**广西民族大学相思湖学院**

**本科毕业论文（设计）**

**论文题目 网球俱乐部系统的设计与实现**

**系 部： 理工学院**

**专 业： 软件工程**

**年级班级： 2016级软件工程1班**

**学 号： 116333570157**

**学生姓名： 李林衡**

**指导教师： 淡婧娴 职称 工程师**

**目录**

[**摘 要** **3**](#_Toc14598_WPSOffice_Level1)

[**Abstract** **5**](#_Toc24613_WPSOffice_Level1)

[**1 引言** **6**](#_Toc24136_WPSOffice_Level1)

[**2 项目分析** **7**](#_Toc2100_WPSOffice_Level1)

[2.1 背景分析 7](#_Toc6886_WPSOffice_Level2)

[2.2 技术性分析 7](#_Toc24613_WPSOffice_Level2)

[**3 需求分析** **9**](#_Toc309_WPSOffice_Level1)

[3.1 概述 9](#_Toc11318_WPSOffice_Level2)

[3.2 需求规格 9](#_Toc24136_WPSOffice_Level2)

[3.3 功能详细需求描述 10](#_Toc2100_WPSOffice_Level2)

[3.4 界面需求 12](#_Toc309_WPSOffice_Level2)

[3.5 运行环境 12](#_Toc19543_WPSOffice_Level2)

[**4 系统设计** **13**](#_Toc19543_WPSOffice_Level1)

[4.1 概述 13](#_Toc14642_WPSOffice_Level2)

[4.2 开发环境 13](#_Toc14642_WPSOffice_Level2)

[4.3 关键技术 13](#_Toc30155_WPSOffice_Level2)

[4.4 数据库结构设计 14](#_Toc11448_WPSOffice_Level2)

[4.6 系统实现 16](#_Toc7991_WPSOffice_Level2)

[**5 系统测试** **22**](#_Toc14642_WPSOffice_Level1)

[5.1 测试目的 22](#_Toc10646_WPSOffice_Level2)

[5.2 测试用例表 22](#_Toc11729_WPSOffice_Level2)

[5.3 测试结果 25](#_Toc7242_WPSOffice_Level2)

[**结论** **26**](#_Toc30155_WPSOffice_Level1)

[**参考文献** **27**](#_Toc11448_WPSOffice_Level1)

[**致谢** **28**](#_Toc7991_WPSOffice_Level1)

# 

# 摘 要

当今，信息技术高度发展，作为当今信息发展的领头行业——计算机行业，衍生出了多种方便人们生活的技术，其中为代表的就是移动互联网，而手机App的开发就显得重中之中。

基于此前提，在移动互联网蓬勃发展以及国内外的网球氛围勃发的时代，研究者决定使用当今非常流行的前端框架Vue.js以及能提供中小型后台服务功能的Node.js来设计与开发一款具有满足网球爱好者日常学习和交流的移动端App，方便网球爱好者学习网球的相关知识、相互探讨学习、掌握网球的最新资讯动态。

此移动端网站将会有以下三个主要模块，分别是管理员模块，网球相关模块和个人信息模块。此移动端网站界面简洁，体验良好，为网球迷提供了许多便利。

**关键词**：移动互联网；简洁；Vue.js；Node.js

# Abstract

Nowadays, with the rapid development of information technology, computer industry, as the leading industry of information development, has derived a variety of technologies that are convenient for people's lives, among which mobile Internet is the representative, and the development of mobile app is very important.

Based on this premise, in the era of vigorous development of mobile Internet and booming tennis atmosphere at home and abroad, researchers decided to use the very popular front-end framework vue.js and node.js which can provide small and medium-sized back-end services to design and develop a mobile app that can meet the daily learning and communication of tennis lovers, so as to facilitate tennis fans to learn tennis related knowledge Learn from each other and master the latest information of tennis.

This mobile website will have the following three main modules: administrator module, tennis related module and personal information module. This mobile website has a simple interface and a good experience, which provides a lot of convenience for net fans.

Keywords: Mobile Internet; simplicity; vue.js; node.js

**网球俱乐部系统的设计与实现**

# 1 引言

移动互联网采用移动无线通信方式获取业务和服务的新兴业务，随着宽带无线接入技术和移动终端技术的飞速发展，人们迫切希望能够随时随地乃至在移动过程中都能方便地从互联网获取信息和服务，移动互联网应运而生并迅猛发展。

放眼当今，网球热潮席卷全球，网球是不分年龄段的长寿运动，网球运动包含的多个技术动作都对手眼协调有很好的作用，同时提高敏捷度和平衡性。便宜的装备、便捷的球场、众多的球友。无论双打还是团体赛，都能有效锻炼你的协作能力。在塑造积极向上的个性和人格魅力方面，网球运动当仁不让！这注定了网球市场的发展是无比光明的。所以，设计与制作一款方便网球迷学习与交流的网球移动端App，是可行的。

本论文将分为三大章节，由需求分析，概要设计以及详细设计三个主要章节组成。网球俱乐部系统从最初的设计到最终的实现都进行了完整和详细的说明讲解。

# 2 项目分析

## 2.1 背景分析

在我国互联网的发展过程中，PC互联网已日趋饱和，移动互联网却呈现井喷式发展。数据显示，截至2018年12月，我国网民规模达8.29亿，占比达98.6%。伴随着移动终端价格的下降及wifi的广泛铺设，移动网民呈现爆发趋势。

因为移动互联网的便捷易接入性、随时随地性、更多功能集成，以及更大普及人群，形成对人信息消费时间的更多占领。表现为用户使用智能手机的时间高速增长，而对电视，报纸等媒体的投入时间则不断下跌。

移动端相较于PC端的一大优势便是方便快捷。在移动互联网成长期间，涌现出许多成熟的前端框架，其中的三大代表便是React、Angular以及Vue.js。Vue.js以其轻量级的体积，容易上手的难度得到了大批编程人员的喜爱。同时，也兴起了方便前端人员处理与获取数据的服务器语言Node.js。

## 2.2 技术性分析

Vue.js是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层，并且非常容易学习，非常容易与其它库或已有项目整合。另一方面，Vue 完全有能力驱动采用单文件组件和Vue生态系统支持的库开发的复杂单页应用。

Vue.js是采用MVVM框架的一种前端框架，MVVM是Model-View-ViewModel的简写。即模型-视图-视图模型。模型指的是后端传递的数据。视图指的是所看到的页面。视图模型mvvm模式的核心，它是连接view和model的桥梁。它有两个方向：一是将模型转化成视图，即将后端传递的数据转化成所看到的页面。实现的方式是：数据绑定。二是将视图转化成模型，即将所看到的页面转化成后端的数据。实现的方式是：DOM 事件监听。这两个方向都实现的，我们称之为数据的双向绑定。在MVVM的框架下视图和模型是不能直接通信的，它们通过ViewModel来通信，ViewModel通常要实现一个observer观察者，当数据发生变化，ViewModel能够监听到数据的这种变化，然后通知到对应的视图做自动更新，而当用户操作视图，ViewModel也能监听到视图的变化，然后通知数据做改动，这实际上就实现了数据的双向绑定。并且MVVM中的View 和 ViewModel可以互相通信。

Vue.js可编写单页面应用（SPA），单页面应用广泛地应用于各种手机移动端app之中，其一大优点便是不需要整个页面的刷新，也可以改变单个位置的数据，给人一种无刷新的良好体验，想象一下，如果每改变一次数据，就要刷新一次整个页面，这种体验显然是很糟糕的。

对于后台服务器的选择，研究者选择近几年来新兴的Node.js作为网球俱乐部系统的后台编程语言，Node.js不同于JAVA以及PHP这些重量级的后台语言，其定位是服务中小型的软件项目。

NodeJs的首要目标是提供一种简单的、用于创建高性能服务器及可在该服务器中运行的各种应用程序的开发工具。严格地说，Node.js是一个用于开发各种web服务器的开发工具。在Node.js服务器中，运行的是高性能V8 JavaScript脚本语言，该语言是一种可以运行在服务器端的脚本语言。在V8 JavaScript引擎内部使用一种全新的编译技术。这意味着开发者编写的高端的 JavaScript 脚本代码与开发者编写的低端的C语言具有非常相近的执行效率，这也是Node.js服务器可以提供的一个重要特性。所以使用node.js作为服务端语言就是一个不错的选择。

# 3 需求分析

## 3.1 概述

在网球俱乐部系统中，主要的两个参与者分别是系统管理员以及用户。详细分析了管理员及用户的主要功能需求。

需求分析不仅规定界面的整体风格主题，而且选择矢量图作为界面图标的标准以及字体图标的提示样式。

## 

## 3.2 需求规格

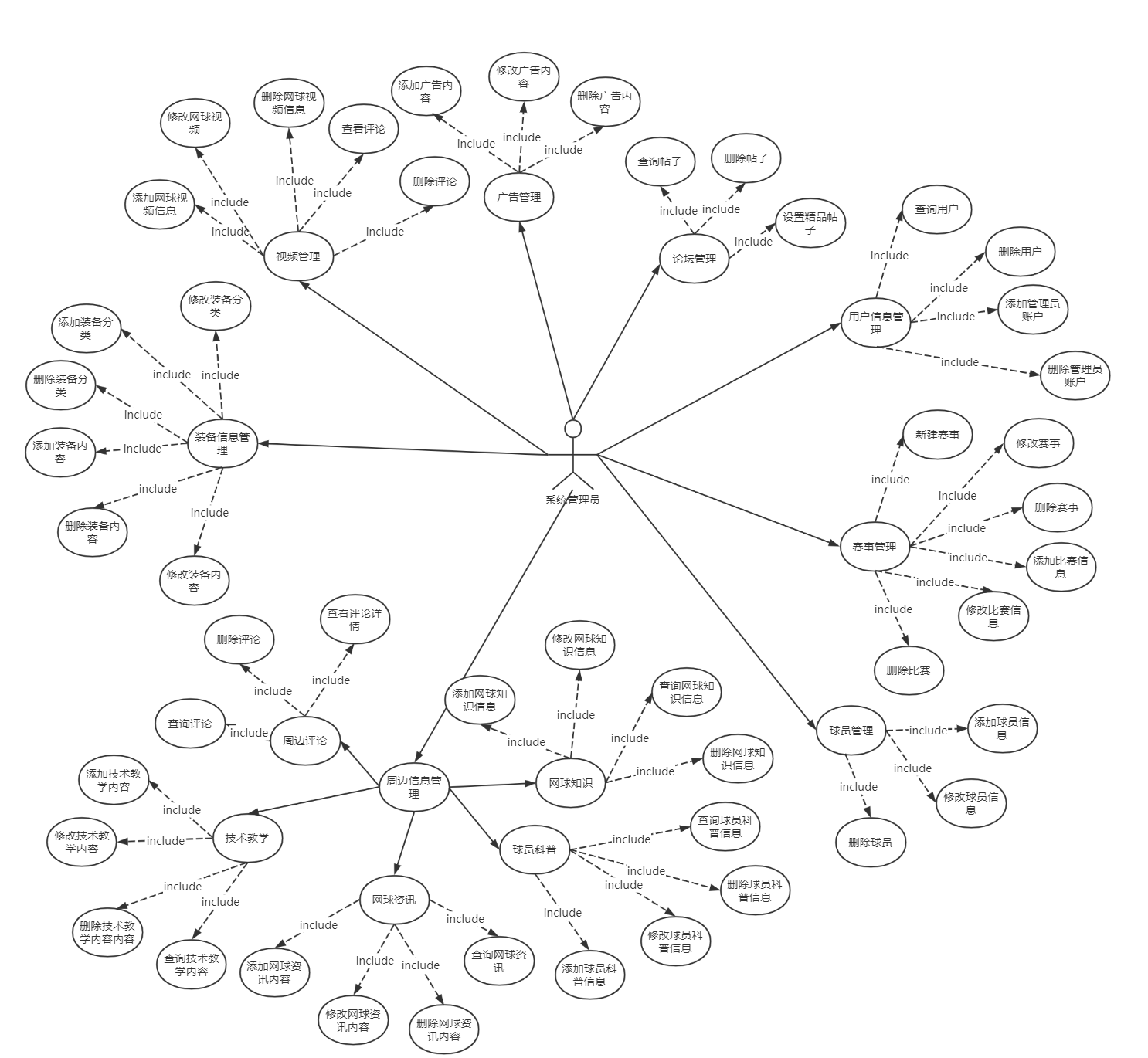


图3-1 系统管理员用例图

系统管理员用例图描述了管理员这个参与者的各项功能

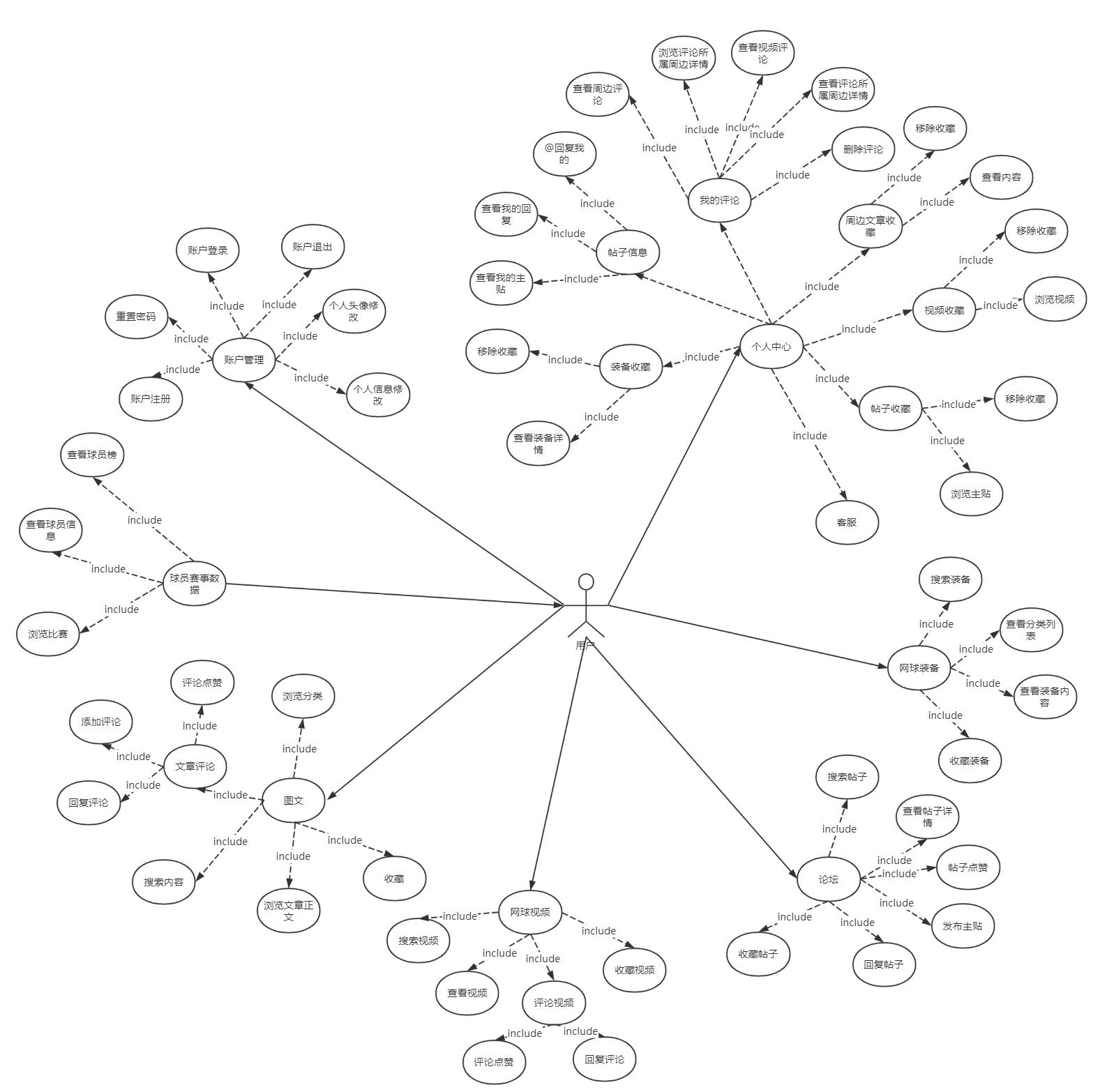


图3-2 用户用例图

用户用例图描述了用户这个参与者的各项功能

## 3.3 功能详细需求描述

### 3.3.1 基础功能

注册功能：用户可以在注册页面，注册具有唯一性的个人账户，且用户名不会重名。

登录功能：用户可以在登录界面，使用已有的用户名和密码登录网球俱乐部系统,管理员可以使用专属的用户名和密码登录系统管理页面。

重置密码：用户可以在密码修改界面验证原始密码的正确性来修改个人账户的密码。

退出功能：用户可以点击个人界面里的退出按钮退出网球俱乐部系统。

### 3.3.2 管理员功能

系统管理员账号管理：管理员可以在管理员账号列表中任意地添加新的管理员账号、修改管理员账号的密码、或删除已有的管理员账号。

查看用户个人信息：管理员可以在用户列表任意查看某位用户的个人信息。

图文信息管理：管理员可以在文章分类列表里任意地添加、修改或删除关于网球俱乐部系统的图文分类，在文章列表里任意地添加、修改或删除关于网球的各类科普文章、体育资讯或学习资料。

帖子管理：管理员可以在帖子管理列表中查看或删除已有的帖子信息。

视频管理：管理员可以在视频列表里任意地添加、修改或删除任何关于网球的视频信息。

网球装备管理：管理员可以在网球装备分类列表里任意地添加、修改或删除关于网球的装备分类项，在网球装备列表里任意地查看、添加、修改或删除关于网球装备的图文展示信息。

广告管理：管理员可以在广告列表里任意地添加、修改或删除关于网球俱乐部系统的轮播广告信息。

赛事管理：管理员可以在比赛列表里任意地添加、修改或删除关于网球运动的比赛。

球员管理：管理员可以在球员列表里任意地添加、修改或删除网球运动员的具体资料。

### 3.3.3 用户功能

修改个人信息：用户可以在个人界面点击修改信息的跳转图标，跳转到个人信息修改页面，修改个人的账号信息，如电话号码，邮箱以及个人签名。也可以点击头像框，跳转到上传头像界面，修改个人展示头像。

浏览文章：用户可以在网球文章区点击图文的分类，浏览网球资讯、技术教学、网球知识以及球员科普等图文内容。

查看网球视频：用户可以在视频列表页，点击任意视频跳转到视频详情页，观看该视频。

查看网球装备详情：用户可以在装备列表页，点击任意的网球装备分类，跳转到该分类的装备列表，点击右上方的展示模式按钮，可以让装备项按横向和纵向的方式排列展示，点击任意装备项则跳转到装备详情页。

浏览主题帖子信息：用户可以在点击论坛页中任意一个主帖，跳转到该主帖的详情页面进行浏览，详情页中分为全部回复和只看楼主，点击全部则显示全部用户的跟帖，点击只看楼主则只显示楼主的跟帖。点击左上方的排序按钮，则会弹出正序和倒序的选项，可对其进行排序选择。

发布主题帖子：用户可以点击论坛页中的发布按钮，跳转到主题帖子编辑页面进行编辑发布操作，完成后点击右上角的发布按钮进行发布操作。

综合搜索功能：用户可以在网球文章、网球视频、主题帖子以及网球装备等列表页面顶部的搜索框中搜索指定的内容。

综合收藏功能：用户可以点击各详情页面的收藏按钮，收藏网球文章、网球视频、主题帖子以及网球装备等，二次点击则会移除收藏。在用户个人中心的收藏区，可以浏览已收藏的网球视频、主题帖子以及网球装备等具体内容。

综合评论功能：用户可以在网球文章、网球视频、主题帖子等详情页面发表评论或回复。在用户个人中心的帖子管理及评论管理可以查看发布过的帖子、已发布的视频评论及文字评论等具体内容。

点赞功能：用户可以对网球文章、网球视频、主题帖子等详情页面中评论或回复进行点赞操作，二次点击则会撤销点赞。

客服帮助：用户可以在我的主页点击客服帮助选项，跳转到客服帮助页面，了解更多信息。

## 3.4 界面需求

整体需求：网球俱乐部系统的整体界面是简洁风格(如图3-3所示)，黑白为主，辅以艳丽的红蓝色，给予使用者一种舒适的观感。



图3-3 移动端基本样式

图标选择：尽可能多地使用用矢量图作为网站的图标（如图3-4所示），避免不同型号的手机查看图标出现失真的情况。



图3-4 网站图标样式

操作提示：用户界面的各种信息提示皆以Q版的矢量图提示为主(如图3-5所示)。

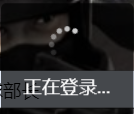
 

图3-5 网站信息提示样式

手机兼容性：兼容大部分不同尺寸的手机型号。

## 3.5 运行环境

* 操作系统： Window10
* 服务器： Node.js
* 数据库： MongoDB

# 4 系统设计

## 4.1 概述

系统设计章节涵盖了以下的设计内容。

开发环境的选择：依据运行环境来选择最终的开发环境。

关键技术：分析网站运用的一些关键性的编程技术。

数据结构的设计：明确数据库信息的命名规范，保持其一致性。清晰地描述表与表之间的联系。

### 4.2 开发环境

* 操作系统： Window10
* 开发语言： Javascript
* 服务器： Node.js
* 数据库： MongoDB
* 依赖： Vue.js

## 4.3 关键技术

处理跨域请求：浏览器受到同源策略的限制，不同的域名, 不同端口, 不同的协议不允许共享资源的。同源策略是针对浏览器设置的门槛。为了实现更好的前后端分离模式，本网站使用Node.js的CORS模块解决跨域请求。

MVVM（Model-ViewModel-Model）：是Vue.js框架的架构模式，View层代表的是视图、模版，负责将数据模型转化为UI展现出来。Model层代表的是模型、数据，可以在Model层中定义数据修改和操作的业务逻辑。而Vue.js则是ViewModel层，负责连接Model和View。数据(Model)和视图(View)是不能直接通讯的，而是需要通过ViewModel来实现双方的通讯。当数据变化的时候，ViewModel能够监听到这种变化，并及时的通知view做出修改。同样的，当页面有事件触发时，ViewModel也能够监听到事件，并通知model进行响应。ViewModel就相当于一个观察者，监控着双方的动作，并及时通知对方进行相应的操作。

路由跳转：页面的路由跳转用到Vue.js框架的Vue-router插件，以hash模式跳转页面。

数据获取：数据获取用到使用Vue.js框架的axios插件，向Node服务器请求数据。

防xss攻击：xss攻击通常指的是通过利用网页开发时留下的漏洞，通过巧妙的方法注入恶意指令代码到网页。服务端中利用node-xss库对需要用户输入的数据进行过滤，达到安全防御的目的。

Token验证：token其实说的更通俗点可以叫暗号，在一些数据传输之前，要先进行暗号的核对，不同的暗号被授权不同的数据操作，在此使用Node.js里的jwt(JSON Web Tokens)模块进行身份验证。

## 4.4 数据库结构设计

## 4.4.1 数据库概念结构设计

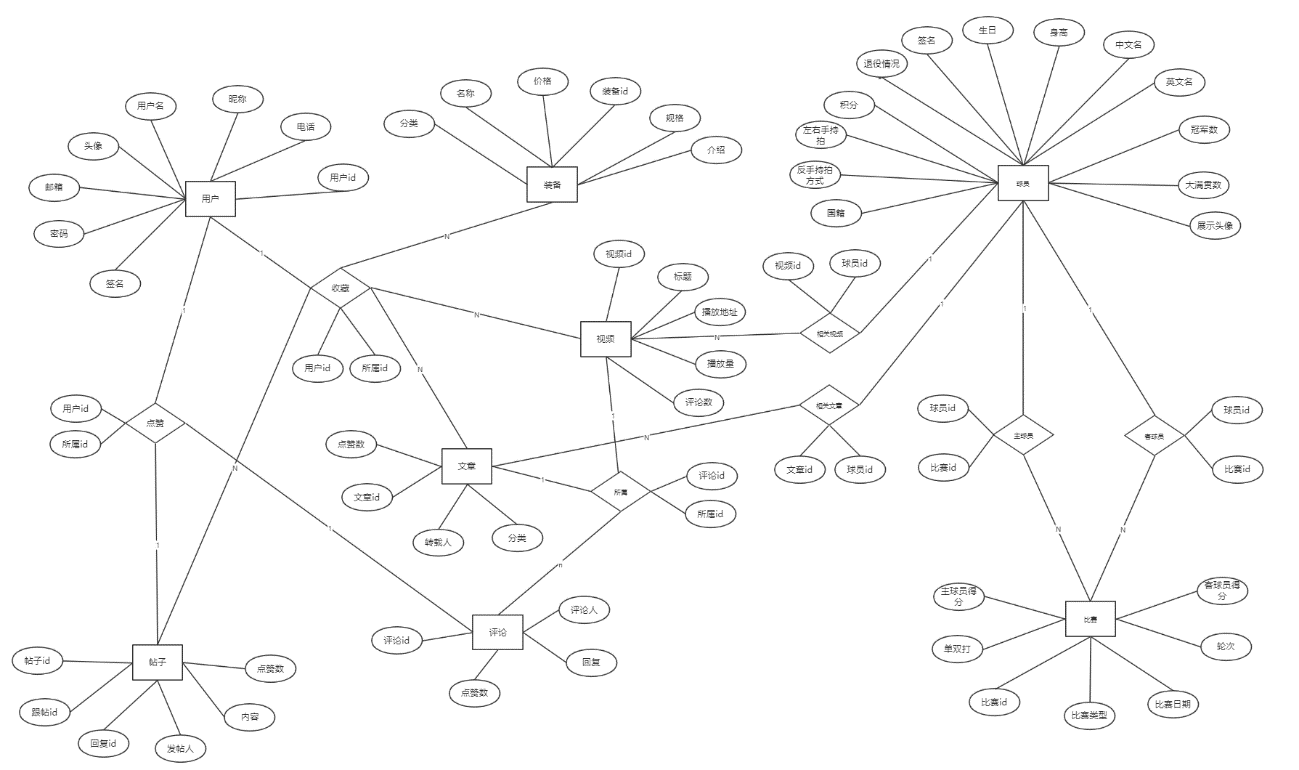


图4-1 数据结构ER图

数据结构ER图描述了系统中数据结构之间的数据关系

user（用户）：在数据表中，user作为一张数据表，其中包含了用户ID，用户名、密码、电话号码、签名、邮箱、性别与头像。

article(文章)：skin表中包含了文章id、转载人、分类，通过具有唯一性的文章id区分文章。

product（装备）：product表中包含装备id、分类、名称、封面、发布时间、规格、价格及介绍，其中以具有唯一性的装备id区分不同的装备。

video（视频）：video表中包含视频id、标题、播放量、转载者、简介、发布日期以及播放地址，其中以具有唯一性的视频id区分视频。

commend（评论）：commend表中包含评论id、评论人id、回复以及点赞。

post（帖子）：post表中包含帖子id、发帖人、回复id、跟帖id、内容以及点赞，通过具有唯一性的帖子id区分帖子。

player（球员）：post表中包含球员id、中文名、英文名、性别、生日、身高、展示头像、国籍、退役情况、左右手持拍、反手持拍类型、积分、个人荣誉、相关文章以及相关视频，通过具有唯一性的球员id区分球员。

contest（比赛）：contest表中包含比赛id、大赛类型、比赛时间、轮次、单双打类型、主球员、客球员、主球员得分、客球员得分。通过具有唯一性的比赛id区分比赛。

### 4.4.2 数据库物理结构设计

表4-1 用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **允许空** | **约束** | **备注** |
| \_id | String | N | 主键 | 用户id |
| username | String | N |  | 用户名 |
| password | String | N |  | 密码 |
| nickname | String | N |  | 昵称 |
| email | String | N |  | 邮箱 |
| tel | String | N |  | 电话 |
| avatar | String | N |  | 头像 |
| note | String |  |  | 签名 |
| gender | Number | N |  | 性别 |

表 4-2 文章表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **允许空** | **约束** | **备注** |
| \_id | String | N | 主键 | 文章id |
| category | ObjectId | N | 外键 | 分类 |
| title | String | N |  | 标题 |
| author | String | N |  | 转载者 |
| cover | String |  |  | 封面图 |
| content | String |  |  | 内容 |
| reads | Number | N |  | 阅读量 |
| createdAt | Date | N |  | 发布时间 |

表 4-3 视频表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **允许空** | **约束** | **备注** |
| \_id | String | N | 主键 | 视频id |
| title | String | N |  | 名称 |
| author | String | N |  | 转载者 |
| reads | Number | N |  | 播放量 |
| cover | String | N |  | 封面图 |
| intro | String |  |  | 简介 |
| url | String | N |  | 视频地址 |
| laber | String |  |  | 标签 |
| createdAt | Date | N |  | 发布时间 |

表 4-4 装备表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **允许空** | **约束** | **备注** |
| \_id | String | N | 主键 | 装备id |
| parent | ObjectId | N | 外键 | 分类 |
| name | String | N |  | 名称 |
| content | String | N |  | 内容 |
| cover | String | N |  | 封面图 |
| recommend | String |  |  | 推荐理由 |
| specification | String |  |  | 规格 |
| price | Number | N |  | 价格 |
| createdAt | Date | N |  | 发布时间 |

表 4-5 帖子表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **允许空** | **约束** | **备注** |
| \_id | String | N | 主键 | 帖子id |
| user | ObjectId | N | 外键 | 用户id |
| parent | ObjectId |  | 外键 | 跟帖id |
| to | ObjectId |  | 外键 | 回复id |
| reply | ObjectId |  | 外键 | 帖子用户id |
| content | String | N |  | 内容 |
| fabulou | ObjectId |  | 外键 | 点赞用户id |
| layer | Number | N |  | 楼层数 |

表 4-6 评论表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **允许空** | **约束** | **备注** |
| \_id | String | N | 主键 | 评论id |
| user | ObjectId | N | 外键 | 用户id |
| commend | ObjectId |  | 外键 | 回复id |
| reply | ObjectId |  | 外键 | 评论用户id |
| content | String | N |  | 内容 |
| fabulou | ObjectId |  | 外键 | 点赞用户id |

表 4-7 球员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **允许空** | **约束** | **备注** |
| \_id | String | N | 主键 | 球员id |
| cname | String | N |  | 中文名 |
| ename | String |  |  | 英文名 |
| height | Number |  |  | 身高 |
| country | String | N |  | 国籍 |
| birthday | Date |  |  | 生日 |
| avatar | String |  |  | 展示头像 |
| retire | Boolean | N |  | 退役情况 |
| forehand | Boolean | N |  | 惯用手 |
| backhand | Boolean | N |  | 反手持拍 |
| integral | Number |  |  | 积分 |
| championship | Number |  |  | 单打冠军数 |
| grandslam | Number |  |  | 大满贯数 |
| articles | ObjectId |  | 外键 | 相关文章 |
| videos | ObjectId |  | 外键 | 相关视频 |

表 4-8 比赛表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **允许空** | **约束** | **备注** |
| \_id | String | N | 主键 | 比赛id |
| date | Date | N |  | 比赛时间 |
| match | ObjectId | N | 外键 | 大赛类型 |
| single | Boolean | N |  | 单双打 |
| round | Number |  |  | 轮次 |
| host | ObjectId | N | 外键 | 主球员 |
| guest | ObjectId | N | 外键 | 客球员 |
| hostscore | Number |  |  | 主球员得分 |
| guestscore | Number |  |  | 客球员得分 |

表 4-9 点赞表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **允许空** | **约束** | **备注** |
| \_id | String | N | 主键 | 赞id |
| from | ObjectId | N | 外键 | 来源 |
| user | ObjectId | N | 外键 | 用户 |

表 4-10 收藏表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **类型** | **允许空** | **约束** | **备注** |
| \_id | String | N | 主键 | 赞id |
| parent | ObjectId | N | 外键 | 来源 |
| user | ObjectId | N | 外键 | 用户 |
| createdAt | Date | N |  | 收藏时间 |

# 5 系统实现

### 5.1 用户端界面

**5.1.1 登陆注册**

用户在此输入已创建的用户名和登录密码 登录网球俱乐部系统。登录界面的效果如图 5-1所示。用户注册时，不正确的输入情况下，输入框右方都会给予相应的友情提示。注册页面的效果如图 5-2所示。

**图5-1用户登录 图5-2用户注册**

**5.1.2 网球文章**

首页推荐展示网球相关的图文分类及具体文章信息，具体效果如下展示。

**图5-3网球文章分类列表 图5-4网球文章正文 图5-5网球文章评论**

**5.1.3 网球赛事**

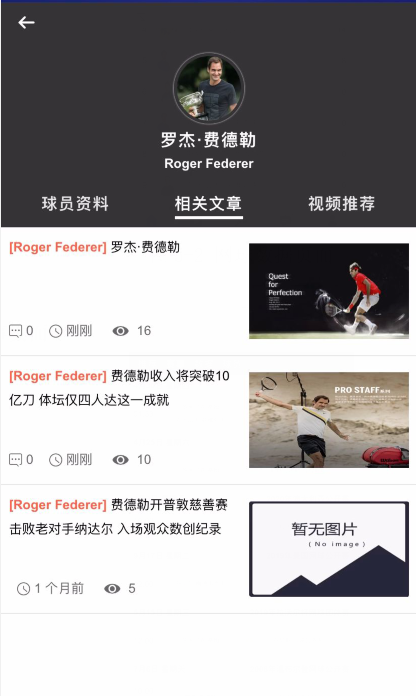
网球比赛模块中展示历年时段的比赛信息，具体效果如下展示。



**图5-6网球赛事列表 图5-7网球赛事分类**

**5.1.4 球员科普与数据**

图 5-8 展示系统首页的球员数据信息内容。图 5-9 展示了系统首页的球员列表内容 ，在顶部的搜索框可搜索指定的球员信息。图 5-10 为球员详细数据页面，具体效果如下展示。



**图5-8球员排名列表 图5-9网球球员列表 图5-10网球球员详情**

**5.1.6 网球视频**

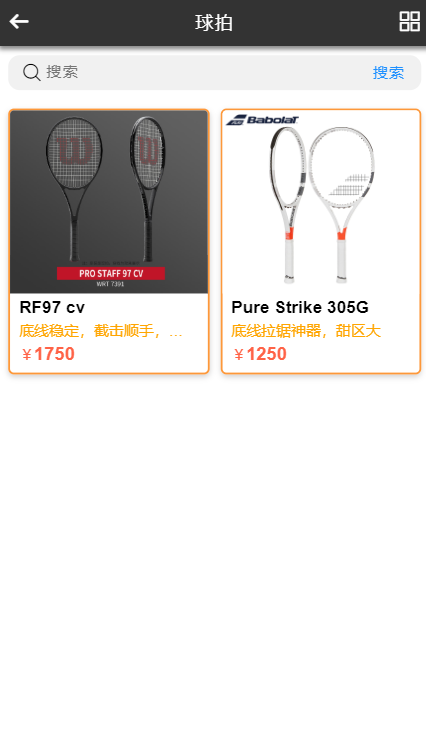
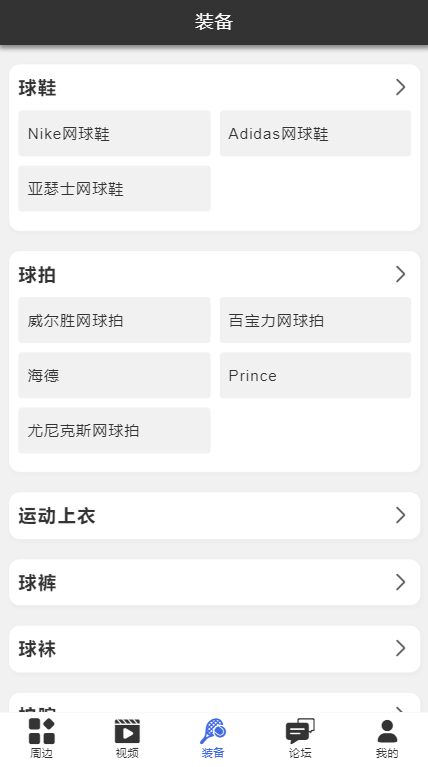
图 5-11 展示了网球视频列表内容，顶部的搜素框可以搜索指定的视频**。**在网球视频详情页中，用户可进行播放视频、查看更多视频、浏览相关评论、收藏视频、评论视频、对评论点赞及回复等操作。具体效果如图5-12与图5-13所示。



**图5-11网球视频列表 图5-12网球视频详情 图5-13网球视频评论**

**5.1.7 网球装备**

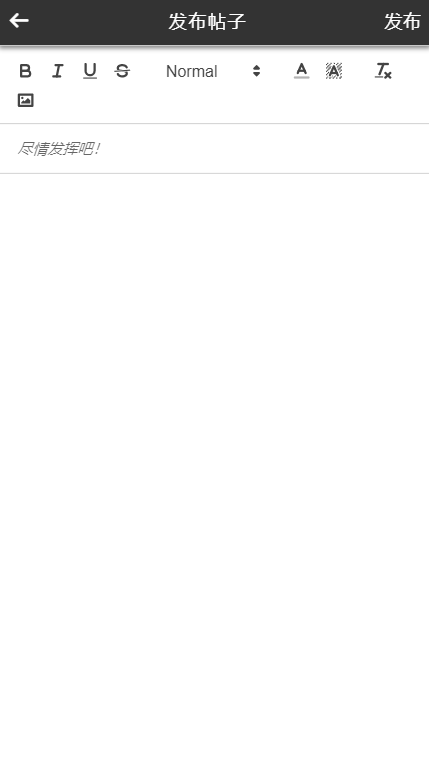
用户可浏览网球装备分类、查看特定分类的装备列表、收藏装备以及查看装备详细介绍与推荐信息，具体效果如下展示。



**图5-14网球装备分类 图5-15网球装备列表 5-16网球装备详情**

**5.1.8 帖子信息**

用户可浏览主题帖子列表的帖子内容，在顶部搜索栏可搜索指定帖子。在帖子详情中可进行收藏帖子、点赞帖子、回复帖子等操作，具体效果如下。



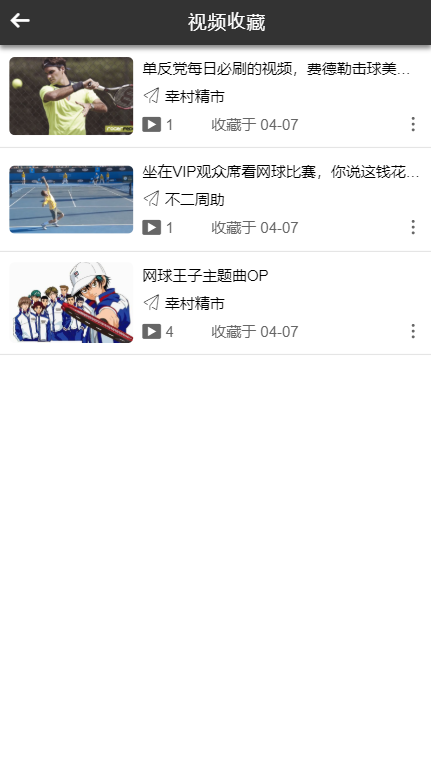
**图5-17主题帖子列表 图5-18主题帖子详情 图5-19帖子编辑**

**5.1.9 个人中心**

个人中心页面可以进行用户信息修改、综合收藏、帖子管理及评论管理等操作，具体效果如下展示。



**图5-20 个人中心 图5-21 个人信息修改 图5-22 用户头像上传**

**图5-23 视频收藏 图5-24帖子管理 图5-25 我的评论**

**5.7 文章管理页面**



图5-16 文章管理页面

**5.7 视频管理页面**



图5-17 视频管理页面

**5.7 装备详情编辑页面**

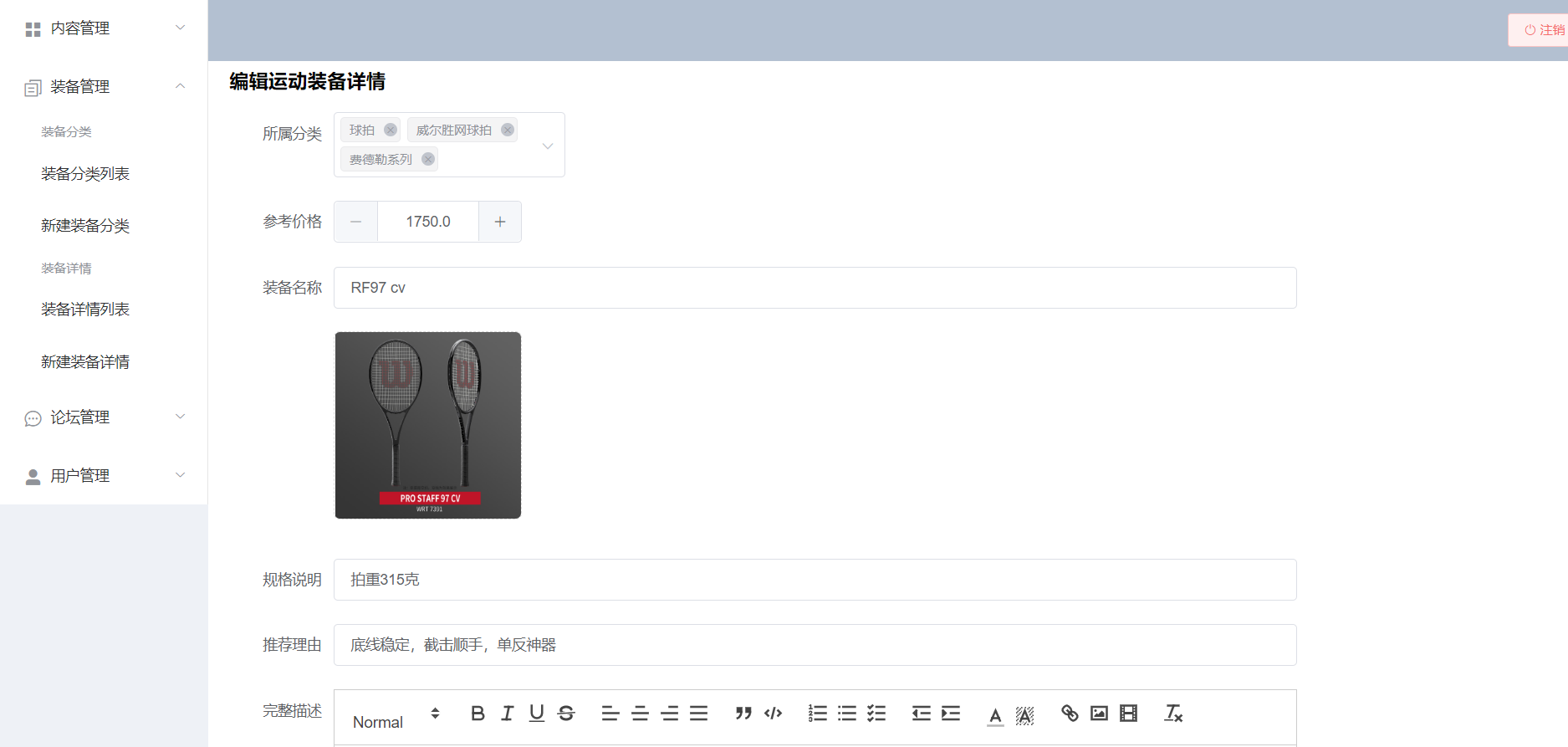


图5-13 视频管理页面

# 

# 5 系统测试

## 5.1 测试目的

软件测试是伴随着软件的产生而产生的。当网球俱乐部系统初步完成的时候，就需要用测试用例来检验一个系统是否合格，质量是否有保证，测试可以发现系统中存在的错误、缺陷和不足。

## 5.2 测试用例表

表6-1 系统基础功能测试

|  |  |  |
| --- | --- | --- |
| 测试方案 | 测试数据 | 预期结果 |
| 不填写用户名或者密码直接登录 |  | 登录按钮无法点击 |
| 使用未注册的账号登录 | 用户名：abc123 密码：cde66688 | 系统提示用户名不存在 |
| 使用用户名不正确的账号的登录 | 用户名：kk3 密码：cde66688 | 系统提示用户名不正确 |
| 使用用户名正确以及密码不正确的账号登录 | 用户名：haha123 密码：123 | 系统提示密码错误 |
| 使用用户名正确以及密码正确的账号登录 | 用户名：haha123 密码：y1s1ahaha | 系统提示欢迎haha123用户，页面跳转至首页 |

## 

表6-2 系统安全性测试流程

|  |  |  |
| --- | --- | --- |
| 测试方案 | 测试数据 | 预期结果 |
| 本地的系统管理员登录权限token被修改或者为空时，在网址输入系统管理主界面的路由地址 | http://wqjlb/admin/home | 路由地址被重定向回系统管理登录界面 |
| 本地的网球俱乐部用户登录权限token被修改或者为空时，在网址直接输入网球俱乐部用户主界面的路由地址 | http://wqjlb/mb/main/personal | 路由地址被重定向回用户登录界面 |

## 5.3 测试结果

网球俱乐部系统的管理员以及用户的基本业务操作以及系统安全性的测试都符合预期结果，页面的刷新与跳转运作良好，满足项目既定要求。

# 

# 结论

完成网球俱乐部系统的设计与代码实现之后，此毕业设计论文也接近尾声。在这次的毕业设计论文的研究中，我使用了Vue.js这个最新的前端框架和Node.js这个新兴的服务端语言完成了一个相对完整的前后端分离的网站项目。

在这次的代码编写过程里，我学到了很多编程的知识 ，比如Vue.js的数据驱动的思想，Vue.js靠数据驱动双向绑定使我们开发页面更简单，开发者不需要手动的去修改DOM，Vue.js通过数据双向绑定让页面的一切操作变得更简单。Vue.js的数据驱动就是通过论中提到的MVVM这种框架来实现的。Node.js作为服务端处理前端页面发来的各种请求数据，处理地特别方便，其模块化的操作方式，也许是未来服务端的一大发展方向。

互联网发展的趋势便是更方便快捷，十年前的编程技术和现在的编程技术不可同日耳语，编程技术的飞速发展，必将造福人类，更深层次的改变人们的日常生活。

# 参考文献

[1] Kyle Simpson. You Don't Know JS: Up & Going [J]. O'Reilly Media, 2015: 88

[2] Kyle Simpson. You Don't Know JS: Async & Performance [J]. O'Reilly Media, 2015: 247

[3] Kyle Simpson. You Don't Know JS: ES6 & Beyond [J]. O'Reilly Media, 2015: 260

[4] (美)Nicholas C.Zakas扎卡斯. javascript高级程序设计（第三版）[M]. 人民邮电出版社, 2012: 329.

[5] (美)弗拉纳根. javascript权威指南[M]. 机械工程出版社， 2007：8.

[6] 朴灵. 深入浅出Node.js[M]. 人民邮电出版社， 2013：1201.

[7] 阮一峰. ES6标准入门（第三版）[M]. 电子工业出版社， 2017: 9.

[8] (美)MongoDb权威指南 Kristina Cbodorow （第二版）2011: 5

[9] 郭家寶 Node.js开发指南[M].人民邮电出版社2012: 7

[10](美) [David Gourley](https://book.douban.com/search/David%20Gourley) &&[Brian Totty](https://book.douban.com/search/Brian%20Totty). HTTP: The Definitive Guide [M].人民邮电出版社2012: 9

# 致谢

经过几个月的查资料、整理材料、写作论文，今天终于可以顺利的完成论文的最后的谢辞了。首先，我要感谢广西民族大学相思湖学院的所有教师，是他们在这四年里不辞辛苦地教授我各种知识。我也要感谢石家庄佳诚公司的吴天志老师、杨胜强老师、淡婧娴老师以及刘秀梅老师对于我的栽培，使我在这宝贵的四年中掌握了必要的编程知识，在毕业之际，对同窗四年的计算机科学与工程系2016届软件工程专业的所有同学和好朋友说一句，谢谢你们给予我的心灵支持，我会倍加珍惜这份同学之间共同奋进的友谊。回首四年大学时光，我很骄傲能成为一名广西民族大学相思湖学院的学生，毕业后的日子里，我也要奋力前行。

**附录**

管理端通用型数据接口的关键代码：

module.exports = app =>{

  const express = require('express');

  const router = express.Router({

    mergeParams:true

  });

  const autoMiddleWare=require('../middleware/auto');

  const resourceMiddleWare=require('../middleware/resource');

  const Storage=require('../model/Storage');

  const Temp=require('../model/Temp');

  const assert=require('http-assert');

  const fs=require('fs');

  const path=require('path');

  /\* 管理端通用接口 \*/

  app.use('/admin/api/rest/:resource',autoMiddleWare(),resourceMiddleWare(),router)

  router.get('/', async(req, res) =>{

    let queryOption={};

    let condition={};

    // 搜索

    if(req.query.match){

      const obj=JSON.parse(req.query.match);

      condition[obj.key]=new RegExp(obj.val,'i');

    }

    // 匹配

    if(req.query.where){

      queryOption.where=JSON.parse(req.query.where);

    }

    //分页

    if(req.query.page &&req.query.limit){

      queryOption.limit=Number(req.query.limit)

      queryOption.skip=Number(req.query.page)\*Number(req.query.limit)

    }

    //填充

    if(req.query.populate){

      queryOption.populate=JSON.parse(req.query.populate);

    }

    const items=await req.Model.find(condition).setOptions(queryOption).lean();

    res.send(items)

  });

  router.get('/:id', async(req, res) =>{

    let queryOption={};

    if(req.query.populate){

      queryOption.populate=JSON.parse(req.query.populate);

    }

    const model=await req.Model.findById(req.params.id).setOptions(queryOption).lean();

    res.send(model);

  });

  router.get('/count/page', async(req, res) =>{

    const amount=await req.Model.find().countDocuments();

    res.send(amount)

  });

  router.post('/', async(req, res) =>{

    const model=await req.Model.create(req.body);

    const temp=await Temp.find();

    if(temp){

      await Storage.create(temp.map(v=>{

        return {

          parent:model['\_id'],

          type:v.type,

          filename:v.filename,

        }

      }));

      await Temp.remove({});

    }

    res.send(model)

  });

  router.put('/:id', async(req, res) =>{

    const model=await req.Model.findByIdAndUpdate(req.params.id,req.body);

    const temp=await Temp.find();

    if(temp.length>0){

      const flag=await Storage.create(temp.map(v=>{

        return {

          parent:model['\_id'],

          type:v.type,

          filename:v.filename,

        }

      }));

      assert(flag,500,'操作失败');

      await Temp.remove({});

    }

    res.send( model)

  });

  router.delete('/:id', async(req, res) =>{

    const flag= await req.Model.findByIdAndDelete(req.params.id);

    assert(flag,500,'操作失败');

    const arr=await Storage.find().where({parent:req.params.id});

    if(arr.length>0){

      arr.forEach(item=>{

        fs.unlinkSync(path.resolve(\_\_dirname,'../',`uploads/${item.type?'imgs':'video'}/${item.filename}`));

      })

      await Storage.remove({parent:req.params.id});

    }

    res.send({

      success:'删除成功'

    })

  });

}

用户端首页代码:

<template>

  <div class="article-page">

    <div class="navbar w-100 h40 bg-dark d-flex jc-start ai-center">

      <div

        v-for="(item, index) in navlist"

        :key="index"

        @click="swiperSlide(index)"

        class="ml-3 fs-xl"

      >

        <span

          class="item letter-s1"

          :class="selected === index ? 'active fw' : 'text-l-grey '"

        >

          {{ item }}

        </span>

      </div>

    </div>

    <swiper

      v-if="categories.length > 0"

      :options="{ autoHeight: true }"

      ref="articleSwiper"

      class="bg-white"

      @slide-change="() **=>** (selected = $refs.articleSwiper.swiper.realIndex)"

    >

      <swiper-slide>

        <div class="swiper" v-if="swipers.length != 0">

          <mt-swipe :auto="6000">

            <mt-swipe-item v-for="(item, index) in swipers" :key="index">

              <img :src="item.url" class="swiper-img" />

              <p class="swiper-text text-white fs-l">{{ item.title }}</p>

            </mt-swipe-item>

          </mt-swipe>

        </div>

        <div class="card" v-for="(item, index) in categories" :key="index">

          <router-link

            v-show="item.articles.length > 0"

            :to="{ name: 'ArticleType', params: { id: item['\_id'] } }"

            tag="div"

            class="fs-m p-2 d-flex jc-between ai-center"

          >

            <span class="pl-1 fw fs-l" style="border-left: 2px solid tomato;">

              {{ item.title }}

            </span>

            <span class="fs-1 text-dark-6"

              >更多<i class="iconfont icon-icon-arrow-right2"></i>

            </span>

          </router-link>

          <ArtItem :list="item.articles" :catTitle="item.title"></ArtItem>

        </div>

      </swiper-slide>

      <swiper-slide>

        <div class="bg-grey-1 px-2 py-2 fs-l fw d-flex jc-between letter-s1">

          <div>

            <span class="text-tomato">{{

              matchPage.match ? matchPage.match.title : "不限定赛事类型"

            }}</span>

          </div>

          <div class="text-steel" @click="matchPage.showMatchList = true">

            选择<i class="ml-1 iconfont icon-gengduo"></i>

          </div>

        </div>

        <MyLoadMore

          v-if="matchPage.list.length > 0"

          :loading="matchPage.loading"

          :noMore="matchPage.noMore"

          @loadMore="

            loadMore(matchPage, fetchMatch, matchPage.list, selected, 1)

          "

        >

          <li v-for="item in matchPage.list" :key="item['\_id']">

            <div class="bg-grey-1 fs-l p-2 d-flex jc-between fw">

              <span>

                {{ item.date | formatMDcn }} 星期{{ item.date | formatWK }}

              </span>

              <span>{{ item.date | formatYear }}年{{ item.match.title }}</span>

            </div>

            <div class="d-flex p-2">

              <div class="d-flex flex-coloum jc-center ai-center">

                <span class="fs-l text-black">{{ item.date | formatDM }}</span>

                <span class="fs-m text-dark-6" v-if="item.round"

                  >第{{ item.round }}轮</span

                >

              </div>

              <div class="flex-1 d-flex fs-sm fw text-dark px-1">

                <div class="flex-1 d-flex flex-coloum ai-center jc-center">

                  <p

                    class="text-center"

                    v-for="player in item.host"

                    :key="player['\_id']"

                  >

                    {{ player.cname }}

                  </p>

                </div>

                <div class="d-flex ai-center text-orange">

                  VS

                </div>

                <div class="flex-1 d-flex flex-coloum ai-center jc-center">

                  <p

                    class="text-center"

                    v-for="player in item.guest"

                    :key="player['\_id']"

                  >

                    {{ player.cname }}

                  </p>

                </div>

              </div>

              <div class="d-flex flex-coloum">

                <span

                  :class="

                    item.hostscore > item.guestscore

                      ? 'text-tomato'

                      : 'text-dark-6'

                  "

                  >{{ item.hostscore }}</span

                >

                <span

                  :class="

                    item.guestscore > item.hostscore

                      ? 'text-tomato'

                      : 'text-dark-6'

                  "

                  >{{ item.guestscore }}</span

                >

              </div>

            </div>

          </li>

        </MyLoadMore>

        <MyBlankPage v-else></MyBlankPage>

      </swiper-slide>

      <swiper-slide>

        <div class="d-flex bb-l letter-s1">

          <div

            class="flex-1 d-flex jc-around"

            v-for="(item, index) in dataPage.sortlist"

            :key="index"

            @click="dataPage.selected = index"

          >

            <span

              class="text-dark my-2 b-radius10 px-2"

              :class="dataPage.selected == index ? 'bg-blue text-white' : ''"

              >{{ item.title }}</span

            >

          </div>

        </div>

*<!-- <mt-switch v-model="dataPage.gender"></mt-switch> -->*

        <div>

          <div

            class="bb-l letter-s1 py-1 px-3 d-flex jc-between fs-m text-dark-6"

          >

            <span>ATP球员</span>

            <span>{{ dataPageSelected === 0 ? "积分" : "大满贯" }}</span>

          </div>

          <ul>

            <li

              class="py-2 pl-4 pr-3 d-flex ai-center bb-l"

              v-for="(item, index) in dataPage.ATP"

              :key="index"

              @click="routeByName('player', item['\_id'])"

            >

              <span

                :class="[

                  index === 0 ? 'text-tomato' : '',

                  index === 1 ? 'text-blue' : '',

                  index === 2 ? 'text-green' : '',

                ]"

                >{{ index + 1 }}</span

              >

              <div class="flex-1 text-dark d-flex ai-center">

                <img

                  :src="item.avatar ? item.avatar : defaultHeader"

                  class="w40 h40 mx-3 b-radius50"

                />

                <span class="letter-s1 fs-l">{{ item.cname }}</span>

              </div>

              <span

                class="fs-l"

                :class="[

                  index === 0 ? 'text-tomato fw' : '',

                  index === 1 ? 'text-blue fw' : '',

                  index === 2 ? 'text-green fw' : '',

                ]"

                >{{

                  dataPageSelected === 0 ? item.integral : item.grandslam

                }}</span

              >

            </li>

          </ul>

          <div

            class="bb-l letter-s1 py-1 px-3 d-flex jc-between fs-m text-dark-6"

          >

            <span>WTA球员</span>

            <span>{{ dataPageSelected === 0 ? "积分" : "大满贯" }}</span>

          </div>

          <ul>

            <li

              class="py-2 pl-4 pr-3 d-flex ai-center bb-l"

              v-for="(item, index) in dataPage.WTA"

              :key="index"

              @click="routeByName('player', item['\_id'])"

            >

              <span

                :class="[

                  index === 0 ? 'text-tomato fw' : '',

                  index === 1 ? 'text-blue fw' : '',

                  index === 2 ? 'text-green fw' : '',

                ]"

                >{{ index + 1 }}</span

              >

              <div class="flex-1 text-dark d-flex ai-center">

                <img

                  :src="item.avatar ? item.avatar : defaultHeader"

                  class="w40 h40 mx-3 b-radius50"

                />

                <span class="letter-s1 fs-l">{{ item.cname }}</span>

              </div>

              <span

                class="fs-l"

                :class="[

                  index === 0 ? 'text-tomato fw' : '',

                  index === 1 ? 'text-blue fw' : '',

                  index === 2 ? 'text-green fw' : '',

                ]"

                >{{

                  dataPageSelected === 0 ? item.integral : item.grandslam

                }}</span

              >

            </li>

          </ul>

        </div> </swiper-slide

      ><swiper-slide>

*<!--   -->*

        <MySearch

          :search="playerPage.search"

          @inputSearch="searchPlayer"

        ></MySearch>

        <MyLoadMore

          v-if="playerPage.list.length > 0"

          :flexWrap="true"

          :loading="playerPage.loading"

          :noMore="playerPage.noMore"

          @loadMore="

            loadMore(playerPage, fetchPlayer, playerPage.list, selected, 3)

          "

        >

          <li

            class="w-50 p-2"

            v-for="player in playerPage.list"

            :key="player['\_id']"

            @click="routeByName('player', player['\_id'])"

          >

            <div class="px-2">

              <img

                :src="player.avatar ? player.avatar : default\_img"

                class="w-100 h100"

              />

              <p class="text-over fs-l fw text-center">{{ player.cname }}</p>

            </div>

          </li>

        </MyLoadMore>

        <MyBlankPage v-else></MyBlankPage>

      </swiper-slide>

    </swiper>

    <mt-popup

      v-model="matchPage.showMatchList"

      style="height: 100vh; width: 70vw;"

      position="right"

    >

      <div class="d-flex px-3 flex-coloum" style="overflow: scroll;">

        <p class="fw my-3">赛事名称</p>

        <div

          class="bborder b-radius5 mb-2 p-2 fs-l text-center"

          v-for="item in matchPage.matches"

          :key="item['\_id']"

          @click="selectMatch(item)"

          :class="

            matchPage.match == item ? 'bg-orange text-white' : 'text-dark'

          "

        >

          {{ item.title }}

        </div>

      </div>

*<!-- <div class="d-flex jc-center ai-start px-5" style="height: 3rem;">*

*<mt-button type="primary" size="large" class="fw letter-s1"*

*>确认</mt-button*

*>*

*</div> -->*

    </mt-popup>

  </div>

</template>

<script>

import MySearch from "../../components/MySearch";

import ArtItem from "../../components/ArtItem";

import { mapState } from "vuex";

export default {

  components: {

    MySearch,

    ArtItem,

  },

  data() {

    return {

      selected: 0,

      categories: [],

      swipers: [],

      ranks: [],

      navlist: ["推荐", "赛事", "数据", "球员"],

      dataPage: {

        selected: 0,

        ATP: [],

        WTA: [],

        sortlist: [

          {

            title: "积分排名",

            sortname: "integral",

          },

          {

            title: "大满贯",

            sortname: "grandslam",

          },

        ],

      },

      matchPage: {

        showMatchList: false,

        limit: 6,

        page: 0,

        list: [],

        matches: [],

        match: null,

        loading: false,

        noMore: false,

      },

      playerPage: {

        search: "",

        list: [],

        limit: 10,

        page: 0,

        loading: false,

        noMore: false,

      },

    };

  },

  computed: {

    dataPageSelected() {

      return this.dataPage.selected;

    },

  },

  watch: {

    dataPageSelected() {

      this.fetchRank();

    },

  },

**async** created() {

    this.fetchSwipers();

    this.fetchCategories();

    this.fetchRank();

    this.initMatchPage();

    this.initPlayerPage();

  },

  methods: {

**async** initMatchPage() {

      this.matchPage.page = 0;

      this.matchPage.noMore = false;

      this.matchPage.matches = await this.fetchMatchStyle();

      this.matchPage.list = await this.fetchMatch();

    },

**async** initPlayerPage() {

      this.playerPage.page = 0;

      this.playerPage.noMore = false;

      this.playerPage.list = await this.fetchPlayer();

    },

**async** fetchCategories() {

      this.categories = await this.fetch("categories/all");

    },

**async** fetchSwipers() {

      this.swipers = await this.fetch("rest/ads");

    },

    swiperSlide(index) {

      this.$refs.articleSwiper.swiper.slideTo(index);

    },

**async** fetchMatchStyle() {

      return await this.fetch("rest/matches", {

        sort: {

          title: -1,

        },

      });

    },

**async** fetchMatch() {

**const** option = this.matchPage;

**const** queryOption = {

        limit: option.limit,

        page: option.page,

        sort: {

          date: -1,

        },

        populate: {

          path: "host guest match",

        },

      };

      if (this.matchPage.match) {

        queryOption.where = {

          match: this.matchPage.match["\_id"],

        };

      }

      return await this.fetch("rest/contest", queryOption);

    },

**async** fetchRank() {

**const** obj = this.dataPage;

**const** ATPOptions = {};

      ATPOptions[obj.sortlist[obj.selected].sortname] = -1;

**const** ATPquery = {

        where: {

          gender: true,

        },

        limit: 20,

        page: 0,

        sort: ATPOptions,

      };

**const** WTAquery = {

        where: {

          gender: false,

        },

        limit: 20,

        page: 0,

        sort: ATPOptions,

      };

      if (this.dataPageSelected === 0) {

        ATPquery.where.retire = false;

        WTAquery.where.retire = false;

      }

      this.dataPage.ATP = await this.fetch("rest/players", ATPquery);

      this.dataPage.WTA = await this.fetch("rest/players", WTAquery);

    },

**async** fetchPlayer(text) {

      return await this.fetch("rest/players", {

        match: {

          key: "cname",

          val: text,

        },

        limit: this.playerPage.limit,

        page: this.playerPage.page,

      });

    },

**async** selectMatch(item) {

      this.matchPage.match && this.matchPage.match["\_id"] == item["\_id"]

        ? (this.matchPage.match = null)

        : (this.matchPage.match = item);

      this.matchPage.showMatchList = false;

      this.matchPage.noMore = false;

      this.matchPage.page = 0;

      this.matchPage.list = await this.fetchMatch();

    },

**async** searchPlayer(text = "") {

      this.playerPage.list = await this.fetchPlayer(text);

    },

  },

};

</script>

<style lang="scss">

@import "@/assets/css/config.scss";

.article-page {

  background-color: map-get($map: $colors, $key: "bg");

  .swiper {

    width: 100%;

    height: 13rem;

    background-color: map-get($map: $colors, $key: "white");

    .mint-swipe-items-wrap {

*// border-radius: 8px;*

      .mint-swipe-item {

        .swiper-img {

          position: absolute;

          width: 100%;

          height: 100%;

          z-index: 5;

          background-size: cover;

        }

        .swiper-text {

          position: absolute;

          left: 0.5rem;

          letter-spacing: 2px;

          bottom: 0.6rem;

          z-index: 99;

        }

      }

    }

  }

  .active {

    color: map-get($map: $colors, $key: "white");

    border-bottom: 3px solid map-get($map: $colors, $key: "blue-1") !important;

  }

  .card {

    border-bottom: 0.5rem solid #f0f0f0;

  }

}

</style>