

**本科生毕业论文(设计)**

**基于Web的许昌学院大学生竞赛管理系统设计与实现**

**学 院 信息工程学院**

**专 业 网络工程**

**班 级 2014级本科1班**

**学 号 5011140023**

**学 生 姓 名 申远**

**联 系 方 式 17603859510**

**指 导 教 师 陈艳格 职称：副教授**

2018年 4 月

基于Web的许昌学院**大学生**竞赛管理系统设计与实现

摘 要

当今社会是一个高度信息化，数据化的时代。在管理计算机数据时，为了方便人们操作，一些基于web的管理系统便应运而生。而对于高校的发展而言，发展智慧型校园也是势不可挡。当前，针对学生的竞赛数不胜数，伴随着竞赛的规模，数量不断扩大，教务部门传统的纸质档案记录成绩不仅会出现一些差错，而且查找困难，总结分析难度大。针对这一情况，采用计算机技术开发竞赛管理系统，使得高校学生的竞赛成绩实现信息化管理。

本文主要讲解了竞赛管理系统的开发背景，需求分析，数据库设计，功能模块几个方面。为了提高学校的统计工作效率和方便学生的成绩信息管理，决定设计并实现一套基于HTML和PHP技术的竞赛信息管理系统。该系统集竞赛信息管理、学生审批管理、数据库导出表格下载、数据分析等功能于一体，采用统一的浏览器平台，向学生及教师提供智能化服务。

关键词：网站开发，竞赛管理，PHP，MYSQL数据库

**Design and Implementation of Xuchang University College Students' Competition Management System Based on Web**

**ABSTRACT**

Today's society is an era of highly information and data. When managing computer data, some web-based management systems came into being to facilitate people's operation. For the development of colleges and universities, the development of a smart campus is also unstoppable. At present, the number of contests for students is innumerable. As the scale of the contest continues to expand, the records of the traditional paper archives of the educational administration department will not only cause some errors, but will also make it difficult to find the documents. In view of this situation, the use of computer technology to develop a competition management system enables college students to achieve information management.

This article mainly explains the development background, requirements analysis, database design and functional modules of the competition management system. In order to improve the statistical work efficiency of the school and facilitate the management of student achievement information, we decided to design and implement a competition information management system based on HTML and PHP technology. The system integrates functions such as contest information management, student approval management, database export form download, and data analysis, and uses a unified browser platform to provide intelligent services to students and teachers.

**Key words：**Website Development, Contest Management, PHP, MYSQL Database

目录

[1．前言 - 1 -](#_Toc512549134)

[1.1研究现状 - 1 -](#_Toc512549135)

[1.2研究的目的及意义 - 2 -](#_Toc512549136)

[1.3相关技术介绍及分析 - 2 -](#_Toc512549137)

[2．需求分析 - 2 -](#_Toc512549138)

[2.1项目需求 - 3 -](#_Toc512549139)

[2.2系统性能需求 - 3 -](#_Toc512549140)

[2.2环境需求 - 3 -](#_Toc512549141)

[2.2.1 版本控制工具git简介 - 3 -](#_Toc512549142)

[2.2.2 angular/cli简介 - 4 -](#_Toc512549143)

[2.2.3 ng-zorro-antd简介 - 4 -](#_Toc512549144)

[2.2.4 后台服务器环境 - 4 -](#_Toc512549145)

[3．数据库设计 - 5 -](#_Toc512549146)

[3.1 数据库功能流程 - 5 -](#_Toc512549147)

[3.2数据库及数据表的建立 - 5 -](#_Toc512549148)

[3.3数据操作类的封装 - 7 -](#_Toc512549149)

[4．系统概要设计 - 10 -](#_Toc512549150)

[4.1系统架构设计 - 10 -](#_Toc512549151)

[4.1.1前端页面架构 - 10 -](#_Toc512549152)

[4.1.2后端脚本架构 - 14 -](#_Toc512549153)

[4.2系统模块划分 - 15 -](#_Toc512549154)

[5．系统详细设计 - 17 -](#_Toc512549155)

[5.1学生用户 - 17 -](#_Toc512549156)

[5.2教师用户 - 22 -](#_Toc512549157)

[5.3公共模块 - 28 -](#_Toc512549158)

[6．单元测试 - 29 -](#_Toc512549159)

[6.1编写单元测试脚本 - 30 -](#_Toc512549160)

[6.2覆盖率报告 - 33 -](#_Toc512549161)

[6.3功能测试 - 34 -](#_Toc512549162)

[结束语 - 41 -](#_Toc512549163)

[参考文献 - 42 -](#_Toc512549164)

[致谢 - 43 -](#_Toc512549165)

基于Web的许昌学院大学生竞赛管理系统设计与实现

# 1．前言

当今社会是一个高度信息化的社会，不管是人工智能、大数据还是区块链技术，都离不开数据信息的支撑。所以，对于信息的收集，存储以及分析就显得尤为重要。随着计算机技术越来越广泛地被用在各行各业，为方便用户操作，都有专用的信息管理系统供人们使用。像这种仅供一个部门或机构的专用系统，通常采用较为轻量级的，开发周期短的PHP+MySQL技术来实现。此次系统的开发就采用相关技术，根据本人现有的工具和知识，完成许昌学院竞赛管理系统的开发。

## 1.1研究现状

由于学科竞赛领域规模的不断扩大，所牵涉到的事物方方面面。管理员对各院校学生所参加的竞赛信息的管理工作日益繁琐，因此当前院校急切地需要一个使用简便、运行安全、快速高效的平台来管理这些竞赛成绩的信息。

现如今，高校在统计学生竞赛成绩信息时，主要面临以下几个问题：

（1）高校院系，班级由于分布较多，不利于成绩的归纳与统计，并由于如果参赛人数众多，统计工作也很繁琐。

（2）在学生进行的竞赛获得成绩时，往往是根据官方网站自主查询获得，然后上报给其指导老师或学校，过程复杂。

（3）学生竞赛的指导老师需要针对学生的竞赛信息，进行逐一审核，浪费时间，人力与精力，得不偿失。

目前，不管是国家或省级设置的各种竞赛，还是社会上的一些认证考试，针对高校的竞赛是各种各样的，德智体美劳，都有相应的竞赛供学生去展示自己。针对这些种类繁多，奖项设立复杂的竞赛，不仅是对学生个人，还是对学校收集，整理竞赛信息都是非常困难的。那么这就需要一套系统将这些信息收集整理出来。方便人们查阅。而学校也可以根据这些竞赛成绩，来作为评估学生的优秀程度的标准，因此，此次论文得选题，就是针对这个方向，开发出这套系统。

## 1.2研究的目的及意义

为实现提升学院管理部门的工作效率，充分利用资源，方便管理员全方面地掌控学院学生的竞赛成果情况等目的，专门研发了此管理系统。系统针对当前管理部门在进行学生竞赛成果的统计工作时存在的各方面难题，提供一个比较可行的解决办法，使得工作效率有所提升，学校资源得以充分利用跟有效配置。

根据学院的实际条件，运用个人所具有编程技术条件，发散思维，开发出一套关于高校竞赛管理的系统来实现一些基本信息的管理。为我们院校的科研管理人员减轻些压力，从根本上来解决工作量大且特别容易出错的问题，同时，这也是智慧校园未来发展的必然趋势。

## 1.3相关技术介绍及分析

HTML技术

HTML，中文名为超文本标记语言。即是万维网上一个超媒体文档，它能够告诉浏览器如何显示其中的内容（如：文字如何处理，画面如何安排，图片如何显示等）。HTML编写并不复杂，但是功能及其强大。目前已发展到了HTML 5.0，简称H5。

Angular技术

Angular是一个以JavaScript编写的框架，是构建前端应用的一款优秀插件。其诞生后被Google收购，然后被广泛的应用在谷歌的多款产品中。Angular有着诸多特性，最为核心的是：MVW（Model-View-Whatever）、模块化、自动化双向数据绑定、语义化标签、依赖注入等等。

PHP技术

PHP（外文名:PHP: Hypertext Preprocessor，中文名：“超文本预处理器”）是一种通用开源脚本语言。语法吸收了众多语言的特点，利于学习，使用广泛，主要适用于Web开发领域。用PHP做出的动态页面与其他的编程语言相比，PHP是将程序嵌入到HTML（标准通用标记语言下的一个应用）文档中去执行；PHP还可以执行编译后代码，编译可以达到加密和优化代码运行，使代码运行更快。

# 2．需求分析

管理系统整体上采用C/S架构实现，应分为客户端(client)和服务端（serve）两种。服务端负责存储用户信息、竞赛信息、审批信息等web端所需要的相关信息，并向web端提供用户登陆、用户密码修改、竞赛信息录入、审批信息录入、用户信息下载等操作接口；web端负责用户体验、强弱提示、展示学生信息、教师信息、审批信息等的UI设计与操作相关功能。

## 2.1项目需求

许昌学院竞赛管理系统，是针对竞赛成绩信息进行统一管理，分析的一套系统。基于教师和学生两个用户角色。主要实现的功能包括以下几个方面：

（1）用户的登录与修改密码；

（2）学生用户发起竞赛成绩审批与修改；

（3）教师用户针对竞赛成绩进行审批；

（4）所有审批的模糊查询，排序以及数据下载；

（5）用户个人信息的展示与修改；

## 2.2系统性能需求

（1）程序能够在PC上正常运行，实现所有功能；

（2）程序流畅运行无bug，针对大量数据能够快速响应；

（3）有完善的功能说明，包括系统设计，算法实现，测试结果等；

（4）针对前端体验进行优化；

## 2.2环境需求

工欲善其事必先利其器，一个符合用户操作习惯，有着良好交互效果的网站或者系统，都要从确认需求开始，进行需求分析，页面设计，最后开始编程。在这一整套流程中，都需要相应的工具或者说环境来支持开发。项目启动后，针对代码管理的版本控制工具git、服务端采用PHP+MYSQL的Apache环境，客户端使用的node环境，以及针对前端代码进行打包压缩的webpack配置。

### 2.2.1 版本控制工具git简介

Git是一款开源的分布式版本控制系统，常用于多人协作开发。不管项目的大小，都可以快速高效的管理项目版本，是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。

Git具有以下几个特点

（1）直接记录快照，而不是差异比较

（2）近乎所有操作都是本地执行

（3）时刻保持数据完整性

### 2.2.2 angular/cli简介

Angular CLI俗称Angular脚手架工具，是一个命令行接口，可用于实现前端自动化开发工作流程。使用Angular/cli可以快速构建并运行一个Angular应用程序，并带有开发服务器，可以在开发过程中预览应用程序，以及相应的单元测试，都有集成。因此在此管理中系统选用Angular/cli。以下是常见的命令

1. ng init my-app (项目初始化)
2. ng g component my-new-component (新建一个组件)
3. ng g module my-module(新建一个模块)
4. ng g service my-new-srevice(新建一个服务)
5. ng serve (编译angular项目，运行在浏览器环境中)
6. ng bulid (项目打包，可选—prod参数，能够极大的压缩项目)
7. ng test (执行单元测试，可选参数，生成覆盖率报告)

### 2.2.3 ng-zorro-antd简介

Ng-zorro-antd是一款由阿里开发并开源的一套基于angular5的组件化框架，他与angular5之间的关系就像bootstrap和jQuery一样。通过封装组件，实现即拿即用的效果。配合angular5的双向数据绑定。可以达到快速构建后台的效果。（https://ng.ant.design）。

### 2.2.4 后台服务器环境

后台服务器选用Apache+PHP+Mysql，针对环境的安装，选用一款快速简便的集成包Xampp。XAMPP同样是一款开源、免费的网络服务器软件，经过简单安装后，就可以在个人电脑上搭建服务器环境。所包含的文件：Apache服务器, MySQL数据库, PHP, Perl, FTP 服务程序和phpMyAdmin数据库图形化管理。

# 3．数据库设计

## 3.1 数据库功能流程

## 3.2数据库及数据表的建立

数据库的设计是本系统设计的核心部分，对数据库的设计不仅要考虑到数据库的性能同时还应该考虑到后期的维护与扩展。此系统数据库中建立4张表，如表3.1~3.4所示。

表3-1 用户信息表（account）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 备注 | 是否为空 | 键值类型 |
| 1 | account\_id | int(255) | 用户id | 否 | 主键 |
| 2 | account | varchar(255) | 账号 | 否 |  |
| 3 | password | varchar(255) | 密码 | 否 |  |
| 4 | Auth | int(128) | 权限 | 否 |  |
| 5 | Createtime | int(128) | 创建时间 | 否 |  |
| 6 | Name | varchar(8) | 姓名 | 否 |  |
| 7 | School | varchar(8) | 学校 | 否 |  |
| 8 | Depart | varchar(8) | 院系 | 是 |  |
| 9 | Profess | varchar(8) | 专业 | 是 |  |
| 10 | Gard | varchar(8) | 年级 | 是 |  |

表3-2 审批表（application）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 备注 | 是否为空 | 键值类型 |
| 1 | app\_id | int(255) | 审批Id | 否 | 主键 |
| 2 | User\_id | int(255) | 用户id | 否 |  |
| 3 | game\_name | varchar(8) | 竞赛Id | 否 |  |
| 4 | item\_name | char(16) | 项目id | 否 |  |
| 5 | radiovalue | int(8) | 获奖类型 | 否 |  |
| 6 | garde | char(128) | 分数或奖项 | 否 |  |
| 7 | status | int(8) | 审批状态 | 否 |  |
| 8 | teacher | int(128) | 审批教师 | 否 |  |
| 9 | desc | char(255) | 审批描述 | 是 |  |
| 10 | time | int(128) | 操作时间 | 否 |  |

表3-3 竞赛信息表（contest）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 备注 | 是否为空 | 键值类型 |
| 1 | contest\_\_id | int(255) | 用户id | 否 | 主键 |
| 2 | Name | varchar(16) | 竞赛名称 | 否 |  |
| 3 | Gard | int(8) | 竞赛级别 | 否 |  |
| 4 | Type | int(8) | 竞赛类型 | 否 |  |
| 5 | Mode | int(8) | 参与方式 | 否 |  |
| 6 | Radiovalue | varchar(8) | 奖项方式 | 否 |  |
| 7 | Award | char(255) | 奖项 | 是 |  |
| 8 | Item | char(255) | 比赛项目 | 是 |  |
| 9 | Status | int(8) | 状态 | 否 |  |
| 10 | End\_time | int(128) | 截止日期 | 是 |  |
| 11 | Update\_name | Varchar(6) | 最后操作人 | 否 |  |

表3-4 审核表（review）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 备注 | 是否为空 | 键值类型 |
| 1 | review\_\_id | int(255) | 审核id | 否 | 主键 |
| 2 | TeachID | varchar(16) | 审批人 | 否 |  |
| 3 | User\_id | int(8) | 提交人 | 否 |  |
| 4 | Appli\_id | int(8) | 竞赛Id | 否 |  |
| 5 | status | int(8) | 审核结果 | 否 |  |
| 6 | Error\_des | varchar(8) | 异常反馈 | 是 |  |
| 7 | Op\_time | char(255) | 操作时间 | 否 |  |

通过以上表的详细信息即可完成该系统核心业务的数据存储工作。由于考虑到后期维护和扩展工作，故有数据表的外键关联不在数据库进行设计，均放到程序去控制外键关联等关系。

## 3.3数据操作类的封装

系统采用PHP操作mysql数据库，PHP提供了较多的可操作性函数，例如PDO和mysqli。此系统采用PDO进行数据库连接，好处就是兼容了更多的数据库，同样的语法可以访问不同数据库。由于数据操作偏向于底层，经常被各种文件使用，因此将数据操作进行封装，方便使用与维护。以下是核心代码：

（1）数据操作对象封装

pdoclass.php

class PdoMysql{

public static $config=array (); //设置数据库链接参数

public static $link=null; //保存链接标识符

public static $PDOStatement=null;//保存statenment对象

public static $sqlStr=null; //数据库语句

public static $error=null; //出错语句

/\*

\*定义链接数据库的构造方法

\*/

public function \_\_construct($dbConfig=''){

if (!class\_exists("PDO")) {

self::throw\_exception('不支持PDO，请先开启');

}

if(!is\_array($dbConfig)){

$dbConfig=array(

'hostname'=>DB\_HOST,

'username'=>DB\_USER,

'password'=>DB\_PWD,

'database'=>DB\_NAME,

'dbms'=>DB\_TYPE,

'dsn'=>DB\_TYPE.":host=".DB\_HOST.";dbname=".DB\_NAME

);

}

try{

self::$link=new PDO($dbConfig['dsn'],$dbConfig['username'],$dbConfig['password']);

}catch(PDOException $e){

self::throw\_exception($e->getMessage());

}

if (!self::$link) {

self::throw\_exception('PDO连接错误');

return false;

}

self::$link->exec('SET NAMES '.DB\_CHARSET);

return self::$link;

}

以上代码实现的是定义一个pdo类，在构造函数中进行连接数据库，成功后将结果传给link属性，便于类中的方法使用，错误就抛出异常返回；连接成功后就可以进行数据库操作，以下是定义操作类的数据库查询方法：

public static function query($sql=null){

$link=self::$link;

self::$sqlStr=$sql;

if(!$link){ return false;};

//判断之前是否有结果集，如果有的话，释放结果集

if(!empty(self::$PDOStatement)){self::free();};

// self::$queryStr=$sql;

self::$PDOStatement=$link->prepare($sql);

$res=self::$PDOStatement->execute();

self::haveErrorThrowException();

return $res;

}

pdoclass.php

以上代码实现的是定义一个数据库查询方法，需要传入正确的数据库操作语句，通过操作link-> prepare()，来执行数据库查询语句。成功返回查询结果，错误抛出异常并返回。下面是使用方法，已登录为例：

loginControll.php

function login($account,$password){

$pdo=new PdoMysql();

$res = $pdo->query("SELECT \* FROM `account` WHERE `account`='{$account}' and `password`='{$password}' ");

if ($res!=null) {

unset($res[0]['password']);

return jsonCode(200,$res[0]);

}else {

return jsonCode(300,null);

}

}

loginControll.php

以上代码实现的就是登陆功能，可以看到使用时，直接进行PDO实例化new PdoMysql()，然后在执行实例化对象下的查询方法，将数据库查询语句传入值查询方法里，并返回查询到的用户信息。

可以看到经过封装后的数据库操作，在使用时十分方便。不仅如此，我们还可以接着进一步的扩展pdo操作类，添加比如进行唯一性验证的方法、进行事务处理的方法等等。更加方便我们的操作，也便于以后的修改维护。

# 4．系统概要设计

## 4.1系统架构设计

系统整体采用前后端分离的模式。前端请求后台接口，获得数据并展示；后台向外暴露api，仅负责输入输出。由此达到前后分离的效果。它的优点显而易见。前后端各司其职，前端负责获取数据与展现，后端负责输入输出以及逻辑处理，这是前后端混编所不能达到的。

需要注意的是。在使用前后端分离的模式时，Apache服务器需要开启跨域的权限。这是因为浏览器的一种同源保护策略：即一个域内的脚本仅具有本域内的权限，本域脚本只能读写本域内的资源，而无法访问其它域的资源。相同ip，相同协议，相同端口可以称为一个域。前端采用angular5技术，会启用一个服务占用4200端口，而Apache服务器则会占用80端口，由此会产生跨域问题。

解决方法即修改Apache的httped.conf，添加Header set Access-Control-Allow-Origin:。\*号指对所有域名开放，http：//www.xcu.edu.cn指仅针对此域名开放权限。如图4-1所示。



图4-1 Apach http.conf 修改文件

### 4.1.1前端页面架构

前端使用angular5为主要技术栈进行开发， 使用angular/cli命令行工具进行项目初始化：ng new scms。然后ng serve启动项目，我们就会得到项目的初始模样。接下来就是对前端架构的搭建。

（1）引入ng-Zorro

进入项目文件里面，在命令行模式下执行命令npm install ng-zorro-antd –save，然后修改appmodule.ts 文件，修改后再运行可以看到页面已经改变。

appmodule.Ts

import { BrowserModule } from '@angular/platform-browser';

import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

import { NgModule } from '@angular/core';

import { FormsModule } from '@angular/forms';

import { HttpClientModule } from '@angular/common/http';

import { NgZorroAntdModule } from 'ng-zorro-antd';

import { AppComponent } from './app.component';

@NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule,

FormsModule,

HttpClientModule,

BrowserAnimationsModule,

NgZorroAntdModule.forRoot()

],

bootstrap: [AppComponent]

})

export class AppModule { }

appmodule.Ts

（2）封装http服务

由于我们是前后端分离的项目，在与后台接口通信时就需要引入Http服务来发送http请求，而这样的请求几乎是存在每个组件里。所以为了方便，需要对http服务进行封装。建立一个api.service.ts文件：

api.service.ts

public async httpservice(url: string, params: any, search?: any): Promise<any> {

params = search?Object.assign(params, search):params;

return await this.http.post(url, JSON.stringify(params))

.map(e => this.handleSuccess(e))

.catch(e => this.handleError(e)).toPromise();

}

public handleSuccess(res: Response) {

let body = res.json();

if (body) {

return body;

}

}

public handleError(res: Response) {

return this.confirmServ.error({

title: '错误',

content: '服务器出错，请稍后再试'

});

}

首先使用es6的语法糖async异步修饰符修饰httpservice函数，它会返回一个promise的异步对象。针对回调对象使用handlesuccess和handleerror过滤。如果返回的是正常json数据的格式，则将字符串对象化返回。否则弹出服务器出错的弹框。函数参数后跟类型修饰（typescript语法），限定参数类型。

以登陆功能为例,使用方法如下：

login(account:string,password:string) {

const url =this.host+’ /api/login.php';

return this.httpservice(url,

{ account: account,

password: password

})

}

api.service.ts

login.component.ts

this.\_api.login(this.user\_name, this.user\_password).then(e => {

// 登陆成功后的逻辑处理

}).catch(e=>console.log(e))

login.component.ts

定义登陆功能函数，传入账号,密码。执行login函数。然后在then（）中的回调中执行后续逻辑操作。由于js的单线程限制，程序在执行过程中遇到错误就会中断执行，导致异常。所以在执行通信函数时，要追加catch（）回调，以保证在网络请求异常时会抛出异常而不中断程序执行。如图为登陆时的网络请求，携带账号和密码。与之对应的响应报文，登陆成功后返回用户信息。如图4-1和4-2所示。

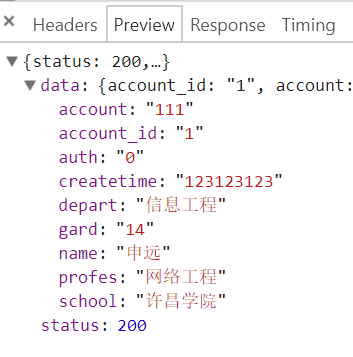
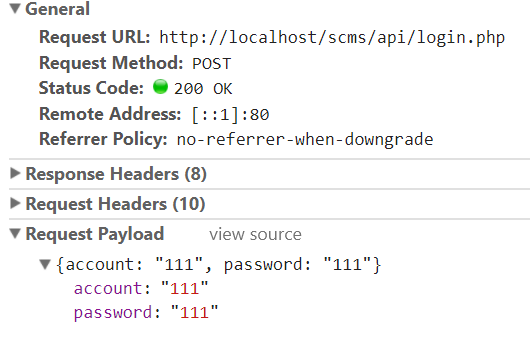


图4-1 登陆请求 图4-2 登陆请求响应

（3）基于角色的访问控制

此系统设定有两种角色：学生和教师。那么针对两种角色分别有不同的权限，对于权限的分配，数据表中以auth字段的值来判断。不同角色展示不同的功能，就需要angular5中的路由守卫来操作。路由守卫常用的有：1、CanActivate: 处理导航到某个路由的情况；2、CanDeactivate:处理从当前路由离开的情；3、Resole:在路由激活之前获取路由数据。

teacher.guard.ts

import{CanActivate,ActivatedRouteSnapshot,RouterStateSnapshot}from'@angular/router';

export class TeacherGuard implements CanActivate {

canActivate(route: ActivatedRouteSnapshot, state: RouterStateSnapshot): boolean {

if (window.localStorage.getItem('auth')=='1') {

return true;

}else{

return false;

}

}

}

teacher.guard.ts

以上代码实现的是：在进入的页面之前，从缓存中读取当前用户的所属权限是否与教师权限对应的字段相同，如果相同证明当前用户即为教师，则返回真，进入页面，否则返回false。拒绝进入该页面。使用方法是将导出的TeacherGuard注入到appmodule中，再在只允许教师进入的路由中进入就可实现。

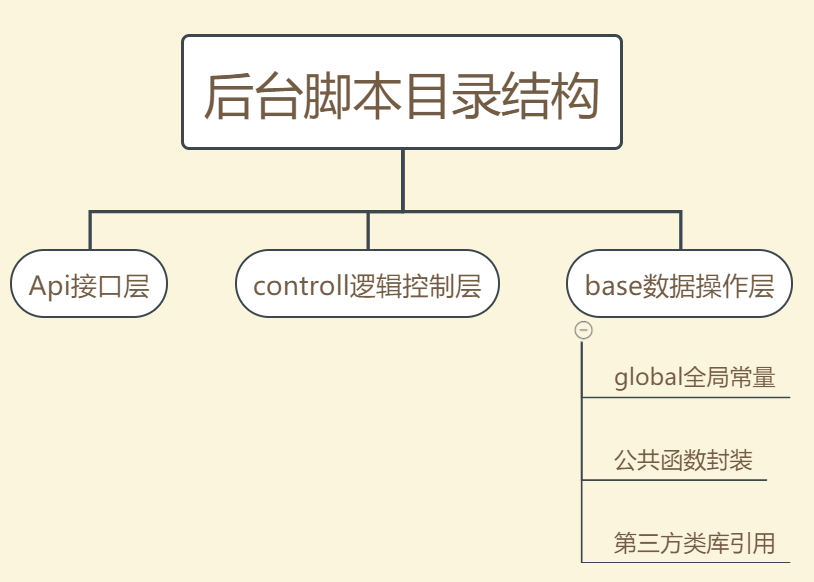
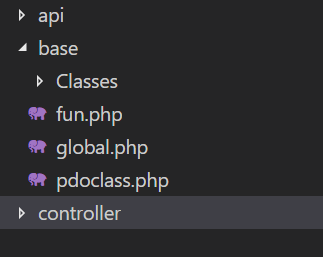
### 4.1.2后端脚本架构

后台将分为三层组织架构，即Api接口层，Controller控制层，Base数据操作层。如图4-3和4-4这样划分结构，使得后台条理清晰，结构明确，便于前期bug修复以及后期迭代和维护。

（1）接口层负责向外暴露Api请求接口，只负责获取网络请求参数以及返回对应结果，与控制层一一对应。

（2）控制层负责逻辑处理，与接口层一一对应，主要是根据传入参数，进行数据查询，将查询的结果作进一步的逻辑处理，并返回给接口层。

（3）数据操作层负责数据查询，分为全局变量定义，公用函数封装和第三方类库引用；主要负责数据库查询操作。

 图4-3 后台脚本结构 1 图4-4 后台脚本结构 2

基于以上的组织结构，绘制出后台工作流程图。如图4-5所示。他的工作流程如下：

1. 前端发送http请求后端暴露的Api接口，报文携带请求参数；
2. Api获得前端请求参数，引入对应控制器，调用控制器函数，并将结果返回给前端；
3. 对应控制器获得参数，引入全局常量以及数据库操作类，接着进行数据库语句拼接并执行，将获得返回结果进行相应的逻辑处理，并进行json格式化，将最后的字符串返回给上一层；
4. 数据操作层获得数据库查询语句，链接数据库并执行操作，并将查询结果返回上一层。

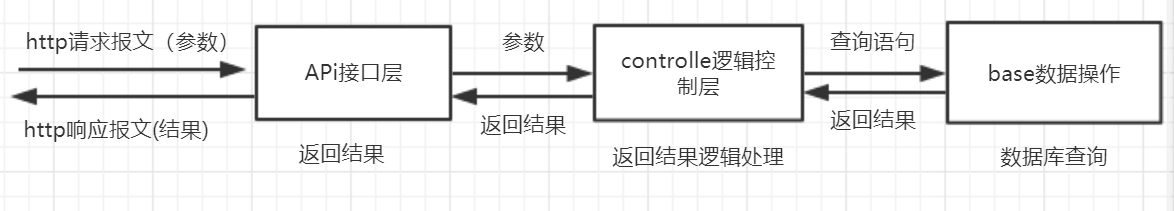


图4-5 后台工作流程图

## 4.2系统模块划分

主要以模块化、组件化的思想进行页面拆分与布局。然后再以搭积木的方式将网页应用搭建出来。整体采用低耦合，高内聚的模式。耦合，即两个或两个以上的模块之间的输入输出关系，低耦合就是要将模块间的输入输出降到最低，期望达到模块之间能够相互独立运作，而不依赖其他模块。高内聚，是指将整个网页应用细致划分模块，模块再划分组件，达到整体是以组件高度聚合而成。

根据低耦合，高内聚的思想，再综合数据库的设计，将系统分为以下几个模块。如图4-6所示

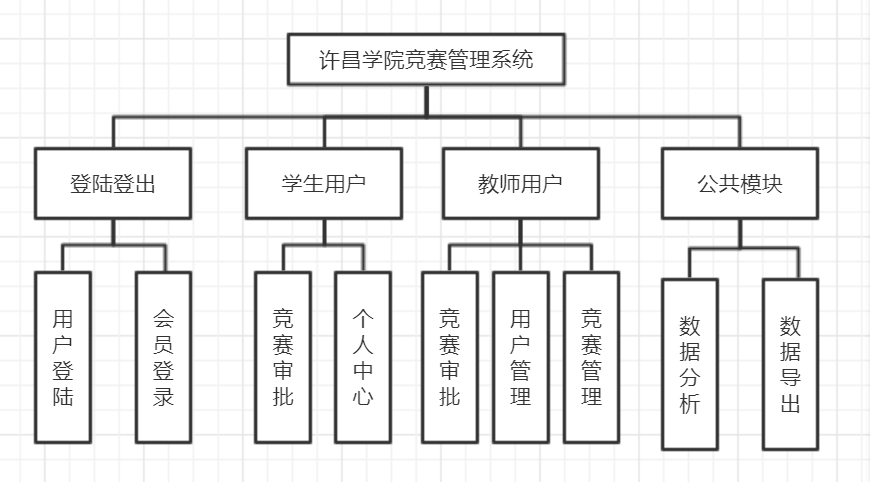


图4-6 前端模块划分示意图

（1）学生用户的设计

学生使用该系统进行竞赛成果存储。包括 登入登出。个人中心数据统计，竞赛流程发起审批，修改以及模糊查询。

（2）教师用户的设计

教师使用该系统进行竞赛成果的审批。包括登入登出，竞赛流程的审批，模糊查询。学生用户管理的排序，模糊查询。竞赛管理的增加，修改，模糊查询的排序。竞赛数据的统计分析。

# 5．系统详细设计

## 5.1学生用户

登陆模块，需要输入账号密码，密码经过md5转换后存入数据库，需要注意的是加密后的哈希值为128，数据库密码字段的设计要符合响应字节长度，否则存入的会是被截断的哈希值，那么在进行密码比对的时候肯定会出错导致登陆不成功。

（1）登陆功能-前端逻辑

login.component.ts

login() {

if (this.user\_name&&this.user\_password) { // 判断当前时候有账号密码

this.\_api.login(this.user\_name, this.user\_password).then(e => {

// 进行登录请求

if (e.status == 200) {

// 登陆状态正常

window.localStorage.setItem('token',e.data.account\_id)

window.localStorage.setItem('name', e.data.name)

window.localStorage.setItem('auth', e.data.auth)

window.localStorage.setItem('info',JSON.stringify(e.data))

this.\_route.navigate(['/index/notice'])

} else {

this.notice = true; // 登陆异常 显示提示

}

}).catch(e=>console.log(e)) // 抛出异常

}else{

this.notice = false;

}

}

login.component.ts

（2）登陆功能-后端逻辑

loginControll.php

<?php

include\_once '../base/pdoclass.php';

include\_once '../base/fun.php';

function login($account,$password){

$pdo=new PdoMysql(); // 数据操作类实例化

$password = md5($password)；// 将传入的密码进行md5加密

$res = $pdo->getAll("SELECT \* FROM `account` WHERE `account`='{$account}' and `password`='{$password}' ");

if ($res!=null) { // 如果查询出结果 ，将密码信息释放返回结果

unset($res[0]['password']);

return jsonCode(200,$res[0]);

}else { // 无结果 返回错误码

return jsonCode(300,null);

}

}

?>

loginControll.php

登陆会请求登陆接口，查询用户表是否有此用户，查无用户就会返回{code：300，data：null}，如果有用户就会返回对应用户信息，信息存入本地缓存中。并进入系统公告页。



图5-1 登陆后显示默认公告页

根据返回的用户信息，判断当前用户权限，给予相应的操作。如以上为学生用户，针对学生管理，仅能进入当前用户的详情，并修改自己的信息。如图5-2所示。

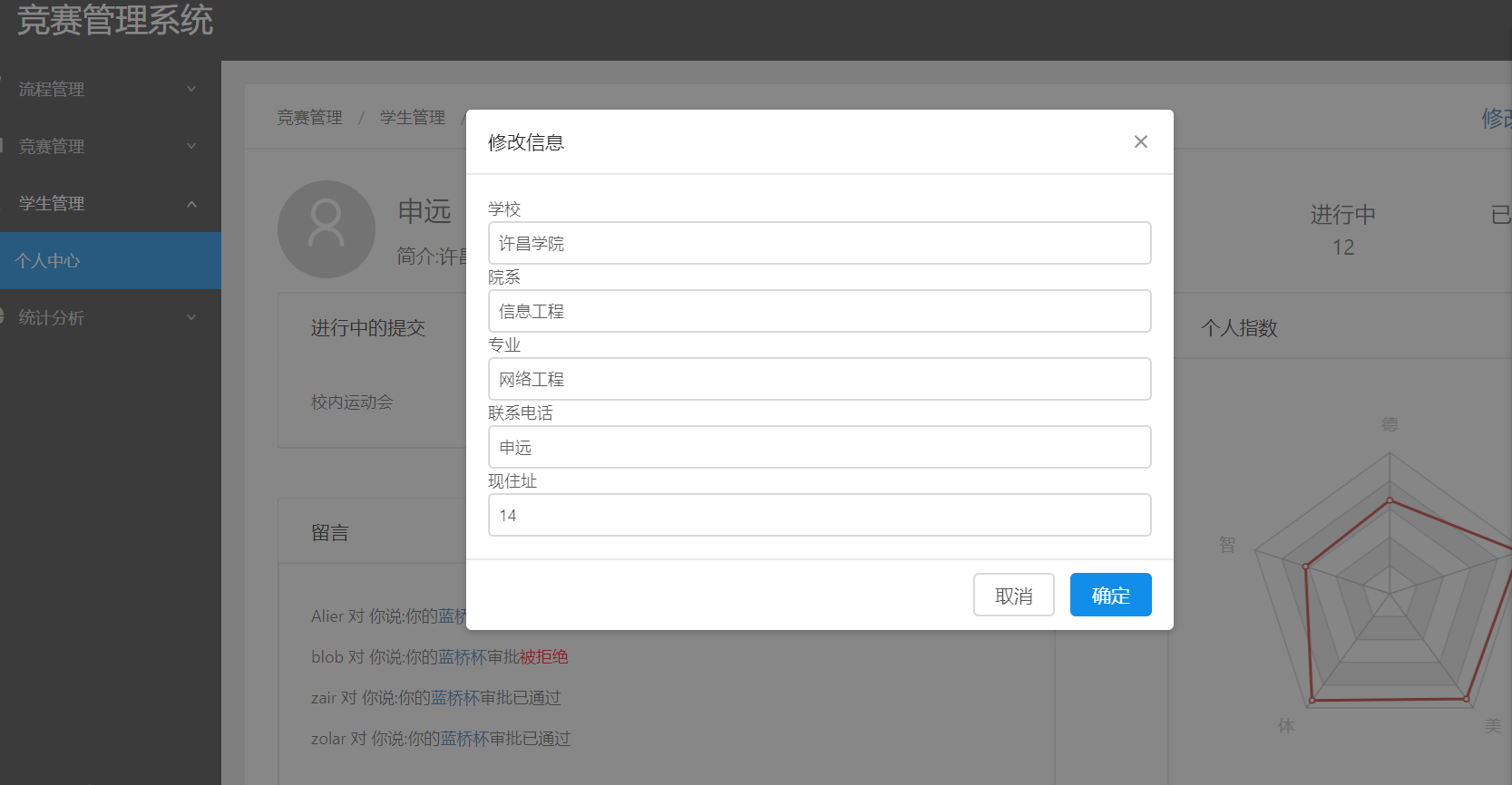


图5-2 个人中心页-修改信息

1. 修改个人信息功能-前端逻辑

member-more.Component.ts

update(){

this.modal = false; // 修改弹框关闭

this.\_api.updateMember(this.obj).then(e=>{

if (e.data) { // 修改成功后返回结果

this.\_notification.success('修改成功','');

this.getData() // 重新获取数据

}else{

this.\_notification.error('修改失败','');

}})

}

member-more.Component.ts

1. 修改个人信息功能-后端逻辑

memberUpdateControll.php

function getmore($info){

$pdo=new PdoMysql();

$res=$pdo->query("update`account`SET`school`='{$info->school}',`depart`='{$info->depart}',`profes`='{$info->profes}',`gard`='{$info->gard}' where `account\_id`={$info->account\_id}");

if ($res) {

return jsonCode(200,$res);

}else {

return jsonCode(300,null);

}

}

memberUpdateControll.php

学生用户针对竞赛信息没有任何权限，仅能查看列表，搜索，排序和查看详情。对于竞赛成绩审批，学生有添加的功能。如图5-3所示。

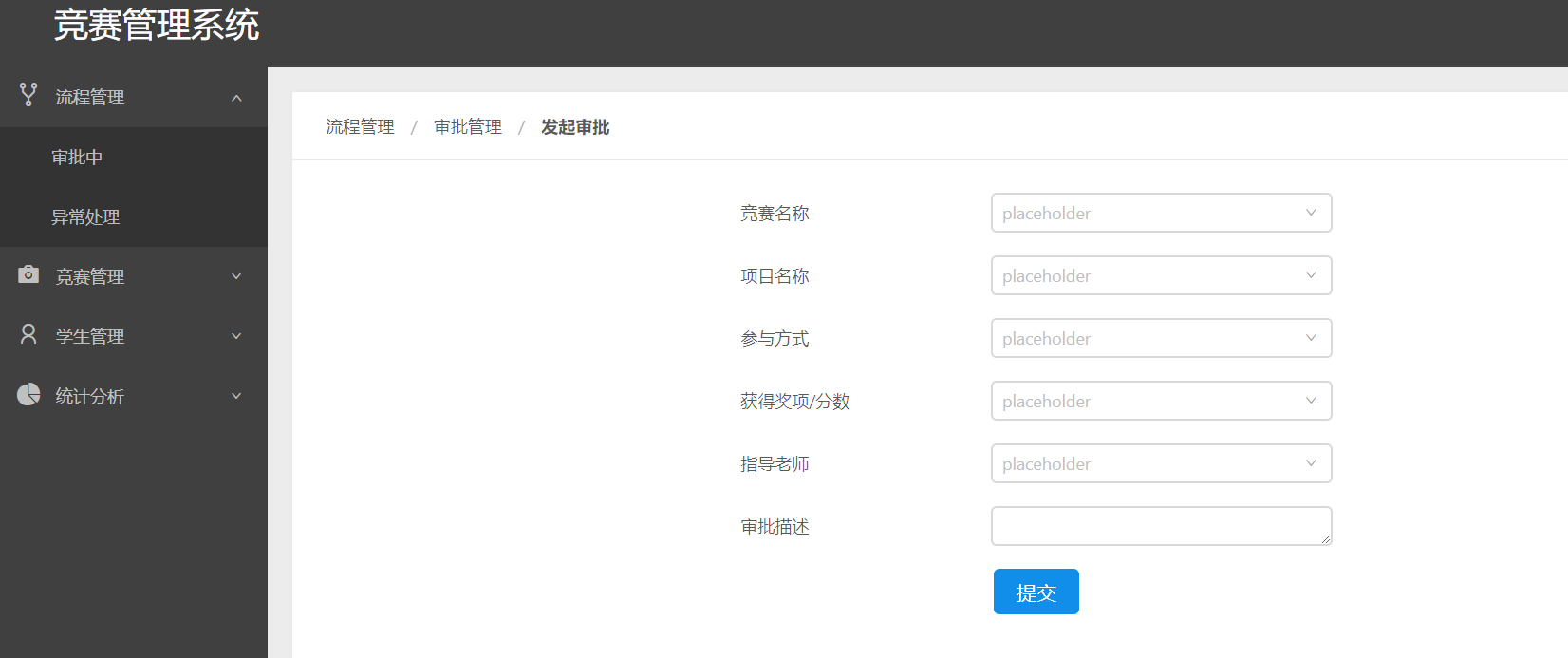


图5-3 流程管理-发起审批

1. 学生添加审批功能-前端逻辑

review.component.ts

put() {

this.obj['token'] = window.localStorage.getItem('token'); // 从缓存中获取用户信息

this.\_api.application(this.obj).then(e => {

if (e.data) { // 返回成功 显示弱提示

this.\_notification.success('添加成功', '申请老师会受到你的通知', { nzDuration: 4000 });

window.history.go(-1); // 页面回退至列表页

}

})

}

review.component.ts

1. 学生添加审批功能-后端逻辑

applicationControll.php

function add($obj)

{

$pdo=new PdoMysql();

$res = $pdo->query("INSERT INTO `application`(`user\_id`, `game\_name`, `item\_name`, `radiovalue`, `garde`, `status`,`teacher`, `desc`,`time`) VALUES ('{$obj->token}',{$obj->game\_id},{$obj->item\_id},{$obj->radiovalue},'{$obj->garde}','0','{$obj->teacher}','{$obj->desc}',{$obj->time})");

return jsonCode(200,$res);

}

applicationControll.php

## 5.2教师用户

教师用户针对流程管理，有进行审批的功能，包括通过审批和拒绝审批。还有向我提交的数据过滤。用于教师一键过滤向本人提交的数据，以便快速审核。如图5-4所示

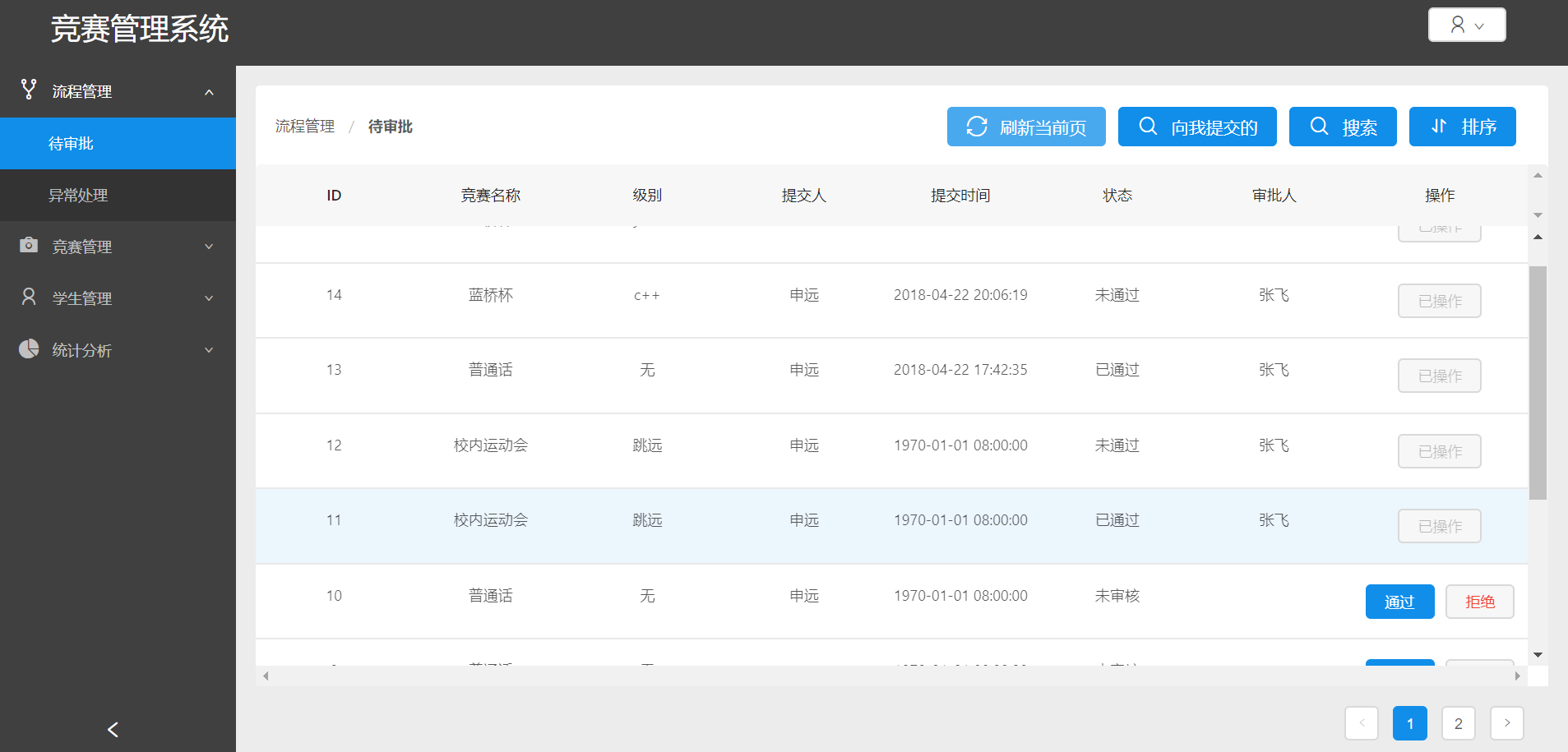


图5-4 流程管理-待审批

1. 流程管理-待审批-前端逻辑

primit.component.ts

primt(i,status){ // 审核通过逻辑

if (i.teacher!=window.localStorage.getItem('token')) {

this.\_notification.error('操作失败', '当前用户非指定教师',{ nzDuration:4000});

return;

}

this.\_api.review(i.app\_id,status).then(e=>{

if (e.data) {

this.\_notification.success('操作成功', '稍后会通知该学生', { nzDuration:4000});

this.getData();

if (status==-1) {

return

}

this.\_api.aduited(i['teacher'],i['user\_id'], i['app\_id'], 1, '');

}else{

this.\_notification.error('操作失败', '', { nzDuration: 4000 });

}

})

}

reject(i){

this.rejectObj = i;

this.searchbox = true;

}

handleOk(e){ / 审核拒绝逻辑

this.searchbox = false;

this.primt(this.rejectObj,-1);

this.\_api.aduited(this.rejectObj['teacher'],this.rejectObj['user\_id'],this.rejectObj['app\_id'],-1,e).then(e=>{

})

}

primit.component.ts

1. 流程管理-待审批-后端逻辑

reviewControll.php

function add($obj){

$pdo=new PdoMysql();

$res = $pdo->query("UPDATE `application` SET `status`={$obj->status} WHERE `app\_id` = {$obj->appid}");

if ($res!=null) {

// unset($res[0]['password']);

return jsonCode(200,$res);

}else {

return jsonCode(300,null);

}

}

reviewControll.php

教师用户针对竞赛信息有添加和修改的功能。能够进行模糊查询和排序。如图5-5所示。

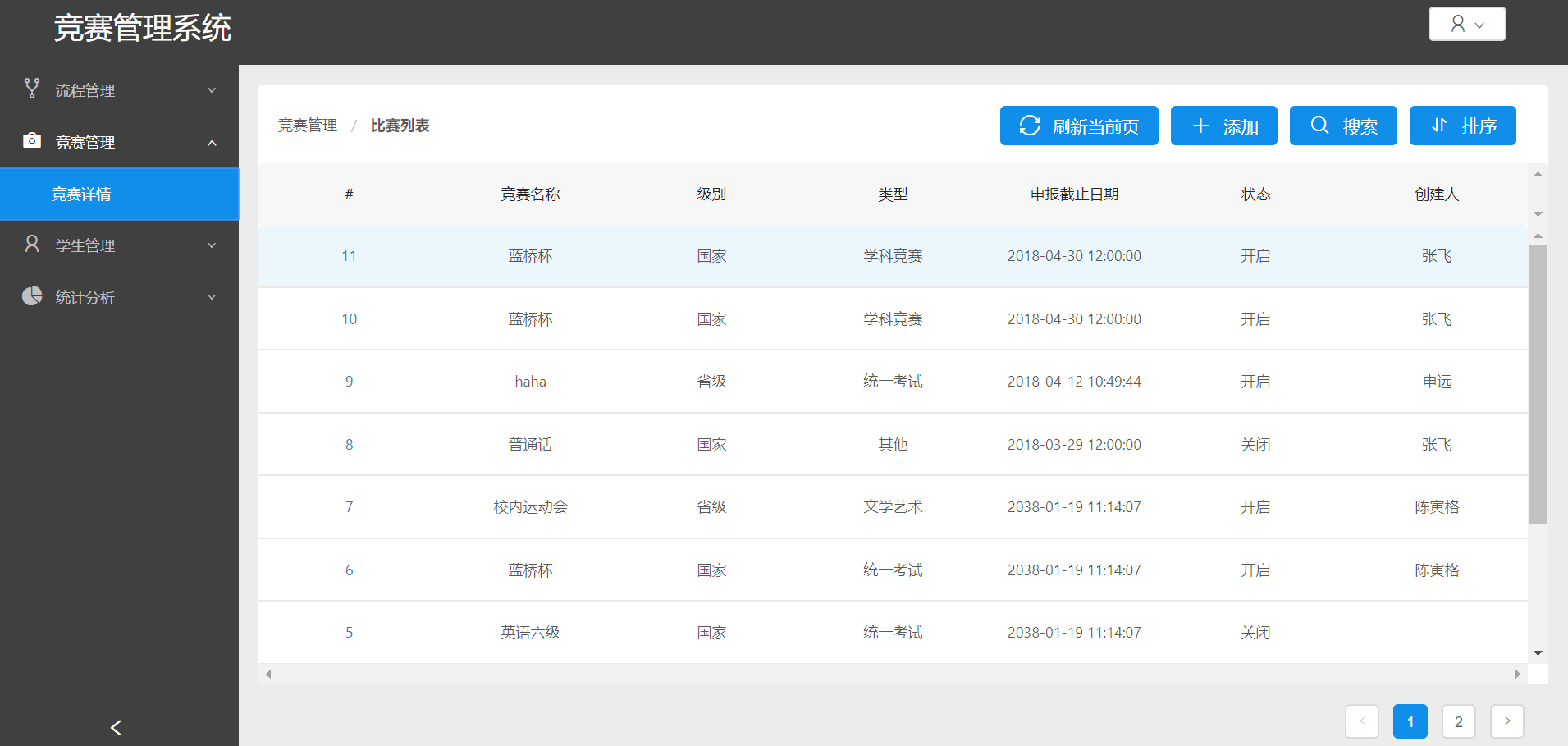


图5-5 竞赛管理-列表页

1. 竞赛列表页刷新、排序、搜索功能-前端逻辑

contestlistList.component.ts

getData() {

this.loading = true; // 开始获取数据 ，加载动画

this.\_api.contestList(this.Sorting, this.searchObj).then(e => {

this.memberList = e.data; // http请求返回数据

this.TotalRecordCount = this.memberList.length; // 记录总条数

})

this.loading = false; // http获取到结果 加载动画结束

}

reload() {

this.searchObj = null; // 将搜索条件置空，在获取数据

this.getData();

}

returnSortChecked(e) {

this.Sorting = `${e.key} ${e.value == 0 ? 'ASC' : 'DESC'}`; // 根据传入的值，判断升序或降序

this.getData();

}

returnSearchChecked(e) {

this.searchObj = e; // 根据传入的值，设置搜索条件，获取对应数据

this.getData();

}

contestlistList.component.ts

1. 竞赛列表页排序、搜索功能-后端逻辑

contestlistListControll.php

function memberlist($Info){

$sql="";

$pdo=new PdoMysql(); // 实例化数据操作类

foreach ($Info as $key => $value) {

if ($key!='sort'&&$value) { // 进行查询语句拼接

$sql.=" and ".$key." like '%".$value."%'";

}

}

$res = $pdo->getAll("SELECT \* FROM `account` WHERE auth=1".$sql." ORDER BY {$Info->sort}"); // 查询数据库 返回所得结果

if ($res!=null) {

return jsonCode(200,$res);

}else {

return jsonCode(300,null);

}

}

contestlistListControll.php

教师用户针对用户管理同样有排序，搜索，功能，原理同上，仅是针对的api不同，另外教师用户具有下载用户表的功能。如图5-6所示。为了优化使用效果，此处使用phpexcle这个类库进行。此处需要注意的是，再试用接口类型下载数据时，由于数据的特殊性（二进制数据流），需要将之转换为Excel类型，故而需要单独设置http报文的响应类型为Blob（默认为text）

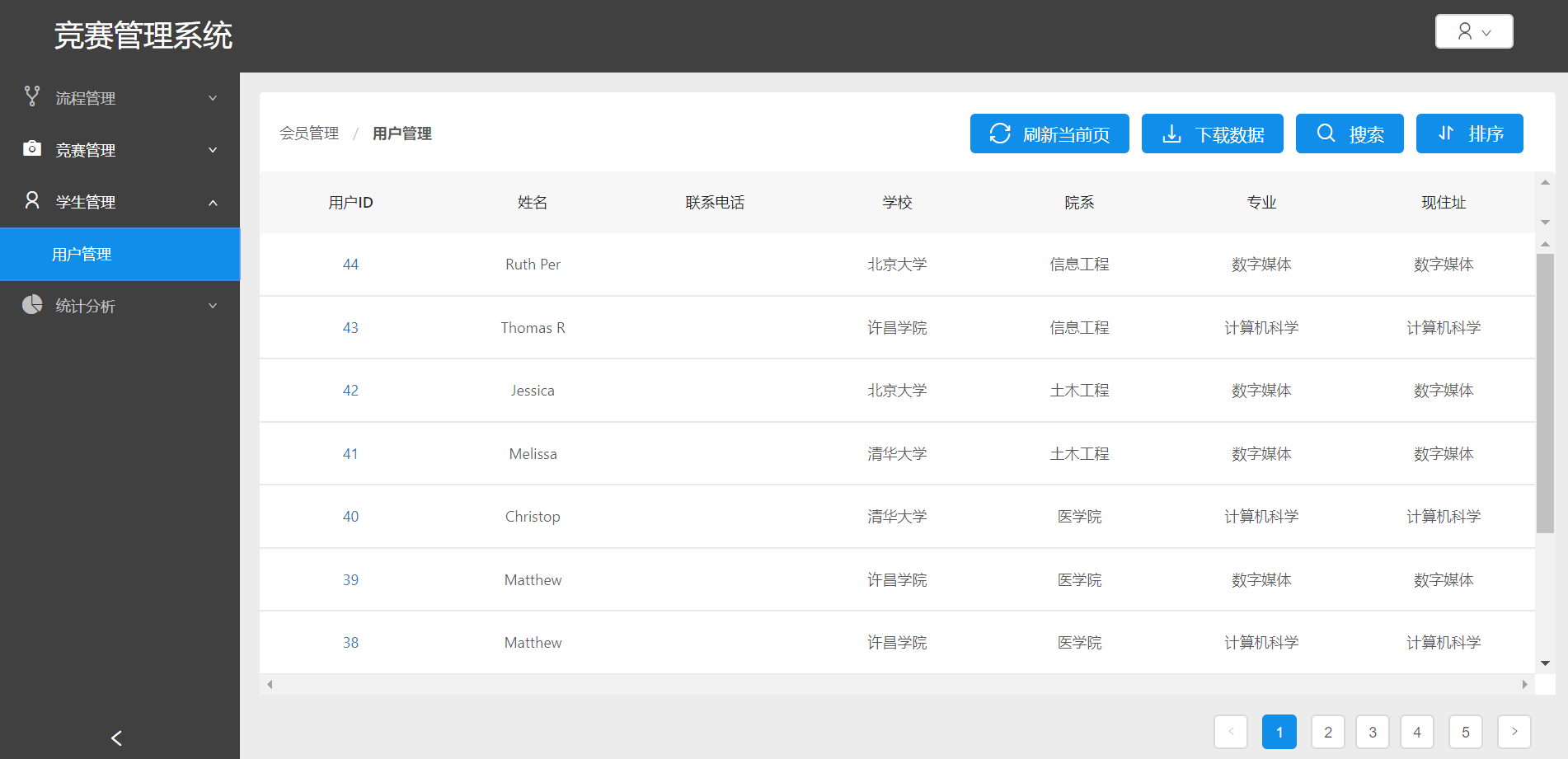


图5-6 用户管理-列表页

1. 用户管理下载功能-前端逻辑

api.service.ts

public async exportexcl() {

var url = 'http://localhost/scms/api/export.php';

this.\_http(url).subscribe(res => {

var blob = new Blob([res.json()], { type: "application/vnd.ms-excel" });

var objectUrl = URL.createObjectURL(blob);

var a = document.createElement('a');

document.body.appendChild(a);

a.setAttribute('style', 'display:none');

a.setAttribute('href', objectUrl);

a.setAttribute('download', 'user');

a.click();

document.body.removeChild(a);

//释放URL地址

URL.revokeObjectURL(objectUrl);

});

}

api.service.ts

1. 用户管理下载功能-后端逻辑

excelControll.php

function exportExcl()

{

$pdo=new PdoMysql();

$res = $pdo->getAll("SELECT \* FROM `account` WHERE 1");

$indexKey = array('account','password','createtime');

$hear = array('account'=>'账号','password'=>'密码','createtime'=>'创建时间');

array\_unshift($res,$hear);

exportExcel($res,'user',$indexKey,$startRow=1,$excel2007=false);

}

excelControll.php

以上代码便是实现将二进制大对象（blob）转换成为Excel表，并创建一个a标签，链接为文件的地址。以此实现数据导出下载。

## 5.3公共模块

公共模块是教师与学生两种角色共有的一些功能，例如密码的修改，数据分析管理页面等。修改密码功能如图5-7所示

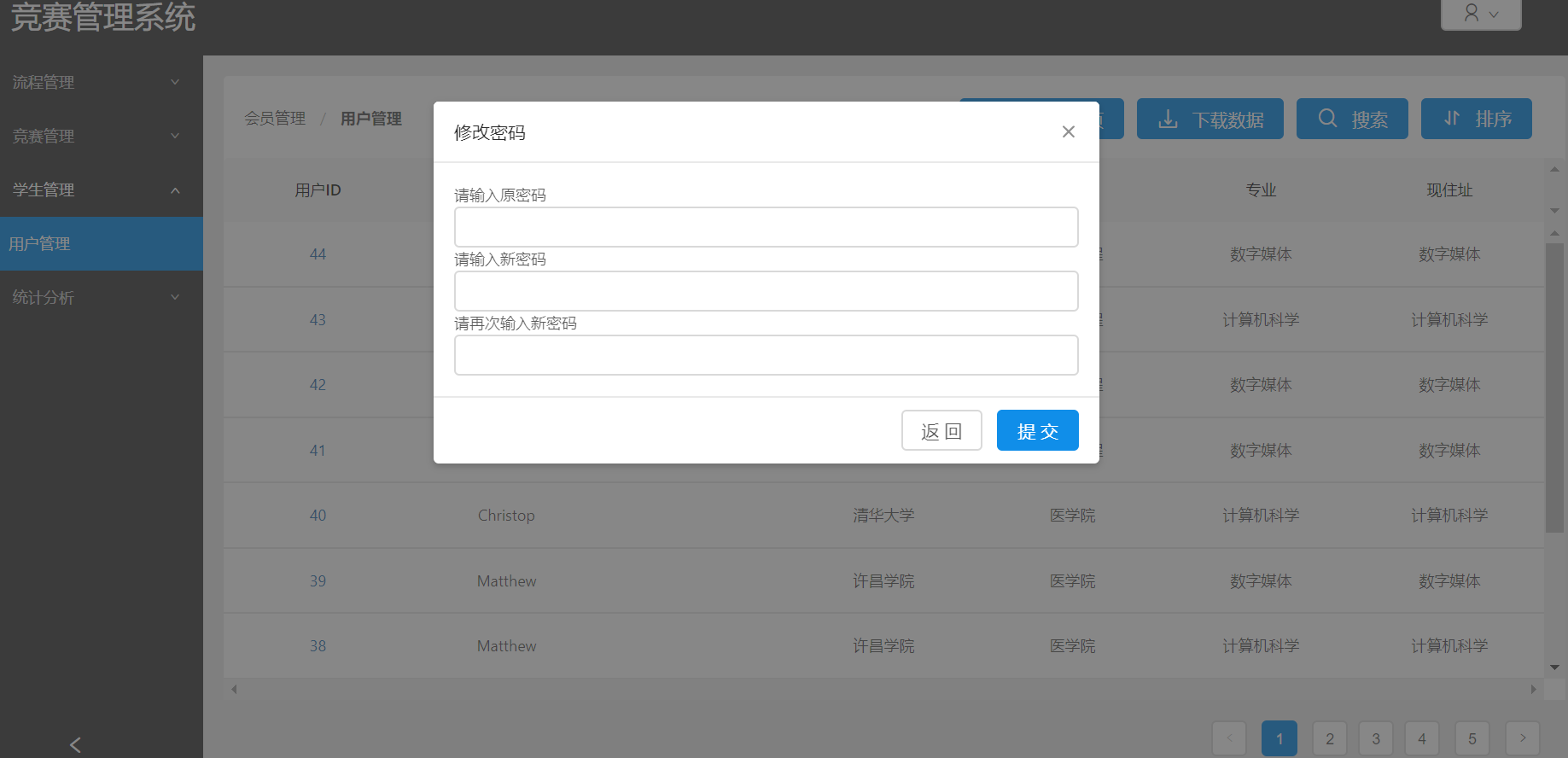


图5-7 修改密码

1. 用户修改密码功能-前端逻辑

index.component.ts

updata(old, new\_pad, new\_again) {

if (new\_pad != new\_again) {

return;

}

this.\_api.updataPwd(old, new\_again).then(e => {

this.isVisible = false;

if (e.data) {

this.\_notification.success('密码修改成功', '再次登陆启用新密码', { nzDuration: 4000 });

} else {

this.\_notification.error('密码修改失败', '请检查原密码是否正确', { nzDuration: 4000 });

}

})

}

index.component.ts

1. 用户修改密码功能-后端逻辑

updataPasswordControll.php

function updata($oldPwd,$newPwd){

$pdo=new PdoMysql();

$res = $pdo->getAll("SELECT \* FROM `account` WHERE `id`='2' and `password`='{$oldPwd}' ");

if ($res!=null) {

$res2 = $pdo->query("UPDATE `account` SET `password`='{$newPwd}' WHERE id='2'");

return jsonCode(200,$res2);

}else {

return jsonCode(300,null);

}

}

updataPasswordControll.php

# 6．单元测试

测试工作是软件开发至关重要的一个环节。技术人员在编码过程中或多或少会遗漏或错误理解逻辑处理。因此在实际开发工作中，会有专门的的测试工程师采用不同的技术对软件进行功能测试（含可使用性测试）、性能测试、安全测试和回归测试，在软件正式发布之前，尽可能的发现问题并提交给技术人员修改。避免给用户和自身造成重大损失。

（1）功能测试

检查被测软件的功能是否正常实现；

视觉效果是否符合UI设计图（此处系统没有UI设计图）；

（2）性能测试

系统运行占用的资源，完成某一步骤需要的时间；

系统能承受的压力,针对并发操作的处理；

（3）安全性测试：

根据需求说明检查系统是否达到安全性要求(同一用户登陆不同机器，同时操作对数据；

写到配置文件或数据库的密码是否经过加密；

（4）回归测试：

验证Bug是否修正；

Bug修正后是否影响其他功能的正常运行。

当然，以上仅是测试工程师要做的内容，而技术人员在编码过程中也是要进行自测的。毕竟，写出高质量，易维护，易阅读的代码是每个程序员所追求的。那么针对代码层面的测试，就需要进行单元测试。所谓的单元测试，就是用来对一个模块、一个函数或者一个类来进行正确性检验的测试工作。最终生成一个测试的覆盖率报告。可以这样说，通过覆盖率报告就可以体现程序员的技术水平。

此系统采用karma+jasmine进行单元测试。

（1）Karma简介

Karma是一个基于Node.js的JavaScript测试执行过程管理工具。该工具可用于测试所有主流Web浏览器，，也可和其他代码编辑器一起使用。这个测试工具的一个强大特性就是，它可以监控(Watch)文件的变化，然后自行执行。

（2）Jasmine简介

Jasmine （茉莉）是一款 JavaScript BDD（行为驱动开发）测试框架，它不依赖于其他任何 JavaScript 组件。它有干净清晰的语法，可以很容易的写出测试代码。对基于 JavaScript 的开发来说，它是一款不错的测试框架选择。

## 6.1编写单元测试脚本

Karma的脚本语法是以js为基础进行封装的，使用方便易懂。测试demo如图6-1~6-3所示

Describe(string,fun) 表示一组测试实例 也就是一组测试用例

It(string,fun)表示单个测试用例

Expect表示期望,这个表达式表示某个值或某种行为

To-这个函数表示匹配 图中的toEqual()用来比较简单的文本和变量，还有对象是否相等，例如以下（包括但不限于）的语法，

toBe()： 用来比较数字或者字符串是否相等，不支持对象的比较

toBeDefined（）：对未定义进行判断，如果定义了则为true

toBeUndefined（）：对未定义进行判断，如果没有定义则为true

toContain（）：判断字符串或者数组中是否包含某个值，包含则为true

toThrow（）：抛出异常时为true

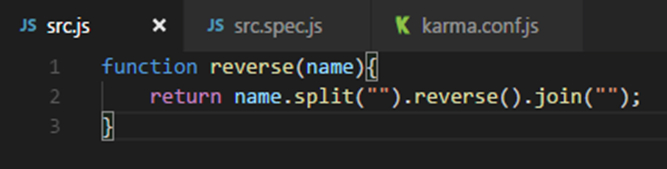


图6-1 测试函数

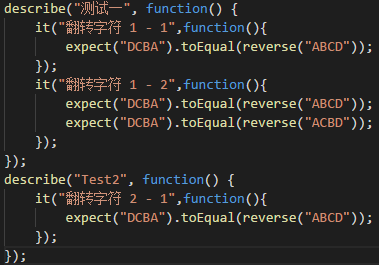


图6-2 测试用例

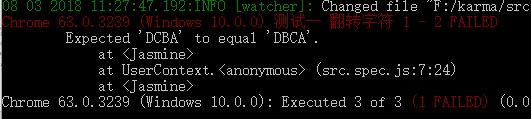


图6-3 执行结果

以上demo仅仅针对reverse函数进行脚本测试，可以看到如果期望值与函数实际执行的值不同就会提示具体的测试用例，反馈给技术人员具体什么样的逻辑导致函数返回值出错。这样就能极大的减少出现问题的概率。由于每个组件的测试脚本都要单独写，下面只写其中app组件的测试脚本

（1） app模块测试用例

app.component.spec.ts

import { TestBed, async } from '@angular/core/testing';

import { AppComponent } from './app.component';

describe('AppComponent', () => {

beforeEach(async(() => {

TestBed.configureTestingModule({

declarations: [

AppComponent

],

}).compileComponents();

}));

it('should create the app', async(() => {

const fixture = TestBed.createComponent(AppComponent);

const app = fixture.debugElement.componentInstance;

expect(app).toBeTruthy();

}));

it(`should have as title 'app'`, async(() => {

const fixture = TestBed.createComponent(AppComponent);

const app = fixture.debugElement.componentInstance;

expect(app.title).toEqual('app');

}));

it('should render title in a h1 tag', async(() => {

const fixture = TestBed.createComponent(AppComponent);

fixture.detectChanges();

const compiled = fixture.debugElement.nativeElement;

expect(compiled.querySelector('h1').textContent).toContain('Welcome to app!');

}));

});

app.component.spec.ts

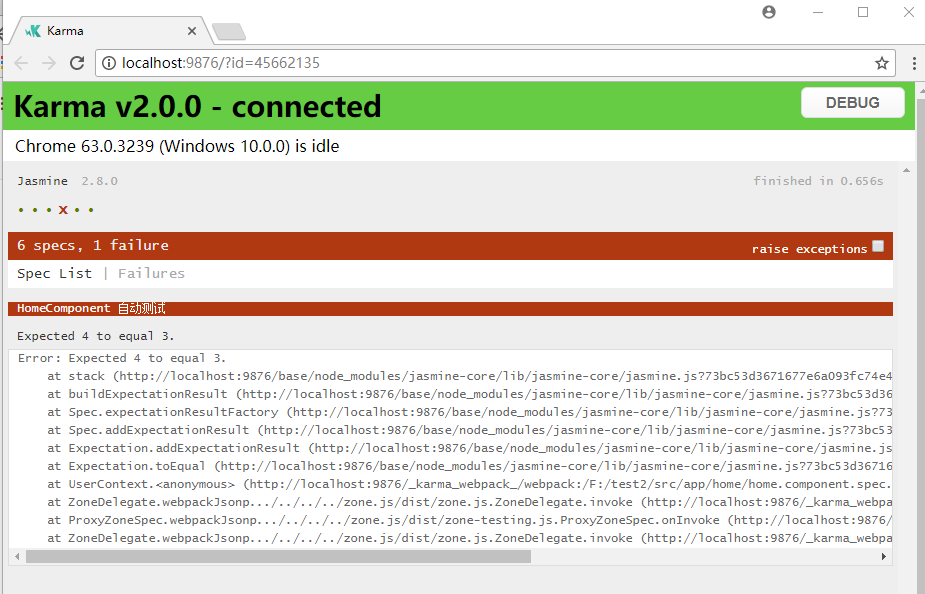
执行以上测试用例，就会打开一个服务，用于展示图形化测试结果。如图6-4所示

图6-4 app.module测试结果

## 6.2覆盖率报告

经过测试脚本检测后，可以导出相应的覆盖率报告，用来表明该测试的覆盖范围，这份报告在一定程度上代表了技术人员的代码水平。将此次脚本的结果导出，结果如图6-5所示

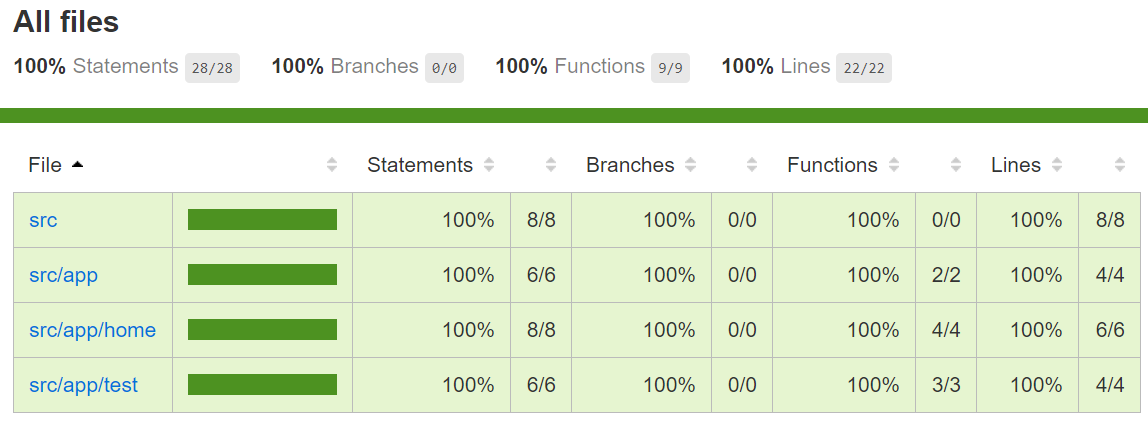


图6-5 app.module覆盖率报告

以上图所示，报告中需要注意的四个参数，即

行覆盖率（line coverage）：是否每一行都执行了？

函数覆盖率（function coverage）：是否每个函数都调用了？

分支覆盖率（branch coverage）：是否每个逻辑处理代码块都执行了？

语句覆盖率（statement coverage）：是否每个语句都执行了？

由于此组件逻辑控制并不复杂，故而经所有条件都测试了一遍，覆盖率为100%；

## 6.3功能测试

（1）教师用户添加竞赛信息，操作如图6-6~6-7。执行添加操作，结果如图6-8~6-9所示

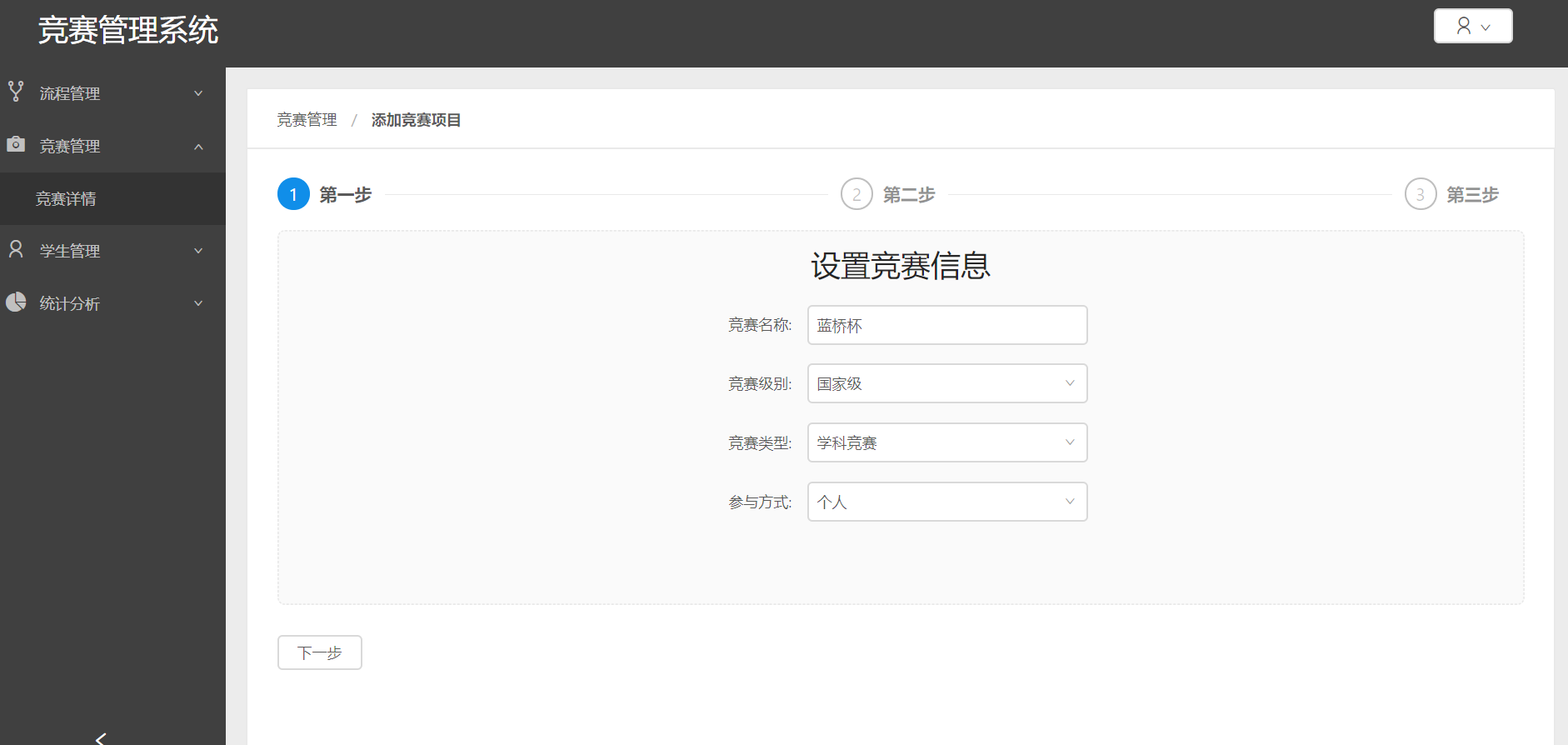


图6-6 添加竞赛信息-第一步

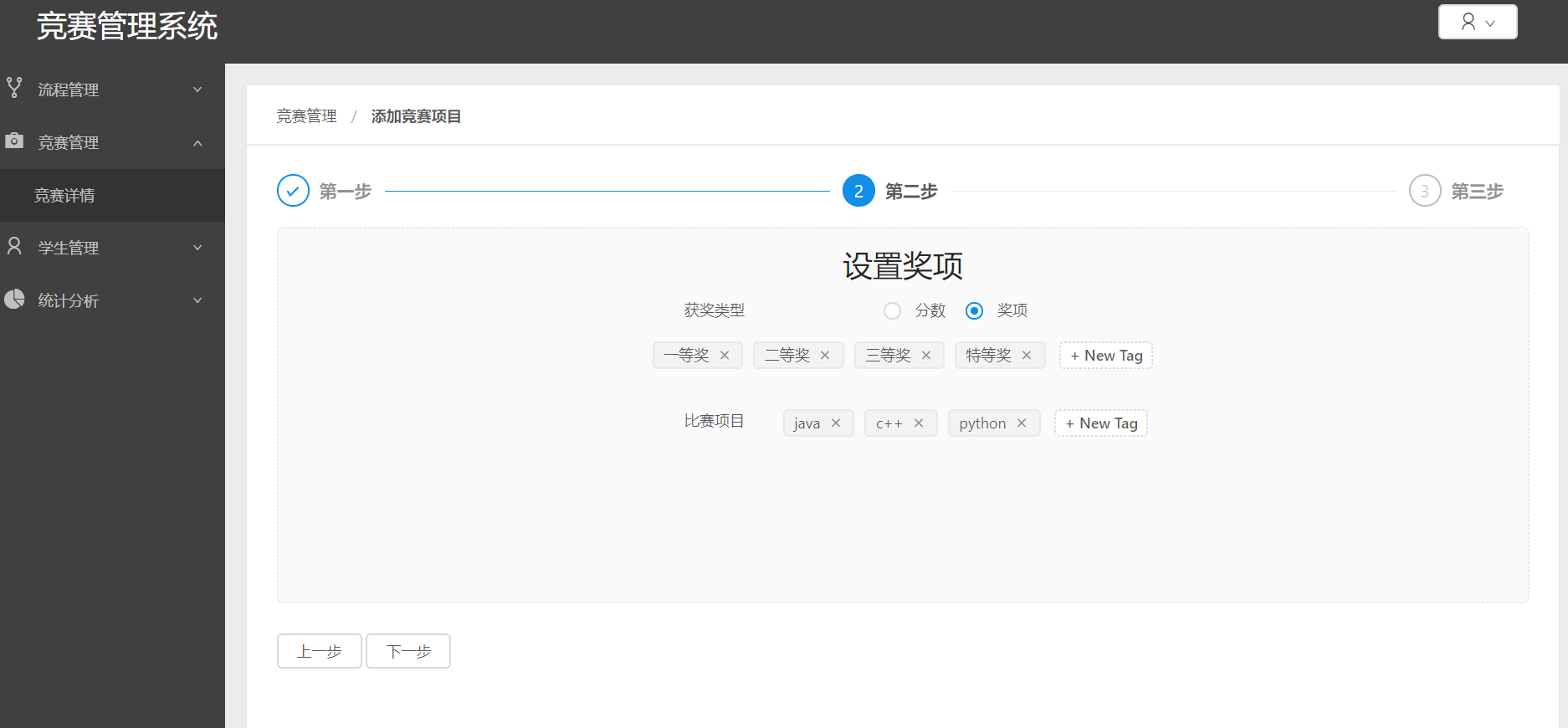
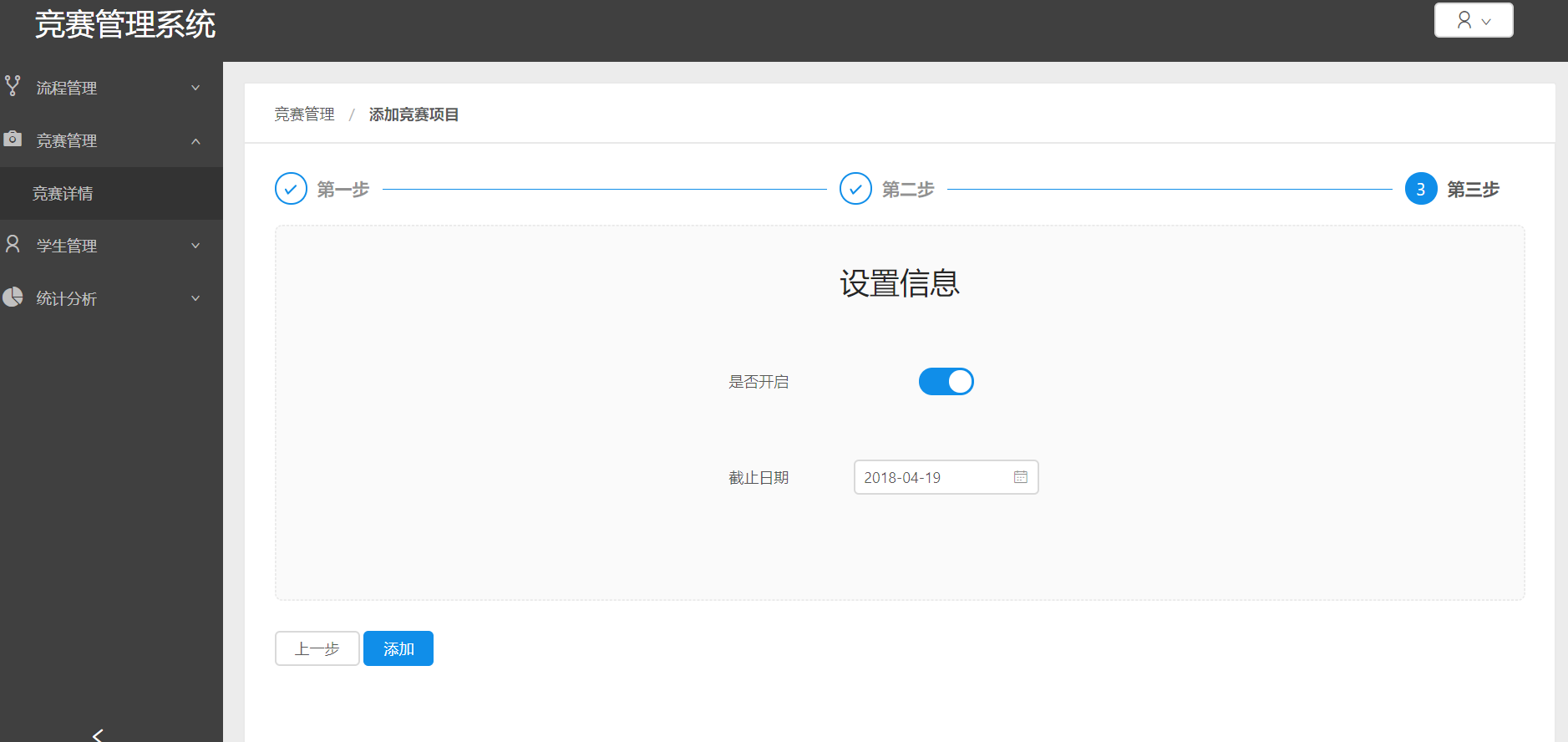


图6-6 添加竞赛信息-第二步



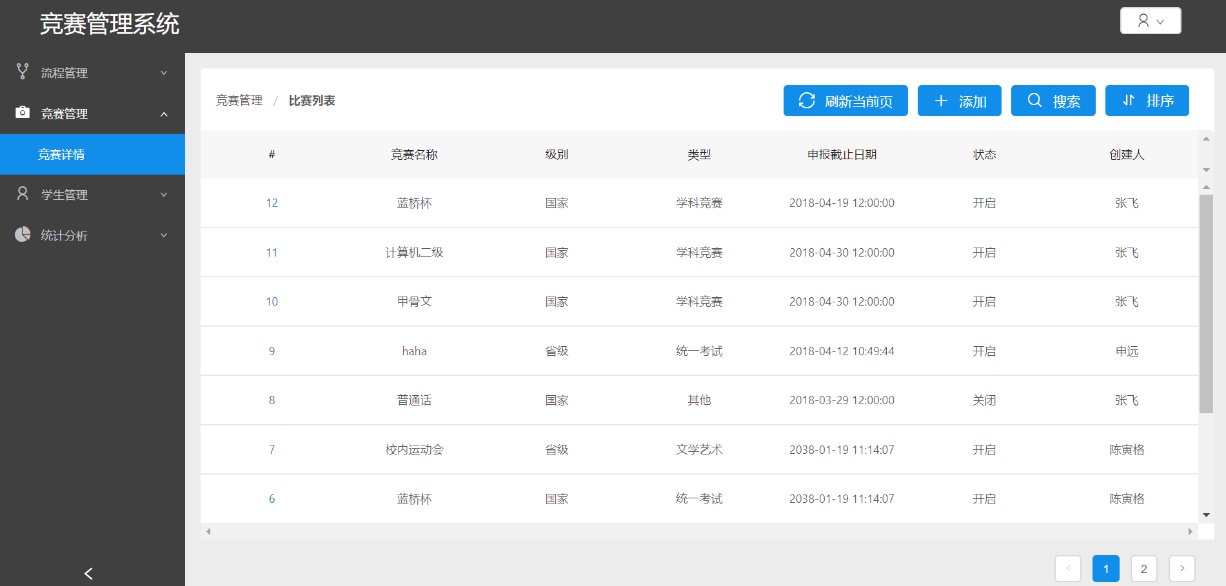
图6-7添加竞赛信息-第三步

图6-8添加竞赛结果-列表第一条

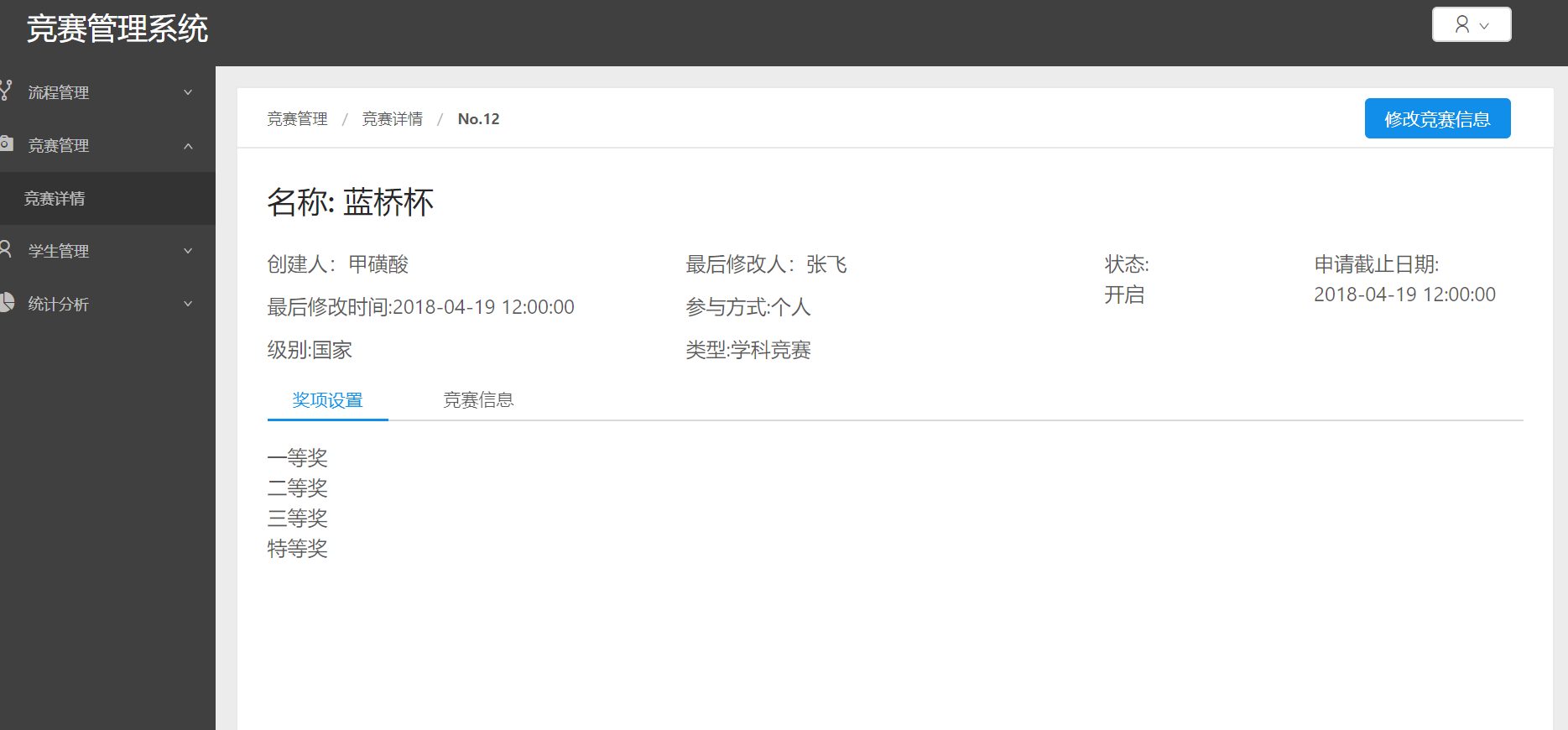


图6-9添加竞赛结果-竞赛详情

（2）学生用户发起审批，操作如图6-10所示，结果如图6-11所示

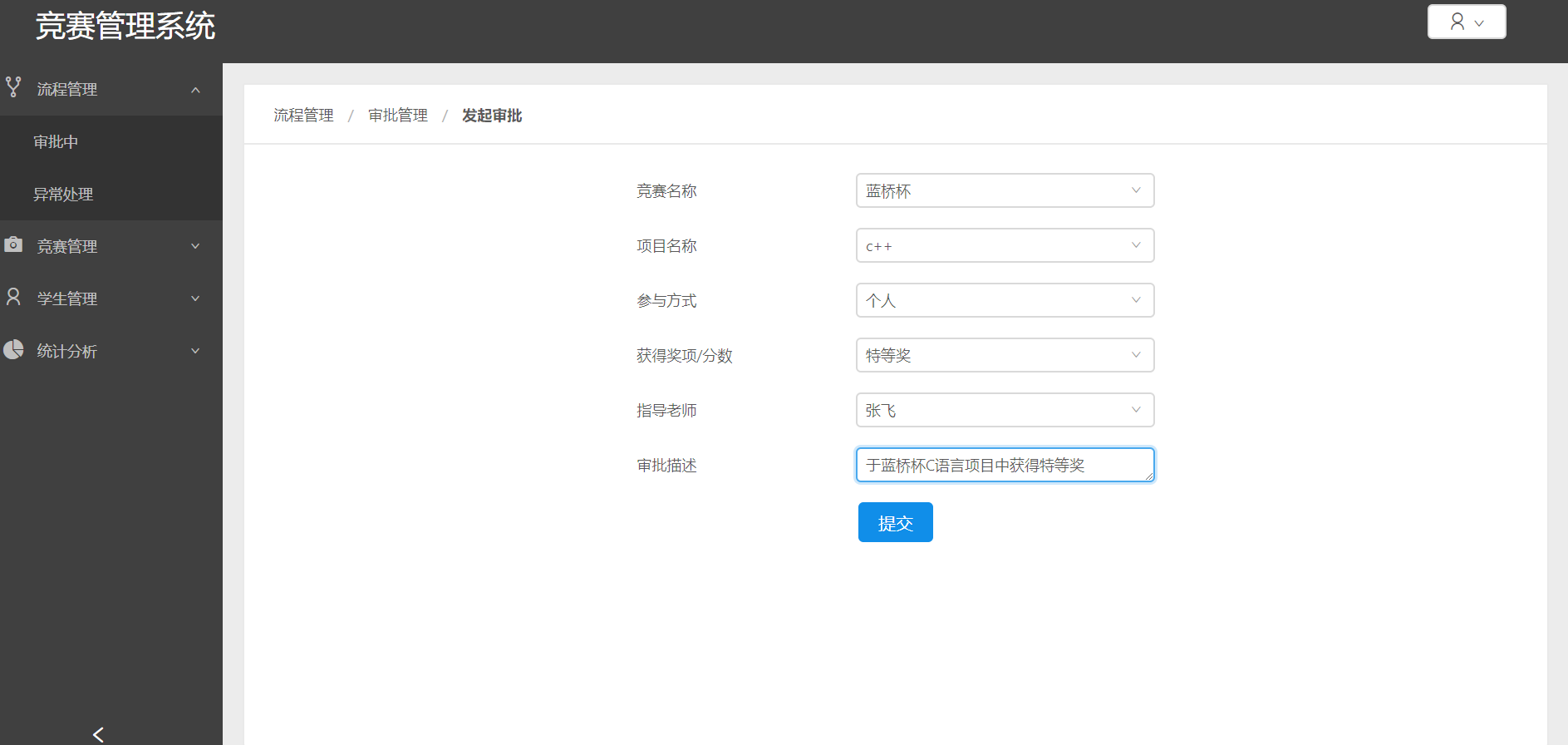


图6-10发起审批流程

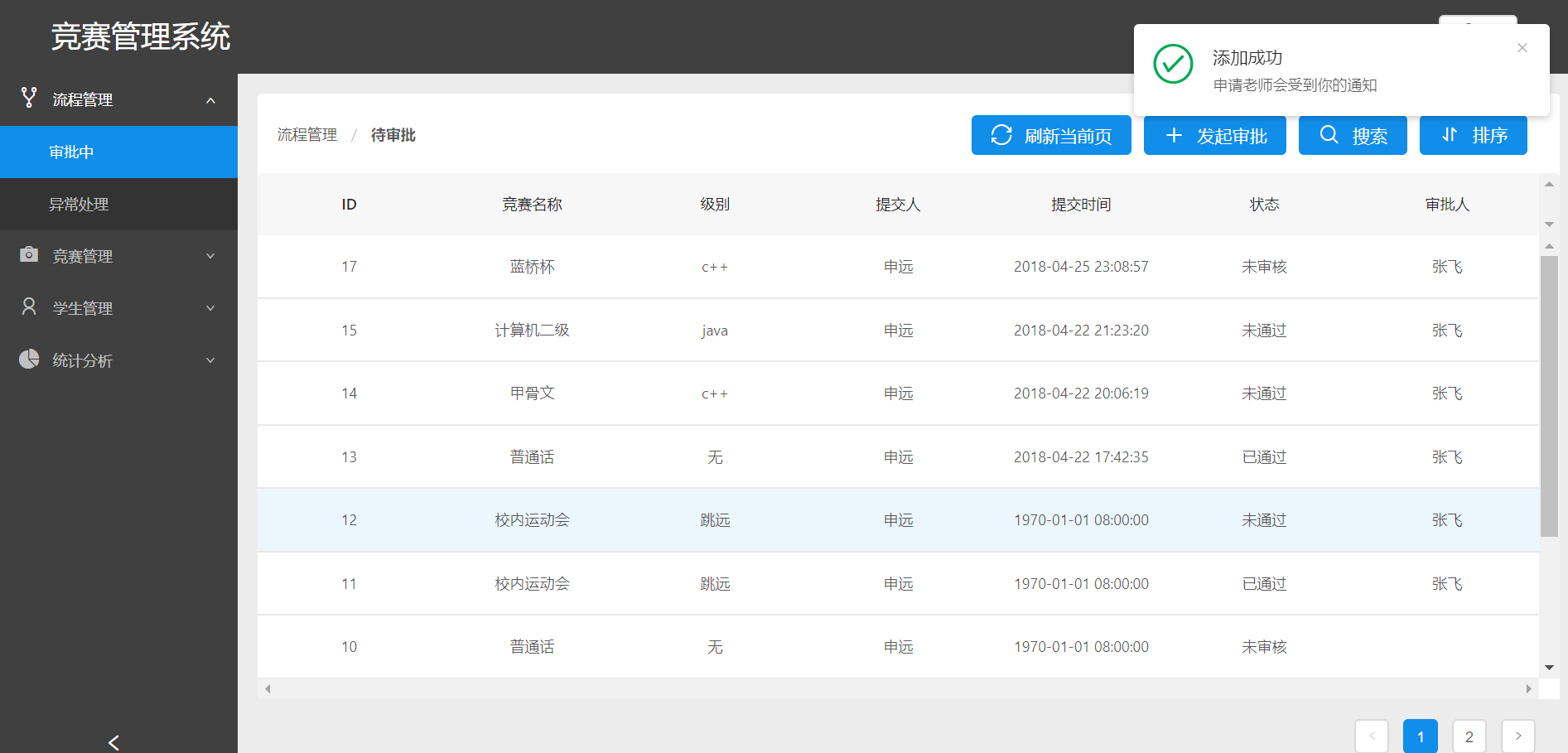
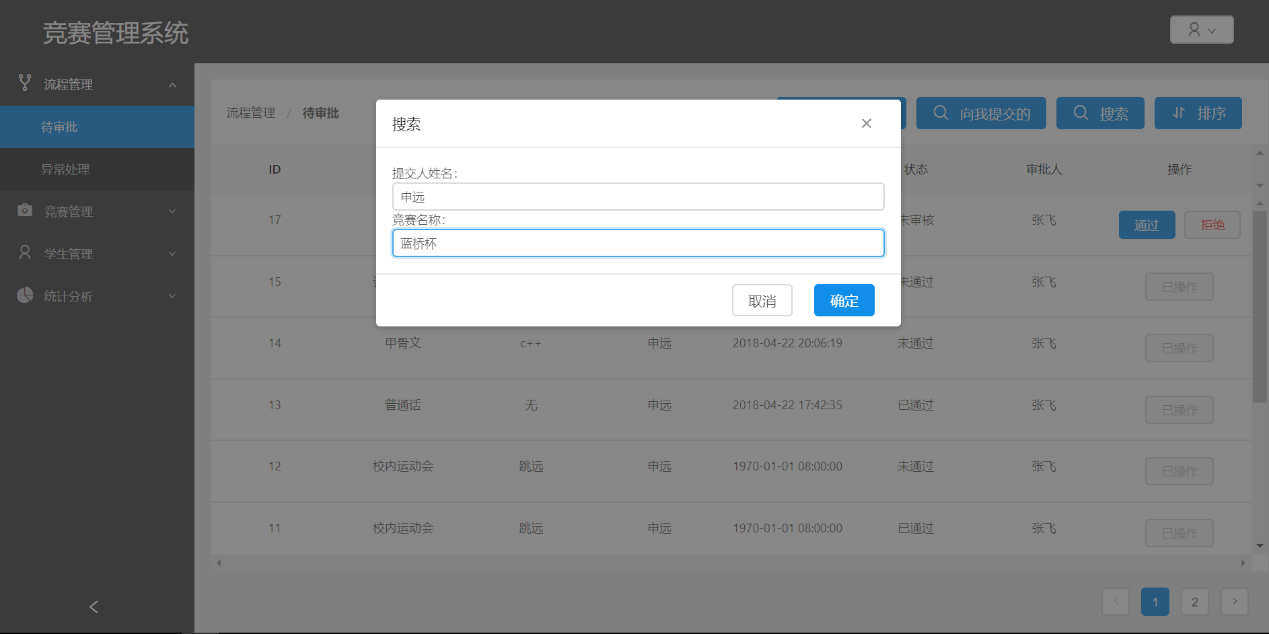


图6-11发起审批流程-成功

（4）教师针对待审批列表进行搜索，排序，以及审核审批内容。如图6-11~6-15所示

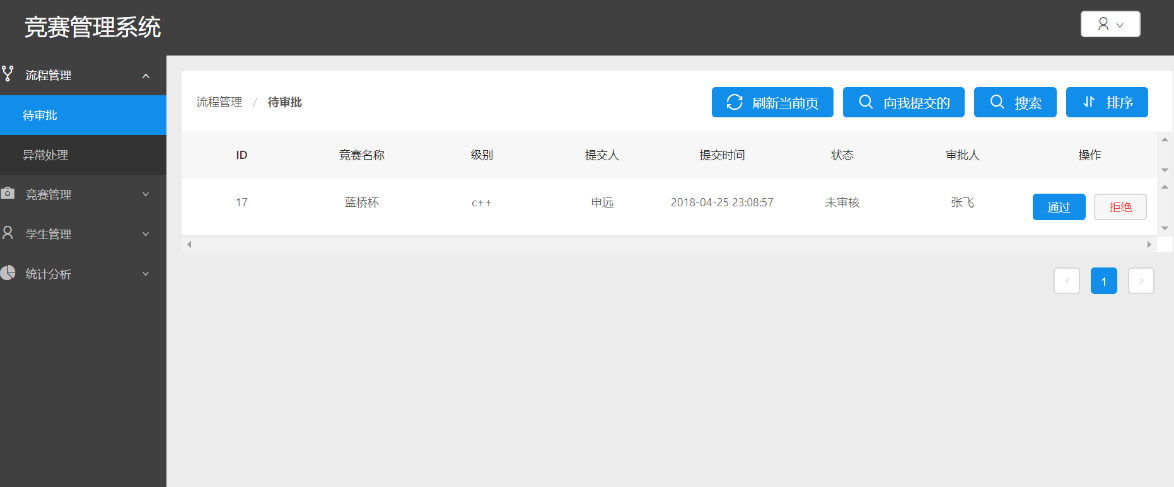
图6-11待审批流程-搜索操作

图6-12待审批流程-搜索结果

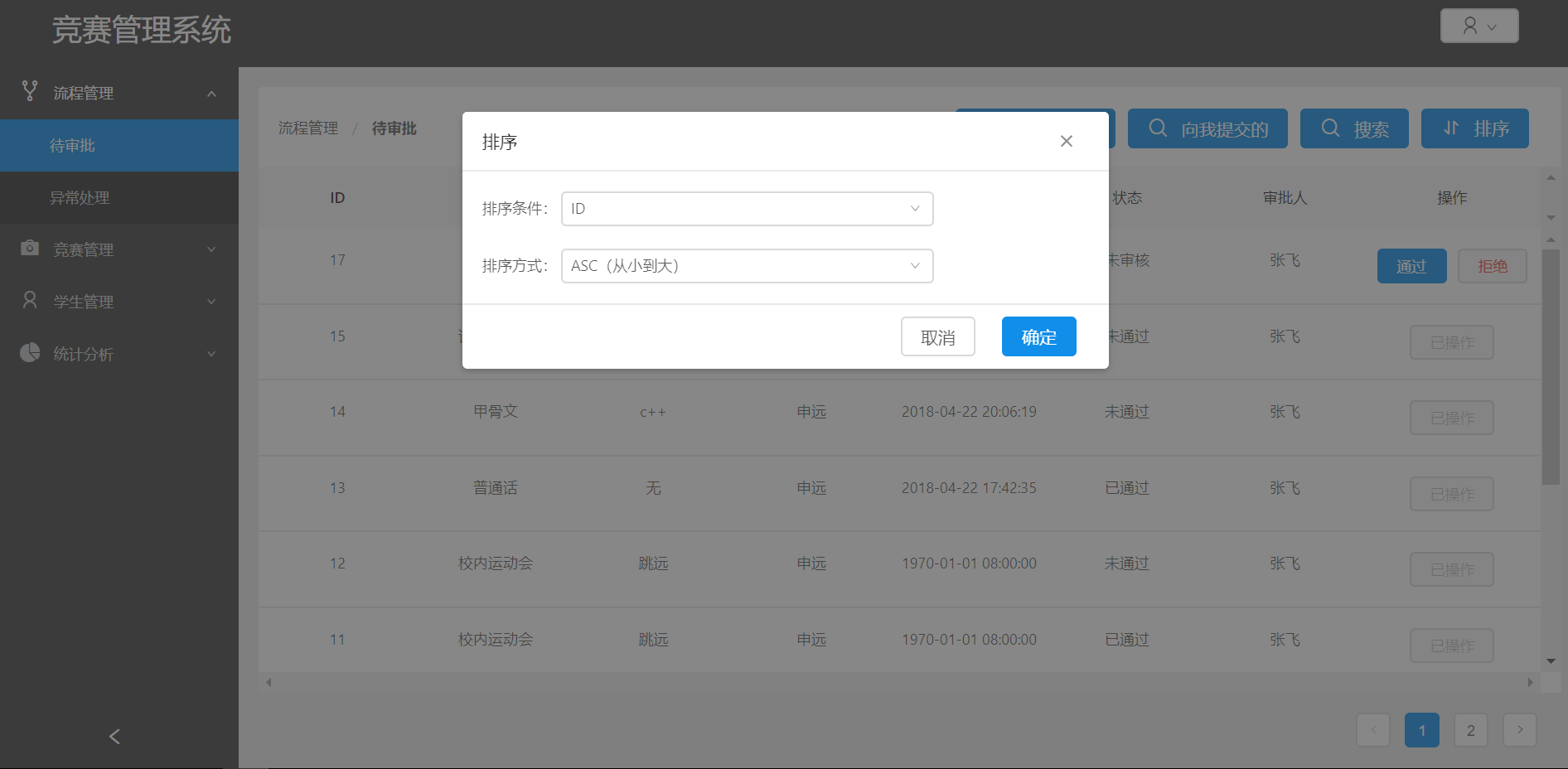


图6-13待审批流程-排序操作

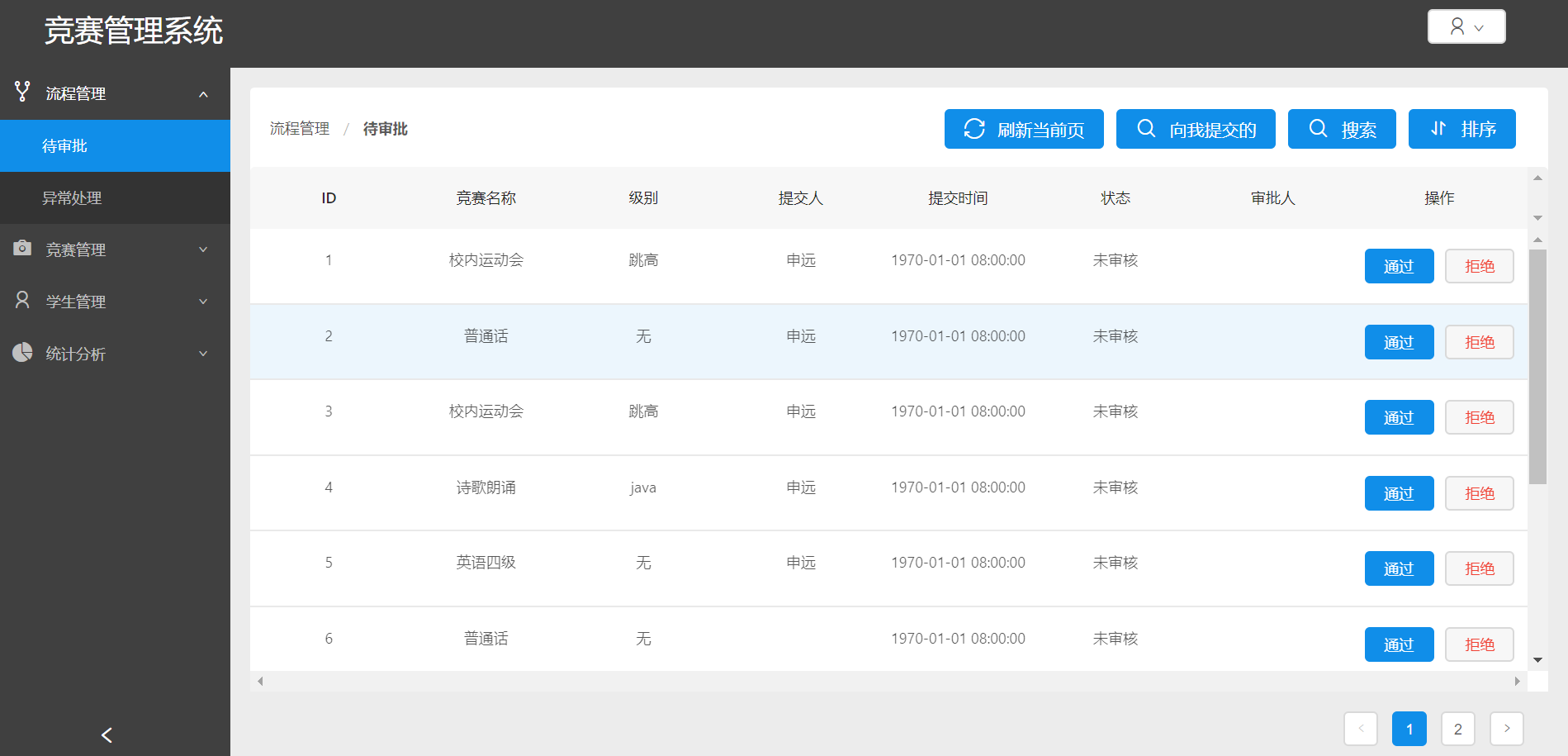


图6-13待审批流程-排序结果

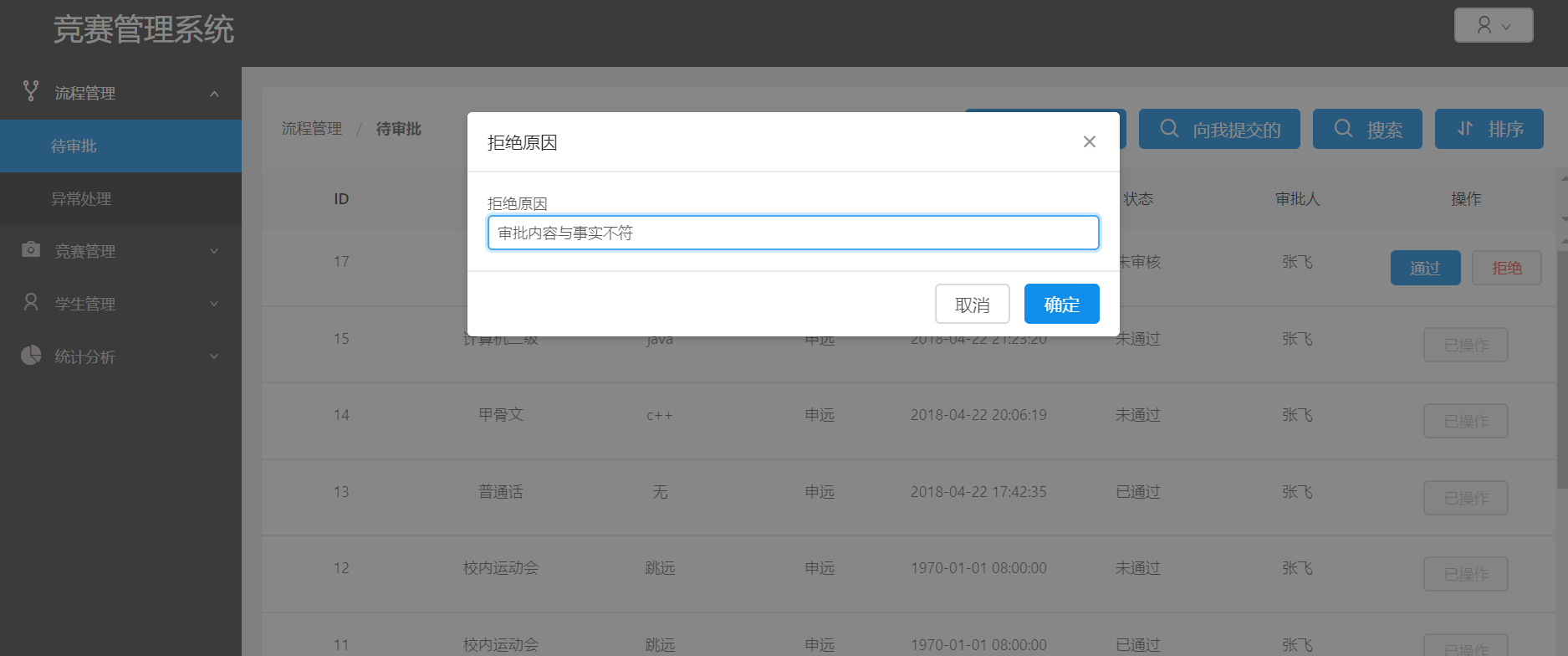


图6-14待审批流程-拒绝

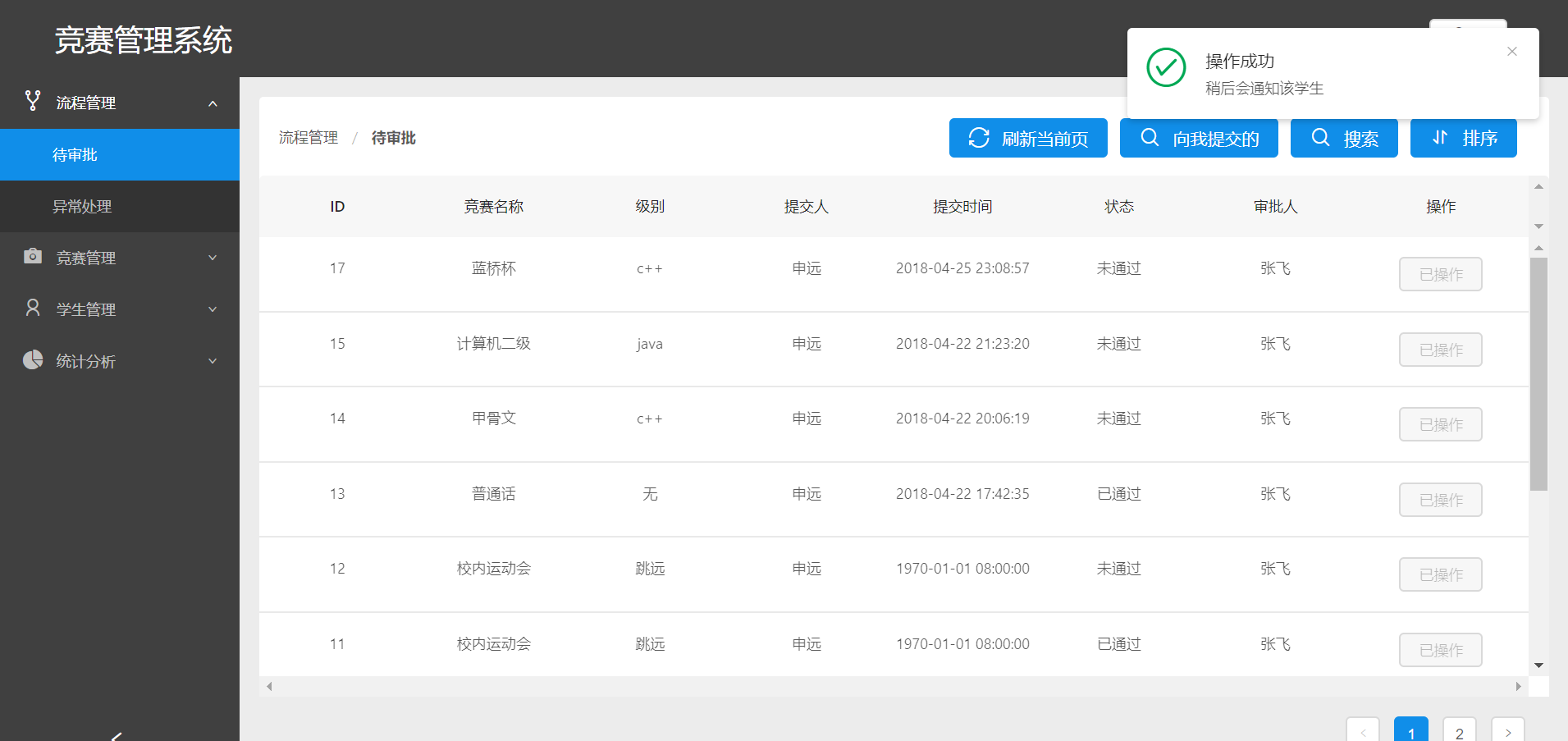
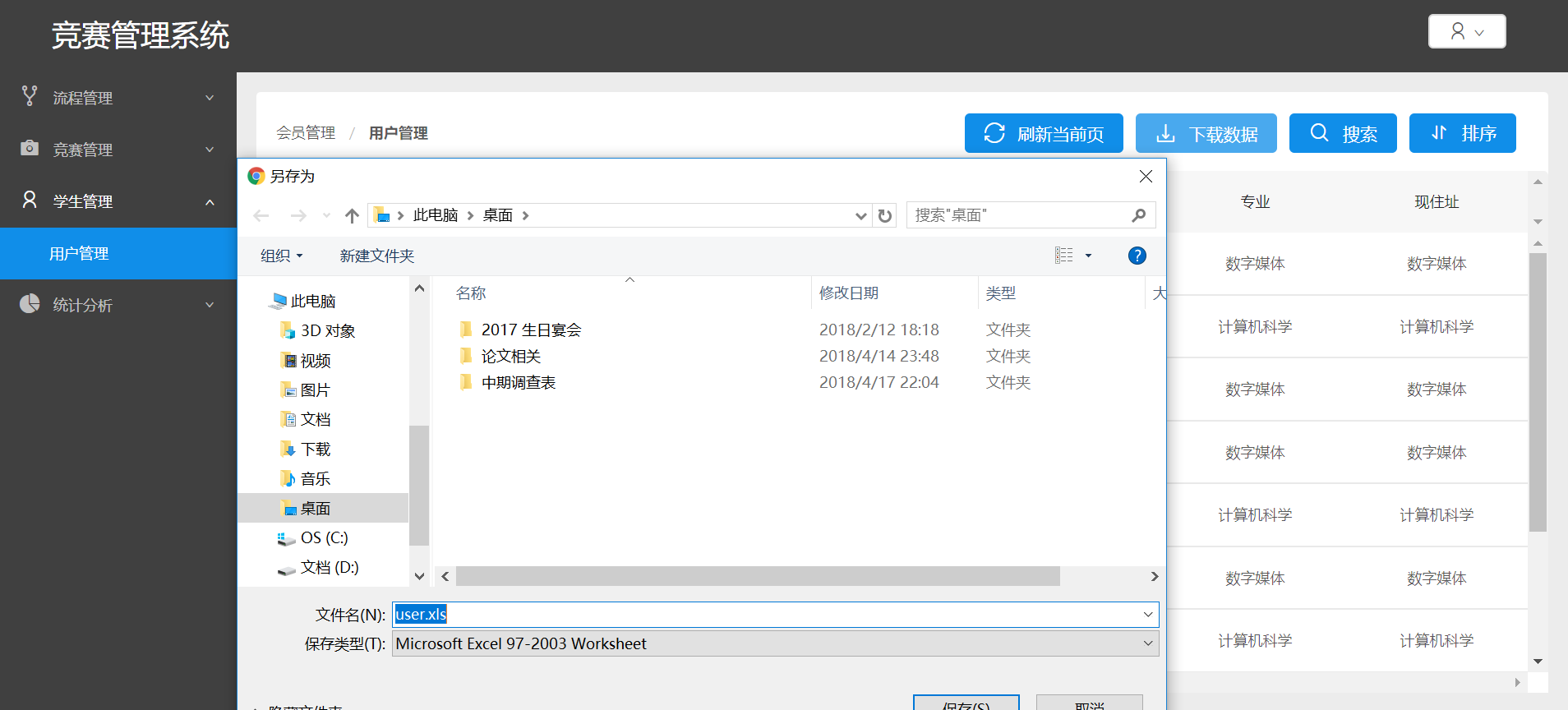


图6-14待审批流程-审核结果为未通过

1. 教师针对学生用户列表进行下载，如图6-15~6-16所示



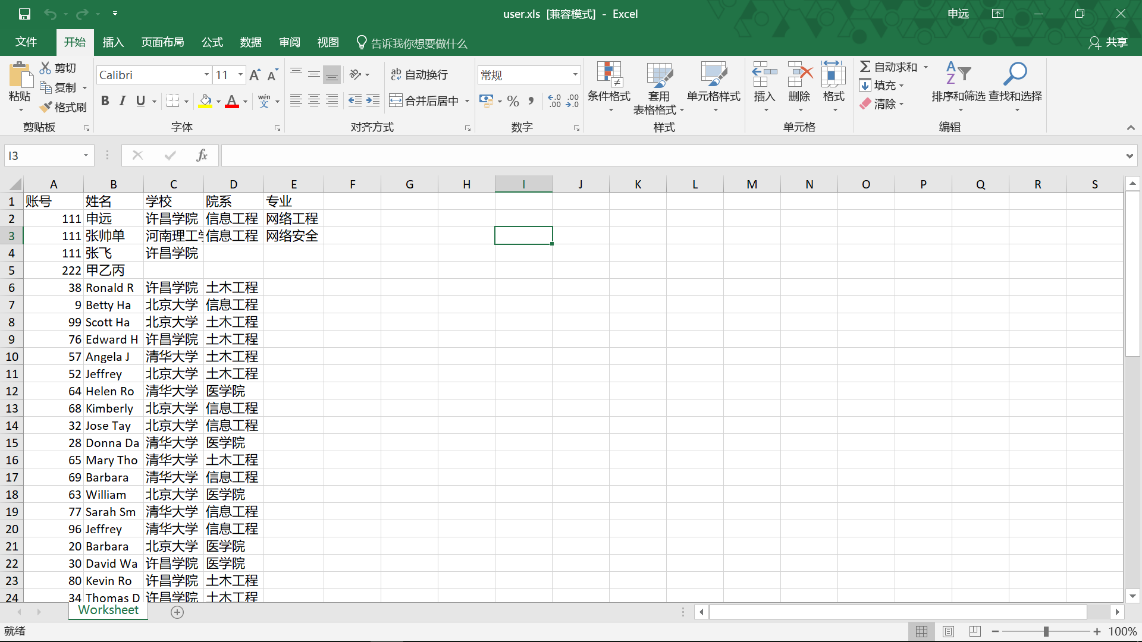
图6-15用户管理-下载操作

图6-16用户管理-下载用户表

（6）教师针对数据分析管理页进行查看和下载，如图6-17~6-18所示

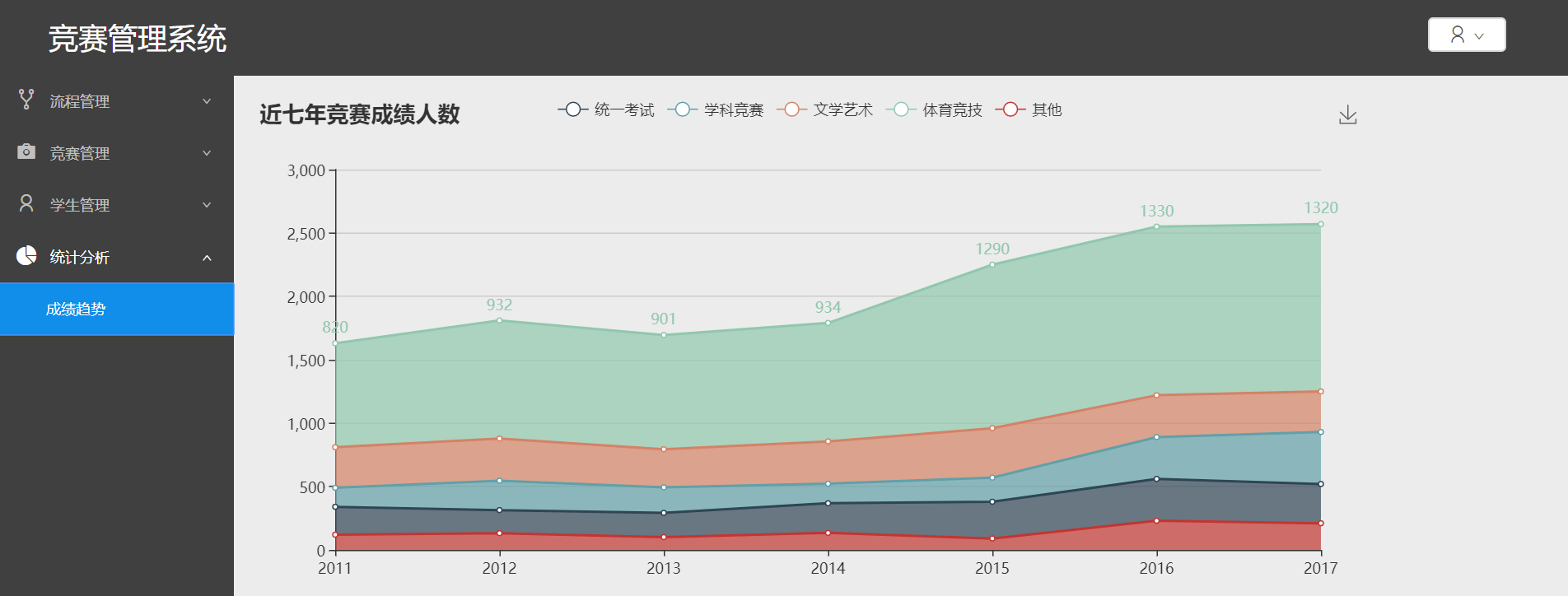


图6-17统计分析-成绩趋势

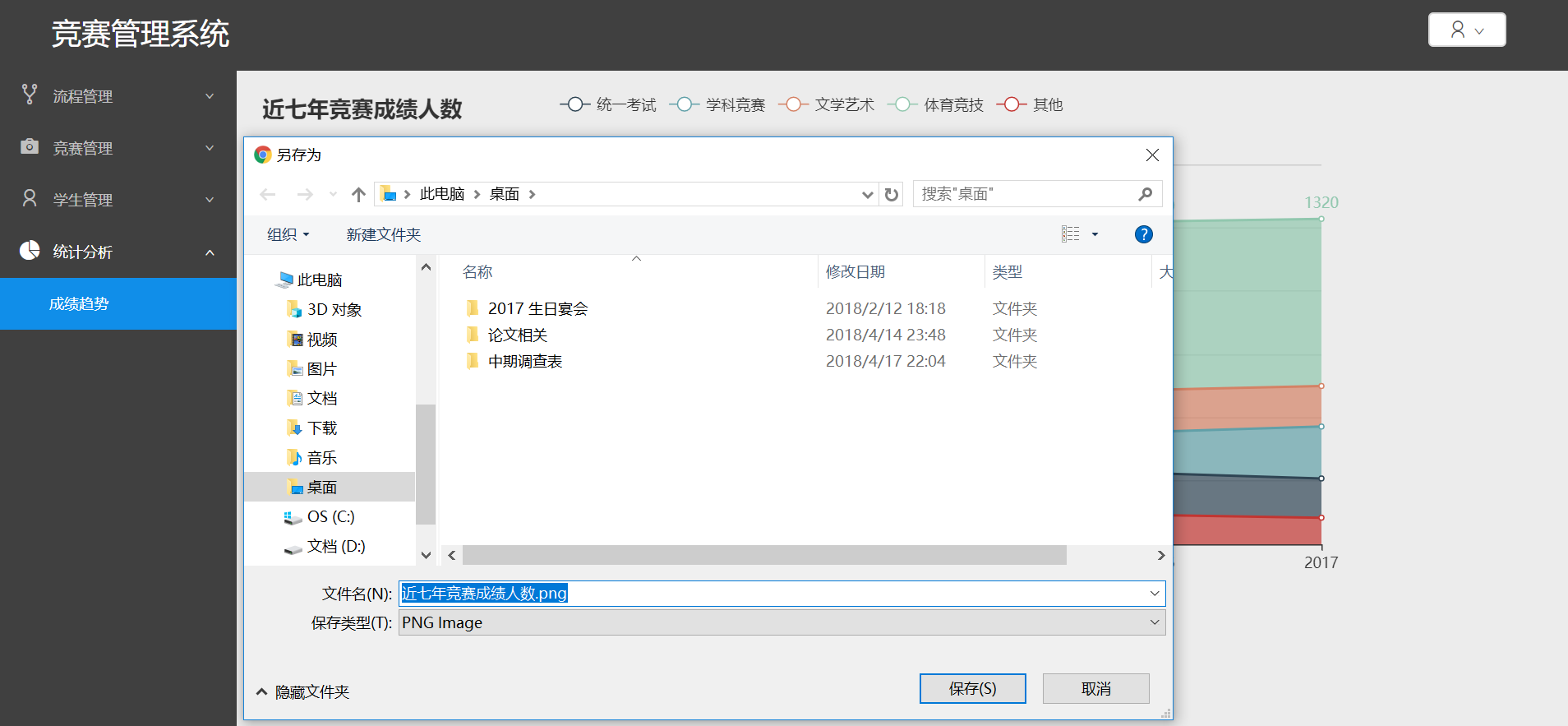


图6-18统计分析-保存为图片

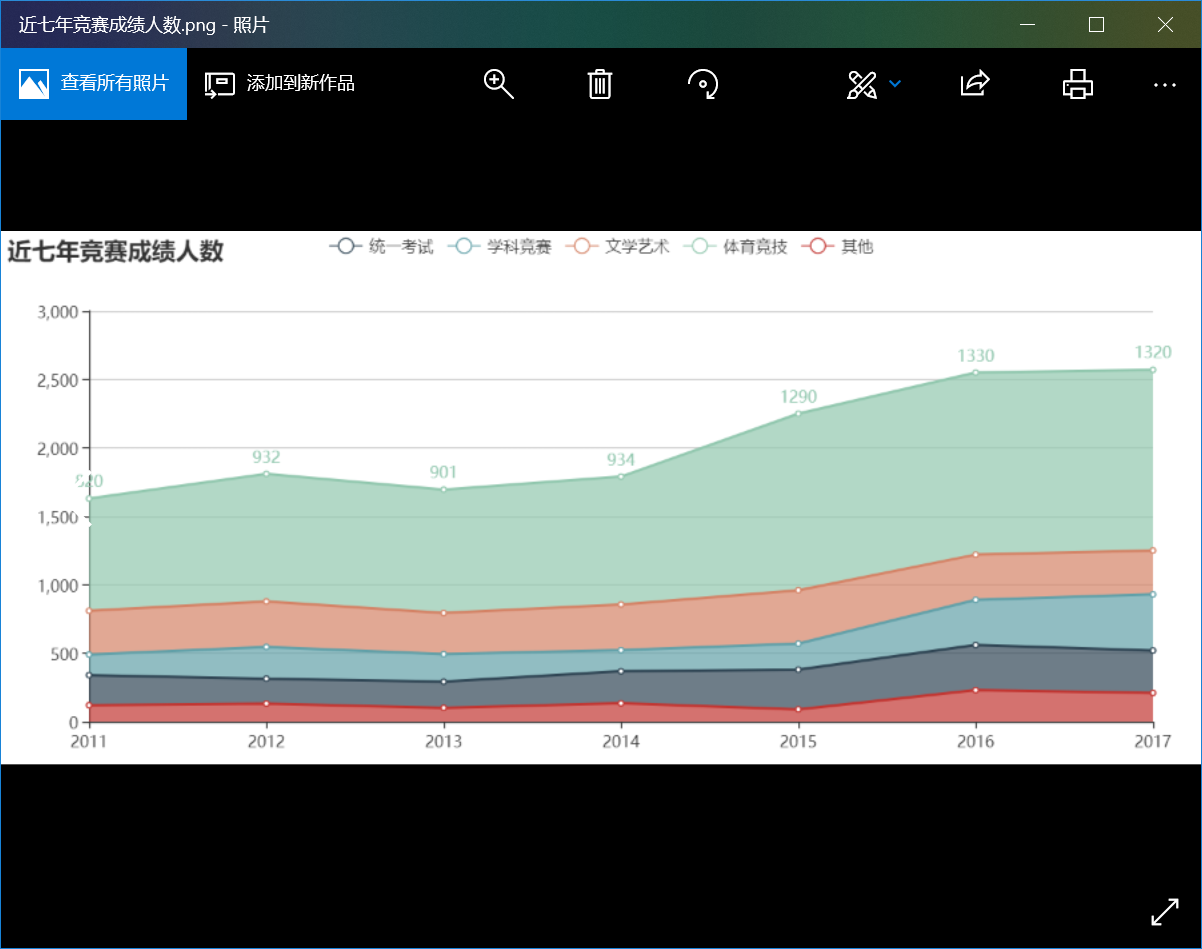


图6-19统计分析-保存图片查看

经以上所有测试步骤之后，证明该系统工作正常，所有功能已经基本实现，能够投入到日常教学生活当中。但该系统中也存在一定程度的不足，还需要在今后的使用过程中，进行不断的维护和优化工作。

# 结束语

本系统集服务端和前端技术，为高校提供了一个能够高效率统计学生竞赛成绩信息的平台。通过web端向服务端请求查询多种数据，并在客户端进行呈现；在服务端，系统接受各种请求参数，然后对其进行相关数据处理及格式化的操作，还能够根据查询数据，并进行下载操作。

在系统编写过程中，由于技术有限，有许多技术点不够明确，后来花费大量的时间去查阅相关的资料才得以解决。例如在实现表格导出下载这个功能过程中，当时由于不清楚后台服务端将查询的数据以二进制的方式返回给前端后，前端如何进行转换至Excel文件，在经过了大量的尝试后，上网查阅了大量资料后才得以解决问题；还有关于前端的异步操作，当时由于是在本机调试操作，后端响应速度特别快，不会出现异步问题。但是在实际线上的环境中，可能会由于各种原因到时服务器不能及时响应，此时就需要前端在请求数据时进行异步操作，从而优化用户的体验。在进行了大量的调查及研究，最终通过w3c的相关JavaScript2015版本的promise得以解决。

# 参考文献

[1] Minko Gechev.Switching to Angular2 [M]北京:中国工业出版社，2017

[2] 阮一峰.ES6标准入门(第三版) [M]北京：清华大学出版社，2017：42

[3] 刘涛/陈学敏.HTML&css设计与构建网站 [M]北京:中国工业出版社，2013

[4] 武欣.PHP和MySQL Web开发[M]北京:机械工业出版社，2009

[5] 姜成尧.MySQL技术内幕2 [M]北京: 机械工业出版社，2012

[6] 杨磊/张志美. JavaScript网页特效经典300例[M]北京:电子工业出版社，2017

[7] 郭家宝. Node.js开发指南 [M]北京:人民邮电出版社，2012

[8] 微软开源. TypeScript中文网[OL] https://www.tslang.cn/，2017

[9] 谷歌开源.Angular开发文档 [OL] https://angular.cn/，2017

[10] 淘宝开源. NG-ZORRO [OL] https://ng.ant.design/#/docs/angular/，2017

# 致谢

毕业设计能够顺利的完成，离不开陈艳格导师的悉心指导。从毕业设计的选题，到牵涉的功能需求，都为我提供了大量宝贵的意见及建议。针对系统的整体架构方面进行了比较详细的指导说明。并在我实现过程中，不厌其烦的为我提供修改和改进意见。陈老师不管是在理论知识还是架构设计方面，都给予了我极大的帮助，使我在毕业设计期间学到了需要东西。

其次我要感谢我的学校和同学，是学校给予了我前端技术的基础知识，使我对前端产生了浓厚的兴趣，。当有技术上的疑问时，是身边的同学帮助我，给我提供了宝贵的意见，使得技术得以解决和实施。

在这里，我要感谢在我毕业设计过程中，所有曾经帮助过我的导师、同学、老师，毕业设计的完成，是离不开你们的帮助的。