

LIBRARY MANAGEMENT SYSTEM

EEI3372 - Programming in Python – Mini Project

Name - A. A. Weerasuriya

S Number – s22010337

Registration Number – 222510555

Group No – WD_G5

INTRODUCTION

Welcome to the Open University of Sri Lanka Library Management System! This system has been created using only Python programming language and a basic interface in command line, without the use of a database or graphical user interface.

Despite its simple design, this library management system offers a range of functions that allow students to borrow resources according to their requirements, as well as add new resources to the system or remove existing ones. Users can also view currently available or unavailable resources in each resource type or filter all resources by subject.

In addition, this system also offers the ability to lend resources to students and update resource status when they are received back. With these functions, the Open University of Sri Lanka Library Management System provides a practical and user-friendly solution for managing library resources in an academic setting.

ASSUMPTIONS

The system is designed solely for use by students at the Open University of Sri Lanka, and not for the general public.

The system does not include a database, indicating that all information about resources, borrowers, and transactions is stored within the program's code itself.

The lack of a graphical user interface suggests that the system is intended to be used primarily by individuals with some familiarity with command-line interfaces and basic programming concepts.

The system is able to handle basic borrowing and lending functions, but may not have more advanced features such as automated reminders for overdue items or the ability to track fines and fees.

The system's resource types are well-defined and that users are able to easily identify which type of resource they need.

The system think that users are responsible for returning borrowed resources on time and in good condition.

The system assumes that the Open University of Sri Lanka has a limited collection of resources that can be managed effectively using a basic command-line interface. If the collection were much larger or more complex, a more advanced system with a database and graphical user interface may be necessary.

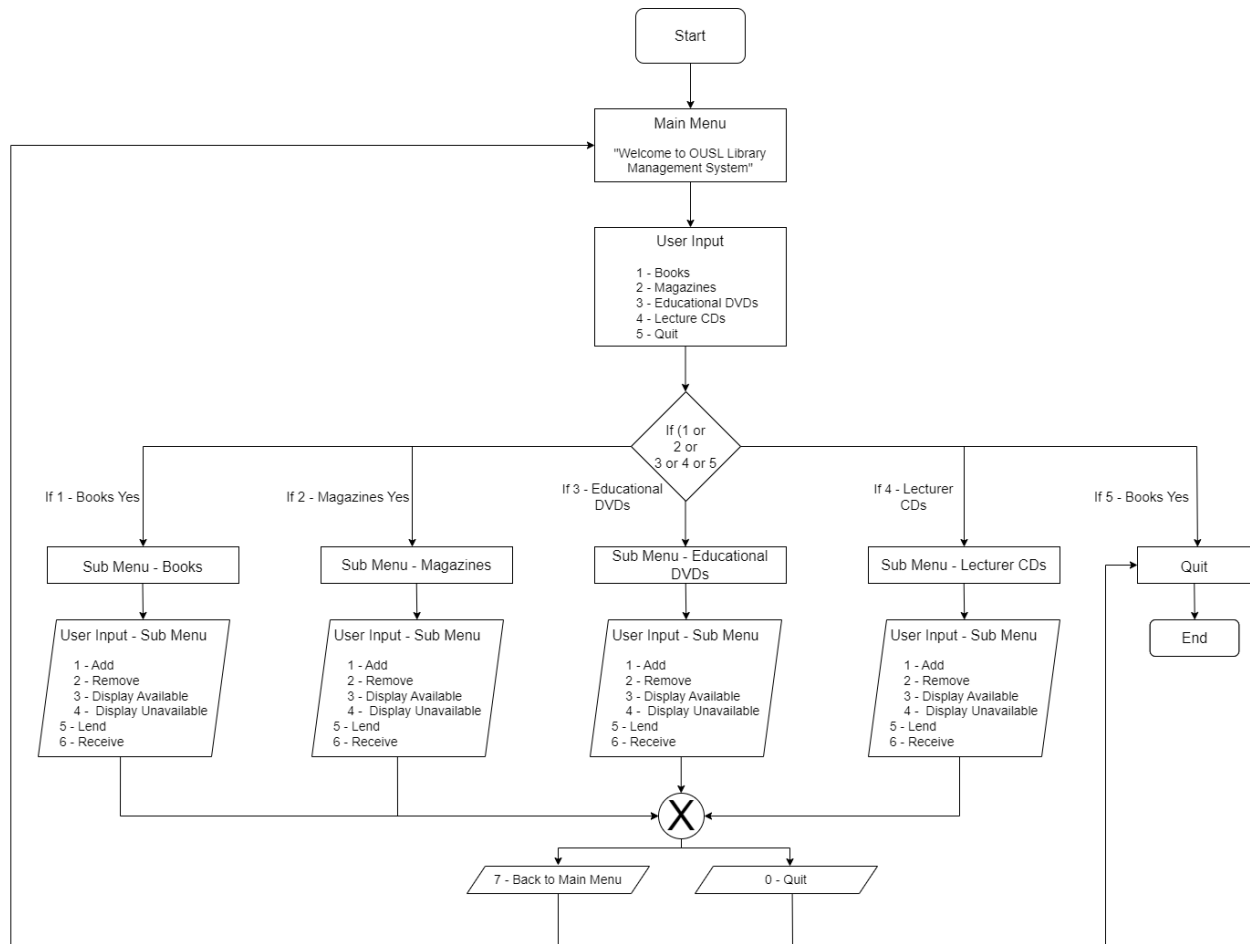
PROBLEMS I FACED WHILE DEVELOPING

- I lacked knowledge about functions, lists, and Python classes.
- I couldn't understand how to create the user interface using Python.
- I had to decide whether to code everything in one page or in multiple pages.
- I had a lack of time to finish the project on time.

HOW I OVERCAME THEM

- I went through study materials again to get a better understanding of Python functions, lists, and classes. I also referred to YouTube and W3School.
- I learned how to make a simple user interface using Python by searching the internet for resources.
- To make the code easier to read and understand, I decided to split it into multiple pages and link them together using the "from" and "import" statements.
- I tried to manage my time better by avoiding unnecessary tasks.

FLOWCHART OF THE PROGRAM



SOURCE CODE AND OUTPUT SCREENSHOTS

Class

```
#-----Book-----
class Book:
    def __init__(self, isbn_no, title, format, subject, rental_price,
copies):
        self.isbn_no = isbn_no
        self.title = title
        self.format = format
        self.subject = subject
        self.rental_price = rental_price
        self.copies = copies

#-----Magazine-----
class Magazine:
    def __init__(self, m_no, title, color, subject, rental_price, copies):
        self.m_no = m_no
        self.title = title
        self.color = color
        self.subject = subject
        self.rental_price = rental_price
        self.copies = copies

#-----Educational_DVD-----
class Educational_DVD:
    def __init__(self, dvd_no, title, subject, rental_price, copies):
        self.dvd_no = dvd_no
        self.title = title
        self.subject = subject
        self.rental_price = rental_price
        self.copies = copies

#-----Lecture_CD-----
class Lecture_CD:
    def __init__(self, cd_no, title, subject, rental_price, copies):
        self.cd_no = cd_no
        self.title = title
        self.subject = subject
        self.rental_price = rental_price
        self.copies = copies
```

Main.py

```
from bookfunction import Book_Function
from magazinefunction import Magazine_Function
from educationaldvdfunction import Educational_DVD_Function
from lecturecdfunction import Lecture_CD_Function

book_func = Book_Function()
magazine_func = Magazine_Function()
eternal_dvd_func = Educational_DVD_Function()
lecture_cd_func = Lecture_CD_Function()

# book menu
def submenu(book_func, selected_resource):
    selected_operation = 1
    selected_resource = "Book"
    while selected_operation > 0:
        print("Please select the operation menu")
        print("-----")
        print(f"1 - Add a {selected_resource}")
        print(f"2 - Remove a {selected_resource}")
        print(f"3 - Display Available {selected_resource}")
        print(f"4 - Display Unavailable {selected_resource}")
        print(f"5 - Lend {selected_resource}")
        print(f"6 - Receive {selected_resource}")
        print("7 - Back to Main Menu")
        print("0 - Quit")
        selected_operation = int(input("Please Type The Number Of Your
Choice: ").strip())

        if selected_operation == 1:
            book_func.add()
        elif selected_operation == 2:
            book_func.remove()
        elif selected_operation == 3:
            book_func.display_available()
        elif selected_operation == 4:
            book_func.display_unavailable()
        elif selected_operation == 5:
            book_func.lend()
        elif selected_operation == 6:
            book_func.receive()
        elif selected_operation == 7:
            mainmenu() # back to main menu
        else:
            print("Invalid Selection")

        if 1 <= selected_operation <= 7:
            input("Press Any Key To Continue...")

# magazine menu
def submenu2(magazine_func, selected_resource):
    selected_operation = 2
    selected_resource = "Magazine"
```

```

while selected_operation > 0:
    print("Please select the operation menu")
    print("-----")
    print(f"1 - Add a {selected_resource}")
    print(f"2 - Remove a {selected_resource}")
    print(f"3 - Display Available {selected_resource}")
    print(f"4 - Display Unavailable {selected_resource}")
    print(f"5 - Lend {selected_resource}")
    print(f"6 - Receive {selected_resource}")
    print("7 - Back to Main Menu")
    print("0 - Quit")
    selected_operation = int(input("Please Type The Number Of Your
Choice: ").strip())

    if selected_operation == 1:
        magazine_func.add()
    elif selected_operation == 2:
        magazine_func.remove()
    elif selected_operation == 3:
        magazine_func.display_available()
    elif selected_operation == 4:
        magazine_func.display_unavailable()
    elif selected_operation == 5:
        magazine_func.lend()
    elif selected_operation == 6:
        magazine_func.receive()
    elif selected_operation == 7:
        mainmenu() # back to main menu
    else:
        print("Invalid Selection")

    if 1 <= selected_operation <= 7:
        input("Press Any Key To Continue...")

# Educational_DVD menu
def submenu3(external_dvd_func, selected_resource):
    selected_operation = 3
    selected_resource = "Educational_DVD"
    while selected_operation > 0:
        print("Please select the operation menu")
        print("-----")
        print(f"1 - Add a {selected_resource}")
        print(f"2 - Remove a {selected_resource}")
        print(f"3 - Display Available {selected_resource}")
        print(f"4 - Display Unavailable {selected_resource}")
        print(f"5 - Lend {selected_resource}")
        print(f"6 - Receive {selected_resource}")
        print("7 - Back to Main Menu")
        print("0 - Quit")
        selected_operation = int(input("Please Type The Number Of Your
Choice: ").strip())

        if selected_operation == 1:
            external_dvd_func.add()
        elif selected_operation == 2:
            external_dvd_func.remove()

```



```

elif selected_operation == 3:
    eternal_dvd_func.display_available()
elif selected_operation == 4:
    eternal_dvd_func.display_unavailable()
elif selected_operation == 5:
    eternal_dvd_func.lend()
elif selected_operation == 6:
    eternal_dvd_func.receive()
elif selected_operation == 7:
    mainmenu() # back to main menu
else:
    print("Invalid Selection")

if 1 <= selected_operation <= 7:
    input("Press Any Key To Continue...")

# Lecture_CD menu
def submenu4(lecture_cd_func, selected_resource):
    selected_operation = 4
    selected_resource = "Lecture_CD"
    while selected_operation > 0:
        print("Please select the operation menu")
        print("-----")
        print(f"1 - Add a {selected_resource}")
        print(f"2 - Remove a {selected_resource}")
        print(f"3 - Display Available {selected_resource}")
        print(f"4 - Display Unavailable {selected_resource}")
        print(f"5 - Lend {selected_resource}")
        print(f"6 - Receive {selected_resource}")
        print("7 - Back to Main Menu")
        print("0 - Quit")
        selected_operation = int(input("Please Type The Number Of Your
Choice: ").strip())

    if selected_operation == 1:
        lecture_cd_func.add()
    elif selected_operation == 2:
        lecture_cd_func.remove()
    elif selected_operation == 3:
        lecture_cd_func.display_available()
    elif selected_operation == 4:
        lecture_cd_func.display_unavailable()
    elif selected_operation == 5:
        lecture_cd_func.lend()
    elif selected_operation == 6:
        lecture_cd_func.receive()
    elif selected_operation == 7:
        mainmenu() # back to main menu
    else:
        print("Invalid Selection")

    if 1 <= selected_operation <= 7:
        input("Press Any Key To Continue...")

def mainmenu():

```

```
print("Main Menu")
print("-----")
print("1 - Books")
print("2 - Magazines")
print("3 - Educational DVDs")
print("4 - Lecture CDs")
print("0 - Quit")

print("WELCOME TO OUSL LIBRARY MANAGEMENT SYSTEM")
print("=====")
selected_resource = 1
while selected_resource > 0:
    mainmenu()
    try:
        selected_resource = int(input("Please Select Your Option: "))
    except ValueError:
        print("Invalid Entry")
        mainmenu()

    if selected_resource == 0:
        print("Thank You For Using Library System")
        quit()
    elif selected_resource == 1:
        function_name = book_func
        submenu(function_name, "Book")
        break
    elif selected_resource == 2:
        function_name = magazine_func
        submenu(function_name, "Magazine")
        break
    elif selected_resource == 3:
        function_name = eternal_dvd_func
        submenu(function_name, "Educational DVD")
        break
    elif selected_resource == 4:
        function_name = lecture_cd_func
        submenu(function_name, "Lecture CD")
        break
```

Bookfunction.py

```
from allclasses import Book

def print_info(book):
    print(f"ISBN NO: {book.isbn_no}, Title: {book.title}, Format: {book.format}, Subject: {book.subject}, Rental price per day: {book.rental_price}, Number of Copies Available: {book.copies}")

class Book_Function:
    operation_cancelled = "Operation Cancelled"

    def __init__(self):
        self.list_of_books = []
        self.__initial_data()

    def __initial_data(self):
        book1 = Book(isbn_no="ISBN1231", title="Game Of Thrones",
format="Hardcover", subject="Science", rental_price=12.50, copies=5)
        book2 = Book(isbn_no="ISBN1232", title="Vikings", format="Paperback",
subject="History", rental_price=50.00, copies=0)
        book3 = Book(isbn_no="ISBN1234", title="The Great Gatsby",
format="Hardcover", subject="Literature", rental_price=150.00, copies=2)
        self.list_of_books.append(book1)
        self.list_of_books.append(book2)
        self.list_of_books.append(book3)

    def add(self):
        __isbn = input("Enter ISBN Number:").strip().upper()
        __title = input("Title:").strip().upper()
        __format = input("Format:")
        __subject = input("Subject:")
        __rental_price = float(input("Rental price per day:"))
        __copies = int(input("Number of Copies Available:"))

        a_book = Book(isbn_no=__isbn, title=__title, format=__format,
subject=__subject, rental_price=__rental_price, copies=__copies)
        self.list_of_books.append(a_book)
        print(f"Book added {a_book.isbn_no}-{a_book.title}")

    def remove(self):
        __isbn = input("Enter ISBN number:")
        matched_data = list(x for x in self.list_of_books if x.isbn_no ==
__isbn)
        for x in matched_data:
            self.list_of_books.remove(x)
            print("Item Removed.")

    def lend(self):
        __isbn = input("Enter ISBN number:")
        __copies = int(input("Enter lend copies:"))
        matched_data = list(x for x in self.list_of_books if x.isbn_no ==
__isbn)
        for x in matched_data:
            x.copies -= __copies
            print("Book Lent")
```

```

def receive(self):
    __isbn = input("Enter ISBN number:")
    __copies = int(input("Enter receive copies:"))
    matched_data = list(x for x in self.list_of_books if x.isbn_no ==
__isbn)
    for x in matched_data:
        x.copies += __copies
        print(f"Book Received with {__copies} Copies")

def display_all(self):
    for x in self.list_of_books:
        print_info(book=x)

def display_available(self):
    matched_data = list(x for x in self.list_of_books if x.copies > 0)
    for x in matched_data:
        print_info(book=x)

def display_unavailable(self):
    matched_data = list(x for x in self.list_of_books if x.copies == 0)
    for x in matched_data:
        print_info(book=x)

```

magazinefunction.py

```

from allclasses import Magazine

def print_info(magazine):
    print(f"Magazine Number: {magazine.m_no}, Title: {magazine.title},
Format: {magazine.color}, Subject: {magazine.subject}, Rental Price:
{magazine.rental_price}, Available Copies:{magazine.copies}")

class Magazine_Function:

    def __init__(self):
        self.list_of_magazines = []
        self.__initial_data()

    def __initial_data(self):
        a_magazine1 = Magazine(m_no="01", title="National Geographic",
color="Blue", subject="Science", rental_price=120.50, copies=1)
        a_magazine2 = Magazine(m_no="02", title="The New Yorker",
color="Black & White", subject="Technology", rental_price=220.50, copies=4)
        a_magazine3 = Magazine(m_no="03", title="Time", color="Format3",
subject="Sports", rental_price=30.50, copies=5)
        self.list_of_magazines.append(a_magazine1)
        self.list_of_magazines.append(a_magazine2)
        self.list_of_magazines.append(a_magazine3)

    def add(self):
        __mnum = input("Enter Magazine Number:").strip().upper()
        __title = input("Title:").strip().upper()

```

```

        __color = input("Color:")
        __subject = input("Subject:")
        __rental_price = float(input("Rental price:"))
        __copies = int(input("Copies:"))

        a_magazine = Magazine(m_no=__mnum, title=__title, color=__color,
subject=__subject, rental_price=__rental_price, copies=__copies)
        self.list_of_magazines.append(a_magazine)
        print(f"Magazine Added { a_magazine.m_no}-{a_magazine.title}")

    def remove(self):
        __mnum = input("Enter Magazine Number:")
        matched_data = list(x for x in self.list_of_magazines if x.m_no ==
__mnum)
        for x in matched_data:
            self.list_of_magazines.remove(x)
            print("Item Removed.")

    def lend(self):
        __mnum = input("Enter Magazine Number:")
        __copies = int(input("Enter Lend Copies:"))
        matched_data = list(x for x in self.list_of_magazines if x.m_no ==
__mnum)
        for x in matched_data:
            x.copies -= __copies
            print("Magazine Lent")

    def receive(self):
        __mnum = input("Enter Magazine Number:")
        __copies = int(input("Enter Receive Copies:"))
        matched_data = list(x for x in self.list_of_magazines if x.m_no ==
__mnum)
        for x in matched_data:
            x.copies += __copies
            print(f"Magazine Received With {__copies} Copies")

    def display_all(self):
        for x in self.list_of_magazines:
            print_info(magazine=x)

    def display_available(self):
        matched_data = list(x for x in self.list_of_magazines if x.copies >
0)
        for x in matched_data:
            print_info(magazine=x)

    def display_unavailable(self):
        matched_data = list(x for x in self.list_of_magazines if x.copies ==
0)
        for x in matched_data:
            print_info(magazine=x)

```

educationdvdfunction.py

```
from allclasses import Educational_DVD

def print_info(educational_dvd):
    print(f"Educational_DVD NO: {educational_dvd.dvd_no}, Title: {educational_dvd.title}, Subject: {educational_dvd.subject}, Rental Price: {educational_dvd.rental_price}, Available Copies:{educational_dvd.copies}")

class Educational_DVD_Function:

    def __init__(self):
        self.list_of_educational_dvds = []
        self.__initial_data()

    def __initial_data(self):
        a_educational_dvd1 = Educational_DVD(dvd_no="10", title="Planet Earth", subject="Astronomy", rental_price=120.50, copies=10)
        a_educational_dvd2 = Educational_DVD(dvd_no="11", title="The History of Maths", subject="Math", rental_price=220.50, copies=0)
        a_educational_dvd3 = Educational_DVD(dvd_no="12", title="The Human Body", subject="Technology", rental_price=320.50, copies=5)
        self.list_of_educational_dvds.append(a_educational_dvd1)
        self.list_of_educational_dvds.append(a_educational_dvd2)
        self.list_of_educational_dvds.append(a_educational_dvd3)

    def add(self):
        __dvd_no = input("Enter Educational DVD Number:").strip().upper()
        __title = input("Title:").strip().upper()
        __subject = input("Subject:")
        __rental_price = float(input("Rental Price:"))
        __copies = int(input("Copies:"))

        a_educational_dvd = Educational_DVD(dvd_no=__dvd_no, title=__title, subject=__subject, rental_price=__rental_price, copies=__copies)
        self.list_of_educational_dvds.append(a_educational_dvd)
        print(f"Educational DVD Added {a_educational_dvd.dvd_no}-{a_educational_dvd.title}")

    def remove(self):
        __dvd_no = input("Enter DVD Number:")
        matched_data = list(x for x in self.list_of_educational_dvds if x.dvd_no == __dvd_no)
        for x in matched_data:
            self.list_of_educational_dvds.remove(x)
            print("Item Removed.")

    def lend(self):
        __dvd_no = input("Enter DVD Number:")
        __copies = int(input("Enter Lend Copies:"))
        matched_data = list(x for x in self.list_of_educational_dvds if x.dvd_no == __dvd_no)
        for x in matched_data:
            x.copies -= __copies
            print("Educational DVD Lent")

    def receive(self):
```

```
        __dvd_no = input("Enter DVD Number:")
        __copies = int(input("Enter Receive Copies:"))
        matched_data = list(x for x in self.list_of_educational_dvds if
x.dvd_no == __dvd_no)
        for x in matched_data:
            x.copies += __copies
            print(f"Educational DVD Received With {__copies} Copies")

    def display_all(self):
        for x in self.list_of_educational_dvds:
            print_info(educational_dvd=x)

    def display_available(self):
        matched_data = list(x for x in self.list_of_educational_dvds if
x.copies > 0)
        for x in matched_data:
            print_info(educational_dvd=x)

    def display_unavailable(self):
        matched_data = list(x for x in self.list_of_educational_dvds if
x.copies == 0)
        for x in matched_data:
            print_info(educational_dvd=x)
```

Lecturecdfunction.py

```
from allclasses import Lecture_CD

def print_info(lecture_cd):
    print(f"CD NO: {lecture_cd.cd_no}, Title: {lecture_cd.title}, Subject: {lecture_cd.subject}, Rental Price: {lecture_cd.rental_price}, Available Copies:{lecture_cd.copies}")

class Lecture_CD_Function:

    def __init__(self):
        self.list_of_lecture_cd = []
        self.__initial_data()

    def __initial_data(self):
        a_lecture_cd1 = Lecture_CD(cd_no="21", title="Basics of Western Music,", subject="Music", rental_price=120.50, copies=5)
        a_lecture_cd2 = Lecture_CD(cd_no="22", title="A Mathematician's Lament", subject="Math", rental_price=100.50, copies=0)
        a_lecture_cd3 = Lecture_CD(cd_no="23", title="One Hundred Years of Solitude", subject="Foreign Languages", rental_price=500.50, copies=10)
        self.list_of_lecture_cd.append(a_lecture_cd1)
        self.list_of_lecture_cd.append(a_lecture_cd2)
        self.list_of_lecture_cd.append(a_lecture_cd3)

    def add(self):
        __cd = input("Enter CD Number:").strip().upper()
        __title = input("Title:").strip().upper()
        __subject = input("Subject:")
        __rental_price = float(input("Rental Price:"))
        __copies = int(input("Copies:"))

        a_lecture_cd = Lecture_CD(cd_no=__cd, title=__title, subject=__subject, rental_price=__rental_price, copies=__copies)
        self.list_of_lecture_cd.append(a_lecture_cd)
        print(f"Book Added {a_lecture_cd.cd_no}-{a_lecture_cd.title}")

    def remove(self):
        __cd = input("Enter CD number:")
        matched_data = list(x for x in self.list_of_lecture_cd if x.cd_no == __cd)

        for x in matched_data:
            self.list_of_lecture_cd.remove(x)
            print("Item Removed.")

    def lend(self):
        __cd = input("Enter CD number:")
        __copies = int(input("Enter Lend Copies:"))
        matched_data = list(x for x in self.list_of_lecture_cd if x.cd_no == __cd)

        for x in matched_data:
            x.copies -= __copies
            print("Lecture CD Lent")

    def receive(self):
        __cd = input("Enter ISBN number:")
```



```

        __copies = int(input("Enter Receive Copies:"))
        matched_data = list(x for x in self.list_of_lecture_cd if x.cd_no ==
__cd)
        for x in matched_data:
            x.copies += __copies
            print(f"Lecture_CD Received with {__copies} Copies")

    def display_all(self):
        for x in self.list_of_lecture_cd:
            print_info(lecture_cd=x)

    def display_available(self):
        matched_data = list(x for x in self.list_of_lecture_cd if x.copies >
0)
        for x in matched_data:
            print_info(lecture_cd=x)

    def display_unavailable(self):
        matched_data = list(x for x in self.list_of_lecture_cd if x.copies ==
0)
        for x in matched_data:
            print_info(lecture_cd=x)

```

OUTPUT SCREENSHOTS

Main menu

```
=====
Main Menu
-----
1 - Books
2 - Magazines
3 - Educational DVDs
4 - Lecture CDs
0 - Quit
Please Select Your Option: |
```

Sub menu

```
WELCOME TO OUSL LIBRARY MANAGEMENT SYSTEM
=====
Main Menu
-----
1 - Books
2 - Magazines
3 - Educational DVDs
4 - Lecture CDs
0 - Quit
Please Select Your Option: 1
Please select the operation menu
-----
1 - Add a Book
2 - Remove a Book
3 - Display Available Book
4 - Display Unavailable Book
5 - Lend Book
6 - Receive Book
7 - Back to Main Menu
0 - Quit
Please Type The Number Of Your Choice: |
```

add

```
-----  
1 - Add a Book  
2 - Remove a Book  
3 - Display Available Book  
4 - Display Unavailable Book  
5 - Lend Book  
6 - Receive Book  
7 - Back to Main Menu  
0 - Quit  
Please Type The Number Of Your Choice: 1  
Enter ISBN Number:ISBN1231  
Title:Game Of Thrones  
Format:Hardcover  
Subject:Science  
Rental price per day:50  
Number of Copies Available:10  
Book added ISBN1231-GAME OF THRONES  
Press Any Key To Continue...|
```

remove

```
-----  
1 - Add a Book  
2 - Remove a Book  
3 - Display Available Book  
4 - Display Unavailable Book  
5 - Lend Book  
6 - Receive Book  
7 - Back to Main Menu  
0 - Quit  
Please Type The Number Of Your Choice: 2  
Enter ISBN number:ISBN1231  
Item Removed.  
Item Removed.  
Press Any Key To Continue...
```

Display available

```
-----
1 - Add a Book
2 - Remove a Book
3 - Display Available Book
4 - Display Unavailable Book
5 - Lend Book
6 - Receive Book
7 - Back to Main Menu
0 - Quit
Please Type The Number Of Your Choice: 3
ISBN NO: ISBN1234, Title: The Great Gatsby, Format: Hardcover, Subject: Literature, Rental price per day: 150.0, Number of Copies Available: 2
Press Any Key To Continue...|
```

Display unavailable

```
-----
1 - Add a Book
2 - Remove a Book
3 - Display Available Book
4 - Display Unavailable Book
5 - Lend Book
6 - Receive Book
7 - Back to Main Menu
0 - Quit
Please Type The Number Of Your Choice: 4
ISBN NO: ISBN1232, Title: Vikings, Format: Paperback, Subject: History, Rental price per day: 50.0, Number of Copies Available: 0
Press Any Key To Continue...|
```

lend

```
-----
1 - Add a Book
2 - Remove a Book
3 - Display Available Book
4 - Display Unavailable Book
5 - Lend Book
6 - Receive Book
7 - Back to Main Menu
0 - Quit
Please Type The Number Of Your Choice: 5
Enter ISBN number: ISBN1231
Enter lend copies: 2
Press Any Key To Continue...|
```

recieve

```
1 - Add a Book
2 - Remove a Book
3 - Display Available Book
4 - Display Unavailable Book
5 - Lend Book
6 - Receive Book
7 - Back to Main Menu
0 - Quit
Please Type The Number Of Your Choice: 6
Enter ISBN number:ISBN1231
Enter receive copies:2
Press Any Key To Continue...|
```

Back to main menu

```
-----
1 - Add a Book
2 - Remove a Book
3 - Display Available Book
4 - Display Unavailable Book
5 - Lend Book
6 - Receive Book
7 - Back to Main Menu
0 - Quit
Please Type The Number Of Your Choice: 7
Main Menu
-----
1 - Books
2 - Magazines
3 - Educational DVDs
4 - Lecture CDs
0 - Quit
Press Any Key To Continue...|
```

</End>