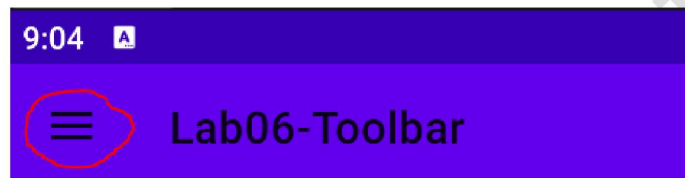


## Lab06 - 1 – Navigation Menu

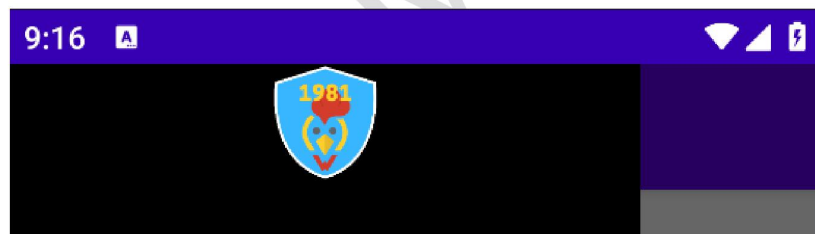
**Navigation Menu** hay **Navigation Drawer** (Menu Điều hướng): Là một loại menu xuất hiện từ bên trái hoặc bên phải của màn hình khi người dùng nhấn vào nút "hamburger" (ba dấu gạch ngang). Menu điều hướng này thường chứa các liên kết hoặc biểu tượng đại diện cho các phần của ứng dụng, cho phép người dùng dễ dàng điều hướng giữa các màn hình hoặc chức năng khác nhau.

Thành phần của Navigation menu:

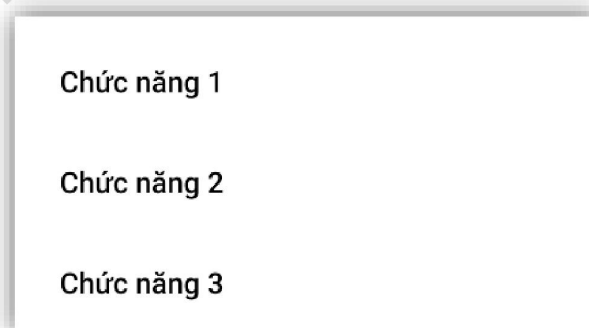
- Nút điều hướng (Hamburger button): Khi click vào nút sẽ mở ra menu.



- Header Menu: Nằm ở phía trên của menu, thường chứa Logo hoặc tên của ứng dụng.



- Danh sách các item có trong menu:

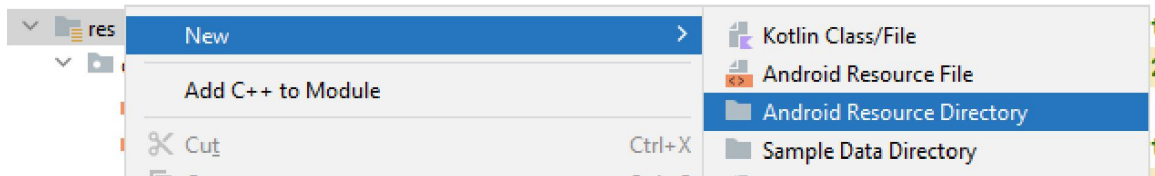


## *Xây dựng các thành phần của menu:*

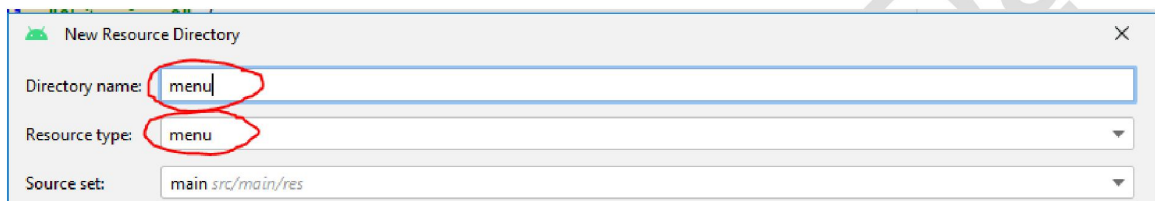
### 1. Xây dựng danh sách item.

Tạo thư mục mới trong res, đặt tên là menu:

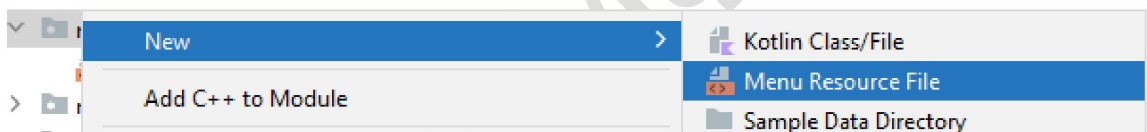
Click chuột phải vào res rồi chọn New/Android Resource Directory:



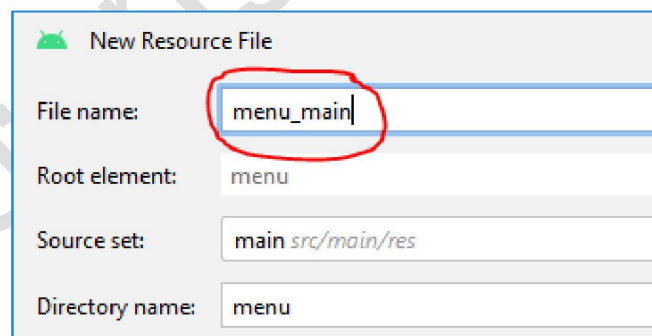
Chọn Resource Type là **menu**, đặt tên thư mục là menu:



Trong thư mục menu, click chuột phải chọn New/Menu Resource File:



Đặt tên cho file, giả sử là **menu\_main**:



```

1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android">
3      <!-- Danh sách các item trong menu -->
4      <!-- Đặt id cho mỗi item để gọi ra khi cần sử dụng -->
5      <item
6          android:id="@+id/nav_function1"
7          android:title="Chức năng 1" />
8      <item
9          android:id="@+id/nav_function2"
10         android:title="Chức năng 2" />
11     <item
12         android:id="@+id/nav_function3"
13         android:title="Chức năng 3" />
14 </menu>

```

## 2. Tạo Header cho menu:

Trong res/layout, tạo file header, giả sử đặt tên là **nav\_header\_main.xml**

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="100dp"
5      android:background="@color/black"
6      android:gravity="center"
7      android:orientation="vertical"
8      android:padding="15dp"
9      android:theme="@style/ThemeOverlay.AppCompat.Dark">
10
11      <!-- Chứa logo của ứng dụng -->
12      <ImageView
13          android:layout_width="50dp"
14          android:layout_height="50dp"
15          android:src="@drawable/logo1981" />
16
17 </LinearLayout>

```

## 3. Tạo thanh Toolbar chứa nút điều hướng.

Trong res/layout, tạo file **content\_main.xml** để xây dựng toolbar:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   tools:context=".MainActivity">
7
8   <androidx.appcompat.widget.Toolbar
9     android:id="@+id/toolbar"
10    android:layout_width="match_parent"
11    android:layout_height="?attr/actionBarSize"
12    android:background="?attr/colorPrimary"
13    android:elevation="4dp"
14    android:theme="@style/ThemeOverlay.AppCompat.ActionBar" />
15 </RelativeLayout>

```

Trong đó:

- **android:layout\_height="?attr/actionBarSize"**: Thuộc tính này đặt chiều cao của Toolbar bằng với kích thước của action bar mặc định của ứng dụng, giúp Toolbar có kích thước phù hợp với các thành phần khác trong action bar.
- **android:background="?attr/colorPrimary"**: Thuộc tính này đặt màu nền cho Toolbar bằng với màu chính được định nghĩa trong theme của ứng dụng (colorPrimary). Điều này giúp Toolbar phù hợp với giao diện chung của ứng dụng.
- **android:elevation="4dp"**: Thuộc tính này nâng Toolbar lên cao hơn so với nền 4dp. Điều này tạo ra một hiệu ứng nâng Toolbar lên khi nó đè lên các thành phần khác, tạo ra một hiệu ứng 3D.
- **android:theme="@style/ThemeOverlay.AppCompat.ActionBar"**: Thuộc tính này áp dụng một theme overlay lên Toolbar, giúp Toolbar có cùng giao diện với action bar của ứng dụng.

#### **4. Xây dựng layout chính của ứng dụng (activity\_main.xml)**

Layout chứa menu để ở dạng **DrawerLayout**. DrawerLayout được thiết kế đặc biệt để triển khai và quản lý thanh điều hướng (navigation drawer) trong ứng dụng Android.

Mặc dù bạn có thể sử dụng LinearLayout hoặc ConstraintLayout để tạo một thanh điều hướng tương tự, nhưng điều này sẽ đòi hỏi bạn phải



tự quản lý việc mở và đóng của thanh điều hướng, cũng như việc xử lý các tương tác của người dùng, như trượt ra và vào.

**DrawerLayout** đã được tối ưu hóa để xử lý những nhiệm vụ này một cách dễ dàng và hiệu quả. Nó cung cấp các phương thức và sự kiện được tích hợp sẵn để bạn có thể dễ dàng điều khiển thanh điều hướng và xử lý các sự kiện liên quan.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.drawerlayout.widget.DrawerLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:id="@+id/drawer_layout"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     android:fitsSystemWindows="true"
10    tools:openDrawer="start">
11
12    <!-- Đưa header đã tạo vào menu-->
13    <include
14        layout="@layout/content_main"
15        android:layout_width="match_parent"
16        android:layout_height="match_parent" />
17
18    <com.google.android.material.navigation.NavigationView
19        android:id="@+id/nav_view"
20        android:layout_width="wrap_content"
21        android:layout_height="match_parent"
22        android:layout_gravity="start"
23        android:fitsSystemWindows="true"
24        app:headerLayout="@layout/nav_header_main"
25        app:menu="@menu/menu_main" />
26
27 </androidx.drawerlayout.widget.DrawerLayout>
```

Trong đó:

- **include**: Là một cách để chèn một layout khác vào trong layout hiện tại. Trong trường hợp này, nó chèn layout content\_main vào DrawerLayout.

- `android:layout_gravity="start"`: Thiết lập vị trí của `NavigationView` sao cho nó xuất hiện từ phía trái của `DrawerLayout`.
- `app:headerLayout="@layout/nav_header_main"`: Định nghĩa layout cho phần header của `NavigationView`.
- `app:menu="@menu/menu_main"`: Đặt menu cho `NavigationView`. Menu này sẽ hiển thị các mục điều hướng trong `NavigationView`.

### 5. Định nghĩa chuỗi mô tả hành động mở/đóng `DrawerLayout`

Mở `res/values/strings.xml` và thêm vào đoạn định nghĩa:

```
1 <resources>
2     <string name="app_name">Lab06-Toolbar</string>
3     <string name="navigation_drawer_open">Mở Navigation Drawer</string>
4     <string name="navigation_drawer_close">Đóng Navigation Drawer</string>
5 </resources>
```

### **Viết mã xử lý cho menu**

```
public class MainActivity extends AppCompatActivity {
    3 usages
    private DrawerLayout drawer;
    3 usages
    private Toolbar toolbar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        drawer = findViewById(R.id.drawer_layout);
        toolbar = findViewById(R.id.toolbar);
        // Thiết lập toolbar của Activity là ActionBar
        setSupportActionBar(toolbar);
        // Tạo đối tượng thực hiện hành động tương tác đóng - mở trên menu bằng nút "Hamburger"
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            activity: this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
        // Theo dõi các sự kiện mở và đóng của menu (drawer)
        drawer.addDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {
            1 usage
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) { return true; }
        });
    }
}
```