

BEIJINGINSTITUTEOFPETROCHEMICALTECHNOLOGY

大数据存储实践

小组报告

League of Legends赛事数据管理系统实验报告

|  |  |
| --- | --- |
| 团 队 组 号： | 4 |
| 组 长： | 张今朝 |
| 组 员： | 周一凡、刘宇恒、侯志昊 |
| 指 导 教 师： | 苏艳 |
| 提 交 时 间： | 2022.12.6 |

**目录**

[1需求分析 3](#_Toc121198101)

[1.1系统开发背景、目的及意义 3](#_Toc121198102)

[1.2功能需求分析 3](#_Toc121198103)

[2系统架构设计 3](#_Toc121198104)

[2.1系统框架设计 4](#_Toc121198105)

[2.1.1前端部分 4](#_Toc121198106)

[2.1.2后端部分 9](#_Toc121198107)

[2.2数据存储架构设计 14](#_Toc121198108)

[2.2.1数据库的选择 14](#_Toc121198109)

[2.2.2数据存储位置 15](#_Toc121198110)

[2.2.3数据存储文档结构 15](#_Toc121198111)

[3系统详细功能设计与实现 17](#_Toc121198112)

[3.1系统运行环境及配置 17](#_Toc121198113)

[3.2分片功能实现 21](#_Toc121198115)

[3.3赛事数据来源 28](#_Toc121198116)

[3.4系统功能实现 28](#_Toc121198117)

[3.4.1预约比赛战队 28](#_Toc121198118)

[3.4.2添加赛事数据 29](#_Toc121198119)

[3.4.3赛事数据管理 30](#_Toc121198120)

[4总结 31](#_Toc121198121)

**1需求分析**

**1.1系统开发背景、目的及意义**

在现代硬件、软件的飞速发展的背景下，人们的物质需求日益满足，科学技术水平日益提高，在人们生活中文化娱乐的占比越来越高。因此不少游戏横空出世，随之电竞行业应运而生，而家长们却把电竞钉上了“误人子弟”的标签。早在2003年11月，电子竞技就已经被体育总局承认为我国正式开展的第99个运动项目。而在2008年，电子竞技被体育总局改批为第78号正式体育竞赛项。尽管电竞行业已经得到国家正视，但是出于人们对于电子竞技的传统观念束缚，所以电竞行业依旧没有很大的起色。直到2013年3月，体育总局决定组建电子竞技国家队后，在各大主流媒体上引起了人们对电子竞技的强烈关注。各种电竞赛事如同雨后春笋一般一个接一个出现。随着电竞行业的不断扩大、规章制度越来越规范、各路媒体以及社会越来越多的关注，电子竞技的影响力也是越来越高，其商业化、规范化也是必然结果。

《英雄联盟》（League of Legends，简称LOL）是由美国拳头游戏（Riot Games）开发、中国内地由腾讯游戏代理运营的英雄对战MOBA竞技网游，自从2013年开始举办LPL春/夏季联赛后，英雄联盟的热度在国内不断提高，关注人群规模庞大，影响群体越来越多，同时随着更多资本入局，联赛奖金的上调以及场馆的建设和周边产品的推出，使得这一游戏已经成为了中国电竞行业的重要组成，这也让更多的赛事选择在中国举办，而中国战队在国际赛事上的表也是精彩纷呈，在S8、S9以及S11三次获得英雄联盟全球总决赛的冠军，中国选手的身价的不断上升，逐渐开始加盟外国战队走向世界，英雄联盟是划时代的产物，它现在的高度不止局限于游戏，它正在慢慢演变成一种潮流、一种文化、一种精神，这也从侧面佐证了我国电竞行业的崛起趋势。

每一年LPL都会举办两季比赛，春季赛和夏季赛。每季比赛又各分为常规赛和季后赛两部分。每年夏季赛结束之后会举办全球总决赛资格赛，选拔当年英雄联盟全球总决赛的参赛队伍。由于每一年都会迎来转会期，所以每个战队在一年后的参赛选手和竞技状态都不同，导致每一年的战队排名也有所变化。

因此设计出一个健全的赛事数据管理系统是非常有必要的，该系统可从头到尾追踪一场比赛的完整流程，从预约比赛战队——添加红蓝方战队以及双方首发队员、添加赛事数据——添加双方比赛数据直至赛事数据管理——通过赛事编号展示对应的战队，成员以及数据。

**1.2功能需求分析**

预约比赛战队：支持添加赛程，包括赛程名称、赛程阶段、赛程时间、双方战队以及首发队员。

添加赛事数据：添加双方战队每场比赛的相关数据，包括击杀、大龙、小龙、防御塔、金钱以及获胜情况。

赛事数据管理：通过赛事编号查询整条赛事数据，显示对应编号的赛事信息，战队，成员以及赛事数据。

**2系统架构设计**

**2.1系统框架设计**

系统引擎使用“Spring”，框架使用“SpringMVC”，并配合“SpringBoot”快速开发整合包搭建系统运行环境；使用Java语言进行后端开发，系统前端部分整体使用layui框架，采用模块化规范，遵循HTML/CSS/JS 的开发方式，组成SpringBoot+Maven+MongoDB的系统架构。

本系统架构分为前端和后端，即客户端和服务器端。而使用系统的管理者和选手即是用户，系统的功能都是基于客户端、服务器端和数据库交互实现的。

系统实现的结构如图所示:

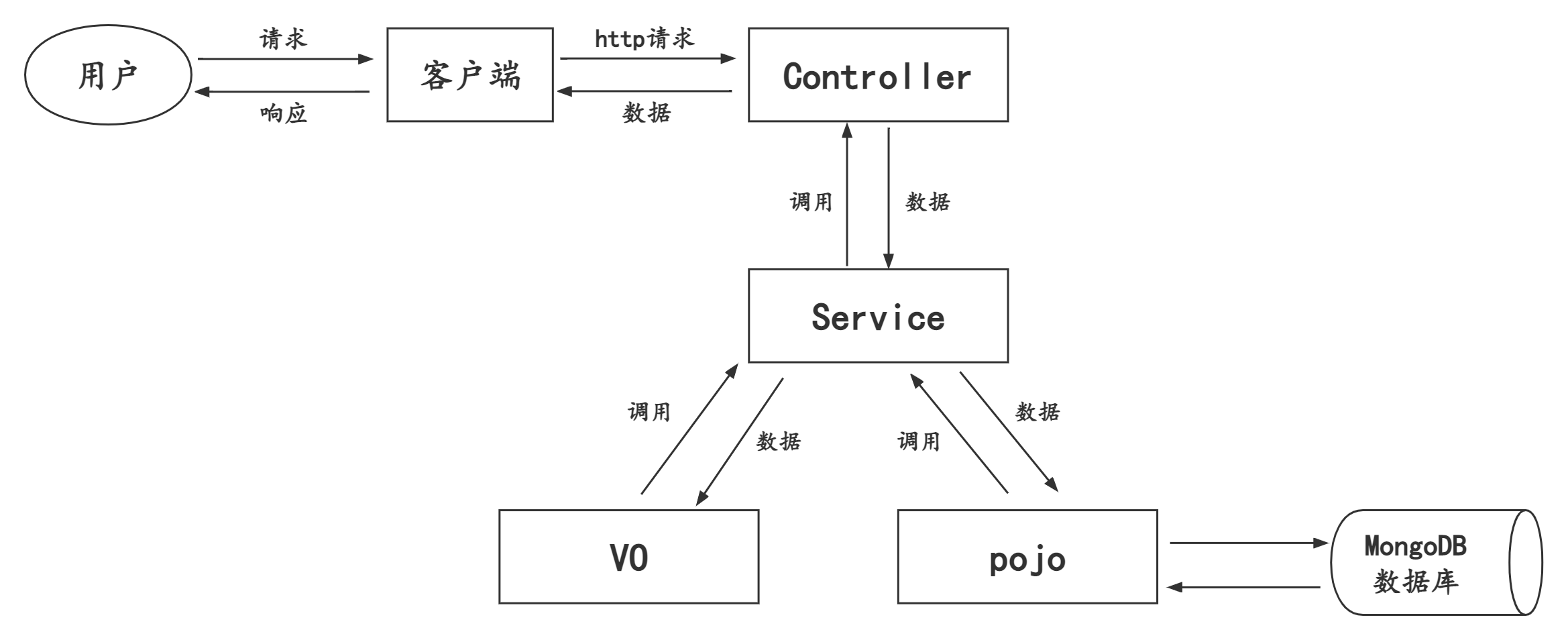


图 1 系统实现结构

2.1.1前端部分

系统从宏观角度仅仅为单个页面，使用户界面结构更加简洁清晰，采用iframe内联框架，其他功能页面内嵌进iframe框架，共计四个页面。

* **index.html**

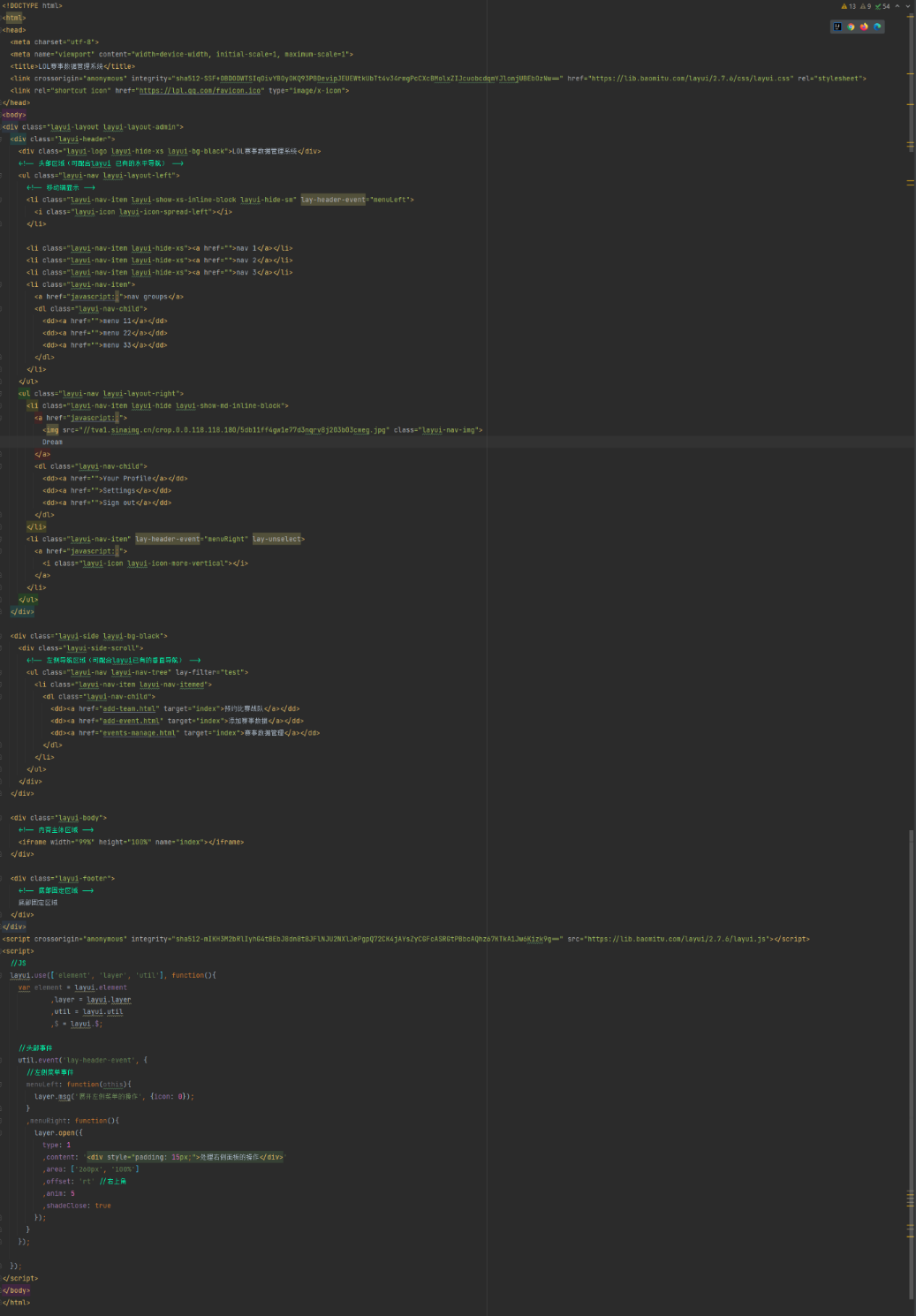


图 2 index.html

* **add-team.html**

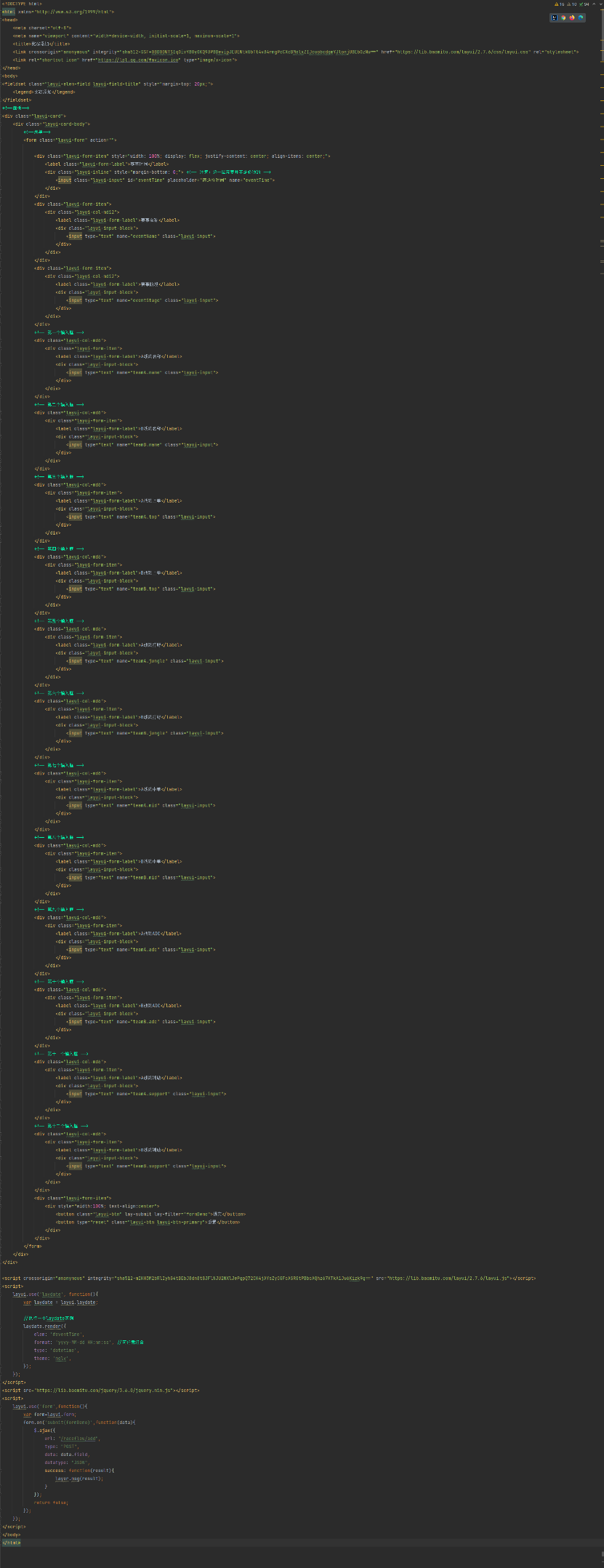


图 3 add-team.html

* **add-event.html**



图 4 add-event.html

* **events-manage.html**



图 5 events-manage.html

2.1.2后端部分

本系统后端共包含7个类，Model层用于存放实体类，构建数据结构，利于数据存储和数据处理，并与数据库中的属性值基本保持一致；RaceFlowController类用于接受前端的传来的参数，通过调用service层，来实现数据的编辑，然后再返回；RaceFlowService类用于存放业务逻辑处理，调用Model层并提供给controller层使用，间接与数据库连接；RaceFlowVO类用于将前端需要的数据做封装，方便前端获取数据；

* **Model层（数据库实体层）：**

RaceFlow类

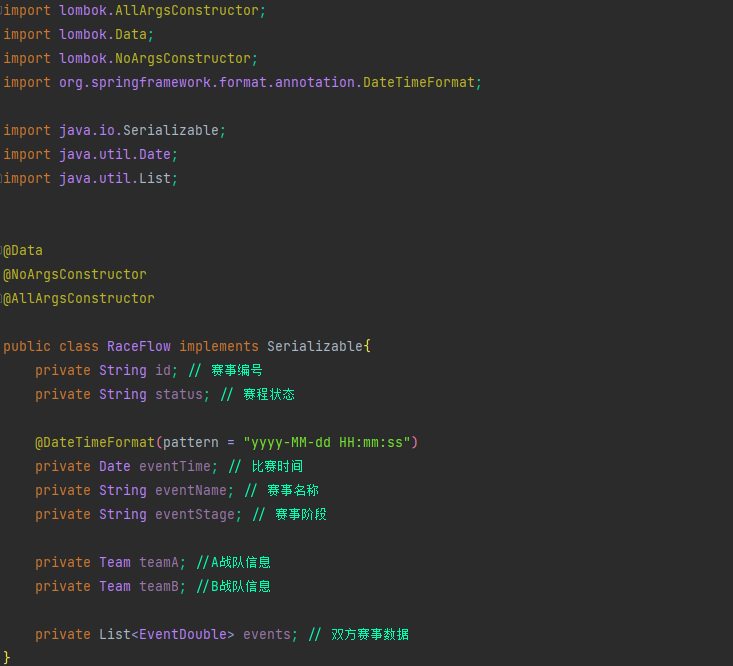


图 6 RaceFlow.java

Team类

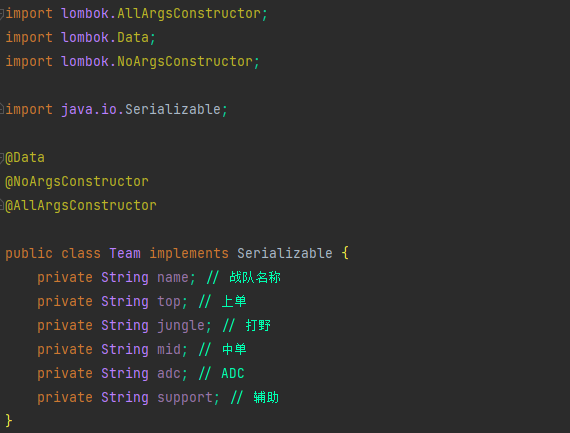


图 7 Team.java

Event类

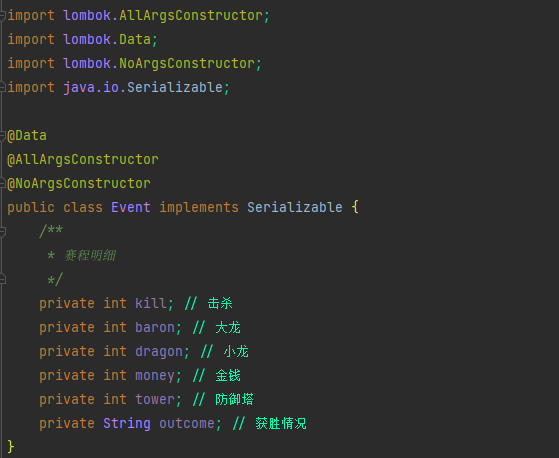


图 8 Event.java

EventDouble类

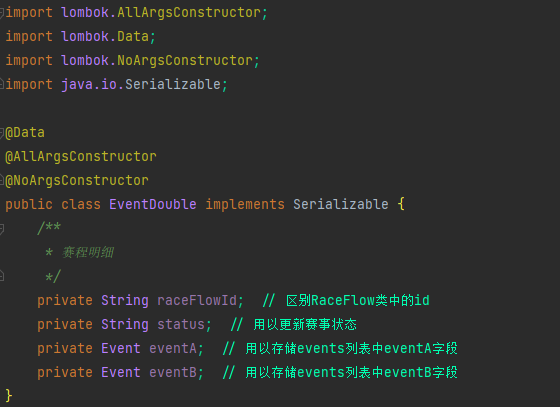


图 9 EventDouble.java

* **VO层（数据展示层）：**

RaceFlowVO类

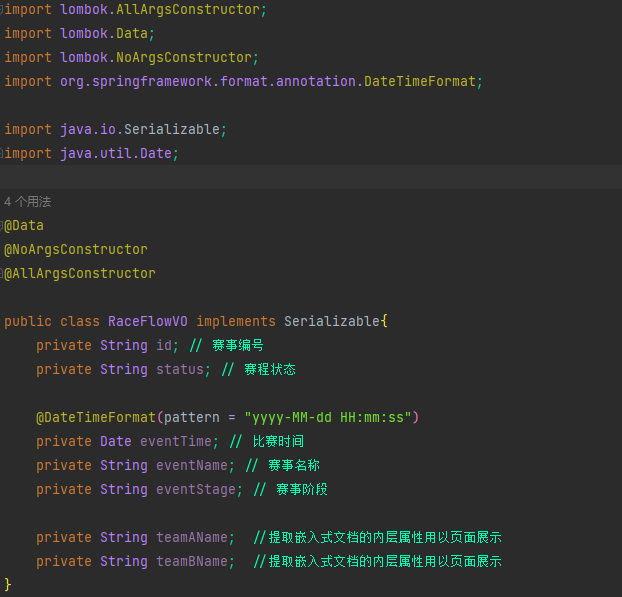


图 10 RaceFlowVO.java

* **Service层（服务层）：**

RaceFlowService类



图 11 RaceFlowService.java

* **Controller层（控制层）：**

RaceFlowController类

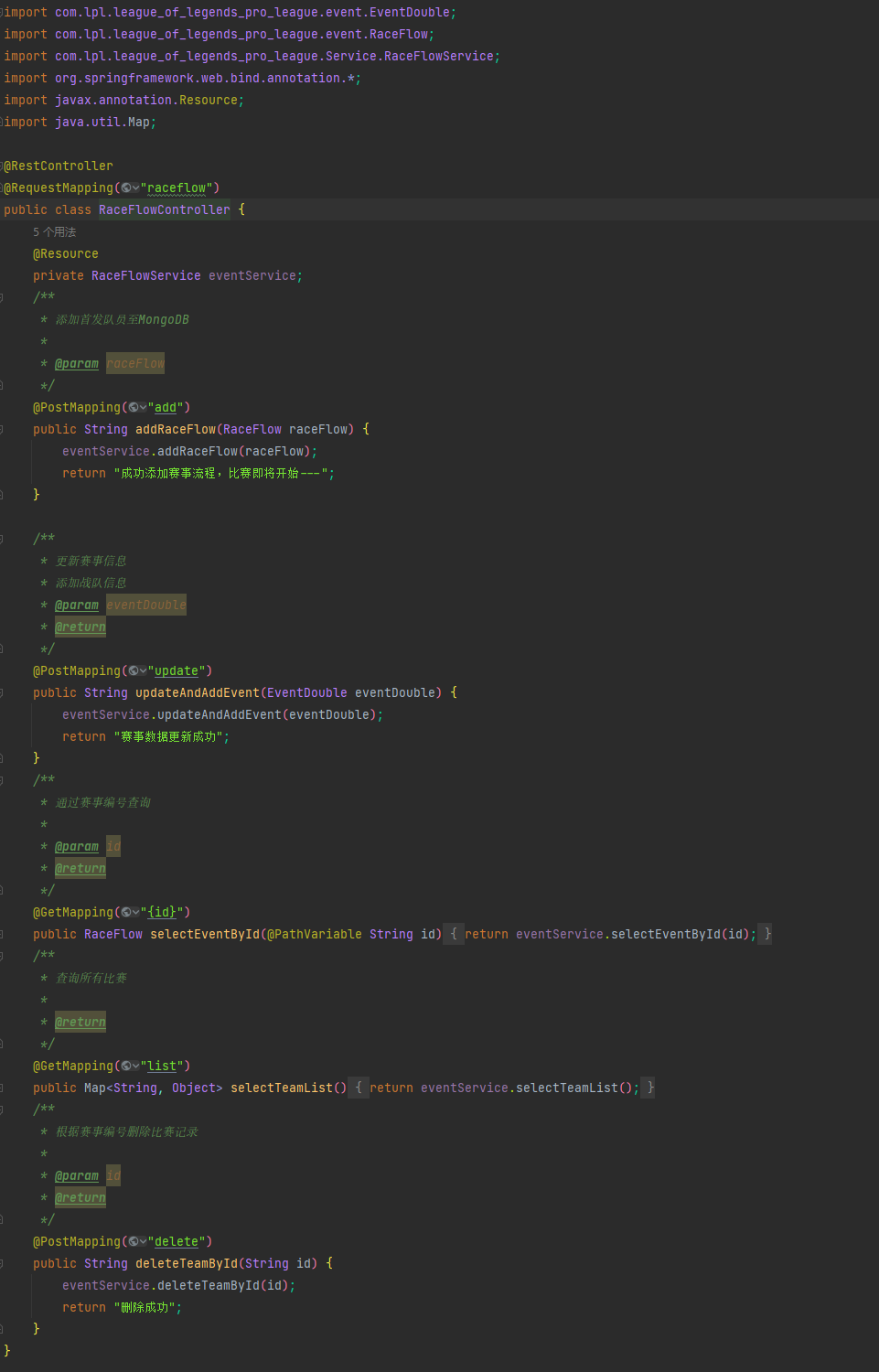


图 12 RaceFlowController.java

**2.2数据存储架构设计**

2.2.1数据库的选择

Mongodb是一款性能优良，功能丰富，易于扩展的文档型非关系型数据库。

优势：

1. 易用性

MongoDB是一个面向文档（document-oriented）的数据库，而不是关系型数据库。

不采用关系型主要是为了获得更好得扩展性。当然还有一些其他好处，与关系数据库相比，面向文档的数据库不再有“行”（row）的概念取而代之的是更为灵活的“文档”（document）模型。

通过在文档中嵌入文档和数组，面向文档的方法能够仅使用一条记录来表现复杂的层级关系，这与现代的面向对象语言的开发者对数据的看法一致。

另外，不再有预定义模式（predefined schema）：文档的键（key）和值（value）不再是固定的类型和大小。由于没有固定的模式，根据需要添加或删除字段变得更容易了。通常由于开发者能够进行快速迭代，所以开发进程得以加快。而且，实验更容易进行。开发者能尝试大量的数据模型，从中选一个最好的。

1. 容器级特性的支持

应用程序数据集的大小正在以不可思议的速度增长。随着可用带宽的增长和存储器价格的下降，即使是一个小规模的应用程序，需要存储的数据量也可能大的惊人，甚至超出了很多数据库的处理能力。过去非常罕见的T级数据，现在已经是司空见惯了。

由于需要存储的数据量不断增长，开发者面临一个问题：应该如何扩展数据库，分为纵向扩展和横向扩展，纵向扩展是最省力的做法，但缺点是大型机一般都非常贵，而且当数据量达到机器的物理极限时，花再多的钱也买不到更强的机器了，此时选择横向扩展更为合适，但横向扩展带来的另外一个问题就是需要管理的机器太多。

MongoDB的设计采用横向扩展。面向文档的数据模型使它能很容易地在多台服务器之间进行数据分割。MongoDB能够自动处理跨集群的数据和负载，自动重新分配文档，以及将用户的请求路由到正确的机器上。这样，开发者能够集中精力编写应用程序，而不需要考虑如何扩展的问题。如果一个集群需要更大的容量，只需要向集群添加新服务器，MongoDB就会自动将现有的数据向新服务器传送

1. 丰富的功能

MongoDB作为一款通用型数据库，除了能够创建、读取、更新和删除数据之外，还提供了一系列不断扩展的独特功能。

索引

支持通用二级索引，允许多种快速查询，且提供唯一索引、复合索引、地理空间索引、全文索引

聚合

支持聚合管道，用户能通过简单的片段创建复杂的集合，并通过数据库自动优化

特殊的集合类型

支持存在时间有限的集合，适用于那些将在某个时刻过期的数据，如会话session。类似地，MongoDB也支持固定大小的集合，用于保存近期数据，如日志

文件存储

支持一种非常易用的协议，用于存储大文件和文件元数据。MongoDB并不具备一些在关系型数据库中很普遍的功能，如链接join和复杂的多行事务。省略这些的功能是处于架构上的考虑，或者说为了得到更好的扩展性，因为在分布式系统中这两个功能难以高效地实现

1. 卓越的性能

MongoDB的一个主要目标是提供卓越的性能，这很大程度上决定了MongoDB的设计。MongoDB把尽可能多的内存用作缓存cache，视图为每次查询自动选择正确的索引。

总之各方面的设计都旨在保持它的高性能

虽然MongoDB非常强大并试图保留关系型数据库的很多特性，但它并不追求具备关系型数据库的所有功能。只要有可能，数据库服务器就会将处理逻辑交给客户端。这种精简方式的设计是MongoDB能够实现如此高性能的原因之一。

本项目内容繁多，数据类型复杂，对于赛事数据信息的存储，赛事列表需要不断的添加新的记录，假设后期赛制更新需求变更需要添加字段，对于关系新数据库又需要重新建立表格，很繁琐，但是对于MongoDB就可做到随时添加随时使用，因此将MongoDB作为数据库是优质且高效的选择。

2.2.2数据存储位置

在Windows服务器（Windows Server2012RS镜像）部署MongoDB服务并实现了MongoDB分片功能：

configServer（123.57.50.204:27019/123.57.50.204:27020、123.57.50.204:27021）

shard（123.57.50.204:3000、123.57.50.204:3001、123.57.50.204:3002）

mongos（123.57.50.204:27010）

2.2.3数据存储文档结构

{

\_id: 字符串类型雪花算法生成id,

status: 最后一小场的赛程状态,

eventTime: 赛程时间,

eventName: "2022职业联赛",

eventStage: "全球总决赛资格赛",

teamA: {

name: 战队名称,

top: 战队上单,

jungle: 战队打野,

mid: 战队中单,

adc: 战队ADC,

support: 战队辅助

},

teamB: {

name: 战队名称,

top: 战队上单,

jungle: 战队打野,

mid: 战队中单,

adc: 战队ADC,

support: 战队辅助

},

\_class: "com.lol.loleventdatamanagement.event.RaceFlow",

events: [

{

raceFlowId: 与id一致,

status: "进行中",

eventA: {

kill: 击杀数,

baron: 大龙数,

dragon: 小龙数,

money: 金钱,

tower: 防御塔数,

outcome: 获胜情况

},

eventB: {

kill: 击杀数,

baron: 大龙数,

dragon: 小龙数,

money: 金钱,

tower: 防御塔数,

outcome: 获胜情况

},

\_class: "com.lol.loleventdatamanagement.event.EventDouble"

}

……

]

}

**3系统详细功能设计与实现**

3.1系统运行环境及配置

框架：SpringMVC+Layui+Maven+MongoDB

开发语言：Java、HTML5、CSS3、JavaScript、Python

JDK：17

Python：3.10.7

Maven：3.6.3

数据库与数据库可视化管理工具：MongoDB4.2.2 、Robo 3T1.3.1

开发工具：Intellij IDEA(Ultimate Edition)2022.2.1

PyCharm(Professional Edition)2022.1.1

浏览器：Microsoft Edge107.0.1418.62 (正式版本) (64 位)

涉及到的技术：MongoDB、Springboot、lombok、layui、spider

3.1.1配置Spring boot运行环境

3.1.1.1创建项目

* 在IDEA中新建项目，生成器选择Spring Initializr。其中语言选择Java，类型选择Maven，组与工件设置好后，名称与软件包名称随之变化。JDK版本大于等于Java版本，点击下一步。

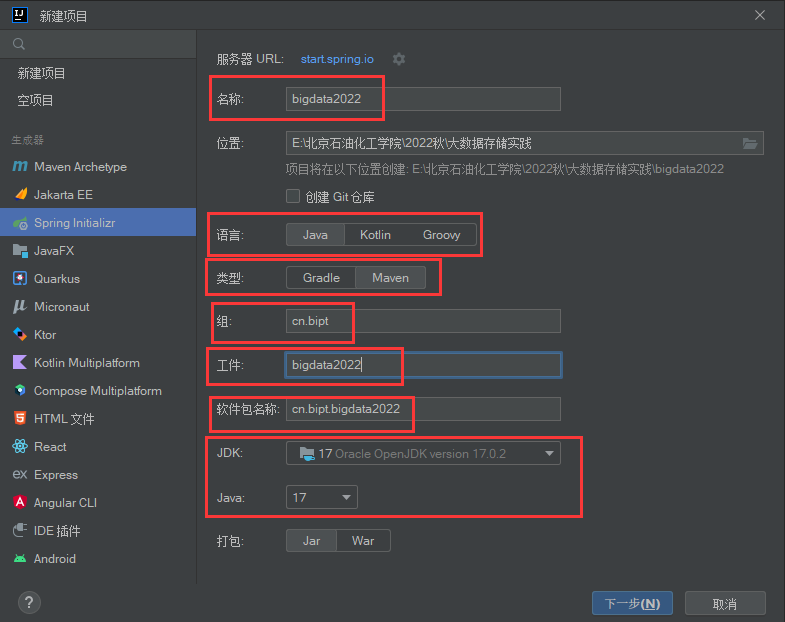


图 13 创建项目

* Spring Boot版本默认为2.7.5，依赖项：Developer Tools选择Lombok，Web选择Spring Web，NoSQL选择Spring Data MongoDB，点击创建。

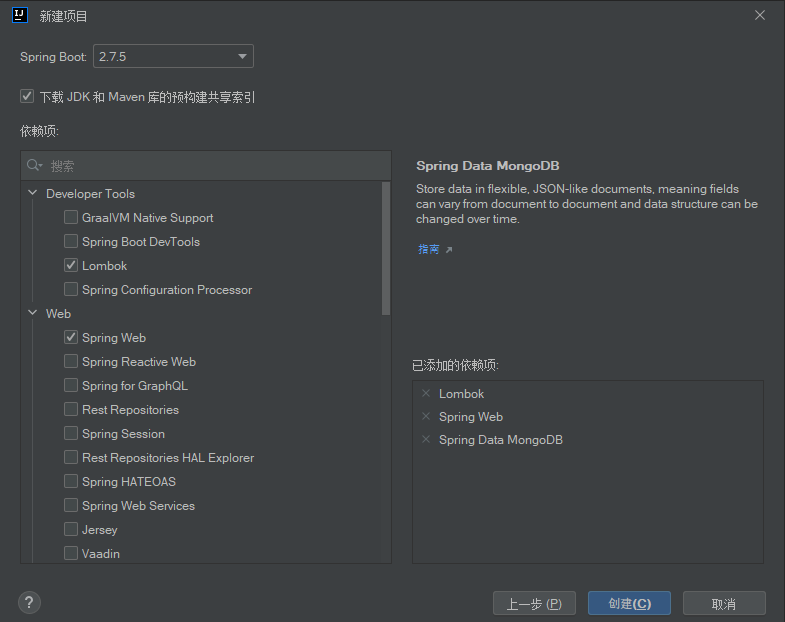


图 14 依赖项

3.1.1.2导入并配置Maven

* IDEA会自动同步该项目的依赖项与插件并导入Maven，等待即可。

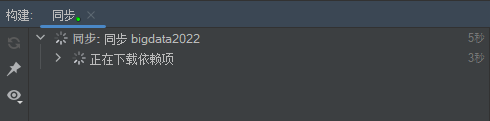


图 15 导入Maven

* 进入IDEA设置，依次进入“构建、执行、部署——构建工具——Maven”，设置Maven主路径、用户设置文件、本地仓库。

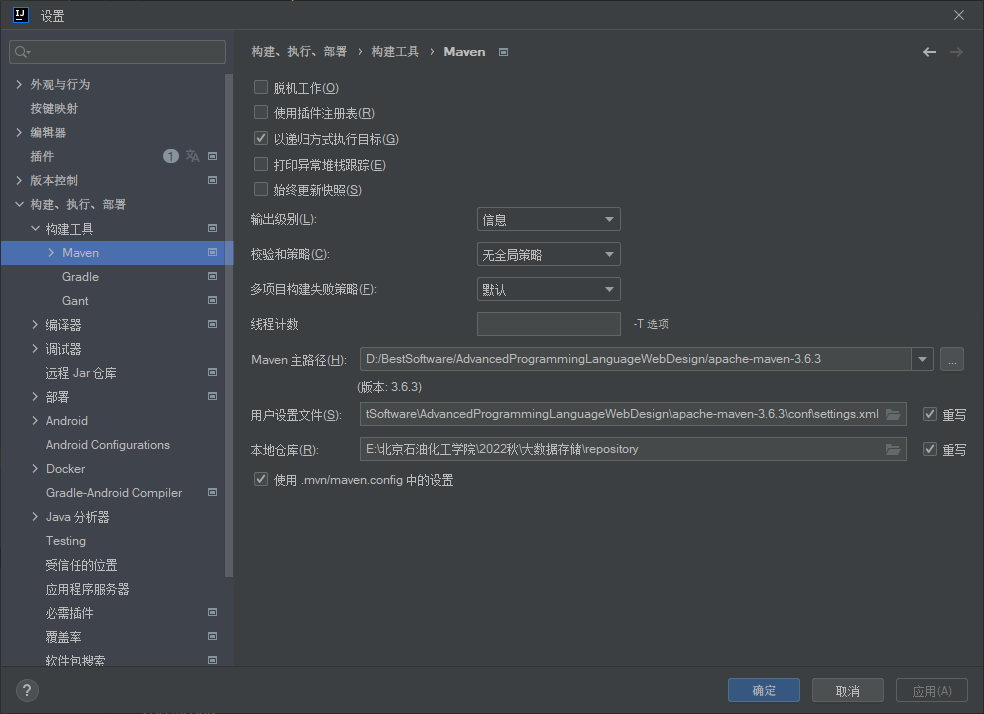


图 16 构建、执行、部署

* 在pom.xml中的dependencies标签下追加以下内容：

<dependency>

<groupId>cn.hutool</groupId>

<artifactId>hutool-all</artifactId>

<version>5.6.3</version>

</dependency>

* 在pom.xml中的build标签下有plugin标签，此处可能会报错，可在artifactId标签后加入version标签，填入2.7.5，点击右上角的变更按钮，即可完成插件同步。

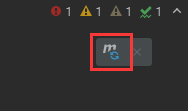


图 17 变更按钮



图 18 插件同步

3.1.1.3配置MongoDB

* 在项目根目录下的src目录下存在resources文件夹，双击文件夹下的application.properties文件，输入spring.data.mongodb.uri = mongodb://123.57.50.204:27010/LeagueofLegends。



图 19 配置MongoDB

* 到此，Spring Boot的运行环境已经搭建完成，下面将进行前后端项目的部署与页面实现。

3.2分片功能实现

根据数据存储位置，选择Windows Server2012R2服务器部署分片集群，下载并安装Windows版本MongoDB，在根目录下创建部署分片的目录，data与logs，创建好之后，在bin目录下运行cmd，输入如下命令用以开启服务：

1. mongod --port 3000 --bind\_ip 0.0.0.0 --shardsvr --replSet rs-a --dbpath "C:\Program Files\mongodb\_4.2.2\shardcluster\data\rs0" --logpath "C:\Program Files\mongodb\_4.2.2\shardcluster\log\rs0.log"

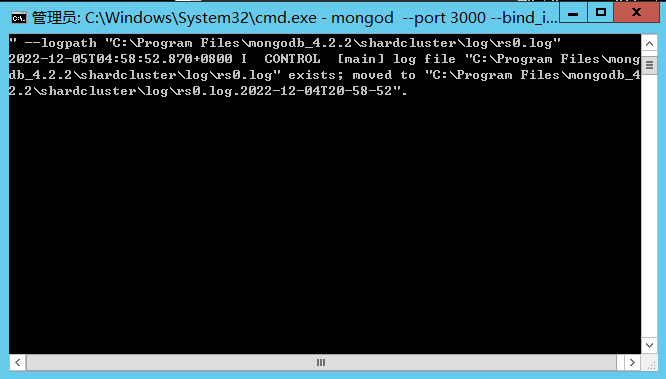


图 20 shard1

1. mongod --port 3001 --bind\_ip 0.0.0.0 --shardsvr --replSet rs-a --dbpath "C:\Program Files\mongodb\_4.2.2\shardcluster\data\rs1" --logpath "C:\Program Files\mongodb\_4.2.2\shardcluster\log\rs1.log"

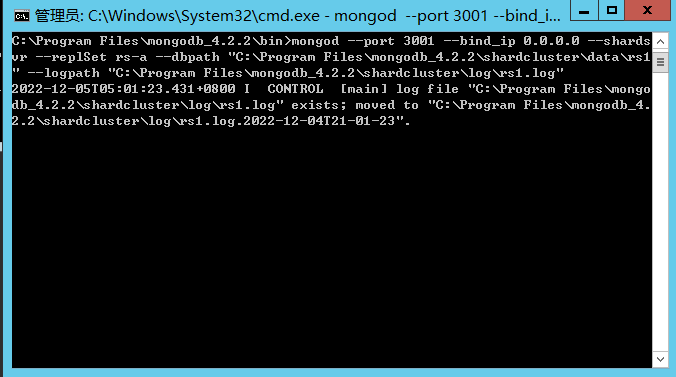


图 21 shard2

1. mongod --port 3002 --bind\_ip 0.0.0.0 --shardsvr --replSet rs-a --dbpath "C:\Program Files\mongodb\_4.2.2\shardcluster\data\rs2" --logpath "C:\Program Files\mongodb\_4.2.2\shardcluster\log\rs2.log"



图 22 shard3

1. mongod --port 27019 --bind\_ip 0.0.0.0 --configsvr --replSet rs-conf --dbpath "C:\Program Files\mongodb\_4.2.2\shardcluster\data\config0" --logpath "C:\Program Files\mongodb\_4.2.2\shardcluster\log\config0.log"

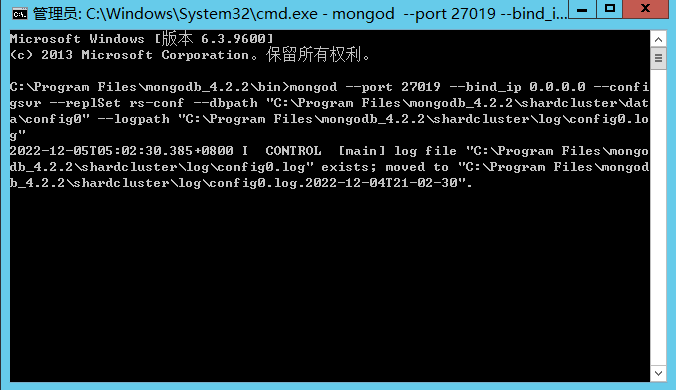


图 23 configserver1

1. mongod --port 27020 --bind\_ip 0.0.0.0 --configsvr --replSet rs-conf --dbpath "C:\Program Files\mongodb\_4.2.2\shardcluster\data\config1" --logpath "C:\Program Files\mongodb\_4.2.2\shardcluster\log\config1.log"

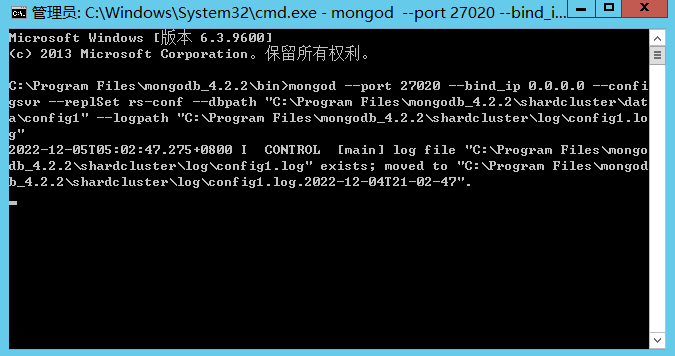


图 24 configserver2

1. mongod --port 27021 --bind\_ip 0.0.0.0 --configsvr --replSet rs-conf --dbpath "C:\Program Files\mongodb\_4.2.2\shardcluster\data\config2" --logpath "C:\Program Files\mongodb\_4.2.2\shardcluster\log\config2.log"



图 25 configserver3

1. mongos --configdb rs-conf/localhost:27019,localhost:27020,localhost:27021 --logpath "C:\Program Files\mongodb\_4.2.2\shardcluster\log\mongos.log" --port 27010 --bind\_ip 0.0.0.0

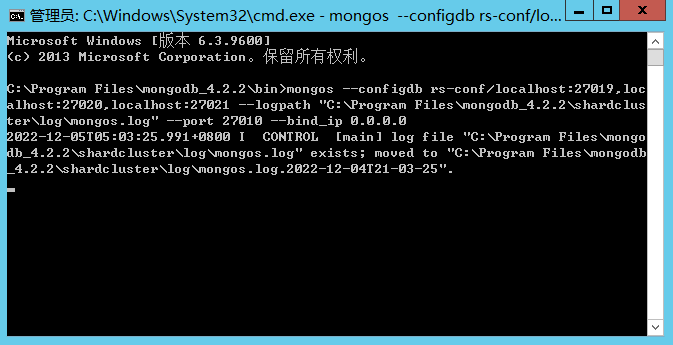


图 26 mongos

随后登录configserver、shard主节点的客户端，进行初始化与添加节点，再登录mongos客户端，添加分片节点；

1. **configserver：**



图 27 configserver client primary

1. **shard：**



图 28 shard client primary

1. **mongos：**



图 29 mongos client

然后输入use config，db.settings.save({"\_id":"chunksize","value":1})；接下来创建LeagueofLegends数据库，在该数据库写入文档；

实现数据库分片功能，sh.enableSharding("LeagueofLegends")，为集合user创建索引，use LeagueofLegends，db.user.createIndex({"id":1})；以索引“id”作为分片键，对集合user进行分片操作，use gateway、sh.shardCollection("LeagueofLegends.user",{"id":1})；

最后在项目中的application.properties配置为分片集群的路由节点，连接至mongos：spring.data.mongodb.uri=mongodb://123.57.50.204:27010/LeagueofLegends即可。

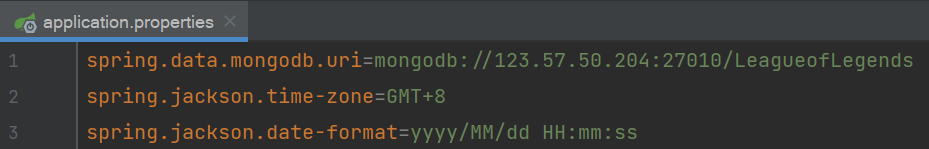


图 30 application.properties

3.3赛事数据来源

为使本项目拥有一定的数据基础，本小组利用Python进行爬取数据，赛事数据来源于https://lpl.qq.com/es/lpl/2022/，采用了selenium模块对网站信息进行获取，首先向网页发起请求并通过网页链接访问相应的页面，运用selenium第三方库爬取相关数据，由于目标数据分布在单页面中（使用JavaScript单页放置元素）导致异步加载，因此需要依靠msedgedriver来模拟鼠标操作，同时，该手段需要依赖selenium.webdriver.common.by的子模块By来定位元素标签，使用browers.find\_element方法来定位元素，By所支持的定位器分类如下：

* CLASS\_NAME = 'class name'
* CSS\_SELECTOR = 'css selector'
* ID = 'id'；LINK\_TEXT = 'link text'
* NAME = 'name'
* PARTIAL\_LINK\_TEXT = 'partial link text'
* TAG\_NAME = 'tag name'
* XPATH = 'xpath'

在多次踩坑之后，解决了模拟鼠标操作之后页面不稳定以及页面焦点不在新页面上的问题，即使用time.sleep方法稳定跳转后的页面。在selenium模块爬取过程中，使用BeautifulSoup的find与find＿all方法来查找元素。在所有场次的数据采集过程中，嵌套了tqdm进度条模块，使得爬取过程可视化，更加直观、清晰。使用time.perf\_counter方法记录程序开始与结束时间，用于统计程序执行耗费的时间。在获取数据以后，使用json.loads()函数将str对象转换为dict（字典）对象，再使用字典和列表的方法就可以将想要的数据提取出来，并且存放到另外一个字典里面。最后将嵌入式文档写入列表，利用tqdm模块将遍历列表写入数据的过程可视化并存储到MongoDB数据库中，至此，数据采集阶段结束，所有数据已经准备就绪。

3.4系统功能实现

3.4.1预约比赛战队

在预约比赛战队页面中，设置提交按钮的“lay-submit lay-filter”属性，值为formDemo。Script标签中有对应名的按钮处理程序form.on('submit(formDemo)',function(data)。点击提交按钮时，将表单中的信息传递给JavaScript处理，其中包含url、type、data、datatype属性，JavaScript程序又将表单提交的值传递至url路径中的程序处理，提交方式为 “POST”，数据为表单中传递的值。数据类型为“JSON”。

因此，对应至本项目，数据传递至RaceFlowController类中的addRaceFlow方法处理。RaceFlowController类中的addRaceFlow方法又调用对象eventService的addRaceFlow方法处理。随后将传递至EventService类对象eventService的参数赋值给形参RaceFlow类的对象raceFlow，并调用addRaceFlow方法，将订单信息写入MongoDB中。执行完毕后RaceFlowController类中的addRaceFlow方法返回“成功添加赛事流程，比赛即将开始---”。add-team页面收到返回信息之后，通过屏幕快闪的方式显示该信息。

预约比赛战队界面如图所示：

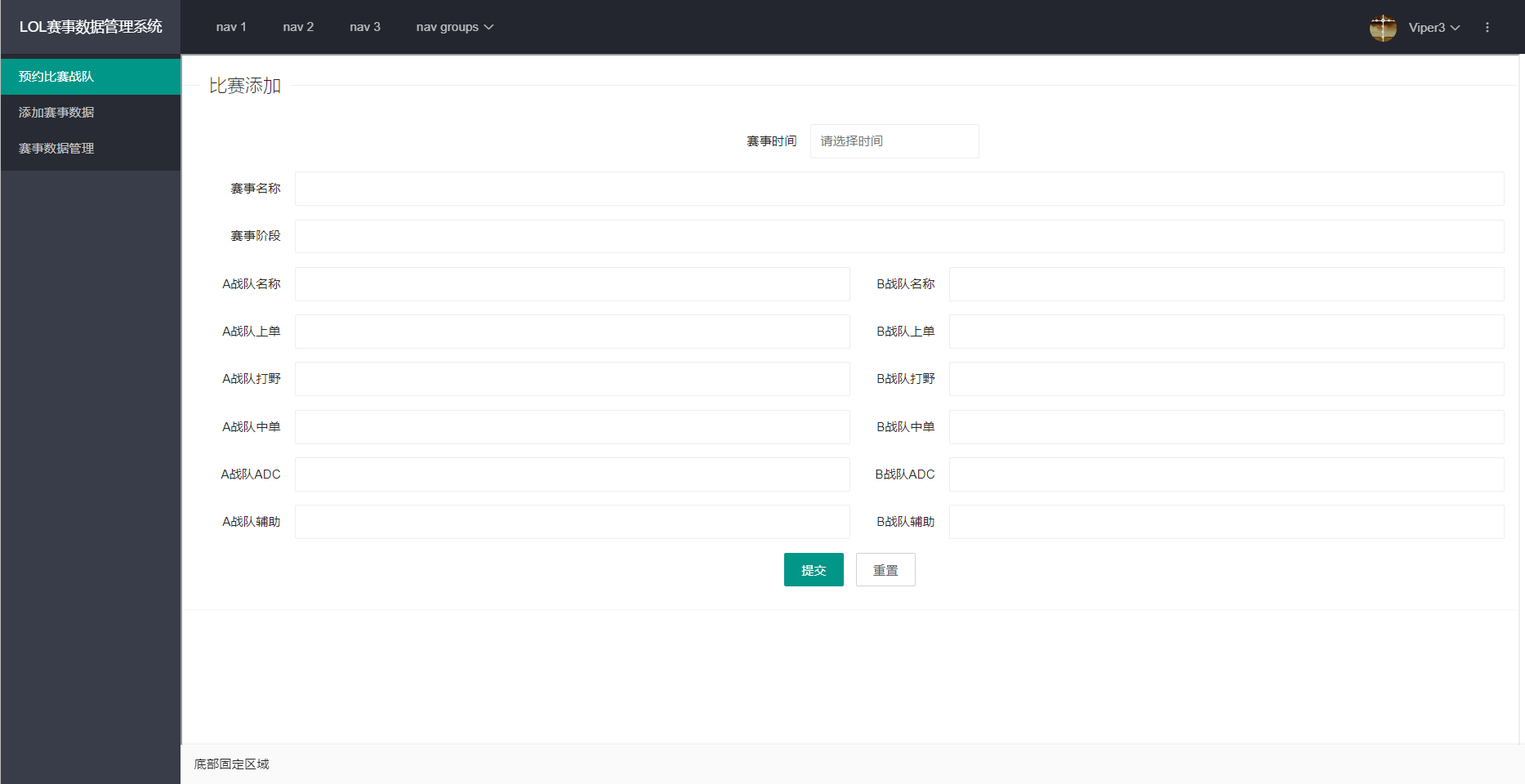


图 31 预约比赛战队

3.4.2添加赛事数据

在添加赛事数据页面中，设置提交按钮的“lay-submit lay-filter”属性，值为formDemo。Script标签中有对应名的按钮处理程序form.on('submit(formDemo)',function(data)。点击提交按钮时，将表单中的信息传递给JavaScript处理，其中包含url、type、data、datatype属性，JavaScript程序又将表单提交的值传递至url路径中的程序处理，提交方式为 “POST”，数据为表单中传递的值。数据类型为“JSON”。

因此，对应至本项目，数据传递至EventController类中的updateAndAddEvent方法处理。RaceFlowController类中的updateAndAddEvent方法又调用对象eventService的updateAndAddEvent方法处理。随后将传递至EventService类对象eventService的参数赋值给形参EventDouble类的对象eventDouble，随后对象eventDouble的参数传至形参Event类的对象eventA和eventB用以存储events列表中eventA和eventB字段，并调用updateAndAddEvent方法，将订单信息写入MongoDB中。执行完毕后RaceFlowController类中的updateAndAddEvent方法返回“赛事数据更新成功”。add-event页面收到返回信息之后，通过屏幕快闪的方式显示该信息。

添加赛事数据界面如图所示：

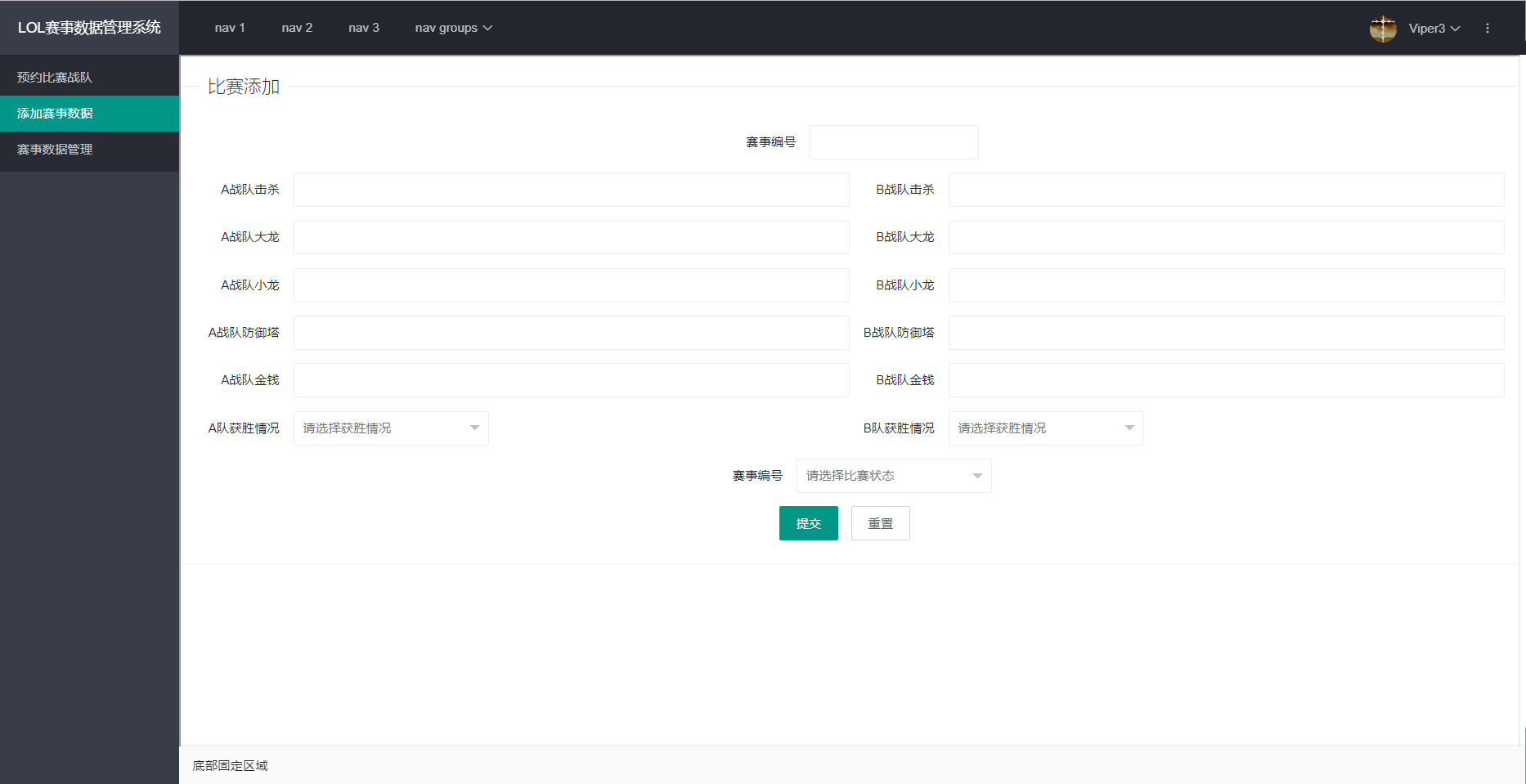


图 32 添加赛事数据

3.4.3赛事数据管理

在赛事数据管理页面中，由于存在两个战队，需要设置两个时间线，设置时间线中ul标签的id为events\_A和events\_B，数据表格id为raceflowList，lay-filter属性为raceflowTable，设置头工具栏事“筛选/导出/打印”的id属性为toolbarDemo，设置行工具栏事件的id属性为barDemo。

因此，对应至本项目，数据传递至RaceFlowController类中的deleteTeamById方法处理。RaceFlowController类中的deleteById方法又调用对象eventService的deleteTeamById方法处理。随后将传递至RaceFlowService类对象eventService的deleteTeamById的参数赋值给形参String类的对象id，并调用mongoTemplate的remove删除查询到的赛事信息。执行完毕后RaceFlowController类中的deleteTeamById方法返回“删除成功”。events-manage页面收到返回信息之后，通过屏幕快闪的方式显示该信息。

通过输入id将数据传递至RaceFlowController类中的selectEventById方法处理。RaceFlowController类中的selectEventById方法又调用对象eventService的selectEventById方法处理。随后将传递至RaceFlowService类对象eventService的selectEventById的参数赋值给形参String类的对象id，执行完毕后RaceFlowController类中的selectEventById方法调用mongoTemplate的findOne返回查询到的订单信息。events-manage页面收到返回信息之后，将数据呈现在面板中。

数据传递至RaceFlowController类中的selectTeamList方法处理。RaceFlowController类中的selectTeamList方法又调用对象eventService的selectTeamList方法处理。随后将传递至RaceFlowService类对象eventService的selectTeamList的参数赋值给Map方法，执行完毕后RaceFlowController类中的selectTeamList方法调用mongoTemplate的findAll查询订单信息,通过for循环将数据存到list列表中。events-manage页面收到返回信息之后，将数据呈现在数据表格中。

赛事数据管理界面如图所示：

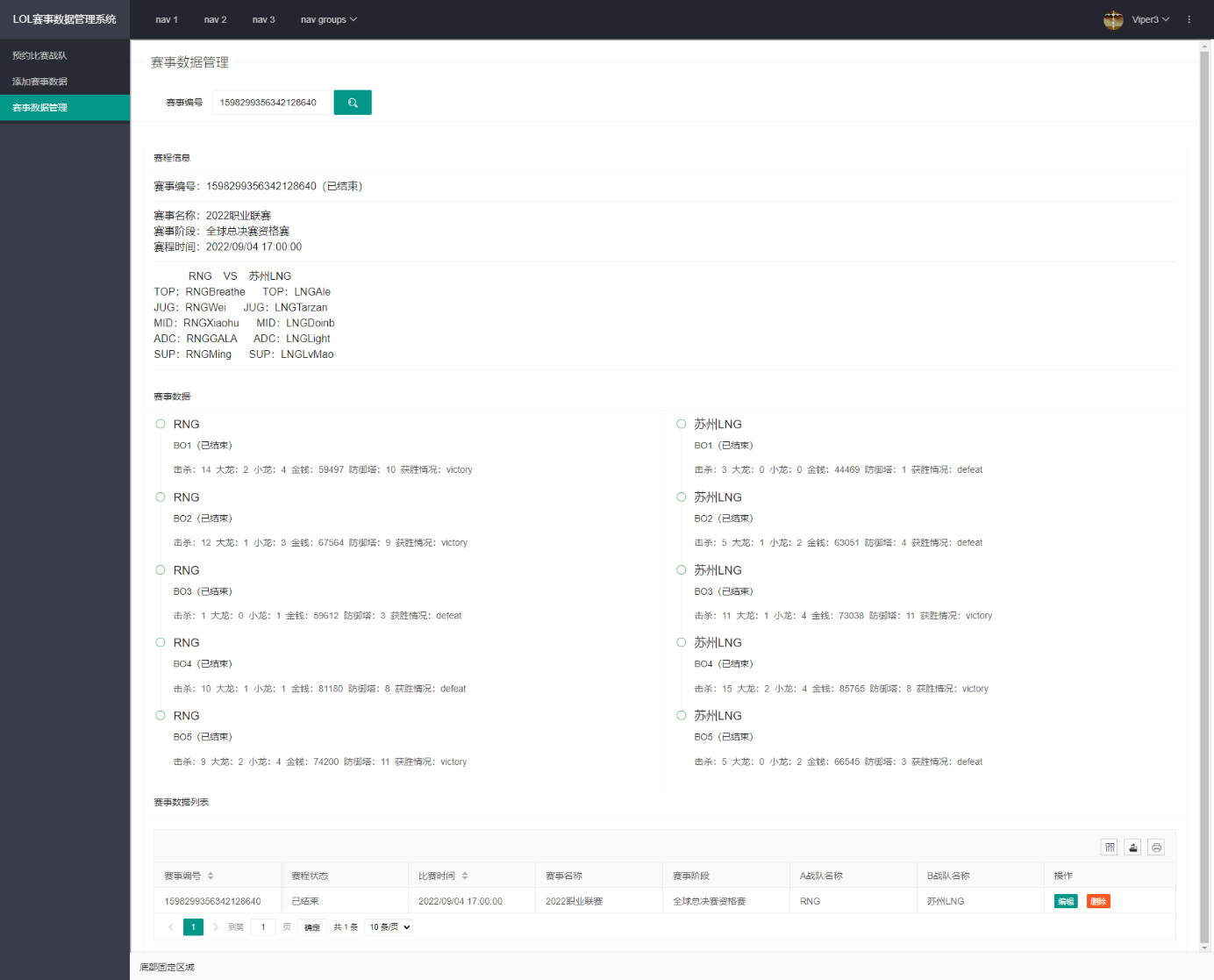


图 33 赛事数据管理

**4总结**

在现代硬件、软件的飞速发展，人民对文化产业的消费比重日益增加的情况下，电竞赛事规模也越来越大，其中也伴随着博彩行业的介入。线上赛的公平、公正难以得到保障，而规模越来越大的情况下，管理事项越来越多，战队、选手、场地、时间、比赛器材、工作人员的统筹工作需要一个管理系统去协助赛事管理者，提高管理者管理的效率，同时保障电子竞技比赛的公平性。

本项目的设计与实现，主要是利用Java与HTML语言来完成的，能够模拟赛事从预约到结束的全过程，对于爬取到的数据可以保存在MongoDB数据库中，基本上体现了信息管理系统各方面的优点。但随着电子竞技的赛事的扩大，管理者会有更多事务处理，更多信息需要调配，所以本系统还有许多待开发功能，例如选手转会期间的自由交易市场、与游戏数据的互通、自动化规划赛程等等功能。

通过本次实践的学习，使得我们对HTML以及Java的编程和应用有了更加深刻的了解，并且在此过程中多次操作MongoDB数据库以及对相关依赖包的使用，让我们更加熟悉关系型数据库和相关技术的使用，也通过解决问题的方式，了解了更多此前仅从书面上不曾学到的东西，让我们不仅体会到了技术发展的日新月异，也感受到了技术与技术间的碰撞以及相互促进带来的创新与发展。

第一阶段：需求分析阶段。只有充分了解了用户的需求才能开发功能完整、性能良好的项目。在这个阶段，我们小组听取了每个同学对于这个网站各个功能模块的描述，并做详细的记录，同时通过英雄联盟官网的数据进行结构设计。这项工作为我们后面项目的度量提供了可靠的材料。

第二阶段：项目分析设计阶段。整个项目在这个阶段的工作要多一点，它直接关系到后一阶段的编码，所以它起到了承上启下的作用。这一阶段的主要任务包括分析项目中对象，再根据对象设计数据库，在此包括其建模设计，在完成数据库后就是数据流程图了，它大体上描述了程序走的流程，以及大体的一个架构。完成上述工作后就是类的设计了，它是根据数据流图的设计来设计的，写好每个模块的每一个类，为下一阶段做好准备。在此，我们就完成了整个系统的一个架构。

第三阶段：编码阶段。用代码将整个系统的业务逻辑表达出来。其中和遇到好多问题：对java中的好多现有的类不熟悉，使得编写的代码质量不高，代码的复用性不高，好多问题还都没有解决。

第四阶段：测试。主要是对项目所涉及的功能进行功能测试。发现问题及时解决。

通过对这门课的学习，我们对这门课程有了更深层次得到了解，也有了更浓厚的兴趣。尤其是对网页制作的过程与一些技巧手法更有了另外一番了解，对网页制作的基础知识也有了一定的掌握。

1. 确定网页设计的内容：一个优秀的网站要有一个明确的主题，整个网站围绕这个主题,也就是你在网页设计之前要明确你这个网站有什么目的，用来做什么，所有页面都是围绕着这个内容来制作，有了明确的内容对排名有很重要的作用。
2. 了解你网站相关的客户：用户是一个网站成败的关键,如果用户要花很多时间进入你网站很有可能用户会立即关掉你网站，或者你网站操作很不方便用户也会马上离开，这种网站是很失败的设计，只会让用户失望的离去。
3. 做一个典型用户：一定要以用户身份亲身体验网站你才能发现问题，才知道那些需要改进，尽量不要让客户去发现问题，在前期设计时就应该把这些问题解决。

在具体的规划一个网站时，可以用树状结构先把每个页面的内容大纲列出来，尤其要制作一个大的网站(有很多页面)的时候，特别需要把这个架构规划好，也要考虑到以后可能的扩充性，免得做好以后又要一改再改整个网站的架构，很浪费时间和财力。大纲列出来后，还必须考虑每个页面之间的链接关系。是星形、树形、或是网形链接。这也是判别一个网站优劣的重要标志，因为链接混乱，层次不清的站点会造成浏览困难，影响内容的发挥。在框架定下来了后，然后开始一步一步有条理，有次序地做来，就胸有成竹得多，也会为网站将来发展打下良好的基础。

通过一步一步制作最终把自己的网页上传到互联网上，那种成就感是无法用语言形容的。在今后的闲余时间，我们将会继续学习制作网页的更多知识，使它成为自己的一项技能。

我们组每位成员都精心付出了自己的努力，相互依赖，齐心协力地进行工作，已保证项目目标的成功实施.同时我们组也做到了以下的关键几点：

1. 对项目目标的清晰理解
2. 对每位成员角色和职责的明确期望
3. 目标导向
4. 高度的合作互助
5. 高度信任

这些都是以后我们在做项目设计时候必须借鉴的。一个绩效良好的项目团队很有必要管理好时间，为有效管理时间，团队成员要明确每周的目标，每天制定一个做事表，集中精力完成当天的做事表。要控制干扰，谢绝参加那些对实现目标没有意义的活动。团队成员也要有效利用等待的时间，一次性处理好文件工作，并要为实现目标奖励自己。我们组的每位成员都尽心尽力地为这个项目付出，期待项目最后成功的实施。