**基于vue的个人博客的设计与实现**

**姓名：周会艳**

**院系：信息工程学院**

**专业：计算机科学与技术**

**导师：郑颖**

**完成时间：2019-00-00**

[摘要： 5](#_Toc1796912677)

[开发背景 9](#_Toc648834578)

[1.1博客的背景 9](#_Toc40792980)

[1.2系统设计的意义 10](#_Toc560331467)

[系统相关技术介绍 10](#_Toc775173774)

[SPA（Single Page App） 10](#_Toc1709816916)

[2.1vue.js介绍 12](#_Toc1414226705)

[Npm NPM 使用介绍 cnpm加速 13](#_Toc559225939)

[Vue-cli 13](#_Toc1521917501)

[2.2.1声明式渲染 14](#_Toc189719890)

[2.2.1创建实例 14](#_Toc1756459082)

[2.2.1 数据绑定 14](#_Toc1497579512)

[2.2路由 14](#_Toc1310515344)

[2.3状态管理 15](#_Toc1239102976)

[2.2.2 vue响应式原理 15](#_Toc1454792673)

[2.2.5 与其他框架相比优点 15](#_Toc1599134016)

[2.4node.js 15](#_Toc887564707)

[2.4.1Node.js模块系统 16](#_Toc878618487)

[Node.js Express 框架 16](#_Toc843354237)

[2.4Node.js GET/POST请求 16](#_Toc862591059)

[2.3Node.js RESTful API 16](#_Toc2053311363)

[2.4Node.js 连接 MySQL 16](#_Toc2089354298)

[系统分析 16](#_Toc125090742)

[3.1需求分析 16](#_Toc13610381)

[3.23.2 可行性分析 17](#_Toc1116406885)

[系统设计 17](#_Toc885892356)

[系统功能设计 17](#_Toc688702641)

[前台 17](#_Toc88429957)

[后台 17](#_Toc183603575)

[数据库设计 18](#_Toc2038767933)

[概念结构设计 18](#_Toc323578399)

[逻辑结构设计 18](#_Toc953557789)

[第五章 系统实现 18](#_Toc1922785809)

**目录：**

**绪论**

**第一章 开发背景 1**

**1.1 博客背景 1**

**1.2 系统设计的意义 2**

**第二章 系统相关技术介绍 5**

**2.1 vue.js 5**

**2.2 node.js 6**

**2.3 iview 6**

**第三章 系统分析 10**

**3.1 需求分析 10**

**3.2可行性分析 10**

**暂定-字数不够在加上**

**3.3 数据流程分析**

**3.3.1 系统数据流程**

**3.3.2 数据字典**

**第四章 系统设计 12**

**4.2 系统功能设计**

**4.2。1 前台功能设计**

**4.2。2 后台功能设计 12**

**4.3 数据库设计 13**

**数据概念结构设计 15 e-r图表示**

**数据逻辑结构设计 15 E-R模型转换为关系模型。**

**第五章 系统实现 19**

**5.1后台功能模块 19**

**5.1.1用户注册模块 19**

**5.1.2博客管理员首页 22**

**5.1.3 博客文章信息管理页面 25**

**5.2前台功能模块 30**

**5.2.1前台主界面 30**

**5.2.2 “博客评论”功能模块 34**

**5.2.3 友情连接管理页实现过程 37**

**第六章 测试 39**

**6.1测试目的 39**

**6.2测试方法 39**

**6.3测试用例 39**

**第七章 结论 41**

**7.1 不足之处与改进 41**

**7.2 结论 41**

**致 谢 42**

**参考文献 43**

**题目**

# 摘要：

互联网成为了当今世界人们对外交流，快速获取、发布和传递信息的最常用也是最便捷的渠道，互联网在人们生活、工作与学习的方方面面都发挥着重要的作用。而博客网站正适应这种人际与技术交流方式的改变，满足个人技术与信息共享的需求。本文对博客的功能与需求进行了完整分析，设计出了一个简单、易用的个人博客系统。为了提高开发效率和代码复用率，本系统使用采用SPA（单页面应用）思想的vue.js进行系统的开发。Vue.js是一个轻量级的基于MVVM模式的渐进式框架。组件与数据绑定思想，简化了前端开发的复杂度。后端将node.js技术与express结合使用，以创建符合RESTful API设计规范的接口。实现前后端分离，有效的进行了解耦。

**关键词 博客，vue.js ,node.js ,前端框架**

**英文题目**

**有道的：**

**Abstract: the Internet has become the most common and convenient channel for people to communicate with others in the world. The Internet plays an important role in people's life, work and study. Blog sites are adapting to this change in interpersonal and technical communication, to meet the needs of personal technology and information sharing. This article has carried on the complete analysis to the blog function and the demand, has designed a simple, easy to use personal blog system. In order to improve the development efficiency and code reuse rate, This system uses the idea of SPA (single page application) vue. Js to develop system . Vue. js is a lightweight progressive framework based on the MVVM pattern. The idea of component and data binding simplifies the complexity of front-end development. The back end USES node.js technology in conjunction with express to create interfaces that conform to RESTful API design specifications. The front and rear ends are separated and decoupled effectively.**

**关键词**

序言

Blog（博客）作为一种个人知识管理工具，在人们日常生活中越来越盛行

互联网成为人们快速获取、发布和传递信息的重要渠道，它在人们政治、经济、生活等各个方面发挥着重要的作用。通用博客技术的出现使得每个人可以零成本、零维护地创建自己的网络媒体，Blog站点所形成的网状结构促成了不同于以往社区的Blog文化，Blog 技术缔造了“博客”文化。通用博客是当今网络的热点，越来越多的网民开始将自己的想法、感受、心情等以文字、图片或多媒体的形式发布到网络中，以便于与朋友或其他浏览者分享，而博客正是承载这些内容的平台。所以开发出一个高效的通用博客系统，使越来越多的人更能方便、快捷的向别人展示自己。

博客(Blogger)的出现,使我们在互联网世界,第一次有了知识积累和文化指向。使人类由粗放的数字化生存,过渡为个人化的精确的目录式生存。博客们将工作、生活和学习融为一体,通过博客日志(Blog),将日常的思想精华及时记录并发布,萃取并链接全球最有价值、最相关、最有意思的信息与资源。在其形成的博客媒介下,更多的知识工作者能够零距离、零壁垒地汲取这些最鲜活的思想。博客传播的出现,标志着以“信息共享”为特征的第一代门户之后,追求“思想共享”为特征的第二代门户正在浮现,互联网开始真正凸现无穷的知识价值。博客传播是网络传播领域出现的个性鲜明的传播现象,其出现改变了网络传播的秩序,重新划分了网络传播的界限。博客的出现还打破了长久以来传统媒体对媒介所传播内容的控制,使得由美国社会心理学家、传播学四大先驱之一的卢因1947年率先提出的“把关人”理论受到强烈冲击,以至于更有甚者宣称,“守门人”时代结束。一个理论的建立,在某种意义上讲是现实实践的结论升华,它反过来必然要指导实践活动。一个理论的结束,对于现实生活是具有重大意义的。博客技术下发展起来的博客群体、博客媒介和博客传播,是否将重新定义互联网的界限...

博客是以自由、开放和共享为文化特征,通过图文音象等表现形式，围绕个人网络存在的五大功能，提供存取读写、组织沟通、评价交换等服务的一种社会化个人服务模式。它并不是纯粹的技术创新,但确是一种逐渐演变的网络应用,一种形式的变化。从某种意义上说,也是一种计算机源代码开放,它像一部傻瓜照相机一样,让所有使用者都成了程序员,这就大大消解了网络与人的距离,也消解了人与人的距离。博客的全民性,让它的传播方式成为所有人对所有人的传播。然而,博客并不如表面般繁荣,当博客毫不掩饰地在大众面前喧闹的时候,接踵而来的问题使得博客乱了方寸。 本文以非独立博客网站为研究对象,从传者、受者、内容、博客服务平台以及互动反馈等几个方面对博客进行分析,并从中得出了以下结论:博客参与者的盲目性导致了博客行为过程中的迷茫与厌倦;由于进入的零门槛和缺少监管,彻底颠覆互联网既有模式的博客,变成了新的信息垃圾场。博客正遭受低俗肤浅成风、网络侵权等因素的困扰；博客平台在服务、管理以及价值导向上忽略了用户体验；博客间的互动也呈现两个极端的尴尬局面。从总体上说，我国博客正处在一个混乱期,隐藏在虚热表面下的是种种的危机。

## 开发背景

## 1.1博客的背景

Blog是继Email、BBS、ICQ之后出现的第四种网络交流方式，至今为止十分受大家的欢迎，是网络时代的个人“读者文摘”，是以超级链接为入口的网络日记，是代表着新的生活方式和新的工作方式，更代表着新的学习方式。简言之，Blog就是以网络作为载体，简易迅速便捷地发布自己的心得，及时有效轻松地与他人进行交流，再集丰富多彩的个性化展示于一体的综合性平台。

博客的分类有很多。比如，按功能分：基本博客、微型博客；按用户分：个人博客、企业博客；按存在方式分：托管博客、自建独立网站的Blogger、附属Blogger、独立博客。本系统就是按用户划分的个人博客。

博客有以下特点：

1. 简单的操作是博客开发的动力。 这是博客收到众多网友青睐的最大特点。许多博客托管网站使用的口号就是“一分钟内就可以轻松拥有一个博客”。

2、不断更新的功能是博客生活的催化剂。现代社会，信息传递超级快速，更新博客可以比作生物的新陈代谢，生物没有新陈代谢就代表着生命的终结，而博客没有更新，博客也就失去了生命力。

3、开放互动功能是博客传播的推动。网络赋予了博客的开放性，博客也就不再是一个纯粹的私人空间。游客与博主给我们的文章写留言与评论，如果我们对他们进行了回复，并通过链接地址进行回访，就达到了互动效应。因此，利用博客开放互动的特点，可以用于交流推广，形成一个固定的博客朋友圈。

4、展示个性的特点是博客精彩的原动力。博主可以简单地从日志内容，博客界面，文章数量，日志分类体现出博主的个性。

## 1.2系统设计的意义

在科技快速进步的今天，互联网成为了当今世界人们对外交流，快速获取、发布和传递信息的最常用也是最便捷的渠道，互联网在人们生活、工作与学习的方方面面都发挥着重要的作用。当今时代，技术日新月异，作为互联网技术从业者，需要不断的汲取新知识，走在技术的前沿，但是大部分新技术文档与书籍都是英文，对于英语能力有限的互联网技术从业者，不可避免的需要借助搜素引擎或者其他人的帮助，而博客网站正适应这种人际与技术交流方式的改变，迎合个人技术与信息共享的需求。它作为一个技术分享的平台，每一个人都可以在此记录自己学习与工作过程的心得体会。

# 系统相关技术介绍

## SPA（Single Page App）

**单页面是指只有一个主页面的应用，浏览器一开始要加载所有必须的 html, js, css。所有的页面内容都包含在这个所谓的主页面中。但在写的时候，还是会分开写（页面片段），然后在交互的时候由路由程序动态载入。**

SPA 可支持丰富客户端功能，当用户执行操作或在应用的各区域间导航时无需重新加载页面。 SPA 很少需要重新加载整个页面，因此加载速度更快，可在后台提取数据，并且对单个用户操作的响应更快。 SPA 支持增量更新，可保存尚未完成的窗体或文档，而无需用户单击按钮提交窗体。 SPA 支持丰富的客户端行为，例如拖放，比传统应用程序更容易操作。 可以将 SPA 设计为在断开连接的模式下运行，对客户端模型进行更新，并在重新建立连接后将更新最终同步回服务器。 如果应用要求包括丰富的功能，且超出了典型 HTML 窗体提供的功能，则应选择 SPA 样式应用程序。

**单页面是指只有一个主页面的应用，浏览器一开始要加载所有必须的 html, js, css。所有的页面内容都包含在这个所谓的主页面中。但在写的时候，还是会分开写（页面片段），然后在交互的时候由路由程序动态载入。**

**相较于传统网页应用，单页应用将 MVC 前置到了浏览器端：**

**服务端**

**浏览器**

**数据层**

**数据服务**

**控制器**

**视图**

**控制器前置，单页应用将路由处理放在浏览器端，即在浏览器端直接响应浏览器地址的变化，分发到对应的路由，向用户呈现对应的界面。**

**以小块组件为功能元件，类似于传统网页中的 Ajax 组件，单页应用以小的组件为功能元件，在路由变化时，不再刷新整个页面，而是组合这些小的组件，替换变化的部分。**

**数据层前置，与 Ajax 组件一个明显的区别是，单页应用在浏览器端通常有一层实实在在的数据层，而服务端则退化成了完全的数据 API。浏览器端的数据层会封装服务端 API，供上层的视图层调用。**

**看上去有点类似于客户端（Android、iOS）开发，交互全都放在客户端，服务端仅仅提供 API。**

**为了说清楚什么是单页应用，我们先来看看传统的网页应用是什么样子的。**

**浏览器**

**服务器**

**刷新、超链接、表单提交**

**HTML 页面**

**浏览器**

**服务器**

**目前绝大部分的网站都还采用这种形式。单个 HTML 页面作为功能元件，通过刷新，超链接、表单提交等方式，组合排列这些 HTML 页面，来为用户提供服务。作为网页应用的传统形式长久不衰，很多流行的开发框架都以之作为范式设计的。比如 Ruby on Rails，Spring MVC，Express 等等。**

**浏览器**

**服务端**

**前端展示**

**控制器**

**视图**

**数据服务**

**在传统的网页应用中，浏览器更多的是充当一个展示层，路由处理、服务调用、页面跳转流程都由服务端来处理。即 MVC 都放在服务器端，而 V 作为用户界面则通过网络发送到浏览器端，作为 UI 与用户交互。**

**这样的范式有以下特点：**

**重服务端，由于 MVC 都存在于服务器上，因此这类应用在开发资源和开发的重心都偏向后端，往往是后端工程师来主导整个项目开发；**

**页面频繁刷新，由于浏览器端只是一个展现层，当页面功能有所变化的时，页面就刷新，这会导致资源的浪费，用户需要花费额外的时间等待页面刷新，用户体验不佳。**

**当然，上面表达的是一种普遍的范式，而 Ajax 则是这个范式的异类。Ajax 的出现使得网页可以局部更新，使得网页上的一部分可以作为一个功能元件来为用户提供服务。这种形式的网页应用已经具备单页应用的雏型，但并不是标准的单页应用。**

**单页应用的好与坏**

**每种技术都有其利弊，单页应用也是如此。我们先说说好的地方：**

**无刷新体验，这个应该是最显著的有点，由于路由分发直接在浏览器端完成，页面是不刷新，对用户的响应非常及时，因此提升了用户体验；**

**完全的前端组件化，前端开发不再以页面为单位，更多地采用组件化的思想，代码结构和组织方式更加规范化，便于修改和调整；**

**API 共享，如果你的服务是多端的（浏览器端、Android、iOS、微信等），单页应用的模式便于你在多个端共用 API，可以显著减少服务端的工作量。容易变化的 UI 部分都已经前置到了多端，只受到业务数据模型影响的 API，更容易稳定下来，便于提供鲁棒的服务；**

**组件共享，在某些对性能体验要求不高的场景，或者产品处于快速试错阶段，借助于一些技术（Hybrid、React Native），可以在多端共享组件，便于产品的快速迭代，节约资源。**

**单页应用的优点有时候也是缺点：**

**首次加载大量资源，要在一个页面上为用户提供产品的所有功能，在这个页面加载的时候，首先要加载大量的静态资源，这个加载时间相对比较长；**

**较高的前端开发门槛，MVC 前置，对前端工程师的要求提高了，不再是『切切图，画画页面这么简单』；同时工作量也会增加数倍，开发这类应用前端工程师的数量往往多于后端；**

**不利于 SEO，单页页面，数据在前端渲染，就意味着没有 SEO，或者需要使用变通的方案。**

**总结**

**在笔者看来，单页应用是对原来 Ajax 组件的一种延伸，相较于传统的网页应用，将 MVC 前置到了浏览器端，浏览器就相当于 iOS 系统，单页应用就像是应用，与服务端仅仅通过 API 来沟通。每种技术方案都有利弊，是否要使用单页应用技术，需要通过对产品需求（跨平台、SEO、性能、用户群属性？）、资源（工程师配比，能力？）时间要求等的评估来做出选择。**

## 2.1vue.js介绍

单页面开发的优点：

良好的用户体验

用户不需要重新刷新页面，减少TTFB的请求耗时，获取数据也是通过Ajax异步获取，页面显示流畅。

前后端分离

前端负责界面显示，后端负责数据存储和计算，各司其职，不会把前后端的逻辑混杂在一起。

减轻服务端压力

减轻服务器压力，服务器只需要提供API接口，不用管页面逻辑和页面的拼接，吞吐能力会提高几倍。

共用一套后端程序代码，适配多端

同一套后端程序代码，不用修改就可以适用于Web、手机、平板。

单页面开发的缺点：

首屏加载过慢

单页面首次加载，需要将所有页面所依赖的css和js 合并后统一加载，所 以css和js文件会较大，影响页面首次打开时间。

SEO

因为页面数据都是前端异步加载的方式，不利于搜索引擎的抓取。

Vue (读音 /vjuː/，类似于 view) 是一套用于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，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

### Npm NPM 使用介绍 cnpm加速

[http://www.runoob.com/nodejs/nodejs-npm.html#taobaonpm](http://www.runoob.com/nodejs/nodejs-npm.html" \l "taobaonpm)

### Vue-cli

什么是webpack呢？请看以下介绍：

WebPack可以看做是模块打包机：它做的事情是，分析你的项目结构，找到JavaScript模块以及其它的一些浏览器不能直接运行的拓展语言（Scss，TypeScript等），并将其打包为合适的格式以供浏览器使用。

### 2.2.1声明式渲染

### 2.2.1创建实例

### 2.2.1 数据绑定

vuex :

vuex是一种集中式状态管理模式，什么意思呢？我们在模块化开发过程中，我们以组件来作为模块单位，模块之间存在于不同的命名空间，作用域互不干预，这样保证了我们模块之间变量函数名称等不会冲突，但是有时候我们我们需要组件之间共享一些数据或者状态，我们通常的做法是传参，但是传参的做法至少有两个弊端，一是麻烦(尤其是当需要传递的参数很多时)，二是不好管理且冗余(给多个组件传参就需要多份参数列表，而且容易出错)。vuex提供的集中式管理就解决了这个问题，通过把要共享的数据或状态集中起来管理，别的组件需要时就去访问变更，大大提高了可维护性和开发效率

vue-router : vue-router是一个前端路由管理器，这个和后端常听说的路由有些不同(个人觉得)，这里的路由管理器更像是一个组件注册器，vue-router为分散的组件注册一个路由或者叫地址也未尝不可，以方便我们控制组件的层级嵌套关系以及隐藏还是显示，这样我们可以很方便高效的构建单页应用

axios :

axios 和 jquery.ajax/vue-resource一样 ， 都是HTTP异步请求的工具，axios和vue-resource的API很像，但是个人觉得，axios的API更丰富一些

用户在App.vue组件中输入，v-model双向绑定了App组件中的data，App组件中的侦听属性发现data改动，立即向vuex的Store实例提交了状态变更，当用户点击show，App组件中methods.login事件被触发，它向Store实例发起一个分发，并导航到/app，Store的action收到分发调用axios去异步请求位于localhost:80下的php后台服务，得到的登陆状态立即提交，与此同时，Router去渲染/app路由对应的组件到模板中去，于是我们就可以看到一个动态响应式的状态提示了

### 2.2路由

Vue-router

vue-router是Vue.js官方的路由插件，它和vue.js是深度集成的，适合用于构建单页面应用。vue的单页面应用是基于路由和组件的，路由用于设定访问路径，并将路径和组件映射起来。传统的页面应用，是用一些超链接来实现页面切换和跳转的。在vue-router单页面应用中，则是路径之间的切换，也就是组件的切换。

### 2.3状态管理

vuex

### 2.2.2 vue响应式原理

### 2.2.5 与其他框架相比优点

vue很好地规范了其开发生态，开发者只需要关注业务代码本身，而不必操心复杂的webpack配置，可以说是对开发人员尤其是新手是相当友好的。

易用

已经会了 HTML、CSS、JavaScript？即刻阅读指南开始构建应用！

灵活

不断繁荣的生态系统，可以在一个库和一套完整框架之间自如伸缩。

高效

20kB min+gzip 运行大小

超快虚拟 DOM

最省心的优化

## 2.4node.js

简单的说 Node.js 就是运行在服务端的 JavaScript。

Node.js 是一个基于Chrome JavaScript 运行时建立的一个平台。

Node.js是一个事件驱动I/O服务端JavaScript环境，基于Google的V8引擎，V8引擎执行Javascript的速度非常快，性能非常好。

Node.js是一个能够在服务器端运行JavaScript的开放源代码、跨平台JavaScript运行环境。Node.js由Node.js基金会持有和维护[4]，并与Linux基金会有合作关系[5]。Node.js采用Google开发的V8运行代码，使用事件驱动、非阻塞和异步输入输出模型等技术来提高性能，可优化应用程序的传输量和规模。这些技术通常用于数据密集的即时应用程序。

Node.js大部分基本模块都用JavaScript语言编写。在Node.js出现之前，JavaScript通常作为客户端程序设计语言使用，以JavaScript写出的程序常在用户的浏览器上运行。Node.js的出现使JavaScript也能用于服务端编程。Node.js含有一系列内置模块，使得程序可以脱离Apache HTTP Server或IIS，作为独立服务器运行。

### 2.4.1Node.js模块系统

<http://www.runoob.com/nodejs/nodejs-module-system.html>

为了让Node.js的文件可以相互调用，Node.js提供了一个简单的模块系统。

模块是Node.js 应用程序的基本组成部分，文件和模块是一一对应的。换言之，一个 Node.js 文件就是一个模块，这个文件可能是JavaScript 代码、JSON 或者编译过的C/C++ 扩展。

**2.3axios**

### **Node.js Express 框架**

<http://www.runoob.com/nodejs/nodejs-express-framework.html>

### **2.4Node.js GET/POST请求**

<http://www.runoob.com/nodejs/node-js-get-post.html>

### **2.3Node.js RESTful API**

<http://www.runoob.com/nodejs/nodejs-restful-api.html>

### **2.4Node.js 连接 MySQL**

<http://www.runoob.com/nodejs/nodejs-mysql.html>

# **系统分析**

## **3.1需求分析**

任何一个软件项目的开发都是从需求调查开始，博客管理系统也不例外。系统管理员通过前台页面进入后台管理模块后，可以对注册的博客用户进行管理，包括对注册用户的添加、查找、修改和删除等操作；而博客用户通过前台登陆后，可以对自己的博客空间进行管理，包括发布自己的网络日志、收藏个人图片、与相关人员进行交流和沟通以及删除访问者的评论等。

针对博客管理系统的一些需求，总结出如下信息：

1.用户分普通用户、管理用户和超级管理用户。

2.超级管理用户员可以对博客管理员进行管理和设置权限。

3.博客管理员涉及对博客的类型管理、文章管理、图片管理、评论管理、留言管理和通信管理。

4.普通用户可一阅读文章、发表回应、留言和评论。

5.一种文章类型有多篇文章。

6.一种文章有多篇评论。

## **3.23.2 可行性分析**

**可行性分析(Feasibility Analysis)也称为可行性研究，是在系统调查的基础上，针对新系统的开发是否具备必要性和可能性，对新系统的开发从技术、经济、社会的方面进行分析和研究，以避免投资失误，保证新系统的开发成功。可行性研究的目的就是用最小的代价在尽可能短的时间内确定问题是否能够解决。为了确定开发具有可行性，对本系统主要进行了以下四个方面的分析。**

**(1) 经济可行性。经济可行性主要是对项目的经济效益进行评价。本系统对系统开发者来说并不需要太高的成本支出，只是对系统的管理者付出管理报酬即可，而且开发周期不需要太长，节省了人力、物力、财力资源，所以本系统在经济上是可行的。**

**(2) 技术上的可行性。技术上的可行性分析主要是分析技术条件能否顺利完成开发工作，硬、软件能否满足开发者的需要等。在软件方面，本系统采用的是ASP（Active Sever Pages）进行开发，前台网页设计使用的是Dreamweaver MX，而数据库系统采用的是Microsoft Access 2000。通过分析，在软、硬件方面现有工具与环境完全可以实现系统的开发，因此具有技术上的可行性。**

**(3) 时机可行性。时机可行性是分析系统开发时机是否成熟。目前越来越多的应用程序都已经是转向基于Web的开发，并且Internet已经广泛使用，因此系统的设计具有时机可行性。**

**(4) 管理上的可行性：主要是管理人员是否支持，现有的管理制度和方法是否科学，规章制度是否齐全，原始数据是否正确等。系统的开发主要就是为了方便校友信息的管理，补充现在传统管理方式的不足，因此具备了管理上的可行性。**

**综上所述，本系统开发目标已明确，在技术和经济等方面具备可行性，并且投入少、见效快，因此系统的开发是完全可行的。**

# **系统设计**

## 系统功能设计

### 前台

### 后台

## 数据库设计

### 概念结构设计

用户

用户组

文章

文章分类

### 逻辑结构设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用户表 | | | | |
|  |  |  |  |  |
|  |  |  |  |  |

# **系统实现**

**文献：**

[1][李宇](http://nvsm.cnki.net/kns/popup/knetsearchNew.aspx?sdb=CJFQ&sfield=%e4%bd%9c%e8%80%85&skey=%e6%9d%8e%e5%ae%87&scode=39717235;22550618;" \t "http://nvsm.cnki.net/kns/brief/knet).[前后端分离框架在软件设计中的应用](http://nvsm.cnki.net/kns/detail/detail.aspx?QueryID=3&CurRec=3&recid=&FileName=WXHK201817018&DbName=CJFDLAST2018&DbCode=CJFQ&yx=&pr=&URLID=" \t "http://nvsm.cnki.net/kns/brief/_blank)[J].无线互联科技.2018,15(17):41-42.

[2]汪彤.[基于Node.js的图书共享平台的设计与实现](http://nvsm.cnki.net/kns/detail/detail.aspx?QueryID=3&CurRec=7&recid=&FileName=1018111510.nh&DbName=CMFD201802&DbCode=CMFD&yx=&pr=&URLID=" \t "http://nvsm.cnki.net/kns/brief/_blank)[D].北京邮电大学.2018.

[3]梁灏.Vue.js实战[M].清华大学出版社.2017.

[4]麦冬.轻量级响应式框架Vue.js应用分析[J].信息与电脑(理论版).2017,(7):58-59.

[5]刘红卫.利用Node.js开发前后端分离的系统——以图书馆地方文献系统为例[J].天津科技.2018,45(7):67-70.

[6]茆玉庭.基于Node.js和WebSocket的即时通信系统的设计与实现[D].南京邮电大学.2018.

[7]程桂花,沈炜,何松林,张珂杰.Node.js中Express框架路由机制的研究[J].工业控制计算机.2016,29(8):101-102.

[8]王伶俐,张传国.基于NodeJS+Express框架的轻应用定制平台的设计与实现[J].计算机科学.2017,44(z2):596-599.

[9]聂鑫.前端编程与数据库设计的合理运用[J].信息与电脑(理论版).2011,(2):100.

[10]陈帅,关玉蓉.基于Java Web的奖助学金系统设计与实现[J].科技广场.2017,(3):190-192.

[11]李玉.Vue框架的前端交互性能优化解决方案的研究[D].华中科技大学.2017

[12]邹竞莹.Node.JS博客系统的设计与实现[D].黑龙江大学.2016.

[13]旷志光,纪婷婷,吴小丽.基于Vue.js的后台单页应用管理系统的研究与实现[J].现代计算机.2017,(30):51-55.

[14]邓雯婷.基于Vue.js构建单页面GIS应用的方法研究[J].科技创新与应用.2018，（14）：5-7，10.

[15]朱二华.基于Vue.js的Web前端应用研究[J].科技与创新.2017,(20):119-121.

[16]王志任.基于Vue.js的开发平台的设计与实现[D].广东工业大学.2018.