

2017 年夏季 Java 小学期大作业

组长 唐适之 2015011308

组员 蔡子熙 2015011507

组员 方闽杰 2015011287

1 功能实现

	功能	子功能	百分比 (%)	是否实现
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。	5	是
	页面布局	布局合理，点击处理正确	10	是
	分类列表	删除和添加操作	10	是
	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。	10	是
		实现新闻的本地存储，看过的新闻列表在离线的环境下也可以浏览	10	是
		上拉获取更多新闻	5	是
		新闻是否看过的页面灰色标记	5	是
		新闻搜索	5	是
	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	10	是
		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。收藏也的正确展示，点击可以进入新闻详情等	10	是
加分功能	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	10	是
	语音播报	可以语音读出新闻等	20	是
	新闻人物地点链接	用户可以跳转到相应的百科词条等	5	是
	用户体验	新闻屏蔽功能，通过进一步询问用户想要屏蔽掉关于什么内容的新闻而实现基于关键词的屏蔽等	5	是
		夜间模式，用户可以调整背景色等	5	是

		文字模式和图片模式转换，文字模式不显示图片，帮助用户节省流量。	5	是
		流畅性强	视效果而定	视效果
		界面颜值高	视效果而定	视效果
		根据新闻文本补上相关的图片（因为部分新闻内容没有图片）	视效果而定	是
	代码和项目管理	使用了较好的框架	酌情加分	酌情
		有较完整的单元测试	5	是
		使用了 <code>github</code> 等好用的代码版本管理工具	2	是
		<code>maven</code> 或者 <code>gradle</code> 等项目管理工具	2	是
		其他提高编码效率的工具或操作	酌情加分	

2 小组分工

功能	实现人
网络 API 调用和返回数据的封装	唐适之
用户行为统计（含已读记录）及新闻推荐	方闽杰
分享到新浪微博、腾讯微博、QQ 空间和 QQ	蔡子熙
新闻及其中图片的长期存储和短期缓存	唐适之
语音播报	蔡子熙
为没有图片的新闻获取图片	蔡子熙
将新闻中人物地点转换为词条链接	方闽杰
多线程框架	唐适之
主页和分类浏览列表 UI	方闽杰和蔡子熙
收藏夹 UI	唐适之
分类管理（调整顺序和增删）UI	唐适之
新闻搜索 UI	蔡子熙
新闻详情 UI	方闽杰
顶栏和侧栏 UI	蔡子熙
夜间模式	唐适之和方闽杰
无图模式	方闽杰
关键词屏蔽	唐适之

3 具体实现

3.1 整体规划

整体规划时，我们尽量去除工程各组件之间的耦合性，使 model（功能）和 view（UI）分离。这样既可以使各组件得到充分地测试、使工程更健壮，也可以使各组件地开发不相互依赖，人员分工可以按需调整。

在进行 model 部分开发时，我们编写了广泛地单元测试，使开发 view 部分时不再需要担心大量上游缺陷。当各单元间的耦合不易解除时，我们使用了 Mockito 测试框架建立 mock 类，使个单元被独立测试。

我们使用 Github 管理工程进度，并使用了 pull request review 制度，既每一个 pull request 必须经作者以外的一人审阅过后，才能合并到主分支，保证了代码的高质量。

3.2 Model

Model 部分主要由以下五大类功能组成：API 调用、用户行为统计、存储、外部应用调用，和新闻文本解析。

API 调用方面，我们注意到此新闻 API 是通过 HTTP 返回 JSON 数据的形式提供服务的。但是在 Java 这种静态类型语言中直接使用 JSON 这样的动态数据类型时较为不变，时不时需要判断类型。并且，API 返回的某些内容，例如图片列表和日期，需要在此组件中进一步解析。于是我们进行了封装，将 API 返回的数据分为两类：新闻和新闻列表。我们的 API 组件不将 JSON 暴露给外界，而是返回封装好的信息。具体实现上，我们使用了 fastjson 第三方库来灵活地将 JSON 信息注入封装好地类中。为了更好地与 UI 中“上拉加载更多”的需求契合，我们还实现了一个“可延长的新闻列表”类。

用户行为统计方面，我们注意到新闻推荐、标记出读过的新闻等功能均需要统计阅读记录，故将用户行为统计单独作为一组件。用户推荐实现原理如下：我们将新闻中的关键词和分类定义为特征，定义全局历史向量 $h_i = \text{第}i\text{个特征被阅读的次数}$ 、单个新闻的特征向量 $f_i = \text{第}i\text{个特征的权重} \cdot I(\text{此新闻含有第}i\text{个特征})$ 。对于某未读过的新闻， $h \cdot f$ 即其分数。若想加载 n 条推荐，就先从 API 获取 $10n$ 条，再选择得分最多的 n 条返回。

存储方面，我们注意到需求主要有两个：存储被收藏的新闻及其图片到文件、缓存不久前被访问的新闻和文件到内存。我们使用 LinkedHashMap 实现了 LRU（Least Recent Used）内存缓存，每当需要访问资源（新闻或图片）时，先查找内存缓存，如果没有则查找文件，如果还是没有则从 API 下载。当需要添加资源到文件（收藏）时，先执行上述步骤，这样缓存中就有了该资源，然后将该资源存储到文件。由于 Android 程序在运行时可能会被杀死，我们使用写入再移动的方式实现了原子文件操作。

外部应用调用方面，分为第三方应用分享、语音播报和给无图新闻获取图片。我们使用 shareSDK 框架实现第三方应用分享、使用 MediaPlayer 实现语音播报。关于获取图片，我们对比了百度搜索标题、百度搜索关键词、必应搜索标题、必应搜索关键词四种方案，认为百度搜索标题是最好的。搜索到的图片的网络访问还不一定有效，我们还验证了其 http response header 确实是成功且返回的是图片。

新闻文本解析方面，我们利用了 API 中的人物地点提取，发出请求验证这些词确实存在于百度百科后，将这些词替换为指向百度百科的链接。

此外，为了确保应用的流畅性，许多操作需要异步完成。我们曾尝试使用第三方异步框架，但陷入了上游 bug 的泥淖。于是我们仿照 Javascript 的 Promise 实现了自己的 Promise 框架，即，将每一个异步事件实例化为一个 Promise 对象，接着通过此对象的 then 方法和 fail 方法分别指定它成功或失败结束后应该执行的下一个对象是什么，then 和 fail 还可以串接成链。考虑到 Android UI 线程的特殊性，我们还实现了 thenUI 和 failUI 方法，特指在 UI 线程上执行新任务。我们的 model 中所有的长时间操作均有返回 Promise 对象的异步方法。

考虑到某些类提供的是全局信息，本应将其设计为单例模式，但这些类可能要使用 Application Context，而 Application 是可能被销毁重建的，若使用单例模式则可能导致 context 引用失效。为了解决这个问题，我们实现了 Application 的子类 ApplicationWithStorage，并设置为启动 Application，用于存储这些需要 context 引用的全局信息类。

3.3 View

UI 方面，我们实现了以下页面(Activity)：首页（新闻推荐）、分类浏览、搜索结果、新闻详情、分类管理、收藏夹。此外，所有这些页面还要共享一个顶栏和侧栏菜单。

首先，我们高效复用了代码：为了实现统一的顶栏和侧栏，所有 Activity 均继承基类 BaseActivity，并于其中实现了包括顶栏、侧栏在内的所有公共部分。首页和分类浏览页都用到了下文将提到的 metro 风格的设计，我们定义了 NewsListActivity，使首页和分类浏览页都继承之。

其次，我们充分考虑了 UI 的流畅性，所有的耗时工作均异步完成。例如，将任务地点替换为百科链接时需要测试词条是否存在，耗时较长，我们就先显示无链接的文本，待链接准备好后再替换。同理，如果新闻中没有图片，从互联网上搜索类似图片也是耗时较长的操作，我们先显示无图片的新闻，待图片准备好后再加载。我们还定义了一个 LoadingImageView 用于加载图片，当图片正在加载时用 AVLoadingIndicator 显示加载动画，加载完毕后自动切换成要展示的图片。在朗读新闻时，由于下载整段新闻的语音数据耗时很长，我们使用了生产者-消费者模式，即一个线程负责下载语音并加入队列，另一个线程负责从队列中提取并朗读。

此外，我们注重 UI 的美观性。我们在首页和分类浏览页使用了 CardView 实现了 metro 风格的磁贴设计，共有四种形状的磁贴，分别是大图、横向三张小图、横向小图与文字，以及仅一张小图。这四种磁贴混合排列，避免界面单调。我们还为侧栏每一项功能的添加了图标，并统一了图标的颜色和样式。另外，我们还通过 Android 的 Theme 实现了夜间模式与普通模式的切换。

最后，我们还改善了 UI 的可交互性。Android 原生 RecyclerView 功能有限，我们使用第三方组件 SwipeRecyclerView 来扩展其功能。我们分别在收藏夹、分类管理页、新闻列表页（含主页、分类浏览及搜索结果）中，使用了滑动菜单、长按拖曳、刷新动画的效果。在没有网络而无法加载时（已存储内容例外），我们也提供了充分的反馈信息。

4 总结与心得

唐适之：

在规划这个项目时，我们选择的策略是自底向上地设计项目的各个模块，而不是在一个开源的新闻 app 框架上修改。这样虽然引入了很大的工作量，但是让我们对整个项目有了完整掌控。在开发后期，我们可以随时引入各种前端组件，而不会受到已有框架的制约。而在前期规划中较为不足的是：由于对 Android 的相关组件缺乏了解，没有合理地规划各个模块，导致各模块在开发中变化较大。但也正因为我们的各模块间独立性较好，并且遵循的是大家一起先做 model 部分再做 view 部分的策略，使分工可以灵活调整，大家掌握的知识也有很大交集，可以互相协助，而不是遇到问题时只能等负责的人解决。总而言之，这次的项目规划，在灵活性上做得很好，但规划前还应多做准备。

我们在 model 部分的开发中广泛编写了单元测试，这在代码逻辑需要调整时保证了可靠性。尤其是原计划使用的 Promise 库 `jdeferred` 有上游 bug 而必须换用自己实现的 Promise 时，单元测试避免了在每处用到 Promise 的地方均需要重新调试的麻烦。这种开发范式极大地保证了代码质量，唯一的缺点就是在初次编写代码时会消耗大量时间。以前我多次受困于难以维护的代码，这次的成功鼓励我以后在有条件的情况下继续使用这样的范式进行开发。

在团队协作方面，虽然任务量远比规划时预想的大，而且大家都有各自其他的任务，但各位都积极投入了大量的时间，也都贡献了高质量的代码。我对各位表示由衷感谢。

方闽杰：

在这次小学期的安卓组队大作业中，我们一起开发了一个新闻浏览的 app。在这个过程中，我学会了很多新的安卓编码知识，对安卓整体框架以及 java 的应用有了更深入的了解，也掌握了更多写 bug 和调 bug 的姿势。同时，在利用 github 进行合作的过程中，我对项目管理有了新的认知。在实现文本处理的过程中，我对 java 的字符串处理变得更加熟悉。在实现 Behavior 类的过程中，我学会了剖析用户的心理，在统计用户行为，计算偏好值的过程中，也用上了在人工智能课程中学到的知识。在实现新闻列表的视图时，我对之前接触过的 RecyclerView 有了更深刻的认知，也得以更加熟练地使用它，实现高效的列表视图。

我觉得在这次开发项目的过程中，实际上进程有些混乱，因为前端后端的接口并没有做到完全统一，事先也没有详细地商量好代码框架、结构，因此开发过程当中白白花了不少时间用来沟通这些东西上面。此外，由于时间限制，很多可以实现的功能并没有能够加入我们的 app。至于 UI，因为缺少时间去发现美观高效的框架，只能自己来美化，效果并不是很理想。希望以后的小学期能够为同学们提供更多的方向性的帮助，介绍一些好用的 android 开源库、UI 框架等等。

蔡子熙：

这次的大作业是难得的几个人组队完成一个比较完善的产品的经历。虽然历经艰辛，熬了不少夜，但是最终还是有满满的成就感。一个一个解决层出不穷的 bug，不断添加新的功能，添加完又产生新的 bug，再一个个解决；美工精细到每一 dp 的雕琢，每种颜色的选取，每个按钮和图标的制作都花费了不少精力。

我从这次的大作业中明白了在团队中沟通的重要性。多人协作完成一项任务，首先要有明确的分工，避免重复劳动；同时要及时地展示自己的 idea，不要不经过商量就顺着自己的 idea 一直做下去，做完才发现不理想或是和同伴做的有冲突，就白费功夫了。其次很重要的一点是，一定要规划好时间，对困难要有足够的估计。这次任务的完成明显前松后紧，到最后几天觉得快来不及才开始拼命熬夜，这样对身体不好是一方面，也无法很好地保证任务完成的质量。

感谢两位队友全程 carry 我，我做的都是一些零碎的活。这也说明了，抱大腿很重要（我的视角）。当你找不到大腿时，只能自己当大腿了（我队友的视角）。