Submitted By Muhammad Nouman Ghauri

CMS 463692

Code:

LibraryManagementSystem Class:

```
import javax.swing.*;
import java.awt.*;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.Scanner;
public class LibraryManagementSystem{
    public static Scanner input = new Scanner(System.in);
    public static void main(String args[]) {

        int temp=0;
        String temp2;
        while (temp!=6) {
            System.out.println("\nPress 1 if you want to add a book");
            System.out.println("Press 2 if you want to add a user");
            System.out.println("Press 3 if you want to search a book");
            System.out.println("Press 4 if you want to checkout a
book");
            System.out.println("Press 5 if you want to return a book");
            System.out.print("Press 6 if you want to exit : ");
            try {
                temp = input.nextInt();
            }
            catch (Exception e){
                temp2=input.nextLine();
                System.out.println("Invalid Input");
                continue;
            }
            if (temp == 1)
                Library.addBook();
            else if (temp == 2)
                Library.addUser();
            else if (temp == 3)
                Book.book_search();
```

```
            else if (temp == 4) {
                User.checkout_book();
            } else if (temp == 5) {
                User.return_book();
            } else if (temp == 6) {
                continue;
            }
            else System.out.println("Invalid Input");
        }
        Library.save();
    }
}
```

Library Class:

```
import java.util.ArrayList;
import java.io.FileWriter;
import java.io.FileReader;
import java.io.BufferedReader;
import java.io.BufferedWriter;
public class Library {
    private static int no_of_books=0, no_of_users = 0;
    static ArrayList<User> Total_users = new ArrayList<>();
    static ArrayList<Book> Total_books = new ArrayList<>();

    public static int getNo_of_users()
    {
        return no_of_users;
    }

    public static int getNo_of_books() {
        return no_of_books;
    }
    public static void addUser() {
        User new_user =new User();
        Total_users.add(no_of_users++,new_user);
        try{
            BufferedWriter writer =new BufferedWriter(new
FileWriter("Users.txt",true));
            writer.write(String.valueOf(new_user.getUser_ID()));
            writer.newLine();
            writer.write(new_user.getUser_name());
            writer.newLine();
            writer.write(String.valueOf(new_user.getContact_number()));
            writer.newLine();
            writer.write(String.valueOf(new_user.getNo_books_taken()));
```

```
            writer.newLine();
            writer.write("||");
            writer.close();

        }catch (Exception e){}
    }
    public static void addBook()
    {
        Book new_book = new Book();
        Total_books.add(no_of_books++,new_book);
        try{
        BufferedWriter writer =new BufferedWriter(new
FileWriter("Books.txt",true));
        writer.write(String.valueOf(new_book.getBook_ID()));
        writer.newLine();
        writer.write(new_book.getTitle());
        writer.newLine();
        writer.write(new_book.getAuthor());
        writer.newLine();
        writer.write(new_book.getGenre());
        writer.newLine();
        writer.write(String.valueOf(new_book.getAvailability_no()));
        writer.newLine();
        writer.close();

        }catch (Exception e){}
    }
    public static void save()
    {
        for (int i=0; i<=no_of_users;i++) {
            try {
                BufferedWriter writer = new BufferedWriter(new
FileWriter("Users.txt", true));

writer.write(String.valueOf(Total_users.get(i).getUser_ID()));
                writer.newLine();
                writer.write(Total_users.get(i).getUser_name());
                writer.newLine();

writer.write(String.valueOf(Total_users.get(i).getContact_number()));
                writer.newLine();

writer.write(String.valueOf(Total_users.get(i).getNo_books_taken()));
                writer.newLine();
                for (int j=0;
j<=Total_users.get(i).getNo_books_taken();j++) {

writer.write(String.valueOf(Total_users.get(i).getBook_taken().get(j)));
                    writer.newLine();
                }
                writer.write("||");
                writer.close();
```

```
                return;
            } catch (Exception e) {
                return;
            }
        }
        for (int i=0; i<=no_of_books;i++){
            try{
                BufferedWriter writer =new BufferedWriter(new
FileWriter("Books.txt",true));

writer.write(String.valueOf(Total_books.get(i).getBook_ID()));
                writer.newLine();
                writer.write(Total_books.get(i).getTitle());
                writer.newLine();
                writer.write(Total_books.get(i).getAuthor());
                writer.newLine();
                writer.write(Total_books.get(i).getGenre());
                writer.newLine();

writer.write(String.valueOf(Total_books.get(i).getAvailability_no()));
                writer.newLine();
                writer.close();

            }catch (Exception e){}
        }
        return;
    }
}
```

User Class:

```
import org.jetbrains.annotations.NotNull;

import java.util.ArrayList;
public class User {
    private int user_ID, no_books_taken = 0;
    private long contact_number;
    private String user_name;
    private ArrayList<Integer> book_taken = new ArrayList<Integer>();

    User() {
        assigntoUser_ID();
        System.out.print("Enter the name of the User : ");
        String temp = LibraryManagementSystem.input.nextLine();
        user_name = LibraryManagementSystem.input.nextLine();
        assigntoUser_name(user_name);
```

```
        assigntoContact_number();
    }

    public void assigntoUser_ID() {
        System.out.print("\nEnter the ID of the user : ");
        try {
            user_ID = LibraryManagementSystem.input.nextInt();
        } catch (Exception e) {
            System.out.print("Invalid user ID.");
            String temp = LibraryManagementSystem.input.nextLine();
            assigntoUser_ID();
        }

        if (user_ID < 0) {
            System.out.print("Invalid user ID.");
            assigntoUser_ID();
            return;
        }
    }

    public void assigntoContact_number() {
        System.out.print("Enter the contact number of the user : ");
        try {
            contact_number = LibraryManagementSystem.input.nextLong();
        } catch (Exception e) {
            System.out.print("Invalid contact.");
            String temp = LibraryManagementSystem.input.nextLine();
            assigntoUser_ID();
        }

        if (contact_number < 0) {
            System.out.print("Invalid user ID.");
            assigntoContact_number();
            return;
        }
    }

    private void assigntoUser_name(String user_name) {
        if (user_name.isBlank()) {
            System.out.print("Invalid author name.\nRe-enter the author
name : ");
            String temp = LibraryManagementSystem.input.nextLine();
            user_name = LibraryManagementSystem.input.nextLine();
            assigntoUser_name(user_name);
            return;
        }
    }



    public int getUser_ID() {
        return user_ID;
```

```
    }

    public ArrayList<Integer> getBook_taken() {
        return this.book_taken;
    }

    public long getContact_number() {
        return contact_number;
    }

    public String getUser_name() {
        return user_name;
    }

    public int getNo_books_taken() {
        return no_books_taken;
    }

    public static int user_search(int user_ID)//returns index of the
user in the arraylist from the user id
    {
        int i = 0;
        for (; i < Library.getNo_of_users(); i++) {
            if (Library.Total_users.get(i).getUser_ID() == user_ID)
                return i;
        }
        if (i > Library.getNo_of_users()) {
            System.out.println("The user doesn't exist");
            return -1;
        }
        return -1;
    }



    public static void return_book(){
        int user_id=0, book_id=0;
        System.out.println("Enter the User id ");
        try{
            user_id= LibraryManagementSystem.input.nextInt();
        }
        catch (Exception e){
            System.out.print("Invalid book ID.");
            checkout_book();
        }
```

```
        if (user_id<0) {
            System.out.print("Invalid book ID.");
            checkout_book();
            return;
        }
        System.out.println("Enter the Book id");
        try{
            book_id= LibraryManagementSystem.input.nextInt();
        }
        catch (Exception e){
            System.out.print("Invalid book ID.");
            checkout_book();
        }

        if (book_id<0) {
            System.out.print("Invalid book ID.");
            checkout_book();
            return;
        }
        int user_index = user_search(user_id);
        if (user_index<0)
        {
            System.out.println("The user doesn't exist.");
            return;
        }
        int book_index = Book.book_search_by_id(book_id);
        if (book_index<0)
        {
            System.out.println("The book doesn't exist.");
            return;
        }
        int i=0,temp=0;
        for(;i<=
Library.Total_users.get(user_index).book_taken.size();i++)
        {

temp=Library.Total_users.get(user_index).book_taken.get(book_index);
            if(temp==book_id)
                break;
        }
        if (i>Library.Total_users.get(user_index).book_taken.size())
        {
            System.out.println("The user doesn't have this book");
            return;}

        else
        {
            Library.Total_users.get(user_index).book_taken.remove(i);
            Library.Total_users.get(user_index).no_books_taken--;
            Library.Total_books.get(book_index).setAvailability_no(1);

        }
```

```
    }
    public static void checkout_book(){
        int user_id=0, book_id=0;
        System.out.println("Enter the User id ");
        try{
            user_id= LibraryManagementSystem.input.nextInt();
        }
        catch (Exception e){
            System.out.print("Invalid book ID.");
            checkout_book();
        }

        if (user_id<0) {
            System.out.print("Invalid book ID.");
            checkout_book();
            return;
        }
        System.out.println("Enter the Book id");
        try{
            book_id= LibraryManagementSystem.input.nextInt();
            System.out.print("Input ha ya");
        }
        catch (Exception e){
            System.out.print("Invalid book ID.");
            checkout_book();
        }

        if (book_id<0) {
            System.out.print("Invalid book ID.");
            checkout_book();
            return;
        }
        int user_index = user_search(user_id);
        if (user_index<0)
        {
            System.out.println("The user doesn't exist.");
            return;
        }
        int book_index = Book.book_search_by_id(book_id);
        if (book_index<0)
        {
            return;
        }

Library.Total_users.get(user_index).book_taken.add(Library.Total_users.g
et(user_index).no_books_taken++, book_id);
```

```
        Library.Total_books.get(book_index).setAvailability_no(-1);
    }
}
```

Book Class:

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;

public class Book{
    private int book_ID,availability_no;
    private String title,author,genre;
    Book()
    {
        assigntoBook_ID();
        title = LibraryManagementSystem.input.nextLine();
        System.out.print("Enter the name of the book : ");
        title = LibraryManagementSystem.input.nextLine();
        assigntoTitle(title);

        System.out.print("Enter the name of the author : ");
        author = LibraryManagementSystem.input.nextLine();
        assigntoAuthor(author);

        System.out.print("Enter the genre of the book : ");
        genre = LibraryManagementSystem.input.nextLine();
        assigntoGenre(genre);

        assigntoAvailability_no();
    }

    //setters
    private void assigntoAvailability_no() {
        System.out.print("Enter the number of books available : ");
        try{
            availability_no= LibraryManagementSystem.input.nextInt();}
        catch (Exception e){
            System.out.print("Invalid no of copies.");
            String temp = LibraryManagementSystem.input.nextLine();
            assigntoAvailability_no();
        }
        if (availability_no<0) {
            System.out.print("Invalid no of copies.\nRe-entry the
availability number : ");
            String temp = LibraryManagementSystem.input.nextLine();
            assigntoAvailability_no();
            return;
        }
```

```
    }

    private void assigntoBook_ID() {
        System.out.print("\nEnter the ID of the book : ");
        try{
            book_ID= LibraryManagementSystem.input.nextInt();
        }
        catch (Exception e){
            System.out.print("Invalid book ID.");
            String temp = LibraryManagementSystem.input.nextLine();
            assigntoBook_ID();
        }

        if (book_ID<0) {
            System.out.print("Invalid book ID.");
            assigntoBook_ID();
            return;
        }
    }

    private void assigntoAuthor(String author) {
        if(author.isBlank()) {
            System.out.print("Invalid author name.\nRe-enter the author
name : ");
            author = LibraryManagementSystem.input.nextLine();
            assigntoAuthor(author);
            return;
        }
    }

    private void assigntoTitle(String title) {
        if(title.isBlank())
        {
            System.out.print("Invalid Title.\nRe-enter the title : ");
            title = LibraryManagementSystem.input.nextLine();
            assigntoTitle(title);
            return;
        }
    }

    private void assigntoGenre(String genre) {
        if(genre.isBlank())
        {
            System.out.print("Invalid Genre.\nRe-enter the genre : ");
            genre = LibraryManagementSystem.input.nextLine();
            assigntoGenre(genre);
```

```
            return;
        }
    }

    //getters
    public String getAuthor() {
        return author;
    }

    public String getTitle() {
        return title;
    }

    public String getGenre() {
        return genre;
    }

    public int getBook_ID() {
        return book_ID;
    }

    public int getAvailability_no() {
        return availability_no;
    }

    public void setAvailability_no(int the_changed_number)
    {
        if (availability_no==0&&the_changed_number<0)
        {
            System.out.println("This book is already not available.");
            return;
        }
        this.availability_no+=the_changed_number;
        System.out.println("Available number of this book has been
changed.");
    }

    public static void display_book(int book_index)
    {
        if(book_index<0)
            return;
        System.out.println("Title : " +
Library.Total_books.get(book_index).title);
        System.out.println("Author : " +
Library.Total_books.get(book_index).author);
        System.out.println("Copies Available : " +
Library.Total_books.get(book_index).availability_no);
    }

    public static void update_book_file(int book_ID,int add_sub)
    {
        int temp,i;
```

```
        String line;
        try {
            BufferedReader reader=new BufferedReader(new
FileReader("Books.txt"));


            for (i =1;(line = reader.readLine()) != null;i++) {

                temp=Integer.parseInt(line);
                if (temp == book_ID) {
                    break;
                }
                else
                    for (int j=i;i<j+4;i++)
                {
                    reader.readLine();
                }
            }
            BufferedWriter writer =new BufferedWriter(new
FileWriter("Books.txt",true));

        }
        catch (Exception e){}
    }


    public static int book_search_by_id(int book_ID)//returns index of
the book in the arraylist from the book id
    {
        int i = 0;
        for (; i <= Library.getNo_of_books(); i++) {
            if (Library.Total_books.get(i).getBook_ID() == book_ID)
                return i;
        }
        if (i > Library.getNo_of_books()) {
            System.out.println("No books by this Name");
            return -1;
        }
        return -1;
    }
    public static void book_search() {
        int temp=0;
        System.out.print("Press 1 if you want to search book by
name.\n");
        System.out.print("Press 2 if you want to search book by
author.\nEnter your choice : ");
```

```
        try{
            temp= LibraryManagementSystem.input.nextInt();
        }
        catch (Exception e){
            System.out.println("Invalid book ID.Try again");
            book_search();
        }
        if(temp==1){
            book_search_by_name();
        }
        else if(temp==2){
            book_search_by_author();
        }
        else
        {
            System.out.println("Invalid book ID.Try again");
            book_search();
        }
    }


    public static void book_search_by_name()
    {
        String book_name=null,temp;
        System.out.print("Enter the name of the book : ");
        temp=LibraryManagementSystem.input.nextLine();
        book_name = LibraryManagementSystem.input.nextLine();
        if(book_name.isBlank()) {
            System.out.println("Invalid title.");
            book_search_by_name();
            return;
        }
        int i = 0;

        for (; i < Library.getNo_of_books(); i++) {
            if (i >= Library.getNo_of_books()) {
                break;
            }
            if
(Library.Total_books.get(i).title.compareTo(book_name)==0) {
                display_book(i);
                break;
            }
        }

    }
    public static void book_search_by_author()
    {
        String author_name=null,temp;
        System.out.print("Enter the name of the author : ");
        temp=LibraryManagementSystem.input.nextLine();
        author_name = LibraryManagementSystem.input.nextLine();
```

```
        if(author_name.isBlank()) {
            System.out.println("Invalid title.");
            book_search_by_author();
            return;
        }
        int i = 0;

        for (; i < Library.getNo_of_books(); i++) {
            if (i >= Library.getNo_of_books()) {
                break;
            }
            if
(Library.Total_books.get(i).author.compareTo(author_name)==0) {
                display_book(i);
                break;
            }
        }
    }

}
```

Output screen Shot:

Main Screen:

```
Press 1 if you want to add a book
Press 2 if you want to add a user
Press 3 if you want to search a book
Press 4 if you want to checkout a book
Press 5 if you want to return a book
Press 6 if you want to exit :
```

Adding Book:

```
Enter the ID of the book : The perks of being a wallflower
Invalid book ID.
Enter the ID of the book : 12
Enter the name of the book : The perks of being a wallflower
Enter the name of the author : Aurther McDonadls
Enter the genre of the book : Horror
Enter the number of books available : 4
```

Adding User:

```
Enter the ID of the user : 32
Enter the name of the User : Ali
Enter the contact number of the user : 03236640695
```

Checking Out Books:

```
Enter the User id
32
Enter the Book id
12
```

Search Book:

```
Press 1 if you want to add a book
Press 2 if you want to add a user
Press 3 if you want to search a book
Press 4 if you want to checkout a book
Press 5 if you want to return a book
Press 6 if you want to exit : 3
Press 1 if you want to search book by name.
Press 2 if you want to search book by author.
Enter your choice : 1
Enter the name of the book : The perks of being a wallflower
Title : The perks of being a wallflower
Author : Aurhtor McDonadls
Copies Available : 3
```

Repository Link: [noumanghauri/Library-Management-System (github.com)]