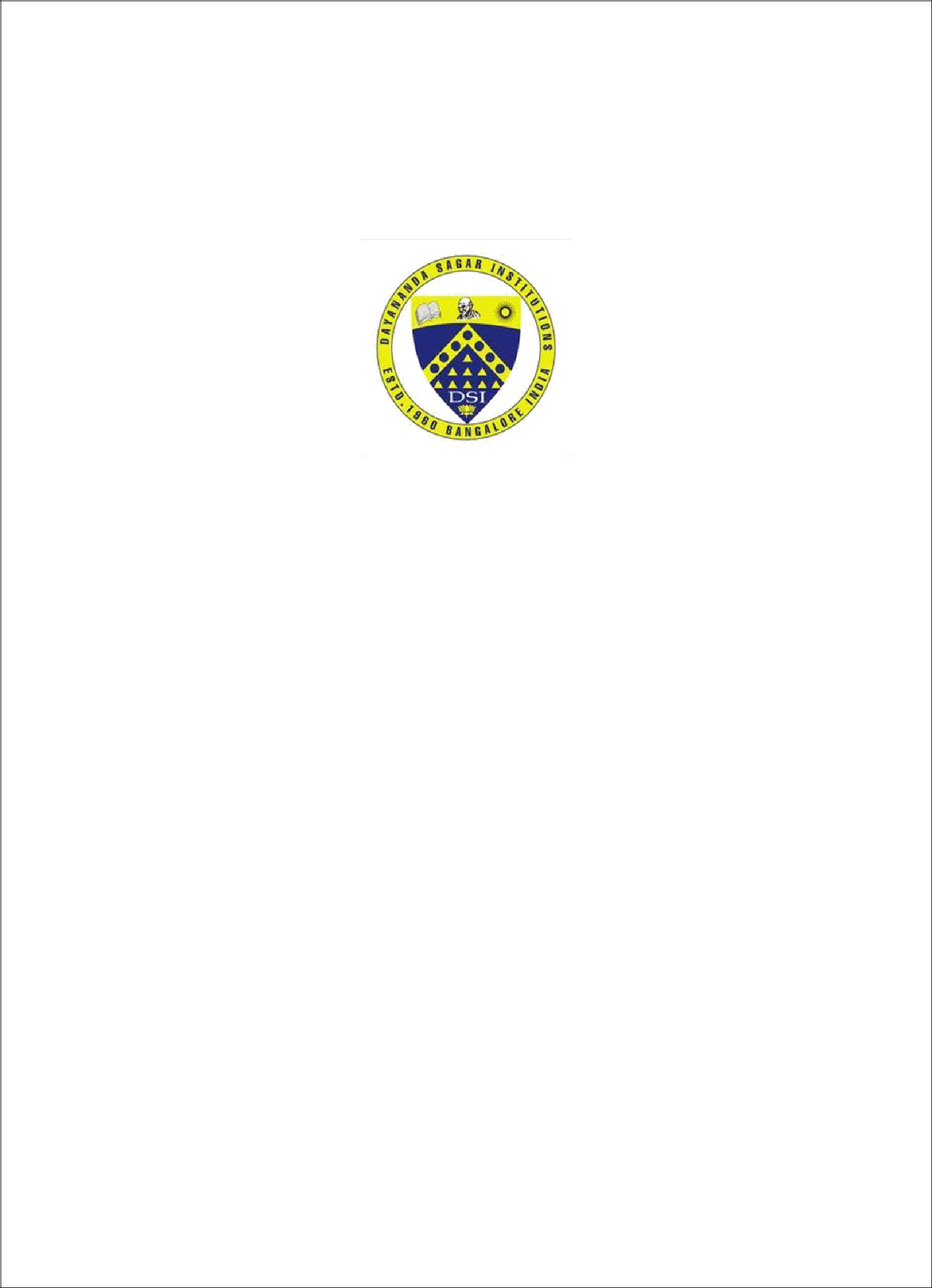
**DAYANANDA SAGAR COLLEGE OF ENGINEERING**

(An Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE & ISO 9001:2008 Certified)

Accredited by National Assessment & Accreditation Council (NAAC) with ‘A’ grade, Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078.

**Mini Project Report**

**On**

**LIBRARY MANAGEMENT SYSTEM**

Submitted By

**Abhinav Bakshi** **[1DS18CS004]**

**Aniruddh Dubey [1DS18CS020]**

**Akhil Sharma** **[1DS18CS015]**

**Aniket Sengar** **[1DS18CS019]**

**[Third Semester B.E (CSE)]**

**in**

**JAVA**

Under the guidance of

**Shravya A R**

**Assistant Professor**

**Dept. of CSE**

**DSCE, Bangalore**

`

**Department of Computer Science and Engineering**

**Dayananda Sagar College of Engineering**

**Bangelore-78**

SYNOPSIS

The main objective of this project on Library Management System is to manage the details of books. It manages all the information about a book (ID, Name, Author, Issued or not). The purpose of the project is to build an application program to reduce the manual work for managing the books and issues. It tracks all the details about the Issues and books.

APPLICATION

* To manage a library.
* Insert a new book in the library.
* Delete a book from the library.
* Issue a book from the library.
* Return a book to the library.
* Display all available and issued books at that particular time.

SAMPLE CODE

package main;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.Map;

import java.util.TreeMap;

public class maintainlibrary

{

static InputStreamReader isr = new InputStreamReader(System.in);

static BufferedReader br = new BufferedReader(isr);

static Map<Long, bookdetails> bm = new TreeMap<Long, bookdetails>();

static Book BookObj = new Book();

static Long index = 0l;

public static void main(String[] args) throws IOException

{

System.out.println(" Welcome to the Library:)");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

mainMenu();

while(true)

{

System.out.println("\*\*\*\*\*\*\*\*\* Please Enter your Option \*\*\*\*\*\*\*\*\*");

try

{

Integer selectedOption = Integer.valueOf(br.readLine());

action(selectedOption);

}

catch (Exception e)

{

System.out.println("Error Input : "+e);

continue;

}

}

}

private static void mainMenu()

{

System.out.println("\*\*\*Please select your option to take action\*\*\*");

System.out.println("Press '1' To Display Existing Available Books");

System.out.println("Press '2' To Display Existing Issued Books");

System.out.println("Press '3' To Add New Book");

System.out.println("Press '4' Issue a Book");

System.out.println("Press '5' Return a Book");

System.out.println("Press '6' Delete a Book");

System.out.println("Press '7' To Exit Application");

}

private static void action(Integer selectedOption) throws IOException{

//pass selected input parameter

switch(selectedOption)

{

case 1:

System.out.println("\*\*\*\*Display Existing Available Books\*\*\*\*");

//Call method to display library information

displayAvailableBookInfo(bm);

break;

case 2:

System.out.println("\*\*\*\*Display Existing Issued Books\*\*\*\*");

//call method to display issued books

displayIssuedBookInfo(bm);

//Call method to display main menu

mainMenu();

break;

case 3:

System.out.println("\*\*\*\*Add New Book\*\*\*\*");

//Call method to add Book

addBook();

//Call method to display main menu

mainMenu();

break;

case 4:

System.out.println("\*\*\*\*Issue a Book\*\*\*\*");

//Call method to issue Book

issueBook();

//Call method to display main menu

mainMenu();

break;

case 5:

System.out.println("\*\*\*\*Return a Book\*\*\*\*");

//Call method to return Book

returnBook();

//Call method to display main menu

mainMenu();

break;

case 6:

System.out.println("\*\*\*\*Delete a Book\*\*\*\*");

//Call method to issue Book

deleteBook(bm);

//Call method to display main menu

mainMenu();

break;

case 7:

System.out.println("Application Stopped......Have a Nice Day! :)");

//Call method to stop application

stopApplication();

//Call method to display main menu

mainMenu();

break;

default:

System.out.println("Option Not Recognized!");

//Call method to display main menu

mainMenu();

break;

}

}

private static void stopApplication()

{

System.exit(0);

}

public static void displayAvailableBookInfo(Map<Long, bookdetails> bm)

{

//Check Map empty

if(bm.isEmpty())

{

System.out.println("No Book Added Yet!");

return;

}

else

{

//Call method to display book

BookObj.displayAvailableBookInfo(bm);

}

}

public static void displayIssuedBookInfo(Map<Long, bookdetails> bm)

{

//Check Map empty

if(bm.isEmpty())

{

System.out.println("No Book Added Yet!");

return;

}

else

{

//Call method to display book

BookObj.displayIssuedBookInfo(bm);

}

}

private static Boolean addBook() throws IOException

{

try

{

//Set default values

bookdetails bd = new bookdetails();

bd.setBookID(++index);

bd.setIssued(false);

//get Book details as input

System.out.println("Enter Book Title -");

bd.setTitle(br.readLine());

System.out.println("Enter Book Author -");

bd.setAuthor(br.readLine() );

System.out.println("Enter Publisher -");

bd.setPublisher(br.readLine());

return BookObj.addBook(bd,bm);

}

catch(NumberFormatException nfe)

{

nfe.getMessage();

}

return false;

}

private static Boolean issueBook() throws IOException

{

//Check Map empty

if(bm.isEmpty())

{

System.out.println("No Book Added Yet!");

return false;

}

else

{

try

{

displayAvailableBookInfo(bm);

System.out.println("Enter BookID from above to issue");

Long bookID = Long.valueOf(br.readLine());

if(bm.containsKey(bookID))

{

//update book

bm.get(bookID).setIssued(true);

}

else

{

System.out.println("BookID Not available!");

//return true once book issues

return true;

}

//return true once book issues

System.out.println("Book has been Issued.");

return true;

}

catch(NumberFormatException e)

{

e.getMessage();

}

}

return false;

}

private static Boolean returnBook() throws IOException

{

//Check Map empty

if(bm.isEmpty())

{

System.out.println("No Book Added Yet!");

return false;

}

else

{

try

{

displayIssuedBookInfo(bm);

System.out.println("Enter BookID from above to return");

Long bookID = Long.valueOf(br.readLine());

if(bm.containsKey(bookID))

{

//update book

bm.get(bookID).setIssued(false);

}

else

{

System.out.println("BookID Not available!");

//return true once book issues

return true;

}

//return true once book issues

System.out.println("Book has been Returned.");

return true;

}

catch(NumberFormatException e)

{

e.getMessage();

}

}

return false;

}

private static Boolean deleteBook(Map<Long,bookdetails> bm) throws IOException{

//Check Map empty

if(bm.isEmpty()){

System.out.println("No Book Added Yet!");

return false;

}else{

try{

displayAvailableBookInfo(bm);

System.out.println("Enter BookID from above to delete");

Long bookID = Long.valueOf(br.readLine());

if(bm.containsKey(bookID)){

//Call method to delete book

BookObj.deleteBook(bookID, bm);

}else{

System.out.println("BookID Not available!");

//return true once book issues

return true;

}

//return true once book issues

return true;

}catch(NumberFormatException nfe){

nfe.getMessage();

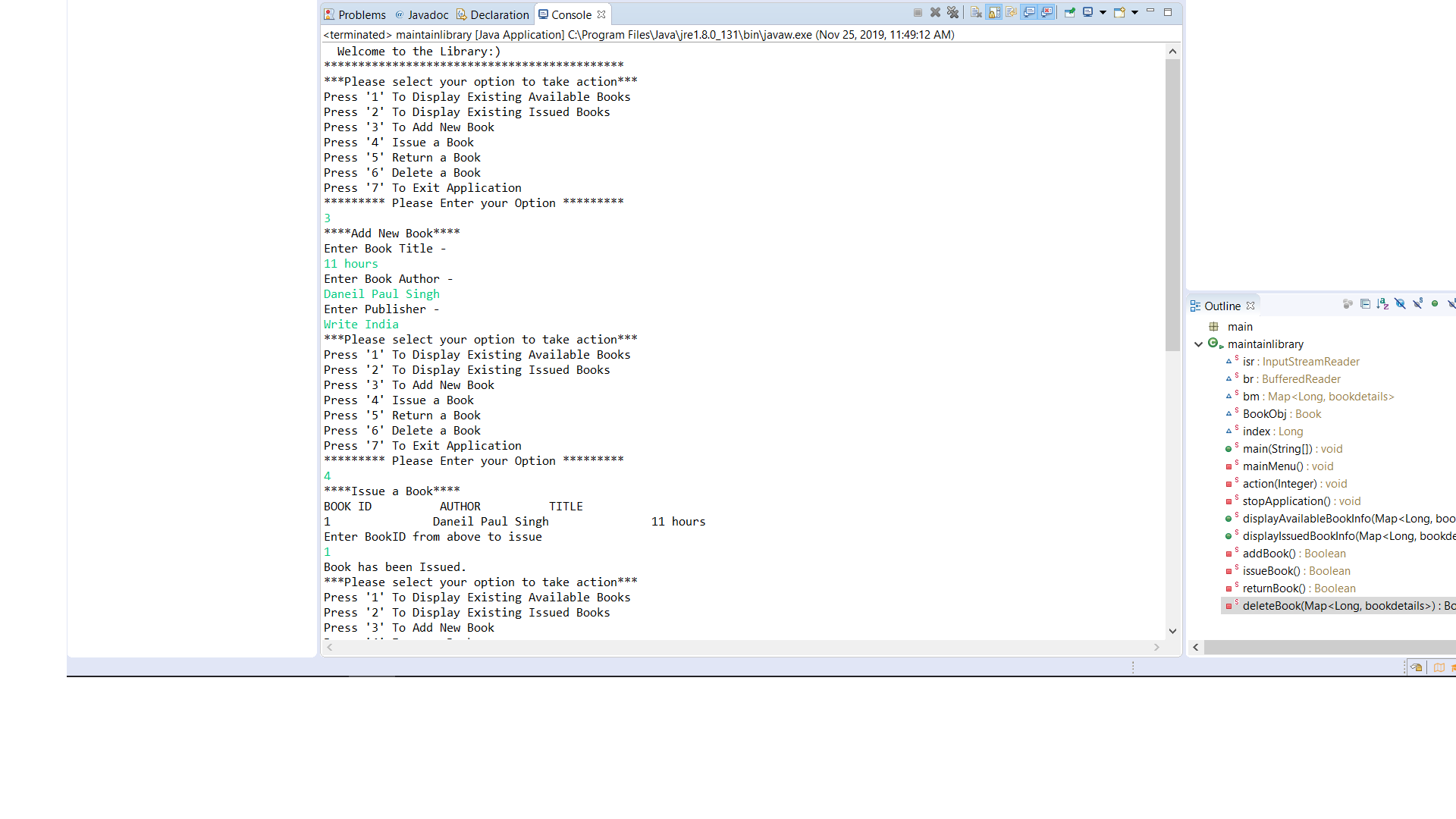
}

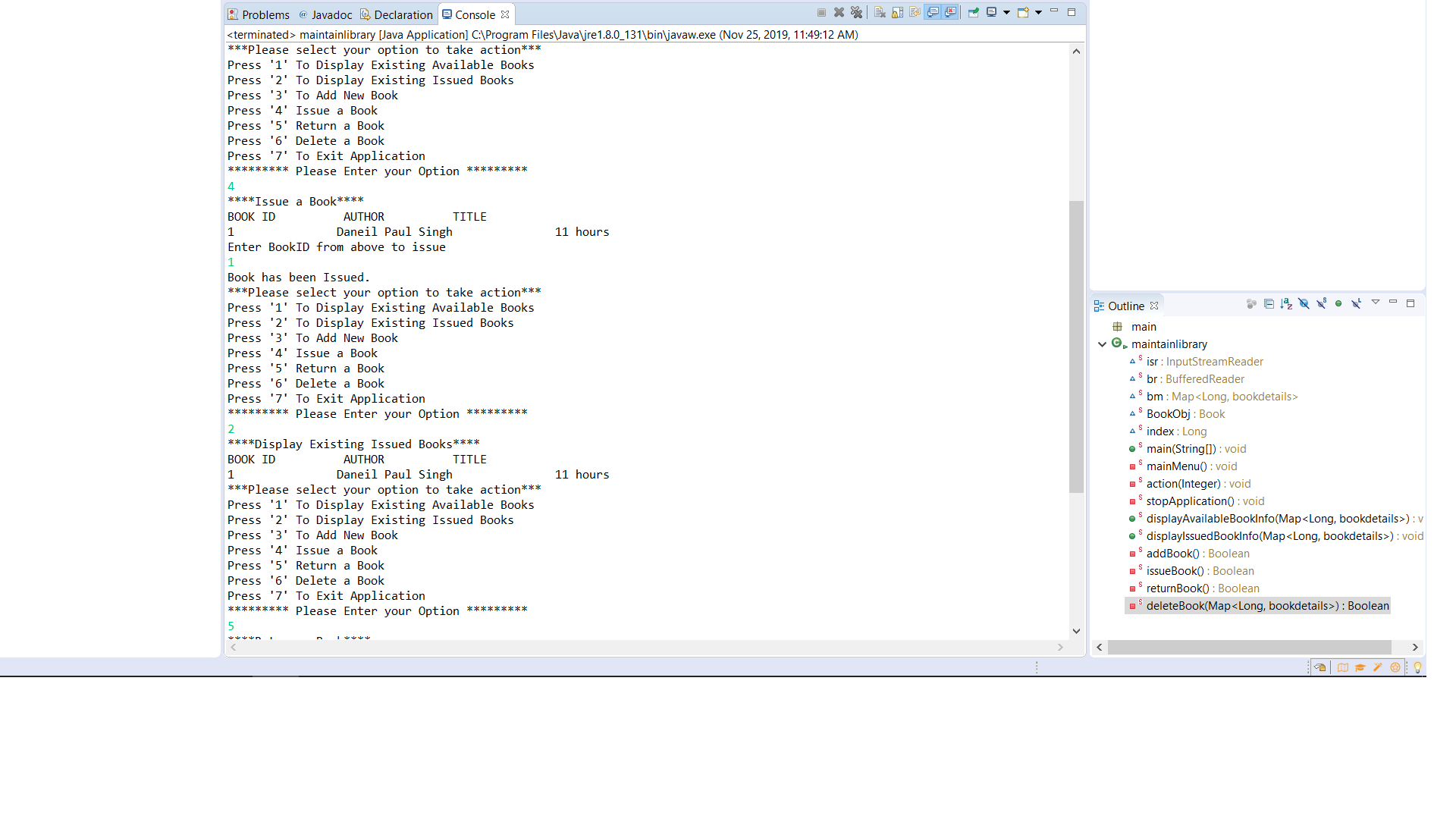
}

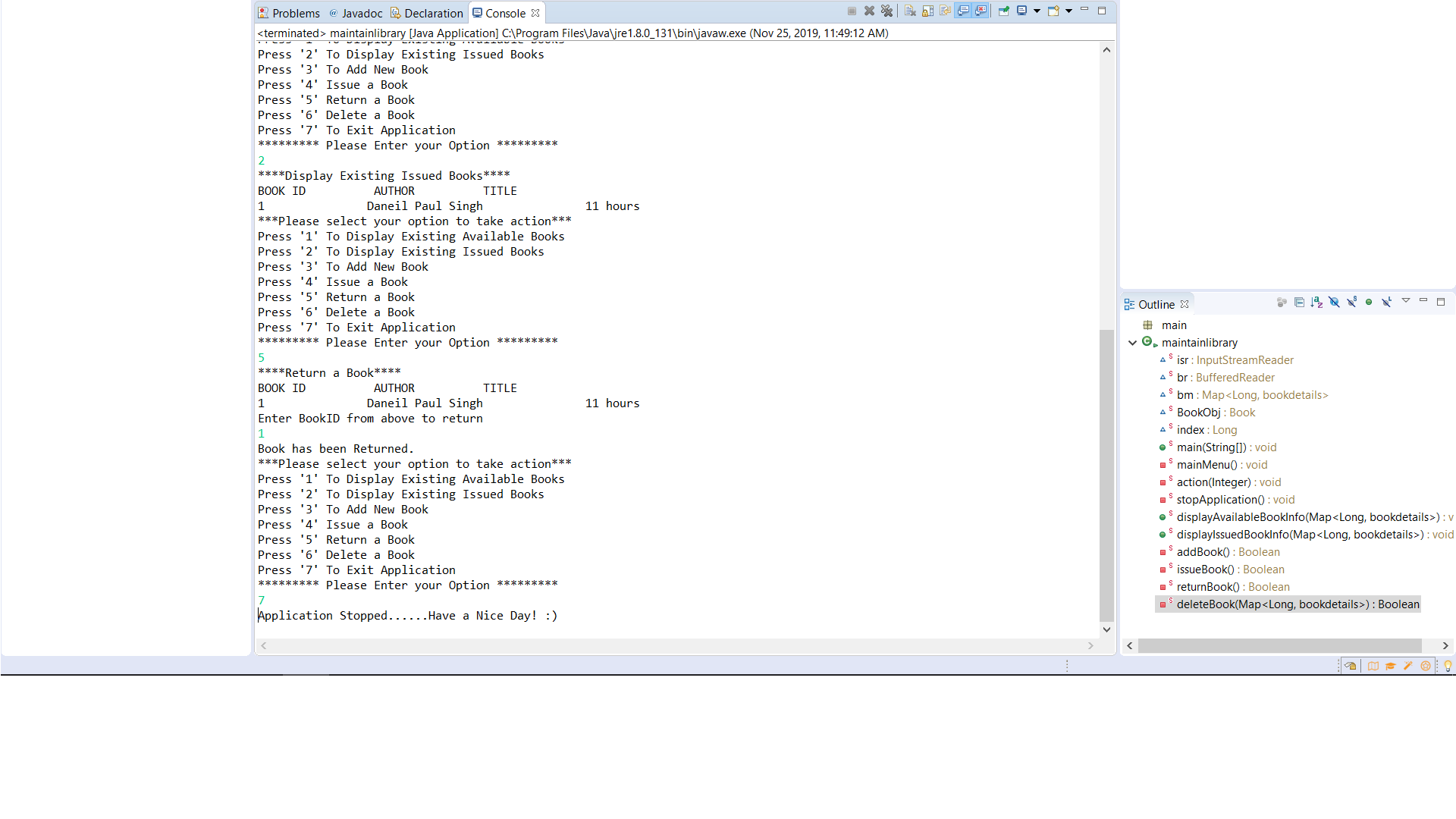
return false;

}

}

OUTPUT:





DESIGN AND IMPLEMENTATION

We have used tree maps to store the details of a book with Book ID as key and other details as its values. Once a book is inserted into the library a new entry in this tree map gets created. If the book is issued we change the Boolean variable issued to true otherwise it is false as default. On deleting a book we just directly delete the whole entry from the map. If the issued book is returned we again change the issued variable back to false. While displaying we display each entry in the tree map.

FUTURE ENHANCEMENTS

* We can make use of database to store book details thus helping in retaining them.
* We can make it more diverse by making it to manage different library branches of a central library.