ASSIGNMENT 5 - LIBRARY MANAGEMENT SYSTEM

CONTENTS

i. **Description**:

This is a simple **Library Management System** with the following functionalities:

1. Add a book
2. View Library
3. Issue/Borrow a book
4. View issued books
5. Remove a book

I have used 2 ADTs both of type RiyaItemType to implement these application functionalities:

1. Binary Tree: Add, View and Removes books from library
2. Queue: Add and View the issued/borrowed books

ii. **Purpose of application:**

- In this age of technology, almost all of us rely on the electronically saved information. It allows easy access and management of real-world objects in the form of data.
- Primary objective of this library system is to collect, store, view, borrow and retrieve/delete the book information to the user.
- The system helps the user to keep a constant track of all the books available in the library electronically.

iii. **OOA design specifics:**

- BhutadaMyLibraryApp class contains the main() method. It displays a MENU to select from 4 options - ADD, VIEW, DELETE or QUIT APP.
- ADD option asks for Title, Author and Published Year of the book and adds to the binary tree in in-order.
- VIEW will display all the books using In-Order Traversal

- ISSUE/BORROW will first check if the book is present in the library in in-order fashion. If the book is found, details such as title and name of the person borrowing the book are stored in the Queue ADT.
- VIEW ISSUED BOOKS will list all the books (stored in the Queue) that were borrowed by the respective people.
- DELETE option deletes the specified book from the tree.
- QUIT will exit the App.
- There is a dependency relationship between Application class and ADT class. There is dependency relationship between both the ADT classes and ItemType class as well. The Tree ADT class implements the Interface. Both ADT classes and Application class uses Custom_Exception class.

iv. Design pattern implemented:

I have implemented the Model-View-Controller (MVC) design pattern. The application logic is separated into three distinct components:

a. The model (or the RiyaItemType class), which represents the data.
b. The view (or the main BhutadaMyLibraryApp class), which represents the user interface and takes user input.
c. The controller (or the CS420ListADT and CS420ListADT2 classes), which handle user interactions and updates the Model and View accordingly.

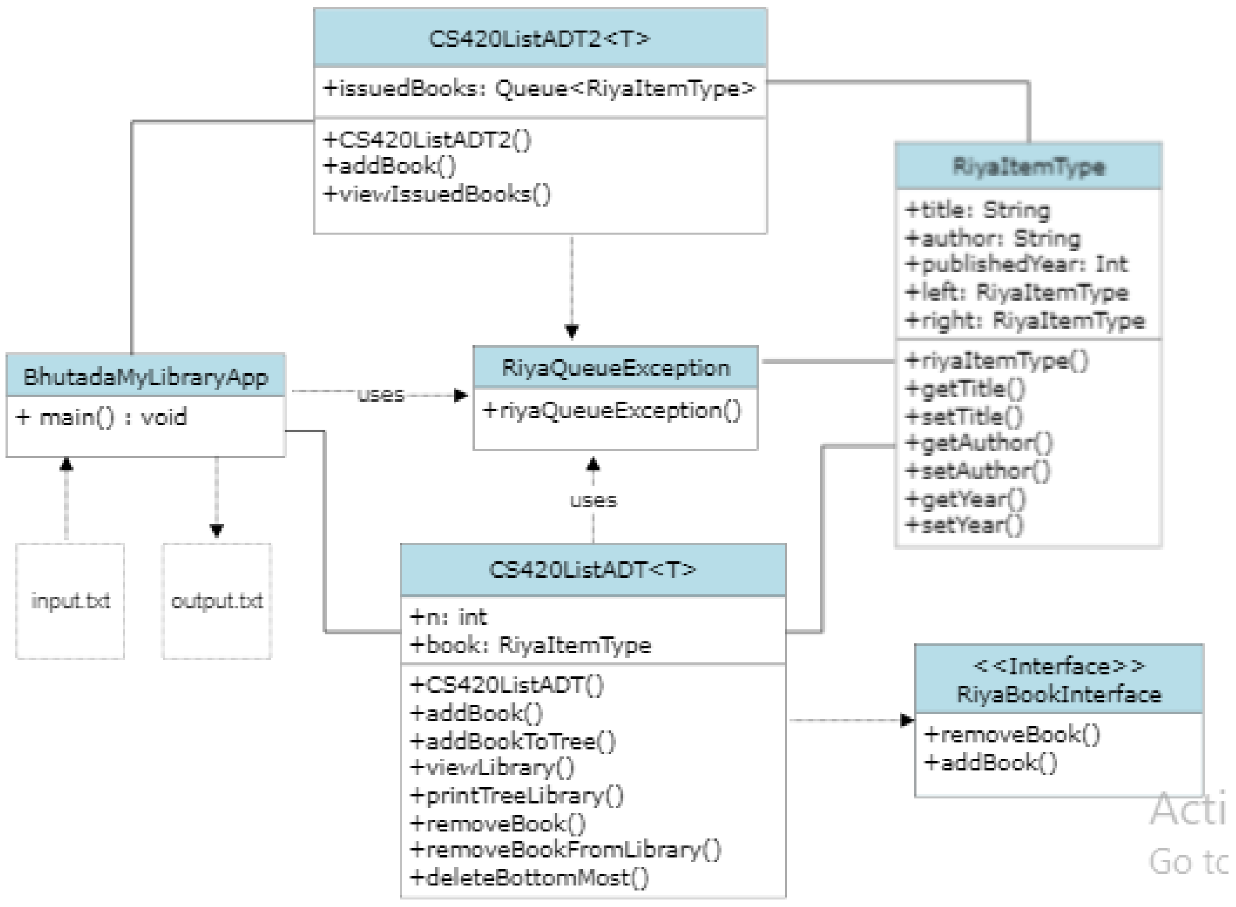
v. Files included:

1. BhutadaMyLibraryApp.java
2. BhutadaMyLibraryApp.class
3. RiyaItemType.java
4. RiyaItemType.class
5. CS420ListADT.java
6. CS420ListADT.class
7. CS420ListADT2.java
8. CS420ListADT2.class
9. RiyaBookInterface.java
10. RiyaBookInterface.class
11. RiyaQueueException.java
12. RiyaQueueException.class
13. input.txt
14. output.txt
15. README.txt
16. RiyaApplicationDesignDocument.doc
17. RiyaUMLDiagram.png
18. RiyaUseCaseDiagram.png

To run: java BhutadaMyLibrary.java

- Input to the application is specified in input.txt
- Output is saved in output.txt file.

vi. UML Diagram:



vii. Use Cases:

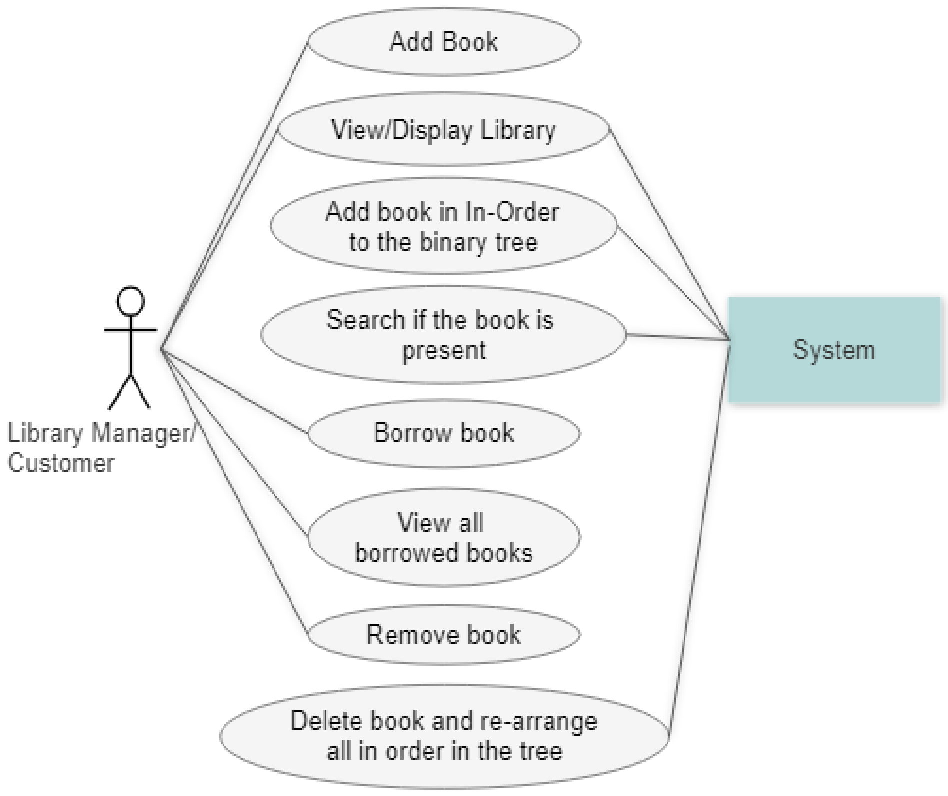
Use Case 1:	Add and View Library
Actor:	Library manager
Description:	Librarian adds a new book and views the complete list of books present in the library. Books are displayed in alphabetical order. If the library is empty, system returns null.

Use Case 2:	Borrow book
Actor:	The customer and library manager
Description:	The library manager enters the name of the book that customer wants to borrow. The system searches for the book. If present, the book title and customer name is added to the ISSUED/BORROWED list. All the borrowed books can be displayed.

Use Case 3:	Remove/Delete book from the library
Actor:	Library manager

Description: The manager enters the book title to be deleted. The system then searches for the book and if present, deletes all the information from the library.

viii. USE CASE DIAGRAM


Add Book
View/Display Library
Add book in In-Order to the binary tree
Search if the book is present
Borrow book
View all borrowed books
Remove book
Delete book and re-arrange all in order in the tree
Library Manager/ Customer
System