**武汉大学计算机学院**

**本科生实验报告**

**IContest平台分析报告**

专 业 名 称 ：计算机科学与技术

课 程 名 称 ：软件设计与体系结构

指 导 教 师 一：王健

团 队 成 员 一： 李禛（2018302110112）

团 队 成 员 二： 章文聪（2018302110099）

# 一·项目介绍

## 1.1 项目简介及功能介绍

在各个大学校园学习生活中，各类级别的学科竞赛，创新创业竞赛是每个人大学生活的重要组成部分，参加竞赛活动有助于丰富大学生的生活经历，提升大学生的综合能力。然而现在的竞赛活动通常因不同的学科类别不同，级别不同，通常由不同的学校部门、学生组织或班委通知到学生，面对如此纷繁复杂的信息获取渠道，许多学生很容易遗漏其中的某些竞赛通知导致遗憾错过。

针对目前竞赛信息获取渠道繁杂，学生获取信息困难的困境。我们计划开发“IContest平台”项目。实现各类竞赛信息的统一收集，为同学们提供较为全面的竞赛信息获取渠道，简化了同学们获取信息的流程。同时会逐步加入其他文体竞赛、校园文体活动信息，丰富内容模块，为同学们提供竞赛信息、文体活动信息获取的一站式平台，使得更多的学生能有更多渠道获得更多的信息，拥有更加灿烂多彩的大学生活。

## 2.2 项目实践目的

本项目立足于校园生活，着眼于利用所学知识解决实际问题，同时锻炼课程所要求的专业能力：

基于UML的软件分析与设计模型建模，能够画出用例图，类图，时序图。

在实现功能的同时应用设计模式

Web服务开发，用java语言实现一个SOAP和REST API并能用客户端进行调用

# 二·需求分析与架构设计

## 2.1. 需求分析

**2.1 系统功能**

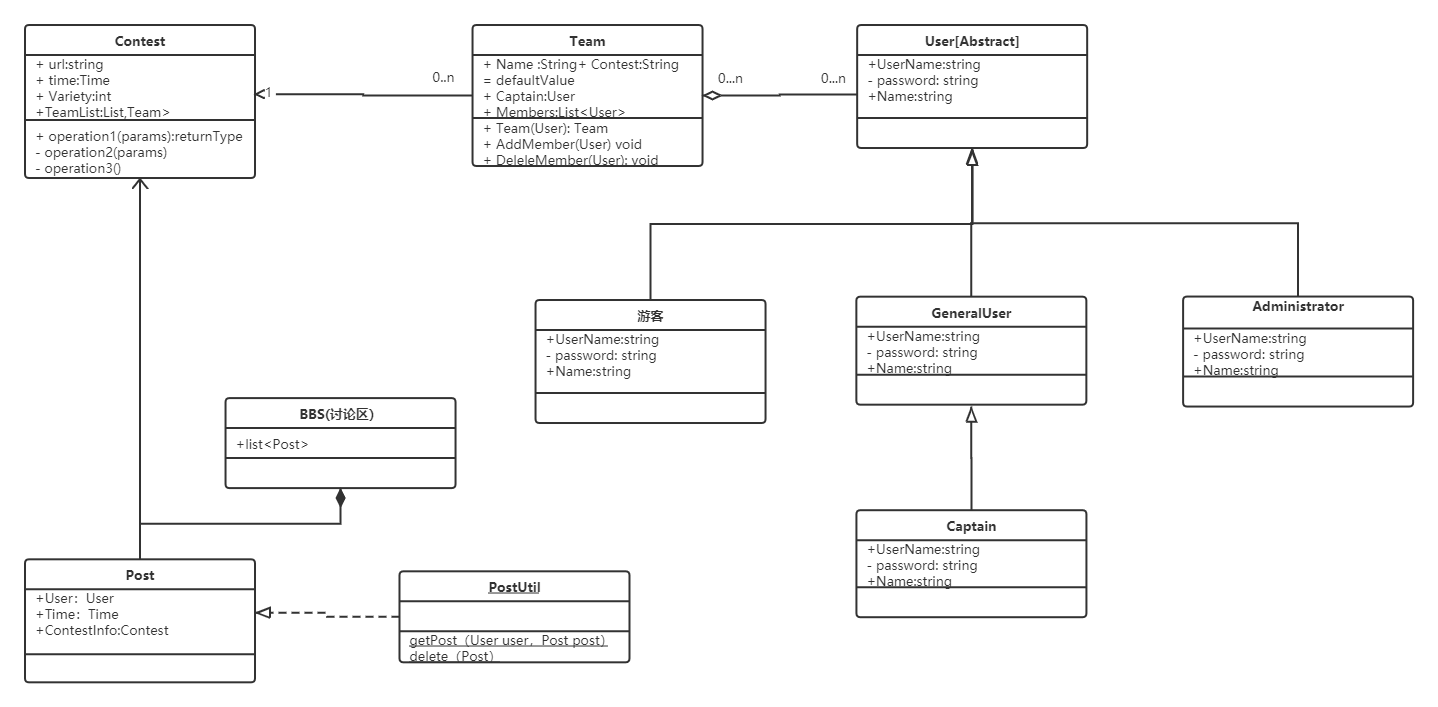
该系统的客户端模块主要提供以下功能

1. 匿名用户（游客）能够浏览平台上的竞赛信息
2. 注册成为平台用户（以下简称为：用户）和登陆
3. 用户能够浏览信息，能够在某一竞赛下创建或加入队伍
4. 讨论区发帖
5. 站内会话，队内会话
6. 队长拥有管理队伍的权限

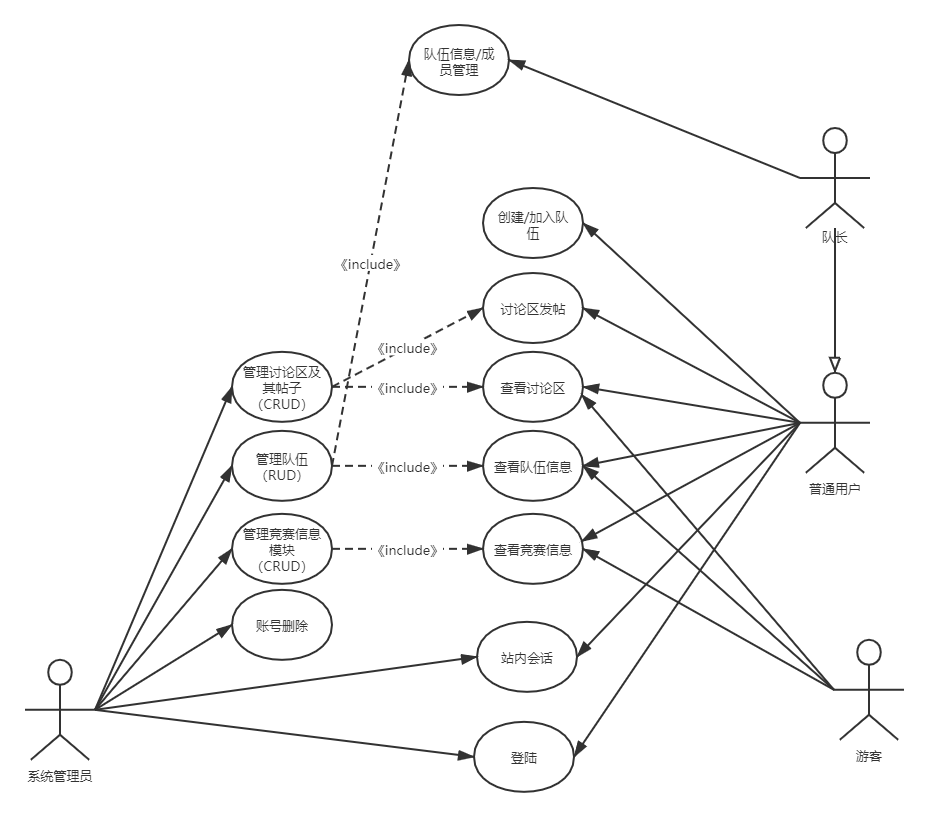
为管理员主要提供的功能：

1. 管理讨论区及其帖子
2. 管理竞赛信息模块
3. 管理队伍
4. 对有恶意行为账号的删除

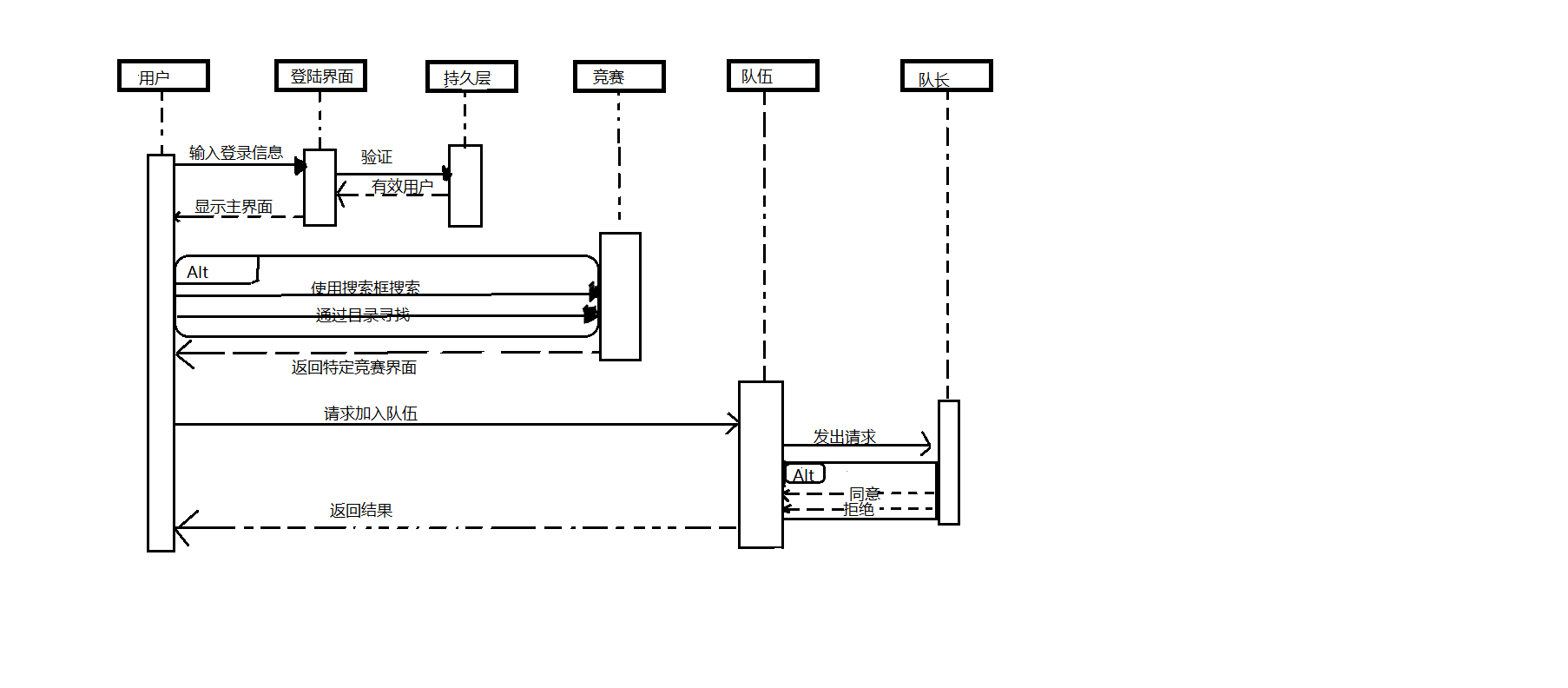
根据上述的功能绘制以下的类图：



根据上述的功能绘制以下的用例图：



时序图：



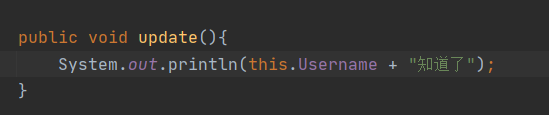
**2.2 设计模式**

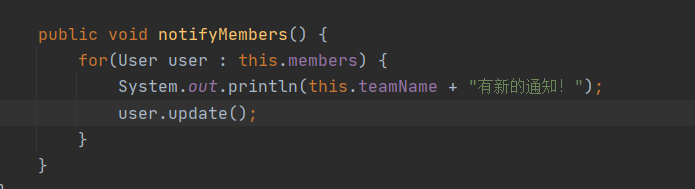
所采用的设计模式主要有3种。

第一种：观察者模式

在User类跟Team类之间存在观察与被观察的关系，观察者为team中的成员，被观察对象为team，现在有以下场景，队伍有了设置了新的通知，要通知队伍中的所有成员，这时候观察者模式就显现出来了优点。主要源码如下，User类实现了一个观察者的接口，Team类中定义一个方法notifymembers（），调用所有成员的更新方法。

代码截图如下：





设计模式主要优点：

1、观察者和被观察者是抽象耦合的。

2、建立一套触发机制。

第二种设计模式：外观模式

隐藏系统的复杂性，并向客户端提供了一个客户端可以访问系统的接口。

这里我们采用teamservice类来封装对于team对象的各种操作，dao类封装数据库操作，提供给客户端的只有service接口。





主要优点：

1、减少系统相互依赖。

2、提高灵活性。

3、提高了安全性。

第三种设计模式：单例模式

将TeamService以及其他实体服务类采用单例模式进行类设计，防止意外创建多余的TeamService对象造成对资源的浪费

截取部分代码：

public class TeameService{

private static Singleton service;

private TeamService(){}

public static synchronized TeameService getInstance() {

if (service== null) {

service= new TeameService();

}

return service;

}

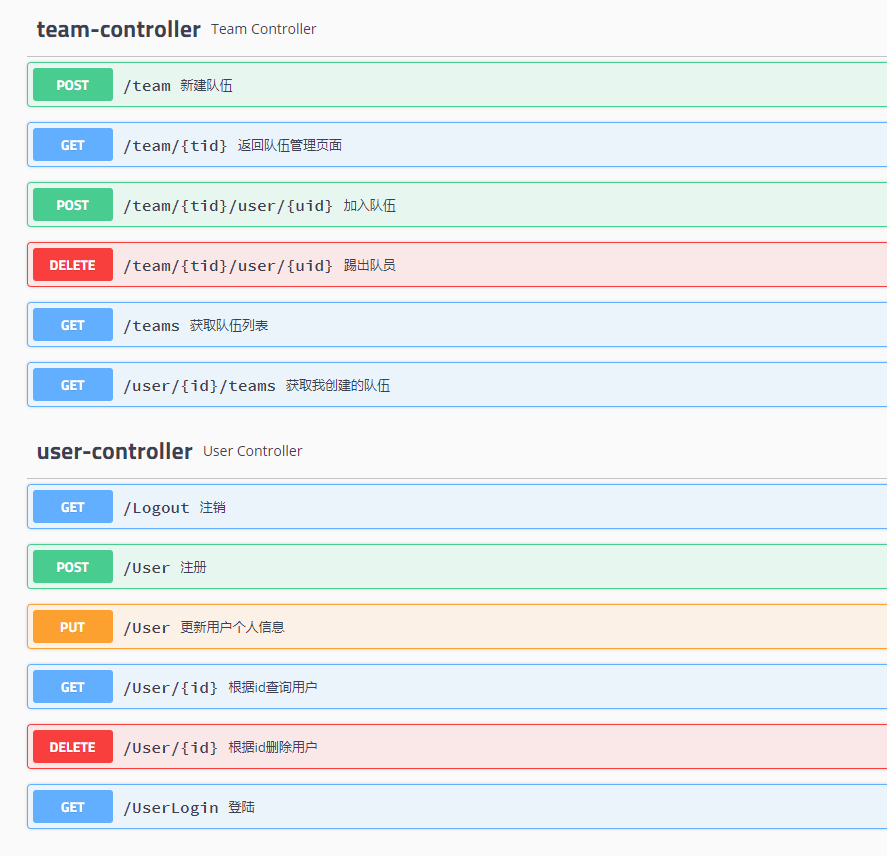
}

主要优点：

1. 在内存里只有一个实例，减少了内存的开销。
2. 避免对资源的多重占用。
3. Synchronized加锁保证线程安全。

**三、RestAPI设计**

3.2 部分后端接口设计

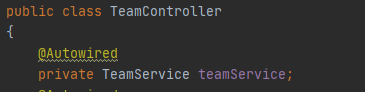


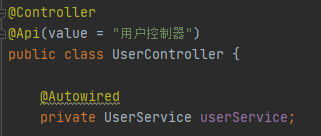
图为使用swagger-ui生成的接口文档

UserController负责提供用户注册登录，和其他CRUD的接口和业务操作

TeamContrller负责提供队伍的CRUD和队员的CRUD的接口和业务操作

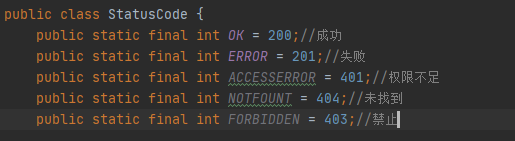
每个controller类都维护一个相应Service层的对象。





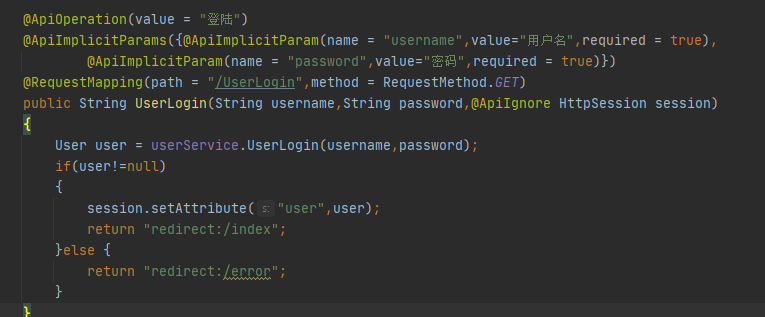
接口返回的类型分为两种：

1.封装了返回码code，返回信息msg和数据data的Result对象，由框架自动转换成Json格式返回给前端。该类型的API需要台加上注释@ResponseBody表明不经过视图解析器。



2. 返回String或ModelAndView类型，由框架的视图解析器解析后返回html页面。

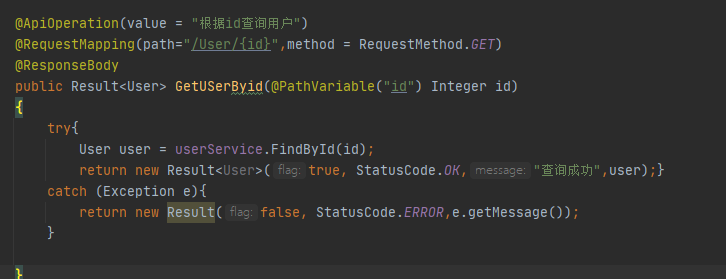
以下选择其中一些接口进行代码展示：



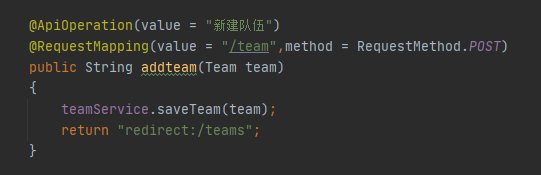
登陆功能的api，如验证正确则返回主页并将user对象储存在session中，否则返回错误页面，返回的字符串交给视图解析器解析出相应的html页面后返回给浏览器。



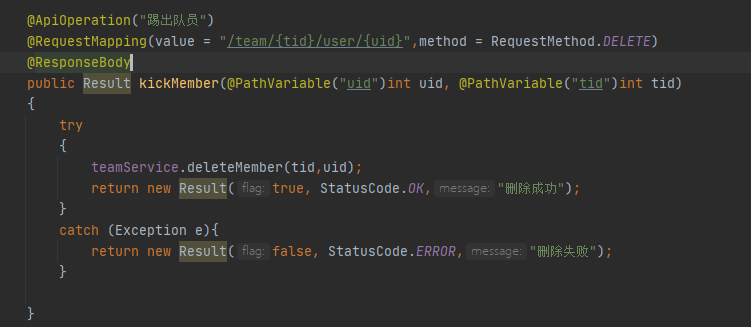
注销功能的api，删除session中维护的user对象并返回主页。



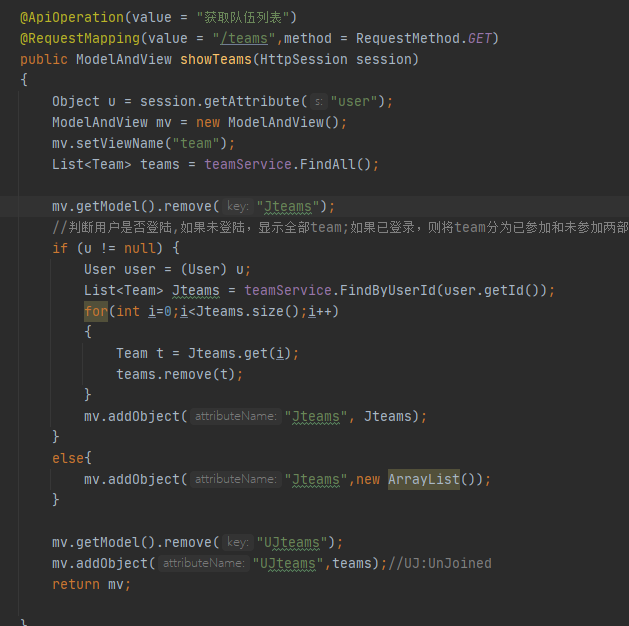
按id搜索用户的api，将返回码与搜索结果封装为Result返回，并由Spring框架封装的组件转化为Json数据格式返回到前端。



创建队伍的api，创建队伍后返回到队伍列表页面



踢出队员的api，并返回封装实体。

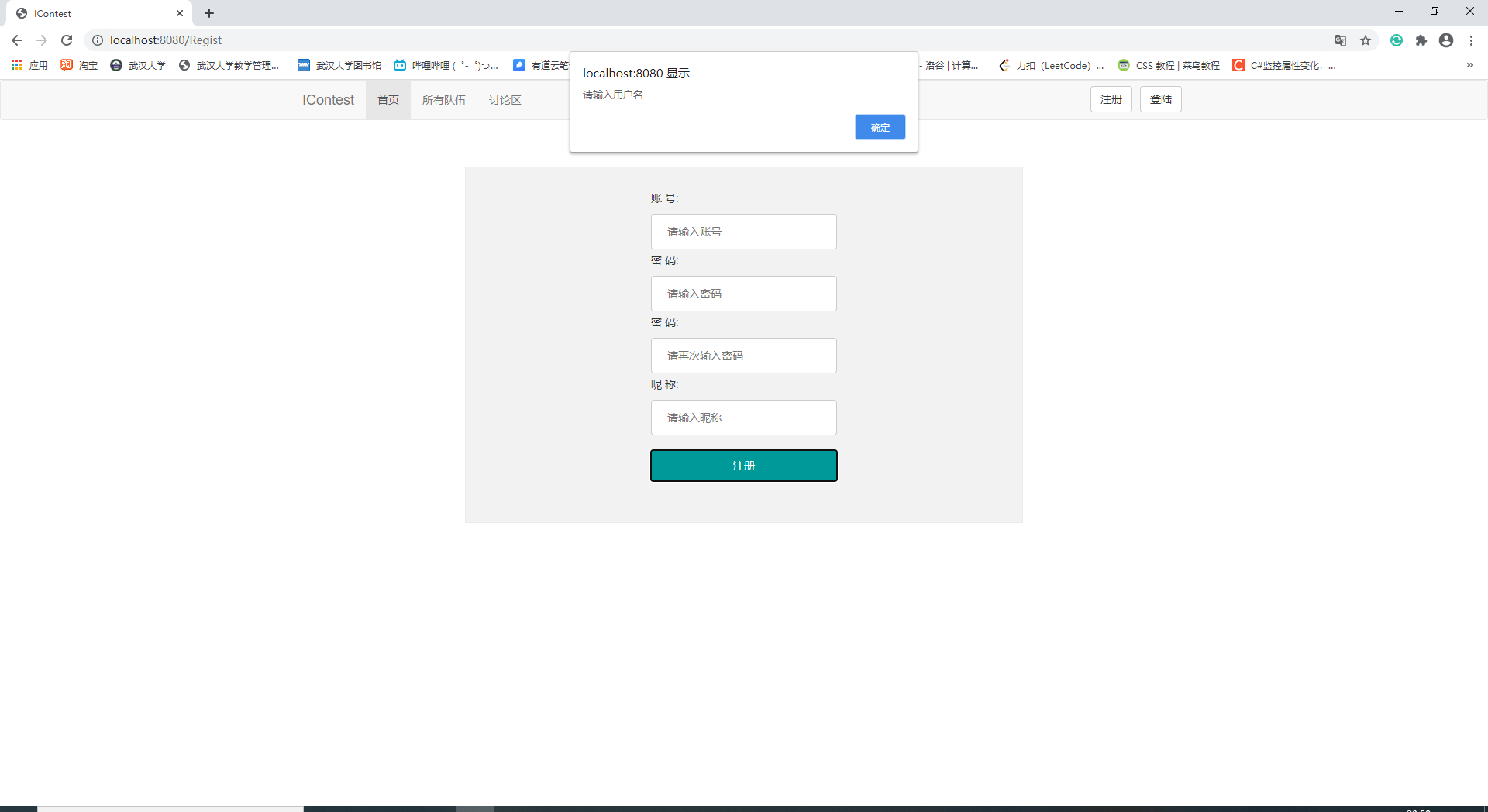


获取队伍列表的api，从数据库获取所有队伍并分为已参加的队伍和未参加的队伍两种类别填入页面的Model域并返回前端显示

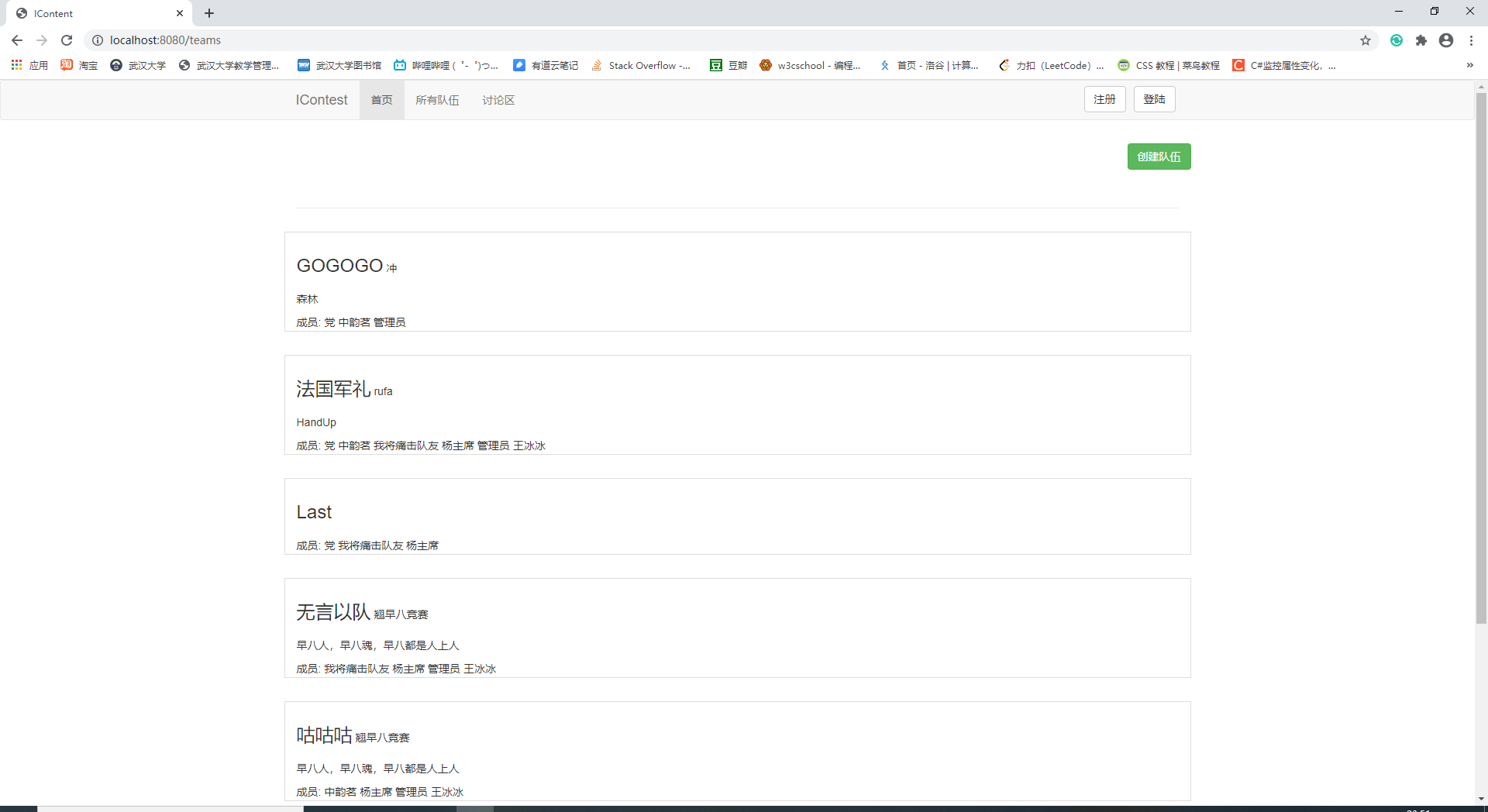
3.3效果展示



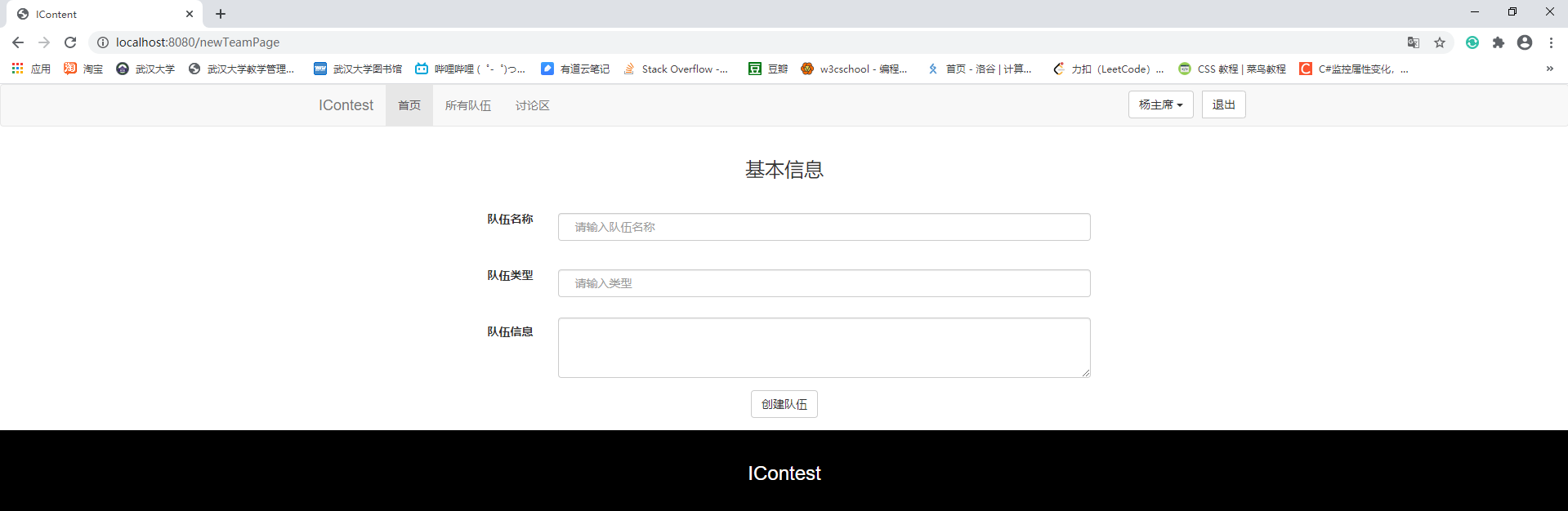
主页面：展示部分竞赛海报信息和校内活动海报信息，点击海报可跳转到相应竞赛的官网或赛氪网。



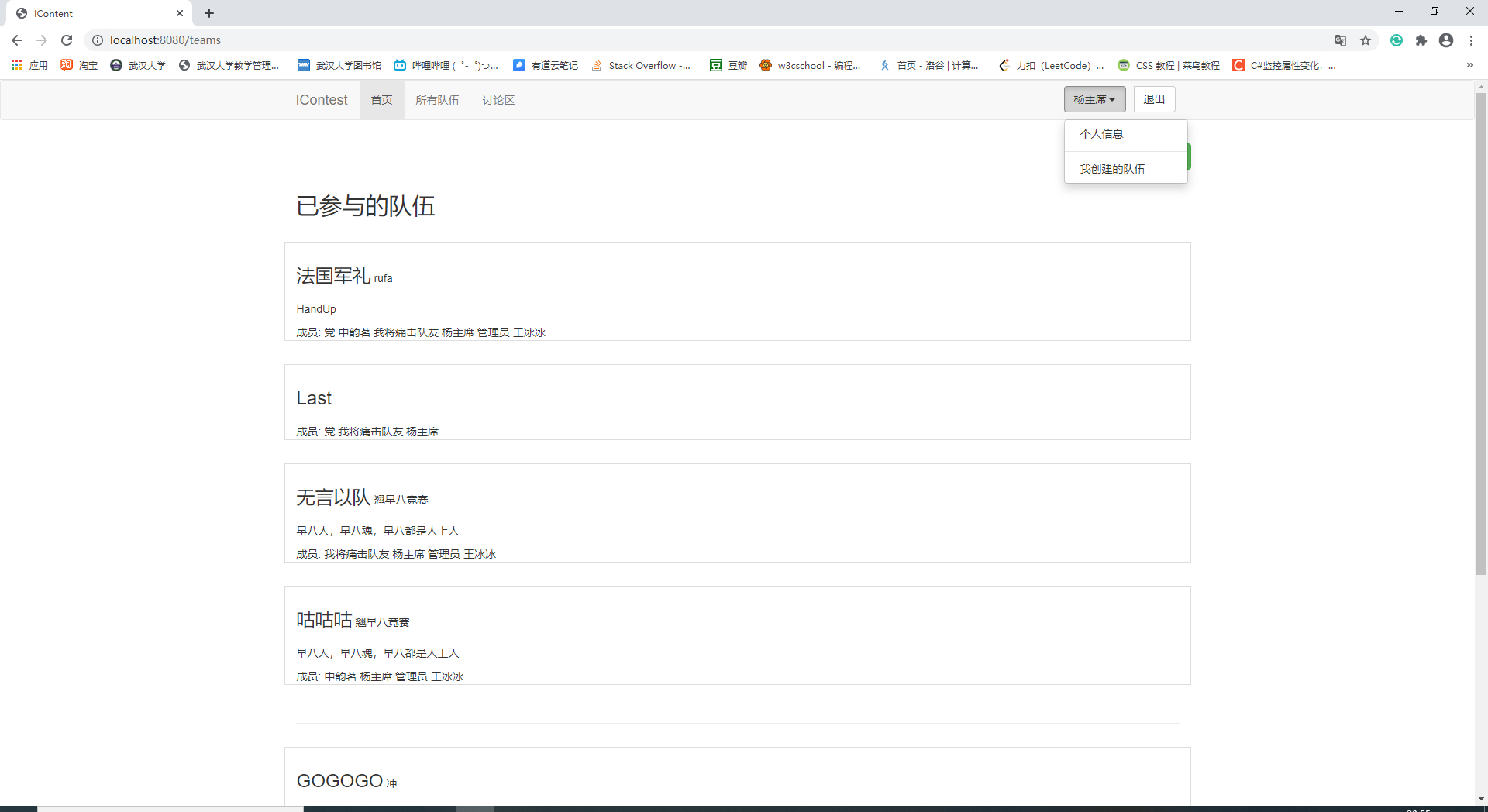
注册页面：如果某一栏没有填入会弹出提示栏。



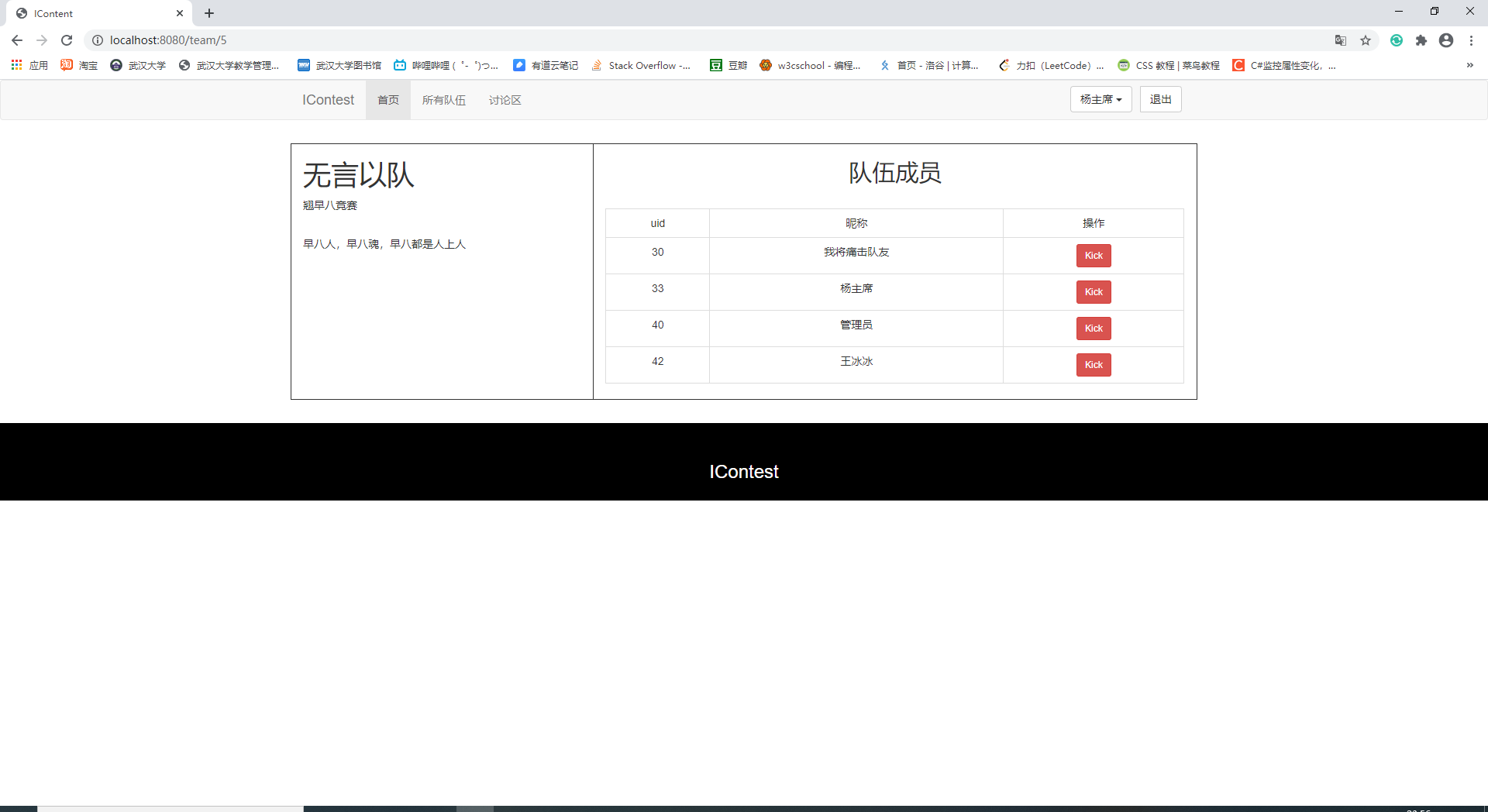
网站支持游客访问，如果没有登陆也可以浏览信息和队伍信息，但只有登陆后才能组队和建立队伍。



创建队伍页面。



登陆后在队伍界面展示已参加的队伍，右上角也会展示个人昵称，未参加队伍的右侧会有加入按钮，加入后该页面会自动刷新。



点进自己创建的队伍后可进行队员管理。