G54MDP Coursework – Android Music Player App

Summary

You are required to build an Android application that implements the functionality of a simple music player. This is an assessed exercise and will account for 50% of your final module mark. This is an individual coursework, and your submission must be entirely your own work – please pay particular attention to the final section of this document regarding plagiarism.

Your application should be submitted no later than:

- **4pm on Wednesday 9th December 2015**

Submission should be made electronically via moodle (http://moodle.nottingham.ac.uk). Standard penalties of 5% per working day will be applied to late submissions.

Your application should be submitted as a .zip or .tar.gz file containing all relevant source code, configuration and related files, and a compiled .apk file – i.e. the contents of the directory containing your Android Studio project. Do not submit RAR files.

Specification

You should create an application with the functionality of a simple music player for Android, which allows users to select from a number of music files stored on the SD card storage of the device to be played, and allows the music to continue to play in the background while the user performs other tasks.

At the minimum your application should support:

- Displaying a list of music files in a specific folder on the SD card
- Playing a selected music file
- Continued playback in the background
- Allowing the user to stop or pause playback

You must implement a **bound Service** to handle the music-playing element of the application, as this is a long-running task and the user can be expected to leave the initial activity to perform other tasks. You should think carefully about the relationship between the Activities and Service in your application, and how these should be used appropriately to perform the task. There is no requirement that your service will be used remotely.

You should consider how you would integrate a ContentProvider component, either using an existing standard SDK component or by developing a new

ContentProvider, to retrieve information about the music already stored on the device.

It is left up to you to decide how best to design and implement Activities for selecting and controlling the music playback.

Your application must be written in Java and make use of the Android SDK. There are no requirements to target a specific Android API version, however you should assume that your application would be tested on a device running Android 4.4.2.

You should consider the following when implementing your application:

- Decomposition of the task into logical, discrete Activity and Service components.

- Appropriate use of Activities, Intents and appreciation of the Activity life-cycle

- Appropriate use of Widgets and ViewGroups for layouts that support devices of differing screen sizes and resolutions

- Appropriate use of Services, Notifications and appreciation of the Service life-cycle

- Appropriate use of ContentProviders, Contract classes, Permissions and Android data storage mechanisms

- Your application should have appropriate comments and variable / class names, so that a reader can easily understand how it works at the code level

Adding further additional functionality to the application is encouraged, for example adding support for queuing multiple files to be played, or more sophisticated controls, but please note that this coursework will be primarily assessed on the above specification, and further additions must not detract from the required functionality given above.

Assessment Criteria

	Marks Available
Basic Application Functionality	
The application allows the user to play and stop music files	20
The application runs appropriately in the background	20
Application Structure and Implementation	
Activities appropriately support the task	5
Appropriate use of Views and Layouts	5
Appropriate use of a Service	10
Handling of the Service life-cycle	5
Appropriate use of notifications	5
Appropriate use of Android data storage and ContentProviders	10
Programming style	
The application is easy to understand, with comments explaining each part of the code, correct formatting, and meaningful variable names	10
Extension	
Appropriate additional functionality not in the specification	10
Total	100

Each element of your coursework will be assessed against the standard criteria:

https://workspace.nottingham.ac.uk/display/CompSci/Marking+Criteria

The following areas will be taken into account for each part of the assessment:

- Demonstrating knowledge of the area
- Quality of the concept, including appropriateness and novelty
- Quality of the technological design, including appropriate use of software design concepts, and appropriate good coding practice (abstraction, commenting, naming)
- Quality of the realization, including how well it works and elaborations over and above the basic requirements
- Including all of the above aspects, clarity of structure, quality of argument / evidence, and insight / novelty

Hints

As always, the best source of information when developing your application is the developers reference, in particular the following sections:

http://developer.android.com/guide/topics/ui/declaring-layout.html

http://developer.android.com/guide/topics/ui/controls.html

http://developer.android.com/guide/components/activities.html

http://developer.android.com/guide/topics/data/data-storage.html

http://developer.android.com/guide/topics/providers/content-providers.html

http://developer.android.com/guide/components/services.html

In particular, the following guide to using the MediaPlayer for playing sound files will be useful, and indeed contains many code fragments for appropriate components. Please note however that solutions heavily derived from this tutorial (or others) will not sufficiently demonstrate your own understanding, and will therefore only achieve low marks.

http://developer.android.com/guide/topics/media/mediaplayer.html

Plagiarism

N.B. Use of third party assets (images, example code, libraries etc) MUST be credited or referenced, and you MUST be able to demonstrate that they are available under a license that allows their reuse.

Making significant use of tutorial code AND referencing it will merely result in a lower mark. Failing to attribute to the original source will result in a mark of zero.

Copying code from other students, from previous students, from any other source, or soliciting code from online sources and submitting it as your own is plagiarism and will be penalized as such – potentially resulting in failure of coursework, module or degree.

All submissions are checked using both plagiarism detection software and manually for signs of cheating. If you have any doubts then please ask.