



**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 |  | 2014年4月 15 日 | 论文答辩日期 |  | 2014年5月 9 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 前后端分离 非关系型数据库

Teaching Activity Management System

He Xuehuan

(College of Mathematics and information, 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 Front end separation Non relational database

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

该系统基于是 B/S 架构的应用，主要开发语言为 JavaScrpt,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

Vue.js是一个构建WEB界面的JavaScript库，它的理念是数据驱动（江庆，叶浩荣，2016）。与其他重量级框架不同的是，Vue采用自底向上增量开发的设计，提供更加简洁、更易于理解的API。通过API提供高效的数据绑定和灵活的组件系统。

Vue的核心库只关注视图层，它不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与[单文件组件](https://cn.vuejs.org/v2/guide/single-file-components.html)和 [Vue 生态系统支持的库](https://github.com/vuejs/awesome-vue" \l "libraries--plugins)结合使用时，Vue也完全能够为像本系统这种复杂的单页应用程序提供驱动。

Vue.js能够成为当下很火的一个JavaScript MVVM库，是由于：

1. MVVM数据绑定

它是以数据驱动和组件化的思想构建的。MVVM的本质是通过数据绑定链View和Model，让数据的变化自动映射为视图的更新。为页面元素绑定指令，当指令对象观测的数据发生变化是，指令会对所绑定的节点元素执行响应的DOM操作。基于指令的数据绑定使DOM操作被合理的封装起来，开发过程中，开发人员无需进行DOM操作；业务代码只需要涉及模板和对数据状态的操作即可，使得开发效率和可维护性大大提高。

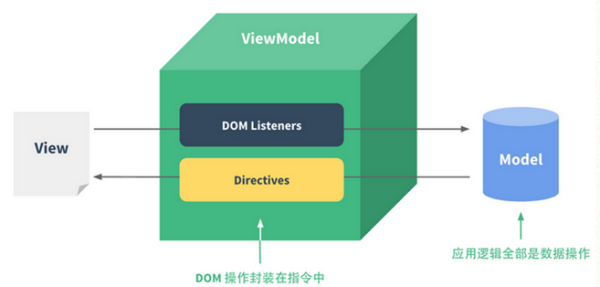


图1 Vue.js数据流图

1. 数据观测的实现

对于数据观测的实现，Vue.js的实现原理和Angular有着本质的不同。Angular采用的是脏检查机制，每个指令都有一个叫watcher的对象用以观察数据。任一数据发生变动时，当前作用域上的每个watcher偶读需被重新求值，当watcher数量庞大是，不可避免的影响应用程序的性能。其次，当数据变动时，框架不能主动侦测到变化的发生，而是需要手动的触发digest cycle，才能使相应的DOM更新被触发。

Vue.js采用的则是基于依赖收集的观测机制，与Angular等的脏检查机制不同。依赖收集的优点在于，能够主动、精确地追踪数据变化，不存在像脏检查那样的弊端

1. 组件系统

在应用开发中，为了分工、可复用和可维护性，需要将应用抽象为多个模块（组件）。而Vue.js的一个核心概念是组件化，每个Vue.js应用都是围绕组件开发的。因此，Vue.js提供了完善的定义、复用和嵌套组件的API，同一颗组件树之间可以通过这些API进行同行，使开发者能够像搭积木一样用组件拼出整个应用界面。

在发布之初，Vue.js原本是着眼于轻量的嵌入式使用场景。在今天，其轻量、高性能的特点，以及完备的组件系统，使得Vue.js在保留了其简洁API设计的同时，也完全能够担当起如本系统这样复杂的大型应用的开发。

## 2.2 Stylus

Stylus是一种创新性的CSS预处理器，提供了一个高效，动态和表达式方法编译生成CSS。Stylus使用Node.js构建，它的设计受到Sass和LESS的影响，被认为是第四个最常用的 CSS 预处理语法。

CSS预处理器是基于CSS扩展了一套属于自己的DSL（domain specific language），解决书写CSS时难以解决的问题：

1. 语法不够强大，比如无法嵌套书写导致模块化开发中需要书写很多重复的选择器；
2. 没有变量和合理的样式复用机制，使得逻辑上相关的属性值必须以字面量的形式重复输出，导致难以维护。

CSS的静态化，使得开发者需要一个真正能够提高开发效率的工具。这就决定了CSS 预处理器的主要目标：提供CSS缺失的样式层复用机制、减少冗余代码，提高样式代码的可维护性。

CSS预处理器的DSL并不存在浏览器兼容性的问题，因为它们能够将所写的代码编译成能够被浏览器使用的、常规的CSS。

Stylus虽然在广泛意义上人气不如其它的CSS预处理器，如Sass、LESS，但Stylus 的功能更加强壮，由于它是使用Node.js构建的，和js的联系也更加紧密。

Stylus 最大的好处是让CSS编写逻辑性更加强；利用缩进、空格和换行减少输入的字符，使得文件更加有条理，简洁清晰，同时又兼容CSS风格代码。变量无疑为CSS增加了一种有效的复用方式，减少了原来在CSS中无法避免的重复「硬编码」。

## 2.3 Echart

当下随着大数据热潮的到来，数据可视化作为一个新兴的领域，受到了学术界和工业界的重视。从可视分析、数据新闻到商业报表，各个领域都在越来越多的使用它。既然是数据可视化，说明数据是主体，可视化只是将数据以可视的形式表达的手段。

ECharts是国内一款非常优秀的可视化图表控件，它底层依赖轻量级的Canvas类库ZRender，提供直观，生动，可交互，可高度个性化定制的数据可视化图表。（王子毅，张春海，2016）

ECharts是一个纯JavaScript图标库，可以流畅的运行在PC和移动设备上，兼容当前绝大部分浏览器。（王雄兵，田茂，范亮等，2017）

新版本的 Echarts 带来了许多的新特性：

1. 提供丰富的图表类型

如折线图、柱状图、散点图等，支持图与图之间的混搭。

1. 支持多个坐标系

支持直角坐标系、极坐标系、地理坐标系，可跨坐标系存在。

1. 轻量

加载速度是网站的一个重要的性能指标之一。正如前文提到的，Echarts 能够在移动设备上流畅的运行，这意味着 Echarts 这库体积需要足够小。压缩后只有 172 KB。

1. 深度的交互式数据探索

随着浏览器性能的不断提高，用户对网站的要求也越来越高，好的交互设计能够吸引更多的用户，并得到用户的肯定。Echarts 一直在交互的路上前进，支持在数据筛选、视图缩放、展示细节等交互能力。

1. 大数据量的展示

借助 Canvas 的能力，Echarts在散点图中能够轻松展现上万甚至上十万的数据。

1. 多维数据的支持以及丰富的视觉编码手段
2. 动态数据
3. 绚丽的特效

## 2.4 Webpack

伴随着移动互联的大潮，当今越来越多的网站已经从网页模式进化到了 Webapp 模式。它们运行在现代的高级浏览器里，使用HTML5、 CSS3、 ES6等更新的技术来开发丰富的功能，现今很多的网页已经不仅仅是完成浏览的基本需求，而是可以看做是功能丰富的应用。并且本系统是一个单页面应用，每一个视图通过异步的方式加载，导致页面初始化和使用过程中会加载越来越多的JavaScript代码，给前端开发的流程和资源组织带来了巨大的挑战。

前端产品的交互是基于浏览器，资源通过增量加载方式运行到浏览器端，在开发环境组织好这些碎片化的代码和资源，并且保证它们在浏览器能够快速、优雅的加载和更新，就需要一个模块化系统，把复杂的程序细化为小的文件。

模块系统主要解决模块的定义、依赖和导出。已有的模块系统：① <script> 标签 ② CommonJS ③ AMD

以上提到的模块仅仅是指JavaScript模块文件，但在前端开发中，还涉及到CSS、图片、字体、HTML 模板等众多资源。如果这些资源都可以视作模块，通过 require 的方式加载，将带来优雅的开发体验。

Webpack是一个前端资源加载和打包工具，它将一切静态资源都当做模块，模块之间可以相互依赖，Webpack对其进行统一管理。（魏云申，2016）它把项目当做一个整体，将许多松散的模块按照依赖和规则打包成符合生产环境部署的前端资源。还可以将按需加载的模块进行代码分隔，等到实际需要的时候再异步加载。在编译的时候，对整个代码进行静态分析，分析出各个模块的类型和依赖关系，将不同类型的模块交给适配的加载器loader处理。通过loader的转换，任何形式的资源都可以视作模块，如 CommonJs 模块、AMD 模块、ES6 模块、CSS、图片、JSON、Coffescript、LESS等。

## 2.5 Node.js

Node.js是基于Chrome V8引擎的JavaScript运行环境，能够款速构建可扩展的应用程序。V8 引擎是目前速度最快的JavaScript引擎，JavaScript程序在V8引擎下的运行速度媲美二进制程序，因此 Node.js 程序的即时编译执行速度与本地代码的执行速度相当。

目前的编程技术最大的浪费来自等待I/O操作的完成。Node.js最初的被开发的目标是成为一个基于事件驱动、非阻塞I/O的Web服务器，以达到更高的性能。

Node.js的目标是快速搭建可扩展的网络应用。（彭娜，2013）Node.js的优点十分明显：

1. 异步I/O

服务器的并发连接最大数量往往是 Web 应用程序架构的瓶颈，传统的后台语言采用的是阻塞式函数，阻塞式函数会导致资源浪费和时间延迟。

在Node.js中，绝大多数的操作都是以异步的方式实现的，在底层有许多异步I/O的API，从文件读取到系统操作均是如此。开发者可以在语言层面上很自然的进行并行I/O操作，编程模式上极大的提升效率。

Node.js非阻塞模式的I\O处理，使其拥有在相对低系统资源耗用下的高性能与出众的负载能力。Node.js轻量高效，可以认为是数据密集型分布式部署环境下的实时应用系统的完美解决方案。

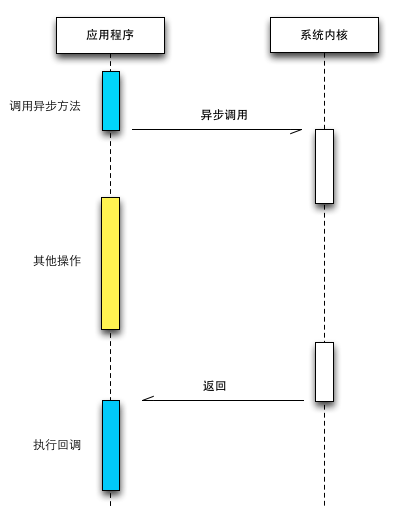


图2 异步回调

1. 事件与回调函数

事件的编程方式具有轻量级、松耦合、只关注事务等优势。在多个异步任务的场景中，事件之间是相互独立的，如何协作成了一个问题。

在Javascrip中，回调函数无处不在，通过异步回调，将函数作为对象传递给方法作为实参调用，实现非阻塞操作；Node.js的设计思想是以事件驱动为核心，提供的大多数API都是基于事件的、异步的风格，为网络服务而设计。优势在于，充分利用了系统资源，执行代码不会因阻塞而等待，有限的资源可以用于其他的任务。这样的设计非常适合于后端的网络服务编程，Node.js的目标也在于此。

1. 单线程

Node.js使用单线程、单进程运行，不需要多线程的资源占用和多进程的上下文切换。Node.js构建的 WEB服务器因为是单线程的，一个进程中保持一个活跃的请求处理线程。（王金龙，宋斌，丁锐，2014）采用事件驱动的设计，在单线程内部高效地维护事件队列。

1. 跨平台

Node.js起初只可以在Linux平台上运行，v6.0.0 版本之后，在操作系统和 Node.js上层模块之间构建了一层平台层架构，即 libuv（系统实现跨平台的基础组件），使 Node.js可以直接在Windows上运行，兼容Windows和\*nux平台。

事件驱动、异步编程，

1. 包管理器

Node.js使用了包管理器npm来对模块进行管理，包管理器npm是全球最大的开源库生态系统。Node.js内建了HTTP，Socket，TLS，FTP，DNS等网络应用模块，能快速开发出各种网络服务和网络服务器的组合，而不需要额外的服务器（如Apache、Tomcat、Ngnix等）支持。

鉴于Node.js跨平台、拥有丰富的包管理器，M Duque等人选择使用Node.js开发Micro-grid平台。（M Duque，E Cando，A Aguinage，F Llulluna等，）

## 2.6 MongoDB

[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）

[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq)无需使用高昂的数据库负荷就可以管理任何结构的、频繁变化的数据。（Hanen Abbes，Faiez Gargouri，2016）

由于[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq" \t "_blank)属于文档的数据库，它支持ad-hoc查询，索引，复制，和mapreduced聚集。（Titus，2017）

[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq)的设计理念就是基于分布式扩展的，它使用自动芬片机制能将数据库中的集合和文档分布在多个不同的数据库节点上。（孙思源，2014）

[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq)的特点和适用场景：

1. 实用性

[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq)是一个面向文档的数据库，它并不是关系型数据库，直接存取BSON；因为可以在文档中直接插入数组之类的复杂数据类型，并且文档的key和value不是固定的数据类型和大小，这意味着[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq)更加灵活，开发者在使用[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq" \t "_blank)时无须预定义关系型数据库中的”表”等数据库对象，设计数据库将变得非常方便，可以大大地提升开发进度。

1. 可用性和负载均衡

[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq)随着并发数的增加，无论是insert吞吐量、update还是select，其吞吐量在不同程度的下降，但较之Mysql的性能会好得多（王光磊，2011）

Jongseong Yoon 在曾对关系型数据库与非关系型数据库进行比较，讨论它们各自的优缺点。

[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq)在高可用和读负载均衡上的实现非常简洁和友好，[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq)自带了副本集的概念，通过设计适合自己业务的副本集和驱动程序，可以非常有效和方便地实现高可用，读负载均衡。而在其他数据库产品中想实现以上功能，往往需要额外安装复杂的中间件，大大提升了系统复杂度，故障排查难度和运维成本。

1. 扩展性

在扩展性方面，假设应用数据增长非常迅猛的话，通过不断地添加磁盘容量和内存容量往往是不现实的，而手工的分库分表又会带来非常繁重的工作量和技术复杂度。在扩展性上，[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq)通过自带的Mongos集群，只需要在适当的时候继续添加Mongo分片，就可以实现程序段自动水平扩展和路由，一方面缓解单个节点的读写压力，另外一方面可有效地均衡磁盘容量的使用情况。整个mongos集群对应用层完全透明，并可完美地做到各个Mongos集群组件的高可用性。

此外，在一个多机[MongoDB](http://baike.baidu.com/link?url=gWFYOJ6GwclQFR70ZjCpeKoilmoVHkhLlmDUKBQuCUYLmEudt7WeP7SuWB44XlDnIk1amyVURwcuTKWcofb6Yq)系统，文件可以分发和复制多次机器之间的透明，从而有效地创建一个负载平衡和容错系统。（Milorad Pantelija Stevic，2013）

# 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（R）；还为管理员提供增删改功能（CUD），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。

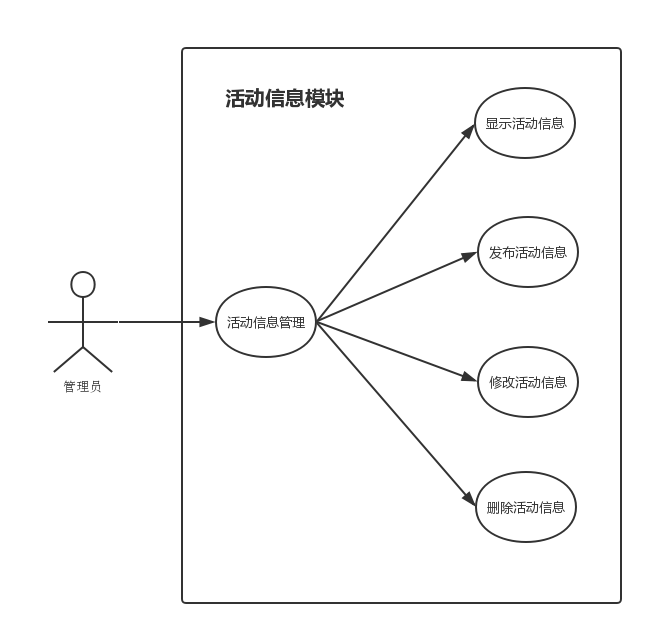


图3 活动信息管理模块用例图

### 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。

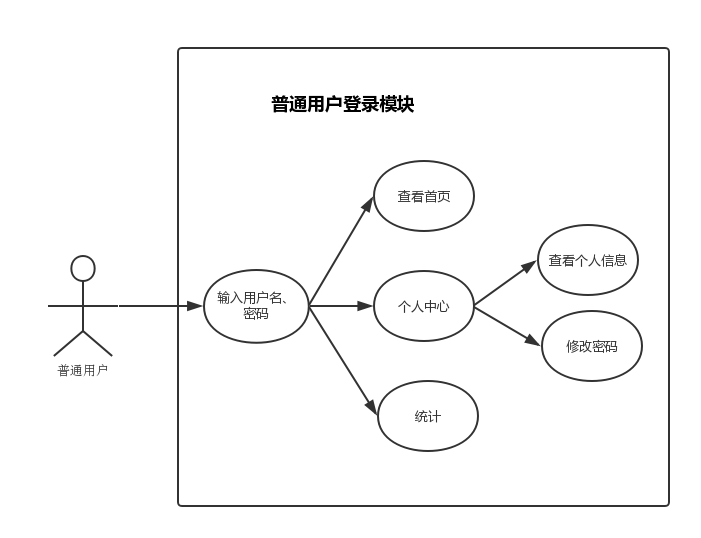


图4 普通用户登录用例图

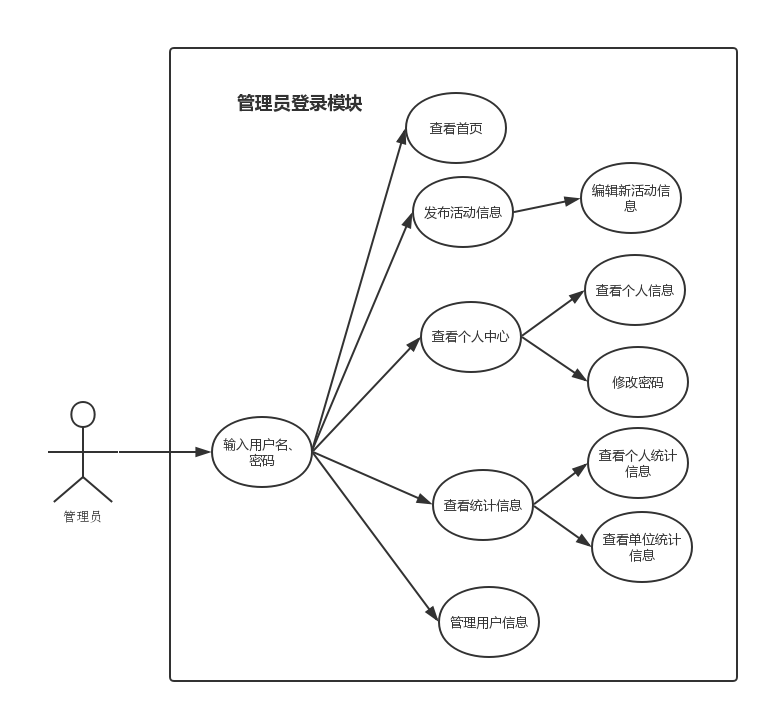


图5 管理员登录用例图

## 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

本系统页面访问流畅，页面中的链接均能正常访问，访问不存在的地址时重定向到首页。对于错误给予用户友好的提示。

## 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。

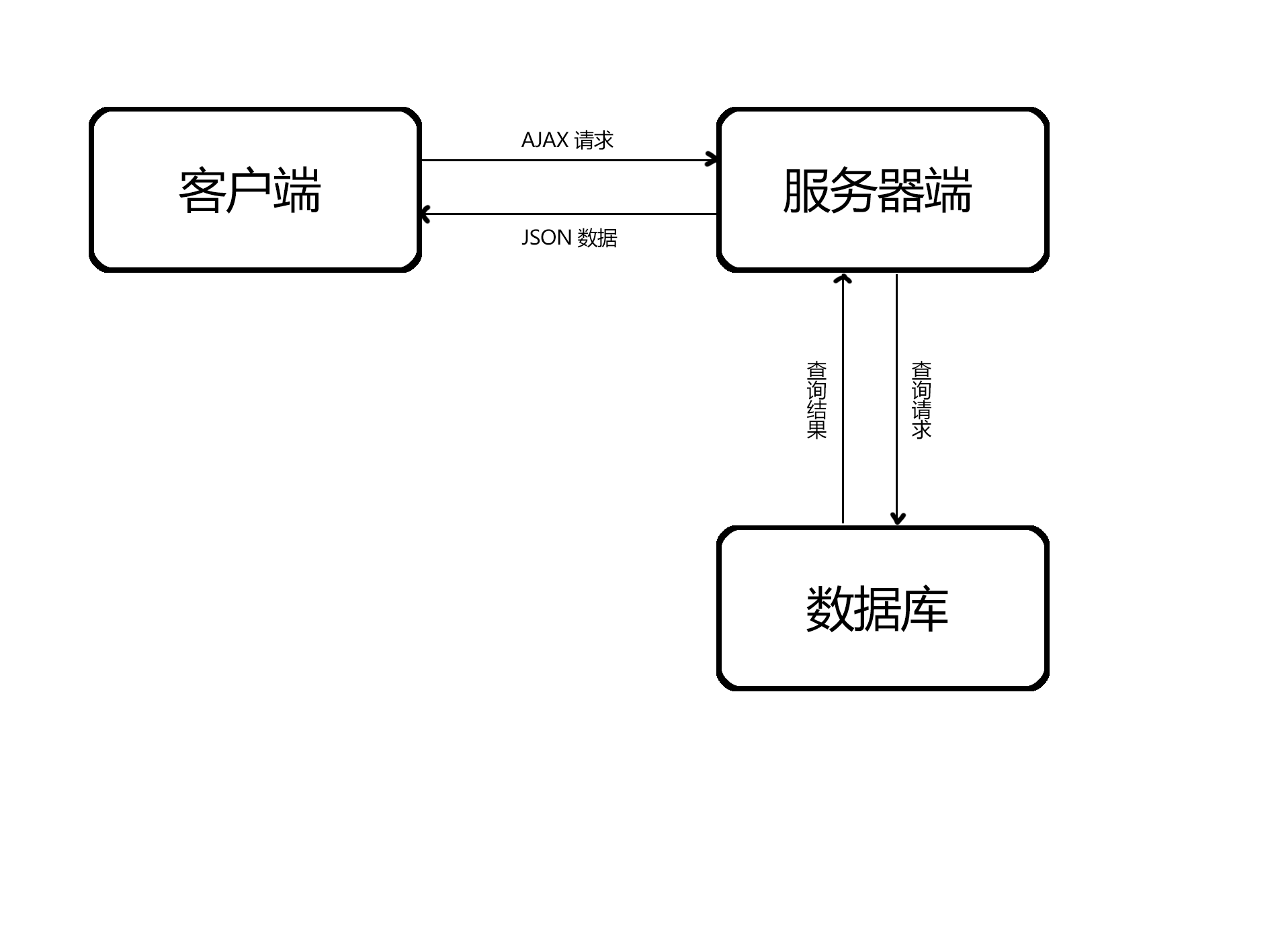


图6 系统数据流图

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：

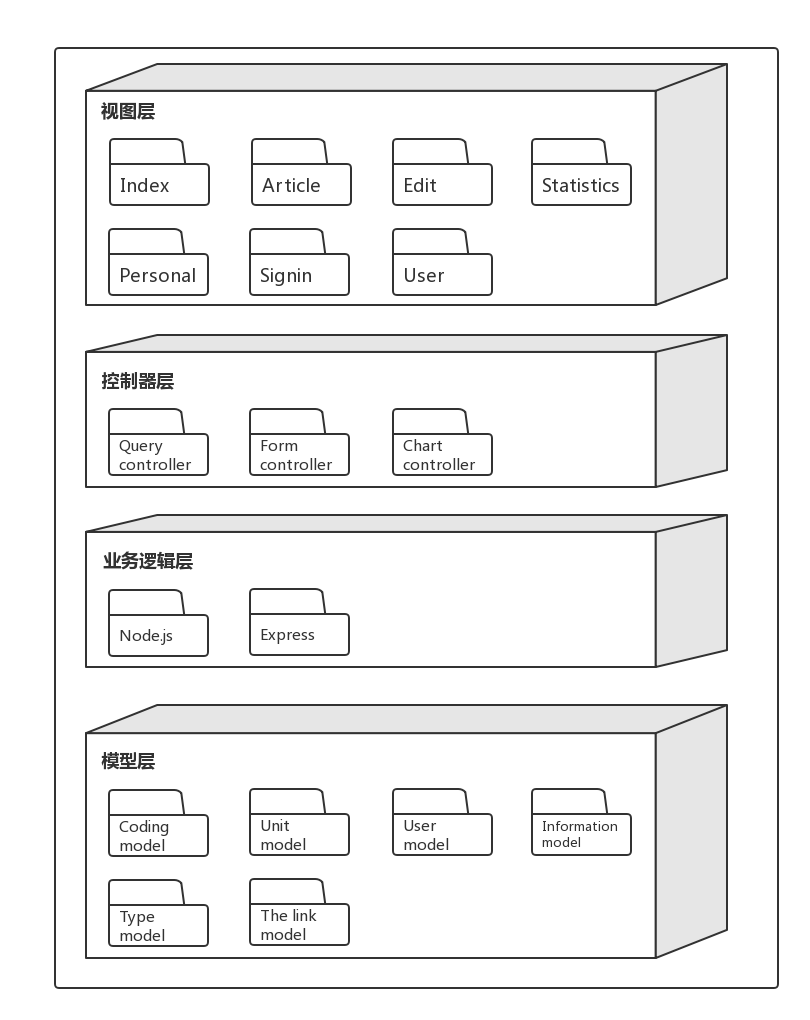


图7 系统整体架构图

### 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。

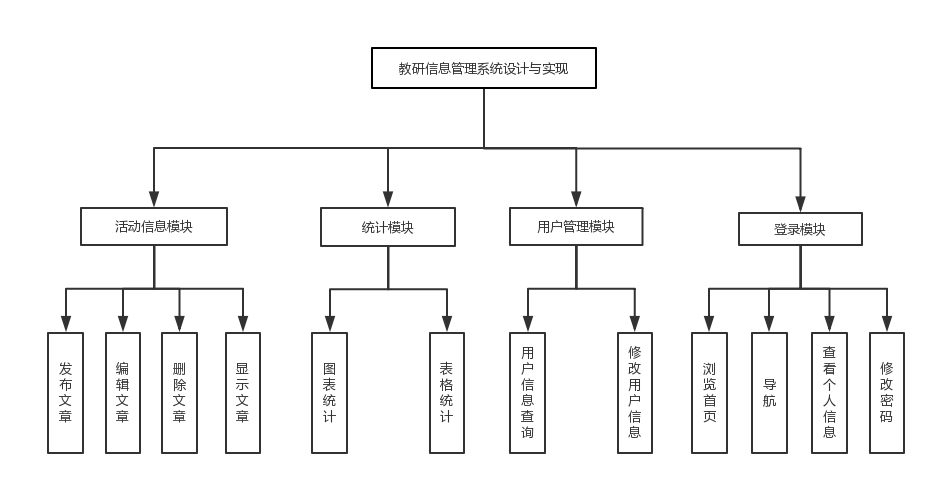


图8 系统功能结构图

## 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

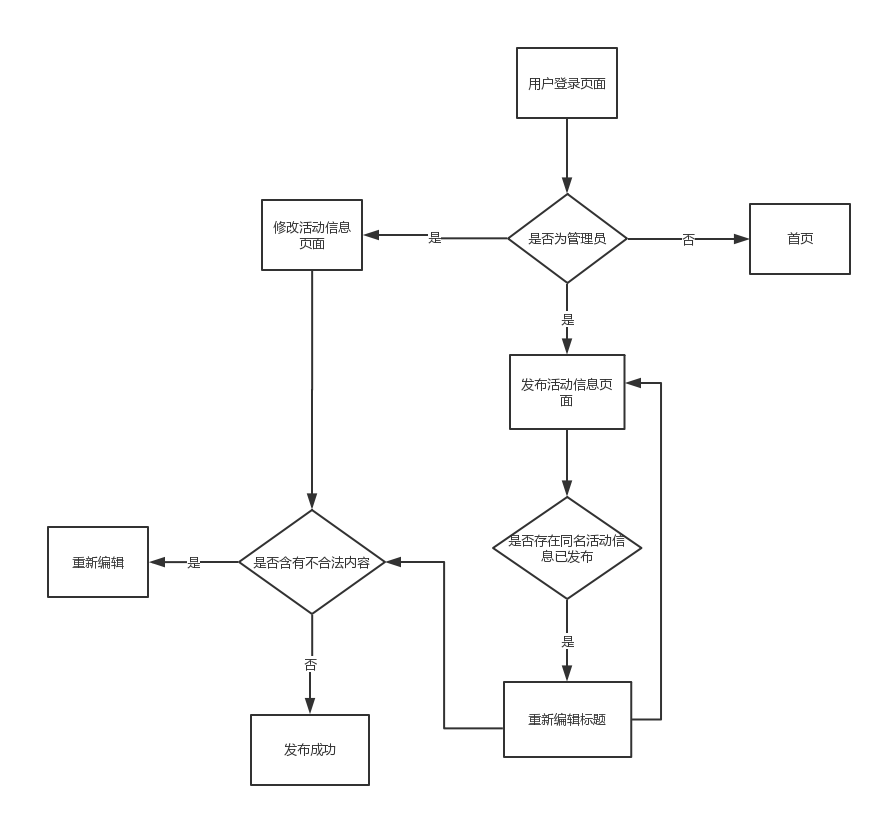


图9 发布教研活动信息流程图

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

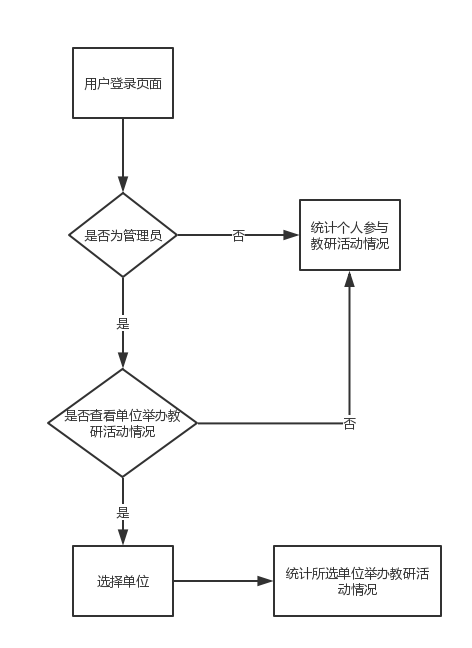


图10 统计教研活动流程图

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

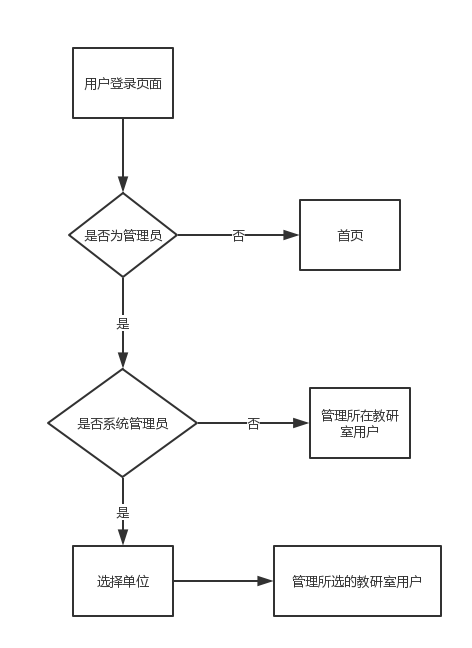


图11 用户信息操作流程图

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（CRUD）所在教研室的用户；

系统管理员可操作（CRUD）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）

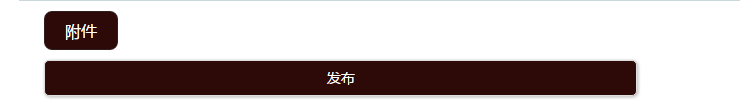


图17 系统编辑页面发布状态（下）

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。

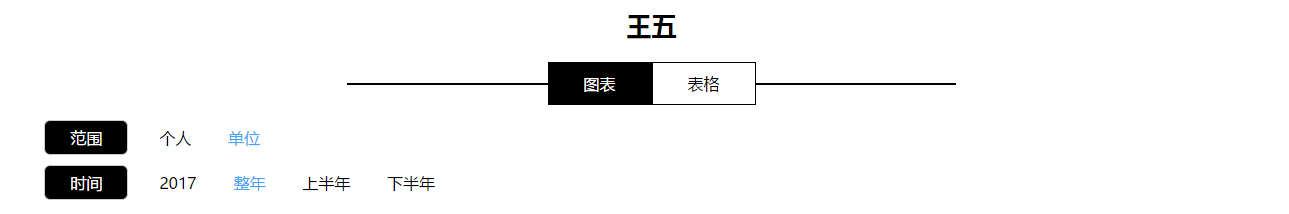


图20 管理员统计页面选择组件

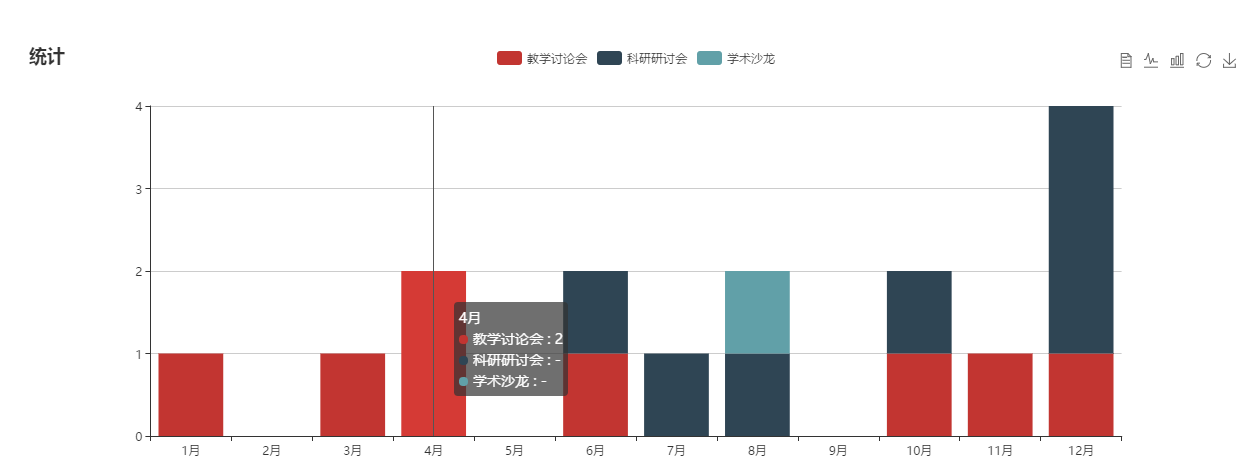


图21 统计页面柱状图

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。

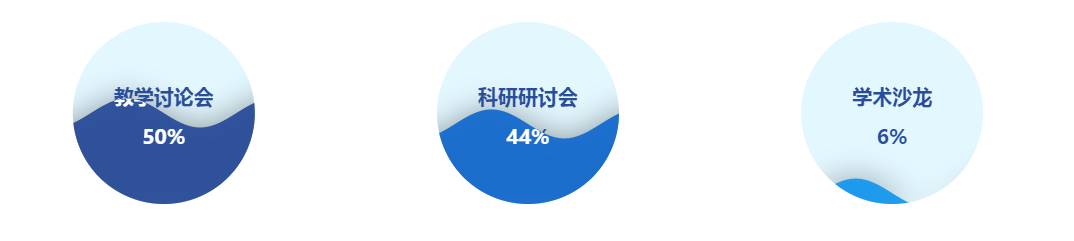


图22 统计页面水球图

## 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。

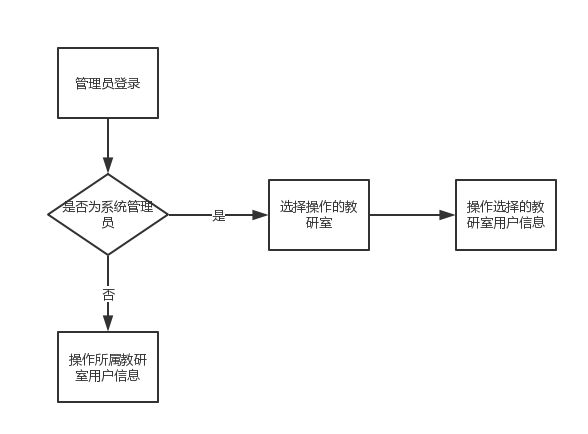


图23 用户管理流程图



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

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。