

## Thực hành Buổi 1: LẬP TRÌNH GIAO DIỆN NGƯỜI DÙNG MỨC CAO (Có hướng dẫn)

### I. LÝ THUYẾT CẦN ÔN TẬP

- View và ViewGroup
- Các đối tượng view cơ bản
- Các loại Layout
- Xử lý sự kiện người dùng tương tác

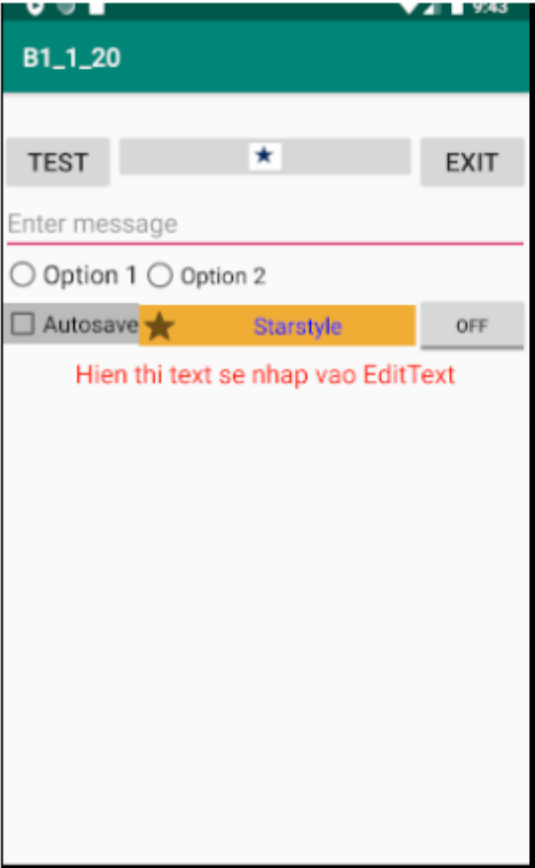
### II. THỰC HÀNH

#### Bài 1:

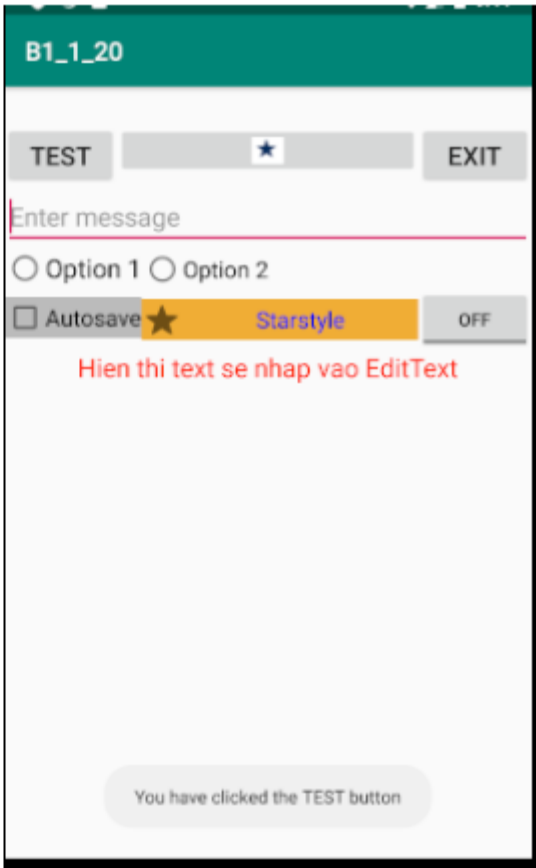
Mở project mới, xây dựng một ứng dụng với giao diện người dùng (UI) cấp cao và lập trình xử lý tương tác trực tiếp trên các view. UI gồm gồm: 2 Button tên là TEST và EXIT, 1 ImageButton (hình sao 4 cánh) nằm giữa 2 Button TEST và EXIT, 1 EditText, 2 RadioButton tên là Option1 và Option 2, 1 CheckBox kiểu thường tên là Autosave, 1 CheckBox kiểu star tên là Starstyle, 1 ToggleButton, 1 TextView (ban đầu có text là “**Hien thi text se nhap vao EditText**”), 2 ImageView (ban đầu chưa hiện ra) (Giao diện như Hình 1.1.a).

- Sử dụng Toast để hiển thị một thông báo khi ấn chọn các đối tượng sau: Khi ấn chọn Button TEST thì xuất hiện thông báo “You have clicked the TEST button” (Hình 1.1.b). Tương tự, khi ấn chọn ImageButton thì xuất hiện thông báo “You have clicked the IMAGEBUTTON button”, khi check chọn CheckBox Autosave thì xuất hiện thông báo “ChechBox is checked” ngược lại xoá check thì xuất hiện thông báo “ChechBox is unchecked”, khi check chọn Starstyle CheckBox thì ngôi sao sáng lên và xuất hiện thông báo “Star style ChechBox is checked” ngược lại xoá check thì xuất hiện thông báo “Star style ChechBox is unchecked”.
- Khi chọn RadioButton option 1 thì xuất hiện ImageView thứ nhất (Hình 1.1.c), khi chọn RadioButton option 2 thì xuất hiện ImageView thứ hai (Hình 1.1.d)
- Ban đầu ToggleButton ở trạng thái “Tắt”, khi ấn vào nó chuyển sang trạng thái “Bật” đồng thời dòng ký tự nhập vào EditText sẽ hiển thị thông qua TextView (chữ màu đỏ, phía trên ImgeView, hình 1.1.e).
- Khi chọn Button Exit thì ứng dụng kết thúc .

Buổi 1: Lập trình giao diện người dùng mức cao



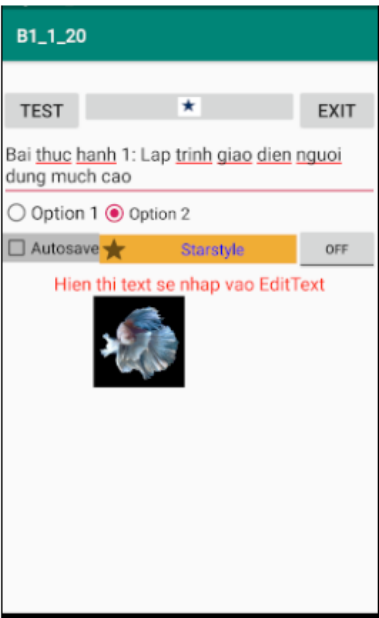
Hình 1.1.a



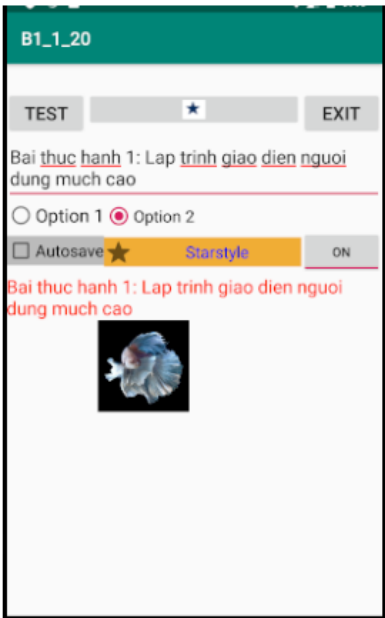
Hình 1.1.b



Hình 1.1.c



Hình 1.1.d



Hình 1.1.e

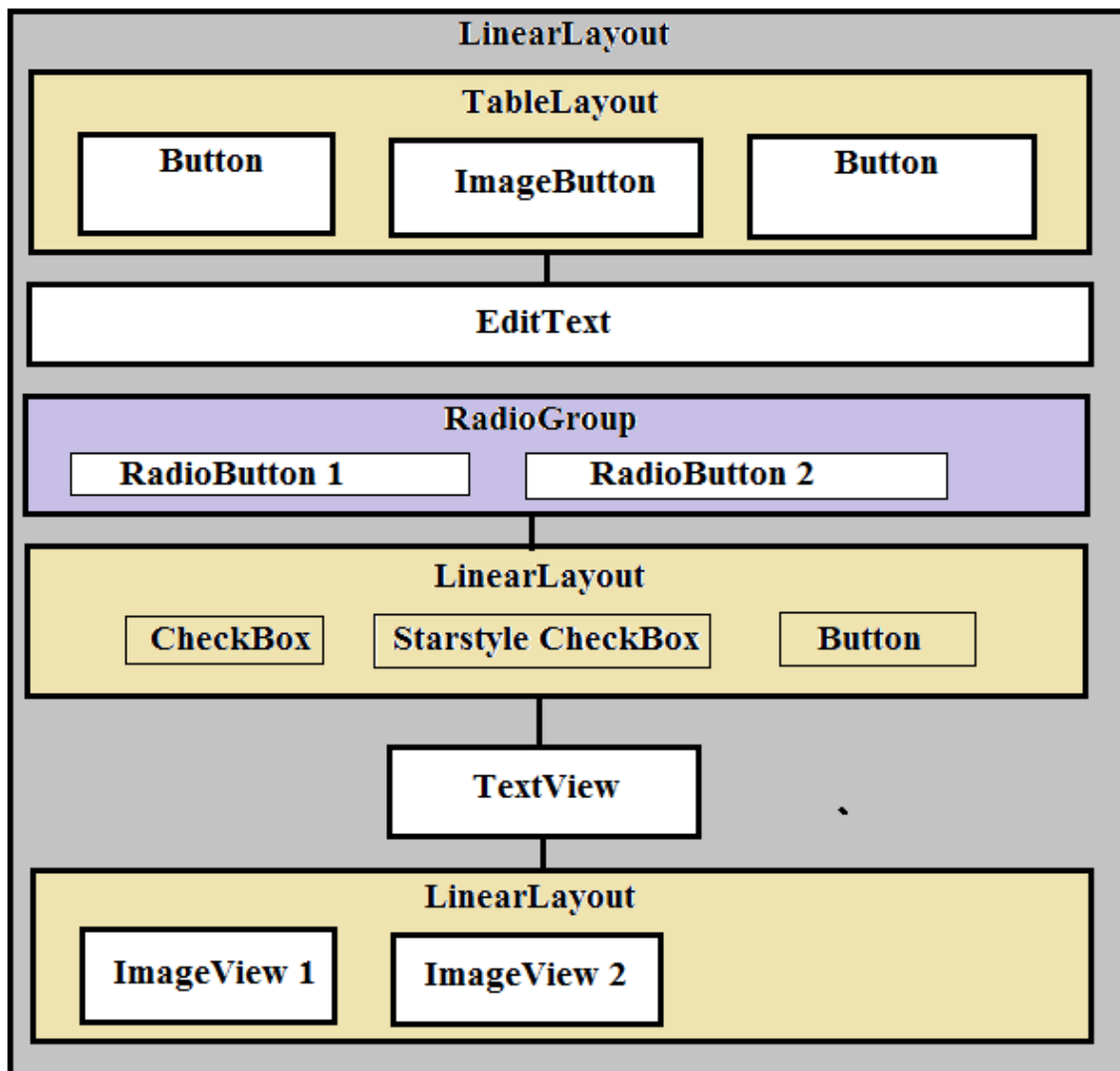
### Các bước thực hiện:

1. **Tạo android project mới:** Tạo project có tên là **B1\_1\_Tên\_mã số SV**, tên Activity chính là **MainActivity**, tên Layout là **activity\_main**.

2. **Xây dựng giao diện:**

- Sửa tập tin Layout **activity\_main.xml** nằm trong thư mục **res / layout** để có giao diện như **các hình 1.1**.
- Chép các tập tin ảnh cần thiết (sử dụng các tập tin ảnh **star\_be\_ti.png**, **ca\_lia\_thia\_1.png**, **ca\_lia\_thia\_2.png** trong thư mục **images\_png** trên trang web [elcit.ctu.edu.vn](http://elcit.ctu.edu.vn)) vào thư mục **res/drawable**.
- Bổ sung tài nguyên **string** vào tập tin **strings.xml** trong thư mục **values**.

Giao diện được thiết kế theo kiểu **LinearLayout**, chứa 3 layout con (1 **TableLayout** và 2 **LinearLayout**) và các View theo sơ đồ hình 1.2:



## Hình 1.2

### Nhận xét:

- Layout mẹ là loại `LinearLayout` với thuộc tính **`android:orientation="vertical"`**.
- `TableLayout` chỉ có 1 hàng và 3 cột (có 3 ô chứa 3 button).
- `RadioGroup` có thuộc tính **`android:orientation="horizontal"`**.
- Các layout con có thuộc tính **`android:orientation="horizontal"`**
- Muốn tạo Star style checkbox (Checkbox “ngôi sao”), ta khai báo thuộc tính **`style`** cho nó, cụ thể là **`style="?android:attr/starStyle"`**, với cú pháp là: **`?[package:]/[type:]/name`**.
- Tất cả các đối tượng đều có thuộc tính nhận dạng **`id`** với tên xác định. Các đối tượng này sẽ được nhận dạng trong mã nguồn Java bằng phương thức **`View.findViewById()`** hoặc **`Activity.findViewById()`**.

Mã nguồn của Layout hoàn chỉnh như sau (*Sinh viên không nên copy và past một cách máy móc, cần tìm hiểu chức năng, cú pháp, đối số, thuộc tính của các câu lệnh, các view; chịu khó nhập từng câu lệnh*):

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout 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"
    android:orientation="vertical"
    android:paddingTop="40dp"
    tools:context=".MainActivity">

    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TableRow>
            <Button
                android:id="@+id/btntest"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textSize="22sp"
                android:text="@string/btntest"/>
            <ImageButton
                android:id="@+id/btnImage1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:src="@drawable/star_1"
                android:contentDescription="@string/content1"/>
            <Button
                android:id="@+id/btnexit"
                android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:textSize="22sp"
        android:text="@string/btnexit"/>
    />
</TableRow>
</TableLayout>
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/edittext"
    android:hint="@string/ed_nam"/>
    <RadioGroup
        android:id="@+id/rg"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <RadioButton
            android:id="@+id/rb1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="20sp"
            android:text="@string/rb1"/>
        <RadioButton
            android:id="@+id/rb2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="18sp"
            android:text="@string/rb2"/>
    </RadioGroup>

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <CheckBox
            android:id="@+id/chkAutosave"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="18sp"
            android:background="#bbb"
            android:text="@string/chkautosave"/>
        <CheckBox
            android:id="@+id/star"
            style="?android:attr/starStyle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:layout_gravity="center"
            android:gravity="center"
            android:textSize="18sp"
            android:background="#F0AD36"
            android:textColor="#00f"
            android:text="@string/star"/>
        <ToggleButton
            android:id="@+id/toggle_1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>

```

```
</LinearLayout>

<TextView android:id="@+id/text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#ff1e00"
    android:textSize="20sp"
    android:layout_gravity="center"
    android:text="@string/txt1" />
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <ImageView
        android:id="@+id/image1"
        android:src="@drawable/ca_lia_thia_1"
        android:contentDescription="@string/content2"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:visibility="invisible"/>
    <ImageView
        android:id="@+id/image2"
        android:src="@drawable/ca_lia_thia_2"
        android:contentDescription="@string/content3"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:visibility="invisible"/>
</LinearLayout>
</LinearLayout>
```

### Chú thích:

- (1) Mặc định, Android sẽ tạo ra layout kiểu **ConstraintLayout** như sau. Để đơn giản, ta đã đổi lại kiểu **LinearLayout**.

```
<?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">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

- (2) Mặc dù thuộc tính Text ta có thể khai báo trực tiếp như sau:

**android:text="Hiển thị text sẽ nhập vào EditText"**

## Buổi 1: Lập trình giao diện người dùng mức cao


Tuy nhiên, ta sẽ có được nhiều thuận lợi sau này nếu dùng cách khai báo gián tiếp bằng 1 định danh như sau:

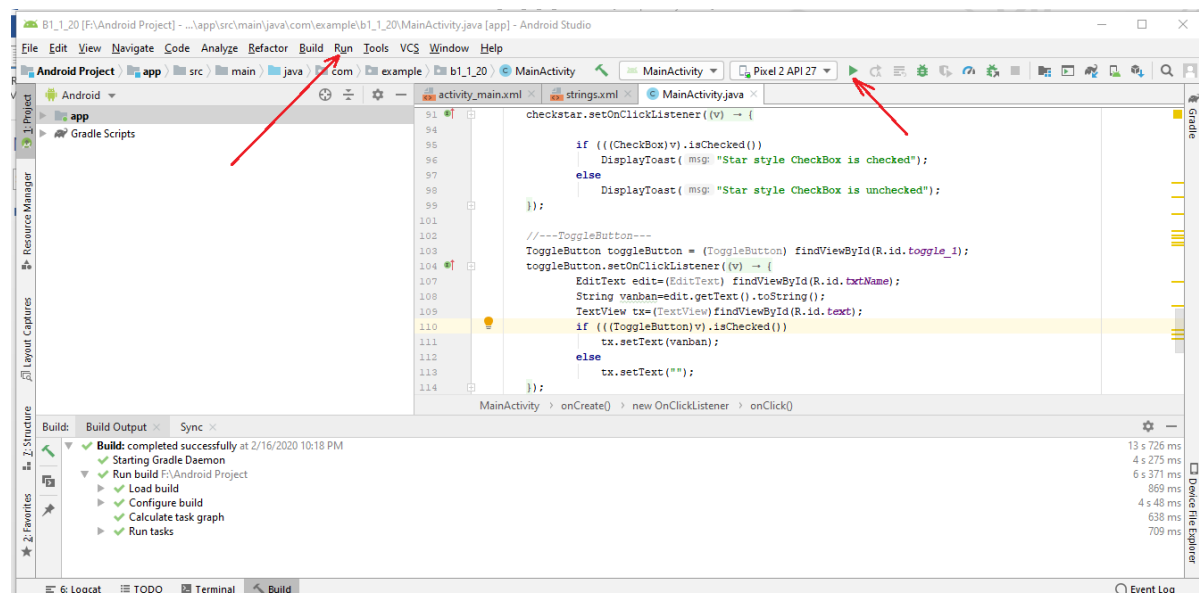
**android:text="@string/txt1"**

Sau đó, dùng định danh này để khai báo text trong tập tin **strings.xml**

### Tập tin strings.xml

```
<resources>
    <string name="app_name">Bai_Buoi1_Hai_B140112233</string>
    <string name="btnetest">TEST</string>
    <string name="btnexit">EXIT</string>
    <string name="content1">btnimage1</string>
    <string name="ed_nam">Enter message</string>
    <string name="txt1">Hien thi van ban duoc nhap vao EditText</string>
    <string name="content2">btnimage2</string>
    <string name="content3">btnimage3</string>
    <string name="star">StarStyle</string>
    <string name="chkautosave">AutoSave</string>
    <string name="rb1">Option 1</string>
    <string name="rb2">Option 2</string>
</resources>
```

3. **Kiểm tra giao diện và gỡ lỗi:** Trên thanh Menu chọn Run, hoặc Click chọn biểu tượng  (Hình 1.3), hoặc nhấn tổ hợp (Alt + Shift + F10) để chạy thử trên Android emulator. Xem giao diện người dùng gồm các view được hiển thị trong trình giả lập Android có đạt yêu cầu (vị trí, phong chữ, cỡ chữ, màu sắc) như giao diện đề bài không, nếu không thì chỉnh sửa lại tập tin activity\_main.xml.



Hình 1.3

## 4. Xử lý sự kiện

Sửa tập tin **MainActivity.java** thành như sau (SV cần đọc hiểu các câu lệnh):

```
package com.example.bai_buoi1_hai_b140112233;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;

public class MainActivity extends AppCompatActivity {


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button btnTest=(Button) findViewById(R.id.btnTest);
        btnTest.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                displayToastr("You have checked TEST Button");
            }
        });
        ImageButton btnImage=(ImageButton) findViewById(R.id.btnImage1);
        btnImage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                displayToastr("You have checked IMAGEButton");
            }
        });
        Button btnExit=(Button) findViewById(R.id.btnExit);
        btnExit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                displayToastr("You have checked EXIT Button");
                System.exit(0);
            }
        });
        ToggleButton btnToggle=(ToggleButton) findViewById(R.id.toggle_1);
        btnToggle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                EditText edit=(EditText) findViewById(R.id.edittext);
                String text=edit.getText().toString();
                TextView txt=(TextView) findViewById(R.id.text);
                if(((ToggleButton)v).isChecked())
                    txt.setText(text);
                else
                    txt.setText("");
            }
        });
    }
}
```



```
    }
    });
    RadioGroup radioGroup=(RadioGroup) findViewById(R.id.rg);
    radioGroup.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(RadioGroup group, int checkedId)
        {
            RadioButton rb1=(RadioButton) findViewById(R.id.rb1);
            RadioButton rb2=(RadioButton) findViewById(R.id.rb2);
            ImageView image1=(ImageView) findViewById(R.id.image1);
            ImageView image2=(ImageView) findViewById(R.id.image2);
            if (rb1.isChecked()){
                image1.setVisibility(ImageView.VISIBLE);
                image2.setVisibility(ImageView.INVISIBLE);
            }
            if (rb2.isChecked()){
                image2.setVisibility(ImageView.VISIBLE);
                image1.setVisibility(ImageView.INVISIBLE);
            }
        }
    });
    CheckBox chkAutoSau=(CheckBox) findViewById(R.id.chkAutosave);
    chkAutoSau.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if ((CheckBox)v.isChecked())
                displayToasr("AutoSave CheckBox have checked");
            else
                displayToasr("AutoSave CheckBox have unchecked");
        }
    });

    CheckBox chkStar=(CheckBox) findViewById(R.id.star);
    chkStar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if ((CheckBox)v.isChecked())
                displayToasr("StarStyle CheckBox have checked");
            else
                displayToasr("StarSyle CheckBox have unchecked");
        }
    });
}

private void displayToasr(String msg){
    Toast.makeText(getApplicationContext(),msg,Toast.LENGTH_LONG).show();
}
}
```

- **Chạy thử:** Trên thanh Menu chọn Run, hoặc Click chọn biểu tượng  (Hình 2.3), hoặc nhấn tổ hợp (Alt + Shift + F10) để chạy thử trên Android emulator.

### Giải thích:

Để xử lý các sự kiện khi người dùng chạm vào các view (widget), đầu tiên ta phải khai báo view được tương tác bằng phương thức **findViewById()** với đối số vào là id của view đó, phương thức này thuộc lớp MainActivity. Sau đó sử dụng phương thức **setOnClickListener()** để tạo ra đáp ứng. Ví dụ trường hợp button TEST, cú pháp của câu lệnh như sau:

```
Button btnTest = (Button) findViewById(R.id.btnTest);
```

Phương thức **setOnClickListener()** được sử dụng để lắng nghe sự kiện khi view được click và phương thức **onClick(View v)** tạo ra hành động với mã nguồn được ghi đè trong phương thức này:

```
btnTest.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        DisplayToast("You have clicked the Test button");  
    }  
});
```

Ở đây hành động là hiển thị một Toast với thông tin là “You have clicked the Test button”, phương thức **DisplayToast** được định nghĩa trong lớp MainActivity.

Để xác định trạng thái của CheckBox, ta phải định kiểu đối số của phương thức **onClick()** với một CheckBox và sau đó dùng phương thức **isChecked()** để xác định nếu nó được check:

```
CheckBox checkBox = (CheckBox) findViewById(R.id.chkAutosave);  
checkBox.setOnClickListener(new View.OnClickListener()  
{  
    public void onClick(View v) {  
        if (((CheckBox)v).isChecked())  
            DisplayToast("CheckBox is checked");  
        else  
            DisplayToast("CheckBox is unchecked");  
    }  
});
```

Đối với **RadioButton**, ta cần phải sử dụng phương thức **setOnCheckedChangeListener()** trên **RadioGroup** để đăng ký một tác động khi kiểm tra **RadioButton** trong nhóm này có sự thay đổi:

```
RadioGroup radioGroup = (RadioGroup) findViewById(R.id.rdbGp1);  
radioGroup.setOnCheckedChangeListener(new OnCheckedChangeListener()  
{  
    public void onCheckedChanged(RadioGroup group, int checkedId) {  
        RadioButton rb1 = (RadioButton) findViewById(R.id.rdb1);  
        if (rb1.isChecked()) {  
            DisplayToast("Option 1 checked!");  
        } else {  
            DisplayToast("Option 2 checked!");  
        }  
    }  
});
```

```
}}  
});
```

Khi một `RadioButton` được chọn, phương thức `onCheckedChanged()` được tác động. Trong phương thức này, ta xác định `RadioButtons` của nhóm và sau đó gọi phương thức `isChecked()` để xác định `RadioButton` này được chọn. Phương thức `onCheckedChanged()` còn chứa một đối số thứ hai, đối số này có chứa một định danh duy nhất của `RadioButton` được lựa chọn.

**ToggleButton** hoạt động giống như các `CheckBox`.

Như vậy, để xử lý các sự kiện trên các view, trước tiên ta phải lấy được một tham khảo tới view và sau đó đăng ký gọi lại để xử lý sự kiện này.

Có một cách khác để xử lý sự kiện view, ví dụ trường hợp `Button`, ta có thể thêm một thuộc tính `onClick` với nó trong tập tin `activity_main.xml`, như sau:

```
<Button  
    android:id="@+id/btnTest"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="@string/btnTest"  
    android:onClick="btnTest_clicked"/>
```

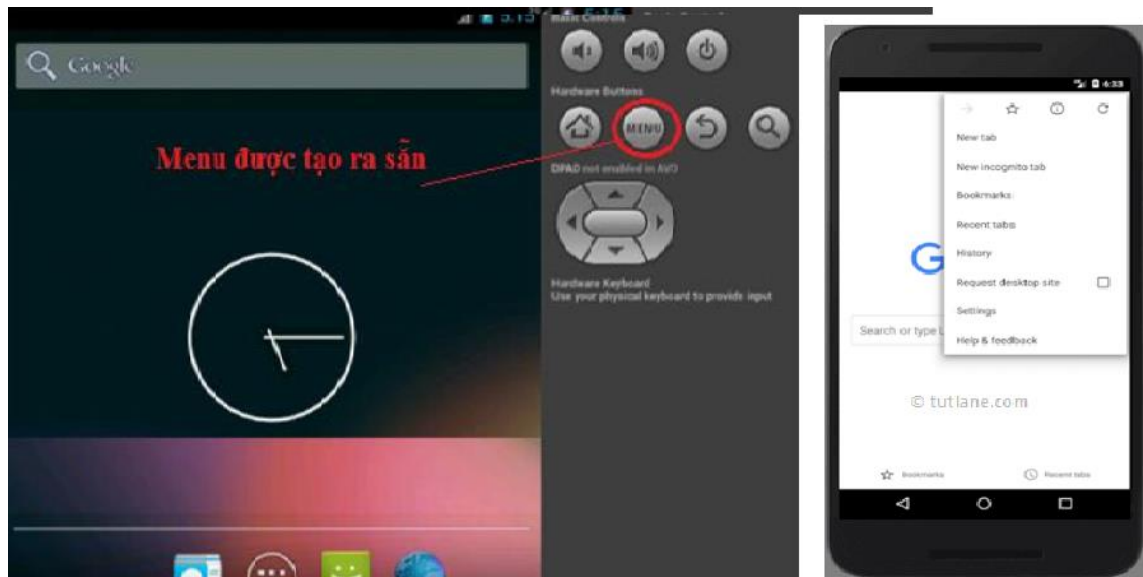
Các thuộc tính `onClick` xác định các sự kiện click của button. Các giá trị của thuộc tính này là tên của các xử lý sự kiện. Vì vậy, để xử lý sự kiện click của button, ta chỉ cần tạo ra một phương thức gọi là `btnTest_clicked`, như trong ví dụ sau (lưu ý rằng phương thức này phải có một tham số của loại `View`):

```
.....  
public void btnTest_clicked (View view) {  
    DisplayToast("You have clicked the Test button");  
}  
/** Called when the activity is first created. */  
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
    //...  
}  
private void DisplayToast(String msg)  
{  
    Toast.makeText(getBaseContext(), msg, Toast.LENGTH_LONG).show();  
}  
.....
```

(Chúng ta sẽ dùng cách này trong một bài thực hành khác).

## Bài 2: Menu

Menu là thành phần rất hữu dụng, đặc biệt khi ta muốn thêm vào các tùy chọn cho ứng dụng mà không muốn hiển thị thường xuyên trên giao diện chính. Từ Android 3.0 (API mức 11), các thiết bị Android không còn phải cung cấp một nút *Menu* chuyên dụng nữa (Nút Menu nằm sẵn trên giao diện chính). Với sự thay đổi này, các ứng dụng Android không còn phụ thuộc vào bảng điều khiển menu 6 mục truyền thống mà thay vào đó cung cấp một thanh hành động nhiều mục (item) thả xuống để người dùng chọn một trong các hành động đó. (Hình 1.4)



Thiết bị Android trước phiên bản 3.0  
có nút Menu chuyên dụng

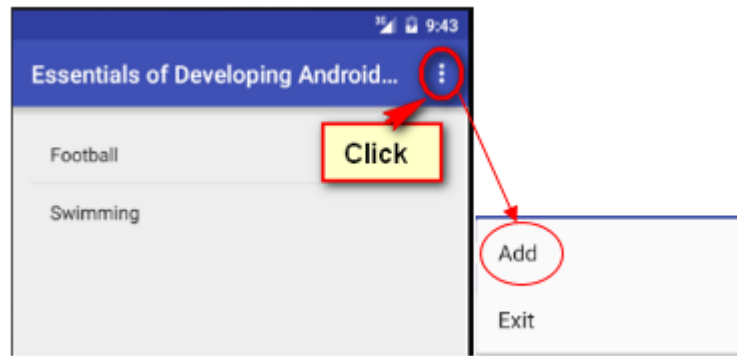
Thiết bị Android từ phiên bản 3.0  
về sau hoạt động với thanh Menu

Hình 1.4

Có 3 loại menu là:

### (1) Menu tùy chọn (Options menu) và thanh hành động

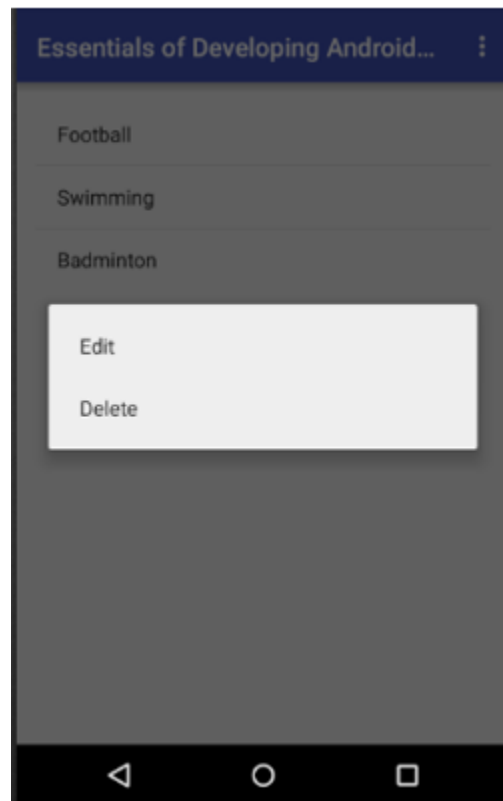
Menu tùy chọn là tập hợp các mục menu (menu items) cơ bản cho một activity. Đó là nơi ta nên đặt các *hành động có tác động chung tới ứng dụng*, chẳng hạn như "Tìm kiếm," "Soạn e-mail" và "Cài đặt." (Hình 1.5)



Hình 1.5. Vị trí menu là ở app bar của ứng dụng.

## (2) Menu ngữ cảnh (Context menu) và chế độ hành động theo ngữ cảnh

Menu ngữ cảnh là một menu nổi (floating menu), xuất hiện khi người dùng thực hiện ***nhấp giữ*** (long click) ***trên một phần tử*** (view). Nó cung cấp các ***hành động ảnh hưởng tới nội dung hoặc phạm vi ngữ cảnh*** được chọn. Chẳng hạn khi người dùng long click vào Listview, một vài thao tác được hiển thị để người dùng lựa chọn (Hình 1.6).



Hình 1.6. Ví dụ Context menu

### (3) Menu bật lên (Popup menu)

Menu bật lên sẽ hiển thị danh sách các mục trong một danh sách thả xuống được neo (anchored) vào một view được dùng để gọi menu (tương tự như menu ngữ cảnh), danh sách sẽ buông xuống nếu phía dưới View có đủ không gian, ngược lại nó sẽ dựng lên trên. Menu bật lên cung cấp các hành động liên quan tới nội dung cụ thể hoặc nhằm cung cấp các tùy chọn của một lệnh. Menu bật lên áp dụng cho các hành động mở rộng liên quan tới các vùng nội dung trong một activity.

Có 2 cách lập trình tạo ra menu:

(1) Xây dựng menu trong tài nguyên XML sau đó gọi nó trong một Activity.

(2) Xây dựng menu trực tiếp bằng mã nguồn Java trong một Activity.

Trong bài thực hành sau đây, chúng ta sẽ xây dựng menu theo cách thứ nhất. Cách thứ hai, thầy sẽ cung cấp tài liệu để sinh viên tham khảo thêm nếu cần.

#### Các bước xây dựng menu trong tài nguyên XML:

Đối với tất cả các loại menu, Android cung cấp một định dạng XML chuẩn để định nghĩa các mục menu. Thay vì xây dựng một menu trong mã của một activity, ta nên định nghĩa một menu và tất cả các mục của nó trong một tài nguyên menu XML.

Sử dụng một tài nguyên menu là một cách làm hay vì một vài lý do:

- Nó dễ trực quan hóa cấu trúc menu trong XML hơn.
- Nó tách riêng nội dung cho menu với mã hành vi của ứng dụng.
- Nó cho phép ta tạo các cấu hình menu phát sinh cho các phiên bản nền tảng, kích cỡ màn hình khác nhau và các cấu hình khác nhau bằng cách tận dụng app resources framework.

#### Bước 1: Định nghĩa menu trong XML.

Để định nghĩa menu, hãy tạo một tập tin XML bên trong thư mục `res/menu/` của ứng dụng (nếu chưa có thư mục menu thì tạo ra thư mục này) và xây dựng menu với các phần tử sau:

`<menu>`

Định nghĩa một [Menu](#), đó là một khung chứa các mục menu. Phần tử `<menu>` phải là một nút gốc cho tập tin và có thể giữ một hoặc nhiều phần tử `<item>` và `<group>`.

`<item>`

Tạo một [MenuItem](#), nó biểu diễn một mục đơn (một hoạt động) trong một menu. Phần tử này có thể chứa một phần tử `<menu>` được lồng vào nhau để tạo một menu con.

`<group>`

Một khung ản chứa các item. Nó cho phép ta phân loại các cùng thuộc tính, như trạng thái hoạt động và khả năng hiển thị.

Sau đây là một menu ví dụ có tên là `game_menu.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/new_game"
        android:icon="@drawable/ic_new_game"
        android:title="@string/new_game"
        android:showAsAction="ifRoom"/>
  <item android:id="@+id/help"
        android:icon="@drawable/ic_help"
        android:title="@string/help" />
</menu>
```

Một `<item>` có thể được định nghĩa bởi các thuộc tính. Trong menu trên, ta đã khai báo các thuộc tính sau:

`android:id`

Một ID tài nguyên duy nhất đối với một item, nó cho phép ứng dụng có thể nhận ra item đó khi người dùng chọn nó.

`android:icon`

Một tham chiếu tới một file ảnh trong thư mục `drawable` được dùng làm biểu tượng của item.

`android:title`

Một tham chiếu tới một string để dùng làm tiêu đề của item.

`android:showAsAction`

Quy định thời điểm và cách thức item này xuất hiện như một item hành động trong thanh hành động (app bar).

Đây là những thuộc tính quan trọng nhất ta nên sử dụng, nhưng còn nhiều thuộc tính sẵn có khác. Để biết thông tin về tất cả thuộc tính được hỗ trợ, hãy tham khảo tại url: <https://developer.android.com/guide/topics/resources/menu-resource.html>.

Ta có thể thêm một menu con vào một item trong bất kỳ menu (ngoại trừ menu con). Các menu con thường hữu ích khi ứng dụng có nhiều chức năng mà có thể được tổ chức thành các chủ đề, như các mục trong thanh menu của một ứng dụng PC (Tập tin, Chỉnh sửa, Dạng xem, v.v.). Ví dụ:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/file"
        android:title="@string/file" >
```

```
<!-- "file" submenu -->
<menu>
  <item android:id="@+id/create_new"
        android:title="@string/create_new" />
  <item android:id="@+id/open"
        android:title="@string/open" />
</menu>
</item>
</menu>
```

## Bước 2: Gọi Menu và xử lý sự kiện người dùng trong Activity

Để sử dụng menu trong activity, ta cần khai báo tài nguyên menu (chuyển tài nguyên XML thành một đối tượng có thể lập trình) với câu lệnh `MenuInflater.inflate()` trong mã nguồn của activity.

Khi người dùng chọn một mục, hệ thống sẽ gọi hành động tương ứng bằng phương thức `onOptionsItemSelected()` (với Option menu hoặc `onContextItemSelected` với Context menu). Phương thức này thông qua `MenuItem` được chọn. Ta có thể nhận biết item bằng phương thức `getItemId()`, nó trả về ID duy nhất cho menu item (được định nghĩa bởi thuộc tính `android:id` trong tài nguyên menu hoặc bằng một số nguyên được cấp cho phương thức `add()`). Ta có thể khớp ID này với các item đã biết để thực hiện hành động phù hợp. Ví dụ:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
        case R.id.new_game:
            newGame();
            return true;
        case R.id.help:
            showHelp();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

Nếu xử lý thành công một mục menu thì trả về `true`. Nếu không xử lý được mục menu, thì nên thực thi superclass của `onOptionsItemSelected()` (mặc định trả về `false`).

## THỰC HÀNH

Lập trình tạo ra UI gồm (Hình 1.7):

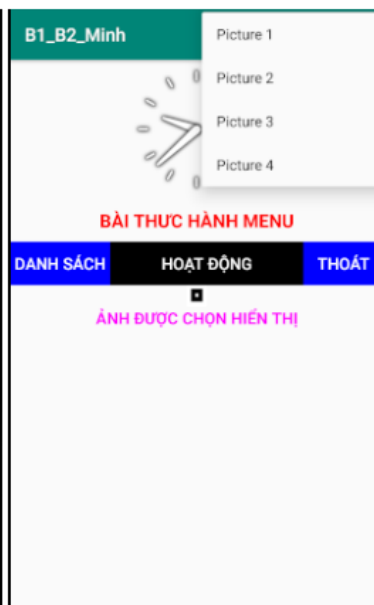


- Một đồng hồ analog hiển thị thời gian của hệ thống;
- Một TextView với text là “BÀI THỰC HÀNH MENU”;
- Một Button dùng cho Context menu tên là “DANH SÁCH” và một Button dùng cho Popup menu tên là “HOẠT ĐỘNG”. Xây dựng 3 loại menu gồm:
  - Options menu với 4 mục (item), khi ấn chọn một mục sẽ thực hiện một hoạt động cụ thể (tùy người lập trình chọn lựa, trong bài này ta cho hiển thị các ảnh khác nhau);
  - Context menu gồm 1 item và 1 menu con có 4 item trong đó, ta sử dụng đối tượng Toast để hiển thị một danh sách tương ứng khi 1 item được chọn.
  - Popup menu gồm 4 item, khi người dùng chọn 1 item thì hành động là thay text “HÀNH ĐỘNG” trong Button thành các terxt MENU COPY, MENU PAST, MENU SAVE hoặc MENU DELETE tùy theo item được click chọn.
- Một Button tên là “THOÁT”, khi nhấn vào ứng dụng sẽ kết thúc.
- Một ImageView để hiển thị ảnh tương ứng khi 1 mục của OptionMenu được chọn (Ảnh khi khởi động là 1 dấu vuông);
- Một TextView hiển thị text “ẢNH ĐƯỢC CHỌN HIỂN THỊ”.

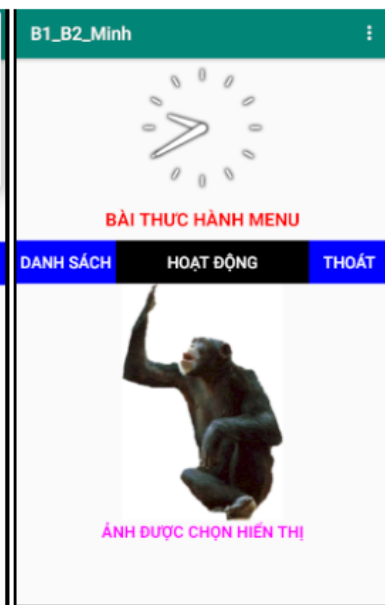
Chú ý: Giao diện phải đúng vị trí, màu sắc, cỡ chữ,...



Hình 7.a. Giao diện khởi động

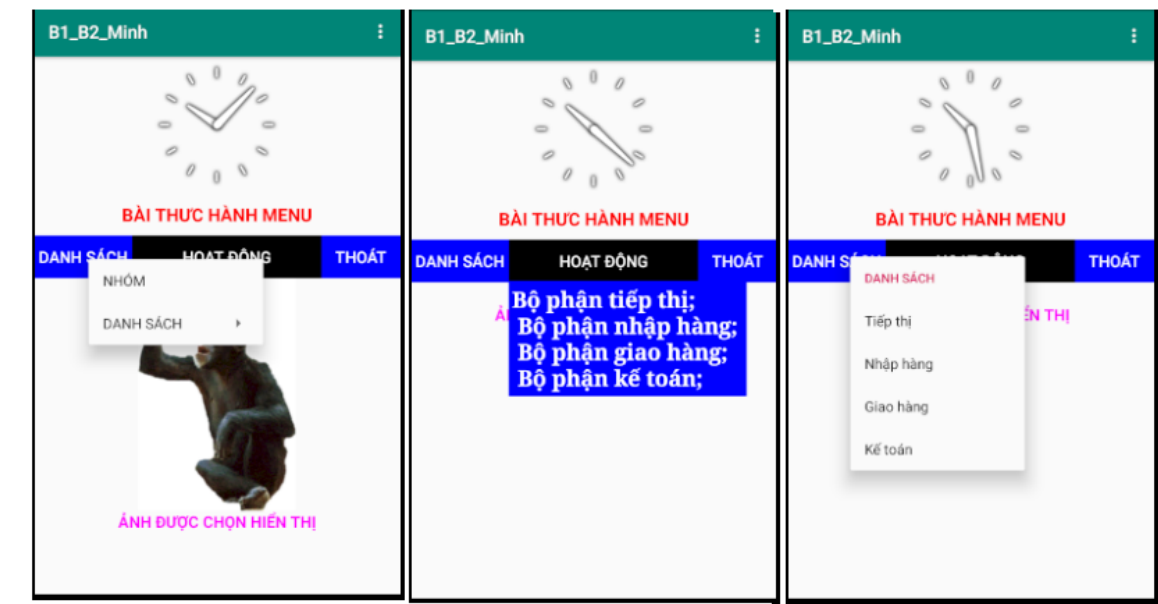


Hình 7.b. Option menu được chọn



Hình 7.c. Item 1 của Option menu được chọn, xuất hiện ảnh.

Buổi 1: Lập trình giao diện người dùng mức cao



Hình 1.7.d. Context menu được chọn

Hình 1.7.e. Item NHÓM của Context menu được chọn

Hình 1.7.f. Menu con DANH SÁCH của Context menu được chọn

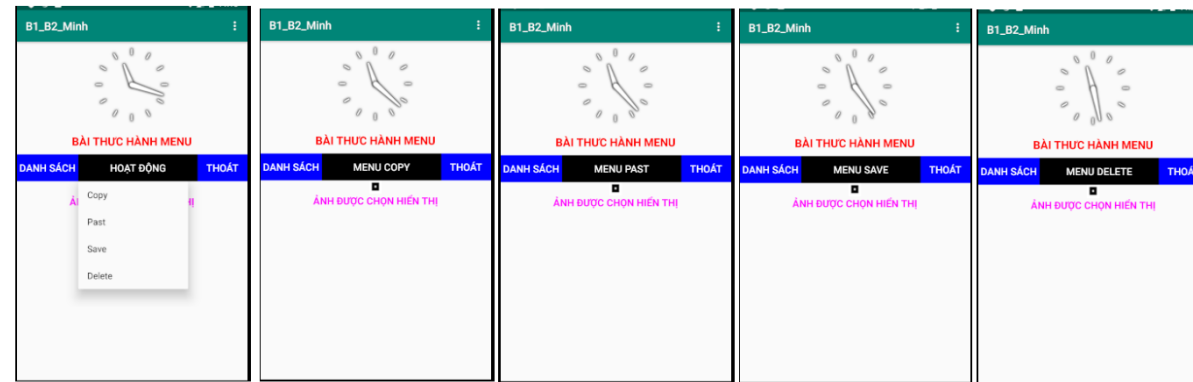


Hình 1.7.g. Item Tiếp thị trong menu con DANH SÁCH được chọn.

Hình 1.7.h. Item Nhập hàng trong menu con DANH SÁCH được chọn

Hình 1.7.i. Item Giao hàng trong menu con DANH SÁCH được chọn

Hình 1.7.j. Item Kế toán trong menu con DANH SÁCH được chọn



Hình 1.7.k. Popup menu được chọn gồm 4 item: Copy, Past, Save, Delete

Hình 1.7.m. Item Copy trong Popup menu được chọn

Hình 1.7.n. Item Past trong Popup menu được chọn

Hình 1.7.l. Item Save trong Popup menu được chọn

Hình 1.7. q. Item Delete trong Popup menu được chọn

Hình 1.7

### Các bước thực hiện:

1. **Tạo android project mới:** Tạo project có tên là **B1\_1\_Tên\_mã\_số\_SV**, tên Activity chính là **MainActivity**, tên Layout là **activity\_main**.
2. **Xây dựng giao diện chính:**
  - Chép các tập tin ảnh cần thiết (ví dụ này đã sử dụng các tập tin ảnh **chimp.png**, **crowned\_crane.png**, **dolphin.png**, **drake.png**, **sao\_blue.png**, **dao\_red.png**, **sao\_vang.png**, **star\_du\_mau.png**, **stop.png** trong thư mục **images\_png** trên trang web **elcit.ctu.edu.vn**) vào thư mục **res/drawable**.
  - Sửa tập tin Layout **activity\_main.xml** nằm trong thư mục **res / layout** để có giao diện như các **Hình 1.7.a**. Trong ví dụ này, không dùng tài nguyên **string** mà khai báo trực tiếp các text trong tập tin **activity\_main.xml**. (Xem tập tin mã nguồn XML kèm theo, cần tìm hiểu để biết các thuộc tính).

### Ghi chú:

#### Tạo đồng hồ tương tự hoặc đồng hồ số (AnalogClock / DigitalClock Views)

View AnalogClock hiển thị một đồng hồ analog với hai kim - một phút và một giờ. Trong khi đó, view DigitalClock hiển thị thời gian bằng kỹ thuật số. Cả hai view chỉ hiển thị thời gian hệ thống, và không cho phép bạn hiển thị thời gian ở một nơi cụ thể (chẳng hạn như thời điểm hiện tại trong múi giờ khác). Do đó, nếu ta muốn hiển thị thời gian cho một khu vực cụ thể, ta phải xây dựng view riêng của mình.

Việc sử dụng các view AnalogClock và DigitalClock là khá dễ dàng, ta khai báo chúng một cách đơn giản trong tập tin Layout như sau:

Đồng hồ tương tự:

```
<AnalogClock
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

Đồng hồ kỹ thuật số:

```
<DigitalClock
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

- Chạy thử, chỉnh sửa Layout cho đạt yêu cầu của đề bài.

Mã nguồn XML:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <AnalogClock
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:padding="10dp"
        android:textColor="#f00"
        android:textSize="20sp"
        android:textStyle="bold"
        android:text="@string/txt" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <Button
        android:id="@+id/btntmenu"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:padding="5dp"
        android:textSize="18sp"
        android:textColor="#fff"
        android:background="#00f"
        android:text="DANH SÁCH" />
    <Button
        android:id="@+id/btnpopup"
        android:layout_width="0sp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="18sp"
        android:textColor="#fff"
        android:background="#000"
        android:layout_gravity="center"
        android:text="HOẠT ĐỘNG"
        android:clickable="true" />
    <Button
        android:id="@+id/btnthoat"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textSize="18sp"
        android:textColor="#fff"
        android:background="#00f"
        android:text="THOÁT" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <ImageView
        android:id="@+id/img"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:contentDescription="image"
        android:src="@drawable/stop"
        android:background="#fff" />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center"

```

```
        android:text="ẢNH ĐƯỢC CHỌN HIỂN THỊ"  
        android:textColor="#f0f"  
        android:textStyle="bold"  
        android:textSize="18sp">  
    </TextView>  
</LinearLayout>  
</LinearLayout>
```

3. **Xử lý nút THOÁT:** Mở MainActivity để lập trình xử lý tương tác của người dùng lên nút THOÁT. Khai báo phần tử Button cho nút THOÁT, button này có id là btnthoat (nên đặt trước hàm onCreate() rồi dùng hàm *findViewById()* để tải đối tượng Button này về, nhân tiện khai báo luôn các button cho Option menu và Popup menu). Dùng hàm *setOnClickListener()* để tạo tương tác cho nút THOÁT. Mã nguồn như sau (*dĩ nhiên phải thêm vào các lệnh import tương ứng*):

```
public class MainActivity extends AppCompatActivity {  
    Button btnmenu, btnpopup, thoat; // Khai báo các đối tượng loại Button  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        btnmenu = (Button) findViewById(R.id.btntmenu);  
        btnmenu.setOnClickListener(this);  
        btnpopup = (Button) findViewById(R.id.btnpopup);  
  
        thoat = (Button) findViewById(R.id.btnthoat);  
        thoat.setOnClickListener(new View.OnClickListener() {  
  
            @Override  
            public void onClick(View v) {  
                System.exit(0);  
            }  
        });  
    }  
}
```

#### 4. Tạo Option menu:

- **Tạo thư mục menu là thư mục con của thư mục res:** Nhấp chuột phải vào thư mục res → Chọn New → Chọn Android Resource Directory → Trong cửa sổ New Resource Directory, điền vào mục Directory name tên là menu (chữ thường), trong mục Resource type ta chọn menu, trong Source set ta chọn main.
- **Tạo tập tin layout cho Option menu:** Nhấp chuột phải vào thư mục menu → Trên cửa sổ New Resource File, tại trường File name ta đặt tên file, trong ví dụ này chọn tên là menu\_tuychon, các trường khác giữ giá trị mặc định. Mở tập tin này và thêm vào các item và thuộc tính của nó. Ví dụ như sau:

```
<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:id="@+id/item1"  
        android:icon="@drawable/sao_blue"  
        android:title="Picture 1"/>  
    <item android:id="@+id/item2"  
        android:icon="@drawable/star_du_mau"
```

```
        android:title="Picture 2"/>
    <item android:id="@+id/item3"
        android:icon="@drawable/sao_red"
        android:title="Picture 3"/>
    <item android:id="@+id/item4"
        android:icon="@drawable/sao_vang"
        android:title="Picture 4"/>
</menu>
```

- **Lập trình tạo Option menu và xử lý tương tác của người dùng trên MainActivity:** Mở tập tin MainActivity.java thêm vào các phương thức tạo Option menu và Click chọn item (nằm ngoài phương thức onCreate()) như sau:

```
// Tạo ra Option menu
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_tuychon, menu);
    return true;
}

// Xử lý thao tác click chọn item
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Tải ImageView đã khai báo trong Layout để đặt hình vào.
    ImageView hình=(ImageView) findViewById(R.id.img);
    // Handle item selection
    switch (item.getItemId()) {
        case R.id.item1:
            // Đặt lại ảnh mới với file chimp.png trong res/drawable
            hình.setImageResource(R.drawable.chimp);
            return true;
        case R.id.item2:
            hình.setImageResource(R.drawable.crowned_crane);
            return true;
        case R.id.item3:
            hình.setImageResource(R.drawable.dolphin);
            return true;
        case R.id.item4:
            hình.setImageResource(R.drawable.drake);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

- Chạy thử và chỉnh sửa nếu cần.

## 5. Tạo Context menu:

- **Tạo tập tin layout cho Context menu:** Nhấp chuột phải vào thư mục menu → Trên cửa sổ New Resource File, tại trường File name ta đặt tên file, trong ví dụ này chọn tên là menu\_ngucanh, các trường khác giữ giá trị mặc định. Mở tập tin này và thêm vào các item và thuộc tính của nó. Ví dụ như sau (menu này có 2 item, trong đó item2 là 1 menu con, trong menu con này có 4 item):

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/item1"
        android:icon="@drawable/stop"
        android:title="NHÓM" />
    <item android:id="@+id/item2"
```

```
        android:icon="@drawable/forward"
        android:title="DANH SÁCH">
        <!-- "file" submenu -->
        <menu>
            <item android:id="@+id/item2_1"
                android:title="Tiếp thị" />
            <item android:id="@+id/item2_2"
                android:title="Nhập hàng" />
            <item android:id="@+id/item2_3"
                android:title="Giao hàng" />
            <item android:id="@+id/item2_4"
                android:title="Kế toán" />
        </menu>
    </item>
</menu>
```

- **Lập trình tạo Context menu và xử lý tương tác của người dùng trên MainActivity:** Mở tập tin MainActivity.java thêm vào các phương thức tạo Context menu và Click chọn item (nằm ngoài phương thức onCreate()) như sau:

```
// Tạo ra Context menu
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                                ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_ngucanh, menu);
}
// Xử lý click chọn item trên context menu
@Override
public boolean onContextItemSelected(MenuItem item) {
    AdapterView.AdapterContextMenuInfo info =
    (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
    switch (item.getItemId()) {
        case R.id.item1:
            Toast toast=new Toast(this); // Tạo 1 đối tượng Toast
            toast.setGravity(Gravity.CENTER,100,100); // Đặt vị trí của Toast.
            TextView tv=new TextView(this); // Khai báo 1 TextView.
            tv.setBackgroundColor(Color.BLUE); // Đặt màu nền.
            tv.setTextColor(Color.WHITE); // Đặt màu Text.
            tv.setTextSize(25); // Đặt cỡ chữ.
            Typeface t=Typeface.create("serif",Typeface.BOLD); //Chọn kiểu chữ.
            tv.setTypeface(t); // Đặt kiểu chữ.
            tv.setPadding(10,10,10,10); // Đặt vùng đệm
            tv.setText("Bộ phận tiếp thị; \n Bộ phận nhập hàng; \n Bộ phận
giao hàng; \n Bộ phận kế toán;");
            toast.setView(tv); // Đặt text cho toast.
            toast.setDuration(Toast.LENGTH_LONG); // Đặt thời gian hiển thị.
            toast.show(); // Hiển thị toast.
            return true;
        case R.id.item2_1:
            Toast toast1=new Toast(this);
            toast1.setGravity(Gravity.CENTER,100,100);
            TextView tv1=new TextView(this);
            tv1.setBackgroundColor(Color.BLACK);
            tv1.setTextColor(Color.YELLOW);
            tv1.setTextSize(25);
            Typeface t1=Typeface.create("serif",Typeface.BOLD);
            tv1.setTypeface(t1);
            tv1.setPadding(10,10,10,10);
            tv1.setText("1. Trần Ngọc Thơ;\n 2. Nguyễn Văn Toàn;\n 3. Lê Thị
Hoa;");
            toast1.setView(tv1);
```



```
        toast1.setDuration	Toast.LENGTH_LONG;
        toast1.show();
        return true;
    case R.id.item2_2:
        Toast toast2=new Toast(this);
        toast2.setGravity(Gravity.CENTER,100,100);
        TextView tv2=new TextView(this);
        tv2.setBackgroundColor(Color.BLACK);
        tv2.setTextColor(Color.YELLOW);
        tv2.setTextSize(25);
        Typeface t2=Typeface.create("serif",Typeface.BOLD);
        tv2.setTypeface(t2);
        tv2.setPadding(10,10,10,10);
        tv2.setText("1. Nguyễn văn A; \n 2. Lâm Văn Mới;");
        toast2.setView(tv2);
        toast2.setDuration	Toast.LENGTH_LONG;
        toast2.show();
        return true;
    case R.id.item2_3:
        Toast toast3=new Toast(this);
        toast3.setGravity(Gravity.CENTER,100,100);
        TextView tv3=new TextView(this);
        tv3.setBackgroundColor(Color.BLACK);
        tv3.setTextColor(Color.YELLOW);
        tv3.setTextSize(25);
        Typeface t3=Typeface.create("serif",Typeface.BOLD);
        tv3.setTypeface(t3);
        tv3.setPadding(10,10,10,10);
        tv3.setText("1. Trần Văn Bảy; \n 2. Nguyễn Văn Tâm; \n 3. Châu
Thành Quý;");
        toast3.setView(tv3);
        toast3.setDuration	Toast.LENGTH_LONG;
        toast3.show();
        return true;
    case R.id.item2_4:
        Toast toast4=new Toast(this);
        toast4.setGravity(Gravity.CENTER,100,100);
        TextView tv4=new TextView(this);
        tv4.setBackgroundColor(Color.BLACK);
        tv4.setTextColor(Color.YELLOW);
        tv4.setTextSize(25);
        Typeface t4=Typeface.create("serif",Typeface.BOLD);
        tv4.setTypeface(t4);
        tv4.setPadding(10,10,10,10);
        tv4.setText("1. Lê Thị Lý; \n Trần Ngọc Mai.");
        toast4.setView(tv4);
        toast4.setDuration	Toast.LENGTH_LONG;
        toast4.show();
        return true;
    default:
        return super.onContextItemSelected(item);
}
```

Chú ý: Thay vì dùng Toast với các thuộc tính mặc định bằng 1 lệnh đơn như sau:

```
Toast.makeText(this, "Bộ phận tiếp thị; \n Bộ phận nhập hàng; \n Bộ phận giao hàng; \n Bộ phận kế toán; ", Toast.LENGTH_LONG).show();
```

Ta đã khai báo Toast với nhiều lệnh để có được cỡ chữ, màu nền, màu chữ, kiểu chữ như ý muốn.

- Chạy thử và chỉnh sửa nếu cần.



## 6. Tạo Popup menu:

- **Tạo tập tin layout cho Context menu:** Nhấp chuột phải vào thư mục menu → Trên cửa sổ New Resource File, tại trường File name ta đặt tên file, trong ví dụ này chọn tên là **menu\_popup**, các trường khác giữ giá trị mặc định. Mở tập tin này và thêm vào các item và thuộc tính của nó. Ví dụ như sau (menu này có 4 item):

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/item3_1"
        android:title="Copy" />
    <item android:id="@+id/item3_2"
        android:title="Past" />
    <item android:id="@+id/item3_3"
        android:title="Save" />
    <item android:id="@+id/item3_4"
        android:title="Delete" />
</menu>
```

- **Chọn 1 Button để làm điểm neo:** Ta đã chuẩn bị sẵn 1 Button tên là HOẠT ĐỘNG và id là btnpopup trong Layout chính và đã khai báo trong MainActivity với tên là btnpopup.
- **Lập trình tạo Context menu và xử lý tương tác của người dùng trên MainActivity:** Mở tập tin MainActivity.java thêm vào các phương thức tạo Popup menu và Click chọn item (nằm ngoài phương thức onCreate()). Việc tạo Popup menu có điểm khác với Option menu và Context menu. Các bước tiếp theo như sau:
  - **Tạo hàm xây dựng Popup menu, xử lý click chọn item và mở (show) menu:** Trong ví dụ này ta đặt tên là showPopupMenu (nằm ngoài phương thức onCreate()), mã nguồn như sau:

```
// Tạo Popup menu
private void showPopupMenu() {
    // Tạo 1 popupmenu.
    PopupMenu popupMenu=new PopupMenu(this,btnpopup);
    //Lấy giao diện menu trong resoucse.

    popupMenu.getMenuInflater().inflate(R.menu.menu_popup,popupMenu.getMenu());
    //Lập trình hành động khi chọn item.
    popupMenu.setOnMenuItemClickListener(new
    PopupMenu.OnMenuItemClickListener() {
        @Override
        public boolean onMenuItemClick(MenuItem item) {
            switch (item.getItemId()) {
                case R.id.item3_1: btnpopup.setText("MENU COPY");
                break;
            }
            switch (item.getItemId()) {
                case R.id.item3_2: btnpopup.setText("MENU PAST");
                break;
            }
            switch (item.getItemId()) {
                case R.id.item3_3: btnpopup.setText("MENU SAVE");
                break;
            }
            switch (item.getItemId()) {
                case R.id.item3_4: btnpopup.setText("MENU DELETE");
            }
        }
    });
}
```

```
                break;
            }
            return false;
        }
    });
    popupMenu.show();
}
```

– **Xây dựng hàm click chọn item trong hàm onCreate():** Trở về phương thức onCreate ta xây dựng phương thức click chọn item, trong ví dụ này ta cũng dùng hàm setOnClickListener để gọi hàm showPopupMenu, mã nguồn như sau:

```
btnpopup.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showPopupMenu();
    }
});
```

- **Chạy thử và điều chỉnh nếu cần.**

Cuối cùng mã nguồn đầy đủ của MainActivity như sau (*Chú ý: Sinh viên phải tìm hiểu các câu lệnh và cách vận dụng để có thể áp dụng sáng tạo sau này, không nên copy và past một cách máy móc*):

```
package com.example.b1_b2_minh;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.ContextMenu;
import android.view.Gravity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.ImageView;
import android.widget.PopupMenu;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    Button btnmenu, btnpopup, thoat; // Khai báo các đối tượng loại Button
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnmenu=(Button) findViewById(R.id.btntmenu);
        btnmenu.onCreateContextMenuListener(this);
        btnpopup=(Button) findViewById(R.id.btnpopup);

        thoat=(Button) findViewById(R.id.bntthoat);
        thoat.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                System.exit(0);
            }
        });
    }
}
```

```

    }
    });
    btnpopup.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showPopupMenu();
        }
    });
}

// Tạo ra Option menu
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_tuychon, menu);
    return true;
}

// Xử lý thao tác click chọn item
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Tải ImageView đã khai báo trong Layout để đặt hình vào
    ImageView hình=(ImageView) findViewById(R.id.img);
    // Handle item selection
    switch (item.getItemId()) {
        case R.id.item1:
            hình.setImageResource(R.drawable.chimp);
            return true;
        case R.id.item2:
            hình.setImageResource(R.drawable.crowned_crane);
            return true;
        case R.id.item3:
            hình.setImageResource(R.drawable.dolphin);
            return true;
        case R.id.item4:
            hình.setImageResource(R.drawable.drake);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

// Tạo ra Context menu
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                                ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_ngucanh, menu);
}

// Xử lý click chọn item trên context menu
@Override
public boolean onContextItemSelected(MenuItem item) {
    AdapterView.AdapterContextMenuInfo info =
    (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
    switch (item.getItemId()) {
        case R.id.item1:
            //Toast.makeText(this, "Bộ phận tiếp thị; \n Bộ phận nhập hàng; \n
Bộ phận giao hàng; \n Bộ phận kế toán;", Toast.LENGTH_LONG).show(); ;
            Toast toast=new Toast(this); // Tạo 1 đối tượng Toast
            toast.setGravity(Gravity.CENTER,100,100); // Đặt vị trí của Toast.
            TextView tv=new TextView(this); // Khai báo 1 TextView.
            tv.setBackgroundColor(Color.BLUE); // Đặt màu nền.
            tv.setTextColor(Color.WHITE); // Đặt màu Text.
            tv.setTextSize(25); // Đặt cỡ chữ.
            Typeface t=Typeface.create("serif",Typeface.BOLD); // Chọn kiểu

```

```

chữ.
        tv.setTypeface(t); // Đặt kiểu chữ.
        tv.setPadding(10,10,10,10); // Đặt vùng đệm
        tv.setText("Bộ phận tiếp thị; \n Bộ phận nhập hàng; \n Bộ phận
giao hàng; \n Bộ phận kế toán;");
        toast.setView(tv); // Đặt text cho toast.
        toast.setDuration	Toast.LENGTH_LONG; // Đặt thời gian hiển thị.
        toast.show(); // Hiển thị toast.
        return true;
    case R.id.item2_1:
        Toast toast1=new Toast(this);
        toast1.setGravity(Gravity.CENTER,100,100);
        TextView tv1=new TextView(this);
        tv1.setBackgroundColor(Color.BLACK);
        tv1.setTextColor(Color.YELLOW);
        tv1.setTextSize(25);
        Typeface t1=Typeface.create("serif",Typeface.BOLD);
        tv1.setTypeface(t1);
        tv1.setPadding(10,10,10,10);
        tv1.setText("1. Trần Ngọc Thơ; \n 2. Nguyễn Văn Toàn; \n 3. Lê Thị
Hoa;");

        toast1.setView(tv1);
        toast1.setDuration	Toast.LENGTH_LONG;
        toast1.show();
        return true;
    case R.id.item2_2:
        //Toast.makeText(this, "1. Nguyễn văn A; \n 2. Lâm Văn Mới;",
Toast.LENGTH_LONG).show(); ;
        Toast toast2=new Toast(this);
        toast2.setGravity(Gravity.CENTER,100,100);
        TextView tv2=new TextView(this);
        tv2.setBackgroundColor(Color.BLACK);
        tv2.setTextColor(Color.YELLOW);
        tv2.setTextSize(25);
        Typeface t2=Typeface.create("serif",Typeface.BOLD);
        tv2.setTypeface(t2);
        tv2.setPadding(10,10,10,10);
        tv2.setText("1. Nguyễn văn A; \n 2. Lâm Văn Mới;");
        toast2.setView(tv2);
        toast2.setDuration	Toast.LENGTH_LONG;
        toast2.show();
        return true;
    case R.id.item2_3:
        Toast toast3=new Toast(this);
        toast3.setGravity(Gravity.CENTER,100,100);
        TextView tv3=new TextView(this);
        tv3.setBackgroundColor(Color.BLACK);
        tv3.setTextColor(Color.YELLOW);
        tv3.setTextSize(25);
        Typeface t3=Typeface.create("serif",Typeface.BOLD);
        tv3.setTypeface(t3);
        tv3.setPadding(10,10,10,10);
        tv3.setText("1. Trần Văn Bảy; \n 2. Nguyễn Văn Tâm; \n 3. Châu
Thành Quý;");

        toast3.setView(tv3);
        toast3.setDuration	Toast.LENGTH_LONG;
        toast3.show();
        return true;
    case R.id.item2_4:
        Toast toast4=new Toast(this);
        toast4.setGravity(Gravity.CENTER,100,100);
        TextView tv4=new TextView(this);
        tv4.setBackgroundColor(Color.BLACK);
        tv4.setTextColor(Color.YELLOW);

```

## Buổi 1: Lập trình giao diện người dùng mức cao

```
        tv4.setTextSize(25);
        Typeface t4=Typeface.create("serif",Typeface.BOLD);
        tv4.setTypeface(t4);
        tv4.setPadding(10,10,10,10);
        tv4.setText("1. Lê Thị Lý; \n Trần Ngọc Mai.");
        toast4.setView(tv4);
        toast4.setDuration	Toast.LENGTH_LONG;
        toast4.show();
        return true;
    default:
        return super.onContextItemSelected(item);
    }
}
// Tạo Popup menu
private void showPopupMenu() {
    // Tạo 1 popupmenu.
    PopupMenu popupMenu=new PopupMenu(this,btnpopup);
    //Lấy giao diện menu trong resource.
    popupMenu.getMenuInflater().inflate(R.menu.menu_popup,popupMenu.getMenu());
    //Lập trình hành động khi chọn item.
    popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
        @Override
        public boolean onMenuItemClick(MenuItem item) {
            switch (item.getItemId()) {
                case R.id.item3_1: btnpopup.setText("MENU COPY");// Đổi tên nút
menu.
                    break;
            }
            switch (item.getItemId()) {
                case R.id.item3_2: btnpopup.setText("MENU PAST");
                    break;
            }
            switch (item.getItemId()) {
                case R.id.item3_3: btnpopup.setText("MENU SAVE");
                    break;
            }
            switch (item.getItemId()) {
                case R.id.item3_4: btnpopup.setText("MENU DELETE");
                    break;
            }
            return false;
        }
    });
    popupMenu.show();
}
```

-Hết-