



吉林大学

## 本科生毕业论文（设计）

中文题目 基于 Android 平台的今日头条 app 的设计与开发

英文题目 Design and development of today's headline app  
based on Android

学生姓名 杨树德 班级 551604 学号 55160411

学 院 软件学院

专 业 软件工程

指导教师 柴胜 职称 讲师

## 吉林大学学士学位论文（设计）承诺书

本人郑重承诺：所呈交的学士学位毕业论文（设计），是本人在指导教师的指导下，独立进行实验、设计、调研等工作基础上取得的成果。除文中已经注明引用的内容外，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作品成果。对本人实验或设计中做出重要贡献的个人或集体，均已在文中以明确的方式注明。本人完全意识到本承诺书的法律结果由本人承担。

学士学位论文（设计）作者签名：

2020 年 5 月 20 日

## 摘要

随着移动互联网的飞速发展，传统媒体为了更加适应新时代的发展需求，迅速向网络媒体转型，于是市场上开始出现各类新闻客户端，以争抢用户流量为核心，展开一场拉锯战。这十几年来，社会生活飞速变迁，很多人因为忙碌而不再去耐心的寻找自己需要的新闻。另一方面，在这种趋势下，人们已经从信息匮乏走向了信息过载，这个时候，如何从海量的信息中筛选出自己需要的那部分往往变成一件很困难甚至是不可能的事情。因此，信息生产者开始将目光投向推荐系统，于是，利用用户行为进行数据挖掘的各种算法被投向市场并不断迭代，其中，基于用户和物品的协同过滤算法使用最为广泛。

本次设计的 app 主要是基于 Android 平台，服务端使用 spring mvc 框架，首先客户端通过调用爬虫将百度新闻中的信息爬取下来显示到 app 上，并将其进行一系列处理，之后把这些新闻整理成 JSON 类型的数据传送到服务端并插入数据库当中便于之后的工作，接着利用这些用户行为信息通过协同过滤算法进行处理并将处理结果返回客户端的推荐栏并与之前的内容合并显示，系统根据用户的评论数量生成一个兴趣排行榜，用户可以在兴趣排行中看到自己最感兴趣的若干条新闻，同时系统也会根据排行榜以及用户的兴趣动态改变推荐一栏的新闻内容。

App 名称为趣新闻，普通用户可以使用的功能包括新闻的分类浏览、用户浏览兴趣排行榜、新闻评论、新闻收藏、新闻搜索、历史记录查询；而管理员除了上述功能以外还可以进行新闻的发布，再次从百度新闻网站上爬取若干条新闻这两个功能。

**关键词：爬虫、Android、推荐系统、协同过滤、新闻、springmvc**

# Design and development of today's headline app based on Android

Author: Yang Shude

Tutor: Chai Sheng

## Abstract

With the rapid development of the mobile Internet, in order to better adapt to the development needs of the new era, traditional media quickly transformed to online media, so various news clients began to appear in the market. Over the past decade or so, social life has changed rapidly, and many people are no longer patiently searching for the news they need because they are busy. On the other hand, under this trend, people have gone from lack of information to information overload. At this time, how to filter out the part that they need from the massive information often becomes a difficult or even impossible thing. Therefore, information producers began to turn their attention to the recommendation system, so various algorithms using user behaviors for data mining were put on the market and iterated continuously. Among them, collaborative filtering algorithms based on users and items were the most widely used.

The app designed this time is mainly based on the Android platform, and the server uses the spring mvcMVC framework. First, the client crawls the information in Baidu News by calling a crawler and displays it on the app, and organizes the news with user behavior into JSON type. The data is sent to the server, and then the user behavior information is used to process through the collaborative filtering algorithm and the processing result is returned to the client 's recommendation bar and merged with the previous content for display. The system generates an interest leaderboard based on the user 's comments and the number of favorites. Users can see some of their most interesting news in the interest ranking, and the system will also dynamically change the recommended news content according to the ranking list.

The name of the app is Fun News. Ordinary users can conduct classified browsing of news, user browsing interest rankings, news comments, news collections, news search, and history query; while administrators can publish news in addition to the above functions The two functions of crawling several news on the Baidu News website.

**Key words : Crawler, Android, Recommendation System, Collaborative Filtering, News, springmvc**

# 目 录

第一章	绪论.....	1
1.1	课题背景.....	1
1.2	研究目的.....	2
1.3	研究意义.....	2
1.4	研究现状.....	2
1.5	研究内容.....	4
1.6	论文结构.....	5
第二章	相关知识简介.....	5
2.1	Andorid 平台介绍.....	5
2.2	开发工具介绍.....	6
2.2.1	IDEA 介绍.....	6
2.2.2	Android studio 介绍.....	7
2.2.3	开发环境的搭建.....	8
第三章	概要描述.....	9
3.1	功能模块描述.....	10
3.1.1	功能名称及对应视图展示.....	10
3.1.2	功能介绍.....	12
3.2	开发及运行环境.....	13
3.3	准备内容.....	13
第四章	功能实现.....	14
4.1	登录注册功能.....	14
4.1.1	功能说明.....	14
4.1.2	数据库表结构说明.....	14
4.1.3	关键代码说明.....	14

4.1.4 实现逻辑.....	16
4.1.5 异常数据提醒.....	16
4.1.6 实现结果.....	17
4.2 新闻的分类浏览功能.....	17
4.2.1 需求分析.....	17
4.2.2 可行性分析.....	18
4.2.3 关键代码说明.....	18
4.2.4 实现逻辑.....	19
4.2.5 实现结果.....	20
4.3 新闻浏览排行榜功能.....	20
4.3.1 需求分析.....	20
4.3.2 可行性分析.....	21
4.3.3 数据库表结构说明.....	21
4.3.4 关键代码说明.....	21
4.3.5 实现逻辑.....	22
4.3.6 实现结果.....	22
4.4 新闻评论功能.....	22
4.4.1 需求分析.....	22
4.4.2 可行性分析.....	23
4.4.3 数据库表结构说明.....	23
4.4.4 关键代码说明.....	23
4.4.5 实现逻辑.....	23
4.4.6 实现结果.....	24
4.5 新闻推荐功能.....	24
4.5.1 需求分析.....	24

4.5.2 可行性分析.....	25
4.5.3 数据库表结构说明.....	25
4.5.4 协同过滤算法说明.....	26
4.5.5 关键代码说明.....	29
4.5.6 实现逻辑.....	30
4.5.7 实现结果.....	30
4.6 新闻收藏功能.....	31
4.6.1 需求分析.....	31
4.6.2 可行性分析.....	31
4.6.3 数据库表结构说明.....	32
4.6.4 关键代码说明.....	32
4.6.5 实现逻辑.....	32
4.6.6 实现结果.....	33
4.7 新闻搜索功能.....	33
4.7.1 需求分析.....	33
4.7.2 可行性分析.....	34
4.7.3 关键代码说明.....	34
4.7.4 实现逻辑.....	34
4.7.5 实现结果.....	35
4.8 爬虫部分.....	36
4.8.1 需求分析.....	36
4.8.2 可行性分析.....	36
4.8.3 具体操作步骤.....	36
4.8.4 关键代码说明.....	40
4.8.5 实现逻辑.....	41



4.9 新闻发布功能.....	41
4.9.1 需求分析.....	41
4.9.2 可行性分析.....	41
4.9.3 数据库表结构说明.....	41
4.9.4 关键代码说明.....	42
4.9.5 异常数据提醒.....	42
4.9.6 实现逻辑.....	42
4.9.7 实现结果.....	42
4.10 历史记录查询功能.....	43
4.10.1 需求分析.....	43
4.10.2 可行性分析.....	43
4.10.3 数据库表结构说明.....	44
4.10.4 关键代码说明.....	44
4.10.5 实现逻辑.....	44
4.10.6 实现结果.....	44
4.11 个人信息管理功能.....	45
4.11.1 功能说明.....	45
4.11.2 关键代码说明.....	45
4.11.3 实现逻辑.....	46
4.11.4 实现结果.....	46
第五章    总结与展望.....	47
5.1 总结.....	47
5.2 展望.....	49
参考文献.....	50

致谢.....	51
---------	----

# 第一章 绪论

## 1.1 课题背景

如果我们想要购买一部手机，可以有多少种实行办法？我们可以去手机厂商的代售点里根据店员的介绍和推荐来选定产品，可以上天猫淘宝京东等电商网站（或者 app）来根据自己的需求搜索产品，比如我们的预算是 3000-5999，希望购买一个全面屏的手机，希望处理器规格不能低于高通骁龙 855 或者麒麟 980，有了这些条件作为筛选，我们可选择的范围就缩小了很多，便于选择。甚至我们可以把购买手机的这个行为委托给能信得过的朋友去做，或者参考他们的意见来购买。上面这几个例子展示了在用户有很明确购买需求的情况下，面对过量的信息所采取的一些措施。在手机店里，能够摆放展示的手机是有限的，因此用户可以逐个上手体验来决定购买哪一部，而在电商网站中，商品数量是相当庞大的，我们不可能再去逐一查看，只能通过条件筛选来不断缩减要选择的范围。

然而，当用户没有特别明确的需求时，上边的过程就显得无从谈起。比如用户只是想随便看个新闻，但当用户打开新闻类门户网站，铺天盖地的新闻瞬间砸在脸上，密密麻麻的文字让人顿时失去了阅读的兴致。此时，我们说用户遇到了信息过载的问题，需要一种工具或者方法来辅助自己做出选择，这便是嵌入推荐算法的资讯类应用软件诞生的初衷。有了这样的软件，我们可以根据用户的行为来对用户的兴趣作出实时的分析，并以此来判断用户到底想要什么，并将用户可能感兴趣的条目推荐给他，以此为用户节约了用于搜索和思考的时间，也让各类新闻资讯得到更好的挖掘。

其实推荐系统并非一个新的领域，早年，亚马逊、谷歌、脸书等公司就已经在自家的系统中加入了推荐算法来为用户增加更棒的使用体验。如今，这类算法已经作为各类面向用户系统的常用组件，在今日头条、知乎、淘宝、豆瓣甚至是第三方浏览器中都有广泛应用。

这些技术的大规模商用，使得信息过载的问题得到了很好地解决，各大互联网公司借此争夺了大批的用户流量，促进了基于用户和物品的各类分析模型的发展和进步。

## 1.2 研究目的

作为一个即将毕业的软件工程专业的学生，在日常使用各种资讯类 app 时候发现这些 app 都能够根据我的使用和浏览习惯来把我可能感兴趣的新闻推送到首页，减少了我用在翻找新闻中的时间，因而对这种系统的机制原理产生了浓厚的兴趣，希望通过自己的学习和研究，来将这种算法简单的应用在自己开发的 app 上，以此提升个人的开发能力和学习能力。

## 1.3 研究意义

大数据发展到今天，人们无时无刻不在面临着信息爆炸带来的冲击，如何能做到于信息“沧海”中取“粟”就显得尤为重要。以协同过滤思想为主的推荐系统作为数据挖掘的一种具体应用，可以帮助用户将存在互联网世界中的“暗信息”发掘出来，以此达到对信息的更加充分的利用。此次课题的研究将使用户能够进行日常新闻资讯的随手浏览，了解最新情况。同时，自己感兴趣的新闻能够以最直接的方式被呈现，大大节约了筛选信息所花费的精力和时间[1]。

## 1.4 研究现状

基于以上的问题，分析目前可以用来实现此系统的形式有 Web 应用、移动客户端、微信小程序、PC 客户端四种主要的方式，Web 应用的好处是开发难度比较低，而且适配也更容易，只需要针对主流的浏览器进行适配就可以，但是基于网页的方式并不能带来良好的体验；PC 客户端的好处是性能比较稳定，但是开发成本较高，而且开发难度较大，维护不容易，况且如今很少有人会专门为了某个应用而去登录 PC，很多人也并没有经常使用电脑的习惯；微信小程序开发难度低，但是就目前的情况来看，普及程度还很低，大家还是习惯于 Android\IOS 系统中 app 的使用，因此本次开发将基于 Android 平台。

对于数据这部分，目前主流新闻资讯主要通过爬虫来进行数据的抓取，理论上 python 在这方面拥有更加良好的性能，但是考虑到前后端开发的统一，决定都采用 java 来进行开发，而 java 中最为常见的爬虫工具就是 Jsoup，因此此次开发使用

java 爬虫工具 Jsoup 来进行。

对于推荐系统这部分，因为个人开发并不涉及太多的用户注册，因此基于用户的协同过滤可得到的数据样本太少，这样算法无法很好的执行，因此采取基于内容的协同过滤。

在 Android 以前，市场上的移动平台大多数不是开放的，只是服务于特定公司的特定产品，因此有应用定制，功能局限，迭代缓慢等问题，而 Android 的出现则一举打破了这个局面。Android 因其开发的方便性、开放性迅速风靡全球，其包括了操作系统、人机交互界面以及应用程序等移动设备所需的所有组件，又因其开源，故不会像往常一样出现专有权的障碍。作为一个近年才异军突起的智能移动平台，Android 借助谷歌牵头的 OHA(Open Handset Alliance)开放手机联盟的商业运作，并联合百家手机制造商共同推出了成百上千款搭载 Android 平台的智能手机，大大增加了该平台的市场份额，使得这类移动设备能够走进千家万户，其商业前景被越来越多的人看好。同时，Android 设备的普及，也加快了互联网全民化的进程，加速了经济全球化的脚步。[2]

Jsoup 是 Java 中最为常见的爬虫工具，初识这个工具可能会觉得它很神秘，但是其底层原理归结起来就还是一款 HTML 的解析器。其工作原理也很直接，就是通过其提供的一些操作接口，能够根据某个 URL 地址、字符串或者是特定文件来解析对应的 HTML 的页面内容。Jsoup 提供的这套 API 是非常简洁易用的，它能够使用类似于我们在 DOM、CSS 中使用的一些选择器的语法来对爬取到的 HTML 页面数据进行提取和操作。Jsoup 的能够完成的功能如下：

- (1) 从某个 URL、字符串或者文件中解析 HTML
- (2) 利用 DOM 遍历或 CSS 选择器来查找、提取数据
- (3) 可操纵 HTML 元素、属性和文本
- (4) 可依据一个安全的白名单过滤用户提交的内容，防止 XSS 攻击

Jsoup 使用方式非常简单直观，其提供的强大的选择器功能让爬取网页数据整个操作一目了然，简洁明快，对于初学者极其友好。[3]

基于内容的推荐首先就是要对用户之前关注过的物品进行分析，将其中的内容信息和特征信息提取出来，将这些数据作统计后得出一个理论的模型然后根据这个模型去作个性化的推荐，这也就是说，通过用户历史上喜欢的物品在内容和其他特征上的情况来为用户推荐与之相似的物品。这一类的推荐方式的优点如下：

### 1.可解释性

该物品会被推荐给该用户的理由是能够给出解释的，用户看到后会理解这个物品出现在推荐列表的缘由，进而去信任系统所给出的推荐结果。

### 2.解决物品冷启动问题

当系统中添加了新的物品项目后，系统会自动的去分析这些物品的内容特征，然后将推荐的结果直接插入到推荐列表中来，而不需要得到用户的评分数据。

以上是基于内容的推荐所具备的优势和便利，当然它也存在很多的缺点：

1.有相当数量的物品内容是很难去作分析的，只能在一些角度上去作简单的剖析，比如新闻、音乐、电影、博客、帖子、图片、打上标签的物品等等，这些信息在给到后台后是很难直接分析并提取出来它们的内容特征的。

2.推荐范围有限。这种推荐的方式只是能简单的推荐一些在当前的情况下，用户可能会感兴趣的这类物品，但是实际情况中，很多用户并没有表现出明确兴趣意向的物品依然有可能是用户感兴趣的，因此这种推荐并不能给用户“惊喜”，也不能够有效的扩展用户的兴趣爱好。[4]

## 1.5 研究内容

通过查阅资料以及调查研究，结合自己的实际体验经验，最终确定了本次设计所包括的功能如下：

- 注册以及修改密码：用户对于趣新闻 app 的账号注册、登录以及个人页面的密码修改。

- 新闻的分类浏览：用户可以根据自己的喜好点击感兴趣的分类进行浏览，经过推荐算法筛选的新闻被显示先推荐栏里。

- 新闻浏览排行榜：根据用户收藏量和评论量来对新闻进行热门度排行，用户可以直接根据排行榜来浏览热门新闻。

- 新闻的发布：此功能只有管理员有权限操作。管理员登录账号，可以自主发布新闻，也可以点击刷新按钮再次从百度新闻上爬取新闻。

- 爬虫模块：通过 Jsoup 组件在前端进行数据爬虫，将取得的数据发送到 app 上。

- 数据推荐模块：通过基于内容的协同过滤对用户过往浏览的情况进行分析，将结果整合到推荐栏。

- 新闻评论：用户可以点击评论按钮对新闻进行评论

- 新闻收藏：用户可以长按某一条新闻，弹出对话框后确认自己是否要收藏该新闻。
- 新闻检索：用户可以提供关键字，在搜索栏输入要搜索的内容以检索目前爬取到 app 中的新闻情况。
- 历史记录：用户可以查看或者删除搜索记录和浏览记录。
- 个人资料管理：用户可以进入个人资料页面，点击对应栏来进行个人信息的修改。

## 1.6 论文结构

本文通过调研使用场景，分析了各类相关的实现技术，选择了合适的开发工具，通过查阅相关书籍、论文以及网络课程，最终实现基于 Android 的今日头条 app 的设计与开发——趣新闻，并已适配 Android 9，因此 Android 9 及以上版本的手机均可以安装运行。

本文是《基于 Android 的今日头条 app 的设计与开发》，论文的组织结构如下：

第一章绪论，介绍研究背景、研究目的、研究意义、Android 应用开发、爬虫以及推荐算法相关内容，期间查阅了许多资料。

第二章相关知识简介，介绍本次开发过程中所能够涉及到的相关技术、语言、框架等知识，主要包括 Android 开发相关内容、爬虫的应用、spring mvc 框架的配置和使用以及基于内容的协同过滤算法的原理简介。

第三章概要描述，介绍了大致的设计思路，还有系统的各个模块和整体框架。

第四章功能实现，本章将对各个功能模块和算法实现原理进行详细的说明，按照模块来进行分析，包括需求分析，可行性分析，功能描述，设计实现等。

第五章总结与展望，对本次毕业设计《基于 Android 的今日头条 app 的设计与开发》一题中存在的问题进行复盘，分析其中难点以及解决的策略，总结有用的知识。

## 第二章 相关知识简介

### 2.1 Android 平台介绍

法国作家利尔亚当(Auguste Villiers de l'Isle-Adam)在 1886 年发表了科幻小说《未来夏娃》，在这部小说中首次出现了 Android 这个词。亚当命名那些外表很像人的机器为 Android。Google 于 2005 年 11 月 5 日正式向外界公布了这款命名为 Android 的移动端操作系统，并联合了全球 34 家手机制造商、软件开发公司、电信运营机构以及芯片制造厂商，成立了 OHA(Open Handset Alliance)开放手机联盟，一个全球性的联盟组织，这个组织今后将联起手来，共同进行 Android 系统的开放源代码的开发和维护。2009 年 4 月 30 日，Android1.5 正式发布，次年搭载该系统的应用数量超过 9 万个。截止目前，Android 对于移动设备的覆盖率已经超过了 87%，遥遥领先于其他移动端操作系统。Android app 使用 java 语言来进行编写，当编译在进行的时候，所有数据和资源文件都被打包成 APK 文件便于安装时候进行解压。Android 操作系统是基于 Linux 内核来进行开发的，其为多用户的操作系统，因此每一个 app 都可以是一个不同的用户。运行期间，因为每个进程都独立的拥有自己的 JVM，所以在某一个 app 运行期间，该 app 的程序代码和其他 app 是隔离开来的。

### 2.2 开发工具介绍

#### 2.2.1 IDEA 介绍

本次项目由客户端和服务端组成，服务端部分使用的开发工具为 IntelliJ IDEA，它可以很方便的进行 maven 项目和 gradle 项目的开发。和传统的开发工具如 eclipse 需要把开发所需 jar 包手动导入不同，IDEA 只需要在配置文件中引入开发所需要的包，点击同步按钮后它会自动根据之前配置好的仓库去相应的 url 中下载所需的资源包并自动导入，省去了不少前期的工作量。如下图 2-1 为 IDEA 的启动界面。



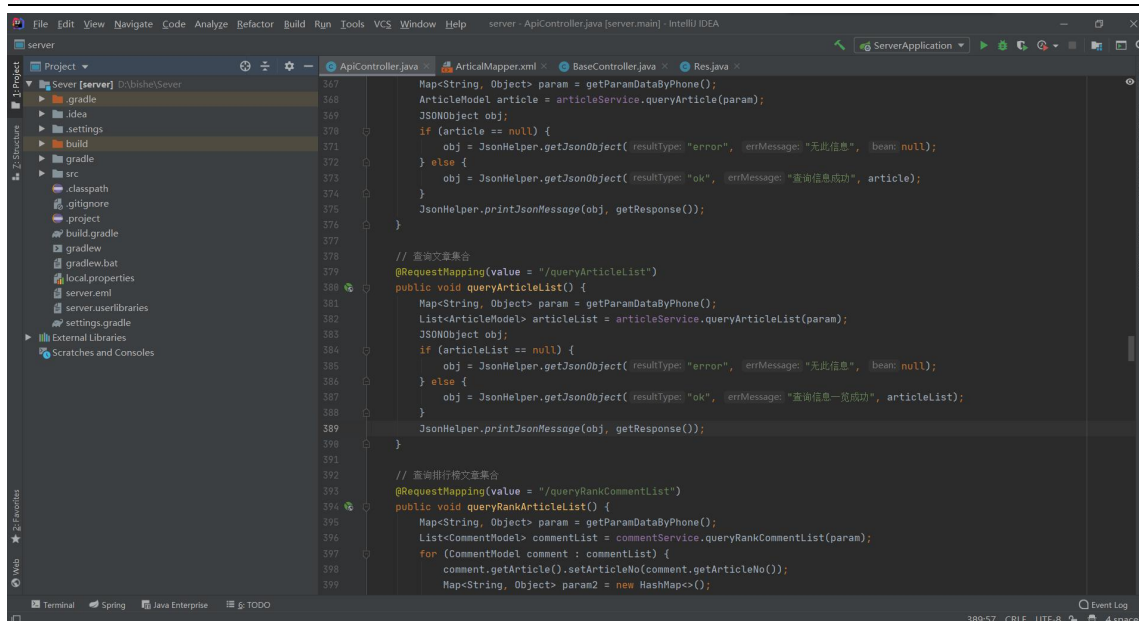


图 2-1

IDEA 支持的语言有 java、scala、Groovy,当然也可以通过安装插件来支持更多的语言；支持的数据库有 mysql、Oracle、Sql Server 等；支持的框架有 spring mvc、GWT、struts、hibernate 等。

### 2.2.2 Android Studio 介绍

本次项目客户端采用 Android Studio 来开发，Android Studio 是谷歌推出的一款基于 IDEA 的集成开发环境，比 IDEA 更进一步的是这款 IDE 对于 Android 开发是极其友好的，在构建 Android 项目上具有其他 IDE 无法比拟的高效率。由于 Android Studio 强大易用性和高效率，谷歌在 2015 年就停止了对 eclipse 版本的 IDE 的维护，旨在引导开发者使用这款统一的 IDE 来进行 Android 应用的开发。Android Studio 开发的应用可以运行在任何搭载了 Android 系统的设备上，包括手机、平板、谷歌眼镜、智能电视甚至是车载系统。在开发方面，这款 IDE 提供了 Android 专属的优势：项目的快速重构和快速修复，其开发界面如图 2-2。

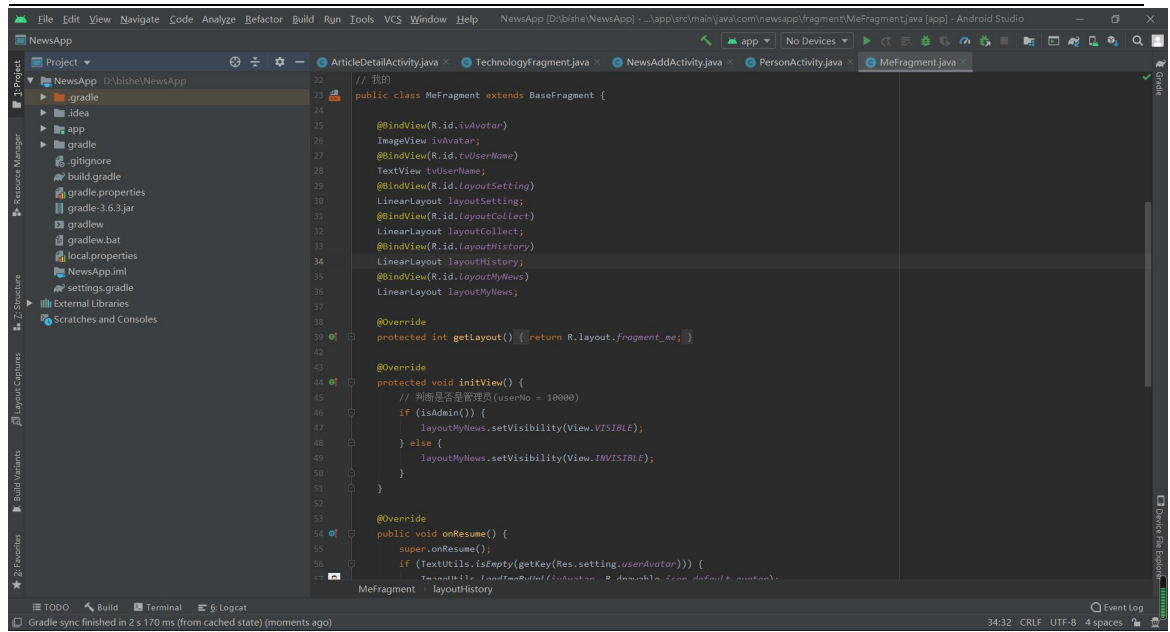


图 2-2

### 2.2.3 开发环境的配置

IDEA 中的 gradle 文件配置如下：

```
dependencies {  
    implementation 'org.springframework.boot:spring-mvc-boot-starter-thymeleaf'  
    implementation 'org.springframework.boot:spring-mvc-boot-starter-web'  
    implementation 'org.mybatis.spring.boot:mybatis-spring  
mvc-boot-starter:1.3.2'  
    implementation 'com.github.pagehelper:pagehelper:5.1.2'  
    implementation 'org.apache.commons:commons-lang3'  
    implementation 'org.apache.commons:commons-email:1.5'  
    implementation 'commons-io:commons-io:2.4'  
    implementation 'com.alibaba:fastjson:1.2.21'  
    implementation "org.springframework.boot:spring-mvc-boot-devtools"  
    implementation 'jstl:jstl:1.2'  
    implementation 'org.apache.poi:poi:3.11'  
    implementation 'org.apache.poi:poi-ooxml:3.9'  
    implementation 'commons-fileupload:commons-fileupload:1.3.1'  
    implementation 'commons-dbcp:commons-dbcp:1.2.2'  
    implementation 'org.springframework.boot:spring  
mvc-boot-configuration-processor:2.1.5.RELEASE'  
    providedRuntime 'org.springframework.boot:spring-mvc-boot-starter-tomcat'  
    runtime 'mysql:mysql-connector-java'  
    testImplementation 'org.springframework.boot:spring-mvc-boot-starter-test'  
}
```

Android Studio 中的 gradle 文件配置如下：

```
dependencies {  
    implementation fileTree(include: ['*.jar'], dir: 'libs')  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
```

androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  
 implementation 'org.apache.commons:commons-lang3:3.8.1'  
 implementation 'commons-io:commons-io:2.4'  
 implementation 'com.jakewharton:butterknife:8.8.1'  
 annotationProcessor 'com.jakewharton:butterknife-compiler:8.8.1'  
 implementation 'com.android.support:recyclerview-v7:28.0.0'  
 implementation 'com.android.support.constraint:constraint-layout:1.1.3'  
 implementation 'com.android.support:appcompat-v7:28.0.0'  
 implementation 'com.android.support:support-v4:28.0.0'  
 implementation 'com.android.support:design:28.0.0'  
 implementation 'com.jzxiang.pickerview:TimePickerDialog:1.0.1'  
 implementation 'com.jpardogo.materialtabstrip:library:1.1.1'  
 implementation 'com.github.smarxpan:NotchScreenTool:0.0.1'  
 implementation 'com.haibin:calendarview:3.6.6'  
 implementation 'com.yanzhenjie.recyclerview:support:1.3.2'  
 implementation 'com.yanzhenjie.permission:support:2.0.1'  
 implementation 'com.github.bumptech.glide:glide:3.7.0'  
 annotationProcessor 'com.github.bumptech.glide:compiler:4.9.0'  
 implementation 'com.zaaach:tabradiobutton:1.0.2'  
 implementation 'com.gyf.immersionbar:immersionbar:3.0.0'  
 implementation 'com.gyf.immersionbar:immersionbar-components:3.0.0'  
 implementation 'com.scwang.smartrefresh:SmartRefreshLayout:1.1.0'  
 implementation 'com.scwang.smartrefresh:SmartRefreshHeader:1.1.0'  
 implementation 'com.github.czy1121:readmoretextview:1.0.0'  
 implementation 'org.apache.commons:commons-lang3:3.8.1'  
 implementation 'commons-io:commons-io:2.4'  
 implementation 'com.yanzhenjie.permission:support:2.0.1'  
 implementation 'com.alibaba:fastjson:1.2.51'  
 implementation 'com.lzy.net:okgo:2.1.4'  
 implementation 'com.zhy:okhttputils:2.6.2'  
 implementation 'top.zibin:Luban:1.1.3'

```
implementation 'com.jzxiang.pickerview:TimePickerDialog:1.0.1'
implementation 'com.afollestad.material-dialogs:core:0.9.6.0'
implementation 'com.afollestad.material-dialogs:commons:0.9.6.0'
implementation 'com.zaaach:tabradiobutton:1.0.2'
implementation 'org.jsoup:jsoup:1.11.3'
}
```

## 第三章 概要描述

### 3.1 功能模块描述

#### 3.1.1 功能名称及对应视图展示

本次 app 的开发包括的功能模块有以下 11 个：

- 1.注册以及修改密码
- 2.新闻的分类浏览
- 3.新闻浏览排行榜
- 4.新闻的发布
- 5.爬虫模块
- 6.数据推荐模块
- 7.新闻评论
- 8.新闻收藏
- 9.新闻检索
- 10.历史记录
- 11.个人资料管理

部分界面如图 3-1，从左到右依次为：登录、注册、新闻分类、新闻浏览、排行榜、用户个人操作、个人资料管理、历史记录



图 3-1



图 3-1

### 3.1.2 功能介绍

- 注册以及修改密码：用户对于趣新闻 app 的账号注册、登录以及个人页面的密码修改。
- 新闻的分类浏览：用户可以根据自己的喜好点击感兴趣的分类进行浏览，经过推荐算法筛选的新闻被显示先推荐栏里。
- 新闻浏览排行榜：根据用户收藏量和评论量来对新闻进行热门度排行，用户可以直接根据排行榜来浏览热门新闻。
- 新闻的发布：此功能只有管理员有权限操作。管理员登录账号，可以自主发布新闻，也可以点击刷新按钮再次从百度新闻上爬取新闻。
- 爬虫模块：通过 Jsoup 组件在前端进行数据爬虫，将取得的数据发送到 app 上。
- 数据推荐模块：通过基于内容的协同过滤对用户过往浏览的情况进行分析，将结果整合到推荐栏。
- 新闻评论：用户可以点击评论按钮对新闻进行评论
- 新闻收藏：用户可以长按某一条新闻，弹出对话框后确认自己是否要收藏该新闻。

- 新闻检索：用户可以提供关键字，在搜索栏输入要搜索的内容以检索目前爬取到 app 中的新闻情况。

- 历史记录：用户可以查看或者删除搜索记录和浏览记录。

- 个人资料管理：用户可以进入个人资料页面，点击对应栏来进行个人信息的修改。

## 3.2 开发及运行环境

硬件平台：常规搭载 Windows 的 PC 机

软件要求：

使用语言：JAVA

数据库：MYSQL

操作系统：win10 专业版

IDE：IntelliJ IDEA、Android Studio

开发框架：spring mvc

## 3.3 准备内容

在开发开始之前，我对于推荐系统、spring mvc 框架以及 Android 开发都不是特别熟练，尤其是推荐系统之前没有接触过。因此我首先阅读了项亮所著的《推荐系统实践》[5]，了解了基本的基于用户和内容的协同过滤算法详细原理，书中所用算法都是用 python 编写，而且有官方的数据集可作为训练集来验证算法，我由于是个人开发，而且也找不到那么详实的官方数据或者调查问卷数据，因此就把主要原理抽出来用 java 代码重写了一下，在具体实现上做了一些简化，以此来更加方便的应用到我的项目中来。对于 Android 开发这一块，一直都不是特别熟练，这次开发是把大二曾经做的两个网课项目结合起来进行了简单的整理，因为当前 app 的界面设计架构大同小异，我就在之前的项目基础上进行了改进，形成了这个毕业设计项目的初始框架。对于 spring mvc 框架，通过其他小型项目的练习，熟练了使用方法。至此，开发本次项目所需的准备工作都已完成。



## 第四章 功能实现

### 4.1 登录注册功能

#### 4.1.1 功能说明

本次设计的登录分为管理员和普通用户，管理员账号固定为 10000，密码为 123456，管理员账号是在一开始就写进数据库的，不需要额外注册。普通用户需要登录注册界面进行注册。管理员和普通用户登录后的界面是略有不同的，管理员可以发布新闻，爬取新闻，而普通用户则不具备以上权限，只能进行新闻的浏览、搜索和个人信息的管理。

#### 4.1.2 数据库表结构说明

User 的集合（对应于 mysql 数据库的 tb\_user 表），表权限没有作限制，因为读写接口在应用中仅仅开放某个用户自己那一条。管理员账号不允许注销。如下表 4-1：

表 4-1：tb\_user 表结构

字段名	字段类型	字段说明	对应 mysql 的字段
userNo	String	用户编号（主键）	user_no(varchar)
userName	String	姓名	user_name(varchar)
password	String	密码	password(varchar)
userAvatar	String	头像	user_avatar(text)
label	String	标签	label(text)
userSex	String	性别	user_sex(text)
userBirth	String	出生年月	user_birth(text)

#### 4.1.3 关键代码说明

前端 LoginActivity 中的关键代码如下表 4-2：

表 4-2 LoginActivity 关键代码

函数名	函数功能
onResume	判断当前用户是否登录
onClickView	根据传入参数来判断跳转到注册页面还是登录页面
onBtnLoginClick	将接受的数据进行简单的判空操作，若不为空则将用户名和密码整理成 JSON 数据包，将其缓存后把数据提交到服务器
onTvRegisterClick	跳转到注册页面

前端 RegisterActivity 中的关键代码如下表 4-3:

表 4-3 RegisterActivity 关键代码

函数名	函数功能
onViewClicked	根据点击的按钮来判定执行哪个函数
onIvAvatarClick	处理点击头像后弹出手机的选择相片或者拍照的对话框
onRegisterSexClick	处理点击性别选择后的动作
onBtnRegisterClick	处理点击注册按钮后进行的动作，将数据进行异常检查，无异常后将数据提交到服务器
onTakePhotoClickListener	根据传入的参数来判定执行调用相机的函数还是执行调用相册的函数
onActivityResult	回调函数
execCamera	调用手机的相机
execGalery	调用手机的相册
showDatePicker onMultiListDialogResult	分别为调用年月日选择器、标签多选窗口

后端 UserService 中的关键代码如下表 4-4

表 4-4 UserService 关键代码

函数名	函数功能
addUser	将从前端接受到的用户数据插入数据库
updateUser	更新用户数据
deleteUser	删除用户数据

#### 4.1.4 实现逻辑

注册和登录各一个界面，第一次启动 app 会自动跳转到登录页面，如果已经注册了账号就直接输入用户名和密码，点击登录后数据会通过 spring mvc 框架提交到后台服务器，服务器进行核对后将信息反馈到前端进行登录；如果没有注册过账号，就点击注册按钮，此时会跳转到注册界面，在每个项目都填写完毕后点击注册，数据提交到服务器后，后台将数据插入到数据库中，并反馈给前端显示注册成功，这时候用户可以使用刚才注册的账号来进行登录。

#### 4.1.5 异常数据提醒

异常登录情况如下表 4-5:

表 4-5 异常登录内容

异常原因	异常提醒
账号存在但是密码不匹配	登录失败，请检查账号和密码是否正确
账号不存在	登录失败，请检查账号和密码是否正确
账号或者密码输入为空	登录失败，请检查账号和密码是否正确

注册异常情况如下表 4-6:

表 4-6 注册异常内容

异常原因	异常提醒
账号经过检查已存在	账号已存在
头像为空	内容不能为空
账号为空	内容不能为空
密码为空	内容不能为空

表 4-6 注册异常内容（续表）

异常原因	异常提醒
性别为空	内容不能为空
出生日期为空	内容不能为空
标签为空	内容不能为空

## 4.1.6 实现结果

实现结果如下图 4-1:



图 4-1 登录注册实现结果

## 4.2 新闻的分类浏览功能

### 4.2.1 需求分析

我们在阅读新闻的时候，有时是漫无目的的，有时是有明确的需求的。例如最近新冠肺炎的新闻比较热门，用户对这种常规的新闻比较感兴趣，有的用户则只想看科技类的新闻。对于这类用户，我们就需要为他们把新闻分好类别，以便于他们可以根据自己的需求来阅读新闻。而对于另外一些用户，他们打开 app 只

是为了看一下当前有什么新闻，并没有明确自己要看什么，这种时候就需要将各类新闻随机抽取一些整理到一个栏目中，供用户来观看。因此，在经过上边的分析思考后，决定对被抓取到 app 中的新闻进行分类。

#### 4.2.2 可行性分析

Android 中有许多可用的滑屏切换实现方式，这里采用最为常见的 `ViewPager + n*Fragment` 的方式来组织大面板，而大面板中标签处理需要使用 `FragmentPagerAdapter` 来实现滑屏的切换，因为是官方封装好的库，所以只需要在创建标签页对应的适配器类，把需要覆盖的方法写好再添加自己的业务逻辑就可以很方便的实现标签页的效果以及滑屏切换。

#### 4.2.3 关键代码说明

前端部分如下：

MainActivity 页面关键代码如下表 4-7

表 4-7 MainActivity 页面关键代码

函数名	函数功能
<code>swiithFragment</code>	切换 fragment 页面
<code>onViewClicked</code>	根据点击的按钮来进行首页、排行榜、新闻发布、我的信息这四个 tab 页面的切换
<code>onClickMenuListener</code>	监听按钮的点击，将点击的按钮颜色加深，未点击的不作处理。同时监听发布按钮是否被点击，若被点击则去跳转到对应的页面进行处理

HomeFragment 页面关键代码如下表 4-8

表 4-8 HomeFragment 页面关键代码

函数名	函数功能
<code>initTabs</code>	初始化标签
<code>setSelectColor</code>	设置被选中标签的颜色
<code>onClickView</code>	如果点击了搜索按钮，就跳转到搜索页面

LifeFragment 页面关键代码如下表 4-9（NBAFragment、NewsFragment、

PhoneFragment、RecommendFragment、TechnologyFragment 这些标签页由于只是爬取的内容不同，其他地方完全相同，因此这里不再列出详细解释）

表 4-9 LifeFragment 页面关键代码

函数名	函数功能
onLoad	从服务器获取文章列表
onClickArticleClick	点击某条新闻跳转进入新闻详情页面
onLongClickArticle	长按确定是否收藏该条新闻
onFabRefreshClick	通过 Jsoup 进行爬虫, 将爬取的数据显示并提交
onWarningConfirmDialogResult	将点击了收藏的新闻通过服务器提交

ArticleDetailActivity 页面关键代码如下表 4-10

表 4-10 ArticleDetailActivity 页面关键代码

函数名	函数功能
AddHistory	将查看的这条新闻的历史记录写入数据库
Onload	根据文章编号把对应的新闻从数据库中取出并显示
OnClickView	根据传入参数判定是返回还是跳到评论

后端部分如下表 4-11:

表 4-11 后端部分关键代码

函数名	函数功能
queryArticleList	查询文章列表
addCollect	将前端要收藏的新闻添加到数据库
addOrderList	添加要发布的新闻到数据库中

#### 4.2.4 实现逻辑

首先 switchFragment 这个函数，通过传入的参数来判定当前要显示的是哪个 tab，然后把当前的正在显示的 tab 隐藏,转去 show 需要显示的 tab,返回的是一个 FragmentTransaction 类型的结果,在 onViewClicked 中传入这个要显示的页面参数，

根据传入的 id 来判定填入的参数是代表哪个 tab 页面，至此实现底部栏根据 tab 进行切换；通过 Fragment 的 FragmentPagerAdapter 来实现滑动切换上部菜单，以此实现新闻的分类浏览；新闻内容通过爬虫取到数据之后整理成 JSON 对象后存入数据库，之后的前台显示是根据文章编号通过 spring mvc 来访问到后台数据库后提交给前台进行显示。关于数据爬虫的详细实现原理将在后边其他模块进行解释。

## 4.2.5 实现结果

实现结果如下图 4-2



图 4-2 新闻分类浏览实现结果

## 4.3 新闻浏览排行榜功能

### 4.3.1 需求分析

在用户刚刚使用本 app 的时候，由于系统中并没有很多用户的行为信息，因此此时的推荐就是随机的，没有根据的，这时候用户可以直接去浏览排行榜的新闻来获取资讯，因为排行榜的新闻是其他用户留下过行为信息的，排行是根据这些信息的数量来排序，显然这代表了这些新闻的热门程度，因为用户不会对一条

自己完全不感兴趣的新闻作评论。当某些新闻的评论数很多的时候，我们有理由认为大多数人对这些新闻是可能感兴趣的，因此这时候整理出这样一个排行榜，有助于刚使用 app 的用户快速找到自己想要浏览的新闻，以此获得更棒的使用体验。

### 4.3.2 可行性分析

因为许多用户可能会在阅读完一条新闻之后点击评论，那么通过统计一条新闻的评论数量就可以粗略判断这条新闻的热门程度，以此来生成一个排行榜。

### 4.3.3 数据库表结构说明

Comment 集合(对应于 mysql 数据库的 tb\_comment 表),详细定义如下表 4-12:

表 4-12 tb\_comment 表结构

字段名	字段类型	字段说明	对应 mysql 的字段
commentNo	String	评论 ID	comment_no(varchar)
userNo	String	用户 ID	user_no(varchar)
articleNo	String	文章 ID	article_no(varchar)
content	String	内容	content(text)
createTime	String	创建时间	create_time(varchar)

### 4.3.4 关键代码说明

RankFragment 页面关键代码如下表 4-13

表 4-13 RankFragment 页面关键代码

函数名	函数功能
onLoad	从数据库中取出评论信息并根据数量降序排列显示
onClickRank	监听排行榜的点击事件，用户点击排行榜新闻可以进入新闻详情页面
queryRankArticleList (后端)	查询数据库并将结果返回前端



### 4.3.5 实现逻辑

首先用户点击排行榜按钮，此时前台的 onLoad 函数中，通过 spring mvc 将请求提交到服务器，服务器收到请求后执行 queryRankArticleList 函数，内部处理时候执行查表语句，将结果返回到集合中，再讲该集合整理整 JSON 对象，最后把结果返回到前端进行显示。值得注意的是每次点击排行榜按钮，系统都会去重新查询一遍数据库以获得最新的热门新闻排行榜。

### 4.3.6 实现结果

新闻浏览排行榜实现效果如下图 4-3 所示：

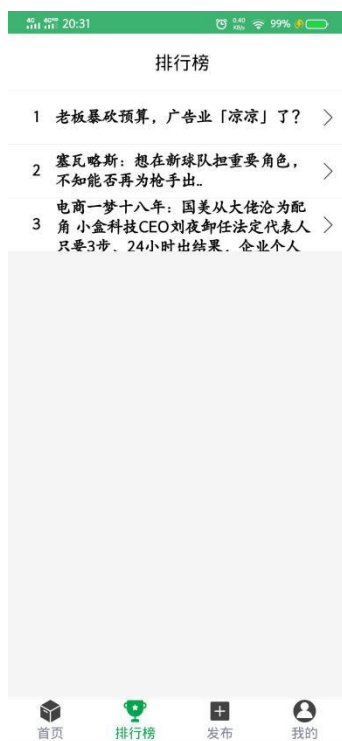


图 4-3 新闻浏览排行榜实现结果

## 4.4 新闻评论功能

### 4.4.1 需求分析

许多用户在浏览新闻后，并不是退出就了事，而是期望看到其他人关于这个新闻中所透露出信息的看法，同时也希望就一些观点与他人进行沟通。因此，对于一款新闻资讯类 app，提供评论功能就显得十分必要。同时，后台的开发人员

也希望用户去评论，因为这样就可以收集到更准确的行为特征来分析用户对某些新闻的感兴趣程度，以便更好的服务用户。

#### 4.4.2 可行性分析

该功能为常规的界面设计布局和数据库调用。

#### 4.4.3 数据库表结构说明

Comment 集合（对应于 mysql 数据库的 tb\_comment 表），详细定义如下表 4-12：

表 4-13 tb\_comment 表结构

字段名	字段类型	字段说明	对应 mysql 的字段
commentNo	String	评论 ID	comment_no(varchar)
userNo	String	用户 ID	user_no(varchar)
articleNo	String	文章 ID	article_no(varchar)
content	String	内容	content(text)
createTime	String	创建时间	create_time(varchar)

#### 4.4.4 关键代码说明

CommentActivity 页面的关键代码如下表 4-14

表 4-14 CommnetActivity 页面关键代码

函数名	函数功能
onLoad	从数据库中取出当前文章评论信息并显示
onClickView	监听按钮，根据输入判定是返回还是评论
onBtnSentClick	处理发送评论的功能
queryCommentList（后端）	查询评论集合
addComment	增加评论

#### 4.4.5 实现逻辑

评论页面的实现逻辑比较直接，首先在点击评论按钮后拉起 CommentActivity 页面，此时系统会根据当前文章的 ID 去查找对应的评论列表并显示，当用户在输入框中输入了评论后点击评论，这时候会作一些判定，我们规定一个用户只能对

一条新闻评论一次，因此这时候会去数据库中查询该文章对应的评论列表中是否有当前用户的 ID，如果有的话会提示“已评论”，并不再更新评论列表；如果用户未对该新闻作过评论那么就请求通过 spring mvc 框架被服务端接受到，服务端将该评论相关信息插入数据库并将信息反馈给前端，前端再去更新评论列表。

#### 4.4.6 实现结果

新闻评论实现效果如下图 4-4 所示：

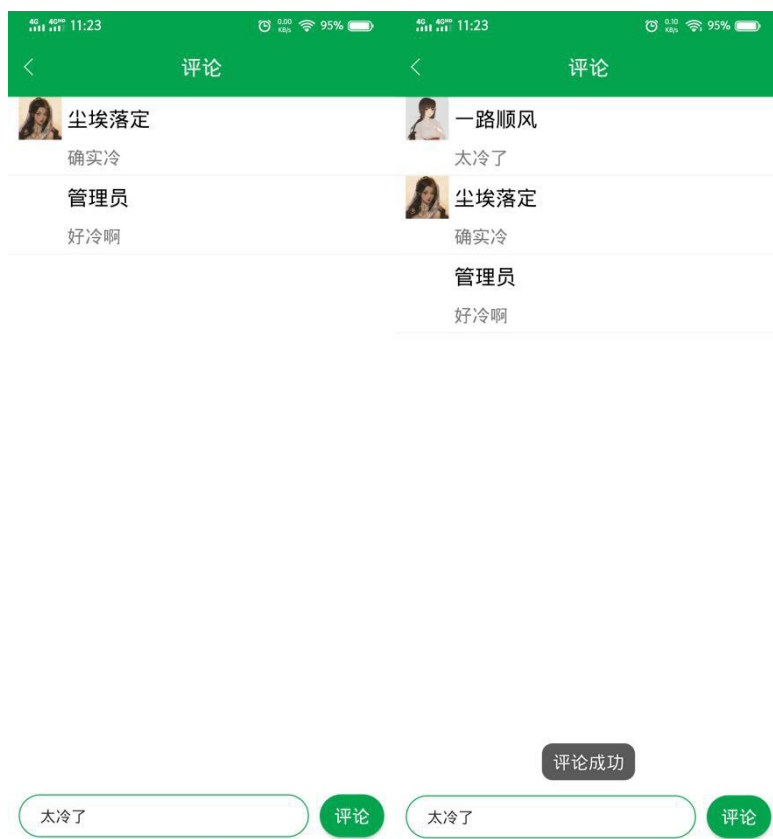


图 4-4 新闻评论实现效果

### 4.5 新闻推荐功能

#### 4.5.1 需求分析

在如今这个信息爆炸的时代，每个人都无时无刻不面临着信息过载的问题，各种条目和类别的资讯往往在我们接入互联网的那一刻就扑面而来，如果不对这些信息进行筛选和处理，我们往往会陷入混乱而不知所措。当我们作为信息的生

产者和搬运者的时候，当然希望呈现给用户的东西都是他们感兴趣的，愿意看的。因此我们就需要在推送信息之前就对用户首先有一个大致的了解，然后随着用户与产品的交互的增加而不断地深入这种了解。根据用户的喜好进行新闻资讯的推荐，以此来大大节约用户的精力和时间，提供更便捷的服务。

#### 4.5.2 可行性分析

主要算法部分我参考了项亮的《推荐系统实践》[4]，对其中的基于用户的协同过滤和基于内容的协同过滤部分作了仔细阅读。基于用户的协同过滤适用于用户信息足够充分的情况，这样便于通过各项指标来衡量算法的性能，但是本次毕业设计由于是个人开发，因为一些众所周知的原因，我也没有条件去作大量的用户调查，因此本次的设计充分参考了《推荐系统实战》中基于内容（物品）的协同过滤算法，同时在用户注册时候就必须得到用户的标签信息用来作为过滤和推荐的指标，以此来构建整个新闻推荐系统的核心部分。

#### 4.5.3 数据库表结构说明

Article 集合（对应于 mysql 的 tb\_article 表）详细定义如下表 4-15

表 4-15 tb\_article 表结构

字段名	字段类型	字段说明	对应 mysql 的字段
articleNo	String	文章 ID	article_no(varchar)
articleType	String	文章类别	article_Type(varchar)
userNo	String	用户 ID	user_no(varchar)
title	String	标题	title(text)
content	String	内容	content(text)
pic	String	图片	pic(text)
createTime	String	创建时间	create_time(varchar)

#### 4.5.4 协同过滤算法说明

##### 1. 准备知识（欧氏距离相似度）

最初是用在计算欧几里得空间中的两个点的距离，假设  $x$ 、 $y$  是  $N$  维空间中

的两个点，他们之间的欧几里得距离可以依据公式 4-1 计算[4]:

$$d(x, y) = \sqrt{\sum (x_i - y_i)^2}$$

公式 4-1

从这个公式中我们可以看出，当  $n=2$  的时候，欧几里得距离表示的是平面上两个点之间的距离。而当欧几里得的这个距离用来表示相似度的时候，可以采用公式 4-2 来进行换算，距离越小，相似度越大。

$$sim(x, y) = \frac{1}{1 + d(x, y)}$$

公式 4-2

## 2. 算法实现原理及过程

(1) 从数据库中取出所有和用户、文章有关的数据，构造用户和相关的文章数组，如下所示：

```
String[] users = new String[]{"张三", "李四", "王五", "赵六", "王七"};
```

```
String[] articles = new String[]{"文章 1", "文章 2", "文章 3", "文章 4", "文章 5", "文章 6", "文章 7"};
```

(2) 通过查找所有文章类型和用户喜欢的标签类型，构造用户对文章的兴趣矩阵（用户喜欢的标签为注册时候选择），矩阵如下所示：

```
int[][] allUserArticleStartList = {
    {1, 0, 1, 0, 0, 1, 1},
    {0, 0, 0, 1, 1, 1, 0},
    {1, 0, 0, 1, 0, 0, 0},
    {0, 1, 1, 1, 1, 0, 0},
    {0, 1, 0, 0, 1, 1, 1}
};
```

代码生成过程如下：

```
Integer[][] allUserArticleList = new Integer[row][col];
// 所有用户
List<String> users = new ArrayList<>();
List<String> articles = new ArrayList<>();
for (int i = 0; i < row; i++) {
```

```

        UserModel user = userList.get(i);

        users.add(user.getUserName());

        //取得用户标签

        String label = user.getLabel();

        // 所有文章

        for (int j = 0; j < col; j++) {

            ArticleModel article = articleList.get(j);

            articles.add(article.getTitle());

            // 用户喜欢列表

            String type = article.getArticleType();

            int value = 0;

            if (label.contains(type)) {

                //将该用户对应的文章位置置 1

                value = 1;

            }

            allUserArticleList[i][j] = value;

        }

    }

```

(3) 转换用户文章列表（取得每个用户对各个文章的兴趣向量，此向量为行向量）

```
targetUserCommentedArticles = Arrays.asList(allUserArticleCommentList[targetUserIndex]);
```

(4) 计算文章的相似度

// 4.1 计算全部文章的相似度

```

convertRow2Col();

articleABSimilaritys = new HashMap();

for (int i = 0; i < allArticleCommentList.length - 1; i++) {

    for (int j = i + 1; j < allArticleCommentList.length; j++) {

        articleABSimilaritys.put(i + "" + j,

            calcTwoArticleSimilarity(allArticleCommentList[i],allArticleCommentList[j]));

    }

}

```

```
}
```

// 4.2 计算两个文章相似度（欧氏距离）

```
private static double calcTwoArticleSimilarity(Integer[] article1Stars, Integer[]
    article2Stars) {
    float sum = 0;
    for (int i = 0; i < article1Stars.length; i++) {
        sum += Math.pow(article1Stars[i] - article2Stars[i], 2);
    }
    return Math.sqrt(sum);
}
```

// 4.3 数组行转列

```
private static void convertRow2Col() {
    for (int i = 0; i < allUserArticleCommentList[0].length; i++) {
        for (int j = 0; j < allUserArticleCommentList.length; j++) {
            allArticleCommentList[i][j] = allUserArticleCommentList[j][i];
        }
    }
}
```

（5）获取全部待推荐文章并检测这些文章中是否存在和排行榜前三重合的内容，若有，则将这些新闻过滤掉。最后将结果返回前端。

//5.3 获取全部推荐文章并排序

```
for (int i = 0; i < targetUserCommentedArticles.size(); i++) {
    if (targetUserCommentedArticles.get(i) == 0) {
        for (int j = 1; j < targetUserCommentedArticles.size(); j++) {
            if (targetUserCommentedArticles.get(j) == 1) {
                articleSimilaritys.put(i, articleABMatrix[i][j]);
            }
        }
    }
}
```

```

    }

    recommendList = new ArrayList<Map.Entry<Integer, Object>>(articleSimilaritys.entrySet());

    Collections.sort(recommendList, new Comparator<Map.Entry<Integer, Object>>() {

        @Override

        public int compare(Map.Entry<Integer, Object> o1, Map.Entry<Integer, Object> o2)

        {

            return o1.getValue().toString().compareTo(o2.getValue().toString());

        }

    });

}

```

#### 4.5.5 关键代码说明

前端 RecommendFragment 页面关键代码如下表 4-16 所示：

表 4-16 RecommendFragment 页面关键代码

函数名	函数功能
onLoad	向服务端提交请求拉取推荐栏新闻
onClickArticleClick	点击新闻进入详情页面
onLongClickArticle	长按弹出是否收藏对话框
onWarningConfirmDialogResult	若点击收藏则请求服务端将该条新闻加入数据库

后端部分关键代码如下表 4-17 所示：

表 4-17 后端部分关键代码

函数名	函数功能
queryRecommends	协同过滤算法
calcRecommendArticle	获取全部推荐文章
calcAllArticleSimilaritys	计算所有文章的相似度



表 4-17 后端部分代码（续表）

函数名	函数功能
calcTwoMovieSimilarity	计算两个文章的相似度
converRow2Col	数组行转列

#### 4.5.6 实现逻辑

本算法实现逻辑较为直接，在 RecommendFragment 页面中提出请求后，该请求被传递到服务端，服务端根据用户和文章数据，执行协同过滤算法，将计算结果再返回前端，最终用户得到了推荐栏的新闻列表。

后端的协同过滤算法部分的思路可以分以下四步：

- 1.得到用户的标签，然后根据该标签结合目前爬取的新闻生成一个用户对文章的兴趣矩阵。
- 2.利用欧式距离的计算公式将每两个文章之间的相似度计算出来，得到一个相似度矩阵（对称矩阵）。
- 3.从客户端取得此时登录的用户 ID，根据这个用户 ID 可以得到该用户的标签并从之前的矩阵当中抽取对应的向量，这个向量中包含了用户的兴趣信息。对应坐标为 0 的位置说明用户对这类新闻不感兴趣，对应坐标为 1 的位置说明用户对这类新闻有兴趣。
- 4.通过结合上边的情况，从相似矩阵中，横坐标抽取用户不感兴趣文章，纵坐标抽取用户感兴趣文章，以此可以最终得到需要给用户推荐的文章列表，最后根据相似度来给这个列表排序，从中剔除列在排行榜前三的文章，将结果返回给用户。

#### 4.5.7 实现结果

以下分别是普通用户的推荐界面和管理员的推荐界面，因为用户和文章都比较少，所以看到的推荐效果不是很明显，值得注意的是排行榜前三的新闻一定会出现在每个人的推荐列表当中。实现效果如下图 4-5



图 4-5 新闻推荐实现效果

## 4.6 新闻收藏功能

### 4.6.1 需求分析

用户有时候看到一条新闻后，觉得这条新闻传递的讯息非常重要，希望每次登陆 app 的时候都能够方便的查阅这条新闻。为了满足用户的这种需求，这里为 app 增加了新闻收藏功能，用户在浏览后如果需要收藏只需要长按该新闻在弹出的窗口中确认即可。

### 4.6.2 可行性分析

本模块的功能为常规的页面设计和数据库连接，主要就是在用户长按某一条新闻后可以弹出对应界面让用户选择是否收藏该新闻。

### 4.6.3 数据库表结构说明

Collect 集合（对应于 mysql 的 tb\_collect 表），详细定义如下表 4-18 所示：

表 4-18 tb\_collect 表结构

字段名	字段类型	字段说明	对应 mysql 的字段
collectNo	String	收藏 ID	collect_no(varchar)
userNo	String	用户 ID	user_no(varchar)
articleNo	String	文章 ID	article_no(varchar)
createTime	String	创建时间	create_time(varchar)

### 4.6.4 关键代码说明

CollectActivity 页面关键代码如下表 4-19 所示：

表 4-19 CollectActivity 页面关键代码

函数名	函数功能
OnLoad	加载收藏列表
OnClickView	返回上一级
OnClickCollectClick onWarningConfirmDialogResult	取消收藏
queryCollectList	查询数据库中的收藏信息并返回

### 4.6.5 实现逻辑

本部分功能模块主要就是在用户长按某一条新闻后，弹出一个对话框，询问用户是否要收藏该新闻，如果点击了确定，那么系统将会访问服务器，服务器执行 addCollect 函数，将要添加的新闻信息加入到数据库中，并给予前端反馈。用户在点击个人中心的收藏按钮时，系统访问服务器执行 queryCollectList 函数来查询数据库，并将结果显示到客户端。用户长按收藏列表中的某一项时候会弹出对话框询问用户是否删除该条新闻的收藏。

## 4.6.6 实现结果

新闻收藏功能的实现结果如下图 4-6，从左到右依次为主界面页面收藏、收藏列表界面、删除收藏反馈：

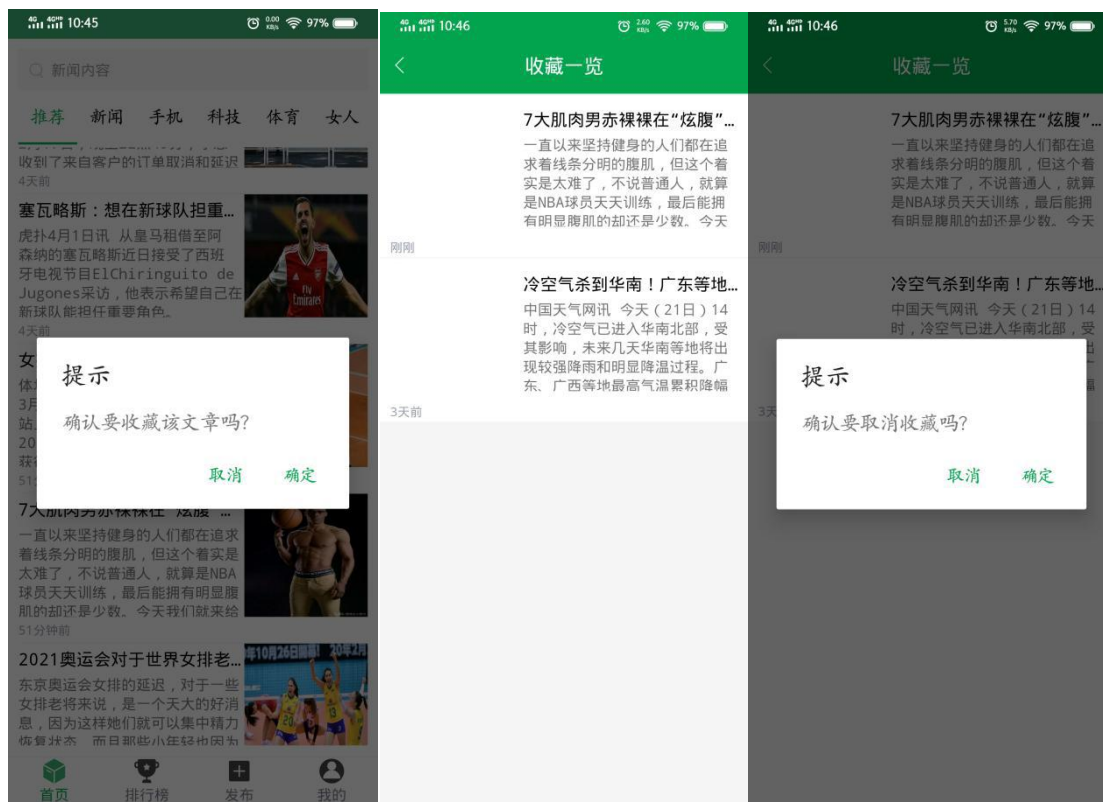


图 4-6 新闻收藏部分实现效果

## 4.7 新闻搜索功能

### 4.7.1 需求分析

用户有时候会有非常明确的阅读需求，比如某个用户今天就是想知道美国新冠肺炎的确诊情况，那么这时候我们就应该为用户提供一个搜索的功能，以便他能够很方便的阅读到这方面的新闻讯息。如果不提供这个功能，用户就只能根据标签类型来费力寻找，这样的设计师不符合用户需求也是不合理的。同时，用户可能也需要系统记忆他的搜索记录，以便下次搜索时候直接点击搜索而不用重新输入。

### 4.7.2 可行性分析

由于我在将数据爬取之后后台直接存入了数据库当中，而 mysql 数据库本身具有模糊查询的功能，因此只要得到关键字就可以根据关键字进行新闻的查询，在获得结果够返回给用户。

### 4.7.3 关键代码说明

SearchActivity 页面关键代码如下表 4-20 所示

表 4-20 SearchActivity 页面关键代码

函数名	函数功能
onLoad	加载搜索记录
onTvTrashClick	清空搜索记录
onBtnSearchClick	新闻检索

SearchResultActivity 页面关键代码如下表 4-21 所示

表 4-21 SearchResultActivity 页面关键代码

函数名	函数功能
onLoad	加载查询到的文章集合
onTvTrashClick	清空搜索记录
onWarningConfirmDialogResult	将搜索记录暂时存放在内存中

### 4.7.4 实现逻辑

SharedPreferences(简称 SP)是 Android 中很常用的数据存储方式，SP 采用 key-value(键值对)形式，主要用于轻量级的数据存储。本次工具类中使用了这个方法暂时存储搜索记录。在前端输入了要搜索的关键词后点击搜索，系统通过 springmvc 框架访问到服务端的 queryArticleList 函数，该函数访问数据库的配置详情代码截取如下：

```
<select id="queryListByMap" resultMap="articleMap">  
    select * from tb_article
```

```

<where>

    <if test="articleType != null and articleType != "">

        and article_type = #{articleType}

    </if>

    <if test="searchContent != null and searchContent != "">

        and (title like concat('%', #{searchContent}, '%') or content like concat('%',
#{searchContent}, '%'))

    </if>

    <if test="userNo != null and userNo != "">

        and user_no = #{userNo}

    </if>

</where>

order by create_time desc

</select>

```

由标红色的地方指示了本次查询的条件可以根据内容或者标题来查询，模糊查询为 mysql 内置的功能。查询成功后将结果返回。

#### 4.7.5 实现结果

新闻搜索功能实现结果如下图 4-8，中为关键词“华为”搜索结果，右为关键词“疫情搜索结果”。

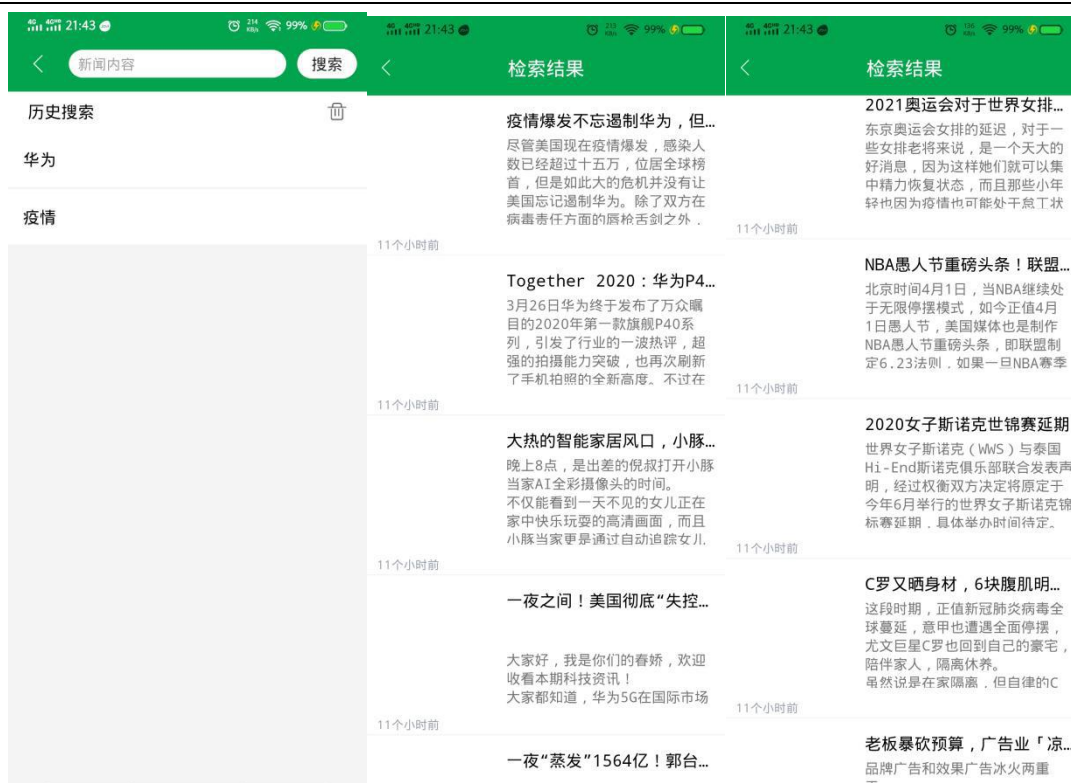


图 4-8 新闻搜索功能实现效果

## 4.8 爬虫部分

### 4.8.1 需求分析

想要进行新闻的推送,首先当然要有新闻,目前市面上主流的新闻资讯类 app 都是通过爬虫来进行数据的获取的。因为只有那些大型的门户网站才能做到大量生产信息,本次毕业设计是个人开发,因此通过爬取其他网站的最新资讯为我所用显得十分方便。将爬取的网站信息整理成 JSON 数据后可以方便的存入数据库中以供后边操作。

### 4.8.2 可行性分析

Java 爬虫使用 Jsoup 最为常见,因为它提供了丰富的选择器和便捷的操作,只要几行代码就可以爬取到需要的信息,因此本次爬虫部分就采用 Jsoup 插件来实现。

### 4.8.3 具体操作步骤

1.在 app 的 gradle 目录中,需要引入 jsoup 包



implementation 'org.jsoup:jsoup:1.11.3'

2.打开百度新闻，以“体育”为例（http://news.baidu.com/sports），爬取红色框里面的信息

利用下面代码进行连接：

```
Connection conn = Jsoup.connect("https://news.baidu.com/sports");
```

```
Document doc = conn.get();
```



图 4-9

3.以 Chrome 浏览器为例，按下 F12 网页元素审查窗口。



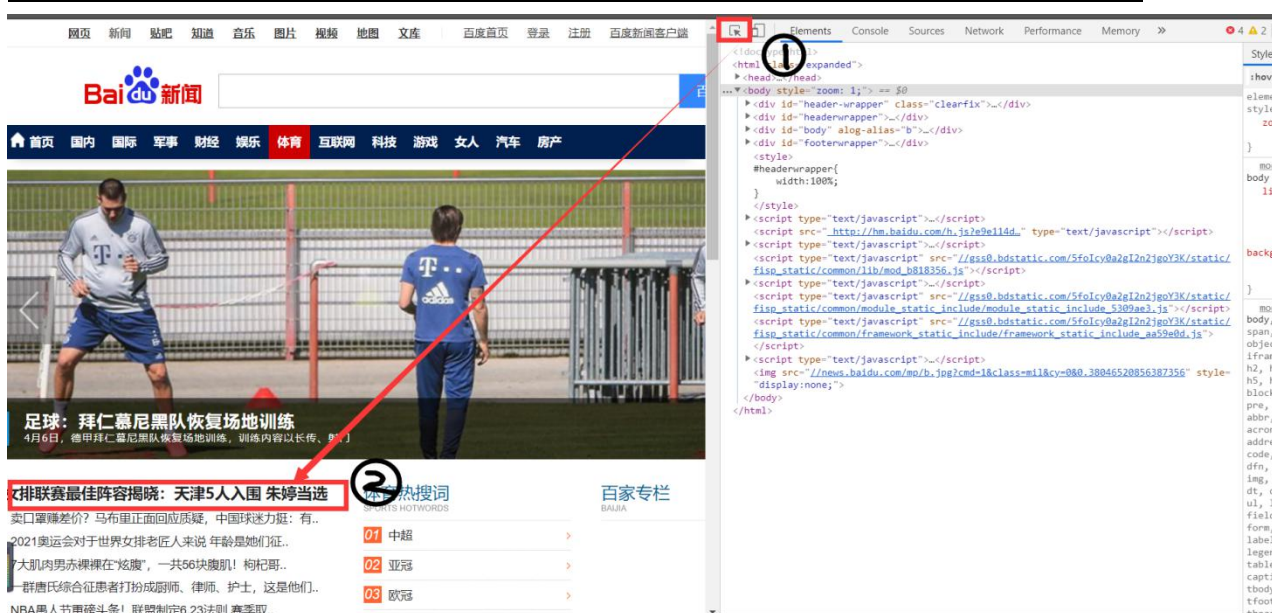


图 4-10

在图 4-10 里面用鼠标点击“1”出的查找元素 icon 按钮，然后移动到“2”所在的元素上面，点击出现下面的页面，绿色框里面的内容就是我们要爬取的内容。

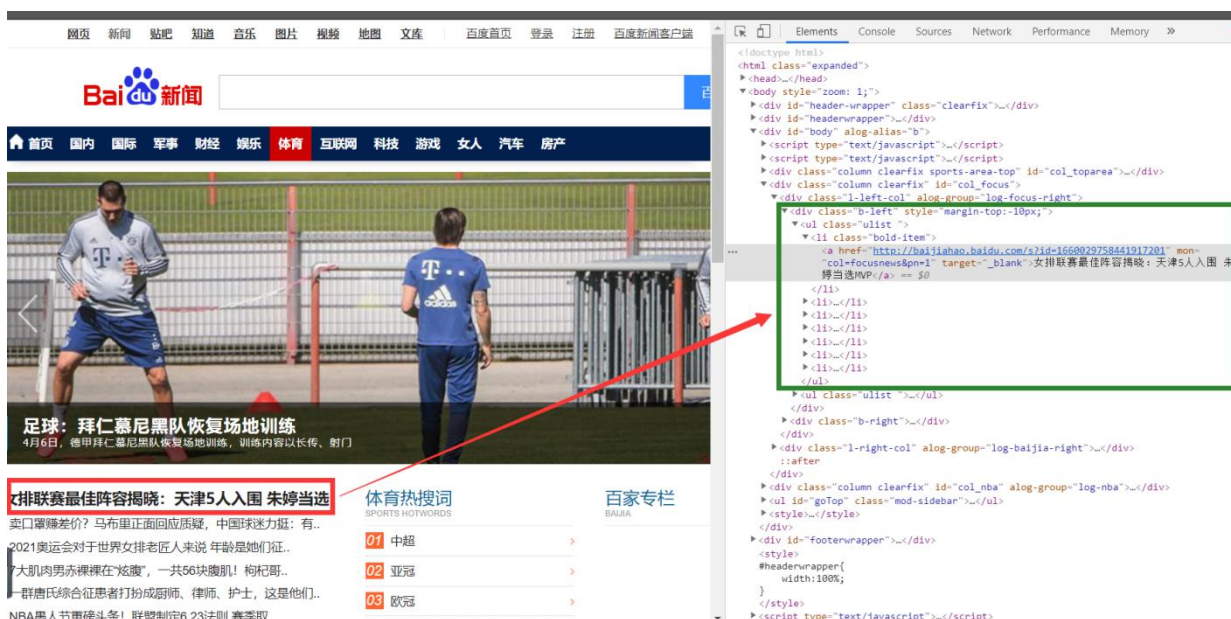


图 4-11

4. 找到绿色框里面，包裹新闻列表 ul 的外部容器 div。在该 div 上面的 class（如：b-left），可以看做第一个识别子。然后找到这个 div 下面 ul 新闻列表的 class（如：ulist），利用下面代码进行获取相关的元素集合：

```
Elements elsList = doc.select("div.b-left").select("ul[class=ulist]");
```

5. 在第 4 步的基础上，找到 ul 下面的子节点，即：li。然后查找 li 标签中的子元素的值。



图 4-12

利用下面代码进行获取取得 title 和详情页的 url:

```

for (int i = 0; i < elsList.size(); i++) {

    Elements dataElsList = elsList.get(i).select("li");

    for (int j = 0; j < dataElsList.size(); j++) {

        String title = dataElsList.get(j).text();

        String url = dataElsList.get(j).select("a").attr("href");

    }

}

```

6. 通过在第 5 步里面取得的详情页面的 url 连接，来获取图片和内容。



图 4-13

利用下面代码进行获取详情页的内容和图片:

```

// 获取详情页的内容和图片

Connection connDetail = Jsoup.connect(url);

```

```

Document docDetail = connDetail.get();

Elements elsDetailList = docDetail.select("div.article-content");

String articleContent = "";

for (int k = 0; k < elsDetailList.size(); k++) {

    // 内容

    Elements elsContentList = elsDetailList.select("p");

    StringBuilder sbContent = new StringBuilder();

    for (Element elsContent : elsContentList) {

        String content = elsContent.select("span[class=bjh-p]").text();

        sbContent.append(content);

        sbContent.append("\n");

    }

    articleContent += sbContent.toString();

}

// 图片

String pic = "";

Elements elsPicList = elsDetailList.select("div[class=img-container]");

if (0 < elsPicList.size()) {

    pic = elsPicList.get(0).select("img").attr("src");

}

```

至此，爬取数据的步骤结束了，在实际的应用中，爬取信息的不同，相对应的查找的节点也会不同。本质上原理都是一样。

#### 4.8.4 关键代码说明(详细说明已经在上边列出)

该段爬虫代码在每一个标签页中都有使用，以 LifeFragment 页面为例，见表 4-22

表 4-22 爬虫函数

函数名	函数功能
OnFabRefreshClick	在管理员点击刷新按钮后进行数据爬虫

## 4.8.5 实现逻辑

本部分功能通过使用 Jsoup 爬虫插件，利用 html 的标签特性和 Jsoup 带有的强大选择器很方便的将数据从网络上爬取出来为我所用。

## 4.9 新闻发布功能

### 4.9.1 需求分析

作为 app 的管理员，应该具有一定的新闻资讯的管理职能，资讯类 app 除了作为各类新闻资讯的搬运者，也应该适当承担新闻资讯生产者和传播者的职能，如今日头条就是从最初的搬运者变为了现在的自成一家，头条系已经成长为国内知名的资讯门户。

### 4.9.2 可行性分析

这部分功能为管理员提供了新闻发布的接口，新闻发布者登录管理员，将要发布的文章信息等填写后点击发布，该文章就会出现在对应的栏目下边，后边的用户在浏览的时候就可以看到。

### 4.9.3 数据库表结构说明

Article 集合（对应于 mysql 的 tb\_article 表）详细定义如下表 4-23

表 4-23 tb\_article 表结构

字段名	字段类型	字段说明	对应 mysql 的字段
articleNo	String	文章 ID	article_no(varchar)
articleType	String	文章类别	article_Type(varchar)
userNo	String	用户 ID	user_no(varchar)
title	String	标题	title(text)
content	String	内容	content(text)
pic	String	图片	pic(text)
createTime	String	创建时间	create_time(varchar)

#### 4.9.4 关键代码说明

NewsAddActivity 页面关键代码如下表 4-24 所示：

表 4-24 NewsAddActivity 页面关键代码

函数名	函数功能
onViewClicked	功能选择函数
onIvAvatarClick	点击头像按钮选择图片
onBtnReleaseClick	监听到发布按钮点击后把新闻同时发布到前端和后端
addOrder（后端）	访问数据库并将发布信息添加到数据库
onTakePhotoClickLister	根据传入的参数判定是调用相机还是调用相册
execCamera	调用相机
execGallery	调用相册

#### 4.9.5 异常数据提醒

NewsAddActivity 页面异常数据提醒如下表 4-25

表 4-25 NewsAddActivity 页面异常数据提醒

异常原因	异常提醒
新闻图片为空	内容不能为空
新闻类别为空	内容不能为空
发布内容为空	内容不能为空

#### 4.9.6 实现逻辑

用户点击新闻发布按钮后跳转到 NewAddActivity 页面，在用户填写好所有信息后点击发布，这时候系统会访问数据库将发布的新闻插入到数据库当中，同时将刚刚发布的新闻更新的对应栏目的资讯列表里。

#### 4.9.7 实现结果

新闻发布界面、发布效果、点击详情效果如下图 4-12 所示



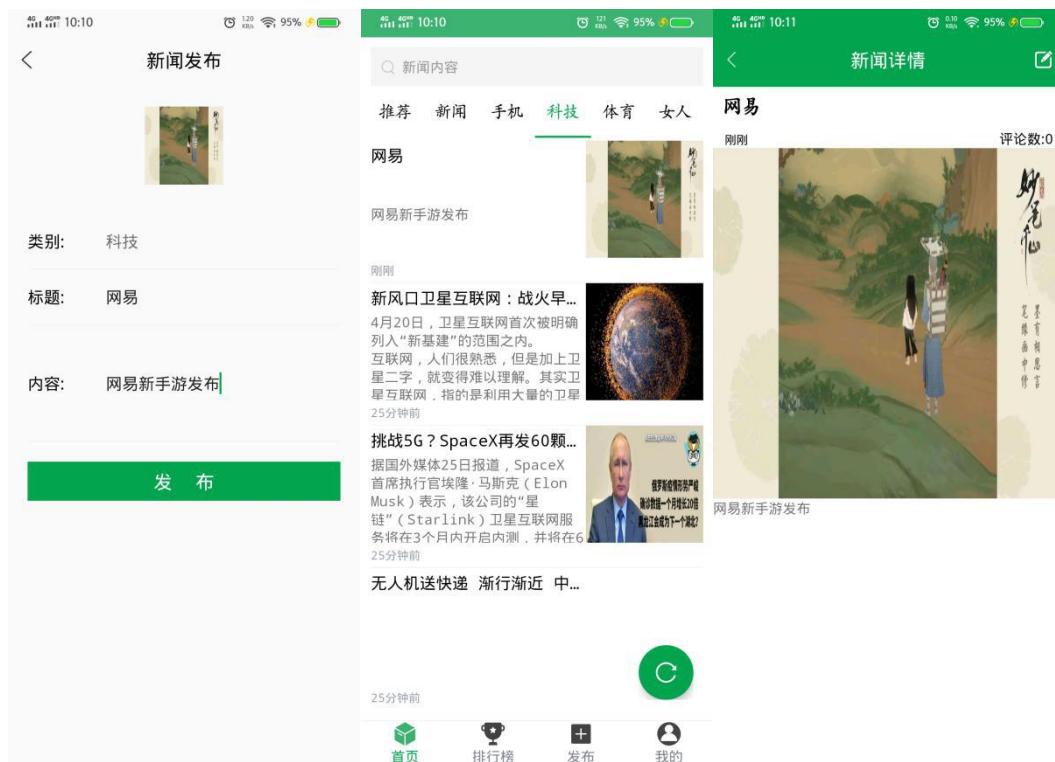


图 4-12 新闻发布实现效果

## 4.10 历史记录查询功能

### 4.10.1 需求分析

很多时候用户在相隔一段时间后会回看自己曾经浏览过的新闻的这样一种习惯，这个功能并非鸡肋，虽然很多人只是随手翻看新闻就完事，但是也有相当一部分人在浏览过一些新闻后一直存有印象，因此就有一定的概率想要去查找自己浏览过的东西，以便获取自己第一次浏览没有注意，而后来却非常需要的信息。因此，这个功能专门为这类用户而添加。

### 4.10.2 可行性分析

这部分功能涉及页面布局为简单的 `LinearLayout`，在用户点击一条新闻后，

系统为用户添加相应的浏览记录到数据库。没有涉及特别复杂的设计。

### 4.10.3 数据库表结构说明

History 集合（对应于 mysql 的 tb\_history 表），详细定义如下表 4-26 所示：

表 4-26 tb\_collect 表结构

字段名	字段类型	字段说明	对应 mysql 的字段
historyNo	String	收藏 ID	collect_no(varchar)
userNo	String	用户 ID	user_no(varchar)
articleNo	String	文章 ID	article_no(varchar)
createTime	String	创建时间	create_time(varchar)

### 4.10.4 关键代码说明

HistoryActivity 页面关键代码如下表 4-27 所示：

表 4-27 HistoryActivity 页面关键代码

函数名	函数功能
onLoad	访问数据库拉取历史记录列表
onClickHistoryClick onWarningConfirmDialogResult	删除历史记录
onClickView	返回上一级
queryHistoryList(后端)	后端数据库查询

### 4.10.5 实现逻辑

这部分实现逻辑比较直接，就是在用户点击某条新闻后将对应的文章信息添加到浏览记录表中，在点击浏览记录按钮后跳转到浏览记录页面的同时将新闻列表从浏览记录表中拉取出来显示。长按某一条浏览记录跳出对话框确认用户是否删除该条浏览记录。

### 4.10.6 实现结果

新闻浏览记录的实现结果如下图 4-13 所示：



图 4-13 历史记录查询功能实现效果

## 4.11 个人信息管理功能

### 4.11.1 功能说明

这部分功能为前边的功能做了一些整理，主页面有设置功能，这里就是用于用户注销以登录其他账号；有收藏功能，点击这个功能可以查看收藏列表；有历史记录功能，点击可以查看用户浏览新闻的历史记录；我的作品功能（仅管理员有此功能），可以查看管理员账号自己发布或者从网络中爬取的新闻列表；点击头像进入个人资料管理，用户可以在这个界面对自己的头像、昵称、密码、性别、出生年月这些信息进行修改。

### 4.11.2 关键代码说明

MeFragment 页面的关键代码如下表 4-28 所示：



表 4-28 MeFragment 页面关键代码

函数名	函数功能
onClickView	根据输入的参数来确定执行跳转的页面
initView	判定是否是管理员以提供不同的界面

PersonActivity 页面的关键代码如下表 4-29 所示：

表 4-29 PersonActivity 页面关键代码

函数名	函数功能
initView	将各组件的数据初始化
onClickView	根据传入参数来确定要执行的功能
onIvAvatarClick onTakePhotoClickListener	处理头像点击事件
onInputTextDialogResult	处理文本框输入
onListDialogResult	处理列表类对话框输入
uploadNick	访问数据库，更新昵称
uploadPassword	访问数据库，更新密码
uploadSex	访问数据库，更新性别
uploadBirth	访问数据库，更新出生日期

### 4.11.3 实现逻辑

这部分就是给用户提供一个简单的功能整理，使得用户能够很方便的去处理相关功能。点击每一个按钮都会去拉起一个新的 Activity，然后进入到那个 Activity 中去处理相应的事情。

### 4.11.4 实现结果

个人信息管理功能的实现结果如图 4-14 所示：



图 4-14 个人信息管理功能实现效果

## 第五章 总结与展望

### 5.1 总结

Android 开发其实我在大二的时候就有接触了，那个项目是在刚刚学习了 java 后利用假期的时间做的，整个项目是跟着视频结合资料来一点一点写的，基本上的框架都在那个时候弄明白了，但是有很多细节的东西没有深究，就是达到了“知道这东西这么用就是能出这个效果但并不知道是因为什么”的程度。这次毕业设计因为是在家里完成，期间遇到的最多的问题就是网络情况，可能是开发工具的升级导致“找不到可用的证书”作为一个时隐时现的 bug 一直存在项目的构建过程中，最后没办法把对应网站的证书都下载下来配置到 jdk 的目录下才解决。这次开发也同时参考了许多论文、书籍以及网课资料，又添加了许多工具类到项目中，相较于大二的进步是对于一些用法的原理能够更深刻的理解了。

Android Studio 这款 IDE 如果能适应好就是很不错的开发工具，但是有很多时候要面临莫名其妙的报错，有时候甚至重新启动一下这些错误就没有了，不过大多数时候项目构建失败是因为各个组件的版本不一致或者 jar 包下载超时。最新版本的 Android Studio 在进行布局文件的书写时候甚至可以完全不用写代码，直接选择组件然后拖动到对应的位置就好，刚开始觉得非常方便非常简单，后来发现这种方式会导致代码结构的混乱，导致自己在前端出现组件数据传递的问题时回来查看而一团乱麻的情况，于是后边布局文件就都自己写了。同时在我开始做 Android 布局文件时候就一直专门留意了市面上流行的 app 的设计结构，发现几乎所有 app 的界面布局都是类似的，即主界面上边有搜索栏和可供滑动的标签页，下边有功能选择按钮这一栏，每一个详情页面其实都可以是 LinearLayout 的布局，因为这种布局是最符合目前手机的尺寸和显示样式的。

对于 IDEA 这款 IDE 软件，我对其印象深刻是因为其对于 maven 项目和 gradle 项目构建的天然优势，因为在 eclipse 中很多时候想要创建一个 springmvc 框架的项目需要手动配置很多东西，而这些操作我并不能记得那么深刻，都是在需要的时候去网络上搜索教程然后跟着配置；而 IDEA 在项目创建时候项目结构就是完整的，配置框架等也十分简单，只需要在 build.gradle 文件中将对应框架的包引入进来点击构建，IDEA 就会自动的把需要的包全部下载并导入，十分方便。

在动手构建项目初期，我先后阅读了《Android 从入门到精通》[6]《基于 Jsoup 的通用网页采集系统的设计与实现》[7]以及《基于 jsoup 的 Web 页面自适应转换系统的实现》[13]，观看了教学视频《Jsoup 项目实战》[8]对于 Android 的基础开发和如何利用 java 进行爬虫有了一个初步的了解，方便了后续开发的进行。

对于推荐系统这一部分，主要理论是参考了《推荐系统实践》[5]、《个性化推荐算法设计》[9]、《统计学习方法》[10]，实现方式因为我对于 python 不是很熟悉，就采取 java 来实现，期间也参考到了许多的网课视频教程，最主要研习的是《Android 高级应用开发课程》《清华大学--数据挖掘：推荐算法》[7]这两个课程。

在论文的组织结构上，参考了《软件工程》[11]这本书，每个功能部分都严格遵循需求分析、可行性分析、数据库结构说明、关键代码说明、实现逻辑、实现效果这几个步骤，让我在写论文的时候不至于没有头绪。

最后想说一下推荐系统的部分，比较可惜的是这次推荐算法最后的结果因为用户数量太少，所以最后算出的推荐列表中，前边的推荐指数都挺接近的，这就

导致在实际使用时候只有排行榜部分起到了很好的热门新闻推荐效果。我因为之前按照视频教程上的说明，从那个视频指定的网站上下载到了一个训练集(电影推荐系统的数据)，使用 python 在执行协同过滤算法之后，确实能起到一定的推荐效果，那一份数据是有 4000 多条，因此我对这个算法最后的效果是肯定的。诚然，市面上那些成熟的推荐类应用使用的算法要比本文使用的复杂许多，评测标准也更加严格，即便这样，我们日常使用时候依然会有生硬的感觉。因为其中许多指标都是在统计数据出来后根据大多数情况来加入的，有经验成分在里边，其实并不是完美的适用于所有人。另一方面，本文所使用的推荐算法是基于用户的兴趣情况来进行计算并推荐的，但其实在实际生活中，有大量的用户并没有体现出明显兴趣意向的项目，用户并非对他们不感兴趣，甚至有可能是非常有兴趣的，如果能有算法能够高效的将这类项目推荐给用户，那么用户就会有惊喜感。但是就目前市面应用推荐类算法的主流软件中，在这一方面的实际效果并不理想。对于这类算法，其前景是广阔的，但是鉴于目前的情况，有理由认为其还有很大的发展空间。

## 5.2 展望

通过这次毕业设计，我的个人技能又得到了很大的提升，同时通过查阅资料和观看学习视频，自学能力也更加加强了。对于我以后从事的职业，学习将是一个终身的话题，只有不断学习，不断实践，才能不断提升自己的竞争力，在未来的生活和工作中才能有更好的表现。同时也希望能够通过自己的努力，写出一款能够给更多人提供服务的好用的实用的软件。

## 参考文献

- [1]Rui Ren,Lingling Zhang,Limeng Cui,Bo Deng,Yong Shi. Personalized Financial News Recommendation Algorithm Based on Ontology[J]. Procedia Computer Science,2015,55.
- [2]吴想想. 基于 Android 平台软件开发方法的研究与应用[D].北京邮电大学,2011.
- [3]孙骏雄. 基于网络爬虫的网站信息采集技术研究[D].大连海事大学,2014.
- [4]曹一鸣. 基于协同过滤的个性化新闻推荐系统的研究与实现[D].北京邮电大学,2013.
- [5]项亮.推荐系统实践.[D].人民邮电出版社.2012.
- [6]孙更新.Android 从入门到精通.[D].电子工业出版社.2011.10
- [7]毛凯. 基于 Jsoup 的通用网页采集系统的设计与实现[D].电子科技大学,2015.
- [8]网课《Android 高级应用开发课程》《清华大学--数据挖掘：推荐算法》《Jsoup 项目实战》
- [9]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.
- [10]李航.统计学习方法.[M].清华大学出版社
- [11]Ian Sommerville.软件工程（原书第 9 版）[M].机械工业出版社出版,2011
- [12]MySQL 查询优化的研究和改进[D]. 孙辉.华中科技大学 2007
- [13]顾问,曹阳.基于 jsoup 的 Web 页面自适应转换系统的实现[J].电子技术与软件工程,2017(18):200-202.

## 致谢

经过了很长一段时间的学习和努力，我终于完成了这次的毕业设计的编写和毕业论文的写作，首先想向在这次设计过程中给我提供帮助的柴胜老师表示特别的感谢，感谢他能在今年这个特殊的时期在各种方面给予我的指导。同时也很感谢学校能在疫情期间开放了学校的vpn,使得我能够在家里访问校内图书馆和上知网等学术网站查阅资料。

当然，还要感谢在我进行这次毕业设计遇到问题的过程中给予我建议和帮助的同学们，最后感谢尚硅谷提供的免费课程《电影推荐系统实战》以及《硅谷新闻》，通过这两个项目的跟进学习，使得我对于app的开发以及推荐算法的使用有了更加深入的理解。

通过这次的毕业设计，使我能够系统的梳理一个新闻资讯类app的前后端实现，当然由于个人能力有限，我只是把今日头条中的部分功能作了简化实现，实际的应用场景肯定比本文所使用的技术要先进的多、严谨的多。这次设计使我收获良多，因此想要再次感谢在这次的设计和开发中为我提供帮助的所有同学和老师。