



Module Code & Module Title

CS4051NT Fundamentals of Computing

Assessment Weightage & Type

60 % Individual Coursework -1

Year and Semester

2020-21 Spring, Semester

Student Name: Shishir Shobhan Dawadi

London Met ID: 20048506

Assignment Submission Date: 09-10-2021

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 mark of zero will be awarded.

Table of Contents

1. Introduction:	1
1.2 Aims and Objectives:	3
2. Algorithm:	4
3. Pseudocode:	6
4. Flowchart:	11
5. Data Structure:	12
5.1. Non-Primitive Data Structure:	12
5.1.1. List:	12
5.1.2. Dictionary:	12
5.1.3. Tuple:	13
5.2. Primitive Data Structure:	13
5.2.1. Integers:	13
5.2.2. Float:	13
5.2.3. String:	14
5.2.4. Boolean:	14
6. Program:	15
7. Testing:	22
7.1 Test 1:	22
7.2. Test 2:	23
7.3. Test 3:	24
7.4. Test 4:	26
7.5. Test 5.	27
7.6. Test 6:	29
7.7. Test 7:	30
7.8. Test 8:	32
7.9. Test 9:	34
8. Weekly Progress:	35
9. Research:	36
10. Conclusion:	39
11. References	40

List of Tables:

Table 1 Test to display the list of books.	22
Table 2 Test to add new book to the book list of library.....	23
Table 3 Test to borrow book from library.	24
Table 4 Test to borrow again by the same user.	26
Table 5 Test to check quantity decrease.....	27
Table 6 To Check billing system while returning book.	29
Table 7 Test to check quantity increase after returning book.....	30
Table 8 To check borrow and return details in different text file.....	32
Table 9 To check program terminates or not.....	34

List of Image:

Figure 1 PyCharm Image	1
Figure 2 IDLE Image.	2
Figure 3 Draw.io Image	2
Figure 4 Flowchart Image.....	11
Figure 5 List used in program.....	12
Figure 6 List used in program append.....	12
Figure 7 Integer used in program.....	13
Figure 8 String used in program.....	14
Figure 9 Boolean used in program	14
Figure 10 __init__ function.....	16
Figure 11 add function.....	16
Figure 12 ask function.....	17
Figure 13 book_divide function	18
Figure 14 borrow function.	18
Figure 15 borrow function1	19
Figure 16 returnBook function.....	20
Figure 17 returnBook1 function.....	20
Figure 18 main function.	21
Figure 19 import list.py in LibraryManagementSystem.py	21
Figure 20 show_books function	21
Figure 21 Test 1: Listing books available on library.....	22
Figure 22 To add new book to the book list.....	23
Figure 23 Displaying added new book in the b book list.....	24
Figure 24 Borrowing book from library.....	25
Figure 25 Image to show borrow again by same user.....	26
Figure 26 Quantity of book before borrowing.	27
Figure 27 Borrowing book to check quantity decrease.....	28
Figure 28 Quantity decrease after borrowing book.....	28
Figure 29 Image showing details in bill while returning book.....	29
Figure 30 Image showing successfully return book.....	30
Figure 31 Quantity Increases after returning book.....	31
Figure 32 New text file auto created in program containing folder	32
Figure 33 Image showing return details in text file	33
Figure 34 Image showing borrow details in text file.....	33
Figure 35 Image showing program terminated.....	34
Figure 36 Uses of Functions in Python	36
Figure 37 Image showing how to import multiple files in python.....	37
Figure 38 Error handling in python.	38

1. Introduction:

This project is about Library management system. It is a computerized system which keeps record of all the day to day work of library. This project has many different features like adding new books to book list, issue book, return book, and display book details. It has also a facility where readers can see issue date and return date along with the billing system. One of the main feature of this project is reader can check the available books, author name, available quantity and their price before searching them in library. This project is very helpful for librarian to maintain the library in the best way and also helps to reduce human efforts.

Library management system is a program which is designed to perform all the works of library. This program can be used in every kinds of library (i.e. home library, school library, public library etc.) to make the work of librarian easier and convenient. This program is very helpful for activities like book issue, return, circulation recording and stock checking. Library management system reduces the need of man power and lower operational costs of the library. It also saves time of user and librarian. This program helps librarian to easily answer the queries of user by just a click. It reduces the enormous time of a manual process within a minutes. This program helps to create smart library by systematically organizing books by their names and authors. It also helps to provide quick information about the books available in library.

I have used PyCharm Community Edition 2021.1.2 to write this program. This APP was very helpful for me because of its features of guidance.

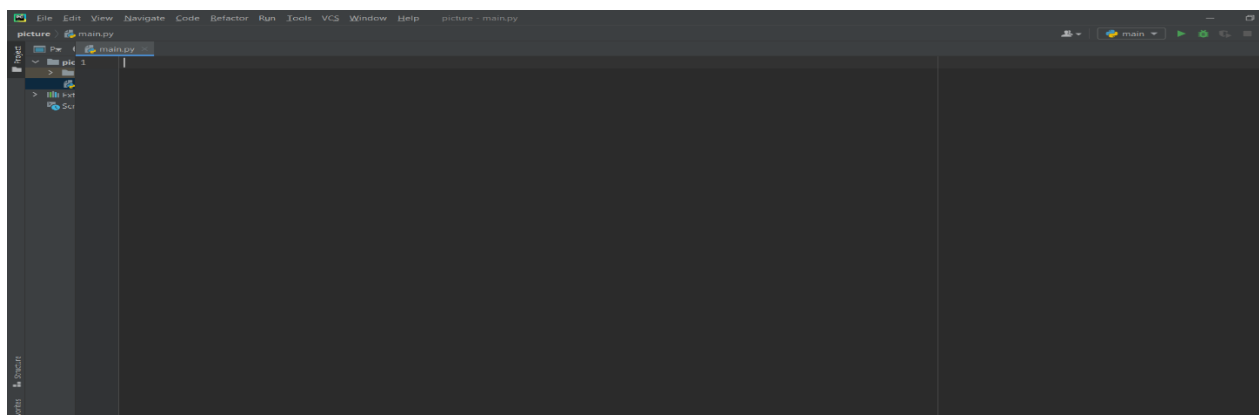


Figure 1 PyCharm Image

I also used IDLE (Python 3.9 64 – bit) which is python integrated development and learning environment.



Figure 2 IDLE Image.

And to create flowchart I have used draw.io.

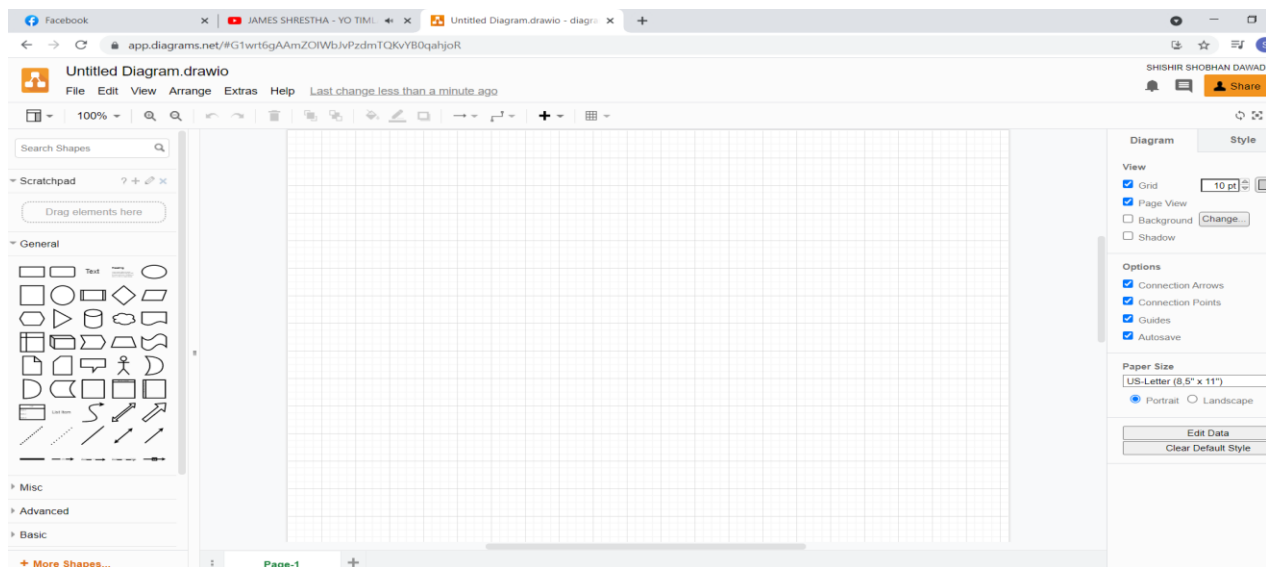


Figure 3 Draw.io Image

In this project I have created class named LMS and list to create user defined data structure to hold data members and member functions which are accessed and used while creating instance of LMS and list. All of my coding portion is written within the class. Also I have used different functions to bundle set of instruction which are used repeatedly and to call them when needed.

1.2 Aims and Objectives:

The main focus of this project is to develop Library management system so that it would be easier for the librarian to keep all the records of books and users. The purpose of creating Library management system is to store every single data related to the books available in library in computerized and efficient way. Aims and objectives of this project are listed below:

1. To provide details about the books available in library.
2. To make easy access to the readers.
3. To prevent loss of time in case of searching availability of particular books.
4. To keep the track record of circulation of books.
5. To make smart library.
6. To reduces the need of man power and lower operational costs.
7. To provide best service experience to readers.
8. To remove manual process of registering.
9. To make library more convenient.

2. Algorithm:

Algorithm can be defined as a set of instruction that are executed to solve the given problems. Algorithm are not language specific and can be implemented in different programming language (Tech Target Contributor, 2019). There are no standard rules for writing algorithm. Algorithm of my project is written below:

Step 1: Start

Step 2: Ask user what do you want to do.

- . if add, go to step 3.
- . if list, go to step 5.
- . if borrow go to step 8.
- . if return go to step 11
- . if quit go to step 16

Step 3: Ask user to input book name, author name, quantity and cost.

- . Display Added successfully.

Step 4: Go to step 2.

Step 5: Read text file containing all available books.

Step 6: Display available books.

Step 7: Go to step 2.

Step 8: Display books available.

Step 9: Ask user to input borrower first name, last name and book number.

- . If borrow again is Yes, ask to add book number other than previous.

Step 10: Display borrowed successfully message and ask user borrow again?

- . if No go to step 2.
- . if Yes go to step 8.

Step 11: Ask user to input Borrower Name.

- . if borrower name is invalid ask to input valid name.
- . if borrower name is valid Display Borrower Details.

Step 12: Check total number of days.

- . if total number of days is less than 10 go to Step 14
- . if total number of days is greater than 10 go to Step 13

Step 13: Add Total Cost + Fine and display Bill.

Step 14: Display Total cost and Bill.

Step 15: Go to step 2

Step 16: End

3. Pseudocode:

Pseudocode can be defined as the informal way of describing program. It does not need any strict programming language syntax. It is especially used for creating outline of a program. It also summaries the flow of the program. Pseudocode should be understood by a programmer to execute the program (TheEconomicTimes, 2021). Pseudocode of my project is written below:

START**IMPORT** datetime**IMPORT** list**CLASS** LMS**FUNCTION** init

self BookName []

self author []

self quantity []

self cost []

FUNCTION show_books **READ** text file containing available books **PRINT** available books**FUNCTION** addd **INPUT** name of the book **INPUT** name of the author **INPUT** total quantity of the book **INPUT** price for the book to borrow **ADD** book name, author, quantity and cost in book list. **DISPLAY** you have successfully added book to the book list**FUNCTION** ask **PRINT** welcome to the library management system.

menu prompt chose one of the following options.

INPUT user choices (add, list, borrow, return, quit) **WHILE** selected option is not quit

```
    IF selected option is add
        self addd
    ELIF selected option is list
        self show_books
    ELIF selected option is borrow
        self borrow
    ELIF selected option is return
        self returnBook
    ELSE
        PRINT selected option is invalid choose valid option
FUNCTION book_divide
    READ book list
    for i in range (len(lines))
        END = 0
        FOR a in lines [ i ]
            IF end == 0
                self bookname.append
            ELIF end == 1
                self authorname.append
            ELIF end == 2
                self quantity.append
            ELIF end == 3
                self cost.append
        END += 1
FUNCTION getDate
    RETURN date
FUNCTION getTime
    RETURN time
FUNCTION borrow
    WHILE true
        INPUT enter first name of the borrower
```

```
INPUT enter last name of the borrower
WHILE false
  FOR i in range (len(self bookname))
    PRINT i number for self bookname
  TRY
    INPUT enter the book number
    PRINT details
  TRY
    IF book quantity > 0
      PRINT book is available.
      WRITE book details in text file
      PRINT return book within given date
      self quantity = total book quantity -1
      PRINT book list
      FOR i in range (lens(self bookname))
        WRITE book name, author name, quantity, cost
    LOOP true
  count =1
  WHILE loop == true
    PRINT do you want to borrow again
    IF yes
      count = count+1
      PRINT select an option below
      FOR i in rage (lens(self.bookname))
        PRINT enter number of book you want to borrow
      IF quantity >0
        PRINT book is available.
        WRITE book information and decrease quantity of book
        PRINT book list
        FOR i in range (lens(self bookname))
          WRITE book name, author name, quantity, cost
```

```

                                TAKEN = false
                                ELSE
                                    loop = false
                                ELIF choice.upper
                                    PRINT you have successfully borrowed
                                    loop = false
                                    taken = true
                                ELSE
                                    PRINT enter the command as instructed
                                ELSE
                                    PRINT sorry book is not available
                                    taken = false
                                EXCEPT IndexError
                                    PRINT choose according to the number provided
                                EXCEPT ValueError
                                    PRINT chose a suggested to borrow book
FUNCTION returnBook
    INPUT enter the name of borrower
    TRY
        READ check name
        PRINT data
    EXCEPT
        PRINT name you enter is in correct
        WRITE user and time information
        FOR i in range(len(self.bookname))
            IF book name is in data
                WRITE book information
            total += cost
        PRINT total cost
        PRINT check whether the book return date is expired or not
        PRINT press y for yes and n for no
```

```
IF y
    PRINT enter the number of delayed days to return book
    fine = 2*day
    WRITE total cost
        total = total + fine
PRINT Total cost
    WRITE total information
    FOR i in range (len(self.bookname))
        WRITE Book Name, author, quantity, cost
FUNCTION main
    Library1 = LMS ()
    Library1.ask ()
main()
END
```

4. Flowchart:

Flowchart can be defined as a graphical or symbolical representation of a process which is represented by a different symbols containing the short description of the process. Flowchart symbols are linked together with the help of arrows which shows the direction flow of process (Lucidchart, 2021). Flowchart of my project is shown below:

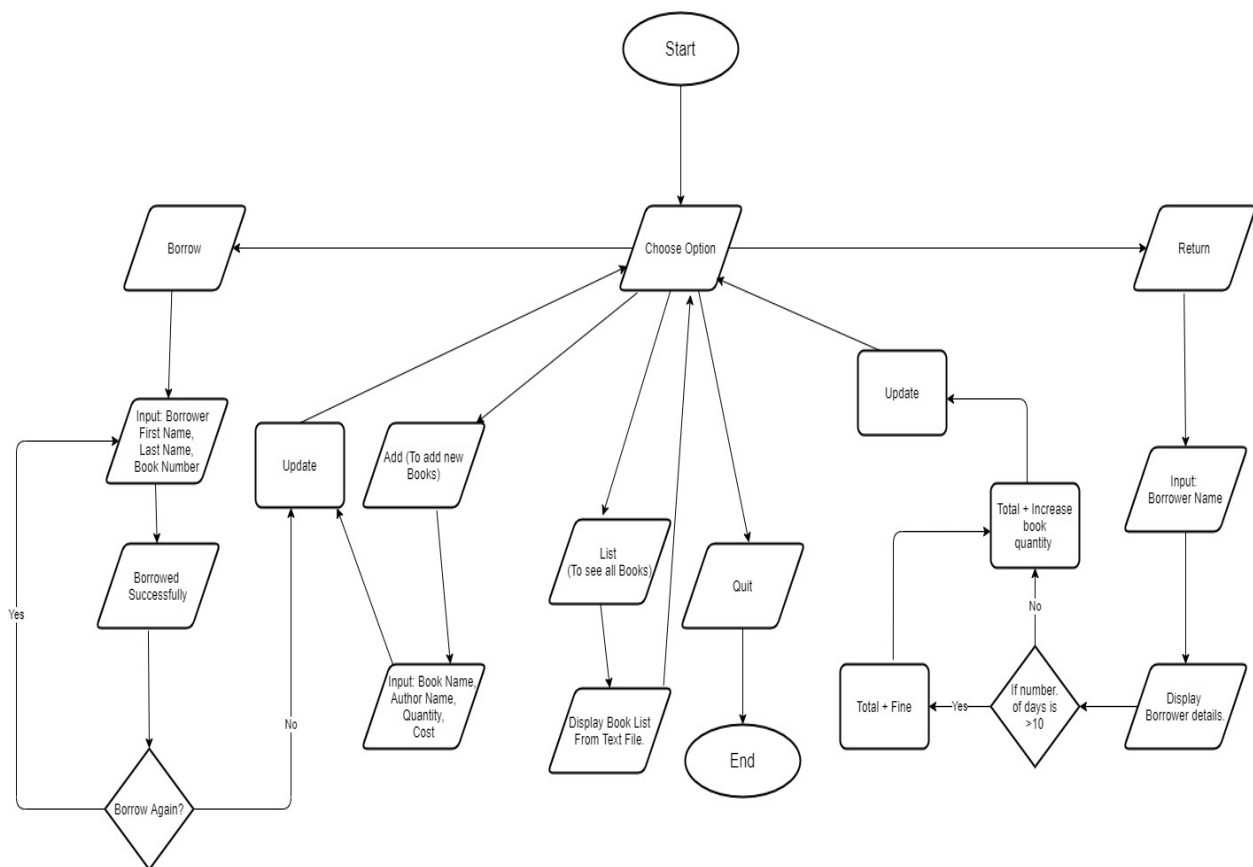


Figure 4 Flowchart Image.

5. Data Structure:

Data structures can be defined as the containers which organize and group data according to their type. Mutability and order creates difference in data structure.

In python there are two types of data structure i.e. Non-Primitive Data Structure and Primitive Data Structure.

5.1. Non-Primitive Data Structure:

There are four built-in Non-Primitive data structures in python i.e. List, Tuple, Dictionary, and Set.

5.1.1. List:

List are the data structure which stores ordered collection of items in python. Generally, list is represented within square brackets. List can have different data types and number of elements. Methods that are supported by list in python are append (), copy (), pop (), sort (), reverse (), and remove (). Example: [11,22,33," shishir",3.3]

In my project I have used List to append the user input to the book text file.

```
def __init__(self):  
    self.BookName = []  
    self.author = []  
    self.quantity = []  
    self.cost = []
```

Figure 5 List used in program.

```
for a in lines[i].split(','):   
    if (end == 0):   
        self.BookName.append(a)   
    elif (end == 1):   
        self.author.append(a)   
    elif (end == 2):   
        self.quantity.append(a)   
    elif (end == 3):   
        self.cost.append(a.strip("Rs."))   
    end += 1
```

Figure 6 List used in program append.

5.1.2. Dictionary:

Dictionary can be defined as a changeable, indexed collection and unordered. It stores data in key-value pairs form where key must be unique. Generally, dictionary is represented within curly brackets with the value and key separated by colon. The key value doesn't follow order in dictionary. Methods used in dictionary are items (), keys (), values (), copy (), clear (), and popitem (). Example: {11:12:13:'shishir'}

5.1.3. Tuple:

Tuple is also the built-in data structure in python as list. One of the most difference between list and tuple is their mutability. List are mutable i.e. they can be modified whereas tuple is immutable i.e. they can't be modified. Tuples are represented within parentheses and its items are separated by comma. Example: (11,22,33," sisir",111.2).

5.1.4. Set:

Set can be defined as unordered and unindexed collection which are written within curly braces. Sets cannot have multiple occurrence of the same element and stores unordered values. Example: {1,2,3,4,5}

5.2. Primitive Data Structure:

Primitive data structures are the building blocks for data manipulation and contain pure, general values of a data. There are four primitive variables in python i.e. Integers, Float, Strings, and Boolean.

5.2.1. Integers:

Integers are used to represent numeric data in python. More specifically it is used to represent whole numbers (numbers might be either negative or positive). Example: 11,12, -13,10, -10 etc.

In my project I have used Integer datatype to reduce the quantity of the books when readers borrow book from the library.

```
try:
    if (int(self.quantity[self.book]) > 0):
        print("Book you want to borrow is available for you. ")
        with open(t, "a") as f:
            f.write("1. \t\t" + self.bookname[self.book] + "\t\t" + self.authername[self.book] + "\n")
        print("Please return the book within " + str(X))
        self.quantity[self.book] = int(self.quantity[self.book]) - 1
        with open("book_list.txt", "w") as f:
            for i in range(len(self.bookname)):
                f.write(
                    f"{self.bookname[i]},{self.authername[i]},{str(self.quantity[i])},{self.cost[i]}\n")
```

Figure 7 Integer used in program

5.2.2. Float:

In python float is used for rational numbers which usually ends with a decimal figure. Example: 2.00, 20.00 etc.

5.2.3. String:

String can be defined as the collection of alphabets, words or any other characters. In python strings are written by enclosing within a pair of single or double quotes.

In my project I have used String data type to convert numeric value into string while displaying bill to the user.

```
if (stat.upper() == "Y"):
    print("Enter the number of delayed days to return book.")
    day = int(input())
    fine = 2 * day
    with open(b, "a") as f:
        f.write("\t\t\t\t\tFine: $" + str(fine) + "\n")
    total = total + fine

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

Figure 8 String used in program.

5.2.4. Boolean:

Boolean data type is used to take up the values either true or false. Boolean are used in conditional and comparison expressions (Sejal Jaiswal , 2017).

In my project I have used Boolean data type to ensure when user wish to borrow book from library, user input no other than alphabet. If user input any other than alphabet, then the program will display message saying please input alphabet from A-Z. If user don't follow the guidance, then he/she can't borrow the book and program will display book list

```
while (True):
    firstName = input("Enter the first name of the borrower: ")
    if firstName.isalpha():
        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")

while taken == False:
    for i in range(len(self.BookName)):
        print(i, " number for ", self.BookName[i])
```

Figure 9 Boolean used in program

6. Program:

The project Library Management System was new for me and requires lots of hard work, and research. At first I was excited and also worried about this project as I possess very low programming skills. Hence, I spend most of time in research and asking questions about related topic with my teacher and friends in order to get prepared for this program.

After going through multiple websites, article, books and taking advice from my teacher, I was prepared to roughly sketch my program. At first I wrote the algorithm, pseudocode and draw the flowchart of the program which helped me to start the coding portion. I also faced lots of problems and error while doing this project.

I created python file named Libirary_Management_System.py and done by coding portion within class LMS. Doing my coding portion within class provides easy way of keeping the data members and methods together also helps to make program more organized. It makes easier to reuse the code where needed and provides clean structure to the code.

I have also used multiple class and functions in order to divide large programs into small groups which helped me to read the code, and to debug the program faster and conveniently. Using multiple functions helped me to bind the different logic within one function and allows to call them where needed and also helps to prevent the repetition of the same code.

A brief explanation of my program (classes used and multiple functions) which will help viewers to understand this project more efficiently is done below. To provide in depth details I have explained every classes and functions and their uses along with their appropriate screen shoots. This programs consist two modules, they are listed below:

1. LibraryManagementSystem.py
2. list.py

I have import list.py in LibraryManagementSystem.py and inside these two different classes I have created multiple functions which are explained in details below:

1. LibraryManagementSystem.py:

This is the main module of this project where datetime, list.py are imported to retrieve data and functions from those modules. Almost all the functions are inside this module and entire execution depends on this module. Here inside class LMS firstly, I have created multiple functions according to their uses.

- **__init__ function:**

```
import datetime
import list
class LMS:
    def __init__(self):
        self.BookName = []
        self.author = []
        self.quantity = []
        self.cost = []
```

Figure 10 __init__ function

As shown in above figure __init__ functions store the BookName, author, quantity and cost in a List. Although we can use dictionary here but it could create a problem while storing details. Hence __init__ function stores the information in List.

- **addd function:**

```
def addd(self):
    booknames = input("Enter the name of the book you want to add: ")
    author = input("Enter the name of the author: ")
    quantity = int(input("Enter the total quantity of the book: "))
    cost = input("Enter the price for the book to borrow: ")
    with open("book_list.txt", "a") as adding:
        adding.write("\n")
        adding.write(f"{booknames},{author},{str(quantity)},{cost}")

    print("You have successfully added book on the book list.")
```

Figure 11 addd function.

As shown in above figure addd functions stores the new book in the text file storing book information. This function allows user to input name of the book, author, total quantity of

the book and its borrowing cost and stores all of those information in book list showing appropriate message.

- **ask function:**

```
def ask(self):
    print("Welcome To Libirary Management System")

    menu_prompt = """Please choose one of the following options."

    -Type 'a' to add a book in Book List
    -Type 'b' to borrow books
    -Type 'r' to return borrowed books
    -Type 'q' to quit the program
    what would you like to do : """

    selected_option = input(menu_prompt).strip().lower()
    while selected_option != "q":
        if selected_option == "a":
            self.add()
        elif selected_option == "l":
            list.show_books()
        elif selected_option == "b":
            self.borrow()
        elif selected_option == "r":
            self.returnBook()
        else:
            print(f"Sorry, '{selected_option}' is a invalid option, please choose a valid option")

    selected_option = input(menu_prompt).strip().lower()
```

Figure 12 ask function.

As shown in above figure ask function consist menu prompt which ask user what they want to do. Firstly, this function shows a welcome message and ask users what they want to do. If user input “q” then the whole program will be terminated. If user input “a” then function addd will be working. If user input “l” then function list from list.py class will be working. If the user input “b” then the function borrow will be working and if the user input “r” then the function returnBook will be working. Finally, if the user input other than the options provided then it will display a message showing the user input and asking to enter the valid option.

- **book_divide function:**

```
def book_divide(self):
    with open("book_list.txt", "r") as f:
        lines = f.readlines()
        lines = [x.strip('\n') for x in lines]
        for i in range(len(lines)):
            end = 0
            for a in lines[i].split(','):
                if (end == 0):
                    self.BookName.append(a)
                elif (end == 1):
                    self.author.append(a)
                elif (end == 2):
                    self.quantity.append(a)
                elif (end == 3):
                    self.cost.append(a.strip("Rs."))
                end += 1

def getDate(self):
    now = datetime.datetime.now
    return str(now().date())

def getTime(self):
    now = datetime.datetime.now
    return str(now().time())
```

Figure 13 book_divide function

As shown in above figure, book_divide function stores name of book in the first position, author name in second position, quantity in third position and cost in fourth position and shows accordingly in book list. Function getDate is used to return current date and getTime is used to return current time while borrowing and returning the book from the library.

- **borrow function:**

```
def borrow(self):
    self.book_divide()
    taken = False
    Date = datetime.date.today()
    delaydate = datetime.timedelta(days=10)
    X = str(Date + delaydate)
    while (True):
        firstName = input("Enter the first name of the borrower: ")
        if firstName.isalpha():
            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")
    while taken == False:
        for i in range(len(self.BookName)):
            print(i, " number for ", self.BookName[i])
        try:
            self.book = int(input("enter book number: "))
            t = "Borrow-" + firstName + ".txt"
            with open(t, "w") as f:
                f.write("Library Management System \n")
                f.write("Borrowed By: " + firstName + " " + lastName + "\n")
                f.write("Borrowed Date: {self.getDate()} Time : {self.getTime()} Returning Date : {X}\n\n")
                f.write("S.N. \t\t Bookname \t\t Authorname \n")
            try:
                if (int(self.quantity[self.book]) > 0):
                    print("Book you want to borrow is available for you. ")
                    with open(t, "a") as f:
                        f.write("1. \t\t" + self.BookName[self.book] + "\t\t" + self.author[self.book] + "\n")
                    print("Please return the book within " + str(X))
                    self.quantity[self.book] = int(self.quantity[self.book]) - 1
                    with open("book_list.txt", "w") as f:
                        for i in range(len(self.BookName)):
                            f.write(
                                f"{self.BookName[i]}, {self.author[i]}, {str(self.quantity[i])}, {self.cost[i]}\n")
            except:
                print("Book not available for borrowing")
        except:
            print("Invalid book number")
    taken = True
```

Figure 14 borrow function.


```

loop = True
count = 1
while loop == True:
    choice = str(input(
        "Do you want to borrow another book? Remember you cannot borrow the same book again until you returned it. Press y for yes and n for no. "))
    if (choice.upper() == "y"):
        count = count + 1
        print("Please select an option below:")
        for i in range(len(self.BookName)):
            print("Enter", i, "to borrow book", self.BookName[i])
        self.book = int(input("Enter the book number you want to borrow: "))
        if (int(self.quantity[self.book]) > 0):
            print("Book you want to borrow is available for you.")
            with open(t, "a") as f:
                f.write(str(count) + ". \t\t" + self.BookName[self.book] + "\t\t" +
                    self.author[self.book] + "\n")
                self.quantity[self.book] = int(self.quantity[self.book]) - 1
            with open("book_list.txt", "w") as f:
                for i in range(len(self.BookName)):
                    f.write(
                        f"{self.BookName[i]},{self.author[i]},{str(self.quantity[i])},{self.cost[i]}\n")
                taken = False
            else:
                loop = False
                break
        elif (choice.upper() == "n"):
            print("You have successfully borrowed the books you want! Thank you for choosing our Library. ")
            print("")
            loop = False
            taken = True
        else:
            print("Please enter the command as instructed")

    else:
        print("Sorry !Book is not available, please wait for some days.")
        taken = False
except IndexError:
    print("")
    print("Please choose book according to the number provided.")
except ValueError:
    print("")
    print("Please choose as suggested to borrow books you want.")

```

Figure 15 borrow function1

As shown in above figure, borrow function stores the information of the user who have borrowed the book from library along with the date and time. At the beginning of this function user requires to input their first name and last name. If the user input other than alphabet while writing their first and last name, then the appropriate message is shown to the user saying please input alphabet. After this the user is asked to enter the number of book they want to borrow. If the user input the actual book number and is available in the stock, then a message is shown to the user saying you have successfully borrowed along with returning date and all of those information is stored in new text file. After successfully borrowing book quantity of the book is decrease by -1. And if the quantity of book is out of stock then the user will be informed showing an appropriate message. After successfully borrowing the book user is again asked if you want to borrow another book. If user enter yes, then again the book list is displayed to the user asking please enter the book number you want to borrow and the information is saved in user text file. But if user enter no, then the message is displayed to the user saying thank you have successfully borrowed and thanks for choosing our library.

- **returnBook function:**

```
def returnBook(self):
    self.book_divide()
    name = input("Enter the name of borrower: ")
    a = "Borrow-" + name + ".txt"
    try:
        with open(a, "r") as f:
            lines = f.readlines()
            lines = [a.strip("$") for a in lines]
        with open(a, "r") as f:
            data = f.read()
            print(data)
    except:
        print("The name of the borrower you entered is incorrect. Please, make sure you entered correctly.")
        self.returnBook()

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

    total = 0.0
    for i in range(len(self.BookName)):
        if self.BookName[i] in data:
            with open(b, "a") as f:
                f.write(str(i + 1) + "\t\t" + self.BookName[i] + "\t\t" + "$" + self.cost[i] + "\n")
                self.quantity[i] = int(self.quantity[i]) + 1
                total += int(self.cost[i])
```

Figure 16 returnBook function

```
print("\t\t\t\t\t" + "$" + str(total))
print("Check whether the book return date is expired or not.")
print("Press Y for Yes and N for No")
stat = input()
if (stat.upper() == "Y"):
    print("Enter the number of delayed days to return book.")
    day = int(input())
    fine = 2 * day
    with open(b, "a") as f:
        f.write("\t\t\t\t\tFine: $" + str(fine) + "\n")
    total = total + fine

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

with open("book_list.txt", "w") as f:
    for i in range(len(self.BookName)):
        f.write(f"{self.BookName[i]}, {self.author[i]}, {str(self.quantity[i])}, {self.cost[i]}\n")

def main():
    Library1 = LMS()
    Library1.ask()
main()
```

Figure 17 returnBook1 function

In function returnBook firstly book_divide function is called to compare the name of the borrower and books, then user is asked to input the name of the borrower. If the user input invalid name of the borrower, then the message is displayed saying make sure you entered correct name of the borrower. If the user input valid name, then the user details

is displayed along with borrowed date and time. After returning the book quantity of the book is increased by +1. And then the user is asked to check whether the return date is expired or not. If user input yes, then he/she is fined and total price is displayed in the bill and if the user input no then his/her details along with total price is showed in the bill.

- **main function:**

```
def main():  
    Library1 = LMS()  
    Library1.ask()  
main()
```

Figure 18 main function.

As shown in figure, main function is used to run the LibraryManagementSystem.py.

2. list.py:

list.py is especially created to show the modularity concept. list.py consist show_books function which display the book list stored in the text file. I have import this file in LibraryManagementSystem.py and can be used in that class also.

```
import datetime  
import list  
class LMS:
```

Figure 19 import list.py in LibraryManagementSystem.py

```
def show_books():  
    f = open("book_list.txt", "r")  
    print(f.read())
```

Figure 20 show_books function

7. Testing:

Black box testing can be defined as the process of testing system without knowing about its internal workings. This testing is completely based on software requirements and specifications. This type of testing helps to identify whether the program/system is responding correctly or not (imperva, 2021).

7.1 Test 1:

Table 1 Test to display the list of books.

Objective	To list the all books available in library.
Action	Run the module and type l in given option
Expected Result	List of all available books in the library should be displayed.
Result Obtained	List of all available books in the library was displayed.
Conclusion	Tested Successfully.

```

Welcome To Libirary Management System
Please choose one of the following options."

    -Type 'a' to add a book in Book List
    -Type 'b' to borrow books
    -Type 'r' to return borrowed books
    -Type 'l' to list the books available in library.
    -Type 'q' to quit the program
    what would you like to do : l
1. In Search of Lost Time, Marcel Proust, 35, $1
2. Ulysses, James Joyce, 30, $1.5
3. Don Quixote, Miguel de Cervantes,27, $1
4. One Hundred Years of Solitude, Gabriel Garcia Marquez, 20, $2
5. The Great Gatsby, F. Scocot Fitzgerald, 28, $1
6. The Art of Computer Programming, Donald Knuth,22, $1.5
7. Hackers: Herroes of Computer Revolution, Steven Levy, 22, $1
8. The Road Ahead, Bill Gates, 19, $1.5
9. Macintosh, Carol Kaehler and Apple, 26, $2
10. Alan Turing: The Enigma, Andrew Hodges, 22, $2.5
11. A Hero Born, Jin Yong,6, $1
12. DragonFlight, Anne McCaffrey, 16, $1.5
13. Christian Apologetics, Cornelius Van Til, 19, 2.5
14. The Problem of Pain, C.S. Lewis, 17, $2
15. Orthodoxy, G.K.Chesterton, 19, $1
16. Treasure Island, Robert, Louis Stevenson, 24
17. Kin, Rudyard Kipling,21, $1
18. The call of the Wild, Jack London, 14, $1.5
19. Lost Horizon, James Hilton, 13, $2.5
20. The warriors, Sol Yurick, 17, $1
21. Physica (Physics), Aristotle,19, $2
22. What is Life?, Erwin Schrodinger, 16, $2.5
23. To kill a Mockingbird, Harperlee, 20, $1.5
24. Beloved, Toni Morrison, 9, $2.5
25. The stranger, Albert Campus,8, 1

Please choose one of the following options."

    -Type 'a' to add a book in Book List
    -Type 'b' to borrow books
    -Type 'r' to return borrowed books
    -Type 'l' to list the books available in library.
    -Type 'q' to quit the program
    what would you like to do : l

```

Figure 21 Test 1: Listing books available on library.

7.2. Test 2:

Table 2 Test to add new book to the book list of library.

Objective	To add new book to the book list.
Action	Run module and type a. Enter name of the book, author, quantity and its cost.
Excepted Result	Book should be added in the book list.
Result Obtained	Book was added on the book list.
Conclusion	Successfully tested.

```

Please choose one of the following options."

    -Type 'a' to add a book in Book List
    -Type 'b' to borrow books
    -Type 'r' to return borrowed books
    -Type 'l' to list the books available in library.
    -Type 'q' to quit the program
    what would you like to do : a
Enter the name of the book you want to add: Every Thing Happens For Reason
Enter the name of the author: Shishir Shobhan
Enter the total quantity of the book: 20
Enter the price for the book to borrow: 2
You have successfully added book on the book list.
Please choose one of the following options."

    -Type 'a' to add a book in Book List
    -Type 'b' to borrow books
    -Type 'r' to return borrowed books
    -Type 'l' to list the books available in library.
    -Type 'q' to quit the program
    what would you like to do : l
  
```

Figure 22 To add new book to the book list.

```

-Type 'a' to add a book in Book List
-Type 'b' to borrow books
-Type 'r' to return borrowed books
-Type 'l' to list the books available in library.
-Type 'q' to quit the program
what would you like to do : 1
1. In Search of Lost Time, Marcel Proust, 35, 1
2. Ulysses, James Joyce, 30, 1.5
3. Don Quixote, Miguel de Cervantes, 27, 1
4. One Hundred Years of Solitude, Gabriel Garcia Marquez, 20, 2
5. The Great Gatsby, F. Scooot Fitzgerald, 28, 1
6. The Art of Computer Programming, Donald Knuth, 22, 1.5
7. Hackers: Herroes of Computer Revolution, Steven Levy, 22, 1
8. The Road Ahead, Bill Gates, 19, 1.5
9. Macintosh, Carol Kaehler and Apple, 26, 2
10. Alan Turing: The Enigma, Andrew Hodges, 22, 2.5
11. A Hero Born, Jin Yong, 6, 1
12. DragonFlight, Anne Mccaffrey, 16, 1.5
13. Christian Apologetics, Cornelius Van Til, 19, 2.5
14. The Problem of Pain, C.S. Lewis, 17, 2
15. Orthodoxy, G.K.Chesterton, 19, 1
16. Treasure Island, Robert, Louis Stevenson, 24
17. Kin, Rudyard Kipling, 21, $1
18. The call of the Wild, Jack London, 14, 1.5
19. Lost Horizon, James Hilton, 13, 2.5
20. The warriors, Sol Yurick, 17, 1
21. Physica (Physics), Aristotle, 19, 2
22. What is Life?, Erwin Schrodinger, 16, 2.5
23. To kill a Mockingbird, Harperlee, 20, 1.5
24. Beloved, Toni Morrison, 9, 2.5
25. The stranger, Albert Campus, 8, 1
26. Every Thing Happens For Reason, Shishir Shobhan, 20, 2
Please choose one of the following options."

-Type 'a' to add a book in Book List
-Type 'b' to borrow books
-Type 'r' to return borrowed books
-Type 'l' to list the books available in library.
-Type 'q' to quit the program
what would you like to do :

```

Figure 23 Displaying added new book in the b book list.

7.3. Test 3:

Table 3 Test to borrow book from library.

Objective	To borrow book from the library.
Action	Run module and type b. Enter first and last name of the borrower and chose the number of you want to borrow.
Expected Result	Book should be borrowed showing message successfully borrowed.
Result Obtained	Book was borrowed successfully.
Conclusion	Tested successfully.


```
-Type 'a' to add a book in Book List
-Type 'b' to borrow books
-Type 'r' to return borrowed books
-Type 'l' to list the books available in library.
-Type 'q' to quit the program
what would you like to do : b
Enter the first name of the borrower: Daniel
Enter the last name of the borrower: Magar
0 number for 1. In Search of Lost Time
1 number for 2. Ulysses
2 number for 3. Don Quixote
3 number for 4. One Hundred Years of Solitude
4 number for 5. The Great Gatsby
5 number for 6. The Art of Computer Programming
6 number for 7. Hackers: Herroes of Computer Revolution
7 number for 8. The Road Ahead
8 number for 9. Macintosh
9 number for 10. Alan Turing: The Enigma
10 number for 11. A Hero Born
11 number for 12. DragonFlight
12 number for 13. Christian Apologetics
13 number for 14. The Problem of Pain
14 number for 15. Orthodoxy
15 number for 16. Treasure Island
16 number for 17. Kin
17 number for 18. The call of the Wild
18 number for 19. Lost Horizon
19 number for 20. The warriors
20 number for 21. Physica (Physics)
21 number for 22. What is Life?
22 number for 23. To kill a Mockingbird
23 number for 24. Beloved
24 number for 25. The stranger
25 number for 26. Every Thing Happens For Reason
enter book number: 10
Book you want to borrow is available for you.
Please return the book within 2021-09-16
```

Figure 24 Borrowing book from library.

7.4. Test 4:

Table 4 Test to borrow again by the same user.

Objective	To borrow book again by the same user.
Action	Type y on the option. and chose book number you want to borrow.
Expected Result	Another book should be borrowed.
Result Obtained	Again another book was borrowed.
Conclusion	Tested successfully.

```

Do you want to borrow another book? Remember you cannot borrow the same book again until you returned it. Press y for yes and n for no.y
Please select an option below:
Enter 0 to borrow book 1. In Search of Lost Time
Enter 1 to borrow book 2. Ulysses
Enter 2 to borrow book 3. Don Quixote
Enter 3 to borrow book 4. One Hundred Years of Solitude
Enter 4 to borrow book 5. The Great Gatsby
Enter 5 to borrow book 6. The Art of Computer Programming
Enter 6 to borrow book 7. Hackers: Herroes of Computer Revolution
Enter 7 to borrow book 8. The Road Ahead
Enter 8 to borrow book 9. Macintosh
Enter 9 to borrow book 10. Alan Turing: The Enigma
Enter 10 to borrow book 11. A Hero Born
Enter 11 to borrow book 12. DragonFlight
Enter 12 to borrow book 13. Christian Apologetics
Enter 13 to borrow book 14. The Problem of Pain
Enter 14 to borrow book 15. Orthodoxy
Enter 15 to borrow book 16. Treasure Island
Enter 16 to borrow book 17. Kin
Enter 17 to borrow book 18. The call of the Wild
Enter 18 to borrow book 19. Lost Horizon
Enter 19 to borrow book 20. The warriors
Enter 20 to borrow book 21. Physica (Physics)
Enter 21 to borrow book 22. What is Life?
Enter 22 to borrow book 23. To kill a Mockingbird
Enter 23 to borrow book 24. Beloved
Enter 24 to borrow book 25. The stranger
Enter 25 to borrow book 26. Every Thing Happens For Reason
Enter the book number you want to borrow: 20
Book you want to borrow is available for you.
Do you want to borrow another book? Remember you cannot borrow the same book again until you returned it. Press y for yes and n for no.

```

Figure 25 Image to show borrow again by same user

7.5. Test 5.

Table 5 Test to check quantity decrease

Objective	To check whether the quantity decreases by -1 or not after borrowing book.
Action	Take screen shoot of the book text file and check the quantity of the particular book. After borrowing again check whether the quantity of the book is decreased by -1 or not.
Expected Result	Quantity of the book must be decreased by -1.
Result Obtained	Quantity of the book was decreased by -1.
Conclusion	Tested successfully.

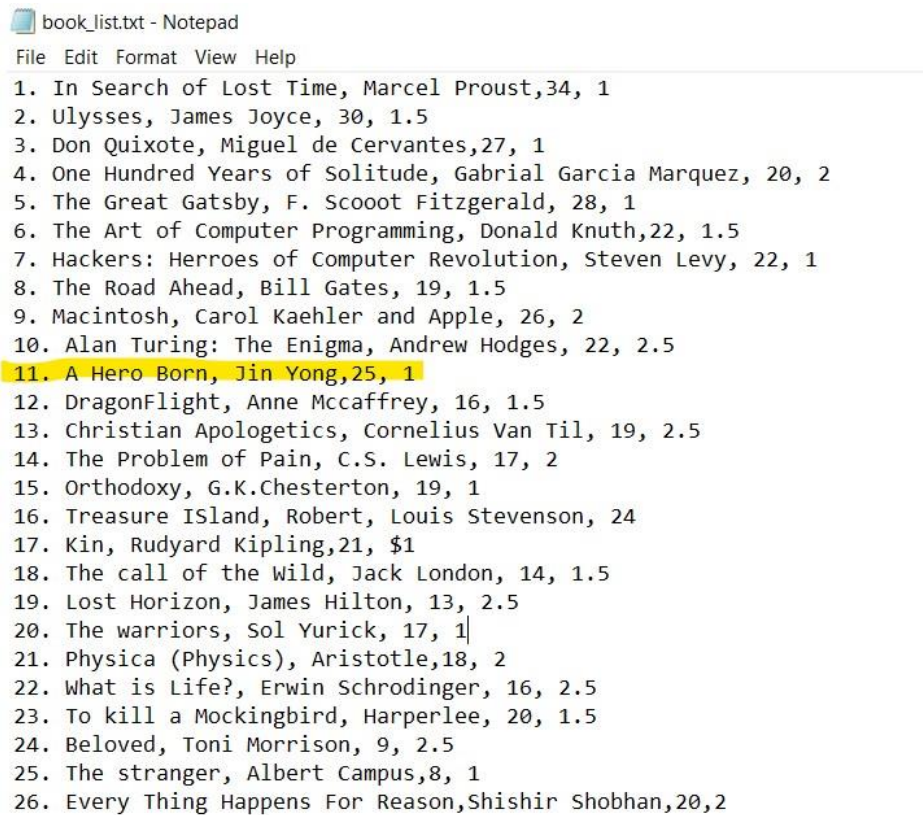


Figure 26 Quantity of book before borrowing.

```

- Type 'a' to add a book in Book List
- Type 'b' to borrow books
- Type 'r' to return borrowed books
- Type 'l' to list the books available in library.
- Type 'q' to quit the program
what would you like to do : b
Enter the first name of the borrower: Daniel
Enter the last name of the borrower: Magar
0 number for 1. In Search of Lost Time
1 number for 2. Ulysses
2 number for 3. Don Quixote
3 number for 4. One Hundred Years of Solitude
4 number for 5. The Great Gatsby
5 number for 6. The Art of Computer Programming
6 number for 7. Hackers: Herroes of Computer Revolution
7 number for 8. The Road Ahead
8 number for 9. Macintosh
9 number for 10. Alan Turing: The Enigma
10 number for 11. A Hero Born
11 number for 12. DragonFlight
12 number for 13. Christian Apologetics
13 number for 14. The Problem of Pain
14 number for 15. Orthodoxy
15 number for 16. Treasure ISland
16 number for 17. Kin
17 number for 18. The call of the Wild
18 number for 19. Lost Horizon
19 number for 20. The warriors
20 number for 21. Physica (Physics)
21 number for 22. What is Life?
22 number for 23. To kill a Mockingbird
23 number for 24. Beloved
24 number for 25. The stranger
25 number for 26. Every Thing Happens For Reason
enter book number: 10
Book you want to borrow is available for you.
Please return the book within 2021-09-16

```

Figure 27 Borrowing book to check quantity decrease.

book_list.txt - Notepad

File Edit Format View Help

```

1. In Search of Lost Time, Marcel Proust,34, 1
2. Ulysses, James Joyce, 30, 1.5
3. Don Quixote, Miguel de Cervantes,27, 1
4. One Hundred Years of Solitude, Gabrial Garcia Marquez, 20, 2
5. The Great Gatsby, F. Scooot Fitzgerald, 28, 1
6. The Art of Computer Programming, Donald Knuth,22, 1.5
7. Hackers: Herroes of Computer Revolution, Steven Levy, 22, 1
8. The Road Ahead, Bill Gates, 19, 1.5
9. Macintosh, Carol Kaehler and Apple, 26, 2
10. Alan Turing: The Enigma, Andrew Hodges, 22, 2.5
11. A Hero Born, Jin Yong,24, 1
12. DragonFlight, Anne Mccaffrey, 16, 1.5
13. Christian Apologetics, Cornelius Van Til, 19, 2.5
14. The Problem of Pain, C.S. Lewis, 17, 2
15. Orthodoxy, G.K.Chesterton, 19, 1
16. Treasure ISland, Robert, Louis Stevenson, 24
17. Kin, Rudyard Kipling,21, $1
18. The call of the Wild, Jack London, 14, 1.5
19. Lost Horizon, James Hilton, 13, 2.5
20. The warriors, Sol Yurick, 17, 1
21. Physica (Physics), Aristotle,17, 2
22. What is Life?, Erwin Schrodinger, 16, 2.5
23. To kill a Mockingbird, Harperlee, 20, 1.5
24. Beloved, Toni Morrison, 9, 2.5
25. The stranger, Albert Campus,8, 1
26. Every Thing Happens For Reason,Shishir Shobhan,20,2

```

Figure 28 Quantity decrease after borrowing book.

7.6. Test 6:

Table 6 To Check billing system while returning book.

Objective	To check the billing system while returning the borrowed book.
Action	Run module and type r to return book. Enter the name of the borrower and bill should be displayed to the user showing all details.
Expected Result	Bill should be displayed showing all details of borrower.
Result Obtained	Bill was displayed.
Conclusion	Tested successfully.

```

Welcome To Library Management System
Please choose one of the following options."

- Type 'a' to add a book in Book List
- Type 'b' to borrow books
- Type 'r' to return borrowed books
- Type 'l' to list the books available in library.
- Type 'q' to quit the program
what would you like to do : r
Enter the name of borrower: Daniel
Library Management System
Borrowed By: Daniel Magar
Borrowed Date: 2021-09-06 Time : 08:20:09.078833 Returning Date : 2021-09-16

S.N.      Bookname      Authername
1.        11. A Hero Born      Jin Yong

$1.0

Check whether the book return date is expired or not.
Press Y for Yes and N for No
|

```

Figure 29 Image showing details in bill while returning book.

7.7. Test 7:

Table 7 Test to check quantity increase after returning book.

Objective	To check whether the quantity of the book will increase by +1 or not after returning book.
Action	Run module and return book.
Expected Result	After returning successfully, quantity of the book should be increased by +1
Result Obtained	Quantity of the book was increased by+1
Conclusion	Tested successfully.

```

Enter the name of borrower: Daniel
Library Management System
Borrowed By: Daniel Magar
Borrowed Date: 2021-09-06      Time : 08:20:09.078833      Returning Date : 2021-09-16

S.N.      Bookname      Authername
1.        11. A Hero Born      Jin Yong

$1.0
Check whether the book return date is expired or not.
Press Y for Yes and N for No
N
Final Total: $1.0
Please choose one of the following options."

-Type 'a' to add a book in Book List
-Type 'b' to borrow books
-Type 'r' to return borrowed books
-Type 'l' to list the books available in library.
-Type 'q' to quit the program
what would you like to do : |

```

Figure 30 Image showing successfully return book.



Figure 31 Quantity Increases after returning book.

7.8. Test 8:

Table 8 To check borrow and return details in different text file.

Objective	To check whether the borrow and return details are saved in new text file with borrower name or not.
Action	Open program containing folder and see newly created text file.
Expected Result	Borrowed and Returned details should be stored in different text file.
Result Obtained	Borrowed and Returned details was stored in different text file.
Conclusion	Tested successfully.

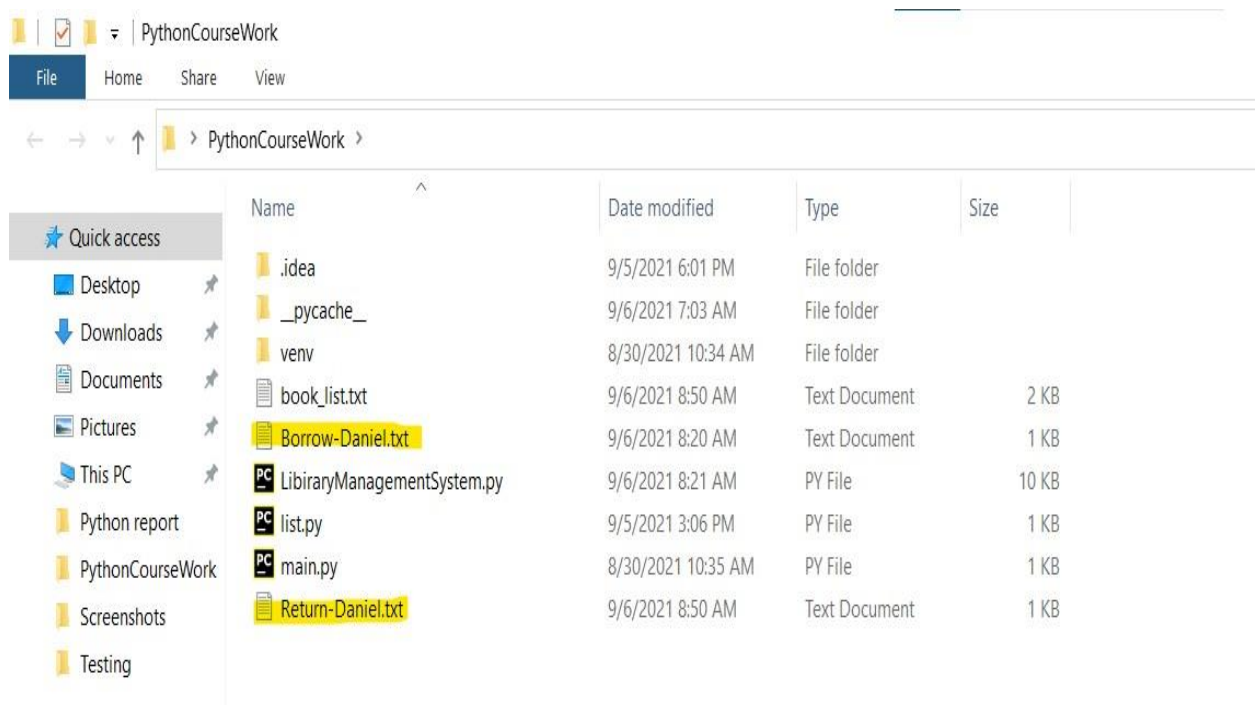


Figure 32 New text file auto created in program containing folder

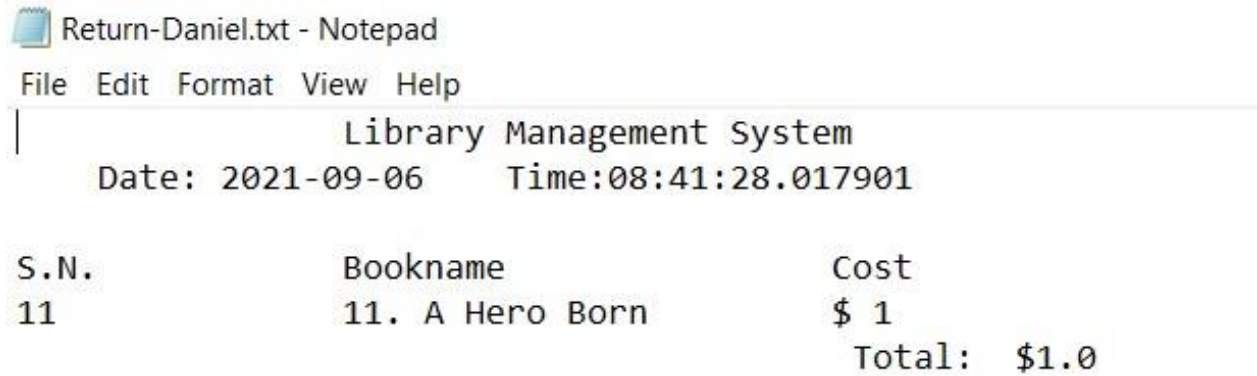


Figure 33 Image showing return details in text file

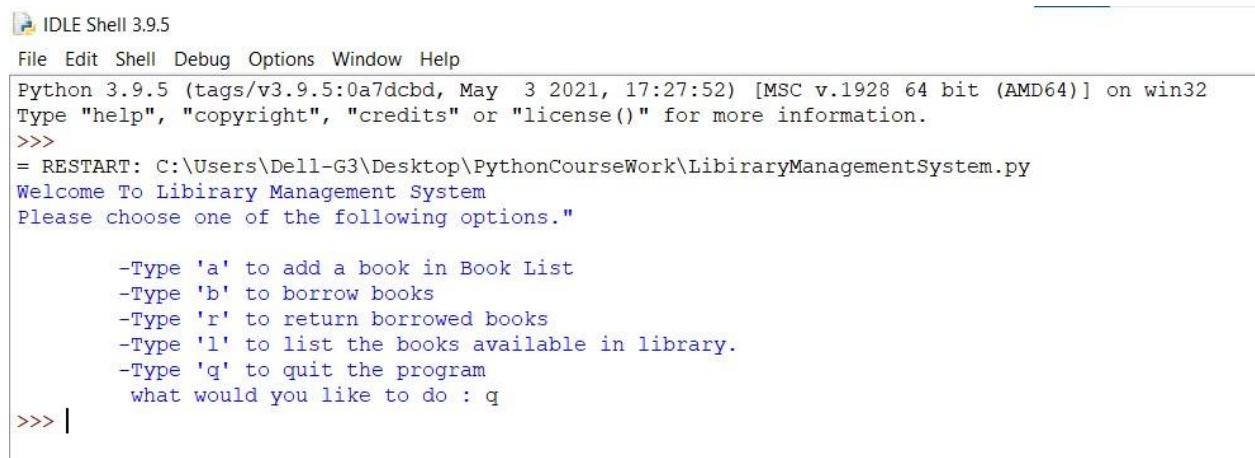


Figure 34 Image showing borrow details in text file.

7.9. Test 9:

Table 9 To check program terminates or not.

Objective	To check whether program terminates or not.
Action	Run module and type q.
Expected Result	The program must terminate when user input q.
Result Obtained	The program was terminated when user input q.
Conclusion	Tested successfully.



```

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Python 3.9.5 (tags/v3.9.5:0a7dcdbd, 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-G3\Desktop\PythonCourseWork\LibiraryManagementSystem.py
Welcome To Libirary Management System
Please choose one of the following options."

    -Type 'a' to add a book in Book List
    -Type 'b' to borrow books
    -Type 'r' to return borrowed books
    -Type 'l' to list the books available in library.
    -Type 'q' to quit the program
    what would you like to do : q
>>> |
  
```

Figure 35 Image showing program terminated.

8. Weekly Progress:

Doing this project needs lot of hard work and dedication. I have to do lots of research as I am very new for this kind of project. Without doing research it was almost impossible for me to complete this project. I started focusing on this project from August fourteen and it took almost four weeks to complete this project. I faced lots of difficulties and error during this project.

In first week I was very excited and nervous as well. So I started this project journey with lots of research, project discussion with teacher and gathering useful information and resources. After doing brief research I got a confidence to do this project and I prepared myself mentally and physically to start.

In second week I started writing algorithm for the program sketching basic logic which are needed. After completing algorithm part, I wrote pseudocode for the program which was very helpful while doing coding portion also I roughly sketch the flowchart using draw.io. Finally, I was ready for the coding portion after second week.

In third week I started coding portion. I used PyCharm and IDLE for developing this program. I selected suitable data structures for writing program and create different functions according to their uses. This week was full of difficulties and errors for me. Despite of all the difficulties I was able to complete development portion without any errors.

In Fourth week I started testing my program using black box testing method, wrote report, debug errors and slightly modified my coding portion where needed. After completing everything I revised and re-check my entire work. Finally, I was able to complete my entire work within a given time.

9. Research:

The project Library Management System revolves around research. Without research it would have been very difficult for me to complete this program as I was unfamiliar with multiple topics. By doing research I was able to gather all the useful sources which gave me idea to start and complete this project. I have also corrected many errors which occurred while developing this program with the help of research.

Websites that I have gone through during research are listed below:

- **Websites:**

I. https://www.w3schools.com/python/python_functions.asp

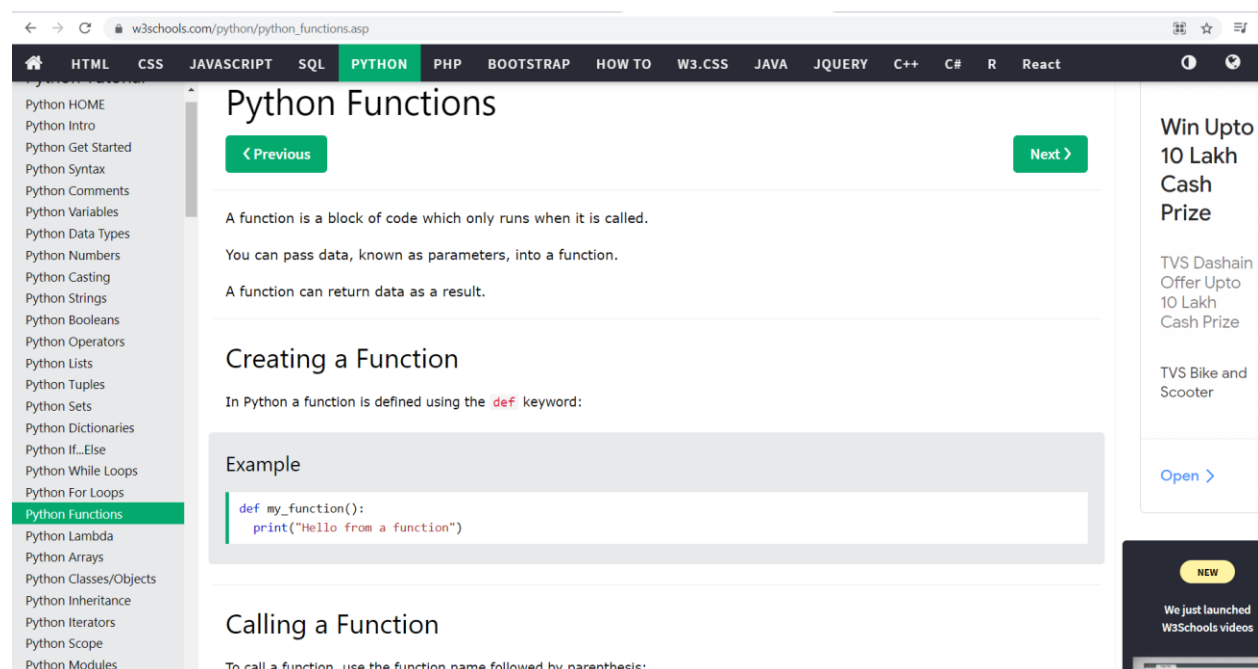


Figure 36 Uses of Functions in Python

From this website I was more clear about the uses of Functions in python. While developing I have separated several lines of code with the help of function which helped me to avoid repetition of same code.

II. https://www.w3schools.com/python/gloss_python_module_import_from.asp

The screenshot shows the W3Schools website's Python tutorial section, specifically the page on importing parts of a module using the `from` keyword. The navigation bar at the top includes links for HTML, CSS, JAVASCRIPT, SQL, PYTHON (highlighted), PHP, BOOTSTRAP, HOW TO, W3.CSS, JAVA, JQUERY, C++, C#, R, and React. A left sidebar lists various Python topics, with 'Python Tutorial' at the top. The main content area explains that you can import parts of a module using the `from` keyword. It provides two examples: 1) A module named `mymodule` containing a `greeting` function and a `person1` dictionary. The code for `mymodule` is shown:

```
def greeting(name):  
    print("Hello, " + name)  
  
person1 = {  
    "name": "John",  
    "age": 36,  
    "country": "Norway"  
}
```

 2) An example showing how to import only the `person1` dictionary from `mymodule`:

```
from mymodule import person1  
  
print (person1["age"])
```

 Below the second example is a green button labeled 'Try it Yourself »'. On the right side of the page, there are two vertical banners: one for 'Compare Gbps SSL TPS, Features, Benefits, Licensing and Support' by Kemp Technologies with a 'Learn More' button, and another for 'We just launched W3Schools videos' with an 'Explore now' button. At the bottom right, there is a 'COLOR PICKER' link.

Figure 37 Image showing how to import multiple files in python

At first I was very unfamiliar with concept of modularity. After visiting this website, I came to know that multiple .py files can be imported in one file. I have also implemented this concept in my work. I have created `list.py` which I have imported in `LibraryManagementSystem.py`.

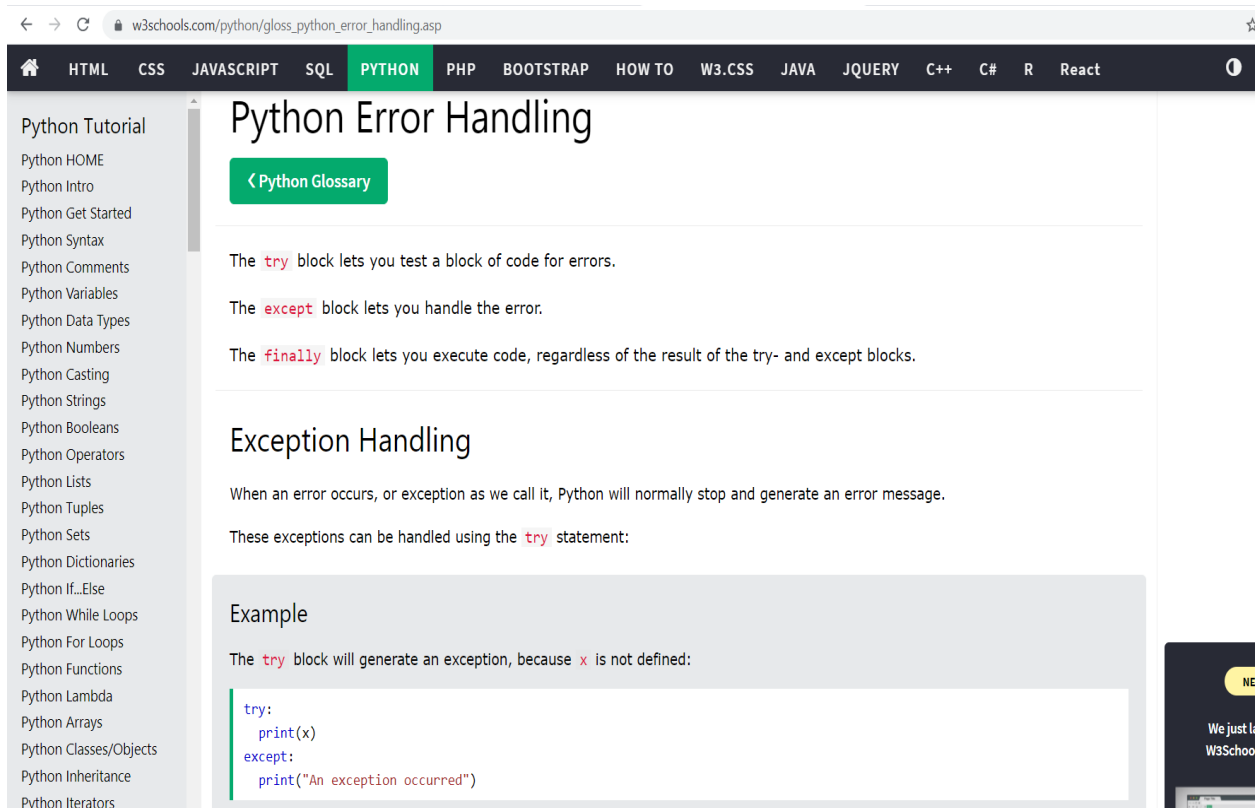
III. https://www.w3schools.com/python/gloss_python_error_handling.asp

Figure 38 Error handling in python.

In my project I have used multiple try and except block to handle the errors that may occurred during execution of program. From this website I gain use of try, except and finally block and also implement them in my work.

10.Conclusion:

The project I have done is Library Management System. It replaces the traditional manual systems that were in practiced to keep record about books available in library which was very tough to manage and also was time consuming while searching particular information. In manual system the record may get destroyed and also has concurrency problems but with the help of Library Management System library activities can be managed systematically and doesn't consume time while searching particular book information.

Though I felt many difficulties during this coursework project, as I wasn't familiar with this kind of project. I need to take help from my teachers and from multiple websites which gave me idea to start and complete my project. In this coursework report I have created different black box tables and attached multiple images to briefly explain my work. I have also drawn flowchart, written algorithm, and wrote pseudocode of the project along with proper explanation, and attached multiple images to briefly explain my work.

These coursework has given me lots of knowledge about uses and usefulness of python programming in future. In short period of time, I am able to develop a program that can be used in library. This program is very useful for librarian and also for students. The main features of this program is it keeps the record of borrowed books and returned books by the borrower name in different text files and also calculate total price of the books and if the returning date is expired shows bill along with fine. This coursework has helped me to learn about uses of functions and concept of modularity in python. It has also helped to build up my research skills.

Overall this coursework provides me great learning experiences and confidence in programming also I would like to thank our module leader Sir Suman Adhakari for guiding this module and giving ideas to complete this project.

11. References

imperva, 2021. *imperva*. [Online]

Available at: <https://www.imperva.com/learn/application-security/black-box-testing/>

[Accessed 06 09 2021].

Lucidchart, 2021. *Lucidchart*. [Online]

Available at: <https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial>

[Accessed 06 09 2021].

Sejal Jaiswal , 2017. *datacamp*. [Online]

Available at: [https://www.datacamp.com/community/tutorials/data-structures-](https://www.datacamp.com/community/tutorials/data-structures-python?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=&utm_creative=332602034361&utm_targetid=ds)

[python?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=&utm_creative=332602034361&utm_targetid=ds](https://www.datacamp.com/community/tutorials/data-structures-python?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=&utm_creative=332602034361&utm_targetid=ds)

[Accessed 06 09 2021].

Tech Target Contributor, 2019. *techtarget*. [Online]

Available at: <https://whatis.techtarget.com/definition/algorithm>

[Accessed 06 09 2021].

TheEconomicTimes, 2021. *TheEconomicTimes*. [Online]

Available at: <https://economictimes.indiatimes.com/definition/pseudocode>

[Accessed 06 09 2021].