



Awsome Media Player

MOBILE APPLICATION

Description.

The objective of the task was prepare functional MP3 player with basic control menu such as Play, Pause and Stop.

Make the Media Player play music from device and to make the media player continue playing when the application is in the background

During development, we had a couple of troubles, which we describe in detail in this documentation.

1) Play music from device

Problems arose when figuring out how to do this as we searched the web for answers we found some different ways to do this task but we finally decided to use a file method witch goes true all folders and gets the name of all files but this for some reason does not work as intended. The intent was to read music from SD card but it reads music from internal storage instead (please give reason for this if you know). We used some time to figure out that in android 5.0 we had to add permission to manifest.

```
private ArrayList<File> fileList = new ArrayList<File>();

public ArrayList<File> getfile(File dir) {

    File listFile[] = dir.listFiles();
    if (listFile != null && listFile.length > 0)
    {
        //checs all files in folder
        for (int i = 0; i < listFile.length; i++)
        {
            if (listFile[i].isDirectory()) {
                fileList.add(listFile[i]);
                getfile(listFile[i]); }
            else {
                if (listFile[i].getName().endsWith(".mp3")
                    || listFile[i].getName().endsWith(".mp4")
                    || listFile[i].getName().endsWith(".Mp3")
                    || listFile[i].getName().endsWith(".mp3"))
                {
                    fileList.add(listFile[i]);
                } } } }
    return fileList; }
```

2) Play in background

we have some trouble with this as the song continues playing when app is not in the front but after the song is complete it stops we tried to implement some code for starting a new song but as we do not have access to the list from the activity class we could not see how to do this but would like to have some clue as to how we could do this for future coding.

class BackgroundAudioService extends Service implements OnCompletionListener

```
{ @Override

    public IBinder onBind(Intent intent) {

        return null;

    }

    @Override

    public void onCreate()

    { root = new File(Environment.getExternalStorageDirectory().getAbsolutePath());

      list = handler.getfile(root);

      mediaPlayer = MediaPlayer.create(this, Uri.parse(list.get(o).getPath().toString()));

      mediaPlayer.setOnCompletionListener(this);  }

    @Override

    public int onStartCommand(Intent intent, int flags, int startId) {

        if (!mediaPlayer.isPlaying()) {

            mediaPlayer.start();    }

        return START_STICKY;  }

    public void onDestroy() {

        if (mediaPlayer.isPlaying()) {

            mediaPlayer.stop();    }

        mediaPlayer.release();  }

    public void onCompletion(MediaPlayer _mediaPlayer) {

        stopSelf();  }
```