

Table of Contents

1. Introduction	1
1.1 Goals and Objectives.....	1
2. Discussion and Analysis.....	3
2.1 Algorithm	3
2.2 Flowchart	9
2.3 Pseudocode.....	20
2.4 Data Structures.....	34
3. Program	49
3.1 Implementation of Program	49
3.2 Borrow Process	51
3.3 Returned Process	60
3.4 Adding process	69
3.4 Termination Process.....	76
4. Testing	77
4.1. Test 1: Showing implementation of try and except	77
4.2. Test 2: For selecting borrow and return option	80
4.3. Test 3: For file generation of borrow	82
4.4. Test 4: For file generation of return.....	86
4.5. Test 5: For showing the update in stock	90
5. Conclusion	97
Appendix	99
References.....	115

List of Figures

Figure 1 : Screenshot of flowchart for main menu.....	9
Figure 2 : Screenshot of flowchart for borrowing part 1.....	10
Figure 3 : Screenshot of flowchart for borrowing part 2.....	11
Figure 4 : Screenshot of flowchart for borrowing part 3.....	12
Figure 5 : Screenshot of flowchart for borrowing part 4.....	13
Figure 6 : Screenshot of flowchart for returning part1.	14
Figure 7 : Screenshot of flowchart for returning part 2.	15
Figure 8 : Screenshot of flowchart for returning part 3.	16
Figure 9 : Screenshot of flowchart for adding part 1.....	17
Figure 10 : Screenshot of flowchart for adding part 2.....	18
Figure 11 : Screenshot of flowchart for adding part 3.....	19
Figure 12 : Screenshot of implementation of integer data structure.	35
Figure 13 : Screenshot of asking integer value from user.	35
Figure 14 : Screenshot of asking float value from user.	36
Figure 15 : Screenshot of implementation of boolean data structure.	37
Figure 16 : Screenshot of implementation of print function with strings.	38
Figure 17 : Screenshot of data type conversion float to string.....	38
Figure 18 : Screenshot of implementation of split and replace function in program.	39
Figure 19 : Screenshot of defining empty list and adding value to it.	40
Figure 20 : Screenshot of list elements access by using indexing.	41
Figure 21 : Screenshot of creating empty dictionary and adding key-value pair to it....	44
Figure 22 : Screenshot of iterating and accessing all key and value of dictionary.....	45
Figure 23 : Screenshot of main user interface of library management system.....	49
Figure 24 : Screenshot of an error message when negative, string, empty, non-existed value are provided as input.	50
Figure 25 : Screenshot of user interface for borrowing books.	51
Figure 26 : Screenshot of an error message when user pass empty value.....	52
Figure 27 : Screenshot of next process after user provide valid borrower's name.	52

Figure 28 : Screenshot of an error message when user pass empty, string, negative, non-existed value to Book Id.....	53
Figure 29 : Screenshot of all available book list where Book ID "6" has book quantity zero.....	54
Figure 30 : Screenshot of an error message when user try to borrow book having quantity zero.....	54
Figure 31 : Screenshot of borrowed book details after valid book ID was given.	55
Figure 32 : Screenshot of an error message when user pass any value beside 'y' and 'n' in upper or lower case.....	55
Figure 33 : Screenshot of next process after user provide 'y' for program query.	56
Figure 34 : Screenshot of borrowing another book by user.....	56
Figure 35 : Screenshot of an error message when user try to borrow same book twice.	57
Figure 36 : Screenshot of process after user pass 'n' value to program query where update book list and borrowed bill is displayed,	58
Figure 37 : screenshot of costumer borrow books details bill.....	59
Figure 38 : Screenshot of text file note generated with unique file name after borrowing process.	59
Figure 39 : Screenshot of costumer borrow books details in text file.	59
Figure 40 : Screenshot of user interface for returning process.....	60
Figure 41 : Screenshot of an error message when user doesn't pass any value to returner's name.	60
Figure 42 : Screenshot of next process after user provide valid name.....	61
Figure 43 : Screenshot of an error message when user pass empty, string, negative, and non-existed value to book ID.....	61
Figure 44 : Screenshot of next process after user provide valid book ID.	62
Figure 45 : Screenshot of an error message after user provide string value to total day borrowed.	62
Figure 46 : Screenshot of an error message after user provide no value to total day borrowed.	62

Figure 47 : Screenshot of an error message after user provide negative value to total day borrowed.....	63
Figure 48 : Screenshot of an error message after user provide zero value to total day borrowed.....	63
Figure 49 : Screenshot of book returned successfully after user provide valid value to total day borrowed.....	63
Figure 50 : Screenshot of an error message after user provide no value to program query to return more books.....	64
Figure 51 : Screenshot of an error message after user provide number value to program query to return more books.....	64
Figure 52 : Screenshot of an error message after user provide other string value to program query to return more books.....	64
Figure 53 : Screenshot of process after user provide 'y' to program query to return more books.....	65
Figure 54 : Screenshot of user providing valid book ID.....	65
Figure 55 : Screenshot of successfully book return by user after providing valid book ID.....	65
Figure 56 : Screenshot of an error message after user try to return same book twice..	66
Figure 57 : Screenshot of successfully return book by user.....	66
Figure 58 : Screenshot of updated book list after user provide 'n' to program query for more returned books.....	67
Figure 59 : Screenshot of costumer return books details bill.....	68
Figure 60 : Screenshot of text file with unique file name generated.....	68
Figure 61 : Screenshot of customer return books details in text file.....	68
Figure 62 : Screenshot of user interface for adding books.....	69
Figure 63 : Screenshot of an error message when user try to add book of same name in book list.....	69
Figure 64 : Screenshot of an error message when user provide no value to book name.....	70
Figure 65 : Screenshot of process after user provide valid book name.....	70

Figure 66 : Screenshot of an error message after user provide no value to author's name.....	71
Figure 67 : Screenshot of process after user provide valid author's name.....	71
Figure 68 : Screenshot of an error message when user provide no value to book quantity.....	71
Figure 69 : Screenshot of an error message when user provide string value to book quantity.....	71
Figure 70 : Screenshot of an error message when user provide negative value to book quantity.....	72
Figure 71 : Screenshot of process after user provide valid book quantity.....	72
Figure 72 : Screenshot of an error message when user provide no value to book price.....	72
Figure 73 : Screenshot of an error message when user provide string value to book price.....	73
Figure 74 : Screenshot of an error message when user provide negative value to book price.....	73
Figure 75 : Screenshot of successfully adding of book after user provide valid price. ..	73
Figure 76 : Screenshot of an error message when user provide no value to program query to add more book.....	74
Figure 77 : Screenshot of an error message when user provide other string value to program query to add more book.....	74
Figure 78 : Screenshot after user provide valid response to program query and adding another book successfully.....	75
Figure 79 : Screenshot after user provide 'n' to program query to add more books and update book list after adding books.....	75
Figure 80 : Screenshot of main library management system user interface.....	77
Figure 81 : Screenshot of user interface after user input value 2 and user providing returner's name.....	78
Figure 82 : Screenshot of user providing input for returner's name and book ID to return book.....	78

Figure 83 : Screenshot of an error message when user provide string value instead of integer value.....	79
Figure 84 : Screenshot of successfully returned book when user provide positive integer value.....	79
Figure 85 : Screenshot of an error message after user provide negative value as input.	80
Figure 86 : Screenshot of an error message after user provide other string value beside 1,2,3,4 value.....	81
Figure 87 : Screenshot of an error message after user provide non-existed value.	81
Figure 88 : Screenshot of user interface for borrowing book after user provide value 1.	81
Figure 89 : Screenshot of user providing 1 value as input.....	82
Figure 90 : Screenshot of user providing borrower's name.....	83
Figure 91 : Screenshot of user providing book Id to borrow book.	83
Figure 92 : Screenshot of user providing 'y' value to program query to borrow more books.	84
Figure 93 : Screenshot of user providing book Id to borrow book.	84
Figure 94 : Screenshot of user providing 'n' value to program query to borrow more book.	85
Figure 95 : Screenshot of costumer borrow book details in text file.	85
Figure 96 : Screenshot of user providing " 2 " value as input. For return process	86
Figure 97 : Screenshot of user providing returner's name.....	87
Figure 98 : Screenshot of user providing book ID "7" to return book.....	87
Figure 99 : Screenshot of user providing number of days borrowed and successfully returning book.	87
Figure 100 : Screenshot of user providing 'y' value to return more books.	88
Figure 101 : Screenshot of user providing book ID to return another book.	88
Figure 102 : Screenshot of user providing number of days borrowed and successfully returning book.	88
Figure 103 : Screenshot of after user providing 'n' value and updated book list and costumer return bill.....	89
Figure 104 : Screenshot of costumer return books details in text file.	89
Figure 105 : Screenshot of bookStock_list.txt file in initial phase.	91
Figure 106 : Screenshot of user providing value "1" to library menu option.	91
Figure 107 : Screenshot of user providing borrower's name.	92
Figure 108 : Screenshot of user providing book ID "4".....	92
Figure 109 : Screenshot of borrowed book details after successfully borrowing book. .	92

Figure 110 : Screenshot of updates book stock list in table and costumer borrow details bill.....	93
Figure 111 : Screenshot of updated stock list in bookStock_list.txt file.	93
Figure 112 : Screenshot of user providing value "2" for returning process.....	94
Figure 113 : Screenshot of user providing returner's name.....	94
Figure 114 : Screenshot of user providing book ID "1" to return book.....	95
Figure 115 : Screenshot of user providing number of days borrowed and successfully returning book.	95
Figure 116 : Screenshot of updates book stock list in table and costumer return details bill.....	96
Figure 117 : Screenshot of updated stock list in bookStock_list.txt file.	96

List of Tables

Table 1 : To show implementation of try and except	77
Table 2 : To demonstrate selection of borrow and return option.	80
Table 3 : To show file generation of borrow.	82
Table 4 : To show file generation of return.	86
Table 5 : To show the update in stock.....	90

1. Introduction

This coursework is completed by using python idle, draw.io and msWord.

The main task of this coursework is to develop a library management system application where tasks like borrowing, returning, adding of book and generating bill for each transaction of borrowing and returning book are done.

Here, library management system application is for library shop that maintains its all information of book (stock) in a text file and read the information on text file to display all the available book's information in table. This application provides a simple user interface that provides a menu option for user to choose value from 1 to 4, where 1 for borrowing book, 2 for returning book ,3 for adding books and 4 for terminating program.

For borrowing, user is asked for their name and bookID which they want to borrow. If user want to borrow multiple books but cannot borrow same book twice then all the books borrowed is written to the text file and the total cost is added up for all books borrowed. A note is generated which contains borrower name, book name, date, time and total cost to be paid for each transaction.

For returning, user is asked for their name, bookID and days of borrowed book to return books. If user want to return multiple books but cannot return same book twice then all the books return is written to the text file and total fine is added up for all return books if days borrowed exceeds 10 days. A note is generated which contains returner name, book name, date, time, day borrowed and total fine to be paid for each transaction.

For adding, user is asked for the book name, author name, quantity and price. Books are added only if user provided book name is not in the text file of book list. If user want to add multiple books, then all the books' details are added directly in to text file to update the book list.

1.1 Goals and Objectives

- To make an application that maintains its book information in text file and display all books available for borrowing by reading that text file.

- To make application more user friendly and improving in handling of exception and errors.
- Generating bill as text file with unique names for each transaction for borrowing and returning books.
- To handle managing operation of library tasks more efficiently and properly..
- To fine to any person if any books are borrowed more than 10 days.
- To have knowledge of different data structures and modular approach of programming.
- To display book details information like ID, book name, author name, quantity and price.

2. Discussion and Analysis

2.1 Algorithm

Algorithm is step wise procedure to solve any problem based on carrying out sequence of specified action. (Contributor, 1999 - 2021)

Step 1: START

Step 2: READ bookStock.txt file

Step 3: PRINT "Welcome to College Library Management System."

Step 4: DISPLAY list of available books in a table.

Step 5: Display library menu option to provide different choices to user.

PRINT "Please, Enter '1' to borrow a book.

PRINT "Please, Enter '2' to return"

PRINT "Please, Enter '3' to add a new book."

PRINT "Please, Enter '4' to exit."

PRINT "Please, Enter a value as 1 or 2 or 3 or 4:"

Step 6: Ask user input value.

If user input == 1, PRINT "You will now borrow the book:" and go to step 7.

Else if user input == 2, PRINT "You will now return the book:" and go to step 22.

Else if user input == 3, PRINT "You will now add new books to book stock List:"

and DISPLAY list of available books in a table then go to step 36.

Else if user input == 4, go to step 45.

Else PRINT "Invalid input!! Please, provide input value as 1, 2,3 or 4."

and go to step 5.

Step 7: DISPLAY list of available books in a table.

Step 8: INITIALIZE borrowedBook_ID_List as empty list

Step 9: Ask user to input borrower's name.

If borrower's name == empty then PRINT "Name cannot be left empty!!! Please, Enter the borrower's name." and go to step 9.

Else move to next step.

Step 10: INITIALIZE date_Borrow = only date import from datetime module

INITIALIZE time_Borrow = only time import from datetime module.

INITIALIZE bookID_List = all key list of dictionary

Step 11: Ask user to input Book ID.

If user input book ID is in the book ID list, then move to next step

Else PRINT "Invalid book Id!!! Please, provide valid Book ID from above book table."

Step 12: if user input book ID is not in borrowedBook_ID_List , then move to next step

Else PRINT "This book is already borrowed by you.

Please, select another book from the stock." and go to step 10.

Step 13: if quantity of book > 0 then move to next step.

Else PRINT "Book is not available for now. Please, select another book from the stock." and go to step 10.

Step 14: Add user Input book ID in to borrowedBook_ID_List .

i.e., borrowedBook_ID_List.append(userInput_BookID)

DISPLAY book name, book price, date and time in table

Step 15: Decrease the quantity of borrowed book by 1.

i.e., update_Quantity = Quantity of borrowed book -1 then,

WRITE the book details with update quantity for borrowed book in bookStock.txt file

Step 16: Ask user to input 'y' for yes and 'N' for No to borrow more books

If user input == y then go to step 11.

Else if user input == n then go to step 17.

Else PRINT "Invalid Input!!! Please, Provide 'Y' for YES and 'N' for NO to borrow more books. " and go to step 16.

Step 17: PRINT "Updated books list in library after borrowing books : "

DISPLAY updated book list after borrowing books.

Step 18: CALCULATE total cost for all borrowed books.

i.e. totalCost = 0

If key in borrowedBook_ID_List
(repeat until key is present in borrowedBook_ID_List)
 totalCost = totalCost + price of book

Step 19: IMPORT datetime module for date and time .

 minute_Value = minute value from datetime module
 microsec_Value = micro second value from datetime module
 fileName = customerName + minute_Value + microsec_Value

Step 20: GENERATE unique_fileName.txt file to make a borrow bill

 WRITE customer name,book name,date and time and totalcost
 to unique_fileName.txt file

Step 21: READ unique_fileName.txt file

 Display customer name, book name, date, time and total cost.
 i.e., Customer Borrow Bill
 PRINT "Thank you for using our library management system."
 and go to step 5.

Step 22: DISPLAY list of available books in a table.

Step 23: INITIALIZE returnBooks_Id, days as empty list

Step 24: Ask user to input returner's name.

 If returner's name == empty then PRINT "Name cannot be left empty!!! Please,
 Enter the returner's name." and go to step 24.
 Else move to next step.

Step 25: INITIALIZE bookID_List = all key list of dictionary

Step 26: Ask user to input Book ID.

 If user input book ID is in the book ID list, then move to next step
 Else PRINT "Invalid book Id!!! Please, provide valid Book ID from
 above book table."

Step 27: if user input book ID is not in returnBooks_Id, then move to next step

 Else PRINT "This book is already returned by you.
 Please, select another book from the stock." and go to step 26.

Step 28: Add user Input book ID in to returnBooks_Id.

i.e., returnBooks_Id.append(userInput_BookID)

PRINT " This book has been successfully returned."

Step 29: Increase the quantity of returned book by 1.

i.e., update_Quantity = Quantity of returned book + 1 then,

WRITE the book details with update quantity for returned book
in bookStock.txt file

Step 30: Ask user to input 'y' for yes and 'N' for No to borrow more books

If user input == y then go to step 26.

Else if user input == n then go to step 31 .

Else PRINT "Invalid Input!!! Please, Provide 'Y' for YES and 'N' for NO to
return more books. " and go to step 30.

Step 31: PRINT "Updated books list in library after returning books : "

DISPLAY updated book list after returning books.

Step 32: CALCULATE total fine for all returned books and move to next step.

i.e. totalFine = 0

noCharge_duration = 10

i = 0

If key in returnBooks_Id

(repeat until key is present in returnBooks_Id)

rate = price of a book

borrow_days = days [i]

If borrow_days > noCharge_duration then

extra_Days = borrow_days - noCharge_duration

fine = 0.05 * rate * extra_Days

totalFine += fine

Else fine = 0

totalFine += fine

Value of i increment by 1 every time loop excuted

totalFine_Amount = "\$" + str(round(totalFine,2))

Returns value of totalFine_Amount

Step 33: IMPORT datetime module for date and time.

INITIALIZE date_Return = only date import from datetime module

INITIALIZE time_Return = only time import from datetime module.

minute_Value = minute valuee from datetime module

microsec_Value = micro second value form datetime module

fileName = costumerName + minute_Value + microsec_Value

Step 34: GENERATE unique_fileName.txt file to make a borrow bill

WRITE customer name, book name, date and time,

total day borrowed and total fine

to unique_fileName.txt file

Step 35: READ unique_fileName.txt file

Display costumer name, book name, date, time,

total day borrowed and total fine.

i.e., Costumer return Bill

PIRNT "Thank you for using our library management system."

and go to step 5.

Step 36: INITIALIZE bookName = list of books name

Step 37: Ask user to input book name.

If book name == empty then PRINT "Book name cannot be left empty,"

Please provide the book name." and go to step 37

ELSE move to next step.

Step 38: If user book name not in bookName list then move to next step

ELSE PRINT " This book is already in book Stock list. Please,

provide another Book Name to add book." And go to step 37.

Step 39: Ask user to input book's Author name.

If author's name == empty then PRINT "Book author's name

cannot be left empty, please provide the book author's name."

and go to step 39.

ELSE move to next step.

Step 40: Ask user to input book quantity number.

If book quantity number ≥ 0 then move to next step.

Else PRINT "Invalid input!!! Please provide valid
quantity and of positive number..", then go to step 40

Step 41: Ask user to input book price.

If book price > 0 , then move to next step.

Else PRINT "Invalid input!!! Please provide
valid price and of positive number.", then go to step 41

Step 42: open bookStock.txt file in APPEND mode(to add)

WRITE book name,book author,quantity,price in bookStock.txt file

PRINT This book is successfully added to book list." and move to next step.

Step 43: Ask user to input 'Y' for yes and 'N' for No to add more books.

If user input == " y" then, repeat process of step 37,38,39,40,41 and 42.

Else if user input equal to "n" then go to step 44.

Else PRINT "Invalid Input!!! Please, provide 'Y' for YES and 'N' for NO
to add more books." then go to step 43.

Step 44: PRINT "Updated books list in library after adding books : "

DISPLAY updated book list table after adding new books

Then go to step 5.

Step 45: PRINT "Thank you for using our library management system."

Then BREAK (the exit both loop and the program).

Step 46: STOP

2.2 Flowchart

A flowchart is a graphical depiction of a series of steps. It is commonly present the flow of algorithms or processes. It shows steps in a sequential manner. A flowchart often depicts the processes as various types of boxes, with arrows linking them in the proper order. (Visual Paradigm, 2021)

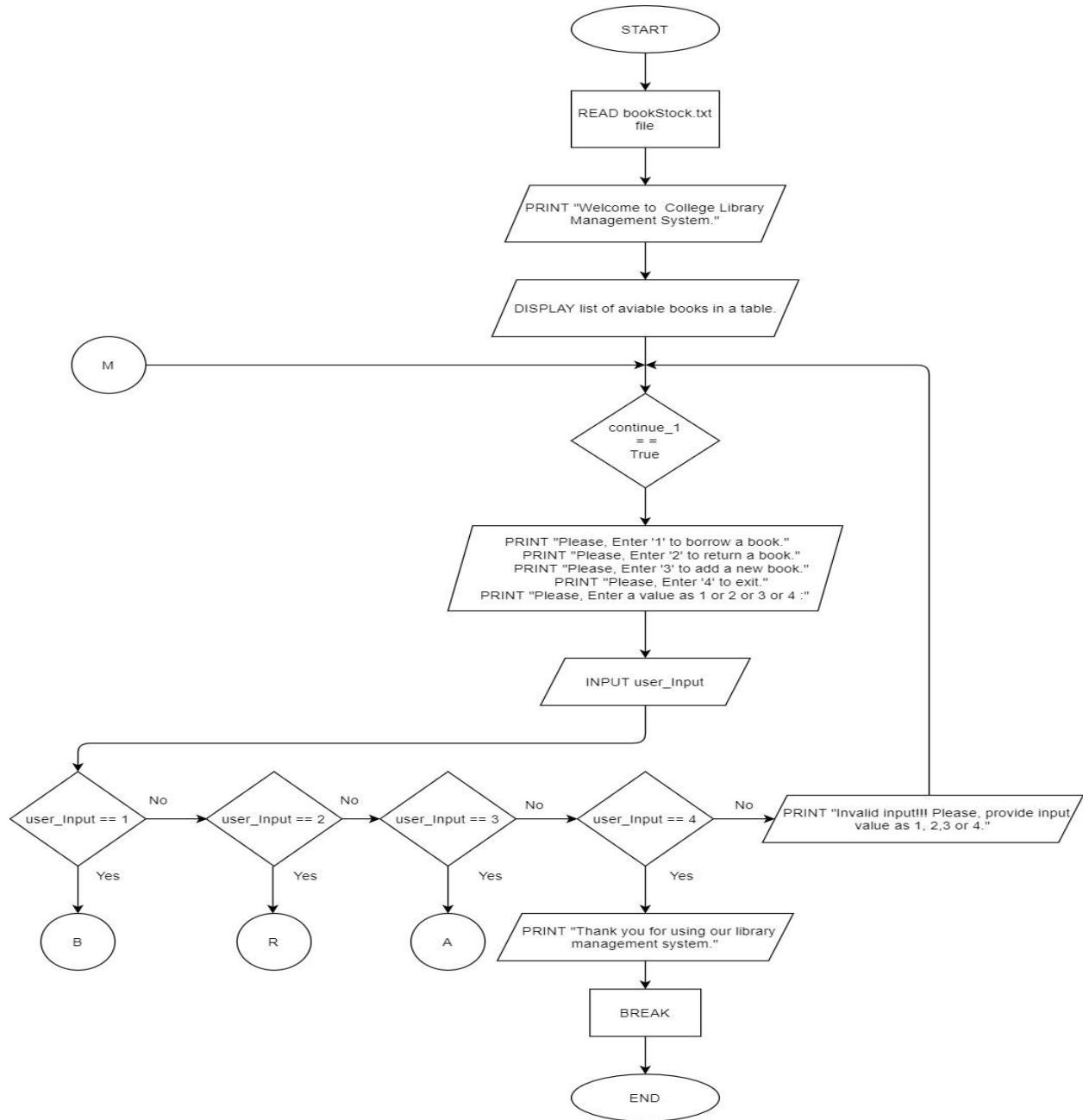


Figure 1 : Screenshot of flowchart for main menu.

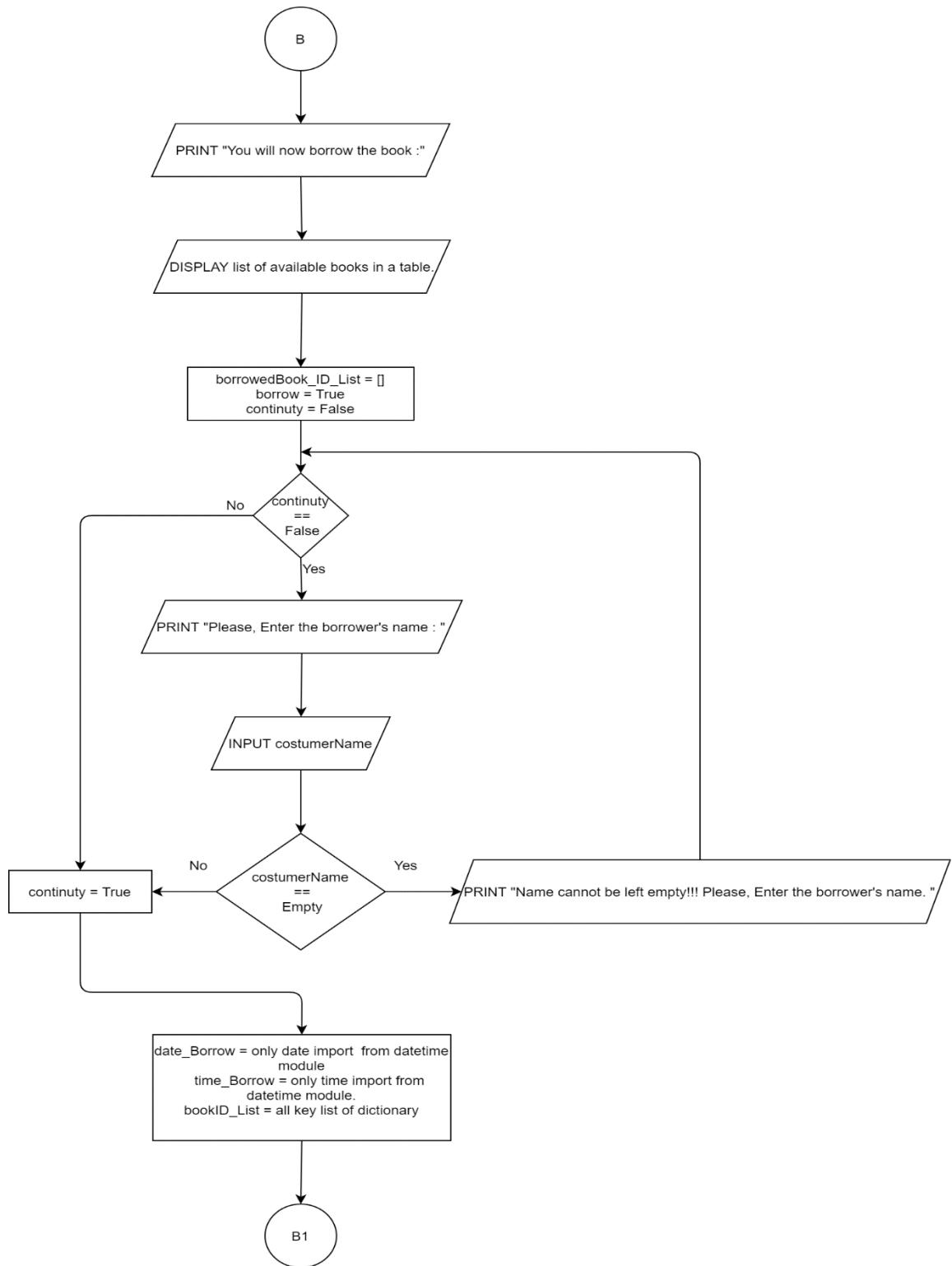


Figure 2 : Screenshot of flowchart for borrowing part 1.

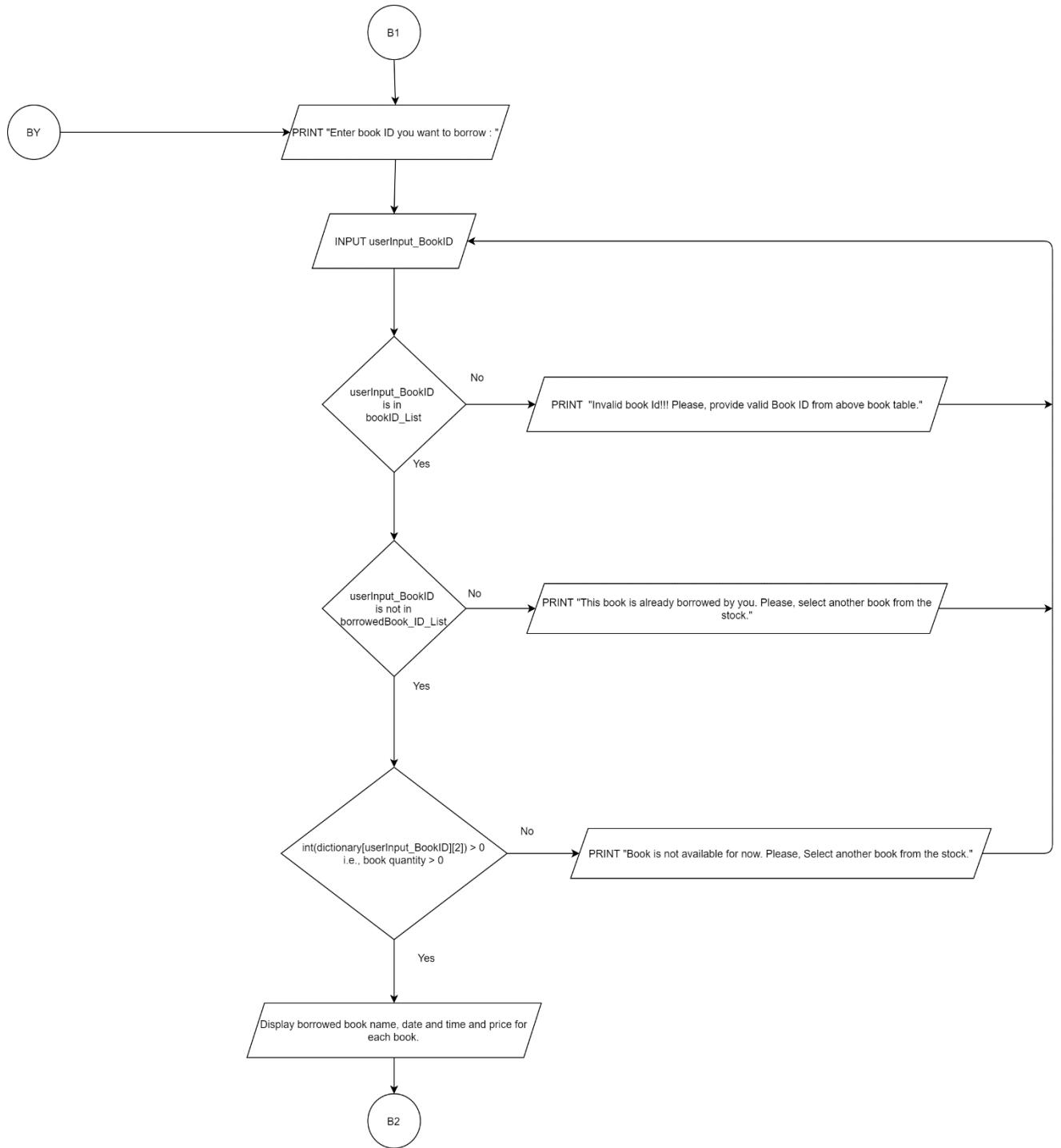


Figure 3 : Screenshot of flowchart for borrowing part 2.

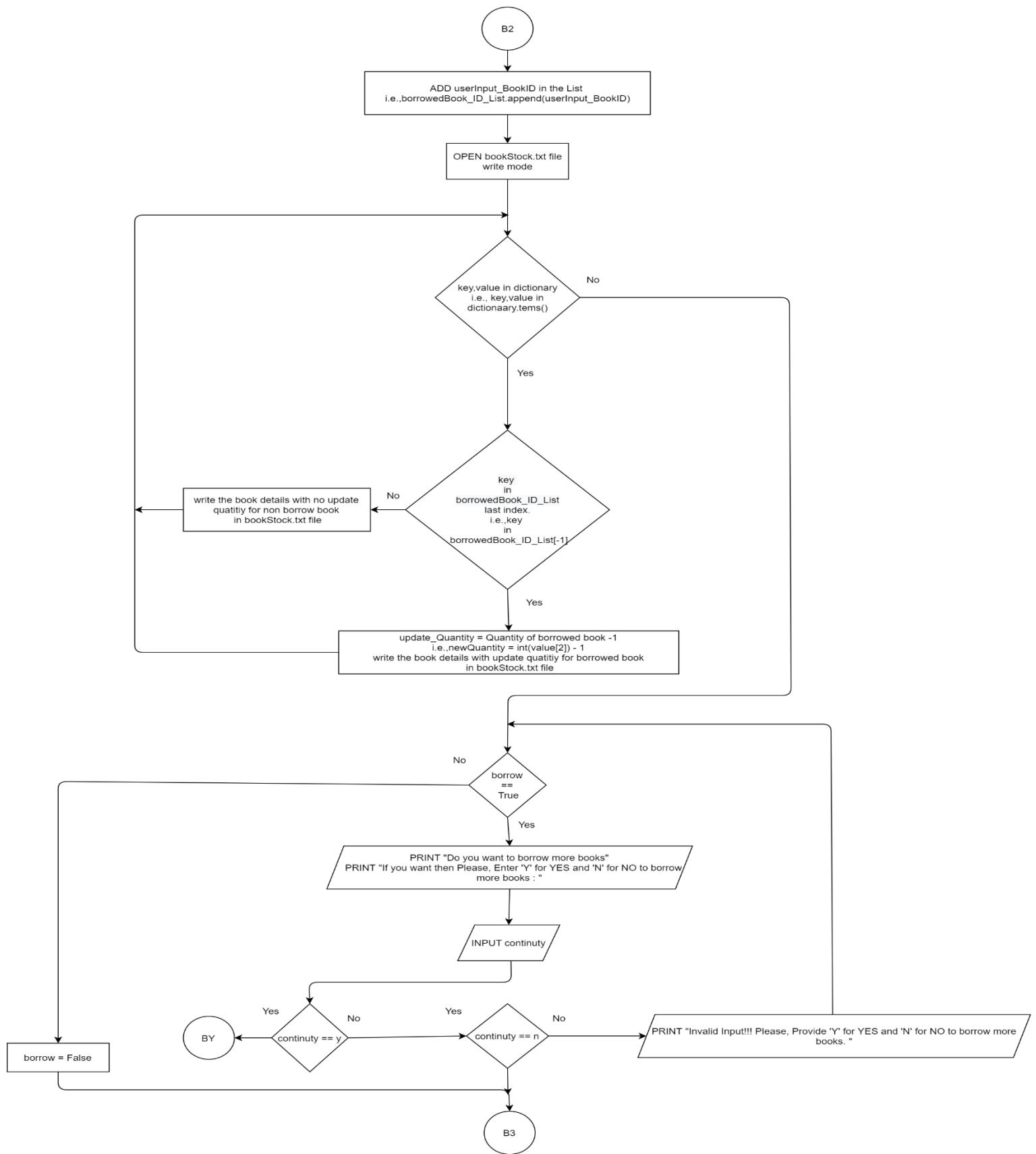


Figure 4 : Screenshot of flowchart for borrowing part 3.

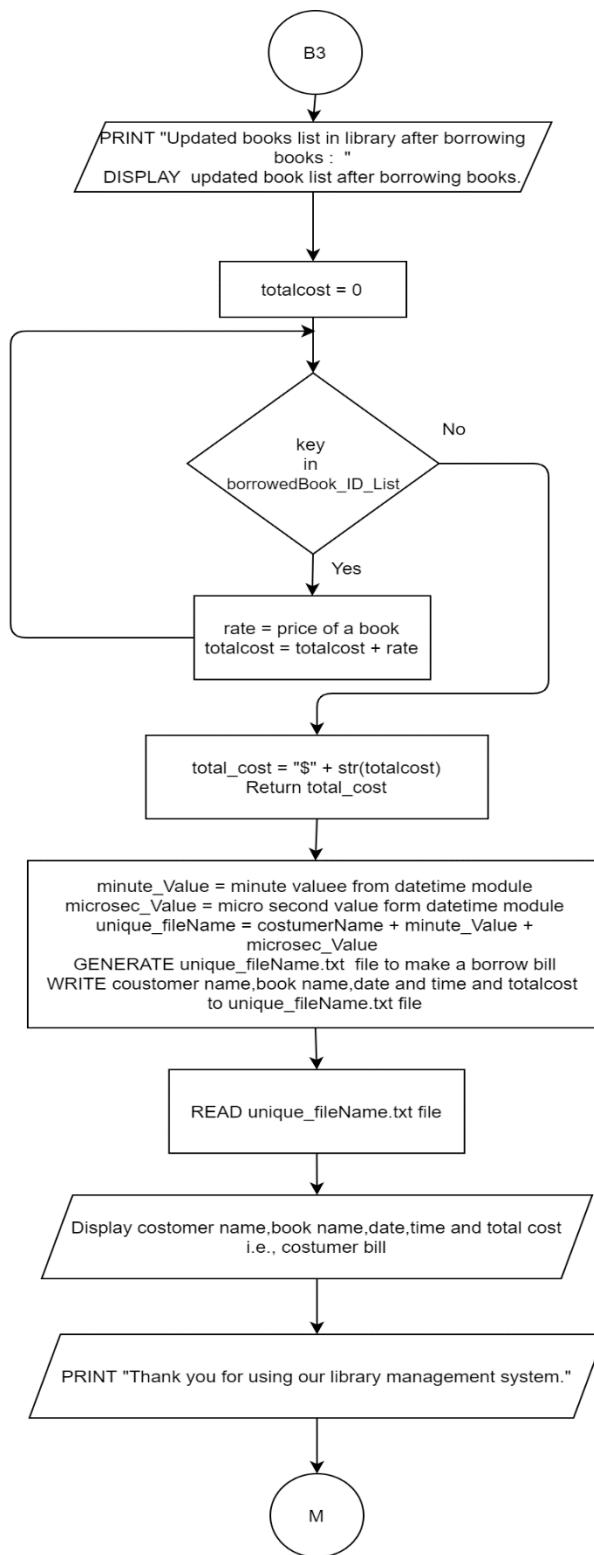


Figure 5 : Screenshot of flowchart for borrowing part 4.

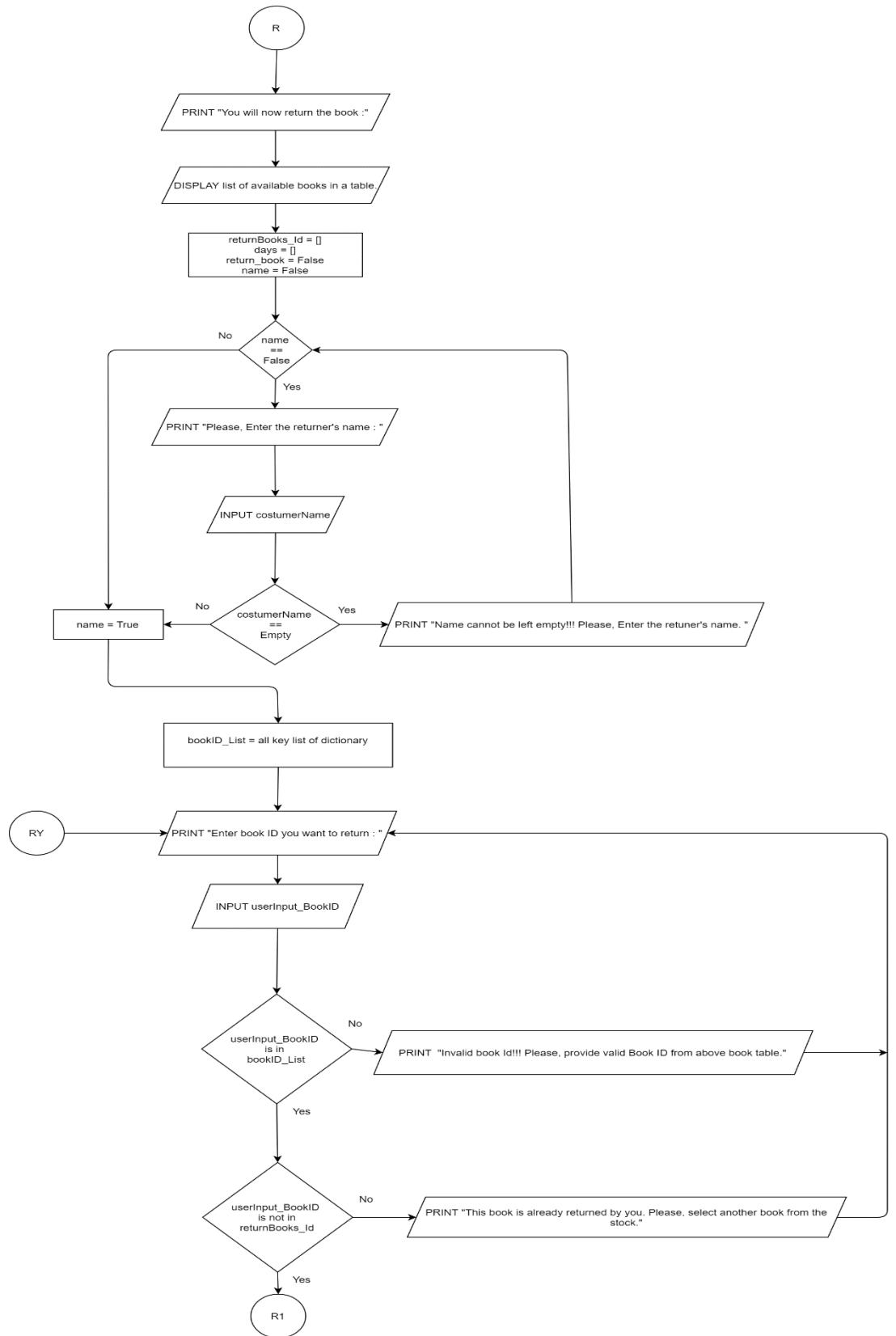


Figure 6 : Screenshot of flowchart for returning part1.

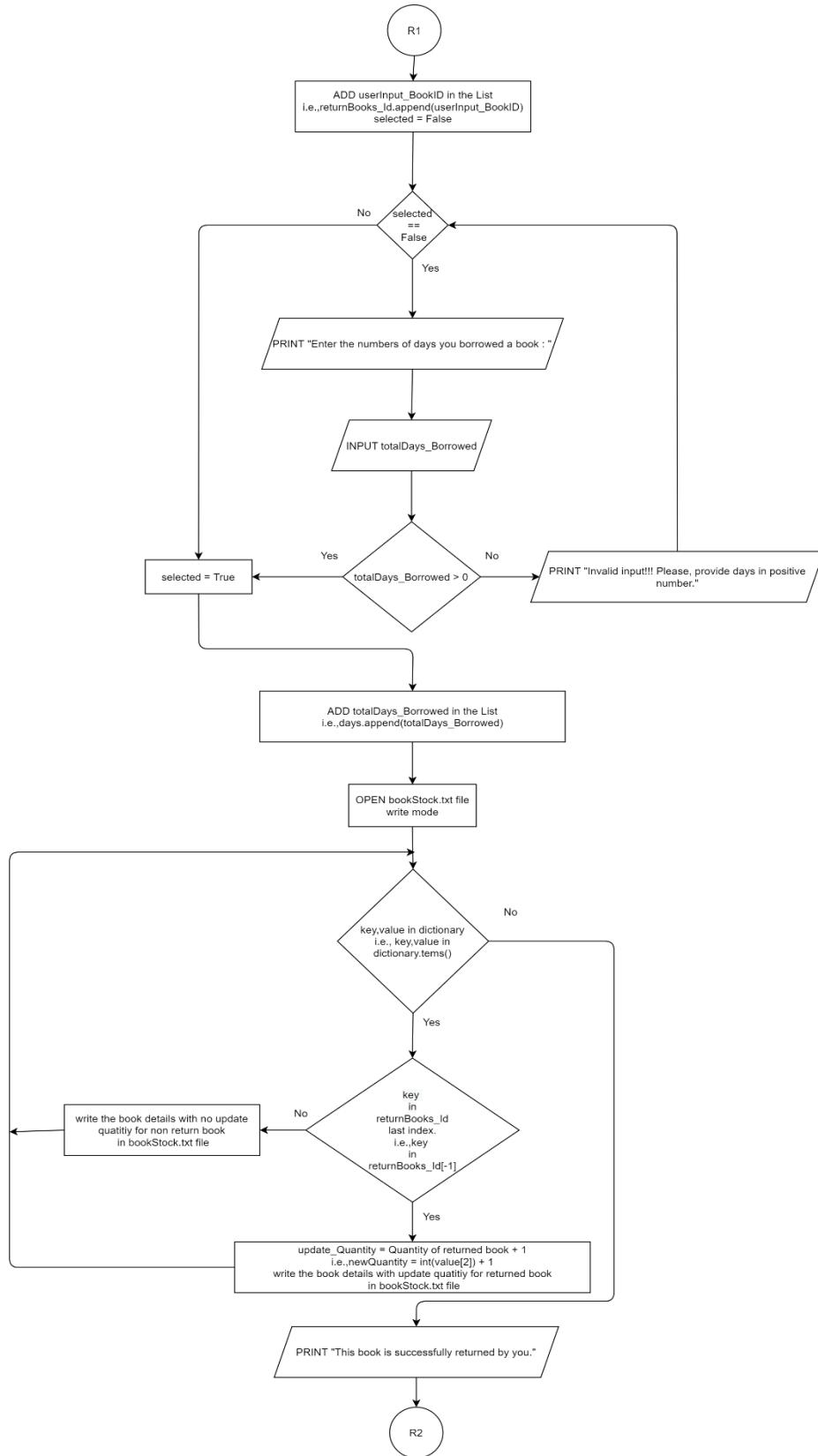


Figure 7 : Screenshot of flowchart for returning part 2.

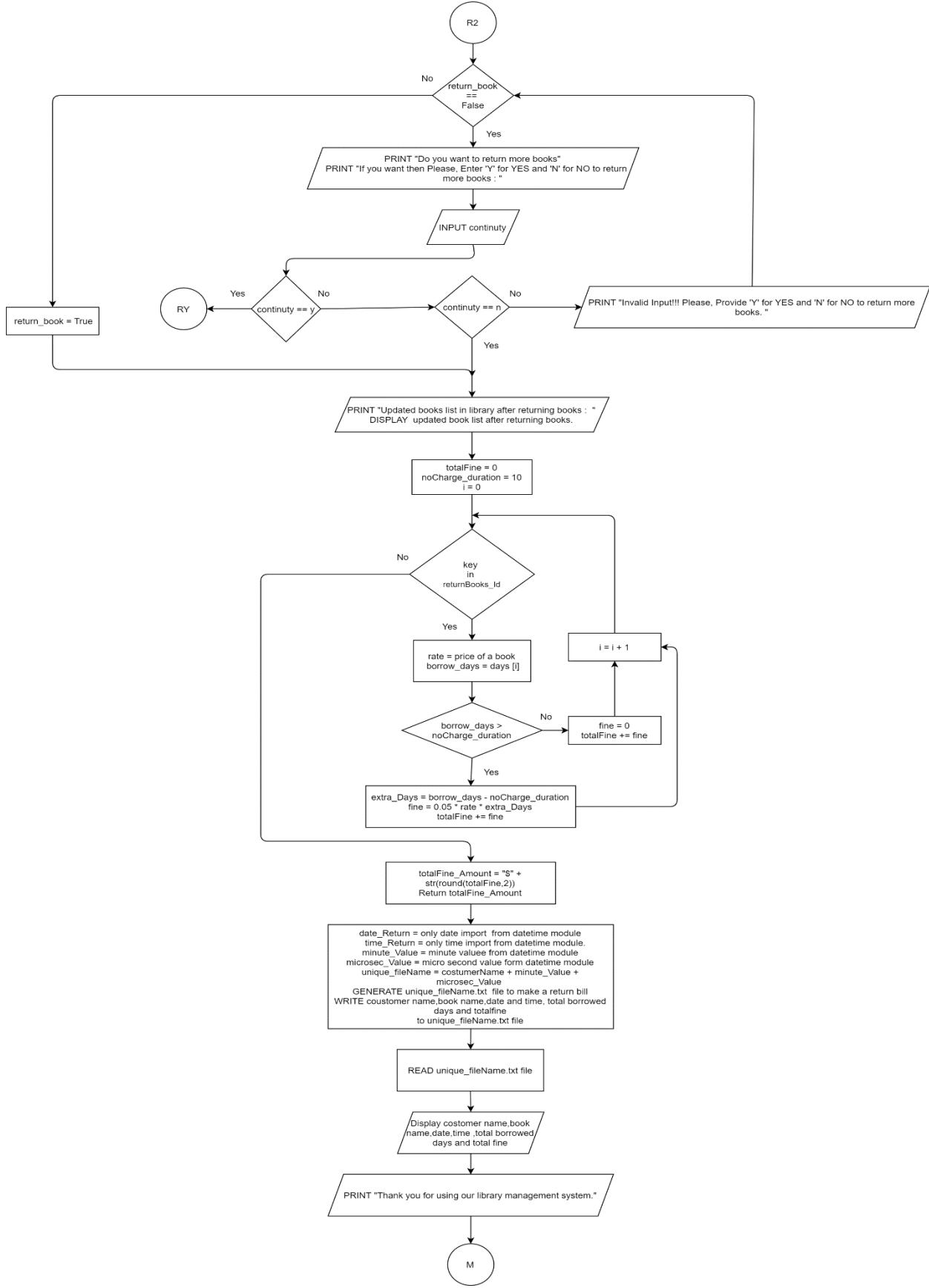


Figure 8 : Screenshot of flowchart for returning part 3.

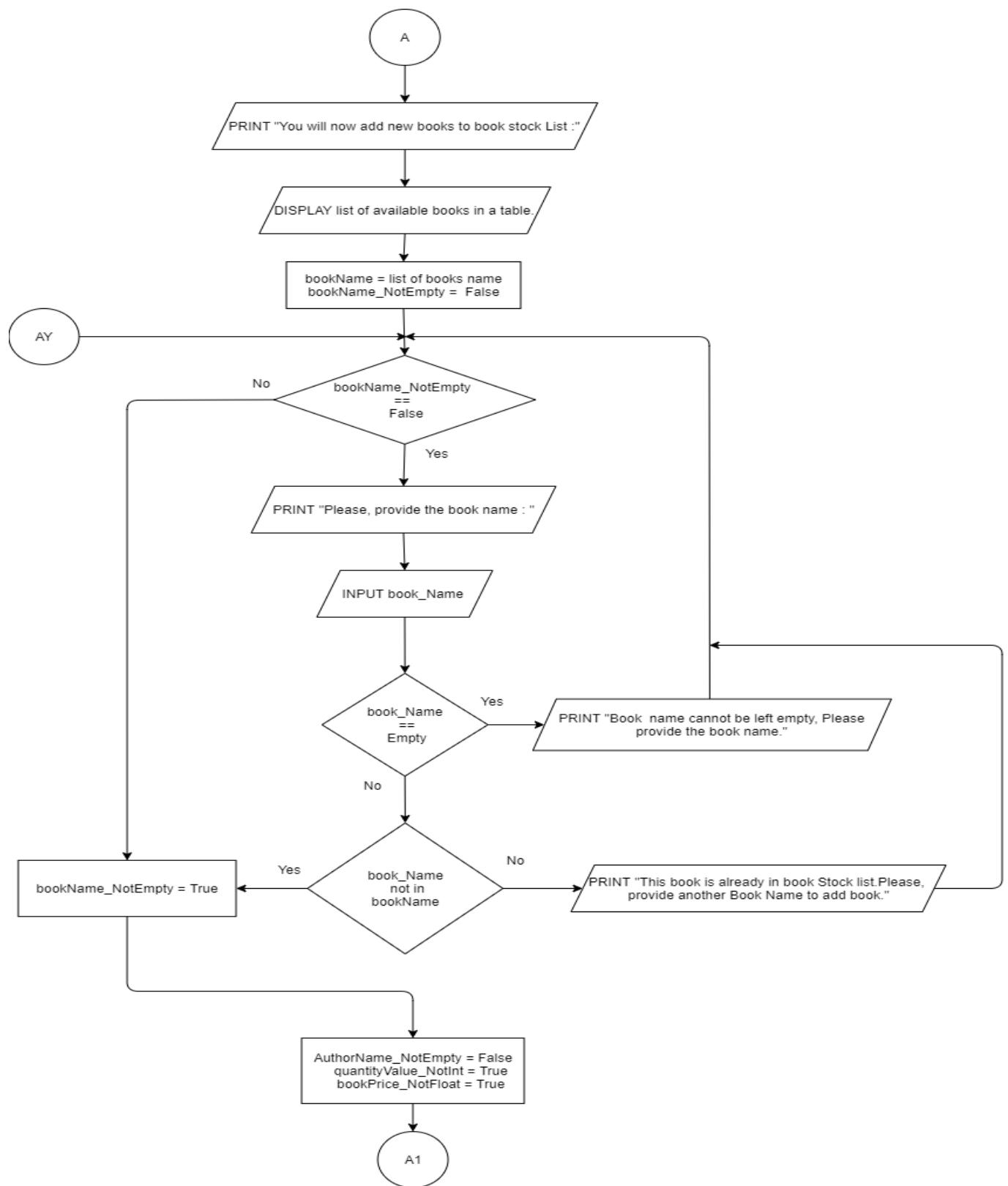


Figure 9 : Screenshot of flowchart for adding part 1.

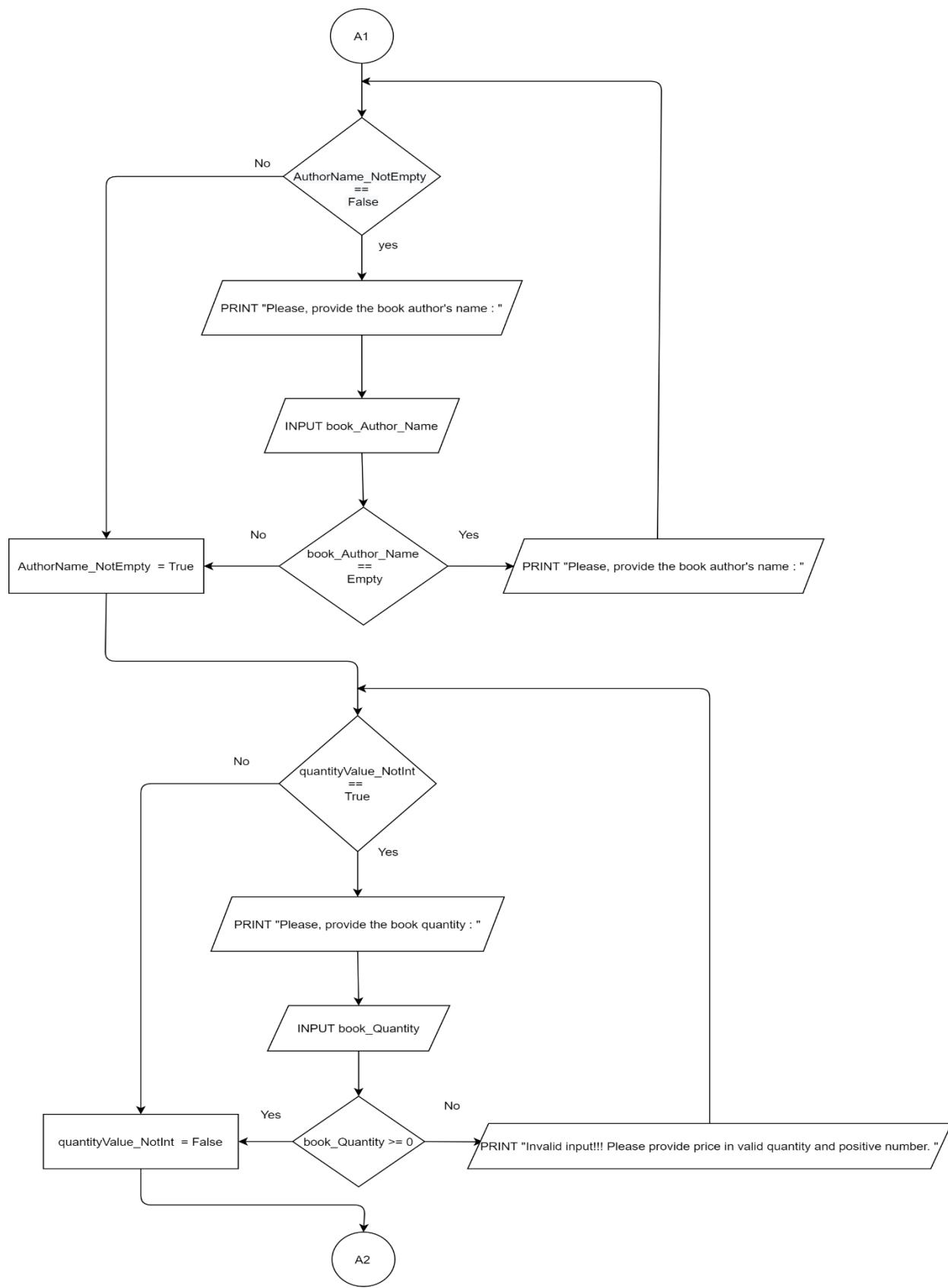


Figure 10 : Screenshot of flowchart for adding part 2.

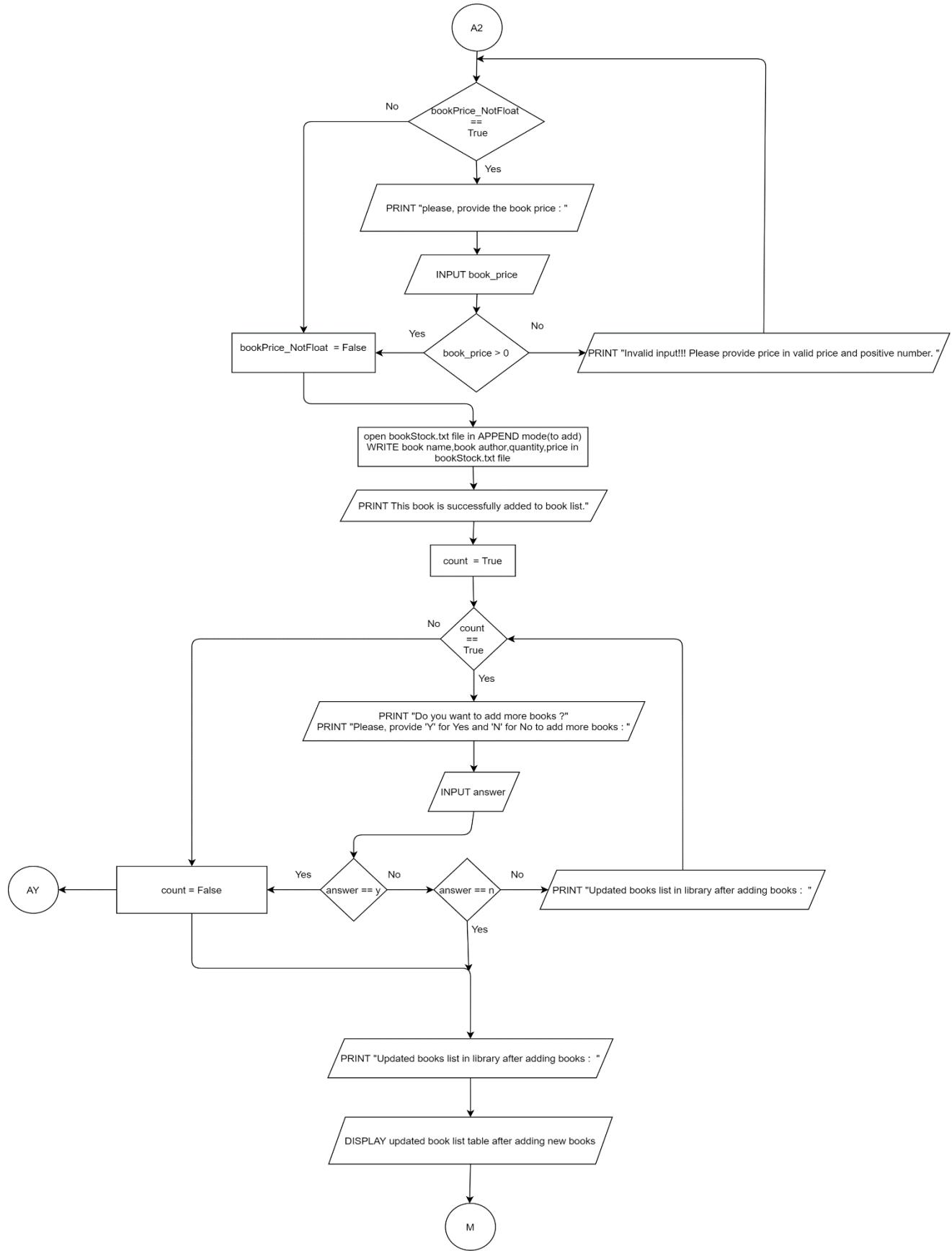


Figure 11 : Screenshot of flowchart for adding part 3.

2.3 Pseudocode

Pseudocode is informal way of method for describing programming which doesn't follow any programming language syntax or underlying technology considerations. It is used to create a program outline or a rough draft. (Coleman & Co. Ltd., 2021)

a. main.py

```
IMPORT function_List module  
CALL main_Function() function of function_List module
```

b. function_List.py

```
IMPORT sys module  
IMPORT datetime module  
DECLARE FUNCTION main_Function()  
    PRINT Welcome to College Library Management System.  
    CALL dict_display() function to display book list  
    SET continue_1 = True  
    while continue_1 == True:  
        PRINT Please, Enter '1' to borrow a book.  
        PRINT Please, Enter '2' to return a book.  
        PRINT Please, Enter '3' to add a new book.  
        PRINT Please, Enter '4' to exit.  
        INPUT user_Input  
        IF user_Input == str(1) THEN  
            PRINT You will now borrow the book  
            CALL borrow_Function() function  
        ELSE IF user_Input == str(2) THEN  
            PRINT You will now return the book  
            CALL return_Function() function  
        ELSE IF user_Input == str(3) THEN  
            PRINT You will now add new books to book stock List  
            CALL add_Books() function
```

```

ELSE IF user_Input == str(4) THEN
    PRINT Thank you for using our library management system
    BREAK
ELSE
    PRINT Invalid input!!! Please, provide input value as 1, 2,3 or 4
END IF
END WHILE

```

```

DECLARE FUNCTION dict_Function()
    OPEN bookStock_List.txt in read mode
    SET dictionary = {}
    SET book_ID = 0
    FOR each line in bookStock_List.txt
        INCREMENT book_ID by 1
        line = replace “\n” from line with empty space and split line form ","
        ADD book_ID as Key string type and line as value in empty dictionary
    END FOR
    CLOSE bookStock_List.txt
    RETURN dictionary

```

```

DECLARE FUNCTION allKey_List()
    SET key_List = []
    SET dictionary = dict_Function()
    FOR each key in dictionary
        ADD key to key_List
    END FOR
    RETURN key_List

```

DECLARE FUNCTION list_BookName()

```

SET bookName_List = []
SET dictionary = dict_Function()
FOR each value in dictionary
    ADD value[0].lower() to bookName_List
END FOR
RETURN bookName_List

```

```

DECLARE FUNCTION dict_display()
PRINT "Book ID"+ "Book Name"+ "Author"+ "Quantity"+ "Price"
SET display_dictioanry = dict_Function()
FOR each key,value in display_dictioanry
    PRINT key + value[0] + value[1] + value[2] + value[3]
END FOR

```

```

DECLARE FUNCTION borrow_Function()
SET costumerName,borrowedBook_ID_List as global
CALL dict_display() function
SET borrowedBook_ID_List = []
SET borrow = True
SET continuity = False
WHILE continuity == False:
    INPUT costumerName
    IF costumerName == "" THEN
        PRINT Name cannot be left empty!!! Please,
        Enter the borrower's name
    ELSE
        SET continuity = True
        CALL validation_BorrowedbookID() function
        WHILE borrow == True:

```

```

PRINT Do you want to borrow more books
INPUT continuity
IF continuity == "y" THEN
    CALL validation_BorrowedbookID() function
ELSE IF continuity == "n" THEN
    SET borrow = False
    PRINT Updated books list in library
        after borrowing books
    CALL dict_display() function
    CALL borrowDetails_Write_ReadOp() function
    PRINT Thank you for using our
        library management system
ELSE
    PRINT Invalid Input!!! Please, Provide 'Y' for YES
        and 'N' for NO to borrow more books
END IF
END WHILE
END IF
END WHILE

```

```

DECLARE FUNCTION validation_BorrowedbookID()
    SET date_Borrow = date only from datetime module
    SET time_Borrow = time only from date time module
    SET bookID_List = allKey_List()
    SET dictionary = dict_Function()
INPUT userInput_BookID
IF userInput_BookID in bookID_List THEN
    IF userInput_BookID not in borrowedBook_ID_List THEN
        IF(int(dictionary[userInput_BookID][2]) > 0) THEN
            PRINT This Book is available for borrow

```

```

PRINT "Book Name\|" + " Price" +
    "Date of Borrow " + "Time of Borrow "
PRINT dictionary[userInput_BookID][0]
    dictionary[userInput_BookID][3] +
    date_Borrow + time_Borrow
ADD userInput_BookID to borrowedBook_ID_List
CALL updateBorrow_File()

ELSE
    PRINT Book is not available for now.
    Please, Select another book from the stock.
    CALL validation_BorrowedbookID() function
END IF

ELSE
    PRINT This book is already borrowed by you.
    Please, select another book from the stock
    CALL validation_BorrowedbookID() function
END IF

ELSE
    PRINT Invalid book Id!!! Please,
        provide valid Book ID from above book table
    CALL validation_BorrowedbookID() function
END IF

DECLARE FUNCTION updateBorrow_File()
    SET bookID_List = allKey_List()
    SET dictionary = dict_Function()
    OPEN bookStock_list.txt in write mode
    FOR each key,value in dictionary
        IF key in borrowedBook_ID_List of last index THEN

```

```

SET newQuantity = borrowed book quantity- 1
IF key == bookID_List of last index THEN
    WRITE value[0], value[1] ,newQuantity,
        value[3] with next line on bookStock_list.txt
ELSE
    WRITE value[0], value[1] ,newQuantity,
        value[3] without next line on bookStock_list.txt
END IF
ELSE
    IF key == bookID_List of last index THEN
        WRITE value[0], value[1] ,value[2],
            value[3] with next line on bookStock_list.txt
    ELSE
        WRITE value[0], value[1] ,value[2],
            value[3] without next line on bookStock_list.txt
    END IF
END IF
END FOR
CLOSE bookStock_list.txt

```

```

DECLARE FUNCTION costBorrow_book()
    SET dictionary = dict_Function()
    SET totalcost = 0
    FOR each key in borrowedBook_ID_List
        SET rate = price of borrowed book
        SET totalcost += rate
    END FOR
    SET total_cost = "$" + str(totalcost)
    RETURN total_cost

```

```

DECLARE FUNCTION borrowDetails_Write_ReadOp()
  SET dictionary = dict_Function()
  SET date_Borrow = date only from datetime module
  SET time_Borrow = time only from datetime module
  SET minute_Value = str(datetime.datetime.now().minute)
  SET microsec_Value = str(datetime.datetime.now().microsecond)
  SET unique_fileName = costumerName + minute_Value + microsec_Value
  OPEN unique_fileName + "B" + ".txt" file in write mode
  WRITE Customer Borrow Books Details
  WRITE "Costumer Name" + "Book Name" +
    "Borrowed Date" + "Borrowed Time" + "Price" + "\n"
  WRITE costumerName
  FOR each key in borrowedBook_ID_List
    WRITE "\t\t\t\t" to create empty space
    WRITE dictionary[key][0] + date_Borrow + time_Borrow +
      dictionary[key][3]+ "\n"
  END FOR
  WRITE "Total Amount" + total_cost(return value of costBorrow_book())
  CLOSE unique_fileName + "B" + ".txt" file
  OPEN unique_fileName + "B" + ".txt" file in read mode
  PRINT unique_fileName + "B" + ".txt".read()
  CLOSE unique_fileName + "B" + ".txt" file

```

```

DECLARE FUNCTION return_Function()
  SET costumerName,returnBooks_Id,days
  CALL dict_display() function

```

```

SET returnBooks_Id = []
SET days = []
SET return_book = False
SET name = False
WHILE name == False:
    INPUT costumerName
    IF costumerName == "" THEN
        PRINT Name cannot be left empty!!!
        Please, Enter the returner's name
    ELSE
        SET name = True
        CALL validation_ReturnBookID() function
        WHILE return_book == False:
            PRINT Do you want to return more books
            INPUT continuity
            IF continuity == "y" THEN
                CALL validation_ReturnBookID() function
            ELSE IF continuity == "n" THEN
                SET return_book = True
                PRINT Updated books list in library
                after returning books
                CALL dict_display() function
                CALL returnDetails_Write_ReadOp() function
                PRINT Thank you for using our
                library management system
            ESLE
                PRINT Invalid Input!!! Please Provide 'Y' for YES
                and 'N' for NO to return books
            END IF
        END WHILE
    END IF

```

END WHILE

```

DECLARE FUNCTION validation_ReturnBookID()
  SET bookID_List = allKey_List()
  SET dictionary = dict_Function()

  INPUT userInput_BookID
  IF userInput_BookID in bookID_List THEN
    IF userInput_BookID not in returnBooks_Id THEN
      ADD userInput_BookID to returnBooks_Id
      SET selected = False
      WHILE selected == False:
        TRY
          INPUT totalDays_Borrowed
          IF totalDays_Borrowed > 0 THEN
            SET selected = True
            ADD totalDays_Borrowed to days
            CALL updateReturn_File() function
            PRINT This book is successfully
              returned by you
        ELSE
          PRINT Invalid input!!! Please,
            provide days in positive number
        END IF
      END TRY
      EXCEPT
        PRINT Invalid input!!! Please, provide days in number
      END EXCEPT
    END WHILE
  ELSE
    PRINT This book is already returned by you.
    Please, provide another Book ID to return a book
    CALL validation_ReturnBookID() mehtod
  
```

```

END IF

ELSE

    PRINT Invalid book ID!!! Please,
        provide a valid book ID from above book table.

    CALL validation_ReturnBookID() mehtod

END IF

```

```

DECLARE FUNCTION updateReturn_File()

    SET bookID_List = allKey_List()

    SET dictionary = dict_Function()

    OPEN bookStock_list.txt in write mode

    FOR each key,value in dictionary

        IF key in returnBooks_Id of last index THEN

            SET newQuantity = returned book quantity + 1

            IF key == bookID_List of last index THEN

                WRITE value[0], value[1] ,newQuantity,
                    value[3] with next line on bookStock_list.txt

            ELSE

                WRITE value[0], value[1] ,newQuantity,
                    value[3] without next line on bookStock_list.txt

            END IF

        ELSE

            IF key == bookID_List of last index THEN

                WRITE value[0], value[1] ,value[2],
                    value[3] with next line on bookStock_list.txt

            ELSE

                WRITE value[0], value[1] ,value[2],
                    value[3] without next line on bookStock_list.txt

            END IF

        END IF

    END FOR

```

CLOSE bookStock_list.txt

```

DECLARE FUNCTION returnDetails_Write_ReadOp()

  SET dictionary = dict_Function()
  SET date_Return = date only from datetime module
  SET time_Return = time only from datetime module
  SET minute_Value = str(datetime.datetime.now().minute)
  SET microsec_Value = str(datetime.datetime.now().microsecond)
  SET unique_fileName = costumerName + minute_Value + microsec_Value
  OPEN unique_fileName + "R" + ".txt" file in write mode
  WRITE Custumer Return Books Details
  WRITE "Costumer Name" + "Book Name" +
    "Return Date" + "Return Time" + " Total Borrowed Days " + "\n"
  WRITE costumerName
  FOR each key in returnBooks_Id list
    WRITE "\t\t\t\t" to create empty space
    WRITE dictionary[key][0] + date_Return + time_Return +
      Days[i]+ "\n"
    INCREMENT i by 1
  END FOR
  WRITE "Total Fine" + totalFine_Amount (return value of cal_Fine())
  WRITE Note : fine is charged 5% of total cost of each book per extra days
  WRITE Fine will be applicable after only exceeding over 10 days
  CLOSE unique_fileName + "B" + ".txt" file
  OPEN unique_fileName + "B" + ".txt" file in read mode
  PRINT unique_fileName + "B" + ".txt".read()
  CLOSE unique_fileName + "B" + ".txt" file

```

```

DECLARE FUNCTION cal_Fine()
  SET dictionary = dict_Function()
  SET totalFine = 0

```

```

SET noCharge_duration = 10
SET i = 0
FOR each key in returnBooks_Id list
    SET rate = price of returned book
    SET borrow_days = days[i]
    IF borrow_days > noCharge_duration THEN
        SET extra_Days = borrow_days - noCharge_duration
        SET fine = 0.05 * rate * extra_Days
        SET totalFine += fine
    ELSE
        SET fine = 0
        SET totalFine += fine
    END IF
    INCREMENT i by 1
END FOR
totalFine_Amount = "$" + str(round(totalFine,2))
RETURN totalFine_Amount

```

```

DECLARE FUNCTION add_Books()
    SET book_Name as global
    SET bookName = list_BookName()
    SET bookName_NotEmpty = False
    WHILE bookName_NotEmpty == False
        INPUT book_Name
        IF book_Name == "" THEN
            PRINT Book name cannot be left empty,
            Please provide the book name.
        ELSE
            IF book_Name.lower() not in bookName
                SET bookName_NotEmpty = True

```

```

CALL userInput() function
CALL add_Write() function
PRINT This book is successfully added to book list
SET count = True
while count == True:
    PRINT Do you want to add more books
    INPUT answer
    IF answer == "y" THEN
        CALL add_Books() function
        SET count = False
    ELSE IF answer == "n"
        SET count = False
        PRINT Updated books list in
            library after adding books
        CALL dict_display() function
    ELSE
        PRINT Invalid input!!! please,
            Provide 'Y' for Yes and 'No' for No
    END IF
END WHILE
ELSE
    PRINT This book is already in book Stock list.
        Please, provide another Book Name to add book
    END IF
END IF
END WHILE

```

```

DECLARE FUNCTION add_Wrtie()
OPEN bookStock_list.txt in append mode
WRITE book_Name, book_Author_Name, book_Quantity, book_price to file
CLOSE bookStock_list.txt

```

```
DECLARE FUNCTION userInput()  
    SET book_Author_Name,book_Quantity,book_price as global  
    SET AuthorName_NotEmpty = False  
    SET quantityValue_NotInt = True  
    SET bookPrice_NotFloat = True  
WHILE AuthorName_NotEmpty == False:  
    INPUT book_Author_Name  
    IF book_Author_Name == "" THEN  
        PRINT Book author's name cannot be left empty,  
        Please provide the book author's name  
    ELSE  
        SET AuthorName_NotEmpty = True  
    END IF  
END WHILE  
WHILE quantityValue_NotInt == True:  
    TRY  
        INPUT book_Quantity  
        IF book_Quantity >= 0 THEN  
            SET quantityValue_NotInt = False  
        ELSE  
            PRINT Invalid input!!! Please provide  
            valid quantity and of positive number  
        END IF  
    END TRY  
    EXCEPT  
        PRINT Invalid input!!! Please provide quantity in number  
    END EXCEPT  
END WHILE  
WHILE bookPrice_NotFloat == True:
```

```
TRY
    INPUT book_price
    IF book_price > 0 THEN
        SET bookPrice_NotFloat = False
    ELSE
        PRINT Invalid input!!! Please provide
        price in valid price and of positive number
    END IF
END TRY
EXCEPT
    PRINT Invalid input!!! Please provide price in number
END EXCEPT
END WHILE
RETURN book_Author_Name,book_Quantity,book_price
```

2.4 Data Structures

Data Structures are a way of managing and storing all the data so that it can be accessed and carryout operation like adding, removing, storing, updating data efficiently. There are different types of data structures like integer, list, dictionary, string etc. While developing this Library management system many data structures are implemented to manage and store all the data. (Jaiswal, 2017)

Primitive Data Structure

Such data types that are already defined and supported by the programming language is called Primitive Data Structure. (Educative, Inc, 2021)

Integers

Integers are whole numeric data that represent from negative to positive all whole numbers. Example 1,2,42, -4, -2.

Integer's data structure is implemented various places in this application while developing it.

```
def Cal_Fine():
    '''calculates the total fine to be paid for late returning of'''

    dictionary = dict_Function()
    totalFine = 0
    noCharge_duration = 10#defining maximum days for no fine charge
    i = 0#setting counter for index
```

Figure 12 : Screenshot of implementation of integer data structure.

In the above figure an integer value 10 is assign to a variable noChare_duration and an integer value 0 is assign to a variable i. In this way integer value is assign and define in python.

```
#handing ValueError exception using try except block
totalDays_Borrowed = int(input("Enter the numbers of days you borrowed a book : "))#taking integer number from user
if totalDays_Borrowed > 0:
    selected = True#boolean value for exiting while loop
```

Figure 13 : Screenshot of asking integer value from user.

In the above figure, an user is asked for value input and that value is converted in to integer using int() function. This function helps to only accepts integer value. If other value beside integer is given then it will throw “ValueError” exception.

Float

Float are usually deals with the rational numbers having decimal value at the end.

Example: 1.2, 0.3, 0.2

Float value are also define same way as integer value .

Example:

a = 1.2

b = 0.3

print(a)

print(b)

here, decimal value are assign to the variables and declare as float variable and print statement help to print value of a and b.

```
#handing ValueError exception using try except block
book_price = float(input("please, provide the book price : "))
if book_price > 0:
```

Figure 14 : Screenshot of asking float value from user.

In the above figure, a user is asked for value input and that value is converted in to float using float() function. This function helps to only accepts integer value or number with decimal. If other value beside accepts integer value or number with decimal is given then it will throw “ValueError” exception.

Boolean

Boolean deals with only values like “True” and “False”. This type of data type is useful in conditional, comparing, expressions.

```

continue_1 = True
while continue_1 == True:#creating a loop to keep asking a value to user if invalid input is given
    print("\nPlease, Enter '1' to borrow a book.")
    print("Please, Enter '2' to return a book.")
    print("Please, Enter '3' to add a new book.")
    print("Please, Enter '4' to exit.")
    user_Input = input("\nPlease, Enter a value as 1 or 2 or 3 or 4 : ")#asking value 1,2,3 from user
    if user_Input == str(1):#converting 1 interger to string and comparing wth user input value
        print("\n\n-----")
        print("You will now borrow the book :")
        print("-----")
        borrow_Function()#borrow_Function method will be invoked if user input value is 1
    else:
        print("-----")

```

Figure 15 : Screenshot of implementation of boolean data structure.

In the above, figure a boolean is assign to a variable `continue_1` and that expressing is checking condition in while loop which allows only if expressing is Ture for next step. In this way, a boolean value is assign and expression are checked for condition being True.

Strings

Strings are like container used to store data which involves a sequence of characters. (e.g., addresses or messages). In Python, Strings are defined or created by enclosing a sequence of characters within a pair of single or double-quotes. Strings are immutable as their original string cannot be modify once created. A group of String are joined using “+” operator in python.

Strings can be defined inside a pair of double quotes like below:

```

s = "Hello"
print(s)
output --- Hello

```

```
continuity -- rate           rate = float(dictionary[key][3])
costumerName = input()           totalcost += rate # adding rate
if costumerName ==               total_cost = "$" + str(totalcost)#
    print("-----")             return total_cost#returns the value
    print("\t\t\t\tNa")
    print("-----")
```

Figure 16 : Screenshot of implementation of print function with strings.

In above, figure Strings are directly pass in to print() function to print messages and strings input can be asked from user simply using input() function as in python input() takes value as strings in default.

Figure 17 : Screenshot of data type conversion float to string.

Here in this figure, using str() function float data type is changes into string datatype and concatenate with “\$” sign using “+” operator. In this way string is joined and assign to variables as in figure.

Some function to manipulate strings value:

`upper()` and `lower()` function

`upper()` function changes all strings characters in to capital letter and `lower()` function changes all strings characters in to small letter but as strings are immutable original strings value is not changed.

Eq.

```
s = "Hello"
```

```
print(s.upper()) #prints "HELLO"
```

```
print(s.lower()) #prints "hello"
print(s) # prints "Hello" no change
```

replace() function

using replace () function, certain letter,symbol etc are replace in strings.

Eg:

```
a = "Hello"
print(a.replace('H','h') #prints "hello"
print(a) # prints "Hello" no change as original strings are immutable
```

split() function and join() function

split() function help to convert string in to list and join() function help to convert list in to string.

Eg:

```
a = 'apple banana mango'
furits = a.split()
b = "".join(furits)
print(furits)#prints ['apple','banana','mango']
print(b) #prints 'apple banana mango'
print(a)#prints 'apple banana mango'
```

```
file = open("bookStock_list.txt","r")#opening txt file in read mode
dictionary = {}#creating empty dictionary
book_ID = 0
for line in file:#iterates all line as string from txt file
    book_ID += 1#increment by 1 each time
    line = line.replace("\n","").split(',')#replace next line character and split by comma
    dictionary[str(book_ID)] = line # adding book ID as key & list as value
file.close()#closing file
return dictionary
```

Figure 18 : Screenshot of implementation of split and replace function in program.

In above, figure strings variable line, using split() function is converted in to list and in that list “\n” is replace with empty string.

Non-Primitive Data Structure

Such data types that are derived from primitive data types which offers increased or some extra functionality in programming is Non-Primitive Data Structure. (Educative, Inc, 2021)

Lists

Lists are ordered sequence of information which is accessible by index that are used to store collections of item or data of same data type or different data type. List is denoted by square brackets “[]” which can hold items or elements and are separated by a comma. List are mutable it means that data or items in list can be changed.

```
key_List = []#creating empty lsit
dictionary = dict_Function()#assiging return value of method to dictionary variable
for key in dictionary.keys():#iterates all key from dictionary one by one
    key_List.append(key)# adding key to list
return key_List#returns list
```

Figure 19 : Screenshot of defining empty list and adding value to it.

In above, figure an empty list is define and assign to key_List variable and using the append() function all the key from dictionary is add to the list. This list help to hold all the key which is actually books ID.

```

dictionary = dict_Function()
totalFine = 0
noCharge_duration = 10#defining maximum days for no fine
i = 0#setting counter for index
for key in returnBooks_Id:#iterates all key from list one
    rate = float(dictionary[key][3].replace("$",""))#replacing $ to float
    borrow_days = days[i]#iterating each days value one by one
    if borrow_days > noCharge_duration:#checking if borrow days is greater than no charge duration
        extra_Days = borrow_days - noCharge_duration#calculating extra days
        fine = 0.05 * rate * extra_Days#fine is charged 5% of extra days
        totalFine += fine#adding rate for calculating total fine
    else:
        fine = 0#setting fine to zero if lending days is less than or equal to no charge duration
        totalFine += fine
    i += 1#incrementing counter value by 1
totalFine_Amount = "$" + str(round(totalFine,2))#converting float to string
return totalFine_Amount#returns the value of totalFine_Amount

```

Figure 20 : Screenshot of list elements access by using indexing.

In above figure, `days[]` list's each items or data is accessed using index variable "i" which value will start from "0" as index of list also from "0" and increment value of "i" by 1 every time loop executed. This index variable helps to point index where data or items or certain value is present and assign to variable by extracting that value or data. Here also, `days[]` list each value is accessed and extracted and assign to `borrow_days` variable.

Defining and changing list items:

```

a = [ 1,2,3,4,5,6]
a[0] = 9#replacing item at index "0" with new item or value
print(a) #prints [9,2,3,4,5,6]

```

Some functions of lists are:

`len()` function

This function returns the length or size of list.

Eg:

```
a = [ 1,2,3,4,5,6]  
print(len(a)) # prints 6 which is length or size of list
```

append() function and extend()

append() function helps to add items or values to the end of the list and extend() function help to add multiple items or values to the end of the list.

Eg.

```
a = [ 1,2,3,4,5,6]  
a.append(7)  
print(a) # prints [ 1,2,3,4,5,6,7]  
a.extend([8,9,10])  
print(a) # prints [1,2,3,4,5,6,7,8,9,10]
```

remove() function, del() function, pop() function

remove() function help to remove specific item or element from the list. This function only removes first occurrence of item or element if item or element occurs multiple times and provide an error if that item or element is not found in list.

Eg:

```
a = [ 1,2,3,4,5,6,6]  
a.remove(1)  
print(a) #prints [2,3,4,5,6,6]  
a.remove(6)  
print(a) #prints [2,3,4,5,6] –here 6 occurs twice so only 1st occurrence is removed
```

del() function help to delete item or element at specific index.

Eg:

```
a = [ 1,2,3,4,5,6]  
a.del(a[1])  
print(a) #prints [1,3,4,5,6]
```

pop() function help to remove item or element at the end of list and returns that item or element.

```
a = [ 1,2,3,4,5,6]
```

```
a.pop()
```

```
print(a)#prints [1,2,3,4,5] and returns 6
```

Tuples

Tuples another type of an ordered sequence of items or elements. Tuples is denoted by parentheses brackets “()” which can hold items or elements and are separated by a comma. Unlike, lists tuples are immutable which means items or elements cannot be changed. It is usually used in declaring constant or storing constant.

Eg:

```
a = (1,2,3,4,5)
```

```
print(len(a)) #prints the length of tuples i.e.,5
```

```
print(a[0])#prints 1 as list items or element is access using index
```

```
a[0] = 3 or a.append(8) #provides error as tuples element or items cannot be changed
```

```
b = () #defining empty tuples
```

Dictionary

A dictionary is an unordered, changeable, and indexed collection of key – value pairs with unique keys. Dictionary are mutable which means items or elements can be added, removed, and changes easily. Dictionary is denoted by curly braces “{ }” which can hold key and value pairs and each pairs are separated by a comma. Keys of dictionary must

be unique and immutable type and value of dictionary can be any mutable type, be duplicates. Dictionary can have lists or even dictionaries as its value

Eg:

```
a = {1: 'Hero', 2:'bisheas'}
b ={1:{'ram':70}, 2:{'hari':80}} #dictionary having its value as dictionary
c = {1:['ram',70],2:['hari',80]} # dictionary having Its value as lists
```

Here, key value is separated by using ":" and key-value pairs is separated by commas.

```
def dict_Function():
    ''' creates dictionary collection data type from bookStock txt file'''

    file = open("bookStock_list.txt","r")#opening txt file in read mode
    dictionary = {}#creating empty dictionary
    book_ID = 0
    for line in file:#iterates all line as string from txt file one line by line
        book_ID += 1#increment by 1 each time
        line = line.replace("\n","").split(',')#replace next line with empty space in s
        dictionary[str(book_ID)] = line # adding book ID as key and  value list of string
    file.close()#closing file
    return dictionary
```

Figure 21 : Screenshot of creating empty dictionary and adding key-value pair to it.

In the above figure, an empty dictionary is created named dictionary and book_ID variable value is used as key string data type and line variable which is lists of string data type that hold the book details created from bookStock_list.txt file is used as value. And all those all added in to empty dictionary using syntax:

Dictionary[key value] = value

```
def dict_display():
    '''displays list of book in a table'''

    print("-----")
    print("Book ID\t\t" + "\tBook Name \t\t" + "\tAuthor \t\t" + "\tQuantity \t\t" + "\tPrice")
    print("-----")
    display_dictioanry = dict_Function()
    for key,value in display_dictioanry.items():#iterates all key,value from dictionary one by one
        print(key + "\t\t" + value[0] + "\t\t" + value[1] + "\t\t" + value[2] + "\t\t" + value[3])
    print("-----\n")

#-----borrow function of program-----
```

Figure 22 : Screenshot of iterating and accessing all key and value of dictionary.

In this figure, dic_Function's return value i.e., dictionary is assign to display_dicitonary to display all the keys as book ID and values as book details using for loop that iterates the key and value of dictionary. Here, to access the particular element in value of dictionary while in for loop, a index number is used as value of dictionary that's holds element is lists.

Operation on dictionaries:

```
english = { 'bishwas' : 88.0, 'hari': 34, 'wanni':50}#defining dictionary  
print(english['hari']) #prints 34 it prints value of that key
```

```
english['hari'] = 70 # modifying the value of hari.
```

```
print(english['hari'])# prints 70
```

```
english['ram'] = 89 # adding new key value pairs to existing dictionary  
print(english) # prints { 'bishwas' : 88.0, 'hari': 34, 'wanni':50, 'ram':89}
```

del() function

This function help to delete particular key value.

```
english = { 'bishwas' : 88.0, 'hari': 34, 'wanni':50}
del(english['bishwas'])
print(english) #prints {"hari": 34, 'wanni':50}
```

Sets

Unordered collection of items or elements are called Sets. Sets is denoted by curly braces “{ }” which can hold items or elements and are separated by a comma. The elements or items in sets are unique which means duplicating items or element is point less as it only gives unique items or elements as output.

Eg:

```
a = {1,2,3,4}
```

Some functions in Sets are:

union()

This function help to print all the unique number present in both sets a and b as duplicated items or elements are ignored or replace automatically.

Eg:

```
a = {1,2,3,4,4}
b = {4,5,6,7,8}
print(a.union(b)) #prints {1,2,3,4,5,6,7,8}
```

intersection() function

This function help to print all the unique number present in both sets a and b as same items or elements only

Eg:

```
a = {1,2,3,4,4}  
b = {4,5,6,7,8}  
print(a. intersection (b)) #prints {4}
```

difference()

This function help to print all the unique number present that only sets a has in respect to set b.

Eg:

```
a = {1,2,3,4,4}  
b = {4,5,6,7,8}  
print(a. difference (b)) #prints {1,2,3}
```

symetic_different()

This function help to print all the unique number that only sets a and set b has as common or duplicated number is replaced or ignored in output

Eg:

```
a = {1,2,3,4,4}  
b = {4,5,6,7,8}  
print(a. symetic_different (b)) #prints {1,2,3,5,6,7,8}
```

len() function

This function returns length or size of sets.

Eg:

```
a = {1,2,3,4}  
print(len(a)) #prints 4 –size of set
```

add() function

This help to add the item or element in set but item or element placement is random can be any place inside curly braces.

Eg:

```
a = {1,2,3,4}
```

```
a.add(5)
```

```
print(a) #prints {1,2,3,4,5}
```

again adding 5 to set a

```
a.add(5)
```

```
print(a) #prints {1,2,3,4,5} replace or ignore duplicate element in output
```

`remove()`

This function help to remove item or element in set.

Eg:

```
a = {1,2,3,4}
```

```
a.remove(1)
```

```
print(a) # prints {2,3,4}
```

3. Program

3.1 Implementation of Program

```
--- REJIAKI : C:\Users\Uyananda\Desktop\FC_Coursework\My Development\main.py ---
*****
***** Welcome to College Library Management System. *****
*****



-----
```

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	26	\$4
2	Start With Why	Simon Sinek	22	\$3.5
3	Programming	Jhon Smith	14	\$3.5
4	IT Networking	William Shakes	12	\$5
5	The Castle	Firdausi	10	\$4
6	Fear of Flying	Erica Jong	0	\$5.5
7	Python program	bishwas limbu	11	\$3.0
8	Silent Hill	J.K Chettri	10	\$2.0
9	Web Design	J.K Macloan	10	\$3.0
10	Watch Maker	Logan Macron	13	\$3.0

```
Please, Enter '1' to borrow a book.
Please, Enter '2' to return a book.
Please, Enter '3' to add a new book.
Please, Enter '4' to exit.

Please, Enter a value as 1 or 2 or 3 or 4 :
```

Figure 23 : Screenshot of main user interface of library management system.

Above figure, user interface is displayed when program is run from the starting point. Here, welcome message is displayed with all available book's list details in table form and library menu option for user is also displayed and it will ask user to provide 1,2,3 or 4 value to continue further.

- If user want to borrow a book, then they have to provide input value “1”.
- If user want to return a book, then they have to provide input value “2”.
- If user want to add a book, then they have to provide input value “3”
- If user want to exit or terminate program then they have to provide input value “4”

7		Python program		bishwas limbu		11		\$3.0
8		Silent Hill		J.K Chettri		10		\$2.0
9		Web Design		J.K Macloan		10		\$3.0
10		Watch Maker		Logan Macron		13		\$3.0
<hr/>								
Please, Enter '1' to borrow a book.								
Please, Enter '2' to return a book.								
Please, Enter '3' to add a new book.								
Please, Enter '4' to exit.								
Please, Enter a value as 1 or 2 or 3 or 4 : <u> </u>								
<hr/>								
Invalid input!!! Please, provide input value as 1, 2,3 or 4.								
<hr/>								
Please, Enter '1' to borrow a book.								
Please, Enter '2' to return a book.								
Please, Enter '3' to add a new book.								
Please, Enter '4' to exit.								
Please, Enter a value as 1 or 2 or 3 or 4 : <u>one</u>								
<hr/>								
Invalid input!!! Please, provide input value as 1, 2,3 or 4.								
<hr/>								
Please, Enter '1' to borrow a book.								
Please, Enter '2' to return a book.								
Please, Enter '3' to add a new book.								
Please, Enter '4' to exit.								
Please, Enter a value as 1 or 2 or 3 or 4 : <u>5</u>								
<hr/>								
Invalid input!!! Please, provide input value as 1, 2,3 or 4.								
<hr/>								
Please, Enter '1' to borrow a book.								
Please, Enter '2' to return a book.								
Please, Enter '3' to add a new book.								
Please, Enter '4' to exit.								
Please, Enter a value as 1 or 2 or 3 or 4 : <u>-1</u>								
<hr/>								
Invalid input!!! Please, provide input value as 1, 2,3 or 4.								
<hr/>								
Please, Enter '1' to borrow a book.								
Please, Enter '2' to return a book.								
Please, Enter '3' to add a new book.								
Please, Enter '4' to exit.								
Please, Enter a value as 1 or 2 or 3 or 4 :								

Figure 24 : Screenshot of an error message when negative, string, empty, non-existed value are provided as input.

After requesting for input to the user, if user provides any empty value, negative number, string value or any positive number beside 1,2,3 and 4. This will give an error

message say “Invalid input!!! Please, provide input value as 1,2,3 or 4 : “ then again displays a menu option to user and request again user to input value showing message “Please, Enter a value as 1 or 2 or 3 or 4:”. This process will repeat again and again until user provides valid input.

3.2 Borrow Process

```
Please, Enter '1' to borrow a book.  
Please, Enter '2' to return a book.  
Please, Enter '3' to add a new book.  
Please, Enter '4' to exit.
```

```
Please, Enter a value as 1 or 2 or 3 or 4 : 1
```

```
You will now borrow the book :
```

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	26	\$4
2	Start With Why	Simon Sinek	22	\$3.5
3	Programming	Jhon Smith	14	\$3.5
4	IT Networking	William Shakes	12	\$5
5	The Castle	Firdausi	10	\$4
6	Fear of Flying	Erica Jong	0	\$5.5
7	Python program	bishwas limbu	11	\$3.0
8	Silent Hill	J.K Chettri	10	\$2.0
9	Web Design	J.K Macloan	10	\$3.0
10	Watch Maker	Logan Macron	13	\$3.0

```
Please, Enter the borrower's name : |
```

Figure 25 : Screenshot of user interface for borrowing books.

After the user provides input value “1”, then borrow process is carried out. Above figure, user interface is displayed for borrow process where it displays a message “ you will now borrow the book” with all available book’s list details in table form and request user to input borrower’s name.

You will now borrow the book :

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	26	\$4
2	Start With Why	Simon Sinek	22	\$3.5
3	Programming	Jhon Smith	14	\$3.5
4	IT Networking	William Shakes	12	\$5
5	The Castle	Firdausi	10	\$4
6	Fear of Flying	Erica Jong	0	\$5.5
7	Python program	bishwas limbu	11	\$3.0
8	Silent Hill	J.K Chettri	10	\$2.0
9	Web Design	J.K Macloan	10	\$3.0
10	Watch Maker	Logan Macron	13	\$3.0

Please, Enter the borrower's name :

Name cannot be left empty!!! Please, Enter the borrower's name.

Please, Enter the borrower's name :

Figure 26 : Screenshot of an error message when user pass empty value.

If user hits the “Enter” button without providing a value for borrower’s name, then it will display error message saying “Name cannot be left empty!!! Please, Enter the borrower’s name”. Program will keep requesting for borrower’s name showing same error message until user provides value for borrower’s name.

You will now borrow the book :

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	26	\$4
2	Start With Why	Simon Sinek	22	\$3.5
3	Programming	Jhon Smith	14	\$3.5
4	IT Networking	William Shakes	12	\$5
5	The Castle	Firdausi	10	\$4
6	Fear of Flying	Erica Jong	0	\$5.5
7	Python program	bishwas limbu	11	\$3.0
8	Silent Hill	J.K Chettri	10	\$2.0
9	Web Design	J.K Macloan	10	\$3.0
10	Watch Maker	Logan Macron	13	\$3.0

Please, Enter the borrower's name :

Name cannot be left empty!!! Please, Enter the borrower's name.

Please, Enter the borrower's name : Hari Rai
Enter book ID you want to borrow :

Figure 27 : Screenshot of next process after user provide valid borrower's name.

After providing name for borrower's name, it will request for the book ID with message saying "Enter book ID you want to borrow :".

```
Please, enter the borrower's name :  
-----  
Name cannot be left empty!!! Please, Enter the borrower's name.  
-----  
Please, Enter the borrower's name : Hari Rai  
Enter book ID you want to borrow : _____  
-----  
Invalid book Id!!! Please, provide valid Book ID from above book table.  
-----  
Enter book ID you want to borrow : one  
_____  
-----  
Invalid book Id!!! Please, provide valid Book ID from above book table.  
-----  
Enter book ID you want to borrow : -1  
_____  
-----  
Invalid book Id!!! Please, provide valid Book ID from above book table.  
-----  
Enter book ID you want to borrow : 11  
_____  
-----  
Invalid book Id!!! Please, provide valid Book ID from above book table.  
-----  
Enter book ID you want to borrow :  
-----
```

Figure 28 : Screenshot of an error message when user pass empty, string, negative, non-existed value to Book Id.

If user provide any string value, negative number value, hits "Enter" without passing a value, or provides positive number value which do not matches any Book ID of available books list, then it will provide an error message saying "Invalid book ID!!! Please, provide valid book ID from above book table.". After that program request user to provide Book ID again and again until user provide valid book ID.

You will now borrow the book :						
Book ID	Book Name	Author	Quantity	Price		
1	Harry Potter	Jk Rowling	26	\$4		
2	Start With Why	Simon Sinek	22	\$3.5		
3	Programming	Jhon Smith	14	\$3.5		
4	IT Networking	William Shakes	12	\$5		
5	The Castle	Firdausi	10	\$4		
<u>6</u>	<u>Fear of Flying</u>	<u>Erica Jong</u>	<u>0</u>	<u>\$5.5</u>		
7	Python program	bishwas limbu	11	\$3.0		
8	Silent Hill	J.K Chettri	10	\$2.0		
9	Web Design	J.K Macloan	10	\$3.0		
10	Watch Maker	Logan Macron	13	\$3.0		

Figure 29 : Screenshot of all available book list where Book ID "6" has book quantity zero.

```
Enter book ID you want to borrow : -1
-----
Invalid book Id!!! Please, provide valid Book ID from above book table.
-----
Enter book ID you want to borrow : 11
-----
Invalid book Id!!! Please, provide valid Book ID from above book table.
-----
Enter book ID you want to borrow : 6
-----
Book is not available for now. Please, Select another book from the stock.
-----
Enter book ID you want to borrow : |
```

Figure 30 : Screenshot of an error message when user try to borrow book having quantity zero.

If user provides valid book ID to borrow a book having book quantity “0”, then it will provide an error message saying “Book is not available for now. Please, Select another book from the stock. “And after that program request user to provide Book ID again and again until user provide valid book ID of book having book quantity value greater than “0”.

```

Enter book ID you want to borrow : 1
*****
This Book is available for borrow.
*****
Book Name | Price | Date of Borrow | Time of Borrow
-----
Harry Potter | $4 | 21-09-09,Thu | 15:16
Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books :

```

Figure 31 : Screenshot of borrowed book details after valid book ID was given.

When user provides valid book ID of book having quantity greater than “0” as in the figure above, book is borrowed having that book ID and displays message “ This Book is available for borrow.” and book is borrowed. After that borrowed book name , price , date and time of borrow is also displayed.

```
Harry Potter | $4 | 21-09-09,Thu | 15:16
```

```
Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books :
```

```
-----  
Invalid Input!!! Please, Provide 'Y' for YES and 'N' for NO to borrow more books.  
-----
```

```
Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books :
```

```
-----  
Invalid Input!!! Please, Provide 'Y' for YES and 'N' for NO to borrow more books.  
-----
```

```
Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books :
```

```
-----  
Invalid Input!!! Please, Provide 'Y' for YES and 'N' for NO to borrow more books.  
-----
```

```
Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books :
```

```
-----  
Invalid Input!!! Please, Provide 'Y' for YES and 'N' for NO to borrow more books.  
-----
```

```
Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books :
```

Figure 32 : Screenshot of an error message when user pass any value beside 'y' and 'n' in upper or lower case.

After borrowing one book, program will display a message saying “Do you want to borrow more books” and ask for user response if they want to borrow other books by showing message “If you want then please, Enter ‘Y’ for YES and ‘N’ for NO to borrow more books”. If user provide any response beside ‘Y’ or ‘N’ then it will provide an error message saying “Invalid input!!! Please, provide ‘Y’ for YES and ‘N’ for NO to borrow more books” and again ask for user response for borrowing more book again and again until user provide response ‘Y’ for YES and ‘N’ for NO to borrow more books.

```
-----
Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books : y
-----
Enter book ID you want to borrow : |
```

Figure 33 : Screenshot of next process after user provide 'y' for program query.

After user provide response by giving input ‘y’, it will start again borrowing process for other books. Program will keep asking user for borrowing more books until user response input will be “n”.

```
-----
Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books : |
-----
Enter book ID you want to borrow : 2
*****
This Book is available for borrow.
*****
Book Name | Price | Date of Borrow | Time of Borrow
-----
Start With Why | $3.5 | 21-09-09,Thu | 15:44
-----
Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books : |
```

Figure 34 : Screenshot of borrowing another book by user.

```

Enter book ID you want to borrow : 2
*****
          This Book is available for borrow.
*****
Book Name      |     Price      |     Date of Borrow      |     Time of Borrow
-----
Start With Why |     $3.5       |     21-09-09,Thu        |     15:44
Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books : y

Enter book ID you want to borrow : 2
-----
          This book is already borrowed by you. Please, select another book from the stock.
-----
Enter book ID you want to borrow : 3
*****
          This Book is available for borrow.
*****
Book Name      |     Price      |     Date of Borrow      |     Time of Borrow
-----
Programming    |     $3.5       |     21-09-09,Thu        |     15:44
Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books : |

```

Figure 35 : Screenshot of an error message when user try to borrow same book twice.

Here, in the above figure program is not allowing user to borrow same book twice in each transaction. If user try to borrow same book like in the figure, it will provide an error message saying “This book is already borrowed by you. Please, select another book from the stock” and again request user to input valid book Id of another book. If user input another valid book id then that book is borrowed and program again ask for user response to borrow more books. This process in repeated until user doesn’t borrow books. If user don’t want to borrow any more book, user need to provide ‘n’ for NO.

Programming	\$3.5	21-09-09,Thu	15:44
-------------	-------	--------------	-------

Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books : n

Updated books list in library after borrowing books :

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	25	\$4
2	Start With Why	Simon Sinek	21	\$3.5
3	Programming	Jhon Smith	13	\$3.5
4	IT Networking	William Shakes	12	\$5
5	The Castle	Firdausi	10	\$4
6	Fear of Flying	Erica Jong	0	\$5.5
7	Python program	bishwas limbu	11	\$3.0
8	Silent Hill	J.K Chettri	10	\$2.0
9	Web Design	J.K Macloan	10	\$3.0
10	Watch Maker	Logan Macron	13	\$3.0

Custumer Borrow Books Details

Costumer Name	Book Name	Borrowed Date	Borrowed Time	Price
Hari Rai	Harry Potter	21-09-09,Thu	15:44	\$4
	Start With Why	21-09-09,Thu	15:44	\$3.5
	Programming	21-09-09,Thu	15:44	\$3.5
Total Amount				\$11.0

Figure 36 : Screenshot of process after user pass 'n' value to program query where update book list and borrowed bill is displayed,

After user provide response input as "n" for No, then as above figure a message will appear saying "Updated books list after borrowing books." and updated available book's list details after borrowing process is displayed in table. After that a costumer borrow bill is generated in text file note and bill is displayed in the program with costumer name, book name, date, time and price of each book and total amount to be paid for borrowed books. Here in the figure book ID 1, 2 and 3 book quantities are decrease by 1 and red box marking indicating updated value.

Customer Borrow Books Details						
Customer Name	Book Name	Borrowed Date	Borrowed Time	Price	Total Amount	
Hari Rai	Harry Potter	21-09-09, Thu	15:44	\$4		
	Start With Why	21-09-09, Thu	15:44	\$3.5		
	Programming	21-09-09, Thu	15:44	\$3.5		
					Total Amount	\$11.0

Thank you for using our library management system.

Please, Enter '1' to borrow a book.
Please, Enter '2' to return a book.
Please, Enter '3' to add a new book.
Please, Enter '4' to exit.

Please, Enter a value as 1 or 2 or 3 or 4 : |

*Figure 37 : screenshot of costumer borrow books details bill.***Creation of text file for borrowing book:**

Hari Rai44109902B	9/9/2021 4:09 PM	Text Document	1 KB
main	9/8/2021 5:29 PM	Python File	1 KB

*Figure 38 : Screenshot of text file note generated with unique file name after borrowing process.***Costumer borrow books details note is generated as bill with unique file name.**

Customer Borrow Books Details						
Customer Name	Book Name	Borrowed Date	Borrowed Time	Price	Total Amount	
Hari Rai	Harry Potter	21-09-09, Thu	15:44	\$4		
	Start With Why	21-09-09, Thu	15:44	\$3.5		
	Programming	21-09-09, Thu	15:44	\$3.5		
					Total Amount	\$11.0

*Figure 39 : Screenshot of costumer borrow books details in text file.***Above figure is bill note for costumer borrow books details generated after borrowing books in text file.**

3.3 Returned Process

```
Please, Enter '1' to borrow a book.  
Please, Enter '2' to return a book.  
Please, Enter '3' to add a new book.  
Please, Enter '4' to exit.  
  
Please, Enter a value as 1 or 2 or 3 or 4 : 2  
  
-----  
You will now return the book :  
-----  


| Book ID |  | Book Name      |
|---------|--|----------------|
| 1       |  | Harry Potter   |
| 2       |  | Start With Why |
| 3       |  | Programming    |
| 4       |  | IT Networking  |
| 5       |  | The Castle     |
| 6       |  | Fear of Flying |
| 7       |  | Python program |
| 8       |  | Silent Hill    |
| 9       |  | Web Design     |
| 10      |  | Watch Maker    |

  
-----  
Please, Enter the returner's name : |
```

Figure 40 : Screenshot of user interface for returning process.

After the user provides input value “2”, then return process is carried out. Above figure, user interface is displayed for return process where it displays a message “ you will now return the book” with all available book’s list details in table form and request user to input returner’s name.

You will now return the book :

Book ID		Book Name		Author		Quantity		Price
1		Harry Potter		Jk Rowling		25		\$4
2		Start With Why		Simon Sinek		21		\$3.5
3		Programming		Jhon Smith		13		\$3.5
4		IT Networking		William Shakes		12		\$5
5		The Castle		Firdausi		10		\$4
6		Fear of Flying		Erica Jong		0		\$5.5
7		Python program		bishwas limbu		11		\$3.0
8		Silent Hill		J.K Chettri		10		\$2.0
9		Web Design		J.K Macloan		10		\$3.0
10		Watch Maker		Logan Macron		13		\$3.0

Figure 41 : Screenshot of an error message when user doesn't pass any value to returner's name.

If user hits the “Enter” button without providing a value for returner’s name, then it will display error message saying “Name cannot be left empty!!! Please, Enter the returner’s name”. Program will keep requesting for returner’s name showing same error message until user provides value for returner’s name.

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	25	\$4
2	Start With Why	Simon Sinek	21	\$3.5
3	Programming	Jhon Smith	13	\$3.5
4	IT Networking	William Shakes	12	\$5
5	The Castle	Firdausi	10	\$4
6	Fear of Flying	Erica Jong	0	\$5.5
7	Python program	bishwas limbu	11	\$3.0
8	Silent Hill	J.K Chettri	10	\$2.0
9	Web Design	J.K Macloan	10	\$3.0
10	Watch Maker	Logan Macron	13	\$3.0

Please, Enter the returner's name :

Name cannot be left empty!!! Please, Enter the returner's name.

Please, Enter the returner's name : Bishwas Limbu
Enter book ID you want to return : |

Figure 42 : Screenshot of next process after user provide valid name.

After providing name for returner’s name, it will request for the book ID with message saying “Enter book ID you want to return:”.

6	Fear of Flying	Erica Jong	0	\$5.5
7	Python program	bishwas limbu	11	\$3.0
8	Silent Hill	J.K Chettri	10	\$2.0
9	Web Design	J.K Macloan	10	\$3.0
10	Watch Maker	Logan Macron	13	\$3.0

Please, Enter the returner's name :

Name cannot be left empty!!! Please, Enter the returner's name.

Please, Enter the returner's name : Bishwas Limbu
Enter book ID you want to return : |

Invalid book ID!!! Please, provide a valid book ID from above book table.

Enter book ID you want to return : two

Invalid book ID!!! Please, provide a valid book ID from above book table.

Enter book ID you want to return : -3

Invalid book ID!!! Please, provide a valid book ID from above book table.

Enter book ID you want to return : 13

Invalid book ID!!! Please, provide a valid book ID from above book table.

Enter book ID you want to return : |

Figure 43 : Screenshot of an error message when user pass empty, string, negative, and non-existed value to book ID.

If user provide any string value, negative number value, hits “Enter” without passing a value, or provides positive number value which do not matches any Book ID of available books list, then it will provide an error message saying “Invalid book ID!!! Please, provide valid book ID from above book table.”. After that program request user to provide Book ID again and again until user provide valid book ID.

```
Enter book ID you want to return : 13
-----
                                Invalid book ID!!! Please, provide a valid book ID from above book table.
-----
Enter book ID you want to return : 6
Enter the numbers of days you borrowed a book : |
```

Figure 44 : Screenshot of next process after user provide valid book ID.

After valid book ID is given as input as in the figure, program will request user to enter the numbers of day book was borrowed with message saying “Enter the number of days you borrowed a book”.

```
Enter the numbers of days you borrowed a book : eleven
-----
                                Invalid input!!! Please, provide days in number.
-----
Enter the numbers of days you borrowed a book : |
```

Figure 45 : Screenshot of an error message after user provide string value to total day borrowed.

```
Enter book ID you want to return : 6
Enter the numbers of days you borrowed a book : |
-----
                                Invalid input!!! Please, provide days in number.
-----
```

Figure 46 : Screenshot of an error message after user provide no value to total day borrowed.

```
Enter the numbers of days you borrowed a book : -1  
-----  
Invalid input!!! Please, provide days in positive number.  
-----
```

Figure 47 : Screenshot of an error message after user provide negative value to total day borrowed.

```
Enter the numbers of days you borrowed a book : 0  
-----  
Invalid input!!! Please, provide days in positive number.  
-----  
Enter the numbers of days you borrowed a book : |
```

Figure 48 : Screenshot of an error message after user provide zero value to total day borrowed.

If user provide 0 or any negative value as input to days borrowed book then it will provide an error message saying “Invalid input!!! Please, provide days n positive number.” and program will request again to provide number of days to user until valid number of days is provided. If user provide string value or hit “Enter” keyword without passing value then it will provide a error message saying “Invalid input!!! Please, provide days in number” and program will request again to provide number of days to user until valid number of days is provided.

```
Enter the numbers of days you borrowed a book : 11  
-----  
This book is successfully returned by you.  
-----  
  
Do you want to return more books ?  
If you want then Enter 'Y' for YES and 'N' for NO to return more books : |
```

Figure 49 : Screenshot of book returned successfully after user provide valid value to total day borrowed.

After valid input for day provided by user, the book will be successfully returned and it will provide a message saying “This book is successfully returned by you” and program will ask user providing a message “Do you want to return more books?” and program request user to provide response ‘y’ for YES and ‘n’ for NO.

```
Do you want to return more books ?  
If you want then Enter 'Y' for YES and 'N' for NO to return more books : _____
```

```
-----  
Invalid Input!!! Please Provide 'Y' for YES and 'N' for NO to return books..  
-----
```

Figure 50 : Screenshot of an error message after user provide no value to program query to return more books.

```
Do you want to return more books ?  
If you want then Enter 'Y' for YES and 'N' for NO to return more books : 1  
_____
```

```
-----  
Invalid Input!!! Please Provide 'Y' for YES and 'N' for NO to return books..  
-----
```

Figure 51 : Screenshot of an error message after user provide number value to program query to return more books.

```
Do you want to return more books ?  
If you want then Enter 'Y' for YES and 'N' for NO to return more books : k  
_____
```

```
-----  
Invalid Input!!! Please Provide 'Y' for YES and 'N' for NO to return books..  
-----
```

```
Do you want to return more books ?  
If you want then Enter 'Y' for YES and 'N' for NO to return more books :  
_____
```

Figure 52 : Screenshot of an error message after user provide other string value to program query to return more books.

If user provide any response beside 'Y' or 'N' then it will provide an error message saying "Invalid input!!! Please, provide 'Y' for YES and 'N' for NO to borrow more books" and again ask for user response for returning more book again and again until user provide response 'Y' for YES and 'N' for NO to return more books.

```
Do you want to return more books ?
If you want then Enter 'Y' for YES and 'N' for NO to return more books : y
```

```
Enter book ID you want to return :
```

Figure 53 : Screenshot of process after user provide 'y' to program query to return more books.

After user provide response by giving input 'y', it will start again returning process for other books. Program will keep asking user for returning more books until user response input will be "n".

```
Do you want to return more books ?
If you want then Enter 'Y' for YES and 'N' for NO to return more books : y
```

```
Enter book ID you want to return : 8
Enter the numbers of days you borrowed a book :
```

Figure 54 : Screenshot of user providing valid book ID.

```
Do you want to return more books ?
If you want then Enter 'Y' for YES and 'N' for NO to return more books : y
```

```
Enter book ID you want to return : 8
Enter the numbers of days you borrowed a book : 10
```

```
-----
This book is successfully returned by you.
-----
```

```
Do you want to return more books ?
If you want then Enter 'Y' for YES and 'N' for NO to return more books : |
```

Figure 55 : Screenshot of successfully book return by user after providing valid book ID.

After giving input for Book ID to return another book, it will ask again for number of days borrowed and after providing valid day then it will provide a message saying "This book is successfully returned by you". Again, loop back to place where program will ask more books to return.

Here, program restrict user from returning same book twice. If user tries same book twice in a particular transaction, then it will provide an error message saying "This book is already returned by you. Please, provide another Book ID to return a book." and program request user for another valid book id as shown in figure below.

```
Do you want to return more books ?  
If you want then Enter 'Y' for YES and 'N' for NO to return more books : y  
  
Enter book ID you want to return : 8  
-----  
This book is already returned by you. Please, provide another Book ID to return a book.  
-----  
Enter book ID you want to return : |
```

Figure 56 : Screenshot of an error message after user try to return same book twice.

After that user provide valid book ID and days to return another book. Now user will not return any book so user will provide 'n' response not to return more books as shown in figure below.

```
Enter book ID you want to return : 1  
Enter the numbers of days you borrowed a book : 11  
-----  
This book is successfully returned by you.  
-----  
Do you want to return more books ?  
If you want then Enter 'Y' for YES and 'N' for NO to return more books : |
```

Figure 57 : Screenshot of successfully return book by user.

After user provide response input as “n” for No, then as above figure a message will appear saying “Updated books list after returning books:” and updated available book’s list details after returning process is displayed in table as shown below in figure.

```
Do you want to return more books ?
If you want then Enter 'Y' for YES and 'N' for NO to return more books : n
```

----- Updated books list in library after returning books : -----								
Book ID		Book Name		Author		Quantity		Price
1		Harry Potter		Jk Rowling		<u>26</u>		\$4
2		Start With Why		Simon Sinek		<u>21</u>		\$3.5
3		Programming		Jhon Smith		<u>13</u>		\$3.5
4		IT Networking		William Shakes		<u>12</u>		\$5
5		The Castle		Firdausi		<u>10</u>		\$4
6		Fear of Flying		Erica Jong		<u>1</u>		\$5.5
7		Python program		bishwas limbu		<u>11</u>		\$3.0
8		Silent Hill		J.K Chettri		<u>11</u>		\$2.0
9		Web Design		J.K Maclean		<u>10</u>		\$3.0
10		Watch Maker		Logan Macron		<u>13</u>		\$3.0

Figure 58 : Screenshot of updated book list after user provide 'n' to program query for more returned books.

After that a costumer return bill is generated in text file note and bill is also displayed in the program with costumer name, book name, date, time, total days borrowed and total fine to be paid for late returned books. Here in the figure book ID 1, 6 and 9 book quantities are increase by 1 and red marking indicating updated value. With the message “Thank you for using our library management system.” Program will loop back to library menu option where user is request to enter value 1,2,3 and 4 as shown in figure below.

Customer Return Books Details						
Customer Name	Book Name	Retrun Date	Retrun Time	Total Borrowed Days		
Bishwas Limbu	Fear of Flying	21-09-09,Thu	17:02	11		
	Silent Hill	21-09-09,Thu	17:02	10		
	Harry Potter	21-09-09,Thu	17:02	11		
					Total Fine	\$0.48
Note : fine is charged 5% of total cost of each book per extra days. Fine will be applicable after only exceeding over 10 days.						
Thank you for using our library management system.						
Please, Enter '1' to borrow a book. Please, Enter '2' to return a book. Please, Enter '3' to add a new book. Please, Enter '4' to exit.						
Please, Enter a value as 1 or 2 or 3 or 4 :						

Figure 59 : Screenshot of costumer return books details bill.

Creation of text file for borrowing book:

Bishwas Limbu2368513R	9/9/2021 6:33 PM	Text Document	2 KB
bookStock list	9/9/2021 6:30 PM	Text Document	1 KB

Figure 60 : Screenshot of text file with unique file name generated.

Costumer return books details note is generated as bill with unique file name.

Customer Return Books Details						
Customer Name	Book Name	Retrun Date	Retrun Time	Total Borrowed Days		
Bishwas Limbu	Fear of Flying	21-09-09,Thu	17:02	11		
	Silent Hill	21-09-09,Thu	17:02	10		
	Harry Potter	21-09-09,Thu	17:02	11		
					Total Fine	\$0.48
Note : fine is charged 5% of total cost of each book per extra days. Fine will be applicable after only exceeding over 10 days.						

Figure 61 : Screenshot of customer return books details in text file.

Above figure is bill note for costumer return books details generated after returning books in text file.

3.4 Adding process

```

Please, Enter '1' to borrow a book.
Please, Enter '2' to return a book.
Please, Enter '3' to add a new book.
Please, Enter '4' to exit.

Please, Enter a value as 1 or 2 or 3 or 4 : 3

-----
You will now add new books to book stock List :

-----  

Book ID | Book Name | Author | Quantity | Price  

-----  

1 | Harry Potter | JK Rowling | 26 | $4  

2 | Start With Why | Simon Sinek | 21 | $3.5  

3 | Programming | Jhon Smith | 13 | $3.5  

4 | IT Networking | William Shakes | 12 | $5  

5 | The Castle | Firdausi | 10 | $4  

6 | Fear of Flying | Erica Jong | 1 | $5.5  

7 | Python program | bishwas limbu | 11 | $3.0  

8 | Silent Hill | J.K Chettri | 11 | $2.0  

9 | Web Design | J.K Macloan | 10 | $3.0  

10 | Watch Maker | Logan Macron | 13 | $3.0
-----  

Please, provide the book name :

```

Figure 62 : Screenshot of user interface for adding books.

After the user provides input value “3”, then adding process is carried out. Above figure, user interface is displayed for adding process where it displays a message “ you will now add new books to book stock list” with all available book’s list details in table form and request user to input book name.

```

-----  

You will now add new books to book stock List :

-----  

Book ID | Book Name | Author | Quantity | Price  

-----  

1 | Harry Potter | JK Rowling | 26 | $4  

2 | Start With Why | Simon Sinek | 21 | $3.5  

3 | Programming | Jhon Smith | 13 | $3.5  

4 | IT Networking | William Shakes | 12 | $5  

5 | The Castle | Firdausi | 10 | $4  

6 | Fear of Flying | Erica Jong | 1 | $5.5  

7 | Python program | bishwas limbu | 11 | $3.0  

8 | Silent Hill | J.K Chettri | 11 | $2.0  

9 | Web Design | J.K Macloan | 10 | $3.0  

10 | Watch Maker | Logan Macron | 13 | $3.0
-----  

Please, provide the book name : Harry Potter  

-----  

This book is already in book Stock list. Please, provide another Book Name to add book.
-----
```

Figure 63 : Screenshot of an error message when user try to add book of same name in book list.

If user provide book name already in the book list, an error message is displayed saying “This book is already in book Stock list. Please, provide another Book Name to add book.” and program request user to provide again book name.

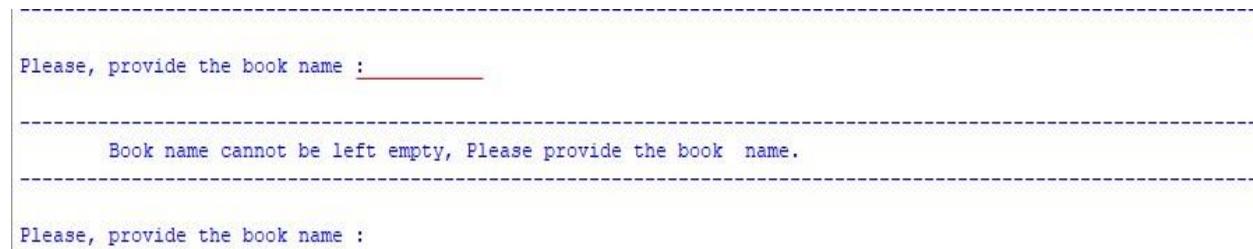


Figure 64 : Screenshot of an error message when user provide no value to book name.

If user hits the “Enter” keyboard without passing value, an error message is displayed saying “Book name cannot be left empty, Please provide the book name.” and program request user to provide book name again.

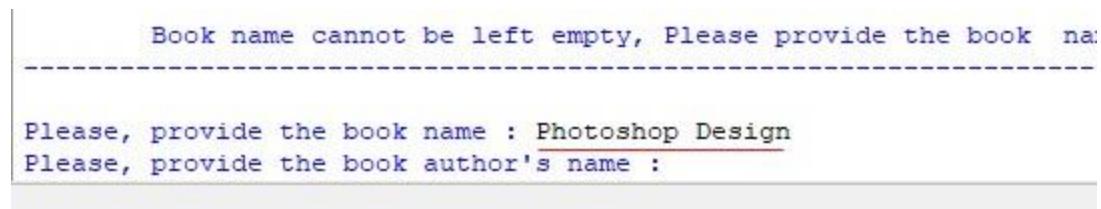


Figure 65 : Screenshot of process after user provide valid book name.

If user provide valid book name then program will request for author name.

```
Please, provide the book author's name : _____  
-----  
Book author's name cannot be left empty, Please provide the book author's name.  
-----  
Please, provide the book author's name : |
```

Figure 66 : Screenshot of an error message after user provide no value to author's name.

If user hits the “Enter” keyboard without passing value, an error message is displayed saying “Book author’s name cannot be left empty, Please provide the book authro’s name.” and program request user to provide book author’s name again.

```
-----  
Please, provide the book author's name : Morgan Marget  
Please, provide the book quantity : |
```

Figure 67 : Screenshot of process after user provide valid author's name.

If user provide valid author’s name then program will request for the quantity of book user want to add.

```
Please, provide the book author's name : Morgan Marget  
Please, provide the book quantity : _____  
-----  
Invalid input!!! Please provide quantity in number.  
-----
```

Figure 68 : Screenshot of an error message when user provide no value to book quantity.

```
Please, provide the book quantity : five  
-----  
Invalid input!!! Please provide quantity in number.  
-----
```

Figure 69 : Screenshot of an error message when user provide string value to book quantity.

If user hits the “Enter” keyboard without passing value or provide stirng value, an error message is displayed saying “Invalid input!!! Please provide quantity in number ” and program request user to provide book quantity again.

```
Please, provide the book quantity : -1
-----
Invalid input!!! Please provide price in valid quantity and positive number.
-----
Please, provide the book quantity : |
```

Figure 70 : Screenshot of an error message when user provide negative value to book quantity.

If user provide negative value to quantity, an error message is displayed saying“ Invalid input!!! Please provide valid quantity and of positive number.” and program request user to provide book quantity again.

```
-----
Please, provide the book quantity : 10
please, provide the book price :
```

Figure 71 : Screenshot of process after user provide valid book quantity.

If user provide valid quantity then program will request for the price of book.

```
Please, provide the book quantity : 10
please, provide the book price :
-----
Invalid input!!! Please provide price in number.
```

Figure 72 : Screenshot of an error message when user provide no value to book price.

```
please, provide the book price : six
-----
Invalid input!!! Please provide price in number.
```

Figure 73 : Screenshot of an error message when user provide string value to book price.

If user hits the “Enter” keyboard without passing value or provide stirng value, an error message is displayed saying “Invalid input!!! Please provide price in number ” and program request user to provide book price again.

```
please, provide the book price : -1
-----
Invalid input!!! Please provide valid price and of positive number.
```

```
please, provide the book price :
```

Figure 74 : Screenshot of an error message when user provide negative value to book price.

If user provide negative value to quantity, an error message is displayed saying“ Invalid input!!! Please provide valid price and of positive number.” and program request user to provide book price.again.

```
please, provide the book price : 5
-----
This book is successfully added to book list.
```

```
Do you want to add more books ?
Please, provide 'Y' for Yes and 'N' for No to add more books : |
```

Figure 75 : Screenshot of successfully adding of book after user provide valid price.

If user provide valid price then program is display a message saying “This book is successfully added to book list” and ask if user want to add more books.

```
Do you want to add more books ?  
Please, provide 'Y' for Yes and 'N' for No to add more books : _____
```

```
-----  
Invalid input!!! please, Provide 'Y' for Yes and 'No' for No.  
-----
```

Figure 76 : Screenshot of an error message when user provide no value to program query to add more book.

```
Do you want to add more books ?  
Please, provide 'Y' for Yes and 'N' for No to add more books : yes _____
```

```
-----  
Invalid input!!! please, Provide 'Y' for Yes and 'No' for No.  
-----
```

```
Do you want to add more books ?  
Please, provide 'Y' for Yes and 'N' for No to add more books : |
```

Figure 77 : Screenshot of an error message when user provide other string value to program query to add more book.

If user hits the “Enter” keyboard without passing value or input any value beside ‘y’ or ‘n’, an error message is displayed saying “Invalid input!!! Please provide price in ‘Y’ for Yes and ‘N’ for No.” and program request user to provide response again to add more books.

```
Do you want to add more books ?
Please, provide 'Y' for Yes and 'N' for No to add more books : y
```

```
Please, provide the book name : Killer Forgotten
Please, provide the book author's name : Macron Miller
Please, provide the book quantity : 19
please, provide the book price : 3
```

```
-----  
This book is successfully added to book list.  
-----
```

```
Do you want to add more books ?
Please, provide 'Y' for Yes and 'N' for No to add more books :
```

Figure 78 : Screenshot after user provide valid response to program query and adding another book successfully.

If user provide response input as 'y' then adding another new book will be carried just like in above figure and program request user to provide response again to add more books. This process will be repeated again again until user provide input response "n".

```
Do you want to add more books ?
Please, provide 'Y' for Yes and 'N' for No to add more books : n
```

```
-----  
Updated books list in library after adding books :
```

Book ID		Book Name		Author		Quantity		Price
1		Harry Potter		Jk Rowling		26		\$4
2		Start With Why		Simon Sinek		21		\$3.5
3		Programming		Jhon Smith		13		\$3.5
4		IT Networking		William Shakes		12		\$5
5		The Castle		Firdausi		10		\$4
6		Fear of Flying		Erica Jong		1		\$5.5
7		Python program		bishwas limbu		11		\$3.0
8		Silent Hill		J.K Chettri		11		\$2.0
9		Web Design		J.K Macloan		10		\$3.0
10		Watch Maker		Logan Macron		13		\$3.0
11		Photoshop Design		Morgan Marget		10		\$5.0
12		Killer Forgotten		Macron Miller		19		\$3.0

```
Please, Enter '1' to borrow a book.
Please, Enter '2' to return a book.
Please, Enter '3' to add a new book.
Please, Enter '4' to exit.
```

```
Please, Enter a value as 1 or 2 or 3 or 4 : |
```

Figure 79 : Screenshot after user provide 'n' to program query to add more books and update book list after adding books.

After user provide input response ‘n’ then a message is display saying “Updated book list in library after adding books” and all available books list with newly added books are also display in table form. After that Program will loop back to library menu option where user is request to enter value 1,2,3 and 4 as shown in figure above.

3.4 Termination Process

```

----- Updated books list in library after adding books : -----
-----+-----+-----+-----+-----+-----+
Book ID | Book Name | Author | Quantity | Price
-----+-----+-----+-----+-----+
1       | Harry Potter | Jk Rowling | 26 | $4
2       | Start With Why | Simon Sinek | 21 | $3.5
3       | Programming | Jhon Smith | 13 | $3.5
4       | IT Networking | William Shakes | 12 | $5
5       | The Castle | Firdausi | 10 | $4
6       | Fear of Flying | Erica Jong | 1 | $5.5
7       | Python program | bishwas limbu | 11 | $3.0
8       | Silent Hill | J.K Chettri | 11 | $2.0
9       | Web Design | J.K Macloan | 10 | $3.0
10      | Watch Maker | Logan Macron | 13 | $3.0
11      | Photoshop Design | Morgan Marget | 10 | $5.0
12      | Killer Forgotten | Macron Miller | 19 | $3.0
-----+-----+-----+-----+-----+
Please, Enter '1' to borrow a book.
Please, Enter '2' to return a book.
Please, Enter '3' to add a new book.
Please, Enter '4' to exit.

Please, Enter a value as 1 or 2 or 3 or 4 : 4
*****
Thank you for using our library management system.
*****
>>> |
-----
```

Here in this figure if user input value is 4 then program displays a message saying “Thank you for using our library management system.” and exits the program.

4. Testing

4.1. Test 1: Showing implementation of try and except

Objective	To show implementation of try and except.
Action	<p>Screenshots were taken from the initial program running phase.</p> <ul style="list-style-type: none"> ➤ To return the book Input value “2” was given in menu option ➤ ‘Bishwas Limbu’ was used for returner’s name input. ➤ Book ID “6” given to return the book. ➤ String value ‘ten’ was input instead of integer value in the totalDays_Borrowed (where try and except is implemented).
Expected Result	An error message “Invalid input!!! Please, provide days in number” will display.
Actual Result	An error message “Invalid input!!! Please, provide days in number” was displayed.
Conclusion	The test was successful.

Table 1 : To show implementation of try and except

Welcome to College Library Management System.								
Book ID		Book Name		Author		Quantity		Price
1		Harry Potter		Jk Rowling		26		\$4
2		Start With Why		Simon Sinek		21		\$3.5
3		Programming		Jhon Smith		13		\$3.5
4		IT Networking		William Shakes		12		\$5
5		The Castle		Firdausi		10		\$4
6		Fear of Flying		Erica Jong		1		\$5.5
7		Python program		bishwas limbu		11		\$3.0

Figure 80 : Screenshot of main library management system user interface.

Please, Enter a value as 1 or 2 or 3 or 4 : 2

You will now return the book :

Book ID		Book Name		Author		Quantity		Price
1		Harry Potter		Jk Rowling		26		\$4
2		Start With Why		Simon Sinek		21		\$3.5
3		Programming		Jhon Smith		13		\$3.5
4		IT Networking		William Shakes		12		\$5
5		The Castle		Firdausi		10		\$4
6		Fear of Flying		Erica Jong		1		\$5.5
7		Python program		bishwas limbu		11		\$3.0

Please, Enter the returner's name : Bishwas Limbu

Figure 81 : Screenshot of user interface after user input value 2 and user providing returner's name.

Book ID		Book Name		Author		Quantity		Price
1		Harry Potter		Jk Rowling		26		\$4
2		Start With Why		Simon Sinek		21		\$3.5
3		Programming		Jhon Smith		13		\$3.5
4		IT Networking		William Shakes		12		\$5
5		The Castle		Firdausi		10		\$4
6		Fear of Flying		Erica Jong		1		\$5.5
7		Python program		bishwas limbu		11		\$3.0

Please, Enter the returner's name : Bishwas Limbu

Enter book ID you want to return : 6

Figure 82 : Screenshot of user providing input for returner's name and book ID to return book.

```

-----|-----|-----|-----|-----|-----|-----|
| You will now return the book : |-----|-----|-----|-----|-----|-----|
-----|-----|-----|-----|-----|-----|-----|
| Book ID | Book Name | Author | Quantity | Price |
-----|-----|-----|-----|-----|-----|-----|
| 1 | Harry Potter | Jk Rowling | 26 | $4 |
| 2 | Start With Why | Simon Sinek | 21 | $3.5 |
| 3 | Programming | Jhon Smith | 13 | $3.5 |
| 4 | IT Networking | William Shakes | 12 | $5 |
| 5 | The Castle | Firdausi | 10 | $4 |
| 6 | Fear of Flying | Erica Jong | 1 | $5.5 |
| 7 | Python program | bishwas limbu | 11 | $3.0 |
-----|-----|-----|-----|-----|-----|-----|
-----|-----|-----|-----|-----|-----|-----|
| Please, Enter the returner's name : Bishwas Limbu |-----|-----|-----|-----|-----|-----|
| Enter book ID you want to return : 6 |-----|-----|-----|-----|-----|-----|
| Enter the numbers of days you borrowed a book : ten |-----|-----|-----|-----|-----|-----|
-----|-----|-----|-----|-----|-----|-----|
| Invalid input!!! Please, provide days in number. |-----|-----|-----|-----|-----|-----|
-----|-----|-----|-----|-----|-----|-----|

```

Figure 83 : Screenshot of an error message when user provide string value instead of integer value.

```

-----|-----|-----|-----|-----|-----|
| Please, Enter the returner's name : Bishwas Limbu |-----|-----|-----|-----|-----|-----|
| Enter book ID you want to return : 6 |-----|-----|-----|-----|-----|-----|
| Enter the numbers of days you borrowed a book : ten |-----|-----|-----|-----|-----|-----|
-----|-----|-----|-----|-----|-----|-----|
| Invalid input!!! Please, provide days in number. |-----|-----|-----|-----|-----|-----|
-----|-----|-----|-----|-----|-----|-----|
| Enter the numbers of days you borrowed a book : 11 |-----|-----|-----|-----|-----|-----|
-----|-----|-----|-----|-----|-----|-----|
| This book is successfully returned by you. |-----|-----|-----|-----|-----|-----|
-----|-----|-----|-----|-----|-----|-----|

```

Figure 84 : Screenshot of successfully returned book when user provide positive integer value.

4.2. Test 2: For selecting borrow and return option

Objective	To demonstrate selection of borrow and return option.
Action	program was initialized and user input was given to borrow or return and, screenshots were taken when: <ul style="list-style-type: none"> ➤ “-2” Negative value was provided as a user input ➤ “two” and “7” Non-existed values (i.e., neither 1, 2, 3 or 4) was provided as input.
Expected Result	An error message “Invalid input!!! Please, provide input value as 1, 2,3 or 4.” will display.
Actual Result	An error message “Invalid input!!! Please, provide input value as 1, 2,3 or 4.” was displayed
Conclusion	The test was successful.

Table 2 : To demonstrate selection of borrow and return option.

```
*****
***** Welcome to College Library Management System. *****
*****



-----|-----|-----|-----|-----|-----|-----|
Book ID | Book Name | Author | Quantity | Price |
-----|-----|-----|-----|-----|-----|-----|
1       | Harry Potter | Jk Rowling | 26 | $4
2       | Start With Why | Simon Sinek | 21 | $3.5
3       | Programming | Jhon Smith | 13 | $3.5
4       | IT Networking | William Shakes | 12 | $5
5       | The Castle | Firdausi | 10 | $4
6       | Fear of Flying | Erica Jong | 3 | $5.5
7       | Python program | bishwas limbu | 11 | $3.0
-----|-----|-----|-----|-----|-----|-----|



Please, Enter '1' to borrow a book.
Please, Enter '2' to return a book.
Please, Enter '3' to add a new book.
Please, Enter '4' to exit.

Please, Enter a value as 1 or 2 or 3 or 4 : -2

-----|-----|-----|-----|-----|-----|-----|
                                         Invalid input!!! Please, provide input value as 1, 2,3 or 4.
-----|-----|-----|-----|-----|-----|-----|
```

Figure 85 : Screenshot of an error message after user provide negative value as input.

```

Please, Enter '1' to borrow a book.
Please, Enter '2' to return a book.
Please, Enter '3' to add a new book.
Please, Enter '4' to exit.

Please, Enter a value as 1 or 2 or 3 or 4 : two

-----
Invalid input!!! Please, provide input value as 1, 2,3 or 4.
-----
```

Figure 86 : Screenshot of an error message after user provide other string value beside 1,2,3,4 value.

```

Please, Enter '1' to borrow a book.
Please, Enter '2' to return a book.
Please, Enter '3' to add a new book.
Please, Enter '4' to exit.

Please, Enter a value as 1 or 2 or 3 or 4 : 7

-----
Invalid input!!! Please, provide input value as 1, 2,3 or 4.
-----
```

Figure 87 : Screenshot of an error message after user provide non-existed value.

```

Please, Enter '1' to borrow a book.
Please, Enter '2' to return a book.
Please, Enter '3' to add a new book.
Please, Enter '4' to exit.

Please, Enter a value as 1 or 2 or 3 or 4 : 1

-----
You will now borrow the book :

-----
```

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	26	\$4
2	Start With Why	Simon Sinek	21	\$3.5
3	Programming	Jhon Smith	13	\$3.5
4	IT Networking	William Shakes	12	\$5
5	The Castle	Firdausi	10	\$4
6	Fear of Flying	Erica Jong	3	\$5.5
7	Python program	bishwas limbu	11	\$3.0

```

Please, Enter the borrower's name : |
```

Figure 88 : Screenshot of user interface for borrowing book after user provide value 1.

4.3. Test 3: For file generation of borrow

Objective	To show file generation of borrow.
Action	<p>The program was initialized and</p> <ul style="list-style-type: none"> ➤ '1' value input was given in order to borrow book. ➤ 'Bishwas Limbu' was given as input for borrower name. ➤ '2' was given as input for bookID to borrow that book. ➤ 'y' value was given to borrow another book. ➤ '3' was given as input for bookID to borrow that book. ➤ 'n' was input to stop the borrow process and generate the text file.
Expected Result	A costumer borrow detail text file will generate.
Actual Result	A costumer borrow detail text file was generated.
Conclusion	The test was successful.

Table 3 : To show file generation of borrow.

Welcome to College Library Management System.						
Book ID	Book Name	Author	Quantity	Price		
1	Harry Potter	Jk Rowling	16	\$4		
2	Start With Why	Simon Sinek	16	\$3.5		
3	Programming	Jhon Smith	10	\$3.5		
4	IT Networking	William Shakes	13	\$5		
5	The Castle	Firdeausi	11	\$4		
6	Fear of Flying	Erica Jong	7	\$5.5		
7	Python program	bishwas limbu	14	\$3.0		

Figure 89 : Screenshot of user providing 1 value as input.

```
Please, Enter a value as 1 or 2 or 3 or 4 : 1

-----
You will now borrow the book :

-----
```

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	16	\$4
2	Start With Why	Simon Sinek	16	\$3.5
3	Programming	Jhon Smith	10	\$3.5
4	IT Networking	William Shakes	13	\$5
5	The Castle	Firdausi	11	\$4
6	Fear of Flying	Erica Jong	7	\$5.5
7	Python program	bishwas limbu	14	\$3.0

```
Please, Enter the borrower's name : Bishwas Limbu
```

Figure 90 : Screenshot of user providing borrower's name.

```
Please, Enter a value as 1 or 2 or 3 or 4 : 1

-----
You will now borrow the book :

-----
```

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	16	\$4
2	Start With Why	Simon Sinek	16	\$3.5
3	Programming	Jhon Smith	10	\$3.5
4	IT Networking	William Shakes	13	\$5
5	The Castle	Firdausi	11	\$4
6	Fear of Flying	Erica Jong	7	\$5.5
7	Python program	bishwas limbu	14	\$3.0

```
Please, Enter the borrower's name : Bishwas Limbu
Enter book ID you want to borrow : 2

*****
This Book is available for borrow.
*****
-----
```

Book Name	Price	Date of Borrow	Time of Borrow
Start With Why	\$3.5	21-09-10,Fri	01:07

Figure 91 : Screenshot of user providing book Id to borrow book.

```

Please, Enter the borrower's name : Bishwas Limbu
Enter book ID you want to borrow : 2

*****
This Book is available for borrow.
*****
Book Name | Price | Date of Borrow | Time of Borrow
-----
Start With Why | $3.5 | 21-09-10,Fri | 01:07

Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books : y

Enter book ID you want to borrow : |

```

Figure 92 : Screenshot of user providing 'y' value to program query to borrow more books.

```

Start With Why | $3.5 | 21-09-10,Fri | 01:07

Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books : y

Enter book ID you want to borrow : 3

*****
This Book is available for borrow.
*****
Book Name | Price | Date of Borrow | Time of Borrow
-----
Programming | $3.5 | 21-09-10,Fri | 01:08

Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books : |

```

Figure 93 : Screenshot of user providing book Id to borrow book.

Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books : n

Updated books list in library after borrowing books :								
Book ID		Book Name		Author		Quantity		Price
1		Harry Potter		JK Rowling		16		\$4
2		Start With Why		Simon Sinek		15		\$3.5
3		Programming		Jhon Smith		9		\$3.5
4		IT Networking		William Shakes		13		\$5
5		The Castle		Firdausi		11		\$4
6		Fear of Flying		Erica Jong		7		\$5.5
7		Python program		bishwas limbu		14		\$3.0

Custumer Borrow Books Details								
Costumer Name		Book Name		Borrowed Date		Borrowed Time		Price
Bishwas Limbu		Start With Why		21-09-10,Fri		01:09		\$3.5
		Programming		21-09-10,Fri		01:09		\$3.5

Total Amount \$7.0						
----------------------	--	--	--	--	--	--

Thank you for using our library management system.						
--	--	--	--	--	--	--

Figure 94 : Screenshot of user providing 'n' value to program query to borrow more book.

Bishwas Limbu91693698 - Notepad								
File Edit Format View Help								
Custumer Borrow Books Details								
Costumer Name Book Name Borrowed Date Borrowed Time Price								
Bishwas Limbu		Start With Why		21-09-10,Fri		01:09		\$3.5
		Programming		21-09-10,Fri		01:09		\$3.5

Total Amount \$7.0						
----------------------	--	--	--	--	--	--

Figure 95 : Screenshot of costumer borrow book details in text file.

4.4. Test 4: For file generation of return

Objective	To show file generation of return.
Action	<p>The program was initialized and</p> <ul style="list-style-type: none"> ➤ '2' value input was given in order to return book. ➤ 'Shyam Limbu' was given as input for returner's name. ➤ '7' was given as input for bookID to return that book. ➤ '11' was given as input for total borrowing days ➤ 'y' value was given to return another book. ➤ '6' was given as input for bookID to return that book. ➤ '12' was given as input for total borrowing days ➤ 'n' was input to stop the return process and generate the text file.
Expected Result	A costumer return detail text file will generate.
Actual Result	A costumer return detail text file was generated.
Conclusion	The test was successful.

Table 4 : To show file generation of return.

```
*****
***** Welcome to College Library Management System. *****
*****



-----|-----|-----|-----|-----|-----|-----|
Book ID | Book Name | Author | Quantity | Price |
-----|-----|-----|-----|-----|-----|-----|
1 | Harry Potter | Jk Rowling | 16 | $4 |
2 | Start With Why | Simon Sinek | 15 | $3.5 |
3 | Programming | Jhon Smith | 9 | $3.5 |
4 | IT Networking | William Shakes | 13 | $5 |
5 | The Castle | Firdausi | 11 | $4 |
6 | Fear of Flying | Erica Jong | 7 | $5.5 |
7 | Python program | bishwas limbu | 14 | $3.0 |
-----|-----|-----|-----|-----|-----|-----|



Please, Enter '1' to borrow a book.
Please, Enter '2' to return a book.
Please, Enter '3' to add a new book.
Please, Enter '4' to exit.

Please, Enter a value as 1 or 2 or 3 or 4 : 2
```

Figure 96 : Screenshot of user providing "2" value as input. For return process

```
Please, Enter a value as 1 or 2 or 3 or 4 : 2
```

```
You will now return the book :
```

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	16	\$4
2	Start With Why	Simon Sinek	15	\$3.5
3	Programming	Jhon Smith	9	\$3.5
4	IT Networking	William Shakes	13	\$5
5	The Castle	Firdausi	11	\$4
6	Fear of Flying	Erica Jong	7	\$5.5
7	Python program	bishwas limbu	14	\$3.0

```
Please, Enter the returner's name : Shyam Limbu
```

Figure 97 : Screenshot of user providing returner's name.

```
Please, Enter a value as 1 or 2 or 3 or 4 : 2
```

```
You will now return the book :
```

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	16	\$4
2	Start With Why	Simon Sinek	15	\$3.5
3	Programming	Jhon Smith	9	\$3.5
4	IT Networking	William Shakes	13	\$5
5	The Castle	Firdausi	11	\$4
6	Fear of Flying	Erica Jong	7	\$5.5
7	Python program	bishwas limbu	14	\$3.0

```
Please, Enter the returner's name : Shyam Limbu
Enter book ID you want to return : 7
```

Figure 98 : Screenshot of user providing book ID "7" to return book.

```
Please, Enter a value as 1 or 2 or 3 or 4 : 2
```

```
You will now return the book :
```

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	16	\$4
2	Start With Why	Simon Sinek	15	\$3.5
3	Programming	Jhon Smith	9	\$3.5
4	IT Networking	William Shakes	13	\$5
5	The Castle	Firdausi	11	\$4
6	Fear of Flying	Erica Jong	7	\$5.5
7	Python program	bishwas limbu	14	\$3.0

```
Please, Enter the returner's name : Shyam Limbu
Enter book ID you want to return : 7
Enter the numbers of days you borrowed a book : 11
```

```
This book is successfully returned by you.
```

Figure 99 : Screenshot of user providing number of days borrowed and successfully returning book.

```
Please, Enter the returner's name : Shyam Limbu
Enter book ID you want to return : 7
Enter the numbers of days you borrowed a book : 11

-----
This book is successfully returned by you.

-----
Do you want to return more books ?
If you want then Enter 'Y' for YES and 'N' for NO to return more books : y

Enter book ID you want to return :
```

Figure 100 : Screenshot of user providing 'y' value to return more books.

```
-----
Do you want to return more books ?
If you want then Enter 'Y' for YES and 'N' for NO to return more books : y

Enter book ID you want to return : 4
Enter the numbers of days you borrowed a book : |
```

Figure 101 : Screenshot of user providing book ID to return another book.

```
Do you want to return more books ?
If you want then Enter 'Y' for YES and 'N' for NO to return more books : y
```

```
Enter book ID you want to return : 4
Enter the numbers of days you borrowed a book : 12
```

```
-----
This book is successfully returned by you.
```

```
Do you want to return more books ?
If you want then Enter 'Y' for YES and 'N' for NO to return more books : |
```

Figure 102 : Screenshot of user providing number of days borrowed and successfully returning book.

Do you want to return more books ?
If you want then Enter 'Y' for YES and 'N' for NO to return more books : n

Updated books list in library after returning books :

Book ID		Book Name		Author		Quantity		Price
1		Harry Potter		Jk Rowling		16		\$4
2		Start With Why		Simon Sinek		15		\$3.5
3		Programming		Jhon Smith		9		\$3.5
4		IT Networking		William Shakes		14		\$5
5		The Castle		Firdausi		11		\$4
6		Fear of Flying		Erica Jong		7		\$5.5
7		Python program		bishwas limbu		15		\$3.0

Customer Return Books Details

Costumer Name		Book Name		Retrun Date		Retrun Time		Total Borrowed Days
Shyam Limbu		Python program		21-09-10,Fri		01:16		11
		IT Networking		21-09-10,Fri		01:16		12

Total Fine \$ 65

Note : fine is charged 5% of total cost of each book per extra days.
Fine will be applicable after only exceeding over 10 days.

Figure 103 : Screenshot of after user providing 'n' value and updated book list and costumer return bill.

Shyam Limbu16317706R - Notepad

Customer Return Books Details

Costumer Name	Book Name	Retrun Date	Retrun Time	Total Borrowed Days
Shyam Limbu	Python program	21-09-10,Fri	01:16	11
	IT Networking	21-09-10,Fri	01:16	12
			Total Fine	\$0.65

Note : fine is charged 5% of total cost of each book per extra days.
Fine will be applicable after only exceeding over 10 days.

Figure 104 : Screenshot of costumer return books details in text file.

4.5. Test 5: For showing the update in stock

Objective	To show the update in stock
Action	<ul style="list-style-type: none"> ➤ Screenshot of bookStock_list.txt file was taken. ➤ The program was initialized ➤ '1' value input was given in order to borrow book. ➤ 'Hari Rai' was given as input for borrower's name. ➤ '4' was given as input for bookID to borrow that book. ➤ 'n' was input to stop the borrow process ➤ comparing quantity of books after borrowing book in table of terminal and text file. ➤ '2' value input was given in order to return book. ➤ 'Sujan Lama' was given as input for returner's name. ➤ '1' was given as input for bookID to return that book. ➤ '13' was given as input for total borrowing days ➤ 'n' was input to stop the return process ➤ comparing quantity of books after returning book in table of terminal and text file.
Expected Result	<ol style="list-style-type: none"> The quantity of the borrowed book will decrease. The quantity of the returned book will increase.
Actual Result	<ol style="list-style-type: none"> The quantity of the borrowed book was decreased. The quantity of the returned book was increased.
Conclusion	The test was successful.

Table 5 : To show the update in stock.

bookStock_list - Notepad

File Edit Format View Help

Harry Potter,Jk Rowling,16,\$4
 Start With Why,Simon Sinek,15,\$3.5
 Programming,Jhon Smith,9,\$3.5
 IT Networking,William Shakes,14,\$5
 The Castle,Firdausi,11,\$4
 Fear of Flying,Erica Jong,7,\$5.5
 Python program,bishwas limbu,15,\$3.0

Figure 105 : Screenshot of bookStock_list.txt file in initial phase.

```
*****
***** Welcome to College Library Management System. *****
*****



Book ID | Book Name | Author | Quantity | Price
-----|-----|-----|-----|-----|
1 | Harry Potter | Jk Rowling | 17 | $4
2 | Start With Why | Simon Sinek | 15 | $3.5
3 | Programming | Jhon Smith | 9 | $3.5
4 | IT Networking | William Shakes | 14 | $5
5 | The Castle | Firdausi | 11 | $4
6 | Fear of Flying | Erica Jong | 7 | $5.5
7 | Python program | bishwas limbu | 15 | $3.0
-----|-----|-----|-----|-----|


| Please, Enter '1' to borrow a book.
| Please, Enter '2' to return a book.
| Please, Enter '3' to add a new book.
| Please, Enter '4' to exit.

Please, Enter a value as 1 or 2 or 3 or 4 : 1
```

Figure 106 : Screenshot of user providing value "1" to library menu option.

Please, Enter a value as 1 or 2 or 3 or 4 : 1

You will now borrow the book :

Book ID		Book Name		Author		Quantity		Price
1		Harry Potter		Jk Rowling		17		\$4
2		Start With Why		Simon Sinek		15		\$3.5
3		Programming		Jhon Smith		9		\$3.5
4		IT Networking		William Shakes		14		\$5
5		The Castle		Firdausi		11		\$4
6		Fear of Flying		Erica Jong		7		\$5.5
7		Python program		bishwas limbu		15		\$3.0

Please, Enter the borrower's name : Hari Rai

Figure 107 : Screenshot of user providing borrower's name.

Please, Enter a value as 1 or 2 or 3 or 4 : 1

You will now borrow the book :

Book ID		Book Name		Author		Quantity		Price
1		Harry Potter		Jk Rowling		17		\$4
2		Start With Why		Simon Sinek		15		\$3.5
3		Programming		Jhon Smith		9		\$3.5
4		IT Networking		William Shakes		14		\$5
5		The Castle		Firdausi		11		\$4
6		Fear of Flying		Erica Jong		7		\$5.5
7		Python program		bishwas limbu		15		\$3.0

Please, Enter the borrower's name : Hari Rai

Enter book ID you want to borrow : 4

Figure 108 : Screenshot of user providing book ID "4".

Please, Enter the borrower's name : Hari Rai
Enter book ID you want to borrow : 4

***** This Book is available for borrow. *****

Book Name		Price		Date of Borrow		Time of Borrow
IT Networking		\$5		21-09-10,Fri		01:49

Do you want to borrow more books ?

Figure 109 : Screenshot of borrowed book details after successfully borrowing book.

Do you want to borrow more books ?
If you want then Please, Enter 'Y' for YES and 'N' for NO to borrow more books : n

Updated books list in library after borrowing books :					
Book ID	Book Name	Author	Quantity	Price	
1	Harry Potter	Jk Rowling	17	\$4	
2	Start With Why	Simon Sinek	15	\$3.5	
3	Programming	Jhon Smith	9	\$3.5	
4	IT Networking	William Shakes	13	\$5	
5	The Castle	Firdausi	11	\$4	
6	Fear of Flying	Erica Jong	7	\$5.5	
7	Python program	bishwas limbu	15	\$3.0	

Custumer Borrow Books Details					
Costumer Name	Book Name	Borrowed Date	Borrowed Time	Price	
Hari Rai	IT Networking	21-09-10,Fri	01:52	\$5	
Total Amount					\$5.0

Thank you for using our library management system.					
--	--	--	--	--	--

Figure 110 : Screenshot of updates book stock list in table and costumer borrow details bill.

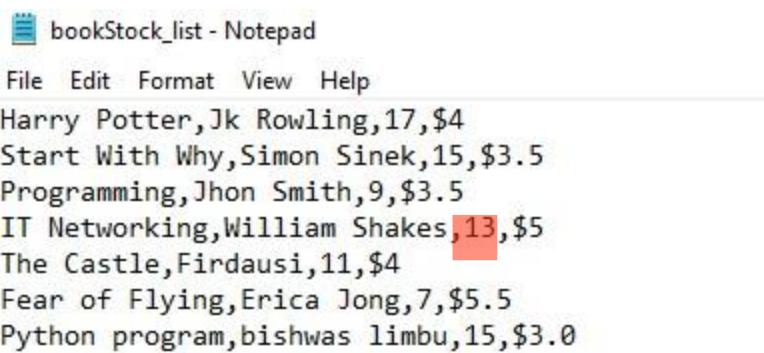


Figure 111 : Screenshot of updated stock list in bookStock_list.txt file.

```

Please, Enter '1' to borrow a book.
Please, Enter '2' to return a book.
Please, Enter '3' to add a new book.
Please, Enter '4' to exit.

Please, Enter a value as 1 or 2 or 3 or 4 : 2

-----
You will now return the book :

-----
```

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	17	\$4
2	Start With Why	Simon Sinek	15	\$3.5
3	Programming	Jhon Smith	9	\$3.5
4	IT Networking	William Shakes	13	\$5
5	The Castle	Firdausi	11	\$4
6	Fear of Flying	Erica Jong	7	\$5.5
7	Python program	bishwas limbu	15	\$3.0

Figure 112 : Screenshot of user providing value "2" for returning process.

```

Please, Enter a value as 1 or 2 or 3 or 4 : 2

-----
You will now return the book :

-----
```

Book ID	Book Name	Author	Quantity	Price
1	Harry Potter	Jk Rowling	17	\$4
2	Start With Why	Simon Sinek	15	\$3.5
3	Programming	Jhon Smith	9	\$3.5
4	IT Networking	William Shakes	13	\$5
5	The Castle	Firdausi	11	\$4
6	Fear of Flying	Erica Jong	7	\$5.5
7	Python program	bishwas limbu	15	\$3.0

```

Please, Enter the returner's name : Sujan Lama
```

Figure 113 : Screenshot of user providing returner's name.

```

Please, Enter a value as 1 or 2 or 3 or 4 : 2

-----
You will now return the book :

-----
Book ID | Book Name | Author | Quantity | Price
-----
1      | Harry Potter | Jk Rowling | 17 | $4
2      | Start With Why | Simon Sinek | 15 | $3.5
3      | Programming | Jhon Smith | 9 | $3.5
4      | IT Networking | William Shakes | 13 | $5
5      | The Castle | Firdausi | 11 | $4
6      | Fear of Flying | Erica Jong | 7 | $5.5
7      | Python program | bishwas limbu | 15 | $3.0
-----
Please, Enter the returner's name : Sujan Lama
Enter book ID you want to return : 1

```

Figure 114 : Screenshot of user providing book ID "1" to return book.

```

-----
Please, Enter the returner's name : Sujan Lama
Enter book ID you want to return : 1
Enter the numbers of days you borrowed a book : 13

-----
This book is successfully returned by you.
-----
```

Figure 115 : Screenshot of user providing number of days borrowed and successfully returning book.

Do you want to return more books ? If you want then Enter 'Y' for YES and 'N' for NO to return more books : <u>n</u>						
----- Updated books list in library after returning books : -----						

Book ID		Book Name		Author		Quantity
1		Harry Potter		Jk Rowling		18
2		Start With Why		Simon Sinek		15
3		Programming		Jhon Smith		9
4		IT Networking		William Shakes		13
5		The Castle		Firdausi		11
6		Fear of Flying		Erica Jong		7
7		Python program		bishwas limbu		15

***** Custumer Return Books Details *****						

Costumer Name		Book Name		Retrun Date		Total Borrowed Days
Sujan Lama		Harry Potter		21-09-10,Fri		13

Total Fine \$0.6						
Note : fine is charged 5% of total cost of each book per extra days. Fine will be applicable after only exceeding over 10 days. *****						

Figure 116 :: Screenshot of updates book stock list in table and costumer return details bill.

bookStock_list - Notepad

File Edit Format View Help

```
Harry Potter,Jk Rowling,18,$4
Start With Why,Simon Sinek,15,$3.5
Programming,Jhon Smith,9,$3.5
IT Networking,William Shakes,13,$5
The Castle,Firdausi,11,$4
Fear of Flying,Erica Jong,7,$5.5
Python program,bishwas limbu,15,$3.0
```

Figure 117 : Screenshot of updated stock list in bookStock_list.txt file.

5. Conclusion

Here, in this coursework I got chance of learning more about python programming was very fruitful to me. After completing this module, I have got significant improvement in coding with python programming language, better understanding of modular approach for coding a application which will help to debug errors easily. I got to learn about designing flowchart and algorithm for any application and how they help to provide clear picture of next step to be taken. Different primitive data structure like int, float, string and non-primitive data structure like list, tuples. Dictionary, sets, files, I learn their uses and operation. Specially by completing this coursework, I got better experience of using file data structure for writing and reading data from the text file. Better knowledge of how to manipulate the keys and values in dictionary and accessing each element in it. I also learn to handle exception error that crashes application if not handle properly. With Try except block, I got experience to handle such error and other exception for string I learn that can be handle by using if else condition.

Here, while doing this coursework various errors like logical and syntax I came across. I was having in logical errors problem most especially while coding for borrow quantity up date. Every time I borrow a book, quantity of all the book was also decrease as I keep borrowing multiple books. I was having confusion of developing pseudocode for the program coding. Develop flow chart was also a big difficulty I have to face because I was not able to implement while loop and for loop condition in to a flow chart.

Despite of such difficulties I completed by coursework on time. This was only possible with extensive hard work, time dedication and researching habit. I research different web site like w3schools, you tube videos for python programming and reference book for this module. I solved the logical error problem that occur during borrow quantity up date by using if condition that will check last index of list. For developing algorithm, pseudocode and flow chart I took some suggestion from my module teacher, take reference from lecture note and look in to videos in YouTube related to them. I also some problem related to date time module where date and time change only after exiting program and re running that program by directly initializing datetime module

object in to variable (i.e., date_Borrow = datetime.datetime.now().strftime("%y-%m-%d,%a").

Appendix

1. main.py

```
import function_List
function_List.main_Function()
```

2. function_List.py

```
#importing sys, datetime module from python
import sys
import datetime #improtng datetime module

#-----main function of program-----
-----

def main_Function():
    """displays book list table and keep asking for integer value 1,2,3 from user
    unless and until valid value is given """

    print("*****")
    print("\t\t\t\tWelcome to College Library Management System. ")

    print("*****\n")
    dict_display()
    continue_1 = True
    while continue_1 == True:#creating a loop to keep asking a value to user if
    invalid input is given
        print("\nPlease, Enter '1' to borrow a book.")
        print("Please, Enter '2' to return a book.")
        print("Please, Enter '3' to add a new book.")
        print("Please, Enter '4' to exit.")
        user_Input = input("\nPlease, Enter a value as 1 or 2 or 3 or 4 : ")#asking
        value 1,2,3 from user which is string datatype
        if user_Input == str(1):#converting 1 interger to string and comparing wth
        user input value
            print("\n\n-----")
            print("You will now borrow the book :")
            print("-----")
```

```

borrow_Function()#borrow_Function method will be invoked if user input
value is 1
elif user_Input == str(2):
    print("\n\n-----")
    print("You will now return the book :")
    print("-----")
return_Function()#return_Function method will be invoked if user input
value is 2

elif user_Input == str(3):
    print("\n\n-----")
    print("You will now add new books to book stock List :")
    print("-----")
    dict_display()
add_Books()#add_Books method will be invoked if user input value is 3

elif user_Input == str(4):

print("\n*****")
print("\t\t\tThank you for using our library management system.")

print("*****")
break#exits the program and loop
else:
    print("\n\n-----")
    print("\t\t\tInvalid input!!! Please, provide input value as 1, 2,3 or 4.")
    print("-----")

#-----making dictionay data collection form txt file-----


def dict_Function():
    """ creates dictionary collection data type from bookStock txt file"""

    file = open("bookStock_list.txt","r")#opening txt file in read mode
    dictionary = {}#creating empty dictionary
    book_ID = 0

```

```

for line in file:#iterates all line as string from txt file one line by line
    book_ID += 1#increment by 1 each time
    line = line.replace("\n","").split(',')#replace next line with empty space in
    string word and split the string word on the basis of commas
    dictionary[str(book_ID)] = line # adding book ID as key and value list of
    sting as to dictionary
    file.close()#closing file
    return dictionary

```

#-----making list of book id-----

```

def allKey_List():
    '''creates a list collection data type which stores all book ID as keys of
    dictionary'''

    key_List = []#creating empty lsit
    dictionary = dict_Function()#assiging return value of method to dictionary
    variable
    for key in dictionary.keys():#iterates all key from dictionary one by one
        key_List.append(key)# adding key to list
    return key_List#returns list

```

#-----making list of book name-----

```

def list_BookName():
    '''creates a list collection data type which stores all book name value from
    dictionary'''

    bookName_List = []#creating empty lsit
    dictionary = dict_Function()#assiging return value of method to dictionary
    variable
    for value in dictionary.values():#iterates all book name from dictionary one by
    one
        bookName_List.append(value[0].lower())# adding book name from
    dictionary to list
    return bookName_List#returns list

```

#-----displaying book table-----

```

def dict_display():
    '''displays list of book in a table'''

    print("-----")

```

```

print("Book ID\t\t| " + "\tBook Name \t\t| " + "\tAuthor \t\t| " + "\tQuantity \t| " +
"\tPrice" )
print("-----")
-----")
display_dictioanry = dict_Function()
for key,value in display_dictioanry.items():#iterates all key,value from
dictionary one by one
    print(key + "\t\t|\t" + value[0] + "\t\t|\t" + value[1] + "\t|\t" + value[2] + "\t\t|\t" +
value[3])
    print("-----\n")

```

#-----borrow function of program-----

```

def borrow_Function():
    """handles all operation for browswing book by integrating all methods for
borrow"""

    global costumerName,borrowedBook_ID_List#declaring variable as global
    dict_display()#calling method to display booklist table
    borrowedBook_ID_List = []#creating empty list
    borrow = True#assigning boolean value to variable
    continuity = False
    while continuity == False:#giving condition for execuation of while loop
        costumerName = input("Please, Enter the borrower's name :
").replace(":", "")#asking costumer name and replacing colon to empty string
        if costumerName == "": #checking value of costumer name is empty
            print("-----")
            print("\t\tName cannot be left empty!!! Please, Enter the borrower's
name. ")
            print("-----\n")
        else:
            continuity = True
            validation_BorrowedbookID()#calling validation_BorrowedbookID
            while borrow == True:
                print("\nDo you want to borrow more books ?")
                continuity = input("If you want then Please, Enter 'Y' for YES and 'N' for
NO to borrow more books : ").lower()
                print("\n")
                if continuity == "y" :
                    validation_BorrowedbookID()#validation_BorrowedbookID method
                    will be invoked if user input is y

```

```

        elif continuity == "n":
            borrow = False
            print("\n-----")
            print("\tUpdated books list in library after borrowing books : ")
            print("-----")
            dict_display()
            borrowDetails_Write_ReadOp()#borrowDetails_Write_ReadOp
method will be invoked if user input is n

print("\n*****")
print("\t\t\tThank you for using our library management system.")

print("*****")
else:
    print("\n-----")
    print("\t\t\tInvalid Input!!! Please, Provide 'Y' for YES and 'N' for NO to
borrow more books. ")
    print("-----")

def validation_BorrowedbookID():
    """Handels validation operation of borrowed book ID provided by user"""
    date_Borrow = datetime.datetime.now().strftime("%y-%m-%d,%a")
    time_Borrow = datetime.datetime.now().strftime("%H:%M")

    bookID_List = allKey_List()#assiging return value of allKey_List method to
variable
    dictionary = dict_Function()#assiging return value of dict_Function method to
variable
    userInput_BookID = input("Enter book ID you want to borrow : ")
    if userInput_BookID in bookID_List:#checking condition to allow execution if
user input book id is in list
        if userInput_BookID not in borrowedBook_ID_List:#checking condition to
allow execution if user input book id is not in list
            if(int(dictionary[userInput_BookID][2]) > 0):#checking in quantity of book
is greater than 0

print("\n*****")
print(" \t\t\tThis Book is avilable for borrow.")

```

```

print("*****")
print("Book Name\t\t| " + "\tPrice \t\t| " + "\tDate of Borrow \t\t| " +
"\tTime of Borrow ")
print("-----")
#prints book name,price,date,weekdays and time of borrow
print(dictionary[userInput_BookID][0] + "\t\t\t" +
dictionary[userInput_BookID][3] + "\t\t\t" + str(date_Borrow) + "\t\t\t" +
str(time_Borrow) )
borrowedBook_ID_List.append(userInput_BookID)#adding user input
book id in list
updateBorrow_File()#calling updateBorrow_File method
else:
    print("\n-----")
    print(" \t\t\tBook is not available for now. Please, Select another book
from the stock.")
    print("-----\n")
validation_BorrowedbookID()#calling validation_BorrowedbookID
method
else:
    print("-----")
    print(" \t\tThis book is already borrowed by you. Please, select another
book from the stock.")
    print("-----\n")
validation_BorrowedbookID()#calling validation_BorrowedbookID method
else:
    print("-----")
    print("\t\t\tInvalid book Id!!! Please, provide valid Book ID from above book
table.")
    print("-----\n")
validation_BorrowedbookID()

def updateBorrow_File():
    """handles writing update of borrowing books details with updated quantity in txt
file """
    bookID_List = allKey_List()
    dictionary = dict_Function()

```

```

file = open("bookStock_list.txt","w")#opening txt file in write mode
for key,value in dictionary.items():#iterates all key,value from dictionary one
by one
    if key in borrowedBook_ID_List[-1]:#checking if value in last index of list
matches key value
        newQuantity = int(value[2]) - 1#converting string value of value at index
2 in to integer, subtracting 1 from it and assigin to variable
        if key == bookID_List[-1]:
            file.write(value[0] + "," + value[1] + "," + str(newQuantity)+ "," +
value[3])# writing in to txt file with update quantity value
        else:
            file.write(value[0] + "," + value[1] + "," + str(newQuantity)+ "," +
value[3] + "\n")
        else:
            if key == bookID_List[-1]:
                file.write(value[0] + "," + value[1] + "," + value[2] + "," +
value[3])#writing same old value to txt file
            else:
                file.write(value[0] + "," + value[1] + "," + value[2] + "," + value[3] + "\n")
file.close()#closing file

def costBorrow_book():
    """calculates the total amount to be paid for borrowing books"""

    dictionary = dict_Function()
    totalcost = 0
    for key in borrowedBook_ID_List: #iterates all key from list one by one
        rate = float(dictionary[key][3].replace("$", ""))#replacing $ sign of value at
index 3 in dictionary value list(price)and assing to varaible as float datatype
        totalcost += rate # adding rate for calculating total
        total_cost = "$" + str(totalcost)#converting total cost value to string and adding
$ sign to sting value
    return total_cost#returns the value of total_cost

def borrowDetails_Write_ReadOp():
    """handles writing operation for generating bill as txt file and reading operation
of same txt file """

    dictionary = dict_Function()
    date_Borrow = datetime.datetime.now().strftime("%y-%m-%d,%a")
    time_Borrow = datetime.datetime.now().strftime("%H:%M")
    minute_Value = str(datetime.datetime.now().minute)#getting only minute from
datetime module and converting it in to string

```



```

returnBooks_Id = []#creating empty list
days = []#creating empty list
return_book = False#assigning boolean value to variable
name = False
while name == False:
    costumerName = input("Please, Enter the returner's name :
").replace(":", "")#asking costumer name and replacing colon to empty string
    if costumerName == "" : #checking value of costumer name is empty
        print("-----")
        print("\t\tName cannot be left empty!!! Please, Enter the returner's name.
")
        print("-----\n")
    else:
        name = True#boolean value for exiting while loop
        validation_ReturnBookID()#calling validation_ReturnBookID method
        while return_book == False:
            print("\nDo you want to return more books ?")
            continuity = input("If you want then Enter 'Y' for YES and 'N' for NO to
return more books : ").lower()
            print("\n")
            if continuity == "y" :
                validation_ReturnBookID()#validation_ReturnBookID method will be
invoked if user input is y
            elif continuity == "n":
                return_book = True#boolean value for exiting while loop
                print("-----")
                print("\tUpdated books list in library after returning books : ")
                print("-----")
                dict_display()
                returnDetails_Write_ReadOp()#returnDetails_Write_ReadOp
method will be invoked if user input is n

print("\n*****")
print("\t\t\t\tThank you for using our library management system.")

print("*****")
else:
    print("\n-----")

```

```

        print("Invalid Input!!! Please Provide 'Y' for YES and 'N' for NO to
return books.. ")
        print("-----")
        -----")

def validation_ReturnBookID():
    """Handels validation operation of borrowed book ID provided by user"""

    bookID_List = allKey_List()#assiging return value of allKey_List method to
variable
    dictionary = dict_Function()#assiging return value of dict_Function method to
variable
    userInput_BookID = input("Enter book ID you want to return : ")

    if userInput_BookID in bookID_List:#checking condition to allow execution if
user input book id is in list
        if userInput_BookID not in returnBooks_Id:#checking condition to allow
execution if user input book id is not in list
            returnBooks_Id.append(userInput_BookID)#adding user input book id in
list
            selected = False#assigning boolean value to variable
            while selected == False:
                try :#handing ValueError exception using try except block
                    totalDays_Borrowed = int(input("Enter the numbers of days you
borrowed a book : "))#taking integer number from user
                    if totalDays_Borrowed > 0:
                        selected = True#boolean value for exiting while loop
                        days.append(totalDays_Borrowed)#adding user input day value in
list
                        updateReturn_File()#calling updateReturn_File method
                        print("\n-----")
                        -----")
                        print("This book is successfully returned by you.")
                        print("-----\n")

                else:
                    print("\n-----")
                    -----")
                    print("Invalid input!!! Please, provide days in positive number.")
                    print("-----\n")

            except:

```

```

        print("\n-----
-----")
        print("Invalid input!!! Please, provide days in number.")
        print("-----\n")
    else:
        print("\n-----
-----")
        print("\tThis book is already returned by you. Please, provide another
Book ID to return a book.")
        print("-----\n")
        validation_ReturnBookID()#calling validation_ReturnBookID method
    else:
        print("\n-----
-----")
        print("\t\t\tInvalid book ID!!! Please, provide a valid book ID from above book
table.")
        print("-----\n")
        validation_ReturnBookID()#calling validation_ReturnBookID method

def updateReturn_File():
    """handles writing update of borrowing books details with updated quantity in txt
file """
    bookID_List = allKey_List()
    dictionary = dict_Function()
    file = open("bookStock_list.txt", "w")#opening txt file in write mode
    for key,value in dictionary.items():#iterates all key,value from dictionary one
    by one
        if key in returnBooks_Id[-1]:#checking if value in last index of list matches
        key value
            newQuantity = int(value[2]) + 1#converting string value of value at index
            2 in to integer, adding 1 to it and assiging to variable
            if key == bookID_List[-1]:
                file.write(value[0] + "," + value[1] + "," + str(newQuantity)+ "," +
                value[3])#writing in to txt file with update quantity value
            else:
                file.write(value[0] + "," + value[1] + "," + str(newQuantity)+ "," +
                value[3] + "\n")
            else:
                if key == bookID_List[-1]:
                    file.write(value[0] + "," + value[1] + "," + value[2] + "," +
                    value[3])#writing same old value to txt file
                else:

```



```

    return_File.write("-----\n")
    return_File.write("Note : fine is charged 5% of total cost of each book per extra
days.\n")
    return_File.write("      Fine will be applicable after only exceeding over 10
days.\n")

return_File.write("*****\n*****\n")
return_File.close()#closing file
return_File = open(str(unique_fileName) + "R" + ".txt","r")#opening and reading
same file which is written above for customer bill
print(return_File.read())#reading txt file content line by line as single string
return_File.close()#closing file

def cal_Fine():
    """calculates the total fine to be paid for late returning of books"""

    dictionary = dict_Function()
    totalFine = 0
    noCharge_duration = 10#defining maximum days for no fine charge
    i = 0#setting counter for index
    for key in returnBooks_Id:#iterates all key from list one by one
        rate = float(dictionary[key][3].replace("$", ""))#replacing $ sign of value at
        index 3 in dictionary value list(price)and assing to variable as float datatype
        borrow_days = days[i]#iterating each days value one by one and assigning
        to variable
        if borrow_days > noCharge_duration:#checking if borrow days exceeds
        charge less duration in days
            extra_Days = borrow_days - noCharge_duration#claculating fineable
            days which are over 10 days
            fine = 0.05 * rate * extra_Days#fine is charged 5% of total cost of each
            book per extra days
            totalFine += fine#adding rate for calculating total
        else:
            fine = 0#setting fine to zero if lending days is less than 10
            totalFine += fine
        i += 1#incrementing counter value by 1
    totalFine_Amount = "$" + str(round(totalFine,2))#converting totalFine value to
    string and adding $ sign to sting value and rounding digit up to 2
    return totalFine_Amount#returns the value of totalFine_Amount

```

```
#-----adding new books to book stock list-----
-----
def add_Books():
    '''handles adding operation of book in book stck list.'''
    global book_Name#defining global variable
    bookName = list_BookName()
    print("\n")
    bookName_NotEmpty = False
    while bookName_NotEmpty == False:
        book_Name = input("Please, provide the book name : ")
        if book_Name == "":
            print("\n-----")
            print("\tBook name cannot be left empty, Please provide the book name.")
            print("-----\n")
        else:
            if book_Name.lower() not in bookName:#coverting string value of book_Name into lower case and checking in the list
                bookName_NotEmpty = True
                userInput()#calling method
                add_Write()
                print("\n-----")
                print("This book is successfully added to book list.")
                print("-----\n")
                count = True
                while count == True:
                    print("\nDo you want to add more books ?")
                    answer = input("Please, provide 'Y' for Yes and 'N' for No to add more books : ").lower()
                    print("\n")
                    if answer == "y":
                        add_Books()
                        count = False
                    elif answer == "n":
                        count = False
                        print("\n-----")
                        print("\tUpdated books list in library after adding books : ")
                        print("-----")
                        dict_display()
                    else:
```

```

        print("\n-----")
        print("-----")
        print("Invalid input!!! please, Provide 'Y' for Yes and 'No' for No.")
        print("-----")
        print("-----\n")

    else:
        print("\n-----")
        print("-----")
        print("\tThis book is already in book Stock list.Please, provide another
Book Name to add book.")
        print("-----")
        print("-----\n")

def add_Write():
    """handles writing operation for adding new book details in txt file."""
    file = open("bookStock_list.txt","a")#opening file in append mode
    file.write("\n" + book_Name + "," + book_Author_Name + "," +
str(book_Quantity) + "," + "$" + str(book_price))#updating txt file with new book
details
    file.close

def userInput():
    """handles query operation for getting values from user"""
    global book_Author_Name,book_Quantity,book_price#defining global variable
    AuthorName_NotEmpty = False
    quantityValue_NotInt = True
    bookPrice_NotFloat = True
    while AuthorName_NotEmpty == False:
        book_Author_Name = input("Please, provide the book author's name : ")
        if book_Author_Name == "":
            print("\n-----")
            print("-----")
            print("Book author's name cannot be left empty, Please provide the book
author's name.")
            print("-----")
            print("-----\n")
        else:
            AuthorName_NotEmpty = True
    while quantityValue_NotInt == True:
        try:#handing ValueError exception using try except block
            book_Quantity = int(input("Please, provide the book quantity : "))
            if book_Quantity >= 0:
                quantityValue_NotInt = False

```

```
else:  
    print("\n-----")  
    print("-----")  
    print("Invalid input!!! Please provide valid quantity and of positive  
number. ")  
    print("-----\n")  
except:  
    print("\n-----")  
    print("-----")  
    print("Invalid input!!! Please provide quantity in number. ")  
    print("-----\n")  
while bookPrice_NotFloat == True:  
    try:#handing ValueError exception using try except block  
        book_price = float(input("please, provide the book price : "))  
        if book_price > 0:  
            bookPrice_NotFloat = False  
        else:  
            print("\n-----")  
            print("-----")  
            print("Invalid input!!! Please provide valid price and of positive number.  
")  
            print("-----\n")  
    except:  
        print("\n-----")  
        print("-----")  
        print("Invalid input!!! Please provide price in number. ")  
        print("-----\n")  
return book_Author_Name,book_Quantity,book_price#returns respective value
```

References

- Coleman & Co. Ltd., 2021. *pseudocode*. [Online]
Available at: <https://economictimes.indiatimes.com/definition/pseudocode>
[Accessed 10 September 2021].
- Contributor, T., 1999 - 2021. *algorithm*. [Online]
Available at: <https://whatis.techtarget.com/definition/algorithm>
[Accessed 09 September 2021].
- Educative, Inc, 2021. *data-types-in-python*. [Online]
Available at: <https://www.educative.io/edpresso/data-types-in-python>
[Accessed 09 September 2021].
- Jaiswal, S., 2017. *data-structures-python*. [Online]
Available at: https://www.datacamp.com/community/tutorials/data-structures-python?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adposition=&utm_creative=278443377095&utm_targetid=ds
[Accessed 09 September 2021].
- Visual Paradigm, 2021. *flowchart-tutorial*. [Online]
Available at: <https://www.visual-paradigm.com/tutorials/flowchart-tutorial/>
[Accessed 10 September 2021].