

JAVA PROJECT REPORT

(Project Term January-May 2023)

Library Management System for easy access to information

Submitted by

Shabd Jain
Yogesh Verma
Srajan Gupta

Registration Number : 12112449
Registration Number : 12112394
Registration Number : 12112768

Project Group Number : 5

Course Code : CSE310

Under the Guidance of

Dr.A.Ranjith Kumar

School of Computer Science and Engineering



L OVELY
P ROFESSIONAL
U NIVERSITY

DECLARATION

We hereby declare that the project work entitled (“Library Management System”) is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Dr.A.Ranjith Kumar, during January to May 2023. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Project Group Number: 5

Name of Student 1: Shabd Jain

Registration Number: 12112449

Name of Student 2: Yogesh Verma

Registration Number: 12112394

Name of Student 3: Srajan Gupta

Registration Number: 12112768

Signature: Shabd

Date: 17/04/23

Signature: Yogesh

Date: 17/04/23

Signature: Srajan

Date: 17/04/23

TABLE OF CONTENTS

Inner first page.....	(i)
Declaration.....	(ii)
Table of Contents.....	(iii)

1. INTRODUCTION	1
2. Scope of the project	2
3. Proposed Technique	3
3.1. Module 1	4
3.2. Module 2	5
4. Sample Code	6
5. Conclusion	7

1. INTRODUCTION

A Library Management System (LMS) is a software application designed to manage and automate the operations of a library. The LMS typically consists of a set of integrated modules that handle different library functions, such as cataloging, circulation, acquisition, serials, and reporting. It is used by librarians and library staff to efficiently manage and provide access to library resources.

The key features of a library management system include:

1. **Cataloging:** The LMS creates and maintains a bibliographic database of all library materials, including books, journals, magazines, audiovisual materials, and other resources.
2. **Circulation:** The LMS automates the process of checking materials in and out of the library, tracking borrowing history, and managing overdue materials.
3. **Acquisition:** The LMS tracks the ordering and receipt of new materials, as well as the processing of invoices and payments.
4. **Serials:** The LMS manages the receipt and processing of periodicals and other serials, including managing subscriptions, tracking issues, and handling renewals.
5. **Reporting:** The LMS generates reports on library usage, circulation, and inventory, providing insights into library operations and usage patterns.

Other features of an LMS may include the ability to manage digital resources, provide online access to library resources, and integrate with other library systems or external databases.

An LMS can improve the efficiency and effectiveness of library operations, making it easier for library staff to manage resources, provide access to materials, and serve the needs of library users. It can also help libraries better understand their users' needs and preferences, and optimize their collections and services accordingly.

2. Scope of the project

Generally, a library management system includes the following features:

1. Cataloging: The system should allow librarians to create and maintain a catalog of library materials, including books, journals, videos, and other items.
2. Circulation: The system should allow librarians to manage the circulation of library materials, including checking items in and out, renewing loans, and tracking overdue items.
3. Patron management: The system should allow librarians to manage patron accounts, including creating new accounts, updating contact information, and tracking borrowing history.
4. Acquisitions: The system should allow librarians to manage the process of acquiring new library materials, including ordering, receiving, and paying for items.
5. Reporting: The system should provide a range of reports that allow librarians to track library usage, collection development, and other metrics.
6. Interlibrary loan: The system should allow librarians to request and manage interlibrary loans, allowing patrons to borrow materials from other libraries.
7. Digital resource management: The system should allow librarians to manage digital resources, including e-books, e-journals, and other electronic materials.

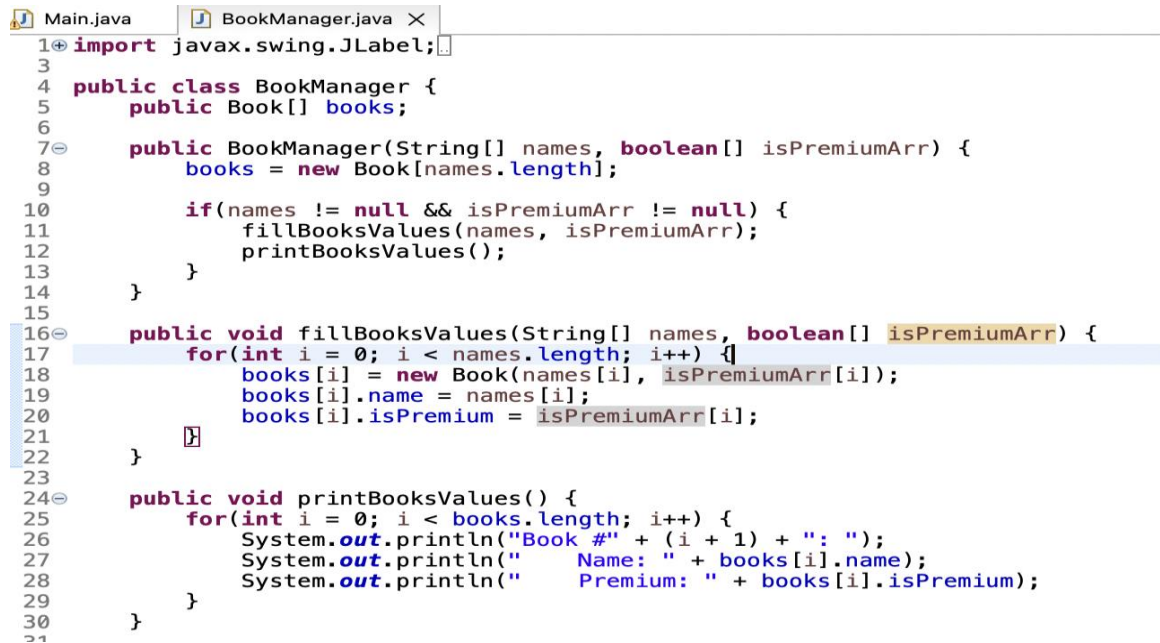
Overall, the scope of a library management system should be comprehensive enough to support the needs of the library and its patrons, while also being flexible enough to adapt to changing needs over time.

3. Proposed Technique

Some proposed techniques for building a library management system:

1. User management: The system should have a feature to manage users, including the ability to create new users, edit user information, and delete users.
2. Book management: The system should have a feature to manage books, including the ability to add new books, edit book information, and delete books.
Additionally, the system should allow users to search for books by title, author, ISBN, or any other relevant keyword.
3. Circulation management: The system should have a feature to manage book circulation, including the ability to check out and check in books, renew books, and track overdue books.
4. Fine management: The system should have a feature to manage fines, including the ability to calculate fines for overdue books, send notifications to users with overdue books, and collect fines from users.
5. Reporting: The system should have a feature to generate reports, including reports on books, users, fines, and circulation.
6. Security: The system should be secure, with features such as password protection, user authentication, and data encryption.
7. Integration with external systems: The system should be able to integrate with external systems, such as online catalogs, databases, and other library systems.
8. Mobile compatibility: The system should be mobile-friendly, allowing users to access the system from their mobile devices.
9. Accessibility: The system should be accessible to users with disabilities, with features such as text-to-speech and screen reader compatibility.

3.1. Module 1



```
1 import javax.swing.JLabel;
2
3
4 public class BookManager {
5     public Book[] books;
6
7     public BookManager(String[] names, boolean[] isPremiumArr) {
8         books = new Book[names.length];
9
10        if(names != null && isPremiumArr != null) {
11            fillBooksValues(names, isPremiumArr);
12            printBooksValues();
13        }
14    }
15
16    public void fillBooksValues(String[] names, boolean[] isPremiumArr) {
17        for(int i = 0; i < names.length; i++) {
18            books[i] = new Book(names[i], isPremiumArr[i]);
19            books[i].name = names[i];
20            books[i].isPremium = isPremiumArr[i];
21        }
22    }
23
24    public void printBooksValues() {
25        for(int i = 0; i < books.length; i++) {
26            System.out.println("Book #" + (i + 1) + ": ");
27            System.out.println("    Name: " + books[i].name);
28            System.out.println("    Premium: " + books[i].isPremium);
29        }
30    }
31}
```

```
public void fillBooksValues(String[] names, boolean[] isPremiumArr) {
```

```
    for(int i = 0; i < names.length; i++) {
```

```
        books[i] = new Book(names[i], isPremiumArr[i]);
```

```
        books[i].name = names[i];
```

```
        books[i].isPremium = isPremiumArr[i];
```

```
    }
```

```
}
```

```
public void printBooksValues() {
```

```
    for(int i = 0; i < books.length; i++) {
```

```
        System.out.println("Book #" + (i + 1) + ": ");
```

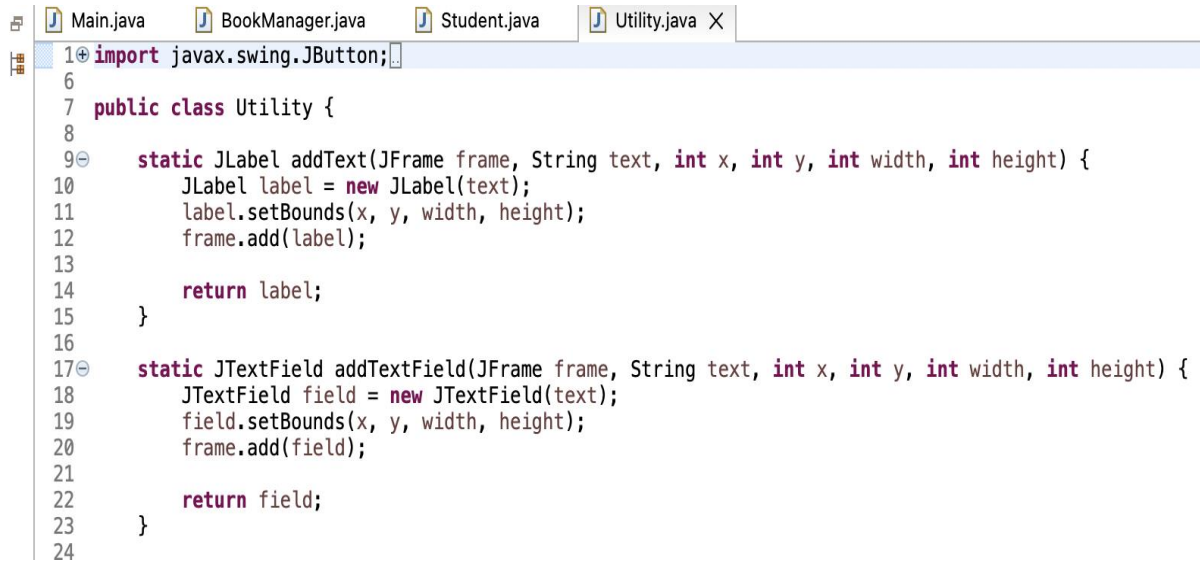
```
        System.out.println("    Name: " + books[i].name);
```

```
        System.out.println("    Premium: " + books[i].isPremium);
```

```
    }
```

```
}
```

3.2. Module 2



```
1 import javax.swing.JButton;
6
7 public class Utility {
8
9     static JLabel addText(JFrame frame, String text, int x, int y, int width, int height) {
10         JLabel label = new JLabel(text);
11         label.setBounds(x, y, width, height);
12         frame.add(label);
13
14         return label;
15     }
16
17     static JTextField addTextField(JFrame frame, String text, int x, int y, int width, int height) {
18         JTextField field = new JTextField(text);
19         field.setBounds(x, y, width, height);
20         frame.add(field);
21
22         return field;
23     }
24
```

```
public class Utility {

    static JLabel addText(JFrame frame, String text, int x, int y, int width, int height) {

JLabel label = new JLabel(text);

label.setBounds(x, y, width, height);

frame.add(label);


    return label;

}

    static JTextField addTextField(JFrame frame, String text, int x, int y, int width, int height) {

JTextField field = new JTextField(text);

field.setBounds(x, y, width, height);

frame.add(field);


    return field;

}
```


4. Sample Code

```
public void actionPerformed(ActionEvent e) {
    System.out.println("Login button pressed!");

    System.out.println("Entered name: " + enteredName.getText());
    System.out.println("Entered password: " + enteredPassword.getText());

    String enteredNameString = enteredName.getText();
    String enteredPasswordString = enteredPassword.getText();

    System.out.println("Button group: " + buttonGroup);

    if(manager.checkIsStudentEligible(enteredNameString, enteredPasswordString)) {
        System.out.println("Logged in!");

        bookManager.displayEligibleBooks(manager.getStudentByName(enteredNameString), area);

        issueButton.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                bookManager.issueBookByName(manager.getStudentByName(enteredNameString), issueField.getText(), issueLabel);
            }

        });

        returnButton.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO Auto-generated method stub
                bookManager.returnBookByName(manager.getStudentByName(enteredNameString), issueLabel);
            }

        });

    } else {
        System.out.println("Can't log in. :(");
    }
}
```

```
public void actionPerformed(ActionEvent e) {

    System.out.println("Login button pressed!");

    System.out.println("Entered name: " + enteredName.getText());

    System.out.println("Entered password: " + enteredPassword.getText());

    String enteredNameString = enteredName.getText();

    String enteredPasswordString = enteredPassword.getText();

    System.out.println("Button group: " + buttonGroup);

    if(manager.checkIsStudentEligible(enteredNameString, enteredPasswordString)) {

        System.out.println("Logged in!");

        bookManager.displayEligibleBooks(manager.getStudentByName(enteredNameString), area);

        issueButton.addActionListener(new ActionListener()
```

5. Conclusion

A library management system is a powerful tool for efficiently managing library operations, from cataloging and circulation to patron management and reporting. It can help librarians save time and improve services for patrons by automating many of the routine tasks associated with library management.

By using a library management system, libraries can track and manage their collections more effectively, improving access to resources and providing better customer service to patrons. The system can also help libraries save money by reducing labor costs and improving inventory management.

Overall, implementing a library management system can help libraries stay competitive in today's digital age and provide a better experience for patrons.