Experiment 06: To develop an android application that creates an alert upon receiving a message

PART A

A.1 Aim: **To develop an android application that creates an alert upon receiving a message**.

A.2 Objectives: To introduce students with various tools like Android Studio, NS2, Wireshark, Cisco packet tracer, WAP supported browser etc.

A.3 Outcomes:

After successful completion of this experiment students will be able to develop an android application that creates an alert upon receiving a message

A.4Theory:

SOFTWARE:

- Android Studio
- The Android SDK (Starter Package)
- Gradle
- Java Development Kit (JDK) 5

DESCRIPTION:

- Open android studio and select new android project . • Give project name and select next
- Choose the android version.
- Enter the package name. package name must be two word separated by comma and click finish
- Go to package explorer in the left hand side and select our project. • Go to res folder and select layout. Double click the main.xml file

- Now you can see the Graphics layout window.

INPUT:

MAINACTIVITY.JAVA

```
package com.alert;

import androidx.appcompat.app.AppCompatActivity; import android.os.Bundle; import
android.app.Notification;
      import   android.app.NotificationManager;  import         android.app.PendingIntent; import
android.content.Intent;
import android.view.View; import android.widget.Button; import android.widget.EditText;

public class MainActivity extends AppCompatActivity{ Button notify; EditText
e;

@Override
                              protected        void    onCreate(Bundle     savedInstanceState)  {
  super.onCreate(savedInstanceState); setContentView(R.layout.activity_main);
notify= (Button) findViewById(R.id.button); e= (EditText) findViewById(R.id.editText);

notify.setOnClickListener(new View.OnClickListener() { @Override public
void onClick(View v) {
Intent intent = new Intent(MainActivity.this, SecondActivity.class);
PendingIntent pending = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);
Notification noti = new Notification.Builder(MainActivity.this).setContentTitle("New
Message").setContentText(e.getText().toString()).setSmallIcon(R.mipmap.ic_l
auncher).setContentIntent(pending).build();
                              NotificationManager    manager       =       (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
noti.flags |= Notification.FLAG_AUTO_CANCEL; manager.notify(0, noti);
}
});
}
}
```

ACTIVITY_MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
android:layout_margin="10dp" android:orientation="vertical">

<TextView
android:layout_width="wrap_content"          android:layout_height="wrap_content"
android:text="Message" android:textSize="30sp" />

<EditText
android:id="@+id/editText" android:layout_width="match_parent"
android:layout_height="wrap_content" android:singleLine="true" android:textSize="30sp" />

<Button
android:id="@+id/button"                                                   android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_margin="30dp"
android:layout_gravity="center" android:text="Notify" android:textSize="30sp"/> </LinearLayout>
```

ANDROIDMANIFEST.XML

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

<application
android:allowBackup="true"   android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"   android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true" android:theme="@style/Theme.Alert">
<activity
android:name=".SecondActivity" android:exported="true" />
<activity
android:name=".MainActivity" android:exported="true">
```

```
<intent-filter>
<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>
</application>
```

CREATING A NEW ACTIVITY

Create a new activity named “SecondActivity”

File -> New -> Activity -> Empty Activity

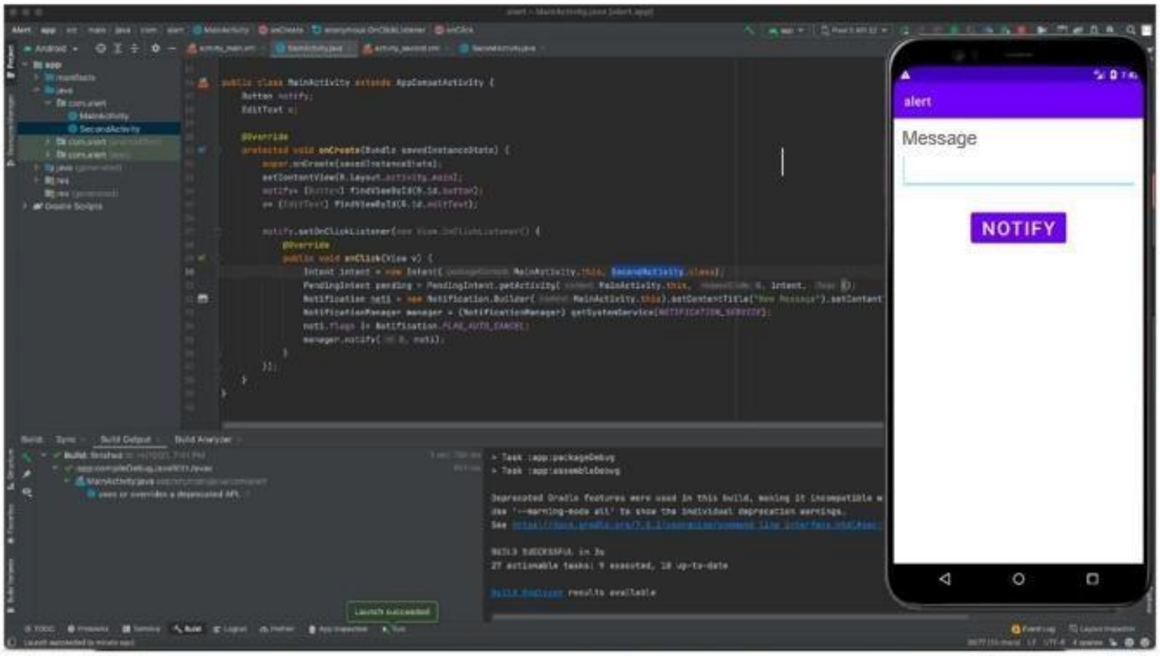
SECONDACTIVITY.JAVA

```
package com.alert;

import androidx.appcompat.app.AppCompatActivity; import android.os.Bundle; public
class SecondActivity extends AppCompatActivity{ @Override
                                   protected     void   onCreate(Bundle    savedInstanceState) {
  super.onCreate(savedInstanceState); setContentView(R.layout.activity_second);
}
}
```

ACTIVITY_SECOND.XML

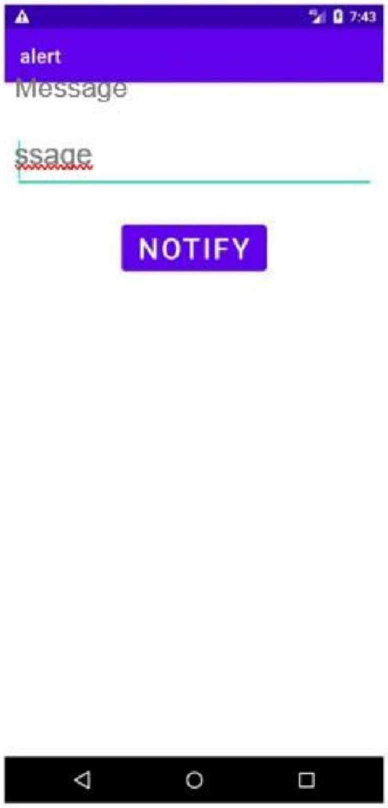
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://
schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
           xmlns:tools="http://schemas.android.com/tools"   android:layout_width="match_parent"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

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

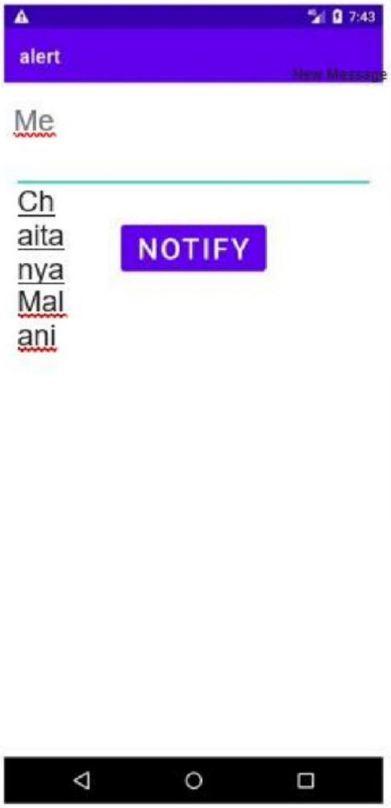
OUTPUT:

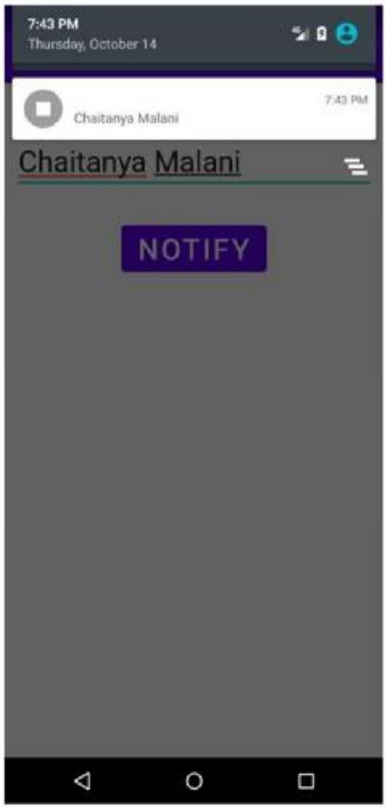


```
public class MainActivity extends AppCompatActivity {
    Button notify;
    EditText e;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        notify= (Button) findViewById(R.id.button);
        e= (EditText) findViewById(R.id.editText);

        notify.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent( packageContext: MainActivity.this, SecondActivity.class);
                PendingIntent pending = PendingIntent.getActivity( context: MainActivity.this, requestCode: 0, intent, flags: 0);
                Notification noti = new Notification.Builder( context: MainActivity.this).setContentTitle("New Message").setContent
                NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
                noti.flags |= Notification.FLAG_AUTO_CANCEL;
                manager.notify( id: 0, noti);
            }
        });
    }
}
```

```
> Task :app:packageDebug
> Task :app:assembleDebug

Deprecated Gradle features were used in this build, making it incompatible w
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/7.0.2/userguide/command_line_interface.html#sec:

BUILD SUCCESSFUL in 3s
27 actionable tasks: 9 executed, 18 up-to-date

Build Analyzer results available
```







PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll. No:B30	Name: Pranjal Bhatt
Class :TE COMPS B	Batch:B2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student/steps:

MainActivity.java:

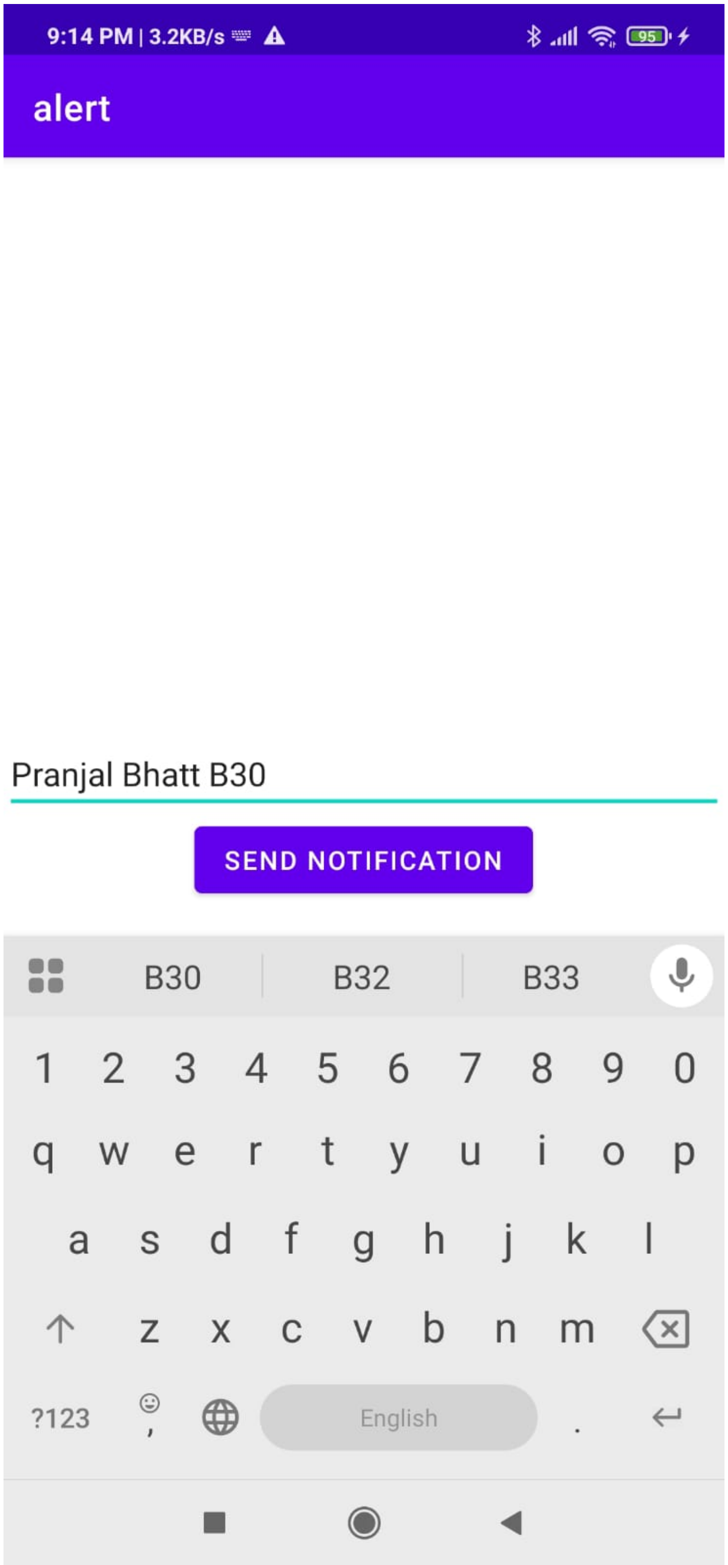
```
package com.example.alert;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
  Button notify;
  EditText e;
  private final String CHANNEL_ID = "alert_channel";

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

    notify = findViewById(R.id.button);
    e = findViewById(R.id.editText);

    // Request notification permission (Android 13+)
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
      if                                                  (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.POST_NOTIFICATIONS)
          != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this,
            new String[]{android.Manifest.permission.POST_NOTIFICATIONS}, 1);
      }
    }

    // Create notification channel (needed for Android 8.0+)
    createNotificationChannel();

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

private void createNotificationChannel() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        CharSequence name = "Alert Notifications";
        String description = "Includes all alert notifications";
        int importance = NotificationManager.IMPORTANCE_HIGH; // Changed to HIGH

        NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name, importance);
        channel.setDescription(description);

        NotificationManager notificationManager = getSystemService(NotificationManager.class);
        notificationManager.createNotificationChannel(channel);
    }
}

private void sendNotification() {
    String message = e.getText().toString().trim(); // Remove unnecessary spaces

    Intent intent = new Intent(MainActivity.this, SecondActivity.class);
    intent.putExtra("message", message); // Ensure the message is passed

    PendingIntent pendingIntent = PendingIntent.getActivity(
            MainActivity.this,
            0,
            intent,
            PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE
    );

    NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
            .setSmallIcon(R.mipmap.ic_launcher)
            .setContentTitle("New Message")
            .setContentText(message.isEmpty() ? "No message entered" : message) // Avoid null text
            .setPriority(NotificationCompat.PRIORITY_HIGH) // Changed to HIGH
            .setAutoCancel(true)
        .setContentIntent(pendingIntent);

    NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);
    if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.POST_NOTIFICATIONS) != PackageManager.PERMISSION_GRANTED) {
      return;
    }

    int notificationId = (int) System.currentTimeMillis(); // Unique notification ID
    notificationManager.notify(notificationId, builder.build());
  }

  @Override
  public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == 1) {
      if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
        // Permission granted
      } else {
        // Permission denied
      }
    }
  }
}
```

AndroidManifest.xml:

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

  <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
  <application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
```

```
        android:supportsRtl="true"
        android:theme="@style/Theme.Alert"
        tools:targetApi="31">
        <activity
            android:name=".SecondActivity"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

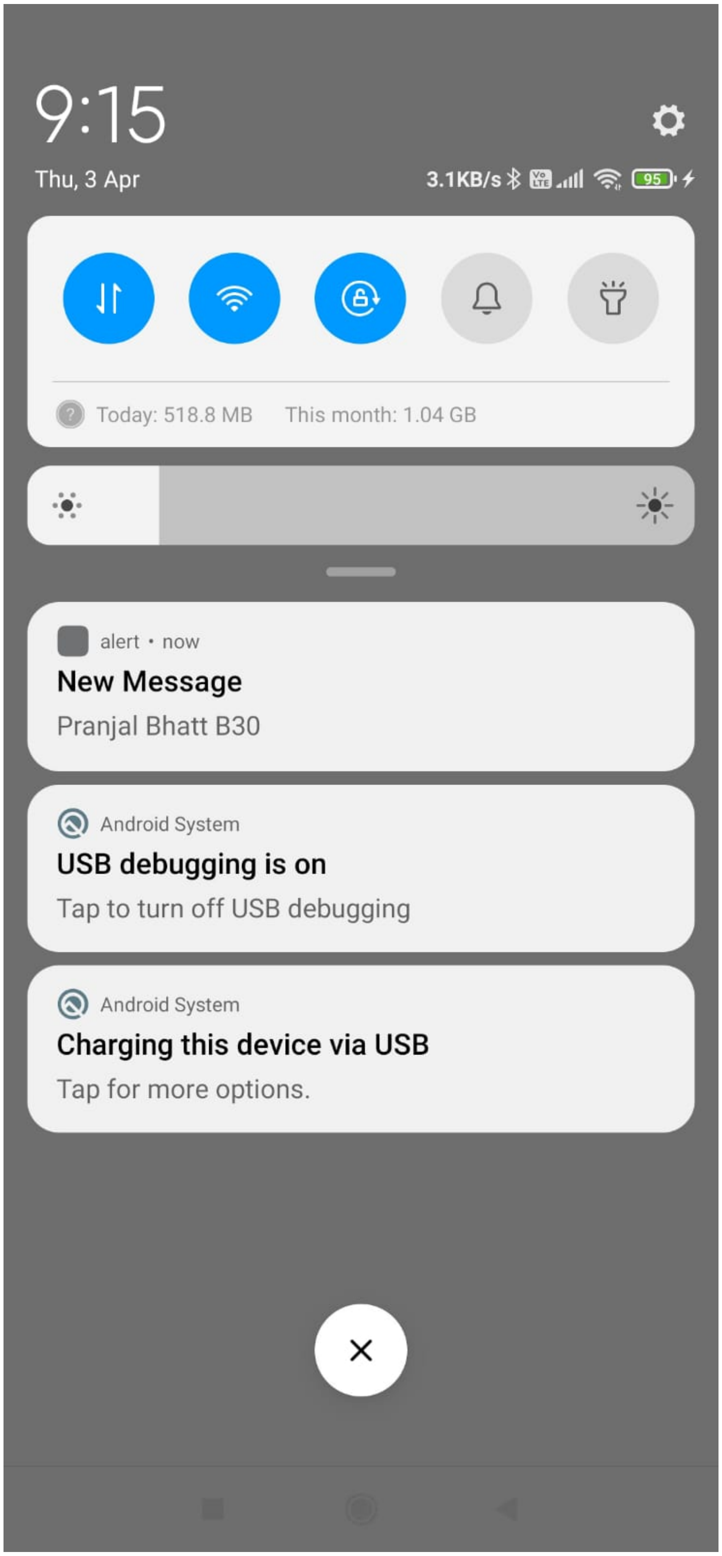
B.2 Input and Output:



9:14 PM | 3.2KB/s

alert

Pranjal Bhatt B30



alert • now

New Message

Pranjal Bhatt B30

Android System

USB debugging is on

Tap to turn off USB debugging

Android System

Charging this device via USB

Tap for more options.

B.3 Observations and learning:

During the experiment, an Android application was developed using Android Studio to create an alert upon receiving a message. The setup involved configuring the Android SDK, JDK, and Gradle, followed by creating a new project. The layout and logic were implemented in XML and Java/Kotlin, respectively. Testing was performed using an emulator or a real device. The application successfully triggered an alert (such as a notification, sound, or pop-up) when a message was received.

B.4 Conclusion:

The experiment demonstrated the practical implementation of Android app development using Android Studio. Students gained hands-on experience with project creation, UI design, and event handling in Android. By working on this alert system, they understood how to integrate background services and notifications into mobile applications.

B.5 Question of Curiosity

1) Explain different steps required to build up this alert project?

- **Set Up Development Environment:**

  - • Install Android Studio, Java Development Kit (JDK), and Android SDK.
  - Configure Gradle for project dependencies.

- **Create a New Android Project:**

  - • Open Android Studio and select "New Project."
  - Choose an appropriate project template (e.g., Empty Activity).
  - Enter the application name, package name, and select the minimum Android version.
  - Click “Finish” to create the project.

- **Design the User Interface (UI):**

  - • Navigate to the `res/layout` folder and open `activity_main.xml`.
  - Use the graphical layout editor to design the interface (e.g., add a `TextView` to display messages).

- **Implement Message Receiving Functionality:**

    - • Modify the `AndroidManifest.xml` file to request the necessary permissions:

  - • Register a `BroadcastReceiver` to listen for incoming SMS messages.

- **Write Java/Kotlin Code to Trigger an Alert:**
  - • Create a new class extending `BroadcastReceiver`:
- **Test the Application:**
  - • Run the application on an emulator or a physical device.
  - Use an SMS testing tool to simulate incoming messages.
- **Debug and Optimize:**
  - • Check logs in Logcat for debugging.
  - Optimize code for better performance.