 | 是 |
| 2 | content | 操作记录 | varchar(1000) | 是 | 否 |
| 3 | time | 时间 | varchar(255) | 是 | 否 |
| 4 | user | 操作人 | varchar(255) | 是 | 否 |
| 5 | ip | ip | varchar(255) | 是 | 否 |

### 

# **第4章** 系统实现

## 4.1登录注册功能

登录功能：

注册用户可以通过登录功能进入系统。用户需要提供注册时使用的账号和密码进行登录。系统应该验证用户提供的账号和密码是否匹配，以确保用户身份的合法性。登录成功后，系统应该为用户分配相应的权限和权限级别，以便用户访问系统的其他功能和资源。



图4-1 登录界面展示图

注册功能：

用户可以通过注册功能创建新的账号，成为系统的注册用户。注册时需要提供必要的注册信息，包括账号、密码、电子邮件等。系统应该对用户提供的注册信息进行验证，确保注册信息的合法性和唯一性。



图4-2 注册界面展示图

代码如下：

代码4-1 登录注册代码示例

**public User getUser() {**

**User user = (User) request.getSession().getAttribute("user");**

**if (user == null) {**

**throw new CustomException("-1", "请登录");**

**}**

**return user;**

} **if (user == null) {**

**throw new CustomException("-1", "请登录");**

**}**

**return user;**

**}**

**@PostMapping("/login")**

**public Result<User> login(@RequestBody User user, HttpServletRequest request) {**

**User res = userService.login(user);**

**request.getSession().setAttribute("user", res);**

**MAP.put(res.getUsername(), res);**

**logService.log(StrUtil.format("用户 {} 登录系统", user.getUsername()));**

**return Result.success(res);**

**}**

**@PostMapping("/register")**

**public Result<User> register(@RequestBody User user, HttpServletRequest request) {**

**if (user.getPassword() == null) {**

**user.setPassword("123456");**

**}**

**User dbUser = userService.register(user);**

**request.getSession().setAttribute("user", user);**

**logService.log(StrUtil.format("用户 {} 注册账号成功", user.getUsername()));**

**return Result.success(dbUser);**

**}**

## 4.2首页

用户进入智谷博客的首页，可以浏览最新发布的博客文章。首页导航展示4个tab，智谷支持博客内容的分类和查询。以便用户最快能找到最具吸引力和相关性的博客内容，吸引用户进行浏览和阅读。

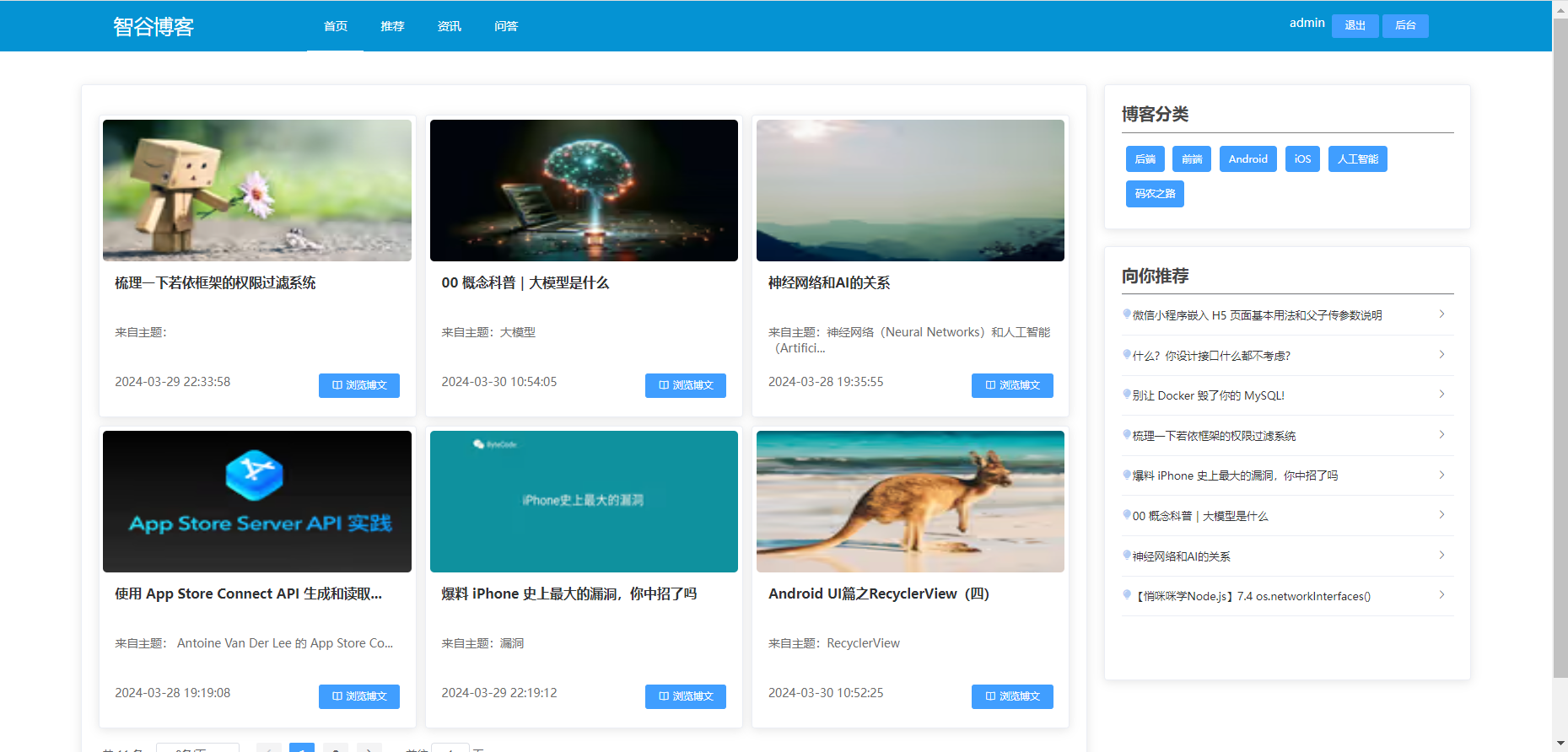


图4-3 首页界面展示图

代码如下：

代码4-2 首页代码示例

**@GetMapping("/page")**

**public Result<?> findPage(@RequestParam(required = false, defaultValue = "") String name,**

**@RequestParam(required = false, defaultValue = "") String tag,**

**@RequestParam(required = false, defaultValue = "1") Integer pageNum,**

**@RequestParam(required = false, defaultValue = "10") Integer pageSize) {**

**LambdaQueryWrapper<Blog> query = Wrappers.<Blog>lambdaQuery().orderByDesc(Blog::getId);**

**if (!StringUtils.isEmpty(name)) {**

**query.like(Blog::getTitle, name);**

**}**

**if (!StringUtils.isEmpty(tag)) {**

**query.eq(Blog::getTag, tag);**

**}**

**return Result.success(blogService.page(new Page<>(pageNum, pageSize), query));**

**}**

## 4.3问答功能

用户登录之后，可以选择问答tab，进入问答界面，用户可以在系统中提出问题，向其他用户或博主寻求帮助或解答。注册用户可以对系统中的问题进行回答和删除。

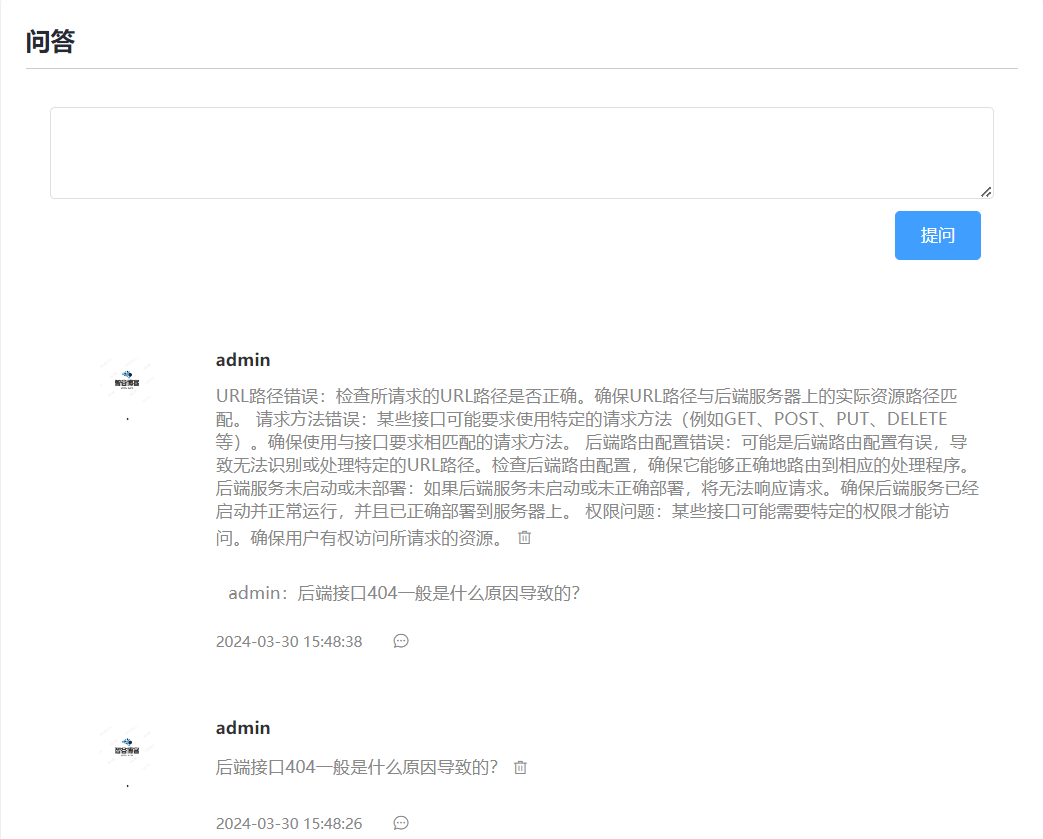


图4-4 问答界面展示图

代码如下：

**@GetMapping("/page")**

**public Result<?> findPage(@RequestParam(required = false, defaultValue = "") String name,**

**@RequestParam(required = false, defaultValue = "1") Integer pageNum,**

**@RequestParam(required = false, defaultValue = "10") Integer pageSize) {**

**LambdaQueryWrapper<Message> query = Wrappers.<Message>lambdaQuery().like(Message::getContent, name).orderByDesc(Message::getId);**

**return Result.success(messageService.page(new Page<>(pageNum, pageSize), query)); public User getUser() {**

**User user = (User) request.getSession().getAttribute("user");**

**if (user == null) {**

**throw new CustomException("-1", "请登录");**

**}**

**return user;**

**}**

**@PostMapping**

**public Result<?> save(@RequestBody Message Message) {**

**Message.setUsername(getUser().getUsername());**

**Message.setTime(DateUtil.formatDateTime(new Date()));**

**return Result.success(messageService.save(Message));**

**}**

代码4-3 问答代码示例

## 4.3资讯功能

系统提供三个板块，新闻，科技，健康。系统会分别展示新闻、科技、健康三个板块近期比较热点的资讯。点击按钮，即可跳转到对应资讯的页面。

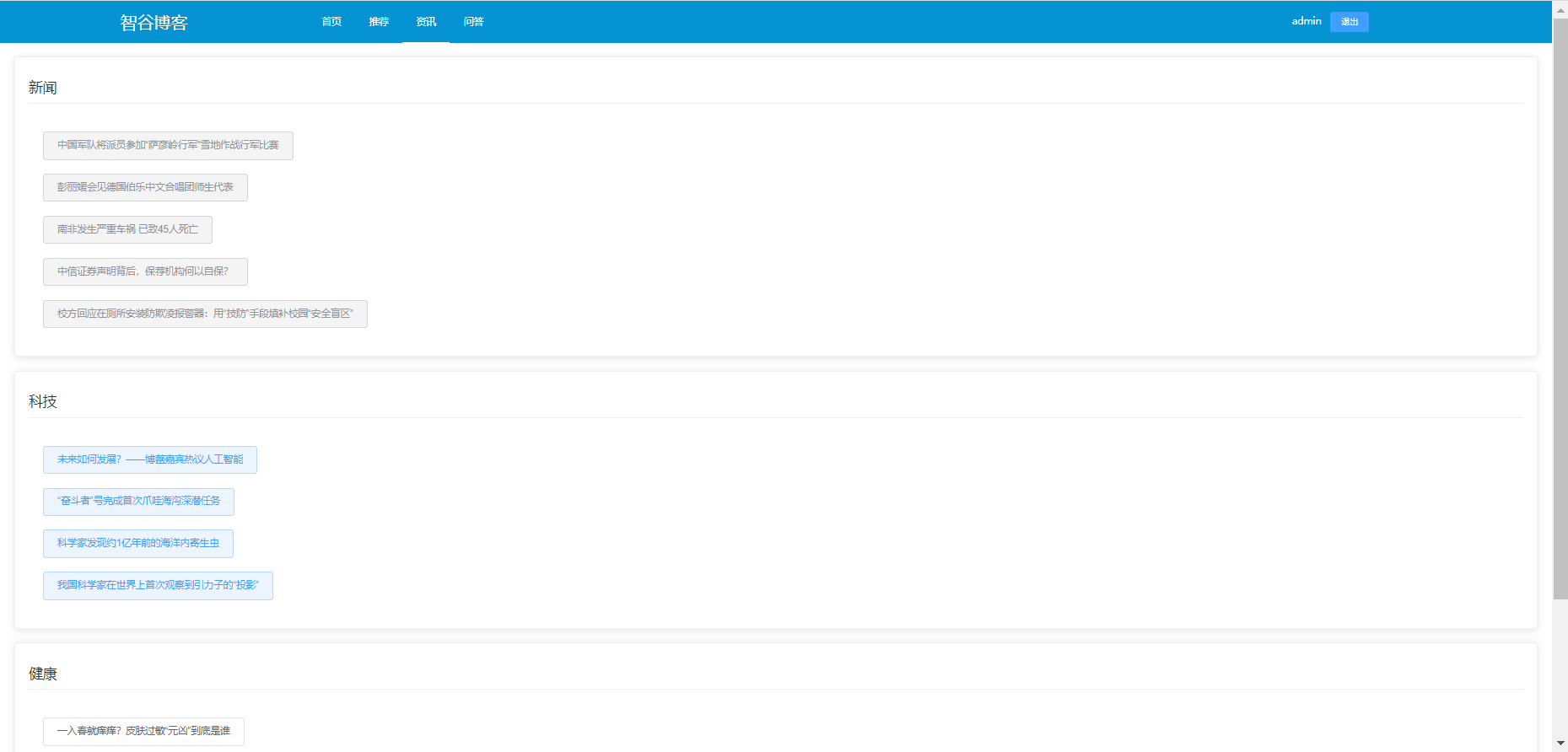


图4-5 资讯界面展示图

代码如下：

代码4-4 资讯代码示例

**@GetMapping("/page")**

**public Result<?> findPage(@RequestParam(required = false, defaultValue = "") String name,**

**@RequestParam(required = false, defaultValue = "1") Integer pageNum,**

**@RequestParam(required = false, defaultValue = "10") Integer pageSize) {**

**LambdaQueryWrapper<Link> query = Wrappers.<Link>lambdaQuery().orderByDesc(Link::getId);**

**if (StrUtil.isNotBlank(name)) {**

**query.like(Link::getName, name);**

**}**

**return Result.success(linkService.page(new Page<>(pageNum, pageSize), query));**

**}**

## 4.4退出功能

登出按钮：在用户登录状态下，智谷博客提供一个退出按钮，用户点击后即可执行退出操作。这个按钮位于用户个人信息区域。

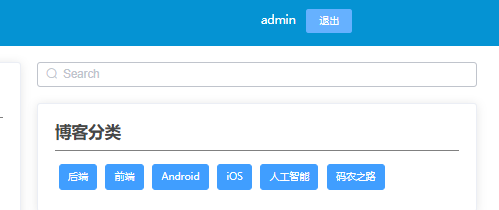


图4-6 退出界面展示图

代码如下：

**@GetMapping("/logout")**

**public Result<?> logout(HttpServletRequest request) {**

**User user = (User) request.getSession().getAttribute("user");**

**if (user != null) {**

**logService.log(StrUtil.format("用户 {} 退出系统", user.getUsername()));**

**request.getSession().removeAttribute("user");**

**MAP.remove(user.getUsername());**

**}**

**return Result.success();**

**}**

代码4-5 退出代码示例

## 4.5数据统计管理功能

管理员进入智谷博客后台可以地区统计和博客浏览量是两个重要的统计指标，收集用户访问博客的地理位置信息，统计不同地区的访问量。统计每篇博客文章的浏览量，了解用户对不同文章的关注程度。

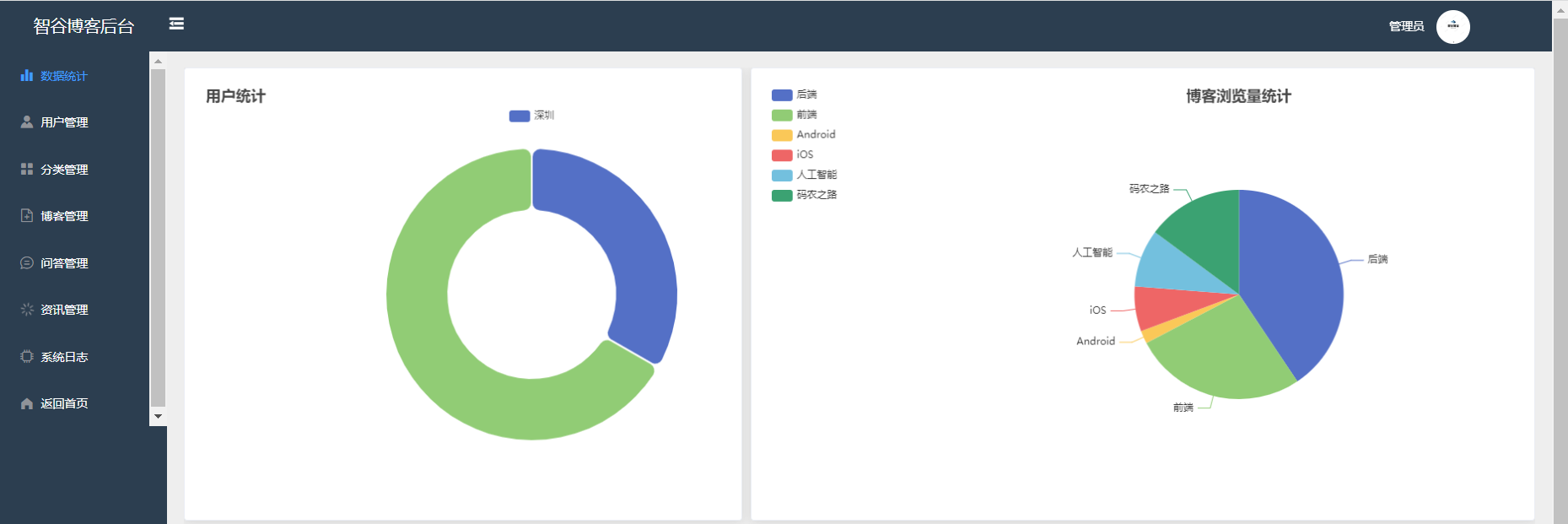


图4-7 数据统计界面展示图

代码如下：

**@GetMapping("/{id}")**

**public Result<User> findById(@PathVariable Long id) {**

**return Result.success(userService.getById(id));**

**}**

**@GetMapping("/detail/{username}")**

**public Result<User> findByUsername(@PathVariable String username) {**

**return Result.success(userService.getbyUsername(username));**

**}**

**@GetMapping**

**public Result<List<User>> findAll() {**

**return**

**Result.success(userService.list( Wrappers.<User>lambdaQuery().ne(User::getUsername, "admin")));**

**}**

**@GetMapping("/page")**

**public Result<IPage<User>> findPage(@RequestParam(required = false, defaultValue = "") String name,**

**@RequestParam(required = false, defaultValue = "1") Integer pageNum,**

**@RequestParam(required = false, defaultValue = "10") Integer pageSize) {**

**LambdaQueryWrapper<User> wrapper = Wrappers.<User>lambdaQuery().ne(User::getUsername, "admin").like(User::getUsername, name).orderByDesc(User::getId);**

**return Result.success(userService.page(new Page<>(pageNum, pageSize), wrapper));**

**} @DeleteMapping("/{id}")**

**public Result<?> delete(@PathVariable Long id) {**

**blogService.removeById(id);**

**return Result.success();**

**}**

**@GetMapping("/{id}")**

**public Result<?> findById(@PathVariable Long id) {**

**Blog blog = blogService.getById(id);**

**blog.setView(blog.getView() + 1);**

**blogService.updateById(blog);**

**return Result.success(blog);**

**}**

**@GetMapping("/carousel")**

**public Result<?> findCarousel() {**

**LambdaQueryWrapper<Blog> query = Wrappers.<Blog>lambdaQuery()**

**.like(Blog::getCarousel, "是")**

**.orderByDesc(Blog::getId);**

**return Result.success(blogService.page(new Page<>(1, 5), query));**

**}**

代码4-6 数据统计代码示例

## 4.6用户管理功能

用户管理界面提供查询功能，管理员可以根据用户名快速定位到特定用户。管理员可以编辑用户信息，包括用户名、昵称、电子邮件、手机号码、年龄，地区等基本信息。管理员可以删除用户信息，将不再需要的用户账号从系统中移除。用户管理模块的查询、编辑和删除功能有助于管理员对用户信息进行有效管理和维护，保证系统的正常运行和用户信息的安全性。



图4-8 用户管理界面展示图

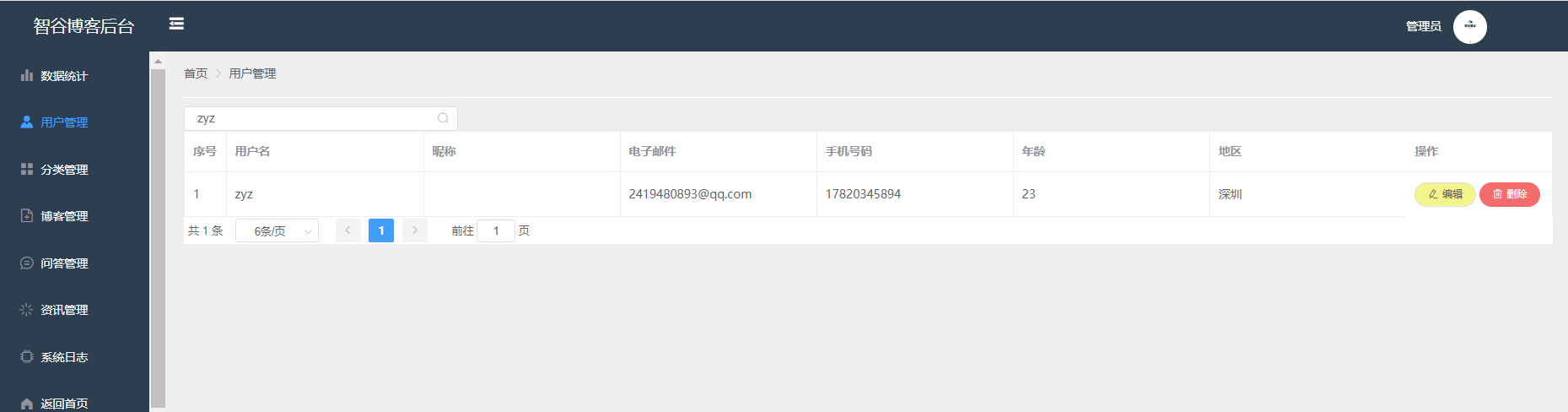


图4-8 用户管理查询界面展示图

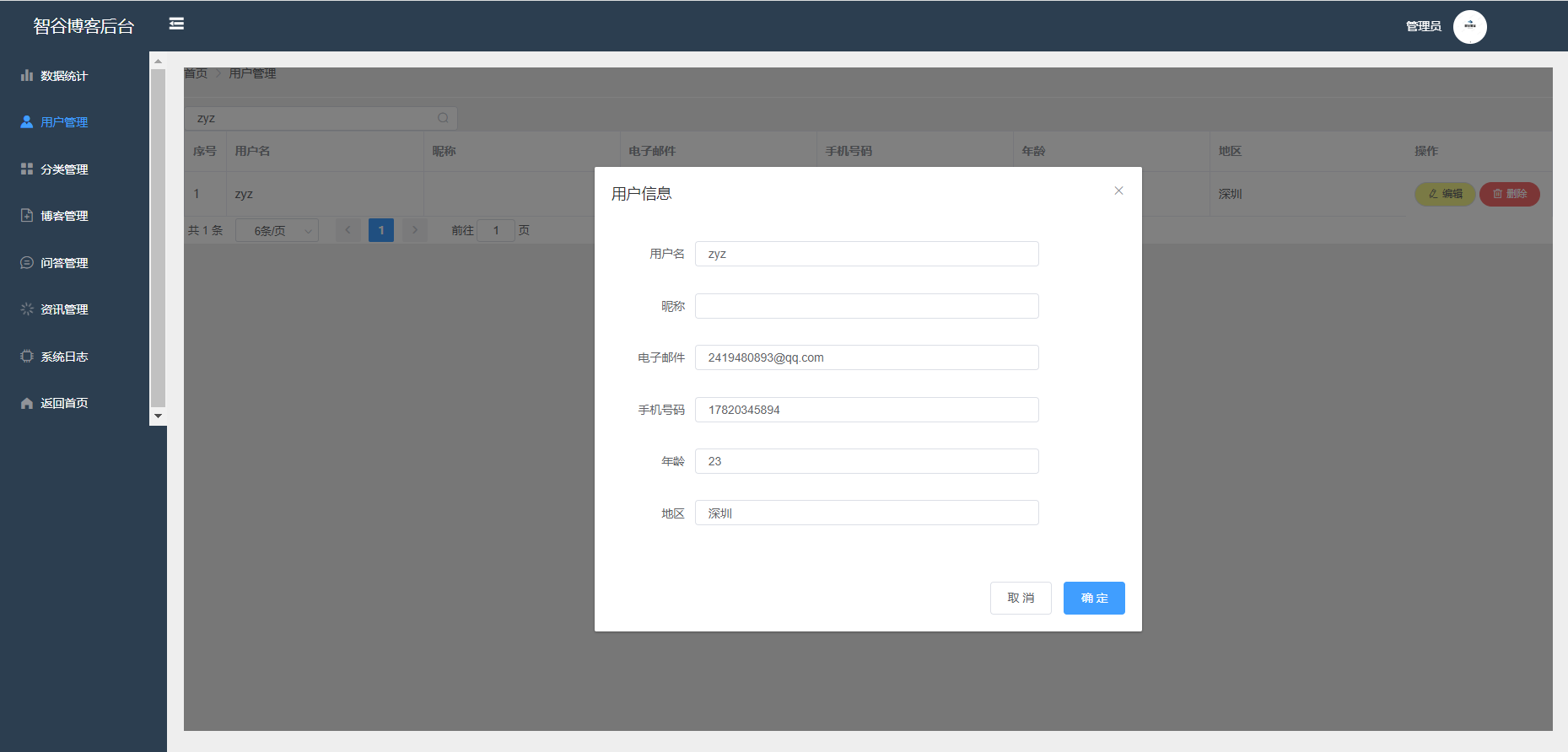


图4-9 用户管理编辑信息界面展示图

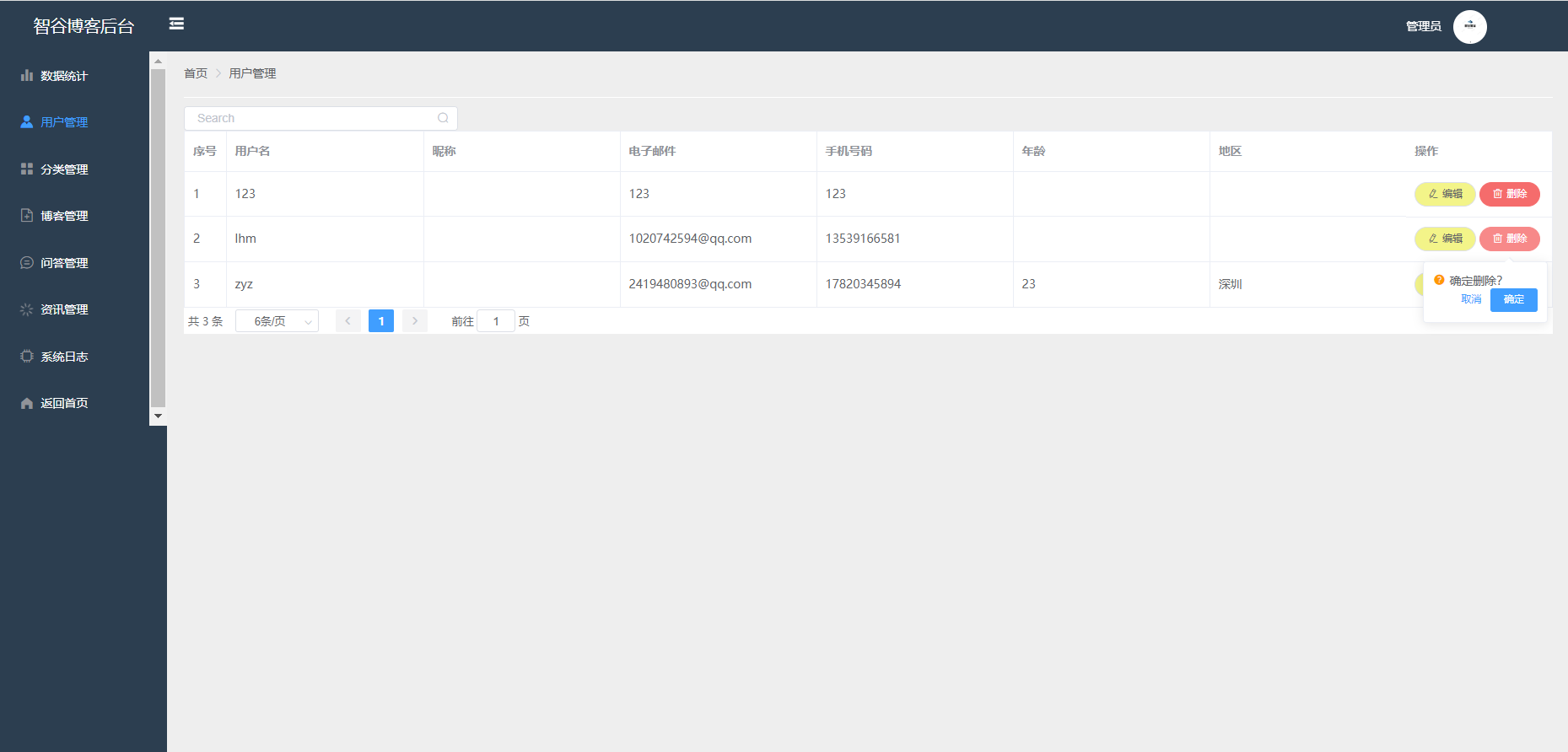


图4-10 用户管理删除界面展示图

代码如下：

**@GetMapping("/page")**

**public Result<IPage<User>> findPage(@RequestParam(required = false, defaultValue = "") String name,**

**@RequestParam(required = false, defaultValue = "1") Integer pageNum,**

**@RequestParam(required = false, defaultValue = "10") Integer pageSize) {**

**LambdaQueryWrapper<User> wrapper = Wrappers.<User>lambdaQuery().ne(User::getUsername, "admin").like(User::getUsername, name).orderByDesc(User::getId);**

**return Result.success(userService.page(new Page<>(pageNum, pageSize), wrapper));**

**}**

**public void export(HttpServletResponse response) throws IOException {**

**List<Map<String, Object>> list = CollUtil.newArrayList();**

**List<User> all = userService.list();**

**for (User user : all) {**

**Map<String, Object> row1 = new LinkedHashMap<>();**

**row1.put("名称", user.getUsername());**

**row1.put("手机", user.getPhone());**

**row1.put("邮箱", user.getEmail());**

**list.add(row1);**

**}**

**ExcelWriter writer = ExcelUtil.getWriter(true);**

**writer.write(list, true);**

**response.setContentType("application/vnd.openxmlformats-officedocument.spreadsheetml.sheet;charset=utf-8");**

**String fileName = URLEncoder.encode("用户信息", "UTF-8");**

**response.setHeader("Content-Disposition", "attachment;filename=" + fileName + ".xlsx");**

**ServletOutputStream out = response.getOutputStream();**

**writer.flush(out, true);**

**writer.close();**

**IoUtil.close(System.out);}**

**}**

代码4-7 用户管理首页代码示例

**@PostMapping**

**public Result<?> save(@RequestBody User user) {**

**if (user.getPassword() == null) {**

**user.setPassword("123456");**

**}**

**logService.log(StrUtil.format("新增用户：{} ", user.getUsername()));**

**return Result.success(userService.save(user));**

**}**

**@PutMapping**

**public Result<?> update(@RequestBody User user) {**

**logService.log(StrUtil.format("更新用户：{} ", user.getUsername()));**

**return Result.success(userService.updateById(user));**

**}**

**public Result<?> upload(@PathVariable String fileId) {**

**String basePath = System.getProperty("user.dir") + "/src/main/resources/static/file/";**

**List<String> fileNames = FileUtil.listFileNames(basePath);**

**String file = fileNames.stream().filter(name -> name.contains(fileId)).findAny().orElse("");**

**List<List<Object>> lists = ExcelUtil.getReader(basePath + file).read(1);**

**List<User> saveList = new ArrayList<>();**

**for (List<Object> row : lists) {**

**User user = new User();**

**user.setUsername((String) row.get(0));**

**user.setNickName((String) row.get(1));**

**user.setEmail((String) row.get(2));**

**user.setPhone((String) row.get(3));**

**saveList.add(user);**

**}**

**userService.saveBatch(saveList);**

**return Result.success();**

**}**

代码4-8 编辑用户信息代码示例

@**DeleteMapping("/{id}")**

**public Result<?> delete(@PathVariable Long id) {**

**User user = userService.getById(id);**

**logService.log(StrUtil.format("删除用户 {} ", user.getUsername()));**

**userService.removeById(id);**

**return Result.success();**

**}**

代码4-9 删除用户信息代码示例

## 4.7分类管理功能

分类管理功能是管理博客文章分类的重要模块，允许管理员查询，新增，删除，编辑分类。

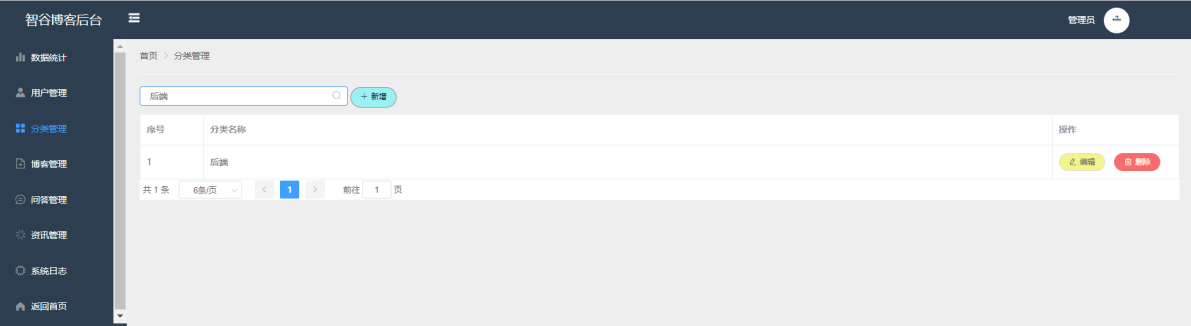


图4-11 分类查询界面展示图



图4-12 分类新增界面展示图



图4-13 编辑界面展示图

图4-14 删除分类界面展示图

代码如下：

代码4-10 编辑分类代码示例

**@PostMapping**

**public Result<?> save(@RequestBody Tag tag) {**

**return Result.success(tagService.save(tag));**

**}**

**@PutMapping**

**public Result<?> update(@RequestBody Tag tag) {**

**return Result.success(tagService.updateById(tag));**

**}**

**@GetMapping("/{id}")**

**public Result<?> findById(@PathVariable Long id) {**

**return Result.success(tagService.getById(id));**

**}**

**@DeleteMapping("/{id}")**

**public Result<?> delete(@PathVariable Long id) {**

**tagService.removeById(id);**

**return Result.success();**

**}**

**@GetMapping("/{id}")**

**public Result<?> findById(@PathVariable Long id) {**

**return Result.success(tagService.getById(id));**

**}**

代码4-11 删除分类代码示例

**@GetMapping("/page")**

**public Result<?> findPage(@RequestParam(required = false, defaultValue = "") String name,**

**@RequestParam(required = false, defaultValue = "1") Integer pageNum,**

**@RequestParam(required = false, defaultValue = "10") Integer pageSize) {**

**LambdaQueryWrapper<Tag> query = Wrappers.<Tag>lambdaQuery().orderByAsc(Tag::getId);**

**if (StrUtil.isNotBlank(name)) {**

**query.like(Tag::getName, name);**

**}**

**return Result.success(tagService.page(new Page<>(pageNum, pageSize), query));**

**}**

代码4-12 查询分类信息代码示例

## 4.8博客管理功能

新增博客：管理员可以通过博客管理界面新增新的博客文章。新增博客时，管理员需要填写博客的标题、内容、分类等相关信息，并可以上传主图。新增文章成功后，系统会生成文章的唯一标识并保存到数据库中，以便后续管理和展示。

编辑博客文章：已存在的博客文章可以进行编辑操作，管理员可以修改博客的标题、内容、分类等信息。编辑操作允许管理员对文章进行修订、更新或者修改，以保持博客内容的及时性和准确性。

查看博客文章：管理员可以在博客管理界面查看已发布的博客列表。

查看博客文章时，系统会展示文章的标题、分类、发布时间等基本信息，以便管理员浏览和管理。删除博客文章：管理员可以删除不再需要的博客文章，将其从系统中移除。删除文章前需要管理员确认操作，以避免误删或数据丢失。

删除文章后，系统会将博客浏览记录等相关数据一并删除或者进行处理。

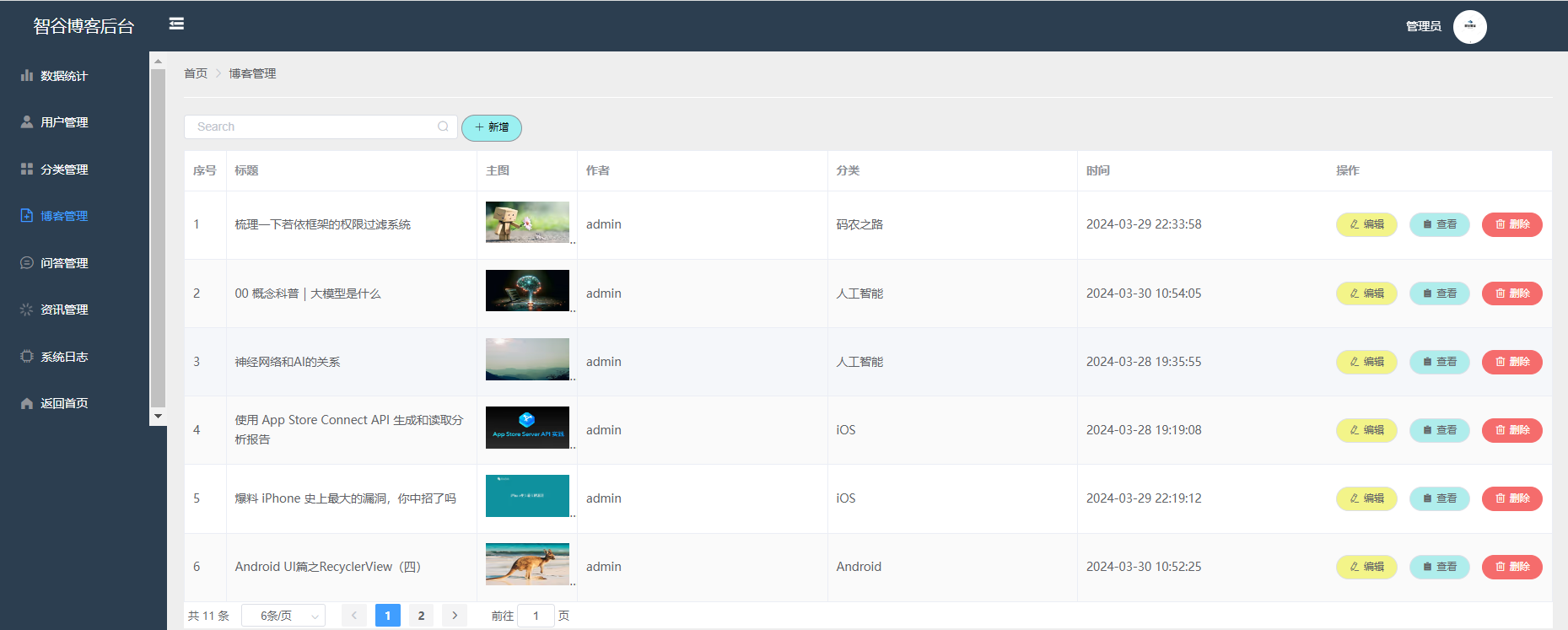


图4-15 博客管理界面展示图

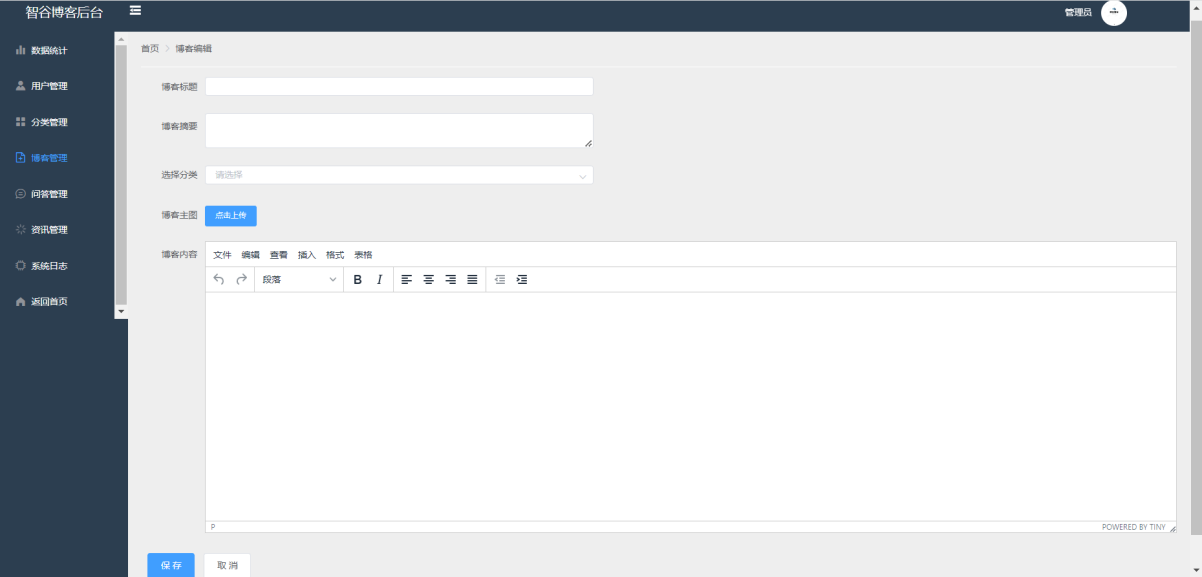


图4-16 新增博客界面展示图

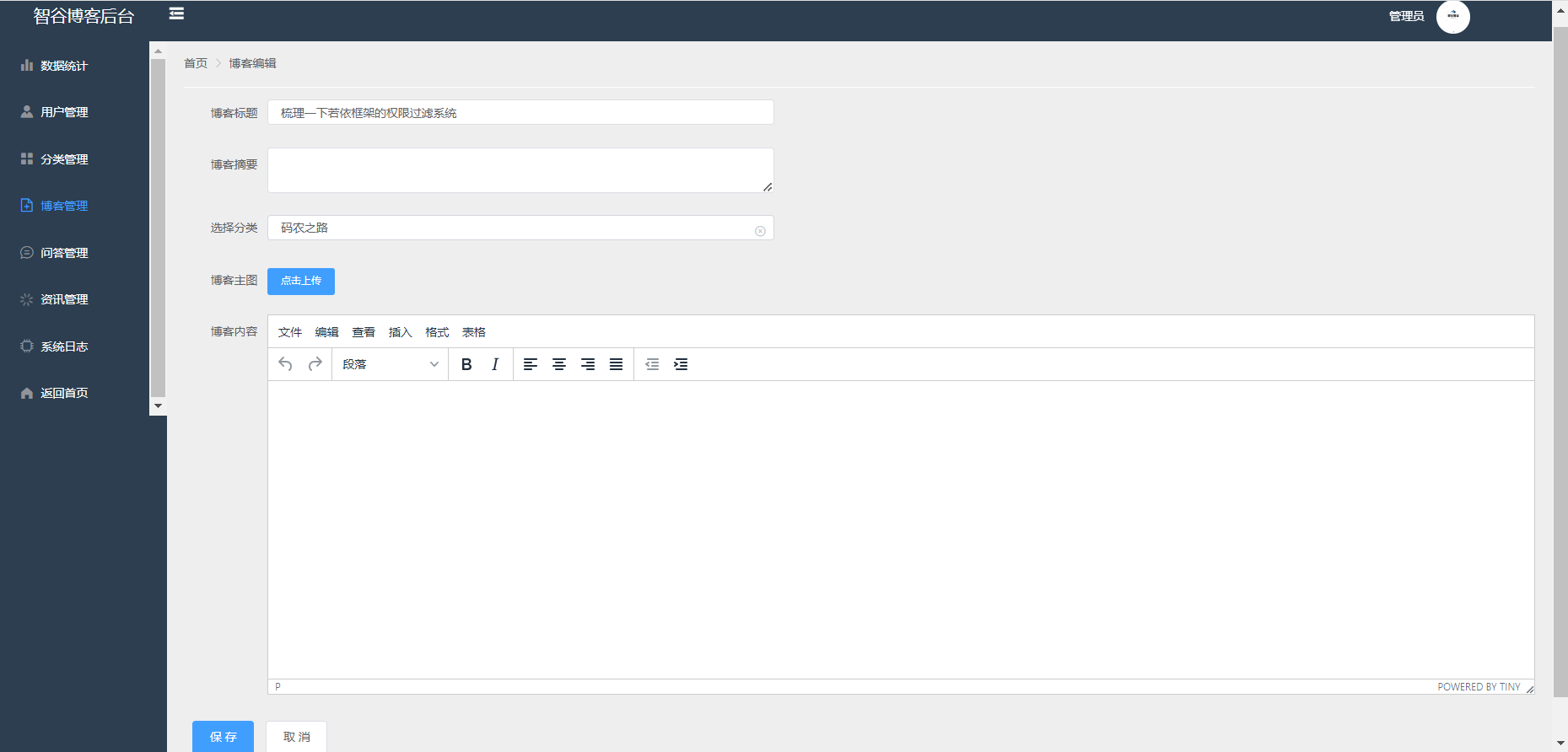


图4-17 编辑博客界面展示图



图4-18 查看博客界面展示图

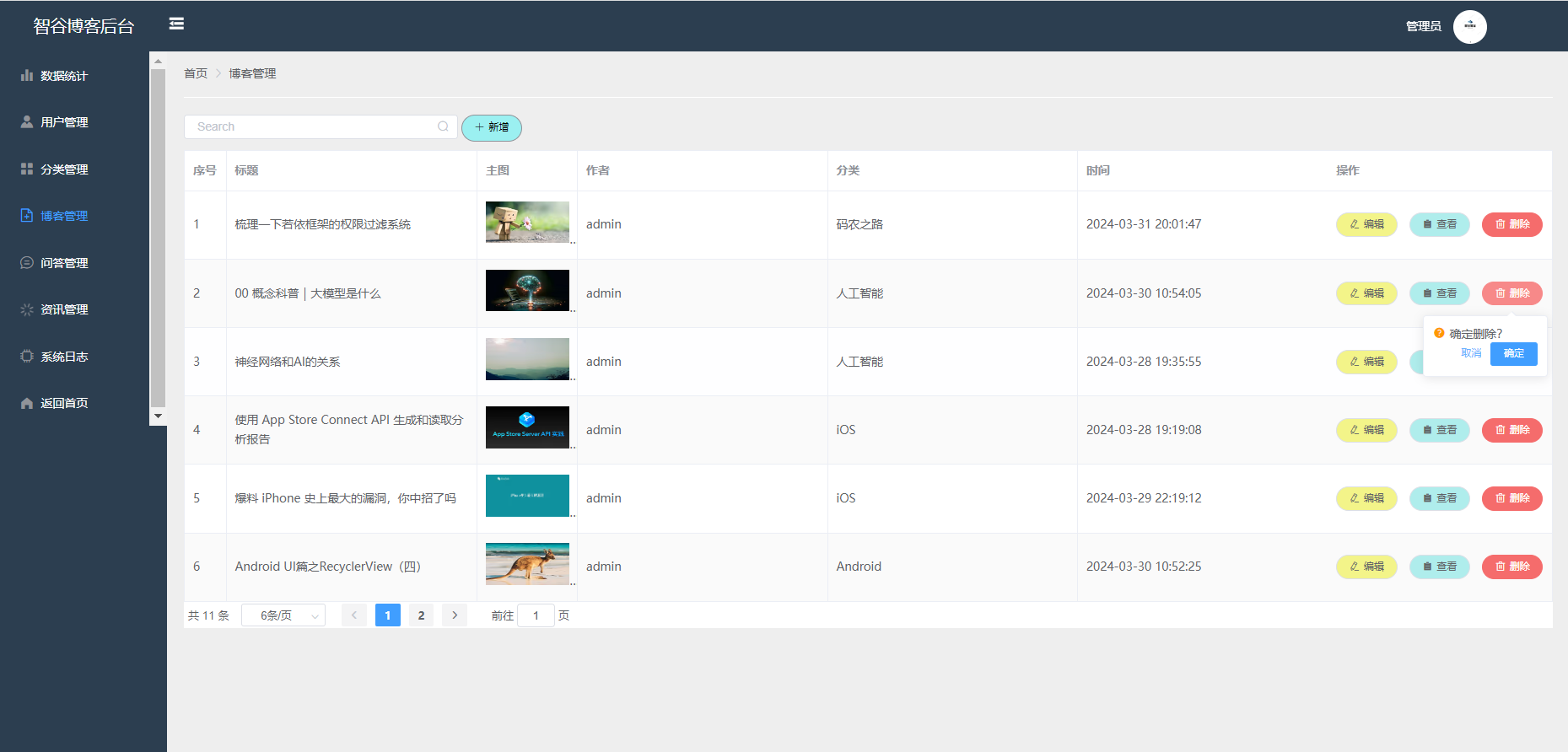


图4-19 删除博客界面展示图

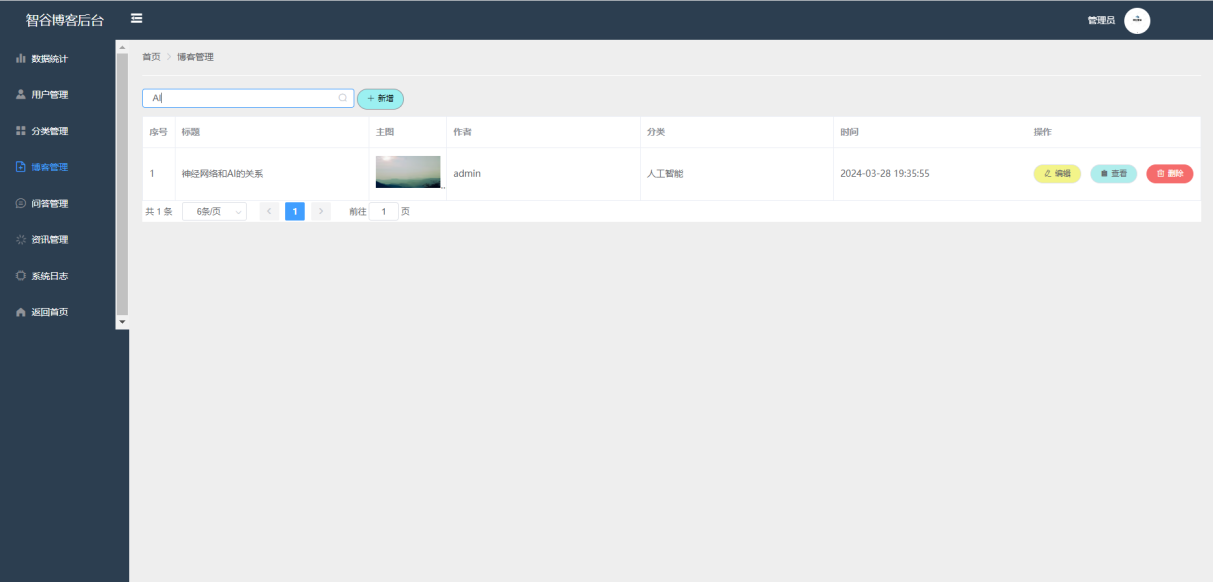


图4-20 查询博客界面展示图

代码如下：

**@PostMapping**

**public Result<?> save(@RequestBody Blog blog) {**

**User user = getUser();**

**blog.setAuthor(user.getUsername());**

**blog.setTime(new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date()));**

**return Result.success(blogService.save(blog));**

**}**

**@PutMapping**

**public Result<?> update(@RequestBody Blog blog) {**

**blog.setTime(new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date()));**

**return Result.success(blogService.updateById(blog));**

**}**

代码4-13 编辑博客代码示例

**@GetMapping("/{id}")**

**public Result<?> findById(@PathVariable Long id) {**

**Blog blog = blogService.getById(id);**

**blog.setView(blog.getView() + 1);**

**blogService.updateById(blog);**

**return Result.success(blog);**

**}**

代码4-14 查看博客代码示例

代码4-15 删除博客代码示例

**@GetMapping("/{id}")**

**public Result<?> findById(@PathVariable Long id) {**

**Blog blog = blogService.getById(id);**

**blog.setView(blog.getView() + 1);**

**blogService.updateById(blog);**

**return Result.success(blog);**

**}**

代码4-16 查询博客代码示例

**@GetMapping("/page")**

**public Result<?> findPage(@RequestParam(required = false, defaultValue = "") String name,**

**@RequestParam(required = false, defaultValue = "") String tag,**

**@RequestParam(required = false, defaultValue = "1") Integer pageNum,**

**@RequestParam(required = false, defaultValue = "10") Integer pageSize) {**

**LambdaQueryWrapper<Blog> query = Wrappers.<Blog>lambdaQuery().orderByDesc(Blog::getId);**

**if (!StringUtils.isEmpty(name)) {**

**query.like(Blog::getTitle, name);**

**}**

**if (!StringUtils.isEmpty(tag)) {**

**query.eq(Blog::getTag, tag);**

**}**

**return Result.success(blogService.page(new Page<>(pageNum, pageSize), query));**

**}**

## 4.9问答管理功能

问答管理功能可以查询，删除问答。



图4-20 问答管理界面展示图



图4-21 删除问答界面展示图

**@GetMapping("/page")**

**public Result<?> findPage(@RequestParam(required = false, defaultValue = "") String name,**

**@RequestParam(required = false, defaultValue = "1") Integer pageNum,**

**@RequestParam(required = false, defaultValue = "10") Integer pageSize) {**

**LambdaQueryWrapper<Message> query = Wrappers.<Message>lambdaQuery().like(Message::getContent, name).orderByDesc(Message::getId);**

**return Result.success(messageService.page(new Page<>(pageNum, pageSize), query));**

**}**

代码4-17查询问答代码示例

**@DeleteMapping("/{id}")**

**public Result<?> delete(@PathVariable Long id) {**

**messageService.removeById(id);**

**return Result.success();**

**}**

**@GetMapping("/{id}")**

**public Result<?> findById(@PathVariable Long id) {**

**return Result.success(messageService.getById(id));**

**}**

代码4-18 删除问答功能代码示例

## 4.10资讯管理

在资讯管理模块，可以对资讯进行查询，新增，编辑，删除操作。

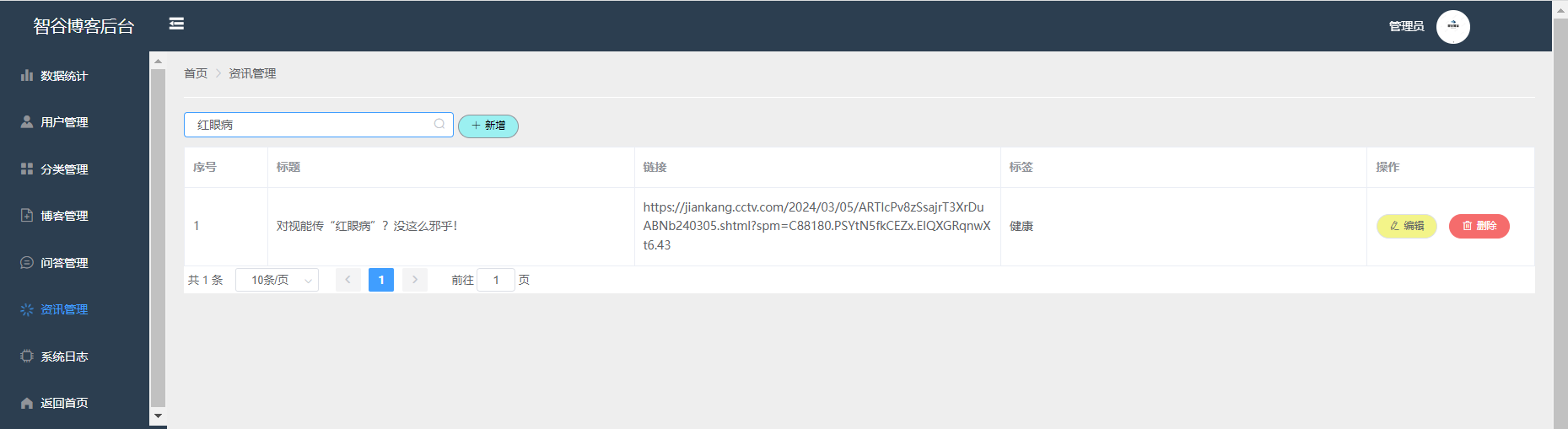


图4-22 查询资讯界面展示图

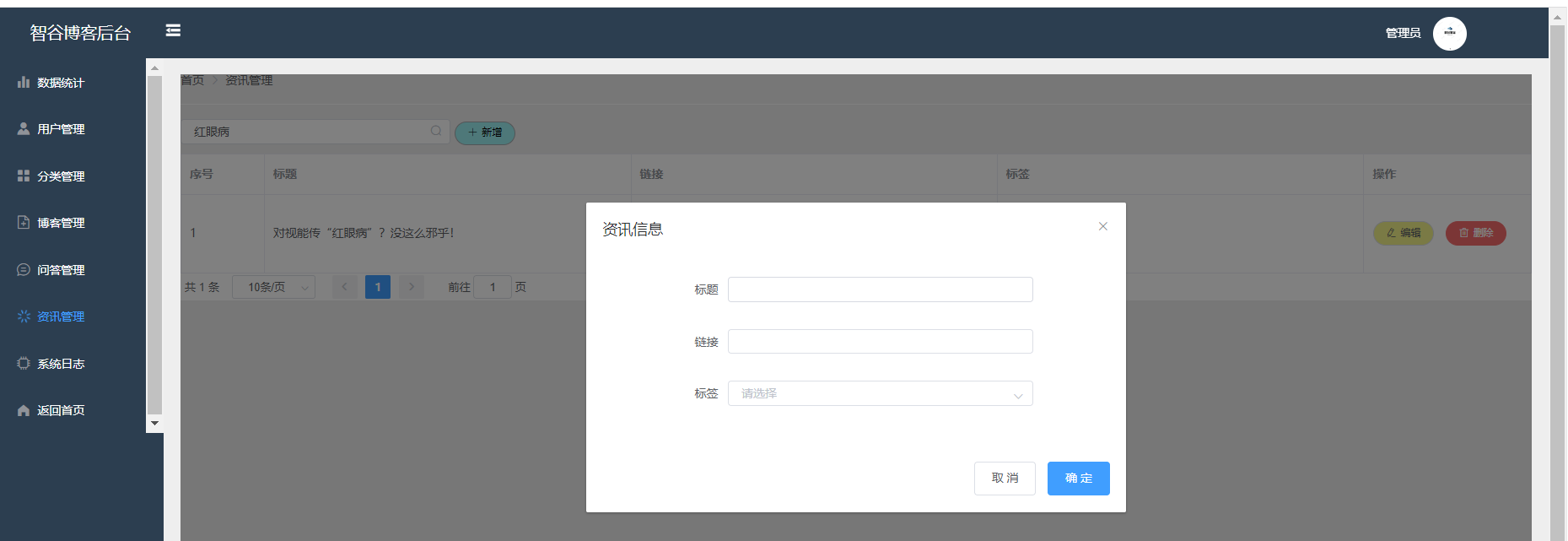


图4-23 新增资讯界面展示图



图4-23 编辑资讯界面展示图

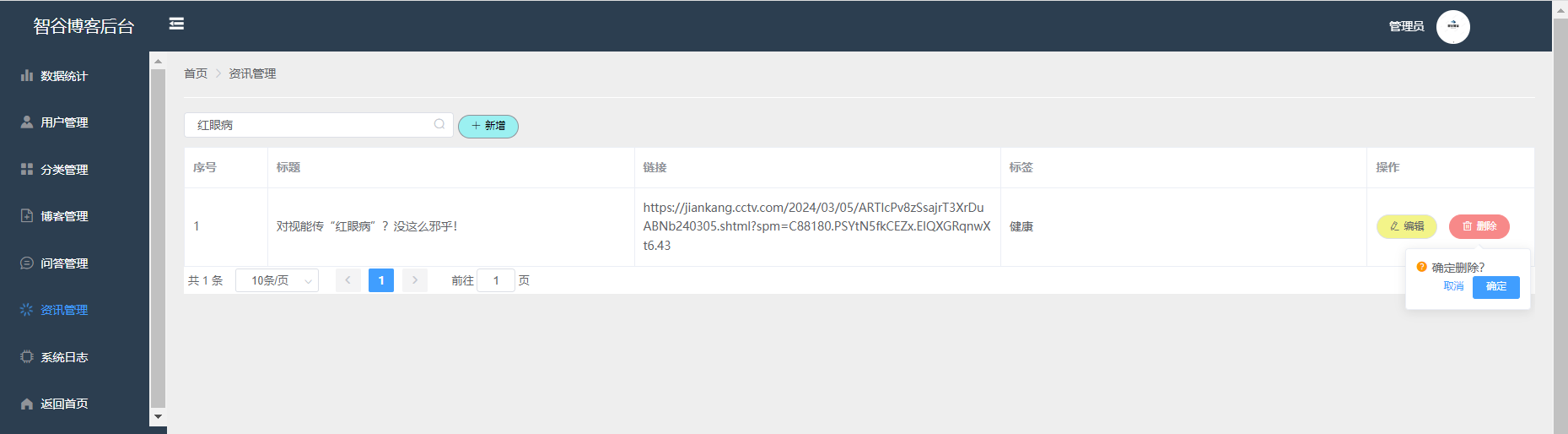


图4-24 删除资讯界面展示图

代码如下：

代码4-19 查询资讯功能代码示例

**@GetMapping("/page")**

**public Result<?> findPage(@RequestParam(required = false, defaultValue = "") String name,**

**@RequestParam(required = false, defaultValue = "1") Integer pageNum,**

**@RequestParam(required = false, defaultValue = "10") Integer pageSize) {**

**LambdaQueryWrapper<Link> query = Wrappers.<Link>lambdaQuery().orderByDesc(Link::getId);**

**if (StrUtil.isNotBlank(name)) {**

**query.like(Link::getName, name);**

**}**

**return Result.success(linkService.page(new Page<>(pageNum, pageSize), query));**

**}**

代码4-20 编辑资讯功能代码示例

**@PostMapping**

**public Result<?> save(@RequestBody Link link) {**

**return Result.success(linkService.save(link));**

**}**

**@PutMapping**

**public Result<?> update(@RequestBody Link link) {**

**return Result.success(linkService.updateById(link));**

**} @GetMapping("/{id}")**

**public Result<?> findById(@PathVariable Long id) {**

**return Result.success(linkService.getById(id));**

**}**

代码4-21 删除资讯功能代码示例

**@DeleteMapping("/{id}")**

**public Result<?> delete(@PathVariable Long id) {**

**linkService.removeById(id);**

**return Result.success();**

**}**

## 4.11系统日志功能

系统日志模块可以查询系统日志，每条日志包含了操作内容、操作时间、操作人、以及操作的IP地址等关键信息。

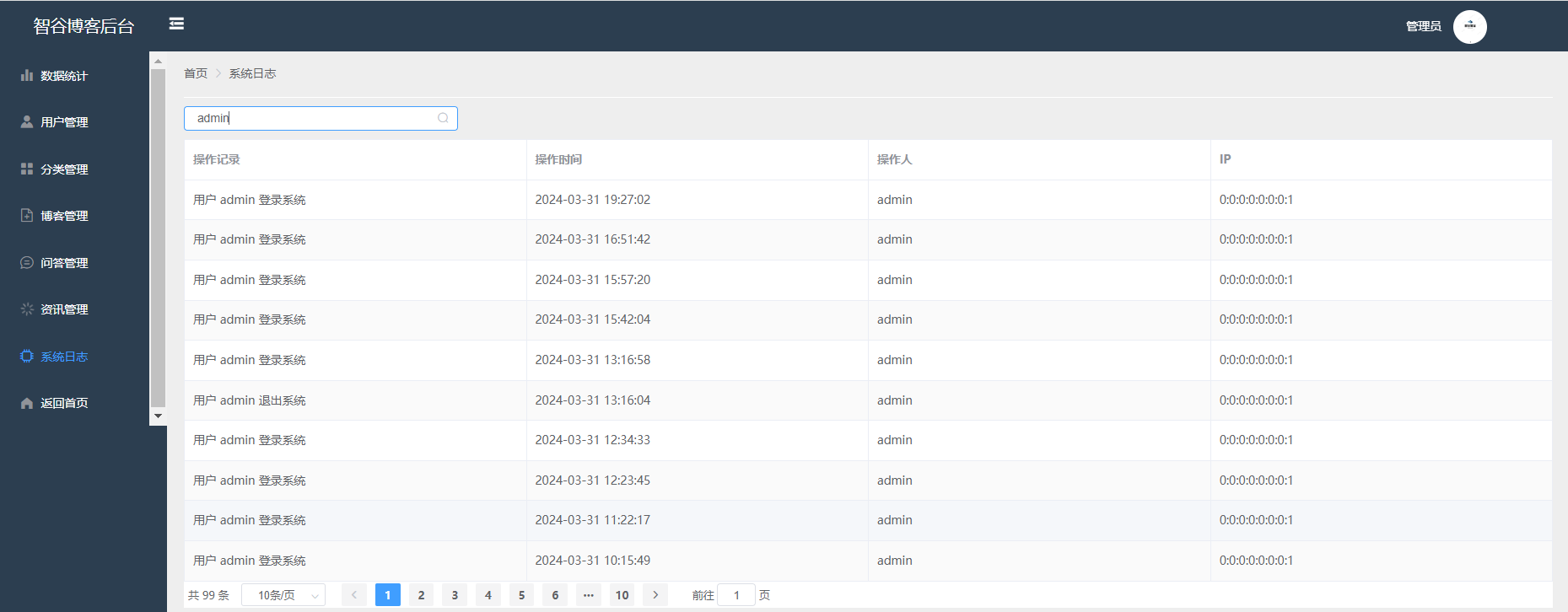


图4-24 查询日志界面展示图

代码4-22 查询日志代码功能模块

**@GetMapping("/page")**

**public Result<?> findPage(@RequestParam(required = false, defaultValue = "") String name,**

**@RequestParam(required = false, defaultValue = "1") Integer pageNum,**

**@RequestParam(required = false, defaultValue = "10") Integer pageSize) {**

**LambdaQueryWrapper<Log> query = Wrappers.<Log>lambdaQuery().orderByDesc(Log::getId);**

**if (StrUtil.isNotBlank(name)) {**

**query.like(Log::getContent, name);**

**}**

**return Result.success(logService.page(new Page<>(pageNum, pageSize), query));**

**}**

# **第5章 系统测试**

## 5.1概述

本次测试旨在验证智谷博客系统各个模块的功能是否符合预期，以保证系统的稳定性、可靠性和用户体验。

## 5.2测试目标

1.验证登录注册功能的正确性和安全性。

2.确保首页、问答、资讯等功能页面的展示和跳转正常。

3.验证用户管理、分类管理、博客管理等后台管理功能的有效性。

4.确保数据统计、系统日志等管理功能的准确性和完整性。

5.确认退出功能的有效性。

## 5.3测试环境

操作系统：Windows 11

浏览器：Chrome、Firefox、Edge

设备：PC

## 5.4测试用例

表5-1 测试用例表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模块 | 功能点 | 用例 | 预期结果 | 优先级 | 测试结果 |
| 登录注册功能 | 登录 | 输入正确的账号和密码，点击登录按钮 | 登录成功 | P0 | Pass |
| 输入错误的账号或密码，点击登录按钮 | 登录失败，提示用账号或密码错误 | P1 | Pass |
| 首页 | 展示博客文章 | 访问首页 | 展示最新的博客文章 | P0 | Pass |
| 查看博客详情 | 点击博客文章标题 | 跳转至对应的博客详情页面 | P0 | Pass |
| 点击分类标签 | 点击分类标签 | 跳转至对应的分类页面 | P0 | Pass |

表5-2 测试用例表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模块 | 功能点 | 用例 | 预期结果 | 优先级 | 测试结果 |
| 问答功能 | 展示问答内容 | 访问问答界面 | 展示最新的问答内容 | P0 | Pass |
| 发布问题 | 输入问题内容，点击提交按钮 | 成功发布问题 | P0 | Pass |
| 回答问题 | 输入回答内容，点击提交按钮 | 成功发布回答 | P0 | Pass |
| 资讯功能 | 展示资讯内容 | 访问资讯页面 | 展示最新的资讯内容 | P0 | Pass |
| 查看资讯详情 | 点击资讯标题 | 跳转至对应的资讯详情页面 | P0 | Pass |
| 退出功能 | 退出登录 | 点击退出按钮 | 成功退出当前登录状态 | P0 | Pass |
| 数据统计管理功能 | 查看统计信息 | 访问数据统计页面 | 展示最新的用户数量和博客浏览量统计信息 | P0 | Pass |
| 查看详细数据 | 点击查看数据图像 | 展示详细的统计图表 | P0 | Pass |
| 用户管理功能 | 展示用户列表 | 访问用户管理页面 | 展示所有注册用户的列表 | P0 | Pass |
| 搜索用户 | 输入关键词进行搜索 | 展示符合条件的用户列表 | P0 | Pass |
| 编辑用户信息 | 点击编辑用户按钮 | 跳转至编辑用户页面 | P0 | Pass |
| 删除用户 | 点击删除用户按钮 | 成功删除用户 | P0 | Pass |
| 分类管理功能 | 展示分类列表 | 访问分类管理页面 | 展示所有分类的列表 | P0 | Pass |
| 搜索分类 | 输入关键词进行搜索 | 展示符合条件的分类列表 | P0 | Pass |
| 编辑分类信息 | 点击编辑分类按钮 | 跳转至编辑分类页面 | P0 | Pass |

表5-3 测试用例表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 模块 | 功能点 | 用例 | 预期结果 | 优先级 | 测试结果 |
| 分类管理功能 | 删除分类 | 点击删除分类按钮 | 成功删除分类 | P0 | Pass |
| 新增分类 | 点击新增按钮 | 新增分类成功 | P0 | Pass |
| 博客管理功能 | 展示博客列表 | 访问博客管理页面 | 展示所有博客文章的列表 | P0 | Pass |
| 删除问答 | 点击删除按钮 | 成功删除问问答 | P0 | Pass |
| 搜索博客 | 输入关键词进行搜索 | 展示符合条件的博客列表 | P0 | Pass |
| 新增博客 | 点击新增按钮 | 成功新增博客 | P0 | Pass |
| 删除博客 | 点击删除博客按钮 | 成功删除博客 | P0 | Pass |
| 问答管理功能 | 展示问答列表 | 访问问答管理页面 | 展示所有问答内容的列表 | P0 | Pass |
| 搜索问答 | 输入关键词进行搜索 | 展示符合条件的问答列表 | P0 | Pass |
| 删除问答 | 点击删除按钮 | 成功删除问问答 | P0 | Pass |
| 资讯管理功能 | 展示资讯列表 | 访问资讯管理页面 | 展示所有资讯内容的列表 | P0 | Pass |
| 搜索资讯 | 输入关键词进行搜索 | 展示符合条件的资讯列表 | P0 | Pass |
| 新增资讯 | 点击新增按钮 | 成功新增资讯 | P0 | Pass |
| 编辑资讯信息 | 点击编辑资讯按钮 | 跳转至编辑资讯页面 | P0 | Pass |
| 删除资讯 | 点击删除资讯按钮 | 成功删除资讯 | P0 | Pass |
| 系统日志功能 | 查看日志列表 | 访问系统日志页面 | 展示所有日志 | P0 | Pass |
| 搜索日志 | 输入关键词进行搜索 | 展示符合条件的日志列表 | P0 | Pass |

# 第6章 总结与展望

6.1总结

本项目是一个基于Spring Boot和Vue.js的博客系统，旨在为用户提供一个方便、高效的博客发布和浏览平台。在项目开发过程中，我们充分利用了Spring Boot和Vue.js等先进的技术，结合了MySQL数据库和Element UI等组件库，实现了用户管理、博客发布、问答、数据统计等多个功能。通过本次项目的开发，提高了我对Spring Boot和Vue.js等技术的理解和应用能力。同时也让我对软件的开发流程有了更深层次的理解。这次项目的开发让我明白了一个道理，在软件开发这条路上，最重要的是坚持学习。勤学善思，不断打磨技术。提高我的专业技术水平。

6.2展望

未来，我们将继续完善博客系统的功能，优化用户体验，提升系统的性能和安全性。同时，我们也将考虑引入更多的技术和功能模块，如音乐社区、直播、商城、邮件通知等功能，使得系统更加丰富和完善。通过持续的用户调研和反馈收集，不断改进和优化系统，以满足用户的需求和期待。同时，我们也将继续关注新技术的发展和应用，不断提升团队的技术实力和创新能力，为用户提供更优质的产品和服务。

# 致 谢

在完成这个项目的过程中，我想要向许许多多的人表示由衷的感谢。首先我要感谢我的指导老师，在项目过程中给予我耐心的指导和宝贵的建议。你们的专业知识和经验对我有着深远的影响，让我在项目中能够不断成长和进步。同时，我要感谢我的朋友和家人，在我遇到困难和挑战时给予我鼓励和支持，你们的陪伴和理解让我能够坚持到最后并取得成功。最后再次感谢大家的支持和帮助，祝愿我们未来的道路越走越宽广，取得更加辉煌的成绩！

# 参考文献

[1] 高洪岩．至简SSH：精通Java Web 实用开发技术[M]．电子工业出版社，2009．

[2] OUJP，SOONG T T，et al.Recent advance in research on applications of passive energy

 dissipation systems [J].Earthquack Eng,1997，38(3):358-361.

[3]

[4]

[5]

[6]

[7]

[8]

[9]

[10]