



浙大城市学院

ZHEJIANG UNIVERSITY CITY COLLEGE

社团管理系统-系统报告

课 程： 软件架构与应用开发

题 目： 社团管理系统

专业班级： 软件工程 1802

组 员： 童峻涛-31801341

组 员： 庄博伟-31801338

文件状态：	文件标识：	社团管理系统-系统报告
<input type="checkbox"/> 草稿	作者：	童峻涛、庄博伟
<input type="checkbox"/> 正式发布	完成日期：	2021-06-14
<input checked="" type="checkbox"/> 正在修改		

目录

1 系统概述	3
2 系统设计	3
2.1 系统方案.....	3
2.2 概要设计.....	4
2.2.1 系统结构图	4
2.2.2 管理员	4
2.2.3 学生	5
2.2.4 社长.....	5
2.2.5 社员	6
2.2.6 部长.....	6
2.3 详细设计.....	7
2.3.1 数据流图	7
2.3.2 模块流程图	8
2.4 数据库设计	8
2.4.1 表结构	8
2.4.2 E-R 图.....	9
3 实现步骤	9
3.1 后端实现.....	10
3.1.1 Mybatis 示例	10
3.1.2 Mysql 数据库.....	12
3.2 前端实现.....	13
3.2.1 制作流程	13
3.2.2 主要模块	13
3.2.3 主要工作	13

1 系统概述

学校社团丰富，参加社团的学生也很多，但是没有一个高效的管理系统来帮助指导老师，社长，部长来更好的管理社团，以及帮助普通学生去了解不同的社团，并加入其中丰富自己的课余生活。

该系统是以便利社团管理者对社团的管理为目标的项目，该系统集社团管理、社团纳新、活动组织申报等功能于一体，为社团的管理者与学生提供了一个数字化管理平台。

2 系统设计

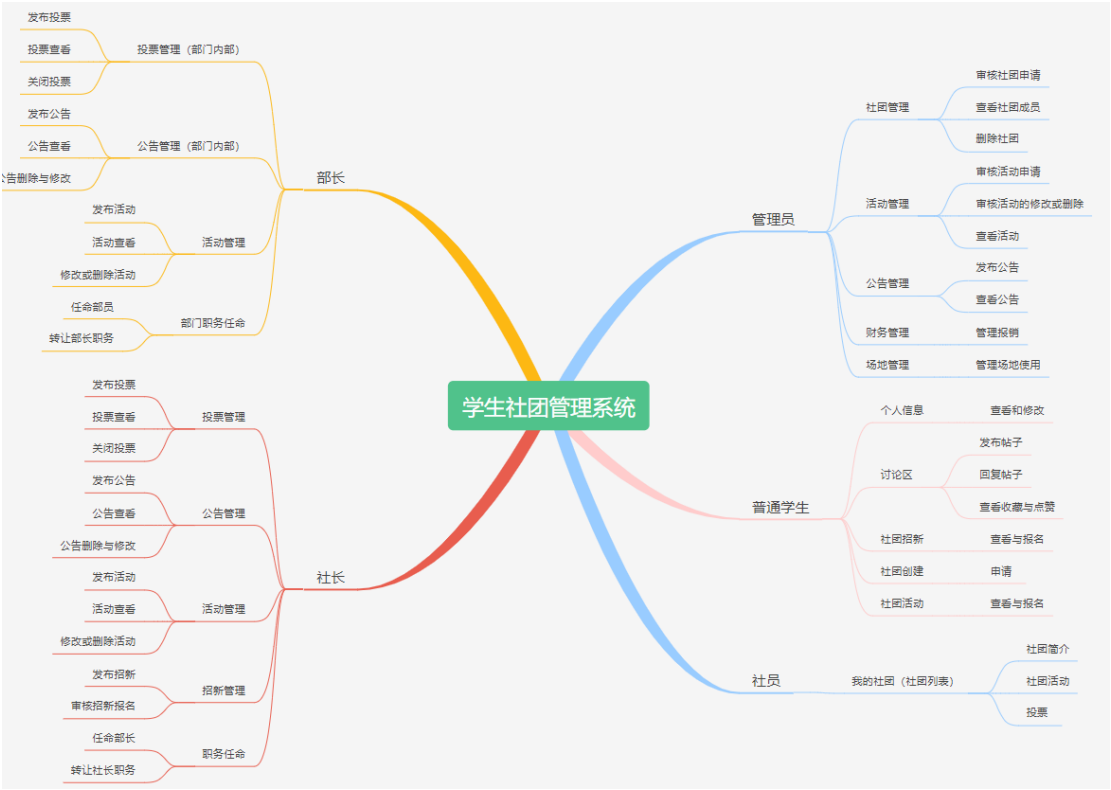
2.1 系统方案

后端：Springboot + Mybatis + JPA + Mysql

前端：微信小程序采用了微信自带的开发者工具，Web 端采用了 Vue.js 这个非常流行的组件式前端框架，且 Elementui 对 vue 的支持非常的好，可以节约很多的时间，在跨域传输技术中采用了 axios 这是官方推荐的。

2.2 概要设计

2.2.1 系统结构图



主要分为部长、社长、社员、普通学生以及管理员五个角色，每个角色拥有自己的功能。

2.2.2 管理员

1. 社团的管理

- a. 审核社团申请：收到学生发来的社团创立申请，进行审核，同意或拒绝，拒绝后给出拒绝的理由
- b. 查看社团及成员：可以看到所有的社团以及社员的信息
- c. 删除社团：管理员可以删除社团，同时级联的删除了所有社员的信息

2. 活动的管理

- a. 审核活动申请：收到社长或部长发来的活动申请，进行审核，同意或拒绝，拒绝后给出拒绝的理由(例如时间和场地的冲突)
- b. 审核活动的修改或删除：收到各个社团的修改或删除申请,同意或拒绝，拒绝后给出拒绝的理由
- c. 查看所有的活动

3. 公告的管理

- a. 审核公告申请：收到社长或部长发来的公告申请，进行审核，同意或拒绝，拒绝后给出拒绝的理由

b.审核公告的修改或删除：收到各个社团的修改或删除申请,同意或拒绝，拒绝后给出拒绝的理由

c.查看所有的公告

d.发布公告：发布的公告在主页上可以被本系统所有成员看到，例如系统维护等

4.场地的管理

a.场地的添加和删除：管理员可以管理可用的活动场地，随时进行增删改查

5.资金的管理

2.2.3 学生

1.个人信息管理

a.注册功能：姓名，学号,密码,性别，专业，班级，联系方式，头像

b.修改个人的信息：密码,专业，班级，联系方式，头像

2.社团创建

a.普通学生可以像管理员申请创建社团，并提交申请表，社团创建时需要至少五人，申请人为社长，并设置职位，申请表内容包括社团名字，社团描述(干什么),社团地点，申请时间，社团类型[学术类，体育类，艺术类，公益类，科技类

3.帖子管理

a.发布主题帖：主题帖包括标题，内容，发布时间

b.回复帖子：可以回复自己和别人的帖子，包括回复内容，回复时间，收藏，点赞

c.查看我收藏的和点赞的帖子

4.活动管理

a.查看社团招新并报名：学生可以在社团招新的界面中看到所有社团招新的内容，以及社团的名字,介绍和社长. 并且可以对心仪的社团报名，报名信息包括学生的个人信息和个人介绍

b.查看,报名社团普通活动(面向全校学生)：学生可以在普通活动界面看到所有社团面向所有学生的活动，并且报名

2.2.4 社长

社长拥有社员的所有功能

1.招新管理

a.发布招新活动：包括社团的名字，介绍，以及社长，可以自动生成

b.审核招新报名：有一个招新报名的单独界面，可以看到所有申请该社团的学生信息，同意或拒绝这些学生的报名，同意报名后,将普通的学生添加到社团中

2.活动管理

a.发布活动：可以选择活动是面向社团内还是面向全校，向管理员提交举办活动申请表，包括报名截止的时间,活动开始到结束时间，地点，活动描述，活动名称，面向全校或社团内部，报销资金

b.修改活动：向管理员申请活动内容的修改，说明理由

c.删除活动：向管理员申请活动的删除，并说明理由

3.公告管理

a.发布公告：先向管理员提交公告，等待管理员审核，审核通过后面向社团的社员发布公告。公告包括标题,内容，时间，选择面向普通学生，社团

b.修改公告：向管理员申请公告内容的修改，说明理由

c.删除公告：向管理员申请公告的删除，并说明理由

4.投票管理

a.发布投票：社长可以面向社团的社员发布投票，投票包括标题，内容，选项，是否匿名，投票开始到结束时间。投票完成后在投票界面显示投票结果

5.人员管理

a.可以查看社团所有的社员的信息

b.职务任命：可以任命社团内的某个社员为部长，或者任命某个部长为社长

2.2.5 社员

社员拥有普通学生的所有功能

1.活动管理

a.查看,报名社团普通活动(面向社员所在社团的所有社员)：学生可以在社团活动界面看到社员所在社团面向该社团所有社员的活动，并且报名

b.查看,报名社团部门内部活动(面向社员所在部门的所有社员)：学生可以在社团部门内部活动界面看到社员所在部门面向部门内部社员的活动，并且报名

2.投票管理

a.投票参与社长的投票(面向社员所在社团的所有社员)：学生可以在社团投票界面看到社员所在社团面向该社团所有社员的投票，并且投票

b.投票参与部长的投票(面向社员所在部门的所有社员)：学生可以在社团部门内部活动界面看到社员所在部门面向部门内部社员的投票，并且投票

2.2.6 部长

部长拥有社员所有的功能

1.活动管理(部门内)

a.发布活动：向管理员提交举办活动申请表，包括报名截止的时间,活动开始到结束时间，地点，活动描述，活动名称，报销资金

b.修改活动：向管理员申请活动内容的修改，说明理由

c.删除活动：向管理员申请活动的删除，并说明理由

2.公告管理(部门内)

a.发布公告：先向管理员提交公告，等待管理员审核，审核通过后面向部门内部的社员发布公告。公告包括标题,内容，时间

b.修改公告：向管理员申请公告内容的修改，说明理由

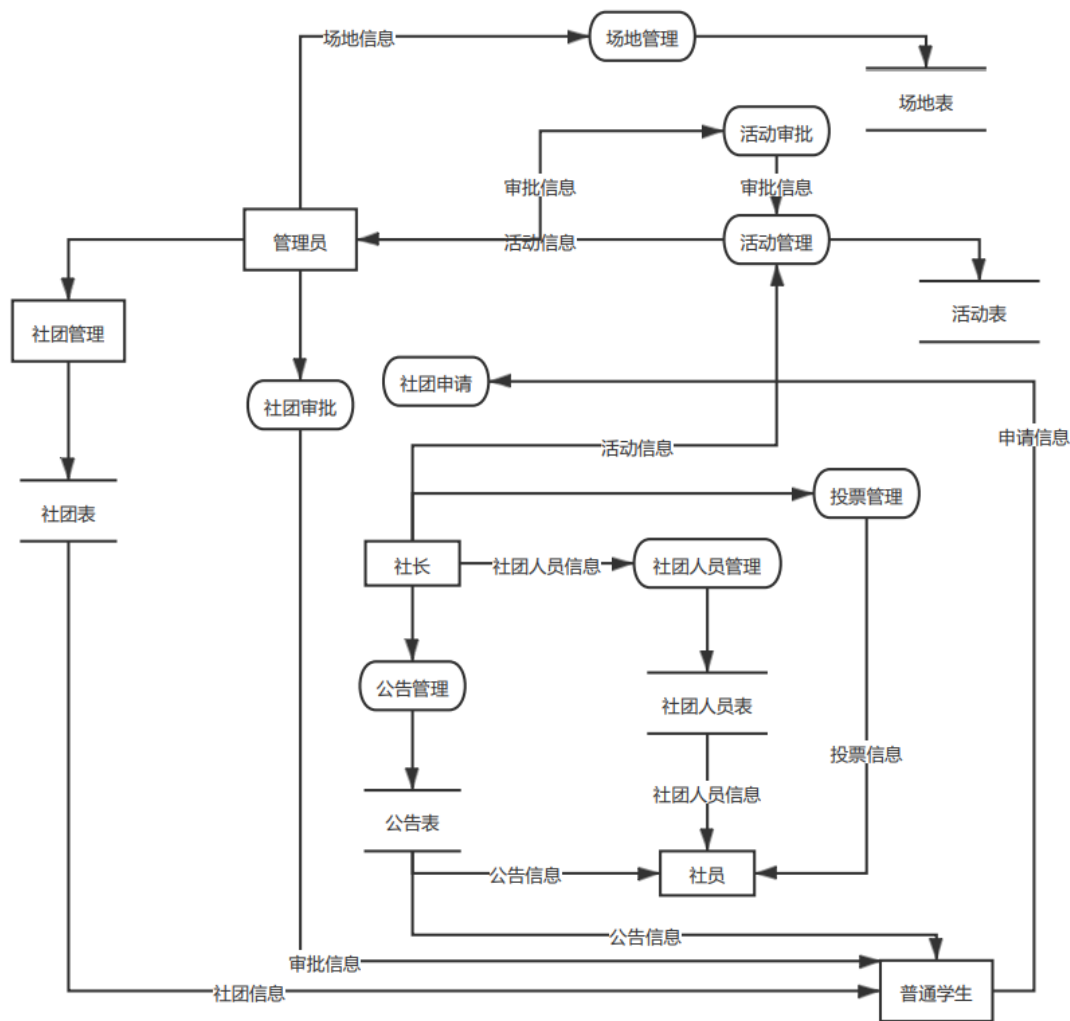
c.删除公告：向管理员申请公告的删除，并说明理由

3.投票管理(部门内)

a.发布投票：部长可以面向部门内部的社员发布投票，投票包括标题，内容，选项，是否匿名，投票开始到结束时间。投票完成后在投票界面显示投票结果

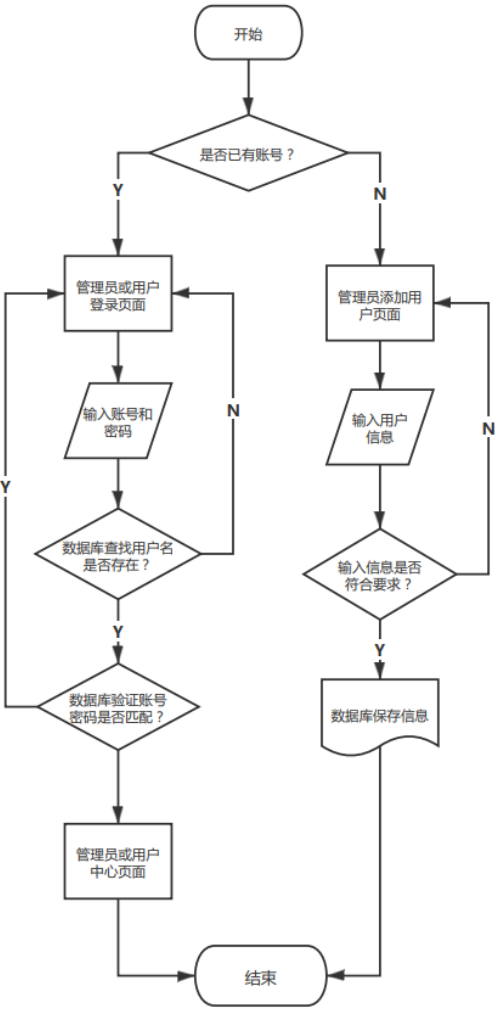
2.3 详细设计

2.3.1 数据流图



2.3.2 模块流程图

登录注册模块



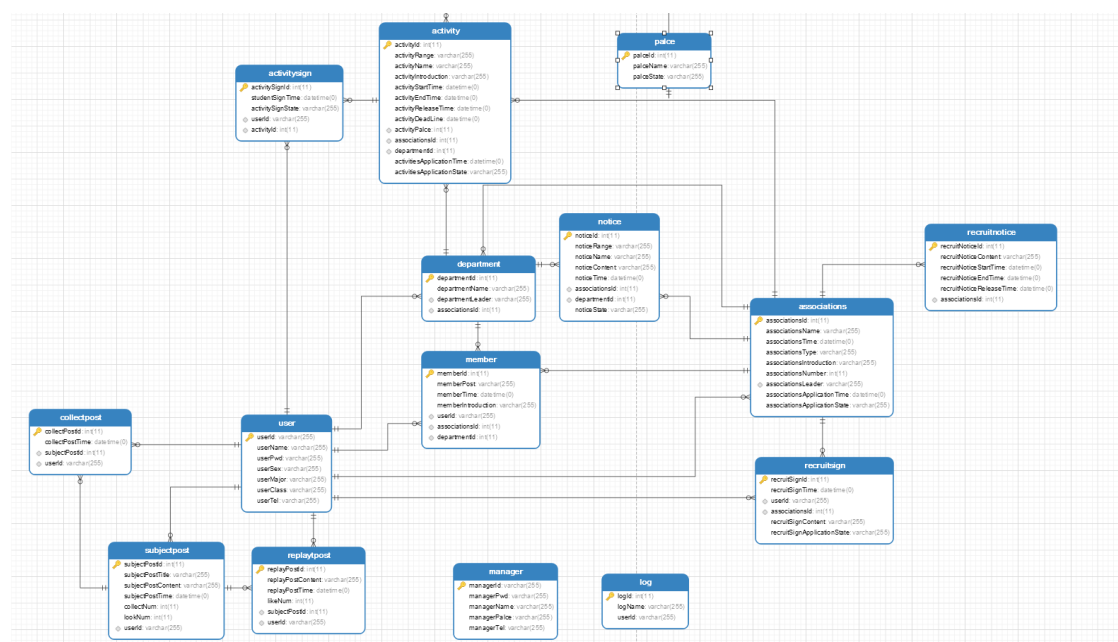
2.4 数据库设计

2.4.1 表结构

表名	表属性
用户	学生学号、名字、密码、性别、专业、班级、联系方式
管理员	管理员编号、名字、密码、办公地点、联系方式
社团	社团编号、名称、建立时间、社团类别、社团简介、人数、社长、地点
部门	部门编号、名称、部长、社团编号
日志	日志序号、时间、操作、操作人

活动	活动序号、范围（校内或社团内或部门内）、名称、起始时间、结束时间、活动发布时间、活动报名截止时间、场地编号、资金、社团编号、部门编号
活动报名	活动报名编号、活动报名时间、活动序号、学生学号
公告	公告序号、范围（校内或社团内或部门内）、内容、公告发布时间、社团编号、部门编号
社员	社员编号、职务、入社时间、个人介绍、学号、社团编号
社团申请	申请编号、申请内容、申请时间、申请状态、申请结果、学生学号
活动申请	申请编号、申请内容、申请时间、申请状态、申请结果、社团编号
招新公告	公告编号、内容、起始时间、结束时间、招新发布时间、社团编号
招新报名	招新报名编号、报名时间、个人介绍、社团编号、学生学号
场地	场地编号、地点

2.4.2 E-R 图



3 实现步骤

我们使用了 spring boot 和 vue 相结合的前后端实现框架。在后端部分，使用 springboot+JPA+Mysql 的组合。springboot 提供基于 Javabeans 的控制反转容器，而 JPA 是一种 Java 持久化规范，提供了对象关联工具，类似于 hibernate，使得

开发者能脱离 DAO 层的操作。在后端的实现中，我学会了上述工具/框架/规范的使用，在理解各层代码的基础上，写了社团和社团成员的部分数据处理代码，并将数据以 `alibaba.fastjson.JSONObject` 的格式返回给前端。

3.1 后端实现

3.1.1 Mybatis 示例

映射关系：

```
<sql id="Base_Column_List" >
    noticeId, noticeRange,noticeName, noticeContent, noticeTime,
associationsId,departmentId, noticeState
</sql>
<select id="selectAllNotice" resultMap="BaseResultMap">
    select *
    from notice
</select>
```

通用关系类：

Control 层：

```
//显示所有公告
@RequestMapping(value = "/allnotice", method = RequestMethod.GET)
public JSONObject showAllNotice() throws Exception {
    NoticeService noticeService = new NoticeService();
    String allnotice = JSON.toJSONString( noticeService.loadNotice());
    JSONArray array= JSONArray.parseArray(allnotice);
    JSONObject result = new JSONObject();
    result.put("allnotice",array);
    return result;
}
```

函数编写：

Service 层：

```
// 显示所有公告
public List<JSONObject> loadNotice() throws Exception {
    List<BeanNotice> noticeList = null;
    JSONObject jsonObject = null;
    List<JSONObject> list = new ArrayList<JSONObject>();
```

```

SqlSession session = MybatisSession.getSession();
noticeList = session.selectList("selectAllNotice");

for(int i=0;i<noticeList.size();i++) {
    jsonObject = JsonUtil.NoticeResult(200, noticeList.get(i));
    list.add(jsonObject);
}

session.commit();
return list;
}

```

工具接口 MybatisSession

方便 service 层调用 mapper 中的具体 sql 语句来进行数据库操作

```

package com.setu.Tool;

import org.apache.ibatis.io.Resources;
import org.apache.ibatis.session.SqlSession;
import org.apache.ibatis.session.SqlSessionFactory;
import org.apache.ibatis.session.SqlSessionFactoryBuilder;

import java.io.IOException;
import java.io.Reader;

public class MybatisSession {
    public static SqlSession getSession() {
        String resources = "mybatis.cfg.xml";
        Reader reader = null;
        SqlSession session = null;
        try {
            reader = Resources.getResourceAsReader(resources);
            SqlSessionFactory sqlMapper = new
SqlSessionFactoryBuilder().build(reader);
            session = sqlMapper.openSession();

        } catch (IOException e) {
            e.printStackTrace();
            return null;
        }
        return session;
    }
}

```

工具接口 JsonUtil

给前端返回 Json 格式的数据类型（端口号+Model 类/报错信息）

```
//公告
public static JSONObject NoticeResult(int port, BeanNotice notice){
    SqlSession session = MybatisSession.getSession();
    BeanAssociations a =
session.selectOne("selectAssociations",notice.getAssociationsid());
    JSONObject jsonObject = new JSONObject(true);
    jsonObject.put("port",port);
    jsonObject.put("noticeid",notice.getNoticeid());
    jsonObject.put("noticerange",notice.getNoticerange());
    jsonObject.put("noticename",notice.getNoticename());
    jsonObject.put("noticetcontent",notice.getNoticecontent());
    jsonObject.put("noticetime",df.format(notice.getNoticetime()));
    jsonObject.put("associationsid",notice.getAssociationsid());
//    jsonObject.put("associationsname",a.getAssociationsname());
    jsonObject.put("departmentid",notice.getDepartmentid());
    jsonObject.put("noticestate",notice.getNoticestate());
    return jsonObject;
}
```

Controller 层:

@RestController :

将当前修饰的类注入 SpringBoot IOC 容器，使得从该类所在的项目跑起来的过程中，这个类就被实例化。当然也有语义化的作用，即代表该类是充当 Controller 的作用

@CrossOrigin: 用来处理跨域请求的注解

比如在本项目中前端启用 8080 端口，后端启用 8880 端口，那么前端也可以访问到后端的服务。

@ResponseBody:

将 controller 的方法返回的对象，通过适当的转换器 转换为 json 格式之后，写入 response 对象的响应体。

通过@ResponseBody 注解将返回的 json 字符串放入响应体中，然后在前台 js 才能拿到 json 字符串进行解析

3.1.2 Mysql 数据库

建表语句:

详见 javaee.sql

javaee	activityName	activityIntroduction	activityStartTime	activityEndTime	activityReleaseTime	activi ^
表	下周打篮球比赛	来来来	2021-06-04 00:00:01	2021-06-08 00:00:00	2021-06-01 20:25:17	2021-
activity	篮球社开会	来	2021-06-04 00:00:01	2021-06-08 00:00:00	2021-06-01 20:17:18	2021-
activitysign	操场跑步	来	2021-06-04 00:00:01	2021-06-08 00:00:00	2021-06-01 20:19:52	2021-
associations	聚餐	聚餐,是容易促进社团成员	2021-06-10 00:00:01	2021-06-11 00:00:00	2021-06-01 23:18:24	2021-
collectpost	演讲	目的: 推广电竞活动。时间: 2021-06-28 00:00:01	2021-06-29 00:00:00	2021-06-01 23:20:12	2021-	2021-
department	爱心募捐	为山区留守儿童献爱心。	2021-06-04 00:00:01	2021-06-13 00:00:00	2021-06-03 15:09:39	2021-
log	花式篮球	花式篮球大赛开始啦, 奖励:	2021-06-06 00:00:01	2021-06-10 00:00:00	2021-06-03 15:24:19	2021-
manager	植树	绿化校园	2021-06-03 23:40:16	2021-06-10 15:40:23	2021-06-07 19:42:34	2021-
member	第一次心理讲座	三天后教3心理讲座, 欢迎:	2021-06-07 22:48:18	2021-06-11 22:48:20	2021-06-07 19:42:31	2021-
notice	篮球比赛	篮球比赛	2021-06-08 09:15:24	2021-06-12 09:15:06	2021-06-05 09:15:55	2021-
palce	第一次集会	快来呀	2021-06-24 08:53:12	2021-06-30 00:00:00	2021-06-14 08:53:58	2021-
recruitnotice						
recruitsign						
replaytpost						
subjectpost						
user						
视图						
函数						
查询						
备份						
mysql						
performance_schema						
sys						
test						

3.2 前端实现

3.2.1 制作流程

- 1.划分模块和对应的界面
- 2.制作原型界面
- 3.前后端连接实现
- 4.测试系统

3.2.2 主要模块

- 1.登录注册模块
- 2.个人信息模块
- 3.活动管理模块
- 4.公告管理模块
- 5.社团管理模块
- 6.招新管理模块
- 7.人员管理模块

3.2.3 主要工作

1. 使用 axios 进行跨域传输, 使用 post 将前端数据传回后端, 并获取后端传回来的数据, 使用 get 直接获取后端传回来的数据
2. 将从后端得到的数据动态地加载到表格, 侧边栏中

3. 使用 **vuex** 存储和读取状态，如用户 **id**，社团 **id**，社团 **name**，部门 **id** 等
4. 使用 **vue.js** 和 **ElementUi** 去进行了界面的绘制，主要包括登录注册界面，侧边栏的制作，创建社团界面，全校活动界面，已经报名活动界面，全校公告界面，管理员查看所有用户和重置密码界面，管理员查看活动和审核活动界面等