**学 士 学 位 论 文**

**基于nodejs的2.5次元论坛的设计于实现**

作 者 姓 名： 胡瀚文

学科、专业 ： 软件工程

学 号 ： 13427020

指 导 教 师： 崔鑫

完 成 日 期： 2017年5月7日

大连大学

Dalian University

# 基于nodejs的2.5次元论坛的设计于实现

总计: 毕业论文 54 页

表 格 15 表

插 图 45 幅

指导教师： 崔鑫

评 阅 人：

完成日期： 2017年5月7日

# 摘 要

2.5次元论坛系统是一个基于互联网为动漫爱好者提供交流平台的系统。本系统的出现是为了解决因为网络版权日益受到重视而导致的动漫版权严重分散问题而开发的。通过本系统，动漫爱好者们可以获取当季动漫及其大陆版权拥有者的完整列表，同时可以在本系统获得当季动漫的一个热度参考，还可以和其他的用户分享自己对每个动漫的评价，在解决问题的同时为动漫爱好者们带来便利。

2.5次元论坛系统主要使用了Nodejs,javascript,html,css等语言开发，使用Express，jQuery，mongoose等框架和MongoDB数据库，主要基于JSON数据结构，使用前后台分离的B/S架构开发。

本文详细说明了2.5次元论坛系统开发的整个过程。首先描述了版权问题受到重视后为动漫爱好者们带来的问题以及2.5次元论坛系统出现的意义。接着按照整个系统的开发流程，从需求分析到系统设计再到系统实现和测试详细的介绍了整个系统的开发过程，以及本人从开发2.5次元论坛系统的过程中得到的一些认识和学到的一些技能。着重介绍了本系统的技术分析，需求分析，系统设计，实现和测试。

关键词：论坛；动漫；nodejs；

# Abstract

This article is mainly to solve the animation enthusiasts in this particular group because of network copyright issues and can not get their favorite animation information and the development of the 2.5 meta-forum system in detail.

2.5 yuan Forum system is based on the Internet for the animation enthusiasts to provide a platform for the exchange platform. The system was developed to address the growing problem of decentralization of animation copyright due to the increasing emphasis on online copyright. Through this system, animation enthusiasts can get a complete list of season animation and its mainland copyright owners, while the system can get a hot season for a hot reference, but also and other users to share their own for each animation Evaluation, in solving the problem at the same time for the animation enthusiasts to bring convenience.

2.5 times the forum system mainly uses Nodejs, javascript, html, css and other language development, the use of Express, jQuery, mongoose framework and MongoDB database, mainly based on JSON data structure, the use of front and back separation of B / S architecture development.

This paper elaborates the whole process of 2.5-level system development. First of all, the copyright issues are taken seriously after the problems brought by the animation enthusiasts and 2.5 yuan forum system significance. Then in accordance with the entire system development process, from the needs analysis to the system design and then to the system to achieve and test a detailed introduction to the entire system development process, as well as from the development of 2.5 meta-forum system in the process of some understanding and learned some skill. This paper introduces the technical analysis, requirement analysis, system design, implementation and testing of the system.

**Key Words：Forum；Anime；Nodejs；**

目 录

[绪论 3](#_Toc482713737)

[1.1 研究背景 3](#_Toc482713738)

[1.2 国内外发展现状 3](#_Toc482713739)

[1.3 论文结构 4](#_Toc482713740)

[2 相关技术分析 5](#_Toc482713741)

[2.1 系统开发环境介绍 5](#_Toc482713742)

[2.2 相关技术介绍 5](#_Toc482713743)

[2.2.1 Nodejs简介 5](#_Toc482713744)

[2.2.2 B/S结构 6](#_Toc482713745)

[2.2.3 MongoDB 简介 7](#_Toc482713746)

[2.2.4 Studio 3T 7](#_Toc482713747)

[3. 需求分析 8](#_Toc482713748)

[3.1需求定位 8](#_Toc482713749)

[3.2 功能确定 9](#_Toc482713750)

[3.3 系统数据流图 9](#_Toc482713751)

[3.3.1 系统顶层数据流图 10](#_Toc482713752)

[3.3.2 系统第一层数据流图 10](#_Toc482713753)

[3.3.3 第二层数据流图 11](#_Toc482713754)

[3.3.4 数据字典 12](#_Toc482713755)

[3.4用例图 15](#_Toc482713756)

[3.4.1 未登录用户用例图 15](#_Toc482713757)

[3.4.2 会员用户用例图 15](#_Toc482713758)

[3.4.3管理员用户用例图 16](#_Toc482713759)

[3.5 数据库设计 18](#_Toc482713760)

[3.5.1 建立概念模型 18](#_Toc482713761)

[3.5.2 系统ER图 18](#_Toc482713762)

[3.5.3 数据库表及字段 23](#_Toc482713763)

[4. 系统设计 28](#_Toc482713764)

[4.1系统架构设计 28](#_Toc482713765)

[4.2 系统功能模块设计 28](#_Toc482713766)

[4.2.1 系统功能图 28](#_Toc482713767)

[4.2.2 系统功能介绍 29](#_Toc482713768)

[5系统实现与测试 31](#_Toc482713769)

[5.1 系统各个功能模块介绍 31](#_Toc482713770)

[5.2 系统功能模块实现 32](#_Toc482713771)

[5.2.1登录模块实现 32](#_Toc482713772)

[5.2.2 注册模块实现 34](#_Toc482713773)

[5.2.3 发布帖子模块实现 35](#_Toc482713774)

[5.2.4 回复功能实现 36](#_Toc482713775)

[5.2.4 主题管理模块实现 37](#_Toc482713776)

[5.2.5 用户管理模块实现 38](#_Toc482713777)

[5.2.6 版权网站管理模块实现 39](#_Toc482713778)

[5.2.7 帖子管理模块实现 41](#_Toc482713779)

[5.2.8 查看帖子模块实现 42](#_Toc482713780)

[5.2.9 查看主题模块实现 43](#_Toc482713781)

[5.2.10 查看排行榜模块实现 44](#_Toc482713782)

[5.3.系统测试 45](#_Toc482713783)

[5.3.1 功能测试 45](#_Toc482713784)

[5.3.2 安全性测试 47](#_Toc482713785)

[5.3.3 可用性测试 48](#_Toc482713786)

[5.3.4 性能测试 48](#_Toc482713787)

[5.3.5 测试结果分析 48](#_Toc482713788)

[结论 49](#_Toc482713789)

[参考文献 50](#_Toc482713790)

[致 谢 51](#_Toc482713791)

[附录一 外文译文 52](#_Toc482713792)

[附录二 外文原文 53](#_Toc482713793)

# 绪论

## 1.1 研究背景

近七年以来，网络版权问题日益受到重视，这毋庸置疑是一件保护创作者智力成果的好事。但是，版权收到重视带来好处的同时也给ACG爱好者们带来了一定麻烦，因为版权一些独占性的问题，以前那种一个网站看遍当季所有动漫的日子如同时间一般随风消逝，不再回头。如果想要像以前那样看遍所有当季热番，那就需要购买几乎所有视频播放网站的vip权限，这个价格平均是每个网站20元/月，大概需要购买vip的网站有优酷，土豆，爱奇艺，PPTV，腾讯视频，乐视TV，搜狐视频，芒果TV等，而一季动漫的播放周期一般而言是四个月，这就意味着这将是一笔数额不小而且持续的支出，尤其是对大部分还是无收入的学生爱好者群体来说，更是一个令人头疼的问题。

版权受到重视是大势所趋，我们不能期待哪天突然所有动漫都不需要版权，所有动漫都可以免费观看了，这不现实。既然这样，那么如何适当的减少支出呢？我们能做的可能就只有根据自己喜欢的动漫，择站购买vip了。那么如何判断需要购买哪些网站的vip，或者说，怎么确定一部动漫是不是自己喜欢的呢？当然，几乎所有动漫在发布之前都会放出各种PV以吸引用户，还有各种宣传海报，广播剧等宣传手段。但是，熟悉这些的人都知道，这些东西都是不可靠的，PV可能造假，海报什么的也可能只是一个空头支票，最可靠的恐怕还是观看后的口碑了。而口碑这种东西开放的讨论中自然而然地产生的东西。

按照这样的思路，本人设计了2.5次元论坛系统。本系统一方面为用户提供了当季动漫的列表以及每部动漫的版权分布，同时为用户发布和讨论观看感想提供了一个平台。同时本系统也提供了根据主题热度生成排行榜的功能，提供了每部动漫口碑的一个参考。

## 1.2 国内外发展现状

本系统后台方面主要使用了Nodejs这门语言。Nodejs自2009年首次发布以来，因为其在高并发非阻塞以及简单的api结构方面的优势，被越来越多的应用于实时系统方面。而得益于自2015年以来这门语言的完全开源，其社区的活跃度也越来越高，也带动着这门语言本身也非常迅速的发展着。在github上，node目前已经有126个branch，1283名contributors,415个releases,其commit更是高达16705，并且每天都在增加。在可以预见的将来，这门语言也势必会更加完善，继续发挥它在高并发非阻塞实时系统领域的作用。

同时前后台分离已经越来越成为公认的合理的开发模式，Nodejs和js的ajax提供的简洁的api在前后端分离方面有着天然的优势，因此Nodejs也必然会越来越多的成为至少是api处理层面优先的语言选择。

因为动漫在国内目前而言大部分特指日本动漫，而日本动漫是按照季度发布的，这里仅以最新的2017年春季发布的53部动漫为例。这53部动漫中，“独播”（版权独占）的有27部，更有3部是全球独播，其中有七部需要付费观看。而根据目前国内知名动漫社团sosg的热度统计，热度和好评综合排行前10的动漫中有4部和付费独播的7部动漫重叠。问题更大的是这四部动漫的版权归属网站是不同的，这就意味这你想看在第一时间看到这些热门的动漫，你就需要同时购买这些版权网站的vip会员，对大部分还是学生的动漫爱好者而言，这并不是一笔小支出。而因为法律和社会观念的大趋势，版权问题只会越来越受到重视，就像不久之前的2017年2月23日，由中国版权保护中心主办的“原动力”动漫版权专题论坛暨《中国网络漫画出版发展报告》发布仪式在京举行，论坛围绕“尊重动漫版权，提升原创价值”展开，由主题发言、发布《中国网络漫画出版发展报告》、发起成立动漫版权联盟三部分内容组成，论坛最后，几十家动漫企业、出版单位和动漫创作者共同发起成立动漫版权联盟，签订《保护动漫版权倡议书》，如此看来，也许5年10年后所有的动漫都是独播，付费，信息的分布也会越来越分散。但是，历史证明，天下大势分久必合，我相信动漫版权领域也不会例外，因为网站要发展，就必须考虑用户的需求，而用户都信息更高的集合度的需求必然会催化动漫的版权向统一的方向发展，就像小说领域的阅文集团的产生就是因为当初读者喜欢的小说作家分布在不同的创作平台导致的信息分散而催生的集合体。因此，2.5次元论坛的核心业务——对动漫信息的讨论，整理，发布必然能解决一大批使用者的问题。所以，开发这样一个资讯讨论论坛是很有必要的。

## 1.3 论文结构

本文的具体内容包括：

第一章：主要介绍2.5次元论坛系统的研发背景和其意义。

第二章：对系统的开发环境和使用的相关技术进行了分析和说明

第三章：介绍了2.5次元论坛系统的需求分析，并介绍了2.5次元论坛的数据流图。

第四章：主要介绍了。25次元论坛系统的系统设计，系统实现和系统测试。

# 2 相关技术分析

## 2.1 系统开发环境介绍

软件体系结构方案：采用B/S模式。浏览器-服务器（Browser/Server）结构，简称B/S结构，与C/S结构不同，其客户端不需要安装专门的软件，只需要浏览器即可，浏览器通过Web服务器与数据库进行交互，可以方便的在不同平台下工作。可谓是论坛类系统的不二之选。

操作系统方案：windows 10. Microsoft操作系统大一统的实施平台，兼容性良好，减少系统移植带来的问题。

数据库：MongoDB。JSON数据格式的最优支持者，最适合2.5次元论坛系统的数据结构和数据库的选择。

开发工具：webstorm和Studio 3T。Studio 3T是目前对MongoDB支持最好的GUI工具，提供的各类广泛而精致的功能让开发完全集中于数据的处理而不用为冗长的命令行命令头疼。

Webstorm则是web开发的最佳选择，因为前后台都使用了javascript语言，webstorm对javascript的强大支持让开发变得简单很多。

开发语言：nodejs，javascript，html，css

开发技术：express，jQuery，mongoose，bootstrap等框架。

## 2.2 相关技术介绍

### 2.2.1 Nodejs简介

关于Nodejs,“Node.js是一个开放源代码、跨平台的JavaScript语言运行环境。它的大部分基本模块都是使用JavaScript语言编写的”。[15]本系统前台的主要语言也是Javascript，这一点也是我选择使用Nodejs的一个很重要的原因。

Node.js目前采用的是Google开发的V8运行代码，在性能提升方面使用了事件驱动、非阻塞和异步输入输出模型等技术，能够对应用程序的传输量和规模有比较好的优化。这些技术通常用于数据密集的事实应用程序，这一点对2.5次元论坛系统而言非常适合。

Node.js同时包含一系列内置模块，使得程序可以脱离Apache HTTP Server或IIS，作为独立的服务器运行。

Node.js还允许通过JavaScript和一系列模块来编写服务器端应用和网络相关的应用。核心模块包括文件系统I/O、网络、二进制数据流、加密算法、数据流等等。其API形式简单，大大降低了编程的复杂度。这些都是非常适合2.5次元论坛系统的优势功能

同时Nodejs还有一个十分成熟的框架Express.js，这是一个”还在不断被维护的可靠的框架，可以大大加快开发速度。”[19]

Node.js的程序可以在Microsoft Windows、Linux、Unix、Mac OS X等服务器上运行。完全不用担心未来可能的移植问题。

Node.js主要用于编写像Web服务器一样的网络应用，这和PHP和Python是类似的。但是Node.js与其他语言最大的不同之处在于，PHP等语言是阻塞的，而Node.js是非阻塞的。这一点对于2.5次元论坛系统这样很多时候需要考虑并发性的系统而言，是一个十分重要的性能

Node.js是事件驱动的。”开发者可以在不使用线程的情况下开发出一个能够承载高并发的服务器。其他服务器端语言难以开发高并发应用，而且即使开发出来，性能也不尽人意。”[6]Node.js正是在这个前提下被创造出来。Node.js把JavaScript的易学易用和Unix网络编程的强大结合到了一起。同时Nodejs还有一个十分强大的包管理器npm，这是目前全球规模最大的包管理系统，里面有大量优秀的组件可供使用。正是由于Nodejs这些优秀的同时和2.5次元论坛系统匹配度极高的功能和便利，我最后选择了Nodejs作为本系统服务器的开发语言。

### 2.2.2 B/S结构

B/S结构，中文名是浏览器-服务器结构，是Browser/Server结构的简称。与C/S结构不同，“此种结构的客户端不需要安装专门的软件，只需要通过浏览器就可以使用系统的各种功能，浏览器通过Web服务器与数据库交互，在不同平台下工作十分的便利。”[11]此种架构的服务器端可采用高性能计算机，并安装各类大型数据库，简化了客户端的工作。它是随着Internet技术兴起而产生的，对C/S技术的改进。

2.5次元论坛系统是一个面向全网络动漫爱好者的交流平台系统，这就要求此系统必须通过互联网进行操作，随时随地的浏览、发布资讯。所以本系统选择基于B/S架构是必然之选。

### 2.2.3 MongoDB 简介

MongoDB是一种可以广泛的应用于各种规模的企业、各个行业以及各类应用程序的开源数据库。它是一种非常适用于敏捷开发的数据库，其数据模式可以随着系统的更新灵活地修改。与此同时，它也为开发人员 提供了传统数据库的功能：二级索引，完整的查询系统以及严格一致性等等。

MongoDB是专为可扩展性，高性能和高可用性而设计的数据库。它可以从单服务器部署扩展到大型、复杂的多数据中心架构。利用内存计算的优势，MongoDB能够提供高性能的数据读写操作。

由于本系统的核心数据结构采用的是JSON数据结构，而mongoDB数据库毫无疑问是当前对JSON数据结构支持的最完美的数据库，因此MongoDB是2.5次元论坛系统数据库的不二之选。同时Nodejs有一个完美支持MongoDB的开发框架mongoose，这一点也是我选择MongoDB的一个很重要的原因。

### 2.2.4 Studio 3T

Studio 3T是一款专门为MongoDB设计的可视化工具，也是目前市面上最好的MongoDB GUI和IDE，包含使用MongoDB开发所需要的一切功能，同时它的最新版本已经达到5.0，是一款非常成熟可以信赖的工具。Studio 3T优于市面上其他的MongoDB可视化工具的地方在于，它不仅仅只是一个可视化工具，它还是MongoDB的一个十分优秀的集成开发环境，对用户连接速度，数据库监控等方面做了相当多的优化。使用它”可以避免在开发时免于编写和调试长命令行并陷入复杂的语法细节的困难”[7] 这在开发2.5次元论坛系统的时候非常有帮助。

# 3. 需求分析

## 3.1需求定位

系统的需求就是要解决动漫爱好者们遇到的问题。

首先，对于动漫资源过于分散的问题。因为版权的相关规定，如果没有购买一部动漫的播放版权，那么一个网站就不能播放这部动漫，但是这些规定只是针对TV版视频的播放，并不针对论坛或者新闻这些文字性或图片性的分析或交流，那么就完全可以制作一个列表，列出所有当季动漫的简要信息，做一个资源的集中。

其次，对于难以确定自己喜欢的作品的问题。喜欢，本身是一个很主观的问题，每个人的口味很难完全一致，很多时候都是看了以后才知道到底好不好看，或者说自己喜不喜欢。难道这就意味着必须要把所有动漫都看一遍吗？其实，还有一种办法可以更高效的让一个人确定一部作品是不是自己的所爱，那就是查看看过的人的评价。虽然这些评价也带有评价人的主观色彩，但是这种方法至少比每一部作品都看过去要高效得多。同时，大多数时候，大多数人喜欢的东西其实自己也会喜欢，看大多数也都爱看的作品也是也个发现自己喜欢的作品的好方法。那么系统就完全可以为用户提供一个开放自由的讨论平台供所有的动漫爱好者们发表自己对自己看过的作品的评价，同时为其他没看过这些作品的爱好者们提供一个参考，同时根据用户对作品的评价，分析后生成一个排行榜，列出一些最受欢迎的作品，帮助爱好者们更高效的定位自己喜欢的作品。同时为了完成这样的分析，系统也需要对用户的点击数或者评论数等信息做一个收集分析的功能。

然后，爱好者们知道了哪些是自己可能喜欢的作品之后，要去哪里观看呢？这也是一个不小的问题，在茫茫多的正规网站和盗版网站中找到自己喜欢作品的可用播放地址其实并不是一件很容易的事，相反，这是一个比较耗时，同时令人焦躁的是，那么这种事情就让本系统来做吧，为所有动漫列出它们的版权所属，虽然对我来说是一个稍微耗时的事情，但是能方便所有爱好者，这也是一件好事。

最后，为了防止一些可能出现的错误和一些大家都不愿意出现的问题，对所有信息的管理和对讨论内容做一定的限制也是必须的，那么也就需要管理员这一群体对整个论坛的各类信息有管理的权限。

## 3.2 功能确定

为了满足上述的种种需求，我最终为2.5次元论坛系统确定的系统功能主要如下：

首先是为了做到动漫资源的集中显示，我将每一种动漫作品作为一个主题，按照年份进行分类，并在主页为用户提供了主题的搜索和查看功能。同时为了防止人为的错误，管理员用户还可以对主题的信息做添加，修改和删除的操作。最终设计了主题搜索和主题查看，添加主题，修改主题，删除主题这五个功能。

然后，为了给动漫爱好者们一个交流的平台，我在每个主题的详情页面为所有登录的用户提供了发帖的功能，也在帖子详情页面（通过在主题详情页点击帖子标题可以进入帖子详情页）为所有登录用户提供了回复的功能，所有登录用户都可以在这里畅所欲言，同时可以从其他人评论中获得自己是否喜欢这部作品的一些参考。同时所有用户对主题详情页的浏览以及所有的发帖回复行为都会被系统记录下来，用于分析作品热度并最终生成作品热度的榜单，进一步为用户提供寻找自己喜欢作品的参考。同时为了防止不正当言论的散播，也为管理员用户提供了删除帖子的功能。最终设计了查看帖子，发布帖子，回复帖子，搜索帖子，点击记录，发帖记录，回复记录，生成排行榜，查看排行榜和删除帖子这八个功能。

然后为了让用户更方便的找到自己喜欢的作品的播放地址，我在所有主题的详情页面显示了当前主题的版权所属信息，同时为了方便实时更新，为管理员用户提供了版权网站信息的增加，搜索，修改，删除，以及在主题信息下修改主题对应版权所属的功能。最终设计了增加版权网站，搜索版权网站，修改版权网站，删除版权网站，修改主题版权所属等六个功能。

最后为了方便用户的使用以及为了防止不合法言论的散播，我还设计了登录，注册，登出功能方便用户使用，同时为管理员用户提供了修改用户权限以及删除用户的功能。同时为了应对HTTP协议无状态的问题同时方便用户使用，我还为每个页面做了登录状态验证的功能，如此用户就避免了在不同页面重复登录的繁琐。最终设计了登录，注册，登出，修改用户权限，删除用户，登录状态验证这六个功能。

上述所有功能都将在第四章系统实现中进行具体的介绍。

## 3.3 系统数据流图

数据流图DFD是描述系统中数据流程的一种图形工具，它标志了一个系统的逻辑输入和逻辑输出，以及把逻辑输入转换逻辑输出所需的加工处理。

值得注意的是，数据流图不是传统的流程图或框图，数据流也不是控制流。数据流图是从数据的角度来描述一个系统，而框图是从对数据进行加工的工作人员的角度来描述系统。下面是2.5次元论坛系统的数据流图。

### 3.3.1 系统顶层数据流图

系统的顶层数据流图如图3.1所示:

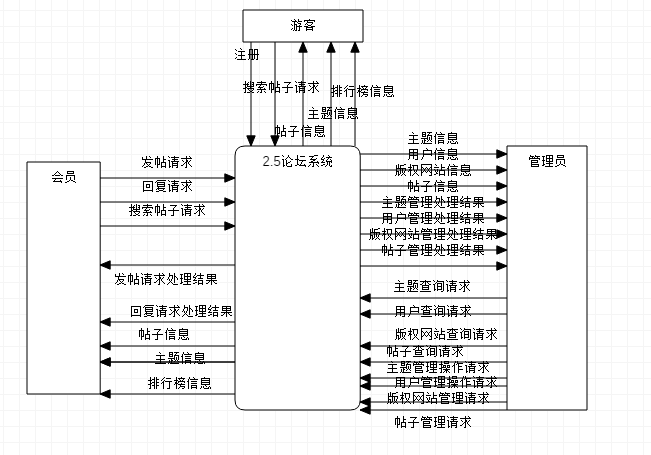


图3. 1系统顶层数据流图

### 3.3.2 系统第一层数据流图

本系统的第一层数据流图如图3.2所示：

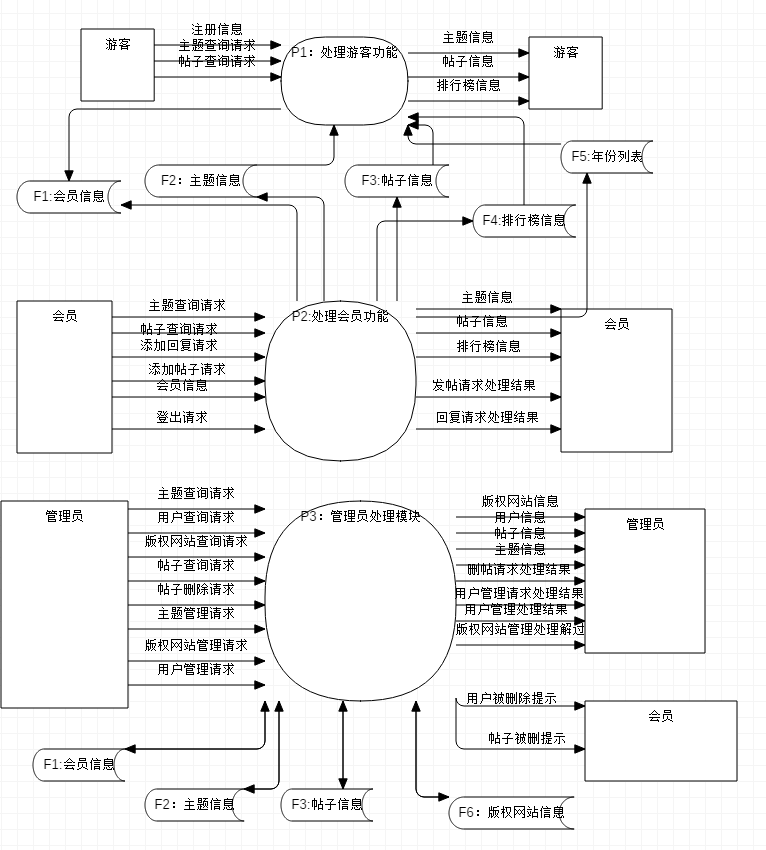


图3.13系统第一层数据流图

图3. 2系统第一层数据流图

### 3.3.3 第二层数据流图

本系统第二层数据流图，如图3.3所示。

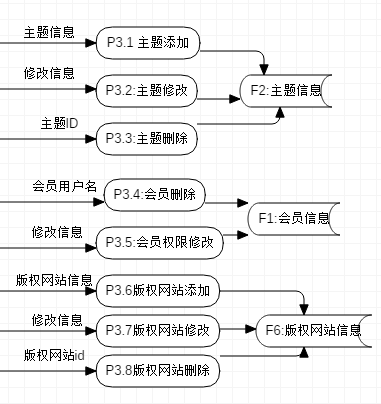
管理员的主题管理请求，版权网站管理请求，用户管理请求可以进一步细化。主题管理请求可以细分为主题添加请求，主题修改请求和主题删除请求。版权网站管理请求可以进一步细分为版权网站添加请求，版权网站修改请求和版权网站删除请求。用户管理请求可进一步细分为用户删除请求和用户权限修改请求。

图3. 3系统第二层数据流图

### 3.3.4 数据字典

本系统的数据字典解释如下：

表 3.1 外部实体定义

|  |  |  |
| --- | --- | --- |
| 名称 | 输入系统数据流 | 从系统获取数据流 |
| 游客 | 会员注册，主题查询，帖子查询 | 主题信息，帖子信息，排行榜信息 |
| 会员 | 登录信息，主题查询信息，帖子查询信息，发帖信息，回复信息，登出请求 | 主题信息，帖子信息，排行榜信息，发帖处理信息，回复处理信息 |
| 管理员 | 主题查询信息，帖子查询信息，用户查询信息，版权网站查询信息，主题添加信息，主题修改信息，主题删除信息，用户删除信息，用户权限修改信息，帖子删除信息，版权网站修改信息，版权网站删除信息 | 主题信息，帖子信息，用户信息，版权网站信息，主题管理处理信息，用户管理处理信息，帖子删除处理信息，版权网站处理信息 |

表 3.2 处理过程定义

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 名称 | 输入数据流 | 输出数据流 |
| P1 | 处理游客逻辑 | 会员注册、主题搜索、帖子搜索 | 主题信息，帖子信息，排行榜信息 |
| P2 | 处理会员逻辑 | 登录信息、主题查询、发布帖子、发布回复、帖子搜索 | 用户信息，主题信息，帖子信息，排行榜信息，发帖操作处理结果，回复操作处理结果 |
| P3 | 处理管理员逻辑 | 主题查询信息，帖子查询信息，用户查询信息，版权网站查询信息，主题添加信息，主题修改信息，主题删除信息，用户删除信息，用户权限修改信息，帖子删除信息，版权网站修改信息，版权网站删除信息 | 主题信息，帖子信息，用户信息，版权网站信息，主题管理处理结果，用户管理处理结果，帖子删除处理结果，版权网站处理结果 |

表 3.3 数据流定义

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 名称 | 数据流描述 | 数据来源 |
| D1 | 会员注册 | 注册的数据 | 游客 |
| D2 | 登录信息 | 会员的登录数据 | 会员 |
| D3 | 主题查询 | 搜索主题所需要的数据 | 游客或会员 |
| D4 | 帖子查询 | 搜索帖子所需要的数据 | 游客或会员 |
| D5 | 帖子信息 | 帖子的数据 | 帖子信息表 |
| D6 | 主题信息 | 主题的数据 | 主题信息表 |
| D7 | 排行榜信息 | 排行榜的数据 | 排行榜表 |
| D8 | 添加回复 | 添加回复需要的数据 | 会员 |
| D9 | 发布帖子 | 发布帖子需要数据 | 会员 |
| D10 | 版权网站查询 | 版权网站网站名信息 | 管理员 |
| D11 | 主题添加 | 主题初始化的数据和操作信息 | 管理员 |
| D12 | 主题修改 | 主题的修改数据和操作信息 | 管理员 |
| D13 | 主题删除 | 主题的Id信息和操作信息 | 管理员 |
| D14 | 会员删除 | 会员用户名和操作信息 | 管理员 |
| D15 | 会员权限修改 | 会员id，修改信息和和操作信息 | 管理员 |
| D16 | 版权网站添加 | 版权网站初始化数据和操作信息 | 管理员 |
| D17 | 版权网站修改 | 版权网站id，版权网站修改数据和操作信息 | 管理员 |
| D18 | 版权网站删除 | 版权网站id和操作信息 | 管理员 |

表 3.4 数据存储定义

|  |  |  |
| --- | --- | --- |
| 编号 | 名称 | 描述 |
| F1 | 会员信息 | 保存会员的基本资料 |
| F2 | 主题信息 | 保存主题的基本资料 |
| F3 | 帖子信息 | 保存帖子的基本资料 |
| F4 | 排行榜信息 | 保存排行榜的订单资料 |
| F5 | 年份列表 | 年份信息和年份所属的主题列表 |
| F6 | 版权网站信息 | 保存版权网站的基本资料 |
| F7 | 管理员信息 | 保存管理员的各类资料 |

## 3.4用例图

### 3.4.1 未登录用户用例图

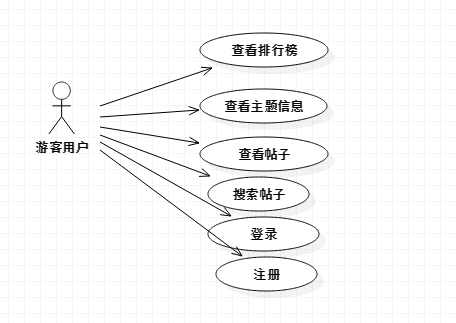
未登录用户的用例图如图3.4所示：

图3. 4游客用户用例图

游客用户拥有的功能有：

1. 查看排行榜
2. 查看主题信息
3. 查看帖子
4. 搜索帖子
5. 登录
6. 注册

### 3.4.2 会员用户用例图

会员用户用例图如图3.5所示：

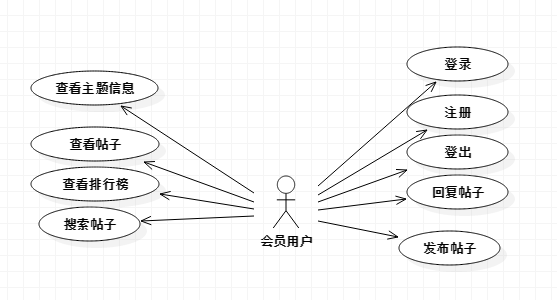


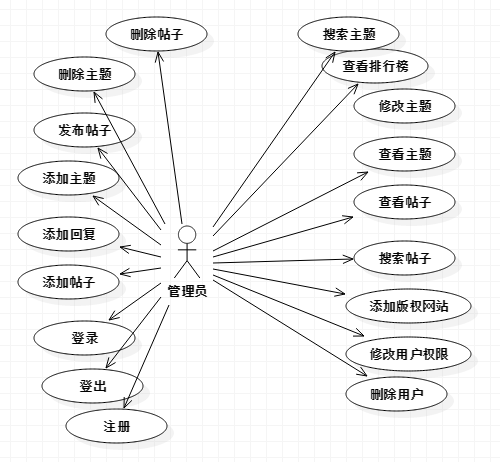
图3.5 会员用户用例图

会员用户可以使用的功能有：

1. 查看主题信息
2. 查看帖子
3. 查看排行榜
4. 搜索帖子
5. 发布帖子
6. 回复帖子
7. 登录
8. 注册
9. 登出

### 3.4.3管理员用户用例图

管理员用户的用例图如图3.6所示：

图3. 6管理员用户用例图

管理员用户可以使用的功能有：

1. 删除帖子
2. 修改用户权限
3. 添加主题
4. 修改主题
5. 删除主题
6. 添加版权网站
7. 修改版权网站信息
8. 删除版权网站信息
9. 搜索帖子
10. 添加回复
11. 查看帖子
12. 添加帖子
13. 查看主题
14. 查看排行榜
15. 登录
16. 注册
17. 登出

## 3.5 数据库设计

### 3.5.1 建立概念模型

确定建立数据库的目的以及完成数据收集后，就进入数据库设计过程的第二阶段——建立概念模型。概念模型，也称信息模型，它是按用户的观点来对数据和信息建模，主要用于数据库设计。这一阶段是整个数据库设计的关键。设计时，一般先根据应用的需求，画出能反映每个应用需求的E-R图，其中包括确定实体、属性和联系的类型。然后优化初始的E-R图，消除冗余和可能存在的矛盾。概念模型是队用户需求的客观反映，并不涉及具体的计算机软、硬件环境。因此，在这一阶段中我们必须将注意力集中在怎样表达出用户对信息的需求，而不考虑具体实现问题。

### 3.5.2 系统ER图

因为可以同时有多个管理员对多个主题，帖子，用户，回复，版权网站，所以管理员和主题，帖子，回复，版权网站，用户都是多对多的关系；一个用户可以发布多个帖子和回复，但是帖子和回复只能有一个发布人，所以用户与帖子和回复都是一对多的关系；因为一个主题只能从属于一个年份，而一个年份可以包含多个主题，所以主题和年份是多对一的关系；因为一个主题可以包含多个版权网站，一个版权网站也可以对应多个主题，所以主题和版权网站之间是多对多的关系；因为一个主题在一个排行榜中只能出现一次，而一个排行榜可以包含多个主题，所以主题和排行榜是一对多的关系；管理员和用户只能使用一个等级权限下的功能，而一个权限可以对应多个管理员和用户，所以权限与管理员和用户都是一对多的关系。

具体系统总体ER图如下图3.7所示：

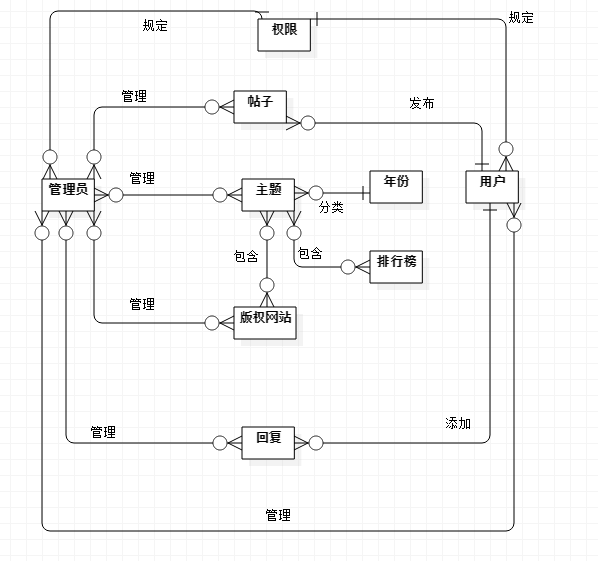


图3. 7系统ER图

以下是系统各实体及其对应的属性图：

（1）论坛用户实体和属性如图3.8所示：

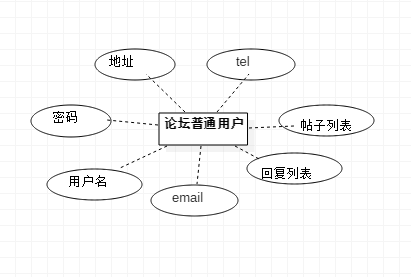


图3. 8论坛普通用户用例图

（2）论坛管理员用户实体和属性如图3.9所示：

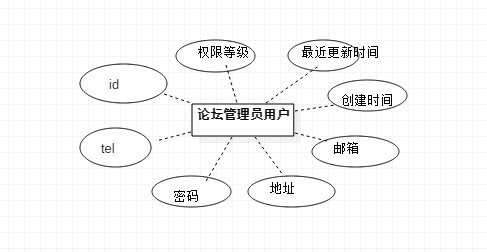


图3. 9管理员用户用力图

（3）主题实体和属性如图3.10所示：

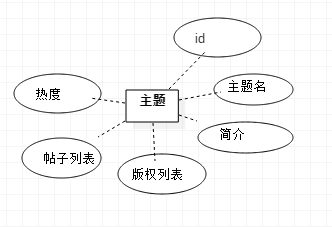


图3. 10主题实体与属性图

（4）帖子实体和属性如图3.11所示

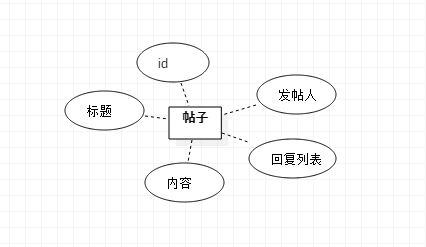


图3. 11帖子实体与属性图

（5）回复实体和属性如图3.12所示：

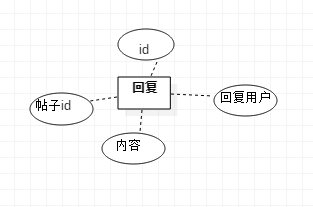


图3. 12回复实体与属性图

（6）排行榜实体和属性如图3.13所示：

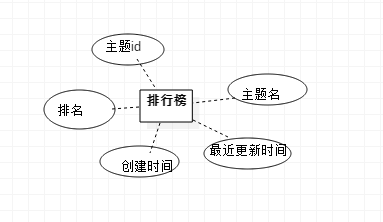


图3. 13排行榜实体与属性图

（7）版权网站实体和属性如图3.14所示：

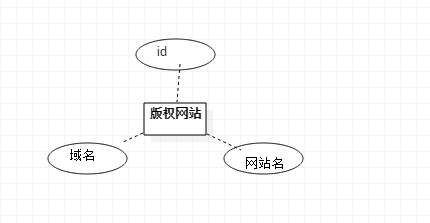


图3. 14版权网站实体与属性图

### 3.5.3 数据库表及字段

（1）用户信息表

表3.5用户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 |
| user | String | 16 | 是 |
| 密码 | String | 20 |  |
| 回复列表 | Array |  |  |
| 帖子列表 | Array |  |  |
| 电话 | String | 11 |  |
| 地址 | String | 50 |  |
| 头像 | String | 50 |  |
| 邮箱 | String | 50 |  |
| 创建时间 | String | 30 |  |
| 最近更新时间 | String | 30 |  |

（2）主题信息表

表3.6主题信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 |
| id | String | 16 | 是 |
| 主题名 | String | 50 |  |
| 图片地址 | String | 50 |  |
| 简介 | String | 500 |  |
| 版权列表 | Array |  |  |
| 点击数 | String | 50 |  |
| 评论数 | String | 50 |  |
| 帖子列表 | Array |  |  |
| 创建时间 | String | 30 |  |
| 最近更新时间 | String | 30 |  |

（3）主题id表

表3.7主题id表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 |
| 年份 | String | 4 | 是 |
| 主题列表 | Array |  |  |
| 创建时间 | String | 30 |  |
| 最近更新时间 | String | 30 |  |

（4）版权网站信息表

表3.8版权网站信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 |
| Id | String | 16 | 是 |
| 域名 | String | 20 |  |
| 网站名 | String | 20 |  |
| 创建时间 | String | 30 |  |
| 最近更新时间 | String | 30 |  |

（5）回复信息表

表3.9回复信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 |
| Id | String | 16 | 是 |
| 回复用户名 | String | 20 |  |
| 内容 | String | 500 |  |
| 创建时间 | String | 30 |  |
| 最近更新时间 | String | 30 |  |

（6）帖子信息表

表3.10帖子信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 |
| Id | String | 16 | 是 |
| 发帖用户名 | String | 20 |  |
| 标题 | String | 50 |  |
| 内容 | String | 500 |  |
| 回复列表 | Array |  |  |
| 创建时间 | String | 30 |  |
| 最近更新时间 | String | 30 |  |

（7）操作信息表

表3.11操作信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 |
| Id | String | 16 | 是 |
| 描述 | String | 20 |  |
| 回复列表 | Array |  |  |
| 创建时间 | String | 30 |  |
| 最近更新时间 | String | 30 |  |

（8）权限信息表

表3.12权限信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 |
| 等级 | String | 2 | 是 |
| 描述 | String | 20 |  |
| 可用操作列表 | Array |  |  |
| 创建时间 | String | 30 |  |
| 最近更新时间 | String | 30 |  |

（9）评论排行榜信息表

表3.13评论排行榜信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 |
| 排名 | String | 10 | 是 |
| 主题id | String | 20 |  |
| 主题名 | String | 50 |  |
| 评论数 | String | 20 |  |
| 创建时间 | String | 30 |  |
| 最近更新时间 | String | 30 |  |

（10）点击排行榜信息表

表3.14点击排行榜信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 |
| 排名 | String | 10 | 是 |
| 主题id | String | 20 |  |
| 主题名 | String | 50 |  |
| 点击数 | String | 20 |  |
| 创建时间 | String | 30 |  |
| 最近更新时间 | String | 30 |  |

（11）管理员信息表

表3.15管理员信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 是否主键 |
| 管理员名 | String | 20 | 是 |
| 密码 | String | 20 |  |
| 权限等级 | String | 2 |  |
| 电话 | String | 11 |  |
| 地址 | String | 50 |  |
| 邮箱 | String | 50 |  |
| 创建时间 | String | 30 |  |
| 最近更新时间 | String | 30 |  |

# 4. 系统设计

按照2.5次元论坛系统的功能需求，对系统的非阻塞性、高并发性都有较高的要求,根据前面的需求可以确定系统的设计应选择B/S结构. 因为开发语言选择的是nodejs，而目前并没有一个非常成熟且流行的nodejs开发基于nodejs的架构方式，所以采用了比较简单的前后台分离架构，同时使用了目前nodejs开发最成熟的框架：Express。

## 4.1系统架构设计

系统架构图如图4.1所示，本系统的整体架构方面，浏览器客户端通过http请求发送Ajax，使用了jQuery框架封装的Ajax接口，获取数据后在客户端进行渲染，数据格式使用json数据结构。服务器和Mongo数据库之间使用mongoose框架实现，通过框架封装的接口对数据库的数据进行操作。

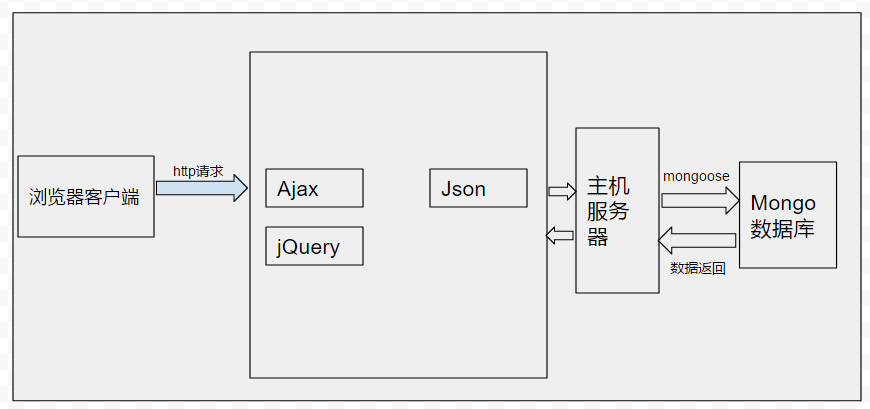


图4.1系统架构图

## 4.2 系统功能模块设计

### 4.2.1 系统功能图

2.5次元论坛系统的功能模块图如图4.2所示。

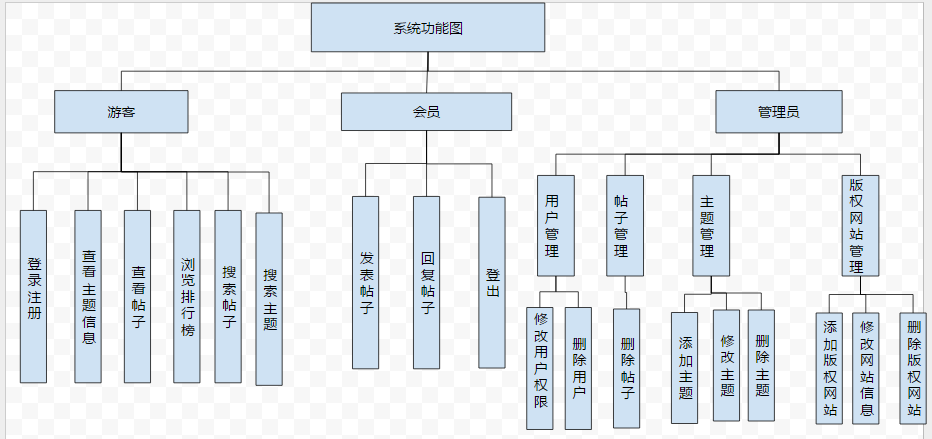


图4.2系统功能模块图

### 4.2.2 系统功能介绍

以下是各个功能的具体描述：

主题搜索：用户可以在论坛主页根据年份搜索主题；管理员用户则可以在论坛管理页面通过主题Id搜索需要操作的主题。

主题查看：用户可以通过在论坛主页点击主题的标题进入主题详情页查看主题的详细信息。

添加主题：管理员用户可以在论坛管理页面通过对主题信息的填写并提交完成主题的添加操作。

修改主题：管理员用户可以在论坛管理页面搜索到要修改的主题信息之后，对该主题的各类信息进行修改。

删除主题：管理员用户可以在论坛管理页面所有到要删除的主题之后点击删除完成对该主题的删除操作。

增加版权网站：管理员用户可以在论坛管理页面通过对版权网站域名和网站名的填写和提交完成增加版权网站的操作。

搜索版权网站：管理员用户可以在论坛管理页面通过网站名搜索版权网站的信息。

修改版权网站：管理员用户可以在论坛管理页面修改搜索到的版权网站信息。

删除版权网站：管理员用户可以在论坛管理页面删除搜索到的版权网站信息。

修改主题版权所属：管理员用户可以在论坛管理页面修改搜索到的主题的版权所属信息。

查看帖子：用户可以在主题详情页查看该主题下所有帖子的列表，同时可以通过对帖子标题的点击进入帖子详情页浏览帖子详细信心。

发布帖子：登录用户可以在主题详情页通过对帖子标题和帖子内容的填写并提交完成在该主题下发送帖子的功能。

回复帖子：登录用户可以在帖子详情页通过对回复内容的填写并提交完成在该帖子下进行回复的功能。

搜索帖子：用户可以在主题详情页通过帖子标题和帖子发帖人对该主题下的帖子进行查找。

点击记录：所有在论坛主页对主题标题的点击都会被记录下来，用于主题点击榜的分析。

发帖记录：所有用户发布帖子的时候都会被记录下来，用于主题评论榜的分析。

回复记录：所有用户回复帖子的时候都会被记录下来，用于主题评论榜的分析。

生成排行榜：服务器自动每隔10分钟对所有主题分别对点击数和评论数进行排名，同时更新点击榜表和评论榜表。

查看排行榜：所有用户都可以在排行榜页面查看当前所有主题的点击排行榜和评论排行榜。

删除帖子：管理员用户可以在论坛管理页面通过主题Id，帖子标题和发帖人对帖子进行查找，同时可以通过点击搜索到的帖子对应的删除按钮完成对帖子的删除操作。

登录：注册用户可以在任意页面通过对导航栏登录按钮的点击和用户名，密码信息的填写和提交完成登录操作。

注册：未注册用户可以在任意页面通过对导航栏注册按钮的点击和用户名，密码信息的填写和提交完成注册操作。

登出：登录用户可以在任意页面通过对导航栏登出按钮的点击完成登出操作回到未登录状态。

修改用户权限：管理员用户可以在论坛管理页面通过修改搜索到的用户的权限等级并提交完成用户权限的修改操作。不同权限的用户可以使用不同的功能。

删除用户：管理员用户可以在论坛管理页面通过点击搜索到的用户对应的删除按钮完成对对应用户的删除操作。

登录状态验证：系统会在用户登录后把用户的登录状态和登录信息记录在浏览器的localStorage中，用户在浏览所有页面的时候都会对这些信息进行查询和提交，如果用户已经登录同时账号和密码匹配正确，则直接进入登录后的页面。

# 5系统实现与测试

## 5.1 系统各个功能模块介绍

系统目前有三个权限等级的用户，分别是游客用户，会员用户和管理员用户。不同的权限等级可以实现不同的操作。

游客用户能够使用的功能如下：

1. 浏览主题：能够浏览主页所有主题列表，主题详情，主题版权所属信息等主题相关信息。
2. 查看帖子：能够浏览主题页面所有帖子列表，帖子详情已经帖子内部回复信息。
3. 查看排行榜：能够浏览排行榜页面所有排名信息，包括排行榜内主题的热度信息。
4. 搜索帖子：能够在帖子列表页面根据帖子id，帖子标题，发帖人等信息对帖子进行组合模糊查询。
5. 登录功能：如果游客用户是已注册用户，那么可以通过在登陆系统填写用户名和密码信息完成登录操作并进入登录之后的页面。
6. 注册功能：游客用户可以通过向注册系统填写注册信息完成注册操作，已注册用户可以使用登录功能。
7. 搜索主题：可以在论坛主页通过年份搜索主题。

会员用户能够使用的功能如下：

1. 查看主题：能够浏览主页所有主题列表，主题详情，主题版权所属信息等主题相关信息。
2. 搜索主题：能够在论坛主页通过年份搜索该年份下的所有主题。
3. 查看帖子：能够浏览主题页面所有帖子列表，帖子详情已经帖子内部回复信息。
4. 查看排行榜：能够浏览排行榜页面所有排名信息，包括排行榜内主题的热度信息。
5. 搜索帖子：能够在帖子列表页面根据帖子id，帖子标题，发帖人等信息对帖子进行组合模糊查询。
6. 发布帖子：能够在主题页面内通过填写帖子标题，帖子内容的信息在该主题内添加帖子。
7. 回复帖子：能够在帖子详情页面通过对回复内容的填写完成对该帖子的回复。
8. 登出功能：已登陆用户可以通过对登出按钮的点击实现登出功能，用户登出后，权限等级回归游客用户，同时显示未登录页面。

管理员用户能够使用的功能如下：

1. 删除帖子：根据帖子的id，在数据库中删除对应的帖子。
2. 修改用户权限：用户的权限等级标志了用户可以使用的操作类型修改用户的权限等级，管理员用户可以根据用户名修改对应用户的权限等级。
3. 删除用户：管理员用户可以删除指定用户。
4. 添加主题：管理员用户可以通过对主题标题，所属年份，简介信息，版权归属网站信息的填写完成对主题的添加操作。
5. 修改主题：管理员用户可以在轮胎管理页修改主题信息。
6. 删除主题：管理员用户可以在论坛管理页删除主题信息。
7. 添加版权网站：管理员用户可以通过对版权网站的域名以及网站名的填写，向数据库中添加版权网站的信息；版权网站被添加之后会自动在下次添加主题的版权归属网站选择页面显示。
8. 修改版权网站：管理员用户可以在论坛管理页修改版权网站信息。
9. 删除版权网站：管理员用户可以在论坛管理页删除版权网站信息。

## 5.2 系统功能模块实现

### 5.2.1登录模块实现

可选的用户类型有用户和管理员两种，用户对应普通用户，管理员对应管理员用户，管理员用户登录后可以进入管理员操作页面，而普通用户则不可以。具体的功能流程图如图5.1所示。

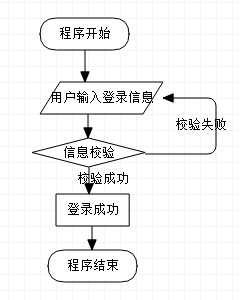


图5.1用户登录流程图

登录实现的页面如图5.2所示。



图5. 2登录功能页面

### 5.2.2 注册模块实现

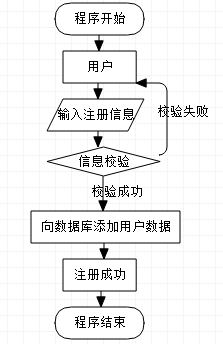
注册模块目前只提供普通等级的用户注册功能，用户可以通过在注册页面提交注册信息完成注册操作。注册后的用户可以使用登录功能。注册功能流程图如图5.3所示：

图5. 3注册功能流程图

注册功能实现的页面如图5.4所示。

图5. 4注册页面

### 5.2.3 发布帖子模块实现

登录普通用户和登录管理员用户能够在主题页面内通过填写帖子标题，帖子内容的信息在该主题内添加帖子。添加的帖子会在刷新后显示在主题主页中。发布帖子的页面如图5.3所示：

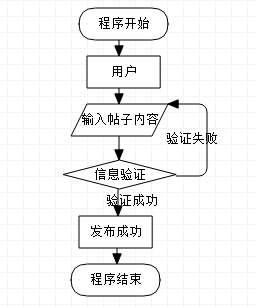
发布帖子的功能流程图如图5.5所示：

图5. 5发帖功能流程图

发布帖子功能实现后的页面如图5.6所示：

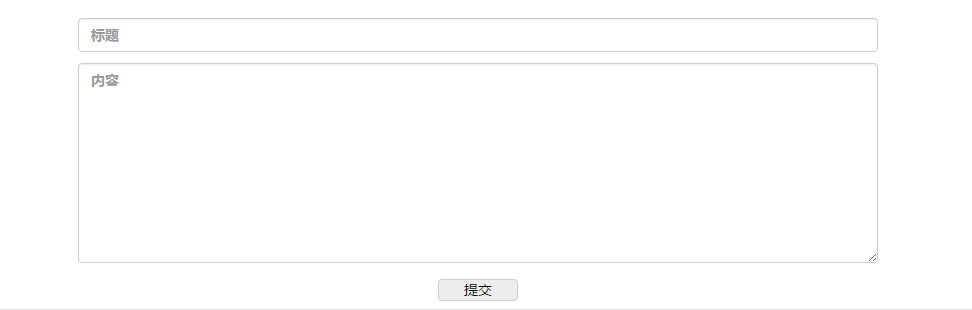


图5. 6发帖功能页面

### 5.2.4 回复功能实现

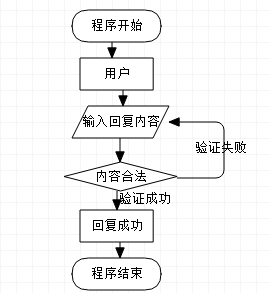
登录用户可以在帖子详情页通过对回复内容的填写并提交完成在该帖子下进行回复的功能。添加回复功能的流程图如图5.7所示：

图5. 7回复功能流程图

添加回复功能实现的页面如下图5.8所示：

图5. 8回复功能页面

### 5.2.4 主题管理模块实现

管理员用户可在论坛管理页面通过对主题信息的填写和提交完成对主题的添加操作，添加后的主题会在用户浏览主题的时候显示。主题管理可以分为添加主题，搜索主题，修改主题，删除主题这四个功能。

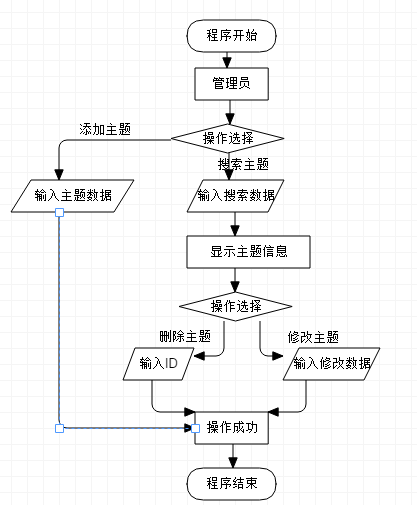
主题管理功能流程图如图5.9所示：

图5. 9主题管理功能流程图

添加主题模块的页面如下图5.10所示：



图5. 10添加主题页面

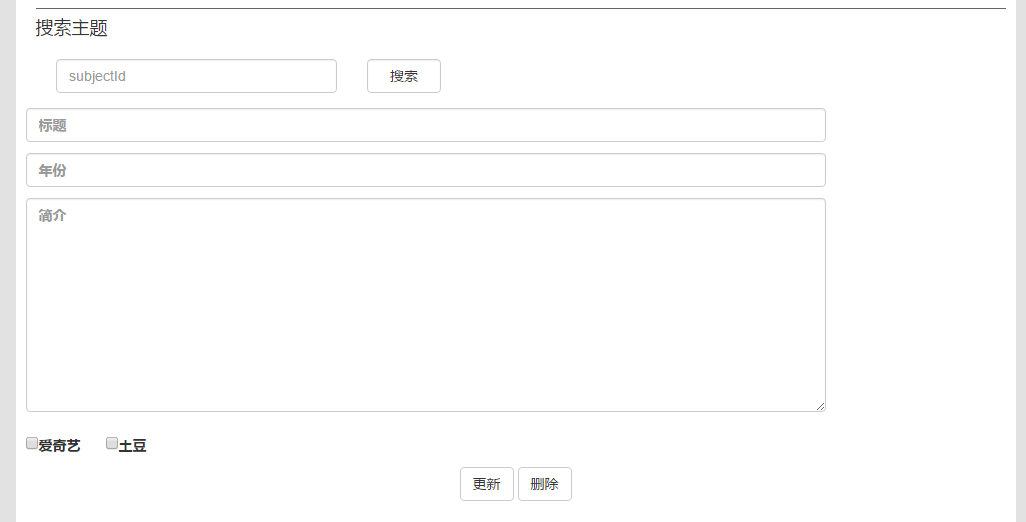
****因为搜索主题和更新、删除主题的页面相似，所以只放一张图片，图片如下图5.11所示：

图5. 11主题管理页面

### 5.2.5 用户管理模块实现

管理员用户可在论坛管理页面对指定id的用户进行删除和权限修改操作，目前可提供的权限等级有普通和管理员两种。

用户管理流程图如图5.12所示：

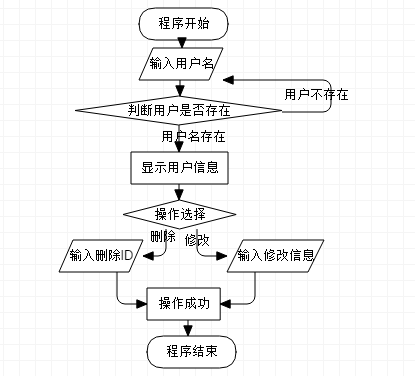


图5. 12用户管理流程图

用户管理模块的页面如下图5.13所示：

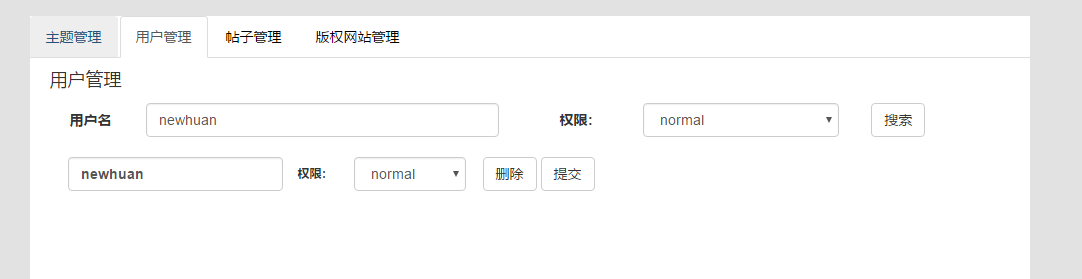


图5. 13用户管理页面

### 5.2.6 版权网站管理模块实现

版权管理同时可以分为：版权网站添加，版权网站搜索，版权网站修改，版权网站删除四个功能。管理员用户可在论坛管理页面通过对版权网站信息的填写和提交完成对版权网站的添加操作，添加后的版权网站会在管理员添加主题选择版权网站的时候显示。

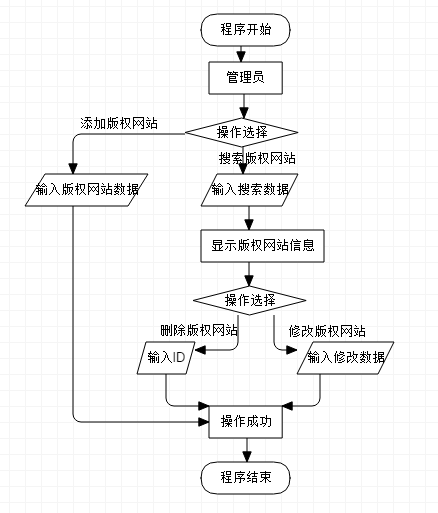
版权网站管理功能流程图如图5.14所示：

图5. 14版权网站管理流程图

添加版权网站模块的页面如下图5.15所示：



图5. 15版权网站添加页面

版权网站信息搜索、修改和删除具体实现页面如图5.16所示：

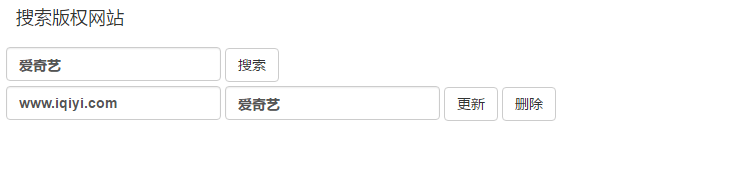


图5. 16版权网站管理页面

### 5.2.7 帖子管理模块实现

帖子管理主要是帖子搜索和删除功能。管理员用户可以在论坛管理业通过帖子id，帖子标题和发帖人用户名等信息对帖子进行组合和模糊查询，查询出的帖子回以列表的形式在页面上显示出来。

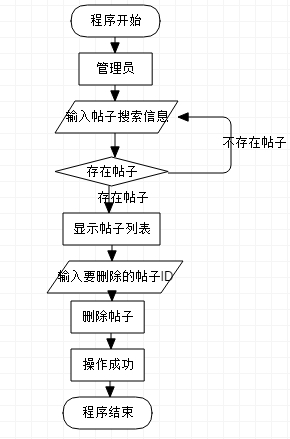
帖子管理功能流程图如图5.17所示：

图5. 17帖子管理流程图

帖子管理模块的实现页面如图5.18所示，管理员用户可以通过点击帖子对应的删除按钮对帖子进行删除操作。

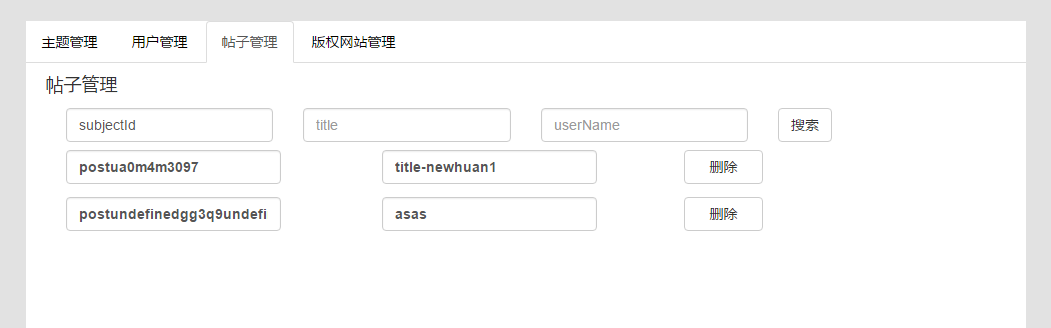


图5. 18帖子管理页面

### 5.2.8 查看帖子模块实现

游客用户，登录普通用户和登录管理员用户均可以通过在主题详情页浏览该主题下所有的帖子列表，也可以在此页面通过对帖子标题的点击进入帖子详情页面。

查看帖子模块的具体实现页面如下图5.19和5.20所示：



图5. 19帖子列表页面

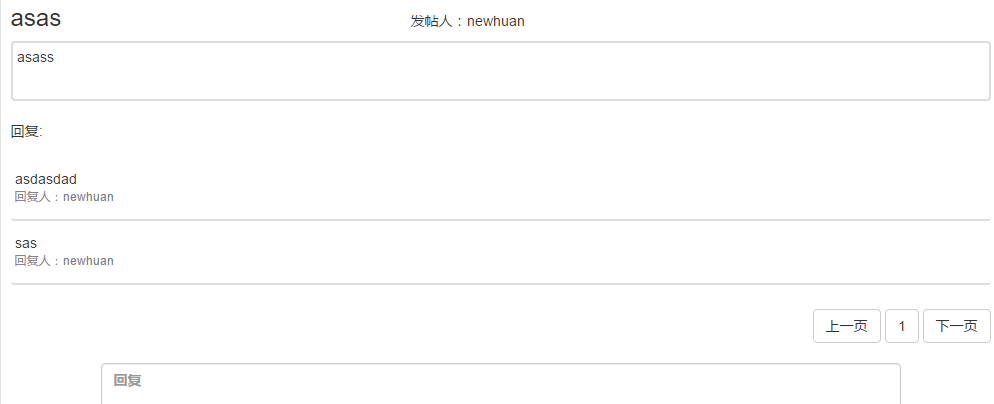


图5. 20帖子详情页面

### 5.2.9 查看主题模块实现

游客用户，登录普通用户和登录管理员用户均可以通过在论坛主页浏览某年份下所有的主题列表，也可以在论坛主页通过对主题标题的点击进入主题详情页面。查看主题模块的页面如下图5-21和5-22所示：



图5. 21主题列表页面

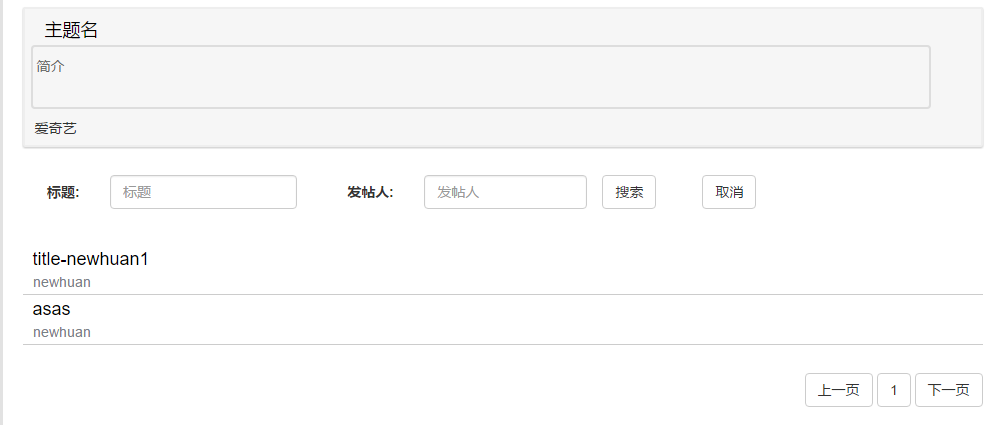


图5. 22主题详情页面

### 5.2.10 查看排行榜模块实现

游客用户，登录普通用户和登录管理员用户均可以通过在论坛主页点击排行榜或者直接输入排行榜页面网址进入排行榜页面，在该页面下可以浏览主题分别根据点击数和评论数生成的排行榜列表，通过对主题名的点击可以进入对应的主题详情页面。浏览排行榜模块的页面如下图5-23所示：

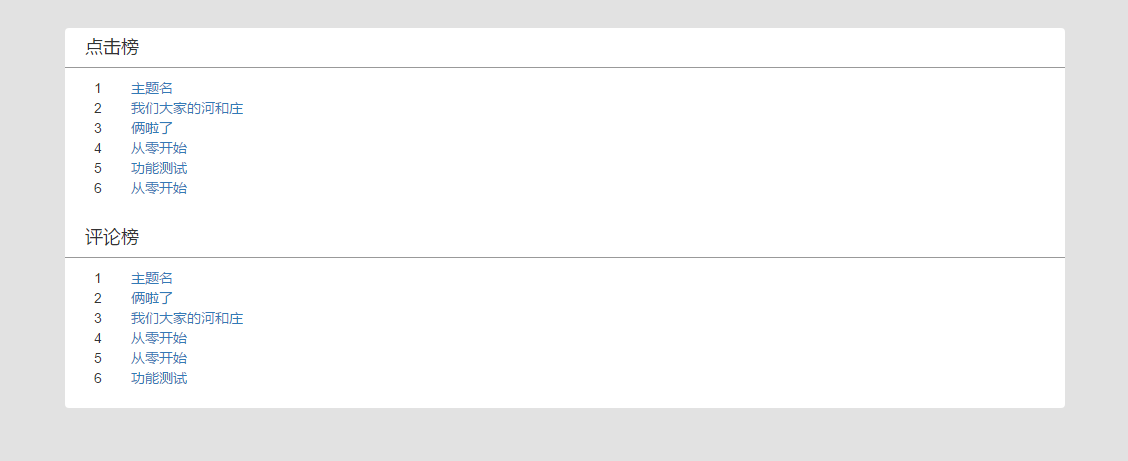


图5. 23排行榜页面

此列表由后台自动每隔10分钟刷新一次，管理员没有修改权限。

## 5.3.系统测试

软件的系统测试是在完整的集成系统上进行测试，以评估系统是否符合其规定的要求。 系统测试属于黑匣子测试的范围，因此，不需要知道代码或逻辑的内部设计。

通常，系统测试将所有通过集成测试的“集成”软件组件以及与任何适用的硬件系统集成在一起的软件系统本身作为输入。 集成测试的目的是检测集成在一起的软件单元（称为组合）之间或任何组合和硬件之间的任何不一致。 系统测试是一种更有限的测试类型; 它试图在“组合之间”以及整个系统内发现缺陷。

在功能要求规范（FRS）和/或系统要求规范（SRS）的上下文中，对整个系统执行系统测试。 系统测试不仅测试设计，而且测试客户的行为甚至相信的期望。 它还旨在测试软件/硬件要求规范中定义的界限。

### 5.3.1 功能测试

功能测试就是运行软件的所有功能，以验证软件系统有无严重错误。下面是对系统几个重要功能的测试：

(1) 添加主题功能测试

添加主题操作页面如下图5.24所示：



图5. 24添加主题操作页面

主题添加成功后在论坛首页显示的情况如下图5.25所示：



图5. 25添加主题后浏览页面

(2)帖子管理功能测试

帖子管理操作页面如下图5.26所示：

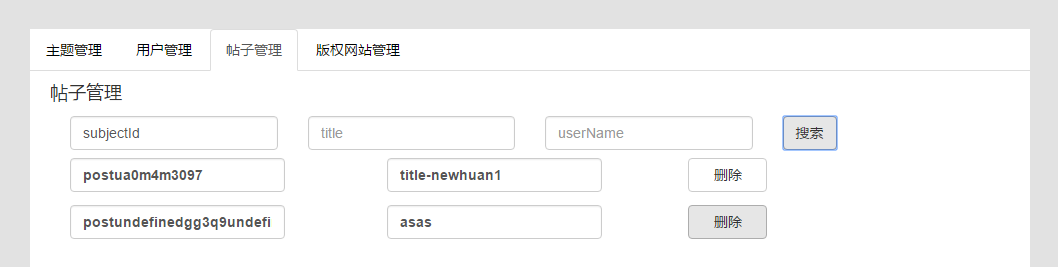


图5. 26帖子管理操作页面

操作成功后再次搜索被删除的帖子时，不会显示，详情如下图5.27所示：

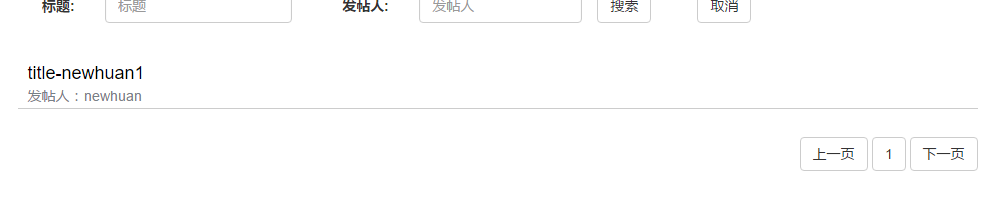


图5. 27操作结束页面

(3)帖子添加功能测试

帖子添加功能的操作页面如下图5.28所示：

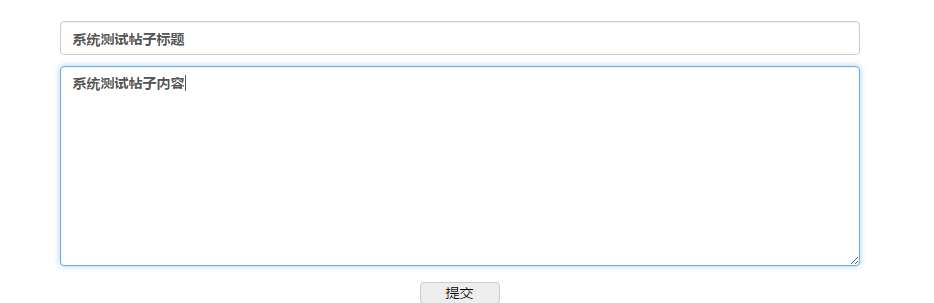


图5. 28帖子添加操作页面

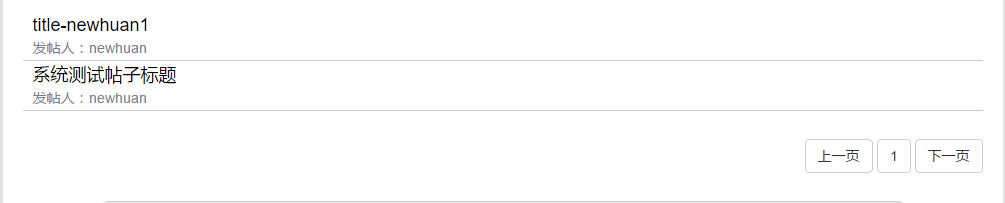
添加完成后刷新页面，新提交的帖子显示在了功能测试主题详情的帖子列表下，详情如下图5.29所示：

图5. 29添加完成页面

### 5.3.2 安全性测试

安全性测试是有关验证应用程序的安全服务和识别潜在安全性缺陷的过程。由于攻击者没有闯入的标准方法，因而也没有实施安全性测试的标准方法。目前只能通过测试人员扮演非法入侵者，尝试测试人员曾经试用过的入侵手段进行测试。本系统安全测试主要从以下几点完成：

1）系统登陆安全测试。系统登录时，无论是何等级用户都必须使用正确的账号和密码登录，而无效的账号或密码是无法登陆系统的。

2）系统管理权限安全测试。通过分别使用管理员级、用户级账号进行系统操作，没发现有任何越权或权限混乱现象。

3）系统数据安全性。本系统对所有账号、密码数据都进行加密处理。对数据库进行伪文件处理（数据库文件后缀名更改成其它格式文件），且具有良好的数据库备份和恢复机制。

### 5.3.3 可用性测试

可用性测试是以用户为中心的交互设计中用于通过对用户进行测试来评估产品的技术。 这可以被看作是不可替代的可用性实践，因为它直接提供了真实用户如何使用系统。 这与专家使用不同方法评估用户界面而不涉及用户的可用性检查方法形成对比。 最终用户可能不会注意会发生什么可用性测试侧重于衡量人造产品满足其预期用途的能力。 通常受益于可用性测试的产品的示例包括食品，消费品，网站或Web应用程序，计算机接口，文档和设备。 可用性测试测量特定对象或一组对象的可用性或易用性，而一般的人机交互研究则试图制定普遍原则。

本系统通过测试人员模拟用户，对系统所有功能都进行了可用性测试，通过对测试人员测试结果的汇总，测试人员并未发现明显的不可用功能模块

### 5.3.4 性能测试

性能测试的目的是使用不同类型的性能测试（例如负载、压力和饱和测试）来确定定义的工作负载下的系统性能。本测试通过对系统运行速度、网络响应时间进行测试，测试结果如下：

1. 系统运行速度：通过测试人员在不同测试机上进行测试，没有发现明显的延迟、卡顿现象。
2. 网络响应时间：测试人员通过使用chrome浏览器的网速调节功能分别在200kb\s,500kb\s和1Mb\s的环境下分别测试，响应时间均小于5s，通过调查结果显示，5s小于用户可接受的最大延迟等待时长。

### 5.3.5 测试结果分析

通过对上述测试结果的综合分析，本系统符合一般论坛系统的用户要求，所有预期功能全部实现，操作简单明了，安全性高，性能较好，是一个可以信赖的论坛系统。

# 结论

近七年以来，网络版权日益受到重视，这是一件保护创作者智力成果的大好事，但是也不免带来了一些麻烦，动漫版权分散问题就是本人亲身体验到的版权收到重视带来的最大烦恼，本系统就是为了解决这个问题而产生的一个解决方案。

在历时半年的调研，需求分析，系统设计和系统实现之后，最终成果就是上面介绍的《2.5次元论坛》系统，充分考虑针对用户人群——动漫爱好者的需要，着重解决了网络版权带来的动漫信息分散问题，即可以让用户方便的获取各种动漫的版权所属信息，方便用户有选择的购买版权网站的vip权限，也为动漫爱好者们量身打造了一个极佳的交流平台。

现在我可以很放心地说，《2.5次元论坛》系统确实是一个能解决问题的系统，同时我也对“计算机软件必然是为了解决问题而出现”这一理念有了更加深刻的理解，计算机软件就是一些解决问题的工具，而软件工程就是一门解决问题的学科!

# 参考文献

[1] 王珊，萨师煊.数据库系统概论 [M].北京:高等教育出版,2014：45-60.

[2] Ethan Brown.Node与Express开发[M].吴海星，苏文.北京：中国邮电出版社 ,2016：20-45

[3] 李松峰，李丽正则表达式入门经典[M].北京：人民邮电出版,2011：6-11

[4] Kristina Chodorow，mongodb权威指南[M].邓强，王明辉.北京：人民邮电出版,2014： 17-44

[5] 单东林,张晓菲,魏然.锋利的jQuery[M].北京：人民邮电出版社，2012:14-135

[6] 朴灵.深入浅出Node.js [M].北京：人民邮电出版社,2016:15-80

[7] 弗兰那根.javascript权威指南.[M].淘宝前端团队,北京：机械工业出版社,2012 :13-200

[8] 尼古拉斯.javascript 高级程序设计[M].李松锋，曹力.北京：中国邮电出版社,2012 :15-148

[9] Hunt Andrew&D.Thomas.The pragmatic programmer[M].美: Addison-Wesley Professional,2004:20-182

[10]上野宣 图解Http[M].于均良.北京：人民邮电出版社,2014: 3-14

[11] 黄文博&燕杨 C/S结构与B/S结构的分析与比较[J]，长春师范大学学报，2006, (25)

[12]Simpson Kyle.You Don’t Know JS:Scope&Closures[M].美:O’Reilly,2016:15-20

[13] Wendy Boggs, Michael Boggs. Mastering UML with Rational XDE [M]. Publishing House Of Electronics Industry，2003:11-56.

[14] Simon Holmes. Getting MEAN with Mongo, Express, Angular, and Node [M].美: Addison-Wesley Professional,2013:40-82

[15] Kyle Simpson.你不知道的javascript（上卷）[M].赵望野，梁杰.北京：中国邮电出版社,2015 :17-168

[16] Kyle Simpson.你不知道的javascript（中卷）[M].单业，姜南.北京：中国邮电出版社,2016 :33-148

[17] Douglas Crockford.Javascript语言精粹.[M]赵泽欣，鄢学鹍.北京：电子工业出版社，2009：41-85

[18] Tom Hughes-Croucher.Node.[M].英：O'Reilly Media，2012：24-69

[19] Ethan Brown.Node与Express开发.[M].吴海星，苏文.北京:中国邮电出版社，2015：50-72

[20] Paul Graham.黑客与画家.[M].阮一峰.北京：人民邮电出版社：2011:30-58

[21] 薛华成.管理信息系统[M].北京.第一版，清华大学出版社,1993:20-50

# 致 谢

时光荏苒，白驹过隙，转眼四年将过；回首往昔，感慨良多，感激弥深。

这里，我首先要感谢我的指导老师崔鑫老师耐心细致的指导，她在我整个课题开发和设计的过程中给予了我极大的帮助，如果没有导师的辅导和指点，我论文绝不可能顺利完成，感谢崔老师，非常感谢！

然后，对学校和学校各位老师的感谢也是必不可少的，如果没有学校和各位老师无怨无悔地对我的悉心教导，这里的这些文字恐怕永远都不会出现。感谢学校，感谢所有的老师们!

当然，各位同门们的恩情也是不敢或忘的，感谢你们四年来的陪伴，因为有你们的陪伴我才能走到现在，谢谢各位！

最后，但是一样感激的还有所有的开源代码贡献者和开放博客书写者们，正是因为有这些人的存在，软件这一行业才会如此兴旺发达，我们在学习的过程中才能如此快速的进步，感谢你们!

# 附录一 外文译文

**Nodejs概述**

Node.js允许使用JavaScript和一系列处理各种核心功能的“模块”来创建Web服务器和网络工具。为文件系统I / O，网络（DNS，HTTP，TCP，TLS / SSL或UDP），二进制数据（缓冲区），加密功能，数据流和其他核心功能提供模块。 Node.js的模块使用一个旨在降低编写服务器应用程序的复杂性的API .

Node.js应用程序可以在Linux，macOS，Microsoft Windows，NonStop和Unix服务器上运行。他们还可以用CoffeeScript（JavaScript替代），Dart或Microsoft TypeScript（强类型JavaScript）或可以编译为JavaScript的任何其他语言编写。

Node.js主要用于构建Web服务器等网络程序。 Node.js和PHP之间最大的区别是，PHP中的大多数函数都将被阻塞直到完成（仅在之前的命令完成后才执行命令），而Node.js中的函数被设计为非阻塞（命令并行执行并使用回叫信号完成或失败）。

Node.js将事件驱动的编程带给Web服务器，从而实现了JavaScript中快速Web服务器的开发。开发人员可以创建高度可扩展的服务器，而无需使用线程，通过使用事件驱动编程的简化模型，该模型使用回调来表示完成任务。创建了Node.js，因为并发在许多服务器端编程语言中是困难的，并且通常导致性能下降。 Node.js将脚本语言（JavaScript）与Unix网络编程的强大联系起来。

Node.js建立在Google V8 JavaScript引擎上，因为它是BSD许可下的开放源码，速度非常快，并且具备互联网基础知识（如HTTP，DNS，TCP）。此外，JavaScript是一种众所周知的语言，使得Node.js可以在整个Web开发社区中立即访问。

# 附录二 外文原文