GROUP ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

PROGRAMMING WITH PYTHON

APD1F2407CS(CYB)

Group 1

Lecturer: Muhammad Huzaifah Bin Ismail

Handout Date: 14 AUGUST 2024

Hand in Date: 04 NOVEMBER 2024

No	Student Name	TP Number
1.	ASEEL M. H. SANAD	TP080667
2.	SARA ELWALID HASSAN FAGIR	TP078376
3.	MAANVI GOORBIN	TP081121
4.	SHUKRI DAHIR AHMED	TP073809

Table of contents

1.0 Introduction 6
2.0 Assumptions 7
3.0 Structure Diagram 9
4.0 Design of the program (Pseudocode) 10
  4.1 Member User 10
    4.1.1Member Login 10
    4.1.2 Password Validation 11
    4.1.3 Contact Number Validation 12
    4.1.4 Member ID Randomising 13
    4.1.5 Email Address Existence check 14
    4.1.6 Member Sign Up 15
    4.1.7 Display members' borrowed books List 16
    4.1.8 Loading File function 17
    4.1.9 Saving and Storing Updated File Function 17
    4.1.10 Return Book Function 18
    4.1.11 Payment Function 19
    4.1.12 Book ID Validation Function 20
    4.1.13 Updating members' profile function 20
    4.1.14 Displaying available Books list 21
    4.1.15 Members' menu 22
  4.2 Librarian User 23
    4.2.1 Librarian Login 23
    4.2.2 Book ID Randomising 24
    4.2.3 Add new book to existing book lists 25
    4.2.4 View Book List Function 26
    4.2.5 Search specific book from book list 27
    4.2.6 Edit book information from book list 28
    4.2.7 Remove specific book from book list 29
    4.2.8 Validating Member ID 30
    4.2.9 Processing a book loan 31
    4.2.10 Librarians' menu 32
  4.3 System Administrator User 33

4.3.1 Add new librarian or new member to existing list 34
4.3.2 Go Back Function 35
4.3.3 Display members' list 35
4.3.4 Search specific member from existing list 36
4.3.5 Edit specific member's information from existing list 37
4.3.6 Remove specific member from existing list 38
4.3.7 Managing members Function (Menu to manage members) 39
4.3.8 Randomising Librarian ID 40
4.3.9 Display Librarians' list 40
4.3.10 Search specific librarian's information 41
4.3.11 Edit specific librarian's information 42
4.3.12 Remove specific librarian from existing list 43
4.3.13 Managing librarians function (Menu to manage librarians) 44
4.3.14 Admin login 45
4.3.15 Admin Menu 46
4.3.16 Show admin menu function 47
4.4 Main Menu for Brickfields Kuala Lumpur Community Library 48
5.0 Programs' Source Code 49
5.1 Implementation 49
5.1.1 Member User 49
5.1.1.1 Member Login 49
5.1.1.2 Password Validation 51
5.1.1.3 Contact Number Validation 52
5.1.1.4 Member ID Randomising 53
5.1.1.5 Email Address Existence check 54
5.1.1.6 Member Sign Up 55
5.1.1.7 Display members' borrowed books List 56
5.1.1.8 Loading File function 57
5.1.1.9 Saving and Storing Updated File Function 57
5.1.1.10 Return Book Function 58
5.1.1.11 Payment Function 60
5.1.1.12 Book ID Validation Function 61
5.1.1.13 Updating members' profile function 62

5.1.1.14 Displaying available Books list 62
5.1.1.15 Members’ menu 63
5.1.2.0 Librarian User 64
5.1.2.1 Librarian Login 64
5.1.2.2 Book ID Randomising 65
5.1.2.3 Add new book to existing book lists 66
5.1.2.4 View Book List Function 67
5.1.2.5 Search specific book from book list 67
5.1.2.6 Edit book information from book list 68
5.1.2.7 Remove specific book from book list 69
5.1.2.8 Validating Member ID 70
5.1.2.9 Processing a book loan 71
5.1.2.10 Librarians’ menu 72
5.1.3 System Administrator User 74
5.1.3.1 Add new librarian or new member to existing list 74
5.1.3.2 Go Back Function 75
5.1.3.3 Display members’ list 75
5.1.3.4 Search specific member from existing list 76
5.1.3.5 Edit specific member’s information from existing list 77
5.1.3.6 Remove specific member from existing list 78
5.1.3.7 Managing members Function (Menu to manage members) 79
5.1.3.8 Randomising Librarian ID 79
5.1.3.9 Display Librarians’ list 80
5.1.3.10 Search specific librarian’s information 80
5.1.3.11 Edit specific librarian’s information 81
5.1.3.12 Remove specific librarian from existing list 82
5.1.3.13 Managing librarians function (Menu to manage librarians) 83
5.1.3.14 Admin login 84
5.1.3.15 Admin Menu 84
5.1.3.16 Show admin menu function 85
5.1.4 Main Menu for Brickfields Kuala Lumpur Community Library 85
5.1.5 Data Storage 86
5.1.5.1 “MemberBorrowedBooksInfo.txt” file 86

5.1.5.2 “availablebooks.txt” file 87
5.1.5.3 “BookList.txt” file 87
5.1.5.4 “librarians.txt” file 87
5.1.5.5 “members.txt” file 88
6.0 Sample Input and Output 89
6.1 member 89
6.2 Librarian 110
6.3 Remove book val 119
6.4 Search by title 122
6.5 Add book validation 123
6.6 System admin 124
7.0 Conclusion 143
8.0 Workload Matrix 144
9.0 References 145

1.0 Introduction

Brickfields Kuala Lumpur Community Library had a manual system to manage data and procedures for the three users: members, librarians, and system administrators. This system caused several problems. For instance, it was vulnerable to human errors. While operating the library, the users could make mistakes, that could affect other procedures like book loaning. However, with an automated system, equipped with validations, recording data and running a huge library could be more efficient, rapid, and less prone to errors. Additionally, although installing an automated system can be costly due to the initial investment in equipment, it could increase efficiency and reduce manpower needed to function, reducing expenses over time. Therefore, compared to the manual system, the automated one is better for the smooth running of the library, and it would also give it a modern touch.

Our team was approached to create this Python-based automated system. Processes like returning books, members' registration, staff registration, and others would be automated. Validations and other functionalities are implemented in the system, to ensure effective management and boost overall efficiency and accuracy.

2.0 Assumptions

- All payments for overdue return fee are made in Malaysian Ringgit (RM) and on a number of days overdue basis, as shown below:

Days	**Fee (RM)**
1 day	2.00
2 days	3.00
3 days	4.00
4 days	5.00
5 days	6.00
More than 5 days	10.00

- There are three primary users, namely, Library Member (Non-Staff), Librarian (Staff), and System Administrator (Staff). Each will have their respective menu page that they can access with their given IDs, email addresses, and passwords. This also promotes access control, preventing users from accessing unauthorised data.

- IDs for the users and books are random but unique. Member ID is in the “MEMnnnn” format, Librarians ID, “LIBnnnn”, Book ID, “Bnnnn”, where n represents a digit.

- Only system administrators can edit members’ information and so only they can process payment of overdue return fees of members.

- Librarians can add new books into the catalogue, view existing books, search specific book, on request by members or to edit book information, and they can also remove books from the catalogue, if ever they are outdated or damaged. Librarians are also the one responsible for loaning books to members.

- Members can process their handing in borrowed books, search books, view their personal profile by themselves.

- Members can borrow up to five books.

- When a member makes a payment, the librarian will edit this information in the system.

- There is only 1 Super Admin Account, which the System Administrators can access. They would usually need to type in ‘admin’ as their username and password (code) change every hour, and only the administrators can access an authenticator application. For the simplicity of this system, the code is assigned as ‘12546’.

- System Administrators can manage members’ accounts, that is, they can add new members’ information, view all members’ information, search member information, edit member information, and remove members from the system.

- System Administrators can also manage librarians’ accounts, that is they can add new librarians’ information, view all librarians’ information, search librarian information, edit librarian information, and remove librarians from the system.

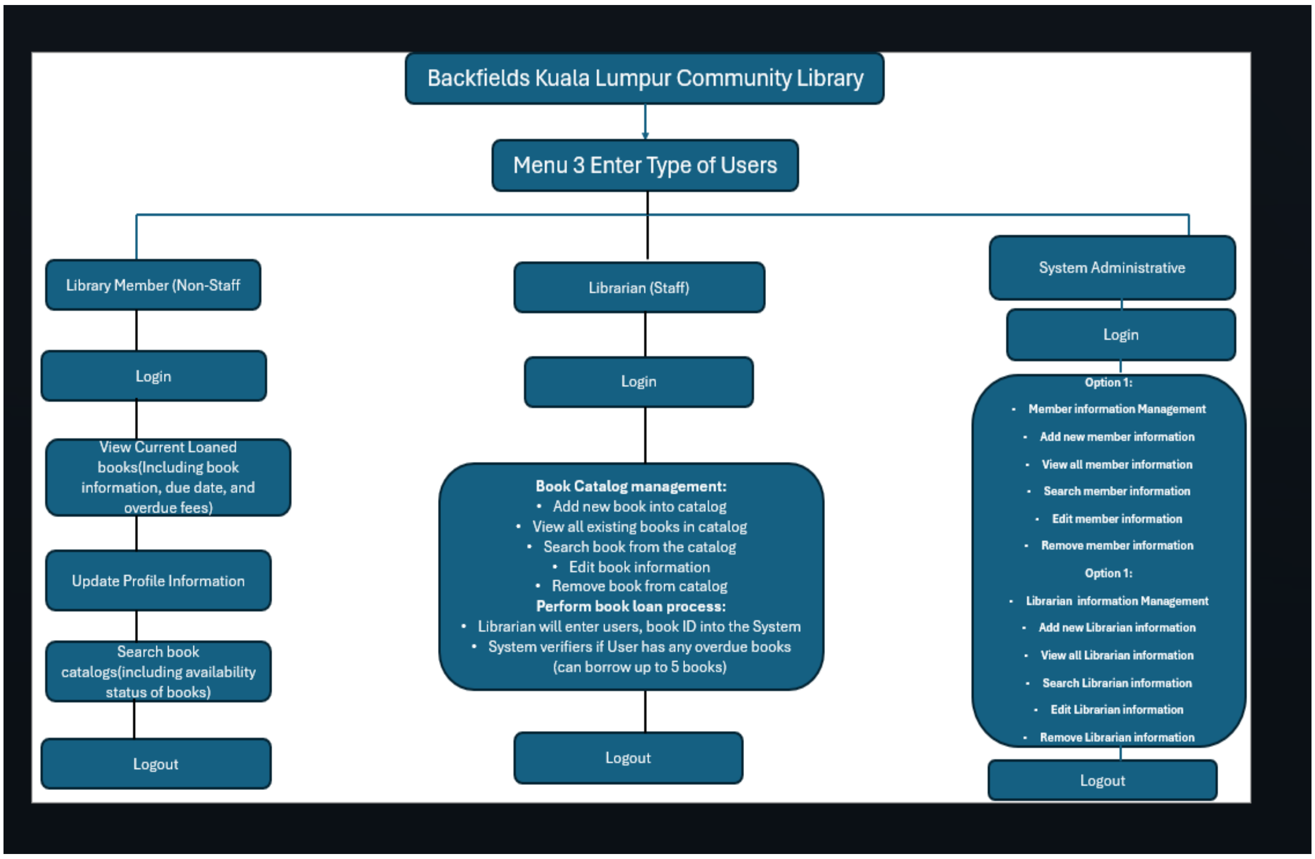
- Data are stored in text files.

- There are four files being used, namely, “MemberBorrowedBooksInfo.txt”, “availablebooks.txt”, “BookList.txt”, “librarians.txt”, “members.txt”.

- Books borrowed should be returned within one week.

3.0 Structure Diagram

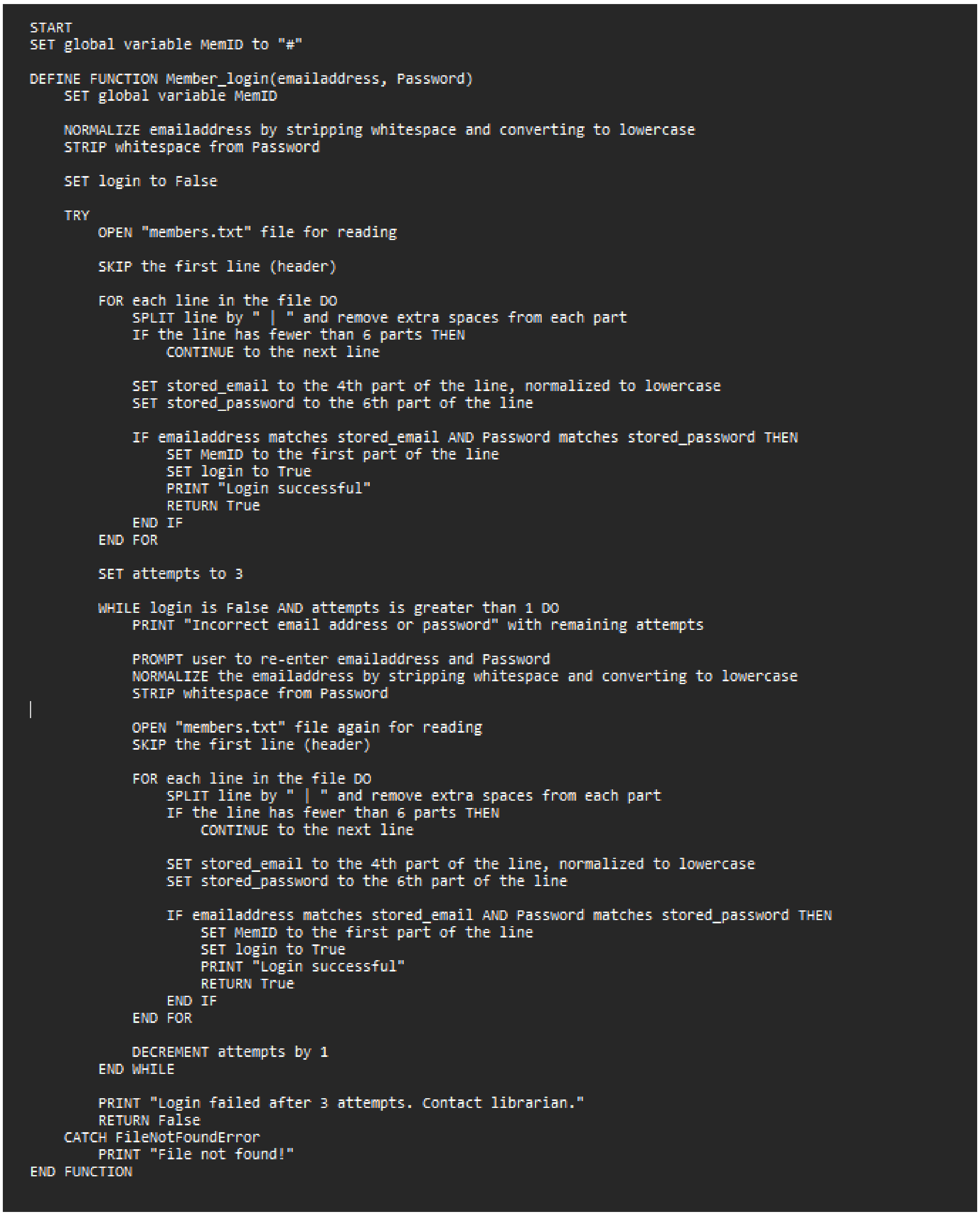
Here is a breakdown of the program:


Backfields Kuala Lumpur Community Library
Menu 3 Enter Type of Users
Library Member (Non-Staff
Login
View Current Loaned books(Including book information, due date, and overdue fees)
Update Profile Information
Search book catalogs(including availability status of books)
Logout
Librarian (Staff)
Login
Book Catalog management:
• Add new book into catalog
• View all existing books in catalog
• Search book from the catalog
• Edit book information
• Remove book from catalog
Perform book loan process:
• Librarian will enter users, book ID into the System
• System verifiers if User has any overdue books (can borrow up to 5 books)
Logout
System Administrative
Login
Option 1:
• Member information Management
• Add new member information
• View all member information
• Search member information
• Edit member information
• Remove member information
Option 1:
• Librarian information Management
• Add new Librarian information
• View all Librarian information
• Search Librarian information
• Edit Librarian information
• Remove Librarian information
Logout

4.0 Design of the program (Pseudocode)

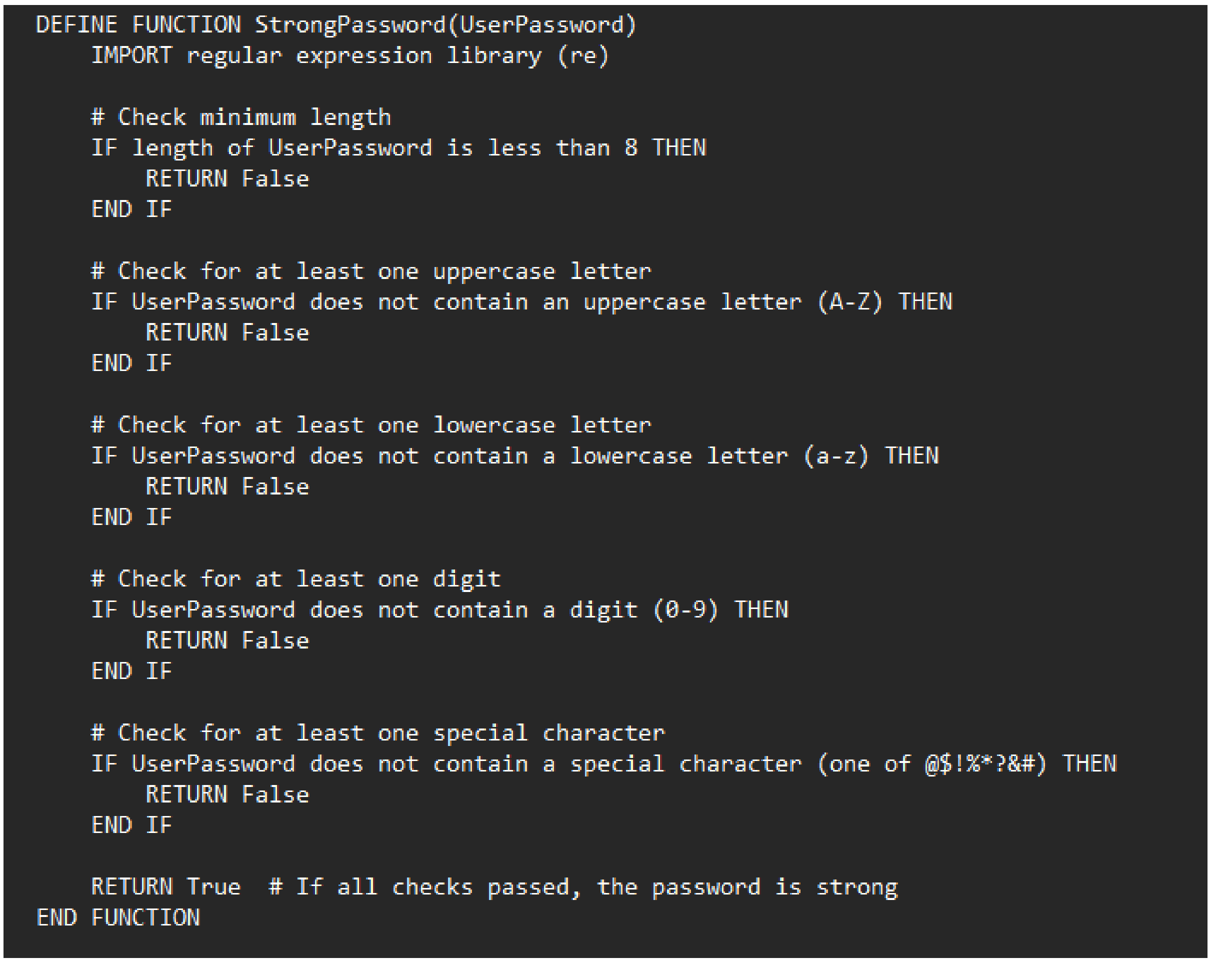
4.1 Member User

4.1.1Member Login



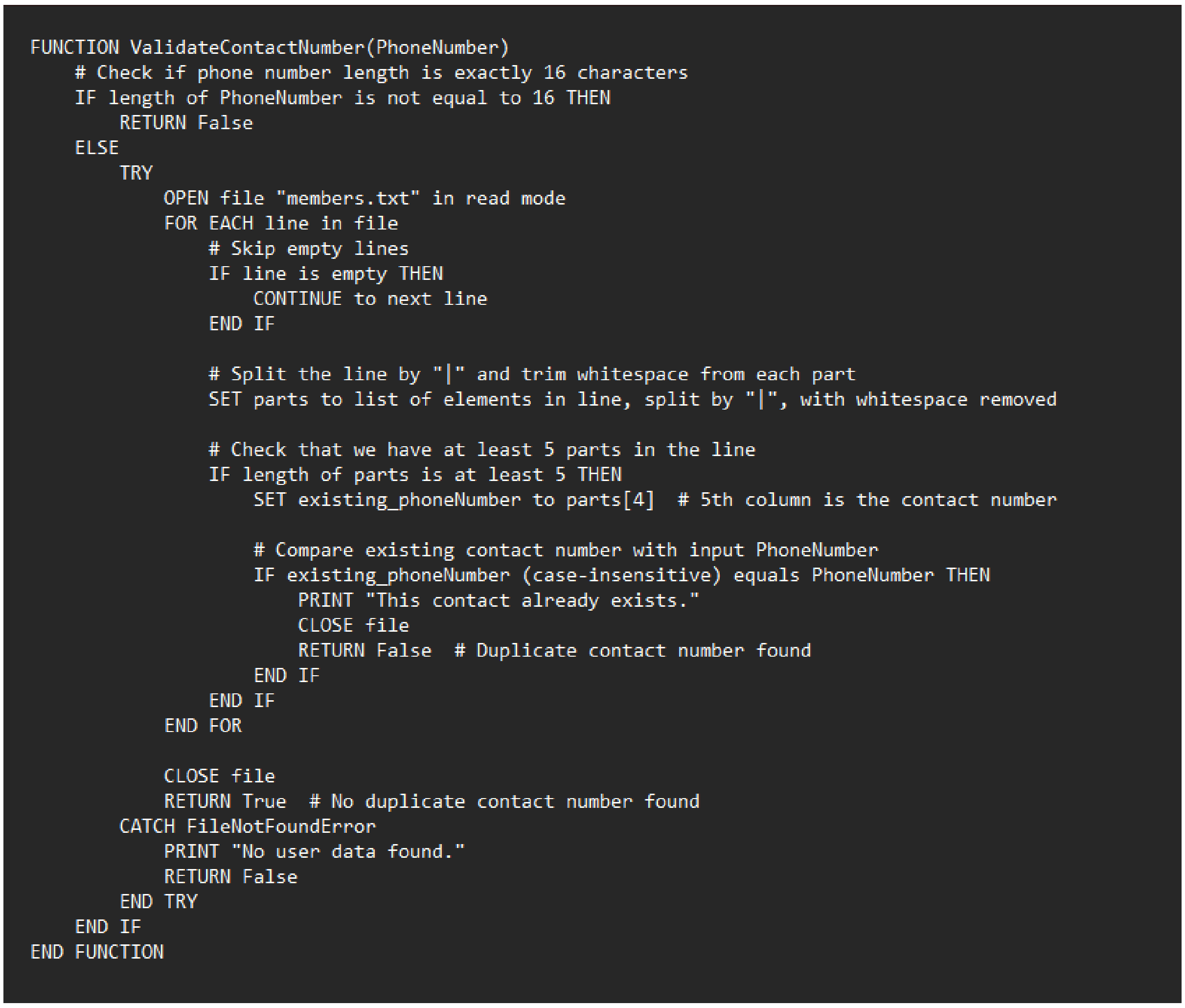
```
START
SET global variable MemID to "#"

DEFINE FUNCTION Member_login(emailaddress, Password)
    SET global variable MemID

    NORMALIZE emailaddress by stripping whitespace and converting to lowercase
    STRIP whitespace from Password

    SET login to False

    TRY
        OPEN "members.txt" file for reading

        SKIP the first line (header)

        FOR each line in the file DO
            SPLIT line by " | " and remove extra spaces from each part
            IF the line has fewer than 6 parts THEN
                CONTINUE to the next line

            SET stored_email to the 4th part of the line, normalized to lowercase
            SET stored_password to the 6th part of the line

            IF emailaddress matches stored_email AND Password matches stored_password THEN
                SET MemID to the first part of the line
                SET login to True
                PRINT "Login successful"
                RETURN True
            END IF
        END FOR

        SET attempts to 3

        WHILE login is False AND attempts is greater than 1 DO
            PRINT "Incorrect email address or password" with remaining attempts

            PROMPT user to re-enter emailaddress and Password
            NORMALIZE the emailaddress by stripping whitespace and converting to lowercase
            STRIP whitespace from Password

            OPEN "members.txt" file again for reading
            SKIP the first line (header)

            FOR each line in the file DO
                SPLIT line by " | " and remove extra spaces from each part
                IF the line has fewer than 6 parts THEN
                    CONTINUE to the next line

                SET stored_email to the 4th part of the line, normalized to lowercase
                SET stored_password to the 6th part of the line

                IF emailaddress matches stored_email AND Password matches stored_password THEN
                    SET MemID to the first part of the line
                    SET login to True
                    PRINT "Login successful"
                    RETURN True
                END IF
            END FOR

            DECREMENT attempts by 1
        END WHILE

        PRINT "Login failed after 3 attempts. Contact librarian."
        RETURN False
    CATCH FileNotFoundError
        PRINT "File not found!"
END FUNCTION
```

4.1.2 Password Validation



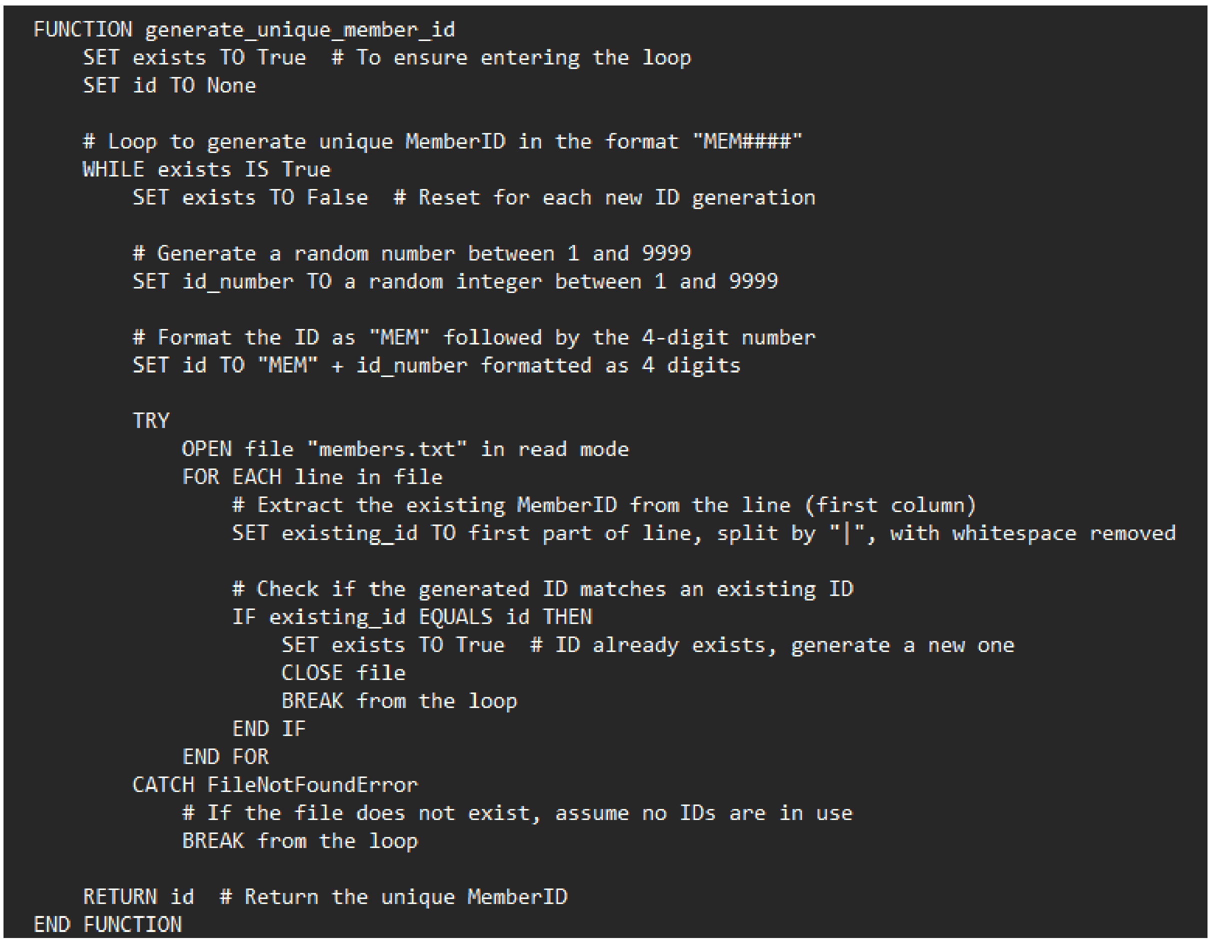
```
DEFINE FUNCTION StrongPassword(UserPassword)
    IMPORT regular expression library (re)

    # Check minimum length
    IF length of UserPassword is less than 8 THEN
        RETURN False
    END IF

    # Check for at least one uppercase letter
    IF UserPassword does not contain an uppercase letter (A-Z) THEN
        RETURN False
    END IF

    # Check for at least one lowercase letter
    IF UserPassword does not contain a lowercase letter (a-z) THEN
        RETURN False
    END IF

    # Check for at least one digit
    IF UserPassword does not contain a digit (0-9) THEN
        RETURN False
    END IF

    # Check for at least one special character
    IF UserPassword does not contain a special character (one of @$!%*?&#) THEN
        RETURN False
    END IF

    RETURN True  # If all checks passed, the password is strong
END FUNCTION
```

4.1.3 Contact Number Validation



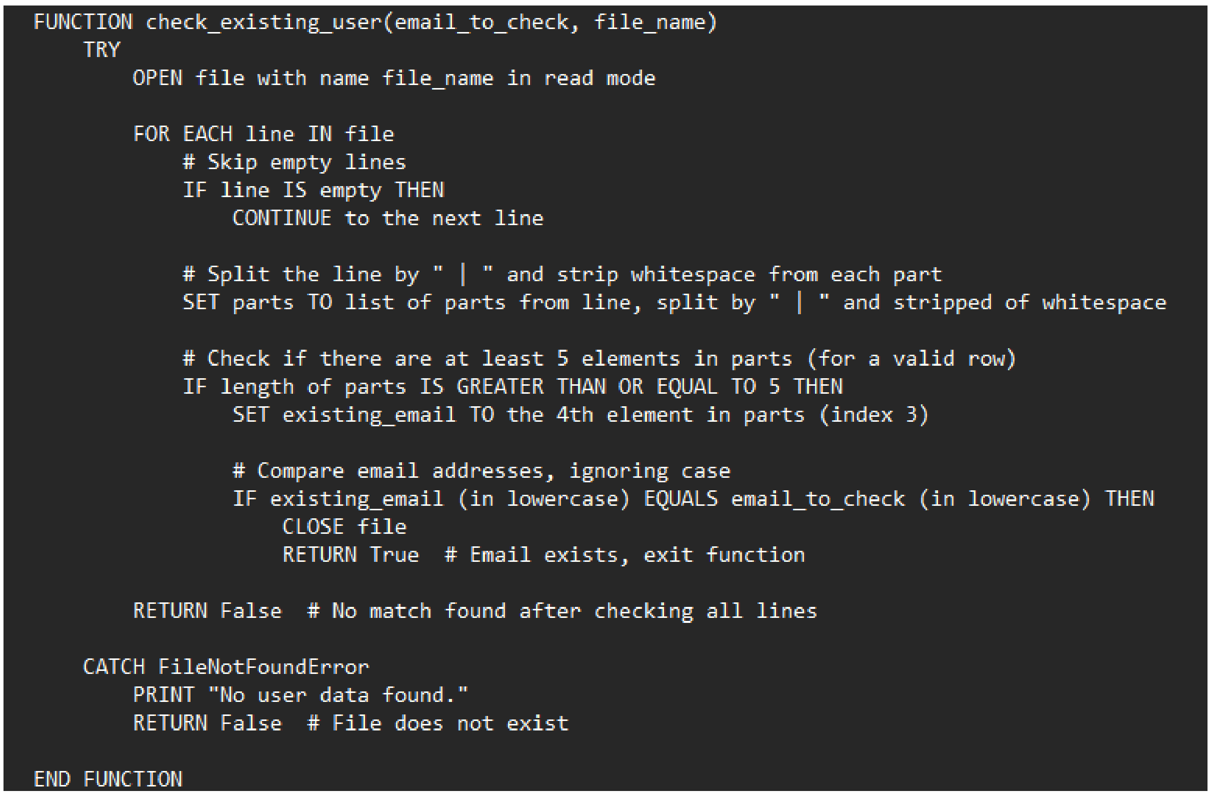
```
FUNCTION ValidateContactNumber(PhoneNumber)
    # Check if phone number length is exactly 16 characters
    IF length of PhoneNumber is not equal to 16 THEN
        RETURN False
    ELSE
        TRY
            OPEN file "members.txt" in read mode
            FOR EACH line in file
                # Skip empty lines
                IF line is empty THEN
                    CONTINUE to next line
                END IF

                # Split the line by "|" and trim whitespace from each part
                SET parts to list of elements in line, split by "|", with whitespace removed

                # Check that we have at least 5 parts in the line
                IF length of parts is at least 5 THEN
                    SET existing_phoneNumber to parts[4]  # 5th column is the contact number

                    # Compare existing contact number with input PhoneNumber
                    IF existing_phoneNumber (case-insensitive) equals PhoneNumber THEN
                        PRINT "This contact already exists."
                        CLOSE file
                        RETURN False  # Duplicate contact number found
                    END IF
                END IF
            END FOR

            CLOSE file
            RETURN True  # No duplicate contact number found
        CATCH FileNotFoundError
            PRINT "No user data found."
            RETURN False
        END TRY
    END IF
END FUNCTION
```

4.1.4 Member ID Randomising



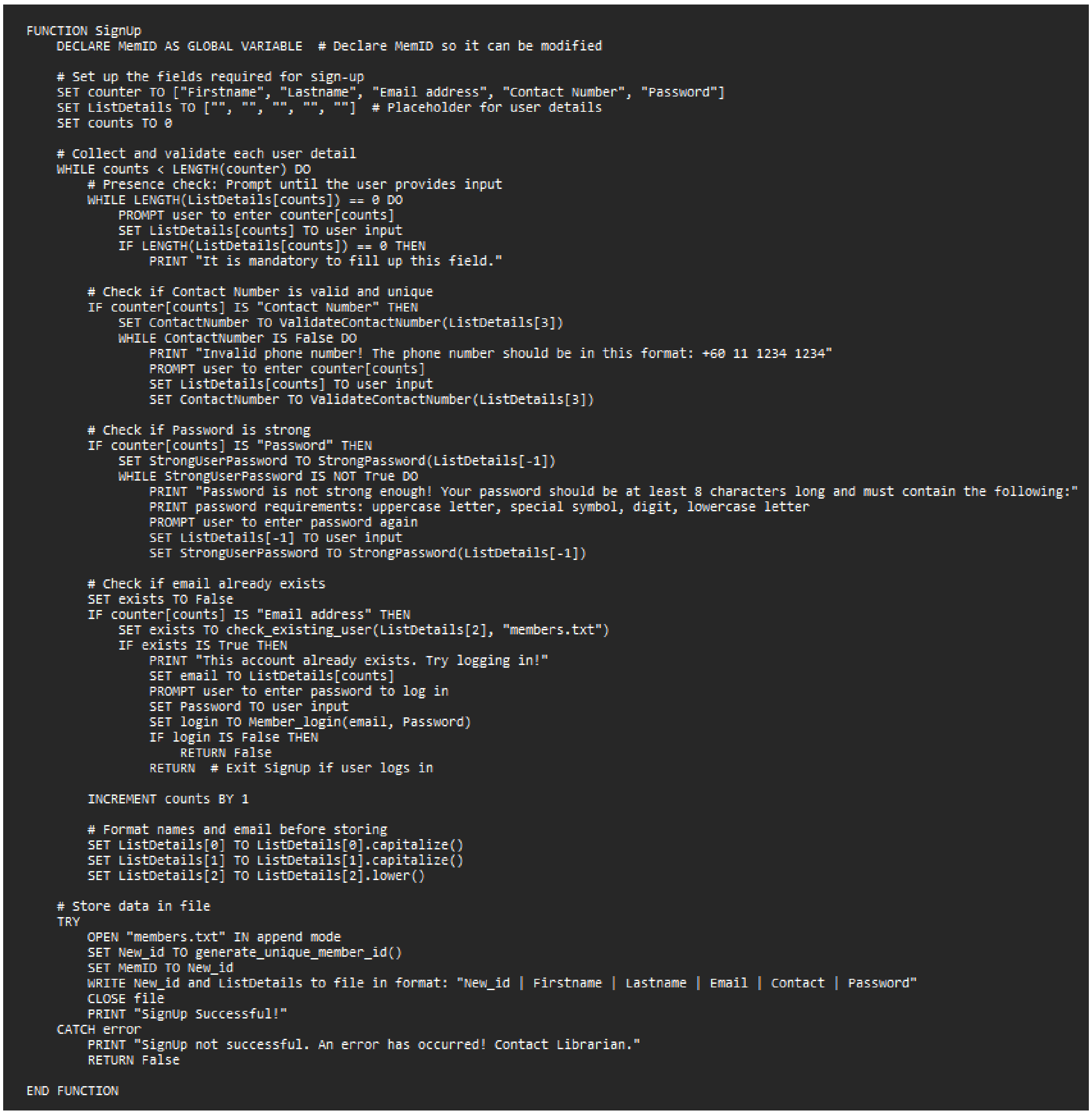
```
FUNCTION generate_unique_member_id
    SET exists TO True  # To ensure entering the loop
    SET id TO None

    # Loop to generate unique MemberID in the format "MEM####"
    WHILE exists IS True
        SET exists TO False  # Reset for each new ID generation

        # Generate a random number between 1 and 9999
        SET id_number TO a random integer between 1 and 9999

        # Format the ID as "MEM" followed by the 4-digit number
        SET id TO "MEM" + id_number formatted as 4 digits

        TRY
            OPEN file "members.txt" in read mode
            FOR EACH line in file
                # Extract the existing MemberID from the line (first column)
                SET existing_id TO first part of line, split by "|", with whitespace removed

                # Check if the generated ID matches an existing ID
                IF existing_id EQUALS id THEN
                    SET exists TO True  # ID already exists, generate a new one
                    CLOSE file
                    BREAK from the loop
                END IF
            END FOR
        CATCH FileNotFoundError
            # If the file does not exist, assume no IDs are in use
            BREAK from the loop

    RETURN id  # Return the unique MemberID
END FUNCTION
```

4.1.5 Email Address Existence check



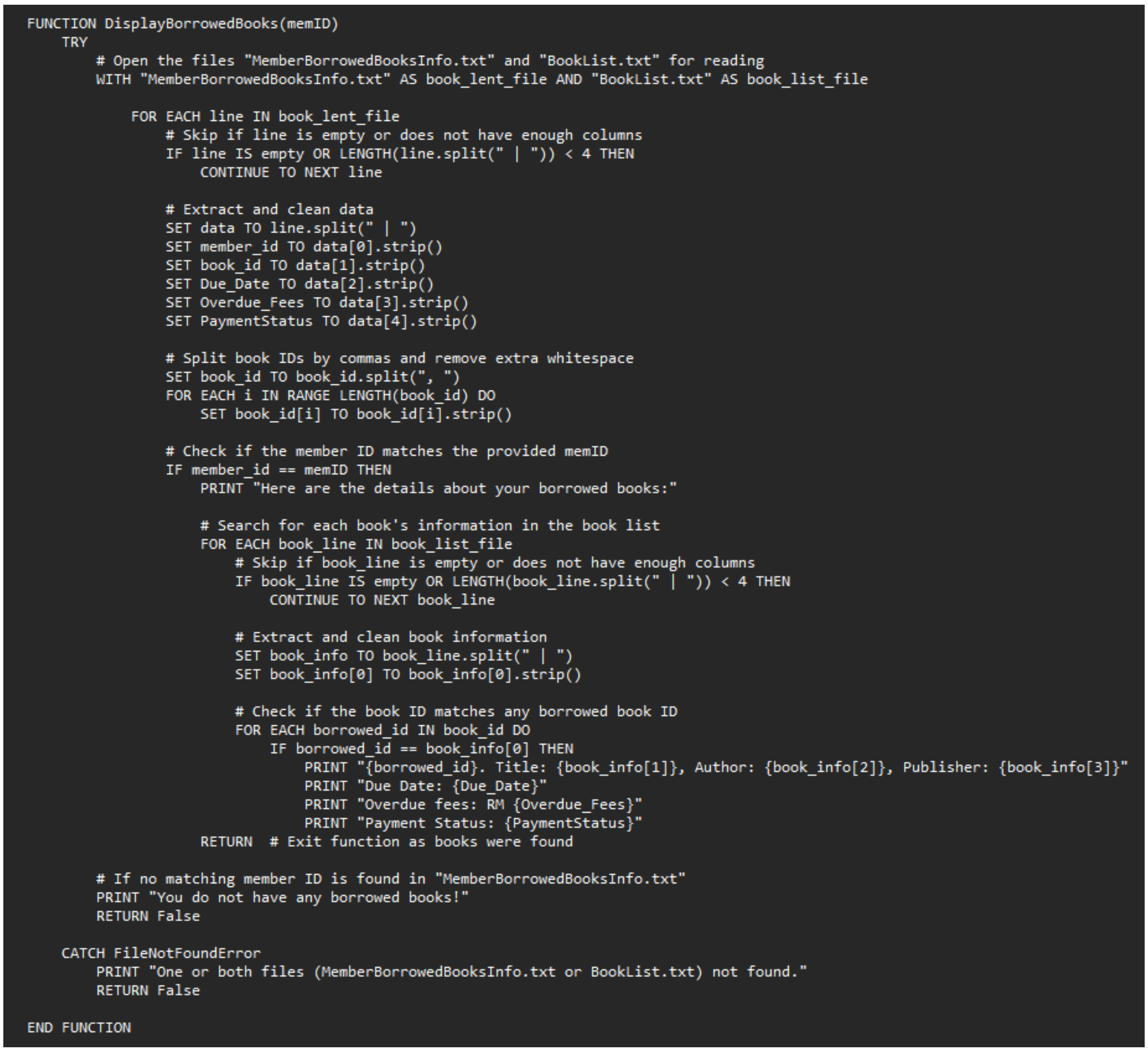
```
FUNCTION check_existing_user(email_to_check, file_name)
    TRY
        OPEN file with name file_name in read mode

        FOR EACH line IN file
            # Skip empty lines
            IF line IS empty THEN
                CONTINUE to the next line

            # Split the line by " | " and strip whitespace from each part
            SET parts TO list of parts from line, split by " | " and stripped of whitespace

            # Check if there are at least 5 elements in parts (for a valid row)
            IF length of parts IS GREATER THAN OR EQUAL TO 5 THEN
                SET existing_email TO the 4th element in parts (index 3)

                # Compare email addresses, ignoring case
                IF existing_email (in lowercase) EQUALS email_to_check (in lowercase) THEN
                    CLOSE file
                    RETURN True  # Email exists, exit function

        RETURN False  # No match found after checking all lines

    CATCH FileNotFoundError
        PRINT "No user data found."
        RETURN False  # File does not exist

END FUNCTION
```

4.1.6 Member Sign Up



```
FUNCTION SignUp
    DECLARE MemID AS GLOBAL VARIABLE  # Declare MemID so it can be modified

    # Set up the fields required for sign-up
    SET counter TO ["Firstname", "Lastname", "Email address", "Contact Number", "Password"]
    SET ListDetails TO ["", "", "", "", ""]  # Placeholder for user details
    SET counts TO 0

    # Collect and validate each user detail
    WHILE counts < LENGTH(counter) DO
        # Presence check: Prompt until the user provides input
        WHILE LENGTH(ListDetails[counts]) == 0 DO
            PROMPT user to enter counter[counts]
            SET ListDetails[counts] TO user input
            IF LENGTH(ListDetails[counts]) == 0 THEN
                PRINT "It is mandatory to fill up this field."

        # Check if Contact Number is valid and unique
        IF counter[counts] IS "Contact Number" THEN
            SET ContactNumber TO ValidateContactNumber(ListDetails[3])
            WHILE ContactNumber IS False DO
                PRINT "Invalid phone number! The phone number should be in this format: +60 11 1234 1234"
                PROMPT user to enter counter[counts]
                SET ListDetails[counts] TO user input
                SET ContactNumber TO ValidateContactNumber(ListDetails[3])

        # Check if Password is strong
        IF counter[counts] IS "Password" THEN
            SET StrongUserPassword TO StrongPassword(ListDetails[-1])
            WHILE StrongUserPassword IS NOT True DO
                PRINT "Password is not strong enough! Your password should be at least 8 characters long and must contain the following:"
                PRINT password requirements: uppercase letter, special symbol, digit, lowercase letter
                PROMPT user to enter password again
                SET ListDetails[-1] TO user input
                SET StrongUserPassword TO StrongPassword(ListDetails[-1])

        # Check if email already exists
        SET exists TO False
        IF counter[counts] IS "Email address" THEN
            SET exists TO check_existing_user(ListDetails[2], "members.txt")
            IF exists IS True THEN
                PRINT "This account already exists. Try logging in!"
                SET email TO ListDetails[counts]
                PROMPT user to enter password to log in
                SET Password TO user input
                SET login TO Member_login(email, Password)
                IF login IS False THEN
                    RETURN False
                RETURN  # Exit SignUp if user logs in

        INCREMENT counts BY 1

        # Format names and email before storing
        SET ListDetails[0] TO ListDetails[0].capitalize()
        SET ListDetails[1] TO ListDetails[1].capitalize()
        SET ListDetails[2] TO ListDetails[2].lower()

    # Store data in file
    TRY
        OPEN "members.txt" IN append mode
        SET New_id TO generate_unique_member_id()
        SET MemID TO New_id
        WRITE New_id and ListDetails to file in format: "New_id | Firstname | Lastname | Email | Contact | Password"
        CLOSE file
        PRINT "SignUp Successful!"
    CATCH error
        PRINT "SignUp not successful. An error has occurred! Contact Librarian."
        RETURN False

END FUNCTION
```

4.1.7 Display members' borrowed books List



```
FUNCTION DisplayBorrowedBooks(memID)
    TRY
        # Open the files "MemberBorrowedBooksInfo.txt" and "BookList.txt" for reading
        WITH "MemberBorrowedBooksInfo.txt" AS book_lent_file AND "BookList.txt" AS book_list_file

            FOR EACH line IN book_lent_file
                # Skip if line is empty or does not have enough columns
                IF line IS empty OR LENGTH(line.split(" | ")) < 4 THEN
                    CONTINUE TO NEXT line

                # Extract and clean data
                SET data TO line.split(" | ")
                SET member_id TO data[0].strip()
                SET book_id TO data[1].strip()
                SET Due_Date TO data[2].strip()
                SET Overdue_Fees TO data[3].strip()
                SET PaymentStatus TO data[4].strip()

                # Split book IDs by commas and remove extra whitespace
                SET book_id TO book_id.split(", ")
                FOR EACH i IN RANGE LENGTH(book_id) DO
                    SET book_id[i] TO book_id[i].strip()

                # Check if the member ID matches the provided memID
                IF member_id == memID THEN
                    PRINT "Here are the details about your borrowed books:"

                    # Search for each book's information in the book list
                    FOR EACH book_line IN book_list_file
                        # Skip if book_line is empty or does not have enough columns
                        IF book_line IS empty OR LENGTH(book_line.split(" | ")) < 4 THEN
                            CONTINUE TO NEXT book_line

                        # Extract and clean book information
                        SET book_info TO book_line.split(" | ")
                        SET book_info[0] TO book_info[0].strip()

                        # Check if the book ID matches any borrowed book ID
                        FOR EACH borrowed_id IN book_id DO
                            IF borrowed_id == book_info[0] THEN
                                PRINT "{borrowed_id}. Title: {book_info[1]}, Author: {book_info[2]}, Publisher: {book_info[3]}"
                                PRINT "Due Date: {Due_Date}"
                                PRINT "Overdue fees: RM {Overdue_Fees}"
                                PRINT "Payment Status: {PaymentStatus}"
                    RETURN  # Exit function as books were found

        # If no matching member ID is found in "MemberBorrowedBooksInfo.txt"
        PRINT "You do not have any borrowed books!"
        RETURN False

    CATCH FileNotFoundError
        PRINT "One or both files (MemberBorrowedBooksInfo.txt or BookList.txt) not found."
        RETURN False

END FUNCTION
```

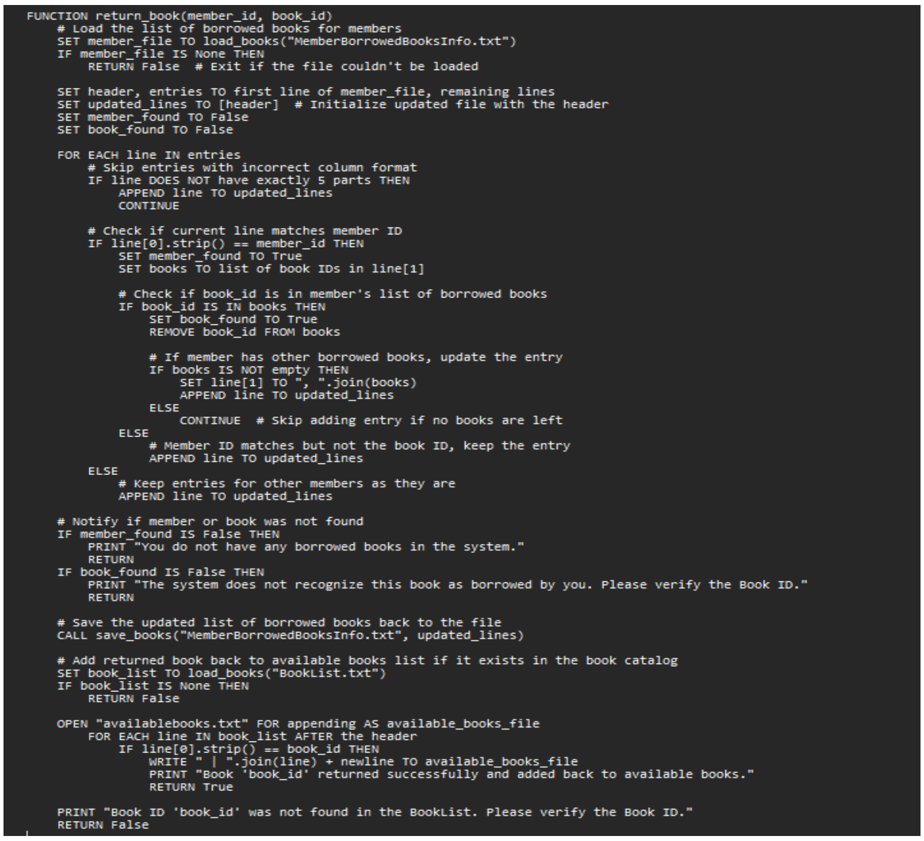
4.1.8 Loading File function

```
FUNCTION load_books(file_name)
    # Load books from a file and return them as a list

    TRY
        # Open the file with the name file_name for reading
        WITH file_name AS file
            # Read each line, strip whitespace, split by " | ", and store in a list
            RETURN [line.strip().split(" | ") FOR EACH line IN file]

    CATCH FileNotFoundError
        # If the file is not found, print an error message and return None
        PRINT "File '{file_name}' not found!"
        RETURN None

END FUNCTION
```

4.1.9 Saving and Storing Updated File Function

```
FUNCTION save_books(file_name, books)
    # Save books to a file

    OPEN file_name FOR writing AS file
        FOR EACH book IN books
            # Join each book's details with " | " separator and write to the file
            WRITE " | ".join(book) + newline TO file
        END FOR
    CLOSE file

END FUNCTION
```

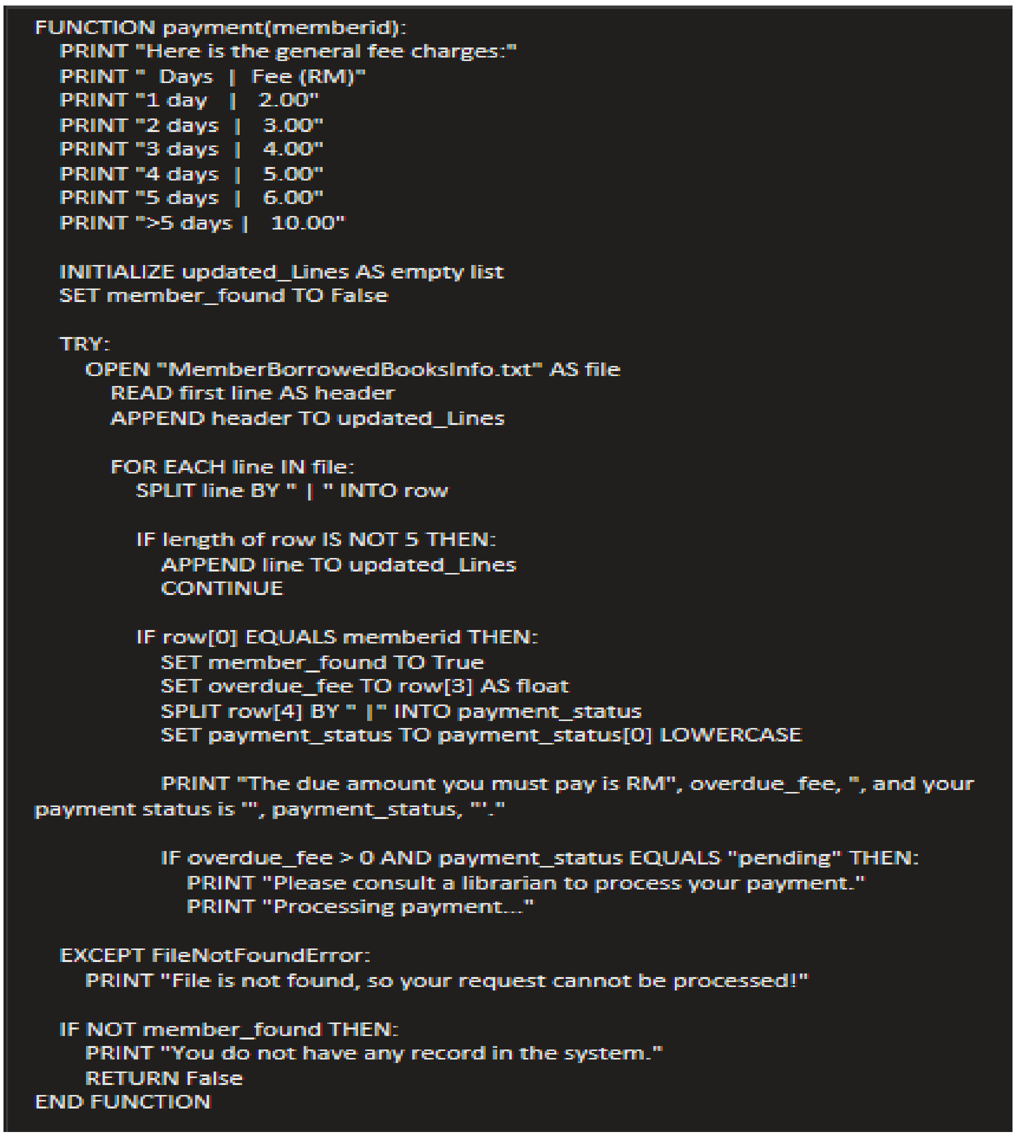
4.1.10 Return Book Function



```
FUNCTION return_book(member_id, book_id)
    # Load the list of borrowed books for members
    SET member_file TO load_books("MemberBorrowedBooksInfo.txt")
    IF member_file IS None THEN
        RETURN False  # Exit if the file couldn't be loaded

    SET header, entries TO first line of member_file, remaining lines
    SET updated_lines TO [header]  # Initialize updated file with the header
    SET member_found TO False
    SET book_found TO False

    FOR EACH line IN entries
        # Skip entries with incorrect column format
        IF line DOES NOT have exactly 5 parts THEN
            APPEND line TO updated_lines
            CONTINUE

        # Check if current line matches member ID
        IF line[0].strip() == member_id THEN
            SET member_found TO True
            SET books TO list of book IDs in line[1]

            # Check if book_id is in member's list of borrowed books
            IF book_id IS IN books THEN
                SET book_found TO True
                REMOVE book_id FROM books

                # If member has other borrowed books, update the entry
                IF books IS NOT empty THEN
                    SET line[1] TO ", ".join(books)
                    APPEND line TO updated_lines
                ELSE
                    CONTINUE  # Skip adding entry if no books are left
            ELSE
                # Member ID matches but not the book ID, keep the entry
                APPEND line TO updated_lines
        ELSE
            # Keep entries for other members as they are
            APPEND line TO updated_lines

    # Notify if member or book was not found
    IF member_found IS False THEN
        PRINT "You do not have any borrowed books in the system."
        RETURN
    IF book_found IS False THEN
        PRINT "The system does not recognize this book as borrowed by you. Please verify the Book ID."
        RETURN

    # Save the updated list of borrowed books back to the file
    CALL save_books("MemberBorrowedBooksInfo.txt", updated_lines)

    # Add returned book back to available books list if it exists in the book catalog
    SET book_list TO load_books("BookList.txt")
    IF book_list IS None THEN
        RETURN False

    OPEN "availablebooks.txt" FOR appending AS available_books_file
        FOR EACH line IN book_list AFTER the header
            IF line[0].strip() == book_id THEN
                WRITE " | ".join(line) + newline TO available_books_file
                PRINT "Book 'book_id' returned successfully and added back to available books."
                RETURN True

    PRINT "Book ID 'book_id' was not found in the BookList. Please verify the Book ID."
    RETURN False
```

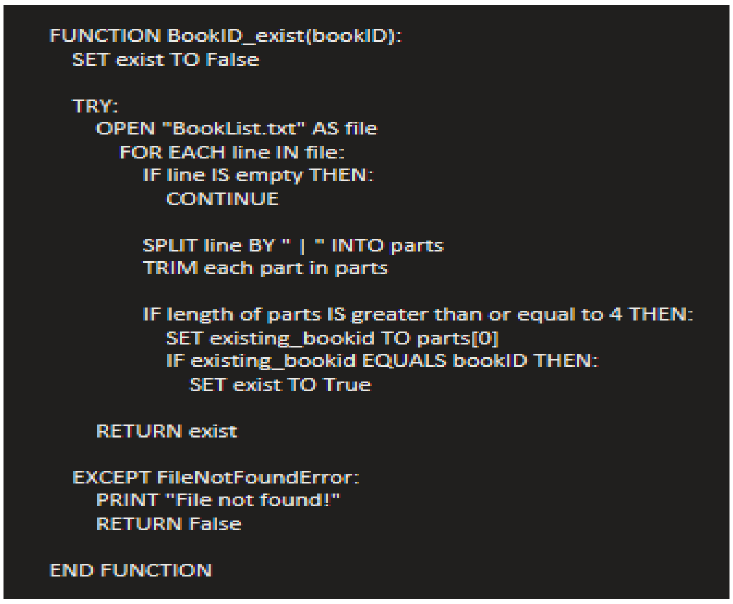
```
END FUNCTION
```

4.1.11 Payment Function



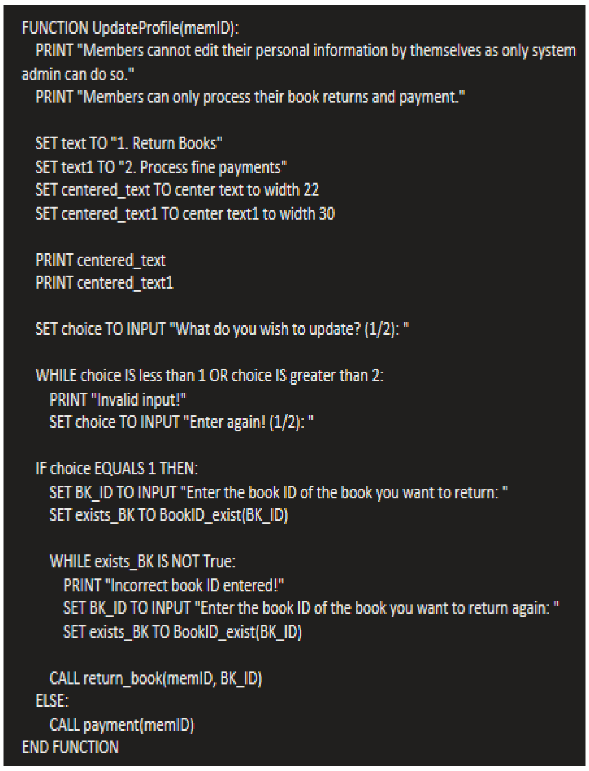
```
FUNCTION payment(memberid):
    PRINT "Here is the general fee charges:"
    PRINT "  Days  |  Fee (RM)"
    PRINT "1 day   |   2.00"
    PRINT "2 days  |   3.00"
    PRINT "3 days  |   4.00"
    PRINT "4 days  |   5.00"
    PRINT "5 days  |   6.00"
    PRINT ">5 days |   10.00"

    INITIALIZE updated_Lines AS empty list
    SET member_found TO False

    TRY:
        OPEN "MemberBorrowedBooksInfo.txt" AS file
            READ first line AS header
            APPEND header TO updated_Lines

            FOR EACH line IN file:
                SPLIT line BY " | " INTO row

                IF length of row IS NOT 5 THEN:
                    APPEND line TO updated_Lines
                    CONTINUE

                IF row[0] EQUALS memberid THEN:
                    SET member_found TO True
                    SET overdue_fee TO row[3] AS float
                    SPLIT row[4] BY " |" INTO payment_status
                    SET payment_status TO payment_status[0] LOWERCASE

                    PRINT "The due amount you must pay is RM", overdue_fee, ", and your
payment status is '", payment_status, "'."

                    IF overdue_fee > 0 AND payment_status EQUALS "pending" THEN:
                        PRINT "Please consult a librarian to process your payment."
                        PRINT "Processing payment..."

    EXCEPT FileNotFoundError:
        PRINT "File is not found, so your request cannot be processed!"

    IF NOT member_found THEN:
        PRINT "You do not have any record in the system."
        RETURN False
END FUNCTION
```

4.1.12 Book ID Validation Function



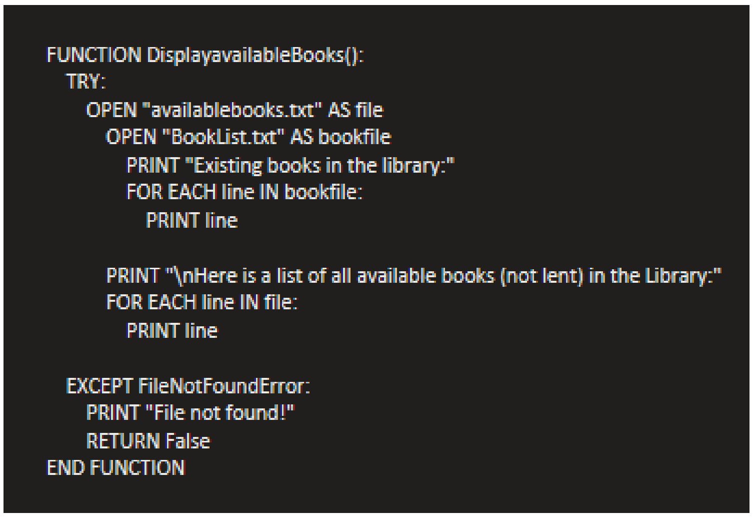
```
FUNCTION BookID_exist(bookID):
   SET exist TO False

   TRY:
      OPEN "BookList.txt" AS file
         FOR EACH line IN file:
            IF line IS empty THEN:
               CONTINUE

            SPLIT line BY " | " INTO parts
            TRIM each part in parts

            IF length of parts IS greater than or equal to 4 THEN:
               SET existing_bookid TO parts[0]
               IF existing_bookid EQUALS bookID THEN:
                  SET exist TO True

      RETURN exist

   EXCEPT FileNotFoundError:
      PRINT "File not found!"
      RETURN False

END FUNCTION
```

4.1.13 Updating members' profile function



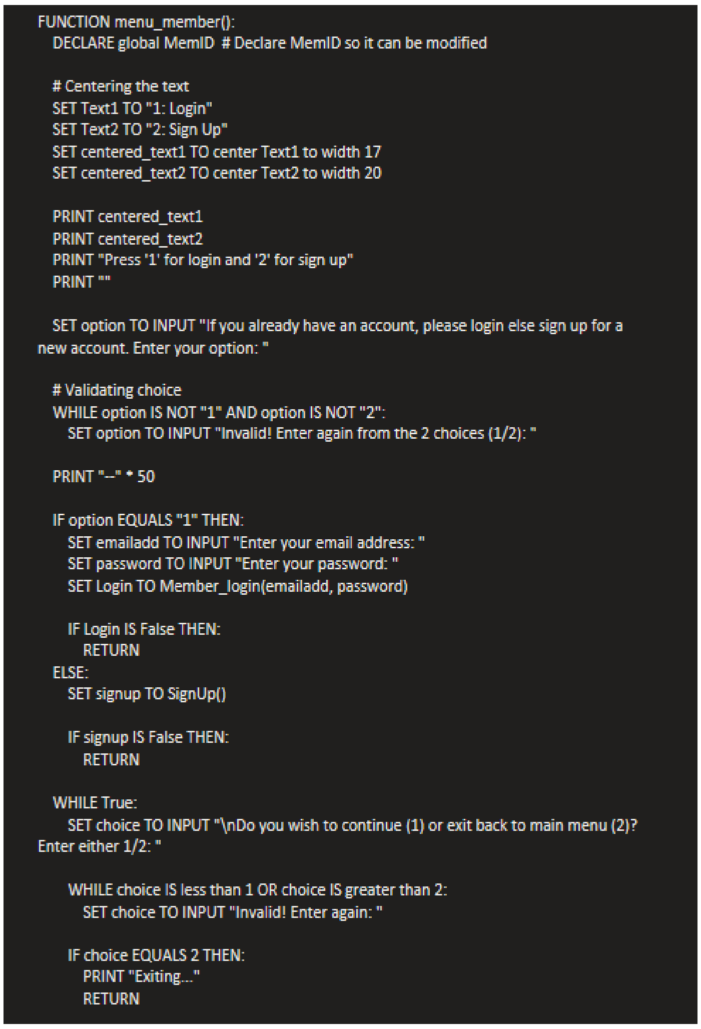
```
FUNCTION UpdateProfile(memID):
   PRINT "Members cannot edit their personal information by themselves as only system
admin can do so."
   PRINT "Members can only process their book returns and payment."

   SET text TO "1. Return Books"
   SET text1 TO "2. Process fine payments"
   SET centered_text TO center text to width 22
   SET centered_text1 TO center text1 to width 30

   PRINT centered_text
   PRINT centered_text1

   SET choice TO INPUT "What do you wish to update? (1/2): "

   WHILE choice IS less than 1 OR choice IS greater than 2:
      PRINT "Invalid input!"
      SET choice TO INPUT "Enter again! (1/2): "

   IF choice EQUALS 1 THEN:
      SET BK_ID TO INPUT "Enter the book ID of the book you want to return: "
      SET exists_BK TO BookID_exist(BK_ID)

      WHILE exists_BK IS NOT True:
         PRINT "Incorrect book ID entered!"
         SET BK_ID TO INPUT "Enter the book ID of the book you want to return again: "
         SET exists_BK TO BookID_exist(BK_ID)

      CALL return_book(memID, BK_ID)
   ELSE:
      CALL payment(memID)
END FUNCTION
```

4.1.14 Displaying available Books list

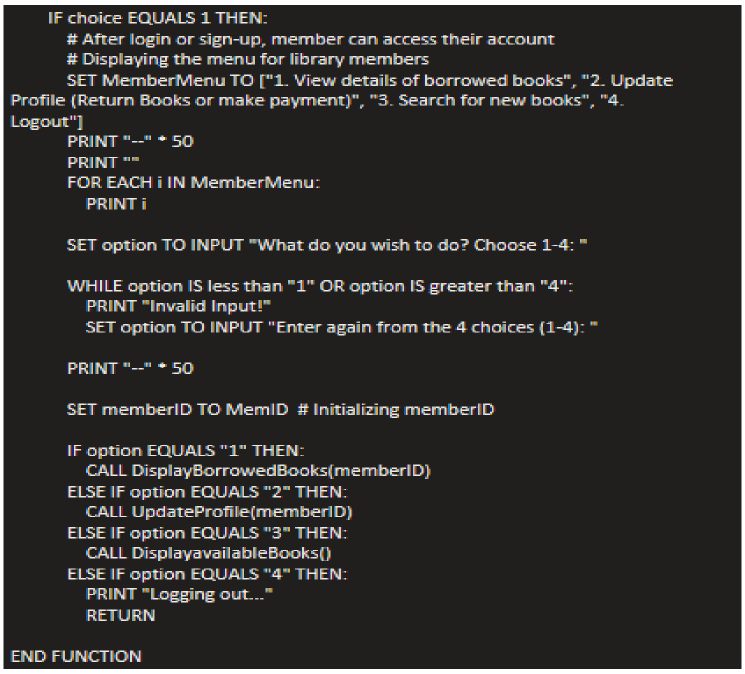


```
FUNCTION DisplayavailableBooks():
  TRY:
    OPEN "availablebooks.txt" AS file
      OPEN "BookList.txt" AS bookfile
        PRINT "Existing books in the library:"
        FOR EACH line IN bookfile:
          PRINT line

      PRINT "\nHere is a list of all available books (not lent) in the Library:"
      FOR EACH line IN file:
        PRINT line

  EXCEPT FileNotFoundError:
    PRINT "File not found!"
    RETURN False
END FUNCTION
```

4.1.15 Members' menu



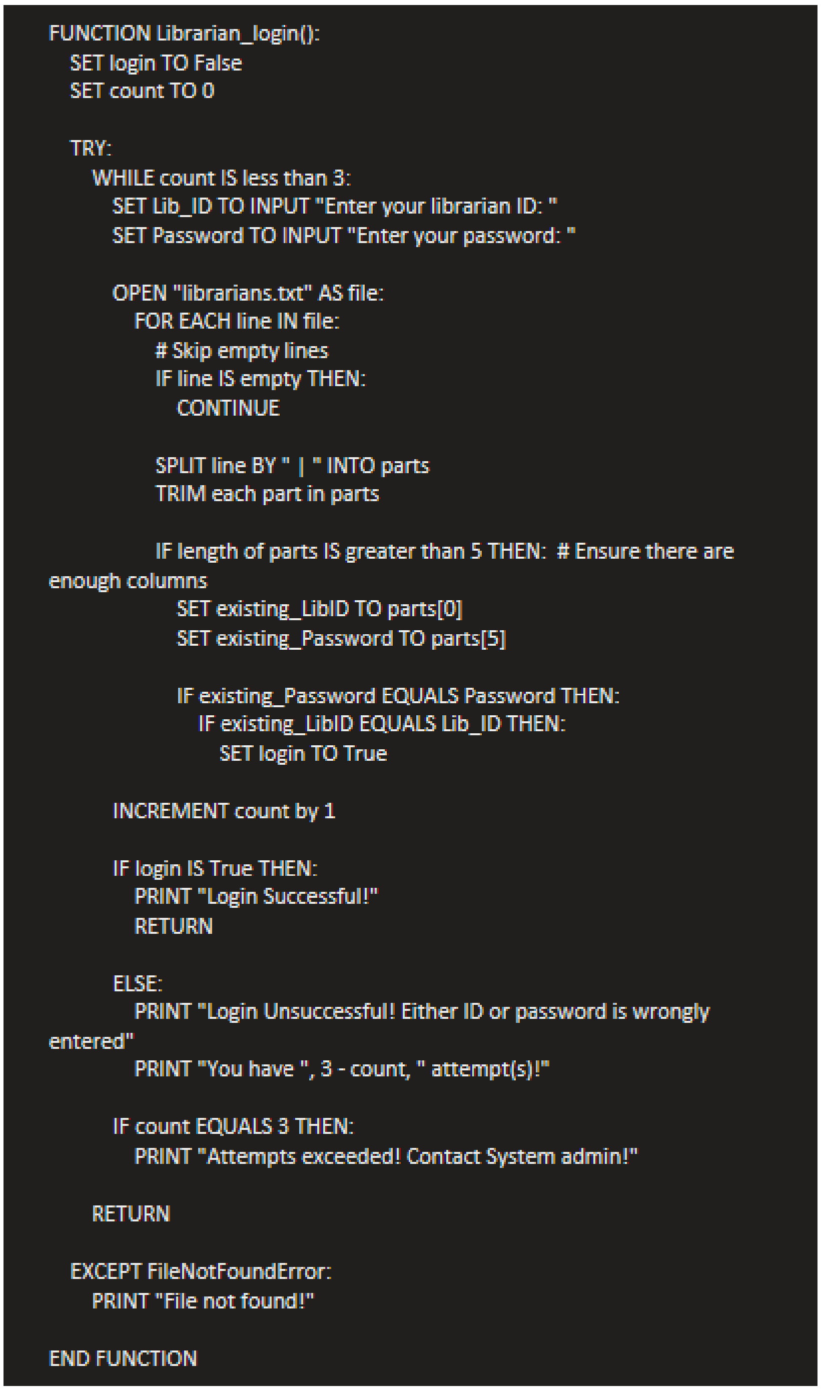
```
FUNCTION menu_member():
    DECLARE global MemID  # Declare MemID so it can be modified

    # Centering the text
    SET Text1 TO "1: Login"
    SET Text2 TO "2: Sign Up"
    SET centered_text1 TO center Text1 to width 17
    SET centered_text2 TO center Text2 to width 20

    PRINT centered_text1
    PRINT centered_text2
    PRINT "Press '1' for login and '2' for sign up"
    PRINT ""

    SET option TO INPUT "If you already have an account, please login else sign up for a
new account. Enter your option: "

    # Validating choice
    WHILE option IS NOT "1" AND option IS NOT "2":
        SET option TO INPUT "Invalid! Enter again from the 2 choices (1/2): "

    PRINT "--" * 50

    IF option EQUALS "1" THEN:
        SET emailadd TO INPUT "Enter your email address: "
        SET password TO INPUT "Enter your password: "
        SET Login TO Member_login(emailadd, password)

        IF Login IS False THEN:
            RETURN
    ELSE:
        SET signup TO SignUp()

        IF signup IS False THEN:
            RETURN

    WHILE True:
        SET choice TO INPUT "\nDo you wish to continue (1) or exit back to main menu (2)?
Enter either 1/2: "

        WHILE choice IS less than 1 OR choice IS greater than 2:
            SET choice TO INPUT "Invalid! Enter again: "

        IF choice EQUALS 2 THEN:
            PRINT "Exiting..."
            RETURN
```



```
        IF choice EQUALS 1 THEN:
            # After login or sign-up, member can access their account
            # Displaying the menu for library members
            SET MemberMenu TO ["1. View details of borrowed books", "2. Update
Profile (Return Books or make payment)", "3. Search for new books", "4.
Logout"]
            PRINT "--" * 50
            PRINT ""
            FOR EACH i IN MemberMenu:
                PRINT i

            SET option TO INPUT "What do you wish to do? Choose 1-4: "

            WHILE option IS less than "1" OR option IS greater than "4":
                PRINT "Invalid Input!"
                SET option TO INPUT "Enter again from the 4 choices (1-4): "

            PRINT "--" * 50

            SET memberID TO MemID  # Initializing memberID

            IF option EQUALS "1" THEN:
                CALL DisplayBorrowedBooks(memberID)
            ELSE IF option EQUALS "2" THEN:
                CALL UpdateProfile(memberID)
            ELSE IF option EQUALS "3" THEN:
                CALL DisplayavailableBooks()
            ELSE IF option EQUALS "4" THEN:
                PRINT "Logging out..."
                RETURN

END FUNCTION
```

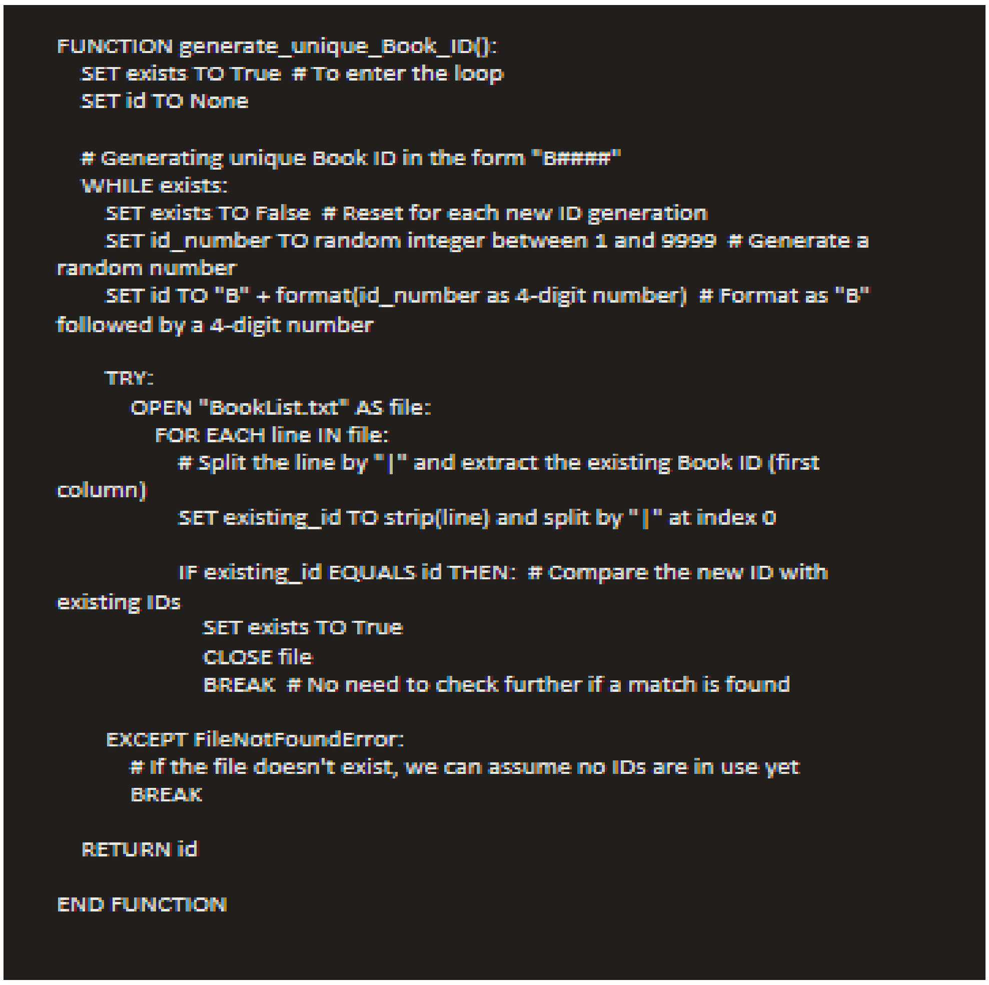
4.2 Librarian User

4.2.1 Librarian Login



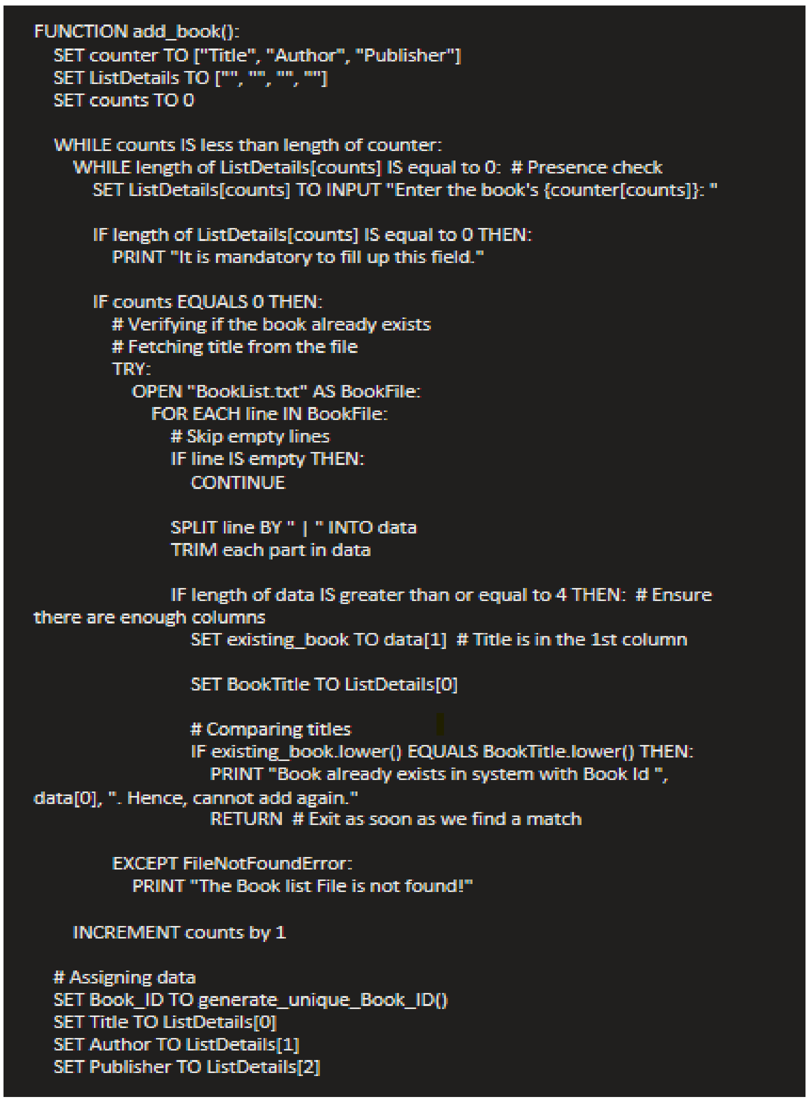
```
FUNCTION Librarian_login():
  SET login TO False
  SET count TO 0

  TRY:
    WHILE count IS less than 3:
      SET Lib_ID TO INPUT "Enter your librarian ID: "
      SET Password TO INPUT "Enter your password: "

      OPEN "librarians.txt" AS file:
        FOR EACH line IN file:
          # Skip empty lines
          IF line IS empty THEN:
            CONTINUE

          SPLIT line BY " | " INTO parts
          TRIM each part in parts

          IF length of parts IS greater than 5 THEN:  # Ensure there are
enough columns
            SET existing_LibID TO parts[0]
            SET existing_Password TO parts[5]

            IF existing_Password EQUALS Password THEN:
              IF existing_LibID EQUALS Lib_ID THEN:
                SET login TO True

      INCREMENT count by 1

      IF login IS True THEN:
        PRINT "Login Successful!"
        RETURN

      ELSE:
        PRINT "Login Unsuccessful! Either ID or password is wrongly
entered"
        PRINT "You have ", 3 - count, " attempt(s)!"

      IF count EQUALS 3 THEN:
        PRINT "Attempts exceeded! Contact System admin!"

    RETURN

  EXCEPT FileNotFoundError:
    PRINT "File not found!"

END FUNCTION
```

4.2.2 Book ID Randomising

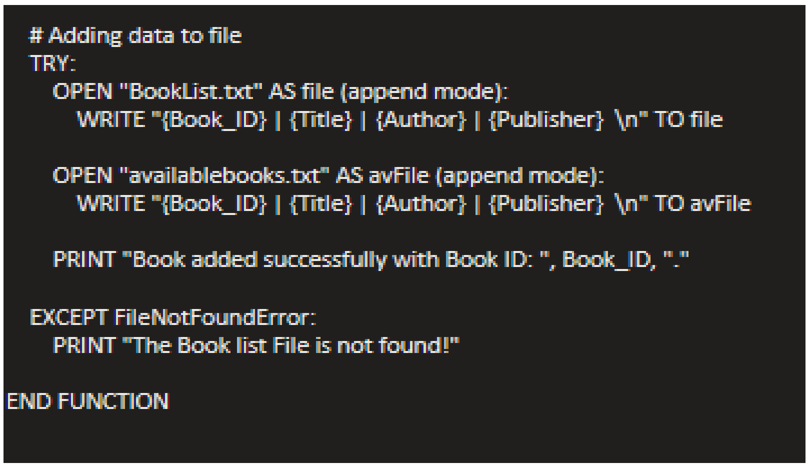


```
FUNCTION generate_unique_Book_ID():
    SET exists TO True  # To enter the loop
    SET id TO None

    # Generating unique Book ID in the form "B####"
    WHILE exists:
        SET exists TO False  # Reset for each new ID generation
        SET id_number TO random integer between 1 and 9999  # Generate a
random number
        SET id TO "B" + format(id_number as 4-digit number)  # Format as "B"
followed by a 4-digit number

        TRY:
            OPEN "BookList.txt" AS file:
                FOR EACH line IN file:
                    # Split the line by "|" and extract the existing Book ID (first
column)
                    SET existing_id TO strip(line) and split by "|" at index 0

                    IF existing_id EQUALS id THEN:  # Compare the new ID with
existing IDs
                        SET exists TO True
                        CLOSE file
                        BREAK  # No need to check further if a match is found

        EXCEPT FileNotFoundError:
            # If the file doesn't exist, we can assume no IDs are in use yet
            BREAK

    RETURN id

END FUNCTION
```

4.2.3 Add new book to existing book lists

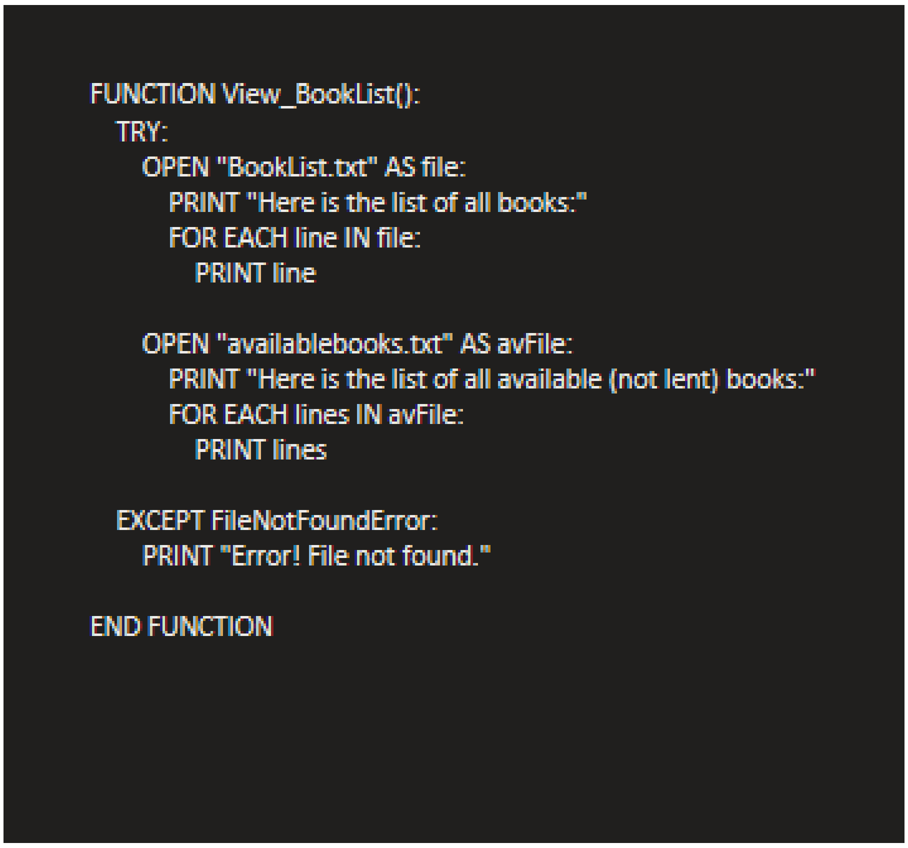


```
FUNCTION add_book():
  SET counter TO ["Title", "Author", "Publisher"]
  SET ListDetails TO ["", "", "", ""]
  SET counts TO 0

  WHILE counts IS less than length of counter:
    WHILE length of ListDetails[counts] IS equal to 0:  # Presence check
      SET ListDetails[counts] TO INPUT "Enter the book's {counter[counts]}: "

      IF length of ListDetails[counts] IS equal to 0 THEN:
        PRINT "It is mandatory to fill up this field."

      IF counts EQUALS 0 THEN:
        # Verifying if the book already exists
        # Fetching title from the file
        TRY:
          OPEN "BookList.txt" AS BookFile:
            FOR EACH line IN BookFile:
              # Skip empty lines
              IF line IS empty THEN:
                CONTINUE

              SPLIT line BY " | " INTO data
              TRIM each part in data

              IF length of data IS greater than or equal to 4 THEN:  # Ensure
there are enough columns
                SET existing_book TO data[1]  # Title is in the 1st column

                SET BookTitle TO ListDetails[0]

                # Comparing titles
                IF existing_book.lower() EQUALS BookTitle.lower() THEN:
                  PRINT "Book already exists in system with Book Id ",
data[0], ". Hence, cannot add again."
                  RETURN  # Exit as soon as we find a match

        EXCEPT FileNotFoundError:
          PRINT "The Book list File is not found!"

    INCREMENT counts by 1

  # Assigning data
  SET Book_ID TO generate_unique_Book_ID()
  SET Title TO ListDetails[0]
  SET Author TO ListDetails[1]
  SET Publisher TO ListDetails[2]
```



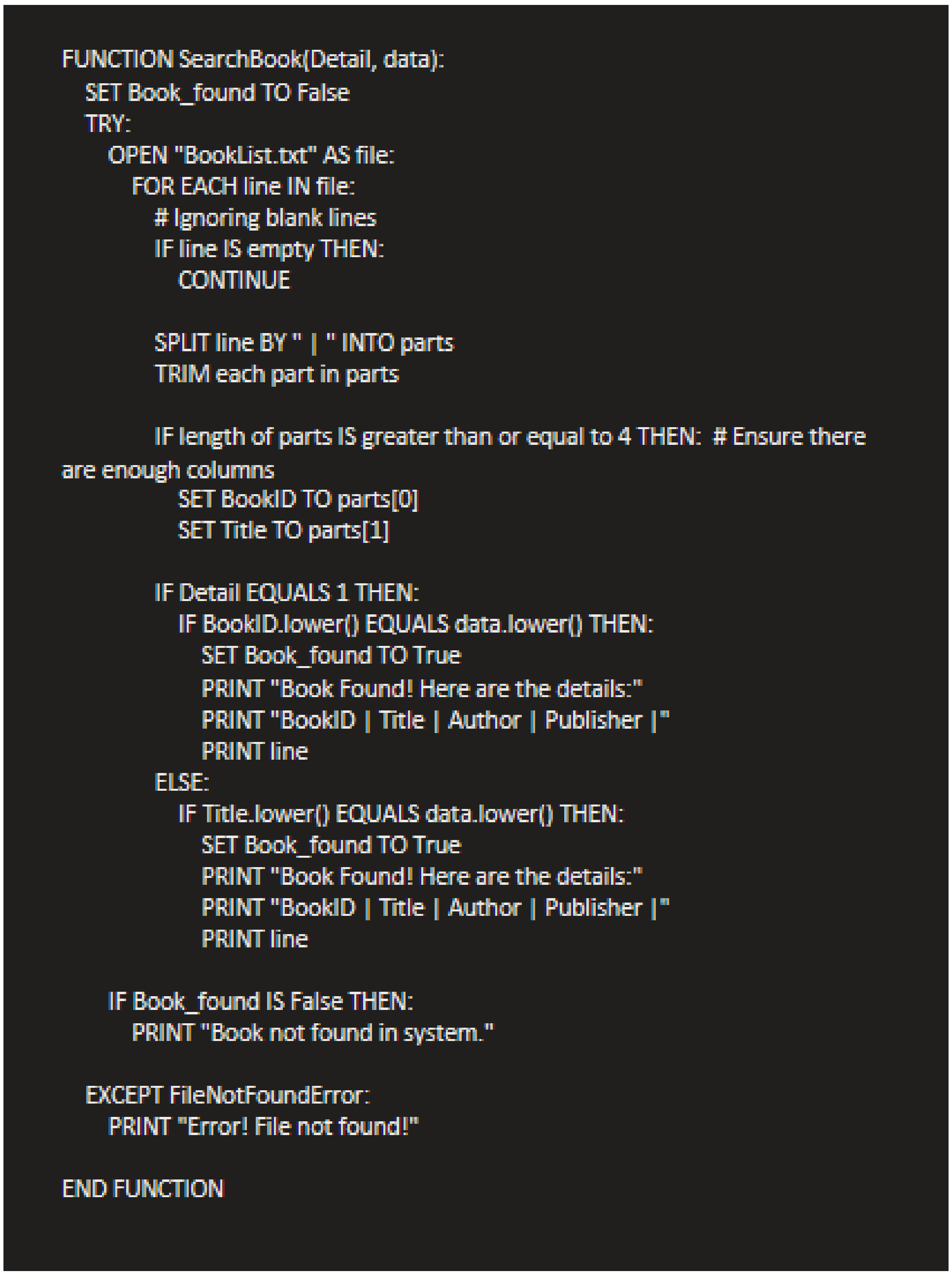
```
 # Adding data to file
 TRY:
   OPEN "BookList.txt" AS file (append mode):
     WRITE "{Book_ID} | {Title} | {Author} | {Publisher}  \n" TO file

   OPEN "availablebooks.txt" AS avFile (append mode):
     WRITE "{Book_ID} | {Title} | {Author} | {Publisher}  \n" TO avFile

   PRINT "Book added successfully with Book ID: ", Book_ID, "."

 EXCEPT FileNotFoundError:
   PRINT "The Book list File is not found!"

END FUNCTION
```

4.2.4 View Book List Function



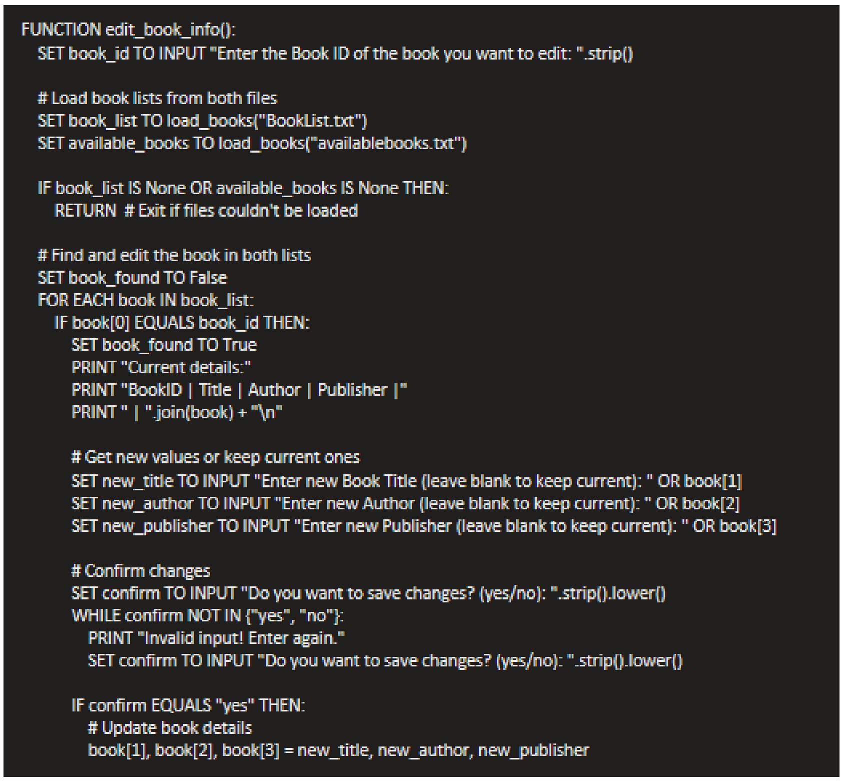
```
FUNCTION View_BookList():
  TRY:
    OPEN "BookList.txt" AS file:
      PRINT "Here is the list of all books:"
      FOR EACH line IN file:
        PRINT line

    OPEN "availablebooks.txt" AS avFile:
      PRINT "Here is the list of all available (not lent) books:"
      FOR EACH lines IN avFile:
        PRINT lines

  EXCEPT FileNotFoundError:
    PRINT "Error! File not found."

END FUNCTION
```

4.2.5 Search specific book from book list

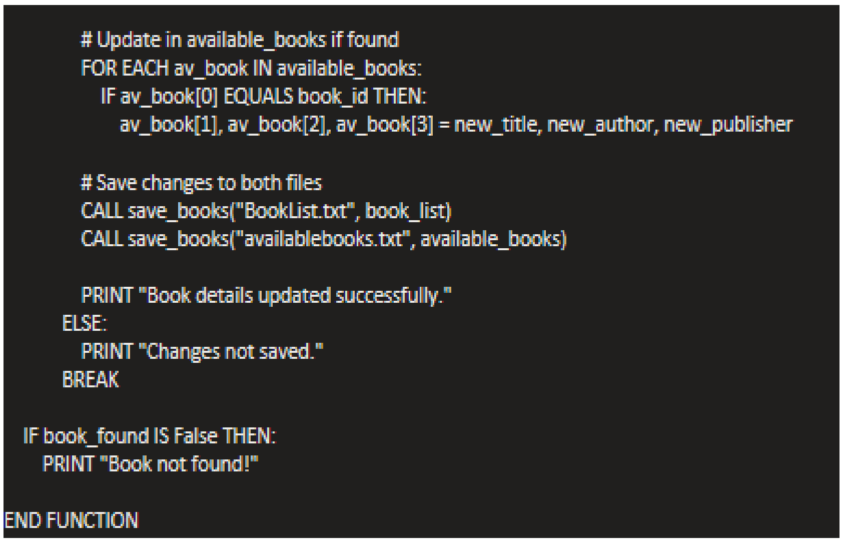


```
FUNCTION SearchBook(Detail, data):
   SET Book_found TO False
   TRY:
      OPEN "BookList.txt" AS file:
         FOR EACH line IN file:
            # Ignoring blank lines
            IF line IS empty THEN:
               CONTINUE

            SPLIT line BY " | " INTO parts
            TRIM each part in parts

            IF length of parts IS greater than or equal to 4 THEN:  # Ensure there
are enough columns
               SET BookID TO parts[0]
               SET Title TO parts[1]

            IF Detail EQUALS 1 THEN:
               IF BookID.lower() EQUALS data.lower() THEN:
                  SET Book_found TO True
                  PRINT "Book Found! Here are the details:"
                  PRINT "BookID | Title | Author | Publisher |"
                  PRINT line
            ELSE:
               IF Title.lower() EQUALS data.lower() THEN:
                  SET Book_found TO True
                  PRINT "Book Found! Here are the details:"
                  PRINT "BookID | Title | Author | Publisher |"
                  PRINT line

      IF Book_found IS False THEN:
         PRINT "Book not found in system."

   EXCEPT FileNotFoundError:
      PRINT "Error! File not found!"

END FUNCTION
```

4.2.6 Edit book information from book list

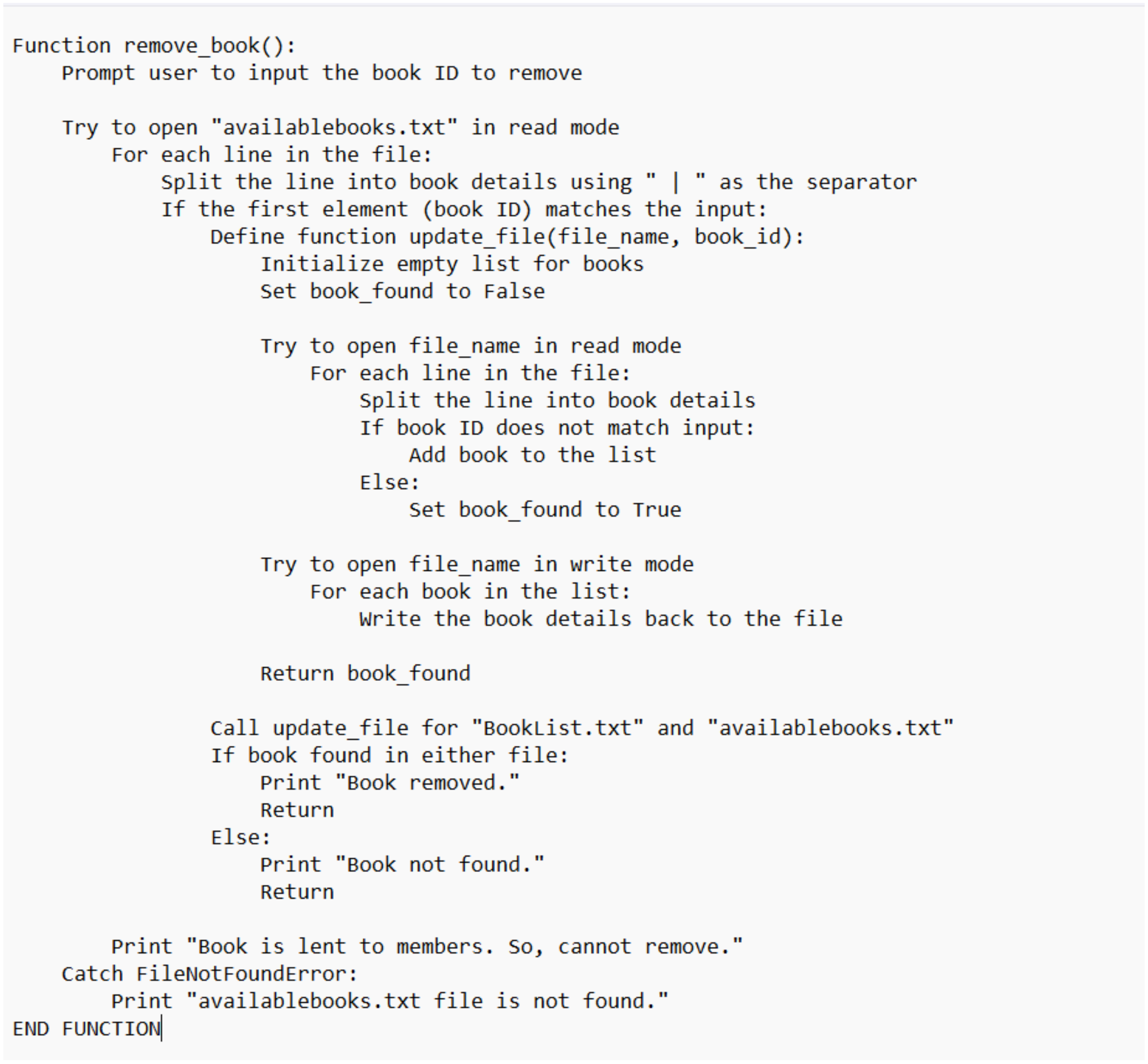


```
FUNCTION edit_book_info():
    SET book_id TO INPUT "Enter the Book ID of the book you want to edit: ".strip()

    # Load book lists from both files
    SET book_list TO load_books("BookList.txt")
    SET available_books TO load_books("availablebooks.txt")

    IF book_list IS None OR available_books IS None THEN:
        RETURN  # Exit if files couldn't be loaded

    # Find and edit the book in both lists
    SET book_found TO False
    FOR EACH book IN book_list:
        IF book[0] EQUALS book_id THEN:
            SET book_found TO True
            PRINT "Current details:"
            PRINT "BookID | Title | Author | Publisher |"
            PRINT " | ".join(book) + "\n"

            # Get new values or keep current ones
            SET new_title TO INPUT "Enter new Book Title (leave blank to keep current): " OR book[1]
            SET new_author TO INPUT "Enter new Author (leave blank to keep current): " OR book[2]
            SET new_publisher TO INPUT "Enter new Publisher (leave blank to keep current): " OR book[3]

            # Confirm changes
            SET confirm TO INPUT "Do you want to save changes? (yes/no): ".strip().lower()
            WHILE confirm NOT IN {"yes", "no"}:
                PRINT "Invalid input! Enter again."
                SET confirm TO INPUT "Do you want to save changes? (yes/no): ".strip().lower()

            IF confirm EQUALS "yes" THEN:
                # Update book details
                book[1], book[2], book[3] = new_title, new_author, new_publisher
```



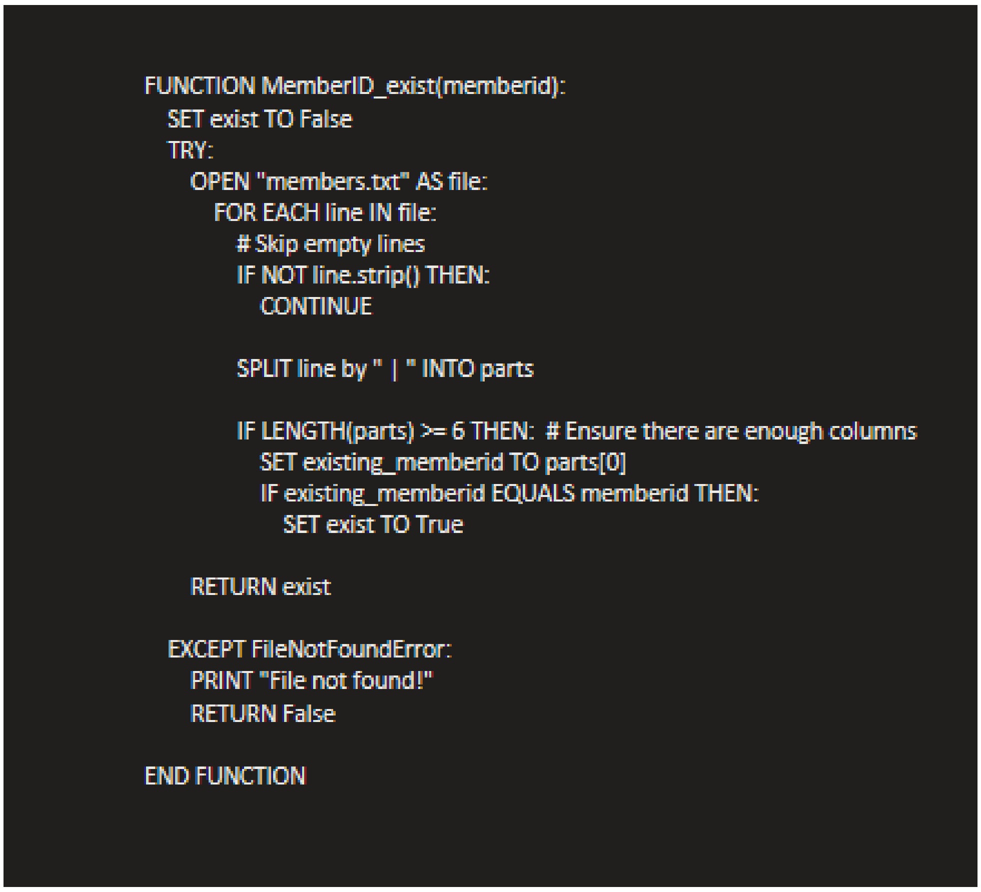
```
                # Update in available_books if found
                FOR EACH av_book IN available_books:
                    IF av_book[0] EQUALS book_id THEN:
                        av_book[1], av_book[2], av_book[3] = new_title, new_author, new_publisher

                # Save changes to both files
                CALL save_books("BookList.txt", book_list)
                CALL save_books("availablebooks.txt", available_books)

                PRINT "Book details updated successfully."
            ELSE:
                PRINT "Changes not saved."
            BREAK

    IF book_found IS False THEN:
        PRINT "Book not found!"

END FUNCTION
```

4.2.7 Remove specific book from book list



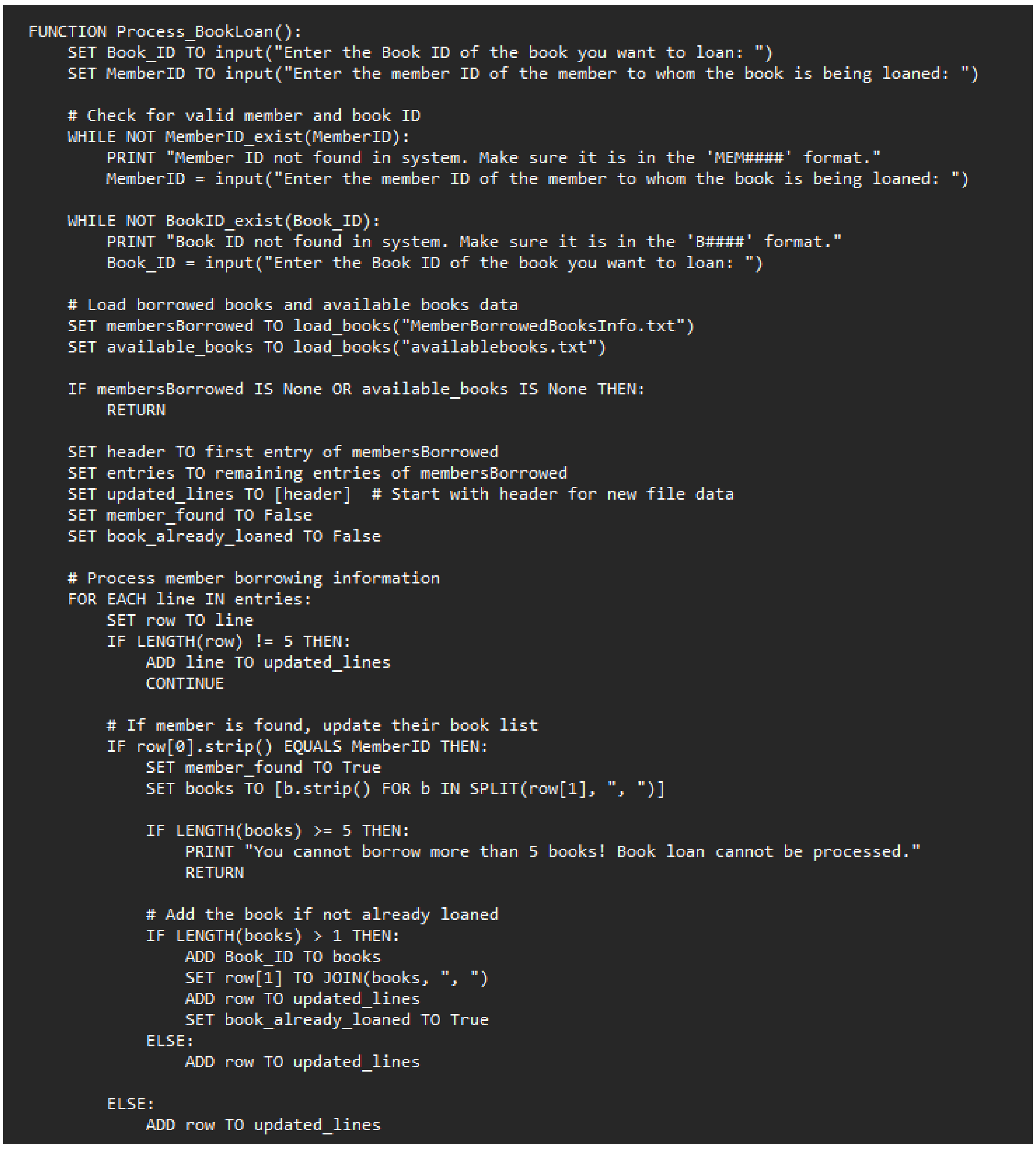
```
Function remove_book():
    Prompt user to input the book ID to remove

    Try to open "availablebooks.txt" in read mode
        For each line in the file:
            Split the line into book details using " | " as the separator
            If the first element (book ID) matches the input:
                Define function update_file(file_name, book_id):
                    Initialize empty list for books
                    Set book_found to False

                    Try to open file_name in read mode
                        For each line in the file:
                            Split the line into book details
                            If book ID does not match input:
                                Add book to the list
                            Else:
                                Set book_found to True

                    Try to open file_name in write mode
                        For each book in the list:
                            Write the book details back to the file

                    Return book_found

                Call update_file for "BookList.txt" and "availablebooks.txt"
                If book found in either file:
                    Print "Book removed."
                    Return
                Else:
                    Print "Book not found."
                    Return

        Print "Book is lent to members. So, cannot remove."
    Catch FileNotFoundError:
        Print "availablebooks.txt file is not found."
END FUNCTION
```

4.2.8 Validating Member ID

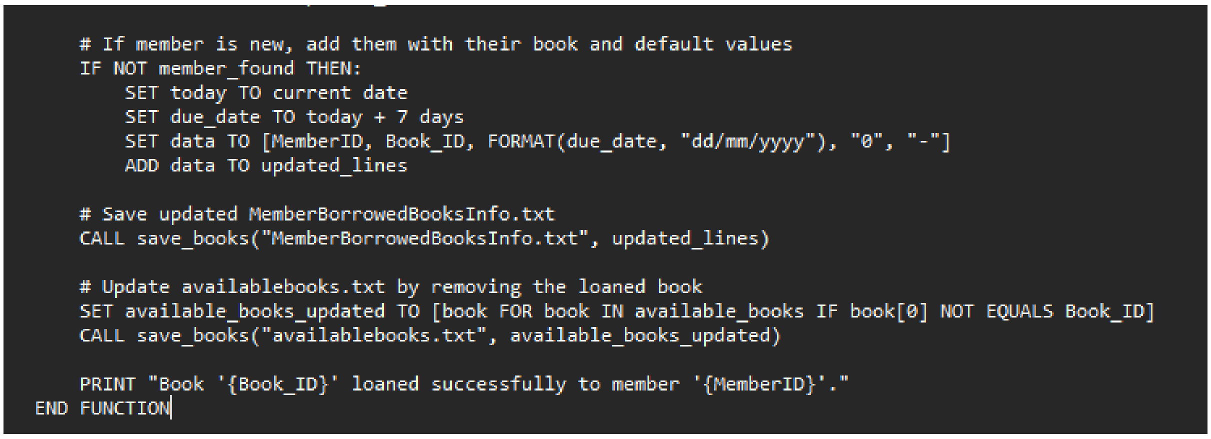


```
FUNCTION MemberID_exist(memberid):
    SET exist TO False
    TRY:
        OPEN "members.txt" AS file:
            FOR EACH line IN file:
                # Skip empty lines
                IF NOT line.strip() THEN:
                    CONTINUE

                SPLIT line by " | " INTO parts

                IF LENGTH(parts) >= 6 THEN:  # Ensure there are enough columns
                    SET existing_memberid TO parts[0]
                    IF existing_memberid EQUALS memberid THEN:
                        SET exist TO True

        RETURN exist

    EXCEPT FileNotFoundError:
        PRINT "File not found!"
        RETURN False

END FUNCTION
```

4.2.9 Processing a book loan

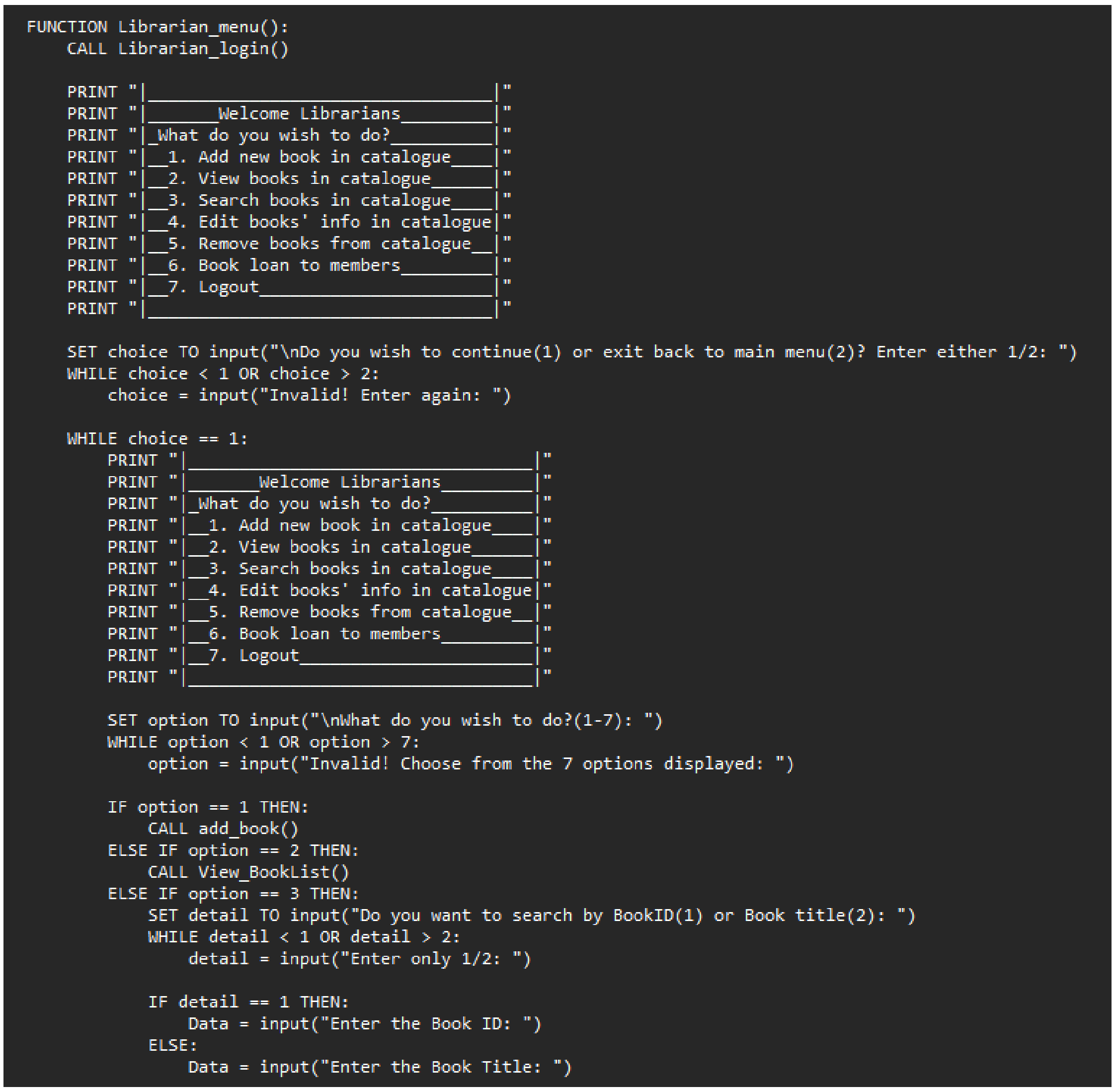


```
FUNCTION Process_BookLoan():
    SET Book_ID TO input("Enter the Book ID of the book you want to loan: ")
    SET MemberID TO input("Enter the member ID of the member to whom the book is being loaned: ")

    # Check for valid member and book ID
    WHILE NOT MemberID_exist(MemberID):
        PRINT "Member ID not found in system. Make sure it is in the 'MEM####' format."
        MemberID = input("Enter the member ID of the member to whom the book is being loaned: ")

    WHILE NOT BookID_exist(Book_ID):
        PRINT "Book ID not found in system. Make sure it is in the 'B####' format."
        Book_ID = input("Enter the Book ID of the book you want to loan: ")

    # Load borrowed books and available books data
    SET membersBorrowed TO load_books("MemberBorrowedBooksInfo.txt")
    SET available_books TO load_books("availablebooks.txt")

    IF membersBorrowed IS None OR available_books IS None THEN:
        RETURN

    SET header TO first entry of membersBorrowed
    SET entries TO remaining entries of membersBorrowed
    SET updated_lines TO [header]  # Start with header for new file data
    SET member_found TO False
    SET book_already_loaned TO False

    # Process member borrowing information
    FOR EACH line IN entries:
        SET row TO line
        IF LENGTH(row) != 5 THEN:
            ADD line TO updated_lines
            CONTINUE

        # If member is found, update their book list
        IF row[0].strip() EQUALS MemberID THEN:
            SET member_found TO True
            SET books TO [b.strip() FOR b IN SPLIT(row[1], ", ")]

            IF LENGTH(books) >= 5 THEN:
                PRINT "You cannot borrow more than 5 books! Book loan cannot be processed."
                RETURN

            # Add the book if not already loaned
            IF LENGTH(books) > 1 THEN:
                ADD Book_ID TO books
                SET row[1] TO JOIN(books, ", ")
                ADD row TO updated_lines
                SET book_already_loaned TO True
            ELSE:
                ADD row TO updated_lines

        ELSE:
            ADD row TO updated_lines
```

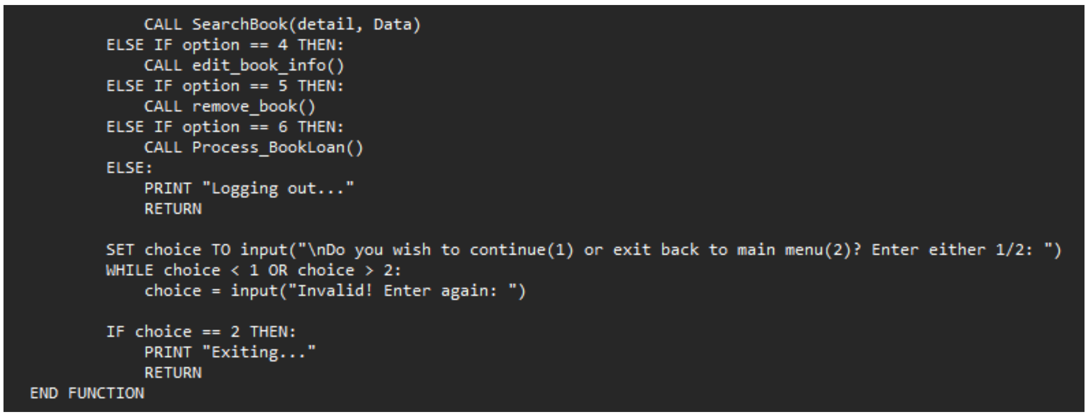


```
    # If member is new, add them with their book and default values
    IF NOT member_found THEN:
        SET today TO current date
        SET due_date TO today + 7 days
        SET data TO [MemberID, Book_ID, FORMAT(due_date, "dd/mm/yyyy"), "0", "-"]
        ADD data TO updated_lines

    # Save updated MemberBorrowedBooksInfo.txt
    CALL save_books("MemberBorrowedBooksInfo.txt", updated_lines)

    # Update availablebooks.txt by removing the loaned book
    SET available_books_updated TO [book FOR book IN available_books IF book[0] NOT EQUALS Book_ID]
    CALL save_books("availablebooks.txt", available_books_updated)

    PRINT "Book '{Book_ID}' loaned successfully to member '{MemberID}'."
END FUNCTION
```

4.2.10 Librarians' menu

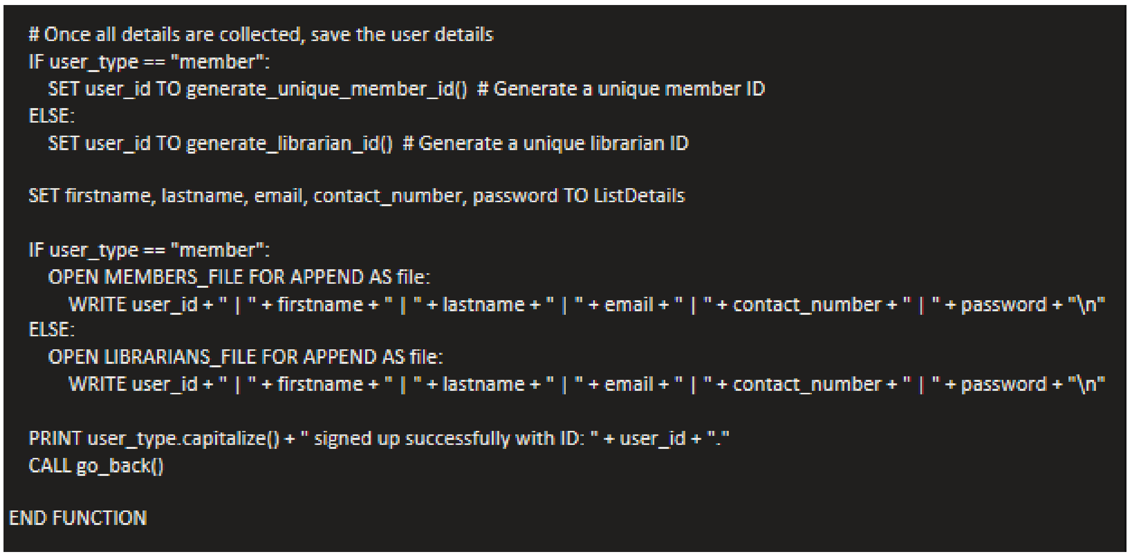


```
FUNCTION Librarian_menu():
    CALL Librarian_login()

    PRINT "|__________________________________|"
    PRINT "|_______Welcome Librarians_________|"
    PRINT "|_What do you wish to do?__________|"
    PRINT "|__1. Add new book in catalogue____|"
    PRINT "|__2. View books in catalogue______|"
    PRINT "|__3. Search books in catalogue____|"
    PRINT "|__4. Edit books' info in catalogue|"
    PRINT "|__5. Remove books from catalogue__|"
    PRINT "|__6. Book loan to members_________|"
    PRINT "|__7. Logout_______________________|"
    PRINT "|__________________________________|"

    SET choice TO input("\nDo you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: ")
    WHILE choice < 1 OR choice > 2:
        choice = input("Invalid! Enter again: ")

    WHILE choice == 1:
        PRINT "|__________________________________|"
        PRINT "|_______Welcome Librarians_________|"
        PRINT "|_What do you wish to do?__________|"
        PRINT "|__1. Add new book in catalogue____|"
        PRINT "|__2. View books in catalogue______|"
        PRINT "|__3. Search books in catalogue____|"
        PRINT "|__4. Edit books' info in catalogue|"
        PRINT "|__5. Remove books from catalogue__|"
        PRINT "|__6. Book loan to members_________|"
        PRINT "|__7. Logout_______________________|"
        PRINT "|__________________________________|"

        SET option TO input("\nWhat do you wish to do?(1-7): ")
        WHILE option < 1 OR option > 7:
            option = input("Invalid! Choose from the 7 options displayed: ")

        IF option == 1 THEN:
            CALL add_book()
        ELSE IF option == 2 THEN:
            CALL View_BookList()
        ELSE IF option == 3 THEN:
            SET detail TO input("Do you want to search by BookID(1) or Book title(2): ")
            WHILE detail < 1 OR detail > 2:
                detail = input("Enter only 1/2: ")

            IF detail == 1 THEN:
                Data = input("Enter the Book ID: ")
            ELSE:
                Data = input("Enter the Book Title: ")
```



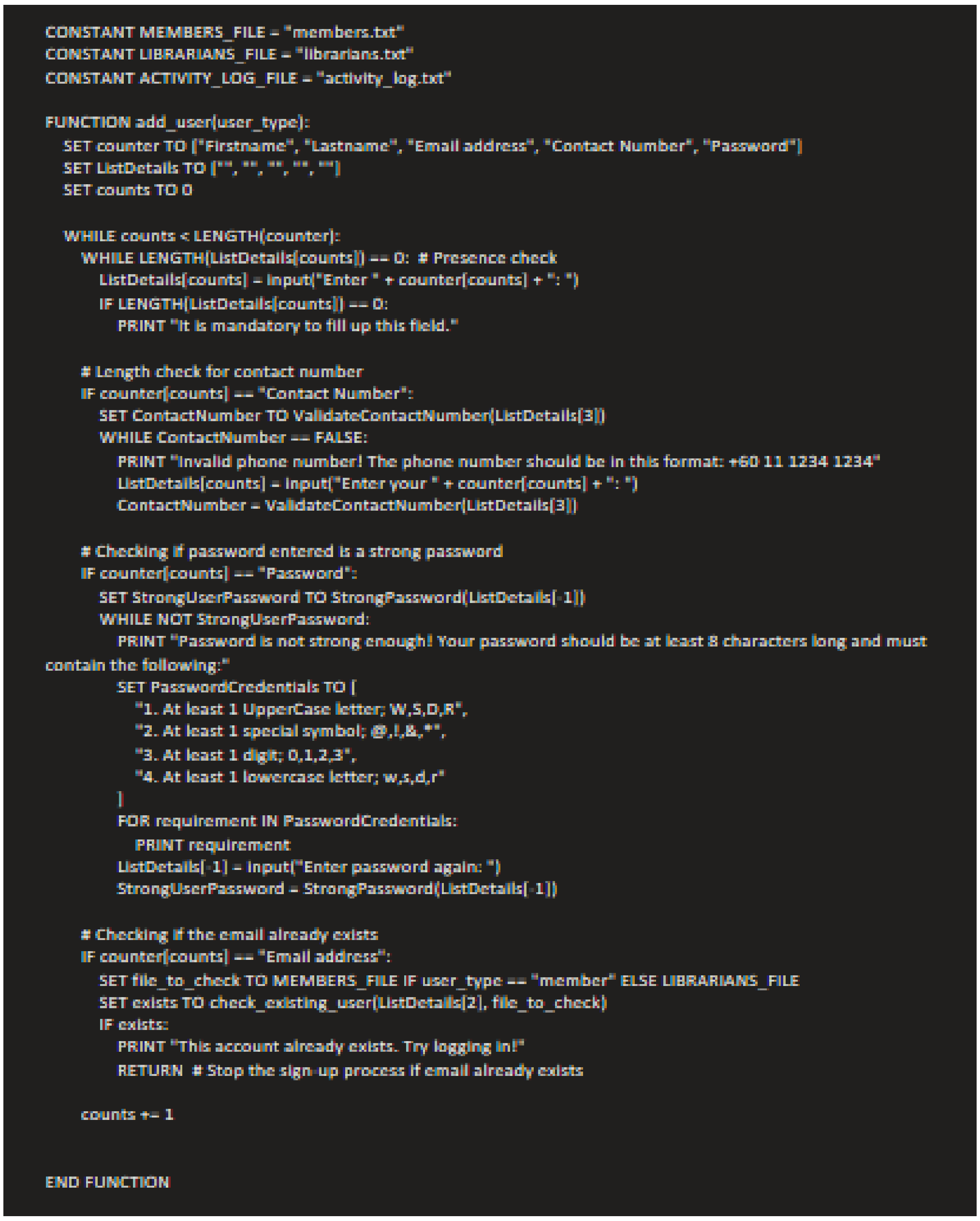
```
            CALL SearchBook(detail, Data)
        ELSE IF option == 4 THEN:
            CALL edit_book_info()
        ELSE IF option == 5 THEN:
            CALL remove_book()
        ELSE IF option == 6 THEN:
            CALL Process_BookLoan()
        ELSE:
            PRINT "Logging out..."
            RETURN

        SET choice TO input("\nDo you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: ")
        WHILE choice < 1 OR choice > 2:
            choice = input("Invalid! Enter again: ")

        IF choice == 2 THEN:
            PRINT "Exiting..."
            RETURN
END FUNCTION
```

4.3 System Administrator User



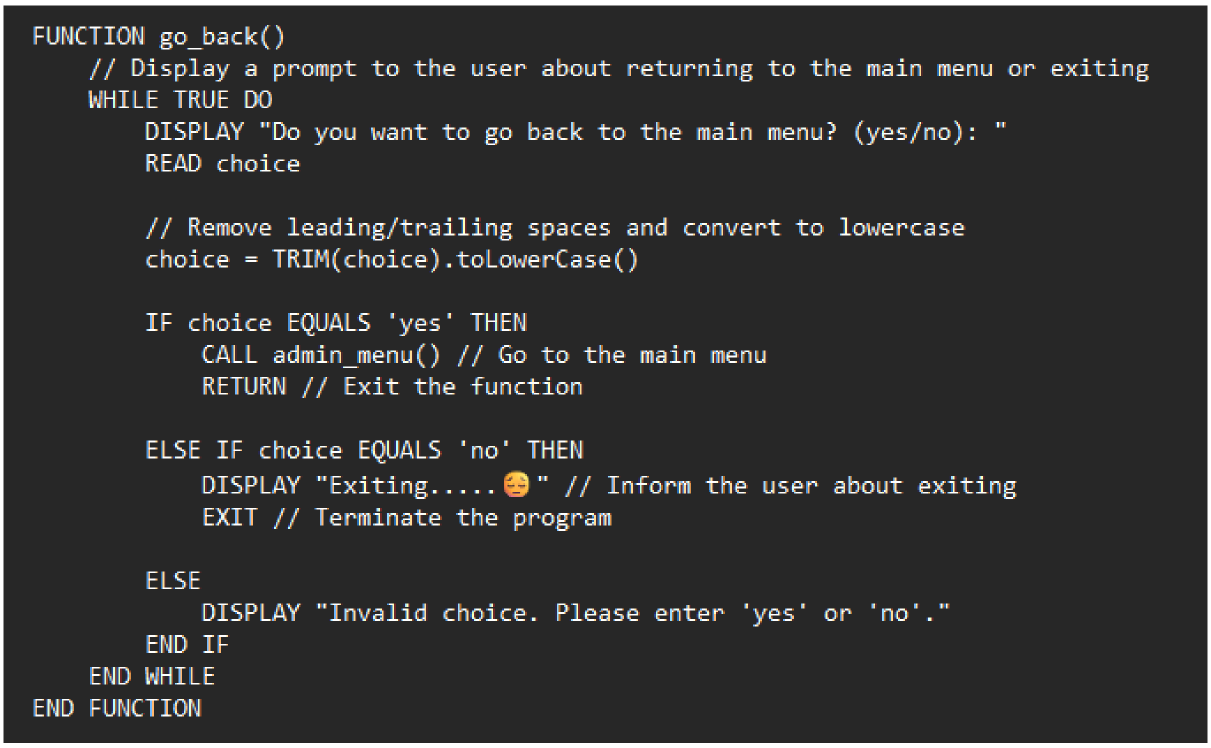
```
  # Once all details are collected, save the user details
  IF user_type == "member":
    SET user_id TO generate_unique_member_id()  # Generate a unique member ID
  ELSE:
    SET user_id TO generate_librarian_id()  # Generate a unique librarian ID

  SET firstname, lastname, email, contact_number, password TO ListDetails

  IF user_type == "member":
    OPEN MEMBERS_FILE FOR APPEND AS file:
      WRITE user_id + " | " + firstname + " | " + lastname + " | " + email + " | " + contact_number + " | " + password + "\n"
  ELSE:
    OPEN LIBRARIANS_FILE FOR APPEND AS file:
      WRITE user_id + " | " + firstname + " | " + lastname + " | " + email + " | " + contact_number + " | " + password + "\n"

  PRINT user_type.capitalize() + " signed up successfully with ID: " + user_id + "."
  CALL go_back()

END FUNCTION
```

4.3.1 Add new librarian or new member to existing list



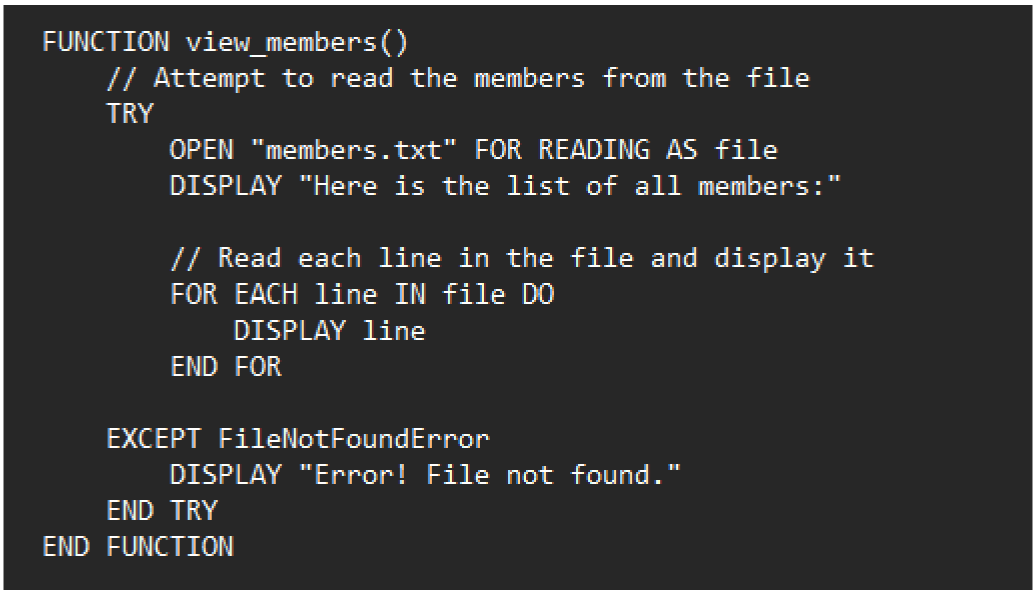
```
CONSTANT MEMBERS_FILE = "members.txt"
CONSTANT LIBRARIANS_FILE = "librarians.txt"
CONSTANT ACTIVITY_LOG_FILE = "activity_log.txt"

FUNCTION add_user(user_type):
   SET counter TO ["Firstname", "Lastname", "Email address", "Contact Number", "Password"]
   SET ListDetails TO ["", "", "", "", ""]
   SET counts TO 0

   WHILE counts < LENGTH(counter):
      WHILE LENGTH(ListDetails[counts]) == 0:  # Presence check
         ListDetails[counts] = input("Enter " + counter[counts] + ": ")
         IF LENGTH(ListDetails[counts]) == 0:
            PRINT "It is mandatory to fill up this field."

      # Length check for contact number
      IF counter[counts] == "Contact Number":
         SET ContactNumber TO ValidateContactNumber(ListDetails[3])
         WHILE ContactNumber == FALSE:
            PRINT "Invalid phone number! The phone number should be in this format: +60 11 1234 1234"
            ListDetails[counts] = input("Enter your " + counter[counts] + ": ")
            ContactNumber = ValidateContactNumber(ListDetails[3])

      # Checking if password entered is a strong password
      IF counter[counts] == "Password":
         SET StrongUserPassword TO StrongPassword(ListDetails[-1])
         WHILE NOT StrongUserPassword:
            PRINT "Password is not strong enough! Your password should be at least 8 characters long and must
contain the following:"
            SET PasswordCredentials TO [
               "1. At least 1 UpperCase letter; W,S,D,R",
               "2. At least 1 special symbol; @,!,&,*",
               "3. At least 1 digit; 0,1,2,3",
               "4. At least 1 lowercase letter; w,s,d,r"
            ]
            FOR requirement IN PasswordCredentials:
               PRINT requirement
            ListDetails[-1] = input("Enter password again: ")
            StrongUserPassword = StrongPassword(ListDetails[-1])

      # Checking if the email already exists
      IF counter[counts] == "Email address":
         SET file_to_check TO MEMBERS_FILE IF user_type == "member" ELSE LIBRARIANS_FILE
         SET exists TO check_existing_user(ListDetails[2], file_to_check)
         IF exists:
            PRINT "This account already exists. Try logging in!"
            RETURN  # Stop the sign-up process if email already exists

      counts += 1


END FUNCTION
```

4.3.2 Go Back Function



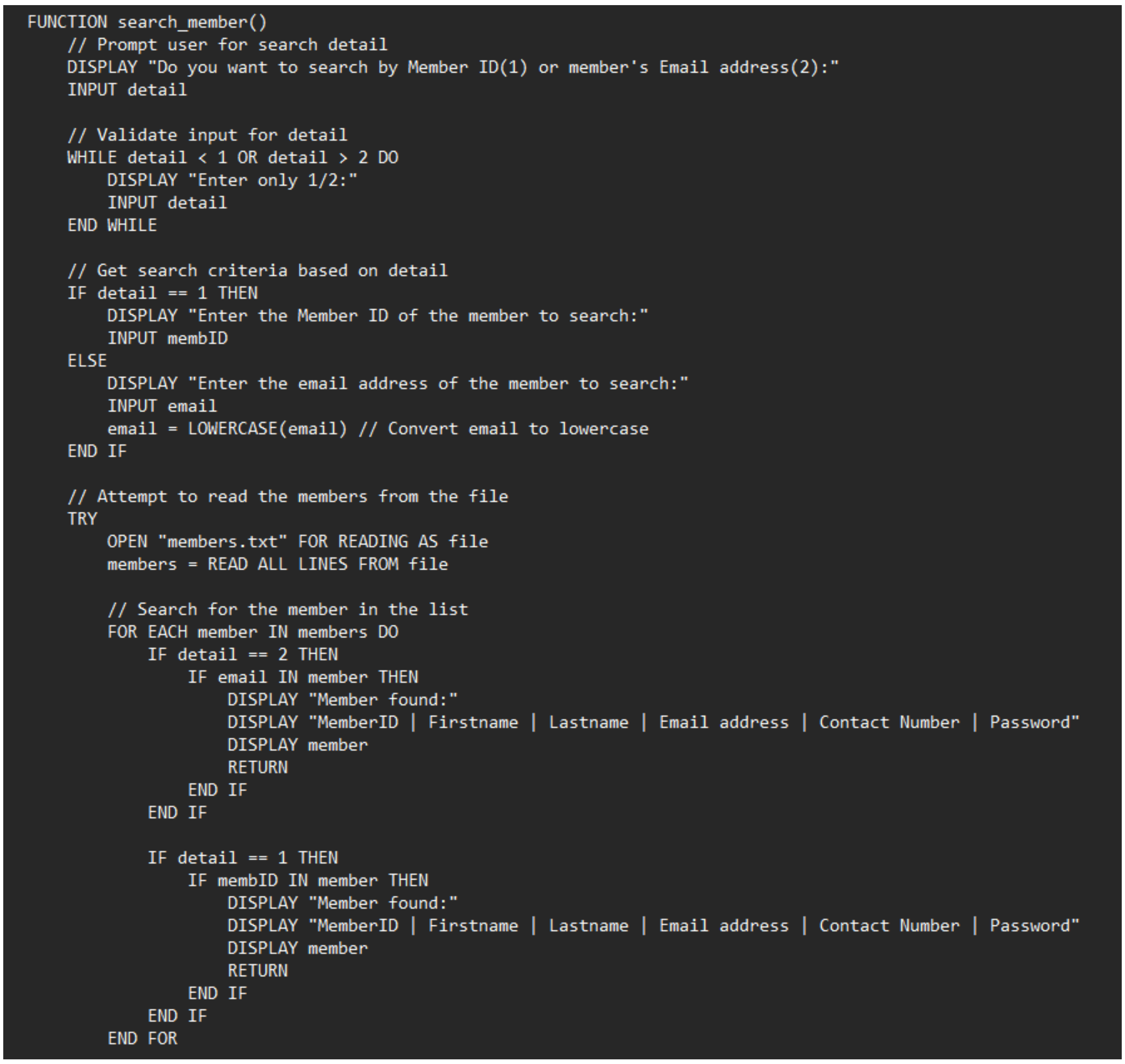
```
FUNCTION go_back()
    // Display a prompt to the user about returning to the main menu or exiting
    WHILE TRUE DO
        DISPLAY "Do you want to go back to the main menu? (yes/no): "
        READ choice

        // Remove leading/trailing spaces and convert to lowercase
        choice = TRIM(choice).toLowerCase()

        IF choice EQUALS 'yes' THEN
            CALL admin_menu() // Go to the main menu
            RETURN // Exit the function

        ELSE IF choice EQUALS 'no' THEN
            DISPLAY "Exiting.....😔" // Inform the user about exiting
            EXIT // Terminate the program

        ELSE
            DISPLAY "Invalid choice. Please enter 'yes' or 'no'."
        END IF
    END WHILE
END FUNCTION
```

4.3.3 Display members' list

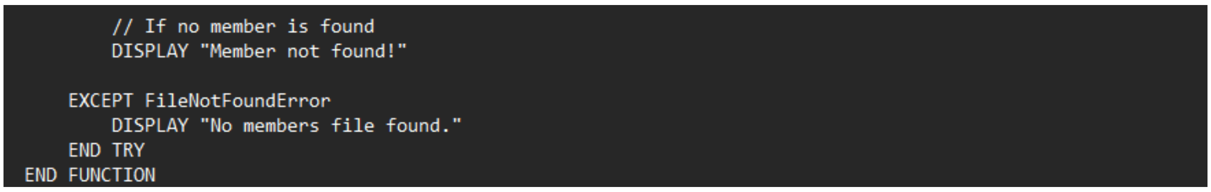


```
FUNCTION view_members()
    // Attempt to read the members from the file
    TRY
        OPEN "members.txt" FOR READING AS file
        DISPLAY "Here is the list of all members:"

        // Read each line in the file and display it
        FOR EACH line IN file DO
            DISPLAY line
        END FOR

    EXCEPT FileNotFoundError
        DISPLAY "Error! File not found."
    END TRY
END FUNCTION
```

4.3.4 Search specific member from existing list

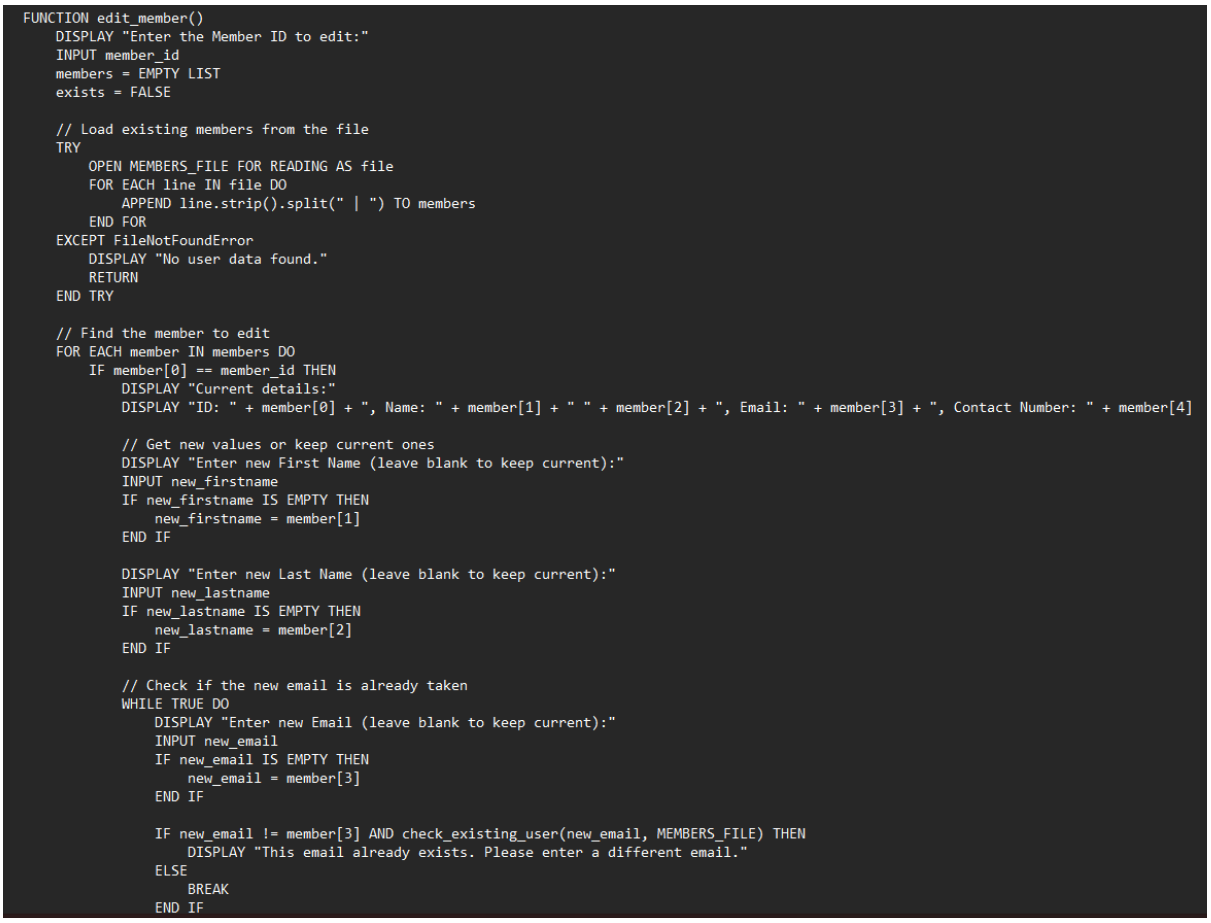


```
FUNCTION search_member()
    // Prompt user for search detail
    DISPLAY "Do you want to search by Member ID(1) or member's Email address(2):"
    INPUT detail

    // Validate input for detail
    WHILE detail < 1 OR detail > 2 DO
        DISPLAY "Enter only 1/2:"
        INPUT detail
    END WHILE

    // Get search criteria based on detail
    IF detail == 1 THEN
        DISPLAY "Enter the Member ID of the member to search:"
        INPUT membID
    ELSE
        DISPLAY "Enter the email address of the member to search:"
        INPUT email
        email = LOWERCASE(email) // Convert email to lowercase
    END IF

    // Attempt to read the members from the file
    TRY
        OPEN "members.txt" FOR READING AS file
        members = READ ALL LINES FROM file

        // Search for the member in the list
        FOR EACH member IN members DO
            IF detail == 2 THEN
                IF email IN member THEN
                    DISPLAY "Member found:"
                    DISPLAY "MemberID | Firstname | Lastname | Email address | Contact Number | Password"
                    DISPLAY member
                    RETURN
                END IF
            END IF

            IF detail == 1 THEN
                IF membID IN member THEN
                    DISPLAY "Member found:"
                    DISPLAY "MemberID | Firstname | Lastname | Email address | Contact Number | Password"
                    DISPLAY member
                    RETURN
                END IF
            END IF
        END FOR
```

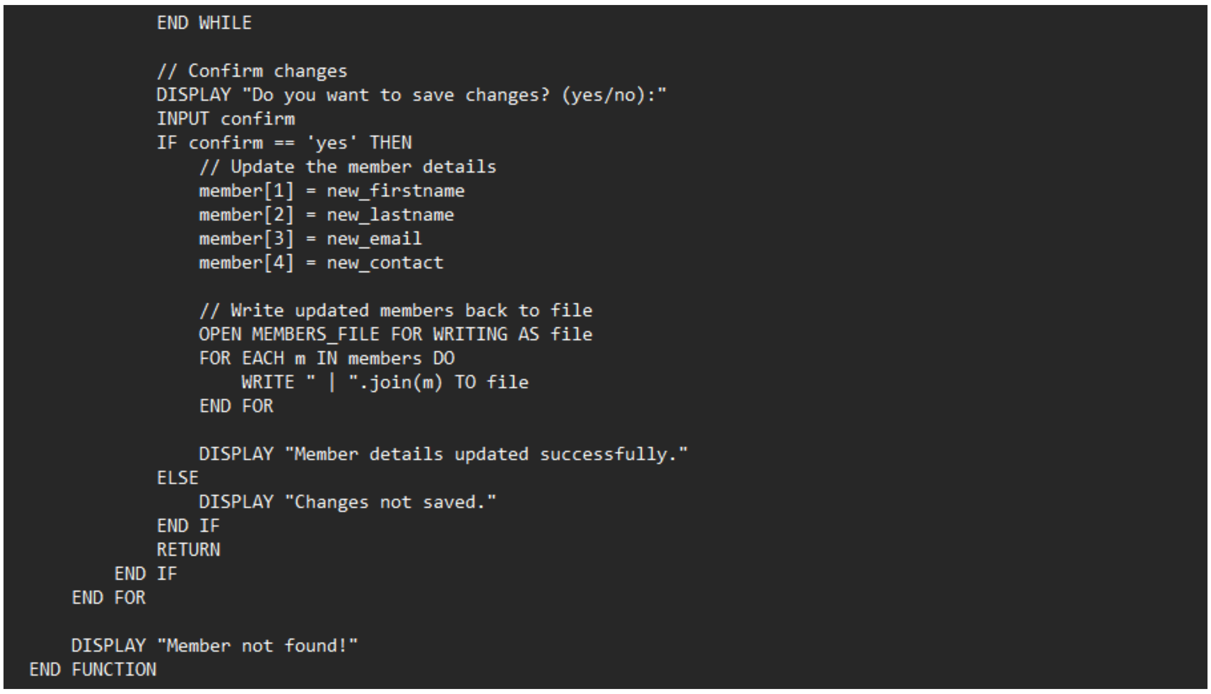


```
        // If no member is found
        DISPLAY "Member not found!"

    EXCEPT FileNotFoundError
        DISPLAY "No members file found."
    END TRY
END FUNCTION
```

4.3.5 Edit specific member’s information from existing list

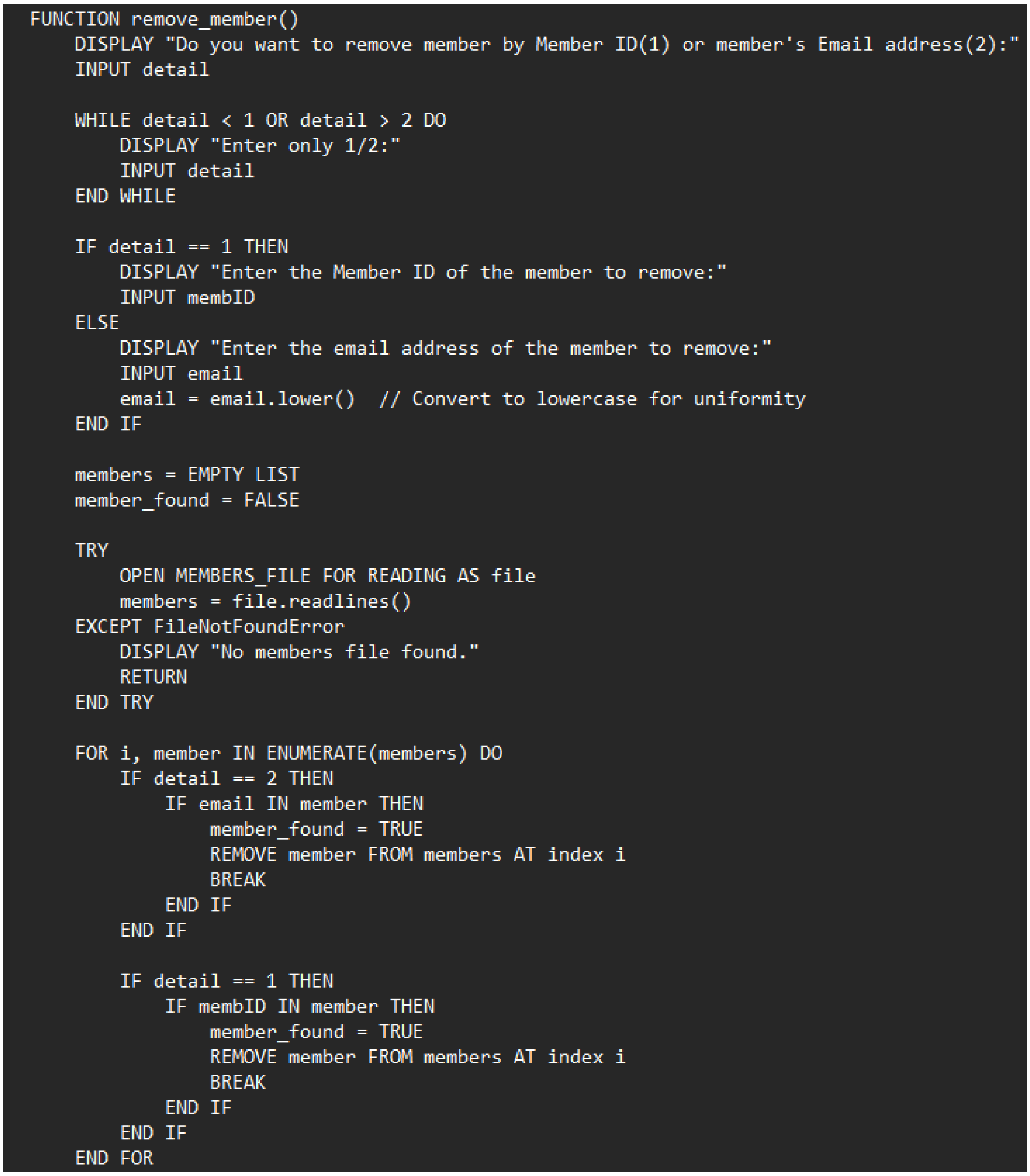


```
FUNCTION edit_member()
    DISPLAY "Enter the Member ID to edit:"
    INPUT member_id
    members = EMPTY LIST
    exists = FALSE

    // Load existing members from the file
    TRY
        OPEN MEMBERS_FILE FOR READING AS file
        FOR EACH line IN file DO
            APPEND line.strip().split(" | ") TO members
        END FOR
    EXCEPT FileNotFoundError
        DISPLAY "No user data found."
        RETURN
    END TRY

    // Find the member to edit
    FOR EACH member IN members DO
        IF member[0] == member_id THEN
            DISPLAY "Current details:"
            DISPLAY "ID: " + member[0] + ", Name: " + member[1] + " " + member[2] + ", Email: " + member[3] + ", Contact Number: " + member[4]

            // Get new values or keep current ones
            DISPLAY "Enter new First Name (leave blank to keep current):"
            INPUT new_firstname
            IF new_firstname IS EMPTY THEN
                new_firstname = member[1]
            END IF

            DISPLAY "Enter new Last Name (leave blank to keep current):"
            INPUT new_lastname
            IF new_lastname IS EMPTY THEN
                new_lastname = member[2]
            END IF

            // Check if the new email is already taken
            WHILE TRUE DO
                DISPLAY "Enter new Email (leave blank to keep current):"
                INPUT new_email
                IF new_email IS EMPTY THEN
                    new_email = member[3]
                END IF

                IF new_email != member[3] AND check_existing_user(new_email, MEMBERS_FILE) THEN
                    DISPLAY "This email already exists. Please enter a different email."
                ELSE
                    BREAK
                END IF
```

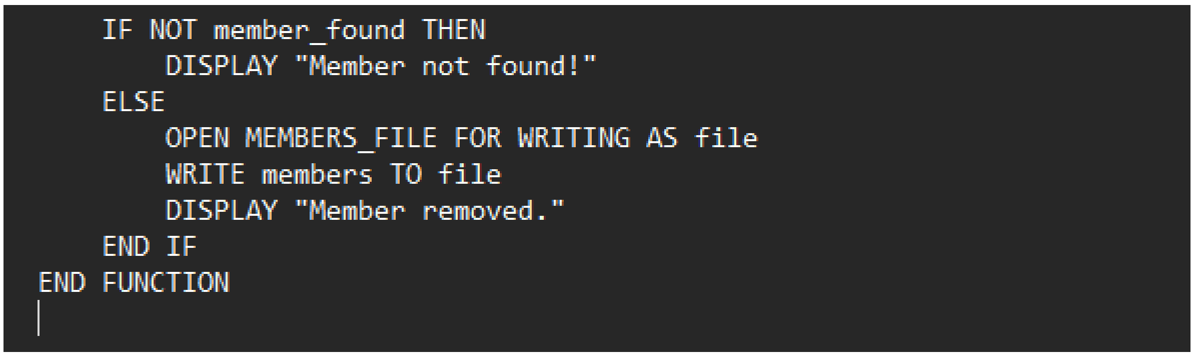


```
            END WHILE

            // Confirm changes
            DISPLAY "Do you want to save changes? (yes/no):"
            INPUT confirm
            IF confirm == 'yes' THEN
                // Update the member details
                member[1] = new_firstname
                member[2] = new_lastname
                member[3] = new_email
                member[4] = new_contact

                // Write updated members back to file
                OPEN MEMBERS_FILE FOR WRITING AS file
                FOR EACH m IN members DO
                    WRITE " | ".join(m) TO file
                END FOR

                DISPLAY "Member details updated successfully."
            ELSE
                DISPLAY "Changes not saved."
            END IF
            RETURN
        END IF
    END FOR

    DISPLAY "Member not found!"
END FUNCTION
```

4.3.6 Remove specific member from existing list

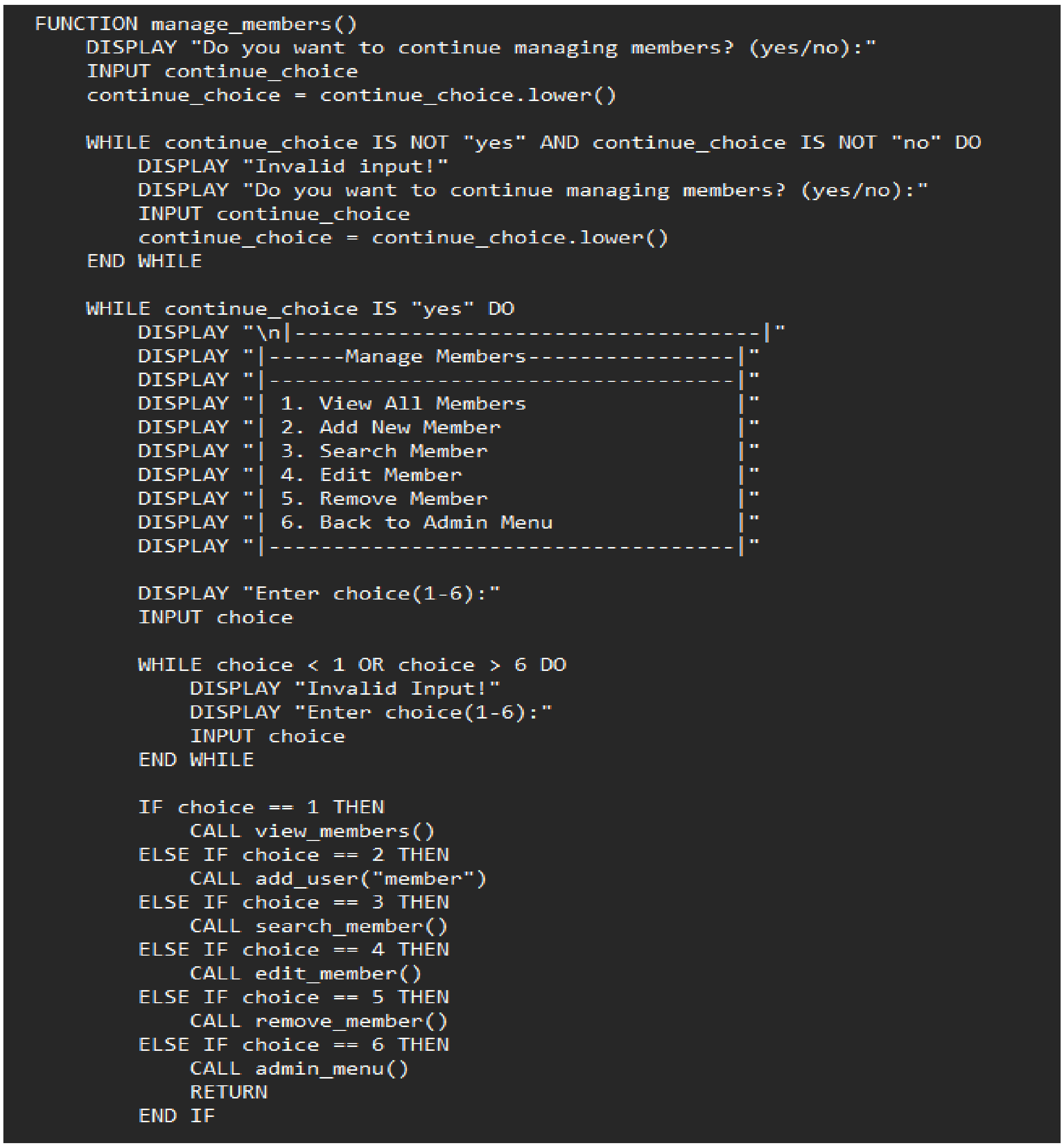


```
FUNCTION remove_member()
    DISPLAY "Do you want to remove member by Member ID(1) or member's Email address(2):"
    INPUT detail

    WHILE detail < 1 OR detail > 2 DO
        DISPLAY "Enter only 1/2:"
        INPUT detail
    END WHILE

    IF detail == 1 THEN
        DISPLAY "Enter the Member ID of the member to remove:"
        INPUT membID
    ELSE
        DISPLAY "Enter the email address of the member to remove:"
        INPUT email
        email = email.lower()  // Convert to lowercase for uniformity
    END IF

    members = EMPTY LIST
    member_found = FALSE

    TRY
        OPEN MEMBERS_FILE FOR READING AS file
        members = file.readlines()
    EXCEPT FileNotFoundError
        DISPLAY "No members file found."
        RETURN
    END TRY

    FOR i, member IN ENUMERATE(members) DO
        IF detail == 2 THEN
            IF email IN member THEN
                member_found = TRUE
                REMOVE member FROM members AT index i
                BREAK
            END IF
        END IF

        IF detail == 1 THEN
            IF membID IN member THEN
                member_found = TRUE
                REMOVE member FROM members AT index i
                BREAK
            END IF
        END IF
    END FOR
```

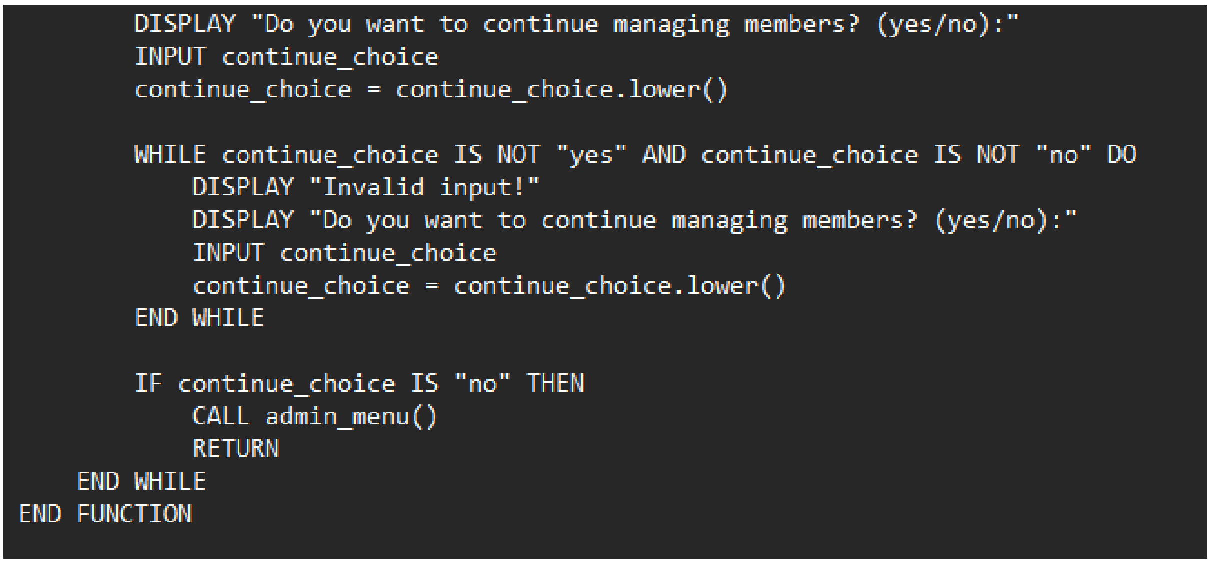


```
    IF NOT member_found THEN
        DISPLAY "Member not found!"
    ELSE
        OPEN MEMBERS_FILE FOR WRITING AS file
        WRITE members TO file
        DISPLAY "Member removed."
    END IF
END FUNCTION
```

4.3.7 Managing members Function (Menu to manage members)



```
FUNCTION manage_members()
    DISPLAY "Do you want to continue managing members? (yes/no):"
    INPUT continue_choice
    continue_choice = continue_choice.lower()

    WHILE continue_choice IS NOT "yes" AND continue_choice IS NOT "no" DO
        DISPLAY "Invalid input!"
        DISPLAY "Do you want to continue managing members? (yes/no):"
        INPUT continue_choice
        continue_choice = continue_choice.lower()
    END WHILE

    WHILE continue_choice IS "yes" DO
        DISPLAY "\n|-----------------------------------|"
        DISPLAY "|------Manage Members---------------|"
        DISPLAY "|-----------------------------------|"
        DISPLAY "| 1. View All Members               |"
        DISPLAY "| 2. Add New Member                 |"
        DISPLAY "| 3. Search Member                  |"
        DISPLAY "| 4. Edit Member                    |"
        DISPLAY "| 5. Remove Member                  |"
        DISPLAY "| 6. Back to Admin Menu             |"
        DISPLAY "|-----------------------------------|"

        DISPLAY "Enter choice(1-6):"
        INPUT choice

        WHILE choice < 1 OR choice > 6 DO
            DISPLAY "Invalid Input!"
            DISPLAY "Enter choice(1-6):"
            INPUT choice
        END WHILE

        IF choice == 1 THEN
            CALL view_members()
        ELSE IF choice == 2 THEN
            CALL add_user("member")
        ELSE IF choice == 3 THEN
            CALL search_member()
        ELSE IF choice == 4 THEN
            CALL edit_member()
        ELSE IF choice == 5 THEN
            CALL remove_member()
        ELSE IF choice == 6 THEN
            CALL admin_menu()
            RETURN
        END IF
```



```
        DISPLAY "Do you want to continue managing members? (yes/no):"
        INPUT continue_choice
        continue_choice = continue_choice.lower()

        WHILE continue_choice IS NOT "yes" AND continue_choice IS NOT "no" DO
            DISPLAY "Invalid input!"
            DISPLAY "Do you want to continue managing members? (yes/no):"
            INPUT continue_choice
            continue_choice = continue_choice.lower()
        END WHILE

        IF continue_choice IS "no" THEN
            CALL admin_menu()
            RETURN
    END WHILE
END FUNCTION
```

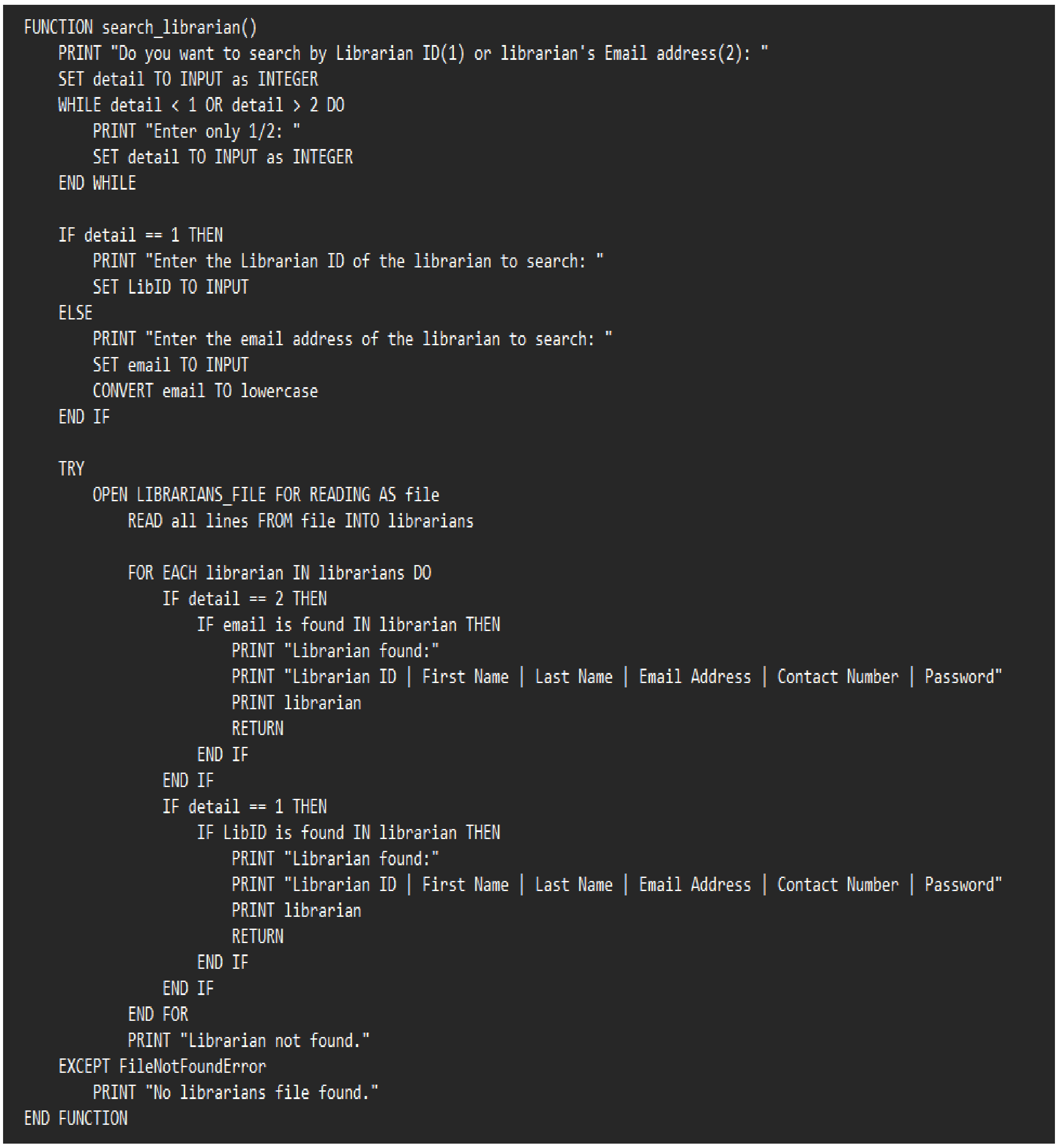
4.3.8 Randomising Librarian ID

```
FUNCTION generate_librarian_id()
    SET librarian_id_prefix = "LIB"
    SET number_of_librarians = LENGTH OF lines IN LIBRARIANS_FILE
    SET unique_librarian_id = librarian_id_prefix + (1000 + number_of_librarians)

    RETURN unique_librarian_id
END FUNCTION
```

4.3.9 Display Librarians' list

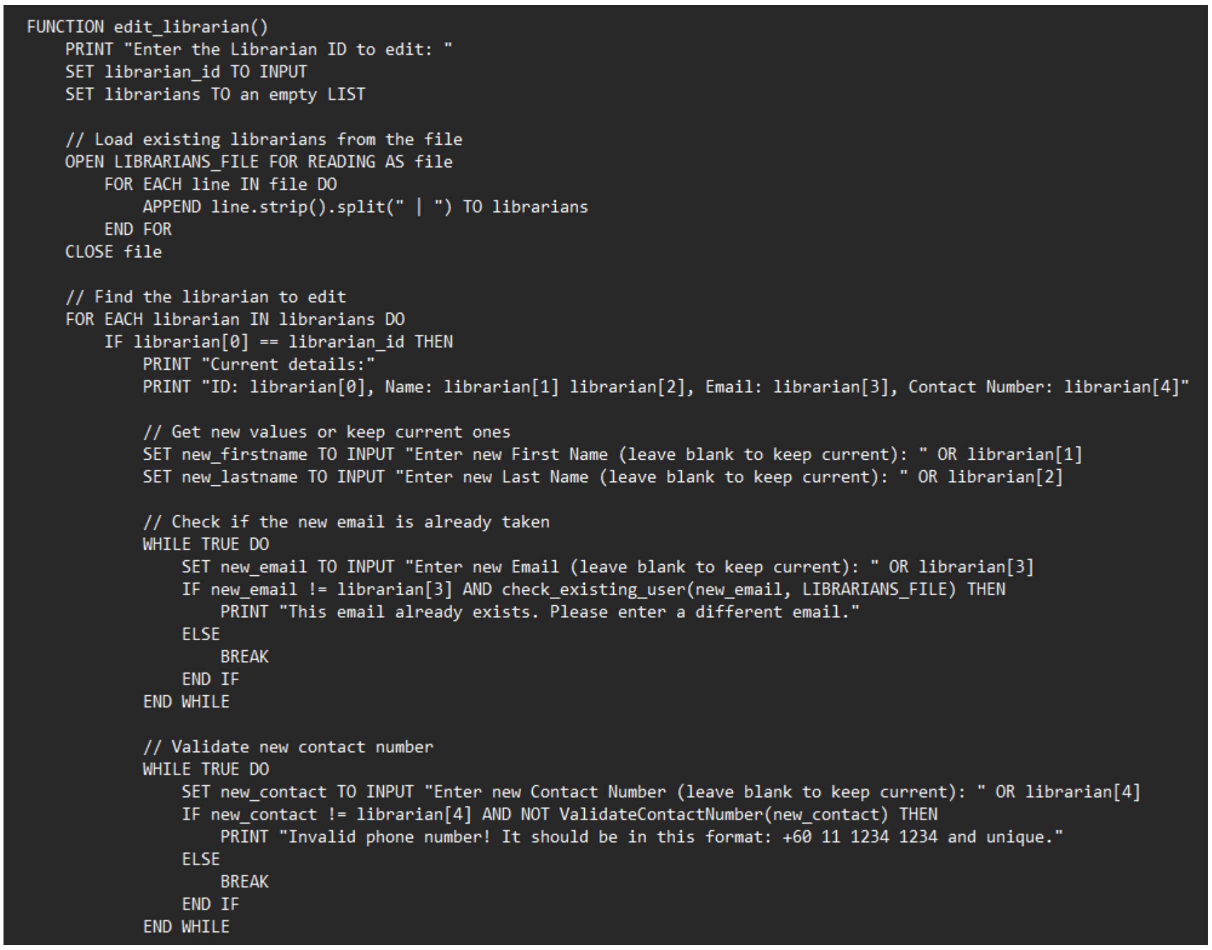
```
FUNCTION view_librarians()
    TRY
        OPEN LIBRARIANS_FILE FOR READING AS file
            READ all lines FROM file INTO librarians

            IF librarians is NOT EMPTY THEN
                PRINT "Librarians List"
                FOR EACH librarian IN librarians DO
                    PRINT librarian
                END FOR
            ELSE
                PRINT "No librarians found."
            END IF
    EXCEPT FileNotFoundError
        PRINT "No librarians file found."
END FUNCTION
```

4.3.10 Search specific librarian's information

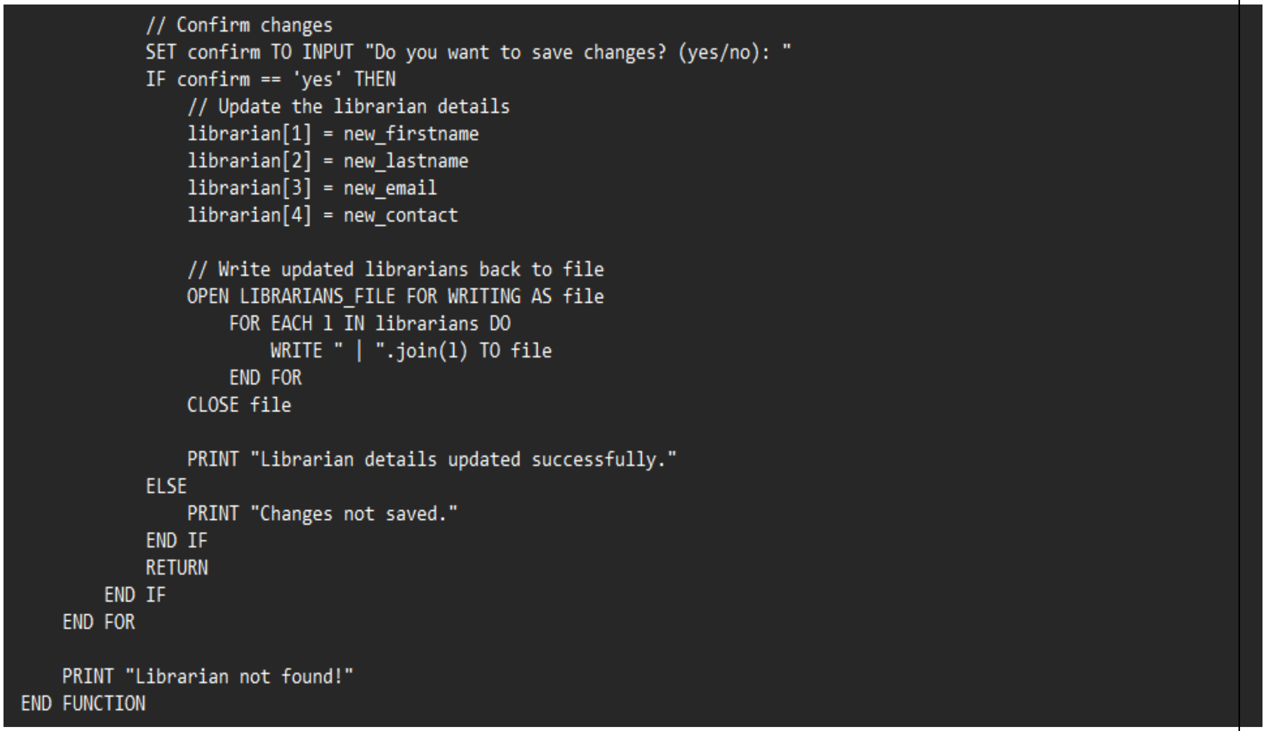


```
FUNCTION search_librarian()
    PRINT "Do you want to search by Librarian ID(1) or librarian's Email address(2): "
    SET detail TO INPUT as INTEGER
    WHILE detail < 1 OR detail > 2 DO
        PRINT "Enter only 1/2: "
        SET detail TO INPUT as INTEGER
    END WHILE

    IF detail == 1 THEN
        PRINT "Enter the Librarian ID of the librarian to search: "
        SET LibID TO INPUT
    ELSE
        PRINT "Enter the email address of the librarian to search: "
        SET email TO INPUT
        CONVERT email TO lowercase
    END IF

    TRY
        OPEN LIBRARIANS_FILE FOR READING AS file
            READ all lines FROM file INTO librarians

            FOR EACH librarian IN librarians DO
                IF detail == 2 THEN
                    IF email is found IN librarian THEN
                        PRINT "Librarian found:"
                        PRINT "Librarian ID | First Name | Last Name | Email Address | Contact Number | Password"
                        PRINT librarian
                        RETURN
                    END IF
                END IF
                IF detail == 1 THEN
                    IF LibID is found IN librarian THEN
                        PRINT "Librarian found:"
                        PRINT "Librarian ID | First Name | Last Name | Email Address | Contact Number | Password"
                        PRINT librarian
                        RETURN
                    END IF
                END IF
            END FOR
            PRINT "Librarian not found."
    EXCEPT FileNotFoundError
        PRINT "No librarians file found."
END FUNCTION
```

4.3.11 Edit specific librarian’s information

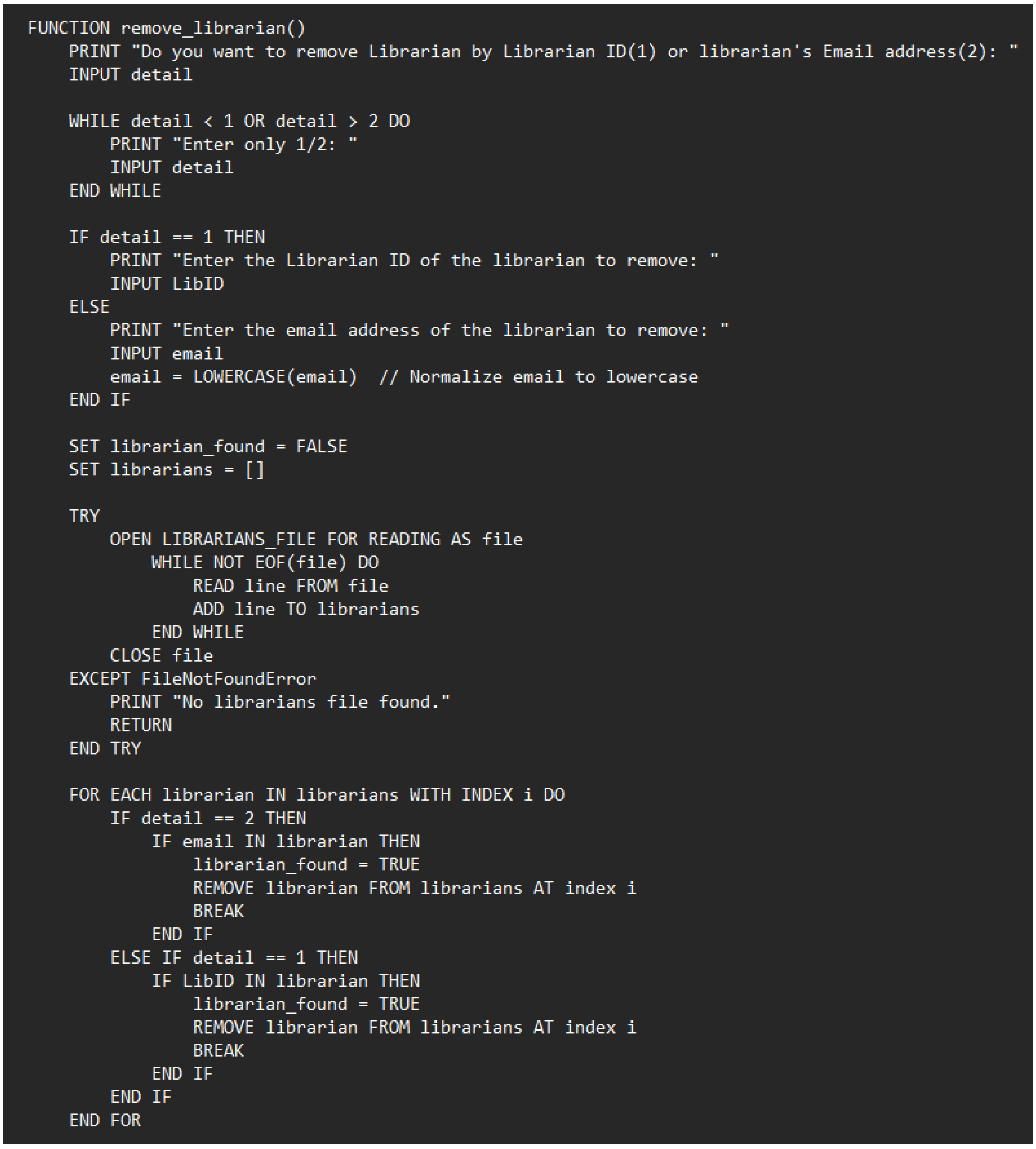


```
FUNCTION edit_librarian()
    PRINT "Enter the Librarian ID to edit: "
    SET librarian_id TO INPUT
    SET librarians TO an empty LIST

    // Load existing librarians from the file
    OPEN LIBRARIANS_FILE FOR READING AS file
        FOR EACH line IN file DO
            APPEND line.strip().split(" | ") TO librarians
        END FOR
    CLOSE file

    // Find the librarian to edit
    FOR EACH librarian IN librarians DO
        IF librarian[0] == librarian_id THEN
            PRINT "Current details:"
            PRINT "ID: librarian[0], Name: librarian[1] librarian[2], Email: librarian[3], Contact Number: librarian[4]"

            // Get new values or keep current ones
            SET new_firstname TO INPUT "Enter new First Name (leave blank to keep current): " OR librarian[1]
            SET new_lastname TO INPUT "Enter new Last Name (leave blank to keep current): " OR librarian[2]

            // Check if the new email is already taken
            WHILE TRUE DO
                SET new_email TO INPUT "Enter new Email (leave blank to keep current): " OR librarian[3]
                IF new_email != librarian[3] AND check_existing_user(new_email, LIBRARIANS_FILE) THEN
                    PRINT "This email already exists. Please enter a different email."
                ELSE
                    BREAK
                END IF
            END WHILE

            // Validate new contact number
            WHILE TRUE DO
                SET new_contact TO INPUT "Enter new Contact Number (leave blank to keep current): " OR librarian[4]
                IF new_contact != librarian[4] AND NOT ValidateContactNumber(new_contact) THEN
                    PRINT "Invalid phone number! It should be in this format: +60 11 1234 1234 and unique."
                ELSE
                    BREAK
                END IF
            END WHILE
```

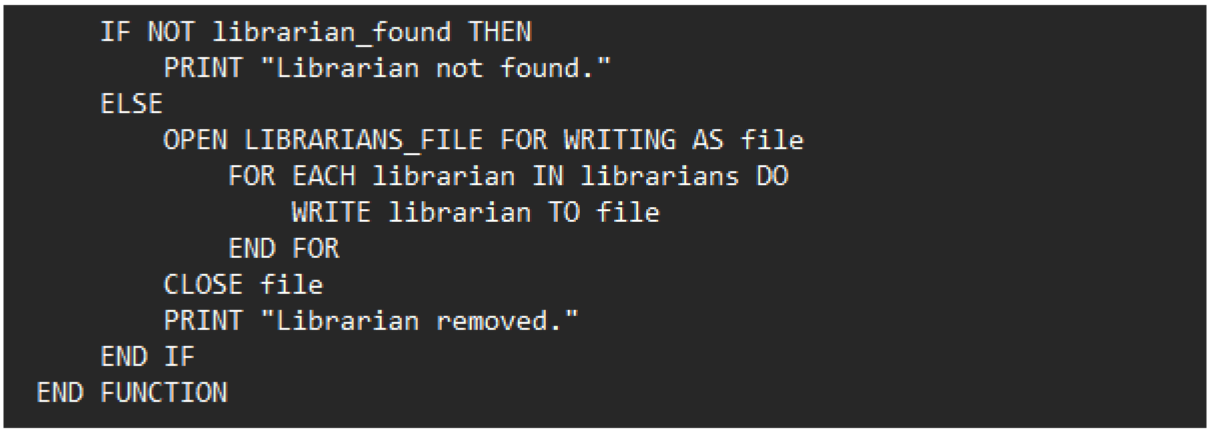


```
            // Confirm changes
            SET confirm TO INPUT "Do you want to save changes? (yes/no): "
            IF confirm == 'yes' THEN
                // Update the librarian details
                librarian[1] = new_firstname
                librarian[2] = new_lastname
                librarian[3] = new_email
                librarian[4] = new_contact

                // Write updated librarians back to file
                OPEN LIBRARIANS_FILE FOR WRITING AS file
                    FOR EACH l IN librarians DO
                        WRITE " | ".join(l) TO file
                    END FOR
                CLOSE file

                PRINT "Librarian details updated successfully."
            ELSE
                PRINT "Changes not saved."
            END IF
            RETURN
        END IF
    END FOR

    PRINT "Librarian not found!"
END FUNCTION
```

4.3.12 Remove specific librarian from existing list

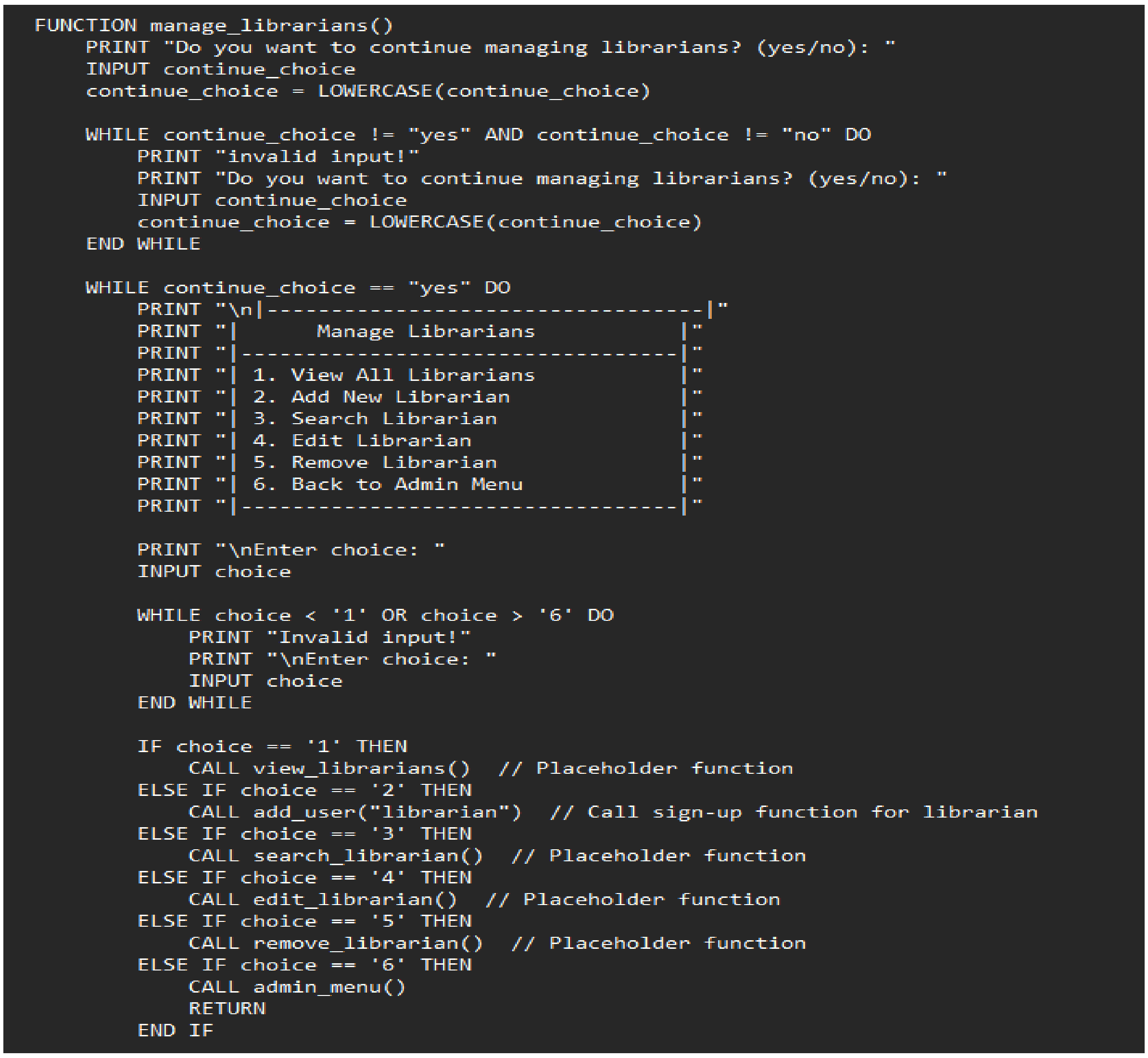


```
FUNCTION remove_librarian()
    PRINT "Do you want to remove Librarian by Librarian ID(1) or librarian's Email address(2): "
    INPUT detail

    WHILE detail < 1 OR detail > 2 DO
        PRINT "Enter only 1/2: "
        INPUT detail
    END WHILE

    IF detail == 1 THEN
        PRINT "Enter the Librarian ID of the librarian to remove: "
        INPUT LibID
    ELSE
        PRINT "Enter the email address of the librarian to remove: "
        INPUT email
        email = LOWERCASE(email)  // Normalize email to lowercase
    END IF

    SET librarian_found = FALSE
    SET librarians = []

    TRY
        OPEN LIBRARIANS_FILE FOR READING AS file
            WHILE NOT EOF(file) DO
                READ line FROM file
                ADD line TO librarians
            END WHILE
        CLOSE file
    EXCEPT FileNotFoundError
        PRINT "No librarians file found."
        RETURN
    END TRY

    FOR EACH librarian IN librarians WITH INDEX i DO
        IF detail == 2 THEN
            IF email IN librarian THEN
                librarian_found = TRUE
                REMOVE librarian FROM librarians AT index i
                BREAK
            END IF
        ELSE IF detail == 1 THEN
            IF LibID IN librarian THEN
                librarian_found = TRUE
                REMOVE librarian FROM librarians AT index i
                BREAK
            END IF
        END IF
    END FOR
```

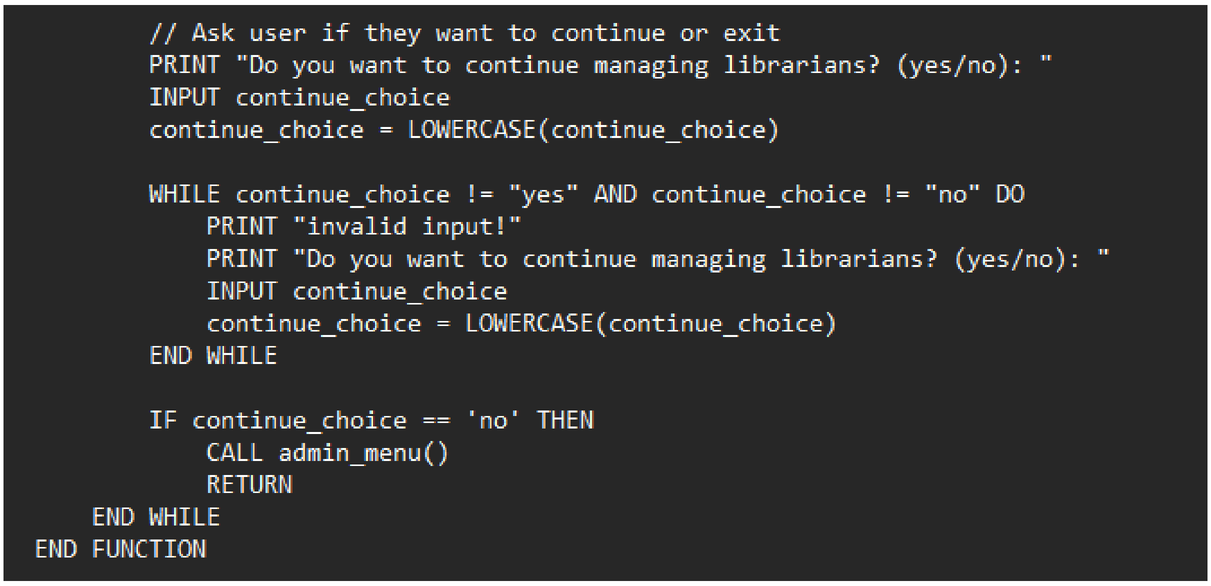


```
    IF NOT librarian_found THEN
        PRINT "Librarian not found."
    ELSE
        OPEN LIBRARIANS_FILE FOR WRITING AS file
            FOR EACH librarian IN librarians DO
                WRITE librarian TO file
            END FOR
        CLOSE file
        PRINT "Librarian removed."
    END IF
END FUNCTION
```

4.3.13 Managing librarians function (Menu to manage librarians)

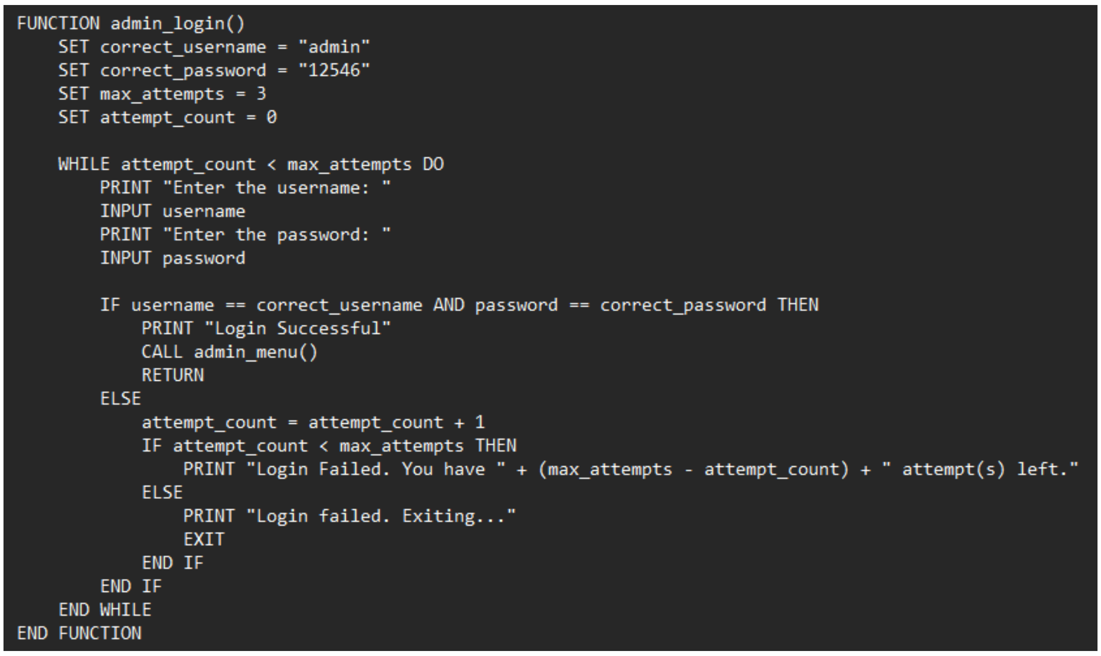


```
FUNCTION manage_librarians()
    PRINT "Do you want to continue managing librarians? (yes/no): "
    INPUT continue_choice
    continue_choice = LOWERCASE(continue_choice)

    WHILE continue_choice != "yes" AND continue_choice != "no" DO
        PRINT "invalid input!"
        PRINT "Do you want to continue managing librarians? (yes/no): "
        INPUT continue_choice
        continue_choice = LOWERCASE(continue_choice)
    END WHILE

    WHILE continue_choice == "yes" DO
        PRINT "\n|----------------------------------|"
        PRINT "|      Manage Librarians           |"
        PRINT "|----------------------------------|"
        PRINT "| 1. View All Librarians           |"
        PRINT "| 2. Add New Librarian             |"
        PRINT "| 3. Search Librarian              |"
        PRINT "| 4. Edit Librarian                |"
        PRINT "| 5. Remove Librarian              |"
        PRINT "| 6. Back to Admin Menu            |"
        PRINT "|----------------------------------|"

        PRINT "\nEnter choice: "
        INPUT choice

        WHILE choice < '1' OR choice > '6' DO
            PRINT "Invalid input!"
            PRINT "\nEnter choice: "
            INPUT choice
        END WHILE

        IF choice == '1' THEN
            CALL view_librarians()  // Placeholder function
        ELSE IF choice == '2' THEN
            CALL add_user("librarian")  // Call sign-up function for librarian
        ELSE IF choice == '3' THEN
            CALL search_librarian()  // Placeholder function
        ELSE IF choice == '4' THEN
            CALL edit_librarian()  // Placeholder function
        ELSE IF choice == '5' THEN
            CALL remove_librarian()  // Placeholder function
        ELSE IF choice == '6' THEN
            CALL admin_menu()
            RETURN
        END IF
```



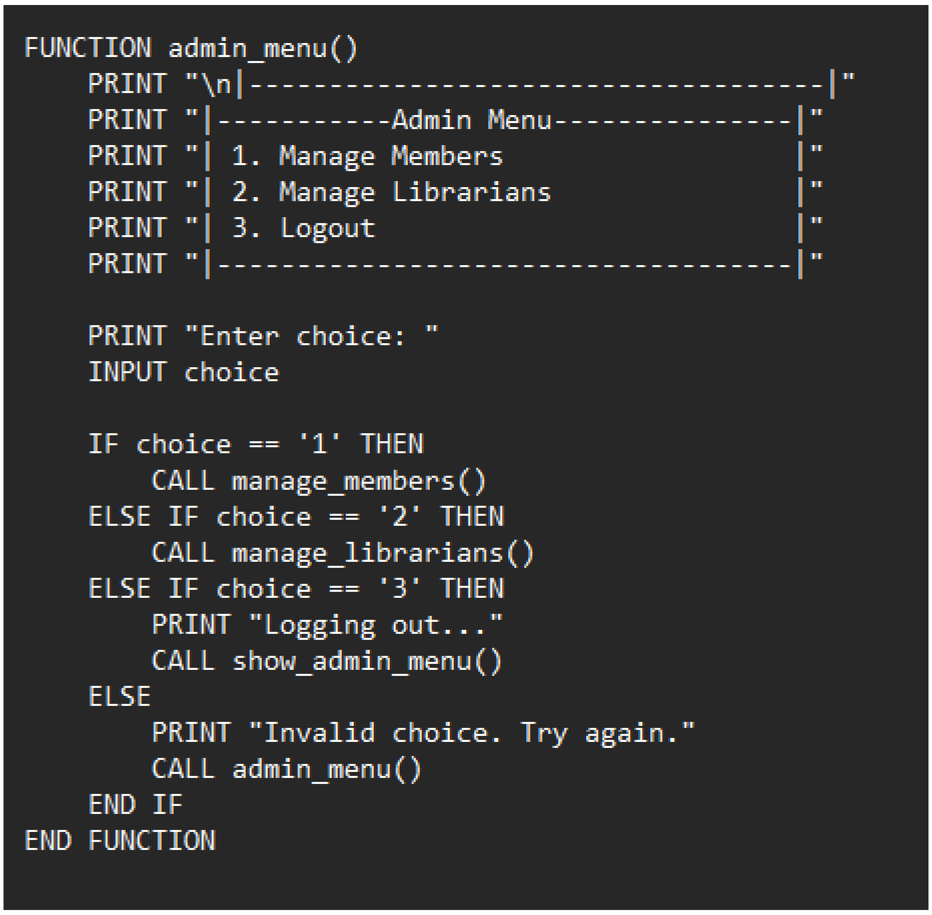
```
            // Ask user if they want to continue or exit
            PRINT "Do you want to continue managing librarians? (yes/no): "
            INPUT continue_choice
            continue_choice = LOWERCASE(continue_choice)

            WHILE continue_choice != "yes" AND continue_choice != "no" DO
                PRINT "invalid input!"
                PRINT "Do you want to continue managing librarians? (yes/no): "
                INPUT continue_choice
                continue_choice = LOWERCASE(continue_choice)
            END WHILE

            IF continue_choice == 'no' THEN
                CALL admin_menu()
                RETURN
        END WHILE
    END FUNCTION
```

4.3.14 Admin login



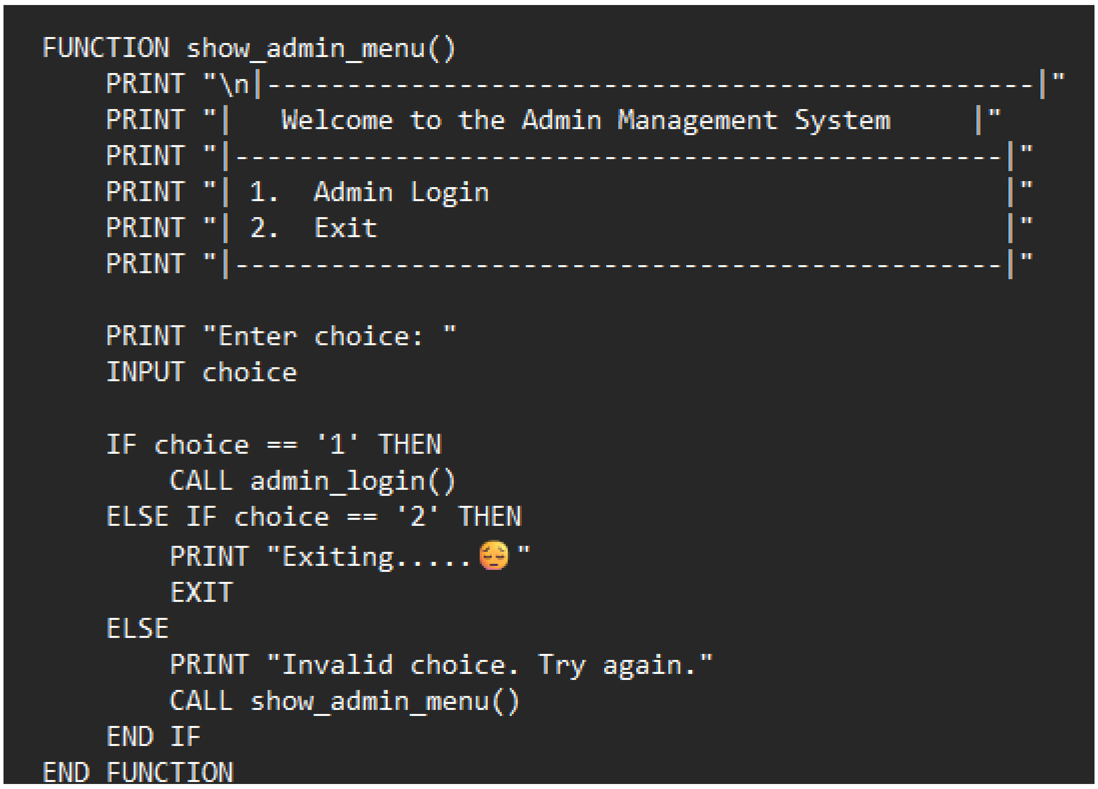
```
FUNCTION admin_login()
    SET correct_username = "admin"
    SET correct_password = "12546"
    SET max_attempts = 3
    SET attempt_count = 0

    WHILE attempt_count < max_attempts DO
        PRINT "Enter the username: "
        INPUT username
        PRINT "Enter the password: "
        INPUT password

        IF username == correct_username AND password == correct_password THEN
            PRINT "Login Successful"
            CALL admin_menu()
            RETURN
        ELSE
            attempt_count = attempt_count + 1
            IF attempt_count < max_attempts THEN
                PRINT "Login Failed. You have " + (max_attempts - attempt_count) + " attempt(s) left."
            ELSE
                PRINT "Login failed. Exiting..."
                EXIT
            END IF
        END IF
    END WHILE
END FUNCTION
```

4.3.15 Admin Menu



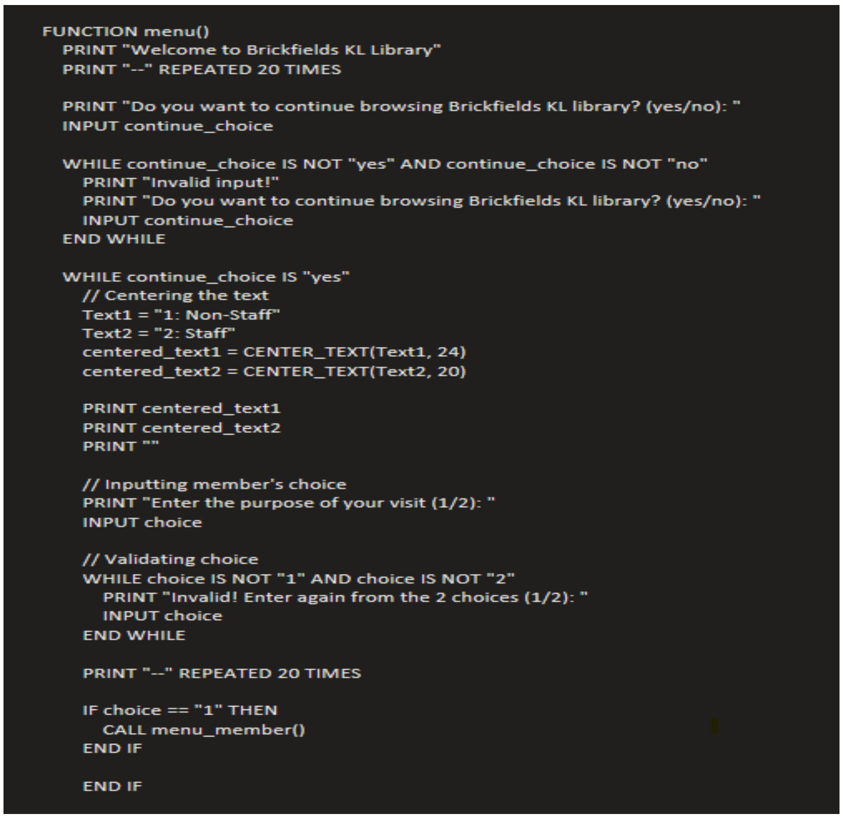
```
FUNCTION admin_menu()
    PRINT "\n|------------------------------------|"
    PRINT "|-----------Admin Menu---------------|"
    PRINT "| 1. Manage Members                  |"
    PRINT "| 2. Manage Librarians               |"
    PRINT "| 3. Logout                          |"
    PRINT "|------------------------------------|"

    PRINT "Enter choice: "
    INPUT choice

    IF choice == '1' THEN
        CALL manage_members()
    ELSE IF choice == '2' THEN
        CALL manage_librarians()
    ELSE IF choice == '3' THEN
        PRINT "Logging out..."
        CALL show_admin_menu()
    ELSE
        PRINT "Invalid choice. Try again."
        CALL admin_menu()
    END IF
END FUNCTION
```

4.3.16 Show admin menu function

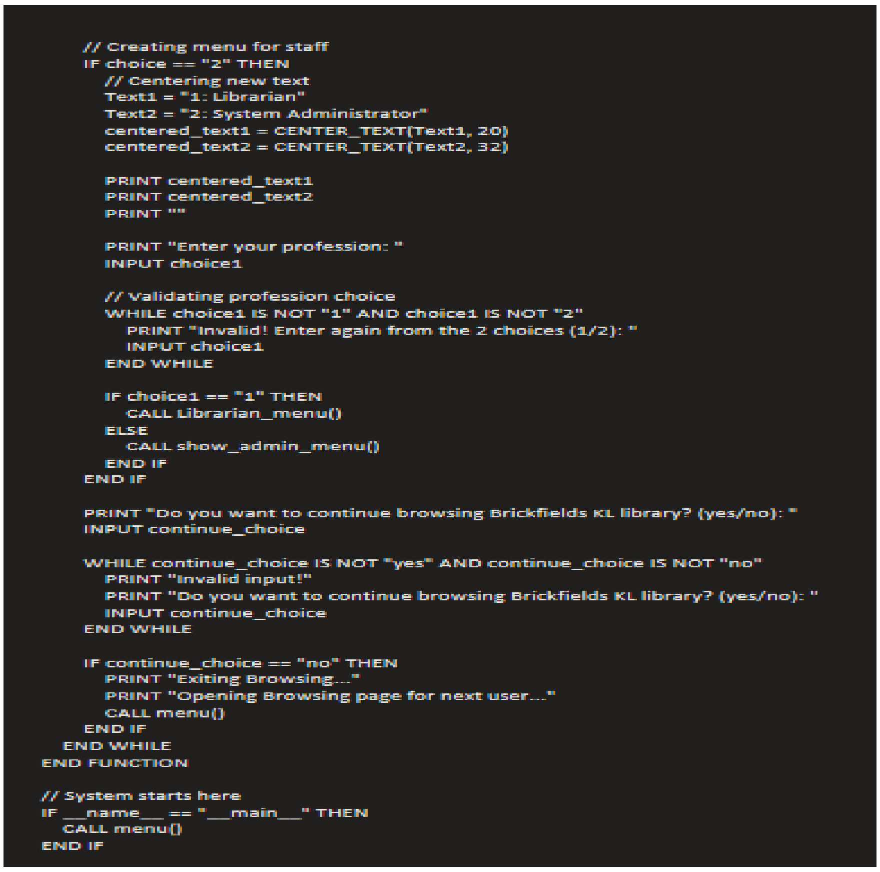


```
FUNCTION show_admin_menu()
    PRINT "\n|------------------------------------------------|"
    PRINT "|   Welcome to the Admin Management System     |"
    PRINT "|------------------------------------------------|"
    PRINT "| 1.  Admin Login                                |"
    PRINT "| 2.  Exit                                       |"
    PRINT "|------------------------------------------------|"

    PRINT "Enter choice: "
    INPUT choice

    IF choice == '1' THEN
        CALL admin_login()
    ELSE IF choice == '2' THEN
        PRINT "Exiting.....😔"
        EXIT
    ELSE
        PRINT "Invalid choice. Try again."
        CALL show_admin_menu()
    END IF
END FUNCTION
```

4.4 Main Menu for Brickfields Kuala Lumpur Community Library



```
FUNCTION menu()
   PRINT "Welcome to Brickfields KL Library"
   PRINT "--" REPEATED 20 TIMES

   PRINT "Do you want to continue browsing Brickfields KL library? (yes/no): "
   INPUT continue_choice

   WHILE continue_choice IS NOT "yes" AND continue_choice IS NOT "no"
      PRINT "Invalid input!"
      PRINT "Do you want to continue browsing Brickfields KL library? (yes/no): "
      INPUT continue_choice
   END WHILE

   WHILE continue_choice IS "yes"
      // Centering the text
      Text1 = "1: Non-Staff"
      Text2 = "2: Staff"
      centered_text1 = CENTER_TEXT(Text1, 24)
      centered_text2 = CENTER_TEXT(Text2, 20)

      PRINT centered_text1
      PRINT centered_text2
      PRINT ""

      // Inputting member's choice
      PRINT "Enter the purpose of your visit (1/2): "
      INPUT choice

      // Validating choice
      WHILE choice IS NOT "1" AND choice IS NOT "2"
         PRINT "Invalid! Enter again from the 2 choices (1/2): "
         INPUT choice
      END WHILE

      PRINT "--" REPEATED 20 TIMES

      IF choice == "1" THEN
         CALL menu_member()
      END IF

      END IF
```



```
      // Creating menu for staff
      IF choice == "2" THEN
         // Centering new text
         Text1 = "1: Librarian"
         Text2 = "2: System Administrator"
         centered_text1 = CENTER_TEXT(Text1, 20)
         centered_text2 = CENTER_TEXT(Text2, 32)

         PRINT centered_text1
         PRINT centered_text2
         PRINT ""

         PRINT "Enter your profession: "
         INPUT choice1

         // Validating profession choice
         WHILE choice1 IS NOT "1" AND choice1 IS NOT "2"
            PRINT "Invalid! Enter again from the 2 choices (1/2): "
            INPUT choice1
         END WHILE

         IF choice1 == "1" THEN
            CALL Librarian_menu()
         ELSE
            CALL show_admin_menu()
         END IF
      END IF

      PRINT "Do you want to continue browsing Brickfields KL library? (yes/no): "
      INPUT continue_choice

      WHILE continue_choice IS NOT "yes" AND continue_choice IS NOT "no"
         PRINT "Invalid input!"
         PRINT "Do you want to continue browsing Brickfields KL library? (yes/no): "
         INPUT continue_choice
      END WHILE

      IF continue_choice == "no" THEN
         PRINT "Exiting Browsing..."
         PRINT "Opening Browsing page for next user..."
         CALL menu()
      END IF
   END WHILE
END FUNCTION

// System starts here
IF __name__ == "__main__" THEN
   CALL menu()
END IF
```

5.0 Programs' Source Code

5.1 Implementation

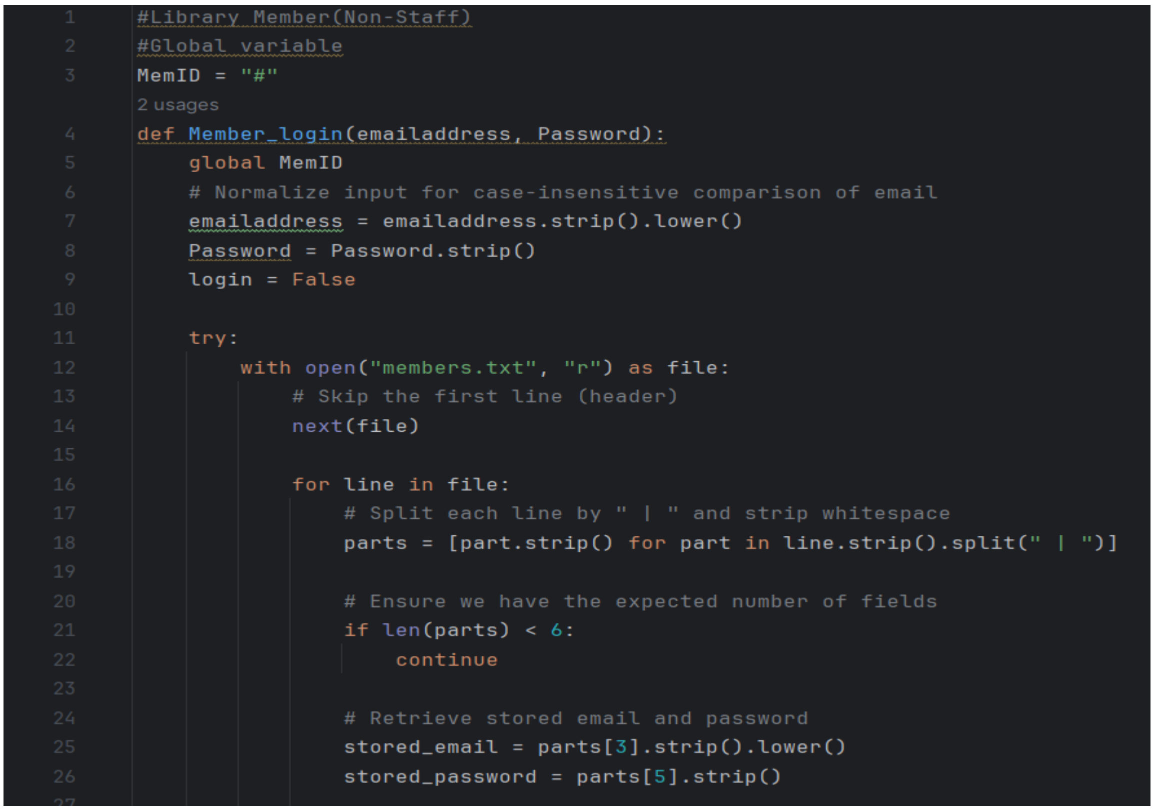
5.1.1 Member User

5.1.1.1 Member Login

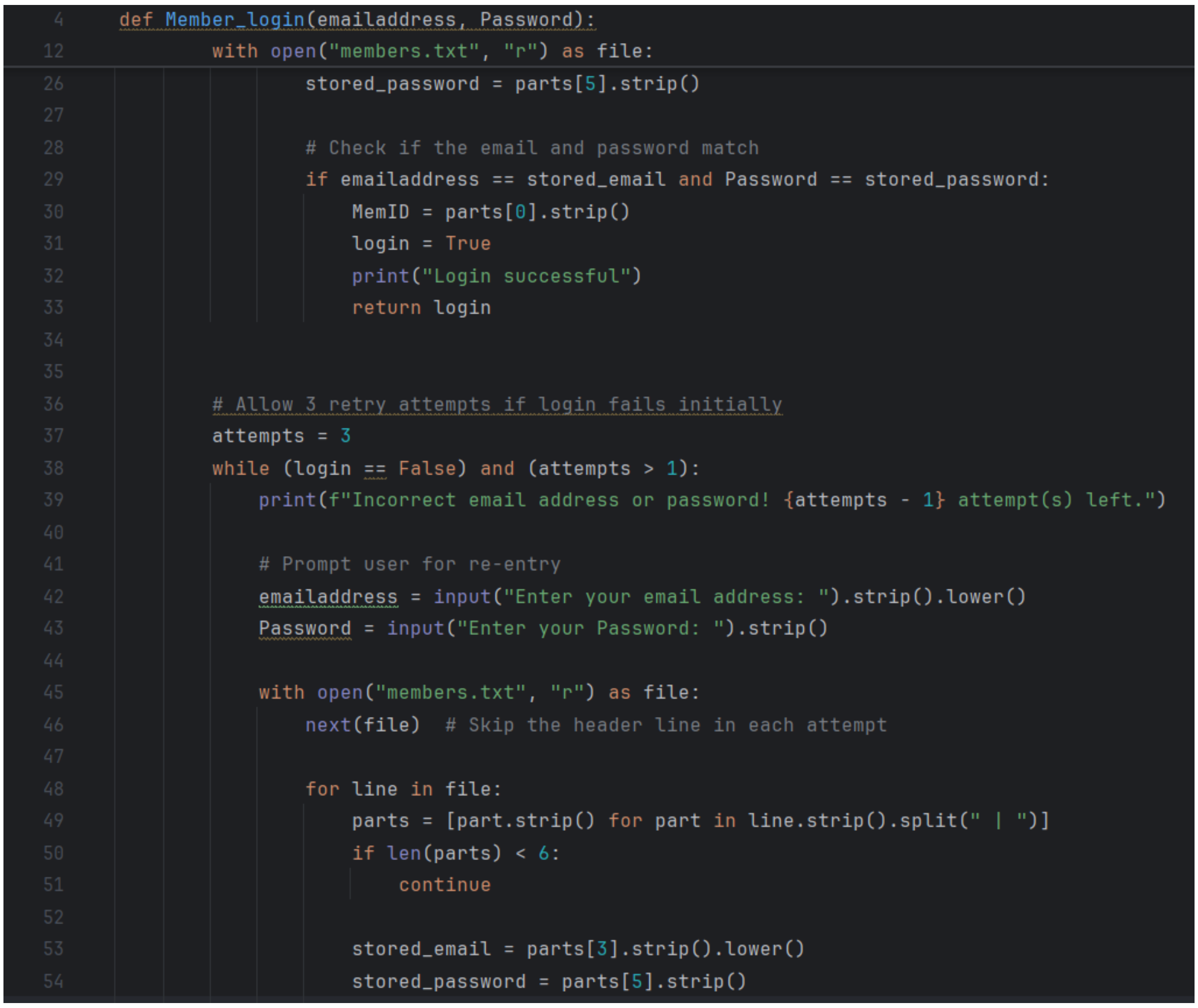
MemID is declared as a global variable as it will be used in other functions. Once a member has logged in or signed up, the system will assign its respective ID to MemID. Members who want to log in will need to input their email address and password. They will get only three attempts to log in, after these 3 chances, the login will fail, and they will leave the member menu. The user email address and password will be compared to each email address and password stored in the file. If they match, the login will be successful.

The member's ID will also be stored in the global variable, MemID.

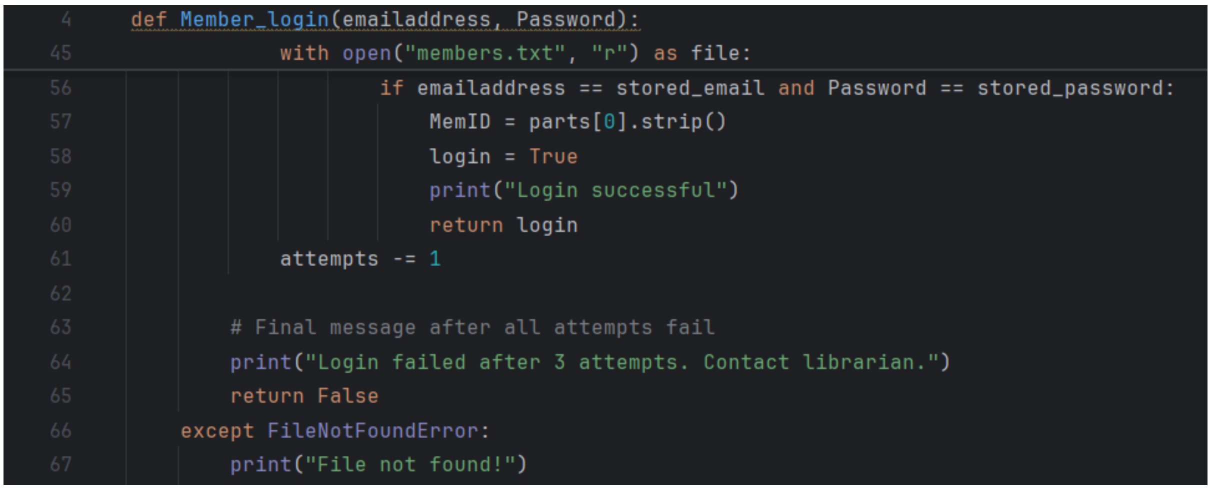
Exception handling is also implemented, if the members.txt file cannot open, it will simply print out an error message.



```python
#Library Member(Non-Staff)
#Global variable
MemID = "#"
2 usages
def Member_login(emailaddress, Password):
    global MemID
    # Normalize input for case-insensitive comparison of email
    emailaddress = emailaddress.strip().lower()
    Password = Password.strip()
    login = False

    try:
        with open("members.txt", "r") as file:
            # Skip the first line (header)
            next(file)

            for line in file:
                # Split each line by " | " and strip whitespace
                parts = [part.strip() for part in line.strip().split(" | ")]

                # Ensure we have the expected number of fields
                if len(parts) < 6:
                    continue

                # Retrieve stored email and password
                stored_email = parts[3].strip().lower()
                stored_password = parts[5].strip()
```

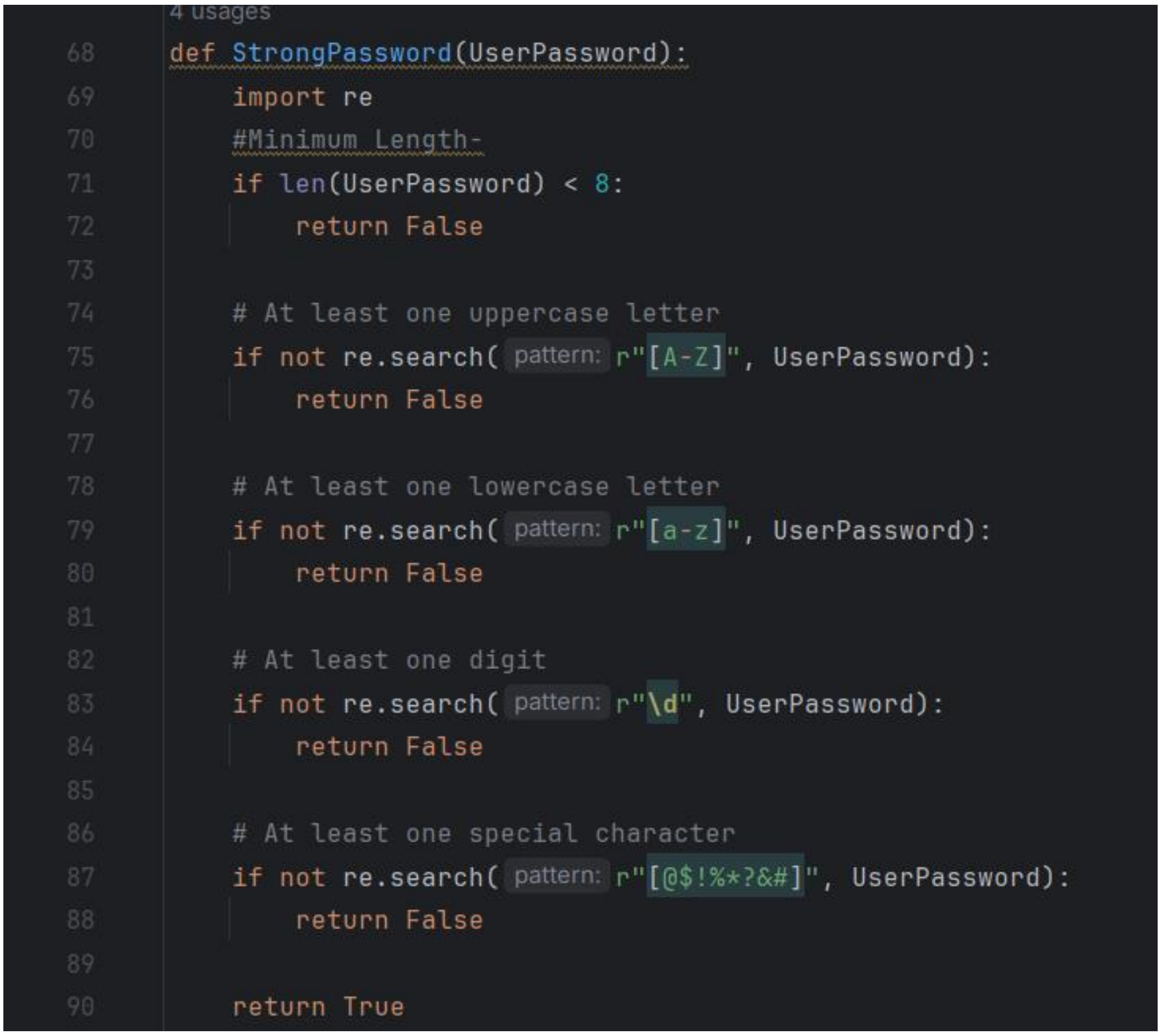


```
def Member_login(emailaddress, Password):
        with open("members.txt", "r") as file:
                stored_password = parts[5].strip()

                # Check if the email and password match
                if emailaddress == stored_email and Password == stored_password:
                    MemID = parts[0].strip()
                    login = True
                    print("Login successful")
                    return login


        # Allow 3 retry attempts if login fails initially
        attempts = 3
        while (login == False) and (attempts > 1):
            print(f"Incorrect email address or password! {attempts - 1} attempt(s) left.")

            # Prompt user for re-entry
            emailaddress = input("Enter your email address: ").strip().lower()
            Password = input("Enter your Password: ").strip()

            with open("members.txt", "r") as file:
                next(file)  # Skip the header line in each attempt

                for line in file:
                    parts = [part.strip() for part in line.strip().split(" | ")]
                    if len(parts) < 6:
                        continue

                    stored_email = parts[3].strip().lower()
                    stored_password = parts[5].strip()
```



```
def Member_login(emailaddress, Password):
            with open("members.txt", "r") as file:
                    if emailaddress == stored_email and Password == stored_password:
                        MemID = parts[0].strip()
                        login = True
                        print("Login successful")
                        return login
            attempts -= 1

        # Final message after all attempts fail
        print("Login failed after 3 attempts. Contact librarian.")
        return False
    except FileNotFoundError:
        print("File not found!")
```

5.1.1.2 Password Validation



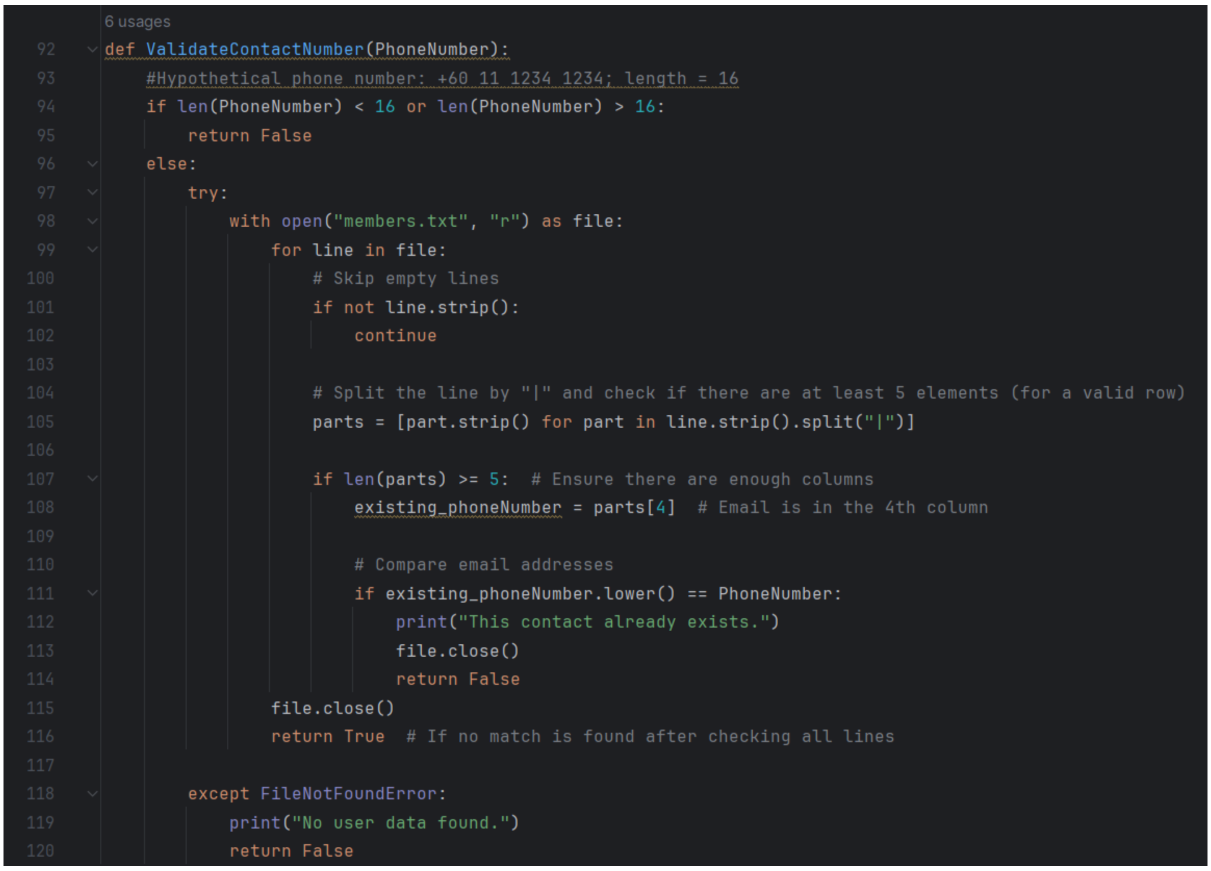
```
4 usages
def StrongPassword(UserPassword):
    import re
    #Minimum Length-
    if len(UserPassword) < 8:
        return False

    # At least one uppercase letter
    if not re.search( pattern: r"[A-Z]", UserPassword):
        return False

    # At least one lowercase letter
    if not re.search( pattern: r"[a-z]", UserPassword):
        return False

    # At least one digit
    if not re.search( pattern: r"\d", UserPassword):
        return False

    # At least one special character
    if not re.search( pattern: r"[@$!%*?&#]", UserPassword):
        return False

    return True
```

The system will import the re module. Python has a built-in package called re, which work with Regular Expressions, that is, a sequence of characters that forms a search pattern. It will check if a string contains the specified search pattern.

In this case it will check if the password entered has characters between, A-Z, a-z, digit numbers, and other special characters like @, #, $, %, and others. It does so to ensure that members have a strong password to log in, keeping their account secure from unwanted breaches.

The credentials for a strong password are at least 1 Uppercase letter, at least 1 special, at least 1 digit and at least 1 lowercase letter.

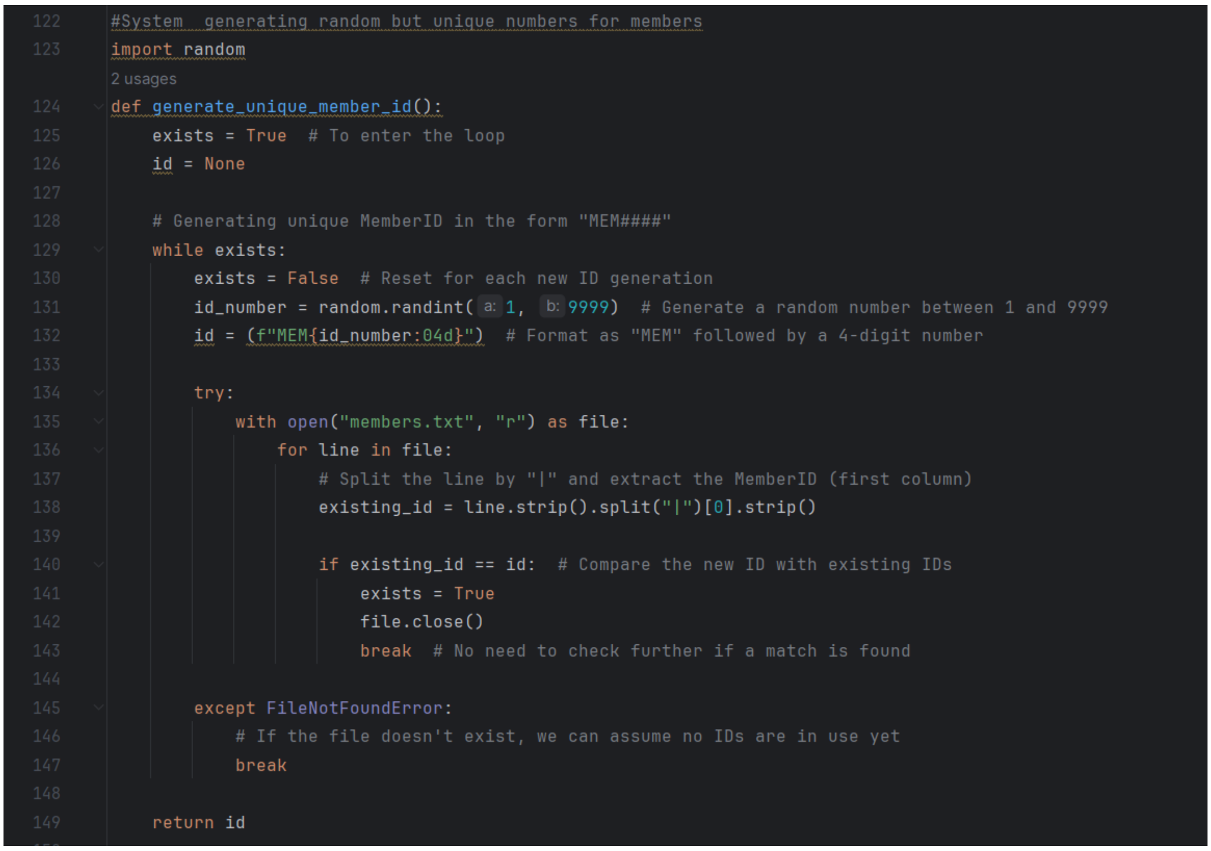
5.1.1.3 Contact Number Validation



```
6 usages
def ValidateContactNumber(PhoneNumber):
    #Hypothetical phone number: +60 11 1234 1234; length = 16
    if len(PhoneNumber) < 16 or len(PhoneNumber) > 16:
        return False
    else:
        try:
            with open("members.txt", "r") as file:
                for line in file:
                    # Skip empty lines
                    if not line.strip():
                        continue

                    # Split the line by "|" and check if there are at least 5 elements (for a valid row)
                    parts = [part.strip() for part in line.strip().split("|")]

                    if len(parts) >= 5:  # Ensure there are enough columns
                        existing_phoneNumber = parts[4]  # Email is in the 4th column

                        # Compare email addresses
                        if existing_phoneNumber.lower() == PhoneNumber:
                            print("This contact already exists.")
                            file.close()
                            return False
                file.close()
                return True  # If no match is found after checking all lines

        except FileNotFoundError:
            print("No user data found.")
            return False
```

This function will ensure that the contact number entered by the user has the same length as the “+60 nn nnnn nnnn” format, where n represents any number. It is assumed that no one has a duplicate number practically, so no need for a uniqueness check. The Length of the contact number, however, should be 16. Any input, exceeding 16 or below will be rejected. This function applies a length check on the contact number.

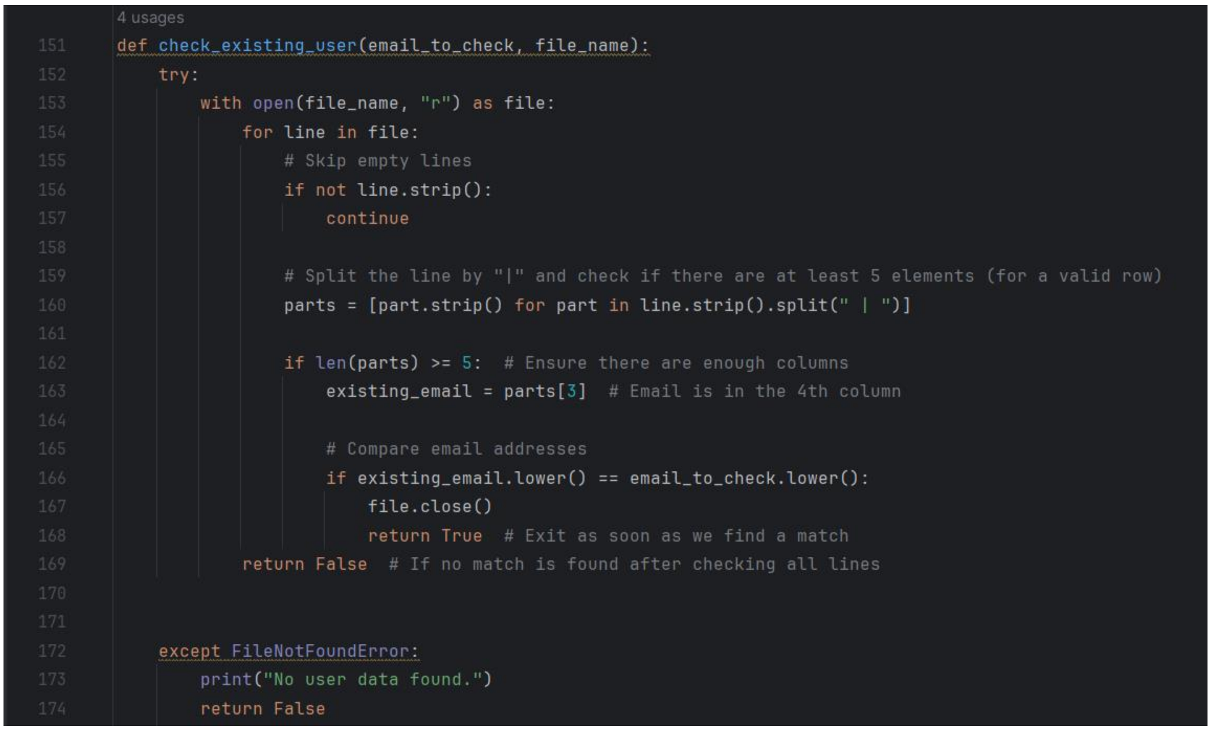
5.1.1.4 Member ID Randomising



```python
#System  generating random but unique numbers for members
import random
2 usages
def generate_unique_member_id():
    exists = True  # To enter the loop
    id = None

    # Generating unique MemberID in the form "MEM####"
    while exists:
        exists = False  # Reset for each new ID generation
        id_number = random.randint( a: 1,  b: 9999)  # Generate a random number between 1 and 9999
        id = (f"MEM{id_number:04d}")  # Format as "MEM" followed by a 4-digit number

        try:
            with open("members.txt", "r") as file:
                for line in file:
                    # Split the line by "|" and extract the MemberID (first column)
                    existing_id = line.strip().split("|")[0].strip()

                    if existing_id == id:  # Compare the new ID with existing IDs
                        exists = True
                        file.close()
                        break  # No need to check further if a match is found

        except FileNotFoundError:
            # If the file doesn't exist, we can assume no IDs are in use yet
            break

    return id
```

This function will generate random four digits numbers and attached to ‘MEM’ to create the member ID. Then, each IDs are compared with the one generated to ensure that there is no repeated member ID.

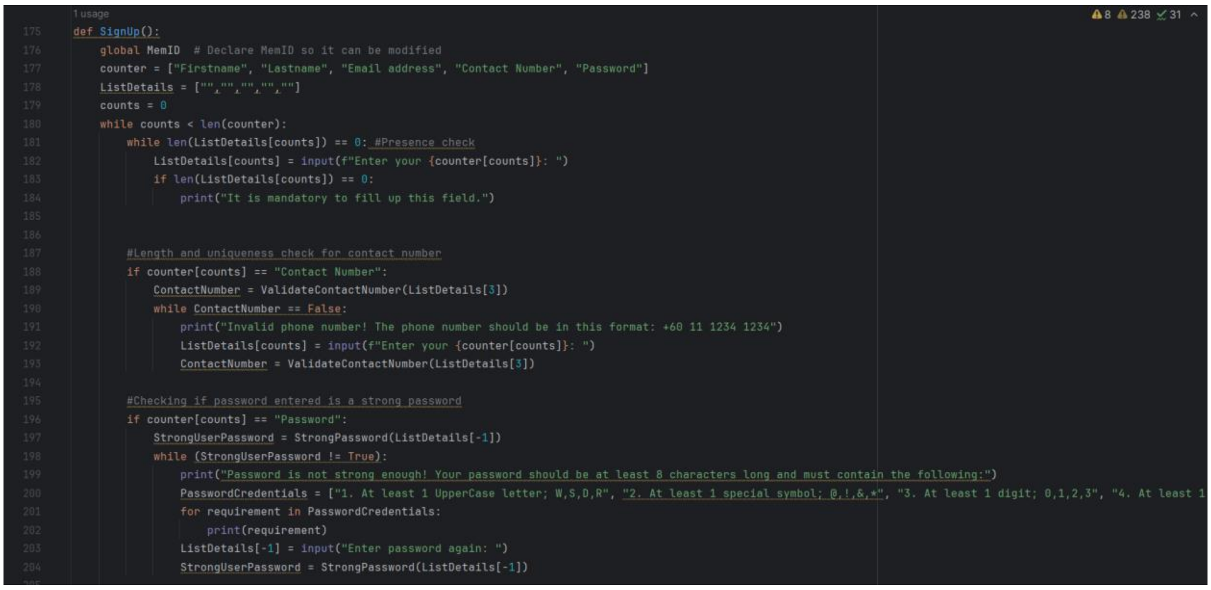
5.1.1.5 Email Address Existence check



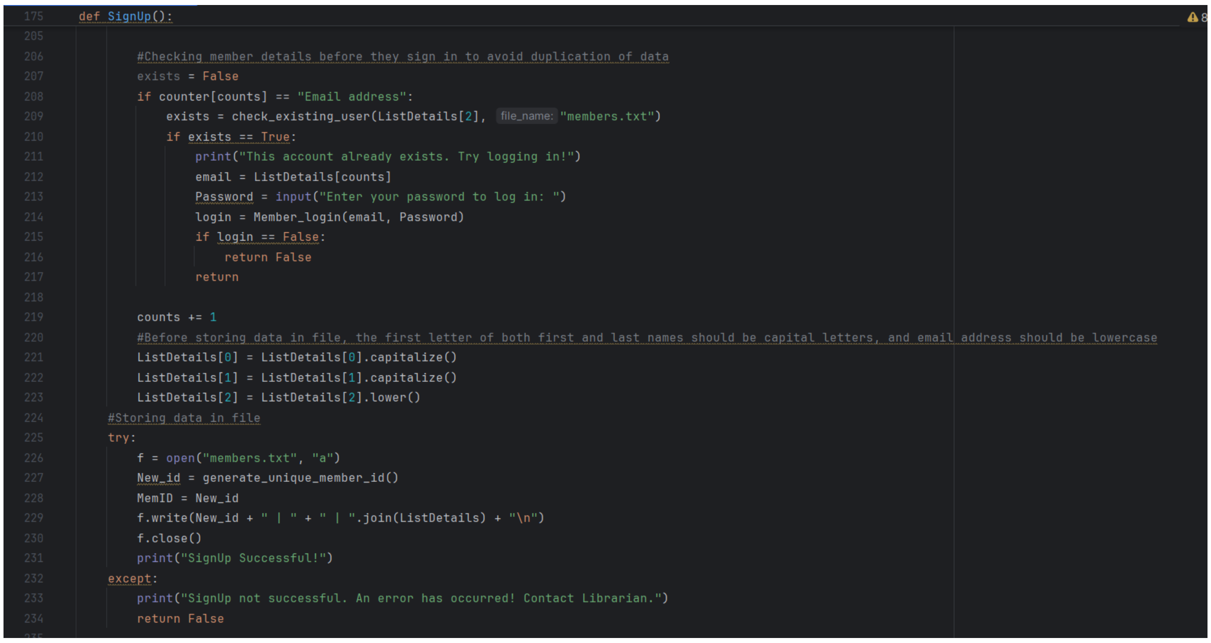
```
4 usages
def check_existing_user(email_to_check, file_name):
    try:
        with open(file_name, "r") as file:
            for line in file:
                # Skip empty lines
                if not line.strip():
                    continue

                # Split the line by "|" and check if there are at least 5 elements (for a valid row)
                parts = [part.strip() for part in line.strip().split(" | ")]

                if len(parts) >= 5:  # Ensure there are enough columns
                    existing_email = parts[3]  # Email is in the 4th column

                    # Compare email addresses
                    if existing_email.lower() == email_to_check.lower():
                        file.close()
                        return True  # Exit as soon as we find a match
            return False  # If no match is found after checking all lines


    except FileNotFoundError:
        print("No user data found.")
        return False
```

The check_existing_user function will check whether the email address entered by the member already exists. If so, instead of signing up, the member will need to log in. Moreover, it is also used when an administrator is adding a new member to the system, so as not to add data that already exists. Since the member ID is generated by the system, the latter cannot be used for this check, and the others, Names may tend to repeat, Contact Number tends to change a lot. Hence, the email address is best for this check.

5.1.1.6 Member Sign Up



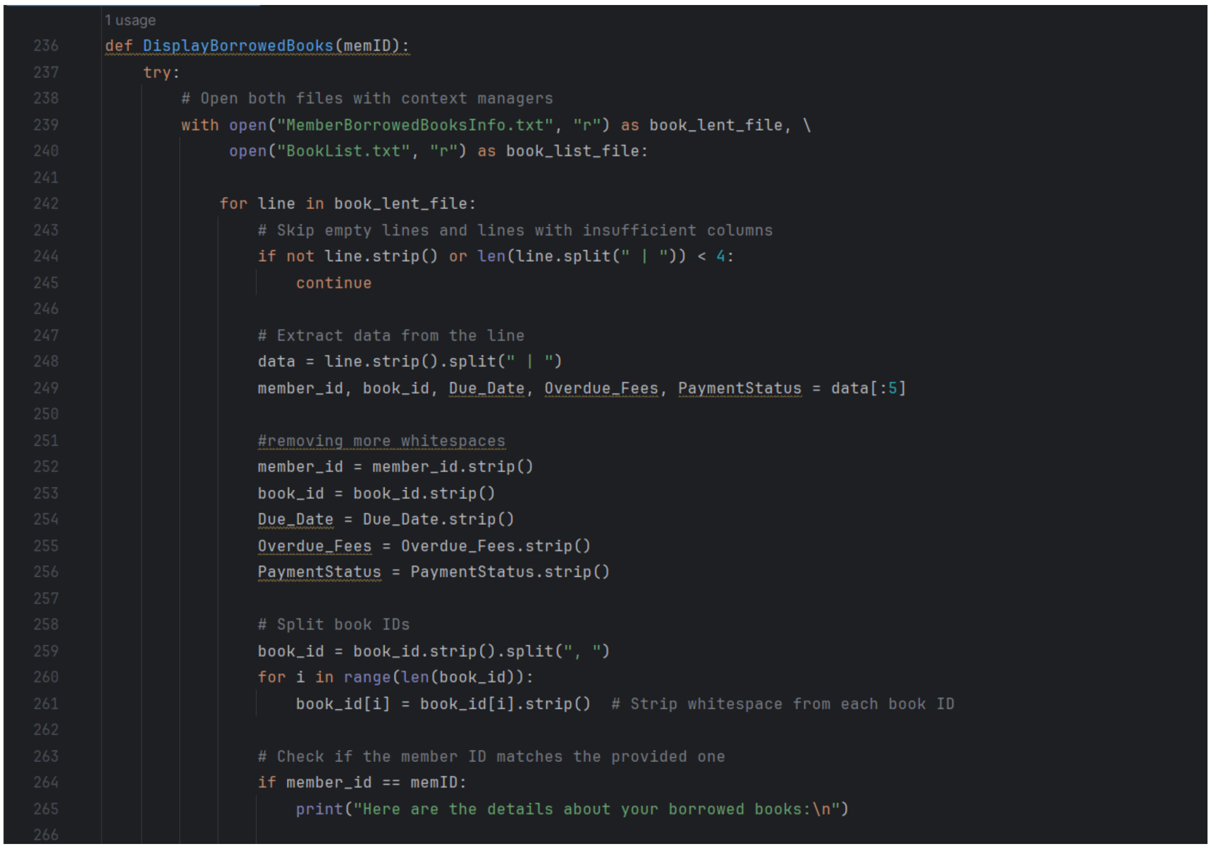
```python
def SignUp():
    global MemID  # Declare MemID so it can be modified
    counter = ["Firstname", "Lastname", "Email address", "Contact Number", "Password"]
    ListDetails = ["","","","",""]
    counts = 0
    while counts < len(counter):
        while len(ListDetails[counts]) == 0: #Presence check
            ListDetails[counts] = input(f"Enter your {counter[counts]}: ")
            if len(ListDetails[counts]) == 0:
                print("It is mandatory to fill up this field.")


        #Length and uniqueness check for contact number
        if counter[counts] == "Contact Number":
            ContactNumber = ValidateContactNumber(ListDetails[3])
            while ContactNumber == False:
                print("Invalid phone number! The phone number should be in this format: +60 11 1234 1234")
                ListDetails[counts] = input(f"Enter your {counter[counts]}: ")
                ContactNumber = ValidateContactNumber(ListDetails[3])

        #Checking if password entered is a strong password
        if counter[counts] == "Password":
            StrongUserPassword = StrongPassword(ListDetails[-1])
            while (StrongUserPassword != True):
                print("Password is not strong enough! Your password should be at least 8 characters long and must contain the following:")
                PasswordCredentials = ["1. At least 1 UpperCase letter; W,S,D,R", "2. At least 1 special symbol; @,!,&,*", "3. At least 1 digit; 0,1,2,3", "4. At least 1
                for requirement in PasswordCredentials:
                    print(requirement)
                ListDetails[-1] = input("Enter password again: ")
                StrongUserPassword = StrongPassword(ListDetails[-1])
```



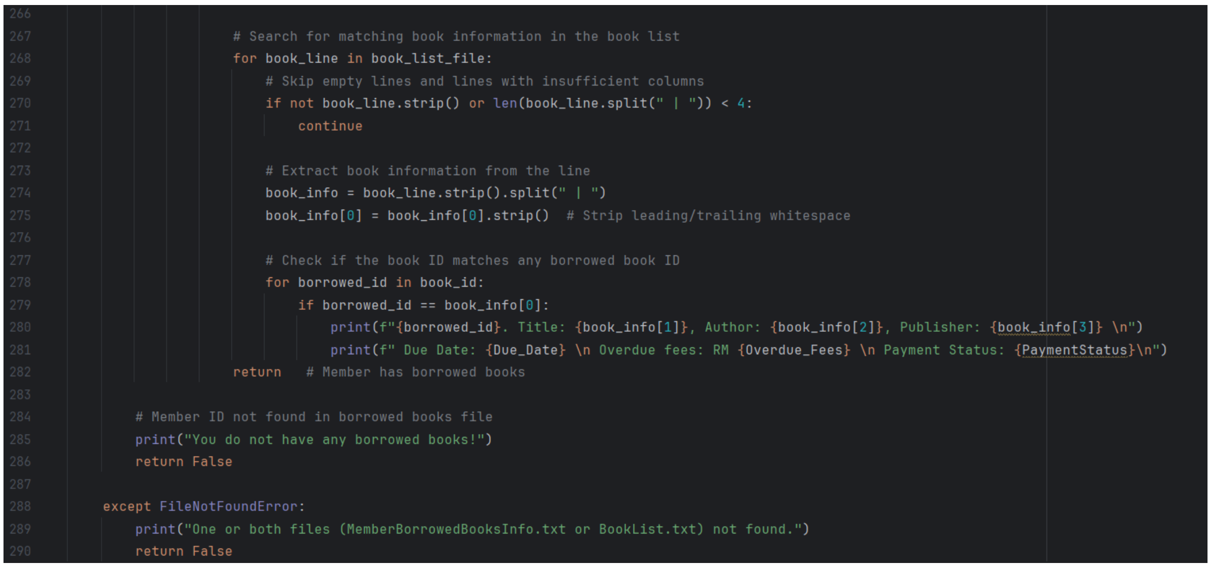
```python
def SignUp():

        #Checking member details before they sign in to avoid duplication of data
        exists = False
        if counter[counts] == "Email address":
            exists = check_existing_user(ListDetails[2], file_name: "members.txt")
            if exists == True:
                print("This account already exists. Try logging in!")
                email = ListDetails[counts]
                Password = input("Enter your password to log in: ")
                login = Member_login(email, Password)
                if login == False:
                    return False
                return

        counts += 1
        #Before storing data in file, the first letter of both first and last names should be capital letters, and email address should be lowercase
        ListDetails[0] = ListDetails[0].capitalize()
        ListDetails[1] = ListDetails[1].capitalize()
        ListDetails[2] = ListDetails[2].lower()
    #Storing data in file
    try:
        f = open("members.txt", "a")
        New_id = generate_unique_member_id()
        MemID = New_id
        f.write(New_id + " | " + " | ".join(ListDetails) + "\n")
        f.close()
        print("SignUp Successful!")
    except:
        print("SignUp not successful. An error has occurred! Contact Librarian.")
        return False
```

The SignUp () function will make members enter their details, and the other functions explained above are used to verify and validate the details entered, ensuring that they are relevant. Then, the new member's data is appended in the members.txt file.

5.1.1.7 Display members' borrowed books List



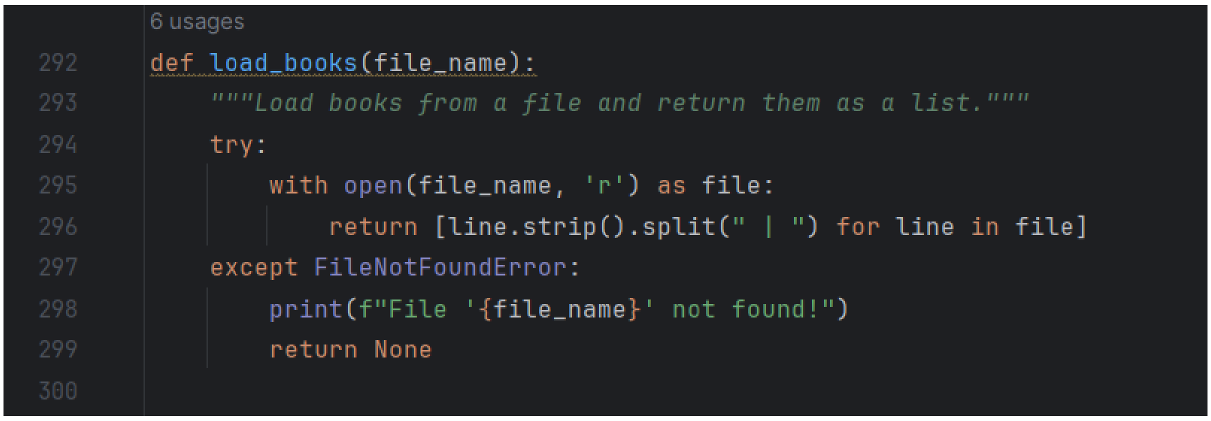
```python
def DisplayBorrowedBooks(memID):
    try:
        # Open both files with context managers
        with open("MemberBorrowedBooksInfo.txt", "r") as book_lent_file, \
             open("BookList.txt", "r") as book_list_file:

            for line in book_lent_file:
                # Skip empty lines and lines with insufficient columns
                if not line.strip() or len(line.split(" | ")) < 4:
                    continue

                # Extract data from the line
                data = line.strip().split(" | ")
                member_id, book_id, Due_Date, Overdue_Fees, PaymentStatus = data[:5]

                #removing more whitespaces
                member_id = member_id.strip()
                book_id = book_id.strip()
                Due_Date = Due_Date.strip()
                Overdue_Fees = Overdue_Fees.strip()
                PaymentStatus = PaymentStatus.strip()

                # Split book IDs
                book_id = book_id.strip().split(", ")
                for i in range(len(book_id)):
                    book_id[i] = book_id[i].strip()  # Strip whitespace from each book ID

                # Check if the member ID matches the provided one
                if member_id == memID:
                    print("Here are the details about your borrowed books:\n")

```



```python

                    # Search for matching book information in the book list
                    for book_line in book_list_file:
                        # Skip empty lines and lines with insufficient columns
                        if not book_line.strip() or len(book_line.split(" | ")) < 4:
                            continue

                        # Extract book information from the line
                        book_info = book_line.strip().split(" | ")
                        book_info[0] = book_info[0].strip()  # Strip leading/trailing whitespace

                        # Check if the book ID matches any borrowed book ID
                        for borrowed_id in book_id:
                            if borrowed_id == book_info[0]:
                                print(f"{borrowed_id}. Title: {book_info[1]}, Author: {book_info[2]}, Publisher: {book_info[3]} \n")
                                print(f" Due Date: {Due_Date} \n Overdue fees: RM {Overdue_Fees} \n Payment Status: {PaymentStatus}\n")
                    return   # Member has borrowed books

        # Member ID not found in borrowed books file
        print("You do not have any borrowed books!")
        return False

    except FileNotFoundError:
        print("One or both files (MemberBorrowedBooksInfo.txt or BookList.txt) not found.")
        return False
```

This function will display the book ID of books borrowed by a member, the return due date, overdue fees, and the payment status of the member. With the Book ID found in the MembersBorrowedBooksInfo.txt file, the details of the book itself, like author, title, and publisher, are also displayed to the member, from the BookList.txt file.

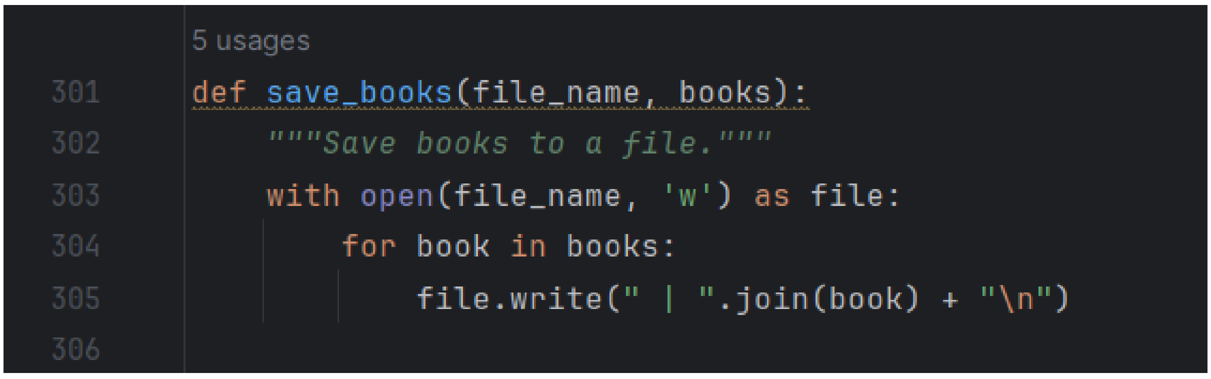
5.1.1.8 Loading File function



```
6 usages
def load_books(file_name):
    """Load books from a file and return them as a list."""
    try:
        with open(file_name, 'r') as file:
            return [line.strip().split(" | ") for line in file]
    except FileNotFoundError:
        print(f"File '{file_name}' not found!")
        return None
```

This function will load data of a specific file as a list when it is called.

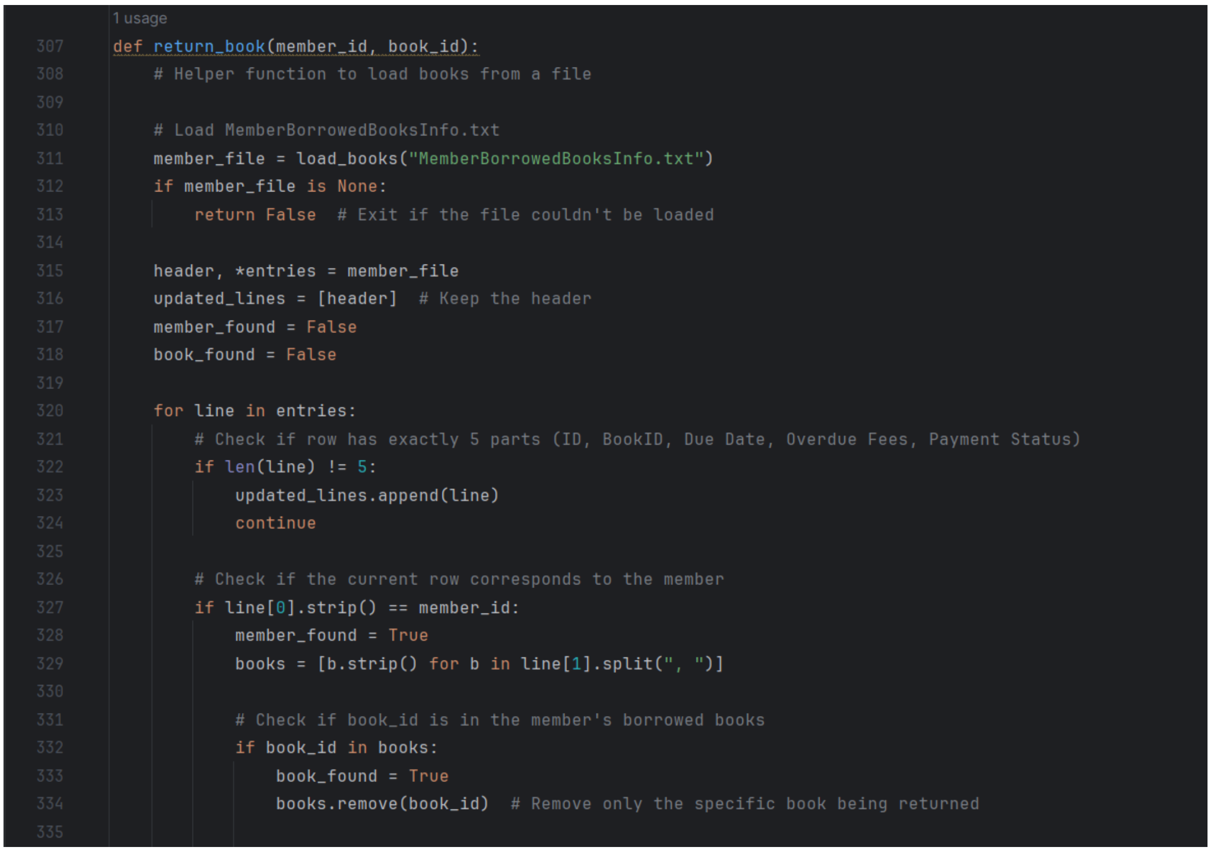
5.1.1.9 Saving and Storing Updated File Function



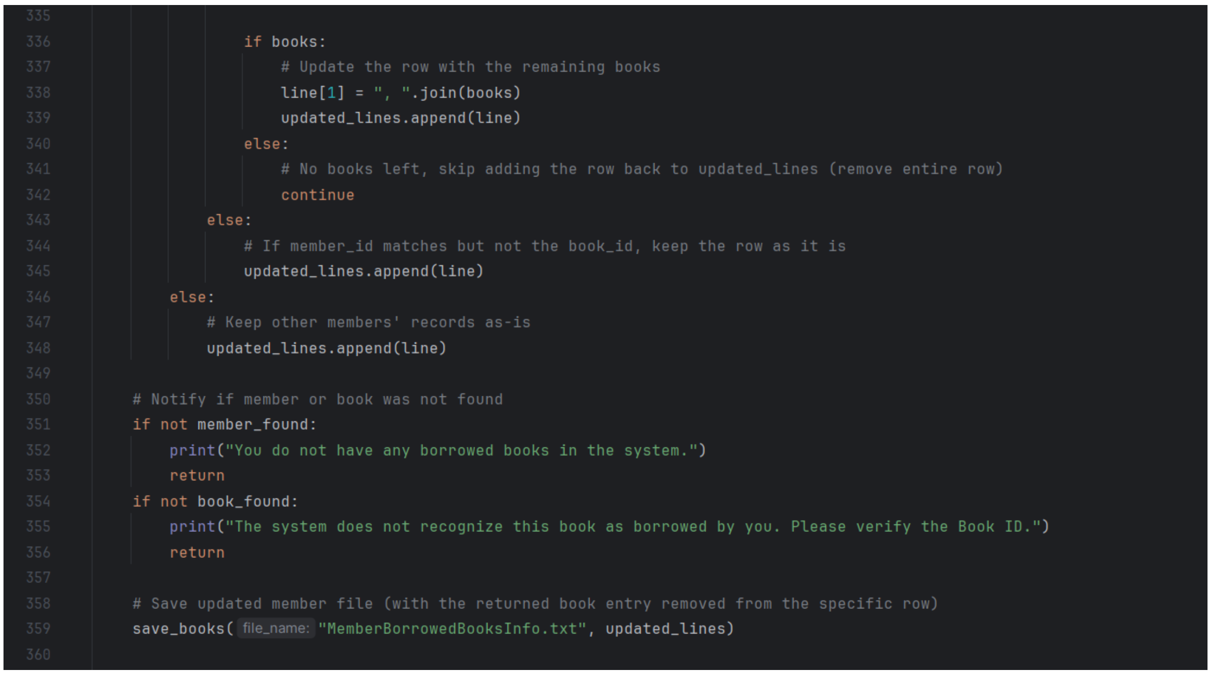
```
5 usages
def save_books(file_name, books):
    """Save books to a file."""
    with open(file_name, 'w') as file:
        for book in books:
            file.write(" | ".join(book) + "\n")
```

This function will save data found in variable ‘books’, in any text file, found in variable ‘file_name’.

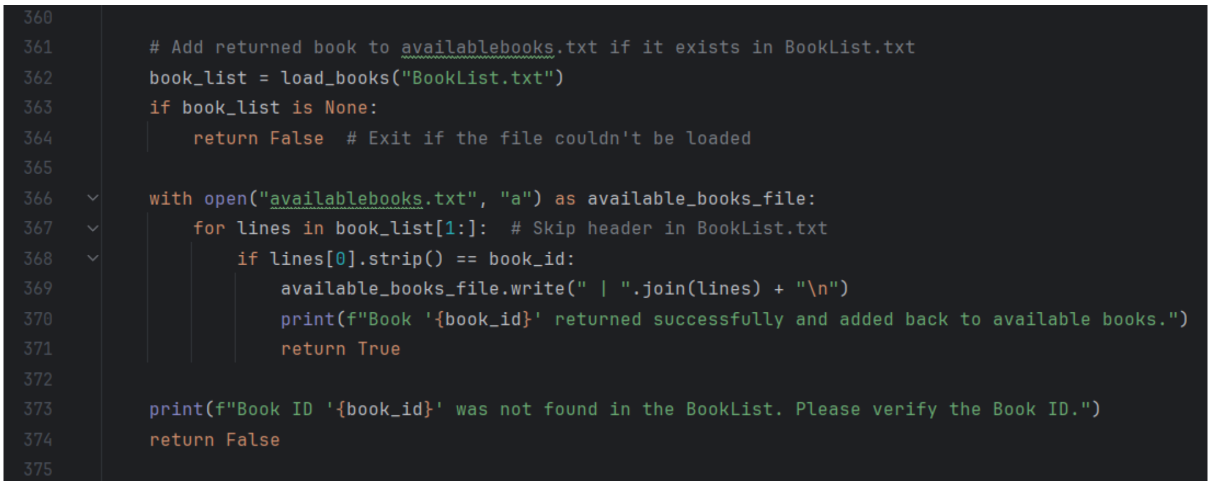
5.1.1.10 Return Book Function



```
1 usage
def return_book(member_id, book_id):
    # Helper function to load books from a file

    # Load MemberBorrowedBooksInfo.txt
    member_file = load_books("MemberBorrowedBooksInfo.txt")
    if member_file is None:
        return False  # Exit if the file couldn't be loaded

    header, *entries = member_file
    updated_lines = [header]  # Keep the header
    member_found = False
    book_found = False

    for line in entries:
        # Check if row has exactly 5 parts (ID, BookID, Due Date, Overdue Fees, Payment Status)
        if len(line) != 5:
            updated_lines.append(line)
            continue

        # Check if the current row corresponds to the member
        if line[0].strip() == member_id:
            member_found = True
            books = [b.strip() for b in line[1].split(", ")]

            # Check if book_id is in the member's borrowed books
            if book_id in books:
                book_found = True
                books.remove(book_id)  # Remove only the specific book being returned

```



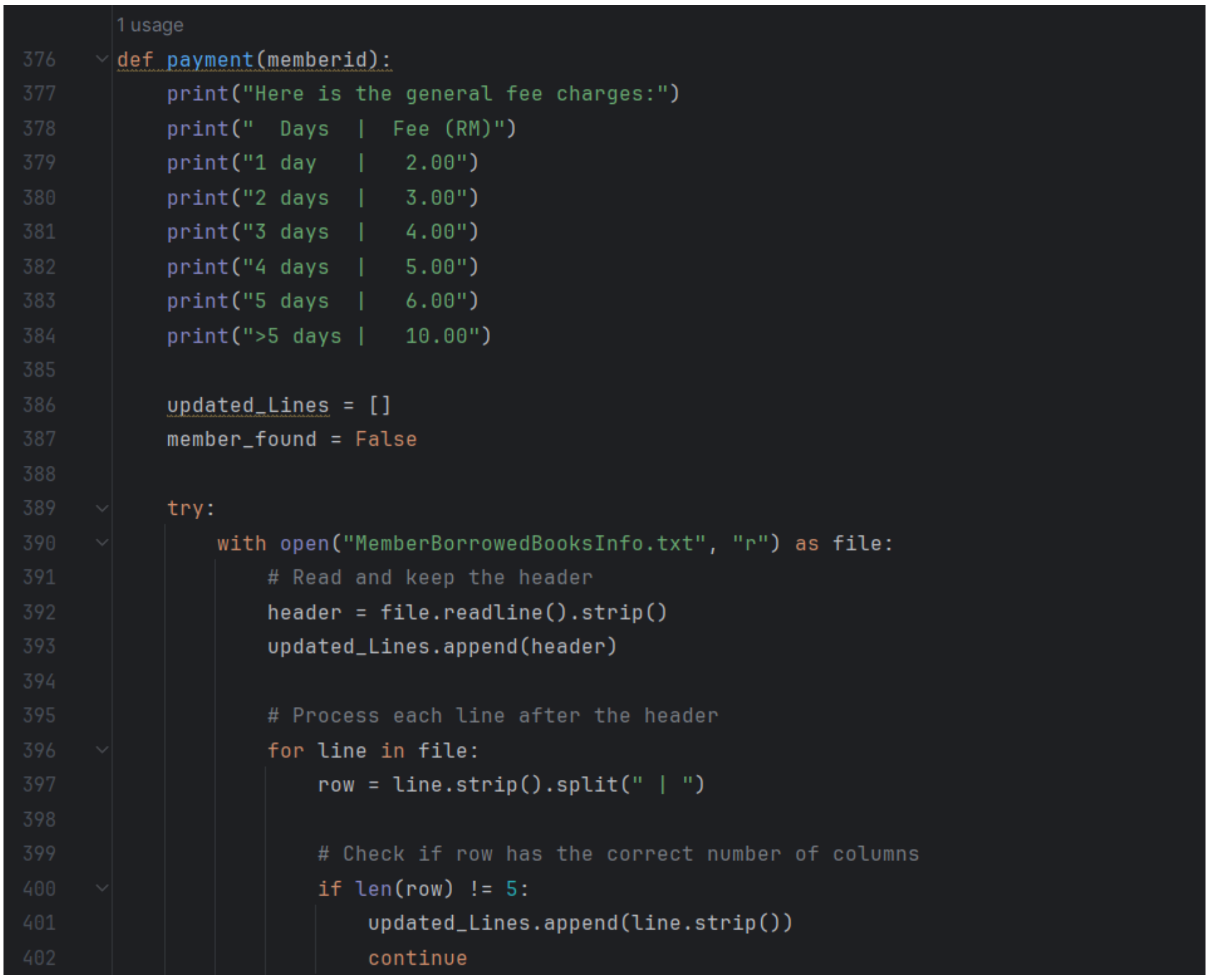
```

                if books:
                    # Update the row with the remaining books
                    line[1] = ", ".join(books)
                    updated_lines.append(line)
                else:
                    # No books left, skip adding the row back to updated_lines (remove entire row)
                    continue
            else:
                # If member_id matches but not the book_id, keep the row as it is
                updated_lines.append(line)
        else:
            # Keep other members' records as-is
            updated_lines.append(line)

    # Notify if member or book was not found
    if not member_found:
        print("You do not have any borrowed books in the system.")
        return
    if not book_found:
        print("The system does not recognize this book as borrowed by you. Please verify the Book ID.")
        return

    # Save updated member file (with the returned book entry removed from the specific row)
    save_books( file_name: "MemberBorrowedBooksInfo.txt", updated_lines)

```



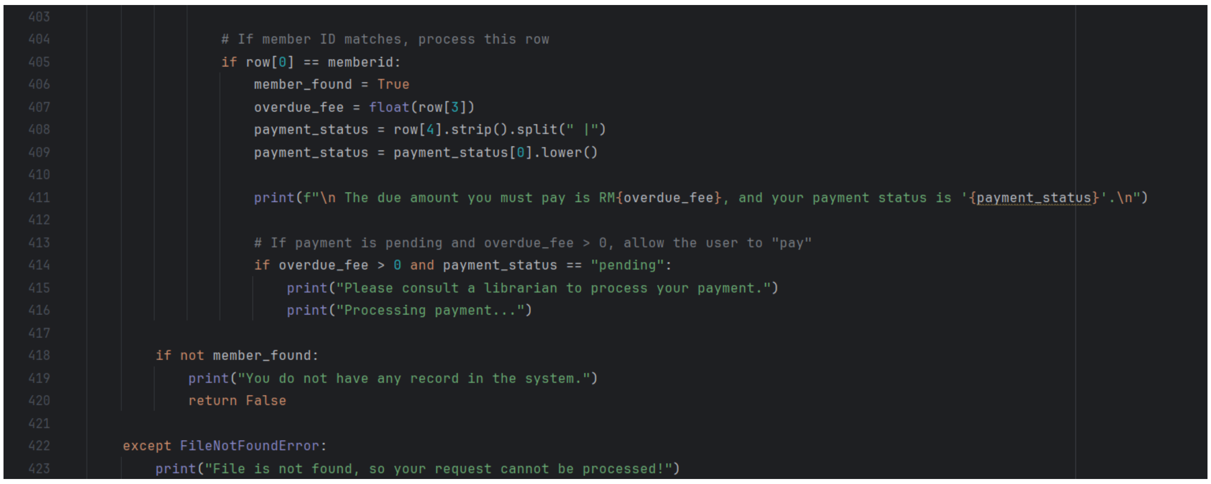
```python

    # Add returned book to availablebooks.txt if it exists in BookList.txt
    book_list = load_books("BookList.txt")
    if book_list is None:
        return False  # Exit if the file couldn't be loaded

    with open("availablebooks.txt", "a") as available_books_file:
        for lines in book_list[1:]:  # Skip header in BookList.txt
            if lines[0].strip() == book_id:
                available_books_file.write(" | ".join(lines) + "\n")
                print(f"Book '{book_id}' returned successfully and added back to available books.")
                return True

    print(f"Book ID '{book_id}' was not found in the BookList. Please verify the Book ID.")
    return False

```

This function will allow members to return books by themselves. They will need to input their member ID and the Book ID of the borrowed book. Then, the system would verify if the data inputted match the data found in MembersBorrowedBooksInfo.txt. If so, the row of the member details in the MembersBorrowedBooksInfo.txt is deleted and the Book ID along with the book's details, derived from the BookList.txt, is appended to the availablebooks.txt file, since now the book is available for others to borrow.

5.1.1.11 Payment Function



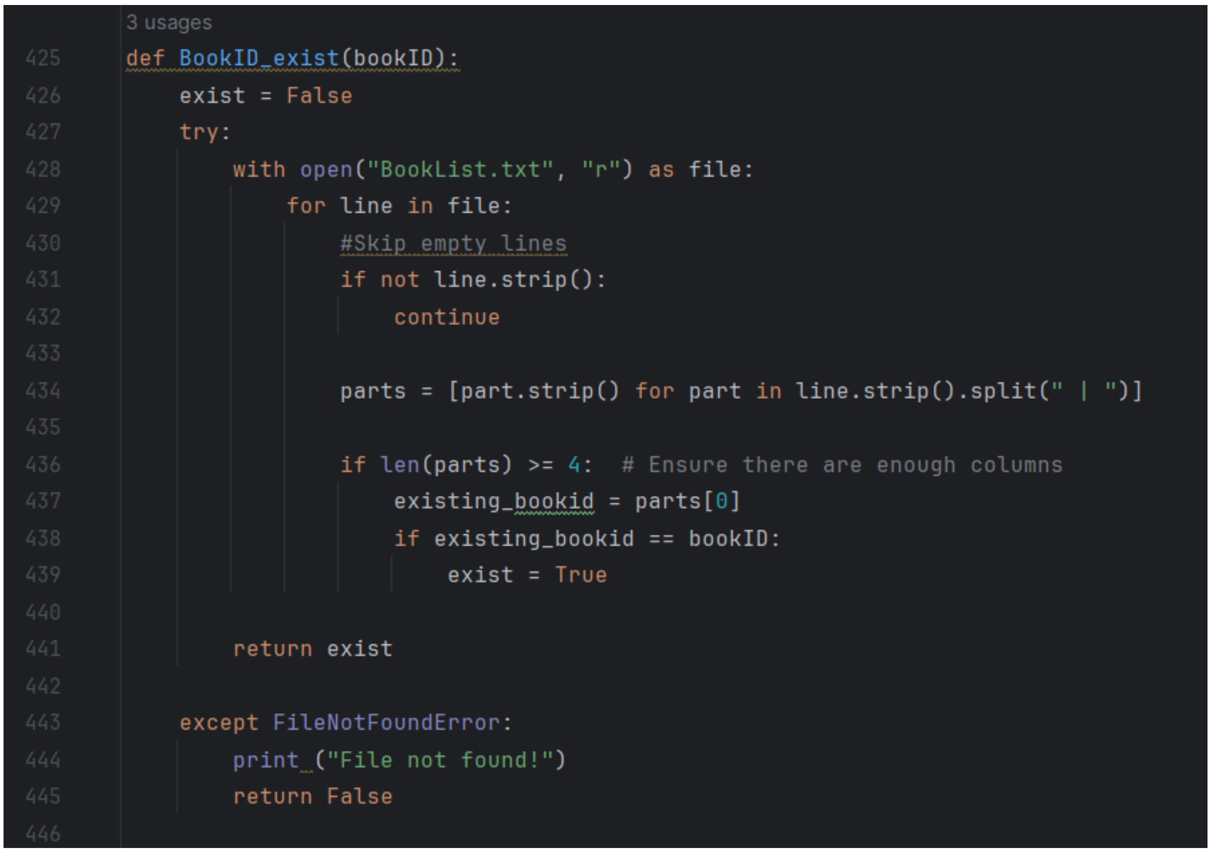
```python
1 usage
def payment(memberid):
    print("Here is the general fee charges:")
    print("  Days  |  Fee (RM)")
    print("1 day   |   2.00")
    print("2 days  |   3.00")
    print("3 days  |   4.00")
    print("4 days  |   5.00")
    print("5 days  |   6.00")
    print(">5 days |   10.00")

    updated_Lines = []
    member_found = False

    try:
        with open("MemberBorrowedBooksInfo.txt", "r") as file:
            # Read and keep the header
            header = file.readline().strip()
            updated_Lines.append(header)

            # Process each line after the header
            for line in file:
                row = line.strip().split(" | ")

                # Check if row has the correct number of columns
                if len(row) != 5:
                    updated_Lines.append(line.strip())
                    continue
```



```python

                # If member ID matches, process this row
                if row[0] == memberid:
                    member_found = True
                    overdue_fee = float(row[3])
                    payment_status = row[4].strip().split(" |")
                    payment_status = payment_status[0].lower()

                    print(f"\n The due amount you must pay is RM{overdue_fee}, and your payment status is '{payment_status}'.\n")

                    # If payment is pending and overdue_fee > 0, allow the user to "pay"
                    if overdue_fee > 0 and payment_status == "pending":
                        print("Please consult a librarian to process your payment.")
                        print("Processing payment...")

        if not member_found:
            print("You do not have any record in the system.")
            return False

    except FileNotFoundError:
        print("File is not found, so your request cannot be processed!")
```

This function will inform members if they have any pending fees and the amount that they need to pay. They will need to contact a librarian, as system administrators are generally in offices, and then they will inform the system administrators to process the payment. When the payment is processed, overdue fees is set to zero and payment status is set to "- ".

5.1.1.12 Book ID Validation Function

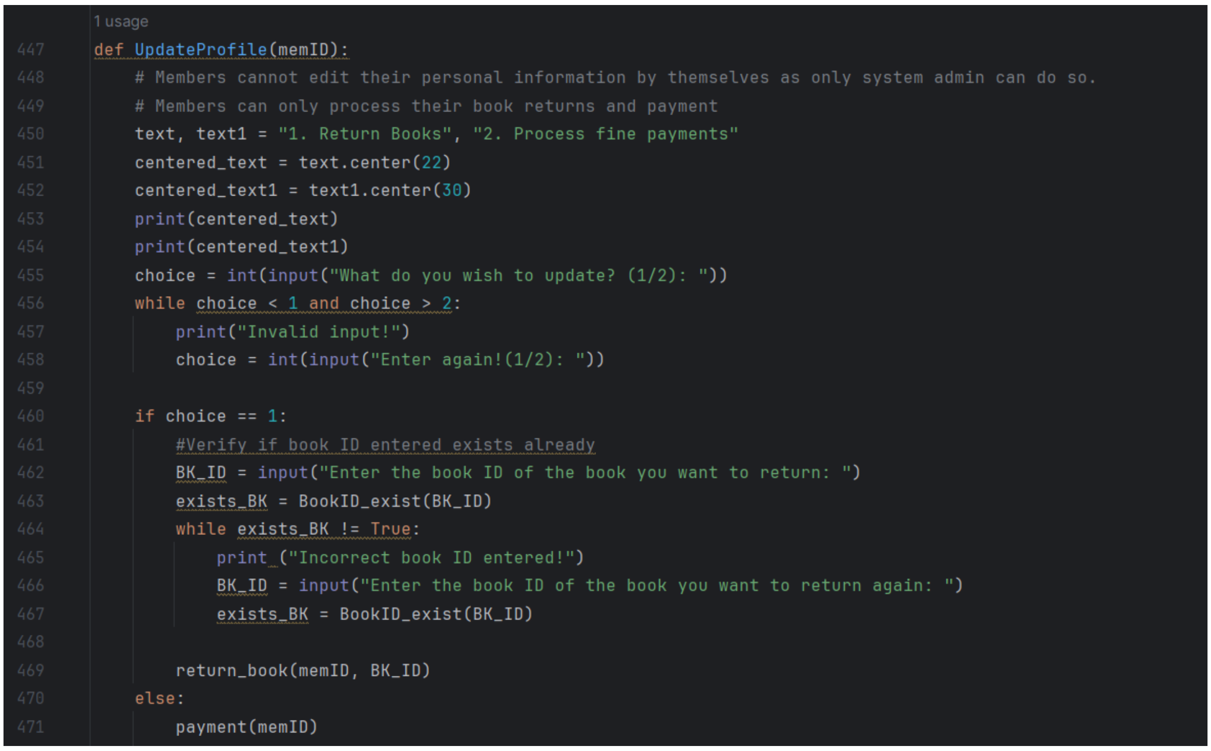


3 usages

```python
def BookID_exist(bookID):
    exist = False
    try:
        with open("BookList.txt", "r") as file:
            for line in file:
                #Skip empty lines
                if not line.strip():
                    continue

                parts = [part.strip() for part in line.strip().split(" | ")]

                if len(parts) >= 4:  # Ensure there are enough columns
                    existing_bookid = parts[0]
                    if existing_bookid == bookID:
                        exist = True

        return exist

    except FileNotFoundError:
        print ("File not found!")
        return False
```

This function will check if Book ID entered exist in the system. It will return ‘True’ if so, otherwise it will return ‘False’.

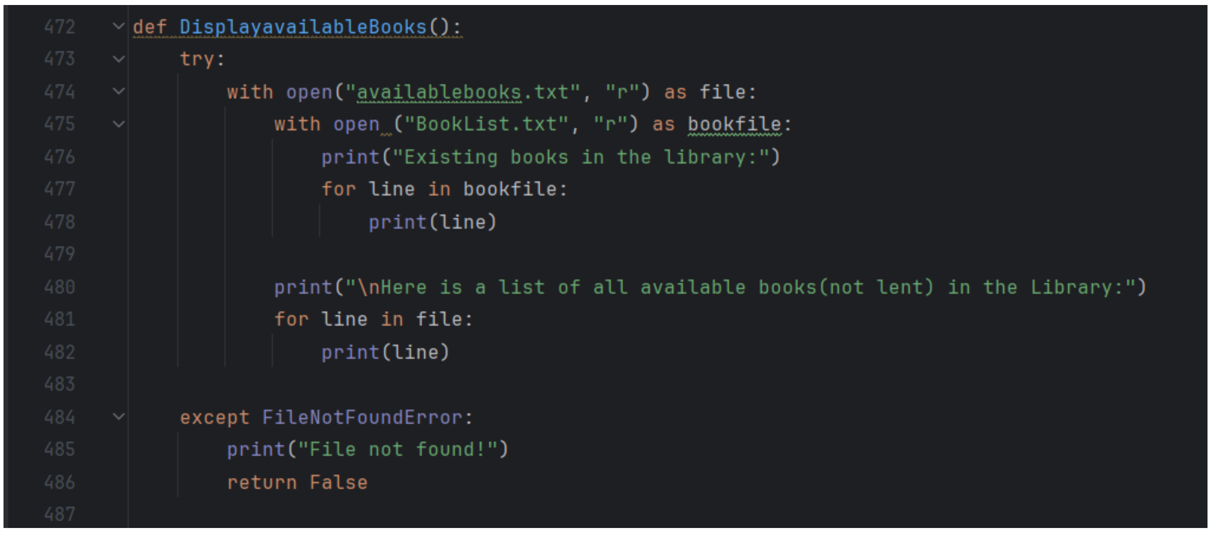
5.1.1.13 Updating members' profile function



```python
def UpdateProfile(memID):
    # Members cannot edit their personal information by themselves as only system admin can do so.
    # Members can only process their book returns and payment
    text, text1 = "1. Return Books", "2. Process fine payments"
    centered_text = text.center(22)
    centered_text1 = text1.center(30)
    print(centered_text)
    print(centered_text1)
    choice = int(input("What do you wish to update? (1/2): "))
    while choice < 1 and choice > 2:
        print("Invalid input!")
        choice = int(input("Enter again!(1/2): "))

    if choice == 1:
        #Verify if book ID entered exists already
        BK_ID = input("Enter the book ID of the book you want to return: ")
        exists_BK = BookID_exist(BK_ID)
        while exists_BK != True:
            print ("Incorrect book ID entered!")
            BK_ID = input("Enter the book ID of the book you want to return again: ")
            exists_BK = BookID_exist(BK_ID)

        return_book(memID, BK_ID)
    else:
        payment(memID)
```

The UpdateProfile () function will call the return_book () function and payment () function. Before assigning Book ID to the return_book () function, it will verify whether the BookID exists in the system. If not, it will inform the member that the Book ID is incorrect.

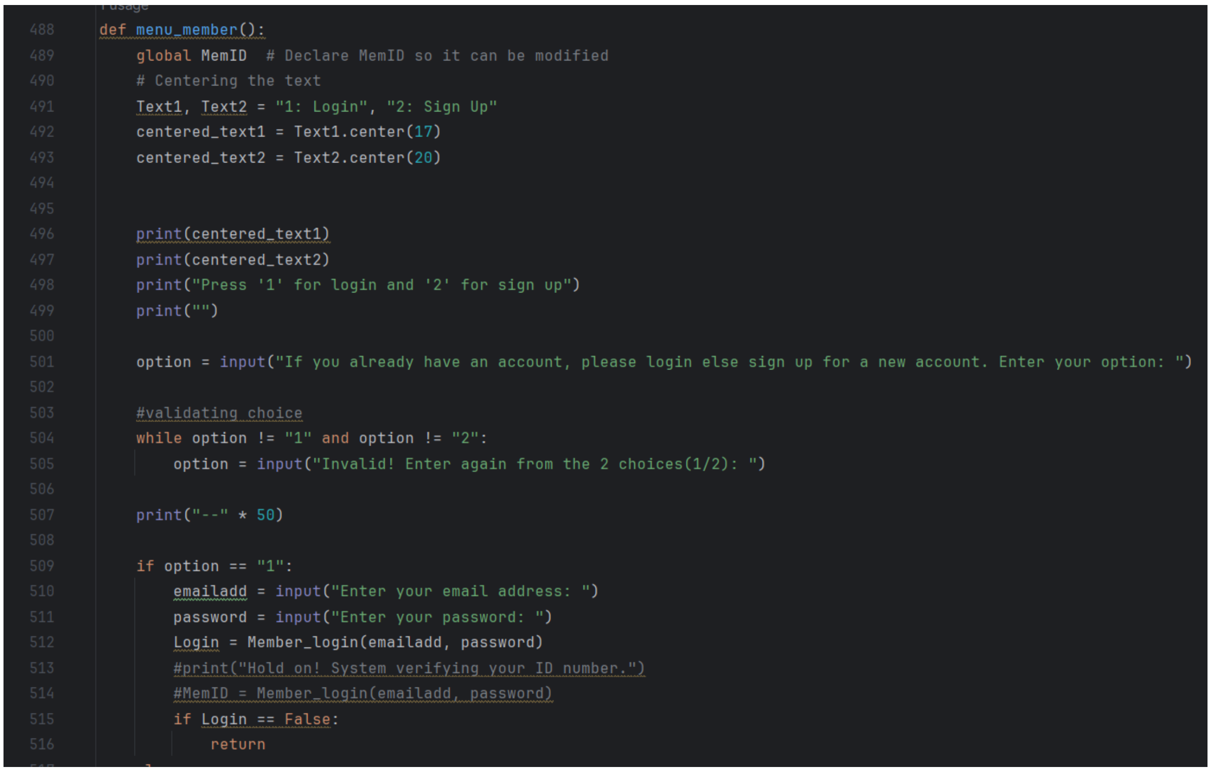
5.1.1.14 Displaying available Books list



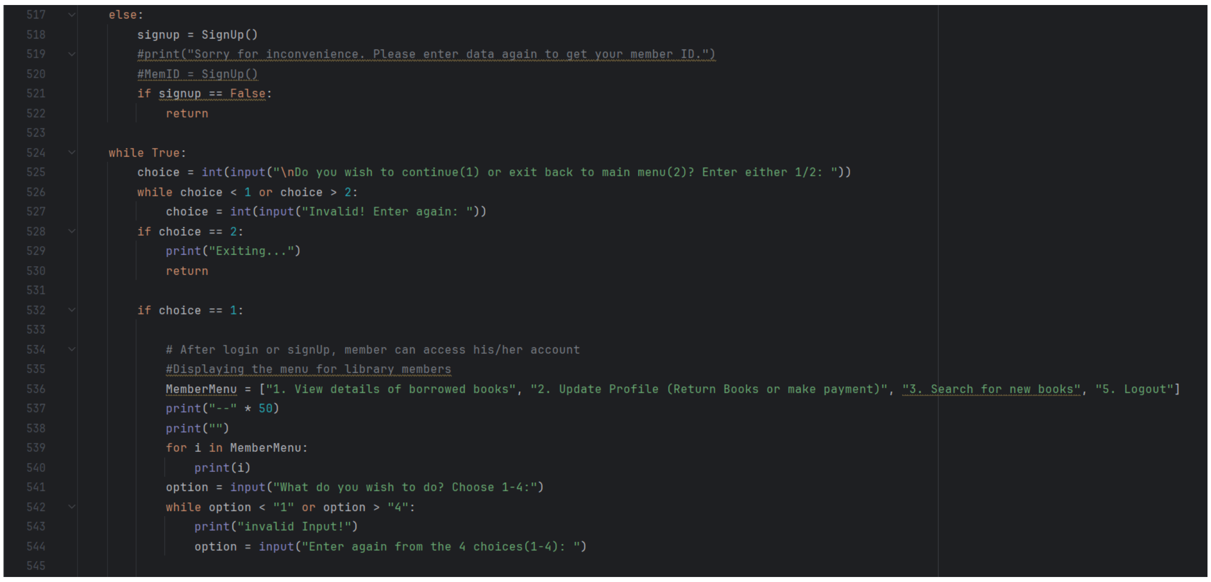
```python
def DisplayavailableBooks():
    try:
        with open("availablebooks.txt", "r") as file:
            with open ("BookList.txt", "r") as bookfile:
                print("Existing books in the library:")
                for line in bookfile:
                    print(line)

            print("\nHere is a list of all available books(not lent) in the Library:")
            for line in file:
                print(line)

    except FileNotFoundError:
        print("File not found!")
        return False
```

This function will print all the books available in the library and it will also display those that are not lent, available to be borrowed.

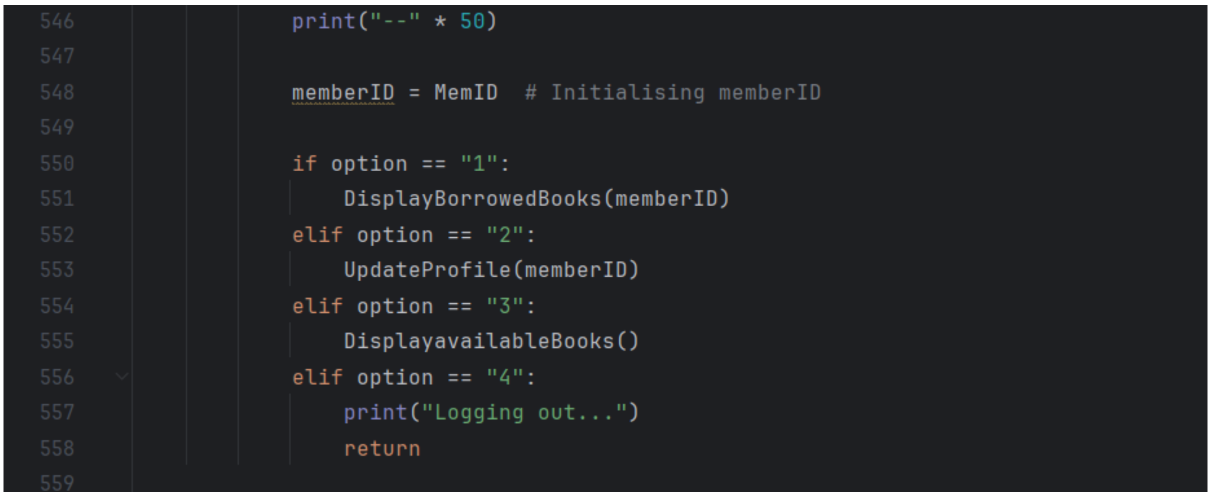
5.1.1.15 Members' menu



```python
def menu_member():
    global MemID  # Declare MemID so it can be modified
    # Centering the text
    Text1, Text2 = "1: Login", "2: Sign Up"
    centered_text1 = Text1.center(17)
    centered_text2 = Text2.center(20)


    print(centered_text1)
    print(centered_text2)
    print("Press '1' for login and '2' for sign up")
    print("")

    option = input("If you already have an account, please login else sign up for a new account. Enter your option: ")

    #validating choice
    while option != "1" and option != "2":
        option = input("Invalid! Enter again from the 2 choices(1/2): ")

    print("--" * 50)

    if option == "1":
        emailadd = input("Enter your email address: ")
        password = input("Enter your password: ")
        Login = Member_login(emailadd, password)
        #print("Hold on! System verifying your ID number.")
        #MemID = Member_login(emailadd, password)
        if Login == False:
            return
```



```python
    else:
        signup = SignUp()
        #print("Sorry for inconvenience. Please enter data again to get your member ID.")
        #MemID = SignUp()
        if signup == False:
            return

    while True:
        choice = int(input("\nDo you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: "))
        while choice < 1 or choice > 2:
            choice = int(input("Invalid! Enter again: "))
        if choice == 2:
            print("Exiting...")
            return

        if choice == 1:

            # After login or signUp, member can access his/her account
            #Displaying the menu for library members
            MemberMenu = ["1. View details of borrowed books", "2. Update Profile (Return Books or make payment)", "3. Search for new books", "5. Logout"]
            print("--" * 50)
            print("")
            for i in MemberMenu:
                print(i)
            option = input("What do you wish to do? Choose 1-4:")
            while option < "1" or option > "4":
                print("invalid Input!")
                option = input("Enter again from the 4 choices(1-4): ")

```

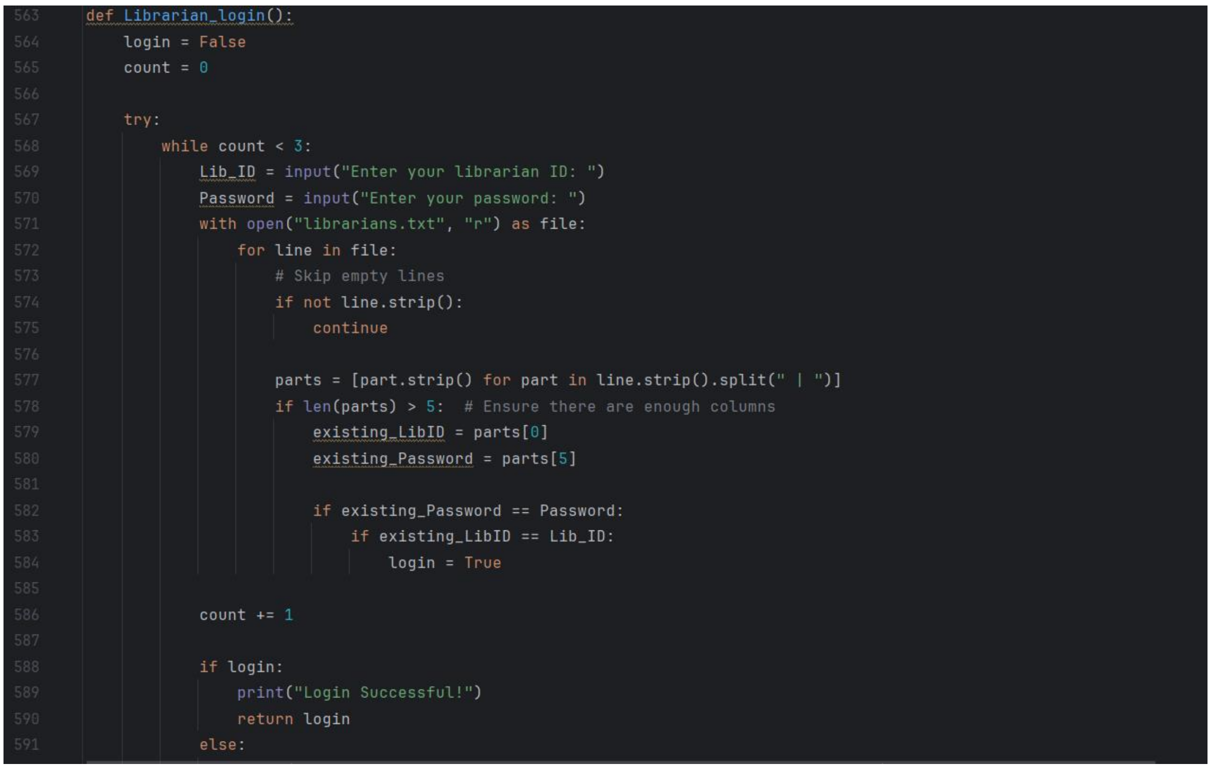


```
            print("--" * 50)

            memberID = MemID  # Initialising memberID

            if option == "1":
                DisplayBorrowedBooks(memberID)
            elif option == "2":
                UpdateProfile(memberID)
            elif option == "3":
                DisplayavailableBooks()
            elif option == "4":
                print("Logging out...")
                return

```

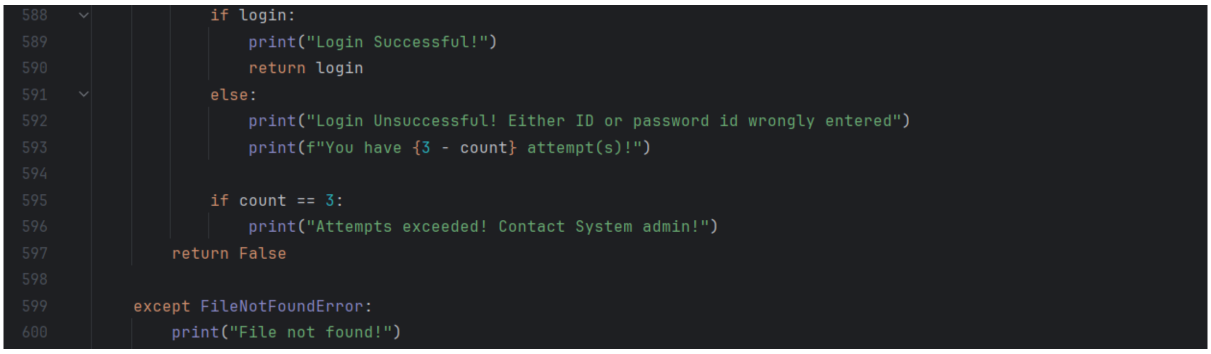
This is the members' menu, which will allow them to login, sign up, return books, view Booklist, pay their overdue fees, view their profile and obviously log out of the system, once they are done.

5.1.2.0 Librarian User

5.1.2.1 Librarian Login



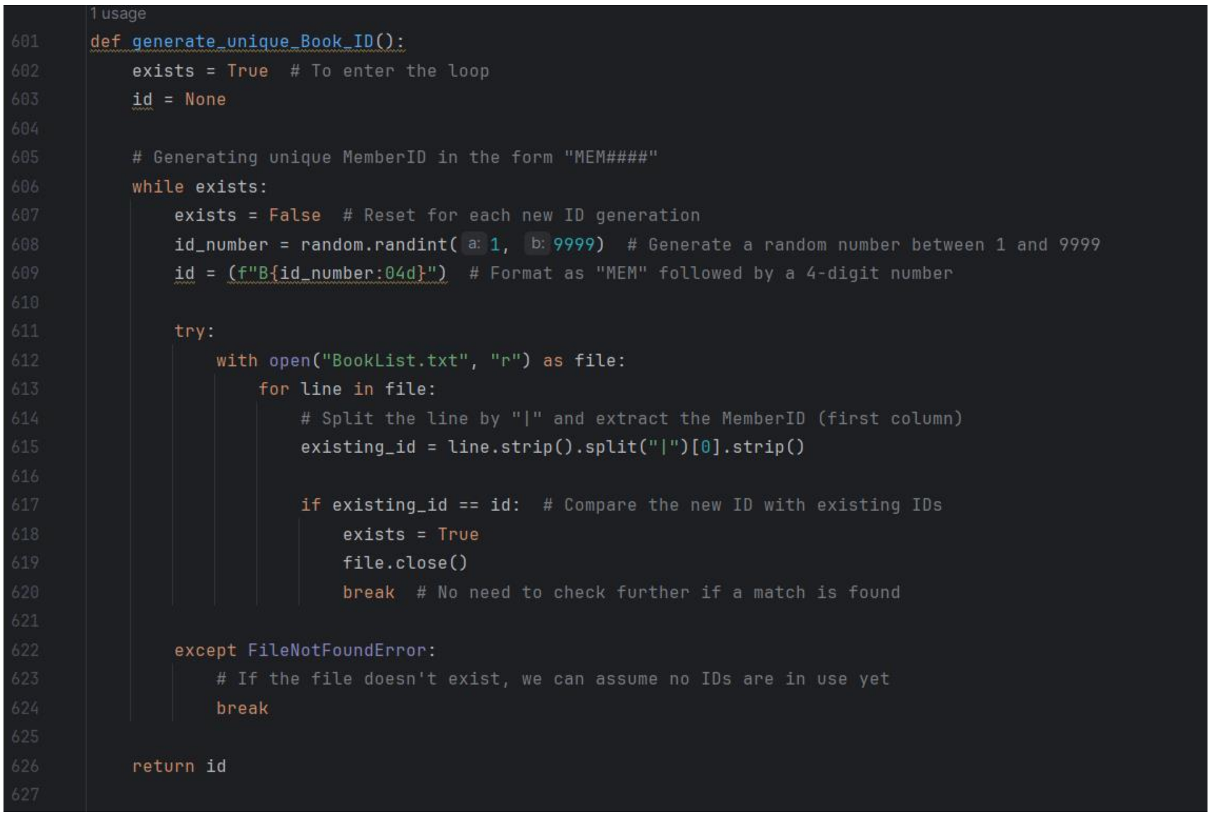
```
def Librarian_login():
    login = False
    count = 0

    try:
        while count < 3:
            Lib_ID = input("Enter your librarian ID: ")
            Password = input("Enter your password: ")
            with open("librarians.txt", "r") as file:
                for line in file:
                    # Skip empty lines
                    if not line.strip():
                        continue

                    parts = [part.strip() for part in line.strip().split(" | ")]
                    if len(parts) > 5:  # Ensure there are enough columns
                        existing_LibID = parts[0]
                        existing_Password = parts[5]

                        if existing_Password == Password:
                            if existing_LibID == Lib_ID:
                                login = True

            count += 1

            if login:
                print("Login Successful!")
                return login
            else:
```



```
            if login:
                print("Login Successful!")
                return login
            else:
                print("Login Unsuccessful! Either ID or password id wrongly entered")
                print(f"You have {3 - count} attempt(s)!")

            if count == 3:
                print("Attempts exceeded! Contact System admin!")
        return False

    except FileNotFoundError:
        print("File not found!")
```

This function will prompt librarian to input their ID and password within three attempts. If those details are found in the librarians.txt file, Login is successful, otherwise it failed. If login is failed, the librarian will exit the menu. Exception handling is also implemented, so if the system cannot open the librarian.txt file, it will output a message informing the user that the file cannot be found.

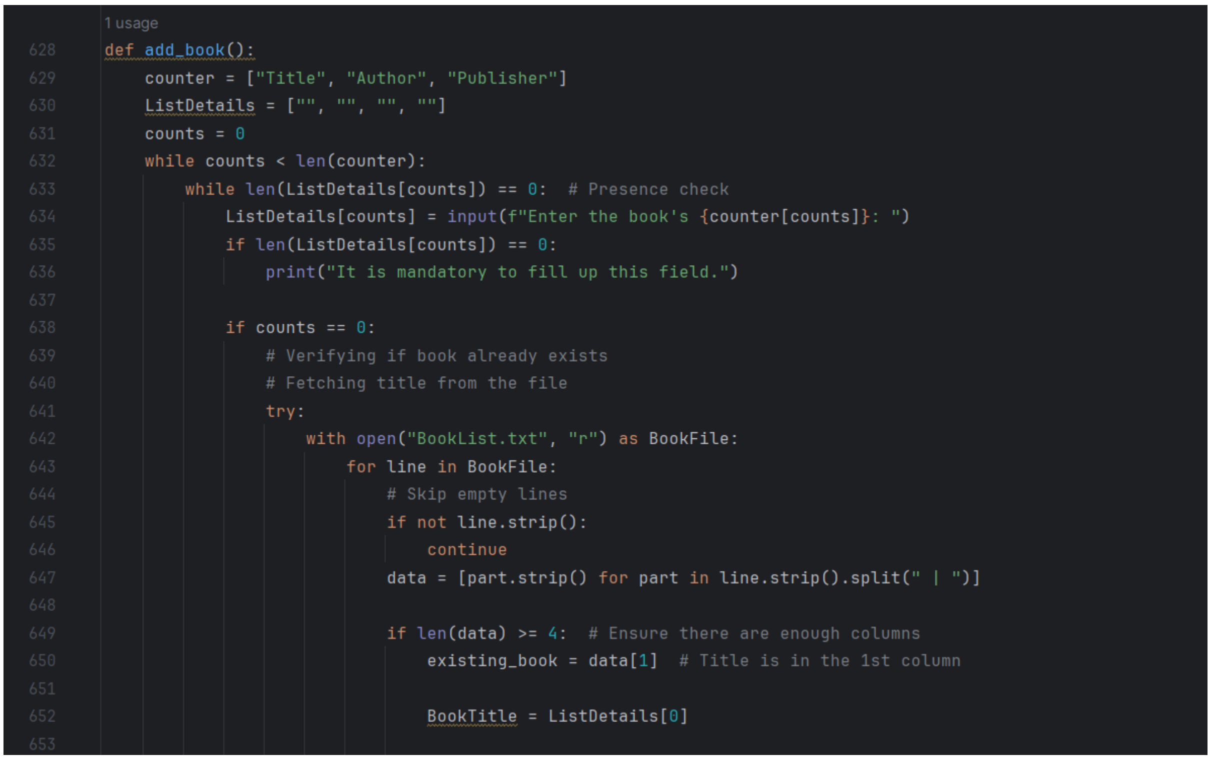
5.1.2.2 Book ID Randomising



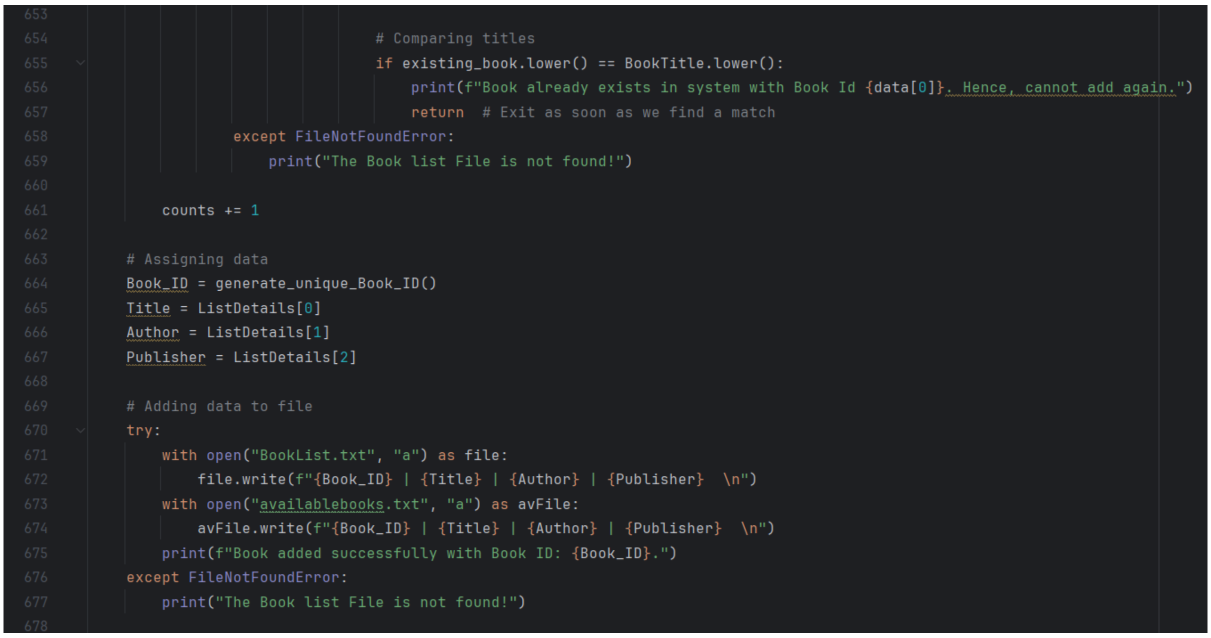
```
1 usage
def generate_unique_Book_ID():
    exists = True  # To enter the loop
    id = None

    # Generating unique MemberID in the form "MEM####"
    while exists:
        exists = False  # Reset for each new ID generation
        id_number = random.randint( a: 1,  b: 9999)  # Generate a random number between 1 and 9999
        id = (f"B{id_number:04d}")  # Format as "MEM" followed by a 4-digit number

        try:
            with open("BookList.txt", "r") as file:
                for line in file:
                    # Split the line by "|" and extract the MemberID (first column)
                    existing_id = line.strip().split("|")[0].strip()

                    if existing_id == id:  # Compare the new ID with existing IDs
                        exists = True
                        file.close()
                        break  # No need to check further if a match is found

        except FileNotFoundError:
            # If the file doesn't exist, we can assume no IDs are in use yet
            break

    return id
```

Random four-digit numbers are generated and attached to ‘B’ to create the Book ID. The system will ensure that the ID generated is unique by carrying out an existence check.

5.1.2.3 Add new book to existing book lists



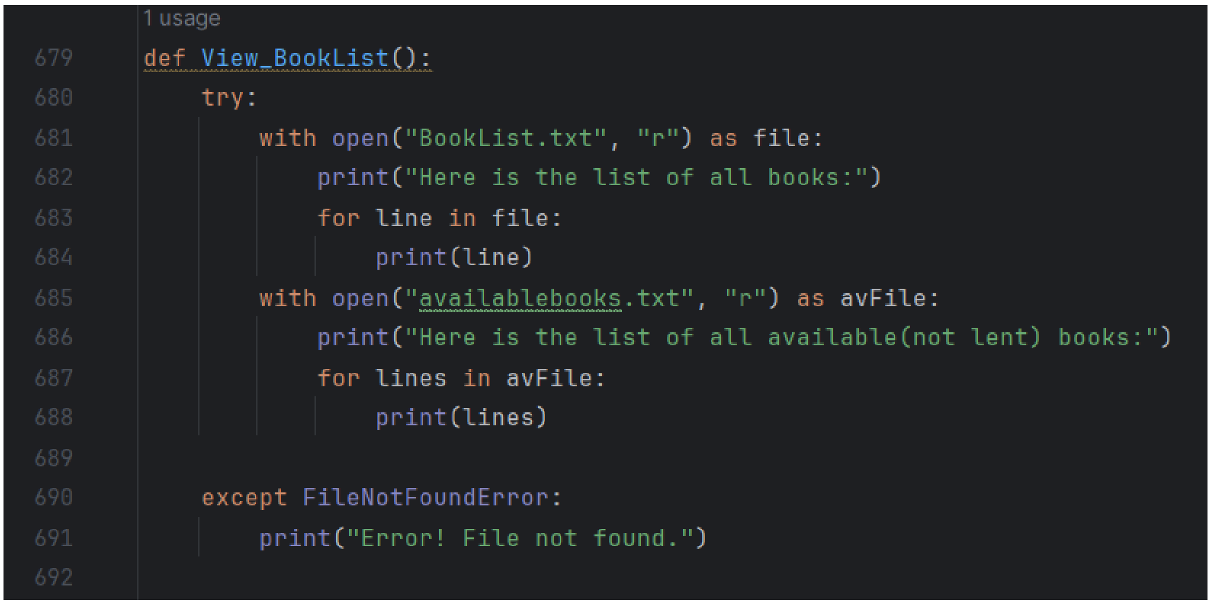
```
1 usage
def add_book():
    counter = ["Title", "Author", "Publisher"]
    ListDetails = ["", "", "", ""]
    counts = 0
    while counts < len(counter):
        while len(ListDetails[counts]) == 0:  # Presence check
            ListDetails[counts] = input(f"Enter the book's {counter[counts]}: ")
            if len(ListDetails[counts]) == 0:
                print("It is mandatory to fill up this field.")

            if counts == 0:
                # Verifying if book already exists
                # Fetching title from the file
                try:
                    with open("BookList.txt", "r") as BookFile:
                        for line in BookFile:
                            # Skip empty lines
                            if not line.strip():
                                continue
                            data = [part.strip() for part in line.strip().split(" | ")]

                            if len(data) >= 4:  # Ensure there are enough columns
                                existing_book = data[1]  # Title is in the 1st column

                                BookTitle = ListDetails[0]

```



```

                                # Comparing titles
                                if existing_book.lower() == BookTitle.lower():
                                    print(f"Book already exists in system with Book Id {data[0]}. Hence, cannot add again.")
                                    return  # Exit as soon as we find a match
                except FileNotFoundError:
                    print("The Book list File is not found!")

        counts += 1

    # Assigning data
    Book_ID = generate_unique_Book_ID()
    Title = ListDetails[0]
    Author = ListDetails[1]
    Publisher = ListDetails[2]

    # Adding data to file
    try:
        with open("BookList.txt", "a") as file:
            file.write(f"{Book_ID} | {Title} | {Author} | {Publisher}  \n")
        with open("availablebooks.txt", "a") as avFile:
            avFile.write(f"{Book_ID} | {Title} | {Author} | {Publisher}  \n")
        print(f"Book added successfully with Book ID: {Book_ID}.")
    except FileNotFoundError:
        print("The Book list File is not found!")

```

This function will allow the librarian to add new books in the BookList.txt file. The user will have to, compulsorily, enter the details of the book like the title, author, and others. The book ID is generated automatically by calling the generate_unique_book_id () function. If the book has an identical title in the BookList.txt file, it is assumed that the book already exists, so the program will output an error message. If book details are successfully appended in the BookList.txt file, the availablebooks.txt file will also be updated.

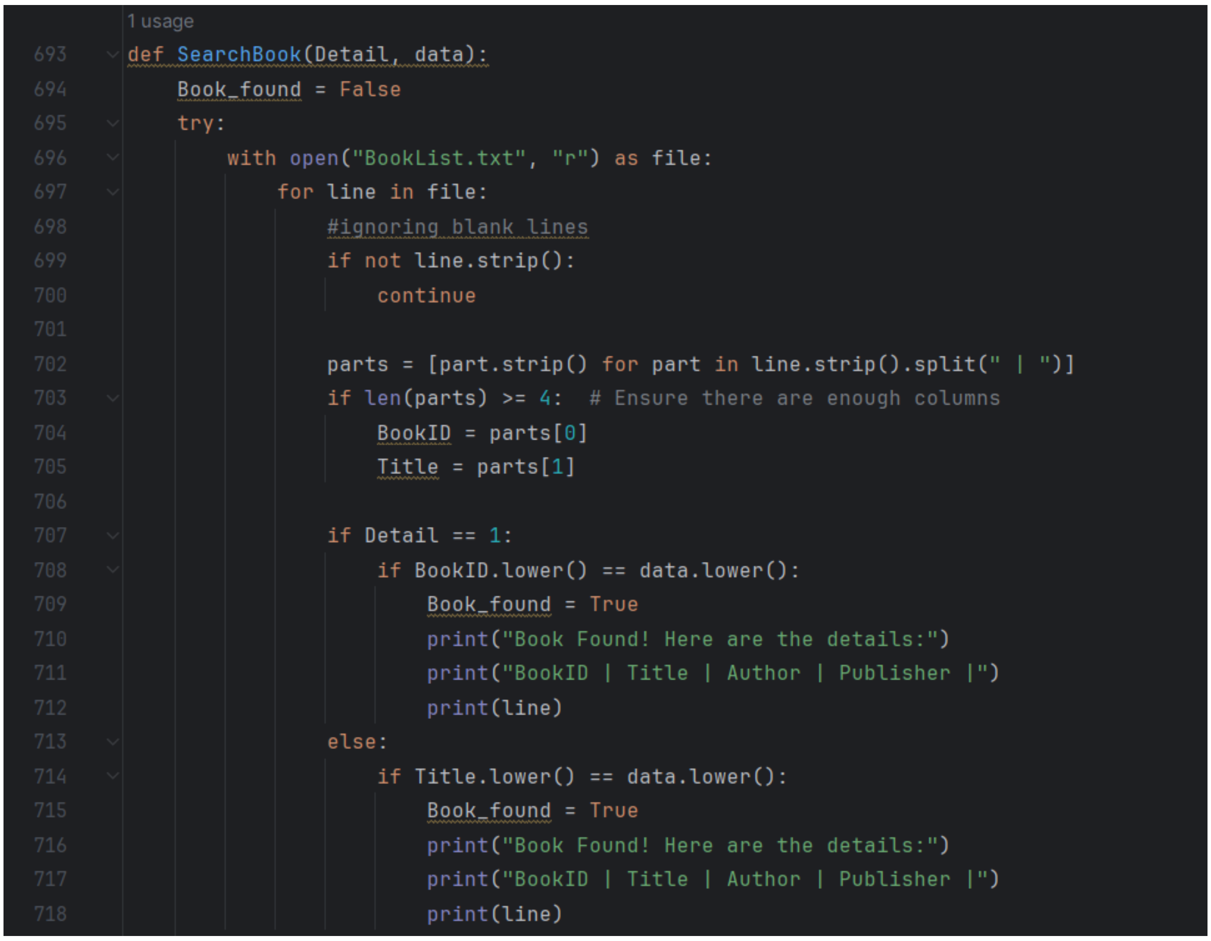
5.1.2.4 View Book List Function



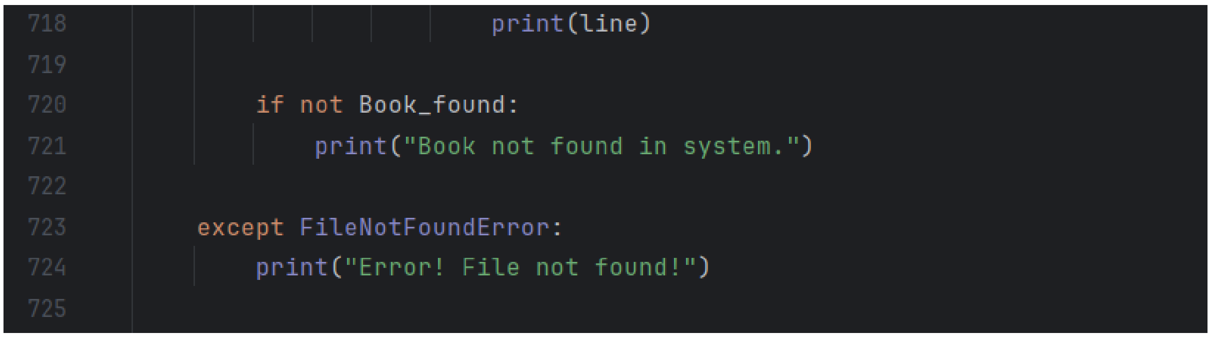
```
1 usage
def View_BookList():
    try:
        with open("BookList.txt", "r") as file:
            print("Here is the list of all books:")
            for line in file:
                print(line)
        with open("availablebooks.txt", "r") as avFile:
            print("Here is the list of all available(not lent) books:")
            for lines in avFile:
                print(lines)

    except FileNotFoundError:
        print("Error! File not found.")

```

This will allow librarian to view all the books and available books in the library

5.1.2.5 Search specific book from book list



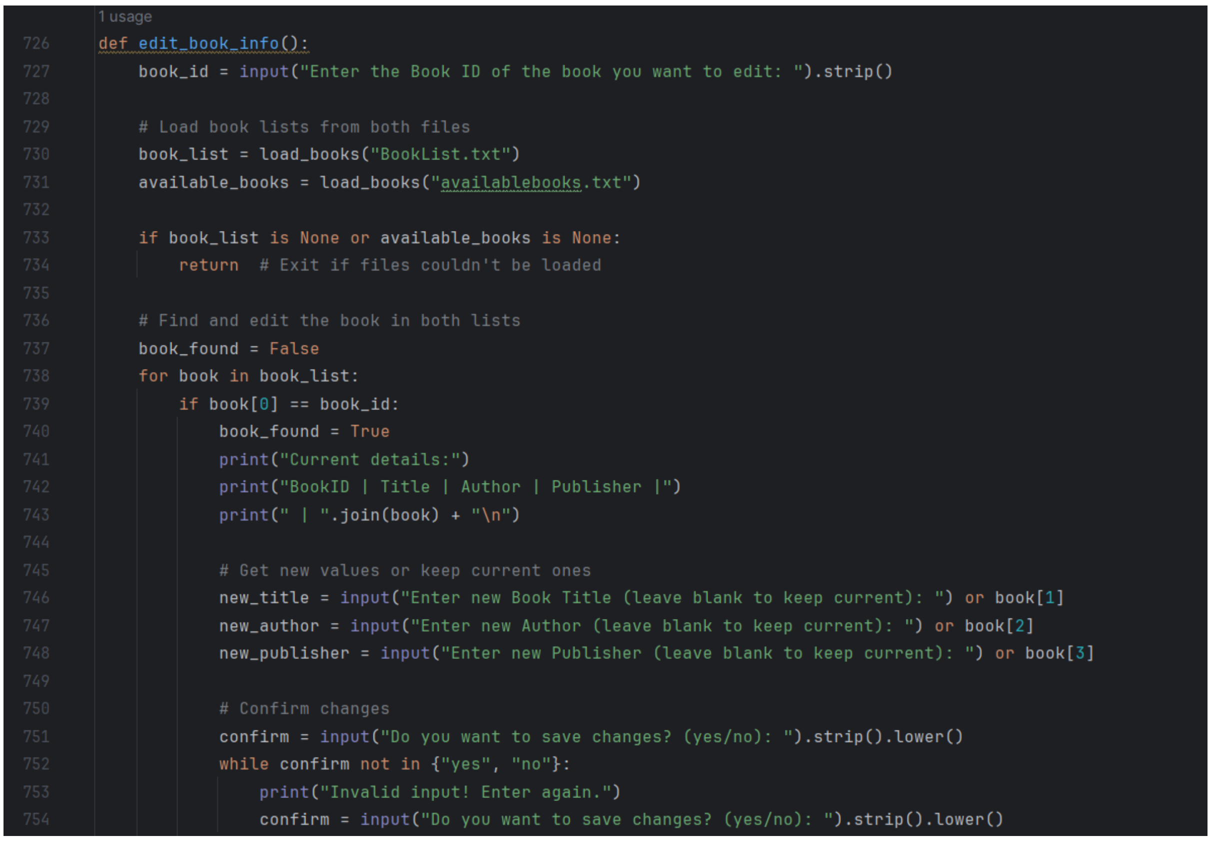
```
1 usage
def SearchBook(Detail, data):
    Book_found = False
    try:
        with open("BookList.txt", "r") as file:
            for line in file:
                #ignoring blank lines
                if not line.strip():
                    continue

                parts = [part.strip() for part in line.strip().split(" | ")]
                if len(parts) >= 4:  # Ensure there are enough columns
                    BookID = parts[0]
                    Title = parts[1]

                if Detail == 1:
                    if BookID.lower() == data.lower():
                        Book_found = True
                        print("Book Found! Here are the details:")
                        print("BookID | Title | Author | Publisher |")
                        print(line)
                else:
                    if Title.lower() == data.lower():
                        Book_found = True
                        print("Book Found! Here are the details:")
                        print("BookID | Title | Author | Publisher |")
                        print(line)
```



```python
                        print(line)

        if not Book_found:
            print("Book not found in system.")

    except FileNotFoundError:
        print("Error! File not found!")

```

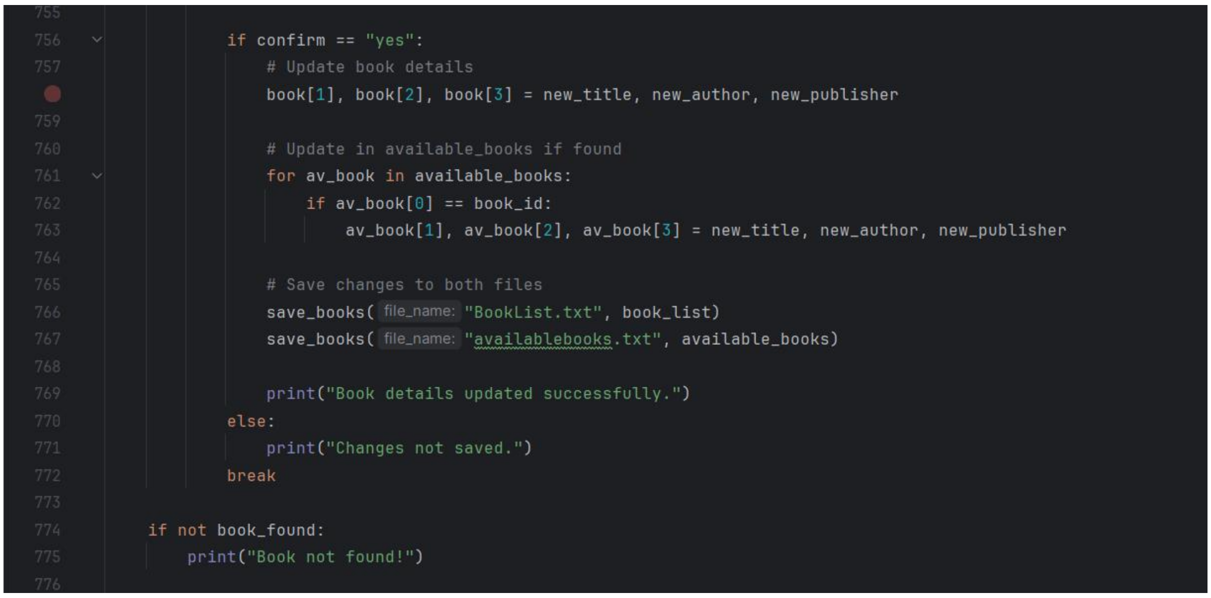
To search for a specific book, the librarian can enter either the Book ID or the Book Title. If book is found in the system, its details are displayed else an error message will be outputted to inform the librarian that the book is not found.

5.1.2.6 Edit book information from book list



1 usage

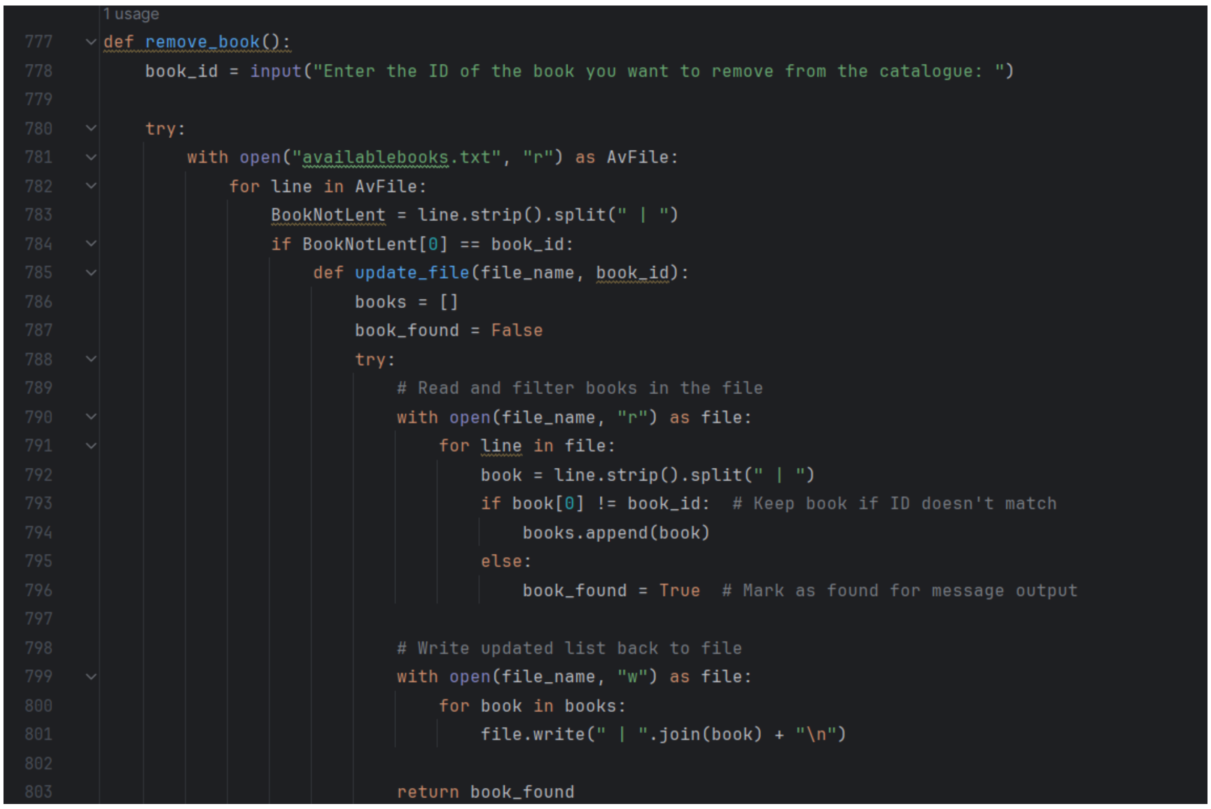
```python
def edit_book_info():
    book_id = input("Enter the Book ID of the book you want to edit: ").strip()

    # Load book lists from both files
    book_list = load_books("BookList.txt")
    available_books = load_books("availablebooks.txt")

    if book_list is None or available_books is None:
        return  # Exit if files couldn't be loaded

    # Find and edit the book in both lists
    book_found = False
    for book in book_list:
        if book[0] == book_id:
            book_found = True
            print("Current details:")
            print("BookID | Title | Author | Publisher |")
            print(" | ".join(book) + "\n")

            # Get new values or keep current ones
            new_title = input("Enter new Book Title (leave blank to keep current): ") or book[1]
            new_author = input("Enter new Author (leave blank to keep current): ") or book[2]
            new_publisher = input("Enter new Publisher (leave blank to keep current): ") or book[3]

            # Confirm changes
            confirm = input("Do you want to save changes? (yes/no): ").strip().lower()
            while confirm not in {"yes", "no"}:
                print("Invalid input! Enter again.")
                confirm = input("Do you want to save changes? (yes/no): ").strip().lower()
```



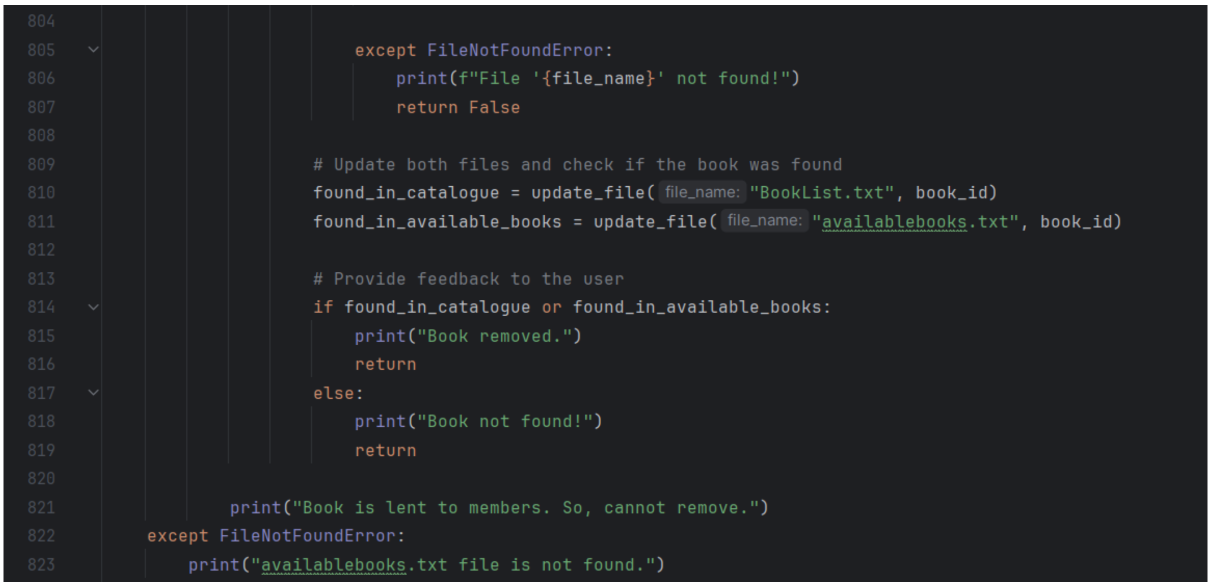
```
            if confirm == "yes":
                # Update book details
                book[1], book[2], book[3] = new_title, new_author, new_publisher

                # Update in available_books if found
                for av_book in available_books:
                    if av_book[0] == book_id:
                        av_book[1], av_book[2], av_book[3] = new_title, new_author, new_publisher

                # Save changes to both files
                save_books( file_name: "BookList.txt", book_list)
                save_books( file_name: "availablebooks.txt", available_books)

                print("Book details updated successfully.")
            else:
                print("Changes not saved.")
            break

    if not book_found:
        print("Book not found!")
```

This function will allow the librarian to edit any book's details and update the BookList.txt and availablebooks.txt file correspondingly. The user can update the title, author, and publisher of the book. If he wishes to edit only one item, it will leave a blank space for the items he does not wish to change.

5.1.2.7 Remove specific book from book list



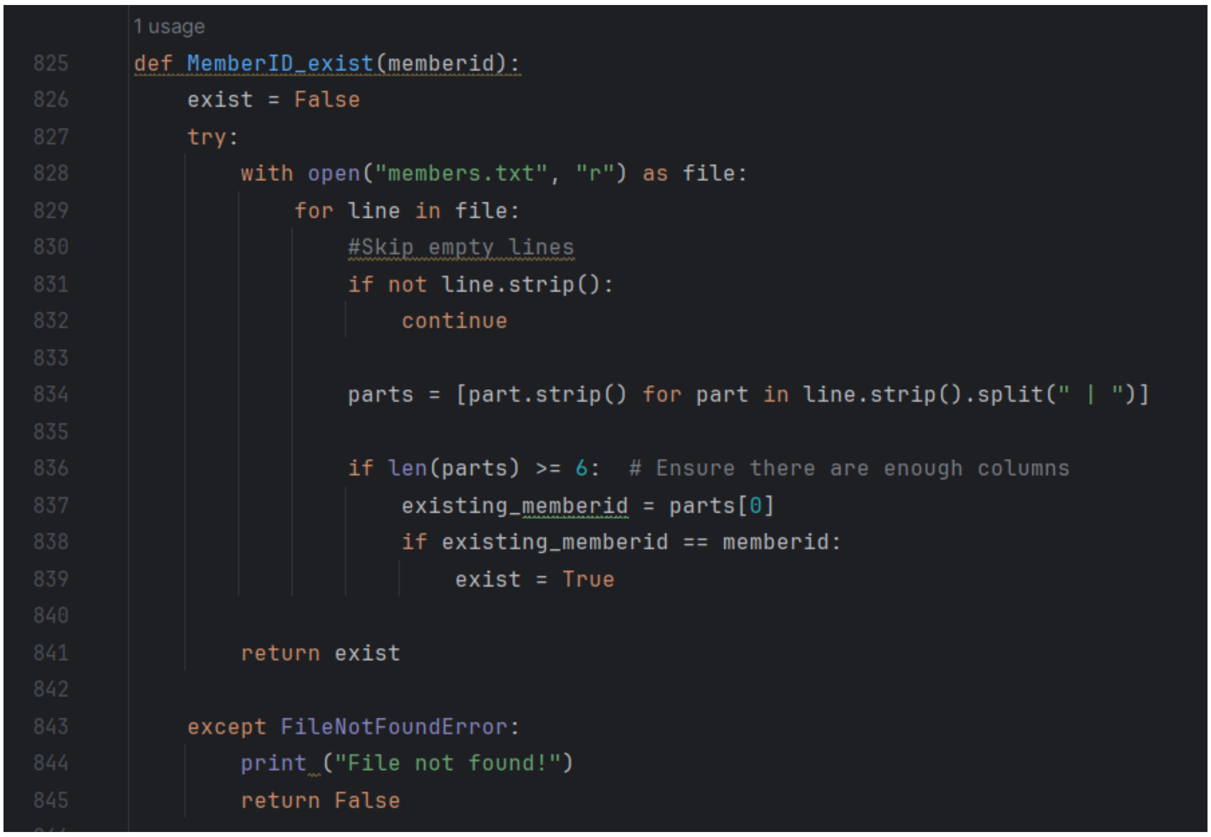
```
1 usage
def remove_book():
    book_id = input("Enter the ID of the book you want to remove from the catalogue: ")

    try:
        with open("availablebooks.txt", "r") as AvFile:
            for line in AvFile:
                BookNotLent = line.strip().split(" | ")
                if BookNotLent[0] == book_id:
                    def update_file(file_name, book_id):
                        books = []
                        book_found = False
                        try:
                            # Read and filter books in the file
                            with open(file_name, "r") as file:
                                for line in file:
                                    book = line.strip().split(" | ")
                                    if book[0] != book_id:  # Keep book if ID doesn't match
                                        books.append(book)
                                    else:
                                        book_found = True  # Mark as found for message output

                            # Write updated list back to file
                            with open(file_name, "w") as file:
                                for book in books:
                                    file.write(" | ".join(book) + "\n")

                            return book_found
```



```python

                        except FileNotFoundError:
                            print(f"File '{file_name}' not found!")
                            return False

                    # Update both files and check if the book was found
                    found_in_catalogue = update_file("BookList.txt", book_id)
                    found_in_available_books = update_file("availablebooks.txt", book_id)

                    # Provide feedback to the user
                    if found_in_catalogue or found_in_available_books:
                        print("Book removed.")
                        return
                    else:
                        print("Book not found!")
                        return

            print("Book is lent to members. So, cannot remove.")
    except FileNotFoundError:
        print("availablebooks.txt file is not found.")
```

The librarian will need to input the Book ID of the book he wishes to remove. If the book is currently borrowed by a member, the book cannot be removed, only books in the availablebooks.txt file can be removed.

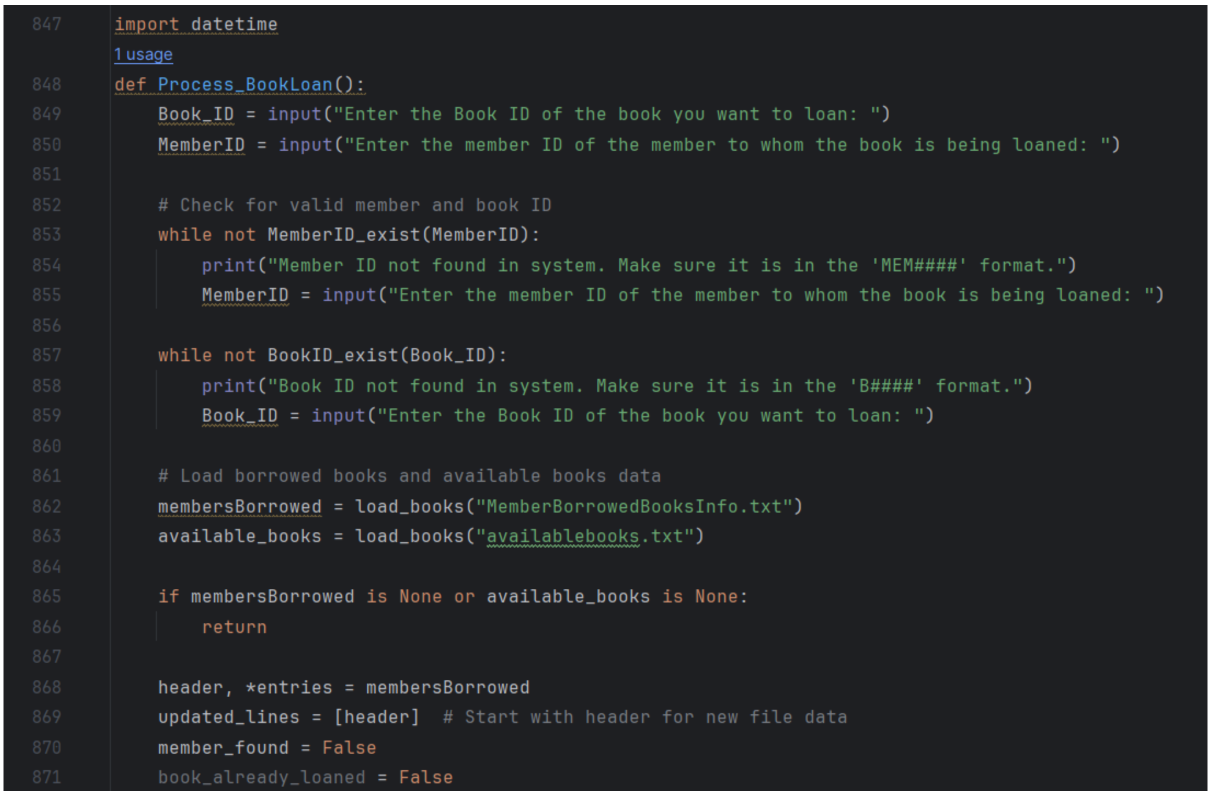
5.1.2.8 Validating Member ID



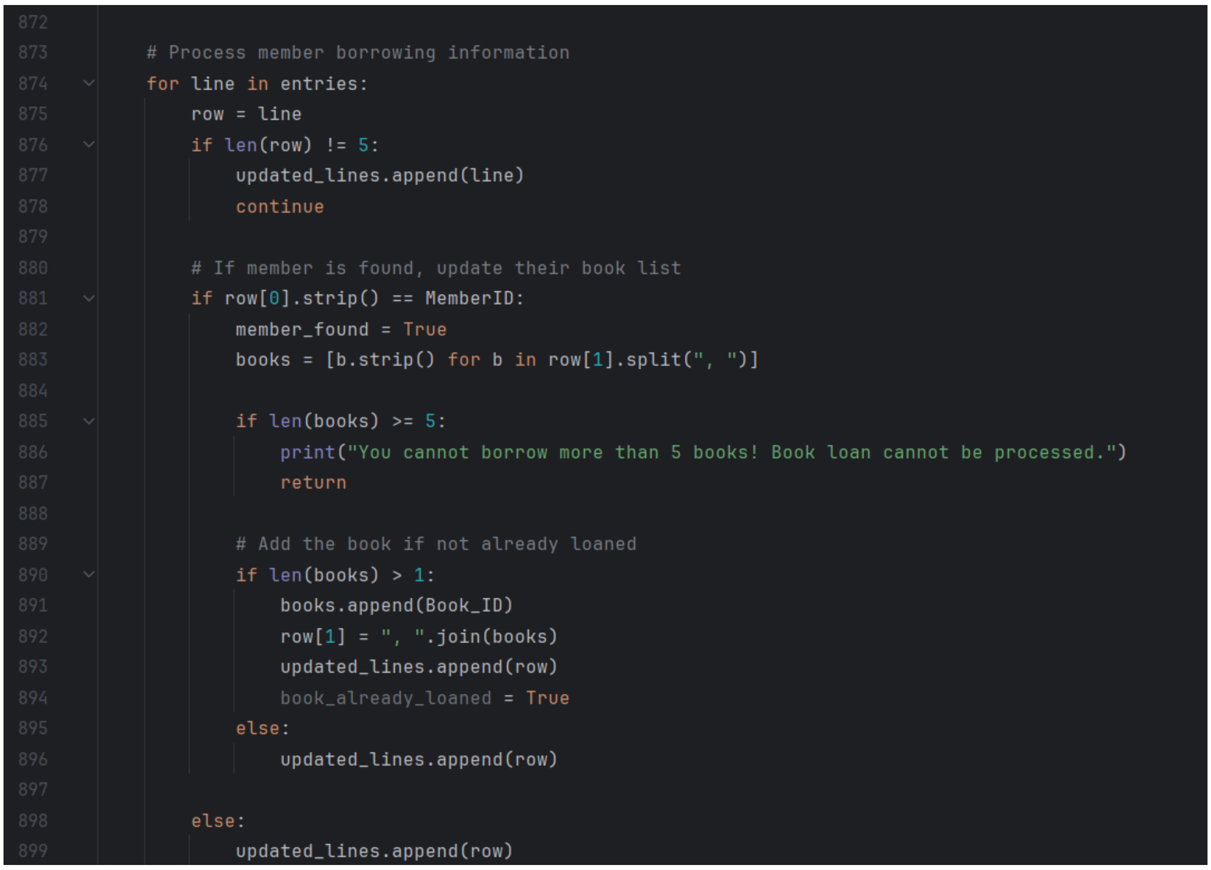
```python
1 usage
def MemberID_exist(memberid):
    exist = False
    try:
        with open("members.txt", "r") as file:
            for line in file:
                #Skip empty lines
                if not line.strip():
                    continue

                parts = [part.strip() for part in line.strip().split(" | ")]

                if len(parts) >= 6:  # Ensure there are enough columns
                    existing_memberid = parts[0]
                    if existing_memberid == memberid:
                        exist = True

        return exist

    except FileNotFoundError:
        print ("File not found!")
        return False
```

This function will ensure that the member ID entered exists in the system. If it does not exist, it will return Boolean value ‘False’.

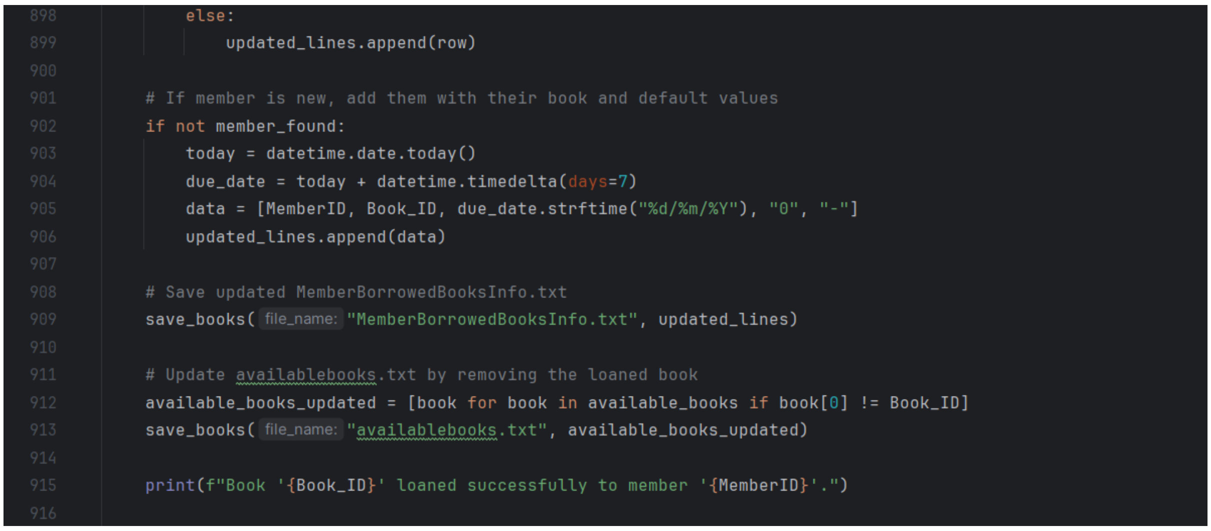
5.1.2.9 Processing a book loan



```python
import datetime
1 usage
def Process_BookLoan():
    Book_ID = input("Enter the Book ID of the book you want to loan: ")
    MemberID = input("Enter the member ID of the member to whom the book is being loaned: ")

    # Check for valid member and book ID
    while not MemberID_exist(MemberID):
        print("Member ID not found in system. Make sure it is in the 'MEM####' format.")
        MemberID = input("Enter the member ID of the member to whom the book is being loaned: ")

    while not BookID_exist(Book_ID):
        print("Book ID not found in system. Make sure it is in the 'B####' format.")
        Book_ID = input("Enter the Book ID of the book you want to loan: ")

    # Load borrowed books and available books data
    membersBorrowed = load_books("MemberBorrowedBooksInfo.txt")
    available_books = load_books("availablebooks.txt")

    if membersBorrowed is None or available_books is None:
        return

    header, *entries = membersBorrowed
    updated_lines = [header]  # Start with header for new file data
    member_found = False
    book_already_loaned = False
```



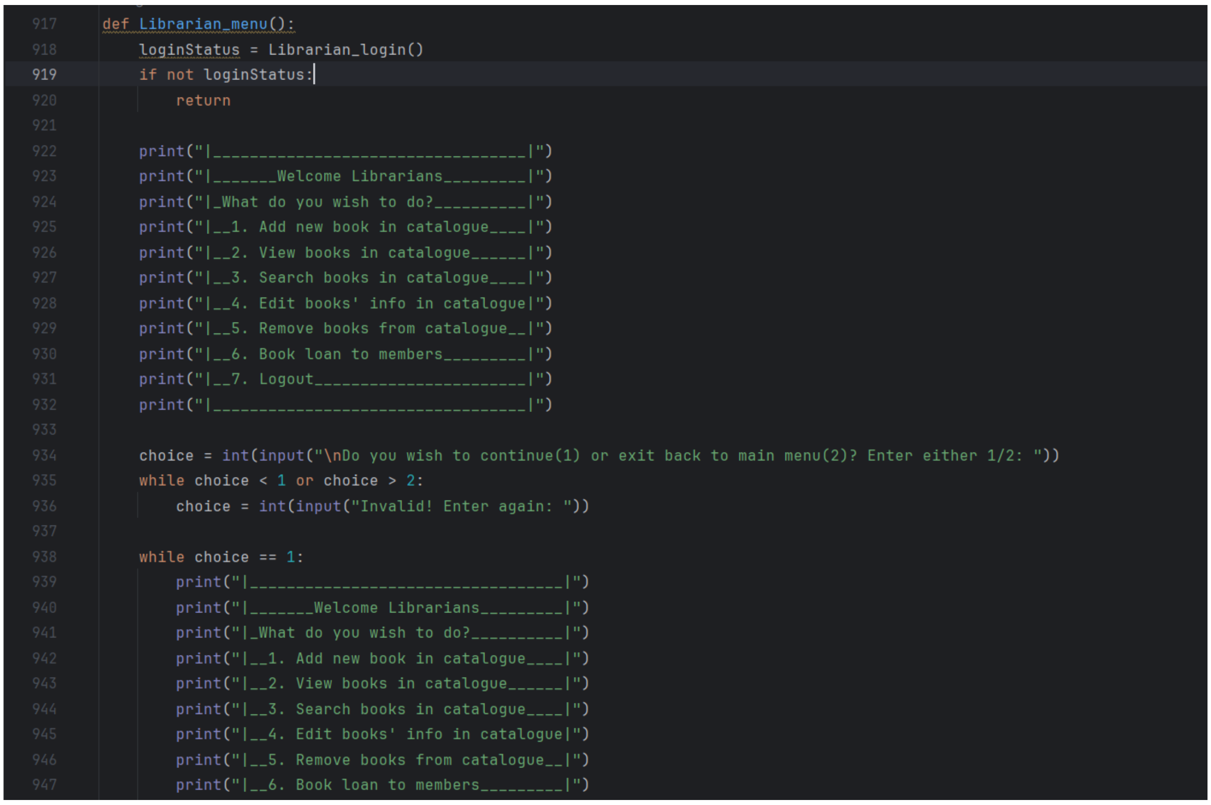
```python

    # Process member borrowing information
    for line in entries:
        row = line
        if len(row) != 5:
            updated_lines.append(line)
            continue

        # If member is found, update their book list
        if row[0].strip() == MemberID:
            member_found = True
            books = [b.strip() for b in row[1].split(", ")]

            if len(books) >= 5:
                print("You cannot borrow more than 5 books! Book loan cannot be processed.")
                return

            # Add the book if not already loaned
            if len(books) > 1:
                books.append(Book_ID)
                row[1] = ", ".join(books)
                updated_lines.append(row)
                book_already_loaned = True
            else:
                updated_lines.append(row)

        else:
            updated_lines.append(row)
```



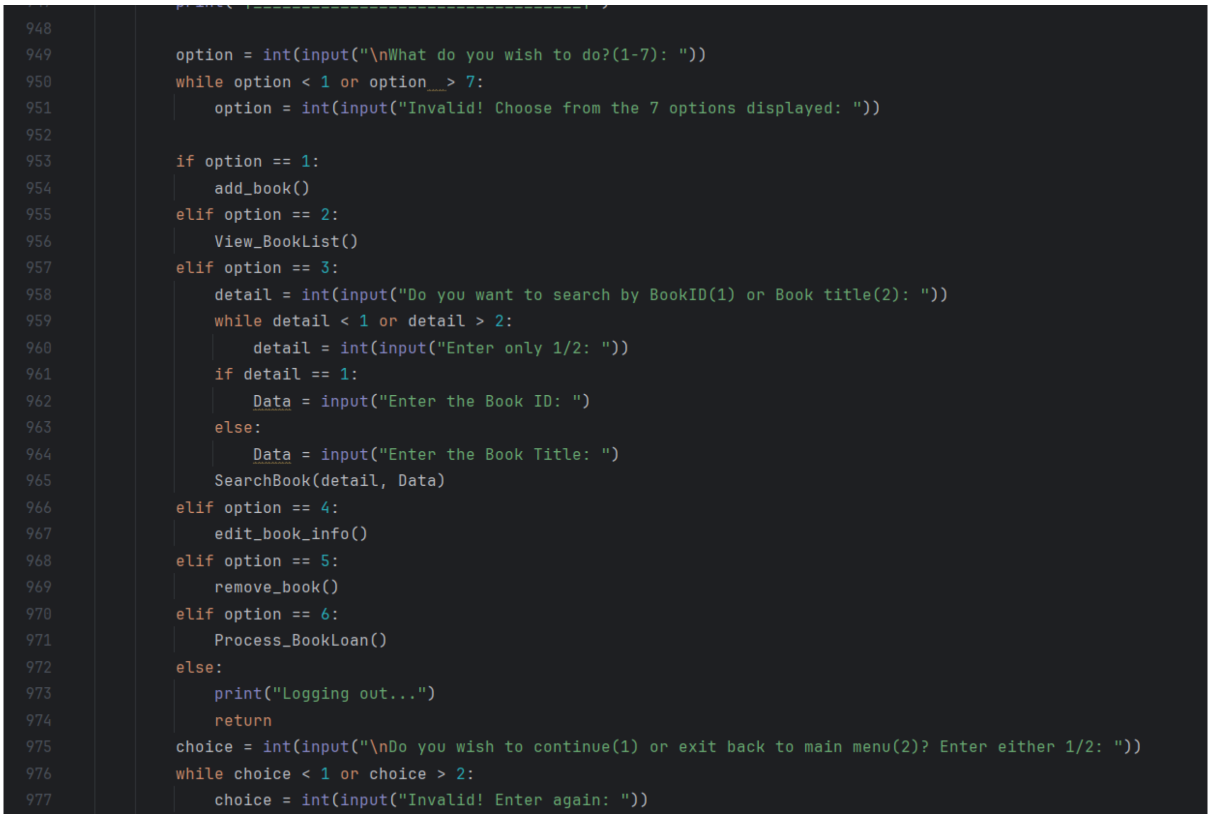
```python
        else:
            updated_lines.append(row)

    # If member is new, add them with their book and default values
    if not member_found:
        today = datetime.date.today()
        due_date = today + datetime.timedelta(days=7)
        data = [MemberID, Book_ID, due_date.strftime("%d/%m/%Y"), "0", "-"]
        updated_lines.append(data)

    # Save updated MemberBorrowedBooksInfo.txt
    save_books( file_name: "MemberBorrowedBooksInfo.txt", updated_lines)

    # Update availablebooks.txt by removing the loaned book
    available_books_updated = [book for book in available_books if book[0] != Book_ID]
    save_books( file_name: "availablebooks.txt", available_books_updated)

    print(f"Book '{Book_ID}' loaned successfully to member '{MemberID}'.")
```

The librarian will need to input the Book ID and the Member ID, and the system will verify if they exist in the system. If not, it will output an error message until the librarian inputs valid IDs. If the member has already borrowed 5 books, the loaning process cannot be fulfilled. If all the criteria to loan a book are met, the function will execute successfully and the Due Date will be set to be a week from the current date, fees will be set as 0, and payment status as '- '. MembersBorrowedBooksInfo.txt file and availablebooks.txt file is updated accordingly.

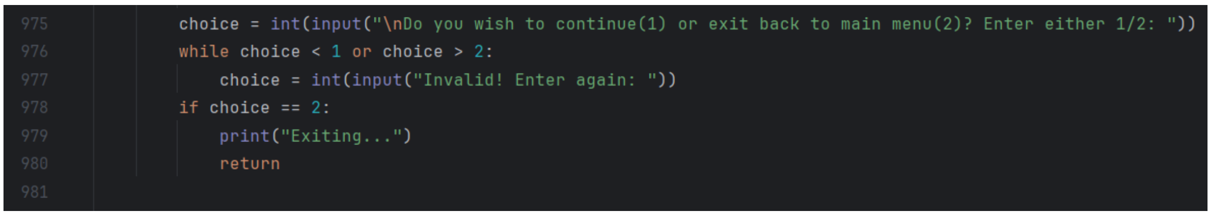
5.1.2.10 Librarians' menu



```python
def Librarian_menu():
    loginStatus = Librarian_login()
    if not loginStatus:
        return

    print("|__________________________________|")
    print("|_______Welcome Librarians_________|")
    print("|_What do you wish to do?__________|")
    print("|__1. Add new book in catalogue____|")
    print("|__2. View books in catalogue______|")
    print("|__3. Search books in catalogue____|")
    print("|__4. Edit books' info in catalogue|")
    print("|__5. Remove books from catalogue__|")
    print("|__6. Book loan to members_________|")
    print("|__7. Logout_______________________|")
    print("|__________________________________|")

    choice = int(input("\nDo you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: "))
    while choice < 1 or choice > 2:
        choice = int(input("Invalid! Enter again: "))

    while choice == 1:
        print("|__________________________________|")
        print("|_______Welcome Librarians_________|")
        print("|_What do you wish to do?__________|")
        print("|__1. Add new book in catalogue____|")
        print("|__2. View books in catalogue______|")
        print("|__3. Search books in catalogue____|")
        print("|__4. Edit books' info in catalogue|")
        print("|__5. Remove books from catalogue__|")
        print("|__6. Book loan to members_________|")
```



```python
        option = int(input("\nWhat do you wish to do?(1-7): "))
        while option < 1 or option  > 7:
            option = int(input("Invalid! Choose from the 7 options displayed: "))

        if option == 1:
            add_book()
        elif option == 2:
            View_BookList()
        elif option == 3:
            detail = int(input("Do you want to search by BookID(1) or Book title(2): "))
            while detail < 1 or detail > 2:
                detail = int(input("Enter only 1/2: "))
            if detail == 1:
                Data = input("Enter the Book ID: ")
            else:
                Data = input("Enter the Book Title: ")
            SearchBook(detail, Data)
        elif option == 4:
            edit_book_info()
        elif option == 5:
            remove_book()
        elif option == 6:
            Process_BookLoan()
        else:
            print("Logging out...")
            return
        choice = int(input("\nDo you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: "))
        while choice < 1 or choice > 2:
            choice = int(input("Invalid! Enter again: "))
```

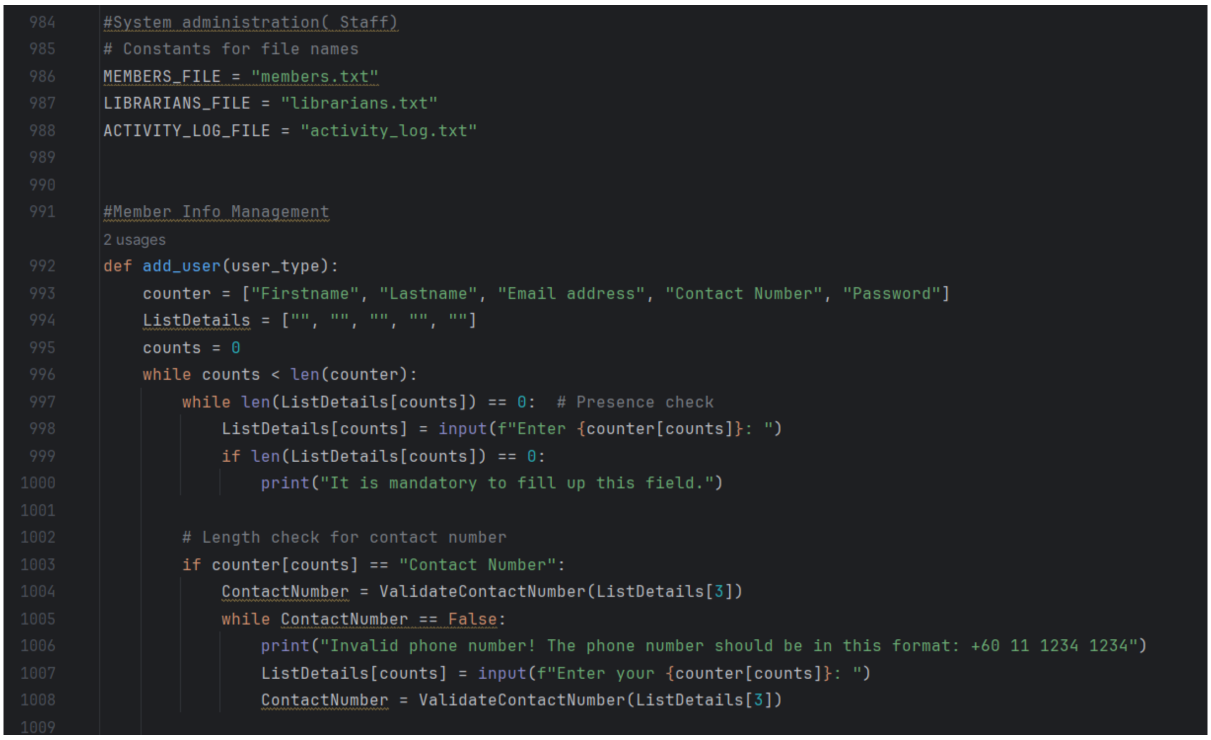


```python
        choice = int(input("\nDo you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: "))
        while choice < 1 or choice > 2:
            choice = int(input("Invalid! Enter again: "))
        if choice == 2:
            print("Exiting...")
            return
```

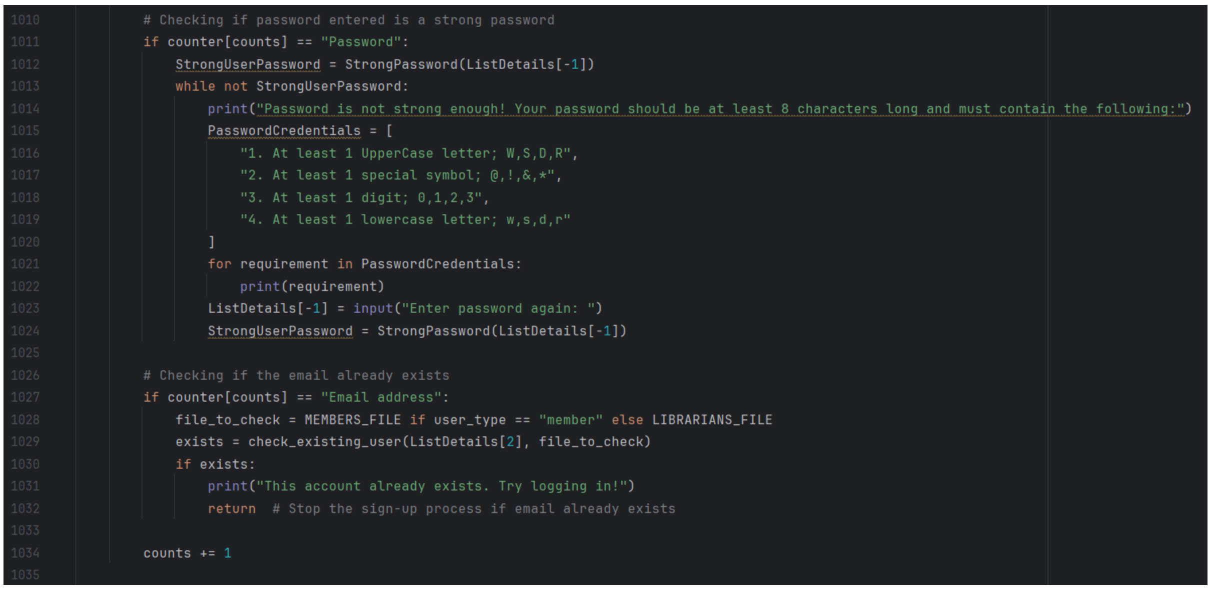
This function will allow the librarian to login, sign up, add a new book, edit book information, search for a book, remove a book from an existing file, view all the books, loan a book, and logout if his task is done. This process is continuous unless the librarian chooses to exit.

5.1.3 System Administrator User

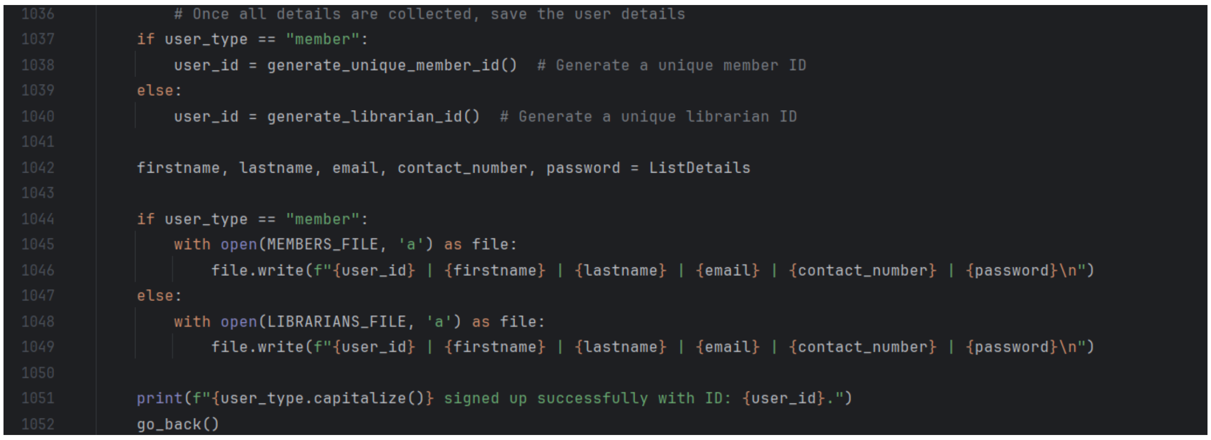
5.1.3.1 Add new librarian or new member to existing list



```python
#System administration( Staff)
# Constants for file names
MEMBERS_FILE = "members.txt"
LIBRARIANS_FILE = "librarians.txt"
ACTIVITY_LOG_FILE = "activity_log.txt"


#Member Info Management
2 usages
def add_user(user_type):
    counter = ["Firstname", "Lastname", "Email address", "Contact Number", "Password"]
    ListDetails = ["", "", "", "", ""]
    counts = 0
    while counts < len(counter):
        while len(ListDetails[counts]) == 0:  # Presence check
            ListDetails[counts] = input(f"Enter {counter[counts]}: ")
            if len(ListDetails[counts]) == 0:
                print("It is mandatory to fill up this field.")

        # Length check for contact number
        if counter[counts] == "Contact Number":
            ContactNumber = ValidateContactNumber(ListDetails[3])
            while ContactNumber == False:
                print("Invalid phone number! The phone number should be in this format: +60 11 1234 1234")
                ListDetails[counts] = input(f"Enter your {counter[counts]}: ")
                ContactNumber = ValidateContactNumber(ListDetails[3])

```



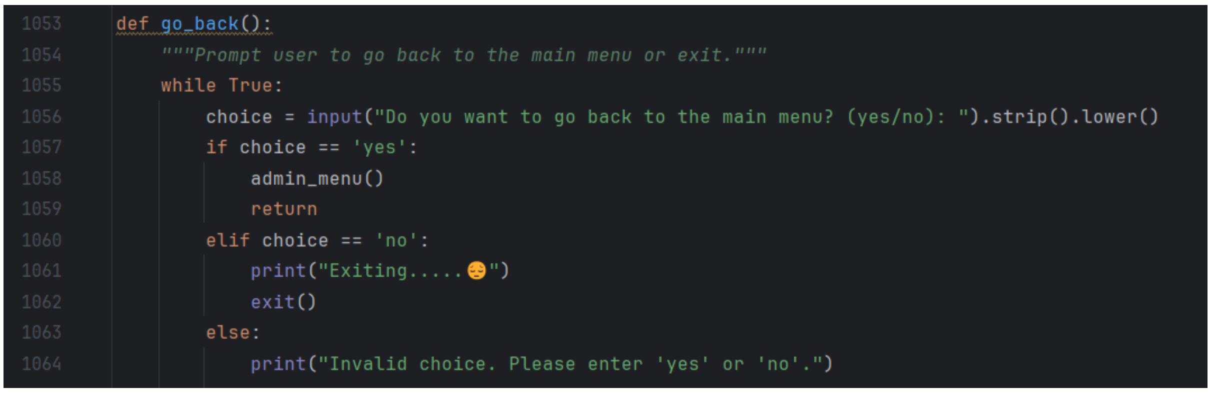
```python
        # Checking if password entered is a strong password
        if counter[counts] == "Password":
            StrongUserPassword = StrongPassword(ListDetails[-1])
            while not StrongUserPassword:
                print("Password is not strong enough! Your password should be at least 8 characters long and must contain the following:")
                PasswordCredentials = [
                    "1. At least 1 UpperCase letter; W,S,D,R",
                    "2. At least 1 special symbol; @,!,&,*",
                    "3. At least 1 digit; 0,1,2,3",
                    "4. At least 1 lowercase letter; w,s,d,r"
                ]
                for requirement in PasswordCredentials:
                    print(requirement)
                ListDetails[-1] = input("Enter password again: ")
                StrongUserPassword = StrongPassword(ListDetails[-1])

        # Checking if the email already exists
        if counter[counts] == "Email address":
            file_to_check = MEMBERS_FILE if user_type == "member" else LIBRARIANS_FILE
            exists = check_existing_user(ListDetails[2], file_to_check)
            if exists:
                print("This account already exists. Try logging in!")
                return  # Stop the sign-up process if email already exists

        counts += 1

```



```python
        # Once all details are collected, save the user details
    if user_type == "member":
        user_id = generate_unique_member_id()  # Generate a unique member ID
    else:
        user_id = generate_librarian_id()  # Generate a unique librarian ID

    firstname, lastname, email, contact_number, password = ListDetails

    if user_type == "member":
        with open(MEMBERS_FILE, 'a') as file:
            file.write(f"{user_id} | {firstname} | {lastname} | {email} | {contact_number} | {password}\n")
    else:
        with open(LIBRARIANS_FILE, 'a') as file:
            file.write(f"{user_id} | {firstname} | {lastname} | {email} | {contact_number} | {password}\n")

    print(f"{user_type.capitalize()} signed up successfully with ID: {user_id}.")
    go_back()
```

This function will allow the admin to add new librarian or new member (user type). All the details are validated before updating the file.

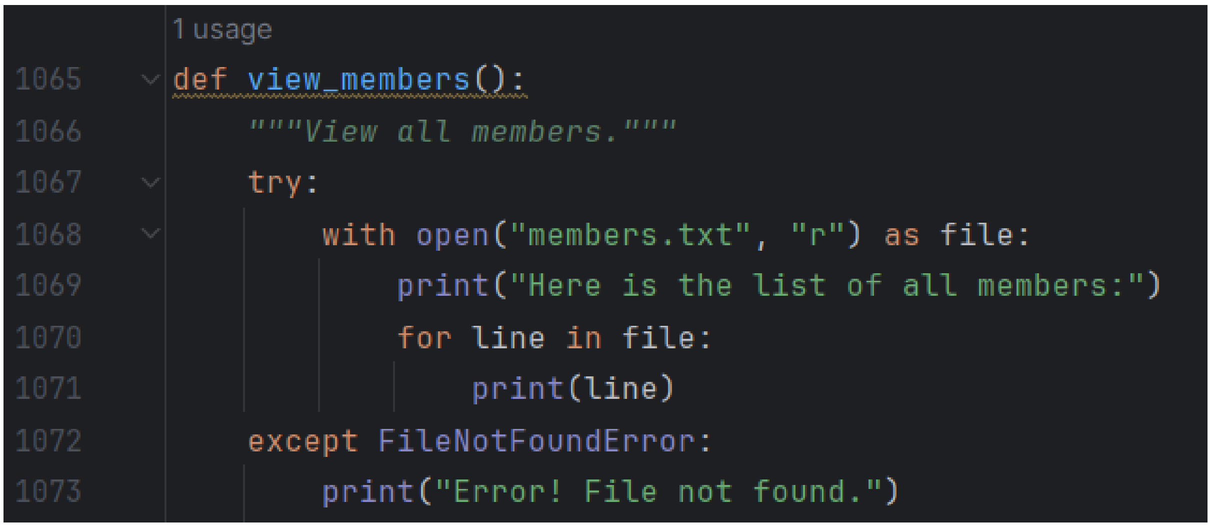
5.1.3.2 Go Back Function



```python
def go_back():
    """Prompt user to go back to the main menu or exit."""
    while True:
        choice = input("Do you want to go back to the main menu? (yes/no): ").strip().lower()
        if choice == 'yes':
            admin_menu()
            return
        elif choice == 'no':
            print("Exiting.....😔")
            exit()
        else:
            print("Invalid choice. Please enter 'yes' or 'no'.")
```

This function will prompt the user to go back to the main menu or exit the system.

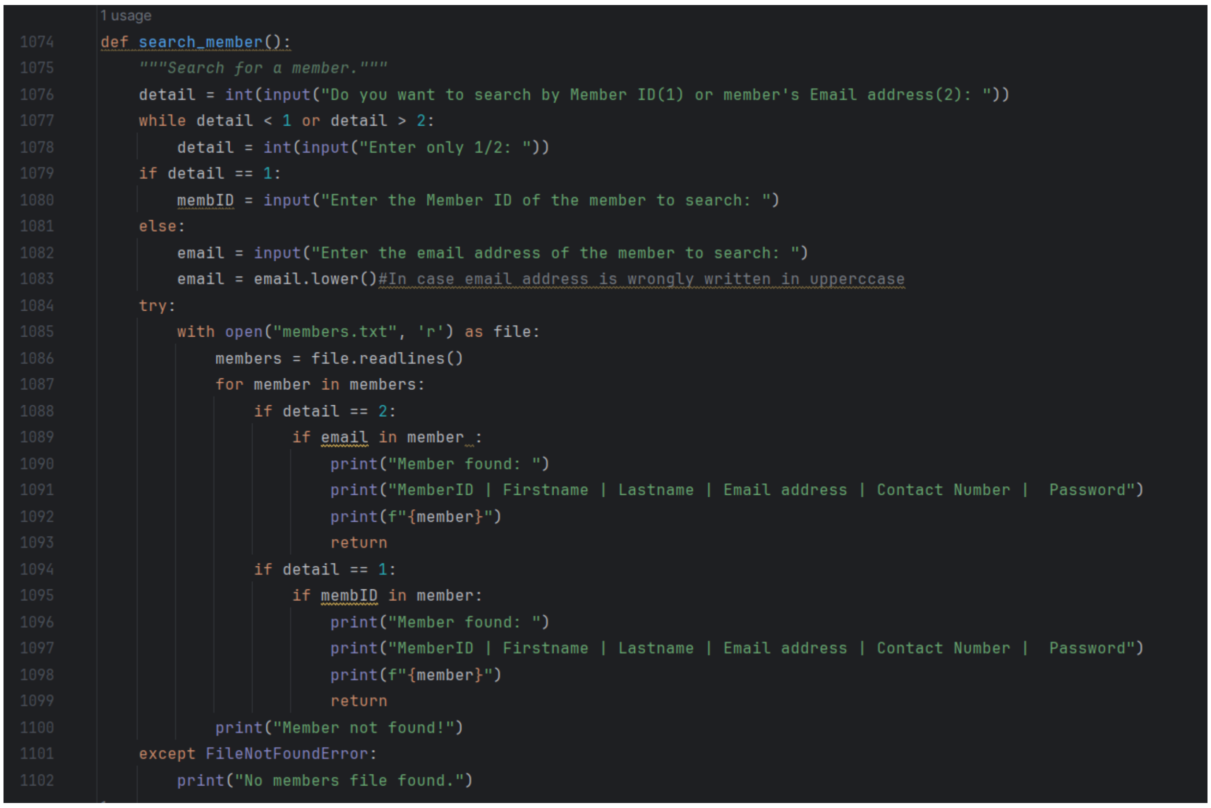
5.1.3.3 Display members' list



```python
1 usage
def view_members():
    """View all members."""
    try:
        with open("members.txt", "r") as file:
            print("Here is the list of all members:")
            for line in file:
                print(line)
    except FileNotFoundError:
        print("Error! File not found.")
```

This function will allow the administrator to view all the members, like previous functions.

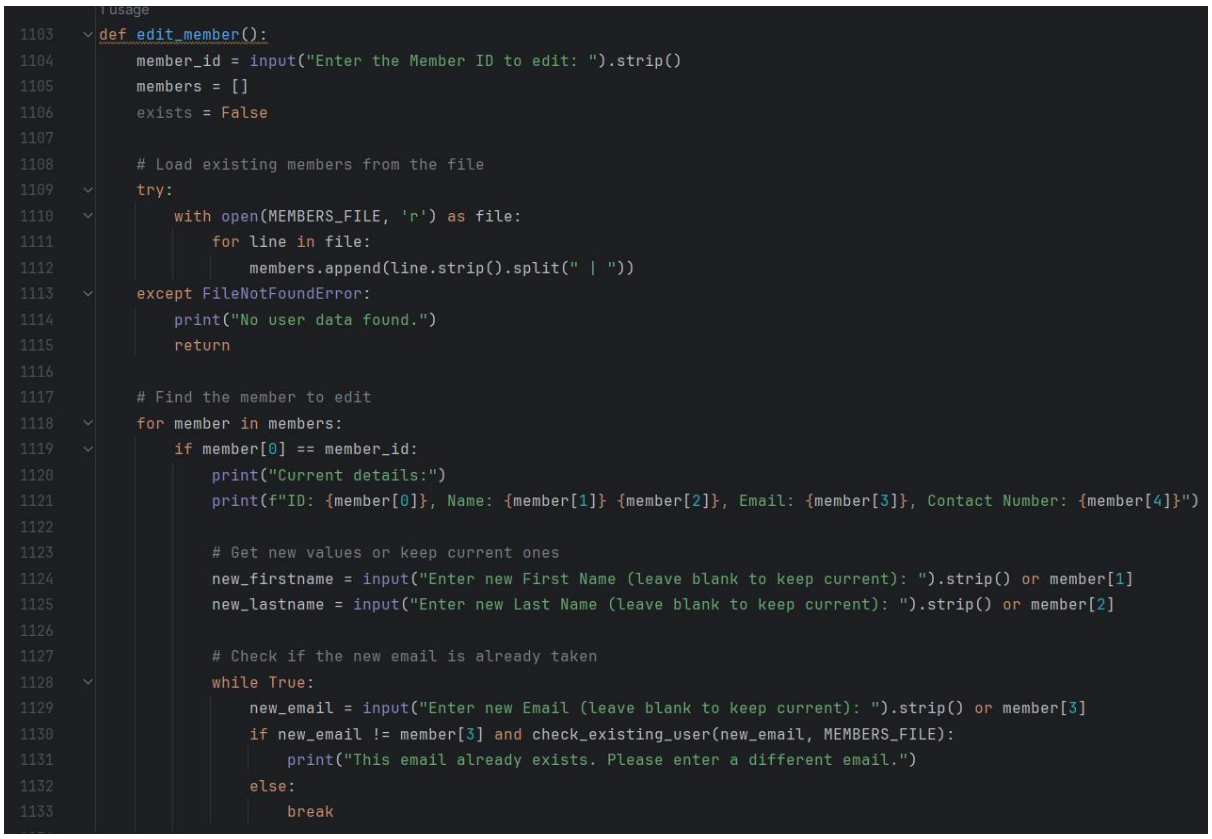
5.1.3.4 Search specific member from existing list



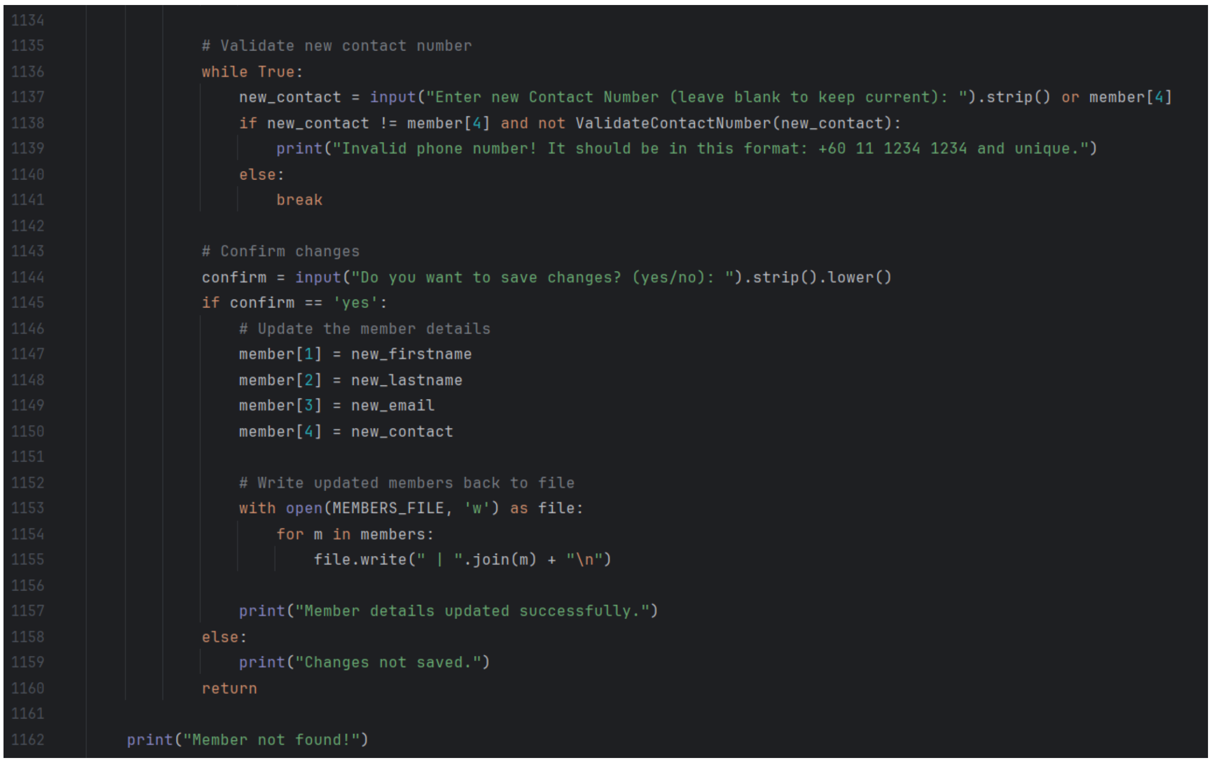
```
1 usage
def search_member():
    """Search for a member."""
    detail = int(input("Do you want to search by Member ID(1) or member's Email address(2): "))
    while detail < 1 or detail > 2:
        detail = int(input("Enter only 1/2: "))
    if detail == 1:
        membID = input("Enter the Member ID of the member to search: ")
    else:
        email = input("Enter the email address of the member to search: ")
        email = email.lower()#In case email address is wrongly written in upperccase
    try:
        with open("members.txt", 'r') as file:
            members = file.readlines()
            for member in members:
                if detail == 2:
                    if email in member :
                        print("Member found: ")
                        print("MemberID | Firstname | Lastname | Email address | Contact Number |  Password")
                        print(f"{member}")
                        return
                if detail == 1:
                    if membID in member:
                        print("Member found: ")
                        print("MemberID | Firstname | Lastname | Email address | Contact Number |  Password")
                        print(f"{member}")
                        return
            print("Member not found!")
    except FileNotFoundError:
        print("No members file found.")
```

This function will allow the administrator to search for a specific member, similar to previous functions

5.1.3.5 Edit specific member's information from existing list



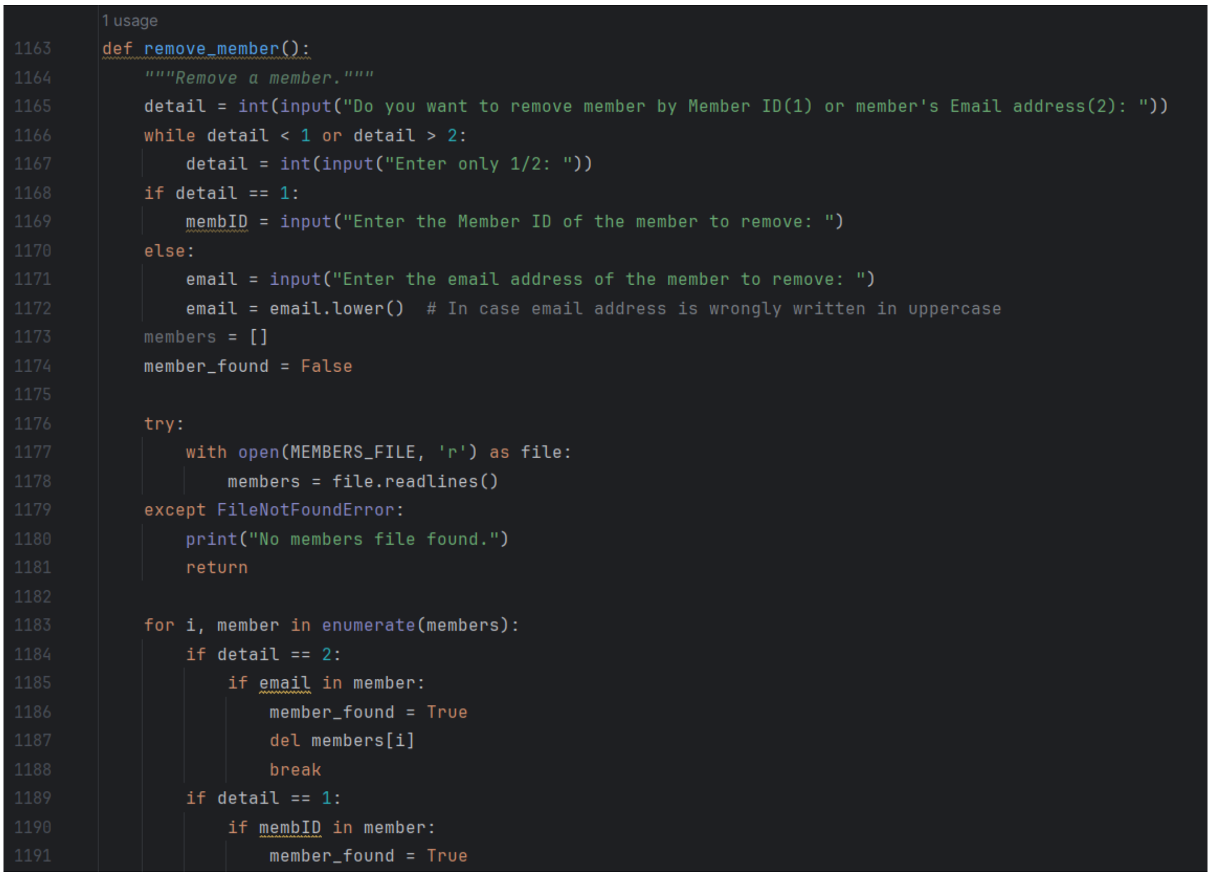
```python
def edit_member():
    member_id = input("Enter the Member ID to edit: ").strip()
    members = []
    exists = False

    # Load existing members from the file
    try:
        with open(MEMBERS_FILE, 'r') as file:
            for line in file:
                members.append(line.strip().split(" | "))
    except FileNotFoundError:
        print("No user data found.")
        return

    # Find the member to edit
    for member in members:
        if member[0] == member_id:
            print("Current details:")
            print(f"ID: {member[0]}, Name: {member[1]} {member[2]}, Email: {member[3]}, Contact Number: {member[4]}")

            # Get new values or keep current ones
            new_firstname = input("Enter new First Name (leave blank to keep current): ").strip() or member[1]
            new_lastname = input("Enter new Last Name (leave blank to keep current): ").strip() or member[2]

            # Check if the new email is already taken
            while True:
                new_email = input("Enter new Email (leave blank to keep current): ").strip() or member[3]
                if new_email != member[3] and check_existing_user(new_email, MEMBERS_FILE):
                    print("This email already exists. Please enter a different email.")
                else:
                    break
```



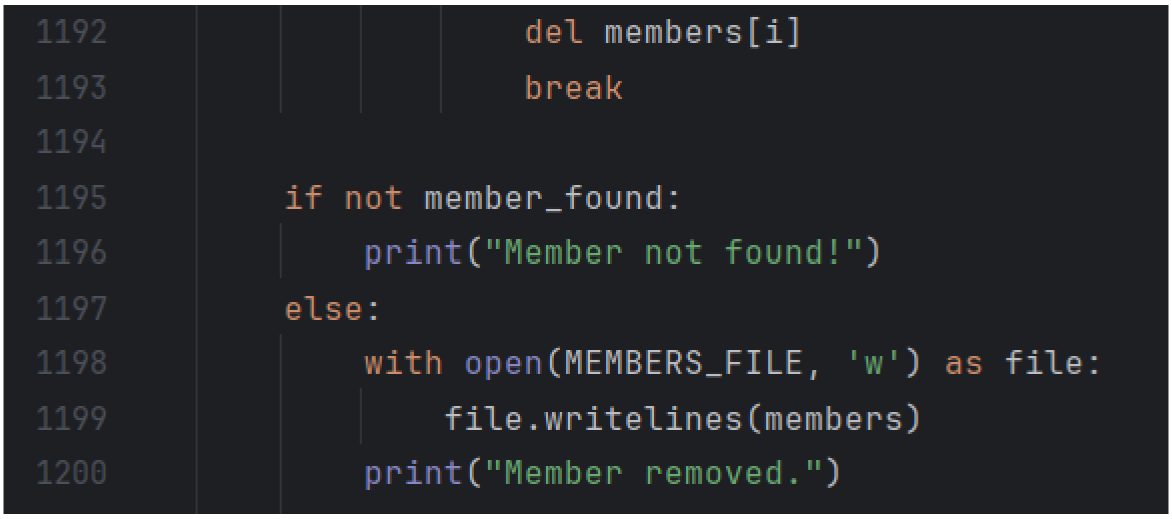
```python

            # Validate new contact number
            while True:
                new_contact = input("Enter new Contact Number (leave blank to keep current): ").strip() or member[4]
                if new_contact != member[4] and not ValidateContactNumber(new_contact):
                    print("Invalid phone number! It should be in this format: +60 11 1234 1234 and unique.")
                else:
                    break

            # Confirm changes
            confirm = input("Do you want to save changes? (yes/no): ").strip().lower()
            if confirm == 'yes':
                # Update the member details
                member[1] = new_firstname
                member[2] = new_lastname
                member[3] = new_email
                member[4] = new_contact

                # Write updated members back to file
                with open(MEMBERS_FILE, 'w') as file:
                    for m in members:
                        file.write(" | ".join(m) + "\n")

                print("Member details updated successfully.")
            else:
                print("Changes not saved.")
            return

    print("Member not found!")
```

This function will allow the administrator to edit specific members' information, similar to previous functions

5.1.3.6 Remove specific member from existing list



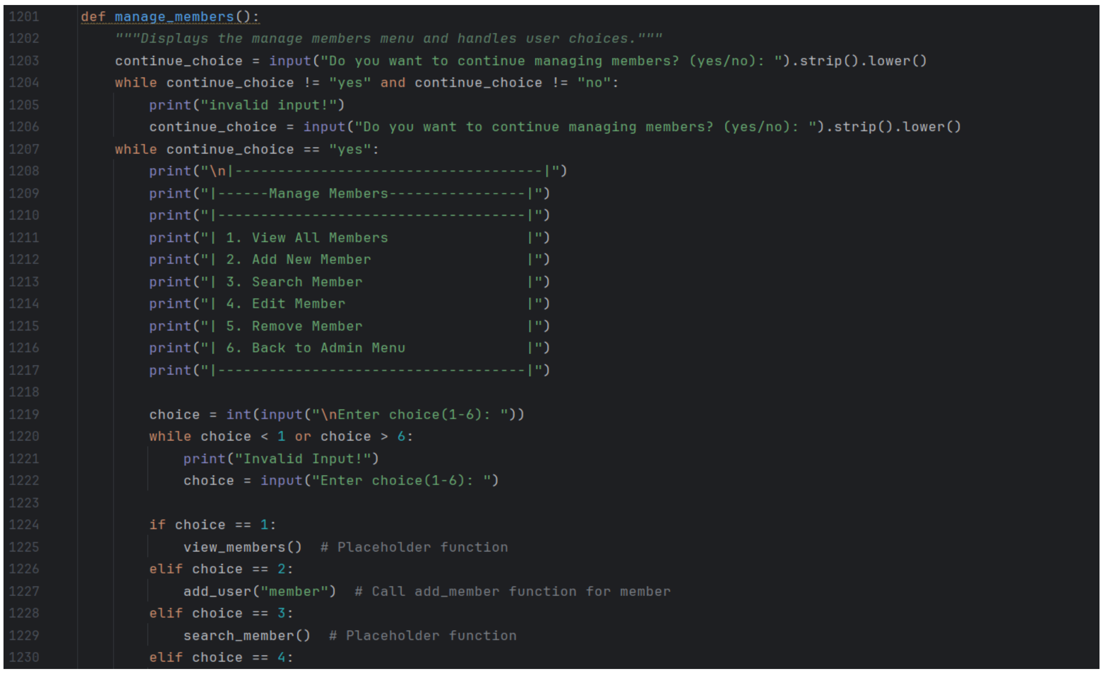
```
1 usage
def remove_member():
    """Remove a member."""
    detail = int(input("Do you want to remove member by Member ID(1) or member's Email address(2): "))
    while detail < 1 or detail > 2:
        detail = int(input("Enter only 1/2: "))
    if detail == 1:
        membID = input("Enter the Member ID of the member to remove: ")
    else:
        email = input("Enter the email address of the member to remove: ")
        email = email.lower()  # In case email address is wrongly written in uppercase
    members = []
    member_found = False

    try:
        with open(MEMBERS_FILE, 'r') as file:
            members = file.readlines()
    except FileNotFoundError:
        print("No members file found.")
        return

    for i, member in enumerate(members):
        if detail == 2:
            if email in member:
                member_found = True
                del members[i]
                break
        if detail == 1:
            if membID in member:
                member_found = True
```



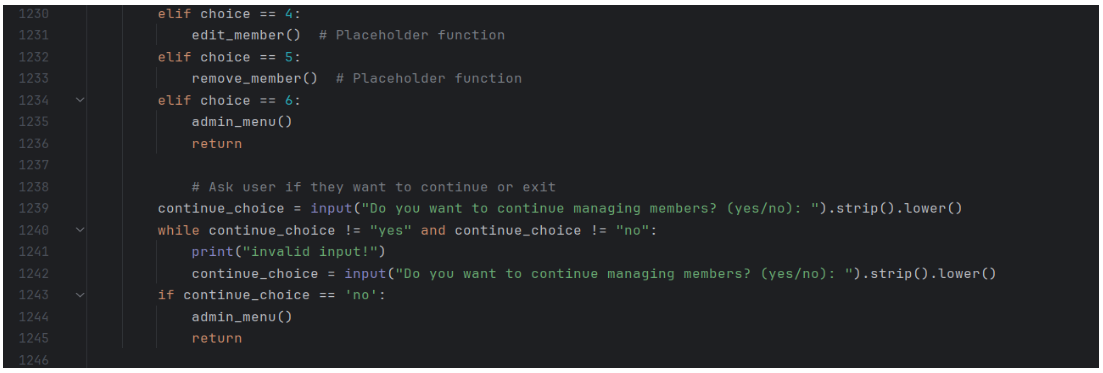
```
                del members[i]
                break

    if not member_found:
        print("Member not found!")
    else:
        with open(MEMBERS_FILE, 'w') as file:
            file.writelines(members)
        print("Member removed.")
```

This function will allow the administrator to remove specific members, similar to previous functions.

5.1.3.7 Managing members Function (Menu to manage members)



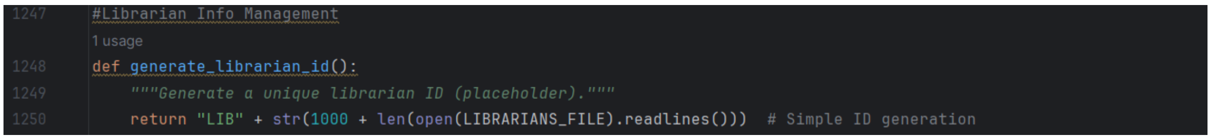
```python
def manage_members():
    """Displays the manage members menu and handles user choices."""
    continue_choice = input("Do you want to continue managing members? (yes/no): ").strip().lower()
    while continue_choice != "yes" and continue_choice != "no":
        print("invalid input!")
        continue_choice = input("Do you want to continue managing members? (yes/no): ").strip().lower()
    while continue_choice == "yes":
        print("\n|------------------------------------|")
        print("|------Manage Members---------------|")
        print("|------------------------------------|")
        print("| 1. View All Members                |")
        print("| 2. Add New Member                  |")
        print("| 3. Search Member                   |")
        print("| 4. Edit Member                     |")
        print("| 5. Remove Member                   |")
        print("| 6. Back to Admin Menu              |")
        print("|------------------------------------|")

        choice = int(input("\nEnter choice(1-6): "))
        while choice < 1 or choice > 6:
            print("Invalid Input!")
            choice = input("Enter choice(1-6): ")

        if choice == 1:
            view_members()  # Placeholder function
        elif choice == 2:
            add_user("member")  # Call add_member function for member
        elif choice == 3:
            search_member()  # Placeholder function
        elif choice == 4:
```



```python
        elif choice == 4:
            edit_member()  # Placeholder function
        elif choice == 5:
            remove_member()  # Placeholder function
        elif choice == 6:
            admin_menu()
            return

            # Ask user if they want to continue or exit
        continue_choice = input("Do you want to continue managing members? (yes/no): ").strip().lower()
        while continue_choice != "yes" and continue_choice != "no":
            print("invalid input!")
            continue_choice = input("Do you want to continue managing members? (yes/no): ").strip().lower()
        if continue_choice == 'no':
            admin_menu()
            return
```

This function will allow the admin to view all the members, add new members, search for a specific member, edit members' information, remove members and return to admin menu. The function is continuous until the administrator wishes to leave.

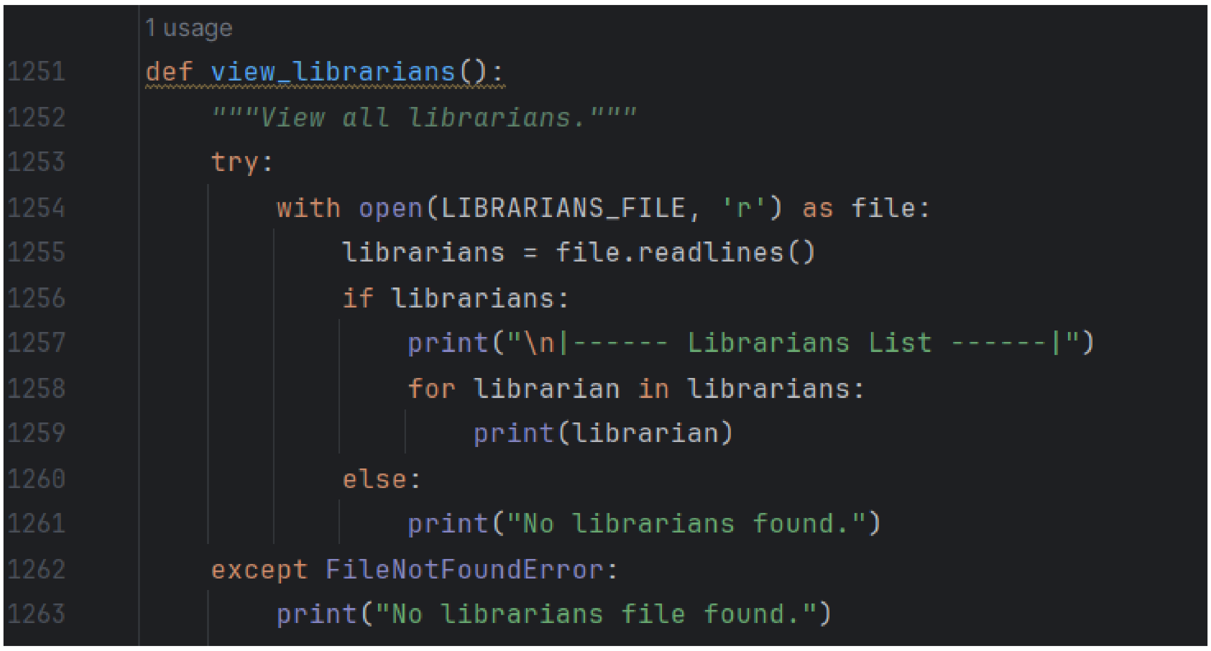
5.1.3.8 Randomising Librarian ID



```python
#Librarian Info Management
1 usage
def generate_librarian_id():
    """Generate a unique librarian ID (placeholder)."""
    return "LIB" + str(1000 + len(open(LIBRARIANS_FILE).readlines()))  # Simple ID generation
```

Random four-digit numbers are generated and attached to 'LIB' to automate generation of unique librarian ID.

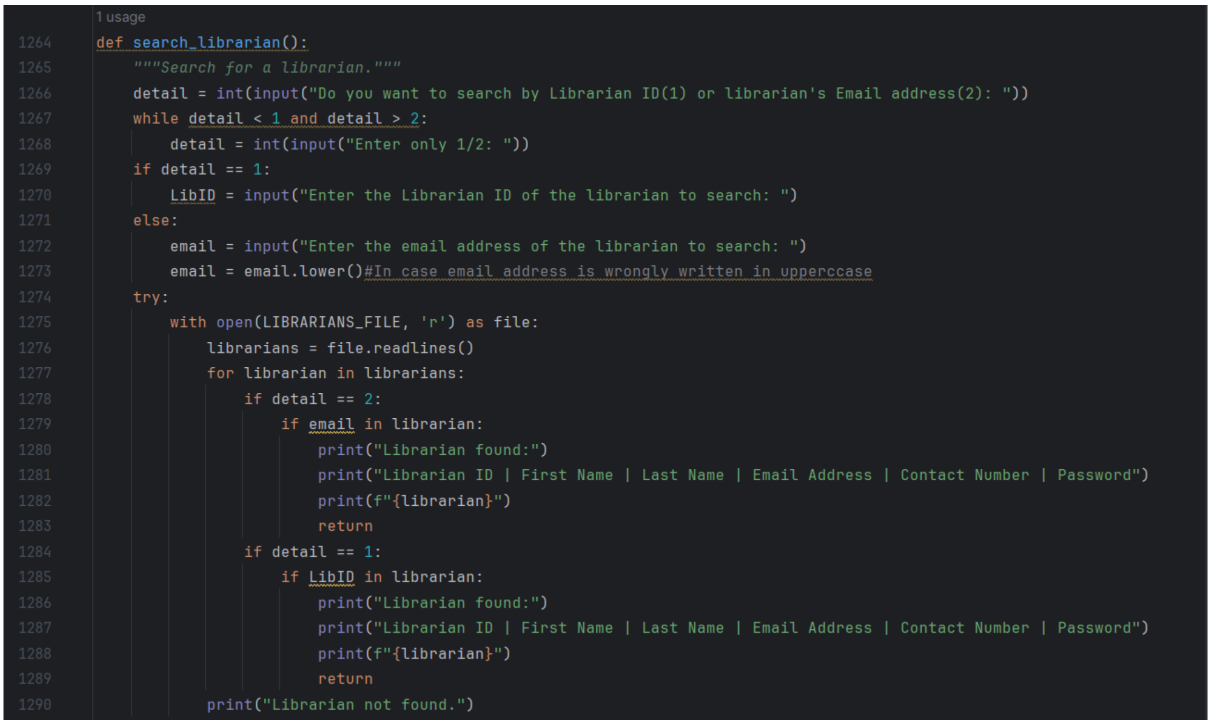
5.1.3.9 Display Librarians' list



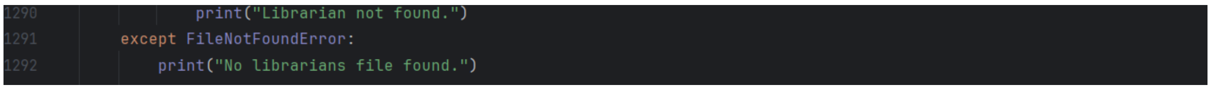
```
1 usage
def view_librarians():
    """View all librarians."""
    try:
        with open(LIBRARIANS_FILE, 'r') as file:
            librarians = file.readlines()
            if librarians:
                print("\n|------ Librarians List ------|")
                for librarian in librarians:
                    print(librarian)
            else:
                print("No librarians found.")
    except FileNotFoundError:
        print("No librarians file found.")
```

This will allow administrator to view all the librarians, similar to previous codes.

5.1.3.10 Search specific librarian's information



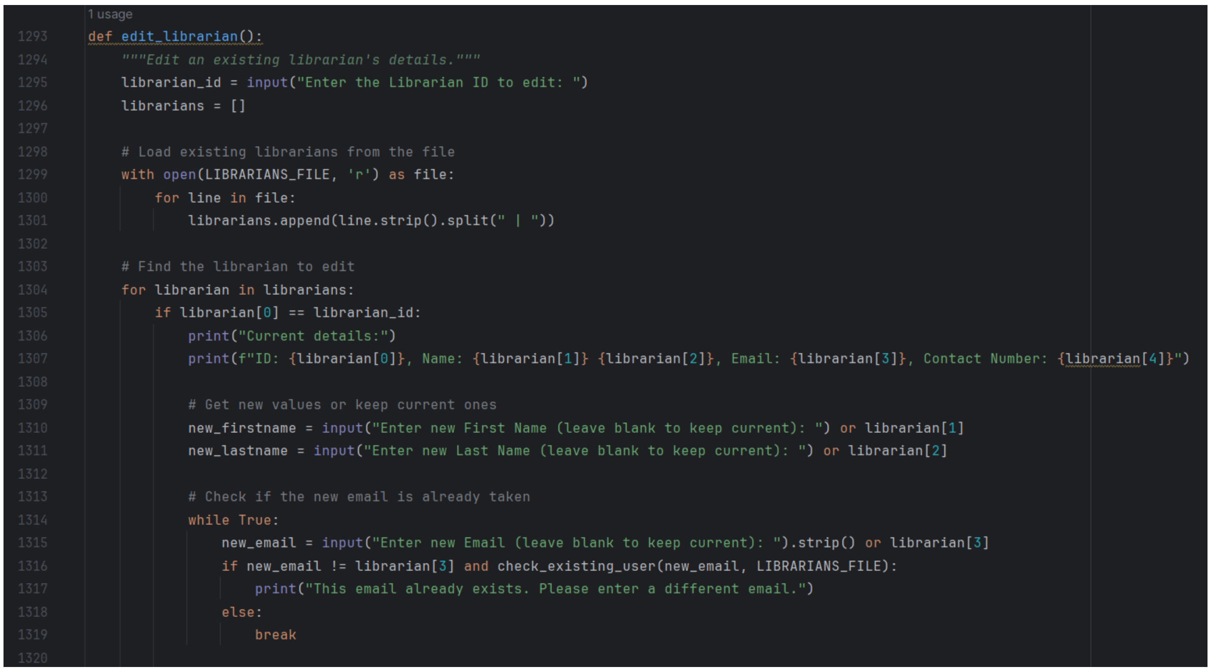
```
1 usage
def search_librarian():
    """Search for a librarian."""
    detail = int(input("Do you want to search by Librarian ID(1) or librarian's Email address(2): "))
    while detail < 1 and detail > 2:
        detail = int(input("Enter only 1/2: "))
    if detail == 1:
        LibID = input("Enter the Librarian ID of the librarian to search: ")
    else:
        email = input("Enter the email address of the librarian to search: ")
        email = email.lower()#In case email address is wrongly written in upperccase
    try:
        with open(LIBRARIANS_FILE, 'r') as file:
            librarians = file.readlines()
            for librarian in librarians:
                if detail == 2:
                    if email in librarian:
                        print("Librarian found:")
                        print("Librarian ID | First Name | Last Name | Email Address | Contact Number | Password")
                        print(f"{librarian}")
                        return
                if detail == 1:
                    if LibID in librarian:
                        print("Librarian found:")
                        print("Librarian ID | First Name | Last Name | Email Address | Contact Number | Password")
                        print(f"{librarian}")
                        return
            print("Librarian not found.")
```



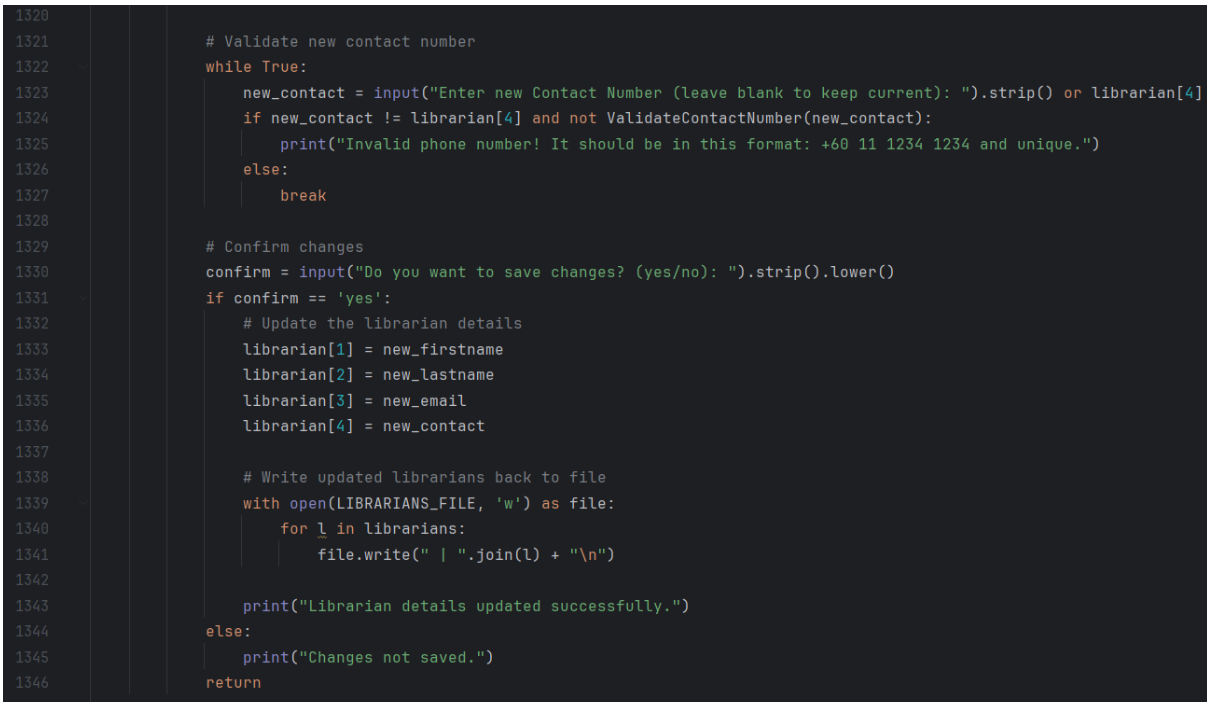
```
            print("Librarian not found.")
    except FileNotFoundError:
        print("No librarians file found.")
```

This will allow the administrator to search for specific librarians’ information.

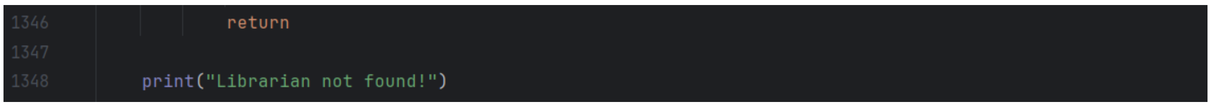
5.1.3.11 Edit specific librarian’s information



```python
def edit_librarian():
    """Edit an existing librarian's details."""
    librarian_id = input("Enter the Librarian ID to edit: ")
    librarians = []

    # Load existing librarians from the file
    with open(LIBRARIANS_FILE, 'r') as file:
        for line in file:
            librarians.append(line.strip().split(" | "))

    # Find the librarian to edit
    for librarian in librarians:
        if librarian[0] == librarian_id:
            print("Current details:")
            print(f"ID: {librarian[0]}, Name: {librarian[1]} {librarian[2]}, Email: {librarian[3]}, Contact Number: {librarian[4]}")

            # Get new values or keep current ones
            new_firstname = input("Enter new First Name (leave blank to keep current): ") or librarian[1]
            new_lastname = input("Enter new Last Name (leave blank to keep current): ") or librarian[2]

            # Check if the new email is already taken
            while True:
                new_email = input("Enter new Email (leave blank to keep current): ").strip() or librarian[3]
                if new_email != librarian[3] and check_existing_user(new_email, LIBRARIANS_FILE):
                    print("This email already exists. Please enter a different email.")
                else:
                    break

```



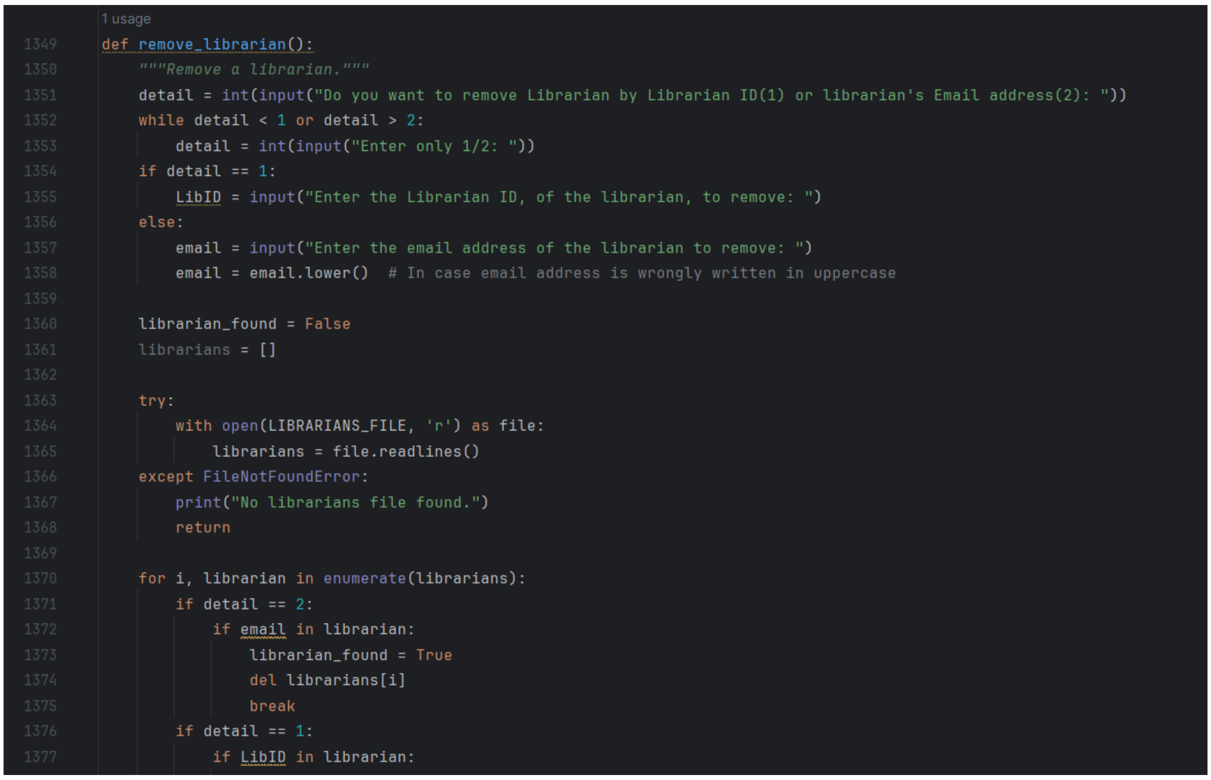
```python

            # Validate new contact number
            while True:
                new_contact = input("Enter new Contact Number (leave blank to keep current): ").strip() or librarian[4]
                if new_contact != librarian[4] and not ValidateContactNumber(new_contact):
                    print("Invalid phone number! It should be in this format: +60 11 1234 1234 and unique.")
                else:
                    break

            # Confirm changes
            confirm = input("Do you want to save changes? (yes/no): ").strip().lower()
            if confirm == 'yes':
                # Update the librarian details
                librarian[1] = new_firstname
                librarian[2] = new_lastname
                librarian[3] = new_email
                librarian[4] = new_contact

                # Write updated librarians back to file
                with open(LIBRARIANS_FILE, 'w') as file:
                    for l in librarians:
                        file.write(" | ".join(l) + "\n")

                print("Librarian details updated successfully.")
            else:
                print("Changes not saved.")
            return
```



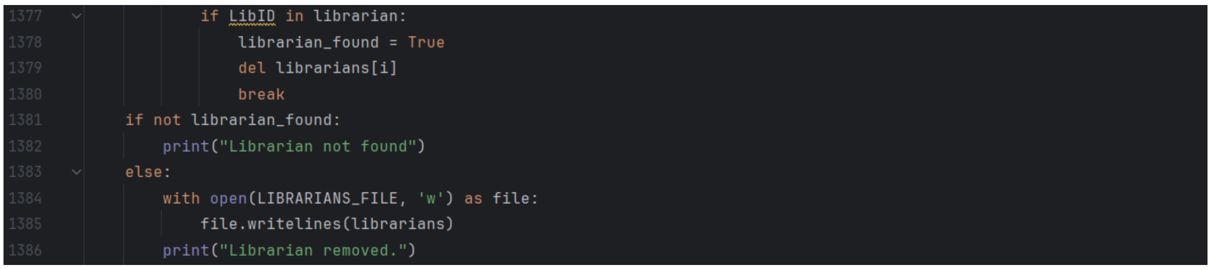
```python
            return

    print("Librarian not found!")
```

This will allow the administrator to edit librarians’ information.

5.1.3.12 Remove specific librarian from existing list



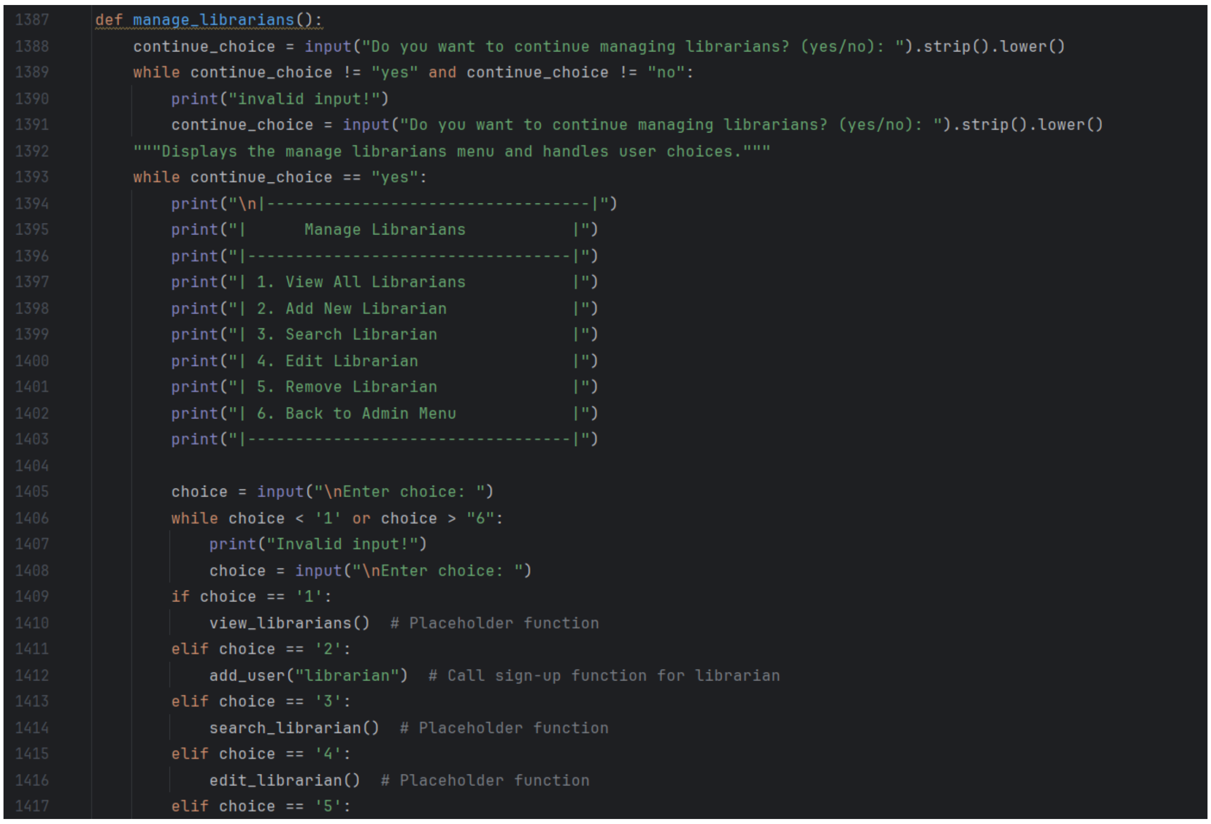
```
1 usage
def remove_librarian():
    """Remove a librarian."""
    detail = int(input("Do you want to remove Librarian by Librarian ID(1) or librarian's Email address(2): "))
    while detail < 1 or detail > 2:
        detail = int(input("Enter only 1/2: "))
    if detail == 1:
        LibID = input("Enter the Librarian ID, of the librarian, to remove: ")
    else:
        email = input("Enter the email address of the librarian to remove: ")
        email = email.lower()  # In case email address is wrongly written in uppercase

    librarian_found = False
    librarians = []

    try:
        with open(LIBRARIANS_FILE, 'r') as file:
            librarians = file.readlines()
    except FileNotFoundError:
        print("No librarians file found.")
        return

    for i, librarian in enumerate(librarians):
        if detail == 2:
            if email in librarian:
                librarian_found = True
                del librarians[i]
                break
        if detail == 1:
            if LibID in librarian:
```



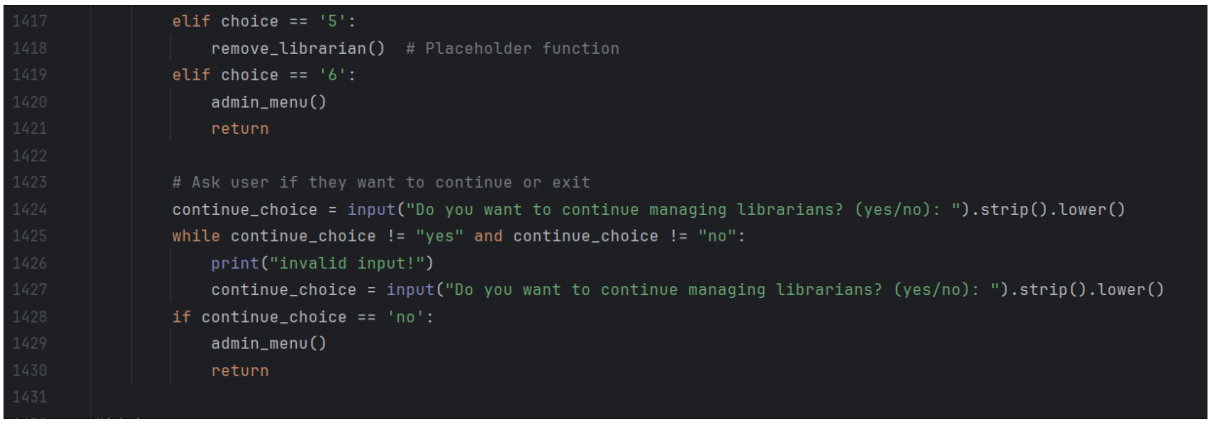
```
            if LibID in librarian:
                librarian_found = True
                del librarians[i]
                break
    if not librarian_found:
        print("Librarian not found")
    else:
        with open(LIBRARIANS_FILE, 'w') as file:
            file.writelines(librarians)
        print("Librarian removed.")
```

This will allow the administrator to remove librarians.

5.1.3.13 Managing librarians function (Menu to manage librarians)



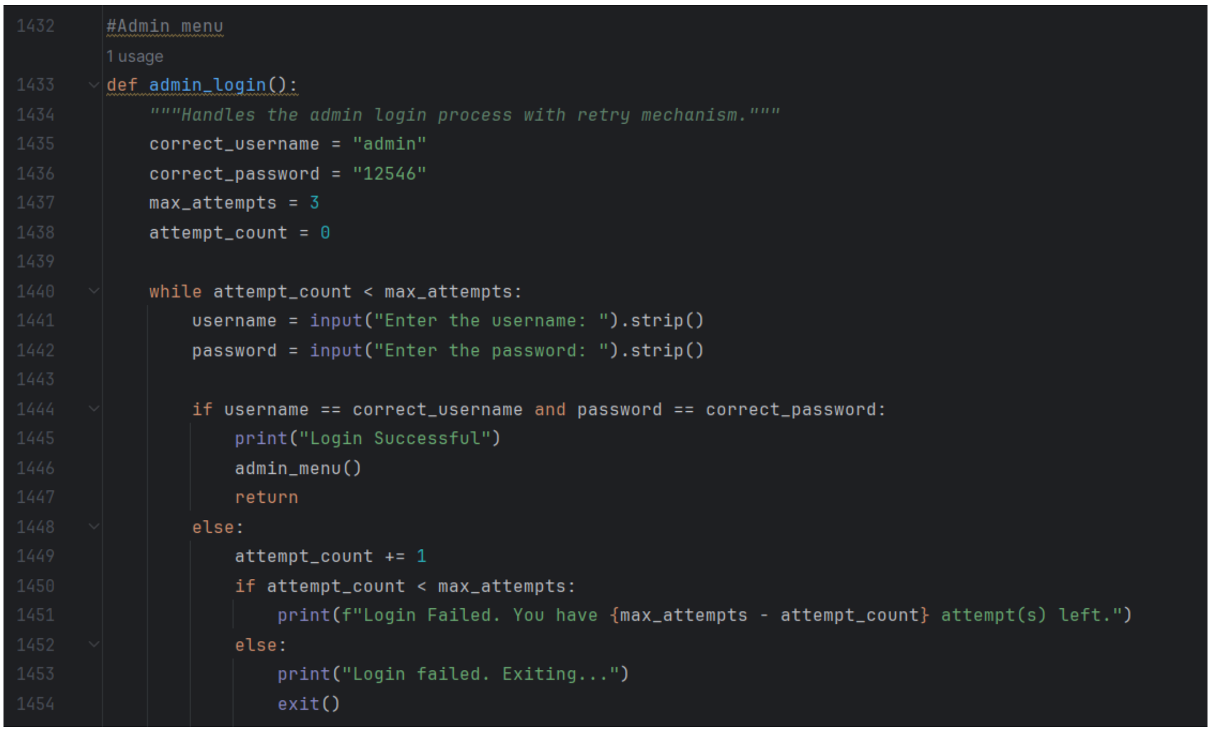
```python
def manage_librarians():
    continue_choice = input("Do you want to continue managing librarians? (yes/no): ").strip().lower()
    while continue_choice != "yes" and continue_choice != "no":
        print("invalid input!")
        continue_choice = input("Do you want to continue managing librarians? (yes/no): ").strip().lower()
    """Displays the manage librarians menu and handles user choices."""
    while continue_choice == "yes":
        print("\n|---------------------------------|")
        print("|      Manage Librarians          |")
        print("|---------------------------------|")
        print("| 1. View All Librarians          |")
        print("| 2. Add New Librarian            |")
        print("| 3. Search Librarian             |")
        print("| 4. Edit Librarian               |")
        print("| 5. Remove Librarian             |")
        print("| 6. Back to Admin Menu           |")
        print("|---------------------------------|")

        choice = input("\nEnter choice: ")
        while choice < '1' or choice > "6":
            print("Invalid input!")
            choice = input("\nEnter choice: ")
        if choice == '1':
            view_librarians()  # Placeholder function
        elif choice == '2':
            add_user("librarian")  # Call sign-up function for librarian
        elif choice == '3':
            search_librarian()  # Placeholder function
        elif choice == '4':
            edit_librarian()  # Placeholder function
        elif choice == '5':
```



```python
        elif choice == '5':
            remove_librarian()  # Placeholder function
        elif choice == '6':
            admin_menu()
            return

        # Ask user if they want to continue or exit
        continue_choice = input("Do you want to continue managing librarians? (yes/no): ").strip().lower()
        while continue_choice != "yes" and continue_choice != "no":
            print("invalid input!")
            continue_choice = input("Do you want to continue managing librarians? (yes/no): ").strip().lower()
        if continue_choice == 'no':
            admin_menu()
            return

```

This is menu for admin to manage librarians, like, viewing all librarians, adding new librarians, searching a specific librarian, editing a librarian information, removing a librarian from the text file and return to admin menu. This function is continuous unless the system administrator chooses to exit.

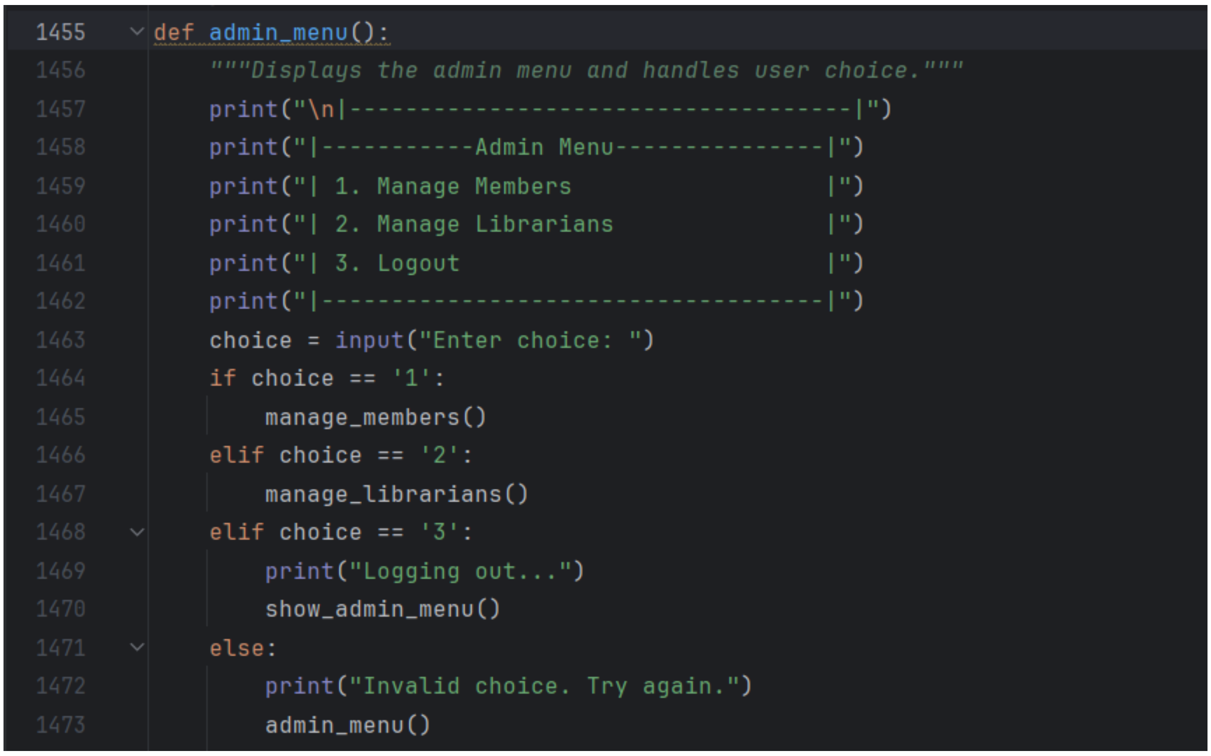
5.1.3.14 Admin login



```
#Admin menu
1 usage
def admin_login():
    """Handles the admin login process with retry mechanism."""
    correct_username = "admin"
    correct_password = "12546"
    max_attempts = 3
    attempt_count = 0

    while attempt_count < max_attempts:
        username = input("Enter the username: ").strip()
        password = input("Enter the password: ").strip()

        if username == correct_username and password == correct_password:
            print("Login Successful")
            admin_menu()
            return
        else:
            attempt_count += 1
            if attempt_count < max_attempts:
                print(f"Login Failed. You have {max_attempts - attempt_count} attempt(s) left.")
            else:
                print("Login failed. Exiting...")
                exit()
```

This function will allow system administrators to log in to the system by inputting their username and, changing code, within three attempts. If Login fails, they will exit the system, else they will be authorised to continue their job.

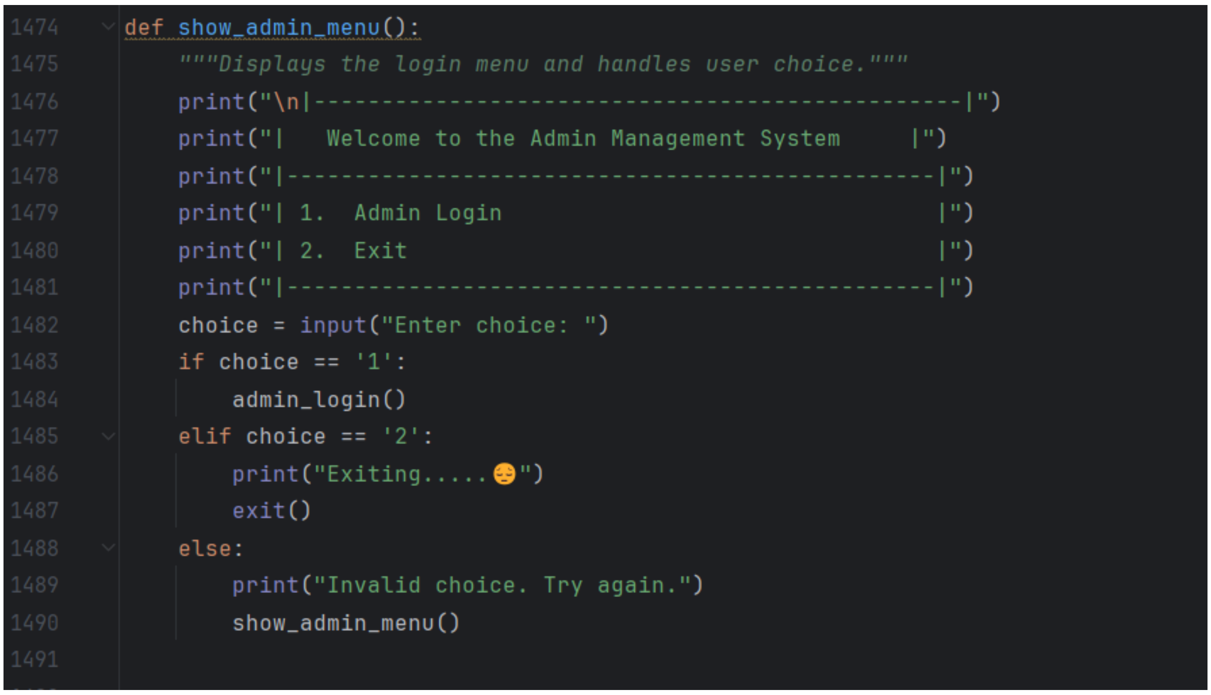
5.1.3.15 Admin Menu



```
def admin_menu():
    """Displays the admin menu and handles user choice."""
    print("\n|------------------------------------|")
    print("|-----------Admin Menu---------------|")
    print("| 1. Manage Members                  |")
    print("| 2. Manage Librarians               |")
    print("| 3. Logout                          |")
    print("|------------------------------------|")
    choice = input("Enter choice: ")
    if choice == '1':
        manage_members()
    elif choice == '2':
        manage_librarians()
    elif choice == '3':
        print("Logging out...")
        show_admin_menu()
    else:
        print("Invalid choice. Try again.")
        admin_menu()
```

The admin_menu () will provide system administrators the option to manage members or librarians or logout after completing their task.

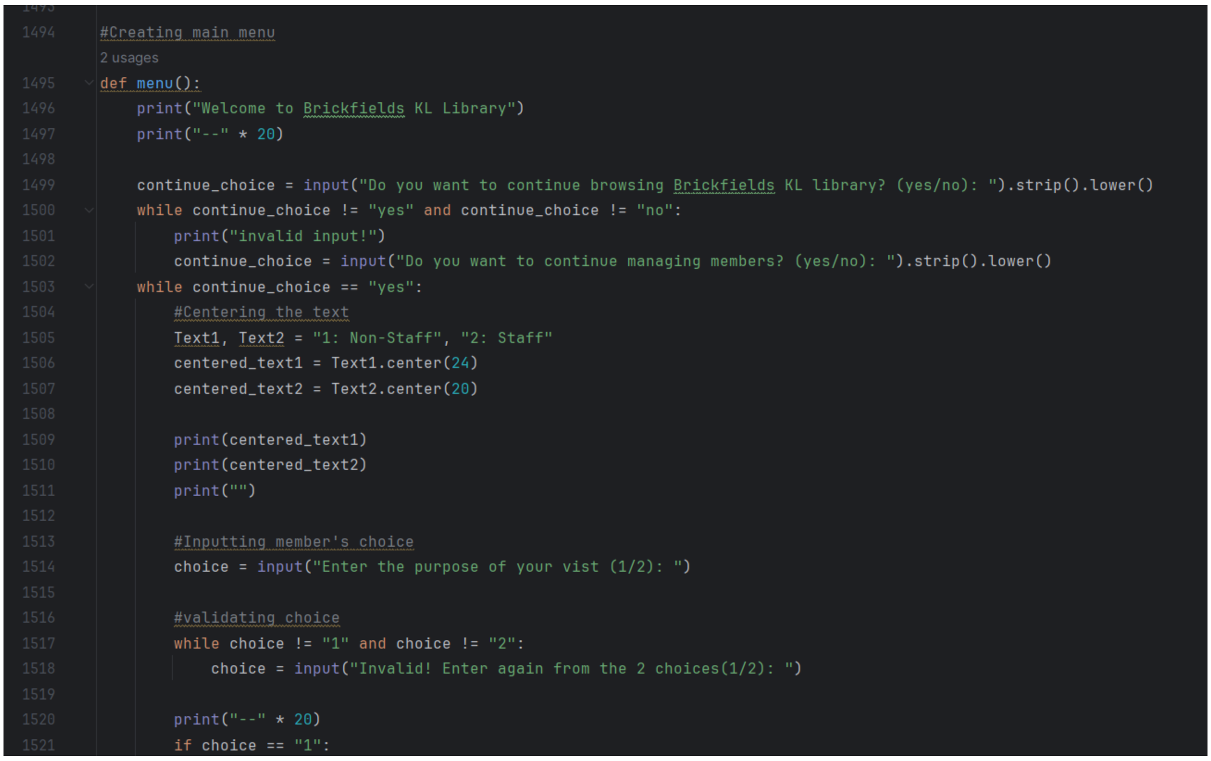
5.1.3.16 Show admin menu function



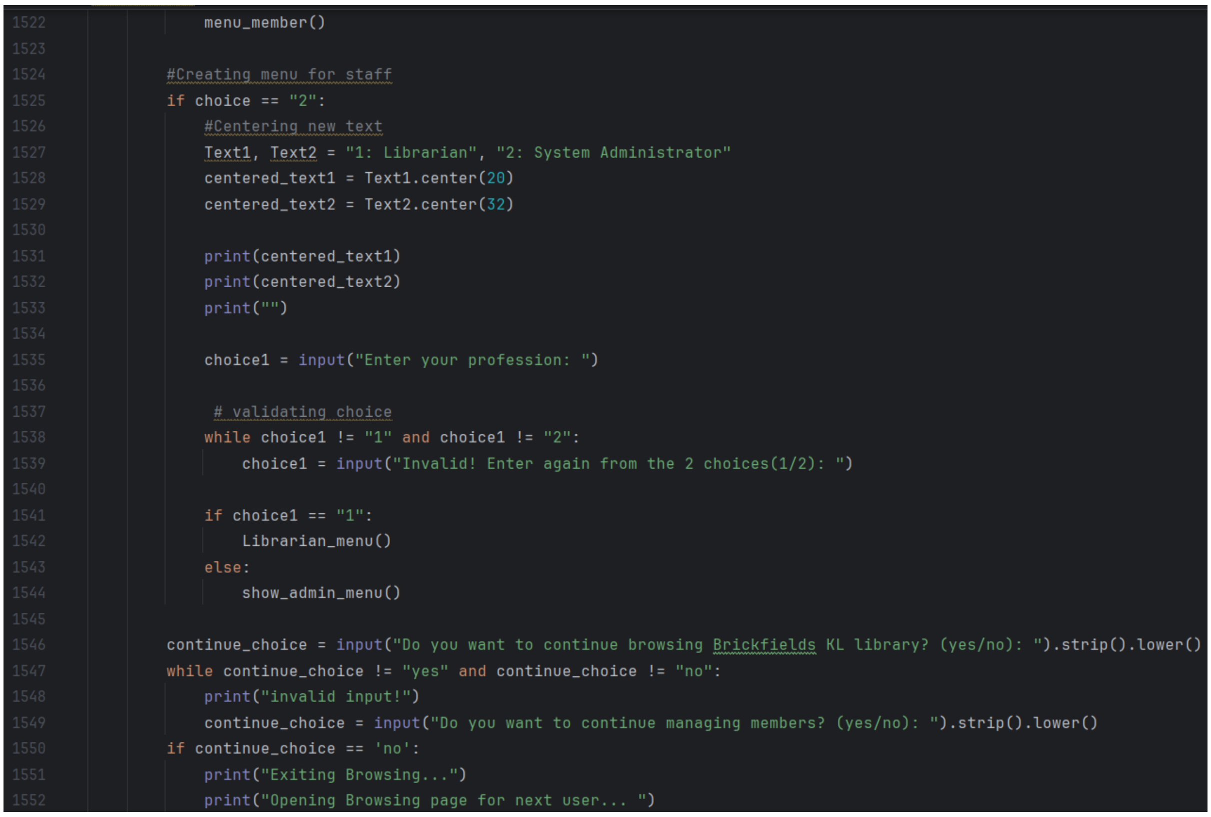
```python
def show_admin_menu():
    """Displays the login menu and handles user choice."""
    print("\n|-----------------------------------------------|")
    print("|   Welcome to the Admin Management System     |")
    print("|-----------------------------------------------|")
    print("| 1.  Admin Login                              |")
    print("| 2.  Exit                                     |")
    print("|-----------------------------------------------|")
    choice = input("Enter choice: ")
    if choice == '1':
        admin_login()
    elif choice == '2':
        print("Exiting.....😔")
        exit()
    else:
        print("Invalid choice. Try again.")
        show_admin_menu()
```

This function will allow administrators to login or exit and return to main menu.

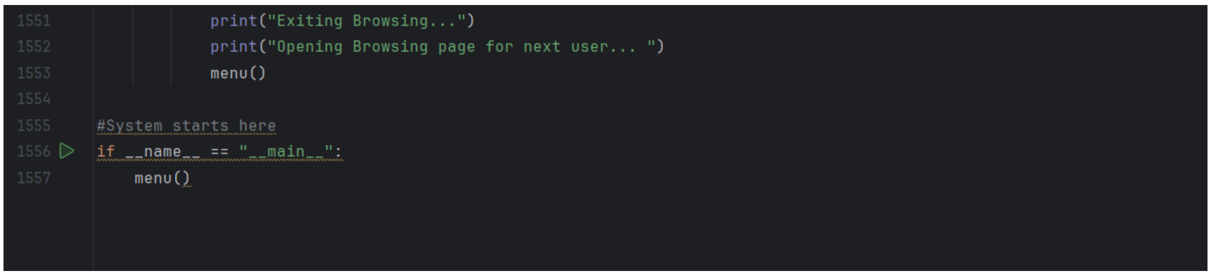
5.1.4 Main Menu for Brickfields Kuala Lumpur Community Library



```python
#Creating main menu
2 usages
def menu():
    print("Welcome to Brickfields KL Library")
    print("--" * 20)

    continue_choice = input("Do you want to continue browsing Brickfields KL library? (yes/no): ").strip().lower()
    while continue_choice != "yes" and continue_choice != "no":
        print("invalid input!")
        continue_choice = input("Do you want to continue managing members? (yes/no): ").strip().lower()
    while continue_choice == "yes":
        #Centering the text
        Text1, Text2 = "1: Non-Staff", "2: Staff"
        centered_text1 = Text1.center(24)
        centered_text2 = Text2.center(20)

        print(centered_text1)
        print(centered_text2)
        print("")

        #Inputting member's choice
        choice = input("Enter the purpose of your vist (1/2): ")

        #validating choice
        while choice != "1" and choice != "2":
            choice = input("Invalid! Enter again from the 2 choices(1/2): ")

        print("--" * 20)
        if choice == "1":
```



```
1522            menu_member()
1523
1524        #Creating menu for staff
1525        if choice == "2":
1526            #Centering new text
1527            Text1, Text2 = "1: Librarian", "2: System Administrator"
1528            centered_text1 = Text1.center(20)
1529            centered_text2 = Text2.center(32)
1530
1531            print(centered_text1)
1532            print(centered_text2)
1533            print("")
1534
1535            choice1 = input("Enter your profession: ")
1536
1537             # validating choice
1538            while choice1 != "1" and choice1 != "2":
1539                choice1 = input("Invalid! Enter again from the 2 choices(1/2): ")
1540
1541            if choice1 == "1":
1542                Librarian_menu()
1543            else:
1544                show_admin_menu()
1545
1546        continue_choice = input("Do you want to continue browsing Brickfields KL library? (yes/no): ").strip().lower()
1547        while continue_choice != "yes" and continue_choice != "no":
1548            print("invalid input!")
1549            continue_choice = input("Do you want to continue managing members? (yes/no): ").strip().lower()
1550        if continue_choice == 'no':
1551            print("Exiting Browsing...")
1552            print("Opening Browsing page for next user... ")
```

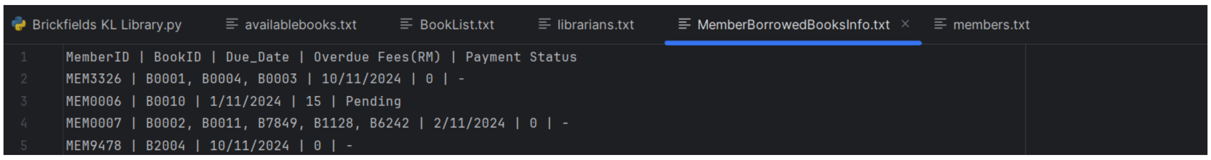


```
1551            print("Exiting Browsing...")
1552            print("Opening Browsing page for next user... ")
1553            menu()
1554
1555 #System starts here
1556 if __name__ == "__main__":
1557     menu()
```

The menu function is first function that will be executed, and it is being called at line 1556. This function will provide the interface for the user to enter as non-staff, if he is a library member, or as Staff, if he is either a librarian or system administrator. Then, based on their validated choices, the user will enter the member menu, librarian menu or admin menu. This process is continuous until the user chooses to exit.

5.1.5 Data Storage

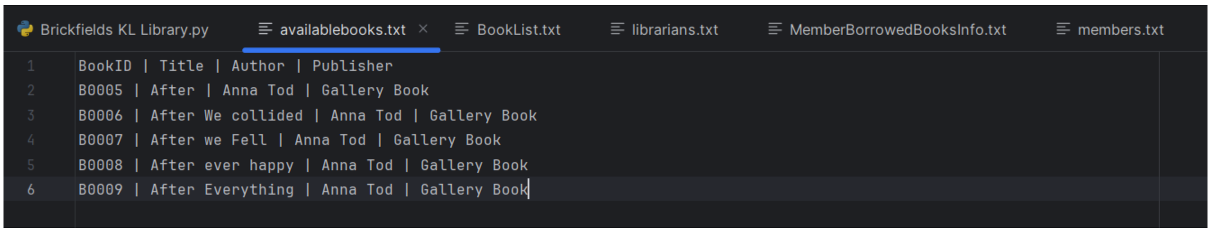
5.1.5.1 "MemberBorrowedBooksInfo.txt" file



```
MemberID | BookID | Due_Date | Overdue Fees(RM) | Payment Status
MEM3326 | B0001, B0004, B0003 | 10/11/2024 | 0 | -
MEM0006 | B0010 | 1/11/2024 | 15 | Pending
MEM0007 | B0002, B0011, B7849, B1128, B6242 | 2/11/2024 | 0 | -
MEM9478 | B2004 | 10/11/2024 | 0 | -
```

This file will store members details about their borrowed book, overdue fees, due date and payment status.

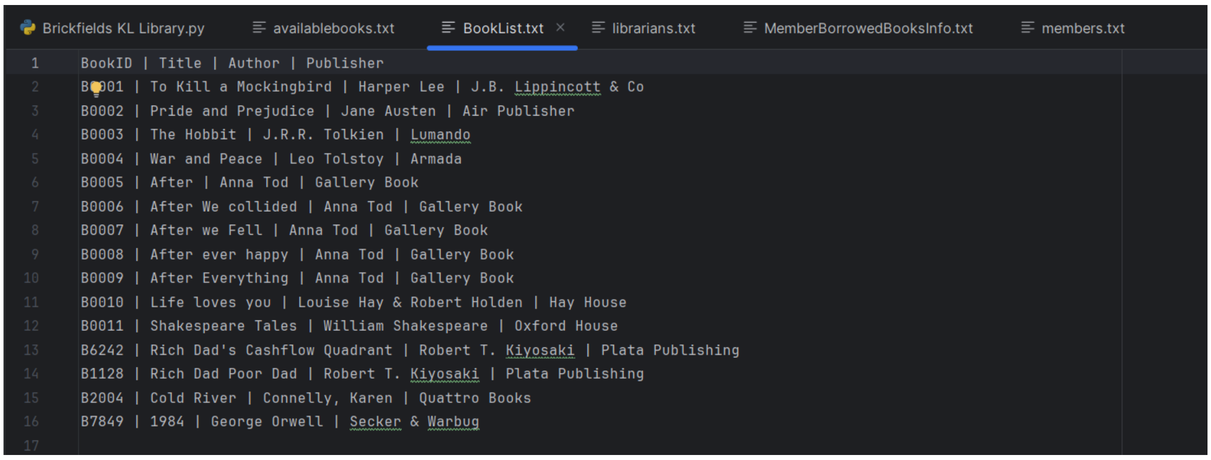
5.1.5.2 "availablebooks.txt" file



```
BookID | Title | Author | Publisher
B0005 | After | Anna Tod | Gallery Book
B0006 | After We collided | Anna Tod | Gallery Book
B0007 | After we Fell | Anna Tod | Gallery Book
B0008 | After ever happy | Anna Tod | Gallery Book
B0009 | After Everything | Anna Tod | Gallery Book
```

This file will store data about all books that are not lent to members and that are available to be borrowed.

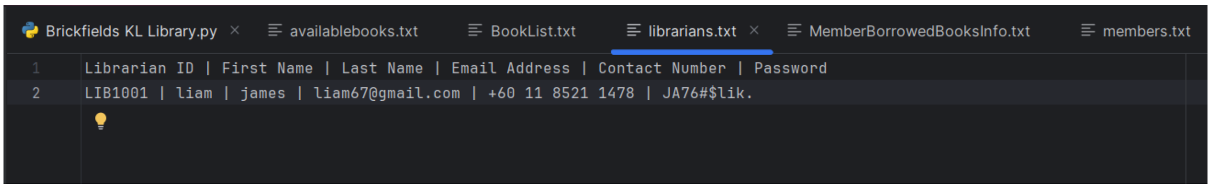
5.1.5.3 "BookList.txt" file



```
BookID | Title | Author | Publisher
B0001 | To Kill a Mockingbird | Harper Lee | J.B. Lippincott & Co
B0002 | Pride and Prejudice | Jane Austen | Air Publisher
B0003 | The Hobbit | J.R.R. Tolkien | Lumando
B0004 | War and Peace | Leo Tolstoy | Armada
B0005 | After | Anna Tod | Gallery Book
B0006 | After We collided | Anna Tod | Gallery Book
B0007 | After we Fell | Anna Tod | Gallery Book
B0008 | After ever happy | Anna Tod | Gallery Book
B0009 | After Everything | Anna Tod | Gallery Book
B0010 | Life loves you | Louise Hay & Robert Holden | Hay House
B0011 | Shakespeare Tales | William Shakespeare | Oxford House
B6242 | Rich Dad's Cashflow Quadrant | Robert T. Kiyosaki | Plata Publishing
B1128 | Rich Dad Poor Dad | Robert T. Kiyosaki | Plata Publishing
B2004 | Cold River | Connelly, Karen | Quattro Books
B7849 | 1984 | George Orwell | Secker & Warbug
```

This file will store all the books that are in the library, regardless of their availability status.

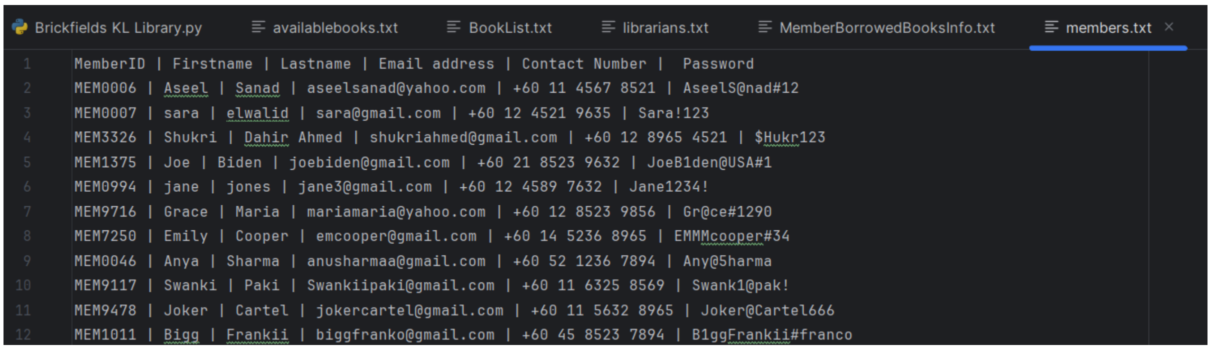
5.1.5.4 "librarians.txt" file



```
Librarian ID | First Name | Last Name | Email Address | Contact Number | Password
LIB1001 | liam | james | liam67@gmail.com | +60 11 8521 1478 | JA76#$lik.
```

This file will store all the data about the librarians.

5.1.5.5 “members.txt” file

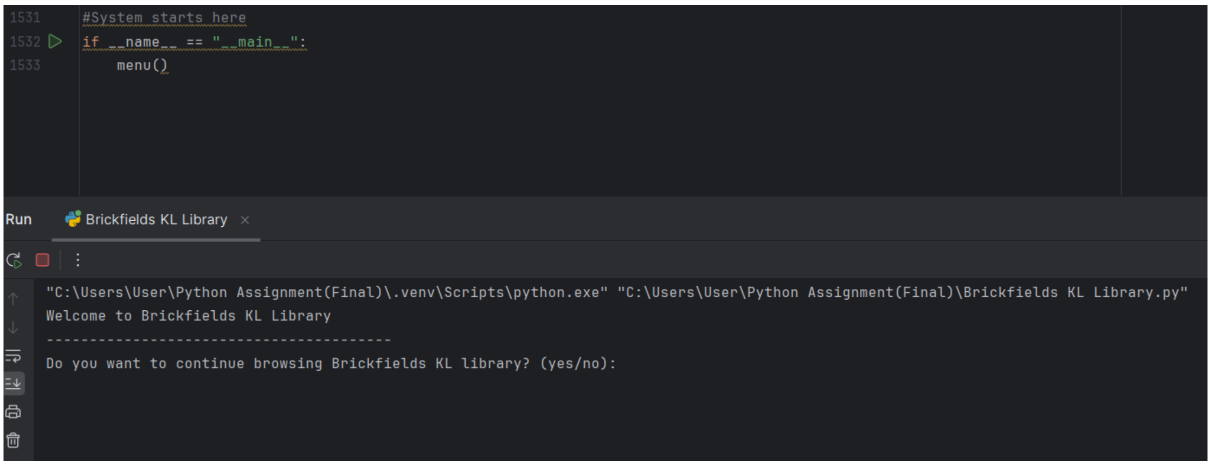


```
MemberID | Firstname | Lastname | Email address | Contact Number |  Password
MEM0006 | Aseel | Sanad | aseelsanad@yahoo.com | +60 11 4567 8521 | AseelS@nad#12
MEM0007 | sara | elwalid | sara@gmail.com | +60 12 4521 9635 | Sara!123
MEM3326 | Shukri | Dahir Ahmed | shukriahmed@gmail.com | +60 12 8965 4521 | $Hukr123
MEM1375 | Joe | Biden | joebiden@gmail.com | +60 21 8523 9632 | JoeB1den@USA#1
MEM0994 | jane | jones | jane3@gmail.com | +60 12 4589 7632 | Jane1234!
MEM9716 | Grace | Maria | mariamaria@yahoo.com | +60 12 8523 9856 | Gr@ce#1290
MEM7250 | Emily | Cooper | emcooper@gmail.com | +60 14 5236 8965 | EMMMcooper#34
MEM0046 | Anya | Sharma | anusharmaa@gmail.com | +60 52 1236 7894 | Any@5harma
MEM9117 | Swanki | Paki | Swankiipaki@gmail.com | +60 11 6325 8569 | Swank1@pak!
MEM9478 | Joker | Cartel | jokercartel@gmail.com | +60 11 5632 8965 | Joker@Cartel666
MEM1011 | Bigg | Frankii | biggfranko@gmail.com | +60 45 8523 7894 | B1ggFrankii#franco
```

This file will store data about library members.

6.0 Sample Input and Output

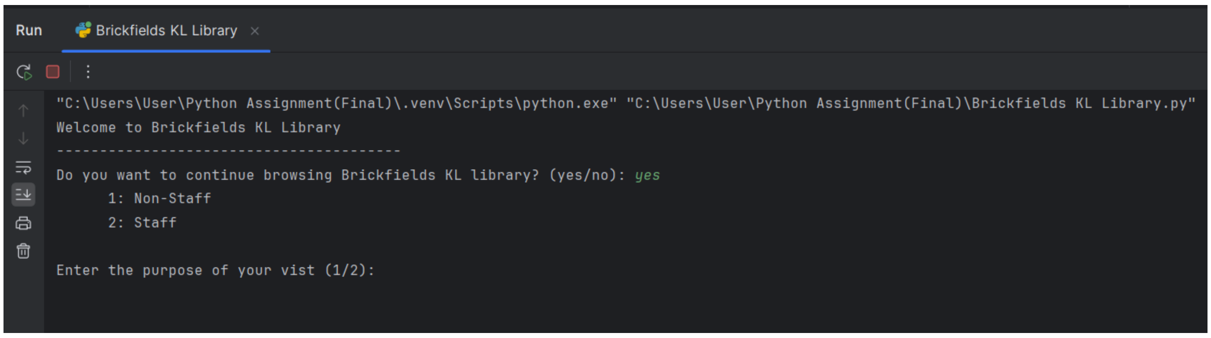
6.1 member



```
#System starts here
if __name__ == "__main__":
    menu()
```

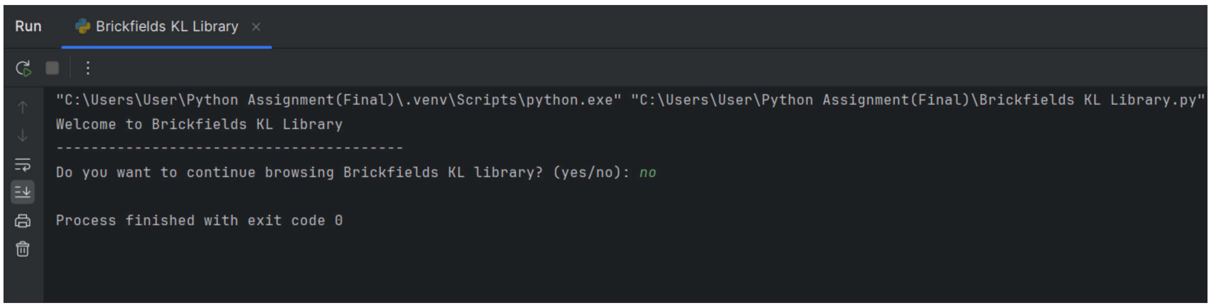
```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no):
```

Upon running the code, there is a welcome message and a message prompting the user to continue browsing or exit the system.



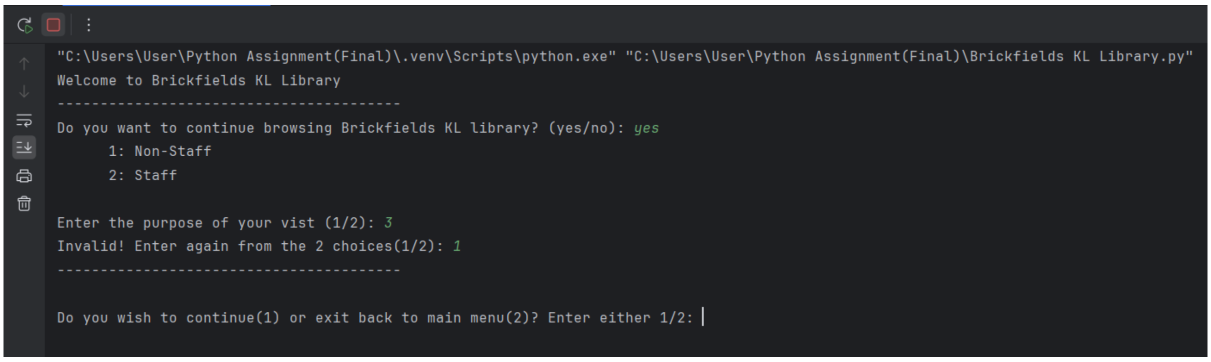
```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2):
```

If the user enters yes, the latter will enter a menu asking about the purpose of their visit, either as Staff (librarian, or system administrator) or Non-Staff (library Member)

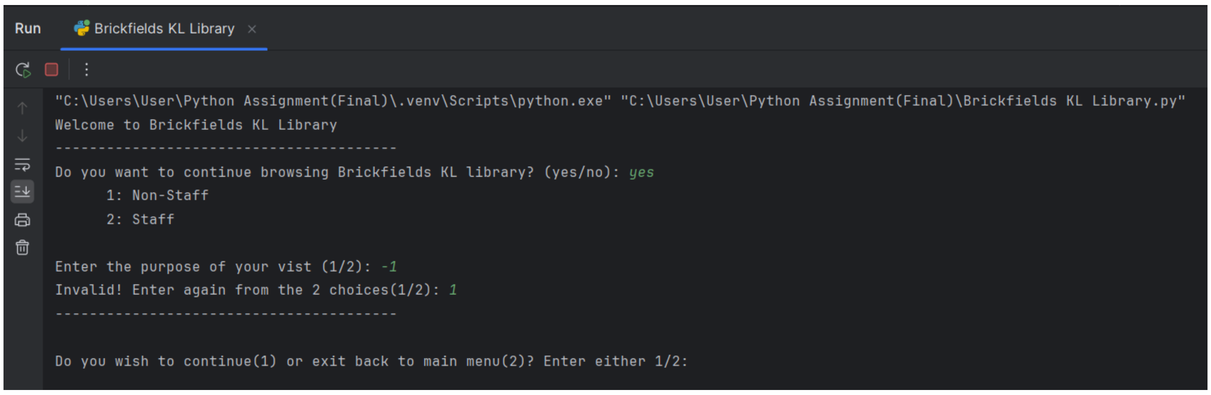


```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): no

Process finished with exit code 0
```

Here, the user is exiting the system.

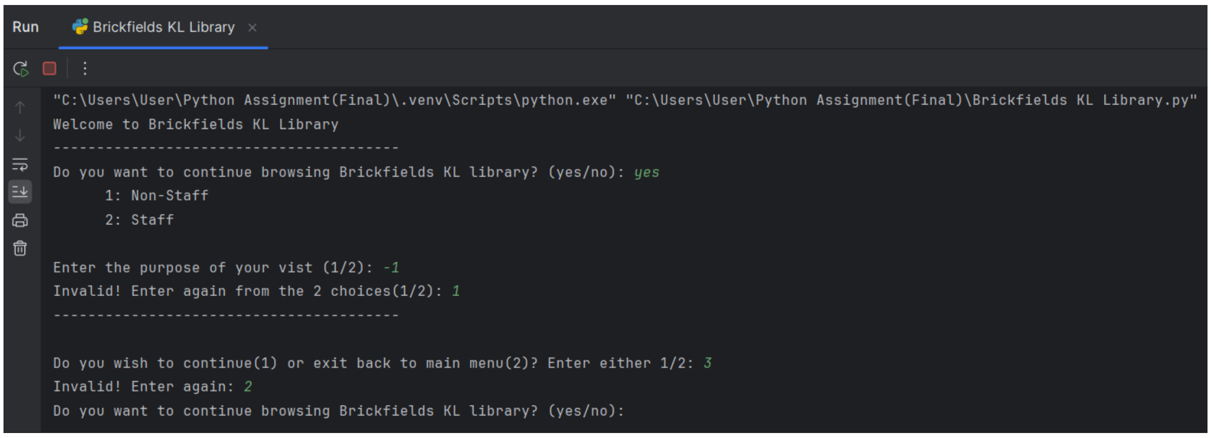


```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): 3
Invalid! Enter again from the 2 choices(1/2): 1
----------------------------------------

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 
```



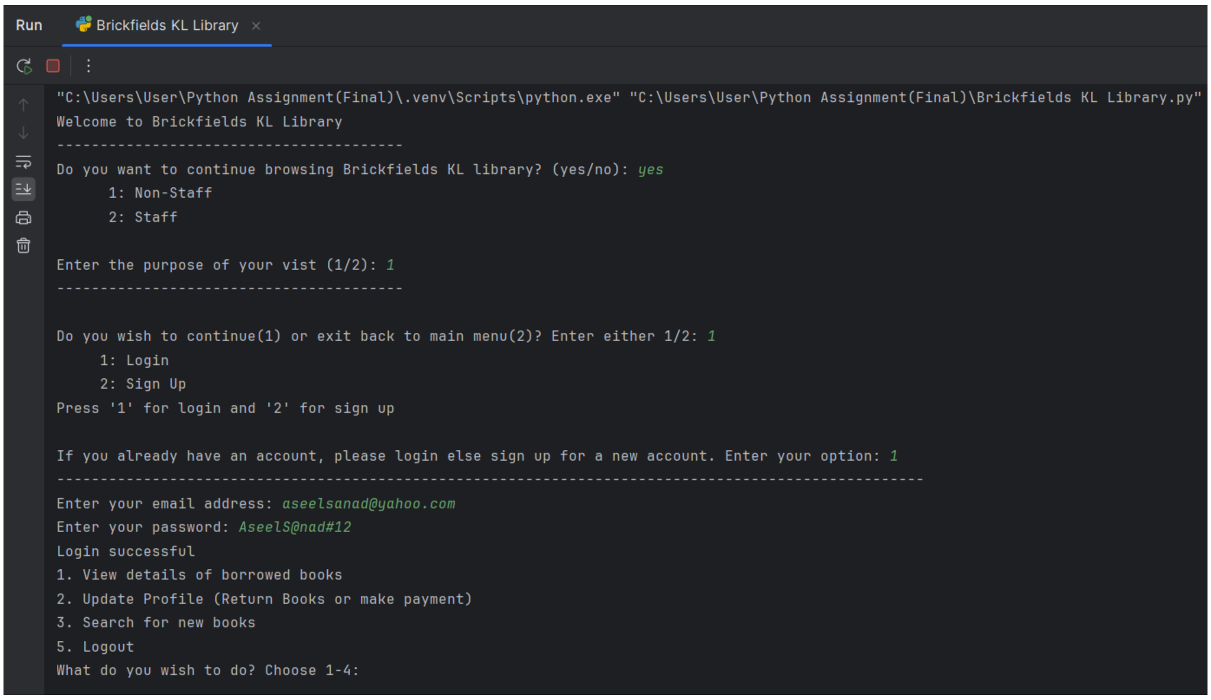
```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): -1
Invalid! Enter again from the 2 choices(1/2): 1
----------------------------------------

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```

The user has only two choices, if he enters a wrong input, he will have to continue enter his choice until the correct input (1/2) is entered.



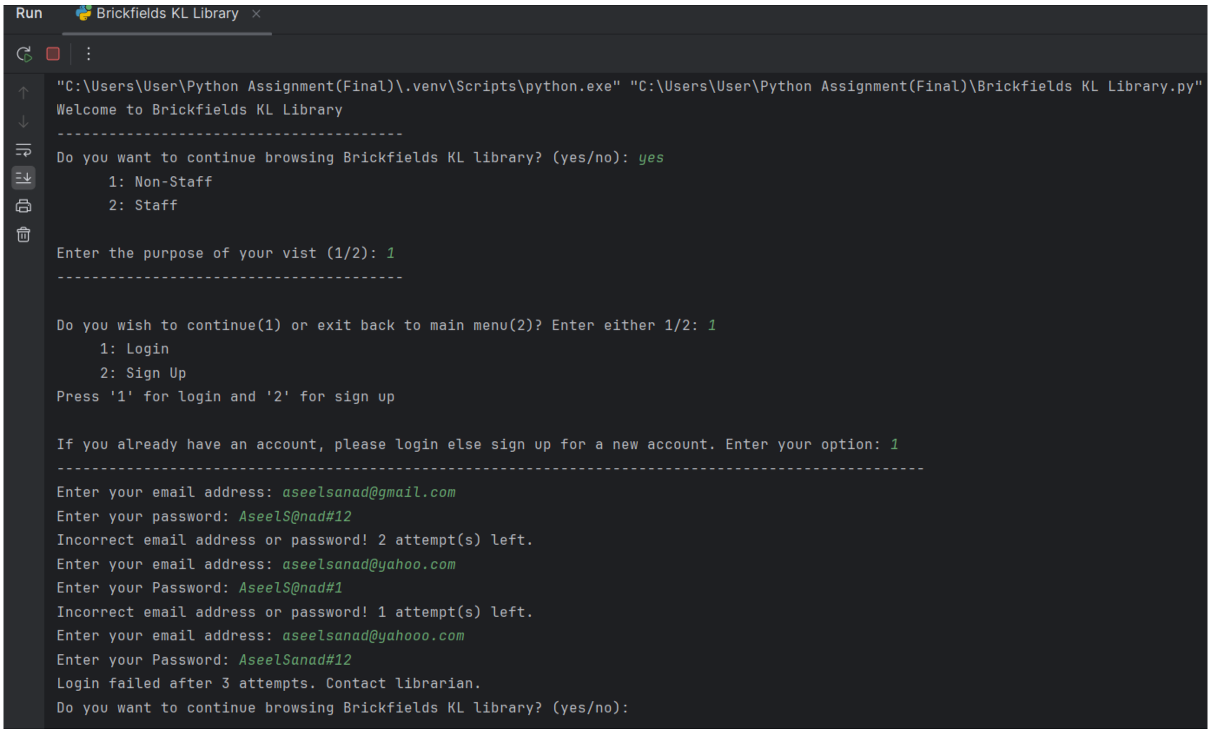
```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): -1
Invalid! Enter again from the 2 choices(1/2): 1
----------------------------------------

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 3
Invalid! Enter again: 2
Do you want to continue browsing Brickfields KL library? (yes/no):
```

If the user chooses one, he is most probably a library member, and hence, he will enter the main menu and where the member menu will be called. In this case, the user has chosen the exit and hence, he is at the interface where he was initially.



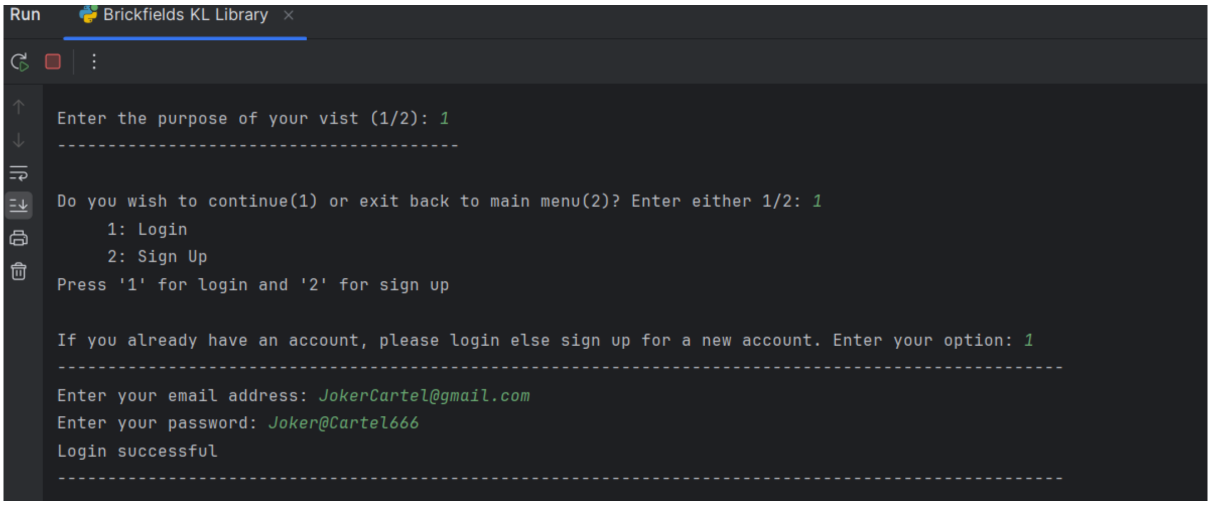
```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): 1
----------------------------------------

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 1
----------------------------------------------------------------------------------------------------
Enter your email address: aseelsanad@yahoo.com
Enter your password: AseelS@nad#12
Login successful
1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:
```

Here the user has chosen to continue and has opted to login. Upon entering the correct email address and password, she has access to member menu.

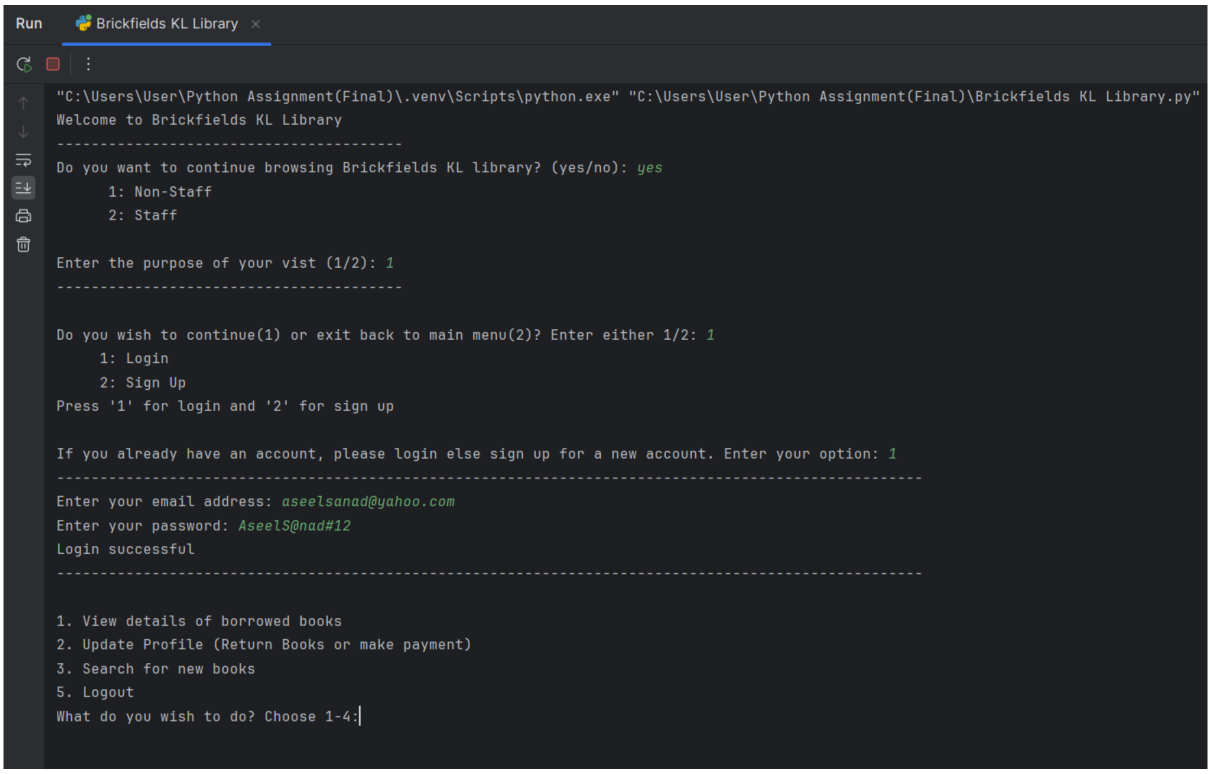


```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): 1
----------------------------------------

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 1
----------------------------------------------------------------------------------------------------
Enter your email address: aseelsanad@gmail.com
Enter your password: AseelS@nad#12
Incorrect email address or password! 2 attempt(s) left.
Enter your email address: aseelsanad@yahoo.com
Enter your Password: AseelS@nad#1
Incorrect email address or password! 1 attempt(s) left.
Enter your email address: aseelsanad@yahooo.com
Enter your Password: AseelSanad#12
Login failed after 3 attempts. Contact librarian.
Do you want to continue browsing Brickfields KL library? (yes/no):
```

However, if she wrongly inputs her email address, in this case, instead of @yahoo.com, she has entered @gmail .com and has also wrongly entered her password. After exhausting her attempts, login has failed, and she is back at the initial interface.

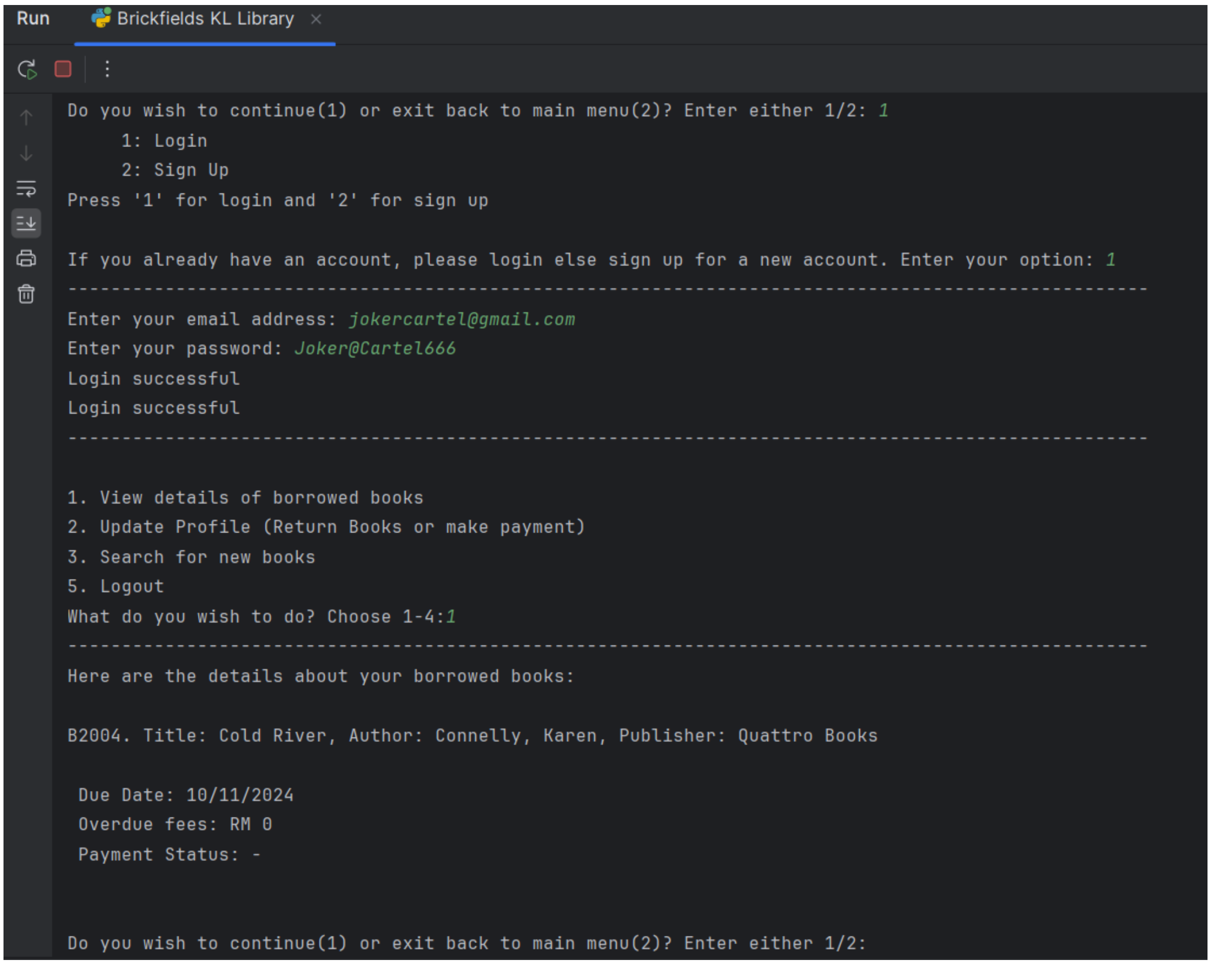


```
Enter the purpose of your vist (1/2): 1
----------------------------------------

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 1
----------------------------------------------------------------------------------------------------
Enter your email address: JokerCartel@gmail.com
Enter your password: Joker@Cartel666
Login successful
----------------------------------------------------------------------------------------------------
```



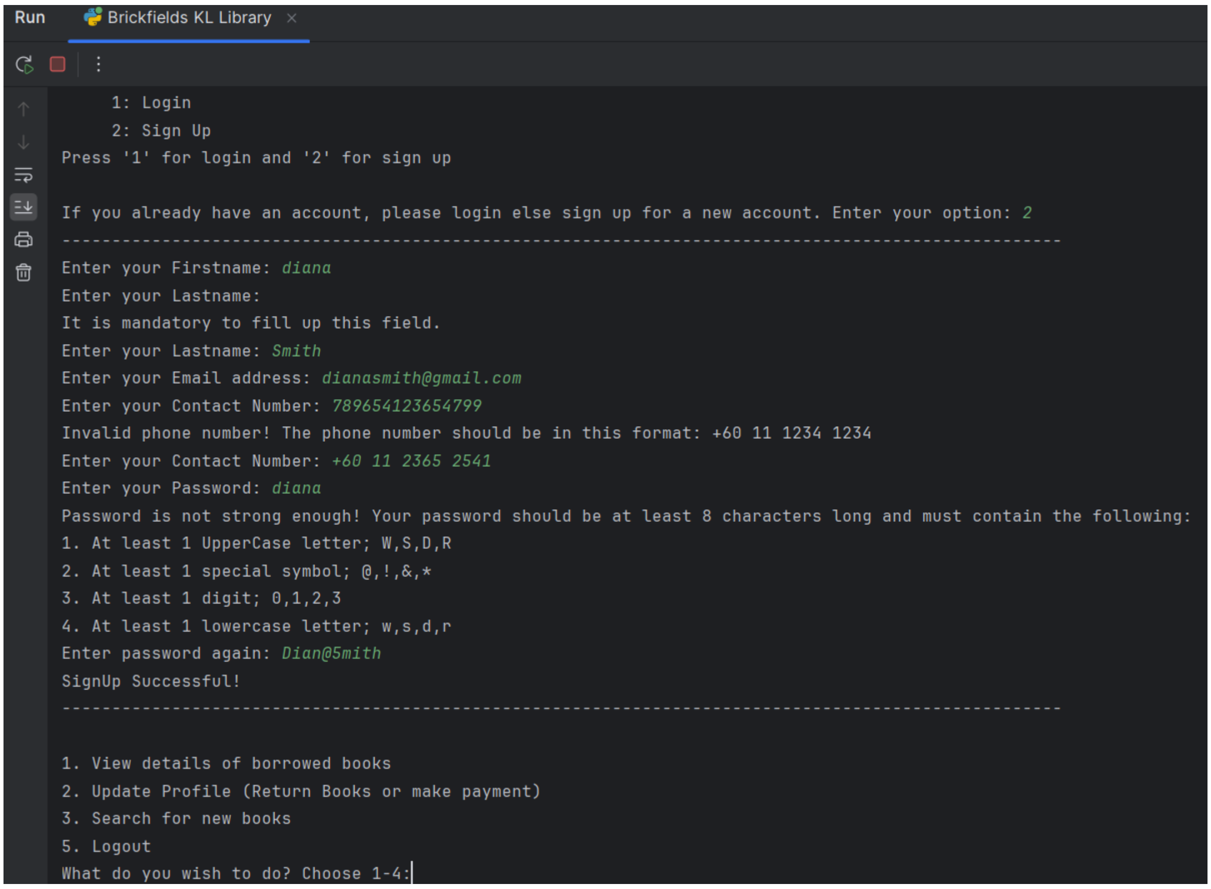
```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
     1: Non-Staff
     2: Staff

Enter the purpose of your vist (1/2): 1
----------------------------------------

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 1
----------------------------------------------------------------------------------------------------
Enter your email address: aseelsanad@yahoo.com
Enter your password: AseelS@nad#12
Login successful
----------------------------------------------------------------------------------------------------

1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:
```

Here are other examples of successful logins.



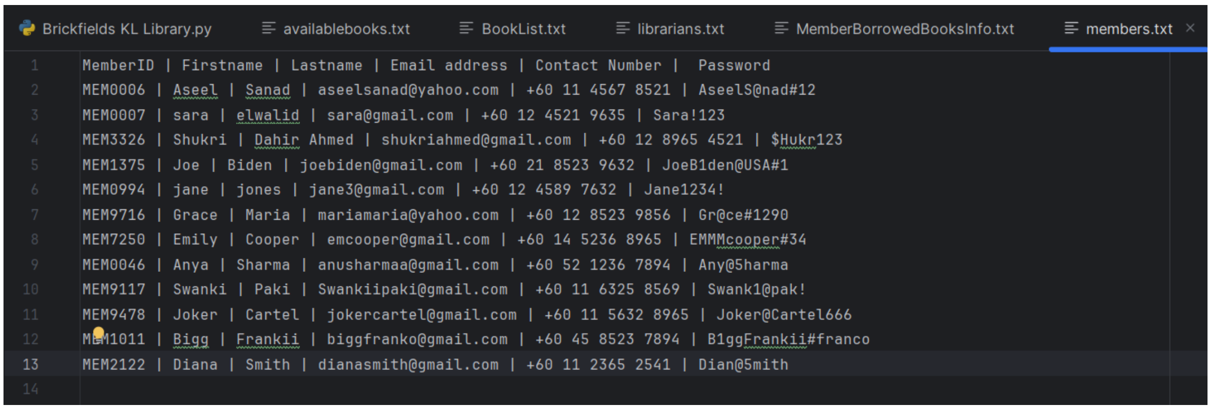
```
Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 1
----------------------------------------------------------------------------------------------------
Enter your email address: jokercartel@gmail.com
Enter your password: Joker@Cartel666
Login successful
Login successful
----------------------------------------------------------------------------------------------------

1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:1
----------------------------------------------------------------------------------------------------
Here are the details about your borrowed books:

B2004. Title: Cold River, Author: Connelly, Karen, Publisher: Quattro Books

 Due Date: 10/11/2024
 Overdue fees: RM 0
 Payment Status: -


Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```

After a successful login, joker Cartel wishes to view details about his borrowed list. From this example, we can see that he has borrowed Cold River Book, with Book ID, B2004, and he is supposed to return it by 10th November. Currently, he has zero fees pending.



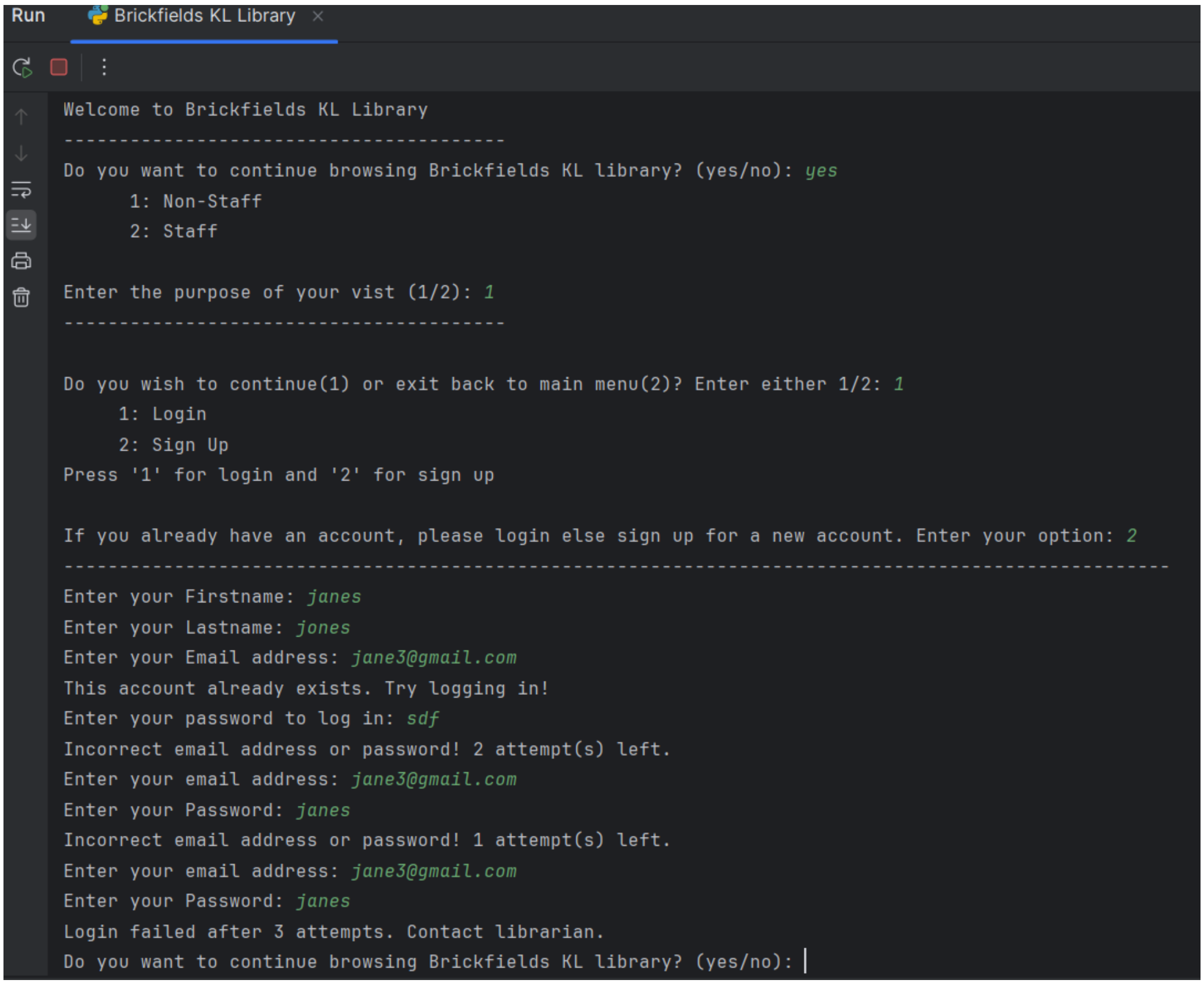
```
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 2
-----------------------------------------------------------------------------------------------------
Enter your Firstname: diana
Enter your Lastname:
It is mandatory to fill up this field.
Enter your Lastname: Smith
Enter your Email address: dianasmith@gmail.com
Enter your Contact Number: 789654123654799
Invalid phone number! The phone number should be in this format: +60 11 1234 1234
Enter your Contact Number: +60 11 2365 2541
Enter your Password: diana
Password is not strong enough! Your password should be at least 8 characters long and must contain the following:
1. At least 1 UpperCase letter; W,S,D,R
2. At least 1 special symbol; @,!,&,*
3. At least 1 digit; 0,1,2,3
4. At least 1 lowercase letter; w,s,d,r
Enter password again: Dian@5mith
SignUp Successful!
-----------------------------------------------------------------------------------------------------


1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:
```

Here, the user named Diana needs to sign up. She cannot leave any empty fields, so she is prompted to continue fill up her last name, until there is data present. The contact number was also incorrect at first, it was too long. Her initial password was not strong enough. After entering valid information, signing up was successful.



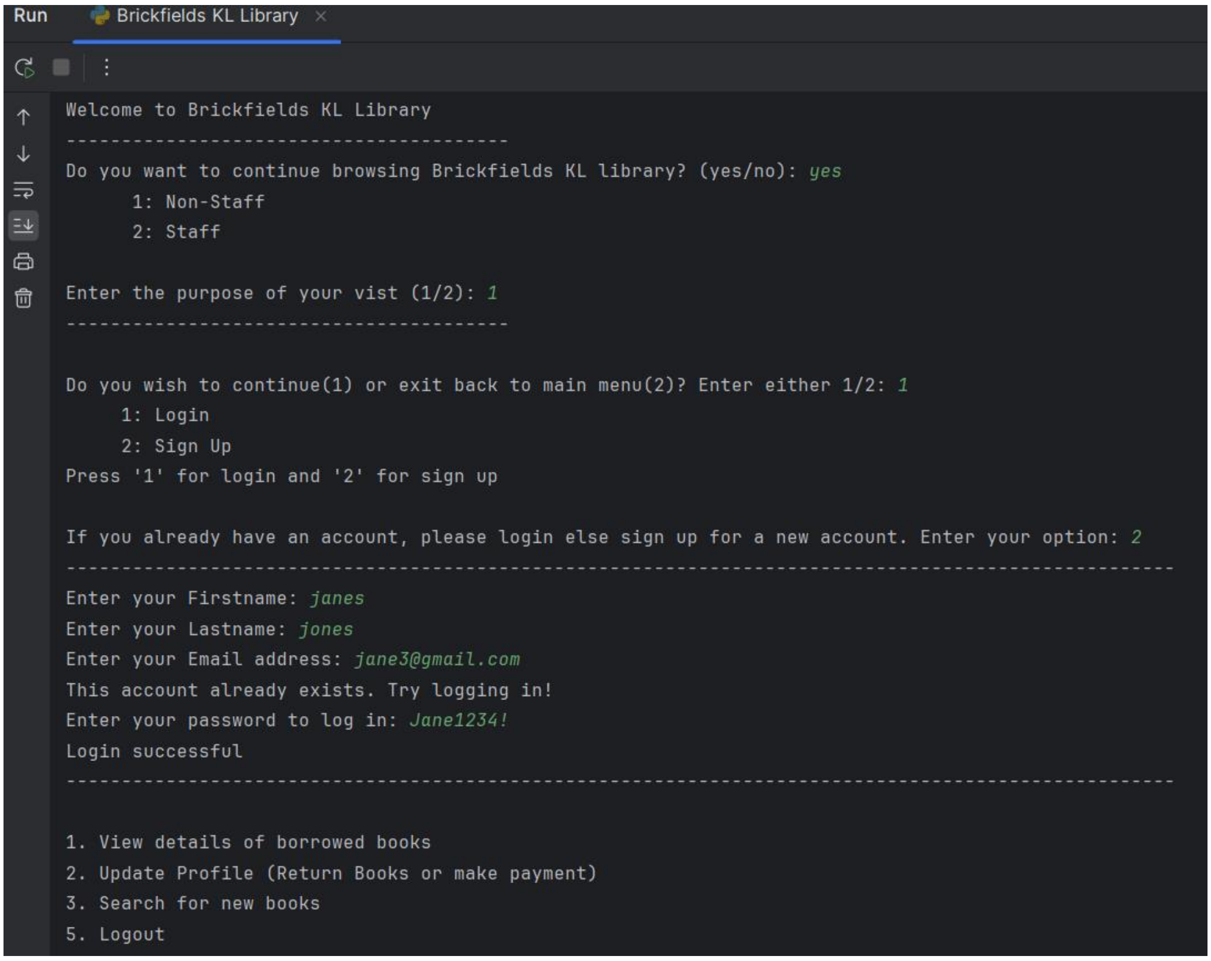
```
MemberID | Firstname | Lastname | Email address | Contact Number |  Password
MEM0006 | Aseel | Sanad | aseelsanad@yahoo.com | +60 11 4567 8521 | AseelS@nad#12
MEM0007 | sara | elwalid | sara@gmail.com | +60 12 4521 9635 | Sara!123
MEM3326 | Shukri | Dahir Ahmed | shukriahmed@gmail.com | +60 12 8965 4521 | $Hukr123
MEM1375 | Joe | Biden | joebiden@gmail.com | +60 21 8523 9632 | JoeB1den@USA#1
MEM0994 | jane | jones | jane3@gmail.com | +60 12 4589 7632 | Jane1234!
MEM9716 | Grace | Maria | mariamaria@yahoo.com | +60 12 8523 9856 | Gr@ce#1290
MEM7250 | Emily | Cooper | emcooper@gmail.com | +60 14 5236 8965 | EMMcooper#34
MEM0046 | Anya | Sharma | anusharmaa@gmail.com | +60 52 1236 7894 | Any@5harma
MEM9117 | Swanki | Paki | Swankiipaki@gmail.com | +60 11 6325 8569 | Swank1@pak!
MEM9478 | Joker | Cartel | jokercartel@gmail.com | +60 11 5632 8965 | Joker@Cartel666
MEM1011 | Bigg | Frankii | biggfranko@gmail.com | +60 45 8523 7894 | B1ggFrankii#franco
MEM2122 | Diana | Smith | dianasmith@gmail.com | +60 11 2365 2541 | Dian@5mith
```

Her data was stored in members.txt file. Despite having written her name in lowercase, the system, using capitalise (), in-built function, has stored it as ‘Diana’.



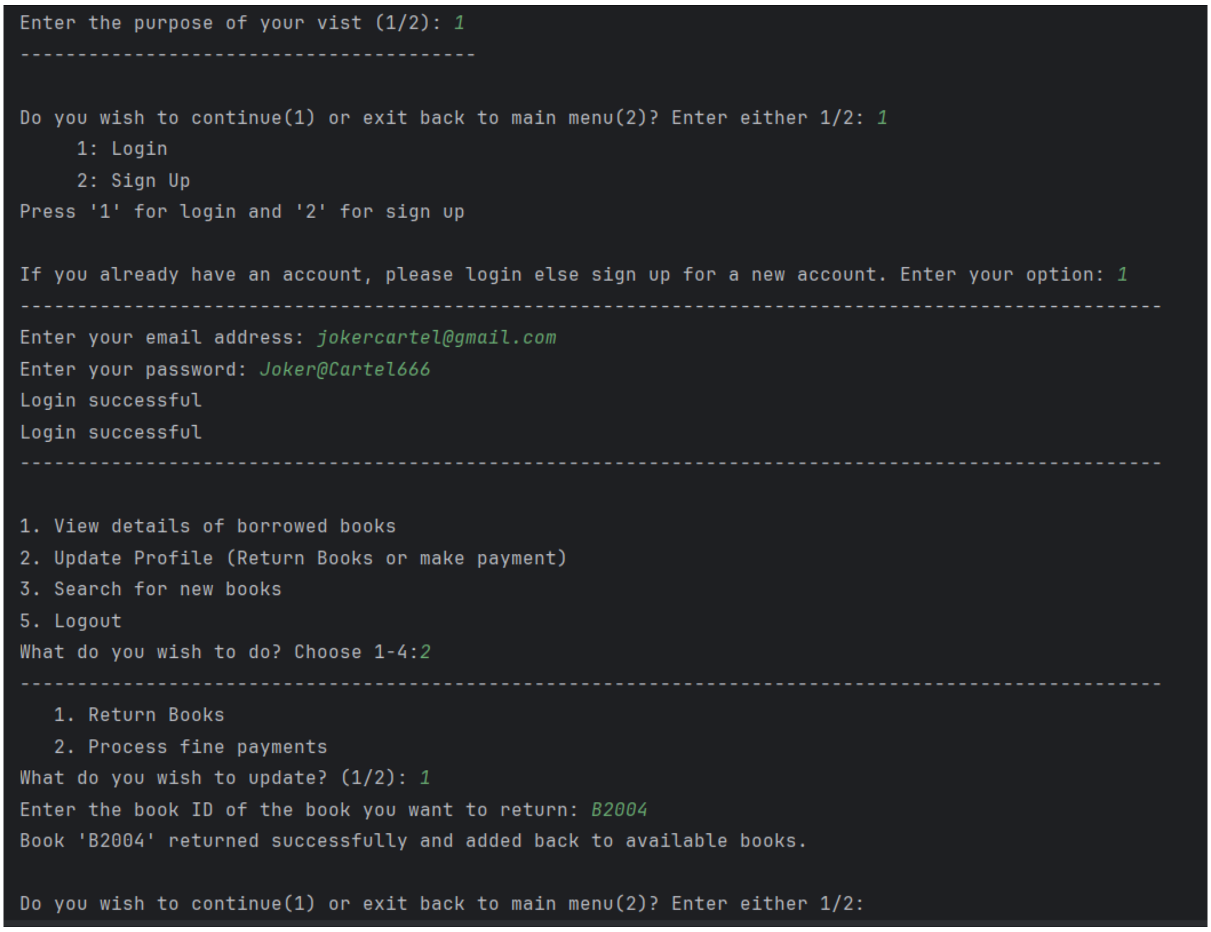
```
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): 1
----------------------------------------

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 2
----------------------------------------------------------------------------------------------------
Enter your Firstname: janes
Enter your Lastname: jones
Enter your Email address: jane3@gmail.com
This account already exists. Try logging in!
Enter your password to log in: sdf
Incorrect email address or password! 2 attempt(s) left.
Enter your email address: jane3@gmail.com
Enter your Password: janes
Incorrect email address or password! 1 attempt(s) left.
Enter your email address: jane3@gmail.com
Enter your Password: janes
Login failed after 3 attempts. Contact librarian.
Do you want to continue browsing Brickfields KL library? (yes/no):
```

An existing member was trying to sign up. After entering her email address that was already in the system, she was prompted to login, instead of signing in. Here, login has failed after 3 attempts.



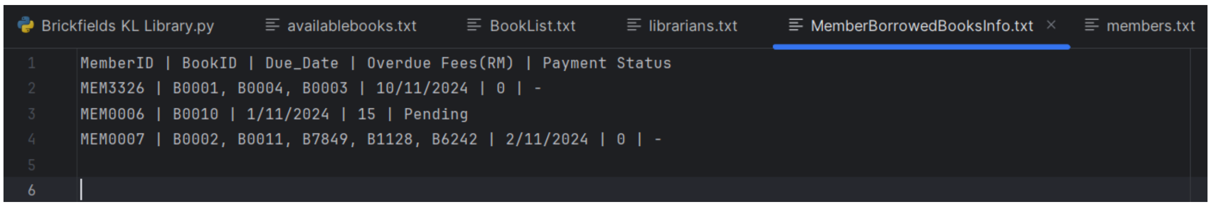
```
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): 1
----------------------------------------

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 2
----------------------------------------------------------------------------------------------------
Enter your Firstname: janes
Enter your Lastname: jones
Enter your Email address: jane3@gmail.com
This account already exists. Try logging in!
Enter your password to log in: Jane1234!
Login successful
----------------------------------------------------------------------------------------------------

1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
```

Here Janes managed to login successfully, instead of signing up and she has entered the member menu.

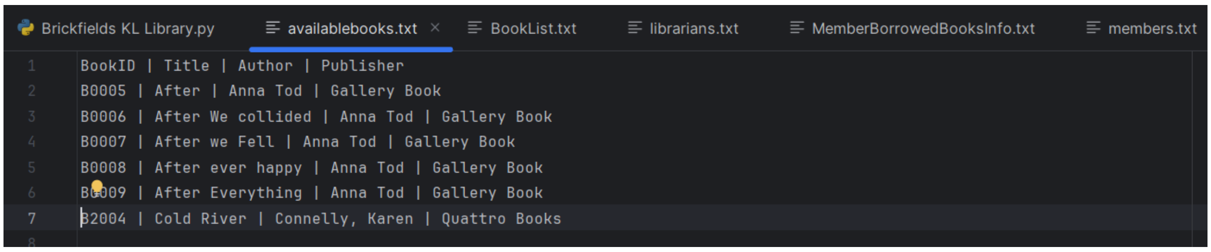


```
Enter the purpose of your vist (1/2): 1
----------------------------------------

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 1
----------------------------------------------------------------------------------------------------
Enter your email address: jokercartel@gmail.com
Enter your password: Joker@Cartel666
Login successful
Login successful
----------------------------------------------------------------------------------------------------

1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:2
----------------------------------------------------------------------------------------------------
   1. Return Books
   2. Process fine payments
What do you wish to update? (1/2): 1
Enter the book ID of the book you want to return: B2004
Book 'B2004' returned successfully and added back to available books.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```

Joker wanted to return his book. He successfully returned the book with Book ID, B2004.

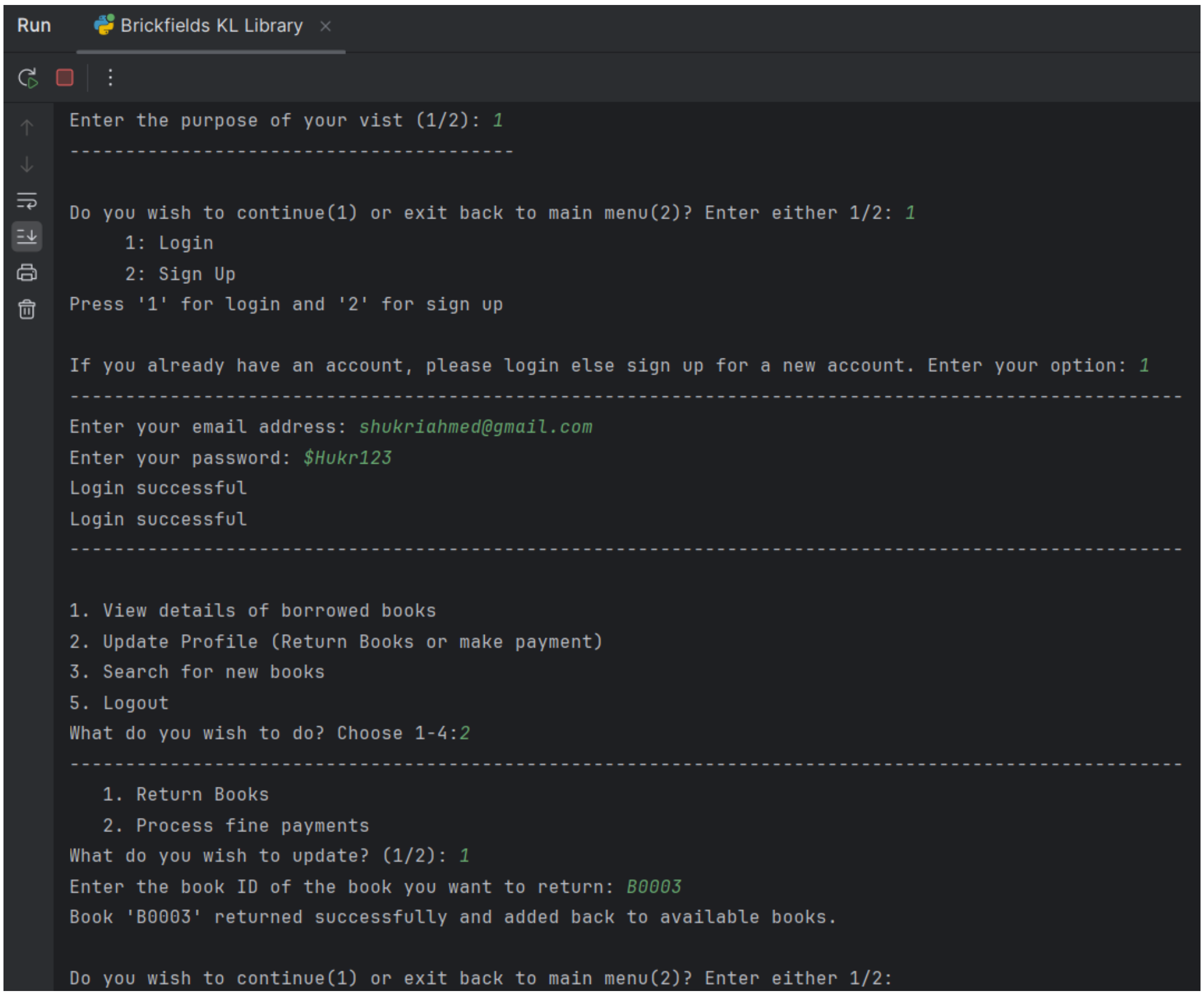


```
MemberID | BookID | Due_Date | Overdue Fees(RM) | Payment Status
MEM3326 | B0001, B0004, B0003 | 10/11/2024 | 0 | -
MEM0006 | B0010 | 1/11/2024 | 15 | Pending
MEM0007 | B0002, B0011, B7849, B1128, B6242 | 2/11/2024 | 0 | -
```



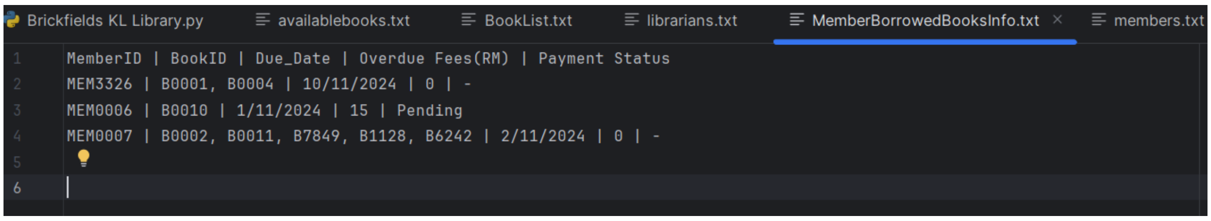
```
BookID | Title | Author | Publisher
B0005 | After | Anna Tod | Gallery Book
B0006 | After We collided | Anna Tod | Gallery Book
B0007 | After we Fell | Anna Tod | Gallery Book
B0008 | After ever happy | Anna Tod | Gallery Book
B0009 | After Everything | Anna Tod | Gallery Book
B2004 | Cold River | Connelly, Karen | Quattro Books
```

His data was deleted in the MemberBorrowedBooksInfo.txt and the book was made available again in availablebooks.txt.

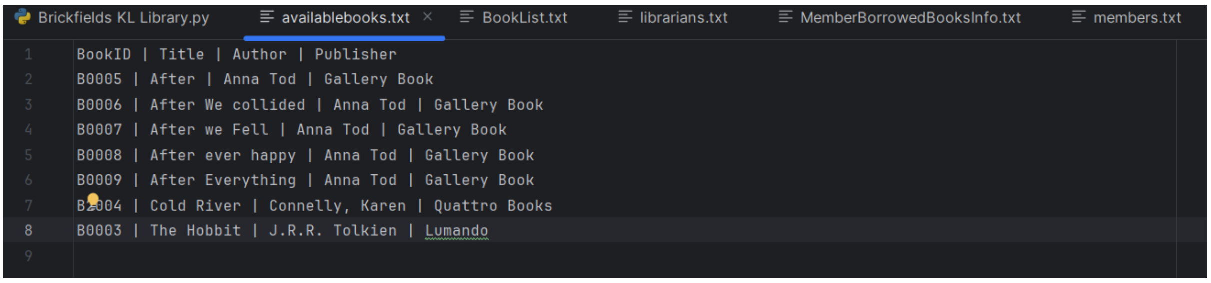


```
Enter the purpose of your vist (1/2): 1
----------------------------------------

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 1
----------------------------------------------------------------------------------------------------
Enter your email address: shukriahmed@gmail.com
Enter your password: $Hukr123
Login successful
Login successful
----------------------------------------------------------------------------------------------------

1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:2
----------------------------------------------------------------------------------------------------
   1. Return Books
   2. Process fine payments
What do you wish to update? (1/2): 1
Enter the book ID of the book you want to return: B0003
Book 'B0003' returned successfully and added back to available books.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```

Shukri wanted to return her book, with Book ID, “B0003”

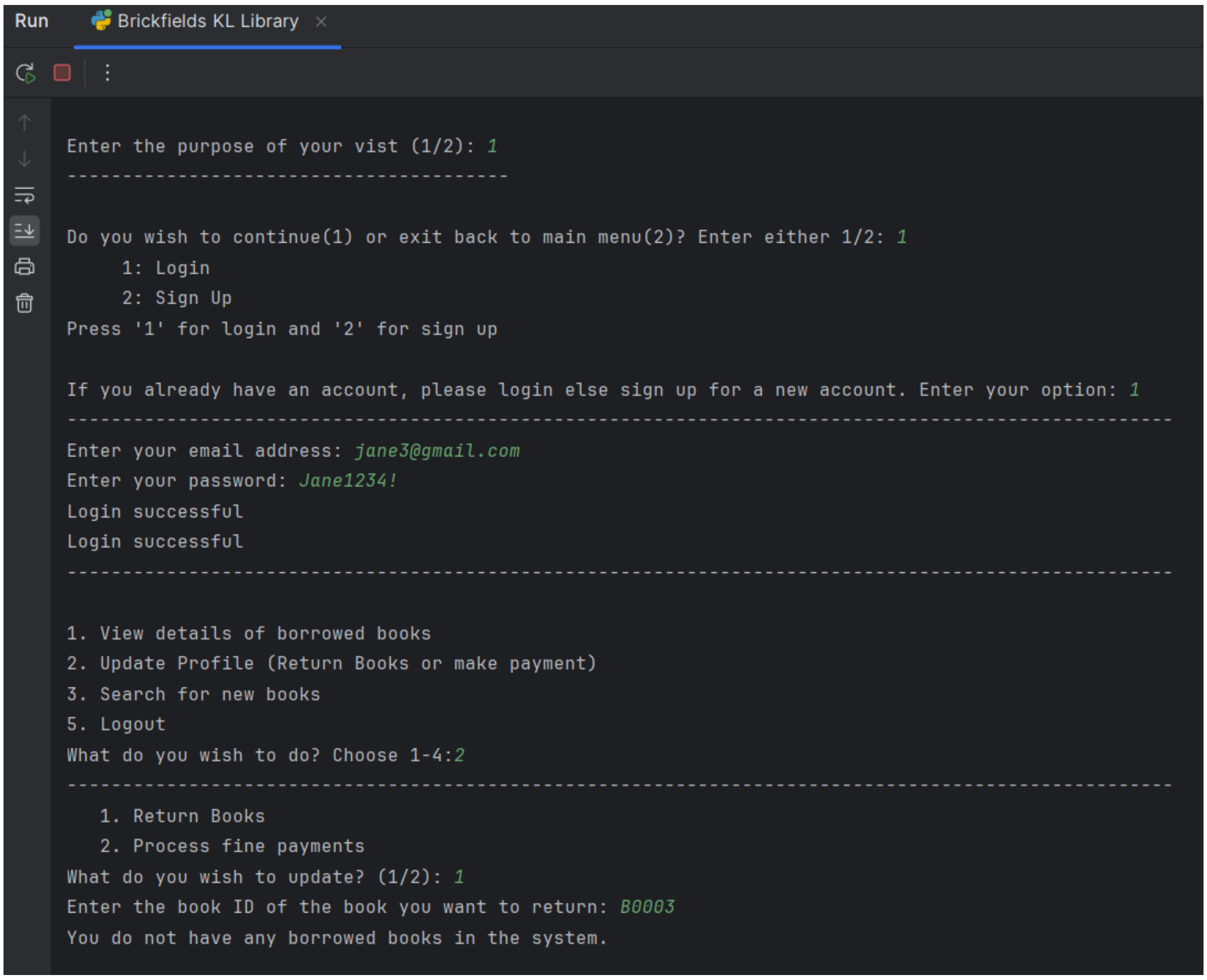


```
MemberID | BookID | Due_Date | Overdue Fees(RM) | Payment Status
MEM3326 | B0001, B0004 | 10/11/2024 | 0 | -
MEM0006 | B0010 | 1/11/2024 | 15 | Pending
MEM0007 | B0002, B0011, B7849, B1128, B6242 | 2/11/2024 | 0 | -
```



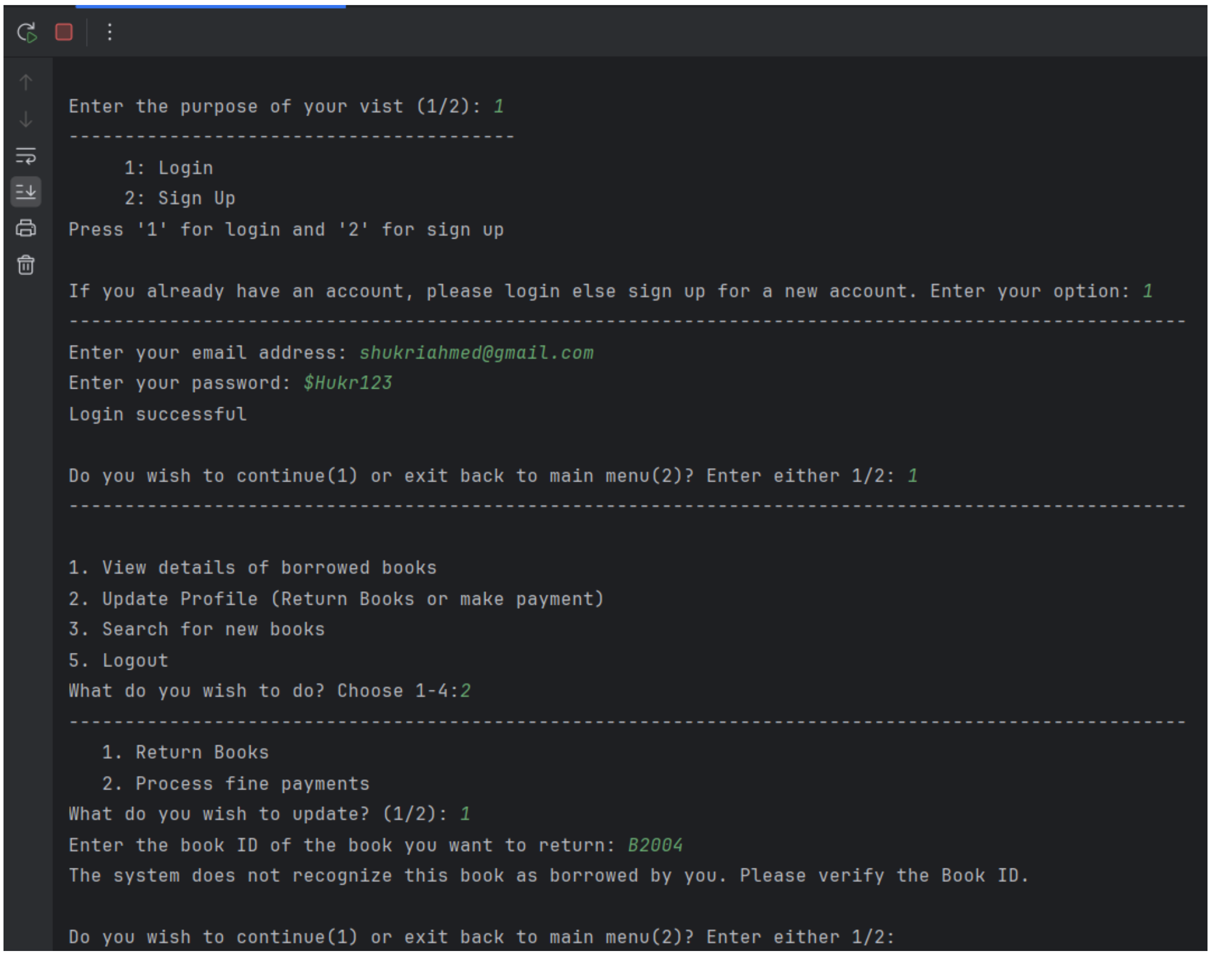
```
BookID | Title | Author | Publisher
B0005 | After | Anna Tod | Gallery Book
B0006 | After We collided | Anna Tod | Gallery Book
B0007 | After we Fell | Anna Tod | Gallery Book
B0008 | After ever happy | Anna Tod | Gallery Book
B0009 | After Everything | Anna Tod | Gallery Book
B0004 | Cold River | Connelly, Karen | Quattro Books
B0003 | The Hobbit | J.R.R. Tolkien | Lumando
```

Since she had borrowed more than one book, only B003 is eliminated in her record. The book is made available again in availablebooks.txt.

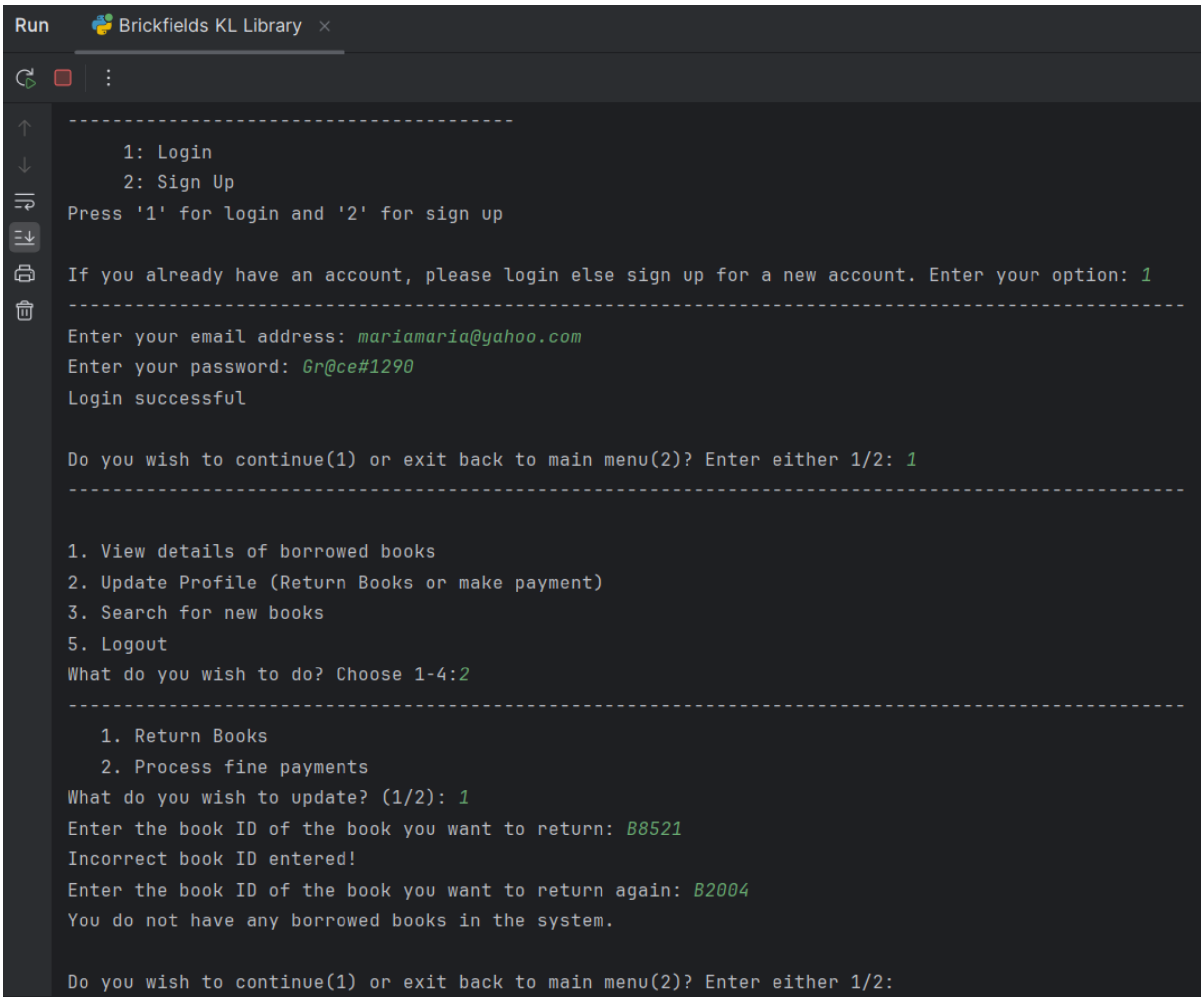


```
Enter the purpose of your vist (1/2): 1
----------------------------------------

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 1
----------------------------------------------------------------------------------------------------
Enter your email address: jane3@gmail.com
Enter your password: Jane1234!
Login successful
Login successful
----------------------------------------------------------------------------------------------------

1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:2
----------------------------------------------------------------------------------------------------
   1. Return Books
   2. Process fine payments
What do you wish to update? (1/2): 1
Enter the book ID of the book you want to return: B0003
You do not have any borrowed books in the system.
```

Janes did not have any books borrowed, hence she could not return the book, ‘B0003’.

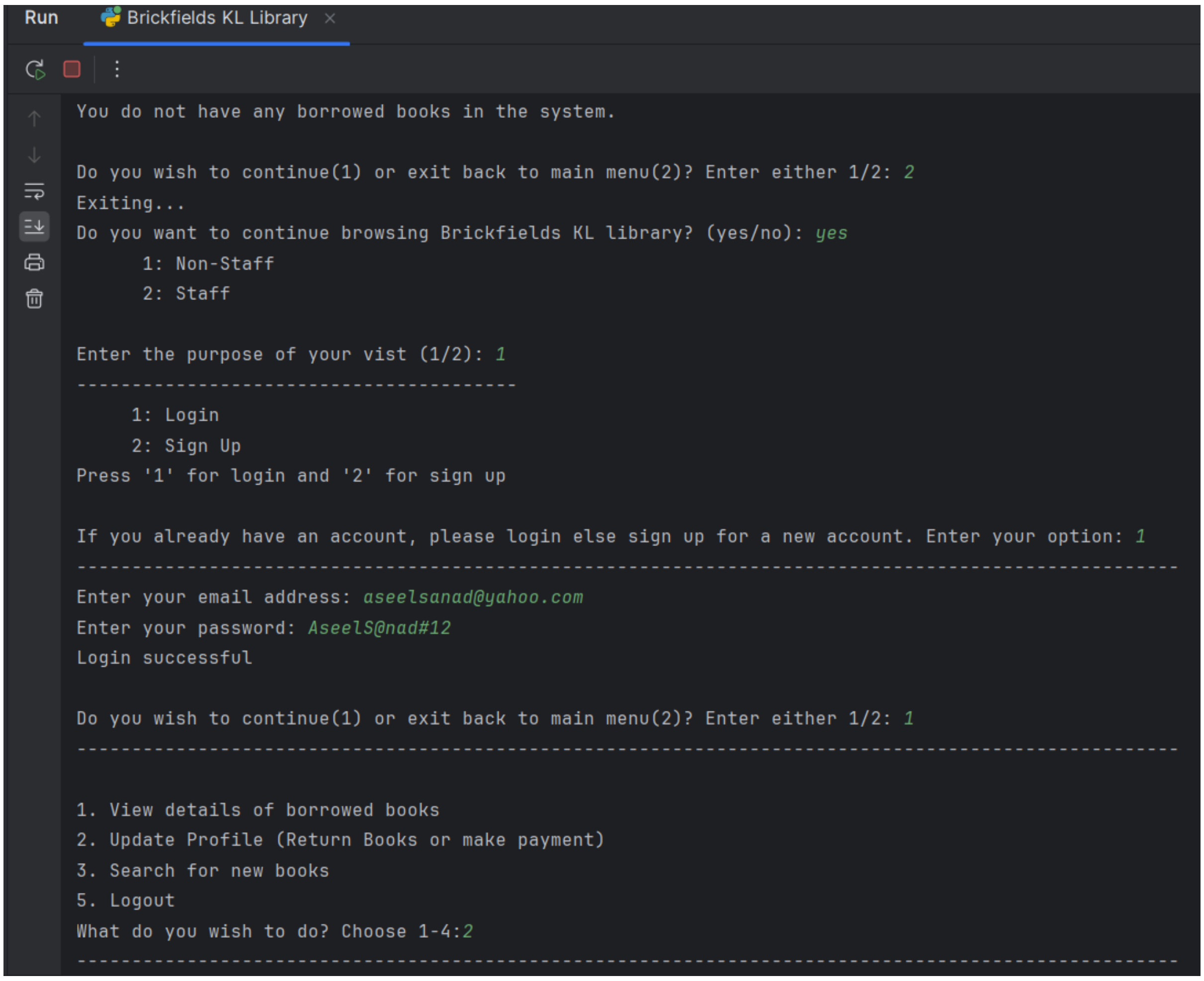


```
Enter the purpose of your vist (1/2): 1
----------------------------------------
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 1
----------------------------------------------------------------------------------------------------
Enter your email address: shukriahmed@gmail.com
Enter your password: $Hukr123
Login successful

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
----------------------------------------------------------------------------------------------------


1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:2
----------------------------------------------------------------------------------------------------
   1. Return Books
   2. Process fine payments
What do you wish to update? (1/2): 1
Enter the book ID of the book you want to return: B2004
The system does not recognize this book as borrowed by you. Please verify the Book ID.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```



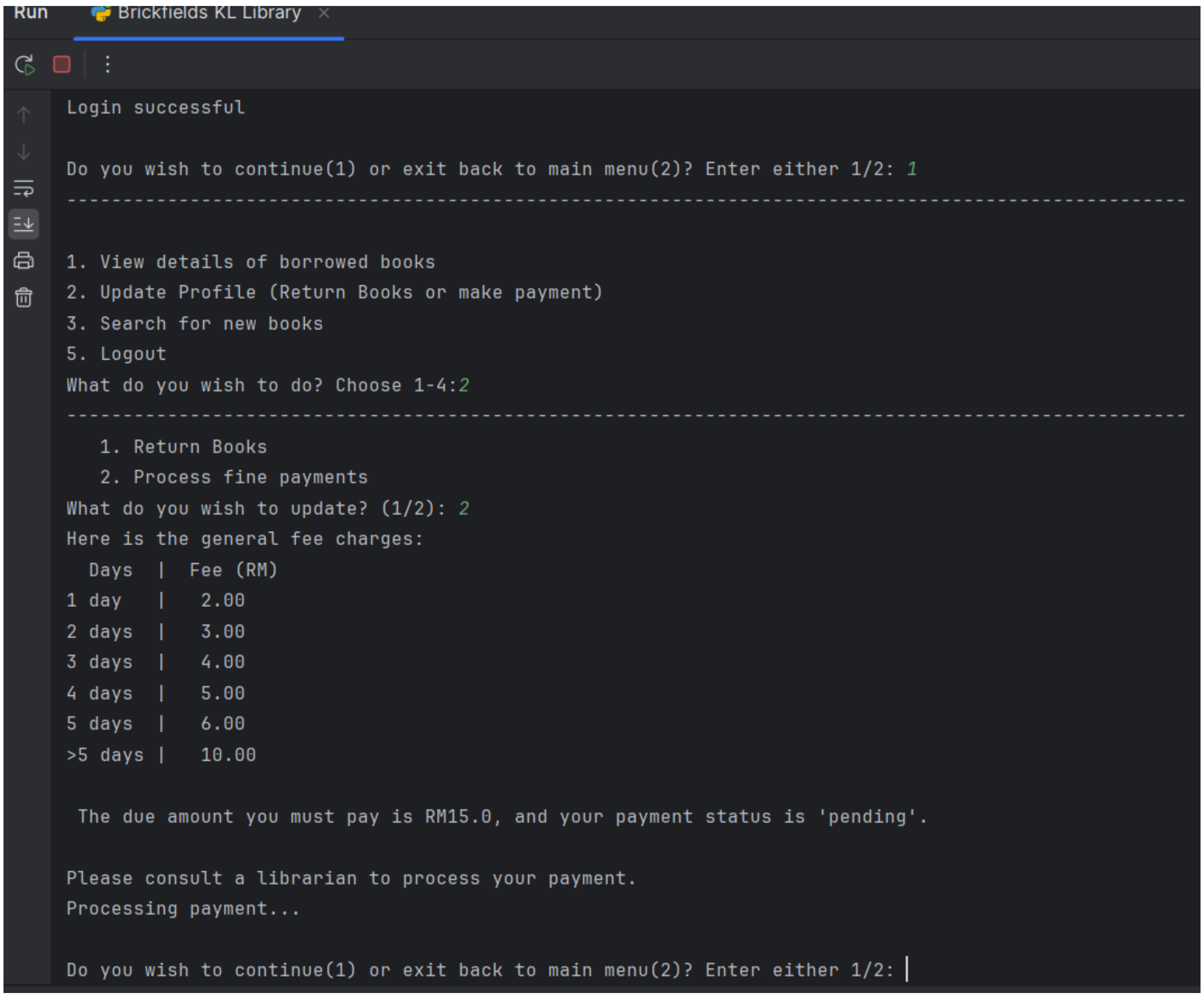
```
----------------------------------------
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 1
----------------------------------------------------------------------------------------------------
Enter your email address: mariamaria@yahoo.com
Enter your password: Gr@ce#1290
Login successful

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
----------------------------------------------------------------------------------------------------


1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:2
----------------------------------------------------------------------------------------------------
   1. Return Books
   2. Process fine payments
What do you wish to update? (1/2): 1
Enter the book ID of the book you want to return: B8521
Incorrect book ID entered!
Enter the book ID of the book you want to return again: B2004
You do not have any borrowed books in the system.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```

A book with Book ID B8521 did not exist in the system, that’s why an error message was displayed. Grace did not have any borrowed book, and so, she could not return the book with book ID, B2004.



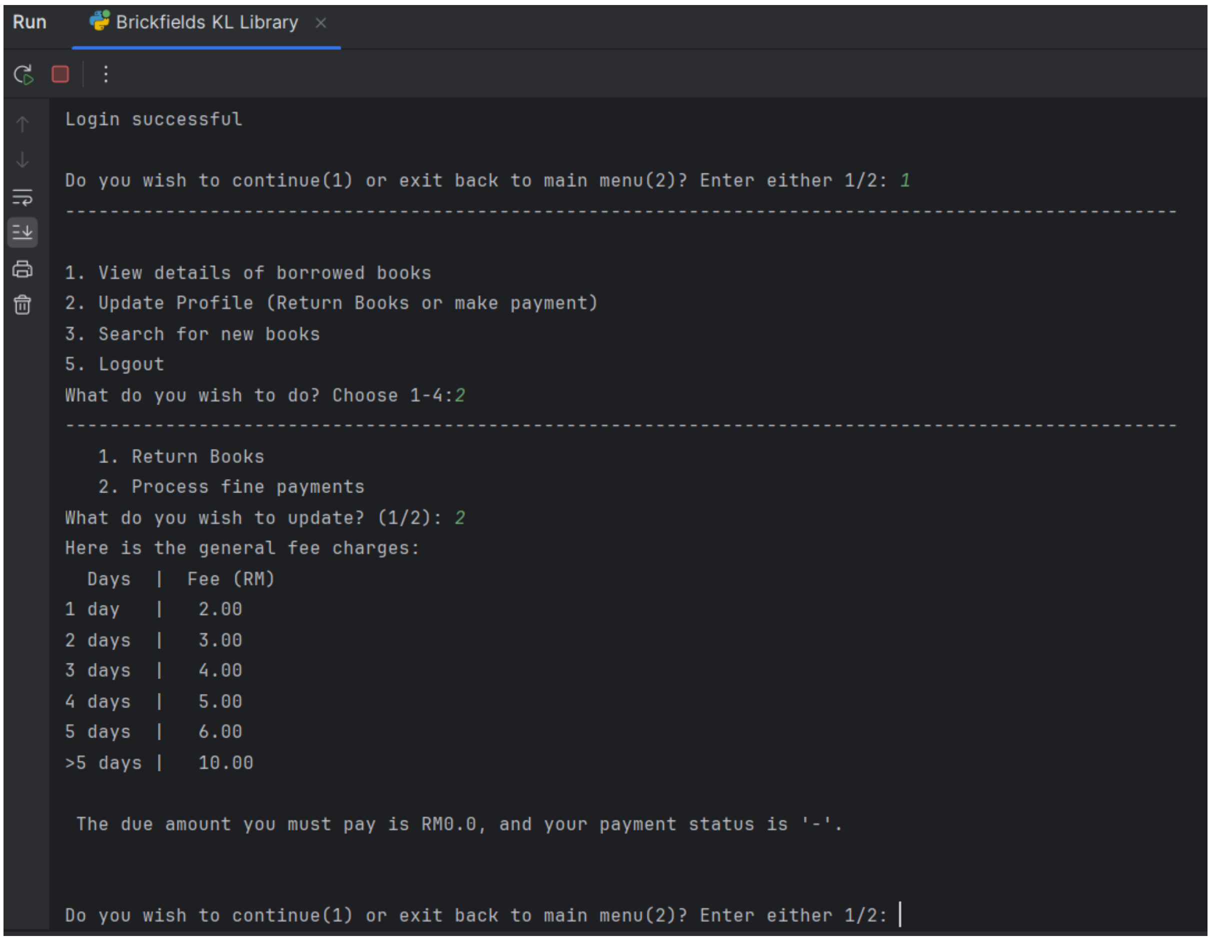
```
You do not have any borrowed books in the system.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 2
Exiting...
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): 1
----------------------------------------
     1: Login
     2: Sign Up
Press '1' for login and '2' for sign up

If you already have an account, please login else sign up for a new account. Enter your option: 1
----------------------------------------------------------------------------------------------------
Enter your email address: aseelsanad@yahoo.com
Enter your password: AseelS@nad#12
Login successful

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
----------------------------------------------------------------------------------------------------


1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:2
----------------------------------------------------------------------------------------------------
```

Aseel opted to update her profile.



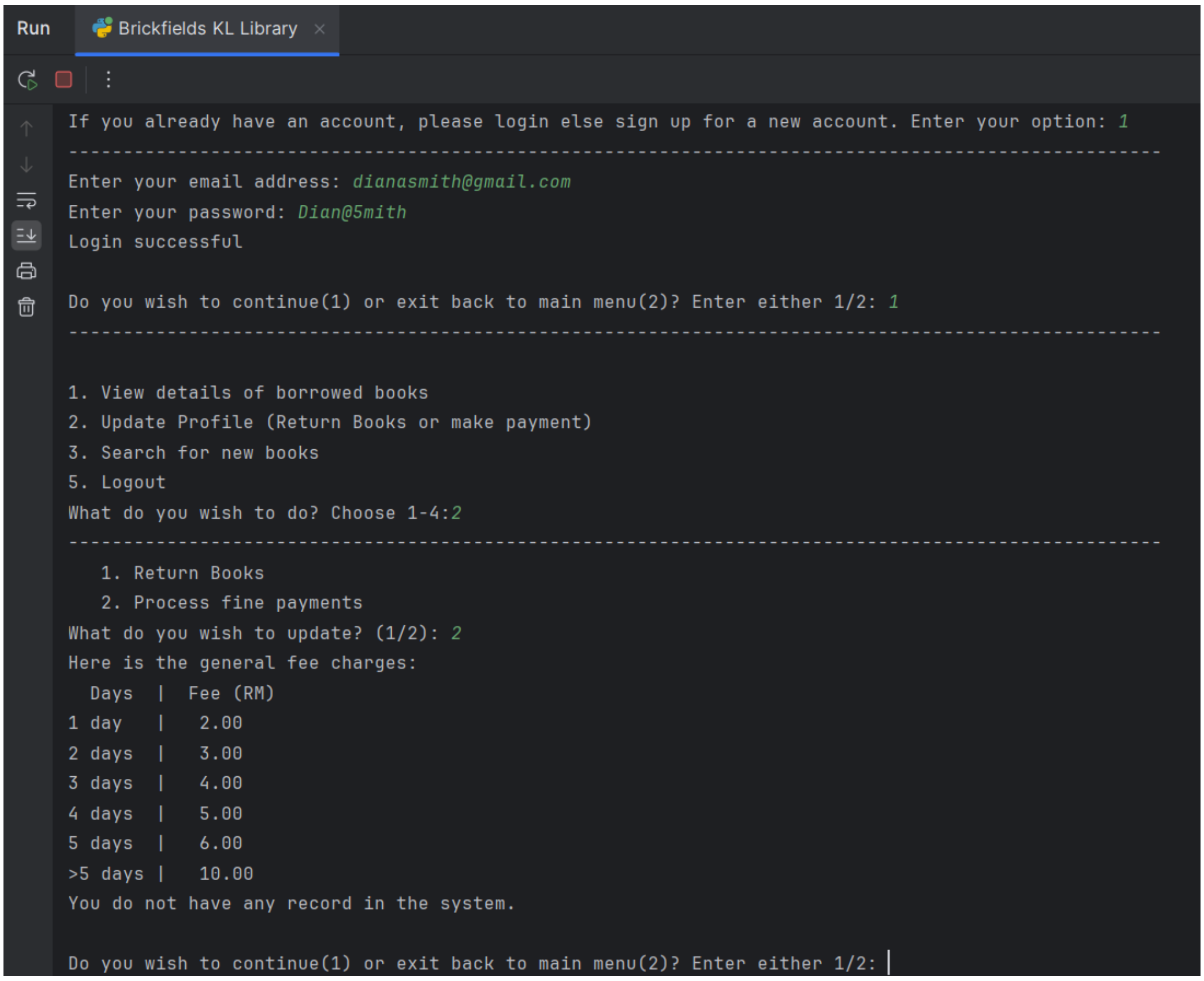
```
Login successful

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
----------------------------------------------------------------------------------------------------


1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:2
----------------------------------------------------------------------------------------------------
   1. Return Books
   2. Process fine payments
What do you wish to update? (1/2): 2
Here is the general fee charges:
  Days   |  Fee (RM)
1 day    |   2.00
2 days   |   3.00
3 days   |   4.00
4 days   |   5.00
5 days   |   6.00
>5 days |   10.00

 The due amount you must pay is RM15.0, and your payment status is 'pending'.

Please consult a librarian to process your payment.
Processing payment...

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 
```

She had a pending overdue fee of RM 15. The payment would be processed after the librarian and administrator’s intervention.



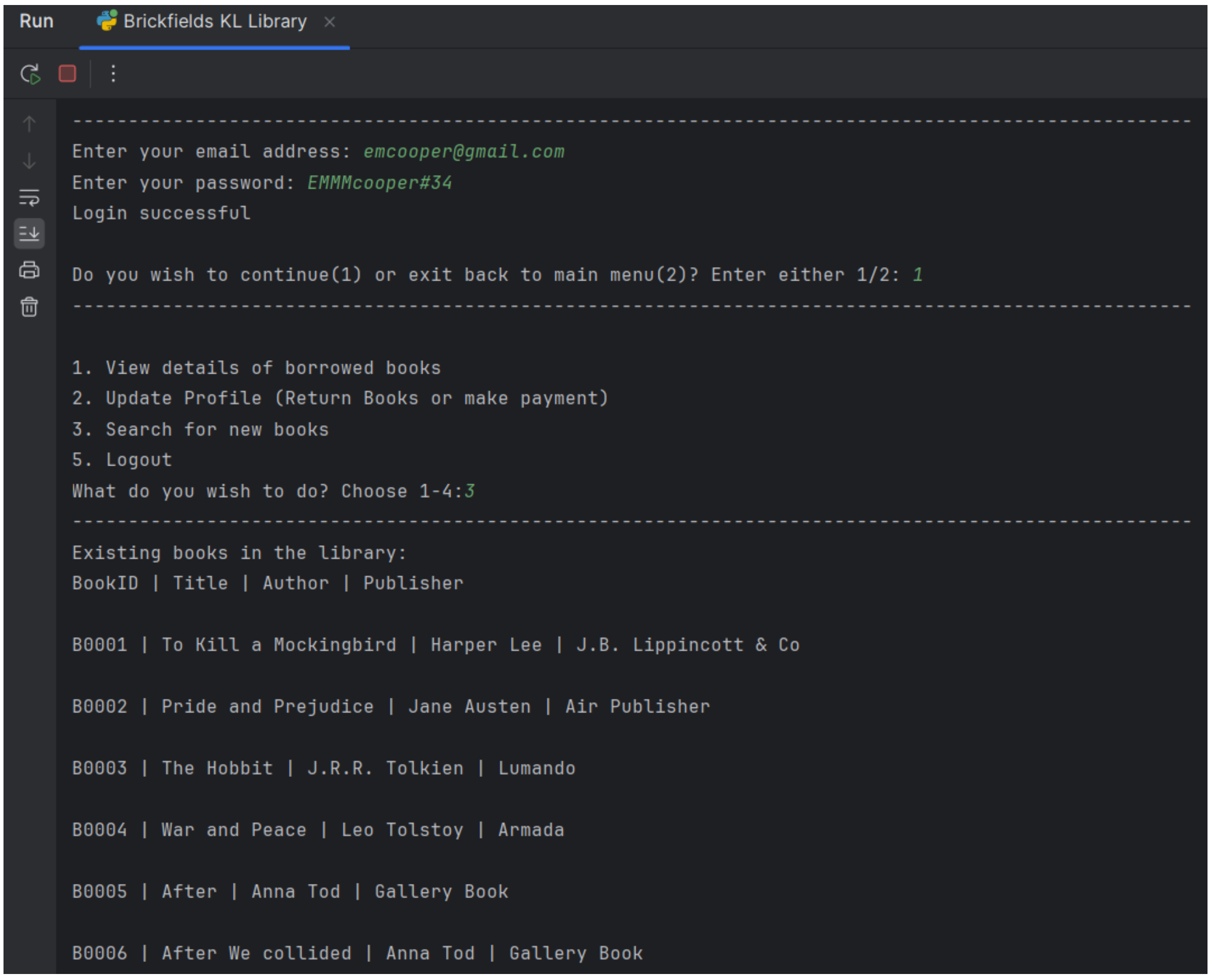
```
Login successful

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
---------------------------------------------------------------------------------------------------


1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:2
---------------------------------------------------------------------------------------------------
   1. Return Books
   2. Process fine payments
What do you wish to update? (1/2): 2
Here is the general fee charges:
  Days   |  Fee (RM)
1 day    |   2.00
2 days   |   3.00
3 days   |   4.00
4 days   |   5.00
5 days   |   6.00
>5 days  |   10.00

 The due amount you must pay is RM0.0, and your payment status is '-'.


Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 
```

Another user had 0 overdue fee.



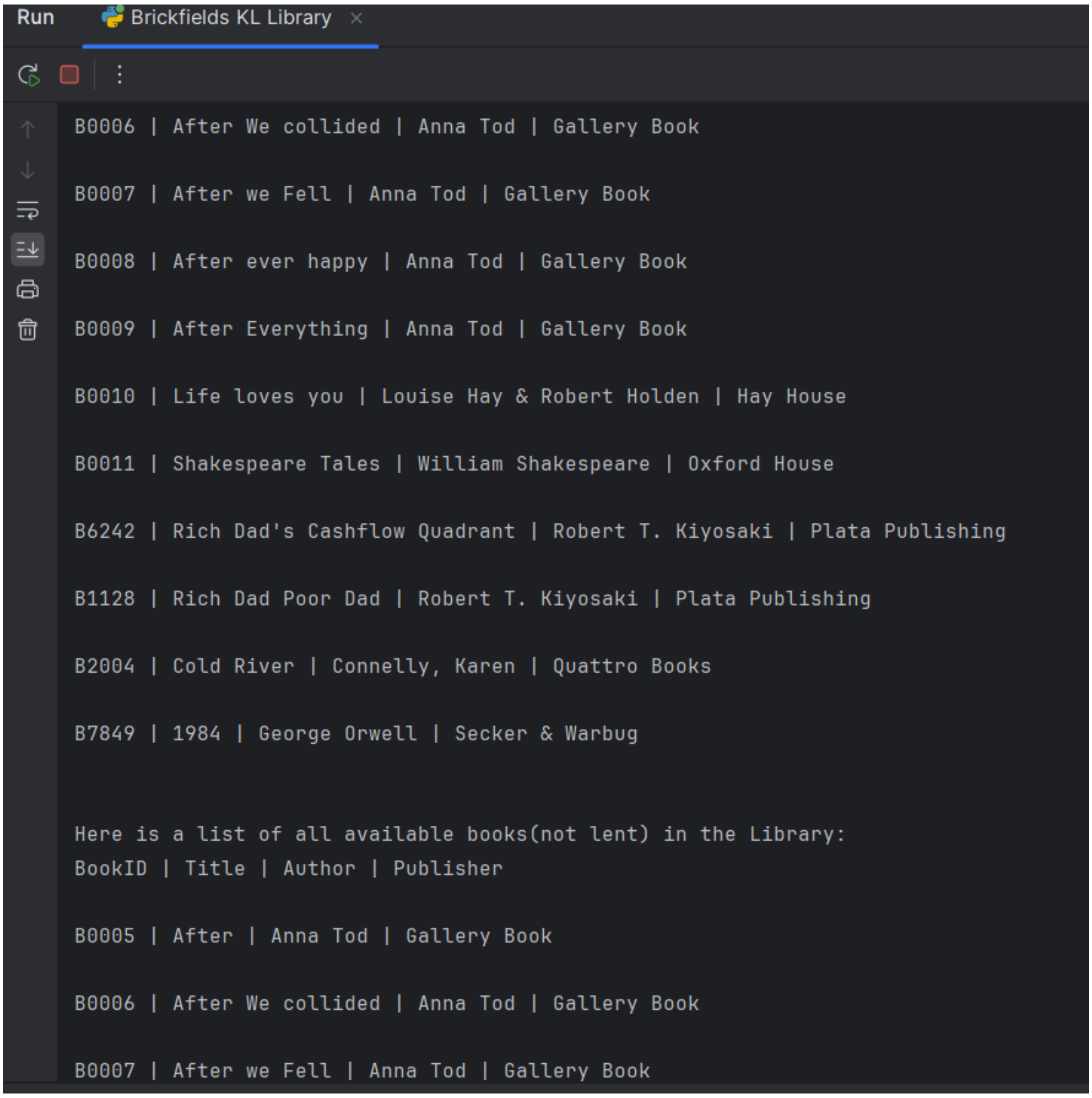
```
If you already have an account, please login else sign up for a new account. Enter your option: 1
-----------------------------------------------------------------------------------------------------
Enter your email address: dianasmith@gmail.com
Enter your password: Dian@5mith
Login successful

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
-----------------------------------------------------------------------------------------------------


1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:2
-----------------------------------------------------------------------------------------------------
   1. Return Books
   2. Process fine payments
What do you wish to update? (1/2): 2
Here is the general fee charges:
  Days   |  Fee (RM)
1 day    |   2.00
2 days   |   3.00
3 days   |   4.00
4 days   |   5.00
5 days   |   6.00
>5 days  |   10.00
You do not have any record in the system.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```

Diana did not have any borrowed book record in the system, and thus, no payment to be processed.

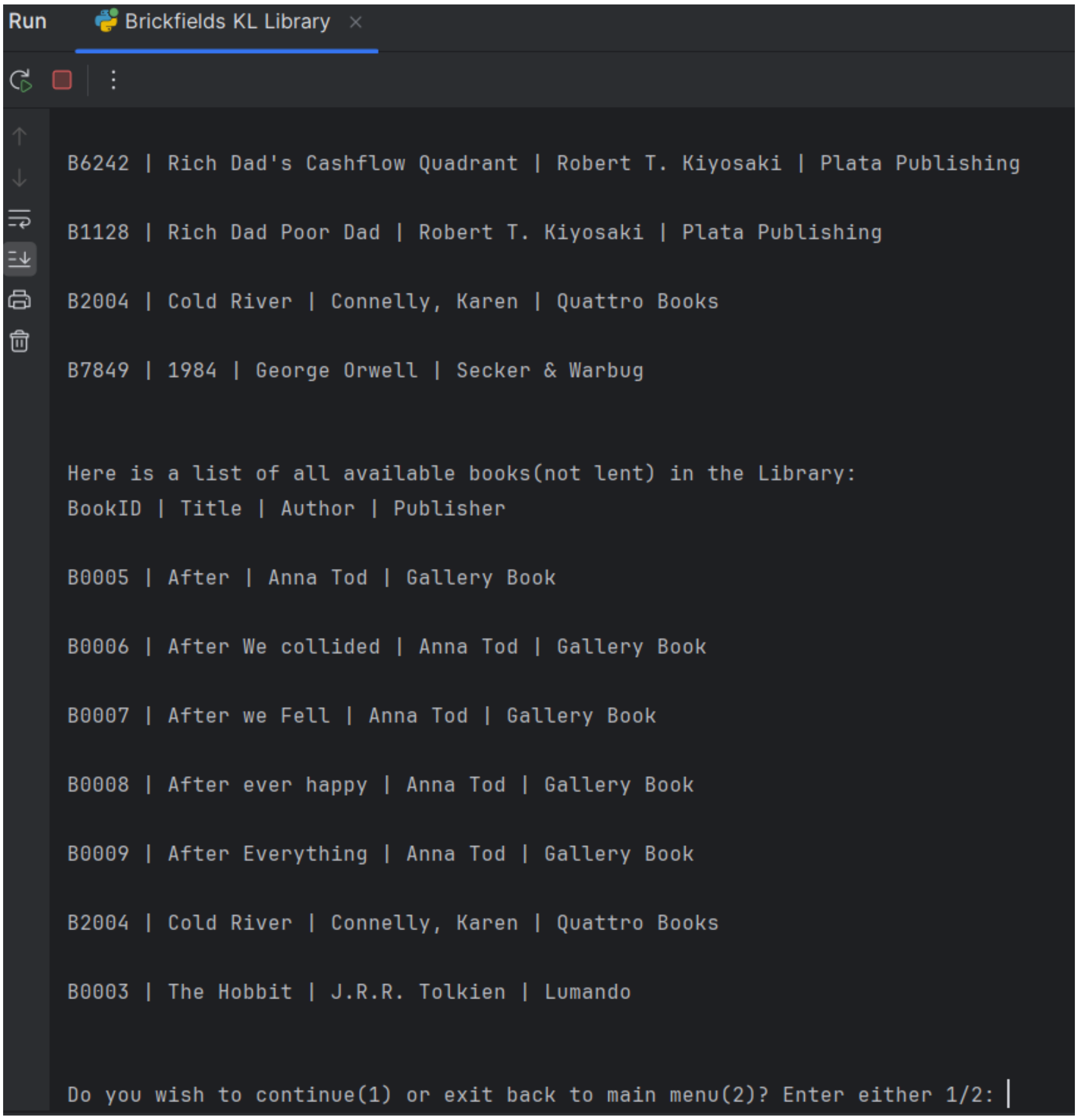


```
----------------------------------------------------------------------------------------------------
Enter your email address: emcooper@gmail.com
Enter your password: EMMMcooper#34
Login successful

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
----------------------------------------------------------------------------------------------------

1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:3
----------------------------------------------------------------------------------------------------
Existing books in the library:
BookID | Title | Author | Publisher

B0001 | To Kill a Mockingbird | Harper Lee | J.B. Lippincott & Co

B0002 | Pride and Prejudice | Jane Austen | Air Publisher

B0003 | The Hobbit | J.R.R. Tolkien | Lumando

B0004 | War and Peace | Leo Tolstoy | Armada

B0005 | After | Anna Tod | Gallery Book

B0006 | After We collided | Anna Tod | Gallery Book
```

Emily opt to search for new book.

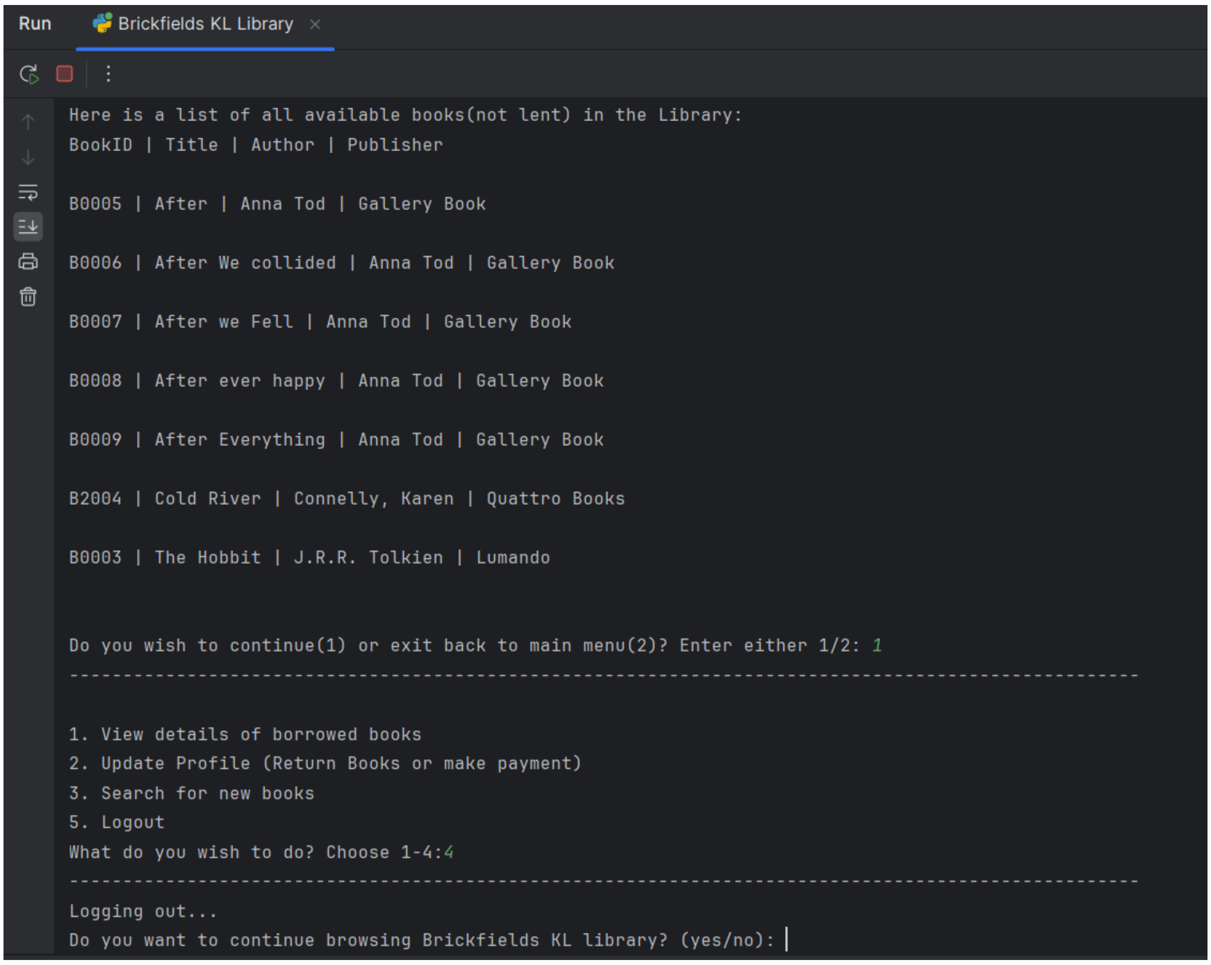


```
B0006 | After We collided | Anna Tod | Gallery Book

B0007 | After we Fell | Anna Tod | Gallery Book

B0008 | After ever happy | Anna Tod | Gallery Book

B0009 | After Everything | Anna Tod | Gallery Book

B0010 | Life loves you | Louise Hay & Robert Holden | Hay House

B0011 | Shakespeare Tales | William Shakespeare | Oxford House

B6242 | Rich Dad's Cashflow Quadrant | Robert T. Kiyosaki | Plata Publishing

B1128 | Rich Dad Poor Dad | Robert T. Kiyosaki | Plata Publishing

B2004 | Cold River | Connelly, Karen | Quattro Books

B7849 | 1984 | George Orwell | Secker & Warbug


Here is a list of all available books(not lent) in the Library:
BookID | Title | Author | Publisher

B0005 | After | Anna Tod | Gallery Book

B0006 | After We collided | Anna Tod | Gallery Book

B0007 | After we Fell | Anna Tod | Gallery Book
```



```
B6242 | Rich Dad's Cashflow Quadrant | Robert T. Kiyosaki | Plata Publishing

B1128 | Rich Dad Poor Dad | Robert T. Kiyosaki | Plata Publishing

B2004 | Cold River | Connelly, Karen | Quattro Books

B7849 | 1984 | George Orwell | Secker & Warbug


Here is a list of all available books(not lent) in the Library:
BookID | Title | Author | Publisher

B0005 | After | Anna Tod | Gallery Book

B0006 | After We collided | Anna Tod | Gallery Book

B0007 | After we Fell | Anna Tod | Gallery Book

B0008 | After ever happy | Anna Tod | Gallery Book

B0009 | After Everything | Anna Tod | Gallery Book

B2004 | Cold River | Connelly, Karen | Quattro Books

B0003 | The Hobbit | J.R.R. Tolkien | Lumando


Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```

Existing books and books available to be borrowed are displayed.

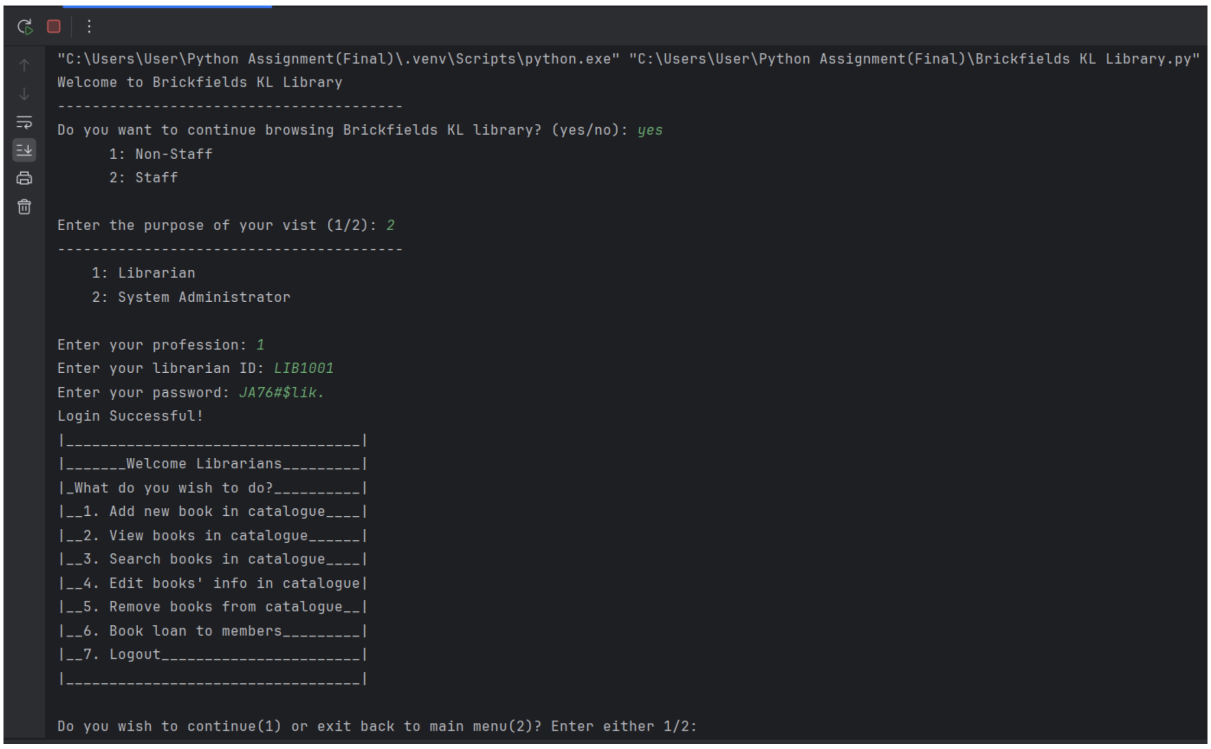


```
Here is a list of all available books(not lent) in the Library:
BookID | Title | Author | Publisher

B0005 | After | Anna Tod | Gallery Book

B0006 | After We collided | Anna Tod | Gallery Book

B0007 | After we Fell | Anna Tod | Gallery Book

B0008 | After ever happy | Anna Tod | Gallery Book

B0009 | After Everything | Anna Tod | Gallery Book

B2004 | Cold River | Connelly, Karen | Quattro Books

B0003 | The Hobbit | J.R.R. Tolkien | Lumando


Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
----------------------------------------------------------------------------------------------------


1. View details of borrowed books
2. Update Profile (Return Books or make payment)
3. Search for new books
5. Logout
What do you wish to do? Choose 1-4:4
----------------------------------------------------------------------------------------------------
Logging out...
Do you want to continue browsing Brickfields KL library? (yes/no):
```

Emily chose to log out.

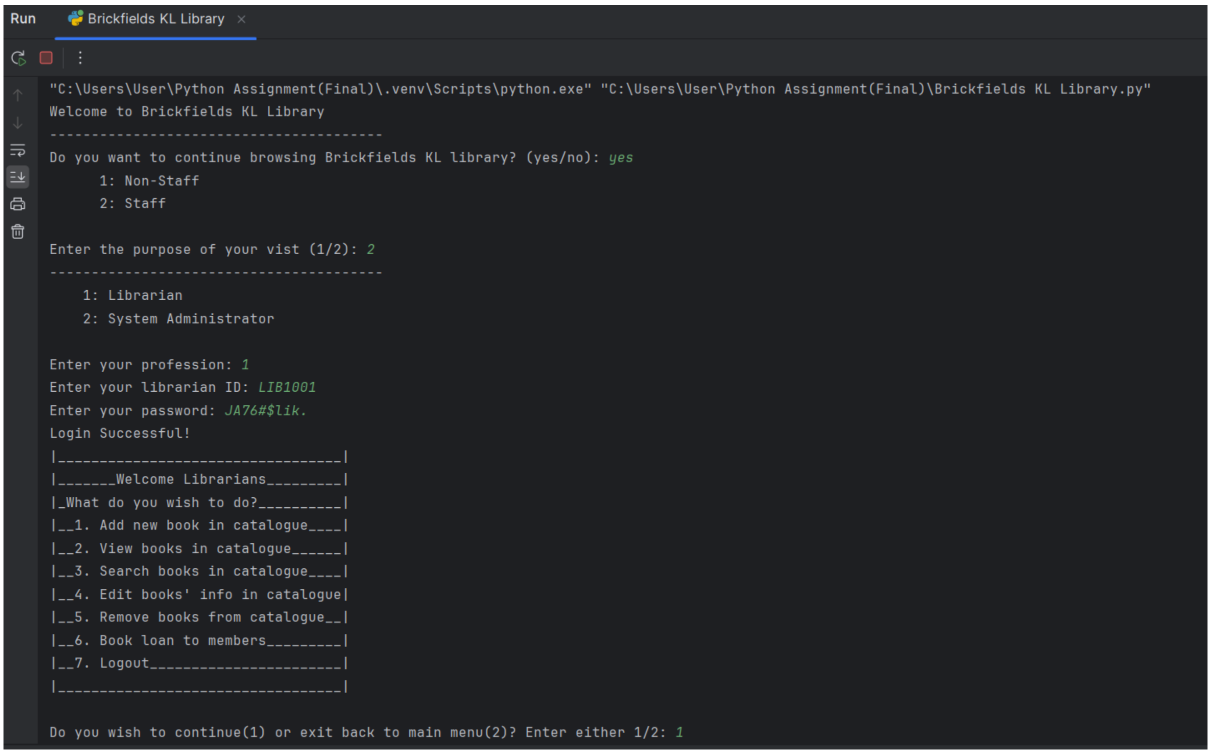
The ‘logging out’ message is printed.

6.2 Librarian



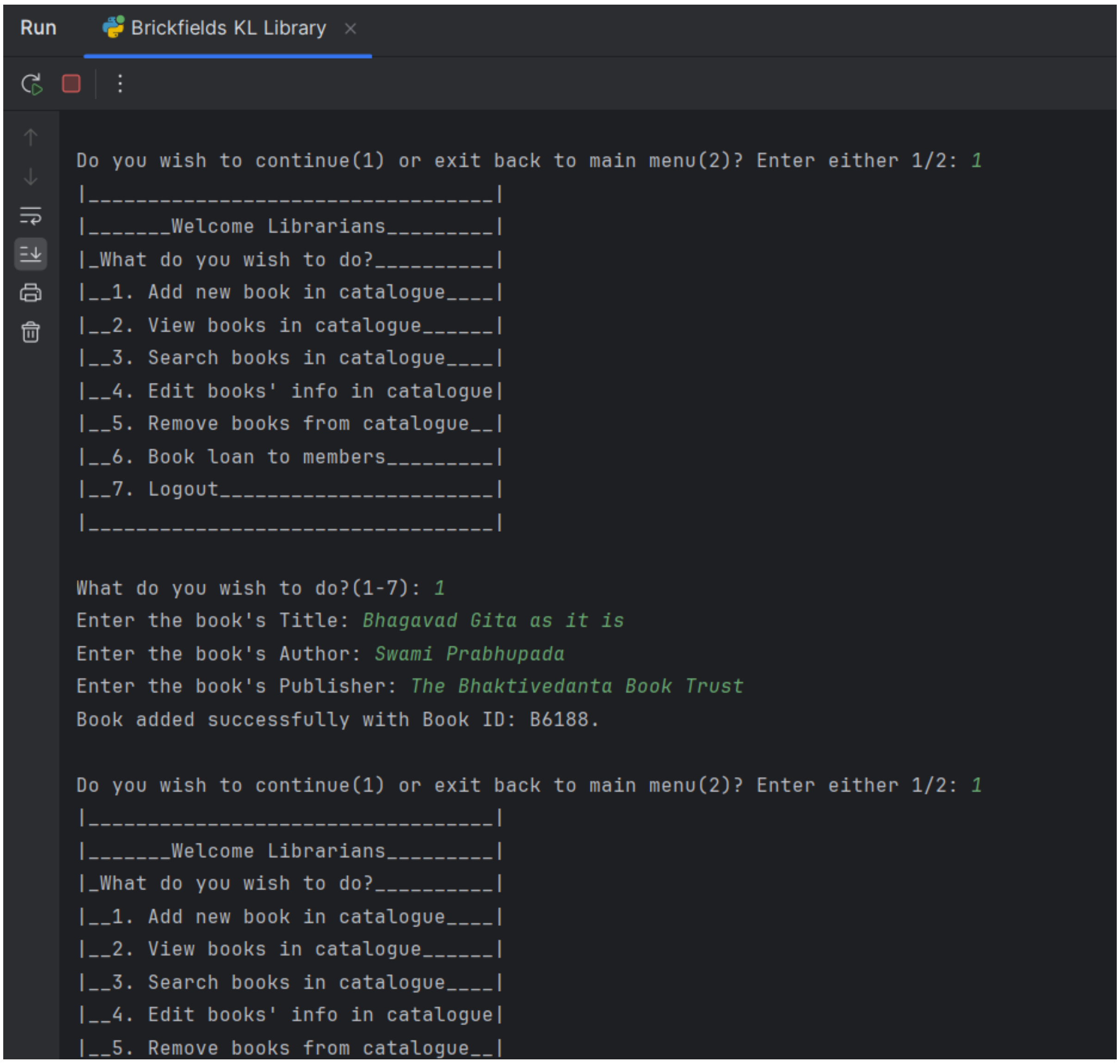
```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): 2
----------------------------------------
    1: Librarian
    2: System Administrator

Enter your profession: 1
Enter your librarian ID: LIB1001
Enter your password: JA76#$lik.
Login Successful!
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```

The librarian needs to input the correct Librarian ID and password to successfully login.



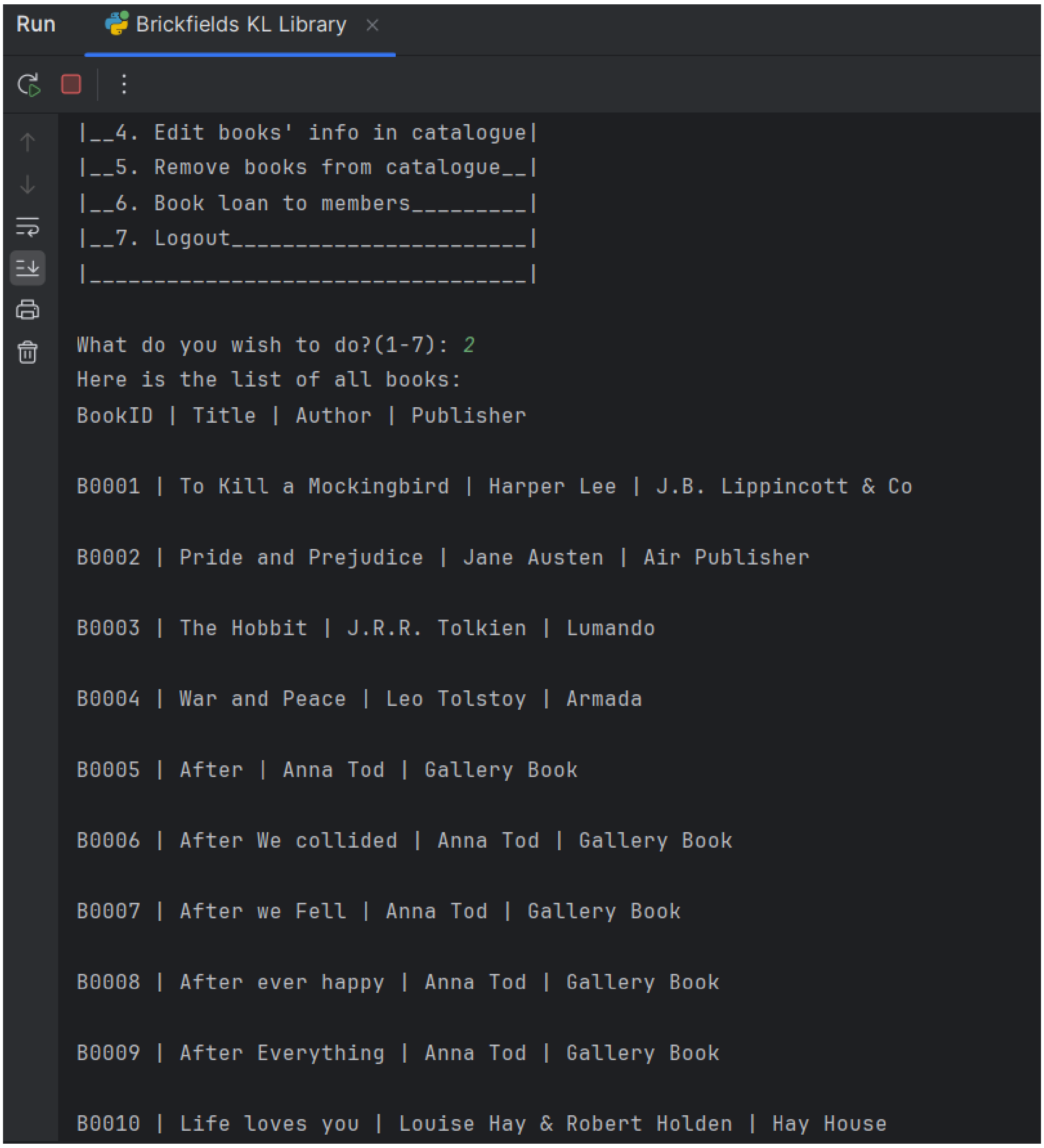
Run   Brickfields KL Library

```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): 2
----------------------------------------
    1: Librarian
    2: System Administrator

Enter your profession: 1
Enter your librarian ID: LIB1001
Enter your password: JA76#$lik.
Login Successful!
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
```



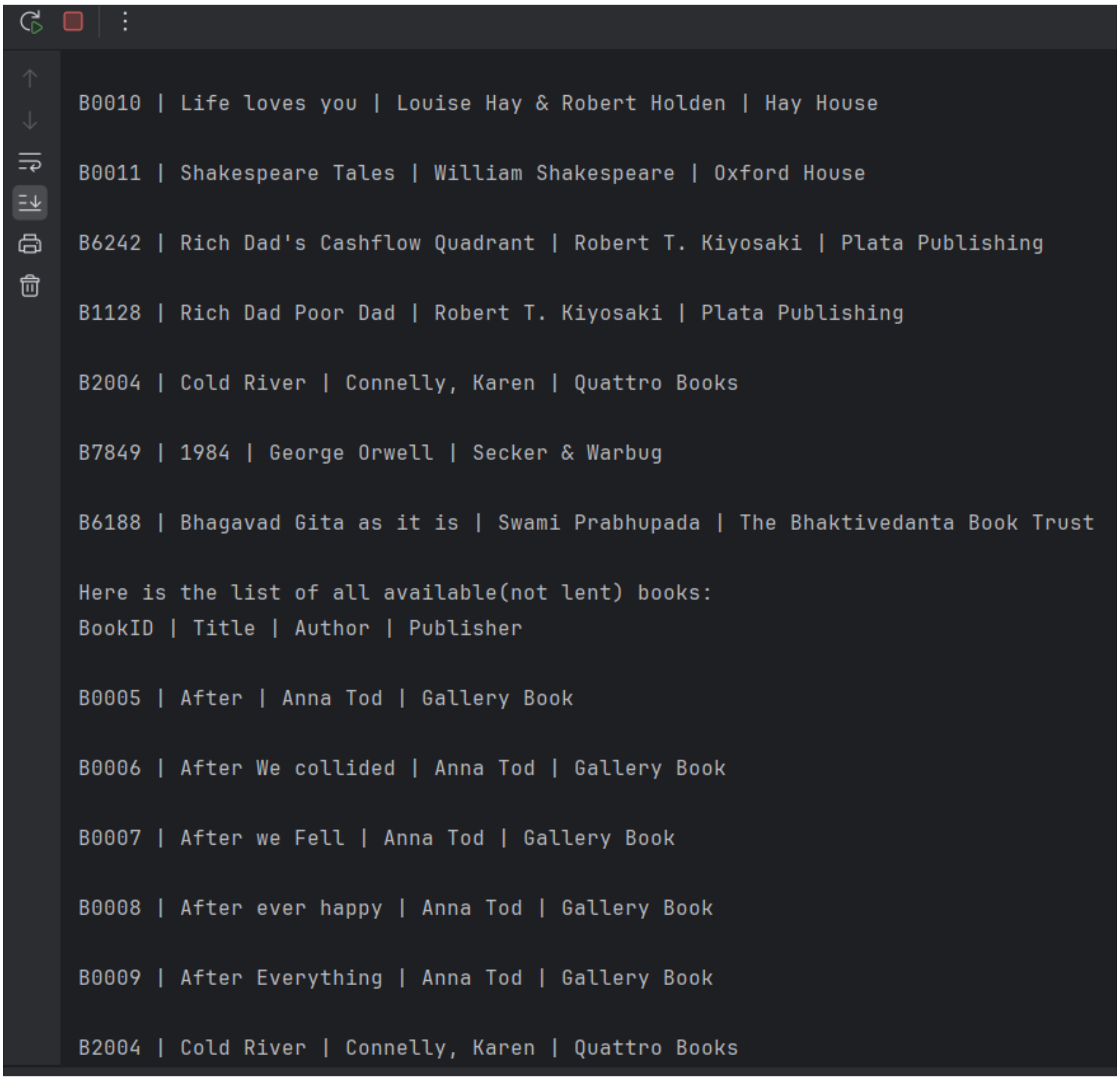
```
Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

What do you wish to do?(1-7): 1
Enter the book's Title: Bhagavad Gita as it is
Enter the book's Author: Swami Prabhupada
Enter the book's Publisher: The Bhaktivedanta Book Trust
Book added successfully with Book ID: B6188.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
```

He chose to enter the librarian menu and add a new book. After entering the book's details, it was appended in the BookList.txt and availablebooks.txt files, with Book ID B6188.



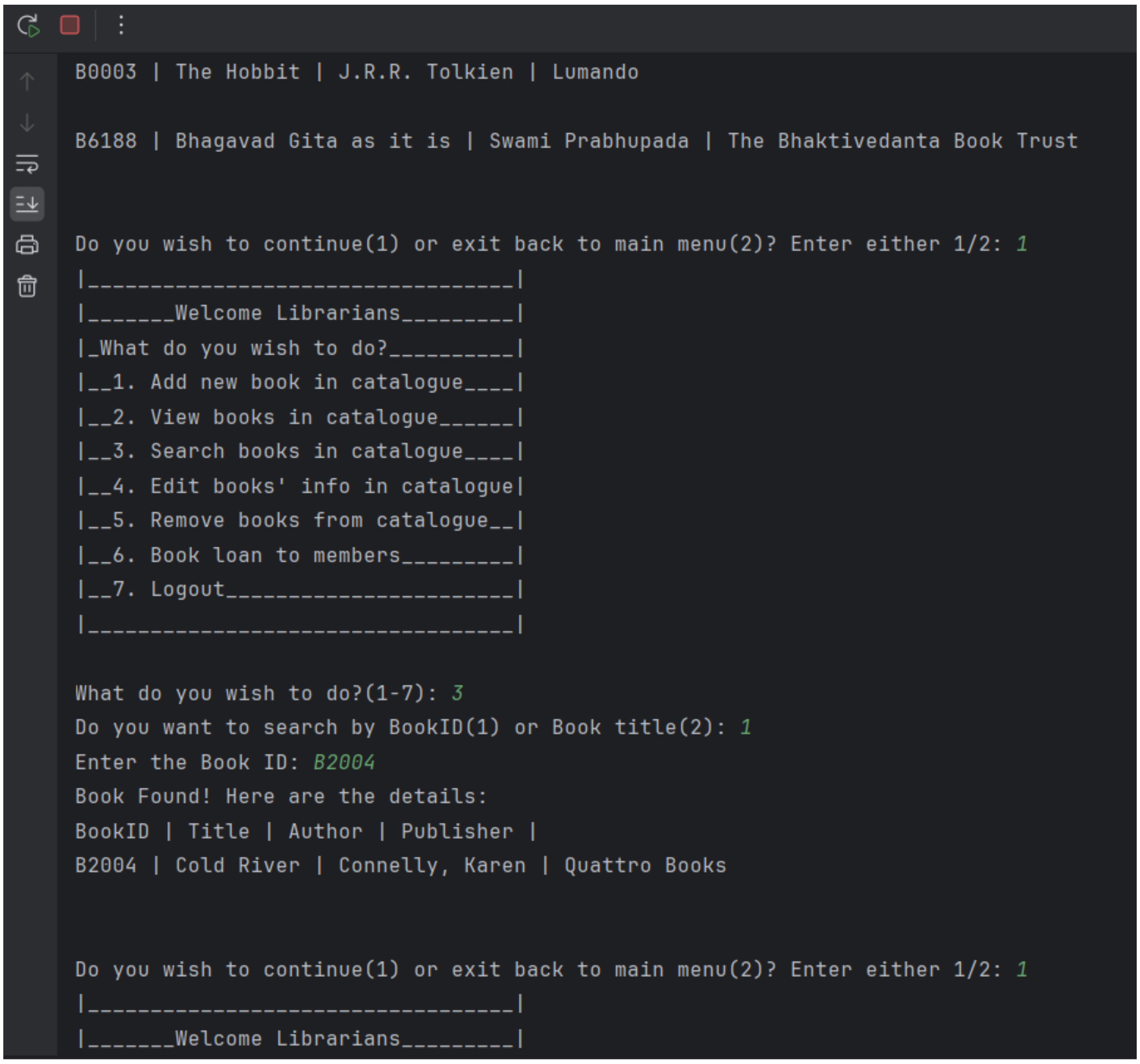
```
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

What do you wish to do?(1-7): 2
Here is the list of all books:
BookID | Title | Author | Publisher

B0001 | To Kill a Mockingbird | Harper Lee | J.B. Lippincott & Co

B0002 | Pride and Prejudice | Jane Austen | Air Publisher

B0003 | The Hobbit | J.R.R. Tolkien | Lumando

B0004 | War and Peace | Leo Tolstoy | Armada

B0005 | After | Anna Tod | Gallery Book

B0006 | After We collided | Anna Tod | Gallery Book

B0007 | After we Fell | Anna Tod | Gallery Book

B0008 | After ever happy | Anna Tod | Gallery Book

B0009 | After Everything | Anna Tod | Gallery Book

B0010 | Life loves you | Louise Hay & Robert Holden | Hay House
```

The librarian can also view the books in the system by entering the 2nd option.



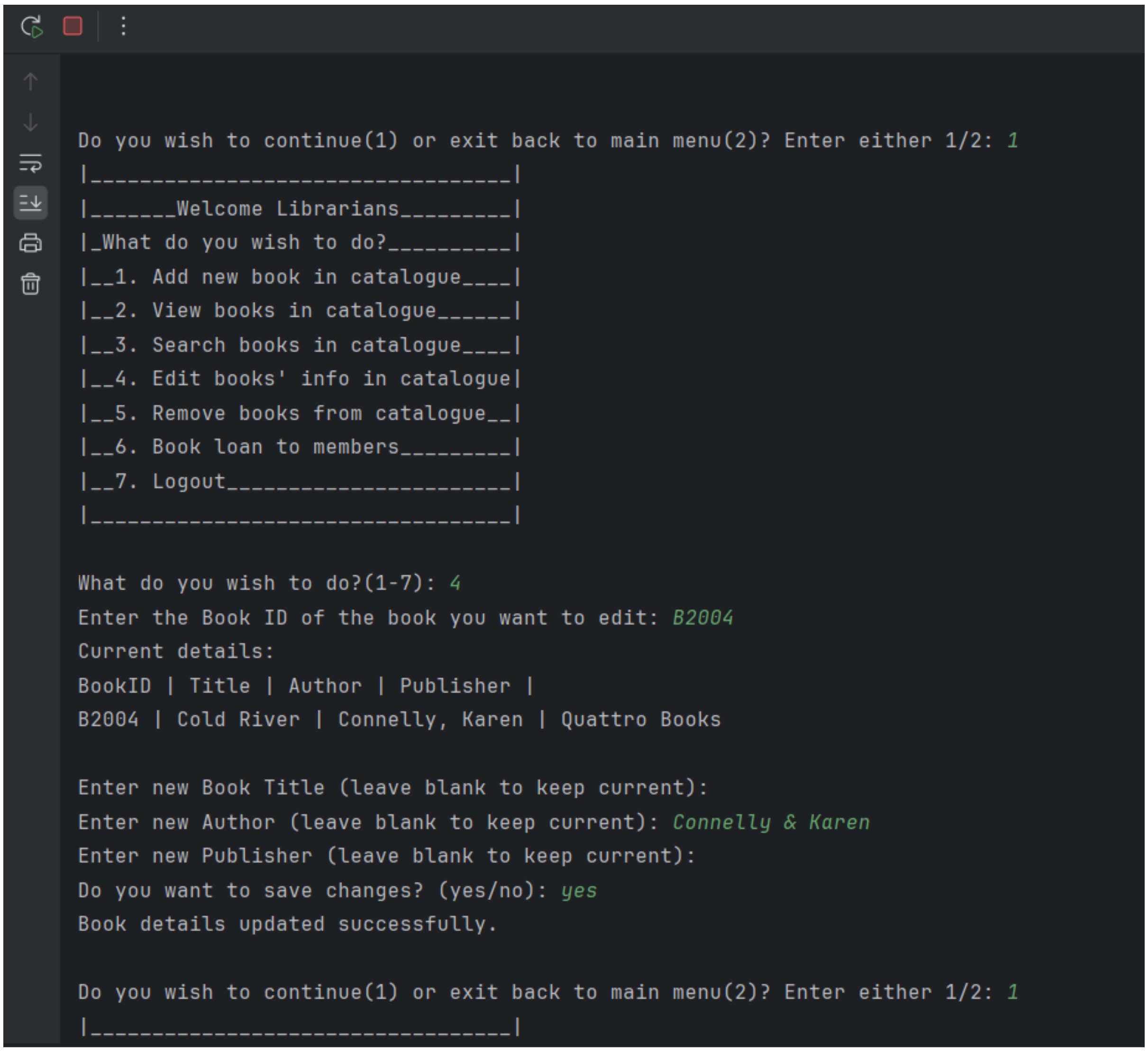
```
B0010 | Life loves you | Louise Hay & Robert Holden | Hay House

B0011 | Shakespeare Tales | William Shakespeare | Oxford House

B6242 | Rich Dad's Cashflow Quadrant | Robert T. Kiyosaki | Plata Publishing

B1128 | Rich Dad Poor Dad | Robert T. Kiyosaki | Plata Publishing

B2004 | Cold River | Connelly, Karen | Quattro Books

B7849 | 1984 | George Orwell | Secker & Warbug

B6188 | Bhagavad Gita as it is | Swami Prabhupada | The Bhaktivedanta Book Trust

Here is the list of all available(not lent) books:
BookID | Title | Author | Publisher

B0005 | After | Anna Tod | Gallery Book

B0006 | After We collided | Anna Tod | Gallery Book

B0007 | After we Fell | Anna Tod | Gallery Book

B0008 | After ever happy | Anna Tod | Gallery Book

B0009 | After Everything | Anna Tod | Gallery Book

B2004 | Cold River | Connelly, Karen | Quattro Books
```

The librarian can also view the books in the system by entering the 2nd option.



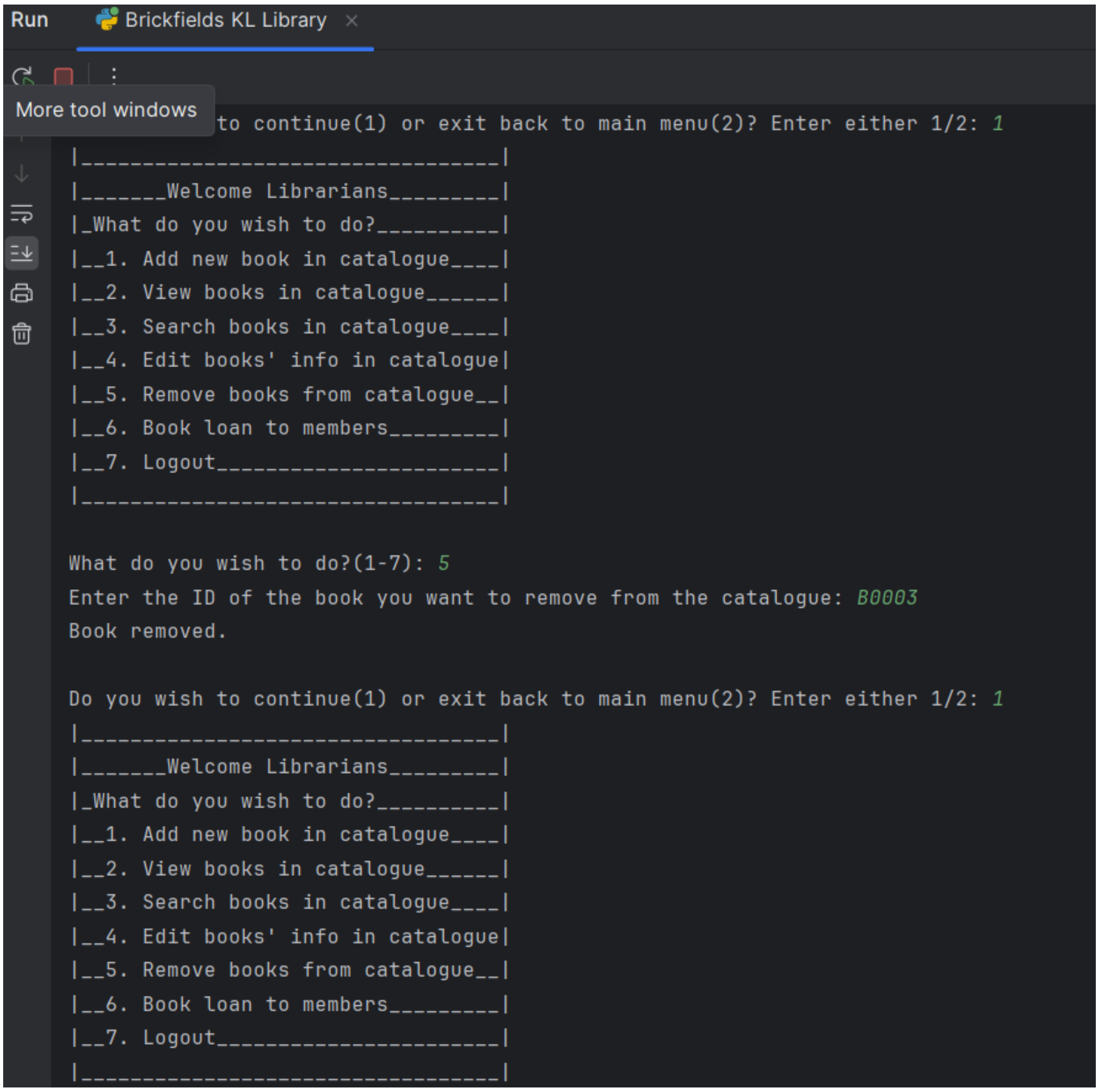
```
B0003 | The Hobbit | J.R.R. Tolkien | Lumando

B6188 | Bhagavad Gita as it is | Swami Prabhupada | The Bhaktivedanta Book Trust


Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

What do you wish to do?(1-7): 3
Do you want to search by BookID(1) or Book title(2): 1
Enter the Book ID: B2004
Book Found! Here are the details:
BookID | Title | Author | Publisher |
B2004 | Cold River | Connelly, Karen | Quattro Books


Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
|_______Welcome Librarians_________|
```

He can also search for specific books by opting for the 3rd option. He can search the book either by entering the ID or Book Title. Then the book’s details are displayed.



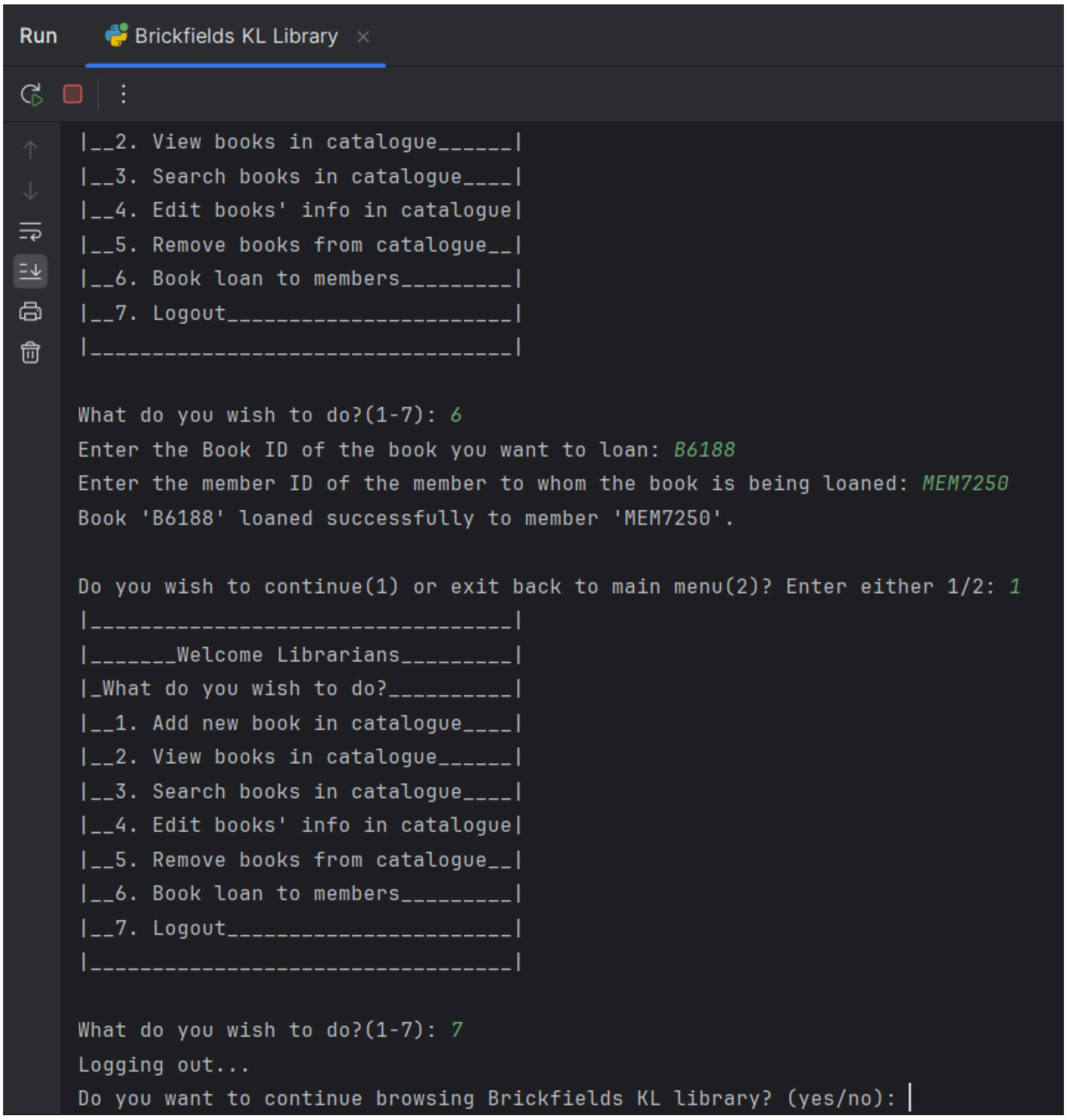
```
Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
|______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout______________________|
|__________________________________|

What do you wish to do?(1-7): 4
Enter the Book ID of the book you want to edit: B2004
Current details:
BookID | Title | Author | Publisher |
B2004 | Cold River | Connelly, Karen | Quattro Books

Enter new Book Title (leave blank to keep current):
Enter new Author (leave blank to keep current): Connelly & Karen
Enter new Publisher (leave blank to keep current):
Do you want to save changes? (yes/no): yes
Book details updated successfully.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
```

He can also edit the book’s information by entering the Book ID. After choosing to save the changes the book’s details are modified in the BookList.txt and availablebooks.txt files.



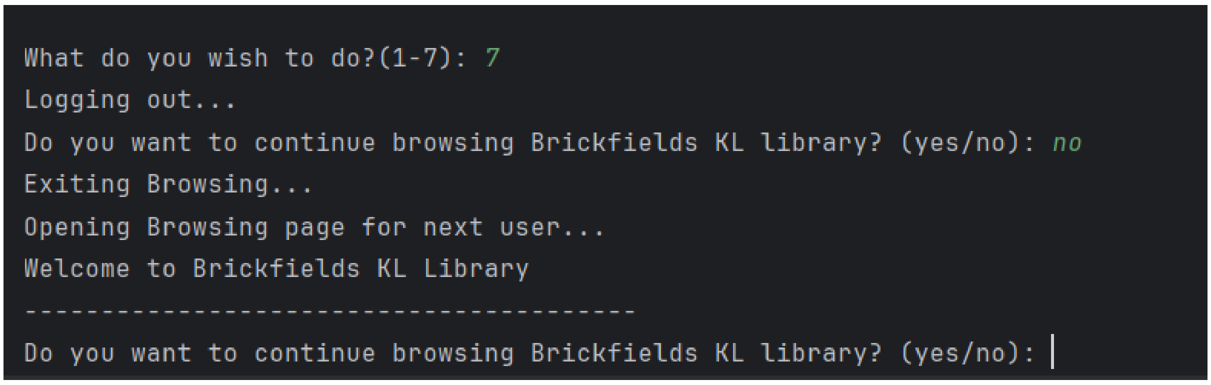
```
to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|___________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|___________________________________|

What do you wish to do?(1-7): 5
Enter the ID of the book you want to remove from the catalogue: B0003
Book removed.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|___________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|___________________________________|
```

He also removed the book with Book ID B0003.



```
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

What do you wish to do?(1-7): 6
Enter the Book ID of the book you want to loan: B6188
Enter the member ID of the member to whom the book is being loaned: MEM7250
Book 'B6188' loaned successfully to member 'MEM7250'.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

What do you wish to do?(1-7): 7
Logging out...
Do you want to continue browsing Brickfields KL library? (yes/no):
```

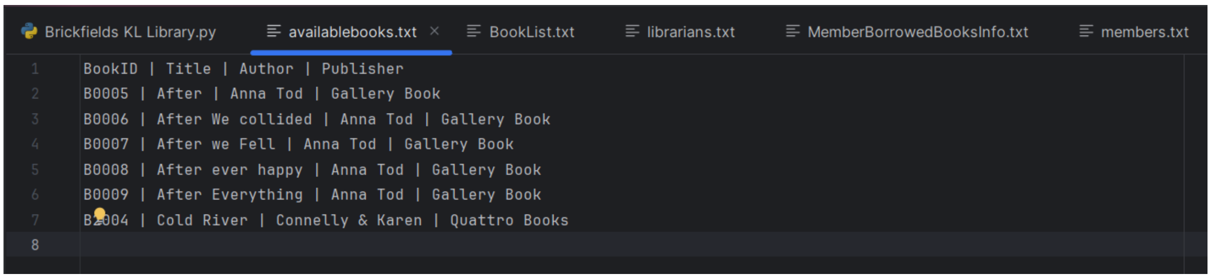
With the 6th option, the librarian has lent the book with Book ID, B6188, to member with member ID, MEM7250



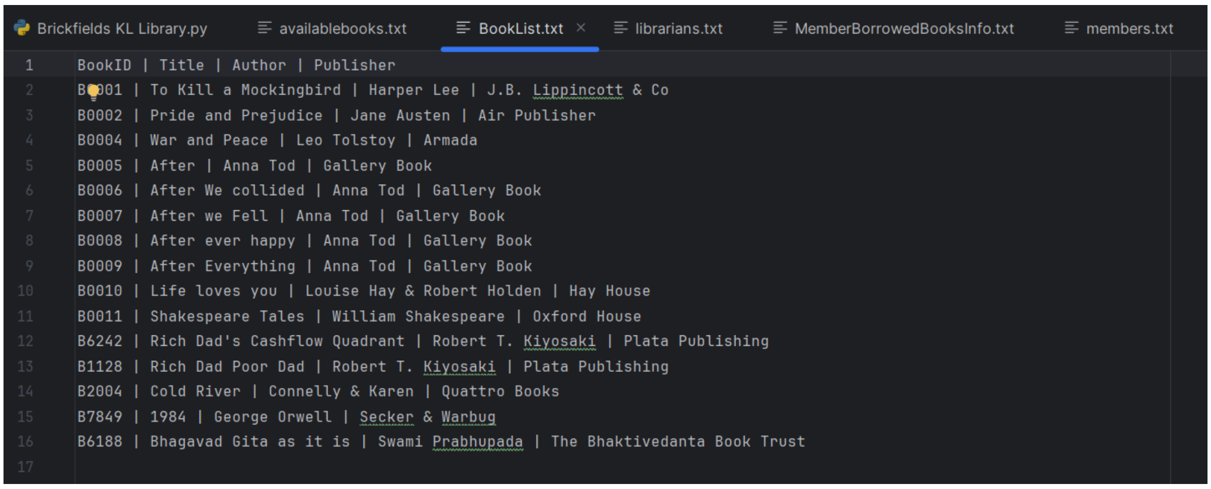
```
What do you wish to do?(1-7): 7
Logging out...
Do you want to continue browsing Brickfields KL library? (yes/no): no
Exiting Browsing...
Opening Browsing page for next user...
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no):
```

By choosing the 7th option, the librarian can log out. The 'logging out' message is printed.

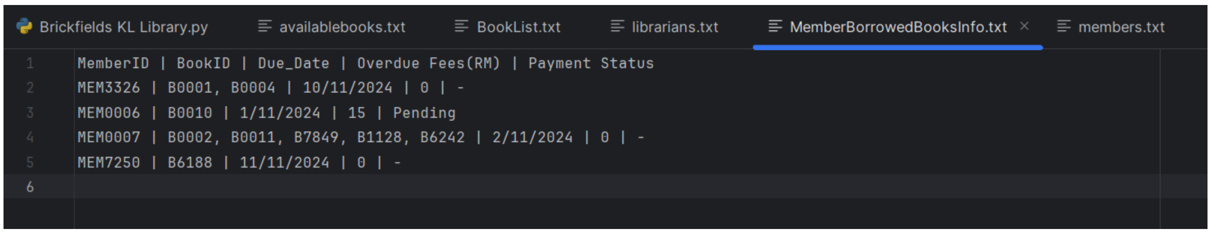
Here are the files after the librarian has made those changes:



```
BookID | Title | Author | Publisher
B0005 | After | Anna Tod | Gallery Book
B0006 | After We collided | Anna Tod | Gallery Book
B0007 | After we Fell | Anna Tod | Gallery Book
B0008 | After ever happy | Anna Tod | Gallery Book
B0009 | After Everything | Anna Tod | Gallery Book
B2004 | Cold River | Connelly & Karen | Quattro Books

```

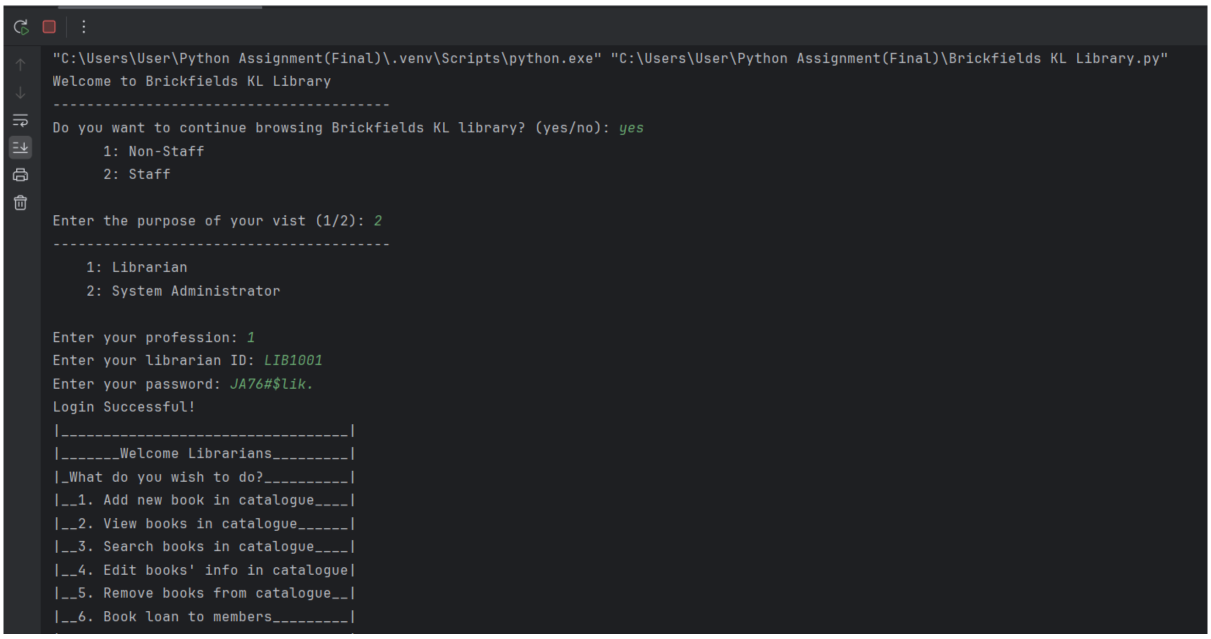


```
BookID | Title | Author | Publisher
B0001 | To Kill a Mockingbird | Harper Lee | J.B. Lippincott & Co
B0002 | Pride and Prejudice | Jane Austen | Air Publisher
B0004 | War and Peace | Leo Tolstoy | Armada
B0005 | After | Anna Tod | Gallery Book
B0006 | After We collided | Anna Tod | Gallery Book
B0007 | After we Fell | Anna Tod | Gallery Book
B0008 | After ever happy | Anna Tod | Gallery Book
B0009 | After Everything | Anna Tod | Gallery Book
B0010 | Life loves you | Louise Hay & Robert Holden | Hay House
B0011 | Shakespeare Tales | William Shakespeare | Oxford House
B6242 | Rich Dad's Cashflow Quadrant | Robert T. Kiyosaki | Plata Publishing
B1128 | Rich Dad Poor Dad | Robert T. Kiyosaki | Plata Publishing
B2004 | Cold River | Connelly & Karen | Quattro Books
B7849 | 1984 | George Orwell | Secker & Warbug
B6188 | Bhagavad Gita as it is | Swami Prabhupada | The Bhaktivedanta Book Trust

```

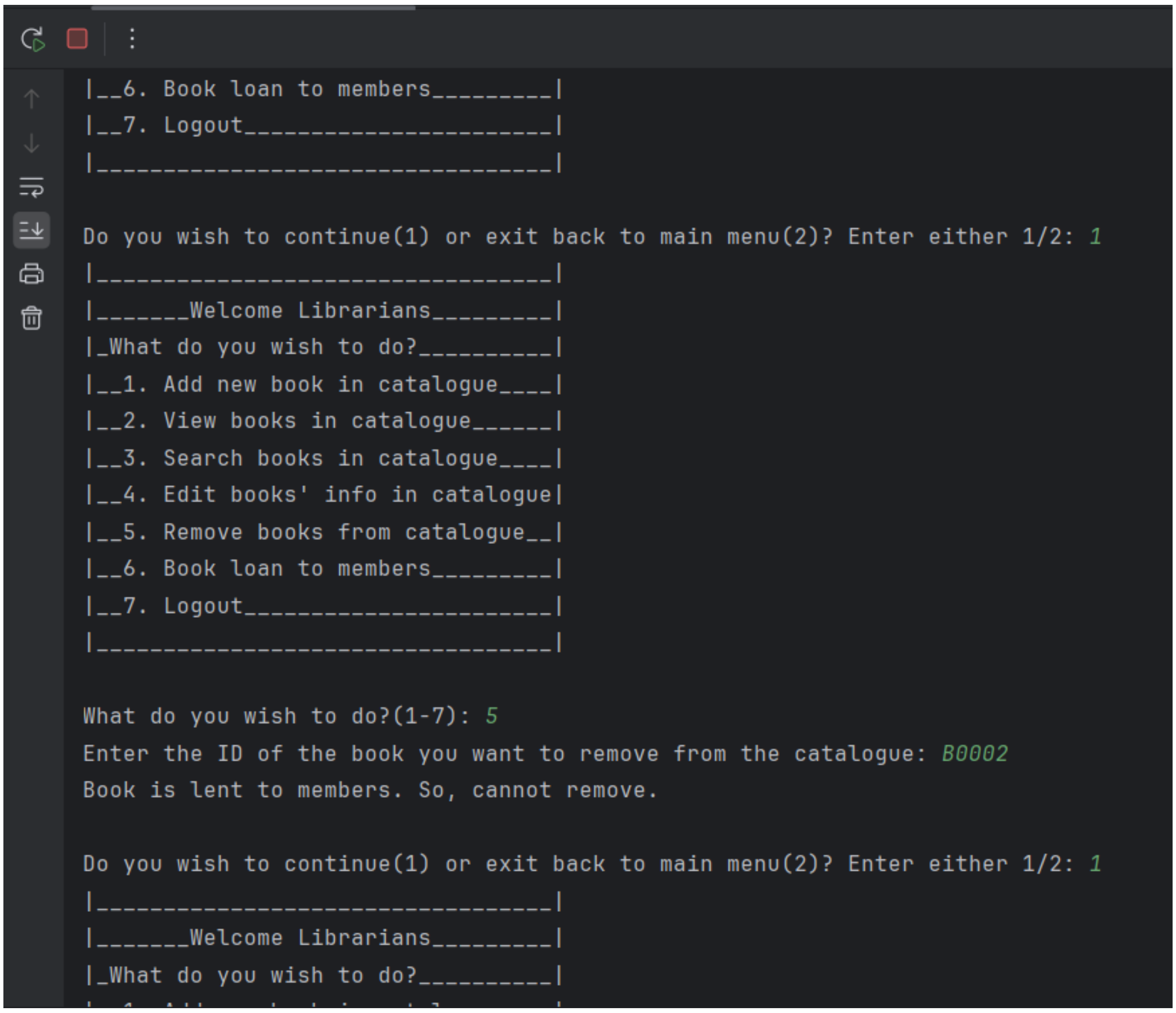


```
MemberID | BookID | Due_Date | Overdue Fees(RM) | Payment Status
MEM3326 | B0001, B0004 | 10/11/2024 | 0 | -
MEM0006 | B0010 | 1/11/2024 | 15 | Pending
MEM0007 | B0002, B0011, B7849, B1128, B6242 | 2/11/2024 | 0 | -
MEM7250 | B6188 | 11/11/2024 | 0 | -

```

6.3 Remove book validation

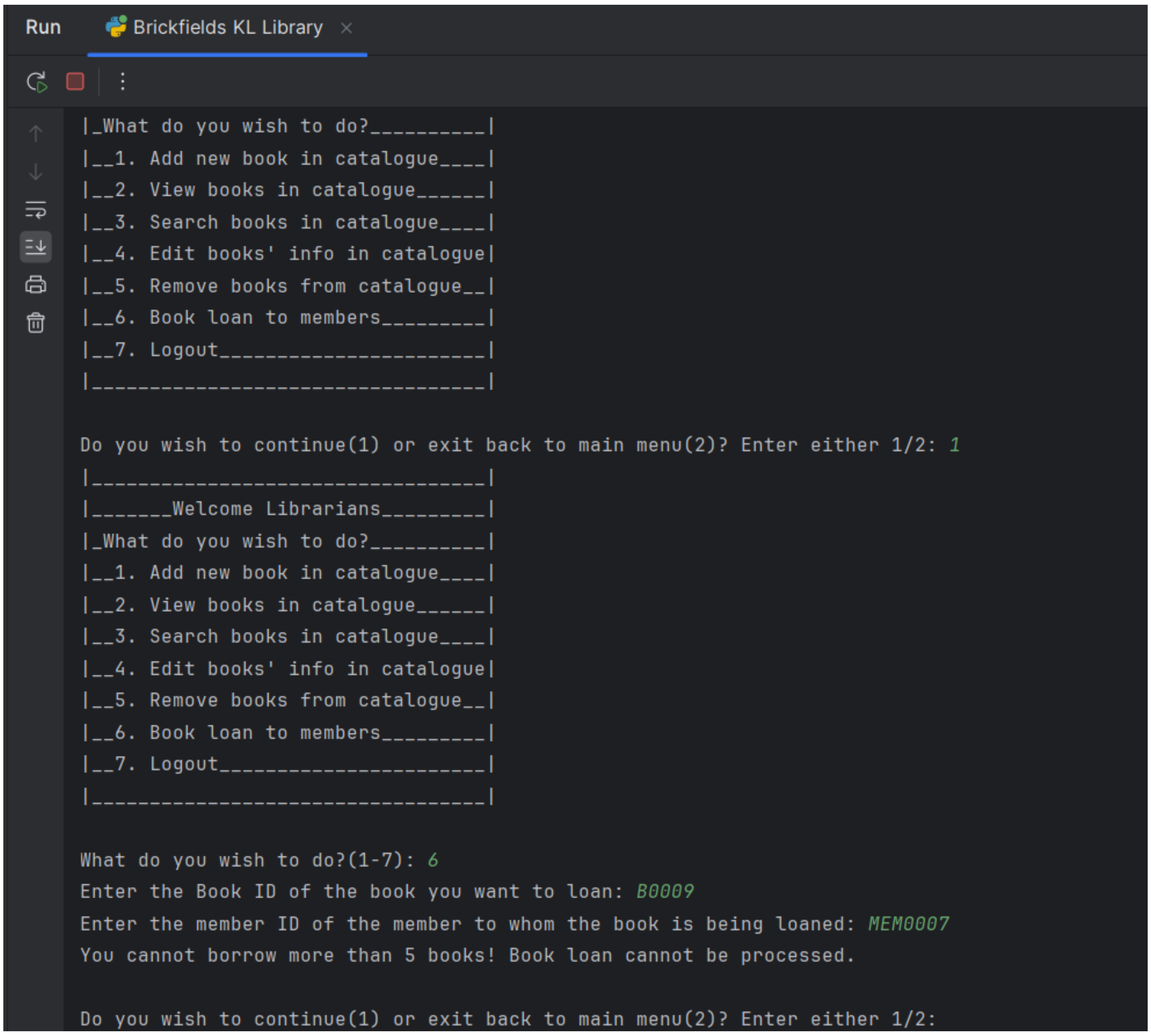


```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): 2
----------------------------------------
    1: Librarian
    2: System Administrator

Enter your profession: 1
Enter your librarian ID: LIB1001
Enter your password: JA76#$lik.
Login Successful!
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
```



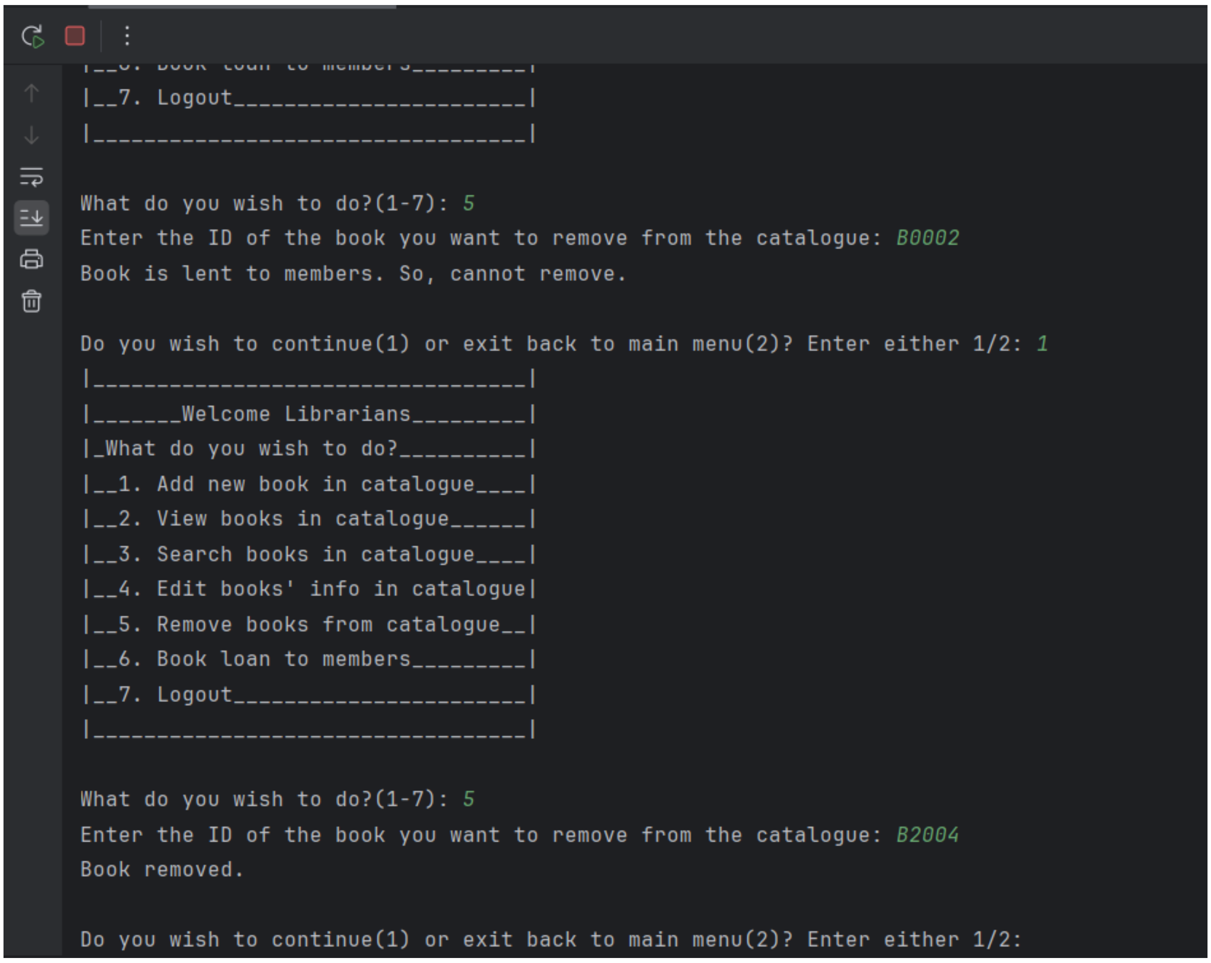
```
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

What do you wish to do?(1-7): 5
Enter the ID of the book you want to remove from the catalogue: B0002
Book is lent to members. So, cannot remove.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
```

The librarian cannot remove a book that a member has borrowed. That's why that message is displayed.



```
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

What do you wish to do?(1-7): 6
Enter the Book ID of the book you want to loan: B0009
Enter the member ID of the member to whom the book is being loaned: MEM0007
You cannot borrow more than 5 books! Book loan cannot be processed.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```

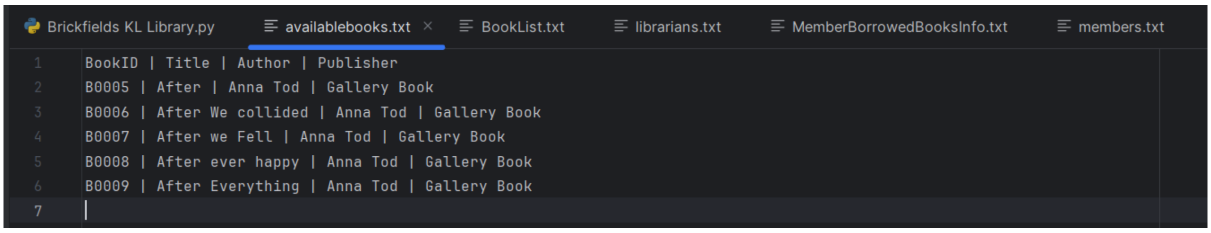
Member with member ID, MEM0007, already has 5 books. So, she cannot borrow more.



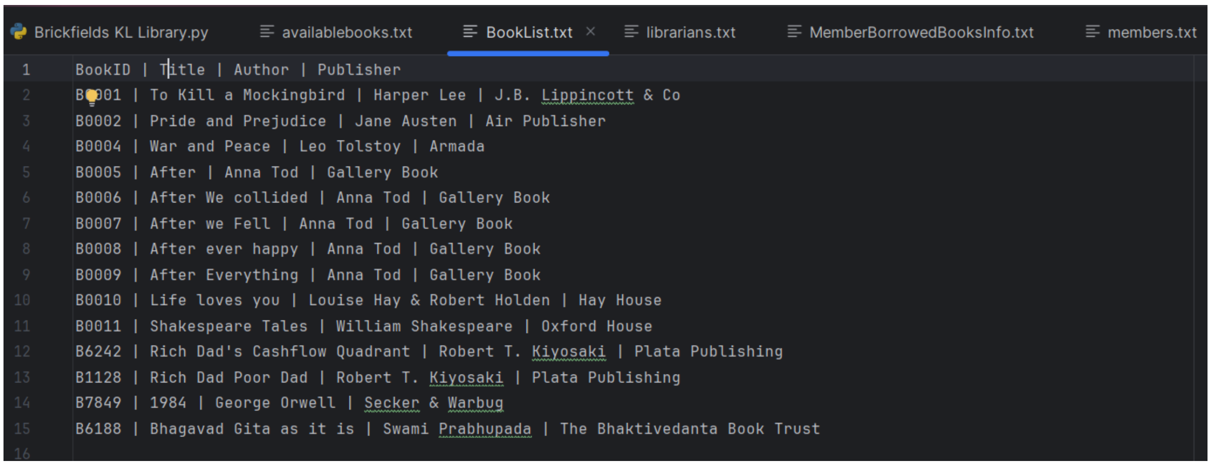
```
|__7. Logout_______________________|
|__________________________________|

What do you wish to do?(1-7): 5
Enter the ID of the book you want to remove from the catalogue: B0002
Book is lent to members. So, cannot remove.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

What do you wish to do?(1-7): 5
Enter the ID of the book you want to remove from the catalogue: B2004
Book removed.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```

The librarian has successfully removed book with Book ID B2004.

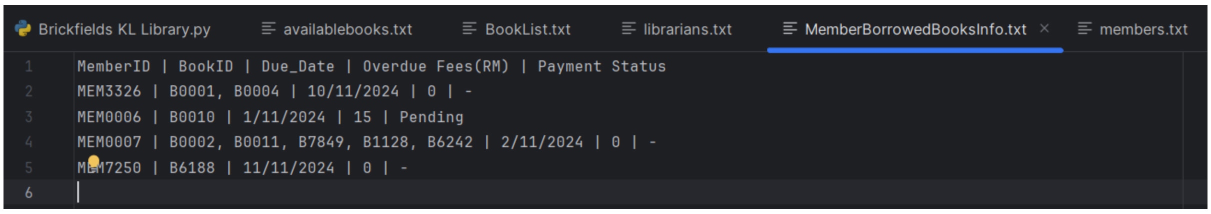
Here are the files after the modifications:



```
BookID | Title | Author | Publisher
B0005 | After | Anna Tod | Gallery Book
B0006 | After We collided | Anna Tod | Gallery Book
B0007 | After we Fell | Anna Tod | Gallery Book
B0008 | After ever happy | Anna Tod | Gallery Book
B0009 | After Everything | Anna Tod | Gallery Book
```

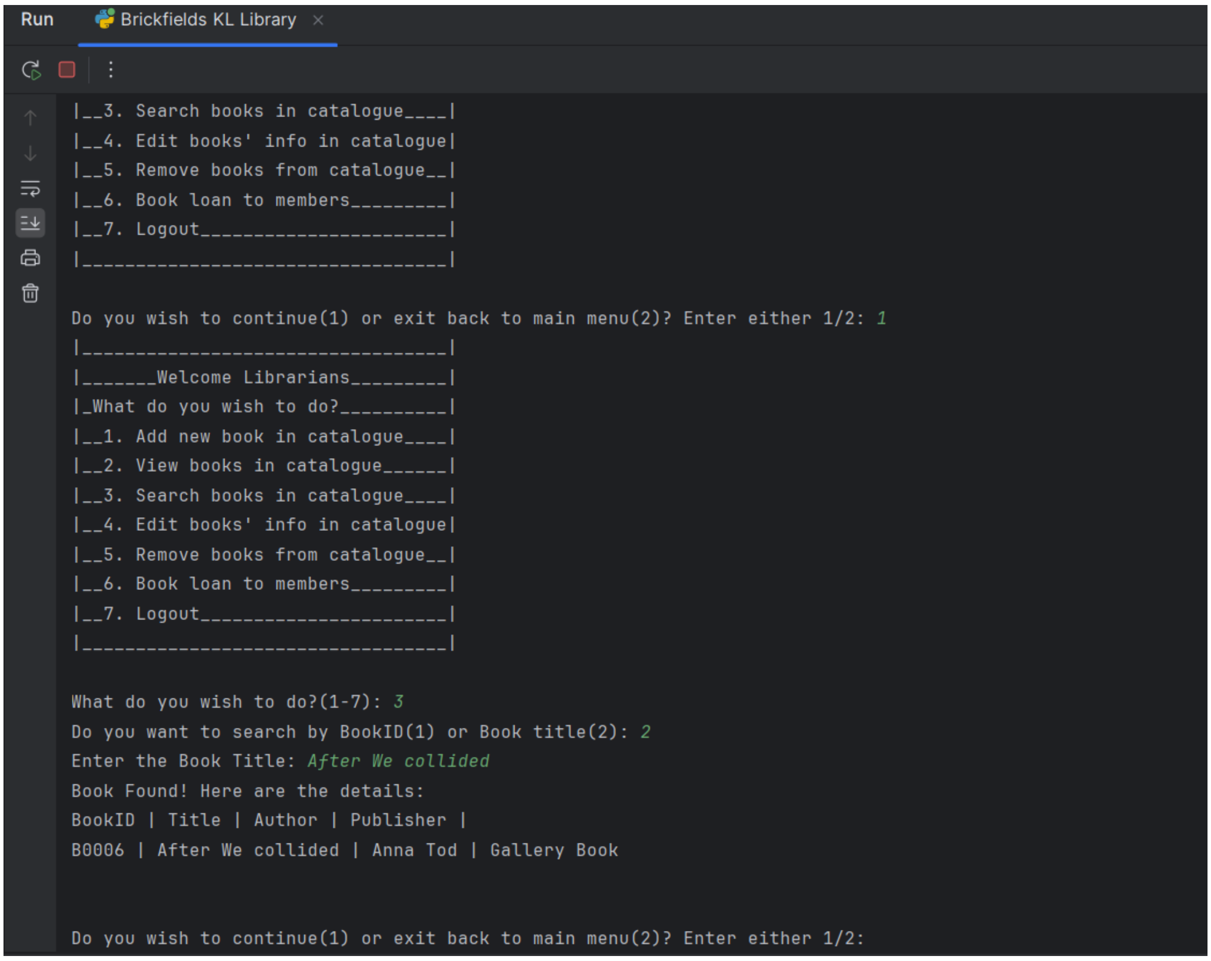


```
BookID | Title | Author | Publisher
B0001 | To Kill a Mockingbird | Harper Lee | J.B. Lippincott & Co
B0002 | Pride and Prejudice | Jane Austen | Air Publisher
B0004 | War and Peace | Leo Tolstoy | Armada
B0005 | After | Anna Tod | Gallery Book
B0006 | After We collided | Anna Tod | Gallery Book
B0007 | After we Fell | Anna Tod | Gallery Book
B0008 | After ever happy | Anna Tod | Gallery Book
B0009 | After Everything | Anna Tod | Gallery Book
B0010 | Life loves you | Louise Hay & Robert Holden | Hay House
B0011 | Shakespeare Tales | William Shakespeare | Oxford House
B6242 | Rich Dad's Cashflow Quadrant | Robert T. Kiyosaki | Plata Publishing
B1128 | Rich Dad Poor Dad | Robert T. Kiyosaki | Plata Publishing
B7849 | 1984 | George Orwell | Secker & Warbug
B6188 | Bhagavad Gita as it is | Swami Prabhupada | The Bhaktivedanta Book Trust
```



```
MemberID | BookID | Due_Date | Overdue Fees(RM) | Payment Status
MEM3326 | B0001, B0004 | 10/11/2024 | 0 | -
MEM0006 | B0010 | 1/11/2024 | 15 | Pending
MEM0007 | B0002, B0011, B7849, B1128, B6242 | 2/11/2024 | 0 | -
MEM7250 | B6188 | 11/11/2024 | 0 | -
```

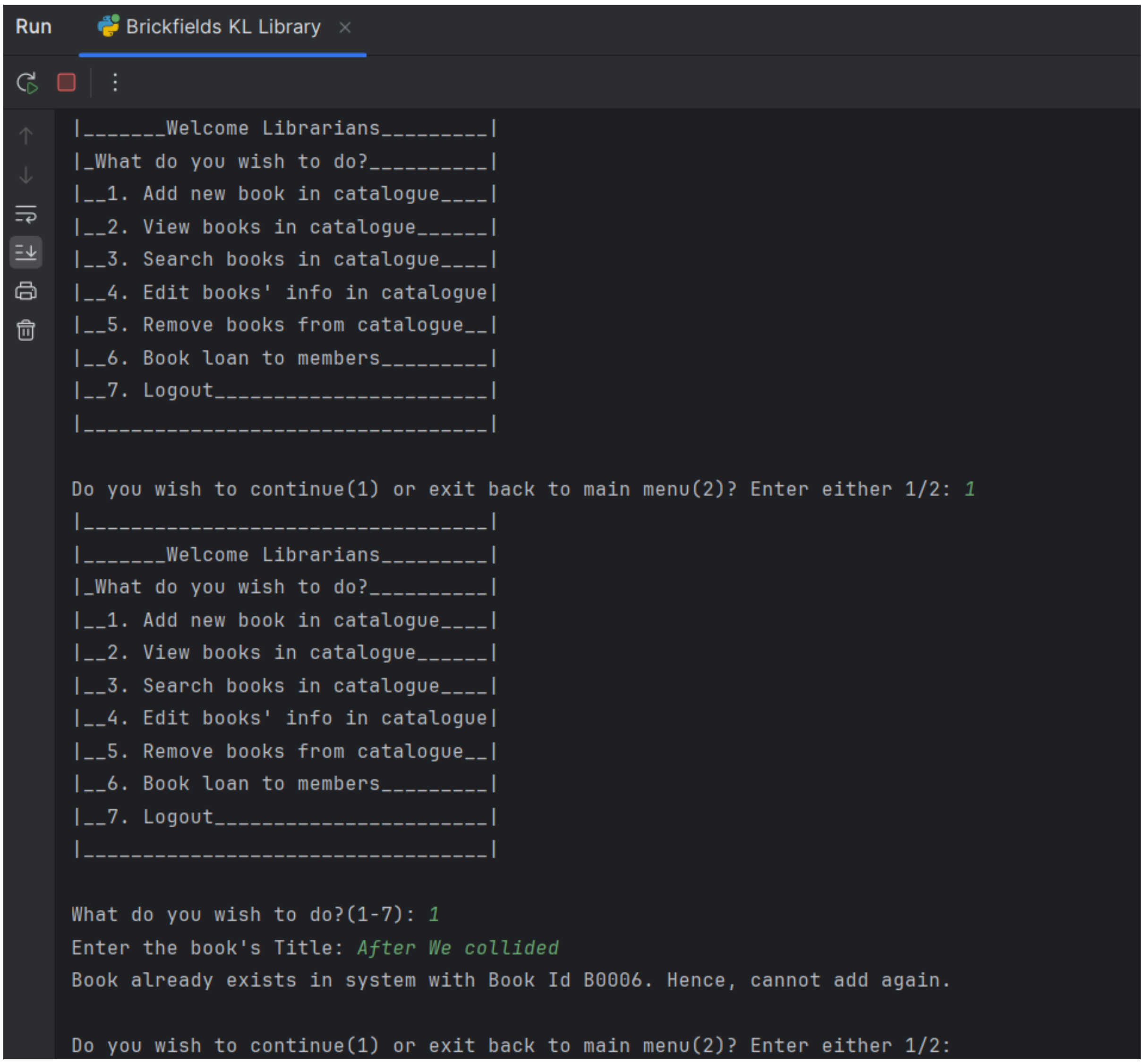
6.4 Search by title



```
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

What do you wish to do?(1-7): 3
Do you want to search by BookID(1) or Book title(2): 2
Enter the Book Title: After We collided
Book Found! Here are the details:
BookID | Title | Author | Publisher |
B0006 | After We collided | Anna Tod | Gallery Book


Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```

The Librarian can also search for book by their title. Here, the librarian searched the book titled, “After we collided”. The details of the book is also displayed.

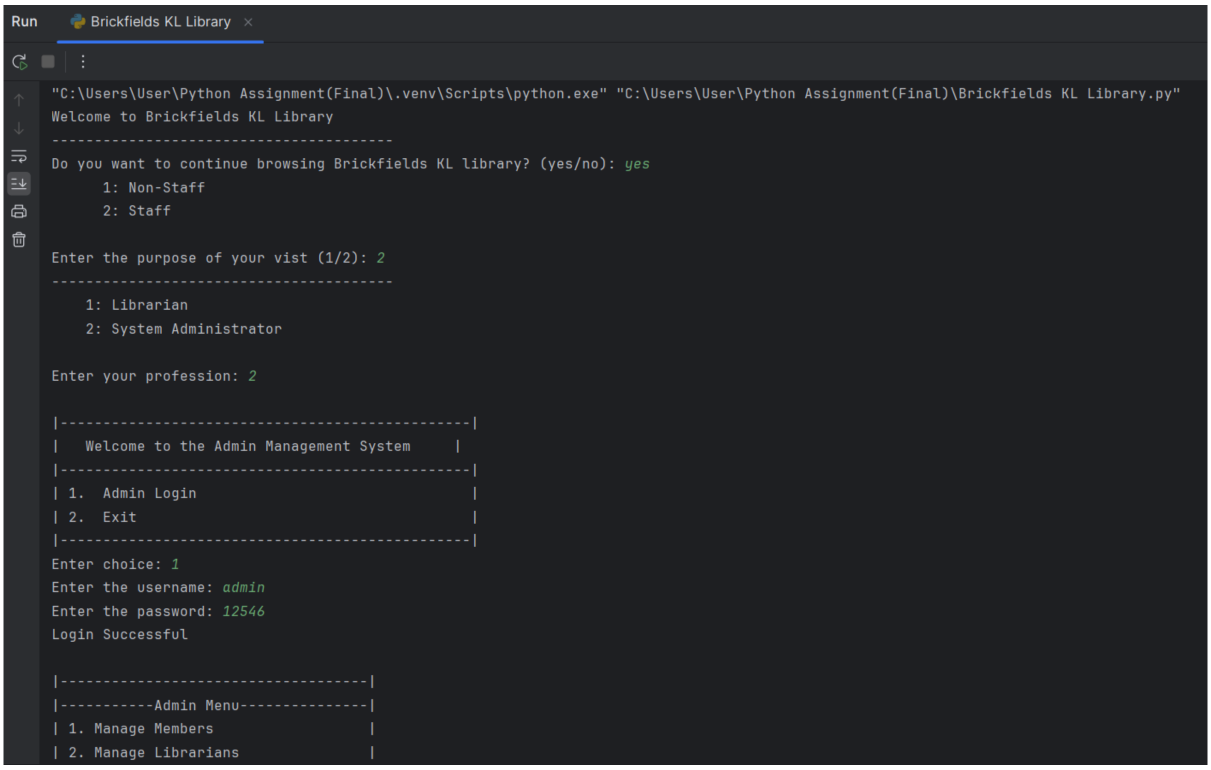
6.5 Add book validation



```
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2: 1
|__________________________________|
|_______Welcome Librarians_________|
|_What do you wish to do?__________|
|__1. Add new book in catalogue____|
|__2. View books in catalogue______|
|__3. Search books in catalogue____|
|__4. Edit books' info in catalogue|
|__5. Remove books from catalogue__|
|__6. Book loan to members_________|
|__7. Logout_______________________|
|__________________________________|

What do you wish to do?(1-7): 1
Enter the book's Title: After We collided
Book already exists in system with Book Id B0006. Hence, cannot add again.

Do you wish to continue(1) or exit back to main menu(2)? Enter either 1/2:
```

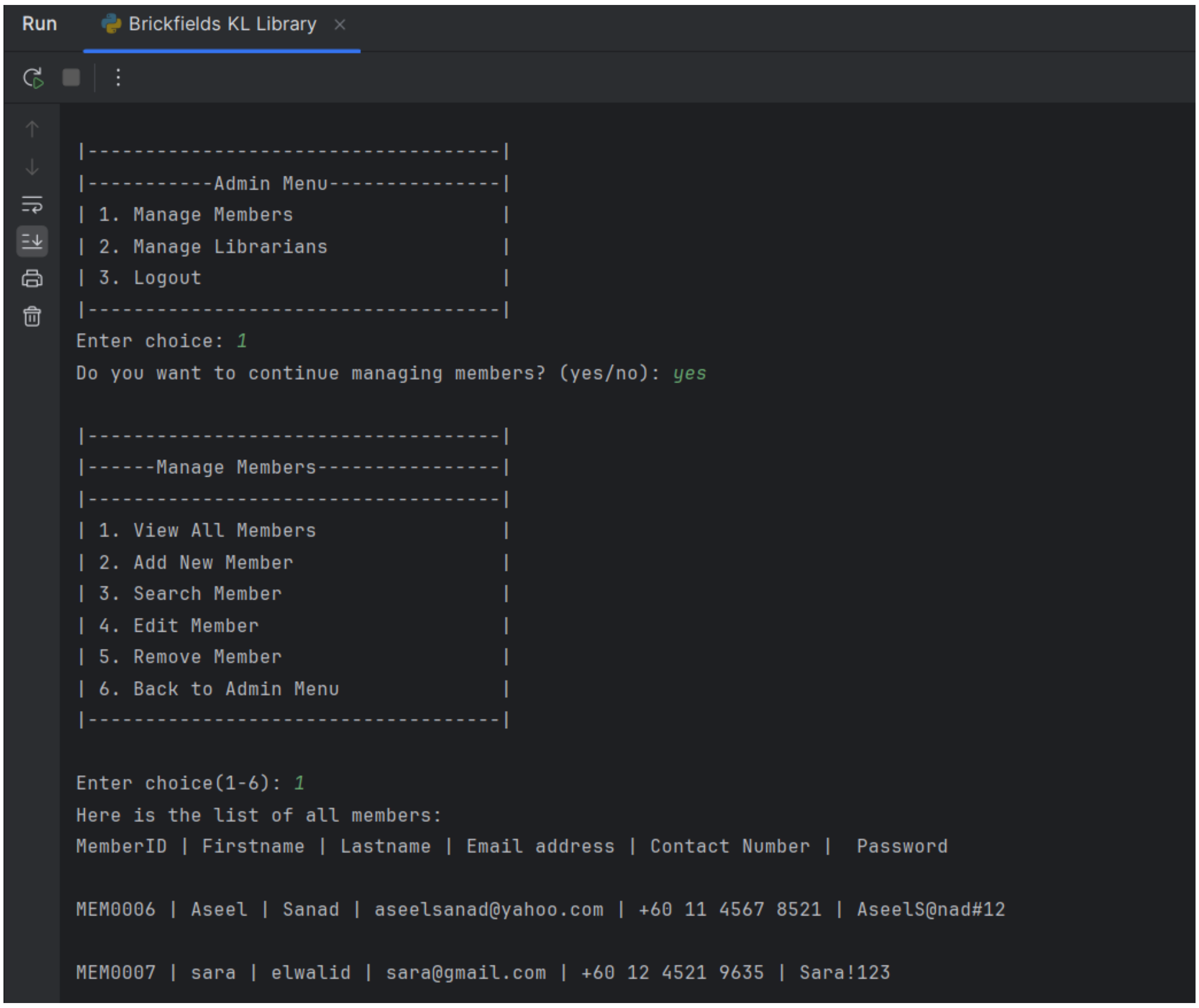
The librarian cannot add books the already exists in the system. The system will detect this, if two books have identical titles.

6.6 System admin

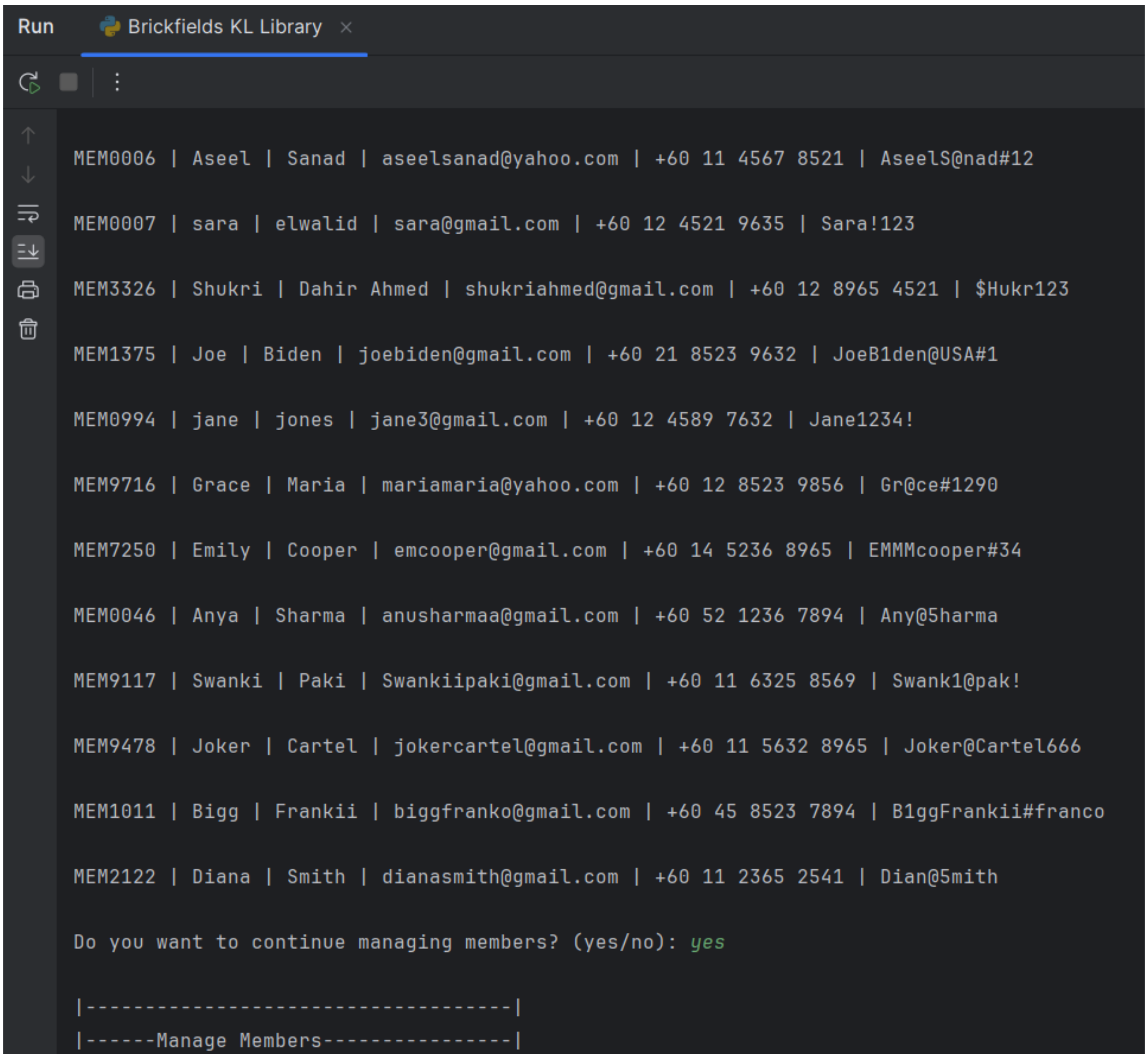


```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): 2
----------------------------------------
    1: Librarian
    2: System Administrator

Enter your profession: 2

|-------------------------------------------------|
|   Welcome to the Admin Management System     |
|-------------------------------------------------|
| 1.  Admin Login                                 |
| 2.  Exit                                        |
|-------------------------------------------------|
Enter choice: 1
Enter the username: admin
Enter the password: 12546
Login Successful

|-------------------------------------|
|-----------Admin Menu---------------|
| 1. Manage Members                   |
| 2. Manage Librarians                |
```

System administrator will enter the username, “admin”, and password, “12546” to log in. Only after entering correct username and password, the administrator can successfully login.

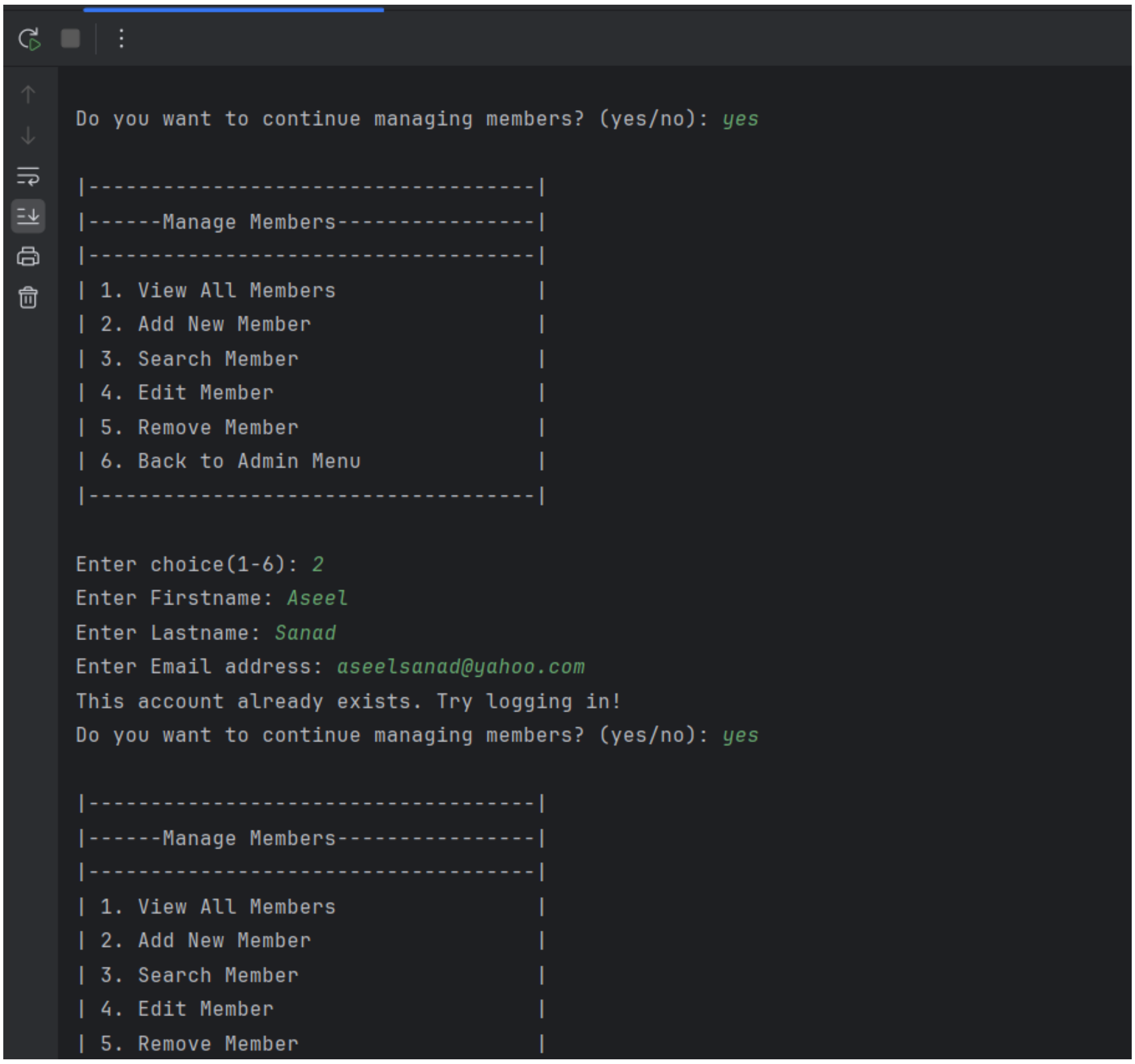


```
|------------------------------------|
|-----------Admin Menu---------------|
| 1. Manage Members                  |
| 2. Manage Librarians               |
| 3. Logout                          |
|------------------------------------|
Enter choice: 1
Do you want to continue managing members? (yes/no): yes

|------------------------------------|
|------Manage Members----------------|
|------------------------------------|
| 1. View All Members                |
| 2. Add New Member                  |
| 3. Search Member                   |
| 4. Edit Member                     |
| 5. Remove Member                   |
| 6. Back to Admin Menu              |
|------------------------------------|

Enter choice(1-6): 1
Here is the list of all members:
MemberID | Firstname | Lastname | Email address | Contact Number |  Password

MEM0006 | Aseel | Sanad | aseelsanad@yahoo.com | +60 11 4567 8521 | AseelS@nad#12

MEM0007 | sara | elwalid | sara@gmail.com | +60 12 4521 9635 | Sara!123
```

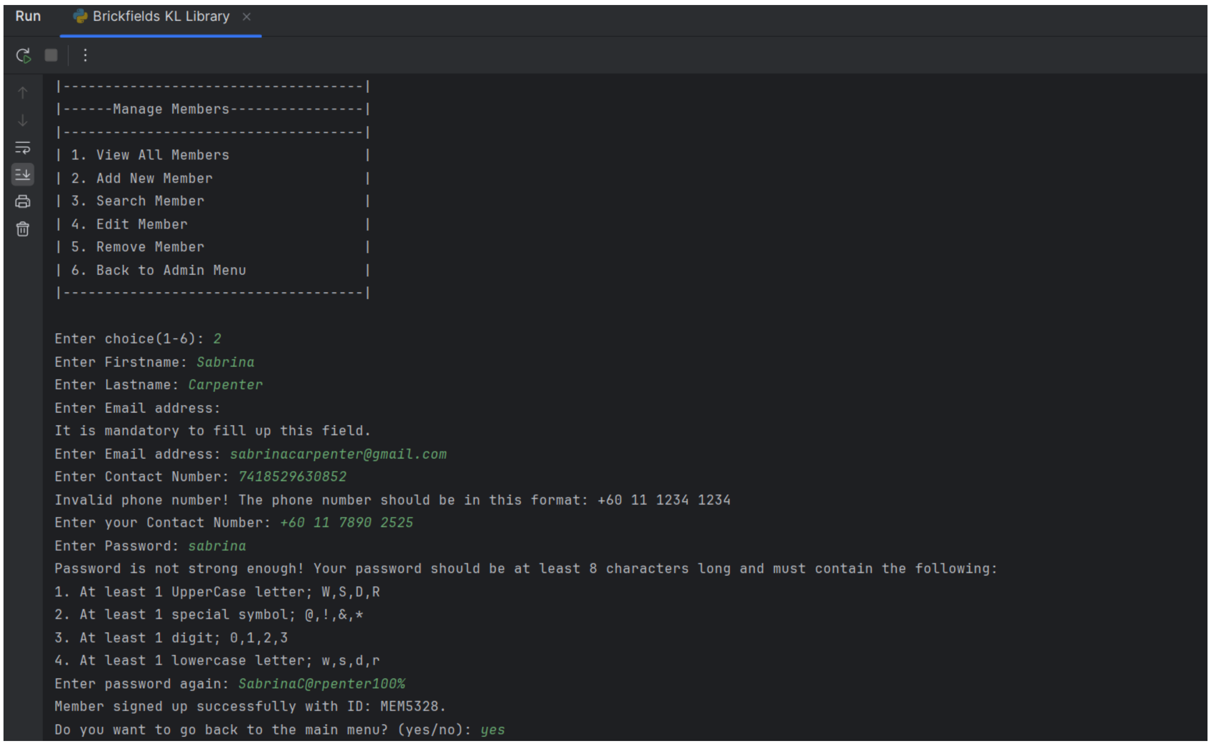


```
MEM0006 | Aseel | Sanad | aseelsanad@yahoo.com | +60 11 4567 8521 | AseelS@nad#12

MEM0007 | sara | elwalid | sara@gmail.com | +60 12 4521 9635 | Sara!123

MEM3326 | Shukri | Dahir Ahmed | shukriahmed@gmail.com | +60 12 8965 4521 | $Hukr123

MEM1375 | Joe | Biden | joebiden@gmail.com | +60 21 8523 9632 | JoeB1den@USA#1

MEM0994 | jane | jones | jane3@gmail.com | +60 12 4589 7632 | Jane1234!

MEM9716 | Grace | Maria | mariamaria@yahoo.com | +60 12 8523 9856 | Gr@ce#1290

MEM7250 | Emily | Cooper | emcooper@gmail.com | +60 14 5236 8965 | EMMMcooper#34

MEM0046 | Anya | Sharma | anusharmaa@gmail.com | +60 52 1236 7894 | Any@5harma

MEM9117 | Swanki | Paki | Swankiipaki@gmail.com | +60 11 6325 8569 | Swank1@pak!

MEM9478 | Joker | Cartel | jokercartel@gmail.com | +60 11 5632 8965 | Joker@Cartel666

MEM1011 | Bigg | Frankii | biggfranko@gmail.com | +60 45 8523 7894 | B1ggFrankii#franco

MEM2122 | Diana | Smith | dianasmith@gmail.com | +60 11 2365 2541 | Dian@5mith

Do you want to continue managing members? (yes/no): yes

|------------------------------------|
|------Manage Members----------------|
```

By choosing the first option, to manage members, the administrator will enter the manage member menu. Here, he chose to view all the members. The members' list from members.txt is displayed.

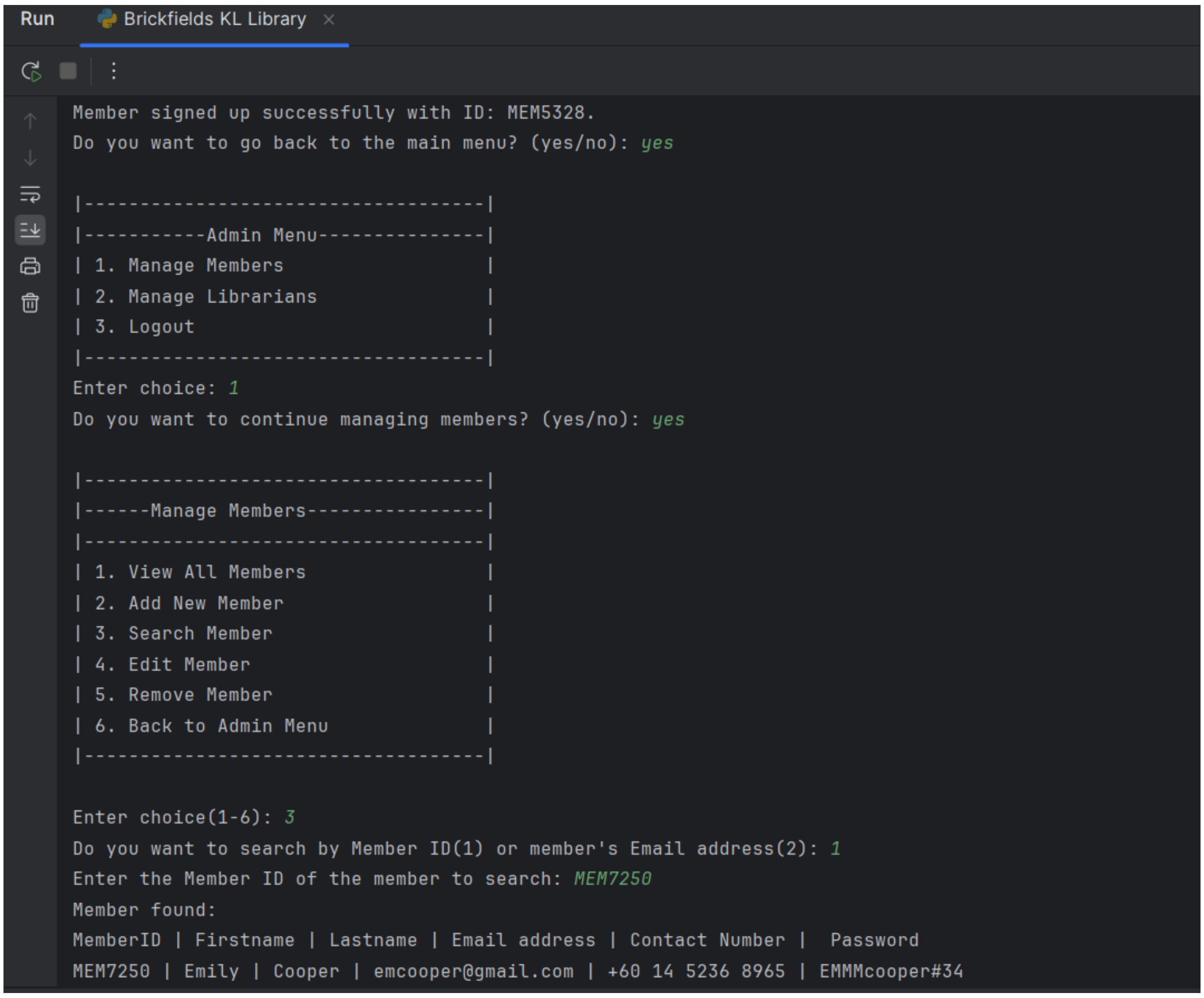

```
Do you want to continue managing members? (yes/no): yes

|-----------------------------------|
|------Manage Members---------------|
|-----------------------------------|
| 1. View All Members               |
| 2. Add New Member                 |
| 3. Search Member                  |
| 4. Edit Member                    |
| 5. Remove Member                  |
| 6. Back to Admin Menu             |
|-----------------------------------|

Enter choice(1-6): 2
Enter Firstname: Aseel
Enter Lastname: Sanad
Enter Email address: aseelsanad@yahoo.com
This account already exists. Try logging in!
Do you want to continue managing members? (yes/no): yes

|-----------------------------------|
|------Manage Members---------------|
|-----------------------------------|
| 1. View All Members               |
| 2. Add New Member                 |
| 3. Search Member                  |
| 4. Edit Member                    |
| 5. Remove Member                  |
```

With the 2nd option, the administrator can add new members. Here, he has entered data of an existing member. Hence, an error message is displayed.



```
|-----------------------------------|
|------Manage Members---------------|
|-----------------------------------|
| 1. View All Members               |
| 2. Add New Member                 |
| 3. Search Member                  |
| 4. Edit Member                    |
| 5. Remove Member                  |
| 6. Back to Admin Menu             |
|-----------------------------------|

Enter choice(1-6): 2
Enter Firstname: Sabrina
Enter Lastname: Carpenter
Enter Email address:
It is mandatory to fill up this field.
Enter Email address: sabrinacarpenter@gmail.com
Enter Contact Number: 7418529630852
Invalid phone number! The phone number should be in this format: +60 11 1234 1234
Enter your Contact Number: +60 11 7890 2525
Enter Password: sabrina
Password is not strong enough! Your password should be at least 8 characters long and must contain the following:
1. At least 1 UpperCase letter; W,S,D,R
2. At least 1 special symbol; @,!,&,*
3. At least 1 digit; 0,1,2,3
4. At least 1 lowercase letter; w,s,d,r
Enter password again: SabrinaC@rpenter100%
Member signed up successfully with ID: MEM5328.
Do you want to go back to the main menu? (yes/no): yes
```

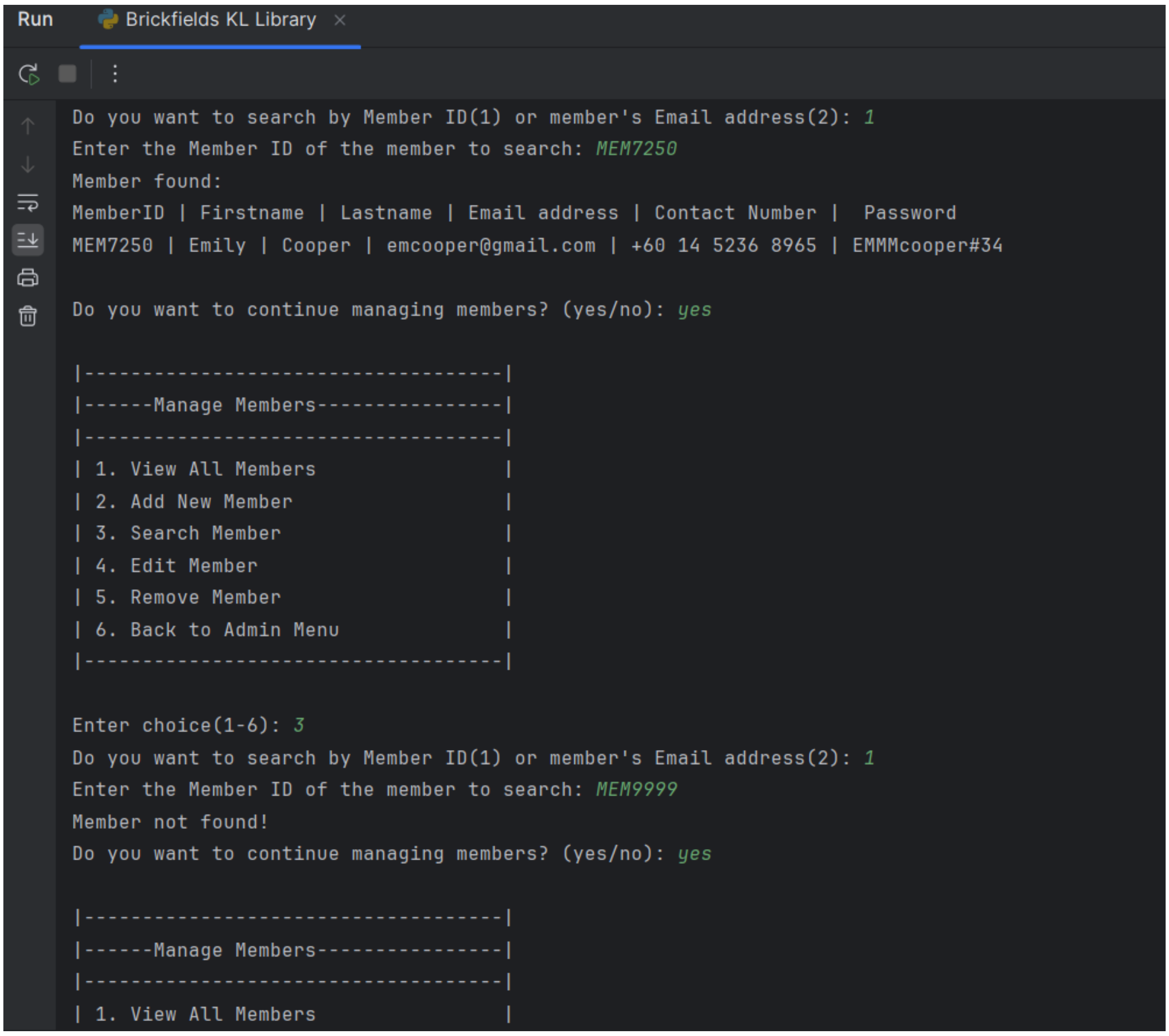
He can also add validated data of new members. A member ID is automatically assigned to the member.



```
Member signed up successfully with ID: MEM5328.
Do you want to go back to the main menu? (yes/no): yes

|-------------------------------------|
|-----------Admin Menu----------------|
| 1. Manage Members                   |
| 2. Manage Librarians                |
| 3. Logout                           |
|-------------------------------------|
Enter choice: 1
Do you want to continue managing members? (yes/no): yes

|-------------------------------------|
|------Manage Members-----------------|
|-------------------------------------|
| 1. View All Members                 |
| 2. Add New Member                   |
| 3. Search Member                    |
| 4. Edit Member                      |
| 5. Remove Member                    |
| 6. Back to Admin Menu               |
|-------------------------------------|

Enter choice(1-6): 3
Do you want to search by Member ID(1) or member's Email address(2): 1
Enter the Member ID of the member to search: MEM7250
Member found:
MemberID | Firstname | Lastname | Email address | Contact Number |  Password
MEM7250 | Emily | Cooper | emcooper@gmail.com | +60 14 5236 8965 | EMMMcooper#34
```

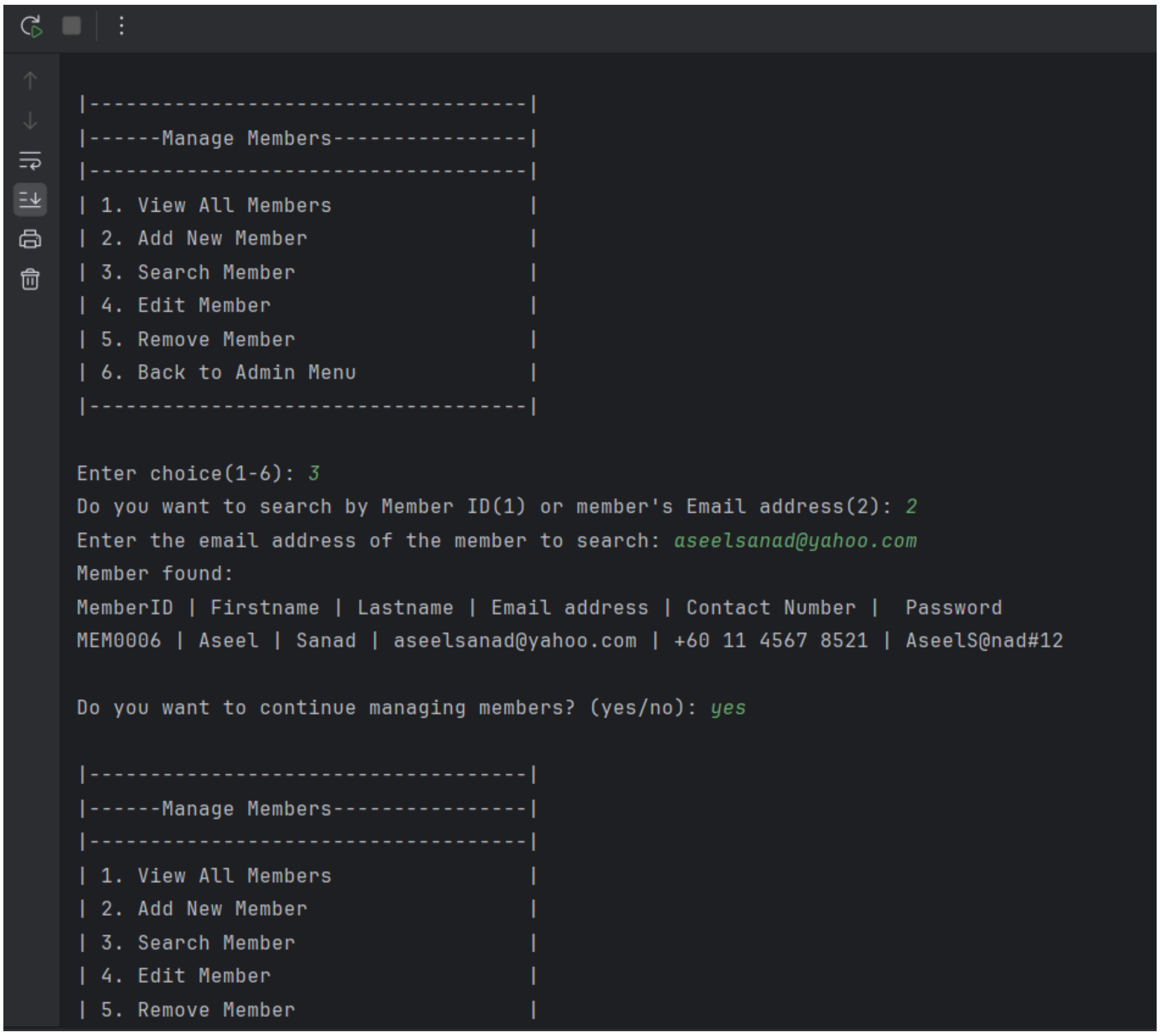
He can also search for specific members by entering the Member ID.

If correct ID is entered, the data about that specific member is displayed. Here, data about Emily Cooper is displayed. He can also search for members by their email address.



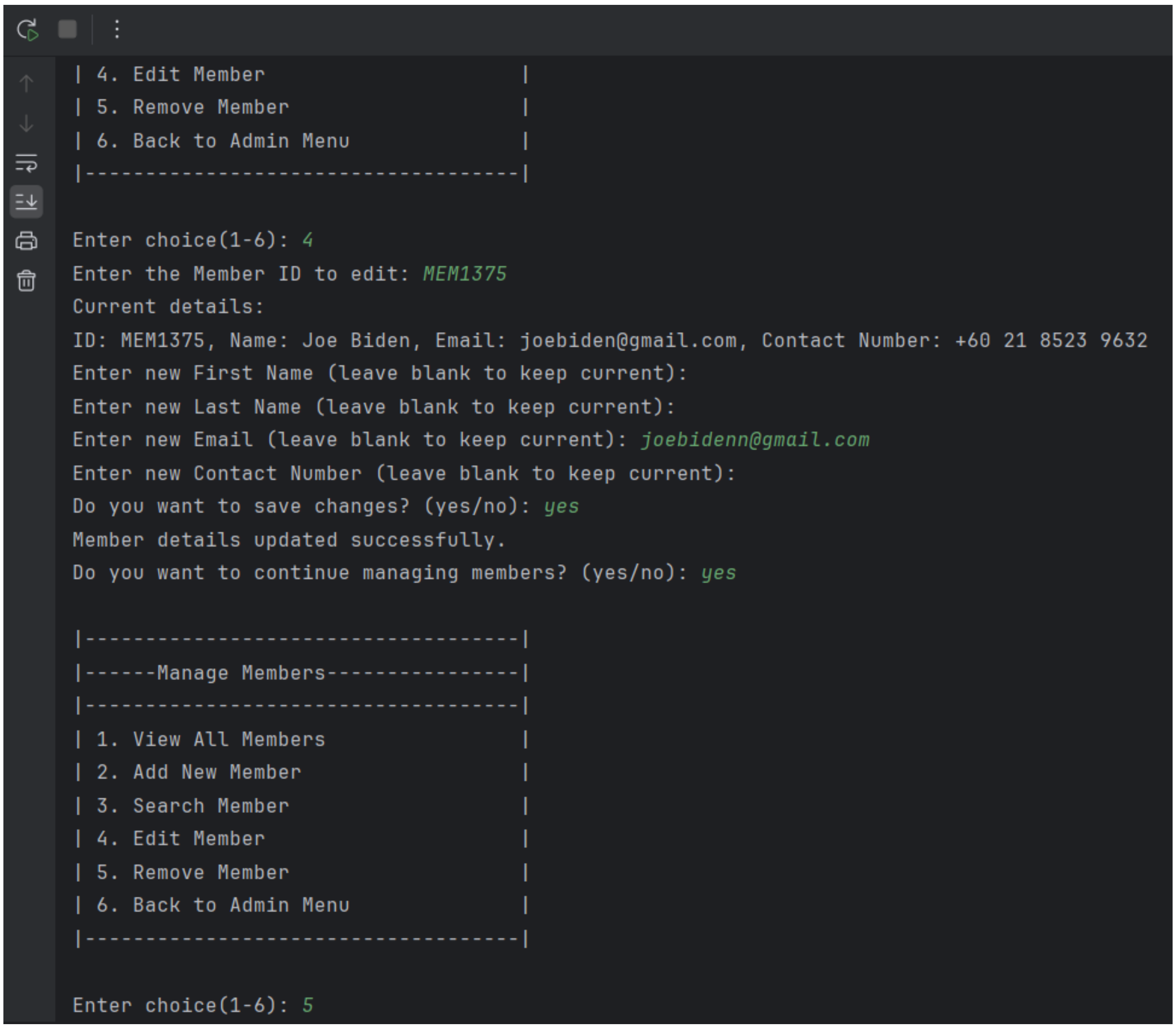
```
Do you want to search by Member ID(1) or member's Email address(2): 1
Enter the Member ID of the member to search: MEM7250
Member found:
MemberID | Firstname | Lastname | Email address | Contact Number |  Password
MEM7250 | Emily | Cooper | emcooper@gmail.com | +60 14 5236 8965 | EMMMcooper#34

Do you want to continue managing members? (yes/no): yes

|-------------------------------------|
|------Manage Members-----------------|
|-------------------------------------|
| 1. View All Members                 |
| 2. Add New Member                   |
| 3. Search Member                    |
| 4. Edit Member                      |
| 5. Remove Member                    |
| 6. Back to Admin Menu               |
|-------------------------------------|

Enter choice(1-6): 3
Do you want to search by Member ID(1) or member's Email address(2): 1
Enter the Member ID of the member to search: MEM9999
Member not found!
Do you want to continue managing members? (yes/no): yes

|-------------------------------------|
|------Manage Members-----------------|
|-------------------------------------|
| 1. View All Members                 |
```

Here, the administrator entered an ID that does not exist in the system, so, an error message is displayed.



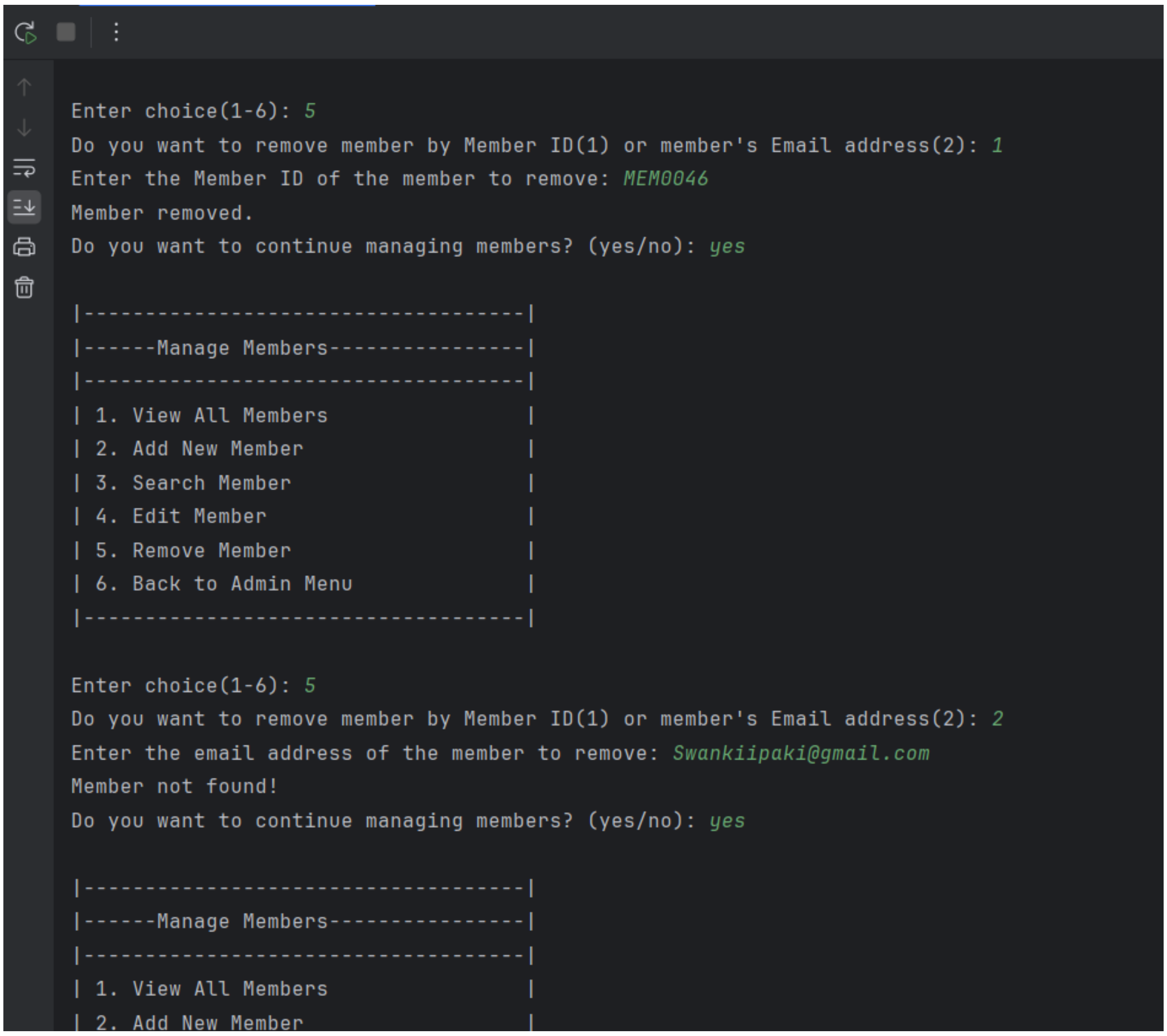
```
|-----------------------------------|
|------Manage Members---------------|
|-----------------------------------|
| 1. View All Members               |
| 2. Add New Member                 |
| 3. Search Member                  |
| 4. Edit Member                    |
| 5. Remove Member                  |
| 6. Back to Admin Menu             |
|-----------------------------------|

Enter choice(1-6): 3
Do you want to search by Member ID(1) or member's Email address(2): 2
Enter the email address of the member to search: aseelsanad@yahoo.com
Member found:
MemberID | Firstname | Lastname | Email address | Contact Number |  Password
MEM0006 | Aseel | Sanad | aseelsanad@yahoo.com | +60 11 4567 8521 | AseelS@nad#12

Do you want to continue managing members? (yes/no): yes

|-----------------------------------|
|------Manage Members---------------|
|-----------------------------------|
| 1. View All Members               |
| 2. Add New Member                 |
| 3. Search Member                  |
| 4. Edit Member                    |
| 5. Remove Member                  |
```

The administrator entered the email address of a member, and her data is displayed on the screen.



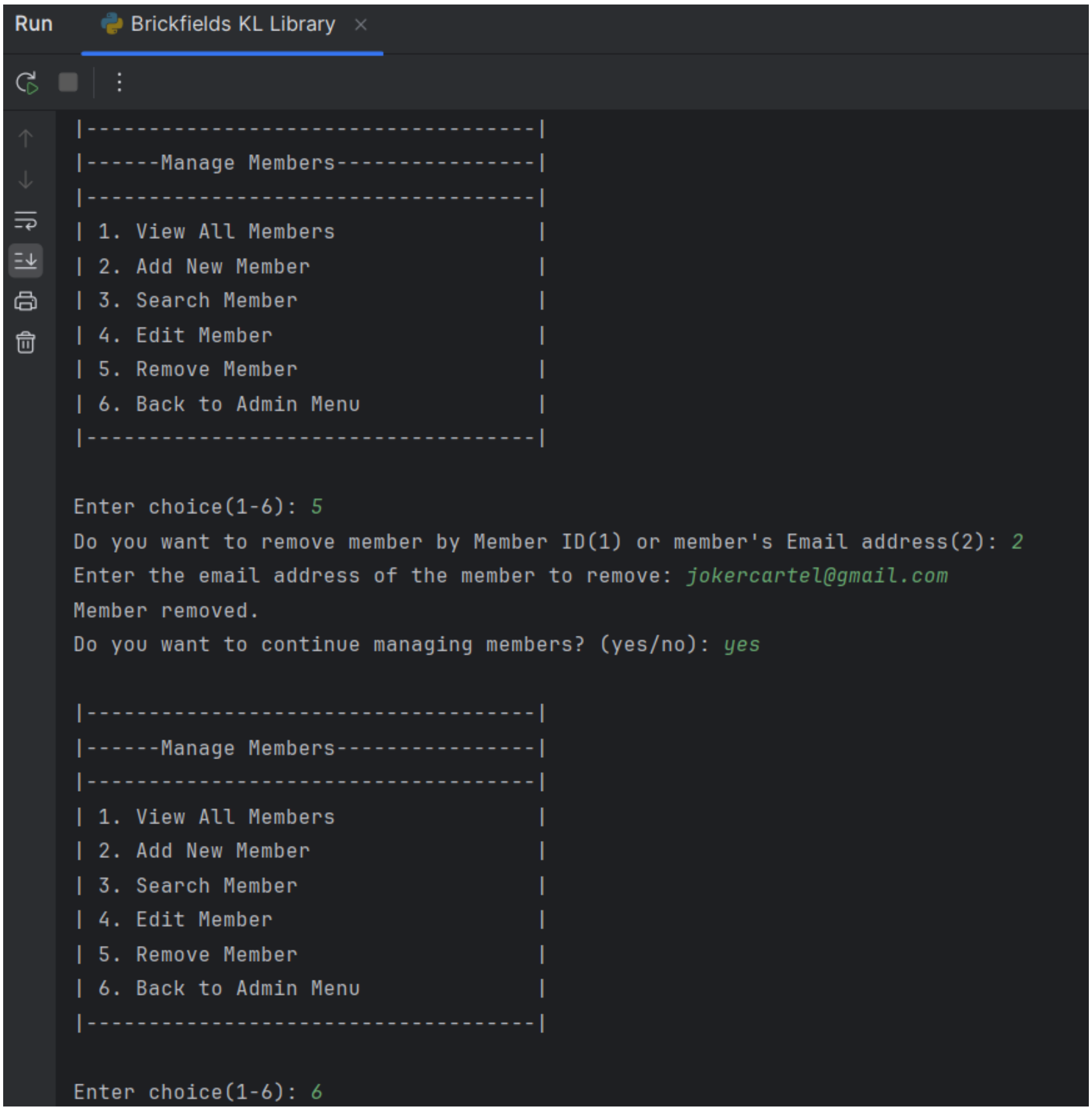
```
| 4. Edit Member                     |
| 5. Remove Member                   |
| 6. Back to Admin Menu              |
|------------------------------------|

Enter choice(1-6): 4
Enter the Member ID to edit: MEM1375
Current details:
ID: MEM1375, Name: Joe Biden, Email: joebiden@gmail.com, Contact Number: +60 21 8523 9632
Enter new First Name (leave blank to keep current):
Enter new Last Name (leave blank to keep current):
Enter new Email (leave blank to keep current): joebidenn@gmail.com
Enter new Contact Number (leave blank to keep current):
Do you want to save changes? (yes/no): yes
Member details updated successfully.
Do you want to continue managing members? (yes/no): yes

|------------------------------------|
|------Manage Members----------------|
|------------------------------------|
| 1. View All Members                |
| 2. Add New Member                  |
| 3. Search Member                   |
| 4. Edit Member                     |
| 5. Remove Member                   |
| 6. Back to Admin Menu              |
|------------------------------------|

Enter choice(1-6): 5
```

He can also edit information about the member in the system. Here, Joe’s data was successfully edited and stored in the file.



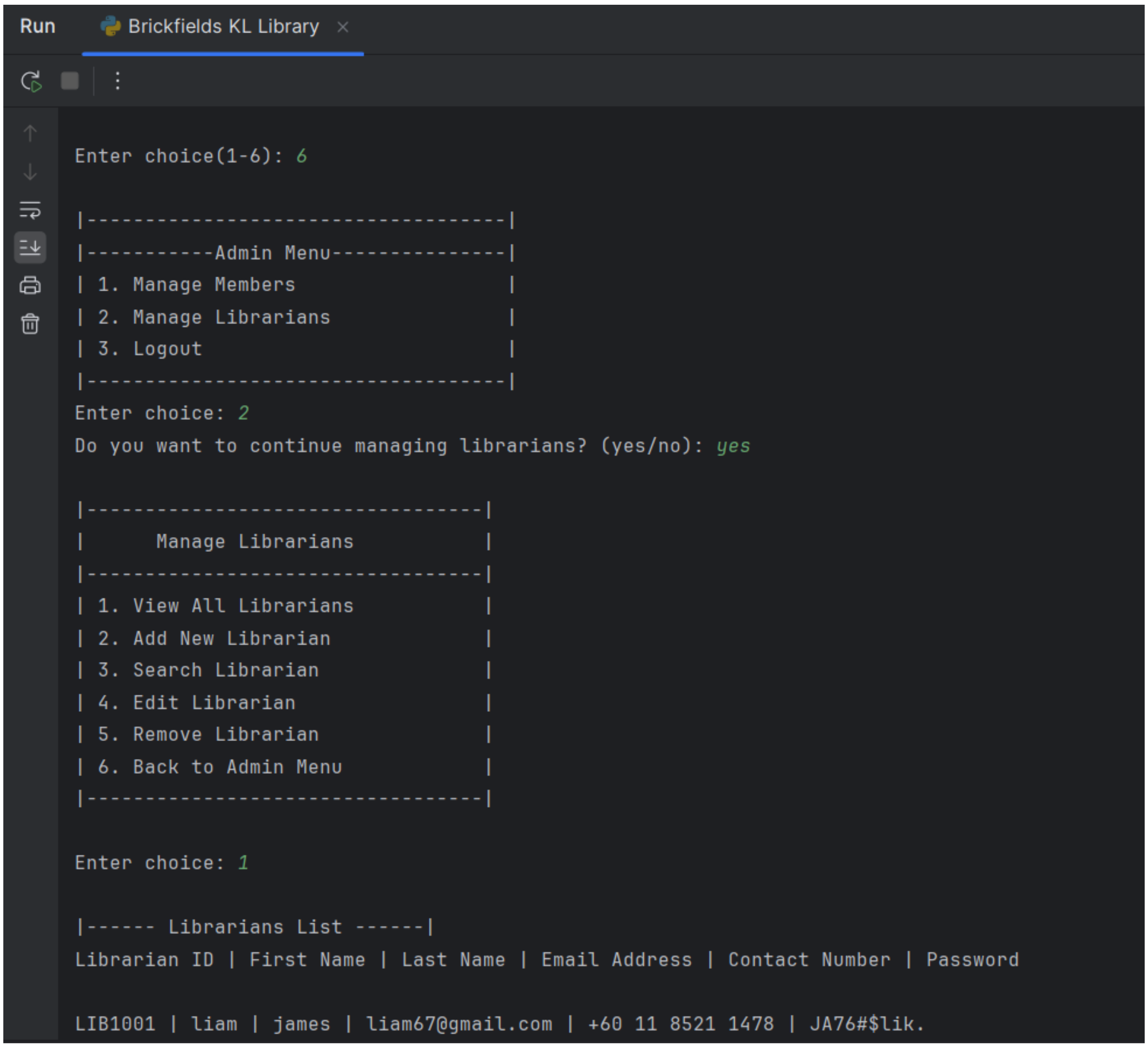
```
Enter choice(1-6): 5
Do you want to remove member by Member ID(1) or member's Email address(2): 1
Enter the Member ID of the member to remove: MEM0046
Member removed.
Do you want to continue managing members? (yes/no): yes

|----------------------------------|
|------Manage Members---------------|
|----------------------------------|
| 1. View All Members              |
| 2. Add New Member                |
| 3. Search Member                 |
| 4. Edit Member                   |
| 5. Remove Member                 |
| 6. Back to Admin Menu            |
|----------------------------------|

Enter choice(1-6): 5
Do you want to remove member by Member ID(1) or member's Email address(2): 2
Enter the email address of the member to remove: Swankiipaki@gmail.com
Member not found!
Do you want to continue managing members? (yes/no): yes

|----------------------------------|
|------Manage Members---------------|
|----------------------------------|
| 1. View All Members              |
| 2. Add New Member                |
```

With option 5, members can be removed from the system. He first successfully removed the member with ID, MEM0046. Then, he entered the email address of a member which was not found. Here in the file, the email address was stored as Swankiipaki@gmail.com, which is not in lowercase. Hence, the system cannot recognise this data.



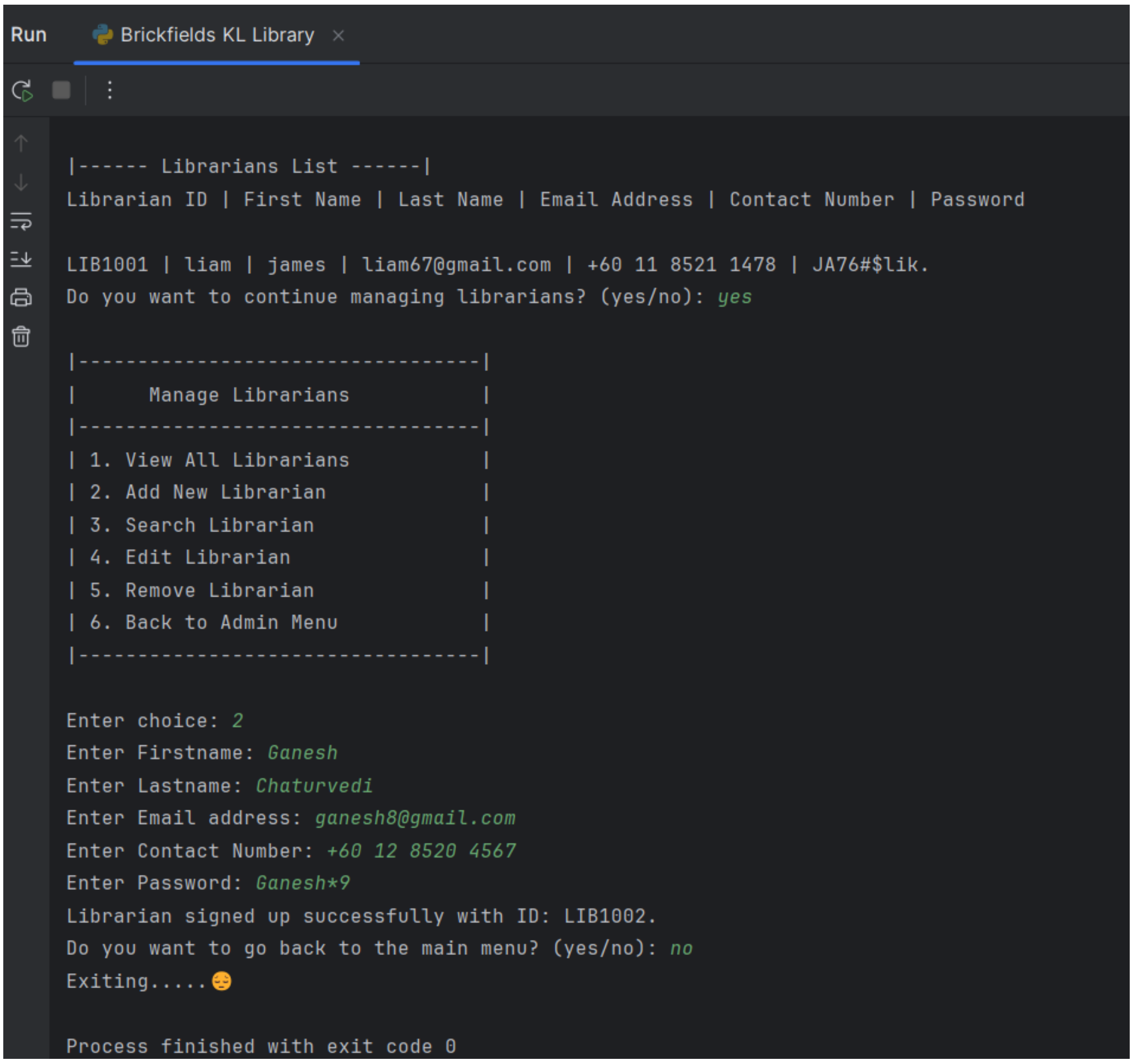
```
|------------------------------------|
|------Manage Members----------------|
|------------------------------------|
| 1. View All Members                |
| 2. Add New Member                  |
| 3. Search Member                   |
| 4. Edit Member                     |
| 5. Remove Member                   |
| 6. Back to Admin Menu              |
|------------------------------------|

Enter choice(1-6): 5
Do you want to remove member by Member ID(1) or member's Email address(2): 2
Enter the email address of the member to remove: jokercartel@gmail.com
Member removed.
Do you want to continue managing members? (yes/no): yes

|------------------------------------|
|------Manage Members----------------|
|------------------------------------|
| 1. View All Members                |
| 2. Add New Member                  |
| 3. Search Member                   |
| 4. Edit Member                     |
| 5. Remove Member                   |
| 6. Back to Admin Menu              |
|------------------------------------|

Enter choice(1-6): 6
```

Joker Cartel was successfully removed from the system be entering his email address.



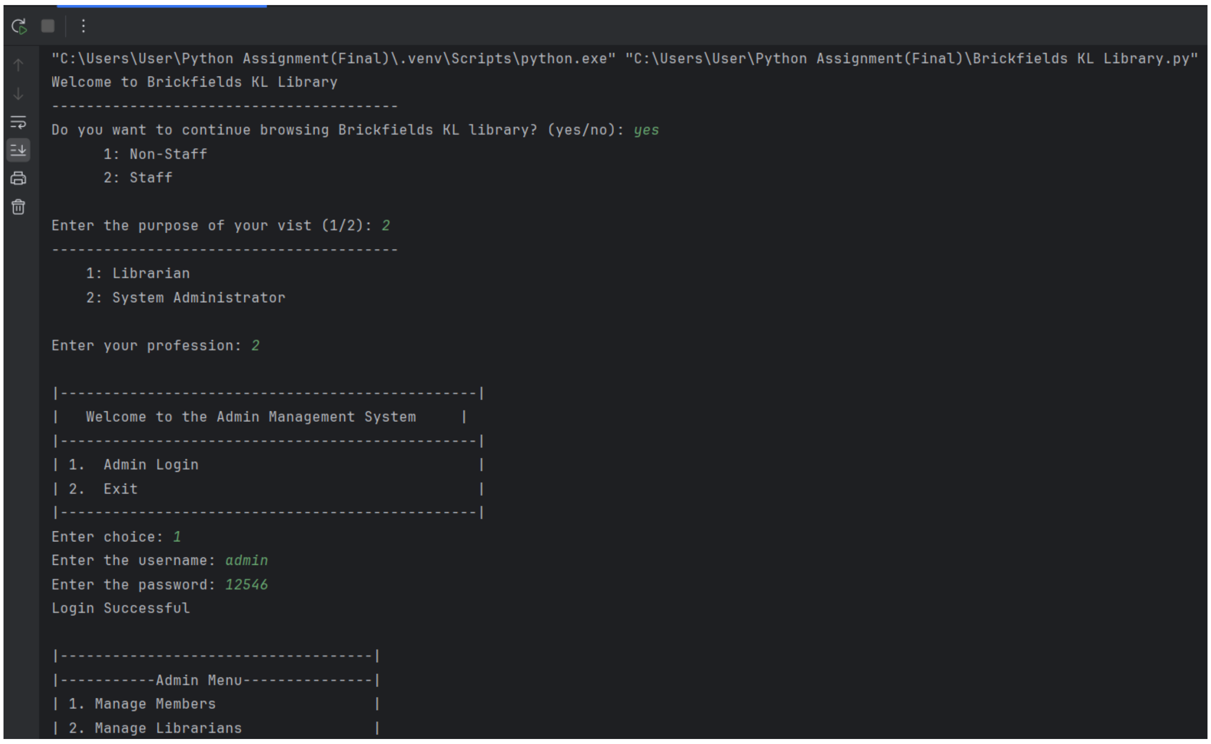
```
Enter choice(1-6): 6

|-----------------------------------|
|-----------Admin Menu--------------|
| 1. Manage Members                 |
| 2. Manage Librarians              |
| 3. Logout                         |
|-----------------------------------|
Enter choice: 2
Do you want to continue managing librarians? (yes/no): yes

|---------------------------------|
|      Manage Librarians          |
|---------------------------------|
| 1. View All Librarians          |
| 2. Add New Librarian            |
| 3. Search Librarian             |
| 4. Edit Librarian               |
| 5. Remove Librarian             |
| 6. Back to Admin Menu           |
|---------------------------------|

Enter choice: 1

|------ Librarians List ------|
Librarian ID | First Name | Last Name | Email Address | Contact Number | Password

LIB1001 | liam | james | liam67@gmail.com | +60 11 8521 1478 | JA76#$lik.
```

In the librarian management system, with option one, the administrator can view all librarians.



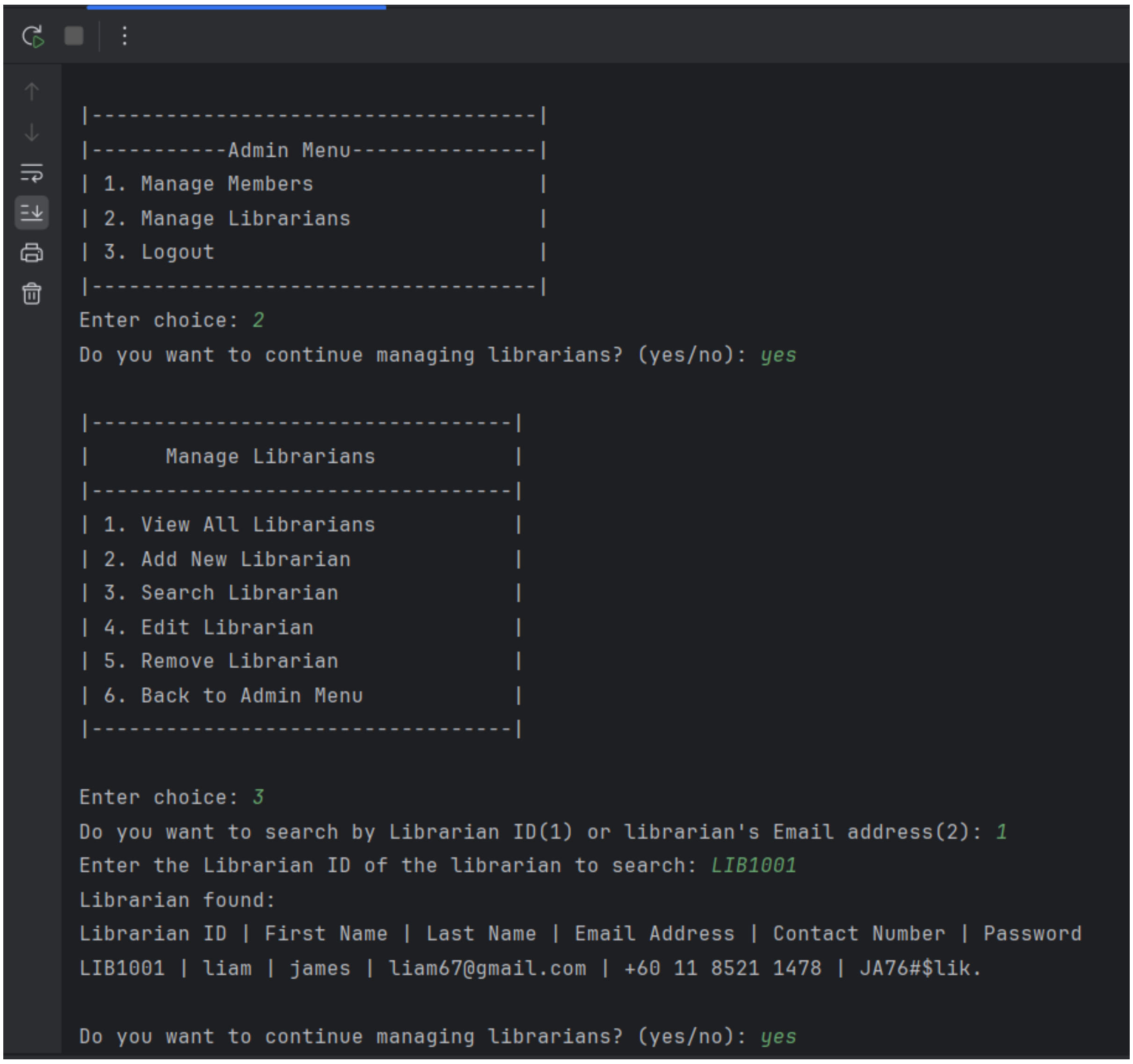
```
|------ Librarians List ------|
Librarian ID | First Name | Last Name | Email Address | Contact Number | Password

LIB1001 | liam | james | liam67@gmail.com | +60 11 8521 1478 | JA76#$lik.
Do you want to continue managing librarians? (yes/no): yes

|----------------------------------|
|      Manage Librarians           |
|----------------------------------|
| 1. View All Librarians           |
| 2. Add New Librarian             |
| 3. Search Librarian              |
| 4. Edit Librarian                |
| 5. Remove Librarian              |
| 6. Back to Admin Menu            |
|----------------------------------|

Enter choice: 2
Enter Firstname: Ganesh
Enter Lastname: Chaturvedi
Enter Email address: ganesh8@gmail.com
Enter Contact Number: +60 12 8520 4567
Enter Password: Ganesh*9
Librarian signed up successfully with ID: LIB1002.
Do you want to go back to the main menu? (yes/no): no
Exiting.....😔

Process finished with exit code 0
```

With option two, new librarian, named Ganesh Chaturvedi was added. His data was stored in the librarians.txt file. Then the administrator chose to exit.



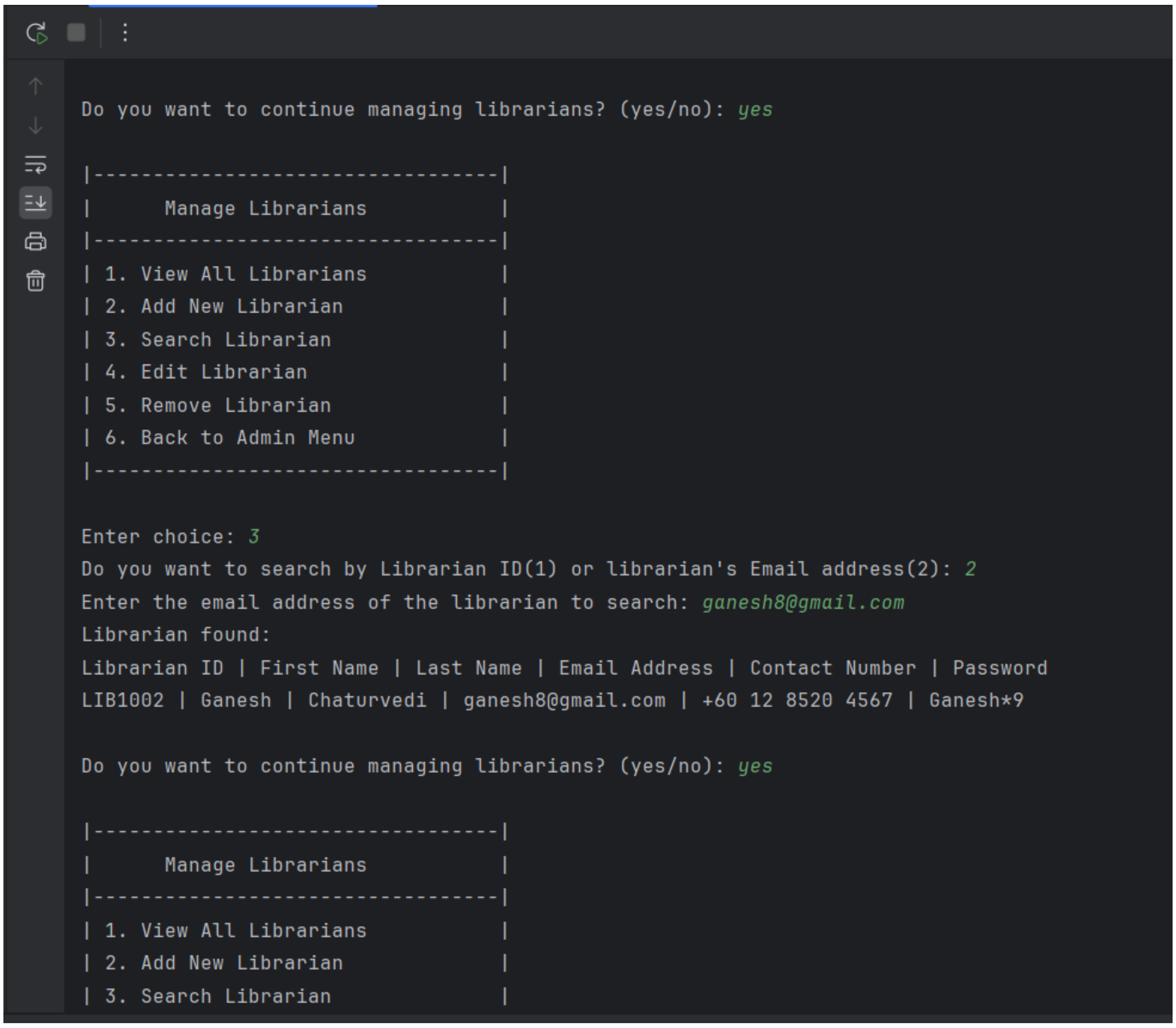
```
"C:\Users\User\Python Assignment(Final)\.venv\Scripts\python.exe" "C:\Users\User\Python Assignment(Final)\Brickfields KL Library.py"
Welcome to Brickfields KL Library
----------------------------------------
Do you want to continue browsing Brickfields KL library? (yes/no): yes
      1: Non-Staff
      2: Staff

Enter the purpose of your vist (1/2): 2
----------------------------------------
    1: Librarian
    2: System Administrator

Enter your profession: 2

|-------------------------------------------------|
|   Welcome to the Admin Management System     |
|-------------------------------------------------|
| 1.  Admin Login                                 |
| 2.  Exit                                        |
|-------------------------------------------------|
Enter choice: 1
Enter the username: admin
Enter the password: 12546
Login Successful

|------------------------------------|
|-----------Admin Menu---------------|
| 1. Manage Members                  |
| 2. Manage Librarians               |
```

He logged in again.



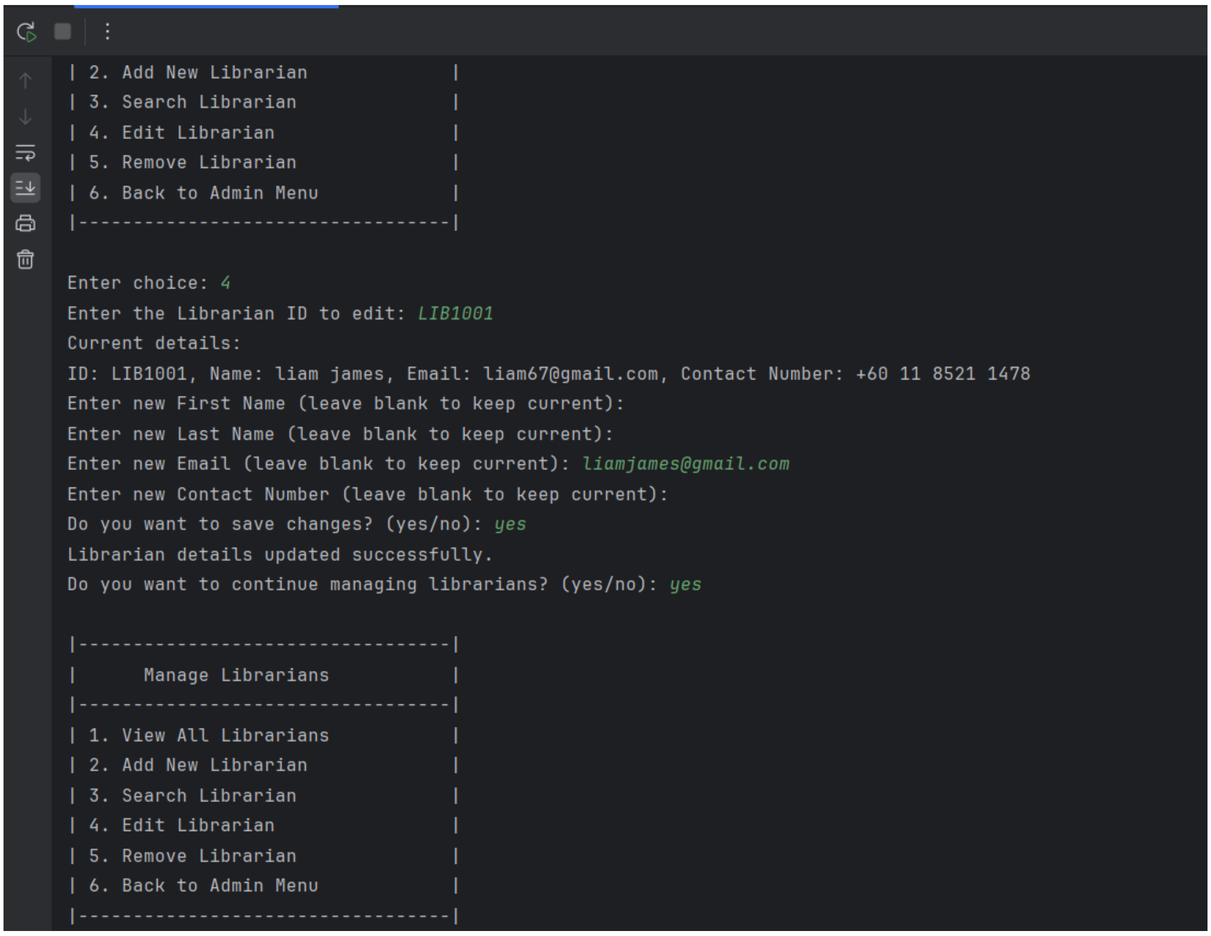
```
|------------------------------------|
|-----------Admin Menu---------------|
| 1. Manage Members                  |
| 2. Manage Librarians               |
| 3. Logout                          |
|------------------------------------|
Enter choice: 2
Do you want to continue managing librarians? (yes/no): yes

|----------------------------------|
|      Manage Librarians           |
|----------------------------------|
| 1. View All Librarians           |
| 2. Add New Librarian             |
| 3. Search Librarian              |
| 4. Edit Librarian                |
| 5. Remove Librarian              |
| 6. Back to Admin Menu            |
|----------------------------------|

Enter choice: 3
Do you want to search by Librarian ID(1) or librarian's Email address(2): 1
Enter the Librarian ID of the librarian to search: LIB1001
Librarian found:
Librarian ID | First Name | Last Name | Email Address | Contact Number | Password
LIB1001 | liam | james | liam67@gmail.com | +60 11 8521 1478 | JA76#$lik.

Do you want to continue managing librarians? (yes/no): yes
```

He searched librarian with librarian ID, LIB1001. Liam’s details was displayed on the screen.



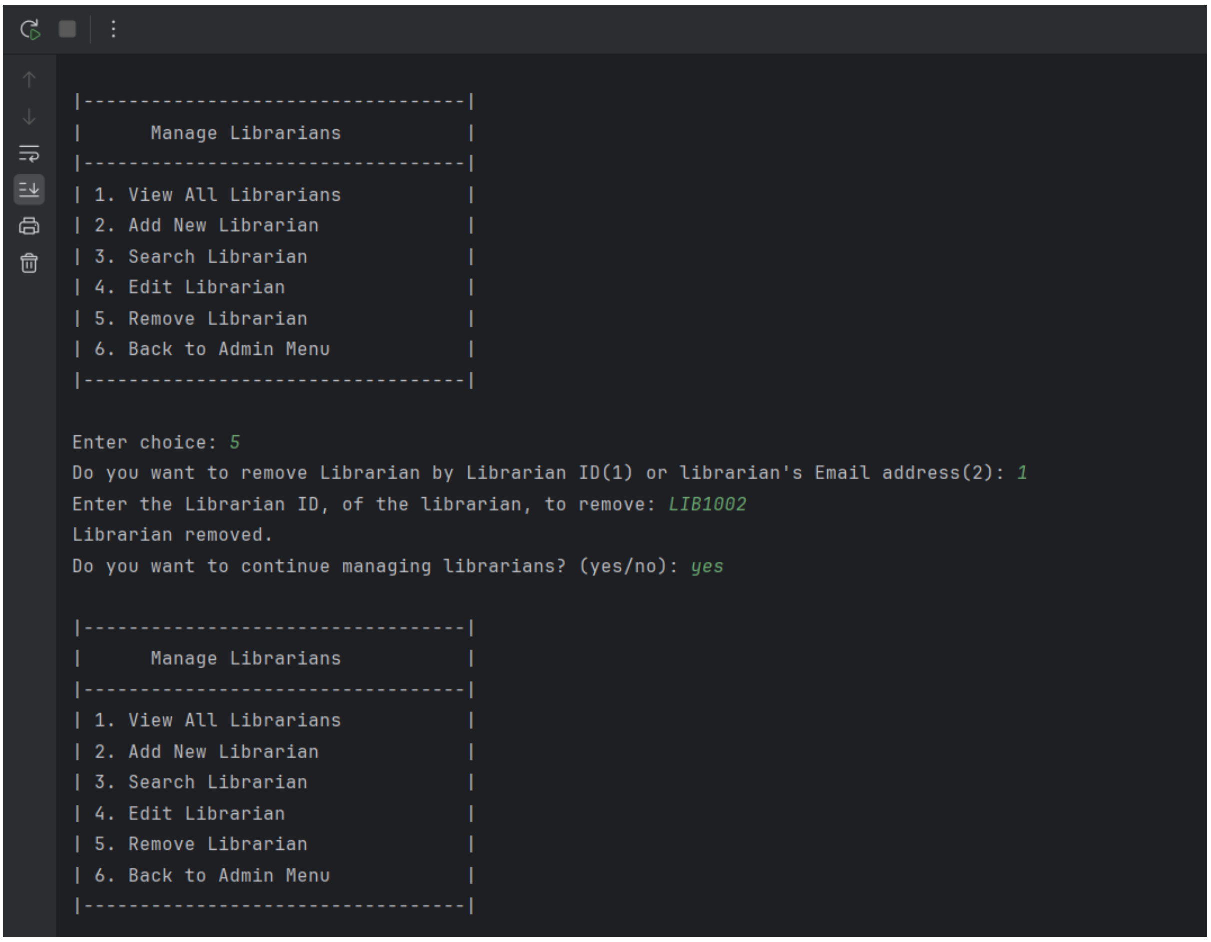
```
Do you want to continue managing librarians? (yes/no): yes

|---------------------------------|
|      Manage Librarians          |
|---------------------------------|
| 1. View All Librarians          |
| 2. Add New Librarian            |
| 3. Search Librarian             |
| 4. Edit Librarian               |
| 5. Remove Librarian             |
| 6. Back to Admin Menu           |
|---------------------------------|

Enter choice: 3
Do you want to search by Librarian ID(1) or librarian's Email address(2): 2
Enter the email address of the librarian to search: ganesh8@gmail.com
Librarian found:
Librarian ID | First Name | Last Name | Email Address | Contact Number | Password
LIB1002 | Ganesh | Chaturvedi | ganesh8@gmail.com | +60 12 8520 4567 | Ganesh*9

Do you want to continue managing librarians? (yes/no): yes

|---------------------------------|
|      Manage Librarians          |
|---------------------------------|
| 1. View All Librarians          |
| 2. Add New Librarian            |
| 3. Search Librarian             |
```

He also searched librarian, Ganesh, with his email address.



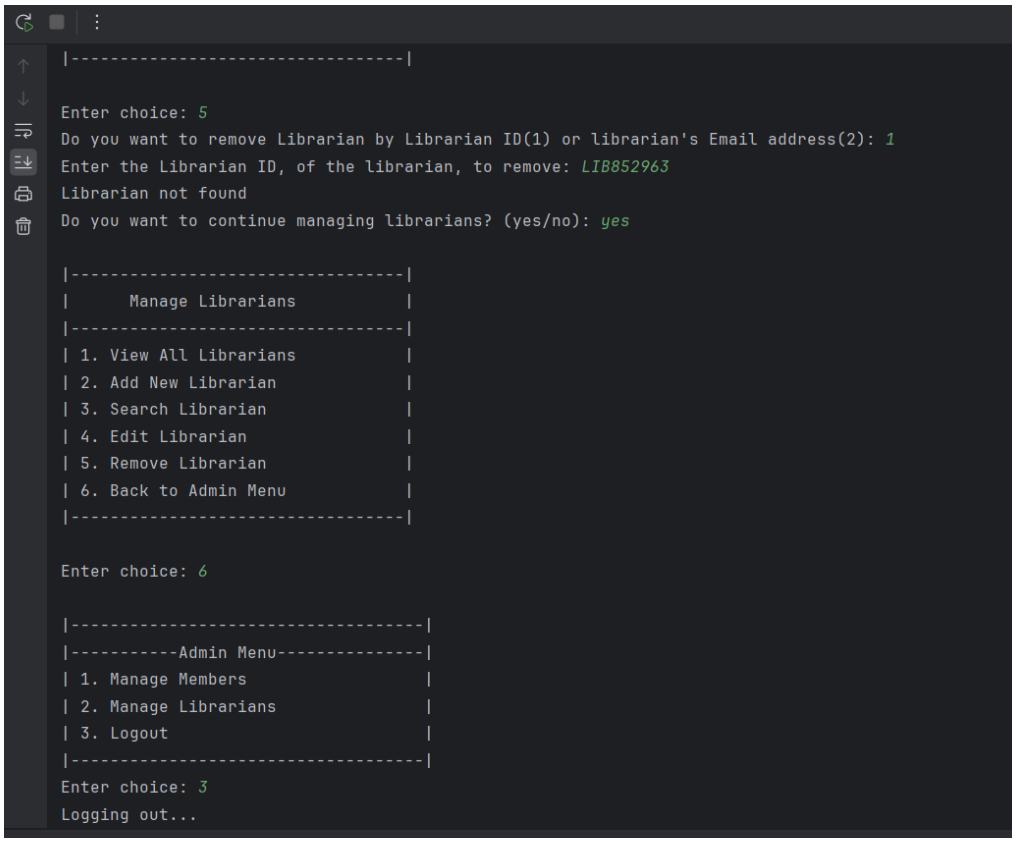
```
| 2. Add New Librarian             |
| 3. Search Librarian              |
| 4. Edit Librarian                |
| 5. Remove Librarian              |
| 6. Back to Admin Menu            |
|----------------------------------|

Enter choice: 4
Enter the Librarian ID to edit: LIB1001
Current details:
ID: LIB1001, Name: liam james, Email: liam67@gmail.com, Contact Number: +60 11 8521 1478
Enter new First Name (leave blank to keep current):
Enter new Last Name (leave blank to keep current):
Enter new Email (leave blank to keep current): liamjames@gmail.com
Enter new Contact Number (leave blank to keep current):
Do you want to save changes? (yes/no): yes
Librarian details updated successfully.
Do you want to continue managing librarians? (yes/no): yes

|----------------------------------|
|      Manage Librarians           |
|----------------------------------|
| 1. View All Librarians           |
| 2. Add New Librarian             |
| 3. Search Librarian              |
| 4. Edit Librarian                |
| 5. Remove Librarian              |
| 6. Back to Admin Menu            |
|----------------------------------|
```

Liam’s email address was edited and successfully stored in the librarians.txt file.



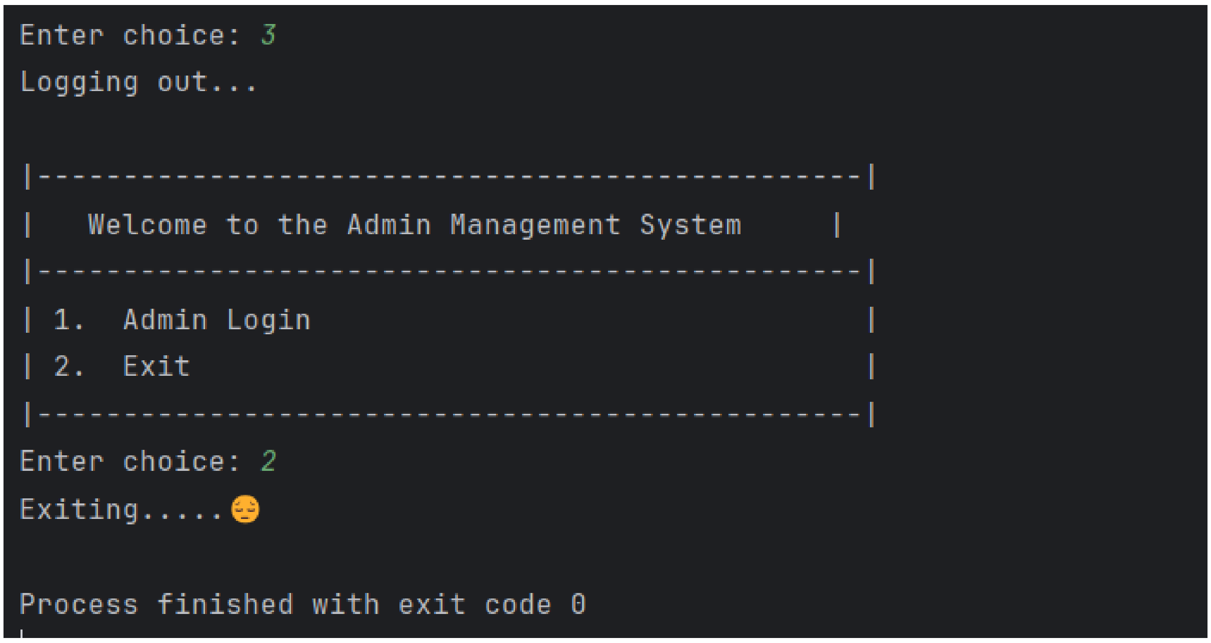
```
|---------------------------------|
|      Manage Librarians          |
|---------------------------------|
| 1. View All Librarians          |
| 2. Add New Librarian            |
| 3. Search Librarian             |
| 4. Edit Librarian               |
| 5. Remove Librarian             |
| 6. Back to Admin Menu           |
|---------------------------------|

Enter choice: 5
Do you want to remove Librarian by Librarian ID(1) or librarian's Email address(2): 1
Enter the Librarian ID, of the librarian, to remove: LIB1002
Librarian removed.
Do you want to continue managing librarians? (yes/no): yes

|---------------------------------|
|      Manage Librarians          |
|---------------------------------|
| 1. View All Librarians          |
| 2. Add New Librarian            |
| 3. Search Librarian             |
| 4. Edit Librarian               |
| 5. Remove Librarian             |
| 6. Back to Admin Menu           |
|---------------------------------|
```

Here, the administrator removed the librarian with librarian ID, LIB1002.



```
|--------------------------------|

Enter choice: 5
Do you want to remove Librarian by Librarian ID(1) or librarian's Email address(2): 1
Enter the Librarian ID, of the librarian, to remove: LIB852963
Librarian not found
Do you want to continue managing librarians? (yes/no): yes

|--------------------------------|
|      Manage Librarians         |
|--------------------------------|
| 1. View All Librarians         |
| 2. Add New Librarian           |
| 3. Search Librarian            |
| 4. Edit Librarian              |
| 5. Remove Librarian            |
| 6. Back to Admin Menu          |
|--------------------------------|

Enter choice: 6

|----------------------------------|
|----------Admin Menu--------------|
| 1. Manage Members                |
| 2. Manage Librarians             |
| 3. Logout                        |
|----------------------------------|
Enter choice: 3
Logging out...
```

The administrator exited the manage librarian menu and entered the admin menu.



```
Enter choice: 3
Logging out...

|------------------------------------------------|
|   Welcome to the Admin Management System      |
|------------------------------------------------|
| 1.  Admin Login                                |
| 2.  Exit                                       |
|------------------------------------------------|
Enter choice: 2
Exiting.....😔

Process finished with exit code 0
```

He logged out of the admin menu and exited the Admin Management System, with the “Logging out” and “Exiting…” messages printed.

7.0 Conclusion

With the automated system, Brickfields KL Library Management System is now more innovative and efficient. Our team developed this system for the optimal operation of Brickfields KL Community Library. Members can efficiently return their borrowed books, manage their payments, view books in the system, and update their profiles. Librarians can also update details about the books in the system, more accurately. System administrators can also efficiently manage the librarians' and the members' details.

Moreover, the library can benefit from reduced manual errors will and many more. This project really emphasises the importance of Python in problem solving.

8.0 Workload Matrix

Component	**ASEEL M. H. SANAD (TP080667)**	**SARA ELWALID HASSAN FAGIR (TP078376)**	**MAANVI GOORBIN (TP081121)**	**SHUKRI DAHIR AHMED (TP073809)**	**Total**
System Development					
g) System Administrator	**25%**	**25%**	**25%**	**25%**	**100%**
g) Librarian	**25%**	**25%**	**25%**	**25%**	**100%**
g) Library Member	**25%**	**25%**	**25%**	**25%**	**100%**
Documentation					
g) Introduction and assumptions	**25%**	**25%**	**25%**	**25%**	**100%**
g) Design (Pseudocode or Flowchart)	**25%**	**25%**	**25%**	**25%**	**100%**
g) Program Source Code with Explanation	**25%**	**25%**	**25%**	**25%**	**100%**
g) Input/Output Screenshots with Explanation	**25%**	**25%**	**25%**	**25%**	**100%**

9.0 References

Beverlee Brick. (2018, August 21). The Disadvantages of a Manual Operating System in a Library. Bizfluent. https://bizfluent.com/12746087/the-disadvantages-of-a-manual-operating-system-in-a-library

W3Schools.com. (n.d.).

https://www.w3schools.com/python/python_regex.asp

Trivedi, S. (2023, January 31). Library Management System Project in Python | Scaler Topics.

https://www.scaler.com/topics/library-management-system-project-in-python/