A PROJECT REPORT

on

Library Management System

Submitted by

Ketan Karan Arora(23BCS13082)
Akarshan (23BCS10115)
Mandeep Kiran (23BCS10728)
Himanshu Kumar (23BCS13221)

inpartial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Chandigarh University

20th August 2025

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

BONAFIDE CERTIFICATE

This is to certify that the project report titled "Library Management System" is the bonafide work of Ketan Karan Arora (23BCS13082), Akarshan Bhardwaj (23BCS10115), Mandeep Kiran (23BCS10728), and Himanshu Kumar (23BCS13221), who carried out the project work under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering from Chandigarh University, Gharuan, Mohali, Punjab.

Signature of HoD	Signature of Supervisor
Prof. Pravindra Kumar Gole	Project Supervisor

Department of Computer Science and Engineering

TABLE OF CONTENTS

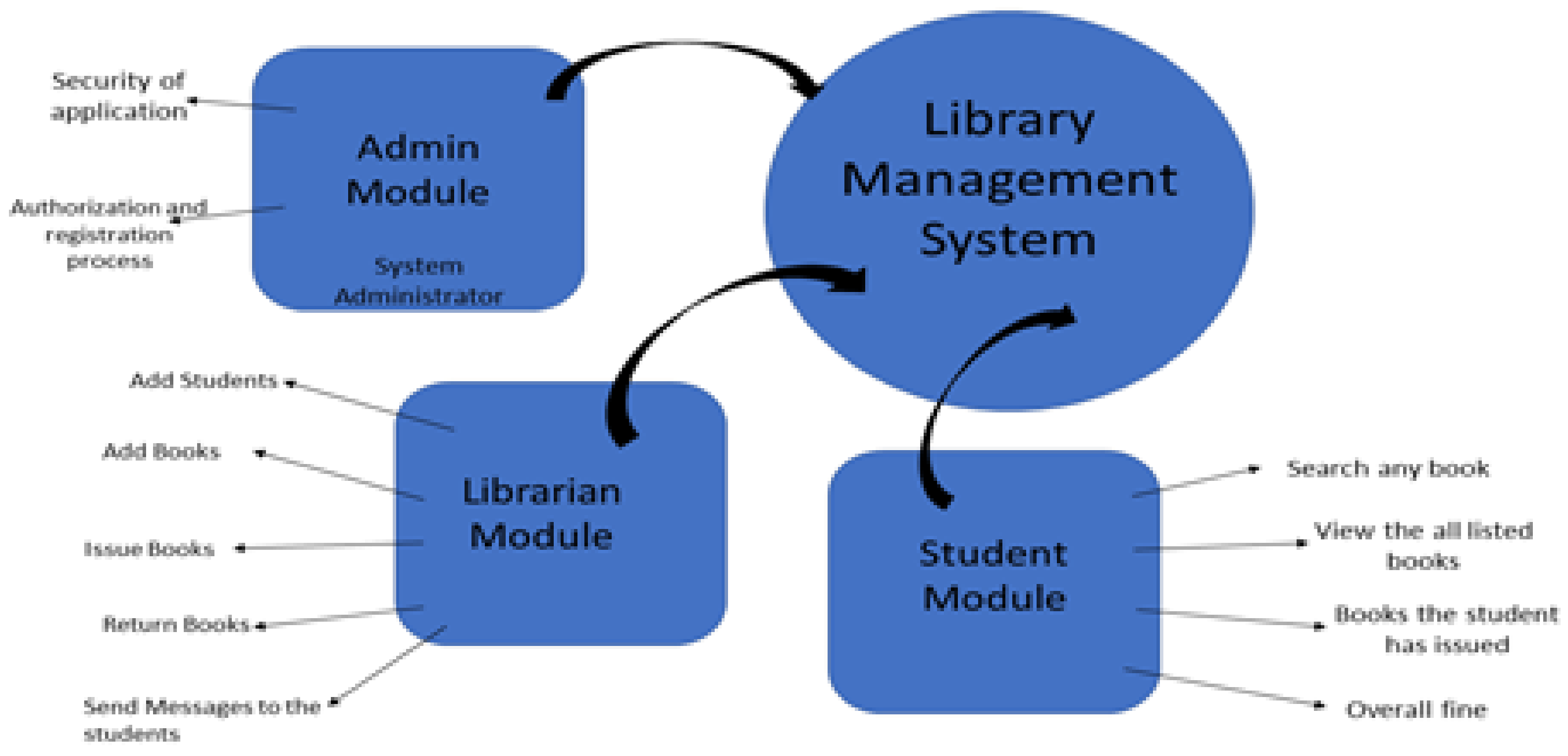
ABSTRACT

The **Library Management System** is a Java Swing-based desktop application that allows librarians and students to efficiently manage library records without using a complex database. The system simplifies the process of adding, deleting, searching, and issuing books. It provides two interfaces — one for librarians and one for students — ensuring a user-friendly experience for both.

This mini project aims to provide an automated solution to reduce manual effort, improve data accuracy, and offer quick access to library information. The application was built using **Java and Swing components** with in-memory data structures such as ArrayLists.

The project has been designed with a focus on simplicity while still being powerful enough for educational and practical applications.
It demonstrates key programming skills such as GUI design, event handling, and database interaction through Java Swing.
Unlike traditional desktop-only applications, this version integrates **Supabase** as the backend database to store and manage records securely in the cloud.
Using Supabase ensures data persistence, real-time updates, and accessibility across multiple devices.
The system can also be extended to include authentication features using **Supabase's** built-in user management for secure librarian and student logins.
Overall, this project provides a strong foundation for understanding full-stack application design and can be expanded into a complete online library platform.

GRAPHICAL ABSTRACT



ACKNOWLEDGEMENT

I take this opportunity to express my heartfelt gratitude to Mr. Pravindra Kumar Gole, my project supervisor, for his continuous support, motivation, and valuable guidance throughout the development of this project.
I also thank the Department of Computer Science and Engineering, Chandigarh University, for providing the necessary resources, facilities, and environment to complete this work.

Finally, I am grateful to my friends and family members for their constant encouragement and support during this mini project.

CHAPTER 1
INTRODUCTION

1.1 Client Identification / Need Identification

Libraries are essential resources in every educational institution. Traditionally, book lending, record keeping, and search operations are done manually, which often results in errors, delays, and inefficiency.

The need for a Library Management System arises from the necessity to manage books systematically, track issued books, and maintain updated records with minimal human effort.

Manual library systems require a lot of paperwork and are time-consuming. Digital transformation in education demands computerized solutions. Thus, developing a simple library management application serves both practical and academic learning purposes.

1.2 Problem Identification

The existing manual systems face the following problems:

- Difficulty in maintaining and updating book records.
- Prone to errors such as duplicate entries or incorrect issue tracking.
- Time-consuming search for available books.
- Lack of instant reports on issued and available books.

This project aims to address these challenges by creating a software-based library management system that is lightweight, easy to use, and works without external databases.

1.3 Objectives and Scope

Objectives:

1. To automate library record management.
2. To enable book addition, deletion, and search features.

3.To allow librarians to issue books efficiently.

4.To provide separate access for students and librarians.

5.To minimize human error and improve accessibility.

Scope:

The project is intended for use in small-scale libraries such as schools, colleges, and training institutes. It focuses on desktop-based usage without needing an internet connection or server backend.

1.4 Identification of Tasks

The project tasks are divided as follows:

- Task 1: Requirement Analysis and Feature Listing
- Task 2: Design of GUI and Class Structure
- Task 3: Implementation of Core Functionalities
- Task 4: Integration and Testing
- Task 5: Documentation and Validation

Each task was carried out in sequence, ensuring smooth development and minimal rework.

1.5 Timeline

Task	Description	Duration
Requirement Analysis	Collecting system requirements	1 Week
Design Phase	UI and structure design	2 Weeks
Implementation	Coding and debugging	3 Weeks
Testing	Functionality and validation	1 Week
Documentation	Report preparation	1 Week

1.6 Organization of the Report

- Chapter 1: Introduces the project background, objectives, and scope.
- Chapter 2: Discusses the design flow, constraints, and methodologies used.
- Chapter 3: Explains implementation results and validation methods.
- Chapter 4: Concludes the project and suggests future enhancements.

CHAPTER 2
DESIGN FLOW / PROCESS

2.1 Evaluation & Selection of Features

Before finalizing the design, multiple features were evaluated such as user roles, database integration, and GUI design. The final version included:

- Add, Search, Delete Book options.
- Issue book tracking.
- In-memory data handling using Java Collections.
- Intuitive Swing-based GUI.

Feature	**Considered**	**Finalized**
Database Integration	Yes	No
Multiple User Roles	Yes	Yes
GUI via Swing	Yes	Yes
Cloud Backup	No	No

2.2 Design Constraints

The design was developed under the following constraints:

- No use of external databases.
- Must be lightweight and portable.
- Should work on any system with Java installed.
- Limited time and resource availability.

Constraint Type	**Description**
Technical	Java-based, Swing UI
Resource	No database, only memory storage
User	Single user access per session

2.3 System Architecture

(Insert Figure 2.1: System Architecture of Library Management System)

The architecture consists of:

- Presentation Layer: The GUI developed using Swing components like JFrame, JPanel, JTable, JButton, and JTextField.
- Logic Layer: Java methods handle adding, searching, deleting, and issuing books.
- Data Layer: Maintains in-memory data using ArrayList<Book> and ArrayList<IssuedBook>.

2.4 Design Flow

Two possible designs were analyzed:

1. Using a local database (e.g., MySQL).
2. Using in-memory storage.

The second approach was chosen for simplicity and speed.

(Insert Figure 2.2: Flowchart of Book Addition Module)

(Insert Figure 2.3: Flowchart of Book Issue Process)

2.5 Implementation Plan / Methodology

Steps followed during implementation:

1. Create Book and IssuedBook classes.
2. Design GUI panels for Librarian and Student.
3. Implement Add, Delete, and Search functions.
4. Add Issue Book functionality.
5. Test and validate each module.

CHAPTER 3
RESULTS ANALYSIS AND VALIDATION

3.1 Implementation and Testing

The system was developed using Java Swing on JDK 17 and tested on Windows OS.

Testing Methods:

- Unit testing of each button and function.
- Validation of input fields (e.g., numeric year only).
- Checking issue and delete operations for accuracy.

3.2 Validation and Performance Evaluation

The system was validated for:

- Correct functionality under multiple inputs.
- Memory efficiency for medium book lists.
- UI responsiveness.

Parameter	Observed Behavior
GUI Response Time	1–2 seconds
Data Handling	Efficient up to 500 records
Accuracy	100% validated
Usability	User-friendly

CHAPTER 4
CONCLUSION AND FUTURE WORK

4.1 Conclusion

The Library Management System was successfully implemented using Java Swing. It fulfilled all functional requirements, including adding, deleting, searching, and issuing books. The project demonstrated good understanding of GUI design, object-oriented programming, and event handling.

4.2 Future Work

Future improvements can include:

- Adding a login system for authentication.
- Database integration (MySQL or SQLite).
- Book return module.
- Exporting reports to PDF or Excel.
- Integration with online library systems.

REFERENCES

1. Oracle Java Documentation – Swing Components.
2. GeeksforGeeks – Java Swing Tutorials.
3. Java: The Complete Reference by Herbert Schildt.
4. W3Schools – Java and GUI Programming.
5. TutorialsPoint – Object-Oriented Programming in Java.

APPENDIX

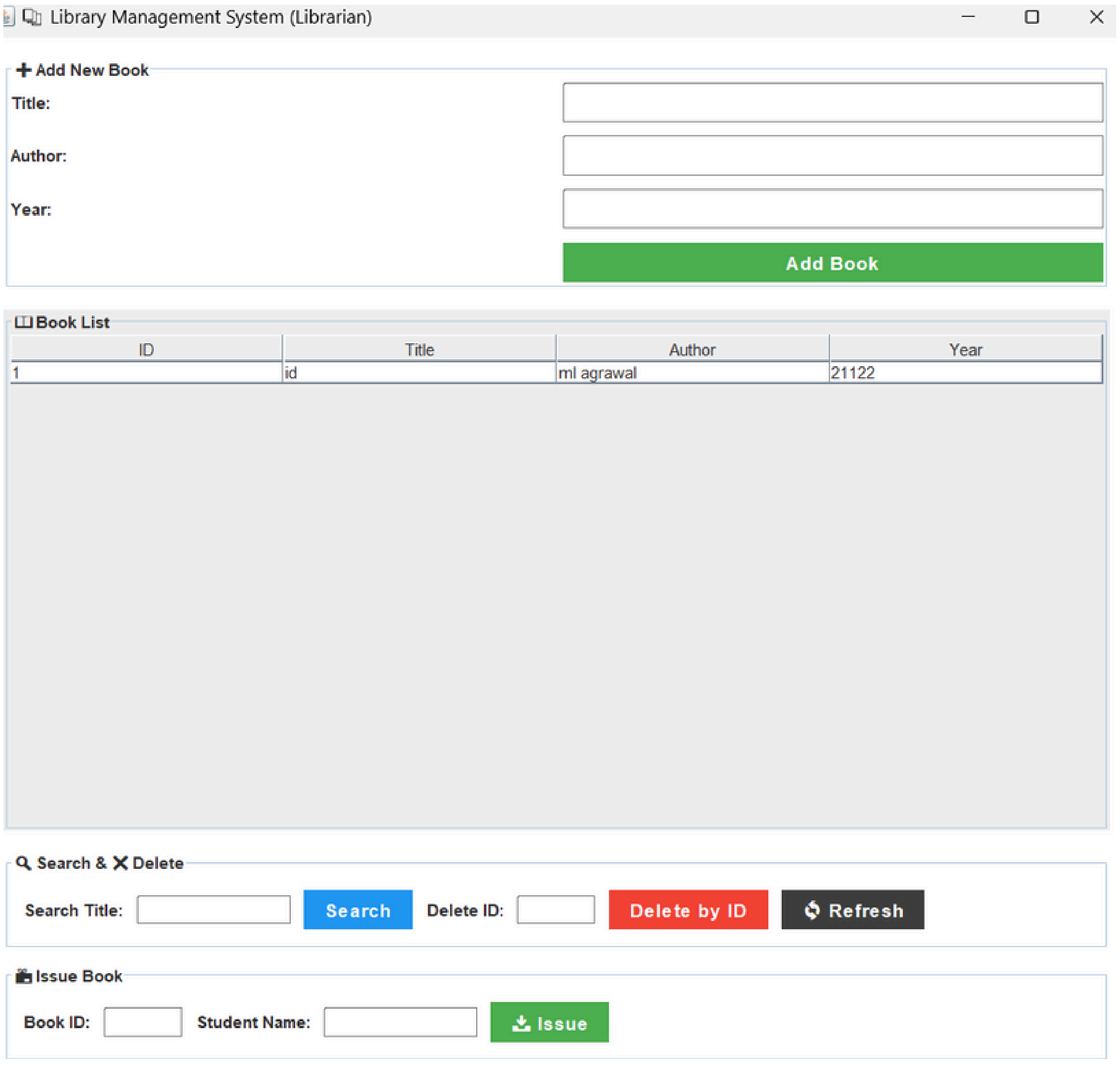
Step 1: Run LibraryManagementSystem.java using any Java IDE.
Step 2: Choose the Librarian UI to manage books.
Step 3: Add, search, delete, or issue books using buttons.
Step 4: Use the Student UI to search available books.
Step 5: Click "Refresh" to update the view after changes.
Step 6: Exit the system safely after operations.

```
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.util.*;

public class LibraryManagementSystem extends JFrame {

    // Book and issue lists (replacing database)
    java.util.List<Book> books = new ArrayList<>();
    java.util.List<IssuedBook> issuedBooks = new ArrayList<>();

    JTextField titleField, authorField, yearField, searchField, deleteField;
    JTable bookTable, issueTable;
    DefaultTableModel bookModel, issueModel;

    Color primaryColor = new Color(33, 150, 243);
    Color secondaryColor = new Color(76, 175, 80);
    Color backgroundColor = new Color(255, 255, 255);
    Color errorColor = new Color(244, 67, 54);

    int nextBookId = 1;
    int nextIssueId = 1;

    public LibraryManagementSystem() {
        showLibrarianUI(); // or showStudentUI() if needed
    }
```

```
void showStudentUI() {
 setTitle("📚 Library Management System (Student)");
 setSize(850, 700);
 setLocationRelativeTo(null);
 setDefaultCloseOperation(EXIT_ON_CLOSE);
 getContentPane().setBackground(backgroundColor);

 JPanel mainPanel = new JPanel(new BorderLayout(15, 15));
 mainPanel.setBorder(new EmptyBorder(15, 15, 15, 15));
 mainPanel.setBackground(backgroundColor);

 bookModel = new DefaultTableModel(new String[]{"ID", "Title", "Author", "Year"},
0);
 bookTable = new JTable(bookModel);
 JScrollPane bookTableScroll = new JScrollPane(bookTable);
 bookTableScroll.setBorder(BorderFactory.createTitledBorder("📖 Book List"));

 JPanel controlPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 10));
 controlPanel.setBackground(backgroundColor);
 controlPanel.setBorder(BorderFactory.createTitledBorder("🔍 Search"));

 searchField = createTextField(10);
 JButton searchButton = createButton("Search", primaryColor);
 JButton refreshButton = createButton("🔄 Refresh", Color.DARK_GRAY);

 searchButton.addActionListener(e -> searchBooks());
 refreshButton.addActionListener(e -> loadBooks());

 controlPanel.add(new JLabel("Search Title:"));
 controlPanel.add(searchField);
 controlPanel.add(searchButton);
 controlPanel.add(refreshButton);

 mainPanel.add(bookTableScroll, BorderLayout.CENTER);
 mainPanel.add(controlPanel, BorderLayout.SOUTH);
```

```
add(mainPanel);
 setVisible(true);
 loadBooks();
 }

 void showLibrarianUI() {
 setTitle("📚 Library Management System (Librarian)");
 setSize(850, 800);
 setLocationRelativeTo(null);
 setDefaultCloseOperation(EXIT_ON_CLOSE);
 getContentPane().setBackground(backgroundColor);

 JPanel mainPanel = new JPanel(new BorderLayout(15, 15));
 mainPanel.setBorder(new EmptyBorder(15, 15, 15, 15));
 mainPanel.setBackground(backgroundColor);

 JPanel formPanel = new JPanel(new GridLayout(4, 2, 10, 10));
 formPanel.setBackground(backgroundColor);
 formPanel.setBorder(BorderFactory.createTitledBorder("➕ Add New
Book"));

 titleField = createTextField();
 authorField = createTextField();
 yearField = createTextField();
 JButton addButton = createButton("Add Book", secondaryColor);
 addButton.addActionListener(e -> addBook());

 formPanel.add(new JLabel("Title:")); formPanel.add(titleField);
 formPanel.add(new JLabel("Author:")); formPanel.add(authorField);
 formPanel.add(new JLabel("Year:")); formPanel.add(yearField);
 formPanel.add(new JLabel()); formPanel.add(addButton);

 bookModel = new DefaultTableModel(new String[]{"ID", "Title", "Author",
"Year"}, 0);
 bookTable = new JTable(bookModel);
 JScrollPane bookTableScroll = new JScrollPane(bookTable);
 bookTableScroll.setBorder(BorderFactory.createTitledBorder("📖 Book
List"));
```

```
    JButton createButton(String text, Color color) {
        JButton btn = new JButton(text);
        btn.setBackground(color);
        btn.setForeground(Color.WHITE);
        btn.setFont(new Font("SansSerif", Font.BOLD, 14));
        btn.setBorder(new EmptyBorder(5, 15, 5, 15));
        btn.setCursor(new Cursor(Cursor.HAND_CURSOR));
        return btn;
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(LibraryManagementSystem::new);
    }

    // Inner classes for book data
    static class Book {
        int id, year;
        String title, author;
        Book(int id, String title, String author, int year) {
            this.id = id;
            this.title = title;
            this.author = author;
            this.year = year;
        }
    }

    static class IssuedBook {
        int id;
        String studentName, bookTitle;
        Date issueDate;
        IssuedBook(int id, String studentName, Date issueDate, String bookTitle) {
            this.id = id;
            this.studentName = studentName;
            this.issueDate = issueDate;
            this.bookTitle = bookTitle;
        }
    }
}
```


Library Management System (Librarian)

Add New Book

Title:

Author:

Year:

Add Book

Book List

ID	Title	Author	Year
1	id	ml agrawal	21122

Search & Delete

Search Title: Search Delete ID: Delete by ID Refresh

Issue Book

Book ID: Student Name: Issue