Group 1

Gonghweh Koo

Cynthia Reynolds

Scott Richardson

December 13, 2020

University of Maryland Global Campus

CMSC 495

Professor Kayed

***Library Management System***

***Final Report***

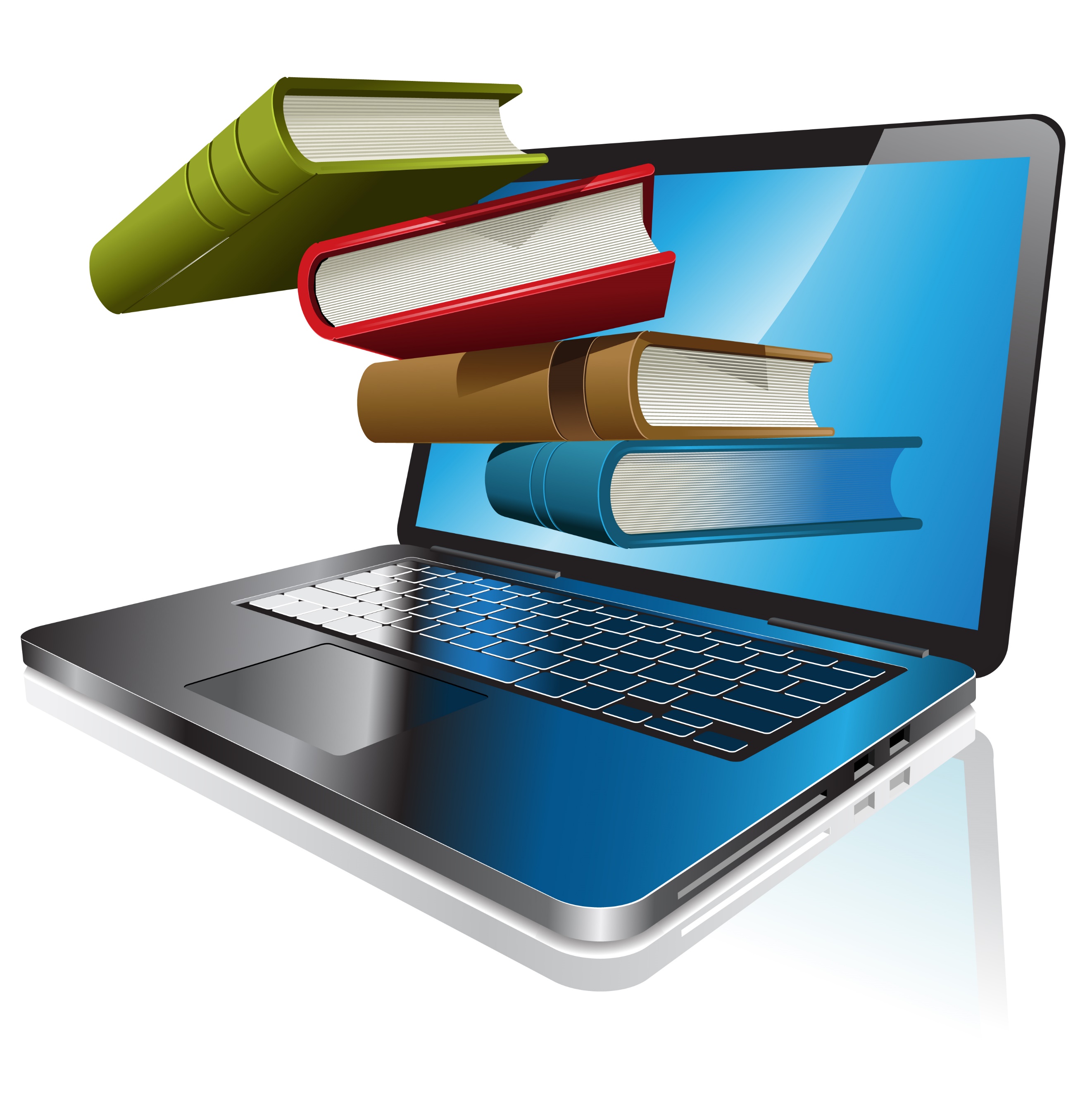


Table of Contents

1. Overview 4

1.1 Background 4

1.2 Glossary 4

1.3 Individual Contributions 4

2. Project Plan 5

2.1 Project Summary 5

2.2 Project Schedule 5

2.2.1 Work Breakdown Structure (WBS) 5

2.2.2 Project Timeline 7

2.3 Tools 7

3. Requirements Specification 7

3.1 Performance Requirements 8

3.2 Response Time 8

3.3 Security Requirements 8

3.4 Availability Requirements 8

3.5 Software Requirements 8

3.6 External Interface Requirements 8

3.6.1 User Interface 8

3.6.2 Hardware Interface 8

3.7 Functional Requirements 9

3.8 Entity Relationship Diagram 9

4. System Specification 10

4.1 Operating System 10

4.2 Software 10

4.3 Hardware 10

5. User’s Guide 10

6. Test Plan and Results 10

6.1 Test Plan Identifier 10

6.2 Introduction 10

6.2.1 Background 10

6.2.2 Purpose/Objectives 10

6.2.3 References 10

6.3 Test Items 11

6.3.1 GUI Test 11

6.3.2 Basic Function Test 11

6.3.3 Database Test 11

6.4 Format of Standard Test Case 11

6.4.1 Test Case Number 11

6.4.2 Requirement Number 11

6.4.3 Test Case Description 11

6.4.4 Expected Results 11

6.4.5 Actual Results 12

6.4.6 Pass/Fail 12

6.5 Features to be Tested 12

6.5.1 GUI test 12

6.5.2 Database Test 12

6.5.3 Basic Function Test 12

6.5.4 Check Connectivity of the System. 12

6.6 Features not to be Tested 12

6.6.1 Maximum simultaneous online users. 12

6.6.2 Licensing requirements. 12

6.7 Approach 13

6.7.1 Unit Testing (class testing) 13

6.7.2 Integrity Testing (thread-based testing) 13

6.7.3 Validation Testing 13

6.8 Item Pass Criteria 13

6.8.1 GUI Test 13

6.8.2 Database Test 13

6.8.3 Basic Function Test 13

6.9 Suspension Criteria and Resumption Requirements 15

6.10 Test Deliverables 15

6.11 Environmental Needs 15

6.11.1 Hardware 15

6.11.2 Software 15

6.12 Scenarios 16

6.12.1 Use Case: Login to LMS 16

6.12.2 Use Case: Register new library patron 16

6.12.3 Use Case: Edit library patron record 17

6.12.4 Use Case: Search for and reserve a book 17

6.12.5 Use Case: View late fees 17

6.12.6 Use Case: View assets currently checked out 18

6.13 UAT Test Cases and Results 18

6.13.1 Login (all users) 18

6.13.2 Register new library patron (librarian) 19

6.13.3 Edit library patron record (library patron) 19

6.13.4 Search for an item (library patron) 19

6.13.5 View late fees (patron) 20

6.13.6 View assets currently checked out 20

6.14 Test Results 20

7. Design 20

7.1 Background 20

7.2 Overall Approach 20

7.3 System Modules 21

7.3.1 User Module 21

7.3.2 Librarian Module 21

7.4 Implementation Design 21

7.4.1 High Level Class Diagram 21

7.4.2 Entity Class Diagram 22

7.4.3 Component Diagram 23

7.5 User Interface Design Mockup 24

7.5.1 User Login 24

7.5.2 Item Search 25

7.5.3 Search Result 25

7.5.4 Advanced Search 26

8. Development History 26

9. Conclusions 27

9.1 Lessons Learned 27

9.2 Design Strengths 27

9.3 Limitations 27

9.4 Suggestions for Future Improvement 28

Table of Figures

[Figure 2‑1. Project Timeline 7](#_Toc58595382)

[Figure 3‑1. Entity Relationship Diagram (ERD) 9](#_Toc58595383)

[Figure 7‑1. High Level Class Diagram 22](#_Toc58595384)

[Figure 7‑2. Entity Class Diagram 23](#_Toc58595385)

[Figure 7‑3. Component Diagram 24](#_Toc58595386)

[Figure 7‑4. User Login Screen 24](#_Toc58595387)

[Figure 7‑5. Item Search Screen 25](#_Toc58595388)

[Figure 7‑6. Search Result Screen 25](#_Toc58595389)

[Figure 7‑7. Advanced Search Screen 26](#_Toc58595390)

[Figure A‑1. Registration (Add User) Screen 4](#_Toc58595391)

[Figure A‑2. Login Screen 4](#_Toc58595392)

[Figure A‑3. Search Screen 5](#_Toc58595393)

[Figure A‑4. User Inactivity Logout Dialog 7](#_Toc58595394)

Table of Tables

[Table 1‑1. Acronyms and Definitions 4](#_Toc58595395)

[Table 1‑2. Individual Roles and Contributions 5](#_Toc58595396)

[Table 2‑1. LMS Work Breakdown Structure 6](#_Toc58595397)

[Table 6‑1. Suspension Criteria/Resumption Requirements 15](#_Toc58595398)

[Table 6‑2. Use Case: Login 16](#_Toc58595399)

[Table 6‑3. Use Case: Register new library patron 16](#_Toc58595400)

[Table 6‑4. Use Case: Edit library patron record 17](#_Toc58595401)

[Table 6‑5 Use Case: Search for and reserve a book 17](#_Toc58595402)

[Table 6‑6. Use Case: View late fees 17](#_Toc58595403)

[Table 6‑7. Use Case: View assets currently checked out 18](#_Toc58595404)

[Table 6‑8. Login Test Case Test Table 18](#_Toc58595405)

[Table 6‑9. Register new library patron Test Table 19](#_Toc58595406)

[Table 6‑10. Edit library patron record Test Table 19](#_Toc58595407)

[Table 6‑11. Search for an item Test Table 19](#_Toc58595408)

[Table 6‑12. View Late Fees Test Table 20](#_Toc58595409)

[Table 6‑13. View assets checked out Test Table 20](#_Toc58595410)

[Table 8‑1. Project Development History 26](#_Toc58595411)

[Table A‑1. Acronyms and Definitions 7](#_Toc58595412)

# Overview

## Background

This report describes the analysis, design, and development of Group 1’s software project entitled “Library Management System”. It includes the full suite of artifacts developed and delivered in support of the project. Report components are organized as follows:

Section 1 describes the project purpose, objectives and...

Section 2 describes the project plan

Section 3 describes the requirements specification

Section 4 describes the system specification

Section 5 describes the [User Guide]

Section 6 describes the all aspects of system testing

Section 7 describes the software design

Section 8 describes the development history

Section 9 describes Lessons Learned, potential future enhancements, and conclusion

Appendix A provides the system User’s Guide

## Glossary

Acronyms and abbreviations used throughout this report are identified and defined in Table **1‑1**.

Table ‑. Acronyms and Definitions

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| ERD | Entity Relationship Diagram |
| GB | Gigabyte |
| GUI | Graphical User Interface |
| ISBN | International Standard Book Number |
| LMS | Library Management System |
| PC | Personal Computer |
| RAM | Random Access Memory |
| SDLC | Software Development Life Cycle |
| STS | Software Test Specification |
| UAT | User Acceptance Testing |
| UIE | User Interface Engine |
| WBS | Work Breakdown Structure |

## Individual Contributions

This project report reflects the collective effort of Group 1 members to apply software engineering methodologies and practices in developing a [software project]. For a list of Group 1 members, along with their roles and contributions, see Table 1‑2**.**

Table ‑. Individual Roles and Contributions

|  |  |  |
| --- | --- | --- |
| **Team Member** | **Roles** | **Individual Contributions** |
| Gonghweh Koo | Design Lead  Coding Lead for Phase 3 | Created Project Design Plan  Collaborated on Phase 2 Source Code  Created Phase 3 Source Code/Report  Completed Peer Review 2 |
| Cynthia Reynolds | Requirements Lead  Documentation Lead | Initiated weekly Google meetings in regards to weekly class assignments  Created the Project Plan and Requirement Specification  Created and maintained the User’s Guide  Created the Final Report  Completed Peer Review 3 |
| Scott Richards | Testing Lead  Coding Lead for Phase 1  Coding Lead for Phase 2 | Designed/Executed Test Cases  Provided completed STS for Final Report  Designed the majority of the classes and modules in the suite  Created Phase 1 and Phase 2 Source Code/Report  Completed Peer Review 1 |

# Project Plan

## Project Summary

The purpose of this project is to develop an electronic Library Management System (LMS) that will enable the automated management of library assets (e.g., books, ebooks, movies). The users of the LMS will be librarians and library patrons. Objectives of the LMS include efficient and accurate registration of library patrons, and management of library assets. The LMS will support an array of library operations from establishing new members to discovery, selection, inventory and circulation management of library assets.

## Project Schedule

### Work Breakdown Structure (WBS)

The WBS in Table 2‑1 defines and organizes the scope of the entire LMS project. The project deliverables are broken down into detailed and quantifiable tasks which are then assigned to the appropriate team member.

Table ‑. LMS Work Breakdown Structure

| **WBS** | **Task** | **Start Date** | **End Date** | **Duration (In Days)** | **Notes** | **Responsible Team Member** |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Project: Library Management System** | 21-Oct | 15-Dec | 56 |  |  |
| **1.0** | **Form Team** | 21-Oct | 27-Oct | 7 |  | All |
| 1.1 | Form group and make introductions | 27-Oct | 27-Oct | 1 |  | All |
| 1.2 | Create group communication page, file storage | 27-Oct | 28-Oct | 2 | GitHub, email, Google Meet | Scott |
| 1.3 | Submit group members | 27-Oct | 27-Oct | 1 |  | Gonghweh |
| **2.0** | **Project Plan** | 29-Oct | 4-Nov | 7 |  | Cynthia |
| 2.1 | Define Requirements | 29-Oct | 3-Nov | 6 |  |  |
| 2.2 | Submit Requirements to GitHub | 3-Nov | 3-Nov | 1 |  |  |
| 2.3 | Complete and submit Project Plan to Assignment folder | 4-Nov | 4-Nov | 1 |  |  |
| **3.0** | **Test Plan** | 5-Nov | 11-Nov | 7 |  | Scott |
| 3.1 | Research and collaborate on test requirements, test data, and test plan | 5-Nov | 7-Nov | 3 |  |  |
| 3.2 | Submit outline to GitHub. | 5-Nov | 7-Nov | 3 |  |  |
| 3.3 | Work on Test Plan | 8-Nov | 11-Nov | 4 |  |  |
| 3.4 | Complete and submit Test Plan to Assignment folder | 11-Nov | 11-Nov | 1 |  |  |
| **4.0** | **Design** | 12-Nov | 18-Nov | 7 |  | Gonghweh |
| 4.1 | Research and collaborate on application design | 12-Nov | 14-Nov | 3 |  |  |
| 4.2 | Finalize design decisions | 12-Nov | 15-Nov | 4 |  |  |
| 4.3 | Submit outline to GitHub | 12-Nov | 14-Nov | 3 |  |  |
| 4.4 | Work on Project Design | 15-Nov | 18-Nov | 4 |  |  |
| **4.5** | Complete and submit Project Design to Assignment folder | 18-Nov | 18-Nov | 1 |  |  |
| **5.0** | **Coding/Implementation** | 19-Nov | 15-Dec | 28 |  | Scott/Gonghweh |
| 5.1 | Phase 1 Source | 19-Nov | 25-Nov | 7 | Will follow the design phase |  |
| 5.1.1 | Complete and submit Phase 1 Source code to Assignment folder | 25-Nov | 25-Nov | 1 |  |  |
| 5.2 | Phase 2 Source | 26-Nov | 2-Dec | 7 | Will follow the design phase |  |
| 5.2.1 | Complete and submit Phase 2 Source code to Assignment folder | 2-Dec | 2-Dec | 1 |  |  |
| 5.3 | Phase 3 Source | 3-Dec | 9-Dec | 7 | Will follow the design phase |  |
| 5.3.1 | Complete and submit Phase 3 Source code to Assignment folder | 9-Dec | 9-Dec | 1 |  |  |
| **6.0** | **Testing** | 19-Nov | 13-Dec |  |  | All |
| 6.1 | Perform testing |  |  |  | Will be performed following completion of the test plan and in coordination with the source code |  |
| **7.0** | **User’s Guide** | 11-Nov | 8-Dec | 28 |  | Cynthia |
| 7.1 | Submit outline to GitHub | 11-Nov | 13-Nov | 3 |  |  |
| 7.2 | Submit preliminary user's guide | 14-Nov | 17-Nov | 4 |  |  |
| 7.3 | Complete and submit User’s Guide to GitHub | 18-Nov | 8-Dec | 21 |  |  |
| **8.0** | **Submit final project deliverables to Assignment folder** | **13-Dec** | **13-Dec** | **1** |  | Cynthia |

### Project Timeline

Figure 2‑1**.** displays the project timeline in a Gantt chart. It shows the dependencies among each task as well as the project milestones with their due dates.

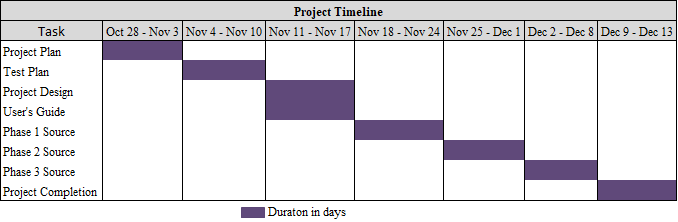


Figure ‑. Project Timeline

## Tools

* Repository hosting service: GitHub for storing and collaborating files and documents.
* Communication tools: Google Meet for weekly check-ins and collaboration; ad hoc email messaging.
* Development tools: Netbeans; Java 8 JDK

# Requirements Specification

The Requirements Specification defines the complete set of software requirements for the LMS. Both functional and non-functional (e.g., availability, performance) requirements are defined.

The Library Management System will consist of a centralized repository for detailed library asset information (e.g., book title, author, ISBN, year of publication) and user data, as well as a graphical user interface (GUI), which will provide an intuitive experience for library patrons and librarians. The LMS menu will be designed to show details of library patrons and assets, including issued assets. Library patrons will use the system to search, reserve, and borrow library assets, and track due dates for items they have borrowed from the library, all via the click of the mouse and keyboard. The system will send notifications via email to library patrons when reserved assets become available, as well as when assets are not returned by the due date.

## Performance Requirements

The system shall generate error messages when the user attempts to enter invalid data.

## Response Time

The system shall provide a response to a submitted request within three seconds. The information in the database will be refreshed immediately.

## Security Requirements

* The system shall ensure only users with an active account can log in to the system according to the access levels granted to them.
* The system shall ensure only users with an active account can log in to the system according to the access levels granted to them.
* The system shall automatically logout a user after five minutes of inactivity.

## Availability Requirements

The system shall be available 24/7, with the exception of semi-annual maintenance.

## Software Requirements

* Java 8 JDK or JRE installed.

## External Interface Requirements

### User Interface

The system shall provide a user-friendly GUI allowing the user to move from one component to the next.

### Hardware Interface

* Input: Keyboard and Mouse
* Output: Monitor

## Functional Requirements

* The system shall allow a user to register an account.
* The system shall allow a user to retrieve a forgotten password.
* The system shall allow a registered user to search for library assets.
* The system shall allow a registered user to reserve library assets.
* The system shall allow a registered user to check out library assets.
* The system shall allow a registered user to track issued assets.
* The system shall notify a registered user when issued assets are overdue.
* The system shall notify a registered user when reserved assets become available.
* The system shall allow the librarian to add a new patron and all their information to the system using the Add User functionality.
* The system shall allow the librarian to remove a patron from the system using the Remove User functionality.
* The system shall allow the librarian to add new items (e.g., books, DVDs) to the system using the Add Item functionality.
* The system shall allow the librarian to remove items (e.g., books, DVDs) from the system using the Remove Item functionality.
* The system shall enable the librarian to search for an item requested by a patron and check it out to them using Check-out functionality.
* The system shall allow the librarian to return an item from a patron and check it back into the system using the Check-out functionality (deselect the Check-out button).

## Entity Relationship Diagram

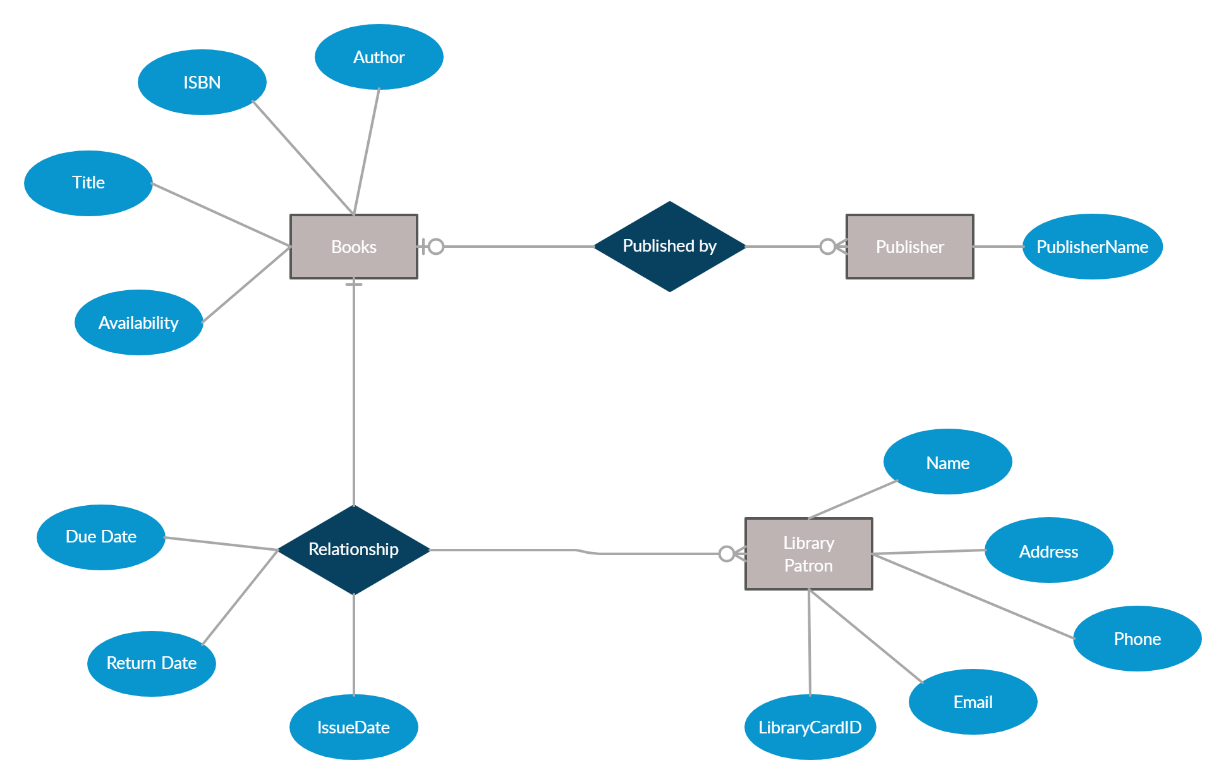


Figure ‑. Entity Relationship Diagram (ERD)

# System Specification

## Operating System

* Windows 7 or later, 64-bit

## Software

* Java 8 JDK or JRE installed

## Hardware

* Personal Computer with at least 2 GB of RAM and Internet Connectivity

# User’s Guide

See [Appendix A](#AppendixA)

# Test Plan and Results

## Test Plan Identifier

* LMS\_STS\_01

## Introduction

### Background

The Library Management System is an online application for assisting library patrons in accessing the assets in the library. The system will provide a basic set of features to add/update library patrons, request/reserve library assets, search for library assets, and manage check-in/checkout processes. The test team will test the system based on the requirement specification.

### Purpose/Objectives

The purpose of this test plan is to provide specific details on how testing processes will be performed for the LMS. Objectives of the test plan include:

* Ensuring that the functionality being delivered works as expected.
* Ensuring the functionality required for library patron scenarios has been delivered.
* Ensuring the delivered functionality works to specification.

### References

* Group 1 Plans and Specifications\_v1.0.
* Group 1 Project Plan – Library Management System\_v1.1.
* “IEEE Std 829-1998”, IEEE Standard for Software Test Documentation. IEEE, 1998 Edition, 1-64. <https://standards.ieee.org/standard/829-1998.html>.
* “IEEE Std 1016-1998”, IEEE Recommended Practice for Software Design Descriptions. IEEE, 1998 Edition, 1-28. <https://standards.ieee.org/standard/1016-1998.html>.
* “IEEE Std 1058-1998”, IEEE Standard for Software Project Management. IEEE, 1998 Edition, 1-28. <https://standards.ieee.org/standard/1058-1998.html>.
* “IEEE Std 1233-1998”, IEEE Guide for Developing System Requirements Speciﬁcations. IEEE, 1998 Edition, 1-36. <https://standards.ieee.org/standard/1233-1998.html>.

## Test Items

### GUI Test

Testing the LMS’ graphical user interface confirming proper functionality based on the specifications.

### Basic Function Test

Performing basic function testing will prove the software is performing the defined functions as expected by the library patron.

### Database Test

Performing database testing will ensure data mapping and data integrity.

## Format of Standard Test Case

### Test Case Number

Unique test case identification number

### Requirement Number

Refers back to a specific section within the requirements documentation.

### Test Case Description

A description of the test to be executed.

### Expected Results

The expected results of each test executed.

### Actual Results

The actual results of each test executed.

### Pass/Fail

Test result indication of whether the test was successful (pass) or unsuccessful (fail.)

## Features to be Tested

### GUI test

System should provide a GUI for library patrons to interface with the frontend library database.

### Database Test

#### Basic operations

Add/update/delete/query database.

#### Advanced operations

### Basic Function Test

#### Add a new library patron.

#### Update library patron personal information.

#### Borrow book.

#### Return book.

#### Search for book.

#### Check-in book.

#### Check-out book.

#### Check late fees.

### Check Connectivity of the System.

## Features not to be Tested

### Maximum simultaneous online users.

### Licensing requirements.

Reason: Since they are free software, they need not be tested.

## Approach

### Unit Testing (class testing)

### Integrity Testing (thread-based testing)

### Validation Testing

## Item Pass Criteria

### GUI Test

Pass Criteria: The application login screen will appear.

### Database Test

Pass criteria: Results of all basic and advanced operations are normal (refer to section 7.5.2).

### Basic Function Test

#### Add a new library patron

Pass criteria: Each new library patron should have the following attributes: Name, Address, Email, Phone number, and Library card ID.

#### Update library patron information

Pass criteria: The record would be selected using the library patrons’ library card ID.

Updates can be made to name, address, email, phone number, but not the library card ID.

The record can be deleted if there are no books or DVDs issued by library patron.

The updated values would be reflected if the same library patron’s library card ID is called for.

#### Request a book

Pass criteria: Each book shall have following attributes: Call Number, ISBN, Title, Author name, Publication year. The retrieved book information should contain the four attributes.

#### Return a book

Pass criteria: The book item can be returned using the call number

The book can be deleted only if no user has issued it.

The updated values would be reflected if the same call number is called for.

If book were deleted, it would not appear in further search queries.

#### Search for book

Pass criteria: The product shall let the library patron query books’ detail information by their ISBN, Title, or Author.

The search results would produce a list of books, which match the search parameters with the following Details: ISBN number, Title, Author

The display would also provide the number of copies which is available for issue

The display shall provide a means to select one or more rows to a user-list

The search display will be restricted to 20 results per page and there would be a means to navigate from sets of search results.

The library patron can perform multiple searches before finally selecting a set of books for check-in or check-out. These should be stored across searches.

A book may have more than one copy. But every copy with the same ISBN number should have same detail information.

#### Check-in book

Pass criteria: Librarian can check in a book using its call number.

The check-in can be initiated from a previous search operation where library patron has selected a set of books.

The return date would automatically reflect the current system date.

Any late fees would be computed as difference between due date and return date at rate of 10 cents a day.

A book, which has been checked in once, should not be checked in again

#### Check-out book

Pass criteria: Librarians can check out a book using its call number.

The checkout can be initiated from a previous search operation where library patron has selected a set of books.

The library card ID who is issuing the book would be entered.

The issue date would automatically reflect the current system date.

The due date would automatically be stamped as 10 days from current date.

A book, which has been checked out once, should not be checked out again.

A library patron who has books overdue should not be allowed to check out any books.

The maximum number of books that can be issued to a customer would be 10. The system should not allow checkout of books beyond this limit.

#### View book detail

Pass criteria: This view would display details about a selected book from search operation.

The details to be displayed are: Call number, IBN, Title, Author, Publication year, Issue status (In library or checked out), If book is checked out it would display, User ID & Name, Checkout date, Due date.

Books checked in should not display user summary.

Books checked out should display correct library patron details.

#### View library patron detail

Pass criteria: Librarians can select a user record for detailed view.

The detail view should show:

1. User name, library card ID, Address, Email & Phone number.
2. The books issued by user with issue date, due date, call number, title.
3. Late fees & Fines summary and total.

The display should match existing library patron profile.

The books checked out should have their statuses marked.

The book search query should show the library card ID correctly.

## Suspension Criteria and Resumption Requirements

Table ‑. Suspension Criteria/Resumption Requirements

|  |  |
| --- | --- |
| **Suspend Testing** | **Resume Testing** |
| Hardware and/or software is not available | Hardware and/or software resources become available |
| The output is not the same as the expected result | Output meets the expected results |
| Any critical defect impacting the testing progress | Defect has been identified and resolved |

## Test Deliverables

* Test Plan – defines what is required to be performed in UAT.
* Designs – Criteria for UAT Acceptance.
* Test Cases – The input values and expected output values from the tests.
* The test data – The data used for testing the LMS functionality.
* Software Test Specifications (STS) – Document that includes the plan for testing the software; specifies test cases and procedures to ensure the software fulfills the requirements as defined in the requirements specification.
* Screen Prototypes – Draft version of the LMS.

## Environmental Needs

### Hardware

The LMS is being developed via the Java programming language. The system testing will be performed on a personal computer (PC) with at least 2 GB of RAM and Internet connectivity.

### Software

Testing of the LMS will be performed on a Windows 7 or later, 64-bit operating system.

## Scenarios

### Use Case: Login to LMS

Table ‑. Use Case: Login

|  |  |
| --- | --- |
| **Use Case Name** | **Login to LMS** |
| Primary Actor | Library Patron |
| Uses | Login Account |
| Pre-conditions | The Library Patron has a library account |
| Post-conditions | The library patron is logged in and on the main page of the LMS |
| Main Scenario | 1. Library Patron enters either their username or library card ID and password in the corresponding text fields. 2. If username/library card ID and/or password are valid, library patron is logged in. 3. If username/library card ID and/or password are invalid, the system will display an invalid entry message. Request them to try again. |

### Use Case: Register new library patron

Table ‑. Use Case: Register new library patron

| **Use Case Name** | **New Library Patron submission** |
| --- | --- |
| Primary Actor | Non-Library Patrons |
| Uses | A Non-Library Patron wants to register for an account in the LMS. |
| Pre-conditions | The person doesn't already have a library account established. |
| Post-conditions | An LMS account has been created. |
| Main Scenario | 1. Non-Library Patron visits the website to access the online library. 2. He/she signs up for a LMS account by providing personal information and establishing a valid password. 3. He/she is now registered as a Library Patron and provided a library card ID. |
| Error Conditions | 1. Non-Library Patron visits the website to access the online library. 2. He/she attempts to sign up for a LMS account. 3. He/she is unable to successfully create a LMS account and receives an error. 4. He/she will call the library for assistance with creating a LMS account. |

### Use Case: Edit library patron record

Table ‑. Use Case: Edit library patron record

|  |  |
| --- | --- |
| **Use Case Name** | **Edit Library Patrons Record** |
| Primary Actor | Library Patron |
| Uses | Library Patron wants to edit their personal information on their LMS account. |
| Pre-conditions | The Library Patron is logged into the LMS. |
| Post-conditions | The Library Patron's personal record has been changed in the LMS. |
| Main Scenario | 1. Selects the menu to change their personal information. 2. Enters the new information in the category fields. 3. Clicks the Save button. |

### Use Case: Search for and reserve a book

Table 6‑5 Use Case: Search for and reserve a book

| **Use Case Name** | **Search for and Reserve book** |
| --- | --- |
| Primary Actor | Library Patron |
| Uses | Reserve a book(s) from the LMS. |
| Pre-conditions | 1. The Library Patron is logged into the LMS and has selected a book to borrow. 2. The book is available in the library database. |
| Post-conditions | Library Patron has reserved a book. |
| Main Scenario | 1. Library Patron enters login information and signs into the LMS. 2. Library Patron searches for a particular book. 3. Library Patron has located the book. 4. Library Patron has clicked the Reservation button. 5. Book has been reserved. |

### Use Case: View late fees

Table ‑. Use Case: View late fees

|  |  |
| --- | --- |
| **Use Case Name** | **View late fees** |
| Primary Actor | Library Patron |
| Uses | Library patron can view all late fees charged to them. |
| Pre-conditions | None. |
| Post-conditions | The Library Patron will be shown all assets checked out. |
| Main Scenario | 1. Library Patron enters login information and signs into the LMS. 2. Library Patron is shown all late fees on their account. If no late fees, the system will display a message stating that. |

### Use Case: View assets currently checked out

Table ‑. Use Case: View assets currently checked out

|  |  |
| --- | --- |
| **Use Case Name** | **View assets currently checked out** |
| Primary Actor | Library Patron |
| Uses | Library patron can view all the assets they currently have checked out of the LMS. |
| Pre-conditions | None. |
| Post-conditions | The Library Patron will be shown all assets checked out. |
| Main Scenario | 1. Library Patron enters login information and signs into the LMS. 2. Library Patron is shown all the assets that are currently signed out to them. If no assets are checked out, the system will display a message stating that. |

## UAT Test Cases and Results

The tables below represent the test case and results for each of the modules in the LMS.

### Login (all users)

Table ‑. Login Test Case Test Table



### Register new library patron (librarian)

Table ‑. Register new library patron Test Table



### Edit library patron record (library patron)

Table ‑. Edit library patron record Test Table



### Search for an item (library patron)

Table ‑. Search for an item Test Table



### View late fees (library patron)

Table ‑. View Late Fees Test Table



### View assets currently checked out (library patron)

Table ‑. View assets checked out Test Table



## Test Results

While there were a few bugs discovered during the unit, integrity, and validation testing cycle, they have all resolved, thereby satisfying all requirements.

# Design

## Background

The Library Management System (LMS) is an online application for assisting library users in accessing the assets in the library. The system provides a basic set of features to register new users, update personal information, search for library assets (e.g., books, DVDs), request and reserve library assets, and manage check-in and checkout processes.

## Overall Approach

The LMS will be using version 8 of the Java programming language for implementation. Because it will be using the latest version of a high-level language, the implementation will take advantage of the latest built-in APIs for input/output and data structures. The architecture of class definition will be based on object-oriented concepts and use design patterns such as data access objects and separation of concerns. The user interface will use Swing components in order to build a desktop GUI application. As part of the design process, we leveraged our familiarity with the local public library services and used their websites as a resource to determine the use cases for our application.

## System Modules

The LMS is divided into the following modules.

### User Module

#### Login

#### Edit Users Record

#### Search for an Item

#### Request an Item

#### View Items Checked Out

#### View Late Fees

### Librarian Module

#### Login

#### Register New User

#### Search for an Item

#### Checkout an Item

#### Check in an Item

#### View Items Checked Out

#### View Late Fees

## Implementation Design

### High Level Class Diagram

The class diagram (Figure 7‑1) demonstrates the main classes and interfaces that are used to implement the application. The ILibraryItem interface will have implementing classes that represent the main entities, such as books, movies, and so on, that are available at the library.

Graphical user interface, diagram

Description automatically generated

Figure ‑. High Level Class Diagram

### Entity Class Diagram

The class diagram (Figure 7‑2) shows in detail the types of entities this application will support. Each of the entities implement the ILibraryItem interface which defines a common set of properties applicable to each item that can be borrowed from the library.

Graphical user interface, diagram

Description automatically generated

Figure ‑. Entity Class Diagram

### Component Diagram

The component diagram (Figure 7‑3) shows the execution flow between different components in the application. The classes have been designed to create separation of concerns between the user interface, the service layer that provides most of the business logic, and the data storage layer. In this application, we will be using a file to store the different data entities used in the application.

Diagram

Description automatically generated

Figure ‑. Component Diagram

## User Interface Design Mockup

### User Login

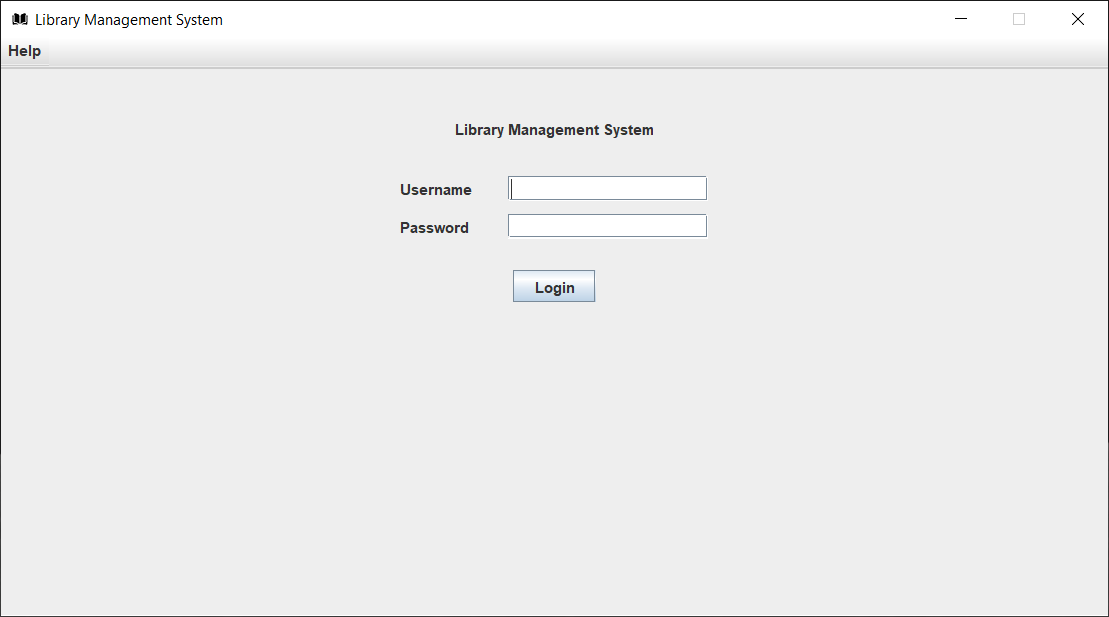


Figure 7-4. User Login Screen

### Item Search

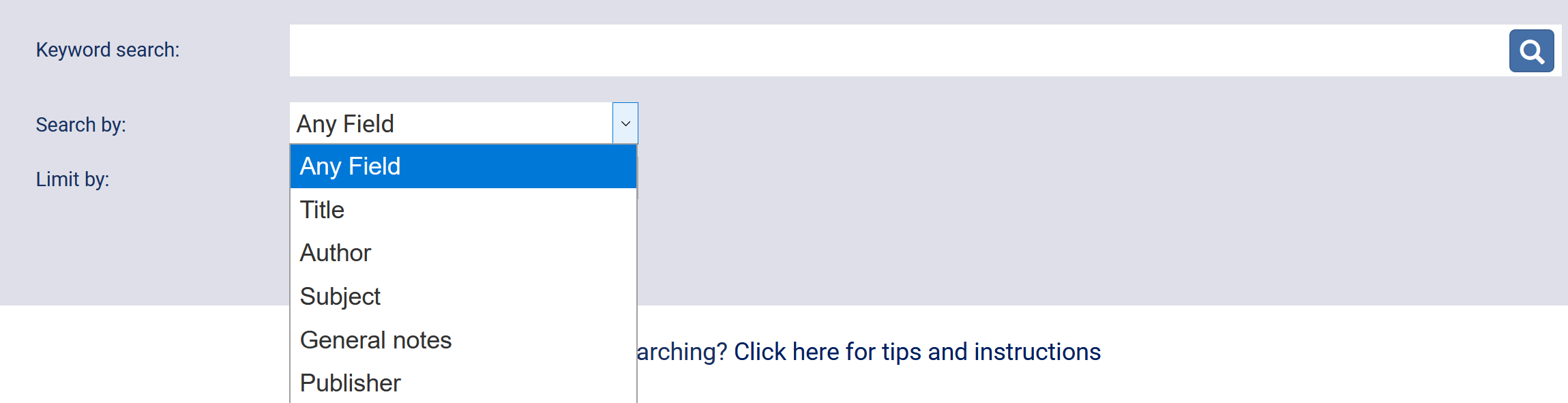


Figure ‑5. Item Search Screen

### Search Result

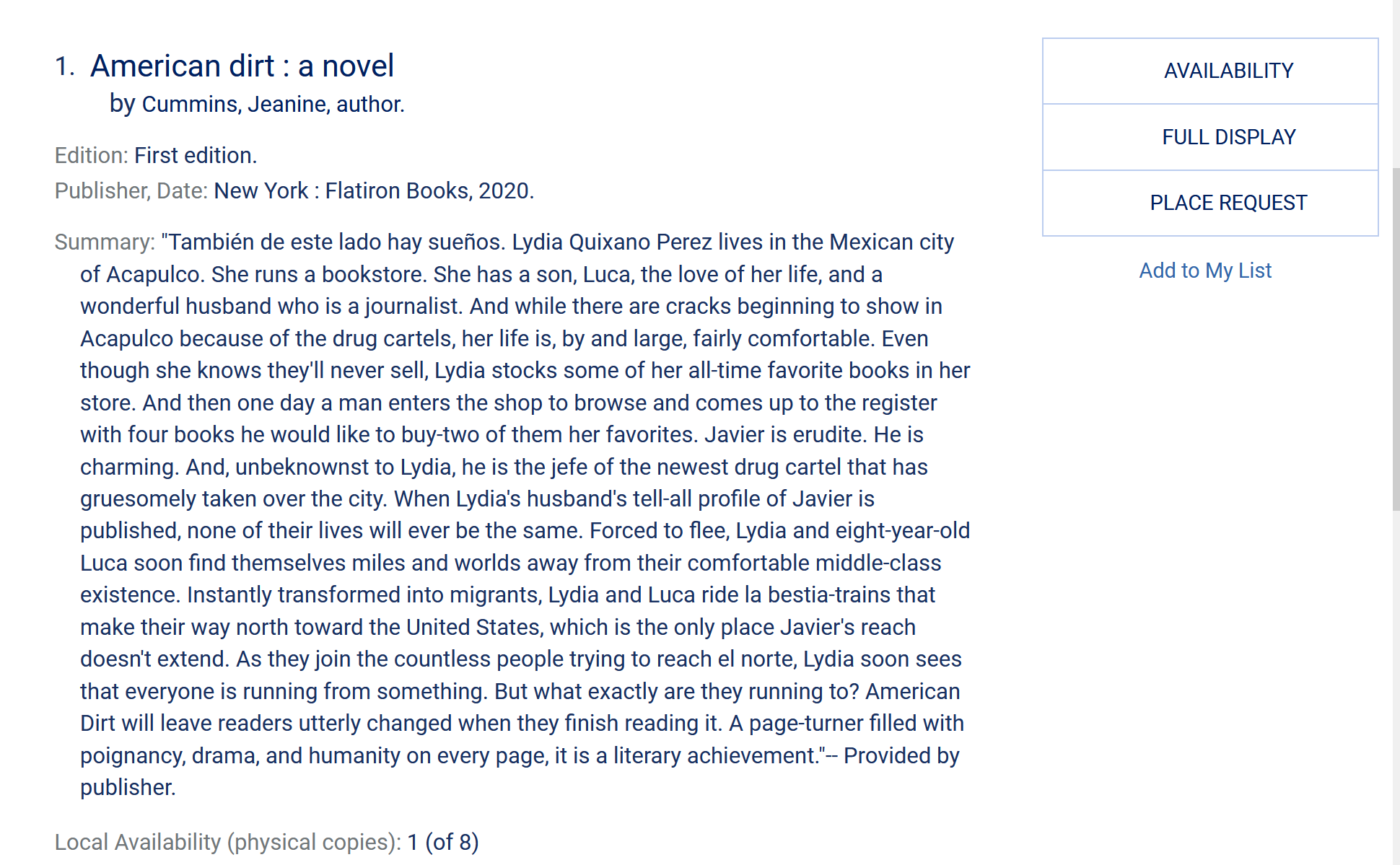


Figure ‑6. Search Result Screen

### Advanced Search

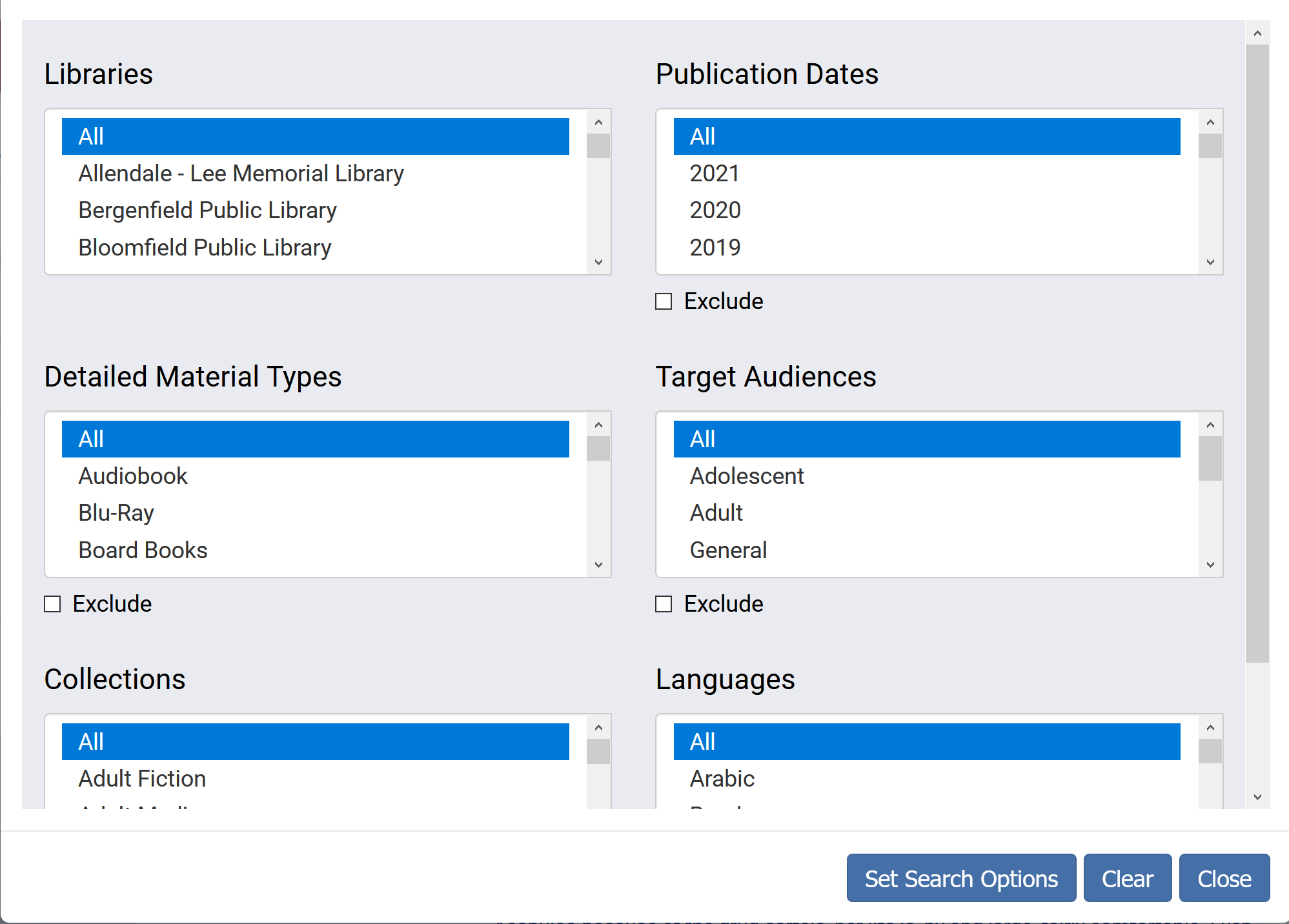


Figure ‑7. Advanced Search Screen

# Development History

Development of the LMS began on 26 October 2020 and concluded on 13 December, for a total duration of 8 weeks. For a breakdown of each project milestone, its due date, and status, refer to Table 8‑1.

Table ‑. Project Development History

| **Milestone** | **WBS** | **Due Date** | **Status** |
| --- | --- | --- | --- |
| Project Plan | 2.3. | 4-Nov-2020 | Completed |
| Test Plan | 3.4 | 11-Nov-2020 | Completed |
| Project Design | 4.5 | 18-Nov-2020 | Completed |
| User’s Guide | 7.3 | 18-Nov-2020 | Completed |
| Coding/Implementation Phase 1 | 5.1.1. | 25-Nov-2020 | Completed |
| Coding/Implementation Phase 2 | 5.2.1. | 2-Dec-2020 | Completed |
| Coding/Implementation Phase 3 | 5.3.1. | 9-Dec-2020 | Completed |
| Testing | 6.1. | 13-Dec-2020 | Completed |
| Final Project | 8.0 | 13-Dec-2020 | Completed |

# Conclusions

In summary, the goals and objectives set forth in the LMS project plan were successfully achieved. All requirements outlined in the Requirements Specification were met and validated by unit and system testing, and the overall design, as deemed in peer reviews, provided an intuitive user experience. Additionally, project deliverables, including this final report, were completed on, or ahead of, schedule.

The success of this project resulted from the collective efforts of the group. The willingness of individuals to step up and work outside their assigned roles to complete group assignments ensured the LMS was delivered on time, without sacrificing quality.

## Lessons Learned

It is not easy to manage a large software project with limited communication in a short time period. It is important to have a good plan going into a large project since it helps greatly in your ability to collaborate with others while producing the final product. Also, that it can be difficult to determine the work status when working in a group online.

## Design Strengths

The system was designed to separate the data, services and graphics which makes it easier to maintain and debug. There is good scalability in the design, since the hibernate database could be reconfigured easily to use a mysql or other type of sql server over a network. Also, I think we can incorporate search indexes and pagination, which means that the application will be able to handle a fairly large database and still be responsive to searches and display of queried results. I used encryption to store the user passwords so authentication is at least somewhat secure although the database itself is not encrypted. I think the user interface will also be fairly easy to use and responsive. There is an inactivity timer that automatically logs a user out after five minutes of inactivity, which is a nice security feature.

## Limitations

The database itself is not encrypted. I think it would be good to have at least the user information encrypted. Currently, it is created to run off a local database file. It would be more useful if it were setup on a sql server that could be connected to the internet. There would also need to be some additional tailoring to make the database responsive to requests from multiple users simultaneously. The GUI uses standard swing components which are responsive and not bad, but they do look a little dated.

## Suggestions for Future Improvement

With more time, it would be beneficial to refine some of the user interface making it even easier to use, as well as adding a fully functioning “Help” menu. While our LMS allows checkout of certain items like books and eBooks, it would be nice to offer other items for checkout. For example, I noticed several public libraries allow adult library patrons the option to checkout fishing poles and science kits (telescopes).

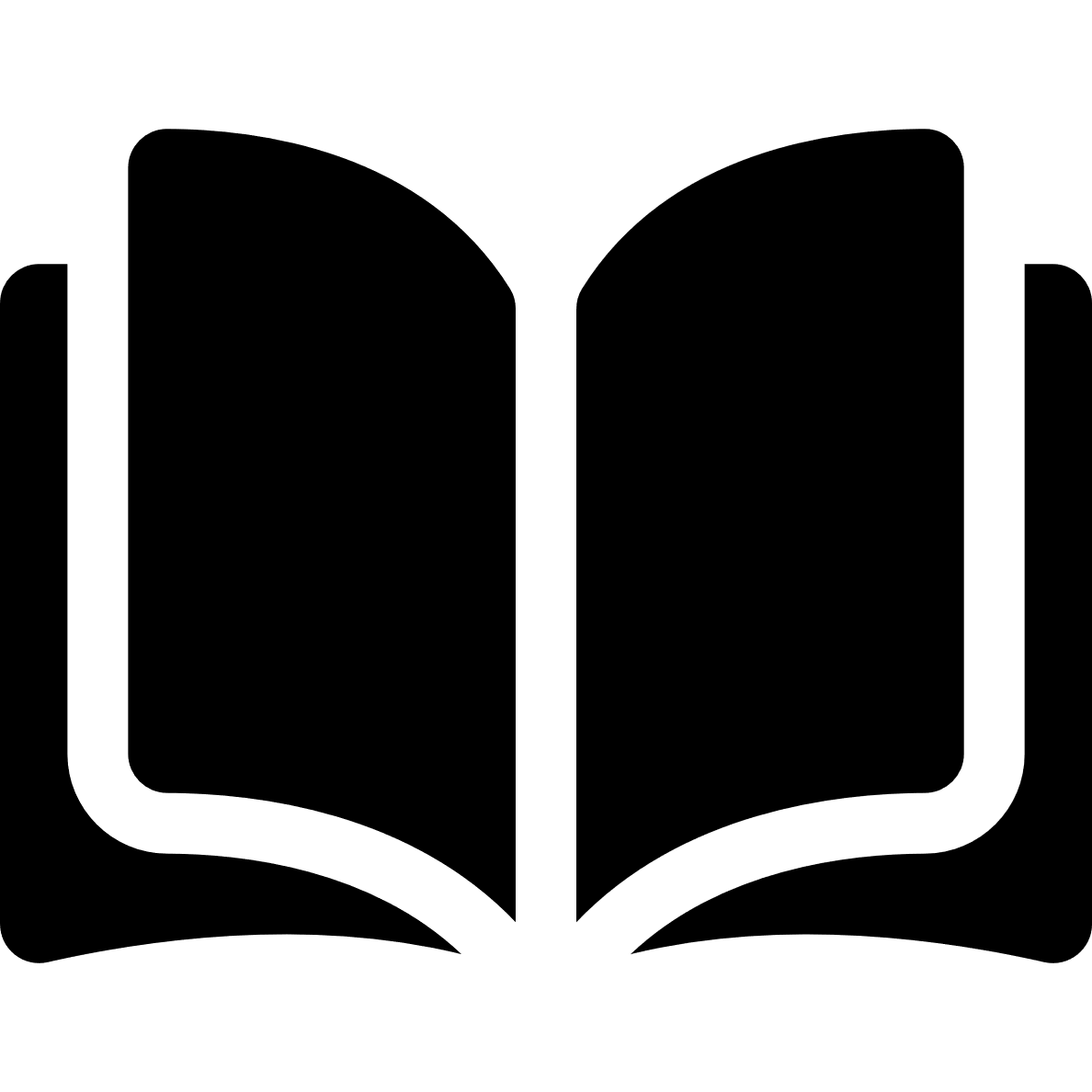
Appendix A

[Return to Main Page](#_User’s_Guide)

User’s Guide

for

Library Management System



Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Team Member** |
| 7-Nov-2020 | 0.1 | Initial draft | Cynthia Reynolds |
| 12-Nov-2020 | 0.2 | Added TOC | Cynthia Reynolds |
| 1-Dec-2020 | 0.3 | Added images | Cynthia Reynolds |
| 9-Dec-2020 | 0.4 | Added additional images | Cynthia Reynolds |

# Overview

## Purpose

The User’s Guide provides librarians and library patrons with step-by-step directions on using the Library Management System (LMS).

## Background

The LMS is an online application for assisting library users in accessing the assets in the library. The system provides a basic set of features to register new users, update personal information, search for library assets (e.g., books, Ebooks, Movies, Video Games), request and reserve library assets, and manage check-in and checkout processes.

## Getting Started

The LMS was developed using the Java programming language. Minimum system requirements to use the LMS are described below.

### Software

Windows 7 or later, 64-bit operating system.

### Hardware

Personal computer (PC) with at least 2 GB of RAM and Internet connectivity.

Input: Keyboard, mouse

Output: Monitor

### Installation and Setup

No installation is required since the LMS is distributed as an executable Jar file.

## Using the LMS

LMS is configured to support two types of users: a librarian and a library patron.

Functions of the librarian are:

* Create a librarian account [for new librarians]: Name, Username, Password)
* Log into their librarian account
* View book requests
* Checkout requested books
* Add a new book to the library database: title, author(s), ISBN, publisher, publication date, number of copies of the book
* Send notifications to patrons when book(s) are ready to be picked up
* Check-in returned books
* View questions from patrons
* Responds to patrons questions
* Confirm new patron registrations
* View patron account information
* Run/view a report to include the name/library card ID of patron, date books checked out and due to be returned, late fees

Functions of the library patron are:

* Create a library account [if one does not exist]: Name, Username, Password, Address, Phone number, Email, Library card ID
* Log into their library account
* Edit their library account information
* Reset their password
* Search for books
* Reserve books
* Request an extension for books
* Submit questions to the librarian
* View reserved books (available and unavailable)
* View late fees
* Receive notification when books are ready for pick-up

### Register a library patron

Prior to becoming a library patron, a new user will be prompted to create a new user account. The following patron information is required to create an account:

* Name (first and last)
* Address
* Phone number
* Email
* Username

Upon creation of an account, a library card ID will be provided.

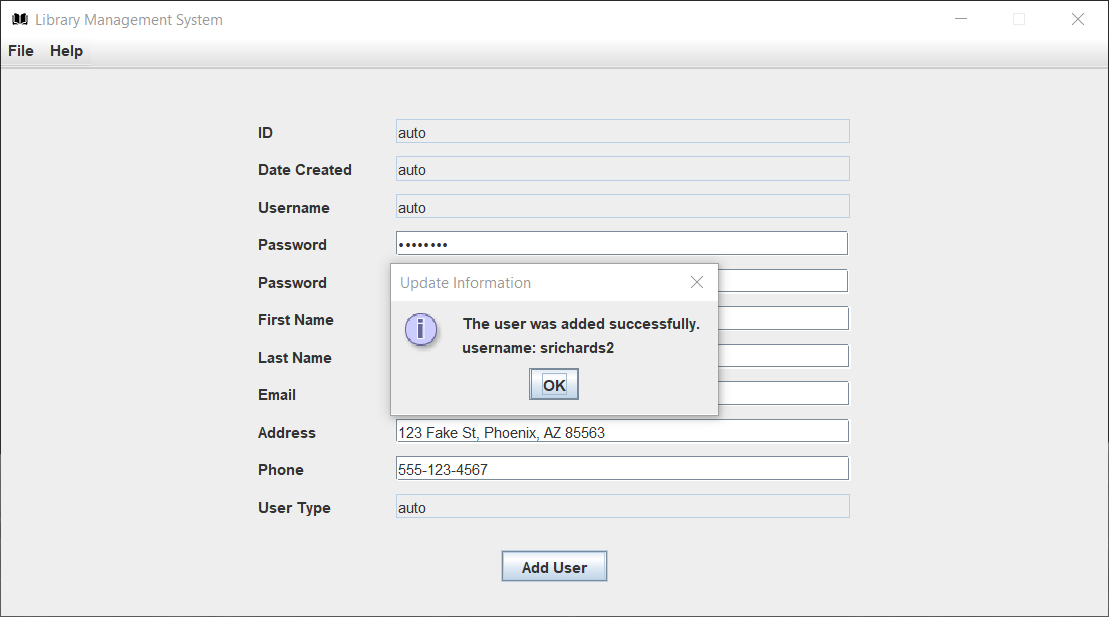


Figure A‑. Registration (Add User) Screen

### Login

Launch the LMS application. A prompt will be displayed to enter username and password.

Type username and password.

Press ENTER. Upon successful login, the home page will be displayed.

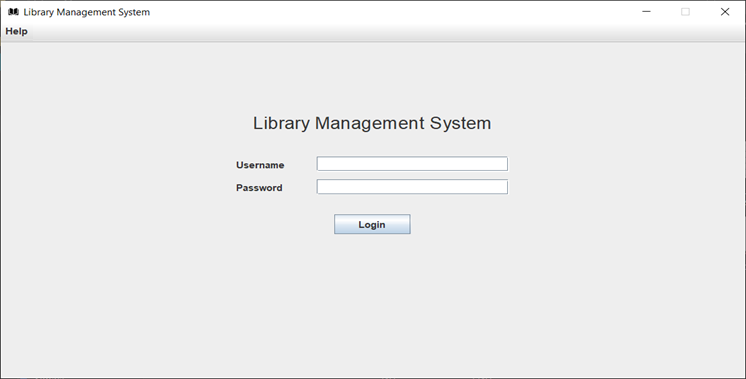


Figure A‑. Login Screen

### View/Modify patron registration information

#### View patron registration information

Login to LMS.

From the main menu, select the **My Account** link.

The user will be directed to the View screen where the following information will be displayed:

* Name
* Address
* Phone number
* Email
* Username
* Library card ID

**NOTE:** The password will not be displayed.

#### Modify Patron Registration Information

Login to LMS.

From the main menu, select the **My Account** link. The user will be taken to the Edit screen.

Edit desired field(s). Options include name, address, phone number, and email.

Press SUBMIT to save changes or press CANCEL to undo the changes you made.

Upon pressing SUBMIT, the following message will be displayed: “Your information has been updated”.

### Search for an Item

From the main menu, select **SEARCH**. Items may be searched by title or author.

Type search criteria (e.g., book title, author, DVD title) into the search box.

Press **ENTER**. Search results will be displayed to include the number of copies available for checkout, author’s name, publication date, and ISBN.

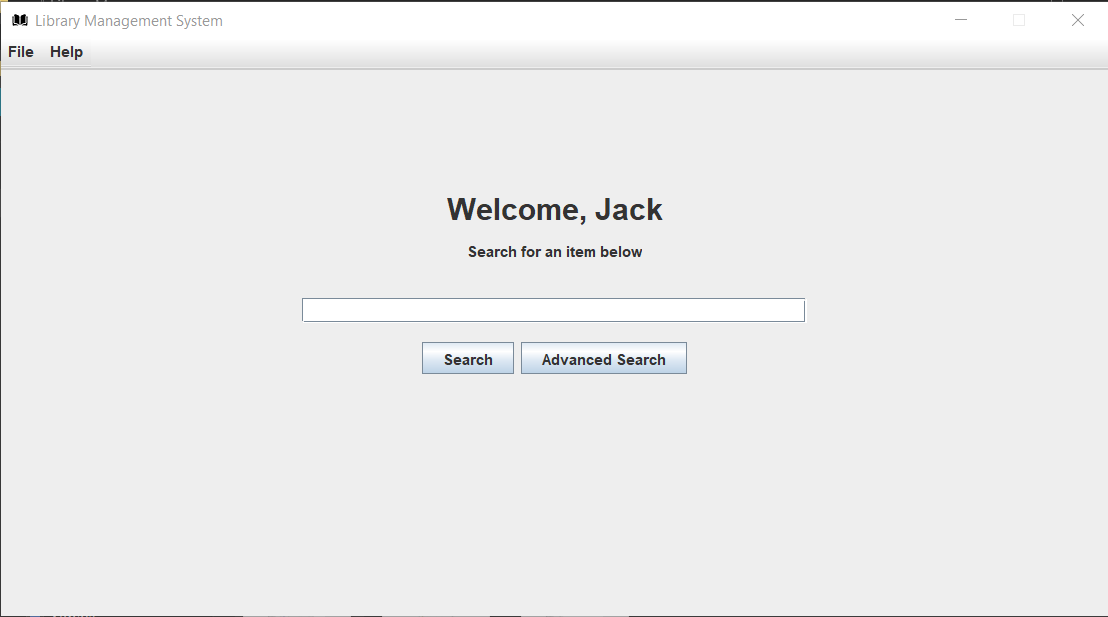


Figure A‑. Search Screen

### Checkout a Book

From the Search results page (see section 1.3.3.3.), click the radio button to select the desired book title.

Press **CHECKOUT**. The user will be redirected to a page displaying the following information regarding the book:

* Availability
* Title
* Checkout date
* Expected return date

### Reserve a Book

From the Search results page (see section 1.3.3.3.), select **Reserve Book**.

Select the checkbox next to the book to be reserved.

Press **SUBMIT**.

### Request Extension to Book Due Date

From the main menu, select the **Request Extension**.

Select the checkbox next to the book to be extended.

Press **SUBMIT**.

**NOTE:** There is a one-time extension for books currently checked out. After the one-time extension, books must be returned to the LMS.

### Check late fees

From the main menu, select **FEES**. Any late fees due to overdue books will be displayed on the page along with the associated book(s).

INSERT IMAGE HERE

### Logout due to inactivity

The system shall automatically logout a user after five minutes of inactivity.

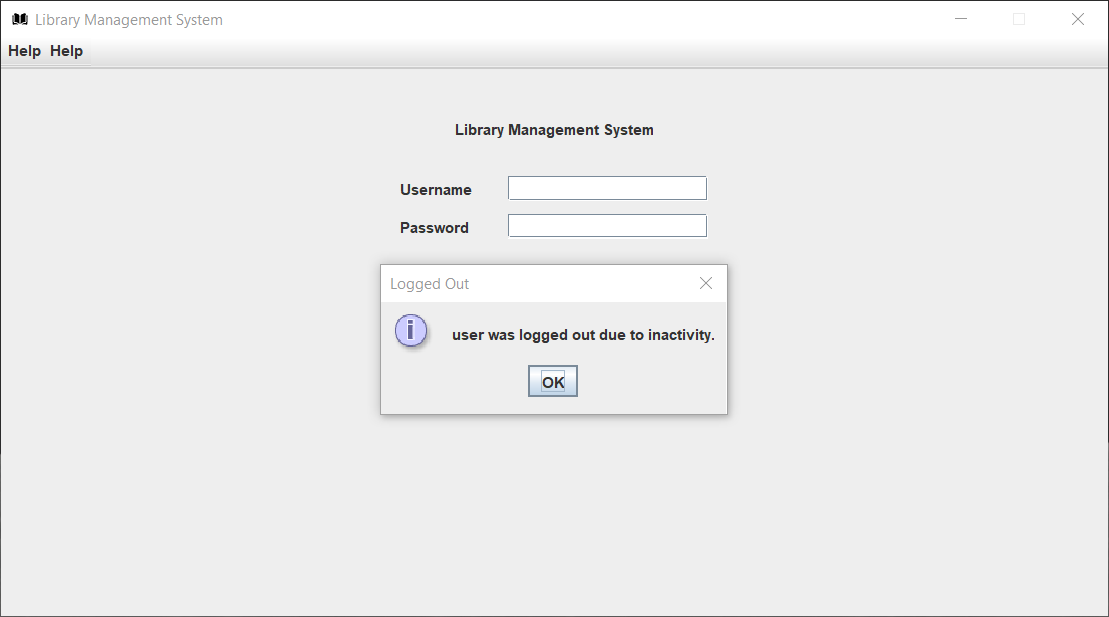


Figure A‑. User Inactivity Logout Dialog

## Glossary

Acronyms and abbreviations used throughout this User Guide are identified and defined in Table A‑1.

Table A-1. Acronyms and Definitions

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| GUI | Graphical User Interface |
| ISBN | International Standard Book Number |
| LMS | Library Management System |
| PC | Personal Computer |
| RAM | Random Access Memory |

**Contact Us**

For additional information or questions, please contact us at [LMS@NotReal.com](mailto:LMS@NotReal.com) or (999) 555-1234.