



**本科生毕业设计（论文）**

|  |  |
| --- | --- |
| **设计（论文）题目** | 基于Spring Cloud的乒乓球论坛项目 |

|  |  |
| --- | --- |
| **学院：** | 计算机学院 |
| **专业：** | 软件工程 |
| **年级班别：** | 20级2班 |
| **学生姓名：** | 刘龙 |
| **学生学号：** | 3119009062 |
| **指导教师：** | 刘鹏 |

# 摘要

乒乓球是我国传统优势项目，乒乓球也被誉为我们国家的国球。随着国家“全民健身”战略的推进，乒乓球以其容易上手，对场地没有过高要求，玩法多种多样，老少皆宜的特点逐渐进入人们的视野当中，越来越多的人投入到乒乓球运动中。

随着乒乓球运动和互联网的进一步发展，出现了各种各样的乒乓球交流的论坛网站。诸如“虎扑体育”，“乒乓网”。前者并不是专业的乒乓球论坛网站，因此和乒乓球相关资讯比较少；后者虽然是乒乓球专业论坛网站，但是网站页面老旧，使用体验及其不佳且对乒乓球入门者学习乒乓球技术及其不方便。

基于上述的客观情况，本课题旨在基于Spring Cloud技术搭建乒乓球交流、讨论论坛项目。在这个论坛当中，用户可以获取最新的乒乓球资讯，同时可以自由发表自己的观点，讨论相关的话题；同时提供视频教程和图文教程帮助新手更快的完成乒乓球入门，同时相比于传统的乒乓球网站项目，提供“线下比赛”的模块，顾名思义，就是在线下组织相关乒乓球比赛，而本网站提供比赛发布，选手报名，积分排名等基本服务。拓展传统论坛网站的功能。

技术层面，使用目前主流的开发技术。前端使用最新的Vue3技术；采用微服务的架构方式，具体使用Spring Cloud实现微服务架构。同时为了提高用户的使用体验，在磁盘数据库Mysql的基础上加入内存数据库Redis，加快指定信息的检索速度。

项目部署层面，依托“腾讯云”平台，项目开发，测试完毕后，将部署到云服务器上，模拟真实项目开发的环境。

同时考虑到目前大多数乒乓球视频和教学视频都有版权，普通用户是不能随便在互联网上传播相关照片和视频，特别是播放视频。就算是“乒乓网”，也只能在里面放置视频的链接。因此为了避免版权的一系列问题，涉及到乒乓球教学的图片和视频都是我本人录制。

总结，相比于传统的乒乓球论坛项目，如“乒乓网”，“虎扑体育”，在业务层面提供乒乓球入门教学功能以及线下比赛组织功能，增强用户的使用体验。在技术选型方面，选用目前主流的最新最稳定开发技术。在项目部署层面，使用腾讯云作为服务器，从而实现公网访问。

关键词：论坛网站，SpringCloud，Vue

# Abstract

Table tennis, known as our national ball, is the traditional advantage of our country in competitive sports. With the promotion of the national "national fitness" strategy, table tennis has gradually entered people's vision with its easiness to use, having a variety of gameplay, being suitable for all ages and without demanding high requirements for the venue. Therefore, more and more people are invested in table tennis.

A variety of table tennis communication forum websites have appeared. Such as "Hupu sports", "ping-pong net" and so on. On the one hand, The former is not a professional table tennis forum website, and there is less information related to table tennis; On the other hand, although the latter is a table tennis professional forum website, the website page is old, the use experience is very poor, and it is inconvenient for table tennis beginners to learn table tennis technology

Based on the above objective situation, this topic aims to build a table tennis exchange and discussion forum project based on Spring Cloud technology. In this forum, users can get the latest table tennis information, at the same time can freely express their own views, discuss related topics; At the same time, it provides video tutorials and graphic tutorials to help beginners to complete the entry of table tennis faster. At the same time, compared with traditional table tennis website projects, it provides the module of "offline competition". As the name suggests, it is to organize related table tennis competitions offline. Expand the function of traditional forum website.

At the technical level, the current mainstream development technology is used. The front end uses the latest Vue3 technology; The microservice architecture is adopted, and Spring Cloud is used to realize the microservice architecture. At the same time, in order to improve the user experience, on the basis of disk database Mysql, memory database Redis is added to speed up the retrieval of specified information.

At the project deployment level, relying on the "Tencent Cloud" platform, the project development, after the test is completed, will be deployed to the cloud server to simulate the real project development environment.

At the same time, considering that most of the table tennis videos and teaching videos are copyrighted, ordinary users are not allowed to spread relevant photos and videos on the Internet, especially playing videos. Even if it is a "ping-pong network", it can only place video links in it. Therefore, in order to avoid a series of copyright problems, the pictures and videos related to table tennis teaching are recorded by myself.

In summary, compared with the traditional table tennis forum projects, such as "Ping-Pong network" and "Hupu Sports", the introduction of table tennis teaching function and offline competition organization function are provided at the business level to enhance the user experience. In terms of technology selection, the latest and most stable development technology of the mainstream is selected. At the project deployment level, Tencent Cloud is used as a server to achieve public network access.

Keywords: Forum website, SpringCloud, Vue

目录

[摘要 2](#_Toc160046636)

[Abstract 2](#_Toc160046637)

[1 绪论 1](#_Toc160046638)

[1.1 课题背景及意义 1](#_Toc160046639)

[1.2 课题研究方案 2](#_Toc160046640)

[1.3 课题主要内容 2](#_Toc160046641)

[1.4 简介 3](#_Toc160046642)

[2 可行性分析 4](#_Toc160046643)

[2.1 个人编码经历 4](#_Toc160046644)

[2.2 模块分解结构图 4](#_Toc160046645)

[2.3 模块技术可行性分析 6](#_Toc160046646)

[3 需求分析 8](#_Toc160046647)

[3.1 功能需求及用例图 8](#_Toc160046648)

[3.2 普通用户用例描述 9](#_Toc160046649)

[3.2.1 账号管理用例 9](#_Toc160046650)

[3.2.2 查看文章用例 9](#_Toc160046651)

[3.2.3 文章推荐用例 11](#_Toc160046652)

[3.2.4 用户举报用例 12](#_Toc160046653)

[3.2.5 核心用例——内容发布 13](#_Toc160046654)

[3.2.6 内容管理用例 16](#_Toc160046655)

[3.3 管理员端用例 16](#_Toc160046656)

[3.3.1 搜索推荐用例 16](#_Toc160046657)

[3.3.2 智能推荐用例 17](#_Toc160046658)

[3.3.3 内容审核用例 18](#_Toc160046659)

[3.3.4 举报审核用例 18](#_Toc160046660)

[4 需求建模 19](#_Toc160046661)

[4.1 静态模型 19](#_Toc160046662)

[4.1.1 实体类图 19](#_Toc160046663)

[5 概要设计 21](#_Toc160046664)

[5.1 微服务设计 21](#_Toc160046665)

[5.1.1 服务概述 21](#_Toc160046666)

[5.1.2 服务之间交互 21](#_Toc160046667)

[5.2 数据库设计 22](#_Toc160046668)

# 绪论

## 课题背景及意义

运动项目的开展,竞技体育和群众体育缺一不可,需要协调发展。体育的本质功能和根本任务：发展体育运动,增强人民体质。在我国乒乓球作为“国球”,在世界乒乓球竞技领域有着很好的成绩的同时凸显出普及性走低的反差。【1】随着国家“全民健身”战略的推进，乒乓球以其容易上手，对场地没有过高要求，玩法多种多样，老少皆宜的特点逐渐进入人们的视野当中，越来越多的人投入到乒乓球运动中。但是和“大球”运动相比，尤其是篮球，乒乓球的推广还有如下不足之处：首先，很多人都会打乒乓球，对乒乓球的熟练程度远远比不上对篮球的熟练程度。因此很多人在打乒乓球的时候，不是在捡球就是在捡球的路上，主要的运动量都花在捡球上面了。用专业的话语来说就是“不能形成多拍”。其次，乒乓球的影响力远不及篮球，足球。打开专业体育网站，如虎扑，腾讯体育，上面讨论最多的话题和篮球有关；对于乒乓球的讨论少之又少，一方面乒乓球技术细腻，打法多样，变化多，大部分的人不了解这些技术，也不愿意去了解，自然无法讨论；另一方面，乒乓球不像篮球运动那样对抗激烈，比赛过程跌宕起伏，在外行人看来乒乓球就是“两个人打来打去，每一回合赢的人吼一声，比赛过程很无聊”。最后不容忽视因素是篮球商业化程度远远超过乒乓球，围绕篮球的流量，话题层出不穷。

鉴于以上乒乓球运动推广中不足之处的分析，要进一步的推广乒乓球，扩大在人群中的影响力，从技术的角度来看，应该解决两个问题：第一，让更多的人了解乒乓球，了解乒乓球运动，了解乒乓球常见的打法，技术，掌握基本的技巧，让“外行人”真的会打，不至于一打乒乓球大部分时间都花在了捡球上面。第二，提供一个平台给人们讨论乒乓球。关于这点，实际上已经有了一个平台——乒乓网。上面有各种各样的乒乓球技术知识，数不胜数的乒乓球比赛视频，但是使用体验对于新手很不友好。互联网技术已经发展很多年了，可以使用新的技术提高用户的使用体验。第三，提升人们的参与度。正如任务书中提到的，因为对场地要求较小，规则简单，使得定期在线下举办小规模得乒乓球比赛成为可能，而对于比赛的组织和管理无疑需要使用互联网技术。

## 课题研究方案

首先通过在知网查看文献，弄清楚我国乒乓球推广过程中所遇到的困境。参考文献【1】是在“全民健身”的大背景下探讨社区，校园如何抓住机遇进一步开展体育运动，扩大体育运动的影响力，指出应当让体育运动走出专业的比赛场馆，走进人们的生活。参考文献【2】将目标群体集中于在校大学生，重点讨论了在篮球，足球日益风靡的情况下，高校开展乒乓球运动的困境以及对应的解决措施。重点指出大学生对乒乓球的固有映像：运动量少，比赛过程单调是导致很多学生觉得乒乓球运动无聊的原因。

然后通过调查，上网浏览当下流行的乒乓球交流网站，像虎扑，腾讯体育以及乒乓网。特别是乒乓网，因为它是一个专门为乒乓球爱好者搭建的网站，但是其网络界面过于老旧，排版很不合理，对新手很不友好，导致很多新手望而却步。但不可否认的是上面提供的技术知识分享非常专业，因此是本次课题参考的主要网站。

在对背景和时下热门乒乓球论坛网站有了一定了解基础上，决定在乒乓网的基础上进行改造，对上面的内容进行整理，重点照顾新手的使用体验，让新手更快入门。同时考虑到线下赛事组织过程很繁琐，在传统论坛网站基础上提供赛事组织服务。

## 课题主要内容

结合上面课题背景的分析，为了让新手更快完成乒乓球入门同时掌握一定的乒乓球基本功，真正感受到乒乓球的魅力；同时在此基础上参与到乒乓球话题讨论中，提高乒乓球的影响力；最后为有需要的人群提供举办线下比赛的赛事报名，选手分组等服务。本课题旨在利用互联网技术在乒乓网的基础上实现一个新的，重点照顾新手使用体验的乒乓球论坛网站。概括来说，包含如下的内容：

1. 循序渐进，由浅入深地提供完善的乒乓球知识，入门技巧以及基本的打法技巧。让用户对乒乓球有一个基本的了解，能够真正参与到乒乓球运动中，参与到和乒乓球相关话题的讨论。同时提供关于乒乓球的趣味知识，娱乐资讯，让用户从多个角度认识乒乓球。
2. 提供基本的乒乓球论坛服务。用户可以在论坛上面发表和乒乓球相关的内容，同时参与到和乒乓球相关话题的讨论中。系统将定期梳理最新的热点内容，创建标签供大家讨论。
3. 提供线下赛事组织服务。包含：选手报名，选手分组，选手排名。用户可以在论坛上查看附近有无线下比赛，通过平台提供的服务完成报名参与到比赛当中。不同于传统的业余赛事，这里提供的线下赛事规模更小，可能是大学某个学院，某个小区。

## 简介

乒乓球论坛项目一方面在完成传统论坛项目共有的操作的基础上，新增乒乓球入门教程以及线下比赛两个功能，旨在将线上的信息交流和线下的比赛结合起来，让网站的功能更加丰富，使网站的实用性更强。概括而言，需要实现如下的几个功能：

1. 账号管理

账号管理是论坛项目必须完成的功能。用户通过网站可以完成账号的注册，登录以及个人信息修改等基本工作，这是用户进行后续操作的前提。

1. 信息发布

用户可以在网站中发表相关信息，为了方便对信息进行管理，将用户发表的信息分为三大类。第一类是和乒乓球相关的资讯信息，这些信息大多来自第三方媒体报道，娱乐性较强，用于满足用户的娱乐需求；第二类是教程信息，这类信息专业性比较强，内容囊括乒乓球入门，进阶以及提升技巧，受众多为对对乒乓球技巧感兴趣的人群，适合乒乓球新手以及有一定乒乓球技巧基础的人。第三类是赛事信息，这类信息通常由具有一定资历的球友发布，包含比赛的场所，时间以及对参赛选手要求的基本信息等。是本项目在业务层面的一个创新，因为诸如“虎扑体育”以及“乒乓网”都只将关注点放在了线上的交流，没有提供线下的比赛组织服务。

1. 评论发布

相较于信息发布，用户发表评论的时候是一种“小规模”的交流。除了发表基本的文字评论以外，还允许用户发表图片评论，同时还包含其他诸如点赞，收藏拉黑和评论类似的操作。

1. 权限管理

管理员特有的操作，审核用户发表的内容，这些内容包含资讯信息，教程信息以及赛事信息。只有通过管理员审核的信息才会在平台上得到展示。同时还需要完成对用户举报的内容的审核，在审核完毕之后，对用户的账号状态进行管理。

（5）线下比赛

该功能是新增的一个功能，用户可以在网站中查看发布的赛事信息，完成比赛的报名。对于比赛发布者，可以实时看到当前参赛选手的报名信息，对他们进行审核决定是否让其参加比赛，在比赛结束后根据选手的排名对选手的积分进行管理。积分是用来衡量一个用户乒乓球水平的，积分越高，表明该用户参加的赛事就越多且排名越高。

# 可行性分析

本项目需要实现的功能比较多，在进行需求分析之前应当先进行模块分解，再对Spring cloud技术进行简单介绍。最后针对模块进行技术层面的可行性分析，为后面的需求分析奠定基础。

## 模块分解结构图

在对模块进行分解的时候，主要是从业务的角度的进行分解的。通过课题简介中的描述可知至少应当分为两个大类模块，一是普通用户模块；二是管理员模块，管理员模块涉及的操作比较简单，主要的内容是账号管理，审核用户发表的内容，举报管理以及变更用户状态。简单而言就是对用户发表的内容和举报的对象进行审核。回到普通用户模块需要进一步细分，首先可以独立出用户管理模块，对用户账号信息进行管理。其次用户需要发表内容，内容包含三大类，为此抽象出三个模块，分别是资讯发布，赛事发布以及教程发布。最后本项目新增线下比赛的服务，其中线下比赛可以单独作为一个模块，因此整个项目的模块分解图如图 2.1所示。

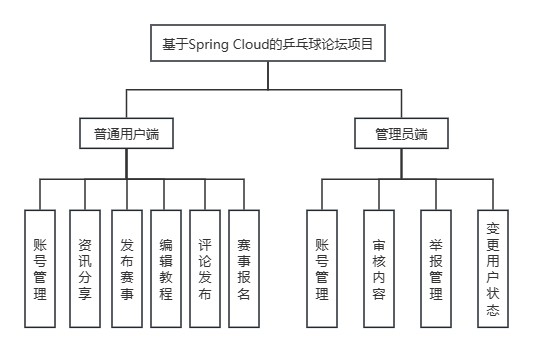


图 2.1 模块分解图

## Spring Cloud技术简介

传统项目大多是单体架构，所有的业务代码都集中在一个大的代码库当中。各个业务之间往往耦合比较严重。一种典型的情况就是在修改某个业务中的某个函数往往会引起其他业务的功能，这就导致项目的可拓展性大大降低。因此只适用于比较小的系统。当项目比较大，功能复杂的时候，此时应当考虑分布式架构。分布式架构相较于单体架构更加的灵活，将业务代码按照业务进行拆分，分成不同的模块。最后针对每一个模块，提供一种实现方式。这种架构带来的问题就是模块之间的交互，为此需要一定的消息通信机制。而微服务架构是分布式架构中已经得到实践检验的一种分布式架构解决方案。其核心在于服务（service），首先根据业务划分出不同的服务，各个服务彼此独立，耦合度低。单个的服务就可以完成一种具体业务，多个服务联系在一起，就可以完成复杂的业务流程。

Spring cloud微服务框架提供了微服务架构的一种实现方式，通过其提供的一系列组件库可以迅速完成微服务应用的设计和开发流程。其提供的核心组件如下所示[]：

（1）服务的注册和发现

Spring Cloud提供的注册中心是Eureka。所谓服务的注册指的是新增加的服务需要能够被其他服务感知，为此需要对它们进行管理。而服务发现指的是各个用服务之间需要有一套交流机制。Eureka提供注册中心组件Eureka Server用来注册其他的微服务，同时提供Eureka Client组件来创建一个微服务。

（2）服务的调用和负载均衡

通常使用feign来实现远程服务调用，其实现基于HTTP协议，并且支持Restful特性，使得服务之间的调用可以跨语言。也就是说一个微服务可以使用Python实现，另一个微服务使用Java实现，后者使用Feign提供的一系列方法可以访问前者；Ribbon的作用是负载均衡，在接收到用户的请求，会按照一定的算法例如轮询算法，优先级优先算法将用户的请求均摊到一个微服务对应的多个实例当中。客户端可以通过组合RestTemplate和Ribbon实现负载均衡来访问微服务。

（3）服务配置中心管理

Spring Cloud Config 负责解决分布式系统的配置管理。它包含以下两部分：Server提供配置文件的存储、输出，而 Client 则透过端口获取数据并初始化本应用。它能够统一管理、动态更新和获取各微服务的配置文件，为所有的微服务应用统一提供外部的依赖配置。

（4） 服务熔断组件

Hystrix同样作为Spring Cloud的一个功能组件。为微服务架构提供了极高的容错能力。在某一个微服务由于各种各样的异常情况，如负载过高，硬件故障而不能正常工作的时候，熔断机制可以为微服务提供一层保护，让其不再进行后续的请求处理。

# 需求分析

需求分析阶段需要完成的工作有功能性需求分析以及非功能性需求分析。首先进行功能性需求分析，先对系统的边界和涉及到的角色进行划分，然后根据绪论中提及的项目简介以及在网上查找到的调研报告，分析出每一个角色包含的操作，最后给出每一个角色的用例图。非功能性需求主要集中在两个方面，首先是安全需求。其次是网站响应时间。

## 功能性需求分析

### 普通用户功能性需求

用户需要进入网站完成后续操作，首先需要一个账号，同时用户还应当对自己的账号进行管理。因此可以独立出管理账号用例，该用例下包含几个子用例，分别是账号注册，账号登录以及账号信息修改。然后在用户登录成功来到网站以后，用户可以发表相应的内容，这些内容包含三类，分别是资讯，教程以及赛事信息。为此发布信息用例下应当包含管理资讯，管理教程以及管理赛事子用例。对于每一个子用例，还应当包含这个子用例中涉及到的信息的增删改查操作，例如管理资讯用例下应当包含发布资讯，查看资讯，修改咨询以及删除资讯操作。对于线下比赛模块，还应当提供一个用例供选手报名以及比赛发布者查看参赛选手信息，为此新建一个线下比赛用例，包含两个子用例，分别是查看选手信息和报名参赛。最后用户可以在网站中发表自己的评论，同时还可以删除自己的评论。为此抽象出两个用例即发布评论和删除评论。完整的用例图如图 3.1所示：

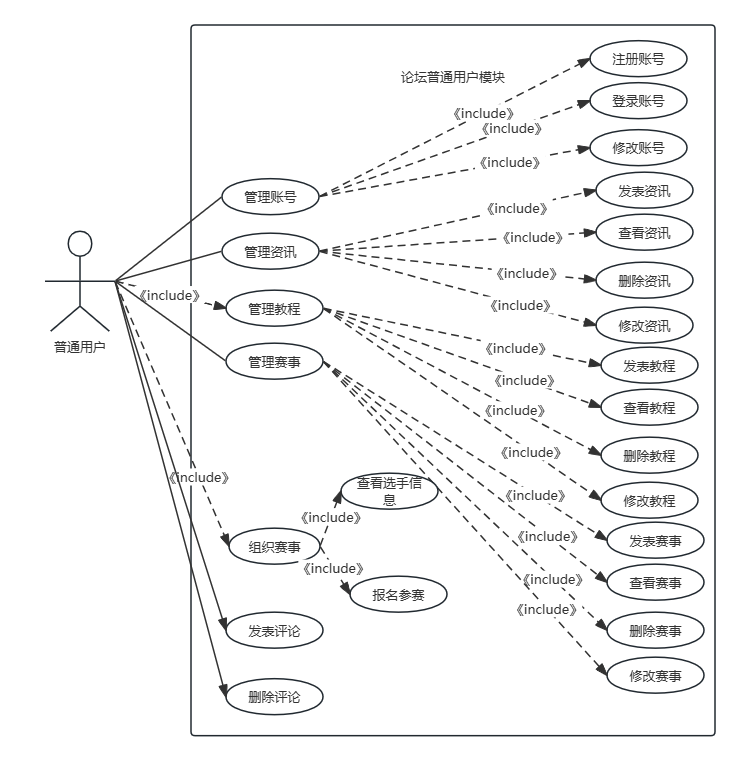
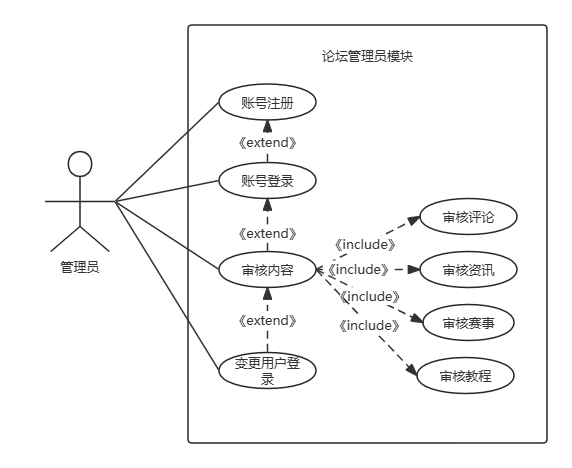


图 3.1 普通用户用例图

### 管理员功能性需求

管理员作为系统的超级用户，享有最高的权限。可以查看用户发表的内容，对相关内容进行审核。审核通过之后，相关信息才会在网站中得到展示。更为重要的是为了维护网站社区的稳定性，需要对违规用户账号状态进行处理。处理的常用操从重到轻有：封禁账号，禁言，禁言7天，警告。同时用户发表的内容有四大类：一是资讯、二是教程，三是赛事以及评论信息。这些内容在形式上有很大差异，需要不同的审核方案。为此独立出四个单独的用例，即审核评论，审核资讯，审核教程以及审核赛事。最后管理员需要根据审核的结果对用户的状态进行变更。完整的用例图如所示：



### 核心用例描述

账号登录用例用来完成用户账号登录，用户在登录前需要先输入账号，密码以及验证码信息。后台验证通过之后才会允许用户登录。当用户反复登录失败的时候，将对用户进行一段时间的禁止登录处理，只有等到规定时间过去之后才会允许用户登录。详细的描述如所示。

表 3.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID |  | 名称 | 账号登录 | 优先级 |  |
| 参与者及目标 | 用户来到登录界面进行登录 | | | | |
| 正常流程 | 1. 系统展示登录界面 2. 用户输入账号信息和密码信息 3. 系统展示图片验证码 4. 用户输入验证码信息 5. 系统进行身份验证，验证成功提示用户登录成功，跳转到主界面 | | | | |
| 扩展流程 | 4.1 用户验证码输入错误  1. 系统显示“验证码错误”信息，提示用户重新输入  5.1 系统身份验证失败，提示用户重新输入账号信息  1. 用户重新输入身份信息  5.2 系统身份验证失败次数超过四次，提示用户在未来1分钟内不能登录 | | | | |
| 业务规划 | 1. 验证码的长度为4，包含字母和数字 2. 关于扩展流程5.2说明，当失败次数超过4次时，接下来用户每登录失败一次将暂停一分钟登录 3. 主界面包含如下信息 4. 主界面相关信息格式  * 时间信息除上次登录时间的展示格式为：yyyy-MM-dd HH:mm:ss，其余的都为yyyy-MM-dd * 点赞量，浏览量，粉丝数数字展示格式：数量小于9999直接展示对应数字；超过10000展示“1万+”；超过20000展示“2万+”依次类推 * 比赛规模：大（超过500人参加），中（超过200人参见），小（不足200人参加） * 比赛场地信息：精确到小区门牌号，不超过60字符 * 标题信息不超过40个字符，文章摘要信息不超过100字符 | | | | |

发布文章用例是项目核心用例之一，同另外两个用例（发布教程、发布赛事）共同协助用户完成内容发布的相关需求。值得注意的是为了更方便获取文章的简略信息，如文章标题以及封面图片，这些信息需要用户单独输入，同文章主体进行区分。详细的描述如所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID |  | 名称 | 发布资讯 | 优先级 |  |
| 前置流程 | 系统登录成功来到主界面，选择发布文章操作 | | | | |
| 参与者及目标 | 用户在发布文章界面，编辑文章，发布 | | | | |
| 正常流程 | 1. 用户输入文章标题，所属分类，文章主体内容 2. 系统每隔5分钟保存用户的输入 3. 用户编辑摘要信息和封面照片之后选择提交 4. 系统保存用户的文章 | | | | |
| 扩展流程 | 3.1 用户没有编辑摘要信息就选择提交  1. 系统抽取文章前100字符生产摘要信息 | | | | |
| 业务规划 | 1. 文章标题不超过40个字符，摘要信息不超过100个字符 2. 封面照片大小不超过24KB 3. 用户在文章中上传的照片大小不超过3MB | | | | |

在用户登录成功以后，允许用户发表评论。值得一提的是，按照最新互联网法制法规，用户在输入评论以后需要显示IP地址。同样在本项目中为了更加真是模拟现实情况，在评论基本属性中也加入IP地址选项。详细的用例描述如所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID |  | 名称 | 发布评论 | 优先级 |  |
| 前置流程 | 用户登录成功来到谋篇文章或者教程的评论界面 | | | | |
| 参与者及目标 | 用户完成对指定内容的评论 | | | | |
| 正常流程 | 1. 用户来到评论界面 2. 系统展示所有的评论信息，每一条评论信息包含如下属性：内容，发布时间，ip所属地址，点赞量，不喜欢量。(默认先展示第一层里面的评论) 3. 用户选择在文章底下评论 4. 系统展示评论框 5. 用户输入评论并点击提交 6. 系统更新当前文章界面下的评论 7. 用户选择回复某条评论 8. 系统展示该评论下的前6条评论以及输入框 9. 用户在输入框输入内容之后提交 10. 系统记录用户的行为 11. 用户举报某条评论，同时输入举报分类以及举报描述 12. 系统记录用户的举报并且提交给管理员审核 | | | | |
| 扩展流程 | 3.1 用户输入的评论字符超过300  1. 系统提示用户重新输入 | | | | |
| 业务规划 | （1）评论内容的字数不超过200字符 | | | | |

报名参赛用例是本次设计中的亮点之一。旨在拓展传统论坛的功能，需要用户输入基本的个人信息，而对这些个人信息需要特别注意信息的安全性。同时用户不能随意参加某一个比赛，只有在积分满足一定条件之后才能参赛。详细的描述如所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID |  | 名称 | 报名参赛 | 优先级 |  |
| 前置流程 | 用户登录成功根据系统发布的赛事信息选择是否报名参赛 | | | | |
| 参与者及目标 | 用户在赛事信息界面选择赛事信息，同时完成报名操作 | | | | |
| 正常流程 | 1. 用户来到赛事信息界面 2. 系统展示赛事信息简要内容，每一条赛事信息包含：发布者，发布时间，比赛时间，比赛场地，比赛描述，积分要求，积分奖励 3. 用户选择某条赛事信息报名 4. 系统展示完整的赛事描述信息供用户浏览 5. 用户选择报名参赛 6. 系统更新报名信息 | | | | |
| 扩展流程 | * 1. 用户选择报名参赛但是人数已满   （1）系统提示用户重新选择赛事 | | | | |
| 业务规划 |  | | | | |

审核文章用例是管理员所特有的用例，旨在对用户发表的文章进行审核。审核通过之后才允许对应的资讯在平台展示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID |  | 名称 | 审核资讯 | 优先级 |  |
| 前置流程 | 管理员登录成功来到审核界面 | | | | |
| 参与者及目标 | 管理员查看用户举报的文章，并对之进行审核。 | | | | |
| 正常流程 | 1. 管理员来到审核界面，选择审核文章 2. 系统展示需要审核的资讯信息，每一条信息包含如下内容：举报来源，举报时间，举报文章标题，举报作者信息，举报类型，举报描述 3. 管理员审核文章之后，认可用户的举报 4. 系统记录管理员的选择，同时修改文章的状态为不可见 | | | | |
| 后置流程 | 进入“变更用户状态”用例对用户账号进行处罚 | | | | |
| 扩展流程 | 3.1 管理员不认可用户的举报  1. 系统向举报来源提示举报失败信息 | | | | |
| 业务规划 | 1. 举报描述仅展示简要信息，不超过30个字符 2. 举报类型包含：和乒乓球话题不相关，内容质量底下，涉嫌违法犯罪 | | | | |

## 非功能性需求

在考察项目的非功能性需求时，主要从如下几个方面考虑。第一是安全，第二是时响应时间，第三是人机交互界面，第四是软件的可维护性，最后是财力需求。接下来进行说明。

1. 安全性需求

首先用户在注册账号的时候需要填写手机号和密码信息等个人信息，这些信息的最基本的要求是不能被非法用户非法获取，由于本次设计的局限性。仅考虑服务器端的安全需求，也就是在数据处理和存储时的安全需求。在数据传输过程中，使用HTTPS协议保证传输过程中数据不被非法监听和篡改。在服务器端的安全需求有应用程序安全以及数据库存储安全。

（2） 响应时间

系统的响应时间对用户体验至关重要，理论上响应时间越快，用户使用体验越好。但是实际上处于业务和开发技术的局限性，通常只能进行一些优化。依据 2/5/10s 原则，当系统响应时间低于 2s 时，用户感觉流畅。但是当响应时间超过5秒之后，用户就会感觉到网络较慢。在项目中需要向后台请求大量的长文本信息，如资讯和教程，对于这些信息的响应时间要求如下。在并发量为500的情况下，返回给用户资讯以及教程信息的响应时间不超过4秒。其他的信息，如文章的描述信息，登录验证信息在并发量1000的情况下响应时间不超过3秒。

（3） 人机交互界面

正如课题简介中提到的那样，本次设计一个重要的亮点就是要提供给对乒乓球感兴趣的人的易操作的人机交互界面。“乒乓网”涉及到大量的专业知识，但是网站使用体验极差，对乒乓球新手的使用体验极差。而判断人机交互界面的易操作性采用随机测试的方法。在有一定乒乓球基础的人群中随机选取15个人进行测试，在对乒乓球感兴趣的人中随机抽取30个人最后随机在普通人群中抽取60个人，收集他们的网站使用体验和评分，满分为100分。当最后收集到的数据中测试的平均分大于75分时，认为网站的人机交互较优。

（4）易维护性

本系统基于微服务架构开发，各个服务体量小，职责单一且彼此独立。因此适合后期的维护和功能拓展。在后续如果有需求变迁的情况，如果原有的微服务不能满足要求，可以通过新建微服务进行拓展。而新增业务代码对原来的业务代码没有影响。

（5） 财力需求

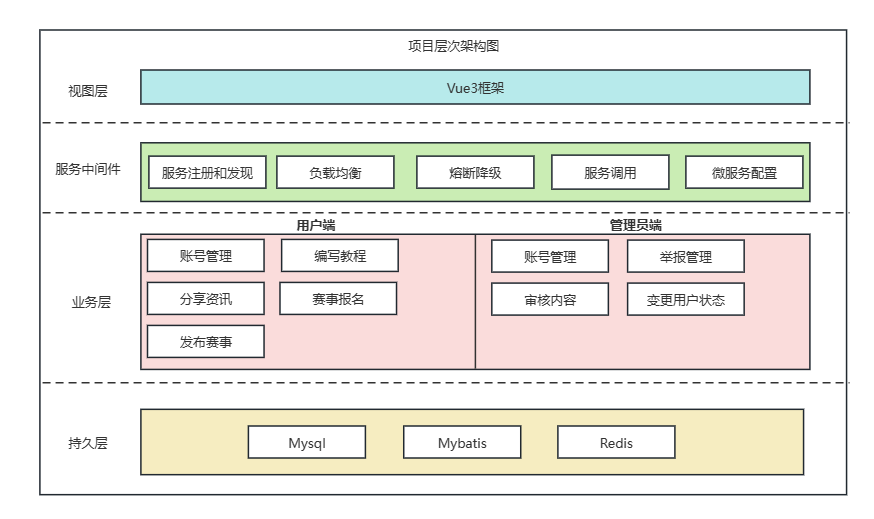
由于最后项目是需要部署到服务器上，含有一个公共IP地址和公共域名。云服务器的月租一般在25原左右，再加上公共域名的月租每个月大概是15元左右。项目部署的时间为3月5号，最终完成毕业答辩需要到5月中下旬左右。同时，还可能使用一些RDB和CDN服务，因此最终的预算不能超过150元。

# 概要设计

在概要涉及阶段需要完成的工作有确定整个项目的层次架构图。同时根据微服务的设计思想，在需求分析的基础上进一步划分出微服务，确定各个服务的功能，服务调用接口以及服务之间的交互。层次架构图的设计的主要依据是使用目前主流的，稳定的开发技术同时结合需求分析中的功能性需求给出层次架构图。微服务的划分中需要对划分微服务，同时给出各个微服务的接口以及服务之间的调用层次，最后给出系统的数据库设计。

## 层次架构图

首先是项目的层次架构，整个项目共分为4层，分别是前端视图层，服务中间件，业务层以及持久层。根据项目需要结合当前热点开发技术。在前端采用Vue3框架完成用户界面的开发。采用Axios发送异步请求，进行前后端的交互。后端采用Spring Cloud框架，使用到的组件有Eureka组件，用来完成服务的注册和发现。其中Eureka Server为服务注册中心，可以配置多个服务集群，从而避免某个单个注册中心负载过高；而Eureka Client用来在注册中心注册某个服务。其次还使用到了Ribbon实现负载均衡，因为项目中有些功能可能需要多次使用，例如获取资讯、教程以及赛事的简单信息。最后还使用到了Hystrix熔断组件，在服务出现异常情况时提供一层保护。因为在项目中用户编辑文本信息时，会插入一定数量的照片，虽然前端会对照片的大小进行限制，但是不可否认当用户上传的文章大小比较大时，可能造成服务不能正常工作。此时需要熔断组件进行服务降级。在业务层需要完成课题简介中涉及到的基本业务，后续微服务划分也是基于此。最后一层是持久层，采用目前主流的Mysql数据库作为持久化存储，同时为了方便操作数据库，加入常用的持久层框架例如Mybatis以及Mybatis Plus框架。最后为了加快查找的速度使用Redis作为缓存。层次架构图如所示



## 微服务设计

在进行微服务设计时，首先根据微服务划分的原则，即单一职责原则【补充】

对微服务进行划分，然后分析各个微服务的负载情况，对负载比较高的微服务采用负载均衡，同时对单次请求信息量大，交互的数据类型复杂的服务进行熔断处理。最后在此基础上确定各个服务的接口以及服务之间的交互，在完成这些工作之后，给出微服务架构图。

### 服务概述

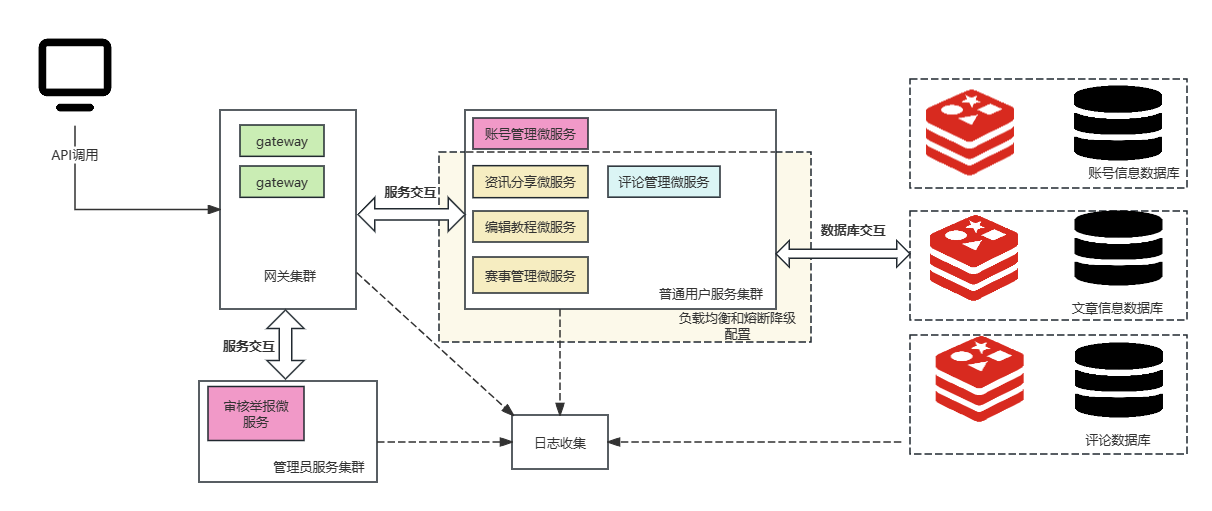
1. 账号管理微服务（UserAccountService）：对用户的账号进行处理，完成账号的注册，登录验证，状态改变等和账号处理有关的逻辑
2. 资讯分享微服务（NewsSharingService）：用户在登录成功以后，实现用户分享资讯，查看资讯以及和管理资讯相关的逻辑
3. 编辑教程微服务（AutorialRegulationService）：在用户登录成功以后，当用户的积分满足一定条件时，允许用户发表乒乓球教程，该微服务旨在完成用户教程的发表以及管理工作
4. 赛事管理微服务（MatchRegulationService）：用户登录成功以后可以发布乒乓球赛事信息，由于赛事信息的格式和内容和普通资讯以及教程有很大差别，且赛事管理涉及到多方的用户：赛事发布者，参赛选手以及管理员，因此单独划分出一个微服务
5. 评论管理微服务（CommentRegulationService）：传统论坛的评论总是和文章内容放在一起，但是在调查之后，发现大多数的网站的评论及其多，形式多种多样，如果还是放在和内容发布相关微服务里面，势必会造成耦合
6. 审核举报微服务（ReportService）：管理员才允许交互的微服务，审核用户发表的内容信息，只有在经过管理员审核通过以后，才允许内容在平台上得到展示

结合实际使用体验以及【补充】进行分析，首先是账号管理微服务，用户主要是在进入网站，以及修改个人信息时才会调用该服务，其余大多数时候并不用和该服务交互。因此该服务并不配置负载均衡。而对于微服务2，3，4是本次实现的重点需要配置多个微服务实例，其中由于资讯、教程以及赛事信息文件大小比较大，在处理的过程中可能会出现超时的情况，因此需要配置熔断机制，在出现超时等其他异常情况的时候进行处理。而对于评论管理微服务，请求量更大。结合我自己的使用体验，发现在校园网环境下，请求B站的评论数据的时间平均为3秒左右，偏慢。因此也需要配置多个微服务实例。最后分析审核举报微服务，只有管理员才允许调用，但是由于处理的信息比较多，还是应该配置多个服务实例。

接下来对服务接口进行划分，测试环境下所有的服务先部署在本机上，根据上面划分的微服务结合Spring Cloud自身组件的一写特点给出各个服务的接口。其中需要配置服务注册中心的地址以及网关地址。待项目完成测试之后在将项目部署在云服务器上。各个微服务的接口如所示

|  |  |  |
| --- | --- | --- |
| 服务名称 | 接口 | 服务实例数量 |
| 服务注册中心 | localhost:5347 | 2 |
| 网关服务 | localhost:80 | 2 |
| 账号管理微服务 | localhost:8001 | 1 |
| 资讯分享微服务 | localhost:8002 | 3 |
| 编辑教程微服务 | localhost:8003 | 3 |
| 赛事管理微服务 | localhost:8004 | 3 |
| 评论管理微服务 | localhost:8005 | 4 |
| 审核举报微服务 | localhost:56789 | 2 |

接下来分析服务之间的的调用方式。微服务的调用方式有两种，一种是接口远程服务调用（RPC）还有一种是事件驱动。虽然后者耦合度比较低，但是由于是采取消息发送的方式，采用消息队列缓存接收到的消息，并不能及时对用户请求进行答复。同时考虑到本次课程设计服务划分粒度并不是很细，即每个微服务本身完成的操作还是比较多的。当然更重要的原因是采用消息队列的方式会增加系统实现的复杂度，将耗费更多的精力。因此本次设计采用RPC远程调用的方式完成服务之间交互。



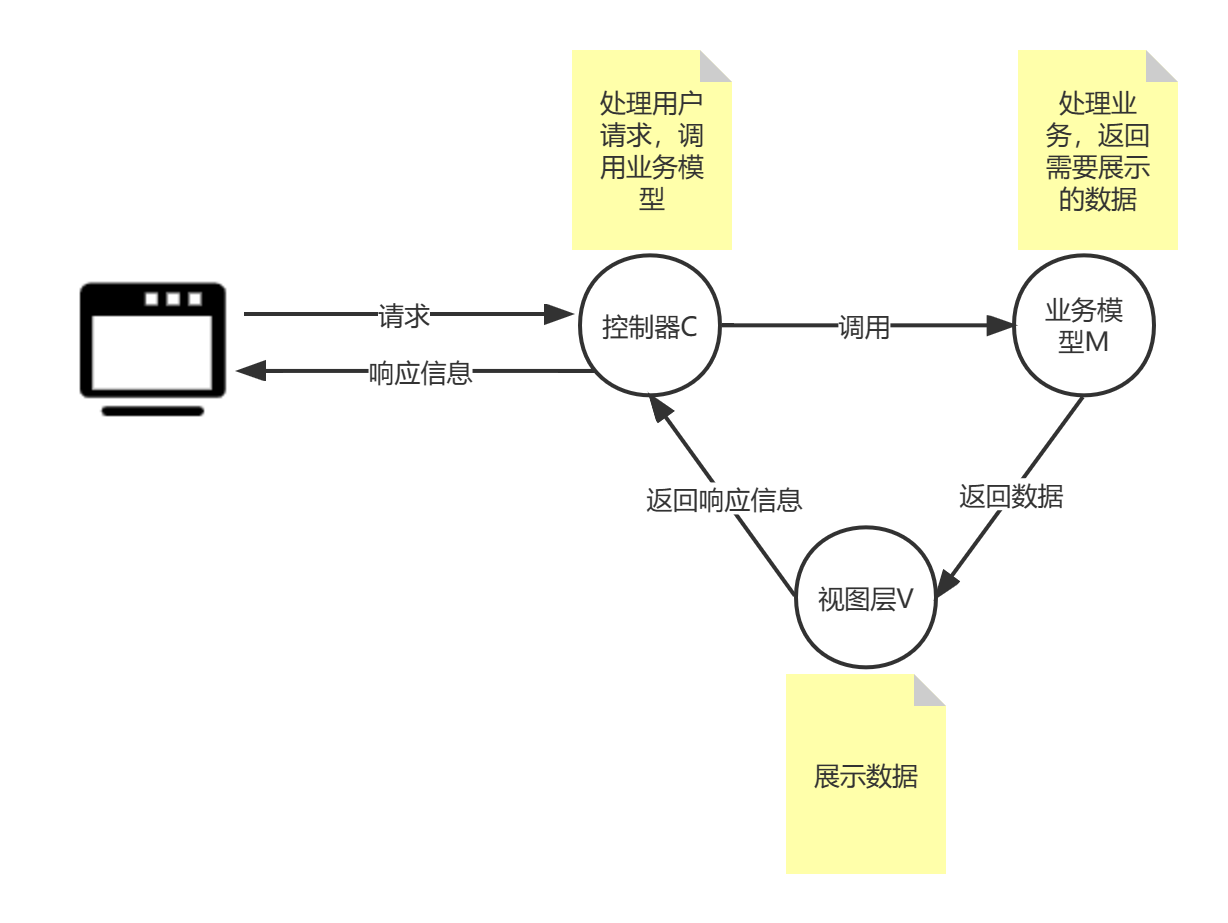
服务交互

### 微服务内部设计

微服务内部采用MVC设计方式。该架构方式将软件系统分为三个层次。分别是模型（Model）、视图（View）和控制器（Controller）。其中Controller负责接收来自视图的请求，同时根据请求参数调用合适的模型。模型负责对数据进行处理，并且将处理之后的结果返回给视图层。在MVC中，核心层次结构为模型层，其中模型层又可以进一步细分为业务层和持久层。其中持久层负责和数据库进行交互，业务层通过调用持久层提供的各种方法来完成具体的业务逻辑。

使用MVC有很多的优点，例如架构简单且易于拓展业务。但是当业务过于复杂时，业务层代码会急速膨胀。而配合微服务架构的方式，通过控制每一个微服务项目的规模避免业务层代码膨胀。

微服务架构方式如所示：



# 详细设计

在详细设计阶段，将根据服务类别完成各个服务的设计工作，服务设计的原则一是根据需求分析中的用例描述进一步细分用户需要完成的操作。同时还需要考虑到系统实现的复杂度，简化业务避免系统过于复杂。最后根据详细设计中各个微服务的设计，给出整个系统的数据库设计，同时给出数据库表的描述。

## 服务网关详细设计

本次设计使用Spring cloud gateway作为服务网关。起到对用户请求进行过滤，分发的作用。进行过滤的时候，主要是根据下面两条规则进行过滤。

（1）基于请求路径（path）过滤

前端发送过来的请求首先经过网关处理，网关分析请求路径，根据请求路径中特定的参数将请求派发给微服务进行处理。例如/user/login和/user/register均和/user/\*\*匹配，网关将上述请求转发给账号管理微服务。

（2）基于用户传来的参数（param）转发

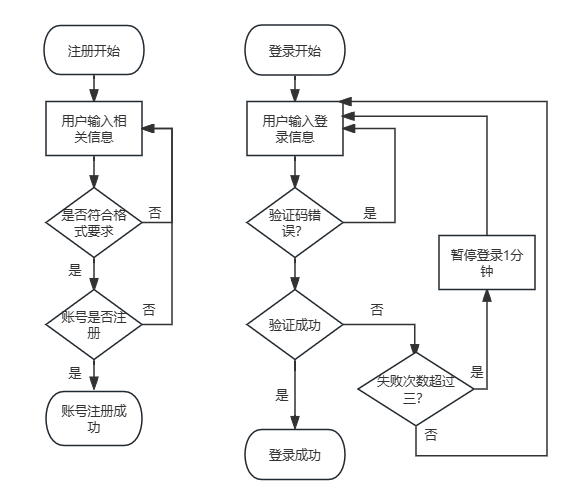
前端发送来的请求头部通常会包含一些参数，例如Cookie相关参数，请求源相关参数。网关同样可以基于这些参数将不同的请求派发给不同的微服务。

经过网关处理的请求的去向有三种情况。第一种是将请求转发至用户请求派发微服务，该微服务专门用来调用其他和用户相关的微服务。例如账号管理，评论管理，从而避免各个微服务自己相互调用，降低实现的复杂度。第二种情况是将请求直接转发给特定的微服务，这类请求通常请求的信息量比较大，如果在处理的时候经过用户请求派发微服务中介，将延迟系统响应时间。例如当用户请求查看资讯的详细的信息的时候。第三种是将相关请求派发给管理员进行处理。

采用上述的设计方式，在处理用户发杂请求的时候。能够合理的根据用户请求的复杂度和请求的数据量大小将请求派发给指定的微服务。

## 账号管理详细设计

用户需要使用该系统，首先需要注册一个账号，其次完成登录才能完成后续的操作。其中用户注册时需要输入如下的信息：账号昵称、手机号、邮箱、个人简介，球龄、用户头像（可选）以及感兴趣的话题。同时为了配合账号管理，还需要记录用户上次登录时间，用户的账号状态。账号管理就是围绕这些属性展开。用户注册账号时，只有在用户输入的格式满足要求且后台数据判断当前账号没有被注册的情况下才允许用户注册。同理适用于用户登录成功，修改自己的个人信息的时候。用户账号注册成功进行登录，登录需要用户输入图形验证码。只有在用户信息准确且允许用户登陆时，当前用户才能登录成功。当用户登录失败的次数超过三次的时候，将对用户进行一段时间的登录限制。详细的处理流程如所示



至于用户个人信息的修改，和用户注册账号类似。需要注意的是用户可以修改自己的头像信息。用户在注册账号的时候使用系统默认的头像，注册成功之后用户可以选择修改。同时用户不能修改手机和邮箱信息。

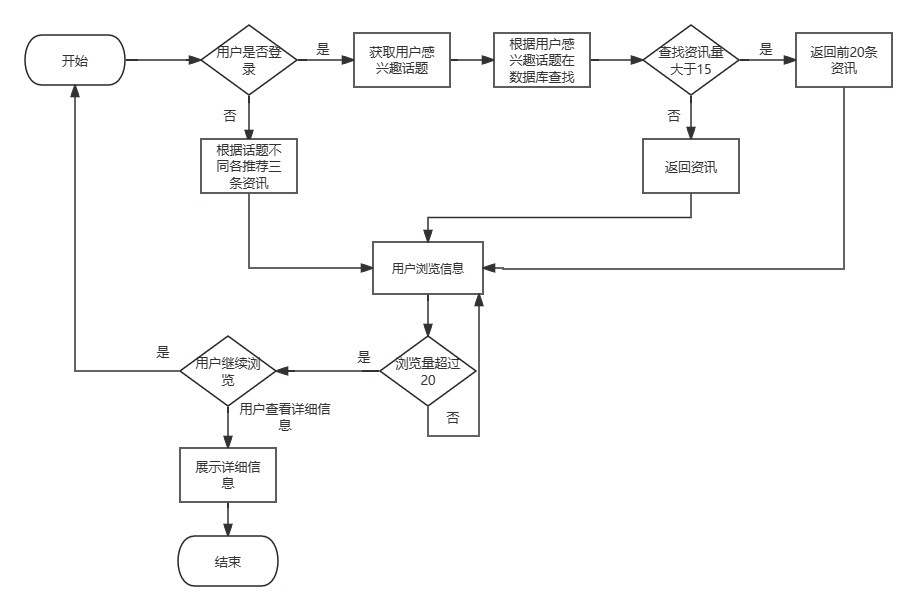
## 资讯分享详细设计

在资讯分享微服务中，需要完成的功能有：资讯信息的展示和编辑发布。在用户登录成功来到系统之后，首先展示的是各种资讯的简要信息，这些信息包含诸如封面图像，标题以及点赞量和收藏量等信息。用户可以查看具体的资讯信息，进入资讯详情页之后。用户可以在该文章底部进行评论。同时用户还可以选择发布资讯，在发布资讯的时候需要用户指定资讯的简要信息。

首先分析界面的详细设计。在展示资讯信息的时候采用列表的形式进行展示。首先展示前20条资讯信息，降低系统的吞吐量，当用户鼠标滑至窗口底部的时候，再加载新的资讯信息。在进行每一条资讯信息展示的时候，展示的样式参照“知乎”界面，同时还应当展示各条资讯的先赞量、拉踩量、收藏量以及分享量。对于发布资讯的界面，使用目前主流的Markdwn编辑器，用户在该编辑器上撰写文章的详细信息，同时为了便于对文章进行管理，还需要用户手动输入一些文章的描述信息。

首先需要解决的一个问题就是资讯信息的推荐。其实现的基本思路分为两步。首先记录用户感兴趣的话题，初始的时候采用“盲推”，随机选择不同类型的话题推荐给用户，记录用户浏览过的信息特别是点赞和收藏的资讯信息。后续再进行推荐时，优先推荐这些类型的信息。需要注意的是，用户可能是在未登录且用户本地浏览器没有存储用户相关信息的情况下访问系统主界面，此时无法根据用户身份推荐用户感兴趣的话题，此时采用默认推荐策略。即在各种类型的资讯中各选3条再返回给用户。

上述资讯推荐可以使用下面的流程图进行描述。



在用户查看资讯详情之后，系统将展示资讯的详细信息，详细信息的展示主要是一个回显的过程中。值得注意的是文章的主体信息通常内容很多；而文章的描述性信息如标题，发布时间，发布者内容比较少可以优先展示，基于此文章的描述信息和文章主体信息的展示是一个异步的过程。同时为了进一步提升用户的使用体验，在加载文章主体信息的时候，采用流加载的形式。也就是延迟加载实现文章内容的动态展示。

最后在资讯详情页的底部还需要展示评论信息，而评论信息的处理专门放到评论管理微服务进行处理，同理下文的编辑教程微服务中评论的处理也是放到评论管理微服务进行统一处理。

# 编码实现

## 服务网关实现

服务网关主要是通过application.yml文件进行配置，其中主要使用到两个参数一个是routes参数，用来定义转发路由。其下面有如下的子参数。id用来表示路由的名称；uri用来表示请求转发的目的微服务；predicates即断言，用来定义请求路径模式或者请求中的参数满足的条件。部分网关配置方式如下所示：

server:

port: 12345

spring:

application:

name: gateway

cloud:

gateway:

routes:

- id: user\_module

uri: lb://User

predicates:

- Path=/user/login

## 账号管理编码实现

首先是账号登录和注册界面。使用Element Plus里面的表单组件接收用户的输入。表单涉及到三种类型：输入框，多选框以及单选框。用来完成用户个人信息的输入。同时使用Vue3数据双向绑定的特点，在用户修改某项数据之后能够及时在视图界面得到展示。最后使用axios发送异步请求给后端进行处理。账号登录和注册界面如所示。



核心代码示例：

// 生成图片验证码

const generateCode = () => {

axios({

url:gatewayUrl+'/code/get',

method: 'get',

responseType:'blob',

}).then((resp) => {

const blob = new Blob([resp.data], { type: resp.data.type }); // 保存文件流为 Blob

const imageUrl = URL.createObjectURL(blob); // 转换 Blob 为 URL

checkCodeImg.value = imageUrl;

})

}

后端在账号管理微服务中主要实现三个Controller用来处理请求，分别是UserController用来处理账号信息的注册和变更，AvatarController专门用来处理用户上传的头像以及专门用来处理验证码的CheckCodeController。其中最核心的功能莫过于对用户登录次数进行控制——在用户登录失败次数超过三次的时候，将暂停用户登录1分钟。为此，借助Redis数据库。存储键login\_fail\_count，用来记录用户登录失败的次数。当用户失败的次数超过三次的时候，此时在存入键suspend\_login，同时设置过期时间为1分钟。当用户每次再登录的时候，首先检查suspend\_login是否存在，如果存在表明不允许用户进行登录。和上面逻辑相关的核心代码如下：

if(redisUtil.get("suspend\_login") != null){

return R.error("失败次数超过三次，将暂停一分钟登录");

}else{

if(redisUtil.get("login\_fail\_count") != null && (Integer)redisUtil.get("login\_fail\_count") >= 3){

redisUtil.del("login\_fail\_count");

}

}

if (user!=null &&user.getPassword().equals(password) && user.getState() == 1) {

//登录成功，删除键值

redisUtil.del("login\_fail\_count","suspend\_login");

return R.success(user, RofAccount.LOGIN\_SUCCESS.getMessage());

}

// 超过三次登录失败此时应当暂停登录一分钟

Object count = redisUtil.get("login\_fail\_count");

if (count == null) {

redisUtil.set("login\_fail\_count", 1);

} else {

redisUtil.set("login\_fail\_count", (Integer) count + 1);

if ((Integer) count + 1 >= 3) {

// 暂停登录一分钟

redisUtil.set("suspend\_login", "true", 60);

}

}

## 资讯分享编码实现

# 测试

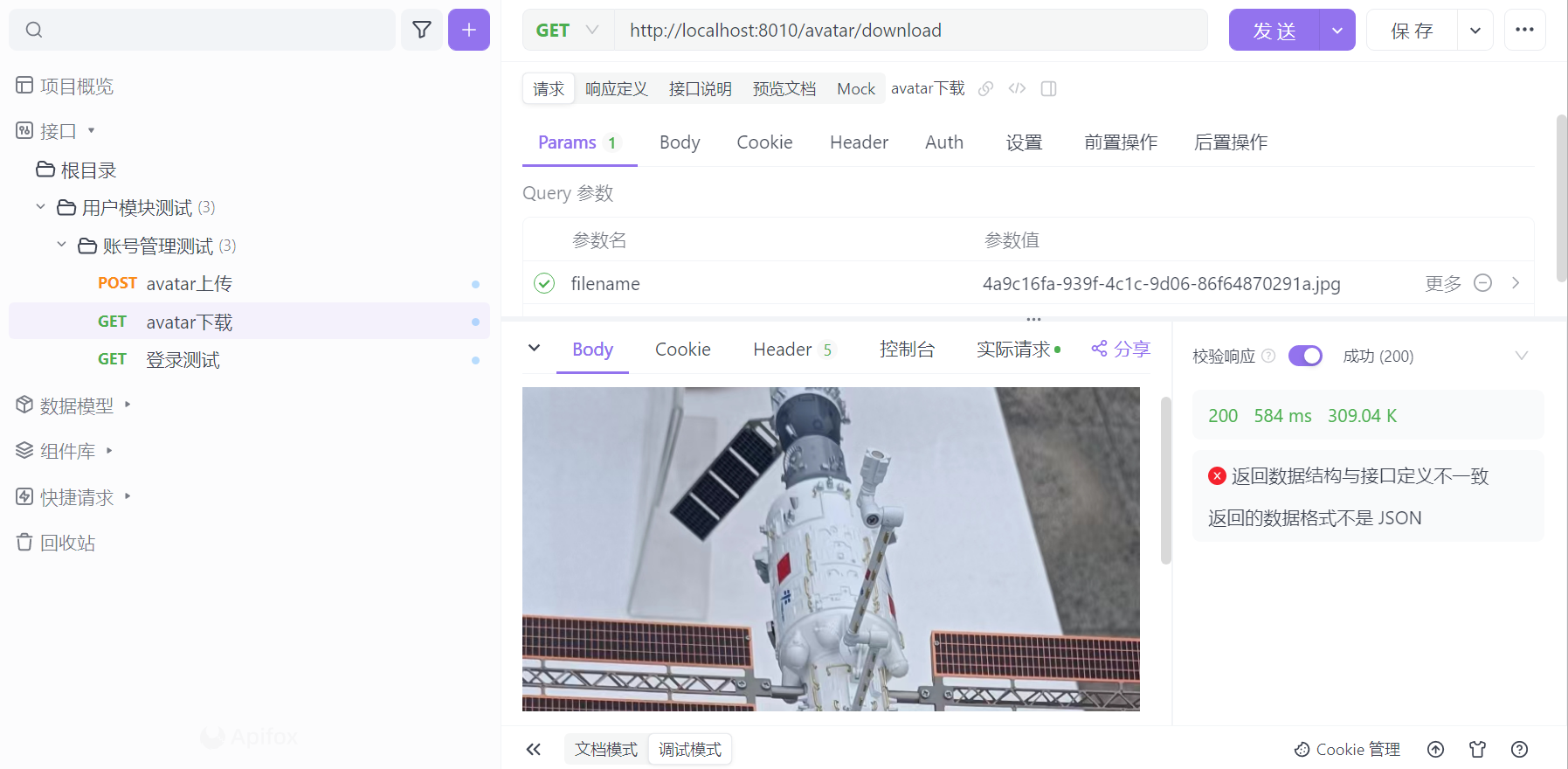
在进行项目测试的时候，首先对各个微服务进行单独测试。测试微服务对外暴露出来的接口是否可以正常工作，测试选用的工具为Apifox。对各个微服务测试完成之后，再对整个系统进行测试。模拟用户实际使用软件的场景，对正常操作流程和异常操作流程进行测试，最后得到系统完整的测试情况。

## 账号管理测试

账号管理测试主要测试账号的注册，登录以及账号信息的修改是否满足预期。测试的方式是模拟用户输入身份信息，这些信息可能符合输入信息格式要求也可能不符合，对于不符合输入格式要求的信息，系统应当给出相关提示来提升用户的网站使用体验。主要完成的测试工作如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 测试编号 | 测试项目 | 测试说明 | 测试结果 |
| 1 | 账号注册 | 首先按输入要求输入个人信息；然后不按照输入要求输入相关信息 | 前者账号正常注册；后者系统会提示错误信息，让用户完成注册 |
| 2 | 账号登录 | 首先正确输入账号密码和验证码信息，然后故意遗漏，录错上述信息 | 只有在用户正确输入账号信息时才能登录成功 |
| 3 | 上传头像 | 用户选择本地图片上传个人头像 | 测试通过 |
| 4 | 账号变更 | 用户修改密码等其他个人信息 | 测试通过 |

使用Apifox完成账号管理微服务部分测试如所示：



# 致谢