

Scalable Library Management System

Sravani Banala



Table of Contents

**Problem
Statement**

01

02

**Design &
Analysis**

Implementation

03

04

Big learnings

Problems with maintenance and care have existed from the beginning of time

Most libraries give readers the impression that they are highly disorganized in the absence of a good system

Maintaining and preserving historical records and materials requires great care

The problem we are trying to solve is the management system in a library



01

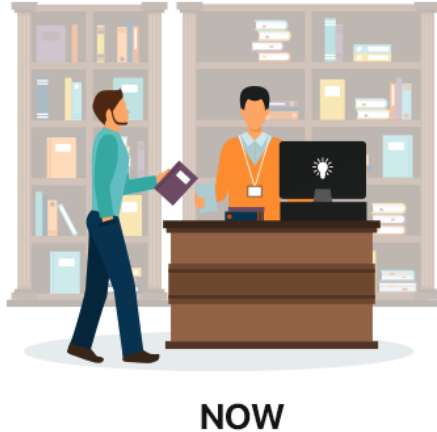
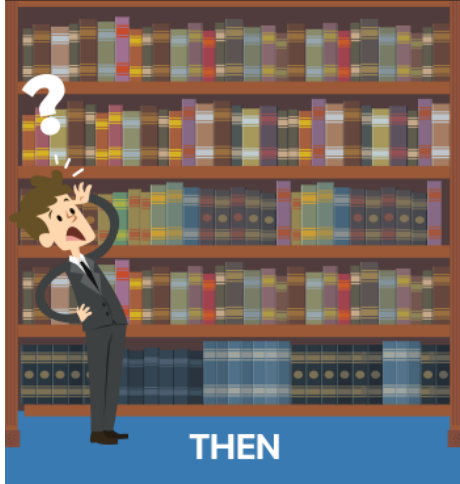
Problem Statement

The Idea

- Library is a collection of different types of books and provides books for large number of readers. Thus, we need a software for managing the Library.
- A library management system allows for the organised handling of book records and book readers
- The primary objective of library Management system is to handle entire activity of library electronically
- Library Management System is built using Java, JavaFX which Mainly focuses on basic operations in a library like adding new students, new books, searching books to issue and return books



WHY SHOULD LIBRARIES INVEST IN A LIBRARY MANAGEMENT SOFTWARE SYSTEM?



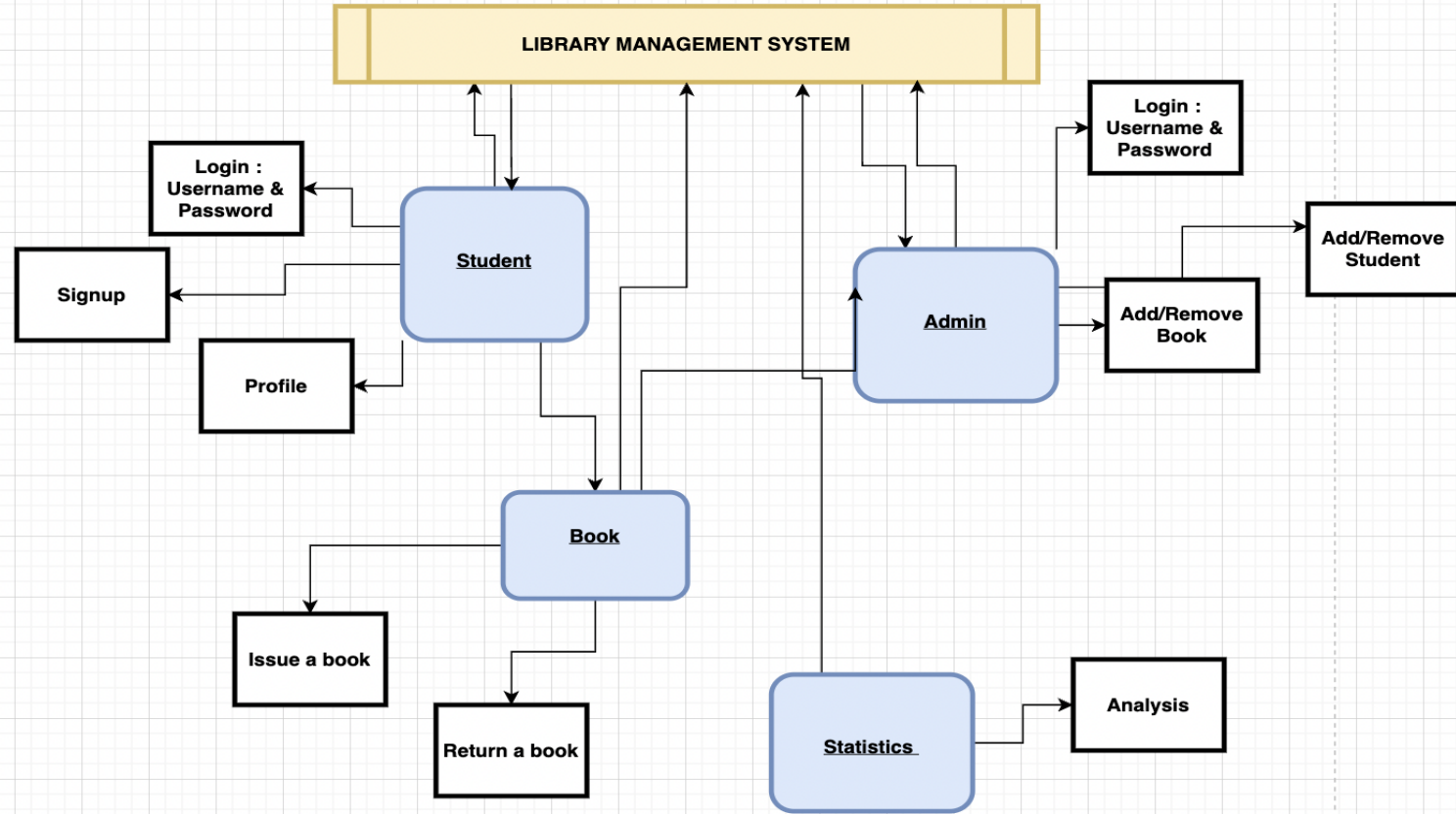
- It provides better and efficient service to students with huge collection of textbooks
- It helps both students and librarian to keep constant track on books available in library and books issued to students.
- All details will be available on a click for both user and librarian

Design & Analysis



- We implemented the project using Java concept, JavaFX
- Tools such as scene builder, MySQL workbench were used to create the UI and manage the data
- We maintained our progress on Trello and used google drive to store all the documents so all of us can access them
- We pushed the source code to a git repository

Use cases and interactions in our design



Design & Analysis



- Our project was designed using two use-cases:
 1. Admin
 2. User/Student
- The Librarian/Admin gets the same interface as the User/Student wherein they can login into their respective dashboards using their Username and Password. In case of a librarian, he/she can add/remove a student as well as books based on their availability
- A Student/User can access their profile by logging in or signup to be a member of said library
- The book is a commodity that can be issued or returned and, in the UI, has an impact on most use cases

Implementation

Student

- Login
- Signup
- Profile



Admin

- Login
- Add/Remove - Student, Books



Issue/Return Books

- Issue
- Return



Statistics

- Analysis



LOGIN

Student/Admin Login

1. There are two buttons: one is for Admin login and the other is for Students
2. Requesting User to input their Student id and Password in the same way they did on the signup page, before login checking whether the user has filled out all the boxes on the student login page, if not asking user to enter all the fields, then switch to the Student.fxml page if the information was entered correctly
3. Similarly, Admin enters Admin's Name and Password; if entered correctly, Admin.fxml is switched to.
4. For the Student Signup page we have name of the student, father's name, date of birth, email, which course student is into. After entering all the details request goes to admin from addstudent database, for the approval and admin sees the details and approves the student and inserts the data into student database

```
if(tfAdminName.getText().length()==0 || pfPass.getText().length()==0)
{
    a.setContentText("Please fill all fields");
    a.show();
    return;
}
try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    con = DriverManager.getConnection(HelloApplication.dbUrl, HelloApplication.userName, HelloApplication.password);
    stmt = con.createStatement();

    String sql = "SELECT * FROM admin";
    ResultSet rs = stmt.executeQuery(sql);
    boolean present = false;
    while(rs.next()) {
        if (rs.getString("username").equalsIgnoreCase(tfAdminName.getText()) && (rs.getString("password").equals(pfPass.getText()))) {
            present = true;
            break;
        }
    }
    if(present){
        HelloApplication.SceneSwitch( "Admin.fxml", "Admin", (Stage)btnLogin.getScene().getWindow() );
    }
    else {
        a.setContentText("Incorrect Name or Password");
    }
}
```

```
}
HelloApplication.studentID=Integer.parseInt(tfStudentId.getText());
try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    con = DriverManager.getConnection(HelloApplication.dbUrl, HelloApplication.userName, HelloApplication.password);
    stmt = con.createStatement();

    String sql = "SELECT id, password " + " FROM student";
    ResultSet rs = stmt.executeQuery(sql);
    boolean present = false;
    while(rs.next()) {
        if (rs.getInt("id")==Integer.parseInt(tfStudentId.getText()) && (rs.getString("password").equals(pfPass.getText()))) {
            present = true;
            break;
        }
    }
    if(present){
        tfStudentId.setText("");
        pfPass.setText("");

        HelloApplication.SceneSwitch( "Student.fxml", "Student", (Stage)btnLogin.getScene().getWindow() );
    }
    else {
        Alert a = new Alert(Alert.AlertType.ERROR);
        a.setContentText("Incorrect Name or Password");
    }
}
```

ISSUE/RETURN BOOKS

Student Requesting Book :

1. In the student issueBook.fxml we see the status of the books as if the book is issued, no issuable, requested, then the student requests the book and send the information the admin to issue the book. Here the student requests the book from book database then sends the requests to the admin using issue book database inserting the values for studentid, bookid, bookname, studentname to the admin to grant the access.

Admin Issuing Book :

1. Here the Admin gets the request from the user and access it out in AdminIssueBookController, to issue the book admin needs to add the issuedate, duedate and then issue the book, while issuing the book the request needs to be deleted and updated to issued from book database as issued and delete the request from issue book
2. In the same way if the admin wants to cancel the request of user, updates the status of book as issuable again and deletes the request from the issue book
3. All the above is viewed on the AdminIssueBook.fxml by using showIssueBook Action



ISSUE/RETURN BOOKS

```
String query = "SELECT * FROM book";
ResultSet rs = stmt.executeQuery(query);

while(rs.next()){
    if(rs.getInt("id") == id){
        if(rs.getString("status").equalsIgnoreCase("Issueable")){
            //String s = "INSERT INTO book(studentid, issuedate)" + "VALUES('"+HelloApplication.studID+"', '" +
            String s = "INSERT INTO issuebook(bookid, bookname, studentid)" + "Values('" + rs.getInt("id")+",'
            Statement st = con.createStatement();
            st.execute(s);

            String update = "UPDATE book SET status = 'Requested' WHERE id = " + id +"";
            PreparedStatement ps = con.prepareStatement(update);
            ps.executeUpdate();
            showBookTable();

            Alert a = new Alert(Alert.AlertType.INFORMATION);
            a.setContentText("Request sent to admin to issue book");
            a.show();
        }
        else if(rs.getString("status").equalsIgnoreCase("Not Issueable")){
            Alert a = new Alert(Alert.AlertType.ERROR);
            a.setContentText("This book can not be issueable to the student");
            a.show();
        }
        else if(rs.getString("status").equalsIgnoreCase("Issued")){
            Alert a = new Alert(Alert.AlertType.ERROR);
```

Student Requesting Book

```
public void IssueAction(ActionEvent actionEvent) {
    int id = tIssueBook.getSelectionModel().getSelectedItem().getId();
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        con = DriverManager.getConnection(HelloApplication.dbUrl, HelloApplication.userName, HelloApplication.p
        stmt = con.createStatement();

        if(dpIssueDate.getValue() == null || dpDueDate.getValue() == null){
            Alert a = new Alert(Alert.AlertType.INFORMATION);
            a.setContentText("Please enter issue and due date");
            a.show();
            return;
        }
        java.util.Date date1 = java.util.Date.from(dpIssueDate.getValue().atStartOfDay(ZoneId.systemDefault()).
        java.sql.Date sqlIssueDate = new java.sql.Date(date1.getTime());
        java.util.Date date2 = java.util.Date.from(dpDueDate.getValue().atStartOfDay(ZoneId.systemDefault()).to
        java.sql.Date sqlDueDate = new java.sql.Date(date2.getTime());
        String sql = "DELETE FROM issuebook WHERE bookid = " + id +"";
        stmt.execute(sql);
        String query = "UPDATE book SET status = 'Issued', studentid = "+tIssueBook.getSelectionModel().getSele
        PreparedStatement ps = con.prepareStatement(query);
        ps.executeUpdate();
        dpIssueDate.setValue(null);
        dpDueDate.setValue(null);
        Alert a = new Alert(Alert.AlertType.INFORMATION);
        a.setContentText("Book Issued to Student Successfully");
        a.show();
```

Admin Issuing Book



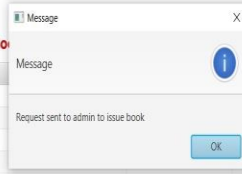
ISSUE BOOK

Select Book Item to Issue or Search Book

BOOK ID	BOOK NAME	Message		TYPE	STATUS
1	Jannat K Patty				Requested
2	Linear Algebra			Reference Book	Not Issueable
9	Peer e Kaamil				Requested
11	Mushaf			Fresh Arrival	Requested
15	Aab e Hayyat	Umerah Ahmad	2022-07-06	Book	Requested
16	The Success Principles	Jack Canfield	2022-07-15	Book	Issueable
18	Namal	Nimrah Ahmad	2022-06-29	Book	Requested
22	Harry Potter and the Deathly ...	J.K Rowling	2022-06-01	Book	Issued
27	Reference	Maheshwar	2022-11-29	Fresh Arrival	Issueable after 2 months...
28	hai	rahul	2022-11-30	Reference Book	Not Issueable
29	29	hello	2019-12-25	Book	Issueable

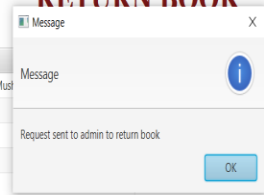
You can issue books of the type "Book" or "Fresh arrival" only if 2 months has been passed since their arrival

ISSUE BOOK



RETURN BOOK

Book ID	DUE DATE
11	2022-12-29



RETURN

STATISTICS

- In statistics screen Admin can get detailed view of issue book and return book details
- We have also implemented the pictorial representation of issue and Return books using through
- We have implemented the Export To Excel for Analysing the book Details

```
public void ShowPieChart() {
    try {
        ObservableList<PieChart.Data> pieChartData = FXCollections.observableArrayList(
            new PieChart.Data("Issues", issueList.size()),
            new PieChart.Data("Returns", returnList.size()));

        pieChart.setData(pieChartData);

    } catch (Exception e) {
        e.printStackTrace();
    }
}

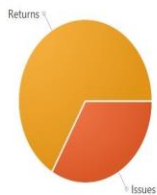
XSSFWorkbook wb = new XSSFWorkbook();
XSSFSheet sheet = wb.createSheet("Book Details");

XSSFRow header = sheet.createRow(0);
header.createCell(0).setCellValue("BookId");
header.createCell(1).setCellValue("BookName");
header.createCell(2).setCellValue("Author Name");
header.createCell(3).setCellValue("Arrival Date");
header.createCell(4).setCellValue("Type");
header.createCell(5).setCellValue("Status");

int index = 1;

while (rs.next()) {
    XSSFRow row = sheet.createRow(index);
    row.createCell(0).setCellValue(rs.getString("id"));
    row.createCell(1).setCellValue(rs.getString("name"));
    row.createCell(2).setCellValue(rs.getString("authorname"));
    row.createCell(3).setCellValue(rs.getString("arrivalDate"));
    row.createCell(4).setCellValue(rs.getString("type"));
    row.createCell(5).setCellValue(rs.getString("status"));
}
```

- □ X

Library Management Progress

● Issues ● Returns

 Books

BOOK

Select Book Item to Delete or Update or Search by ID

SEARCH

Type Of Book

 Book

Reference Book

 Fresh Arrival

DELETE

UPDATE

ADD BOOK

The background of the slide features a close-up photograph of a person's hand turning the pages of a thick, old book. The pages are aged and yellowed. A large, semi-transparent light blue rectangle is centered over the image, containing the word "DEMO" in a dark blue, serif font. Below the blue rectangle, a solid dark blue square is positioned on a light gray background.

DEMO

Big Learnings



Design and UML
Diagrams for Admin
and student



Collections ,Event driven
Programming, Integrating
with database



Time Management



Statistical Integration



Any questions?

Thank you for your time! 😊

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**