



太原理工大学  
TAIYUAN UNIVERSITY OF TECHNOLOGY

## 课程设计 小组报告

课程名称：\_\_\_\_\_Web 系统课程设计\_\_\_\_\_

项目名称：\_\_\_\_\_基于 Vue3 + SpringBoot 的前后端分离的  
个人博客系统的设计\_\_\_\_\_

专业班级：\_\_\_\_\_软件 2126 班\_\_\_\_\_

组长姓名：\_\_\_\_\_郑申\_\_\_\_\_

组员姓名：\_\_\_\_\_王浩仲、张羽喆、辛子昂\_\_\_\_\_

指导教师：\_\_\_\_\_廖丽娟\_\_\_\_\_

2023 年 06 月 20 日

## 太原理工大学课程设计任务书

学生姓名	郑申、王浩仲、张羽喆、 辛子昂	专业班级	软件 2126 班
课程名称	Web 系统课程设计		
设计名称	基于 Vue3 + SpringBoot 的前后端分离的 个人博客系统	设计周数	3
设计 任务 主要 设计 内容	<p>1. 搭建开发环境：根据具体需求配置开发环境，例如安装和配置 Vue3 和 SpringBoot 的开发环境。</p> <p>2. 分组和选择设计题目：根据指导教师的要求，进行分组，并选择基于 Vue3 + SpringBoot 的前后端分离的个人博客系统作为设计题目。</p> <p>3. 前台系统设计：用户注册和登录功能、文章浏览和搜索功能、音乐检索和播放功能、访客评论和回复功能；</p> <p>后台系统设计：写博文功能、系统管理功能、内容管理功能。</p> <p>4. 完成小组课设报告：小组成员合作完成课设报告，包括团队分工、时间进度安排、需求分析、系统设计、系统实现等内容。</p>		
设计要求	<p>1. 开发、设计与运行环境要求： 后台：JDK8+SpringBoot2+mybatisplus+mysql5+redis， 前台：Vue3+typescript+vite 等。</p> <p>2. 前台系统和后台系统应分别实现前端和后端的功能。</p> <p>3. 前台系统主要包括用户注册和登录功能、文章浏览和搜索功能、音乐检索和播放功能、访客评论和回复功能。</p> <p>4. 后台系统主要包括写博文功能和系统管理功能、内容管理功能。</p>		
主要参考 资    料	<p>1. Spring Boot 官方文档 [OL]. [引用网址: <a href="https://springdoc.cn/spring-boot/">https://springdoc.cn/spring-boot/</a>] (引用日期: 2023 年 7 月 1 日)</p> <p>2. MyBatis Plus 官方文档 [OL]. [引用网址: <a href="https://baomidou.com/pages/24112f/">https://baomidou.com/pages/24112f/</a>] (引用日期: 2023 年 7 月 1 日)</p> <p>3. Redis 官方文档 [OL]. [引用网址: <a href="https://redis.io/docs/">https://redis.io/docs/</a>] (引用日期: 2023 年 7 月 1 日)</p> <p>4. 程沛权. 前端工程化：基于 Vue.js 3.0 的设计与实践 [M]. 机械工业出版社, 2023.</p> <p>5. 张海藩, 牟永敏. 软件工程导论（第 6 版）[M]. 清华大学出版社, 2013.</p>		
学生提交 归档文件	<p>项目小组必须提交一份完整的项目技术说明文档（纸质版和电子版），文档内容必须包含团队分工、时间进度安排、需求分析、系统设计、系统实现等部分。</p>		

注：1.课程设计完成后，学生提交的归档文件应按照：封面—任务书—正文的顺序进行装订上交。

2.可根据实际内容需要续表，但应保持原格式不变。

指导教师签名：

日期：2023.06.20

# 目录

一、项目简介.....	1
(一)    项目概述.....	1
(二)    项目背景.....	1
(三)    项目目标.....	1
二、项目分工.....	2
(一)    团队成员.....	2
(二)    角色分配.....	2
(三)    责任和职责.....	2
(四)    分工情况表.....	2
三、进度安排.....	3
(一)    项目计划.....	3
(二)    里程碑.....	3
(三)    时间表.....	4
四、需求分析.....	5
(一)    用户需求.....	5
(二)    功能需求.....	7
(三)    非功能需求.....	8
五、总体设计.....	9
(一)    系统整体架构（共 12 个模块）.....	9
(二)    系统数据流图.....	9
(三)    数据库设计.....	9
六、详细设计.....	13
(一)    模块设计.....	13
七、编码与测试.....	18
(一)    代码目录总览.....	18
(二)    局部代码展示.....	21
(三)    针对各模块的单元测试.....	21
(四)    集成测试.....	21
八、结论与展望.....	22
(一)    项目总结.....	22
(二)    存在问题.....	22
(三)    后续改进计划.....	22

## 一、项目简介

### （一）项目概述

本项目是一个个人博客系统，采用前后端分离的架构设计。前台系统包括用户登录注册、文章浏览、评论回复、音乐播放等功能，用户可以注册登录账号，在前台浏览和搜索文章，与其他用户进行互动和评论，还可以欣赏音乐。后台系统提供文章撰写、系统管理和内容管理等功能，管理员可以编写、发布和管理文章，对用户进行权限管理以及对系统和内容进行管理。

### （二）项目背景

随着互联网的快速发展，个人博客成为了人们分享经验、表达观点和展示作品的重要平台。然而，市场上的博客系统大多数功能繁杂，使用不便，用户体验欠佳，因此需要一个简洁、易用、功能完善的个人博客系统来满足用户的需求。本项目针对这一需求而设计，并结合前后端分离的技术架构，旨在打造一款优秀的个人博客系统，提供良好的用户体验和管理功能，满足用户和管理员的各种需求。

### （三）项目目标

随着互联网的快速发展，个人博客成为了人们分享经验、表达观点和展示作品的重要平台。然而，市场上的博客系统大多数功能繁杂，使用不便，用户体验欠佳，因此需要一个简洁、易用、功能完善的个人博客系统来满足用户的需求。本项目针对这一需求而设计，并结合前后端分离的技术架构，旨在打造一款优秀的个人博客系统，提供良好的用户体验和管理功能，满足用户和管理员的各种需求。

二、项目分工																	
<p>(一) 团队成员</p> <p>本项目的团队共有 4 名成员，包括郑申、辛子昂、王浩仲和张羽喆。</p>																	
<p>(二) 角色分配</p> <p>在团队中，郑申担任项目总负责人的角色，负责整个项目的构想、设计和落地，并负责前端工程的设计与实现。</p> <p>王浩仲担任项目副负责人的角色，负责后端部分的实现，并与总负责人一起负责前后端项目的对接工作。</p> <p>辛子昂和张羽喆作为组员，分别在前端小组和后端小组中工作，在各自组长的指导下完成各模块的具体实现。</p>																	
<p>(三) 责任和职责</p> <p>郑申作为项目总负责人，负责整个项目的把控，确保项目的有序进行和高质量的完成。他还承担了前端工程的设计与实现的任务，保证前台前端和后台前端两个总模块的顺利实现。</p> <p>王浩仲作为项目副负责人，负责后端部分的实现工作，同时与总负责人合作完成前后端的对接，确保项目按时上线。</p> <p>辛子昂和张羽喆作为组员，分别在前端小组和后端小组中工作，根据组长的指导完成各自模块的具体实现，是项目中不可或缺的人员。</p> <p>他们共同努力，协作完成项目的开发工作，确保项目的顺利进行和高质量的交付。</p>																	
<p>(四) 分工情况表</p> <p>综上所述，整个团队的分工情况如表 1-1 所示。</p> <p style="text-align: center;">表 1-1 团队分工情况表</p> <table> <tr> <th>团队成员</th><th>角色</th><th>职责</th></tr> <tr> <td>郑申</td><td>项目总负责人</td><td> <ul style="list-style-type: none"> <li>- 负责整体项目规划和管理</li> <li>- 指导团队成员</li> <li>- 设计与实现前端工程</li> </ul> </td></tr> <tr> <td>王浩仲</td><td>项目副负责人</td><td> <ul style="list-style-type: none"> <li>- 负责后端部分的实现</li> <li>- 与总负责人协作进行前后端对接</li> <li>- 协助项目进度管理</li> </ul> </td></tr> <tr> <td>辛子昂</td><td>前端小组成员</td><td> <ul style="list-style-type: none"> <li>- 实现前台前端模块的功能和界面设计</li> <li>- 与后端小组对接并进行集成测试</li> </ul> </td></tr> <tr> <td>张羽喆</td><td>后端小组成员</td><td> <ul style="list-style-type: none"> <li>- 实现后台后端模块的功能和 API 设计</li> <li>- 与前端小组对接并进行集成测试</li> </ul> </td></tr> </table>			团队成员	角色	职责	郑申	项目总负责人	<ul style="list-style-type: none"> <li>- 负责整体项目规划和管理</li> <li>- 指导团队成员</li> <li>- 设计与实现前端工程</li> </ul>	王浩仲	项目副负责人	<ul style="list-style-type: none"> <li>- 负责后端部分的实现</li> <li>- 与总负责人协作进行前后端对接</li> <li>- 协助项目进度管理</li> </ul>	辛子昂	前端小组成员	<ul style="list-style-type: none"> <li>- 实现前台前端模块的功能和界面设计</li> <li>- 与后端小组对接并进行集成测试</li> </ul>	张羽喆	后端小组成员	<ul style="list-style-type: none"> <li>- 实现后台后端模块的功能和 API 设计</li> <li>- 与前端小组对接并进行集成测试</li> </ul>
团队成员	角色	职责															
郑申	项目总负责人	<ul style="list-style-type: none"> <li>- 负责整体项目规划和管理</li> <li>- 指导团队成员</li> <li>- 设计与实现前端工程</li> </ul>															
王浩仲	项目副负责人	<ul style="list-style-type: none"> <li>- 负责后端部分的实现</li> <li>- 与总负责人协作进行前后端对接</li> <li>- 协助项目进度管理</li> </ul>															
辛子昂	前端小组成员	<ul style="list-style-type: none"> <li>- 实现前台前端模块的功能和界面设计</li> <li>- 与后端小组对接并进行集成测试</li> </ul>															
张羽喆	后端小组成员	<ul style="list-style-type: none"> <li>- 实现后台后端模块的功能和 API 设计</li> <li>- 与前端小组对接并进行集成测试</li> </ul>															

### 三、进度安排

#### (一) 项目计划

##### (一) 收集技术资料阶段：2 天

- 1) 确定分组及成员角色
- 2) 确定每组设计选题
- 3) 理解选题设计任务
- 4) 查阅收集相关资料和明确系统需求

##### (二) 确定总体设计方案：1 天

- 1) 进行方案论证比较
- 2) 确定总体设计方案
- 3) 绘制系统整体功能结构图
- 4) 完成该阶段文档

##### (三) 详细设计：3 天

- 1) 设计系统各功能模块具体实现方法
- 2) 使用时序图或流程图等工具表示设计
- 3) 进行数据库设计
- 4) 完成该阶段文档

##### (四) 编码与实现：4 天

- 1) 采用相关开发技术完成各模块的具体编码和功能实现

##### (五) 集成、测试与完善：3 天

- 1) 进行系统的集成、调试和测试
- 2) 完善各功能，修复问题和 bug

##### (六) 完成课程设计报告：1 天

- 1) 按照课程设计报告的要求，编写设计报告

#### (二) 里程碑

##### 1. 里程碑 1：收集技术资料阶段完成

- 开始日期：2023-06-20
- 结束日期：2023-06-21

##### 2. 里程碑 2：确定总体设计方案完成

- 开始日期：2023-06-22
- 结束日期：2023-06-22

##### 3. 里程碑 3：详细设计完成

- 开始日期：2023-06-23
- 结束日期：2023-06-26

4. 里程碑 4：编码与实现完成

- 开始日期：2023-06-27

- 结束日期：2023-07-02

5. 里程碑 5：集成、测试与完善完成

- 开始日期：2023-07-03

- 结束日期：2023-07-05

6. 里程碑 6：完成课程设计报告

- 开始日期：2023-07-06

- 结束日期：2023-07-06

（三） 时间表

综上所述，项目的时间安排如下表 3-1 所示。

表 3- 1 进度表

阶段	开始日期	结束日期
收集技术资料阶段	2023-06-20	2023-06-21
确定总体设计方案	2023-06-22	2023-06-22
详细设计	2023-06-23	2023-06-26
编码与实现	2023-06-27	2023-07-02
集成、测试与完善	2023-07-03	2023-07-05
完成课程设计报告	2023-07-06	2023-07-06

## 四、需求分析

### （一） 用户需求

经过五次需求分析会议，总结出如下的用户需求

#### 1. 访客需求

- 访客希望能够进行用户登录和注册操作。
- 访客希望能够浏览文章列表。
- 访客希望能够查看具体的文章内容。
- 访客希望能够按照分类浏览文章。
- 访客希望能够根据标签搜索相关文章。
- 访客希望能够查看热门文章。
- 访客希望能够查看友情链接。
- 访客希望能够在文章下方进行评论。
- 访客希望能够回复其他用户的评论。
- 访客希望能够在前台浏览音乐厅。
- 访客希望能够查看推荐单曲和推荐歌单。
- 访客希望能够进行全局音乐播放。

#### 2. 管理员需求

- 管理员希望能够增加、删除、修改和查询用户信息。
- 管理员希望能够进行角色分配和权限控制以及可访问选项卡的设置。
- 管理员希望能够撰写和发布文章。
- 管理员希望能够管理文章分类，包括增加、删除、修改和查询分类信息。
- 管理员希望能够管理文章标签，包括增加、删除、修改和查询标签信息。
- 管理员希望能够增加、删除、修改和查询友情链接信息。
- 管理员希望能够增加、删除、修改和查询后台子菜单信息。

用例图如图 4-1、4-2 所示：



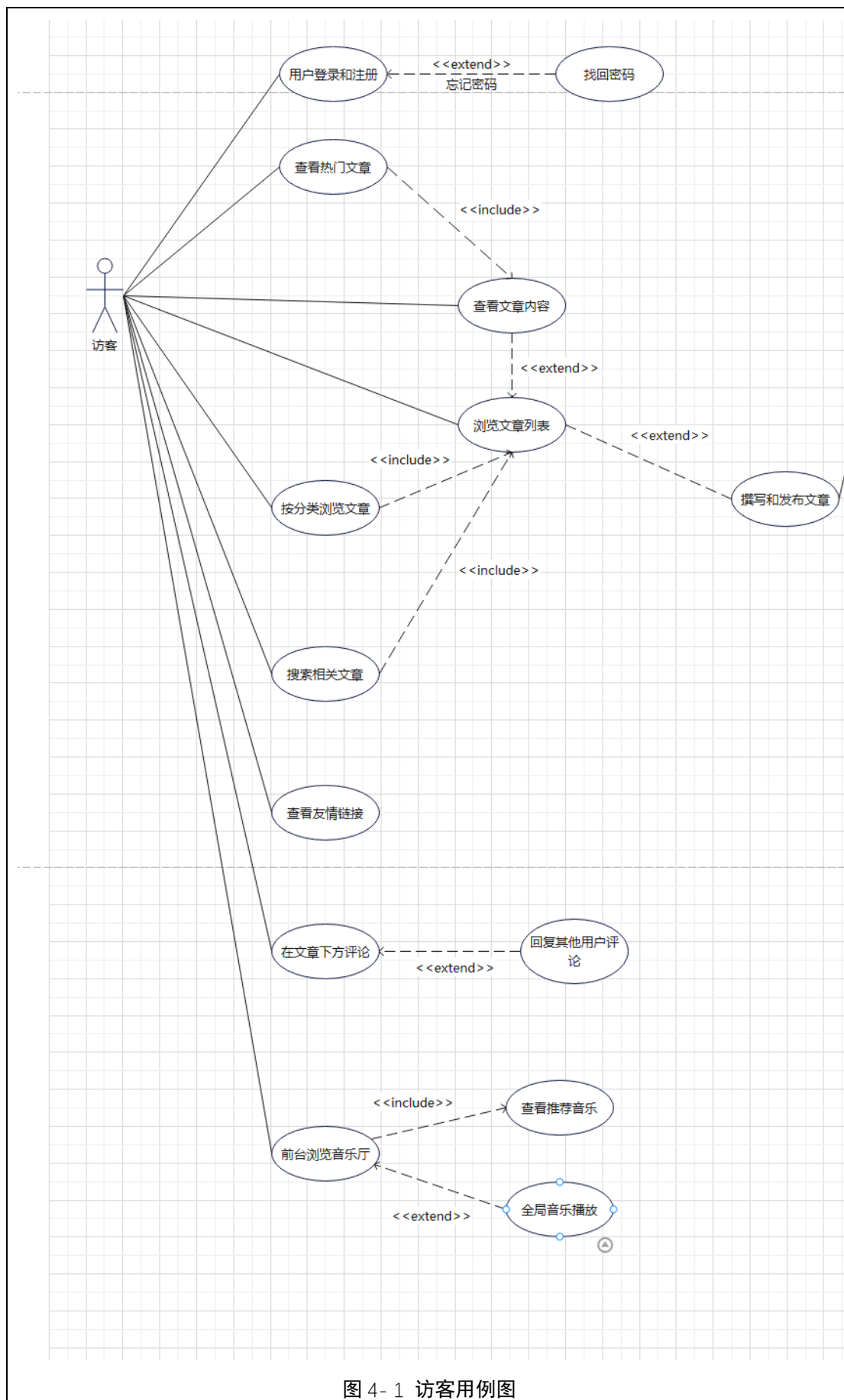


图 4-1 访客用例图

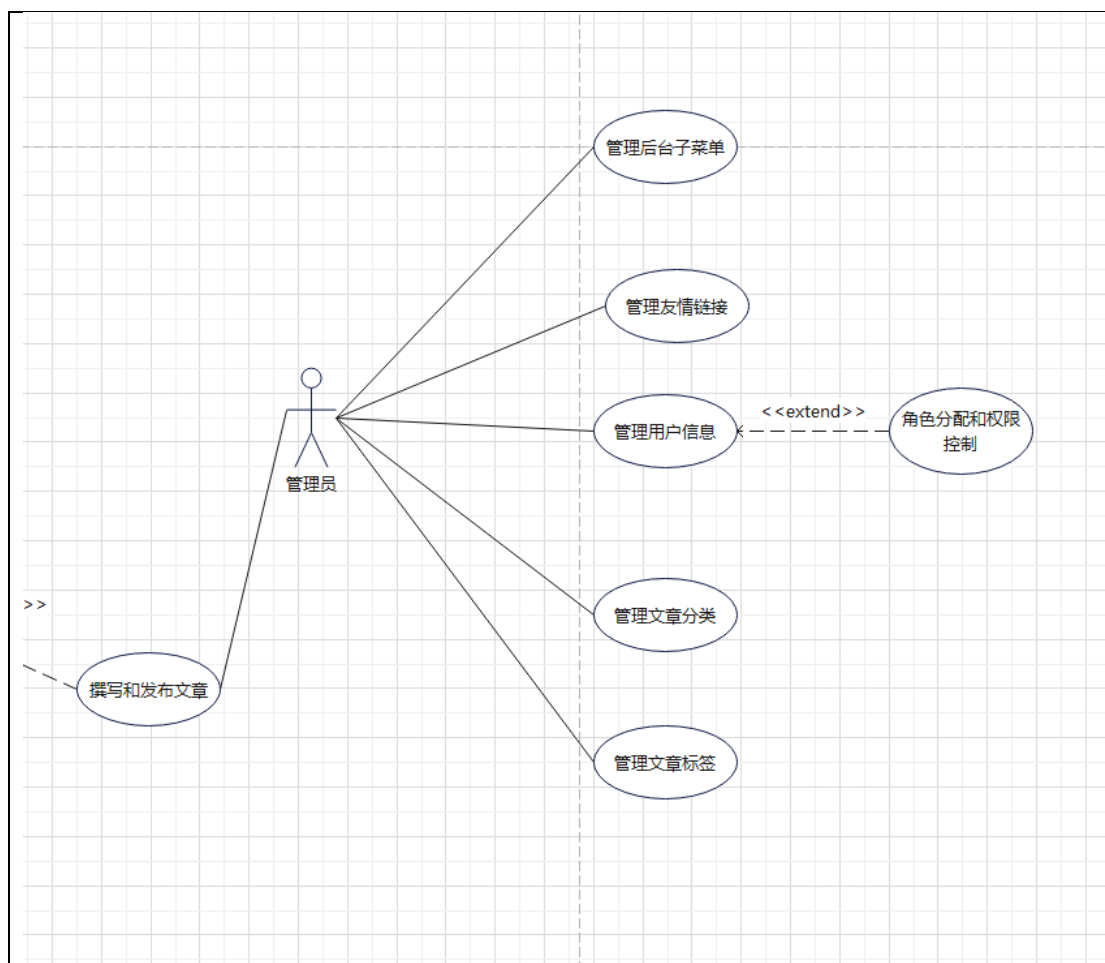


图 4- 2 管理员用例图

## （二） 功能需求

根据以上用户需求，分析总结出以下系统模块：

### 1. 用户模块：

- 提供用户登录和注册功能，包括用户身份验证、账号创建等。
- 提供用户管理功能，允许管理员进行增删改查操作。
- 提供用户权限管理功能，允许管理员分配角色和权限。
- 实现权限控制，根据用户的角色权限显示对应菜单。

### 2. 文章模块：

- 提供文章展示功能，包括文章列表和详情页的显示。
- 提供文章分类管理功能，允许管理员对分类进行增删改查操作。
- 提供文章标签管理功能，允许管理员对标签进行增删改查操作。
- 提供文章撰写与发布功能，允许管理员在后台编写和发布文章。
- 在前台提供热门文章展示、按类别获取文章、根据标签检索文章的功能。

### 3. 友链模块：

- 提供友链展示功能，在前台展示友链信息。
- 提供友链管理功能，允许管理员对友链进行增删改查操作。

4. 评论模块：

提供文章评论功能，让用户在前台对文章进行评论。

提供回复评论功能，允许用户对其他用户的评论进行回复。

允许用户发送 emoji 表情。

5. 音乐厅模块：

在前台提供推荐单曲、推荐歌单的展示。

提供歌曲检索功能，允许用户搜索特定的歌曲。

提供全局音乐播放功能。

6. 菜单管理模块：

提供后台子菜单管理功能，允许管理员对子菜单进行增删改查操作。

（三） 非功能需求

除了功能需求之外，还需要考虑以下非功能需求：

1.性能要求：网站应具备快速响应和加载速度，以确保良好的用户体验。

2.安全性要求：网站应采取必要的安全措施，保护用户的个人信息和数据安全，防止未授权访问和恶意攻击。

3.可扩展性要求：网站应具备可扩展性，能够容纳未来的增长和功能扩展需求，并保持高效运行。

4.可靠性和稳定性要求：网站应稳定运行，确保服务的可用性和持续性，最小化系统崩溃和故障的风险。

5.跨平台兼容性要求：网站应能够在不同的浏览器和设备上正常运行，以满足用户的多样化需求。

6.数据备份与恢复要求：网站应定期备份数据，并提供数据恢复机制，以防止数据丢失或损坏。

7.可管理性要求：后台管理系统应具备简单易用的界面，方便管理员进行用户、文章、评论、友链等信息的管理和维护。

## 五、总体设计

### （一） 系统整体架构（共 12 个模块）

1. 用户模块
  - 1) 登录注册子模块
  - 2) 用户管理子模块
  - 3) 权限管理子模块
2. 文章模块
  - 1) 文章展示子模块
  - 2) 文章分类管理子模块
  - 3) 文章标签管理子模块
  - 4) 文章撰写与发布子模块
3. 友链模块
  - 1) 友链展示子模块
  - 2) 友链管理子模块
4. 评论模块
5. 音乐厅模块
6. 菜单管理模块

### （二） 系统数据流图 待定

### （三） 数据库设计 与用户模块相关的数据库表设计如图 5-1 所示：



图 5- 1 与用户模块相关的 ER 图

与文章模块有关的数据库表设计如图 5-2 所示：

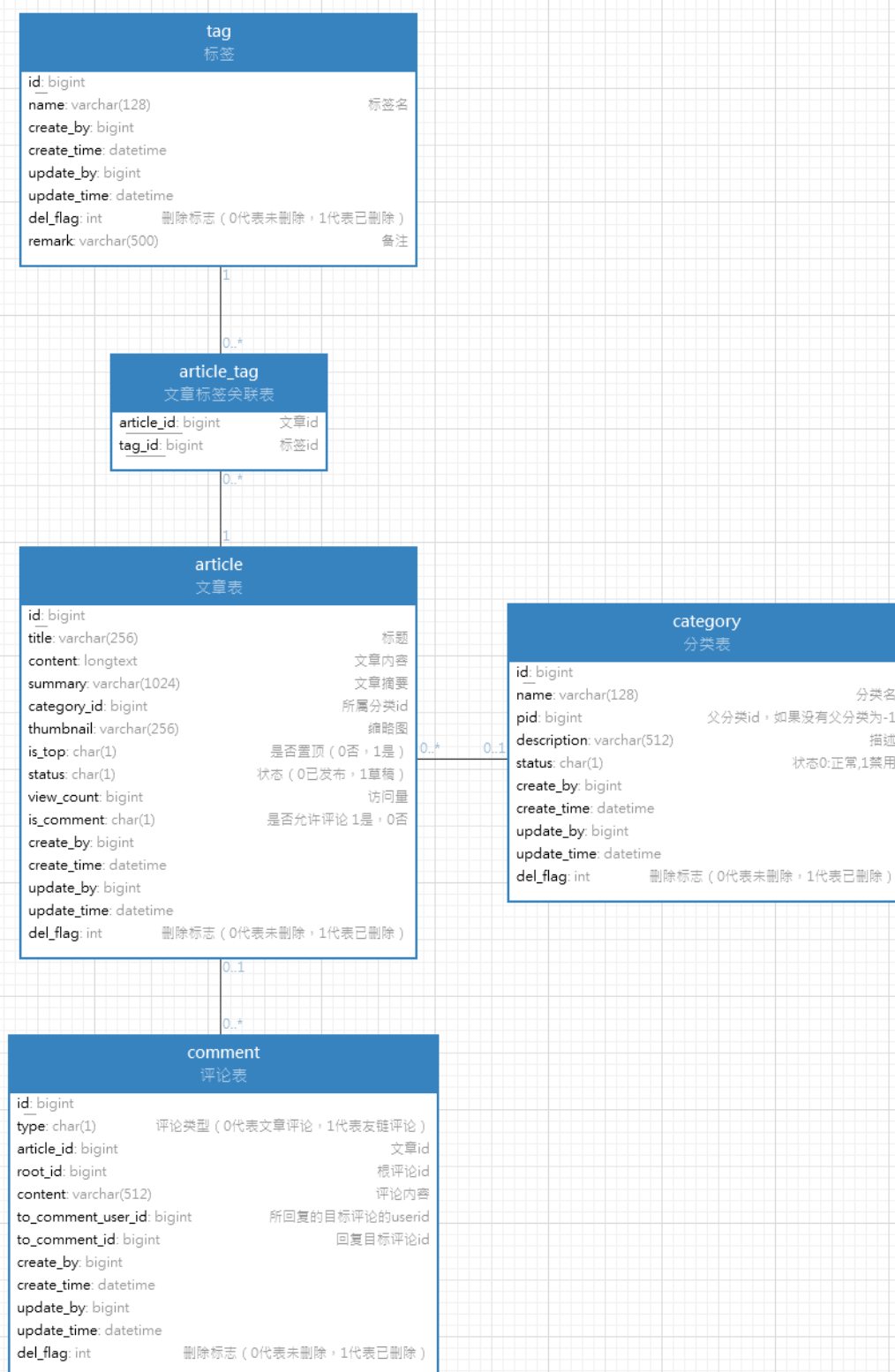


图 5- 2 与文章模块有关的 ER 图

与其他相对独立的子模块有关的数据库表设计如图 5-3 所示：



图 5- 3 其他独立子模块的 ER 图

## 六、详细设计

### (一) 模块设计

#### 1. 文章模块详细设计

##### A. 前台文章详情功能:

- 1) 要求: 在文章列表点击阅读全文时能够跳转到文章详情页面, 可以让用户阅读文章正文。
- 2) 请求方式: Get
- 3) 响应格式:

```
{
  "code": 200,
  "data": {
    "categoryId": "1",
    "categoryName": "java",
    "content": "内容",
    "createTime": "2022-01-23 23:20:11",
    "id": "1",
    "isComment": "0",
    "title": "SpringSecurity 从入门到精通",
    "viewCount": "114"
  },
  "msg": "操作成功"
}
```

##### 4) 详细流程:

- a. 首先, 通过调用 `getId(id)` 方法根据传入的文章 ID 查询对应的文章对象。
- b. 使用 `BeanCopyUtils` 工具类的 `copyBean` 方法将查询到的文章对象转换成 `ArticleDetailVo` 对象。其中, `ArticleDetailVo` 是一个表示文章详情的视图对象, 与 `Article` 对象有一定的差异或附加属性。
- c. 通过获取 `ArticleDetailVo` 对象的 `categoryId` 属性, 调用分类服务的 `getId(categoryId)` 方法查询对应分类的分类对象。
- d. 如果查询到的分类对象不为空, 则将分类对象的名称设置到 `ArticleDetailVo` 对象的 `categoryName` 属性中。
- e. 最后, 封装响应结果, 通过调用 `ResponseResult.okResult()` 方法返回一个表示成功的响应结果, 并将 `ArticleDetailVo` 对象作为结果数据返回。

##### B. 前台分页查询文章列表功能:

- 1) 需求: 在首页和分类页面都需要查询文章列表。
- 2) 首页: 查询所有的文章
- 3) 分类页面: 查询对应分类下的文章
- 4) 要求:
  - a. 只能查询正式发布的文章



- b. 置顶的文章要显示在最前面
- C. `hotArticleList()` 方法：该方法用于获取热门文章列表。
  - 1) 具体实现如下：
    - a. 创建一个 `LambdaQueryWrapper` 对象 `queryWrapper` 用于构建查询条件。
    - b. 添加查询条件，要求文章状态为正常
    - c. 设置按照浏览量进行降序排序。
    - d. 创建一个分页对象 `page`，设置每页显示 10 条记录。
    - e. 调用 `page()` 方法进行分页查询，将查询结果存储在 `page` 对象中。
    - f. 从 `page` 对象中获取查询到的文章列表。
    - g. 使用 `BeanCopyUtils.copyBeanList()` 方法将文章列表转换为 `HotArticleVo` 对象列表。
    - h. 封装查询结果，通过调用 `ResponseResult.okResult()` 方法返回一个表示成功的响应结果，并将 `HotArticleVo` 对象列表作为结果数据返回。
  - 2) 该方法的 UML 时序图如图 6-1 所示：

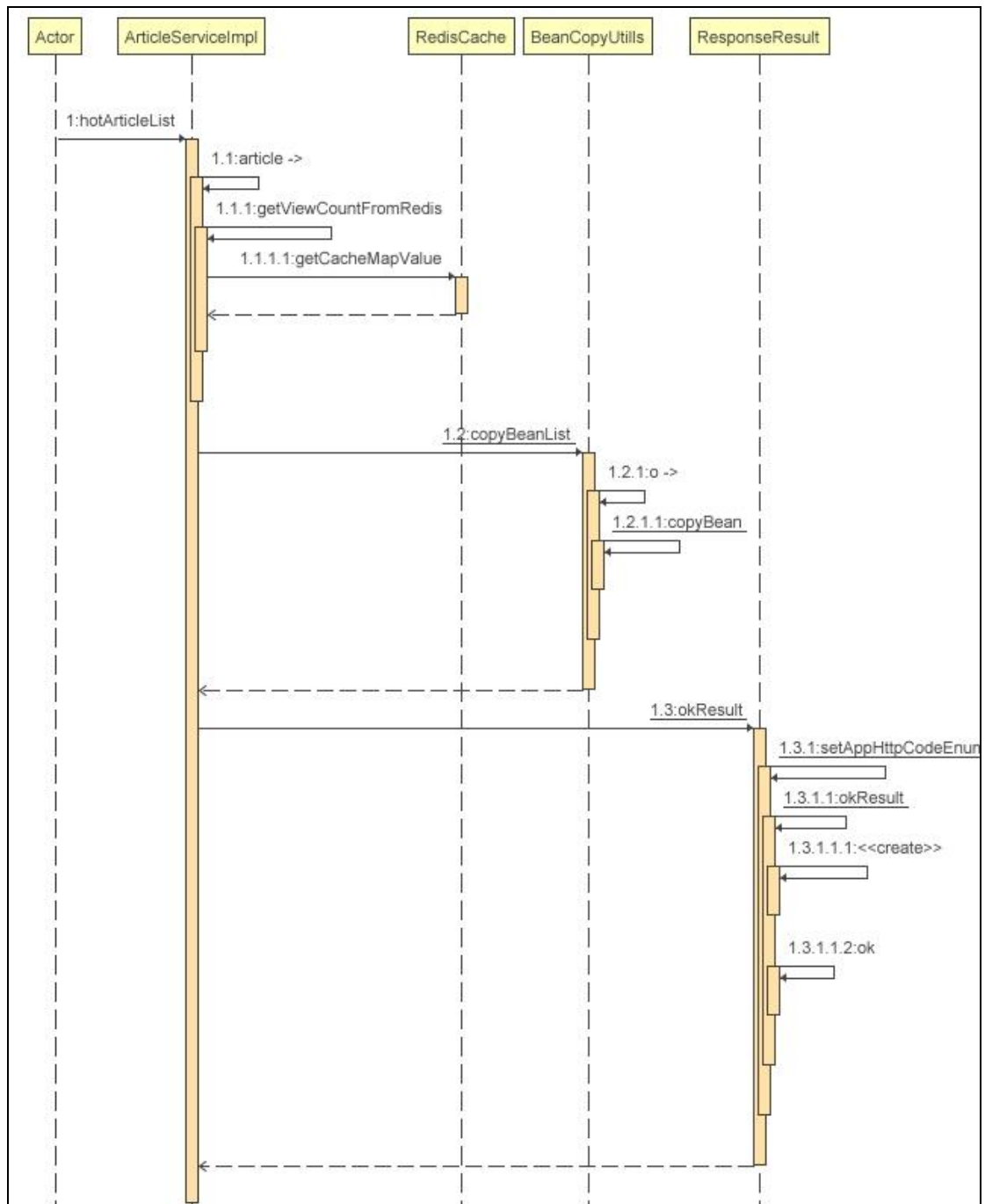


图 6- 1 hotArticleList() 的 UML 时序图

D. articleList() 方法：该方法用于根据分页和分类条件获取文章列表。

1) 具体实现如下：

- a. 创建一个 LambdaQueryWrapper 对象 lambdaQueryWrapper 用于构建查询条件。
- b. 添加查询条件，要求文章状态为正常（status = SystemConstants.ARTICLE\_STATUS\_NORMAL）。
- c. 如果传入的 categoryId 不为空且大于 0，则添加与传入的 categoryId 相同的查询条件。
- d. 设置按照 isTop 降序排序。
- e. 创建一个分页对象 page，设置页码和每页显示的记录数。

- f. 调用 `page()` 方法进行分页查询，将查询结果存储在 `page` 对象中。
  - g. 从 `page` 对象中获取查询到的文章列表。
  - h. 遍历文章列表，通过调用分类服务的 `getById(categoryId)` 方法查询到对应分类对象，并将分类名称设置到文章对象的 `categoryName` 属性中。
  - i. 使用 `BeanCopyUtils.copyBeanList()` 方法将文章列表转换为 `ArticleListVo` 对象列表。
  - j. 创建一个 `PageVo` 对象，将转换后的 `ArticleListVo` 对象列表和总记录数设置进去。
  - k. 封装查询结果，通过调用 `ResponseResult.okResult()` 方法返回一个表示成功的响应结果，并将 `PageVo` 对象作为结果数据返回。
- 2) 该方法的 UML 时序图如图 6-2 所示：

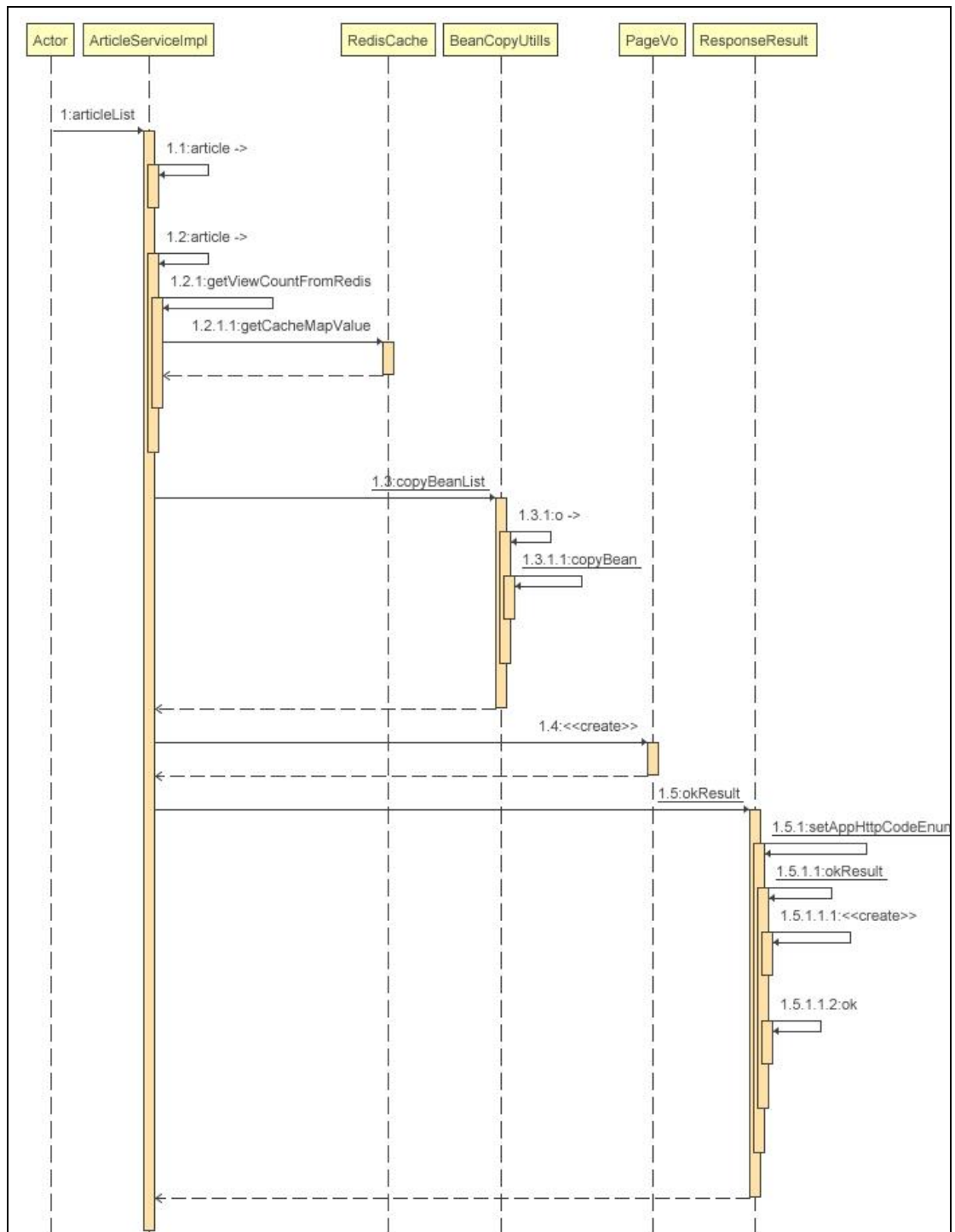


图 6- 2 articleList() 的 UML 时序图

## 七、编码与测试

### (一) 代码目录总览

1. 后端代码目录结构如图 7-1 所示：

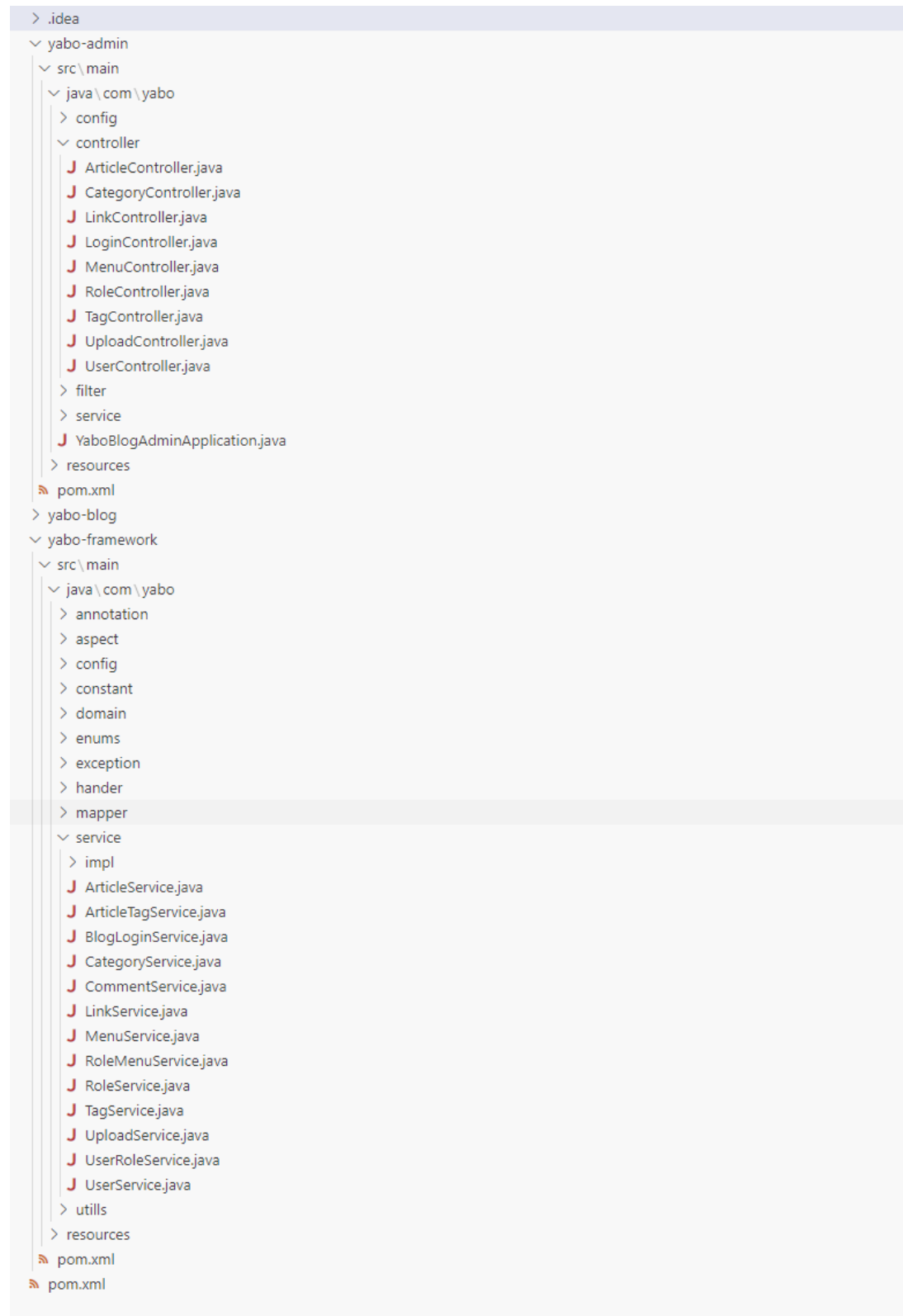


图 7-1 后端代码目录

## 2. 前端代码目录结构如图 7-2、7-3 所示：

```
> .idea
> config
> node_modules
> public
> scripts
< src
  > api
  > assets
  < components
    < blog
      ArticleDetail.vue
      ArticleList.vue
      Friends.vue
      Header.vue
      Message.vue
      RightList.vue
    < music
      APlayer.vue
      MusicController.vue
      PlayList.vue
      SingleSong.vue
    HelloWorld.vue
    ToggleTheme.vue
    TransferStation.vue
    WeatherWidget.vue
  > constants
  < hooks
    TS auth.ts
    TS errorCode.ts
    TS getRecommendedPlayList.ts
    TS getRecommendedSongs.ts
    TS index.ts
    JS plug.js
    TS request.ts
    TS server.ts
    JS theater.js
    TS useStorage.ts
  > router
  < stores
    TS index.ts
    TS main.ts
    TS message.ts
    TS music.ts
  > styles
  > types
  < views
    > foo
    404.vue
    article-detail.vue
    file.vue
    friend-link.vue
    home.vue
    login.vue
    music.vue
    user-info.vue
    App.vue
    TS main.ts
  .editorconfig
  .eslintignore
  .eslintrc.js
  .gitignore
  .prettierrc
  index.html
  package-lock.json
  package.json
  README.md
  tsconfig.json
  TS vite.config.ts
```

图 7-2 前台前端代码目录

- > .idea
- > build
- > dist
- > node\_modules
- > public
- ▼ src
  - ▼ api
    - > content
    - > system
    - JS login.js
    - JS menu.js
    - > assets
  - ▼ components
    - > Breadcrumb
    - > Hamburger
    - > ParentView
    - > SvgIcon
  - > directive
  - > layout
  - > plugins
  - > router
  - > store
  - > styles
  - > utils
  - ▼ views
    - > content
    - > dashboard
    - > login
    - > nested
    - ▼ system
      - ▼ menu
        - ▼ index.vue
      - ▼ role
        - ▼ index.vue
      - ▼ user
        - ▼ index.vue
  - ▼ App.vue
  - JS main.js
  - JS permission.js
  - JS settings.js
- > tests
- ≡ .editorconfig
- \$ .env.development
- \$ .env.production
- \$ .env.staging
- ⊙ .eslintignore
- ⊙ .eslintrc.js
- ◆ .gitignore
- ! .travis.yml
- 📄 babel.config.js
- JS jest.config.js
- { } jsconfig.json
- 📄 LICENSE
- ≡ npm
- { } package-lock.json
- { } package.json
- JS postcss.config.js
- 📖 README-zh.md
- 📖 README.md
- JS vue.config.js

图 7- 3 后台前端代码目录

(二)	局部代码展示
(三)	针对各模块的单元测试
(四)	集成测试



八、结论与展望
（一） 项目总结
（二） 存在问题
（三） 后续改进计划