CIS 3515 Assignment Worksheet 8

Instructions: **Create a new GitHub Branch** and update your **BookShelf** app with new features.

You application will use a Web Service API (See Appendix below) to fetch book information, and display that information in its various modes. You application will allow a user to retrieve all available book titles available via the API, as well as allow a user to search for a subset of titles that match a user-provided search string.

Additionally, you will replace the current method off representing a book (using a HashMap) with an actual Book class. The book class is defined below.

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

    1. id: int – A system provided ID for the book

    2. title: String – The title of the book

    3. author: String – The author of the book

    4. coverURL: String – A URL to the image representing the book cover

    All previous references to HashMap<String, String> should be replaced by the Book class.

2. Use either an ArrayList (ArrayList<Book>) or your own collection class to keep a collection of book objects.

3. Modify **BookDetailsFragment** to accept a Book object instead of a HashMap, and have it display the book cover image, the book title, and the author.

4. In your activity, add a search function (An EditText to allow the user to enter a search term, and a Button to perform the search).

    1. When a user performs a search, they will receive a subset of books which may include zero, some, or all available books.

    2. A user can search for books based on title or author

    3. When a search result is returned, the collection of books being displayed to the user in the BookListFragment should be updated to only show the books returned.

    4. When a user performs a search, the application must always display the BookListFragment showing the results returned by the search. Therefore, if the application is in single-pane mode, and is not currently displaying the BookListFragment, it should remove the BookDetailsFragment and show the list of books in the BookListFragment. If it is already showing the BookListFragment, or it is in two-pane mode, it doesn't have to change the fragment(s); it should just update the BookListFragment that is already being displayed.

    5. Once the list of books have been searched for and the display updated, that subset of books must always be maintained by the application until the user performs a new search, even if the activity is restarted. Utilize the instance state feature of an activity to retain the current book collection.

    6. Similarly, whenever a single book has been selected from the list of displayed books, that book should remain selected if the activity is restarted.

5. When queried, the API will return all available books that match the user's search string. The returned titles should be used to populate the collection of books (requirement 2), and allow the user to browse

    1. As mentioned before, if the activity is reloaded after a search is preformed (because of a configuration change, such as rotating the device), then the activity should not forget the books, nor should it perform another search. Instead, it should retain the list of books from the previous search.

6. **Push your project to GitHub and upload the link to the new repository branch to Canvas.**

CONSIDERATIONS

1) Whenever you update the data set (add or remove elements) that an Adapter uses to provide views to an AdapterView, you must call notifyDataSetChanged() on the adapter for the AdapterView to be updated and show the views matching the new data.

https://developer.android.com/reference/android/widget/BaseAdapter#notifyDataSetChanged()

Here is a simple blog post describing the proper use of the method: https://androidadapternotifiydatasetchanged.blogspot.com/2013/02/android-notifydatasetchanged.html

2) When placing data inside bundles to use as arguments for fragments, you might notice that *most* of the setter methods are for primitive data types. When you need to pass non-primitive data types (which may be the case in this app since you might end up passing around Book objects and collections), you need to ensure that the class whose objects you intend to pass implements one of the compatible interfaces, such as Parcelable (https://developer.android.com/reference/android/os/Parcelable), or Serializable (https://developer.android.com/reference/java/io/Serializable). When you implement one of these interfaces, you can use bundle setter methods to place objects of those class types inside a bundle object. Eg:

```
Bundle bundle = new Bundle();
bundle.putParcelable("myBook", book);
```

or

```
Bundle bundle = new Bundle();
ArrayList<Book> books = getBooks();
bundle.putParcelableArrayList("myBooks", books);
```

Appendix

BookCase API

Get a subset of books based on search string in JSON format:

https://kamorris.com/lab/abp/booksearch.php?search=***search_string***

Rubric

Replaced all ArrayList<HashMap> references to ArrayList<Book> (or custom collection class)	15%
Allows user to perform a search to receive a subset of books	25%
Properly updates displays (ListView in BookListFragment) with new books after search is performed	20%
Once retrieved, a list of books is retained if the activity is restarted, until the user performs another search	25%
Once selected, application maintains and shows selected book when activity is restarted	15%