

**本科毕业设计（论文）**

题目：基于k-means聚类算法的

新闻推送系统的设计与实现

学 院： 计算机学院

专业班级： 计算机科学与技术2017级1班

姓 名： 王琼弟

学 号： 201701060326

指导教师： 孙承爱

完成日期： 2021年6月6日

教务处制

**BACHELOR'S DEGREE THESIS OF SHANDONG UNIVERSITY**

**OF SCIENCE AND TECHNOLOGY**

**Based on k-means clustering algorithm**

**Design and Implementation of News Push System**

College：College of Computer Science and Engineering

Subject： Computer Science and Technology

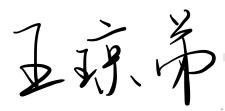
Name： Wang qiongdi

Directed by： Sun Chengai Professor

**QINGDAO CHINA**

**郑 重 声 明**

本人呈交的毕业论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本设计（论文）的研究成果不包含他人享有著作权的内容。对本设计（论文）所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本设计（论文）的知识产权归属于培养单位。



本人签名： 日期：

摘 要

随着互联网的飞速发展，网络世界进入了大数据的时代，互联网的信息数据量正以一种极其恐怖的速度增长，针对信息过载和用户兴趣不明确的问题，当下的互联网更需要一种推荐系统来代替传统的搜索引擎，从而帮助用户去寻找和发掘自己的兴趣，进而定位到感兴趣和想要的讯息。所以推荐系统的出现满足了当下人们面对海量数据的需求，用来连接用户和讯息，在大数据的时代非常具有研究价值。

系统综合运用UML建模技术，采用MVC三层软件体系架构，MySQL数据库，使用Redis作为缓存中间件，Nginx作为负载均衡中间件；使用Spring Boot + Vue前后端分离框架对新闻推荐系统进行搭建；给出了系统功能结构图、E-R图、流程图、顺序图、活动图、架构图、类图、系统用例图和组件图等，实现了用户的登陆注册、用户信息管理、新闻后台管理、新闻推荐、新闻展示和新闻爬虫等功能。

采用前后端分离模式，开发速度快，操作简便，具有实用性。在协同过滤算法的基础上利用k-means聚类算法进行改进，相比较传统的协同过滤算法，性能更优、推送列表的新闻分类更具多样性、推荐新闻不需要足够的用户行为数据和新闻数据、缓解了冷启动问题并且改进推荐的响应速度。

**关键词：**新闻推荐系统；协同过滤算法；k-means聚类算法；UML建模技术

**ABSTRACT**

With the rapid development of the Internet, the Internet world has entered the era of big data. The amount of information data on the Internet is growing at an extremely terrible speed. In view of the problem of information overload and unclear user interests, the current Internet needs a recommendation system to replace the traditional search engine, so as to help users find and explore their own interests, Then we can locate the information we are interested in and want. So the emergence of recommender system meets the current needs of people facing massive data, which is used to connect users and information. It has great research value in the era of big data.

The system uses UML modeling technology, MVC three-tier software architecture and MySQL database; Redis is used as cache middleware and nginx as load balancing middleware; Spring boot + Vue is used to build the news recommendation system; The system function structure diagram, E-R diagram, flow chart, sequence diagram, activity diagram, architecture diagram, class diagram, system use case diagram and component diagram are given. The functions of user login registration, user information management, news background management, news recommendation, news display and news crawler are realized;

Adopting the mode of front end and back end separation, the development speed is fast, the operation is simple and practical.On the basis of collaborative filtering algorithm, K-means clustering algorithm is used to improve. Compared with the traditional collaborative filtering algorithm, the performance is better, the news classification of push list is more diversified, the recommended news does not need enough user behavior data and news data, the cold start problem is alleviated, and the response speed of recommendation is improved.

**KEY WORDS:** News recommendation system; Collaborative filtering algorithm; K-means clustering algorithm; UML modeling technology

**目录**

[第1章 绪论 1](#_Toc8086)

[1.1 研究背景 5](#_Toc22629)

[1.2 研究开发现状分析 6](#_Toc30835)

[1.3 主要研究内容 8](#_Toc17099)

[1.4 本文的组织结构 9](#_Toc1643)

[第2章 相关技术综述 10](#_Toc11524)

[2.1 Spring Boot + Vue 前后端分离开发 10](#_Toc6268)

[2.2 Mahout 10](#_Toc23864)

[2.3 TF-IDF算法 10](#_Toc19148)

[2.4 K-Means聚类算法 11](#_Toc6952)

[2.5 典型推荐算法 11](#_Toc14719)

[2.6 本章小结 12](#_Toc24616)

[第3章 系统需求分析 13](#_Toc5970)

[3.1 业务分析 13](#_Toc8148)

[3.2 系统功能分析 15](#_Toc16035)

[3.3 系统非功能分析 17](#_Toc12413)

[3.4 本章小结 18](#_Toc21316)

[第4章 系统设计 19](#_Toc5543)

[4.1 系统总体架构设计 19](#_Toc15878)

[4.2 系统静态结构设计 20](#_Toc26435)

[4.3 业务用例的实现 21](#_Toc29103)

[4.4 数据库设计 22](#_Toc14449)

[4.5 类的详细设计 25](#_Toc28838)

[4.6 本章小结 26](#_Toc1488)

[第5章 系统实现 27](#_Toc4897)

[5.1 系统物理结构 27](#_Toc31659)

[5.2 系统主要功能实现 27](#_Toc27590)

[5.3 实现中遇到的问题和解决方法 31](#_Toc19643)

[5.4 本章小结 32](#_Toc19132)

[第6章 系统测试 33](#_Toc4106)

[6.1 单元测试 33](#_Toc23647)

[6.2 黑盒测试 33](#_Toc21526)

[6.3 接口测试 34](#_Toc637)

[6.4 本章小结 35](#_Toc29592)

[第7章 总结与展望 36](#_Toc13443)

[7.1 总结 36](#_Toc22904)

[7.2 展望 36](#_Toc1579)

**第1章 绪论**

**1.1** **研究背景**

随着计算机网络的飞速发展和互联网的普及，截止至2018年的11月，中国互联网累计在线用户已经超过了8亿人。移动通讯设备和网络技术的发展和实际应用为人们获取信息提供了非常大的帮助和便利，人们可以在任何地方任何时间获取自己想要的信息。但是正是因为互联网的快速发展，网络世界进入了大数据的时代，互联网的信息数据量正以一种恐怖的速度增长，人们每天都要面对海量的数据。

因此也进一步的引出了新的问题：如何从海量的数据中获取我们想要的讯息。在互联网的早期，人们拥有传统的搜索引擎，可以通过输入关键词来搜索信息，并且很快的定位到自己想要的讯息。但是这种方式只能满足人们明确的清楚自己想要的信息的情况，而对于不确定或者不知道自己喜好的用户，则不能有效的发挥作用。针对信息过载和用户兴趣不明确的问题，当下的互联网更需要一种推荐系统来代替传统的搜索引擎，从而帮助用户去寻找和发掘自己的兴趣，进而定位到感兴趣和想要的讯息。所以推荐系统的出现满足了当下人们面对海量数据的需求，用来连接用户和讯息，在大数据的时代非常具有研究价值。

发掘新闻阅读者的兴趣并且搜索到他们想要的新闻，对于系统的用户来说很有开发意义。推送系统作为一种讯息过滤的技术，相比较传统的搜索引擎，它更具有主动性，当使用者无法准确描述自己的需求时，推送系统也能够为用户提供精确的推送，从使用者的角度看，是一种有效的解决信息过载的方式，通过分析用户和物品的数据，对用户和物品的行为数据进行建模，从而主动地为用户进行推送。

发掘更多的“长尾”信息，对于系统的开发者和商家来说也具有开发意义。大部分的用户或者消费者的需求总是集中在热门的内容或者主要的商品上，并不在意那些大量存在且相对冷门的“长尾”信息和商品。

根据《长尾理论综述》中所提到的：“只要存储和畅达的渠道足够大，需求不旺或销量不佳的产品共同占有的市场份额就可以和那些数量不多的热卖品所占据的市场份额相匹配甚至更大，这就是长尾理论。”[1]

对于长尾信息和数据的发掘，可以给网站开发人员和持有者带来意想不到的收益，冷门新闻的阅读总和可以给网站带来超过其本身的访问量。

大量的用户和大量的新闻信息引发了人们对新闻网站功能的进一步扩大完善。面对海量的数据，用户需要的是快速地定位自己需要的新闻信息，具有很大的实用价值和研究意义。

**1.2** **研究开发现状分析**

推荐系统的研究最早开始于上个世纪的90年代，推荐算法是推荐系统中最重要的部分，一般分为基于内容的过滤算法，基于用户行为的协同过滤算法和混合算法。

加州大学的Pazzani等人研究了基于内容的推荐系统，施乐公司的Palo Alto完成邮件推荐系统Tapestry,是第一个基于协同过滤技术实现的系统，明尼苏达大学的GroupLens研究团队完成了基于自动协作过滤的推荐系统。[2]

2001年，著名电子商务网站亚马逊将推送系统运用到其网站中，个性化推送开始从学术研究向实际应用中迈进。

2016年Hernando等人提出使用矩阵分解的方法。LeeK提出了基于图的推荐算法。除此之外，Huynh、Bellogin、chen等人提出了使用KNN、归一化切割、交替最小二乘法等方法。[2]

与国外推荐系统相比，直至2000年左右，推送系统才引起国内的关注。国内推荐系统的应用和研究起步相对较晚，但也逐渐成为计算机领域研究的热点。

我国最早在推荐方向和领域中进行研究和总结的是清华大学的路海明教授，他的团队提出了基于Agent的个性化主动信息服务系统。[2]

清华大学曾春等人对个性化服务技术进行了全面的综述性研究，总结了个性化推荐技术的体系结构，集中讨论了三种推荐算法所涉及的核心技术，对国内的推荐系统研究具有一定的指导意义。[2]

当下新闻推荐系统开发具有如下的重难点问题：

（1）冷启动问题

在系统启动的初期，由于系统中用户的行为数据较少，并且存储很少的新用户的行为信息，模型训练就无法获取准确的训练效果，无法基于规则从而进行准确的推理。

冷启动问题又包括用户冷启动和物品冷启动问题。

针对用户冷启动问题，常见的思路有：1.可以通过热门推荐的结果来进行补充。2.获取更多用户本身自带的信息，如用户的ip地址，交互行为发生的时间等数据，进行充分的利用。3.对新用户设置喜好标签的选择，针对喜好的标签来进行初期的推荐。

针对物品冷启动问题，通常采用基于内容的方式，采用TF-IDF算法根据物品与物品之间的相似度进行推荐。虽然基于内容的方式作为推荐算法来说精度不是很高，但是对于解决冷启动问题却是非常的有效。

在实际开发的过程中往往都是结合多种方法混合进行处理，从而达到在某个业务场景下的最优解。

（2）大数据处理和增量问题

当下的互联网正处在一个大数据的时代，每天都会产生数以千万的新闻。在百千万级别的新闻系统基础上，每个新用户的注册、登录和交互行为和每个新新闻的添加都会给新闻推荐系统带来巨大的影响，每一次的变更都需要进行重新运算。在大数据量的情况下，如何保证推荐的时效性和有效性，如何保证系统的稳定性是个巨大的问题。

某些实时推荐在大数据下也面临挑战，在大数据的背景之下，一次推荐模块的运算需要花费大量的时间，用户通常难以忍受超过1秒的推荐响应速度。

针对大数据的问题，通常采用分布式文件系统来管理数据，Hadoop的分布式文件系统架构是其中的典型。采用基于集群技术的分布式计算框架，利用mahout实现基于协同过滤的推荐算法。在数据量大的情况下，将推荐算法进行分布式并行化，使得推荐算法能够在海量的数据环境下高效运行。

（3）数据稀疏性

新闻推荐系统用户通常只阅读新闻而不进行评分和收藏，也就是用户的行为数据，用户和物品的交互行为非常的少。在大数据的背景之下，数据的稀疏性问题则更为明显，当下大多数的推荐算法都是采用用户和物品的行为或者评分矩阵进行计算，而用户行为数据量的增长速度却远远低于网站的新闻数据量的增长速度，导致用户-行为矩阵越来越稀疏，从而会导致推荐算法的准确度大大降低。

针对该问题有以下几种解决方案：1.采用基于项目的协同过滤推荐，由于项目与项目之间的相似度通常较为稳定，而项目的数量相比较用户数量来说大大减少，因此可以降低矩阵的稀疏程度。2.降低矩阵的维度技术，将用户新闻矩阵进行分解，抽出主要的信息。3.对用户进行聚类，使得每一个簇都具有共同属性，在小的簇中进行推荐，从而也能减小矩阵的稀疏程度，也加快了计算速度。

（4）特征提取问题

推荐系统种类繁多，推荐的对象也是千奇百怪。针对不同的对象，不同的特征提取方式就是一项难题。针对文本类对象，如新闻、博客等，可以借助问题提取技术如jieba分词和TF-IDF技术来提取文本中的关键词，这些关键词就代表了文本的特征。对于一些多媒体对象如视频、图像、音乐等对象目前还缺乏特别有效的解决方案。

（5）效率问题

对于一些实时性要求很高的在线推荐系统，由于数据量的不断提高，效率大大降低，从而需要花费大量的时间才能得到推荐结果，用户和用户之间或者物品与物品之间的相似度的计算，机器学习算法，求解全局最优解问题等都需要花费大量的时间，这对于实时推荐来说是难以接受的。针对效率问题，通常采用的方法是离线计算，然后线上只需要获取离线的数据进行推荐即可。还有一种方法就是增量SVD协同过滤，当有新的评分加入时，不需要对整个矩阵进行计算，只需要进行局部调整，从而大幅度提高矩阵计算的速度。

**1.3** **主要研究内容**

（1）缓解了冷启动问题，即当用户行为数据较小的情况下的冷启动问题，利用用户刚注册时选择标签，从标签中推送热门新闻。

（2）缓解了大数据量的问题，即当用户行为数据非常大的情况下，推送算法响应过慢的问题，利用将新闻推送数据和热门数据存到redis中，提高了推送的速度和用户的体验，与此同时利用K-means算法对用户进行聚类，缩小协同过滤的范围，从而提高协同过滤的速度。

（3）搭建了Spring Boot + Vue的前后端分离框架，在此基础上还了解了新闻推送系统常见的板块和功能实现并且实现了用户登录、用户注册、用户信息查询、新闻分页、具体新闻、新闻评分、新闻收藏、推送新闻等功能，并将推送模块和前后端显示模块两个模块进行了整合。

（4）使用BeautifulSoup爬虫来爬取新闻数据，并且进行数据处理和整理，最后根据数据集来设计数据库，设置了定时器和衰退指数，可以自动的定时定期的爬取新闻和删除过期的新闻。

（5）利用Mahout实现了基于用户行为和新闻内容相似度的协同过滤算法，在此基础上实现了基于K-means的推送算法。

**1.4** **本文的组织结构**

第1章 绪论。主要介绍和分析了研究的动机、研究的理由和背景、国内外现状、目的和意义以及主要的研究内容，并对论文的组织结构进行了分析和概述。

第2章 相关技术和工具综述。具体阐述了项目搭建过程中所使用到的主要技术、理论和方法。主要包括Spring Boot + Vue 前后端分离开发技术的原理概述、Mahout工具介绍、TF-IDF算法概述、K-Means聚类算法概述以及常见的推荐算法概述。

第3章 系统需求分析。阐述了系统的需求分析过程，包括业务分析、系统功能分析和系统非功能分析。通过系统功能结构图、E-R图、流程图、顺序图、活动图、类图和系统用例图等uml图的形式来描述系统的功能属性和非功能属性。

第4章 系统设计。通过架构图的形式描述系统的整体架构，介绍了系统的总体设计，包括数据库设计，类设计和静态结构设计。

第5章 系统实现。阐述了系统的详细设计过程，通过组件图的形式描述系统的物理结构，对系统所具有的功能做了详细的阐述，展示系统实现的关键代码的实现技术以及系统运行的界面等。

第6章 系统测试。通过单元测试、黑盒测试和接口测试等方法对系统进行了测试。

第7章 总结与展望。对系统的开发过程和本次毕业设计进行了分析，总结和展望。阐述自己工作的优点在哪里、不足之处在哪里、进一步的改进方案如何以及比较与他人工作的优、劣等。

**第2章 相关技术综述**

**2.1** **Spring Boot + Vue 前后端分离开发**

Spring Boot是由众多的开源框架进行整合的一个框架，在Spring框架的基础上针对业务逻辑而设计的一个升华版本。Spring Boot使得用户摆脱了传统开发模式中需要大量项目配置的工作，使得系统的开发人员更加关注于业务逻辑和功能的完成和实现，而不需要把精力花费在繁琐而千篇一律的配置上去，简化了开发人员的开发流程，实现项目的快速部署[4]。

Vue.js是一个前端的开发框架，一个通过逐步构建用户界面的渐进式框架。Vue.js将代码的编写和视图的展示进行分离，使得代码更加清晰，提高了后期的可维护性和可扩展性，利用双向绑定和视图组件的配合使用提供尽可能简单的API接口，能够更好更简单地进行web应用开发[5]。

Spring Boot + Vue前后端分离框架开发指的是利用Vue开发系统的前端页面，利用Spring Boot搭建系统的后端，两者各自独立部署。前端利用axios来与后端建立http连接，调用后端提供的接口，获取数据。Vue通过双向绑定将数据渲染到系统页面上去。后端Java只需要提供相应的接口返回Json数据，然后专注于业务逻辑的实现即可，前端只需要调整页面的样式和数据的展示，前端和后端必须定义好一致的接口，可以进行同步开发。前后端分离的开发模式使得项目开发的更加简便迅速，工作人员开发思路更加清晰。

**2.2** **Mahout**

Apache Mahout 是 Apache Software Foundation（ASF）开发的一个开源的项目，主要应用于大数据集下的机器学习领域的经典算法的实现。Mahout主要的核心在于实现推荐系统中的协同过滤，聚类和分类。通过Mahout可以实现基于物品的协同过滤推荐和基于用户的协同过滤推荐，也可以实现常见的聚类算法，并且提供了许多处理数据，提取特征，训练算法模型等方法，满足用户为了实际运用情况的二次开发需求[6]。

**2.3** **TF-IDF算法**

TF-IDF是一种关键词提取的技术，目前应用最多的是基于统计信息的方法。TF是指词语在文档里面出现的频率，IDF代表了某个单词的重要程度，具体表现为IDF越高，该单词在某一个文本中出现的频率越高，而在其他的文本中出现的频率越低。通过对TF和IDF两个数值进行加权处理并且统计分析，来提取一个文档中的关键词。

但是传统的TF-IDF算法具有一些不足，它并没有考虑到一些新出现的单词，这些单词具有特殊性，会对提取关键词的准确度造成巨大的影响。在此基础上可以通过特征权重算法进行改进，特定情境下的词可以赋予更高的权重来优化提取关键词的准确程度。

**2.4** **K-Means聚类算法**

K-Means聚类算法是聚类算法中当下最流行的算法，该算法以效果好、思想简单以及实现简单的种种优点得到了大量的应用。

K-Means算法采用无监督的学习方式，由于欧氏距离运算简单，K-Means聚类算法通常利用欧式距离作为划分簇与簇以及对象之间的相似度的标准，距离越近相似度越高。

K-Means算法的核心思想是：先定义一个K值，随机选取K个簇中心，旨在划分为K个簇。计算所有的数据与簇中心的欧式距离，将其划分到与其距离最近的簇类当中去。然后不断进行迭代，每次重新计算每个簇的平均值，然后建立新的簇中心，当簇中心不再变化或者距离小于某个设定值或达到最大的设定的迭代次数就停止[7]。

**2.5 典型推荐算法**

（1）基于内容的推荐算法

基于内容的推荐算法通常运用于文本类的推荐系统，如新闻和图书等。算法的主要内容在于计算文本和文本之间的相似度，构建用户特征然后计算项目间的相似度进行个性化的推荐。

基于内容的推荐算法通常包含以下三步：

首先利用TF-IDF等算法为每个新闻提取关键词和特征来代表这个新闻。

然后利用一个用户过去喜好的新闻进行特征学习，来推算出用户的特征喜好。

最后通过用户的特征喜好和候选的新闻列表，为用户推荐一组相似度最高的新闻列表。

基于内容的推荐算法具有用户独立性高、天生约束冷启动问题、不存在马太效应等优点；相应地，新用户在没有行为数据的情况下无法进行基于内容的推荐，无法挖掘用户的潜在兴趣，并且推荐粒度过粗，缺乏多样性[8]。

（2）基于协同过滤的推荐算法

协同过滤推荐算法是一种信息过滤的算法，在新闻推荐系统中应用最为广泛，协同过滤包括基于物品和基于用户两种推荐算法，本质上都是基于用户和物品之间的交互行为来进行过滤。

基于用户的协同过滤推荐算法的主要思路是：先构建所有用户评分数据集作为输入，然后给出当前的用户ID，找出当前用户和用户评分数据集中有着相似喜好的其他用户作为最近邻，在最近邻中挑选当前用户没看过的评分最高的TopN个新闻给当前用户。

基于物品的协同过滤推荐算法的主要思路是：先构建物品相似度矩阵，针对当前用户偏好的物品从相似度矩阵中计算出相似度最高的TopN个新闻给当前用户。

协同过滤推荐算法可以挖掘用户的潜在信息，对于复杂的对象具有通用的解决模型，相应地，协同过滤本身具有数据稀疏性和冷启动等当下推荐系统最核心的问题并且推荐的可解释较低[8]。

（3）用聚类算法改进协同过滤的推荐算法

基于聚类算法的协同过滤与以往的基于用户或项目的协同过滤相似，可以按用户或距离度量对对象进行聚类，如果基于用户聚类，则可以根据一定的距离将用户分为不同的目标组。在文章聚类的基础上，将相似的高得分文章推荐给用户。常用的聚类推荐算法有K-Means, BIRCH, DBSCAN和谱聚类，而本文则采用K-Means聚类来改进协同过滤推荐算法。

**2.6 本章小结**

首先介绍了项目的总体搭建框架Spring Bot + Vue，并介绍了该前后端分离框架的概述和优点。介绍了新闻推荐过程中所需要的算法，包括TF-IDF算法、K-Means算法和常见的推荐算法。介绍了实现协同过滤算法的开源项目Mahout。

**第3章 系统需求分析**

**3.1** **业务分析**

**3.1.1** **业务范围**

普通用户的用例包含个人维护用例、收藏用例、评分用例、查询用例。

个人维护用例又包含登录、注册、注销、修改用户信息、修改密码等。

查询用例包含查询收藏记录、查询评分记录、查询每日新闻、查询推荐新闻、查询热门新闻等。

普通用户的用例图如图3.1所示：

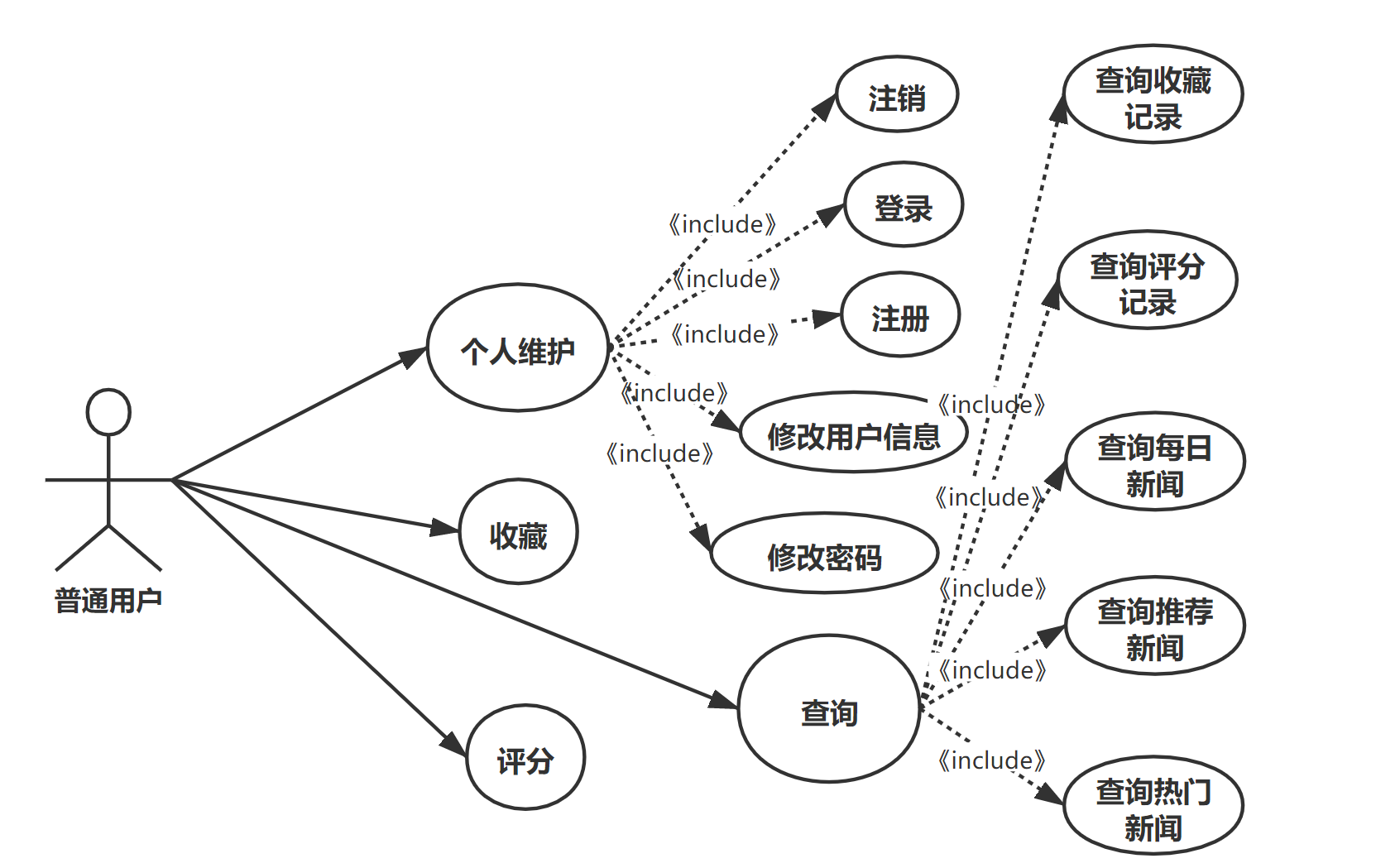


图3.1 普通用户的用例图

管理员的用例包含用户权限管理用例、新闻管理用例、推荐管理用例。

新闻管理用例包含新闻定时清理、爬取新闻、修改新闻、删除新闻。

推荐管理用例包含实时推荐和离线推荐。

管理员的用例图如图3.2所示：

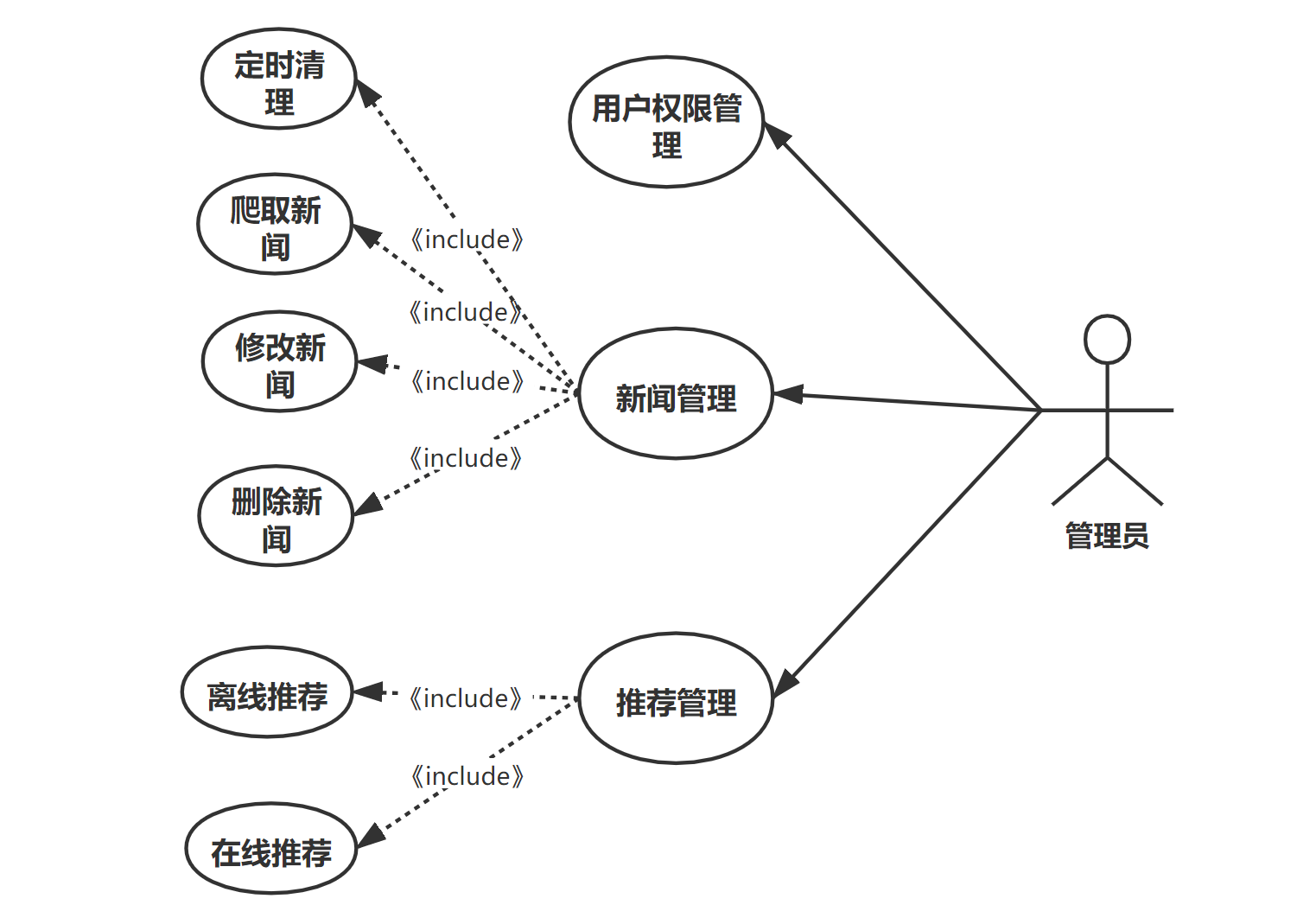


图3.2 管理员的用例图

**3.1.2 核心业务流程**

首先爬取新闻到数据库当中，用户进行注册登录，登陆后显示所有新闻的分页显示；若用户是首次注册则会显示选择标签信息，用户选择喜欢的标签，系统根据标签更新推荐的新闻；用户选择喜欢阅读的新闻，并且进行评分和收藏，系统根据评分和收藏结合衰退指数对推荐新闻进行更新。

普通用户的活动图如图3.3所示：

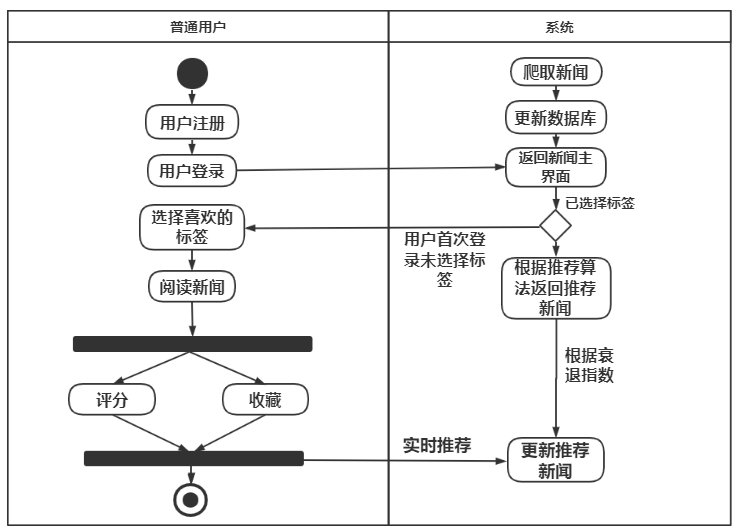


图3.3 普通用户的活动图

**3.2** **系统功能分析**

**3.2.1** **系统功能要求**

系统主要应用Java语言与MySQL数据库语言实现了用户的登陆注册、用户信息管理、新闻后台管理、新闻推荐、新闻展示和新闻爬虫等功能。系统具体的功能模块如下：

（1）新闻展示模块：包括新闻分页展示、具体新闻展示等。

（2）新闻爬取模块：包括具体新闻爬取、新闻分类爬取等。

（3）新闻后台管理模块：包括新闻管理、新闻类别管理、评分管理、收藏管理、用户管理、权限管理等。

（4）用户模块：包括用户登录、用户注册、用户信息修改、用户信息展示、用户评分收藏历史展示等。

（5）新闻推荐模块：包括基于物品的协同过滤推荐、基于用户的协同过滤推荐、基于K-Means的推荐、猜你喜欢等。

系统功能模块图如图3.4所示：

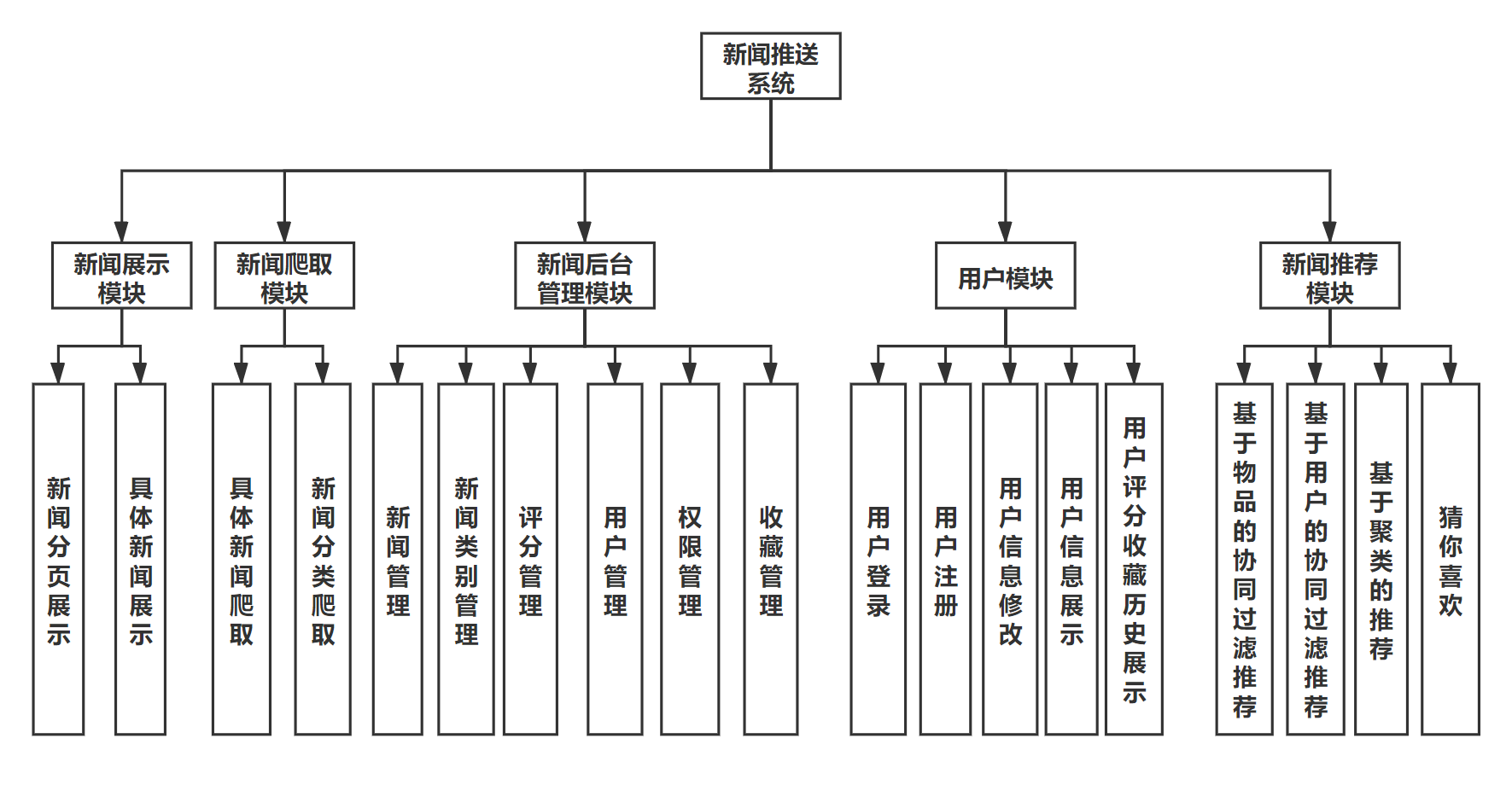


图3.4 系统功能模块图

**3.2.2** **主要用例规约**

表3.1 查询推送新闻用例

|  |  |
| --- | --- |
| 用例名称 | 查询推送新闻用例 |
| 用例编号 | REC001 |
| 执行者 | 普通用户 |
| 用例描述 | 该用例规定了普通用户如何利用系统进行查询推送新闻过程 |
| 涉众及兴趣 | 普通用户：希望能够准确的获取到自己想要阅读的新闻，并且能够发掘自身的兴趣。  管理员：希望能够方便快速的选择推送算法。 |
| 前置条件 | 用户已经登录并且选择了自己喜欢的标签。 |
| 后置条件 | 存储用户的评分收藏行为，根据用户的行为更新推送算法，返回更新的推送新闻。 |
| 基本流程 | 1.管理员选择合适的推送算法。  2.普通用户注册，登录并且选择喜欢的标签。  3.系统记录用户信息，包括用户姓名、用户密码、用户喜欢的标签。  4.系统根据推送算法和用户喜欢的标签返回推送新闻。  5.普通用户点击想要阅读的新闻。  6.系统返回单个新闻的数据。  7.普通用户进行评分和收藏。  8.系统根据用户的行为实时的根据推送算法进行更新，并且实时返回推送新闻。  9.循环反复5-8。  10.普通用户注销。 |
| 扩展流程 | 2a.用户注册登录输入信息不合法。  1.拒绝输入并返回错误信息。  3a.数据库出现异常。  1.回滚至系统一致性状态并提醒用户操作未记录。 |
| 非功能需求 | 1.所有用户的点击都应该在0.5秒内得到响应。  2.所有推送都应该在1秒内得出结果。 |

表3.2 新闻爬虫的用例规约

|  |  |
| --- | --- |
| 用例名称 | 新闻爬虫用例 |
| 用例编号 | REC002 |
| 执行者 | 管理员 |
| 用例描述 | 该用例规定了管理员如何利用系统进行新闻爬取过程 |
| 涉众及兴趣 | 管理员：希望能够方便快速的进行新闻爬取功能并且存入数据库当中 |
| 前置条件 | 管理员已经获得了相应的爬取权限 |
| 后置条件 | 将爬取到的新闻存入到数据库当中 |
| 基本流程 | 1.管理员选择要爬取的新闻网页。  2.管理员选择爬取新闻的类别。  3.系统显示爬取的新闻信息，包括新闻标题、新闻内容、新闻类别、新闻创建时间、新闻来源、新闻连接。  4.系统将爬取到的新闻存入到数据库中。  5.反复重复1-4。 |
| 扩展流程 | 1a.爬取的新闻网页url不合法。  1.拒绝输入并返回错误信息。  2a.选择新闻类别在数据库中不存在。  1.拒绝执行并返回错误信息。 |
| 非功能需求 | 1.能够爬取中国新闻网近一周的所有新闻。 |

**3.3** **系统非功能分析**

（1）易用性

易用性表示用户在使用该软件系统时需要付出的努力和代价，包括使用的难易程度，记忆系统使用方法的难易程度，针对于刚刚使用该系统的用户所需要满足的一个属性。

系统在用户登录注册界面拥有简洁清晰的错误提示和指导，并且所有界面都设计了快速的导航，易于用户的理解和操作，具有易理解性。

所有新闻的分页设计和具体新闻的评分收藏组件都方便了用户的操作和使用，并且有根据分类的下拉列表和模糊查询，具有易操作性。

（2）效率

效率是指使用系统完成某些任务的速度，通常针对于已经熟练掌握该系统使用方法的用户完成某些业务的速度，也包括了系统执行业务完成的速度。

系统利用redis存储推送新闻的数据，既提高了数据缓存查询的速度，也提高了系统执行新闻推送业务的速度，不需要每次实时推送都要到数据库中访问，并且系统搭建了新闻后台系统，方便系统管理员对系统的各个资源进行管理，提高了时间效率。

（3）维护性

维护性是指系统面对外界环境变换或者面对需求的变化时，对系统做出的改变的难易程度，在系统的维护阶段，需求的改变会带来设计实现的修改以及java代码的修改，所以需要有好的设计模式来进行项目的架构，减少需求变化而导致的代码变化。越符合开闭原则等设计模式的系统就更加具有维护性，软件系统才能存活更长的时间。

系统采用策略模式，即定义一系列的算法，把它们一个个封装起来，并且使它们可相互替换。系统将各个推送算法封装起来，系统管理员可以需求选择不同的推送算法，具有高内聚低耦合的特性，提高了系统的可维护性。

（4）功能性

功能性是指满足明确或者隐约的需求的功能，也指满足业务逻辑需求的功能。

系统包含了新闻推送所拥有的所有模块，包括用户登录、用户注册、新闻展示、新闻评分、新闻收藏、新闻推送、个人信息展示、后台管理等模块。

**3.4** **本章小结**

首先进行业务分析，通过用例图展示系统所包含的所有业务用例，确定业务的范围，通过活动图来展示项目核心业务的逻辑实现。然后进行系统的功能分析，通过系统功能模块图罗列系统所有的功能模块，再对主要的用例进行用例规约描述。最后从易用性、效率、维护性和功能性入手对系统进行非功能性分析。

**第4章 系统设计**

**4.1** **系统总体架构设计**

前端采用Vue构建前端新闻系统和后台管理系统，Vue-Cli脚手架帮助快速的构建Vue项目，element-ui是一个ui库，不依赖于Vue但是在开发过程中和Vue配合最方便，提高前端页面的一致性以及提高项目开发的效率。

利用Nginx进行负载均衡，当用户数量很大时或者用户行为频繁时，后台服务器压力会很大，负载均衡可以减少服务器的压力，防止服务器崩溃。

服务层共包含收藏模块、评分模块、用户模块、管理员模块、新闻类别模块、登录模块、注册模块、新闻模块以及最重要的推送模块，推送模块中利用mahout实现协同过滤推送，然后利用K-means算法进行改进。这些模块为前端提供多种接口以供调用，并连接数据库返回数据给前端。

缓存采用redis内存数据库，用户缓存推送新闻和热点新闻，由于内存数据库性能极高的特性，可以提高推送的速度和用户的体验。

采用Spring AOP进行日志记录，在pom中引入logging，把日志记录业务和主业务分离开来，放在代码之外，避免在服务层多次出现调用，可以精简代码，简化开发。

系统整体架构图如图4.1所示：

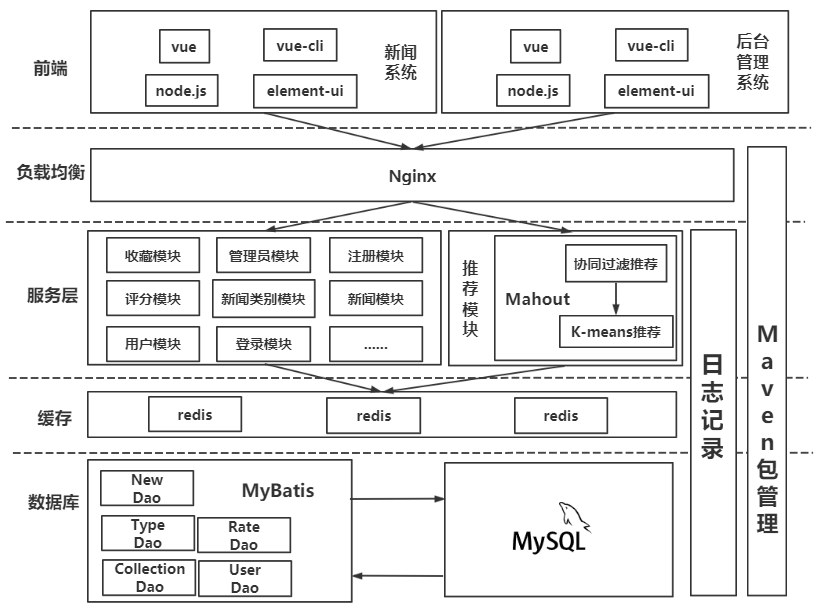


图4.1 系统整体架构图

后端采用MVC架构来管理数据，MVC架构中的M指的是Model模型，V指的是View视图，C指的是控制器Controller。Model负责存储系统的中心数据，在本系统中用实体类来表示，利用Mybatis实现dao层，本文封装了数据持久层的主要工作和与数据库的连接任务，服务层主要负责业务逻辑模块，通过调用定义好的DAO接口层封装通用业务逻辑，它有助于业务逻辑的独立性和可重用性，控制器层负责对特定业务模块的过程控制，并调用服务层的接口来控制业务流程。View层则将controller层的信息以json的形式返回给前端。

DAO层和服务层首先设计接口，然后设计实现类，这是软件开发中面向接口的设计原则。这种面向接口编程的优点是减少了程序的耦合。在系统扩展时，只需创建一个新的重写类接口方法进行更新和扩展，就可以在不修改源代码的情况下实现扩展。具有易于扩展和维护的优点。

后端三层MVC架构图如图4.2所示：

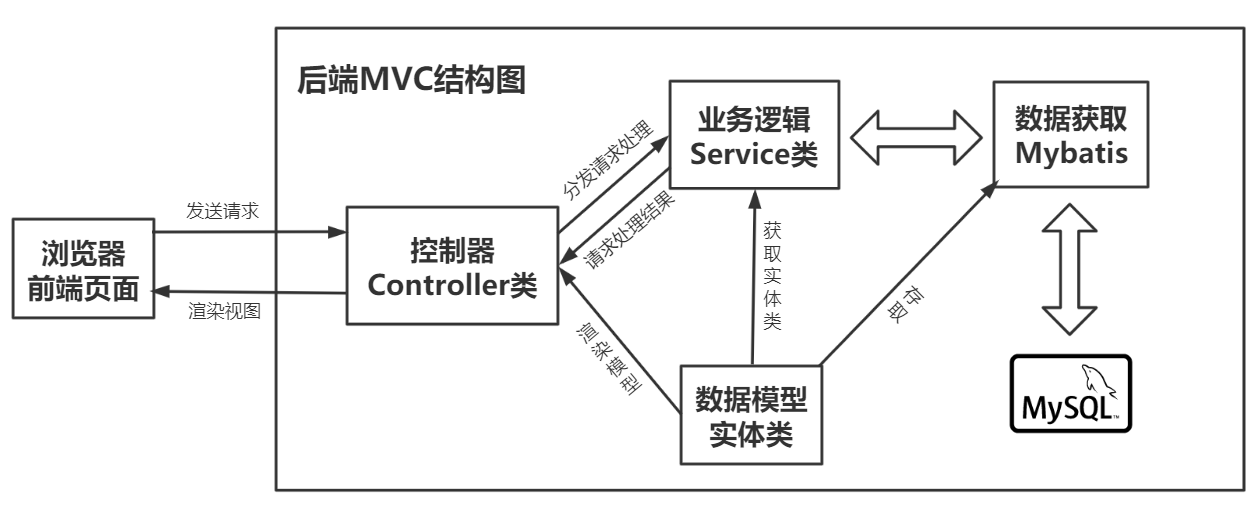


图4.2 后端三层MVC架构图

**4.2** **系统静态结构设计**

在系统用例图的基础之上，抽象出系统的静态设计视图，即类图。新闻推送系统的类图包括普通用户的类图和管理员的类图。

普通用户的类图设计和管理员的类图设计如图4.3和图4.4所示：

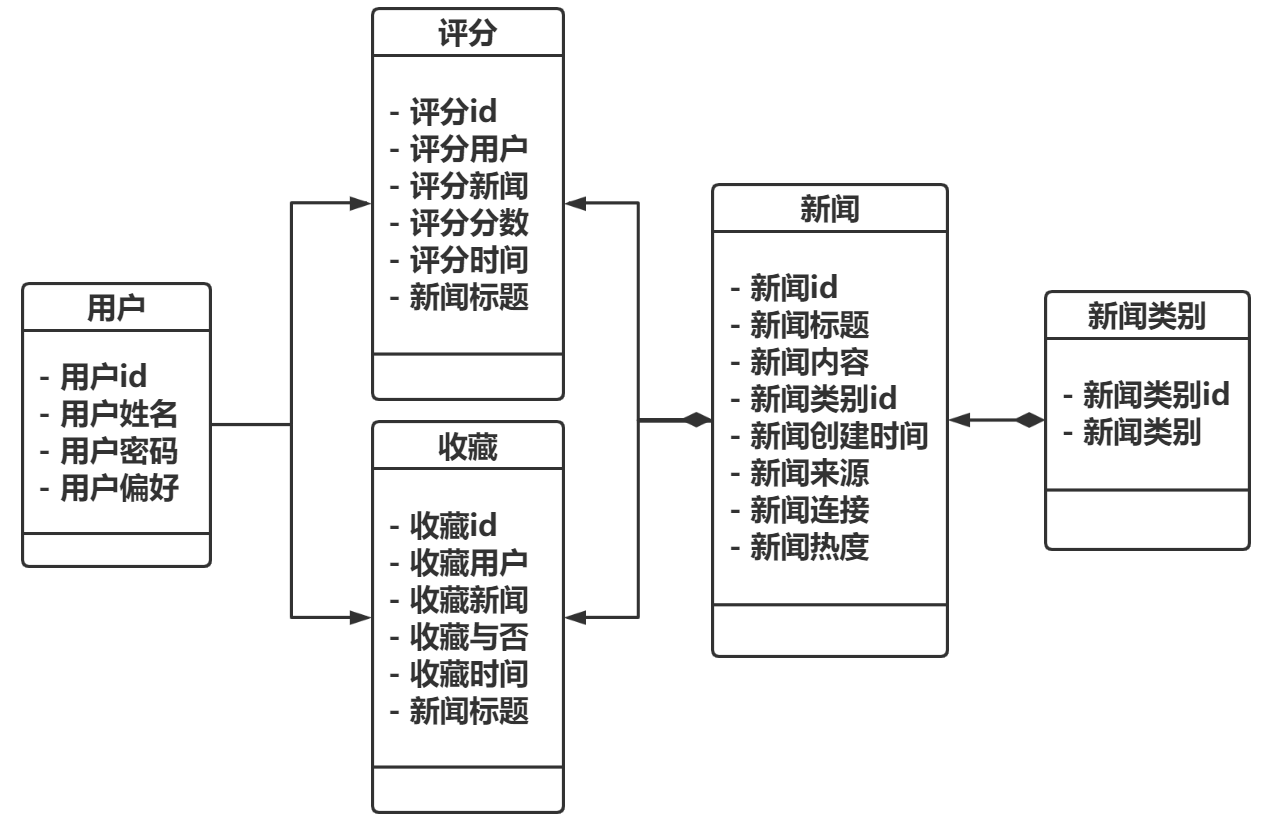


图4.3 普通用户的类图设计

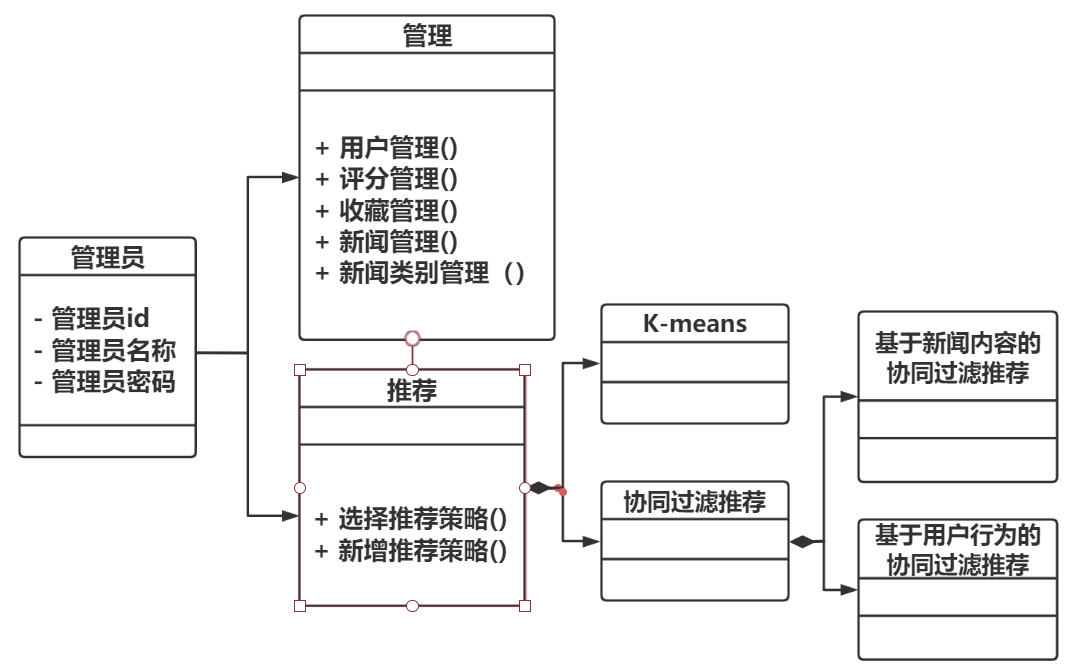


图4.4 管理员的类图设计

**4.3** **业务用例的实现**

在系统流程的分析基础之上，设计系统的动态视图，即顺序图。主要的用例图包括推送新闻的顺序图和新闻管理的顺序图。推送新闻的顺序图和新闻管理的顺序图如图4.5和图4.6所示：

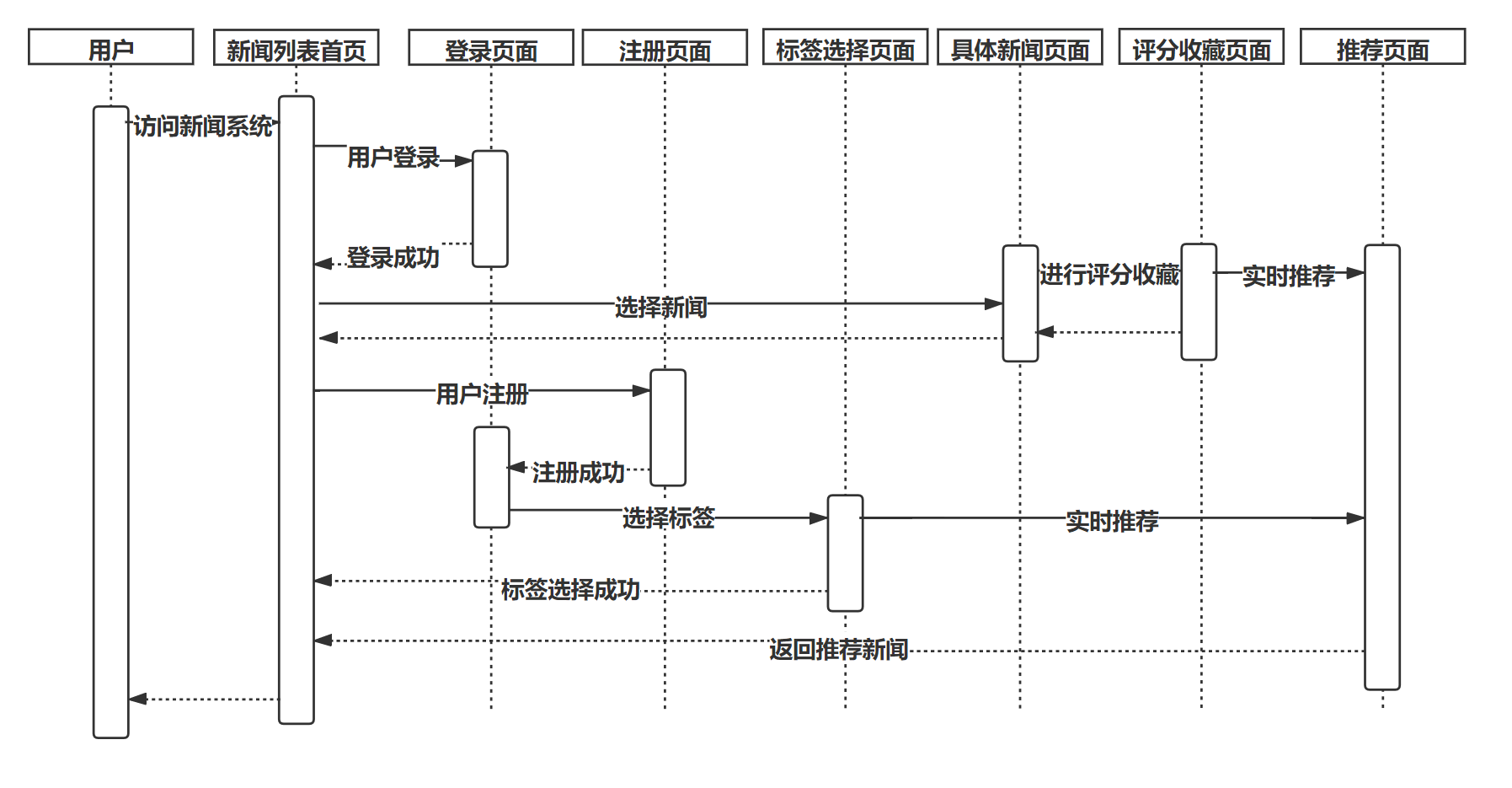


图4.5 推送新闻的顺序图

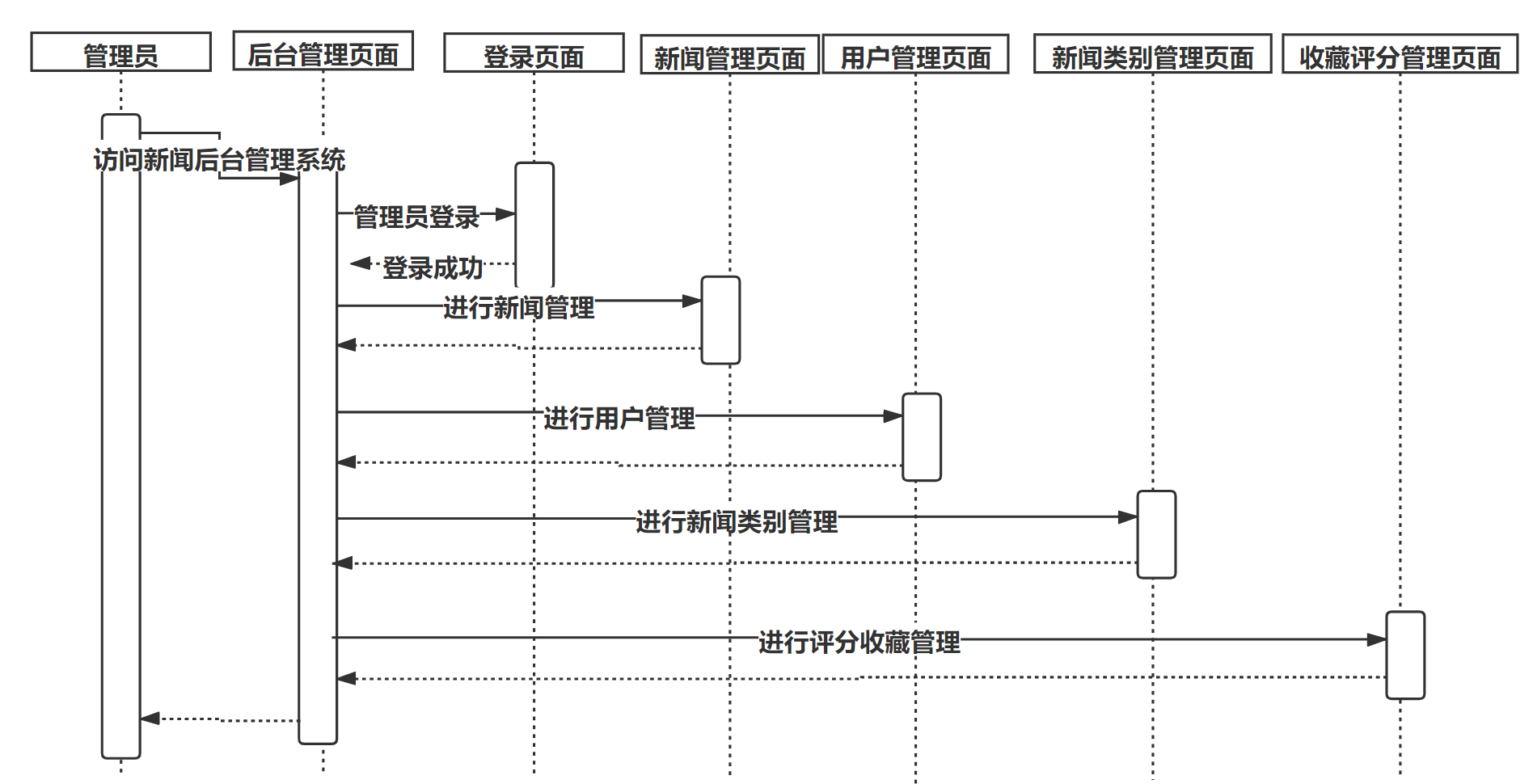


图4.6 新闻管理的顺序图

**4.4** **数据库设计**

**4.4.1** **逻辑结构设计**

主要针对数据库进行设计，通过E-R图来表示实体与实体之间的联系，主要包括以下联系：

（1）用户与新闻的评分联系。一个用户可以评分多个新闻，一个新闻可以被多个用户评分，他们之间存在着一对多的关系。

（2）用户与新闻的收藏联系。一个用户可以收藏多个新闻，一个新闻可以被多个用户收藏，他们之间存在着一对多的关系。

（3）新闻与新闻类别的联系。一个新闻可以拥有多个新闻类别，而一个新闻类别可以被多个新闻所拥有，他们之间存在着多对多的关系。

用户-新闻的E-R图如图4.7所示：

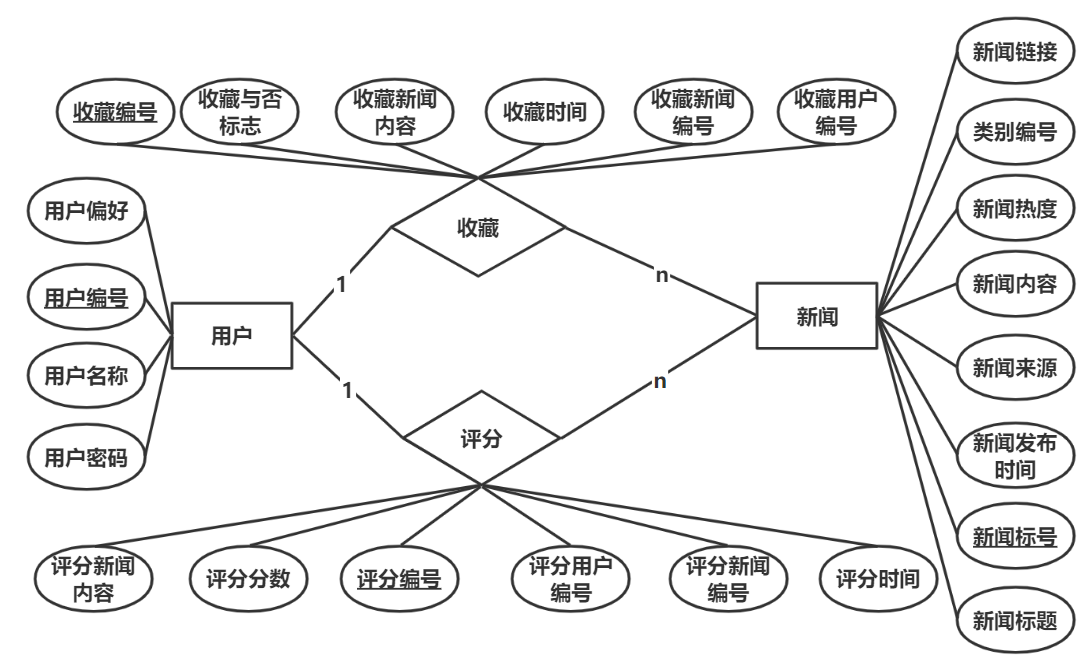


图4.7 用户-新闻的E-R图

新闻-新闻类别的E-R图如图4.8所示：

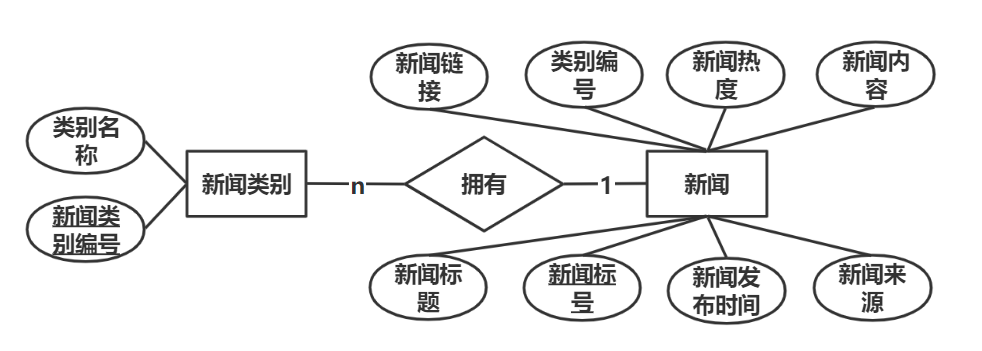


图4.8 新闻-新闻类别的E-R图

**4.4.2** **物理设计**

数据库采用的是MySQL数据库。共用到五个数据表，分别是：新闻表（news），新闻类别表（newstype）,评分表（score）,收藏表（collection）,用户信息表（users），如表4.1至表4.5所示：

表4.1 新闻类别的数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 是否允许为空 | 默认值 | 字段备注 |
| Id | Int（11） | 否 | AUTO\_INCREMENT | 新闻类型id |
| Type | Varchar（11） | 是 | <空> | 新闻类别 |

表4.2 新闻的数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 是否允许为空 | 默认值 | 字段备注 |
| news\_id | int（11） | 否 | AUTO\_INCREMENT | 新闻主键 |
| news\_title | varchar（255） | 是 |  | 新闻标题 |
| news\_content | text | 是 |  | 新闻内容 |
| type\_id | int（11） | 是 | <空> | 新闻类型 |
| news\_creattime | varchar（20） | 是 | <空> | 新闻发布时间 |
| news\_recourse | varchar（20） | 是 | <空> | 新闻来源 |
| news\_link | varchar（255） | 是 | <空> | 新闻链接 |
| Heat\_num | Int（11） | 是 | 0 | 新闻热度 |

表4.3 用户信息的数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 是否允许为空 | 默认值 | 字段备注 |
| User\_id | Int（11） | 否 | AUTO\_INCREMENT | 用户id |
| User\_name | Varchar（255） | 是 | <空> | 用户名称 |
| User\_password | Varchar（255） | 是 | <空> | 用户密码 |
| Prefer | Varchar（255） | 是 | “null” | 偏好标签 |

表4.4 评分的数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 是否允许为空 | 默认值 | 字段备注 |
| Id | Int（11） | 否 | AUTO\_INCREMENT | 评分id |
| Uid | Int（11） | 是 | <空> | 用户id |
| Nid | Int（11） | 是 | <空> | 新闻id |
| Score | Int（11） | 是 | <空> | 评分分数 |
| Time | Timestamp | 是 | <Insert-TimeStamp> | 评分时间 |
| Title | Varchar（255） | 是 | <空> | 新闻标题 |

表4.5 收藏的数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 是否允许为空 | 默认值 | 字段备注 |
| Id | Int（11） | 否 | AUTO\_INCREMENT | 收藏id |
| Uid | Int（11） | 是 | <空> | 用户id |
| Nid | Int（11） | 是 | <空> | 新闻id |
| Collection | Int（10） | 是 | <空> | 是否收藏 |
| Time | Timestamp | 是 | <Insert-TimeStamp> | 收藏时间 |
| Title | Varchar（255） | 是 | <空> | 新闻标题 |

**4.5** **类的详细设计**

实现K-means算法是最主要的类。首先从数据库中读取用户的行为数据，然后设置定义的K值，即分为几个簇，然后随机选择K个点作为聚类的中心。然后根据用户的评分进行聚类，进行循环判断当簇心不变时或者是小于某个设定值时循环结束返回聚类结果，每次循环时找出每个用户与聚类中心的距离，找出距离最小的用户下标，并归入到各个最近的聚类当中去，计算距离采用的是欧几里得算法，计算的速度最快。然后判断新旧聚类中心是否一样，不一样则通过计算平均值创建新的聚类中心。通过聚类算法之后，用户就被分成了多个簇，然后根据用户所在的簇进行下一步的协同过滤推荐。

K-Means类的流程图如图4.9所示：

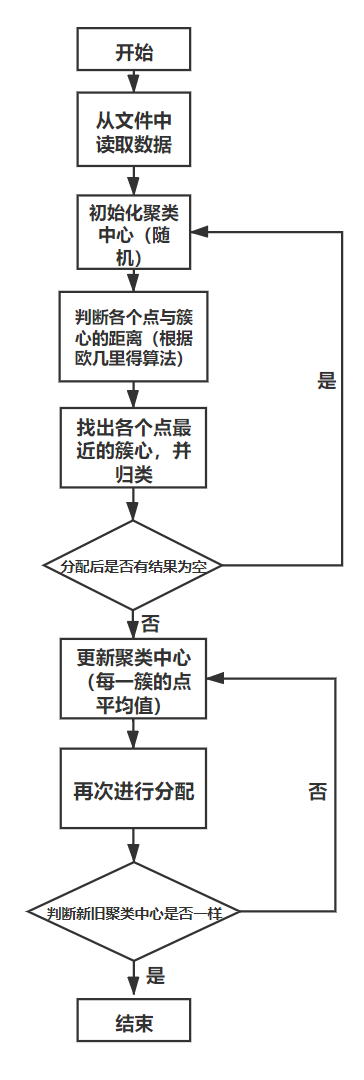


图4.9 K-Means类的流程图

**4.6** **本章小结**

前端首先采用架构图的方式进行系统的整体设计，包括系统的整体架构图和后端的MVC架构图；然后在架构图的基础上利用类图进行系统的静态结构设计；其次利用主要业务的顺序图描述业务用例的实现；最后描述数据库的逻辑结构设计和物理设计并给出主要类的流程图。

**第5章 系统实现**

**5.1** **系统物理结构**

前端共包含用户注册、用户登录、选择标签、新闻浏览、推送新闻、评分新闻和收藏新闻组件。

后端共包含用户管理、新闻管理、推送管理、新闻类别管理、评分管理和收藏管理组件。

数据库包含用户数据、收藏数据、新闻类别数据、评分数据和新闻数据。

前端组件用于响应用户的动作和请求、显示页面、页面与页面之间的跳转以及向后端发送网络请求获取数据。

后端组件用于从数据库获取数据并响应前端发送的网络请求、对数据进行增删改查的管理以及推送算法的实现。

爬虫组件用于爬取中国新闻网的实时新闻并分类存入数据库中。

系统组件图如图5.1所示：

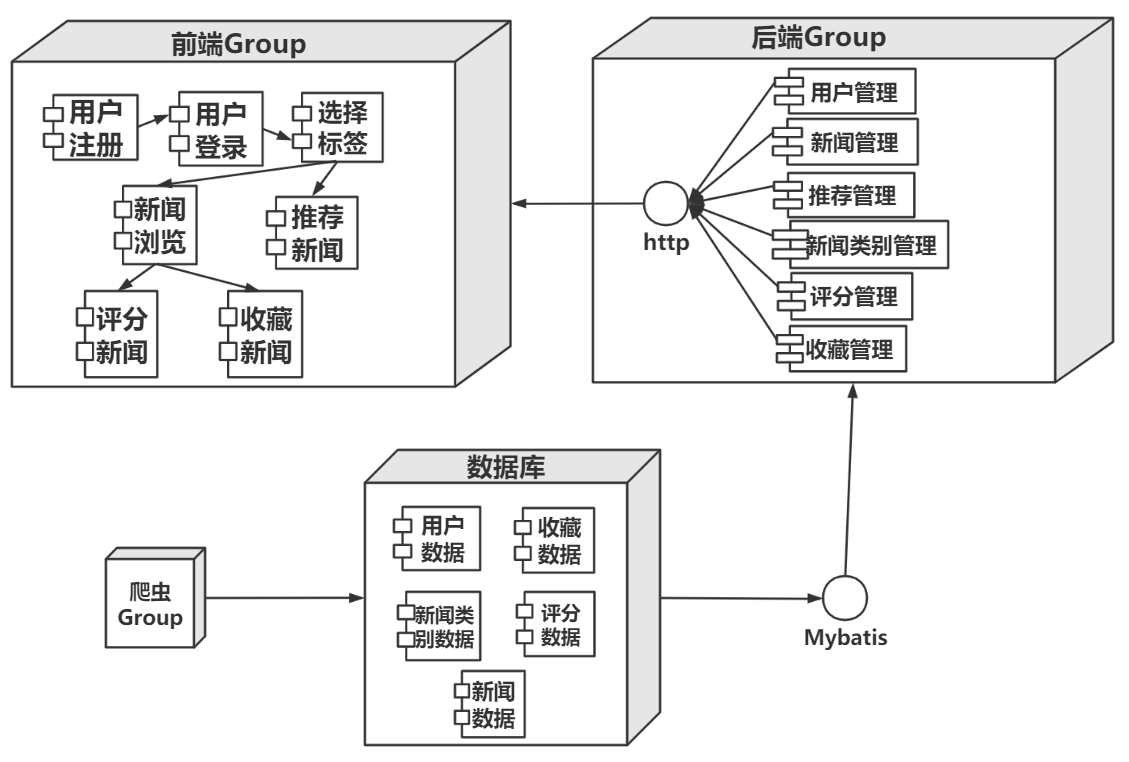


图5.1 系统组件图

**5.2** **系统主要功能实现**

在用户未登录或者首次登录时提供选择标签的功能，可以用于解决系统冷启动问题。选择喜欢的标签页面如图5.2所示：



图5.2 选择喜欢的标签

基于K-Means算法的协同过滤推送功能的实现页面如图5.3所示：



图5.3 基于K-Means算法的推送功能

基于用户行为的协同过滤算法的推送功能的实现页面如图5.4所示：



图5.4 基于用户行为的协同过滤算法的推送功能

基于物品相似度的协同过滤算法的推送功能的实现页面如图5.5所示：



图5.5 基于物品相似度的协同过滤算法的推送功能

猜你喜欢的实现页面如图5.6所示：



图5.6 猜你喜欢

新闻分页展示的实现页面如图5.7所示：



图5.7 新闻的分页展示

具体新闻展示的实现页面如图5.8所示：



图5.8 具体新闻展示

用户个人信息界面的实现页面如图5.9所示：

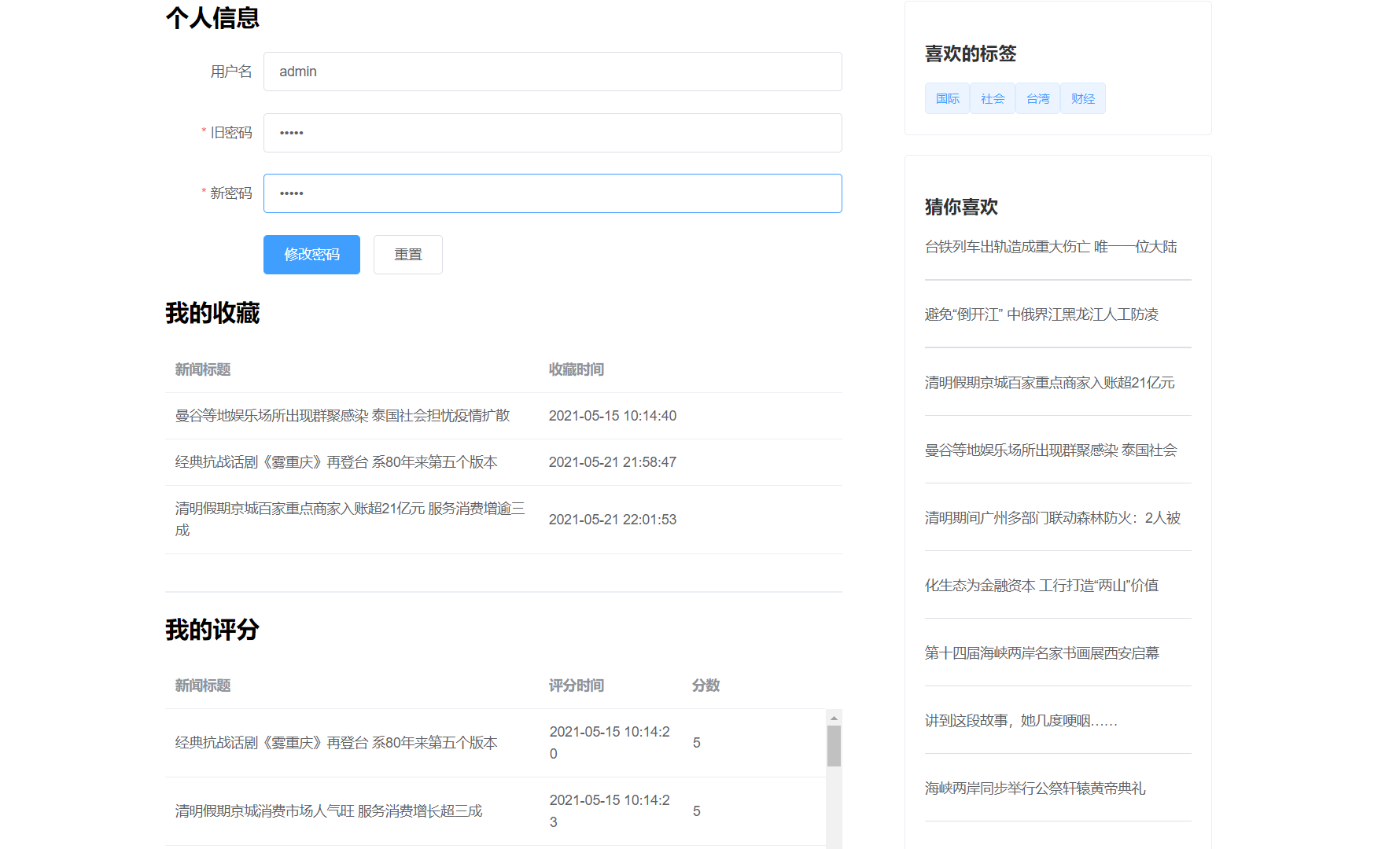


图5.9 用户的个人信息界面

**5.3** **实现中遇到的问题和解决方法**

（1）用户冷启动问题，即新注册的用户在没有行为数据或者行为数据较少的情况则无法进行准确的推荐。

针对该问题通过在用户注册完成并且首次登录之时页面罗列所有新闻的标签，用户选择自己喜欢的标签来解决；在用户行为数据较少的情况下，先推送标签中的热门新闻给用户，当用户行为的数据逐渐增多后，再将协同过滤的推荐列表和热门新闻的推荐列表进行加权推荐。

（2）新闻爬取模块需要手动分类型爬取的问题，每次爬取新闻都需要打开PyCharm输入需要爬取的网站和爬取新闻的类别，爬取效率低且不具备自动化。

针对该问题设置定时器和将所有类型整合搭建一个定期自动爬取新闻模块。利用定时器固定每天爬取新闻，并且删除一个星期之前的新闻，以此保证新闻的实时性和时效性。

（3）基于新闻内容推送问题，在代码实现早期新闻内容推荐方面实现困难。

针对该问题总结出新闻推送的思路，首先在数据库中为每个用户维持一个关键词列表，利用TF-IDF算法提取每个新闻的关键词，为关键词列表设置一个衰减系数，定期对用户的喜好关键词的TF-IDF进行更新。可以利用mahout实现基于新闻内容相似度的系统过滤算法。

（4）推送模块和新闻模块的交互问题。

针对该问题将推送模块写在java中作为接口从数据库中选择推送的新闻，设置一个推荐列表，如果列表内新闻数量不够则用用户喜欢的标签中选取热门新闻填补，前端使用axios调用后台的接口获取新闻数据，并且利用Vue的双向绑定进行显示。

（5）大数据问题，当新闻或者用户行为数据达到很大数量级时，基于用户行为和物品相似度的协同过滤算法的实时推送会有很大的影响，推送的速度很慢，用户的体验会变差。

针对该问题一是将新闻数据或者用户行为数据存到redis中，因为redis是基于内存的数据库，寻址速度相比较MySql数据库相差极大的数量级。

二是在此基础上利用Kmeans算法进行聚类，将大数据量分成多个小数据量然后再进行协同过滤算法。

（6）页面样式不协调问题。

针对该问题利用Vue整合element-ui进行样式的修改，通过阅读element-ui的官方文档来搭建页面的前端，使得页面样式更具一致性，用户的体验更好。

**5.4** **本章小结**

首先利用系统组件图阐述系统的物理结构；然后展示了系统的主要实现，包括推荐新闻的实现、新闻展示和个人信息展示等功能；最后叙述了在项目开发过程中遇到的问题，并给出相应的解决方法。

**第6章 系统测试**

**6.1 单元测试**

单元测试是对程序中可进行测试的最小单位进行测试，通常是对系统中的每一个单元模块进行独立测试，测试他们的功能是否与预期一致。

推荐模块的单元测试如表6.1所示：

表6.1 推荐模块的单元测试

|  |  |  |
| --- | --- | --- |
| 测试项目 | 期待结果 | 测试结果 |
| 用户未登录 | 提醒用户登录并返回到主页面 | 成功 |
| 用户首次登录 | 跳转到选择喜好标签的页面 | 成功 |
| 用户进行评分和收藏 | 返回的推荐的列表进行更新 | 成功 |
| 用户取消评分和收藏 | 返回的推荐的列表进行更新 | 成功 |

爬虫模块的单元测试如表6.2所示：

表6.2 爬虫模块的单元测试

|  |  |  |
| --- | --- | --- |
| 测试项目 | 期待结果 | 测试结果 |
| 输入错误的网页url | 提醒url错误的信息，并且拒绝访问 | 成功 |
| 设置错误的定时时间 | 提醒定时时间错误，并且拒绝访问 | 成功 |
| 爬取的新闻和类别不符 | 提醒类型匹配不符合，并且拒绝访问 | 成功 |
| 输入正确的信息 | 爬取新闻到数据库 | 成功 |

**6.2 黑盒测试**

黑盒测试将测试对象看作一个黑盒子，只根据输入输出来判定测试对象的正确性，主要的测试方法分为等价类划分和边界值分析。等价类划分将数据集分为多个子集，从各个子集中选出具有代表性的数据作为测试用例，边界值分析是在等价类划分的基础上侧重于边界值的数据。

针对用户的登录注册进行等价类划分黑盒测试，等价类的划分如表6.3所示：

表6.3 用户登录注册测试的等价类划分

|  |  |  |
| --- | --- | --- |
| 输入 | 有效等价类 | 无效等价类 |
| 用户名 | （1）长度3-5位的字符 | （2）少于3位的字符  （3）多余5位的字符  （4）存在非法的字符 |
| 密码 | （5）长度在3-7位的字符 | （6）少于3位的字符  （7）多余7位的字符  （8）存在非法的字符 |

测试的用例和结果如表6.4所示：

表6.4 测试用例和结果

|  |  |  |
| --- | --- | --- |
| 测试对象 | 测试用例 | 测试结果 |
| 用户名 | Admin | True |
| 用户名 | Ad | False |
| 用户名 | LongAdmin | False |
| 用户名 | AdminWith@#￥ | False |
| 密码 | Passwod | True |
| 密码 | Ps | False |
| 密码 | LongPassword | False |
| 密码 | PasswordWith@#￥ | False |

**6.3 接口测试**

对Java后端所提供的接口进行测试，安装idea所带的插件RestService，插件会显示当前项目所有的请求地址，项目的接口如图6.1所示：

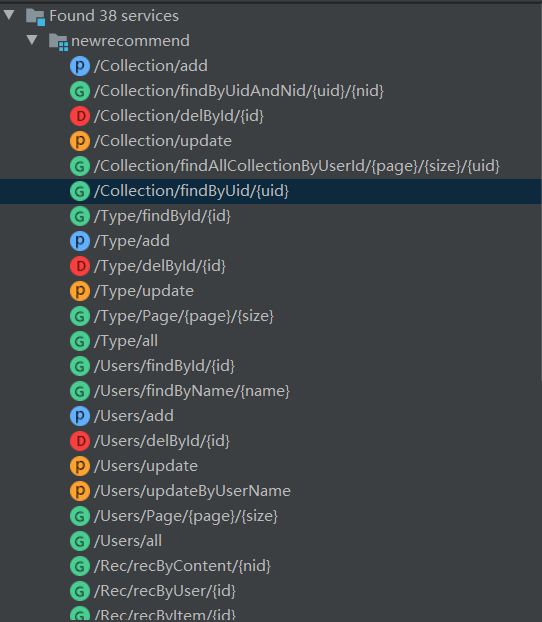


图6.1 项目的接口

可以输入请求方式，地址和参数来进行请求Http请求的测试，请求的结果数据以Json格式返回，如图6.2所示：

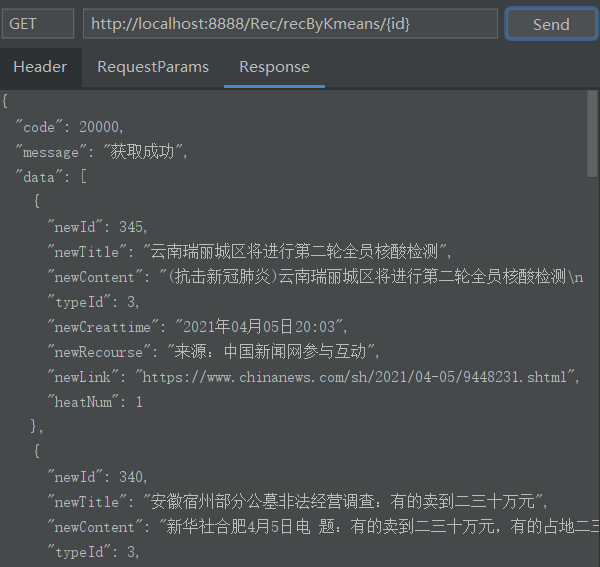


图6.2 接口返回的数据

通过该插件就可以对项目的接口进行测试，只要返回的数据code为20000，那么前端就可以通过Http请求调用该接口获取到数据。

**6.4** **本章小结**

分别对系统进行了单元测试、黑盒测试已经接口测试，并给出了相应的测试用例和测试结果。

**第7章 总结与展望**

**7.1 总结**

针对当下互联网飞速发展而产生的信息过载和用户兴趣不明确的问题，设计和实现了新闻推荐系统，旨在帮助用户更好的体验新闻的阅读，并且推动新闻推荐系统的发展，总结系统共实现以下几点：

（1）包含完整的新闻系统所拥有的模块，具备新闻推荐系统所需要的功能，在软件设计方面，系统具备良好的完整性，效率，维护性，功能性等；

（2）使用当下最流行的Spring Boot + Vue 前后端分离框架进行搭建，项目开发速度快，可维护性高。Spring Boot + Vue 前后端分离的框架也是当下企业流行的框架，对于我此后面向就业奠定了项目基础和实践基础。

（3）算法方面实现了基于用户行为的协同过滤算法和基于物品的协同过滤算法，并在此基础上利用聚类算法进行改进，降低了协同过滤所具有的冷启动问题和大数据情况下推荐缓慢问题带来的影响，提高了推荐的速度。

（4）相比较其他新闻推送系统，本系统针对协同过滤算法的缺点进行改进，在协同过滤的基础上结合k-means算法实现系统的推送模块。在大数据的情况下，常规的协同过滤进行实时推送时，推送新闻响应给用户通常需要1-2秒，数据量越大响应时间越久。结合了k-means算法并利用redis存储推送新闻数据之后，用户几乎感受不到延迟，大幅度的提高了推送的效果和用户的体验。

**7.2 展望**

由于时间紧迫，新闻推送系统的设计和实现还不是很完善，例如，用户的信息更改功能没实现，新闻爬虫部分未实现自动爬取并分类的功能，推送新闻存储到redis之后和数据库的数据一致性还没有得到很好的解决等等。

进一步我还会学习k-means算法，学习它的改进算法，包括kmeans++, ISODATA,Kernel K-means等，并且在本系统上进行改进和应用。此外还要学习最新的推荐算法，学习分库分表来进一步应对大数据的情况。

参考文献

[1] 吴青劼,洪涛,马骏.长尾理论综述[J].周口师范学院学报,2010,27(01):124-129.

[2] 刘玲,黄丽蓉,刘胜宗.论文推荐系统的关键技术研究[J].湖南工程学院学报(自然科学版),2017,27(04):43-47.

[3] 殷锋,曹旭,漆翔宇.基于聚类的推荐算法综述[J].西南民族大学学报(自然科学版),2021,47(03):303-309.

[4] 张峰.应用SpringBoot改变Web应用开发模式[J].科技创新与应用,2017(23):193- 194.

[5] 麦冬,陈涛,梁宗湾 轻量级响应式框架Vue.js应用分析[J].信息与电脑(理论版),2017(7):58 -59.

[6] 李冲.基于用户的协同过滤推荐算法MapReduce并行化实现[J].软件导刊,2018,17(10):76-80.

[7] 杨俊闯,赵超.K-Means聚类算法研究综述[J].计算机工程与应用,2019,55(23):7-14+63.

[8] 孟开元,岳宇航,曹庆年.新闻推荐系统研究综述[J].软件导刊,2021,20(01):249-252.

[9] 魏浩,张伟,郭新明.应用聚类分析的协同过滤推荐算法[J].福建电脑,2021,37(05):1-4.

[10] 赵亮,胡乃静,张守志.个性化推荐算法设计[J].计算机研究与发展,2002(08):986-991.

[11] Miller BN, Albert I, Lam SK, Konstan JA, Riedl J. MovieLens unplugged: Experienceswith anoecasionally connected recommender system[C].In:Proe.of the Int'1 Conf. On Intelligent User Interfaces.New York: ACMPress, 2003.263-266.

[12] Nanopoulos A , Manolopoulos Y.Mining patterms from graphtraversals[J].Data and Knowledge Engineering, 2001, 37(3) :243-266.

[13] Claypool M,Le P, Waseda M, et al.Implicitinte restindicators[C]//Proceedings of the ACM Intelligent User Interfaces Conference(IUI ).New York: ACM Press ,2001:14-17.

[14] Zhao Jinghe, Liu Guiquan.Automatic modeling based on interestclustering[J].AI Commun, 2001, 14(3):129-147.

[15] 邓爱林,左子叶,朱扬勇.基于项目聚类的协同过滤推荐算法[J].小型微型计算机系统,2004(09):1665-1670.

[16] 王千,王成,冯振元,叶金凤.K-means聚类算法研究综述[J].电子设计工程,2012,20(07):21-24.

[17] 邓爱林,左子叶,朱扬勇.基于项目聚类的协同过滤推荐算法[J].小型微型计算机系统,2004(09):1665-1670.

[18] 冷亚军,陆青,梁昌勇.协同过滤推荐技术综述[J].模式识别与人工智能,2014,27(08):720-734.

[19] 袁方,周志勇,宋鑫.初始聚类中心优化的k-means算法[J].计算机工程,2007(03):65-66.

[20]杨燕,靳蕃,KAMEL Mohamed.聚类有效性评价综述[J].计算机应用研究,2008(06):1630-1632+1638.

[21]杨智鸿. 基于深度学习的新闻推荐算法研究[D].北京邮电大学,2020.

致谢

转眼间，四年的悠悠时光已经如同秋天的落叶，无声无息的飘散了，我的本科生涯也在这个夏天就要接近尾声了。

我要感谢我的学校，山东科技大学。回想过去四年的校园生活里，有为了考研辗转于教室和自习室的画面，也有为了篮球校联赛而早起训练、和队友们球场拼搏的画面，有过挫折打击，也有过欢呼雀跃。四年校园生活很美好，我享受这期间在山东科技大学的每一个镜头和每一个过程，也感谢母校对我的栽培和教育。

感谢我的人生导师孙承爱，她也是我毕业设计的指导老师。孙老师是一名学术严谨，知识渊博的老师。在我的整个毕业设计实现和毕业论文编写过程中，孙承爱老师都会在百忙的时间里给予我细心的指导和分析，这种负责的态度让我感动和敬佩。

感谢计算机学院的各位老师，你们的精彩授课让我对计算机专业领域有了更深入的感悟和体会，尤其是Java课程，老师幽默的上课形式激发了我对计算机行业的热爱，也是我决定我未来发展路线的启蒙。在计算机学院的4年，从计算机小白到能做出自己想要的项目，到最后完成整个本科阶段的毕业设计，都离不开各个阶段老师的辛勤栽培，在此对他们表示敬意。

感谢我的舍友和朋友们，我们一起上课，一起运动，一起考研，一起探讨毕业论文，一起畅谈人生理想。还要感谢我篮球队的队友们，他们是我整个大学阶段最好的兄弟甚至是人生中最好的挚友，一起训练和拼搏的日子我会一直铭记。

正因为有他们的陪伴、支持、鼓励和帮助，我的大学生活才如此充实和精彩。

最后还要感谢父母和家人，他们为我提供经济和精神的支持，让我在大学期间没有经济上的负担，可以放心的去追寻自己的理想。在考研期间也不断给予我鼓励，促使我勇敢的去面对孤独的考研生活，最后的考研成功离不开父母的家人的陪伴和支持。

通过这次毕业设计和毕业论文的编写和实现，我查阅了大量的文献和论文，发现前人在这个推荐方面已经研究到了很深的地步并且取得了很好的效果和应用。大量的文献和论文也给了我大量的帮助。这很大程度上培养了我学习的能力，对于我未来的发展很有帮助。

感谢四年的大学经历，使我更有热情和信心地迈向人生的下一阶段。