MainActivity：

界面布局：其主界面布局方式为一个典型DrawerLayout布局，主体界面为Toolbar和一个FrameLayout为基础的Fragment Container。Toolbar内包括了登陆后的用户头像、名字和个人签名。侧滑栏内内容与官方例子不太一样，其没有使用NavigationView，而是自己定义了一个线性布局，里面依次排布了header、RecyclerView和两个ImageView+TextView的一般选择项。

逻辑实现：继承了基Activity类，在onCreate中先通过setTheme()函数（来自AppCompatActivity）设置了设置文件中保存的Theme，再进行了ButterKnife关联，设置碎片容器等操作。而各种初始化操作则是在onPostCreate()中完成的，其中包括：创建一个保存用户常数的懒单例，创建对话框对象，创建两个不同内容的碎片，加载原NavigationView栏应有的项目，设置Networks的TOKEN（initData()中）；初始化Toolbar和上面创建的Dialog，检测是否已为登录状态，如果是的话则获取登陆信息(以Entity封装信息，并且将其作为InfoEntity类的成员变量，间接调用)并更新对应UI(包括设置登陆情况下一些UI的可见性和添加Navigation列表)，最后将前面创建好的碎片设置为当前的碎片(设置方式为通过isAdded()判断用hide() show()还是add()将其添加进当前界面)(initView()中)。

开源库记录：

com.facebook.drawee.view.SimpleDraweeView 图片控件，可以通过同一库内的ImageRequest从网络或者本地获取图片(实际上是预先注册的Fresco承担了加载任务)，能设置属性使其显示为圆形图片，设置逐渐加载以及各种加载状态显示。

ButterKnife 注释开源库，可以用于简化View初始化。

首页Fragment：

同上，在onPostCreate()中初始化创建并初始化Adapter(initData()中)，将此Adapter赋予RecyclerView(initView()中),设置各种控件监听(initListener()中)，通过Presenter-Model-Networks API的途径获取数据(initLoad()中)//①。

阶段总结：总的来说MVP框架的实现流程是这样的——建立View、Presenter和Model的接口，根据功能定义各自需要的函数（包括View加载流程的监听回调，Presenter的VM关联）；定义该框架下View的抽象类（Activity、Fragment抽象类），在其中将MVP共同的模块绑定过程写在声明周期函数中，便于该框架下其他类的继承使用；建立具体View类及其对应的Presenter和Model，但这些模块需要根据当前功能实现进行二次封装，添加必要的抽象函数，而不是直接继承Presenter或Model基类。

1. :其具体过程为

——mPresenter.getTopicListByForum(TYPE, mPageIndex)->构建一个新的Observable对象加入CompositeSubscription中->其中正常构造所求对象（话题列表）的过程为mModel.getTopicByForum(type, pageIndex)->而这个函数调用了API方法Networks.getInstance().getTopicApi()  
 .getTopics(getOptionsByForum(type, pageIndex))  
 .compose(RxSchedulers.<TopicListEntity>io\_main())

->Observable对象构造的最后定义了消息接收方.subscribe(getTopicListObserver(pageIndex)))

->而此方法构造了一个Observer对象，调用了各种情况下View的预设回调接口，如refreshTopicList(TopicListEntity topicListEntity)、onRequestError(String msg)等。至此一个完整的网络数据请求过程结束，但其中RxAndroid和Retrofit的调用过程仍需探索。

开源库记录：

org.ocpsoft.prettytime 时间格式化开源库，可以用于计算时间差和输出符合当地语言风格的时间表达。

cn.bingoogolapple.badgeview.BGABadgeRelativeLayout消息计数标签效果自定义RelativeLayout控件，可通过其自定义属性设置标记的大小颜色、字体颜色和是否可拖动。

com.kennyc.view.MultiStateView 拥有四个状态的FrameLayout子布局，可以通过其状态设定不同的布局（其实现原理貌似只是将四种状态的布局都预先加载好，然后根据当前状态进行隐藏和展示。）。

com.melnykov.fab.FloatingActionButton 可以通过与RecyclerView类的控件关联隐藏按钮。

MarkDown ，轻量级文本编辑语法，可以通过简单的标记实现css装饰效果。

android.support.percent.PercentRelativeLayout 官方定义RelativeLayout的子类，可以通过百分比的方式定义边距和长宽。

in.srain.cube:ultra-ptr 下拉刷新上拉加载开源库。

个人信息页Fragment：

通过上一个Activity传来的用户ID信息获取相关的数据填入UI。其布局为典型CoordinatorLayout，滚动折叠AppBarLayout至顶部，并且设置滚动监听，为信息界面设置透明度变化动画。设置网址相关TextView点击事件为打开相应网址的WebView，进入个人信息编辑界面的方式为startActivityForResult，编辑信息通过一个带编辑框的Dialog，编辑完成后提交，并且在编辑界面关闭时将修改信息以序列化方式，返回一个已修改的信息实体。

开源库记录：

com.bumptech.glide 通过一个链式结构对图片进行加载、处理和设置给ImageView，实现高斯模糊等效果。

扫描登陆Activity：

通过将二维码扫描界面设置为主界面，获取用户的名称与标记信息，然后向服务器发出请求取得UserInfoEntity。

开源库记录：

me.dm7.barcodescanner.zxing 基于com.google.zxing开源库的项目，用于扫描并解析二维码。