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: Sahil Sainju

Group:C9

London Met ID: 20048999

College ID: NP01CP4S210243

Assignment Due Date: 10th September 2021

Assignment Submission Date: 10th September 2021

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

Table of Contents

List of Figures

List of Tables

1. Introduction

Library management system is a project for managing borrowing and returning of book form a small library. Management of book in library includes keeping record of available books, books borrowed by users, books returned by the user, individual ledger of user and proper messages. Displaying appropriate messages after each borrowing and returning of book is very important as it gives sense of standing ground to current user.

This application shows books available to borrow by a user. Handling invalid inputs and waiting for a valid input from user is also to be included in this project. If the user is a returning user book borrowing or returning process can be proceed, if the user is not in user database i.e., s/he is a new user, new user is to be created in user database. After each transaction of book, the book stock database is required to be updated. For returning the borrowed book user can proceed the book returning process using return command. Like the book borrowing process, after returning the book an update is to be made to the book database along with appropriate message

The cost of borrowing a book should be shown to user and appropriate fine should be added if book is not returned within minimum return time. This application displays the list of books available to borrow. If the user selected book is available to borrow, book borrowing process is proceeding. A message is generated displaying details.

To return a borrowed book, return command is to be provided to the program which enables the book returning process. After completion of either book borrowing or book returning the main menu displays the books available to borrow and waits for user command. To exit the program, exit command is to be entered

This project was developed to complete fulfilment of first coursework. The course work was not an easy task There were a lot of hardships and sleepless nights to go through to complete the project in time. Lots of research, discussion, analysis and follow up meetings with teachers were required to be done to develop a library management system as required in the course work.

2. Discussion and Analysis

The course work is compilation of various tasks like performing research, preparing algorithm, flowchart, and pseudo code, writing code, and documenting the entire procedure for preparation of the report. With dedication and effort these tasks were accomplished

Python is a very simple yet powerful programming language which is accepted by wide variety of hardware and firmware extending form small raspberry pie to a supercomputer. Operating systems like macOS comes with inbuilt installation of python. These characteristics of python has made it one of the most widely accepted programming languages. Even for extensive tasks like machine learning and artificial intelligence python is considered as the first choice of programming language to go for. Python is not only limited to this, but various mobile apps can also be built using python only. This wide acceptance of python has built up large communities of developers which make sharing of idea and debugging a lot easier and more efficient task

To develop the program lecture sides, tutorials and lab workshop were revisited. From there idea of generating workflow of the coursework was developed. After performing some research proper data structure was selected and coding procedure was initiated. To code Python IDLE was used which was very helpful as it generated error messages which directed toward debugging of the program. For developing flowchart Draw.io was used. Draw.io is an opensource online drawing tool that can be used to develop flowchart and other diagrammatical infographics

For debugging of code, many google search were conducted and appropriate search result were selected form generated results. This gave wide angle idea of what the error message was about and how to solve the issue.

3.1 Algorithm

An Algorithm is step wise procedure to be followed to solve a problem. Developing algorithm is first step towards programming. It defines the steps to be taken which later leads the way towards creating a software. An algorithm is not limited to one platform. In fact, same algorithm can be used to develop programs in almost all programming language available till date. Here’s the algorithm for this project:

```
Step 1:   Start
Step 2:   Display menu
Step 3:   Input
Step 4:   If input==1 go to step a1, if input==2 go to step b1,
           If input==3 go to step c1 if input==4 go to step d1

Step a1: Read a text files containing book stock and display books available to
              Borrow
Step a2: Go to Step 2
Step b1: Takes input and If input is valid, check availability of book in Stock
Step b2: If book is available generates bill in txt form.
Step b3: Decrease amount of book borrowed by user, write to file, display message
Step b4: Go to Step 2
Step c1: Check if username is valid
Step c2: Increase the quantity of book returned, write to file, and generate bill in txt form
Step c3: Go to Step 2
Step d1: Exit the Program
```

3.2 Flowchart

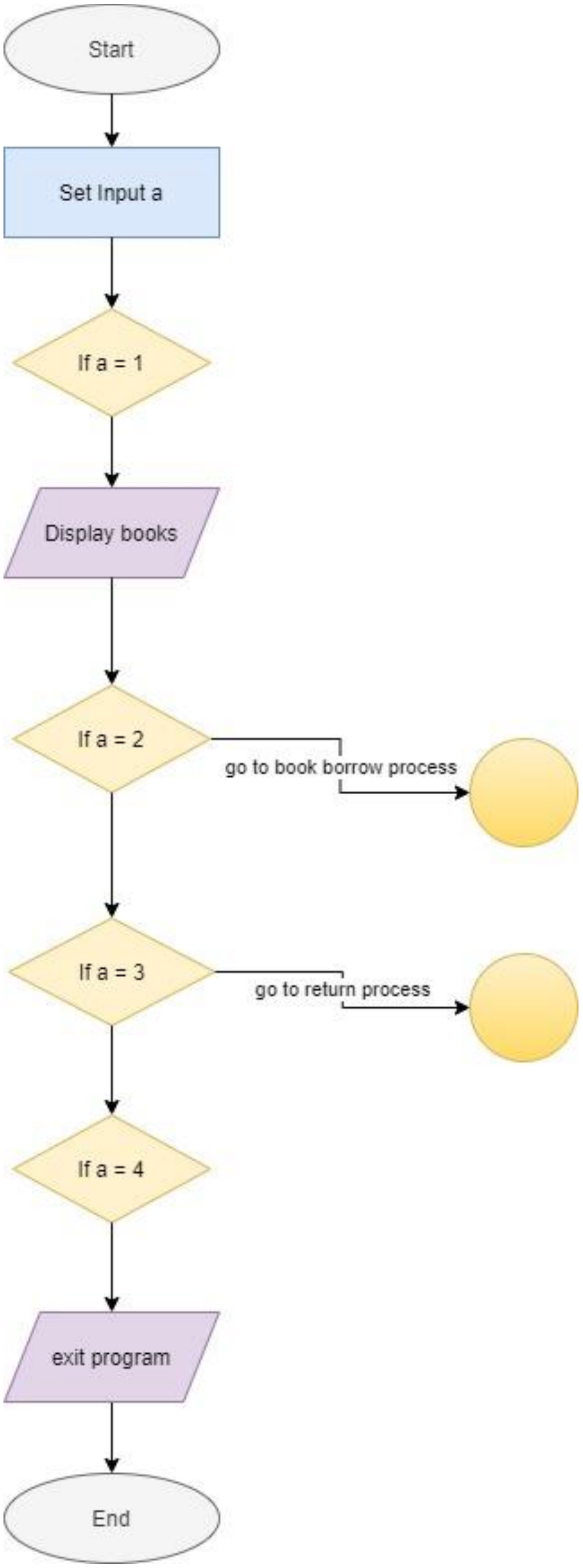


Figure 1: Flow Chart

3.3 Pseudocode

Pseudo code are false code. These codes are not understood by any compiler or interpreter, but they are very helpful while writing actual code. Pseudo code acts as a layer of false screen which is visible to developer but not to computer. It is programming written in human readable language. A pseudo code contains lines of statements which have clear meaning of what is needed to be done. Once a developer has written a pseudo code, s/he gets clear idea of programming. There is always opportunity of revisiting the pseudo code in case of any forgetfulness. The pseudo code of the program is as follows:

Main:

```
algorithm Start _Screen
Input a as choose a number from 1-4

          If a==1 Display books available to borrow

         IF a== 2, Go to Borrow process

          IF a== 3, Go to Return process

          IF a== 4, exit the program

  END IF

Else

         PRINT error message

END Start_Screen
```

Listsplit:

```
Algorithm listsplit
Set bookname as global
Set authorname as global
Set quantity as global
Set cost as global

Insert bookname to list
Insert author to list
Insert quantiy to list
Insert cost to list
Open txt file as mf
Read line in mf
For a in range (len(lines)):
        Set index as 0
End for

For  a in lines[i].split(','):
            if(ind==0):
               bookname.append(a)
            elif(ind==1):
               authorname.append(a)
            elif(ind==2):
               quantity.append(a)
            elif(ind==3):
                 cost.append(a.strip("$"))
           set ind+=1
end for
end listplit
```

Display:

```
Algorithm display
Read txt file
For line in bookstock
Print line
End for
End display
```

DateTime:

```
Algorithm datetime
# getdate
Gets datetime
Sets datetime as current date
Reutrns date
End getdate

# gettime
Gets datetime
Sets datetime as current time
Reutrns gettime
End time
End datetime
```

Borrow:

```
Algorithm borrow
Get borrowers name
Check validity of borrowers name
If valid borrowers name
            Go to book selection
If valid choice
             Decrease quantity of book in book stock
```

```
Generates a bill in txt file
IF multiple books
Set loop true
End borrow
```

Return

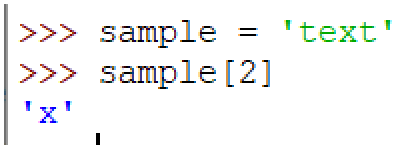
```
Algorithm return
Gets borrower name
Check validity of borrower name
If valid borrower name
        CHECK IF the book is past deadline
        Gets the days past deadline
        Generate a bill with fine
        Increases quantity of book in book stock
End return
```

3.4 Data Structure

Python has a wide range of data types that make working with groupings of data much easier. Simpler. Strings, tuples, lists, and dictionaries are the data types in Python. These data structures are a collection of data kinds.
We can introduce these data types briefly like strings are pieces of text, tuples and lists are ordered groups of individual data items. Dictionaries are groups of key-value pairs. Strings, tuples, and lists have same process of accessing data in sequence.
The data types are also categorized based on their capacity to be modified. Immutable objects such as strings and tuples cannot be modified, but mutable objects such as lists may be modified.

Strings:

Strings are simply pieces of text. Each character of the string can be accessed by their index. The index is zero-based, and it is specified as a whole number in square brackets following the variable name like sample[I] will retrieve the item at position I of sequence sample. Strings can be used anytime when a text is required in a program.

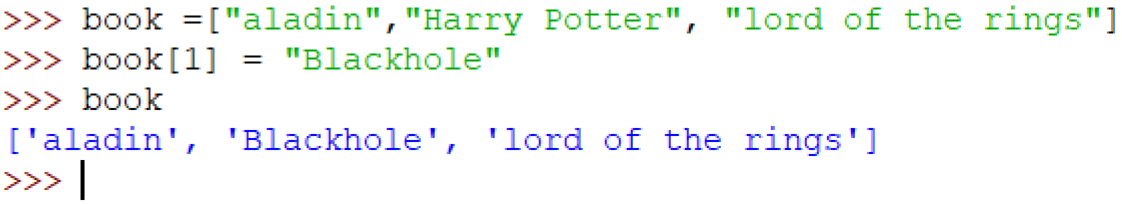


```
>>> sample = 'text'
>>> sample[2]
'x'
```

Figure 2: Accessing String using index

List

A list is a comma-separated list of objects, not all of which must be of the same type, contained in square brackets. To create a multidimensional list, a list can contain another list. Individual items in a list can be sliced, concatenated, indexed, and updated. When working with stacks, queues, sorting things, matrixes, and vast amounts of data, a list may be quite useful.



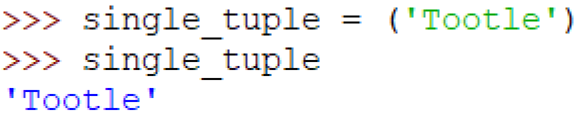
```
>>> book =["aladin","Harry Potter", "lord of the rings"]
>>> book[1] = "Blackhole"
>>> book
['aladin', 'Blackhole', 'lord of the rings']
>>> 
```

Figure 3: Creating list

Tuples

A tuple is an ordered collection of objects that cannot be changed. Tuples are sealed packed once they've been formed, which means they can't be changed once they've been made, but they can be rewritten.

Tuples are useful when a program requires a set of fixed values to be utilized repeatedly.

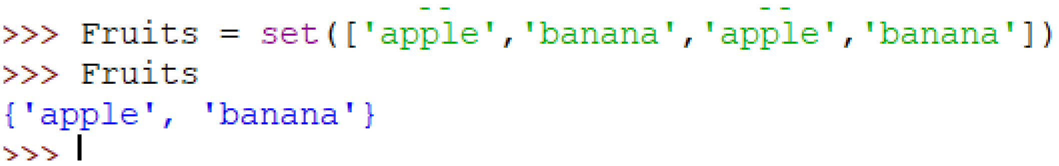


```
>>> single_tuple = ('Tootle')
>>> single_tuple
'Tootle'
```

Figure 4: Creating tuple

Sets

Sets are used to check for membership, separate duplicate entries into single items, and determine what is similar or different between two sets of things. In Python, we can use sets to execute all mathematical set operations. Here are a few examples:

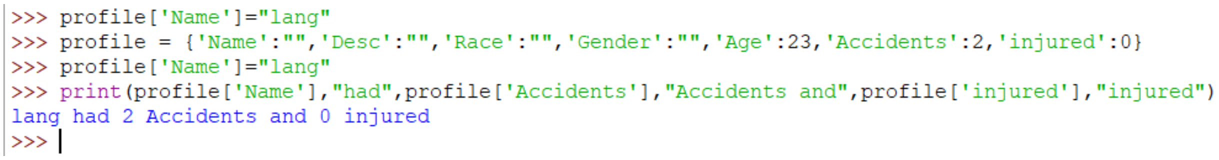


```
>>> Fruits = set(['apple','banana','apple','banana'])
>>> Fruits
{'apple', 'banana'}
>>> 
```

Figure 5: Creating set

Disctionary

A dictionary is a collection of unsorted things. Key-value pairs separated by colons are specified inside curly braces to build dictionaries. Keys can be used to access the values of dictionaries. An immutable type, such as text, integer, or tuple, must be used as the key. It doesn't matter what the value is. We can add, delete, and alter key-value pairs in a dictionary since it is a mutable data type. Dictionaries may be effectively used to add values such as a person's/information item's to keys such as the person's/name. item's



```
>>> profile['Name']="lang"
>>> profile = {'Name':"",'Desc':"",'Race':"",'Gender':"",'Age':23,'Accidents':2,'injured':0}
>>> profile['Name']="lang"
>>> print(profile['Name'],"had",profile['Accidents'],"Accidents and",profile['injured'],"injured")
lang had 2 Accidents and 0 injured
>>> 
```

Figure 6 : Creating a Dictionary

4. Program

In order to build a full functioning library management system, entire code was divided into six modules for proper management of code. A module is a file containing python definitions and statements

These modules are as listed below:

1. Main
2. ListSplit
3. dt
4. Display

5. Borrow
6. Return

Main

As the name implies, the main program contains the library management system's foundation. It imports data from all other modules, calls their functions as needed with the required parameters, and gathers information as needed.

It sets a variable true to execute the main loop indefinitely at first.

If the user selects an option 1 it Displays book available to borrow.

If the user selects an option 2 the book borrowing procedure proceeds.

If the user selects an option 3 the return procedure is continued.

If the user selects an option, the application will end.

If the user selects something other than the options listed above, the software shows a notice and continues to run while waiting for legitimate user input.

The main program delegated control of the execution flow to several smaller modules, which take user input, validate it, perform transactions, and write to the database file.

Display

It consists of a function which reads form book stock database file and displays the contents of the file with some formatting. It takes no arguments and returns no value

Listsplit

. The listsplit function reads the contents from the book stock database file and returns a two-dimensional list with book details. Here global keyword is used. The global keyword allows a user to change a variable outside of the current scope.... Only when we wish to make assignments or modify a variable do we utilize the global keyword inside a function

dt

It consists of a function which give time and date for the borrowed or return book to generate a bill for borrow process or return process

Borrow

The index of the book requested by the user, as well as the contents of the file, are provided as parameters in this module. The book's volume is then decreased, and a Boolean with new contents is returned. It’s also generated a bill with name of the borrower, borrowed date and time, borrowed book.

Return

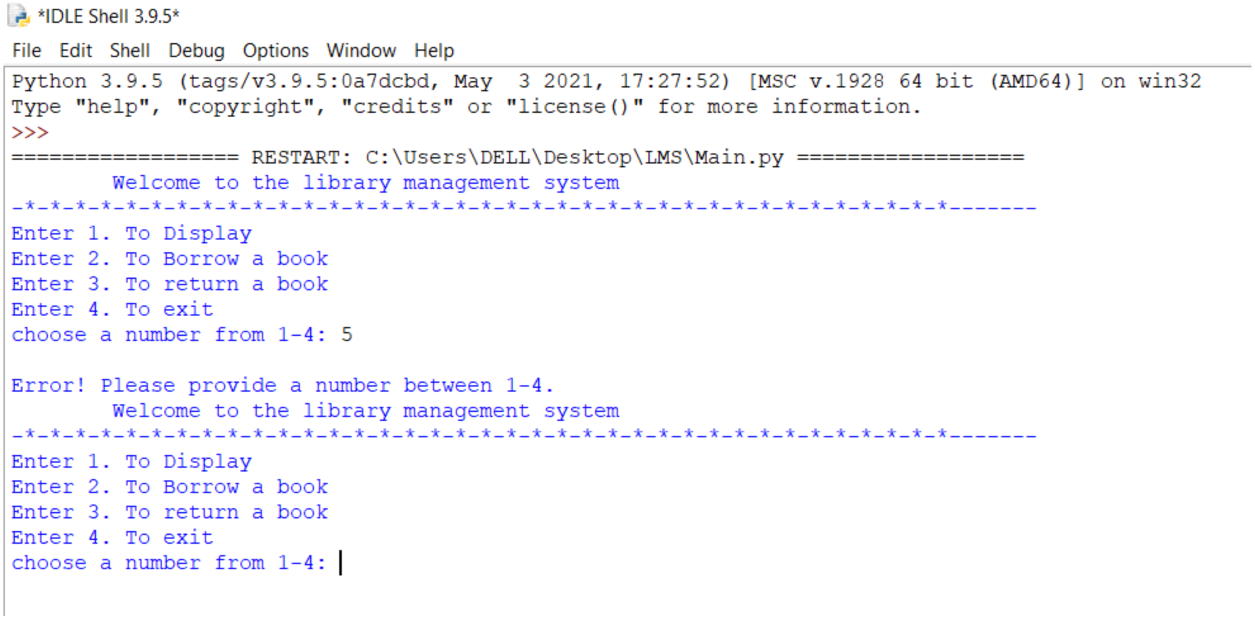
The index of the book requested by the user, as well as the contents of the file, are provided as parameters in this module. After that, the book's quantity is raised, and a Boolean with new contents is returned.

5. Tests

Test 1

Objective	To Show implementation of try, except
Action	Invalid input was provided
Expected Result	Safely generate appropriate message then return to main menu
Actual Result	A message “Error! Please provide a number between 1-4” was shown
Conclusion	Passed successfully

Table 1: Test 1



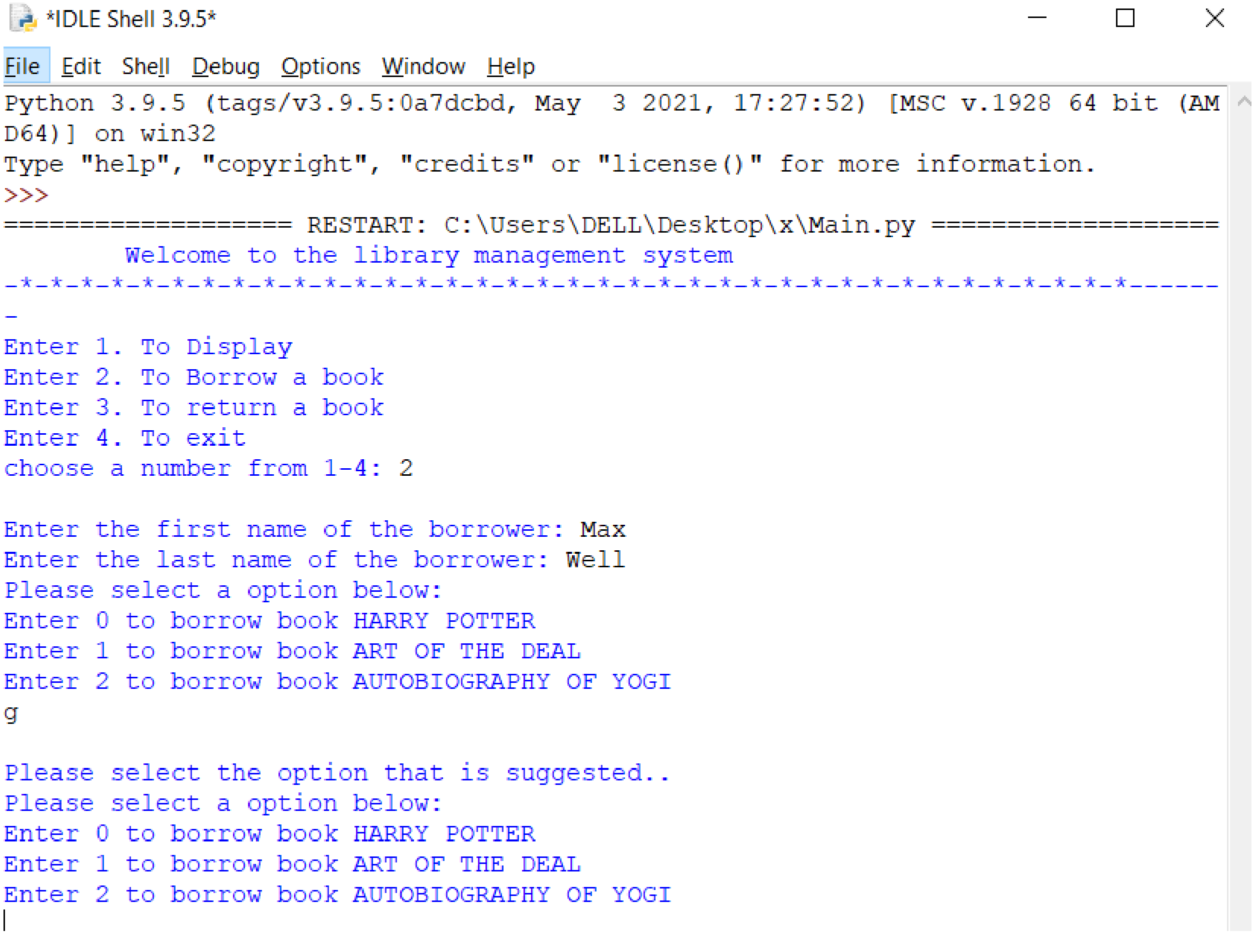
```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
================== RESTART: C:\Users\DELL\Desktop\LMS\Main.py ==================
        Welcome to the library management system
-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-------
Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
choose a number from 1-4: 5

Error! Please provide a number between 1-4.
        Welcome to the library management system
-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-------
Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
choose a number from 1-4: 
```

Figure 7 :To Show implementation of try, except

Test 2

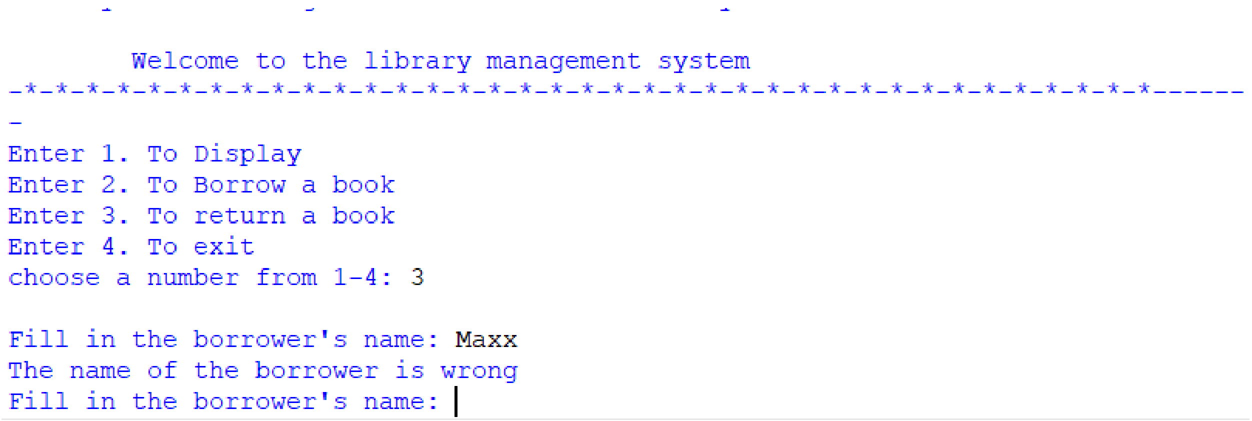
Objective	To Show implementation of Selection borrow and return option
Action	Non existed value was provided as input for borrow selection and return option
Expected Result	Safely generate appropriate message then return to Selection borrow for Selection borrow Safely generate appropriate message then return to return option for return option
Actual Result	A message “Please select the option that is suggested.” was shown for Selection borrow A message “The name of the borrower is wrong “was shown for Return option
Conclusion	Passed successfully

Table 2: Test 2



```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
=================== RESTART: C:\Users\DELL\Desktop\x\Main.py ===================
        Welcome to the library management system
-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*------
-
Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
choose a number from 1-4: 2

Enter the first name of the borrower: Max
Enter the last name of the borrower: Well
Please select a option below:
Enter 0 to borrow book HARRY POTTER
Enter 1 to borrow book ART OF THE DEAL
Enter 2 to borrow book AUTOBIOGRAPHY OF YOGI
g

Please select the option that is suggested..
Please select a option below:
Enter 0 to borrow book HARRY POTTER
Enter 1 to borrow book ART OF THE DEAL
Enter 2 to borrow book AUTOBIOGRAPHY OF YOGI
```

Figure 8: Providing non existing value for borrow selection



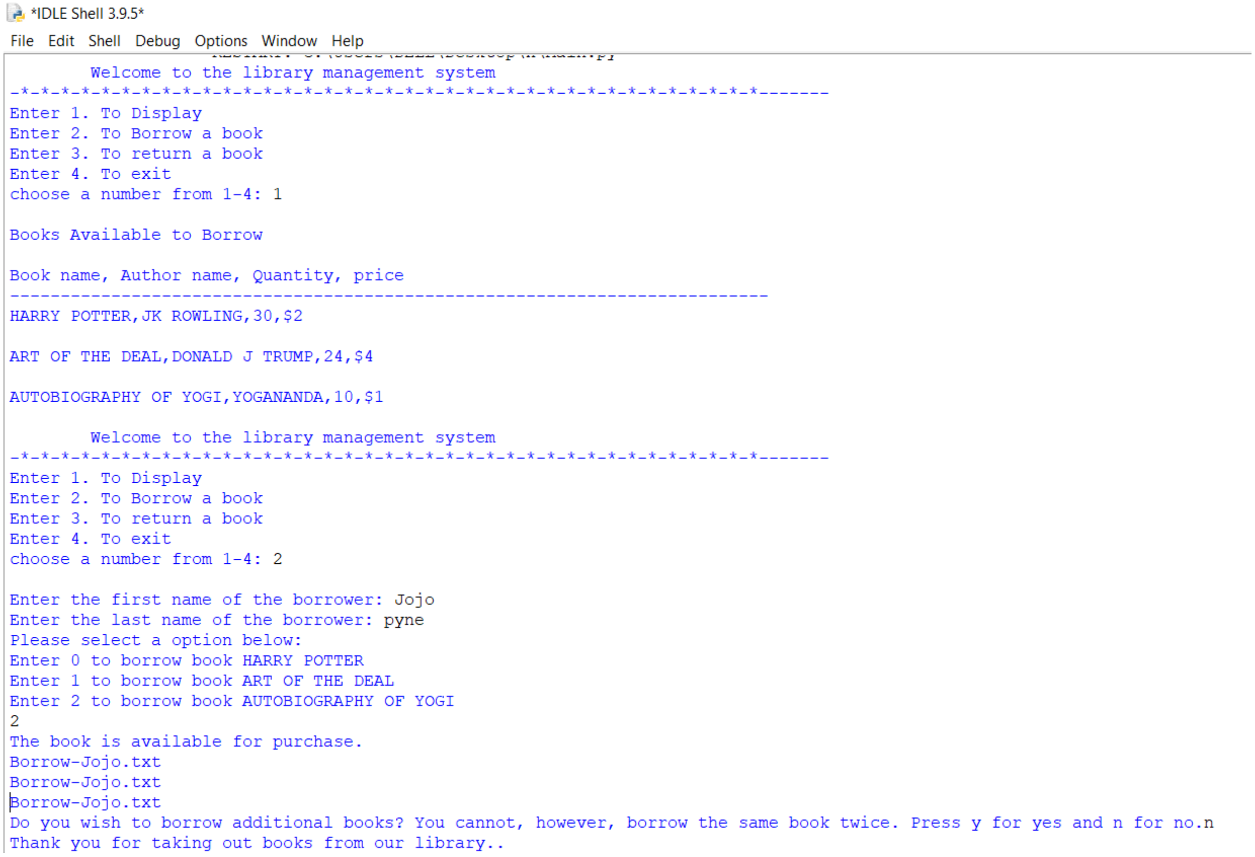
```
        Welcome to the library management system
-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*------
-
Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
choose a number from 1-4: 3

Fill in the borrower's name: Maxx
The name of the borrower is wrong
Fill in the borrower's name: 
```

Figure 9: Providing non existing value for Return option

Test 3:

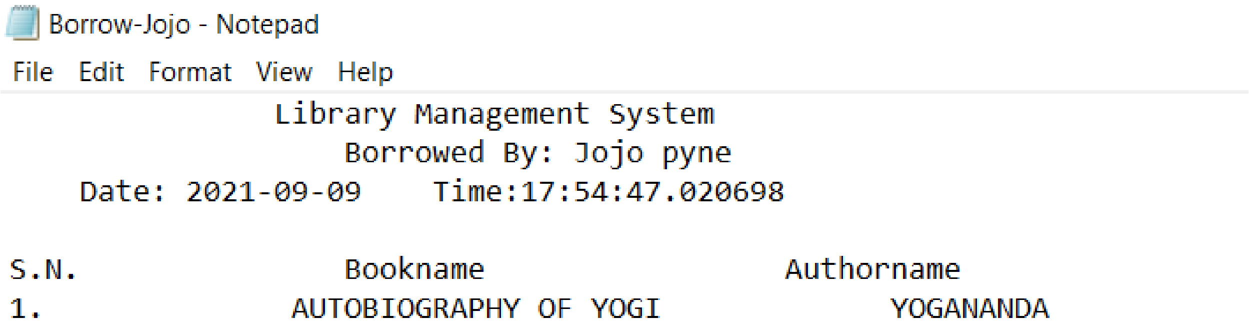
Objective	To Show File generation of borrow
Action	-Provide input 2 for borrow option -Input borrowers first and last name -Input existing value for book selection
Expected Result	File should be generated for borrow in text form.
Actual Result	File is generated for borrow in text form
Conclusion	Passed successfully

Table 3: Test 3



```
        Welcome to the library management system
-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-------
Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
choose a number from 1-4: 1

Books Available to Borrow

Book name, Author name, Quantity, price
---------------------------------------------------------------------------
HARRY POTTER,JK ROWLING,30,$2

ART OF THE DEAL,DONALD J TRUMP,24,$4

AUTOBIOGRAPHY OF YOGI,YOGANANDA,10,$1

        Welcome to the library management system
-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-------
Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
choose a number from 1-4: 2

Enter the first name of the borrower: Jojo
Enter the last name of the borrower: pyne
Please select a option below:
Enter 0 to borrow book HARRY POTTER
Enter 1 to borrow book ART OF THE DEAL
Enter 2 to borrow book AUTOBIOGRAPHY OF YOGI
2
The book is available for purchase.
Borrow-Jojo.txt
Borrow-Jojo.txt
Borrow-Jojo.txt
Do you wish to borrow additional books? You cannot, however, borrow the same book twice. Press y for yes and n for no.n
Thank you for taking out books from our library..
```

Figure 10 : Book borrowing process



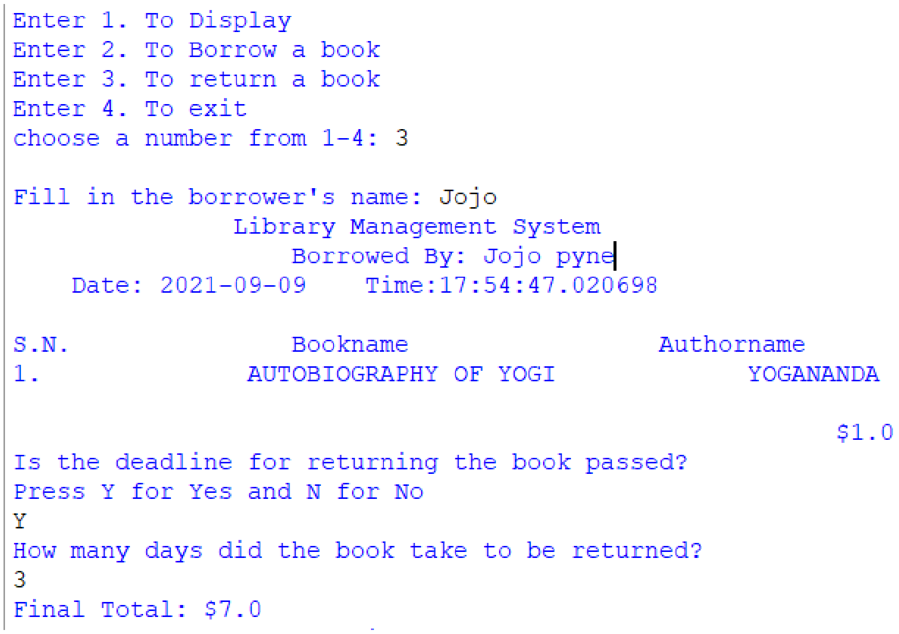
```
                Library Management System
                    Borrowed By: Jojo pyne
    Date: 2021-09-09    Time:17:54:47.020698

S.N.                Bookname                 Authorname
1.               AUTOBIOGRAPHY OF YOGI             YOGANANDA
```

Figure 11: Bill File generation of borrow

Test 4

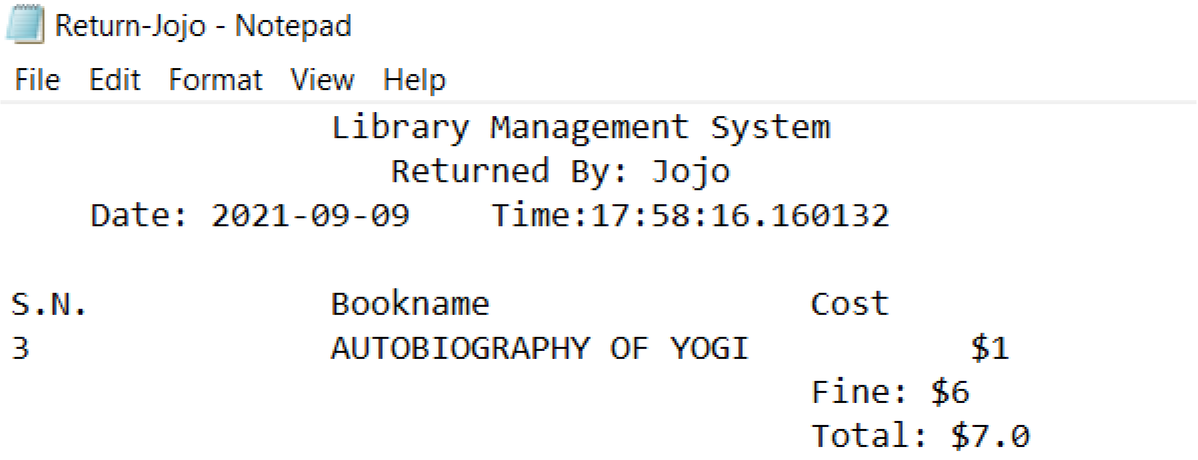
Objective	To Show File generation of Return
Action	-Provide input 3 for Return option -Input existing borrowers first and last name.
Expected Result	File should be generated for return in text file.
Actual Result	File is generated for return in text form
Conclusion	Passed successfully

Table 4:Test 4



```
Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
choose a number from 1-4: 3

Fill in the borrower's name: Jojo
               Library Management System
                   Borrowed By: Jojo pyne
    Date: 2021-09-09    Time:17:54:47.020698

S.N.                  Bookname                 Authorname
1.                 AUTOBIOGRAPHY OF YOGI             YOGANANDA

                                                           $1.0
Is the deadline for returning the book passed?
Press Y for Yes and N for No
Y
How many days did the book take to be returned?
3
Final Total: $7.0
```

Figure 12: Book return process



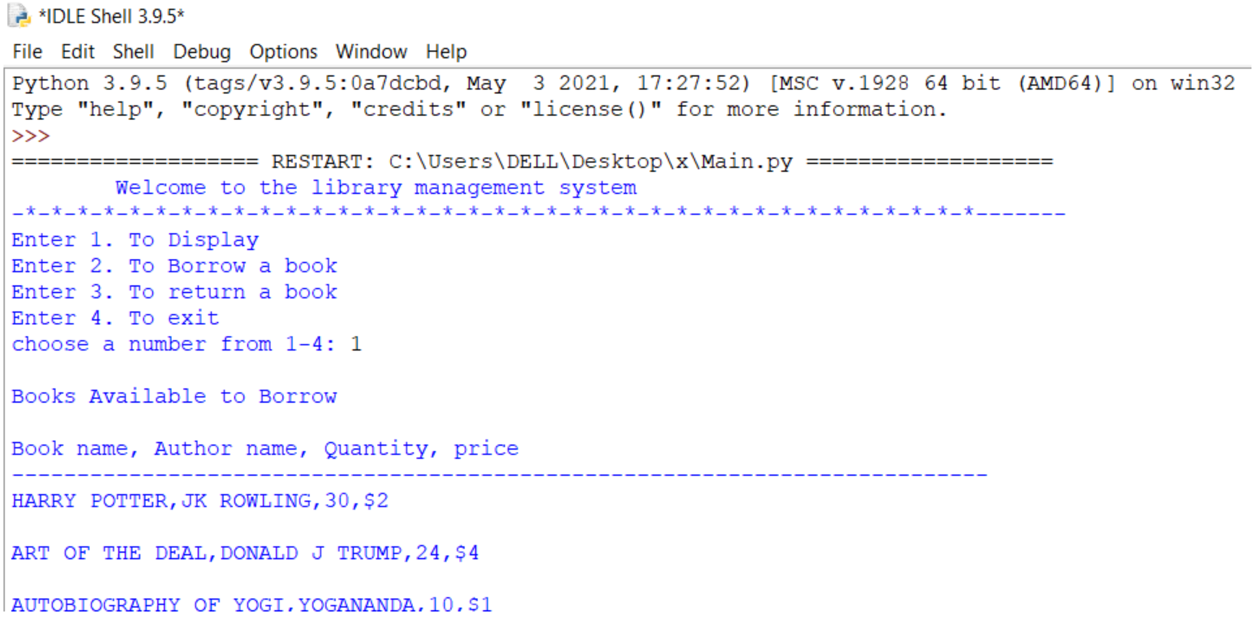
```
                Library Management System
                   Returned By: Jojo
    Date: 2021-09-09    Time:17:58:16.160132

S.N.            Bookname                Cost
3               AUTOBIOGRAPHY OF YOGI           $1
                                        Fine: $6
                                        Total: $7.0
```

Figure 13:Bill File generation for Return of borrowed book

Test 5

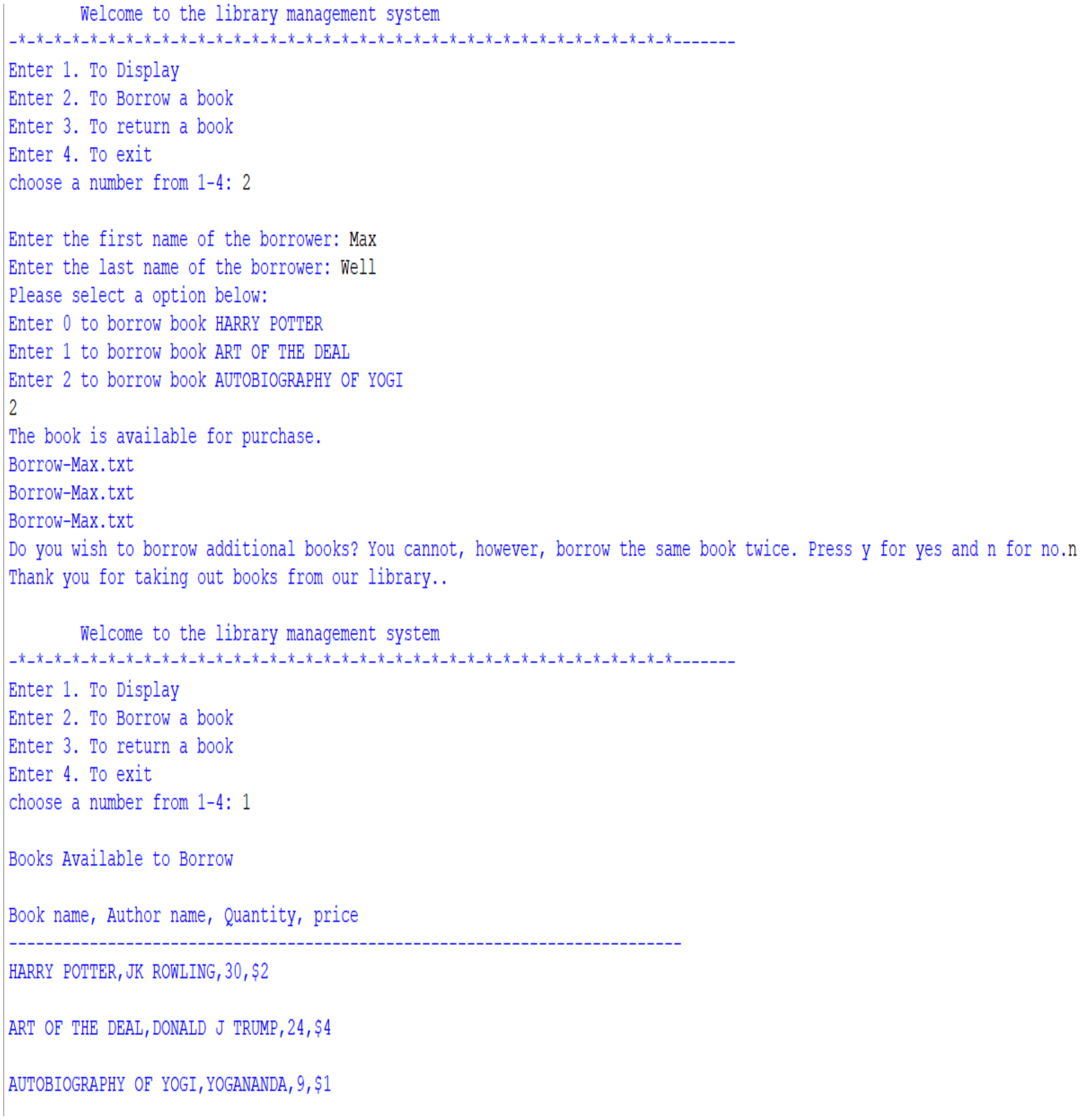
Objective	To Show the update in stock
Action	-borrow process is implemented -return process is implemented
Expected Result	Quantity of book borrowed should be reduced Quantity of book returned should be added
Actual Result	Quantity of book borrowed is reduced Quantity of book returned is added
Conclusion	Passed successfully

Table 5: Test 5



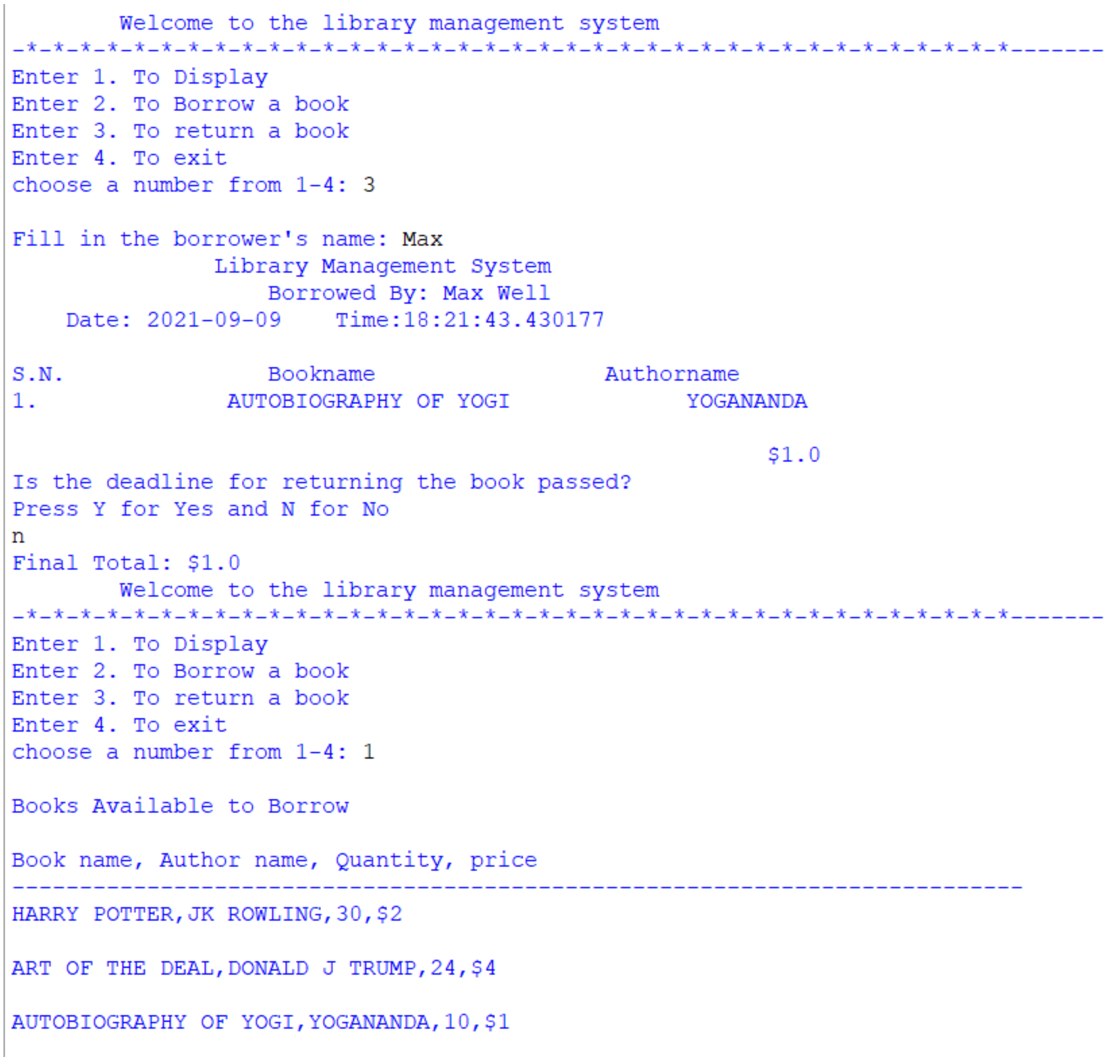
```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
=================== RESTART: C:\Users\DELL\Desktop\x\Main.py ===================
        Welcome to the library management system
-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-------
Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
choose a number from 1-4: 1

Books Available to Borrow

Book name, Author name, Quantity, price
---------------------------------------------------------------------------
HARRY POTTER,JK ROWLING,30,$2

ART OF THE DEAL,DONALD J TRUMP,24,$4

AUTOBIOGRAPHY OF YOGI,YOGANANDA,10,$1
```

Figure 14: Display Book available to borrow



```
        Welcome to the library management system
-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-------
Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
choose a number from 1-4: 2

Enter the first name of the borrower: Max
Enter the last name of the borrower: Well
Please select a option below:
Enter 0 to borrow book HARRY POTTER
Enter 1 to borrow book ART OF THE DEAL
Enter 2 to borrow book AUTOBIOGRAPHY OF YOGI
2
The book is available for purchase.
Borrow-Max.txt
Borrow-Max.txt
Borrow-Max.txt
Do you wish to borrow additional books? You cannot, however, borrow the same book twice. Press y for yes and n for no.n
Thank you for taking out books from our library..

        Welcome to the library management system
-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-------
Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
choose a number from 1-4: 1

Books Available to Borrow

Book name, Author name, Quantity, price
---------------------------------------------------------------------------
HARRY POTTER,JK ROWLING,30,$2

ART OF THE DEAL,DONALD J TRUMP,24,$4

AUTOBIOGRAPHY OF YOGI,YOGANANDA,9,$1
```

Figure 15: Showing Quantity of book borrowed is reduced



```
        Welcome to the library management system
-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-------
Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
choose a number from 1-4: 3

Fill in the borrower's name: Max
               Library Management System
                   Borrowed By: Max Well
    Date: 2021-09-09    Time:18:21:43.430177

S.N.               Bookname                 Authorname
1.              AUTOBIOGRAPHY OF YOGI             YOGANANDA

                                                        $1.0
Is the deadline for returning the book passed?
Press Y for Yes and N for No
n
Final Total: $1.0
        Welcome to the library management system
-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-------
Enter 1. To Display
Enter 2. To Borrow a book
Enter 3. To return a book
Enter 4. To exit
choose a number from 1-4: 1

Books Available to Borrow

Book name, Author name, Quantity, price
---------------------------------------------------------------------------
HARRY POTTER,JK ROWLING,30,$2

ART OF THE DEAL,DONALD J TRUMP,24,$4

AUTOBIOGRAPHY OF YOGI,YOGANANDA,10,$1
```

Figure 16: Showing Quantity of book returned is added

6. Conclusion

The task assigned in the course work was completed with lots of effort and hardships. Tons of research and analysis were conducted to get through each steps of the work. Routine meetup with teachers were conducted where progressive work was shown, and valuable suggestions were undertaken. With iteration of above steps again and again finally an executable program performing assigned task was developed

This course work helped a lot in shaping a junior programmer in real programming world. Where a problem is to be solved or set of tasks is to be accomplished with a computer software. This task has sharpened the self-research and analysis skills to carve steps for developing a program, selecting the best way to develop the program, debugging errors generated during development and regular follow-up with senior developer to avoid deviation from main path.

The course work also required a proper documented report of all the task performed during development of the program. As documentation of any project helps to overview and understand the project, it was very useful and helpful in building our skill of preparing report in a standardized format accepted globally. Time management and working accordingly to complete assigned task before deadline was one of the most important lesson learned from this course work. The course work was a hefty task to be completed within three weeks of time. All required research, analysis, planning, development, testing and documenting the process was to be done with in that time frame. It was a very tight schedule as course work of other subject were also to be completed in those same weeks

To sum up, the first course work of information system where a library management system was to be developed using python programming language was successfully completed. There were many valuable lessons learned during the program development process. All in all, this course work gave a sneak peak of what life of a future programmer is going to be like.

7. Appendix

Main

```
import Return
import List Split
import dt
import Borrow
import Display

def start screen ():
    while(True):
        print("        Welcome to the library management system     ")
        print("-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-------")
        print("Enter 1. To Display")
        print("Enter 2. To Borrow a book")
        print("Enter 3. To return a book")
        print("Enter 4. To exit")
        try:
            a=int(input("choose a number from 1-4: "))
            print()
            if(a==1):
                print ("Books Available to Borrow\n")
                print ("Book name, Author name, Quantity, price")
                print("-------------------------------------------------------------------------")
                Display.display()

            elif(a==2):
                ListSplit.listSplit()
                Borrow.borrowBook()
            elif(a==3):
                ListSplit.listSplit()
                Return.returnBook()
            elif(a==4):
                print("Thank you for making use of the library management system.")
                break
            else:
                print("Error! Please provide a number between 1-4.")
        except ValueError:
            print("Please enter the information as given..")
start_screen()
```

ListSplit

```
def listSplit():
    global bookname
    global authorname
    global quantity
    global cost
    bookname=[]
    authorname=[]
    quantity=[]
    cost=[]
    with open("Bookstocks.txt","r") as mf:

        lines=mf.readlines()
        lines=[x.strip('\n') for x in lines] # strip() method removes any spaces or specified
characters at the start and end of a string
        for i in range(len(lines)):
            ind=0
            for a in lines[i].split(','):
                if(ind==0):
                    bookname.append(a)
                elif(ind==1):
                    authorname.append(a)
                elif(ind==2):
                    quantity.append(a)
                elif(ind==3):
                    cost.append(a.strip("$"))
                ind+=1
```

Display

```
def display():
    bookstocks=open('Bookstocks.txt','r')
    for line in bookstocks:
        print (line)
```

Dt

```
def getDate():
    import datetime
    now=datetime.datetime.now
    #print("Date: ",now().date())
    return str(now().date())


def getTime():
    import datetime
    now=datetime.datetime.now
    #print("Time: ",now().time())
    return str(now().time())
```

Borrow

```
import dt
import ListSplit

def borrowBook():
    success=False
    while(True):
        firstName=input("Enter the first name of the borrower: ")
        if firstName.isalpha(): # isalpha() method returns the Boolean value True if each character in a string is a letter
```

```
            break
        print("please input alphabet from A-Z")
    while(True):
        lastName=input("Enter the last name of the borrower: ")
        if lastName.isalpha():
            break
        print("please input alphabet from A-Z")

    t="Borrow-"+firstName+".txt"
    with open(t,"w+") as mf:
        mf.write("               Library Management System  \n")
        mf.write("                   Borrowed By: "+ firstName+" "+lastName+"\n")
        mf.write("    Date: " + dt.getDate()+"    Time:"+ dt.getTime()+"\n\n")
        mf.write("S.N. \t\t   Bookname \t            Authorname \n" )

    while success==False:
        print("Please select a option below:")
        for i in range(len(ListSplit.bookname)):
            print("Enter", i, "to borrow book", ListSplit.bookname[i])

        try:
            a=int(input())
            try:
                if(int(ListSplit.quantity[a])>0):
                    print("The book is available for purchase.")
                    with open(t,"a") as mf:
                        mf.write("1. \t\t"+ ListSplit.bookname[a]+"\t\t
"+ListSplit.authorname[a]+"\n")

                    ListSplit.quantity[a]=int(ListSplit.quantity[a])-1
                    with open("Bookstocks.txt","w+") as mf:
```

```
            for i in range(3):

mf.write(ListSplit.bookname[i]+","+ListSplit.authorname[i]+","+str(ListSplit.quantity[i])+","
+"$"+ListSplit.cost[i]+"\n")
              print(t)

          #multiple book borrowing code
          loop=True
          count=1
          while loop==True:
            choice=str(input("Do you wish to borrow additional books? You cannot,
however, borrow the same book twice. Press y for yes and n for no."))
            if(choice.upper()=="Y"):
              count=count+1
              print("Please choose an option from the list below.:")
              for i in range(len(ListSplit.bookname)):
                print("Enter", i, "to borrow book", ListSplit.bookname[i])
              a=int(input())
              if(int(ListSplit.quantity[a])>0):
                print("Book is available")
                with open(t,"a") as mf:
                  mf.write(str(count) +". \t\t"+ ListSplit.bookname[a]+"\t\t
"+ListSplit.authorname[a]+"\n")

                ListSplit.quantity[a]=int(ListSplit.quantity[a])-1
                with open("Bookstocks.txt","w+") as mf:
                  for i in range(3):

mf.write(ListSplit.bookname[i]+","+ListSplit.authorname[i]+","+str(ListSplit.quantity[i])+","
+"$"+ListSplit.cost[i]+"\n")
                    success=False
```

```
                            else:
                                loop=False
                                break
                        elif (choice.upper()=="N"):
                            print ("Thank you for taking out books from our library.. ")
                            print("")
                            loop=False
                            success=True
                        else:
                            print("Please make your selection according to the instructions.")

                else:
                    print("This book is no longer available.")
                    borrowBook()
                    success=False
            except IndexError:
                print("")
                print("Please select a book based on its number..")
        except ValueError:
            print("")
            print("Please select the option that is suggested..")
```

Return

```
import ListSplit
import dt
def returnBook():
    name=input("Fill in the borrower's name: ")
    a="Borrow-"+name+".txt"
    try:
        with open(a,"r") as mf:
            lines=mf.readlines()
            lines=[a.strip("$") for a in lines]
        with open(a,"r") as mf:
            data=mf.read()
            print(data)
    except:
        print("The name of the borrower is wrong")
        returnBook()

    b="Return-"+name+".txt"
    with open(b,"w+")as mf:
        mf.write("                Library Management System \n")
        mf.write("                   Returned By: "+ name+"\n")
        mf.write("    Date: " + dt.getDate()+"    Time:"+ dt.getTime()+"\n\n")
        mf.write("S.N.\t\tBookname\t\tCost\n")



    total=0.0
    for i in range(3):
        if ListSplit.bookname[i] in data:
            with open(b,"a") as mf:
                mf.write(str(i+1)+"\t\t"+ListSplit.bookname[i]+"\t\t$"+ListSplit.cost[i]+"\n")
                ListSplit.quantity[i]=int(ListSplit.quantity[i])+1
            total+=float(ListSplit.cost[i])

    print("\t\t\t\t\t\t\t"+"$"+str(total))
    print("Is the deadline for returning the book passed?")
    print("Press Y for Yes and N for No")
    stat=input()
    if(stat.upper()=="Y"):
        print("How many days did the book take to be returned?")
        day=int(input())
        fine=2*day
        with open(b,"a")as mf:
            mf.write("\t\t\t\t\tFine: $"+ str(fine)+"\n")
        total=total+fine



    print("Final Total: "+ "$"+str(total))
    with open(b,"a")as mf:
        mf.write("\t\t\t\t\tTotal: $"+ str(total))


    with open("Bookstocks.txt","w+") as mf:
            for i in range(3):

mf.write(ListSplit.bookname[i]+","+ListSplit.authorname[i]+","+str(ListSplit.quantity[i])+","
+"$"+ListSplit.cost[i]+"\n")
```