

# Library Management System

## MODULE 2: SDLC ASSIGNMENT PART 1

TIFFANY ALBELTO

SOFTWARE DEVELOPMENT I | 202410-CEN-3024C-16046

# LIBRARY MANAGEMENT SYSTEM

August 28<sup>th</sup>,2023

## OVERVIEW

### 1. Define Requirements

- ✓ When building the LMS software it should be a console-based application used for managing a library's collection of books.

*The LMS software should allow users the accessibility to do as followed:*

- Add new books to the collection from a text file.
- All books should have a unique ID number, title, and author.
- Remove a book from the collection using its ID number.
- List all books currently in the collection.
- The text file for adding books should be formatted where each line represents a book, ID, title, and author are separated by a comma.

### 2. Gather Requirements:

- ✓ Schedule a meeting with the appropriate teams. (Librarian, Library Technician, Library Assistant, Public Relations Specialist, Administrative Service Manager, Computer Support Specialist) etc..

*Verify and determine the intended users: (Librarian, other staff members)*

*Verify and determine how the users will interact/ use the system:*

- Provide a text file that includes the book details when adding new books to the collection.
- Entering a unique ID number associated to the book when removing books from the collection.

- *Selecting an option when listing all books in the collection.*
- *Any system fallouts such as errors should provide an explanation or overview of the issue at hand.*

### 3. Implementation Plan:

- ✓ *Declared Programming Language: Java*

#### **Classes Suggested:**

##### **Library**

- **Class use:** *Manages the collection of books.*
- **Method use:** *add, remove, and list all books.*

##### **Book**

- **Class use:** *The Book class will symbolize an actual book and its attributes will be ID, Title, and Author*
- **Method use:** *Setters will be used to set or update its value, and Getters to return the value requested by user.*

#### **Methods Suggested:**

##### **Book Class Methods**

- *Int: setID*
  - *Void: sets the ID of the book*
- *String: setTitle*
  - *Void: sets the title of the book.*

- *String: setAuthor*
  - *Void: sets the author of the book.*
- *Int: getID() – retrieves or searches for the ID of the book.*
- *String: getTitle() – retrieves or searches for the title of the book.*
- *`String: getAuthor – retrieves or searches for the author of the book.*

### **Library Class Methods:**

- *addBook(book:Book)*
  - *void: adds a book to the book collection.*
- *removeBook(id:int)*
  - *void: removes a book from the book collection based off the ID that is entered by user.*
- *listBooks()*
  - *void: lists all books within the book collection.*

### **Code Summary Description:**

*The code implemented should read the text file that list the collection of books. Per requirements it should separate each line of the test file using a comma as the spacer. The book will be classified as the object with its attributes being ID, Title, and Author that will be using the comma as the spacer. The book object will be added to the Library's collection of books using the addBook() method. When removing a book from the collection of books a unique ID will be required by user where the system will search the database to retrieve the book associated with the ID entered by the user. If it is a match within the database the book will be removed using the removeBook() method. When user request to list all book collections the listBooks() method will be used to retrieve all books and its details associated to each individual book such as (ID, Title, Author).*

#### 4. Testing Plan:

- ✓ *Implement a testing script to test and validate the anticipated workflow of each code to ensure all class methods are successfully built according to requirements specified.*

*\*\* Ensure all testing teams are included on both the front end and back end to confirm coding logic is working as expected.*

#### 5. Deploy Software

- ✓ **Checklist:**

- *Implement the code according to build requirements and implementation plan.*
- *Validate all methods applied are reviewed, formatted, and correctly implemented to be packaged in an executable file.*
- *Save the code in your development, testing, production environments.*
- *Make sure all documentation, build requirements, and implementation plan are saved and can be accessible if needed.*
- *Ensure software is installed and readily available for all users.*
- *Ensure Training is scheduled for all users, as well as appropriate documentation is ready such as User guides.*
- *Send communication out to all users within ample time to ensure all users are aware and prepared for go-live.*
- *Have onsite support readily available to assist users as needed, and to provide feedback on any problems that may occur.*