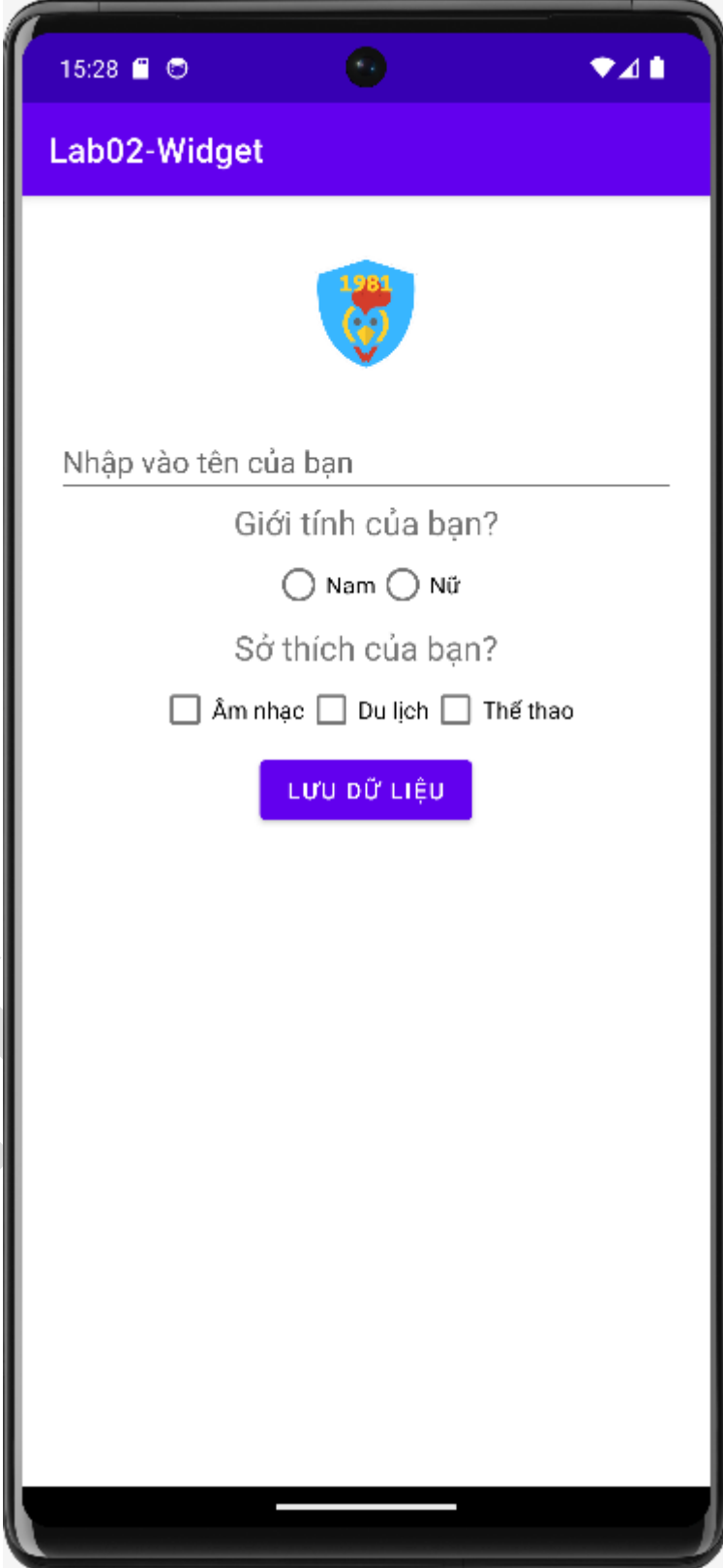


## Lab02

### Bài 1.

#### *Yêu cầu:*

Sử dụng các View đã học thiết kế giao diện sau:



The screenshot shows a mobile application interface titled "Lab02-Widget". At the top, there is a status bar with the time "15:28" and icons for battery, signal, and Wi-Fi. Below the title bar, there is a blue header with the text "Lab02-Widget". The main content area is white and contains a blue shield icon with a yellow bird and the year "1981". Below the icon, there is a text input field with the placeholder "Nhập vào tên của bạn". Underneath the input field, there is a label "Giới tính của bạn?" followed by two radio buttons labeled "Nam" and "Nữ". Below this, there is a label "Sở thích của bạn?" followed by three checkboxes labeled "Âm nhạc", "Du lịch", and "Thể thao". At the bottom of the form, there is a blue button with the text "LƯU DỮ LIỆU".

## Hướng dẫn:

- ☐ Sử dụng `LinearLayout` làm layout mặc định.
- ☐ Trong `LinearLayout` sử dụng một số thuộc tính:
  - ❖ **`android:paddingTop`** - xác định khoảng cách từ cạnh trên màn hình đến nội dung
  - ❖ **`android:gravity`** – căn nội dung, có thể dùng kết hợp 2 giá trị
  - ❖ **`android:orientation`** – xác định chiều hiển thị của nội dung (ngang – dọc)
- ☐ Sử dụng `ImageView` để hiển thị Logo.
  - ❖ **`android:src`** – lấy đường dẫn bức hình và hiển thị
- ☐ Sử dụng `EditText` để tạo ô nhập dữ liệu:
  - ❖ **`android:hint`** – hiển thị nội dung gợi ý nhập dữ liệu vào `EditText`
- ☐ Sử dụng `RadioButton` để tạo lựa chọn "Giới tính"
  - ❖ Để sử dụng được đúng chức năng của `RadioButton` ta cần sử dụng thêm `RadioGroup` để gộp nhóm các `RadioButton` lại với nhau.
- ☐ Sử dụng `CheckBox` để tạo lựa chọn "Sở thích"
- ☐ Sử dụng `Button` để tạo nút "Lưu dữ liệu"

Trong Android, `android:layout_width` và `android:layout_height` là hai thuộc tính quan trọng được sử dụng để định cấu hình kích thước của một phần tử trong giao diện người dùng. Dưới đây là ý nghĩa của các giá trị `match_parent` và `wrap_content`:

### 1. `match_parent`:

- Khi bạn đặt `android:layout_width="match_parent"` hoặc `android:layout_height="match_parent"`, phần tử sẽ được mở rộng để điền vào toàn bộ không gian cha của nó.
- Nếu áp dụng cho `width`, nó sẽ mở rộng theo chiều ngang của cha.
- Nếu áp dụng cho `height`, nó sẽ mở rộng theo chiều dọc của cha.

### 2. `wrap_content`:

- Khi bạn đặt `android:layout_width="wrap_content"` hoặc `android:layout_height="wrap_content"`, phần tử sẽ chỉ mở rộng để bao quanh nội dung bên trong nó.

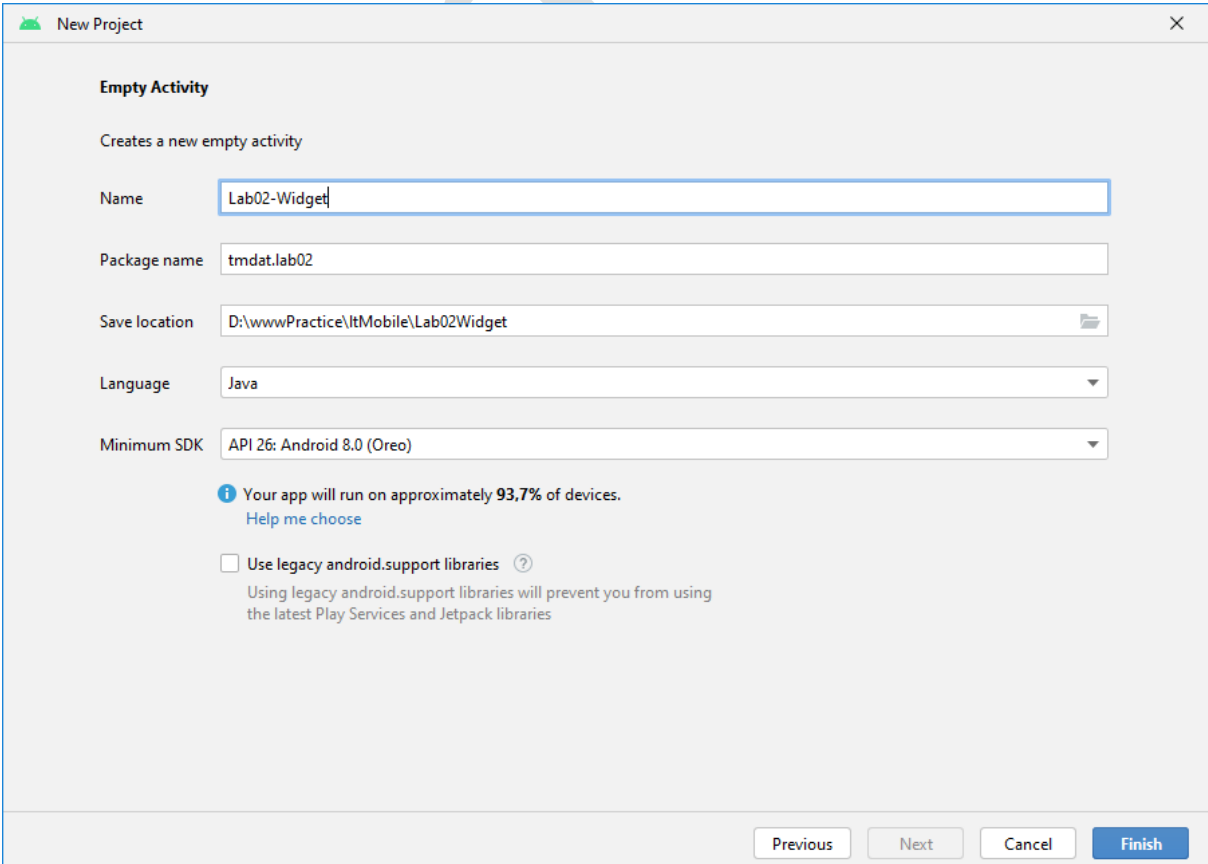
- Nếu áp dụng cho `width`, phần tử sẽ có chiều ngang chỉ đủ để đựng nội dung bên trong nó.

- Nếu áp dụng cho `height`, phần tử sẽ có chiều dọc chỉ đủ để đựng nội dung bên trong nó.

Ví dụ, nếu bạn có một `TextView` và muốn nó chiếm toàn bộ chiều rộng của màn hình, bạn có thể đặt `android:layout_width="match_parent"` để nó mở rộng theo chiều ngang. Nếu bạn muốn nó chỉ có chiều rộng bằng đúng đủ để chứa nội dung, bạn có thể đặt `android:layout_width="wrap_content"`. Tương tự, bạn có thể áp dụng nguyên tắc tương tự cho chiều dọc bằng cách sử dụng `android:layout_height`.

### Thao tác

Tạo Project mới:



New Project

Empty Activity

Creates a new empty activity

Name: Lab02-Widget

Package name: tmdat.lab02

Save location: D:\wwwPractice\ltMobile\Lab02Widget

Language: Java

Minimum SDK: API 26: Android 8.0 (Oreo)

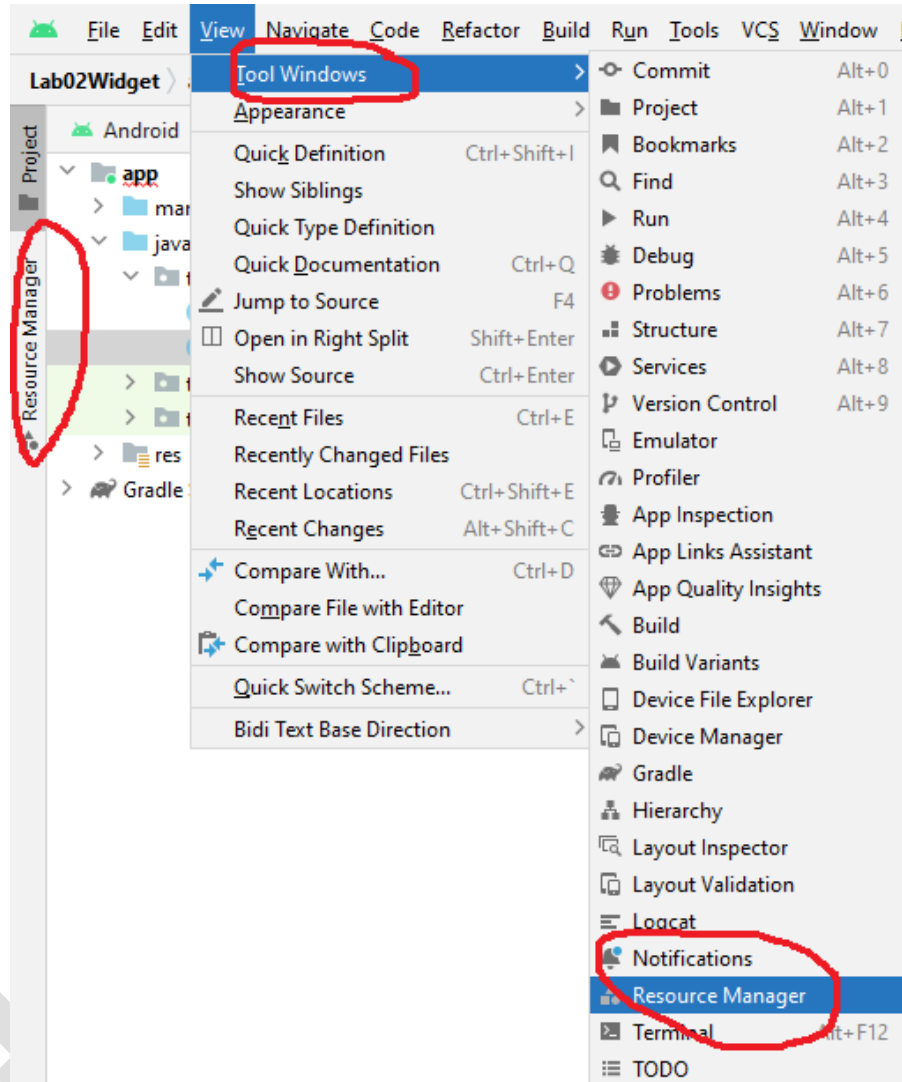
*Your app will run on approximately 93,7% of devices.*  
[Help me choose](#)

☐ Use legacy android.support libraries [?](#)  
Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

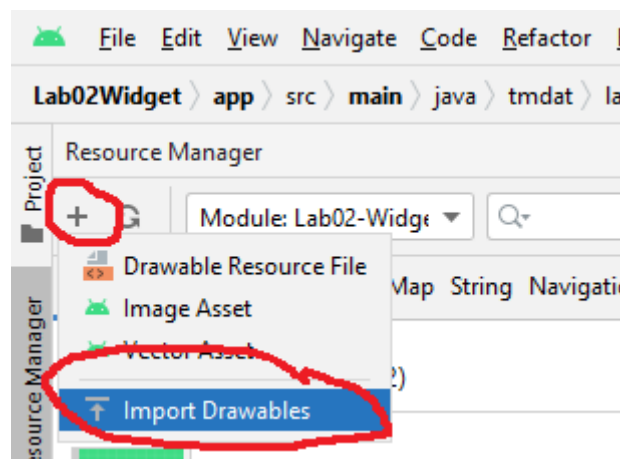
Previous Next Cancel Finish

Thêm ảnh vào Project:

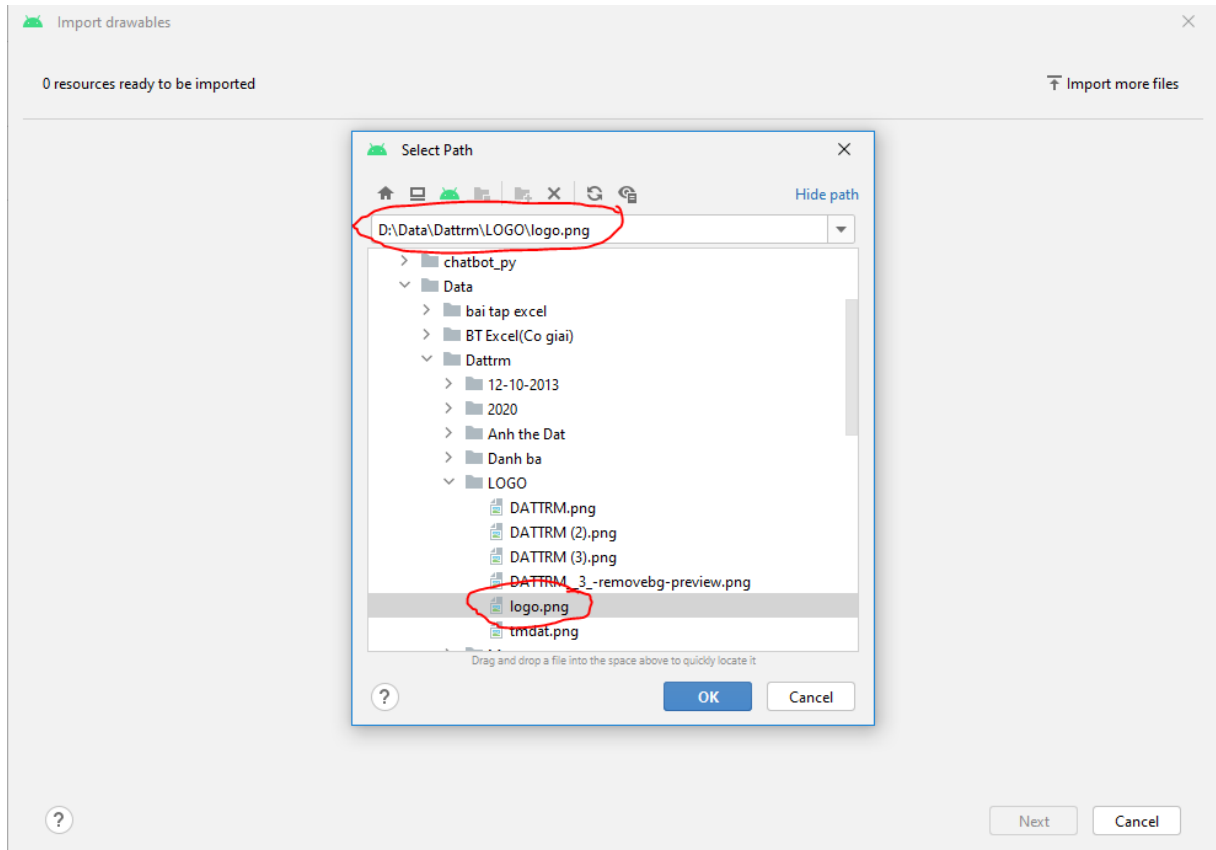
Trong Android Studio, hãy nhấp vào View > Tool Windows > Resource Manager hoặc nhấp vào thẻ Resource Manager ở bên cạnh cửa sổ Project.



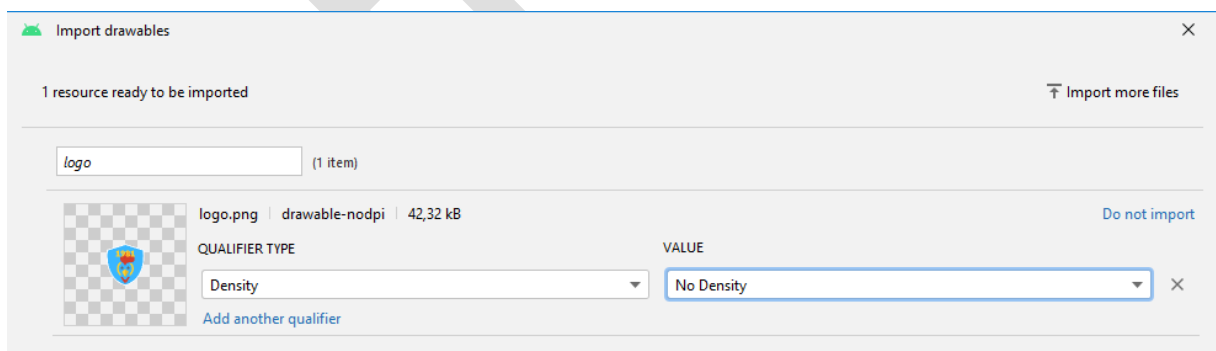
Nhấp vào + (Add resources to the module) > Import Drawables.



Trở đến vị trí chứa ảnh:



Android Studio sẽ cho thấy bản xem trước của hình ảnh được tải xuống. Chọn Density (Mật độ) trên danh sách thả xuống của Qualifier Type (Loại bộ hạn định). Ví dụ:



## Ví dụ mã nguồn thiết kế layout trong activity main.xml:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:gravity="center|top"
8     android:orientation="vertical"
9     android:paddingHorizontal="20dp"
10    android:paddingTop="20dp"
11    tools:context=".MainActivity">
12
13    <ImageView
14        android:layout_width="250dp"
15        android:layout_height="100dp"
16        android:src="@drawable/logo"
17        android:contentDescription="Logo ứng dụng"/>
18
19    <EditText
20        android:layout_width="match_parent"
21        android:layout_height="wrap_content"
22        android:layout_marginTop="16dp"
23        android:hint="Nhập vào tên của bạn"/>
24
25    <TextView
26        android:layout_width="wrap_content"
27        android:layout_height="wrap_content"
28        android:text="Giới tính của bạn?"
29        android:textSize="20dp" />
30
31    <RadioGroup
32        android:layout_width="wrap_content"
33        android:layout_height="wrap_content"
34        android:orientation="horizontal">
35
36        <RadioButton
37            android:layout_width="wrap_content"
38            android:layout_height="wrap_content"
39            android:text="Nam"/>
40
41        <RadioButton
42            android:layout_width="wrap_content"
43            android:layout_height="wrap_content"
44            android:text="Nữ"/>
45    </RadioGroup>
```

```

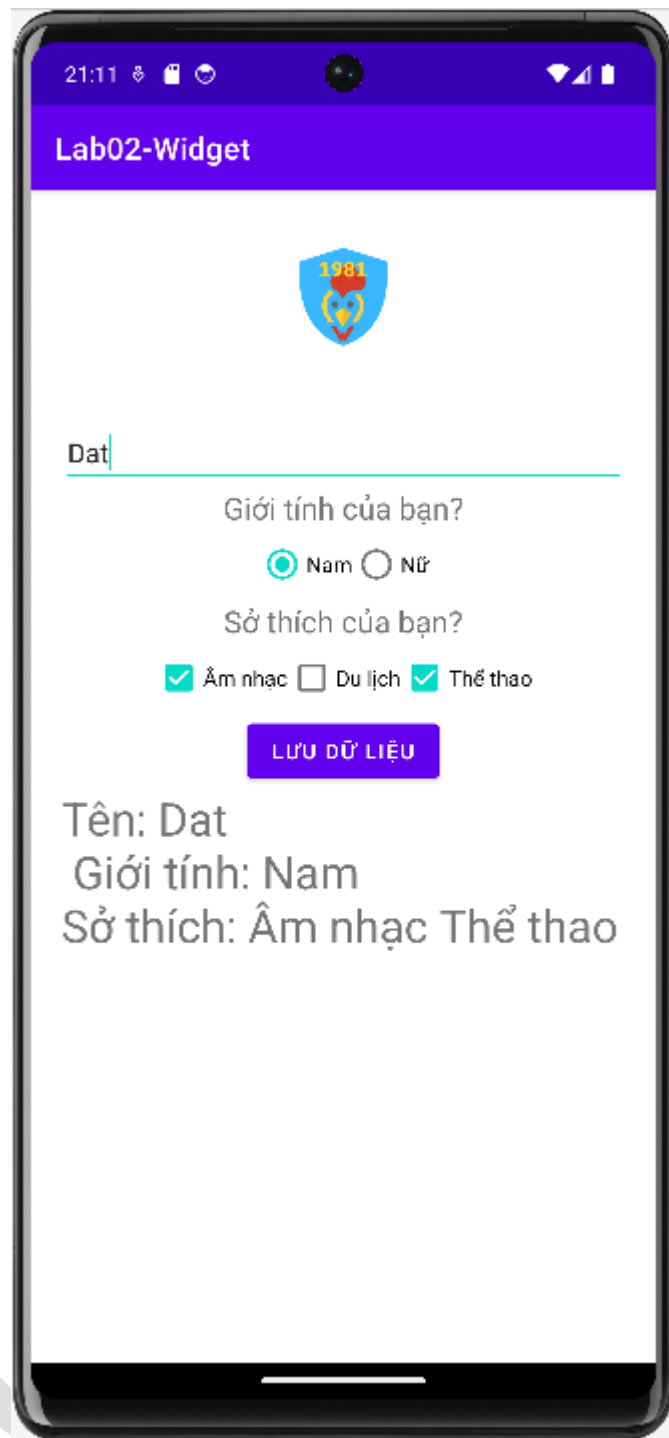
47 <TextView
48     android:layout_width="wrap_content"
49     android:layout_height="wrap_content"
50     android:text="Sở thích của bạn?"
51     android:textSize="20dp" />
52
53 <LinearLayout
54     android:layout_width="match_parent"
55     android:layout_height="wrap_content"
56     android:orientation="horizontal"
57     android:gravity="center|top">
58
59     <CheckBox
60         android:layout_width="wrap_content"
61         android:layout_height="wrap_content"
62         android:text="Âm nhạc"/>
63
64     <CheckBox
65         android:layout_width="wrap_content"
66         android:layout_height="wrap_content"
67         android:text="Đu lịch"/>
68
69     <CheckBox
70         android:layout_width="wrap_content"
71         android:layout_height="wrap_content"
72         android:text="Thể thao"/>
73 </LinearLayout>
74
75 <Button
76     android:layout_width="wrap_content"
77     android:layout_height="wrap_content"
78     android:text="Lưu dữ liệu"/>
79 </LinearLayout>

```

## **Bài 2.**

### **Yêu cầu**

Phát triển tiếp bài 1, các thông tin nhập vào các widget sẽ hiển thị ra màn hình



### **Hướng dẫn:**

***Đặt id cho các widget trong `activity_main.xml`.***

Lưu ý: id của mỗi widget không được trùng nhau và đều phải có thành phần @+id, hãy đặt id có chứa widget cần để tương tác.

Ví dụ:



```
<EditText
    android:id="@+id/nameEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="30dp"
    android:hint="Nhập vào tên của bạn"/>
```

id được đặt là nameEditText vì đây là Widget EditText và dùng để nhập vào Tên.

Thực hiện tương tự với các Widget khác.

**Để hiển thị dữ liệu, sử dụng 1 widget `TextView`:**

```
<TextView
    android:id="@+id/hienThiTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="30sp"/>
```

**Trong file MainActivity, ánh xạ cho các widget đã được đặt id trong file layout**

```
public class MainActivity extends AppCompatActivity {

    private TextView hienThiTextView;
    private EditText nameEditText;
    private RadioButton maleRadioButton;
    private RadioButton femaleRadioButton;
    private CheckBox amNhacCheckBox;
    private CheckBox duLichCheckBox;
    private CheckBox theThaoCheckBox;
    private Button luuDulieuButton;

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

        hienThiTextView = findViewById(R.id.hienThiTextView);
        nameEditText = findViewById(R.id.nameEditText);
        maleRadioButton = findViewById(R.id.maleRadioButton);
        femaleRadioButton = findViewById(R.id.femaleRadioButton);
        amNhacCheckBox = findViewById(R.id.amNhacCheckBox);
```

```

    duLichCheckBox = findViewById(R.id.duLichCheckBox);
    theThaoCheckBox = findViewById(R.id.theThaoCheckBox);
    luuDuLieuButton = findViewById(R.id.luuDuLieuButton);

    // Thêm sự kiện nghe cho nút luuDuLieuButton
    luuDuLieuButton.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            // Lấy dữ liệu từ các Widget
            String name = nameEditText.getText().toString();
            String gender = maleRadioButton.isChecked() ? "Nam": "Nữ";
            String hobbies = getHobbiesString();
            // Hiển thị dữ liệu trong hienThiTextView
            String displayText = "Tên: " + name + "\nGiới tính: "
            + gender + "\nSở thích: " + hobbies;
            hienThiTextView.setText(displayText);
        }
    });
}

// Xử lý Mục sở thích trả về nhiều giá trị
private String getHobbiesString() {
    StringBuilder stringBuilder = new StringBuilder();
    if (amNhacCheckBox.isChecked()) {
        stringBuilder.append("Âm nhạc ");
    }
    if (duLichCheckBox.isChecked()) {
        stringBuilder.append("Du lịch ");
    }
    if (theThaoCheckBox.isChecked()) {
        stringBuilder.append("Thể thao ");
    }
    return stringBuilder.toString().trim();
}
}

```

**onCreate(Bundle savedInstanceState)** là một phương thức của lớp AppCompatActivity trong Android. Nó chịu trách nhiệm cho quá trình tạo

ra và khởi tạo một hoạt động (activity) trong Android khi hoạt động đó được tạo mới.

Phương thức **onCreate** là một trong những phương thức quan trọng nhất trong vòng đời của một hoạt động Android. Nó thường được sử dụng để thực hiện công việc khởi tạo cần thiết cho hoạt động, chẳng hạn như ánh xạ các thành phần giao diện người dùng (widgets), xử lý dữ liệu đầu vào, và thiết lập các tương tác với người dùng.

**setOnClickListener**: Phương thức này được sử dụng để đặt một người nghe sự kiện cho một đối tượng View. Khi đối tượng View đó được nhấn (click), người nghe sự kiện sẽ được kích hoạt.

**new View.OnClickListener()**: Đây là cách tạo một đối tượng mới của một lớp triển khai giao diện OnClickListener. Giao diện này có một phương thức là `onClick(View v)`, và bạn cần triển khai phương thức này để xác định hành động cụ thể mà bạn muốn thực hiện khi sự kiện click xảy ra.

```
new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        // Xử lý hành động khi nút được nhấn  
    }  
}
```