Material Design 规范相关（简略参数版本）：

颜色不宜过多。选取一种主色、一种辅助色（非必需），在此基础上进行明度、饱和度变化，构成配色方案。

Appbar背景使用主色，状态栏背景使用深一级的主色或20%透明度的纯黑。

小面积需要高亮显示的地方使用辅助色。

其余颜色通过纯黑#000000与纯白#ffffff的透明度变化来展现（包括图标和分隔线），而且透明度限定了几个值。黑色：[87% 普通文字] [54% 减淡文字] [26% 禁用状态/提示文字] [12% 分隔线]  
白色：[100% 普通文字] [70% 减淡文字] [30% 禁用状态/提示文字] [12% 分隔线]

桌面图标尺寸是48dp X 48dp优先使用material design默认图标。设计小图标时，使用最简练的图形来表达，图形不要带空间感。小图标尺寸是24dp X 24dp。图形限制在中央20dp X 20dp区域内。小图标的颜色使用纯黑与纯白，通过透明度调整：

黑色：[54% 正常状态] [26% 禁用状态]  
白色：[100% 正常状态] [30% 禁用状态]

图片上的文字，需要淡淡的遮罩确保其可读性。深色的遮罩透明度在20%-40%之间，浅色的遮罩透明度在40%-60%之间。

英文字体使用Roboto，中文字体使用Noto。

* 12sp 小字提示
* 14sp（桌面端13sp） 正文/按钮文字
* 16sp（桌面端15sp） 小标题
* 20sp Appbar文字
* 24sp 大标题
* 34sp/45sp/56sp/112sp 超大号文字

长篇幅正文，每行建议60字符（英文）左右。短文本，建议每行30字符（英文）左右。

所有可操作元素最小点击区域尺寸：48dp X 48dp。

栅格系统的最小单位是8dp，一切距离、尺寸都应该是8dp的整数倍。以下是一些常见的尺寸与距离：

* 顶部状态栏高度：24dp
* Appbar最小高度：56dp
* 底部导航栏高度：48dp
* 悬浮按钮尺寸：56x56dp/40x40dp
* 用户头像尺寸：64x64dp/40x40dp
* 小图标点击区域：48x48dp
* 侧边抽屉到屏幕右边的距离：56dp
* 卡片间距：8dp
* 分隔线上下留白：8dp
* 大多元素的留白距离：16dp
* 屏幕左右对齐基线：16dp
* 文字左侧对齐基线：72dp

最重要且随处用到的操作，建议使用悬浮按钮。信息较多时，选用凸起按钮可以有效突出重要操作，但注意纸片不要叠太多层。扁平按钮适合用在简单的界面，例如对话框中。

使用悬浮按钮要遵循以下规则：

* 建议只用一个悬浮按钮
* 悬浮按钮可以贴在纸片边缘或者接缝处，但不要贴在对话框、侧边抽屉和菜单的边缘
* 悬浮按钮不能被其他元素盖住，也不能挡住其他按钮
* 列表滚动至底部时，悬浮按钮应该隐藏，防止它挡住列表项
* 悬浮按钮的位置不能随意摆放，可以贴着左右两边的对齐基线

悬浮按钮通常触发正向的操作，添加、创建、收藏之类。不能触发负面、破坏性或不重要的操作，也不应该有数字角标。悬浮按钮有两种尺寸：56x56dp/40x40dp

卡片统一带有2dp的圆角。卡片最多有两块操作区域。辅助操作区至多包含两个操作项，更多操作需要使用下拉菜单。其余部分都是主操作区。

对话框包含标题、内容和操作项。点击对话框外的区域，不会关闭对话框。

通常情况，避免出现滚动条。空间不足时允许滚动，滚动条建议默认显示。

对话框中，取消类操作项放在左边，引起变化的操作项放在右边。要写明操作项的具体效果，不要只写“是”和“否”。标题文字要明确，即使不读正文内容也能知道在干什么，标题不要用“确定吗”这样的含糊措辞。

在对话框中改变内容，不会提交数据，点击确定后，才会发生变化。

对话框上方不能再层叠对话框。

对话框可以是全屏式的，全屏对话框上方可以再层叠对话框。一个普通界面，其中的任何改动立即生效。一个全屏对话框，其中任何改动，要点击保存后才生效，点击X取消。

全屏对话框右上角的操作项，可以是诸如保存、发送、添加、分享、更新、创建之类的操作，不要使用完成、OK、关闭这样的含糊措辞。

只有必填项都填了，右上角的操作项才变为可点击状态。

内容发生了改变，点左上角的X，需要有个确认对话框，提示是否忽略修改。内容没有发生改变，点左上角的X，直接退出全屏对话框。

对话框的四周留白比较大，通常是24dp。

列表中有头像、图片等元素时，使用内嵌分隔线，左端与文字对齐。没有头像、图标等元素时，需要用通栏分隔线。图片本身就起到划定区域的作用，相册列表不需要分隔线。

Grid-cell包含主操作区和副操作区，副操作区的位置可以在上下左右4个角落。在同一个网格中，主、副操作区的内容与位置要保持一致。两者的操作都应该直接生效，不能触发菜单。

网格只能垂直滚动。单个cell不支持滑动手势，也不鼓励使用拖放操作。

网格中的单元格间距是2dp或8dp。

列表由行构成，行内包含瓦片。如果列表项内容文字超过3行，请改用卡片。如果列表项的主要区别在于图片，请改用网格。列表包含主操作区与副操作区。副操作区位于列表右侧，其余都是主操作区。在同一个列表中，主、副操作区的内容与位置要保持一致。在同一个列表中，滑动手势操作保持一致。

主操作区与副操作区的图标或图形元素是列表控制项，列表的控制项可以是勾选框、开关、拖动排序、展开/收起等操作，也可以包含快捷键提示、二级菜单等提示信息。

顺序固定的菜单，操作频繁的选项放在上面。顺序可变的菜单，可以把之前用过的选项排在前面，动态排序。菜单尽量不要超过2级。当前不可用的选项要显示出来，让用户知道在特定条件可以触发这些操作。菜单原地展开，盖住当前选项，当前选项应该成为菜单的第一项。

菜单的当前选项，始终与当前选项水平对齐。靠近屏幕边缘时，位置可适当错开。菜单过长时，需要显示滚动条。菜单从当前选项固定位置展开，不要跟随点击位置改变。菜单到上下留出8dp距离。

日期和时间选择器是固定组件，在小屏幕设备中，通常以对话框形式展现。

线形进度条只出现在纸片的边缘。

确定结束时间的由左至右前进，不确定时间的由左至右循环滚动，正在确定时间的由右至左循环滚动。

环形进度条也分时间已知和时间未知两种。环形进度条可以用在悬浮按钮上。加载详细信息时，也可以使用进度条。下拉刷新的动画比较特殊，列表不动，出现一张带有环形进度条的纸片。

滑块左右两边可以放置图标，或可编辑文本框。非连续的滑块，需要标出具体数值。

Snackbars至多包含一个操作项，不能包含图标。不能出现一个以上的Snackbars。

Snackbars在移动设备上，出现在底部。在PC上，应该悬浮在屏幕左下角。Snackbars不能遮挡住悬浮按钮，悬浮按钮要上移让出位置。Snackbars的留白比较大，24dp。

必须所有选项保持可见时，才用Radio button。不然可以使用下拉菜单，节省空间。单个开关项建议使用Switch。

tab只用来展现不同类型的内容，不能当导航菜单使用。tab至少2项，至多6项。超出6项，tab需要变为滚动式，左右翻页。tab文字要显示完整，字号保持一致，不能折行，文字与图标不能混用。tab选中项的下划线高度是2dp。

EditView激活状态和错误状态，横线的宽度变为2dp，颜色改变。输入框点击区域高度至少48dp，但横线并不在点击区域的底部，还有8dp距离。整个点击区域增高，提示文字也是点击区域的一部分。右下角可以加入字数统计。字数统计不要默认显示，字数接近上限时再显示出来。通栏输入框也可以有字数统计，单行的字数统计显示在同一行右侧。错误提示显示在输入框的左下方。默认提示文本可以转换为错误提示。输入框尽量带有自动补全功能。

提示只用在小图标上，文字不需要提示。鼠标悬停、获得焦点、手指长按都可以触发提示。触摸提示（左）和鼠标提示（右）的尺寸是不同的，背景都带有90%的透明度。

侧边栏从左侧滑出，占据整个屏幕高度，遵循普通列表的布局规则。手机端的侧边栏距离屏幕右侧56dp。

设置和帮助反馈通常放在侧边抽屉中。如果没有侧边抽屉，则放在Appbar的下拉菜单底部。

设置界面只能包含设置项，诸如关于、反馈之类的界面，入口应该放在其他地方。

设置项使用通栏分隔线来分组。7项以下不必分组。如果某项独立一组，考虑把它放在顶部（重要）或放在底部的“其他”一栏中（不重要）。设置项较多时尝试合并，比如把两个相关的勾选项合并成一个多选项。设置项非常多时，使用子界面。

可用控件相关：

导入依赖库：compile 'com.android.support:design:24.2.1'

控件所属包：android.support.design.widget

1. SnakeBar

类似Toast，但不同点在于其依赖于父视图存在，并且有事件响应处理能力。

代码中使用：

SnakeBar.make(rootView,”message”,SnakeBar.LENGTH\_SHORT)

.setAction(“ActionName”,OnClickListener).show();//过程中还能控制视图的颜色等属性，表达式是链式的。

1. TextInputLayout

貌似就是为EditText设置了拓展效果，具体功能待实验。

代码中使用：

<android.support.design.widget.TextInputLayout>

…

<EditText

…

/>

</android.support.design.widget.TextInputLayout>

三、FloatingActionButton

一个悬浮按钮，一般呆在界面右下角。

代码中使用：

<android.support.design.widget.FloatingActionButton

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom|end"

android:layout\_margin="16dp"

android:src="@android:drawable/ic\_dialog\_email"

app:borderWidth="2dp"//边框宽度

app:backgroundTint=””//按钮背景颜色，默认为ColorAccent

app:pressedTranslationZ=”12dp”//点击时阴影深度，默认为12dp

app:elevation=”6dp”//阴影深度，默认为6dp

app:fabSize="normal" //大小设置

app:rippleColor="#ff0000"//点击水纹颜色 />

1. CoordilatorLayout

协作式布局，可是使布局内的子控件有一定的动作同步或配合（如AppBar在浏览正文时的隐藏效果）

在CoordinatorLayout中，当AppbarLayout中的Toolbar声明协作flag并且其他控件响应其触发事件时，其他布局会自动处于AppbarLayout下方，否则会按照一般FramLayout布局对其进行覆盖。

ScrollView和CoordinatorLayout有冲突，需要用NestedScrollView取代。上划隐藏Appbar时会出现覆盖状态栏的问题，此时应该取消fitSystemWindow，但于此同时状态栏颜色会消失。

layout\_scrollFlags中的几个值：

scroll:将此布局和滚动时间关联。这个标识要设置在其他标识之前，没有这个标识则布局不会滚动且其他标识设置无效。

enterAlways:任何向下滚动操作都会使此布局可见。这个标识通常被称为“快速返回”模式。

enterAlwaysCollapsed：假设你定义了一个最小高度（minHeight）同时enterAlways也定义了，那么view将在到达这个最小高度的时候开始显示，并且从这个时候开始慢慢展开，当滚动到顶部的时候展开完。

exitUntilCollapsed：当你定义了一个minHeight，此布局将在滚动到达这个最小高度的时候折叠。

snap:当一个滚动事件结束，如果视图是部分可见的，那么它将被滚动到收缩或展开。例如，如果视图只有底部25%显示，它将折叠。相反，如果它的底部75%可见，那么它将完全展开。

事件产生源需要设置 app:layout\_behavior:="" 其value可以是系统的已有定义，也可以是自定义Behavior。

Toolbar的点击事件设置可以在OnOptionsItemSelected中或者OnMenuItemSelected中，后者需要在设置supportActionBar后再设置监听。其左上角返回键的id为android.R.id.home。

1. NavigationView

导航页，分为需要外部导入的headerLayout部分和menu部分，可以作为DrawLayout的导航页布局。

代码中使用：

Xml布局中 <android.support.design.widget.NavigationView

…

app:headerLayout=”@layout/navigation\_header”

app:menu=”@menu/drawer”/>

Java代码中设置点击事件：navigationView.setNavigationItemSelectedListener(new NavigationVIew.OnNavifationItemSelectedListener(){

@Override

Public Boolean onNavigationItemSelected(MenuItem item){

…

}

});

1. AppBarLayout

Toolbar的外部拓展布局，与CoordinatorLayout共同使用时可以让其内的Toolbar响应ScrollingViewBehavior事件

1. CollapsingToolbarLayout

让Toolbar可伸缩，实现Toolbar滚出屏幕或者部分收起固定在顶部的效果

app:layout\_collapseMode 设置子控件的模式 有parallax(视差效果) pin(固定) 和off(关闭效果) 三种模式可选  
app:layout\_collapseParallaxMultiplier 设置视差比例 0-1

app:collapsedTitleGravity 设置折叠时标题的位置

app:collapsedTitleAppearance 完全折叠时标题的字体

app:contentScrim Toolbar 屏幕充分滚动时内容部分的遮罩

app:statusBarScrim Toolbar屏幕充分滚动时状态栏的遮罩

app:title 标题

app:titleEnable 是否显示标题

app:toolbarId 需要折叠效果的Toolbar Id

app:expandedTitleGravity 布局展开时的标题位置

app:expandedTitleMargin 布局展开时标题的边距

TabLayout

实现ViewPager标签效果。

1. CardView

卡片布局。

app:cardBackgroundColor 背景颜色  
app:cardCornerRadius     设置圆角。  
app:cardElevation         阴影。  
app:cardMaxElevation       最大阴影。  
app:cardPreventCornerOverlap 在v20和之前的版本中添加内边距，这个属性是为了防止卡片内容和边角的重叠。  
app:cardUseCompatPadding   设置内边距，v21+的版本和之前的版本仍旧具有一样的计算方式，设置其为true才能使卡片有阴影。