University of Maryland Global Campus

CMSC 495

Professor Kayed

Project Plan

Library Management System

Group 1

Gonghweh Koo

Scott Richardson

Cynthia Reynolds

Jonathan Dreksler

November 4, 2020

Table of Contents

[1. Purpose/Objectives 2](#_Toc55424048)

[2. Requirement Specifications 2](#_Toc55424049)

[3. Scenarios 2](#_Toc55424050)

[4. System Specifications 5](#_Toc55424051)

[5. User Interface 5](#_Toc55424052)

[6. Input/Output 5](#_Toc55424053)

[7. Design 5](#_Toc55424054)

[7.1 Purpose 5](#_Toc55424055)

[7.2 Abbreviations 5](#_Toc55424056)

[7.3 Summary 6](#_Toc55424057)

[7.4 System Overview 6](#_Toc55424058)

[8. Test Plan 6](#_Toc55424059)

[8.1 Background 6](#_Toc55424060)

[8.2 Introduction 6](#_Toc55424061)

[8.3 Test Items 6](#_Toc55424062)

[8.4 Features To Be Tested 7](#_Toc55424063)

[8.5 Features Not to Be Tested 7](#_Toc55424064)

[8.6 Approach 7](#_Toc55424065)

[8.7 Item Pass Criteria 7](#_Toc55424066)

[9. Work Breakdown Structure (WBS) 9](#_Toc55424067)

[10. Key Milestones (Status: To do, In-progress, Completed) 10](#_Toc55424068)

[11. Roles and Responsibilities 12](#_Toc55424069)

[12. Tools 12](#_Toc55424070)

# Purpose/Objectives

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, periodicals, DVDs). The user of the LMS will be a library patron. 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.

# Requirement Specifications

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. 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.

Functional requirements of the LMS include:

* The system shall allow a user to register an account.
* 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 following non-functional requirements will address availability and security:

* The system shall be available 24/7, with the exception of semi-annual maintenance.
* The system shall ensure only library patrons, with an active account, can log in to use the system.

# Scenarios

### 3.1 Use Case: Log in to LMS

|  |  |
| --- | --- |
| **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. |

### 3.2 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. |

### 3.3 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. |

### 3.4 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. |

### 3.5 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. |

### 3.6 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. |

# System Specifications

Software: Windows 7 or later OS; Java 8 JDK or JRE installed

Hardware: Personal computer with at least 2GB of RAM and Internet connectivity

# User Interface

Graphical User Interface (GUI)

# Input/Output

Input: Keyboard and mouse

Output: Monitor

# Design

## 7.1 Purpose

This document is based on the LMS requirements. The main purpose of the Design Documentation is to show the description of design and structure of the LMS’s interface. The final version of document will be provided to the developer for implementation of the selected structure.

## 7.2 Abbreviations

* GUI – Graphical User Interface
* LMS – Library management system
* SRS – Software requirements specification
* PC – Personal Computer
* UIE – User Interface Engine
* RD – Requirements Documentation
* DD – Design Documentation

## 7.3 Summary

* **Design Considerations:** provides an overall description of the software system including its functionality and matters related to the overall system.
* **Architectural Strategies:** describes the design decisions and strategies that affect the overall organization of the system and its higher-level structures.
* **System Architecture:** provides a high-level overview of how the functionality and responsibilities of system were partitioned and then assigned to subsystems.

## 7.4 System Overview

7.4.1 Assumptions and Dependencies

7.4.1.1 Related software, hardware and operating system

7.4.1.2 End-user characteristics

There are no special requirements for users other than understanding english. All of the user interfaces will be coded in this language.

# Test Plan

## 8.1 Background

The Library Management System is an online application for assisting library patrons in accessing the assets in the library. The system would 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.

## 8.2 Introduction

This test plan is a basic guideline for future testing in the LMS. It mainly focuses on two problems: what we will test and how we will test.

## 8.3 Test Items

8.3.1 GUI test

8.3.2 Basic function test

8.3.3 Database test

## 8.4 Features To Be Tested

8.4.1 GUI test

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

8.4.2 Database test

8.4.2.1 Basic operations: add/update/delete/query database

8.4.2.2 Advanced operations:

8.4.3 Basic function test

8.4.3.1 Add a new library patron

8.4.3.2 Update library patron personal information

8.4.3.3 Borrow book

8.4.3.4 Borrow DVD

8.4.3.5 Return book

8.4.3.6 Return DVD

8.4.3.7 Search for book

8.4.3.8 Search for DVD

8.4.3.9 Check-in book

8.4.3.10 Check-out book

8.4.3.11 Check-in DVD

8.4.3.12 Check late fees

8.4.4 Check connectivity of the system

## 8.5 Features Not to Be Tested

8.5.1 Maximum simultaneous online-users

8.5.2 Licensing requirement

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

## 8.6 Approach

8.6.1 Unit testing (class testing)

8.6.2 Integrity testing (thread-based testing)

8.6.3 Validation testing

## 8.7 Item Pass Criteria

8.7.1 GUI test

8.7.2 Database test

Pass criteria: Results of all basic and advanced operations are normal (refer to section 6.4)

8.7.3 Basic function test

8.7.3.1 Add a new library patron

Pass criteria:

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

8.7.3.2 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 DVD’s issued by library patron.

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

8.7.3.3 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.

8.7.3.4 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.

8.7.3.5 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.

8.7.3.6 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

8.7.3.7 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.

8.7.3.8 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.

8.7.3.9 View library patron detail

Pass criteria:

Librarians can select a user record for detailed view

The detail view should show:

a. User name, library card ID, Address, Email & Phone number

b. The books issued by user with issue date, due date, call number, title

c. 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.

# Work Breakdown Structure (WBS)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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** | **Design** | 5-Nov | 11-Nov | 7 |  | **Gonghweh** |
| 3.1 | Research and collaborate on application design | 5-Nov | 7-Nov | 3 |  |  |
| 3.2 | Finalize design decisions | 5-Nov | 8-Nov | 4 |  |  |
| 3.3 | Submit outline to GitHub | 5-Nov | 7-Nov | 3 |  |  |
| 3.4 | Work on Project Design | 8-Nov | 11-Nov | 4 |  |  |
| **3.5** | Complete and submit Project Design to Assignment folder | 11-Nov | 11-Nov | 1 |  |  |
| **4.0** | **Test Plan** | 12-Nov | 18-Nov | 7 |  | **Jonathan** |
| 4.1 | Research and collaborate on test requirements, test data, and test plan | 12-Nov | 14-Nov | 3 |  |  |
| 4.2 | Submit outline to GitHub. | 12-Nov | 14-Nov | 3 |  |  |
| 4.3 | Work on Test Plan | 15-Nov | 18-Nov | 4 |  |  |
| 4.4 | Complete and submit Test Plan to Assignment folder | 18-Nov | 18-Nov | 1 |  |  |
| 5.0 | **Coding/Implementation** | 19-Nov | 15-Dec | 28 |  | **Scott** |
| 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 | TBD |  |  | **Jonathan** |
| 6.1 | Perform testing |  |  |  | Will be performed following completion of the test plan |  |
| **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** | **15-Dec** | **15-Dec** | **1** |  | **Cynthia** |

# Key Milestones (Status: To do, In-progress, Completed)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Milestone** | **WBS** | **Responsible Team Member** | **Due Date** | **Status** |
| Group member names and topic submitted | 1.3 | Gonghweh | 27-Oct | Completed |
| Project Plan complete and submitted | 2.7 | Cynthia | 4-Nov | In-progress |
| Finalize Design decisions | 3.2 | Gonghweh | 8-Nov | To do |
| Project Design complete and submitted | 3.5 | Gonghweh | 11-Nov | To do |
| Test Plan complete and submitted | 4.4 | Jonathan | 18-Nov | To do |
| Coding/Implementation Phase 1 | 5.1.1 | Scott | 25-Nov | To do |
| Coding/Implementation Phase 2 | 5.2.1 | Scott | 2-Dec | To do |
| Coding/Implementation Phase 3 | 5.3.1 | Scott | 9-Dec | To do |
| Coding/Implementation Phase Final | 5.4 | Scott | 15-Dec | To do |
| Testing\* | 7.1 | Jonathan | TBD | To do |
| User’s Guide complete | 6.2 | Cynthia | 8-Dec | To do |
| Final project deliverables submitted | 8.0 | Cynthia | 15-Dec | To do |

\* Testing milestones will be outlined once the Test Plan has been finalized.

# Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| Team Member | Role | Responsibilities |
| Cynthia Reynolds | Planning Lead | Develop the project plan, manage deliverables according to the plan, establish the project schedule and determine each phase, track key milestones, assign tasks to team members. Ensure project remains on schedule. |
| Cynthia Reynolds | Requirements Lead | Define project requirements. Create and complete requirements documentation and system specifications. Ensure the requirements are being met. |
| Gonghweh Koo | Design Lead | Collaborate with the developer to outline the Library Management System design specifications. Create and complete design documentation. |
| Scott Richardson | Developer Lead | Develop the system software; define and manage coding and comment standards as well as implementation phases and milestones. |
| Jonathan Dreksler | Test Lead | Focus on evaluating the quality of a user's experience when interacting with the product. Create and complete the test plan. Collaborate with developers in order to create test cases and test data. Report results, and makes recommendations to improve effectiveness and overall satisfaction of the user. |

# Tools

Repository hosting service: Group 1 will use GitHub for storing and collaborating files and documents.

Communication tools: Google Meet will be used for our weekly check-ins and collaboration. We will also use ad hoc email messaging.

Development tools: Netbeans or Eclipse IDE; Java 8 JDK