

湖南人文科技学院

本科生毕业设计说明书

|  |  |  |  |
| --- | --- | --- | --- |
| 项目名称： | **大学生知识分享问答平台** | | |
| 学院名称： | **信息学院** | | |
| 学生姓名： | **蒋芳** | 学号 | **16436213** |
| 专业年级： | **软件工程2016级** | | |
| 指导教师： | **付又香/陈洁** | | |
| 教师职称： | 高级实验师/副教授 | | |

湖南人文科技学院教务处

二〇一九年制

**毕业设计原创性及知识产权声明**

本人郑重声明：所呈交毕业设计是本人在导师指导下独立进行研究所取得的成果，对本设计的研究做出重要贡献的个人和集体，均已在文中明确标明。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

本毕业设计成果归湖南人文科技学院所有。

作者签名： 201 年 月 日

导师签名： 201 年 月 日

**大学生知识分享问答平台**

# 摘 要：

近年来，随着科学与信息计算的进步，以及人们对信息资源需求的增加，在这个互联网时代，通过网络平台进行学习已经成为一种常态。知识分享的方式开始多种多样，随之而来的是许多知识分享问答平台快速成长，给人提供了一种在线学习、求解、答题的渠道。通过网络平台进行交流学习，提高了学习的效率，降低了学习的成本。

知识共享是知识管理的重要部分、也是知识积累的需求。在知识共享的过程中，在将自己所拥有的知识积累起来的同时必然会发现自己知识面的缺陷，并完善自己所欠缺的地方。知识分享问答平台是用户针对自身的需求进行提出问题，寻求解答。其他用户提供该问题的答案。随着知识分享范围的扩大，用户在分享的同时，知识面也在不断的增加。

本次选题采用MVVC模式，选用Vue、Node、MongoDB作为开发工具，设计一个大学生知识分享问答平台。旨在给大学生提供一个知识交流学习，问题讨论，技术交流、拓宽知识面、知识共享的社交类平台。选题首先是对系统进行分析，介绍系统设计的目的、相关的国内外研究动态、系统的功能、系统用例分析和系统的运行环境，明确了用户的需求；接下介绍了数据库的详细设计；最后通过截图展示和文字描述详细介绍大学生知识分享问答平台的前端和后台功能设计与实现。

**关键词：** MVVC；Vu；Node；MongoDB；知识共享

# **Abstract：**

In recent years, with the progress of science and information computing, as well as the increase of people's demand for information resources, learning through network platform has become a norm in this Internet era. The ways of knowledge sharing are varied, followed by the rapid growth of many knowledge sharing question-and-answer platforms, which provide people with an online learning, solving and answering channels. Through the network platform to exchange learning, improve the efficiency of learning, reduce the cost of learning.

Knowledge sharing is an important part of knowledge management and also the requirement of knowledge accumulation. In the process of knowledge sharing, when we accumulate our knowledge, we will inevitably find the defects of our knowledge and improve what we lack. Knowledge sharing q&a platform is a platform for users to raise questions and seek answers to their own needs. Other users provide answers to this question. With the expansion of the scope of knowledge sharing, users are sharing at the same time, the scope of knowledge is also increasing.

This topic adopts MVVC mode, and USES Vue, Node and MongoDB as development tools to design a question-and-answer platform for college students to share knowledge. It aims to provide a social platform for college students to exchange knowledge, discuss problems, exchange technologies, broaden the scope of knowledge and share knowledge. The first topic is to analyze the system, introduce the purpose of system design, relevant domestic and foreign research trends, system functions, system use case analysis and system operating environment, and make clear the needs of users; Next, the detailed design of the database is introduced. Finally, through screenshot display and text description, the paper introduces in detail the design and implementation of the front-end and back-end functions of the question-and-answer platform for college students' knowledge sharing.

**Key Words:** MVVC；Vu；Node；MongoDB；Knowledge Sharing

**目录**

[摘 要： I](#_Toc16963)

[Abstract： II](#_Toc11692)

[目录 3](#_Toc3476)

[前言 5](#_Toc24036)

[第1章 系统分析概述 5](#_Toc22924)

[1.1 系统设计目的 5](#_Toc10935)

[1.2 国内外研究现状 6](#_Toc3328)

[第2章 功能需求分析 7](#_Toc15934)

[2.1 业务逻辑设计 7](#_Toc21992)

[2.2 文章分类： 7](#_Toc12717)

[2.3 用户： 7](#_Toc1862)

[2.3.1 登录用户 7](#_Toc6193)

[2.3.2 未登录用户 7](#_Toc2811)

[2.4 用例图 7](#_Toc29163)

[2.4.1 未登录用户 7](#_Toc12128)

[2.4.2 登录用户 8](#_Toc20387)

[2.5 开发选择 9](#_Toc5799)

[2.5.1 MVVM模式 9](#_Toc29985)

[2.5.2 Vue 9](#_Toc32338)

[2.5.3 Node.js 10](#_Toc30548)

[2.5.4 MongoDB 10](#_Toc1851)

[第3章 数据库设计 12](#_Toc31920)

[3.1 E-R图 12](#_Toc11011)

[3.2 数据库需求分析 12](#_Toc14692)

[3.3 系统功能设计 13](#_Toc12775)

[3.3.1 数据库设计 13](#_Toc20055)

[3.3.2 用户集合 13](#_Toc21579)

[3.3.3 文章集合 13](#_Toc10518)

[3.3.4 评论集合 14](#_Toc19828)

[3.3.5 问题集合 14](#_Toc13327)

[第4章 项目详解 15](#_Toc4429)

[4.1 项目目录详解 15](#_Toc25043)

[第5章 系统功能实现 17](#_Toc9521)

[5.1 后台功能设计 17](#_Toc13107)

[5.1.1 后台分析 17](#_Toc23559)

[5.1.2 注册 17](#_Toc6002)

[5.1.3 登录 18](#_Toc30769)

[5.1.4 退出登录 19](#_Toc16100)

[5.1.5 发表文章 20](#_Toc2599)

[5.1.6 删除文章 22](#_Toc31670)

[5.1.7 查看文章列表 23](#_Toc22707)

[5.1.8 用户验证 24](#_Toc27052)

[5.2 前端功能设计 25](#_Toc18464)

[5.2.1 前端分析 25](#_Toc7448)

[5.2.2 页面路由 26](#_Toc8162)

[5.2.3 请求跨域 27](#_Toc23137)

[5.2.4 富文本 28](#_Toc19070)

[设计总结 28](#_Toc8903)

[致谢 30](#_Toc18879)

[参考文献 31](#_Toc2125)

# 前言

近年来，随着科学与信息技术的进步，以及人们对信息资源需求的增加，在这个互联网时代，通过网络平台进行学习已经成为一种常态。知识分享的方式开始多种多样，随之而来的是许多知识分享问答平台快速成长，给人提供了一种在线学习、求解、答题的渠道。通过网络平台进行交流学习，提高了学习的效率，降低了学习的成本。

知识共享是知识管理的重要部分、也是知识积累的需求。在知识共享的过程中，在将自己所拥有的知识积累起来的同时必然会发现自己知识面的缺陷，并完善自己所欠缺的地方。知识分享问答平台是用户针对自身的需求进行提出问题，寻求解答。其他用户提供该问题的答案。随着知识分享范围的扩大，用户在分享的同时，知识面也在不断的增加。

大学生知识分享问答平台是为了给大学生提供一个知识交流学习，问题讨论，技术交流、拓宽知识面的社交类平台。在平台中用户可以将自己平时积累的知识分享到平台上，也可以针对自身的需要通过搜索问题查找答案，亦可以向他人寻求解答，与其他用户进行交流探讨。这个讨论交流的过程也是知识的交流与共享是一种互动的过程，使得成员间的知识得以分享，并间接使得学生之间能分享彼此的知识，有利学生成长。

本次毕业设计采用Vue + Node + MongoDB，独立完成整个系统的设计与实现，独立完成整个系统的设计与实现。前端采用vue.js，后台使用node+mongodb，本次毕设，选择使用市场上比较前沿的技术开发，融合所学知识，加深对平台开发相关技能的理解，进行一次解决社会现实问题的项目实践，提高自身对知识理解和综合运用的能力。

1. 系统分析概述

## 系统设计目的

大学生分享问答平台采用MVVC模式进旨在设计一个以Vue + node.js + mongodb为基础，功能简单且轻量、给大学生提供一个知识交流学习，问题讨论，技术交流、拓宽知识面的社交类平台。在平台中用户可以将自己平时积累的知识分享到平台上，也可以针对自身的需要通过搜索问题查找与答案相关的文章，亦可以向他人寻求解答，与其他用户进行交流探讨。有利于大学生促进学习、记录学习收获、交流学习心得看法

## 国内外研究现状

近年来，随着信息技术与经济的快速发展，以及人们对信息资源需求的增加，在这个互联网时代，人们可以通过网络平台快速的获取到自己所需的知识，也可以快速的分享和传播自身所获取的知识，通过网络平台进行学习已经成为一种常态。

在我国，网络社区平台蓬勃发展，鼓励用户在会联网上进行互动，加强用户之间的交流与沟通，促进用户之间的知识交换；肯定用户在知识共享上的劳动成果，使知识共享成为互联网发展的源泉动力[12]。通过网络进行社交和知识共享，逐渐成为人们知识共享的重要渠道；国内外2000年至2017年有关社交网络用户知识共享研究的478篇文献进行全面深入分析；对该领域的"日常知识共享"实证研究的186篇文献进行内容分析,构建了实证研究、理论基础编码表,并对知识共享行为的"影响因素"进行抽取和归纳分析；内容分析结果表明,当前社交网络用户知识共享研究呈现文献数量增长迅速、实证研究取向突出、研究主题较为集中、研究框架初显成型的特征;研究内容主要包括日常知识共享的行为特性、单个社交平台的知识共享、基于社会资本理论探讨、知识共享行为的前置影响因素等;未来将呈现研究热度继续增加、研究视角学科交叉性、研究方法多元化、理论创新不断增强的发展趋势[11]。

# 功能需求分析

## 业务逻辑设计

1. 业务逻辑描述如下：
2. 首次使用的用户查看文章和查看部分评论。若想进行其他操作则需要进行注册登录。
3. 新用户登录后可发表文章，并且可在个人中心对发表的文章进行查看、修改、删除；在浏览其他用户发表的文章时可以发表评论；亦可对其他用户提出的问题进行回答。
4. 用户发表评论后，可对发表的评论进行查看、修改删除
5. 用户可对平台用户提出问题，等带其他用户回答
6. 用户可退出登录

## 文章分类：

1. 文章分类：公共知识（就业、考研）、专业知识。

## 用户：

### 登录用户

1. 登录、注册、退出
2. 提问、答复
3. 发表文章、查看文章、修改文章、删除文章
4. 新增评论、修改评论、删除评论，查看评论

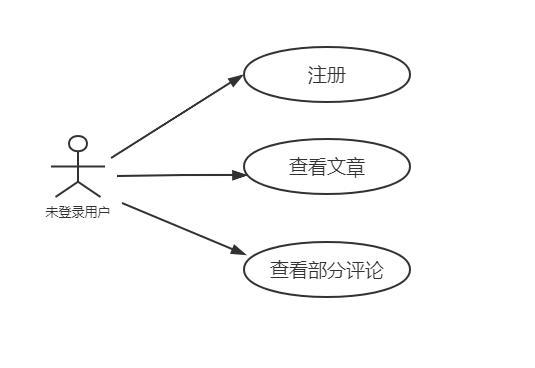
### 未登录用户

1. 注册、查看文章、查看评论

## 用例图

### 未登录用户

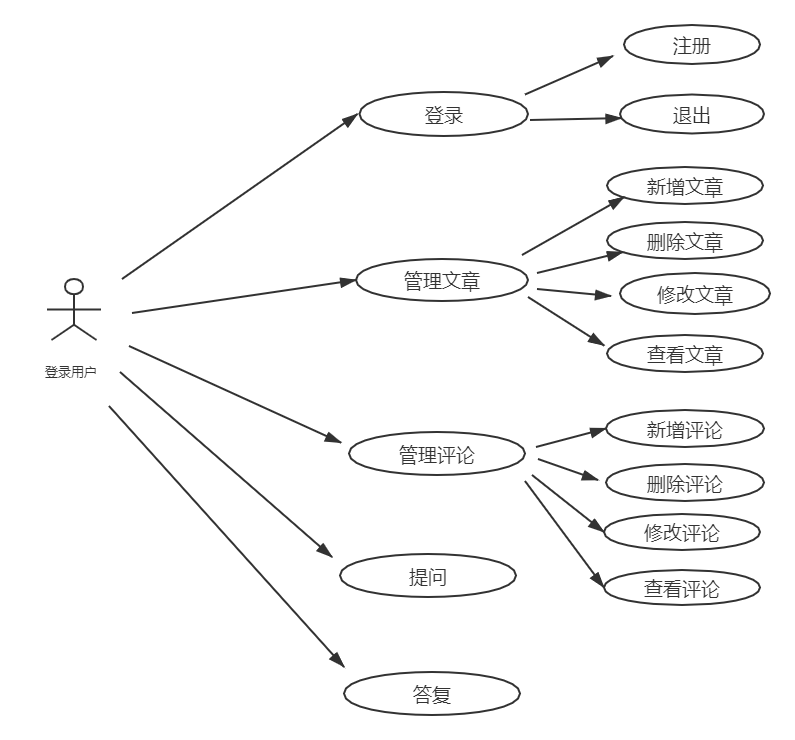
允许未登录用户进行注册、查看文章、查看部分评论的功能



### 登录用户

用户登录后，可执行以下操作：

1. 登录、注册、退出
2. 提问、答复
3. 文章： 发布文章、修改文章、删除文章、查看文章
4. 评论：新增评论、修改评论、删除评论，查看评论



1. 用户发表文章后，用户具有对自己发布的文章进行修改文章内容、查看发布的文章、删除已发布文章等功能。同时，用户可以查看他人对自身所发布文章的评论。
2. 用户对某一篇文章发布评论后，可以对发布的评论进行修改评论、删除已发布评论、查看发布的评论等功能。同时也可以查看他人关于这篇文章所发布的评论

## 开发选择

### MVVM模式

MVVM是Model-View-ViewModel的简写，由经典的软件架构MVC衍生而来，最早是由微软公司提出并运用，是MVP（Model-View-Presenter）模式与WPF结合的应用方式时发展演变过来的一种新型架构架构；MVVM有助于将图形用户界面的开发与业务逻辑或后端逻辑（数据模型）的开发分离开来 [15]。

MVVM是一种数据双向绑定的响应式框架。主要把每个页面分层了M、V、VM，其中VM是MVVC的思想核心，负责M、V 之间的调度。Model(模型):即数据层，负责页面中数据存储。View(视图层):他的任务就是从ViewModel层获取数据，然后显示，也就是每个页面的HTML结构。ViewModel(视图模型):监听View和Model改变，负责封装业务逻辑处理，封装网络处理，封装数据缓存。View(视图)和ViewModel(视图模型)之间建立联系通过双向绑定。

在 MVVM 架构下，View和Model之间没有直接联系，他们之间的交互通过ViewmModel进行。模型(Model) 和视图模型( ViewModel) 双向交互， 因此 ，View上数据的发生改变会同步到 Model 中，而 Model中 数据的变化也会马上反应到 View 上。

ViewModel 通过数据双向绑定将 View 层和 Model 层进行连接，View 和 Model 之间工作自动同步，不需要人为去干涉，所以，开发者只需专注于业务逻辑，不需要手动去操作 DOM, 也不需要注意数据状态同步的问题，复杂的数据状态完全由 MVVM 来统一管理、维护。

### Vue

Vue.js是一个轻巧、高性能、可组件化的MVVM库，同时拥有非常容易上手的API；Vue.js是一个构建数据驱动的Web界面的库；是一套构建用户界面的渐进式框架；与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合；另一方面，Vue 完全有能力驱动采用单文件组件和 Vue 生态系统支持的库开发的复杂单页应用；Vue.js 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件[2]。

Vue.js具有插件化、数据双向绑定、轻量级架构、提供指令和过滤器等特点。Vue 的核心是一个 "响应式的数据绑定系统"。在项目开发过程中，数据和视图绑定后,数据将保持同步。每当修改数据后， DOM 也进行相应的更新；反之，在视图上进行操作，相关的数据也会随之改变。即关于DOM的操作都由Vue执行了，我们只需操作数据即可。除此之外，在项目的开发过程中，使用Vue.js 可以引入插件和第三方库，能够很方便的搭建前端页面和实现一部分功能，并且完成与后台的交互功能，便于开发。

### Node.js

Node.js是一个基于Chrome V8 引擎的 JavaScript 运行环境；也是运行在服务端的JavaScript。发布于2009年5月，由Ryan Dahl开发，实质是对Chrome V8引擎进行了封装。Node.js对一些特殊用例进行优化，提供替代的API，使得V8在非浏览器环境下运行得更好。 V8引擎执行Javascript的速度非常快，性能非常好。Node.js是一个基于Chrome JavaScript运行时建立的平台， 用于方便地搭建响应速度快、易于扩展的网络应用。Node.js 使用事件驱动， 非阻塞I/O模型而得以轻量和高效，非常适合在分布式设备上运行数据密集型的实时应用[5]。

Node.js的出现打破了前端脚本语言和后台开发语言之间的屏障，使JavaScript不仅能用于前端开发，而且还能用于后台开发，让Web开发变得更简单；更重要的是，Node.js在性能、实时处理和高并发等多个领域都有非常优异的表现；Node.js提供了一种简单的、用于创建高性能服务器及可在该服务器中运行的各种应用程序的开发工具。

### MongoDB

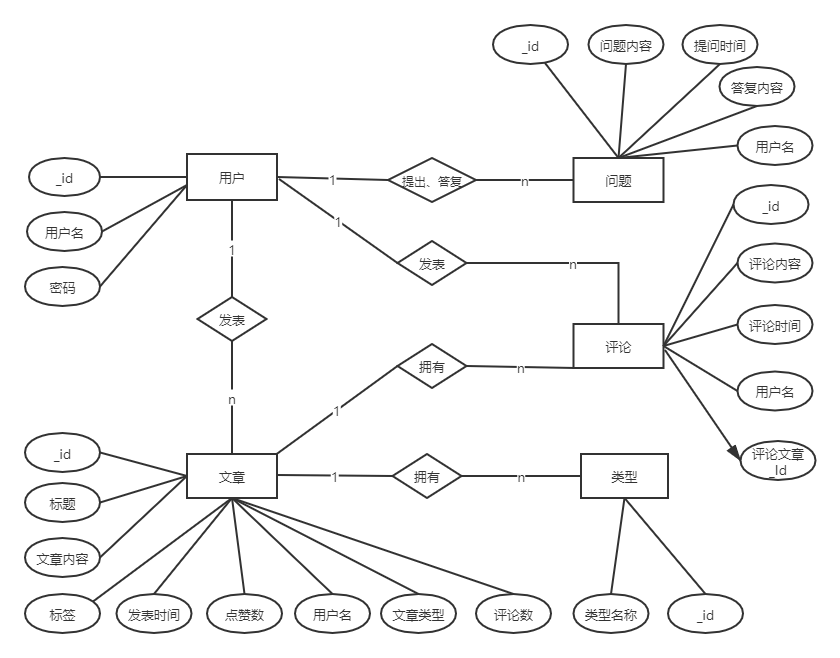
与传统的关系型数据库不同，MongoDB是一种面向文档的数据库；是一个基于分布式文件存储的开源数据库系统；MongoDB 将数据存储为一个文档，数据结构由键值(key=>value)对组成；MongoDB 文档类似于 JSON 对象；字段值可以包含其他文档，数组及文档数组[3]。

在MongoDB中多个文档组成集合，同样的是多个集合组成数据库。其中文档是一组键值(key-value)对(即 BSON)的形式，即{ "name":"MongoDB" }。MongoDB 的文档中的值中支持设置不同的字段，并且相同的字段也支持不同的数据类型，这与关系型数据库有很大的区别。当然，文档中也不能存在相同的键，MongoDB类型不同、大小写不同对应的键也是不一样的。若是说文档类似于数据库中的行，那么集合是如同数据库的表。MongoDB中集合是无模式的，也就是说集合中的文档不但值的类型可以不同，键也可以不同。当然也可以将不同类型的文档放到不同的集合中。

在本项目设计开发的过程中可以使用MongoDB丰富的表达式对数据进行操作。在搭配Node.js进行后台开发时，使用update修改器: $inc $set $unset $push $pull，对数据或指定的数据字段进行增加、修改、删除等操作。同样的是可以使用JSON形式的指令也可以轻易地对文档中的数组及对象进行查询。并且在向数据库中插入数据时，不需要事先在数据库的集合中建立字段，只需将前端页面需要的字段与后台对接上即可。

# 数据库设计

## E-R图



## 数据库需求分析

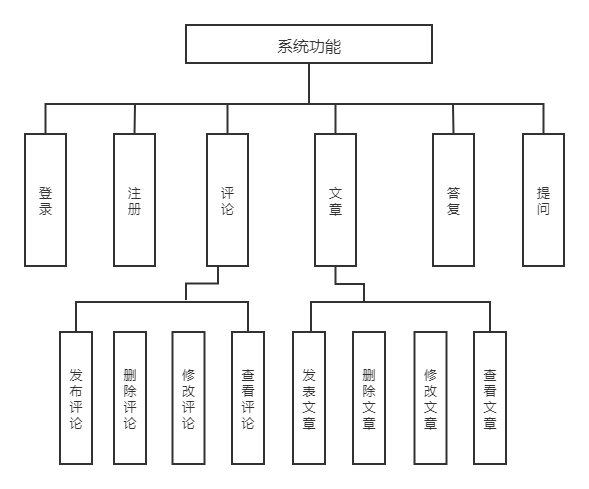
1. 用户分为登录用户和未登录用户
2. 未登录用户可以注册、查看文章、查看部分评论
3. 登录用户可以对文章进行发布、删除、修改、查看；亦可对评论进行发表、删除、修改、查看；还可以提问、答复
4. 一篇文章可拥有多评论
5. 一个提问可以有多答复
6. 一种类型对应多篇文章
7. 一个问题对应多个答复

实体属性：

1. 用户：唯一\_id、用户名、密码
2. 文章：唯一ID、文章标题、文章内容、文章标签、发表时间、点赞数、评论数、文章类型、用户名
3. 评论：唯一ID、评论内容、评论时间、评论文章Id、用户名
4. 提问：唯一ID、提问内容、提问时间、答复内容、用户名

## 系统功能设计

### 数据库设计



### 用户集合

| 字段名 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- |
| \_id | ObjectId | 是 | 唯一ID |
| username | String | 是 | 用户名 |
| password | String | 是 | 密码 |

### 文章集合

| 字段名 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- |
| \_id | ObjectId | 是 | 唯一ID |
| articlTitle | String | 是 | 文章标题 |
| articleContent | String | 是 | 文章内容 |
| articleTag | Array | 是 | 文章标签 |
| publishedTime | Date | 是 | 发表时间 |
| likeCount | Number | 否 | 点赞数 |
| commentsCount | Number | 否 | 评论数 |
| articleType | String | 是 | 文章类型 |
| userName | String | 是 | 用户名 |

### 评论集合

| 字段名 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- |
| \_id | ObjectId | 是 | 唯一ID |
| commentscontent | String | 是 | 评论内容 |
| commentsTime | String | 是 | 评论时间 |
| commentsArticleId | String | 是 | 评论文章Id |
| username | String | 是 | 用户名 |

### 问题集合

| 字段名 | 类型 | 必选 | 说明 |
| --- | --- | --- | --- |
| \_id | ObjectId | 是 | 唯一ID |
| questions | String | 是 | 提问内容 |
| time | String | 是 | 提问时间 |
| answer | Array | 否 | 答复内容 |
| username | String | 是 | 用户名 |

# 项目详解

## 项目目录详解

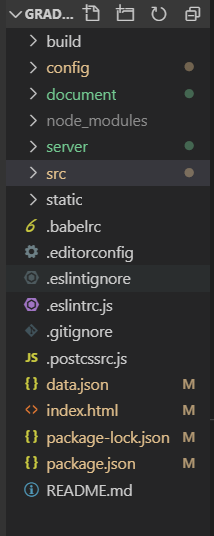


图 4-1

项目目录结构

1. build：构建脚本目录 编辑相关的文件夹
2. build.js ==> 生产环境构建脚本；
3. check-versions.js ==> 检查npm，node.js版本；
4. utils.js ==> 构建相关工具方法；
5. vue-loader.conf.js ==> 配置css加载器以及编译css之后自动添加前缀；
6. webpack.base.conf.js ==> webpack基本配置；
7. webpack.dev.conf.js ==> webpack开发环境配置；
8. webpack.prod.conf.js ==> webpack生产环境配置；
9. config：项目配置 webpack的配置文件
10. dev.env.js ==> 开发环境变量；
11. index.js ==> 项目配置文件；
12. prod.env.js ==> 生产环境变量；
13. node\_modules：npm 加载的项目依赖模块
14. server: 后后台代码
15. src：主要要开发的目录，基本上要做的事情都在这个目录里。
16. static：静态资源目录，如图片、字体等。不会被webpack构建
17. .babelrc： babel-loader 配置 把es6代码转换成浏览器能识别的代码的工具。可以理解为babel是javascript语法的编译器
18. .editorconfig： 是编辑器自身的配置文件，定义编辑器的设置，如字符类型等
19. .eslintignore： 是忽略eslintrc规则的文件列表，加入其中的文件名可以避免eslint的语法检查
20. .eslintrc.js： 是eslint配置文件，ESLint是可组装的JavaScript和JSX检查工具， 是一个ES/JS 语法规则和代码风格的检查工具，保证代码的一致性和避免错误
21. .gitignore: 配置在该文件中的文件列表不会被提交到github或者其他代码仓库上
22. .postcssrc.js： PostCSS是一款使用插件去转换CSS的工具
23. index.html：项目本身模本文件，首页入口文件，可添加一些 meta 信息等
24. package.json：存放基础数据、创建时的填写的信息，还有配置和依赖包等信息
25. README.md：项目说明文档，markdown 格式

# 系统功能实现

## 后台功能设计

### 后台分析

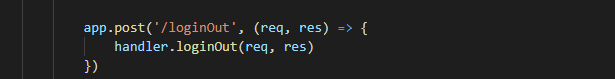
后台使用Node来搭建服务器，使用MongoDB作为数据库， 完成数据的增删改查

目录位置： server/

1. server/base-server.js: 搭建服务器，写后台接口，使用bodyParser解析中间件
2. server/dev-server.js: 设置请求头，实现跨域，开发的后台这此运行，进入server 文件夹后运行 node dev-server.js
3. server/handle.js: 具体实现一系列的后台接口，并暴露出去
4. server/token.js：生成登录时的token，有效期一周，校验token
5. server/utils.js：获取客户端ip地址方法

### 注册

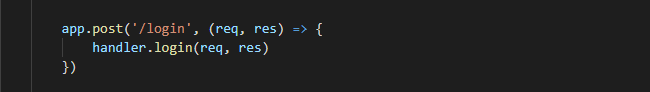
用户点击注册，输入用户名和密码，实现注册功能。用户名唯一，在点击注册时，会先从数据库中查询是否含有该用户名，若是不存在则完成注册，否则提示用户名已存在。注册完成后即可进入登录组件进行登录并进行其操作。





### 登录

用户在完成注册功能后，则可以点击登录，进入登录组件输入用户名和密码完成登录功能。在登录过程中，会将查询前端输入的用户名数据库中是否存在，若是不存在，则提示用户不存在。若是存在，则将生成一个token，将其插入数据库并返回给前端。若是用户名和密码都正确，则登录成功，否则提示登录失败。

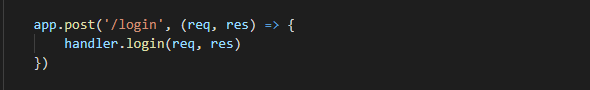






### 退出登录

在登录过后，用户离开该平台可以选择退出登录功能，退出登录后，会删除数据库存储的token

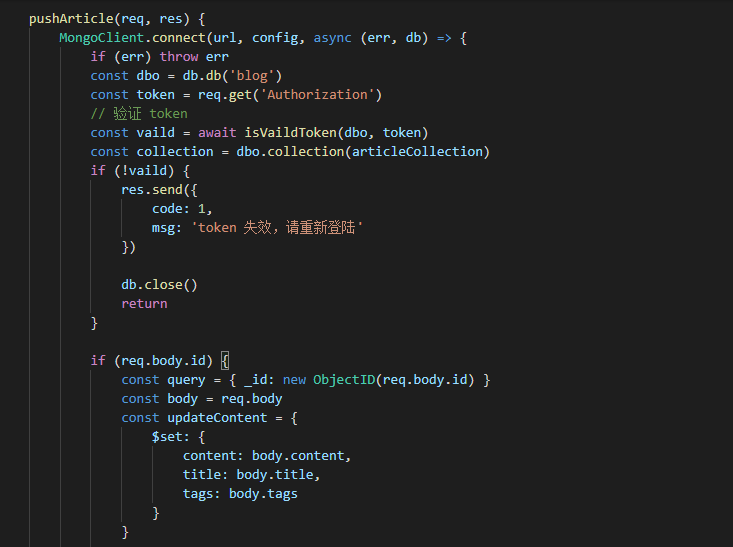




### 发表文章

用户在完成登录功能后，可点击发表文章，进入文章编辑组件，完成文章的编写后，可进行文章的发表。或者对某篇已经发表的文章进行修改编辑后进行文章的发表。点击发表文章后，若是对文章进行修改更新，则会将修改的字段在数据库中进行替换。若是发表一篇新的文章，则会将前端填写关于文章的内容插入到数据库中。



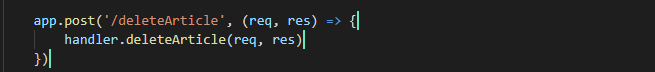






### 删除文章

用户在删除文章时，会验证token是否失效，若是失效则提示重新登录，反之则根据文章的ObjectID对文章进行删除。

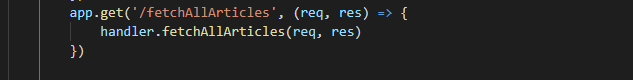


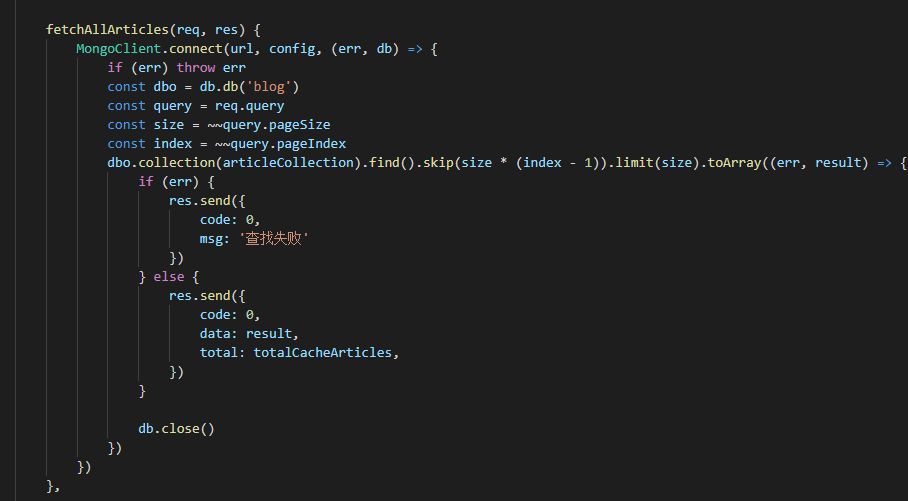




### 查看文章列表

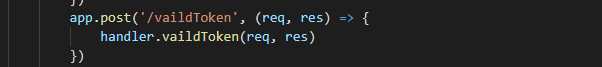
查询数据库中的所有文章，并对其进行分页处理





### 用户验证

用户登录后将token插入数据库，并将token返回给前端，由前端将其存储在浏览器的Local Storage 中，有效期为七天。当token失效后，用户在进行操作时会被提示重新登。





## 前端功能设计

### 前端分析

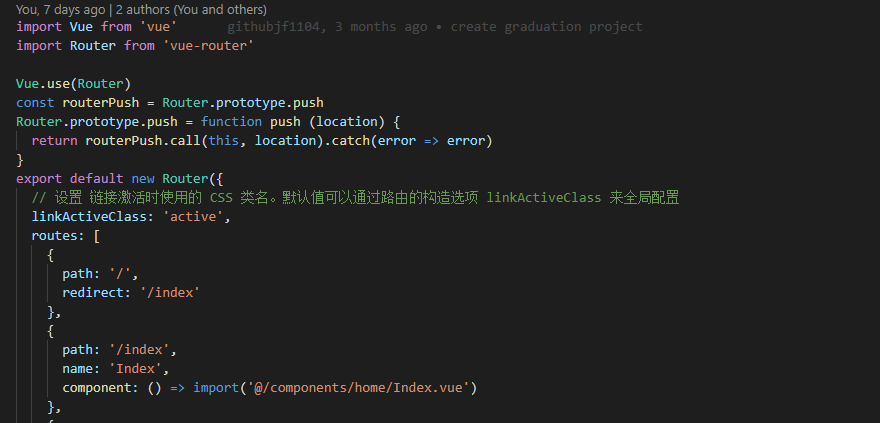
前端项目使用vue-cli脚手架搭建项目，使用Vue，并在项目中引入组件和第三方库，将不同页面分成不同的组件进行开发。

目录位置：src/

* 1. api: 存放前端与后台进行请求交互的接口，
  2. assets：资源目录，放置图片或者公共js、公共css。文件夹下的资源会被webpack构建；
  3. components：组件目录，项目组件所放位置；
     1. components/home/Index.vue 项目的首页，显示文章列表信息，可文章进行点赞功能
     2. components/nav/Nav.vue 项目头部导航栏，包括登录、注册、退出登录、搜索、发表文章、提问等功能
     3. components/register/Register.vue： 注册组件,点击导航栏的注册填写用户名和密码完成注册功能，其中用户名是唯一的
     4. components/login/Login.vue: 登录组件，用户点击导航栏的登录完成登录功能
     5. components/edit/Edit.vue: 发表文章组件，用户点击导航栏的发表文章进入该组件填写项对应的内容，完成发表文章的功能
     6. components/details/Detail.vue: 详情组件，在文章列表点击某篇文章，进入该组件查看文章的内容信息
     7. components/personal/Personal.vue： 个人中心组件，用户可在该组件中查看自己个个人信息，以及所发表的文章
     8. components/category/category.vue: 专栏组件，包括不同类型的文章
     9. components/common/： 公共组件，供其他组件复用
     10. components/comments/Comments.vue: 评论组件，当用户想对一篇文章进行评论时，点击文章页面的评论按钮进入该组件完成对文章的评论功能
  4. router：前端路由，实现路由跳转。配置的路由路径写在index.js里面；
  5. store: vuex状态管理文件，需要共享的数据在这进行管理
  6. utils: 组件所需的公共方法
  7. App.vue：根组件；
  8. main.js：入口js文件；

### 页面路由

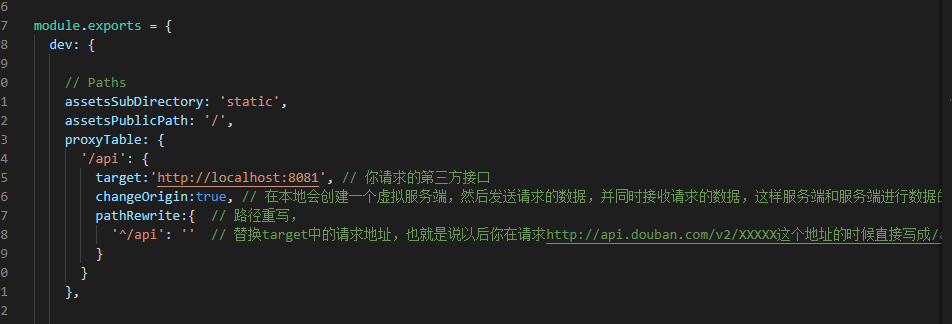
使用vue插件 vue-router 实现路由功能



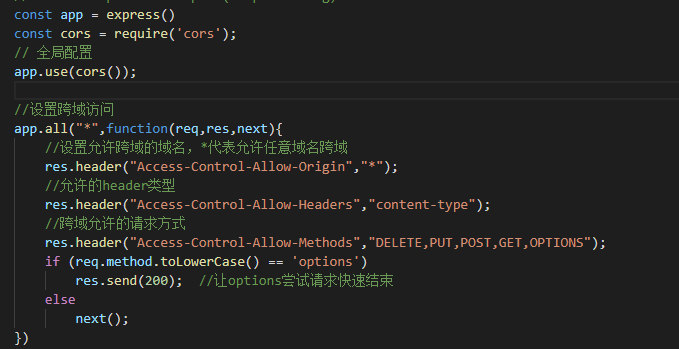


### 请求跨域

前端设置跨域在配置文件重设置代理



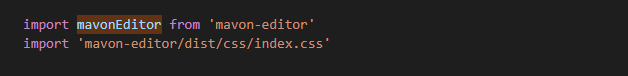
后台设置跨域使用cors和设置请求头实现跨域



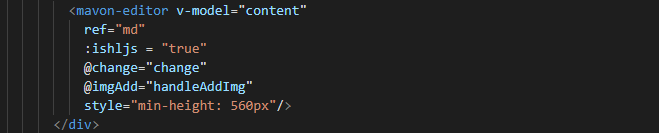
### 富文本

文章编辑组件采用富文本编辑器实现

main.js中引入组件



组件组件内使用富文本组件



# 设计总结

本文选择了基于Vue + Node + MongoDB开发作为研究的重点，知乎的内容管理系统为分析实例，对平台的开发进行探讨和实践。主要针对数据双向绑定的响应式框架MVVM三层开发模式的特点分析。并在此基础上进行平台的开发。本文的主要工作如下所示：

1. 阐述整个大学生分享问答平台的分析设计、系统基本架构和模块结构；
2. 分析并使用MVVM三层模式开发项目，总结出自己对MVVC开发模式的一点看法；
3. 开发阶段的技术选择上，我并没有按以前系统设计的经验来，而是选择使用市场上比较前沿的技术开发。系统前端使用Vue，它是一套用于构建用户界面的渐进式JavaScript框架，它出色的性能和组件化的设计让我选择使用它来开发系统前端，为了使得系统变得更加轻量，具有更好的性能。在数据库选择上使用了mongdb，它是一款出色的缓存数据库，同时支持数据的持久化。后台的服务搭建上使用node.js作为技术支持，它能让 JavaScript 运行在服务端的开发平台，同时提供对各种底层组件的支持，比如操作数据库等。这些技术栈能实现系统功能的完善，并保证系统的稳定性。

在这次的完成这次设计的过程中，不但对vue，node，MongoDB进行了学习和掌握，更加发现了自己知识方面的欠缺，在改正的同时加以整理。更加的完善了自己在专业方面的知识积累

# 致谢

本课题在选题及研究过程中得到陈洁老师的指导。感谢陈洁、侯海良等老师在论文工作的各方面给予我的关心、指导和帮助。老师严谨的治学作风、认真的工作态度和勤奋的工作精神深深地影响着我，使我受益终身。在此我对他们表示由衷的感谢。

感谢在大学期间专业老师们对我的指导，让我学到了很多的专业知识，为此次毕业论文设计奠定了基础，还让我树立了正确的价值观，收货很多，是一辈子的财富，学校和领导们给我们创建的学习环境也使论文在一个很好的学术氛围中完成。同时，感谢同学的支持以及学习和生活上的帮助，感谢父母多年来无条件支持帮助我，在生活上处处关照。他们是我一直努力的动力。

最后，向审阅我论文和参加答辩的老师们表示我衷心的感谢，感谢你们抽出宝贵的时候参加我的论文答辩会，感谢你们对论文不当之处提出的宝贵意见和建议。

# 参考文献

1. 梁灏. Vue.js实战[M]. 清华大学出版社,2017.
2. Vue.js. <https://cn.vuejs.org/>
3. 菜鸟教程. <https://www.runoob.com/>
4. 弗[拉纳根](https://book.douban.com/author/1031096/). JavaScript权威指南[M]. 机械工业出版社,2007.
5. 陆凌牛. Node.js 权威指南[M]. 北京机械工业出版社，2014.4.
6. GItHu开源. node.js最佳实践.
7. 沈剑翘,陈泽椿.Vue.js在构建系统前端SPA的应用[J].科技创新与应用,2020(03):181-182.
8. 朱晓阳,刘苑如,范仲言.基于Node.js的学习平台后端系统设计与实现[J].电脑知识与技术,2019,15(13):116-118.
9. 任明飞,李学军,崔蒙蒙,杨双龙,孙小奇.基于MongoDB的非关系型数据库的设计与开发[J].电脑知识与技术,2019,15(34):1-2.
10. 杨锦.关于网页设计与制作方法研究[J].计算机产品与流通,2019(08):210
11. Robert Hoekman,Jr. , [美] Jared Spool. 网站设计解构[M]. 人民邮电出版社,2010-09.
12. 闫安. 虚拟社区中知识获取方式对获取结果的影响研究——以知乎社区为例[J]. 图书馆理论与实践. 2020年.(1):G252
13. 耿瑞利, 申静. A Research on Knowledge Sharing of Social Networking Users: Features, Contents, and Prospects%社交网络用户知识共享研究:特 征、内容与展望[J]. 图书情报知识, 2018, 000(001):16-26.
14. 徐浩东. 网络社群虚拟奖励对用户知识共享意愿的影响研究[J]. 中文信息, 2017(9).
15. 刘立. MVVM模式分析与应用%Analysis and Application of MVVM Pattern[J]. 微型电脑应用, 2012, 028(012):57-60.
16. Smith, Josh. WPF Apps with the Model-View-ViewModel Design Pattern. MSDN Magazine. February 2009.