

诚信声明

我声明，所呈交的毕业论文是本人在老师指导下进行的研究工作及取得的研究成果。据我查证，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得其他教育机构的学位或证书而使用过的材料。我承诺，论文中的所有内容均真实、可信。

毕业论文作者签名：

签名日期： 年 月 日

基于 Web 的师生科研互动平台

[摘 要]

针对目前学生要参加各级创新竞赛、本科生研究项目或开放实验项目，与老师的交流不是十分畅通的现状，设计了基于 Web 的科研交流平台，解决学生盲目选题或对选题理解不深等问题，实现学生和老师的相关项目和课题的网上交流与合作以及老师在线指导学生，讨论和答疑。系统采用 Node.js 技术，以 Mongo DB 作为数据库，通过电脑或手机实现了教师以及学生发布、选择项目以及项目交流的功能，具有高并发处理性能优秀以及跨平台、跨设备的特点。通过测试，达到了比较理想的效果，该系统弥补了暨南大学在学生科研与教师交互缺乏互动平台的不足。

[关键词]

WEB; AJAX; NODE.JS; MONGO DB; JQUERY

Web-based research and communicate platform between teachers and students

Abstract:

For the current status of the students to participate in competitions at all levels of innovation, undergraduate research projects or open experiment projects with the teachers is not very clear, I have design a Web-based research and communicate platform to address the issues of students' blindly topic selections or poorly understanding of the topic, to achieve the online communication and collaboration between students and teachers on related projects or topics. The system is based on Node.js with Mongo DB database, teachers and students can publish or select projects, communicate with each other using desktop computers or mobile devices, has the characteristics of high concurrency handling and platform/devices across. The system has acquired ideal results through tests, and recovers the lack of scientific research and communication platform between students and teachers of JNU.

Keywords:

WEB; AJAX; NODE.JS; MONGO DB; JQUERY

目 录

1 绪论	1
1.1 选题背景	1
1.2 研究框架	1
1.2.1 研究目标	1
1.2.2 研究内容	2
1.2.3 创新之处	2
1.2.4 研究方法	2
1.3 术语说明	3
1.3.1 NODE.JS	3
1.3.2 MONGO DB	3
1.3.3 AJAX	3
1.3.4 JQUERY	4
1.3.5 BOOTSTRAP	4
1.4 设计环境	4
2 设计方案	5
2.1 技术路线	5
2.1.1 服务端技术	5
2.1.2 数据库技术	6
2.1.3 前端技术	6
2.2 功能模块设计	7
2.2.1 用户模块	8

2.2.2 项目模块	8
2.2.3 团队模块	8
2.2.4 评论模块	8
2.3 数据库设计	9
2.4 架构设计	12
3 路由设计	14
3.1 EXPRESS 简介	14
3.2 路由入口	14
3.3 具体路由示例	18
3.3.1 页面 GET 请求	18
3.3.2 页面 POST 请求	19
4 重点模块设计	22
4.1 用户模块	22
4.1.1 注册控制	22
4.1.2 登录控制	23
4.1.3 功能控制	23
4.2 项目模块	25
4.2.1 项目权限	25
4.2.2 项目加入	26
4.2.3 项目更新	26
4.2.4 项目分享	26
4.3 评论模块	26
4.3.1 直接评论	27
4.3.2 回复评论	27

5 系统测试	28
5.1 PC 浏览器端	28
5.2 移动设备浏览器端	32
结论	35
致谢	36
参考文献	37

1 绪论

1.1 选题背景

如今，Web 应用已十分普遍，我们每天都在使用各种各样的 Web 服务，特别是进入云计算时代后，HTML5 的快速发展给 Web 应用带来了更广阔的前景，目前已出现基于 Web 技术实现的桌面应用，甚至操作系统等。同时，Web 应用的方便性也是其最大的特点之一，用户只需要使用浏览器，就可以访问几乎所有 Web 应用并使用其服务。

针对目前学生要参加各级创新竞赛、本科生研究项目或开放实验项目，与老师的交流不是十分畅通的现状，设计了基于 Web 的科研交流平台，方便学生了解、参与科研项目，并且方便学生与教师在整个项目过程中的交流，从而弥补暨南大学在学生科研与教师交互缺乏互动平台的不足。

1.2 研究框架

1.2.1 研究目标

研制开发本平台的目的，主要是方便学生参加各级创新竞赛、本科生研究项目或开放实验项目，以及方便学生与教师在整个项目过程中的交流，克服学生盲目选题或对选题理解不深问题。

本设计将建立基于 Web 的师生科研信息交流和合作的互动平台，通过该平台，教师和学生通过电脑或手机，发布相关科研信息并实现互动。包括教师寻找学生参与的课题、各级创新项目的互动、挑战杯项目、开放实验项目以及毕业设计的互动等，实现学生和老师的相关项目和课题的网上交

流与合作以及老师在线指导学生，讨论和答疑。

1.2.2 研究内容

包括 Web 平台整体架构的设计，服务端技术的选用及设计，数据库技术的选用及设计，前端技术的选用及设计，以及对 Web 平台安全、性能、伸缩、易用等方面的研究实现等。实现基本的以及扩展的功能，同时实现 PC 端与手机端平台的自适应用户界面。

1.2.3 创新之处

响应式设计已成为 Web 界的热门话题，考虑到用户群的特点，平台除提供通过常规计算机浏览器使用的方式以外，还将同时为移动设备（如移动电话、平板电脑等）设计界面，充分利用 HTML5 以及 CSS3 的高级特性，在保证其在各大浏览器下表现一致的前提下，使其成为一个跨设备的应用。同时考虑到我国实际情况，将尽量保持平台对较低版本浏览器的兼容性。

在服务端选择了较新的 Node.js 技术以及非关系型数据库来进行开发，同时对平台的安全方面做充分的考虑，将是一项充满挑战的任务。

1.2.4 研究方法

对已有技术主要通过阅读开发文档以及开发案例等方法进行学习及使用，在完成基础功能的同时结合课堂学习的软件工程知识，将软件工程思想应用到平台的每一个角落。

1.3 术语说明

1.3.1 Node.js

Node.js 是一个基于 Chrome V8 引擎的 JavaScript 执行平台，它可以快速构建网络服务及相关应用。借助事件驱动，异步非阻塞 I / O 等特性，对于数据密集型，I / O 密集型的应用场景 有着优秀的处理能力。^[1]

1.3.2 Mongo DB

Mongo DB 是一个开源免费的基于分布式文件存储的数据库，旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。Mongo DB 是一个介于关系数据库和非关系数据库之间的产品，他支持的数据结构非常松散，是类似 JSON 的 BSON 格式，因此可以存储比较复杂的数据类型。Mongo 最大的特点是他支持的查询语言非常强大，其语法类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。^[2]

1.3.3 AJAX

AJAX（Asynchronous JavaScript And XML）是指一种创建交互式网页应用的网页开发技术。通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。AJAX 可以消除传统 HTTP 通信模式中对表单的依赖，从而实现快速而轻量的异步通信。^[3]

1.3.4 jQuery

jQuery 是一个开源免费的兼容多浏览器的 JavaScript 库，也是如今是最流行的 JavaScript 库，在世界前 10000 个访问最多的网站中，有超过 55% 在使用 jQuery。jQuery 的语法设计可以使开发者更加便捷。

1.3.5 Bootstrap

Bootstrap 是目前最受欢迎的开源免费前端框架。它由 Twitter 的设计师 Mark Otto 和 Jacob Thornton 合作开发，是一个 CSS/HTML 框架。

1.4 设计环境

见表 1-1：

表 1-1 软件环境

操作系统	Windows 7 professional x64 SP1
数据库	Mongo DB v3.0.2
运行环境	Node.js v0.10.36
开发工具	Brackets Release 1.1 build 1.1.0-15558
调试工具	Chrome @ 42.0.2311.90 m Developer Tools

2 设计方案

本章节主要对平台的技术选用、功能模块设计、数据库设计、架构设计等方面进行分析与论述，是系统成败的关键所在。

2.1 技术路线

2.1.1 服务端技术

Web2.0 时代，性能是永恒的主题。举例说，我们暨南大学的综合教务管理系统，功能方面非常完备，能够很好地完成各种各样的教学日常任务，是非常优秀的 Web1.0 时代产物。但是到了今天，其短板也暴露无遗，最主要的体现就是每到全校选课时段，系统就会因为无法承受高并发带来的压力而效率大减，甚至崩溃，使着急选课的同学感到非常无奈。在互联网愈来愈普及的今天，如何处理高并发带来的服务器压力是一项炙手可热的话题，其方法主要有两个：一是提高硬件基础；二则是从软件方面着手。在硬件条件有限的情况下，软件技术的选择和使用就成为了非常重要的环节。

因此，在选择 Web 平台的服务端技术时，性能、效率、成本、发展等因素都需要经过缜密的考虑与求证，最终得到适合于平台特点与需求的答案。在对比研究 Java, Microsoft .NET, PHP, Node.js 等技术后，最终选择了 Node.js 作为平台的服务端开发技术。

Node.js 是一个服务器端的 JavaScript 环境。Node 用谷歌公司开发的 V8 引擎运行 JavaScript。借助于 V8，Node 得以提供一个服务器端的运行环境来快速的编译和执行 JavaScript 语言。它速度的提升主要是因为 V8 把

JavaScript 编译为本地机器代码（直接解释执行），而不是编译为字节码或中间代码（再解释执行）。^[4]Node.js 是开源的，跨平台的，能够运行在各种操作系统上。Node.js 的核心是事件循环机制。为了处理超大规模的并发请求，Node.js 中所有的 I/O 集中操作都是异步的、非阻塞的。作为一种较新的技术，目前网络上有很多针对 Node 与其它服务端技术在 IO 与 REST 性能方面的对比，结论是其远超如 Java，PHP 等的传统技术。这意味着在相同的硬件条件下，使用 Node 的 Web 应用可以处理更多的客户端请求。

2.1.2 数据库技术

随着 Web2.0 的兴起，传统的关系数据库在应付超大规模和高并发的 Web2.0 纯动态网站已经显得力不从心，暴露了很多难以克服的问题，而非关系型的数据库则由于其本身的特点得到了非常迅速的发展。NoSQL（Not Only SQL）数据库的产生就是为了解决大规模数据集合多重数据种类带来的挑战，尤其是大数据应用难题。^[5]

结合 Node 的特点，决定选用开源、免费的文档型 NoSQL 数据库 Mongo DB 作为平台数据库，这样得以搭建一个全栈的 JavaScript Web 应用框架（Mongo DB 使用基于 JavaScript 的 Shell 进行操控）。

2.1.3 前端技术

在 Web 前端选用较常规的 jQuery 库作为 JavaScript 支持，同时选用 Twitter Bootstrap 作为 HTML/CSS 基本架构，搭配其它开源支持项目进行开发。本平台在 Web 前端的主要特点是充分利用了 HTML5 与 CSS3 的新特

性，除提供通过常规计算机浏览器使用的方式以外，还将同时为移动设备（如移动电话、平板电脑等）设计界面，使其成为一个跨平台、跨设备的云应用。

在 Web 前端与服务端的交互方面，全部采用 AJAX 的方式进行数据交互。这样做的好处是将页面展示与后台逻辑完全解耦，在减轻服务器负担的同时，也更加符合软件工程理念的基本思想，以及便于扩展开发和维护。

2.2 功能模块设计

在学习项目任务书以及跟指导教师讨论以后，确定了以下的一些主要功能模块，见图 2-1。

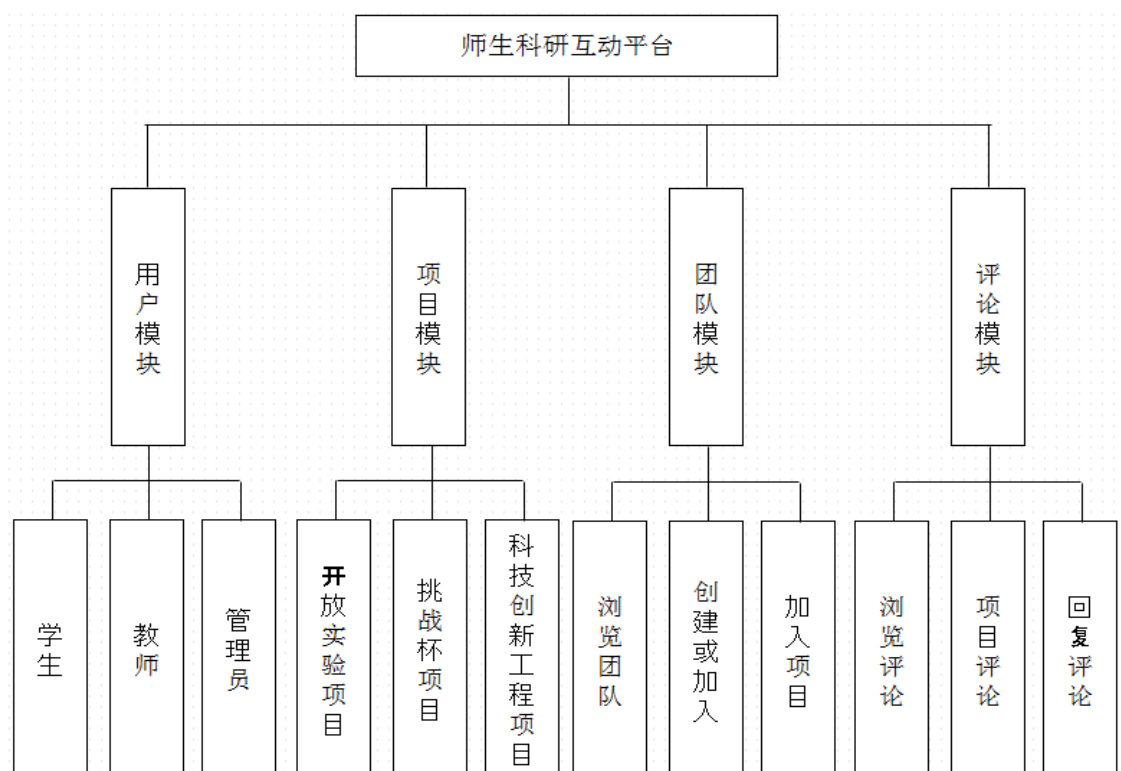


图 2-1 功能模块

2.2.1 用户模块

主要包括用户的注册控制、登录控制以及管理控制等。同时，学生和教师角色可以在个人中心查看自己拥有或申请加入的项目、团队，或查看其他人对自己的回复评论，管理自己的个人信息（姓名、班级等）以及密码。管理员角色可以对平台的所有信息进行管理，拥有最高权限，使用独立的登录以及注册模块。

2.2.2 项目模块

平台的初步设计实现阶段主要涉及三种项目：开放实验项目、挑战杯项目以及科技创新工程项目。其中，开放实验项目只允许由教师开设，学生可以独立选课。挑战杯项目、科技创新工程项目允许由教师开设或学生申请并寻找指导教师，学生必须组成团队后方可由团队负责人进行选课。

2.2.3 团队模块

包括团队的浏览、寻找以及创建、加入、选课等功能。团队只能由学生创建，创建者为负责人，其它学生以及教师可以申请加入。挑战杯项目、科技创新工程项目需要以团队为单位选课，选课团队必须有指导教师。

2.2.4 评论模块

用户可以对各种项目以评论、回复等方式进行交流，未登录用户可以查看评论，登录后可以发表评论。回复目标将可在其个人中心查看到对其的所有回复评论。

2.3 数据库设计

由于使用的是 NoSQL 数据库，在数据库的设计上会与关系数据库有一些不同之处，其一是数据表的主键将全部由 Mongo DB 在数据插入时自动生成的 Object ID 担任。但为了保持平台数据的一致性与完整性，将沿袭关系数据库的一些设计方法（外键等）。

系统共设计了 6 个数据表，如下表所示。

表 2-1 用户表

键名	类型	属性	备注
Id	String	非空，唯一	用户的学号或工号，作为登录名
Password	String	非空	用户密码，以 MD5 码储存
Key	String	非空	随机生成的密钥，以 MD5 码储存
Name	String	非空	用户姓名
Sex	String	枚举	用户性别：男、女或其它
Type	String	枚举	用户类型：同学、老师或管理员
Active	Boolean	非空，默认为 false	未激活用户不能登录，用于注册审核及毕业后失效
Email	String		用户邮箱地址
Phone	String		用户电话号码
Student Attr	Object		此对象用于储存“同学”类用户的特有属性，包括学院、年级、专业等

Teacher Attr	Object		此对象用于储存“老师”类用户的特有属性，包括部门、职称等
--------------	--------	--	------------------------------

表 2-2 项目表

键名	类型	属性	备注
Name	String	非空	项目名
Type	String	非空，枚举	项目类型：开放实验项目、挑战杯项目或科技创新工程项目
Description	String	非空	项目简介
College	String	非空	项目所属学院
Date Start	Date	非空	项目预计开始时间
Date End	Date	非空	项目预计结束时间
Date Update	Date	非空	项目最后更新时间
Teacher	Object Id	外键	项目指导教师
Creator	Object Id	外键	项目创建者
Open Experiment Attr	Object		此对象用于储存“开放实验项目”的特有属性，包括容量、学时、实验室等
Challenge Cup Attr	Object		此对象用于储存“挑战杯项目”的特有属性，包括类型、基础、目标等
Innovation Project Attr	Object		此对象用于储存“科技创新工程项目”的特有属性，包括摘要、关键词等

表 2-3 团队表

键名	类型	属性	备注
Name	String	非空	团队名
Leader	Object Id	外键	团队负责人
Description	String	非空	团队描述
Date Create	Date	非空，默认为创建的时间	团队创建时间

表 2-4 选课表

键名	类型	属性	备注
Student	Object Id	外键	选课学生
Team	Object Id	外键	选课团队
Project	Object Id	外键	所选项目
Active	Boolean	非空，默认为 false	选课激活状态（是否通过审核）
Date	Date	非空，默认为创建的时间	选课时间

表 2-5 入团表

键名	类型	属性	备注
User	Object Id	外键	申请学生
Team	Object Id	外键	申请团队
Active	Boolean	非空，默认为 false	申请激活状态（是否通过审核）
Date	Date	非空，默认为创建的时间	申请时间

表 2-6 评论表

键名	类型	属性	备注
From	Object Id	非空，外键	评论用户
To	Object Id	外键	评论回复对象
Project	Object Id	外键	评论项目
Body	String	非空	评论内容
Date	Date	非空，默认为创建的时间	评论时间

2.4 架构设计

架构设计是软件开发的重要环节，本平台由于使用的服务端技术需要实现 HTTP 服务器，所以客户端请求的路由设计成为了架构设计的重点。整体设计见下图。



图 2-2 架构设计

3 路由设计

路由设计是为所有客户端的请求路径设计处理方法以及响应逻辑，包括接收请求、处理请求、发送响应三个阶段。由于网站需要实现 HTTP 服务器的客户端请求过程以及处理客户端 AJAX 请求，所以路由设计非常重要，直接影响到整个平台的结构以及性能。本章节主要介绍平台的路由系统设计。

3.1 Express 简介

Express 是一个简洁、灵活的 Node.js Web 应用开发框架，对 Node 的 HTTP 模块进行了高度封装。^[6] 它提供一系列强大的特性、丰富的 HTTP 快捷方法和任意排列组合的 Connect 中间件，能够达到相当于 Node 原生应用的性能。使用 Express 框架主要的好处是快速开发和易于维护。

3.2 路由入口

```
// app.js

var express = require('express'),

    session = require('express-session'),

    mongoose = require('mongoose'),

    path = require('path'),

    favicon = require('serve-favicon'),

    logger = require('morgan'),

    cookieParser = require('cookie-parser'),

    bodyParser = require('body-parser'),

    moment = require('moment'),
```

```
    app = express();
mongoose.connect('mongodb://localhost/test');
moment.locale('zh-cn');
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');
app.use(favicon(__dirname + '/public/favicon.ico'));
app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({
    extended: true,
    limit: 100000
}));
app.use(session({
    secret: '666',
    resave: false,
    saveUninitialized: true
}));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));
app.use('/', require('./routes/index'));
app.use('/api/get', require('./routes/api/get'));
app.use('/api/post', require('./routes/api/post'));
app.use(function (req, res, next) {
    var err = new Error('Not Found');
```

```
    err.status = 404;

    next(err);
  });

  if (app.get('env') === 'development') {
    app.use(function (err, req, res, next) {
      res.status(err.status || 500);

      if (err.status === 404) {
        res.render('404', {
          user: req.session.user
        });
      } else {
        res.render('error', {
          message: err.message,
          error: err
        });
      }
    });
  }

  module.exports = app;
```

代码说明：该文件代码是单个客户端请求进入服务器后的整个处理及路由过程，即平台程序的直接入口。每个客户端请求都会经历路由的整个过程，最后由平台做出相应的响应。

以下是对主要代码的详细说明。

```
var express = require('express'),
```

```
...
```

```
app = express();
```

这段代码的作用是加载依赖的 Node 模块，并启动 Express 应用。

```
mongoose.connect('mongodb://localhost/test');
```

使用 Mongoose 连接本地数据库。

```
app.set('views', path.join(__dirname, 'views'));
```

```
app.set('view engine', 'ejs');
```

设置表现层页面的路径，并设置页面引擎为 EJS 渲染引擎。

```
app.use(bodyParser.json());
```

```
app.use(bodyParser.urlencoded({
```

```
  extended: true,
```

```
  limit: 100000
```

```
}));
```

```
app.use(session({
```

```
...
```

```
}));
```

将请求的 Post 数据（如果存在）转换为 JSON 格式，并限制大小为 100KB。

设置 Session。

```
app.use('/', require('./routes/index'));
```

```
app.use('/api/get', require('./routes/api/get'));
```

```
app.use('/api/post', require('./routes/api/post'));  
app.use(function (req, res, next) {  
    var err = new Error('Not Found');  
    err.status = 404;  
    next(err);  
});
```

路由主体，按顺序监听了三个路径，分别是 '/'、'/api/get' 与 '/api/post'，其请求具体处理逻辑包含在相应的文件中。若请求路径不在其中，则抛出 404 异常。

3.3 具体路由示例

3.3.1 页面 GET 请求

```
router.get('/center', function (req, res) {  
    if (req.session.user) {  
        if (req.session.user.type === '管理员') {  
            res.redirect('/admin');  
        } else {  
            res.render('center', {  
                user: req.session.user  
            });  
        }  
    } else {  
        res.redirect('/login');
```



```
}
})
```

这段代码接收了客户端对 <http://hostname/center> URL 的 GET 请求，并在处理前首先进行了判断：

- a) 若已登录且登录类型为管理员，则重定向到管理员页面；
- b) 若已登录且登录类型不是管理员，则渲染 `center` 页面并发送给客户端；
- c) 若未登录，则重定向到登录页面。

3.3.2 页面 POST 请求

```
router.post('/signin', function (req, res) {
  if (req.query.type === 'admin') {
    model.User.findOne({
      type: '管理员'
    }, function (err, doc) {
      ...
      if (md5(req.body.password + doc.key) === doc.password) {
        req.session.user = doc;
        res.sendStatus(200);
      } else {
        res.sendStatus(401);
      }
    }
  }
});
```

```

        ...
    });
} else {
    ...
    model.User.findOne(condition).lean().exec(function (err, doc) {
        if (err || !doc) {
            res.sendStatus(401);
        } else {
            if (md5(req.body.password + doc.key) === doc.password) {
                if (doc.active) {
                    ...
                    req.session.user = doc;
                    res.sendStatus(200);
                } else res.sendStatus(403);
            } else {
                res.sendStatus(401);
            }
        }
    })
}
})

```

这段代码处理了客户端对 <http://hostname/api/post/signin> URL 的 POST 请求

- a) 若请求类型为管理员，则从 User 表中取得管理员记录与用户输入密

码比对，匹配则设置 Session 并返回 200 OK，否则报错

- b) 若请求类型为其它，则从 User 表中根据响应 ID 获取记录后与用户输入密码比对，匹配则设置 Session 并返回 200 OK，否则报错

4 重点模块设计

4.1 用户模块

考虑到用户在大部分行为以及管理上的一致性，并没有将用户根据类别（学生、教师、管理员）分开到不同的数据表存放，而是统一存入用户表，根据表中的类型属性来区分。

4.1.1 注册控制

控制流程：

- a) 用户填写注册信息
- b) 页面使用 MD5 算法对密码加密后使用 AJAX 提交数据
- c) 后台对提交数据进行 XSS 过滤，并设置激活状态为 FALSE
- d) 后台生成随机密钥，并使用该密钥与提交的密码组合后再使用 MD5 算法加密，生成储存在数据库中的密码
- e) 后台尝试储存该用户记录，若储存成功则返回 200 OK，不成功（ID 冲突）则返回 500 错误
- f) 页面根据返回的 HTTP 状态码做出响应，告知用户注册成功，或 ID 已存在

详细代码：

```
user.active = false;
```

```
xssUser(user);  
  
user.key = md5(new Date());  
  
user.password = md5(user.password + user.key);  
  
user = new model.User(user);  
  
user.save(function (err) {  
    res.sendStatus(err ? 500 : 200);  
});
```

4.1.2 登录控制

控制流程：

- a) 用户填写登录名及密码
- b) 页面使用 MD5 算法对密码加密后使用 AJAX 提交数据
- c) 后台根据登录名获取正确密码与密钥，比对后根据匹配结果判断是否登录成功，并返回响应的 HTTP 状态码
- d) 页面根据返回的 HTTP 状态码做出响应，告知用户登录成功，或遇到了问题（密码不匹配，账号未激活等）

详细代码：请见 3.3.2 节

4.1.3 功能控制

某些功能需要用户登录后才提供使用，如发布项目、发表评论等，某些功能只对管理员开放，如各种数据增删改等，对此平台主要从两个方面对此进行控制：一是在渲染页面时根据登录情况调整渲染元素，以及在页面

脚本中加入是否登录的判断从而决定是否提交请求以及提醒用户登录（简称前端控制），二是在服务端对用户提交的请求根据 Session 情况判断是否作出响应（简称后台控制），这样做的好处是可以使平台的性能与安全性都得到了一定的保障。

详细代码：

a) 前端举例（教师或学生可以创建项目，学生不能创建开放实验项目）

```
<% if(typeof user!=='undefined' && user.type!=='管理员' && ((user.type==='老师' && projectType==='开放实验项目') || projectType==='挑战杯项目' || projectType==='科技创新工程项目')){%>
```

```
<button ...> ... 创建</button>
```

```
<% } %>
```

b) 后台举例（管理员可以获取所有用户的信息，已登录用户可以获取自己的信息，未登录用户不能获取用户信息）

```
router.get('/user', function (req, res) {
  if (req.session.user) {
    if (req.session.user.type === '管理员') {
      ...
    } else {
      if (req.query.self) {
        res.status(200).send(req.session.user)
      }
    }
  }
})
```

```
        } else {  
            req.sendStatus(404);  
        }  
    }  
} else {  
    res.sendStatus(401)  
}  
})
```

4.2 项目模块

同用户模块，没有将不同类型的项目划分到不同的表中储存，主要目的是便于管理以及扩展，若将来需要添加一种新的项目，只需要在数据库中为其增加一种类型即可，同时也无需求改增删改查方法。

为防止恶意提交，在创建一个项目时，项目所有者将被强制设定为当前 Session 的用户，且项目内容将会经过 XSS 过滤。

4.2.1 项目权限

- a) 未登录用户可以查看项目列表，查看项目详细内容以及指导教师简要信息等
- b) 已登录学生可以选课、提交报告、选择自己的团队参加项目，以及发布挑战杯项目、科技创新工程项目以寻求导师等
- c) 已登录教师可以发布所有类型的项目，并处理学生以及团队对项目的

申请

4.2.2 项目加入

学生可以申请加入开放实验项目，或者组建团队并以团队为单位申请加入挑战杯项目或科技创新工程项目。

学生提交申请后，教师可以在“个人中心”查看并处理（接受或拒绝）申请。对于挑战杯项目或科技创新工程项目，在接收一个团队后，其它团队将不能再加入该项目，并且其它团队对该项目提交的申请也会失效。

4.2.3 项目更新

项目所有者可以对项目进行内容的更新，学生发布的项目在指导教师加入后所有者即变更为该指导教师。

4.2.4 项目分享

项目详细页面添加了一系列分享按钮，提供了将页面分享到各大社交网络的接口，任何浏览者都可以参与页面分享，从而便于教师寻找学生加入项目，也便于学生寻找其它有兴趣的同学组建团队一同加入项目。该分享功能使用了“百度分享”的部分模块。

4.3 评论模块

为防止恶意提交，在提交一条评论时，评论的“来自用户”属性将被强制设定为当前 Session 的用户，且评论内容将会经过 XSS 过滤。代码如下。

```
req.body.from = req.session.user._id;
```



```
xssComment(req.body);  
var comment = new model.Comment(req.body);  
comment.save(function (err) {  
    res.sendStatus(err ? 500 : 200)  
})
```

4.3.1 直接评论

已登录用户可以在项目内发表自己对项目的评论以及想法等，是实现平台师生交流的重要方法。浏览者在打开该页后即可查看历史评论内容。

4.3.2 回复评论

已登录用户可以针对某个用户的某条评论进行回复，同时用户可以在其个人中心发现其他用户对自己的回复。

5 系统测试

5.1 PC 浏览器端

平台首页展示了系统目前的数据概要，以及对主要项目类型的简介与入口。除提供基本的导航条外，用户也可以从直接搜索感兴趣的项目名，以及从任何页面登录、注册等，如图 5-1 所示。

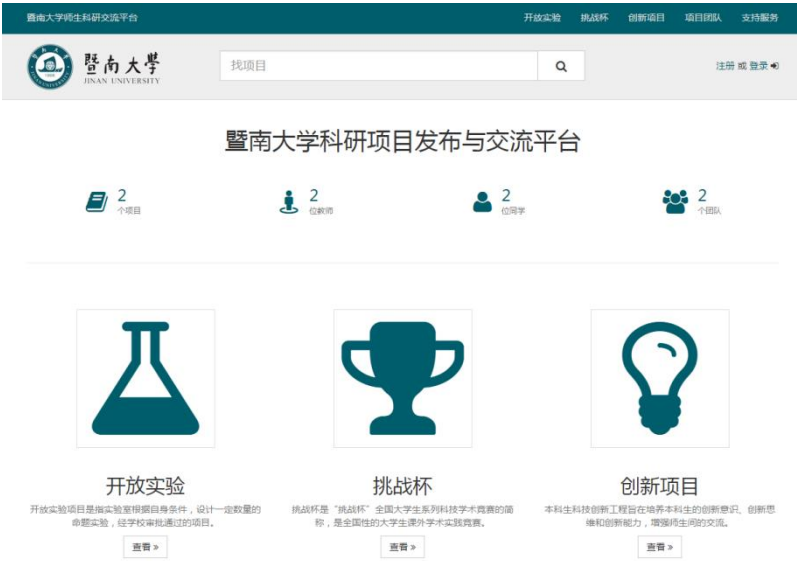


图 5-1 平台首页



图 5-2 学生创建项目

教师可以创建所有项目，学生可以创建除开放实验外的项目。学生创建的项目需要有指导教师加入后才能激活并允许团队加入。学生与教师在此期间可以在项目内进行交流，如图 5-2 所示（上一页）。

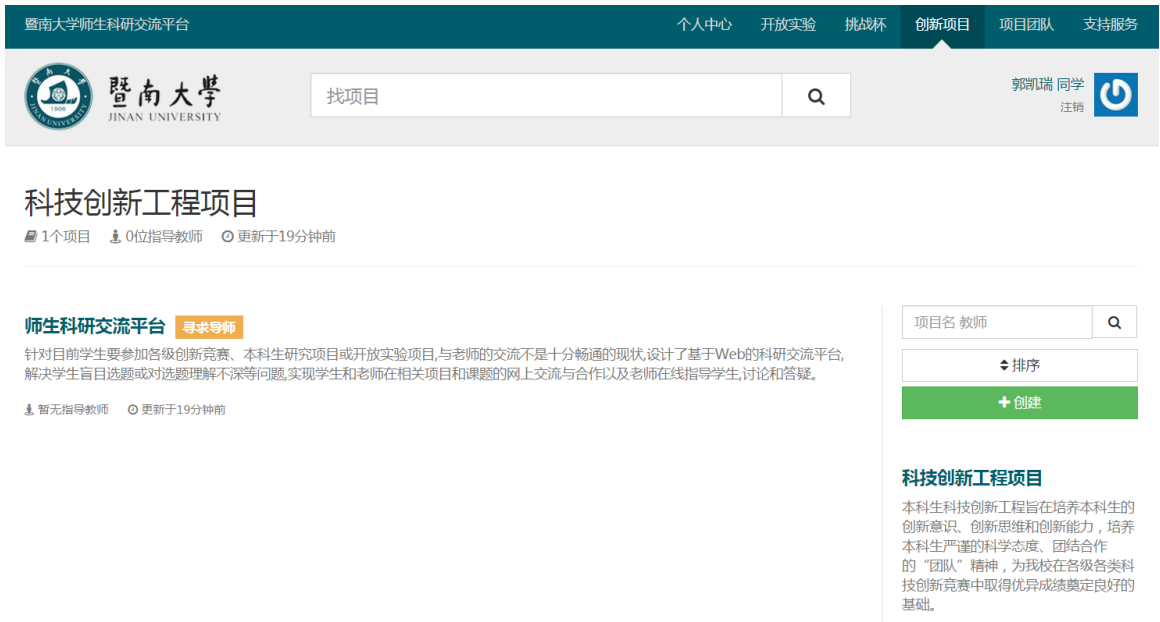


图 5-3 创建成功

项目列表页，如图 5-3 所示，对于一个项目在列表页中主要显示的是名称、简介、进行状态以及导师、更新时间。用户可以在结果中搜索，也可以根据要求对结果进行排序以快速查找。“创建”按钮只会在用户权限足够时显示。

以下为项目具体页，左侧显示的是关于项目的一些主要信息，右侧则是附加信息。右侧上方有用户操作区，根据用户角色的不同会有不同的按钮，如更新、选课、删除等。页面提供了分享接口以方便用户将项目分享至社交网络以寻找感兴趣的同学或教师。

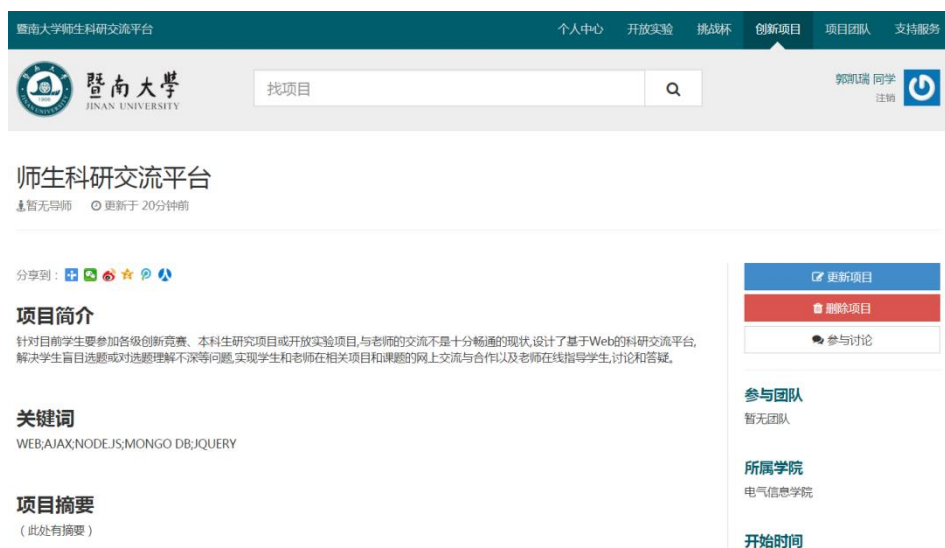


图 5-4 查看项目

经费预算

(此处有经费预算)


评论 (0 条评论)

对项目有任何疑问都可以在这里与老师或同学交流

说点什么


A ▼

Tt ▼


 ▼


B


I





U

















求指导

8294

8294

取消

发布评论

图 5-5 发表评论

项目页的下方是评论模块，已登录的用户可以在此发表评论，是平台实现师生交流的主要方法。用户也可以对他人的评论进行回复，被回复者将在其个人中心收到所有对自己的评论的回复，如图 5-5 所示（上一页）。

学生可以直接申请教师开设的开放实验项目，也可以组建团队后由负责人申请加入挑战杯与创新工程项目。申请后指导教师可以在个人中心对同学提交的申请进行处理，选择接收或者拒绝。

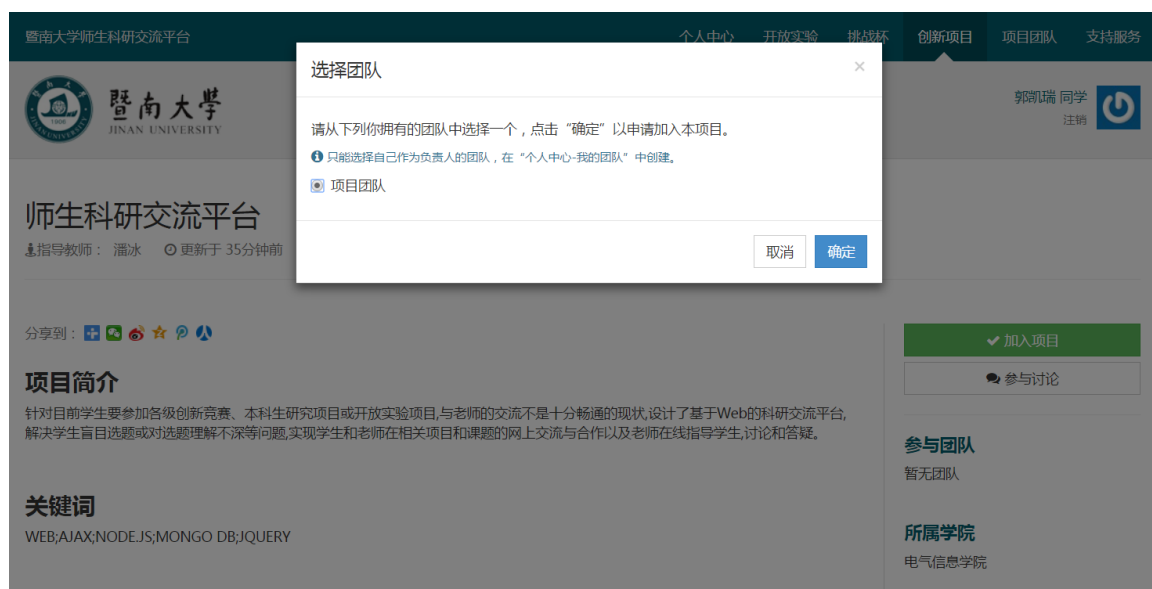


图 5-6 加入项目

5.2 移动设备浏览器端

手机端使用本平台无需安装任何软件，只需与 PC 端一样在浏览器输入网址即可打开。平台能够适应小屏设备自动调整布局，导航点击后将向下扩展，同时页面的布局也发生了变化，如图 5-7。

学生创建科技创新工程项目的页面，如图 5-8。页面上的所有控件都能够根据屏幕大小自动调整外观及布局。



5-7 首页

新建科技创新工程项目 注意: ×

同学发布的项目在寻求指导教师后方可激活, 激活后项目所有权归指导教师。

项目名称

基于Web的师生科研互动平台

所属学院

电气信息学院 ▼

预期开始时间

2015-05-14 × 日历

预期结束时间

2015-05-31 × 日历

项目简介 (此处内容将显示在列表摘要中)

针对目前学生要参加各级创新竞赛、本科生研究项目或开放实验项目, 与老师

5-8 学生创建项目

项目列表页，如图 5-9 所示。出于对内容优先级以及屏幕宽度限制的考虑，侧边栏的内容将变为在页面下方呈现。其它页面同理。

项目具体页，如图 5-10 所示，具有良好的阅读体验。



5-9 创建成功

基于Web的师生科研互动平台

暂无导师 更新于 8分钟前

分享到：

项目简介

针对目前学生要参加各级创新竞赛、本科科研项目或开放实验项目,与老师的交流不是十分畅通的现状,设计了基于Web的科研交流平台,解决学生盲目选题或对选题理解不深等问题,实现学生和老师在相关项目和课题的网上交流与合作以及老师在线指导学生,讨论和答疑。

5-10 查看项目

项目具体页内的评论模块，如图 5-11 所示。

同学选择团队以加入项目的场景，如图 5-12 所示。

对项目有任何疑问都可以在这里与老师或同学交流

说点什么

A ▼ TI ▼

B I S U

求指导

2269

2269

取消

发布评论

5-11 发表评论

选择团队

×

请从下列你拥有的团队中选择一个，点击“确定”以申请加入本项目。

i 只能选择自己作为负责人的团队，在“个人中心-我的团队”中创建。

☐ 项目团队

取消

确定

参与团队

暂无团队

所属学院

电气信息学院

开始时间

5-12 加入项目

结论

研制开发本平台的目的，主要是方便学生参加各级创新竞赛、本科生研究项目或开放实验项目，以及方便学生与教师在整个项目过程中的交流，克服学生盲目选题或对选题理解不深问题。平台服务器端采用 Node.js 技术搭建，以 Mongo DB 作为数据库，搭配以目前热门的 Web 前端技术制作。在选择年轻、快速发展的技术的同时，考虑到移动设备访问的方便性以及必要性，为平台设计了适应移动设备的用户界面。以上两点是本平台的最大的创新之处。经过了近两个月的开发、调整与测试，很好地达到了当初设想的目标，弥补了暨南大学在学生科研与教师交互缺乏互动平台的不足。

在完成毕业设计期间，我遇到了许多问题，其中有自己翻阅资料解决的，有询问指导教师解决的，也有询问同学解决的。最终能顺利完成毕业设计，大家的帮助是非常重要的。

在我看来，毕业设计是对学生所学专业知识的综合检验。虽然在学校期间并没有开设过相关技术的课程，但是核心理念都是一致的，就是软件工程思想，软件架构设计与设计模式等。同时，Web 编程技术的思想也是非常重要的，如何提高性能，如何防范攻击，如何提高用户体验等，都是完成一项 Web 应用最需要的知识。

致谢

感谢我的导师潘冰老师，谢谢他对我的悉心指导。在毕设选题以后我对题目有非常多的疑惑之处，潘冰老师对我一一进行了说明与指导，在技术选择方面也给予了关键的支持。

感谢我的班主任陈双平老师，在大学期间接收我作为 ACM 成员，虽然由于个人原因没有坚持到最后，但期间收获了非常多宝贵的知识。

感谢周密老师，在软件工程设计理念系列课程中学到的知识，是大学后半段时期中非常重要的收获。

同时也感谢四年来教导过我的其它老师，感谢陪伴我走过大学生活的同学们，感谢支持我的父母。

毕业设计中使用到了大量开源框架、作品以及服务，参考了大量前人的经验、书籍等，可以说是站在巨人的肩膀上完成的，感谢他们的无私奉献。

参考文献

- [1] 许会元,何利力.NodeJS 的异步非阻塞 I/O 研究.工业控制计算机,2015, 28(3): 127-129
- [2] 张善友.分布式文件储存的数据库开源项目 MongoDB. (2010-05-20)[2015-05-04].
<http://www.cnblogs.com/shanyou/archive/2010/05/20/1739987.html>.
- [3] 杨国瑞,张思博.基于 AJAX 的 Web 应用架构设计.现代电子技术, 2006, 15: 95-98
- [4] OurJs.现在你为什么应该学 Node.js.[CP/OL]. (2014-05-10)[2015-05-12].
<http://ourjs.com/detail/536e3ef99c8a60846e000010>
- [5] Jack Vaughan.NoSQL 数据库家族图谱.[CP/OL]. (2014-08-06)[2015-05-12].
http://www.searchbi.com.cn/showcontent_84063.htm
- [6] TJ Holowaychuk.Express 中文开发文档.[CP/OL]. (2012)[2015-05-12].
<http://www.expressjs.com.cn/>