
| | | |
|---|---|---|
| **Module Code** | **:** | CT088-0-M-PIP – Programming In Python |
| **Intake Code** | **:** | APDMF2310AI |
| **Lecturer Name** | **:** | Ms. Nurul Izzatie Husna Binti Muhamad Fauzi |
| **Hand in Date** | **:** | Thursday, 9 November 2023, before 10.00 PM. |
| **Student Name:** | **:** | Nurlan Amanov |
| **Student Number:** | **:** | TP077526 |

# TABLE OF CONTENTS

# 1. INTRODUCTION AND ASSUMPTIONS

Library Management System is a system created in the Python programming language, using a json database. This system allows you to enter and manipulate information about books in the database. The books themselves in the database contain such information as: title, author, year of publication. LMS contains such functions as:

1) Adding a book;
2) Changing the information of an existing book;
3) Viewing the list of all books in the database;
4) Search for a book by one of three criteria;
5) Deleting a book from the system;
6) Additional features not listed in the numbered list inside the programme: help with menus and a randomiser showing a randomly selected book from the database.

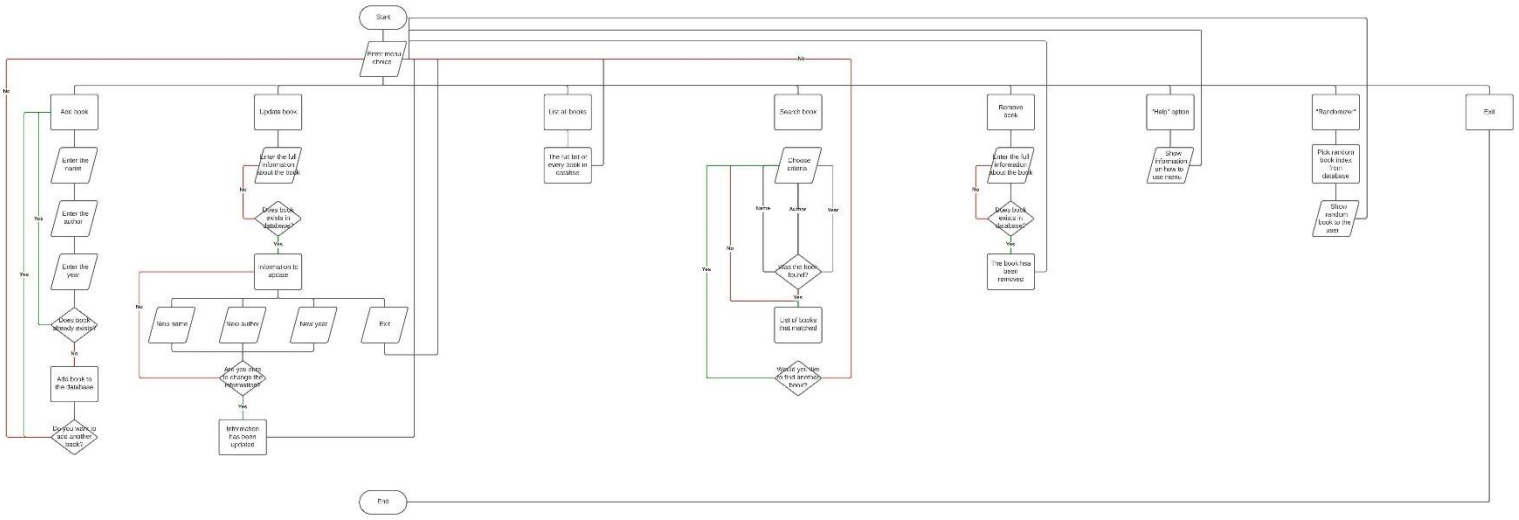
All tests have been carried out by the creator of this programme and have been considered from the perspective of the future user, thus all functions have been created intuitive and simple, which makes it easy to navigate.

Since this programme is used within the Python environment, all menu inputs have been reduced to numbers, in cases where the user's intentions are confirmed, it is necessary to enter the words "YES" or "NO". Also in some cases it is necessary to enter the full information of the book stored in the database. This is used to reduce the risk of selecting the wrong book from the system. The exception where whole words need to be entered are additional features that are not numbered in the main programme menu, such as: "help" and "randomiser".

The "help" function helps the user to quickly navigate through the main menu, while the "randomiser" function helps the user to select a random book. This function can be used in case the user wants to read any book available in the system, and the selection of this book is completely random.

## 2. DESIGN OF THE PROGRAM

Below is a screenshot taken from the "Lucidchart" site that made this Flowchart possible. For a more detailed study of the screenshot, it will be attached along with other files.



# 3. PROGRAMING CONCEPTS WITH SOURCE CODE FOR EXPLANATION

## 3.1. Python Code

```
# NURLAN AMANOV
# TP077526

import json  # using it as database for the system
import random  # using it to get random number


def addBook():  # function to add book to the database system, to add book it asks for three
    print('Welcome to the process of adding book into database system!')
    while True:
        database = open("bookDatabase.json")  # every function has those three lines at the start because
        data = json.load(database)  # everytime it will use updated information from database
        len_data = len(data)

        name = input('Please, enter name of the book: ')
        author = input('Please, enter the author of the book: ')
        year = input('Please, enter the year when book was published: ')

        existence = 0
        for book in range(len_data):
            if name == data[book]["name"]:
                if author == data[book]["author"]:
                    if year == data[book]["year"]:
                        print('')
                        print('This book already exists in database')   # also, we have to check if the book already
                        # exists
                        print('')
                        existence = 1

        if existence == 0:
            data.append({
                "name": name,
                "author": author,
                "year": year
            })
            with open("bookDatabase.json", 'w') as json_file:
                json.dump(data, json_file, indent=4, separators=(',', ': '))

            print('')
            print('The book has been added to the database!')
            print('')

            continue_way = 0
            while True:
                print('Would you like to add another book? ("YES" or "NO")')
                continue_choice = input('Enter your choice here: ')
                continue_choice = continue_choice.upper()

                if continue_choice == 'YES':
```

```
                    print('')
                    continue_way = 1
                    break
                elif continue_choice == 'NO':
                    print('')
                    continue_way = 0
                    break
                else:
                    print('')
                    print('!WRONG INPUT!')
                    print('')
                    continue
            if continue_way == 1:
                continue
            elif continue_way == 0:
                break
        else:
            existence = 0
            continue


def updateBook():   # process of updating information about book, it asks for
name, author and year of the book
    print("Welcome to the process of updating book's information!")
    while True:
        database = open("bookDatabase.json")
        data = json.load(database)
        len_data = len(data)

        print('Please type full information about the book in format: Name -
Author - Year')
        print('Example: A Tale of Two Cities - Charles Dickens - 1859')
        information = input('Type here: ')
        print('')

        information = information.strip()
        spl_information = information.split(' - ')
        matched_info = 0
        book_index = 0

        try:
            for book in range(len_data):
                if data[book]["name"] == spl_information[0]:
                    if data[book]["author"] == spl_information[1]:
                        if data[book]["year"] == spl_information[2]:
                            matched_info = 1
                            book_index = book
        except IndexError:
            matched_info = 0

        if matched_info == 0:
            print('The book has not been found')
            print('')
            continue
        else:
            while True:
                print('What information you like to update?')
```

```
                print('1. Name')
                print('2. Author')
                print('3. Year')
                print('4. Exit')
                upd_choice = input('Enter your choice: ')
                print('')

                if upd_choice == '1':
                    while True:
                        new_name = input('Type new name: ')

                        confirmation = input(f'Are you sure you want to
change from "{data[book_index]["name"]}" to '
                                             f'"{new_name}" (YES or NO)?: ')
                        print('')
                        confirmation = confirmation.upper()

                        if confirmation == 'YES':
                            print("The book's name has been updated")
                            print('')
                            data[book_index]["name"] = new_name
                            with open("bookDatabase.json", 'w') as json_file:
                                json.dump(data, json_file, indent=4,
separators=(',', ': '))

                            break
                        elif confirmation == 'NO':
                            break
                        else:
                            print('!WRONG INPUT!')
                            print('')
                            continue
                elif upd_choice == '2':
                    while True:
                        new_author = input('Type new author: ')

                        confirmation = input(f'Are you sure you want to
change from "{data[book_index]["author"]}" to '
                                             f'"{new_author}" (YES or NO)?:
')
                        print('')
                        confirmation = confirmation.upper()

                        if confirmation == 'YES':
                            print("The book's author has been updated")
                            print('')
                            data[book_index]["author"] = new_author
                            with open("bookDatabase.json", 'w') as json_file:
                                json.dump(data, json_file, indent=4,
separators=(',', ': '))

                            break
                        elif confirmation == 'NO':
                            break
                        else:
                            print('!WRONG INPUT!')
                            print('')
```

```
                            continue
                elif upd_choice == '3':
                    while True:
                        new_year = input('Type new year: ')

                        confirmation = input(f'Are you sure you want to
change from "{data[book_index]["year"]}" to '
                                             f'"{new_year}" (YES or NO)?: ')
                        print('')
                        confirmation = confirmation.upper()

                        if confirmation == 'YES':
                            print("The book's year has been updated")
                            print('')
                            data[book_index]["year"] = new_year
                            with open("bookDatabase.json", 'w') as json_file:
                                json.dump(data, json_file, indent=4,
separators=(',', ': '))

                            break
                        elif confirmation == 'NO':
                            break
                        else:
                            print('!WRONG INPUT!')
                            print('')
                            continue
                elif upd_choice == '4':
                    break
                else:
                    print('!WRONG INPUT!')
                    print('')
                continue
        break


def listAllBook():  # listing all the books
    database = open("bookDatabase.json")
    data = json.load(database)
    len_data = len(data)

    print("Here's the list of all books in database: ")
    list_counter = 1

    for book in range(len_data):
        print(f'{list_counter}. "{data[book]["name"]}" by
{data[book]["author"]}, {data[book]["year"]}')
        list_counter += 1
    print('')


def searchBook():   # searching for book; do not need the full name of the
book or author, search by three or more
    # consecutive similar letters
    print("Welcome to the process of finding book!")
    while True:
        database = open("bookDatabase.json")
        data = json.load(database)
```

```
        len_data = len(data)

        print('Choose the criteria to find the book: ')
        print('1. by Name')
        print('2. by Author')
        print('3. by Year')
        print('4. Exit')
        find_choice = input('Enter your choice here: ')
        print('')

        if find_choice == '1':
            while True:
                find_name = input('Please, enter the name (min 3 characters):
')
                print('')
                if len(find_name) < 3:
                    print('Minimum 3 characters')
                    print('')
                else:
                    break
            find_name = find_name.upper()
            list_counter = 0
            print("Here's the results: ")
            for book in range(len_data):
                book_name_upper = data[book]["name"].upper()
                if find_name in book_name_upper:
                    list_counter += 1
                    print(f'{list_counter}. "{data[book]["name"]}" by
{data[book]["author"]}, {data[book]["year"]}')
            if list_counter == 0:
                print('No results found')
        if find_choice == '2':
            while True:
                find_author = input('Please, enter the author (min 3
characters): ')
                print('')
                if len(find_author) < 3:
                    print('Minimum 3 characters')
                    print('')
                else:
                    break
            find_author = find_author.upper()
            list_counter = 0
            print("Here's the results: ")
            for book in range(len_data):
                book_author_upper = data[book]["author"].upper()
                if find_author in book_author_upper:
                    list_counter += 1
                    print(f'{list_counter}. "{data[book]["name"]}" by
{data[book]["author"]}, {data[book]["year"]}')
            if list_counter == 0:
                print('No results found')
        if find_choice == '3':
            find_year = input('Please, enter the year: ')
            print('')
            list_counter = 0
            print("Here's the results: ")
```

```
            for book in range(len_data):
                if data[book]["year"] == find_year:
                    list_counter += 1
                    print(f'{list_counter}. "{data[book]["name"]}" by
{data[book]["author"]}, {data[book]["year"]}')
            if list_counter == 0:
                print('No results found')
        if find_choice == '4':
            break
        print('')


def removeBook():   # removes book from the database, needs full info about
the book to remove it
    print("Welcome to the process of removing book form the database!")
    while True:
        database = open("bookDatabase.json")
        data = json.load(database)
        len_data = len(data)

        print('Please, type the full information about the book in format:
Name - Author - Year')
        print('Example: A Tale of Two Cities - Charles Dickens - 1859')
        information = input('Type here: ')
        information = information.strip()
        div_info = information.split(' - ')

        existence = 0
        for book in range(len_data):
            if data[book]["name"] == div_info[0]:
                if data[book]["author"] == div_info[1]:
                    if data[book]["year"] == div_info[2]:
                        existence = 1
                        data.pop(book)
                        with open("bookDatabase.json", 'w') as json_file:
                            json.dump(data, json_file, indent=4,
separators=(',', ': '))
                        print('')
                        print("The book has been removed")
                        print('')
                        break
        if existence == 0:
            print('')
            print('The book has not been found')
            print('')
        else:
            existence = 0

        continue_way = 0
        while True:
            print('Would you like to delete another book? ("YES" or "NO")')
            continue_choice = input('Enter your choice here: ')
            continue_choice = continue_choice.upper()
            print('')

            if continue_choice == 'YES':
                continue_way = 1
```

```
                break
            elif continue_choice == 'NO':
                continue_way = 0
                break
            else:
                print('!WRONG INPUT!')
                print('')
                continue

        if continue_way == 1:
            continue
        elif continue_way == 0:
            break


def helpMenu():  # simple help function that every system has, add more
functionality to the menu
    print('To interact with the menu you need to type numbers between 1-6.
Each number has its own function that can'
          ' help you to interact with the Library Management System')
    goback = input('Type anything to go back to the menu ')
    print('')


def randomBook():   # function to give random book from the database, add
more functionality to the menu
    database = open("bookDatabase.json")
    data = json.load(database)
    len_data = len(data)

    randomIndex = random.randint(0, len_data - 1)
    print(f'Random book for you: {data[randomIndex]["name"]} by
{data[randomIndex]["author"]}, '
          f'{data[randomIndex]["year"]}')
    goback = input('Type anything to go back to the menu ')
    print('')


def main():  # the main menu of the system
    while True:
        print('**********************')
        print('1. Add Book')
        print('2. Update Book')
        print('3. List All Books')
        print('4. Search Book')
        print('5. Remove Book')
        print('6. Exit')
        print('**********************')
        print("Type 'help' to get more information about menu")
        print('')
        print(
            'Try out our new feature - "Randomizer" to get random book from
our database that you might want to read!')
        print("Type 'random' to use it")
        print('')
        choice = input('Enter your choice: ')
        print('')
```

```
        if choice == '1':
            addBook()
        elif choice == '2':
            updateBook()
        elif choice == '3':
            listAllBook()
        elif choice == '4':
            searchBook()
        elif choice == '5':
            removeBook()
        elif choice == '6':
            exit()
        elif choice == 'help':
            helpMenu()
        elif choice == 'random':
            randomBook()
        else:
            print('!WRONG INPUT!')
            print('')


if __name__ == "__main__":
    main()
```

3.2. Json Database

```
[
    {
        "name": "The Posthumous Papers of the Pickwick Club",
        "author": "Charles Dickens",
        "year": "1836"
    },
    {
        "name": "The Da Vinci",
        "author": "Dan Brown",
        "year": "2003"
    },
    {
        "name": "Black Beauty",
        "author": "Anna Sewell",
        "year": "1877"
    },
    {
        "name": "Angels and Demons",
        "author": "Dan Brown",
        "year": "2000"
    },
    {
        "name": "New Test",
        "author": "New Author",
        "year": "New year"
    },
    {
        "name": "Test3",
```

```
        "author": "Test3",
        "year": "Test3"
    },
    {
        "name": "Test4",
        "author": "Test4",
        "year": "Test4"
    },
    {
        "name": "Test5",
        "author": "Test5",
        "year": "Test5"
    }
]
```

# 4. SCREENSHOTS OF SAMPLE INPUT/OUTPUT FOR EXPLANATION

## 4.1. Main Menu and Add Book

In Figure 1, we can see starting main menu of our program. Here we can see all the functions that we can use: add, update, list all, search, remove books. Also two additional functions: ‘help’ and ‘randomizer’

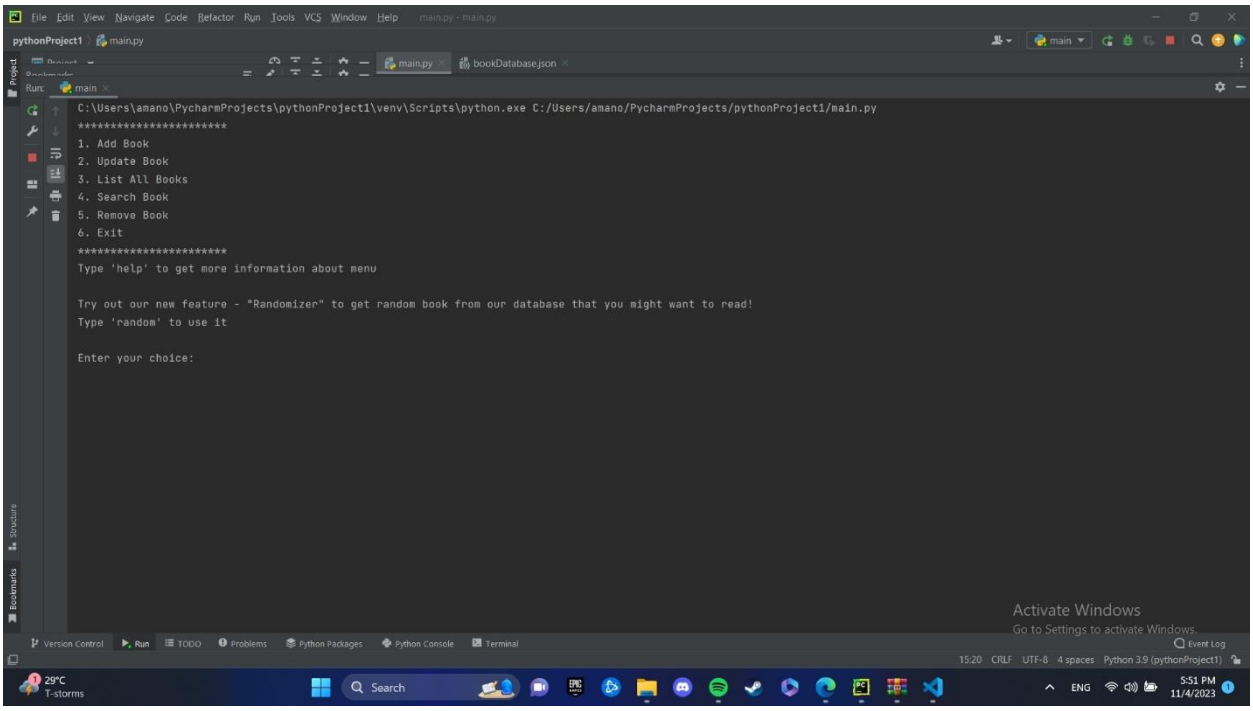



Figure 1.

As we can see in Figure 2, if we enter wrong option rather than showed in main menu program will output ‘!WRONG INPUT!’ message.

```
1. Add Book
2. Update Book
3. List All Books
4. Search Book
5. Remove Book
6. Exit
**********
Type 'help' to get more information about menu

Try out our new feature - "Randomizer" to get random book from our database that you might want to read!
Type 'random' to use it

Enter your choice: test

!WRONG INPUT!
```

Figure 2.

In the Figure 3, after we enter our first choice in the menu which is ‘Add Book’ we can see first two messages which is welcome message and then after program asks to enter the name of the book that we want to add

```
**********************
1. Add Book
2. Update Book
3. List All Books
4. Search Book
5. Remove Book
6. Exit
**********************
Type 'help' to get more information about menu

Try out our new feature - "Randomizer" to get random book from our database that you might want to read!
Type 'random' to use it

Enter your choice: 1

Welcome to the process of adding book into database system!
Please, enter name of the book:
```

Figure 3.

After the name program will ask for the author and year information the same way. In Figure 4, it shown that if book that we want to add already in the database we will see message about it.

```
Welcome to the process of adding book into database system!
Please, enter name of the book: Test
Please, enter the author of the book: Test
Please, enter the year when book was published: Test

This book already exists in database
```

Figure 4.

In Figure 5, it shown that if there is no book in the database that we want to add it will show message that the book has been added with the following messages about continuing our process of adding another book to the database. It provides two simple choices: “YES” or “NO”.

```
Please, enter name of the book: Test2
Please, enter the author of the book: Test2
Please, enter the year when book was published: Test2

The book has been added to the database!

Would you like to add another book? ("YES" or "NO")
Enter your choice here:
```

Figure 5.

If we type option “YES” then it will comeback to the process of adding new book by asking for the same information which is name, author and year as it shown in Figure 6.

```
Would you like to add another book? ("YES" or "NO")
Enter your choice here: yes

Please, enter name of the book:
```

Figure 6.

If our choice other than “YES” or “NO” it will show error message ‘!WRONG INPUT!’ and ask the same question again as we can see in Figure 7.

```
Would you like to add another book? ("YES" or "NO")
Enter your choice here: s

!WRONG INPUT!

Would you like to add another book? ("YES" or "NO")
Enter your choice here:
```

Figure 7.

In Figure 8 shown situation if we choose option ‘NO’ which send us back to the main menu of the system.

```
Would you like to add another book? ("YES" or "NO")
Enter your choice here: no

**********************
1. Add Book
2. Update Book
3. List All Books
4. Search Book
5. Remove Book
6. Exit
**********************
Type 'help' to get more information about menu

Try out our new feature - "Randomizer" to get random book from our database that you might want to read!
Type 'random' to use it

Enter your choice:
```

Figure 8.

4.2. Update Book

Onto to the next function which is update book. As it shown in Figure 8, after we type choice 2 from the main menu we can see welcoming message and example on how to input information correctly in case for our function to work.

```
Enter your choice: 2

Welcome to the process of updating book's information!
Please type full information about the book in format: Name - Author - Year
Example: A Tale of Two Cities - Charles Dickens - 1859
Type here:
```

Figure 9.

In Figure 10, it is shown that if the information that we wrote incorrectly or such a book basically does not exists in database we are going to see message about it.

```
Please type full information about the book in format: Name - Author - Year
Example: A Tale of Two Cities - Charles Dickens - 1859
Type here: test

The book has not been found

Please type full information about the book in format: Name - Author - Year
Example: A Tale of Two Cities - Charles Dickens - 1859
Type here:
```

Figure 10.

If the information is right and the book exists in the database we will see next menu, where we need to choose the type of information that we want to edit about book that we chose, as it shown in Figure 11.

```
Please type full information about the book in format: Name - Author - Year
Example: A Tale of Two Cities - Charles Dickens - 1859
Type here: Test - Test - Test

What information you like to update?
1. Name
2. Author
3. Year
4. Exit
Enter your choice:
```

Figure 11.

If we choose first option to edit name of the book we need to enter new name and confirm it, as it shown in Figure 12.

```
Type new name: New Test
Are you sure you want to change from "Test" to "New Test" (YES or NO)?:
```

Figure 12.

In Figure 13, if our choice is not “YES” or “NO” we will see error message ‘!WRONG INPUT!’ and it will send us back to the point where we need to enter the new name once again.

```
Type new name: New Test
Are you sure you want to change from "Test" to "New Test" (YES or NO)?: s

!WRONG INPUT!

Type new name:
```

Figure 13.

If we choose option ‘NO’ it will send us back to the menu where we need to choose type of information that we want to edit, that we can see in the Figure 14.

```
Type new name: New Test
Are you sure you want to change from "Test" to "New Test" (YES or NO)?: no

What information you like to update?
1. Name
2. Author
3. Year
4. Exit
Enter your choice:
```

Figure 14.

If choose option ‘YES’ it will update book’s name and send us back to the menu, where we need to choose the type of information that we need to update, as it shown in the Figure 15.

```
Type new name: New Test
Are you sure you want to change from "Test" to "New Test" (YES or NO)?: yes

The book's name has been updated

What information you like to update?
1. Name
2. Author
3. Year
4. Exit
Enter your choice:
```

Figure 15.

The same process, that we did to change the name of the book that exists in the database, goes with author and year of the book and we can see it in Figure 16.

```
What information you like to update?
1. Name
2. Author
3. Year
4. Exit
Enter your choice: 2

Type new author: New Author
Are you sure you want to change from "Test" to "New Author" (YES or NO)?: yes

The book's author has been updated

What information you like to update?
1. Name
2. Author
3. Year
4. Exit
Enter your choice: 3

Type new year: New year
Are you sure you want to change from "Test" to "New year" (YES or NO)?: yes

The book's year has been updated
```

Figure 16.

In Figure 17, after we updated the book’s information we can go back to the main menu through the option of ‘Exit’ in menu of ‘Update Book’.

```
What information you like to update?
1. Name
2. Author
3. Year
4. Exit
Enter your choice: 4

**********************
1. Add Book
2. Update Book
3. List All Books
4. Search Book
5. Remove Book
6. Exit
**********************
Type 'help' to get more information about menu

Try out our new feature - "Randomizer" to get random book from our database that you might want to read!
Type 'random' to use it

Enter your choice:
```

Figure 17.

### 4.3. List All Books

The third option in the main menu is ‘List All Books’ which is basically provide us the full list of books that exists in the database. We can see the whole process on Figure 18.

```
Enter your choice: 3

Here's the list of all books in database:
1. "The Posthumous Papers of the Pickwick Club" by Charles Dickens, 1836
2. "The Da Vinci" by Dan Brown, 2003
3. "Black Beauty" by Anna Sewell, 1877
4. "Angels and Demons" by Dan Brown, 2000
5. "New Test" by New Author, New year
6. "Test2" by Test2, Test2
7. "Test3" by Test3, Test3
8. "Test4" by Test4, Test4
9. "Test5" by Test5, Test5
```

Figure 18.

## 4.4. Search Book

The next option is ‘Search Book’ function which helps us to find the book in LMS. In Figure 19, after we chose option 4 we see menu where we need to enter on what criteria book should be chosen from which is name, author and year.

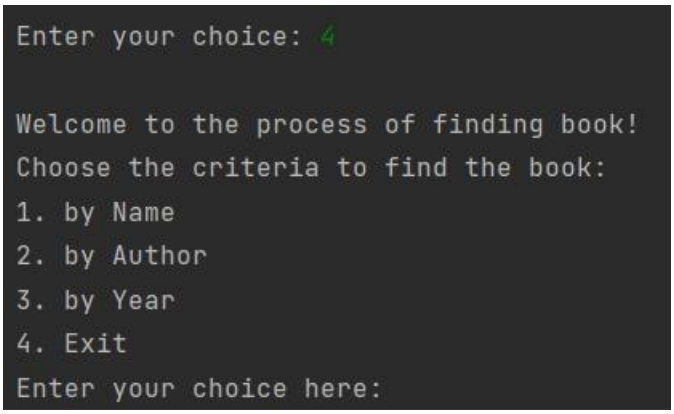



Figure 19.

If we want to find book by the name we need to enter three consecutive characters that appears in the name of book that we are looking for, as it shown in Figure 20.

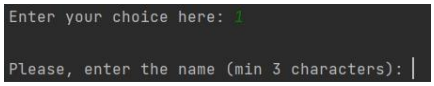



Figure 20.

In Figure 21, it shown that if the number of characters that we enter is less than 3 we will see error message.

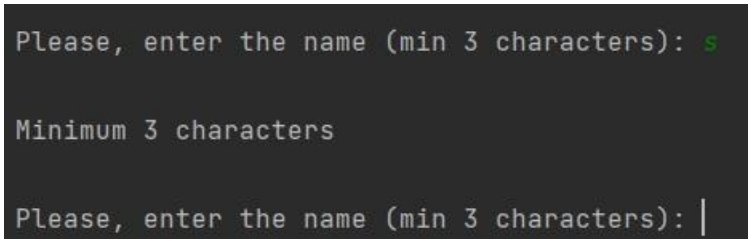



Figure 21.

If we enter it right and there is matching names it will provide us list of every book that contains that combination of characters that we input and send us back to the menu of ‘Search Book’ function. We can see the whole process in Figure 22.

```
Please, enter the name (min 3 characters): test

Here's the results:
1. "New Test" by New Author, New year
2. "Test2" by Test2, Test2
3. "Test3" by Test3, Test3
4. "Test4" by Test4, Test4
5. "Test5" by Test5, Test5

Choose the criteria to find the book:
1. by Name
2. by Author
3. by Year
4. Exit
Enter your choice here:
```

Figure 22.

The same process goes for author and year, except when we enter year it asks for the exact year that book was published in. You can see it on Figure 23 and 24.

```
Please, enter the author (min 3 characters): test

Here's the results:
1. "Test2" by Test2, Test2
2. "Test3" by Test3, Test3
3. "Test4" by Test4, Test4
4. "Test5" by Test5, Test5
```

```
Please, enter the year: 1836

Here's the results:
1. "The Posthumous Papers of the Pickwick Club" by Charles Dickens, 1836
```

Figure 23 and 24.

If the entered combination of characters could not been found in the database we will see ‘No results found’ message, as it shown in Figure 25.

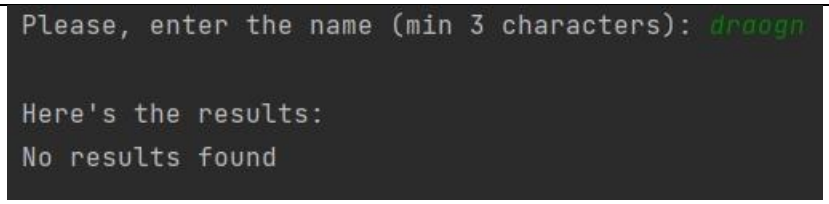



Figure 25.

In Figure 26, after we done with the ‘Search Book’ in its menu we can choose ‘Exit’ option which send us to main menu.

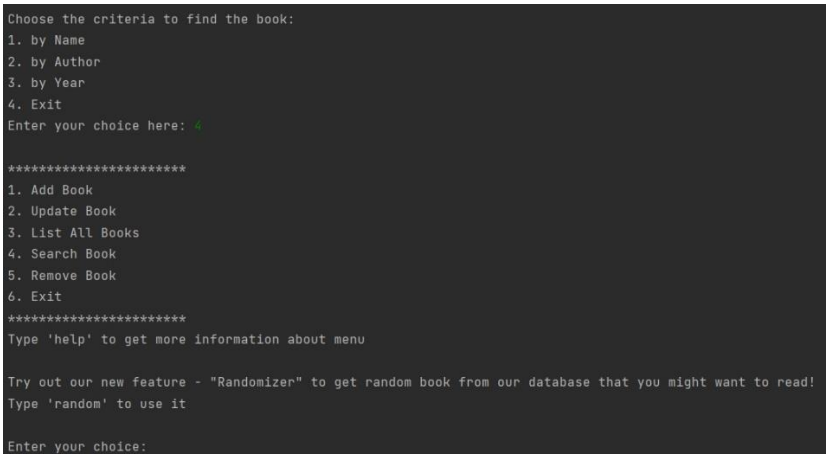



Figure 26.

4.5. Remove Book

The last numerated option is ‘Remove Book’. After we enter ‘5’ we will go to the menu of that function where we need to enter full book’s information, as it shown in Figure 27.

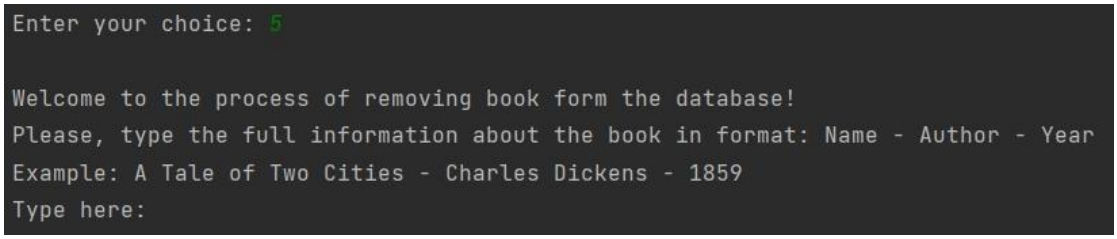



Figure 27.

In Figure 28, the wrong provided full information wil lead us to the ‘The book has not been found’ message and confirmation if we want to continue the process of removing the book with two option ‘YES’ or ‘NO’.

```
Type here: test

The book has not been found

Would you like to delete another book? ("YES" or "NO")
Enter your choice here:
```

Figure 28.

If choice is not ‘YES’ or ‘NO’ it will shown error message, as it shown in Figure 29.

```
Would you like to delete another book? ("YES" or "NO")
Enter your choice here: s

!WRONG INPUT!

Would you like to delete another book? ("YES" or "NO")
Enter your choice here:
```

Figure 29.

With option ‘YES’ it will send us back to the point where we need to enter full information about the book. We can see it on Figure 30.

```
Would you like to delete another book? ("YES" or "NO")
Enter your choice here: yes

Please, type the full information about the book in format: Name - Author - Year
Example: A Tale of Two Cities - Charles Dickens - 1859
Type here:
```

Figure 30.

Option ‘NO’ send us back to the main menu of the program, as it shown in Figure 31.

```
Would you like to delete another book? ("YES" or "NO")
Enter your choice here: no

**********
1. Add Book
2. Update Book
3. List All Books
4. Search Book
5. Remove Book
6. Exit
**********
Type 'help' to get more information about menu

Try out our new feature - "Randomizer" to get random book from our database that you might want to read!
Type 'random' to use it

Enter your choice:
```

Figure 31.

In Figure 32, if we enter full information in the right way it will delete that book from the database.

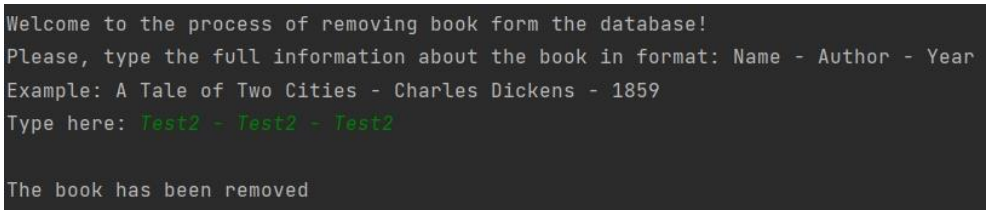



Figure 32.

4.6. Additional features: ‘Help’ and ‘Randomizer’

In Figure 33, shown the whole process of additional features of ‘Help’ option that we can see in main menu.

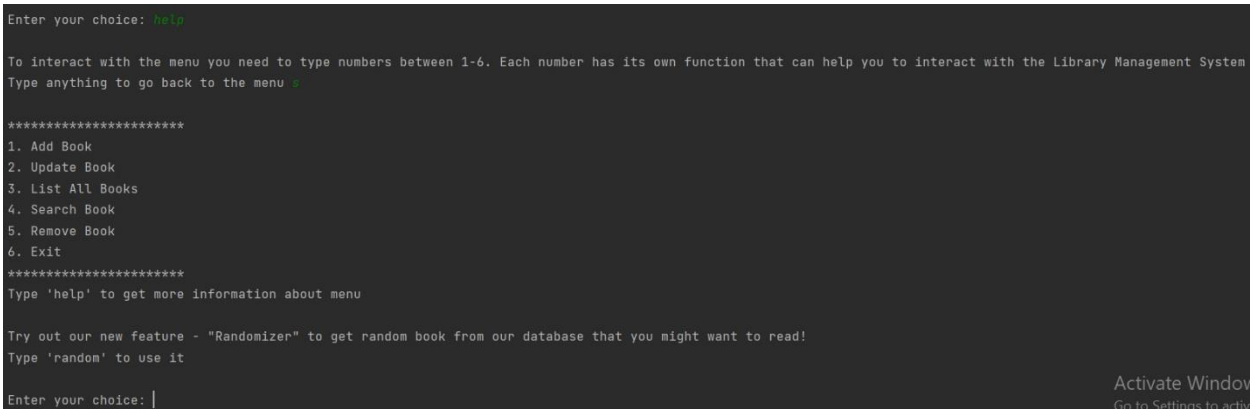



Figure 33.

In Figure 34, we can see how system help us to choose random book that exists in the database.

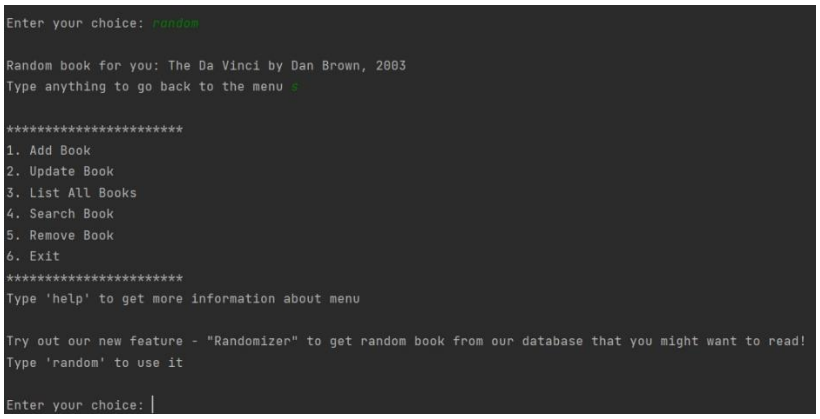



Figure 34.

# 5. CONCLUSION

In conclusion, this system fulfils the given conditions and allows us to perform various actions related to the database, where we can find books that have been added by the user. The additional functions are a great addition and similar functions can be added more. Library Management System is easily understood by the user and does not require any additional knowledge for its use.

Many tests were carried out personally by the creator of the programme, various screenshots and Flowchart were shown, which only shows the simplicity and efficiency of the system.

The system has potential for improvement, ranging from design to functionality that could be changed or added in the future. Design is the main consideration for a prime candidate for improvement, for all functionality is used within the Python programming environment.