

2019年夏季Java小学期大作业报告

欣闻 / XinNews

赵成钢 2017011362

1. 功能实现

	功能	子功能	百分比 (%)	是否实现	子性能	百分比 (%)
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	是	运行流畅不卡顿	2
	页面布局	布局合理，点击处理正确	8	是	美观，图片布局合理	2
	分类列表	删除和添加操作	4	是	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	是	正确展示图片和视频（如果有的话）	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	是	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	是	上拉下拉时添加特效	1
		显示新闻的来源和时间	4	是	布局合理美观	1

		新闻关键词搜索，历史记录	4	是	搜索页面合理美观	1
	分 享 收藏	使用微信、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、URL和图片	4	是	分享页面合理美观	1
		新闻详情页面点击收藏的添加和删除，实现收藏新闻的本地存储。收藏页的正确展示，点击可以进入新闻详情等	8	是	收藏页面合理美观	2
	新 闻 推荐	根据用户看过的新闻推荐相关的新闻，参考今日头条等	8	是	推荐新闻准确	2
附 加 功能 附 加 功能 附 加 功能	无					

2. 具体实现

首先对于整体的框架，我没有套用其他网络上非常成熟的框架而都是自己实现的，在/app/src/main/java/com/chenggang/xinnews下是主要的代码，分为几个模块最后再耦合，主要的模块有下面几个：

模块A 数据库模块

对应代码：database/NewsDao.java， database/NewsEntry.java， database/NewsRoomDatabase.java， DbBridge.java

这里我本来参考了Google官方的SQLite -> Dao(Data Access Object) -> RoomDatabase -> Repository -> ViewModel 的架构，然后用LiveData的方式来观测数据库的变化实现UI的刷新，最后决定LiveData对于作业的需求比较重，

就进行了精简，删去了Repository层和LiveData策略，所有的UI变化靠用户行为触发异步行为控制。

这里的NewsDao直接接入SQL语句进行查询；NewsEntry是数据库中的新闻词条设置，同时还实现了序列化，保证了不同Activity之间的传输；而NewsRoomDatabase是对Android Room的一层封装，用了Singleton的设计模式，用来创建数据库和取得数据库的DAO；最后的DbBridge是再根据需求对数据库的第二层封装，包括异步的插入和更新新闻，根据不同分类来获取新闻等等。

模块B 用户行为追踪

对应代码：BehaviorTracer.java

根据浏览历史提供推荐的来源，还会记录搜索历史记录等用户操作的行为，具体的做法是维护一个关于关键词和关键词得分的堆，用户一旦阅读一篇文章就把对应的关键词添加到堆中，保持高频关键词在堆顶，要获取推荐时，把堆顶的几个取出然后调用API进行搜索，同时随机一个最近阅读的关键词进行搜索从而达到推荐的效果。

对于关键词的存储，我使用了SharedPreferences存在文件中，全部操作均为异步保证UI线程的流畅性。

这个行为追踪也记录了用户的搜索记录和一些其他信息。

模块C 网络和文件系统接口

对应代码：Bridge.java

负责和提供的API进行通信，下载和搜索对应的新闻，也可以从网络上下载图片等资源并保存到本地生成URI，方便分享等功能，也支持从文件系统中读取对应的新闻。

所有的新闻都放在了Cache目录，如果App过大，系统也可以选择清理Cache，同时保证有原链接方便再次下载。

模块D 收藏、分享和已浏览

对应代码：CommonActions.java

提供收藏、分享和标记已浏览的功能，其中收藏和标记已浏览会调用DbBridge来刷新数据库，而分享会调用隐式的Intent分享图文。

特别注意的是，因为没有微信、微博等开发者的AppId，所以我这里只用了系统默认的股份，支持微信的文字分享，和微博的多图文分享。

模块E 新闻列表Adapter

对应代码：NewsListAdapter.java

提供对主UI的新闻列表的RecyclerView的适配器，支持对新闻的挂载和添加，包括翻页等功能。

同时对于有图片的新闻，Adapter会异步加载图片，也保证了UI线程的流畅

性。

模块F 图片缓存

对应代码：PicsCache.java

这里的图片缓冲并非放在目录中，而是在内存中的一个String到Bitmap的HashMap保证图片加载的流畅，当其他组件加载图片时，会先请求这个封装好的HashMap，如果在不在内存中会从文件系统中找，再不在就在网络下载，形成了这样一个框架来保证运行的速度。

模块G 常用工具

对应代码：Utility.java

一些常量和常用函数。

模块H MainActivity

对应代码：MainActivity.java

主界面的全部代码，包含了导航栏、新闻列表、搜索框的全部控制。

其中导航栏就用DrawerLayout实现，用SharedPreferences来保存分类的设置，点击后切换分类触发新闻列表的操作。

新闻列表用SmartRefreshLayout+RecyclerView实现，用上文中所说的NewsListAdapter实现数据的桥接。

搜索框则是直接调用了一个AlertDialog并设置成我所用的搜索布局。

值得一提的是我加载新闻的策略，当刷新列表时，首先直接把UI设置成空的Layout，同时异步从数据库加载，再同时从网络中加载，并在之前加一个锁，当数据库加载完之后并挂载到UI之后进行解锁，然后再把网络中加载出来的挂载到UI上，所有挂载操作都用一个在UI线程上的Handler实现，保证流畅性，在挂载的同时，把从网络上下下载的数据放到数据库中。

而当要加载更多新闻时，就直接通过当前的新闻数量来判断请求API的页数，同时再进行一些去重之类的操作，保证正确性。

模块I NewsPage

对应代码：NewsPage.java

新闻详情页的逻辑代码，控制新闻的加载和图片的异步加载等操作。

主要的困难和挑战

因为我在写这个作业时比较侧重于性能和流畅，所以大多数bug都是因为异步的线程操作导致，错误比较难复现也比较难调试，也算是一个挑战。另外一点就是一个人完成纯代码上的工作量比较大，连续写了约一周才写完。

另外就是加载新闻和图片的策略，本来我就单纯的以为这个作业只是写一个界面，但是其实觉得要更加流畅还是需要动一些脑筋的，我也尝试了几种不同的加载策略，原来都是用Android的LiveData来实现的，发现没有自己控制比较灵活，就改成了纯手写。而关于图片加载的策略之前也在想要不要文件系统

这一级，最后综合考虑才得到了最后的策略，在我实际的测试中效果也算比较好。

3. 总结与心得

考虑到本次的新闻App的功能耦合性强，也想进一步锻炼自己的代码能力，于是我选择了一个人一队完成本次的作业，最后也有很大的收获。除了老师上课的讲解，我学习完了Google官方的Android Developer Training的所有章节，在了解如何写的同时也在一定程度上对Android的复杂、精妙和严格的设计有了一定的认识，最后采用增量开发的方式把功能一层层实现也同时对代码不断Review和重构提高性能和可用性，最后在几乎所有模块都没有调用完整框架的情况下也达到了不错的效果。

还有一个很特别的感受是非常感慨Intellij IDEA这款IDE的强大，极大提高了我写代码的效率和准确性，包括Android一些严格的策略（比如不能在UI线程执行网络 and 文件IO等等），让我进一步地在代码能力上有个晋升，从只是写完能运行这个阶段到开始斟酌变量的名字、可读性以及代码的工整，让我更加享受这个过程。

对于这个大作业的建议：首先，两个人组队写不太合理，因为这个App的耦合性很强，基本不太好分开，我认识的同学很多是一个人写了另一个人接着写的，分工按照模块写的比较少，我觉得这样反而不能锻炼合作能力，因为合作的方式不太合理，也不一定会节约时间，如果再有专项活动这样的情况，我觉得应该调整分数而不是分工；第二点，我觉得分值设置不太好，完全就是谁写得更多谁就分高，而其实掌握了基础的开发技术之后剩下的很多工作都是Dirty work，我觉得这不是一个好事，不如让同学多花些时间了解Android的架构和设计，堆叠代码也没有太大的意义，不如让同学们思考如何设计一个更好的框架和策略，而不是更多的复用别人的框架，当然也只是我个人的想法，我的建议是把代码的质量而不是附加功能性加入评分标准，比如代码风格的一致性、复用性、有无Hardcode、框架和结构的合理性、性能、鲁棒性、单元测试等等等加入标准。

也感谢老师的讲解和助教的帮助，这门课上下下来对Java有了一定的认识也有很大收获，当然也希望系里可以进一步改进课程，优化同学们的体验。