



**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 |  | 2017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加、删除、修改、查询功能，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 and Research Activities 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 for teaching and research activities to increase the information, delete, modify, query function, the statistics of the teaching and research activities, the user management and other functions.

**Key words:** Teaching and Research Activities management system B/S MongoDB

目 录

[1 绪论 1](#_Toc480279714)

[1.1 研究的目的和意义 1](#_Toc480279715)

[1.2 主要研究内容 1](#_Toc480279716)

[2 系统关键技术 2](#_Toc480279717)

[2.1 Vue.js 2](#_Toc480279718)

[2.2 Node.js 3](#_Toc480279719)

[2.3 MongoDB 3](#_Toc480279720)

[2.4 Webpack 4](#_Toc480279721)

[3 系统需求分析 4](#_Toc480279722)

[3.1 系统整体概述 4](#_Toc480279723)

[3.2 系统功能性需求分析 5](#_Toc480279724)

[3.2.1 总体功能性分析 5](#_Toc480279725)

[3.2.2 活动信息模块分析 5](#_Toc480279726)

[3.2.3统计模块分析 6](#_Toc480279727)

[3.2.4 用户管理模块 6](#_Toc480279728)

[3.2.5 登录模块分析 7](#_Toc480279729)

[3.3 系统非功能性需求分析 9](#_Toc480279730)

[3.3.1 可用性 9](#_Toc480279731)

[3.3.2 安全性 10](#_Toc480279732)

[3.3.3 性能 10](#_Toc480279733)

[3.4可行性分析 10](#_Toc480279734)

[4 系统设计 11](#_Toc480279735)

[4.1 总体设计 11](#_Toc480279736)

[4.1.1 系统整体架构设计 11](#_Toc480279737)

[4.1.2 系统API设计 15](#_Toc480279738)

[4.1.2 系统模块划分 16](#_Toc480279739)

[4.2 数据库设计 17](#_Toc480279740)

[4.3 详细设计 20](#_Toc480279741)

[4.3.1 编辑教研活动信息设计 21](#_Toc480279742)

[4.3.2 统计教研活动设计 22](#_Toc480279743)

[4.3.3 用户信息操作设计 23](#_Toc480279744)

[5 系统实现 24](#_Toc480279745)

[5.1 权限控制实现 24](#_Toc480279746)

[5.2 系统首页界面实现 25](#_Toc480279747)

[5.2 编辑功能的实现 27](#_Toc480279748)

[5.3 统计信息的实现 31](#_Toc480279749)

[5.4 用户管理的实现 36](#_Toc480279750)

[6 结论与展望 37](#_Toc480279751)

[6.1 总结 37](#_Toc480279752)

[6.2 展望 37](#_Toc480279753)

[参 考 文 献 38](#_Toc480279754)

[致 谢 40](#_Toc480279755)

华南农业大学本科生毕业论文成绩评定表

# 1 绪论

## 1.1 研究的目的和意义

社会的快速发展让人们逐渐明白：当今世界各国的竞争依赖于高素质的人才，而人才的培养靠的是高素质的教育，而教育的成败依赖于高素质的教师（张群，2011）。可以通过教学基本功训练活动、更新教师观念、开展教研活动等多种形式提高教师素质。

教研活动是推动教学改革、提升教师教学能力的重要途径，是促进课程与专业建设、培育优秀教学成果的有效手段（刘维周，2014）。本系统针对教研活动形式，研究如何更有效的对研究活动进行管理。

教研活动与教学活动不同，教研活动探讨的是教研活动组织者（如教研组组长、教研员）、教师以及教研环境之间形成的活动，属于同质文化的活动（龚兴英，2014）。也就是说，教研活动以教师为研究主题，以专业研究人员为合作伙伴的以校为本的实践性研究活动。因此，本系统的用户主要为教师以及教研活动组织者（管理层）。

教研活动开展的效果如何，直接影响着教学活动的开展与科研水平，从长远看会促进或制约高校的发展（谢英，许明远，张彦，2015）。因此有效的开展教研活动，对教学工作起到促进作用，使教学质量得到提高。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）。计算机的超强计算能力，实现了信息和数据的快速输入与处理，计算机应用在各行各业中得到了广泛的使用。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）。

信息化是充分利用信息技术，开发利用信息资源，促进信息交流和知识共享（黄堂红，2009）。教研活动管理信息化的意义在于，提高管理质量，促进管理的公开化，为管理者提供准确及时的信息，实现教研活动信息的资源共享。

## 1.2 主要研究内容

本系统的目标是收集教研活动信息，帮助人们展示这些活动信息，快速进行统计。教师和管理层随时随地都能够查看教研活动相关信息和数据，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，让人们有更多的时间学习和享受生活。

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，使教师增强自身积累，提高工作效率，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，主要开发技术为基于 JavaScript的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。以下，将对前端框架Vue.js、服务端语言Node.js、数据库MongoDB以及自动化打包工具Webpack进行介绍。

## 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。

利用组件化的抽象方式，有助于我们提高代码复用率，从而提高开发效率（梧忌，2015）。组件化的设计能够很好的解决以上问题。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是一个非阻塞、数据密集型、事件驱动、实时跨终端的JavaScript开发和运行平台（E. Besada-Portas, J. Bermúdez-Ortega, L. de la Torre，etc,2016）。

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，I/O 操作不会阻塞其他请求，每个请求也不会增加线程/进程的开销（mixu,2011）。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。

Paul T. Grogan, Olivier L. de Weck, Adam M. Ross等人开发的ISRM服务段是基于Node.js和MongoDB的，同时在他们发表的论文中也表示，选择它们的原因是因为在整个应用层，包括客户端（浏览器）、服务端和数据库文档使用同一开发语言，能够方便的重用代码和最小化适配（Paul T. Grogan,Olivier L. de Weck, Adam M. Ross,etc,2015）。开发过程无需切换语言语法，有利于提高开发效率以及代码重用性。

## 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。传统的关系型数据库并不符合Node.js的这一特性，而MongoDB文件类似于JSON对象。在MongoDB中一个记录是一个文档，文档是由键值对组成数据结构（Naresh Kumar Gundla,Zhengxin Chen,2016）。Node.js从数据库MongoDB中获取数据后，即可直接操作，不需要进行数据转换和匹配

MongoDB是一种面向文档的NoSQL数据库（Tsukasa Kudo,Masahiko Ishino,Kenji Saotome,etc,2016）。在基于Web的结构中，关系型数据库很难横向扩展，而NoSQL数据库没有固定的结构，扩展性强，在不改变其他文档的情况下，允许为个别文档添加新属性。

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）。

MongoDB是增长最快的数据库。它提供了丰富的动态查询的面向文档的结构，它也允许将数据分割成集合（Rania Yangui, Ahlem Nabli, Faiez Gargouri,2016）。加上结构灵活，以冗余的方式提供读取性能，需要的信息可以一次获取。

## 2.4 Webpack

本系统为Web App，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自动化打包工具，Webpack的主要目的是将JavaScript文件打包，但它也能够转换和打包其它的资源（sokra，2017）。本系统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（single page web application，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能够将模块分割成很多小模块。对分割后的模块进行按需请求加载。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，具有信息分散、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（学院、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。

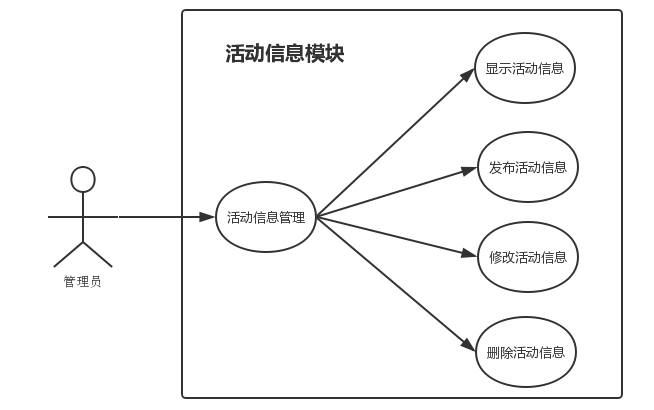


图1 活动信息管理模块用例图

### 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。

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。
     3. 统计：查看自己以及校内所有教研室举办活动总体的情况。
3. 用户管理：可对所有用户进行增删改查管理。

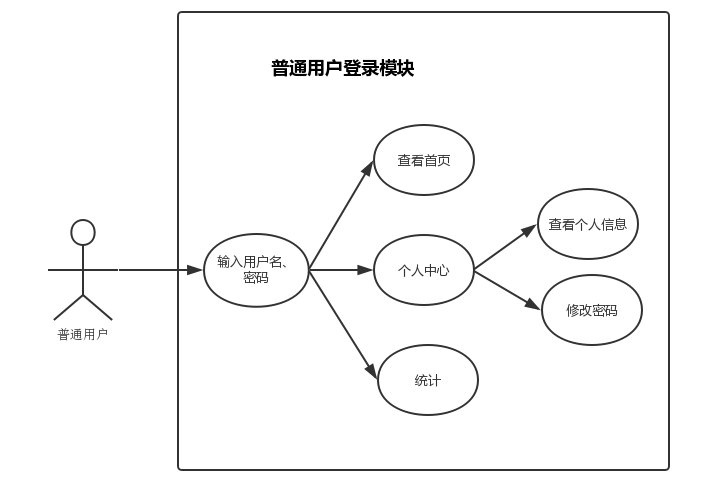


图2 普通用户登录用例图

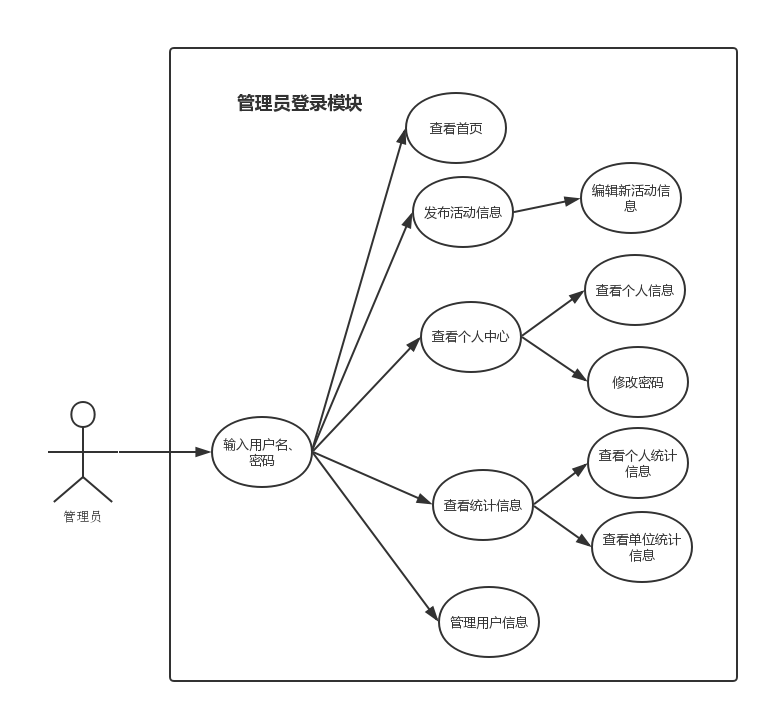


图3 管理员登录用例图

## 3.3 系统非功能性需求分析

### 3.3.1 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，不同的模块功能设置在导航栏中；

当发生操作错误，提供友好的提示信息，如创建新的教研活动信息时，必须项用户未填写完整，以提示框的形式提示用户。

创建、修改教研活动信息时，当用户提交一次后，“提交”按钮样式会发送变化，且不可点击，防止用户多次提交。

### 3.3.2 安全性

系统无论大小，保障用户数据以及信息的安全是最基本的要求。

本系统在添加用户信息或者用户修改密码，数据插入到数据库之前，会先对用户设置的密码预处理，经过sha256加密后，将密文存入数据库。这样，即使数据库泄露，也不用担心用户密码丢失，信息被篡改。同时，保证了数据库管理员不能窃取用户密码。

发布、修改教研活动信息时，系统会对提交的内容进行过滤，若含有不合法内容，如“<script>”等脚本代码时，自动转换成字符串，再存进数据库中，避免XSS漏洞。

系统对用户权限进行严格的分类管理，前后端双重控制，用户只能拥有与等级相符的操作。如编辑页面、用户管理页面，即使用户在浏览器上直接输入地址，系统会判断用户等级，若用户为普通用户，则重定向到首页。

### 3.3.3 性能

本系统前端采用的是SPA设计模式，页面切换过程不会有频繁的等待，操作体验流畅，能够媲美本地应用。

服务端使用Node.js搭建，支持数万条并发连接，足够学校内部使用。

## 3.4可行性分析

（1）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。

（2）技术可行性

本系统开发过程中实现前后端分离的思想，前端采用了SPA设计模式，SPA需要浏览器的缓存机制、HTML5以及路由技术的支持。

SPA是仅包含一个网页的应用，所有业务功能都是它的子模块。由于整个系统只有一个页面，页面内容发生变化使用的是路由进行切换。例如，管理员在首页通过导航栏上的“发布”选项进入到编辑页面这一过程，路由从浏览器缓存中获取编辑页面的内容，让浏览器渲染展示，而无需向服务端发送任何请求。页面切换过程仅发生在浏览器中，省去了DNS解析、IP查询、TCP三次握手等繁琐的连接程序，使得操作流畅。

SPA需要浏览器的缓存机制支持，浏览器缓存早早HTTP协议1.0就有定义，HTTP1.1又对浏览器缓存规则进行添加补充。浏览器的缓存机制已经很成熟，随着HTML5的出现，使浏览器拥有更多的缓存方式：localStorage、sessionStorage、离线缓存，为开发者提供更多的缓存接口：setItem、getItem，能够方便、快速的操作浏览器缓存。现今许多应用都使用浏览器缓存机制，对变动不频繁的静态文件进行存储。

本系统前端使用Vue.js搭建，Vue.js拥有配套的路由器，结合HTML5的History API，可方便的对页面进行自由切换。

SPA需要的浏览器缓存、HTML5以及路由的技术支持都已发展的很成熟，因此本系统使用SPA设计模式开发是可行的。

# 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，每个学院又包含一个以上的教研室。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。

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)）进行增加、删除、修改、查询功能操作，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。

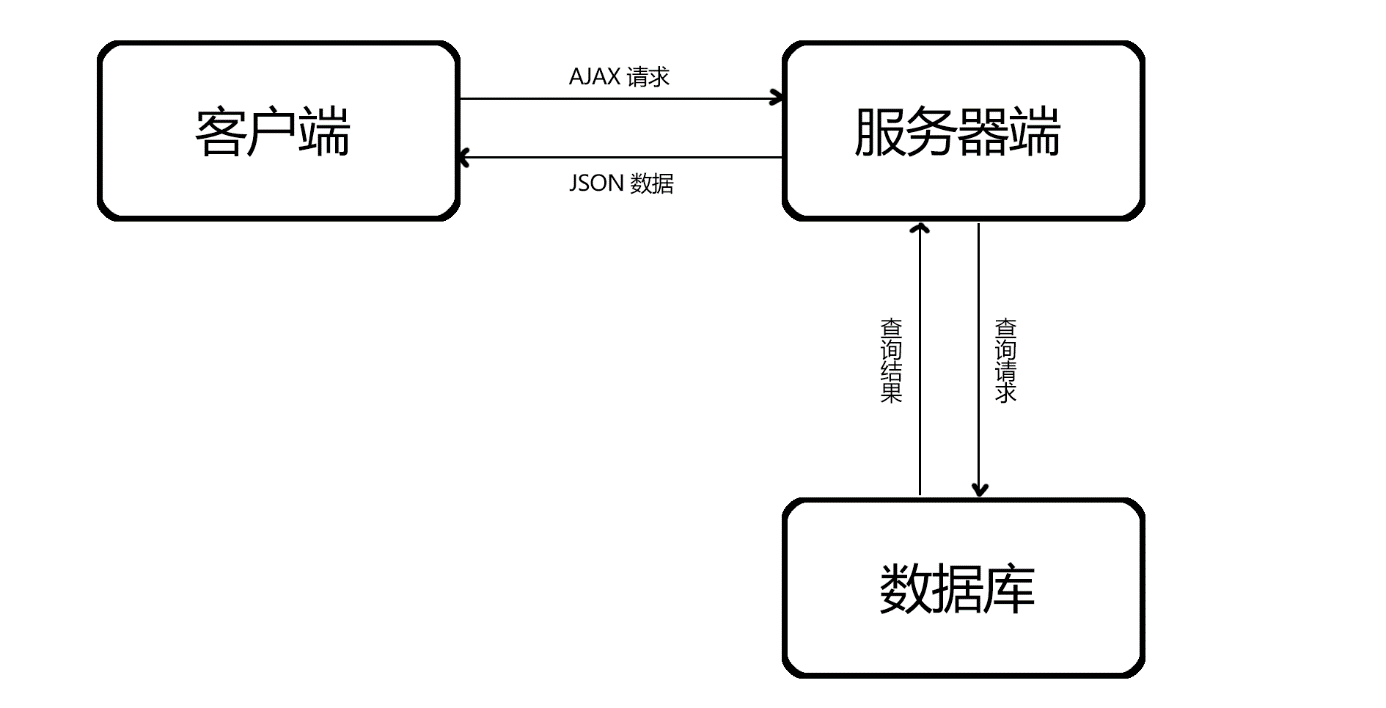


图4 系统数据流图

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。

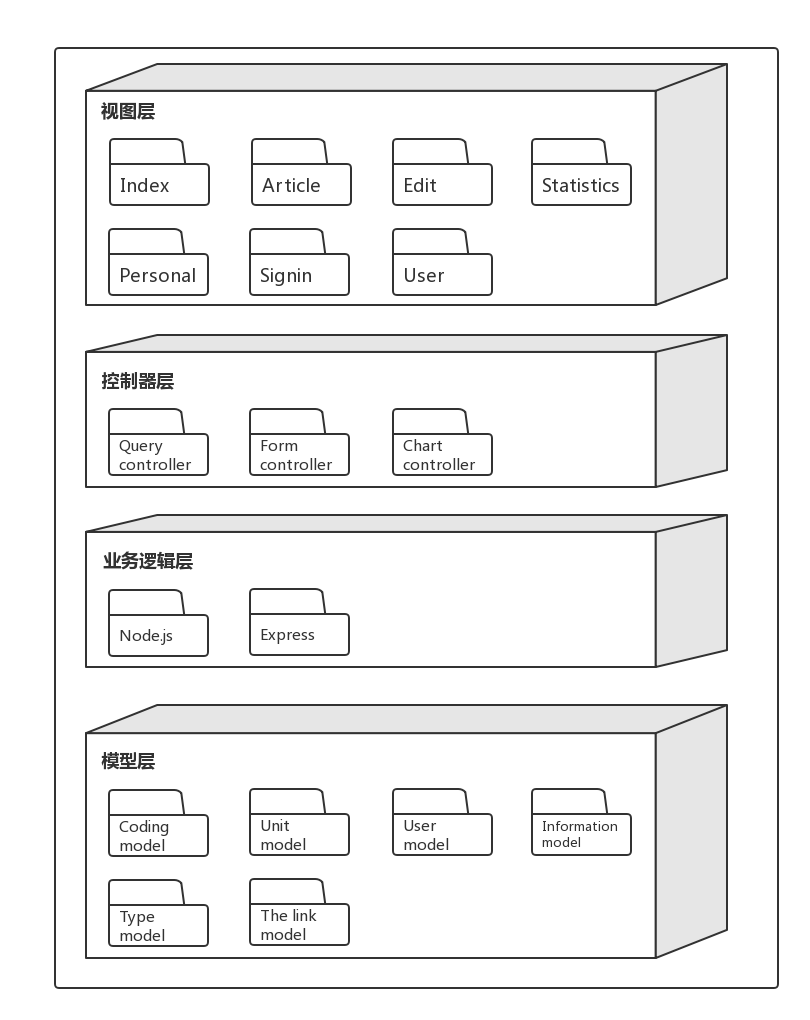
系统整体架构如图5所示。

图5 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：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?id=: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 | GET |

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 | /userManage/modifyPwd | 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。

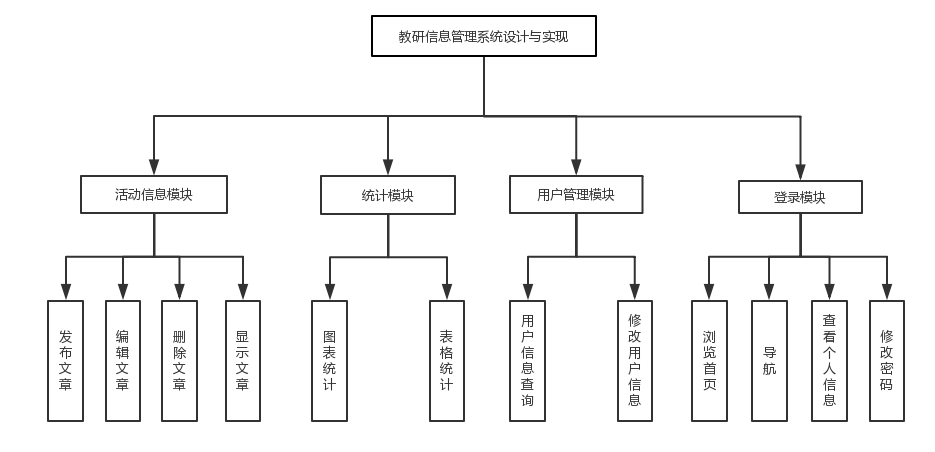


图6 系统功能结构图

## 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：

(1)一对很少（one-to-few），使用内嵌；如每个学院包含多个教研室；

(2) 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；

(3) 双向关联， one 端和 many 端同时保存对方的引用；

MongoDB是面向集合存储的，由文档、集合、数据库组成；MongoDB中的概念与SQL的概念对比如下表：

表7 MongoDB与SOL术语映射表

|  |  |  |
| --- | --- | --- |
| MongoDB术语/概念 | SQL术语/概念 | 解释/说明 |
| Database | Database | 数据库 |
| Table | Collection | 数据库表/集合 |
| Row | 数据记录行/文档 | 数据记录行/文档 |
| Column | Field | 数据字段/域 |
| Index | Index | 索引 |
| Primary key | Primary key | 主键，MongoDB自动将——id字段设置为主键 |

本系统共设计有5个集合：User、Article、FootLink、ArticleType、Academy。系统会为每一个文档自动生成唯一标识的主键。

1. User集合

表8 User集合

|  |  |
| --- | --- |
| 域 | 类型 |
| \_id | ObjectId |
| name | String |
| account | Number |
| pwd | String |
| rank | Number |
| faculty | String |
| title | String |

该集合用于保存本系统所有用户的信息，包含了用户名称、账号、用户密码、用户等级、用户所属教研室、用户职称。数据库中存储的用户密码为经过sha256加密后的密文。

1. Article集合

表9 User集合

|  |  |
| --- | --- |
| 域 | 类型 |
| \_id | ObjectId |
| author | String |
| abs | String |
| time | Date |
| startTime | Date |
| endTime | Date |
| address | String |
| content | String |
| enclosure | String |
| faculty | String |
| type | String |
| participator | [String] |
| title | String |

该集合用于保存本系统中已发布的教研活动信息，包含了教研活动标题、活动信息链接地址、发布人、活动信息摘要、发布时间、教研活动开始时间、教研活动结束时间、活动举办地址、活动举办单位、主要内容、附件、教研活动类型、参与者。

活动信息链接地址系统根据\_id自动分配。

发布人、发布时间为默认信息：发布人为当前用户，发布时间为插入数据库时间（一般为创建时间）。

当操作用户为普通管理员，举办单位默认为用户所属教研室；当操作用户为系统管理员，可选择举办单位。

参与者只在修改状态下可填。

（3）FootLink集合

表10 User集合

|  |  |
| --- | --- |
| 域 | 类型 |
| \_id | ObjectId |
| url | String |
| text | String |

该集合用于保存页面底部友情链接，包括链接名称和链接地址。

（4）ArticleType集合

表11 User集合

|  |  |
| --- | --- |
| 域 | 类型 |
| \_id | ObjectId |
| name | String |
| index | Number |

该集合用于保存教研活动类型，包括类型名称，对应的索引。

设计该集合的目的是：

1. 首页以及编辑页面，需要显示可选择的教研活动类型，当需要添加教研活动类型时，只需要修改该集合；
2. 当有教研活动类型名称改变时，只需要修改该集合，无需修改包含该教研活动类型的所有Article文档。

（5）Academy集合

表12 User集合

|  |  |
| --- | --- |
| 域 | 类型 |
| \_id | ObjectId |
| name | String |
| index | Number |
| staff | [selectType] |

该集合用于保存学校所有的学院，以及每个学院的所有教研室，包括了学院名称、学院索引、学院下的教研室。

selectType为自定义类型，该类型包含了String类型的type域和index域。

设计该集合的目的与ArticleType集合同理：

首页以及用户为系统管理员时的编辑页面，需要显示学院以及所选学院下的教研室，当需要学院以及教研室结构发生改变时，只需要修改该集合。

## 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 编辑教研活动信息设计

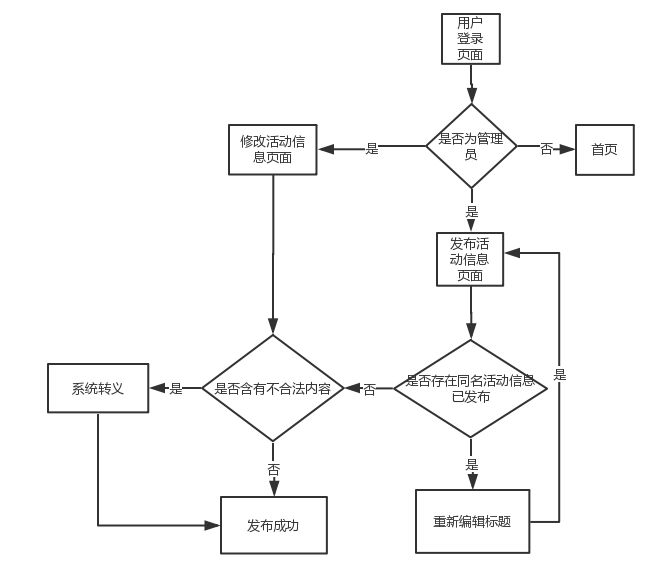


图7 发布教研活动信息流程图

编辑功能主要在编辑页面实现，编辑页面分为发布和修改教研活动信息两个状态。

用户需登录后，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，只有管理员的导航中出现“发布”选项。若已登录普通用户在浏览器地址栏上输入编辑页面的，系统会重定向到首页，未登录用户则会重定向到等登录页面。

发布信息状态：管理员可通过导航栏中的“发布”选项，进入编辑页面的发布状态。管理员填写好需要发布的教研活动信息，提交请求后，系统先查询数据库，判断该用户是否有已发布的相同标题的教研活动信息。若有，则以提示框的形式，告知用户该标题已存在，用户须修改标题才能发布。若没有，系统先过滤发布内容，如含有不合法内容，则自动转换成字符串，再插入数据库；否则直接插入数据库，插入成功，以提示框形式提示用户发布成功，用户点击确定后，系统自动跳转到教研活动信息详情页面。

修改信息状态：管理员可以通过编辑工具栏中的“编辑”选项，进入编辑页面的编辑状态。编辑状态下，可填写参与者名单，以中文或英文全角逗号分隔。管理员修改好信息后，提交请求。系统过滤发布内容，如含有不合法内容，则自动转换成字符串，再插入数据库；否则直接插入数据库，插入成功，以提示框形式提示用户修改成功，用户点击确定后，系统自动跳转到教研活动信息详情页面。

### 4.3.2 统计教研活动设计

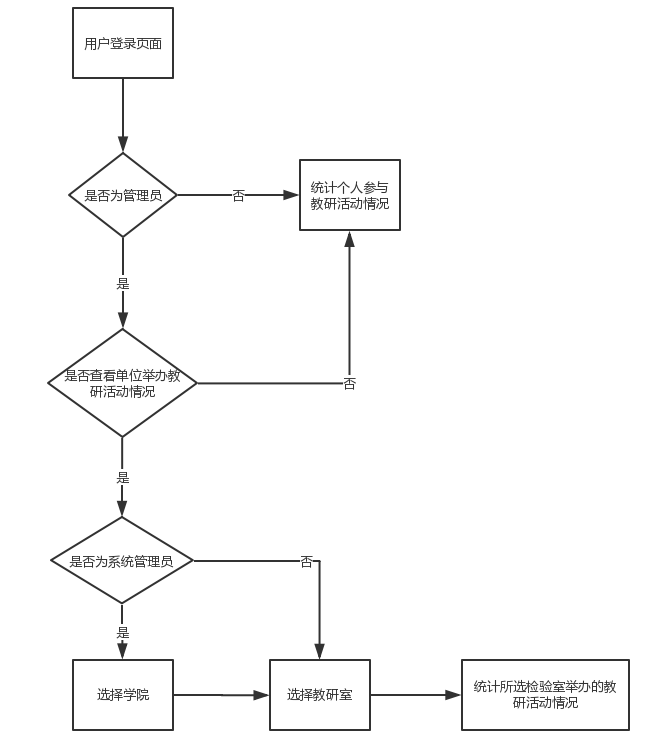


图8 统计教研活动流程图

所有用户登录后，均可在导航栏的“统计”选项中进入统计页面，选择不同的时间段（年份、全年、上半年、下半年），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，以及所属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。

统计页面同包含4张动态图表：一张包含每个月三种教研活动类型举办次数的柱状图，以及三张各类型（现有3中类型）教研活动举办次数百分比的水球图。其中，柱状图右上角有工具栏，可自由切换为柱状图、折线图以及表格形式，同时还能下载为图片，保存到本地

### 4.3.3 用户信息操作设计

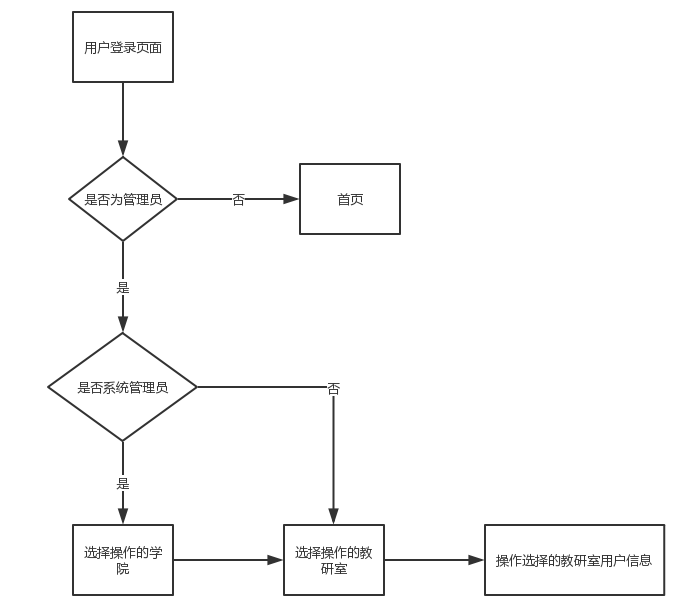


图9 用户信息操作流程图

管理员可通过导航栏中“用户管理”选项进入用户管理页面。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。

用户信息以表格形式展示，每一行包括了用户的账号、用户名称、用户所属教研室、用户权限等级，以及编辑工具栏，可修改、或删除对应的用户信息。

# 系统实现

本系统实现前后端分离的开发思想，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。同时页面切换由前端进行控制，减少服务端的连接请求，减轻服务器的负担。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。

## 5.1 权限控制实现

本系统开发以前后端分离的思想实现，因此，权限信息由数据库保存，权限控制由为前后端共同负责，确保系统权限安全。

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，用户登录之后，系统在服务端创建新会话，保存当前用户信息，同时将用户信息如用户账号、用户等级等一些相关信息返回给客户端。客户端与服务器根据用户信息进行权限控制。

用户输入登录信息，浏览器向服务端发送登录请求。服务端向数据库查询该用户是否存在；若不存在，返回消息，由页面提示用户所输入的账号不存在；若存在，数据库将用户信息返回给服务器，服务器将用户输入的密码进行sha256加密，与数据库中的密文密码进行对比，相同则登录成功，创建新的会话；否则返回密码错误消息，由页面提示用户。

登录成功后，客户端获得来自服务器的用户信息并存储起来，根据用户等级，在导航栏添加相应的功能模块，此为前端层面对用户权限的控制。



图10 未登录状态导航栏



图11 普通用户状态导航栏



图12 管理员状态导航栏

本系统设置的有效期为1天，即用户登录后1天内，重新打开本系统，系统从浏览器缓存中获取用户信息，因此用户无需登录。若缓存中无用户信息，页面自动向服务端发送请求，获取用户信息。

前后端分离后，服务端负责提供API，处理业务逻辑。当有创建、修改、删除教研活动信息、用户信息，查询统计情况等功能时，服务端先获取会话中的用户信息，判断用户操作的权限，若权限不够，则返回操作失败消息；若会话中无该用户信息，则需要用户进行登录后再操作。此为服务端（后端）层面对用户权限的控制。

## 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。



图13 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得到的信息，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。

其中，学院、教研室以级联方式实现，用户需要先选择学院，才会有相应的教研室可以选择，默认为“全部”。若没有选择学院，即学院为“全部”的情况下，选择教研室，教研室只有“全部”选项。



图14 首页搜索栏

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。列表包含了教研活动标题、活动开始时间以及举办地点。若用户为系统管理员，列表中每项活动下都会显示编辑工具栏，方便管理员对活动信息进行操作。若用户为普通管理员，列表中只有用户自己发布的、以及属于用户所在教研室的教研活动信息，会在每项活动下显示编辑工具栏。

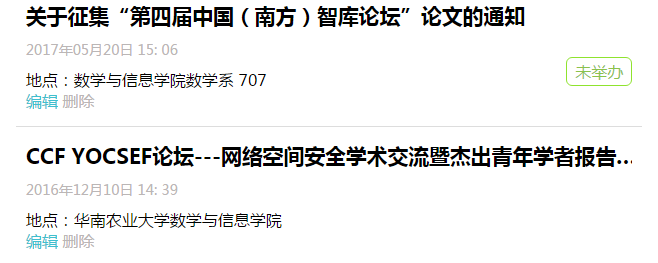


图15 首页教研活动列表栏

考虑到教研活动信息会越来越多，一次加载完所有信息，不仅耗时长，且很多过时的活动信息对于用户是无用的。因此，本系统首页的教研活动列表进行分页显示。通过分页器进行控制。

浏览器每次向服务端请求教研活动数据时，都会带上pageSize（需要的数据数量）和page（请求页数）两个参数，便于服务器定位的数据。服务器返回的数据中，total表示数据库中符合分类条件的数据总数量。浏览器通过pageSize和total计算出总页数，每次请求都会更新总页数。



图16 首页分页器

当教研活动列表位于第一页（最后一页）时，“上一页（下一页）”按钮为灰色且不可点击。用户可在输入框中直接输入希望查看的页数，点解“跳转”按钮直接跳到相应页数；若用户输入不合法，如输入除数字以外的字符，页面会提示用户“请输入有效数字”；若输入的数字超过总页数，则跳转到最后一页。另外，每次翻页浏览器滚动条会自动回到列表顶部上方，方便用户查看。

## 5.2 编辑功能的实现

编辑功能包括了编辑页面中的创建状态和修改状态，以及编辑工具栏。编辑页面的主要功能为编辑信息，为了让使用户能够专注于信息编辑，不被干扰，因此界面设计尽量的清晰简洁。

管理员通过导航栏的“发布”选项，进入到编辑页面的发布状态，填写需要发布的教研活动信息。输入框的提示语已内嵌形式展示，输入框前带“\*”号的为必填项，需填写完整才能发布，否则系统将提示用户填写完整。

下图为系统管理员下的编辑页面，若用户为普通管理员，则没有“学院”选择框，“举办单位”默认为用户所在教研室。

这里的“学院”与“举办单位”下拉框同样以级联的方式实现，系统管理员选择了某个学院之后，页面自动向服务端发送请求，获取所选学院下的教研室，并动态加载到“举办单位”下拉框的选项中。若管理员在未选择学院的情况下，点击“举办单位”下拉框，页面会以提示框形式，提醒管理员，需先选择学院。

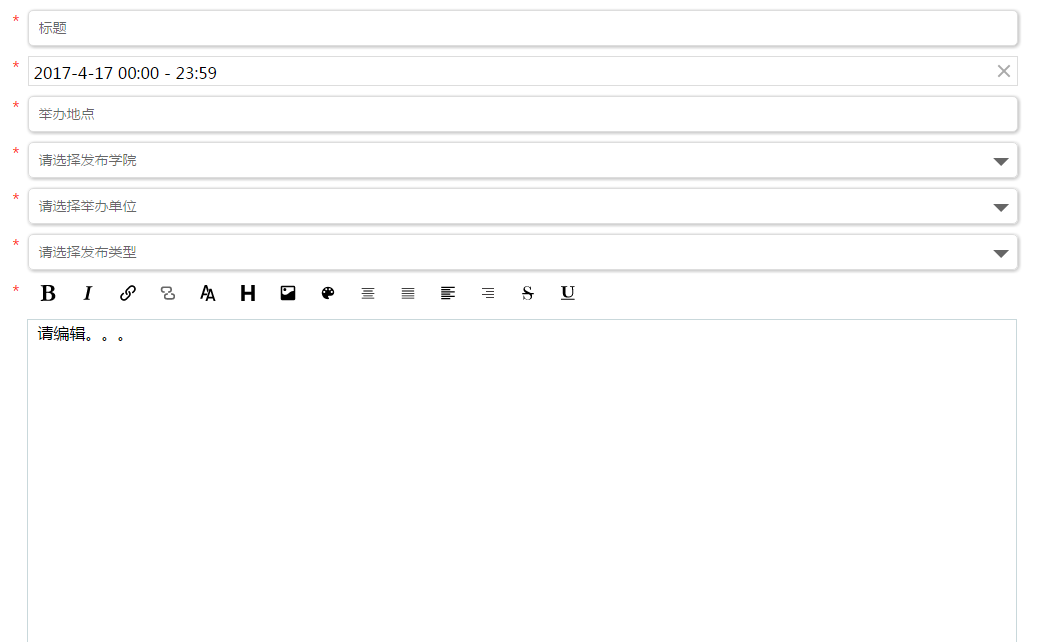


图17 系统管理员编辑页面发布状态（上）

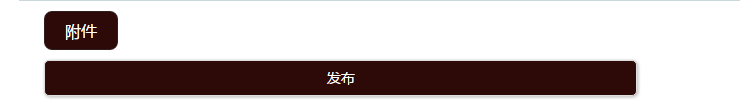


图18 编辑页面发布状态（下）

时间选择器以日历形式展示，由于教研活动举办时长一般为一天以内，因此该时间选择器设计为只能选择一天，可在按钮上方输入框输入开始时间与结束时间，默认为00：00到23:59.即一天。为防止用户输入不合法时间，若输入非数字字符，系统会自动清零；若结束时间早于开始时间，系统会自动进行调整，如输入10：00-02：00，点击确认后，系统自动调整为10：00到10：00。

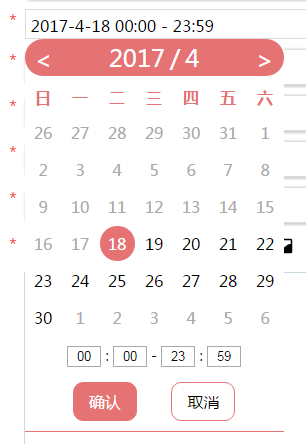


图19 编辑页面时间选择器

富文本编辑器工具栏包含了加粗、斜体、链接、解除链接、字体大小、2-6等级的标题、插入图片、颜色选择、居中、对齐、左对齐、右对齐、划线以及下划线14种基本功能。

编辑器的大小是固定的，当内容超出编辑器大小，则以滚动条形式展示。

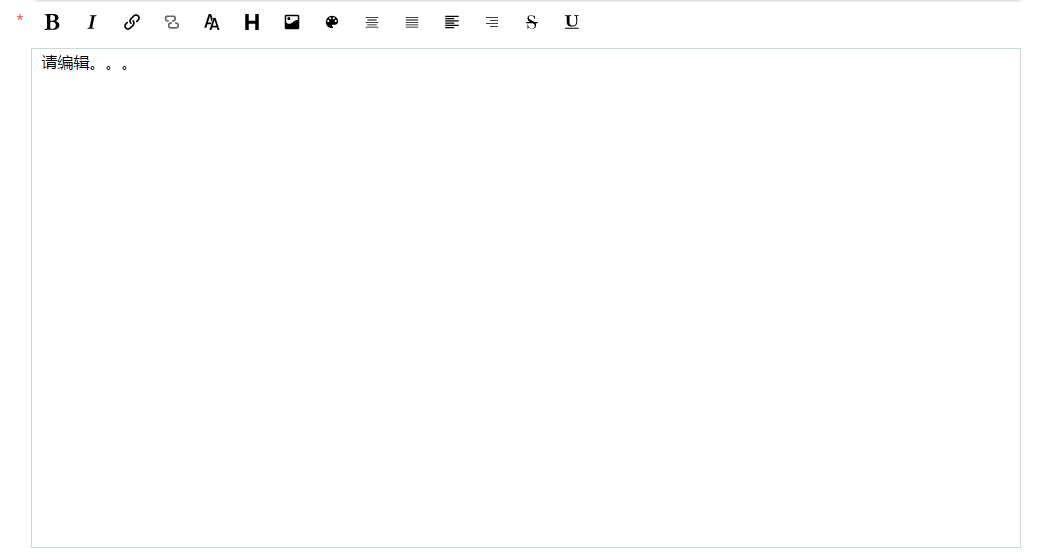


图20 富文本编辑器

管理员可在首页的教研活动列表，或者教研活动信息详情标题右下方，找到编辑工具栏，编辑或者删除教研活动信息。

点击“删除”按钮，页面将向服务器发送请求，带数据库删除该教研活动信息并返回消息，由页面以提示框形式告诉用户删除成功，用户点击确认后，页面自动跳转到首页。



图21 教研活动信息详情页面

点击“编辑”，进入编辑页面的修改状态。页面向服务器请求需要编辑的教研活动信息的详情，获得数据后，页面自动将数据导入到对应的地方展示。



图22 教研活动信息编辑状态

在编辑状态下，管理员可填入活动参与者名称，以中文或英文全角逗号作为分隔符。

## 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，统计相应教研室举办教研活动的情况。

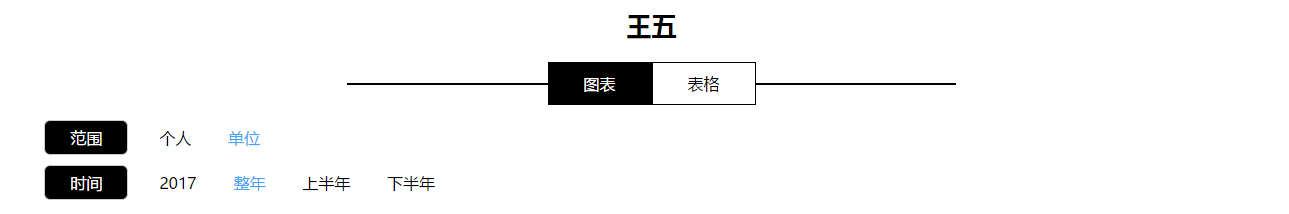


图23 管理员统计页面选择组件

用户点击“时间”标签后的年份，系统弹出年份选项模态框，切换不同的年份；默认为全年，可选择上半年（1-6月）或者下半年（7-12月）



图24 年份选项模态框

用户鼠标悬在“单位”选项上时，会出现可选择的教研室，点击选项，图表自动切换为对应的统计情数据。

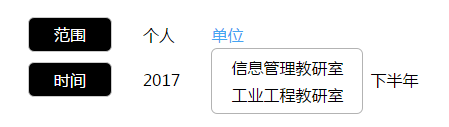


图25 单位选项

总体统计图默认为柱状图，以动画形式进入页面。红色矩形代表“教学讨论会”，深蓝色代表“科研研讨会”，浅蓝色代表“学术沙龙”。

统计页面的4张图表均用canvas绘制。用户每次切换选择组件上的选项，页面会自动向后台发送请求。后台根据相关参数，查询数据库中Article集合中符合条件的文档；得到数据后，对数据进行处理，返回整理后的数据给页面。

如用户希望查看2015年全年，工业工程教研室的统计情况，则后台向数据库查询时间在2015年1月1日至2015年12月31日内，且举办单位为工业工程教研室的所有教研活动信息；得到数据后，过滤多余的数据，每个文档只保留startTime（活动举办时间）以及type（教研类型），减少冗余数据的传输，加快响应速度。

页面接收到数据，对数据进行预处理：根据type和startTime计算每个月每种教研类型的举办次数，并存储在统计对象中；同时统计每种教研类型占总数的百分比，存储在百分比对象中。得到处理后的数据，更新柱状图的矩形面积、坐标，以及水球图，重新绘制图表。

若用户连续点击相同的选项，页面不会发送请求，图表也不会更新，达到减少浏览器工作量，节省内存的效果。

下图为用户选择2015年全年，工业工程教研室的统计情况。

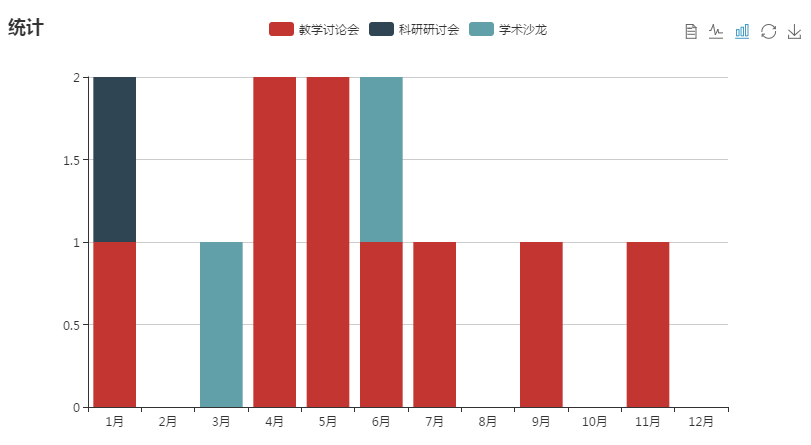


图26 系统管理员统计页面柱状图

鼠标悬浮在矩形上，鼠标右下角会出现提示框。通过提示框可得知，2015年6月，工业工程教研室共举办了1次教学研讨会、1次学术沙龙，没有举办科研研讨会。



图27 系统管理员统计页面柱状图提示框

柱状图右上角有工具栏，可切换不同表格。



图28 系统管理员统计页面柱状图工具栏

工具栏上的选项分别为数据视图、折线图、柱状图、还原、下载。

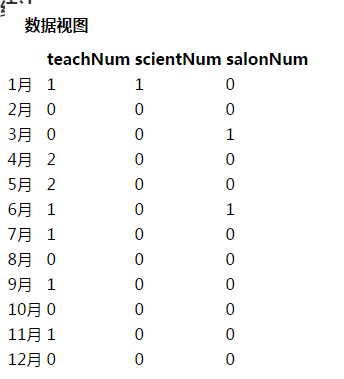


图29 系统管理员统计页面数据视图

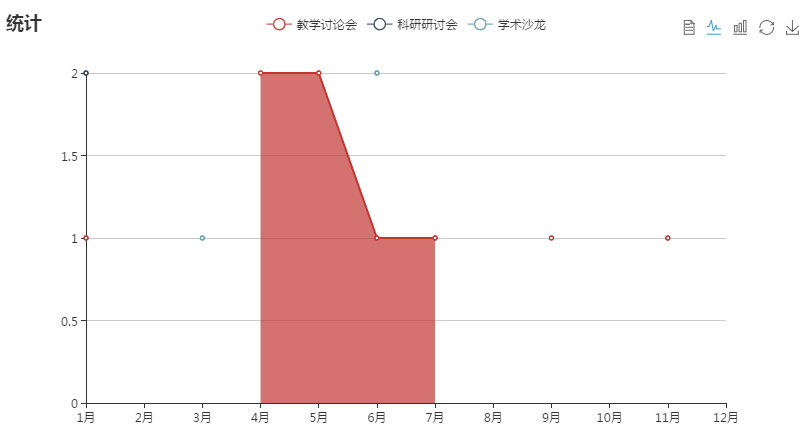


图30 系统管理员统计页面折线图

下载选项会将页面上当前的表格，下载为png格式的图片，保存到本地。

下图为三种教研活动类型举办次数百分比占总数的比例，水球图展示的数据较简单，为了避免图表太单一，因此添加了动画效果。通过水球图可知，2015年工业工程教研室举办的教研活动中，教学讨论会举办次数占总次数的75%，科学研讨会占8%，学术沙龙占17%。



图31 统计页面水球图

当没有数据时，不会展示柱状图和水球图。以“暂无记录”代替。

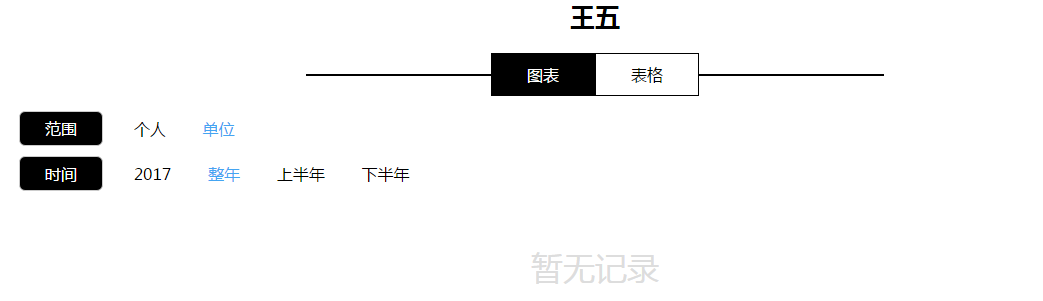


图32 统计页面无数据情况

## 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操作，系统管理员能对所有用户进行增删改查操作。

用户管理分为添加用户、用户信息两个板块。

添加用户板块中，“学院”和“教研室”以级联方式实现，系统管理员需先选择“学院”后，才能选择“教研室”，所有选项都需填写完整才能添加用户。用户账号不能有重复。



图33 添加用户信息页面

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，展示的用户信息包括用户名称、用户账号、用户职称、用户所属系、用户等级，每个用户都有对应的修改和删除按钮。



图34 查看用户信息页面

# 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SPA，提高重用性；页面使用了响应式设计，适应不同尺寸的屏幕。服务器使用了与前端语言一致的 Node.js 语言搭建，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，本人完成了项目的原型图设计、用户界面设计，前端架构的搭建，服务端API设计，以及数据库建模。

本系统实现的主要功能有：对教研活动信息创建、修改和删除；对不同教研室举办的教研活动进行分时间段的统计；对用户的添加、修改以及删除；实现了查看教研活动信息的大致内容以及详细内容。

## 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。但在细节上仍有许多可以完善的地方：

编辑功能：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。

用户管理功能：只展示了用户的基本信息，没有更加详细的，如每个人对于教研活动的参与情况等，信息量大，需要重新设计数据库中的User表，暂时没实现。

搜索功能：目前首页只实现了分类查询功能，并未实现精确搜索、模糊搜索等搜索功能。

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

陈屹峤. 我们为什么需要Webpack？[EB/OL]. 西安：github，2016[20170413]：<https://chenyiqiao.gitbooks.io/webpack/content/motivation_of_webpack.html>

龚兴英.中小学教师教研活动研究[D].重庆：西南大学，2014

黄堂红. 教研信息化的内涵、意义及发展对策探讨[J].电化教育研究，2009，38（3）：26-28

劳卜. 浅谈Vue.js[EB/OL].杭州：segmentfault，2016[20170413]. <https://segmentfault.com/a/1190000004704498>

刘维周. 南京晓庄学院文件[EB/OL]. 南京.：南京晓庄学院，2014

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，2010，22（6）：35-36

梧忌.已买到的宝贝前端组件化探索[EB/OL].杭州：淘宝前端团队，2015[20170418]

谢英，许明远，张彦.高校教研活动的现状调查与对策研究[J].科教文汇，2015，14（7）：120-121

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

尤雨溪. Vue.js：a (re)introduction[EB/OL].杭州：知乎，2016[20170413]：<https://zhuanlan.zhihu.com/p/20302927?columnSlug=evanyou>

张群. 校本教研活动促进教师专业发展的个案研究[D].武汉：华中师范大学，2011

Zqeesoom. mongodb与mysql相比的优缺点[EB/OL].北京：新浪博客，2012[20170413]：<http://blog.sina.com.cn/s/blog_966e430001019s8v.html>

E. Besada-Portas, J. Bermúdez-Ortega, L. de la Torre,etc.

Lightweight Node.js & EJsS-based Web Server for Remote Control Laboratories[J]. IFAC-PapersOnLine,2016,49(6):127-132

Naresh Kumar Gundla, Zhengxin Chen. Creating NoSQL Biological Databases with Ontologies for Query Relaxation[J]. Procedia Computer Science,2016,91(7):460-469

Rania Yangui, Ahlem Nabli, Faiez Gargouri.

Automatic Transformation of Data Warehouse Schema to NoSQL Data Base: Comparative Study[J]. Procedia Computer Science,2016,96(8):255-264

Tsukasa Kudo,Masahiko Ishino,Kenji Saotome,etc.A Proposal of Transaction Processing Method for MongoDB[J].Procedia Computer Science,2016,96(8):801-810

Paul T. Grogan,Olivier L. de Weck, Adam M. Ross,etc. Interactive Models as a System Design Tool: Applications to System Project Management[J]. Procedia Computer Science,2015,44(3):285-294

Kenneth Peeples. What are the Benefits of Node.js?[EB/OL]. Menlo Park.DZone.2015[20170418]. https://dzone.com/articles/what-are-benefits-nodejs

Mixu. Understanding the node.js event loop[EB/OL]. America.MIXU’S BLOG.2011[20170418]. <http://blog.mixu.net/2011/02/01/understanding-the-node-js-event-loop/>

Sokra. Webpack[EB/OL].San Francisco.github.com,2017[20170418]. https://github.com/webpack/webpack/blob/master/README.md

Wikipedia.Node.js[EB/OL]. America.:Wikipedia,2015[20170413].https://en.wikipedia.org/wiki/Node.js

Tomislav Capan. Why The Hell Would I Use Node.js? A Case-by-Case Tutorial.[EB.OL]. Croatia:developers,2013[20170413]. <https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>

Uma Narayanan. MongoDB NoSQL Solution: Advantages and Disadvantages [EB/OL]. America.developer.com,2015[20170418]: http://www.developer.com/db/mongodb-nosql-solution-advantages-and-disadvantages.html

致 谢

在母校的三年多生活学些中，收货颇丰。从开始准备毕业设计到现在，许多人给予了我无私的帮助与悉心教导。

首先，非常感谢我的毕业指导老师杨振刚，毕业设计选题以及开发过程，在导师的指导下完成的，老师给予了很多的帮助以及指导。导师严谨的治学态度，精益求精的工作作风，为我将来的工作和学习树立了很好的榜样，在论文修改规程中，从内容结构到格式，都给予详细的修改意见。

感谢这三年多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。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。

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。

# 附录A

统计功能模板以及交互功能代码如下。

<template>

<div>

<ul class='select-list'>

<li v-show='userRank > 0'>

<div>

<label>范围</label>

<a :class="{'active': rangeTab == 0}" @click='changeRangeTab' data-type='0'>

个人

</a>

<a class='select-box' :class="{'active': rangeTab == 1}" @click='changeRangeTab' data-type='1'>

单位

<ul class='faculty-lsit'>

<li v-for='item in faculty'>{{item}}</li>

</ul>

</a>

</div>

</li>

<li>

<div>

<label>时间</label>

<a @click='selectYears'>

{{selectYear}}

</a>

<a :class="{'active': timeTab == 0}" @click='changeTimeTab' data-type='0'>

整年

</a>

<a :class="{'active': timeTab == 1}" @click='changeTimeTab' data-type='1'>

上半年

</a>

<a :class="{'active': timeTab == 2}" @click='changeTimeTab' data-type='2'>

下半年

</a>

</div>

</li>

</ul>

<div v-show='empty' class='empty'>

暂无记录

</div>

<div id='canvas-wrapper' v-show='!empty'>

<div v-for='item in charts' :id='item' width='300' height='250' class='canvas'></div>

</div>

<div class='dialog-wrapper' @click='hideDialog' v-show='showDialog'>

<ul class='dialog'>

<li v-for='year in years' :class="{'active': year == selectYear}" @click='changeYear'>{{year}}</li>

</ul>

</div>

</div>

</template>

<script>

import {mapState} from 'vuex';

import {init, refresh} from '../js/setChart';

import {isEmpty} from '../js/common.js';

export default {

data() {

let year = new Date().getFullYear();

/\*\*

\* @param {Object} opt: 存储服务端返回的图表信息

\* @param {Date} curYear: 当前年份

\* @param {Date} selectYear: 已选择年份

\* @param {Number} rangeTab: 查询类型，0-个人，1-单位

\* @param {Number} timeTab: 查询时间范围，0-全年，1-上半年，2-下半年

\* @param {Boolean} showDialog: 是否显示年份选择框

\* @param {Array} charts: 需创建图表的 id

\* @param {Boolean} empty: 是否有数据

\* @param {Array} faculty: 能够查看的单位

\*/

return {

opt: {},

curYear: year,

selectYear: year,

rangeTab: 1,

timeTab: 0,

showDialog: false,

charts: [

'sumNum',

'itemNum'

],

empty: false,

faculty: []

}

},

mounted() {

let a = setInterval(() => {

if(this.$store.state.userId) {

clearInterval(a);

this.getUnitText();

this.init();

this.rangeTab = this.userRank > 0 ? 1 : 0;

}

}, 100);

},

methods: {

init() {

this.initCharts();

},

selectYears() {

this.showDialog = true;

},

hideDialog() {

this.showDialog = false;

},

changeYear(e) {

this.selectYear = e.target.innerHTML;

this.timeTab = 0;

this.refreshCharts();

},

changeRangeTab(e) {

this.rangeTab = e.target.getAttribute('data-type') || 0;

this.refreshCharts();

},

changeTimeTab(e) {

this.timeTab = e.target.getAttribute('data-type') || 0;

this.refreshCharts();

},

initCharts() {

let id = this.charts,

queryId = this.rangeTab == 0 ? this.$route.params.id : this.$store.state.userFaculty.index,

time = this.timeTab;

let ref = document.getElementById('canvas-wrapper');

if(ref) {

this.$store.dispatch('GET\_CHARTS\_DATA', {id: queryId, tab: this.rangeTab, time: time, year: this.selectYear})

.then(res => {

console.log(res[0])

if(!isEmpty(res)) {

this.empty = false;

this.opt = res;

this.opt.type = this.timeTab;

init(id, this.opt || {});

}else {

this.empty = true;

}

})

}

},

refreshCharts() {

let id = this.charts,

queryId = this.rangeTab == 0 ? this.$route.params.id : this.$store.state.userFaculty.index,

time = this.timeTab;

this.$store.dispatch('GET\_CHARTS\_DATA', {id: queryId, tab: this.rangeTab, time: time, year: this.selectYear})

.then(res => {

if(!isEmpty(res)) {

this.empty = false;

this.opt = res;

this.opt.type = this.timeTab;

refresh(this.opt || {});

}else {

this.empty = true;

}

})

},

getUnitText() {

let faculty = this.$store.state.userFaculty;

this.$store.dispatch('GET\_UNIT\_TEXT', {faculty: faculty.index.split('-')[0]})

.then(res => {

if(this.userRank > 1) {

res.forEach((item, index) => {

if(index > 0) {

this.faculty.push(item.type)

}

})

}else if(this.userRank == 1){

for(let i = 0, len = res.length; i < len; ++i) {

if(res[i].index == faculty.index) {

this.faculty.push(res[i].type)

break;

}

}

}

})

.catch(err => alert(err));

}

},

computed: {

userFaculty() {

return this.$store.state.userFaculty;

},

userRank() {

return this.$store.state.userRank;

},

years() {

let year = this.curYear,

arr = [],

i = year - 5;

for(; year > i; --year) {

arr.push(year)

}

return arr;

}

}

}

</script>

<p></p>

<style scoped lang='stylus'>

@import '../css/funs';

.select-list

margin-bottom 50px

li

width 100%

margin 5px 0

label,

a,

select

display inline-block

label

padding 6px 25px

margin-right 10px

border 1px solid #ccc

color #fff

background #000

border-radius 6px

a

padding 6px 15px

border 1px solid transparent

border-radius 6px

cursor pointer

&.active,

&:hover

color #44a5f2

select

width 80px

height 35px

border 1px solid transparent

outline none

.select-box

relative()

&:hover

.faculty-lsit

display block

li

color #000

&:hover

color #fff

background #44a5f2

.faculty-lsit

absolute(top 30px)

display none

width 150px

padding 5px 0

border 1px solid rgba(0, 0, 0, .3)

background #fff

border-radius 6px

z-index 1

li

padding 3px 0

margin 0

text-align center

.dialog-wrapper

absolute(left 0 top 0 right 0 bottom 0)

background-color rgba(0, 0, 0, .6)

z-index 99

.dialog

absolute(left 0 top 0 right 0 bottom 0)

width 300px

height 300px

margin auto

text-align center

line-height 60px

border-radius 6px

background-color #fff

li

width 100%

cursor pointer

&.active,

&:hover

color #fff

background-color rgba(97, 171, 241, .6)

.canvas

display inline-block

width 45%

height 300px

margin 15px 0

#sumNum

width 100%

height 450px

#itemNum

width 100%

height 280px

.empty

margin 30px 0

color #ddd

text-align center

font-size 2rem

</style>

统计功能图表绘制代码如下。

import echarts from 'echarts';

import liquidfill from './echarts-liquidfill.min.js';

import {isEmpty} from './common.js';

let count = {

teachNum: {

sum: 0,

data: [],

name: 'teachNum',

text: '教学讨论会',

color: '#c23531'

},

scientNum: {

sum: 0,

data: [],

name: 'scientNum',

text: '科研研讨会',

color: '#03a9f4'

},

salonNum: {

sum: 0,

data: [],

name: 'salonNum',

text: '学术沙龙',

color: '#8bc34a'

},

sumNum: {

sum: 0,

data: [],

name: 'sumNum',

text: '统计',

color: '#8bc34a',

sum: 0

}

};

let myCharts = {};

/\*

\* @method instance: 创建图表

\* @param {Array} ids: 需创建的图表 id

\* @return {void}

\*/

function instance(ids) {

ids.forEach((value, i) => {

// 实例化 echarts

myCharts[value] = echarts.init(document.getElementById(value));

})

}

// 横轴

let axis = ['1月', '2月', '3月', '4月', '5月', '6月', '7月', '8月', '9月', '10月', '11月', '12月'];

/\*

\* @method xAxis: 设置横轴

\* @param {Number} range: 横轴范围，1：上半年，2：下半年，其它：全年

\* @return {Array}

\*/

function xAxis(range) {

if(range == 1) {

return axis.slice(0, 6);

}else if(range == 2) {

return axis.slice(6);

}else {

return axis;

}

}

/\*

\* @method getData: 处理数据

\* @param {Object} opt: [{

\* startTime: String, 活动开始时间

\* type: String 活动信息类型

\* }]

\* @return {Object} count

\*/

function getData(opt) {

// console.log(opt)

let mon = '',

time,

type;

//清空数据

for(let k in count) {

count[k].data = [];

count[k].sum = 0;

}

opt.forEach(value => {

time = new Date(value.startTime),

mon = time.getMonth();

switch(value.type) {

case '1':

type = 'teachNum';

break;

case '2':

type = 'scientNum';

break;

case '3':

type = 'salonNum';

break;

}

if(opt.type == 2) {

mon -= 6;

}

count[type].data[mon] = count[type].data[mon] ? count[type].data[mon] + 1 : 1;

count.sumNum.data[mon] = count.sumNum.data[mon] ? count.sumNum.data[mon] + 1 : 1;

count[type].sum += 1;

++count.sumNum.sum;

})

count.sumNum.minCount = Math.min(count.teachNum.sum, count.scientNum.sum, count.salonNum.sum)

return count;

}

/\*

\* @method sumOption: 'sumNum' 柱状图

\* @param {Object} data: count

\* @param {String} type: 图表类型 'sumNum'

\* @param {Array} xAxis: 横坐标

\* @return {Object} 图表配置项

\*/

function sumOption(data, type, xAxis) {

let series = [];

for(let key in data) {

if(key === type) {

continue;

}

// console.log(data)

let value = data[key];

series.push({

name: value.text,

text: value.name,

type: 'bar',

stack: 'sum',

areaStyle: {

normal: {}

},

data: value.data

})

}

return {

title : {

text: data[type].text,

subtext: '',

padding: [0]

},

toolbox: {

show : true,

feature : {

mark : {show: true},

dataView : {

show: true,

readOnly: true,

optionToContent: (opt) => {

let axisData = opt.xAxis[0].data;

let series = opt.series;

let table = `<table>

<theader>

<th>

</th>

<th>

${series[0].text}

</th>

<th>

${series[1].text}

</th>

<th>

${series[2].text}

</th>

</theader>

<tbody>`;

for(let i = 0, len = axisData.length; i < len; ++i) {

table += `<tr>

<td>

${axisData[i]}

</td>

<td>

${series[0].data[i] || 0}

</td>

<td>

${series[1].data[i] || 0}

</td>

<td>

${series[2].data[i] || 0}

</td>

</tr>`;

}

table += '</tbody></table>';

return table;

}

},

magicType : {show: true, type: ['line', 'bar']},

restore : {show: true},

saveAsImage : {show: true}

}

},

calculable : false,

tooltip: {

trigger: 'axis'

},

legend: {

data: ['教学讨论会', '科研研讨会', '学术沙龙']

},

xAxis : [

{

type : 'category',

boundaryGap : true,

data : xAxis

}

],

yAxis : [

{

type : 'value'

}

],

series : series

}

}

/\*

\* @method liquid: 'itemNum' 水球图

\* @param {Object} opt: [{

\* salonNum: {

\* data: String,

\* name: String,

\* text: String

\* },

\* scientNum: {...},

\* teachNum: {...}

\* }]

\* @return {Object} 图表配置项

\*/

function liquid(opt) {

let series = [],

i = 1;

for(let key in opt) {

let value = opt[key];

series.push({

name: value.name,

text: value.text,

type: 'liquidFill',

// data: [(value.data / sum).toFixed(2)],

data: [value.data],

center: [`${i++\*30 - 10}%`, '35%'],

radius: '65%',

outline: {

show: false

},

label: {

normal: {

formatter: param => `${param.seriesName }\n\n${(param.value\*100).toFixed(0)}%`,

textStyle: {

fontSize: 20

}

}

}

})

}

return {

series: series

}

}

/\*

\* @method getPercent: 获取各项百分比

\* @param {Object} data: count

\* @return {Object} 图表配置项

\*/

function getPercent(data) {

let percent = {};

for(let key in data) {

if(key === 'sumNum') {

continue;

}

let itemData = 0;

// 确保除数 > 0

if(data.sumNum.sum > 0) {

itemData = (data[key].sum / data.sumNum.sum).toFixed(2);

}

percent[key] = {

text: data[key].name,

name: data[key].text,

data: itemData

}

}

return percent;

}

/\*

\* @method init: 初始化

\* @param {Array} ids: 需要创建的图表id

\* @param {Array} options: [{

\* startTime: String, 活动开始时间

\* type: String 活动信息类型

\* }]

\* @return {void}

\*/

export function init(ids, options) {

instance(ids);

let data = getData(options);

let axis = xAxis(options.type);

// 配置 options

for(let value in myCharts) {

let option = {};

// 处理数据

let percent = getPercent(data);

if(value === 'sumNum'){

option = sumOption(data, 'sumNum', axis);

}else if(value === 'itemNum'){

option = liquid(percent)

}

myCharts[value].setOption(option);

}

}

/\*

\* @method refresh: 更新图表

\* @param {Object} options: [{

\* startTime: String, 活动开始时间

\* type: String 活动信息类型

\* }]

\* @return {void}

\*/

export function refresh(options) {

if(isEmpty(myCharts)) {

return;

}

console.log(myCharts + ' myCharts')

let data = getData(options);

let axis = xAxis(options.type);

let percent = getPercent(data);

for(let key in myCharts) {

let opt = myCharts[key].getOption();

if(key === 'sumNum') {

opt.series.forEach(value => {

value.data = data[value.text].data;

})

opt.xAxis[0].data = axis;

}else {

opt.series.forEach(value => {

let fixData = percent[value.text].data;

value.data = [fixData];

})

}

myCharts[key].setOption(opt);

}

}

路由代码如下。

import Vue from 'vue';

import VueRouter from 'vue-router';

import Index from '../pages/index.vue';

import NotFound from '../pages/404.vue';

import SignIn from '../pages/signin.vue';

import Edit from '../pages/edit.vue';

import Article from '../pages/article.vue';

import Personal from '../pages/personal.vue';

import User from '../pages/user.vue';

import PerArticle from '../components/per-article.vue';

import Statistics from '../pages/statistics.vue';

import Chart from '../components/chart.vue';

import Form from '../components/form.vue';

import Modify from '../components/modifyPwd.vue';

import Comments from '../components/comment.vue';

import {get} from '../assets/cookieUtil';

const routes = [{

path: '/', component: Index, name: 'index'

}, {

path: '/index', component: Index, name: 'index1'

}, {

path: '/signin', component: SignIn, name: 'signin'

}, {

path: '/article', component: Article, name: 'article',

}, {

path: '/user/modify/:artId', component: Edit, name: 'articleEdit',

}, {

path: '/main', name: 'main',

redirect: to => {

const {hash, params, query} = to;

let user = get('user');

if(user) {

return '/personal/' + user + '/article';

}else {

return {name: 'signin'}

}

}

}, {

path: '/edit/:id', name: 'edit',

redirect: to => {

const {hash, params, query} = to;

if(params.id) {

return '/user/edit/';

}else {

return {name: 'signin'}

}

}

}, {

path: '/user/edit', component: Edit, name: 'userEdit'

},

{

path: '/personal/:id', component: Personal, name: 'personal',

children: [{

path: 'article',

component: PerArticle,

name: 'article'

}, {

path: 'modify',

component: Modify,

name: 'modify'

}]

}, {

path: '/manage/user', component: User, name: 'user'

}, {

path: '/statistics/:id', component: Statistics, name: 'statistics',

children: [{

path: 'charts',

component: Chart,

name: 'chart'

}, {

path: 'form',

component: Form,

name: 'form'

}]

}, {

path: '/notfound', component: NotFound, name: 'notfound'

}, {

path: '/comments', component: Comments, name: 'comment'

}, {

path: '\*', redirect: '/'

}];

Vue.use(VueRouter);

export default new VueRouter({

mode: 'history',

routes: routes,

linkActiveClass: 'active'

})

服务端代码如下。

const express = require('express');

const router = express.Router();

const db = require('./db');

const fs = require('fs');

const path = require('path');

const crypto = require('crypto');

const session = require('express-session');

const resolve = file => path.resolve(\_\_dirname, file);

const fn = () => {};

//获取页脚信息

router.get('/getFooterLink', (req, res) => {

db.FootLink.find(null, (err, links) => {

if(err) {

res.send({state: 1, msg: '查询失败！'});

}else {

res.send({state: 0, data: links});

}

})

})

// 获取首页文章类型

router.get('/getTypeLists', (req, res) => {

db.Search.find(null, (err, lists) => {

if(err) {

res.send({state: 1, msg: '查询失败！'});

}else {

res.send({state: 0, data: lists});

}

})

})

// 获取首页学院信息

router.get('/getAcademyLists', (req, res) => {

if(req.query.id) {

db.Academy.find({index: req.query.id}, (err, lists) => {

if(err) {

res.send({state: 1, msg: '查询失败！'});

}else {

res.send({state: 0, data: lists});

}

})

}else {

db.Academy.find(null, (err, lists) => {

if(err) {

res.send({state: 1, msg: '查询失败！'});

}else {

res.send({state: 0, data: lists});

}

})

}

})

// 获取首页教研室类型

router.get('/getFacultiesLists/:id', (req, res) => {

db.Academy.findOne({index: req.params.id}, (err, doc) => {

if(err) {

res.send({state: 1, msg: '查询失败！'});

}else {

res.send({state: 0, data: doc.staff});

}

})

})

// 登录

router.post('/signin', (req, res) => {

let {id, pwd} = req.body;

// console.log(pwd)

db.User.findOne({account: id}, (err, user) => {

if(err) {

res.send({state: 1, msg: '查询失败！'});

}else if(!user){

res.send({state: 2, msg: '账号不存在!'});

}else {

let hash = crypto.createHash('sha256');

hash.update(pwd);

pwd = hash.digest(pwd);

if(pwd.toString('hex') === user.pwd) {

req.session.regenerate((err) => {

if(err) {

res.send({state: 4, msg: '登录失败，请重新尝试'})

}else {

req.session.user = user;

res.cookie('username', user.name);

res.send({state: 0, data: {id: id, rank: user.rank, faculty: user.faculty}});

}

})

}else {

res.send({state: 3, msg: '密码错误!'});

}

}

})

})

// 登出

router.get('/signout', (req, res) => {

req.session.user = null;

res.send({state: 0, msg: '登出成功'})

})

// 创建文章

router.post('/edit/create', (req, res) => {

db.Article.findOne({title: req.body.form.title, author: req.body.form.author}, (err, doc) => {

if(err) {

res.send({state: 3, msg: '操作失败'});

}else if(doc) {

res.send({state: 1, msg: '标题已存在！'});

}else {

db.Article.create(req.body.form, (err, article) => {

if(err) {

res.send({state: 2, msg: '发布失败，请重试！'});

}

res.send({state: 0, data: {id: article.\_id}});

})

}

})

})

// 修改文章

router.put('/edit/modify', (req, res) => {

let form = req.body.form;

db.Article.findByIdAndUpdate(req.body.id, {$set: {

url: form.url,

author: form.author,

title: form.title,

abs: form.abs,

time: form.time,

heldTime: form.heldTime,

endTime: form.endTime,

address: form.address,

unit: form.unit,

explain: form.explain,

content: form.content,

enclosure: form.enclosure,

faculty: form.faculty,

type: form.type,

participator: form.participator

}}, (err, doc) => {

if(err) {

res.send({state: 1, msg: '操作失败'});

}else {

res.send({state: 0, data: {id: doc.\_id}});

}

})

})

// 获取修改的文章

router.get('/edit/article/:id', (req, res) => {

db.Article.find({\_id: req.params.id}, (err, doc) => {

if(err) {

res.send({state: 1, msg: '查询失败！'})

}

if(!doc.length) {

res.send({state: 2, msg: '文章不存在！'})

}else {

res.send({state: 0, data: doc[0]})

}

})

})

// 删除文章

router.delete('/edit/delete/:id', (req, res) => {

db.Article.remove({\_id: req.params.id}, (err, doc) => {

if(err) {

res.send({state: 1, msg: '操作失败！'})

}else {

res.send({state: 0, msg: '删除成功！'});

}

})

})

// 获取文章详情

router.get('/article/:id', (req, res) => {

db.Article.findById({\_id: req.params.id}, (err, doc) => {

if(err) {

res.send({state: 2, msg: '查询失败！'})

}

if(doc) {

res.send({state: 0, data: doc});

}else {

res.send({state: 1, msg: '文章不存在！'});

}

})

})

// 获取个人发布文章

router.get('/articles/user/:id', (req, res) => {

db.Article.find({author: req.params.id}, (err, doc) => {

if(err) {

res.send({state: 2, msg: '查询失败！'})

}

if(doc.length) {

res.send({state: 0, data: doc});

}else {

res.send({state: 1, msg: '文章不存在！'});

}

})

})

// 获取所有文章

router.get('/articles/all', (req, res, next) => {

console.log(req.query)

let academy = req.query.faculty.charAt(0),

faculty = req.query.faculty.charAt(2),

type = req.query.type,

obj = {},

query = '',

total = 0;

const size = req.query.pageSize \* 1;

if(academy == 0) {

if(type != 0) {

obj.type = type

}else {

obj = null;

}

}else {

let faCondition = faculty == 0 ? new RegExp(`^${academy}-`) : req.query.faculty;

if(type != 0) {

obj.type = type;

}

obj.faculty = faCondition;

}

// 查询总量

db.Article.count(obj, (err, sum) => {

if(err) {

res.send({state: 2, msg: '查询失败！'});

}else {

total = sum;

if(total > 0) {

pagedQuery(obj, req.query.page, size, total, res);

}else {

res.send({state: 0, data: {lists: [], total: total}})

}

}

})

})

// 分页查询

function pagedQuery(obj, page, size, total, res) {

db.Article.find(obj)

.skip((page - 1) \* size)

.limit(size)

.sort({startTime: -1})

.exec((err, doc) => {

if(err) {

res.send({state: 1, msg: '查询失败！'});

}else {

res.send({state: 0, data: {lists: doc, total: total}});

}

})

}

// 统计图表

router.get('/count/:id/:tab/:year/:time', (req, res) => {

console.log(req.params);

let params = req.params,

sTime = new Date(),

eTime = new Date();

if(params.time == 0) {

sTime = new Date(`${params.year}-01-01`);

eTime = new Date(`${params.year}-12-31`);

}else if(params.time == 1) {

sTime = new Date(`${params.year}-01-01`);

eTime = new Date(`${params.year}-07-01`);

}else {

sTime = new Date(`${params.year}-07-01`);

eTime = new Date(`${params.year \* 1 + 1}-01-01`);

}

db.Article.find({title: '阿尔法'}, (err, doc) => {

console.log(doc[0])

})

if(params.tab == 0) {

db.Article.find({participator: {$in: [params.id]}, $and: [{startTime: {$gt: sTime}}, {startTime: {$lt: eTime}}]}, (err, doc) => {

console.log(doc)

if(err) {

res.send({state: 1, msg: '查询失败！'})

}else {

let count = filter(doc);

res.send({state: 0, data: count})

}

})

}else if(params.tab == 1) {

db.Article.find({faculty: {$in: [params.id]} ,$and: [{startTime: {$gt: sTime}}, {startTime: {$lt: eTime}}]}, (err, doc) => {

console.log(doc.length)

if(err) {

res.send({state: 1, msg: '查询失败！'})

}else {

console.log(doc[0])

let count = filter(doc);

res.send({state: 0, data: count})

}

})

}

})

/\*

\* @method filter: 过滤图表数据

\* @param {Array} doc: Article Entity

\* @return {Array} [{

\* startTime: String, 活动开始时间

\* type: String 活动信息类型

\* }]

\*/

function filter(doc) {

if(doc.length === 0) {

return {};

}

let result = [];

doc.forEach(value => {

let tmp = {};

tmp.startTime = value.startTime;

tmp.type = value.type;

result.push(tmp);

})

return result;

}

// 获取当前用户个人信息

router.get('/userManage/selfInfo', (req, res) => {

console.log(req.session)

if(req.session.user) {

db.User.find({name: req.query.name}, (err, doc) => {

if(err) {

res.send({state: 1, msg: '操作失败'});

}

if(!doc.length) {

res.send({state: 2, msg: '账号不存在'});

}else {

res.send({state: 0, data: {id: doc[0].name, rank: doc[0].rank, faculty: doc[0].faculty}});

}

})

}

})

// 获取用户信息

router.get('/userManage/info', (req, res) => {

db.User.find((err, doc) => {

if(err) {

res.send({state: 1, msg: '查询失败！'});

}else {

doc.forEach(value => {

//清空pwd，不传递真实pwd

value.pwd = '';

})

res.send({state: 0, data: doc});

}

})

})

// 添加用户

router.post('/userManage/add', (req, res) => {

dbUser.create(req.body, (err, doc) => {

if(err) {

res.send({state: 1, msg: '操作失败'})

}else {

res.send({state: 0, data: doc})

}

})

})

// 修改用户信息

router.post('/userManage/modify', (req, res) => {

console.log(req.body)

db.User.update({\_id: req.body.\_id}, {$set: {

name: req.body.name,

account: req.body.account,

rank: req.body.rank,

faculty: req.body.faculty,

title: req.body.title

}}, (err, doc) => {

if(err) {

res.send({state: 1, msg: err})

}else {

res.send({state: 0, msg: '修改成功！'})

}

})

})

//TODO

// 删除用户

router.delete('/userManage/delete/:id', (req, res) => {

console.log(req.session.user)

db.User.remove({account: req.params.id}, (err, doc) => {

if(err) {

res.send({state: 1, msg: err})

}else {

db.User.find(null, (err, users)=> {

if(err) {

res.send({state: 1, msg: err})

}else {

res.send({state: 0, msg: '删除成功！'})

}

})

}

})

})

// 获取统计单位可选项

router.get('/count/unitText', (req, res) => {

db.Academy.find({index: req.query.faculty}, (err, doc) => {

if(err) {

res.send({state: 1, msg: err})

}else {

res.send({state: 0, data: doc[0].staff})

}

})

})

module.exports = router;