CIS 3515 Assignment Worksheet 7

Instructions: Create a new application, **BookCase**, that will take shape over the next few labs.

Your application will incorporate a dynamic (responsive) UI, that will display a **single fragment** at a time if the application is being run **on a small screen (cell phone) in portrait mode**, but must display **two fragments simultaneously** if being run **on a larger device (tablet)** or if being run **in landscape mode, regardless of screen size**.

1. In your activity, create an **ArrayList<String>** that contains a list of arbitrary books (just pick some random books from titles for your array list values – you should have at least 10 books). You can load your books from a resource if you would like, or simply hard code the values.

2. Create 3 layout files for the following scenarios:

    1. Small-portrait: this will contain a single display pane

    2. Small-landscape: this will contain 2 display panes

    3. Tablet: this will contain 2 display panes

3. Create two fragments: BookListFragment, and BookDetailsFragment.

    1. BookListFragment will contain an ListView that, when provided with an arraylist of books, will display each book title in its ListView. When the user clicks one of the titles in the ListView, the fragment should invoke a method in its parent with the book **index** that was clicked.

    2. BookDetailsFragment will, for now, have the ability to display a book title when provided with one. It will have a single public API method, **displayBook(String title)**, which will display the book title in a TextView with a large font size whenever the method is invoked.

4. When your application is displaying a single pane, it should use a **ViewPager** (https://developer.android.com/training/animation/screen-slide – you do not need to create any animations) that will allow users to swipe through a collection of **BookDetailsFragment**s inside your array list.

    1. Your activity must create an instance of **BookDetailsFragment** for each book inside the ArrayList. Each fragment should be initialized using a Factory Method (**newInstance(**...**)**) to provide it the book information it should display.

5. When your application is displaying two panes, it should use the master-detail display pattern, where the *left* pane is used to display a **BookListFragment**, and the *right* pane is used to display a **BookDetailsFragment**.

    1. 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 BookListFragment should be loaded, and it should then change the book that is displayed whenever the user clicks a new book from the BookListFragment.

6. **You have two weeks to complete this assignment. Commit your project to GitHub and upload your link to Canvas.**

   NOTE

   While this application only explicitly calls for the creation of 2 fragment classes, you might actually find it useful to create 3. Why? Because the application as described will require that you activity contain a lot of code that goes beyond simply coordinating fragments. It will also be responsible for managing the view pager, etc.

   To address the potential complexities of this issue, when you application is in “single pane” mode and calls for using a ViewPager, etc, you may choose to employ another fragment that will serve as a container for the ViewPager (let’s call it ViewPagerFragment) that will deal with all the considerations found therein. If you take this approach, then your activity’s code becomes much more simplified since its responsibilities will only be to initialize the array list of books, and to coordinate fragment creation and communication.

   Inside your ViewPagerFragment, when you need to initialize a FragmentStatePagerAdapter that calls for a FragmentManager, you can get a reference to a fragment’s own fragment manager by calling getChildFragmentManager() (https://developer.android.com/reference/android/app/Fragment.html#getChildFragmentManager()) on the parent fragment.