整个APP：AndroidMainfest.xml是控制整个APP的核心。

这个文件描述了APP所需要的权限,整个界面的主题，APP中拥有的Activity(安卓中的activity即是主要场景)以及，各个Activity的属性。属性中包括场景启动时的启动类，场景可以采取的动作等属性。

Android中场景的启动或切换是通过发送Intent来实现的。Intent中存有希望用哪个类启动或者想启动的Activity可以进行的动作。

而AndroidMainfest.xml中Activity的属性指定了该Activity的启动类或者该Activity的动作。因此实际上通过寻找AndroidMainfest.xml中匹配的Activity，然后进行启动。

区别在于：指定启动类的话，只能是当前应用内。而没有指定启动类，指定想要进行的动作的话，安卓系统会在所有APP中进行寻找可以进行响应的Activity。

各个Activity的动作属性是在该APP不启动的情况下,单个界面可以被其他应用呼出的原因所在.

代码方面：整个应用 （逻辑代码） 与 （界面设计和资源属性等）分离。 逻辑代码都是Java代码，相互之间可以call，new。 界面和资源方面，在资源中写代码时，可以通过“@restype/resname”的形式相互调用。谷歌推荐将界面设计与界面中的数值（包括文本，颜色RGB值，尺寸值等）分开。那么可以做到界面与数据（不是狭义上的数据）分离，便于控制和保持一致。这样可以让界面设计的代码更加清晰明了，且增加了复用性。（道理有点类似于写html的时候，标签的style一般不会直接写，而都是抽出来拿到专门的css文件中）（可以看到zxing为了国际化，放了多个文本数值文件，安卓系统会自动根据系统设置的语言采用相应的文本数值文件，）。

Java代码是如何操作界面设计中的控件和使用各种资源？在生成apk的时候，会生成一个Java类，叫做R，会将各种资源，控件的id等转换成静态的int变量进行保存。在Java代码中可以通过类似于”R.id.controlName”进行获取控件，然后对控件进行操作。对于资源，也是类似，通过类似于“R.restype.resname”获取或者加载资源，然后使用。

（这里的资源包括直接图片，代码描述的图片，文本，颜色，数值，界面设计等）

