开发框架：APP采用基于 Kotlin + Jetpack开发组件 + MVVM架构+ 协程+ RxHTTP+ Glide的开发框架。

主要技术有：

1. 项目架构：使用MVVM架构模式搭建,封装BaseVmActivity,BaseVmFragment,BaseApp等
2. UI基本遵循Google Material Design设计风格。
3. 网络请求：使用Rxhttp+协程+ okhttp 进行网络请求。封装Rxhttp为HttpManager，统一管理app的http网络请求。
4. 数据库操作：使用jetpack room进行数据库的操作。
5. kotlin协程：使用协程异步进行网络请求、缓存、线程调度。
6. 图片加载：Glide加载监听，获取缓存，圆角图片，高斯模糊。
7. 组件化开发，横向解耦。
8. 封装各种工具类，比如压缩图片、轮播器、查看大图、缓存工具、图片选择。
9. 各种封装好的依赖库，比如recyclerview万能适配器、经常使用到的几种Dialog，QMUI等。
10. 侧滑菜单：DrawerLayout+NavigationView。
11. 使用leakcanary引入检测内存泄漏。
12. 使用FlowLayout实现Android流式布局，支持点击、单选、多选，实现热门标签效果。
13. 使用ViewPagerIndicator实现顶部tab，主页底部tab，引导页，无限轮播banner等。
14. 使用Android Studio ViewBind进行自动视图绑定。
15. 使用GsonFormat & Gson json 进行数据解析。
16. 使用Logger打印日志。
17. 使用LiveEventBus进行事件发布/订阅框架，组件之间通讯。
18. 使用takephoto拍照上传。
19. 使用jetpack中的lifecycle组件进行view声明周期的管理。
20. RxPermissions ：基于RxJava开发的用于帮助在Android 6.0中处理运行时权限检测。
21. BaseRecyclerViewAdapterHelper ：RecyclerView多功能适配器。
22. QMUI：由腾讯出品的一组开源UI库,可方便快速高效实现下拉加载,上拉刷新,圆形头像,状态栏修改等。
23. LFilePicker：检索手机目录方式的文件选择器。
24. ViewPager 搭配 Fragment 实现懒加载。
25. 使用 ItemTouchHelper 实现应用频道排序、移动等管理。
26. 使用jetpack navigation进行fragment之间的跳转管理
27. 使用MMKV代替sharedpreferences进行缓存