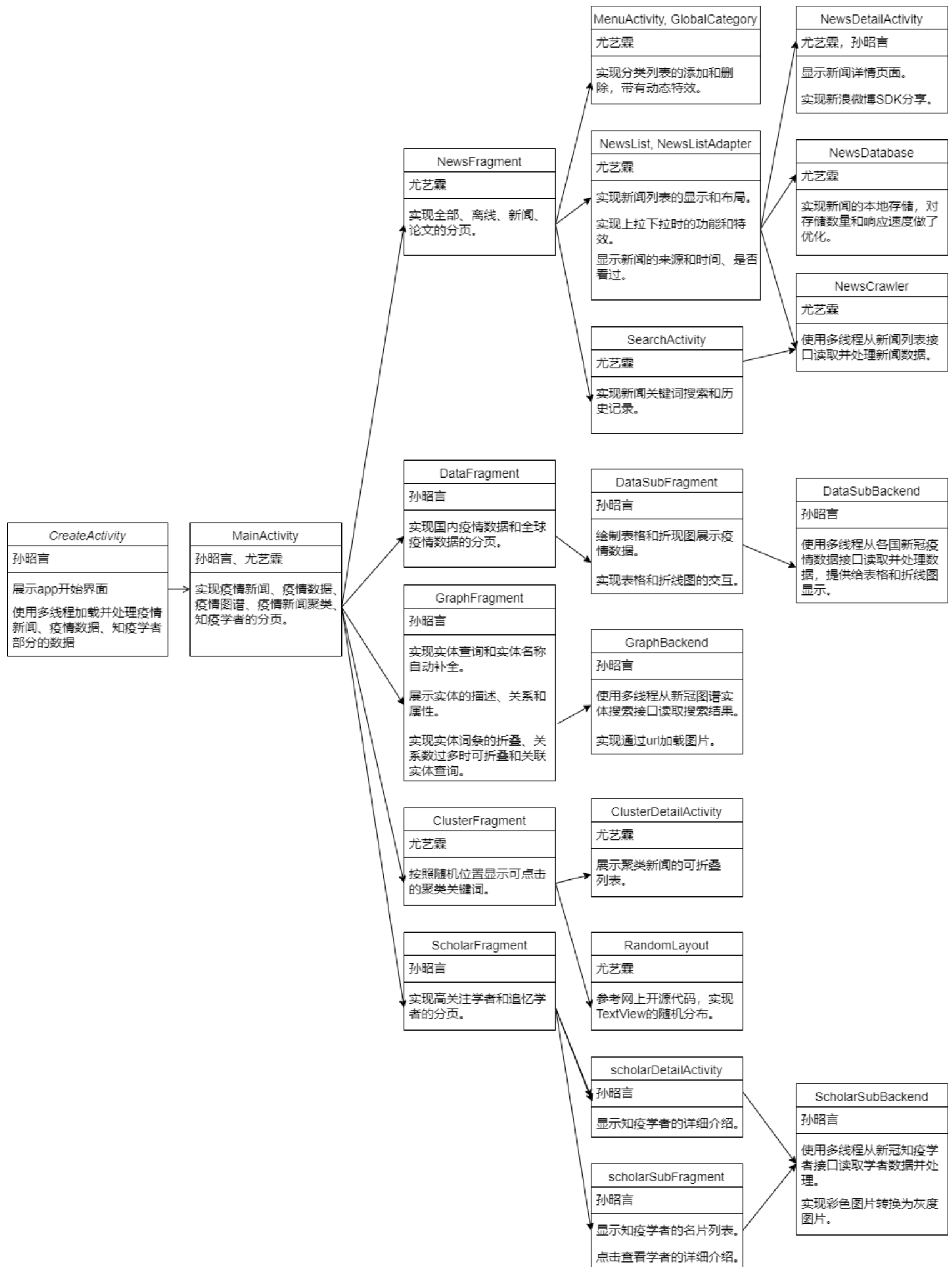


2020 年夏季 Java 小学期大作业

SysCOVID19

尤艺霖 2018011324 孙昭言 2018011308

1 代码结构



2 具体实现

2.1 项目分工

我们尽自己所能实现了所有的基础功能和附加功能。

| | 功能 | 子功能 | 百分比 (%) | 子功能 | 百分比(%) | 分工 |
|------|------|----------------------------------------------------------------------|------------------------------------------------------------------|-----------------------|--------------|-------------|
| 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 | 尤艺霖、孙 昭言 |
| | 页面布局 | 布局合理，点击 处理正确 | 8 | 美观，图片 布局合理 | 2 | 孙昭言、尤 艺霖 |
| | 分类列表 | 删除和添加操作 | 4 | 修改时有动 态特效 | 1 | 尤艺霖 |
| | 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 | 尤艺霖 |
| | | 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 | 尤艺霖 |
| | | 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 | 尤艺霖 |
| | | 显示新闻的来源 和 时间 | 4 | 布局合理美观 | 1 | 尤艺霖 |
| | | 新 闻 关 键 词 搜 索，历史记录 | 4 | 搜索页面合理 美观 | 1 | 尤艺霖、孙 昭言 |
| | | 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 |

| | | | | | | |
|------|--------|---------------------------------------------------------------------------------------------------------------------------------|---|----------------------|---|-----|
| | 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 | 孙昭言 |
| | 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（可参考 https://covid-19.aminer.cn/kg/ 对实体的查询和展示） | 8 | 实体词条展示美观，正确显示图片（如果有） | 2 | 孙昭言 |
| 附加功能 | 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 | 尤艺霖 |
| | 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（可参考 https://2019-ncov.aminer.cn/ ） | 4 | 页面合理美观 | 1 | 孙昭言 |

2.2 尤艺霖的工作

我主要完成的内容是新闻相关的功能（分享除外），以及附加分里的新闻聚类。

从网络获取新闻这一功能用了一个单例类 `NewsCrawler` 实现，其中的函数针对搜索、离线存储、普通页面有不同的实现。在获取新闻列表的时候，可以同时完成对于正文和 `Url` 的获取，其中正文直接用一个 `HashMap` 存下，以备后用。

在其他地方调用这个 `NewsCrawler` 的函数的时候，都会用 `Rxjava` 的 `subscribe` 到 `Consumer`，配合 `subscribeOn` 和 `observeOn` 来让网络操作跑在另一条线程上。

分类列表在 `NewsFragment` 中采用的是 `TabLayout` 和 `ViewPager` 进行展现的。

`ViewPager` 里装入一个新的 `Fragment`，在项目中名为 `NewsList`，同时这个 `NewsList` 也作为搜索结果的展示框。在 `NewsList` 中，为了支持上拉下拉，使用了 `SwipeRefreshLayout` 嵌套 `RecyclerView` 的方式来完成新闻列表的显示。点击其中的新闻即可进入新闻详情页。新闻详情页使用一个 `NestedScrollView` 完成长文章的滚动。

下拉刷新使用的是 `SwipeRefreshLayout` 自带的刷新功能。设置对应的 `onRefresh` 即可完成。

上拉加载更多使用了 `RecyclerView` 的 `onScrolled` 系列函数来完成。在滚动的时候，记录下可见的最后一个条目的位置，在滚动状态进入 `IDLE` 时，则查看最后一次看到的是不是最

后一个，如果是则进行更新。

上拉更新更多的逻辑就是朴素的获取下一页新闻，而下拉刷新新闻的逻辑则是，对于顶部的五个页面，做一个循环，每次下拉都会让最顶部页面在循环里加一，从而能在新闻来源没有更新的时候完成刷新，也能通过不断刷新来获取真正的最新新闻。

点击新闻对应的 `Fragment` 顶部的搜索栏，会打开一个新的搜索 `Activity`，这个搜索 `Activity` 包含一个 `SearchView` 和一个 `NewsList`。

在 `SearchView` 下面有显示搜索记录的部分，点击搜索记录可以直接将这一条提交到搜索栏。而负责搜索的这个 `NewsList` 逻辑和其他 `NewsList` 逻辑有些不一样，在 `Adaptor` 里做了单独讨论。

本地存储使用的是 `JSON` 的格式，存储搜索的历史记录、以及所有阅读过的新闻（含正文）。在 `App` 启动的时候加载本地文件，在 `App` 结束的时候写回文件并销毁。管理这一块的是单例 `NewsDatabase`。而这里的 `JSON` 只保留了需要的内容，而不是接口中所有内容，最大程度上提高了存储容量。同时对于查询，也使用了 `HashMap` 做了优化，达到高速响应的效果。

对于正文的存储，我在设计上实际上分成了两部分，一部分是看过正文的，这部分会永久保存，而没看过正文、只看过列表的，就只会存在临时的 `HashMap` 中。这样既可以确保打开新闻详情页的速度，也可以确保本地空间不被浪费。

按照第二次下发的接口的要求，我对 `[type=event]` 的数据进行了新闻聚类，使用的算法是从图论方法出发的。

观察到获取的 `JSON` 中，每一条 `event` 的下面都有相关的 `related event`，这些 `related event` 关系使得所有的 `event` 构成了一张无向图，聚类问题被转化为带权图划分问题。对于带权图划分这类问题目前基本都没有多项式解法，所以我这里选择了类似爬山法的做法。

首先要获取一个初始分类。先把所有的边按照权值从大到小排序，然后依次加入这个图，如果新加入的边连起来的连通块超过了一个阈值，就不要这条边，然后通过不断的反复遍历来添加一些漏网之鱼。

最后，要排除掉过小的分类，这些分类是由于 `related events` 不全才出现的，需要舍弃。

有了初始分类，就要对最终结果进行近似。这里选择的方法是不断通过随机顺序遍历所有的点，看看每个点更倾向于和哪个现有分类相连，然后将这个点归过去。不断持续遍历，直到结果最终收敛。

我还使用了一些自动计算出的指标来衡量聚类好不好，通过不同的参数选取最终有了现在正在使用的聚类结果。最终聚类的关键词，是由程序按照 `seg_text` 字段统计出现最高的词频，然后由我手动选择关键词。

具体的算法实现放在了 `src` 目录下的 `jsontaker.py` 中。对于新闻聚类的展示，我将本地完成的 `cluster.json` 直接放入 `assets` 文件夹，在展示新闻聚类时直接读取。

为了增加聚类展示的美观性，使用了开源的 `RandomLayout`，并对其做出了一些符合开源协议的修改，让显示聚类的按钮可以随机出现在界面的各个位置。对于聚类展示的背景，我使用了所有 `event` 使用 `related event` 关系连接而成的无向图，通过 `python` 的开源库完成绘制，然后设置一个透明度作为背景。

2.3 孙昭言的工作

我负责疫情数据、疫情图谱和知疫学者部分的实现，同时对其它部分的布局 and 美观做了一些优化。

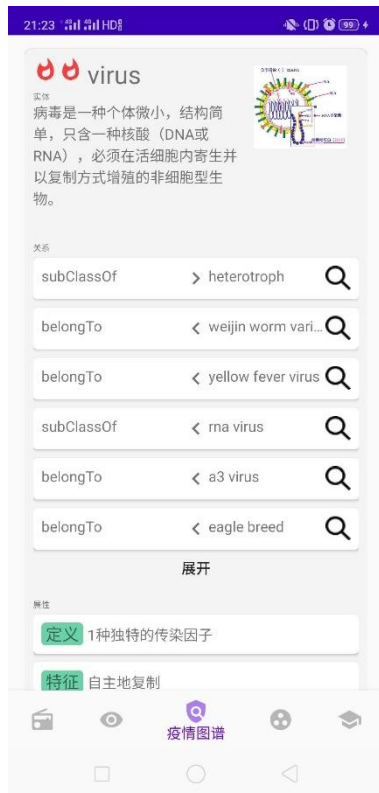
疫情数据部分的难点在于对 `MPAndroidChart` 和 `SmartTable` 开源框架的学习和使用。由于 `Android` 情境中对表格的使用较少，`SmartTable` 包的文档很不完善，很多功能需要我翻阅源代码后才能了解如何调用接口。我还通过 `checked` 成员记录了点击的表项，并在点击时展

示不同表项对应省份或国家的疫情趋势折线图。



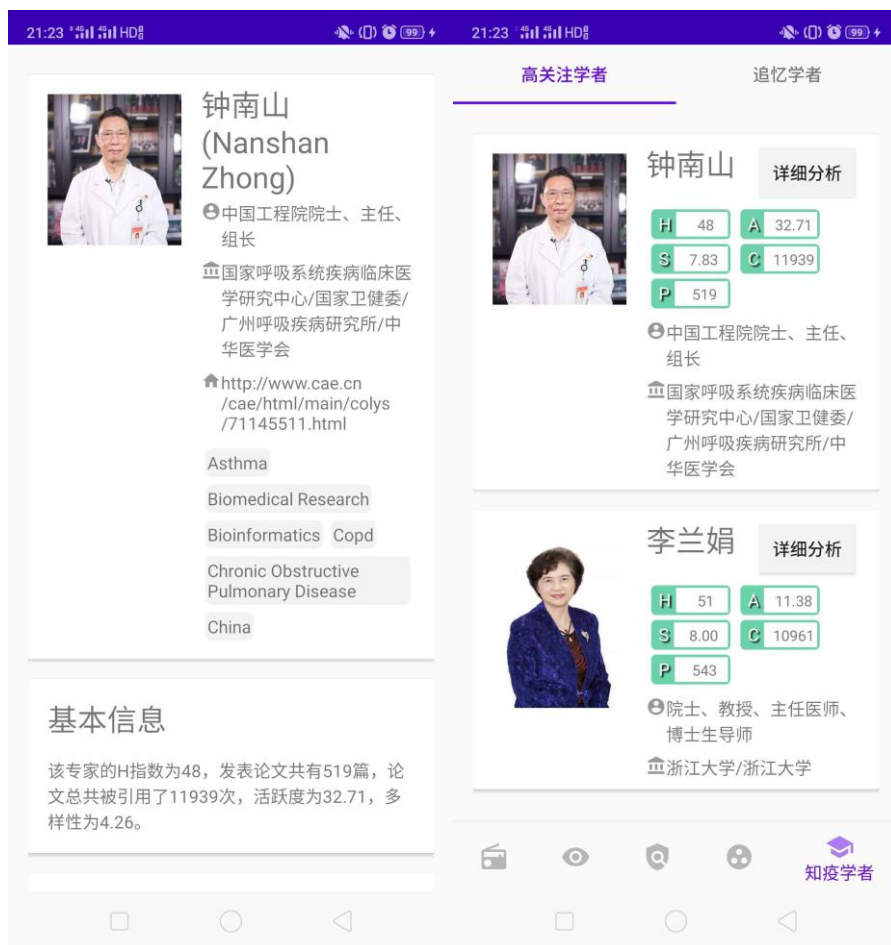
疫情数据表格和折线图

疫情图谱部分的难点在于通过嵌套的 `RecyclerView` 结构实现了外层实体词条和内层关系、属性的双重动态展示。这里的动态不仅体现在实体词条数和每个实体关系、属性数目的不确定，还体现在在点击时响应外层实体词条的展开和收回、内层关系部分的展开和上拉。我还通过在多处添加小图片使显示更加美观，不过这也给布局带来很大难度。我还从 `AMiner` 网站上读取了所有的实体名称保存在本地的 `graph_entity.xml`，从而实现了实体查询时的自动补全功能。



疫情图谱实体词条

知疫学者部分的难点在于对学者的诸多信息进行处理和展示。学者的部分信息可能缺失，在 `ScholarBackend` 中处理学者数据时要对键值是否存在做很多额外的处理。学者的照片数量较多，在加载学者列表页面时可能花费较长的时间。我将现有学者的照片保存在 `assets` 文件夹中，如果有新增学者再从新冠知疫学者接口提供的 `url` 加载图片。我通过 `RelativeLayout` 实现了学者详细信息页面中学者标签的动态排列。这里的动态体现在标签依次向右排列，如果标签的长度超出长度限制则自动换行。我还通过 `TextView` 手动调节做出了学者名片中比较精致的 `indices` 部分。



学者标签自适应排列

学者 indices 精致展示

详细介绍项目分工，每个人的工作，突出工作的难点和亮点。

3 总结与心得

3.1 尤艺霖的总结与心得

在这次大作业中，我完成了新闻和聚类两部分。在整个过程中，阅读了不少文档和开源代码，学习了一些实用技巧，也巩固了自身的代码能力。

对于这个大作业，我的建议是在第三周教学过程中，多给一些工程方面的教学，比如各种 View 的一些精妙运用。这些运用单靠查文档是不太可能了解的，但如果是在其他开源项目里看到，几乎立刻就能学会。

3.2 孙昭言的总结与心得

我的工作主要集中在对复杂数据的处理和对美观、布局的优化。在此过程中，我对前端页面设计有了更多的经验和了解，对多线程的使用有了更深的体会。自己动手设计出精致的页面确实能带来很高的成就感。

总结下自己的工作，谈一下过程中的收获以及对这个 project 的意见，方便我们以后改进。