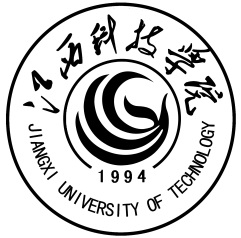
**密级：**

**学号：**

**本科生毕业论文（设计）**

**基于SpringCloud微服务架构的游戏社区的设计与实现**

**学 院： 信息工程学院**

**专 业： 计算机科学与技术**

**班 级： 18本计算机1班**

**学 号： 2018004214 姓 名： 肖海南**

**指导教师1： 金欢 指导教师2：**

**学士学位论文原创性声明**

本人郑重声明：所呈交的论文（设计）是本人在指导老师的指导下独立进行研究，所取得的研究成果。除了文中特别加以标注引用的内容外，本论文（设计）不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式表明。本人完全意识到本声明的法律后果由本人承担。

学位论文作者签名（手写）： 签字日期： 年 月 日

**学位论文版权使用授权书**

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权**江西科技学院**可以将本论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

|  |  |
| --- | --- |
| 本学位论文属于 | 保 密 □， 在 年解密后适用本授权书。 |
| 不保密 □。 |

（请在以上相应方框内打“√” ）

学位论文作者签名（手写）： 指导老师签名（手写）：

签字日期： 年 月 日 签字日期： 年 月 日

**摘 要**

**摘要是对全文的梗概，要能够简明、确切地记述论文的重要内容的短文。**

**可以从系统的背景、来源，所用技术、具体功能、价值意义等方面去提炼论文的梗概。**

**撰写建议：**

摘要应该主要从三个方面去写：

**（1）简单阐述项目的背景意义。**（**要求：简明扼要**）

也就是你的第1段可以简单一点写。

**（2）阐述项目的主要功能或者能提供的服务**（**要求：稍微具体一点，这是摘要重点部分**）

也就是描述一下你的英语主观题自动评分系统具体提供了哪些服务功能？

**（3）简单阐述实现系统的相关技术及工具及意义。**（**要求：简明扼要**）

阐述一下你的系统涉及的相关技术和工具及意义。

**全文控制在500字左右即可。**

据统计，在2021年中国的游戏市场实际销售收入超过2900亿元，游戏用户群体庞大，

系统的主要核心就是社区，在系统中每一个游戏都有对应的一个社区，当然也有非游戏的社区，对于一个普通用户来说，可以游览各个社区当中的内容，以及文章的资讯，游戏的详情，本系统还支持对游戏继续评分和评论，当然用户也可以发表文章，在其他的文章下面进行评论，对于热门的文章，会在首页显示，在首页还有3大游戏排行榜，热门的4个经典大作和4个最近发布的游戏，还有一些热门的游戏动态也会在首页显示，首页还会有轮播图，以及一些游戏的显示，本系统还有登录注册功能，用户第一次用手机验证登录既可注册，登录

本系统主要的实现是Java EE开发相关的技术，涉及到了一些开发框架，主要使用SpringBoot，这可以快速开发，ORM框架mybatis，mybatis-plus,

**关键字： XXX；YYY；ZZZZ（关键词一般3~5个）**

**Abstract**

**Key Words :** XXX ; YYY ; ZZZZ（对应中文关键词）

目 录

[第1章 引言/绪论 1](#_Toc96897424)

[1.1 概述 1](#_Toc96897425)

[1.2 国内外研究现状 1](#_Toc96897426)

[1.2.1 国外研究现状 1](#_Toc96897427)

[1.2.2 国内研究现状 1](#_Toc96897428)

[1.3 研究目的与意义 1](#_Toc96897429)

[第2章 系统开发技术及工具简介 3](#_Toc96897430)

[2.1 系统开发技术简介 3](#_Toc96897431)

[2.1.1 ElementUI,Ant Design以及less技术简介 3](#_Toc96897432)

[2.1.2 Vue.js及组件技术简介 3](#_Toc96897433)

[2.1.4 Redis技术简介 4](#_Toc96897434)

[2.1.5 Docker技术简介 4](#_Toc96897435)

[2.1.6 Elasticsearch技术简介 5](#_Toc96897436)

[2.1.7 RabbitMQ技术简介 6](#_Toc96897437)

[2.1.8 云服务技术简介 6](#_Toc96897438)

[2.1.9 MySQL技术简介 6](#_Toc96897439)

[2.1.10 Mybatis技术简介 6](#_Toc96897440)

[2.1.11 Nginx技术简介 6](#_Toc96897441)

[2.1.12 Git技术简介 7](#_Toc96897442)

[2.1.13 Maven技术简介 7](#_Toc96897443)

[2.1.14 SpringBoot技术简介 7](#_Toc96897444)

[2.1.15 SpringCloud相关组件技术简介 7](#_Toc96897445)

[2.2 系统开发工具简介 9](#_Toc96897446)

[2.2.1 IDEA工具简介 9](#_Toc96897447)

[2.2.2 Xshell工具简介 9](#_Toc96897448)

[2.2.3 webStrom工具简介 10](#_Toc96897449)

[2.2.4 Navicat Premium工具简介 11](#_Toc96897450)

[2.2.5 Postman工具简介 11](#_Toc96897451)

[第3章 系统需求分析 13](#_Toc96897452)

[3.1 系统可行性分析 13](#_Toc96897453)

[3.1.1 经济可行性分析 13](#_Toc96897454)

[3.1.2 技术可行性分析 13](#_Toc96897455)

[3.2 系统功能需求分析 14](#_Toc96897456)

[3.2.1 功能需求分析 14](#_Toc96897457)

[3.2.2 系统用例分析 14](#_Toc96897458)

[3.3 系统性能需求分析 14](#_Toc96897459)

[第4章 系统概要设计 15](#_Toc96897460)

[4.1 系统功能模块设计 15](#_Toc96897461)

[4.2 数据库设计 15](#_Toc96897462)

[4.2.1 数据库概要设计 15](#_Toc96897463)

[4.2.2 数据库逻辑设计 15](#_Toc96897464)

[第5章 系统详细设计与实现 17](#_Toc96897465)

[5.1　用户模块的设计与实现 17](#_Toc96897466)

[5.1.1　登录注册功能的实现 17](#_Toc96897467)

[5.1.2　找回密码功能的实现 19](#_Toc96897468)

[5.1.3　修改个人资料功能的实现 20](#_Toc96897469)

[5.1.4　收藏夹功能的实现 21](#_Toc96897470)

[5.1.5　历史记录功能的实现 22](#_Toc96897471)

[5.1.6　被点赞消息提醒的实现 23](#_Toc96897472)

[5.1.7　查看点赞消息的实现 24](#_Toc96897473)

[5.1.8　文章发表的实现 25](#_Toc96897474)

[5.1.9　点赞文章的实现 26](#_Toc96897475)

[5.1.10　文章收藏的实现 27](#_Toc96897476)

[5.1.11　用户发表的文章功能的实现 28](#_Toc96897477)

[5.1.12　全文检索功能的实现 29](#_Toc96897478)

[5.1.13　排行榜功能的实现 30](#_Toc96897479)

[5.1.14　游戏评论功能的实现 31](#_Toc96897480)

[5.1.15　文章评论功能的实现 32](#_Toc96897481)

[5.1.16　百科一节菜单功能的实现 33](#_Toc96897482)

[5.1.17　百科详细内容功能的实现 34](#_Toc96897483)

[5.2 管理模块的设计与实现 34](#_Toc96897484)

[5.2.1　管理用户登录功能的实现 34](#_Toc96897485)

[5.2.2　首页轮播图管理功能的实现 35](#_Toc96897486)

[5.2.3　邮件推送功能的实现 36](#_Toc96897487)

[5.2.4　游戏管理功能的实现 37](#_Toc96897488)

[5.2.5　排行榜管理功能的实现 38](#_Toc96897489)

[5.2.6　游戏轮播图管理功能的实现 38](#_Toc96897490)

[5.2.7　游戏评分功能的实现 39](#_Toc96897491)

[5.2.8　游戏标签功能的实现 40](#_Toc96897492)

[5.2.9　推荐游戏管理功能的实现 41](#_Toc96897493)

[5.2.10　设置游戏优惠功能的实现 42](#_Toc96897494)

[5.2.11　游戏首页推荐管理功能的实现 43](#_Toc96897495)

[5.2.12　游戏产商管理功能的实现 44](#_Toc96897496)

[5.2.13　游戏类型管理功能的实现 44](#_Toc96897497)

[5.2.14　社区管理功能的实现 45](#_Toc96897498)

[5.2.15　文章管理功能的实现 46](#_Toc96897499)

[5.2.16　用户管理功能的实现 47](#_Toc96897500)

[5.3 运维模块的设计与实现 48](#_Toc96897501)

[5.3.1　elasticsearch管理功能的实现 48](#_Toc96897502)

[5.3.2　nacos管理功能的实现 48](#_Toc96897503)

[5.3.3　sentinel管理功能的实现 49](#_Toc96897504)

[第6章 系统总结与展望 50](#_Toc96897505)

[6.1 系统总结 50](#_Toc96897506)

[6.2 展望 50](#_Toc96897507)

[参考文献 51](#_Toc96897508)

[致 谢 52](#_Toc96897509)

# 第1章 引言/绪论

## 1.1 概述

## 1.2 国内外研究现状

### 1.2.1 国外研究现状

### 1.2.2 国内研究现状

## 1.3 研究目的与意义

从20世纪的电子游戏出现开始，游戏经过不断发展，近年来，全球游戏市场规模逐渐扩大，2019年达到1457亿美元，全球游戏市场收入将在2020年达到1593亿美元，同比增长9.33%。全球游戏玩家数量也将持续增长，据Newzoo预测，到2023年全球游戏玩家将超过三十亿。电子游戏自发明到现在，已经极大地改变了人类的生活环境和生活。它已不仅是一个产业，更发展成为一门艺术。在未来，游戏将更加深入地渗透到人们的生活中去，成为人们生活中不可缺少的一部分，而一个社区可以给各种各样的游戏获得关注的平台，可以给各种各样的玩家提供一个交流的平台。

本社区主要可以给各种各样的玩家群体提供服务。作为游戏本身而言，游戏所具有的特性是未知，所以才会有趣，如果提前知道那个未知的部分，则或多或少会让玩家对游戏本身减少一些好感度，尽管对于一部分的游戏而言，游戏攻略必不可少，但终归是少数现象，而不应该称之为“绝大部分”，如果说一个游戏攻略必不可少，那么我想请问各位玩家，游戏乐趣何在？游戏本身的难度调节是一个值得讨论的话题，所以才会造就游戏攻略的主因之一，其次一个游戏攻略，有另外一个目的—— 挖掘游戏深层次的内容或“游戏之美”，来让玩家更进一步体会游戏内涵和真正的乐趣所在。而本社区提供的攻略只是其中一项。一个优秀的社区可以给用户提供优质的内容，极大的给用户增加游戏体验。面对千亿美元以上，数十亿的玩家用户，一个好的游戏社区可以给玩家带来良好的体验。这个社区可以给用户提供一站式服务，可以查看游戏的资讯，攻略，百科等。同时每个人多可以发布自己的东西，即使你不是一个高玩，这可以极大的增加玩家的归属感和参与感

# 第2章 系统开发技术及工具简介

## 2.1 系统开发技术简介

### 2.1.1 ElementUI,Ant Design以及less技术简介

ElementUI是一套基于Vue2.0的桌面端，遵循一致性（与现实生活一致，在界面中一致），反馈（控制反馈，页面反馈），效率（简化流程，清晰明确，帮助用户识别），可控（用户决策，结果可控）的设计原则，由饿了么公司前端团队开源。

Ant Design同样是一套组件库，也可以在React，Vue上面使用，这是蚂蚁集团的企业级产品，这是一个庞大且复杂的系统，数量多且功能复杂，Ant Design基于「自然」、「确定性」、「意义感」、「生长性」四大设计价值观，通过模块化解决方案，降低冗余的生产成本，让设计者专注于更好的用户体验。Ant Design 提供了一套非常完整的 组件化设计规范 与 组件化编码规范，大幅提高了部分产品的设计研发效率及质量。Ant Design 与 G2 的代码质量和文档质量都非常高。

Less是一门向后兼容的 CSS 扩展语言。Less 仅对 CSS 语言增加了少许方便的扩展，这让Less十分易学。Less扩充了 CSS 语言，增加了诸如变量、混合（mixin）、函数等功能，让 CSS 更易维护、方便制作主题、扩充。Less 可以运行在 Node 或浏览器端。本质上，Less 包含一套自定义的语法及一个解析器，用户根据这些语法定义自己的样式规则，这些规则最终会通过解析器，编译生成对应的 CSS 文件。Less 并没有裁剪 CSS 原有的特性，更不是用来取代 CSS 的，而是在现有 CSS 语法的基础上，为 CSS 加入程序式语言的特性。

### 2.1.2 Vue.js及组件技术简介

Vue是用于构建用户界面的渐进式框架，Vue只关注视图，易于上手，方便和第三方库整合，采用自底向上增量开发的设计。Vue 最独特的特性之一，就是其非侵入性的响应式系统。数据模型仅仅是普通的 JavaScript 对象。而当你修改它们时，视图会进行更新。这使得状态管理非常简单直接。在开发中和Vue Router构建单页面应用, Vue Router是 Vue.js (opens new window)官方的路由管理器。

Vuex 是一个专为 Vue.js 应用程序开发的状态管理模式 + 库。它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。

Vue CLI是一个基于Vue.js进行快速开发的系统，可以通过插件进行扩展，致力于将 Vue 生态中的工具基础标准化。它确保了各种构建工具能够基于智能的默认配置即可平稳衔接，这样可以专注在撰写应用上，而不必花好几天去纠结配置的问题。

2.1.3 Node.js技术简介

Node.js是运行在服务端的JavaScript，基于Chrome V8的JavaScript运行环境，在设计上Node.js更深入地考虑了事件模型，它将事件循环作为一个运行时结构而不是作为一个库来呈现。在其他系统中，总是有一个阻塞调用来启动事件循环。通常情况下，要执行的行为是通过脚本开始时的回调来定义的，然后通过 EventMachine::run() 这样的阻塞调用来启动服务器。而在 Node.js 中，没有这种启动事件循环的调用。Node.js 在执行输入脚本后直接进入事件循环，当没有更多的回调要执行时，Node.js 就会退出事件循环。这种行为就像浏览器的 JavaScript 一样 —— 事件循环对用户是隐藏的。

Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。可以从 node.js 创建 http 请求，从浏览器中创建 XMLHttpRequests，支持 Promise API，拦截请求和响应，转换请求数据和响应数据，取消请求，自动转换 JSON 数据，客户端支持防御 XSRF。

### 2.1.4 Redis技术简介

Redis 是一个开源的使用 ANSI C 语言编写、遵守 BSD 协议、支持网络、可基于内存、分布式、可选持久性的键值对(Key-Value)存储数据库，并提供多种语言的 API。Redis支持数据的持久化，可以将内存中的数据保存在磁盘中，重启的时候可以再次加载进行使用。Redis不仅仅支持简单的key-value类型的数据，同时还提供list，set，zset，hash等数据结构的存储。Redis支持数据的备份，即master-slave模式的数据备份。Redis有着更为复杂的数据结构并且提供对他们的原子性操作，这是一个不同于其他数据库的进化路径。Redis的数据类型都是基于基本数据结构的同时对程序员透明，无需进行额外的抽象。Redis运行在内存中但是可以持久化到磁盘，所以在对不同数据集进行高速读写时需要权衡内存，因为数据量不能大于硬件内存。在内存数据库方面的另一个优点是，相比在磁盘上相同的复杂的数据结构，在内存中操作起来非常简单，这样Redis可以做很多内部复杂性很强的事情。同时，在磁盘格式方面他们是紧凑的以追加的方式产生的，因为他们并不需要进行随机访问。

### 2.1.5 Docker技术简介

Docker开源的应用容器引擎。容器是完全使用沙箱机制，相互之间不会有任何接口（类似 iPhone 的 app）,更重要的是容器性能开销极低。Docker是基于 Go 语言 并遵从 Apache2.0 协议开源。Docker 使您能够将应用程序与基础架构分开，从而可以快速交付软件。借助 Docker，您可以与管理应用程序相同的方式来管理基础架构。通过利用 Docker 的方法来快速交付，测试和部署代码，您可以大大减少编写代码和在生产环境中运行代码之间的延迟。Docker 允许开发人员使用您提供的应用程序或服务的本地容器在标准化环境中工作，从而简化了开发的生命周期。容器非常适合持续集成和持续交付（CI / CD）工作流程。Docker 是基于容器的平台，允许高度可移植的工作负载。Docker 容器可以在开发人员的本机上，数据中心的物理或虚拟机上，云服务上或混合环境中运行。Docker 轻巧快速。它为基于虚拟机管理程序的虚拟机提供了可行、经济、高效的替代方案，因此您可以利用更多的计算能力来实现业务目标。Docker 非常适合于高密度环境以及中小型部署，而您可以用更少的资源做更多的事情。这就是Docker的优点快速，一致地交付您的应用程序，响应式部署和扩展和在同一硬件上运行更多工作负载。

Docker 包括三个基本概念: 镜像: Docker 镜像（Image），就相当于是一个 root 文件系统。比如官方镜像 ubuntu:16.04 就包含了完整的一套 Ubuntu16.04 最小系统的 root 文件系统。容器：镜像（Image）和容器（Container）的关系，就像是面向对象程序设计中的类和实例一样，镜像是静态的定义，容器是镜像运行时的实体。容器可以被创建、启动、停止、删除、暂停等。仓库：仓库可看成一个代码控制中心，用来保存镜像。Docker 使用客户端-服务器 (C/S) 架构模式，使用远程API来管理和创建Docker容器。Docker 容器通过 Docker 镜像来创建。容器与镜像的关系类似于面向 对象编程中的对象与类。

### 2.1.6 Elasticsearch技术简介

Elasticsearch是一个开源搜索和分析引擎，支持分布式和RESTful风格，在速度和可扩展性方面都表现的很出色，而且可以对很多的类型的内容进行索引，这就决定了他的应用场景，比如网站搜索，日志处理和分析，基础设施指标和容器监测，应用程序性能监测，地理空间数据分析和可视化，安全分析，业务分析等等，可以执行及合并多种类型的搜索（结构化数据、非结构化数据、地理位置、指标），可以轻松的水平扩展，每秒钟处理海量事件，同时能够自动管理索引和查询在集群中的分布方式，以实现极其流畅的操作。并且保障了集群或者数据的安全性和可用性，在设计之初就考虑到了分布式的安全性和可用性的问题

Elastic的底层是Apache开源库的Lucene,对于Lucene来说，需要通过编写的代码去调用接口，Elastic对Lucene进行了封装，提供了REST API的操作接口，天然的跨平台。在Elastic中使用了倒排索引，这就使得在搜索的方面上比关系型数据库要快速

### 2.1.7 RabbitMQ技术简介

消息（Message）是指在应用间传送的数据。消息可以非常简单，比如只包含文本字符串，也可以更复杂，可能包含嵌入对象。消息队列（Message Queue）是一种应用间的通信方式，消息发送后可以立即返回，由消息系统来确保消息的可靠传递。消息发布者只管把消息发布到 MQ 中而不用管谁来取，消息使用者只管从 MQ 中取消息而不管是谁发布的。这样发布者和使用者都不用知道对方的存在。异步、削峰、解耦是消息队列的经典场景。RabbitMQ 是一个由 Erlang 语言开发的 AMQP 的开源实现。RabbitMQ 最初起源于金融系统，用于在分布式系统中存储转发消息，在易用性、扩展性、高可用性等方面表现不俗。

### 2.1.8 云服务技术简介

### 2.1.9 MySQL技术简介

MySQL 是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS(关系数据库管理系统)应用软件之一。同为关系型数据库的代表有: Oracle、DB2、SQL Server、SyBase 等。

### 2.1.10 Mybatis技术简介

Mybatis是一个ORM框架，是基于Java的持久层框架，支持自定义SQL，存储过程，以及高级映射，免除了几乎所有的 JDBC 代码以及设置参数和获取结果集的工作。只需要通过简单的注解或者XML配置和映射原始类型、接口和 Java POJO（Plain Old Java Objects，普通老式 Java 对象）为数据库中的记录。

Mybatis-plus是一个mybatis增强工具，对于mybatis只做增强不做改变

### 2.1.11 Nginx技术简介

Nginx 是一个高性能的HTTP和反向代理web服务器，同时也提供了IMAP/POP3/SMTP服务。由俄罗斯的程序设计师Igor Sysoev所开发。其特点是占有内存少，并发能力强，事实上nginx的并发能力在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：百度、京东、新浪、网易、腾讯、淘宝等。Nginx专为性能优化而开发，性能是器最重要的考量，实现上非常注重效率，能经受高负载的考验，据报告能支持高达50,000个并发连接数。Nginx不仅能做反向代理，实现负载均衡；还能可以作正向代理来进行上网等功能。Nginx主要功能有反向代理，负载均衡，动静分离。

### 2.1.12 Git技术简介

Git是版本控制系统，他是开源的，分布式的。因为采用分布式版本库的方式，不必服务端软件的支持，

### 2.1.13 Maven技术简介

Apache Maven 是一个软件项目管理和构建工具。基于项目对象模型 (POM) 的概念，Maven 可以从一条中心信息管理项目的构建、报告和文档。Maven 的主要目标是让开发人员能够在最短的时间内构建开发工作的完整环境。

### 2.1.14 SpringBoot技术简介

SpringBoot可以快速的创建直接运行的，独立的，生产级的，基于Spring的应用程序，他内嵌了服务器，可以不用部署服务器，直接使用命令运行，极大的简化了Spring的开发，以及Spring的配置，有两个重要的策略：开箱即用和约定大于配置

SpringBoot具备许多强大的功能，在Spring的基础上，对Spring进行了大量的自动配置，同时可以快速的整合其他技术，比如本社区使用到的邮件服务

### 2.1.15 SpringCloud相关组件技术简介

Spring Cloud Ribbon是一套客户端的负载均衡的工具，他是基于Netflix Ribbon实现的，Ribbon是Netflix发布的开源项目，主要功能是提供客户端的软件负载均衡算法和服务调用。Ribbon客户端组件提供一系列完善的配置项如连接超时，重试等。简单的说，就是在配置文件中列出Load Balancer（简称LB）后面所有的机器，Ribbon会自动的帮助你基于某种规则（如简单轮询，随机连接等）去连接这些机器。我们很容易使用Ribbon实现自定义的负载均衡算法。他的原理就是第几次请求数 % 服务器集群总数量 = 实际调用服务器位置下标 ，每次服务重启动后计数从1开始。

Spring Cloud OpenFeign是Spring Cloud在Feign的基础上支持了SpringMVC的注解，比如@RequestMapper。Feign是一个声明式WebService客户端。使用Feign能让编写Web Service客户端更加简单。它的使用方法是定义一个服务接口然后在上面添加注解。Feign也支持可拔插式的编码器和解码器。Spring Cloud对Feign进行了封装，使其支持了Spring MVC标准注解和HttpMessageConverters。Feign可以与Eureka和Ribbon组合使用以支持负载均衡在Feign当中内置了Ribbon, 用来做客户端负载均衡，去调用服务注册中心的服务。Feign的使用方式是：使用Feign的注解定义接口，调用这个接口，就可以调用服务注册中心的服务。

Hystrix是一个用于处理分布式系统的延迟和容错的开源库，在分布式系统里，许多依赖不可避免的会调用失败，比如超时、异常等，Hystrix能够保证在一个依赖出问题的情况下，不会导致整体服务失败，避免级联故障，以提高分布式系统的弹性。主要的功能就是服务降级，服务熔断，接近实时的监控。

Gateway是在Spring生态系统之上构建的API网关服务，基于Spring 5，Spring Boot 2和 Project Reactor等技术。Gateway旨在提供一种简单而有效的方式来对API进行路由，以及提供一些强大的过滤器功能， 例如：熔断、限流、重试等。目标是替代 Zuul，在Spring Cloud 2.0以上版本中，没有对新版本的Zuul 2.0以上最新高性能版本进行集成，仍然还是使用的Zuul 1.x非Reactor模式的老版本。而为了提升网关的性能，SpringCloud Gateway是基于WebFlux框架实现的，而WebFlux框架底层则使用了高性能的Reactor模式通信框架Netty。总的来说就是Webflux中的reactor-netty响应式编程组件，底层使用了Netty通讯框架。他可以反向代理，鉴权，流量控制，熔断，日志监控

Spring Cloud Sleuth提供了一套完整的服务跟踪的解决方案，在分布式系统中提供追踪解决方案并且兼容支持了zipkin。

Nacos是Spring Cloud Alibaba的一套组件，主要帮助您发现、配置和管理微服务。Nacos 提供了一组简单易用的特性集，帮助您快速实现动态服务发现、服务配置、服务元数据及流量，管理。Nacos是注册中心和配置中心。他的关键特性包括服务发现和服务健康监测，动态配置服务，动态 DNS 服务和服务及其元数据管理，并且Nacos还支持AP和CP的切换

Sentinel 是面向分布式服务架构的流量控制组件，主要以流量为切入点，从流量控制、熔断降级、系统自适应保护等多个维度来帮助您保障微服务的稳定性。Sentinel 的关键概念是资源。它可以是 Java 应用程序中的任何内容，例如，由应用程序提供的服务，或由应用程序调用的其它应用提供的服务，甚至可以是一段代码。在 Sentinel 里面，所有的资源都对应一个资源名称以及一个 Entry。Entry 可以通过对主流框架的适配自动创建，也可以通过注解的方式或调用 API 显式创建；每一个 Entry 创建的时候，同时也会创建一系列功能插槽（slot chain）。这些插槽有不同的职责。

## 2.2 系统开发工具简介

### 2.2.1 IDEA工具简介

IntelliJ IDEA 的每个方面都旨在最大限度地提高开发人员的生产力。智能编码辅助和符合人体工程学的设计共同使开发不仅富有成效，而且令人愉快。他深度智能，在 IntelliJ IDEA 为您的源代码编制索引后，它会通过在每个上下文中提供相关建议来提供极速和智能的体验：即时和智能的代码完成、动态代码分析和可靠的重构工具。有着开箱即用的体验，诸如集成版本控制系统和各种受支持的语言和框架之类的关键任务工具都在手边——没有插件麻烦。可以智能代码完成，虽然基本补全建议可见范围内的类、方法、字段和关键字的名称，但智能补全仅建议当前上下文中预期的那些类型。特定于框架的援助，虽然 IntelliJ IDEA 是 Java 的 IDE，但它也可以理解并为大量其他语言（如 SQL、JPQL、HTML、JavaScript 等）提供智能编码帮助，即使语言表达式被注入到您的字符串文字中。 Java 代码。这无疑是软件开发人员的首选 IDE。效率和智能内置于设计中，从设计、实现、构建、部署、测试和调试到重构，都可以实现非常流畅的开发工作流程体验！它加载了许多功能，还提供了许多可以集成到编辑器中的插件。

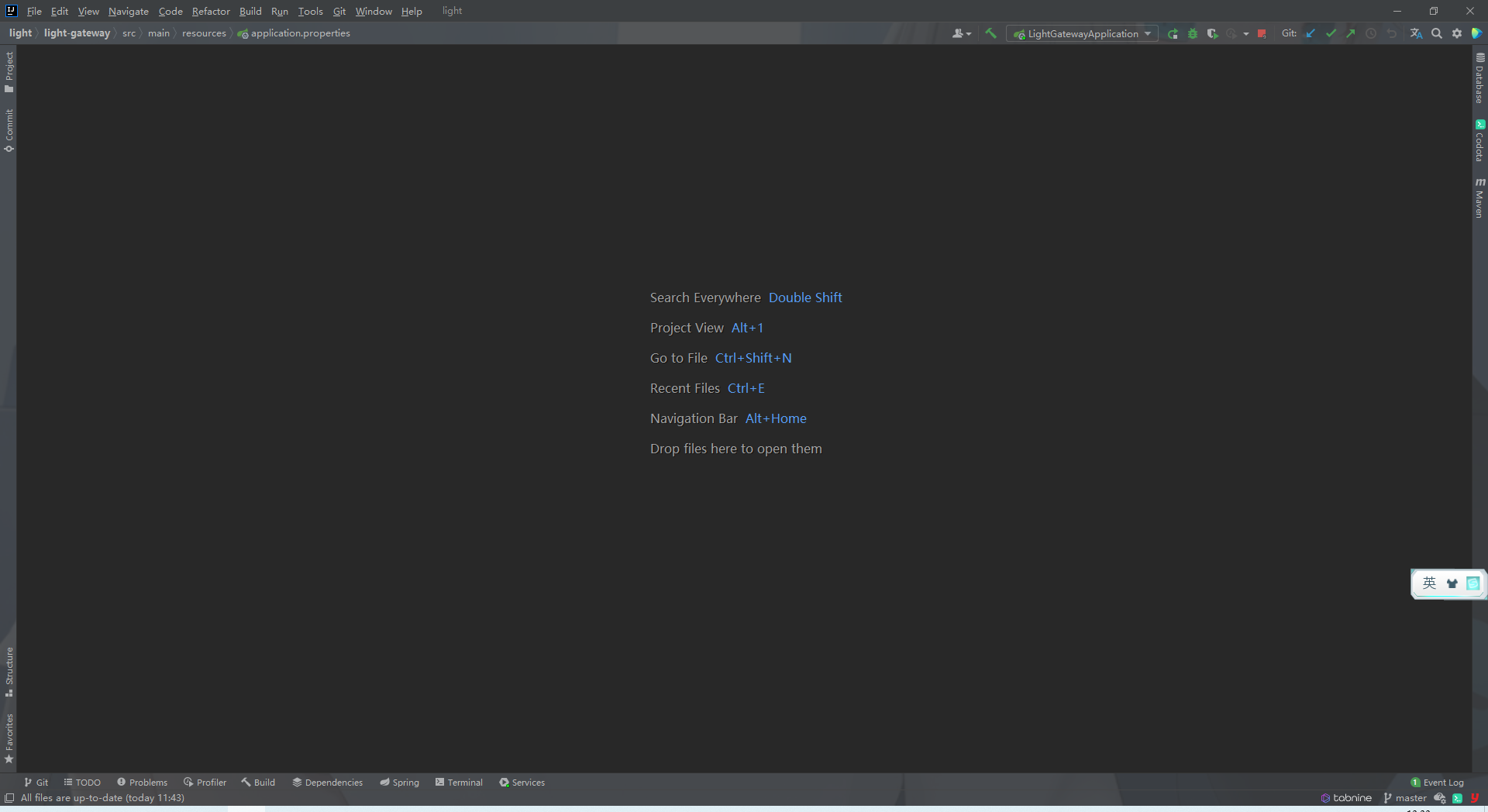


图2.1 IDEA工具结构图

### 2.2.2 Xshell工具简介

Xshell 是一款功能强大的终端模拟器，支持SSh2，SSh3，SFTP，TELNET，RLOGIN和SERIAL。通过提供业界先进的性能，Xshell 包含了其他SSH客户端无法发现的功能和优势。Xshell 7支持RSA/DSA/ECDSA/ED25516公钥，密码和键盘交互，GSSAPI，和PKCS#11身份验证。使用不断更新的加密算法列表加密您的流量，并使用Xshell的主密码特性在云中安全存储会话文件。将Xshell 7的强大特性与其选项卡环境相结合，可以使您与会话进行前 所未有的交互。使用诸如会话管理器、组合窗格、突出显示集、快速命 令、即时隧道等特性优化工作流。Xshell 7能轻松处理这些繁琐的事情。

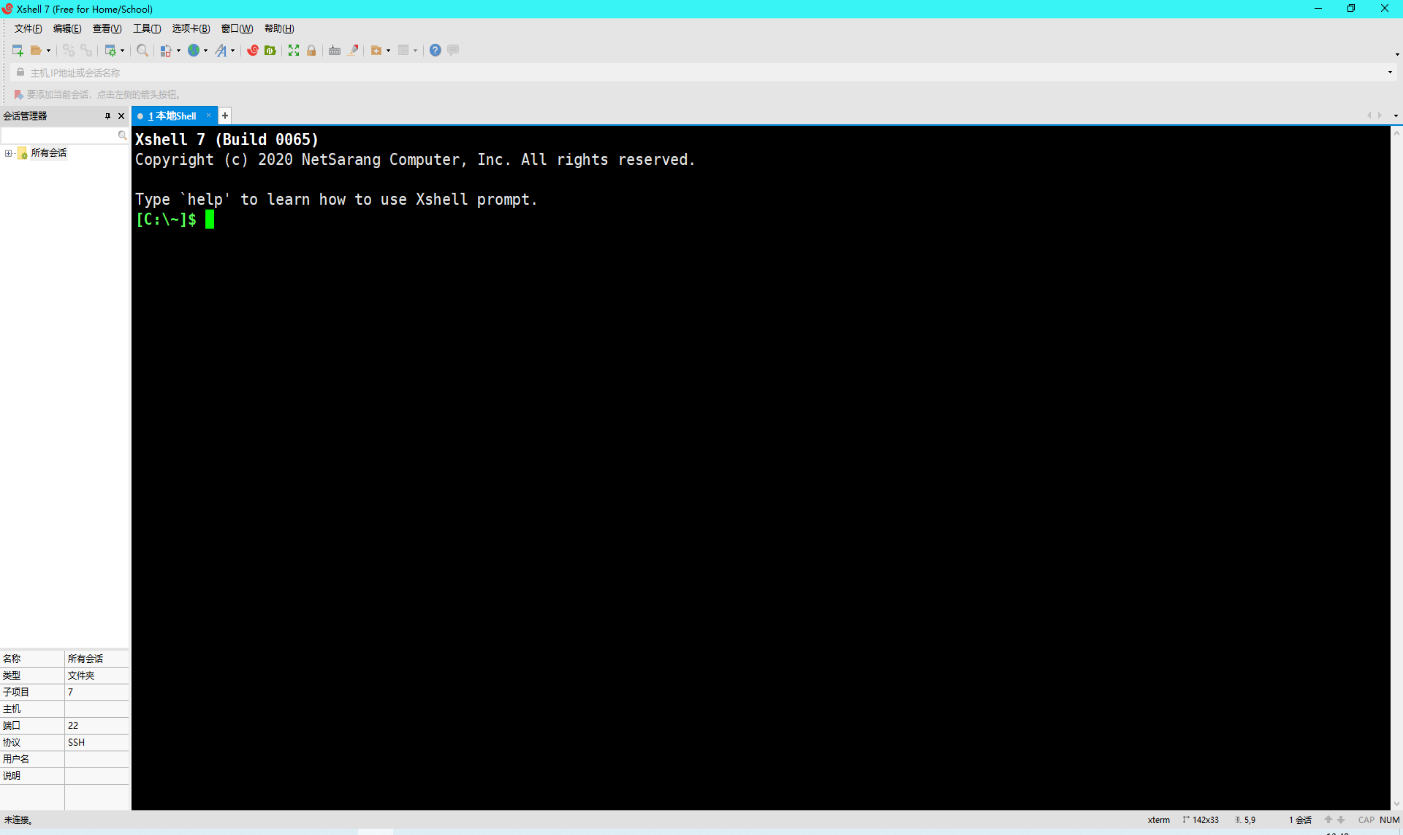


图2.2 Xshell工具结构图

### 2.2.3 webStrom工具简介

WebStorm 是一个适用于 JavaScript 和相关技术的集成开发环境。类似于其他 JetBrains IDE，它也会使您的开发体验更有趣，自动执行常规工作并帮助您轻松处理复杂任务。专门针对 JavaScript 打造, 最智能的编辑器, 内建开发者工具，快速导航和搜索，高效的团队合作，可自定义环境，WebStorm 附带了对 JavaScript、TypeScript、React、React Native、Electron、Vue、Angular、Node.js、HTML、样式表和许多其他技术的开箱即用支持。 这对您的意义是什么？ 您可以直接进行编码，无需考虑安装或维护任何插件。无论您是前端开发者还是偶尔处理 UI 的全栈开发者，WebStorm 都可以帮您专注于当前最重要的技术堆栈。 您将获得使用 JavaScript 开发高品质应用程序所需的全部工具。

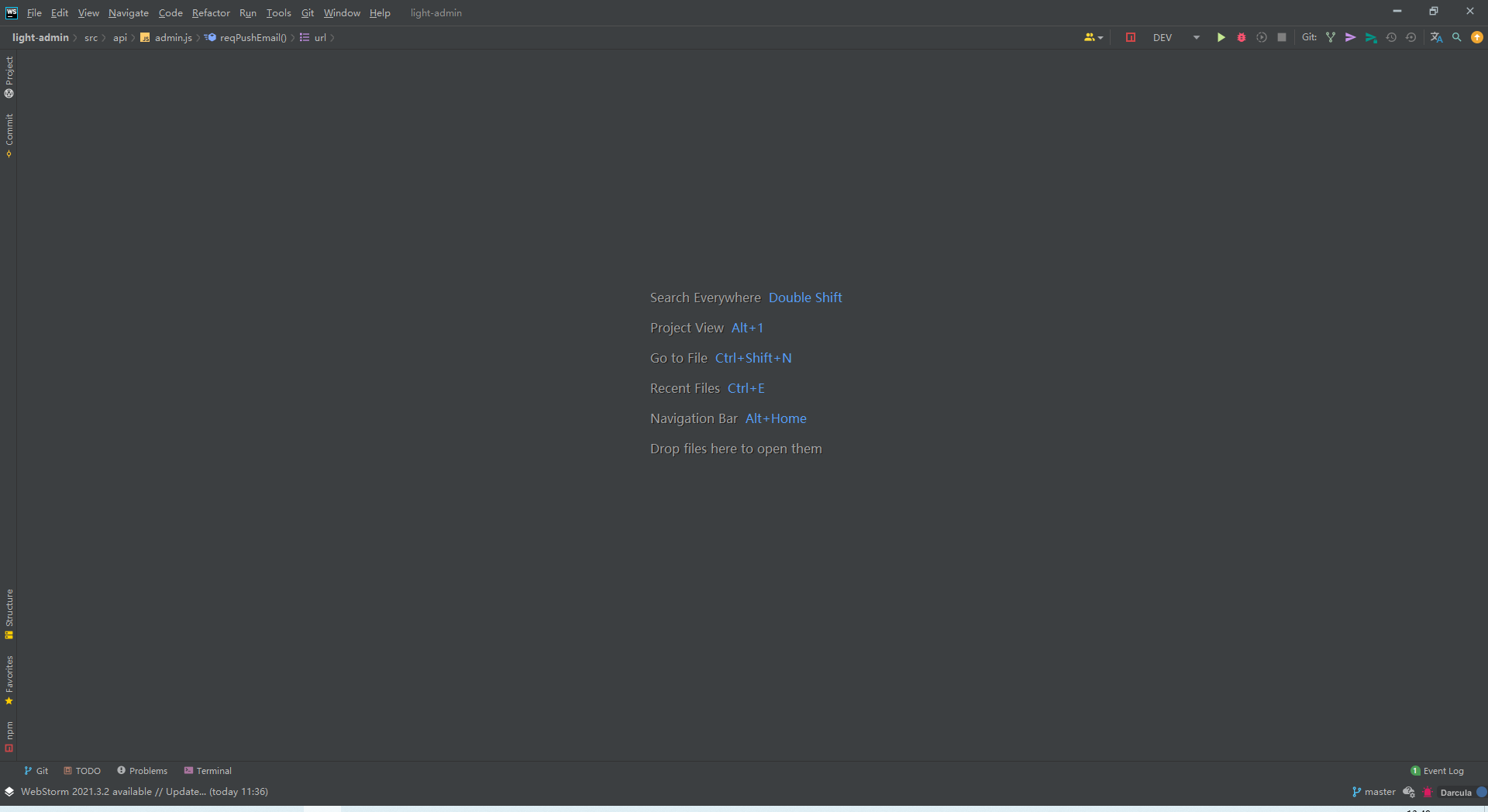


图2.3 webStrom工具结构图

### 2.2.4 Navicat Premium工具简介

Navicat Premium 是一套数据库开发工具，让你从单一应用程序中同时连接 MySQL、MariaDB、MongoDB、SQL Server、Oracle、PostgreSQL 和 SQLite 数据库。它与 Amazon RDS、Amazon Aurora、Amazon Redshift、Microsoft Azure、Oracle Cloud、MongoDB Atlas、阿里云、腾讯云和华为云等云数据库兼容。你可以快速轻松地创建、管理和维护数据库。

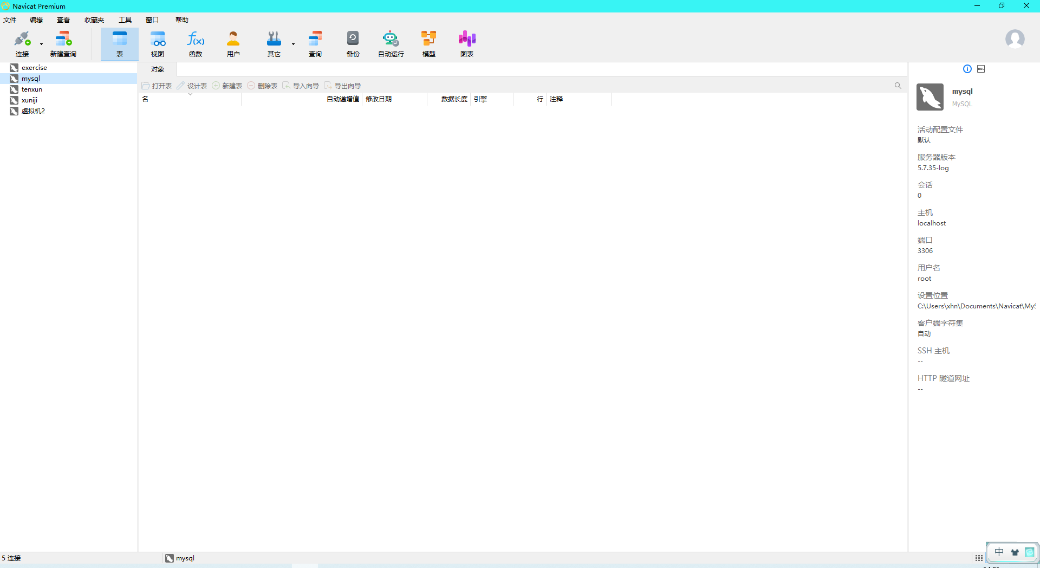


图2.4 Navicat Premium工具结构图

### 2.2.5 Postman工具简介

Postman 是一个用于构建和使用 API 的 API 平台。Postman 简化了 API 生命周期的每个步骤并简化了协作，因此您可以更快地创建更好的 API。Postman相当于一个客户端,它可以模拟用户发起的各类HTTP请求,将请求数据发送至服务端,获取对应的响应结果, 从而验证响应中的结果数据是否和预期值相匹配;并确保开发人员能够及时处理接口中的bug,进而保证产品上线之后的稳定性和安全性。 它主要是用来模拟各种HTTP请求的(如:get/post/delete/put..等等),Postman与浏览器的区别在于有的浏览器不能输出Json格式,而Postman更直观接口返回的结果。

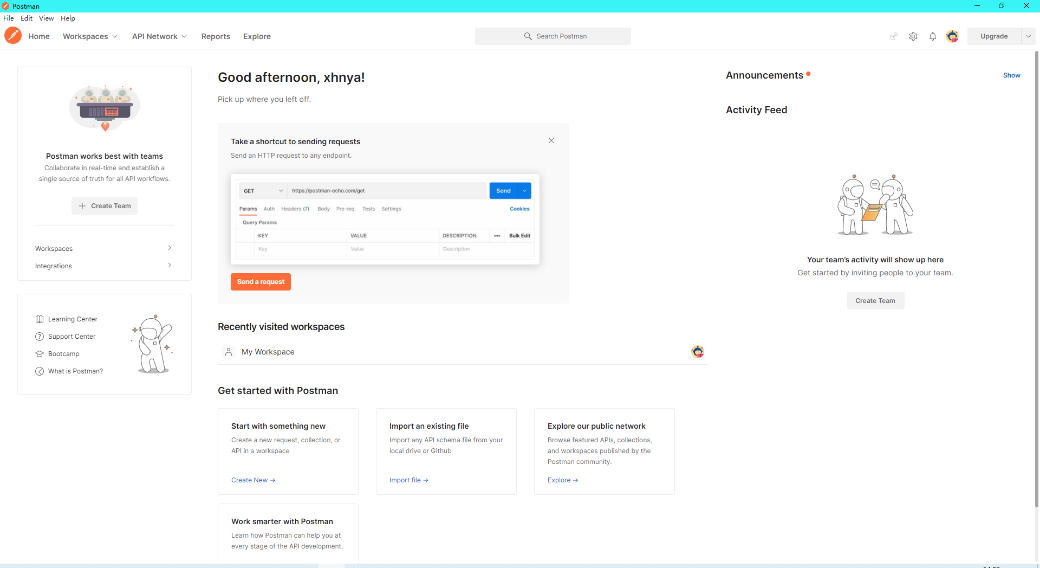


图2.5 Postman工具结构图

# 第3章 系统需求分析

## 3.1 系统可行性分析

在一个系统的开发当中，在开发子初就要分析和论证系统的可行性，消费的资源，使用的技术

### 3.1.1 经济可行性分析

本系统是利用spring boot开发的，对于开发的环境来说，比如rabbitmq,Redis等等，可以运行在docker上面，因为是分服务开发的，所以在部署的时候可以把这些打包成一个docker镜像，而这来说，在想要扩容的时候就可以轻易的扩容，并且docker具有轻量的特点，所以在这经济可行。

### 3.1.2 技术可行性分析

在本作开发的软件当中，开发的核心就是SpringBoot，采用的是微服务的方式，遵守服务自治的原则，为了解决微服务带来的问题引入了SpringCloud，使用nacos做为注册中心和配置中心，使用openFeign就行服务调用，使用gateway做为网关，加入了Sentinel进行服务限流和熔断处理，使用Sleuth进行链路追踪。在开发之初考虑到持续集成/持续部署，使用了功能强大的docker进行mysql数据库安装，和redis非关系数据库的安装，以及RabbitMQ，elasticsearch和kibana。在系统中引入redis解决缓存问题，为了把系统异步和解耦，加入了RabbitMQ消息队列。这些都是主流热门技术。考虑到mysql的全文检索能力，使用elasticsearch作为搜索功能，mysql主要还是作为数据的持久化，在开发中使用原生技术操作数据库步骤繁琐，使用ORM框架mybatis-plus和mybatis。 在服务器领域，nginx具备十分强大的功能，比如反向代理，负载均衡，动静分离，所以一些静态的资源存放在nginx当中。

在前端的页面中采用Vue.js，这是一个渐进式的JavaScript 框架，可以快速的构建出一个单页面应用，使用axios作为请求发送。使用了两个ui框架，一个是ElementUI,一个是Ant Design of Vue。为了方便管理vue的包管理，使用node.js的管理工具。

以上使用的技术都是非常主流的技术，都是当下开发当中最好的解决方案由此来看，在技术上是可行的

**第3章 系统需求分析**

**撰写说明：**

**（1）系统可行性分析**

**主要从系统的经济可行性、技术可行性和社会可行性进行系统的可行性分析。**

**（2）功能需求分析**

**从系统的角色（哪些人使用系统，一类人归为一种角色）分析入手，分析系统为每一种角色需要提供哪些功能？（也就是这种角色使用系统时，用系统的什么功能？）**

**然后按角色详细描述系统应该提供的功能需求。**

**（3）系统用例分析**

**根据3.1.1 的角色功能需求分析，为每一种角色画出用例图。**

**（4）系统性能需求分析**

**分析系统在性能、界面设计等的需求。**

## 3.2 系统功能需求分析

### 3.2.1 功能需求分析

### 3.2.2 系统用例分析

## 3.3 系统性能需求分析

在设计之初，本系统就考虑到了性能方面的要求，所以引入了Redis，消息中间件等等，并且把整个的系统部署在docker上面，对于docker来说，有一个最大的特点就是易于扩容，在你的性能需要增加的时候，可以快速的增加容器。而对于消息中间件来说，消息中间件的主要三大作用就是，异步，解耦，流量削峰，在访问量突然增加的时候，不会对系统照成剧烈的影响，让系统平稳的度过。在本系统当中使用到了Sentinel，可以对系统继续限流，熔断处理，在网关整合了sentinel。在服务器上考虑NGINX，NGINX是天然支持并发的，最大支持能够支持高达 50,000 个并发连接数的响应。考虑到开发环境计算机性能问题。在对系统使用\_\_\_\_进行压力测试的时候，\_\_\_\_\_

# 第4章 系统概要设计

## 4.1 系统功能模块设计

本系统一共三个大的模块，一个是前台，一个是后台管理页面，最后一个是运维人员相关。这是总体模块的设计。

在前台页面当中，用户可以进行登录，注册操作，当然还可以找回密码，和修改密码。用户可以游览的内容非常

对于后台管理用户来说

在本社区中还需要运维人员，运维人员可以通过Kibana对elasticsearch进行管理，上传热点词汇。使用sentinel对系统进行熔断和限流，使用sleuth进行请求的链路追踪。使用Nacos对系统的配置文件进行更新等

## 4.2 数据库设计

### 4.2.1 数据库概要设计

### 4.2.2 数据库逻辑设计

# 第5章 系统详细设计与实现

## 5.1　用户模块的设计与实现

### 5.1.1　登录注册功能的实现

在本系统当中考虑到注册的功能的繁琐，大大的简化了注册的流程，如果你是第一次登录该平台，只需要通过手机号验证登录即可自动注册一个账号，账号的用户名会自动的生成，在这之后就可以一直使用手机号登录，如果想要通过用户名/手机号/邮箱加密码的形式登录，那你需要通过找回密码，给自己设置一个密码，这样就可以通过密码登录了，在数据库当中存储的密码的加密方式是使用SHA-256 +随机盐+密钥对密码进行加密，这样的好处就是即使是相同的密码在数据库显示的数据都是不一致的。在使用手机号或者邮箱验证登陆的时候会给手机号或者邮箱发送验证码在发送验证码后会把验证码存入到redis当中，并且设置有效期，在用户登陆的时候取出验证码，运行界面如图5-1所示

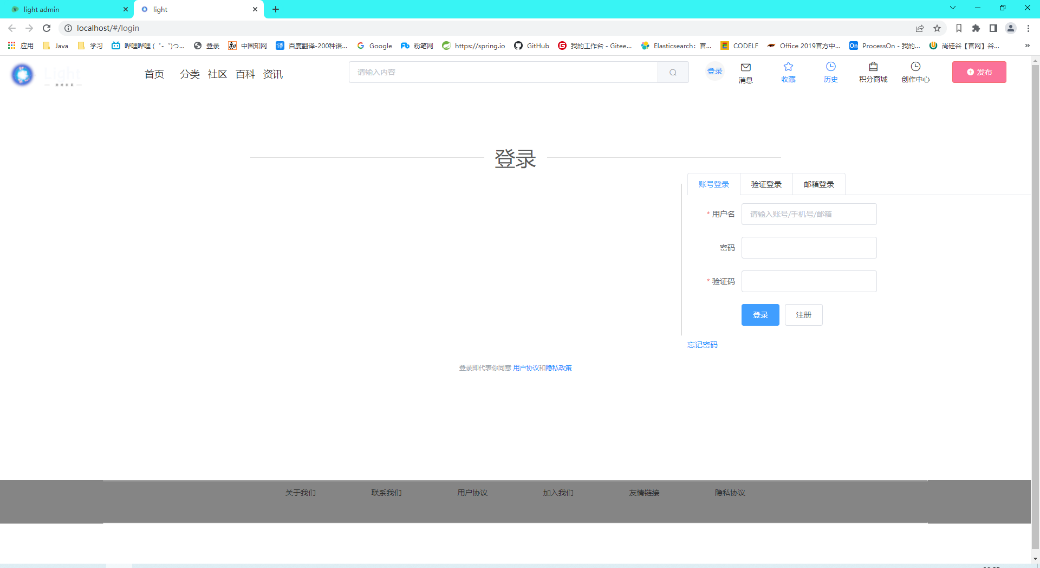


图5-1 登录注册运行图

核心代码如下：



图5-2 账号密码登录核心代码图

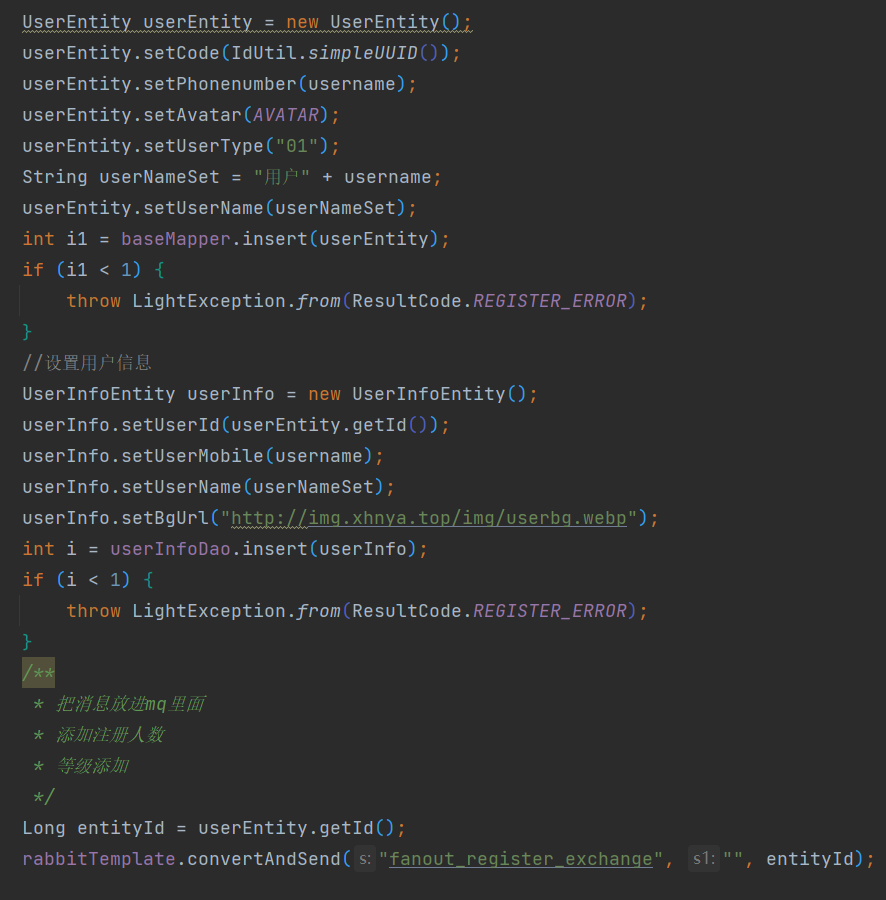


图5-3 注册核心代码图

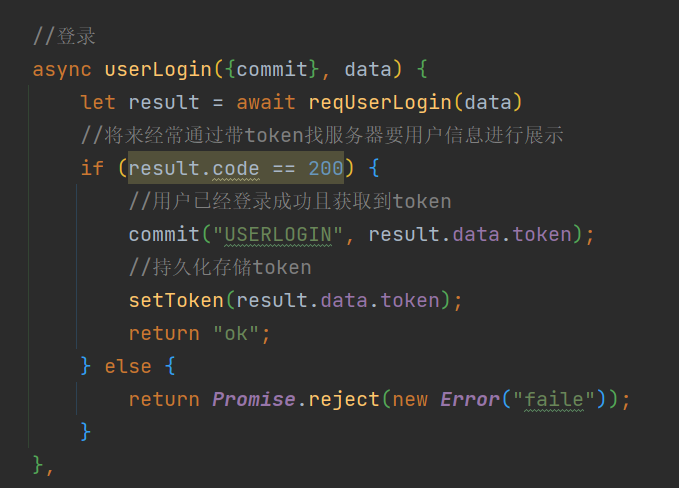


图5-3 请求登录核心代码图

### 5.1.2　找回密码功能的实现

用户可以通过手机号和邮箱重置密码，在输入手机号或者邮箱之后用户，系统会给用户发送短信验证码或者邮箱，然后就会把这个验证码存入redis当中，设置一个有效期，如果过了有效期，验证码就会失效

运行界面如下图：

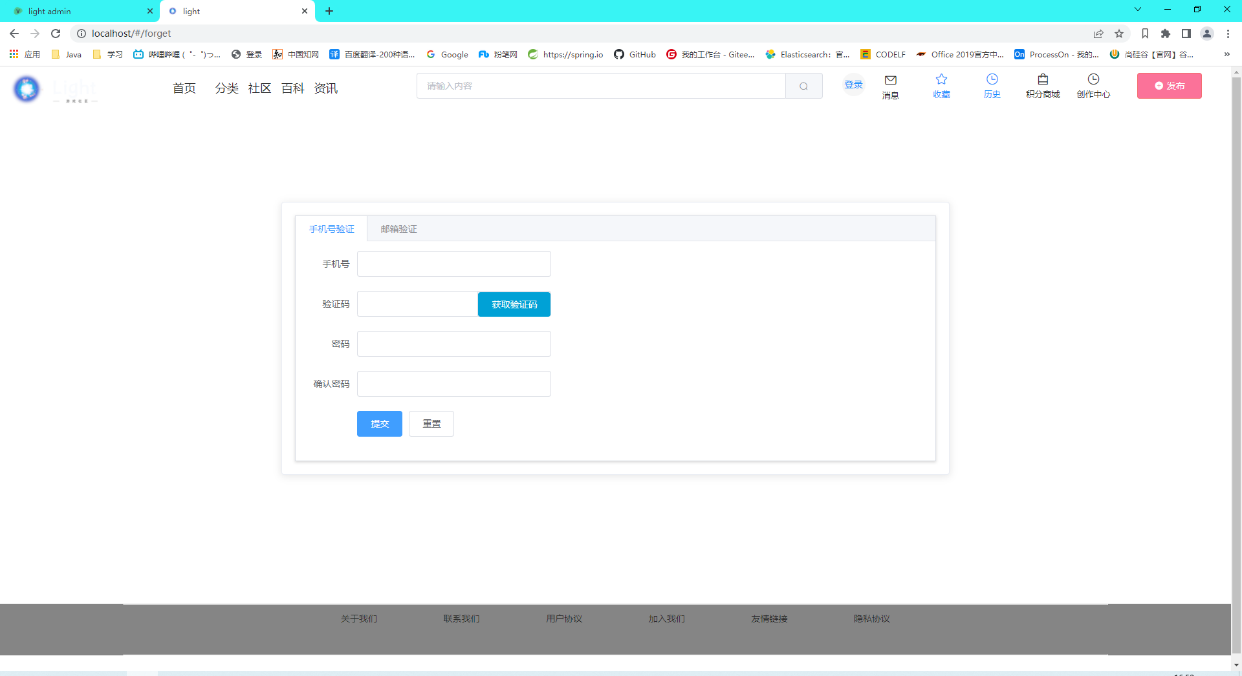


图5-4 找回密码运行界面图

核心代码如下：



图5-5 找回密码核心代码图

### 5.1.3　修改个人资料功能的实现

用户可以修改自己的一些个人资料，比如一些基本的信息，用户名，城市，个性签名，头像和背景图片。运行界面如图5-5所示



图5-6 修改个人资料运行图

核心代码如下：

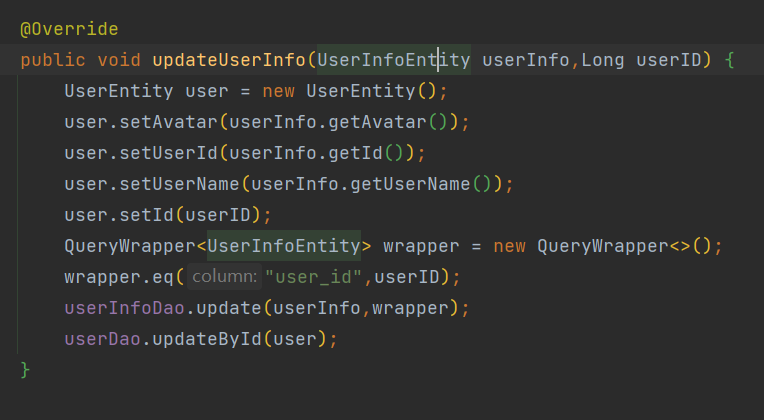


图5-7 个人资料核心代码图

### 5.1.4　收藏夹功能的实现

用户可以创建收藏夹，并且设置收藏夹的可见类型，加上备注。运行界面如图5-8

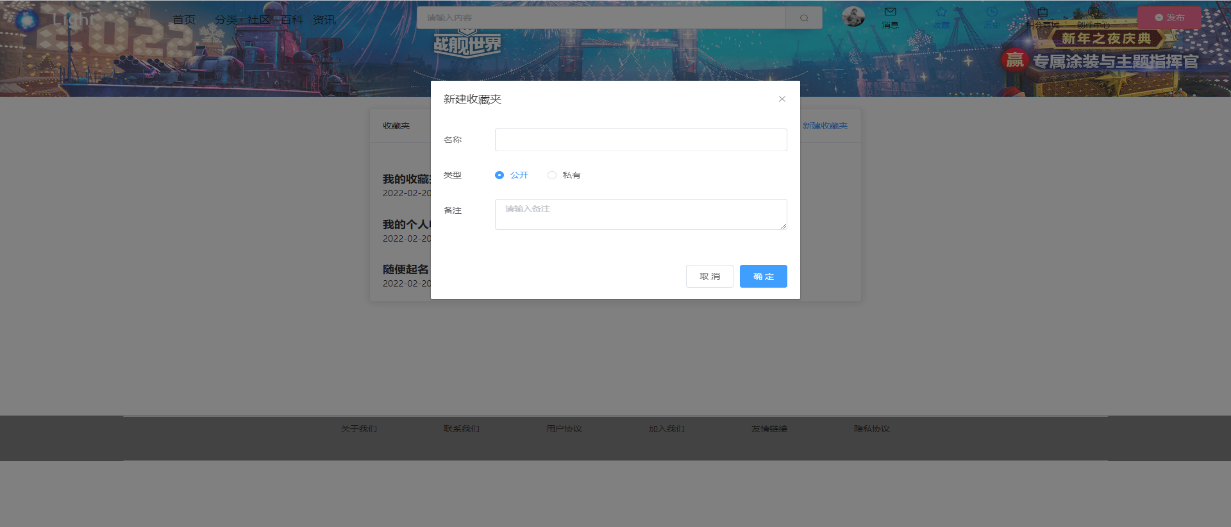


图5-8 收藏夹运行图

核心代码如下：

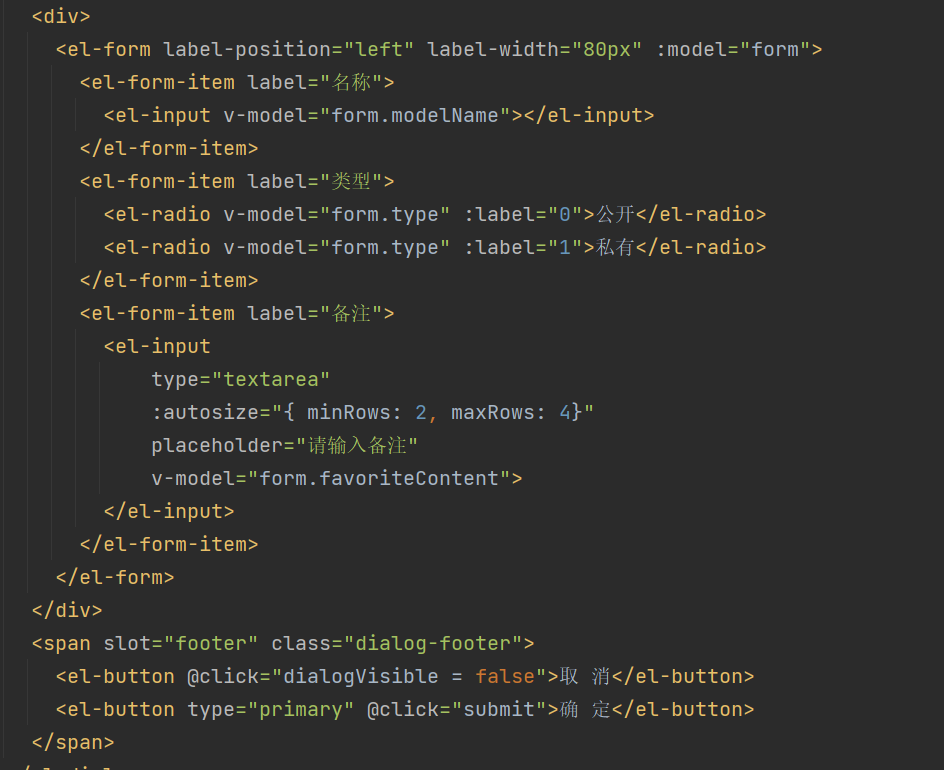


图5-9 收藏夹核心代码图

### 5.1.5　历史记录功能的实现

用户在游览社区的时候会产生历史记录，用户可以查看这些历史记录，可以对历史记录进行单个删除，以及一键清空。运行界面如图5-10



图5-10 用户历史记录图

核心代码如下：

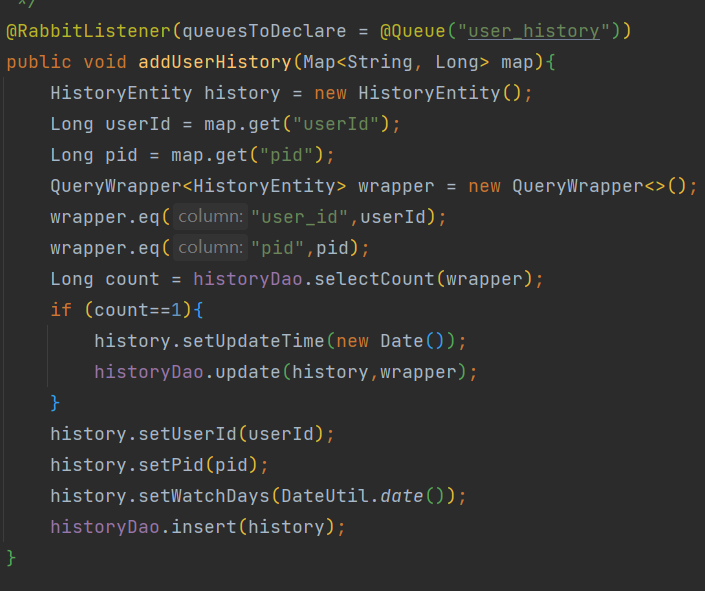


图5-11 用户历史记录核心代码图

### 5.1.6　被点赞消息提醒的实现

在发表的文章被点赞时系统会发送一条消息通知给用户，用户如果存在着未读消息则会显示这头部，运行界面如图所示



图5-12 存在未读消息提示图

核心代码如下：

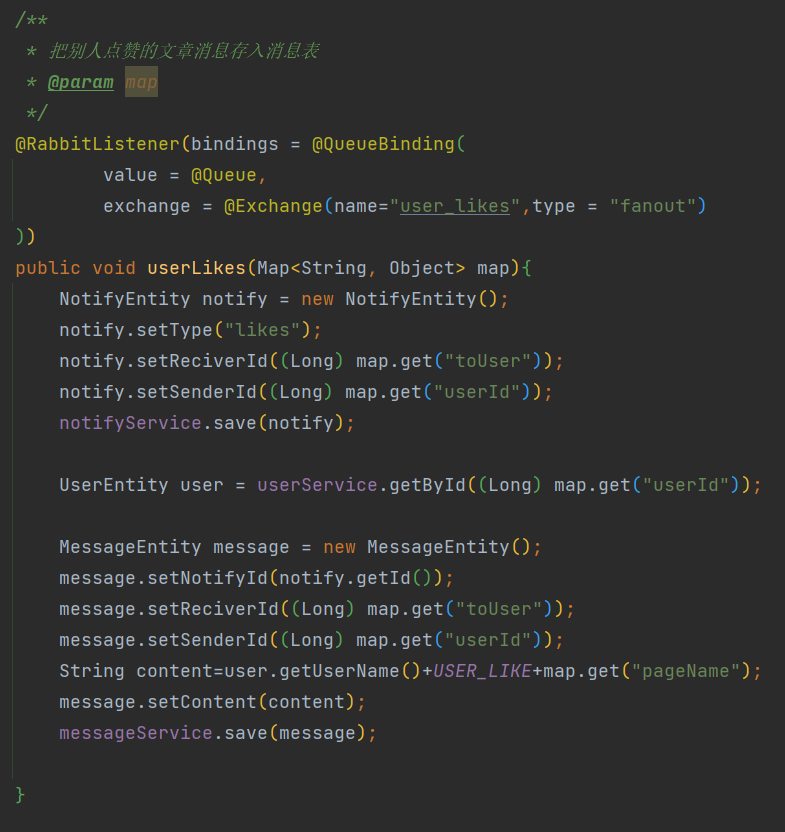


图5-13 未读消息核心代码图

### 5.1.7　查看点赞消息的实现

用户可以查看全部的点赞的消息，在查看全部的点赞消息的时候，会一键已读全部的消息，运行界面如图所示



图5-15 点赞消息运行界面图

核心代码如下：

文本

描述已自动生成

图5-15 点赞消息核心代码图

### 5.1.8　文章发表的实现

用户可以发表文章，在发表文章的时候需要选择分区，可以添加描述，可以上传封面。

运行界面如下图所示

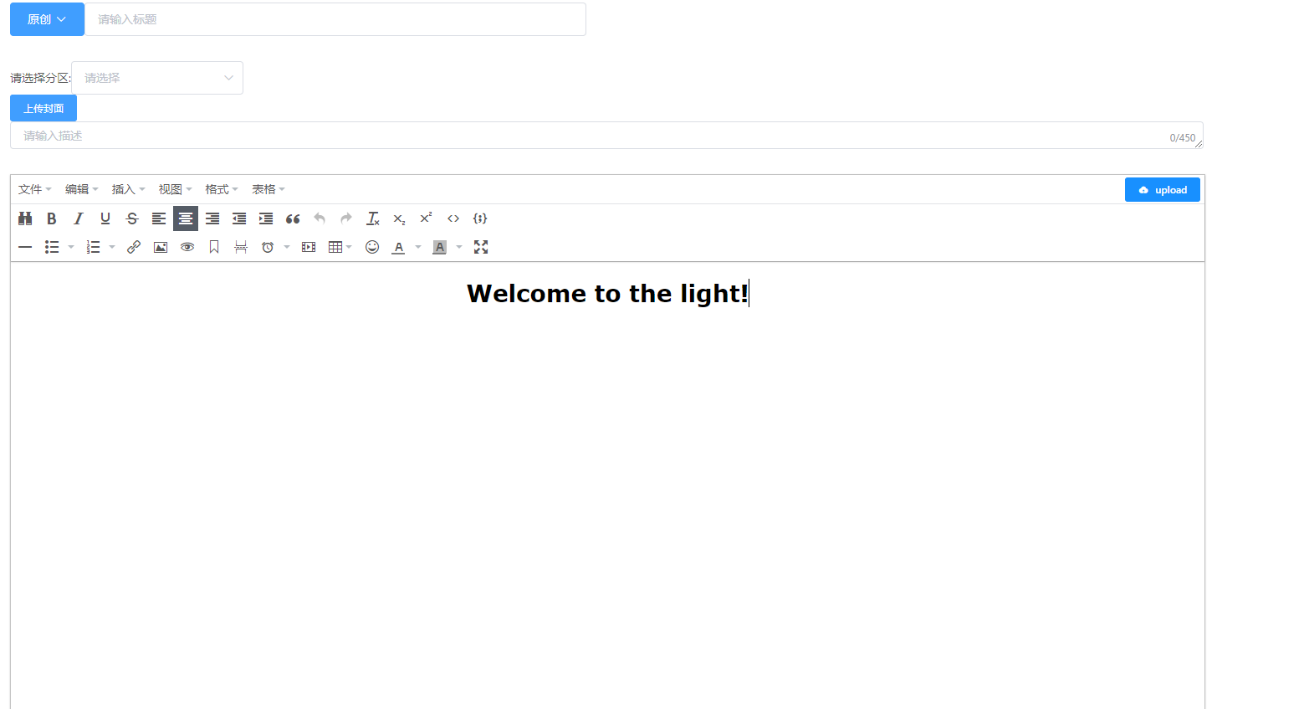


图5-16 发表文章界面图

核心代码如图：



图5-17 发表核心代码图

### 5.1.9　点赞文章的实现

用户可以在我文章的列表上点赞其他用于发表的文章，并且还会通知被点赞的用户，运行界面如图所示：

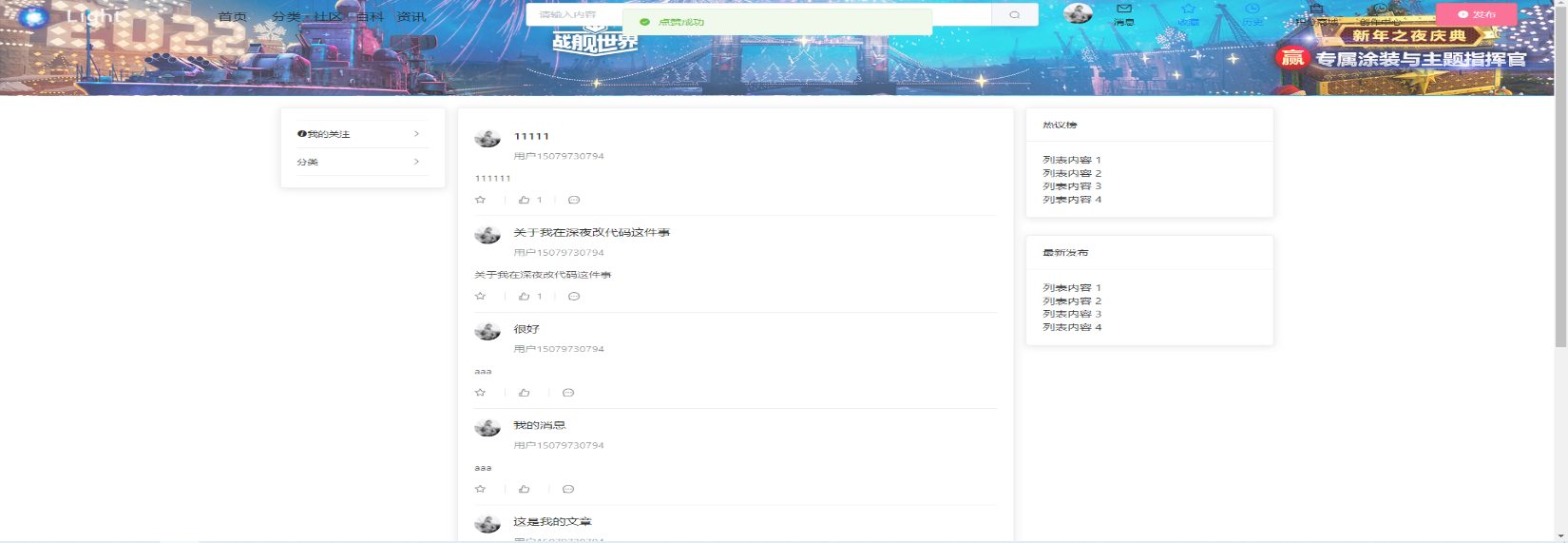


图5-18 点赞运行界面图

核心代码实现：

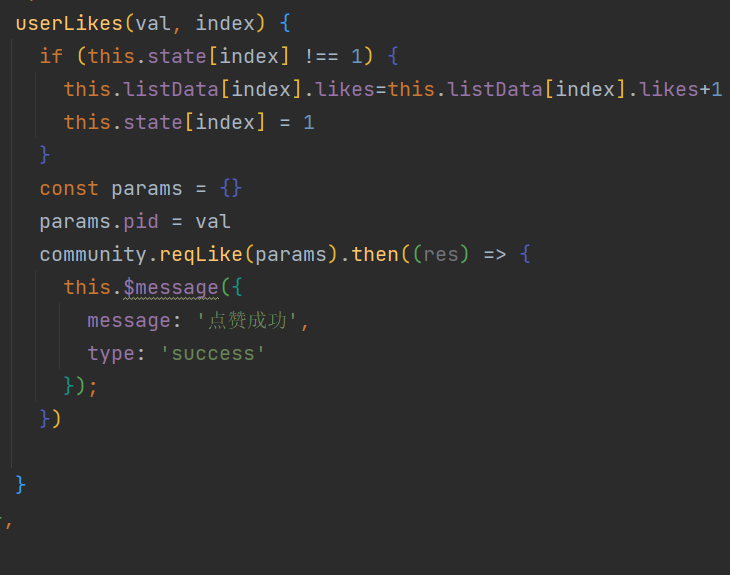


图5-19 点赞核心代码图

### 5.1.10　文章收藏的实现

用户可以收藏文章在指定的收藏夹里面，收藏了的文章会在收藏夹显示，运行界面如图所示



图5-20 收藏运行界面图

核心代码如下

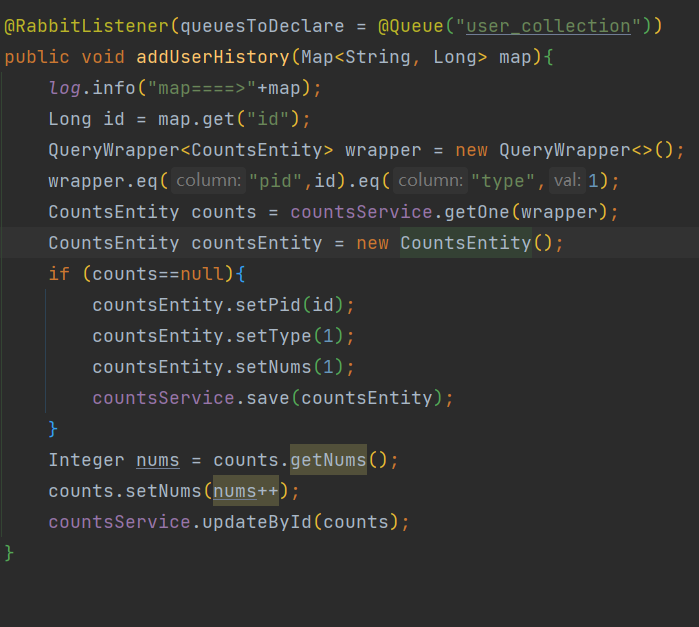


图5-21 收藏核心代码图

### 5.1.11　用户发表的文章功能的实现

用户可以在我的界面查看自己发表的文章，并且可以进行搜索等功能。运行界面如图所示

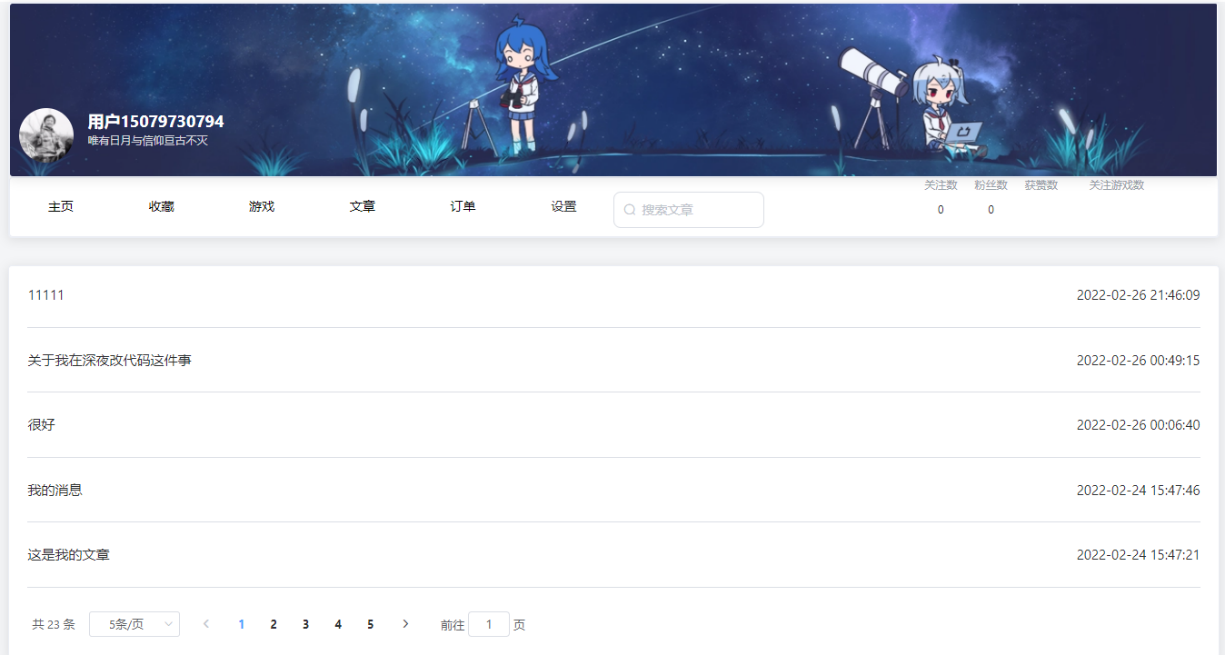


图5-22 我的文章运行界面图

核心代码如下图：



图5-23 我的文章核心代码图

### 5.1.12　全文检索功能的实现

用户可以通过顶部的搜索栏进行全文的检索，结果包括用户，游戏，文章，百科，完全匹配的会高亮显示，在搜索结果的前端会有一个标签，来标识搜索的结果是什么类型，运行界面如下：

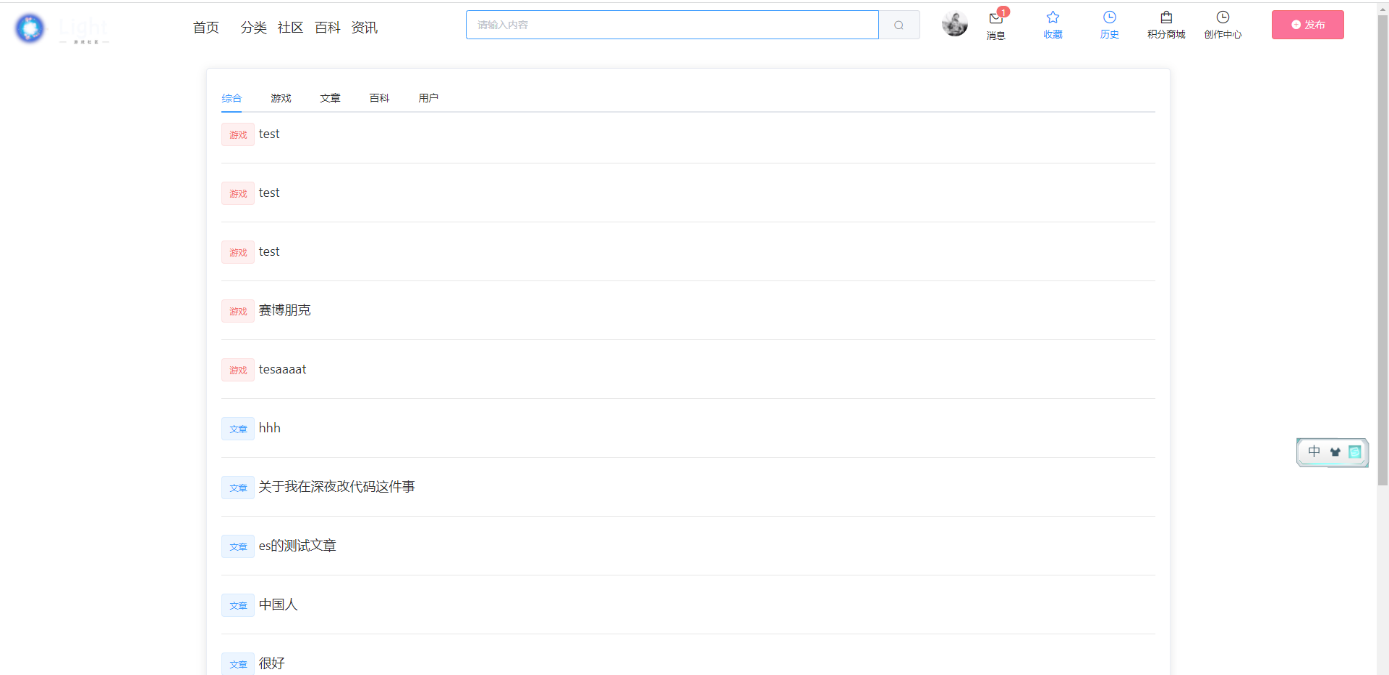


图5-24 全文检索运行图

核心代码如图所示：



图5-25 全文检索核心代码图

### 5.1.13　排行榜功能的实现

在首页显示3个排行榜，分别是新游榜，热门榜和期待榜，在移动到相应位置的时候就会显示详细信息。运行界面如图：



图5-26 排行榜运行界面图

核心代码如下：

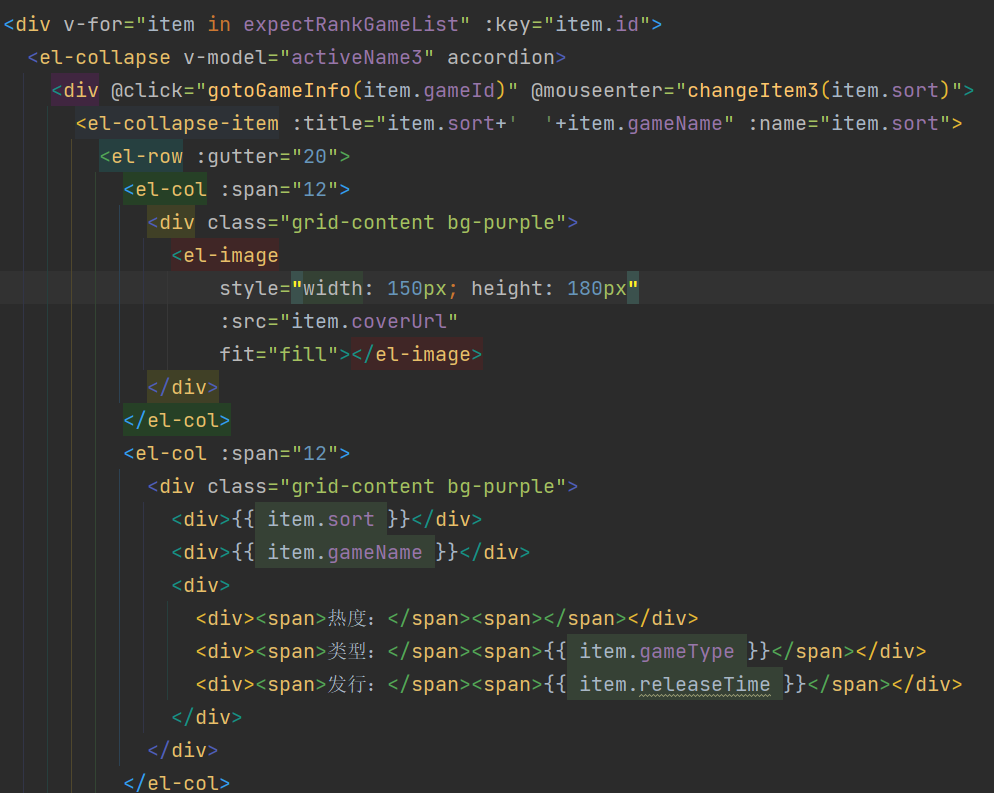


图5-27 排行榜核心代码图

### 5.1.14　游戏评论功能的实现

用户可以对每一个游戏进行评论，评论会展示在游戏的最下方，运行界面如下图：

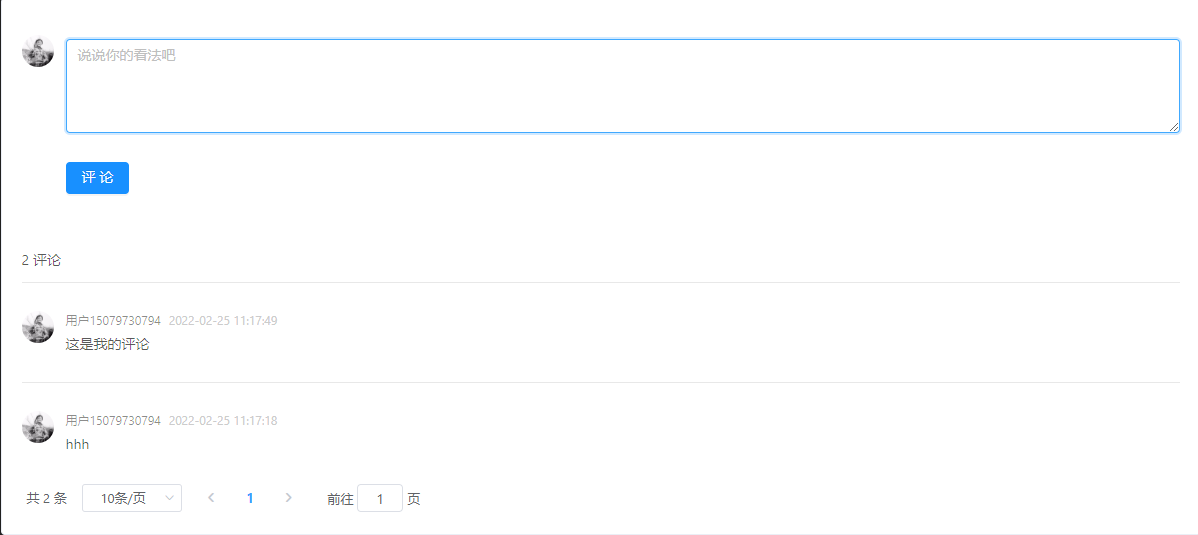


图5-28 游戏评论运行图

核心代码如下：

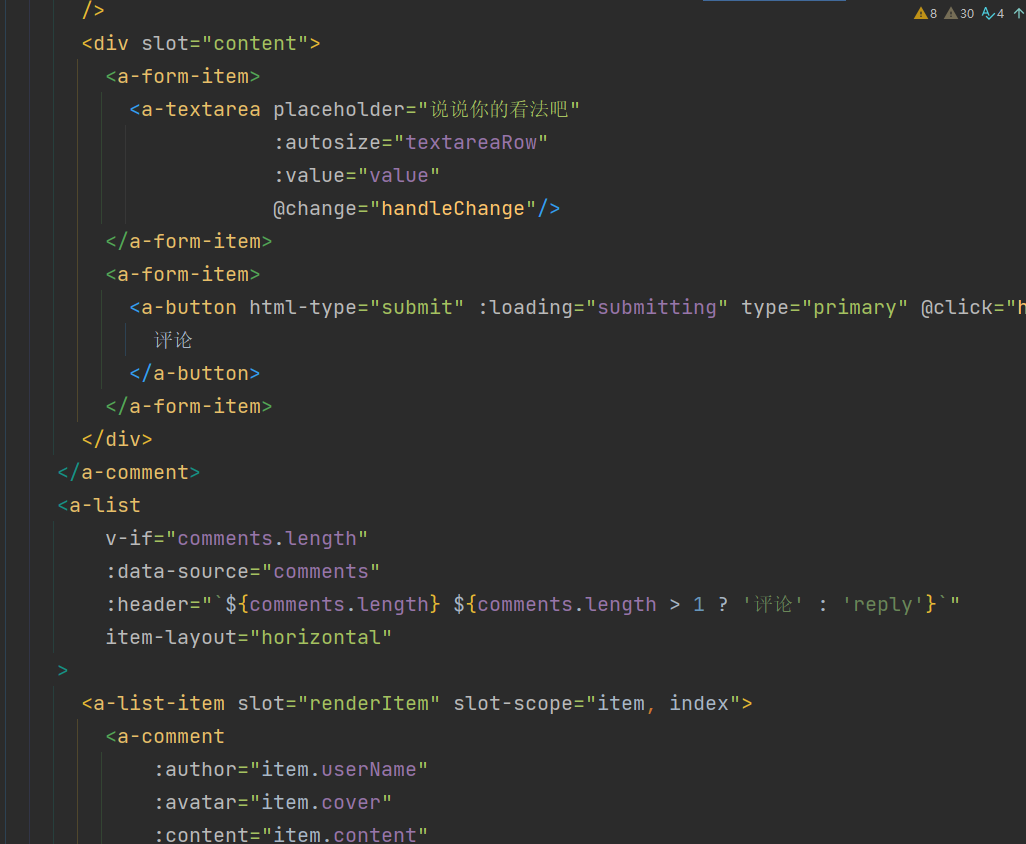


图5-29 游戏评论核心代码图

### 5.1.15　文章评论功能的实现

用户可以对每一个文章进行评论，运行界面如图所示：

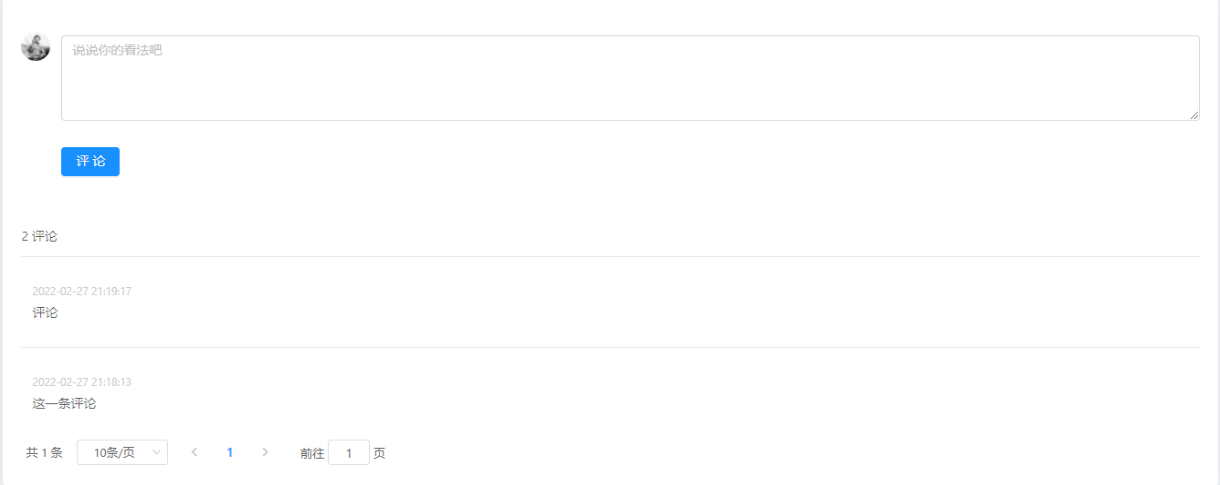


图5-30 文章评论运行界面图

核心代码图：

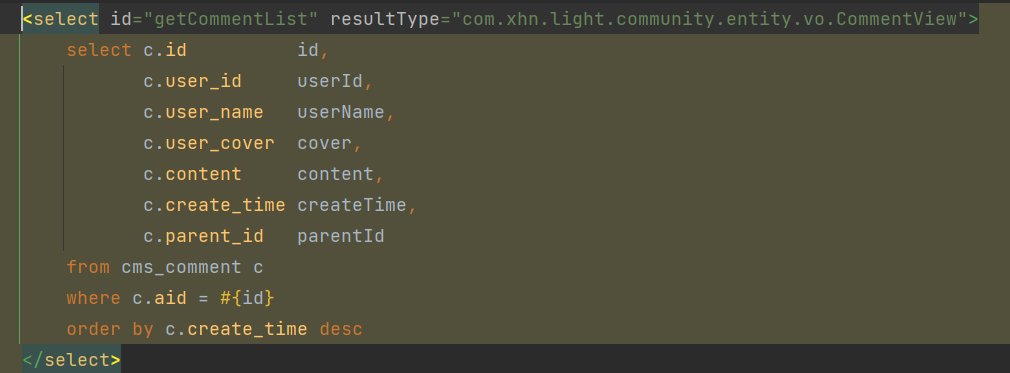


图5-31 文章评论核心代码图

### 5.1.16　百科一节菜单功能的实现

每个游戏都有对应的百科，用户可以给百科添加一级菜单。运行的代码如图所示：

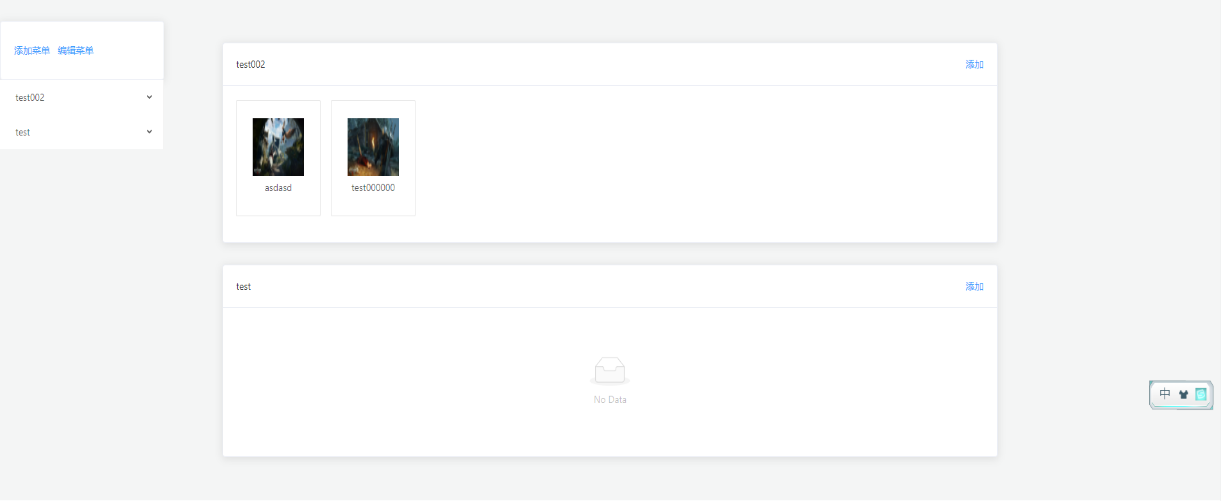


图5-32 游戏百科一级菜单运行界面图

核心代码如图所示：



图5-33 游戏百科一级菜单核心代码图

### 5.1.17　百科详细内容功能的实现

百科有一个二级菜单，在添加二级菜单的时候需要添加详细的内容，运行界面如图所示：

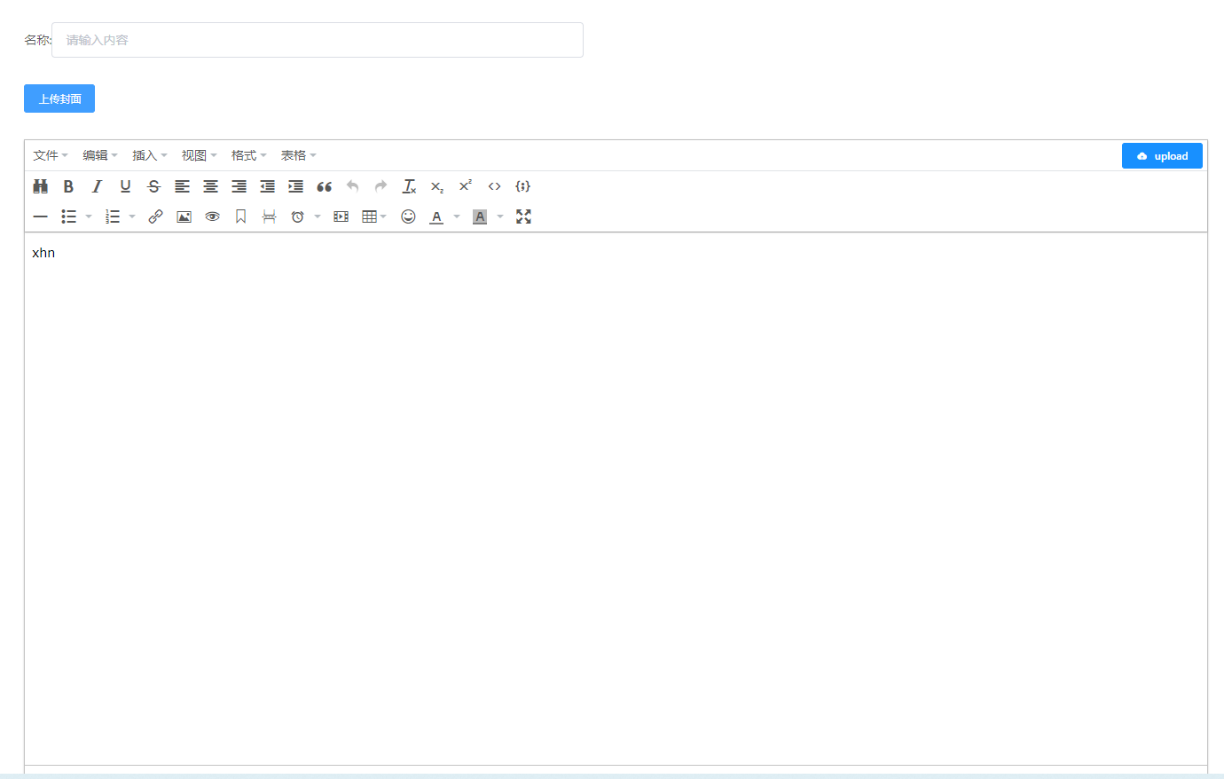


图5-34 百科详情运行界面图

核心代码如下：

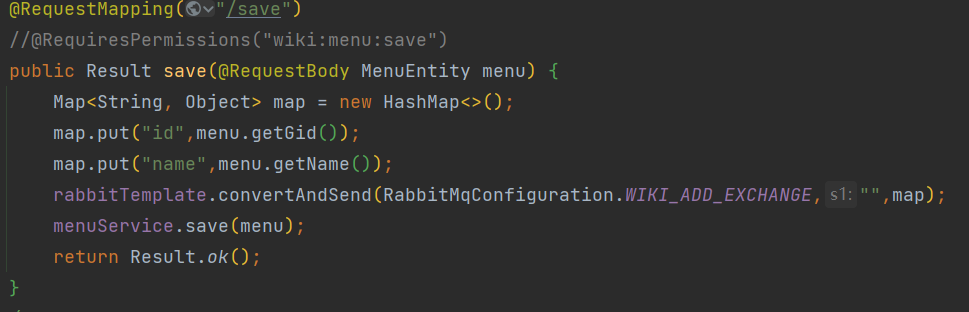


图5-34 百科详情核心代码图

## 5.2 管理模块的设计与实现

### 5.2.1　管理用户登录功能的实现

进入管理后台系统需要管理员账号密码登录，运行界面如图所示：

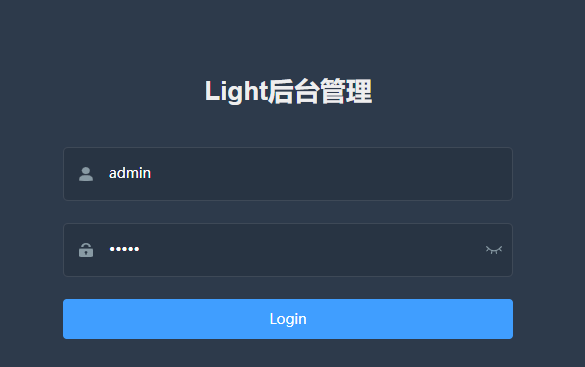


图5-35 后台管理登录运行界面图

核心代码如下：

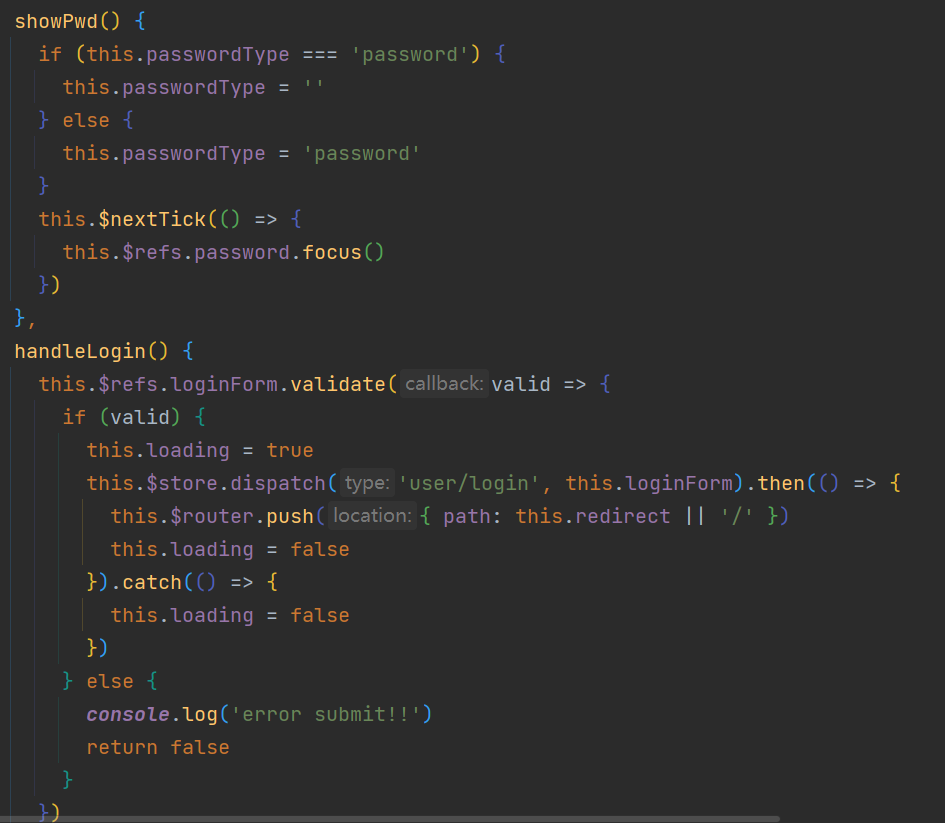


图5-36 后台管理登录核心代码图

### 5.2.2　首页轮播图管理功能的实现

管理员可以对首页的轮播图进行管理，筛选，添加，删除运行界面如图所示：



图5-37 首页轮播图运行界面图

核心代码如下：



图5-38 首页轮播图核心代码图

### 5.2.3　邮件推送功能的实现

邮件推送是管理员利用邮件给订阅了本系统的消息人员发送邮件的功能，在管理员输入内容后，可以给每一个订阅的人发送邮件，运行界面如图所示：

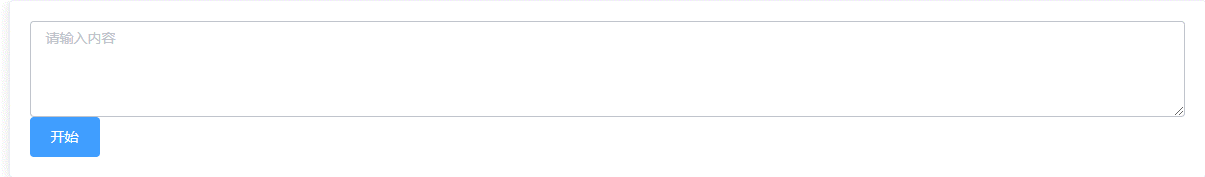


图5-39 邮件推送运行界面图

核心代码如下：

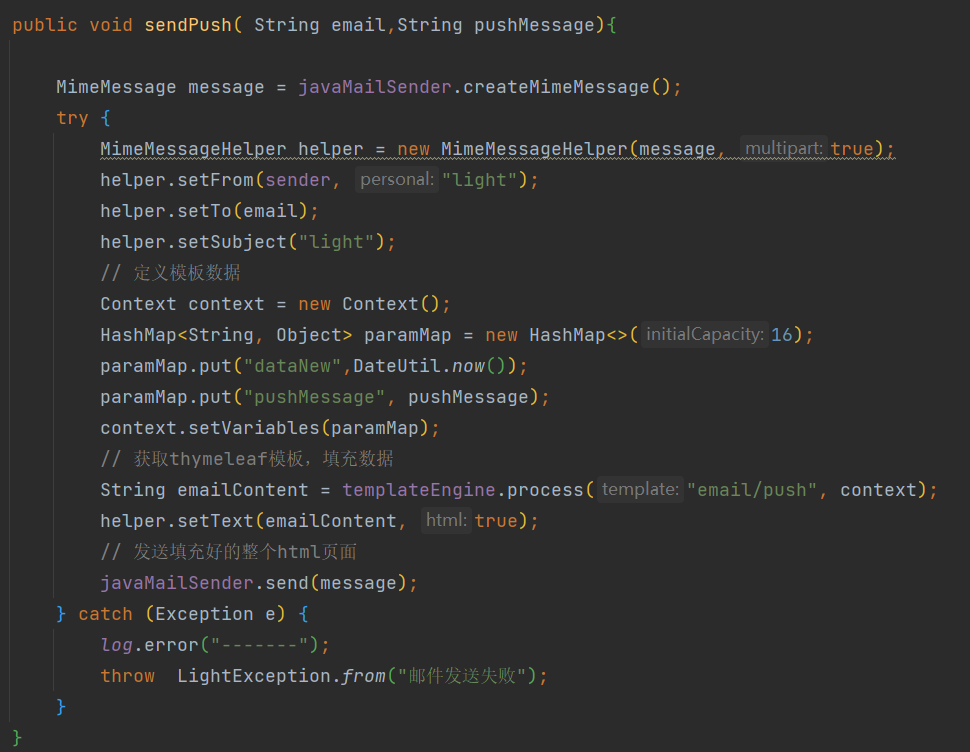


图5-40 邮件推送核心代码图

### 5.2.4　游戏管理功能的实现

管理员可以对游戏进行管理，对游戏进行增加，查看游戏的情况，删除，以及搜索，运行界面如图所示：

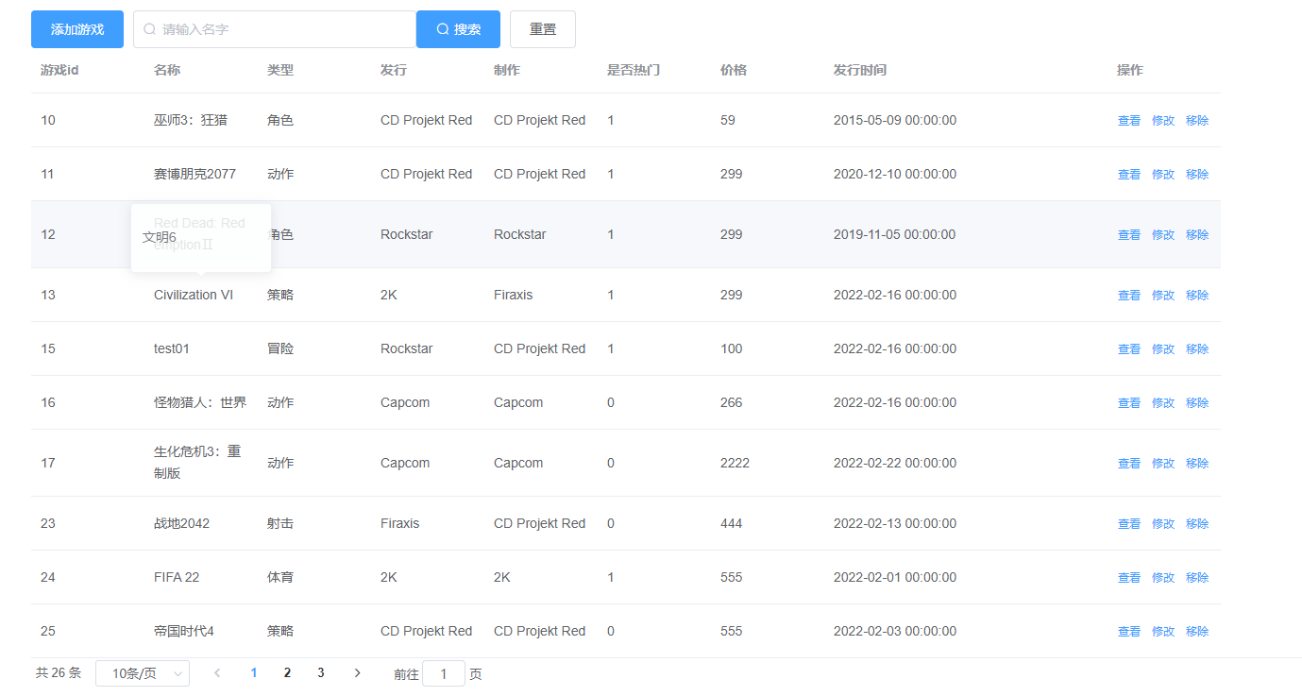


图5-41 游戏管理运行界面图

核心代码如下：

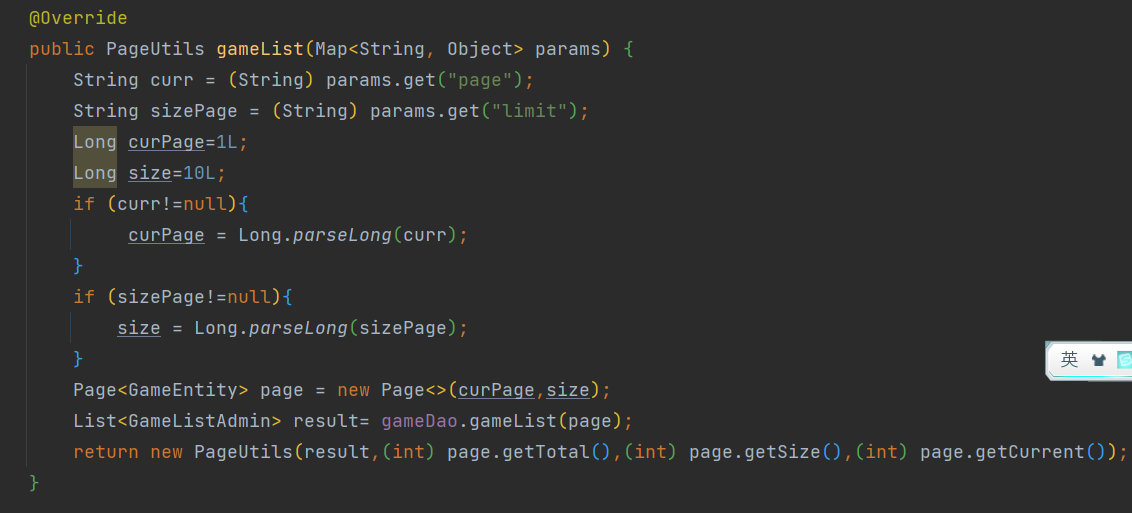


图5-42 游戏管理核心代码图

### 5.2.5　排行榜管理功能的实现

管理员可以对显示在首页的三大排行榜进行管理，在排行榜上添加游戏，删除一个游戏的排行，运行界面如图所示：

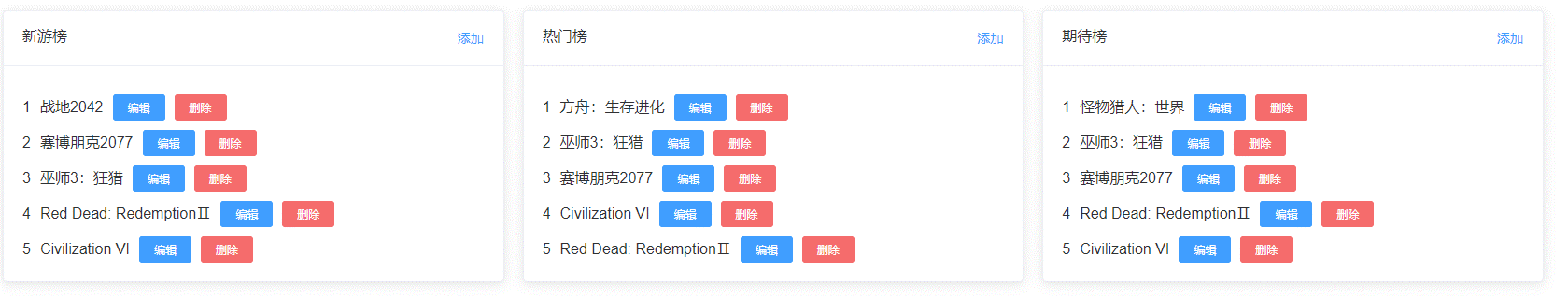


图5-43 游戏排行榜管理运行界面图

核心代码如下：

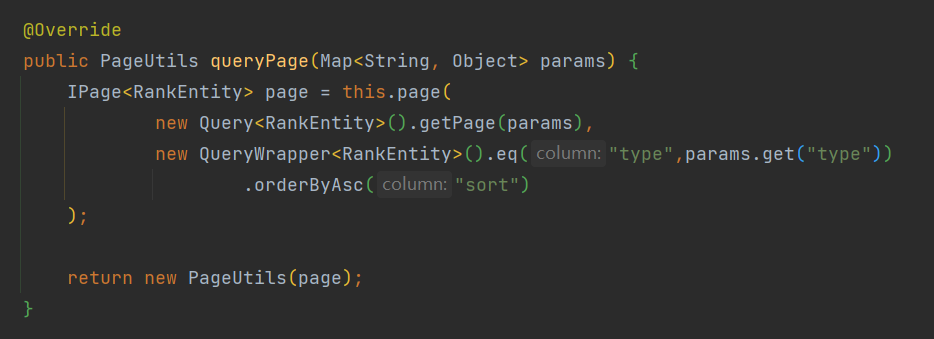


图5-44 游戏排行榜管理核心代码图

### 5.2.6　游戏轮播图管理功能的实现

在每个游戏的详情页会展示轮播图，管理员可以对这些轮播图进行修改，添加以及删除。运行界面如图所示：

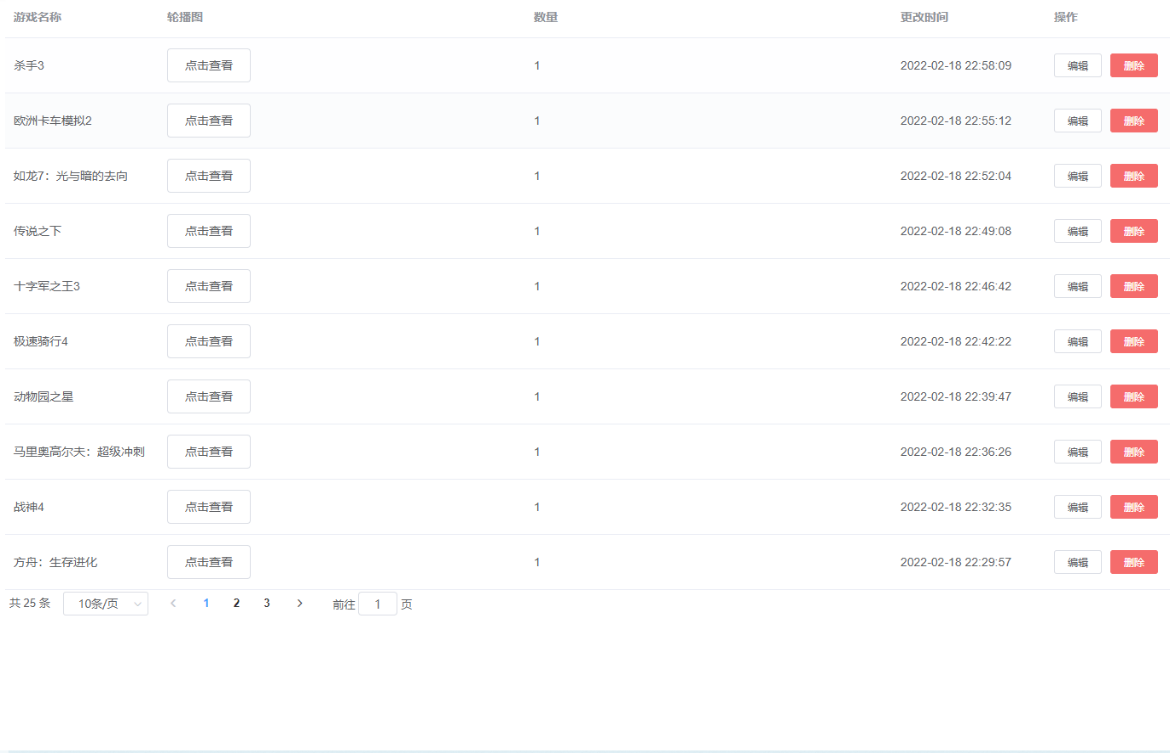


图5-45 游戏轮播图管理运行界面图

核心代码如下：



图5-46 游戏轮播图管理核心代码图

### 5.2.7　游戏评分功能的实现

管理员可以给游戏设置评分，并且为这个评分设置一篇文章，运行界面如图所示：



图5-47 游戏评分功能运行界面图

核心代码如下：



图5-48 游戏评分功能核心代码图

### 5.2.8　游戏标签功能的实现

管理员可以对游戏的标签进行管理，对游戏标签进行添加，修改以及删除，运行界面如图所示：



图5-49 游戏标签运行界面图

核心代码如下：

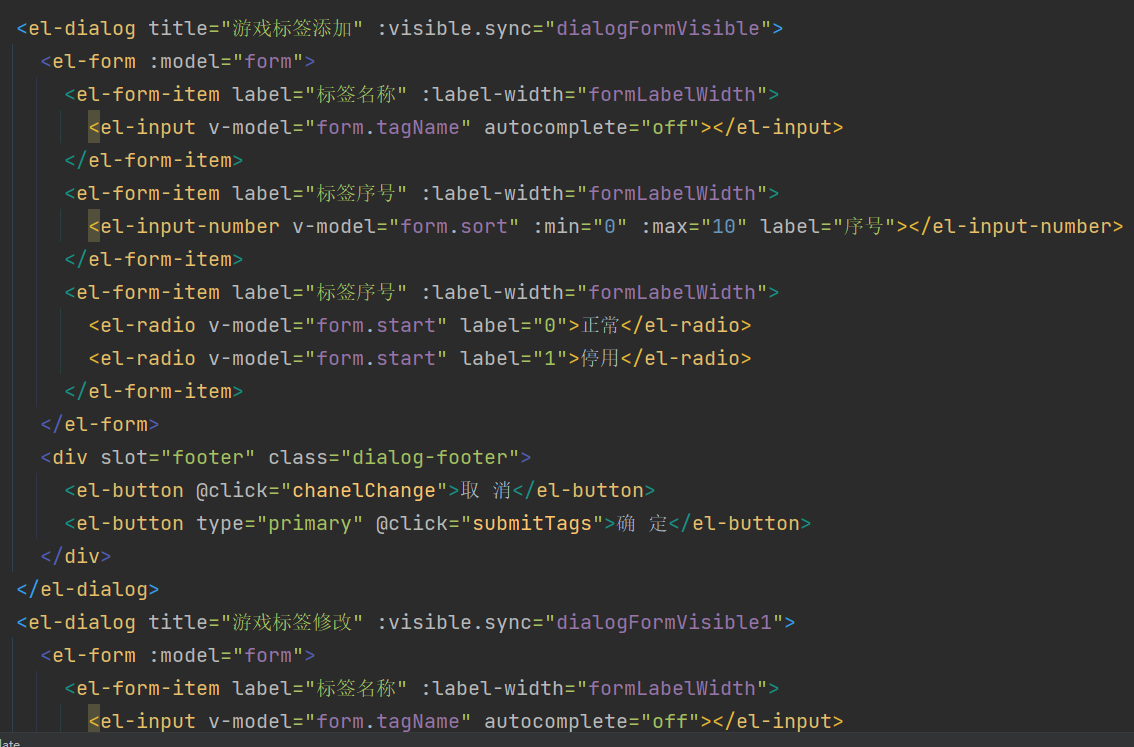


图5-50 游戏标签核心代码图

### 5.2.9　推荐游戏管理功能的实现

管理员可以对前台页面显示的推荐游戏进行管理，对游戏进行推荐，也可以把一个游戏从推荐当中去除运行界面如图所示：



图5-51 游戏推荐运行界面图

核心代码如下:

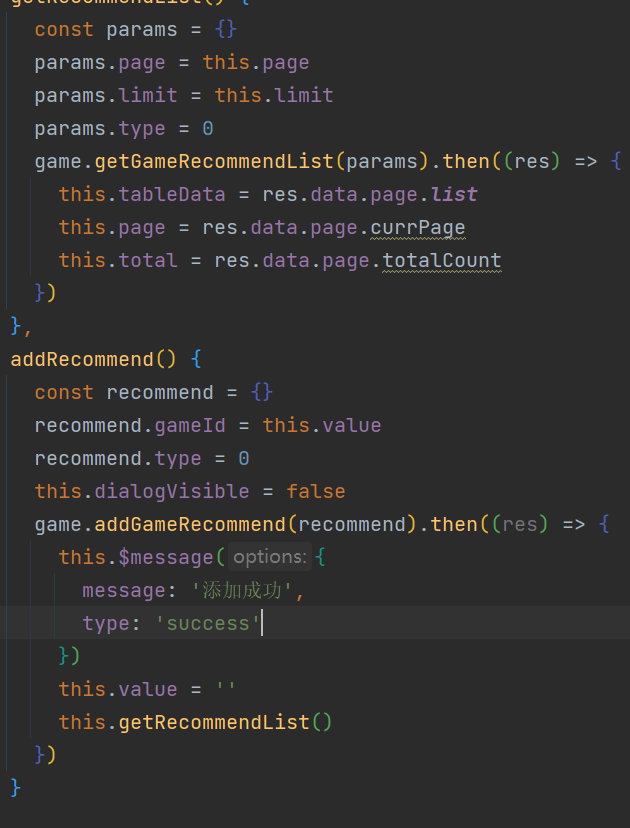


图5-52 游戏推荐核心代码图

### 5.2.10　设置游戏优惠功能的实现

管理员可以对游戏进行有优惠设置，如果一个游戏正在进行优惠，那么他就会前台页面显示，运行界面如图所示：



图5-53 游戏优惠运行界面图

核心代码如下:

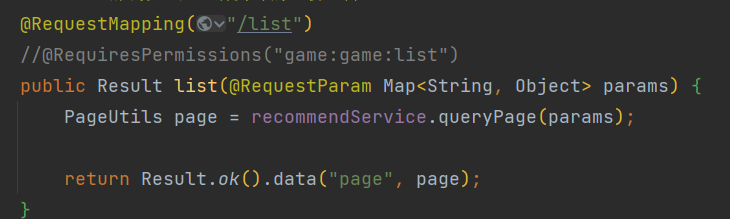


图5-54 游戏优惠核心代码图

### 5.2.11　游戏首页推荐管理功能的实现

管理员可以对首页的banner部分的游戏推荐进行管理，运行界面如图所示：



图5-55 游戏首页推荐管理运行界面图

核心代码如下:

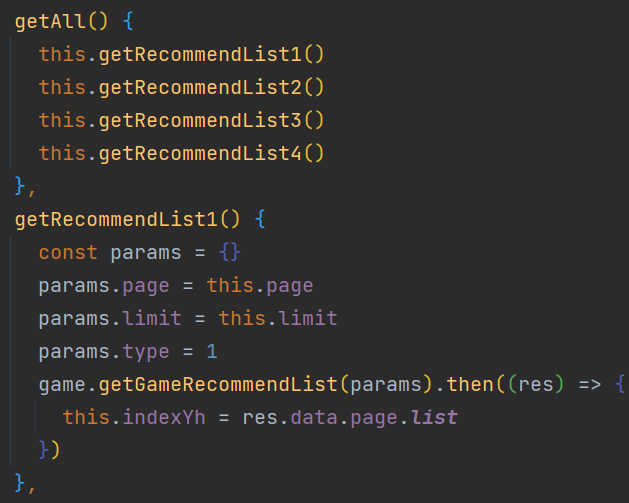


图5-56 游戏首页推荐管理核心代码图

### 5.2.12　游戏产商管理功能的实现

对于一个游戏来说，肯定有其制作商和发行商，在后台管理当中可以对游戏产商进行管理，运行界面如图所示：

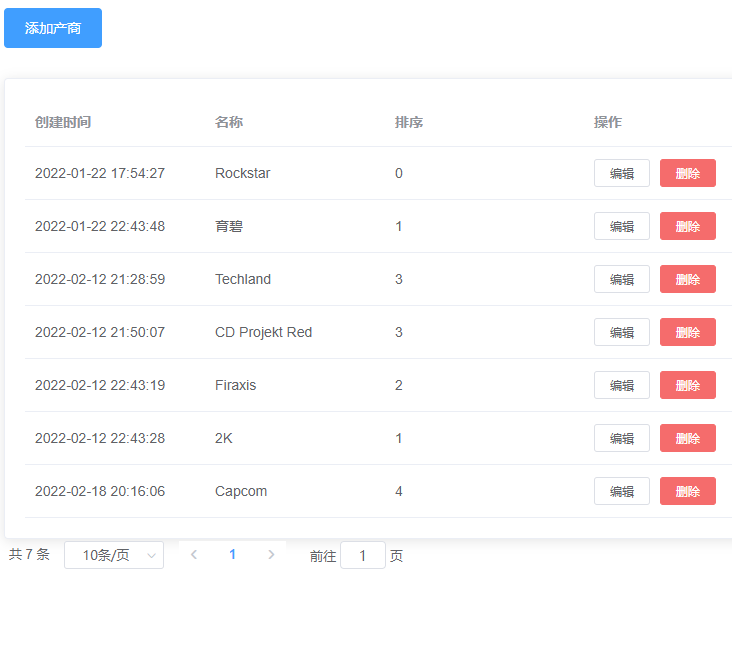


图5-57 游戏厂商管理运行界面图

核心代码如下:



图5-58 游戏厂商管理核心代码图

### 5.2.13　游戏类型管理功能的实现

对于一个游戏来说，游戏有着类型，一般来说游戏有着七大经典类型，但是本系统对其进行了动态的管理，运行界面如图所示：

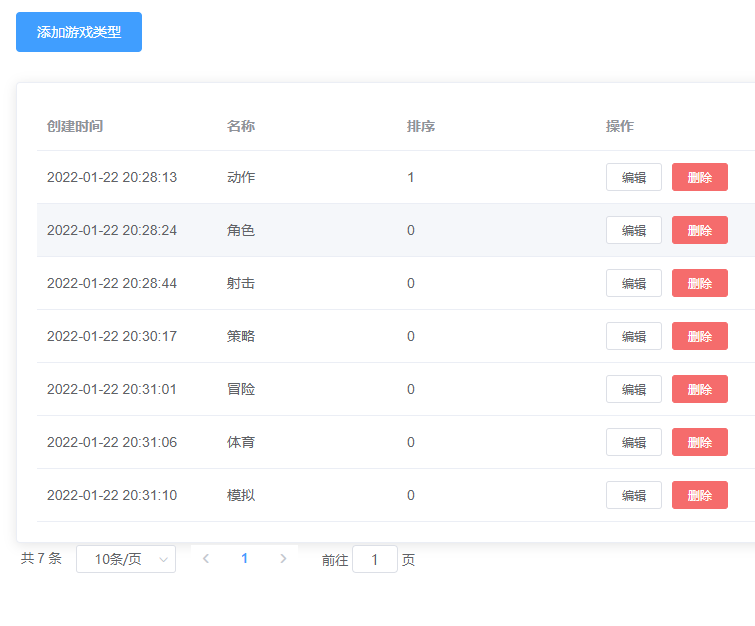


图5-59 游戏类型管理运行界面图

核心代码如下:

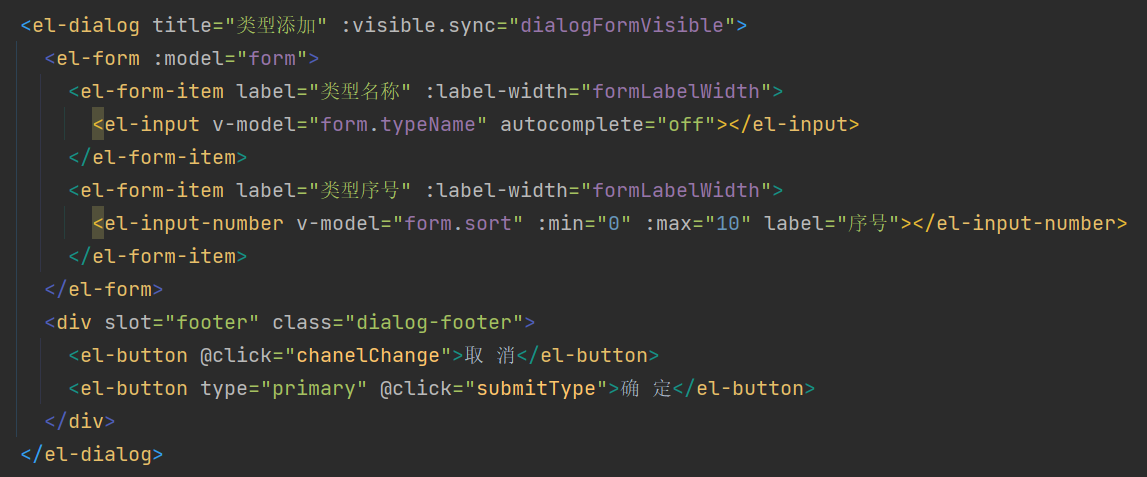


图5-60 游戏类型管理核心代码图

### 5.2.14　社区管理功能的实现

在本系统当中，每个游戏都有对应的社区，出了这些社区以为，还有其他类型的社区，管理员可以对这些社区进行修改，运行界面如图所示：

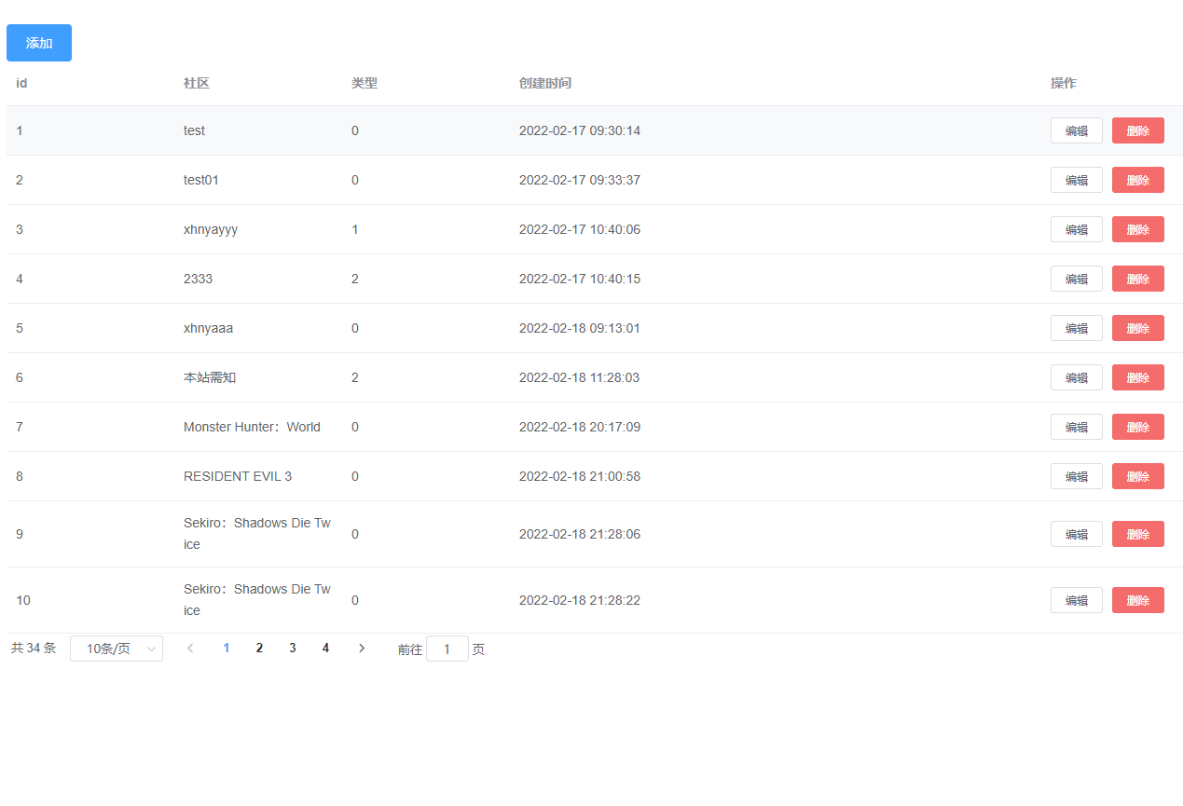


图5-61 社区管理运行界面图

核心代码如下:



图5-62 社区管理核心代码图

### 5.2.15　文章管理功能的实现

管理员可以对本系统所有的文章进行修改，删除，管理员还可以在后台添加文章，运行界面如图所示：

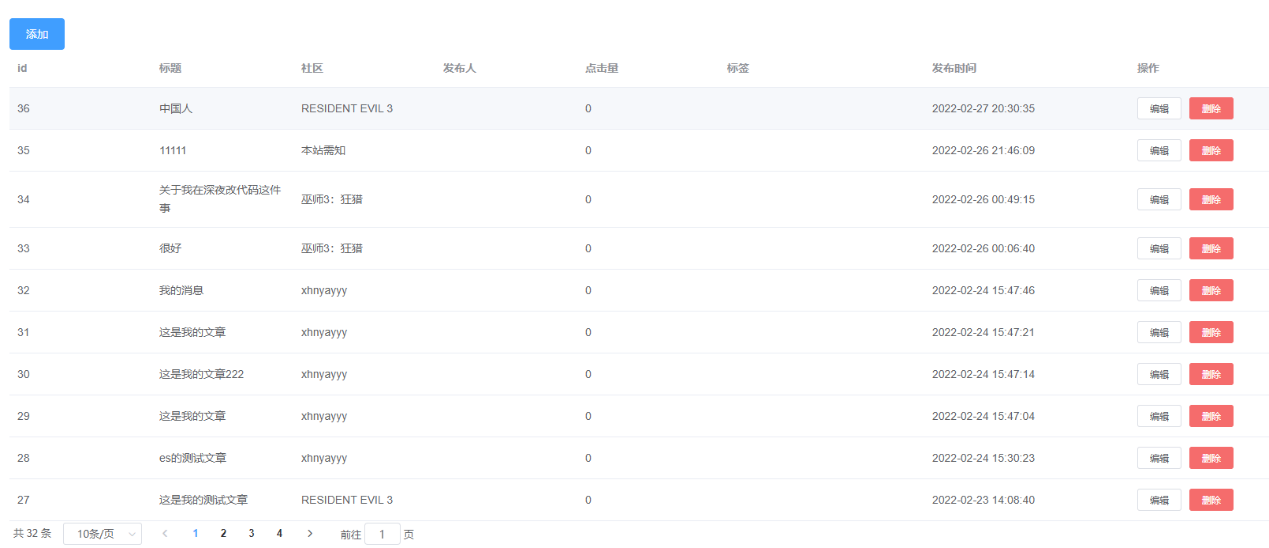


图5-63文章管理运行界面图

核心代码如下:



图5-64 文章管理核心代码图

### 5.2.16　用户管理功能的实现

对于在本系统中的用户，管理员可以进行管理，比如编辑，删除。运行界面如图所示：



图5-65用户管理运行界面图

核心代码如下:

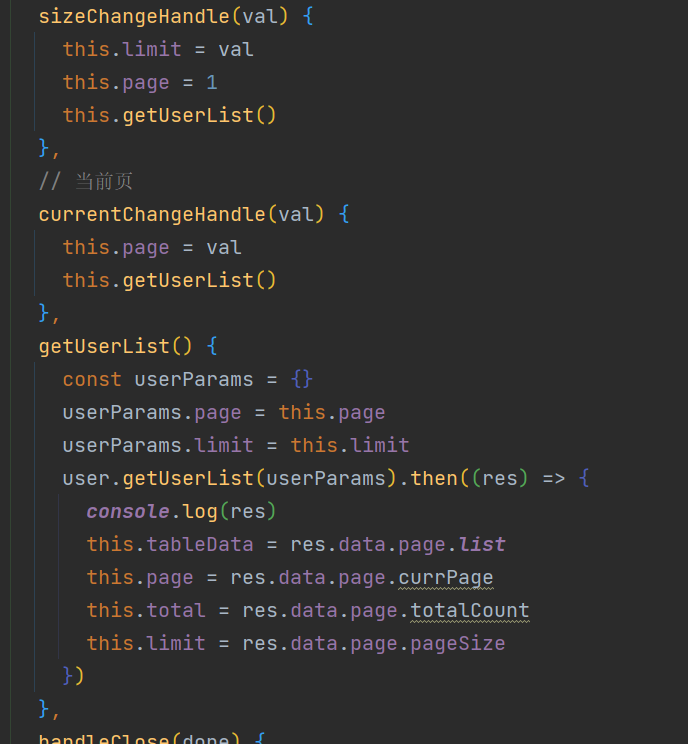


图5-66 用户管理核心代码图

## 5.3 运维模块的设计与实现

对于本模块来说，主要就是依托实现的技术说提供的可视化界面

### 5.3.1　elasticsearch管理功能的实现

可以通过访问kibana来对elasticsearch进行管理，管理里面的一些数据，运行界面如图所示：

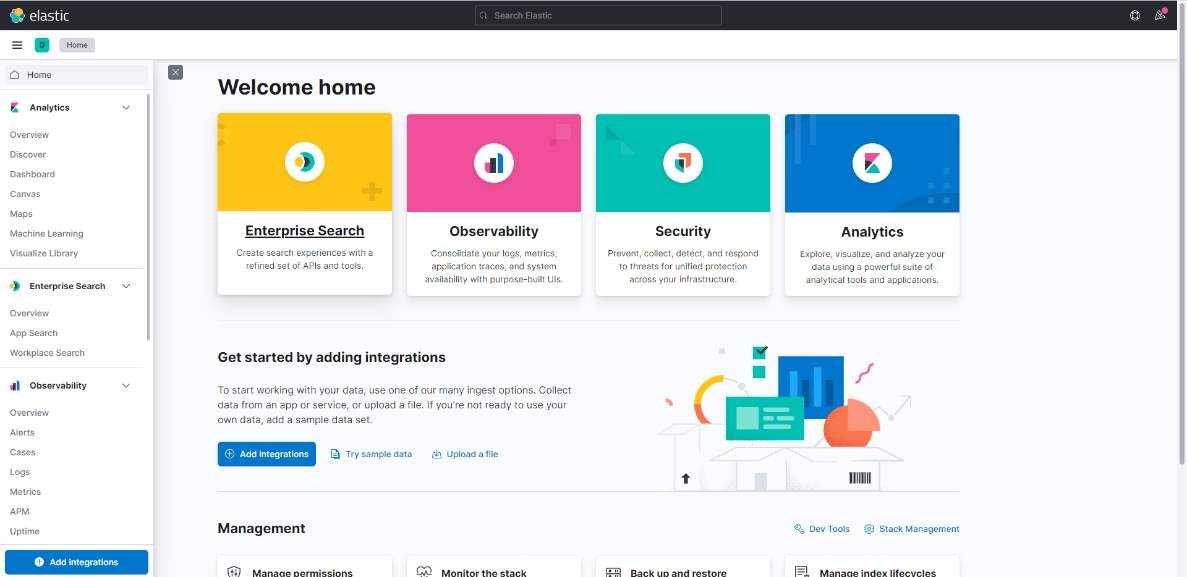


图5-67 Kibana运行界面图

### 5.3.2　nacos管理功能的实现

在本系统中采用的是nacos作为服务的注册中心和配置中心，相关人员可以在nacos对服务进行监控，以及修改配置文件，运行界面如图所示：

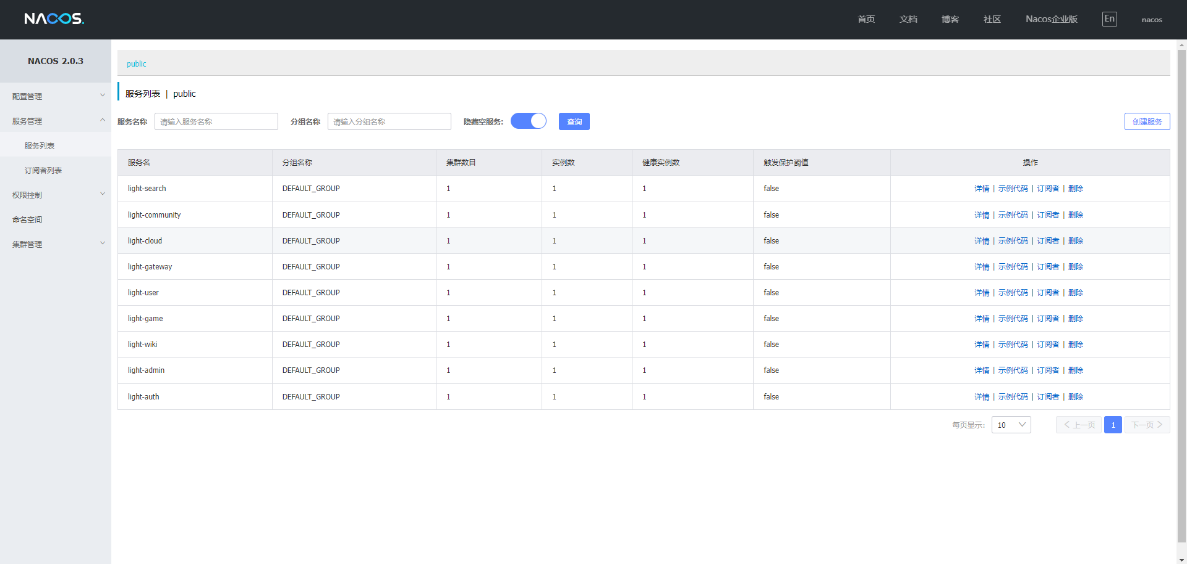


图5-68 Nacos运行界面图

### 5.3.3　sentinel管理功能的实现

在本系统中引入了sentinel对流量进行控制，相关人员可以在这里对系统进行热点，监控，降级等进行配置。运行界面如图所示：

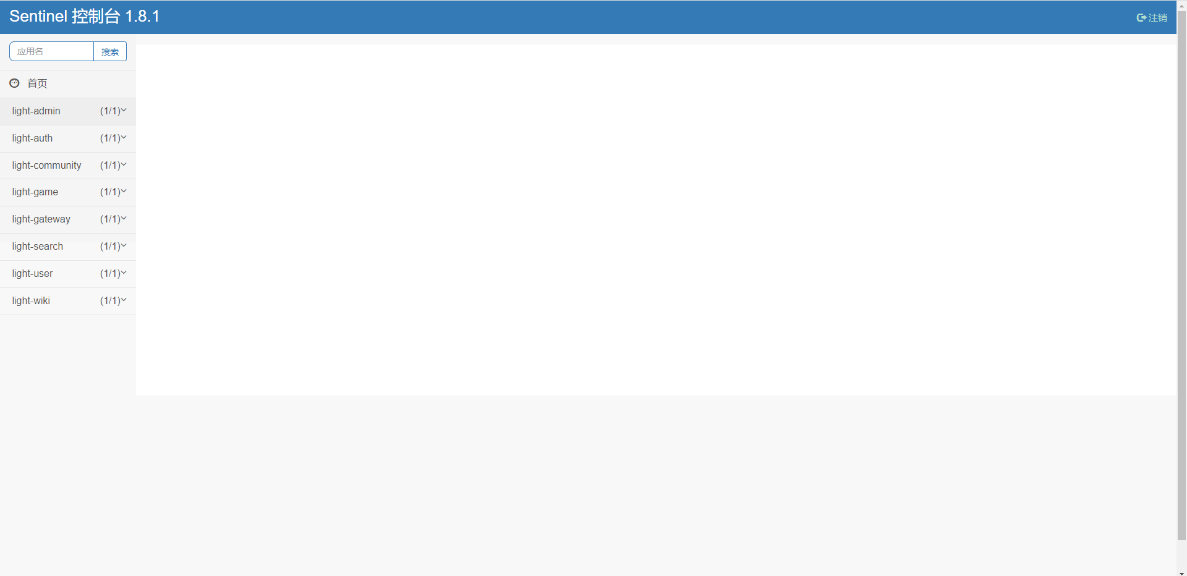


图5-69 Nacos运行界面图

# 第6章 系统总结与展望

## 6.1 系统总结

在本系统当中，使用到的技术比较的多，在分析系统的时候关于一些技术的选型，以及版本控制比较繁琐。但是在开发的过程当中一直遵循软件工程的设计思想，在数据库的设计上也遵循数据库的设计原则，相对来说在开发当中花费的时间是最多的，在设计上改动的是最多的。在建立搜索服务的时候，遇到的问题是最困难的，不过这也一步一步的走了过来，但是总体来说，对于这个系统本人还是不大满意的，因为时间的缘故，所以有很多的功能都没来得及完善，并且对于一些复杂的需要，完成的不是很好。现把相关的优点和缺点总结如下：

1.本系统的优点，本系统的主要开发环境都是运行在docker容器上面的，十分的易于扩展，本系统是一个一个的服务，在部署的时候也可以使用docker进行部署。并且在本系统当中引入了消息中间件，可以对整体的系统进行解耦。

2.本系统的缺点，有许多的功能并没有完善，对于一些推荐来说，应该是自动的进行推荐和热门管理，在用户方面，与用户的互动做的还是偏少部分

## 6.2 展望

在后续的开发中将继续完善系统的功能，引入一些推荐算法，热点计算等等，在用户消息的模块，实现私信等功能，建立起积分，等级对应的服务，并且对整个系统进行优化，总之，接下的任务还要继续，我也会不断的去完善与维护本系统。

# 参考文献

[1]倪小璐, 王旭英.微服务软件架构设计及其应用[J].杭州师范大学学报(自然科学版). 2021,20(04): 442-448.

[2]邓志强,邓林强.Maven在Java项目开发中的应用[J].电子元器件与信息技术,2019,3(05):1-4.

[3]Jiaqiao Xu.Programming Features and Technical Analysis of Computer Software Java[J].International Core Journal of EngineeringVolume 7, Issue 8. 2021.

[4]Ben Forta.Mysql必知必会.人民邮电出版社，2009

[5]Cay S.Horstmann.Core Java Volume I - Fundamentals [M].机械工业出版社 ，2020

[6]Cay S.Horstmann.Core Java Volume II – Advanced Features [M].机械工业出版社,2020

[7]Ravi Tomar,Sarishma Dangi. JavaScript:Syntax and Practices[M].CRC Press:2021-09-02.

[8]秦溧,艾青.基于Redis技术的页面缓存研究[J].科技风,2021(12):74-76.

[9]Matthew Tyson. What is an API gateway? API simplicity and stability[J]. InfoWorld.com,2021:

[10]姚刚,吴海莉,王从镔.浅析微服务架构API网关的作用[J].信息系统工程,2020(12):16-18.

[11]巢晟盛.基于SpringBoot微服务架构下前后端分离的MVVM模型浅析[J].电脑知识与技术,2021,17(23):128-129+141.

[12]魏榆静. 基于node.js的股票行情浏览及分析系统的设计与实现[D].山东大学,2021.

[13]方生.基于“Vue.js”前端框架技术的研究[J].电脑知识与技术,2021,17(19):59-60+64.

[14]方生.基于“MVVM”模式的“Web”前端的设计与实现[J].电脑知识与技术,2021,17(20):147-149.

[15]宋馨来,张海涛.基于前后端分离和Spring Security的用户登录功能的设计与实现[J].科技创新与应用,2021,11(18):19-21+24.

[16]杜恒.下一代云原生消息队列——Apache RocketMQ 5.0[J].软件和集成电路,2021(12):52-53.DOI:10.19609/j.cnki.cn10-1339/tn.2021.12.025.

[17]李强.消息中间件的生产端一致性设计与实战[J].电脑编程技巧与维护,2021(07):67-71.DOI:10.16184/j.cnki.comprg.2021.07.027.

[18]王帅. 一种高吞吐分布式消息中间件的设计与实现[D].东南大学,2020.DOI:10.27014/d.cnki.gdnau.2020.002732.

[19]谭畅,谭歆,胡磊,熊炜.云中心基于Nginx的动态权重负载均衡算法[J].重庆邮电大学学报(自然科学版),2021,33(06):991-998.

[20]陈思. 基于Nginx和Redis的高并发Web场景下缓存的研究与设计[D].东华理工大学,2021.

# 致 谢

xxx（学生姓名落款）

2022 年 4 月 10 日