

Library Management System (LMS)

Bryant Le

Indiana University - Indianapolis

INFO- C 451: SYSTEM IMPLEMENTATION

Jafrina Jabin

May 5, 2025

TABLE OF CONTENTS

Problem Statement & System Requirements	3
Functional Requirements Specification	4
System Sequence Diagram	5
Activity Diagram	5
User Interface Specification	6
Implementation Planning	7
System Architecture And System Design	8
Algorithms And Data Model Implementation	9
User Interface Design And Implementation	11
Design Of Tests	12
Error Handling And Debugging	13
Collaboration And Code Integration	14
Performance Optimization Plan	14
Maintenance And Support Strategy	15
References	15

Problem Statement

A library management system is utilized to trace book transactions, which include book check-ins and check-outs, and current inventory. The current system keeps track of everything through physical media (e.g., pen and paper), which has resulted in inefficient check-ins and check-outs, librarians failing to keep locating and updating information due to lost paperwork and other acts of human error, and user inconvenience.

To rectify this issue, a digital library management system will provide an opportunity to make library operations more effective by reducing manual errors and improving the user experience. By automating how librarians manage the book database and the process of borrowing and returning books, employee physical workload will be reduced, and users will be able to locate books in real time and therefore increase system efficiency.

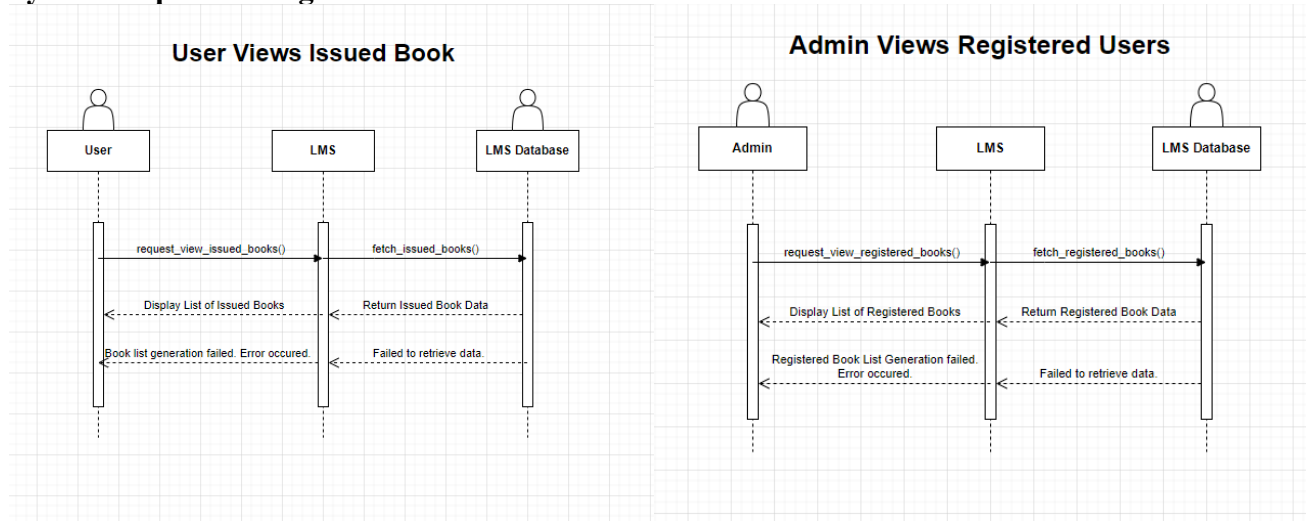
System Requirements:

- Software
 - Operating System
 - Windows OS
- Back-end
 - PHP
 - MySQL
- Front-end
 - HTML
- Framework
 - Bootstrap
- Hardware
 - Standard desktop for librarians and users
- Server Requirements
 - CPU: Quad-Core Processor (Intel or AMD)
 - RAM: 16 GB
 - Storage: 500 GB SSD (expandable)
- Network
 - 1 Gbps LAN Connection
 - Wi-Fi Access for staff and users
 - WP2 or WP3 Recommended

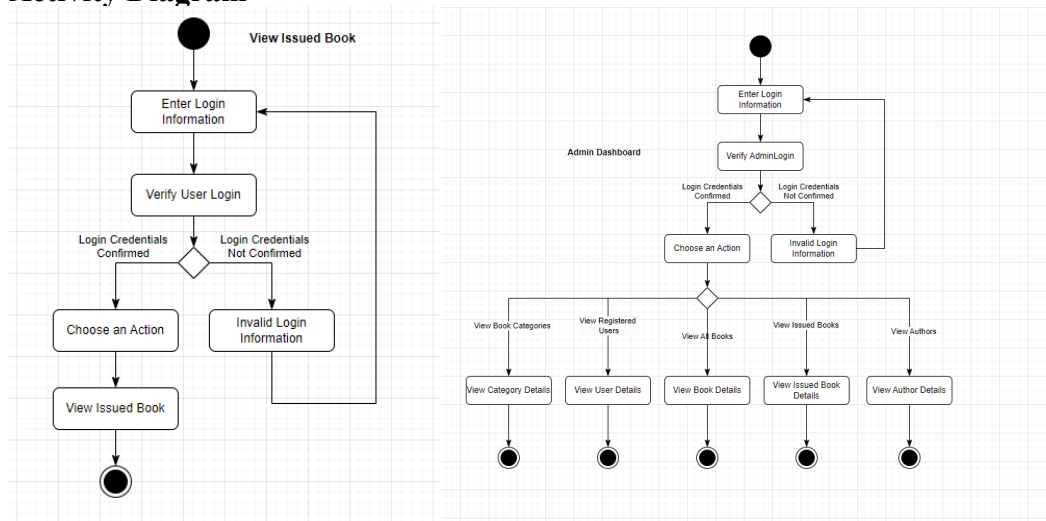
Functional Requirements Specification

Requirement	Description	Priority
User Registration	Users should be able to sign up with a unique email and password.	High
User Login	Users should be able to log in with valid credentials.	High
Password Reset	Users should be able to change/reset passwords securely.	Medium
Admin Authentication	Admins should have separate login credentials with higher privileges.	High
View Profile	Users should be able to view their personal information.	High
Edit Profile	Users should be able to update their details.	Medium
Admin User Management	Admins should be able to view, update, and delete user accounts.	High
Add Books	Admins should be able to add books with relevant details.	High
Update Books	Admins should be able to update book details.	High
Delete Books	Admins should be able to remove books.	Medium
View Books	Users should be able to browse all available books.	High
Search Books	Users should be able to search for books by title, author, or category.	High
Add Authors	Admins should be able to add new authors.	Medium
Edit Authors	Admins should be able to update the author's details.	Low
Add Book Categories	Admins should be able to create and manage book categories.	Medium
Issue Books to Students	Admins should be able to assign books to students.	High
Track Issued Books	Maintain a record of issued books, including issue date.	High
Return Books	Admins should be able to mark books as returned.	High
Check Book Availability	Users should be able to check if a book is available.	Medium
User View	Users should be able to see the books they have borrowed.	High
Admin View	Admins should be able to see all the books issued and the students who borrowed them.	High
Manual Logout	Users should be able to log out securely.	High
Access Control	Regular users should not be able to access admin features.	High

System Sequence Diagram



Activity Diagram



User Interface Specification

Library Management System

[User Login](#)
[Admin Login](#)
[Sign Up](#)

User Login Form

Email ID:

Password:

[Login](#)
[Sign Up](#)

Library Management System

[User Login](#)
[Admin Login](#)
[Sign Up](#)

User Registration Form

Full Name:

Email ID:

Password:

Mobile:

Address:

[Register](#)

Library Management System (LMS)

Welcome user
Email: user@gmail.com
My Profile
Logout

Book Issued
No of book issued: 2
[View Issued Books](#)

Library Management System (LMS)

Welcome user
Email: user@gmail.com
My Profile
Logout

Issued Books Details

Name	Author	Book Number
Data Structure	D S Gupta	5595
The Shining	Stephen King	1111

Library Management System (LMS)

Welcome admin
Email: admin@gmail.com
My Profile
Logout

[Dashboard](#)
[Books](#)
[Categories](#)
[Authors](#)
[Issue Book](#)

Registered User
No. total Users: 6
[View Registered Users](#)

Total Book
No of books available: 8
[View All Books](#)

Book Categories
No of Book's Categories: 9
[View Categories](#)

No. of Authors
No of Authors: 10
[View Authors](#)

Book Issued
No of book issued: 10
[View Issued Books](#)

Library Management System (LMS)

Welcome admin
Email: admin@gmail.com
My Profile
Logout

[Dashboard](#)
[Books](#)
[Categories](#)
[Authors](#)
[Issue Book](#)

Registered Books

Name	Author	Price	Number
Software Engineering	M D Gupta	75	4119
Data Structure	Charles Ruggel	350	5545
Harry Potter and the Philosopher's Stone	J.K. Rowling	450	1214
1984	George Orwell	170	3675
The Hobbit	J.R.R. Tolkien	500	7880
The Shining	Stephen King	600	1111
Murder on the Orient Express	Agatha Christie	300	2222
The Adventures of Tom Sawyer	Mark Twain	400	3333

Library Management System (LMS)

Welcome admin

Email: admin@gmail.com

No Profile * Logout

Dashboard

Books

Categories

Authors

Issue Book

Manage Books

Name	Author	Category	ISBN No.	Price	Action
Software Engineering	102	1	4518	3	Edit Delete
Data Structures	103	2	6541	300	Edit Delete
Harry Potter and the Philosopher's Stone	105	2	1234	450	Edit Delete
1984	106	1	5678	350	Edit Delete
The Hobbit	108	4	7890	500	Edit Delete
The Shining	109	4	1111	600	Edit Delete
Murder on the Orient Express	110	5	2222	300	Edit Delete
The Adventures of Tom Sawyer	107	5	3333	400	Edit Delete

Implementation Planning

Week 1 & 2 – Establish the essential system functionalities (e.g., book and user management), connect the front end to the back end, connect the back end to the database. Make a final determination for desktop or web-based design.

Week 3 & 4 – Build a system login, register librarian and administrator accounts, and plan database design. Plan UX design.

Week 5 & 7 – Implement user authentication, build a book database system, build borrowing and returning book features, and work on librarian dashboard for managing books and user tracking. Plan for user login. Continue working on UX design. Create test books, users, and librarians for database testing.

Week 8 – Test Features for Midterm, record a demonstration.

Week 9 – Implement author management and clean bugs. Improve security features.

Week 10 – Clean up the UI/UX Design

Week 11 – Have editing user details and change password features complete. Have input and form validation.

~~Week 12 – Work on notification feature.~~ **Failed to implement. Out of scope.**

Week 13 – Pretty up the CSS

Week 14 – Double check for bugs

Week 15/16 – Prepare for final presentation.

System Architecture And System Design

The Library Management System consists of three layers:

- Presentation Layer (Frontend)
- Application Logic Layer (PHP backend)
- Data Layer (MySQL Database)

Key Technologies

- Frontend: HTML5, CSS3, Bootstrap 5
- Backend: PHP 8+
- Database: MySQL
- Testing: PHPUnit (Unit + Integration)
- Development Tools: VS Code, XAMPP, Git

System Components

- issue_book.php
 - Form to issue books to users; validates and inserts data into issued_books
- manage_book.php
 - Displays books on a table with edit/delete options
- auth.php
 - Validates admin session and access rights
- add_author.php
 - Admin form to add a new author

Database Design

- books
 - book_id, book_name, book_no, book_price, author_id, cat_id
- authors
 - author_id, author_name
- category
 - cat_id, cat_name
- users
 - id, name, email, mobile, address
- issued_books
 - s_no, book_no, book_name, book_author, student_id, status

The LMS has input validation to ensure that valid data is being posted before processing. The navigation bar is consistent with all webpages that require it.

Algorithms And Data Model Implementation

Key Algorithms:

- Issue a Book to the User
 - Purpose: Prevent issuing the same book to the same user if the same book has not been returned,
- Duplicate Email and Phone Number Detection
 - Purpose: Ensure that no user or administrator can update their profile with an existing email/phone number used by another user.
 - From User:

```
if (mysqli_num_rows(result: $check_result) > 0) {
    echo "<script>alert('This book is already issued to this user.');

```

```
if (mysqli_num_rows(result: $email_check_result) > 0) {
    ?>
    <script type="text/javascript">
        alert("The email address is already taken. Please choose another.");
        window.location.href = "edit_profile.php";
    </script>
    <?php
} elseif (mysqli_num_rows(result: $mobile_check_result) > 0) {
    ?>
    <script type="text/javascript">
        alert("The mobile number is already taken. Please use another one.");
        window.location.href = "edit_profile.php";
    </script>
    <?php
} else {
    $update_query = "UPDATE users SET name='$name', email='$email', mobile='$mobile',
    address='$address' WHERE email='$_SESSION[email]';";
    $update_result = mysqli_query(mysql: $connection, query: $update_query);

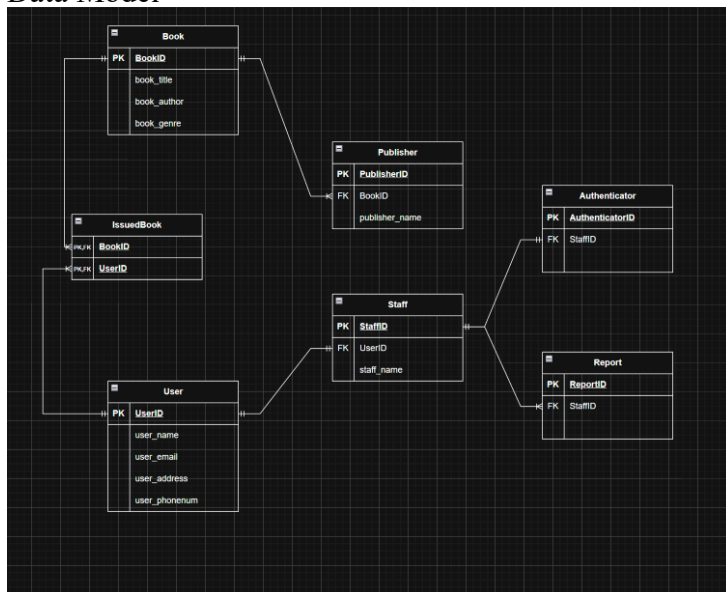
    $_SESSION['email'] = $email;
    $_SESSION['name'] = $name;
    ?>
    <script type="text/javascript">
        alert("Profile updated successfully.");
        window.location.href = "user_dashboard.php";
    </script>
    <?php
}
}
```

Database Implementation

- Each book belongs to a category and an author. (Many-to-One)
- Users can be issued multiple books, given that it isn't the same book. (Many-to-One)
- Each issued book has a status of 1 or 0. 1 for issued, and 0 for returned.

```
CREATE TABLE books (
  book_id INT PRIMARY KEY AUTO_INCREMENT,
  book_name VARCHAR(255),
  book_no VARCHAR(50) UNIQUE,
  author_id INT,
  cat_id INT,
  book_price DECIMAL(5,2),
  FOREIGN KEY (author_id) REFERENCES authors(author_id),
  FOREIGN KEY (cat_id) REFERENCES category(cat_id)
);
```

Data Model



User Interface Design And Implementation

Library Management System (LMS)
User Login
Admin Login
Signup

User Login

Email ID

Password

Login

Don't have an account? [Sign up here](#)

Library Management System (LMS)
Welcome: username
Email: user@gmail.com
My Profile
Logout

Book Issued

Number of books issued: 3

View Issued Books

Library Management System (LMS)
Welcome: admin
Email: admin@gmail.com
My Profile
Logout

Dashboard
Books
Category
Authors
Issue Book

Registered Users
Total Users: 2
View Users

Books
Available Books: 15
View Books

Categories
Total Categories: 5
View Categories

Authors
Total Authors: 7
View Authors

Books Issued
Books Issued: 4
View Issued Books

Library Management System (LMS)
Welcome: username
Email: user@gmail.com
My Profile
Logout

Issued Books Details

ISBN	Name	Author	Action
1004	The Shining	Stephen King	Return
1002	Harry Potter and the Sorcerer's Stone	J.K. Rowling	Return
1011	The Hobbit	J.R.R. Tolkien	Return

Library Management System (LMS)
Welcome: admin
Email: admin@gmail.com
My Profile
Logout

Dashboard
Books
Category
Authors
Issue Book

Registered Books

Book ID	Book Name	Book Author	Price	ISBN
1	1984	George Orwell	3.00	1001
2	Harry Potter and the Sorcerer's Stone	J.K. Rowling	3.25	1002
3	Murder on the Orient Express	Agatha Christie	2.75	1003
4	The Shining	Stephen King	3.10	1004
5	Pride and Prejudice	Jane Austen	2.80	1005
20	Animal Farm	George Orwell	2.50	1006
21	Harry Potter and the Chamber of Secrets	J.K. Rowling	3.30	1007
22	And Then There Were None	Agatha Christie	2.90	1008
23	IT	Stephen King	3.40	1009
24	Emma	Jane Austen	2.95	1010
25	The Hobbit	J.R.R. Tolkien	3.50	1011
26	The Time Machine	H.G. Wells	2.85	1012
27	The Murder of Roger Ackroyd	Agatha Christie	2.70	1013
28	Carrie	Stephen King	3.20	1014
29	Sense and Sensibility	Jane Austen	2.80	1015

Library Management System (LMS)
Welcome: admin
Email: admin@gmail.com
My Profile
Logout

Dashboard
Books
Category
Authors
Issue Book

Manage Books

Book ID	Book Name	Author	Category	ISBN	Price	Action
1	1984	George Orwell	Science Fiction	1001	3.00	Edit Delete
2	Harry Potter and the Sorcerer's Stone	J.K. Rowling	Fantasy	1002	3.25	Edit Delete
3	Murder on the Orient Express	Agatha Christie	Mystery	1003	2.75	Edit Delete
4	The Shining	Stephen King	Horror	1004	3.10	Edit Delete
5	Pride and Prejudice	Jane Austen	Classic Literature	1005	2.80	Edit Delete
20	Animal Farm	George Orwell	Science Fiction	1006	2.50	Edit Delete
21	Harry Potter and the Chamber of Secrets	J.K. Rowling	Fantasy	1007	3.30	Edit Delete
22	And Then There Were None	Agatha Christie	Mystery	1008	2.90	Edit Delete
23	IT	Stephen King	Horror	1009	3.40	Edit Delete
24	Emma	Jane Austen	Classic Literature	1010	2.95	Edit Delete
25	The Hobbit	J.R.R. Tolkien	Fantasy	1011	3.50	Edit Delete
26	The Time Machine	H.G. Wells	Science Fiction	1012	2.85	Edit Delete
27	The Murder of Roger Ackroyd	Agatha Christie	Mystery	1013	2.70	Edit Delete
28	Carrie	Stephen King	Horror	1014	3.20	Edit Delete
29	Sense and Sensibility	Jane Austen	Classic Literature	1015	2.80	Edit Delete

The image displays two side-by-side screenshots of a web application titled 'Library Management System (LMS)'. Both screenshots show a top navigation bar with the text 'Welcome: admin' and 'Email: admin@gmail.com', along with links for 'My Profile' and 'Logout'. Below the navigation bar is a breadcrumb trail: 'Dashboard > Books > Category > Authors > Issue Book'.

The left screenshot shows the 'Edit Book' form. It has a blue header bar with the text 'Edit Book'. Below this, there are five input fields: 'Book Number (ISBN)' with the value '1001', 'Book Name' with the value '1984', 'Select Author' with the value 'George Orwell', 'Select Category' with the value 'Science Fiction', and 'Book Price' with the value '3.00'. At the bottom right of the form is a blue button labeled 'Update Book'.

The right screenshot shows the 'Issue Book' form. It has a blue header bar with the text 'Issue Book'. Below this, there are three input fields: 'Select Book' with the value '-- Select Book --', 'Select User' with the value '-- Select User --', and 'Issue Date' with the value '2025-05-02'. At the bottom right of the form is a blue button labeled 'Issue Book'.

*View, Manage, and Edit Tables Following the Same UI conventions.

Design Of Tests

The objective of these tests is to ensure that book issuance, proper access control, and database operations work as intended. This includes testing the functions, interactions with components, and the UI works as intended.

Unit Testing:

- Test: isValidBookTitle with null
 - Expect: false
- Test: Issue book already issued
 - Expect: error message
- Test: Login with wrong password
 - Expect: blocked

Integration Testing:

- Tests interactions between the UI (Frontend) and the database (backend)

UI Testing:

- Tests user interface in the web browser

Tools Used:

- PHPUnit
 - For unit and integration tests
- Manual testing in the browser
 - For user interface
- phpMyAdmin
 - To verify database state

Error Handling And Debugging

The purpose of these actions is to ensure that the LMS remains stable, user-friendly, and maintainable.

Database Connection:

- Checks if system is connected to database. If not, show the following message:

```
$connection = mysqli_connect(hostname: "localhost", username: "root", password: "",
database: "lms");

if (!$connection) {
    die("Database connection failed: " . mysqli_connect_error());
}
```

User Input Validation

- Forms will validate the user input using frontend and backend integration. Prevents missing fields and ensures only information that matches the database will be implemented.

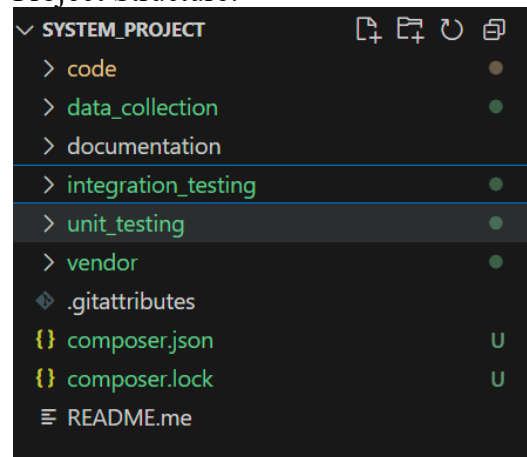
```
if ($password === $db_password) {
    $_SESSION['name'] = $name;
    $_SESSION['email'] = $email_db;
    $_SESSION['role'] = 'admin';
    header(header: "Location: admin_dashboard.php");
    exit();
} else {
    echo "<script>alert('Invalid email or password.');
```

- Only users can access the user pages, and admins can access the administrative pages.
- Alerts are implemented to instruct users to enter the correct information.
- Error messages are presented in a clear and concise manner.

Collaboration And Code Integration

This is an individual project, but room for collaboration is possible thanks to potential utilization of Git for version control. GitHub contains history of commits and branching to monitor changes by user and allows pull and merge requests.

Project Structure:



Code Integration:

File names and folder organization follow consistent naming conventions and supports scripts that test the system. README files provide clear directions on how to set up the project and run tests on the system's capabilities. This project is only available on GitHub, so all other links that do not directly access the repository are invalid and should not be downloaded.

Performance Optimization Plan

The LMS is designed to be developed for local environments, so ensuring that the system's performance is optimized is critical. This includes:

- Database optimization
- Clean and efficient code structure
 - Limited redundancies and has reusable logic
- Bootstrap Integration for fast page loading

This system is open for optimization to ensure that it follows the best industry standard and is maintained to allow future edits.

Maintenance And Support Strategy

To ensure that the LMS will be used for the future, the following practices will be used:

Bug Fixes and Updates:

- Listen to user feedback to identify issues and apply updates ASAP.
- Conduct reviews on the codebase to ensure that the best practices are being implemented.
 - If not, discuss potential refactoring or improvements.

Database Maintenance:

- Regularly backup data to reduce the risk of data loss
- Make sure the tables are optimized for fast queries.
- Clean unused and orphaned records to maintain data integrity.

Documentation and Change Tracking:

- Update the documentation if any changes are made.

User Support:

- To offer a reliable means to provide customer feedback
 - Helps gauge if system capabilities are adequate
 - If there are any features that need to be added
 - Any bugs to report.

Scalability:

- Check the system's performance regularly and be prepared to update

References

I used this tutorial to help me understand the scope of the project, but the tutorial itself was incomplete. I had to determine bugs and errors for the rest of the project, as well as Sprucing the UI design to match my tastes. The tutorial used a previous version of Bootstrap, so I had to update the code to match Bootstrap 5.

Tutorial: <https://www.geeksforgeeks.org/library-management-system/#er-model-of-library-management-system>