

LIBRARY MANAGEMENT SYSTEM

A MINI-PROJECT REPORT

Submitted by

MAHESH

241801149

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE



RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project “**LIBRARY MANAGEMENT SYSTEM**” is the bonafide work of “**MAHESH K**” who carried out the project work under my supervision.

SIGNATURE**Dr. E.K.SUBRAMANIAN****ASSOCIATE PROFESSOR**

Dept. of Artificial Intelligence and Data
Science,
Rajalakshmi Engineering College
Chennai

This mini project report is submitted for the viva voce examination to be held on

INTERNAL EXAMINER**EXTERNAL EXAMINER**

ABSTRACT

This project, the Library Management System, is a **command-line application** developed using **Java** with a **SQL** backend to automate and streamline the daily operations of a library. The primary objective is to transition from a manual, paper-based system to an efficient, computerized one, thereby minimizing data loss, reducing human error, and saving time for both library staff and patrons. The system, with its backend powered by a robust SQL database, offers a centralized platform for managing the library's book collection, member information, and all circulation activities through a text-based terminal interface.

Key functionalities of the system include a comprehensive book management module, which allows an administrator to add, update, and remove books from the digital catalog stored in the **SQL database**. It also features a robust user management system for registering new members and maintaining their records. The circulation module automates the process of issuing and returning books, tracking due dates, and calculating fines for overdue materials, with all transaction data managed within the SQL database. A powerful search feature enables both librarians and

users to quickly locate available books by title, author, or genre directly from the command line.

Built as a console-based application, the system ensures functionality and efficiency without the need for a graphical front end, making it lightweight and portable. It aims to enhance the overall efficiency and accessibility of library services through straightforward text commands. By digitizing and automating library operations with **Java** and **SQL** in a terminal environment, this project provides an effective, scalable, and manageable solution for libraries, contributing to an enriched learning and research environment.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the Chairperson **DR. M. THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head of the Department **Dr. J.M. GNANASEKAR** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide **Dr. E.K. SUBRAMANIAN**, for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

MAHESH K 241801149

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	iv
1	INTRODUCTION	1
1.1	INTRODUCTION	8
1.2	SCOPE OF THE WORK	8
1.3	PROBLEM STATEMENT	8
1.4	AIM AND OBJECTIVES OF THE PROJECT	8
2	SYSTEM SPECIFICATIONS	9
2.1	HARDWARE SPECIFICATIONS	9
2.2	SOFTWARE SPECIFICATIONS	9
3	MODULE DESCRIPTION	10
4	CODING	11
5	SCREENSHOTS	16
6	CONCLUSION AND FUTURE ENHANCEMENT	18
	REFERENCES	19

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
5.1	System Architecture Diagram	15
5.2	ER (Entity-Relationship) Diagram for the Database	15
5.3	Data Flow Diagram (DFD)	16
5.4	Database Schema (Tables and Relationships)	16
5.5	Main Menu Interface in the Terminal	17
5.6	User Login and Registration Process	17
5.7	Adding a New Book via Command Line	18
5.8	Searching for a Book by Title	18
5.9	Issuing a Book to a Member	19

FIGURE NO.	TITLE	PAGE NO.
5.10	Returning a Book and Fine Calculation	19
5.11	Viewing All Books in the Library	20
5.12	Deleting a Book from the Catalog	20

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Library Management System is a software application designed to manage the catalog and daily operations of a library. This project helps librarians and members by providing necessary information about the availability of books and the status of borrowed items. The system replaces the traditional, manual method of record-keeping with an efficient, automated process, ensuring that all information, from book details to member borrowing history, is accurate and easily accessible

1.2 SCOPE OF THE WORK

The scope of this project is to develop a command-line application using Java and SQL that automates core library functions. It will help librarians manage the book inventory, register and track library members, and handle the circulation process of issuing and returning books. The system is designed to provide quick and easy access to the library's database for both administrative tasks and user inquiries, all within a terminal environment

1.3 PROBLEM STATEMENT

The need for this project arises from the inefficiencies and limitations of manual library systems. Traditional methods are often slow, prone to human error, and make it difficult to track books and manage member records effectively. Searching for a specific book can be a time-consuming task, and maintaining paper records risks data loss or damage. This inconvenience affects the efficiency of the library staff and the satisfaction of the library's members.

1.4 AIM AND OBJECTIVES OF THE PROJECT

The main **aim** of this project is to design and develop a robust Library Management System to automate and streamline all library operations.

The **objectives** of the project are:

- To create a centralized SQL database to store and manage all book and member information securely.
- To build a system in Java that automates the process of issuing, returning, and renewing books.
- To implement a feature for automatically tracking due dates and calculating fines for overdue books.
- To provide a simple, command-line interface for librarians to perform administrative tasks and for users to search the library catalog efficiently.
- To improve the overall efficiency, accuracy, and user experience of the library's services by replacing manual paperwork

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor	:	Intel i3
Memory Size	:	8GB (Minimum)
HDD	:	512 GB (Minimum)

2.2 SOFTWARE SPECIFICATIONS

Operating System	:	WINDOWS 10
Front – End	:	Java
Back - End	:	MySql
Language	:	Java,SQL

CHAPTER 3

MODULE DESCRIPTION

The Library Management System is structured into several distinct modules, each responsible for a specific set of tasks. The application operates through a command-line interface, with a Java-based backend connected to a SQL database.

1. Administrator Module

This is the core module that provides full control over the library's operations and is accessible only to the librarian or administrator. All functions within this module involve direct interaction with the SQL database to manage records.

- **Book Management:** Allows the administrator to add new books to the catalog, update existing book details (like title or author), view the entire list of books, and remove books from the library's collection.
- **Member Management:** Enables the administrator to register new library members, view a list of all members, update their contact information, and remove members from the system.

2. Circulation Module

This module automates the process of lending books and tracking their status. It is a critical function for daily library operations.

- **Issue Book:** Facilitates the process of issuing a book to a registered member. The system records the book ID, member ID, and the date of issue in the database.
- **Return Book:** Handles the return of a book from a member. Upon return, the system updates the book's availability status and records the return date.

- **View Issued Books:** Allows the administrator to see a list of all books that are currently checked out, including who borrowed them and when they are due.
- **Fine Calculation:** The system can automatically calculate fines for overdue books based on the return date versus the due date.

3. User Module (Search and View)

This module provides essential functions for library members or users who need to find books without requiring administrative privileges.

- **Search for Books:** Provides a command-line option for users to search the library catalog. Searches can be performed based on criteria like book title, author, or genre.
- **Check Book Availability:** After finding a book, the user can check its current status to see if it is available for borrowing or currently issued to another member.

4. Database Connection Module

This is a backend module that is not directly visible to the user but is fundamental to the system's operation.

- **SQL Database Connectivity:** This component uses Java Database Connectivity (JDBC) to establish and manage the connection between the Java application and the SQL database. It handles all data transactions, ensuring that information is read from and written to the database reliably. It executes SQL queries for creating, reading, updating, and deleting records in the Books, Members, and IssuedBooks tables.

CHAPTER 4

SAMPLE CODING

```
package com.example.library;

import com.example.library.models.Book;

import com.example.library.models.Member;

import java.sql.SQLException;

import java.util.List;

import java.util.Scanner;

public class Main {

    private static void printMenu() {

        System.out.println();

        System.out.println("=== Library Management System ===");

        System.out.println("1) Add book");

        System.out.println("2) List books");

        System.out.println("3) Add member");

        System.out.println("4) List members");

        System.out.println("5) Borrow book");

        System.out.println("6) Return book");
```

```
        System.out.println("0) Exit");

        System.out.print("Choose: ");

    }

    public static void main(String[] args) {

        try {

            Database.initializeSchema();

        } catch (SQLException e) {

            System.err.println("Failed to initialize database: " + e.getMessage());

            return;

        }

        LibraryService service = new LibraryService();

        Scanner scanner = new Scanner(System.in);

        while (true) {

            printMenu();

            String choice = scanner.nextLine().trim();

            try {

                switch (choice) {

                    case "1": {
```

```
System.out.print("Title: ");

String title = scanner.nextLine();

System.out.print("Author: ");

String author = scanner.nextLine();

System.out.print("ISBN: ");

String isbn = scanner.nextLine();

int id = service.addBook(title, author, isbn);

System.out.println("Added book with id " + id);

break;

}

case "2": {

    List<Book> books = service.listBooks();

    if (books.isEmpty()) System.out.println("No books yet.");

    for (Book b : books) System.out.println(b);

    break;

}

case "3": {

    System.out.print("Name: ");
```



```
String name = scanner.nextLine();

System.out.print("Email: ");

String email = scanner.nextLine();

int id = service.addMember(name, email);

System.out.println("Added member with id " + id);

break;

}

case "4": {

    List<Member> members = service.listMembers();

    if (members.isEmpty()) System.out.println("No members yet.");

    for (Member m : members) System.out.println(m);

    break;

}

case "5": {

    System.out.print("Book id: ");

    int bookId = Integer.parseInt(scanner.nextLine());

    System.out.print("Member id: ");

    int memberId = Integer.parseInt(scanner.nextLine());
```

```
        boolean ok = service.borrowBook(bookId, memberId);

        System.out.println(ok ? "Borrowed." : "Not available or invalid.");

        break;

    }

    case "6": {

        System.out.print("Book id: ");

        int bookId = Integer.parseInt(scanner.nextLine());

        boolean ok = service.returnBook(bookId);

        System.out.println(ok ? "Returned." : "No active loan found.");

        break;

    }

    case "0":

        System.out.println("Bye!");

        return;

    default:

        System.out.println("Invalid option.");

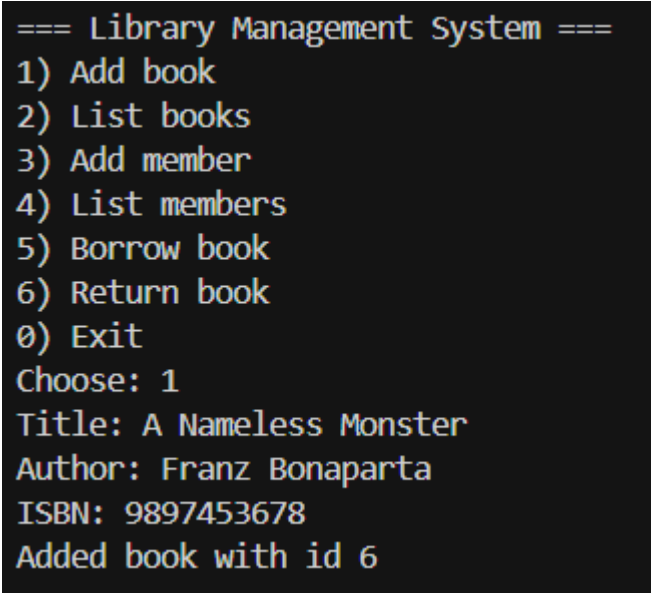
    }

} catch (SQLException ex) {
```

```
        System.err.println("Error: " + ex.getMessage());  
    }  
  
    }  
  
    }  
  
}
```

CHAPTER 5

SCREEN SHOTS



```
=== Library Management System ===  
1) Add book  
2) List books  
3) Add member  
4) List members  
5) Borrow book  
6) Return book  
0) Exit  
Choose: 1  
Title: A Nameless Monster  
Author: Franz Bonaparta  
ISBN: 9897453678  
Added book with id 6
```

Fig 5.1 Adding a Book

```
=== Library Management System ===
1) Add book
2) List books
3) Add member
4) List members
5) Borrow book
6) Return book
0) Exit
Choose: 2
[1] One Piece by Eichiro Oda | ISBN: 1569319014 | On loan
[2] Dragon Ball by Akira Toriyama | ISBN: 9781974708710 | Available
[3] Naruto by Masashi Kishimoto | ISBN: 9784088728407 | Available
[4] Reverend Insanity by Gu Zhen Ren | ISBN: B0D6LRRKYG | Available
[5] Berserk by Kentaro Miura | ISBN: 9782723448123 | Available
[6] A Nameless Monster by Franz Bonaparta | ISBN: 9897453678 | Available
```

Fig 5.2 Listing all Books

```
=== Library Management System ===
1) Add book
2) List books
3) Add member
4) List members
5) Borrow book
6) Return book
0) Exit
Choose: 3
Name: Adhitya
Email: adhitya@gmail.com
Added member with id 4
```

Fig 5.3 Adding a member

```
=== Library Management System ===
1) Add book
2) List books
3) Add member
4) List members
5) Borrow book
6) Return book
0) Exit
Choose: 4
[1] Mahesh | mahesh@gmail.com
[2] Manikandan | mani@gmail.com
[3] Manoj | manoj@gmail.com
[4] Adhitya | adhitya@gmail.com
```

Fig 5.4 List all members

```
=== Library Management System ===
1) Add book
2) List books
3) Add member
4) List members
5) Borrow book
6) Return book
0) Exit
Choose: 5
Book id: 3
Member id: 1
Borrowed.
```

Fig 5.5 Borrowing a book

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

This project successfully developed a **Library Management System** as a command-line application using Java and a SQL database. The system effectively transitions library operations from a manual, error-prone process to an efficient, automated one. By providing core functionalities such as book and member management, automated book issuance and returns, and search capabilities, the project has achieved its primary goal of streamlining library administration.

The application successfully centralizes data within a robust SQL database, which minimizes the risk of data loss and ensures that records are accurate and easy to access. The console-based interface, while simple, provides all the necessary tools for a librarian to manage the library's daily activities efficiently. The implementation of this system demonstrates a practical solution to the problems of traditional library management, leading to significant savings in time and effort for both staff and patrons.

While the current system fulfills its core objectives, there are several opportunities for future development that could further improve its functionality and user experience:

- **Graphical User Interface (GUI):** The most significant enhancement would be to develop a graphical front end using Java frameworks like Swing or JavaFX. This would make the system more intuitive and visually appealing for users who are not comfortable with command-line interfaces.
- **Web-Based Interface:** The system could be expanded into a full-stack web application, allowing users to search for books, check their account status, and even reserve books online from any device.
- **Barcode/QR Code Integration:** Integrating a barcode or QR code scanner would automate the book issuing and returning process even further. The

librarian could simply scan a book's barcode and a member's ID card to complete a transaction instantly.

- **Automated Email/SMS Notifications:** An automated notification system could be implemented to send reminders to members about upcoming due dates or to notify them when an overdue fine has been applied.
- **Advanced Search and Recommendation Engine:** The search functionality could be enhanced with more advanced filters (e.g., by publication year, genre). Additionally, a recommendation engine could be built to suggest books to users based on their borrowing history.
- **Role-Based Access Control:** A more sophisticated login system could be created to support different roles, such as 'Administrator', 'Librarian', and 'Member'. Each role would have different levels of access and permissions within the system.
- **Data Analytics and Reporting:** A module could be added to generate insightful reports, such as identifying the most popular books, tracking peak borrowing times, and analyzing user demographics. This data would help the library optimize its collection and services.

REFERENCES

1. <https://www.w3schools.com/java/>
2. <https://www.w3schools.com/sql/>
3. <https://www.geeksforgeeks.org/library-management-system-project-in-java/>
4. <https://www.javatpoint.com/java-jdbc>
5. <https://www.tutorialspoint.com/java/index.htm>