

# conor\_healy\_project\_1\_reflections\_document

March 13, 2019

## 1 My Project: book\_manager.py

### 1.1 Summary

- book\_manager.py is a tool to manage books, ratings, reviews and thoughts.
  - It includes 4 classes:
  - Bookshelf(): An entity that represents a place for all my books.
  - Book(): An entity that represents a book and associated data.
  - Review(): An entity that represents a book review and associated data.
  - BookshelfManager: An entity that creates new /loads instances of the bookshelf, book, and review objects and allows for user interaction.
  - I tried to manage data quality by managing user input at data entry, usually by requiring the user to input valid data before moving on. For this project, it was not practical to include validation for most text input fields (book title, Big Idea, etc.).
  - When book\_manager.py is loaded, all 4 classes are loaded and a single call is made to BookshelfManager(). From there, the user interacts with the BookshelfManager() instance.
1. First, the user is provided with a description of the program and instructions to load or add a bookshelf.
    - The user can choose an existing bookshelf file to load. The bookshelf files are created using the pickle module and automatically saved by book\_manager.py with a ".bkshlf" extension.
    - The user can also choose to add a new bookshelf and is prompted to name their bookshelf
  2. Once the bookshelf is loaded/created, the user is provided with the Main Menu and instructed to choose an option. The menu will repeat until the user chooses a valid option. The user can Enter:
    - (L) to (L)ist my books
    - (F) to (F)ilter my books
    - (T) to sor(T) my books
    - (G) to (G)et the info about a book
    - (A) to (A)dd a book
    - (R) to (R)evuew a book
    - (C) to save and (C)lose

- (Q) to (Q)uit without saving
  - (S) to (S)ettings
3. Each menu option has unique behavior:
- (L) to (L)ist my books
    - Prints out the books in an ascii bookshelf
  - (F) to (F)ilter my books
    - Asks the user to choose a method to filter the books
    - Multiple filters can be applied. Each selection in the filter menu is applied against the books remaining after previous filters.
    - Also, users can reset all filters and sorts or back out to the Main Menu
  - (T) to sor(T) my books
    - Asks the user to choose a method to sort the books. Once a sort is chosen, the user is prompted to choose whether to sort ascending or descending
    - Multiple sorts can be applied. Each selection in the filter menu is applied against the book list sorted by previous sorts. If one wanted to sort by multiple sorts intentionally (e.g. by Author then by Title), then the sorts should be applied backwards (First by Title, then by Author) to get the right result.
    - Also, users can back out to the Main Menu.
  - (G) to (G)et the info about a book
    - Prints out the books in an ascii bookshelf
    - Asks the user to choose a book based on it's position in the bookshelf
    - Prints details about the book
  - (A) to (A)dd a book
    - Prompts the user to add a book.
    - If "Want to Review" is chosen as the status, the user is prompted for whether they want to review the book now.
      - \* If yes, the user is prompted to add a review
      - \* If no, the book status is changed to "Read, but not reviewed"
  - (R) to (R)evuew a book
    - Prints out the books in an ascii bookshelf
    - Asks the user to choose a book based on it's position in the bookshelf
    - Prompts the user to add a review.
  - (C) to save and (C)lose
    - saves the bookshelf with the pickle method using the name of the bookshelf and a ".bkshlf" file extension
    - Exits the program
  - (Q) to (Q)uit without saving
    - Prompts the user to enter "Quit"
      - \* If the user enters "Quit", exits the program.
      - \* If not, returns to the Main Menu
  - (S) to (S)ettings

- Prompts the user with choices to to:
  - \* Reset the bookshelf by removing all filters and sort and recalculating all counters.
  - \* See any Beta test features (currently a printout of the bookshelf, books, an example book, and the bookshelf associated with the example book)
  - \* Back out to the Main Menu.

## 1.2 Testing & Using `book_manager.py`:

- Hopefully, the Summary above covers the functionality pretty well. A simple python `book_manager.py` command should load and run the project.
- I created this project using a Mac, but don't anticipate any problems running it on windows or linux.
- I am providing a demo bookshelf (`demo.bkshlf`) pre-loaded with 25 books and 4 reviews. The file can be loaded by choosing (L) at the initial prompt, then typing the path to the file.
- A new bookshelf can also be created. Be careful to name it something other than "demo" or the file I provided will be overwritten.

## 1.3 Challenges

- This was a great project that helped me learn quite a bit about python and object oriented programming.
- I also tried my hand at git branching, and think it could be recommended for students in the future. Class development seems ideal as a toy example for feature development, and a reference document / optional async would have been useful.
- My biggest challenge was project scope. Adding book details with user prompts + sort + filters + printing out bookshelves, book details, and review details made for a bigger project than I first anticipated. (I got instructor approval to go over the 750 line limit to get it all in, and still left some valuable features on the Future Enhancements list).
- My second biggest challenge was workflow related. We haven't covered use of an IDE in this course, but jupyter notebooks are not well suited for this kind of development. I ended up with a hybrid workflow that got me done, but I look forward to finding a better tool for development work.
- I tried briefly to get Goodreads integration working, but I found that the Goodreads import and export .csv files are not compatible and are not documented. I plan to play with the import functionality in the future, and then build a Save to Goodreads.csv option once I understand what Goodreads wants as an import.
- I ran into bugs while developing the project, but most of my issues seemed to be about the quirks of this project that would likely be replaced by more sophisticated tools in production:
- building the ascii bookshelf to display the books
- setting up the `multi_input` function to allow users to enter multiple lines of input.
- There are a few places where I use a long series of `elif` statements to dynamically return different attributes/results, and I'm sure there's a better way to code them. However, it is functional so I chose to focus elsewhere.