2020 年夏季 Java 小学期大作业

郑凯文 2018011314

1 代码结构

此次双人大作业我申请了单人组队，所有功能均由我一人完成。除布局等资源文件之外，代码大致分为四个模块：

界面与显示

-com.java.zhengkw.newsList 新闻列表

-com.java.zhengkw.newsPage 新闻详情

-com.java.zhengkw.search 新闻搜索

-com.java.zhengkw.history 浏览历史

-com.java.zhengkw.epidemicData 疫情数据

-com.java.zhengkw.epidemicEvents 疫情事件聚类

-com.java.zhengkw.epidemicExperts 知疫学者

-com.java.zhengkw.epidemicExpertsDetail 知疫学者详情页面

-com.java.zhengkw.epidemicGraph 疫情图谱

-com.java.zhengkw.epidemicGraphDetail 疫情图谱详情页面

数据类型与数据库

-com.java.zhengkw.data

--EpidemicData 疫情数据类

--EpidemicDataPersistenceContract 一些与疫情数据有关的常量

--EpidemicExpert 知疫学者类

--EpidemicGraph 疫情图谱类

--News 新闻类

--NewsDataSource 新闻来源接口，同时定义了一些回调函数

--NewsDbHelper 继承了 SQLiteOpenHelper，通过其直接访问新闻的 SQLite 数据库

--NewsPersistenceContract 一些与新闻/数据库有关的常量

--NewsLocalDataSource 新闻的本地来源，与数据库交互

--NewsRemoteDataSource 新闻的远程来源，从网络接口获取信息

--NewsRepository 新闻库，全权负责从数据库/网络接口获取新闻信息，是界面的唯一新闻信息源，二者通过回调函数进行多线程信息传输

Json 处理、文本处理等类库

-com.java.zhengkw.utils

--ActivityUtil 将 Fragment 添加到 Activity 中的辅助类

--DatabaseManager 一个多线程安全的 SQLite 管理器

--EpidemicDataUtil 将疫情数据的 Json 解析为数据类型

--EpidemicExpertsUtil 将知疫学者的 Json 解析为数据类型

--EpidemicGraphUtil 将疫情图谱的 Json 解析为数据类型

--LoadingDialog “加载中”对话框

--NewsDataUtil 将新闻的 Json 解析为数据类型

--NewsJsonObject,NewsListJsonObject 使用 Jackson 解析 http 流 Json 的装载容器

--NewsShareActivity 新闻分享对话框

--StringUtil 一些文本处理

聚类算法

-com.java.zhengkw.cluster

--Clusterer 聚类器，传入新闻列表后返回聚类后的新闻列表与关键词

--DBSCANClusterer DBSCAN 算法类（已弃用）

--KMeans KMeans 算法类

后三个模块的每个类已进行简要描述，而界面与显示模块的每个包下有两个或三个类：XXXActivity,XXXFragment(,XXXPresenter)。若需要显示的对象是内存中的即时数据（也就是不会卡界面），只需要 Activity 和子界面 Fragement。而对于需要从数据库/网络中获取的不稳定数据源，采取了 **MVP 模式**，将 Model（数据层）、View（界面显示）和 Presenter（向界面传递数据）分离，View 不能直接访问 Model 层，而需要向 Presenter 发送请求，当数据准备完毕时，Presenter 将数据传给 View，并唤醒 View 的主线程完成刷新。

2 具体实现

功能完成度：

基础功能	系统支持	要保证程序在安卓机上正常运行，测试过程中程序不崩溃。	8	运行流畅不卡顿	2
	页面布局	布局合理，点击处理正确	8	美观，图片布局合理	2

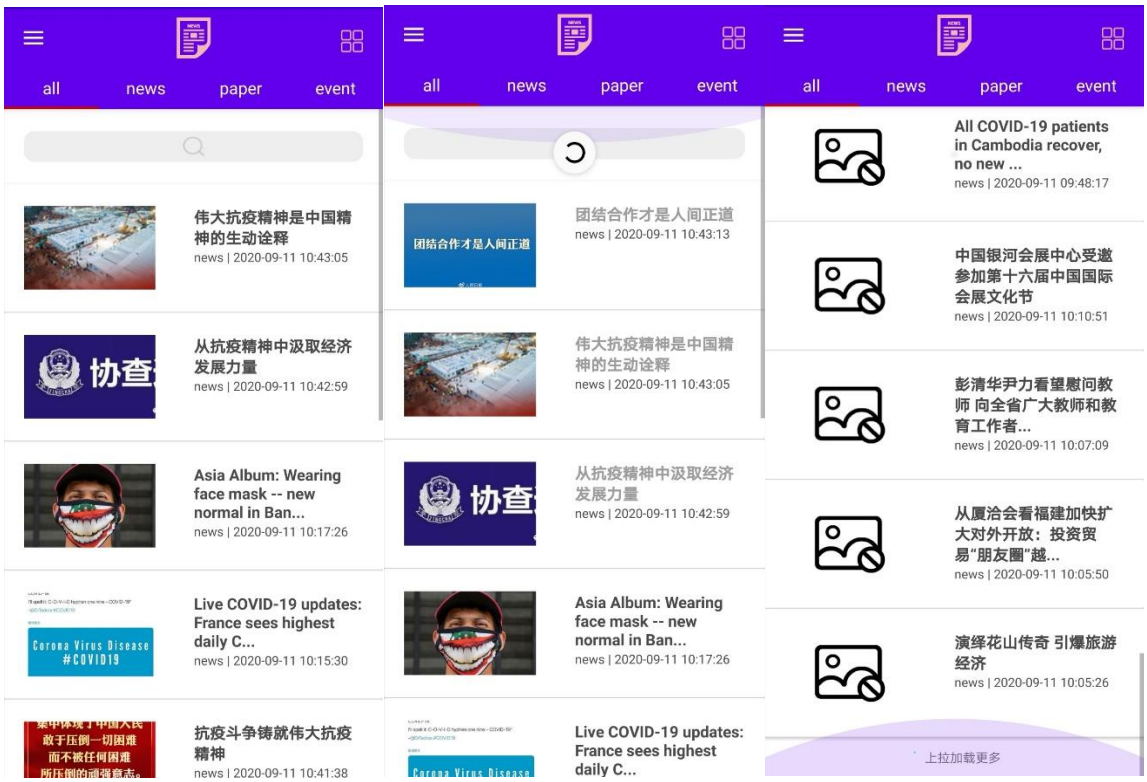
	分类列表	删除和添加操作	4	修改时有动态特效（参照今日头条）	1
	新闻列表	正确显示新闻列表的消息，布局和展示，点击进入新闻详情页面正确。	8	布局合理美观	2
		实现新闻的本地存储，看过的新闻列表在离线的情况下也可以浏览。新闻是否看过的页面灰色标记。	8	本地可存储新闻数量大响应快	2
		上拉获取更多新闻，下拉刷新最新新闻。	4	上拉下拉时添加特效	1
		显示新闻的来源和时间	4	布局合理美观	1
		新闻关键词搜索，历史记录	4	搜索页面合理美观	1
	分享收藏	使用微博等 SDK 分享，新闻详情页面点击分享可以分享到常用的 app，分享内容带有新闻摘要	4	分享页面合理美观	1
扩展功能	疫情数据	全国各省疫情统计，全球各国疫情统计可视化，以表格或折线图或柱状图形式呈现	8	数据展示美观	2
	疫情图谱	对新冠疫情图谱内容进行查询和展示，需展示实体词条的描述，包含的关系和属性	8	实体词条展示美观，正确显示图片（如果有）	2
	疫情新闻聚类	对疫情相关的新闻事件进行聚类、展示关键词和聚类新闻	8	页面展示美观	2
	知疫学者	显示在疫情领域的高关注学者和追忆学者，点开可以查看学者的详细介绍	4	页面合理美观	1
额外功能	新闻列表	调用公共接口，自动为新闻配图		图片与列表信息异步加载，图片加载时列表可流畅滑动	
	开始动画	添加了程序启动时的动画效果		动画效果美观炫酷	

上表中，除红色标注外，其余均已完成。此外，为了 App 的美观和实用，我还实现了一些要求之外的附加功能，希望可以获得一定的加分。

2.1 启动时的动画

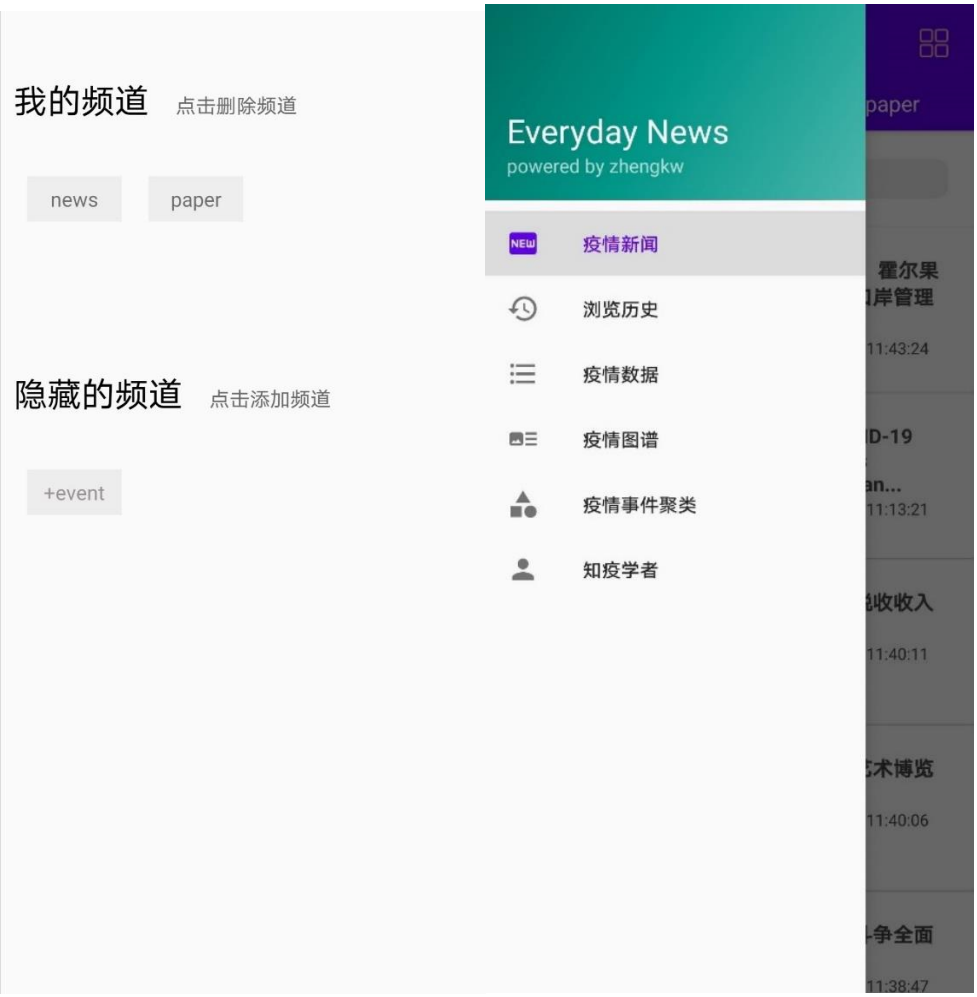
使用 Android 自带的 View Animation 完成了一个简单的动画效果，包括图标的缩放与移动，以及利用遮挡完成文字的渐出效果（见视频演示）。动画的另一个目的是留出若干秒的等待时间，本程序为了方便搜索功能的使用，在第一次运行时，会加载整个新闻列表并存储，之后进行增量更新，因此视网速快慢可能有数秒到十几秒的加载时间。

2.2 新闻列表



实现了新闻列表的分类展示，以及上拉下拉的特效。点击过的新闻标题会变成灰色。

亮点与难点：根据新闻标题，使用百度图片的搜索接口得到返回的 html，使用正则表达式匹配第一张图，自动为新闻配图；新闻加载、图片加载均是多线程，数据准备完毕后使用 handler 或 post 回到主线程，完成界面的刷新，使用 Glide 异步缓存并加载图片，因此完全不卡 UI，流畅丝滑；使用 ViewPager+RecyclerView 使得上下左右可流畅滑动，使用 SwipeRefreshLayout 监听并处理上拉和下拉事件，特效合理美观，达到与大多数 APP 相似的效果。

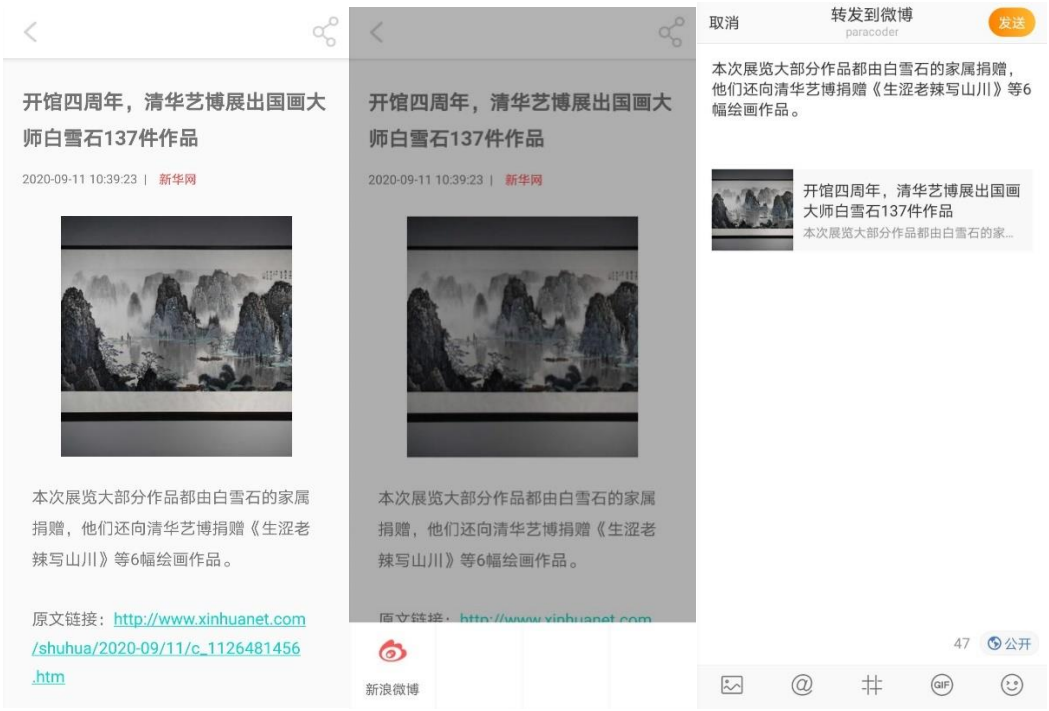


分类的增/删上，使用了一个很常见的频道管理页面，但受个人能力所限，也仅限于点击进行增加和删除，而没有实现拖拽排序、增加删除时炫酷的特效。

新闻列表页面是程序的主页，它具有 DrawerLayout 布局，其它功能（浏览历史、疫情数据、疫情图谱、疫情事件聚类、知疫学者）可通过侧边栏访问。

亮点与难点：侧边栏布局美观，菜单图标设置合理；切换到其它功能以及返回时，操作合理流畅。

2.3 新闻详情



在任何列表（浏览历史，新闻搜索）里点开新闻，都会跳转到新闻详情页面。点击右上角的分享按钮可打开分享框，只申请了容易通过的微博 SDK，并使用 ShareSDK 进行配置。

难点与亮点：新闻详情在线加载，同样使用多线程，不卡 UI；新闻被点开后，会在数据库、原列表的内存数据等各种位置进行标记，各处的数据同步，因此其余列表刷新后以及原列表中都能看到新闻标题变为灰色。

2.4 新闻搜索

在存储的新闻列表中进行搜索，响应迅速；搜索结果按照时间排序，点击过的新闻标题变为灰色。

难点与亮点：在新闻列表中点击搜索框直接进入此页面，操作顺畅；添加监听，上拉可显示下一页，每页 10 条新闻；由于新闻的时间格式不整齐，可能有 xxxx/xx/xx 或者 xxxx-xx-xx，可能带有具体时间 xx:xx:xx 也可能不带，因此对新闻的时间进行了预处理，使得它们的格式一致。

2.5 浏览历史

从数据库中读取点击过的新闻并显示，多线程异步进行。按照时间排序，分页显示，上拉查看更多。点击右上角的清除按钮将清空历史，并在数据库中清除对应标记。

2.6 疫情数据

疫情数据界面分为折线图和列表，折线图显示近 5 天的疫情数据，使用 MPAndroidCharts 进行绘制，列表使用了 RecyclerView，显示内置 10 多个国家/34 个中国地区的三项数据。

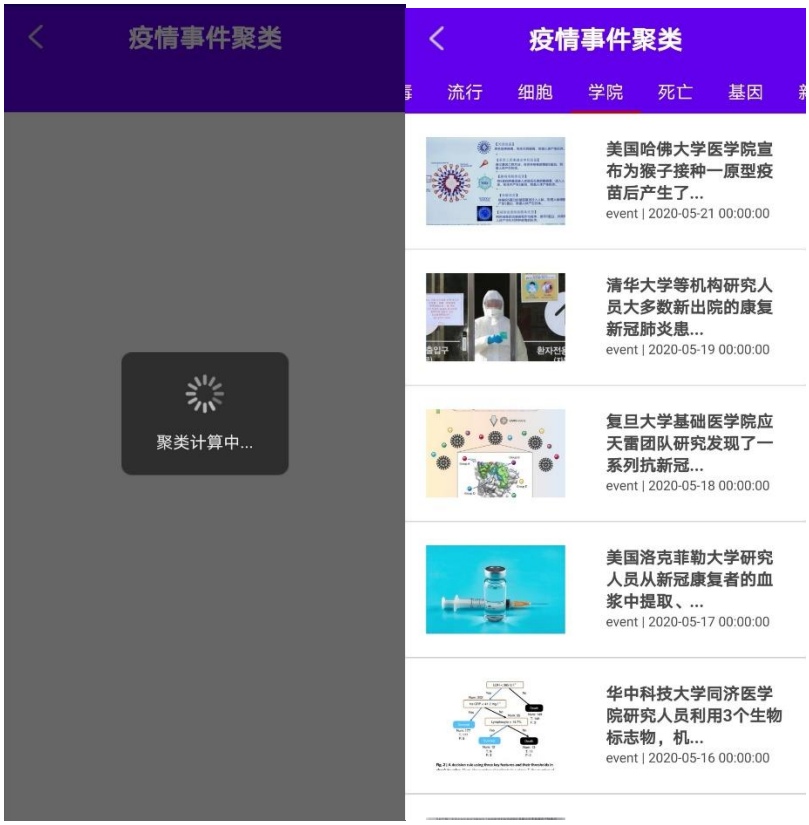
难点与亮点：疫情数据在线异步加载，由于数据较大（1M），使用一个“加载中”对话框来掩盖加载时丑陋的空白界面；列表与折线图的数据均为动态加载，列表根据感染人数降序排序，点击列表中某一项可显示对应折线图。

2.7 疫情图谱

在搜索框中输入内容并按回车进行在线搜索，显示实体列表。点开某一项进入图谱详情页面，可在 ViewPager 中查看各项关系与属性；

亮点与难点：图片显示正确，页面布局美观，关系和属性列表显示美观；加载时不卡 UI，且使用 RecyclerView 可以轻松显示很长的列表（如群中提到的药物这一实体，其关系属性过多，在网页中会卡好几分钟才显示出来，这里极其流畅）。

2.8 疫情新闻聚类



同样是在线加载并计算，使用加载框将加载时的界面美观化。加载并计算完毕后，以各个类别的关键词作为 Tab 的标题，列表中显示每个类别下所有 event，按照时间降序排序。

算法选用：算法包括三个部分：特征向量计算，聚类算法，关键词提取。在特征向量计算上，我利用自带的分词 seg_text，使用归一化的 tfidf 作为每个事件的特征向量。在聚类算法上，我实现了 KMeans 和 DBSCAN 两个算法（见对应类，Clusterer 中注释掉的代码使用了 DBSCAN），但 DBSCAN 调参比较玄学，效果不好，因此弃用了。在聚类完成后，我统计每一类中出现次数最多的词作为这一类的关键词，若两类的关键词相同则将这两类合并。

亮点与难点：不同于一些人用 python 聚类后硬编码，我采用的是在线聚类，每次查看可能有十秒左右的加载时间，但可以接受。在线聚类的难度要大很多，需要用 Java 手动实现聚类算法，还要不卡 UI，而 python 甚至可以直接调用 sklearn 等库中现成的聚类算法，几

乎不需要写什么代码。

2.9 知疫学者

几乎是套用之前的框架，异步在线加载学者信息，之后显示为列表。点击列表中的学者，进入学者信息详情页，可查看各项信息。

亮点与难点：图片加载正确，不卡 UI；实现了学者简介、教育经历、工作经历中
和 \n 的处理，正确显示简介中的换行，并将经历分成若干条。

3 总结与心得

3.1 工作总结

由于系里熟悉的人不多，动身组队又比较迟，我一开始就决定了单人完成这个双人大作业。即便如此，因为个人原因，直到 9.5 我才开始动手写，经过 6 天的鏖战算是大功告成。整个项目的 java 代码大约 5300 行，xml 代码约 2500 行（不计入空行），是我大学以来这么短时间内写过的最多的代码了。开始写的前两天我整个人都是焦躁的，因为之前从没接触过 Android 编程，老师的最后两节课也没怎么听，简直一无所知。不过我还是参照各种博客上的示例和讲解硬啃了下来。前两天我几乎放弃一切娱乐活动，每天写到很晚，直到新闻列表功能成形，我才有了点豁然开朗的感觉，之后的若干功能虽然杂但并不难，大半都可以套用

写好的轮子。最后一天，就只剩下了缝缝补补和美化界面的工作。

本次作业中，我完成了除分类特效的 1 分外所有的工作，且使用了较好的框架，使得开发时代码耦合度低，逻辑清晰。单人完成的好处也在这里，从数据到界面，我的思路是一以贯通的，不需要复杂的沟通，但也丧失了一些锻炼合作的机会。除了要求的功能外，我也思索着向今日头条、知乎等主流 APP 的处理方式靠近，使得成品的操作合理流畅且交互性强。

3.2 收获与心得

Android 编程是我舒适区之外的一块技术，因为之前的编程课中接触到图形界面的几乎没有，只有大一小学期的 Qt 写了一些图形界面，但是完全不同的技术。Android 涉及到 UI 的创建、刷新、各种事件监听，就连切换到主线程进行刷新的方式我也是经过重重搜索才略知一二。在编写的过程中，我实践了 MVP 模式这样低耦合的框架，也初步了解了回调函数、多线程等在 UI 程序中的作用。

3.3 建议

希望之后的大作业要求能做到**优化接口**与**明确要求**。

此次的接口有一些不便之处，如没有提供新闻搜索功能的在线接口，但又要求我们实现在线搜索，那只能采取各种曲线救国的方式：无论是离线缓冲+在线增量构建整个新闻列表，还是在线读取部分新闻进行搜索，都有着自身的缺陷：前者无法做到无缝衔接，后者读取的新闻量无法预计，要么牺牲效果，要么牺牲性能。我认为大作业的目的是让我们熟悉多线程的运用和搜索如何实现，而并非这些细枝末节的东西。

在要求上，助教应该提前考虑各种实现的思路，或者自己尝试实现一下，给出更细化的标准。有些要求十分模糊，如“疫情事件聚类”，需要在线还是离线？是提取每个类别的关键词展示还是如何展示？有同学使用 python 调库直接就完成 tfidf 和聚类算法了，然后硬编码到程序里，有的同学则用 Java 手写了 tfidf 和聚类算法，并且在线进行，这两种做法在得分上会有差别吗？又比如知疫学者这些需要展示哪些东西，怎么样才合理美观，还是比较模糊的。