



School: ..... Campus: .....  
Academic Year: ..... Subject Name: ..... Subject Code: .....  
Semester: ..... Program: ..... Branch: ..... Specialization: .....  
Date: .....

## Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiment :

### Coding Phase: Pseudo Code / Flow Chart / Algorithm

#### Open Android Studio

- Launch Android Studio and open your existing project or create a new one.

#### Create or Open an XML Layout File

- In your project's res/layout folder, open the XML file for the activity or fragment where you want to display the Alert Dialog.
- Add a Button (or any other trigger) that will initiate the Alert Dialog.

#### Go to the Corresponding Activity/Fragment

- Navigate to the .java or .kt file linked to the XML layout you modified.

#### Initialize the Button in Code

- Use findViewById() to reference the Button in your Activity/Fragment.

#### Set an OnClickListener for the Button

- Inside the listener, define what should happen when the Button is clicked.

#### Create the AlertDialog Builder

- Inside the onClick method, instantiate an AlertDialog.Builder.
- Set the dialog title, message, and define positive and negative buttons for user interaction.

#### Handle Dialog Button Clicks

- In the .setPositiveButton() and .setNegativeButton() methods, define the actions to be performed when the user clicks each button.

#### Show the Dialog

- Call .show() on the AlertDialog.Builder instance to display the dialog.

#### Run and Test the Application

- Build and run your project to ensure the Alert Dialog appears and behaves as expected.

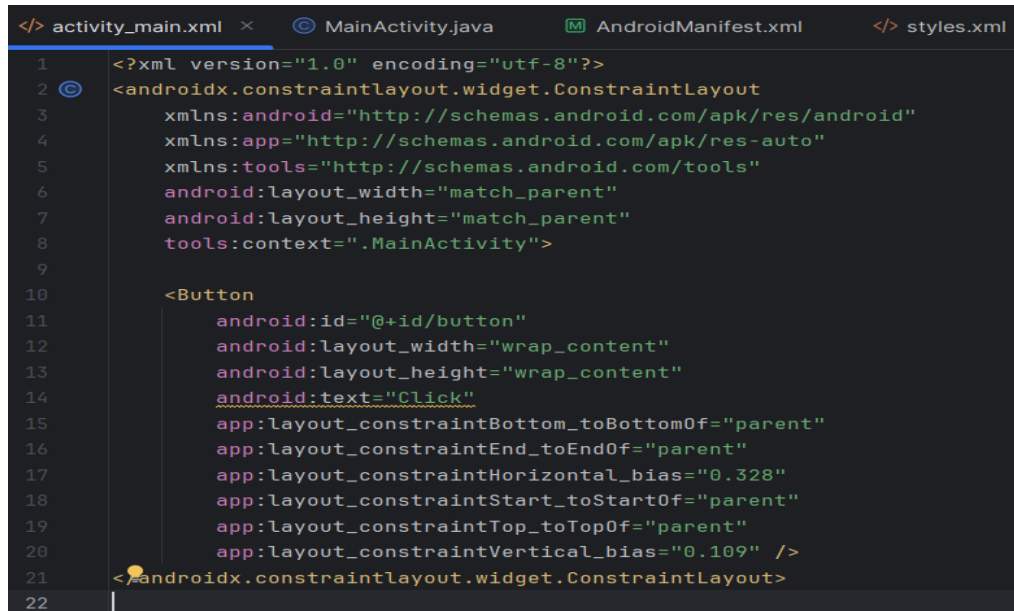
### Testing Phase: Compilation of Code (error detection)

There is no error.

## Implementation Phase: Final Output(no error)

### Step 1: Design the Layout

- Open activity\_main.xml from the res → layout folder.
- Drag a **Button** from the **Palette** to the design view.



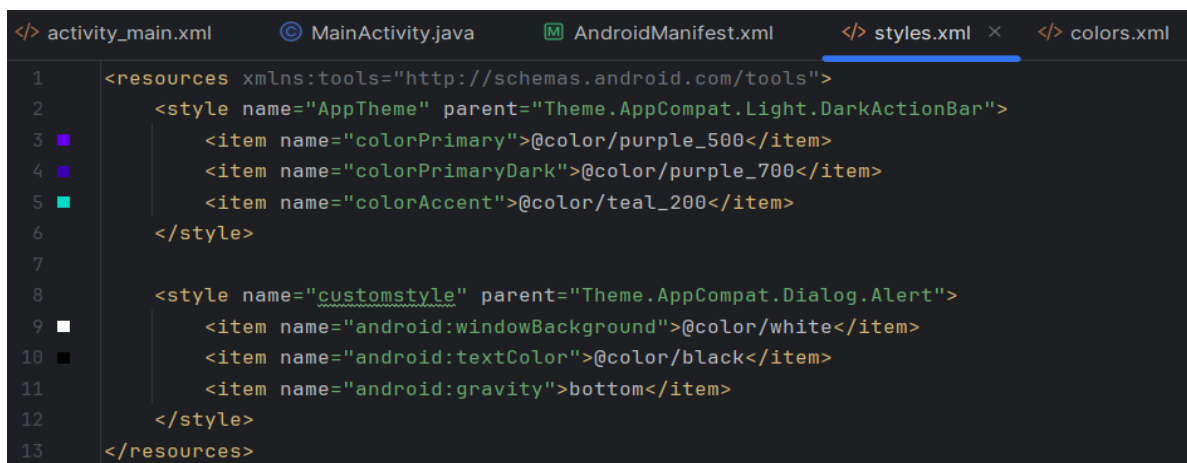
```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
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:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <Button
11         android:id="@+id/button"
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="Click"
15         app:layout_constraintBottom_toBottomOf="parent"
16         app:layout_constraintEnd_toEndOf="parent"
17         app:layout_constraintHorizontal_bias="0.328"
18         app:layout_constraintStart_toStartOf="parent"
19         app:layout_constraintTop_toTopOf="parent"
20         app:layout_constraintVertical_bias="0.109" />
21 </androidx.constraintlayout.widget.ConstraintLayout>
22

```

### Configuring styles.xml (For Custom Theme/Style)

Location: res → values → styles.xml



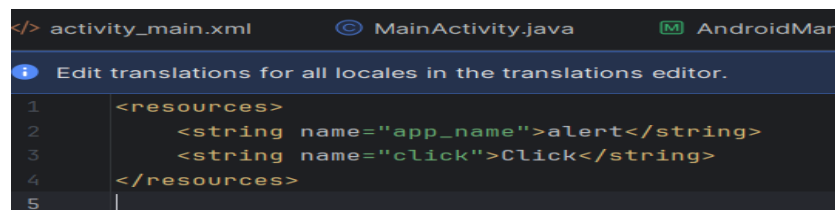
```

1  <resources xmlns:tools="http://schemas.android.com/tools">
2      <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
3          <item name="colorPrimary">@color/purple_500</item>
4          <item name="colorPrimaryDark">@color/purple_700</item>
5          <item name="colorAccent">@color/teal_200</item>
6      </style>
7
8      <style name="customstyle" parent="Theme.AppCompat.Dialog.Alert">
9          <item name="android:windowBackground">@color/white</item>
10         <item name="android:textColor">@color/black</item>
11         <item name="android:gravity">bottom</item>
12     </style>
13 </resources>

```

🔗 Open strings.xml from res → values.

Add string values for app name, alert title, message, and button text



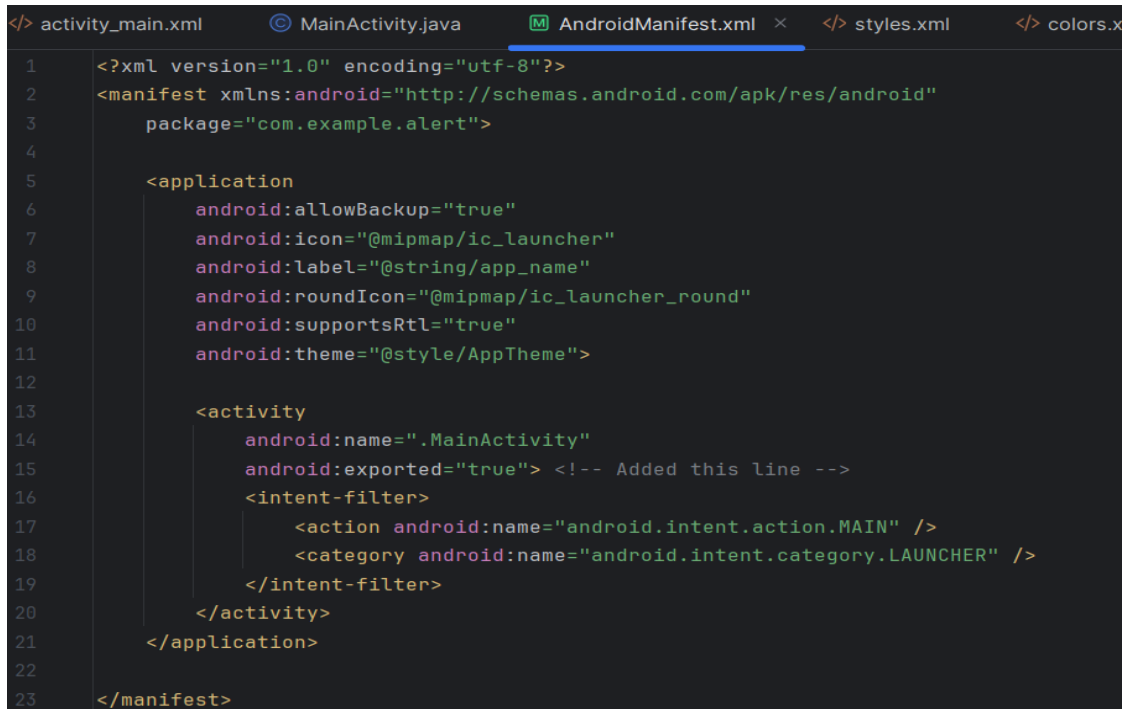
```

1  <resources>
2      <string name="app_name">alert</string>
3      <string name="click">Click</string>
4  </resources>
5

```

## Implementation Phase: Final Output(no error)

Open AndroidManifest.xml in the app → src → main folder.  
add android:exported="true" to the activity tag with an intent filter.

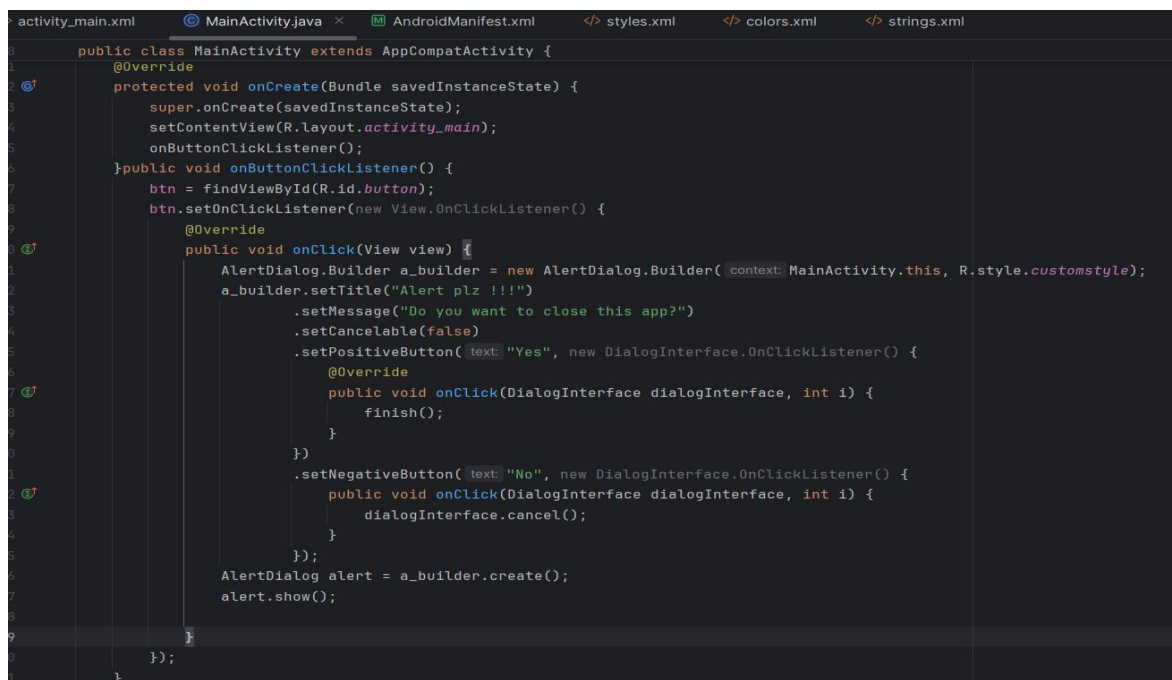


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.alert">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">

        <activity
            android:name=".MainActivity"
            android:exported="true"> <!-- Added this line -->
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

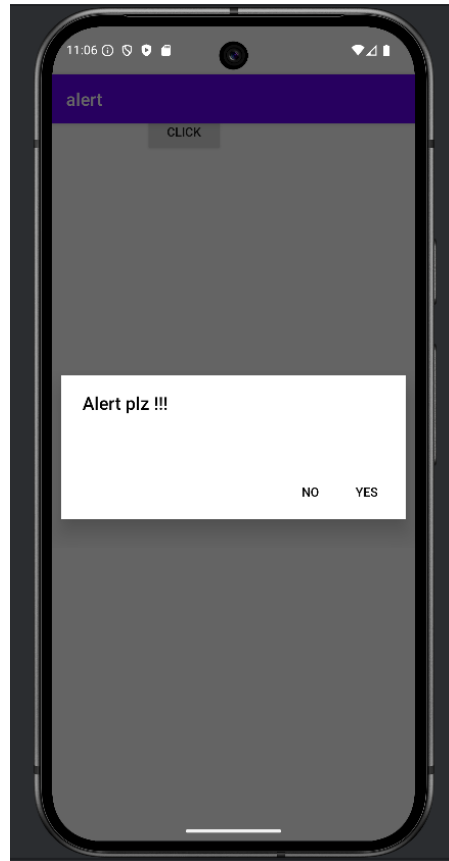
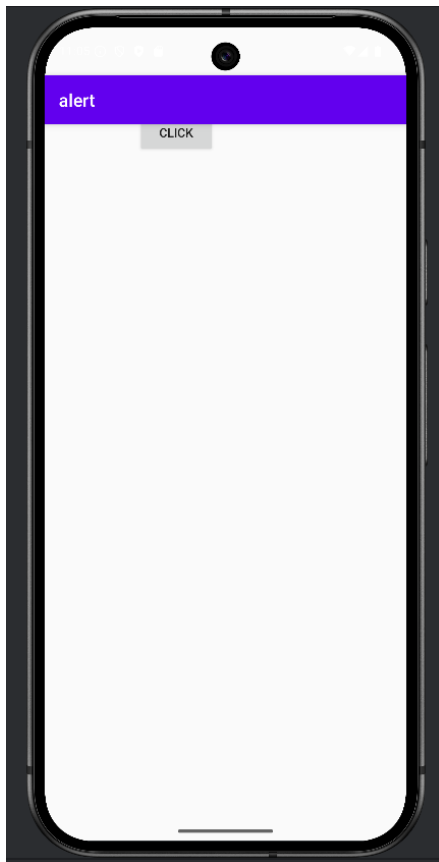
### Configuring MainActivity.java



```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        findViewById(R.id.button).setOnClickListener();
    }

    public void onClickButtonClickListener() {
        btn = findViewById(R.id.button);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                AlertDialog.Builder a_builder = new AlertDialog.Builder(context, R.style.customstyle);
                a_builder.setTitle("Alert plz !!!")
                    .setMessage("Do you want to close this app?")
                    .setCancelable(false)
                    .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialogInterface, int i) {
                            finish();
                        }
                    })
                    .setNegativeButton("No", new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialogInterface, int i) {
                            dialogInterface.cancel();
                        }
                    });
                AlertDialog alert = a_builder.create();
                alert.show();
            }
        });
    }
}
```

Click **Run** to test the Alert Dialog.



## ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
<b>Total</b>	<b>50</b>		

**Signature of the Student:**

**Name :**

**Regn. No. :**

**Signature of the Faculty:**

