LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS4051NI Fundamentals of Computing

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2020-21 Autumn

Student Name: Aayush Khanal

Group: N2

London Met ID: 20049123

College ID: NP01NT4S210050

Assignment Due Date: 2021/10/9

Assignment Submission Date:2021/10/9

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

List of Figures

List of Tables

1. Introduction:

This semester's coursework required us to create an application to manage a Library Management System. The Python programming language, as well as algorithms, flowcharts, and pseudo-codes, were utilized to create this software. This was our first python coursework; therefore, it was very new to us. It was even more interesting because it was difficult to study and execute. As we were told to program a library management system. The case scenario is a library that maintains the books in a text file. So, we need to develop a Python based application that can read the text file and display the details for borrowing a book. The inputs were entered to the program and the program should be able to write the entered data in a new text file. As, the process of borrowing books was repeated the number of stocks should be changed.

In the case of borrow, the borrower details were taken like name, books borrowed, total price, date and time of borrowed. All of these details were written in a txt file that also worked as the customer bill. In the case of return, the returner details are taken that contains the name, name of book, date and time of return and were written in a txt file. As, our lending duration is of 10 days exceeding that time will lead to additional penalty on a daily basis. The fine should be applied and then the file should be generated.

Introduction to Python is a programming language that is commonly used to create websites and applications, automate operations, and perform data analysis. Python is a general-purpose programming language, which means it can be used to develop a wide range of applications and isn't specialized for any particular problem. Because of its versatility and beginner-friendliness, it has become one of the most widely used programming languages today.

It is mainly used in the following applications: internet design (server-side), developing software, mathematics, Scripting on the system.

2. **Algorithm:**

2.1 FOR Main Function

Step 1: Start
Step 2: Display welcome message
Step 3: Enter 1,2 or 3 Try
Step 4: IF userInput = 1 Go to borrowBook
Step 5: elif userInput = 2 Go to return_books
Step 6: elif userInput = 3
    Step 6.1: Display Message
    Step 6.2 Break
Step 7: Except Value Error and show error message.
    Step 7.1: Go to step 2.
Step 8: Stop

2.2 For Book Function:

Step 1: Print BookID, Book-Name, Author, quantity price
Step 2: Open file books.txt
Step 3: Read lines of file.
Step 4: Print lines from txt file
Step 5: close File

2.3 FOR borrowBook Function:

Step 1: Define Borrow function.
Step 2: Create Variable sum
Step 3: For bookDictonary append each bookIDList
Step 4: Take bookID as input
Step 5: IF bookId not in bookIDList
    Step 5.1: Display message
    Step 5.2: Return to borrowBook
Step 6: IF bookId in bookIDList
    Step 6.1: Append borrowBooks to bookDictonary bookID
    Step 6.2: borrowedID append to bookID
    Step 6.3: quantity = bookDictonary bookID
    Step 6.4: IF quantity = 0
        Step 6.4.1: Show appropriate Message
    Step 6.5: Else Print appropriate message
        Step 6.5.1: Take name input as f_name
        Step6.5.2: Display Price of Book
        Step 6.5.3: Display date and time of borrow

Step 6.5.4: Open books.txt file as f
Step 6.5.5: Update stock list in books.txt file
Step 6.6.6: Write new values to the books.txt file
Step 6.6.7: Display list after borrow.
Step 6.6.9: Call Bill function

2.4 For moreBooks Function

Step 1: Create bookIdList
Step 2: append bookIDList
Step 3: Input bookId
Step 4: If bookId is not in bookId list
    Step 4.1: Display appropriate message and go to step 3
Step 5: else
    Step 5.1: Show price of book and ask if another book is borrowed
    Step 5.2: If y go to 5
        Else return.

2.5. For Bill Function:

Step 1: Show borrowedBooks
Step 2: Display customer bill, Name, sum date and time.
Step 3: Write in fileName i.e. “Borrow_f_name,date and time”
Step 4: open fileName
Step 5: Write Details to file
Step 6: Close file
Step 7: call main function

2.6 For return_books function:

Step 1: Start
Step 2: Enter name as f_name
Step 3: Enter bookId to return book.
Step 4: If BookId is true
    Step4.1: Ask for days exceeded of borrow
    Step 4.2: Add fine 3$ per day
    Step 4.3: Add book to list
    Step 4.4: Open File as return_f_name, date and time
    Step4.5 Write to file
    Step 4.6 Display total bill
    Step 4.7 Go to main
Step 5: Else
    Step 5.1 Display message and go to step 3.

3. Flow Chart:

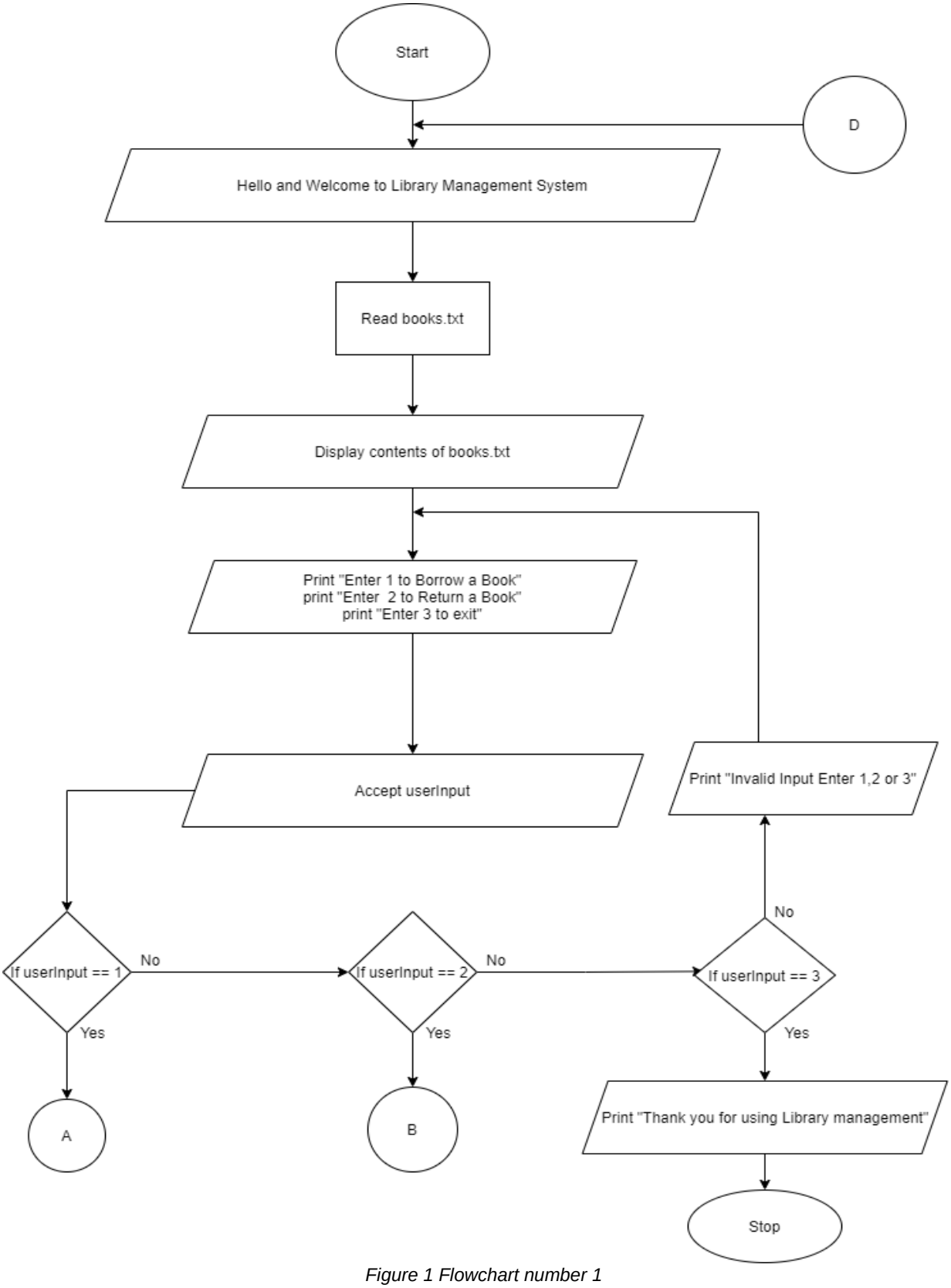

Figure 1 Flowchart number 1

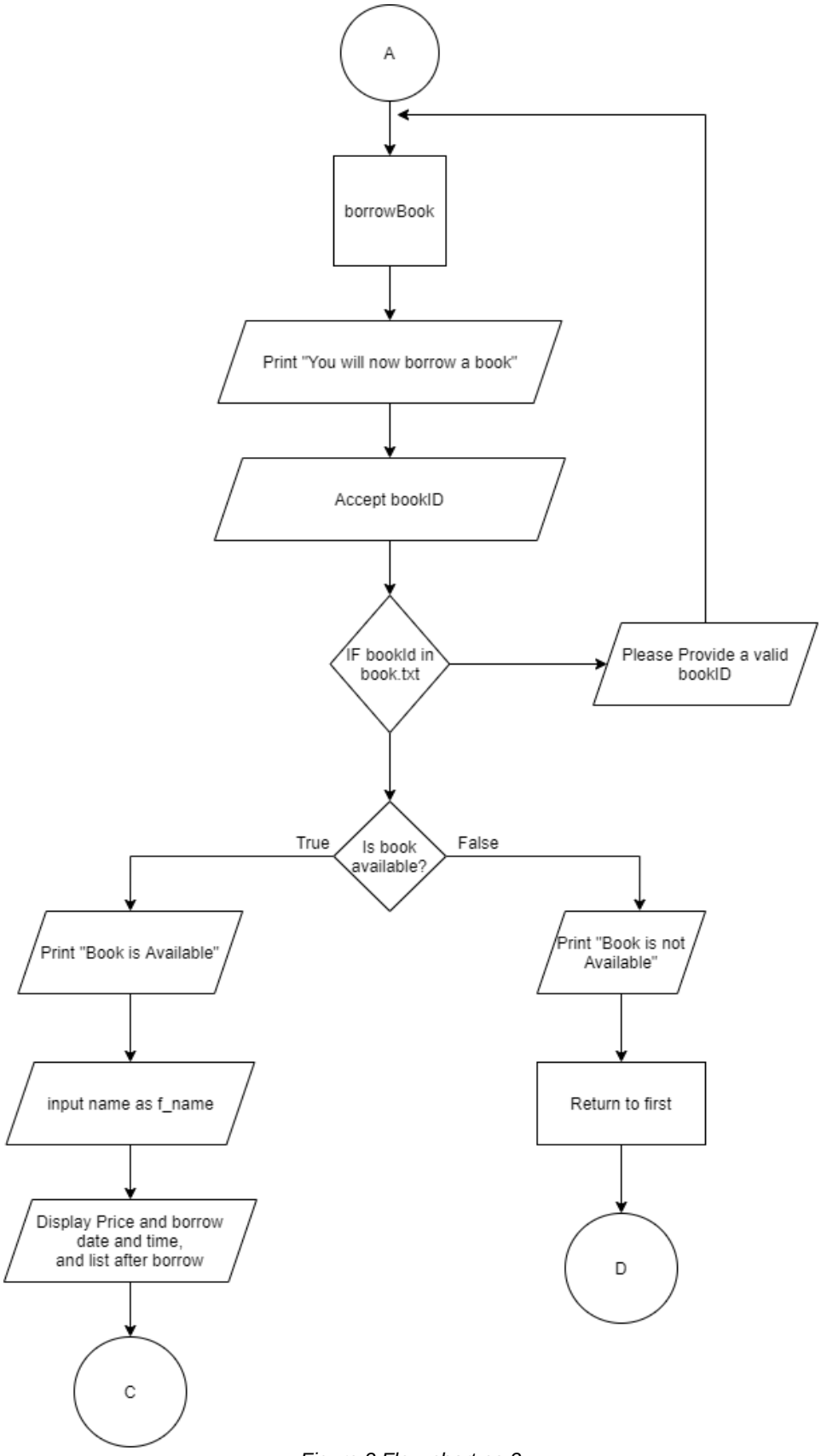

Figure 2 Flow chart no.2

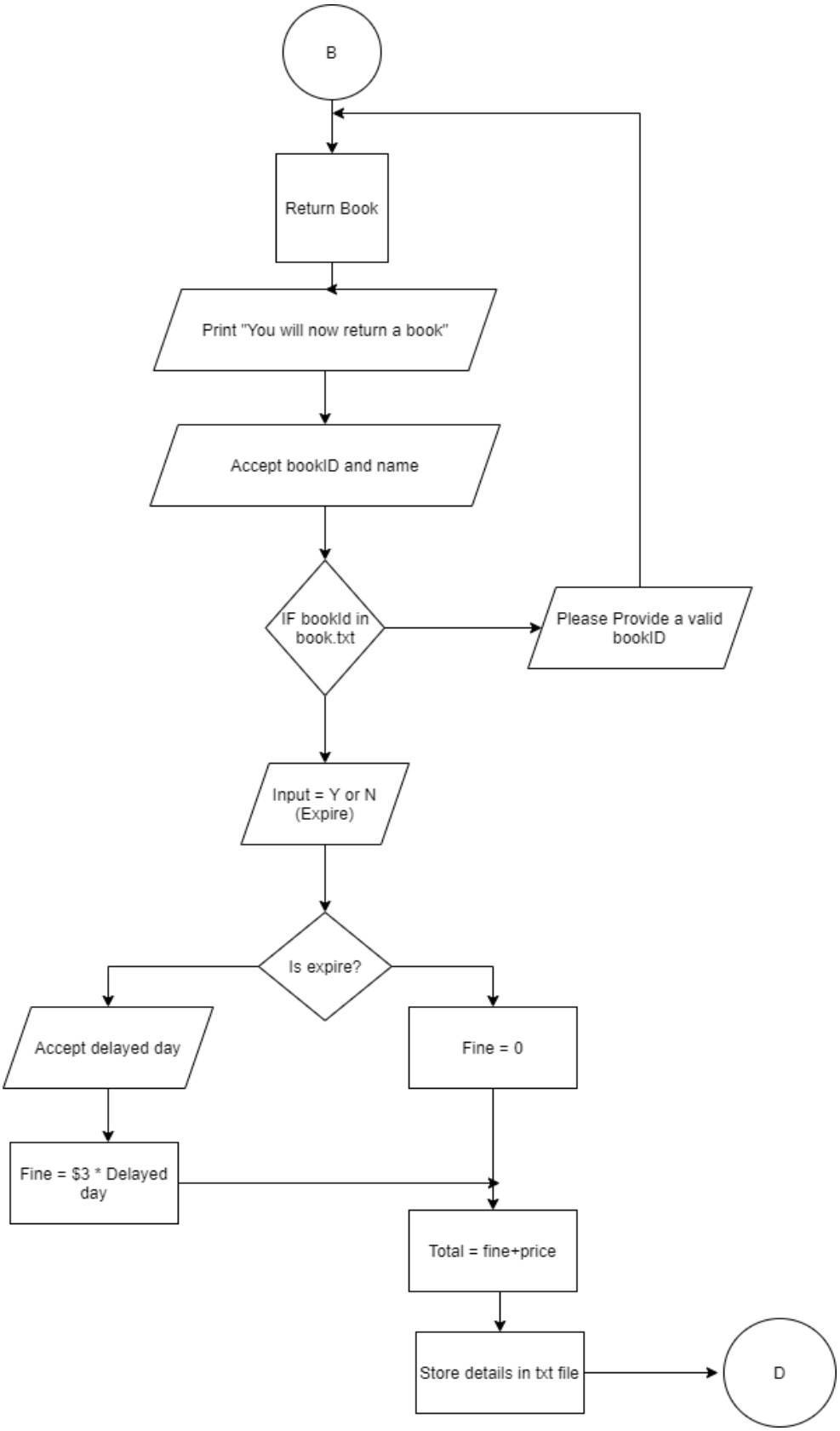

Figure 3 Flow chart no.3

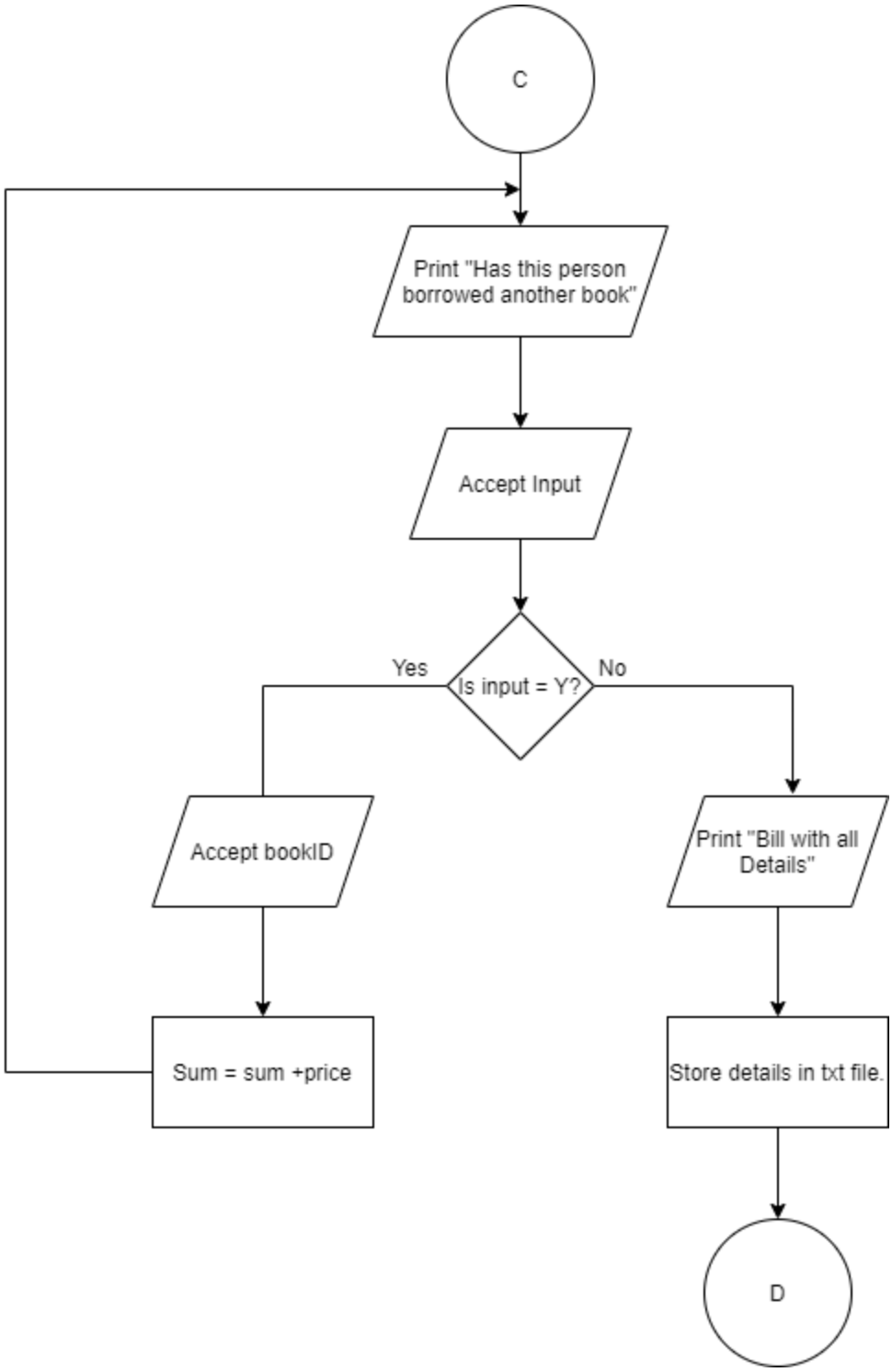

Figure 4 Flowchart no.4

4. Pseudocode:

4.1 main :

```
IMPORT borrow

IMPORT Book

IMPORT Return

IMPORT Bill


DEFINE FUNCTION main():

    while(True):

        OUTPUT("+++" * 35)

        OUTPUT("Hello and Welcome to library management system".center(100))

        OUTPUT("+++" * 35+ "\n\n")

        Book.Book()

        OUTPUT("Enter 1 to Borrow a Book")

        OUTPUT("Enter 2 to Return a book")

        OUTPUT("Enter 3 to Exit")

        TRY:

            SET userInput TO int(INPUT("Please enter a value: "))

            OUTPUT()

            if(userInput==1):

                OUTPUT("You will now borrow a book")

                borrow.borrowBook()

                main()
```

```
            break

        elif(userInput==2):

            OUTPUT("You will now reuturn a Book")

            Return.RETURN_books()


        elif(userInput==3):

            OUTPUT("---"* 30)

            OUTPUT("\n Thank you FOR using Library management
system\n".center(100))

            OUTPUT("---"* 30)

            break

    except ValueError:

        OUTPUT("++" *30)

        OUTPUT("Invalid Input!!\n Please enter 1, 2 or 3")

        OUTPUT("++" *30+ "\n\n")
main()
```

4.2 Borrow:

```
from datetime IMPORT date

from datetime IMPORT datetime

IMPORT Book

IMPORT Bill


SET bookDictionary TO {}

SET borrowedBooks TO []
```

```
SET borrowedID TO []


DEFINE FUNCTION borrowBook():
    SET sum TO 0
    SET bookIDList TO []
    FOR each IN bookDictionary:
        bookIDList.append(each)
    SET bookID TO int(INPUT("Enter book id to borrow book: "))


    IF bookID not IN bookIDList:
        OUTPUT("\n" + "++" * 30 + "\n")
        OUTPUT("Please provide a valid Book ID !!!".center(50) + "\n" + "++" * 30 + "\n")
        borrowBook()


    IF bookID IN bookIDList:


        borrowedBooks.append(bookDictionary[bookID][0])
        borrowedID.append(bookID)
        SET quantity TO bookDictionary[bookID][2]
        IF quantity EQUALS 0:
            OUTPUT("\n" + "++" * 30 + "\n" + "Book is not available!!!".center(50) + "\n" +
"++" * 30 + "\n")
            from main IMPORT main
```

```
    ELSE:
        OUTPUT("\n" + "++" * 30 + "\n" + "Book is available!!!".center(50) + "\n" + "++" *
30 + "\n")


        SET f_name TO INPUT("Enter the Name of person who borrowed book: ")
        SET price TO bookDictionary[bookID][3]
        OUTPUT("The price of book is: ",price)
        SET today TO date.today()
        SET now TO datetime.now()
        SET current_time TO now.strftime("%H:%M:%S")


        OUTPUT("Today's date and Time: ", today,current_time, "\n")


        bookDictionary[bookID][2]-= 1


        with open('books.txt', 'w') as f:
            FOR i IN range(1, 6):
                f.write(str(i))
                f.write(",")
                FOR j IN range(4):
                    if(j EQUALS 3):


                        f.write(str(bookDictionary[i][j]))
```

```
            ELSE:
                f.write(str(bookDictionary[i][j]))
            f.write(", ")
        f.write("\n")
```

```
OUTPUT("List after a Borrow: ")
Book.Book()
OUTPUT("Has This Person Borrowed Another Book?")
SET AnotherBook TO INPUT("If yes then type Y ,if not they type n: ")
IF AnotherBook.lower() EQUALS "y":
    Book.moreBooks()


FOR x in(borrowedID):
    SET p TO bookDictionary[x][3]
    SET price TO p[1:]
    SET sum TO sum + float(price)
Bill.Bill(f_name,sum,today,current_time)
```

4.3 Return:

```
from datetime IMPORT date

from datetime IMPORT datetime

IMPORT borrow

DEFINE FUNCTION RETURN_books():

    SET total TO 0

    SET bookIDList TO [ ]



    SET today TO date.today()

    SET now TO datetime.now()

    SET current_time TO now.strftime("%H:%M:%S")


    FOR each IN borrow.bookDictionary:

        bookIDList.append(each)

    SET f_name TO INPUT("Name of person who RETURNed Book: ")
    SET bookID TO int(INPUT("Enter book id to RETURN book: "))
    IF bookID not IN bookIDList:

        OUTPUT("\n" + "++" * 30 + "\n")

        OUTPUT("Please provide a valid Book ID !!!".center(50) + "\n" + "++" * 30 + "\n")

        RETURN_books()

    IF bookID IN bookIDList:

        SET quantity TO borrow.bookDictionary[bookID][2]

        SET price TO borrow.bookDictionary[bookID][3]

        SET book_name TO borrow.bookDictionary[bookID][0]

        SET author_name TO borrow.bookDictionary[bookID][1]
```

```
    quantity =int(quantity)+1
    total+=float(price[1:])
    OUTPUT(str(total))
    OUTPUT("Is the book RETURN date expired?")
    SET Expire TO INPUT("Press Y FOR Yes and N FOR No: ")
    x=current_time.replace(":","")
    if(Expire.upper()=="Y"):
        SET day TO int(INPUT("By how many days was the book RETURNed late?"))
        fine=3*day
        total=total+fine
    ELSE:
        SET fine TO 0
    SET total TO "{:.2f}".format(total)

    fileName="Return-"+f_name+str(today)+ x+".txt"
    SET file TO open(fileName,'w')
    file.write("Return Details: \n")
    file.write("Name of Person: "+ f_name+"\n")
    file.write("Date and time of RETURN is: "+str(today)+ " " )
    file.write(str(current_time) +"\n")
    file.write("Books Returend are: "+ borrow.bookDictionary[bookID][0]+"\n")
    file.write("Price is: " +borrow.bookDictionary[bookID][3]+ "\n")
```

```
    file.write("Fine is: "+ str(fine)+"\n")
    OUTPUT("Total Bill: "+ "$"+str(total))
    file.write("Total: $"+ str(total))


    borrow.bookDictionary[bookID][2]+= 1
    with open('books.txt', 'w') as f:
        FOR i IN range(1, 6):
            f.write(str(i))
            f.write(",")
            FOR j IN range(4):
                if(j EQUALS 3):
                    f.write(str(borrow.bookDictionary[i][j]))
                ELSE:
                    f.write(str(borrow.bookDictionary[i][j]))
                    f.write(", ")
            f.write("\n")
```

4.4 Book:

```
IMPORT borrow


DEFINE FUNCTION Book():
    OUTPUT("---" * 35)
```

```
    OUTPUT("Book ID".ljust(17), "Book-Name".ljust(30)  + "Author".ljust(20) +
"Quantity".ljust(17) + "Price".ljust(15))

    OUTPUT("---" * 35)



    SET file TO open('books.txt', 'r')

    SET content TO file.readlines()

    DEFINE FUNCTION splitLines():

        FOR each IN content:

            SET splitList TO each.split(',')

            SET bookID TO splitList[0]

            SET title TO splitList[1].ljust(25)

            IF len(title) > 25:

                SET title TO title[0:22].ljust(25, '.')

            SET author TO splitList[2].ljust(25)

            SET quantity TO splitList[3].ljust(15)

            SET price TO splitList[4].ljust(15)

            OUTPUT(bookID.ljust(15) + title + " " * 5 + author + quantity + price)

            SET borrow.bookDictionary[int(splitList[0])] TO [splitList[1].strip(),
splitList[2].strip(), int(splitList[3].strip()), splitList[4].strip()]

        OUTPUT("\n",borrow.bookDictionary)

        OUTPUT("\n")

    splitLines()

    file.close()
```

```
DEFINE FUNCTION moreBooks():


    SET bookIDList TO []
    FOR each IN borrow.bookDictionary:
        bookIDList.append(each)
    SET bookID TO int(INPUT("Enter another borrowed book: "))


    IF bookID not IN bookIDList:
        OUTPUT("\n" + "++" * 30 + "\n")
        OUTPUT("Please provide a valid Book ID !!!".center(50) + "\n" + "++" * 30 + "\n")
        moreBooks(sum)


    borrow.borrowedID.append(bookID)
    borrow.borrowedBooks.append(borrow.bookDictionary[bookID][0])
    SET price2 TO borrow.bookDictionary[bookID][3]
    OUTPUT("The price of book is: ",price2 + "\n")
    OUTPUT("Has This Person Borrowed Another Book?")
    SET AnotherBook TO INPUT("If yes then type Y ,if not they type n: " )
    IF AnotherBook.lower() EQUALS "y":
        moreBooks()
    ELSE:
        RETURN

Bill:
        IMPORT borrow
```

```
DEFINE FUNCTION Bill(f_name,sum,today,current_time):
    OUTPUT(borrow.borrowedBooks)
    SET sum TO "{:.2f}".format(sum)
    OUTPUT("\n"+"++" * 30)
    OUTPUT("Customer Bill!!!".center(50)+ "\n" + "++" * 30 + "\n")
    OUTPUT("Name of Customer: ",f_name)
    OUTPUT("Sum is $",sum)
    OUTPUT("Date and Time of Borrow is: ",today,current_time)
    OUTPUT("Name of Books Borrowed: ")

    FOR i in(borrow.borrowedBooks):
       OUTPUT(i)
    x=current_time.replace(":","")
    SET fileName TO "Borrow_"+ f_name + str(today)+ x+ ".txt"
    OUTPUT("++" *30 +"\n\n\n")
    file= open(fileName,'w')
    file.write("Borrow Details:"+ "\n")
    file.write("Name of Person: "+ f_name +"\n")
    file.write("Total Price of Books: "+str(sum) +"\n")
    file.write("Date and time of Borrow is: "+str(today) )
    file.write( str(current_time) +"\n")
    file.write("Books Borrowed are: "+"\n")
```

```
    FOR books IN (borrow.borrowedBooks):
        file.write(books + "\n")
    file.close()
    borrow.borrowedBooks.clear()
    borrow.borrowedID.clear()
```

5. Data Structures:

Data should be stored in a managed order. In python this can be done using data structure. Data structure are used to organize data in a way that it can be accessed in an easier way and various operations can be performed on they accordingly. (Akash, 2021)

Some of the data structure in python programming are as follows:

5.1 List:

List is a type of data structure that can be used to store different data types in a sequential order. Creating a list in python is very easy and, in a list, there are addresses assigned known as index. This data in the list can be accessed from last to first of first to last as there is both positive and negative indexing. We also have created list in our library management system and is shown in the image below.

```
borrowedBooks = []
borrowedID = []
```

Figure 5 List example 1

```
bookIDList = []
for each in bookDictionary:
    bookIDList.append(each)
bookID = int(input("Enter book id to borrow book: "))

if bookID not in bookIDList:
    print("\n" + "++" * 30 + "\n")
    print("Please provide a valid Book ID !!!".center(50) + "\n" + "++" * 30
    borrowBook()

if bookID in bookIDList:

    borrowedBooks.append(bookDictionary[bookID][0])
    borrowedID.append(bookID)
    quantity = bookDictionary[bookID][2]
    if quantity == 0:
        print("\n" + "++" * 30 + "\n" + "Book is not available!!!".center(50
        first()
```

Figure 6 List example 2

5.2 Dictionaries:

Dictionaries are the type of data structure that are used to store key-values pairs. Data can be stored as key or as values in pairs. That means if we have access to keys than we can also find the values. Dictionaries was a must while creating a library management system and some of the examples of dictionary are;

```
bookDictionary = {}
```

Figure 7 Dictionary example 1

```
f_name = input("Enter the Name of person who borrowed book: ")
price = bookDictionary[bookID][3]
print("The price of book is: ",price)
today = date.today()
now = datetime.now()
current_time = now.strftime("%H:%M:%S")

print("Today's date and Time: ", today,current_time, "\n")

bookDictionary[bookID][2]-= 1

with open('books.txt', 'w') as f:
    for i in range(1, 6):
        f.write(str(i))
        f.write(",")
        for j in range(4):
            if(j == 3):

                f.write(str(bookDictionary[i][j]))
            else:
                f.write(str(bookDictionary[i][j]))
            f.write(", ")
        f.write("\n")
```

Figure 8 Dictionary example 2

5.3 Sets:

Set is generally collection of unordered elements that are unique. So when we try to enter same data multiple times In a it only stores the data only once as it only stores unique data. It is similar to the set-in mathematics.

Example of set

My_set = {1,2,3,4,5}

To add elements in a set

My_set.add(6) output= {1,2,3,4,5,6}

5.4 Tuple:

Tuple are similar to lists the major difference is once the data is entered in the tuple it cannot be changed. That means if we entered a certain data then no matter what happens the data cannot be tampered. However, we can add elements using '+'operator. Example, of tuple is

```
My_tuple = (1,2,3)
My_tuple = My_tuple + (4,5,6)
Print(My_tuple)
Output= (1,2,3,4,5,6)
```

6. Program:

This program is a library management system and is developed using python. A library stores its data in a txt file so a txt was created that contains the list of the books. Various actions were carried out in this program that are show below

- When the program starts it displays a Welcome message and displays the list of books that is written in the library txt file.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                          Hello and Welcome to library management system
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


------------------------------------------------------------------------------------------------------------------
Book ID           Book-Name                    Author              Quantity         Price
------------------------------------------------------------------------------------------------------------------
1              The Alchemist                Paulo Coelho             83             $4.99

2              Harry Potter                 J.K Rowling              9              $9.99

3              vampire Diaries              L.J Smith                2              $12.00

4              Don Quixote                  Miguel de Cervantes      96             $15.00

5              Beloved                      Toni Morrison            93             $20


 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 9, '$9.99'], 3:
', 'Miguel de Cervantes', 96, '$15.00'], 5: ['Beloved', 'Toni Morrison', 93, '$20']}


Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value:
```

Figure 9 Welcome message of Program

- When 1 is entered as the value then it shows a message you will now borrow a book. Then it ask the user to input book id to borrow book. It checks the book list and if the book is available it shows book is available and asks the user to input the borrower name.

```
Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value: 1

You will now borrow a book
Enter book id to borrow book: 2

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                Book is available!!!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Enter the Name of person who borrowed book: Aayush
The price of book is:  $9.99
Today's date and Time:  2021-09-10 11:42:01

List after a Borrow:
```

Figure 10 Entering valid value to program

- Then it's shows the updated list after the borrow and asks if the person has borrowed another book. If yes then type y else type n. If n is typed then it shows the customer bill and list of books borrowed with date and time. This data is also written in a txt file name borrower_f_name date and time. Then the program re runs again.

```
List after a Borrow:
-------------------------------------------------------------------------------------------------------
Book ID          Book-Name                    Author             Quantity        Price
-------------------------------------------------------------------------------------------------------
1              The Alchemist                Paulo Coelho            83             $4.99
2              Harry Potter                 J.K Rowling             8              $9.99
3              vampire Diaries              L.J Smith               2              $12.00
4              Don Quixote                  Miguel de Cervantes     96             $15.00
5              Beloved                      Toni Morrison           93             $20

 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 8, '$9.99'], 3:
', 'Miguel de Cervantes', 96, '$15.00'], 5: ['Beloved', 'Toni Morrison', 93, '$20']}


Has This Person Borrowed Another Book?
If yes then type Y ,if not they type n: n
['Harry Potter']

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                Customer Bill!!!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Name of Customer:  Aayush
Sum is $ 9.99
Date and Time of Borrow is:  2021-09-10 11:42:01
Name of Books Borrowed:
Harry Potter
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure 11 Bill of borrow

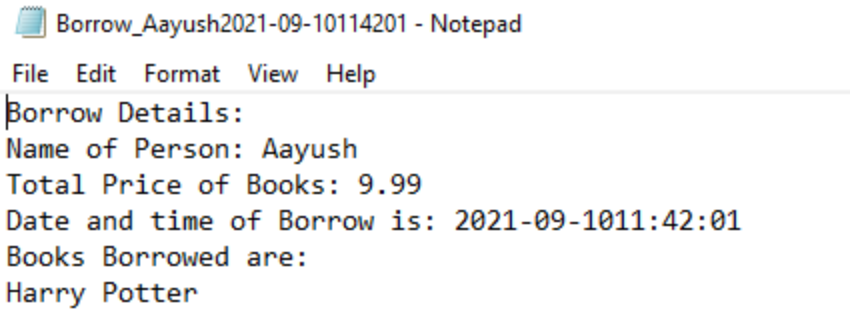

Figure 12 Bill in txt file for borrow

- In case if y is types then it asks the book id of the other borrowed book and prints the price and again asks to input y or n until n is typed. After n is typed it goes to the same process as above prints the bill and writes the bill in txt file and reruns the program.

```
Has This Person Borrowed Another Book?
If yes then type Y ,if not they type n: y
Enter another borrowed book: 4
The price of book is:  $15.00

Has This Person Borrowed Another Book?
If yes then type Y ,if not they type n: n
['Harry Potter', 'Don Quixote']

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                 Customer Bill!!!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Name of Customer:  aayush
Sum is $ 24.99
Date and Time of Borrow is:  2021-09-10 11:51:07
Name of Books Borrowed:
Harry Potter
Don Quixote
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure 13 Bill for multiple book Borrow

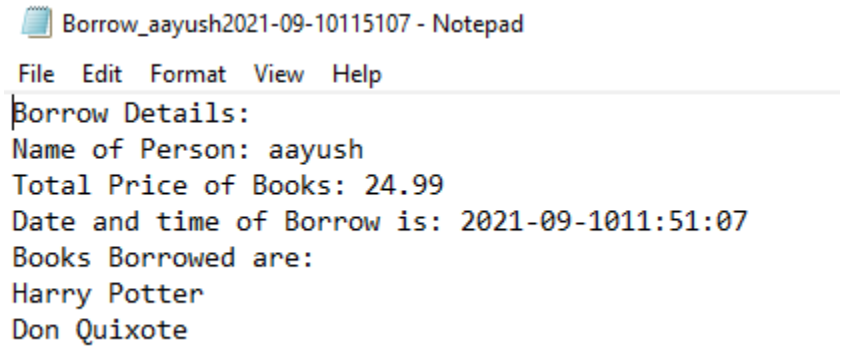

Figure 14 Figure 13 Bill for multiple book Borrow(txtfile)

- In case if someone provides an invalid bookId then it's shows a message and asks to re-enter the bookId again.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                          Hello and Welcome to library management system
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


------------------------------------------------------------------------------------------------------------------------
Book ID          Book-Name                    Author              Quantity         Price
------------------------------------------------------------------------------------------------------------------------
1              The Alchemist                Paulo Coelho            83             $4.99
2              Harry Potter                 J.K Rowling             7              $9.99
3              vampire Diaries              L.J Smith               2              $12.00
4              Don Quixote                  Miguel de Cervantes     96             $15.00
5              Beloved                      Toni Morrison           93             $20

 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 7, '$9.99'], 3:
', 'Miguel de Cervantes', 96, '$15.00'], 5: ['Beloved', 'Toni Morrison', 93, '$20']}


Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value: 1

You will now borrow a book
Enter book id to borrow book: 7

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

        Please provide a valid Book ID !!!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Enter book id to borrow book:
```

Figure 15 Providing Invalid input

- If a book is out of stock then it displays a message that book is not available.

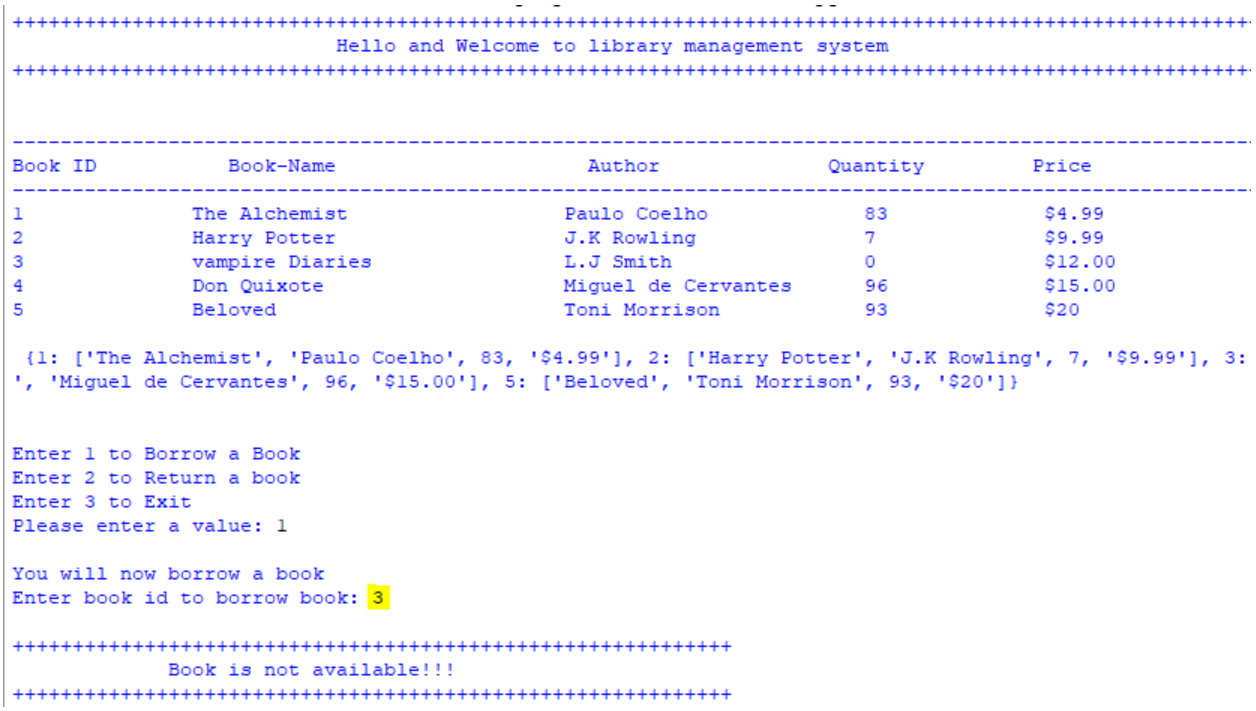

Figure 16 Borrowing book out of stock

- When 2 is entered then it asks the name of the person returning the book. Then it requests the id of the book that is going to be returned. Also, displays the price of the book.

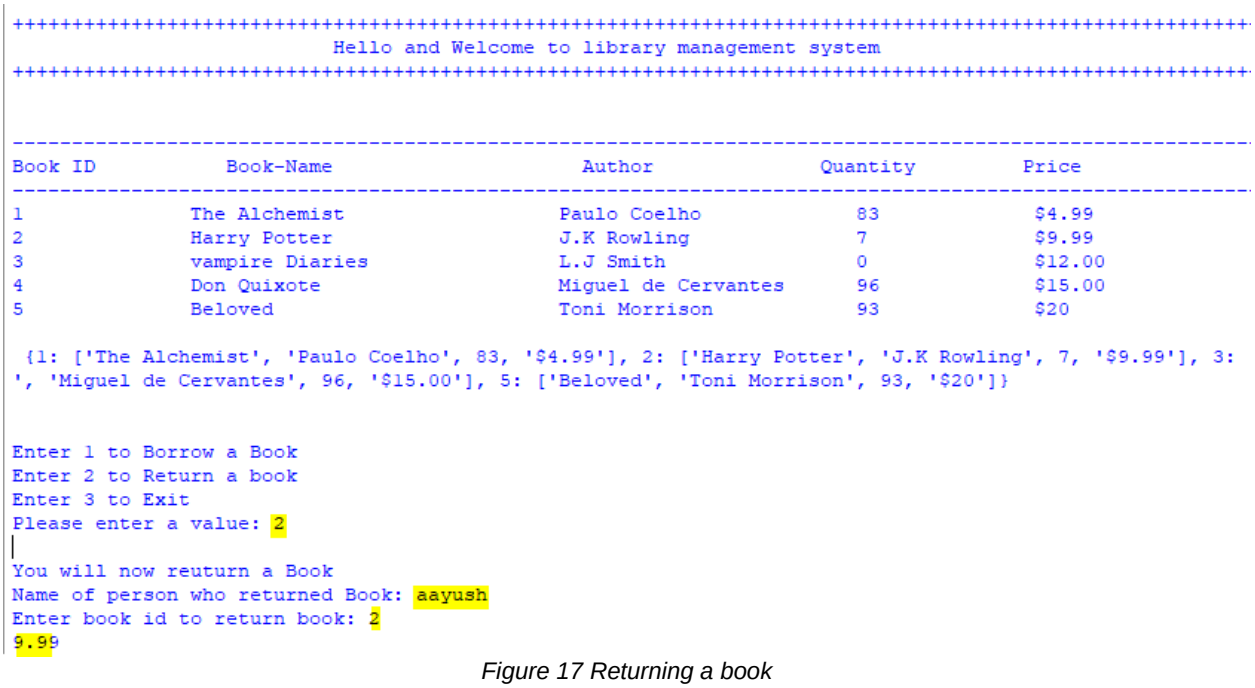

Figure 17 Returning a book

- Then it asks if the return date has been exceeded or not if y then it adds fine that is $3 per day and shows the final bill and writes the bill in txt file. And the stock is updated by +1

```
Is the book return date expired?
Press Y for Yes and N for No: y
By how many days was the book returned late?3
Total Bill: $18.99
```

Figure 18 Checking for days exceeded of borrow

Return-aayush2021-09-10121419 - Notepad

File Edit Format View Help

```
Return Details:
Name of Person: aayush
Date and time of return is: 2021-09-10 12:14:19
Books Returend are: Harry Potter
Price is: $9.99
Fine is: 9
Total: $18.99
```

Figure 19 Bill in txt file

Book ID	Book-Name	Author	Quantity	Price
1	The Alchemist	Paulo Coelho	83	$4.99
2	Harry Potter	J.K Rowling	8	$9.99
3	vampire Diaries	L.J Smith	0	$12.00
4	Don Quixote	Miguel de Cervantes	96	$15.00
5	Beloved	Toni Morrison	93	$20

```
 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 8, '$9.99'], 3:
', 'Miguel de Cervantes', 96, '$15.00'], 5: ['Beloved', 'Toni Morrison', 93, '$20']}
```

Figure 20 Proof that books.txt is updated

- If n is typed then the bill is shown and this stock is updated by 1 and a txtfile is generated.

```
Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value: 2

You will now reuturn a Book
Name of person who returned Book: aayush
Enter book id to return book: 3
12.0
Is the book return date expired?
Press Y for Yes and N for No: n
Total Bill: $12.00
```

Figure 21 Fine for exceeding borrow date

```
-------------------------------------------------------------------------------------------------------------
Book ID          Book-Name                    Author              Quantity        Price
-------------------------------------------------------------------------------------------------------------
1                The Alchemist                Paulo Coelho        83              $4.99

2                Harry Potter                 J.K Rowling         8               $9.99

3                vampire Diaries              L.J Smith           1               $12.00

4                Don Quixote                  Miguel de Cervantes 96              $15.00

5                Beloved                      Toni Morrison       93              $20

 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 8, '$9.99'], 3:
', 'Miguel de Cervantes', 96, '$15.00'], 5: ['Beloved', 'Toni Morrison', 93, '$20']}
```

Figure 22 Stock increasing after book returned

Return-aayush2021-09-10123813 - Notepad

File Edit Format View Help

```
Return Details:
Name of Person: aayush
Date and time of return is: 2021-09-10 12:38:13
Books Returend are: vampire Diaries
Price is: $12.00
Fine is: 0
Total: $12.00
```

Figure 23 Showing total bill in a txt file.

- If 3 is typed then the program is terminated.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                          Hello and Welcome to library management system
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


------------------------------------------------------------------------------------------------------------------------
Book ID          Book-Name                  Author             Quantity         Price
------------------------------------------------------------------------------------------------------------------------
1              The Alchemist              Paulo Coelho            83             $4.99

2              Harry Potter               J.K Rowling             8              $9.99

3              vampire Diaries            L.J Smith               1              $12.00

4              Don Quixote                Miguel de Cervantes     96             $15.00

5              Beloved                    Toni Morrison           93             $20


 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 8, '$9.99'], 3: ['
', 'Miguel de Cervantes', 96, '$15.00'], 5: ['Beloved', 'Toni Morrison', 93, '$20']}


Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value: 3

------------------------------------------------------------------------------------------

 Thank you for using Library management system

------------------------------------------------------------------------------------------
```

Figure 24 Terminating the Program

- If string value is entered in place of the integer value then the program shows invalid input and asks the to enter 1,2 or 3.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                          Hello and Welcome to library management system
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


------------------------------------------------------------------------------------------------------------------------
Book ID          Book-Name                  Author             Quantity         Price
------------------------------------------------------------------------------------------------------------------------
1              The Alchemist              Paulo Coelho            83             $4.99

2              Harry Potter               J.K Rowling             8              $9.99

3              vampire Diaries            L.J Smith               1              $12.00

4              Don Quixote                Miguel de Cervantes     96             $15.00

5              Beloved                    Toni Morrison           93             $20


 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 8, '$9.99'], 3: ['
', 'Miguel de Cervantes', 96, '$15.00'], 5: ['Beloved', 'Toni Morrison', 93, '$20']}


Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value: 3

------------------------------------------------------------------------------------------

 Thank you for using Library management system

------------------------------------------------------------------------------------------
```

Figure 25 Ending Message of program

7. Testing

7.1 Test no. 1

Objective	• To show the implementation of try, except.
Action	• As The program runs it requires input of 1,2 or 3 (integer) to work. • However, a string value was entered in the place of the integer value.
Expected Result	• An error message should be shown “Invalid Input Enter 1,2 or 3” • The program should request for input once again.
Actual Result	• An error message was shown as expected. • And the user was requested to input again.
conclusion	• The Test was successful and required output was obtained.

Table 1 Test 1

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                          Hello and Welcome to library management system
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


------------------------------------------------------------------------------------------------------------------
Book ID           Book-Name                    Author              Quantity         Price
------------------------------------------------------------------------------------------------------------------
1              The Alchemist                Paulo Coelho             83             $4.99

2              Harry Potter                 J.K Rowling              37             $9.99

3              vampire Diaries              L.J Smith                2              $12.00

4              Don Quixote                  Miguel de Cervantes      98             $15.00

5              Beloved                      Toni Morrison            99             $20


 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 37, '$9.99'], 3: ['vampire Diaries',
e', 'Miguel de Cervantes', 98, '$15.00'], 5: ['Beloved', 'Toni Morrison', 99, '$20']}


Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value:
```

Figure 26 Test 1 Image 1

```
Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value: Test 1
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Invalid Input!!
 Please enter 1, 2 or 3
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                          Hello and Welcome to library management system
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


------------------------------------------------------------------------------------------------------------------------------------
Book ID           Book-Name                      Author              Quantity         Price
------------------------------------------------------------------------------------------------------------------------------------
1              The Alchemist                  Paulo Coelho             84             $4.99

2              Harry Potter                   J.K Rowling              37             $9.99

3              vampire Diaries                L.J Smith                2              $12.00

4              Don Quixote                    Miguel de Cervantes      98             $15.00

5              Beloved                        Toni Morrison            99             $20


 {1: ['The Alchemist', 'Paulo Coelho', 84, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 37, '$9.99'], 3:
e', 'Miguel de Cervantes', 98, '$15.00'], 5: ['Beloved', 'Toni Morrison', 99, '$20']}


Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value:
```

Figure 27 Test 1 Image 2

7.2 Test no. 2

Objective	• Selection of borrow and return option. • To provide a negative value and a non-existed value.
Action	• First borrow process was selected. • A negative value was provided and output was observed. • Secondly return process was executed and non-existed value was provided.
Expected Result	• Error message should show up “Please provide a valid book ID” for both borrow and return case.
Actual Result	• Error message was shown as expected for both cases.
conclusion	• Hence, the test was successful.

Table 2 Test 2

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                              Hello and Welcome to library management system
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


------------------------------------------------------------------------------------------------------------------------------
Book ID           Book-Name                       Author              Quantity          Price
------------------------------------------------------------------------------------------------------------------------------
1               The Alchemist                   Paulo Coelho            84              $4.99

2               Harry Potter                    J.K Rowling             37              $9.99

3               vampire Diaries                 L.J Smith               2               $12.00

4               Don Quixote                     Miguel de Cervantes     98              $15.00

5               Beloved                         Toni Morrison           99              $20


 {1: ['The Alchemist', 'Paulo Coelho', 84, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 37, '$9.99'],
e', 'Miguel de Cervantes', 98, '$15.00'], 5: ['Beloved', 'Toni Morrison', 99, '$20']}


Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value: 1

You will now borrow a book
Enter book id to borrow book: -88

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

        Please provide a valid Book ID !!!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Enter book id to borrow book:
```

Figure 28 Test 2 Image 1

```
------------------------------------------------------------------------------------------------------------------------------
Book ID           Book-Name                       Author              Quantity          Price
------------------------------------------------------------------------------------------------------------------------------
1               The Alchemist                   Paulo Coelho            84              $4.99

2               Harry Potter                    J.K Rowling             37              $9.99

3               vampire Diaries                 L.J Smith               2               $12.00

4               Don Quixote                     Miguel de Cervantes     98              $15.00

5               Beloved                         Toni Morrison           99              $20


 {1: ['The Alchemist', 'Paulo Coelho', 84, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 37, '$9.99'],
e', 'Miguel de Cervantes', 98, '$15.00'], 5: ['Beloved', 'Toni Morrison', 99, '$20']}


Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value: 2

You will now reuturn a Book
Name of person who returned Book: aayush
Enter book id to return book: 99

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

        Please provide a valid Book ID !!!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Name of person who returned Book:
```

Figure 29 Test 2 Image

7.3 Test no 3

Objective	• To test the entire, borrow process.
Action	• The borrow process was selected. • Books Id was entered for borrowing. • The borrower's name was entered as requested. • The person has borrowed another book so "Y" was entered. • Another book borrowed Id was entered and n was typed. .
Expected Result	• A bill should be printed in the name entered and the bill should be written in a txt file.
Actual Result	• The bill was shown and the bill was written in a new txt file.
conclusion	• Hence, the test was successful.

Table 3 Test 3

```
-------------------------------------------------------------------------------------------------------
Book ID          Book-Name                    Author              Quantity         Price
-------------------------------------------------------------------------------------------------------
1              The Alchemist                Paulo Coelho            84             $4.99

2              Harry Potter                 J.K Rowling             37             $9.99

3              vampire Diaries              L.J Smith               2              $12.00

4              Don Quixote                  Miguel de Cervantes     98             $15.00

5              Beloved                      Toni Morrison           99             $20


 {1: ['The Alchemist', 'Paulo Coelho', 84, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 37, '$9.99'],
e', 'Miguel de Cervantes', 98, '$15.00'], 5: ['Beloved', 'Toni Morrison', 99, '$20']}


Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value: 1

You will now borrow a book
Enter book id to borrow book: 1

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
               Book is available!!!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Enter the Name of person who borrowed book: aayush
The price of book is:  $4.99
Today's date and Time:  2021-09-09 16:46:00
```

Figure 30 Test 3 Image 1

```
List after a Borrow:
-------------------------------------------------------------------------------------------------------
Book ID          Book-Name                    Author              Quantity         Price
-------------------------------------------------------------------------------------------------------
1              The Alchemist                Paulo Coelho            83             $4.99
2              Harry Potter                 J.K Rowling             37             $9.99
3              vampire Diaries              L.J Smith               2              $12.00
4              Don Quixote                  Miguel de Cervantes     98             $15.00
5              Beloved                      Toni Morrison           99             $20

 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 37, '$9.99'],
e', 'Miguel de Cervantes', 98, '$15.00'], 5: ['Beloved', 'Toni Morrison', 99, '$20']}


Has This Person Borrowed Another Book?
If yes then type Y ,if not they type n: y
Enter another borrowed book: 2
The price of book is:  $9.99

Has This Person Borrowed Another Book?
If yes then type Y ,if not they type n: n
['The Alchemist', 'Harry Potter']

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                Customer Bill!!!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Name of Customer:  aayush
Sum is $ 14.98
Date and Time of Borrow is:  2021-09-09 16:46:00
Name of Books Borrowed:
The Alchemist
Harry Potter
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure 31 Test 3 Image 2

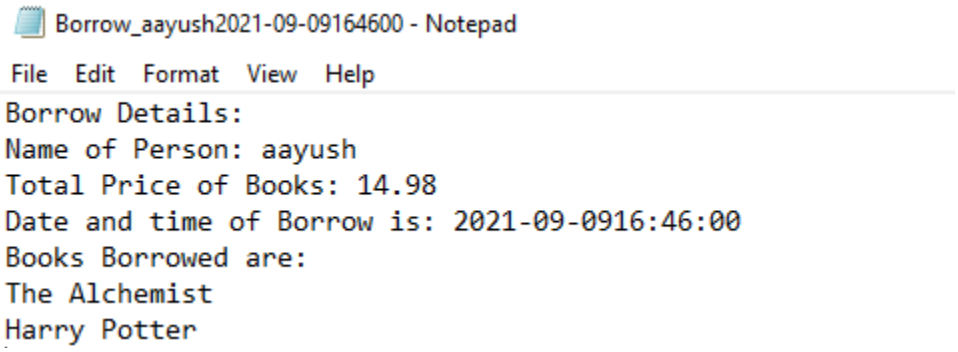

Figure 32 Test 3 Image 3

7.4 Test no. 4

Objective	• To test the entire, return process.
Action	• The return process was selected. • Books Id was entered for returning. • The person's name was entered as requested. • The person had exceeded time to return the book so "Y" was entered. • Delayed days were taken as input and the person was fined $3 per day.
Expected Result	• Total bill should be shown. • The bill should be written in a new txt file.
Actual Result	• The total bill was shown and the bill was written in a new txt file.
conclusion	• Hence, the test was successful.

Table 4 Test 4

```
-------------------------------------------------------------------------------------------------------------
Book ID          Book-Name                    Author              Quantity         Price
-------------------------------------------------------------------------------------------------------------
1              The Alchemist                Paulo Coelho            83             $4.99
2              Harry Potter                 J.K Rowling             37             $9.99
3              vampire Diaries              L.J Smith               2              $12.00
4              Don Quixote                  Miguel de Cervantes     98             $15.00
5              Beloved                      Toni Morrison           99             $20

 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 37, '$9.99'], 3: [
e', 'Miguel de Cervantes', 98, '$15.00'], 5: ['Beloved', 'Toni Morrison', 99, '$20']}


Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value: 2

You will now reuturn a Book
Name of person who returned Book: Aayush
Enter book id to return book: 4
15.0
Is the book return date expired?
Press Y for Yes and N for No: y
By how many days was the book returned late?3
Total Bill: $24.00
```

Figure 33 Test 4 Image 1

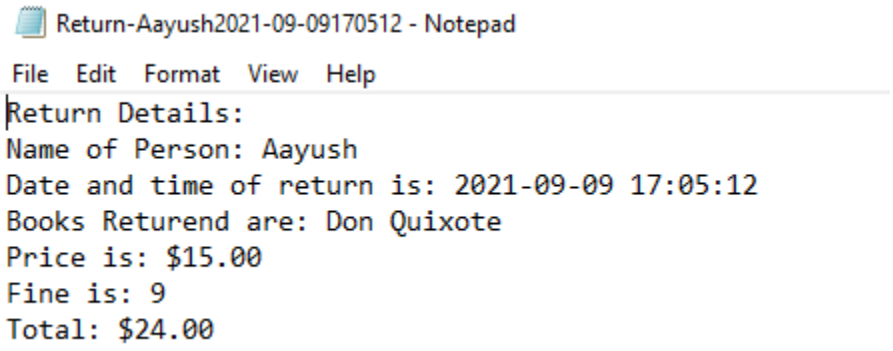

Figure 34 Test 4 Image 2

7.5 Test no. 5

Objective	• Testing if the stocks are updated.
Action	• A book was borrowed and the stock was checked. • The txt file was also checked for conformation. • A book was returned and stock was checked. • The txt file was also checked for conformation.
Expected Result	• The book stock should be decreased as the books are borrowed. • The book stock should be increased as the books were returned.
Actual Result	• The book stock was changed accordingly as expected.
conclusion	• Hence, the test was successful.

Table 5 Test 5

```
-------------------------------------------------------------------------------------------------------------
Book ID          Book-Name                    Author              Quantity        Price
-------------------------------------------------------------------------------------------------------------
1              The Alchemist                Paulo Coelho            83            $4.99

2              Harry Potter                 J.K Rowling             37            $9.99

3              vampire Diaries              L.J Smith               2             $12.00

4              Don Quixote                  Miguel de Cervantes     100           $15.00

5              Beloved                      Toni Morrison           99            $20


 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 37, '$9.99'], 3:
e', 'Miguel de Cervantes', 100, '$15.00'], 5: ['Beloved', 'Toni Morrison', 99, '$20']}


Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value: 1

You will now borrow a book
Enter book id to borrow book: 4

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
               Book is available!!!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Enter the Name of person who borrowed book: aayush
The price of book is:  $15.00
Today's date and Time:  2021-09-09 17:14:03

List after a Borrow:
```

Figure 35 List Before Borrow

```
List after a Borrow:
-------------------------------------------------------------------------------------------------------------
Book ID          Book-Name                    Author              Quantity        Price
-------------------------------------------------------------------------------------------------------------
1              The Alchemist                Paulo Coelho            83            $4.99
2              Harry Potter                 J.K Rowling             37            $9.99
3              vampire Diaries              L.J Smith               2             $12.00
4              Don Quixote                  Miguel de Cervantes     99            $15.00
5              Beloved                      Toni Morrison           99            $20

 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 37, '$9.99'],
e', 'Miguel de Cervantes', 99, '$15.00'], 5: ['Beloved', 'Toni Morrison', 99, '$20']}
```

Figure 36 List after borrow

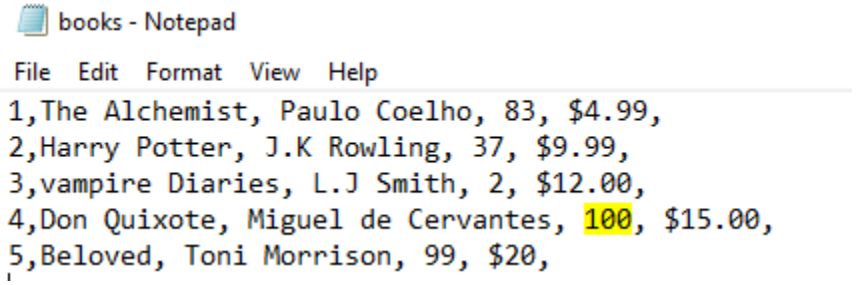

Figure 37 txt File before Borrow

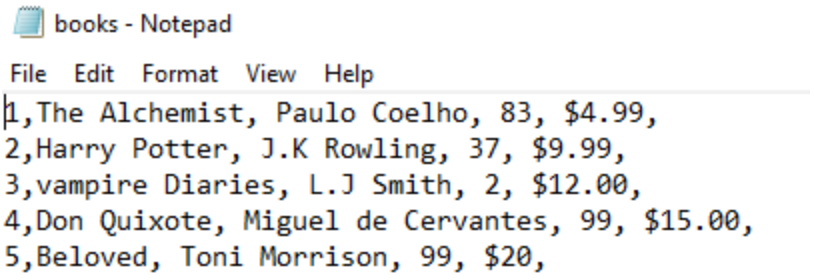

Figure 38 txt File after borrow

```
-------------------------------------------------------------------------------------------------------
Book ID          Book-Name                  Author              Quantity         Price
-------------------------------------------------------------------------------------------------------
1              The Alchemist              Paulo Coelho            83             $4.99
2              Harry Potter               J.K Rowling             37             $9.99
3              vampire Diaries            L.J Smith               2              $12.00
4              Don Quixote                Miguel de Cervantes     99             $15.00
5              Beloved                    Toni Morrison           99             $20

 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 37, '$9.99'], 3:
e', 'Miguel de Cervantes', 99, '$15.00'], 5: ['Beloved', 'Toni Morrison', 99, '$20']}


Enter 1 to Borrow a Book
Enter 2 to Return a book
Enter 3 to Exit
Please enter a value: 2

You will now reuturn a Book
Name of person who returned Book: aayush
Enter book id to return book: 3
```

Figure 39 List before return

```
-------------------------------------------------------------------------------------------------------
Book ID          Book-Name                  Author              Quantity         Price
-------------------------------------------------------------------------------------------------------
1              The Alchemist              Paulo Coelho            83             $4.99

2              Harry Potter               J.K Rowling             37             $9.99

3              vampire Diaries            L.J Smith               3              $12.00

4              Don Quixote                Miguel de Cervantes     99             $15.00

5              Beloved                    Toni Morrison           99             $20


 {1: ['The Alchemist', 'Paulo Coelho', 83, '$4.99'], 2: ['Harry Potter', 'J.K Rowling', 37, '$9.99'], 3:
e', 'Miguel de Cervantes', 99, '$15.00'], 5: ['Beloved', 'Toni Morrison', 99, '$20']}
```

Figure 40 List after return

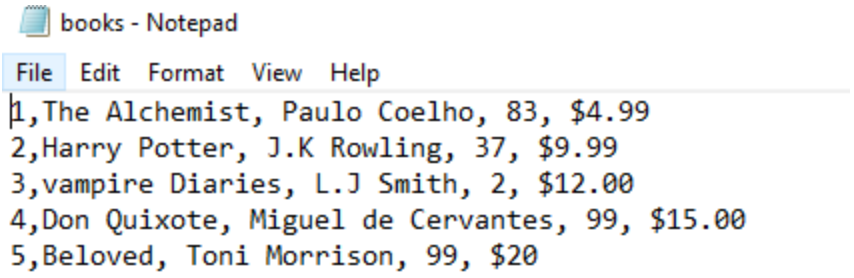

Figure 41 txt File before Return

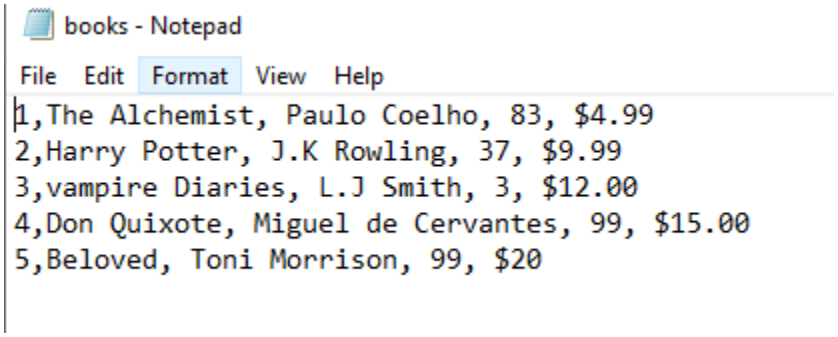

Figure 42 txt file after Return

8. Conclusion:

The given coursework was developed with the use of python programming language. The task assigned was not an easy work to do but with a lot of research and mending a lot of error and learning it made this coursework possible to complete. The code was written with the goal of being error-free and presenting the accurate information. Each task was completed in stages to guarantee that all tasks were completed effectively, helping the process go more smoothly.

This coursework was really important to us. The coding for this was a difficult task. The topic we discussed in class during the second semester provided the basis for the task. This program was prepared thoroughly by forming and algorithm and flowchart. This program we developed was a replacement for the traditional way of the library management system. Doing this coursework made me realize how a long-lived hard way thing could be handled so easily with the help of technologies and programming language. This coursework helped us to guide and demonstrate the use of python programming language and many more possibilities awaited in the future as an IT student. This was the one difficult and important coursework which was completed with the help of a lot of research.

9. Appendix

9.1 Main:

```
import borrow
import Book
import Return
import Bill

#Defining main function
def main():
    # using while loop
    while(True):
        print("+++" * 35)
        print("Hello and Welcome to library management system".center(100))
        print("+++" * 35+ "\n\n")
        Book.Book()
        print("Enter 1 to Borrow a Book")
        print("Enter 2 to Return a book")
        print("Enter 3 to Exit")

        #Using try and catch for exception Handling
```

```
        try:
            userInput = int(input("Please enter a value: "))
            print()
            # Taking input from user
            if(userInput==1):
                print("You will now borrow a book")
                borrow.borrowBook()
                main()
                break
            elif(userInput==2):
                print("You will now reuturn a Book")
                Return.return_books()

            elif(userInput==3):
                print("---"* 30)
                print("\n Thank you for using Library management system\n".center(100))
                print("---"* 30)
                break
        except ValueError:
            print("++" *30)
            print("Invalid Input!!\n Please enter 1, 2 or 3")
            print("++" *30+ "\n\n")

# calling main function
main()
```

9.2 borrow:

```
from datetime import date
from datetime import datetime
import Book
import Bill


#definingempty dictionary and list
bookDictionary = {}
borrowedBooks = []
borrowedID = []

#Creating borrowBook Function
def borrowBook():
    sum = 0
    bookIDList = []
    #checking user input
    for each in bookDictionary:
```

```
        bookIDList.append(each)
    bookID = int(input("Enter book id to borrow book: "))

    if bookID not in bookIDList:
        print("\n" + "++" * 30 + "\n")
        print("Please provide a valid Book ID !!!".center(50) + "\n" + "++" * 30 + "\n")
        borrowBook()

    if bookID in bookIDList:

        borrowedBooks.append(bookDictionary[bookID][0])
        borrowedID.append(bookID)
        quantity = bookDictionary[bookID][2]
        if quantity == 0:
            print("\n" + "++" * 30 + "\n" + "Book is not available!!!".center(50) + "\n" + "++" *
30 + "\n")
            from main import main

        else:
            print("\n" + "++" * 30 + "\n" + "Book is available!!!".center(50) + "\n" + "++" * 30 +
"\n")

            f_name = input("Enter the Name of person who borrowed book: ")
            price = bookDictionary[bookID][3]
            print("The price of book is: ",price)

            #adding date and time
            today = date.today()
            now = datetime.now()
            current_time = now.strftime("%H:%M:%S")

            print("Today's date and Time: ", today,current_time, "\n")

            #writing in file
            bookDictionary[bookID][2]-= 1

            with open('books.txt', 'w') as f:
                for i in range(1, 6):
                    f.write(str(i))
                    f.write(",")
                    for j in range(4):
                        if(j == 3):

                            f.write(str(bookDictionary[i][j]))
                        else:
                            f.write(str(bookDictionary[i][j]))
```

```
                f.write(", ")
            f.write("\n")


        print("List after a Borrow: ")
        Book.Book()
        print("Has This Person Borrowed Another Book?")
        AnotherBook = input("If yes then type Y ,if not they type n: ")
        if AnotherBook.lower() == "y":
            Book.moreBooks()


        for x in(borrowedID):
            p = bookDictionary[x][3]
            price = p[1:]
            sum = sum + float(price)
        #printing bill
        Bill.Bill(f_name,sum,today,current_time)
```

9.3 Return:

```
from datetime import date
from datetime import datetime
import borrow
# Creating return_books function
def return_books():
    total = 0
    bookIDList = []

    today = date.today()
    now = datetime.now()
    current_time = now.strftime("%H:%M:%S")

   #Asking user details
    for each in borrow.bookDictionary:
        bookIDList.append(each)
    f_name = input("Name of person who returned Book: ")
    bookID = int(input("Enter book id to return book: "))

    #Checking user input
    if bookID not in bookIDList:
        print("\n" + "++" * 30 + "\n")
        print("Please provide a valid Book ID !!!".center(50) + "\n" + "++" * 30 + "\n")
```

```
        return_books()
    if bookID in bookIDList:
        quantity = borrow.bookDictionary[bookID][2]
        price = borrow.bookDictionary[bookID][3]
        book_name = borrow.bookDictionary[bookID][0]
        author_name = borrow.bookDictionary[bookID][1]

    quantity =int(quantity)+1
    total+=float(price[1:])
    print(str(total))
    # checking for fine
    print("Is the book return date expired?")
    Expire = input("Press Y for Yes and N for No: ")
    x=current_time.replace(":","")
    if(Expire.upper()=="Y"):
        day = int(input("By how many days was the book returned late?"))
        fine=3*day
        total=total+fine
    else:
        fine = 0
    total = "{:.2f}".format(total)

    fileName="Return-"+f_name+str(today)+ x+".txt"
    file = open(fileName,'w')
    file.write("Return Details: \n")
    file.write("Name of Person: "+ f_name+"\n")
    file.write("Date and time of return is: "+str(today)+ " " )
    file.write(str(current_time) +"\n")
    file.write("Books Returend are: "+ borrow.bookDictionary[bookID][0]+"\n")
    file.write("Price is: " +borrow.bookDictionary[bookID][3]+ "\n")
    file.write("Fine is: "+ str(fine)+"\n")
    print("Total Bill: "+ "$"+str(total))
    file.write("Total: $"+ str(total))

    # Writing in file
    borrow.bookDictionary[bookID][2]+= 1
    with open('books.txt', 'w') as f:
        for i in range(1, 6):
            f.write(str(i))
            f.write(",")
            for j in range(4):
                if(j == 3):
                    f.write(str(borrow.bookDictionary[i][j]))
                else:
                    f.write(str(borrow.bookDictionary[i][j]))
                    f.write(", ")
```

```
        f.write("\n")
```

9.4 Book

```
import borrow

#Creating Book function
def Book():
    print("---" * 35)
    print("Book ID".ljust(17), "Book-Name".ljust(30)  + "Author".ljust(20) +
"Quantity".ljust(17) + "Price".ljust(15))
    print("---" * 35)

    #Opening and reading txt file
    file = open('books.txt', 'r')
    content = file.readlines()
    def splitLines():
        for each in content:
            splitList = each.split(',')
            bookID = splitList[0]
            title = splitList[1].ljust(25)
            if len(title) > 25:
                title = title[0:22].ljust(25, '.')
            author = splitList[2].ljust(25)
            quantity = splitList[3].ljust(15)
            price = splitList[4].ljust(15)

            print(bookID.ljust(15) + title + " " * 5 + author + quantity + price)
            borrow.bookDictionary[int(splitList[0])] = [splitList[1].strip(), splitList[2].strip(),
int(splitList[3].strip()), splitList[4].strip()]
        print("\n",borrow.bookDictionary)
        print("\n")
    splitLines()
    file.close()

#Creating moreBooks functon
def moreBooks():

    bookIDList = []
    for each in borrow.bookDictionary:
        bookIDList.append(each)
    bookID = int(input("Enter another borrowed book: "))

    if bookID not in bookIDList:
        print("\n" + "++" * 30 + "\n")
        print("Please provide a valid Book ID !!!".center(50) + "\n" + "++" * 30 + "\n")
```

```
        moreBooks(sum)

    borrow.borrowedID.append(bookID)
    borrow.borrowedBooks.append(borrow.bookDictionary[bookID][0])
    price2 = borrow.bookDictionary[bookID][3]
    print("The price of book is: ",price2 + "\n")
    print("Has This Person Borrowed Another Book?")
    AnotherBook = input("If yes then type Y ,if not they type n: " )
    if AnotherBook.lower() == "y":
        moreBooks()
    else:
        return
```

9.5 Bill

```
import borrow

# creating Bill function
def Bill(f_name,sum,today,current_time):
    print(borrow.borrowedBooks)
    sum = "{:.2f}".format(sum)
    print("\n"+"++" * 30)
    print("Customer Bill!!!".center(50)+ "\n" + "++" * 30 + "\n")
    print("Name of Customer: ",f_name)
    print("Sum is $",sum)
    print("Date and Time of Borrow is: ",today,current_time)
    print("Name of Books Borrowed: ")

    for i in(borrow.borrowedBooks):
        print(i)
    x=current_time.replace(":","")
    fileName = "Borrow_"+ f_name + str(today)+ x+ ".txt"
    print("++" *30 +"\n\n\n")

    file= open(fileName,'w')
    file.write("Borrow Details:"+ "\n")
    file.write("Name of Person: "+ f_name +"\n")
    file.write("Total Price of Books: "+str(sum) +"\n")
    file.write("Date and time of Borrow is: "+str(today) )
    file.write( str(current_time) +"\n")
    file.write("Books Borrowed are: "+"\n")

    for books in (borrow.borrowedBooks):
        file.write(books + "\n")
```

```
    file.close()
    borrow.borrowedBooks.clear()
    borrow.borrowedID.clear()
```

References

Akash, 2021. *edureka.* [Online]
Available at: https://www.edureka.co/blog/data-structures-in-python/
[Accessed 5 9 2021].