# 综合实验1 《小吃APP》页面架构设计

## 一、实验简介

本综合实验将以《小吃APP》作为本课程的结课项目，该项目共有4个用户页面，分别为：【新品页面】、【点菜页面】、【下单页面】和【我的页面】，其页面效果如下图所示。

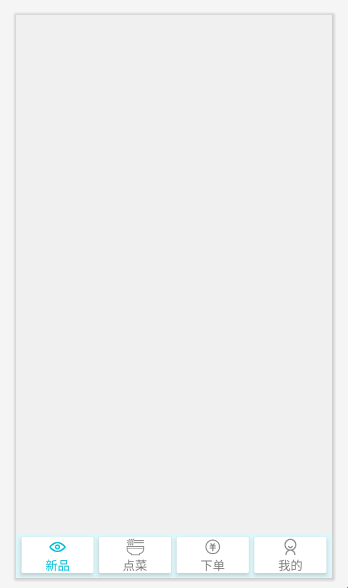
一个APP通常都有多个页面，有的甚至有几十个页面，如果没有一个框架统筹管理这么多的页面，实现时将会比较凌乱，本综合实验将把理论课上的Fragment技术和ViewPager2技术结合使用作为本App的主页面框架技术来管理项目的四个页面。

## 二、实验目标

* 掌握CardView控件的布局
* 实现页面的整体架构排版布局

## 实验操作步骤

### 实现效果



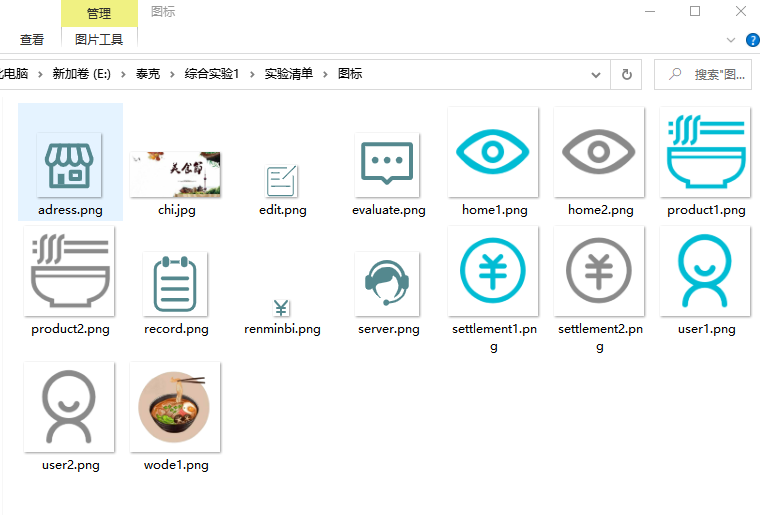
### 实现步骤

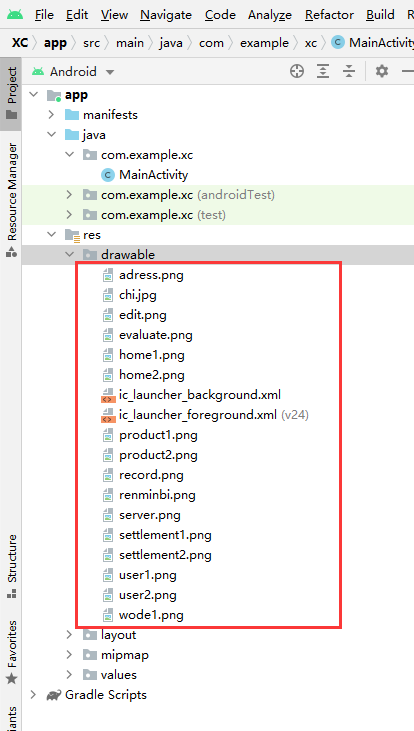
#### 第1步：创建一个名为【XC】的项目；

具体创建步骤详见实验1。

#### 第2步：复制图标到项目中；

打开【实验清单】里的【图标】文件夹，复制里面的所有图片（共16张），并粘贴到<app\res\drawable\>目录下，如下图所示，具体复制过程可参考实验5第1.2小节第2步。





#### 第3步：修改根布局为LinearLayout布局控件

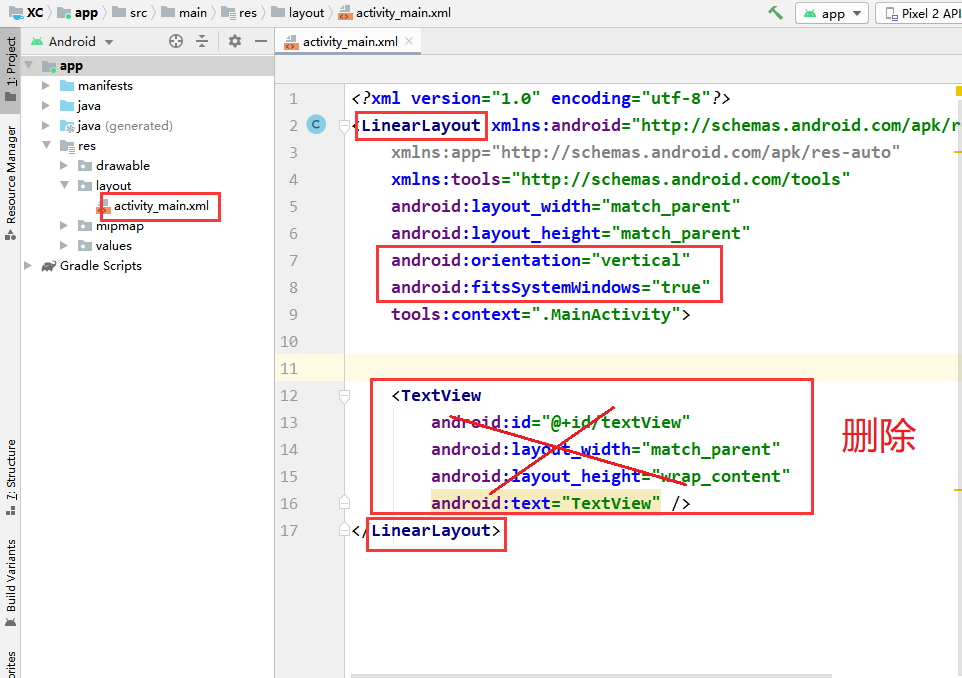
以【Split】形式打开<app\res\layout\>目录下的【activity\_main.xml】布局文件，并插入或编辑为下面的两段代码，其作用为：让布局里的控件整体上呈上下结构的排版。

LinearLayout

android:orientation="vertical"

android:fitsSystemWindows="true"

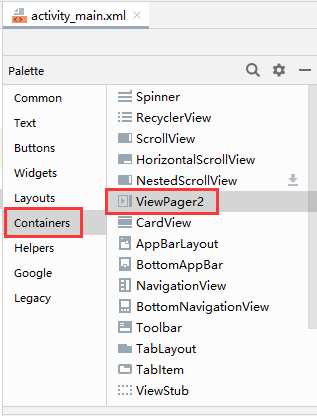
上述代码插入或编辑的具体位置如下所示



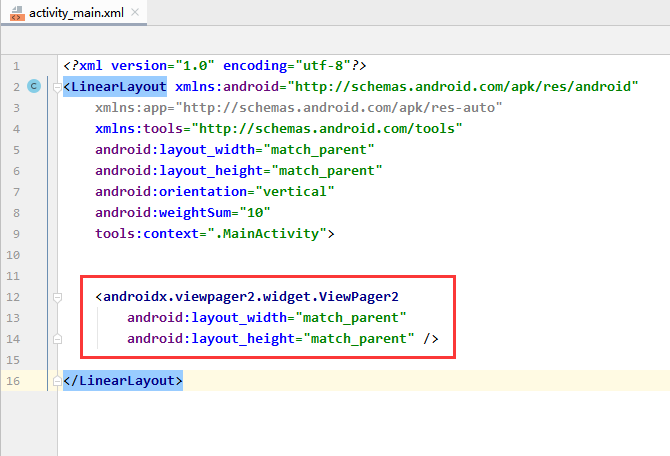
其中【android:fitsSystemWindows="true"】用于布局时给系统窗口留出空间，比如系统自带的导航栏等。

#### 第4步：添加ViewPager2控件

先以【Design】形式打开【activity\_main.xml】布局文件，ViewPager2控件在控件栏里的【Containers】选项中，如下图所示。

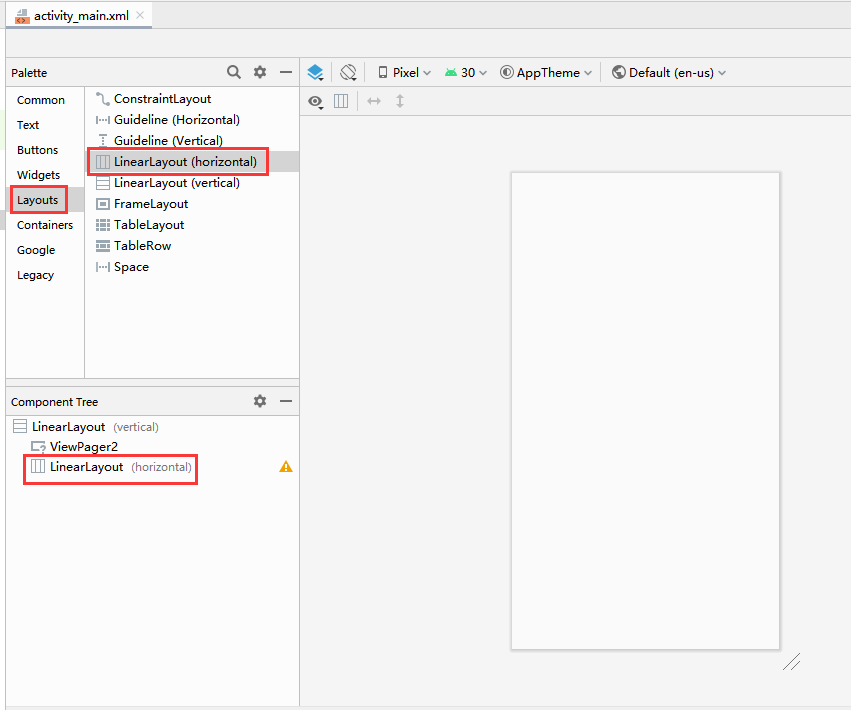


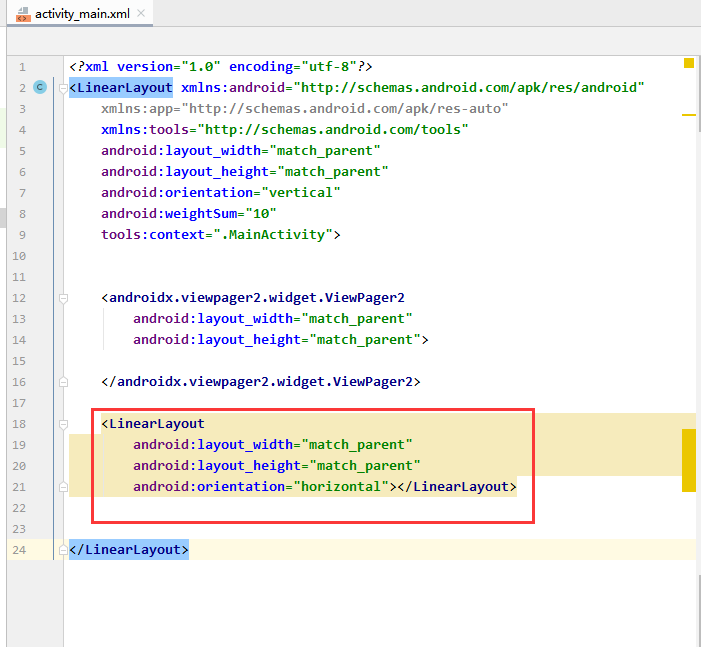
然后用拖拽形式或手动编写形式，把ViewPager2控件添加到布局中，如下图所示。



#### 第5步：添加LinearLayout控件

在【activity\_main.xml】文件里以拖拽形式或手动编写形式，添加一个LinearLayout(horizontal)控件，如下图所示。





#### 第6步：按比例分配控件大小

把下面的三段代码插入或编辑到【activity\_main.xml】文件中，其效果为：把整个页面分为10等分，其中ViewPager2控件占9.2等分，LinearLayout(horizontal)控件占0.8等分。

android:weightSum="10"

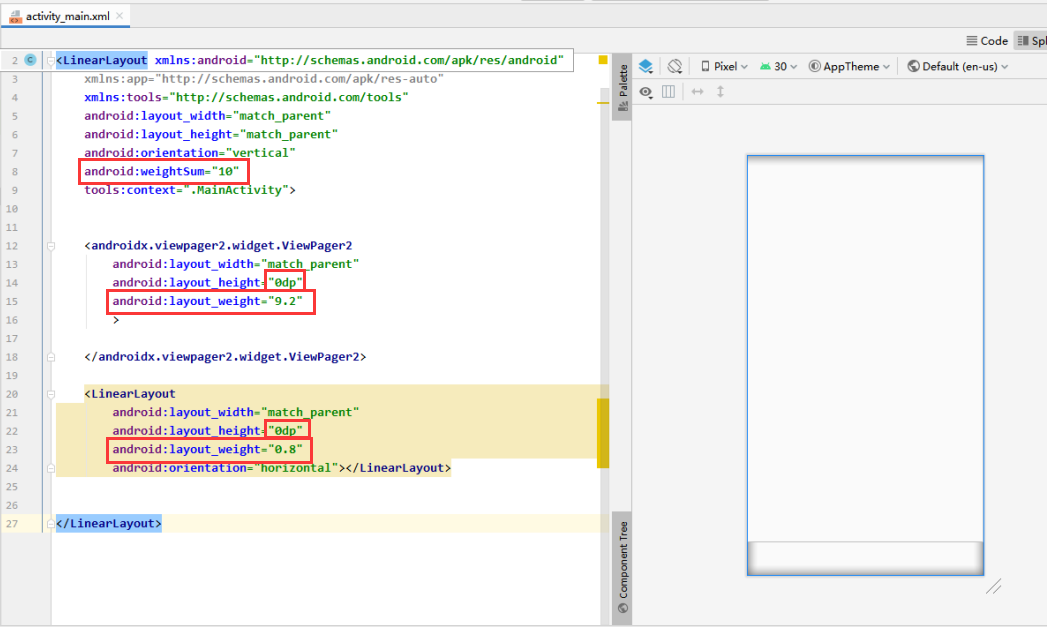
android:layout\_height="0dp"

android:layout\_weight="9.2"

android:layout\_height="0dp"

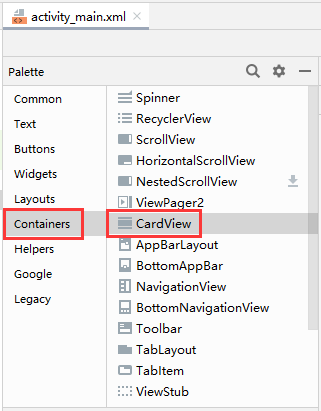
android:layout\_weight="0.8"

上述代码插入或编辑的具体位置入下图所示

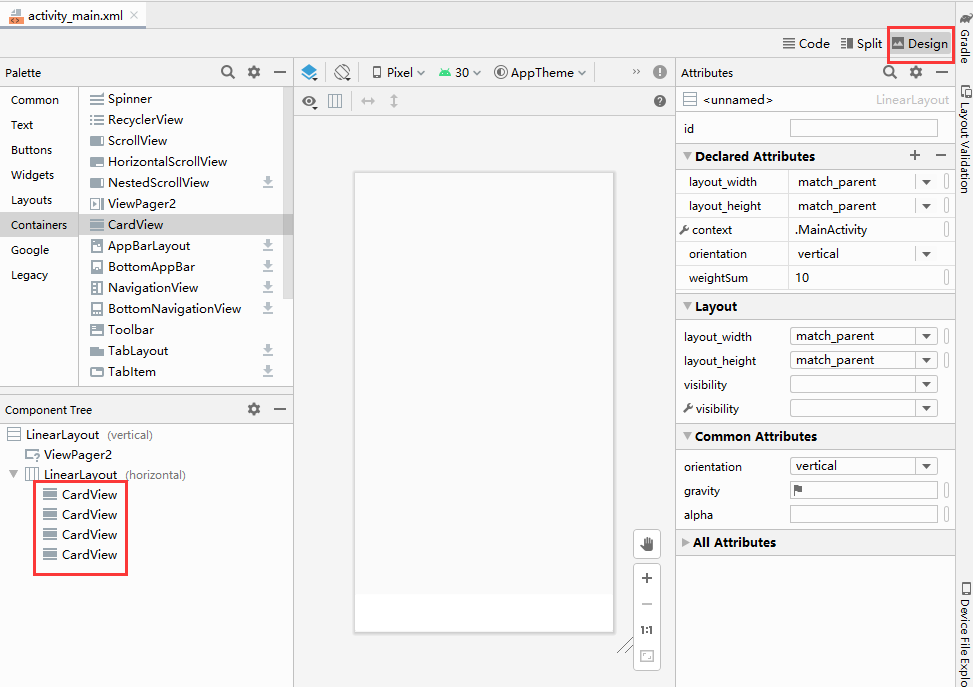


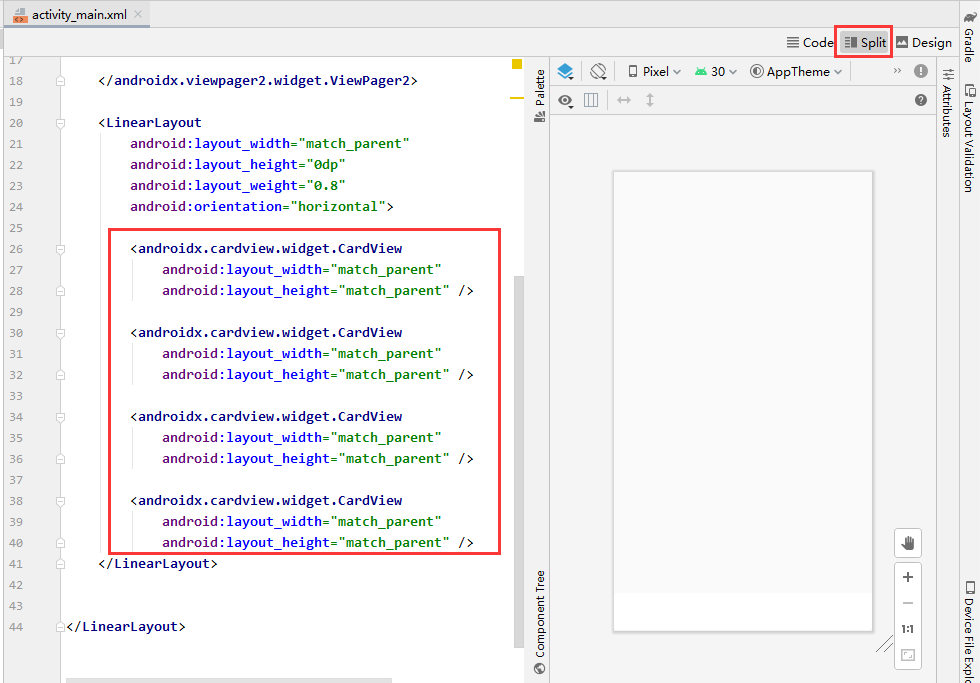
#### 第7步：添加四个CardView控件

先以【Design】形式打开【activity\_main.xml】文件，CardView控件在控件栏里的【Containers】选项中，如下图所示。



然后用拖拽形式或手动编写形式，添加四个CardView控件到【activity\_main.xml】文件的LinearLayout(horizontal)控件标签内，具体添加位置如下图所示，其作用为：以卡片形式呈现底部导航选项。





#### 第8步：按比例分配控件大小

把下面两段代码插入或编辑到【activity\_main.xml】文件里，其效果为：把LinearLayout(horizontal)控件所在区块分成4等分，其中每个CardView控件占1等分。

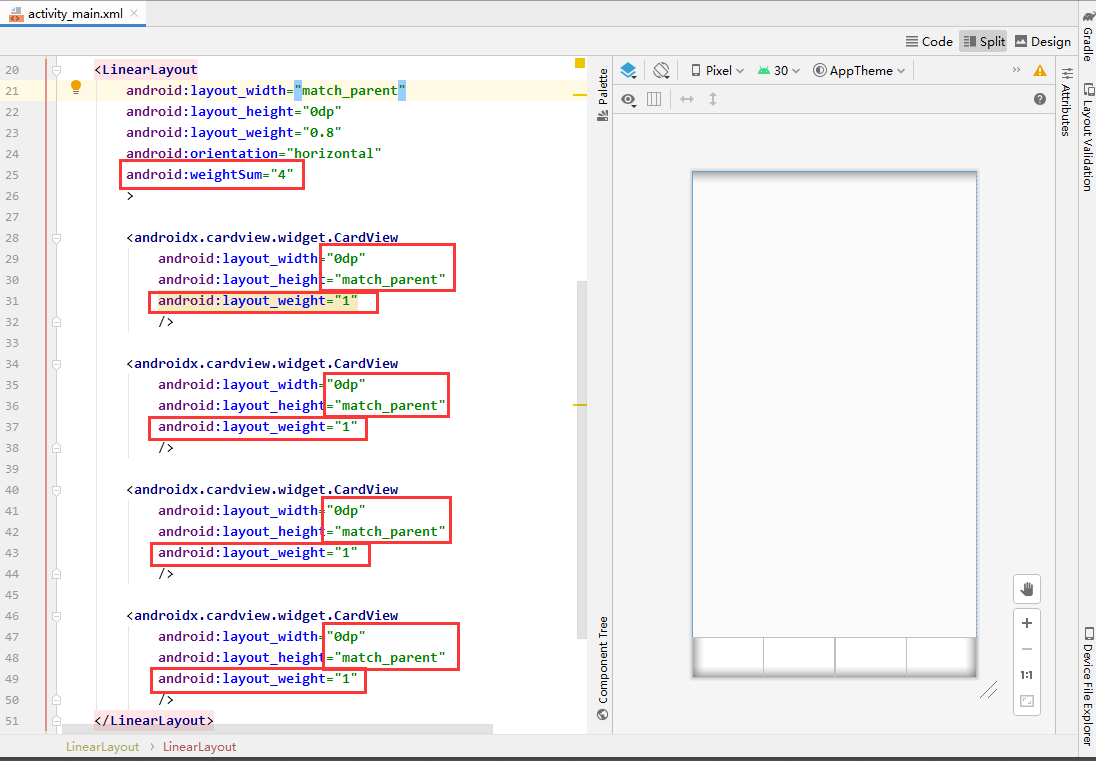
android:weightSum="4"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="1"

上述代码插入或编辑的具体位置入下图所示



#### 第9步：修改CardView控件

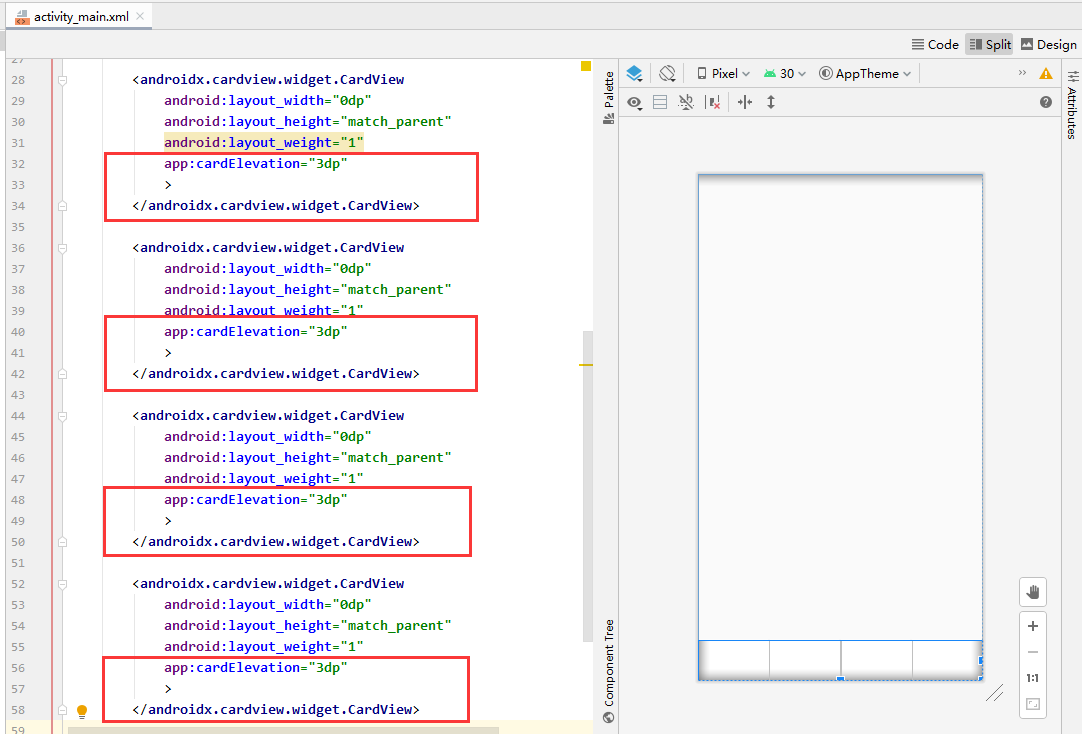
把下面的代码插入或编辑到每个CardView控件，其作用为：给卡片（CardView）增加阴影效果，并使CardView控件标签内可插入其他控件。

app:cardElevation="3dp"

>

</androidx.cardview.widget.CardView>

上述代码插入或编辑的具体位置入下图所示



#### 第10步：设计CardView布局

把下面的代码插入到每个CardView控件标签内，其效果为：为卡片（CardView）添加图片控件和文本控件，并按比列分配好大小，卡片控件CardView所在区块分成5等分，其中图片控件占2.8等分，文本控件占2.2等分。

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:weightSum="5"

>

<ImageView

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="2.8" />

<TextView

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="2.2" />

</LinearLayout>

上述代码插入或编辑的具体位置入下图所示



#### 第11步：设置选项图片和文字

给上面添加的4个ImageView控件和4个TextView控件插入下面的八段代码，其效果为：给每个选项添加相应的图片和文字，并让文字居中显示。

4个ImageView控件分别插入的代码为

android:src="@drawable/home1"

android:src="@drawable/product2"

android:src="@drawable/settlement2"

android:src="@drawable/user2"

4个TextView控件分别插入的代码为

android:text="新品"

android:gravity="center"

android:text="点菜"

android:gravity="center"

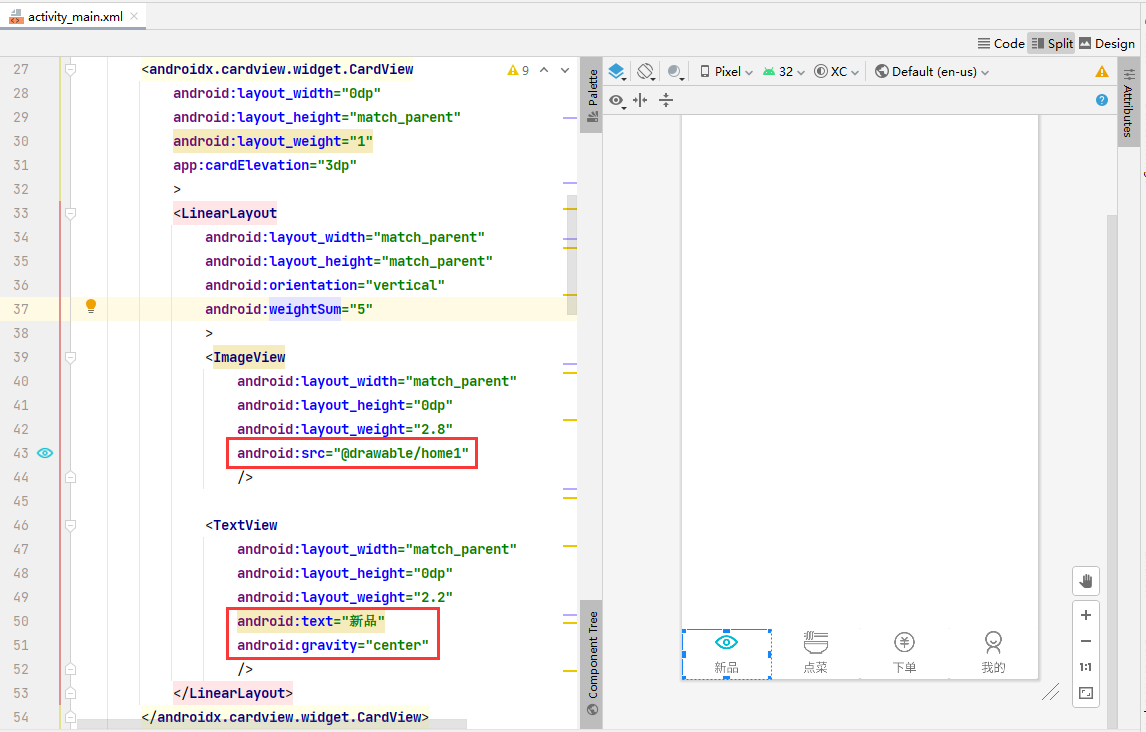
android:text="下单"

android:gravity="center"

android:text="我的"

android:gravity="center"

上述代码插入的具体位置和效果如下图所示



#### 第12步：设置控件边距位置

把下面的两段属性代码插入到每个CardView控件标签，其效果为：每个选项离其他控件的左边距为7dp，上边距为5dp，右边距为7dp，下边距为6dp。

【新品】、【点菜】、【下单】选项所在CardView控件要插入的代码为

android:layout\_marginLeft="7dp"

android:layout\_marginBottom="6dp"

android:layout\_marginTop="5dp"

【我的】选项所在CardView控件要插入的代码为

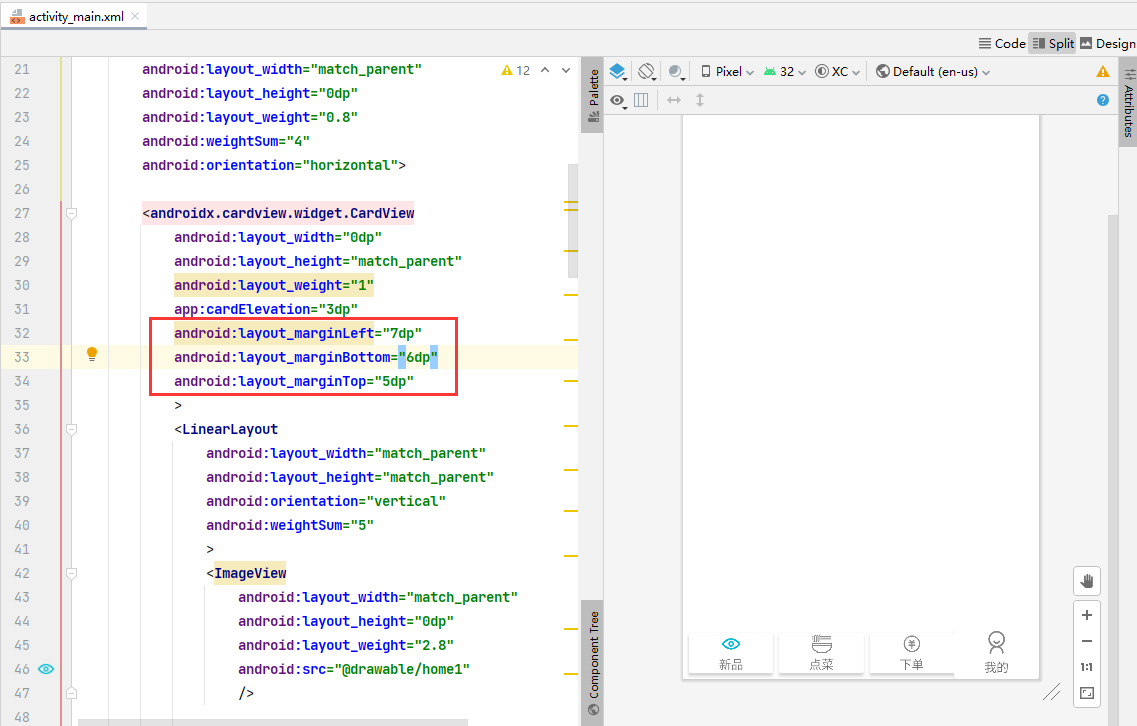
android:layout\_marginLeft="7dp"

android:layout\_marginBottom="6dp"

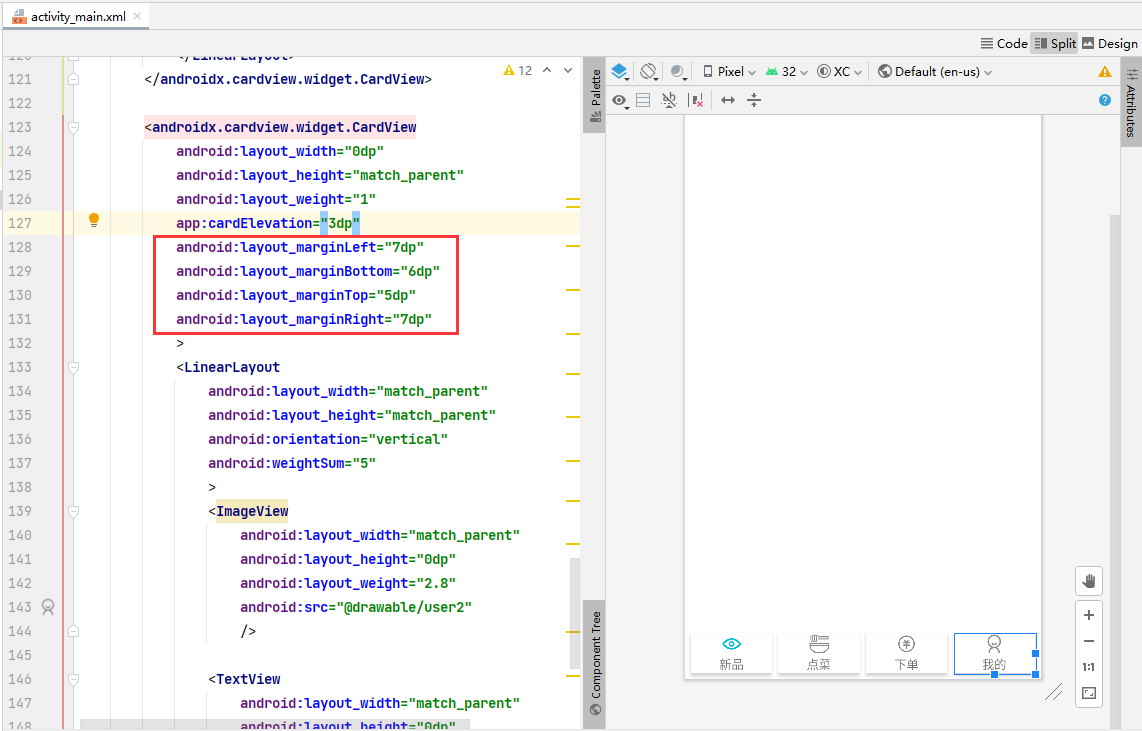
android:layout\_marginTop="5dp"

android:layout\_marginRight="7dp"

上述代码插入到【新品】、【点菜】、【下单】选项所在CardView控件的具体位置和效果如下图所示



上述代码插入到【我的】选项所在CardView控件的具体位置和效果如下图所示



#### 第13步：设置颜色

先在<app\res\values\>目录下的【colors.xml】文件的【resources】标签里插入下面的代码，其作用是定义颜色属性。

<color name="colorPrimary">#00BCD4</color>

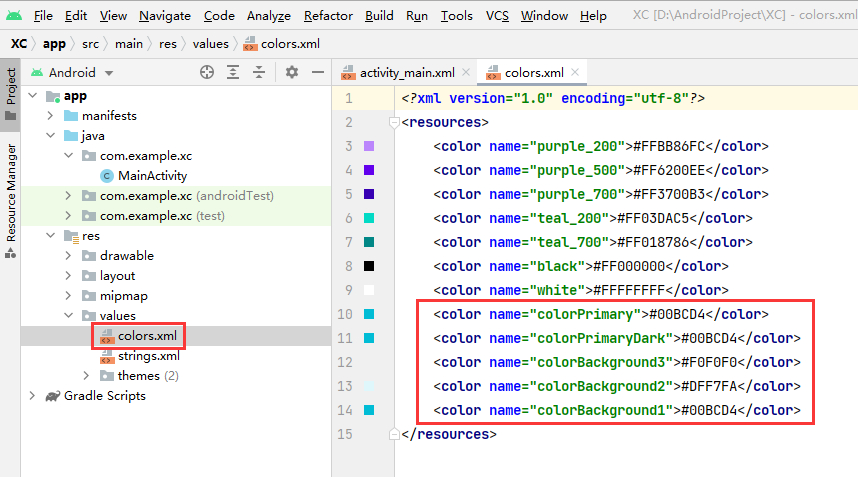
<color name="colorPrimaryDark">#00BCD4</color>

<color name="colorBackground3">#F0F0F0</color>

<color name="colorBackground2">#DFF7FA</color>

<color name="colorBackground1">#00BCD4</color>

上述代码插入的具体位置如下图所示



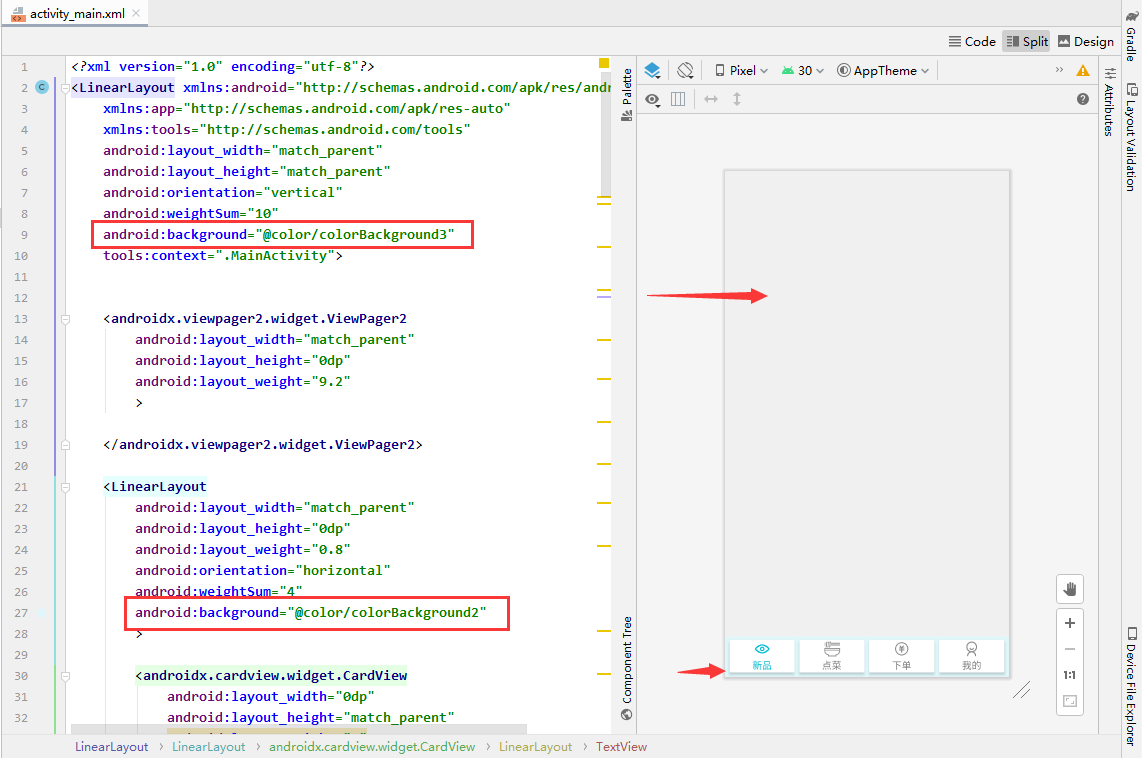
然后再在布局文件里用下面几段代码调用上述颜色值属性，其效果为：把整个布局背景设置为淡灰色，把底部导航选项背景设置为淡青色，把默认选项文字设置为青蓝色。

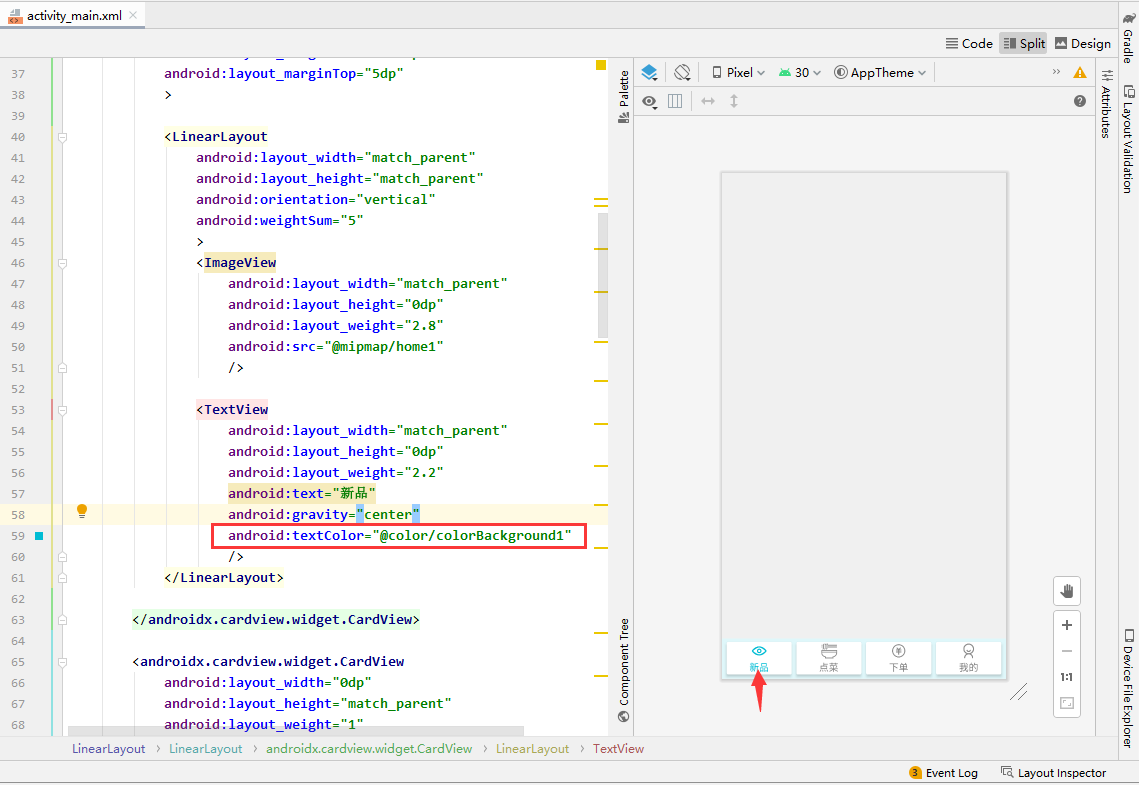
android:background="@color/colorBackground3"

android:background="@color/colorBackground2"

android:textColor="@color/colorBackground1"

上述代码插入的具体位置和效果如下图所示



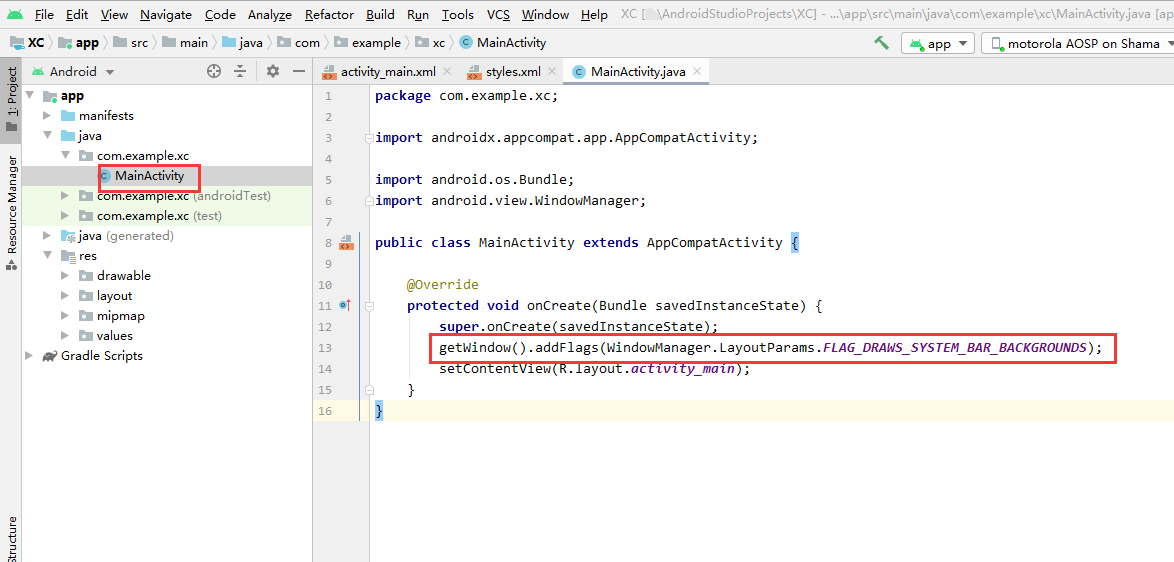


#### 第14步：设置标题栏状态栏

先在<app\java\com.example.xc\>目录下的【MainActivity】类里插入下面的代码，其作用是去掉默认标题栏。

getWindow().addFlags(WindowManager.LayoutParams.FLAG\_DRAWS\_SYSTEM\_BAR\_BACKGROUNDS);

上述代码插入的具体位置如下图所示



上述代码在编写的时候需对WindowManager接口导包，需要导入

import android.view.WindowManager;

在Android里快捷键导包的方式是，先鼠标点中对应的单词或类，再按住【Alt】键+【Enter】键即可；

再在<app\res\values\>目录下两个【themes.xml】文件里的【resources】标签内插入下面的代码，其效果为：把状态栏背景色设置为透明，并让整个布局界面位于其下面。

<style name="Theme.XC" parent="Theme.AppCompat.DayNight.NoActionBar">

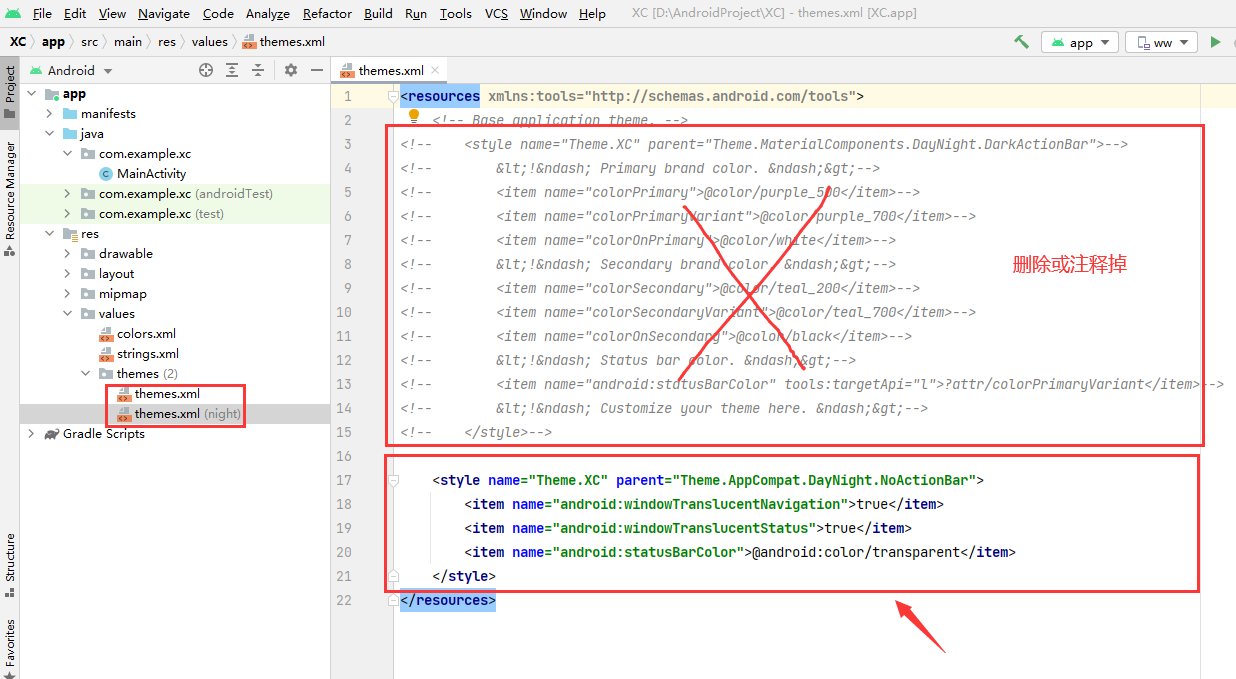
<item name="android:windowTranslucentNavigation">true</item>

<item name="android:windowTranslucentStatus">true</item>

<item name="android:statusBarColor">@android:color/transparent</item>

</style>

上述代码插入的具体位置如下图所示



其中，【Theme.XC】里的XC是自己的项目名。

#### 第15步：运行测试

上面所有的步骤都正确实现后，就可以生成apk文件运行测试了。具体步骤参考实验1的第2大步和第3大步，把App安装到手机后的效果如下图所示。

