CIS 3515 Assignment 7

Instructions: Create a new application called **AudioBookPlayer**, that will take shape over the next 4 labs.

Your application will incorporate a dynamic (responsive) UI, that will display a **single fragment** at a time if the application is being run **in portrait mode**, but must display **two fragments simultaneously** if being run **in landscape mode**.

1. Create a **Book** class in your application that contains the following fields for a book:

    1. title: String – The title of the book

    2. author: String – The author of the book

2. Create a **BookList** class in your application that is used to store a set of Book objects. Internally, your BookList class should organize its book objects using an **ArrayList**. Your BookList class should provide the standard set of collection-like operations, such as:

    1. add(Book) : void

    2. remove(Book) : void

    3. get(int) : Book

    4. size(): int

    5. Any other function you might find useful

3. In your activity:

    1. Create an instance of your BookList class, and populate it with a random assortment of Book objects. Simply pick some random book titles, real or made up – you should have at least 10 books. You can store and fetch that information from a string resource, or simply create a helper method or helper class to generate the books for you – all that matters is that in the end you have an BookList object that contains at least 10 Book objects, where each Book has a title and author value.

4. Have (at least) 2 layout files for your activity for the following scenarios:

    1. portrait: this will contain a single display container (this could be the default)

    2. landscape: this will contain 2 display containers

5. Create two fragments: BookListFragment, and BookDetailsFragment.

   1. BookListFragment will contain a RecyclerView that, when provided with a BookList object, will display the book title and book author for each book object that the BookList contains. The title and author should each be displayed in their own TextView – that is, each item in the RecyclerView will contain 2 TextViews, one showing the book title, and the other showing the book author. The fragment should have a factory method (**newInstance(booklist: BookList)**) that creates a fragment using the provided books to set up the fragment’s initial state

   2. BookDetailsFragment will have the ability to display a book title and author when a book is clicked in the BookListFragment. It will display the title in a TextView with a relatively large font size, and the author in a TextView with a slightly smaller font size.

6. Your activity will use the List-Detail pattern when determining when to display a certain fragment.

   1. If the activity is using a layout that contains a single container (portrait mode on a small screen), it should display an instance of the BookListFragment at startup. When the user clicks a book from the BookListFragment, an instance of BookDetailsFragment should be launched to replace the BookListFragment, displaying information on the book that was clicked. The user should then be able to reverse this operation (by clicking the back button) and select a new book.

   2. If the activity is using a layout that contains two containers, then one container is used to display a **BookListFragment**, and the other to display a **BookDetailsFragment**. Clicking a book title in the BookListFragment should have the BooksDetailsFragment display the book title, but it should not load a new instance for each click. Instead, a single instance of BookDetailsFragment should be loaded, and it should change the book that it displays whenever the user clicks a new book from the BookListFragment.

   3. If the user is viewing a selected book in Portrait mode (on a small screen), and then switches the device to Landscape mode, the previously selected book should still be shown. The same is true if the user switches from Landscape to Portrait after selecting a book – the selected book should be visible first instead of the list of books.

   4. Your application must incorporate a ViewModel to provide state consistency across fragments/activity.

7. **COMMIT and PUSH to GitHub FREQUENTLY! Do NOT perform a single commit and push at the end of development. There will be a 20% penalty if there is not a proper commit trail for your project. Upload your GitHub link to Canvas.**

Rubric

Application has layouts for portrait and landscape configurations	10%
BookListFragment has a factory method that can create new fragment instances	10%
BookListFragment has RecyclerView that shows both title and author in their own TextViews	10%
BookListFragment properly communicates selected book using ViewModel	15%
Proper master-detail implementation when in small-portrait mode (new BookDetailsFragment created when book is clicked which replaces BookListFragment)	15%
Proper master-detail implementation when in large-portrait or landscape mode (Single instance of BookDetailsFragment being updated when book is clicked)	20%
Selected book and fragment display is remembered across orientation changes	20%