

# 《数字百团》项目报告

项目负责人：谢锦培（组长）、黎谨华、吕杰、陈天蔚、谢靖宁

## 目录

- 引言
- 项目背景与目标
- 项目演示
- 项目设计及分工
- 结论及展望

## 引言

随着信息技术的飞速发展和数字化转型的深入推进，高校社团管理也迎来了新的变革机遇。为了提升社团管理的效率与透明度，同时为学生提供更加便捷、丰富的社团参与体验，我们倾力打造了“数字百团”社团管理系统。本系统作为一个综合性平台，旨在通过数字化手段全面管理和协调校园内的社团活动，覆盖校团委、各社团及申请者三大用户端，实现从社团信息管理到申请处理的一系列功能。在传统的高校社团管理模式下，信息流通不畅、管理效率低下、学生参与度不高等问题日益凸显。为了解决这些问题，我们提出了“数字百团”社团管理系统的开发计划。本项目的主要目标是构建一个集社团信息管理、活动组织、成员招募、申请审批等多功能于一体的数字化平台，以提高社团管理的效率与透明度，并为学生提供更加便捷、丰富的社团参与体验。本报告将详细阐述该项目的背景、目标、系统设计、实现过程、特色功能及实际应用效果，以期类似项目的开发与实施提供参考与借鉴。

## 项目背景与目标

项目开源仓库网址：[“数字百团”仓库](#)

### 一、行业背景

- 社团管理现状：**当前，高校社团数量众多，成员构成复杂，活动频繁。传统的管理方式往往依赖于纸质文档或简单的电子表格，存在信息更新不及时、查询不便、易丢失等问题，严重制约了社团管理的效率和效果。
- 数字化趋势：**随着互联网的普及和信息技术的快速发展，数字化、信息化已成为社会各领域的发展趋势。教育行业也不例外，高校正在积极推进智慧校园建设，社团管理作为其中的重要组成部分，其数字化转型势在必行。
- 用户需求：**社团成员、管理员及超级管理员等角色对社团管理系统有着不同的需求。学生需要方便快捷地查看社团信息、参与活动；管理员需要高效地管理社团成员、组织活动、分配资源；超级管理员则需要对整个系统进行监控和维护，确保系统的稳定运行。

## 二、技术背景

- 1. Java技术栈：**Java作为一种成熟、稳定、跨平台的编程语言，在企业级应用开发中具有广泛应用。SpringBoot作为Spring框架的简化版，提供了快速开发Spring应用的工具，极大地提高了开发效率。MybatisPlus作为Mybatis的增强工具，在数据库操作方面提供了更为便捷的方式。
- 2. Vue3与Element UI：**Vue3作为前端框架的佼佼者，以其轻量级、响应式、组件化的特点受到开发者的青睐。Element UI作为Vue的UI组件库，提供了丰富的界面元素，有助于快速构建美观、易用的前端界面。
- 3. 其他关键技术：**
  - **Md5加密：**保障用户密码安全，防止信息泄露。
  - **文件上传与Excel识别：**支持批量导入数据，提高管理效率。
  - **定时任务：**实现日志定时保存等功能，保障系统稳定运行。
  - **流量控制：**防止服务器因并发请求过多而崩溃。
  - **全局异常处理：**提高系统健壮性，快速定位并解决问题。
  - **Redis缓存：**提高数据访问速度，减轻数据库压力。
  - **数据可视化：**通过图表展示数据，直观呈现社团管理情况。
  - **邮箱验证：**增强用户注册及密码找回的安全性。

## 三、项目目标

“数字百团”社团管理系统的开发，可以解决了高校社团管理中的实际问题，提高了管理效率和质量，推动了高校社团管理的数字化转型和智慧化升级。同时，我们希望该项目的实施能为类似系统的开发提供了宝贵的经验和参考，可以具有重要的实践意义和推广价值。

## 四、项目逻辑

学生端：申请加入社团（向管理员端）、搜索查看社团、修改个人信息、申请加入活动（向管理员端）、查看已加入的社团

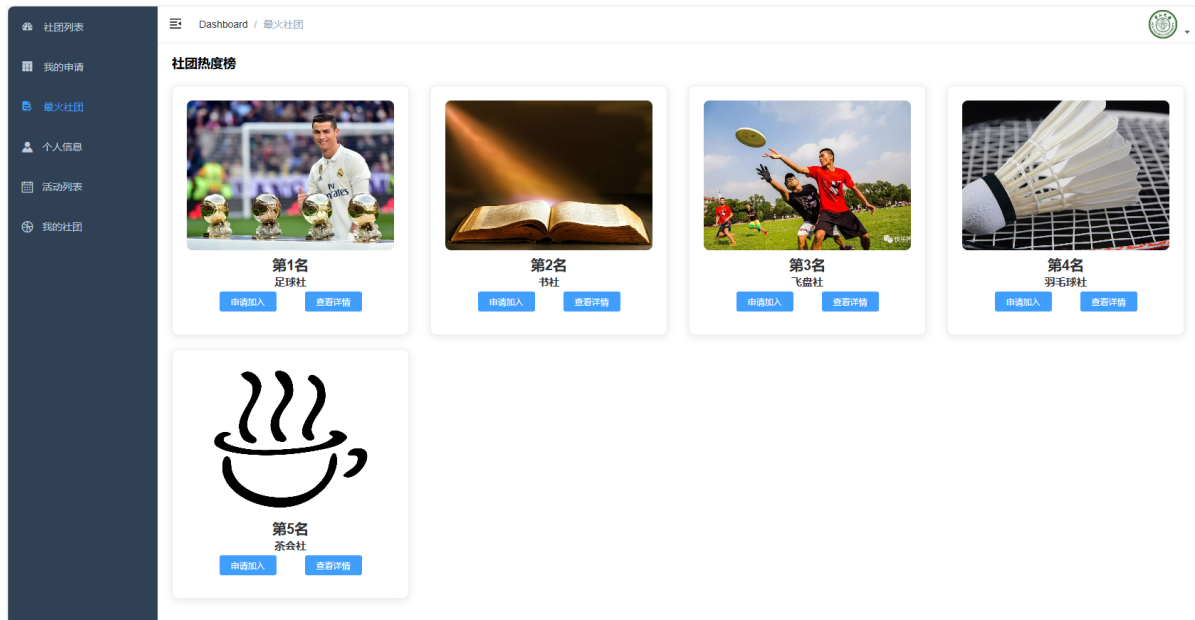
管理员端：查看申请加入社团学生列表（及批准或拒绝）、查看社团成员、申请举办活动（向超管端）、编辑社团详情

超级管理员端：查看相关可视化数据、查看所有社团/学生、审批活动、批量添加学生/管理员/社团

## 项目演示

基本功能展示：

学生端部分截图：



管理员端部分截图：

申请列表

社团成员

社团编辑详情

社团活动

申请活动

更改活动

Dashboard / 申请列表

姓名	学号	申请时间	申请状态
吴舜宇	21311272	2024年5月18日 16:16:37	已批准
陈建铨	21952107	1970年1月1日 08:00:00	已批准
杨沛粤	21311296	1970年1月1日 08:00:00	已批准
陈建铨	21311245	1970年1月1日 08:00:00	已拒绝
大神	21311272	1970年1月1日 08:00:00	已拒绝
是是是	22222222	1970年1月1日 08:00:00	已拒绝
大师	21318888	1970年1月1日 08:00:00	已拒绝
吴舜宇	21311272	1970年1月1日 08:00:00	已批准
陈建铨	21952107	1970年1月1日 08:00:00	待审核
张三	21114514	1970年1月1日 08:00:00	已批准
吕杰	21311366	2024年7月9日 10:13:05	待审核

申请列表

社团成员

社团编辑详情

社团活动

申请活动

更改活动

Dashboard / 社团编辑详情

社团描述 (必填)

上传附件 (可选)

上传图片 (可选)

确认修改 (必点)

社团名称

足球社

社长

B熊

\* 详细介绍

C罗sluuuuuuul

上传附件

只能上传zip文件,且不超过10MB

将社团图片拖到此处, 或点击上传

只能上传jpg/png文件

确认修改

申请列表

社团成员

社团编辑详情

社团活动

申请活动

更改活动

Dashboard / 更改活动

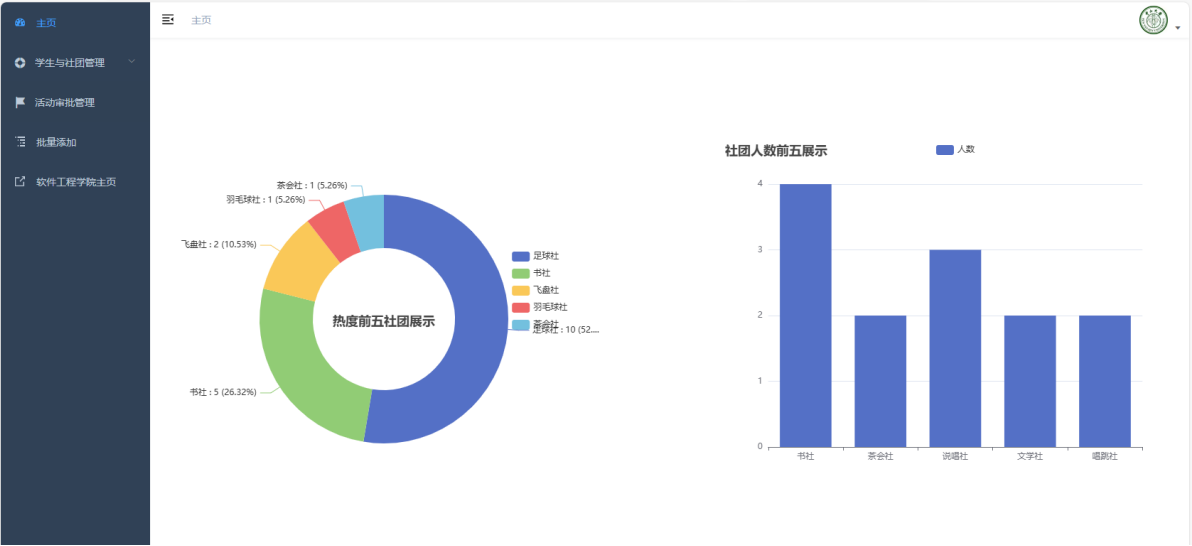
搜索活动

Q

清空搜索

活动名	活动编号	组织者	申请时间	申请状态
对唱歌和篮球关系的研究	2	吕杰	1970年1月1日 08:00:00	已通过 <div>修改</div>
WutheringWaves	4	吕杰	2024年7月1日 09:11:30	已通过 <div>修改</div>
训练	6	吕杰	2024年6月5日 09:11:26	已通过 <div>修改</div>
团建	7	吕杰	2024年6月6日 09:11:36	已通过 <div>修改</div>
Yuanshen	12	Sora	1919年8月10日 11:46:14	已通过 <div>修改</div>

超级管理员端部分截图：



🏠 主页

👤 学生与社团管理

📅 社团列表

👤 学生列表

📅 活动审批管理

📊 批量添加

🏠 软件工程学院主页

Dashboard / 学生与社团管理 / 学生列表

Filter keyword

▼ 社团

▼ 足球社

21311366吕杰

▼ 飞盘社

21311366吕杰

▼ 文学社

21311366吕杰

21301666哈基米

▼ 说唱社

21311366吕杰

21301666哈基米

21301666小王

▼ 唱歌社

21301011王五

21301666小王

▼ 羽毛球社

21301011王五

▼ 书社

21301011王五

21301666哈基米

21301666小王

▼ 茶会社

21301011王五

21301666小王

🏠 主页

👤 学生与社团管理

📅 社团列表

👤 学生列表

📅 活动审批管理

📊 批量添加

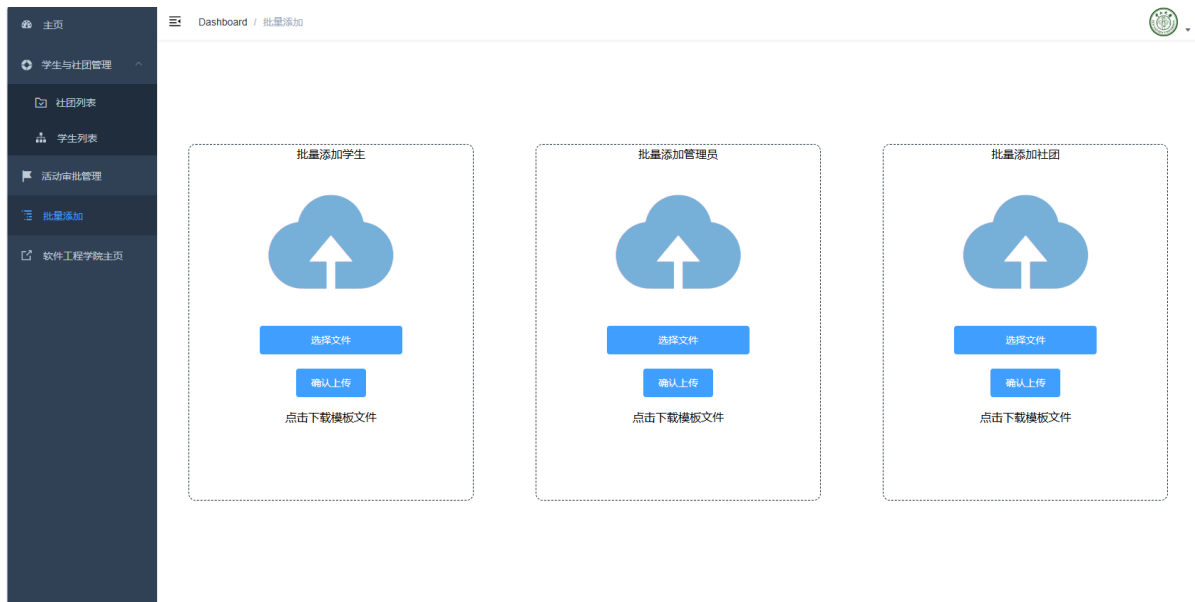
🏠 软件工程学院主页

Dashboard / 活动审批管理

活动数: 17 | 已通过: 9 | 已拒绝: 2 | 待审核: 6

搜索活动 | 清空搜索

活动名	活动编号	组织者	活动时间	申请状态
对唱歌和篮球关系的研究	1	吕杰	1970年1月1日 08:00:00	已批准
对唱歌和篮球关系的研究	2	吕杰	1970年1月1日 08:00:00	已批准
GenshinImpact	3	吕杰	2024年7月27日 09:11:33	待审核
WutheringWaves	4	吕杰	2024年7月1日 09:11:30	已批准
学习会	5	吕杰	2024年7月9日 09:11:23	已拒绝
训练	6	吕杰	2024年6月6日 09:11:26	已批准
团建	7	吕杰	2024年6月6日 09:11:36	已批准
YuanShen	8	Sora	1919年8月10日 11:45:14	待审核
StarRail	9	Sora	1919年8月10日 11:45:14	待审核
ZenlessZoneZero	10	吕杰	2024年7月27日 09:11:33	待审核
YuanShen	11	Sora	1919年8月10日 11:45:14	待审核



# 项目设计及分工

## 一、负责人：谢锦培-21311610

负责部分：主要负责前端页面搭建、前后端对接、整体项目把控、宣传ppt/海报制作、项目报告整合。

在项目过程中，我们组的前端压力比较大，因此我将任务分配的重心放在了前端（前端的学习成本较高），我会先将大体的方向搭建好，再将各个模块任务发给组员，按照讨论好的目标样式一步步推进。同时，与组员们积极沟通，以高效率对接前后端和高效率定位bug。

项目主要完成的任务：

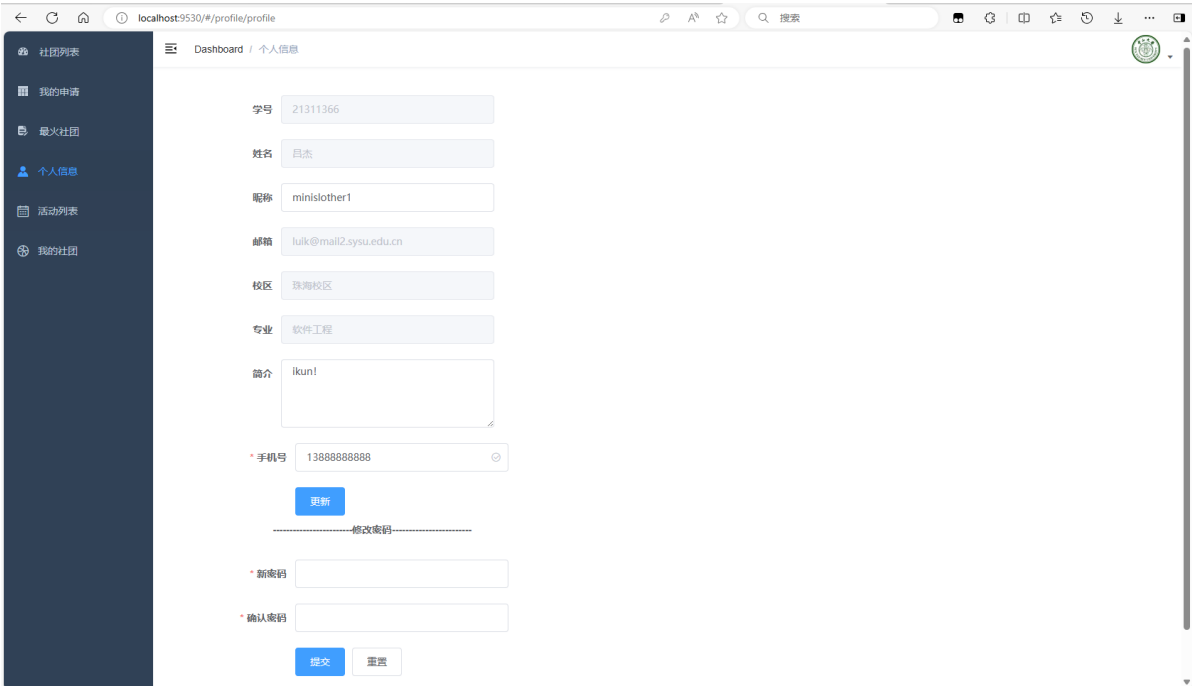
- 1、学生端：个人信息页面、社团列表显示
- 2、管理员端：社团编辑页面、申请列表
- 3、超级管理员端：导航栏搭建、批量导入页面
- 4、全端：上传文件/附件、登陆页面、store储存与调用

其他：

- 1、三前端一后端框架搭建
- 2、前端md5适配
- 4、前端全端规范化命名修改
- 5、各种bug修复等

下面是其中部分负责的前端功能展示：

## 1、前端学生端个人信息部分



The screenshot shows a web browser at localhost:9530/#/profile/profile. The page has a dark blue sidebar on the left with navigation links: 社团列表, 我的申请, 星火社团, 个人信息 (highlighted), 活动列表, and 我的社团. The main content area is titled 'Dashboard / 个人信息' and contains a form for editing profile information. The form fields are: 学号 (21311366), 姓名 (吕杰), 昵称 (minislolter1), 邮箱 (luik@mail2.sysu.edu.cn), 校区 (珠海校区), 专业 (软件工程), 简介 (Ikun!), and 手机号 (13888888888). Below these fields is a '更新' (Update) button. A section for '修改密码' (Change Password) includes fields for 新密码 (New Password) and 确认密码 (Confirm Password), with '提交' (Submit) and '重置' (Reset) buttons at the bottom.

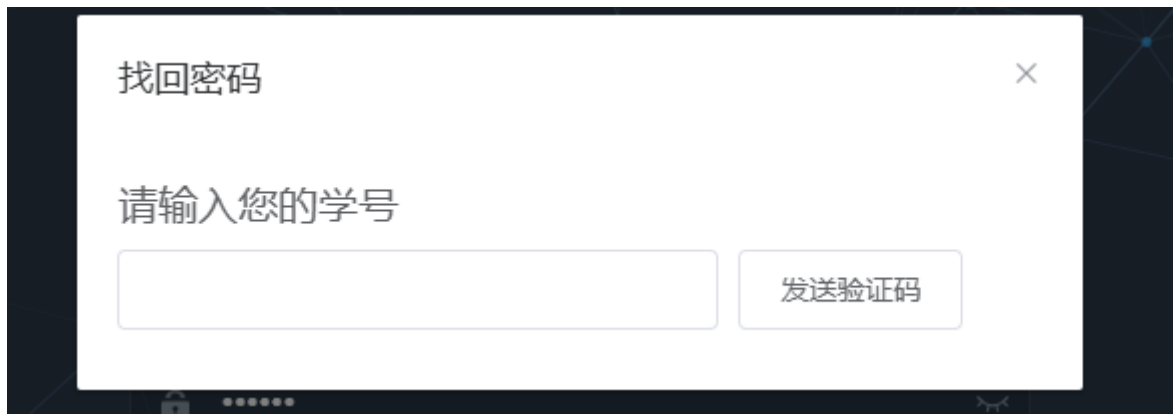
## 2、前端学生端忘记密码部分

该功能主要是用户忘记密码时希望重置密码，点击登陆界面的这个按钮后，



The screenshot shows a login page titled '社团招新申请者登录' (Student Union Recruitment Applicant Login). The background is dark blue with a network pattern. The login form includes a 学号 (Student ID) field with the value 21311366, a password field with masked characters, and a 'Login' button. Below the login form is a 'Register' button. A red box highlights a '忘记密码' (Forgot Password) link at the bottom left of the form.

再输入学号后



即会发送验证码至绑定的企业邮箱中，将验证码输入后进行验证，通过即可直接修改密码



### 3、前端管理员端登陆部分

该端考虑到可能会有部分管理员同时管理多个社团，因此为了方便管理，我们会在验证管理员账号密码后，从后端获取该管理员管辖的所有社团编号及名字并由用户选择：

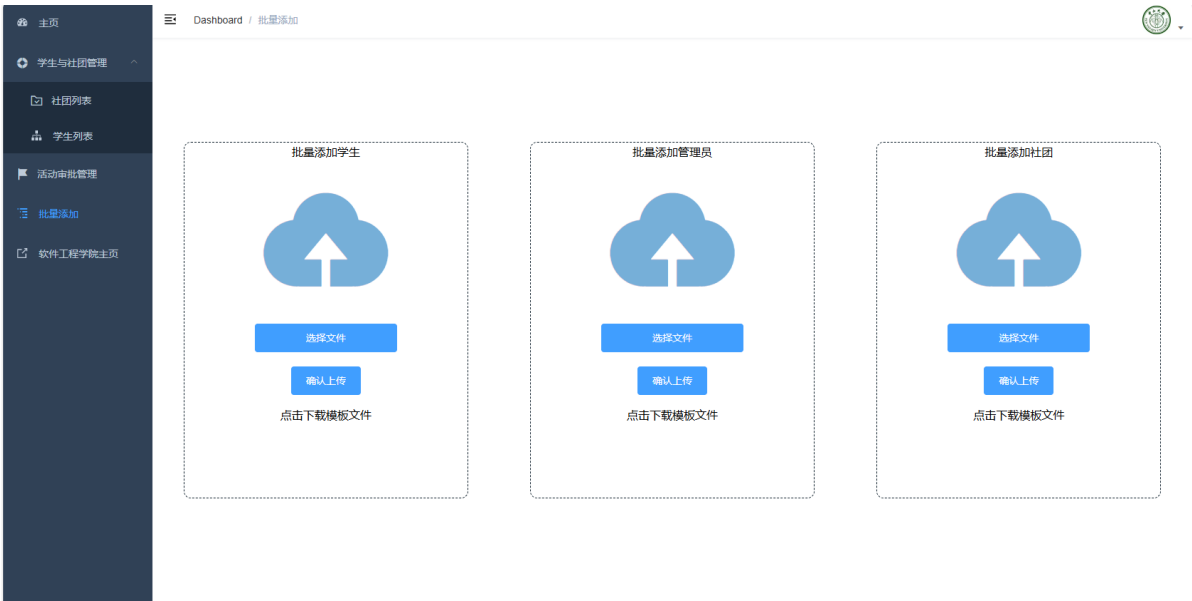




在用户选择希望选择的社团之后系统会将相应的社团id传入store中（this.\$store.state.clubid）以便后续调用。

#### 4、前端超级管理员端批量添加部分

在此处可以选择文件上传，前端会限制只能上传excel文件，且一次只能上传一个文件，上传文件之后会将其传给后端，后端进行excel自动化识别并录入数据库。点击下载模板文件后，会调用后端接口，下载模板文件，以方便用户使用批量导入时的使用正确。



#### 二、负责人：黎谨华-21311608

负责部分：关于这次项目，我主要负责的是前端超管端和学生端的部分界面的构建以及后端测试集的构建编辑，构建的界面有：超管端最火社团的构建、学生端首页可视化的构建以及学生端我的社团界面的构建，后端构建了所有的测试集。

主要内容：

- 采用了Vue.js构建了三个界面，其中使用e-charts对学生端首页进行动态的可视化显示
- 通过相似的代码模板完成所有数据库链接测试集的构建，以达到代码的泛用性和规划性
- 使用postcode向后端的接口输入和传输数据
- 辅助后端组员完善后端的传输接口
- 负责测试需求分析、测试用例编写、测试执行
- 编写测试计划、测试报告、操作手册，对Bug进行跟踪管理
- 向老师、助教、组长和组员定期汇报项目进展情况
- 和组员密切讨论交流，合作完成需求分析、接口规约、编码和测试工作

项目心得：

在本次项目中，我主要负责前端超管端和学生端的部分界面构建以及后端测试集的构建编辑工作，通过这段时间的努力和实践，我收获颇丰。

在前端界面构建方面，我采用 Vue.js 技术构建了超管端最火社团、学生端首页可视化以及学生端我的社团等界面。其中，在学生端首页可视化的构建中，使用 e-charts 实现动态的可视化显示是一个具有挑战性的任务。为了实现这一功能，我需要深入了解 e-charts 的各种配置和数据交互方式，不断调试和优化参数，以确保数据的准确展示和良好的用户体验。在这个过程中，我不仅提升了自己对于前端可视化技术的掌握能力，还深刻体会到了数据可视化对于提升用户交互和信息传递的重要性。

而后端测试集构建编辑同样具有重要意义。通过相似的代码模板完成数据库链接测试集的构建，旨在实现代码的泛用性和规划性。这要求我在设计测试集时，充分考虑各种可能的场景和边界情况，以确保系统的稳定性和可靠性。在这个过程中，我对数据库的操作和数据交互有了更深入的理解，也学会了如何从整体上规划和设计测试用例，以提高测试的覆盖率和效率。

在数据传输方面，使用 postcode 向后端的接口输入和传输数据，让我了解到前后端数据交互的关键技术和要点。在与后端组员的合作中，辅助他们完善后端的传输接口，使我明白了团队协作和沟通在项目中的重要性。我们相互学习、相互支持，共同解决了许多技术难题，这种团队合作的经验让我深刻认识到，一个项目的成功离不开团队成员之间的密切配合和相互信任。

在测试工作中，从测试需求分析、测试用例编写、测试执行，到编写测试计划、测试报告、操作手册，再到对 Bug 进行跟踪管理，每一个环节都需要严谨和细致。通过这些工作，我培养了自己的问题发现和解决能力，也学会了如何系统地对项目进行质量把控。

在项目进行过程中，定期向老师、助教、组长和组员汇报项目进展情况也是我的重要工作之一。这不仅有助于让团队成员了解项目的整体状态，及时调整策略和计划，也锻炼了我的沟通和表达能力。

总的来说，这次项目让我在技术能力、团队协作、问题解决和项目管理等方面都得到了全面的锻炼和提升。通过与组员的密切讨论交流，我们共同完成了需求分析、接口规约、编码和测试工作，每一个阶段都充满了挑战和收获。我相信，这次项目积累的经验和知识，将对我未来的学习和工作产生深远的影响，也让我更加期待在未来的项目中，能够不断突破自己，取得更大的进步。

### 三、负责人：吕杰-21311366

负责部分：主要负责使用SpringBoot架构完成对项目账户管理、信息维护、年检与审批、申请处理、数据批量管理、日志与流量控制等功能的后端开发。

主要完成的功能：

1. 完成了项目后端大部分基础应用功能的开发，总开发接口数约为100个。
2. 在综合账户管理部分，完成了包括校团委、管理端和学生端的登录功能，账户信息的编辑和管理，以及管理者权限转让等功能；
3. 在社团信息维护部分，完成了社团基本信息的管理，包括成员信息、社团活动资料等。
4. 在活动信息维护部分，完成了活动基本信息的管理，包括参与成员信息、负责主办社团等。
5. 在年检与审批部分，完成了社团年检审批功能，社团管理员提交年检资料，校团委对社团年检进行审核、批准或反馈。
6. 在申请处理部分，完成了新建社团申请处理、新建活动申请处理、加入社团申请处理、参加活动申请处理等功能。

特色部分：

1. 使用SpringBoot默认的日志框架 logback进行日志编写与保存，通过logback-spring.xml对日志配置进行设置，可以在日志中查看执行结果、运行时间、报错处理等。日志按日期保存到文件中，指定保存路径为“项目地址/club\_springboot/log/”，可以保存30天，日志文件不超过100M。

日志命名形式：



部分日志形式：

```
19 03:08:30.342 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - ----- 开 123 ^ v
20 03:08:30.342 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - 请求地址: http://localhost:8081/student/login POST
21 03:08:30.343 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - 类名方法: com.chinahitech.shop.controller.StuController.login
22 03:08:30.343 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - 远程地址: 0:0:0:0:0:0:1
23 03:08:30.384 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - 请求参数: ["21311366"]
24 03:08:30.919 logback-spring [http-nio-8081-exec-4] INFO com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Starting...
25 03:08:31.406 logback-spring [http-nio-8081-exec-4] INFO com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Start completed.
26 03:08:31.561 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - 返回结果: {"code":20000,"data":{"total":1}}
27 03:08:31.562 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - ----- 结束 耗时: 1222 ms -----
28 03:10:34.020 logback-spring [http-nio-8081-exec-5] INFO com.chinahitech.shop.aop.LogAspect - ----- 开始 -----
29 03:10:34.020 logback-spring [http-nio-8081-exec-5] INFO com.chinahitech.shop.aop.LogAspect - 请求地址: http://localhost:8081/student/login POST
30 03:10:34.020 logback-spring [http-nio-8081-exec-5] INFO com.chinahitech.shop.aop.LogAspect - 类名方法: com.chinahitech.shop.controller.StuController.login
31 03:10:34.020 logback-spring [http-nio-8081-exec-5] INFO com.chinahitech.shop.aop.LogAspect - 远程地址: 0:0:0:0:0:0:1
32 03:10:34.020 logback-spring [http-nio-8081-exec-5] INFO com.chinahitech.shop.aop.LogAspect - 请求参数: ["21311366"]
33 03:10:34.029 logback-spring [http-nio-8081-exec-5] INFO com.chinahitech.shop.aop.LogAspect - 返回结果: {"code":20000,"data":{"total":1}}
34 03:10:34.029 logback-spring [http-nio-8081-exec-5] INFO com.chinahitech.shop.aop.LogAspect - ----- 结束 耗时: 9 ms -----
35 03:10:35.130 logback-spring [http-nio-8081-exec-6] INFO com.chinahitech.shop.aop.LogAspect - ----- 开始 -----
36 03:10:35.132 logback-spring [http-nio-8081-exec-6] INFO com.chinahitech.shop.aop.LogAspect - 请求地址: http://localhost:8081/student/login POST
37 03:10:35.130 logback-spring [http-nio-8081-exec-6] INFO com.chinahitech.shop.aop.LogAspect - 类名方法: com.chinahitech.shop.controller.StuController.login
38 03:10:35.130 logback-spring [http-nio-8081-exec-6] INFO com.chinahitech.shop.aop.LogAspect - 远程地址: 0:0:0:0:0:0:1
39 03:10:35.130 logback-spring [http-nio-8081-exec-6] INFO com.chinahitech.shop.aop.LogAspect - 请求参数: ["21311366"]
40 03:10:35.132 logback-spring [http-nio-8081-exec-6] INFO com.chinahitech.shop.aop.LogAspect - 返回结果: {"code":20000,"data":{"total":1}}
41 03:10:35.132 logback-spring [http-nio-8081-exec-6] INFO com.chinahitech.shop.aop.LogAspect - ----- 结束 耗时: 2 ms -----
42 03:14:25.366 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - ----- 开始 -----
43 03:14:25.366 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - 请求地址: http://localhost:8081/student/login POST
44 03:14:25.366 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - 类名方法: com.chinahitech.shop.controller.StuController.login
45 03:14:25.366 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - 远程地址: 0:0:0:0:0:0:1
46 03:14:25.366 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - 请求参数: ["21311366"]
47 03:14:25.373 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - 返回结果: {"code":20000,"data":{"total":1}}
48 03:14:25.373 logback-spring [http-nio-8081-exec-4] INFO com.chinahitech.shop.aop.LogAspect - ----- 结束 耗时: 7 ms -----
```

2. 使用redis的原子操作进行流量控制，设置请求锁定时间为10s，时间内请求上限为100。若申请请求数超过上限，会提示“当前服务繁忙，请稍后再试”。

提示示例：

```
ExceptionHandlerExceptionResolver : Using @ExceptionHandler com.chinahitech.shop.exception.GlobalExceptionHandler
estResponseBodyMethodProcessor : Using 'application/json', given [*/] and supported [application/json]
estResponseBodyMethodProcessor : Writing [com.chinahitech.shop.utils.Result@998288c]
ExceptionHandlerExceptionResolver : Resolved [java.lang.RuntimeException: 当前服务繁忙，请稍后再试]
```

3. 使用redis统一对返回的报错结果进行处理，将后端报错结果返回前端。

报错示例：

```
[nio-8081-exec-6] com.chinahitech.shop.aop.LogAspect : 请求地址: http://localhost:8081/student/login POST
[nio-8081-exec-6] com.chinahitech.shop.aop.LogAspect : 类名方法: com.chinahitech.shop.controller.StuController.login
[nio-8081-exec-6] com.chinahitech.shop.aop.LogAspect : 远程地址: 0:0:0:0:0:0:1
[nio-8081-exec-6] com.chinahitech.shop.aop.LogAspect : 请求参数: ["21311366","123456"]
[nio-8081-exec-6] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
[nio-8081-exec-6] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
[nio-8081-exec-6] c.c.shop.mapper.StuMapper.getByNum : ==> Preparing: SELECT * FROM user WHERE user_id = ?
[nio-8081-exec-6] c.c.shop.mapper.StuMapper.getByNum : ==> Parameters: 21311366(String)
[nio-8081-exec-6] c.c.shop.mapper.StuMapper.getByNum : <== Total: 1
[nio-8081-exec-6] com.chinahitech.shop.aop.LogAspect : 返回结果: {"code":20001,"data":{"total":1},"message":"密码不正确","success":false}
[nio-8081-exec-6] com.chinahitech.shop.aop.LogAspect : ----- 结束 耗时: 1070 ms -----
```

4. 使用对excel表头关键字的识别，能够识别特定表头对应的用户数据，完成用户批量导入功能。

使用示例：

```
INFO 9940 --- [nio-8081-exec-2] com.chinahitech.shop.aop.LogAspect : 类名方法: com.chinahitech.shop.controller.TopManagerController.uploadExcel
INFO 9940 --- [nio-8081-exec-2] com.chinahitech.shop.aop.LogAspect : 远程地址: 0:0:0:0:0:0:1
INFO 9940 --- [nio-8081-exec-2] com.chinahitech.shop.aop.LogAspect : 请求参数: [null]
INFO 9940 --- [nio-8081-exec-2] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
INFO 9940 --- [nio-8081-exec-2] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
DEBUG 9940 --- [nio-8081-exec-2] c.c.s.m.TopManagerMapper.getUserByNum : ==> Preparing: SELECT * FROM user WHERE user_id = ?
DEBUG 9940 --- [nio-8081-exec-2] c.c.s.m.TopManagerMapper.getUserByNum : ==> Parameters: 21313000(String)
DEBUG 9940 --- [nio-8081-exec-2] c.c.s.m.TopManagerMapper.getUserByNum : <== Total: 0
DEBUG 9940 --- [nio-8081-exec-2] c.c.shop.mapper.TopManagerMapper.insert : ==> Preparing: INSERT INTO user ( user_id, password, user_name, status, sa
DEBUG 9940 --- [nio-8081-exec-2] c.c.shop.mapper.TopManagerMapper.insert : ==> Parameters: 21313000(String), CA5FFADE12F18F87F08BFEC1AA332582(String),
DEBUG 9940 --- [nio-8081-exec-2] c.c.shop.mapper.TopManagerMapper.insert : <== Updates: 1
INFO 9940 --- [nio-8081-exec-2] com.chinahitech.shop.ShopApplication : 插入用户21313000成功
DEBUG 9940 --- [nio-8081-exec-2] c.c.s.m.TopManagerMapper.getUserByNum : ==> Preparing: SELECT * FROM user WHERE user_id = ?
DEBUG 9940 --- [nio-8081-exec-2] c.c.s.m.TopManagerMapper.getUserByNum : ==> Parameters: 21313010(String)
DEBUG 9940 --- [nio-8081-exec-2] c.c.s.m.TopManagerMapper.getUserByNum : <== Total: 0
DEBUG 9940 --- [nio-8081-exec-2] c.c.shop.mapper.TopManagerMapper.insert : ==> Preparing: INSERT INTO user ( user_id, password, user_name, status, sa
DEBUG 9940 --- [nio-8081-exec-2] c.c.shop.mapper.TopManagerMapper.insert : ==> Parameters: 21313010(String), E37A66658C26F34A40FE02001F315201(String),
DEBUG 9940 --- [nio-8081-exec-2] c.c.shop.mapper.TopManagerMapper.insert : <== Updates: 1
INFO 9940 --- [nio-8081-exec-2] com.chinahitech.shop.ShopApplication : 插入用户21313010成功
DEBUG 9940 --- [nio-8081-exec-2] c.c.s.m.TopManagerMapper.getUserByNum : ==> Preparing: SELECT * FROM user WHERE user_id = ?
DEBUG 9940 --- [nio-8081-exec-2] c.c.s.m.TopManagerMapper.getUserByNum : ==> Parameters: 21313020(String)
DEBUG 9940 --- [nio-8081-exec-2] c.c.s.m.TopManagerMapper.getUserByNum : <== Total: 0
DEBUG 9940 --- [nio-8081-exec-2] c.c.shop.mapper.TopManagerMapper.insert : ==> Preparing: INSERT INTO user ( user_id, password, user_name, status, sa
DEBUG 9940 --- [nio-8081-exec-2] c.c.shop.mapper.TopManagerMapper.insert : ==> Parameters: 21313020(String), F650FFC2F5237B8CBBE57C0F53072E8E(String),
DEBUG 9940 --- [nio-8081-exec-2] c.c.shop.mapper.TopManagerMapper.insert : <== Updates: 1
INFO 9940 --- [nio-8081-exec-2] com.chinahitech.shop.ShopApplication : 插入用户21313020成功
INFO 9940 --- [nio-8081-exec-2] com.chinahitech.shop.aop.LogAspect : 返回结果: {"body":{"fileUrl":"http://localhost:8081/upload/20240712213627396
INFO 9940 --- [nio-8081-exec-2] com.chinahitech.shop.aop.LogAspect : ----- 结束 耗时: 1048 ms -----
```

5. 使用md5算法对密码进行加密。

密码储存样式示例：

ABC phone	ABC password	ABC campus
[NULL]	9AA4F6A4BB9F1D1D932AEBA70AFE052	[NULL]
[NULL]	3F90481BFFA9AAECEFE84498D2D60F04	[NULL]
[NULL]	16E6B7A21EE73A744524AC59C162D17	[NULL]
	D7C9F17B47F39B3994233F517323CF43	[NULL]

## 四、负责人：陈天蔚-21311609

负责部分：实现前端的活动、社团界面与对应申请界面，学生列表及各个学生详情界面并修复其他前端界面的bug。

以制作活动界面为例，活动界面需要展示所有的活动以及提供搜索功能，且显示活动的功能需要一进入界面就运行，因此需要使用created钩子函数尽快将数据传入，因为created钩子函数在组件初始化之前就可以获得数据，再将数据传入实例中。展示活动的组件使用了el-row行容器与el-col列容器，通过限制容器内每个元素的网格大小来显示多个活动，组件的数据绑定实例，因此组件初始化时就可以得到数据并显示全部活动。而搜索功能的实现中按钮点击事件绑定了api中的搜索函数，通过路由与后端相接，在后端返回搜索数据后组件根据返回元素数量调整组件显示数量与实例数组元素，避免重复显示同一活动。

每一个活动元素中都绑定了图片，文本与按钮等组件，每个元素都有两个按钮，一个用于查看详情，点击事件会切换一个小窗的可视状态，小窗包含该活动的详细信息，另一个用于申请加入，点击事件绑定跳转界面的函数，通过路由进行切换。

申请界面中，由于使用该系统时需要登录，因此申请表的信息只有自我陈述以及上传文件可以填写，其他信息都根据账号信息即store实例与钩子函数所填写，上传文件功能使用el-upload上传组件实现，在该组件中设置了上传位置、上传时各种情况会返回的信息、对文件的检查以及文件的限制，最后完成申请按钮再将上传信息打印到控制台。社团及申请界面与以上内容大致相同。

学生显示界面则使用了两种方式，一种是管理员端的列表形式，一种是超级管理员端的树状图，使用两种不同的形式原因是超级管理员端需要分社团管理，普通管理员则没有这种分类需求。

列表形式中，使用了el-table组件，且设置每一行都有点击事件，点击时进入详情界面，初始数据由store实例与钩子函数所填写，el-table组件与数组变量绑定并设置好表头即可显示数组中需要显示的内容，其中显示申请日期使用了Date类型，标准化时间数据。

点击事件中获取点击行的数据传入跳转界面中。

学生详情界面则使用表格显示，初始数据使用传入数据提供的id通过查询函数赋值给实例，表格内容通过事先设置的行数、表头以及绑定实例数据展示。

树状图形式中，使用了el-tree组件，使用该组件最大的问题就是该组件绑定的数据要求是树形结构，而后端是难以返回树形结构给前端的，这要求前端自己将后端返回的数组数据处理成树形结构，这里我使用了arrayTotree的包实现数据处理，将两个数组合并成一个树形结构要求这两个数组的成员拥有同一类型且没有重复的元素即两个数组拥有同一个主键，因此需要使用循环给两个数组都设置一个index作为两个数组的主键，又为了将两个数组合并成树形结构，则将group数组的index设置成与groupId一致，student数组则往后递增就好，最后合并时设置index为每个节点的主键，student数组里的成员成为子节点，父节点根据groupId的值选取，因为group数组与student数组元素不同，设置显示内容时可根据元素内容区分父、子节点显示不同内容，其余设置如初始是否显示子节点、点击节点是否可以缩放都可以按需求更改。

最后的工作为寻找bug并修复，如文字内容不显示、切换界面出现报错信息、图标显示错误等等，问题出现原因有数据绑定没有绑定该界面实例、参数名与传参函数所用参数名不同、图标绑定死一个图标，设置变量更改图标也无法更改、绑定空函数等等。

## 五、负责人：谢靖宁-21307351

### 个人职责: 前端开发、接口测试

- 采用 Vue.js 和 Element 开发了学生端、管理员端和超级管理员端的部分页面
- 高质量完成代码编写确保符合团队规范，高还原度还原视觉稿
- 对 WEB 页面进行性能优化，打造良好的用户体验
- 配合后端负责人完成网站开发
- 使用 Postcode 对后端的提供的接口进行测试
- 负责测试需求分析、测试用例编写、测试执行
- 编写测试计划、测试报告、操作手册，对Bug进行跟踪管理
- 向老师、助教、组长和组员定期汇报项目进展情况
- 和组员密切讨论交流，合作完成需求分析、接口规约、编码和测试工作

### 个人总结

本次实训尝试的从未了解过的前端开发，从 html / JavaScript / css 三件套起手，再到 Vue.js 框架，我深刻体会到这个框架的设计思想是一种“渐进式框架”，淡化框架本身的主张，降低框架作为工具的复杂度，从而降低对使用者的要求。

相比于 Angular 和 React，Vue 的设计哲学大道至简，先让开发者注意到核心的部分，然后按照实际需求在此基础上拓展，利用社区良好的生态，借助已有的工具和库搭建我们的项目，用最小的成本、最快的时间一步步搭建。

有待改善的地方在于此项目也是在一个流产的项目的基础上开发的：

1. 考虑到时间和精力成本，无暇推倒重来，不得不带着一些遗留问题继续开发，实际上这和生产环境的情形相似，反而恰好提升了我们上手维护和再开发一个现有项目的能力。

2. 我们组 UI 设计相关的技能比较缺少，而网络上的模板又弥补了这一点，如果直接套模板，就会显得项目比较大众化，会出现千篇一律的现象，所以现在最好的办法就是，在套模板的同时，给我们的项目做一个 UI 重构。
3. 在接口测试方面，无暇学习业界先进的自动化测试框架，也没有充分学习消化和使用 Postman 的高级功能，网站无法保证鲁棒性
4. 技术栈方面，没有参与用 SpringBoot 的后端开发，也没有落地自动化部署，没有充分利用实训课程提供的各种资源，这对于小组和项目来说是一种遗憾

## 部分代码展示

### 404 小游戏页面逻辑代码

```
update: function () {
    this.updatePending = false;

    var now = getTimestamp();
    var deltaTime = now - (this.time || now);
    this.time = now;

    if (this.playing) {
        this.clearCanvas();

        if (this.tRex.jumping) {
            this.tRex.updateJump(deltaTime);
        }

        this.runningTime += deltaTime;
        var hasObstacles = this.runningTime > this.config.CLEAR_TIME;

        if (this.tRex.jumpCount == 1 && !this.playingIntro) {
            this.playIntro();
        }

        if (this.playingIntro) {
            this.horizon.update(0, this.currentSpeed, hasObstacles);
        } else {
            deltaTime = !this.activated ? 0 : deltaTime;
            this.horizon.update(deltaTime, this.currentSpeed, hasObstacles,
                this.inverted);
        }

        var collision = hasObstacles &&
            checkForCollision(this.horizon.obstacles[0], this.tRex);

        if (!collision) {
            this.distanceRan += this.currentSpeed * deltaTime / this.msPerFrame;

            if (this.currentSpeed < this.config.MAX_SPEED) {
                this.currentSpeed += this.config.ACCELERATION;
            }
        } else {
            this.gameOver();
        }
    }
}
```

```

        var playAchievementSound = this.distanceMeter.update(deltaTime,
            Math.ceil(this.distanceRan));

        if (playAchievementSound) {
            this.playSound(this.soundFx.SCORE);
        }

        if (this.invertTimer > this.config.INVERT_FADE_DURATION) {
            this.invertTimer = 0;
            this.invertTrigger = false;
            this.invert();
        } else if (this.invertTimer) {
            this.invertTimer += deltaTime;
        } else {
            var actualDistance =

this.distanceMeter.getActualDistance(Math.ceil(this.distanceRan));

            if (actualDistance > 0) {
                this.invertTrigger = !(actualDistance %
                    this.config.INVERT_DISTANCE);

                if (this.invertTrigger && this.invertTimer === 0) {
                    this.invertTimer += deltaTime;
                    this.invert();
                }
            }
        }
    },

    if (this.playing || (!this.activated &&
        this.tRex.blinkCount < Runner.config.MAX_BLINK_COUNT)) {
        this.tRex.update(deltaTime);
        this.scheduleNextUpdate();
    }
},

onKeyDown: function (e) {
    if (IS_MOBILE && this.playing) {
        e.preventDefault();
    }

    if (e.target !== this.detailsButton) {
        if (!this.crashed && (Runner.keycodes.JUMP[e.keyCode] ||
            e.type === Runner.events.TOUCHSTART)) {
            if (!this.playing) {
                this.loadSounds();
                this.playing = true;
                this.update();
                if (window.errorPageController) {
                    errorPageController.trackEasterEgg();
                }
            }
            if (!this.tRex.jumping && !this.tRex.ducking) {
                this.playSound(this.soundFx.BUTTON_PRESS);
                this.tRex.startJump(this.currentSpeed);
            }
        }
    }
}

```



```

    }
}

if (this.crashed && e.type == Runner.events.TOUCHSTART &&
    e.currentTarget == this.containerEl) {
    this.restart();
}

}

if (this.playing && !this.crashed && Runner.keycodes.DUCK[e.keyCode]) {
    e.preventDefault();
    if (this.tRex.jumping) {
        this.tRex.setSpeedDrop();
    } else if (!this.tRex.jumping && !this.tRex.ducking) {
        this.tRex.setDuck(true);
    }
}

},
onKeyUp: function (e) {
    var keyCode = String(e.keyCode);
    var isjumpkey = Runner.keycodes.JUMP[keyCode] ||
        e.type == Runner.events.TOUCHEND ||
        e.type == Runner.events.MOUSEDOWN;

    if (this.isRunning() && isjumpkey) {
        this.tRex.endJump();
    } else if (Runner.keycodes.DUCK[keyCode]) {
        this.tRex.speedDrop = false;
        this.tRex.setDuck(false);
    } else if (this.crashed) {
        var deltaTime = getTimestamp() - this.time;

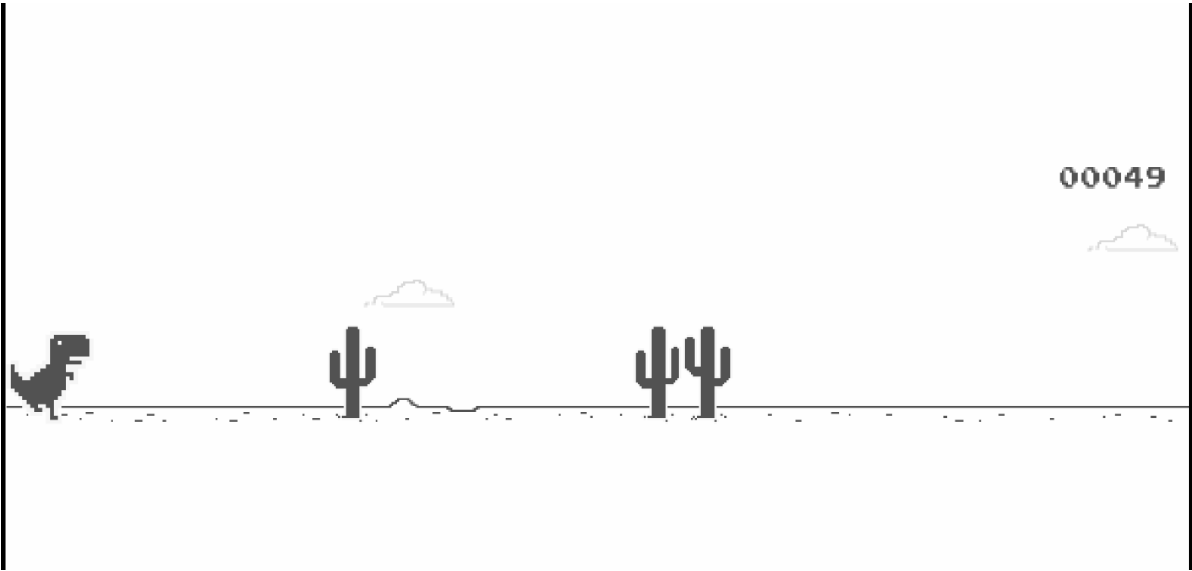
        if (Runner.keycodes.RESTART[keyCode] || this.isLeftClickOnCanvas(e) ||
            (deltaTime >= this.config.GAMEOVER_CLEAR_TIME &&
                Runner.keycodes.JUMP[keyCode])) {
            this.restart();
        }
    } else if (this.paused && isjumpkey) {
        this.tRex.reset();
        this.play();
    }
},

```

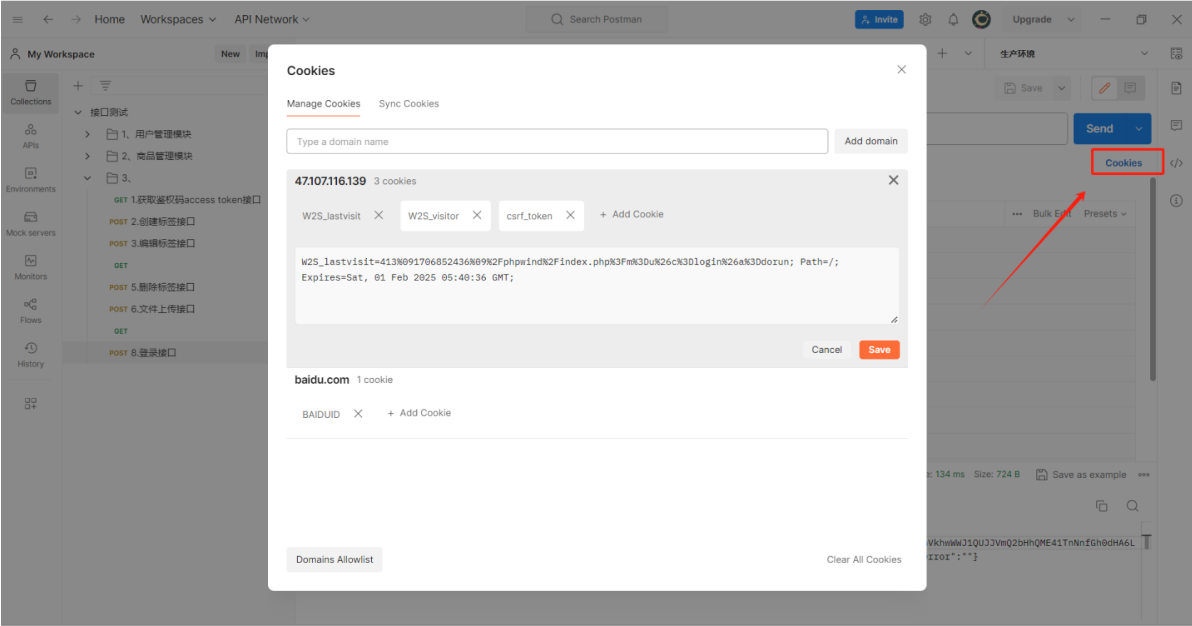
## 部分截图展示

404 页面小游戏截图





Postman 接口测试过程截图



## 结论及展望

“数字百团”社团管理系统在实际应用中取得了显著的效果。它极大地提高了社团管理的效率，使得社团信息的更新、活动组织的安排以及成员招募的流程都变得更加便捷和高效。同时，系统的透明度也得到了提升，学生可以随时了解到社团的最新动态和活动信息。此外，系统的易用性和丰富性也得到了广大用户的认可和好评。

综上所述，“数字百团”社团管理系统项目的成功实施为高校社团管理带来了新的变革。它不仅提高了管理效率，还提升了学生的参与度和满意度。展望未来，我们将继续完善系统的功能，优化用户体验，并探索更多创新的应用场景，以为高校社团管理的发展贡献更大的力量。