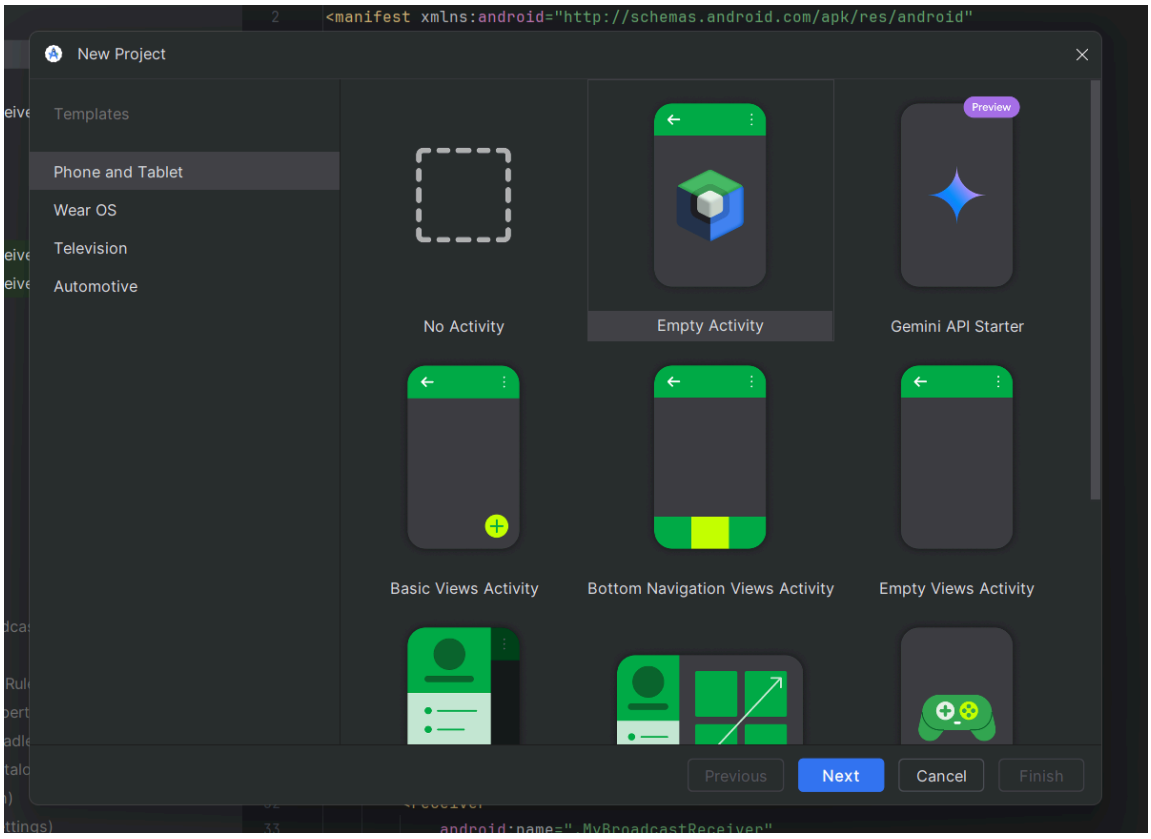
Practical 05: Android Application Development

Aim:- Create an Android application to demonstrate the use of Broadcast listeners.

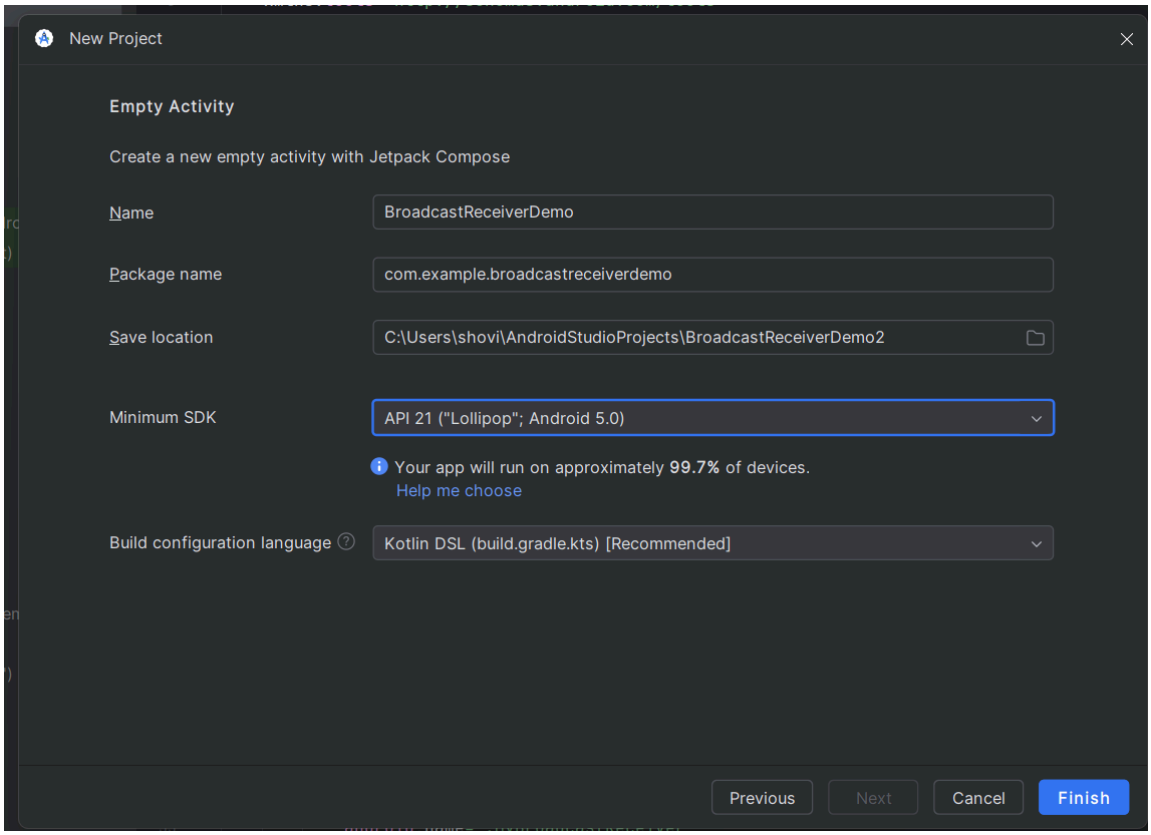
📌 Step 1: Create a New Android Project

1. **Open Android Studio (Ladybug version).**
2. **Click New Project → Select Empty Activity → Click Next.**



3. **Name the project BroadcastListenerApp.**
4. **Set Minimum SDK to API 21 (Lollipop) for better compatibility.**

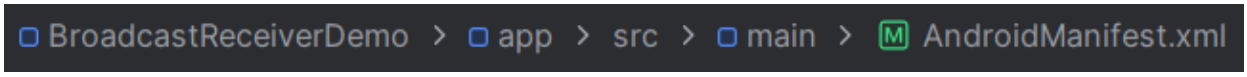
5. **Click Finish and let Android Studio set up your project.**



📌 Step 2: Modify the Manifest File

We need to declare permissions and register the broadcast receiver in `AndroidManifest.xml`.

- **File Path:** `app/src/main/AndroidManifest.xml`



Code:-

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools">
```

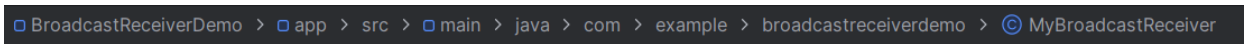
```xml
<!-- Permission for receiving system broadcast -->
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>

<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:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.BroadcastReceiverDemo"
    tools:targetApi="31">

    <!-- Main Activity -->
    <activity
        android:name=".MainActivity"
        android:exported="true"
        android:label="@string/app_name"
        android:theme="@style/Theme.BroadcastReceiverDemo">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
```

```
        <!-- Broadcast Receiver -->
        <receiver
            android:name=".MyBroadcastReceiver"
            android:exported="true"
            android:enabled="true">
            <intent-filter>
                <action android:name="android.intent.action.AIRPLANE_MODE"/>
            </intent-filter>
        </receiver>

    </application>
</manifest>
```

📌 Step 3: Create the Broadcast Receiver

A Broadcast Receiver listens for system events like Airplane Mode changes.

🔹 **File Path:**
`app/src/main/java/com/example/broadcastlistenerapp/MyBroadcastReceiver.java`
Create a new Java class `MyBroadcastReceiver`.java inside the



Code:-

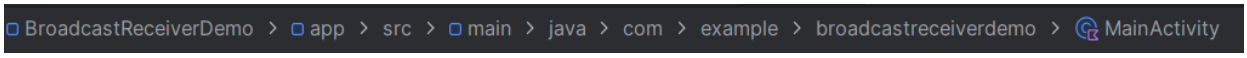
```
package com.example.broadcastlistenerapp;


import android.content.BroadcastReceiver;
```

```
import android.content.Context;

import android.content.Intent;

import android.widget.Toast;



public class AirplaneModeReceiver extends BroadcastReceiver {

   @Override

   public void onReceive(Context context, Intent intent) {

       boolean isAirplaneMode = intent.getBooleanExtra("state", false);

       if (isAirplaneMode) {

           Toast.makeText(context, "Airplane Mode Enabled",
Toast.LENGTH_SHORT).show();

       } else {

           Toast.makeText(context, "Airplane Mode Disabled",
Toast.LENGTH_SHORT).show();

       }

   }

}
```

- **File Path:**

app/src/main/java/com/example/broadcastlistenerapp/`MainActivity.java`

Create a new Java class `MainActivity`.java inside the



Code:-

```
package com.example.broadcastreceiverdemo
```

```
import android.content.Context
```

```
import android.content.Intent

import android.os.Bundle

import android.provider.Settings

import android.widget.Toast

import android.widget.ToggleButton

import androidx.appcompat.app.AppCompatActivity


class MainActivity : AppCompatActivity() {


   private var toggleButton: ToggleButton? = null


   override fun onCreate(savedInstanceState: Bundle?) {

       super.onCreate(savedInstanceState)

       setContentView(R.layout.activity_main)


       toggleButton = findViewById(R.id.toggleButton)


       // Set ToggleButton listener to handle clicks

       toggleButton?.setOnCheckedChangeListener { _, isChecked ->

           if (isChecked) {

               // Guide user to the Airplane Mode settings page

               val intent = Intent(Settings.ACTION_AIRPLANE_MODE_SETTINGS)

               startActivity(intent)

               // Show a Toast message

               Toast.makeText(this, "Please toggle Airplane Mode manually",
Toast.LENGTH_SHORT).show()
```

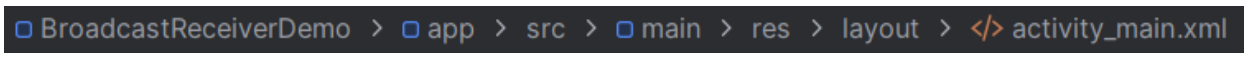
```
            } else {

                // Guide user to the Airplane Mode settings page

                val intent = Intent(Settings.ACTION_AIRPLANE_MODE_SETTINGS)

                startActivity(intent)

                // Show a Toast message

                Toast.makeText(this, "Please toggle Airplane Mode manually",
Toast.LENGTH_SHORT).show()

            }

        }

    }

}
```

✅ **This will listen for Airplane Mode changes and display a toast message when toggled.**

📌 Step 4: Create a Simple UI

We'll create a simple layout with a TextView to display instructions.

- **File Path:** `app/src/main/res/layout/activity_main.xml`



Code:-

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="match_parent"

    android:layout_height="match_parent"
```

```
   android:orientation="vertical"

   android:gravity="center"

   android:padding="20dp">


   <TextView

       android:id="@+id/tv_status"

       android:layout_width="wrap_content"

       android:layout_height="wrap_content"

       android:text="Toggle Airplane Mode to see the Broadcast Receiver in
action."

       android:textSize="18sp"

       android:textAlignment="center"/>


   <LinearLayout

       android:layout_width="match_parent"

       android:layout_height="match_parent"

       android:orientation="vertical"

       android:gravity="center">


       <ToggleButton

           android:id="@+id/toggleButton"

           android:layout_width="wrap_content"

           android:layout_height="wrap_content"

           android:textOn="ON"

           android:textOff="OFF"

           android:text="Toggle" />
```

```
    </LinearLayout>


</LinearLayout>
```

We'll create a simple layout with a TextView to display instructions.

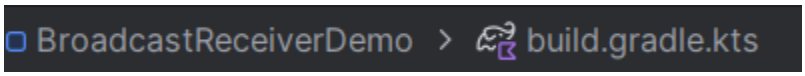
- **File Path: `app/src/main/res/values/styles.xml`**

Code : -

```
<?xml version="1.0" encoding="utf-8"?>
<resources>


    <!-- Base application theme -->
    <style name="Theme.BroadcastReceiverDemo"
parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>


    <color name="colorPrimary">#B94141</color>
    <color name="colorPrimaryDark">#1C65BD</color>
    <color name="colorAccent">#26439A</color>
</resources>
```

📌 Step 5: gradle Setup

🔹 **File Path:/build.gradle.kts**

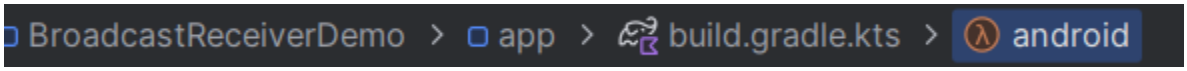


Code:-

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.
```

```
plugins {

    alias(libs.plugins.android.application) apply false

    alias(libs.plugins.kotlin.android) apply false

    alias(libs.plugins.kotlin.compose) apply false

}
```

🔹 **File Path:**



Code:-

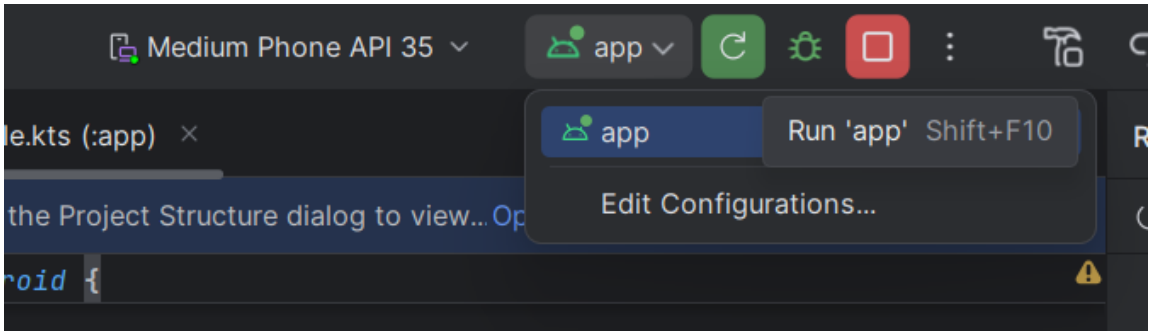
```
plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
    alias(libs.plugins.kotlin.compose)
}

android {
    namespace = "com.example.broadcastreceiverdemo"
    compileSdk = 35

    defaultConfig {
        applicationId = "com.example.broadcastreceiverdemo"
        minSdk = 22
        targetSdk = 35
        versionCode = 1
        versionName = "1.0"
```

```
        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
    kotlinOptions {
        jvmTarget = "11"
    }
    buildFeatures {
        compose = true
    }
}

dependencies {
    // Adding AppCompat dependency
    implementation("androidx.appcompat:appcompat:1.5.1")

    // Other dependencies
    implementation(libs.androidx.core.ktx)
    implementation(libs.androidx.lifecycle.runtime.ktx)
    implementation(libs.androidx.activity.compose)
    implementation(platform(libs.androidx.compose.bom))
    implementation(libs.androidx.ui)
    implementation(libs.androidx.ui.graphics)
    implementation(libs.androidx.ui.tooling.preview)
    implementation(libs.androidx.material3)

    // Testing dependencies
    testImplementation(libs.junit)
    androidTestImplementation(libs.androidx.junit)
    androidTestImplementation(libs.androidx.espresso.core)
    androidTestImplementation(platform(libs.androidx.compose.bom))
    androidTestImplementation(libs.androidx.ui.test.junit4)

    // Debug dependencies
    debugImplementation(libs.androidx.ui.tooling)
    debugImplementation(libs.androidx.ui.test.manifest)
```

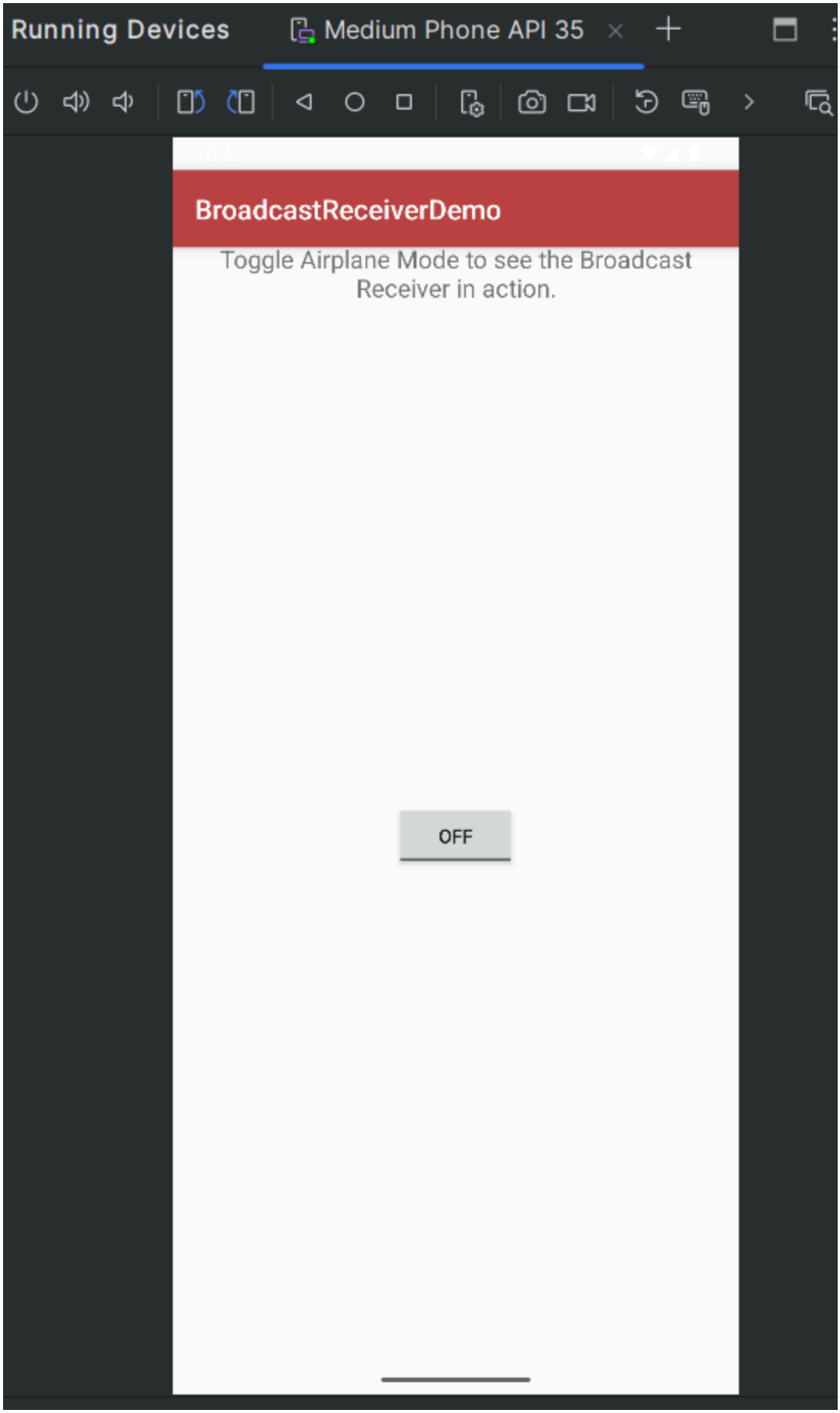
```
}
```

📌 Step 6: Run and Test the Application

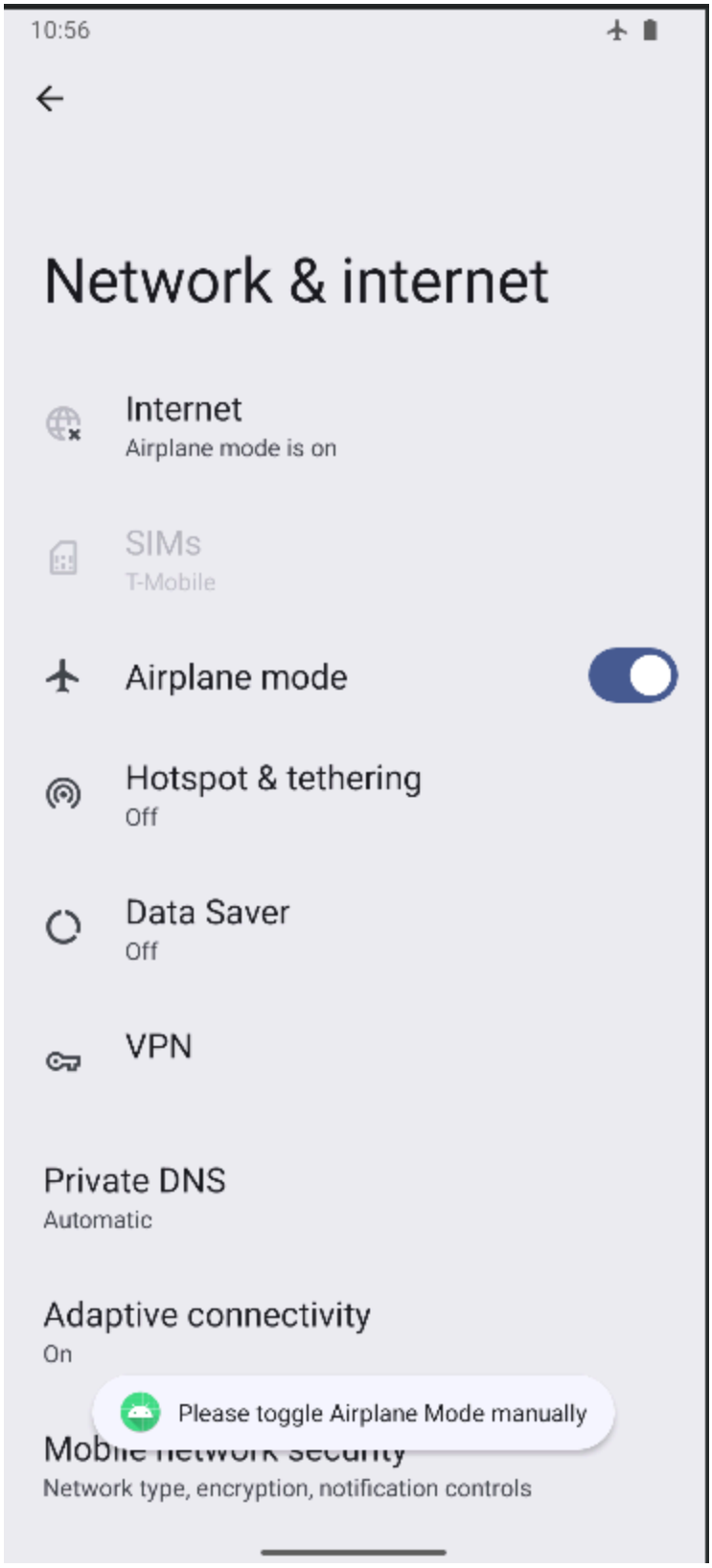
1. **Connect an emulator or real device.**
2. **Run the application.**

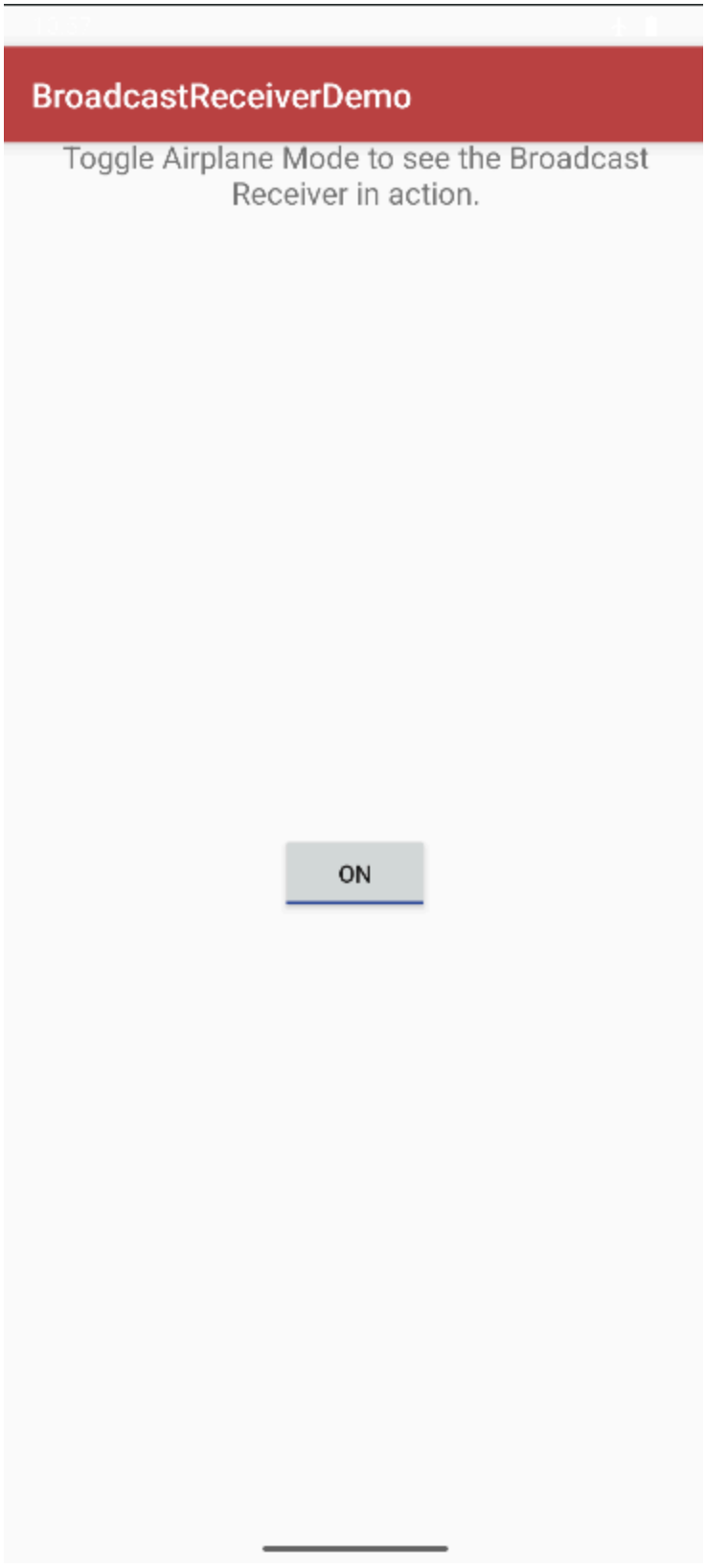


3. **You should see a Toast message when Airplane Mode is turned ON or OFF.**


Running Devices
Medium Phone API 35
BroadcastReceiverDemo
Toggle Airplane Mode to see the Broadcast Receiver in action.
OFF

4. **Toggle Airplane Mode (Settings > Network & Internet > Airplane Mode).**


10:56
Network & internet
Internet
Airplane mode is on
SIMs
T-Mobile
Airplane mode
Hotspot & tethering
Off
Data Saver
Off
VPN
Private DNS
Automatic
Adaptive connectivity
On
Please toggle Airplane Mode manually
Network type, encryption, notification controls


BroadcastReceiverDemo
Toggle Airplane Mode to see the Broadcast Receiver in action.
ON

✅ **Done!**