Exercise 9
Mobile Device Development
Android Services
Due: 11/24/2020@11:59pm

Part 1

The following example demonstrates how to use a service to download a file from the Internet based on a button click from an activity. Once done, the service notifies the activity via a broadcast receiver that the download is complete.

In this exercise you use the `IntentService` class, as this class provides automatic background processing. You must use API 29.

Create a new project called *edu.sjsu.android.receiver* with the activity called *MainActivity*.

Create the following class for the service.

```
package edu.sjsu.android.receiver;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import android.app.Activity;
import android.app.IntentService;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.os.Message;
import android.os.Messenger;
import android.util.Log;

public class DownloadService extends IntentService {

    private int result = Activity.RESULT_CANCELED;
    public static final String URL = "urlpath";
    public static final String FILENAME = "filename";
    public static final String FILEPATH = "filepath";
    public static final String RESULT = "result";
    public static final String NOTIFICATION = "edu.sjsu.android.receiver";

    public DownloadService() {
        super("DownloadService");
    }

    // will be called asynchronously by Android
    @Override
    protected void onHandleIntent(Intent intent) {
        String urlPath = intent.getStringExtra(URL);
        String fileName = intent.getStringExtra(FILENAME);
        File output = new File(Environment.getExternalStorageDirectory(),
                fileName);
        if (output.exists()) {
            output.delete();
        }

        InputStream stream = null;
        FileOutputStream fos = null;
        try {

            URL url = new URL(urlPath);
            stream = url.openConnection().getInputStream();
            InputStreamReader reader = new InputStreamReader(stream);
            fos = new FileOutputStream(output.getPath());
            int next = -1;
            while ((next = reader.read()) != -1) {
                fos.write(next);
            }
            // successfully finished
            result = Activity.RESULT_OK;
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (stream != null) {
                try {
                    stream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
            if (fos != null) {
                try {
                    fos.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
        publishResults(output.getAbsolutePath(), result);
    }
    private void publishResults(String outputPath, int result) {
        Intent intent = new Intent(NOTIFICATION);
        intent.putExtra(FILEPATH, outputPath);
        intent.putExtra(RESULT, result);
        sendBroadcast(intent);
    }
}
```

Add this class to the AndroidManifest.xml file. Also add the permission to write to external storage and to access the Internet. Set requestLegacyExternalStorage to be true. The resulting AndroidManifest.xml file should look similar to the following listing.

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

    <uses-sdk android:minSdkVersion="19" android:targetSdkVersion="29" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme"
        android:requestLegacyExternalStorage="true">
        <activity
            android:name="edu.sjsu.android.receiver.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <service android:name="edu.sjsu.android.receiver.DownloadService" >
        </service>
    </application>
</manifest>
```

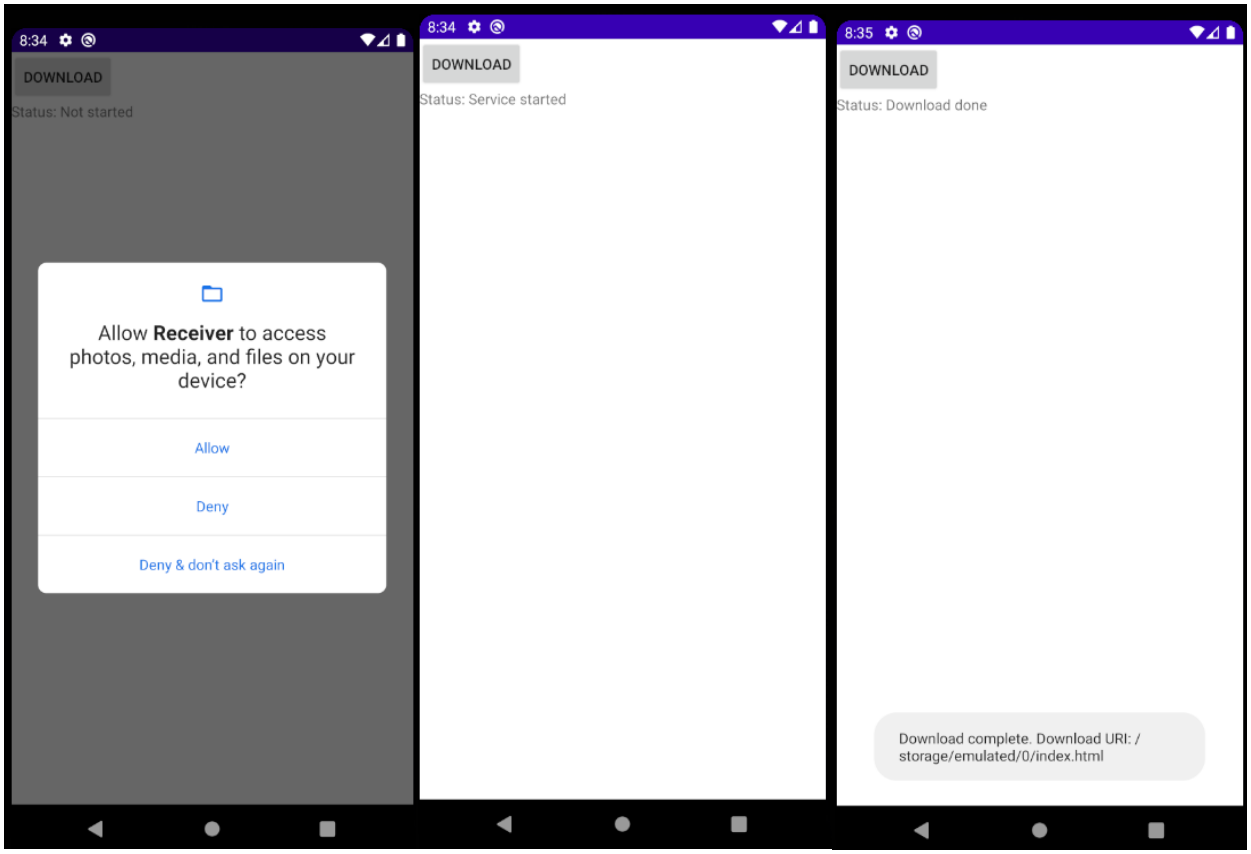
Change the layout file of your activity to the following.

```
<?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" >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="Download" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Status: " />
        <TextView
            android:id="@+id/status"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Not started" />
    </LinearLayout>
</LinearLayout>
```

Change `MainActivity` to the following.

```
package edu.sjsu.android.receiver;

import android.app.Activity;
import android.content.*;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

public class MainActivity extends Activity {

    private TextView textView;
    private BroadcastReceiver receiver = new BroadcastReceiver() {

        @Override
        public void onReceive(Context context, Intent intent) {
            Bundle bundle = intent.getExtras();
            if (bundle != null) {
                String string = bundle.getString(DownloadService.FILEPATH);
                int resultCode = bundle.getInt(DownloadService.RESULT);
                if (resultCode == RESULT_OK) {
                    Toast.makeText(MainActivity.this,
                            "Download complete. Download URI: " + string,
                            Toast.LENGTH_LONG).show();
                    textView.setText("Download done");
                } else {
                    Toast.makeText(MainActivity.this, "Download failed",
                            Toast.LENGTH_LONG).show();
                    textView.setText("Download failed");
                }
            }
        }
    };
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView = (TextView) findViewById(R.id.status);
    }

    @Override
    protected void onResume() {
        super.onResume();
        registerReceiver(receiver, new IntentFilter(
                DownloadService.NOTIFICATION));
    }
    @Override
    protected void onPause() {
        super.onPause();
        unregisterReceiver(receiver);
    }
    public void onClick(View view) {
        if (checkPermission()) {
            Intent intent = new Intent(this, DownloadService.class);
            // add infos for the service which file to download and where to
store
            intent.putExtra(DownloadService.FILENAME, "index.html");
            intent.putExtra(DownloadService.URL,
                    " https://www.sjsu.edu/cs/index.html");
            startService(intent);
            textView.setText("Service started");
        } else {
            ActivityCompat.requestPermissions(MainActivity.this,
                    new
String[]{android.Manifest.permission.WRITE_EXTERNAL_STORAGE}, 100);
        }
    }

    private boolean checkPermission() {
        return ContextCompat.checkSelfPermission(MainActivity.this,
                android.Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED;
    }
}
```

To run your example, press the button, and the download should be performed by the service. Once done, the user interface is updated and a `Toast` with the file name is shown.



Status: Not started

Allow **Receiver** to access photos, media, and files on your device?

Status: Service started

Status: Download done

Download complete. Download URI: /storage/emulated/0/index.html

Part 2

In this exercise you define a broadcast receiver which listens to telephone state changes. If the phone receives a phone call, then our receiver will be notified and log a message.

Create a new project called `edu.sjsu.android.receiver.phone`. Also create an activity.

TIP: Remember that your receiver is only called if the user started it once. This requires an activity.

Implement receiver for the phone event

Create the `MyPhoneReceiver` class.

```
package de.vogella.android.receiver.phone;
import android.content.*;
import android.os.Bundle;
import android.telephony.TelephonyManager;
import android.util.Log;

public class MyPhoneReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        Bundle extras = intent.getExtras();
        if (extras != null) {
            String state = extras.getString(TelephonyManager.EXTRA_STATE);
            Log.w("MY_DEBUG_TAG", state);
            if (state.equals(TelephonyManager.EXTRA_STATE_RINGING)) {
                String phoneNumber = extras
                        .getString(TelephonyManager.EXTRA_INCOMING_NUMBER);
                Log.w("MY_DEBUG_TAG", phoneNumber);
            }
        }
    }
}
```

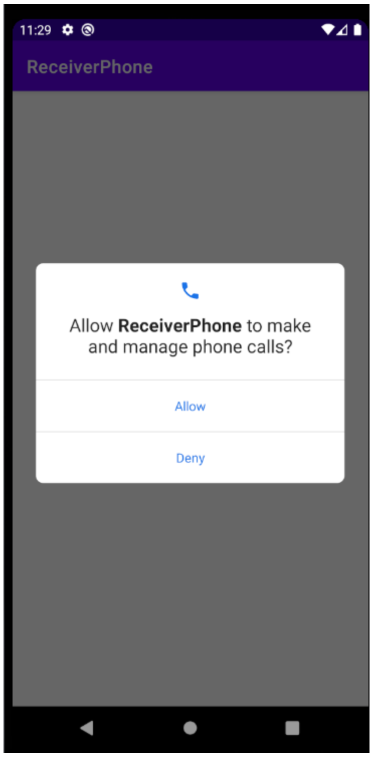
Request permission

Add the `android.permission.READ_PHONE_STATE` and `android.permission.READ_CALL_LOG` permission to your manifest file which allow you to listen to state changes in your receiver and read call log information. Also Register your receiver in your manifest file. The resulting manifest should be similar to the following listing.

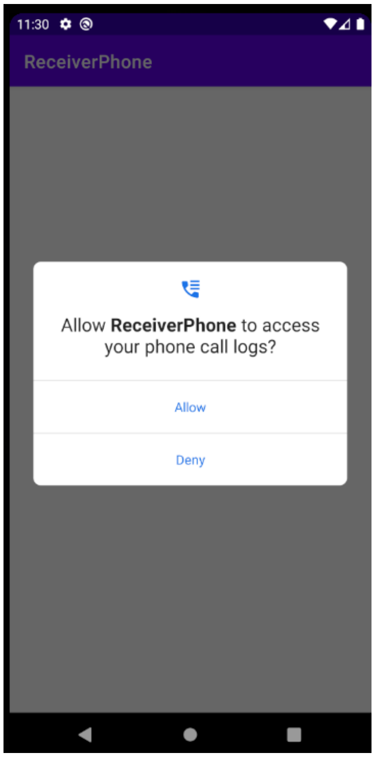
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="de.vogella.android.receiver.phone"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.READ_CALL_LOG" />
    <application
        android:icon="@drawable/icon"
        android:label="@string/app_name" >
           <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

        <receiver android:name="MyPhoneReceiver" >
            <intent-filter>
                <action android:name="android.intent.action.PHONE_STATE" >
                </action>
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

Also, in your MainActivity, request runtime permission for `READ_PHONE_STATE` and `READ_CALL_LOG`, similar to what you did in part 1 for `WRITE_EXTERNAL_STORAGE` permission. Note that the second parameter of `ActivityCompat.requestPermissions` is an array of string, which can request multiple permissions.

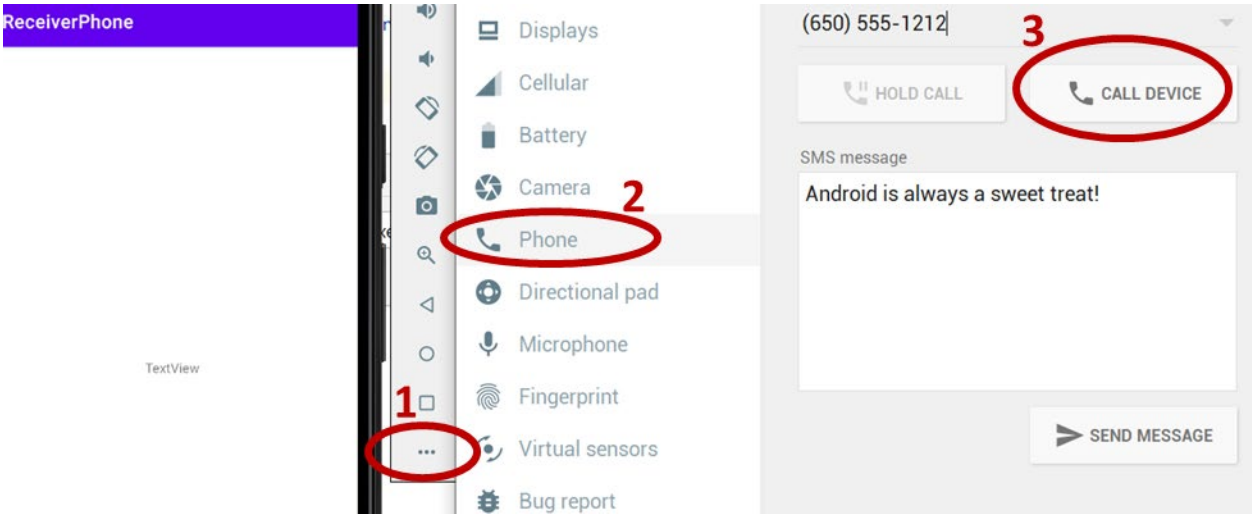
Install your application and simulate a phone call via the emulator controls. Validate that your receiver is called and logs a message to the `LogCat` view.



Allow **ReceiverPhone** to make and manage phone calls?



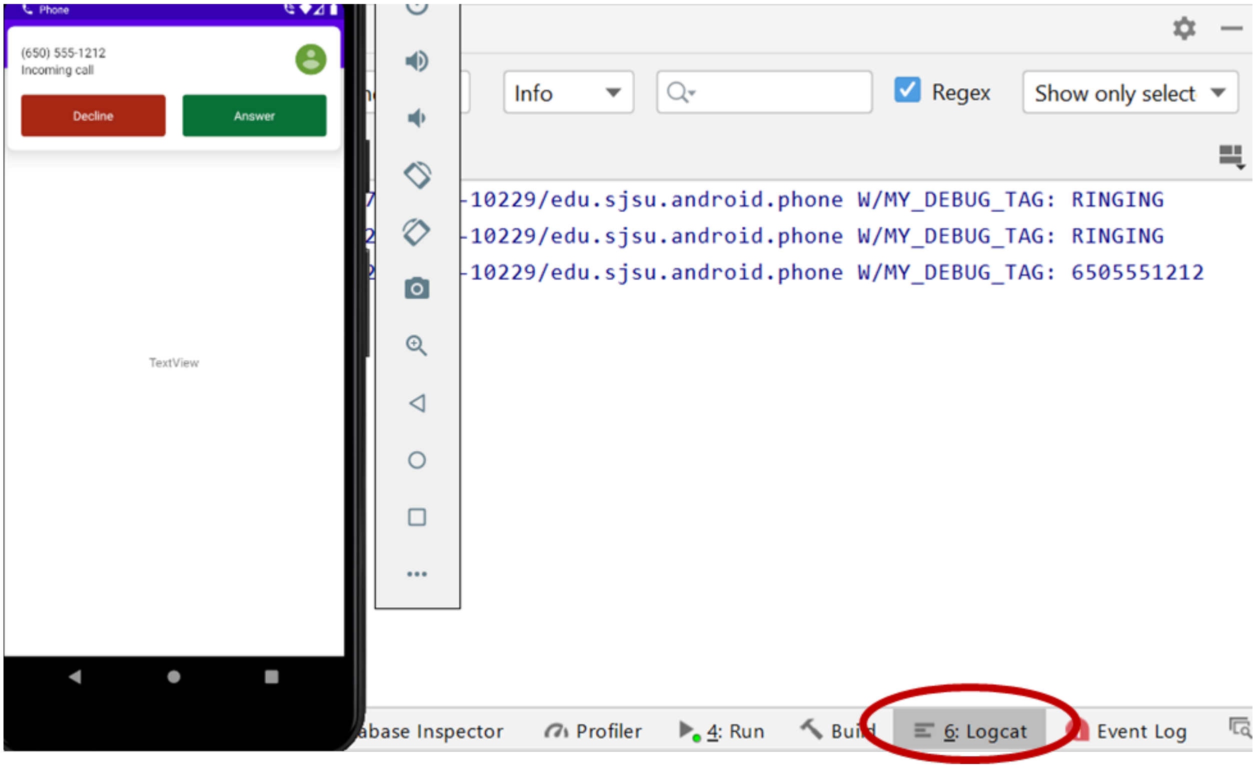
Allow **ReceiverPhone** to access your phone call logs?



(650) 555-1212

SMS message

Android is always a sweet treat!



```
-10229/edu.sjsu.android.phone W/MY_DEBUG_TAG: RINGING
-10229/edu.sjsu.android.phone W/MY_DEBUG_TAG: RINGING
-10229/edu.sjsu.android.phone W/MY_DEBUG_TAG: 6505551212
```

Submission

1. Push your project directory along with the source to remote bitbucket repository by the due date.
2. Invite and share your project repository the Grader (yan.chen01@sjsu.edu) and Instructor (ramin.moazeni@sjsu.edu).
3. Submit a Readme.pdf to Canvas including your name, repository access link, instructions to run your program (if any), snapshot of your running program
4. Your project directory will be graded according to the state your project directory was in at due time when fetched.