

Lab06 - 2 – Context Menu

Trong Android, "Context menu" là một loại menu xuất hiện khi người dùng thực hiện một hành động dài (long press) trên một phần tử giao diện người dùng như một nút, một hình ảnh, hoặc một vùng văn bản. Context menu cung cấp các tùy chọn hoặc hành động liên quan đến phần tử đó, phụ thuộc vào ngữ cảnh hoặc trạng thái của ứng dụng.

Thông thường, context menu xuất hiện dưới dạng một danh sách các tùy chọn được hiển thị trong một popup menu, cho phép người dùng chọn một hành động phù hợp với nhu cầu của họ. Context menu thường được sử dụng để cung cấp các tùy chọn điều khiển nhanh chóng và thuận tiện, giúp tăng cường trải nghiệm người dùng trong ứng dụng Android.

Để tạo context menu trong Android, bạn cần thực hiện các bước sau:

Định nghĩa menu trong tệp res/menu.

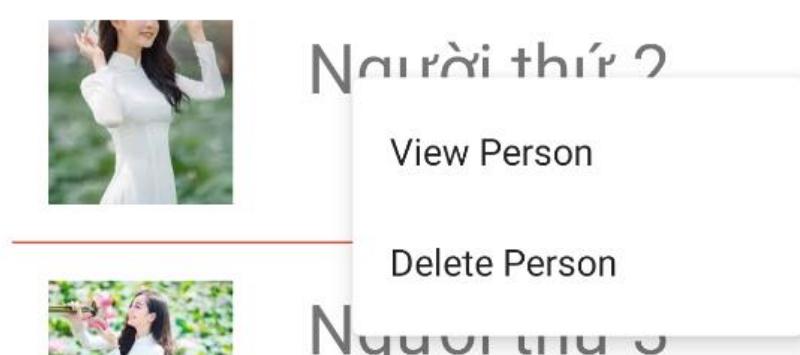
Gắn menu với phần tử giao diện người dùng bằng cách sử dụng phương thức `registerForContextMenu()` trong Activity hoặc Fragment.

Override phương thức `onCreateContextMenu()` để tạo context menu và định nghĩa các tùy chọn.

Override phương thức `onContextItemSelected()` để xử lý sự kiện khi người dùng chọn một tùy chọn trong context menu.

Ví dụ tạo Context Menu: Dựa trên bài thực hành tạo ListView

Khi người dùng ấn và giữ 1 phần tử nào đó trên ListView sẽ hiển thị ContextMenu:



Hướng dẫn

1. Tạo ListView

- activity_main.xml:



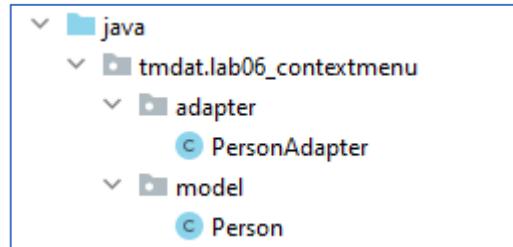
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <!--activity_main.xml-->
    <ListView
        android:id="@+id/listViewPerson"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:divider="#F44336"
        android:dividerHeight="1dp"
        android:padding="16dp" />
</RelativeLayout>
```

- list_item_person.xml:



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp">
    <!-- list_item_person.xml -->
    <ImageView
        android:id="@+id/imageViewPerson"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:src="@mipmap/ic_launcher"
        android:scaleType="centerCrop"
        android:layout_marginEnd="16dp"/>
    <!--Trong src có thể sử dụng ảnh mặc định-->
    <TextView
        android:id="@+id/textViewPersonName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="30sp"
        android:layout_toEndOf="@+id/imageViewPerson"
        android:layout_marginStart="16dp"
        android:text="Person Name" />
    <!-- Giá trị text đang để mặc định -->
</RelativeLayout>
```

- Tạo Model và Adapter tương tự bài học trước:



- Xây dựng lớp **MainActivity** xử lý hiển thị thông tin vào ListView:

```
22 public class MainActivity extends AppCompatActivity {
23     private List<Person> personList;
24     private ArrayAdapter<Person> personAdapter;
25
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         setContentView(R.layout.activity_main);
30         //Tạo danh sách dữ liệu
31         personList = new ArrayList<>();
32         personList.add(new Person(R.drawable.anh01, namePerson: "Người thứ 1"));
33         personList.add(new Person(R.drawable.anh02, namePerson: "Người thứ 2"));
34         personList.add(new Person(R.drawable.anh03, namePerson: "Người thứ 3"));
35         personList.add(new Person(R.drawable.anh04, namePerson: "Người thứ 4"));
36         //Tạo Adapter
37         personAdapter = new PersonAdapter( context: this, R.layout.list_item_person, personList);
38         //Liên kết Adapter với ListView
39         ListView listViewPerson = findViewById(R.id.listViewPerson);
40         listViewPerson.setAdapter(personAdapter);
41
42         //Kích hoạt contextMenu trên ListView
43         registerForContextMenu(listViewPerson);
44
45         //Thêm sự kiện click vào mỗi item trên ListView
46         listViewPerson.setOnItemClickListener(new AdapterView.OnItemClickListener() {
47             @Override
48             public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
49                 //Lấy thông tin người được chọn
50                 Person selectedPerson = personAdapter.getItem(position);
51                 //Hiển thị thông báo
52                 if(selectedPerson!=null){
53                     Toast.makeText( context: MainActivity.this, text: "Bạn vừa chọn "+selectedPerson.getNamePerson(),
54                                     Toast.LENGTH_SHORT).show();
55                 }
56             }
57         });
58     }
```

2. Xử lý Context Menu

- Tạo Context Menu: Tạo thư mục res/menu, trong thư mục này tạo file **context_menu.xml**



- Xử lý hành động ấn và giữ 1 item trên ListView, khi đó sẽ xuất hiện ContextMenu:

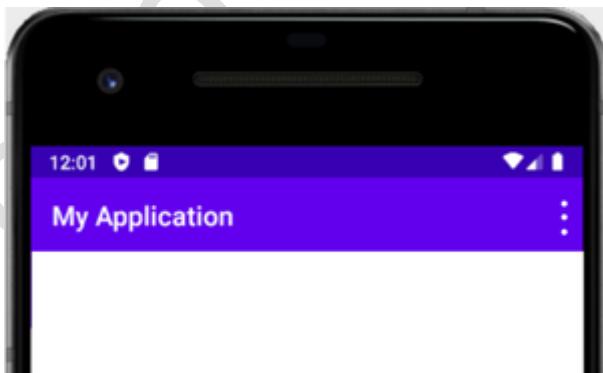
- Ví dụ các hành động trong Context Menu:

<img alt="Screenshot of the Android Studio Java code editor showing the onContextItemSelected() method. The code handles item selection from the context menu. The code is as follows:
70 // Thực hiện hành động khi người dùng chọn 1 item trong menu
71 @Override
72 public boolean onContextItemSelected(MenuItem menuItem){
73 AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) menuItem.getMenuInfo();
74 int position = info.position;
75
76 switch (menuItem.getItemId()){
77 case R.id.menu_view:
78 // Xem thông tin người tương ứng
79 showPersonInfo(position);
80 return true;
81 case R.id.menu_delete:
82 // Xóa người tương ứng
83 deletePerson(position);
84 return true;
85 default:
86 return super.onContextItemSelected(menuItem);
87 }
88 }

```
89     private void showPersonInfo(int position) {
90         // Hiển thị thông tin người tương ứng
91         Person selectedPerson = personList.get(position);
92         AlertDialog.Builder builder = new AlertDialog.Builder( context: this );
93         builder.setTitle(selectedPerson.getNamePerson())
94             .setMessage("Bạn vừa lựa chọn hiển thị thông tin của "+selectedPerson.getNamePerson())
95             .setPositiveButton( text: "OK", (dialog, which) -> dialog.dismiss() )
96             .create().show();
97     }
98     1 usage
99     private void deletePerson(int position) {
100        // Xóa người tương ứng
101        personList.remove(position);
102        personAdapter.notifyDataSetChanged();
103        Toast.makeText( context: this, text: "Đã xóa khỏi ListView", Toast.LENGTH_SHORT ).show();
}
```

Lab 6 - 3 – ActionBar

ActionBar hay Toolbar là một thành phần giao diện người dùng trong ứng dụng Android được sử dụng để hiển thị các tùy chọn điều khiển, tiêu đề và các hành động liên quan đến ngữ cảnh hiện tại của ứng dụng. ActionBar thường được đặt ở đầu màn hình và là một phần quan trọng của trải nghiệm người dùng trên Android.



Hình ảnh ActionBar (ToolBar)

1. Tạo ActionBar trong activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbarTop"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#009688"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent">

        <ImageView
            android:id="@+id/imageViewLogo"
            android:layout_width="48dp"
            android:layout_height="48dp"
            android:src="@drawable/logo"
            android:scaleType="centerInside"/>
    </androidx.appcompat.widget.Toolbar>
</androidx.constraintlayout.widget.ConstraintLayout>
```

2. Thay đổi thiết lập về ActionBar cho ứng dụng:

Mở res/values/themes/**themes.xml** để thay đổi giá trị **NoActionBar** thành giá trị khác, ví dụ:

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.Lab06ActionBar" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/purple_500</item>
        <item name="colorPrimaryVariant">@color/purple_700</item>
        <item name="colorOnPrimary">@color/white</item>
```

3. Thiết lập các thuộc tính cho ActionBar trong MainActivity:

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

    //Tim Toolbar từ layout
    Toolbar toolbarTop = findViewById(R.id.toolbarTop);
    //Đặt toolbar là actionBar
    setSupportActionBar(toolbarTop);

    //Hiển thị nút back
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    getSupportActionBar().setDisplayShowHomeEnabled(true);

    //Đặt tiêu đề
    getSupportActionBar().setTitle("Tiêu đề ứng dụng");
    getSupportActionBar().setSubTitle("Tiêu đề phụ");

    //Xử lý sự kiện khi nút back được ấn
    toolbarTop.setNavigationOnClickListener(view->getOnBackPressedDispatcher().onBackPressed());
}
```