**A picture containing text, shelf, book, indoor

Description automatically generatedFigure 1 – Initial GUI Status**

The initial GUI status, with placeholder text in each of the 4 input fields. Utilising the ‘**ClearTextField’** function found in the menu.py file, a click system is used that counts and function found in the menu.py file, a click system is used that counts and deletes the placeholder text when user clicks in input box for the first time.

A screenshot of a computer

Description automatically generated with medium confidence**bookSearch.py**

**Figure 2 – Book Search with valid book title**

By inputting a book title into the search box, the user can search for all versions of that book in the library, being returned all the ID’s and said ISBNS of that book.

**A screenshot of a computer

Description automatically generated with medium confidenceFigure 3- Book Search with invalid book title**

Validation errors are handled and reported back in the library management system, when a book title searched for doesn’t exist, the message is shown to them.

**bookCheckout.py**

A screenshot of a computer

Description automatically generated with medium confidenceSting interpolation especially useful here where inserting new record into loan History, updates to Member ID of book record and increments the TimesTakenOut of book.

**Figure 4 – Book checkout with invalid memberID**

Validation rules are also created to ensure that memberID’s of only 4 digits will be accepted, where if not the error is handled and a messagebox.showerror pops up on the users display.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 5 – Book checkout attempt with non-available book**

A screenshot of a computer

Description automatically generated with low confidenceIf the book has already been taken out, then another message will pop up in the user screen to inform them that the book isn’t available to be checked out.

**Figure 6- Book checkout successful with valid memberID and bookID**

Finally, when a valid member ID is inputted and the book searched for exist and is available it is then checked out by the user, displaying the member id of the person who checked out the book as well as the book title and book ID, all whilst making it unavailable for others to check this book out.

**bookReturn.py**

incorporating nested if else statements for the queries of

1. if the bookID of the search was found
2. to check the member id of the book that was found
3. A screenshot of a computer

   Description automatically generated with low confidenceupdate the book return status

**Figure 7 – book return attempt with bookID that has already been taken out by someone else**

when a book is attempted to be returned that is given the wrong member id then this message is shown to the user, informing them to try again.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 8 – Book Return with bookID of a book that hasn’t been checked out**

Additionally, when a book is attempted to be returned that hasn’t been checked out, this error message is shown to the librarian.

**A screenshot of a computer

Description automatically generated with low confidenceFigure 9 – Book return attempt successful with matching memberID to bookID**

Lastly, when the bookID matches the Member ID assigned to it, that the member used to try and return the book, the librarian will be informed and the message will come up in the text box, as well as this the book may now be checked out by anyone again after this.

Chart, bar chart

Description automatically generated**Book Popularity**

**Figure 10- Book Popularity button- visualisation of descending order times taken out**

By pressing the book popularity button, Matplotlib will be used to display the book titles in popularity order based on how many times they’ve been taken out.

A screenshot of a computer

Description automatically generated with medium confidence**Remove Worst Book**

**Figure 11 – Remove worst book button- asks librarian if the least popular book can be removed**

The remove worst book button will attempt to remove the book which has been taken out the least number of times from the library. A pop-up message will be shown to the user to confirm their actions before the removal.

A screenshot of a computer

Description automatically generated with medium confidence**Remove Book**

**Figure 12 – Remove Book – asks librarian if designated BookTitle can be removed**

The Remove book button will ask user to remove a designated book from the library based on the inputted title.

A screenshot of a computer

Description automatically generated with low confidence**Reset Library Machine**

**Figure 14 – Reset Button to reset state**

The reset library button will reset the GUI back to its factory state, resetting the click counters and placeholders

**Book.Py**

A book class is created instead of a dictionary due to it being more flexible and allowing for changes without breaking where the column headers can be changed and allowing for operations to be performed on them.

A book class object is created by getting the results from the book search and setting the array to = the resulting columns e.g. [0] = ID [1] = ISBN