SOFTWARE REQUIREMENTS SPECIFICATION

For

Book Borrowing Management System

Prepared by:

Ronaldo Repe Jr.

Jan Carlo Rabe

Kia Balucos

Submitted in fulfillment requirements for Integrative Programming and Technologies 2 (IT305)

August 09, 2025

Table of Contents

1. Introduction

Our proposed Book Borrowing Management System (BBMS) will be developed to replace the manual logbook method of recording the borrowing and returning of books. At present, all transactions are recorded manually, which may seem manageable at first, but eventually causes problems such as difficulty in tracking records, lack of accuracy, and challenges in monitoring book availability. The purpose of our proposed system will be to automate these processes by providing an efficient and user-friendly platform. Through this, the library staff will be able to manage records more effectively, while students, faculty, and staff will have a more convenient way of borrowing and returning books.

1.1. Purpose

The purpose of this system will be to provide a systematic, web-based solution that removes the inefficiencies of manual record-keeping. Our proposed system will be designed to assist the library staff in managing borrowing and returning transactions, ensuring that book records are always accurate and updated. At the same time, the system will benefit students and other borrowers by offering them an easy and reliable way to check book availability and complete transactions. Overall, the system will aim to save time, reduce errors, and improve the productivity of the library office.

1.2. Intended Audience

The primary users will be the library staff, who will use the system for daily tasks such as recording borrow and return transactions, checking the status of books, and updating records. The secondary users will include students, faculty, and staff, who will interact with the system to search for books, borrow or return them, and reserve resources when needed.

1.3. Product Scope

The scope of our proposed Book Borrowing Management System will focus on automating the essential functions of borrowing and returning books. The system will include features such as secure login for staff, book catalog management, a search tool to locate books by title or author, and transaction tracking using unique identifiers. Borrowers will also be able to reserve books through the system. However, the system will be limited to web-based use and will not yet include features such as e-book integration, fine or payment processing, or mobile application support. The main goal of our proposed system will be to provide a simple, reliable, and accessible platform that addresses the current needs of the library.

1.4. Definitions, Acronyms, and Abbreviations

The following are some important terms used in our document:

- **BBMS** – Book Borrowing Management System.
- **Catalog** – A structured list of available books with details such as title, author, and availability.
- **Transaction Number** – A unique identifier that will be generated for each borrowing or returning process to ensure accuracy in tracking.
- **User Roles** – The groups of people who will use the system, namely the administrator, librarian, and borrower.

2. Overall Description

2.1. User Characteristics

Our proposed system will be designed to serve two main groups of users. The first group will be the library staff, who usually have basic computer skills and will be responsible for managing and updating records. The system will be made intuitive and easy to use so they can operate it effectively with only minimal training. The second group will include students, faculty, and staff who will use the system as borrowers. Their interaction with the system will only involve basic actions such as searching for books, borrowing, returning, and reserving them.

2.2. Constraints

Our proposed system will need to work within certain constraints. On the hardware side, it will require computers with internet access for the staff. On the software side, it will run on web browsers and depend on a stable internet connection to function smoothly. The system will also need to comply with school regulations regarding the privacy and protection of student and staff data. In addition, it will follow database standards to ensure that all book records remain accurate and consistent.

2.3. Assumptions and Dependencies

The development of our proposed system will be based on certain assumptions and dependencies within the scope of our project. It will be assumed that the library staff will have access to a computer with a web browser and a stable internet connection to run the system. It is also assumed that the secondary users will have their laptops or smartphones with a browser and a stable internet connection to access the system. The system will depend on Apache24, which we are using as our local server to host the database and the web application during development and testing. Our system will also require API integrations, specifically the Google Account API for login authentication and the Google Calendar API for handling reminders, dates, and time management.

3. Requirements Specification Functional and Non-Functional Requirements

This section outlines the functional and non-functional requirements of the proposed system, ensuring that the identified needs of the clients are met and that they will be satisfied with the features provided.

3.1. Functional Requirements

- **User Authentication** - The system shall allow users (students, staff, or admin) to log in using a valid username and password. The system shall allow the librarian to send email notification to the user's account. The system shall allow users to securely log out after each session.
- **Book Management** - The system shall allow the Librarian to add new books with details such as title, author, and category. The system shall allow the Librarian to update or delete existing book records. And the system shall display the availability status of each book (Available, Borrowed, or Reserved).
- **Book Catalog -** The system shall allow users to search or look up books through the system. The system shall display the availability, title, author, and year published for each book. The system shall help borrowers easily determine what book they need.
- **Borrowing and Returning -** The system shall allow borrowers to reserve books online to replace the manual borrowing process. The system shall record the borrowing details including borrower name, book title, and date borrowed. The system shall allow the admin or librarian to record the return of borrowed books. The system shall automatically update the book's status to "Available" after return.

- **Transaction Number** - The system shall generate a unique transaction number for every reservation made by the borrower. The borrower shall use this transaction number to claim their reserved book(s) in the library for convenience.
- **Reminder Notification and Email -** The system shall send a notification to the borrower when the return date is near. When the due date has passed, the Librarian shall be able to send an email reminder to the borrower to return the book.
- **Favorite Feature -** The system shall allow users to add their favorite books. The system shall display the user's favorite books in a designated interface.
- **Borrowing History / Record Management -** The system shall maintain a complete history of all borrowing and returning transactions. Users shall be able to view their personal borrowing history. The Librarian shall be able to view and manage all transaction records.
- **User Roles and Access Control -** The system shall assign specific roles (Librarian and Borrower) with appropriate access rights. The librarian shall have control over book records, borrowing transactions, and notifications. The librarian shall **not** have permission to delete or modify borrower accounts. Borrowers shall only have access to search, borrow, and view book-related features.

3.2. Non-Functional Requirements

Our system also follows non-functional requirements to ensure quality and usability. The system shall provide a user-friendly interface so that both librarians and borrowers can easily navigate and use the system without requiring extensive training. It shall process transactions such as searching for books and reserving items within a short response time to ensure efficiency and responsiveness. The system shall be available 24 hours a day, 7 days a week, to guarantee continuous access for both staff and borrowers. It shall be compatible with smartphones, laptops, and desktop computers through web browsers, allowing users to access the system conveniently from multiple platforms. Additionally, the system shall implement strong login authentication with role-based access control to prevent unauthorized users from accessing either the staff interface or the borrower interface.

3.3. External Interface Requirements

User Interfaces

Our system will provide a web-based graphical user interface (GUI) accessible to both library staff and users (students, faculty, and staff) via both desktop and mobile browsers. The interface will include a login page, user dashboard, and admin panel. Navigation will be simple and responsive to support both desktop and mobile users.

Hardware Interfaces

The Book Borrowing Management System (BBMS) will be designed to operate on standard desktop computers, laptops, and mobile devices equipped with internet connectivity. No specialized hardware is required.

Software Interfaces

The (BBMS) will interact with:

- ***Database*****:** MySQL for storing book records, user accounts, and transactions.
- ***Operating Systems***: Compatible with Windows, Linux, and Android.
- ***Web Browsers***: Web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, etc.
- ***Third-party APIs***: Gmail and Google Calendar for sending due date reminders and notifications.

Communication Interfaces

- ***HTTP/HTTPS*****:** Protocols will be used for secure communication between clients and the server.
- ***RESTful APIs***: Handle book data and user interactions within the system.
- ***SMTP****:* Send email notifications regarding book due dates and borrowing confirmations.
- ***TCP/IP****: S*erve as the underlying network communication protocol.

4. System Models

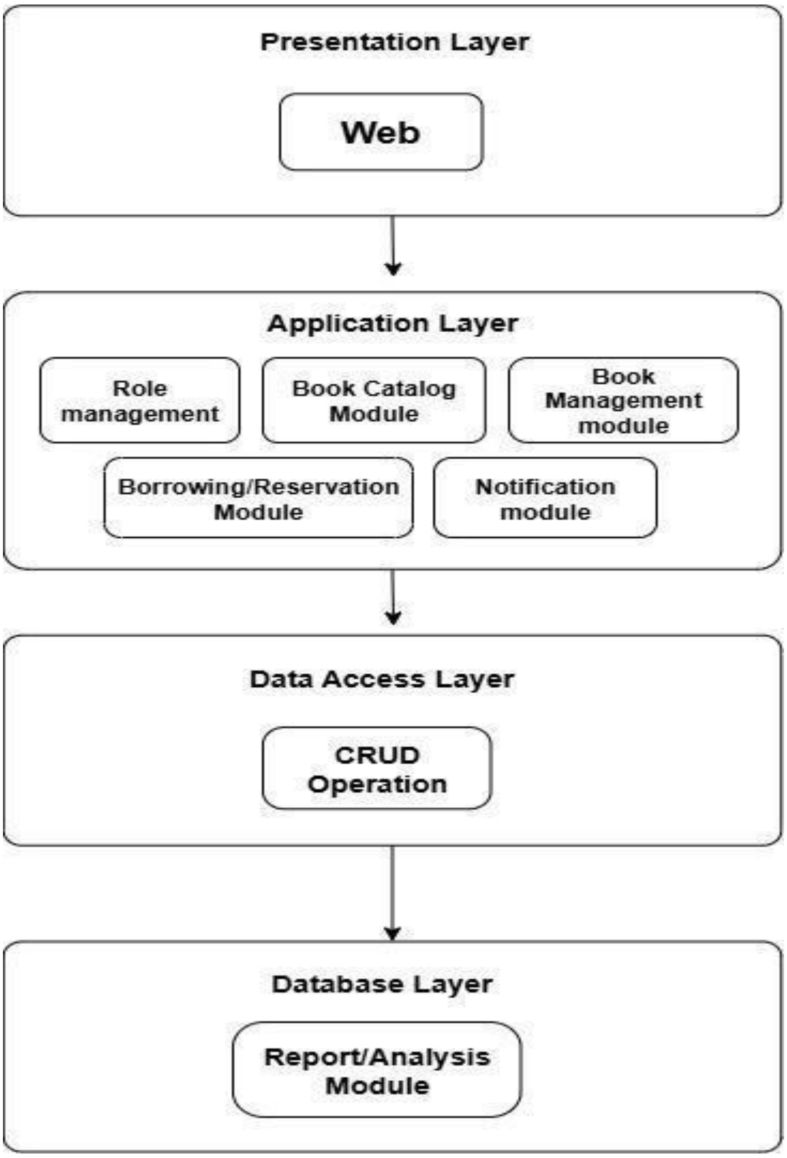

Figure 1: System Architecture

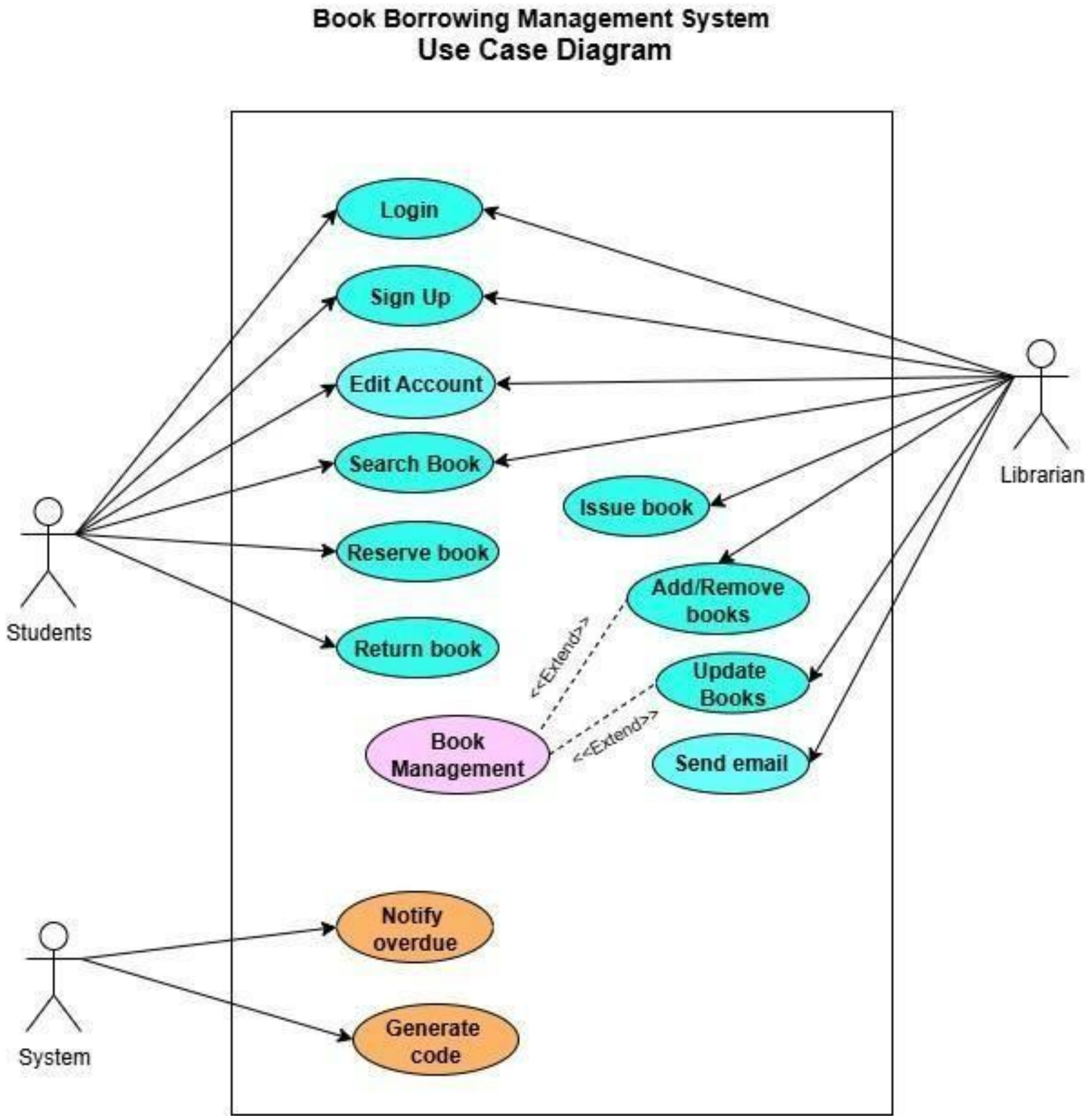

Figure 2: Use-Case Diagram

5. Database Design

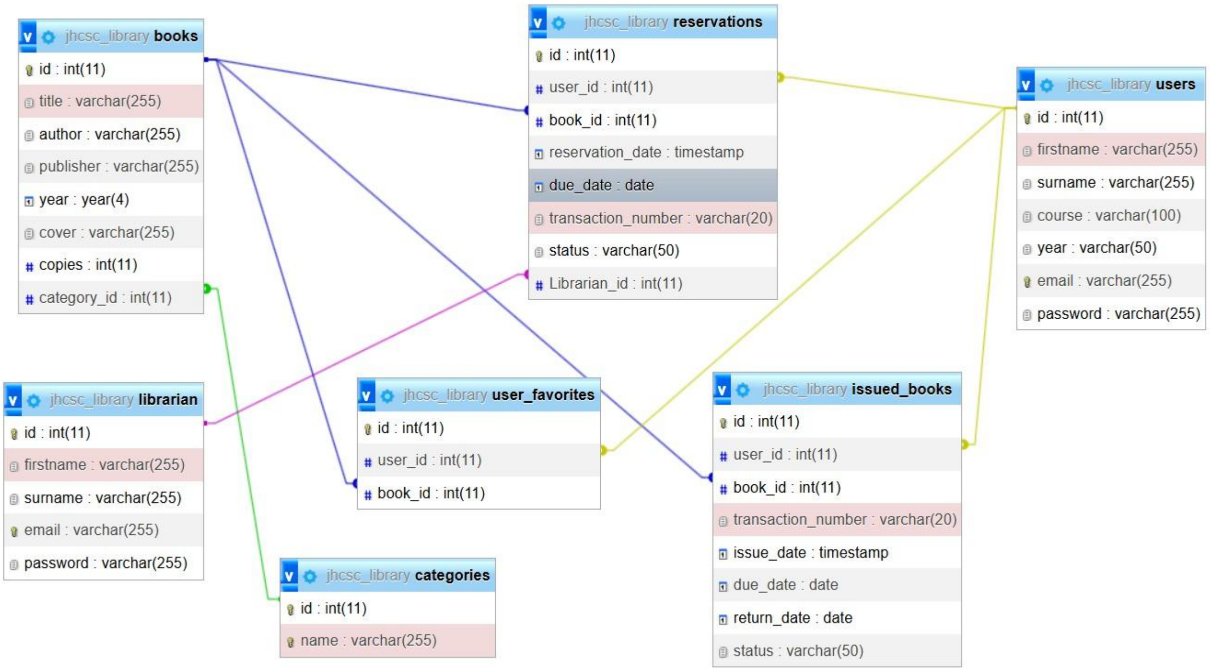

Figure 3: ERD Design

6. Implementation

Development Environment:

The setup of our system was developed using laptops as our main devices. We used Visual Studio Code for programming and coding to develop the system. The system files were stored in Apache24, and for the database, we used XAMPP with MySQL as the database management system. This setup allowed us to develop, test, and run the system efficiently in a local environment.

Programming Languages:

The programming languages used in developing the system were HTML, CSS, JavaScript, and PHP.

- HTML and CSS were used for designing and structuring the web pages.
- JavaScript was used to make the web pages more interactive.

- PHP was used for connecting the database to the back-end and front-end to process and display data.

  The **MySQL** database was used to store and manage all the system records and information.

Framework:

No specific framework was used in developing the system. The system was built manually using HTML, CSS, JavaScript, and PHP.

Tools Used:

The following tools were used to support the development process:

- **Visual Studio Code** – used as the main code editor for developing the system.
- **XAMPP** – used as the local server environment that runs Apache and MySQL.
- **phpMyAdmin** – used to access, manage, and organize the database.
- **Apache24** – used to store and host the system files locally.
- **Google Chrome** – used to test and run the system to check its functionality and design.