LIBRARY MANAGEMENT SYSTEM

CS23333 – Object Oriented Programming Using JAVA Project Report

Submitted by

ANU N - 231001014

Of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE

NOVEMBER-2024

RAJALAKSHMI ENGINEERING COLLEGE

BONAFIDE CERTIFICATE

Certified that this project titled **"LIBRARY MANAGEMENT SYSTEM"**is the bonafide work of **ANU N (231001014)** who carried out the project work under my supervision.Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr P.Valarmathie

HEAD OF THE DEPARTMENT

Information Technology

Rajalakshmi Engineering College

SIGNATURE

Mrs.S Usha

SUPERVISOR

ASSISTANT PROFESSOR (SG),

Information Technology

Rajalakshmi Engineering College

Submitted to Project Viva-Voce Examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First, we thank the almighty God for the successful completion of the project. Our sincere thanks to our chairman Mr.S. Meganathan,B.E., F.I.E for his sincere endeavour in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson Dr. Thangam Meganathan, for her enthusiastic motivation which inspired us a lot in completing this project and Vice-Chairman Mr. Abhay Shankar Meganathan B.E., M.S., for providing us with the requisite infrastructure We also express our sincere gratitude to our college principal,Dr.S.N.Murugesan M.E.,PhD.,for his kind support and facilities to complete our work on time.We extend heartfelt gratitude to Dr.P.Valarmathie,Professor and Head of the Department of Information Technology for her guidance and encouragement throughout the work. We are very glad to thank our course faculty Dr.A.Kalaivani,Professor of our department for their encouragement and support towards the successful completion of this project.We extend our thanks to our parents, friends, all faculty members,and supporting staff for their direct and indirect involvement in the successful completion of the project for their encouragement and support.

ANU N

TABLE OF CONTENTS

1.LIBRARY MANAGEMENT SYSTEM

1.1 Abstract :

The Library Management System (LMS) is an automated application designed to enhance the efficiency and accuracy of library operations. Built using Java Swing and MySQL in the NetBeans Integrated Development Environment (IDE), the LMS aims to simplify resource management for libraries by replacing traditional manual processes. The system provides a comprehensive solution for book inventory management, user registration, and ensuring seamless library operations.

Key functionalities of the LMS include secure administrator login, book cataloging and user registration. Administrators can add, update, and delete book records and register users. The system is equipped with a reporting module to generate insights on inventory.Its intuitive user interface ensures ease of use, even for non-technical staff.

The LMS is built on a scalable and robust architecture, featuring a Java-based front end and a MySQL database for efficient data storage and retrieval. The client-server architecture ensures reliable data handling, allowing real-time updates and multi-user access. The system's modular design enables easy maintenance and future enhancements, making it suitable for small- to medium-sized libraries.

The LMS also incorporates essential non-functional requirements, such as high performance, reliability, security, and maintainability. It is compatible with standard Windows operating systems and requires minimal hardware resources. The system's user-friendly design ensures smooth adoption and improved productivity for library staff, ultimately enhancing the library experience for users.

In summary, the Library Management System is a versatile and efficient solution for modern library operations. By automating repetitive tasks, reducing errors, and offering a centralized platform for managing resources, the LMS empowers libraries to focus on their core mission: promoting knowledge and learning.

1.2 Introduction :

Libraries serve as essential hubs for knowledge, fostering learning and intellectual growth. They provide access to a wide range of resources, including books, journals, and digital media, supporting students, researchers, and the general public. However, managing these resources effectively poses significant challenges, particularly in traditional manual systems. Issues such as tracking inventory, handling book borrowing and returns, and maintaining accurate records can lead to inefficiencies and errors, undermining the library's mission to serve its users effectively.

The Library Management System offers a comprehensive solution to these challenges by automating critical library operations. By integrating a robust database and a user-friendly graphical user interface (GUI), the system simplifies processes for both administrators and users. Key features include the ability to search for books by genre or title, check availability in real-time, and seamlessly manage borrowing and returning of books. The automation reduces manual effort, minimizes errors, and ensures the accurate tracking of library resources.

This system's intuitive design enhances the user experience by providing clear navigation and instant access to library data. Real-time updates ensure that inventory information remains accurate, enabling users to make informed decisions when borrowing books. Library administrators benefit from reduced administrative overhead, as the system efficiently handles routine tasks such as updating inventory and tracking borrowed items.

Beyond meeting current needs, the Library Management System is designed with scalability in mind, allowing for future enhancements such as the integration of advanced search filters, digital media management, and user analytics. By adopting this system, libraries can optimize resource utilization, improve operational efficiency, and provide a seamless experience for their patrons, ensuring they remain vital resources in the digital age.

1.3 Purpose :

The purpose of this project is to develop a robust application that:

- Simplifies the process of managing books and users.
- Tracks book loans and returns efficiently.
- Provides a secure and scalable solution for library operations.

1.4 Scope of the Project :

The Library Management System is intended for small to medium-sized libraries. It offers:

- Administrator capabilities for book and user management.
- User functionalities to search and borrow books.
- Integration with relational databases to store and retrieve data.
- A foundation for future enhancements, such as advanced reporting or cloud-based deployment.

1.5 Software Requirement Specification :

Introduction :

The Library Management System is designed to manage all aspects of library operations, including book inventory, user registration, and loan transactions.

Document Purpose :

This Software Requirements Specification (SRS) document outlines the technical and functional requirements for the Library Management System. It provides detailed insights into the system's design decisions, architectural framework, and implementation strategies, which are essential for successful deployment and future maintenance.

Product Scope :

The Library Management System is developed to streamline library operations, aiming to replace outdated manual processes. It simplifies book management, user tracking, and loan processing, providing a comprehensive and flexible solution for small- to medium-sized libraries. The system's architecture supports efficient data handling, scalability, and adaptability to changing requirements.

References and Acknowledgement :

[1] Java Swing Tutorials

[2] Java JDBC Guide

Overall Description :

The Library Management System enables librarians and authorized personnel to manage library records seamlessly, including book inventories, user details, and transaction histories. The system automates daily library activities, ensuring smooth operations and minimizing human errors.

Product Perspective :

The system employs a client-server architecture for efficient data handling and processing. The front end is developed using Java Swing for graphical interfaces, while the backend uses MySQL for robust data storage. It is designed to operate on Windows/Linux platforms and integrates seamlessly with relational databases.

Product Functionality :

Key modules of the Library Management System include:

a) Books Available: Displays the current inventory of books.

b) Staff Details: Provides a list of library staff with their information.

c) Add Books: Allows administrators to add new books to the library system

d) Add Staff: Facilitates adding new staff members.

e) Remove Books: Enables the deletion of book records from the system.

f) Remove Staff: Allows administrators to remove staff details.

User and Characteristics :

- Qualification: Users should have basic computer literacy and familiarity with library operations.
- Experience: Experience with library systems or resource management is beneficial.
- Technical Knowledge: Users should be comfortable navigating graphical user interfaces.

Operating Environment :

- Hardware Requirements:
- Processor: Intel i3 or higher
- RAM: 4GB or more
- Hard Disk: 500GB or more
- Display: Resolution of 1024 x 768 or higher

Software Requirements :

- Operating System: Windows 8/10/11 or Linux
- Database: MySQL
- Frontend: Java Swing
- Technology: Java (JDBC)

Constraints :

- System access is restricted to authorized users.
- Database changes are accessible only to administrators.
- Deletion of data is final and requires caution to maintain data integrity.

Assumptions and Dependencies :

- Admins are responsible for creating and managing user credentials securely.
- The system assumes a stable network connection for database interactions.

Specific Requirements :

User Interface :

The Library Management System provides a user-friendly interface for the following operations:

a) Admin Registration and Login.

b) Adding, viewing, and updating book records.

c) Issuing and returning books.

d) Managing member details.

e) Admin account management.

Hardware Interface :

- Screen resolution of at least 640 x 480 pixels
- Compatible with Windows 8, 10, and 11.

Software Interface :

- MS-Windows Operating System.
- Java Swing for the front end.
- MySQL for backend database management.
- IDE: NetBeans.

Functional Requirements :

1. Login Module (LM) :

- Allows administrators to log in securely.
- Validates username and password against the database.
- Implements masked passwords for enhanced security.

2. Book Management Module (BMM) :

- Enables adding, updating, and removing book records.
- Displays available books with search filters like title, author, or genre.

3. Staff Management Module (SMM) :

- Handles adding, updating, and removing staff records.
- Displays staff details, including roles and contact information.

4. Transaction Module (TM) :

- The system must maintain high reliability; in case of abnormal operation or downtime, immediate measures should be implemented to resolve issues and restore functionality.
- Data consistency across book inventory and member records must be ensured at all times.

5. Server Module (SM) :

- Mediates communication between the front end and the database.
- Validates and processes all requests for data manipulation.
- Ensures data consistency and integrity.

Non-functional Requirements :

1.Performance :

- The system must handle real-time operations efficiently, ensuring a response time of less than 2 seconds for accessing book details, staff information, or member records.
- Critical tasks, such as book issuing and returning, must be processed instantly to provide a seamless user experience.

2.Reliability :

- Tracks issued and returned books.
- Updates book availability in real time.

3.Availability :

- Under normal operating conditions, user requests for accessing or updating book and staff details should be processed within 2 seconds to maintain a smooth workflow.
- Immediate feedback should be provided to users during operations like adding or removing books, ensuring clarity and reducing errors.

4.Security :

- A robust security mechanism must be implemented to prevent unauthorized access, safeguard sensitive staff and member information, and maintain the integrity of the system.
- User privacy, including member and staff data, must be securely stored and managed to uphold confidentiality.

5.Maintainability :

- Design documents detailing software and database maintenance procedures must be available to ensure efficient updates and modifications to the Library Management System.
- Administrative access should be provided for maintaining both front-end and back-end components, ensuring the system's long-term reliability and adaptability.

2.SYSTEM FLOW DIAGRAMS

2.1 Use Case Diagrams :

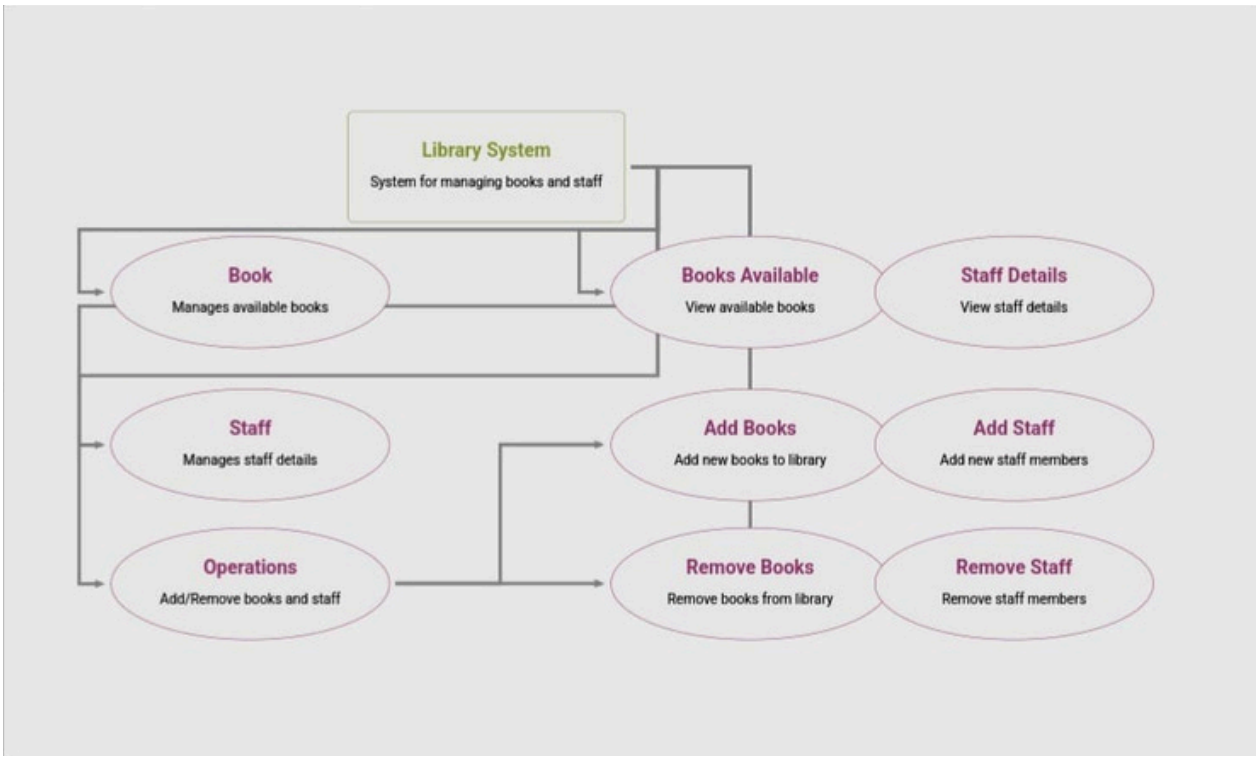


Fig 2.1.1 Use Case Diagram

2.2 Entity-Relationship Diagram :

E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table.

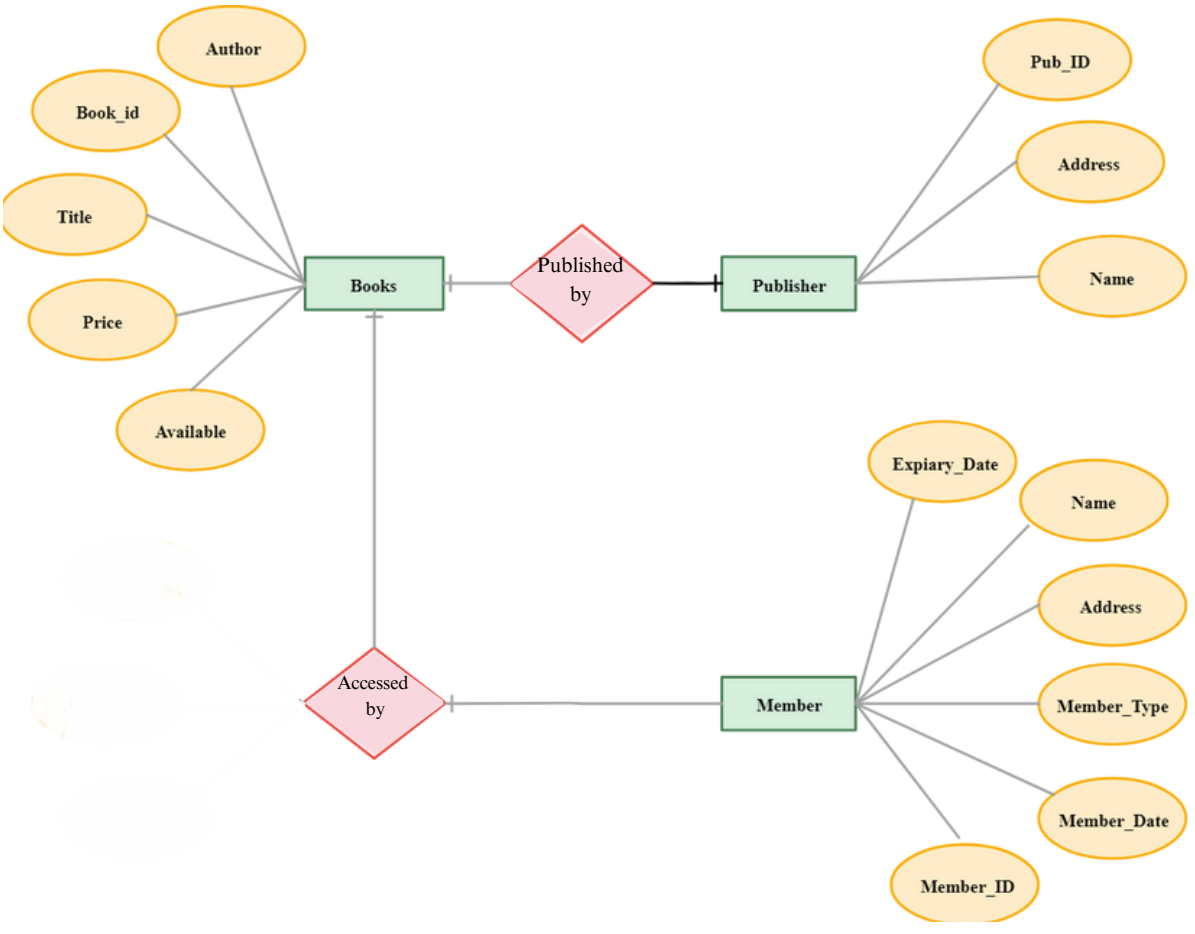


Fig 2.2.1 Entity-Relationship Diagram

2.3 Data-Flow Diagram :

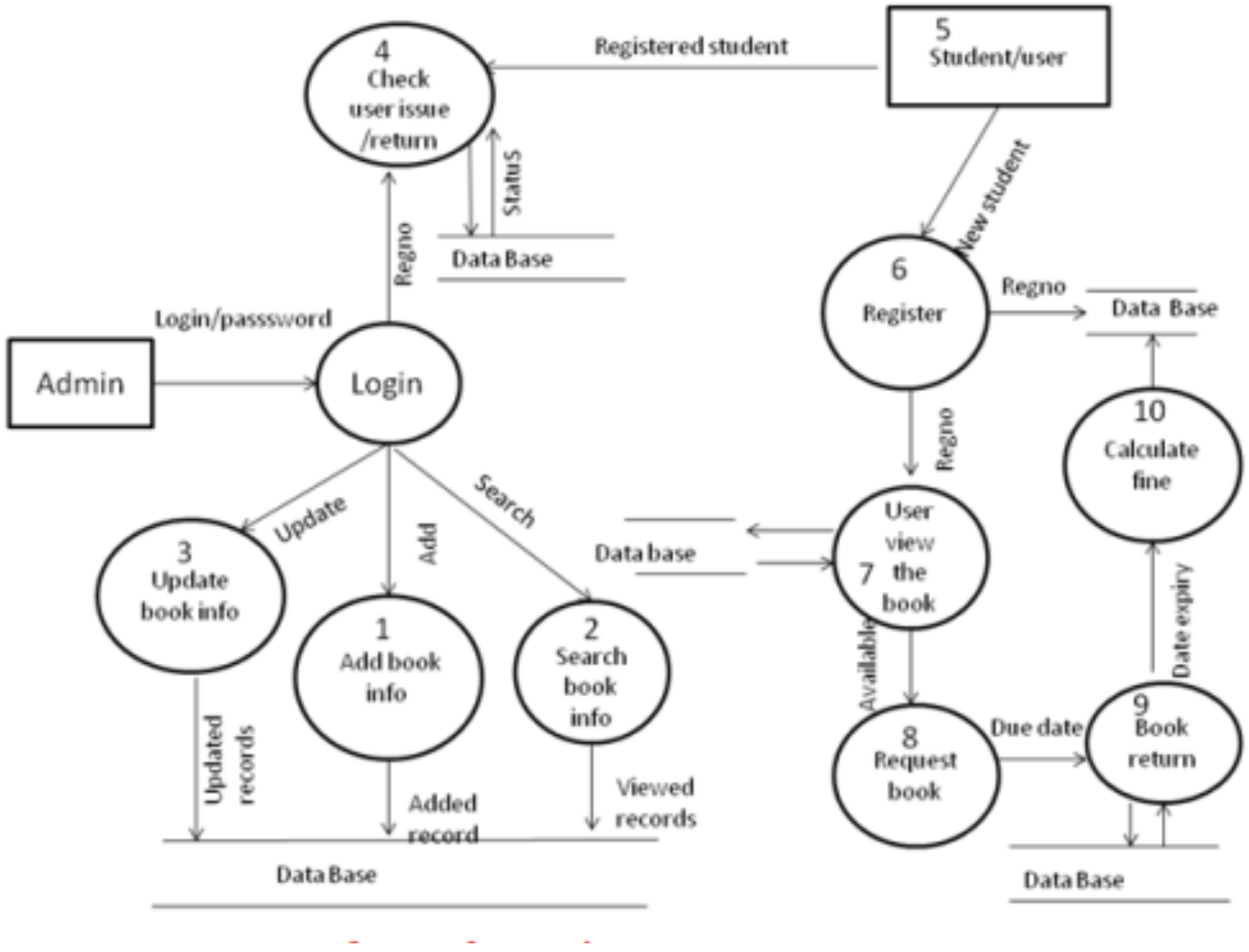


Fig 2.3.1 Data Flow Diagram

3.MODULE DESCRIPTION :

Admin Module :

Registration :

Admin can register a new account using a unique username and password for secure access.

Login :

Admin can log in using their registered username and password to access the system dashboard.

Dashboard Features:

View the list of all available books in the library along with their details, such as title, author, genre, and availability status.

1. Books Available :

View the details of staff members, including their name, role, contact information, and assigned responsibilities.

2.Staff Details :

Add new books to the library database, specifying details such as title, author, ISBN, genre, and quantity.

3. Add Books :

Add new staff members to the system, including their name, contact details, and assigned role.

4.Add Staff :

Add new staff members to the system, including their name, contact details, and assigned role.

5. Remove Books :

Remove books from the library database. This feature allows the admin to delete books no longer available or relevant.

6. Remove Staff :

Remove staff details from the system if they are no longer part of the library team.

This module allows the admin to effectively manage library resources and staff, ensuring smooth operations.

4.DESIGN

4.1 Web Design

Login page :

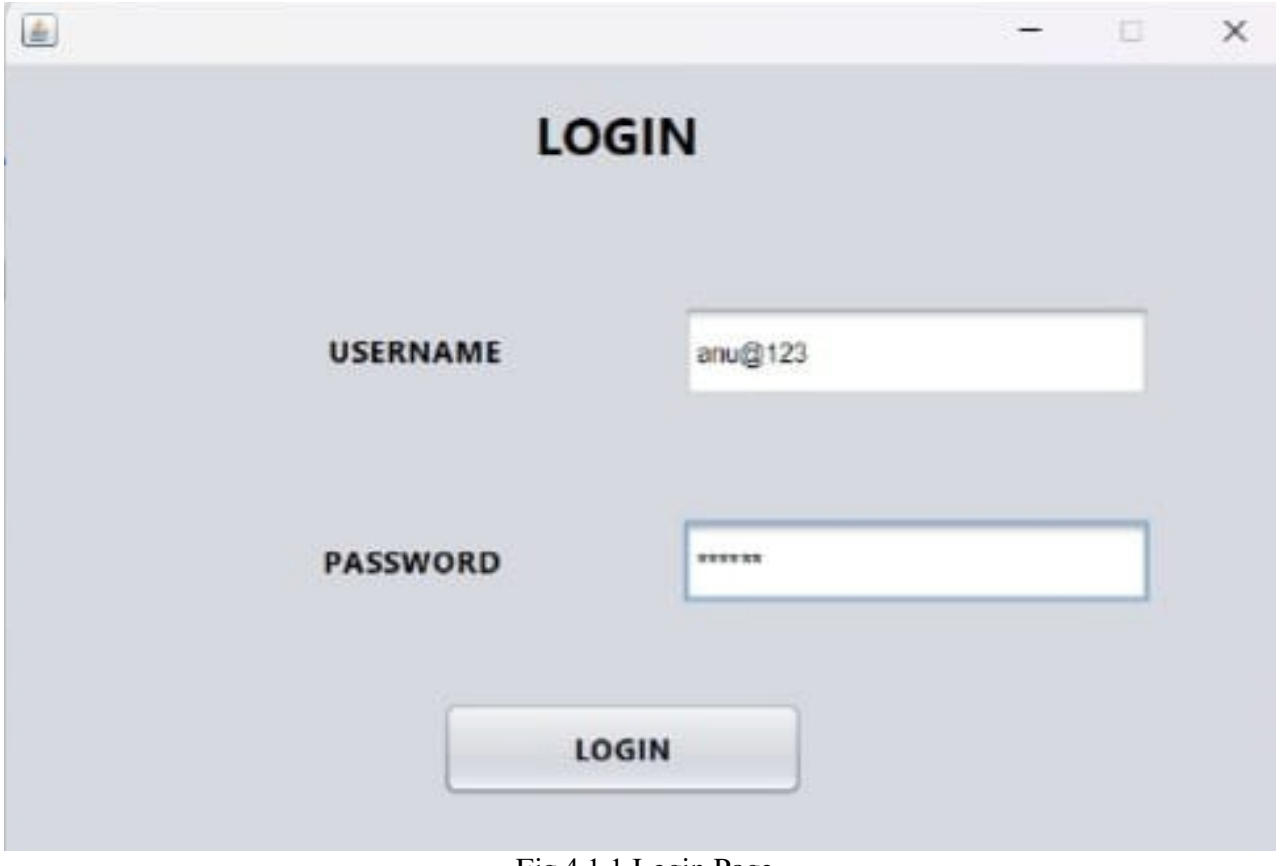


Fig 4.1.1 Login Page

Dashboard :

Fig 4.1.2 Dashboard

Books Available :

Fig 4.1.3 Books Available

Staff Details :

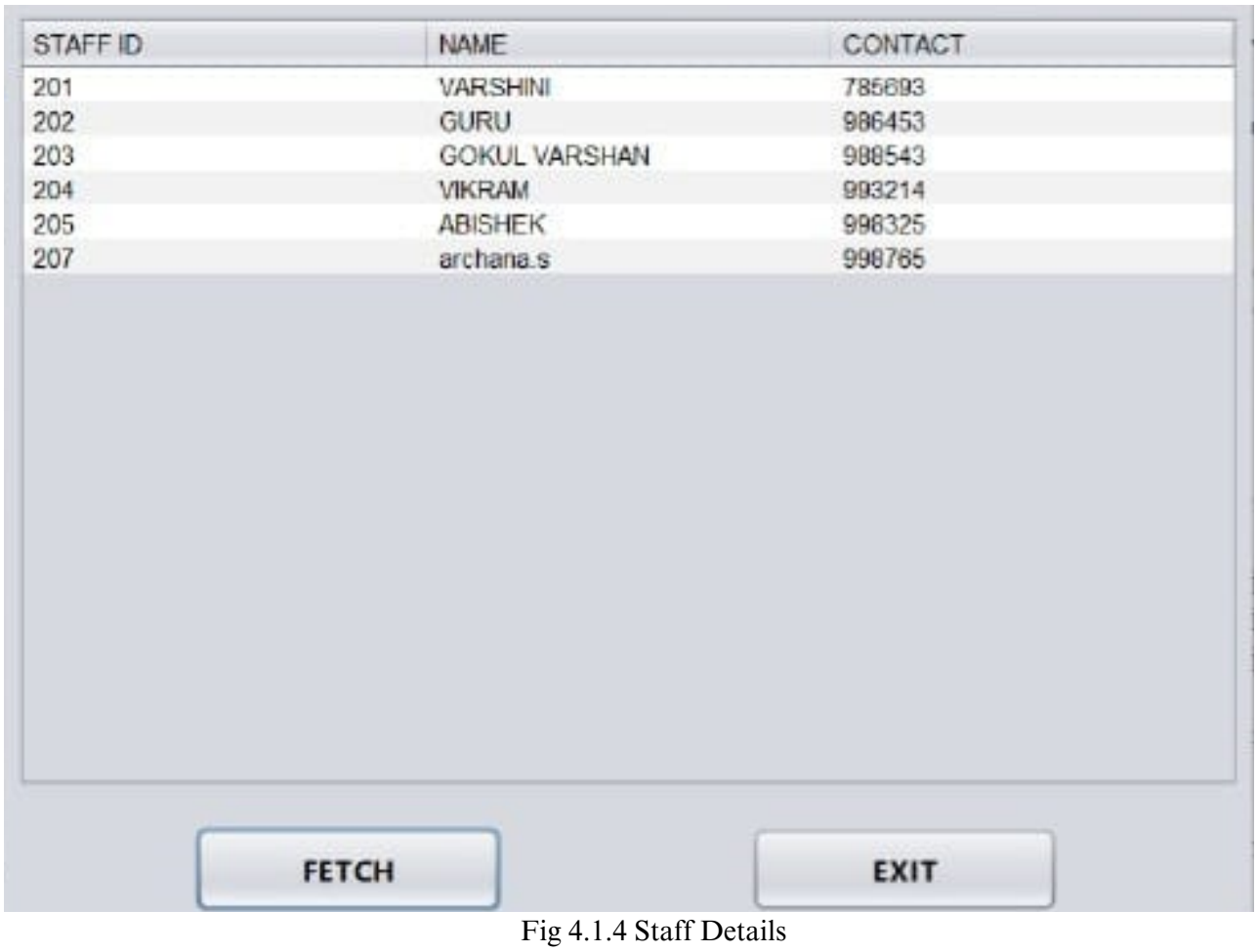


Fig 4.1.4 Staff Details

Add Books :

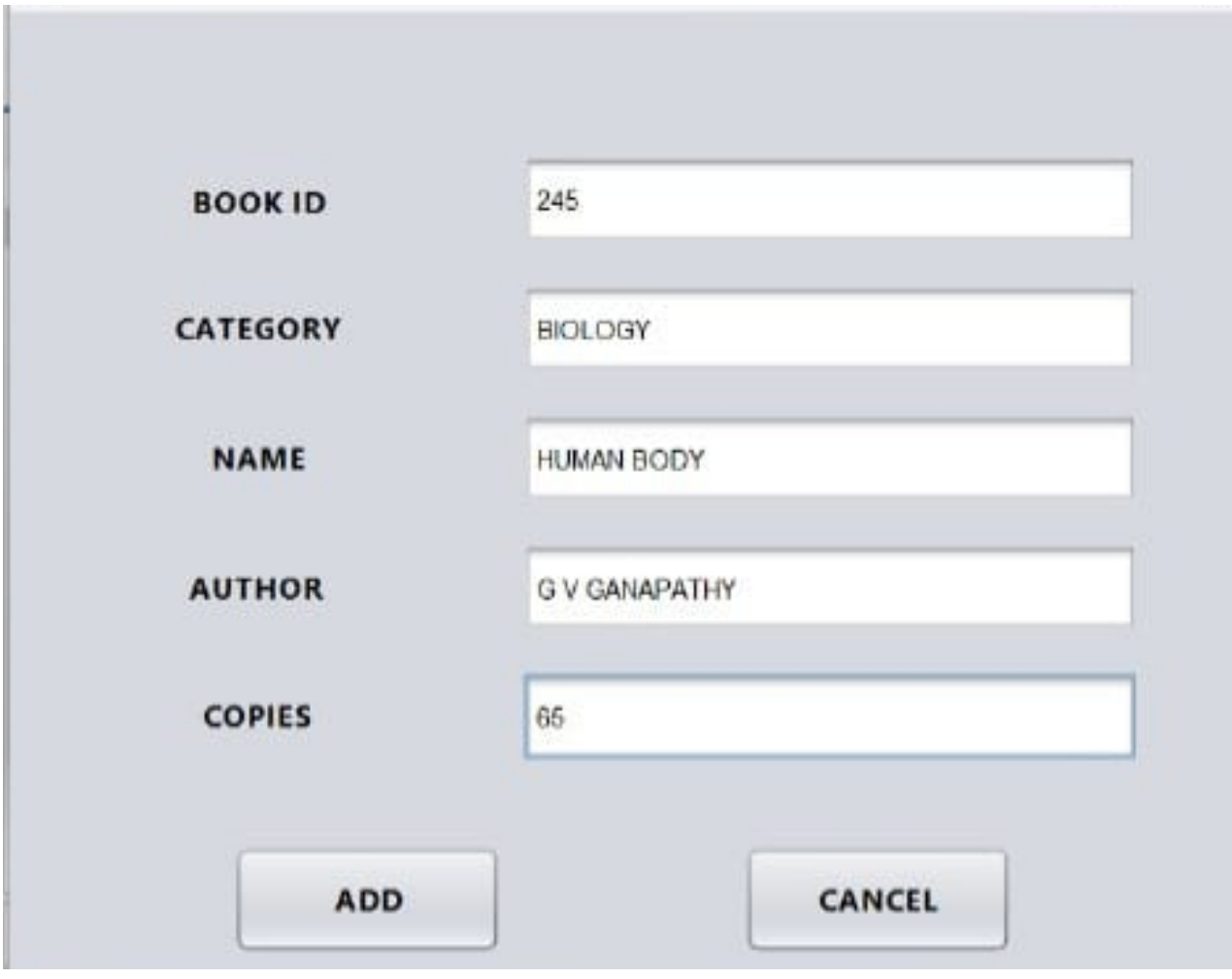


Fig 4.1.5 Add Books

Add Staff :

Fig 4.1.6 Add Staff

Remove Books :

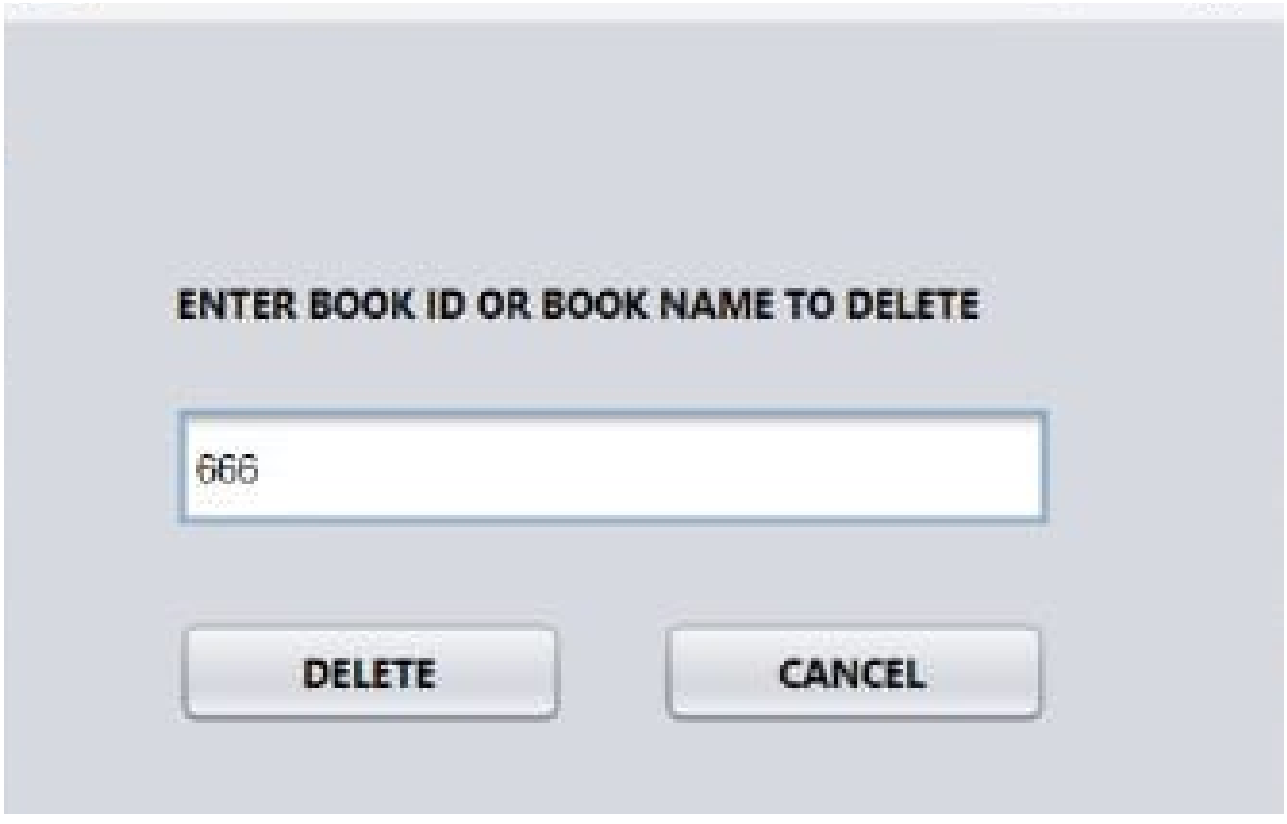


Fig 4.1.7 Remove Books

Edit Staff :

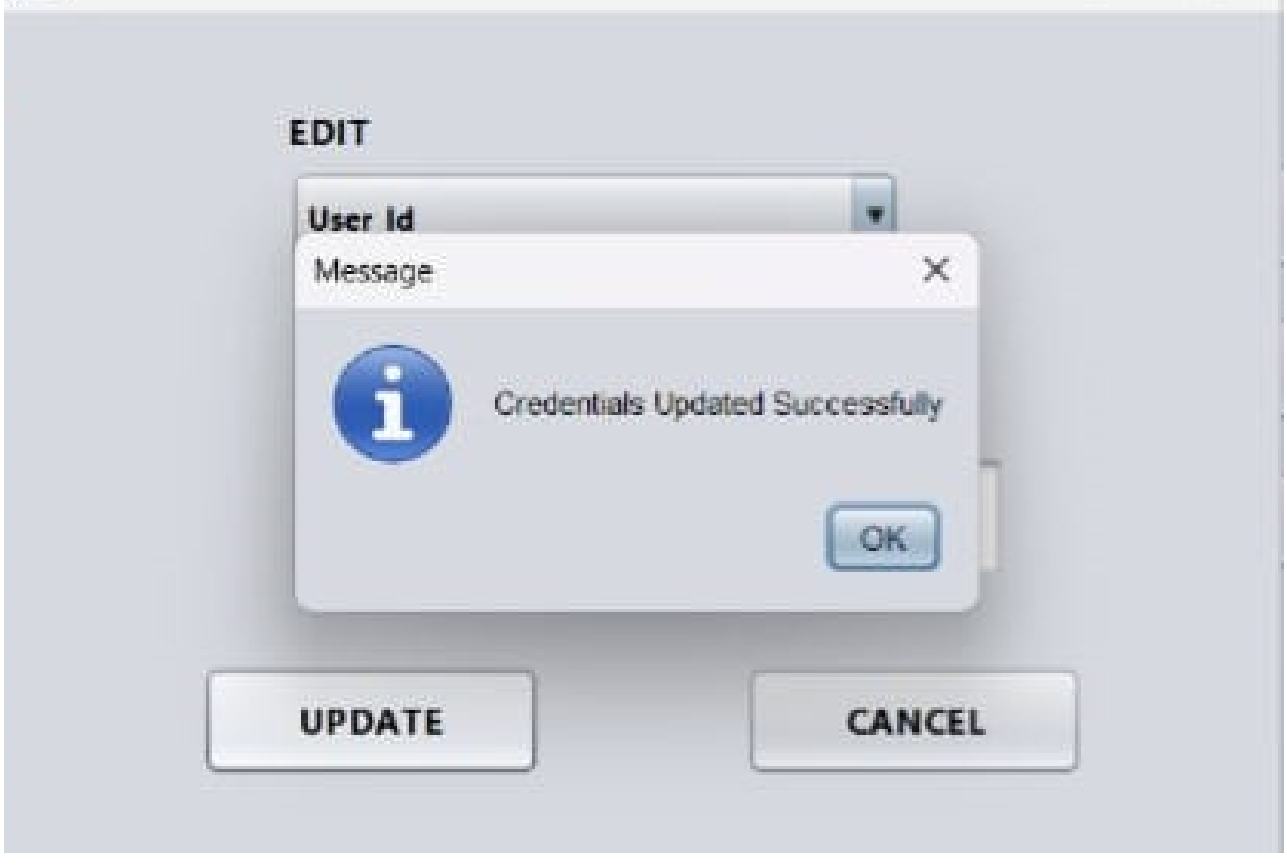
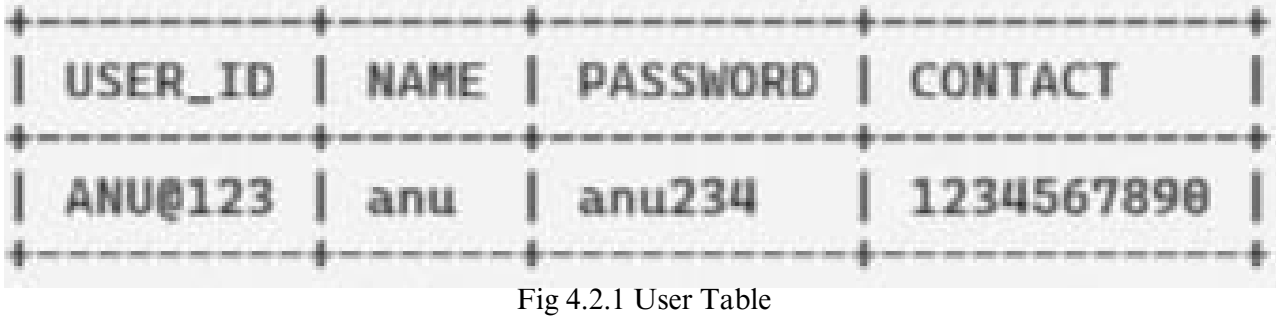


Fig 4.1.8 Edit Staff

4.2 Database Design :

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed.Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required,minimizing chances of data inconsistencies and optimizing for updates. The MySQL database has been chosen for developing the relevant databases.

Library Management System which contains 3 MySQL tables :



USER_ID	NAME	PASSWORD	CONTACT
ANU@123	anu	anu234	1234567890

Fig 4.2.1 User Table

STAFF_ID	NAME	CONTACT
201	VARSHINI	785693
202	GURU	986453
203	GOKUL VARSHAN	988543
204	VIKRAM	993214
205	ABISHEK	998325
207	archana.s	998765

Fig 4.2.2 Staff Table

BOOK_ID	CATEGORY	NAME	AUTHOR	COPIES
201	BIOLOGY	PLANTS AND PHOTOLIGHT	SATHYAMURTHY	20
202	ENGLISH	A TOUGH LIFE	BINDU	15
203	BIOCHEMISTRY	CHEMICAL REACTIONS	ANAND RAO	55
204	SHORT STORIES	ITS A UFO	ELIZABETH COIREE	44
205	ANATOMY	INSIDE US	SATYAJIT YADAV	70
206	BLACK HOLE	OUTER SPACE	ABINAY VARSHAN	55
207	NOVEL	COSMO	GOKUL VARMAN	40
208	PHYSCOLOGY	THOUGHTS	TEHJSHREE ARSHANTH	30
209	SHORT STORIES	FEARLESS	GEETHRANI	50
210	ZOOLOGY	ANIMALS	SHREE JAIKANTH	77
211	zoology	animal life	madhavan	22

Fig 4.2.3 Books Table

4.3 Implementation(Code) :

```
import java.sql.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class LibraryManagementSystem {
    private Connection con;
    private Statement stmt;
    private JComboBox<String> genreComboBox;
    private JComboBox<String> bookComboBox;
    private JTextField numCopiesTextField;
    private JTextArea resultTextArea;
    private JButton borrowButton;
     public LibraryManagementSystem() {
       JFrame frame = new JFrame("Library Management System");
       frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create the main panel
        JPanel mainPanel = new JPanel();
        mainPanel.setLayout(new BorderLayout());

        // Create the top panel (genre, book, copies)
        JPanel topPanel = createTopPanel();
        topPanel.setBackground(Color.LIGHT_GRAY);

        // Create the result panel
        JPanel resultPanel = createResultPanel();

        // Create the button panel
        JPanel buttonPanel= new JPanel(new GridBagLayout());

        GridBagConstraints gbc = new GridBagConstraints();
        gbc.anchor = GridBagConstraints.CENTER;
        buttonPanel.add(createBorrowButton(), gbc);
```

```
    // Add panels to the frame
    frame.add(topPanel, BorderLayout.NORTH);
    frame.add(resultPanel, BorderLayout.CENTER);
    frame.add(buttonPanel, BorderLayout.SOUTH);
    frame.pack();
    frame.setVisible(true);
    setupDatabaseConnection();
}
private JPanel createTopPanel() {
    JPanel topPanel = new JPanel(new FlowLayout());
    genreComboBox = new JComboBox<>();
    genreComboBox.addItem("Select Genre");
    genreComboBox.addItem("Fiction");
    genreComboBox.addItem("Science");
    genreComboBox.addItem("History");
    genreComboBox.setSelectedItem("Select Genre");
    bookComboBox = new JComboBox<>();
    bookComboBox.addItem("Select Book");
    numCopiesTextField = new JTextField(5);
    topPanel.add(new JLabel("Genre:"));
    topPanel.add(genreComboBox);
    topPanel.add(new JLabel("Book:"));
    topPanel.add(bookComboBox);
    topPanel.add(new JLabel("Copies:"));
    topPanel.add(numCopiesTextField);

    genreComboBox.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            populateBooks();
        }
    });

    return topPanel;
}
```

```
private JPanel createResultPanel() {
    JPanel resultPanel = new JPanel(new BorderLayout());
    resultTextArea = new JTextArea(10, 30);
        resultPanel.add(new JScrollPane(resultTextArea), BorderLayout.CENTER)
    return resultPanel;
}

private JButton createBorrowButton() {
    borrowButton = new JButton("Borrow Book");
    borrowButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            borrowBook();
        }
    });
    return borrowButton;
}

private void setupDatabaseConnection() {
    try {
        con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/library","root","root");
        stmt = con.createStatement();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
private void populateBooks() {
       String selectedGenre = (String)
       genreComboBox.getSelectedItem();
       bookComboBox.removeAllItems();
       bookComboBox.addItem("Select Book");
```

```
    if (!selectedGenre.equals("Select Genre")) {
        try {
            String query = "SELECT BookTitle FROM books
            WHERE Genre = '" + selectedGenre + "';";
            ResultSet rs = stmt.executeQuery(query);
            while (rs.next()) {
              bookComboBox.addItem(rs.getString
              ("BookTitle"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

private void borrowBook() {
        String selectedGenre=(String)
        genreComboBox.getSelectedItem();
        String selectedBook=(String)
        bookComboBox.getSelectedItem();
       int numCopies;

       try {
        numCopies = Integer.parseInt
        (numCopiesTextField.getText());
    } catch (NumberFormatException e) {
        resultTextArea.setText
        ("Please enter a valid number of copies.");
        return;
    }

    if (selectedGenre.equals("Select Genre") ||
    selectedBook.equals("Select Book")) {
              resultTextArea.setText("Please select a valid  genre and book.");
     return;
   }
```

```
try {
    String query = "SELECT CopiesAvailable FROM books WHERE BookTitle = '" + selectedBook + "';";
    ResultSet rs = stmt.executeQuery(query);

    if (rs.next()) {
        int availableCopies = rs.getInt("CopiesAvailable");

        if (availableCopies >= numCopies) {
            query = "UPDATE books SET CopiesAvailable = " + (availableCopies - numCopies) +
                    " WHERE BookTitle = '" + selectedBook + "';";
            stmt.executeUpdate(query);

            resultTextArea.setText("Borrow successful!\nRemaining copies of \"" +
                                   selectedBook + "\": " + (availableCopies - numCopies));
        } else {
            resultTextArea.setText("Not enough copies available.");
        }
    } else {
        resultTextArea.setText("Book not found.");
    }
} catch (SQLException e) {
    e.printStackTrace();
}

// Clear inputs
genreComboBox.setSelectedItem("Select Genre");
bookComboBox.removeAllItems();
bookComboBox.addItem("Select Book");
numCopiesTextField.setText("");

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

5.CONCLUSION

The Library Management System project, developed with diligent planning and implementation, reflects a user-centric approach to efficient library management. With features like browsing available books, borrowing, and genre-based search, the system provides an intuitive interface for both administrators and users. The incorporation of database integration ensures accurate book tracking and availability management, enhancing operational efficiency.

The project emphasizes data integrity and usability, demonstrated through robust handling of borrow and availability features. This comprehensive solution not only addresses current library requirements but also offers scalability for future enhancements, such as adding advanced search options, member management, and detailed analytics.

6.REFERENCES

[1] https://www.javatpoint.com/java-awt

[2] https://www.javatpoint.com/java-swing

[3] https://www.mysql.com/