

2020/2021 – Semester 2

SECJ2154

OBJECT ORIENTED PROGRAMMING

Lecturer Name: Dr Ruhaidah Samsudin

Section 10

MINI PROJECT

**Library System to Store User Details and Books
Information**

Group Members:

1. Srivani Murugan (A19EC0305)
2. Nurul Aqilah Binti Ahmad (A19EC0203)
3. Nurul Syamira Binti Amat Jifri (A19EC0145)

1. Introduction

The modern generation continues to rely on computers. The **Library System to Store User Details and Books Information** is a project that aims to develop a computerized system to support of all the daily work of the library. The project has many features not generally found in manual library systems that store information about various types of books separately and store user information and their account details.

The librarian after logging into this system can access different types of options such as add new users, issues books, and return book details. The library consists of different types of printed work. The system as differentiate each printed work as journals, magazines and study books. This will easy for the librarian to issues the books details.

This program also will be more comfortable to user where the users can get the book details through this program. The only thing user needs to do is register an account for them to access the system. This system store user details and their account information.

Overall, the project is being developed to help librarians and users to reduce human effort especially for librarians to maintain library more easily.

2. Problem Statement

The problem encountered in the manual system is that the user needs an efficient way to use any feature in the library system without using the card catalog system. There are a number of problems in the manual system with the use of card catalogs. It is very expensive and the workflow is slow and inefficient. It is a waste of time when looking for a long list of books and can result in book loss. This happens when a user borrows a book and does not return it on time, and the manual system is unable to use it. This in turn causes usage limitations for users as library material information may not be fully displayed and difficulty in maintaining the system as it requires a lot of staff and expense.

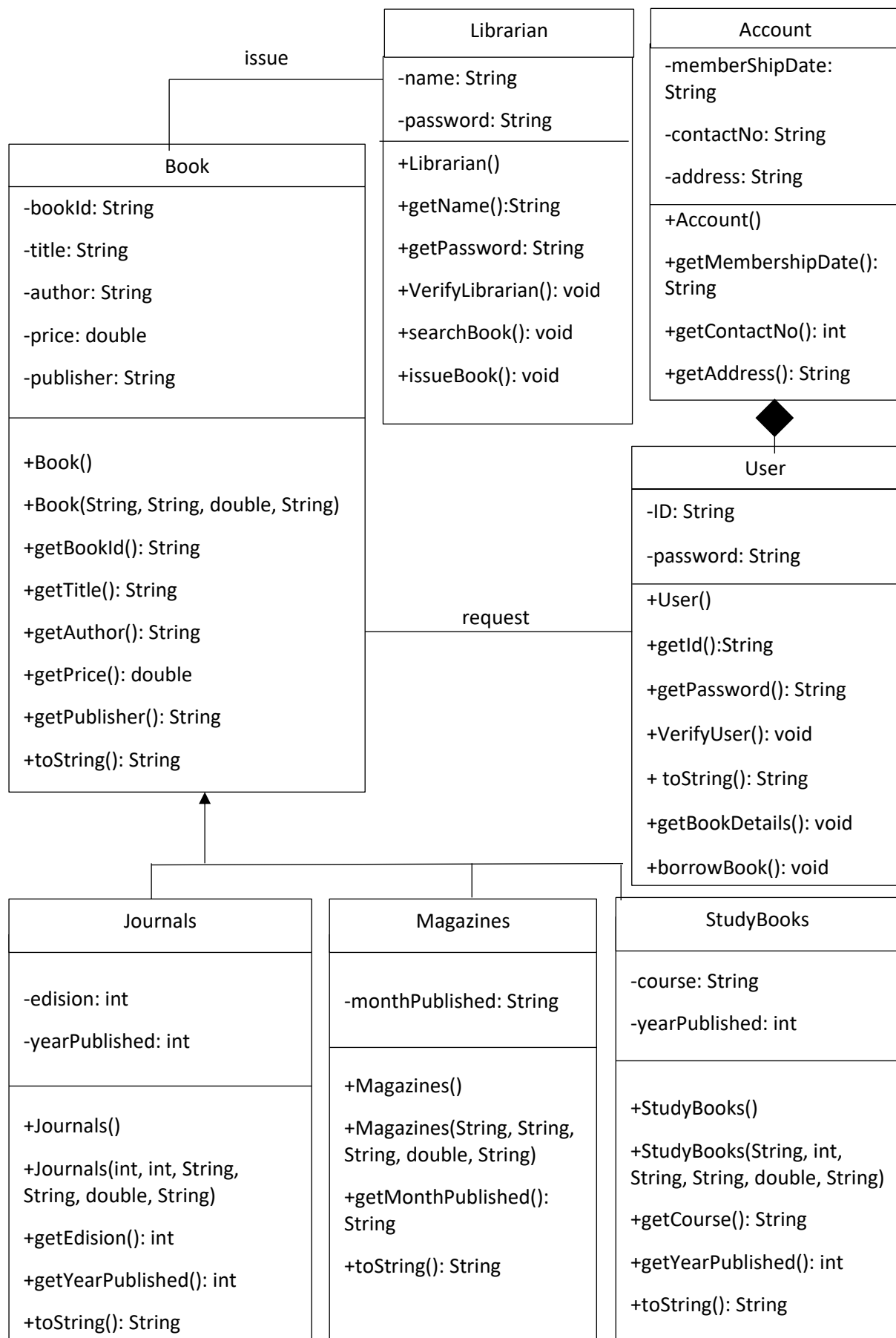
The proposed function to overcome this manual library problem is to develop a computerized library system that can store complete information of all materials in an organized list that will make it easier for librarians to access the system and create a systematic system to borrow and return books to avoid book loss.

3. Objective

The objective of this project is to develop a system that can handle and manage the daily activities of the library efficiently. The main goal of this project is to make the library system have less staff to manage and the availability of easy search and management of user information.

- Develop a computerized system to replace the manual library managing system.
- To enable online searching library's materials
- To allow people borrow and return the library's material without the card system
- To do cataloguing and indexing the material

4. Project Design (UML Diagram)



5. Project Output

(a) Librarian Log In, Add Book

```
LOGIN PAGE
Log In as (1)Librarian or (2)Non-librarian or (3)Quit -> 1
Enter name -> Aqilah
Enter password -> lib002
Librarian logged in.

LIBRARIAN WORK PAGE
(1)Search or (2)Add book or (3)Back to Login Page-> 2

ADD BOOK PAGE
(1)Journals (2)Magazines (3)Studybook (4)Quit adding -> 3
Book ID -> bk003
Title -> Skills for Success
Author -> Nigel A. Caplan
Price -> 25
Publisher -> Oxford
Course -> UHLB
Year Published -> 2020
Book added.

ADD BOOK PAGE
(1)Journals (2)Magazines (3)Studybook (4)Quit adding -> _
```

(b) Librarian Search Book

```
LOGIN PAGE
Log In as (1)Librarian or (2)Non-librarian or (3)Quit -> 1
Enter name -> Syamira
Enter password -> lib003
Librarian logged in.

LIBRARIAN WORK PAGE
(1)Search or (2)Add book or (3)Back to Login Page-> 1

SEARCH PAGE
Search by book ID -> bk002
Book Details
Book ID: bk002
Title: Human's Energy
Author: Joseph Paul
Price: 40.0
Publisher: Nat Geo
Edition: 2
Year Published: 2012

LIBRARIAN WORK PAGE
(1)Search or (2)Add book or (3)Back to Login Page-> _
```

(c) User Log In, View Book Details, View User Details, Borrow Book

```
LOGIN PAGE
Log In as (1)Librarian or (2)Non-librarian or (3)Quit -> 2
Enter ID -> NL002
Enter password -> user002
User logged in.

USER MAIN PAGE
(1)Book details or (2)User details or (3)Borrow Book (4)Back to Login Page -> 1
Book Details
Book ID: bk001
Title: Object-Oriented Programming
Author: Jailani Y.
Price: 20.3
Publisher: Sasbadi
Course: Programming
Year Published: 2013

Book Details
Book ID: bk002
Title: Human's Energy
Author: Joseph Paul
Price: 40.0
Publisher: Nat Geo
Edition: 2
Year Published: 2012

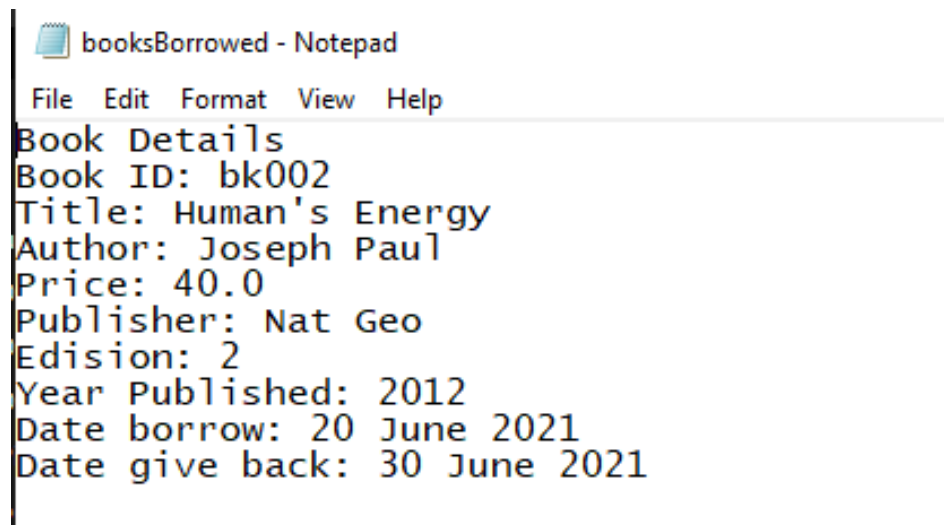
USER MAIN PAGE
(1)Book details or (2)User details or (3)Borrow Book (4)Back to Login Page -> 2
User Details
User ID: NL002
User password: user002
Membership date: 13 May 2019
Contact number: 0104567434
Address: Klang, Selangor

USER MAIN PAGE
(1)Book details or (2)User details or (3)Borrow Book (4)Back to Login Page -> 3

BORROW BOOK PAGE
Enter the book ID that wanted to borrow -> bk002
Date borrow -> 20 June 2021
Date give back -> 30 June 2021
Borrowed book has been recorded

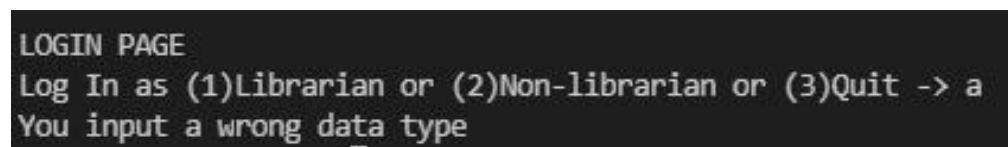
USER MAIN PAGE
(1)Book details or (2)User details or (3)Borrow Book (4)Back to Login Page -> _
```

(d) Write to file of books borrowed



```
booksBorrowed - Notepad
File Edit Format View Help
Book Details
Book ID: bk002
Title: Human's Energy
Author: Joseph Paul
Price: 40.0
Publisher: Nat Geo
Edition: 2
Year Published: 2012
Date borrow: 20 June 2021
Date give back: 30 June 2021
```

(e) Exception



```
LOGIN PAGE
Log In as (1)Librarian or (2)Non-librarian or (3)Quit -> a
You input a wrong data type
```

6. Conclusion

The system provides a computerized version of the library management system that will benefit librarians as well as library users. This is used by librarian to manage library using a computerized system where they can record various tasks such as book information, add new books and users. In addition to computerized system there will be no loss of book records or user accounts. However, those problems are usually occurring when a non -computerized system is used. All these functions of this system will help librarian to manage library in a more efficient way compared to non -computerized library systems.

7. Appendices

```
import java.util.*;
import java.io.*;

abstract class Book {
    private String bookId;
    private String title;
    private String author;
    private double price;
    private String publisher;

    public Book (String bookId, String title, String
author, double price, String publisher) {
        this.bookId=bookId;
        this.title=title;
        this.author=author;
        this.price=price;
        this.publisher=publisher;
    }

    public String getBookId()
    {return bookId;}
    public String getTitle()
    {return title;}
    public String getAuthor()
    {return author;}
    public double getPrice()
    {return price;}
    public String getPublisher()
    {return publisher;}

    public String toString()
    {
        return "Book Details" + "\nBook ID: " + bookId +
            "\nTitle: " + title + "\nAuthor: " + author +
            "\nPrice: " + price + "\nPublisher: " + publisher;
    }
}
```

```

    }

    class Journals extends Book {
        private int edision;
        private int yearPublished;

        public Journals(String bookId, String title, String
author, double price, String publisher, int edision,
int yearPublished){
            super(bookId, title, author, price, publisher);
            this.edision=edision;
            this.yearPublished=yearPublished;
        }

        public int getEdision()
        {return edision;}

        public int getYearPublished()
        {return yearPublished;}

        public String toString() {
            return super.toString() + "\nEdision: " +
edision + "\nYear Published: " + yearPublished;
        }
    }

    class Magazines extends Book{
        private String monthPublished;

        public Magazines(String bookId, String title,
String author, double price, String publisher, String
monthPublished){
            super(bookId, title, author, price, publisher);
            this.monthPublished=monthPublished;
        }

        public String getMonthPublished()

```

```

        {return monthPublished;}

        public String toString() {
            return super.toString() + "\nMonth Published: "
+ monthPublished;
        }
    }

class StudyBooks extends Book{
    private String course;
    private int yearPublished;

    public StudyBooks(String bookId, String title,
String author, double price, String publisher, String
course, int yearPublished){
        super(bookId, title, author, price, publisher);
        this.course=course;
        this.yearPublished=yearPublished;
    }

    public String getCourse()
    {return course;}

    public int getYearPublished()
    {return yearPublished;}

    public String toString() {
        return super.toString() + "\nCourse: " +
course + "\nYear Published: " + yearPublished;
    }
}

class Account {
    private String memberShipDate;
    private String contactNo;
    private String address;

```

```

        public Account(String m, String c, String a) {
            memberShipDate=m;
            contactNo=c;
            address=a;
        }

        public String getMemberShipDate() {
            return memberShipDate;
        }

        public String getContactNo() {
            return contactNo;
        }

        public String getAddress() {
            return address;
        }
    }

    class User {
        private String id;
        private String password;
        private Account acct;          ////composition with
Account class
        Book bk;

        public User(String i, String p, String m, String c,
String a) {
            id=i;
            password=p;
            acct = new Account(m, c, a);
        }

        public String getID() {
            return id;
        }
    }

```

```

        public String getPassword() {
            return password;
        }

        public String toString()
        {
            return "User Details"
                + "\nUser ID: " +id + "\nUser password: "
+password
                + "\nMembership date: "
+acct.getMemberShipDate()
                + "\nContact number: " +acct.getContactNo()
                + "\nAddress: " +acct.getAddress();
        }

        public static void getBookDetails(Vector<Book>
book)
        {
            for (int i=0; i<book.size(); i++) {
                System.out.println(book.get(i).toString());
                System.out.println();
            }
        }

        public static void borrowBook(Vector<Book> book) {
            try {
                FileWriter brwbook=new
FileWriter("booksBorrowed.txt");
                Scanner input=new Scanner(System.in);
                System.out.println("\nBORROW BOOK PAGE");
                System.out.print("Enter the book ID that
wanted to borrow -> ");
                String bookId=input.nextLine();
                for (int b=0; b<book.size(); b++) {

                    if(book.get(b).getBookId().equals(bookId)) {

```

```

        brwbook.write(book.get(b).toString());

        System.out.print("Date borrow ->
");

        String borrowDate=input.nextLine();
        System.out.print("Date give back ->
");

        String giveDate=input.nextLine();
        brwbook.write("\nDate borrow:
"+borrowDate);

        brwbook.write("\nDate give back:
"+giveDate);

    }

}

brwbook.close();

System.out.println("Borrowed book has been
recorded");

}

catch (IOException e) {
    System.out.println("An error occurred.");
    e.printStackTrace();
}

}

}

class Librarian {
    private String name;
    private String password;

    public Librarian(String n, String p) {
        name=n;
        password=p;
    }

    public String getName() {
        return name;
    }
}

```

```

    }

    public String getPassword() {
        return password;
    }

    //public static void searchBook() {
    public static void searchBook(Vector<Book> book) {
        Scanner input=new Scanner(System.in);
        String keyword;
        System.out.println("\nSEARCH PAGE");
        System.out.print("Search by book ID -> ");
        keyword=input.nextLine();
        for (int i=0; i<book.size(); i++) {
            if(book.get(i).getBookId().equals(keyword))
        {

System.out.println(book.get(i).toString());

                break;
            }
        }
    }

    public static Vector<Book> issueBook(Vector<Book>
book) {
        String id, title, author, publisher,
monthPublished, course;
        double price;
        int edision, yearPublished;
        Scanner input=new Scanner(System.in);
        try{
            int kind;
            do {
                System.out.println("\nADD BOOK PAGE");
                System.out.print("(1)Journals (2)Magazines
(3)Studybook (4)Quit adding -> ");
                kind=input.nextInt();

```

```
switch (kind) {
    case 1:
        System.out.print("Book ID -> ");
        input.nextLine();
        id=input.nextLine();
        System.out.print("Title -> ");
        title=input.nextLine();
        System.out.print("Author -> ");
        author=input.nextLine();
        System.out.print("Price -> ");
        price=input.nextDouble();
        System.out.print("Publisher -> ");
        input.nextLine();
        publisher=input.nextLine();
        System.out.print("Edision -> ");
        edision=input.nextInt();
        System.out.print("Year Published -> ");
        yearPublished=input.nextInt();
        book.add(new Journals(id, title,
author, price, publisher, edision, yearPublished));
        System.out.println("Book added.");
        break;

    case 2:
        System.out.print("Book ID -> ");
        input.nextLine();
        id=input.nextLine();
        System.out.print("Title -> ");
        title=input.nextLine();
        System.out.print("Author -> ");
        author=input.nextLine();
        System.out.print("Price -> ");
        price=input.nextDouble();
        System.out.print("Publisher -> ");
        input.nextLine();
        publisher=input.nextLine();
```

```

        System.out.print("Month Published ->
");

        monthPublished=input.nextLine();
        book.add(new Magazines(id, title,
author, price, publisher, monthPublished));
        System.out.println("Book added.");
        break;

case 3:
        System.out.print("Book ID -> ");
        input.nextLine();
        id=input.nextLine();
        System.out.print("Title -> ");
        title=input.nextLine();
        System.out.print("Author -> ");
        author=input.nextLine();
        System.out.print("Price -> ");
        price=input.nextDouble();
        System.out.print("Publisher -> ");
        input.nextLine();
        publisher=input.nextLine();
        System.out.print("Course -> ");
        course=input.nextLine();
        System.out.print("Year Published -> ");
        yearPublished=input.nextInt();
        book.add(new StudyBooks(id, title,
author, price, publisher, course, yearPublished));
        System.out.println("Book added.");
        break;

case 4:
        System.out.println("Going back to
Librarian Work Page...");
        break;

default:

```

```

        System.out.println("INVALID! Enter a
valid input.");
        continue;
    } }while(kind==1||kind==2||kind==3);
    }
    catch (InputMismatchException e)
    {System.out.println("You input a wrong data
type");}
    return book;
}
}

public class LibrarySystem {
    public static void main(String[] args) {
        int librarianNo, libWork, userOptions, kind;
        String libName, libPassw, keyword;
        String userID, userPassw;
        Scanner input=new Scanner(System.in);

        Vector<Book> book=new Vector<Book>();
        book.add(new StudyBooks("bk001", "Object-
Oriented Programming", "Jailani Y.", 20.30, "Sasbadi",
"Programming", 2013));
        book.add(new Journals("bk002", "Human's
Energy", "Joseph Paul", 40.00, "Nat Geo", 2, 2012));

        Vector<Librarian> librarian=new
Vector<Librarian>();
        librarian.add(new Librarian("Srivani",
"lib001"));
        librarian.add(new Librarian("Aqilah",
"lib002"));
        librarian.add(new Librarian("Syamira",
"lib003"));

        Vector<User> user=new Vector<User>();

```

```

        user.add(new User("NL001", "user001", "18 June
2019", "0123456789", "Shah Alam, Selangor"));
        user.add(new User("NL002", "user002", "13 May
2019", "0104567434", "Klang, Selangor"));
        user.add(new User("NL003", "user003", "8 August
2019", "0108732197", "Bangi, Selangor"));

        try{
            int login;
            do {
                System.out.println("\nLOGIN PAGE");
                System.out.print("Log In as (1)Librarian or
(2)Non-librarian or (3)Quit -> ");
                login=input.nextInt();

                switch (login) {
                    //Log in as Librarian
                    case 1:
                        System.out.print("Enter name -> ");
                        input.nextLine();
                        libName=input.nextLine();
                        System.out.print("Enter password -> ");
                        libPassw=input.nextLine();
                        //verify librarian
                        for (int l=0; l<librarian.size(); l++)
                        {

                            if(librarian.get(l).getName().equals(libName)&&libraria
n.get(l).getPassword().equals(libPassw)) {
                                System.out.println("Librarian
logged in.");

                                //librarian work
                                do {

                                    System.out.println("\nLIBRARIAN WORK PAGE");
                                    System.out.print("(1)Search
or (2)Add book or (3)Back to Login Page-> ");

```

```

libWork=input.nextInt();
switch (libWork) {
    case 1:
        //search book

librarian.get(1).searchBook(book);

        continue;

    case 2:
        //add book

book=librarian.get(1).issueBook(book);

        continue;

    case 3:

System.out.println("Going back to Login Page...");

        break;

    default:

System.out.println("INVALID! Enter a valid input.");

        continue;

    }
}while(libWork==1||libWork==2);
}
}
continue;

//Log in as Non-Librarian
case 2:
    System.out.print("Enter ID -> ");
    input.nextLine();
    userID = input.nextLine();
    System.out.print("Enter password
-> ");

    userPassw = input.nextLine();

```

```

        //verify user
        for (int u=0; u<user.size(); u++) {
            if
            (user.get(u).getID().equals(userID)&&user.get(u).getPas
            sword().equals(userPassw)) {
                System.out.println("User logged
            in.");
                do {
                    System.out.println("\nUSER
            MAIN PAGE");
                    System.out.print("(1)Book
            details or (2)User details or (3)Borrow Book (4)Back to
            Login Page -> ");
                    userOptions=input.nextInt();
                    switch (userOptions)
                    {
                        case 1: //get book
            details
            user.get(u).getBookDetails(book);
                                continue;
                        case 2: //get user
            details
            System.out.println(user.get(u).toString());
                                continue;
                        case 3: //borrow book
            user.get(u).borrowBook(book);
                                continue;
                        case 4: //back to login
            page

```

```

        System.out.println("Going back to Login Page...");
        break;

        default:

        System.out.println("INVALID! Enter a valid input.");
        continue;
    }

    }while(userOptions==1||userOptions==2||userOptions==3);
    }
    continue;

    //Quit program
    case 3:
        System.out.println("Thank you!");
        break;

    //invalid input
    default:
        System.out.println("INVALID! Enter a
valid input");
        continue;
    } }while(login==1||login==2);
    }
    catch (InputMismatchException e)
    {System.out.println("You input a wrong data
type");}
    }
}

```