

# 面向学习交流的知识共享平台设计与实现<sup>\*</sup>

李文鑫 林宁

左悦<sup>\*\*</sup>

南宁学院信息工程学院, 南宁 530200

南宁学院土木与建筑工程学院, 南宁 530200

**摘要** 随着互联网技术的飞速发展,知识共享已成为学习和信息传播的重要方式。传统的知识分享模式存在许多问题,如信息不对称、传播效率低等。为此,本文设计并实现了一个基于SSM框架的知识分享平台,旨在为用户提供一个便捷、高效的在线学习与互动交流环境。平台采用SSM架构进行后端开发,前端使用Vue3框架,并通过MySQL数据库存储数据。平台实现了注册登录、发帖评论、点赞收藏等功能,并具备管理员后台管理与数据统计功能。通过系统的实现与测试,证明其具有良好的可扩展性与用户体验,为知识分享平台的建设提供了参考。

**关键字** 知识分享平台; Vue3; MySQL; SSM 架构

## Design and Implementation of a Knowledge Sharing Platform for Learning and Communication

LiWenXin LinNing

ZuoYue<sup>\*\*</sup>

College of Information Engineering  
Nanning University  
Nanning 530200 China

College of Architecture and Civil Engineering  
Nanning University  
Nanning 530200 China

**Abstract**—With the rapid development of internet technology, knowledge sharing has become a crucial means of learning and information dissemination. Traditional knowledge sharing models suffer from numerous challenges, such as information asymmetry and inefficient dissemination. To address this, this paper designs and implements a knowledge sharing platform based on the SSM framework, aiming to provide users with a convenient and efficient online learning and interactive communication environment. The platform utilizes the SSM architecture for back-end development, the Vue3 framework for the front-end, and a MySQL database for data storage. The platform implements functions such as registration and login, posting comments, liking and adding favorites, and includes administrator backend management and data statistics. Through system implementation and testing, the system demonstrates excellent scalability and user experience, providing a reference for the development of knowledge sharing platforms.

**Keywords**—Knowledge sharing platform; Vue3; MySQL; SSM architecture

## 1 引言

随着信息技术的不断进步,知识分享平台已经成为现代社会学习和沟通的重要工具。传统的线下学习方式和信息传播手段,已经无法满足快速增长的知识需求和用户的互动需求。因此,基于现代Web技术构建高效、便捷的知识分享平台,已成为提升学习效率、推动知识传播的重要途径。本文基于SSM框架,结合Vue3技术,设计并实现了一个集知识发布、互动交流与后台管理为一体的知识分享平台<sup>[1]</sup>。该平台通过SSM架构和MySQL数据库的结合,确保了系统的高效性与稳定性,旨在为用户提供一个便捷、互动性强的学习与知识分享环境。

## 2 相关技术与工具介绍

<sup>\*</sup> 基金资助: 南宁学院 2025 年横向课题 (2025HX004) 资助。

<sup>\*\*</sup> 通讯作者: 左悦 zuoyue@unn.edu.cn

### 2.1 SSM 框架介绍

SSM 框架是由 Spring、SpringMVC 和 MyBatis 组成的主流 Java Web 开发框架<sup>[2]</sup>。Spring 主要负责业务对象的管理与事务控制, SpringMVC 用于处理 Web 层的请求和响应, 而 MyBatis 则提供了灵活高效的持久层解决方案。三者相互配合, 使得系统架构清晰、层次分明, 既保证了代码的可维护性和扩展性, 又提升了开发效率<sup>[3]</sup>。

### 2.2 Vue3 前端框架介绍

Vue3 是一个用于构建现代 Web 界面的 JavaScript 框架。它采用了响应式数据绑定和组件化开发模式, 能够实现高效、灵活的前端开发。Vue3 相较于 Vue2 在性能和语法结构上做了优化, 提供了更简洁的 API 和更强的扩展性, 能够支持复杂的单页应用 (SPA) 开发, 适合用于本系统的前端开发<sup>[4]</sup>。

### 2.3 MySQL 与 MyBatis Plus 介绍

MySQL 是一个流行的关系型数据库，具有高效的查询能力和良好的扩展性，广泛应用于各类 Web 应用中。MyBatis Plus 是 MyBatis 的增强工具，提供了简化数据库操作的功能，如自动生成 CRUD 操作、分页查询和性能优化等，能够大大提升开发效率并降低出错率。

3 系统设计

3.1 系统架构设计

本平台采用分层架构设计，确保系统的模块化、高效性和可扩展性。系统分为表现层、控制层、服务层、持久层和数据库层，每层有清晰的职责，且能够有效支持功能的扩展和维护<sup>[5]</sup>。

表现层负责用户界面的呈现与交互，采用 Vue3 框架实现响应式组件化开发，前端页面通过组件管理和数据绑定，提高了开发效率和用户体验。Vue3 的响应式机制使得页面能够及时响应用户操作，并高效地更新页面内容。前端与后端的通信通过 Axios 实现，前端通过 HTTP 请求与后端进行数据交换

控制层则负责接收并处理来自前端的请求，使用 SpringMVC 进行请求分发。控制层将请求转发到具体的服务层进行业务处理，并将结果返回给前端。为了实现接口的高效管理和调用，控制层采用了 RESTful API 风格的接口设计，确保了系统接口的清晰和一致性。

服务层是系统的核心，处理具体的业务逻辑。在服务层，系统会根据前端传来的数据进行相应的计算和处理，然后调用持久层与数据库进行数据操作。服务层的设计确保了业务规则的封装和逻辑清晰，使得系统的扩展更加简便。

持久层基于 MyBatis Plus 操作 MySQL 数据库，负责系统中的数据持久化工作。MyBatis Plus 是 MyBatis 的增强工具，能够简化数据库操作，提升了开发效率和代码的简洁性。持久层实现了对数据库中各类实体（如用户、帖子、评论等）的增删查改功能，确保了数据与业务逻辑的良好协作。

数据库层则是系统的数据存储和管理部分。平台使用 MySQL 数据库来存储所有平台数据，包括用户信息、帖子内容、评论记录等。数据库表设计合理，并通过外键关系将不同数据表连接起来，确保了数据的完整性和一致性。为了应对高并发和海量数据，系统在数据库层设计了合理的索引和查询策略，保证了数据存取的高效性。

这种分层架构的设计充分利用了各层职责明确、解耦性强的特点，保障了系统在功能扩展、维护和高

并发情况下的高效性。

3.2 系统功能模块设计

系统功能以用户为核心，围绕内容发布与知识交流构建模块化服务。用户模块支持注册、登录、修改资料、查看粉丝与关注者等功能；帖子模块支持用户发布图文内容、添加图片、删除自己的帖子并接受他人互动；评论与点赞模块构成了平台的核心互动机制，结合消息模块对用户行为反馈提供通知提醒<sup>[6]</sup>。搜索模块则提升了信息获取效率，支持关键字模糊匹配查找帖子、用户或视频资源。管理员通过后台管理模块可监控用户行为，上传学习视频，处理违规内容，并通过活跃度图表观察平台运行状态<sup>[7]</sup>。

3.3 系统 E-R 图

系统总体 E-R 图如图 3-1 所示，涵盖了用户、帖子、评论、视频、消息、点赞、收藏、关注等核心业务实体及其间的关系，清晰地展示了数据库设计的逻辑结构与实体之间的关联：

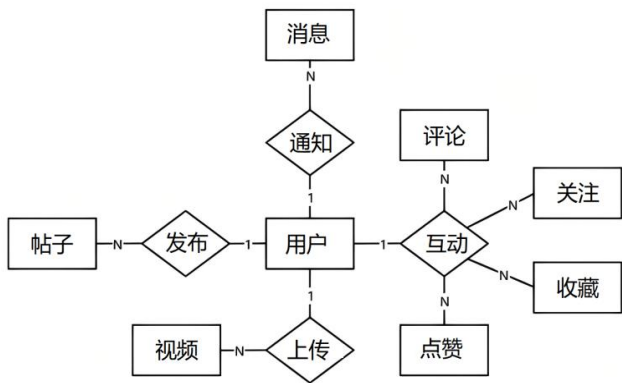


图 3-1 系统总体 E-R 图

3.4 数据库设计

系统采用 MySQL 数据库进行数据存储，设计了多个数据表来支持平台的各项功能。主要的数据库表包括：

(1) 用户表：

用户表的说明如表 3-1 所示。

(2) 帖子表：

帖子表的说明如表 3-2 所示。

(3) 评论表：

评论表的说明如表 3-3 所示。

(4) 消息表

消息表的说明如表 3-4 所示。

表 3-1 用户表

字段名称	类型	长度	空值	说明
user_id	int	10	否	用户编号
user_name	varchar	64	否	用户名
password	varchar	32	否	用户密码
phone	bigint	11	是	手机号
email	varchar	64	是	用户邮箱
region	varchar	32	是	地区
age	int	4	是	年龄
birthday	datetime	0	是	生日
gender	tinyint	2	否	性别
introduction	varchar	256	是	简介
registration_time	datetime	0	否	注册时间
school	varchar	32	是	毕业学校
profession	varchar	32	是	专业
degree	varchar	32	是	学历
graduate_time	datetime	0	是	毕业时间
avatar	varchar	128	是	头像
likes	int	32	是	获赞量
focus	int	16	是	关注量
fans	int	16	是	粉丝量
role	int	2	是	用户角色

表 3-2 帖子表

字段名称	类型	长度	空值	说明
content_id	int	10	否	帖子编号
user_id	int	10	否	用户编号
title	varchar	128	否	标题
detail	text		否	详细内容
publish_time	datetime	0	否	发布时间
like	int	8	是	好评量
dislike	int	8	是	差评量
images	text		是	图片路径

表 3-3 评论表

字段名称	类型	长度	空值	说明
id	int	10	否	评论编号
user_id	int	10	否	用户编号
content_id	int	10	否	帖子编号
comments_detail	varchar	256	否	评论内容
comments_time	datetime	0	否	评论时间

表 3-4 消息表

字段名称	类型	长度	空值	说明
id	bigint	10	否	消息编号
to_user_id	int	10	否	接收消息的用户 ID
from_user_id	int	10	否	发送消息的用户 ID
type	varchar	20	否	消息类型
content_id	int	10	是	帖子编号
content	text		是	消息内容
create_time	datetime	0	否	创建时间
is_read	tinyint	1	是	是否已读

## 4 核心功能实现

### 4.1 用户模块实现

用户模块负责用户的注册、登录、密码找回等功能。在注册时，系统会检查用户名是否已被使用，确保用户名的唯一性。登录时，通过输入的用户名和密码与数据库中的记录进行比对，验证身份。

注册：

```
@Override
public Result register(Users users) {
    try {
        if (this.lambdaQuery().eq(Users::getUserName,
            users.getUserName()).exists()) {
            return Result.err(501, "用户名已存在");
        }

        users.setFans(0);
        users.setLikes(0);
```

```

        users.setFocus(0);
        users.setAvatar("upload/default.jpg");
        users.setRegistrationTime(Timestamp.valueOf(LocalDateTime.now()));
        this.save(users);
        return Result.ok("注册成功");
    } catch (Exception e) {
        log.error("注册失败: username={}",
            users.getUserName(), e);
        return Result.err(500, "注册失败: " +
            e.getMessage());
    }
}

登录:
@Override
public Map<String, Object> login(String
    username, String password) {
    try {
        Users user = this.lambdaQuery()
            .eq(Users::getUserName, username).one();
        if (user != null &&
            password.equals(user.getPassword())) {
            String token =
                jwtUtil.generateToken(user.getId(),
                    user.getUserName());
            Map<String, Object> userInfo = new
                HashMap<>();
            userInfo.put("userId", user.getId());
            userInfo.put("userName", user.getUserName());
            userInfo.put("avatar", user.getAvatar());

            userInfo.put("role", user.getRole());
            userInfo.put("token", token);
            return userInfo;
        }
        return null;
    } catch (Exception e) {
        log.error("登录失败: username={}", username,
            e);
        throw new RuntimeException("登录失败", e);
    }
}

```

## 4.2 帖子管理模块实现

帖子管理模块实现了帖子发布、编辑、删除、查看等功能。用户可以通过前端页面发布帖子，帖子内容和相关信息会存储到数据库中，并返回发布成功的提示。管理员可以对发布的帖子进行审核、删除等操作。

```

@Override
public Result createContent(Content content) {
    try {
        if
            (StringUtils.isBlank(content.getTitle())) {
            return Result.err(501, "标题不能为空");
        }
        if
            (StringUtils.isBlank(content.getDetail())) {
            return Result.err(501, "内容不能为空");
        }
        Content contentEntity = new Content();
        contentEntity.setTitle(content.getTitle());
        contentEntity.setDetail(content.getDetail());
        contentEntity.setImages(content.getImages());
        contentEntity.setUserId(content.getUserId());
        contentEntity.setPublishTime(new Date());
        contentEntity.setLike(0);
        contentEntity.setDislike(0);
        boolean success =
            this.save(contentEntity);
        return success ? Result.ok("发布成功") : Result.err(501, "发布失败");
    } catch (Exception e) {
        log.error("发布内容失败", e);
        throw new RuntimeException("发布失败: " + e.getMessage());
    }
}

```

## 4.3 评论模块实现

评论和互动模块支持用户对帖子进行评论、点赞和收藏等操作。每条评论都与相应的帖子和用户关联，并实时更新数据库中的数据。用户的点赞和收藏记录也会同步更新。

```

@PostMapping("/sendcomments")
public Result sendComment(@RequestBody
    Comments comments, HttpServletRequest request)
{
    Integer userId =
        tokenService.getCurrentUserId(request);
    if (userId == null) {
        return Result.err(401, "未登录");
    }
    Integer contentId =
        comments.getContentId();
    QueryWrapper<Content> queryWrapper = new
        QueryWrapper<>();
    queryWrapper.eq("content_id", contentId);
    Content content =

```

```

contentMapper.selectOne(queryWrapper);
    if (content == null) {
        throw new RuntimeException("内容不存在");
    }
    Integer authorId = content.getUserId();
    // 不给自己发消息
    if (!authorId.equals(userId)) {
        messagesService.createMessage(
            authorId,
            userId,
            "COMMENT",
            contentId,
            comments.getCommentsDetail()
        );
    }
    comments.setUserId(userId);
    return
    commentsService.addComments(comments);
}

```

#### 4.4 后台管理模块实现

后台管理模块提供了用户管理、帖子管理、数据统计等功能。管理员可以在后台管理界面查看平台的活跃度、用户行为等数据，并进行数据的增删改查操作。

```

@Override
public Statistics getStatistics() {
    try {
        // 获取总用户数（排除管理员）
        QueryWrapper<Users> userQuery = new
        QueryWrapper<>();
        userQuery.ne("user_id", 1);
        Long totalUsers = count(userQuery);
        // 获取今日活跃用户数
        Long activeUsers =
        baseMapper.getActiveUsersToday();
        // 获取今日发帖数
        QueryWrapper<Content> contentQuery =
        new QueryWrapper<>();
        contentQuery.apply("DATE(publish_time) =
        CURDATE()");
        Long todayPosts =
        contentMapper.selectCount(contentQuery);
        // 获取总互动数

```

```

        Long totalComments =
        commentsMapper.selectCount(null);
        Long totalFollows =
        followMapper.selectCount(null);
        Long totalInteractions = totalComments
        + totalFollows+ totalActions + totalStar;
        return Statistics.builder()
            .totalUsers(totalUsers)
            .activeUsers(activeUsers)
            .todayPosts(todayPosts)
            .totalInteractions(totalInteractions)
            .build();
    } catch (Exception e) {
        log.error("获取统计数据失败", e);
        throw new RuntimeException("获取统计数
        据失败");
    }
}

```

## 5 结束语

本文设计并实现了一个基于 SSM 框架的知识分享平台，系统实现了用户注册、帖子管理、评论互动等核心功能，具有良好的用户体验和管理界面。未来，可以进一步引入 AI 技术进行内容推荐，提升平台的智能化水平；同时，移动端支持也是系统未来优化的一个方向。

## 参考文献

- [1] 周洪宇,魏小迪.基于SSM框架的人脸识别智能取件系统的研究与实现[J].计算机技术与教育学报,2025:177-181.DOI:10.26914/c.cnkihy.2025.012288.
- [2] 曹华山.SSM框架在Web应用开发中的设计与实现[J].无线互联科技,2021,18(11):108-109.
- [3] 欧阳宏基,葛萌,程海波.MyBatis框架在数据持久层中的应用研究[J].微型电脑应用,2023,39(01):73-75.
- [4] 刘亚茹,张军.Vue.js框架在网站前端开发中的研究[J].电脑编程技巧与维护,2022,(01):18-19+39.DOI:10.16184/j.cnki.comprg.2022.01.009.
- [5] 左悦,林宁.基于SpringBoot的古建筑数字化信息管理系统设计与实现[J].计算机技术与教育学报,2023:142-147.DOI:10.26914/c.cnkihy.2023.109404.
- [6] 王辰霏,郑秋汉.基于云计算的高校教学资源共享平台设计与实现[J].自动化应用,2025,66(14):253-255+259.DOI:10.19769/j.zdhy.2025.14.070.
- [7] 林宁,左悦.基于云平台的C语言课程学习平台设计与实现[J].计算机技术与教育学报,2023:179-183.DOI:10.26914/c.cnkihy.2023.109411.