Online Library Management System Using JAVA (BCSE0352)

NIET Greater Noida
Autonomous Institute
Get Future Ready

Department of Computer Science

CS

Submitted To: **Mr. Ajay Kumar (Assistant Professor)**

Name of Member	Roll Number	Role	Department n Section
Mukul Sharma	2401330120097	CODE	Computer Science ‘B’
Riya Chaudhary	2401330120152	Documentation	Computer Science ‘C’
Nitya Yadav	2401330120111	Printing	Computer Science ‘B’

Index

- Introduction
- Literature Survey/Existing System
- Problem Statement
- Proposed Methodology
- Flowcharts and Use Case/ER Diagrams
- Feasibility Study
- Hardware /Software Requirement
- Results and Conclusion
- References
- Live Demonstration of Project with screenshots in ppt

Introduction

- The Library Management System (LMS) is a Java-based desktop application designed to automate and streamline the operations of a library. It helps librarians and users manage books, issue and return records efficiently.
- Developed using Java Swing for GUI and File Handling for data storage, this system replaces manual operations with an efficient and reliable computerized process.

Literature Survey / Existing System

- Traditional library systems involve manual record keeping that is prone to human errors and inefficiency.
- Existing digital systems often depend on heavy database setups or online frameworks, making them complex for small institutions.
- Our proposed system provides a lightweight, standalone Java Swing solution requiring no external database.

Problem Statement

- Manual record management in libraries is time-consuming and error-prone.
- Tracking issued and returned books manually often causes data inconsistency.
- There is a need for an efficient, user-friendly, and automated solution using Java technology.

Proposed Methodology

- Develop a desktop-based Library Management System using Java Swing for the user interface and File Handling for backend storage.
- Include modules for adding, viewing, issuing, and returning books with data validation.
- Provide an intuitive GUI for easy navigation and operation by both library staff and students.

Flowcharts and Use Case / ER Diagrams

- Use Case Diagram:
- - Actors: Librarian, Student
- - Use Cases: Add Book, Issue Book, Return Book, Search Book, View Records
- Flowchart Overview:
- - User logs in → Select operation → Validate input → Update records → Display confirmation

Feasibility Study

- Technical Feasibility: Java Swing is platform-independent and supports rapid desktop GUI development.
- Operational Feasibility: Simple interface requires minimal training for users.
- Economic Feasibility: As it uses only core Java and file handling, no additional hardware or license cost is required.

Hardware / Software Requirement

- Hardware Requirements:
- - Processor: Intel i3 or higher
- - RAM: Minimum 4GB
- - Hard Disk: 100MB free space
- Software Requirements:
- - Operating System: Windows / Linux
- - Development Environment: NetBeans / Eclipse
- - Programming Language: Java (JDK 17 or above)

Results and Conclusion

- The Library Management System efficiently handles book records, issues, and returns through an intuitive Java Swing interface.
- It improves accuracy, reduces manual effort, and enhances record management efficiency.
- Future improvements may include MySQL integration, user authentication, and report generation modules.

LibraryManagementSystem.java > Book > Book(int, String, String)

```java
import java.util.*;

class Book {
    private int id;
    private String title;
    private String author;
    private boolean isBorrowed;

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

    public int getId() { return id; }
    public String getTitle() { return title; }
    public String getAuthor() { return author; }
    public boolean isBorrowed() { return isBorrowed; }

    public void borrowBook() { this.isBorrowed = true; }
    public void returnBook() { this.isBorrowed = false; }

    @Override
    public String toString() {
        return String.format("ID: %d | Title: %s | Author: %s | %s",
                id, title, author, (isBorrowed ? "Borrowed" : "Available"));
    }
}

class Library {
    private List<Book> books = new ArrayList<>();

    public void addBook(Book book) {
        books.add(book);
        System.out.println("✅ Book added successfully!");
    }
```

```
PS C:\Users\Riya choudhary\OneDrive\Desktop\riya java
System.java } ; if ($?) { java LibraryManagementSyster

===== ONLINE LIBRARY MANAGEMENT SYSTEM =====
1. Add Book
2. View Books
3. Borrow Book
4. Return Book
5. Exit
Enter your choice:
```

Thank You