ROYAL UNIVERSITY OF BHUTAN
COLLEGE OF SCIENCE AND TECHNOLOGY
PHUENTSHOLING: BHUTAN

## PLAGIARISM DECLARATION FORM

Student Name: Divyash Chhetri

Student No: 02200174

Module No and Title of the module: CTE306 - Mobile Application Development
Assignment no and Title of the Assignment: Lab Wok 07

Section H2 of the Royal University of Bhutan's *Wheel of Academic Law* provides the following definition of academic dishonesty:

"Academic dishonesty may be defined as any attempt by a student to gain an unfair advantage in any assessment. It may be demonstrated by one of the following:

**Collusion:** the representation of a piece of unauthorized group work as the work of a single candidate.

**Commissioning:** submitting an assignment done by another person as the student's own work.

**Duplication**: the inclusion in coursework of material identical or substantially similar to material which has already been submitted for any other assessment within the University.

**False declaration**: making a false declaration in order to receive special consideration by an Examination Board or to obtain extensions to deadlines or exemption from work.

**Falsification of data**: presentation of data in laboratory reports, projects, etc., based on work purported to have been carried out by the student, which have been invented, altered or copied by the student.

**Plagiarism**: the unacknowledged use of another's work as if it were one's own.

Examples are:

- verbatim copying of another's work without acknowledgement
- paraphrasing of another's work by simply changing a few words or altering the order of presentation, without acknowledgement
- ideas or intellectual data in any form presented as one's own without acknowledging the source(s)
- making significant use of unattributed digital images such as graphs, tables, photographs, etc. taken from test books, articles, films, plays, handouts, internet, or any other source, whether published or unpublished
- submission of a piece of work which has previously been assessed for a different award or module or at a different institution as if it were new work
- use of any material without prior permission of copyright from appropriate authority or owner of the materials used"

### Student Declaration

I confirm that I have read and understood the above definitions of academic dishonesty. I declare that I have not committed any academic dishonesty when completing the attached piece of work.

Signature of Student: ivyash Chhetri

Date: 20 Sept 2022

# Lab 07

CTE306 – Mobile Application Development

Date – 20th September 2022

Divyash Chhetri
02200174
BE 3 IT

Module Tutor: Mr. Pema Galey

Department of Information Technology
**College of Science and Technology**

## Aim

To implement BroadCast with certain BroadCastRecievers in Android Studio

## Task

Implement the following Broadcasts with a BroadcastReceiver:

a. System broadcasts such as ACTION_BATTERY_LOW, ACTION_POWER_CONNECTED, ACTION_POWER_DISCONNECTED, ACTION_AIRPLANE_MODE_CHANGED.
b. CustomBroadcasts to notify through Toast message by defining the constant string CUSTOM_TOAST_MESSAGE
c. LocalBroadcast through LocalBroadcastManager.

## Theory

An Android component called a broadcast receiver enables users to send and receive events from android application. The Android runtime alerts all installed applications when an event takes place. It functions similarly to the publish-subscribe design pattern and is used for asynchronous inter-process communication.
Applications can sign up to get notifications from the Android system when certain events occur, such as the completion of bootup or a low battery. Any program is able to produce original broadcasts of its own.

Sensitive information must never be broadcasted globally for obvious reasons. However, one may broadcast similar information locally by using the LocalBroadcastManager class from the Android Support Library.
The LocalBroadcastManager is significantly more effective since it does not require inter-process communication.

# Broadcast and Broadcast Receiver

## Program Code

### *AndroidManifest.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.broadcast">
    <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.Broadcast">
        <receiver
            android:name=".ReceiverClass"
            android:enabled="true"
            android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.action.BATTERY_LOW"></action>
                <action
android:name="com.example.broadcast.CUSTOM_TOAST_MESSAGE"></action>
            </intent-filter>
        </receiver>

        <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>
```

*activity_main.xml*

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

    <Button
        android:id="@+id/sendBroadcast"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="90sp"
        android:layout_marginTop="180sp"
        android:backgroundTint="#64B5F6"
        android:text="@string/send_custom_broadcast"
        android:textColor="@color/black"/>

    <Button
        android:id="@+id/localBroadcast"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="96sp"
        android:layout_marginTop="50sp"
        android:backgroundTint="#64B5F6"
        android:text="@string/send_local_broadcast"
        android:textColor="@color/black"/>
</LinearLayout>
```

*LocalReceiver.java*

```
package com.example.broadcast;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;

public class LocalReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
    // TODO: This method is called when the BroadcastReceiver is receiving
    // an Intent broadcast.
        String action = intent.getAction();

if(MainActivity.CUSTOM_TOAST_MESSAGE_FOR_LOCAL_BROADCAST.equals(action)) {
            Toast.makeText(context, "Local Broadcast Received ",
Toast.LENGTH_LONG).show();
        }
    }
}
```

*ReceiverClass.java*

```
package com.example.broadcast;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;

public class ReceiverClass extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
    // TODO: This method is called when the BroadcastReceiver is receiving
    // an Intent broadcast.
```

```
        String intentType = intent.getAction();
        if (intentType != null) {
            String toastMessage = "No intent Action";

            switch (intentType) {
                case Intent.ACTION_AIRPLANE_MODE_CHANGED:
                    boolean status = intent.getBooleanExtra("state", true);
                    if (status) {
                        toastMessage = "Airplane Mode Enabled";
                    } else {
                        toastMessage = "Airplane Mode Disabled";
                    }
                    break;

                case Intent.ACTION_BATTERY_LOW:
                    toastMessage = "Battery Low, Connect To A Power
Source";

                    break;

                case Intent.ACTION_POWER_CONNECTED:
                    toastMessage = "Power Source Connected";
                    break;

                case Intent.ACTION_POWER_DISCONNECTED:
                    toastMessage = "Power Source Disconnected";
                    break;

                case MainActivity.CUSTOM_TOAST_MESSAGE:
                    toastMessage = "Custom Broadcast Received";
                    break;
            }

            Toast.makeText(context, toastMessage,
Toast.LENGTH_SHORT).show();
        }
    }
}
```

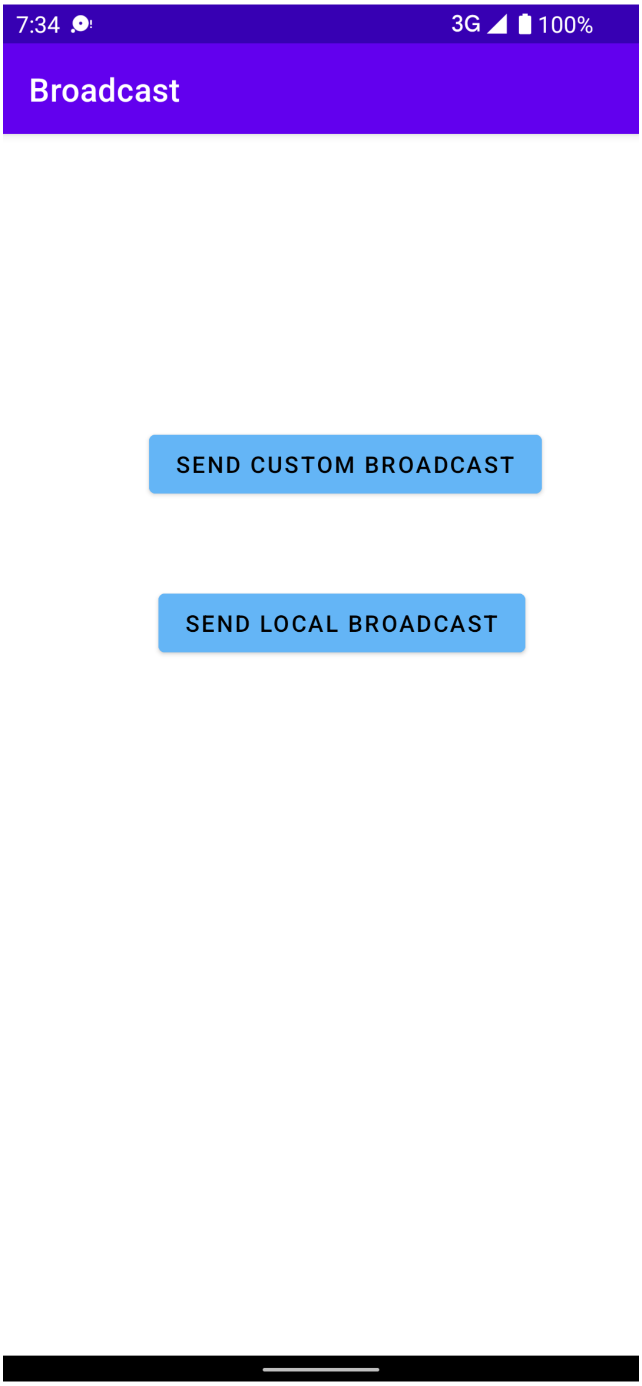
*MainActivity.java*

```
package com.example.broadcast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.localbroadcastmanager.content.LocalBroadcastManager;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    public static final String CUSTOM_TOAST_MESSAGE =
"com.example.broadcast.CUSTOM_TOAST_MESSAGE";
    public static final String CUSTOM_TOAST_MESSAGE_FOR_LOCAL_BROADCAST =
"com.example.broadcast.CUSTOM_TOAST_MESSAGE_FOR_LOCAL_BROADCAST";

    Button customBtn, localBtn;

    ReceiverClass myReceiver = new ReceiverClass();
    IntentFilter filter = new IntentFilter();

    private LocalBroadcastManager localBroadcastManager = null;
    private LocalReceiver localBReceiver;

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

        customBtn = findViewById(R.id.sendBroadcast);

        filter.addAction(Intent.ACTION_AIRPLANE_MODE_CHANGED);
        filter.addAction(Intent.ACTION_POWER_CONNECTED);
        filter.addAction(CUSTOM_TOAST_MESSAGE);
        this.registerReceiver(myReceiver, filter);
        customBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent();
                i.setAction(MainActivity.CUSTOM_TOAST_MESSAGE);
                sendBroadcast(i);
            }
        });
        localBroadcastManager = LocalBroadcastManager.getInstance(this);
        localBReceiver = new LocalReceiver();
        filter.addAction(CUSTOM_TOAST_MESSAGE_FOR_LOCAL_BROADCAST);
        localBroadcastManager.registerReceiver(localBReceiver, filter);

        localBtn = findViewById(R.id.localBroadcast);

        localBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new
Intent(CUSTOM_TOAST_MESSAGE_FOR_LOCAL_BROADCAST);
                localBroadcastManager.sendBroadcast(i);
            }
        });
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        MainActivity.this.unregisterReceiver(myReceiver);
        if (localBroadcastManager != null) {
            localBroadcastManager.unregisterReceiver(localBReceiver);
        }
    }
}
```

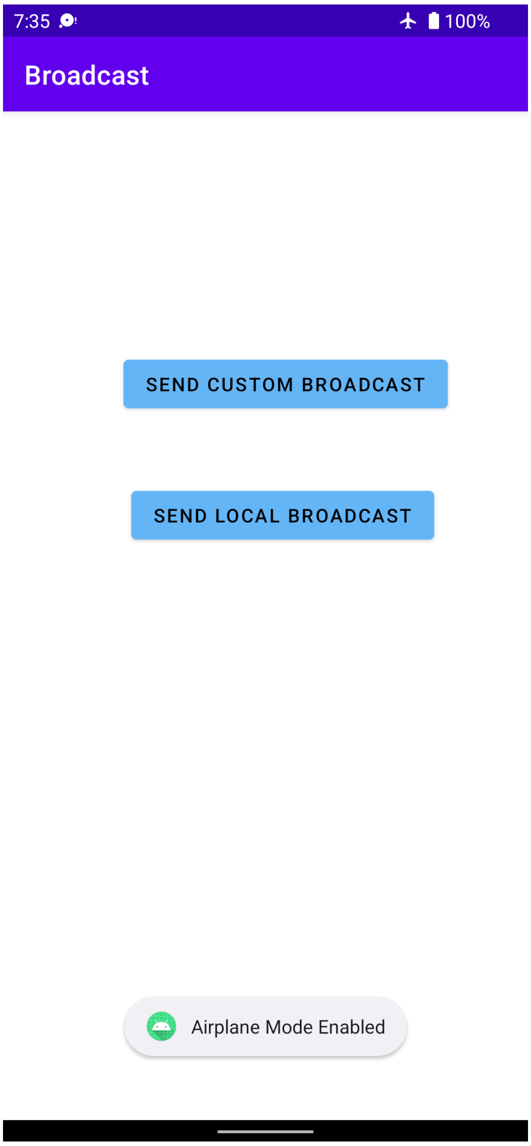
## Output

System broadcasts such as ACTION_BATTERY_LOW, ACTION_POWER_CONNECTED and ACTION_POWER_DISCONNECTED are also implemented but screenshots are unavailable through the emulator.

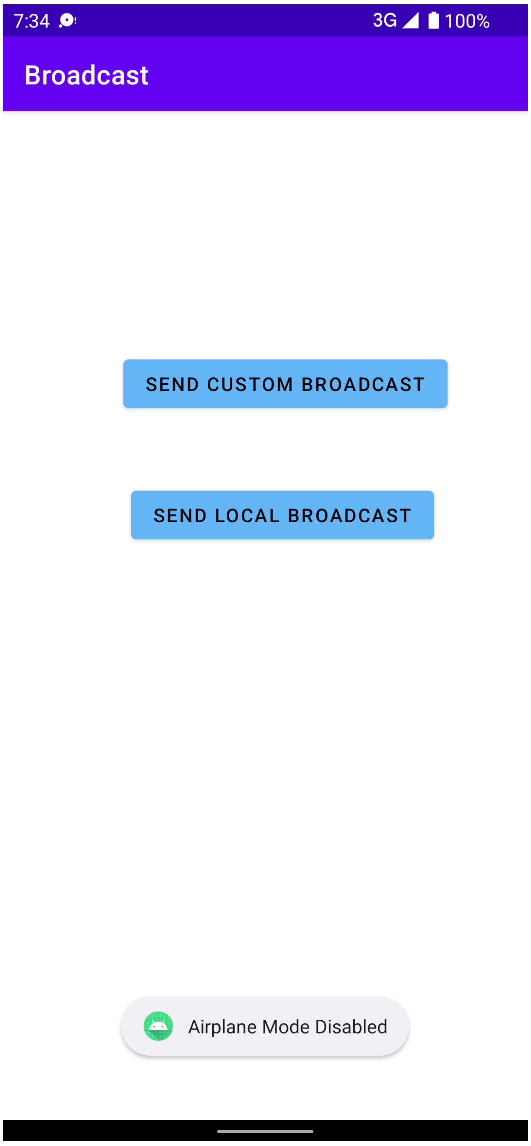
*Layout*



*Airplane Mode*

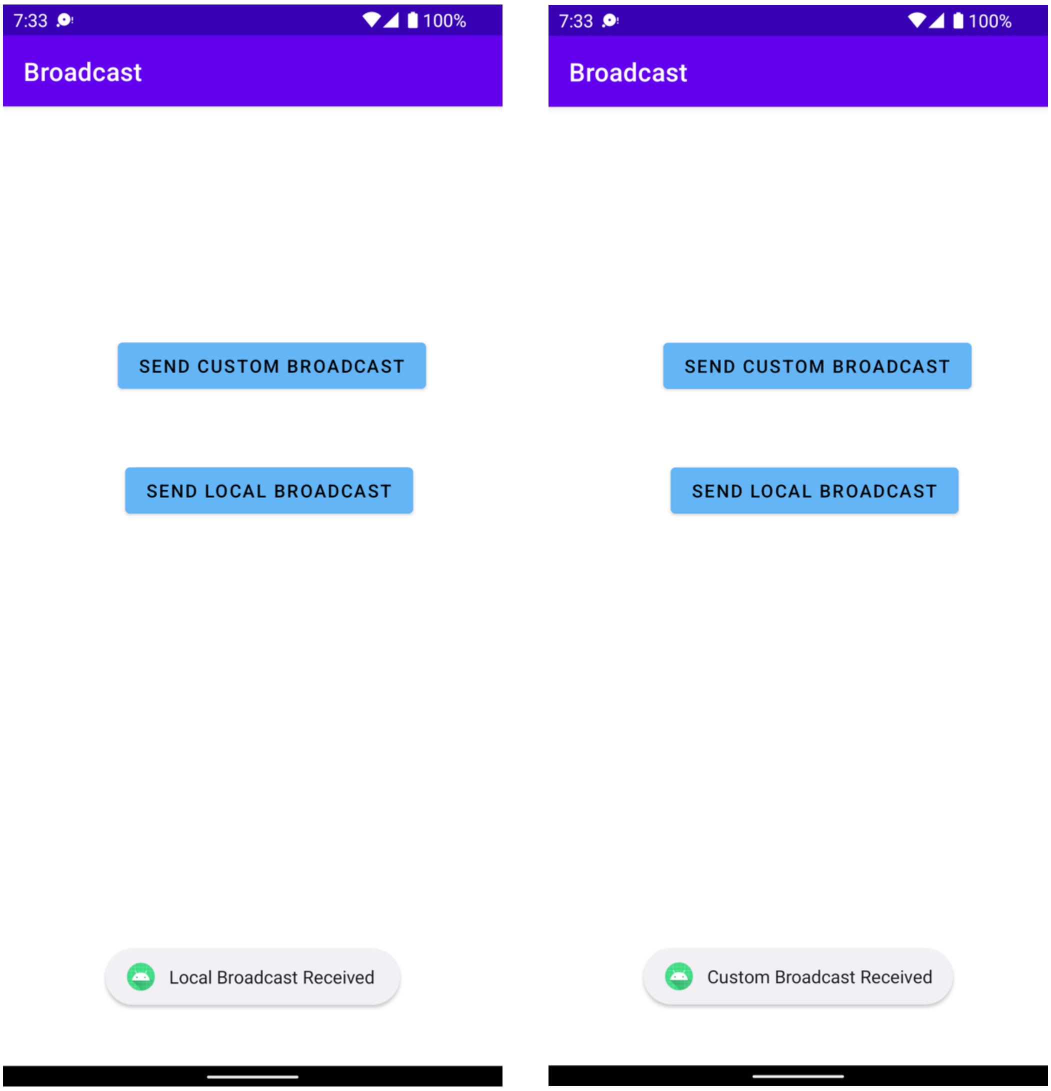


Airplane Mode Enabled



Airplane Mode Disabled

*Local Broadcast and Custom Broadcast*



Local Broadcast Received

Custom Broadcast Received

## Conclusion

I learnt what broadcast receivers are and how they operate from the lab work done above. Android component known as a broadcast receiver enables an app to reply to messages broadcast by the Android operating system or by another app. I was able to put the idea of local and global broadcast receivers into practice. Broadcast receivers may be used for many different purposes, from storing data to receiving messages on the battery's state.

# References

Android. (n.d.). *Android Studio Docs*. Retrieved 08 07, 2022, from Developers Android: https://developer.android.com/docs

Sinhal, A. (2017, April 29). *Local Broadcast, less overhead and secure in Android*. Retrieved Sept 20, 2022, from Medium: https://medium.com/android-news/local-broadcast-less-overhead-and-secure-in-android-cfa343bb05be

Gupta, M. (2019, March 4). *Broadcast Receivers: Getting Started*. Retrieved Sept 22, 2022, from Coding with Mitch: https://codingwithmitch.com/blog/broadcast-receivers-getting-started/