**摘要**

# 绪论

## 研究背景及意义

随着互联网技术的发展，成熟，以及手机、计算机等终端的日益普及，各事业单位部门都基本上配置了办公电脑。信息化时代的来临，增加的新的信息管理与获取的方式，相比于传统通过纸质文件的信息储存、分享、获取与管理的方式，与互联网结合的方式更加快捷，简便，也更具多样化、信息化、高效化、专业化。

同时也是响应，国家档案局于2016年4月1日颁发的《全国档案事业发展"十三五"规划纲要》纲要中提出的指导思想和目标:到2020年, 科学规划，资源合理布局，加快档案管理体系的建设,，实现以信息化为核心的档案管理现代化。

在国家大数据发展战略与“互联网+”快速发展的背景下，必将影响高校工作的理念、技术、方法及模式。在此环境下，针对高校的学术活动的管理与宣传，需要创造一种管理理念与模式，使之与互联网技术的发展结合更加紧密。

学术活动的举办、交流有助于我们了解前沿技术的发展、与他人分享研究成果、向同行进行介绍、总结，并于他人讨论，验证自己的想法、使用的方法是否正确，也可以让同行提出一些建议，有助于我们开拓科研的思路，与他人进行思想的碰撞，激发灵感。在听他人汇报的过程中，进行自我反省，学习他人的优点，重视自己的不足。

我国的高等院校是信息技术人才的培养基地，在高等院校中，进行举办学术活动有利于培养大学生的创新意识、科研精神，在使我国高等院校的学生对学术科研内容有更加深刻的理解的同时，也让我国高等院校的学生对所学的知识有更深刻的理解的同时，有利于将所学知识与实际相结合，并且灵活应用，切实提升大学生的动手能力、实践能力。在高等院校，教师在课堂上所传授的知识仅仅只是基础，只是书本上的知识，学术活动的举办、交流可加深大学生对本专业知识的了解与认识，从而提高学习的兴趣，激发学习的主动性、积极性。

随着经济科技的发展，高等院校学术交流活动也变得日益频繁，活动和举办频率的不断增加，学术活动的规模也越来越大，信息之间的相互关系也越来越紧密。学术活动管理系统是高等院校的日常管理工作之一，也是高校管理中的一项重要内容，涉及到人力、物力、财力、时间等诸多方面。华南农业大学共有26学院，95个本科专业[华南农业大学学校简介（2018年9月）](https://baike.baidu.com/reference/409310/68398EtATXoeJG2yE0sRmTHsJdgFRIqwNJNH4Vr6cfNIQPRG46mhEKx633v2MGkFkVnAQ3gGm5p5bEF0Qu_H12A" \t "_blank)，如果继续使用传统的人工管理与一般的计算机信息储存，将好费许多的人力物力，效率低，容易出错。一旦时间一长，数据、文件日积月累，使得信息的维护与查找变得困难，并且学术的研究成果难以向他人展示。因此对学术活动管理提出了更高的要求，以前的管理方式不再适用，需要开发新的学术活动那个管理系统，一方面为了实现学术活动的信息化管理，帮助管理人员从重复、繁琐的工作中解放出来，使得学术活动管理更具无纸化和规范化，另一方面可以借助互联网向他人宣传、展示学术活动的成果，以此提升他人对学术活动的兴趣，吸引他人来参加活动。

## 研究内容

本系统基于B/S结构，主要有前端应用程序，使用VUE框架进行搭建，后台系统采用Java编程语言进行编写，且以IntelliJ IDEA为开发工具，运用MySQL数据库技术，开发一个学术活动管理系统。该系统肩负是当某个学科的学者在一起讨论某个研究方向或课题的现状、发展和研究成果等内容的时候，需要将学术活动举办的过程，包括通知，进行记录，后期的管理、维护，并且具有一定的数据分析功能。

此系统的具体功能模块如下：

1. 用户管理：
2. 用户分为三类：
   * + 1. 游客（一般为教师和学生）：可以查看将要或已经举办的学术活动的具体情况，浏览每个活动的具体时间、地点、举办单位、活动题目、内容、主持人或报告人，以及其它相关信息、图片。
       2. 普通管理员：在登录后可以进行所在系的学术活动信息发布（活动未举行前）、活动信息修改、活动过程信息录入（如录入活动举行中拍摄的图片及其说明文字）。
       3. 系统管理员：登录后可以对普通管理员的账号和密码进行设置、可以对所有举办的学术活动进行统计（如按学院统计、按学期统计）、等。
3. 用户个人信息的设置、修改，包括邮箱设置。
4. 学术成果管理：

学术成果包括：论文、专利等。

1. 对外：供游客查看已获得的学术成果，浏览每个学术研究的研究人员、研究内容、取得成果、时间等信息。
2. 对内：

普通管理员对学术成果进行管理的权限如下：

* + - 1. 可以对学术成果根据学科分类。
      2. 增加、修改、删除学术成果，以及是否对外展示。

系统管理员对学术成果进行管理的权限如下：

* + - 1. 管理供游客浏览的学术成果模块。如置顶某项学术活动研究的成果。

1. 学科管理：
2. 只有普通管理员可以对学科进行管理，并且该管理员只能维护所属的学院的学科。

增加学科：填写学科所需的信息，如学科名称、描述等信息。

修改学科：可修改学科名称、描述等信息。

1. 学术活动管理：
2. 普通管理员可以对学术活动进行管理。
   * + 1. 发布学术活动：增加学术活动，需要填写具体时间、地点、举办单位、活动题目、内容、主持人或报告人等信息、删除学术活动、修改学术活动的信息。
       2. 通知参加活动的学者举办学术活动的时间、地点、哪些学者将出席该活动等信息。
3. 系统管理员：拥有游客浏览的近期将举办学术活动模块的权限，如置顶某个学术活动的权限。
4. 统计分析：
5. 系统管理员：拥有对所有举办的学术活动进行管理的权限。如统计学术活动举办的次数、参加人数、研究成果。可以按学院统计、按学期统计。

# 系统关键技术

## MySQL

MySQL是一种开源的小型关系数据库管理系统，适用于用户量不是很大的管理系统（PHP和MySQL在网站开发中的优势，刘建宏）。关系数据库是根据数据之间的关系，将数据进行拆分，保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加存储、查询的速度、灵活性。

## 开发工具

在目前的Java IDE市场已经形成三足鼎力:Eclipse、IntelliJ IDEA和NetBeans,对于国内很多开发人员来说, Eclipse众所周知, NetBeans之,IntelliJ IDEA确很少被人了解。IDEA不被人了解的主要原因是其开发公司不大,而且中文文档比较少,宣传也很少，IntelliJ IDEA号称最智能的Java开发工具（IntelliJ IDEA——开发人员利器，陈立兵）。IDEA不仅能进行代码提示、补全，还支持各类版本工具，例如：git、svn等版本工具，尤其在解决maven的jar包冲突方面更具优势，可以生成视图，方便快捷的找出的jar包。

## Java编程语言

如今网站后台的编程编程语很多，如Java、python、php等。但是Java支持其客户在多个平台 (设备) 中编写代码。更重要的是, Java有一个著名的口号, 叫做WORE (Write Once Run Everywhere) 的可移植编程语言（Java语言的跨平台优势，王理华.）。Java的可移植性，意味着改变代码的应用环境时，可以不用重写代码，只要Java代码编写完成，进行编译，该部分代码就可以到不同计算机中运行，只要计算机的环境中存在JDK环境变量。例如在windows环境中开发系统，编写代码，当部署到Linux环境中时，将大大减少工作量，只需在Linux中安装JDK，配置相关环境变量，在执行相关命令，该系统代码就能够运行起来。

## VUE框架

VUE框架基于MVVM架构模式，是当前非常流行、热门的一个前端框架。VUE是一个轻量，便捷的框架，拥有强大的渲染能力，还便于第三方库或现有的项目进行整合。VUE框架上手简单, 对于新手来说不需要太多的时间就可以学会。对于DOM的更新, Vue使用了异步批量处理方式, 所以会非常快速, 同时它还可以将每个组件进行一系列组合, 这样对于程序的解耦性和重用性都得到了大大提高（基于VueJs的WEB前端开发研究，徐頔）

## Spring boot框架

Spring boot框架，是由Pivotal团队创建的一个应用于JavaEE领域的一个全新的spring框架（SpringBoot框架在web应用开发中的探讨，吕宇琛） 。Spring boot的出现为了解决Spring日渐繁重的xml配置，以及复杂的依赖管理。Spring boot框架简化了JavaEE项目开发的环境搭建、开发、配置、调试、部署等工作，使开发人员不再需要编写xml文件。除此之外，spring boot内置许多流行框架与容器，如tomcat、Jetty、日志框架、JMS框架、缓存框架等。Spring boot为开发人员提供一中更加简洁、方便的方式进行开发，减少开发时间，提高开发效率，摆脱xml配置，使代码更加简洁，美化代码，给开发人员更加好的体验。

## Spring、SpringMVC、Mybaits框架

Spring以MVC结构模式，提供了一个非常灵活的SSM框架，集成当前流行的Web 框架技术, 具有良好的实现和扩展方式,。它与其他web 框架技术相比较, Spring框架技术在依赖注入、AOP方面也更加优秀（expert one-on-one J2EE Development without EJB，詹森·赫鲁）。

SpringMVC框架是基于servlet机制原理的，在实际的工作中，是拦截浏览器的发送的请求，进行解析，转发到对应的controller类，进行相应的逻辑、数据处理，并将处理结果返回浏览器，进行显示。

Mybatis框架负责系统中数据的访问获取，以及数据的持久化。数据的访问可通过xml或者注解的方式进行配置，将sql语句返回对象与Java对象映射，转化为所需的Java对象。

Spring框架，是web开发中的一个重要的轻量级框架，替代早期复杂而庞大EJB组件, 实现模型层的轻量级化，IOC于AOP是Spring框架的核心。Spring框架主要通过这两个核心功能实现MVC架构模式的应用系统各个模块间的解耦。（Spring框架在模型层的应用及原理，温立辉）。Spring能够整与表现层、持久层的框架进行整合，实现与不同框架的无缝整合，为软件应用开发提供了更大的便捷与可能性。

## UML建模语言

UML，全称Unified Modeling Language，统一建模语言，是一种规范定义、文档化、或者可视化的最标准的建模语言。UML存在于整个软件开发周期，在软件对应的开发阶段都有对应的UML模型、制品、视图，UML能让软件开发过程更好的展示，更好的理解系统的结构跟需求。UML建模与平台无关、与具体的实现的编程语言也无关，适用于多个语言与平台。总的来说，UML建模可以提高软件工程的开发效率，使开发人员对项目需求更加清晰，提高软件工程的质量。

## B/S结构

B/S结构是浏览器、服务器的组合结构，通过B/S结构，用户不用再下载软件或添加其他配置，手机或电脑等终端只要拥有浏览器就可以访问，实现与系统的交互，并且B/S结构基本上支持所有的浏览器跟系统环境，不会对系统跟浏览器有特定的要求。对用户来说，终端系统的性能跟配置要求不高，用户可以有更多的选择，用户可以根据自己的喜好、要求来进行选择。

B/S结构维护、升级的方式简单，因为只需要管理、维护系统所部署的服务器，用户是通过浏览器进行使用，没有额外的客户端，不需要进行客户端的维护，不用考虑到客户端方面的资金投入。

# 系统需求分析

## 管理系统功能需求

系统管理员、管理员可通过该模块对管理系统模块的子模块进行操作，整个系统的管理用例，如图1管理系统用例图



图1系统信息管理用例图

## 学术活动专栏功能需求

### 学术活动专栏用例图

游客可通过模块进行学术活动相关信息的浏览，可获取近期将举办的学术活动、已举办的学术活动、已获得的学术成果，并且可以通过搜索功能，进行模糊查询，输入学术活动名称或学术成果名称，获取相关文章。活动学术专栏的用例图如图3-2所示。



图2活动学术专栏用例图

### 学术活动用例场景描述

|  |  |
| --- | --- |
| 用例名称 | 搜索学术活动 |
| 主要参与者 | 游客、管理员 |
| 用例描述 | 游客可通过输入学术活动的名称，查询学术活动的信息。 |
| 前置条件 | 1. 进入学术活动专栏界面 2. 选择正确的搜索选项 |
| 基本流程 | 1. 游客、管理员在进行搜索前，在搜索选项处，选择学术活动选项。 2. 游客、管理员输入正确的学术活动名称，该学术活动名称可以不是学术活动全称，可以只是学术活动名称的一部分，游客、管理员点击搜索按钮。 3. 系统接收搜索内容，进行查询，并将查询出来的结果，以列表的方式在浏览器中显示。 |
| 替代流程 | 若系统中不存在名称的学术活动，在浏览器中会出现暂无数据的文字。 |
| 后置条件 | 无 |

## 学术活动管理功能需求

### 学术活动管理用例图

系统管理员、普通管理员可以通过该模块对未举办学术活动子模块、已举办学术活动子模块进行管理。

未举办活动子模块包括：新增、删除学术活动，对学术活动的信息进行编辑修改，如修改举办的时间、地点等信息。此外还有根据学术活动的名称进行模糊查询的搜索功能。

已举办学术活动子模块包括：对已举办活动进行活动总结，可上传活动举办过程中的图片，以及文字说明。此外还有根据学术活动的名称进行模糊查询的搜索功能。

学术活动管理用例图如图3所示



图3学术活动管理用例图

### 学术活动管理用例场景描述

|  |  |
| --- | --- |
| 用例名称 | 更新学术活动信息 |
| 主要参与者 | 管理员、系统管理员 |
| 用例描述 | 管理员选择某个学术活动，编辑其信息，更新学术活动信息的操作。 |
| 前置条件 | 1. 登录学术活动管理系统 2. 拥有学术活动管理的权限 |
| 基本流程 | 1. 系统每页显示10条未举办的学术活动，并显示学术活动举办的时间，地点、举办人等信息。 2. 管理员在未学术活动列表中，选择想要更新的学术活动，点击编辑按钮。 3. 界面将该学术活动的信息，显示在一个弹窗之中。 4. 管理员修改学术活动的信息，点击确认按钮。 5. 系统接收修改后的学术活动信息，进行更新保存。 6. 更新学术活动成功，界面出现“更新成功”文字，并刷新界面数据。 |
| 替代流程 | 1. 管理员选择错学术活动 2. 点击取消按钮关闭窗口 3. 点击空白处，关闭窗口 4. 管理员上传图片后取消编辑 5. 删除该编辑过程中上传的图片 |
| 后置条件 | 更新数据库中的学术活动信息，刷新界面数据 |

## 用户管理功能需求

### 用户管理用例图

系统管理员可通过该模块对普通管理进行管理，添加、删除普通管理员，对普通管理员的基础信息进行编辑，更新等。用户管理用例图如图4所示。



图4用户管理用例

### 用户管理用例场景描述

|  |  |
| --- | --- |
| 用例名称 | 更新用户信息 |
| 主要参与者 | 系统管理员 |
| 用例描述 | 系统管理员选择某个普通管理员，编辑其信息，更新普通管理员用户信息的操作。 |
| 前置条件 | 登录学术活动管理系统  拥有用户管理的权限 |
| 基本流程 | * + - 1. 系统每页显示10条普通管理员信息，并显示普通管理员名称、邮箱、权限等信息       2. 管理员在普通管理员列表中，选择想要更新的普通管理员，点击编辑按钮。       3. 界面将该普通管理员的信息，显示在一个弹窗之中。       4. 系统管理员修改普通管理员的信息，点击确认按钮。       5. 系统接收修改后的用户信息，并进行更新保存。       6. 更新普通管理员信息成功，界面出现“更新成功”文字，并刷新界面数据。 |
| 替代流程 | 管理员选择错学术活动   1. 点击取消按钮关闭窗口 2. 点击空白处，关闭窗口 |
| 后置条件 | 更新数据库中的学术活动信息，刷新界面数据 |

## 数据分析功能需求

### 数据分析用例图

系统管理员通过该模块可以对一定时间内，一个学院或多个学院的学术活动举办的数量、参加学术活动的人数、获得学术成果的数量进行统计。

普通管理员通过该模块可以对一定时间内，自身所处学院的学术活动举办的s护肩、参加学术活动的人数、获得学术成果的数量进行统计。

数据分析用例图如图5所示



图5数据分析用例

### 数据分析用例场景描述

|  |  |
| --- | --- |
| 用例名称 | 数据分析 |
| 主要参与者 | 管理员、系统管理员 |
| 用例描述 | 管理员、系统管理员按照某个时间范围、学院范围进行学术活动、学术成果相关数据的分析。 |
| 前置条件 | 登录学术活动管理系统 |
| 基本流程 | 系统首页默认显示今日新增的学术活动数量、学术成果数量、参加学术活动人数，再按照分类对上述信息进行分析。  管理员输入时间范围，并选择某个学院，点击确认按钮  系统接收输入的时间范围，以及学院等信息，进行数据的获取，以及数据分析。  系统将数据分析的结果显示在界面。 |
| 替代流程 | 无 |
| 后置条件 | 刷新界面数据 |

# 系统设计

## 总体设计

### 系统总体架构设计

学术活动管理系统中后台系统采用SSM框架，Java编程语言，前端使用VUE框架进行开发，其中JSON、axios技术被应用于辅助开发，数据库使用MySQL关系型数据库。

学术活动管理系统为MVC架构体系模型，每一层都通过接口进行访问，整体的架构图，如图6所示

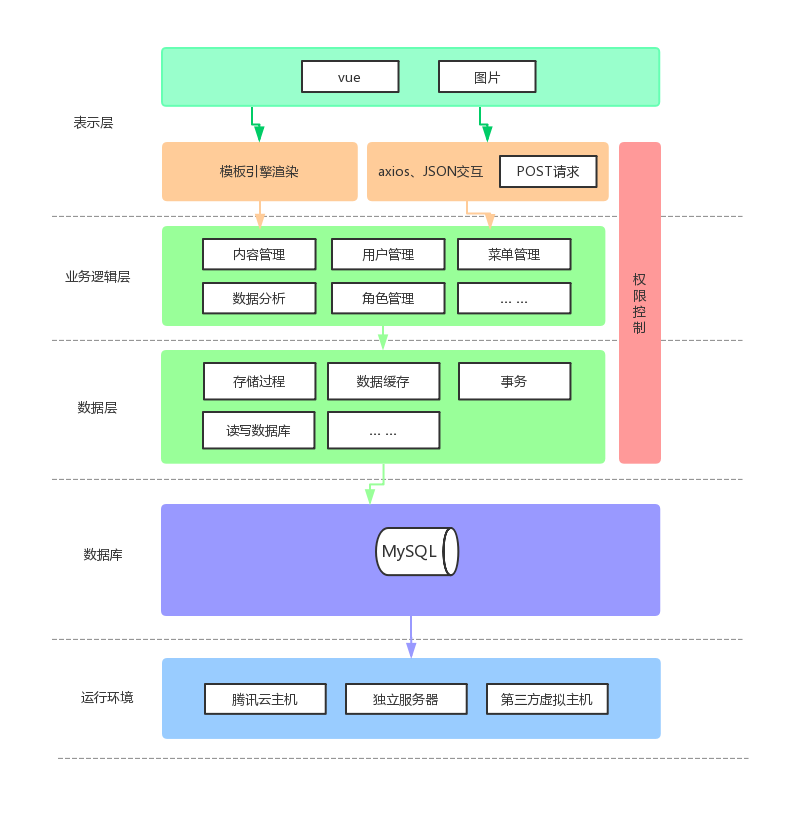


图6系统架构图

表示层，处于整个系统架构的最外层。通过VUE框架进行开发，再编译成js、css，html文件，从而为用户提供可视化界面。可接收用户输入的数据，连接业务逻辑层，接收业务逻辑层返回的数据，进行处理，重新渲染界面，展示给用户看。

业务逻辑层，处于整个系统架构的中间层。接收处理表示层的请求，进行相应的逻辑处理后，与数据层进行交互，对数据库层进行操作。在进行一定的逻辑处理后，将处理的结果返回给表示层。

数据层，对数据库进行操作为业务逻辑层提供数据服务，学术活动管理系统的相关数据都存放于这里。处于整个系统架构的最下层。

### 系统功能模块设计

学术活动管理系统，按照前端界面的需求，主要分为用户管理、角色管理、学术成果管理、学术活动管理、学科管理、学院管理、数据分析、菜单管理等，共7大功能模块。系统功能模块图如7所示。



图7学术活动管理系统功能模块图

## 数据库设计

### E-R模型

学术活动管理系统共有12个实体，实体各自的属性及实体间的关系图，如图8所示



图8 E-R模型

### 数据库表设计

#### 学术活动记录

主要用来保存学术活动的基本信息。主要字段有学术活动ID，标题、举办时间、举办地点、活动内容等信息。主键使id。学术活动信息表详细设计如表1所示。

表1学术活动信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 备注 |
| id | int | 11 | 主键 | ID |
| title | varchar | 100 | 否 | 标题 |
| author | varchar | 10 | 否 | 发布作者 |
| time | dateTime | 0 | 否 | 发布时间 |
| hold\_time | dateTime | 0 | 否 | 举办时间 |
| location | varchar | 20 | 否 | 举办地点 |
| content | varchar | 4000 | 否 | 活动内容 |
| fund | varchar | 10 | 否 | 所需资金 |
| organizer | varchar | 10 | 否 | 举办者 |
| planned\_attendance | int | 11 | 否 | 实际出席人数 |
| actual\_attendance | int | 11 | 否 | 实际出席人数 |
| ordering | int | 11 | 否 | 显示quan重 |
| type\_id | int | 11 | 外键 | 学科ID |
| read\_num | int | 11 | 否 | 阅读量 |
| toshow | tinyint | 4 | 否 | 是否对游客展示 |
| college\_id | int | 11 | 外键 | 学院 |

#### 学术活动阅读量记录

主要用来保存游客在什么时间，浏览阅读了哪个活动。主要的字段有活动学术ID，阅读活动的时间等。主键是read\_id。学术活动阅读信息表的详细设计如表2所示。

表2学术活动阅读信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 备注 |
| read\_id | int | 11 | 主键 | ID |
| id | int | 11 | 外键 | 学术活动ID |
| read\_time | datetime | 0 | 否 | 阅读时间 |

#### 学术活动总结记录

主要用来保存学术活动举办过程中的图片，活动结束的收获等文字、图片信息。主要的字段有学术活动的ID，学术活动总结的内容等，主键是outcome\_id。学术活动总结信息表如3所示。

表3学术活动总结信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 长度 | 主键/外键 | 备注 |
| outcome\_id | int | 11 | 主键 | ID |
| content | varchar | 2500 | 否 | 学术活动总结的内容 |
| id | int | 11 | 外键 | 学术活动 |

## 详细设计

### 用户管理模块

#### 添加管理员

该模块只能由拥有对应权限角色的管理员才能进行操作， 该管理员在界面，点击相应的按钮，并输入相应、正确的用户信息，就可以添加新用户。

添加普通管理员的流程图，如图9所示

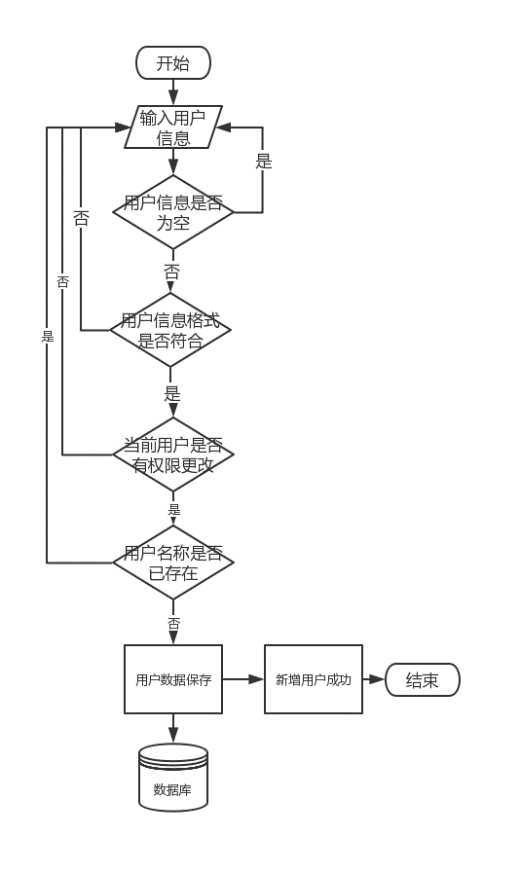


图9所示添加普通管理员流程图

管理员在用户管理模块中，点击添加按钮，在前端界面就会出现一个弹窗，管理员在界面输入用户信息，如用户名称、密码、邮箱、角色等信息，之后点击确认按钮。前端会对管理员输入的用户参数，进行一个参数非空检查，参数格式检查，如：邮箱格式是否正确，密码长度是否6-10位。若参数为空或格式不符合规定，则会在对应文本框的下方出现，红色的提示文字。只有在参数正确无误才会调用后台接口。在检查完用户信息之后，将用户信息组装成一个JSON字符串，同时在header中添加token，通过axios与post请求服务器端，在调用UserContraller.addUser方法之前，会被AuthenticationInterceptor所拦截，进行权限判断，确认当前请求用户拥有添加用户的操作权限后，才能进行下一步操作。在addUser方法中会先对JSON字符串进行解析，取出里面的参数。解析完JSON字符串后，调用IUserService. addUser方法，该方法会先进行用户名称是否重复的检查，通过调用USerMapper.findUserByUserName方法，检查用户名称是否已经存在该名称的用户，若该用户名称已被占用，则直接返回信息，前端会出现该“用户名已被占用，请重新输入用户名称”的提示。若用户名称未被占用，则会继续进行下面的流程，调用IUserService.方法进行用户的添加，保存到数据库中。若添加成功，前端后有添加成功的提示窗口，并对界面数据进行刷新。若添加失败，则会在前端有添加失败的提示。添加用户的顺序图，如图10所示。



图10添加用户顺序图

### 角色管理模块

#### 更新角色 重复校验、流程图

该模块只能由拥有对应权限角色的管理员才能进行操作，该管理员在界面，点击相应的按钮，前端就会出现一个弹窗，并输入相应、正确的角色信息，就可以更新角色信息。

更新角色信息的流程图，如图11所示

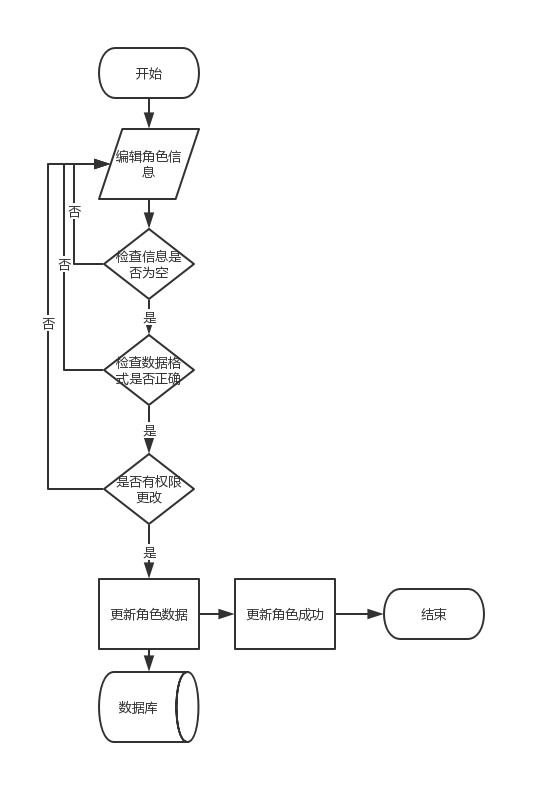


图11更新角色流程图

管理员在角色管理模块点击，浏览角色列表，或输入改角色的名称，点击修改按钮，前端界面就会出现一个弹窗，该角色的信息对应也会出现在弹窗的文本框内。管理员编辑角色的名称、权限、角色描述等信息。点击确认按钮，前端会对管理员编辑输入的信息，进行参数为非空检查，参数格式是否正确的检查。若参数为空或格式不符合规定，则会在对应文本框的下方出现，红色的提示文字。只有在角色信息确认无误之后，才会将角色信息组装成一个JSON字符串，同时，从cookie中获取token，将其添加到header中，命名为Authorization，通过axios与post方法对服务器端发出请求，服务器端接收的请求会被AuthenticationInterceptor类所拦截，该类会进行权限验证，确认当前请求用户拥有更新角色信息的操作权限后，才会进行下一步操作。在updateIdentity的方法中，会先对JSON字符串进行解析，取出其中的参数。解析完JSON字符串后，调用IIdentityService.updateIdentity方法，该方法会进行角色名称是否参数的检查，通过调用IdentityMapper.findIdentityByName方法，检查角色名称是否已经存在该名称的角色，若该角色名称已被占用，则直接返回信息，前端会出现“该角色名已被占用，请重新输入角色名称”的提示。若角色名称未被占用，则会继续进行下面的流程，调用IdentityMapper. updateIdentity方法进行角色信息的更新，保存到数据库中。若更新成功，前端后有添加成功的提示窗口，并对界面数据进行刷新。若更新失败，则会在前端有更新失败的提示。更新角色信息的顺序图，如图12所示。



图12更新角色信息顺序图

### 学术活动管理模块

#### 发布学术活动

该模块只能由拥有对应权限角色的管理员才能进行操作，该管理员在界面，点击相应的按钮，前端就会出现一个弹窗，并输入相应、正确的学术活动信息，就可以发布学术活动信息。

发布学术活动的流程图，如图13所示

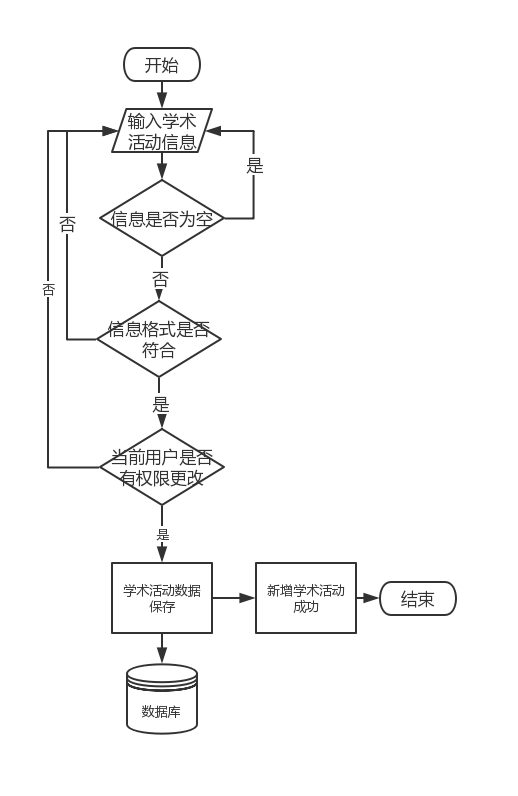


图13添加学术活动流程图

管理员在学术活动管理模块点击，发布学术活动按钮，前端界面就会出现一个弹窗，该弹窗集成了富文本编辑器，可添加图片。管理员输入学术活动的信息，如：学术活动的名称、举办的时间、举办的地方、举办者等信息。当管理员点击确认按钮，前端会对管理员输入的学术活动信息，进行参数为非空检查，参数格式是否正确的检查。若参数为空或格式不符合规定，则会在对应文本框的下方出现，红色的提示文字。只有在确认输入的学术活动信息无误之后，才会将学术活动信息组装成一个JSON字符串，同时，从cookie中获取token，将其添加到header中，命名为Authorization，通过axios与post方法对服务器端发出请求，服务器端接收的请求会被AuthenticationInterceptor类所拦截，该类会进行权限验证，确认当前请求用户拥有发布学术活动的操作权限后，才会进行下一步操作。在addEssay的方法中，会先对JSON字符串进行解析，取出其中的参数。解析完JSON字符串后，调用IEssayService.addEssay方法，该方法会将学术活动的信息保存到数据库中。保存成功之后，返回学术活动的ID，再通过循环调用IPhotService. addEssayPhoto，将学术活动文本中上传的图片保存的路径，进行保存。若保存学术活动信息与保存图片路径的操作都执行成功，前端后有添加成功的提示窗口，并对界面数据进行刷新。若更新失败，则会在前端有更新失败的提示。发布学术活动信息的顺序图，如图14所示。



图14发布学术活动顺序图

### 数据分析模块

#### 数据分析操作

该模块普通管理员与系统管理员都可以进行操作。但普通管理员只能获取到该管理员所处学院的数据。而系统管理员则可以获取到整个学校或者某个学院的数据。该管理员在界面，输入或选择某个时间范围，就可以获取该时间段内发布学术活动数量、获取的学术成果数量等数据。

数据分析的流程图，如图15所示

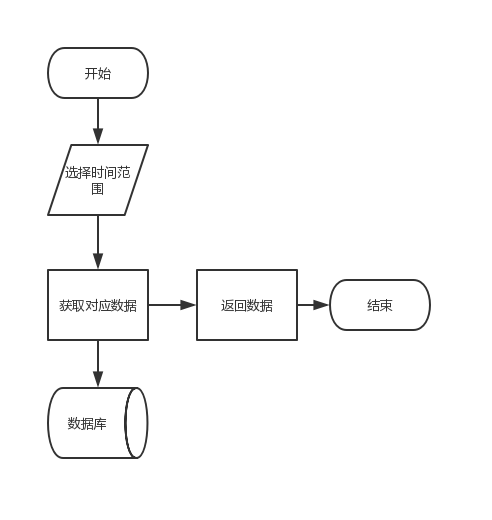


图15数据分析流程图

管理员在学术活动管理系统首页，输入某个时间范围或选择时间段选项，如：2019年2月1日至2019年3月1日，点击确认按钮，前端会该时间范围进行检查，确保前一个日期大于后一个日期。之后，才会将学术活动信息组装成一个JSON字符串，同时，从cookie中获取token，将其添加到header中，命名为Authorization，通过axios与post方法对服务器端发出请求，服务器端接收的请求会被AuthenticationInterceptor类所拦截，该类会进行权限验证，确认当前请求用户拥有发布学术活动的操作权限后，才会进行下一步操作。在getInforCardData、getEssayPieData、getAchievementPieData等方法中，会先对JSON字符串进行解析，取出其中的参数。解析完JSON字符串后，调用analysisService中的. getRangeReadNum方法获取学术活动与学术成果的总阅读量，getAddAchievementNum方法获取添加学术成果的数量，getHoldActivityNum方法获取举办学术活动的数量，getAttendanceNum方法获取参加活动人数，getEssayPieData方法获取各个类别的学术活动新增的数量、举办活动的数量、参数活动数量，getAddAchievementNumByType\_id方法获取不同类别新增学术成果的数量。当成功获取数据之后，前端界面会重新渲染，该渲染过程不是整个界面刷新，在用户没有发觉的情况下，将数据加载到前端界面。数据分析操作的顺序图，如图16所示。



图16数据分析顺序图

# 系统实现

## 学术活动专栏首页

学术活动专栏首页是提供游客浏览的 界面，游客可以查看已取得的学术成果，已举办的学术活动，近期将举办的学术活动。如图17所示。



图17学术活动专栏首页

## 学术活动/成果信息列表

学术活动、成果列表以信息的标题，发布的时间的形式，每页显示10条信息。若游客想阅读浏览其他板块，或返回首页，可通过点击导航栏中其他模块的标题，就可跳转到其他模块。此外返回首页还可通过点击页签中加粗字体的首页跳转回首页。若游客想阅读其中某篇信息的详情，可通过点击信息的标题，页面将跳转信息的详情页。如点击《立体视觉的工件快速识别定位系统》信息的标题，页面将跳转该信息的详情页，如图19所示。



图18 学术活动/成果信息列表

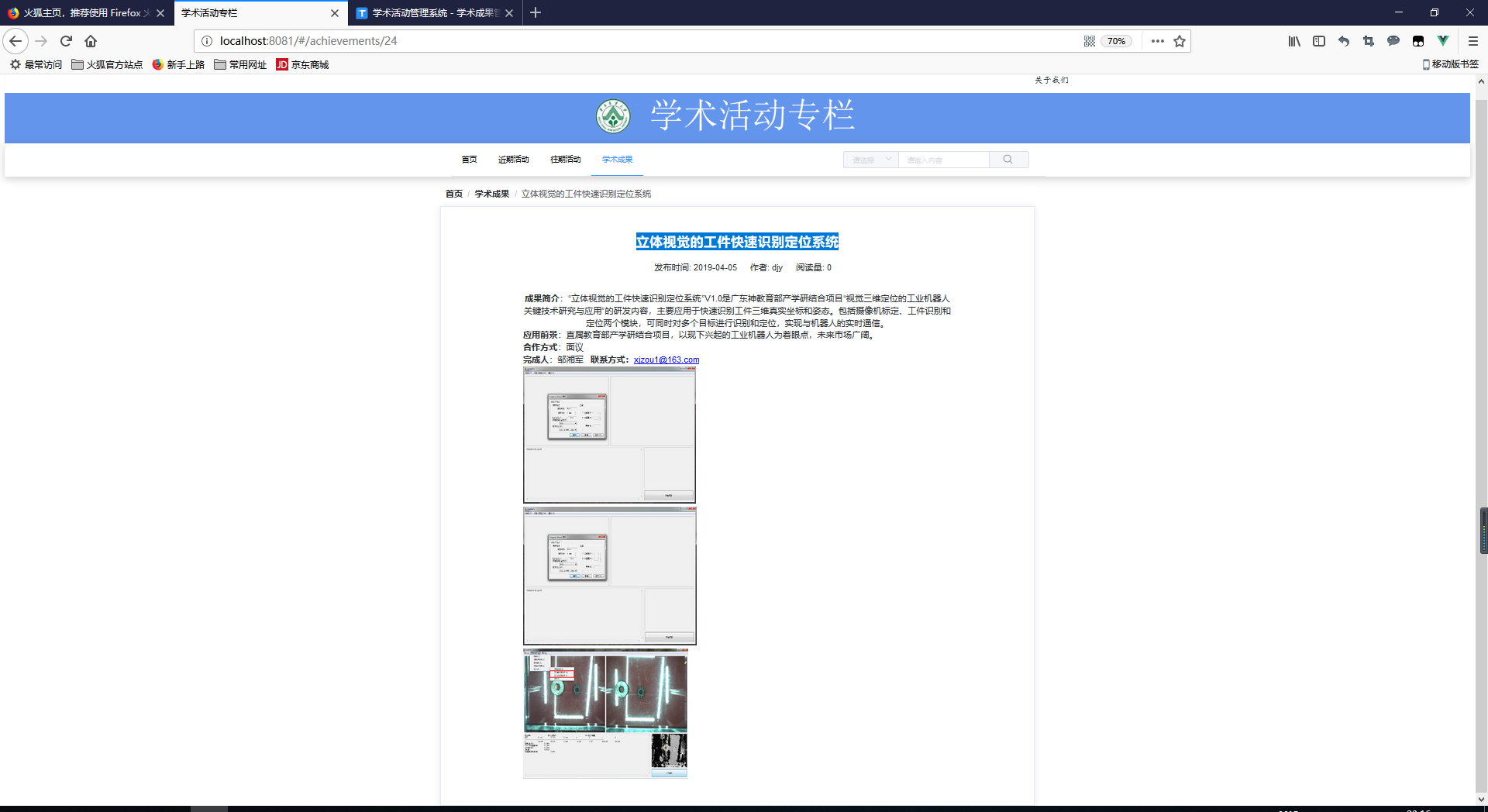


图19信息详情页

## 学术活动专栏搜索模块

搜索信息如图20所示在导航栏中的右边有搜索框，游客可通过先选择想要搜索模块的信息，再输入关键词的方法来搜索信息。如搜索学术成果中，系统相关的信息，搜索结果如图21所示。游客可看到相关的信息列表。游客点击信息的标题则进入到信息的详情页中。

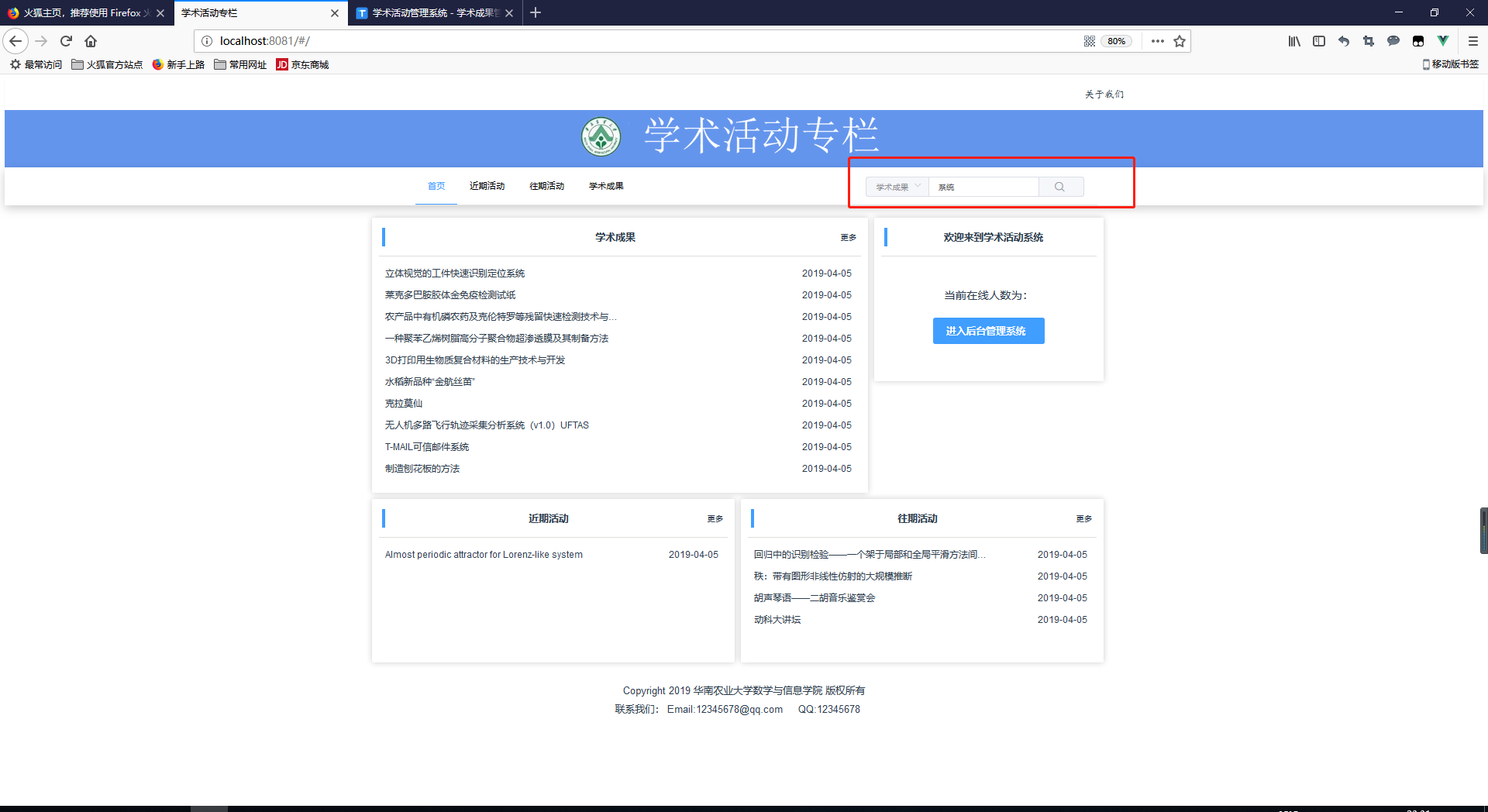


图20搜索信息

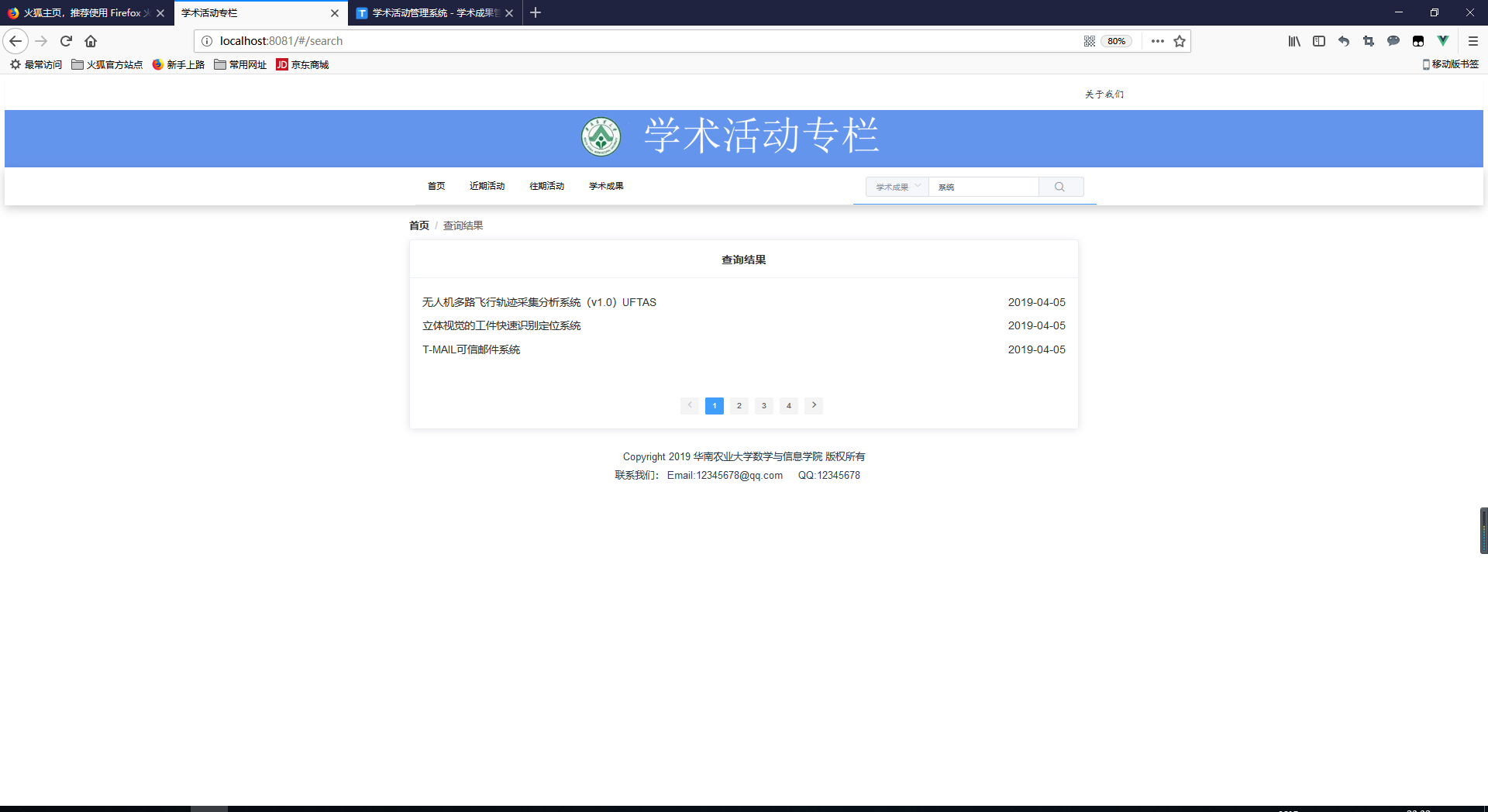


图21搜索结果

## 学术活动管理系统首页

该系统是后台管理系统，供管理人员使用，当管理人员登录该系统时，系统会根据登录用户的学院显示数据。首页展示各个模块信息的总阅读量、新增成果数量、举办活动数据、参加活动人数、饼图为学术活动、学术成果分类分析的数据。如图22 所示，图中登录的用户为系统管理员，因为系统管理员不分学院所以显示的为全部学院的数据。

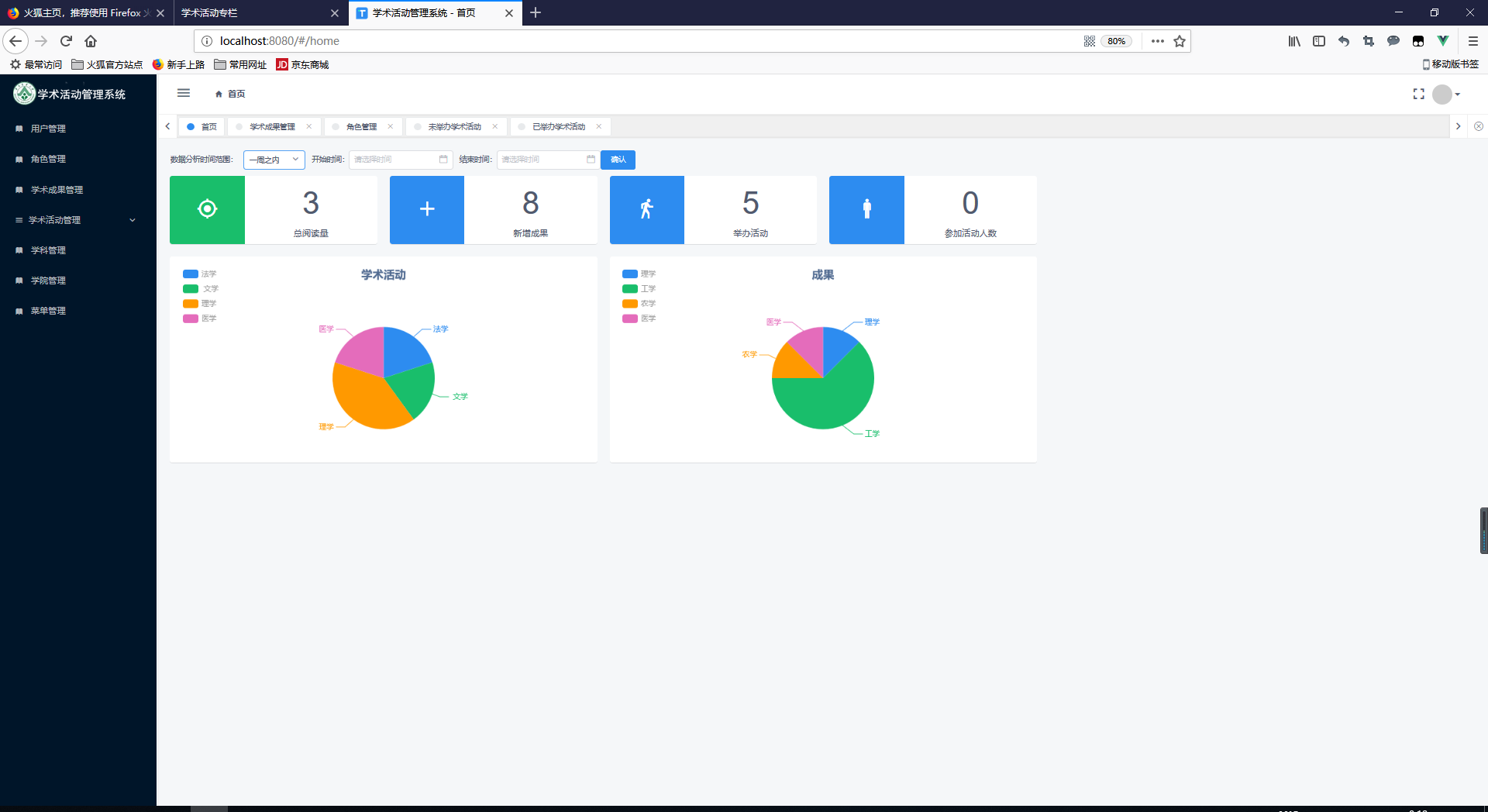


图22学术活动管理系统首页

## 学术活动管理

学术活动管理分为未举办学术活动、已举办学术活动。在未举办学术活动界面中，可以新增学术活动，如图23，输入相关学术活动点击确认之后，如新增成果，该学术活动就会保存到数据库，该新增的学术活动会在该界面中显示，如图24所示，同时游客也可通过学术活动专栏进行查看。



图23新增学术活动

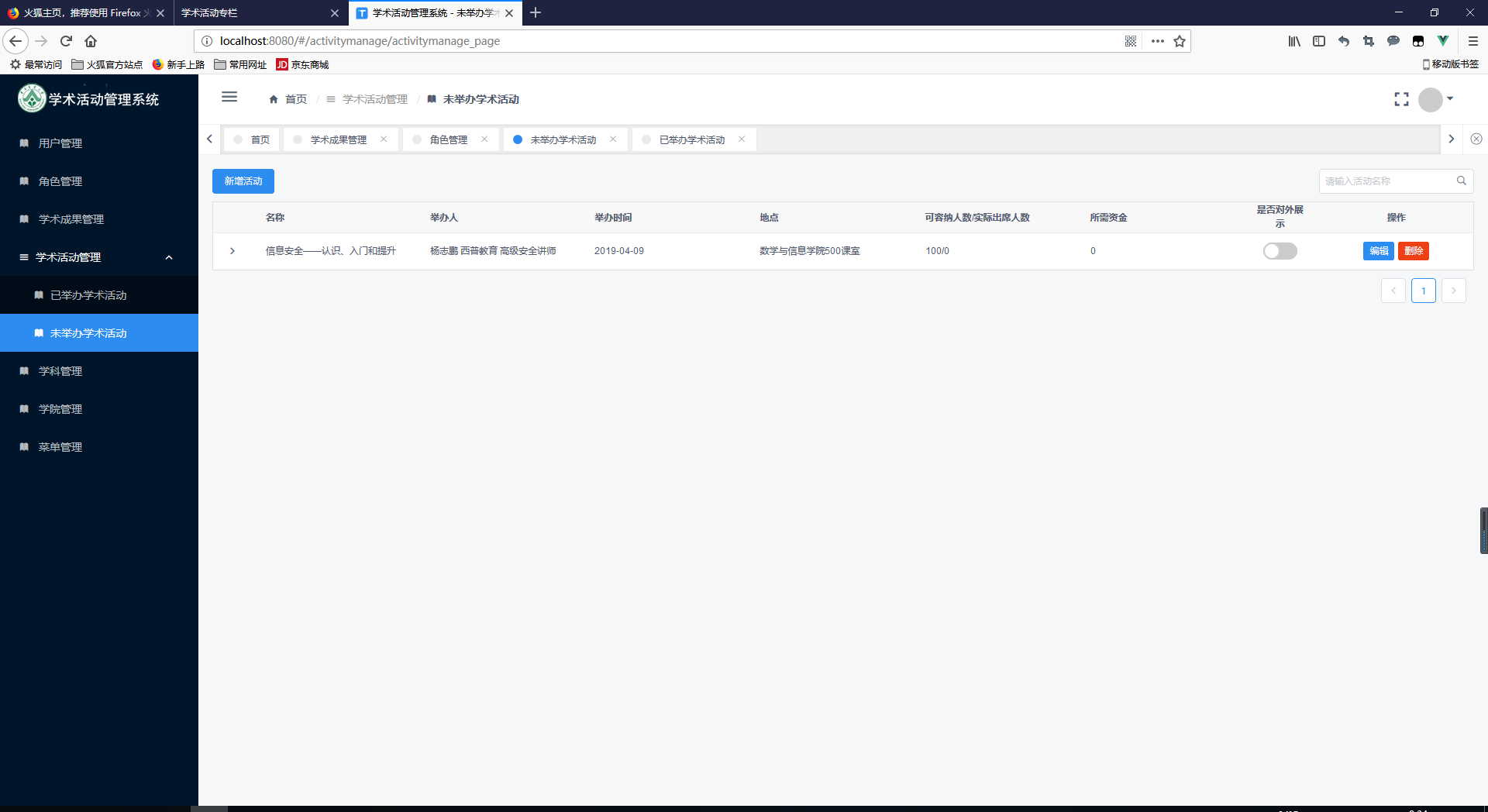


图24新增结果

# 结论与展望

# 参考文献：

PHP和MySQL在网站开发中的优势 【作者】 [刘建宏](http://ss.zhizhen.com/s?sw=author%28%E5%88%98%E5%BB%BA%E5%AE%8F%29authorcompy%28%E5%BB%B6%E8%BE%B9%E5%A4%A7%E5%AD%A6%29)， [钟嫩妹](http://ss.zhizhen.com/s?sw=author%28%E9%92%9F%E5%AB%A9%E5%A6%B9%29authorcompy%28%E5%BB%B6%E8%BE%B9%E5%A4%A7%E5%AD%A6%29)

[1]王理华.Java语言的跨平台优势[J].科技风,2017(25):64.

expert one-on-one J2EE Development without EJB 作者：詹森·赫鲁 来源：[B].电子工业出版社2005

[1]温立辉.Spring框架在模型层的应用及原理[J].福建电脑,2017,33(05):147-148.

[1]吕宇琛.SpringBoot框架在web应用开发中的探讨[J].科技创新导报,2018,15(08):168+173.