

A PROJECT REPORT

SUBMITTED BY

**Batch B Group 15**

**MEMBERS**

**YESHWANTH BALAJI**

**(CB.EN.U4AIE22102)**

**ASWIN RAVIDEV**

**(CB.EN.U4AIE22108)**

**KISHOR S**

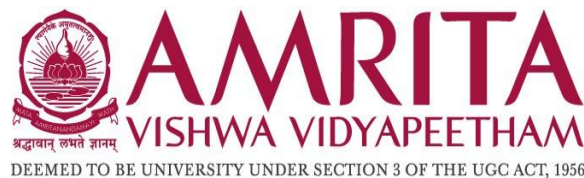
**(CB.EN.U4AIE22128)**

**SARVESH K**

**(CB.EN.U4AIE22153)**

*As a part of the subject*

**DATA STRUCTURES AND ALGORITHMS - 1**



**Centre for Computational Engineering and  
Networking**

**AMRITA SCHOOL OF ARTIFICIAL  
INTELLIGENCE  
AMRITA VISHWA VIDYAPEETHAM**

**COIMBATORE - 641 112**

**(INDIA)**

**July - 2023**

**AMRITA SCHOOL OF ARTIFICIAL  
INTELLIGENCE**  
**AMRITA VISHWA VIDYAPEETHAM**  
**COIMBATORE - 641 112**

**DECLARATION**

We hereby declare that our Project Report is a bonafide record of the project work which we have submitted to Amrita School of Computing, in partial fulfillment of the credit requirements for the degree of B.Tech in Artificial Intelligence is our authentic work. This project report has not been copied, duplicated, or plagiarized from any other paper, journal, document, or book.

This is an authentic piece of work and in case there is any query regarding the same, we shall be held responsible for answering any queries in this regard.

Place: Ettimadai  
Date: 05-07-2023

## **Acknowledgement**

We would like to express our sincere gratitude to our project subject handler, Dr. Pratiti Bhadra, for her invaluable guidance, support, and encouragement throughout the duration of this project. Her expertise and knowledge in the field were instrumental in the success of this project.

We would like to express our profound gratitude to Dr. Soman K P of the CEN department for his contributions to the completion of our project.

This project would not have been possible without the help and guidance of every member of our faculty.

## TABLE OF CONTENTS

<b>TABLE OF FIGURES.....</b>	<b>5</b>
<b>LIBRARY MANAGEMENT SYSTEM.....</b>	<b>6</b>
<b>1. What are we going to solve and why? .....</b>	<b>6</b>
<b>2. Methodology .....</b>	<b>7</b>
<b>2.1 The data Structure we have implemented : .....</b>	<b>7</b>
<b>2.2 The main advantages of using BST's over other Data Structures: .....</b>	<b>8</b>
<b>2.3 BST's also have their own set of limitations : .....</b>	<b>8</b>
<b>3. Result : .....</b>	<b>9</b>
<b>4. Analysis and Discussion: .....</b>	<b>12</b>
<b>4.1 Time complexity of our program:.....</b>	<b>12</b>
<b>4.2 The limitations of our program: .....</b>	<b>12</b>
<b>4.3 Converting our project into a better version:.....</b>	<b>13</b>
<b>5. Appendix:.....</b>	<b>14</b>

## TABLE OF FIGURES.

Figure 1A Binary Search Tree .....	7
Figure 2 Terminal .....	9
Figure 3 User Registration .....	10
Figure 4 Adding Books .....	11
Figure 5 Issuing Books .....	11
Figure 6 Exception Handled .....	11

## **LIBRARY MANAGEMENT SYSTEM**

### **1. What are we going to solve and why?**

Our main objective in this project is to create a library management system wherein students can issue books and the admin or librarian can update/delete the record of books kept in the library. So we have implemented the system into two parts: from the user's perspective and from the admin's perspective. Our program's main aim is to manage a library's operations efficiently. First of all, the admin must login to handle the accounts where the username and password are already set. After he has logged in successfully, he can add, delete and update the books. He can add any new book in the already existing list of books. Similarly he can also delete any existing book. By implementing this program we believe it'll be helpful in reducing the orthodox way of organizing the library which is done manually by persons in real life, as a solution to this we've implemented this simple library management which makes the job easier for both the user as well the admin.

## 2. Methodology

### 2.1 The data Structure we have implemented :

We have used Binary Search Tree in our Java program, it is used to search for a particular subject where the searching operation performs an in-order traversal on the created binary search tree to get the elements in sorted order. We have implemented our Binary Trees in Java using something similar to linked lists node which distinguishes itself from the regular Linked Lists

The reason why we used Binary Search Tree over other Data Structures is because of the time complexity.

Insertion : We use insertion operation to add books in the binary search tree. And the tree will contain the name of the books added

Deletion: We use deletion operation to delete the node of that particular book from the binary search tree. After the node is deleted, the remaining elements get rearranged in the tree.

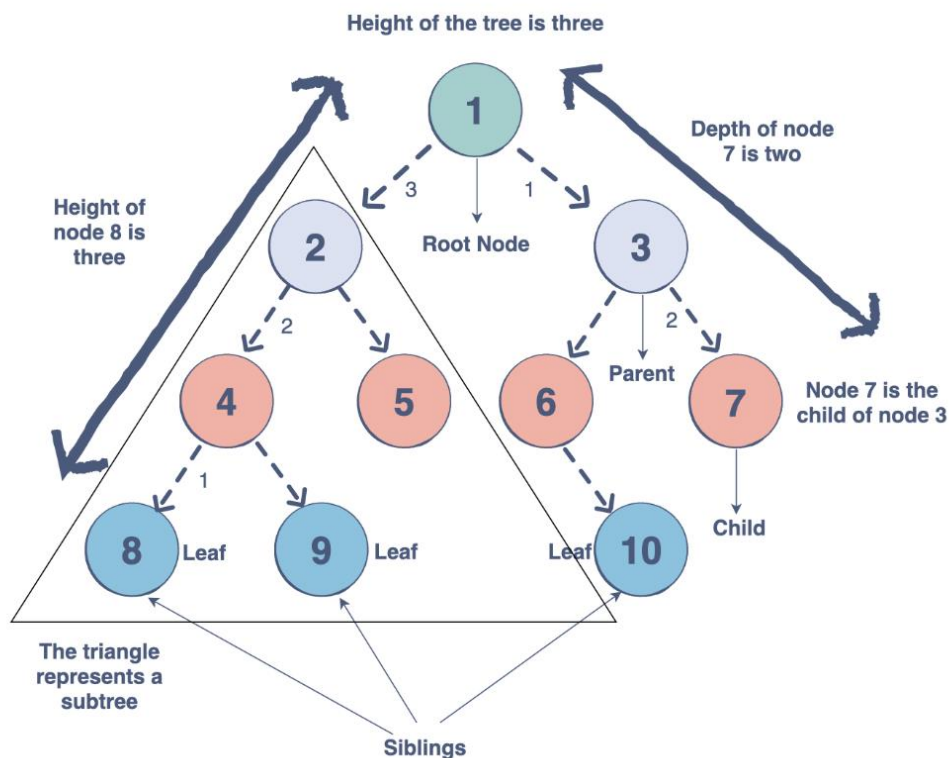


Figure 1 A Binary Search Tree

## **2.2 The main advantages of using BST's over other Data Structures:**

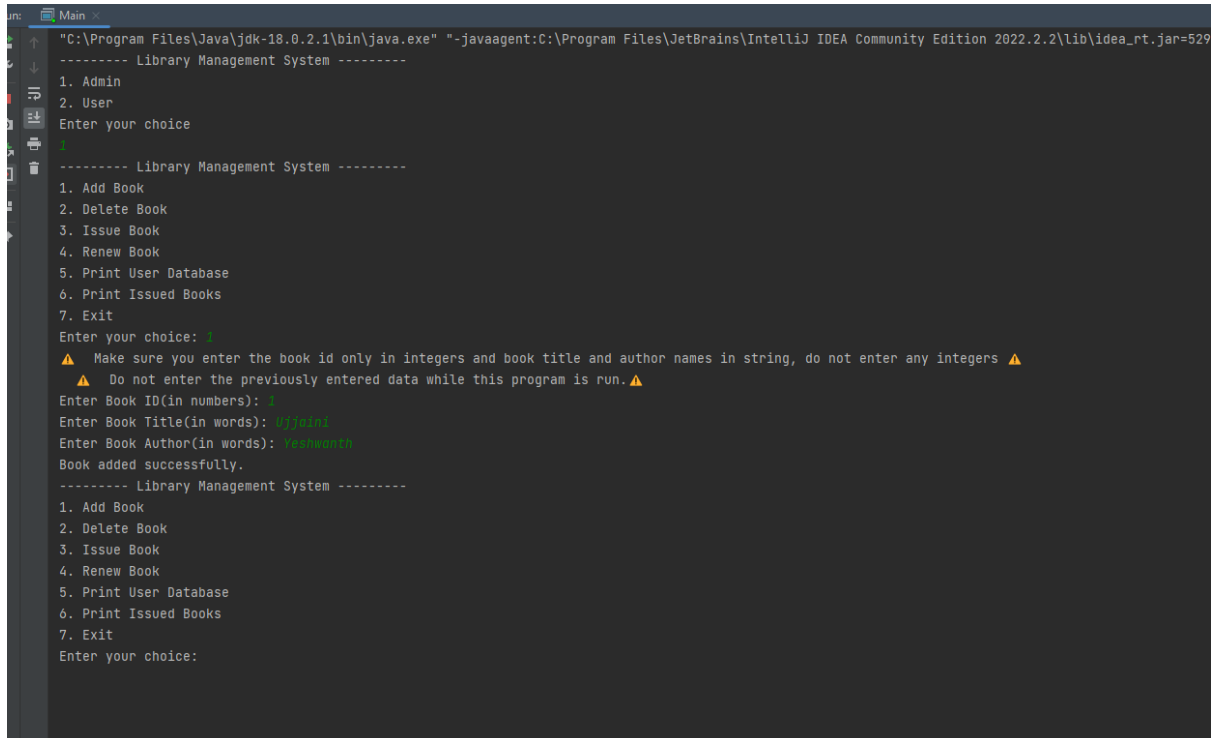
- It provides an efficient search operation with a time complexity of  $O(\log n)$  on average. This makes them ideal for applications where fast searching is required.
- It also provides insertion and deletion function at the ease, compared to other data structures BST's provide faster insertion and deletion times, especially when the data needs to be sorted .
- **Dynamic Size:** Unlike other data structures like arrays who have fixed size data structures, BST's can dynamically grow or shrink as the elements are inserted and removed. This makes BST's more flexible when dealing with varying amounts of data.

## **2.3 BST's also have their own set of limitations :**

- Binary Search Trees are unsuitable for storing large data sets due to their unbalanced nature. As the height grows, resulting in slower search and insertion operations.
- Only a balanced Binary Search Tree can be implemented, otherwise the cost of the operations may not be logarithmic and degenerate into a linear search on an array.
- They also do not support certain operations that are possible with ordered data structures.

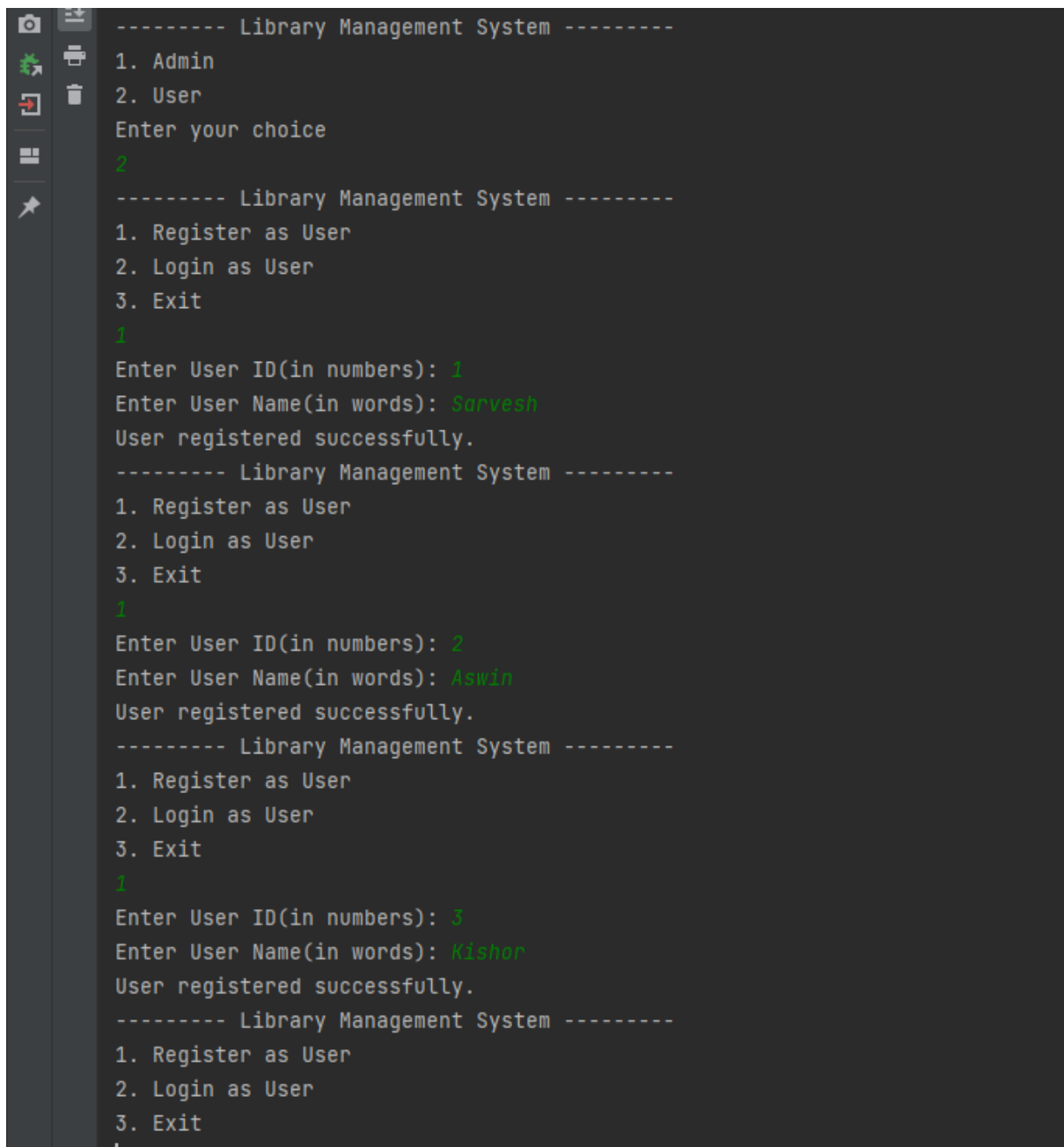


### 3. Result :



```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.2.2\lib\idea_rt.jar=529
----- Library Management System -----
1. Admin
2. User
Enter your choice
1
----- Library Management System -----
1. Add Book
2. Delete Book
3. Issue Book
4. Renew Book
5. Print User Database
6. Print Issued Books
7. Exit
Enter your choice: 1
▲ Make sure you enter the book id only in integers and book title and author names in string, do not enter any integers ▲
▲ Do not enter the previously entered data while this program is run.▲
Enter Book ID(in numbers): 1
Enter Book Title(in words): Ujjaini
Enter Book Author(in words): Yashwanth
Book added successfully.
----- Library Management System -----
1. Add Book
2. Delete Book
3. Issue Book
4. Renew Book
5. Print User Database
6. Print Issued Books
7. Exit
Enter your choice:
```

Figure 2 Terminal



```
----- Library Management System -----
1. Admin
2. User
Enter your choice
2
----- Library Management System -----
1. Register as User
2. Login as User
3. Exit
1
Enter User ID(in numbers): 1
Enter User Name(in words): Sarvesh
User registered successfully.
----- Library Management System -----
1. Register as User
2. Login as User
3. Exit
1
Enter User ID(in numbers): 2
Enter User Name(in words): Aswin
User registered successfully.
----- Library Management System -----
1. Register as User
2. Login as User
3. Exit
1
Enter User ID(in numbers): 3
Enter User Name(in words): Kishor
User registered successfully.
----- Library Management System -----
1. Register as User
2. Login as User
3. Exit
1
```

Figure 3 User Registration

```
----- Library Management System -----
1. Register as User
2. Login as User
3. Exit
2
Enter User ID(in numbers): 2
Welcome, Aswin!
1. Request List of Books
2. Exit
1
Books:
Book ID: 1| BookTitle: Ujjaini| Author: Yeshwanth
1. Request List of Books
2. Exit
```

Figure 4 Adding Books

```
Book or user not found.
----- Library Management System -----
1. Add Book
2. Delete Book
3. Issue Book
4. Renew Book
5. Print User Database
6. Print Issued Books
7. Exit
Enter your choice: 3
Enter Book ID(in numbers): 1
Enter User ID(in numbers): 2
Book 'ujjaini' issued to user 'Aswin'.
----- Library Management System -----
```

Figure 5 Issuing Books

```
----- Library Management System -----
1. Add Book
2. Delete Book
3. Issue Book
4. Renew Book
5. Print User Database
6. Print Issued Books
7. Exit
Enter your choice: ko
● Enter appropriate values only ● Please try again

Process finished with exit code 0
```

Figure 6 Exception Handled

## 4. Analysis and Discussion:

### 4.1 Time complexity of our program:

Function Name	The Function's Time Complexity
1)registerUser	The insertNode operation is used to add a user to the binary search tree (BST). The time complexity of the insertNode operation in a BST is $O(\log n)$ in the average case and $O(n)$ in the worst case, where $n$ is the number of nodes in the tree.
2) loginUser	The findUserNode operation is used to search for a user in the BST. The time complexity of the findUserNode operation in a BST is $O(\log n)$ in the average case and $O(n)$ in the worst case, where $n$ is the number of nodes in the tree.
3) addBook	Similar to registerUser, the insertNode operation is used to add a book to the binary search tree. The time complexity is $O(\log n)$ in the average case and $O(n)$ in the worst case, where $n$ is the number of nodes in the tree.
4) deleteBook	The deleteNode operation is used to remove a book from the binary search tree. The time complexity of the deleteNode operation in a BST is $O(\log n)$ in the average case and $O(n)$ in the worst case, where $n$ is the number of nodes in the tree.
5) issueBook and renewBook	These operations involve finding the book and user nodes in their respective BSTs. The time complexity is $O(\log n)$ in the average case and $O(n)$ in the worst case.
6) printUsers and printBooks	These operations involve traversing the respective BSTs and printing the nodes. The time complexity is $O(n)$ , where $n$ is the number of nodes in the tree.

### 4.2 The limitations of our program:

Our code has its own set of limitations too i.e our code does not include any security measures like authentication and authorization unlike high level Library Management System which are being in use currently, also our program lacks Data persistence like in all the user data stored including the user and book information will be lost once the program terminates to overcome it we'd have to have a separate database which stores all the data entered by the user.

### **4.3 Converting our project into a better version:**

So, for converting our project into a better version of itself we'd firstly like to integrate a DBMS like SQL for saving the data inputted by the user so that it doesn't get lost once after the program is terminated. Apart from this we'd also like to add a GUI part into the project which would make it even more user interactive unlike what we have right now which is just limited to the command line interface.

## 5. Appendix:

### Our code:

```
import java.util.Scanner;

class Book {
    int id;
    String title;
    String author;

    public Book(int id, String title, String author) {
        this.id = id;
        this.title = title;
        this.author = author;
    }

    @Override
    public String toString() {
        return "Book ID: " + id + "| BookTitle: " + title + "| Author: " +
author;
    }
}

class User {
    int id;
    String name;

    public User(int id, String name) {
        this.id = id;
        this.name = name;
    }

    @Override
    public String toString() {
        return "User ID: " + id + "| Name: " + name;
    }
}

class TreeNode<T> {
    T data;
    TreeNode<T> left;
    TreeNode<T> right;

    public TreeNode(T data) {
        this.data = data;
        left = null;
        right = null;
    }
}
```

```

}

class LibraryManagementSystem {
    private TreeNode<Book> bookRoot;
    private TreeNode<User> userRoot;

    public LibraryManagementSystem() {
        bookRoot = null;
        userRoot = null;
    }

    public void registerUser(int id, String name) {
        User newUser = new User(id, name);
        userRoot = insertNode(userRoot, newUser);
        System.out.println("User registered successfully.");
    }

    public boolean loginUser(int id) {
        TreeNode<User> userNode = findUserNode(userRoot, id);
        if (userNode == null) {
            System.out.println("User not found.");
            return false;
        }
        else {
            System.out.println("Welcome, " + userNode.data.name + "!");
            return true;
        }
    }

    public void addBook(int id, String title, String author) {
        Book newBook = new Book(id, title, author);
        bookRoot = insertNode(bookRoot, newBook);
        System.out.println("Book added successfully.");
    }

    public void deleteBook(int id) {
        bookRoot = deleteNode(bookRoot, id);
        System.out.println("Book deleted successfully.");
    }

    public void issueBook(int bookId, int userId) {
        TreeNode<Book> bookNode = findBookNode(bookRoot, bookId);
        TreeNode<User> userNode = findUserNode(userRoot, userId);

        if (bookNode == null || userNode == null) {
            System.out.println("Book or user not found.");
            return;
        }
    }
}

```

```

    }

    System.out.println("Book '" + bookNode.data.title + "' issued to user '" + userNode.data.name + "'.");
}

public void renewBook(int bookId, int userId) {
    TreeNode<Book> bookNode = findBookNode(bookRoot, bookId);
    TreeNode<User> userNode = findUserNode(userRoot, userId);

    if (bookNode == null || userNode == null) {
        System.out.println("Book or user not found.");
        return;
    }

    System.out.println("Book '" + bookNode.data.title + "' renewed for user '" + userNode.data.name + "'.");
}

public void printUsers() {
    System.out.println("User Database:");
    printInorder(userRoot);
}

public void printBooks() {
    if(bookRoot == null){
        System.out.println("No Books are in them library. Request admin to add the books...");
    }
    else {
        System.out.println("Books:");
        printInorder(bookRoot);
    }
}

private TreeNode<Book> insertNode(TreeNode<Book> root, Book book) {
    if (root == null) {
        return new TreeNode<>(book);
    }

    if (book.id < root.data.id) {
        root.left = insertNode(root.left, book);
    } else if (book.id > root.data.id) {
        root.right = insertNode(root.right, book);
    }

    return root;
}

```



```

}

private TreeNode<User> insertNode(TreeNode<User> root, User user) {
    if (root == null) {
        return new TreeNode<>(user);
    }

    if (user.id < root.data.id) {
        root.left = insertNode(root.left, user);
    } else if (user.id > root.data.id) {
        root.right = insertNode(root.right, user);
    }

    return root;
}

private TreeNode<Book> deleteNode(TreeNode<Book> root, int id) {
    if (root == null) {
        return null;
    }

    if (id < root.data.id) {
        root.left = deleteNode(root.left, id);
    } else if (id > root.data.id) {
        root.right = deleteNode(root.right, id);
    } else {
        if (root.left == null) {
            return root.right;
        } else if (root.right == null) {
            return root.left;
        }

        TreeNode<Book> successor = findMin(root.right);
        root.data = successor.data;
        root.right = deleteNode(root.right, successor.data.id);
    }

    return root;
}

private TreeNode<Book> findMin(TreeNode<Book> node) {
    if (node.left != null) {
        return findMin(node.left);
    }

    return node;
}

```

```

private TreeNode<User> findUserNode(TreeNode<User> root, int id) {
    if (root == null || ((User)root.data).id == id) {
        return root;
    }

    if (id < ((User)root.data).id) {
        return findUserNode(root.left, id);
    } else {
        return findUserNode(root.right, id);
    }
}

private TreeNode<Book> findBookNode(TreeNode<Book> root, int id) {
    if (root == null || ((Book)root.data).id == id) {
        return root;
    }

    if (id < ((Book)root.data).id) {
        return findBookNode(root.left, id);
    } else {
        return findBookNode(root.right, id);
    }
}

private <T> void printInorder(TreeNode<T> root) {
    if (root != null) {
        printInorder(root.left);
        System.out.println(root.data);
        printInorder(root.right);
    }
}

}

public class Main {
    public static void main(String[] args) {

        try {
            LibraryManagementSystem library = new LibraryManagementSystem();
            Scanner scanner = new Scanner(System.in);
            int choice;
            int adminoruser;

            do {

                System.out.println("----- Library Management System -----");
                System.out.println("1. Admin");
                System.out.println("2. User");
            } while (true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

        System.out.println("Enter your choice");
        adminoruser = scanner.nextInt();

        switch (adminoruser) {
            case 1:
                do {
                    System.out.println("----- Library Management
System -----");

                    System.out.println("1. Add Book");
                    System.out.println("2. Delete Book");
                    System.out.println("3. Issue Book");
                    System.out.println("4. Renew Book");
                    System.out.println("5. Print User Database");
                    System.out.println("6. Print Issued Books");
                    System.out.println("7. Exit");
                    System.out.print("Enter your choice: ");
                    choice = scanner.nextInt();

                    switch (choice) {

                        case 1:
                            System.out.println("⚠️Make sure you enter
the book id only in integers and book title and author names in string, do not
enter any integers ⚠️ \n ⚠️Do not enter the previously entered data while
this program is run.⚠️");

                            System.out.print("Enter Book ID(in
numbers): ");

                            int bookId = scanner.nextInt();
                            scanner.nextLine();
                            System.out.print("Enter Book Title(in
words): ");

                            String bookTitle = scanner.nextLine();
                            System.out.print("Enter Book Author(in
words): ");

                            String bookAuthor = scanner.nextLine();
                            library.addBook(bookId, bookTitle,
bookAuthor);

                            break;
                        case 2:
                            System.out.print("Enter Book ID(in
numbers): ");

                            int deleteId = scanner.nextInt();
                            library.deleteBook(deleteId);
                            break;
                        case 3:
                            System.out.print("Enter Book ID(in
numbers): ");

                            int issueBookId = scanner.nextInt();

```

```

                                System.out.print("Enter User ID(in
numbers): ");
                                int issueUserId = scanner.nextInt();
                                library.issueBook(issueBookId,
issueUserId);
                                break;
                                case 4:
                                System.out.print("Enter Book ID(in
numbers): ");
                                int renewBookId = scanner.nextInt();
                                System.out.print("Enter User ID(in
numbers): ");
                                int renewUserId = scanner.nextInt();
                                library.renewBook(renewBookId,
renewUserId);
                                break;
                                case 5:
                                library.printUsers();
                                break;
                                case 6:
                                library.printBooks();
                                break;
                                case 7:
                                System.out.println("Exiting Library
Management System. Goodbye!");
                                break;
                                default:
                                System.out.println("Invalid choice. Please
try again.");
                                }
                                } while (choice != 7);
                                break;

                                case 2:
                                do {
                                System.out.println("----- Library Management
System -----");

                                System.out.println("1. Register as User");
                                System.out.println("2. Login as User");
                                System.out.println("3. Exit");
                                choice = scanner.nextInt();
                                switch (choice) {
                                case 1:
                                System.out.print("Enter User ID(in
numbers): ");

                                int userId = scanner.nextInt();
                                scanner.nextLine();

```

```

        System.out.print("Enter User Name(in
words): ");

        String userName = scanner.nextLine();
        library.registerUser(userId, userName);
        break;

    case 2:
        System.out.print("Enter User ID(in
numbers): ");

        int loginId = scanner.nextInt();
        boolean x = library.loginUser(loginId);
        if (x == true) {

            do {
                System.out.println("1. Request
List of Books");

                System.out.println("2. Exit");
                choice = scanner.nextInt();

                switch (choice) {
                    case 1:
                        library.printBooks();
                        break;
                    case 2:
                        System.out.println("Exitin
g Library Management System. Goodbye!");
                        break;
                    default:
                        System.out.println("Invali
d choice. Please try again.");
                }
            } while (choice != 2);

        } else {
            System.out.println("Unable to login
please try again");
        }
        break;
    case 3:
        System.out.println("Exiting Library
Management System. Goodbye!");
        break;
    default:
        System.out.println("Invalid choice. Please
try again.");
}

```

```

        } while (choice != 3);
        break;

        default:
            System.out.println("Invalid choice");

    }

    } while (adminoruser != 3);

}

catch (Exception e) {
    System.out.println("⦿Enter appropriate values only⦿ Please try
again");
} catch (Error e) {
    System.out.println("Something went wrong due to the input, please try
again");
}
}
}
}

```