保密类别     学号  20180401162

**武汉晴川学院**

**毕 业 论 文（设 计）**

基于SpringCloudNetflix的在线外语学习

平台设计与实现

学 院 计算机学院

专 业 计算机科学与技术

学 生 章俊杰

指导教师 裴浪

2022年5月5日

**BACHELOR’S DEGREE THESISOF WUHAN QINGCHUAN UNIVERSITY**

**Design And Implementation Of Online Foreign Language Learning Platform Based On SpringCloudNetflix**

**Candidate：Zhang Junjie**

**Supervisor：Pei Lang**

May.5th,2022

**武汉晴川学院本科毕业论文（设计）**

**原创性声明**

本人郑重声明：

1．所呈交的毕业论文（设计），是本人在导师的指导下，独立进行研究工作所取得的成果。

2．除文中已经注明引用的内容外，本论文（设计）不含任何其他个人或集体已经发表或撰写过的作品或成果。

3．对本论文（设计）的研究做出重要贡献的个人和集体，均已在论文（设计）中以明确方式标明。

本声明的法律结果由本人承担。

论文作者签名：

年 月 日

# 摘　　要

在当下疫情背景下，随着网络教学的普及，以及当代大学生日均网上冲浪时间高涨，越来越多的大学生已经接受了网上学习这种非传统的学习方式，而且这种方式使得同学们能够灵活且合理的安排自己的学习时间。

由于传统单词记忆方式过于死板，不利于学生灵活学习，所以我们需要构建一个平台来为同学们提供听说读写一站式服务。本系统运用到的集成开发工具有IDEA、WebStorm,数据库使用到了Redis、MongoDB、Mysql、Elasticsearch，分别用来缓存、存储大文档数据、存储结构化数据、日志等需要检索的数据。

在学一学页面每个用户每天都能学习到后台专为用户推送的单词，让用户轻松了解该单词常用的短语、标准发音、相似单词、中文解释等。更有在线检索工具帮助用户精准查找特定单词的信息进行针对性学习。同时也为用户提供能练一练以及每日一句等板块，为用户提供听说读写的空间。为了方便用户观察近期的学习曲线，还特意为用户提供了详细的学习数据分析图供用户查看以调整当前学习方式。

关键词：IDEA；WebStorm；Mysql；Redis；MongoDB；Elasticsearch；

# **ABSTRACT**

In the context of the current epidemic, with the popularity of online teaching and the increasing daily online surfing time of contemporary college students, more and more college students have accepted online learning as a non-traditional way of learning, which enables students to flexibly and reasonably arrange their study time.

Since the traditional way of memorizing words is too rigid, which is not conducive to students' flexible learning, we need to build a platform to provide students with one-stop services of listening, speaking, reading and writing. The integrated development tools used in this system include IDEA and WebStrom. The database uses Redis, MongoDB, Mysql and ElasticSearch, which are used to cache and store large document data, store structured data, log and other data that need to be retrieved.

Every user can learn the words pushed by the background every day, so that users can easily understand the commonly used phrases, standard pronunciation, similar words, Chinese explanations of the word. There are also online retrieval tools to help users accurately find specific word information for targeted learning. At the same time, it also provides users with a practice and a sentence of the day and other sections to provide users with listening, speaking, reading and writing space. In order to facilitate users to observe the recent learning curve, detailed learning data analysis chart is specially provided for users to view and adjust the current learning mode.

**Keywords**：IDEA；WebStorm；Mysql；Redis；MongoDB；Elasticsearch

目　　录

[1　引言 1](#_Toc101538494)

[1.1　研究背景 1](#_Toc101538495)

[1.2　国内外研究现状 1](#_Toc101538496)

[1.3　本选题的研究内容 2](#_Toc101538497)

[1.4　本选题的研究意义 2](#_Toc101538498)

[1.5　论文的组织结构 2](#_Toc101538499)

[2　关键技术 4](#_Toc101538500)

[2.1　前端技术 4](#_Toc101538501)

[2.2　后端技术 4](#_Toc101538502)

[3　系统分析 6](#_Toc101538503)

[3.1　可行性分析 6](#_Toc101538504)

[3.2　系统功能分析 6](#_Toc101538505)

[3.2.1　系统功能需求分析 6](#_Toc101538506)

[3.2.2　前端功能模块分析 7](#_Toc101538507)

[3.2.3　后端功能模块分析 8](#_Toc101538508)

[3.3　数据库需求分析 9](#_Toc101538509)

[4　系统设计 11](#_Toc101538510)

[4.1　系统总体功能设计 11](#_Toc101538511)

[4.1.1　前端和后端功能设计 11](#_Toc101538512)

[4.2　数据库设计 19](#_Toc101538513)

[4.2.1　数据库表的设计 20](#_Toc101538514)

[4.3　小结 24](#_Toc101538515)

[5　系统实现 26](#_Toc101538516)

[5.1　系统功能模块实现展示 26](#_Toc101538517)

[5.2　数据库实现 44](#_Toc101538518)

[6　系统测试与调试 47](#_Toc101538519)

[6.1　测试目的 47](#_Toc101538520)

[6.2　测试概述 47](#_Toc101538521)

[6.3　单元测试 47](#_Toc101538522)

[6.3.1　用户注册测试 47](#_Toc101538523)

[6.3.2　单词发音音频分享测试 48](#_Toc101538524)

[6.3.3　单词检索功能测试 49](#_Toc101538525)

[6.3.4　单词管理测试 50](#_Toc101538526)

[6.3.5　习题管理测试 50](#_Toc101538527)

[6.3.6　用户管理测试 51](#_Toc101538528)

[6.4　集成测试 52](#_Toc101538529)

[7　总结与展望 53](#_Toc101538530)

[7.1　总结 53](#_Toc101538531)

[7.2　展望 53](#_Toc101538532)

[参考文献 55](#_Toc101538530)

# 1　引言

随着网课时代的到来，越来越多的同学开始在网络上进行相关知识的学习，也有越来越多的贡献者为广大网民们提供着海量的学习资料。线下课程一般上课时间固定进度恒定，但是每个同学的学习能力是不同的。但线上课程同学们可以根据自己的日程安排上课时间，网课时间没听明白的甚至能在课后重新学习。这种学习方式无疑给予了同学们更加灵活的时间安排。

## 1.1　研究背景

虽然目前市面上有很多非常流行的外语学习软件，但是我在使用这些软件的时候总会感觉自己和其他正在使用软件的陌生朋友们存在距离感。Goforit为广大学者们朋友提供了每个模块的评论板块，让志同道合的陌生人能够一同分享学习中的所感所获，异或是指正题库或词库的问题，以便巡查的管理员及时更正。Goforit为了拓展记忆，在每个单词的学习页面，会显示出贴切的图片、常用短语以及好记的例句。在我心中，语言的学习绝不仅仅只包含写，它理应包含“听说读”,Goforit为每个已登录的合法用户提供了单词学习模块的读音分享上传接口，让大家听见你的声音！Goforit还将推出点对点的事实视频聊天功能，为全站用户提供“听说读写”全方位的一站式服务。

## 1.2　国内外研究现状

国内现状：

百词斩：百词斩是一个以图片记忆为主的软件，图片能够给你的情境是非常有限的，虽然看起来比较直观，但不看到图片根本想不起来单词是啥，这也是图片记忆的一个弱点。

不背单词：拓展记忆这应该是它比较大的亮点，首先是词义多样，大家看到上面有核心词义，实际上还有一个引申词义，每一种词义都有一组例句，例句是来自一些比较出名的外文书籍、电影、电视剧、演讲稿等。

国外现状：

多邻国：界面可爱新颖，提示信息引人注意，包含各类音效，个人认为非常适合用来记单词。但是测试有点简单，更适合初学者或者单纯来背背单词的用户。

## 1.3　本选题的研究内容

本选题研究的内容主要是设计出为想要学习外语的同学们提供“听说读写”一站式服务的系统，同时还包括：

1.服务高容错性：保障极端情况下各个服务的可用性。

2.权限控制：不同角色的用户拥有不同的操作权限。

3.数据分析：分析用户的学习情况。

4.数据可视化：用手绘风的图表与炫酷的Echart结合展示出复杂抽象的数据。

5.前端动画：使用原生的帧动画设计出丝滑的动画。

6.分布式文件存储

7.定时调度任务

8.数据检索

## 1.4　本选题的研究意义

分析出目前主流的各类外语学习网站或APP的优缺点，最大限度的吸取它们的优势，补充完善它们不足的一些小细节。对于习外语需求的同学们来说，对这么多种类的学习软件或网站时，他们又多了一个选择，这个选择能够极大的提升他们学习效率。后台分析用户的学习情况，给用户提供最合理的学习建议，让用户学的更省心！

为想要学习各种外语的广大网友们提供“听说读写”一站式服务。分析每个合法用户的行为以及学习情况，分时推送目前最适合用户的学习信息。让用户能够仅仅在一个网站内，找到他们所需要的能为外语学习提供便利的功能模块。为有社交困难的朋友们，提供一个练习口语的平台；为健谈的朋友们提供一个畅所欲言的环境；为渴望交流的朋友们提供一个即使通讯的窗口。针对不同基础的人群提供不同的学习通道，测试题也会跟着用户的学习难易程度进行变更。后台定时分析用户的学习情况，通过生动形象的图标展示给用户，让用户能够清晰的看见自己近期的学习状况，系统也会更具这些信息给出用户合理的建议。

## 1.5　论文的组织结构

本系统总共分为七个章节，论文的具体章节结构安排如下：

1.引言：介绍了本次研究项目的背景、国内外现状以及目的和意义。

2.关键技术：主要介绍本系统所运用的相应技术。

3.系统分析：包含可行性分析、系统功能分析，系统功能分析包括系统功能需求分析、数据库需求分析、前端功能模块分析、后端功能模块分析。

4.系统设计：包含了系统总体功能设计、数据库设计，关键技术的设计思路。

5.系统实现：各系统功能模块的展示和数据库实现，包含一些重要代码的展示。

6.系统测试与调试：主要包括测试目的、测试概述以及单元测试。

7.总结与不足：对整个系统做一个总结以及回顾。

# 2　关键技术

## 2.1　前端技术

前台使用渐进式的Javascript框架Vue,前端视图组件库使用到了ElementUI以及ViewUI,本站摒弃了Ajax转而使用了一个基于Promise的网络请求库Axios。前台路由使用到了VueRouter，统一状态管理使用了Vuex。其他开源组件库有用到Hover.css、Scss、FullPage.Js、Echarts等等。

整个前台系统使用Vue-cli脚手架进行快速开发,不同于往常的开发，此次前端的样式设计模块并未使用css而是选择了新起之秀Scss。

Hover.css里面内置了大量的hover样式供使用者选择。FullPage.js为本站提供了十分华丽的单页面全屏滚动功能，Echart为本站的数据分析统计页面提供了炫酷的图表组件。

## 2.2　后端技术

后台开发使用的是微服务架构而不是传统的单体式应用，网关层没用使用市面上常用的SpringCloud Gateway,而是自定义的网关，方便后期进行自定义操作。服务调度使用到了Openfeign，各微服务之间的注册中心使用的是Netflix旗下的Eureka而不是Zookeeper，由于本站做了比较好的系统防护功能不仅使用了基于GuavaRatelimiter的令牌桶限流算法对关键接口进行限流操作，还在各层之间使用redis做了失效30分钟的缓存,甚至还使用了Netflix的Hystrix对每一个接口做了服务降级的处理。本站的注册功能需配合接收到的邮箱提供的验证码在能够注册成功，所以这里有使用到阿里云的邮件服务，用户上传的文件以及本站所词库、题库、发音库都使用了MongoDB和阿里云的Oss对象存储服务进行持久化。使用Mysql对一些结构性数据(例如：用户信息)进行持久化。由于本站使用了SSO单点登录，所以我有使用到JWTToken用来标识用户。本站还有日志数据分析模块，使用到了Elasticsearch和MongoDB来双写持久化日志以及用户的做题记录等信息，为了削峰解耦使用到消息队列RabbitMQ来传送消息，并使用Kibana进行可视化显示，最终定时调度任务分析各用户的学习情况，推送建议给用户。

由于项目工程量较大，开发时间跨度大，所以有时用到Git进行分布式版本控制，所有的源代码放在Gitee私有仓库进行统一管理，项目的构建等依赖于Maven，Mysql、Nginx、MongoDB等服务也是运行在容器Docker之中，极大简化了部署的难度以及流程。

# 3　系统分析

## 3.1　可行性分析

可行性分析是为了花费最小的人力和物力来确定问题能否被解决，避免过早盲目开发造成不可估量的损失。

首先从目前比较流行的几款外语学习软件来看，它们分别有着下列这些小瑕疵：

1.视频学习中穿插了大量的广告，偏离了视频交流的核心作用，商业化过于严重。

2.小部分APP页面相关按钮排版较凌乱，逻辑关系不清晰。

3.各家产品都有各自的特点，“听说读写”基本不能再一个App中实现。

基于以上传统选修课的报名方式的困难，至此开发了基于SpringCloudNetflix的在线外语学习平台。用户使用该平台后，就能够体验到“听说读写”一站式服务，不用来回辗转于各大App之间。并且由于平台拥有对日志及用户行为信息的分析统计功能，可以很好的向用户展示他目前的学习情况，让用户清晰的了解到目前的学习进度以及效率。

## 3.2　系统功能分析

根据在线外语学习平台的各功能分析，对该系统的整体的功能和模块进行布局和分布安排，对用户的使用操作以及系统的进程有了比较清晰的思路。

首先是对系统功能的需求分析，其次为前端功能模块分析，然后是后端各微服务模块分析，最后是对数据库的需求分析。

### 3.2.1　系统功能需求分析

普通用户视角：

1.登陆注册注销

2.对不同类别词汇（少儿英语、四级英语、六级英语、程序员英语、俄语）以及相似单词、短语等基本信息的查询功能。

3.发表评论功能

4.举报不当发言功能

5.分享自己的单词发音功能

6.单词精准查询功能

7.练习题操作功能

8.查看自己的学习情况分析图表功能

9.查看Rank排行榜功能

10.进行视频交流功能

开发者与管理员视角：

1.单词基本信息的增删改查操作功能

2.相似单词基本信息的增删改查操作功能

3.相关短语信息的增删改查操作功能

4.题库的增删改查操作功能

5.对用户的封禁与解封禁操作功能

6.对不当言论的审核操作功能

7.对题目反馈信息的操作功能

8.对各系统运行情况的监控功能

9.日志可视化功能

### 3.2.2　前端功能模块分析

总体为SPA单页面应用，通过各类点击事件配合VueRouter进行路由操作实现页面刷新和弹出层的显示。

前端功能模块主要包括下列十类（如图3.1所示）：

1.用户登陆注册注销：用于显示用户登陆注册注销界面，所有人可见。

2.首页概述页：简单概述本站的基本情况以及开发者信息，所有人可见。

3.学习单词页面：本站核心页面之一，用于显示用户待学习单词队列的各4.单词信息以及其他信息等，仅已登录用户可见。

5.练习页面：本站核心页面之一，用于显示用户当日的待学习单词队列相对应的习题，仅已登录用户可见。

6.排行页面：用于显示不同分类下的排行榜，所有人可见。

7.视频交流页面：为用户提供视频交流，仅已登录用户可见。

8.个人信息页面：显示用户基本信息，仅已登录用户可见。

9.设置页面：显示用户自定义的各类设置，仅已登录用户可见。

10.服务健康情况监控页面：显示各微服务之间的健康情况，仅管理员可见。

11.后台管理页面：用于辅助管理员管理平台各类信息，仅管理员可见。

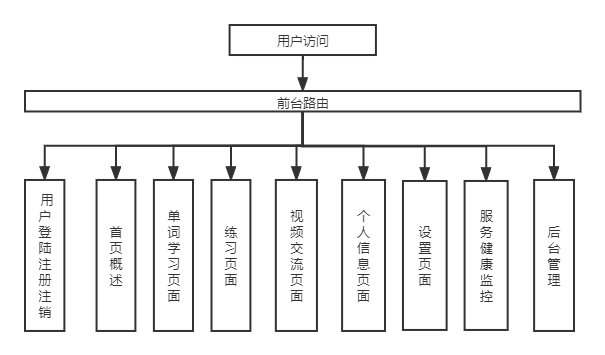


图3.1前端功能模块图

### 3.2.3　后端功能模块分析

后端为微服务架构，所有请求统一经由网关层进行转发聚合后返回给前台视图层进行数据展示。

后端功能模块主要包括（如图3.2所示）：

1.后台管理模块（goforit-admin）:为管理员提供数据操作的接口。

2.分析模块（goforit-analyse）：用于分析用户生成的各类信息

3.认证模块（goforit-auth）：用于验证用户身份以及提供登陆注册功能。

4.注册中心（goforit-eureka）：作为各微服务的注册中心。

5.网关（goforit-gateway）：统一网关，前台所有请求都得通过网关。

6.监控模块（goforit-monitor）：用于监控各微服务运行情况。

7.习题模块（goforit-practice）：为习题模块提供后台各类功能支持。

8.推送模块（goforit-push）：推送各类信息，例如各用户的待学习单词队列。

9.排行模块（goforit-rank）：定时调度计算排行数据。

10.单词学习模块（goforit-recite）：为单词学习模块提供后台各类功能支持。

11.三方服务模块（goforit-thirdparty）：提供各类三方服务，例如阿里云邮件服务以及AliyunOSS等。



图3.2后端功能模块图

## 3.3　数据库需求分析

针对选修课报名后端管理的需求和实际的需要对数据库进行详细的设计和安排，做好相应的需求分析和大数据分析。

数据库需要使用：

1.Mysql对结构化数据（用户信息、单词信息等）进行持久化：

* 1. User表：存储用户信息
  2. Word表：存储单词基本信息
  3. Learningprocess表：记录用户学习进度
  4. Meaning表：存储各单词的中文解释
  5. Share\_sound表：存储用户上传的发音文件存放位置信息等
  6. Phrase表：存储单词的相关短语

2.MongoDB对文档型数据进行持久化：

* 1. 待学习单词队列
  2. 题库
  3. 作业提交详情
  4. 排行数据

3.Redis对所有查询数据进行统一缓存管理、保存一些临时变量：

4.ElasticSearch对日志信息等进行持久化：

* 1. 网关层日志
  2. 服务层日志
  3. 持久层日志
  4. 其他日志
  5. 用于检索的全量单词基本信息

# 4　系统设计

## 4.1　系统总体功能设计

本系统需要实现用户的登陆注册注销、学习单词、修改个人信息、做题、查看个人学习情况分析数据、查看排行榜、分享音频、发表评论、举报反馈、每日一句及经验值增长功能；管理员进行用户管理、习题上传修改、处理举报反馈信息、查看服务运行情况等的功能。

### 4.1.1　前端和后端功能设计

1.注册功能实现

前台传递注册信息（用户名、密码、邮箱、验证码、生日、性别）给后台，网关校验验证码等信息后完成注册，返回前台状态码。前台接收到状态码之后通过Notify组件弹窗提示注册成功，或者通过读取响应码相应信息进行相应的信息反馈。流程图如下图4.1所示：

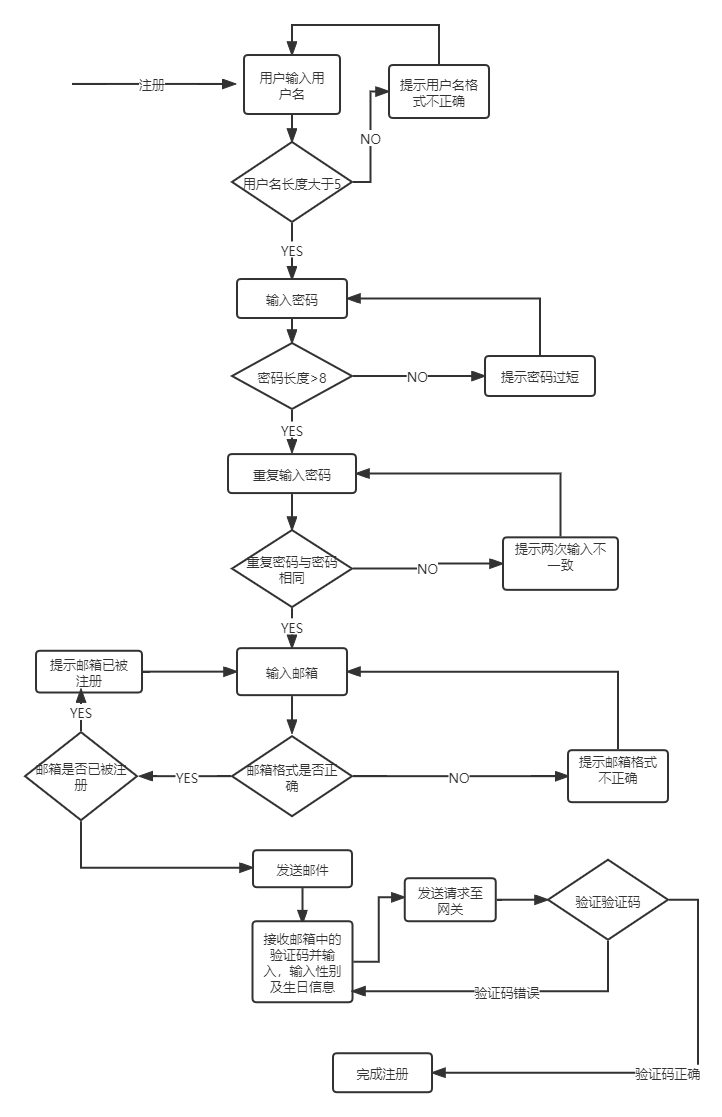


图4.1　用户注册操作流程图

2.用户学习单词

用户点击学一学，前台发送请求至后台，后台接收到请求后，调用GOFORIT-PUSH服务查询用户今日的待学习单词队列，链式调用Recite模块查询单词基本信息、相似单词信息、相关短语、单词讨论、发音分享等信息，然后返回给前台进行数据展示，如图4.2。

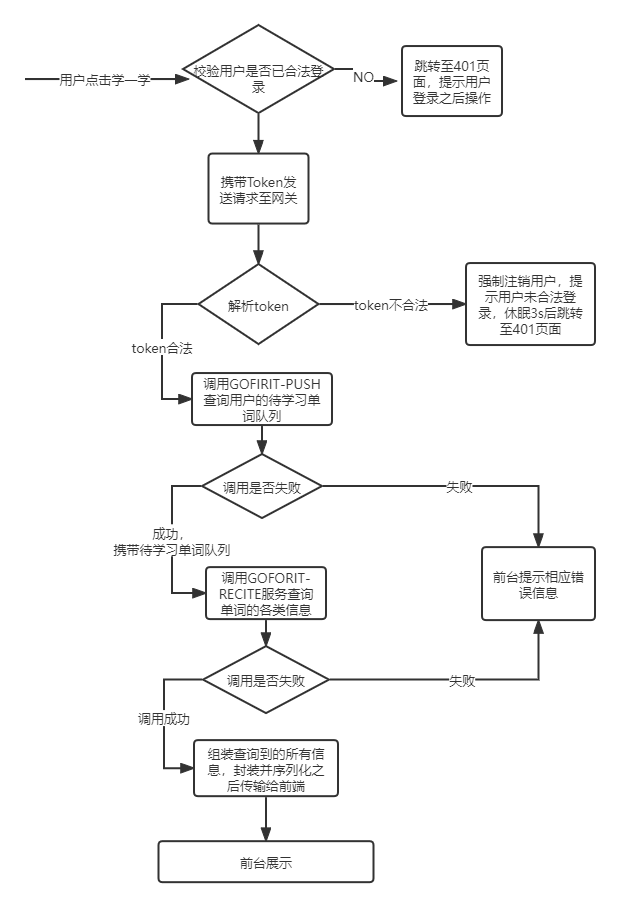


图4.2用户学习单词操作流程图

3.用户做题

查询：

用户点击练一练，前台发送请求至后台，后台接收到请求后，调用GOFORIT-PUSH服务查询用户今日的待学习单词队列，链式调用Practice模块查询相应单词所对应的选择题及填空题，为每个单词随机选择题目之后惊醒shuffle运算然后乱序输出。前台接收到信息后进行展示。流程如图4.3所示。

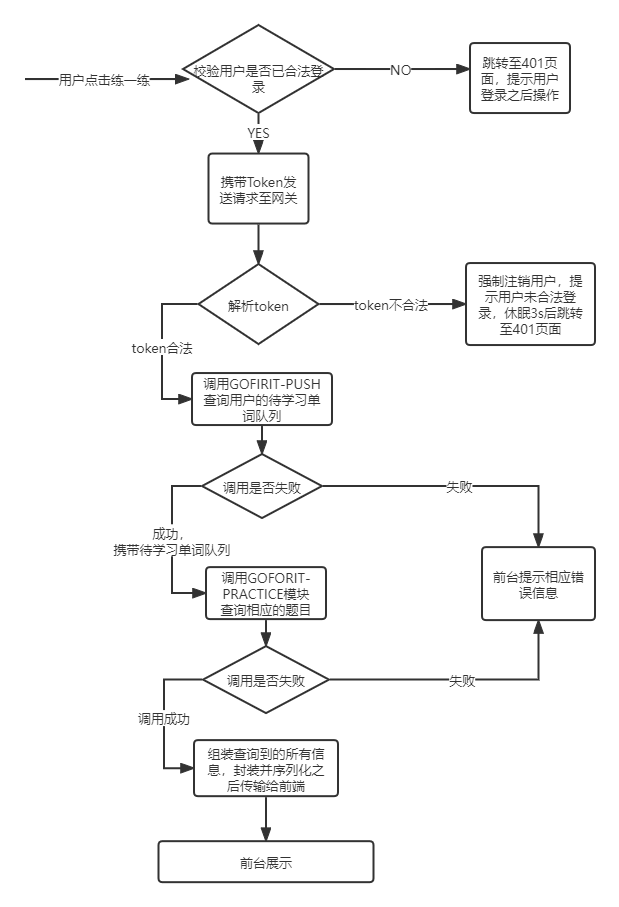


图4.3练一练查询操作流程图

提交：

用户掉进右下角机器人头像进行提交，前台会校验题目是否全部填写， 完成填写则会提交上传，否则提示反馈错误。流程如图4.4所示。

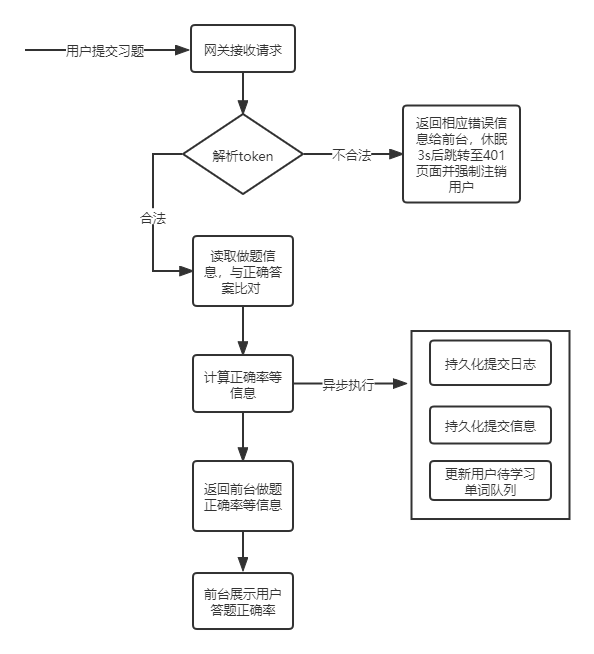
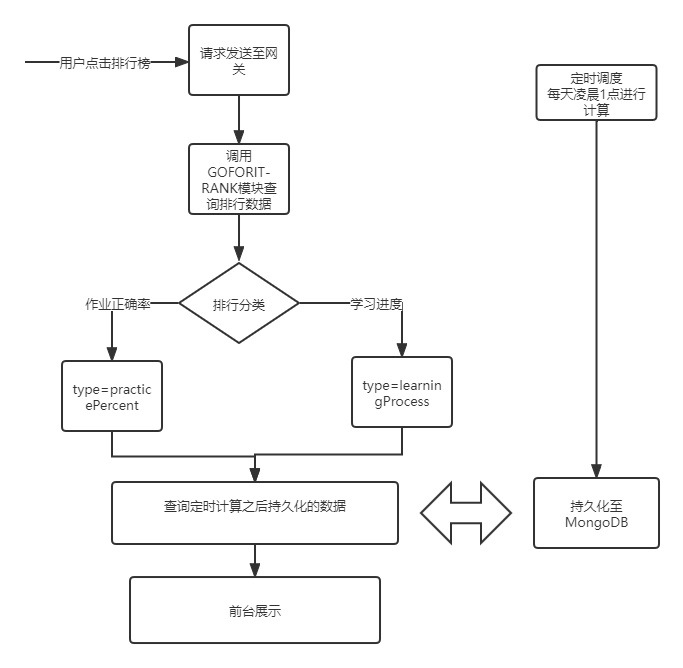


图4.4　练一练提交操作流程图

4.查看排行榜

GOFORIT-RANK模块每天定时计算用户的排行数据并持久化数据至MongoDB，方便日后复用统计数据。网关接收到前台请求的查询Rank信息时，会读取相关的排序规则，例如四级单词模块按学习进度排序等。然后调用Rank模块暴露出来的接口查询缓存，如若缓存存在数据则直接返回，否则出擦寻MongoDB然后缓存此数据。具体流程如图4.5所示：



图**4.5**　排行榜查询操作流程图

5.分享音频：

用户在Recite页面单词详细信息页面中可以提交啊自己对此单词的发音，让所有用户都能听到你的声音。

前台发送上传请求，会先判断上传文件的类别是否为mp3或者wav文件，不是就会禁止上传并提示用户。调用三方服务模块中的AliyunOSS服务上传文件之后持久化文件路径，此时异步刷新Redis中缓存的此单词相关的所有share\_sound信息，并记录当前时间的日志供后续数据分析使用。具体流程如图4.6所示。

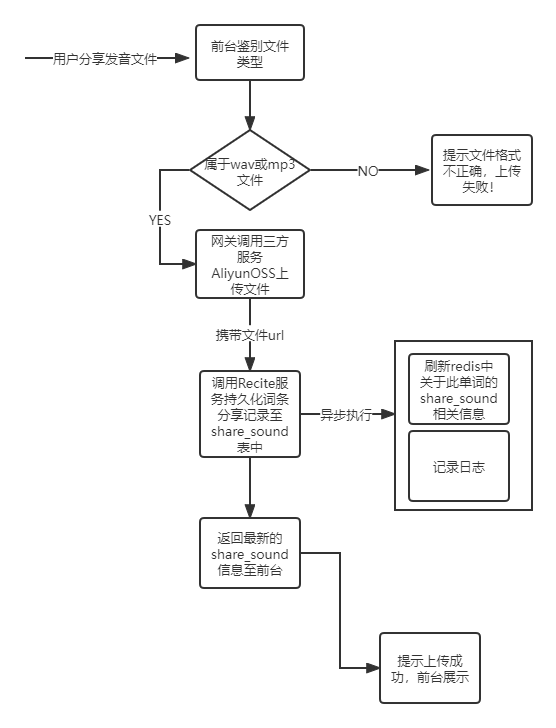


图4.6　音频分享操作流程图

6.后台词库单词录入

本站并未暴露任何进入后台管理页面的按钮给用户，而是需要键入相应的url进入后台管理系统。进入后台管理系统后会先校验用户的权限，如若不是管理员则会提示权限不足拒绝访问，管理员需要数据单词基本信息、相似单词信息、相关短语等基本信息，然后上传该单词的标准读音，最终持久化至Mysql，之后刷新redis缓存与此单词相关的基本信息并记录当前操作日志。具体流程如图4.7所示。

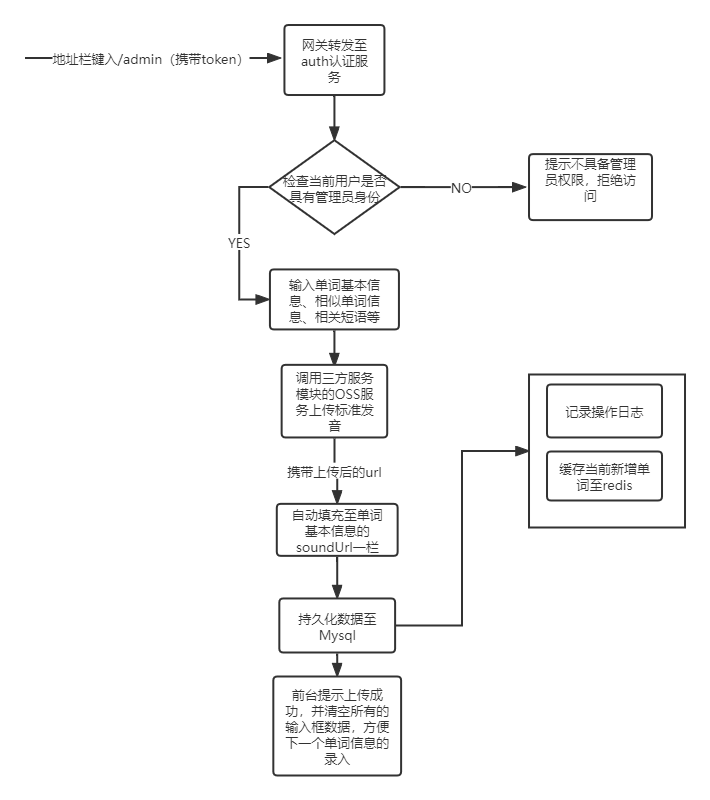


图4.7后台词库单词录入操作流程图

7.单词检索功能

用户在非首页的任意页面可以通过双击CTRL键呼出单词查询框，此功能不会校验用户权限，任何人都可以使用此功能。在管理员及开发者新增词库信息的时候，在数据入库Mysql的同时也会有一份相同的数据流入ElasticSearch，之后会对单词进行分词，当用户键入单词的时候会只能推荐出最符合的至多5个单词，用户点击按钮即可跳转至相应的详情展示页面。检索推荐的单词会带上标准发音文件，让用户能够快速查询学习单词。具体流程如图4.8所示。

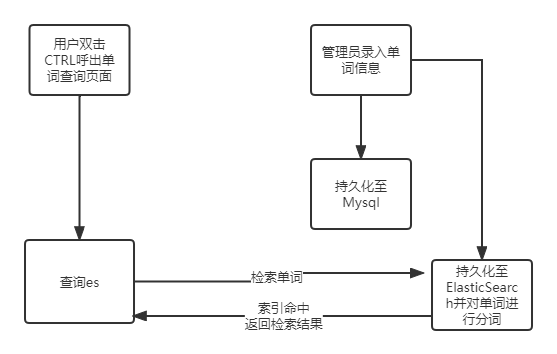


图4.8　查询单词流程图

8.经验值增长功能

由于本系统存在用户单词学习页面、练习页面、以及一些提供分享服务的类似交流大厅功能，且绝大部分目标用户都会是游戏玩家，对经验值以及对应的等级有一定的喜好，故特设经验值模块，每升一级所需的经验值为100点，当用户浏览不同的单词页面时会给予用户2点经验值，提交作业会给予用户5点经验值，分享自己的发音等也会增长用户的经验值，依托此成长体系让用户养成每天学习的习惯。经验值增长流程图4.9所示。

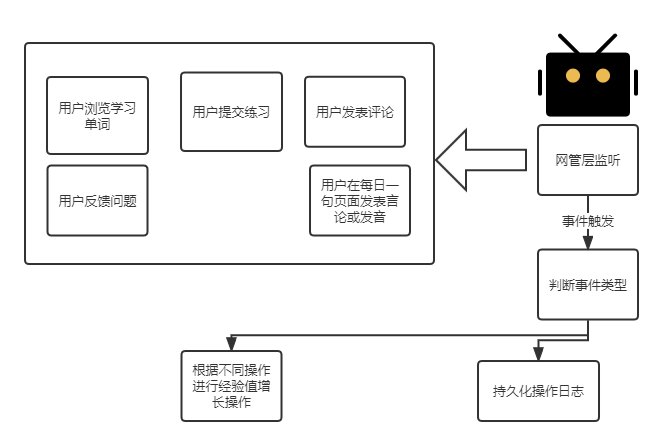


图4.9　经验值增长流程图

9.用户近期操作记录分析统计

每天定时统计用户操作信息，如果服务重启会先查找调度分析操作的日志，如果当日并未发生调度工作则会主动进行一次调度聚合用户操作信息，然后聚合之后的数据流转向mysql进行持久化备份，之后在用户调取个人分析数据时设置缓存，大大降低查询延时以及数据库压力。在此操作中主要会聚合运算用户一年内每天的学习热力图数据、各分类单词学习进度、近七天作业提交次数、七天内每日平均作业提交正确率、一周内用户发表言论及分享发音文件频率等。详细调度流程如图4.10所示。

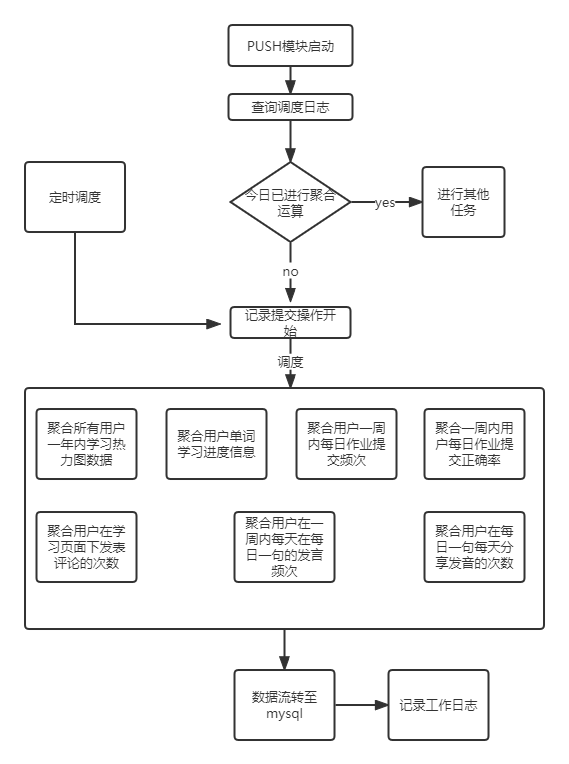


图4.10　用户数据分析流程图

## 4.2　数据库设计

在本项目中关系型数据库以及非关系型数据库共使用了四款。所使用的关系型数据库为Mysql，在对Mysql的设计中满足了三大范式。非关系型数据库用到了基于内存的Redis、文档型数据库MongoDB、以及能够提供高速检索功能的ElasticSerach。

### 4.2.1　数据库表的设计

其中Mysql表的结构如下：

1.表learningprocess (用户学习进度表（练习成绩表（复习成绩不计入其中））)，用于存放每个用户对应类别词汇的学习进度

表**4.10**learningprocess

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| id | int | 10 | 0 | N | Y |  |
| username | varchar | 30 | 0 | N | N | 用户名 |
| subject | varchar | 20 | 0 | N | N | 例：四级英语 |
| haveLearned | int | 10 | 0 | N | N | 以学习单词个数 |
| practicePercentAll | double | 22 | 0 | N | N | 作业正确率累加和 |
| submitCount | int | 10 | 0 | Y | N | 提交次数 |
| practicePercent | double | 22 | 0 | Y | N | 平均正确率 |

2.表meaning (中文对照表):用于存放单词的中文意思，每个单词都可以拥有不止一个中文意思。

表**4.11**　meaning

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| id | varchar | 255 | 0 | N | Y |  |
| word | varchar | 255 | 0 | N | N |  |
| type | varchar | 50 | 0 | N | N |  |
| chinese | varchar | 255 | 0 | Y | N |  |

3.表phrase，此表用于存放单词常见的短语以及该短语的中文翻译，在单词学习页面会随着单词的基本信息一起推送至前台页面，供用户学习。

表**4.12**　phrase

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| id | int | 10 | 0 | N | Y |  |
| word | varchar | 255 | 0 | N | N |  |
| phrase | varchar | 255 | 0 | Y | N |  |
| chinese | varchar | 255 | 0 | Y | N |  |
| updateTime | date | 10 | 0 | Y | N |  |
| updater | varchar | 100 | 0 | Y | N |  |

4.表share\_sound (用户在recite页面上传的音频（单词发音）)，用户在学习页面分享完特定单词发音之后，文件会上传至阿里云对象存储，相应的文件url就会持久化至这张表中，供后续用户查询相应单词时响应给前台。

表**4.13**　share\_sound

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| id | bigint | 19 | 0 | N | Y |  |
| uploader | varchar | 100 | 0 | Y | N |  |
| sound\_url | varchar | 255 | 0 | Y | N |  |
| upload\_time | datetime | 19 | 0 | Y | N |  |
| word\_for | varchar | 100 | 0 | Y | N | 单词 |

5.表user，此表用于存放用户的基本信息。

表**4.14**　user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| uid | varchar | 255 | 0 | N | Y |  |
| username | varchar | 50 | 0 | N | N |  |
| password | varchar | 50 | 0 | N | N |  |
| gender | int | 10 | 0 | Y | N | 1：男 0：女 |
| birthday | date | 10 | 0 | Y | N |  |
| email | varchar | 100 | 0 | N | N |  |
| tel | bigint | 19 | 0 | Y | N |  |
| joinTime | date | 10 | 0 | Y | N |  |
| lastLoginTime | date | 10 | 0 | Y | N |  |
| avatarUrl | varchar | 255 | 0 | Y | N |  |
| role | varchar | 30 | 0 | Y | N |  |
| state | int | 10 | 0 | Y | N | 1:有效 0：无效 -1：baned |

6.表word (单词表)，此表用于存放各类单词的基本信息。

表**4.15**　word

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| id | int | 10 | 0 | N | Y |  |
| word | varchar | 50 | 0 | N | N |  |
| soundUrl | varchar | 255 | 0 | Y | N |  |
| imgUrl | varchar | 255 | 0 | Y | N |  |
| symbol | varchar | 255 | 0 | Y | N |  |
| addTime | date | 10 | 0 | Y | N |  |
| subject | varchar | 255 | 0 | Y | N |  |
| root | varchar | 50 | 0 | Y | N | 词根 |

7.表aggregation\_log (用户统计信息聚合表)，此表用于每天定时任务持久化对用户近期学习情况的统计数据，这样就大大降低了实时分析用户数据所产生的巨大开销。甚至用户查看自己近期学习情况的请求延时也大幅降低了，极大的保证了用户体验。

表**4.16**　aggregation\_log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 小数位 | 允许空值 | 说明 |
| id | bigint | 0 | N |  |
| username | varchar | 0 | N | 用户名 |
| learning\_process\_child | int | 0 | Y | 少儿英语单词已学数量 |
| learning\_process\_cet4 | int | 0 | Y | 四级英语单词已学数量 |
| learning\_process\_cet6 | int | 0 | Y | 六级英语单词已学数量 |
| learning\_process\_programer | int | 0 | Y | 程序员英语单词已学数量 |
| learning\_process\_russian | int | 0 | Y | 俄语单词已学数量 |
| practice\_submit\_count\_child | int | 0 | Y | 少儿英语练习提交次数 |
| practice\_submit\_count\_cet4 | int | 0 | Y | cet4练习提交次数 |
| practice\_submit\_count\_cet6 | int | 0 | Y | cet6练习提交次数 |
| practice\_submit\_count\_programer | int | 0 | Y | 程序员英语练习提交次数 |
| practice\_submit\_count\_russain | int | 0 | Y | 俄语练习提交次数 |
| send\_comment\_log\_7\_days\_count | varchar | 0 | Y | 七天内再单词页面发表言论次数 1,2,3,11 逗号分隔 |
| send\_daily\_sentence\_comment\_log\_7\_days\_count | varchar | 0 | Y | 七天内在每日一句中发表言论次数（comment+soundUrl） |
| send\_daily\_sentence\_soundUrl\_log\_7\_days\_count | varchar | 0 | Y | 七天内在每日一句中发表发音次数(仅计算soudUrl次数) |
| practice\_result\_7\_days\_child | varchar | 0 | Y | 七天内作业提交平均正确率-child |
| practice\_result\_7\_days\_cet4 | varchar | 0 | Y | 七天内作业提交平均正确率-cet4 |
| practice\_result\_7\_days\_cet6 | varchar | 0 | Y | 七天内作业提交平均正确率-cet6 |
| practice\_result\_7\_days\_programer | varchar | 0 | Y | 七天内作业提交平均正确率-programer |
| practice\_result\_7\_days\_russian | varchar | 0 | Y | 七天内作业提交平均正确率-russian |
| learning\_heat\_map | varchar | 0 | Y | 一年内的学习热度图，存放["2022-1-1",100],["2022-3-1",200]逗号分割 |
| seven\_days | varchar | 0 | Y | 七天的日期逗号分隔 2022-01-03,2011-05-02 |
| refresh\_date | date | 0 | Y | 刷新日期 |

8.表daily\_sentence，此表用于缓存每日一句的基本信息，此表的数据也会对后期的数据分析提供有力的帮助。

表**4.17**　daily\_sentence

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| id | bigint | 19 | 0 | N | Y |  |
| sentence | varchar | 255 | 0 | N | N | 每日一句 |
| chinese | varchar | 255 | 0 | N | N | 中文翻译 |
| publish\_man | varchar | 255 | 0 | Y | N | 发布者 |
| publish\_date | date | 10 | 0 | Y | N | 发布时间 |
| sound\_url | varchar | 255 | 0 | Y | N | 发音地址 |

9.表daily\_sentence\_share\_log，此表记录了用户在每日一句模块的一言一行，包含用户所发送的文字以及语音地址。

表**4.18**daily\_sentence\_share\_log

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| id | bigint | 19 | 0 | N | Y |  |
| username | varchar | 255 | 0 | N | N | 用户名 |
| daily\_sentence\_id | bigint | 19 | 0 | N | N | 每日一句主键 |
| sound\_url | varchar | 255 | 0 | Y | N | 发音地址 |
| comment | varchar | 255 | 0 | Y | N | 言论 |
| publish\_time | datetime | 19 | 0 | Y | N | 发表时间 |
| state | int | 10 | 0 | Y | N | 标记位 1：可用 0：封禁 |

10.表experience。此张表被用来存储用户的经验值，每1获得100经验值用户能够升一级，通过升级激发用户的兴趣。

表**4.19**　experience

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| id | int unsigned | 10 | 0 | N | Y | 自增主键 |
| username | varchar | 255 | 0 | N | N | 用户名 |
| experience | bigint | 19 | 0 | Y | N | 经验值 |
| update\_time | datetime | 19 | 0 | Y | N | 上次修改时间 |
| update\_msg | varchar | 255 | 0 | Y | N | 上次修改附带信息 |

11.表full\_analyse\_job\_log (统计分析数据任务执行记录表)，此表记录了每次push模块触发分析任务的日志，通过日志的方式记录分析状态，方便了后期的运维工作。

表**4.20**　full\_analyse\_job\_log

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| id | bigint | 19 | 0 | N | Y |  |
| trigger\_way | varchar | 100 | 0 | N | N | 触发方式：启动触发/定时任务 |
| duration | bigint | 19 | 0 | Y | N | 耗时 ms |
| job\_begin\_time | datetime | 19 | 0 | Y | N | 任务开始时间 |
| job\_end\_time | datetime | 19 | 0 | Y | N | 任务结束时间 |
| log\_time | datetime | 19 | 0 | Y | N | 记录时间 |

## 4.3　小结

在开发本系统的过程中可谓是踩坑无数。一方面是由于自己的开发经验不足，另一方面是一味追求新技术。

由于本系统是微服务架构的，不可避免的就会由很多提供不同功能的微服务模块组成。开发前期，认为微服务模块越多越能体现一个系统的健壮性，诚然，但也并不完全正确。随着微服务模块的增多，服务之间的调度以及管理问题就变得越来越棘手。在网关层最容易发生的问题就是B服务的入参是通过A服务查询获得的。服务正常的情况下，想要得到正确的值是没问题的，但是当A阻塞或者宕机就极大可能会导致整个服务链路的不可用，最终导致服务不可用，极大影响用户体验。现在我才意识到服务多并一定是一件值得炫耀的事情，服务多并能高效合理的掌控整个业务流程的流向才是一个值得炫耀的高级架构。开发中期我才意识到正常的企业开发会把常用的实体类抽取到一个公共模块中，让各个服务依赖此公共模块，以此来简化开发。呆呆的我遇到不同服务模块需要使用相同实体类的时候，会选择在这两个模块中分别建立一个相同的对象。这样虽然也不影响程序的执行，但是一旦后期业务需求有变更，需要修改实体类，那这个工程难度就有点大了，毕竟对于一个开发时间间隔这么久的工程来说，寻找哪些业务引用了此类也是一个很繁琐的过程。开发中遇到的问题很多，但是在这段CODING和DEBUG的旅程中，我也学会了很多新技术的使用细节，以及一些开发的标准化流程与标准，这必定会在我未来的路上起到至关重要的重要。

# 5　系统实现

## 5.1　系统功能模块实现展示

1.用户注册

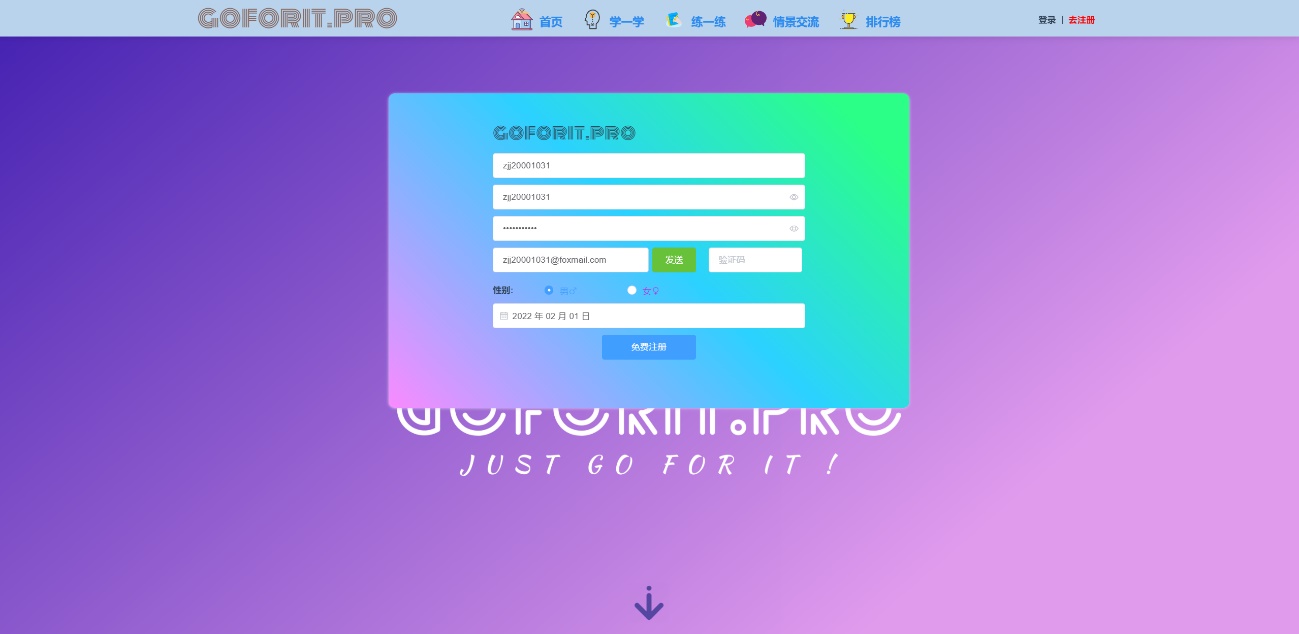
输入用户名、密码、重复密码、邮箱、性别及生日，如图5.1所示：

图5.1　用户注册填写信息

点击“发送”接收到邮件如图5.2所示：

图5.2邮箱接收验证码

然后填写验证码，点击“免费注册”发送请求之后端进行注册，反正这后返回状态码至前台进行展示如图5.3所示：

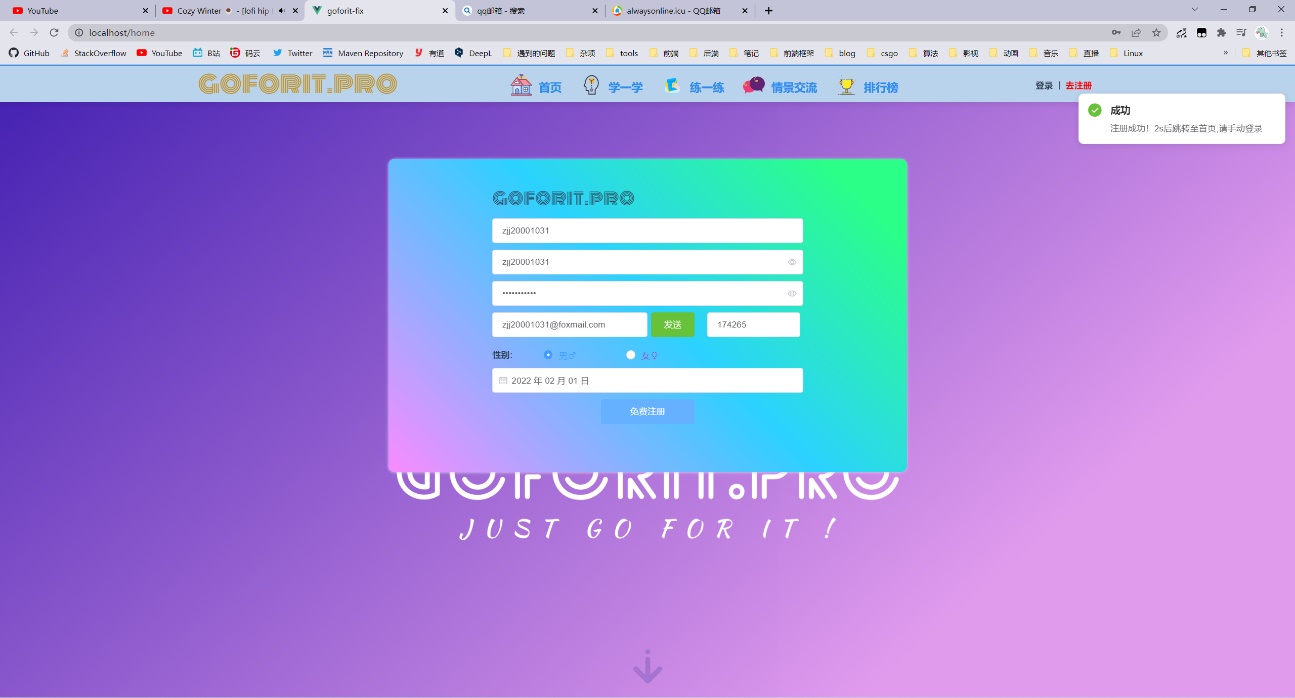


图5.3　注册成功

网关层用户注册代码实现：

@PostMapping("/registe")

@Limit(needPermitCounts = 10,useR2Return = false)

@HystrixCommand(fallbackMethod = "fallback\_registe")

public String registe(@RequestBody Map map){

String validateCode = (String) map.get("validateCode");

String username = (String) map.get("username");

String email = (String) map.get("email");

String code = validateCodeService.getValidateCodeByUsernameAndEmail(username, email);

String result = "验证码不正确";

if (validateCode.equals(code)){

result = authService.registe(map);

//删除redis中缓存的的验证码

validateCodeService.deleteValidateCodeByUsernameAndEmail(username,email);

}

return result;

}

前端用户注册代码如下：

doRegist(){

if (this.validateStep1() && this.validateStep2()){

Axios.post("/registe",{

username:this.username,

password:this.password,

email:this.email,

validateCode:this.validateCode,

gender:this.gender,

birthday:new Date(this.birthday).getTime()

}).then(res=>{

console.log(res.data);

if (res.data==="注册成功"){

this.success('注册成功！2s后跳转至首页,请手动登录')

setTimeout(()=>{

location.reload()

},2000)

}else{

this.error(res.data)

this.validateCode=''

}

}).catch(err=>{

this.error("网络异常")

console.log(err.message);

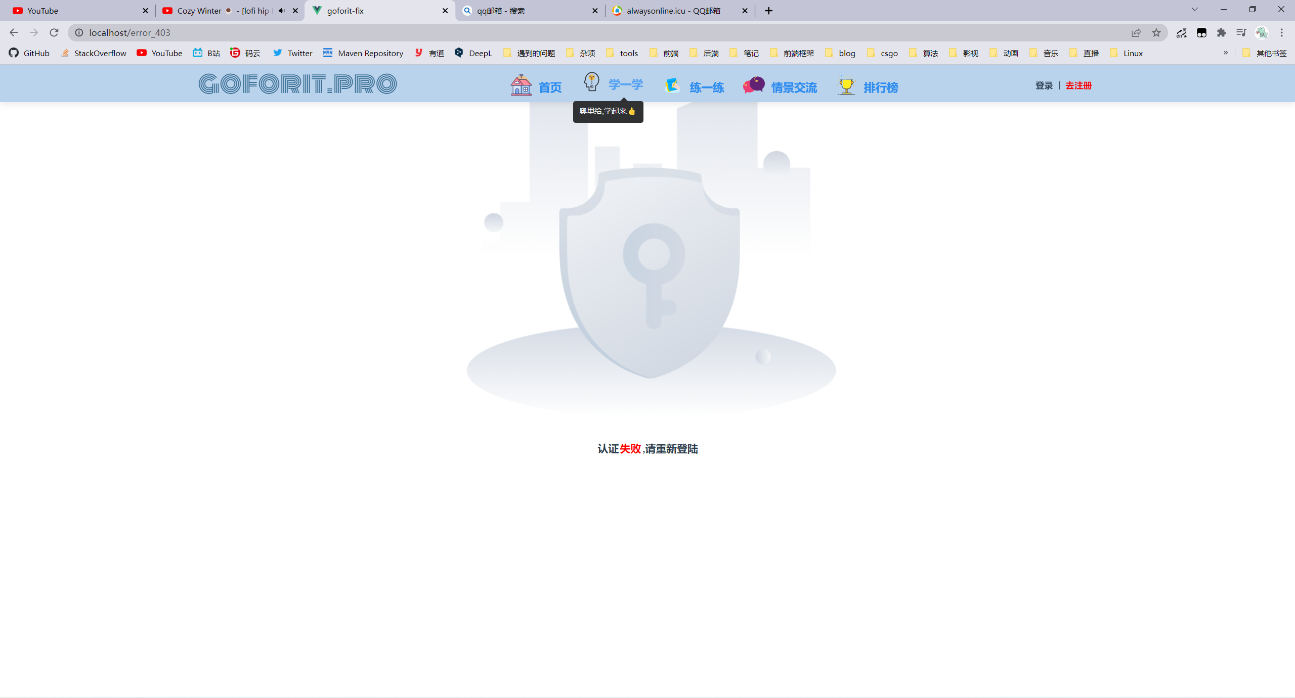
})

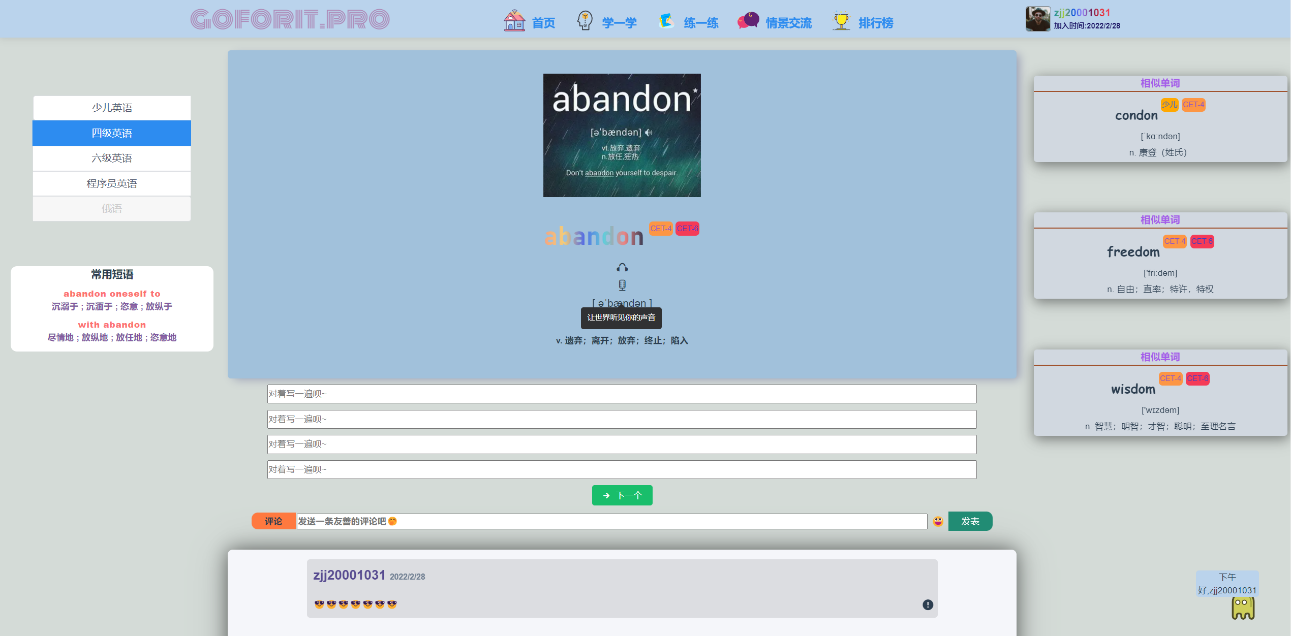
}

}

2.学一学页面

首先验证用户是否已合法登录，如未正常合法登录则会自动跳转至401页面，并强制注销用户如图5.4所示：



图**5.4**认证失败401页面

否则查询用户带学习列表返回用户学习页面如图5.5所示：

图**5.5**　注册成功

后端代码实现:

@PostMapping("/getReciteData")

@HystrixCommand(fallbackMethod = "fallback\_getReciteData")

public String getReciteData(@RequestBody Map map,HttpServletRequest request){

//校验登录信息

String auth = request.getHeader("auth");

String username = (String) map.get("username");

Map parse = JwtUtil.parse(auth);

if (parse.isEmpty()){

return JSON.toJSONString(R.authFail("登陆信息有误"));

}else{

if (!username.equals(parse.get("username"))){

return JSON.toJSONString(R.authFail("登陆信息有误，请勿非法调试控制台"));

}

}

String word = pushService.getWordByIndex(map);

map.put("word",word);

return reciteService.getReciteDataByWord(map);

}

前端代码实现：

getInfos(){

Axios.post("/getReciteData",{

currentIndex:this.currentIndex,

username:LocalStorageUtil.getUsername(),

subject:this.overview.subject

}).then(res=>{

// console.log(res.data);

let code = res.data.code

if (code == 666){

let phraseBoxValue = res.data.data.phrases

let mainBoxValue = res.data.data.reciteMainBoxDto

let similarBoxValue = res.data.data.reciteSimilarBoxDtos

let commentsBoxValue = res.data.data.reciteCommentsBoxDtos

// console.log(similarBoxValue);

//mainbox

this.changeMainBoxValue(mainBoxValue)

//similarBox

this.changeSimilarBoxValue(similarBoxValue)

//commentsBox

this.changeCommentsBoxValue(commentsBoxValue)

//phraseBox

this.changePhraseBoxValue(phraseBoxValue)

}else if (code == 999){

this.overview.subject="四级英语"

this.$notify.error({message:res.data.message,duration:2500})

}else if (code==222){

LocalStorageUtil.resetLocalStorage()

this.$Modal.confirm({

title:'登录信息有误',

content:res.data.message,

onOk(){

location.reload()

},

onCancel(){

location.reload()

}

})

}

}).catch(err=>{

this.$notify.error({message:err.message,duration:1300})

})

}

3.练一练页面：

首先验证用户是否已合法登录，如未正常合法登录则会自动跳转至401页面，并强制注销用户，否则查询用户待学习列表返回用户今日的习题，如图5.6所示：

图**5.6**　管理员登入实现图

后台代码实现：

@PostMapping("/getPractices")

@HystrixCommand(fallbackMethod = "fallback\_getPractices")

public String getPractices(@RequestBody Map map, HttpServletRequest request){

R<PracticeDto> r = null;

String username = ((String) map.get("username"));

String token = request.getHeader("auth");

Map parse = JwtUtil.parse(token);

if (parse.isEmpty()){

r = R.authFail("登录信息有误!");

}else{

if (username.equals(parse.get("username").toString())){

try {

//通过push服务查询单词列表

List<String> words = pushService.getAllWordsBySubject(map);

map.put("words",words);

PracticeDto practices = practiceService.getPractices(map);

//清空答案

for(ChoicePracticechoicePractice: practices.getChoicePracticeList()) {

choicePractice.setRightAnswer(null);

}

for (WritePracticewritePractice :practices.getWritePracticeList()) {

writePractice.setRightAnswer(null);

}

r = R.success("查询题目成功",practices);

}catch (Exception e){

r = R.fail("查询题目失败!");

}

}else{

r = R.authFail("登陆信息有误,请勿非法调试控制台");

}

}

return JSON.toJSONString(r);

}

前台代码：

loadPractices(){

Axios.post("/getPractices",{

username:LocalStorageUtil.getUsername(),

subject:this.overview.subject

}).then(res=>{

// console.log(res.data);

let code = res.data.code

if (code==666){

//开启提交功能

this.$notify.success({message:res.data.message,duration:1200,offset:20})

this.submitAble = true

let choicePracticeList = res.data.data.choicePracticeList

let writePracticeList = res.data.data.writePracticeList

//加载选择题

this.injectChoicePractice(choicePracticeList)

//加载填空题

this.injectWritePractice(writePracticeList)

}else if (code == 999){

//关闭提交功能

this.submitAble = false

this.$notify.error({message:res.data.message,duration:2000})

}else if (code === 222){

LocalStorageUtil.resetLocalStorage()

this.$Modal.confirm({

title:'登录信息有误',

content:res.data.message,

onOk(){

location.reload()

},

onCancel(){

location.reload()

}

})

}

}).catch(err=>{

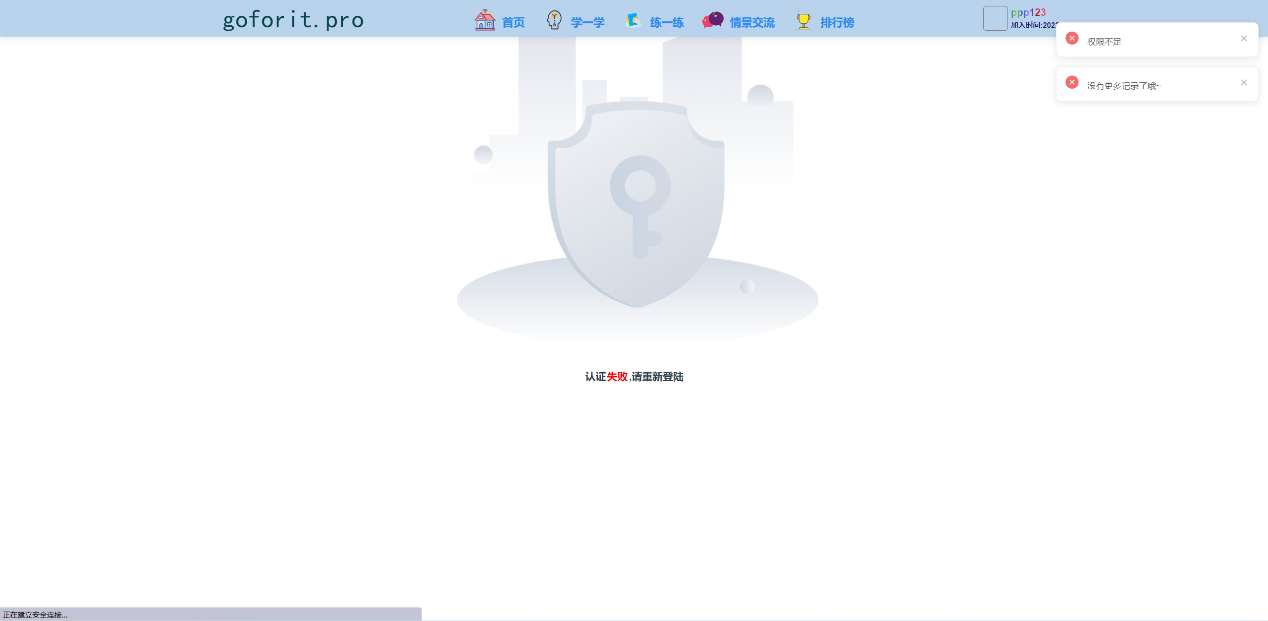
this.submitAble = false

this.$notify.error({message:err.message,duration:1500,offset:20})

}) }

4.管理员后台新增词库

管理员进入后台页面进行词库信息的新增操作。在前台进行路由之前会先发送一个auth请求至后台，如果该用户具有管理员权限则会允许进入并进行相关操作，如若没有此权限则会弹窗提示权限不足禁止访问如图5.7所示，并重定向至401页面 。

图5.7　不具备管理员访问权限

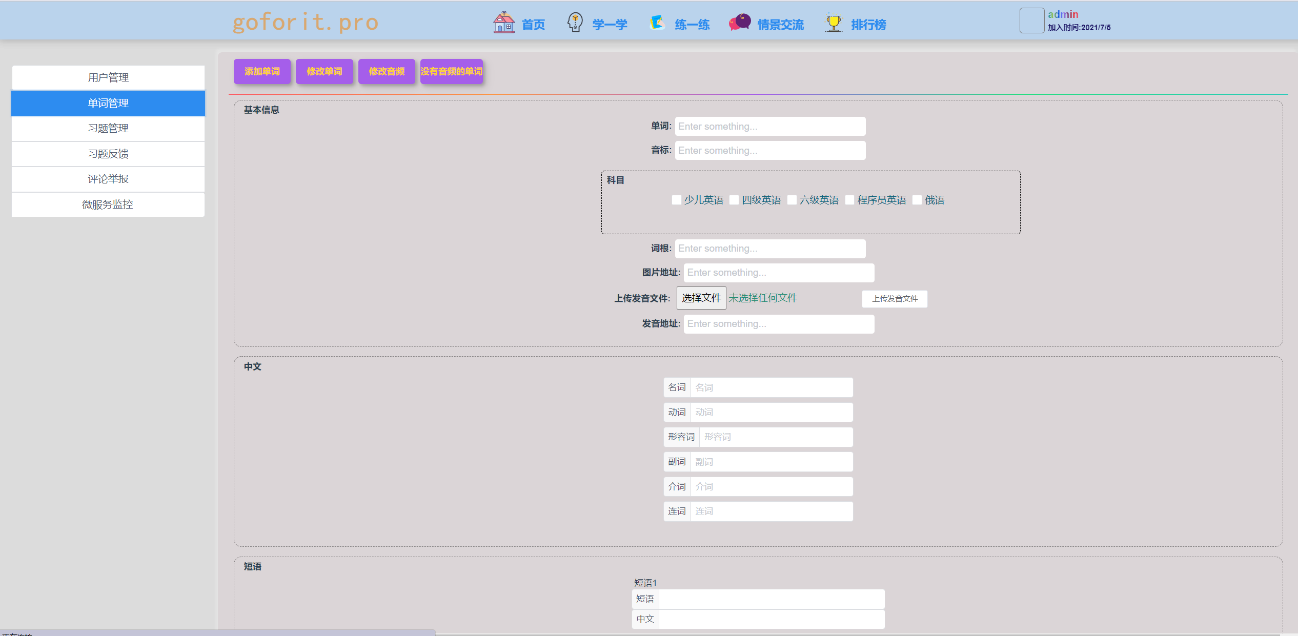
否则进入后台页面如图5.8所示。需要注意的是，此时填写信息默写字段（单词、科目是必填的），否则前台在发送请求之前校验参数是会拦截请求并弹窗显示相应提示信息。

图5.8进入后台词库信息录入界面

前台发送请求代码实现：

addOneWord(){

if (!this.cheackBeforAddOneWord()){return}

Axios.post("/addOneWordAndMeangingsAndPhrases",{

word:this.addBox.baseInfo.word,

symbol:this.addBox.baseInfo.symbol,

subjects:this.addBox.baseInfo.subjects,

root:this.addBox.baseInfo.root,

imgUrl:this.addBox.baseInfo.imgUrl,

soundUrl:this.addBox.baseInfo.soundUrl,

meanings:this.addBox.chineseBox,

phrases:this.addBox.phraseBox

}).then(res=>{

let data = res.data

if (data.code==666){

this.$notify.success({message:data.message,duration:1200,offset:20})

PrintUtil.printSuccess(res.data.message)

this.clearAddBoxValue()

}else if (data.code==222){

this.$notify.warning({message:data.message,duration:1200,offset:20})

PrintUtil.printFail(res.data.message)

}else if (data.code==999){

this.$notify.error({message:data.message,duration:1200,offset:20})

PrintUtil.printFail(res.data.message)

}

}).catch(err=>{

PrintUtil.printFail(err.message)

})

}

后台网关层代码实现：

@PostMapping("/updateWordWordAndMeangingsAndPhrases")

public String updateWordWordAndMeangingsAndPhrases(@RequestBody Map map,HttpServletRequest request){

String auth = request.getHeader("auth");

Map parse = JwtUtil.parse(auth);

R r = null;

if (parse.isEmpty()){

r = R.authFail("登陆信息有误");

}else{

String uploader = ((String) parse.get("username"));

map.put("uploader",uploader);

List<String> logs = adminPageService.updateWordWordAndMeangingsAndPhrases(map);

if (logs.size()!=0){

StringBuilder sb = new StringBuilder();

for (String log : logs) {

sb.append(log+" | ");

}

r = R.fail(sb.toString());

}else{

r = R.success("批量更新单词及其信息成功",null);

}

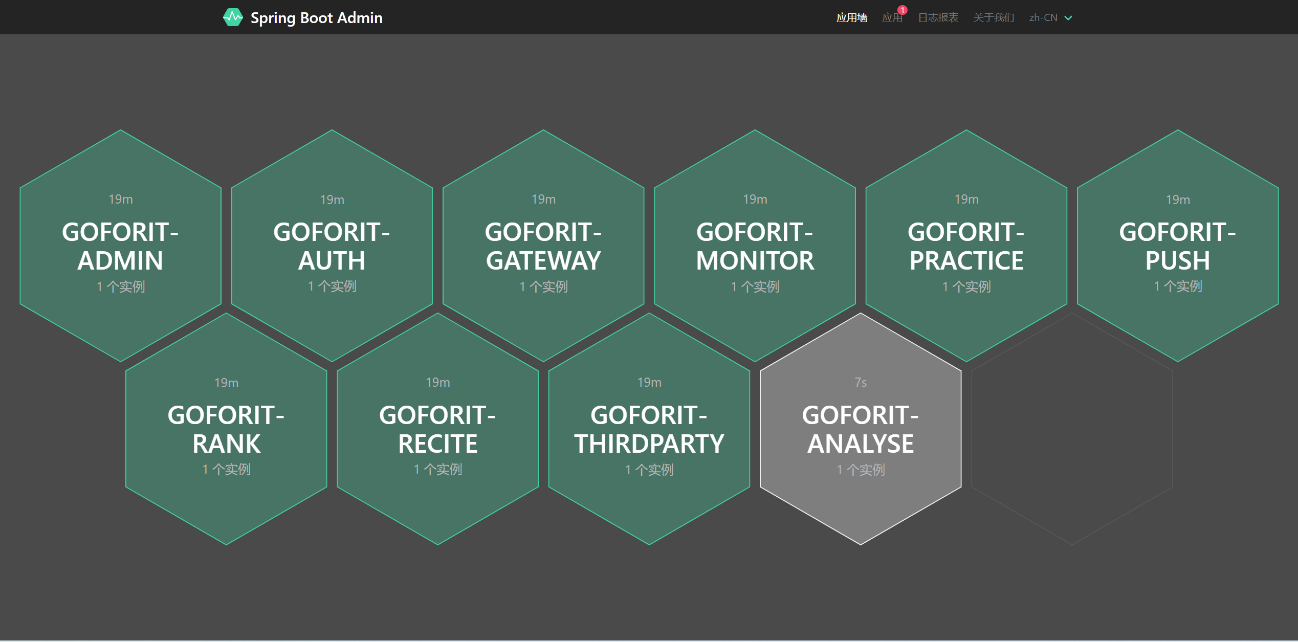
}

return JSON.toJSONString(r);

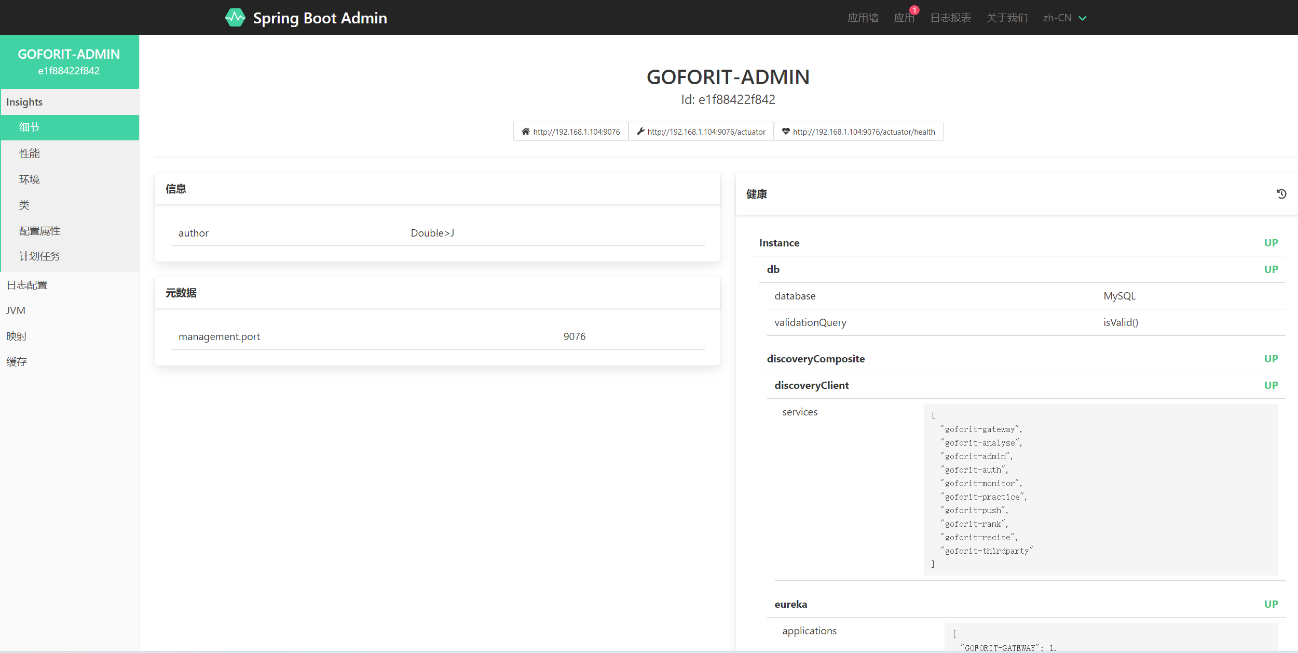
}

5.服务监控

基于spring-boot-starter-actuator监控各微服务，配合springbootadmin进行可视化显示。首页显示的是各微服务的概述运行状态如图5.9所示：

图5.9　微服务运行概述

点击进入此服务的详细信息页面如图5.10所示：

图5.10微服务详细运行情况

6.限流缓存及服务降级的实现

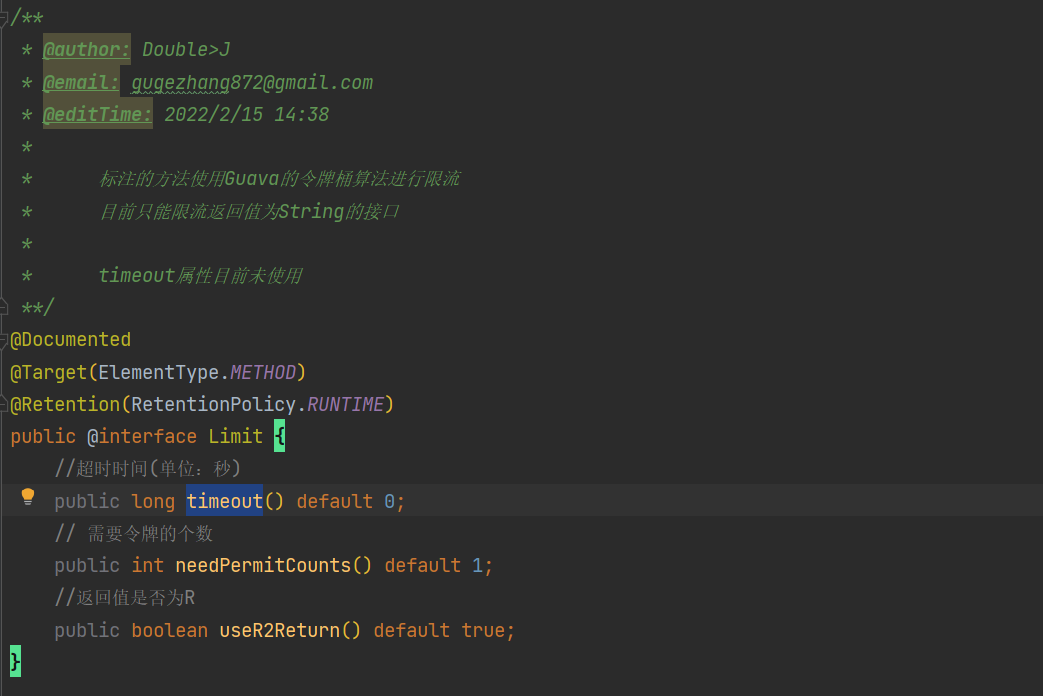
[首先是自定义注解@Limit自带参数如图5.11所示](mailto:首先是自定义注解@Limit自带参数如图5.11所示) ，然后使用AOP在网关层对所有标记了@Limit注解的方法进行统一限流操作，如果触发限流操作则返回限流操作自定义返回语句，否则正常执行业务流程。业务流程中设计到读操作时，会先查询redis中是否已经存在需要查询的信息，存在则直接返回，否则查库然后写入redis后进行返回。如图当前服务访问频率过高或者当前服务暂时不可用，hystrix则起到了作用，会进行服务降级然后为了业务流程的正常流向，此时会自动执行我们事先准备好的fallbackMethod然后然会，至此限流、缓存及降级就是实现了。

图5.11　自定义限流注解

统一限流AOP代码实现：

/\*\*

\* @author: Double>J

\* @email: gugezhang872@gmail.com

\* @editTime: 2022/2/15 14:41

\*\*/

@Aspect

@Component

public class LimitAspect {

//令牌桶每秒生成100个令牌

private static RateLimiter limiter = RateLimiter.create(100);

private static Logger logger = LoggerFactory.getLogger(LimitAspect.class);

@Around("@annotation(pro.goforit.anno.Limit)")

public String limitMethod(ProceedingJoinPointjoinPoint) {

MethodSignature signature = (MethodSignature) joinPoint.getSignature();

Method method = signature.getMethod();

Limit limit = method.getAnnotation(Limit.class);

//需获取令牌数

int permits = limit.needPermitCounts();

//超时默认10s

long timeout = limit.timeout();

//方法的返回值为R类型

boolean useR2Return = limit.useR2Return();

logger.info("permits:{},timeoutL{}s",permits,timeout);

//入参

Object[] args = joinPoint.getArgs();

//booleancanVisit = limiter.tryAcquire(permits, timeout,TimeUnit.SECONDS);

booleancanVisit = limiter.tryAcquire(permits);

if (!canVisit) {

if (!useR2Return){

return "该接口限流中,请稍后重试!";

}else{

return JSON.toJSONString(R.fail("该接口限流中,请稍后重试!"));

}

}

String result = null;

try {

result= (String) joinPoint.proceed(args);

} catch (Throwable throwable) {

result = throwable.getMessage();

}

return result;

}

}

7.统计用户数据实现

PUSH模块启动时会先去查询任务调度日志表，如果今日已经触发过调度任务则不会直接启动调度，反之会出调度任务从mysql、MongoDB、ElasticSearch中拉取数据进行聚合运算，分别计算出今年每一天用户在本站的浏览访问情况、各分类词汇学习进度数据、七天内用户每日作业提交次数统计信息、七天内用户每日作业提交平均正确率、七天内用户在单词学习页面每日发表言论次数、一周内用户在每日一句页面分享发言的次数以及单纯分享自己的好句发音的次数。聚合运算所有用户的数据之后会把这些数据统一格式化之后入库Mysql，方便下一级业务调用，然后在业务与持久层之间加上缓存大大提升请求相应速率以及用户体验。具体的各项数据分析展示效果有本年学习热力图如图5.12所示：

图5.12　今年学习热力图

各分类单词学习进度图如图5.13所示：

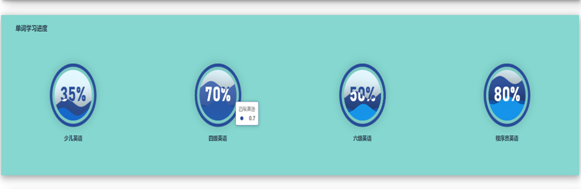


图5.13　各分类词汇学习进度

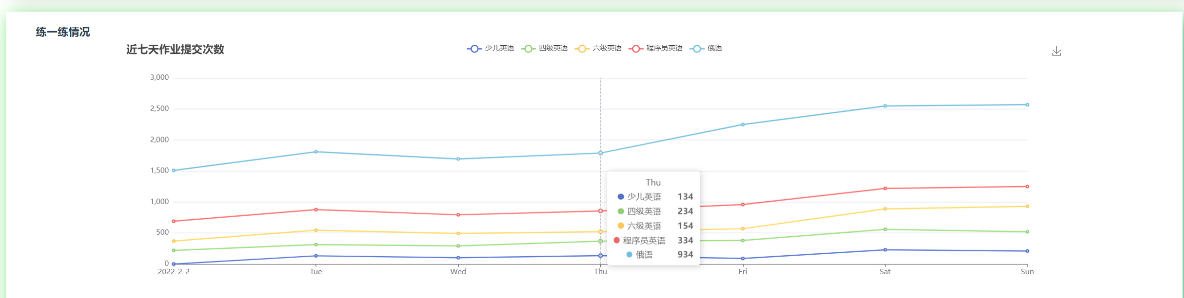
七日内每日作业提交次数如图5.14所示：

图5.14　七日内每日作业提交次数

七日内每日作业平均正确率分为两个图显示，如图5.15、图5.16所示：

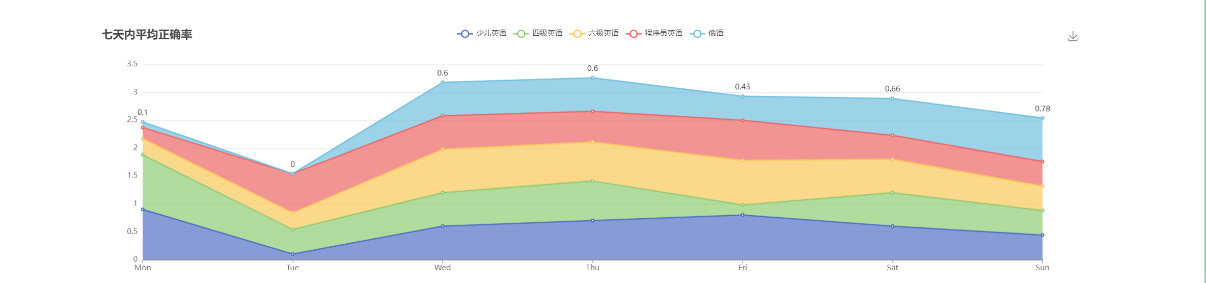
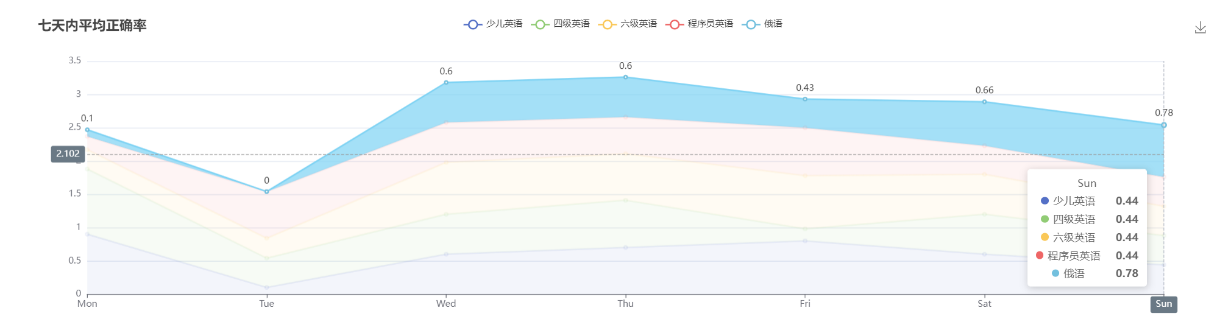


图5.15　七日内每日作业正确率1

图5.16　七日内每日作业正确率2

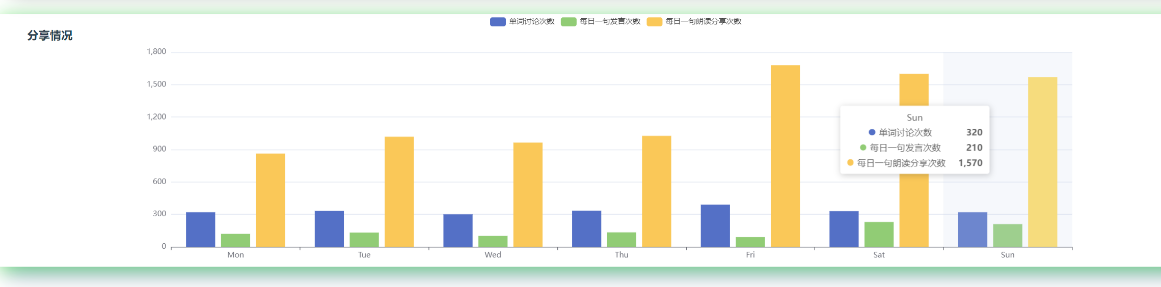
七日内每日分享数据如图5.17所示：

图5.17　七日内每日分享数据

8.单词检索功能实现

本文的单词检索主要是用到了Elasticsearch，由于它具有倒排索引所有非常的时候用来检索数据，设计思路是管理员或开发人员在后台录入词库的时候这份数据不仅会流入Mysql进行持久化，同时它也会流入ElasticSearch进行持久化并对单词进行分词，这样的化所有已入库的单词都没了相应的索引，当用户在页面上双击CTRL的时候就会呼出单词检索框如图5.18所示：

图5.18单词检索框

然后在框中键入联想的单词，即使不输入整个单词后台也能够精准的推送您想要的单词如图5.19所示：

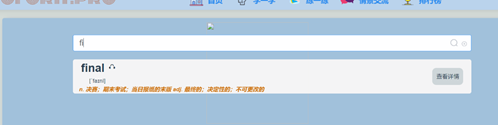


图5.19　单词检索结果

点击单词项里面的耳机按钮即可听见标准发音，点击单词项后的查看详情就可跳转至该单词的详细信息页如图5.20所示，进行针对性学习。

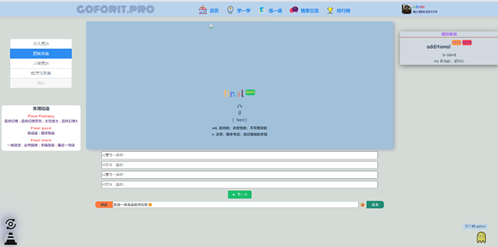
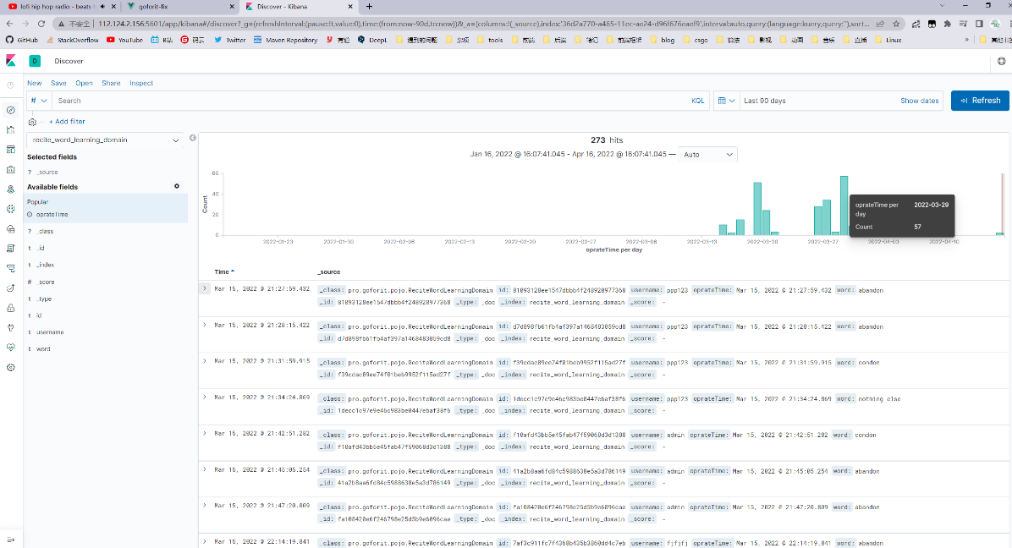


图5.20　单词检索跳转详情页

9.搭建可视化平台

由于本项目使用到了ElasticSearch以及消息队列RabbitMQ，那么对他们的运行状况的监控以及一些内部数据的可视化工作就显得很有必要了，对于ElasticSearch的可视化这边没有时用到ElasticSearch-head这一传统的工具而是ElasticStack技术栈的Kibana对各日志信息进行可视化展示如图5.21所示：

图 5.21　Kibana日志可视化

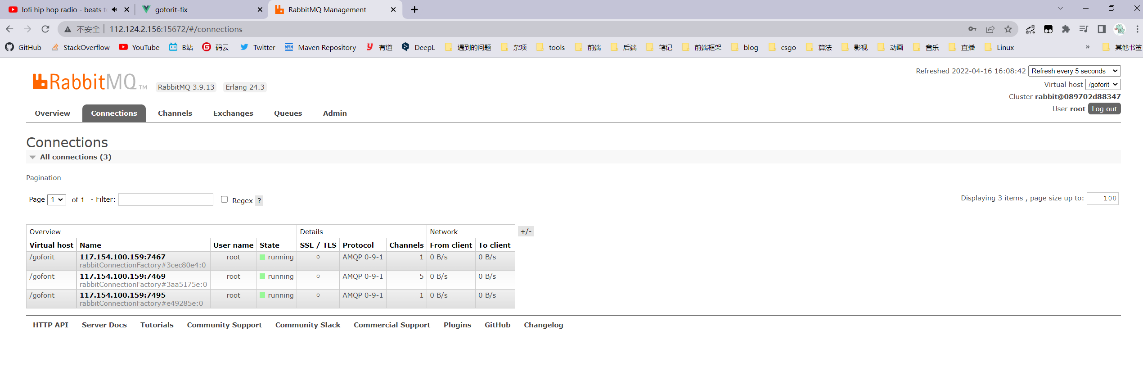
对于RabbitMq我也使用了它自带的web页面进行监控如图5.22所示：

图5.22　RabbitMQ监控

## 5.2　数据库实现

Mysql数据库数据展示：

1.User表：用于存放用户信息如图5.23所示：

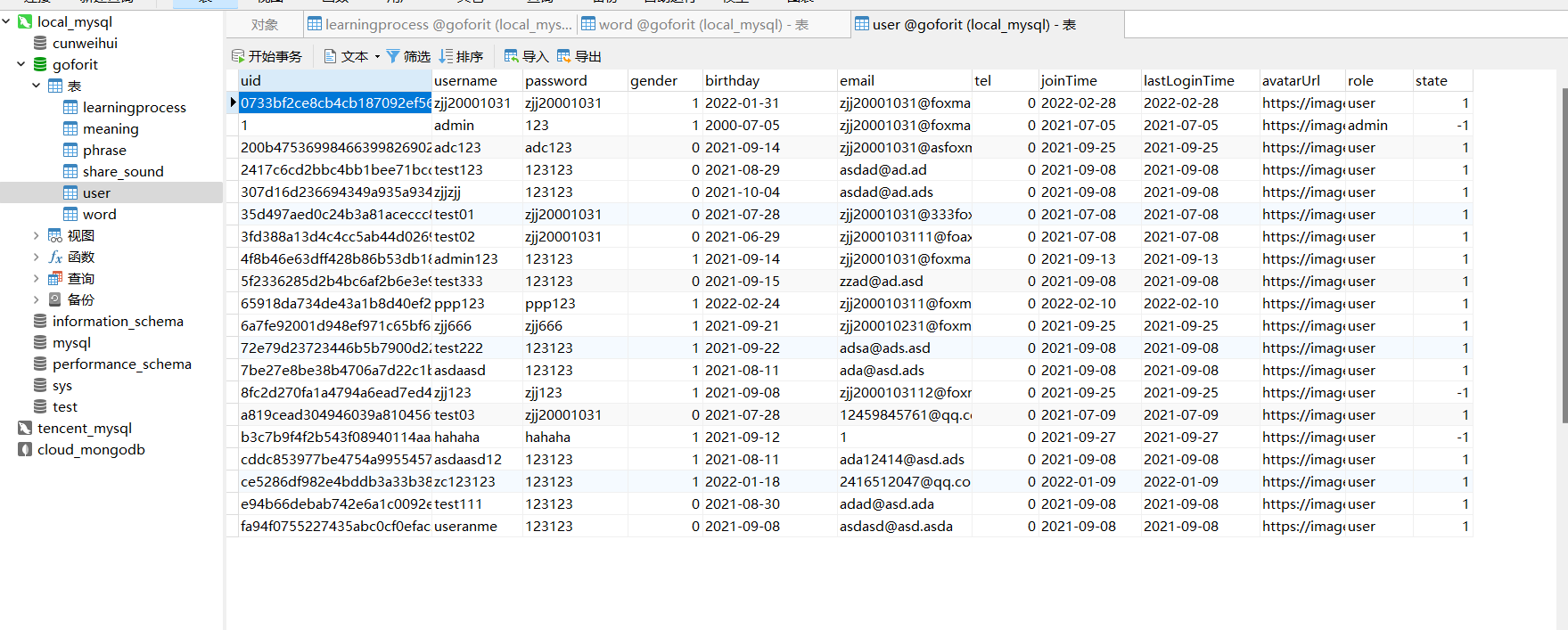


图5.23　User表数据展示

2.LearningProcess表：用于持久化用户的学习进度等如图5.24所示：

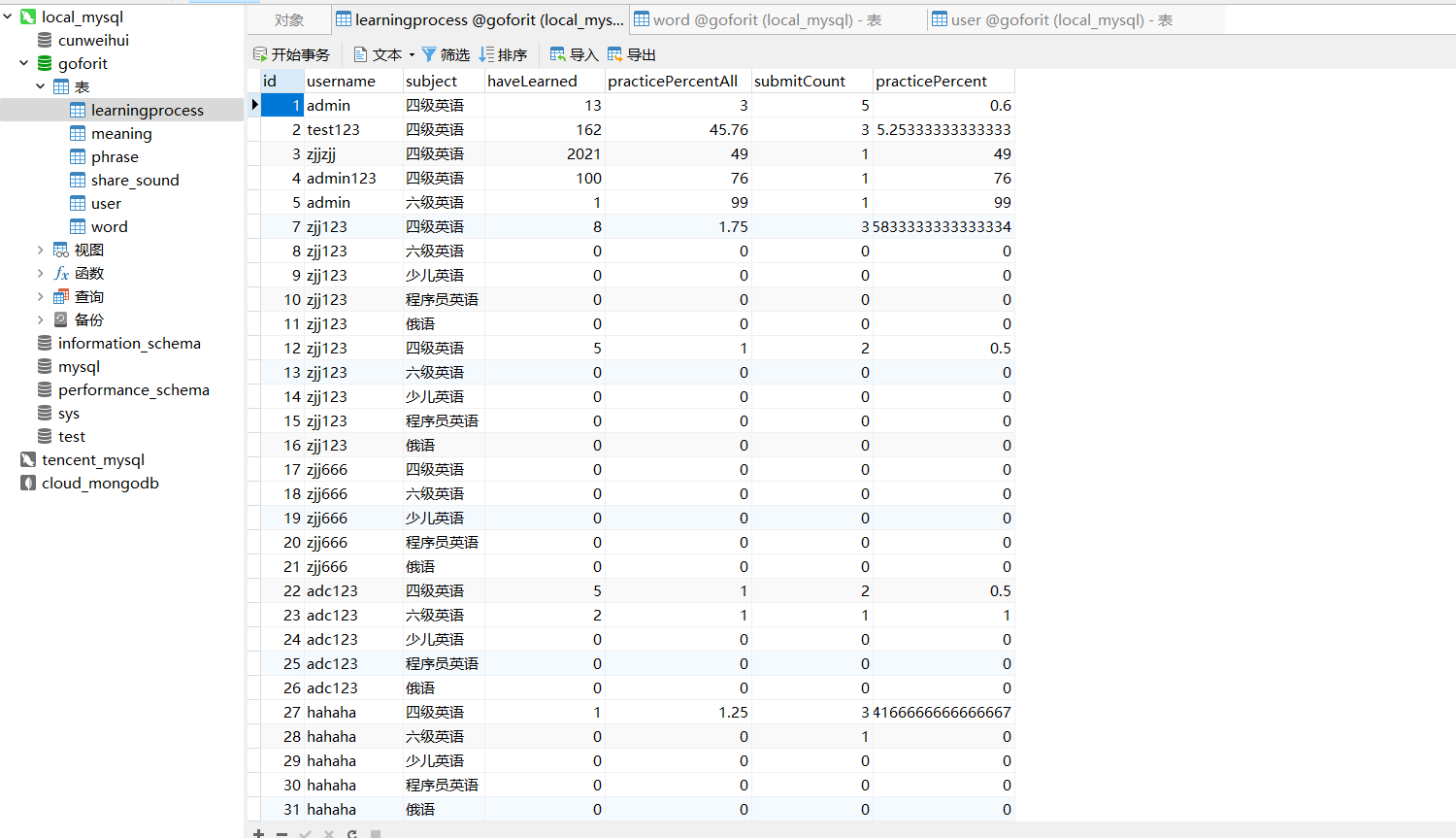


图5.24LearningProcess表数据展示

3.Phraes表：用户存放单词相关的短语及基本信息如图5.25所示：

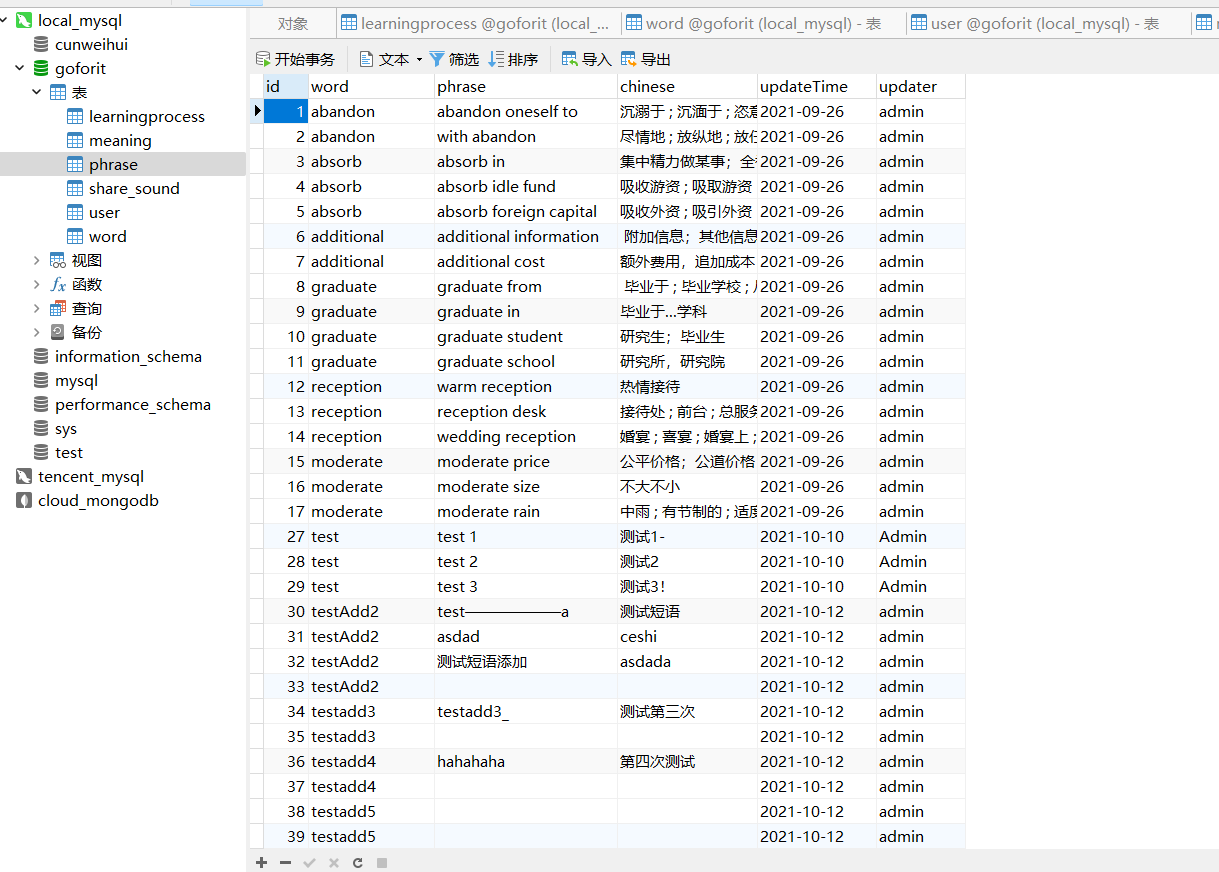


图5.25　Phrase表数据展示

4.Word表：词库。用来存放单词的基本信息，主要包括：单词、音标、图片地址、标准发音地址、科目类别、词根等，如图5.26所示：

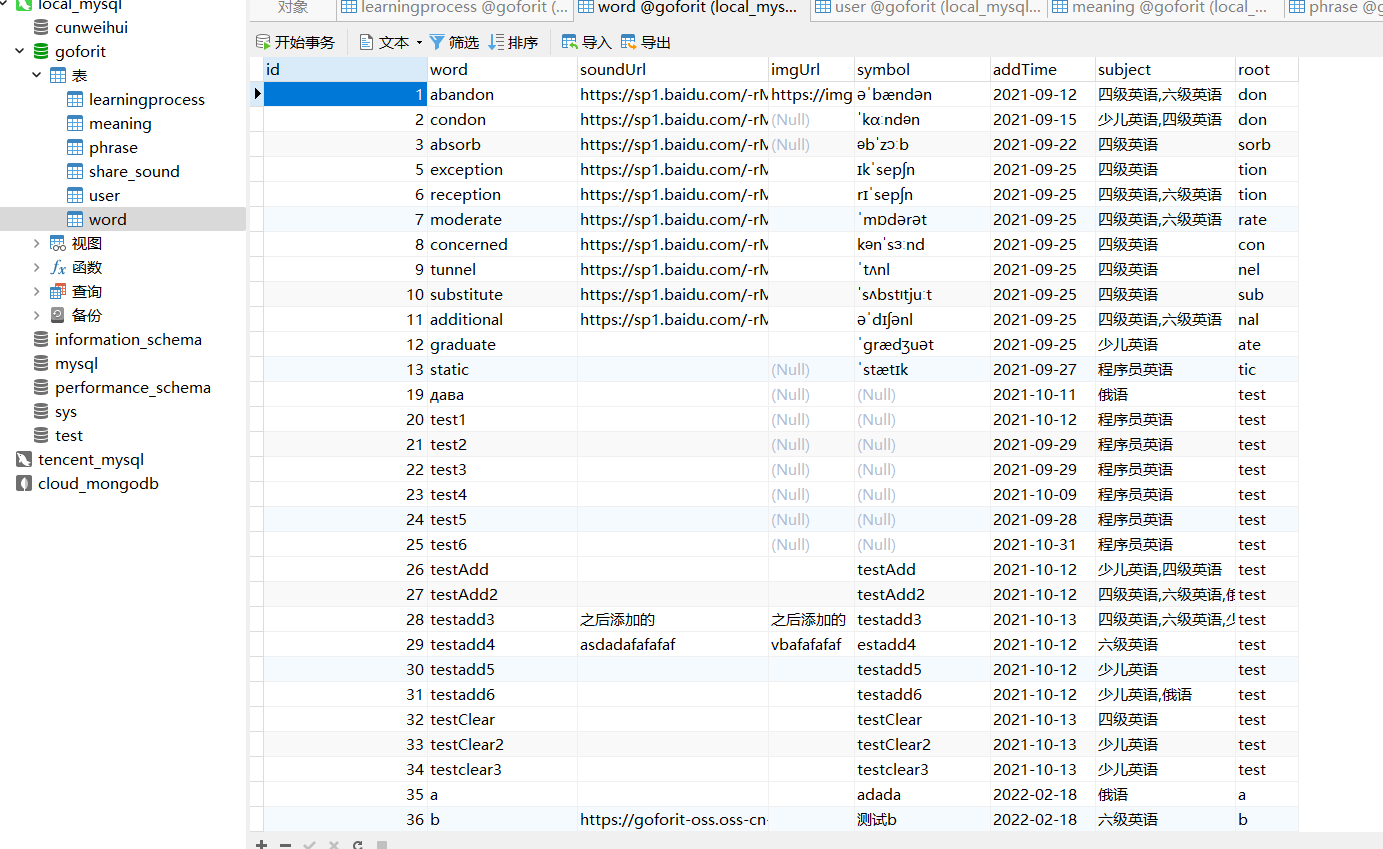


图5.26　Word表数据展示

5.share\_sound表：用于存放用户上传的发音的单词以及发音文件的映射如图5.27所示：

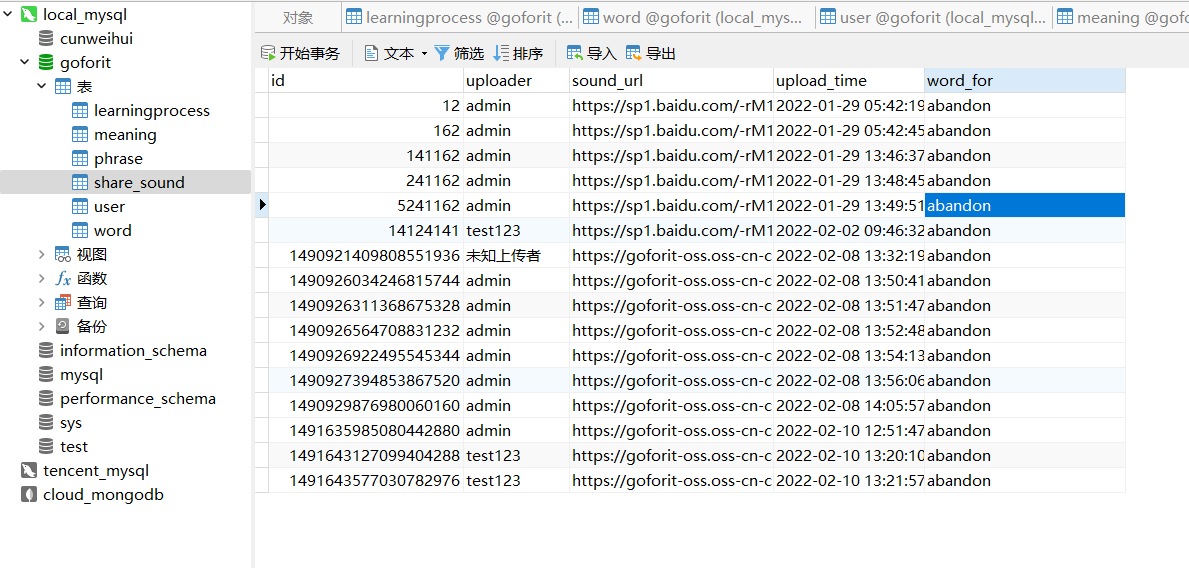


图5.27　Phrase表数据展示

# 6　系统测试与调试

## 6.1　测试目的

本次测试主体分为两个流程，首先是对普通用户拥有相关功能进行测试，其次是对开发者及管理员相关功能接口进行测试。对普通用户进行测试内容主要包括登陆注册、正常单词学习模块的查询功能、习题操作功能、学习情况分析模块、排行榜模块、评论及反馈模块等。管理员接口功能测试主要包含对用户操作模块、单词操作模块、习题操作模块、监控模块、反馈模块的测试。

## 6.2　测试概述

拥有不同接口操作权限的用户在访问特定接口时，如果拥有访问权限则正常返回查询值，否则返回相应的提示信息，通过ElementUI的Notify组件进行提示。

## 6.3　单元测试

### 6.3.1　用户注册测试

用户注册需要填写的基本信息有：

1. 用户名:长度不低于5
2. 密码：长度不低于8
3. 重复密码：与密码输入一致
4. 邮箱：合法邮箱且没有账号使用此邮箱注册
5. 验证码：6位随机字符
6. 性别
7. 生日

具体测试用例见下表表6.1所示。

表**6.1**　用户注册测试用例表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 优先 等级 | 测试模块 | 测试项 | 预置  条件 | 测试步骤 | 预期结果 | 测试结果 |
| 1 | P0 | 注册 | 注册账号 | 进入注册页面 | 全部安全要求正确填写 | 注册成功，跳转到登录界面 | PASS |
| 2 | P0 | 注册 | 注册账号 | 进入注册页面 | 用户名长度低于五，其他项正确填写 | 注册失败，弹出弹窗：用户名需要大于五 | PASS |
| 3 | P0 | 注册 | 注册账号 | 进入注册页面 | 密码长度低于8，其他项正确填写 | 注册失败，弹出弹窗：密码长度需要大于8 | PASS |
| 4 | P0 | 注册 | 注册账号 | 进入注册页面 | 重复密码与初始密码不一样，其他项正确填写 | 注册失败，弹出弹窗：重复密码需与密码一致 | PASS |
| 5 | P0 | 注册 | 注册账号 | 进入注册页面 | 邮箱不可用/已注册，其他项正确填写 | 注册失败，弹出弹窗：此邮箱不可用/已注册 | PASS |
| 6 | P0 | 注册 | 注册账号 | 进入注册页面 | 验证码错误填写 | 注册失败，弹出弹窗：验证码错误 | PASS |
| 7 | P0 | 注册 | 注册账号 | 进入注册页面 | 不填写性别，其他项正确填写 | 注册失败，弹出弹窗：请填写必填项 | PASS |
| 8 | P0 | 注册 | 注册账号 | 进入注册页面 | 不填写生日，其他项正确填写 | 注册失败，弹出弹窗：请填写必填项 | PASS |

### 6.3.2　单词发音音频分享测试

单词发音分享模块需要测试的问题有：登陆超时用户是否可上传、上传超大文件的解决方式、上传不合规范的文件的反馈。

上传规则：

1. 文件格式为wav或mp3
2. 文件大小不得大于2mb

表**6.2**　单词发音音频分享测试用例表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 优先 等级 | 测试模块 | 测试项 | 预置条件 | 测试步骤 | 预期结果 | 测试结果 |
| 1 | P0 | 选修课报名模块 | 单词发音音频分享 | 登录超时 | 上传正确格式与大小文件 | 上传失败并强制注销用户，休眠3s后跳转至401页面 | PASS |
| 2 | P0 | 选修课报名模块 | 单词发音音频分享 | 登录成功 | 上传3mb的wav/MP3文件 | 上传失败，弹窗提示：文件过大 | PASS |
| 3 | P0 | 选修课报名模块 | 单词发音音频分享 | 登录成功 | 上传1mb的MP4文件 | 上传失败，弹窗提示：文件格式不正确，不允许上传 | PASS |
| 4 | P0 | 选修课报名模块 | 单词发音音频分享 | 登录成功 | 上传1mb的MP3/wav文件 | 上传成功 | PASS |

### 6.3.3　单词检索功能测试

单词检索功能需要测试的问题有：双击CTRL是否能正确呼出检索框，检索出的单词是否正确，检索结果中点击查看详情是否能正确跳转至相应单词学习页面。

表**6.3**　单词检索用例表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 优先 等级 | 测试模块 | 测试项 | 预置条件 | 测试步骤 | 预期结果 | 测试结果 |
| 1 | P0 | 单词检索 | 单词检索弹出 | 登录成功 | 除首页外任意界面双击ctrl | 单词检索框成功跳出 | PASS |
| 2 | P0 | 单词检索 | 单词检索弹出 | 登录成功 | 首页双击ctrl | 无弹窗显示 | PASS |
| 3 | P0 | 单词检索 | 检索  单词 | 词库存在单词 | 输入完整单词 | 检索成功 | PASS |
| 4 | P0 | 单词检索 | 检索  单词 | 词库存在单词 | 输入首字母 | 弹出五个该字母相同的单词 | PASS |
| 5 | P0 | 单词检索 | 检索  单词 | 词库不存在单词 | 输入完整单词 | 搜索不到该单词 | PASS |
| 6 | P0 | 单词检索 | 查看  详情 | 单词检索成功 | 点击耳机标识 | 准确发出该单词读音 | PASS |
| 7 | P0 | 单词检索 | 查看  详情 | 单词检索成功 | 点击查看详情 | 跳转到单词学习界面 | PASS |

### 6.3.4　单词管理测试

词库管理模块需要测试的问题有：是否能够拦截不具备开发者权限的用户的能力，错误输入或者漏输信息反馈信息是否合理，成功添加之后个输入框是否能够成功情况原有输入等。

表**6.4**　单词管理测试用例表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 优先 等级 | 测试  模块 | 测试项 | 预置条件 | 测试  步骤 | 预期结果 | 测试  结果 |
| 1 | P0 | 单词  管理 | 添加单词 | 管理员登录成功，进入单词管理界面 | 按要求填写好所有内容 | 单词添加成功 | PASS |
| 2 | P0 | 单词  管理 | 添加单词 | 管理员登录成功，进入单词管理界面 | 不填写单词，其他按要求填写 | 添加失败，弹窗提示：至少把单词的基本信息填完呗！ | PASS |
| 3 | P0 | 单词  管理 | 添加单词 | 管理员登录成功，进入单词管理界面 | 不选择科目，其他按要求填写 | 添加失败，弹窗提示：至少把单词的基本信息填完呗！ | PASS |
| 4 | P0 | 单词  管理 | 添加单词 | 管理员登录成功，进入单词管理界面 | 只填写单词和科目选择 | 单词添加成功 | PASS |

### 6.3.5　习题管理测试

习题管理模块需要测试的问题有：错误填写信息或者漏填必填参数时反馈信息是否合理，成功添加题目信息后输入框中原始输入信息是否能够及时正确地被清空。

表**6.5**　习题管理测试用例表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 优先 等级 | 测试模块 | 测试项 | 预置条件 | 测试步骤 | 预期结果 | 测试结果 |
| 1 | P0 | 习题管理 | 添加习题 | 管理员登录成功，进入习题管理界面 | 添加题目，填写考察项，设置好选项与答案，点击提交 | 习题添加成功 | PASS |
| 2 | P0 | 习题管理 | 添加习题 | 管理员登录成功，进入习题管理界面 | 不填写题目，其他按要求填写 | 添加失败，弹窗提示：请填写完整信息 | PASS |
| 3 | P0 | 习题管理 | 添加习题 | 管理员登录成功，进入习题管理界面 | 不填写考察单词，其他按要求填写 | 添加失败，弹窗提示：请填写完整信息 | PASS |
| 4 | P0 | 习题管理 | 添加习题 | 管理员登录成功，进入习题管理界面 | 选项不填写完整，其他按要求填写 | 添加失败，弹窗提示：请填写完整信息 | PASS |
| 4 | P0 | 习题管理 | 添加习题 | 管理员登录成功，进入习题管理界面 | 不填写答案，其他按要求填写 | 添加失败，弹窗提示：请填写完整信息 | PASS |

### 6.3.6　用户管理测试

用户管理需要测试的问题有：通过用户uuid以及用户名是否能够正确查询出各个用户的信息，管理员封禁用户之后该用户是否不被允许登录等。

表**6.6**　用户管理测试用例表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 优先 等级 | 测试  模块 | 测试项 | 预置条件 | 测试步骤 | 预期结果 | 测试结果 |
| 1 | P0 | 管理员管理 | 用户  管理 | 管理员登录成功，进入用户管理界面，用户存在 | 输入UID查找用户 | 查找成功 | PASS |
| 2 | P0 | 管理员管理 | 用户  管理 | 管理员登录成功，进入用户管理界面，用户存在 | 输入用户名精确查找用户 | 查找成功 | PASS |
| 3 | P0 | 管理员管理 | 用户  管理 | 查找用户成功 | 查看其单词学习进度 | 查看成功 | PASS |
| 4 | P0 | 管理员管理 | 用户  管理 | 查找用户成功 | 点击模式切换 | 切换成功 | PASS |
| 5 | P0 | 管理员管理 | 用户  管理 | 查找用户成功 | 查看其练习进度 | 查看成功 | PASS |
| 6 | P0 | 管理员管理 | 用户  管理 | 查找用户成功 | 点击封禁该用户 | 封禁成功 | PASS |
| 7 | P0 | 管理员管理 | 用户  管理 | 查找用户成功，该用户被封禁 | 点击解除封禁 | 解除封禁成功 | PASS |

## 6.4　集成测试

对于本系统，目前存在普通用户、开发者及管理员三种角色，而开发者和管理员具有相同的权限，下文统称管理员。而管理员除了具有对词库等信息的增删改操作外还拥有普通用户所拥有的所用操作权限。

首先打包好前端及后端项目后运行起来，打开其他非代码的所有服务（Redis、Mysql、MongoDB、ElasticSerach、RabbitMQ等）。

分别登录拥有不同角色的账号，在系统中进行各类操作，确定权限之间的隔离生效，保证了不具备访问权限的用户访问不到特定内容。

对核心的接口进行压力测试，观察到限流及服务降级流程正常触发，相应的缓存日志也正常输出，说明本站的限流缓存降级策略生效。

邀请非开发人员对页面布局进行点评，结论色调基本一致，比较和谐。

# 7　总结与展望

## 7.1　总结

开发过程跌宕起伏，有完成小模块开发后的欢呼，也有遇棘手Bug后想砸屏幕的愤怒，以及完成阶段性开发后的放松。总体来说，本次设计还是让我获益匪浅的，尤其是以下几点心得：

1.软件的开发不是一蹴而就的，一个程序的成长是需要长久的迭代的。

2.好的架构并不一定是花里胡哨的，合适的才是最好的。

3.遇到问题第一时间考虑的事情不应该询问他人，而应该是独立解决，实在解决不了了再去向他人请教。

4.使用Linux时，尤其是使用Docker的时候一定记得不要指定数据的存放路径为/tmp。

5.当开发到一定进度的时候记得备份数据并且备份或托管代码。

6.做好开发环境以及生产环境的数据隔离。

7.做好空值的处理方式，控制很有可能会影响到整个调用链。

8.开发过程中写好注释，不然几个月后真的就不认识了。

9.不要害怕控制台和日志打印出来的错误信息，正确使用这些信息能够更快的帮助我们定位错误解决问题。

10.不熟悉的三方开源组件不要单纯看几篇别人分享的博客就上手直接怼到生产环境，因为在没熟悉这些组件的情况下直接使用，遇到问题很难自己解决。毕竟别人也不是单纯为你开发的，调研工作是一定得做到位的。

### 7.2　展望

本系统是存在日志分析功能的，但是并没有引入相应的Hadoop生态进行真实的离线批计算统计操作，而是使用的RabbitMq以及Elasticsearch进行数据的传输与存储并定时进行处理，小数据量此套方案是勉强可行的，一旦数据量暴增，很可能后台分析就会内存溢出导致宕机，进而影响整体服务的运行。

常见的解决方案应该是，所有服务提供kafka的接口传输信息或者是ELK对日志进行操作，然后相应模块对数据进行统一整合然后对数据进行清洗操作，然后定时调度计算集群对数据进行分析之后，把计算结果持久化并缓存，方便其他相关依赖业务的调用，极大减少了调度延时以及保证了服务的可用性。

在不久的将来我会逐步学习掌握大数据的相关技术栈，通过正统且专业的方式来解决海量数据的多维度分析与处理工作，并且标准化可视化操作流程让分析过程不再像现在这样繁琐耦合低效。

参考文献

[1] 柳伟卫.Spring Boot 企业级应用开发实战[M].北京大学出版社.2018.12-15.

[2] 孙卫琴.Tomcat与Java Web开发技术详解（第2版）[M].电子工业出版社.2009.17-20.

[3] 陈雄华.Spring企业级应用开发详解[M].电子工业出版社.2009.21-25.

[4] 毕建信.基于MVC设计模式WEB应用研究与实现[D].武汉理工大学.2006.25-30.

[5] 舒礼莲.基于SpringMVC的Web应用开发[J].计算机与现代化.2013.31-35.

[6] 未来科技.Bootstrap实战从入门到精通[M].中国水利水电出版社.2017.40-45.

[7] 施尧.jQuery EasyUI从零开始学[M].清华大学出版社.2018.46-50.

[8]杨世文，孙会军.Maven应用实战[M].清华大学出版社.2018.51-55.

[9]James Goodwill.Pure Java Server Pages .Indianapolis Ind: Sams, 2000.60-70.

[10] John W. Satzinger， Robert B. Jackson， Stephen D. Burd.系统分析与设计[J].机械工业出版社.2002.(10):79-82.