LIBRARY MANAGEMENT SYSTEM

Object Oriented Programming with Java (CSEN1111) Case Study Report

Semester-IV

GITAM SCHOOL OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

Submitted by

Student Name - POTNURU GOWTHAM **Roll No: 2023001620**

Student Name - VULLI KOUSHIK **Roll No: 2023001661**

Student Name - YELLAPU BHUPENDRA **Roll No: 2023008358**

Under the esteemed guidance of

Nagul Shaik, Assistant Professor.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GITAM SCHOOL OF TECHNOLOGY

GITAM (Deemed to be University)

DECLARATION

We, hereby declare that the **Object Oriented Programming with Java** Case study report entitled

"**LIBRARY MANAGEMENT SYSTEM**" is an original work done in

Semester IV, Department of Computer Science and Engineering,

GITAM School of Technology, GITAM (Deemed to be University).

Date: 27-03-2025

Registration No(s).	Name(s)	Signatures(s)
2023001620	**POTNURU GOWTHAM**	
2023001661	**VULLI KOUSHIK**	
2023008358	**YELLAPU BHUPENDRA**	

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GITAM SCHOOL OF TECHNOLOGY

GITAM (Deemed to be University)

CERTIFICATE

This is to certify that the report entitled "**LIBRARY MANAGEMENT SYSTEM**" is a bonafide record of work carried out by our team in Semester IV, Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University).

Faculty In-Charge

Nagul Shaik, Assistant Professor.
Dept. of CSE, GST,
GITAM (Deemed to be University)
Visakhapatnam

Head of the Department

Dr. G Lakshmeeswari (HoD, CSE),
Associate Professor
Dept. of CSE, GST,
GITAM (Deemed to be University)
Visakhapatnam

INTRODUCTION

A Library Management System (LMS) is a software application designed to manage and automate various library operations, such as storing book details, tracking borrowed books, managing library members, and ensuring smooth book transactions.

This **Java-based Library Management System** efficiently handles the core functionalities of a library, including:

1. **Adding Books** – Librarians can add books with titles and authors.
2. **Viewing Available Books** – Users can browse the list of books along with their availability status.
3. **Borrowing Books** – Library members can borrow books if they are available.
4. **Returning Books** – Members can return borrowed books to make them available for others.
5. **Searching Books** – Users can search for books by title.

Abstract

The Library Management System (LMS) is a software solution designed to streamline and automate the management of books and library members. This Java-based system provides an efficient way to handle book transactions, ensuring smooth borrowing and returning processes. It aims to enhance the user experience by offering a simple interface for viewing available books, searching for specific titles, and updating book availability status dynamically.

The system is developed using Object-Oriented Programming (OOP) principles and consists of multiple classes, including Book, Library, and LibraryMember, each performing essential roles in book management. The Library class maintains a collection of books, allowing users to search, add, and manage book information. The LibraryMember class enables members to borrow and return books, ensuring that only available books can be issued.

Through a menu-driven console interface, users can interact with the system by choosing from various options such as viewing books, borrowing, and returning. The system enforces book availability rules and updates records in real time. Although currently implemented as a simple console-based application, this project lays the foundation for future enhancements, including database integration, user authentication, and fine management.

This Library Management System is a fundamental yet effective tool for automating library operations, reducing manual workload, and improving efficiency in book tracking. It is particularly useful for small to medium-sized libraries and can be expanded to accommodate larger systems with advanced functionalities.

Overview of the System

The system is implemented using **Object-Oriented Programming (OOP) principles** in Java, with the following key classes:

- **Book Class:** Represents a book with attributes such as title, author, and availability status. It includes methods to borrow, return, and retrieve book details.
- **Library Class:** Manages the collection of books and provides methods for adding books, displaying available books, and searching for a book by title.
- **LibraryMember Class:** Represents a library member who can borrow and return books.
- **Main Class:** Contains the user interface, allowing users to interact with the system through a menu-driven approach.

Working of the System

1. The program initializes with a few books added to the library.
2. The user (library member) enters their name to start using the system.
3. A menu-driven approach allows the user to:
    - View available books.
    - Borrow a book if it is available.
    - Return a borrowed book.
    - Exit the system.
4. The system ensures that only available books can be borrowed and updates the status accordingly.
5. When a book is returned, it becomes available again for other users.

IMPLEMENTED JAVA PROGRAM:

LibraryManagementSystem.java — Share — Run

```java
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class LibraryManagementSystem {

    static class Book {
        private String title;
        private String author;
        private boolean isAvailable;

        public Book(String title, String author) {
            this.title = title;
            this.author = author;
            this.isAvailable = true;
        }

        public void borrowBook() {
            if (isAvailable) {
                isAvailable = false;
                System.out.println("You have successfully borrowed the book:
                    " + title);
            } else {
                System.out.println("Sorry, the book '" + title + "' is not
                    available.");
            }
        }
```

LibraryManagementSystem.java — Share — Run

```java
        public void returnBook() {
            isAvailable = true;
            System.out.println("You have successfully returned the book: " +
                title);
        }

        public boolean isAvailable() {
            return isAvailable;
        }

        public String getBookDetails() {
            return "Title: " + title + ", Author: " + author + ", Available:
                " + (isAvailable ? "Yes" : "No");
        }

        public String getTitle() {
            return title;
        }
    }


    static class Library {
        private List<Book> books;

        public Library() {
            books = new ArrayList<>();
        }
```

LibraryManagementSystem.java

Share Run

```
53
54
55          public void addBook(Book book) {
56              books.add(book);
57          }
58
59
60          public void displayBooks() {
61              System.out.println("\nAvailable Books in the Library:");
62              for (Book book : books) {
63                  System.out.println(book.getBookDetails());
64              }
65          }
66
67
68          public Book searchBookByTitle(String title) {
69              for (Book book : books) {
70                  if (book.getTitle().equalsIgnoreCase(title)) {
71                      return book;
72                  }
73              }
74              return null;
75          }
76      }
77
78      static class LibraryMember {
79          private String name;
80
```

LibraryManagementSystem.java

Share Run

```
81          public LibraryMember(String name) {
82              this.name = name;
83          }
84
85
86          public void borrowBook(Library library, String bookTitle) {
87              Book book = library.searchBookByTitle(bookTitle);
88              if (book != null) {
89                  book.borrowBook();
90              } else {
91                  System.out.println("Book '" + bookTitle + "' not found in the
                        library.");
92              }
93          }
94
95
96          public void returnBook(Library library, String bookTitle) {
97              Book book = library.searchBookByTitle(bookTitle);
98              if (book != null) {
99                  book.returnBook();
100             } else {
101                 System.out.println("Book '" + bookTitle + "' not found in the
                        library.");
102             }
103         }
104     }
105
106
```

LibraryManagementSystem.java

Share Run

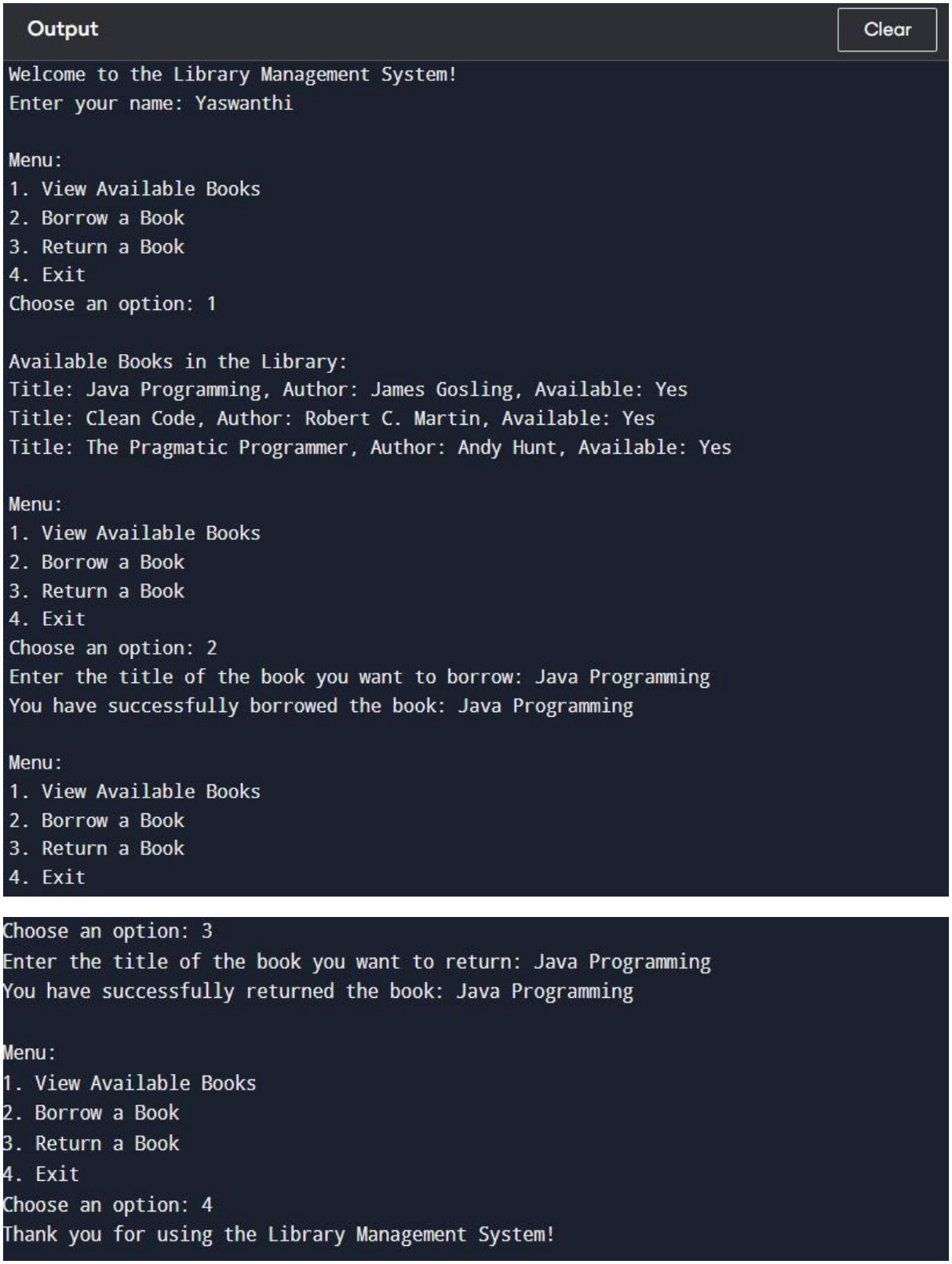
```
107     public static void main(String[] args) {
108         Scanner scanner = new Scanner(System.in);
109         Library library = new Library();
110 
111 
112         library.addBook(new Book("Java Programming", "James Gosling"));
113         library.addBook(new Book("Clean Code", "Robert C. Martin"));
114         library.addBook(new Book("The Pragmatic Programmer", "Andy Hunt"));
115 
116         System.out.println("Welcome to the Library Management System!");
117 
118 
119         System.out.print("Enter your name: ");
120         String memberName = scanner.nextLine();
121 
122         LibraryMember member = new LibraryMember(memberName);
123 
124         boolean exit = false;
125 
126 
127         while (!exit) {
128             System.out.println("\nMenu:");
129             System.out.println("1. View Available Books");
130             System.out.println("2. Borrow a Book");
131             System.out.println("3. Return a Book");
132             System.out.println("4. Exit");
133             System.out.print("Choose an option: ");
134             int choice = scanner.nextInt();
135             scanner.nextLine();
```

LibraryManagementSystem.java

Share Run

```
136 
137             switch (choice) {
138                 case 1:
139                     library.displayBooks();
140                     break;
141 
142                 case 2:
143                     System.out.print("Enter the title of the book you want to
                        borrow: ");
144                     String borrowTitle = scanner.nextLine();
145                     member.borrowBook(library, borrowTitle);
146                     break;
147 
148                 case 3:
149                     System.out.print("Enter the title of the book you want to
                        return: ");
150                     String returnTitle = scanner.nextLine();
151                     member.returnBook(library, returnTitle);
152                     break;
153 
154                 case 4:
155                     exit = true;
156                     System.out.println("Thank you for using the Library
                        Management System!");
157                     break;
158 
159                 default:
160                     System.out.println("Invalid option. Please try again.");
161             }
162         }
163 
164         scanner.close();
165     }
166 }
```

OUTPUT:



```
Output                                                    Clear
Welcome to the Library Management System!
Enter your name: Yaswanthi

Menu:
1. View Available Books
2. Borrow a Book
3. Return a Book
4. Exit
Choose an option: 1

Available Books in the Library:
Title: Java Programming, Author: James Gosling, Available: Yes
Title: Clean Code, Author: Robert C. Martin, Available: Yes
Title: The Pragmatic Programmer, Author: Andy Hunt, Available: Yes

Menu:
1. View Available Books
2. Borrow a Book
3. Return a Book
4. Exit
Choose an option: 2
Enter the title of the book you want to borrow: Java Programming
You have successfully borrowed the book: Java Programming

Menu:
1. View Available Books
2. Borrow a Book
3. Return a Book
4. Exit
```

```
Choose an option: 3
Enter the title of the book you want to return: Java Programming
You have successfully returned the book: Java Programming

Menu:
1. View Available Books
2. Borrow a Book
3. Return a Book
4. Exit
Choose an option: 4
Thank you for using the Library Management System!
```

CONCLUSION:

The Library Management System is an essential tool for automating and streamlining library operations. By efficiently managing book records, borrowing, and returning processes, it reduces manual workload and enhances user experience. This Java-based system demonstrates the application of Object-Oriented Programming (OOP) principles to create a structured and scalable solution for library management.

Through its menu-driven interface, users can easily interact with the system, search for books, and update availability status dynamically. While this implementation serves as a basic console-based model, it provides a strong foundation for further enhancements such as database integration, user authentication, and online book reservations.

In conclusion, this system significantly improves library efficiency, ensuring that books are well-managed and easily accessible. With future advancements, it can evolve into a fully digital library solution, catering to the growing needs of modern educational institutions and public libraries.