

西南石油大学

本科毕业设计（论文）



S

题 目 融创富文本音视频集合的

文档社区平台的设计与实现

学生姓名 李世娇 学 号 202131782239

教学院系 计算机与软件学院

专业年级 数字媒体技术 2021 级

指导教师 任兆刚 职 称 教授

单 位 西南石油大学

辅导教师 职 称

单 位

完成日期 2024 年 5 月 25 日

Southwest Petroleum University

Graduation Thesis



Design and Implementation of an Integrated and Innovative Rich Text, Audio, and Video Document Community Platform

Grade: 2021

Name: Li Shijiao

Speciality: Digital Media Technology

Instructor: Ren Zhaogang

School of Computer Science and Software Engineering

2025-5-25

摘要

随着互联网技术的飞速发展，信息传播与共享的方式日益多元化，传统文档社区平台已经难以满足用户对丰富信息的需求。因此，设计一个融合了富文本、音视频的创新型文档社区平台，具有重要的现实意义。本次设计的主要内容是，实现一个功能完备、界面友好的文档社区平台。系统主要划分为用户前台功能部分、管理后台功能部分。用户前台功能部分主要包含了以下的几个业务板块：用户登录注册模块、用户个人中心模块、用户综合发布模块、用户协同发布模块。管理后台功能部分主要包含了以下的几个业务板块：用户信息管理模块、综合发布管理模块、协同发布管理模块、系统公告管理模块、系统日志管理模块。

项目系统将提供前后端分离的架构，采用的是以 Vue 和 ElementUI 为前端框架，Spring Boot 和 Mybatis 为后端框架的开发方案。系统基础功能包括用户登录注册、个人信息管理；此外还支持富文本编辑、音视频播放等高级功能，以满足用户对文档处理的多样化需求；且每项业务板块根据需要合理布局，便于用户快速定位和使用。未来研究应加强技术融合与创新，注重用户体验提升，进一步开拓发展平台的应用范围，以满足人们对于多样化信息日益增长的交流与共享需求。

关键词：Spring Boot；Vue；富文本；音视频；文档社区平台

Abstract

With the rapid development of Internet technology, the ways of information dissemination and sharing have become increasingly diversified. Traditional document community platforms can no longer meet users' needs for rich information. Therefore, designing an innovative document community platform that integrates rich text and audio-visual content has important practical significance. The main content of this design is to implement a fully functional and user-friendly document community platform. The system is mainly divided into two parts: the user front-end functionality and the management back-end functionality. The user front-end functionality mainly includes the following business modules: user login and registration module, user personal center module, user comprehensive publishing module, and user collaborative publishing module. The management back-end functionality mainly includes the following business modules: user information management module, comprehensive publishing management module, collaborative publishing management module, system announcement management module, and system log management module.

The project system will adopt a front-end and back-end separation architecture, using Vue and ElementUI as the front-end frameworks, and Spring Boot and Mybatis as the back-end frameworks. The basic functions of the system include supporting user login and registration, personal information management, as well as advanced features such as rich text editing and audio-visual playback, to meet users' diverse needs for document processing. Moreover, each business module is reasonably laid out according to its needs, facilitating quick user access and utilization. Future research should focus on strengthening technological integration and innovation, paying more attention to user experience enhancement, and further expanding the application scope of the platform to meet the growing needs of people for diverse information exchange and sharing.

Key words: Spring Boot; Vue; Rich Text; Audio-Visual; Document Community Platform;

目 录

1 绪 论	1
1.1 课题的背景、目的及意义	1
1.2 发展现状	2
1.3 论文各章简介	2
2 可行性分析及技术介绍	4
2.1 技术可行性	4
2.2 经济可行性	4
2.3 操作可行性	4
2.4 相关技术介绍	4
2.4.1 Spring Boot.....	4
2.4.2 MySQL	5
2.4.3 MyBatis	6
2.4.4 前端技术	6
3 总体设计	8
3.1 系统功能模块	8
3.2 数据库设计	9
4 详细设计	17
4.1 用户前台功能部分	17
4.1.1 用户登录注册模块	19
4.1.2 用户个人中心模块	21
4.1.3 用户自定义分类模块	22
4.1.4 用户自定义上传模块	23
4.1.5 用户富文本编辑模块	24
4.1.6 多用户协同编辑模块	26
4.2 管理后台功能部分	30
4.2.1 用户信息管理模块	30
4.2.2 用户发布管理模块	31
4.2.3 系统日志管理模块	32
4.2.4 系统公告管理模块	33

5 系统测试	35
5.1 测试方案	35
5.2 测试设备及环境	35
5.3 功能测试	36
5.3.1 登录注册模块	36
5.3.2 个人中心模块	36
5.3.3 用户发布模块	37
5.3.4 协同编辑模块	37
5.3.5 后台管理模块	38
6 结论	39
谢 辞	40
参考文献	41

1 绪论

1.1 课题的背景、目的及意义

随着互联网技术的飞速发展，信息传播与共同分享的方式日益多元化。传统的文档社区平台大多以纯文本内容为主，难以满足用户对丰富多样信息的需求。然而富文本、音视频等多媒体元素的融合平台，则能够为用户提供更加直观、生动、全面的信息体验，通过捕获人们的感官感受，增强信息的吸引力和传播效果。

本研究旨在设计并实现一个功能全面、操作便捷的融合创新富文本音视频集合的文档社区平台。该平台将打破传统文档社区的局限，不再局限于单一的文本内容，而是将富文本编辑的强大功能与音视频的生动表现力完美融合，为用户提供一个新的信息创作、知识分享与传播交流的方式。^[16]

社区平台或者说论坛，就是一个为大家展现自己想法，进行创作和交流互动的空间，想象一般的社区平台，常规的分类有常规的分类有文章类社区平台如知乎，视频类社区平台如 B 站，或者其他分类包括音乐创作平台例如网易云，如图片类创作平台例如花瓣，这些社区平台的共同目标就是，为用户提供创作空间——即发布自己的作品；创作灵感——即浏览他人的作品；以及交流互动的平台。而这些社区平台的共同特点就是，为各自的主体内容打造一个专属的共享空间：看视频就到视频网站、看小说就到小说网站、听歌就到音乐网站，它们都是为各自的内容选择了一种专属的展现形式，而这既是平台的优势也是局限。像广为人知的新浪微博，则集合了图文、视频更加丰富的内容表现形式，由于能够展示更加多样化形式的内容和优秀的互动体验从而收获了庞大的用户群体。由此可知，社区平台是否能吸引用户的关键，就在于能否灵活展现用户的想法，落实到开发上则在于能否处理不同格式的文件内容，用于展示，甚至用于制定自定义、个性化的编辑方案。

网络社区本就是一个及其重要的学习交流的平台，一旦平台为用户提供了适宜的表达的窗口，就会有源源不断的用户这个加入社区、就会有源源不断的用户发表自己的观点，并且进一步促进平台功能的完善，这将为知识的传播起到了极大的良性循环作用。考虑到开发人员是一个天天需要和各种软件、技术文档打交道的用户群体，他们对于网络学习交流的需求是极大的，由于专业的特殊性，普通的文章社区平台根本无法满足广大技术人员学习交流的需求，因此构建一个适宜的社区平台很有必要。由于专业的原因，我需要经常在网站跟看视频教程、搜索问答等，并从别人的经验从中收获许多。从本人的上网经验来看，由于学习任务的需要，有时要

访问各种网站、跳转各种链接，辗转于各大网盘之间才能下载到需要的文件。于是我希望能够构建一个文档社区平台，不仅能通过视频进行操作演示、口头讲解等，还可以通过图文内容解释进行详细的书面展示，更重要的是不必跳转第三方，可以直接在页面上获取到关键文件，以此能够方便各种教学需要、创作需要。于是就诞生了创作本项目平台的思路，本文档社区平台不仅要求实现多种文件格式的预览方案，还能为用户提供直接获取源文件的下载渠道，方便用户二次创作。

1.2 发展现状

我国互联网内容社区的发展历程已有二十余年，根据中国互联网络信息中心（CNNIC）的发布，第 55 次《中国互联网络发展状况统计报告》中显示，截至 2024 年 12 月，中国网民规模达到 11.08 亿人，互联网普及率升至 78.6%。互联网内容社区发展的内驱力是用户对于多元化信息元素的需求增强，用户对于参与信息创作、信息传播和互动的渴望增加。互联网提供的技术和搭建的模块使得用户养成深度使用的习惯，用户对内容创作、传播和互动需求的提升，推动社区平台走向繁荣。

文档社区平台的开发目的是对个人知识库进行总结梳理，以及与他人进行知识交流分享。协同编辑的出现为用户对文档处理的协作需求增加了新的解决方案。协同编辑技术的发展最早可以追溯到 20 世纪后期，当时的应用场景则局限于少数人共同的编辑。随着互联网技术的更新与工具库的完善，当今时代的协同编辑技术已经得到了普遍地应用和创新性的发展，在个人、团队、企业的工作学习使用中成为得力的帮手。协同编辑旨在通过实施协作，允许多人同时对同一个编辑器的内容进行保存更改，而不发生数据污染与合并冲突。随着多人办公、在线创作的普遍应用和快速发展，在线协同编辑技术已经从最初的 WebSocket，发展为支持 OT、CRDT 等高级同步机制，并辅以更多其他的现代技术以提升时效与稳定性。

协同编辑，一种新兴的协同技术，正成为支撑团队协作重要组成部分，并且已经在在线办公、文档协同、多人内容创作等领域中得到广泛的应用。未来，文档社区平台辅以协同编辑将促进协同编辑技术不断发展，新的技术将与之结合，继续创造更便捷的编辑方案和协作环境，进一步努力促进团队之间的知识传播、技术交流。

1.3 论文各章简介

这次编写的论文包含了 6 个部分的内容，主要内容如下：

第 1 章主要内容是：背景、目的意义及发展现状。

第 2 章主要内容是：可行性分析，从技术上、经济上、操作上进行可行性分析并对相关技术的进行介绍。

第 3 章主要内容是：总体设计，主要包括了系统功能模块的设计以及数据库的概念设计。

第 4 章主要内容是：详细设计，主要由用户前台功能部分与管理后台功能部分组成。

第 5 章主要内容是：系统测试，主要由测试方案、测试设备及环境、功能测试三个部分组成。

第 6 章主要内容是：本次毕业设计的结论。

第 7 章主要内容是：致辞感谢。

第 8 章主要内容是：参考文献。

2 可行性分析及技术介绍

2.1 技术可行性

Spring Boot 是一个成熟的 Java 开发框架，它让开发者能够快速启动新项目，减少所需的配置和编码工作，已有广泛的应用和支持，在技术上具备高度的可行性。Vue.js 不但集成了许多网页上常用的组件模块，而且它的安装部署与使用方式都很简便，且同样适用于构建动态的 Web 界面。在本次毕业设计中，使用 Spring Boot 搭建后端框架，和采用 Vue.js 来部署前端界面，为本文档社区平台系统提供了丰富的参考案例和技术支持，极大程度地简便了开发步骤，Spring Boot 和 Vue 技术为本次文档社区平台的开发从技术上提供了可行性。

2.2 经济可行性

考虑到项目开发采用的都是免费的开源框架，因此开发本文档社区平台项目所需成本的很低。同时，两个开源技术拥有活跃的社区平台，提供了大量的教程和插件，有助于降低开发支出和维护开发成本。在本次毕业设计中，可通过网络快速搜索相关技术文档与教程案例，为本次开发系统过程中遇到问题时，能提供良好思路以及主流的建设性解决办法，这些源自于广大经验丰富开发者主动开源、非付费获取的资源，为本次文档社区平台的开发在经济上提供了可行性。

2.3 操作可行性

系统设计符合现代 Web 开发技术潮流，满足操作可行性。使用 Spring Boot 作为后端和 Vue 作为前端的技术的合理选型，可以确保项目的顺利实施。在本次毕业设计中，由经上述对技术可行性和经济可行性的论证分析，可见，开发本系统所需要准备的条件是一部可联机的电脑，安装部署好的编程环境，即可为本次文档社区平台的开发从操作上提供了可行性。

2.4 相关技术介绍

在当今的软件开发领域，Vue.js、Element UI、Spring Boot 和 MyBatis 这些技术栈已经成为构建现代 Web 应用的热门选择。这些技术不仅各自具有强大的功能和广泛的社区支持，而且它们之间的组合也能够高效地满足各种复杂的业务需求。

2.4.1 Spring Boot

Spring Boot 是基于 Spring 开发的简化开源 Java 框架。Spring 是一个旨在让 Java 开发变得更简单的后端开源框架，在 Spring 的官网可以看到旗下提供了若干个开源

子项目，如 Spring Framework、Spring Boot、Spring Cloud 等，用于完成特定的功能。在 Spring 家族提供的这一套技术中，最核心的是 Spring Framework，其他 Spring 家族的技术都是基于此的，它提供了很多实用功能包括常见的依赖注入、事务管理、web 开发支持、数据访问、消息服务等。不过我们的开发一般不直接基于 Spring Framework 进行开发，以避免配置繁琐、入门难度大的问题，而 Spring 官方推荐的开发方案，便是目前最流行的基于 Spring Boot 技术的后端开发方案。

Spring Boot 最大的特点是简化配置，从而达到快速开发的目的。Spring Boot 的内部带有自动配置功能，它能帮助开发人员快速地构建应用程序、启动新项目，减少所需的配置和编码工作。比如，Spring Boot 基于 pom.xml 和 build.gradle 可自动处理许多常见 Spring 应用程序的配置，避免了手动编写复杂的 .xml 和 Java 配置文件，使项目的文件结构保持简洁。不仅如此 Spring Boot 还集成了 Tomcat 服务器，可方便地将开发完成的项目打包部署到单个 .jar 文件中，并在通过简单的命令行 java -jar 即可启动。Spring Boot 另一大特点就是还支持使用例如 MyBatis 这样的持久层框架来简化数据库操作，直接引入 MyBatis 半自动的 ORM 功能，允许开发人员定制 SQL 语句操作数据库中的记录。在本次设计的文档社区平台中，将使用 MyBatis 与 MySQL 执行查询、新增、删除和更改，完成数据库的交互。^[1]

由此可知，Spring Boot，这款轻量级的开源 Java 框架，以其配置简易、启动快速和项目创建模块化的特点，极大地简化了 Spring 项目的搭建和开发过程，显著提高了开发效率。^[20]

2.4.2 MySQL

在基于 Spring Boot 技术开发的 Web 应用项目中，为了程序职责的单一达到方便维护的目的，我们一般讲 Web 应用程序划分为 Controller、Service、Dao、三个层次，分别对应后端控制层、后端业务层、后端持久层。控制层即为前后端的访问接口，前端发送的请求将首先根据对应路径找到后端的 Controller 层，在控制层引入业务层即 Service 层资源进行调用，在 Service 层中调用 Dao 层，Dao 层最终根据以.xml 为后缀的映射类文件，对数据库进行操作。

Xml 文件中可以存储数据，但在真实的开发项目中，不会采用因为不便管理、操作难度大而不会采用，在企业开发中最终数据的存储和管理都是数据库负责的。若需要对数据库进行操作，则将涉及到数据库管理系统（DataBase Management System, DBMS），DBMS 会通过结构化语言（Structured Query Language, SQL）操作对应的数据库内容，而 SQL 语句是由开发人员提供给后端的。

在众多类型的数据库中，MySQL 是一个开源免费的中小型数据库，后来被收购后也推出了企业版，可以根据所需情况不同自行选择，个人使用已能满足常规的开发需求，MySQL 亦是市面上开发数据库的主流选择之一。SQL 语句根据其功能分为数据定义语言（Data Definition Language, DDL）、数据操作语言（Data Manipulation Language, DML）、数据查询语言（Data Query Language, DQL）、数据控制语言（Data Control Language, DCL），分别用于定义数据库对象（数据库、表、字段）、对数据库表中的数据进行增删改、查询数据库中表的记录，和创建数据库用户、控制数据库的访问权限。^[6]

2.4.3 MyBatis

在学习 MySQL 数据库时，可以通过可视化软件操作数据库，作为后端应用的开发者，通常会使用 Java 程序来完成对数据库的操作，而现行主流的方式则是采用 MyBatis。

MyBatis 是一款用于简化 JDBC（Java Database Connectivity）代码的持久层框架。^[9]JDBC 是一种用于 Java 程序访问关系型数据库的 API（应用程序接口），允许 Java 程序员使用标准的 SQL 语句来访问和操作关系型数据库。MyBatis 作为持久层框架的意义在于说，后端是通过持久层即 Dao 层来操作数据库的。

在未使用 MyBatis 时，需要在图形化客户端工具中编写 SQL 代码，发送给数据库，让数据库返回执行后的操作结果。使用 Mybatis 操作数据库，就是将 Mybatis 中编写的 SQL 语句发送给数据库，当数据库执行并返回结果后，Mybatis 会根据实体类把数据库的返回结果封装起来。MyBatis 通过使用.xml 文件或简单的注解即可配置和映射实体类数据，将接口和普通的旧 Java 对象映射成数据库中的记录，在很大程度上避免了所有需要手动设置 JDBC 代码和参数及检索结果的步骤。^[3]

MyBatis 作为一个优秀的持久层框架，它提供了简单而强大的数据访问层，使得数据库操作更加高效和灵活。支持定制化 SQL、存储过程以及高级映射。^[21]

2.4.4 前端技术

前端主要使用了 Vue、ElementUI 技术。这些技术都旨在提供更方便快捷的前端框架，构造一个简洁美观的前端。

Vue.js: Vue 是 Vue.js 的简称，是一个前端框架，支持与其他第三方库进行整合，用于快速构建用户界面。Vue 基于 MVVM (Model-View-ViewModel) 的前端开发思想，便于开发复杂的单页应用。M 即模型, 包括数据和一些基本操作; V: 即视图、

页面渲染结果；VM:即模型与视图间的双向操作。当用户操作视图组件，前端框架需要通过 DOM 操作获取视图中的数据，同步到模型中；开发人员则从后端获取数据模型，通过 DOM 操作模型，然后把数据渲染到视图中。MVVM 中的 VM 将 DOM 操作完全封装，使得开发人员不必关心模型和视图之间相互影响的底层逻辑，只要操作了 DOM，就可以立即在页面上显示视图；而当用户操作了视图，亦可以改变模型中的数据。[22]

Element UI: 这是基于 Vue.js 的 UI 组件库。在众多前端 UI 框架中，ElementUI 脱颖而出的原因多种多样。Element UI 的安装使用操作非常简单，易于上手，在安装 Node.js 的开发环境中使用 npm 命令行即可快速启用。Element UI 是专为 Vue.js 设计的组件库，提供了丰富的前端组件样式。Element UI 与 Vue 的紧密集成，由于拥有易于使用的 API，Element UI 能够让开发者快速构建高效且美观的网页应用，极大地提高了开发效率。最后，ElementUI 还拥有活跃的社区，持续更新丰富的学习资源，为开发者提供了强大的技术支持。

综上所述，上述技术栈即 Vue.js、Element UI、Spring Boot 和 MyBatis 都具有强大的功能，它们在各 Vue 和 Spring Boot 前后端分离项目领域内有着广泛地应用，开发者可以根据项目需求和自身技术栈的熟悉程度，灵活选择和组合这些技术，构建高效、稳定且用户友好的 Web 应用。尽管它们各自也面临着不同的挑战，通过技术的不断更新和社区的大力支持，这些技术将继续在现代 Web 开发中发挥重要作用。

3 总体设计

3.1 系统功能模块

本系统包括用户登录注册、用户个人中心、用户自定义分类、用户自定义上传、用户富文本编辑、多用户协同编辑等前台功能模块，以及支持用户信息管理、用户发布管理、系统公告管理和系统日志管理等后台功能模块。其中用户自定义分类、用户自定义上传、用户富文本编辑模块统一定义为综合发布模块。系统是基于 Spring Boot 加 Vue 技术的前后端分离项目，主要目的是设计并实现的一个功能完备、界面友好的文档社区平台。

在登录注册模块，用户需要通过账号密码及登录验证码进行鉴权确认，也可以使用初始密码和初始身份权限进行注册。在个人中心页面，当用户登入系统后，可以对账户信息进行编辑，对身份展示进行自定义更改。在综合发布页面，用户可以自定义文件分类、可以自定义上传文件格式、可以使用富文本编辑器进行撰写文档。在多用户协同编辑页面，用户可以创建新的协同编辑，也可以选择加入已有的协同编辑号，与他人进行合作。

系统用例图如图 3.1 所示：

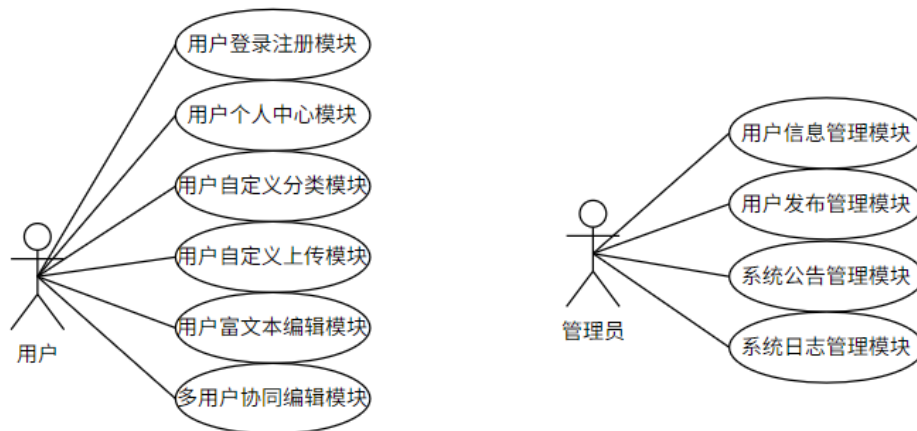


图 3.1 系统用例图

该文档社区平台系统根据用户身份划分为两种角色：一般用户通过前台界面访问浏览并使用系统功能，管理员通过后台对页面前台信息进行管理。管理员登录后台后，可以对用户信息进行统一管理；可以对发布数据进行统一管理；可以对操作日志进行统一管理；可以对公告信息进行统一管理。管理员的统一管理操作包括新增、删除、修改、查询等基础操作，以前端页面可视化的形式对数据库信息进行更新操作。

功能结构图如图 3.2 所示：

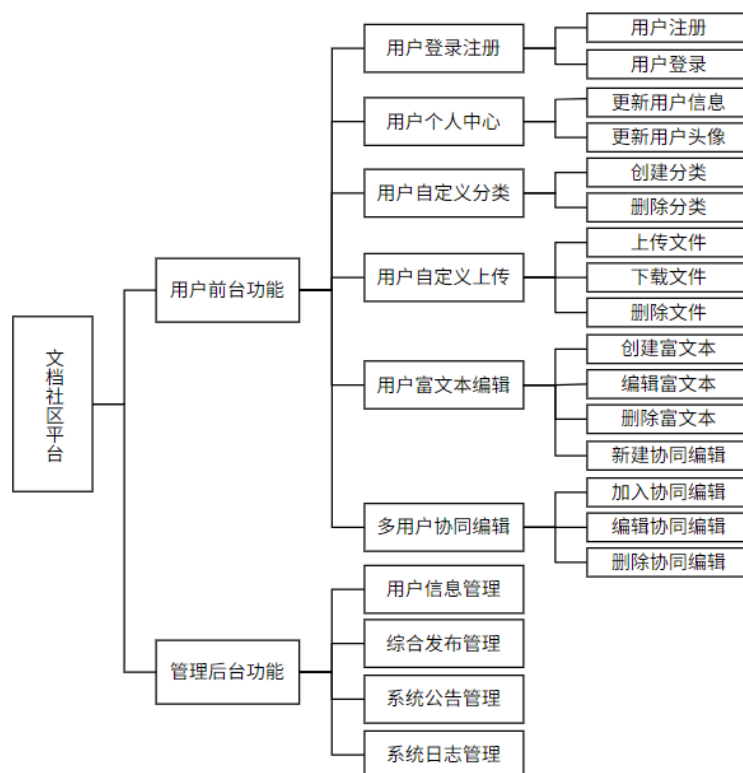


图 3.2 功能结构图

3.2 数据库设计

该文档社区平台划分为两类用户：普通用户或管理员。在数据库的分析和设计过程中，对用户信息、发布分类、上传文件、系统公告、系统日志等实体进行了深入分析。

用户信息实体类，对用户的基础信息进行统一处理，包括用户 ID、用户姓名，和其他信息属性。

用户头像实体类，对用户上传头像的图片进行统一处理，用户头像需要根据用户对象进行设置，需要根据图片存储路径表示进行获取并展示，因此包括用户 ID、图片标识和其他属性。

登录状态实体类，对用户的登录状态进行统一处理，登录状态需要根据用户对象进行设置，因此包括用户 ID、登录状态和其他属性。

系统日志实体类，对系统记录用户的操作进行统一处理，系统日志需要记录操作日志的用户对象以及操作的内容，因此包括用户 ID、用户 IP、日志内容和其他属性。

系统公告实体类，对系统管理员发布的公告进行统一处理，系统公告需要记录公告标题、公告内容，和公告发布的时间。

发布分类实体类，对用户创建的分类进行统一处理，创建的分类需要根据登录用户的不同和用户各自定义的分类名称进行创建，因此包括用户 ID、分类名和其他属性。

富文本编辑实体类，对用户在分类下新建的富文本编辑进行统一处理，新建的富文本编辑需要依赖创建的分类进行增加，还需要根据登录的不同用户和各自定义的标题进行创建，因此包括分类 ID、用户 ID、富文本标题和其他属性。

音视频文件实体类，对用户分类下上传的文件进行统一处理，上传的文件需要依赖分类的标识进行增加，还需要根据不同用户各自上传文件的不同进行区别，因此包括分类 ID、用户 ID、上传标识、上传文件名和其他属性。

富文本内容实体类，对用户富文本编辑下添加的内容子项进行统一处理，添加的内容子项需要依赖富文本的标识进行增加，还需要根据不同用户各自定义的标题、各自编辑的内容进行区分，因此包括富文本 ID、用户 ID、富文本标题、富文本内容和其他属性。

协同新建实体类，对用户创建的协同编辑进行统一处理，创建的协同需要根据不同的创建者各自定义的协同标题进行增加，因此包括创建者 ID、协同标题和其他属性。

协同加入实体类，对用户加入的协同编辑进行统一处理，一个用户可以创建、也可以加入一个协同编辑。每个协同编辑可包含零或多个加入成员，但有且仅包含一个创建者，每个用户的加入协同编辑集合不同，因此实体类包括协同加入对象 ID、用户 ID、加入的协同 ID、协同创建者 ID 和其他属性。

协同成员实体类，对协同编辑下加入的用户成员进行统一处理，每个协同编辑包含一个创建者，可包含零个或多个加入成员，因此实体类包括加入协同成员对象 ID、成员 ID、加入成员名和其他属性。

根据此实体关系分析，画出以下系统 E-R 模型如图 3.3 所示：

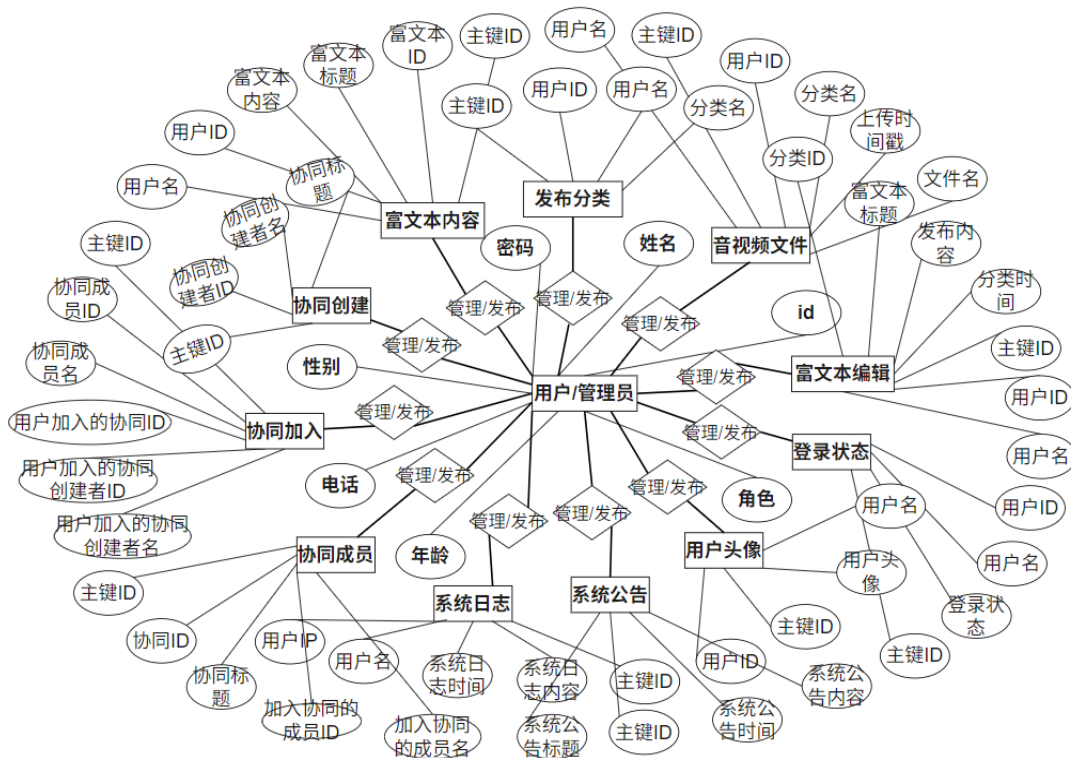


图 3.3 数据库 E-R 图

通过 E-R 图可以建立面向每个实体对象或功能对象的数据库表，如下所示：

（1）用户信息表包含的字段有：主键 ID、姓名、密码、性别、年龄、电话、角色。如表 3-1 所示：

表 3-1 用户信息表 admin.sql

序号	列名	数据类型	长度	主键	自增	不是空	默认值	列说明
1	id	int		√	√	√		主键 ID
2	name	varchar	255					姓名
3	password	varchar	255					密码
4	sex	varchar	255					性别
5	age	int						年龄
6	phone	varchar	255					电话
7	role	varchar	255					角色

（2）用户头像表包含的字段有：主键 ID、用户 ID、用户名、用户头像。如表

3-2 所示:

表 3-2 用户头像表 img. sql

序号	列名	数据类型	长度	主键	自增	不是空	默认值	列说明
1	id	int		√	√	√		主键 ID
2	userid	int						用户 ID
3	username	varchar	255					用户名
4	img	varchar	255					用户头像

(3) 登录状态表包含的字段有: 主键 ID、用户 ID、用户名、登录状态。如表 3.3 所示:

表 3-3 登录状态表 login. sql

序号	列名	数据类型	长度	主键	自增	不是空	默认值	列说明
1	id	int		√	√	√		主键 ID
2	userid	int						用户 ID
3	username	varchar	255					用户名
4	status	varchar	255					登录状态

(4) 系统日志包含的字段有: 主键 ID、系统日志内容、系统日志时间、用户名、用户 IP。如表 3-4 所示:

表 3-4 系统日志表 log. sql

序号	列名	数据类型	长度	主键	自增	不是空	默认值	列说明
1	id	int		√	√	√		主键 ID
2	content	varchar	255					系统日志内容
3	time	varchar	255					系统日志时间
4	username	varchar	255					用户名
5	ip	varchar	255					用户 IP

(5) 系统公告表包含的字段有: 主键 ID、系统公告标题、系统公告内容、系统公告时间。如表 3-5 所示:

表 3-5 系统公告表 notice. sql

序号	列名	数据类型	长度	主键	自增	不是空	默认值	列说明
1	id	int		√	√	√		主键 ID
2	name	varchar	255					系统公告标题
3	content	varchar	255					系统公告内容
4	time	varchar	255					系统公告时间

(6) 发布分类表包含的字段有：主键 ID、用户 ID、用户名、分类名。如表 3-6 所示：

表 3-6 发布分类表 category.sql

序号	列名	数据类型	长度	主键	自增	不是空	默认值	列说明
1	id	int		√	√	√		主键 ID
2	userid	int						用户 ID
3	username	varchar	255					用户名
4	category	varchar	255					分类名

(7) 富文本编辑表包含的字段有：主键 ID、用户 ID、用户名、分类 ID、分类标题、发布内容、发布时间。如表 3-7 所示：

表 3-7 富文本编辑表 document.sql

序号	列名	数据类型	长度	主键	自增	不是空	默认值	列说明
1	id	int		√	√	√		主键 ID
2	userid	int						用户 ID
3	username	varchar	255					用户名
4	categoryid	int						分类 ID
5	title	varchar	255					富文本标题
6	content	varchar	255					发布内容
7	time	varchar	255					发布时间

(8) 音视频文件表包含的字段有：主键 ID、分类名、用户 ID、用户名、上传时间戳、文件名。如表 3-8 所示：

表 3-8 音视频文件表 files.sql

序号	列名	数据类型	长度	主键	自增	不是空	默认值	列说明
1	id	int		√	√	√		主键 ID
2	category	int						分类名
3	userid	int						用户 ID
4	username	varchar	255					用户名
5	flag	varchar	255					上传时间戳
6	filename	varchar	255					文件名

（9）富文本内容表包含的字段有：主键 ID、发布 ID、发布标题、用户 ID、用户名、发布内容。如表 3-9 所示：

表 3-9 富文本内容表 doceditor.sql

序号	列名	数据类型	长度	主键	自增	不是空	默认值	列说明
1	id	int		√	√	√		主键 ID
2	docid	int						富文本 ID
3	doctitle	varchar	255					富文本标题
4	userid	int						用户 ID
5	username	varchar	255					用户名
6	doceditorCotent	longtext						富文本内容

（10）协同创建表包含的字段有：主键 ID、协同创建者 ID、协同创建者名、协同标题。如表 3-10 所示：

表 3-10 协同新建表 sync.sql

序号	列名	数据类型	长度	主键	自增	不是空	默认值	列说明
1	id	int		√	√	√		主键 ID
2	authorID	int						协同创建者 ID
3	authorname	varchar	255					协同创建者名
4	title	varchar	255					协同标题

（11）协同加入表包含的字段有：主键 ID、协同成员 ID、协同成员名、用户加

入的协同 ID、用户加入的协同创建者 ID、用户加入的协同创建者名。如表 3-11 所示：

表 3-11 协同加入表 svncjoin.sql

序号	列名	数据类型	长度	主键	自增	不是空	默认值	列说明
1	id	int		√	√	√		主键 ID
2	userid	int						协同成员 ID
3	username	varchar	255					协同成员名
4	joinid	int						用户加入的协同 ID
5	joinauthorid	int						用户加入的协同创建者 ID
6	joinauthorname	varchar	255					用户加入的协同创建者名

(12) 协同成员表包含的字段有：主键 ID、协同 ID、协同标题、加入协同的成员 ID、加入协同的成员名。如表 3-12 所示：

表 3-12 协同成员表 syncuser.sql

序号	列名	数据类型	长度	主键	自增	不是空	默认值	列说明
1	id	int		√	√	√		主键 ID
2	syncid	int						协同 ID
3	synctitle	varchar	255					协同标题
4	userid	int						加入协同的成员 ID
5	username	varchar	255					加入协同的成员名

综合以上全部的数据库表得到数据库的物理模型如图 3.4 所示：

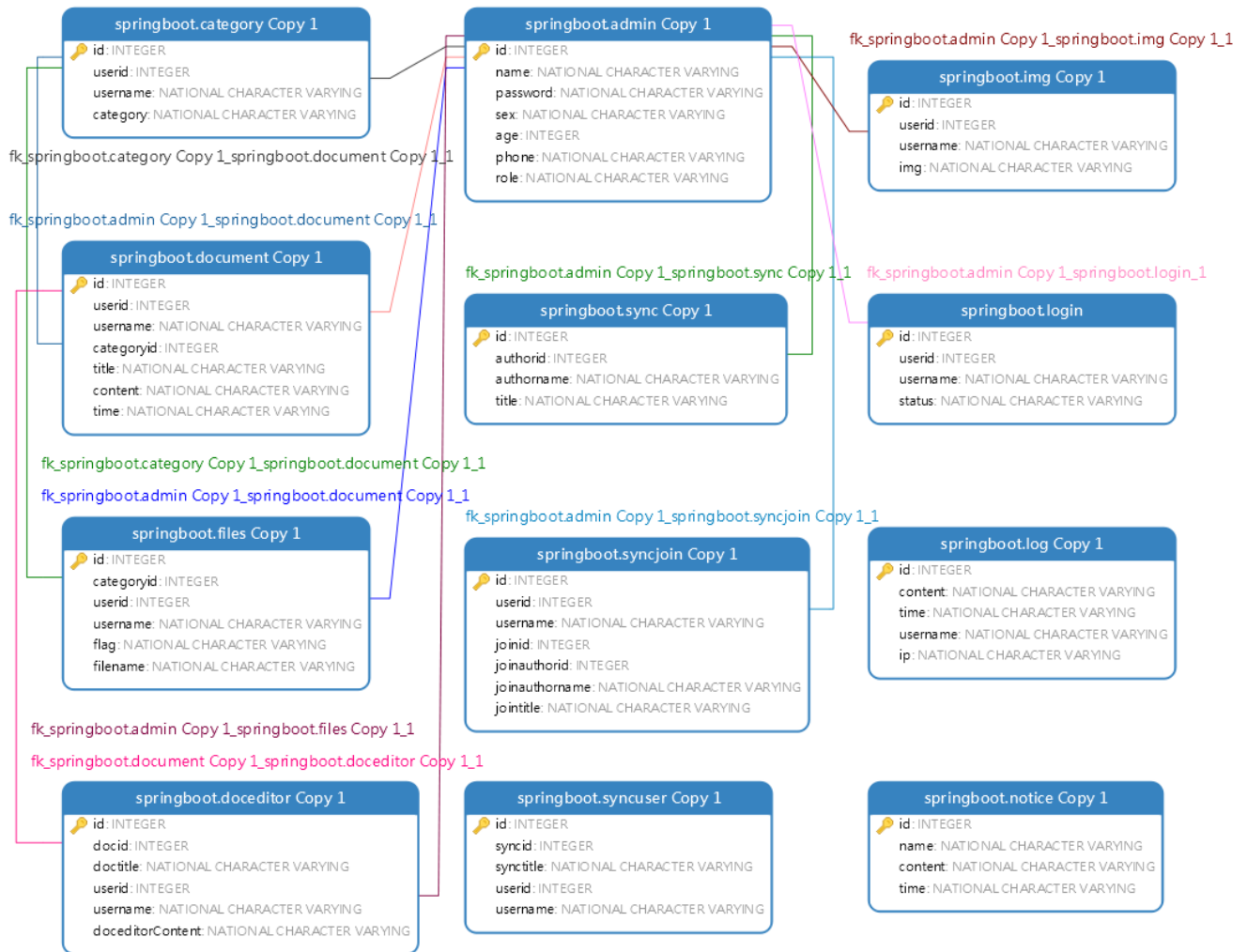


图 3.4 物理模型

4 详细设计

4.1 用户前台功能部分

前台界面用于对已有的数据信息进行布局设计和合理展示，后台管理则对数据内容进行权限控制和统一管理，前台界面对于一个完整的系统来说是必不可少的，它的作用是将用户操作变得可视化。用户前台功能模块主要包括了以下几个模块：用户登录注册模块、用户个人中心模块、用户综合发布模块、用户协同发布模块。

其中综合发布模块是在一个前端页面上实现了多个功能模块，包括了用户自定义分类、自定义上传、富文本编辑等功能模块。其整体发布流程是这样的，用户点击新增“分类”按键，首先创建分类文件夹，这时候需要先对新建的分类进行命名，然后才可以在此分类下添加富文本或者上传音视频文件。一名用户可以添加多个自定义分类。在分类下有新增富文本集或者上传文件两种选择。一个分类下可以添加多个富文本集，亦可以上传多个自定义文件；一个富文本集下可以发布多个内容片段，称之为富文本的“内容子项”。一个“内容子项”实则代表了一个独立的富文本编辑器的创建，而“富文本集”则指的是多个编辑器集合的父容器。上述创建的“分类”及其子元素均可按照用户需要进行移除，当父容器的子元素不为空时，需要首先批量删除全部存在的子元素，最后再对父容器进行删除。

在综合发布页面，这个页面是系统的主体功能的结合页面，它的设计难点在于如何分层次地控制前端由 `v-for` 嵌套渲染的 `div` 组件，并且如果有需要，要求能够单独地对每一个嵌套中的子组件进行控制，以达到能够操作元素组件产生不同的效果，例如单独地控制每个元素的显示状态、独立控制其不同的点击事件等。用户在综合发布页面，能够操作以多媒体文件、富文本为基础要素的“发布”，形式上类似于文件管理功能，但主要目的仍是为了更好的展示与预览发布的内容。

在综合发布页面中可了解到，其主要功能是为了让用户能够选择更灵活的文档撰写模式，如果遇到用户在创作过程中难以使用单一的文字描述或单一的视频讲解表达自己想法的时候，还能有选择地使用上传其他文件附件，如视频演示、音频呈现等更多元的方式来展现自己的思考。从本地上传的文件会根据其自身命名，对于一些较为常见的文件格式如图片 PNG/JPG、视频 MP4、音频 MP3，用户可以直接通过点击页面项目进行预览或播放，避免了跳转第三方。对于一些无法直接预览的文件格式，用户也可以通过下载到本地进而达到便捷地获取资源、分享传达的目的。

综合发布的操作流程是，在新增发布页面，点击图标进行创建分类，输入分类标题，点击提交，点击图标可以删除创建的分类。在创建的分类下点击添加图标，

可选择新增上传音视频文件或者新增富文本编辑选项。在分类下选择新增上传，可以上传不同格式的文件，点击图标可以下载上传的文件、删除上传的文件。在分类下选择新增富文本编辑，输入富文本标题，点击提交，点击图标可以删除新增的富文本。在富文本下点击新增内容子项图标，在富文本编辑器中可以编辑内容，点击保存，点击图标可以删除添加的内容子项。

综合发布流程如图 4.5 所示：

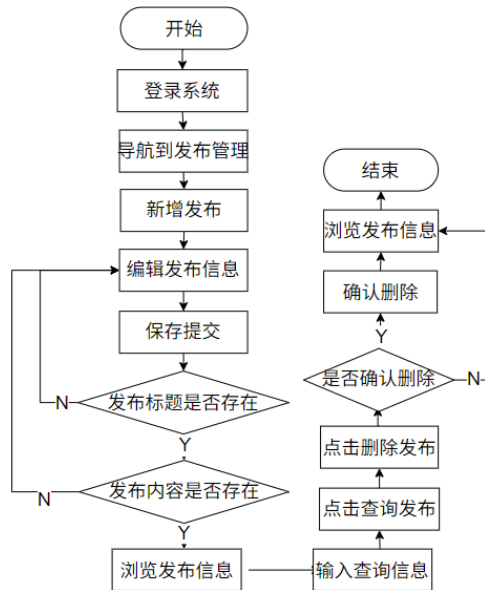


图 4.5 综合发布流程图

综合发布的运行结果是，登录用户可以创建多个分类；在分类下可以创建多个富文本或上传多个文件；在富文本下可以添加多个富文本内容片段。用户每次创建成功，页面视图将新的项目根据名称添加到列表区，用户可以删除自己的创建。在列表区，当用户点击上传的不同文件，预览区则会切换显示不同格式的文件，支持播放预览的文件有图片、音频、视频等多媒体文件；当用户点击创建的不同富文本，预览区则会切换显示不同富文本编辑的文本内容。

综合发布运行结果如图 4.6 所示：



图 4.6 综合发布运行图

综合发布的每个功能模块将在后续详细介绍。

4.1.1 用户登录注册模块

登录注册的操作流程是，用户进入注册页面，输入用户名和密码，选择用户角色，点击注册用户，成功注册后将跳转到登录页面。用户进入登录页面，输入用户名、密码和验证码，点击登录按钮，登录成功后将跳转到系统首页。

登录注册流程如图 4.1 所示：

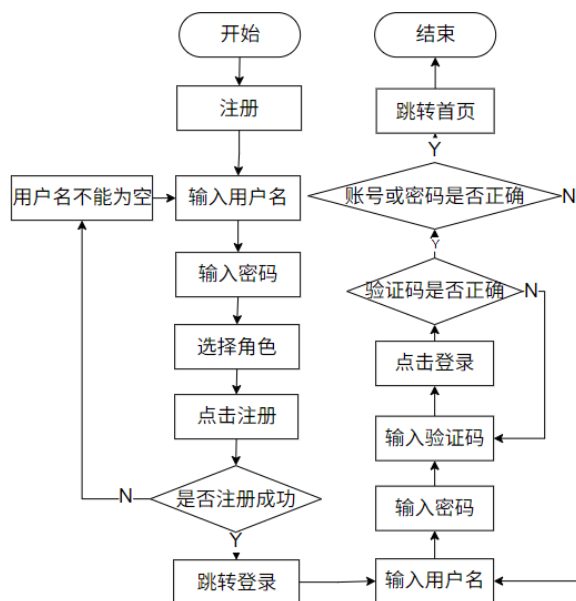


图 4.1 登录注册流程图

登录注册的运行结果是，在用户注册页面，如果未输入用户名或输入了重复的用户名，则会注册失败；如果未输入注册密码，则会使用默认初始化密码进行注册；如果未输入注册角色，则会使用默认初始化角色进行注册。在用户登录页面，如果输入了错误的验证码，则会登录失败；如果输入了错误的登录用户或密码，则会登录失败。

登录注册运行结果如图 4.2 所示：

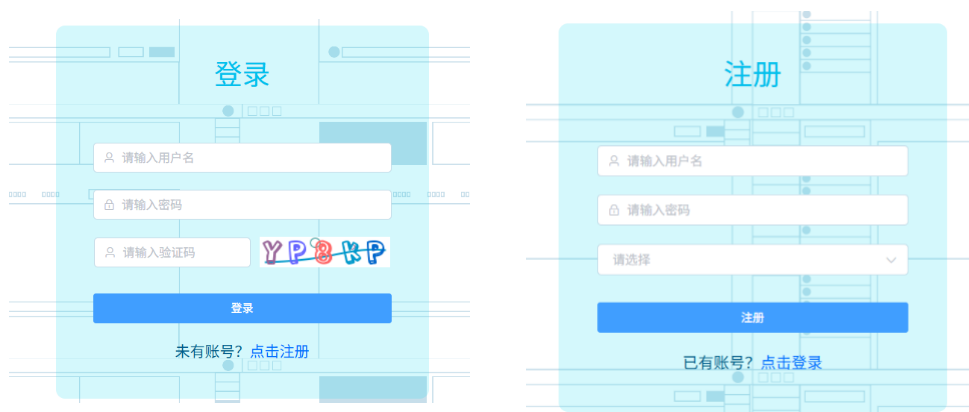


图 4.2 登录注册运行图

登录注册的详细设计过程如下。

4.1.1.1 用户注册模块

在用户注册部分，前端将获取用户输入或者设置默认初始化值，并将输入结果作为注册参数，通过封装的 Axios 对象发起 HTTP 请求并携带传递参数，根据请求的路径找到后端接口，通过后端封装的参数实体类接收前端传递的参数，再由 Controller 层、Service 层、Dao 层，逐层调用，通过 Mapper 层传递给 MyBatis 的 SQL 语句，操作 Entity 层实体类对应的数据库表，向用户信息表中新增一条用户记录。

4.1.1.2 用户登录模块

在用户登录部分，前端将获取用户输入并将结果作为登录参数，向后端发送携带参数的请求。当用户点击登录时，先鉴别验证码是否输入正确，再鉴别登录信息是否输入正确。验证码的生成是通过在后端引入的验证码图形工具依赖实现的，可定义字符个数和长宽属性，每次刷新页面或点击图形，都会根据随机参数重新生成验证码，鉴别时也会根据生成的关键值解码鉴别。在验证码的鉴别过程中，将用户输入的字母统一转换成小写，用户输入时不必再区分大小写。通过验证码和账号密码后，浏览器将用户信息保存到本地存储，用户可成功访问系统页面，页面根据从本地存储获取到的用户信息进行区别化的渲染。

在登录注册模块，本系统在后端的配置类中对前端请求设置了统一拦截，不允许未登录的用户通过浏览器输入地址非法访问系统内部。登录鉴权方案采用了 JWT 鉴权法则，即在用户登录时设置唯一 token 凭证，并且设置 token 有效期，如果用户登录时携带的 token 是假的或者携带的 token 超出了有效时限，系统则会拒绝用户登录访问。仅当用户携带的 token 是能够被后台正确解码的、并且在有效期内的，才会允许登录。用户的登录信息被简单存储在浏览器的本地存储 Local Storage 中，在页面上可通过 get、set 等方法快速获取，并根据用户身份渲染对应内容。当用户点击登出时，则会清除遗留在浏览器中的缓存，保护用户信息的安全。这样的处理方式避免了每次使用用户数据时都要向后端查询的繁琐步骤，进一步提高了系统的性能。

4.1.2 用户个人中心模块

个人中心的操作流程是，登录用户进入个人中心页面，点击编辑按钮，输入待更改的用户信息后，点击确认按钮进行提交。点击上传按钮，选择本地文件夹的图片文件进行上传，更新当前登录用户的头像。

个人中心流程如图 4.3 所示：

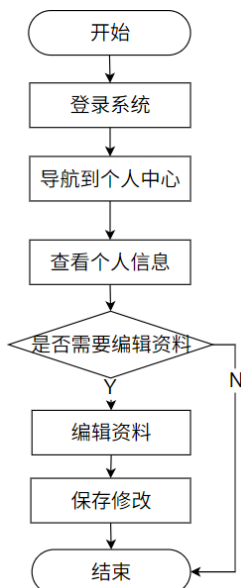


图 4.3 个人中心流程图

个人中心的运行结果是，当用户点击编辑按钮时，编辑信息的对话框开启，且对话框的可编辑数据已经显示为当前登录用户信息，用户改动信息，点击确认按钮后，页面数据信息更新为刚刚编辑后的信息。点击上传按钮，提交本地上传以后，用户头像更新为刚刚上传后的信息。再次退出重新登录系统，刚刚的编辑操作永久生效。不同用户个人中心页面的信息展示不会相互干扰。

个人中心运行结果如图 4.4 所示：

个人中心运行图展示了用户编辑个人信息的界面。左侧是个人中心页面，右侧是弹出的“请填写个人信息”对话框。

左侧页面显示：

- 个人设置
- 账户资料
- 其他设置
- 参考
- 参考
- 参考
- 参考
- 参考
- 参考
- 参考
- 其他设置
- 参考
- 参考
- 参考

右侧对话框显示：

- 姓名：admin
- 性别：☐ 男 ☒ 女
- 年龄：25
- 电话：18888889999
- 角色：ROLE_ADMIN
- 取消 确定

图 4.4 个人中心运行图

个人中心详细设计过程如下。

4.1.2.1 更新用户信息模块

在更新用户信息部分，首先开启更新个人信息的对话框，前端将获取用户的点击事件控制编辑对话框的显示与隐藏，前端将根据登录用户 ID 向后端发送请求，后端将根据 SQL 语句查询用户信息表并将结果返回给前端页面，渲染为用户信息。

然后再提交更新的个人信息，前端将通过输入框获取并绑定用户更改后的表单数据，作为用户信息参数，并向后端发送携带参数的请求，后端将根据 SQL 语句对用户信息表的内容进行更新，用户信息表的关键字段为用户 ID。数据库表的内容更改后，前端用户信息数据将随之改变。

4.1.2.2 更改用户头像模块

在更新用户头像部分，前端将当前登录用户作为参数向后端发起请求，后端将根据 SQL 语句查询用户头像表，如果查询到当前用户的头像唯一标识，则返回给前端进行统一处理。前端将通过用户的点击事件向后端配置的文件上传接口发送请求，文件上传的工具类是根据 MyBatis 中添加的依赖进行构建的。用户上传图片文件成功以后，将得到后台对上传文件处理后返回的唯一标识，前端将获取到的标识作为参数再次向后端发起请求，后端根据 SQL 语句向用户头像表中新增或者更新一条记录，用户头像表的关键字段包括用户 ID、头像唯一标识。用户上传图片文件唯一标识保存到数据库以后，前端还需根据查询得到或者文件接口返回得到的图片唯一标识，再次访问文件处理接口，以在页面上加载对图片文件的预览。

4.1.3 用户自定义分类模块

4.1.3.1 创建自定义分类模块

在创建分类部分，前端将获取用户输入的分类名并作为分类对象的参数，并向后端发送携带参数的请求，后端根据 SQL 语句向数据库的分类表新增一条记录，分类表的关键字段包括用户 ID、分类名。创建以后将调用查询方法，根据当前登录用户查询出创建的分类集合，并在页面视图以分类名的形式进行展示。

4.1.3.2 删除自定义分类模块

在删除分类部分，如果登录用户需要删除自己创建的分类，则需要同时一起删除分类下自定义的音视频文件或富文本内容。前端将根据用户的点击事件获取待删除的分类 ID，并作为参数向后端发送请求，后端根据 SQL 语句先查询数据库的音视频文件表，删除此分类 ID 对应的全部文件；然后查询富文本内容表，删除此用户操作的分类 ID 对应的富文本内容及其全部子项。最后再调用查询分类表，在前端页面上刷新删除后的视图界面。

自定义分类运行结果如图 4.5 所示：



图 4.5 自定义分类运行图

4.1.4 用户自定义上传模块

4.1.4.1 上传音视频文件模块

在上传文件部分，每当用户上传一个文件，系统将根据文件上传的时间得到一个时间戳，作为唯一标识“flag”保存到后台数据库，以此获取当前文件信息，用于在前端页面上展示上传的文件信息和在后端处理文件的下载请求。前端将通过用户的点击事件获取待上传的分类 ID 并向后端配置的文件上传接口发送请求，文件上传的工具类是根据 MyBatis 中添加的依赖进行构建的。用户上传自定义文件成功以后，将得到后台对上传文件处理后返回的唯一标识，前端将获取到的标识和点击事件获取到的分类 ID 作为参数再次向后端发起请求，后端根据 SQL 语句向文件表中新增一条记录，文件表的关键字段包括用户 ID、分类 ID、文件唯一标识。用户上传文件记录保存到数据库以后，前端还需根据通过查询或通过文件接口返回得到的唯一标识，再次访问文件处理接口，以在页面上加载对上传文件的预览。前端将当前登录用户作为参数向后端发起请求，后端将根据 SQL 语句查询文件表，将当前用户 ID 对应的文件唯一标识返回给前端进行统一处理。

4.1.4.2 下载音视频文件模块

在下载文件部分，前端将通过用户的点击事件获取待下载的文件 ID 并作为参数

向后端发起请求，后端根据 SQL 语句查询文件表并将关键字段文件唯一标识返回给前端，前端通过获得的唯一标识向后端配置的文件上传接口发送请求，后端配置工具类将根据唯一标识下载对应的文件，并反馈给前端用户用于自定义保存地址。

4.1.4.3 删除音视频文件模块

在删除文件部分，前端将根据用户的点击事件获取待删除的文件 ID，并作为参数向后端发送请求，后端根据 SQL 语句查询数据库的文件表，删除此 ID 对应的文件。最后再调用查询全部文件方法，根据当前登录用 ID，在前端页面上刷新显示删除后的视图界面。

自定义上传运行结果如图 4.6 所示：

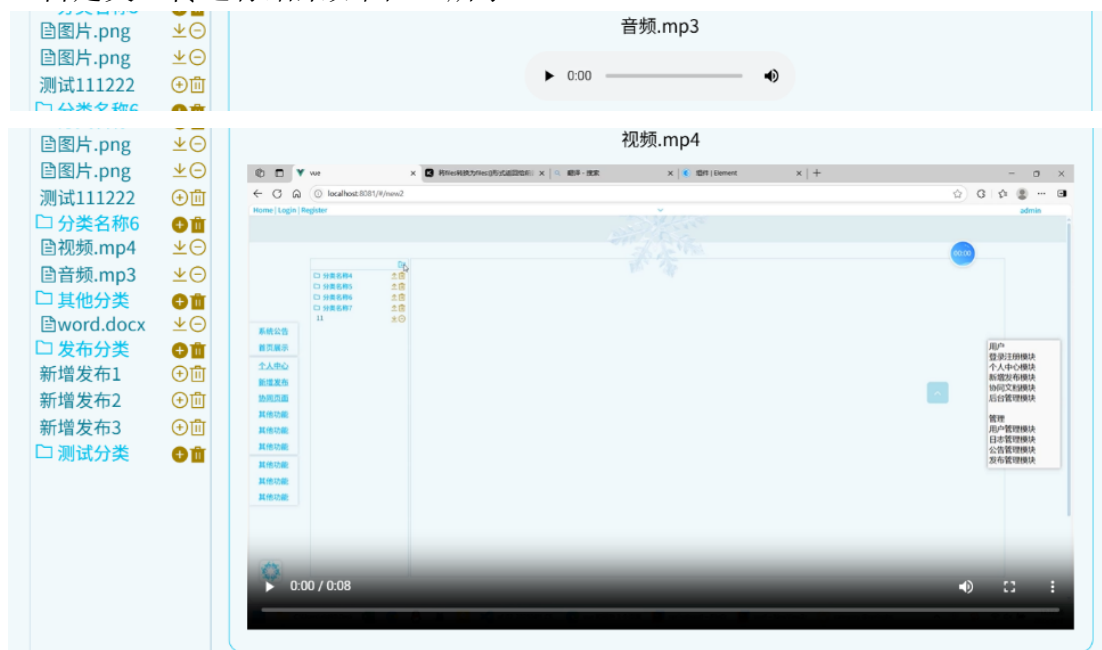


图 4.6 自定义上传运行图

4.1.5 用户富文本编辑模块

4.1.5.1 创建富文本内容模块

在创建富文本内容部分，富文本内容包括多个内容子项，用户需要先创建富文本编辑父类，再进行内容项的添加。前端首先通过用户点击事件获取待新增的分类 ID 并通过绑定输入获取的内容标题，作参数向后端发送的请求，后端根据 SQL 语句向数据库的富文本编辑表新增一条记录，富文本编辑表的关键字段包括分类 ID、用户 ID、富文本标题。创建以后将调用查询方法，根据当前登录用户查询创建的富文本编辑集合，并在页面视图中以标题命名的形式进行展示。

然后用户可以在新建的富文本编辑下，添加内容项。前端通过用户的点击事件获取待添加的富文本编辑 ID，并作为参数向后端发送请求，后端根据 SQL 语句向数据库的富文本内容表新增一条记录，富文本内容表的关键字段包括富文本编辑 ID、用户 ID、富文本内容。新增以后将调用查询方法，根据当前登录用户以及点击事件获取的富文本编辑 ID，查询创建的富文本内容子项集合，并在用户点击不同的富文本编辑标题时在页面视图切换展示对应的富文本内容视图。

4.1.5.2 编辑富文本内容模块

在编辑富文本内容部分，每当用户选中一个内容子项时，前端将为选中的内容区创建并打开一个富文本编辑器，当用户编辑内容并点击保存后，前端则将销毁当前的编辑器并将用户编辑的内容以 HTML 形式渲染在原内容区的位置。前端通过点击事件获取用户选中的内容子项 ID，通过编辑器绑定获取用户输入的内容，并向后端发送携带参数请求，后端根据 SQL 语句向数据库的富文本内容表新增一条记录，富文本内容表的关键字段包括用户 ID、富文本编辑 ID、富文本内容。新增以后将调用查询方法，获取当前用户编辑的全部富文本内容，前端页面将根据用户点击事件的不同更改 v-for 渲染的条件，用于切换展示不同富文本编辑的内容。

4.1.5.3 删除富文本内容模块

在删除富文本内容部分，如果用户需要删除自己创建的富文本编辑，则需要同时一起删除富文本编辑下的内容子项。前端将根据用户的点击事件获取待删除的富文本编辑 ID，并作为参数向后端发送请求，后端根据 SQL 语句查询数据库的富文本内容表，删除此用户创建的富文本编辑 ID 对应的内容子项。最后再调用查询方法，在前端页面上刷新删除后的视图界面。

富文本编辑运行结果如图 4.7 所示：

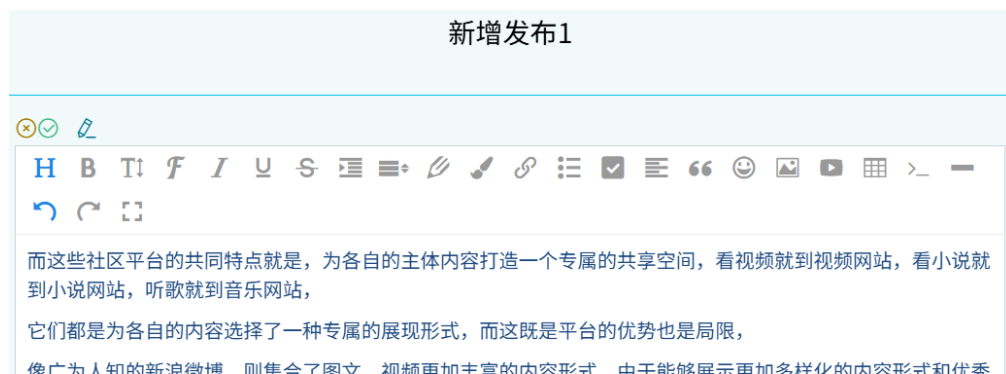


图 4.7 富文本编辑运行图

4.1.6 多用户协同编辑模块

协同编辑的操作流程是，用户进入协同页面，点击新建协同图标，选择新建协同编辑或者加入协同编辑。选择了新建协同编辑，则需要新建协同标题，选择了加入协同，则输入加入协同号。用户可以查看加入到列表中协同的 ID，也可以删除自己创建的协同。用户点击列表中已有的协同编辑，可以切换进入不同的协同编辑，并在文本编辑器中输入自定义内容或者浏览他人已编辑的内容。

协同编辑流程如图 4.8 所示：

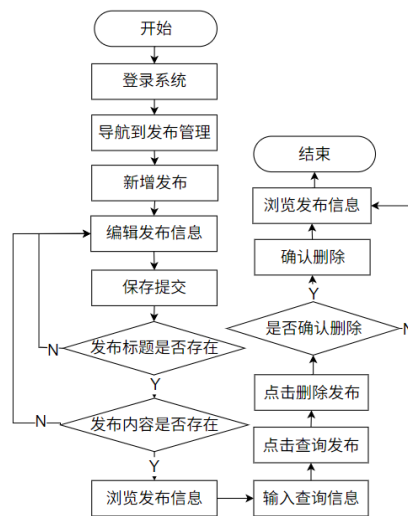


图 4.8 协同编辑流程图

协同编辑的运行结果是，当用户新建了一个协同编辑，新建的协同将会被添加在“我创建的协同”列表区域，同时以协同的标题和协同的创建者形式展现在页面上。当用户加入了一个协同编辑，被加入的协同编辑将会被添加到“我加入的协同”列表区域，每个协同除了展示标题和创建者，还会在下方显示全部的加入成员。当有多个登录用户加入了同一个协同编辑时，不同用户在文本区编辑的内容将会进行同步展示。

协同编辑运行结果如图 4.9 所示：



图 4.9 协同编辑运行图

协同编辑的详细设计过程如下。

4.1.6.1 新建协同编辑模块

在新建协同部分，前端将获取用户输入的标题并作为新建协同对象参数，并向后端发送携带参数的请求，后端根据 SQL 语句向数据库的协同创建表新增一条记录，协同创建表的关键字段包括创建者 ID、协同标题。创建以后将调用查询方法，根据当前登录用户查询出创建的协同编辑集合，并在页面视图中以协同标题、创建者的形式进行展示。

4.1.6.2 加入协同编辑模块

在加入协同部分，前端将获取用户输入的协同号并作为加入协同对象参数，首先根据协同 ID 向协同加入表中新增一条记录，用于将用户加入的协同添加到页面列表进行展示；然后根据操作用户向协同成员表中新增一条记录，用于在加入的协同编辑下展示全部的成员用户。前端首先向后端发送携带参数的请求，后端根据 SQL 语句向协同加入表中增加一条记录，关键字段包括成员 ID、协同 ID；再向协同成员表中增加一条记录，关键字段包括协同 ID、成员 ID。然后前端向后端发送查询请求，后端根据 SQL 语句分别查询数据库的加入协同表、协同成员表，分别将查询结果得到的加入协同名、协同成员属性渲染到页面视图上。

4.1.6.3 编辑协同编辑模块

在编辑协同部分，前端将根据用户的点击事件获取待编辑的协同 ID，并作为参数向后端发送请求，后端根据 SQL 语句查询数据库的协同创建表，查询结果的关键属性包括协同 ID、协同标题。然后前端根据后端返回的结果，在文本区创建对应协同 ID 的文本编辑器，用于进行用户的协同编辑操作和获取协同编辑的内容。该文本编辑器的内容，是根据 WebSocket 创建的连接而非通过封装的 Axios 请求传递到后端的，后端定义的配置类将接收到的结果进行统一处理，用户可在前端页面再次通过 WebSocket 连接获取到后端处理后的结果，结果的关键属性包含了 WebSocket 连接 ID、连接的状态和编辑的文本内容。当一个登录用户在文本区编辑了内容后，WebSocket 会以全双工通信方式允许用户建立持久的连接，实现用户与后台实时传输、双向传输数据的功能。因此当有其他登录用户，在编辑同一个协同 ID 对应的文本区内容时，实际上是在建立的同一个 WebSocket 连接下进行数据通信，将编辑操

作、编辑内容封装为消息，再次通过连接广播给其他成员，体现为实时同步编辑的文本数据。以此来达到简易的协同编辑效果。

4.1.6.4 删除协同编辑模块

在删除协同部分，如果一个创建者需要删除自己创建的协同，则需要同时一起删除协同下的加入成员。前端将根据用户的点击事件获取待删除的协同 ID，并作为参数向后端发送请求，后端根据 SQL 语句查询数据库的协同成员表，删除此协同 ID 加入的全部成员，然后查询协同加入表，删除此操作用户 ID 操作的协同 ID 对应的协同编辑。最后再调用查询协同加入表，在前端页面上刷新删除后的视图界面。

4.1.6.5 协同编辑冲突处理

在协同编辑部分，通过 WebSocket 连接可以很好地解决消息广播消息的实时通信需求，但解决协同编辑的冲突问题仍然存在不足之处。如果当前协同编辑的拥有不止一个成员用户，就必须考虑协同编辑的冲突问题。常见的协同编辑的冲突包括更新时的编辑冲突、合并时的版本冲突、编辑时的权限冲突等，常用的解决方法已有悲观乐观锁机制、版本控制机制、实时同步机制等，每种方法针对不同的情形各有优缺点，本项目则采用操作转换（Operational Transformation, OT）的思想为协同编辑的冲突问题提供简单的解决方案，适用于协同用户量较少的应用场景。

前端通过用户输入向后端发送请求，并传递有关编辑操作的数据信息，包括编辑 ID、操作类型、操作内容、操作位置、操作时间等，后端根据这些信息进行操作转换处理，以解决可能发生的冲突。首先，后端将当前文档编辑的全部用户操作引入一个集合并且按照时间戳进行升序排序，然后判断当前操作类型，比如如果是“插入”文本，就需要根据时间戳适当地调整插入文本的位置：如果当前操作的位置大于或等于现有操作的位置，则将当前操作的位置向后移动，移动的距离为已有操作的内容长度。如有必要还可以进一步设置优先级和时间戳一起作为排序条件。

协同编辑模块，由于使用目标决定了功能的上限，本文档系统设计此模块的目的，则是为了更好地促进用户对于文档编辑的协作需求。在协同页面，用户可以通过点击按钮来“新建协同”或“加入协同”。协同编辑，顾名思义就是允许登录的不同用户，在同一个文档编辑区，对文档内容进行操作处理，而且不会因为多人交叉编辑发生合并冲突或污染数据的问题。在这个模块中，每一个登录用户都可以作为协同编辑的创建者，而且可以创建不止一个协同编辑，创建后可以查看当前协同文

档的信息，其中有一个唯一标识的协同号，可用于分享给其他用户加入合作。每当一个用户加入到初始人创建的协同时，在列表区即可看到加入此文档编辑的用户信息，进而增强了数据可视化、方便了操作性能。协同编辑模块采用 **WebSocket** 实现，这个配置来源于在 **Mybatis** 中添加的依赖，简单理解来说，它的核心功能就是监听用户互操作，并且可以将当前捕获到的操作以全局状态共享给整个服务器，因此所有登录的用户都可以获取到文档编辑器的更新内容。

4.2 管理后台功能部分

前台界面用于对已有的数据信息进行布局设计和合理展示，后台管理则对数据内容进行权限控制和统一管理，后台管理对于一个完整的系统来说是必不可少的，它的作用是将数据库操作变得可视化。管理后台功能部分主要包括了以下几个模块：用户信息管理模块、综合发布管理模块、协同发布管理模块、系统日志管理模块、系统公告管理模块。

管理部分是在用户功能的基础上实现的，管理员除了拥有更高权限还拥有系统公告管理、系统日志管理功能。根据项目特点，可以把系统的主要功能，分为“基础 CRUD”和其他部分，因为几乎每个模块都是在新增、查询、编辑、删除的基础上，附加其他功能实现的。用户操作界面时，前端发送请求，调用后端接口，后端通过 MyBatis 操作数据库并将数据结果返回给前端。在前端发送请求和后端返回结果之前，都要进行数据封装：前端通过“Request”封装请求消息；后端通过“Params”封装，用于接收前端参数；通过“Result”封装，用于返回最终结果。在后台调用方面，将按照 Controller、Service、Dao/Mapper、Entity 的顺序依次进行。

4.2.1 用户信息管理模块

在用户管理模块中，用户通过注册页面向后台数据库添加一条用户信息，管理员身份登入系统后，可对所有用户信息进行统一管理，包查询用户信息，更改用户信息，更改用户身份，添加用户信息，删除用户信息。

用户管理流程如图 4.10 所示：

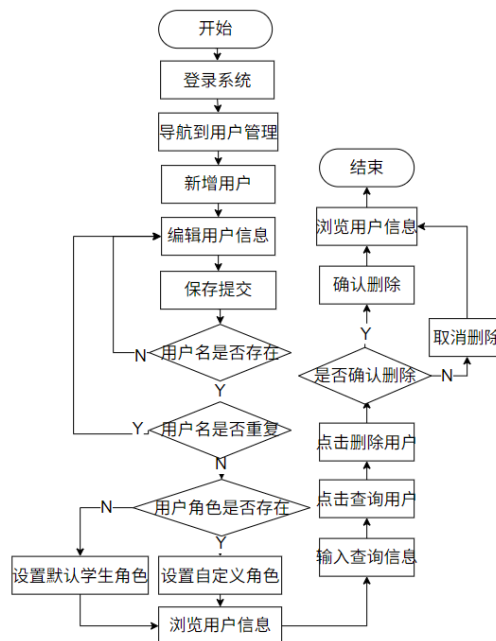


图 4.10 用户管理流程图

用户管理运行结果如图 4.11 所示：



<input type="text" value="请输入姓名"/> <input type="text" value="请输入电话"/> <input type="button" value="查询"/> <input type="button" value="清空"/> <input type="button" value="新增"/>						<input type="text" value="请输入用户名"/> <input type="button" value="查询"/> <input type="button" value="清空"/>		
姓名	性别	年龄	电话	角色	操作	用户ID	用户名	用户头像
张老师	女	1	18877778888	ROLE_TEACHER	<input type="button" value="编辑"/> <input type="button" value="删除"/>	5	张三	
赵老师	女	25	18866668888	ROLE_TEACHER	<input type="button" value="编辑"/> <input type="button" value="删除"/>			
张三	男	25	18855558888	ROLE_STUDENT	<input type="button" value="编辑"/> <input type="button" value="删除"/>			
李四	女	25	18844448888	ROLE_STUDENT	<input type="button" value="编辑"/> <input type="button" value="删除"/>	1	admin	
王五	女	25	18833338888	ROLE_STUDENT	<input type="button" value="编辑"/> <input type="button" value="删除"/>			
共 8 条 5条/页 < 1 2 >						共 2 条 5条/页 < 1 >		

图 4.11 用户管理运行图

4.2.2 用户发布管理模块

在用户发布管理模块中，登录用户通过新增发布页面向后台数据库添加一条发布信息，管理员身份登入系统后，可以对所有用户的发布内容进行统一管理，包括查询用户发布，更改用户发布，添加用户发布，删除用户发布。

用户发布管理流程如图 4.12 所示：

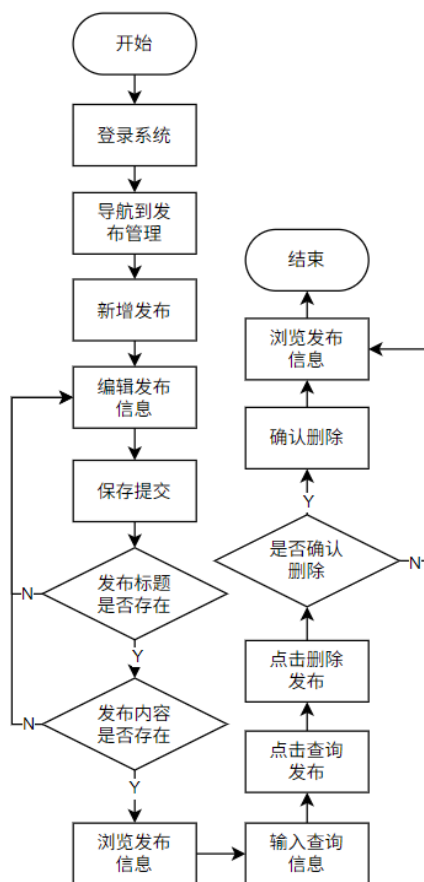


图 4.12 用户发布管理流程图

用户发布管理运行结果如图 4.13 所示：



图 4.13 用户发布管理运行图

4.2.3 系统日志管理模块

在日志管理模块中，系统后端使用 AOP 即面向切面编程技术，通过自定义一个注解来捕获用户操作，并返回相应的消息，以此完成日志模块。当用户在系统中进行操作如登录、编辑、删除等之后，系统便会捕获到当前的操作人、操作内容、操作时间和操作 IP，形成一条日志管理记录，并在系统日志页面进行加载。管理员可以查询系统日志，删除系统日志。

日志管理流程如图 4.14 所示：

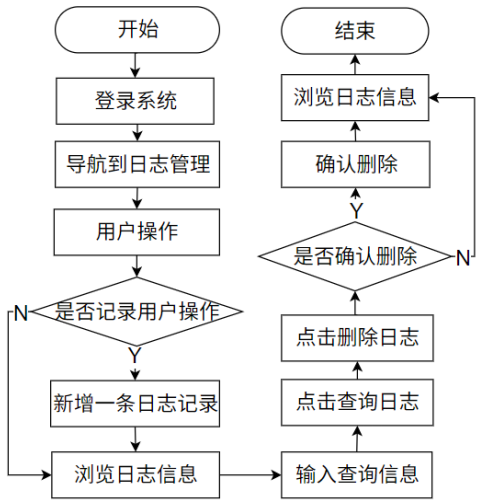


图 4.14 日志管理流程图

日志管理运行结果如图 4.15 所示：

请输入操作内容 请输入操作人 查询 清空			
操作时间	操作人	ip	操作
2025-05-22 13:19:31	admin	0:0:0:0:0:0:1	删除
2025-05-22 12:25:00	张三	0:0:0:0:0:0:1	确定删除吗? 取消 确定
2025-05-22 11:18:16	admin	0:0:0:0:0:0:1	删除
2025-05-22 09:13:08	admin	0:0:0:0:0:0:1	删除
2025-05-22 07:12:45	张三	0:0:0:0:0:0:1	删除
共 31 条 5条/页 < 1 2 3 4 5 6 7 >			

系统日志
05-24 09:20:03 admin
操作时间: 2025-05-24 09:20:03 操作用户: admin 操作IP: 0:0:0:0:0:0:1 操作内容: 登录该系统
05-23 17:13:37 admin
05-23 14:51:30 admin
05-22 23:19:31 admin

图 4.15 日志管理运行图

系统日志管理的详细设计过程如下。

日志记录部分采用了面向切面编程（Aspect-Oriented Programming, AOP）的原理，也就是 AOP 拦截机制。首先在后端需要创建一个自定义注解“@AutoLog”，它里面有一个 value 属性，就是用于记录日志的内容。比如需要记录登录日志，就在登录方法上面加上自定义的注解，并写上注解内容“登录系统”。而“@Around”环绕通知注解，就是用于监听“@AutoLog”注解的全部方法，当“@AutoLog”注解的方法被调用后，程序就会拦截当前方法的执行，切换到“@AutoLog”注解，先执行里面的方法。在环绕通知方法里面通过“autoLog.value()”方法即可获取注解内容，从而得到操作内容、操作用户、操作 IP 和时间等日志信息，保存到数据库。最后经过前端调用，返回给用户端。

4.2.4 系统公告管理模块

在公告管理模块中，用户可以在首页公告区查看新增的公告内容，也可以点击进入公告详情页面查看内容。管理员可以查询系统公告，编辑系统公告，新增系统公告，删除系统公告。

公告管理流程如图 4.16 所示：

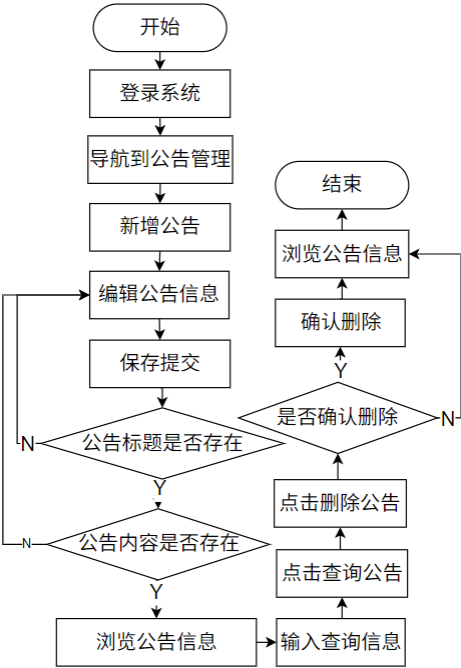


图 4.16 公告管理流程图

公告管理运行结果如图 4.17 所示：

请输入公告标题

查询

清空

新增

公告标题	公告内容	更新时间	操作
公告标题	555 666 777 8889	2025-06-03 14:35:34	<div>编辑</div> <div>删除</div>
新的公告标题	公告内容测试	2025-05-18 08:31:51	<div>编辑</div> <div>删除</div>
111		2025-05-15 21:20:41	<div>编辑</div> <div>删除</div>
公告标题	公告内容	2025-05-07 08:58:48	<div>编辑</div> <div>删除</div>
公告标题	公告内容	2025-05-07 08:58:45	<div>编辑</div> <div>删除</div>

共 10 条5条/页<12>

系统公告

05-18 08:31:51 新的公告标题

发布时间：2025-05-18 08:31:51
发布用户：admin
公告标题：新的公告标题
公告内容：公告内容测试

05-15 21:20:41 111

05-07 08:58:48 公告标题

05-07 08:58:45 公告标题

图 4.17 公告管理运行图

5 系统测试

5.1 测试方案

本次系统测试方案主要是测试该文档社区平台能否达到设计目标，功能是否完备。用户在使用过程中的交互效果是否能正确显示。在用户操作后台数据的过程中，是否能正确出现对应提示：如果操作成功，响应成功提示、返回成功数据；如果操作失败，响应失败提示，返回失败信息等。确保系统的前端界面不影响用户体验，且后端数据细节正确操作与同步。

测试方案将根据本文档社区系统的各个功能模块专门设计。包括登录注册模块、个人中心模块、用户发布模块、协同编辑模块、后台管理模块。

测试方法将不考虑代码逻辑和设计架构，仅从用户界面和系统功能角度完成测试。

5.2 测试设备及环境

用于测试本系统性能运行状况的测试设备、硬件环境和软件环境等。

本次测试设备及测试环境如下表 5.1 所示：

表 5.1 测试设备及测试环境

测试设备名称	硬件环境	软件环境
华硕 VivoBook 15 X 电脑	处理器：英特尔 11 代-i5 内存：16.0GB 外存：256GB 分辨率：1920x1080 像素	Windows11

5.3 功能测试

5.3.1 登录注册模块

在登录注册模块中，系统将对用户权限加以控制，阻止非法用户访问系统界面，并对合法用户进行放行。如表 5-1 所示：

表 5-1 登录注册模块测试

测试项目	操作	测试结果	测试结论
登录成功提示	输入正确用户名、密码和验证码	“登录成功”跳转到首页	成功
登录输入为空提示	空输入用户名或密码，点击登录按钮	“系统异常”对应弹框 3 秒后消失	成功
登录输入错误提示	输入错误用户名、密码，点击登录按钮	“用户名或密码输入错误”	成功
登录验证码错误提示	输入错误验证码	“验证码不正确”	成功
登录验证码图片刷新	点击验证码图片	验证码图片更新	成功
注册成功提示	输入未注册的用户名及其他注册信息	“注册成功”跳转到登录页	成功
注册用户名重复提示	输入已注册用户名	“用户名已存在”	成功
注册用户名为空提示	空输入用户名	“用户名不能为空”	成功
注册用户默认角色	空选择角色，点击注册按钮	注册用户默认角色为“ROLE_STUDENT”	成功
注册用户默认密码	空输入密码，点击注册按钮	注册用户默认密码为“123456”	成功

5.3.2 个人中心模块

在个人中心模块中，用户能够根据需要自定义个人信息，如修改初始化密码、修改个人信息等。如表 5-2 所示：

表 5-2 个人中心模块测试

测试项目	操作	测试结果	测试结论
编辑信息	编辑信息，并保存提交	“用户更新成功”	成功
头像上传	点击上传选择本地图片	用户头像更改	成功

5.3.3 用户发布模块

在综合发布模块中，用户用过图标可以成功创建分类、新增发布、上传发布文件、添加发布编辑。用户可以下载上传的文件，可以保存编辑的发布内容。用户可以通过图标删除分类、删除发布、删除文件、删除编辑。如表 5-3 所示：

表 5-3 用户发布模块测试

测试项目	操作	测试结果	测试结论
新增分类	点击新增分类图标 输入类别名称	新增一条分类	成功
删除分类	点击删除分类图标	删除一条分类	成功
新增发布	点击新增发布图标 输入发布标题	新增一条发布	成功
编辑发布	编辑发布内容	切换页面时被编辑的 发布内容保存更改	成功
提交发布	点击提交发布按钮	管理员身份登录， 可见待审核发布， 审核通过首页可见	成功
删除发布	点击删除发布图标	弹出气泡确认框 取消，保留发布 确认，删除发布	成功
上传文件	点击上传文件图标	弹出上传界面 选择上传文件 新增一条文件	成功
删除上传	点击删除文件图标	弹出气泡确认框 取消，保留文件 确认，删除文件	成功
下载文件	点击下载文件图标	确认下载，将指定文 件保存到本地	成功

5.3.4 协同编辑模块

在协同编辑模块中，用户可以新增协同编辑，可以加入已有的协同编辑，可以删除创建的协同编辑，不同登录用户的编辑内容能够同步显示。如表 5-4 所示：

表 5-4 协同编辑模块测试

测试项目	操作	测试结果	测试结论
新增协同编辑	点击新增协同选项	新增一条协同编辑	成功
加入协同编辑	点击加入协同选项	加入一条协同编辑	成功
进行协同编辑	登录不同用户，在文本区编辑内容	文本区同步不同用户编辑的不同内容	成功
删除协同编辑	点击删除协同按钮	弹出气泡确认框 取消，保留文件 确认，删除文件	成功

5.3.5 后台管理模块

在后台管理模块中，对本系统用户信息、用户发布、系统日志、系统公告进行统一管理，包括新增一条记录、编辑已有记录、删除一条记录、查询已有记录等操作。如表 5-5 所示：

表 5-5 后台管理模块测试

测试项目	操作	测试结果	测试结论
用户管理	对用户信息进行操作	增删改查用户信息	成功
日志管理	对日志信息进行操作	删改查日志信息	成功
公告管理	对公告信息进行操作	增删改查公告信息	成功
发布管理	对发布信息进行操作	增删改查发布信息	成功

6 结论

本次毕业设计的想法来源于平时在网上浏览的一些技术站点，在互联网世界中，随着技术的发展以及人们需求的增加，提升表达的形式已经和完善表达的内容同样重要，在这个信息大爆炸的时代，一个优秀的表达形式能够迅速在繁杂的信息洪流中脱颖而出，被大众所熟知，并迅速发酵传播，进而引发连锁效应。

我的设计项目主要是集成了一些很基础，但也比较重要常用的功能，虽然说网上已经有很多案例，但是对我来说是一个全新的尝试，里面也有很多技术点值得我去学习，而且也说明了这些功能虽然基础，但是却是不可或缺的功能。因为我的这个项目是一个文档社区平台，它的特点就是不同于单一的图文内容，还支持用户嵌入音视频这样的功能，也能上传不同格式的文件。特别是在教学、设计领域，多样化的形式，能够让用户更好地传达表达自己的思想。而单纯的文字或者是单纯的视频其实都有各自的缺点，我的出发点就是为了弥补这样的不足，在传统的文档社区上加以创新。虽然每一个功能拆开看很简单，但是要将他们全部实现也并不容易。

从人类传播知识途径的发展历程来看，早可追溯至远古时代的石刻结绳，古代的信书竹简，知识承载与传播的方式是多样且不断发展变化的，而人类对于信息的接收最终依赖于生物器官。当今现代的电子信息、多媒体载体、数字文化的发展，使我们正在接收更加多元化的信息接收方式，从大来看，广播电台、新闻报道、讲座演说、舞台演绎，这些都可以看做是信息宏观的传播载体，进一步划分载体则表现为一个 demo、一个网页、一个 App，再划分则包括图文、视频、音频这些更为基础的载体。这些都不过是日常生活中随处可见的信息形式，却极为重要地改变了我们的生活。每一种信息元素也衍生出了相应的承载方式，并通过更加细化到每一种对应的文件格式，以适配不同的应用场景。

本设计融创富文本音视频集合的文档社区平台旨在充分发挥在数字媒体技术专业和技术特长，在传统富文本集成的基础上，利用与新兴媒体的结合，为用户良好体验，更重要的是为逻辑表达的呈现、知识传播的方式提供更好的思路。

谢 辞

在完成毕业设计的过程中，我再次温习了很多以前学过或遗漏的知识，并且在不断发现解决新的问题中的到进步。这次的项目对我来说是一个挑战和考验，也是这大学四年来对学过知识的回顾和对自己的交代，我能顺利完成毕业设计，离不开这些老师曾经的倾囊相授，感谢你们。此次毕业设计包括毕业论文的完成都是在王丹东老师和任兆刚老师的指导下完成的，他们给予耐心讲解和灵活处理问题的方法，每一次材料的提交都认真修改，对我悉心指导，平易近人的态度让我如沐春风，由此我表示衷心的感谢！

经过本次为时几个月的毕业设计，我从中收获到了很多，包括但不限于对专业知识的深入实践，如训练独立解决问题的能力，还包括对生活的感悟与思考，如工作与休息时间的平衡协调。这些积累的经验与深刻的感悟与我个人而言是在未来的工作和生活将会收获与日俱增的助力，与此同时于社会而言则是令我逐渐成长为一个对人类科学事业发展更有用的帮手。

此外，同时也感谢网络上无私分享专业技术的开发者们，由于有着这样对技术充满热忱与不懈精神的你们，计算机技术得以日益壮大，我才得以完成这次毕业设计。光阴如梭，四年的大学生涯随着毕业设计的完成也将结束。感谢这些所有提供帮助和支持的人！

参考文献

- [1] 田海晴.基于 SpringBoot 和 Vue 框架的共享运营管理平台的设计与实现[D]. 山东大学,2020.DOI:10.27272/d.cnki.gshdu.2020.004528.
- [2] 吕学婷.基于 Springmvc 和 Mybatis 框架的门户网站及其内容管理系统的设计与实现[D].东华理工大学,2016.
- [3] 荣艳冬.关于 Mybatis 持久层框架的应用研究[J].信息安全与技术,2015,6(12):86-88.
- [4] 赵巧玲.基于 B/S 架构的软件项目开发[J].计算机光盘软件与应用,2014,17(24):130-131.
- [5] 张源伟,杨铭,郭昊.基于 PHP 技术的网络文件管理系统设计[J].物联网技术,2013,3(04):79-80+83.DOI:10.16667/j.issn.2095-1302.2013.04.028.
- [6] 邱祝文.基于 redis 的分布式缓存系统架构研究[J].网络安全技术与应用,2014,(10):52+54.
- [7] 瞿裕忠,张剑锋,陈峥,等.XML 语言及相关技术综述[J].计算机工程,2000,(12):4-6+30.
- [8] 欧阳桂秀.基于 Java 和 MySQL 的数据库管理系统的设计与实现[J].信息记录材料,2022,23(09):240-242.DOI:10.16009/j.cnki.cn13-1295/tq.2022.09.007.
- [9] 刘子凡,郭昱君.基于 SpringBoot+Mybatis 的个人博客系统设计与实现[J].现代信息科技,2021,5(08):104-107+111.DOI:10.19850/j.cnki.2096-4706.2021.08.029.
- [10] 陈涵. 视频影音后台管理系统的设计与实现[D]. 北京交通大学, 2020. DOI:10.26944/d.cnki.gbfju.2020.002180.
- [11] 丁峰,刁鸣.FFMPEG 的音视频格式转换设计[J].应用科技,2013,40(02):11-14.
- [12] 王宁.分享网站视频处理系统的设计与实现[D].北京工业大学,2016.
- [13] 冯瑞成.基于 React 的设备管理系统前端框架的设计[D].华中科技大学,2018.
- [14] 庄丽君,汤海林.基于 Java Web 技术的校园论坛网页的设计与实现[J].现代信息科技,2024,8(15):74-77.DOI:10.19850/j.cnki.2096-4706.2024.15.016.
- [15] 朱军.基于 JavaWeb 的编程技术论坛的设计与实现[J].电子制作,2022,30(10):51-54.DOI:10.16589/j.cnki.cn11-3571/tn.2022.10.024.
- [16] 李鹏飞.基于 Web 技术的校园论坛设计与实现[D].内蒙古科技大学,2019.DOI:10.27724/d.cnki.gnmkg.2019.000570.

- [17] 杨世文,侯超钧.基于 SSM 框架的学术论坛管理系统设计与实现[J].计算机时代,2021,(02):25-28+33.DOI:10.16644/j.cnki.cn33-1094/tp.2021.02.007.
- [18] 周杰.基于 Spring MVC 的教务管理系统[J].电子技术与软件工程,2020,(04):203-207.
- [19] 葛萌,黄素萍,欧阳宏基.基于 Spring MVC 框架的 Java Web 应用[J].计算机与现代化,2018,(08):97-101.
- [20] 张峰.应用 SpringBoot 改变 web 应用开发模式[J].科技创新与应用,2017,(23):193-194.
- [21] 陈倩怡,何军.Vue+Springboot+MyBatis 技术应用解析[J].电脑编程技巧与维护,2020,(01):14-15+28.DOI:10.16184/j.cnki.comprg.2020.01.005.
- [22] 单树倩,任佳勋.基于 SpringBoot 和 Vue 框架的数据库原理网站设计与实现[J].电脑知识与技术,2021,17(30):40-41+50.DOI:10.14004/j.cnki.ckt.2021.2868.