

Student's Name: Grace McLeod (and the rest of group two)

20 points total.

Design a Mini-Kindle App that conforms with the MVC pattern. At this time, the Mini-Kindle App only has two views, the Library view and the Book view.

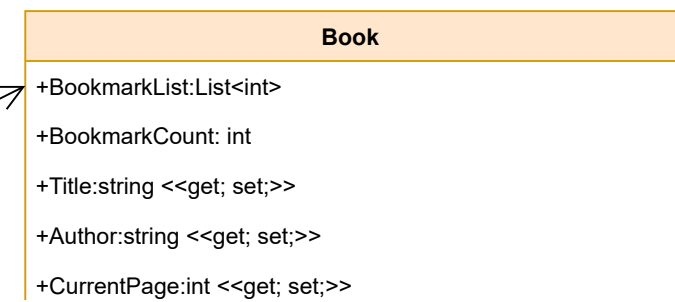
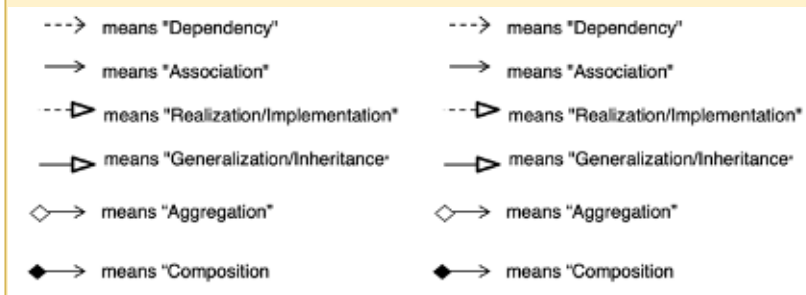
- In the Library view the user can (i) synchronize the book library with the library in the cloud, and (ii) select (open) a book to read.
- When the library is synchronized, the library view is updated with the new content (if any) from the cloud library.
- In the Book view, the user can (i) flip pages back and forward, (ii) set/remove bookmarks (up to 5 bookmarks per book), (iii) go to a specific page (by providing a page number)
- When the user selects/opens a book, the latest/current page is displayed
- You may use enums with different states to update the views

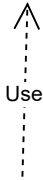
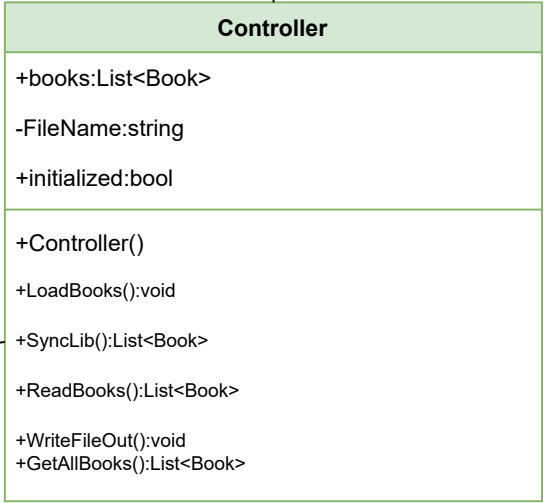
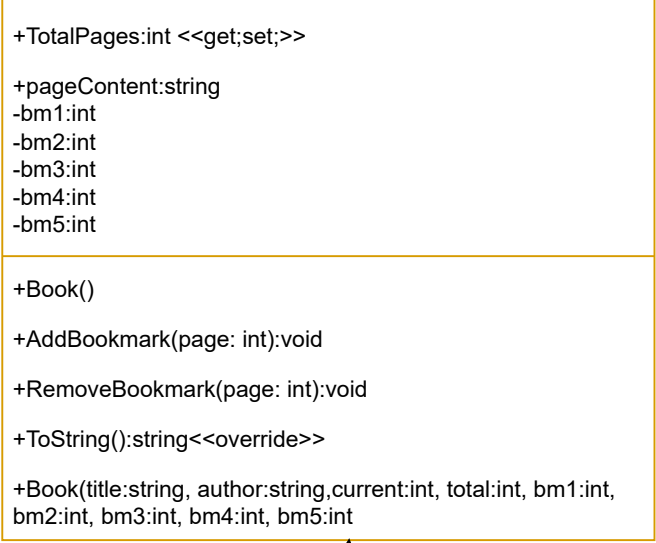
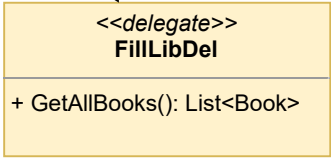
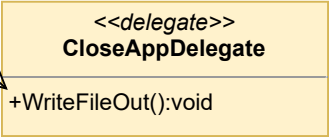
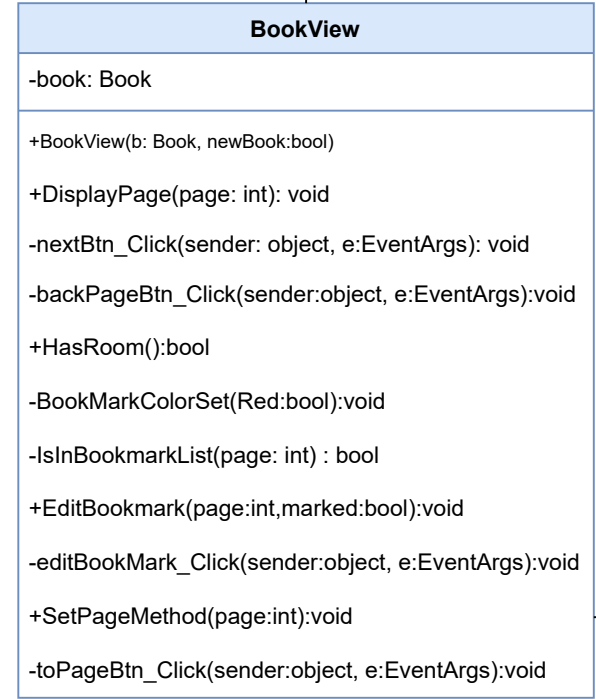
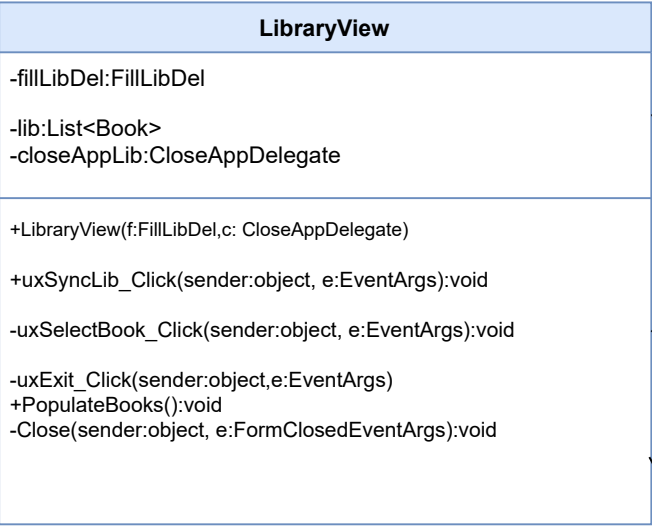
The user can execute the commands from a GUI. You can assume that there are controls in the GUI for each of the operations mentioned above.

Your class diagram must follow the MVC as we discussed in class or if you feel more comfortable with the MVVM you can use it instead. You **MUST** use delegates to decouple the View(s) from the Controller(s).

In each class include the minimum attributes and methods needed to satisfy the requirements described above. It is expected full method signatures, appropriate visibility modifiers, descriptive naming convention for classes, attributes, methods, and delegates. All relationships that accept multiplicity must be included in the diagram.

For your class diagram, remember:





HAS_A

HAS_A

Implements

-Implements

HAS_A

HAS_A