CIS 3515 Assignment 9

Instructions: Create a branch of your BookShelf app and update it with new features.

Your application will allow a user to play the audio from any book in the collection available on the web service. To facilitate the audio feature, you will use a library linked in the appendix below. Your goal will be to integrate the library, and thereafter provide controls and feedback to the user when playing an audiobook.

1. Refactor your Book class to add a new field:

    duration: int – the length of the book in seconds

    You will find this value available in the JSON book objects received from the web service API.

2. Download the library file linked in the appendix below. The library will provide you with a service implementation, **edu.temple.audiobookplayer.AudiobookService**, capable of playing an audiobook hosted on the web service by simply providing the ID for the book you wish to play. NOTE you should **not** use the method in the service that allows you to play an audiobook by using a file. Only use the method that asks for the book ID. The service will stream the book automatically.

3. Add the library to your project as a dependency by following the instructions here: https://developer.android.com/studio/projects/android-library#AddDependency. This will allow you to access AudiobookService. **Do NOT download the source code of the service to add it to your project manually. Be sure to add it as a library!**

4. Once added, modify your app to add the following features:

    1. Allow a user to play the book they are currently viewing

    2. Activity must display the title of the currently playing book with a “Now Playing” header or prefix

    3. Allow a user to pause the currently playing book

    4. Allow a user to stop the currently playing book

    5. Show the progress of the currently playing book (between 0 and 100%) using a SeekBar

    6. Allow the user to move forward and backward within a book (between 0 and 100%) by dragging the position of the same SeekBar that shows book progress

5. Audiobook controls will be split across two components: The **Play** button will be placed in your BookDetailsFragment’s layout, while all other controls (Pause button, Stop button, SeekBar) should be placed in your activity. Note that even though the Play button is located inside the BookDetailsFragment, communication with the service should only be performed in your main

activity. That is, the commands to play, pause, and stop an audiobook should be issued from the activity, even though the Play button is located in a fragment. No fragment should have access to the binder object or communicate with the service directly.

6. There should be no interruption in audio playback if the activity is restarted. That is, if the user, for example, rotates the device and forces a configuration change while a book is playing, the audio should continue to play uninterrupted while the activity restarts, and the user must still be able to control the book once the activity is restarted.

Appendix

Source code, the aar (library) file, and instructions on using the library can be found here:
https://github.com/karlmorris/AudiobookPlayer

Rubric

Integrated AudioBookService	10%
Properly integrate controls (Play, Stop, Pause) across activity and BookDetailsFragment	10%
Play function initiated in fragment is routed through activity before sent to service	10%
SeekBar progress is associated with book duration	10%
SeekBar updates as book plays to show current progress	10%
SeekBar controls book progress when user moves slider	20%
Book continues playing when activity is restarted	20%
Controls continue to function and *Now Playing* shows correct book when activity is restarted	10%