**设**

**计**

**开**

**发**

**文**

**档**

目录

[1、需求分析 1](#_Toc71310934)

[2、概要设计 1](#_Toc71310935)

[2.1 系统架构设计 1](#_Toc71310936)

[2.2 软件结构设计 3](#_Toc71310937)

[2.2.1 概述 3](#_Toc71310938)

[2.2.2 前端结构设计 3](#_Toc71310939)

[2.2.3 后端结构设计 3](#_Toc71310940)

[3、详细设计 6](#_Toc71310941)

[3.1 运行环境 6](#_Toc71310942)

[3.1.1 软件环境 6](#_Toc71310943)

[3.2 开发环境 6](#_Toc71310944)

[3.2.1 软件环境 6](#_Toc71310945)

[3.3 界面设计 7](#_Toc71310946)

[3.3.1 界面及功能展示 7](#_Toc71310947)

[4、测试报告 15](#_Toc71310948)

[4.1 后端单元测试 15](#_Toc71310949)

[4.1.1 测试方式: 15](#_Toc71310950)

[4.2.2 测试结果： 15](#_Toc71310951)

[4.2 后端集成测试 15](#_Toc71310952)

[4.2.1 测试方式: 15](#_Toc71310953)

[4.1.2 测试结果： 15](#_Toc71310954)

[4.1 前端兼容性测试 15](#_Toc71310955)

[4.1.1 测试方式: 15](#_Toc71310956)

[4.1.2 测试结果： 15](#_Toc71310957)

[4.2 前端可视化质量测试 17](#_Toc71310958)

[4.2.1 测试方式: 17](#_Toc71310959)

[4.2.2 测试结果： 18](#_Toc71310960)

[4.3 前后端结合测试 19](#_Toc71310961)

[4.3.1 测试方式: 19](#_Toc71310962)

[4.3.2 测试结果： 19](#_Toc71310963)

[4.4 环境配置 20](#_Toc71310964)

[4.4.1 JDK安装 20](#_Toc71310965)

[4.4.2 MySQL安装 20](#_Toc71310966)

[4.4.3 Maven-3.5.4安装 20](#_Toc71310967)

[4.5 系统部署 21](#_Toc71310968)

[4.5.1 前端部署 21](#_Toc71310969)

[4.5.2 后端部署 21](#_Toc71310970)

# 1、需求分析

对于大学生来说，学科竞赛活动是在紧密结合课堂教学的基础上，以竞赛的方式，激发学生理论联系实际和独立工作的能力，通过实践来发现问题、解决问题，增强学生学习和工作自信心的系列活动,学科竞赛具有探索性、创造性和科学性,既无任何捷径可走，又需要付出艰苦的劳动。

而对于软件学院的学生来说，最为重要竞赛类型之一便是程序设计类竞赛。我院的ACM团队自建立以来，已参加了数百场的竞赛和组织了数千场的训练的同时也积累了大量宝贵的数据。为了更好的收集、管理和分析这些数据，进一步提高团队信息化水平和监督效能，同时实现数字化团队管理，我们将基于B/S、MVC、JavaEE三种基本架构进行数据管理分析及可视化平台开发。

# 2、概要设计

## 2.1 系统架构设计

本系统架构如图1所示，我们采用前后端分离的开发模式实现高内聚低耦合，可以使后台能更好的追求高可用、高性能，使前端能更好的追求页面表现、速度流畅、兼容性、用户体验，同时降低维护成本，客户端的问题不再需要后台人员参与及调试，代码重构及可维护性增强。通过本系统团队管理人员能够更加高效便捷的进行数字化团队管理，随时了解队员最新训练情况，同时能够对多种数据进行可视化分析。

前端我们使用webpack + Vue.js构建前端工程化，并采用Vue.js框架，实现组件化和模块化开发，配合Vue Router以及Vuex从而可以进行单页面应用开发并存储管理应用的所有组件的状态，还可以实现页面局部刷新，同时将数据和结构分离，不用每次跳转页面都要请求所有数据和DOM，大幅度提升了访问速度和用户体验。界面美化采用Element UI框架。同时网络请求库选择采用Axios，其浏览器兼容性好（能兼容IE7），且简单易用，api接近于jquery。

接口方面采用RESTful架构风格，进一步前后端分离，减少流量，使前端无关化，后端只负责数据处理，让前端和后端人员更加专注于各自开发，只需接口文档便可完成前后端交互，无需过多相互了解

为了后端api能跨平台部署，我们将采用Java语言进行后端api平台的开发，后端运用SpringBoot技术进行高效快速的模块化开发，使用Maven简化项目依赖管理，同时遵循阿里巴巴Java开发规范，SpringBoot减少了大量的硬编码的同时内嵌了Srvlet容器，降低了对环境的要求，机器有java运行环境即可运行，并且能够快速整合第三方框架，无需配置文件。后端返回的所有数据皆以Json格式进行传输，该格式的数据体积更轻量，读写更快且结构简单。安全方面采用Shiro框架，轻便灵活且能与Spring无缝结合，身份验证、授权、加密和会话管理。

|  |
| --- |
| 软件学院数据管理分析系统-【系统架构图】  图1 系统架构 |

## 2.2 软件结构设计

### 2.2.1 概述

对于软件整体我们是基于模块化开发的思想，将前端和后端分别划分为5个不同的子模模块。

### 2.2.2 前端结构设计

平台前端结构如图2所示，我们引用了Vue.js功能性框架，实现了简单API的响应式数据绑定和组合，同时在vue.js的基础上，作品还引用了element-ui UI库与之配合，美化界面。在工程化方面引用了Webpack开源工具，对静态资源进行打包，减少io请求。同时遵守了eslint的代码规范，保证代码的一致性。

代码方面，我们采用模块化开发的方法，分为队员模块、竞赛模块、获奖模块、分析模块四个模块及登录接口。

|  |
| --- |
| 图2 前端结构设计 |

### 2.2.3 后端结构设计

|  |
| --- |
| 后端结构如图3所示，我们采用了流行的SpringBoot+SpringMVC+MyBatis-Plus框架，安全方面采用的是较为轻量级的Shiro框架，同时遵守阿里巴巴的Java开发代码规范，并使用Git做了版本控制，避免源码的混乱。  在代码方面，我们将抽取公共功能模块进行抽取，提高了开发效率同时提升了代码的可复性。最后基于模块化开发的思想，分为安全模块（如图4）、队员模块（如图5）、比赛模块（如图6）、获奖模块（如图7）、牛客训练记录模块（如图8）这五大模块。    图3 后端结构设计 |
| 图4 安全模块结构图    图5 队员模块结构图    图6 比赛模块结构图    图7 获奖模块结构图    图8 牛客训练记录模块结构图  图4至图8为各模块的基本结构图 |

# 3、详细设计

## 3.1 运行环境

### 3.1.1 软件环境

|  |  |  |  |
| --- | --- | --- | --- |
| **分类** | **名称** | **版本** | **语种** |
| **操作系统** | Windows 10 | X64 | 简体中文 |
| **操作系统的附加功能** | 64位 | —— | —— |
| **数据库平台** | MySQL | 5.7 | 简体中文 |
| **Java运行环境** | JDK | 1.8.0\_261 | 简体中文 |

## 3.2 开发环境

### 3.2.1 软件环境

|  |  |  |  |
| --- | --- | --- | --- |
| **分类** | **名称** | **版本** | **语种** |
| **操作系统** | Windows 10 | x64 | 简体中文 |
| **操作系统的附加功能** | 64位 | —— | —— |
| **数据库平台** | MySQL | 5.7 | 简体中文 |
| **编译器** | IntelliJ IDEA | 2020.1 | 简体中文 |
| Visual Studio Code | 1.55.2 | 简体中文 |
| **开发工具** | Navicat Premium 15 | 15.0.22 | 简体中文 |
| ApiPost | 5.2.5 | 简体中文 |
| Power Designer | 16.0.5.3982 | 英文 |
| Git | 2.28.0.windows.1 | 简体中文 |
| JDK | 1.8.0\_261 | 简体中文 |
| Maven | 3.5.4 | 英文 |
| vue-devtools | 5.3.4 | 英文 |
| Chrome | 90.0.4430.85 | 简体中文 |

## 3.3 界面设计

### 3.3.1 界面及功能展示

本系统前端各界面如图9至图24所示（图下方为该图操作说明）。

|  |
| --- |
| 图9 登录界面  图9展示了系统登录界面，此时管理员输入账户密码即可完成登陆操作。 |
| 图10 首页  图10为主页面，展示了成员分布和历史获奖。 |
| 图11 队员信息管理界面  图11为队员模块的信息管理界面，可以对队员信息进行查看，修改，搜索等操作 |
| 图12 队员信息修改界面  图12队员的信息修改界面，管理员可以在此界面修改队员的信息 |
| 图13 队员数据上传录入界面  图13为队员数据上传录入界面，选取Excel文件之后上传即可，也可下载上传模板 |
| 图14 队员牛客训练详情界面  图14展示了队员牛客训练详情界面，管理通过队员信息管理界面查看队员最近的牛客训练情况，点击标题可跳转至该场比赛的详情页 |
| 图15 队员学科成绩界面  图15展示了队员学科成绩情况，管理员通过点击队员牛客训练详情界面的学科详情的按钮即可查看该队员的学科成绩 |
| 图16比赛管理界面  图16展示了已经录入数据库的比赛信息，点击可查看这场比赛详情 |
| 图17 比赛详情可视化界面  图17将该场比赛的数据进行了可视化分析展示 |
| 图18 比赛信息修改界面  图18管理员可以在此界面修改比赛的信息 |
| 图19 比赛信息删除界面  图19管理员可以通过此界面删除比赛 |
| 图21 比赛数据上传界面  图21为数据上传界面，管理员可上传Excel模板的数据 |
| 图22 队员数据可视化分析界面  图22为某位队员数据可视化分析界面，展示了该队员的三类重要学科的成绩分布情况，以及牛客训练走势及获奖情况 |
| 图23 历年获奖界面  图23展示了历年的获奖情况，管理员可以点击学号姓名跳转至图14队员牛客训练详情界面查看该队员的训练情况和学科成绩 |
| 图24 获奖上传界面  管理员在此界面可以上传获奖信息，可以选择Excel上传也可以通过表单逐一上传 |

# 4、测试报告

## 4.1 后端单元测试

### 4.1.1 测试方式:

1. 利用JUnit框架对所有Controller和Service得方法进行单元测试

### 4.2.2 测试结果：

全部通过，无任何异常和错误

## 4.2 后端集成测试

### 4.2.1 测试方式:

1. 启动后端项目
2. 使用ApiPost模拟远程调用所有得Controller的Api

### 4.1.2 测试结果：

全部Api均能正常调用且返回的Json数据均为期望数据

## 4.1 前端兼容性测试

### 4.1.1 测试方式:

1、在不同平台进行页面访问

2、使用不同浏览器进行访问

### 4.1.2 测试结果：

1. 平台兼容性好，页面兼容PC端和移动端

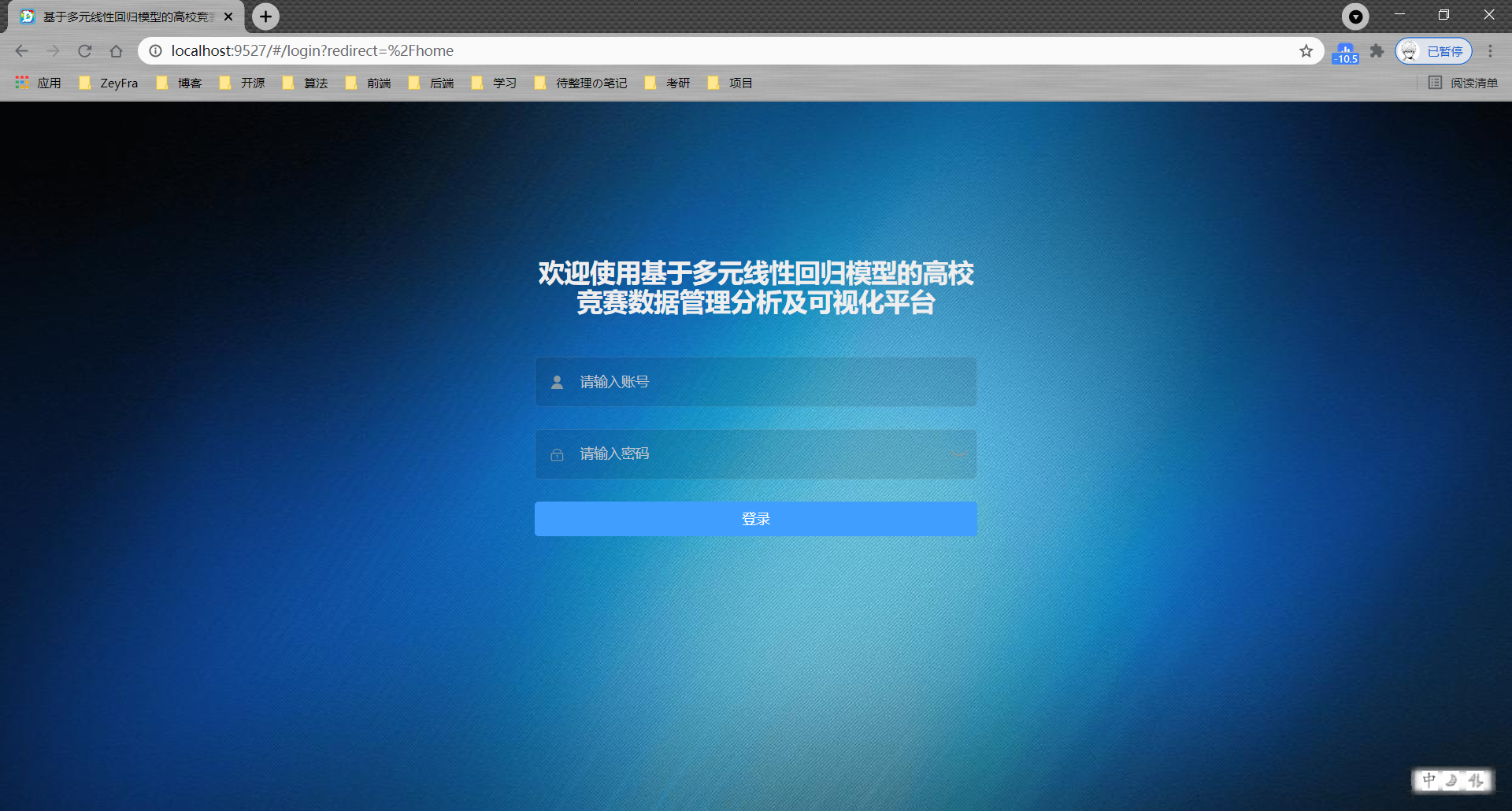


图25 PC端登录窗口

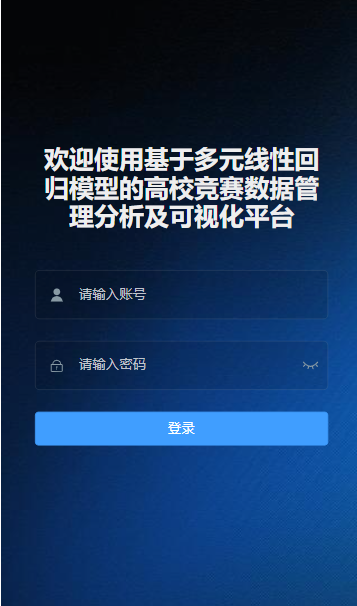


图26 移动端登录窗口



图27 PC端页面展示



图28 移动端页面展示

## 4.2 前端可视化质量测试

### 4.2.1 测试方式:

1、进行E2E测试，渲染结果像素基本验证

2、使用cypress模块进行E2E测试，进行自动化的交互操作

### 4.2.2 测试结果：

渲染结果像素基本验证

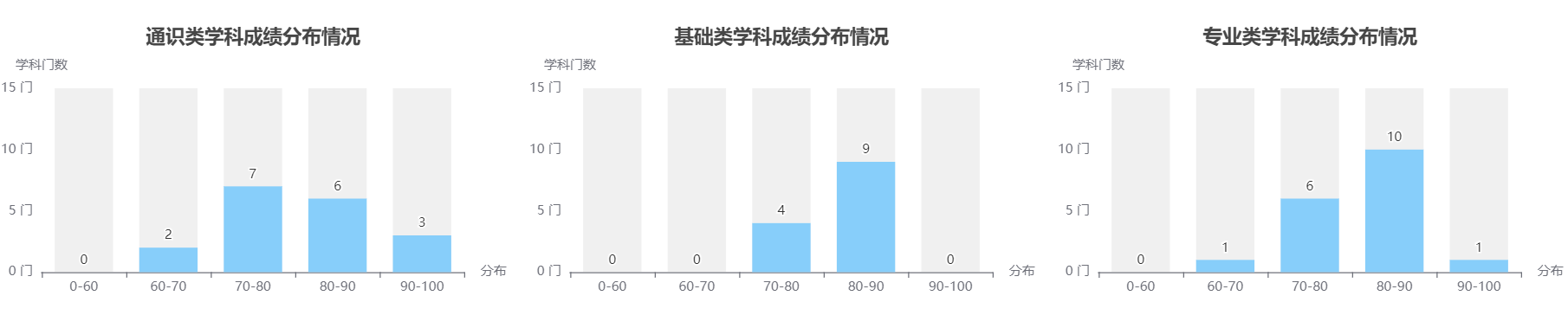


图28 各科成绩分布情况

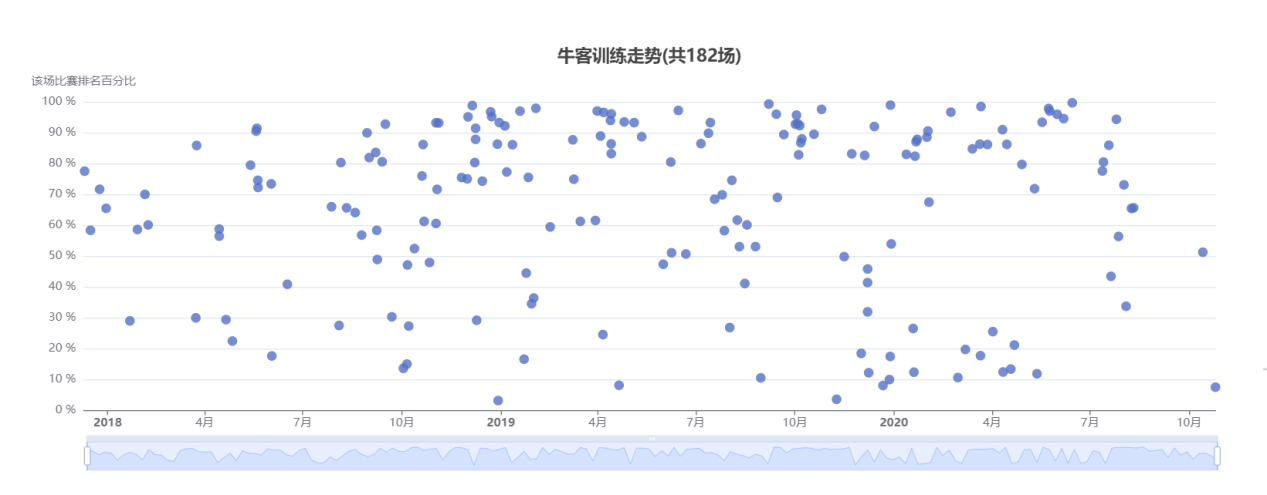


图29 训练走势情况



图30 获奖情况

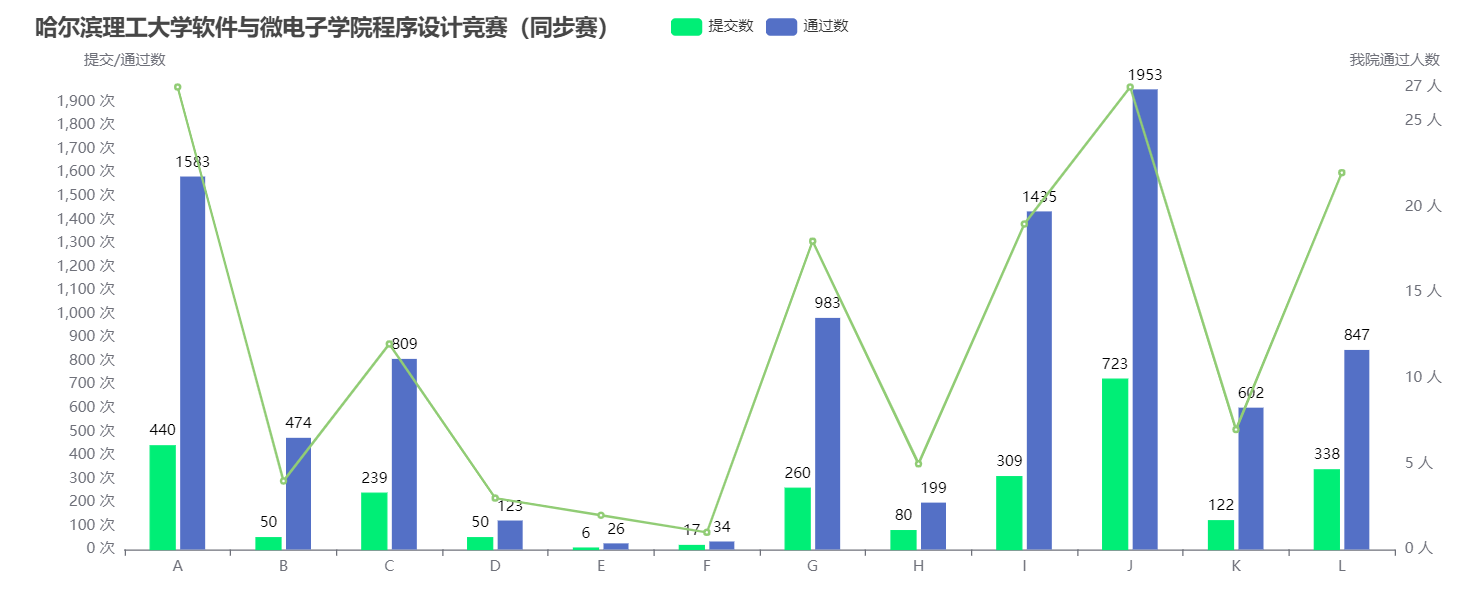


图31 竞赛情况

## 4.3 前后端结合测试

### 4.3.1 测试方式:

1. 启动前端项目
2. 后端通过配置类添加跨越过滤器解决跨域问题
3. 启动后端项目
4. 在浏览器进行正常操作

### 4.3.2 测试结果：

1. 输入账户密码，能够正常登录，且其余功能全部正常

|  |
| --- |
| 图32    图33  图32至图33为输入账号密码后点击登录，登陆成功 |

## 4.4 环境配置

### 4.4.1 JDK安装

1、把下载好的Linux版本的jdk1.8的压缩文件，用xftp工具上传到Linux服务器中的usr/local文件中

cd/usr/local

jdk-8u261-linux-x64.tar.gz

2、解压安装包

tar -zxvf jdk-8u261-linux-x64.tar.gz

3、建立软连接(可以省略)

ln -s /usr/local/jdk1.8.0\_121/ /usr/local/jdk

4、修改环境变量

vim /etc/profile

export JAVA\_HOME=/usr/local/jdk

export JRE\_HOME=$JAVA\_HOME/jre

export CLASSPATH=.:$CLASSPATH:$JAVA\_HOME/lib:$JRE\_HOME/lib

export PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

（修改完成后按ESO键，然后输入:wq 保存退出）

通过命令source /etc/profile让profile文件立即生效

source /etc/profile

5、测试是否安装成功

java -version

### 4.4.2 MySQL安装

双击打开路径下的ZK\_Driver.exe，完整安装直至安装程序完成退出。

### 4.4.3 Maven-3.5.4安装

1、上传或下载安装包

cd/usr/local

apache-maven-3.5.4-bin.tar.gz

2、解压安装包

tar -zxvf apache-maven-3.6.1-bin.tar.gz

3、建立软连接(可以省略)

ln -s /usr/local/apache-maven-3.6.1/ /usr/local/maven

4、修改环境变量

vim /etc/profile

export MAVEN\_HOME=/usr/local/maven

export PATH=$PATH:$MAVEN\_HOME/bin

（修改完成后按ESO键，然后输入:wq 保存退出）

source /etc/profile（通过命令source /etc/profile让profile文件立即生效）

5、测试是否安装成功

mvn –v

## 4.5 系统部署

### 4.5.1 前端部署

1. 执行**npm run build**命令对项目打包（在windows中可以直接在项目文件夹直接cmd进行打包；在Linux中进行打包，不过要先在Linux系统中安装node.js

注意：打包前修改vue.config.js中的publicPath，将其修改为./，同时修改路由为后端地址

1. 利用xftp工具把打包后的dist文件夹上传到Linux中的www文件下
2. 在服务器Linux上安装web服务器nginx1.18
3. 进入到Nginx目录下的sbin目录，输入“./nginx”命令启动Nginx
4. 打包到admin文件下，所以配置nginx的http部分如下即可访问

server {

listen 80;

server\_name 175.24.147.35;

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

root /root/MyProjects/dmas-web/admin;

try\_files $uri $uri/ /index.html;

index index.html index.htm;

}

}

### 4.5.2 后端部署

1. 安装MySQL 5.7数据库
2. 安装JDK 1.8.0\_261
3. 安装Maven-3.5.4
4. 使用“mvn package”命令进行项目打包
5. 编写脚本
   1. 启动脚本（start.sh）

nohup java -jar \*\*\*\*\*\*\*.jar --spring.profiles.active=prod > nohup.out 2>&1 &

* 1. 停止脚本（stop.h）

PID=$(ps -ef | grep eladmin-system-2.6.jar | grep -v grep | awk '{ print $2 }')

if [ -z "$PID" ]

then

echo Application is already stopped

else

echo kill -9 $PID

kill -9 $PID

fi

* 1. 查看日志脚本（log.sh）

tail -f nohup.out

* 1. 配置nginx

server {

listen 80;

server\_name 域名/当前服务器外网IP;

location / {

proxy\_pass http://127.0.0.1:8000; #这里的端口记得改成项目对应的哦

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header X-Forwarded-Port $server\_port;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

}

}