



**本科毕业论文**

|  |
| --- |
| 教研活动管理系统 |
|  |

**何雪欢**

201330810406

|  |  |
| --- | --- |
| 指导教师 | **杨振刚 副教授** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学院名称 |  | **数学与信息学院** | 专业名称 |  | **信息管理与信息系统** |
| 论文提交日期 |  | 2017年4月 21 日 | 论文答辩日期 |  | 2074年5月 6 日 |

摘 要

学校以教学为中心，教学质量是学校的生命线。提高教学质量的关键在于加强对教学过程的管理，教研活动则是教学管理的重要环节。教研活动是强化教学研究的主要手段，是提高教师素质的重要途径，是促进学校教学质量提高的重要保证。

学校每个教研室每学期都会不定期的举办不同类型的教研活动，学校管理层会在期末对这些信息进行收集统计。由于时间间隔长，且这些信息的记录、统计都是人工操作。人工操作的方式存在着效率低、信息反馈及时性产、易出错等问题。

本系统对本校教研活动相关流程进行分析与研究，对基于 WEB 的教研活动管理系统进行设计。本系统采用了 B/S 的体系结构，前后端分离的开发模式，使用非关系型数据库 MongoDB 进行数据存储。本系统的主要功能为教研活动信息的增删改查操作，教研活动的统计，用户管理等功能。

关键词 教研活动 管理系统 B/S MongoDB

Teaching Activity Management System

He Xuehuan

(College of Mathematics and informatics, South China Agricultural University, Guangzhou 510642, China)

**Abstract:** School is teaching- centred, and the teaching quality is the lifeline of school. The key to improve teaching quality is to strengthen the management of teaching process, and teaching research is an important part of teaching management. Teaching research is the main means to strengthen teaching research, is an important way to improve the quality of teachers, and is an important guarantee to promote the improvement of teaching quality.

Each teaching and research section of the school will hold different types of teaching and research activities each semester, the school management will collect information on the statistics at the end of the semester. Due to the long time interval, and the records of these information, statistics are manual operation. There are many problems such as low efficiency, timely information feedback, error prone and so on.

This system carries on the analysis and the research to the school teaching research activity related process, carries on the design to the teaching and research activity management system based on WEB. The system uses the B/S architecture, the development of the separation of the front and rear ends, the use of non relational database MongoDB for data storage. The main function of the system is to add and delete the information of teaching and research activities, the statistics of the teaching and research activities, the user management and other functions.

**Key words:** Teaching activity management system B/S MongoDB

目 录

[1 绪论 5](#_Toc479530517)

[1.1 研究的目的和意义 5](#_Toc479530518)

[1.2 主要研究内容 5](#_Toc479530519)

[2 系统关键技术 6](#_Toc479530520)

[2.1 Vue.js 6](#_Toc479530521)

[2.2 Stylus 7](#_Toc479530522)

[2.3 Echart 8](#_Toc479530523)

[2.4 Webpack 9](#_Toc479530524)

[2.5 Node.js 9](#_Toc479530525)

[2.6 MongoDB 11](#_Toc479530526)

[3 系统需求分析 12](#_Toc479530527)

[3.1 系统整体概述 12](#_Toc479530528)

[3.2 系统功能性需求分析 13](#_Toc479530529)

[3.2.1 总体功能性分析 13](#_Toc479530530)

[3.2.2 活动信息模块分析 13](#_Toc479530531)

[3.2.3统计模块分析 14](#_Toc479530532)

[3.2.4 用户管理模块 15](#_Toc479530533)

[3.2.5 登录模块分析 15](#_Toc479530534)

[3.3 系统非功能性需求分析 17](#_Toc479530535)

[3.3.1 可靠性 17](#_Toc479530536)

[3.3.2 可用性 17](#_Toc479530537)

[3.3.3 安全性 18](#_Toc479530538)

[3.3.4 性能 18](#_Toc479530539)

[3.4可行性分析 18](#_Toc479530540)

[4 系统设计 19](#_Toc479530541)

[4.1 总体设计 19](#_Toc479530542)

[4.1.1 系统整体架构设计 19](#_Toc479530543)

[4.1.2 系统API设计 23](#_Toc479530544)

[4.1.2 系统模块划分 24](#_Toc479530545)

[4.2 数据库设计 25](#_Toc479530546)

[4.3 详细设计 25](#_Toc479530547)

[4.3.1 编辑教研活动信息实现 26](#_Toc479530548)

[4.3.2 统计教研活动实现 27](#_Toc479530549)

[4.3.3 用户信息操作实现 28](#_Toc479530550)

[5 系统实现 29](#_Toc479530551)

[5.1 用户登录实现 29](#_Toc479530552)

[5.2 系统首页界面实现 29](#_Toc479530553)

[5.2 编辑功能的实现 30](#_Toc479530554)

[5.3 统计信息的实现 31](#_Toc479530555)

[5.4 用户管理的实现 32](#_Toc479530556)

[6 结论与展望 34](#_Toc479530557)

[6.1 总结 34](#_Toc479530558)

[6.2 展望 34](#_Toc479530559)

[参 考 文 献 35](#_Toc479530560)

[致 谢 36](#_Toc479530561)

# 1 绪论

## 1.1 研究的目的和意义

计算机技术的快速发展，使得计算机技术应用在各行各业得到广泛的使用。信息技术在全球范围内的广泛应用促进了经济社会的全方位变革，正在改变着人们生存、生活、工作和学习的基本方式。（叶小波，2012）

学校的核心教育工作教学工作，教学工作的正常进行需要教学管理提供保证，教学管理也是教学质量的有效保障。

为了落实教研活动，提高教学质量，学校、学院每年都会对教研活动进行统计。而传统的管理模式是，有关负责人通过纸质方式对活动进行通知、记录。定期统计向上级汇报。这种人工收集的、传统的教研活动管理模式，耗时长，效率低，易出错，已经不适应当今信息化时代的要求。

现代学校教学管理的发展方向应为网络化和信息化。这种管理与以往人工管理无法比拟的优越性，教学管理信息化为教研活动的展开及管理提供了新方式和新思路。（芮勇，2010）

教研活动管理系统能够收集活动信息，帮助人们展示这些活动信息，快速进行统计，简化管理操作，提高办事效率，减轻工作人员繁忙的工作，减少出错概率，是人们有更多的时间学习和享受生活。

## 1.2 主要研究内容

本系统以我校为例，了解我校教研活动管理的具体流程。找出关键的环节，实现信息化、网络化，为教师提供一个方便、易用的网络应用程序。

教研活动管理系统是对学校教研活动进行综合管理的平台，满足于学校管理层、教师。目的是共享学校各种资源，增强教师自身积累，提高员工工作效率，规范工作流程，具有很强的实用性。本系统实现教研活动资源和信息的整合与发布。教师通过本系统发布教研活动信息，领导通过本系统能够对各种活动进行实时管理、统计。

本系统采用的是前后端分离的开发模式，前后端职责更加分明，前端专注于视图层、控制器层的实现，为用户提供简洁、流畅的界面；将准确的数据信息展示给用户。前端开发是最接近用户的工作，需要为用户提供好的交互体验。后台专注于业务逻辑的处理，根据客户端传来的请求，快速、准确的做出处理。

# 2 系统关键技术

本系统是基于B/S 架构的应用，主要开发语言为 JavaScript,使用Vue.js搭建前端 View层及Controller层。 Stylus 对CSS样式进行预处理，webpack对静态资源进行打包、管理；express和nodeJs结合搭建后端 Model 层，进行业务处理等，使用非关系型数据库MongoDB及其框架Mongoose对数据进行管理。

## 2.1 Vue.js

本系统前端方面使用Vue.js搭建。Vue.js 是一个用来开发 web 界面的前端库（尤雨溪，2016）。对于网页，响应速度是其重要指标之一，而响应速度很大一部分取决于页面所应用资源的体积，Vue.js压缩后只有25.11kb，非常的小巧。

为了保持风格统一，本系统具有相同功能的部分都采用相同的代码。如果使用手动或者复制粘贴代码，一旦有小改动，都需要全部修改。这样的方式，容易造成遗漏，相同的部分越多，工作量越大，效率低，不易维护。

组件化的思想：提高复用性，减少工作量，提高效率，降低维护成本。因此组件化能够很好的解决这个问题。对于Vue.js，一切都是组件。例如本系统的编辑组件，包含“编辑”和“删除”两个按钮，用于实现文章的跳转到修改页面和删除功能。被用于首页、活动信息详情以及个人中心。

在Vue.js中，可以一个文件保存一个组件。本系统的编辑组件保存在modify.vue文件中，编辑组件的html、css、js代码全放入这个文件中，当组件有修改时，只需改动该文件，无需多处改动，不用担心遗漏。。

本系统为单页面应用，需要有路由支持切换页面。虽然Vue.js本省不带路由功能，但有vue-router这个可选库配合，提供了细致的路径跳转控制。通过路由功能，我们可以实现各个组件的按需加载，轻松构建单页应用（劳卜，2016）。

## 2.2 Node.js

本系统服务器使用Node.js搭建。Node.js 是谷歌 V8 引擎、libuv平台抽象层 以及主体使用 Javscript 编写的核心库三者集合的一个包装外壳（维基百科，2015）。

对于WEB应用程序，瓶颈在于服务器能够处理的最大并发连接数量。传统的网络服务技术是，新增一个连接就生成一个新的线程，新的线程会占用系统内存，随着连接的增多，最终系统所有内存会被占掉，导致网站无法正常访问。

Node 真正的亮点在于建设高性能，高扩展性的互联网应用——因为它能够处理数万高吞吐量的并发连接（Tomislav Capan，2013）。

Node.js则通过改变连接服务器的方式，很好的解决了高并发量的问题。主要思路是：Node.js是单线程的，使用非阻塞的异步I/O调用，所有连接都有该线程处理。本系统的主要用户为教师与管理员，也就是说，即使是所有用户同时访问本系统，服务器也能够处理，所有用户均能正常访问本系统。

对于多线程服务器，操作数据库时，为了防止多个线程同时操作统一对象，使用锁进行同步操作，防止破坏数据。不小心就会造成死锁，导致进程永远处于封锁状态，浪费大量系统资源，甚至导致系统崩溃。正如前文所说，Node.js是单线程的，事件驱动管理事件队列，根本不需要锁，也不允许使用锁。

## 2.3 MongoDB

由于本系统选择使用Node.js搭建服务器。Node.js十分适合通过对象数据库来查询数据。以JSON格式存储的数据允许Node.js直接处理，不需要纠结数据转换和匹配的问题（Tomislav Capan，2013）。传统的关系型数据库并不符合Node.js的这一特性，但基于BSON（Binary Serialized Document Format）格式存储的数据库，如MongoDB完全符合。

[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq)是基于分布式文件存储的数据库开源项目，属于NoSQL数据库的一种，[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq" \t "_blank)是一个可扩展、高性能的数据库。（吴飞，2014）

与mysql相比，MongoDB对有索引的ID的查询不会比mysql慢，而对非索引字段的查询，则是全面胜出。mysql实际无法胜任大数据量下任意字段的查询（xqeesoom，2012）。

## 2.4 Webpack

本系统为Web App（网络应用程序），随着页面的增多，功能的复杂，需要的js代码会随之增长。Js代码放在一个文件中，逻辑性不强，且难以维护；若分开多文件保存，存在着依赖问题，一旦页面中引入顺序错误，将导致浏览器报错。

这意味着需要适当的组织代码。本系统使用Webpack打包工具，开发时将代码按照逻辑功能分成不同的模块，由Webpack管理模块之间的依赖关系。

本系统为SAP（单页面应用），SAP其中一个缺点是初次加载耗时相对长，这是因为所有页面代码都一次加载的，之后使用缓存，导致初次加载数据量相对较大。Webapck讷讷够将模块分割成很多小模块。而分割后的模块只有在需要的时候才会被请求。所以初始的请求不会包含所有的代码，从而减小传输压力（chenyiqiao，2016）。

# 3 系统需求分析

## 3.1 系统整体概述

传统的教研活动管理方式，使得信息分散、查询不便等弊端；本系统作为一个信息综合管理平台，通过互联网的方式为管理层与教师提供服务，系统需简单易用，能够提高效率，节省时间。因此本系统的主要目标为：

1. 界面友好：界面简单、清晰，能让用户以最短的时间获取他们想了解的信息，功能明确，无Not Found 页面；
2. 响应式设计：能够根据多种终端设备，调整自身，减少滚动；
3. 角色管理：根据用户的不同等级，拥有不同的权限和系统使用范围；
4. 信息管理：根据用户的不同等级，对文章的增删改查具有不同操作范围，保障了信息的安全性、可靠性。
5. 提高管理效率：分类详细，方便管理、选择；将数据转化为可视化的图表，便于决策分析。

## 3.2 系统功能性需求分析

### 3.2.1 总体功能性分析

由于不同用户（系、学校等各级管理部门）对信息的需求范围、内容、使用频率等有所差别。管理层为了决策分析，需要一定范围的、综合性强的信息，如对教研活动按时间、类型，进行分类统计；同时也需要对教研活动进行管理，如发布、修改、删除活动信息等。教师为了参与活动，了解绩效是否达标，需要具体的内容，如活动的详细信息，以及自身的参与记录。因此，本系统需对用户进行权限管理，只有管理员可对教研活动信息进行操作，不同等级管理员能够统计不同范围所举办的教研活动情况。

### 3.2.2 活动信息模块分析

活动信息模块是本系统的主要功能之一，为所有普通用户提供查看已发布的活动信息功能；还为管理员提供增删改功能，同时又根据管理员的等级提供不同范围的增删改查功能：

1. 发布新活动信息：

创建新的活动信息，并存储于数据库中。

1. 编辑活动信息：

可在首页、活动信息详情进入编辑状态。对自己已发布的活动信息进行修改，如上传活动照片、添加活动参与者等；

普通管理员可对所属教研室发布的活动信息进行修改，系统管理员可对所有活动信息进行修改。

1. 显示活动信息：

首页显示所有活动的简要信息，包括标题、举办时间，内容详情页面显示活动信息的详情。

1. 删除活动信息：

普通管理员可删除自己以及所属教研室已发布的活动信息，系统管理员可删除所有已发布的活动信息。

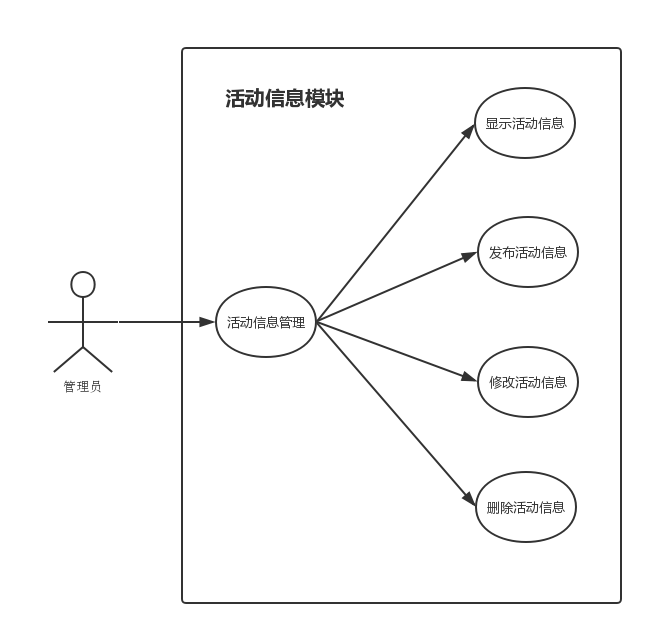


图3 活动信息管理模块用例图

### 3.2.3统计模块分析

本系统的另一个主要功能是为用户提供对教研活动的统计管理。

统计功能按分类查看，包括类型分类（教学讨论会、科研研讨会、学术沙龙）和时间分类（不同年份的全年、上半年、下半年）。

客户端从服务器中获取数据后，对数据进行可视化处理，以图表以及表格的形式展示，并可下将图表载为图片保存于本地。

其中所有用户均可查看自己参加的教研活动统计信息，普通管理员还能查看所属教研室举办的教研活动的统计信息，系统管理员能查看所有教研室教研活动的统计信息。

### 3.2.4 用户管理模块

本系统主要提供给管理层以及教师使用，需要管理员提供对用户的管理功能，包括增删改查（CRUD）。

其中，普通管理员能够对所属教研室的用户进行增删改查，系统管理员能够对所有用户进行增删改查。

### 3.2.5 登录模块分析

为防止非法用户对本系统的恶意、无意修改，保障系统安全。未登录的用户，仅能浏览首页的信息及具体活动信息详情。登录后，系统根据用户的等级，在导航增加不同的功能选项：

（1）普通用户：

1. 个人中心：可查看个人信息，以及修改密码。
2. 统计：可查看自己参与教研活动的情况。

（2）普通管理员：

* + 1. 个人中心：可查看个人信息，参与教研活动的情况，以及修改密码。
    2. 发布：编辑新的活动信息。

② 统计：查看自己以及所在教研室所举办的活动总体情况。

1. 用户管理：可对所在教研室用户进行增删改查管理。
2. 系统管理员：
   * 1. 个人中心：可查看个人信息，参与教研活动的情况，以及修改密码。
     2. 发布：编辑新的活动信息。

② 统计：查看自己以及校内所有教研室举办活动总体的情况。

1. 用户管理：可对所有用户进行增删改查管理。

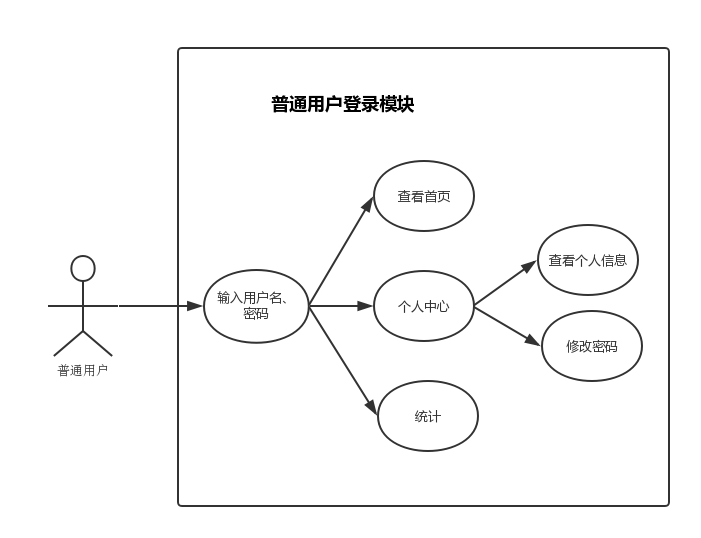


图4 普通用户登录用例图

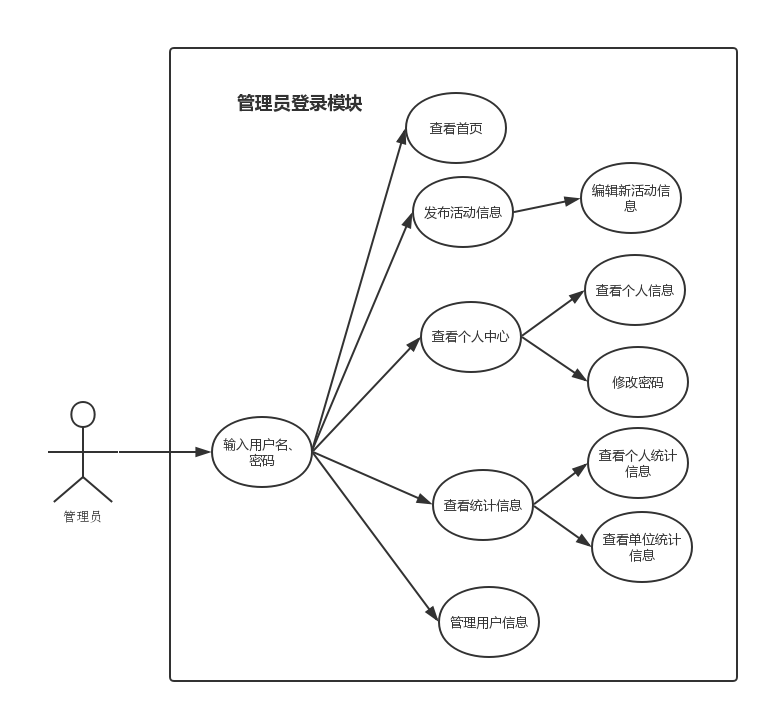


图5 管理员登录用例图

## 3.3 系统非功能性需求分析

### 3.3.1 可靠性

当系统出现异常情况时，为用户提供友好的提示信息，或采取相应的解决方法。当客户端发送请求后，长时间未收到返回数据时，有明确的提示信息，避免用户以为系统“假死”。当提交信息时，禁止用户重复提交；并在收到返回数据时，给予用于恰当的提示信息，避免数据丢失。用户操作不规范、数据库查询出错等，后台返回明确的原因。

### 3.3.2 可用性

本系统功需界面友好，简单易用。真正能够帮助教师提高办事效率，节省时间，降低错误，减轻工作负担。功能上分类明确，在必要的地方提供友好的提示信息。

### 3.3.3 安全性

系统无论大小，都需要保障系统数据安全。

登录密码用于用户身份验证，防止用户信息被盗用、伪造信息。因此本系统在将用户信息插入数据库之前，对用户密码进行加密处理，将密文替换原有密码存储于数据库中。这样即使是数据库管理员也不知道也不知道所有用户的密码。

登录之后，系统将本次会话的sessionId放在cookie中，发送给浏览器；当系统接收到客户端请求，通过请求中的cookie信息进行身份验证，用户在有效期内无需登录。

【系统须保障用户数据以及信息的安全。登录以及添加用户时，对用户密码进行加密处理。发布、修改教研活动信息时，对提交的内容进行过滤，防止 XSS 、CSRF攻击。系统对用户权限分类进行严格，用户只能拥有与等级相符的操作。对表单输入信息进行验证，确保用户输入合法，防止用户信息及教研活动信息被恶意篡改、删除。】

### 3.3.4 性能

为保障用户体验，当访问本系统不存在的地址，或者在教研活动信息详情中删除该信息，页面自动重定向到首页。

当出现请求错误时，如登录，登录信息有错，会提示账号不存在或者密码错误等。

## 3.4可行性分析

（一）经济可行性

经济可行性主要对开发本系统进行的经济合理性综合分析。本系统为毕业设计系统，投入的人力、物力较小，但需要大量的时间与精力，对本系统进行界面设计、数据流分析、数据库表设计等。但由于传统的教研活动管理方式，采用的均为人工操作，流程繁琐，且耗时长，容易造成数据丢失、出错的情况。通过本系统，对信息进行收集、统计，实现网络化综合管理，有助于提高办事效率，减少人员工作量，降低出错率，也避免了人工管理的诸多弊端。因此，本系统在经济上是可行的。

（二）技术可行性

本系统采用了SPA （单页面应用）。单页面应用是仅包含一个网页的应用，所有业务功能都是它的子模块。为用户提供了更接近一个本地应用或桌面应用程序的体验。其优点在于：

（1）更好的用户体验

前端部分几乎都是静态文件，且数据层和 UI 层分离，通过利用浏览器的缓存机制，内容发生改动，页面只需加载局部页面，使得访问更加流畅、快速。

（2）高效

服务器只需要对数据进行管理，无需管展示逻辑与页面合成，减轻服务器压力，提高吞吐能力。

（3）前后端分离，逻辑清晰

分离前后端，前端负责界面显示，后端负责数据存储和计算，各司其职，逻辑分明，同时也能减少带宽的消耗。前后端分离，同一套后端代码，能够同时为多种客户端提供服务，便于以后的扩展开发与维护。

本系统对于硬件要求低，设计上为了简单易用，技术是可行的。

# 4 系统设计

## 4.1 总体设计

为了使用户能够快速、便捷的访问教研活动管理系统，本系统采用 B/S 架构设计。用户可随时随地通过互联网，借助浏览器访问教研活动管理系统。无需安装软件，方便用户的查询，以及管理员的管理操作。

由于学校存在多个学院，每个学院分为多个系，每个系有包含一个以上的教研室。而教研室每年都需要举办一定次数的教研活动，数据信息庞大，需要对数据进行合理有效的管理，才能提高工作效率。

同时需要保证教研活动信息透明度，不同用户之间信息需对称，方便教师参与教研活动，减少错误率。

### 4.1.1 系统整体架构设计

本系统采用前后端分离的开发模式，前端负责视图层、控制器层，后台负责模型层、业务处理等。前后端分离使得各端的职责分明，提高开发效率，降低维护成本。

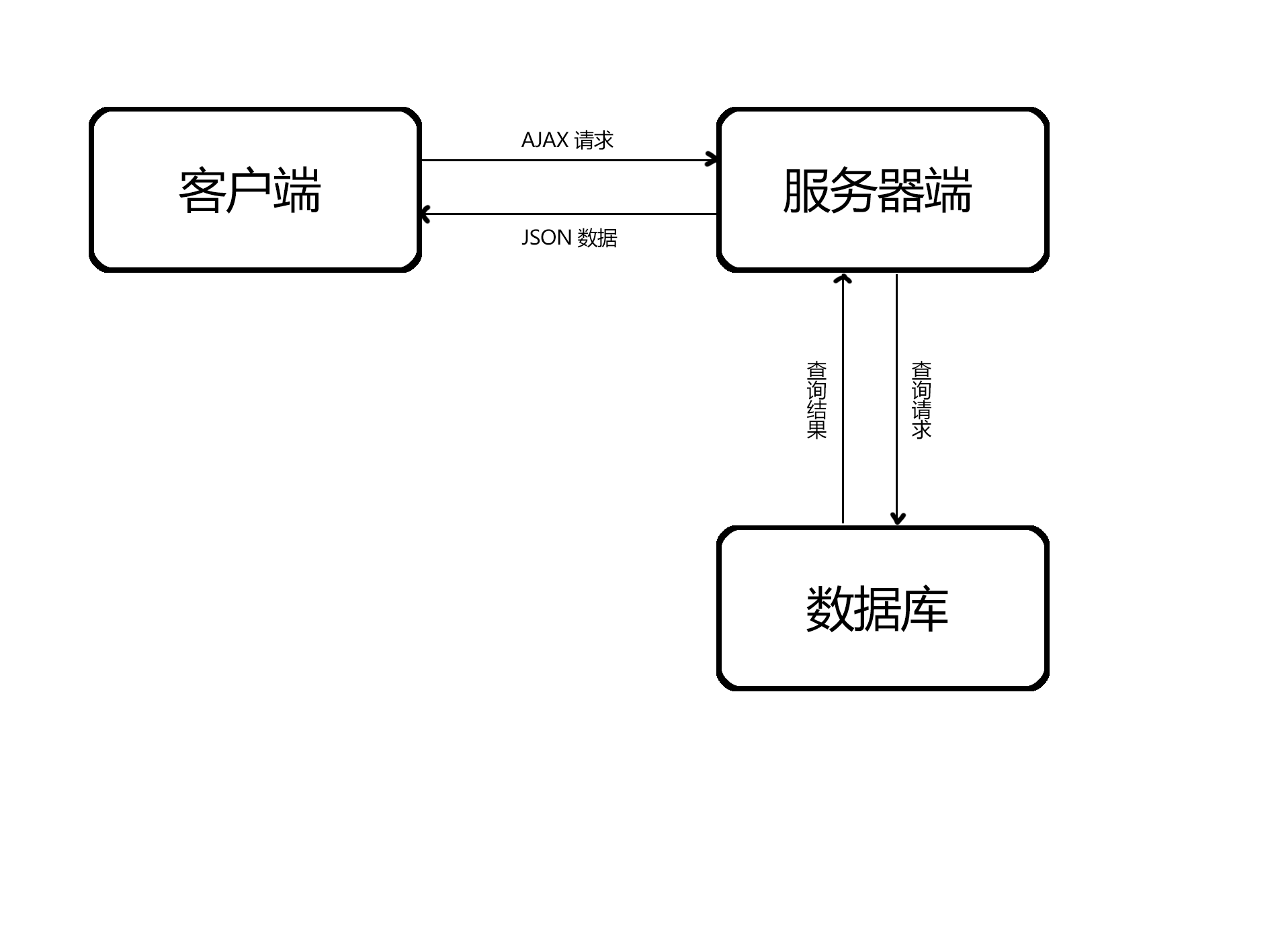


图6 系统数据流图

客户端（即浏览器）向服务端（即Node服务器）发送请求，服务端对数据库（即[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq" \t "_blank)）进行增删改查操作，并对数据库返回的结果做响应的处理，再传递给客户端。

前端采用 MVVM 架构模式，核心是通过提供View和ViewModel的双向数据绑定，使数据变化映射成页面更新。前端使用的 Vue.js 就是注重 ViewModel 的典型的 MVVM框架，其专注于 View 层， ViewModel负责连接View和Model，保证了视图和数据的一致性，这种轻量级的架构使前端开发更加高效、便捷。

基于MVVM架构模式，进行组件化开发，将各个模块拆分为多个组件，即降低了模块之间的耦合度，同时也提高了组件的重用性。

本系统分为四层：视图层、控制器层、业务逻辑层、模型层。

1. 视图层（View）

主要负责系统的用户界面显示，与用户进行交互，将用户操作信息传递给控制器层。本系统视图层包含七个视图，分别为：

* 1. Index：首页视图，展示全部教研活动信息。
  2. Article：详情视图，展示具体教研活动详细信息。
  3. Edit：编辑视图，用于创建、修改教研活动信息。
  4. Statistics：统计视图，展示教研活动统计信息。
  5. Personal：个人中心视图，展示个人信息。
  6. Signin：登录视图，输入登录信息。
  7. User：用户管理视图，展示用户信息，对用户信息进行增删改查操作。

视图可看作是父组件，根据展示要求，包含多个子组件；不同视图可能包含一个或多个相同的组件，提高组件的重用性。

1. 控制器层（Controller）

负责接受视图层传来的应用户操作信息，并作出响应；与后台进行交互，将后台传递的数据进行处理，返回给视图层。控制层包含三个模块：

* 1. Query controller：用于处理用户的查询操作，如教研活动信息分类、教研活动详情、统计信息、用户信息等的查询。
  2. Form controller：用于处理用户提交的表单数据，如创建新的教研活动信息、添加新用户，修改已有教研活动信息、用户信息，登录登出等。
  3. Chart controller：用户处理统计信息，绘制出相应的图表，或展示对应的表格。

1. 业务逻辑层

负责业务逻辑的处理。服务端为客户端提供 API，接受来自客户端的请求，进行相应的业务处理，如查询数据库，过滤输入信息等。

1. 模型层（Model）

负责对数据建模，存储用户数据和业务数据。本系统采用 MongoDB基于文档的非关系型数据库，以文档格式存储数据。模型层包含了个模型：

* 1. Coding model：存储学院-教研室对应关系信息，便于数据库进行信息筛选。
  2. Unit model：存储学校学院、教研室相关信息。
  3. User model：存储用户相关信息。
  4. Information model：存储教研活动相关信息。
  5. Type model：存储教研活动类型相关信息，便于数据库进行分类筛选。
  6. The link model：存储页面底部外链相关信息。

系统整体架构图如下：

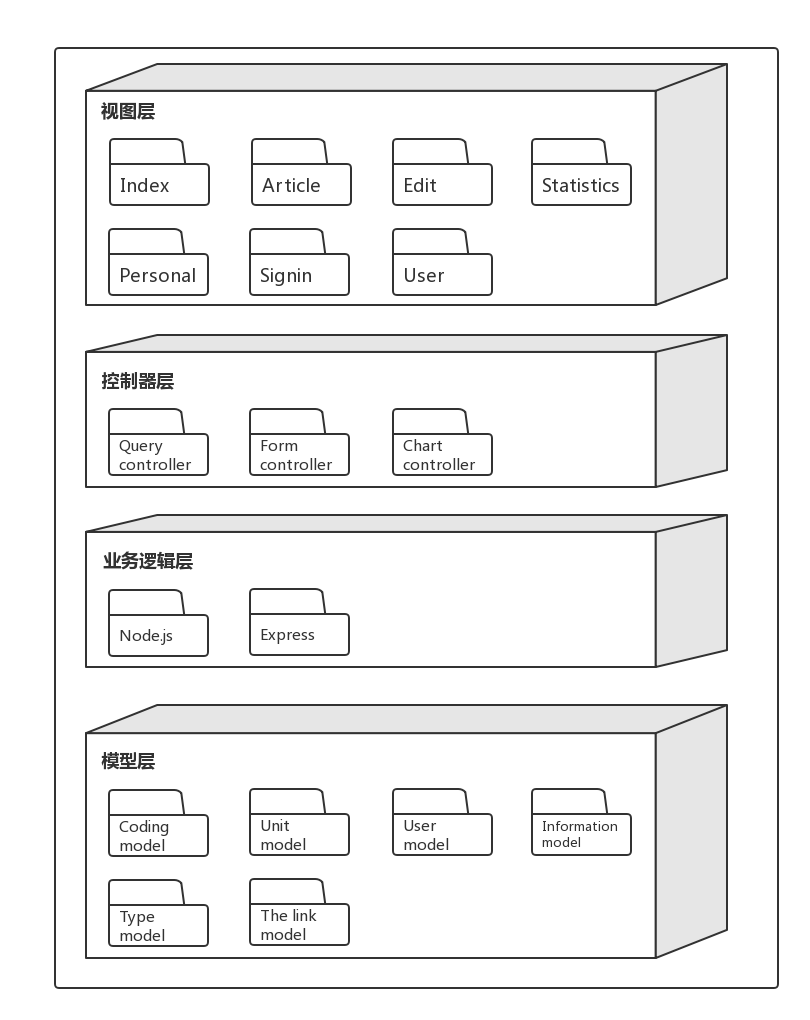


图7 系统整体架构图

### 4.1.2 系统API设计

本系统采用REST 风格的API。本系统的API设计思想为：

1. 将涉及的实体抽象成资源，即按 id 访问资源;
2. 使用 HTTP 动词对资源进行增删改查（CRUD）：get->查，post->增，put->改，delete->删;

根据功能模块提供不同的 API 接口：活动信息模块接口、统计模块接口、用户管理模块接口、登录模块接口、其它相关功能接口：

表1 活动信息模块接口表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 接口描述 | 接口定义 | 请求方法 |
| 活动信息 | 查询活动详情 | /article/:aid | GET |
| 创建新活动信息 | /edit/create | POST |
| 修改活动信息 | /edit/modify | PUT |
| 删除活动信息 | /edit/delete/:id | DELETE |
| 获取修改的活动信息 | /edit/article/:aid |  |

表2 统计模块接口表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 接口描述 | 接口定义 | 请求方法 |
| 统计信息 | 分类查询 | /count/:id/:tab/:year/:time | GET |
| 获取单位信息 | /count/unitText | GET |

表3 用户管理模块接口表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 接口描述 | 接口定义 | 请求方法 |
| 用户管理信息 | 获取用户信息 | /userManage/info | GET |
| 添加用户信息 | /userManage/add | POST |
| 修改用户信息 | /userManage/modify | PUT |
| 删除用户信息 | /userManage/delete/:id | DELETE |

表4 登录模块接口表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 接口描述 | 接口定义 | 请求方法 |
| 登录 | 登录信息 | /signin | POST |
| 登出信息 | /siginout | GET |
| 获取所有活动信息 | /articles/all | GET |

表5 个人中心模块接口表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 接口描述 | 接口定义 | 请求方法 |
| 个人信息 | 获取个人信息 | /userManage/selfInfo | GET |
| 修改密码 |  | PUT |
| 获取个人发布的活动信息 | /articles/user/:uid |  |

表6 其它相关功能接口表

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 接口描述 | 接口定义 | 请求方法 |
| 其它相关功能 | 获取页脚链接信息 | /getFooterLink | GET |
| 获取首页活动类型信息 | /getTypeLists | GET |
| 获取首页学院信息 | /getAcademyLists | GET |
| 获取首页教研室类型信息 | /getFacultiesLists/:id | GET |
| 获取统计单位可选项 | /count/unitText | GET |

由于JSON是在前后端都得到很好的支持的数据交换格式，轻量、易处理、易理解，因此本系统的数据均以JSON文件格式进行传输。

### 4.1.2 系统模块划分

系统功能结构分为四个模块：登录模块、活动信息模块、统计模块、用户管理模块。

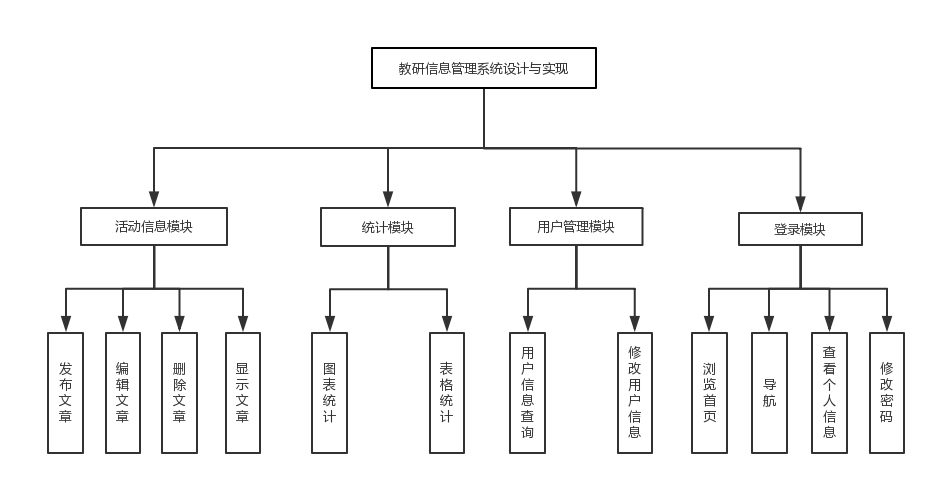


图8 系统功能结构图

## 4.2 数据库设计

MongoDB 与关系型数据库的建模有很多的不同。MongoDB 支持内嵌对象和数组类型。因此 MongoDB 建模有两种方式，一种是内嵌，另一种是连接。本系统的总体设计原则为：

一对很少（one-to-few），使用内嵌；如学院包含的教研室；

一对很多（one-to-many），使用间接引用；如每个教研活动有多个参与者；

双向关联， one 端和 many 端同时保存对方的引用；

## 4.3 详细设计

根据本系统对模块的划分，下面将对本系统关键流程进行详细描述。通过流程图描述用户与系统之间的交互。

### 4.3.1 编辑教研活动信息实现

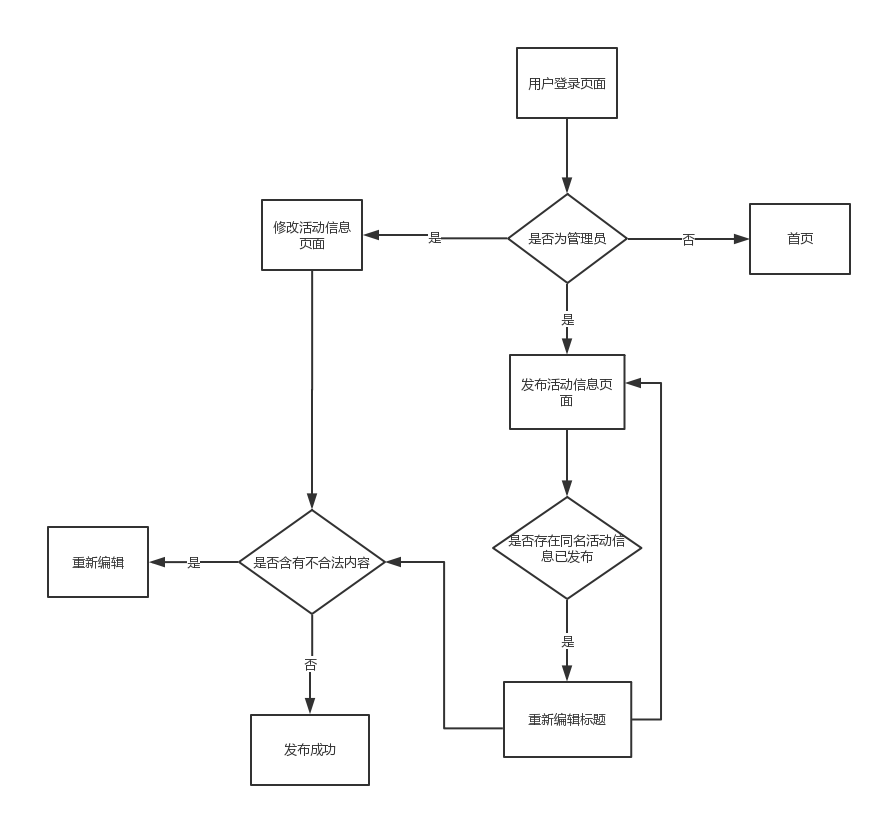


图9 发布教研活动信息流程图

编辑功能分为发布和修改教研活动信息。用户均需登录后，系统通过用户等级判断用户是否有权限发布教研活动信息。管理员均可发布或修改信息。

发布信息：管理员编辑好需要发布的教研活动信息，提交请求后，系统查询同一用户是否已发布同名活动信息。是，则给予用户相应信息，提示用户修改标题。否，系统过滤发布内容，是否含有不合法信息；是，则提示用户重新编辑；否，则插入数据库，插入成功后提示用户发布成功。

修改信息：管理员编辑好信息后，提交请求。系统过滤发布内容，是否含有不合法信息；是，则提示用户重新编辑；否，则插入数据库，插入成功后提示用户发布成功。

### 4.3.2 统计教研活动实现

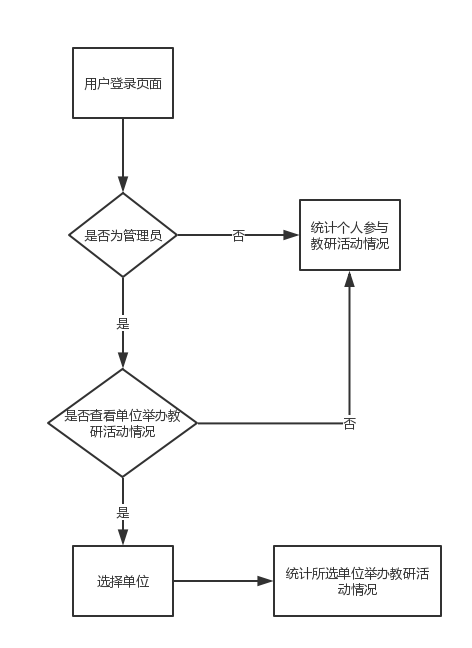


图10 统计教研活动流程图

所有用户登录后，均可在导航统计选项中查看统计信息。

普通用户只能查看自己参与教研活动的统计信息；

普通管理员可查看自己参与，以及所在教研室举办的教研活动的统计信息；

系统管理员可查看自己参与，以及所有教研室举办的教研活动的统计信息。

### 4.3.3 用户信息操作实现

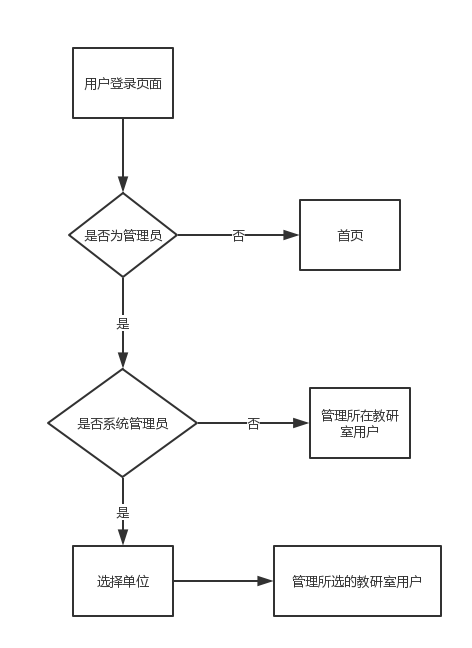


图11 用户信息操作流程图

用户登录后，系统根据用户等级判断权限。

普通管理员只能操作所在教研室的用户；

系统管理员可操作所有教研室举办的用户。

# 5 系统实现

## 5.1 用户登录实现

为了方便使用，本系统采用集中登录的方式。用户与管理员在同一登录页面登录。登录页面没有设置忘记密码选项，如果发生忘记密码的情况，只能寻找管理员。



图12 系统登录页面

登录成功后，系统进行权限判断，在导航添加响应的功能模块。



图12 未登录导航栏



图13 普通用户导航栏



图14 管理员导航栏

## 5.2 系统首页界面实现

系统首页是对所有教研活动信息的展示，为了增强系统的用户友好性，未登录的用户也能浏览。用户登录成功后，会自动跳转到系统首页。

首页默认展示所有的教研活动信息。为了让用户能够快速找到希望的信息，在顶部有分类工具组件，能够按教研活动信息所属的学院、教研室、类型，进行分类查看。

活动列表按举办的开始时间降序排列。未举办的活动，用醒目的标签标出，方便用户快速寻找能够参与的教研活动。



图15 系统首页

## 5.2 编辑功能的实现

编辑功能分为创建新的教研活动信息和修改已发布的教研活动信息。编辑页面的主要功能为编辑信息，因此界面设计尽量的简洁，使用户能够专注于信息编辑，不被干扰。

管理员可通过导航栏的“发布选项”，进入到发布页面，对需要发布的教研活动信息进行编辑。



图16 系统编辑页面发布状态（上）

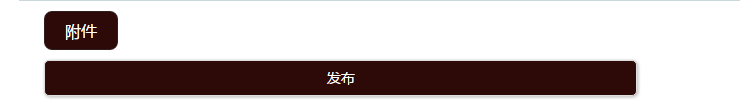


图17 系统编辑页面发布状态（下）

管理员可在首页，或者教研活动信息详情标题右下方，找到编辑组件，编辑或者删除教研活动信息。点击“编辑”，进入教研活动信息修改状态。



图18 教研活动信息详情页面

编辑状态与发布状态为同一页面，但系统会自动导入需要编辑的教研活动的信息。



图19 教研活动信息编辑状态

## 5.3 统计信息的实现

本系统所有已登录用户均可使用统计功能。普通用户仅能统计自己参与的教研活动信息情况。管理员还能根据权限，通过选择，统计相应单位举办教研活动信息情况。

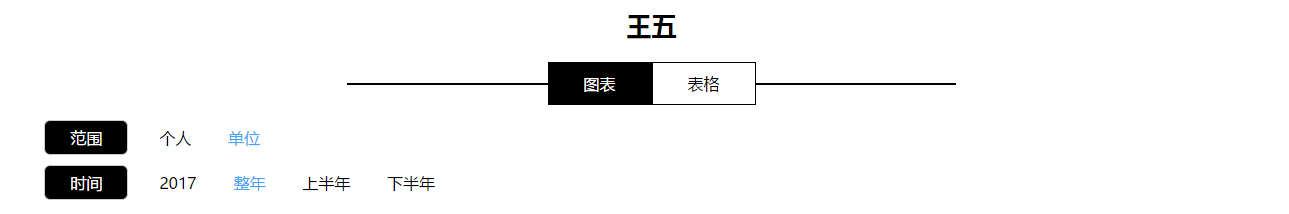


图20 管理员统计页面选择组件

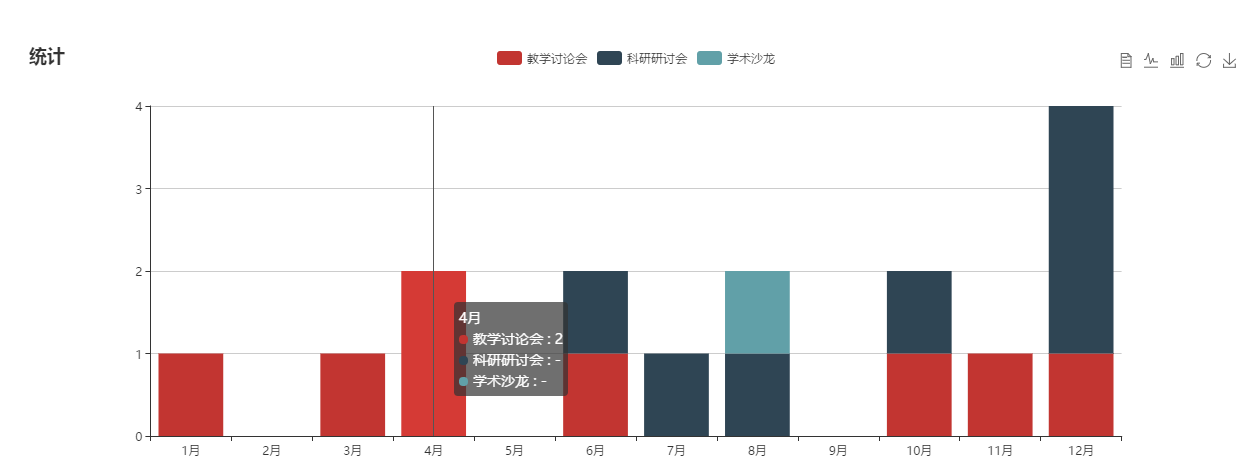


图21 统计页面柱状图

在柱状图中，右上角有一个工具条，可选择将柱状图转换为折线图、表格，还能下载保存为本地图片。

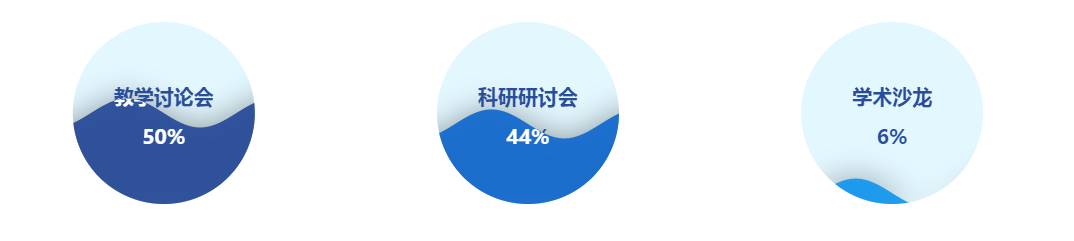


图22 统计页面水球图

## 5.4 用户管理的实现

本系统用户分为普通用户、普通管理员和系统管理员三种不同的角色。用户管理功能只有管理员级别的用户才有权限进行操作。

普通管理员能对所在教研室的用户进行增删改查（CRUD）操作，系统管理员能对所有用户进行增删改查（CRUD）操作。

用户管理中，系统根据管理员的等级，向管理员展示权限范围内的所有用户信息（包括用户名称，用户职称，用户所属系，以及用户等级）；允许管理员对这些用户信息修改；普通管理员只能添加以及删除所在教研室的用户，系统管理员可选择教研室并添加或删除用户。添加用户时能都用户进行权限分配，默认为普通用户。

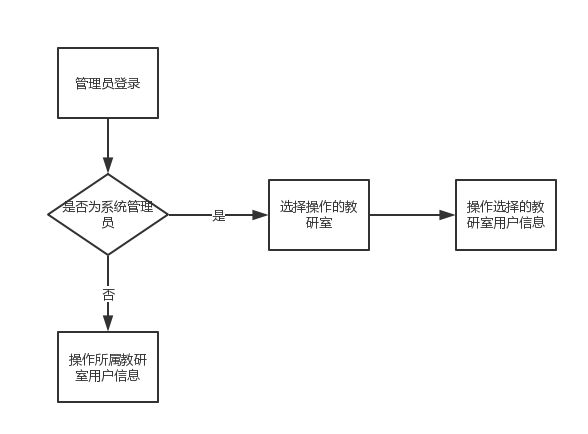


图23 用户管理流程图



图24 添加用户信息页面

# 6 结论与展望

## 6.1 总结

本系统前端使用现今流行MVVM架构设计，Vue.js 框架，以组件化的形式搭建页面，提高重用性；页面使用了响应式设计，适应不同尺寸的屏幕。后台使用了与前端语言一致的 Node.js 语言，按照REST风格设计接口。数据库使用非关系型数据库MongoDB进行数据存储，Mongoose定义数据模板，完成了整个项目。

整个系统最终使用户可以查看所有的教研活动信息，全面了解学校的教研情况，了解自己过去参与教研活动的情况，便于用于对自己过去积累的总结。管理员可以随时随地的发布、修改教研活动信息；还能对使用本系统的用户进行管理；查看教研活动的举办情况，提高管理员的效率，以可视化的展示形式，便于管理员决策。

在系统开发中，本人主要完成了项目的设计原型图设计、UI设计，前端架构的搭建，后台API设计，以及数据库建模。这次开发系统，使我积累了更多的项目经验，学到许多新的东西，也有许多新的体会，对本人今后的发展有着很重要的意义。

## 6.2 展望

本系统实现了查询、发布、修改、删除教研活动信息，统计信息，用户管理这些基本功能。但在发布教研活动的时候对于图片的上传没有进行一个合理的处理。现在的处理方式是直接转化为base64格式，这样在展示教研活动详情的时候，能够减少HTTP请求次数；但当教研活动信息图片多的时候，传输的数据量就会增大很多，传输时间会增加，用户体验不够好。

本系统使用了浏览器尚未完全支持的的ES6语法，需要通过转换器编译成浏览器支持的ES5；且打包工具生成的代码通常会出现臃肿的情况，代码有待改善，减少冗余。

还有许多的页面细节可以进行优化，以增加用户的使用体验，能够得到用户的肯定。

参 考 文 献

江庆，叶浩荣. Vue\_Webpack框架在银行App前端开发的应用\_江庆[J].专家视点，2016，（11）：15-19

彭娜. 基于Node\_js高并发web系统的研究与应用\_陈瑶[D].大连：大连理工大学，2013

芮勇. 浅议信息化条件下的教研活动管理 [J].中国教育信息化·基础教育,2010,(3):40-43

孙思源. 基于MongoDB的网站日志分析系统的设计与实现[D].北京：中国地质大学，2014

王光磊.MongoDB数据库的应用研究和方案优化[D].北京：北京邮电大学网络教育学院，2011

王金龙，宋斌，丁锐. Node\_js\_一种新的Web应用构建技术[D].南京：南京理工大学，2014

王子毅，张春海. 基于ECharts的数据可视化分析组件设计实现[J].图片与多媒体，2016，（14）：46-48

王雄兵，田茂，范亮等.基于ECharts光伏发电信息化管理平台数据可视化效果的实现[J].物联网技术，2017，（1）：54-58

魏云申. 基于WebGL的全景3D漫游系统的设计与实现\_魏云申[D].南京：南京大学，2016

吴飞.基于MongoDB的LBS数据管理系统关键技术研究[D].福建：福建省基础地理信息中心，2014

叶小波. 基于WEB的招生管理系统的设计与实现[D].广州：中山大学，2010

Hanen Abbes，Faiez Gargouri. Big Data Integration: A MongoDB Database and Modular Ontologies based Approach[J]. Procedia Computer Science,2016(96)446-455

Jongseong Yoon .Forensic investigation framework for the document store NoSQL DBMS: MongoDB as a case study[J]. Digital Investigation,2016(17):53-65

M Duque，E Cando，A Aguinaga.Micro-grid platform based on NODE.JS architecture, implemented in electrical network instrumentation[J].IOP Science.2016(10):1-8

Titus. Journal of Systems and Software[J].ELSEVIER.2017(127):28-51

Milorad Pantelija Stevic. Enhancing the management of unstructured data in e-learning systems using MongoDB[J]. Program.2013(49):91-114

陈屹峤. 我们为什么需要Webpack？[DB/OL]. 西安：github，2016：https://chenyiqiao.gitbooks.io/webpack/content/motivation\_of\_webpack.html

劳卜. 浅谈Vue.js[DB/OL].杭州：segmentfault，2016. <https://segmentfault.com/a/1190000004704498>

尤雨溪. Vue.js：a (re)introduction[DB/OL].杭州：知乎，2016：https://zhuanlan.zhihu.com/p/20302927?columnSlug=evanyou

Zqeesoom. mongodb与mysql相比的优缺点[DB/OL].北京：新浪博客，2012：http://blog.sina.com.cn/s/blog\_966e430001019s8v.html

Wikipedia.Node.js[DB/OL]. America.:Wikipedia,2015.https://en.wikipedia.org/wiki/Node.js

Tomislav Capan. Why The Hell Would I Use Node.js? A Case-by-Case Tutorial.[DB.OL]. Croatia:developers,2013. <https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>

致 谢

首先，非常感谢我的毕业指导老师杨振刚，在毕业设计选题以及开发过程，都是在导师的悉心指导下完成的，老师给予了很多的帮助以及指导。导师渊博的专业知识，严谨的治学态度，精益求精的工作作风，诲人不倦的高尚师德，严以律己、宽以待人的崇高风范，朴实无华、平易近人的人格魅力对我影响深远。不禁使我树立了远大的学术目标、掌握了基本的研究方法，还使我明白了许多待人接物与为人处事的道理。

本论文的顺利完成，离不开各位老师、同学和朋友的关心和帮助。同时也要感谢大学四年中，我的授课老师给我学习上指导和帮助，以及同学们给予我的关心、支持和帮助。在此表示深深的感谢，同窗之间的友谊永远长存。

最后，衷心的感谢各位评委老师百忙之中抽空审阅我的毕业论文，希望各位评委老师能够给予批评和指导。