



分类号.....

UDC.....

密级 公 开.....

黑龙江大学

硕士研究生学位论文

Node.JS 博客系统的设计与实现

申 请 人: 邹竞莹

学 号: 3131873

培养单位: 计算机科学与技术学院

学科专业: 计算机科学技术

专业方向: 数据库应用

指导教师: 于海洋 副教授

李景文 高级工程师

完成日期: 2016 年 10 月 9 日

中文摘要

在互联网社交发展越来越迅速的今天，博客不但是一个新的、轻量级的新闻业，而且同样是技术分享的方法。撰写博客的驱动是各种各样的，博客包含的内容也是多样化的。从网络日志到科技技术议题的评论；从个人的情绪表达达到社区、团队的支持分享。

在技术飞速发展的现在，博客技术也相对发展迅速，从最初的 C 或 C++ 书写 CGI 到后来风靡一时的 PHP 到重量级的 WordPress。最近几年内相对流行的 Ruby on Rails，Python 都有其相对的博客引擎来实现博客系统的搭建。本设计的博客采用最新的技术来实现。Node.js 使用事件驱动，非阻塞 I/O 模型而得以轻量 and 高效，非常适合在分布式设备上运行的数据密集型的实时应用。对于 Node.js 的异步 I/O 的特性，比较一下几个数据库的特点，Mysql 关系型数据库、Redis Key 缓存数据库和 MongoDB 分布式文件存储的数据库，它是介于关系数据库和非关系数据库之间，而且 MongoDB 的数据在 DB 中的最小单位是 Document，适合博客的数据存储，所以本博客系统将采用 MongoDB 作为存储。

本文对现阶段国内外的博客进行简单的分析和介绍，根据实际的研究情况提出了本课题的研究目的和意义。为了构建更为人性化的交互界面，本博客系统将采用 Express.js 为 Node.js 的 Web Framework，Jade 为 Web 模板引擎，SASS 作为 CSS 的预编译和 Mongoose 模块来管理 MongoDB，采用基于 B/S 的 MVVM 结构的 Web 应用系统。本系统主要包括文章模块、用户模块和评论模块。文章模块主要功能是文章的发布和编辑；用户模块用于博主的登录，以及用户浏览文章后对文章的评论。

关键词：Nodejs；MVVM；npm 包管理；MongoDB；异步

Abstract

In today's social development of the Internet is more and more quickly, the blog is not only a new, lightweight, journalism, and technology is the same way. Writing a blog drive is various, blog contains content is also diversified. From the web log to review issues in science and technology; From the individual's emotional expression of share to the community, team support.

Rapid development in technology now, the blog technology is also relatively rapid development, from the initial C or C++ to write CGI PHP by heavyweight WordPress later. In recent years is relatively popular Ruby on Rails, Python has its relative blog engine to realize the blog system. The design of the blog using the latest technology to achieve. NodeJS using event-driven, non-blocking I/O model and light weight and high efficiency, very suitable for real-time in the distributed equipment running on data intensive applications. For the NodeJS asynchronous I/O features, compare the characteristics of several Database, MySQL relational database, Redis Key cache database and directing a distributed file storage distributed file storage database, it is between relational databases and the relational database, and directing the smallest unit of data in DB is the Document, is suitable for the data storage, blog so this blog will use mongo as a storage system.

In this paper makes a brief analysis and introduction of the domestic and foreign blog at this stage. according to the research of the actual situation proposed this topic research purpose and meaning. In order to build a humanized interface, Express is adopted in this blog system. The js for the NodeJS Web Framework, Jade as a Web template engine, SASS as precompiled of CSS and Mongoose module to manage the mongo, MVVM based on B/S structure of Web application system. This system mainly includes articles module, user module and comment module. The article module is the main function of the article publishing and editing; User login module with bloggers, and user comments on the article after article.

Keywords: Node.js; Express.js; npm package manager; MongoDB; Async

目 录

中文摘要	III
ABSTRACT	IV
第 1 章 绪 论	1
1.1 项目的背景及意义	1
1.1.1 研究的背景	1
1.1.2 研究的意义	1
1.2 国内外研究现状	2
1.2.1 国外博客平台主要研究成果	2
1.2.2 国内博客系统主要研究成果	3
1.2.3 当前网络博客存在的问题及发展趋势	4
1.3 本文所做的工作	5
1.4 本文的主要内容及结构	5
第 2 章 系统相关技术介绍	7
2.1 B/S 结构技术介绍	7
2.1.1 B/S 架构简介	7
2.1.2 B/S 架构的优点	7
2.1.3 B/S 结构的缺点	9
2.2 NODEJS 介绍	9
2.2.1 NodeJS 简介	9
2.3 MONGODB 介绍	10
2.3.1 MongoDB 简介	10
2.3.2 MongoDB 优点	10
2.3.3 MongoDB 的缺点	12

2.4 MVVM 架构介绍	13
2.5 相关工具介绍	14
2.5.1 ExpressJS 介绍.....	14
2.5.2 Jade 介绍.....	14
2.5.3 Sass 介绍.....	15
2.5.4 YSlow 介绍	15
第 3 章 系统分析	16
3.1 系统可行性分析	16
3.2 功能需求分析	16
3.2.1 登录模块需求分析	17
3.2.2 文章管理需求分析	19
3.2.3 评论管理需求分析	19
3.2.4 页面需求分析	20
3.3 非功能性需求分析	21
3.3.1 性能需求分析	21
3.3.2 安全性需求分析	22
3.4 数据库需求分析	22
第 4 章 系统设计	24
4.1 数据库概述	24
4.2 数据库的选择	24
4.3 数据库概念设计	24
4.4 数据库逻辑设计	25
4.4.1 用户登录表	25
4.4.2 用户组登录表	26
4.4.3 文章信息表	26
4.4.4 评论信息表	27

4.4.5 其他相关信息表	27
第 5 章 系统实现	28
5.1 系统架构设计	28
5.1.1 表现层	29
5.1.2 业务逻辑层	32
5.1.3 数据模型层	33
5.2 MD5 加密	34
5.3 系统的依赖与配置	34
5.3.1 系统依赖安装	34
5.3.2 系统框架及优化工具	35
5.4 系统页面展示	36
5.4.1 系统首页展示	36
5.4.2 文章页面展示	37
5.4.3 评论页面展示	40
5.4.4 关于页面展示	40
第 6 章 系统测试	42
6.1 系统测试	42
6.1.1 测试目的	42
6.1.2 测试方法	42
6.2 测试用例	43
6.2.1 用户功能测试	43
6.2.2 文章功能测试	43
6.2.3 评论功能测试	44
6.2.4 关于页面测试	44
第 7 章 总结与展望	45
参考文献	46

致 谢	50
独创性声明	51

第 1 章 绪 论

1.1 项目的背景及意义

1.1.1 研究的背景

随着计算机技术和互联网的飞速发展，信息技术在人们的生活方式，工作方式，以及教育领域产生了极大影响^[1]。传统的媒体行业已经不能满足人们对信息的获取的多样化需求，网络博客已经成为了信息传递的重要途径和手段^[2]。网络博客，就是把网络为载体，简单快捷方便地发布自己的心得与经验，能够及时且有效的跟他人进行交流，以丰富且个性化展示为一体的综合性分享平台^[3-4]。至今博客以非常受大家青睐、欢迎，代表着新的工作方式和新的生活方式，更代表着人们学习知识的新方式。确切的来说，博客在互联网上可解释为特定的软件，在网络上能够出版，发表和传播个人的优秀文章^[5]。

博客被称为是一种新型的、轻量级的新闻业，它是自媒体、大众媒体、传统新闻行业之外信息传播的一个方法^[6]。博客也是一种技术分享的方法。撰写博客的驱动是各种各样的，博客包含的内容也是多样化的。从网络日志到科技技术议题的评论；从个人的情绪表达达到社区、团队的支持分享^[7]。在技术飞速发展的现在，博客技术也相对发展迅速，从最初的 C 或 C++ 书写 CGI 到后来风靡一时的 PHP 到重量级的 WordPress。最近几年内相对流行的 Ruby on Rails，Python 都有其相对的博客引擎来实现博客系统的搭建^[8]。

1.1.2 研究的意义

在互联网快速发展的今天，设计结构清晰、功能健全、技术先进的博客系统成为了具有重要意义的工作。研究意义主要体现在如下几个方面：

- 1、降低了信息传递的成本，节省了大量的人力物力以及资源的浪费，避免了资源的重复消耗。运用计算机网络的技术优势，把先进的网络技术与传统的信息传递方式相结合，融合传统媒体形式的优点，为信息的传播和人们获取信息的途径提供了一个有效而实用的系统^[9]。

2、加快传统信息媒体行业与信息技术的有机整合，发挥技术优势提高信息传播的速度和广度。随着智能设备终端的广泛普及，人们可以利用碎片时间进行信息的获取，而不再局限于传统的电视、报纸等基本形式，而且网络中的信息并不仅涵盖于文字，还包含图像和音乐视频资源等等^[10-11]。博客技术能将网络中的信息进行有效的整合，形成多媒体的信息资源，改善了传统信息产业单调的形式^[12]。

3、博客技术带来了信息获取观念以及信息传递模式上的转变^[13]。信息首重可以根据自己的喜好、兴趣等，将博客文章进行转发，从而实现信息的二次传递。博客系统将传统信息流动的由点发散的方式演进为一个庞大的信息流网络，信息不再是由单一的信息源单向传播到受众，而是一个双向交互过程^[14]。

4、网络博客带来了灵活便捷的信息获取方式。博客系统让每个人作为信息受众的同时成为了信息的发布者，这一点在一定程度上影响了许多领域，学生可以通过博客系统的信息尽心学习，依靠互联网这个成本低廉的载体，降低了学习的门槛，并提高了学习的效果和灵活性^[15]。

1.2 国内外研究现状

1.2.1 国外博客平台主要研究成果

国外发达国家在博客、网络新闻、知识传播方面的开发相对我国较为先进，无论是技术方面还是博客系统的优良性都要好于国内，因此针对博客系统的研究也就相对比较多，经过激烈的竞争后出现了一些较为优秀的博客系统平台，其中比较流行且使用较为广泛的有：

1、Medium 是互联网上一个全新的轻量级的内容文章发布平台，允许单人或多人协作写作，可以用户将自己创作的内容作为主题的形式集结为专题，分享给用户进行阅读和传播。Medium 强调以视觉化的图文样式来提高用户阅读体验、用专题为核心的内容组织成集合内容、通过用户赞成的形式进一步精选一系列文章、以精美的网站模版提高用户的阅读体验^[16]。对于 Medium 这种新的内容或文章发布形式来说，品牌、作者这些都不是最重要的，专题才是核心。Medium 不希望用户只是因为某个作者所写的某篇文章而去阅读，而是希望用户的阅读理由是因

为该文章的内容是其真正的关注点^[17]。

2、Movable Type 博客系统，简称为 MT，是由 Six Apart 公司推出的网络博客发布系统。它是全球最受欢迎的网络博客系统之一，其中包含多人用户，主题，引用，评论等功能，而且广泛支持多种第三方插件。Movable Type 博客系统不仅可以用于个人的网络日志，还可以应用于教育、商业等领域^[18]。Movable Type 博客系统于 2007 年正式宣布开源。Movable Type 的平台开源，很大程度的增加了它的用户。

3、Drupal 系统的优势在于它的多功能性。它拥有强大的用户系统，且有大量的用户交互功能，Drupal 还带有一个大型社区的模块。开发者可以利用这些模块开发建立任何类型的网站。很多顶尖的网站都使用 Drupal 的社区和多人博客^[19]。Ubuntu，Spread Firefox，Performancing The Onion 等。Drupal 是一个相对完美的博客系统。

4、Joomla 是采用 PHP+MySQL 开发的一套系统，可以跨多平台运行的内容管理系统。Joomla 除了具有文档/图片管理，新闻/文章管理，模板/主题管理，网站布局设置等一些简单的基本功能外^[20]。还可以通过插件进行扩展功能，其中包含：论坛与聊天、博客、电子报、电子商务与购物车引擎、日历、广告管理系统、目录分类管理、期刊订阅、数据收集与报表工具服务等^[21]。

5、WordPress 系统是采用 PHP、MySQL 开发的开源内容管理系统和博客系统。WordPress 具有模板系统和多插件架构。提供多种个性主题，用户可以安装和切换多种主题。主题让用户在不改变博客结构和内容的情况下更改操作页面和站点的功能。在 WordPress 的外观管理工具中安装主题，也可以通过 FTP 上传到主题文件夹进行安装；也可以自定义主题。WordPress 中比较流行的一个特点是它有丰富的插件结构，插件可以使开发者扩展程序功能。多用户共同也是 Wordpress 的一大特点，目前互联网上最流行的博客系统之一^[22-23]。

1.2.2 国内博客系统主要研究成果

与国外的网络博客现状相比，我国的网络博客尚处于起步阶段。在我国网络

博客技术开发者的努力下，已经有越来越多的人开始使用网络博客，利用网络博客传播知识，相对已经出现一些有代表性的博客系统：

1、简书是一个将阅读和写作整合到一起的博客系统。目的是为用户打造优雅的阅读社区，为作者打造出优秀的写作工具。简书是定位在写作者的一个写作工具，非常简洁的界面，还有最大特色是能够支持 Markdown，为作者营造出一种沉浸的写作氛围，进而能够专注于写作^[24]。简书还支持传统写作模式，在简书笔记中可以打开写作模式即让写作窗口全屏化，再通过使用浏览器的全屏功能。

2、Z-Blog 系统是由 RainbowSoft Studio 开发的一款基于 Asp 的博客系统。支持插件、主题等功能和配置，使用户打造出个性化的博客；数据库可以使用 Access+MYSQL 双数据库，可以实现动态页面、伪静态、静态化；支持大部分 IE、Safari、Chrome、Opera、Firefox 多种浏览器；还有移动版博客，支持智能手机的管理以及离线写作系统^[25]。

3、emlog 是翻译后意为点滴记忆。它是一款采用 PHP 和 MySQL 开发，功能很强大的博客以及内容管理系统^[26]。致力于为用户提供稳定、快速，且在用户使用上能够非常简单、舒适。

1.2.3 当前网络博客存在的问题及发展趋势

国外的网络博客系统虽然有功能强大、技术先进、内容相对完善，但它存在着一些不可避免的问题。博客的空间浪费是一个很大的问题。在接近 4000 万的博客空间中，其中有超过 60% 的为睡眠博客即为平均一个月更新不超过一次的博客。因为睡眠博客的存在，对于互联网络上的资源造成了严重的浪费。博客托管服务商的服务不理想还导致用户的满意度低下。服务商提供服务时需要注意：注册项目不应过于繁琐，更不要涉及过多的个人隐私问题；日志管理界面应该尽可能简洁。博客作为发表文章的平台，其中要突出体现的是博主的个性，而不是托管服务商的个性。随着社交媒体的发展，博客呈现出越来越多的媒体化元素，出现了博客视频相音频、图片、及 RSS 等^[27]。

以上通过对国内外网络博客系统的研究，发现大部分产品较为臃肿的，不够

简洁，博客流动性问题。目前人们对功能、效率的追求提高了，对网络博客中的文章也是如此，为阅读者打造最优雅的阅读博客^[28]。因此，对于自主开发的基于 NodeJS 的博客系统就具有一定的优势，可以自主设计、开发的博客系统，满足个性化的需求，同时功能的扩展以及系统的升级需要很方便，要具有较好的实用性和灵活性^[29]。

1.3 本文所做的工作

本论文对国内外博客及网络新闻文章的现状、存在问题以及发展趋势进行了深入分析研究之后，结合博客系统的实用性及文章知识的传播性，运用了 NodeJS，MongoDB 非关系型数据库，ExpressJS 等开发工具，结合前沿的 Web 开发技术，最终实现了博客系统。

本系统主要包括文章模块、用户模块、评论模块。文章模块主要包括文章的添加，文章编辑，文章浏览，文章列表；用户模块用于博主登录以及文章的编写；评论模块主要用于用户浏览文章后的评论。

1.4 本文的主要内容及结构

本文论述的内容共分为 7 个章节，论文具体结构如下：

第 1 章为绪论部分。首先介绍了博客的背景、历史，随后根据调研博客系统的国内外现状，论述了博客系统的开发意义。最后对本文的结构进行简要介绍。

第 2 章为系统相关技术介绍部分。选择适合博客系统的开发工具和编程语言，讨论了各种技术的优势和劣势。

第 3 章为系统分析部分。包括系统的可行性的分析，需求分析，性能的规范，数据流图，数据分析。

第 4 章为系统设计部分。首先对系统的架构进行阐述，介绍了系统的设计思路和设计原则，阐述了 MD5 加密工具的使用以及系统的安装配置。根据系统的结构划分，分别对各功能模块进行了简要介绍。

第 5 章为系统实现部分。在前一章的基础上，对每个模块进行了详细的叙述，结合系统的运行图展现系统的运行情况及页面展示。

第 6 章为系统测试部分。阐述了测试的目的以及测试的重要性，并设计了测试用例，对系统进行了全面的测试。

结论部分。主要对本系统设计实现进行归纳和总结，并指出了系统尚有改进之处，根据具体环境的需求，对功能设计进行完善。

第 2 章 系统相关技术介绍

2.1 B/S 结构技术介绍

2.1.1 B/S 架构简介

NodeJS 博客系统使用三层 B/S 架构模式，即为浏览器、服务器结构^[30]。它是随着互联网技术的兴起，对 C/S 结构的改进或变化的结构。在 BS 结构下，用户操作界面是通过浏览器请求实现，很少一部分业务逻辑需要前端实现，但是大多数情况下主要业务逻辑是在服务器端实现，形成所谓的三层架构。B/S 结构是 WEB 开发兴起后的系统结构模式，浏览器是客户端最为主要的应用程序^[31]。统一客户端，将系统的核心功能集中到服务器上来实现，简化系统开发、维护以及使用。客户机上只需安装浏览器，如 Chrome、Internet Explorer 或 Firefox，服务器安装 SQL Server、Sybase、Oracle、Informix 或等数据库。浏览器客户端通过 Web 服务器与数据库进行数据交互^[32]。这样就能够极大的简化了客户端对电脑的载荷，减少了对系统的维护、升级的工作量和成本，降低了用户的整体成本。

2.1.2 B/S 架构的优点

B/S 架构是以 WEB 技术的快速发展中从传统的 C/S 架构发展得到的，而且一举成为现今主要的网络架构^[33]。目前大多数网站，尤其是商务类型的网站都是基于 B/S 架构开发的。B/S 结构优点如下：

1、浏览器

- (1) 能够面对广大用户，实现其信息共享传播。
- (2) 浏览器只需要处理一些简单的业务逻辑，负担较小。
- (3) 页面上的更新能够实现同步，所有的用户都可以看到。
- (4) 界面设计开发较简单，降低程序员的要求，减少大量培训。
- (5) 业务逻辑扩展简便，只需要增加业务扩展的页面就可以，减少大量的编程。
- (6) 界面无需编写太多，相对通用，修改页面也比较简单，所以几乎无需维

护。

(7) 由于 Web 的平台无关性, 可以从几个用户的工作组级, 一台服务器扩展出来, 做成拥有上万用户的大型系统。

2、服务器

(1) 数据都集中存储在数据库服务器中, 不存在数据不一致状态。

(2) 服务器数据库很安全, 能够有效地管理访问权限和保护数据平台。

(3) 选择的更多, 成本变得更少, 在数据库服务器上可以使用多种操作系统。

(4) 异地部署能力, 只需将服务器连接到专网即可, 实现远程的维护、共享和升级。

(5) 数据存储存储在数据库服务器, 客户端不能够存放数据库连接信息或任何业务逻辑数据, 因此数据相对比较安全。

(6) 随着服务器负载的增加, 可以建立集群服务器系统且不影响系统的功能, 在多个服务器间做负载均衡。

3、用户

(1) 外部用户也可以在浏览器上进行访问。

(2) B/S 结构建立在广域网上, 所需要的网速不高。

(3) 界面更加人性化, 更加通用化, 不需要大量培训就能够掌握。

(4) 无需安装客户端, 只要连上网, 就能都随时随地得浏览页面。

(5) 凭借互联网强大的信息传送能力和信息发布, 非常有效地解决企业内部大量且不规则的信息交换。

(6) 用户数基本上是无限制的, 所有的用户都需要通过 JDBC 连接缓冲池连接到数据库的, 用户不保持对数据库的连接。

4、成本

(1) 能够节省大量成本, 软件重用性非常强。

(2) 无需安装其他的软件, 客户端几乎为零维护。

(3) 开发相对简单, 升级较为便捷, 开发人员无需过多的培训, 能够节省大量的资金。

(4) 一般情况下只有初期需要一次性的投入成本,有利于避免 IT 黑洞和软件项目控制。

(5) 建立在广域网之上,不需要专门的网络硬件环境,比如租用设备,电话上网等,成本相对较小。

2.1.3 B/S 结构的缺点

虽然说 B/S 架构有很多的优越性,但是也有不可避免的缺陷。不过,在理论上,既然 B/S 是 C/S 的改进版,应该克服了 C/S 的很多缺陷,所以缺点应该不是很多。B/S 结构的缺点如下:

1、客户端

(1) 不突出个性,页面通用化。

(2) 费时,页面需要不断的刷新,尤其是在用户增多且网速慢的情况下。

(3) 由于无需安装客户端,客户端容易扩展,还有外部的用户也可访问,因此 B/S 面对大量不可统计用户。

2、服务器

(1) 当用户增多时,服务器的响应速度变慢。

(2) 功能虽多样化,但不能专一化,不能实现过于复杂的功能。

(3) 服务器承担很重要的责任,数据负荷量较大。服务器一旦出现宕机等问题,后果很严重。

2.2 NodeJS 介绍

2.2.1 NodeJS 简介

Node 是服务器端的 JavaScript,它将改变 JavaScript 在服务器的工作方式^[34]。它的目标是帮助开发者构建可伸缩且高度化的应用程序,编写能够处理上万条同时连接在一个物理机的高并发代码^[35-36]。

NodeJS 是个让 JavaScript 在浏览器之外运行的平台。它实现了网络通信、文件系统包、操作系统 API、模块、等 JavaScript 没有或不完善的一些功能^[37]。历史中要将 JavaScript 转移到浏览器之外的计划不止一个,但是 NodeJS 是最成功的一

个。随着 NodeJS 的成功，各种浏览器之外的 JavaScript 实现逐渐兴起，因而产生了 CommonJS 规范^[38]。CommonJS 试图拟定出相对的完整的一套 JavaScript 规范，用以弥补普通程序所需要的 API，比如文件系统的访问、模块管理、命令行、函数库集成等功能。CommonJS 的制定者希望服务端 JavaScript 遵循 CommonJS 规范，以方便代码复用和相互兼容^[39]。NodeJS 的部分实现遵循了 CommonJS，但是由于两者都处于诞生初期的快速变化阶段，会有一些不一致的地方。

2.3 MongoDB 介绍

2.3.1 MongoDB 简介

MongoDB 是一个以分布式为基础的文件存储数据库。由 C++ 编写。目的在为 WEB 应用程序提供可以扩展的高性能的数据存储的解决方案。

MongoDB 属于一个介于关系型数据库和非关系型数据库之间的数据库，在非关系数据库中属于功能最丰富，最相似于关系数据库的^[40]。它支持的数据结构非常的松散，类似于 json 的 bson 格式，所以可存储相对比较复杂的数据类型。MongoDB 最大的特点是它支持查询语言很强大，语法有些类似面向对象的查询语言，基本上可以实现相似关系型数据库单表查询的很大一部分功能，还可以对数据建立索引^[41]。

2.3.2 MongoDB 优点

与关系型数据库相比较，MongoDB 的优点有：

1、较弱的一致性，因此更能够保证用户访问速度：

举例说明，在传统关系型数据库中，一个 COUNT 的类型操作会锁住数据集，只有这样才可保证得到当前情况下精确值。然而在某些情况中，比如通过 ATM 查询账户信息的时候非常重要，但是对于 Wordnik 来讲，数据是需要不断增长和更新的，这类精确的保证可以说基本上没有任何意义，反而会产生非常大的延迟。用户需要的是一个相对大约的数字以及更加快速的处理速度能力^[42]。

但在某些情况下 MongoDB 会锁定数据库。假如此时发出数百个请求，它们会堆积起来，造成很多问题。开发者使用下面的优化方式可以避免锁住，在每次更

新之前，开发者先查询记录。查询操作会把对象放入内存，这样更新就会尽可能的快速。在主从部署方案中，在从节点下可使用 `-pretouch` 参数运行，这样可以得到相同的结果。在使用多个 `mongod` 进程情况下，开发者会根据访问的模式把数据库拆分成几个进程^[43]。

2、能够更加便捷的获取数据，得益于文档结构的存储方式。

对一个层级式数据结构来讲，如果要把这样的数据以扁平式来使用，表状的结构用来保存数据，查询还是获取数据时这无疑是十分困难的。

3、能够支持较大容量的存储，内置 GridFS 系统。

GridFS 是一个很出色的分布式的文件系统，可支持海量的数据存储。MongoDB 内置 GridFS，满足了对大数据集的快速查询。

4、内置的 Sharding。

提供了 Range 的 AutoSharding 机制，一个 collection 需要按照记录的范围，分为若干个段，切分于不同的 shard 上。

Shards 可以与复制相结合，与 Replicaset 配合能够实现 Sharding fail-over，不同 Shard 之间可以做负载均衡。查询对于客户端是透明的，客户端可以执行查询、MapReduce、统计等操作，这些都会被 MongoDB 的自动路由加到后端的数据节点。这能够让我们更加关注于自己的业务，适当的时候可以做到无障碍升级。MongoDB 中的 Sharding 设计能力可以最大的支持量约 20petabytes，足够支撑大部分一般应用^[44]。

这能够保证 MongoDB 运行在相对便宜的 PC 服务器集群上。PC 集群在扩充起来可以说非常的方便且成本非常低，有效的避免了 sharding 操作的成本和复杂性。

5、第三方支持很丰富。

目前网络中的很多 NoSQL 的开源数据库很大一部分完全属于社区型的，没有相关的支持，给开发者带来很大的风险。然而开源文档的数据库 MongoDB，有商业公司 10gen 能够为其提供商业支持和培训。

而且现在 MongoDB 的社区非常的活跃，许多开发框架都可以迅速的提供对

MongoDB 的支持。而且很多知名的网站和大公司也在生产环境中使用 MongoDB，非常多的创新企业转向以 MongoDB 作为和 RoR，Django 来搭配使用的技术方案。

6、性能的优越性

在使用情况下，千万级的文档对象，接近 10G 数据，对于有索引的 ID 查询效率相对来说并不差于关系型数据库，然而对于非索引字段的查询，它是远远高于关系型的数据库。mysql 实际情况下无法胜任较大数据量下的任意字段查询，但是 mongodb 的查询性能则可表现优异^[45]。

2.3.3 MongoDB 的缺点

MongoDB 与关系型数据库相比，其缺点：

1、mongodb 对事务操作不支持，所以在事务相对要求严格的系统下，不可以使用 MongoDB 作为数据库。

2、mongodb 占用的空间偏大。主要体现以下方面

(1) 空间预分配能力：为了避免产生过多的磁盘碎片，mongodb 在每次空间不足的时候，都需要申请生成一大部分的硬盘空间，而且相对申请的容量以 64M、128M、256M 的指数增长，一直到 2GB 的单个文件为最大。然而随着数据量的增多，使用者可以在它的数据目录中看到那些生成容量不断递增的文件^[46]。

(2) 字段名所需要占用的空间：为了能够以便于查询每个记录中的结构信息，mongodb 需要将每个字段中的 key-value 以 BSON 的数据形式存储，如果 value 的域相对于 key 的域比较后并不大，当中一种可以减少空间占用的方法就是将字段名尽量缩短一些，这样占用的空间就可以就小了，但这就要求在空间占用和易读性上作为权衡了。

(3) 删除记录不能释放存储空间：为了尽可能的避免删除记录后的数据大规模挪动，不删除原记录的空间，只标记出已删除，这样是为了以后还可以重复利用。

(4) 能够定期运行 db.repairDatabase()来整理数据记录，但这个过程相对比较缓慢

3、MongoDB 没有其他数据库像 MySQL 那些比较成熟的维护工具，这点对于运维和开发都是需要值得注意的地方。

2.4 MVVM 架构介绍

MVVM 即为 Model-View-ViewModel。其中以微软的 WPF 推动了新的技术体验，比如 3D、Silverlight、视频、音频、动画等等，这导致了软件在 UI 层能够更加可定制化、细节化。同时，在技术方面，WPF 也推动了像 ControlTemplate、Binding、Routed Events、DataTemplate、Dependency Property、Command 等新的特性。MVVM 框架的由来则是 MVP 即为 Model-View-Presenter 模式与 WPF 相结合的应用方式，以此发展演变过来的一种新的架构框架。它在原有 MVP 框架的基础上能够把 WPF 的新特性相结合，以此来应对客户对需求的日益复杂变化性^[47]。

MVVM 模式的优点：MVVM 模式是 MVC 模式的改进优化结构，因此主要目标还是一样提供域逻辑和表示层之间的明确分工 MVVM 模式的优缺点如下：

最关键的好处是让您从具有获得视图和模型之外实现分离和效率之间的真正分离。这意味着什么实质是，当你的模式需要改变，它可以很容易地不认为需要以反之亦然改变。有迹象表明，流出应用 MVVM 这是如下三个重要的关键的东西。

1、可维护性

不同类型的代码完全分离应该更容易进入一个或多个的更精细，重点部位，并进行更改，无需担心。这意味着可以保持灵活，并保持快速移动到新版本。

2、可测性

随着 MVVM 每一段代码是更精细，如果它是正确执行的内部和外部的依赖性是从你想测试的核心逻辑部分的代码独立的部分。这使得它容易得多写对一个核心逻辑单元测试。确保书面和当事情在维护更改继续工作，即使它的作品的权利。

3、可扩展性

它有时与可维护性重叠，因为完全分离边界和代码更精细件。它还能够更换或添加新的代码枚做类似的事情到架构中的适当位置的能力。

MVVM 模式的明显的目的是减少代码背后的商业逻辑的量观的抽象。然而，

以下是一些其他固体优势 -

该视图模型更容易比代码隐藏或事件驱动的代码单元测试。可以在不干扰 UI 自动化，互动化测试，表示层和逻辑松耦合。

MVVM 缺点：

有些人认为简单的用户界面，MVVM 可能是矫枉过正。同样，在更大的情况下，它可能很难设计视图模型。当我们有复杂的数据绑定调试会有点困难^[48]。

2.5 相关工具介绍

2.5.1 ExpressJS 介绍

ExpressJS 是基于 Node.js 的一个灵活的、极简的 web 应用开发框架，它提供了一系列非常强大的特性，帮助开发者创建各类 Web 和移动设备应用程序。丰富的 HTTP 快捷方法和任意排列组合的 Connect 中间件，让你创建健壮、友好的 API 变得既快速又简单。ExpressJS 不只是在 NodeJS 已有的特性进行的二次的抽象扩展，开发者只需要在它的基础之上扩展了 相关 Web 应用所需要的基础功能^[49]。

ExpressJS 可以设置中间件来响应 HTTP 请求、定义了路由表用于执行不同的 HTTP 请求动作、可以通过向模板传递参数来动态渲染 HTML 页面。

2.5.2 Jade 介绍

Jade 是一个高性能的模板引擎，它深受 Haml 影响，它是用 JavaScript 实现的，并且可以供 Node 使用。特性：客户端支持，代码高可读，灵活的缩进，块展开，Mixins，静态包含，属性改写，安全，默认代码是转义的，运行时和编译时上下文错误报告，命令行下编译 jade 模板，HTML5 模式（使用 !!! 5 文档类型），在内存中缓存(可选)，合并动态和静态标签类，可以通过 filters 修改树，模板继承，原生支持 Express JS，通过 each 枚举对象、数组甚至是不能枚举的对象，块注释，没有前缀的标签，AST Filters，过滤器^[50]。

2.5.3 Sass 介绍

Sass 即为 Syntactically Awesome Stylesheet Sass，由开发者 Hampton Catlin 开发的可扩展 CSS。Sass 能够简化 CSS 的工作流，而且可以将你的 CSS 维护和扩展工作变的非常容易。比如，客户需求的多次频繁变更，开发者则必须不断的查找、替换来修改满足客户的像素数据^[51]。

Sass 引入一些相对比较新的概念，如变量、选择器继承、嵌套和混合。Sass 跟 CSS 比较相像，区别在于它没有大括号和分号。

2.5.4 YSlow 介绍

YSlow 是雅虎基于网站优化规则推出的工具，帮助你分析并优化网站性能。雅虎网站优化规则在十几个方面给你的网站提出优化建议，包括尽可能的减少 HTTP 的请求数、使用 Gzip 压缩、将 CSS 样式放在页面的上方、将脚本移动到底部、减少 DNS 查询等十几条规则，YSlow 会根据这些规则分析你的网站，并给出评级。

第 3 章 系统分析

3.1 系统可行性分析

对于可行性的分析，就是针对项目的条件、内容、要求；像需求供应，所需的设备规模等其他方面进行的相应调查与分析，需要预测到在项目完成后能够获得的效益，提出该项目后判断是否需要投入相应的资金来运营的系统分析法的一种。可行性分析是为了能够保证投资效果真实有效，还有在项目实施前需要从技术、经济、管理等等方面对项目进行全方面的分析。其最基本要求就是包含从经济角度出发进而全方面的分析研究，对几个待确定的方案论证中选择、针对项目运营后带来的经济效益进行相应的预测评估。其目标是最大化的利用可用资源，实现预期带来的社会效益与经济效益。

本章主要对于技术可行性的研究，通过以项目实践的技术角度来讲，合理设计采用的技术方案并进行评估和对比。NodeJS 框架 Expressjs 能够做到简单的实现开发者需要的多种功能，创建 Server 服务器，设置路由，配置相关的参数，中间件，渲染，错误处理，查找 View，参数预处理，模板引擎的相协作，Session 以及 Partial 支持非常的全面，在终端中执行命令 `npm install mongoose`，NodeJS 会通过数据源安装 Mongoose 模块，Mongoose 很便于管理并操作 MongoDB。MongoDB 存储数据的方式比较适合用来存储博客的数据。所以，本系统的数据库方面的开发是可以实现的。

3.2 功能需求分析

在对技术的可行性分析里已经很简明地分析了和研究本系统的基本需求，同时也论证了对于开发本博客系统的可行性，但是能够为了非常明确本博客系统设计的目标，能够了解每个用户的需求，进而可以对用户的需求进行调研，以便能够进一步的确定所要实现设计的子系统的功能。系统的功能结构图如图 3.1 所示。

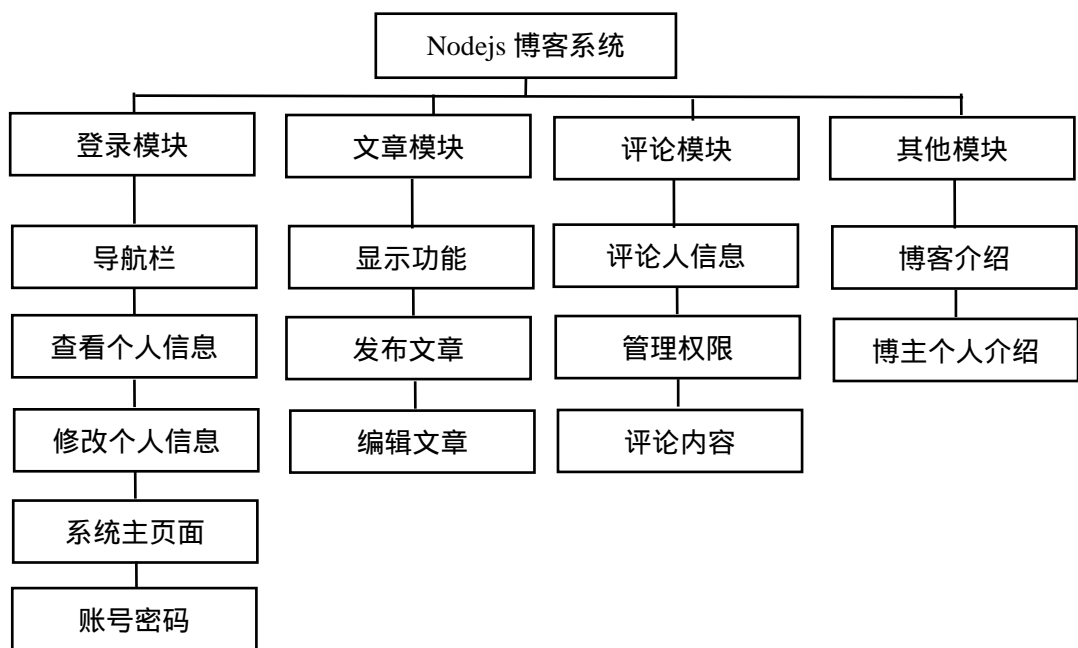


图 3.1 系统功能结构图

Fig.3.1 System Function Structure diagram

3.2.1 登录模块需求分析

本博客系统共分两个用户权限，分别是访客用户、博主。

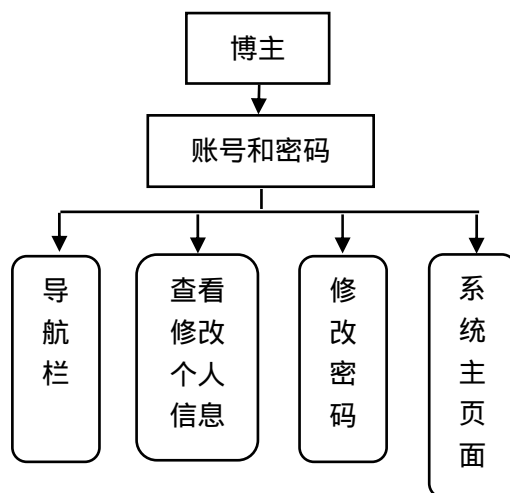


图 3.2 博主登录模块图

Fig.3.2 Blogger login module diagram

博主能够拥有博客系统最高权限，访客用户不能够对博客中的文章进行修改或删除，所以本系统中加入博主相关的登录系统，博主按照要求输入自己拥有的账户，登录之后对博客文章等内容进行修改或删除。这样就可以达到访客只在允许范围内进行操作，不能对博客内容或文章进行修改，保证博客数据的安全。图 3.2 为博主登录模块图。

博主用户登录、注册系统详细设计的流程图如图 3.3 所示。

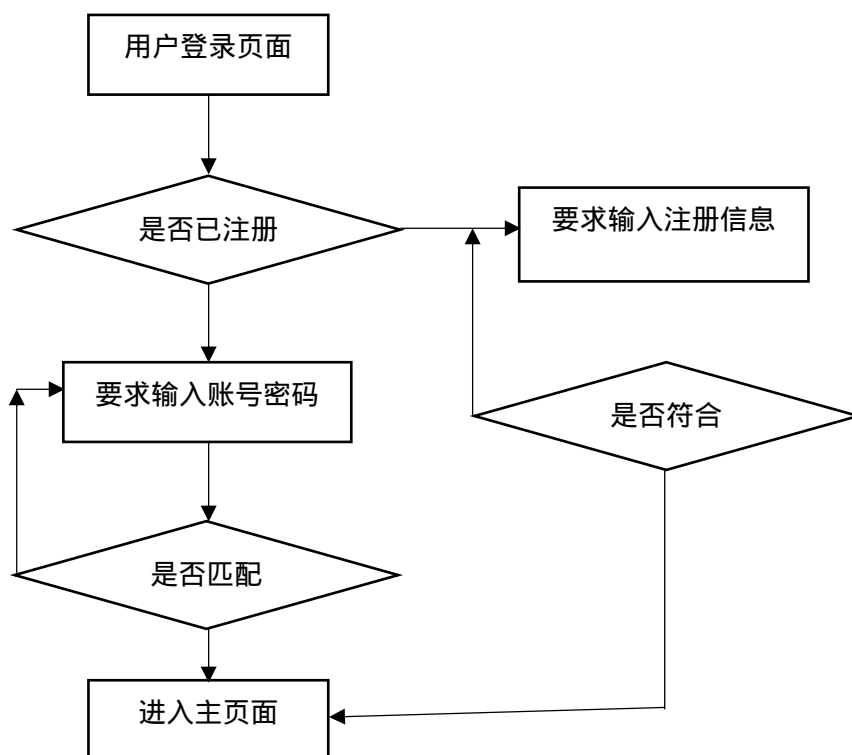


图 3.3 用户注册、登陆模块流程图

Fig.3.3 User registration, login module flow chart

用户登录填入用户账号登录网站时，操作登录主页面后，假如此用户已经是注册的用户只需要输入用户名、密码提交进入主页面，如多提交的表单数据与数据库中的数据一致则用户登录成功，用户进入个人主页面。如果提交的用户数据与数据库中的数据不一致时则提示登录失败，当前的页面跳转回登录页面，用户就需要重新登录，如果当前用户没有注册过本系统，就需要注册新的账号，用户成功注册后即可跳转到注册页面，用户在注册页中需要填写用户名和密码，完成后点击确定，提交，如果提交成功即用户输入信息符合要求，则提示用户注册

成功，用户则登录页面，操作同上。如果用户输入信息不符合要求则会跳转到注册页面，用户需要重新填写注册信息。

3.2.2 文章管理需求分析

本系统的主要功能体现就是文章，因此开发者要为博主用户提供必要的文章增删改查功能(CRUD)，如图 3.4 所示。

- (1)文章的显示功能：显示文章的标题、摘要；列表按照索引列出文章；
- (2)发布新文章功能：创建新文章，而且文章内容能够在数据库中持久存储；
- (3)编辑修改文章功能：可以对现有文章进行修改，标题或者内容；
- (4)删除文章功能：对现有文章进行删除；

此外，每篇文章都有独立的评论内容,博主用户可以对评论进行回复或删除；而访客仅可对文章进行评论,不可删除他人评论。

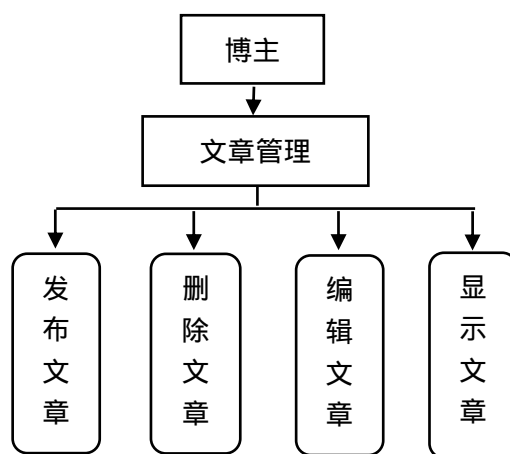


图 3.4 文章管理的实例图

Fig.3.4 Article management example diagram

3.2.3 评论管理需求分析

评论的用户其用户名会在这篇文章的页面底部显示，以及会显示出评论的时间以及评论的详细内容。评论的页面中，能够显示评论用户的名字以及留言时间、评论的具体内容。在本博客系统设计中相应的每个评论的记录时间用相应的时间

间隔分开，这样能够方便对评论进行管理，对评论有删除权限下的用户可操作，普通的用户也只能查看和留下留言内容并没有删除留言的内容。如图 3.5 所示。

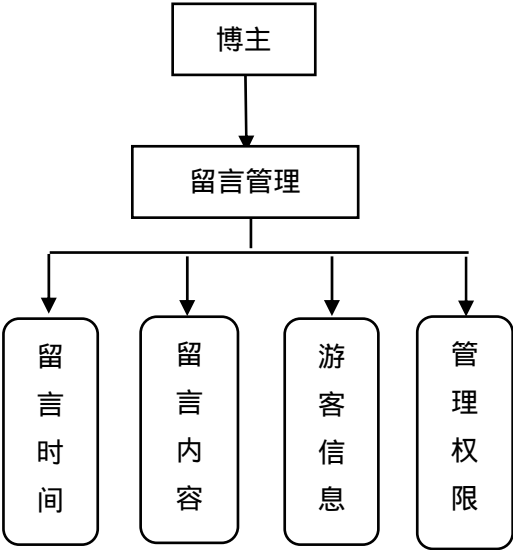


图 3.5 留言管理模块实例图

Fig.3.5 Example of message management module diagram

3.2.4 页面需求分析

大部分的博客中都有关于自己信息的的关于页面，用于对博客的相关介绍以及博主自身的的信息介绍。在本博客系统中，关于页面相当于一篇文章，通过文章的形式存储在数据库中，博主用户以对自身的关于页面进行相关的修改编辑等一些操作。相对其他文章不同的是，关于页面有它单独的个性化的页面模板，开发者可在模板中的样式或是结构上进行修改或调整。博客的页面模块图如图 3.6 所示。

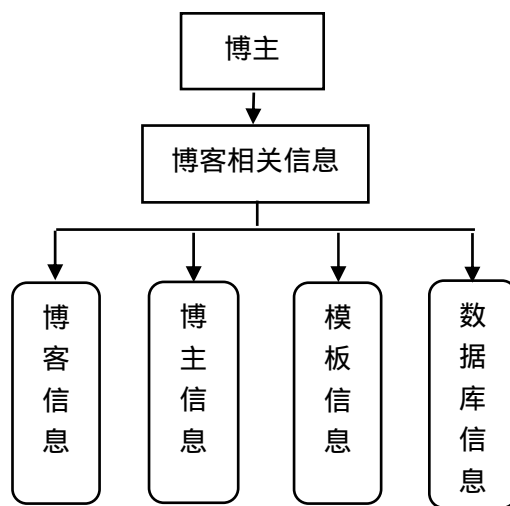


图 3.6 博客相关页面实例图

Fig.3.6 Example of message management module diagram

3.3 非功能性需求分析

3.3.1 性能需求分析

系统性能需求能够满足 15000 的用户量，同时并发数在线最高为 380 人，响应时间平均不可以超过 3 秒，在访问最高峰时，响应的时间最长不可以超过 7 秒。在不超过功能保护周期情况下，系统可以稳定的运行时间 2 至 3 年，同时系统也需要跟随服务器的过保周期对服务的迁移。针对系统并发量的需求，系统通过采用集群来解决对外的进行服务提供，一方面能够保证服务的处理能力，另一方面能够保证系统服务更新和升级，能够进行流量切换，确保的服务的不间断，降低系统发生故障概率。

对于系统数据存储一定要保障安全，设计数据库存储数据为主从关系数据库，主数据库主要用于博客文件的数据存储,从数据库主要用于对主数据库数据的同步备份;针对索引模块的数据,使用文件的格式来存储,索引检索服务器同也要配置主从服务器，一方面方便程序更新升级，能够有效的减小故障的风险,另一方面保障

数据的安全性且提供数据备份。

3.3.2 安全性需求分析

在安全方面，必须系统的每个方面都需要进行监测，还要对生产环境中的服务采取实时监测，定时对系统的各项服务进行功能扫描和检测，只要一出现问题，立即通过短信或邮件向开发者发送消息，能够及时的对服务进行问题排查，解决问题，确保服务器宕机时间不可以超过 15 分钟，大大提高系统的稳定性。在系统的生产环境的服务更新时，选择用户使用频率较低的时间段升级，比如凌晨 2-3 点左右，从而减小对用户系统的影响。

在数据存储中对用户的信息加密处理，在数据库中的数据删除必须以逻辑删除来取代物理删除，防范 SQL 注入攻击。用户访问系统要确保权限，对系统的所有错误都必须采取日志记录，页面上的错误也都必须在页面上能够显示出来，供用户浏览。

3.4 数据库需求分析

数据库数据存储的数据分为两部分，分别是文章、评论。在设计数据库之前可以简单参照一下 WordPress 的数据库表图来进行进一步的分析，如图 3.7 所示，我们可以清楚的看到这个是相对来说比较简单的。

由于内容过多，我们只需要了解其文章，账户和评论三大类。综上，系统所需要的创建的数据表主要有三类：文章，评论和账户。

文章需要的字段有：id、标题、作者、时间、内容、摘要；评论需要的字段有：id、作者、时间、邮件、内容；账号需要的字段有：用户名和密码，其中密码需要使用 MD5 加密，绝对不可以使用明文存储密码。

实际中，由于评论是需要和文章相互关联的，然而 MongoDB 提供的 path 方法可以选取 Document 中需要的关联数据，因此，我们可以将评论放到文章中。

因此，最后调整后文章的字段为 id、标题、作者、时间、摘要、评论。

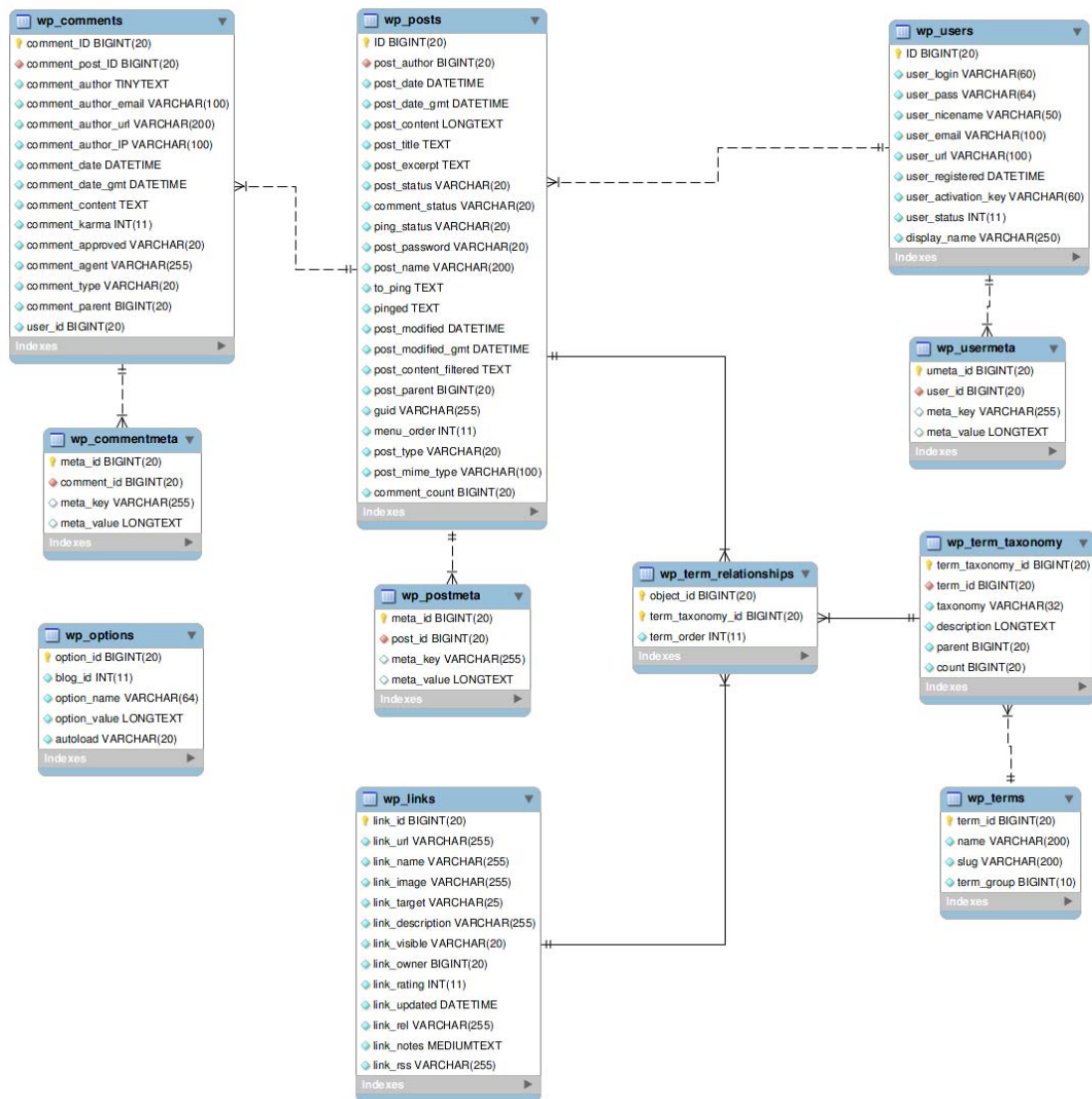


图 3.7 Wordpress 相关数据表

Fig.3.7 Related database tables of Wordpress

第 4 章 系统设计

4.1 数据库概述

数据库是按照数据结构来组织、存储和管理数据的建立在计算机存储设备上的仓库。

通俗的说，数据库就像是电子版的文件夹，用户可以对文件夹中的文件管理，比如删除、添加、修改、查看等操作。

日常工作的经济管理中，通常需要将相关的数据文件存放到这样的仓库中，并可以对这些文件进行相关的处理。

4.2 数据库的选择

本系统采用非关系型数据库—MongoDB，在 Nodejs 中需要使用特定的模块来与 MongoDB 连接即为 Mongoose 模块，MongoDB 属于非关系型数据库，面向文件式，因此十分适合用于博客文章的存储，而且开源，免费，非常适合博客系统。

4.3 数据库概念设计

数据库是整个系统的主要内容，基石，数据库设计的好坏直接影响整个系统的优劣，下面对数据库的设计进行详细的分析。

博客的数据库存储主要包含：文章数据、用户数据、用户组数据、推送数据。需要将这些信息抽象化，转化成数据项。

- 1、用户组信息，包含用户组 id，用户组名称，用户组权限。
- 2、用户信息，包含用户 id，用户名，用户密码，用户组，邮件，注册时间，推送状态。
- 3、文章记录，包含文章 id，标题，日期，描述，发送的对象。
- 4、推送信息，包括编号 id，推送 URI，用户 id，推送的对象。

下面使用 E-R 图来描述信息推送的数据模型，该信息推送系统的 E-R 图清晰描述了系统中涉及的实体之间的关系，具体关系结构如图 4.1 所示。

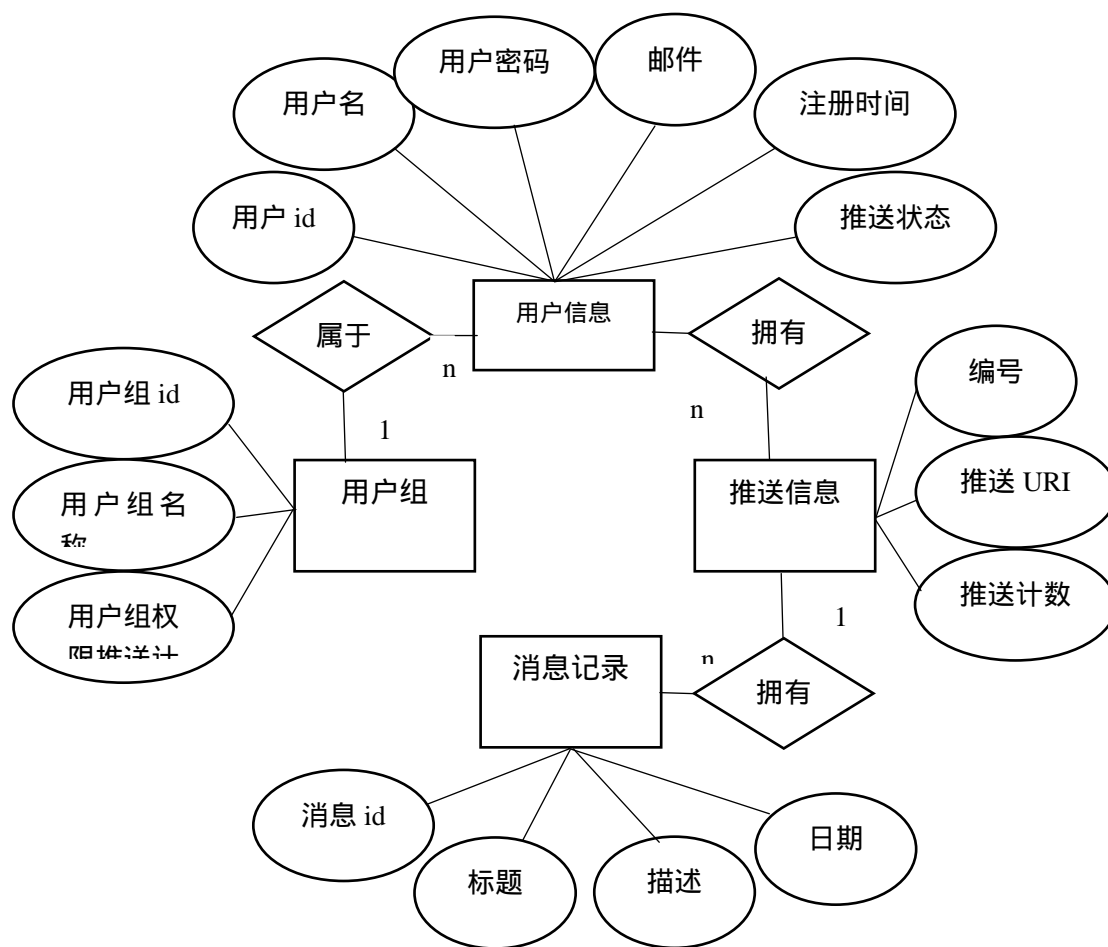


图 4.1 系统 E-R 图

Fig.4.1 E-R Diagram of System

数据库的最后的设计中需要在给定的逻辑结构放到具体的环境中，逻辑数据就要选取一个相对具体的工作环境，但是由于硬件和特定的 DBMS 决定物理结构，因此需要充分理解 RDBMS 的结构，特征，以及存取方式会影响开发人员的开发，数据库的物理设计通常分为两部分，第一是要定义空间和时间的效率，第二是要确定数据库的物理结构。

4.4 数据库逻辑设计

4.4.1 用户登录表

用户信息表主要用来存储本系统中用户的相关信息，其中部分信息是需要在用户注册时插入到数据表中，用户数据表主要用以管理用户、管理员的管理，其

中登录的验证有需要此表。表中具体内容如表 4.1 所示。

表 4.1 用户信息表

Tab.4.1 User Information Table

字段名称	字段名称	类型	是否索引	备注
ID	ID	ObjectID	是	
账号名	Account	String	是	Not Null
密码	Password	String	否	MD5 加密
文章	Posts	Array	否	
作者	Author	String	否	

4.4.2 用户组登录表

用户组信息表主要用来存储系统中用户组的权限信息，其中的数据是需要系统在最初初始化时添加的，该表主要是对系统中用户的访问权限进行管理，还包括用户的分组管理，具体内容如表 4.2 所示。

表 4.2 用户组信息表

Tab.4.2 User Group Information Table

字段名称	字段名称	类型	是否索引	备注
用户组 ID	GroupID	Char(30)	是	Not Null
用户组名称	GroupName	Varchar(35)	是	Not Null
用户组权限	GroupPower	Varchar(35)	是	Not Null

4.4.3 文章信息表

文章信息表主要用来存储博客系统中的文章以及对应的作者信息，具体信息如表 4.3 所示。

表 4.3 文章信息表

Tab.4.3 Article Information Table

字段名称	字段名称	类型	是否索引	备注
ID	ID	ObjectID	是	
时间	Date	Date	是	
标题	Title	String	否	Not Null
摘要	Except	String	否	Not Null
内容	Content	String	否	Not Null
评论	Comments	Array	否	

4.4.4 评论信息表

评论信息表组要用来存储博客系统中对应文章下的评论信息，具体内容如表 4.4 所示。

表 4.4 评论信息表
Tab.4.4 Comments Information Table

字段名称	字段名称	类型	是否索引	备注
ID	ID	ObjectID	是	
时间	Date	Date	是	
作者	Author	String	否	Not Null
Email	Email	String	否	Not Null
内容	Content	String	否	Not Null

4.4.5 其他相关信息表

其他相关信息表主要用来存储博客系统中管理员对相关页面的编辑及修改。具体信息如表 4.5 所示。

表 4.5 其他相关信息表
Tab.4.5 Other Related Information Table

字段名称	字段名称	类型	是否索引	备注
ID	ID	ObjectID	是	
时间	Date	Date	是	
标题	Title	String	否	Not Null
摘要	Except	String	否	Not Null
内容	Content	String	否	Not Null
分类	Class	Array	否	

第 5 章 系统实现

5.1 系统架构设计

本系统采用的是 MVVM 的架构方式。用户界面，数据模型，业务逻辑分别分布在三个层次，而且各个层次之间耦合性较低，相对来说比较容易维护。

该架构模型属于 MVVM 设计模式，其中用户页面是 View，数据模型 ViewModel，业务逻辑处理 View Controller。

Model-View-ViewModel 的缩写为 MVVM，即模型-视图-视图模型，这三部分组成了 MVVM 应用。View 持有 ViewModel：在 View 中通过绑定的方式关联 UI 元素和 ViewModel 的属性，UI 元素的属性变化和 ViewMode 的数据属性进行双向影响。ViewModel 持有 Model：Model 为 ViewModel 提供数据支持，同时 ViewModel 的数据属性变化也会影响 Model 上，用于数据提交等。Model 和 ViewModel 的功能区别在于，ViewModel 是 View 提供数据支持的，其结构和 View 保持一致。Model 来源于业务处理，保持业务数据的基本关联和完整性。当 Model 层结束查询或者运算时将结果更新到 ViewModel 层，ViewModel 层因为存在和 View 的绑定，使得 UI 层得到通知更新，体现了数据驱动界面的思想^[52]。

MVVM 架构模式理论上跟 MVC 架构模式相似，目的主要是分离了视图和模型有三大优点：低耦合：通过 ViewMode 隔离了 View 和 Model，使得 View 和 Model 彼此独立，一方的变化和修改不会影响到另外一方；可重用性：一个 Model 和 ViewModel 可以服务于很多 View，独立开发：让 UI 开发工作和业务处理工作分开，适合团队工作；便于测试，开发者可对 ViewModel 来写出测试。

MVVM 的调用关系和 MVP 一样。但是，在 ViewModel 当中会有一个叫 Binder，或者是 Data-binding engine 的东西。以前全部由 Presenter 负责的 View 和 Model 之间数据同步操作交由给 Binder 处理。你只需要在 View 的模版语法当中，指令式地声明 View 上的显示的内容是和 Model 的哪一块数据绑定的。当 ViewModel 对进行 Model 更新的时候，Binder 会自动把数据更新到 View 上去，当用户对 View 进行操作（例如表单输入），Binder 也会自动把数据更新到 Model 上去。这种方式称

为：Two-way data-binding，双向数据绑定。可以简单而不恰当地理解为一个模版引擎，但是会根据数据变更实时渲染^[53]。

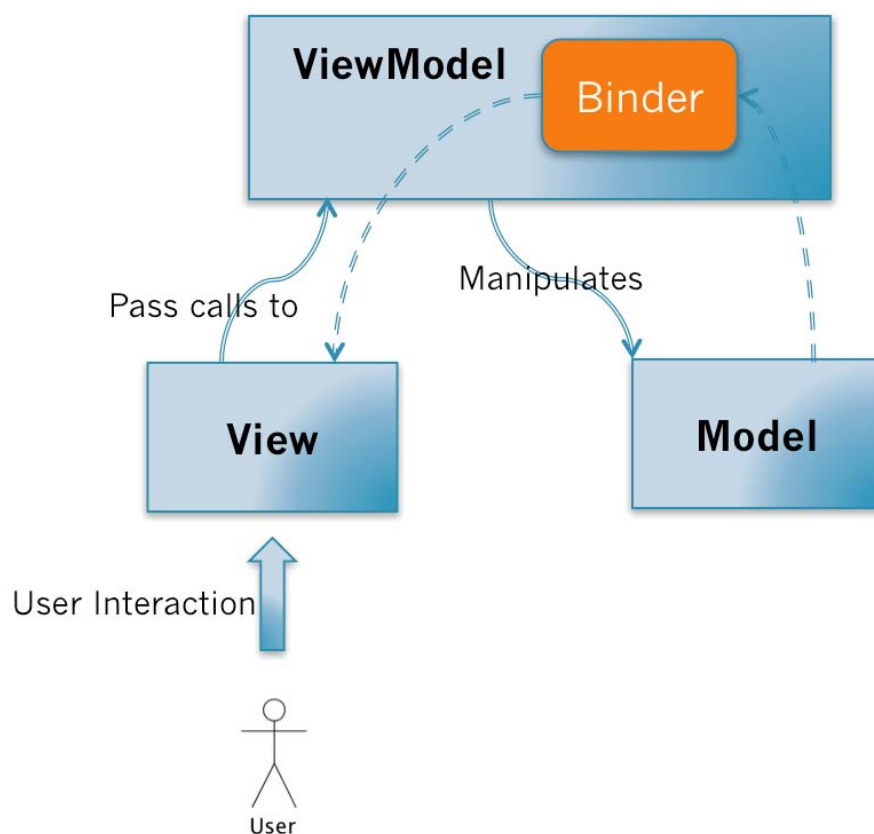


图 5.1 MVVM 调用关系

Fig.5.1 Call Relationship of MVVM

5.1.1 表现层

表现层为用户在浏览器中所浏览的前端页面，主要的页面为 layout.jade:

```
doctype
```

```
html
```

```
head
```

```
meta(charset="utf-8")
```

```
title #{title}
```

```

    include ./includes/head

body

    include ./includes/header

    block content

    include ./includes/footer

```

其中 head , header , footer 内容分别为：

head.jade

```

meta(name="viewport", content="width=device-width, initial-scale=1")
link(href="/libs/bootstrap/dist/css/bootstrap.min.css", rel="stylesheet")
script(src="/libs/jquery/dist/jquery.min.js")
script(src="/libs/bootstrap/dist/js/bootstrap.min.js")
script(src="/build/index.css")

```

header.jade

```

.container

  .row

    .page-header.clearfix

      h1= title

    .col-md-4

      small the kingzez's blog

    .col-md-8

      form(method='GET', action='/results')

        .input-group.col-sm-4.pull-right

          input.form-control(type='text', name='q')

          span.input-group-btn

            button.btn.btn-default(type='submit') search

.navbar.navbar-default.navbar-fixed-bottom

  .container

```

```

    .navbar-header
      a.navbar-brand(href="/") Home
    ul
      li.nav:a(href="/archives ")Archives
      li.nav:a(href="/categories ")Categories
      li.nav:a(href="/tags")Tags
      li.nav:a(href="/about") About
  footer.jade
    container
      .row
        p.lead.text-center.text-danger

```

以上三个页面为抽象出可复用的部分代码，能够减小代码的冗余。render 函数在 block content 内添加内容模板，其中相关的模板有：

- home.jade 首页主界面
- post.jade 文章列表页面
- detail.jade 文章详情页面
- new.jade 文章创建页面
- edit.jade 文章编辑修改页面

在 jade 模板文件中通过 render.js 来使用 JavaScript 变量，比如用户请求编辑修改文章页面时，文章内容需要默认为存入原本文章的内容，在 render.js 文件中，需要使用：

```

response.render('edit', {
  locals: {
    title: article.title,
    post: post
  }
});

```

进而来达到渲染对应的页面。从以上程序代码语句中可以看出，通过对象参数的传递给 edit.jade 文件需要用到的变量，因此 jade 就可以针对变量的内容对相关部分进行替代。

5.1.2 业务逻辑层

业务逻辑层是专门处理软件业务需求的一层，处于数据库之上，服务层之下，完成一些列对 Domain Object 的 CRUD，作为一组微服务提供给服务层来组织在暴露给表现层，如库存检查，用法合法性检查，订单创建^[54]。业务逻辑中主要核心的概念就是领域对象模型，业务逻辑，领域的实体，业务流程。

博客中的文章信息是随时变化更迭的，在业务逻辑层的代码实现过程中需要不断更改，要做到需要更改较大需求的情况下，能够做到最小的修改量，本系统将主要的业务逻辑层的代码放在 route.js 文件中，这样可以做到在更改时不影响到其他业务逻辑。

业务逻辑生效时即为客户端向服务端发送请求时，因此在 route.js 文件中需要定义出博客系统的路由，而且一些相关的配置项都需要在文件中定义，系统的启动文件以及 Express 的服务器配置文件具体如下：

```
var express = require('express')
var path = require('path')
var mongoose = require('mongoose')
var mongoStore = require('connect-mongo')(express)
var port = process.env.PORT || 3000
var app = express()
var fs = require('fs')
var dbUrl = 'mongodb://localhost/imooc'
mongoose.connect(dbUrl)
// models loading
var models_path = __dirname + '/app/models'
```

```

// Index
app.get('/', Index.index)

// User
app.post('/user/signup', User.signup)
app.post('/user/signin', User.signin)
app.get('/signin', User.showSignin)
app.get('/signup', User.showSignup)
app.get('/logout', User.logout)
app.get('/admin/user/list', User.signinRequired, User.adminRequired, User.list)

// article
app.get('/article/:id', article.detail)
app.get('/admin/article/new', User.signinRequired, User.adminRequired,
article.new)
app.get('/admin/article/update/:id', User.signinRequired, User.adminRequired,
article.update)
app.post('/admin/article', User.signinRequired, User.adminRequired,
article.savePoster, article.save)
app.get('/admin/article/list', User.signinRequired, User.adminRequired, article.list)
app.delete('/admin/article/list', User.signinRequired, User.adminRequired,
article.del)

```

5.1.3 数据模型层

系统中的开发必不可少的是与数据库的交互，本博客系统中的个人账户信息，所编写的文章信息，文章的评论信息等等，这些信息都需要同步存储到数据库中。数据库的增删查改的 Restful 操作主要为系统开发的核心，本系统中为了实现 NodeJS 与 MongoDB 的连接，方便其操作需要使用 Mongoose 实现数据库的操作。

首先要做的是对数据库的配置操作，以下以文章的数据模型作为实例分析：


```
var mongoose = require('mongoose')
var Schema = mongoose.Schema
var ObjectId = Schema.Types.ObjectId
```

创建 mongoose 对象，并创建相应的数据对象：

```
var ArticleSchema = new Schema({
  doctor: String,
  title: String,
  language: String,
  country: String,
  summary: String,
  flash: String,
  poster: String,
  year: Number,
  pv: {
    type: Number,
    default: 0
  },
})
```

通过以上的数据模型的数据操作，系统可以对文章的数据进行增删改查操作。

5.2 MD5 加密

用户的个人信息如密码等一些重要的信息在数据库中不可以使用明文存储，如果数据库一旦被攻击后，用户资料的泄露是非常严重的问题，因此用户的密码在数据库中需要使用 MD5 加密。MD5 加密的作用是让已知的信息用数字签名软件签订私人的密钥前加密的一种格式^[55]。

5.3 系统的依赖与配置

5.3.1 系统依赖安装

本博客系统的服务端属于跨平台型，可以在 Linux，Windows 以及 MacOS 上

运行，但是建议在 Linux 上使用，相对比较简单，友好的终端操作，首先需要安装系统的依赖 NodeJS。

Linux 系统下需要满足以下条件，才能保证编译成功：

Python，需要用到 gyp，在 shell 下执行 python，查看系统是否安装了 python，而且需要确认 Python 的版本是否满足需求最好是 2.6。

源代码编译器，正常情况下 Linux 系统下都配备了 C++编译器。如果系统中没有，可以通过发行版的软件包安装 g++，make 这类编译工具。

检查开发环境并安装，满足以上条件，通过获取 NodeJS 源码并进行环境检查：

执行命令

```
wget http://nodejs.org/dist/v4.4.0/node-v4.4.0.tar.gz
```

解压 NodeJS 源码执行以下命令

```
tar zxvf node-v4.4.0.tar.gz
```

进入到 node-v4.4.0 文件夹下执行以下命令

```
cd node-v4.4.0
```

对 NodeJS 进行配置，执行以下命令

```
./configure
```

执行命令后，configure 命令执行成功，即可执行编译的命令，执行以下命令

```
make
```

```
make install
```

NodeJS 需要通过 make 编译、make install 安装，若 make install 安装不成功，需要使用 sudo 确保此命令有执行的权限。完成以上操作，通过执行以下命令检查一下是否安装成功：node -v

执行命令后若返回 v4.4.0，则说明 NodeJS 编译成功并安装完成。

5.3.2 系统框架及优化工具

系统中需要使用的框架及工具需要时用 NodeJS 中的应用 NPM 安装，NPM 是 NodeJS 的包管理工具，只有通过 NPM 才能安装 NodeJS 开发过程中需要的框架。

通过 NPM 安装工具非常方便，在终端中执行 `npm install packagename` 即可，下面安装本系统需要的框架及工具：

```
npm install express jade sass mongoose --save
```

在结尾处加上 `save` 即可保存到 `package.json` 文件中，方便之后再服务端的安装，更换服务器之后只需要执行 `npm install` 即可全部安装所需的矿建和工具。

当所有模块全部安装成功后，在终端中执行 `node app` 即可启动服务器。客户端发送数据到服务端，服务端即可返回相应的请求数据到客户端。

5.3.3 系统实现的难点

其中的难点在于可视化的富文本编辑器以及结合 Markdown 编辑器，需要自己编写一个富文本及 Markdown 编辑器的生成程序，该生成程序需要引入相关的 js 类库，本系统需要的类库有 Twitter 的开源项目中的 Bootstrap，以及 Github 上的 Fontawesome 一个字体图标库，Jquery，具体操作如下

```
<script src="http://bootcdn.com/jquery/jquery.min.js"></script>
```

```
<script src="http://bootcdn.com/twitter-bootstrap/js/bootstrap.js"></script>
```

```
@import url('http://bootcdn.com/twitter-bootstrap/css/bootstrap.css');
```

```
@import url('http://bootcdn.com/font-awesome/css/font-awesome.min.css');
```

在文章修改，编辑区创建一个 div，用于生成可视化编辑器，

```
<div id="mk-edit"></div>
```

最后需要调用前文中编写的生成编辑器的方法：

```
$('#mk-edit').createMyEditor();
```

这样就可以达到我想要的编辑器，如图系统页面展示中图 5.5 所示。

5.4 系统页面展示

5.4.1 系统首页展示

系统首页是系统整体功能的展示页面，主要用于展示博主所写的文章，首页页面设计主要以简洁为主，顶部导航主要有主页、存档、分类、文章标签、关于；页面左侧主要分布博主个人信息，让浏览的用户一目了然，中部则是文章的列表

内容，页面右侧主要是最近的文章，如图 5.2 所示为系统首页。



图 5.2 系统首页

Fig. 5.2 Home Page of System

5.4.2 文章页面展示

文章页面为整个系统最核心的页面，用户浏览博客的目的就是要阅读文章，文章页面主要分为三大部分：文章内容，评论内容，评论输入框。具体如图 5.3 所示。

博客系统文章过多时，查找特定文章时需要对文章进行检索，搜索内容可以使文章题目中的任何一个字都可以搜得到相关文章，具体页面如图 5.4 所示。

文章的编辑修改页面采用富文本编辑器，方便修改文章内容文字的格式，具体页面如图 5.5 所示。

所有文章的列表即为文章的档案，按照时间顺序依次排列，方便查看博主的文章数，具体页面如图 5.6 所示。

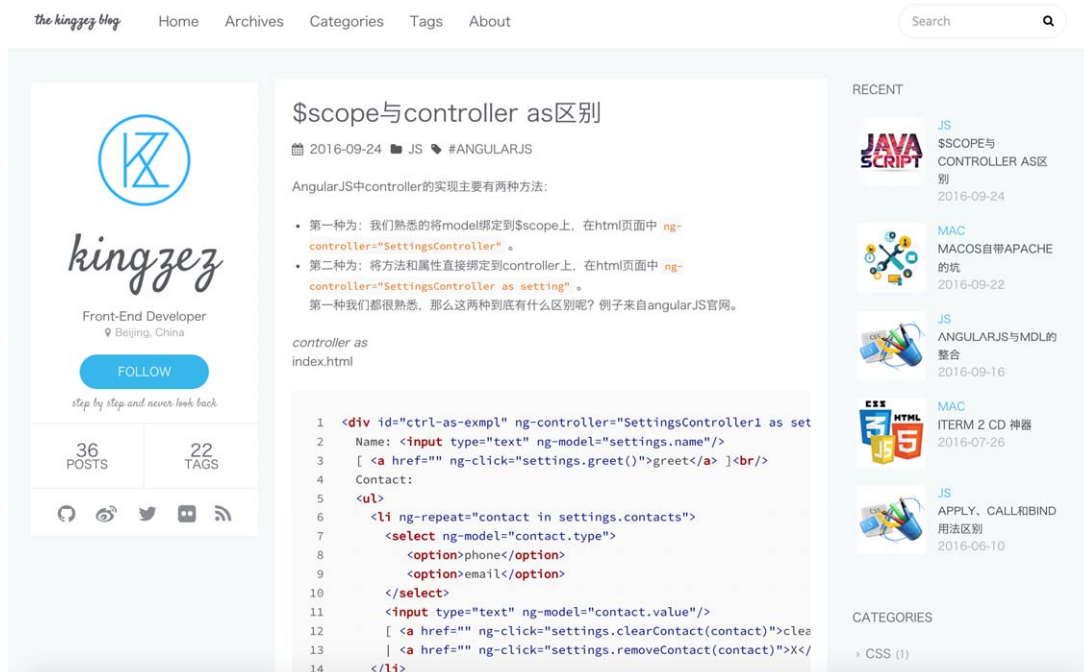


图 5.3 文章页面

Fig. 5.3 Article Page

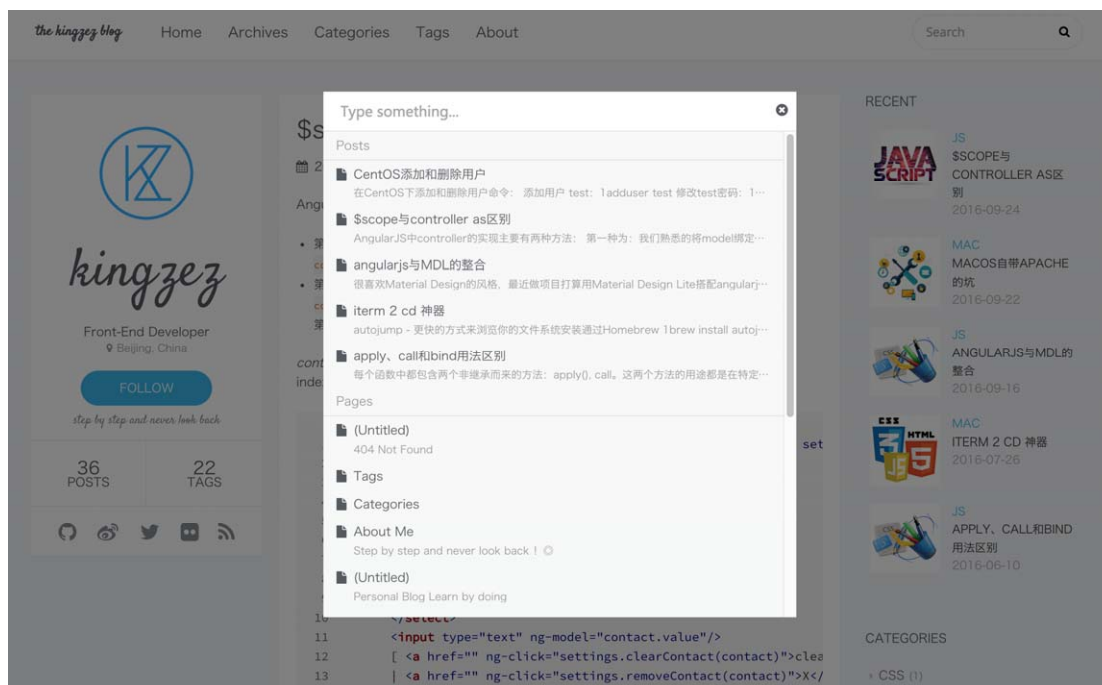


图 5.4 搜索页面

Fig. 5.4 Search Page

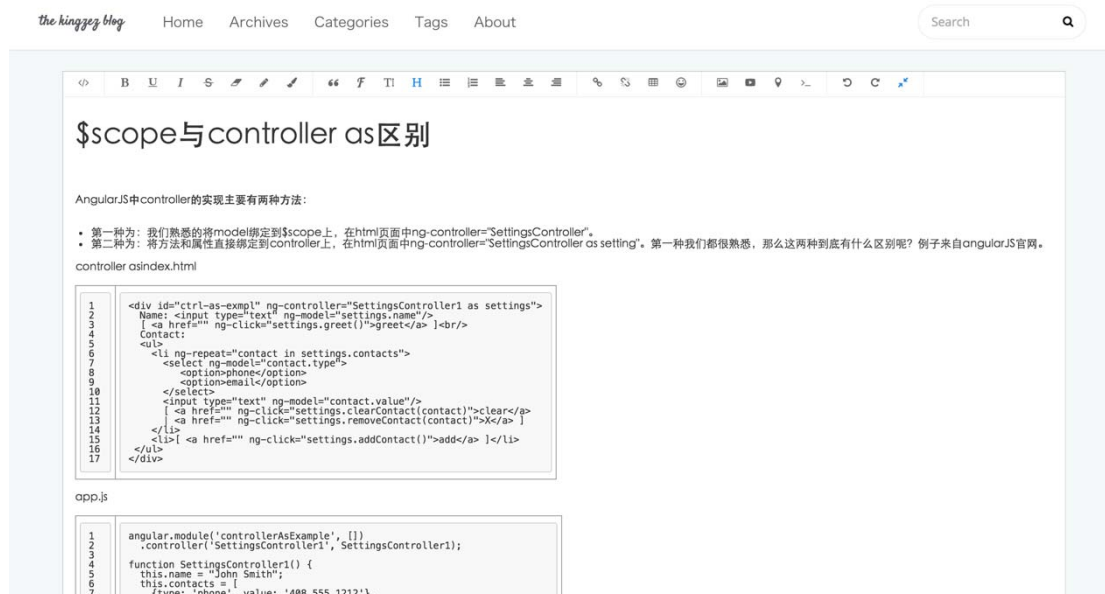


图 5.5 文章编辑页面

Fig. 5.5 Article Edit Page

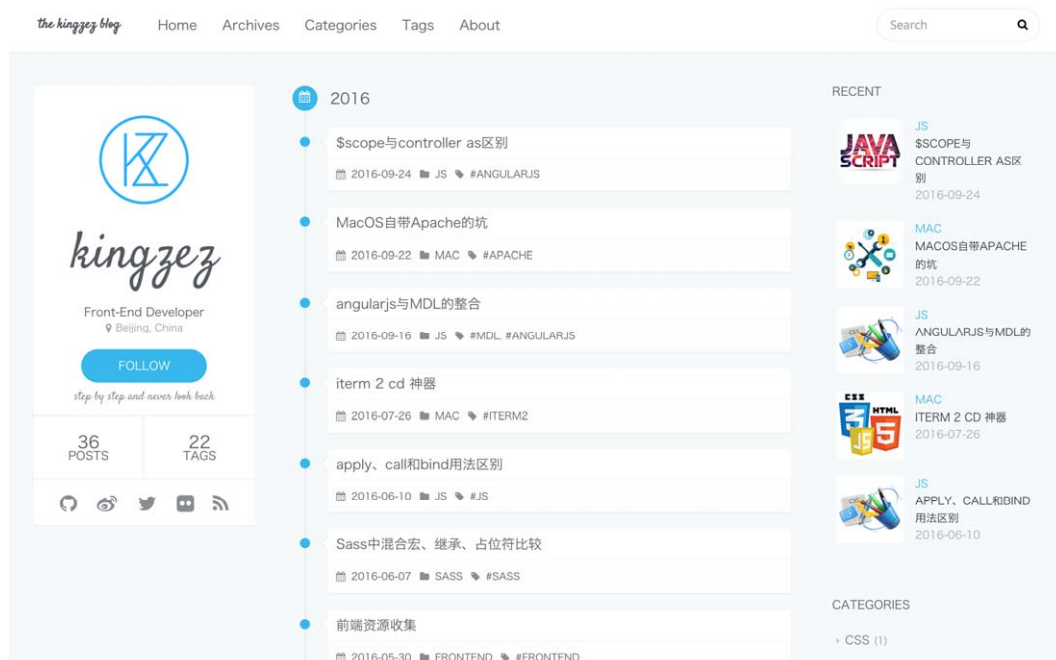


图 5.6 文章档案页面

Fig.5.6 Article Archives Page

5.4.3 评论页面展示

每篇文章下都有评论区，用户可以针对所读的文章进行相关的评论，评论页面中会显示评论者的昵称，评论时间，以及评论内容，同时通过引入分享的插件可以对该评论分享到微博等其他平台。具体页面如图 5.7 所示。



图 5.7 文章评论页面

Fig. 5.7 Article Comment Page

5.4.4 关于页面展示

关于页面是关于本博主的介绍，用户通过关于页面中的信息来了解博主的信息，博主可以对关于页面进行个人的介绍以供用户浏览了解自己，关于页面中，博主可以自由发挥来表达关于自己的介绍，能够让用户看到博主的信息后能够了解博主的详细信息，而且博主也可以自定义关于页面的网页布局，页面中的最近可以显示或隐藏，还有分类标签也可以进行显示或隐藏的设置，这样能够更加人性化的设置修改个人的关于页面，具体页面如图 5.8 所示。

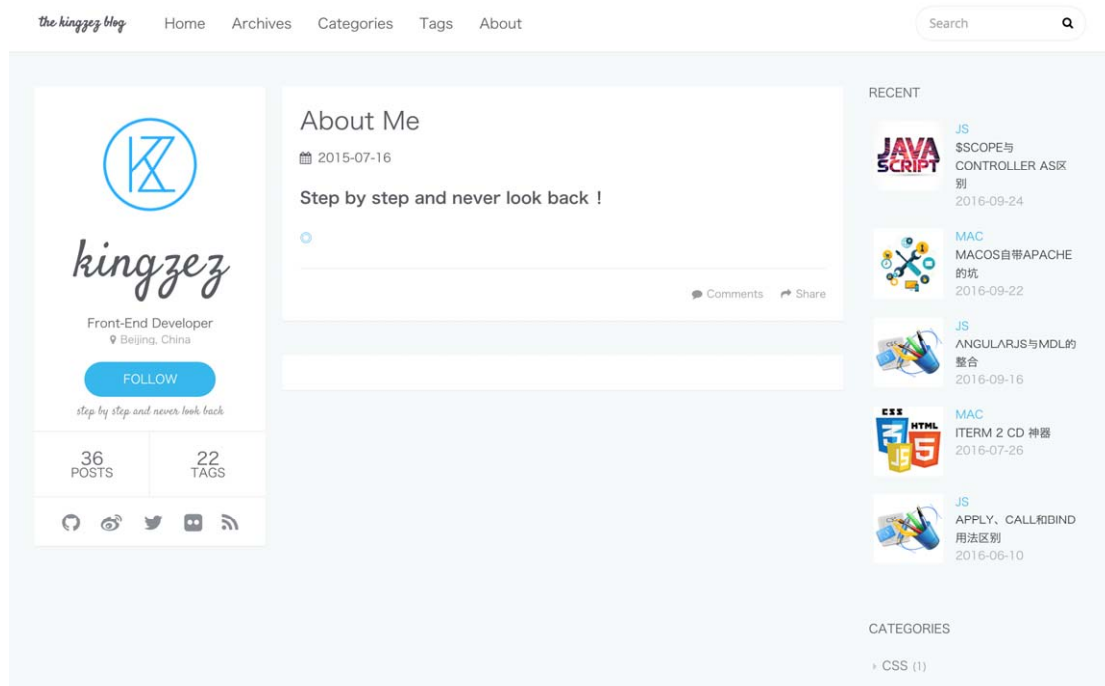


图 5.8 关于页面

Fig.5.8 About Page

第 6 章 系统测试

6.1 系统测试

6.1.1 测试目的

测试的目的在于为了能够发现系统存在的缺陷。这里提到的缺陷包括功能的错误,易用性差,性能低下等等。由此看来,测试属于破坏行为。测试目的就是为了发现软件程序中的功能错误,相对的是为了测试程序有错,而不是为了让程序没有错误。软件测试是为了将证明程序没有错误作为测试的目的是不对的,而且这样做就失去了测试的意义^[56]。软件测试一定要想办法让软件发生错误,只有发生了错误才能够发现程序达到测试的目的,否则测试就没有任何意义。

测试的过程中主要就是根据软件开发过程,根据相关的需求来使用测试该软件,了解软件的架构,根据这些内容可以写出高效的测试用例,通过执行这些测试用例,找出软件中隐藏的问题或缺陷^[57]。

由于开发这在主观上对该软件存在固有的思维局限性,从客观上来讲,目前的系统的结构以及功能都是极其复杂的,开发者在开发软件的过程中发生错误是不可避免的。如果在开发过程中能够及时的处理软件的错误,就能够就绝软件后期维护带来的一系列问题,也就是说在维护或是修改问题的过程中减少了时间与金钱的付出,以此能够很大程度的提高系统的开发效率,所以测试在软件开发过程中是必不可少的。

所以测试的最终目标就是要保证软件的质量。

6.1.2 测试方法

测试通常情况下分为白箱测试、黑箱测试^[58]。

黑箱测试,也成为黑盒测试,测试软件的功能,通过测试用例测试程序的功能。测试人员无需软件系统的开发能力以及编程语言的知识。测试者只要知道这个所测系统能够完成的功能,怎么使用该软件以及当输入与软件功能不相符的数

据或操作来检查该软件能否达到所要求的功能^[59]。

白箱测试又称为白盒测试也是软件测试的一种方法，需要测试软件系统的内容结构以及内部的运行方式，不需要测试软件的系统功能，而不是测试系统功能。在测试中与黑盒测试不同，白盒需要测试者有开发能力，要了解程序的结构及架构以开发者的思想去设计测试用例^[60]。

白箱测试多用于单元测试、系统的软件测试、和集成测试^[61]。

6.2 测试用例

6.2.1 用户功能测试

测试用例 1：

测试内容：当用户访问主页，是否会出现“已登录”内容。

操作方式：以访客身份访问博客。

测试结果：没有出现“已登录”的标签，测试结果成功。

测试结论：测试成功。

测试用例 2：

测试内容：当博主访问主页，是否会出现“已登录”内容。

操作方式：以博主身份访问博客。

测试结果：出现“已登录”的标签，测试结果成功。

测试结论：测试成功。

6.2.2 文章功能测试

测试用例 1：

测试内容：当用户访问主页，是否会看到博客的文章内容及文章详细内容。

操作方式：以访客身份访问博客，并点击文章标题或 Read More。

测试结果：能看到文章内容及文章详情，测试结果成功。

测试结论：测试成功。

测试用例 2：

测试内容：当博主访问主页，是否会看到博客的文章内容及文章详细内容。

操作方式：以博主身份访问博客，并点击文章标题或 Read More。

测试结果：能看到文章内容即文章详情，测试结果成功。

测试结论：测试成功。

6.2.3 评论功能测试

测试内容：查看相应的文章并查看文章下部评论区是否显示评论内容，并评论文章。

操作方式：随机进入某篇文章中，查看能否看到评论区的内容，并在评论输入框留下评论信息。

测试结果：评论内容成功显示，新的评论能够显示在评论区。

测试结论：测试成功。

6.2.4 关于页面测试

测试用例 1：

测试内容：查看关于页面能否出现关于博主的信息。

操作方式：用户点击 About 链接查看。

测试结果：显示关于页面的博主信息。

测试结论：测试成功。

测试用例 2：

测试内容：博主对关于页面的信息进行修改。

操作方式：博主编辑 About 的信息。

测试结果：可以成功修改博主个人的信息。

测试结论：测试成功。

6.2.5 测试结论

通过以上的多组测试，本博客系统通过各项操作的测试，且测试所用数据及测试用例基本上覆盖到系统全方面，综上所述，本系统通过测试。

第 7 章 总结与展望

本系统实现的是个个人的博客系统，基本上实现了国内互联网中一些主流博客的功能，文章，评论等功能。

本博客系统属于一个小型的软件系统，使用了 NodeJS 语言和 ExpressJS 系统框架，相对于 PHP 等技术，NodeJS 目前算是发展很快的，社区也很活跃，各种相应的库也都在不断更新中，技术文档也在不断更新中，使用 NodeJS 写博客系统主要还是以学习 NodeJS 的目的，在系统设计和代码编写过程中，本人感受到了 NodeJS 异步回调处理事件的方便好用的特点。系统设计前参考了 Wordpress 的系统设计，了解到很多关于博客系统的很多要点，博客的基本系统功能，如何设计页面，划分模块，设计数据库，页面设计的有效排版等等。

本系统是基于 NodeJS 语言开发，NodeJS 的异步 I/O 模型，因此能够大幅度的提高 Web 服务的高并发性。

本系统采用 MVVM 三层架构设计，包括表现层，业务逻辑模型，数据模型，各个层次之间松散耦合，能够让系统更容易的维护以及功能扩展。

本系统着重于数据的安全性，对重要信息采用加密处理，而且数据库中的数据删除采用逻辑删除替代物理删除，而且能够有效的防止 SQL 注入攻击，用户对东西访问进行了严格的权限分级，任何系统错误信息都会记录。

前端页面采用 Jade 模板引擎，更有效的书写 HTML，简洁清晰易懂，免去 HTML 中的冗余代码，对开发者更友好。

系统应用 NodeJS 的新兴技术，它独特的地方必会开发出一些优秀的应用。由于技术水平有限，以后在技术提升后相应会完善博客系统，本人相信会使用到更多 NodeJS 的特性。

参考文献

- [1] 李兴华. 基于 WebSocket 的移动即时通信系统[D]. 重庆大学,2013.
- [2] 黄明亮. 物联网开放平台的研究与设计[D]. 中国海洋大学,2013.
- [3] 郭宗辉. 传感器服务平台的设计与实现[D]. 中国海洋大学,2013.
- [4] 姚立. IBM 云计算平台下 NodeJS 应用支持环境的设计与实现[D]. 哈尔滨工业大学,2013.
- [5] 娄启林. 基于 Webkit 内核的桌面应用程序通用框架的构建与应用[D]. 北京交通大学,2014.
- [6] 李爽. 操作系统自动化部署探究[D]. 华南理工大学,2014.
- [7] 巴特尔. 基于 NodeJS 的理财应用的设计与实现[D]. 吉林大学,2014.
- [8] 安康. xPayPad 金融支付业务系统的设计与实现[D]. 北京邮电大学,2014.
- [9] 李博洋. 基于 Node.js 的分布式数字资源开放服务系统的设计与实现[D]. 北京邮电大学,2014.
- [10] 刘派. 基于 Node.js 的热计量监控预警系统的设计与实现[D]. 北京邮电大学,2014.
- [11] 许会元,何利力. NodeJS 的异步非阻塞 I/O 研究[J]. 工业控制计算机,2015,03:127-129.
- [12] 缪进,何加铭. 基于 WebRTC 的浏览器端 Web 服务器的设计与实现[J]. 移动通信,2015,08:56-60.
- [13] 柯肇丰,曾霞霞. 基于 HTML5+nodeJS+MongoDB 构建在线图像编辑器系统[J]. 福建电脑,2015,06:42-44.
- [14] 郑立国,于宏伟. 基于 Nodejs 的本机应用与 Web 应用统一编程模型的研究[J]. 电子技术与软件工程,2015,12:256-257.
- [15] 李勇. NORADLE——融合 nodejs 和 oracle 生态圈的连接设计[J]. 通讯世界,2015,15:46-47.
- [16] 李正学,许捍卫. 异步非阻塞瓦片地图服务器的实现[J]. 测绘科

学,2015,10:128-132.

- [17] 黄扬子. 基于 NodeJS 平台搭建 REST 风格 Web 服务[J]. 无线互联科技,2015,16:57-59.
- [18] 隋春霞. 基于 B/S 的期货行情分析系统的设计和实现[D]. 大连理工大学,2014.
- [19] 李桂林. 基于云计算的 CDN 平台系统设计研究[D]. 北京邮电大学,2015.
- [20] 王令宇. 基于 SaaS 模式的即时通信平台的设计与实现[D]. 北京邮电大学,2015.
- [21] 丁磊. 基于 WebSocket 的消息推送系统设计与事件匹配技术研究[D]. 北京邮电大学,2014.
- [22] 付予. 基于 WebRTC 的电子白板实现机制的设计与实现[D]. 北京邮电大学,2015.
- [23] 林瑶. web 实时数据同步研究[D]. 北京邮电大学,2015.
- [24] 荣林林. 自适应浏览器的高性能 Web 扩展应用开发系统的研究与设计[D]. 北京邮电大学,2015.
- [25] 张镭翕. 基于 Node.js 的学习交流平台的设计研究[D]. 云南大学,2015.
- [26] 吴川环. 基于微信控制物联网设备的服务器系统[D]. 广东工业大学,2015.
- [27] 董纪国. 基于 B/S 架构的物联网应用组态平台研发[D]. 南京大学,2015.
- [28] 高原. 服务器端 javascript 技术研究[J]. 信息与电脑(理论版),2012,01:78+80.
- [29] 唐诚. Redis 数据库在微博系统中的实践 [J]. 厦门城市职业学院学报,2012,03:55-59.
- [30] 肖在昌,杨文晖,刘兵. 基于 WebSocket 的实时技术 [J]. 电脑与电信,2012,12:40-42.
- [31] 林振,邵乾飞,陈坤如. 基于 Nodejs 的校园智能售货机实践[J]. 电脑知识与技术,2015,32:142-143.
- [32] 单振华,王舒憬,陈凯,强杰. 基于 NodeJS 的智能家居语音控制系统服务器端设计与实现[J]. 工业控制计算机,2016,04:66-67.
- [33] 王金龙. 基于 Node.js 的图片分享应用的设计与实现[D]. 南京理工大学,2015.

- [34] 宋平亮. 基于 MongoDB 的航道数据 Web 服务研究[D]. 大连海事大学,2015.
- [35] 陈达. 基于 HTML5 的跨平台技术在公交服务系统中的应用研究[D]. 大连海事大学,2015.
- [36] 周鹏. 分布式聊天服务子系统的设计与实现[D]. 南京大学,2015.
- [37] 王越. 基于 nodejs 的微博系统的设计与实现[D]. 电子科技大学,2014.
- [38] 王强. 某高校创意产品网络信息平台设计与实现[D]. 电子科技大学,2015.
- [39] 马鹏飞. 基于智能手机的景区二维码导游系统的设计与实现[D]. 西安电子科技大学,2014.
- [40] 罗剑洪. 基于即时通讯的工程会议系统的应用研究[D]. 华南理工大学,2015.
- [41] 王耀武. 基于 3G 和 WebGIS 的基层农情管理信息系统[D]. 中国农业科学院,2012.
- [42] 陈浩. NodeJS 中 ModelProxy 接口配置建模研究[J]. 软件导刊,2014,09:67-69.
- [43] 仇钧. 基于微博社会网络的用户兴趣模型研究[D]. 上海交通大学,2013.
- [44] 彭娜. 基于 Node.JS 博客系统的设计与实现[D]. 大连理工大学,2013.
- [45] 李晶. NodeJS——服务器端 JavaScript 运行环境[J]. 程序员,2010,12:29.
- [46] 李正学,许捍卫. 利用 NodeJS 的异步非阻塞瓦片地图服务器[J]. 测绘科学,,:1.
- [47] 田旭. 基于 WebRTC 技术的远程面试系统的设计与实现[D]. 华中师范大学,2014.
- [48] 武路. 移动应用实时统计系统设计与实现[D]. 西安电子科技大学,2014.
- [49] 魏云申. 基于 WebGL 的全景 3D 漫游系统的设计与实现[D]. 南京大学,2016.
- [50] 李国栋. T-MPLS 的 OAM 研究[D]. 北京邮电大学,2009.
- [51] 赵明砚,单世民,赵风强. 数据库原理与开发[M]. 北设:人民邮电出版社,2008.
- [52] Kristina Chodorow, Michael Dirolf. MongoDB:权威指南[M]. 北京:人民邮电出版社,2011(01):6-7.
- [53] Stefan Tilkov, Steve Vinoski. Node.js: Using JavaScript to Build High Performance Network Programs[J]. 2010(12):80-83.

- [54] Steve Souders. 高性能网站建设指南[M]. 北京.电子工业出版社,2008.
- [55] 赵恺. 基于 MVC 模式和 Struts 框架的 Web 应用设计与实现[J]. 兰州工业高等专科学校学报.
- [56] Mozilla 重写 Firefox 代码[J]. 软件世界,2007(10): 1-10.
- [57] 蔡柳青. 基于 MongoDB 的云监控设计与应用[D]. 北京交通大学,2011.
- [58] 陈敏. 基于 JADE 平台的信息代理的研究及实现[D] . 暨南火卞,2005.
- [59] 王顺奎. 国外信息融合技术概述[J]. 红外与激光技术 , 1995 ,(1): 1-4.
- [60] Simon Willson. Node.JS is genuinely exciting[J]. Simon Will son Weblog, 2009.
- [61] 陈慧玲 , 帅立国 , 姜昌金. 基于 HTTP 协议的论坛群发技术的研究[J]. 计算机技术与发展.2007(03):82-84.

致 谢

本文是在于海洋老师的悉心指导下完成的，首先我衷心的感谢我的指导老师。于老师在论文选题、研究方法、论文的写作等诸多方面都给予了细致、精心的指导，他严谨的治学态度，深厚的理论知识使我受益匪浅，在此衷心的感谢于老师！

在设计过程中，我通过查阅大量有关资料，温习了所学的相关课程，通过网上信息检索、与同学交流经验、和向老师请教等方式，使自己学到了不少知识，尽管前期在硬件的调试和程序代码编写的过程中经历了不少艰辛，但是后来还是在老师、同学们的帮助下完成了论文研究工作。撰写论文的过程中我学到了很多东西，提高了我自身独立完成工作的能力，增强了对工作处理能力的信心，坚信一定会对之后的工作学习生活有很重要的影响。最后感谢在学校期间教我课程的各位老师，感谢陪伴我走过学校生活的同学们，是你们让我的大学生活充实而有意义！

由于本人的学识水平有限，论文中或多或少会有些偏颇和不恰之处，恳切希望老师能够予以批评和指正。

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得黑龙江大学或其他教育机构的学位或证书而使用过的材料。

学位论文作者签名：郭竞莹

签字日期：2016 年 11 月 15 日

学位论文版权使用授权书

本人完全了解黑龙江大学有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权黑龙江大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

学位论文作者签名：郭竞莹

导师签名：于海平

签字日期：2016 年 11 月 15 日

签字日期：2016 年 11 月 15 日

学位论文作者毕业后去向：

工作单位：

电话：

通讯地址：

邮编：