1. 引言

随着影视行业的快速发展，用户对影视内容的交流需求日益增长。现有的影视论坛存在信息分类混乱、交互体验不足、内容管理不完善等问题。因此，本系统旨在构建一个基于SpringBoot+Vue的影视论坛，为用户提供高效、便捷的影视交流平台，同时为管理员提供完善的后台管理功能，提升平台的整体运营效率。

2. 功能需求分析

2.1 普通用户

用户注册与登录：

支持用户注册、登录、退出，采用加密算法保障用户信息安全。

内容展示：

提供影视分类展示，支持按类型、评分、热度等维度筛选；

推荐近期热门影视剧，提升用户观影体验；

展示用户发布的帖子，用户可进入详情页进行讨论。

帖子发布：

支持文本、图片、视频等多媒体内容发布；

可为帖子添加分类标签，方便搜索与推荐；

允许预定发布时间，增强内容管理灵活性。

影视详情页：

提供影视剧的详细信息，包括预告片、剧情简介、演员阵容、海报等；

展示其他用户评价、评分柱状图，并支持用户评分。

帖子详情页：

用户可评论、点赞、收藏帖子；

提供举报功能，以维护社区环境。

搜索与筛选：

根据关键字进行搜索，并支持按时间、评分、热度等排序。

影视资源上传：

允许用户上传影视资源，管理员审核后可上线。

个人中心：

展示用户信息，支持修改昵称、密码等；

允许管理收藏、发布的内容。

历史记录：

记录用户评分和浏览过的影视剧，支持筛选回顾。

2.2 管理员用户

用户管理：

查看和管理用户列表，支持封禁、解封等操作；

统计用户活跃度，如发帖数、评论数等。

影视数据管理：

添加、修改、删除影视资源；

支持上传海报、编辑影视信息。

内容审核：

审核用户上传的影视剧资源，确保内容合规；

处理用户举报的违规内容。

统计与分析：

统计平台热门影视、热门帖子；

生成数据可视化报表，支持趋势分析。

3. 性能需求

响应时间：确保系统在正常访问下响应迅速，提供流畅的用户体验。

并发处理能力：能够支持一定数量的用户同时在线访问，保障系统稳定运行。

数据存储：数据库需具备良好的查询效率，并通过优化索引提升访问性能。

缓存优化：可适当引入缓存机制，提升热门数据的访问速度，优化系统性能

4. 系统架构设计

系统采用前后端分离架构，前端使用Vue框架实现响应式界面，后端采用SpringBoot提供RESTful API服务，数据库选用MySQL存储数据，结合Redis进行缓存优化。前后端通过JSON进行数据交互，并使用消息队列优化高并发请求处理，确保系统的稳定性和可扩展性。

5. 安全性要求

用户认证：采用JWT进行身份认证，确保用户身份的合法性。

数据加密：对用户密码进行安全存储，并保障数据传输的安全性。

权限管理：根据用户角色（普通用户、管理员）设定不同的权限，防止越权操作。

攻击防护：采取必要的安全措施，减少SQL注入、跨站请求伪造等安全风险。

6. 结论

本系统通过细化需求分析，合理规划各功能模块和架构设计，力求提升影视论坛的稳定性、交互性和安全性，为用户提供良好的使用体验，为管理员提供便捷的管理工具。系统在数据存储优化、内容推荐等方面进行探索，尽可能满足影视论坛的基本需求，并具备一定的扩展性，以适应后续优化和改进的可能性。