

2017 年夏季 Java 小学期大作业

Group 26

组长 张卡尔 2015011025
组员 路橙 2015010137
组员 金展宇 2015011380

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	
		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	5	是

		图片，帮助用户节省流量。		
		流畅性强	视效果而定	是
		界面颜值高	视效果而定	是
		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	是
		有较完整的单元测试	5	
		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	
	收藏页面每个item的滑动删除(类似微信)	此功能主要考虑用户交互操作便捷性	希望获得2分的加分	

小组分工

张卡尔

- MVP架构的搭建，*Model*层接口的定制
- 本地数据库的搭建与相关操作
- 本地缓存机制的实现
- 通过新闻api获取并解析新闻列表、新闻详情、关键词、人物地点等信息
- 新闻搜索页面及其底层实现
- 新闻收藏页面、本地存储
- 新闻推荐算法设计及其底层实现
- 图片模式与无图模式切换
- 看过新闻的灰色标记及其底层实现

路橙

- MVP 架构的搭建，*View* 层和 *Presenter* 层接口的定制
- 分类列表的前端页面
- 新闻列表的*View* 层和 *Presenter* 层
- 新闻详情页面
- 上拉获取更多新闻
- 新闻搜索的前端页面
- 夜间模式
- 侧滑栏 *DrawerLayout* 与 *NavigationView* 的实现
- 界面设计，咨询美院视觉传达系的同学

金展宇

- 语音播报功能
- 多种社交平台的分享功能
- 根据新闻内容自动配图功能
- 对新闻中地点和人物链接功能
- 使用真机进行测试
- 创建密钥打包apk

具体实现

MVP 架构

使用了 *MVP* 架构，将程序分为 *Model*、*View* 和 *Presenter* 三部分。模型与视图分离，对于数据、网络的操作和对于 UI 的操作进行了有效的隔离，便于分工合作，也便于代码的维护和更新。

如下的做法保证了 *Model* 层不与 *View* 层进行交互：

- 当 *Presenter* 需要 *Model* 层的数据反馈时，*Model* 层不可以直接获取 *Presenter* 的引用，而是 *Presenter* 传递一个**回调函数类**，该回调函数规定了 *Presenter* 和 *Model* 特定逻辑的沟通方式，从而**保证 *Model* 层无法直接访问 *View* 层**。
- 当 *View* 层通过 *Presenter* 层访问数据或模型逻辑时，双方都不可以持有对方的引用，以避免 *View* 层通过 *Presenter* 的引用来直接访问 *Model* 层。因此，我们规定了一个**接口类 *Contract***，*View* 和 *Presenter* 都持有对方关于这个接口的**弱引用**，从而规定了 *View* 和 *Presenter* 的沟通方式，从而**保证 *View* 层无法直接访问 *Model* 层**。

Model

Model 层主要负责程序的底层逻辑和数据管理，并提供给 *Presenter* 层获取数据的接口。

Model 部分主要由以下几个类构成：

```
News.java
NewsDataSource.java
NewsPersistenceContract.java
NewsRepository.java
NewsLocalDataSource.java
NewsRemoteDataSource.java
NewsDbHelper.java
NewsFavoriteDbHelper.java
```

News 类是数据的基本单元，负责存储一篇新闻的全部信息并提供获取数据的接口。

NewsPersistenceContract 类定义了数据库中新闻

NewsDbHelper 类和 *NewsFavoriteDbHelper* 类继承了 *SQLiteOpenHelper* 类，负责创建、管理数据库。创建了两个数据库 *News.db* 和 *NewsFavorite.db*，分别是存储全部新闻和用户收藏新闻的数据库。

NewsDataSource 是数据源的接口类，定义了 *Model* 层需要提供给 *Presenter* 层的全部接口，包括获取数据接口、修改数据接口和回调函数接口。其中所有的获取数据接口均使用了回调函数的方式实现，原因是回调函数的方式能更好的对返回的数据进行处理。

`NewsLocalDataSource` 类是本地数据源，实现了 `NewsDataSource` 接口。其中所有的实现均通过读写数据库实现。

`NewsRemoteDataSource` 类是远程数据源，实现了 `NewsDataSource` 接口。其中所有的接口实现均通过创建新线程 `Thread` 并通过 `URLConnection` 连接助教提供的新闻api和其他网站获取html，并通过 `json` 和正则表达式 `regex` 解析网页内容获得数据。

`NewsRepository` 类是总数据源,负责与 `Presenter` 层直接交互，并决定从缓存或本地数据库还是从远端获得数据，并实时更新缓存和本地数据库。其获取数据的逻辑为：

- 若该数据存在于缓存中，则直接返回
- 若该数据存在本地数据库中，则更新缓存，并返回
- 否则从远端获取数据，并更新缓存和本地数据库，并返回

View

View 层负责显示数据和提供用户交互，通过监听用户行为，将用户行为传递给 *Presenter* 进行逻辑处理，最终由 *Presenter* 反馈更新UI界面。

View层只负责显示和更新界面，不处理数据读取或调用的逻辑。

以 `NestListActivity` 为例：

由于每个分类页面需要实时动态切换，因此采用 `Activity + Fragment + Viewpager` 的框架，每个分类页面为一个`Fragment`，每个 `Fragment` 在 `Activity` 中被初始化，并设置好 `Presenter` 和 `Listener`，当监听到用户操作时，调用 `Presenter` 的对应方法进行读取数据、更新界面。

`Fragment` 的使用保证了分类列表切换的流畅性。

Presenter

*Presenter*层负责处理中心逻辑，沟通 *View* 层与 *Model* 层。每个 *View* 持有 *Presenter* 的弱引用，*Model* 通过回调函数将数据结果返回给 *Presenter*。

以 `NewsListPresenter` 为例：

每个分类对应的 `Fragment` 持有一个 *Presenter* 的弱引用，*Presenter* 也持有该 `Fragment` 的弱引用，当需要更新界面时调用 *Presenter* 的特定函数完成对数据和网络的访问，并把新闻具体信息传递回 *View* 层。

功能实现

获取新闻列表及新闻详情

- 难点：

多线程访问网络，使用 handler 在线程之间通信。遵循 *主线程不访问网络，子线程不操作 UI* 的原则，在访问网络得到数据时新开子线程，用 handler 通知主线程的 RecyclerView Adapter 调用 `notifyDataSetChanged()` 来在主线程完成更新UI的操作。从而保证了加载新闻详情和图片时不卡UI界面。

- 亮点：

加载新闻标题先于加载新闻图片，并且在获取新闻图片时不会卡UI。通过 Glide 库异步加载和缓存图片，做到与 RecyclerView 完美结合。

调用 NewsRemoteDataSource 的 `getNewsList()` 方法和 `getNewsDetail()` 方法获得新闻列表和新闻详情。具体来说，在方法中通过创建新线程 Thread 并通过 HttpURLConnection 连接助教提供的新闻api和其他网站获取html，并通过 json 和正则表达式 regex 解析网页内容获得数据。

主要使用 RecyclerView 和 RecyclerView.Adapter<RecyclerView.ViewHolder> 显示新闻列表。



分类列表

- 难点：

1. 不同分类列表切换的流畅性。使用 TabLayout + Viewpager + Fragment 的结构，每个分类列表对应一个 Fragment，用 Viewpager 保证左右切换的流畅性。
2. 删除和添加分类。使用 SharedPreferences 保存用户保留的分类标签，每次在 NewsListActivity 的 `onCreate()` 中读取对应的数据，保证用户设置在退出程序后

依然保留。

- 亮点：

1. ViewPager 的左右滑动切换效果很流畅。
2. 标签删除操作设置为点击高亮或变暗，采用浮在界面上的标签页的风格，提升了界面的美观性和整洁性，也使用户操作方便简洁。





每种分类产生的 *Fragment* 只有分类不同，其他逻辑都完全相同。而标签删除的界面通过自己实现的 *Layout* 进行手动调整外观和位置。

新闻的本地存储

- **难点**：数据库操作，SQL语句
- **亮点**：实现了缓存功能，已缓存的情况下可减少大量数据库的操作

把从网络中获取的数据通过 *NewsLocalDataSource* 中的 *saveNewsList()*、*saveNews()*、*updateNewsDetail()*、*updateNewsPicture()* 方法把数据保存在数据库中。先通过 *SQLiteDatabase db = mDbHelper.getWritableDatabase()* 获取数据库对象，再调用 *db.rawQuery()* 执行SQL语句进行查询，获得*Cursor*，再用 *db.insert()*、*db.update()* 方法修改数据库。

上拉获取更多新闻

- **亮点**：
 1. 增加了 *Footer* 栏，上拉刷新时可以实时看到刷新界面。



2. 增加了顶部下拉刷新按钮，当界面卡死时可以在顶部下拉刷新使界面重新读取最新的新闻。

使用 `OnScrollListener` 监听下拉刷新操作。在 *Fragment* 里添加 *page* 属性，每次 `getNews` 时获取下一页的内容，从而既做到分页，又做到上拉刷新。

看过新闻的灰色标记

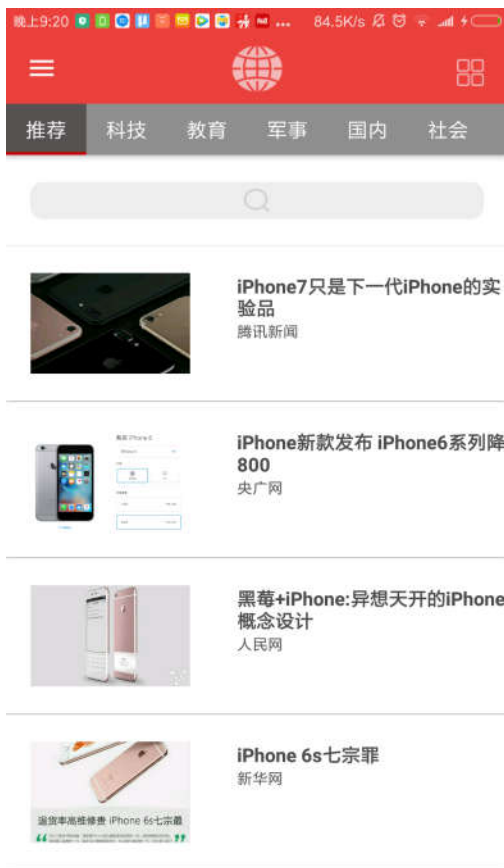
在 `RecyclerView` 中的 `onBindViewHolder` 中，判断新闻是否已读，若已读则改为灰色，否则为默认字体颜色。



新闻搜索

- **亮点：**在键入关键词的过程中会实时显示搜索结果。

通过新闻api获取关键词搜索结果。通过 `editView.addTextChangedListener()` 方法添加 `TextWatcher` 判断输入状态文本修改结束时获取搜索结果。

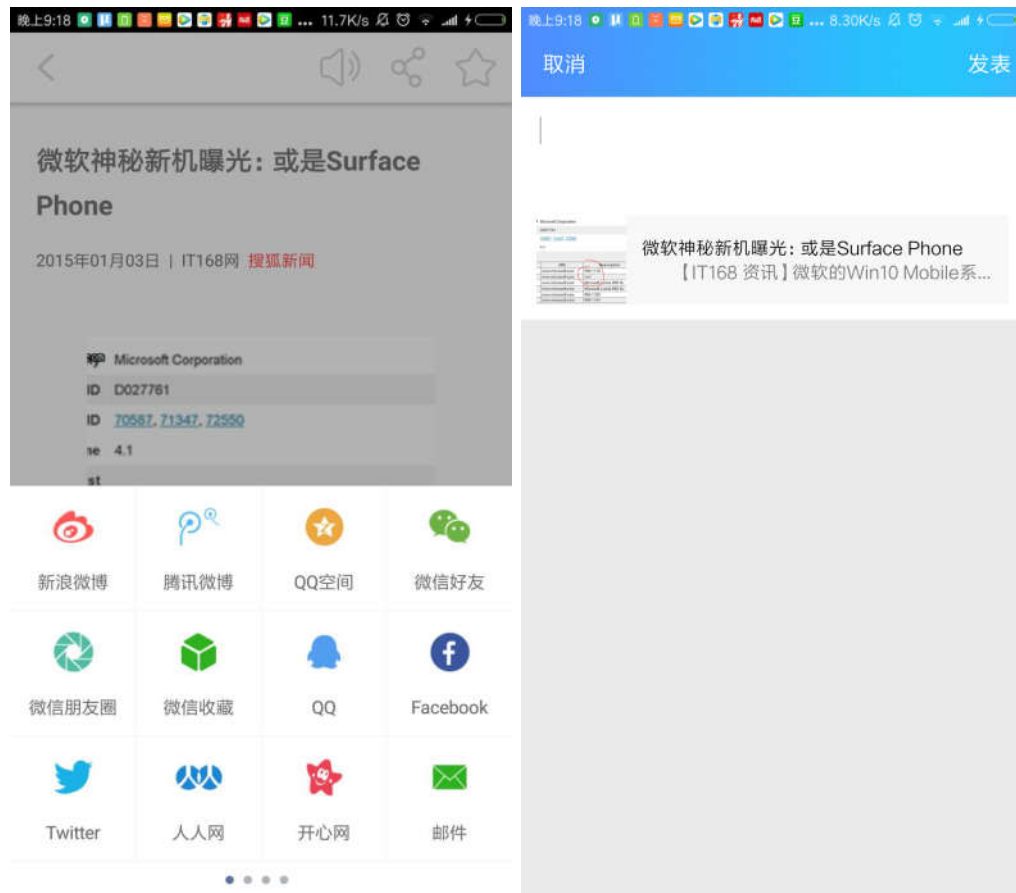


新闻分享

• 亮点：

1. 支持微信、朋友圈、豆瓣、Facebook、开心网、网易微博、人人网、新浪微博、搜狐微博、QQ空间、腾讯微博、Twitter和有道云笔记等多个平台的分享。
2. 朋友圈分享获得了微信公众平台的认证，可以分享配有标题的具体内容（类似推送）。

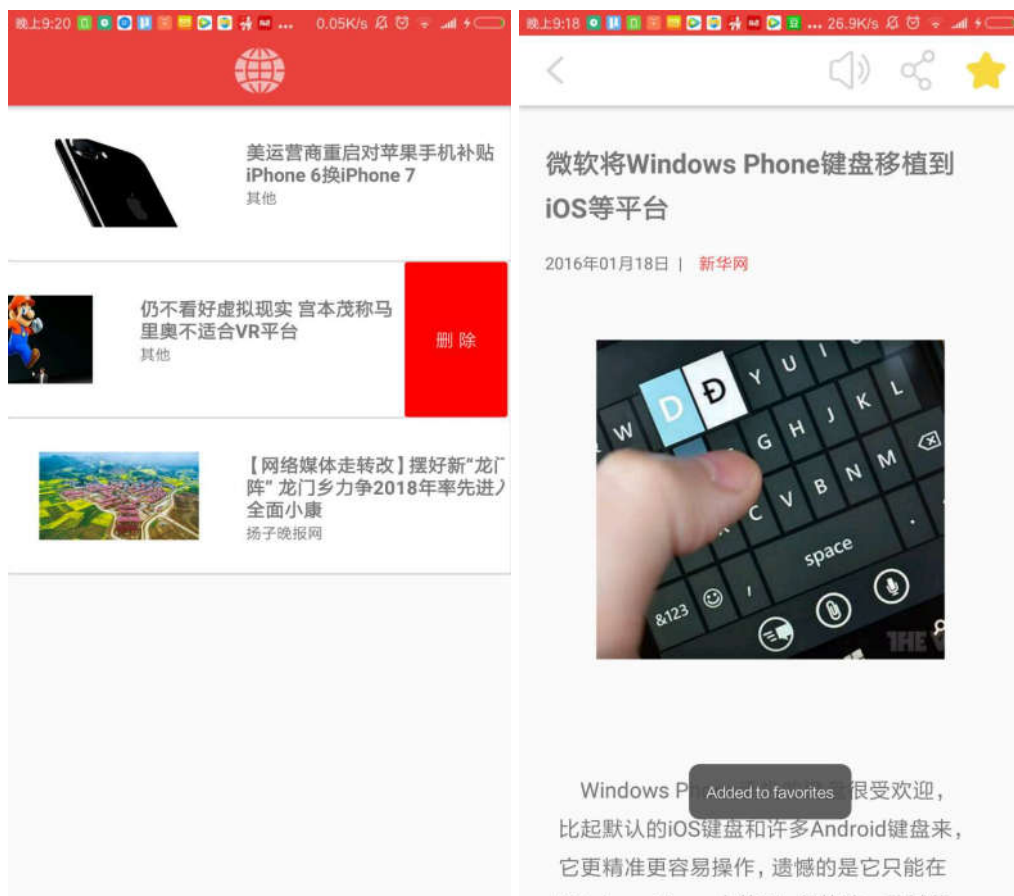
调用第三方库 *Share SDK* 进行分享，配置过程很难。



新闻收藏

- 亮点：

1. 收藏的新闻会按照收藏的时间顺序降序排序，增加了易用性
2. 左滑删除收藏操作，且滑动过程中会自动归正为显示删除按钮或不显示按钮，快捷方便
使用 `NewsFavorite.db` 数据库存储收藏新闻。自定义 `FavoriteItemView` 类，继承了 `HorizontalScrollView` 类，支持滑动删除和滑动归正。



根据用户看过的新闻推荐相关新闻

- **难点：**

1. 合理地提取用户浏览新闻的偏好
2. 将用户偏好储存在本地
3. 用合理的新闻推荐算法寻找用户可能感兴趣的新闻
4. 排除重复新闻

- **亮点：**

1. 可以在每次用户浏览新的新闻以后实时更新推荐新闻
2. 基于关键词权重的随机新闻推荐算法

使用HashMap记录用户看过的新闻的关键词中score最高的三个关键词并累计关键词的score，使用SharedPreferences保存在本地。推荐新闻时对HashMap中的关键词按照score排序，将score最高的五个关键词按照score的权重进行俄罗斯轮盘赌，随机使用新闻api搜索相关关键词的新闻，并记录，若已经推荐过则继续搜索。

语音播报

- **难点：**语音SDK的学习和使用

- **亮点：**

1. 朗读清晰，有节奏感
2. 一键阅读，再按停止



精锐尽出 中俄南海军演今“亮舰”

2016年09月12日 | 其他



大公网9月12日讯 (记者 [马静](#)) 代号为“海上联合-2016”的中俄联合军事演习今起于 [中国广东湛江](#) 以 [东海](#) 空域举行。中国海军昨公布了这次为期一周的军演科目内容。著名军事专家 [洪源](#) 在此间对本报表示，本次联演

调用第三方库 com.baidu.tts。

新闻人物地点链接

- 亮点：

1. 新闻中的人物地点红色高亮显示
2. 点击人物地点自动跳转到浏览器百度百科相关内容

使用正则表达式 `regex` 查找新闻中的人物和地点位置，使用 `SpannableString` 类为文字添加 `onClick()` 监听事件，使用 `Intent.ACTION_VIEW` 启动浏览器。



郭德纲

9月12日报道郭德纲现身北京，参加《坑王驾到》发布会，发布会上首次回应曹云金事件，虽然风波刚刚过去几天，但依然有不少人在关注着这段持续多年的师徒恩怨，郭德纲声称“可能你们觉得挺大事儿，对我来说真没往心里去。一篇作文就改变人生观？那我给你整本故事会你还不得飞起来。”并笑称：“我要是心窄活不到今天。”他还表示最近手上事儿忒多，想看热闹的再等等。

夜间模式

• 亮点：

1. 支持一键切换，不会退出Activity.
2. 将Switch通过ActionView集成到了侧滑栏的NavigationView中，方便用户操作，减少了界面个数。
3. 将用户设置保存在 SharedPreference 中，即使用户退出程序重新打开，依然保留上一次的设置。

调用以下函数初始化是否为夜间模式：

- `AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_YES)`
 - `AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_NO)`
- 调用以下函数来动态切换夜间模式与正常模式：
- `getDelegate().setLocalNightMode(AppCompatActivity.MODE_NIGHT_YES);`

- `getDelegate().setLocalNightMode(AppCompatDelegate.MODE_NIGHT_NO);`



文字模式和图片模式转换

- **亮点：**将用户设置保存在 SharedPreference 中，即使用户退出程序重新打开，依然保留上一次的设置。

使用SharedPreferences将设置存储在本地，在加载图片之前判断模式，若为有图模式，则加载图片，否则不加载。



根据新闻文本补上相关图片

- **难点：**
 1. 加载图片时不卡UI
 2. 需新开线程搜索相关图片
- **亮点：**相关新闻和原新闻内容的相关度极高

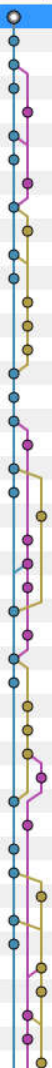




















以新闻标题为关键词进行百度图片搜索，并解析得到的html，得到相关图片。



界面颜值高

- 亮点：

通过求助美院视觉传达系五字班同学，专业化设计了app的页面结构与配色。整体界面严格按照尺寸设置，采用扁平化的设计风格。来自该同学的部分尺寸标注如下：

Graph	Description	Date	Author	
	Uncommitted changes	14 9月 2017 2:13	*	*
	  master  origin/master  origin/HEAD Update README.md	13 9月 2017 15:05	LuChengTHU <c-lu15@mails.tsinghua.edu.cn>	b2afe77
	final, picture and logo wati to be updated	13 9月 2017 14:58	LuChengTHU <lucheng.lc15@gmail.com>	b78ca44
	switch, bugs fixed	13 9月 2017 13:48	LuChengTHU <lucheng.lc15@gmail.com>	da3ecce
	 origin/picture_mode  picture_mode fix picture bug	13 9月 2017 13:27	Karl <zke_1996@126.com>	d1525a4
	Merge branch 'picture_mode' into lucheng-develop	13 9月 2017 11:44	LuChengTHU <lucheng.lc15@gmail.com>	057280d
	yejian	13 9月 2017 11:43	LuChengTHU <lucheng.lc15@gmail.com>	1c3a716
	add picture with cache	13 9月 2017 10:29	Karl <zke_1996@126.com>	24f272b
	 origin/lucheng-develop  lucheng-develop fix bugs with merge	12 9月 2017 21:46	Karl <zke_1996@126.com>	9875dc1
	 origin/recommendation  recommendation add recommend user setting	12 9月 2017 19:58	Karl <zke_1996@126.com>	df66e1f
	merge, with bugs	12 9月 2017 19:48	LuChengTHU <lucheng.lc15@gmail.com>	7dd2248
	ui modify, day night	12 9月 2017 19:40	LuChengTHU <lucheng.lc15@gmail.com>	20e8ec7
	add recommendat on	12 9月 2017 18:41	Karl <zke_1996@126.com>	242d28d
	 origin/develop  develop add people and location link	11 9月 2017 22:38	jinzy15 <851162585@qq.com>	0fba2c2
	parse people and location from json	11 9月 2017 22:17	Karl <zke_1996@126.com>	64d2702
	 origin/animation  animation modify text content	11 9月 2017 1:37	LuChengTHU <lucheng.lc15@gmail.com>	c89ad9e
	fix favorite height bug	11 9月 2017 0:51	Karl <zke_1996@126.com>	77b5a04
	add voice	11 9月 2017 0:09	JinZhanYu <kim@JinZhanYudeMacBook-Pro.local>	a56e140
	can speak	10 9月 2017 23:43	JinZhanYu <kim@JinZhanYudeMacBook-Pro.local>	4520167
	merge, with favorite 1 page 1 item bug	10 9月 2017 22:49	LuChengTHU <lucheng.lc15@gmail.com>	492123f
	add draerlayout, searchbutton	10 9月 2017 22:45	LuChengTHU <lucheng.lc15@gmail.com>	c0b9536
	 origin/favorite  favorite add favorite slite delete	10 9月 2017 22:38	Karl <zke_1996@126.com>	80a631e
	 origin/zhanYu-develop  zhanYu-develop merge voice	9 9月 2017 18:46	JinZhanYu <kim@JinZhanYudeMacBook-Pro.local>	7e56ccf
	Merge branch 'develop' into zhanYu-develop	9 9月 2017 18:38	JinZhanYu <kim@JinZhanYudeMacBook-Pro.local>	74ad234
	can read, main activity	9 9月 2017 18:36	JinZhanYu <kim@JinZhanYudeMacBook-Pro.local>	e6e6ce2
	fix cachelist bug	9 9月 2017 12:35	Karl <zke_1996@126.com>	37f4f20
	I think i have done my work	9 9月 2017 11:53	JinZhanYu <kim@JinZhanYudeMacBook-Pro.local>	39c7d45
	 origin/data  data merge, bugs with current page	9 9月 2017 2:36	Karl <zke_1996@126.com>	4894c37
	add favorite, fix search	9 9月 2017 2:26	Karl <zke_1996@126.com>	1ed92eb
	category ok, with close icon and scroll bugs	9 9月 2017 2:22	LuChengTHU <lucheng.lc15@gmail.com>	7bf6037
	category and voice icon	8 9月 2017 22:24	LuChengTHU <lucheng.lc15@gmail.com>	d12f166
	ok to merge with category icon	8 9月 2017 22:12	LuChengTHU <lucheng.lc15@gmail.com>	56c06e4
	ready to merge, add category icon	8 9月 2017 22:09	LuChengTHU <lucheng.lc15@gmail.com>	afa18c7
	add category	8 9月 2017 21:42	Karl <zke_1996@126.com>	2215551
	image on the top, then tab	8 9月 2017 20:54	LuChengTHU <lucheng.lc15@gmail.com>	54838b6
	clean directories	8 9月 2017 16:43	Karl <zke_1996@126.com>	c27418d
	Merge branch 'data' into develop	8 9月 2017 16:04	Karl <zke_1996@126.com>	6b79e25
	add grey textview	8 9月 2017 16:03	Karl <zke_1996@126.com>	a9ebe2a
	Merge branch 'develop' of github.com:Karl8/CaiVaNews into develop share sdk	8 9月 2017 15:43	JinZhanYu <kim@JinZhanYudeMacBook-Pro.local>	67770e5
	share bugs with wechat string, other fixed	8 9月 2017 15:43	JinZhanYu <kim@JinZhanYudeMacBook-Pro.local>	83c4eb5
	update todo list	8 9月 2017 14:11	Karl <zke_1996@126.com>	ce541ba
	update todo list	8 9月 2017 14:10	Karl <zke_1996@126.com>	5d53453
	Merge branch 'develop' of github.com:Karl8/CaiVaNews into develop	8 9月 2017 13:41	LuChengTHU <lucheng.lc15@gmail.com>	040a1d7
	fixed news_page showing bugs	8 9月 2017 13:37	LuChengTHU <lucheng.lc15@gmail.com>	f036936
	fix favorite bugs	8 9月 2017 13:15	Karl <zke_1996@126.com>	06418eb

总结与心得

工作总结

- 本次作业中，我们小组完成了几乎所有的给定任务，并在专业设计师的指导下完成了UI设计。
- 本次开发首次尝试使用成熟开发架构，使得代码逻辑清晰，耦合度低且易于分工。
- 最终产品界面简洁大方美观，流畅性高，可用性强，用户交互的友好度高。

任务收获

- 本次大作业首次尝试小组合作的方式进行开发，熟练使用了Github这一强大工具。
- 安卓开发前期入门较难，尤其在贵系没有任何前端课程基础的情况下直接开始写前端，这项任务是一个挑战。但在写出第一个能跑的版本后，逐渐熟悉了安卓开发的流程和术语，之后很多的工作都在调试UI界面，体验到了任务由难到易的过程。
- 我们较早地选择了MVP架构，因此在编写程序时思路更清晰，代码耦合度较低，在代码整合时减少了很多不必要的麻烦。

建议

- 建议以后的java大作业的实用性更强一点（例如让新闻app中的新闻能够与时俱进，或者其他实用性app），让同学们充分体会到java编程的趣味和做出成品后的成就感。
- 建议以后开发的app可以让同学们做完之后可以直接用。本次新闻app由于API的问题，无法做到自身使用。