

User Doc w/ screenshots

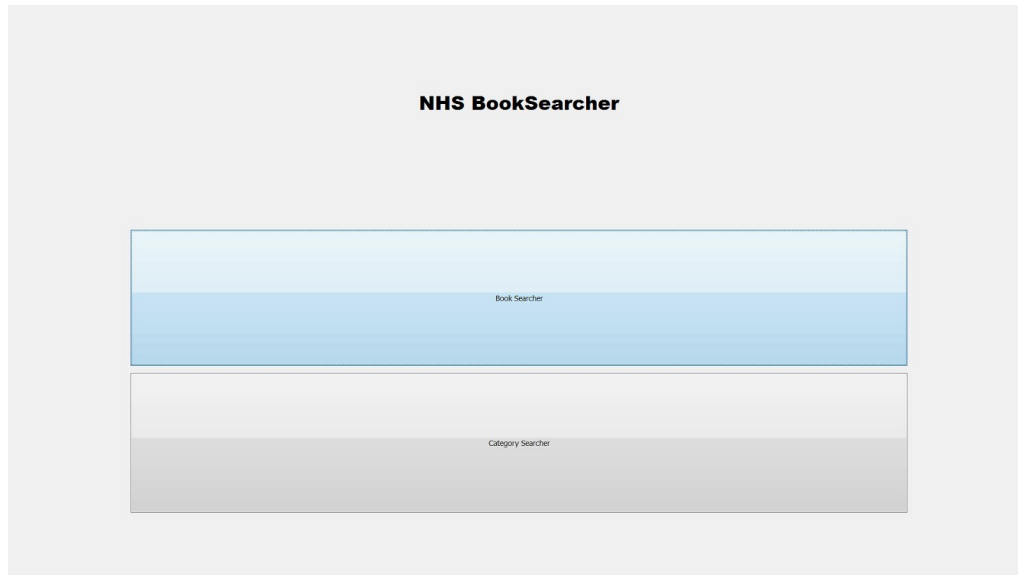
This program will be used by those in a library setting who wish to receive more information about a particular book in the library. The user will simply select the “BookSearcher” feature from the mainmenu of the program, and scan the barcode containing the ISBN of the particular book. Upon scanning the book, our program will list a large variety of information about said book, including the author, publisher, date published. Furthermore, the local reviews will be displayed to the right of this information area, so the user can read reviews written by readers from the same library, and sort these reviews based on rating. The average rating of the book based on other reviews will also be displayed upon scanning the book, and the user can add their own review and rating out of five stars to the scanned book. These added reviews will appear the next time anyone scans this same book. At the bottom of the description area is where the program displays MLA and APA formatting style guides for the particular book, so that the user can easily cite any book for a project, essay, or homework. This feature will improve the student’s use of reliable books as sources for projects, as the MLA and APA citation formatting is provided without any worry for formatting issues, thus increasing one’s likelihood to read and reference a book. The program can also be used to recommend books to any reader. The user must simply select “Category Searcher” from the main menu, and then select a category based on books that have already been scanned into the system. Selecting a category will reveal a number of books read by others in the library, ordered from highest rating to lowest. From this menu, the user may select the book to receive more information about the book, and have the opportunity to read reviews left by other readers. To enable proper functionality of the program, jsoup-1.10.2.jar must be properly placed into the ‘lib’ folder of the program - a subdirectory of the program’s main executable jar file. A functional internet connection must also be present, and it is recommended that Java 7 or higher be installed on the computer running the program. Our imagined use-case involves a Raspberry Pi running Raspbian Linux, which automatically launches the program upon boot (disable Raspbian GUI, boot into terminal). Thanks to the RPIs silent operation, auto-booting when attached to a power source, and linux operating system, the device should require no maintenance to ensure proper operation. A book-scanning station would then be created with a monitor, keyboard, mouse, and scanner, where users could walk up and scan their books without clicking a single button (scanner pressed enter key upon scanning). A barcode scanner, mouse, keyboard, and monitor are all requirements of the proper function of the program.

Administrative notes: All storage files (bookDB.txt, categories.txt) are delimited by the ASCII Character 31, and to delete a review or remove a book from the program will require a text editor (Notepad++ recommended) to manually delete the desired data. Note: When removing a book from system, it must be removed from all databases to ensure categories and book database align properly. Error meanings: Error 1: Valid ISBN entered, but Google did not have any information about the book in the Google books database. Error 2: Invalid ISBN entered, ISBNs must be a length of 10 or 13 characters, will all numbers with the exception of an “X” at the end of some ISBNs - check ISBN formatting and re-try. Error 3: Empty review tried to be submitted, enter text and re-submit. Error 4: 403 Error received from Google books server due to too many requests being received from program, user must wait 30 seconds before

performing any more search actions. Error 5: Rating of 0 attempted to be placed to a book. Reviews must be of rating 1-5 stars, user must select proper amount and re-submit.

User instructions

The main menu screen gives the user the option between the 'book searcher' menu that searches for the books by ISBN and the 'category searcher' menu that searches for multiple books that are already in the program's memory by category.



The book searcher accepts an ISBN in the field in the bottom left of the screen.

The screenshot displays the search and review interface of the NHS BookSearcher application. On the left side, there is a form with the following labels: "Title:", "Google Rating:", "Author:", "Category:", "Publisher:", and "Published:". Below these labels is a large, empty text input field. At the bottom left, there is an "ISBN:" label followed by a text input field and a "Search" button. On the right side, there is a "Reviews:" section with a "Low/High" dropdown menu. Below this is a large, empty text input field. At the bottom right, there is a "Create Review:" label followed by a "5/5" rating display and a "Submit Review" button.

Once an ISBN is searched, the screen that appears will resemble this image:

[Back](#)

Title: Les Miserables a New Unabridged Translation

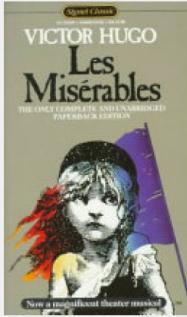
Google Rating: 1.5

Author: Victor Hugo

Category: Fiction

Publisher: Penguin

Published: 1987



After nineteen years in prison, Jean Valjean has difficulty adjusting to the outside world, which scorns and shuns him
MLA: Hugo, Victor. *Les Misérables a New Unabridged Translation*. Penguin, 1987.
APA: Hugo, Victor. (1987). *Les Misérables a New Unabridged Translation*. Canada: Penguin.

ISBN:

Reviews: [LowTough](#)

1 - Every google review EVER ~ 1/5
5 - Ryan ~ Google is WRONG

Create Review:

3/5

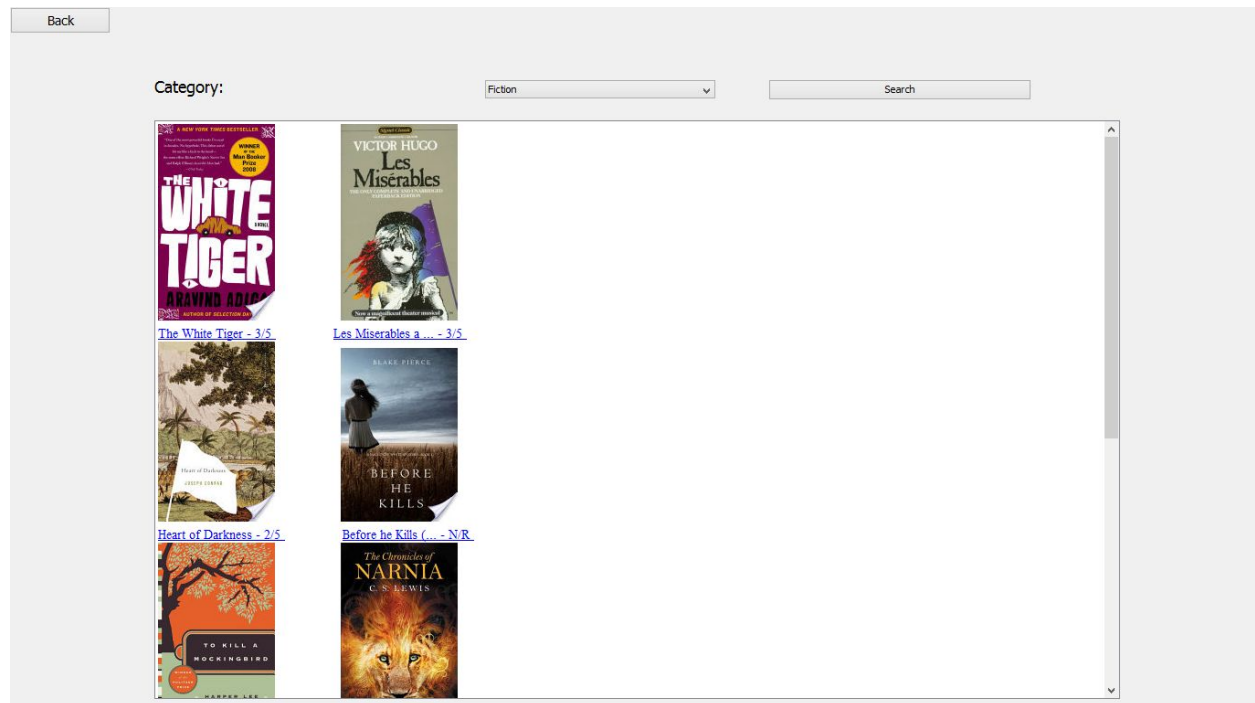
Information appears along the left side of the screen, including the book's title, author, description, image and proper citation formatting in both APA and MLA formats. On the right side of the screen, reviews left by other people who have viewed the book appear in the top right of the screen. The book may be given a star rating out of 5 (minimum 1), with an accompanying review.

The category searcher has a drop-down menu of all of the categories of the books that have already been entered into the program's memory.

[Back](#)

Category:

When the search button is pressed, all books in the program's memory in the selected category are displayed with a hyperlink to the book searcher's display of the book's information.



Testing Doc

We have tested for all possibilities that we have considered, ensuring that these possibilities either produce results or gracefully handle problems. Possibilities include:

- Invalid ISBN being searched
- Empty reviews written
- Inappropriate words in review written
- Missing pieces of information about the book searched (including no images found)
- Enter key pressed while writing a review
- Run program on different screen resolutions to test resizability
- Internet connection issues, server connectivity issues (403 error from Google)
- ISBN of book not in Google's database being searched
- Submit reviews with various symbols and numbers to ensure delimiter works as intended
- Run search on many different books

Bugs

If too many books are searched for in a short span of time, a 403 error is thrown. The error 403 message will display multiple times upon searching a book. It does not break the program, preventing it from running, and waiting for some time before searching again solves the problem. Searching for the same book multiple times increases the size of the image shown. When searching using categories, occasionally images will not show up until highlighted or double clicked, due to Java's poor implementation of CSS and HTML (used for rendering categories information). Hitting enter on category list pops up with ISBN not found. If somehow

an invalid ISBN is submitted (unsure of how it managed to add when it was invalid) but searching categories will not run with an invalid ISBN. When searching for book sometimes stars will disappear. Very slow to load book search from clicking on link within categories.

Incomplete Features

The graphics could be improved to look better than they currently do.

The program could be modified in future (possibly by others) to incorporate additional features that would be useful in the library, such as a peer tutor login or the ability to search for a book and discover where it would be found in the library.

The idea of having a account and login system was considered but not implemented. Users would create accounts with their student number and it would have shown their student number for reviews and recommended books based on the user's courses.

Created by: Liam Eaton, Ryan Lithgow, Spencer Neal (January 5th - January 25th, 2017).