COMP3018 Coursework 1: Multiple Activity Audiobook player

Summary

In this exercise you are required to build an Android Audiobook player application. This is an assessed exercise and will account for **25% 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 section of this document regarding plagiarism. This document sets out general requirements and broad instructions for developing the application.

Your application should be submitted no later than:

- **3pm on Thursday the 21/11/24**

Submissions should be made electronically via Moodle. Standard penalties of 5% per working day will be applied to late submissions.

Your application should be submitted as a .zip or 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 after a clean rebuild.

Specification

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

Your application must consist of **three** Activity components:

- An activity displaying all the audiobooks and bookmarks available on the device which can be played. (**Audiobook list**)
    - Displays and allows the user to select from and play audiobooks files from the /sdcard/Music
    - Allows the user to click a bookmark, and the audiobook associated with the bookmark will start playing at the displayed timestamp

- An activity that has the play / pause / stop / skip buttons alongside a display of the current progress of the currently playing audiobook, and the current playback speed (**Audiobook player**)
    - Allows the user to stop or pause playback, and select the next or previous Audiobook
    - Displays the current progress of the playback (i.e. the elapsed time)
    - Allows the user to create a bookmark at the specified time
- An activity where the settings which control playback speed and background color can be set (**Settings**).
    - Specify the value for playback speed
    - Specify the background colour

And :

- A *Service* that provides continued playback in the background
- A *Notification* that that is displayed while the Audiobook is playing

You must implement a **Service** to handle the AudioBook-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 Activity 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, i.e. you do not need to use *AIDL.*

A simple Audiobook class is provided that wraps a basic Media Player object for loading and playing an mp3 file. You must use this class to play the mp3 files, and you must not edit this file.

Your application will be tested on an emulated Pixel 5 (1080x1920: 420dpi) device running Android 14 - Upside down cake, Android API version 34. **You must ensure that your work functions appropriately on this device, as this is how it will be marked.** The range of different ways that the user might engage with the different elements of the task while also intermittently or for extended periods engaging with other tasks, and how well the application supports this, is also part of the assessment criteria.

You should consider the following when implementing your application:

- Decomposition of the task into discrete Activity components
- Appropriate use of Intents, communication between Activities and appreciation of the Activity life-cycle with a focus on persistence for coherent user experience
- Appropriate use of Widgets and ViewGroups for layouts that support devices of differing screen sizes and resolutions
- Your application is expected to have appropriate comments and variable / class names, so that a reader can easily understand how it works if necessary.
- Appropriate use of Activities, Intents and appreciation of the Activity life-cycle
- Appropriate use of Services, Notifications and management of the Service life-cycle
- Appropriate communication between components
- Appropriate architectural best practice

Assessment Criteria

As this is a constrained exercise marks are awarded for achieving specific functionality as follows. For these elements either 0 or full marks are awarded as appropriate. Additional marks are awarded for consideration of good practice. There are no additional marks available for additional functionality in this exercise:

	Marks
Basic Functionality	
The application has an Activity that allows the user to view the songs / bookmarks on the device (AudioBook list)	1
The application has an Activity that allows the user to set the playback speed and background colour (settings)	1
The application has an Activity that allows the user to play, pause and stop a playing audiobook, as well as cycle through books (skip track)	1
The Activity displays the current progress of the playback and the playback speed	2
The currently chosen playback speed is passed to and displayed by the currently	2

playing activity (audiobook player).	
The Activity maintains speed / playing AudioBook throughout its expected lifecycle (e.g. when phone is rotated or when another application is in use, i.e. receiving a phone call)	2
The application has a Service that handles playing the track when another application is in foreground use	2
The bookmarks load the specified audiobook at the specified time	2
The application displays a notification when a AudioBook is playing	2
Best Practice (and adherence to specification)	
The application supports appropriate navigation, and appropriate communication between and management of components. alongside good architectural design.	15
Total	30

Plagiarism

N.B. Use of third party assets (tutorials, images, example code, libraries etc.) MUST be credited and 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 while referencing it is poor academic practice, and will result in a lower mark that reflects the significance of your own original contribution.

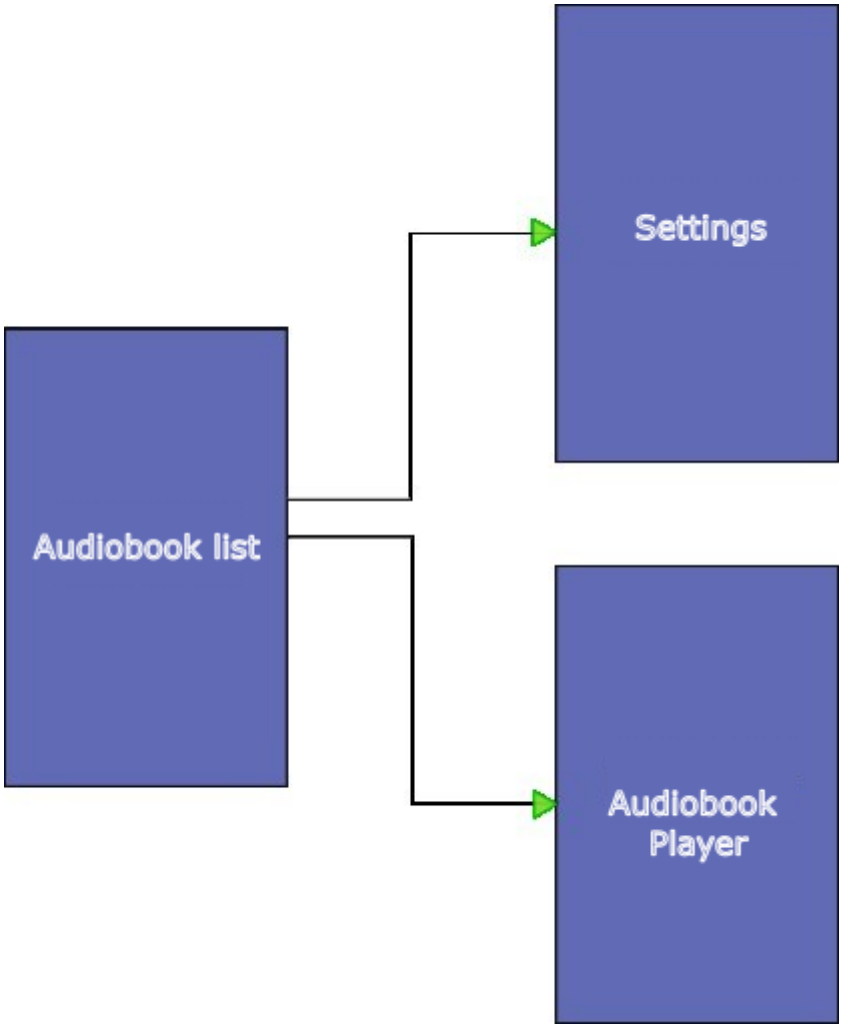
Copying code from other students, from previous students, from any other source, or soliciting code from online sources and submitting it as your own is plagiarism and will be penalized as such. FAILING TO ATTRIBUTE a source will result in a mark of zero – and can potentially result in failure of coursework, 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.

Instructions

Note that this coursework aims to serve as an assessment of the Activities and Services material covered in lectures and used in the lab exercises – as such try to think about how to make use of the concepts covered in those. It is a mistake to begin by trying to find out how to build a generic AudioBook player app.

The Task (**This architecture is mandatory**)



Persistence - Important

Begin by creating a new application in Android Studio as usual. The state of the components, for example the current playback speed and background colour are not persisted. You should handle the activity lifecycle for your activities to ensure a coherent user experience if, for example, the device is rotated. **From the user perspective, the currently playing Audiobook must continue playing when moving between activities or selecting options such as playback speed.**

Getting started

AudiobookPlayer is a simplistic wrapper for an Android MediaPlayer object. MediaPlayer has its own internal thread for actually doing the work of playing an mp3, so there is no need to spawn a new thread to contain it. There is also no need to asynchronously load an mp3 in a separate thread. It is worth adding the Audiobook class directly to an Activity and controlling it directly with buttons to make sure you understand how it works before moving it into a Service. **There is a minimum working example of an audiobook player on the moodle website. This is just to ensure audio is working.**

To transfer mp3 files onto the directory (/storage/self/primary/Music/) of the emulator you can drag them from your computer onto the music folder in the "Device Explorer" tab in android studio.

If you don't have access to any Audiobooks in mp3 format, a variety of royalty free / creative commons licensed files are available here https://freemusicarchive.org/home