Software Requirement Specifications

[Automated Book Reservation System]

Supervisor	*Shazia Paras*
Project Team	*M. Sajjad* *Huzaifa Naveed* *Syed Talha Shafiq*
Submission Date	*7 May 2024*

Table of Contents:

1. Introduction:

1.1 Purpose:

The Automated Book Reservation System aims to streamline and enhance the book reservation process in libraries. By providing a user-friendly platform, it enables patrons to conveniently browse, check availability, and reserve books online. This system reduces administrative workload, improves user satisfaction, and enhances resource management through insights into reservation trends.

1.2 Scope:

It aims to provide users with an efficient way to read books. The system caters to users, administrators, and librarians, providing each with specific functionalities tailored to their roles.

1.3 Audience:

This document is intended for developers, designers, testers, and stakeholders involved in the development, testing, and implementation of the Automated Book Reservation System. It serves as a comprehensive guide to understanding the system's design and functionality.

2. Overall System Description:

2.1 Project Background:

The Automated Book Reservation System project emerged from the necessity to modernize library services. Recognizing the inefficiencies of manual reservation methods, our team is developing a digital platform to streamline the process. Through user feedback and research, we aim to create a convenient and efficient system for patrons to reserve books online. This project aligns with the goal of leveraging technology to enhance library accessibility and user experience in the digital era.

2.2 Project Scope:

The project scope involves developing an intuitive user interface catering to users, administrators, and librarians. Additionally, it encompasses implementing the necessary back-end logic for user authentication, book management, reservation handling, and borrowing processes. Furthermore, it includes establishing a robust database infrastructure to securely store user information, book details, and reservation records. These efforts aim to create the Automated Book Reservation System, streamlining book reservations and enhancing overall library management efficiency.

2.3 Project Objectives

The project objectives are to create an Automated Book Reservation System that addresses the inefficiencies of manual library processes, enhancing accessibility and user experience. This system aims to streamline book reservations, improve administrative efficiency, and optimize resource management. Ultimately, the project seeks to deliver a user-friendly platform that enables patrons to reserve books conveniently online while providing administrators and librarians with tools for effective book management. The end result will be a modernized library service that meets the evolving needs of patrons in the digital age.

2.4 Operating Environment

The Automated Book Reservation System will operate in an environment that supports C++ programming language, utilizing the Standard Template Library (STL) for efficient data manipulation. It will feature a command-line interface (CLI) for user interaction. As for hardware requirements, the system will run on processors with a minimum of Intel Core i3, clocked at 1 GHz or higher. A minimum of 1GB DDR3 RAM and 1 GB of free storage memory are required for smooth operation. Currently, the system does not require a separate database, as database functionality is integrated within the software.

3. External Interface Requirements

3.1 Hardware Interfaces

The software interfaces directly with the hardware components, primarily through the processor, RAM, and storage units. It supports device types compatible with the Intel Core i3 processor and DDR3 RAM. Data interactions involve efficient data manipulation utilizing the processor's capabilities, while control interactions manage storage and memory allocation for smooth operation. These interfaces ensure seamless communication between the software and hardware components, optimizing system performance.

3.2 Software Interfaces

Since the Automated Book Reservation System does not rely on external software components or services, its software interfaces primarily involve internal connections between its own modules. The system's core components, including the user interface, back-end logic, and database, communicate internally using C++ and the Standard Template Library (STL). Data items such as user information, book details, and reservation records are exchanged among these components for seamless operation. The communication nature is tightly integrated, with data shared directly between modules to ensure efficient book reservation processes and effective library management.

4. Functional Requirements

4.1 Functional Hierarchy

1. **Admin**
   - Login
   - Manage Librarians
     - Add Librarian
     - Add User
2. **Librarian**
   - Login
   - Manage Books
     - Update Books
     - Add Book

        - Remove Book
        - Checkout Book
        - Display Books

3. **User**
    - Login
    - Manage Borrowed Books
        - Extend Borrowed Books
        - View Borrowed Books
        - Report Loss
    - Manage Reservations
        - View Reservations
        - Reserve a Book
        - Extend Reservation
        - Borrow a Book
    - Manage Fines
        - View Fines

Library Database

- Interact with all the above functionalities

4.2 Use Case

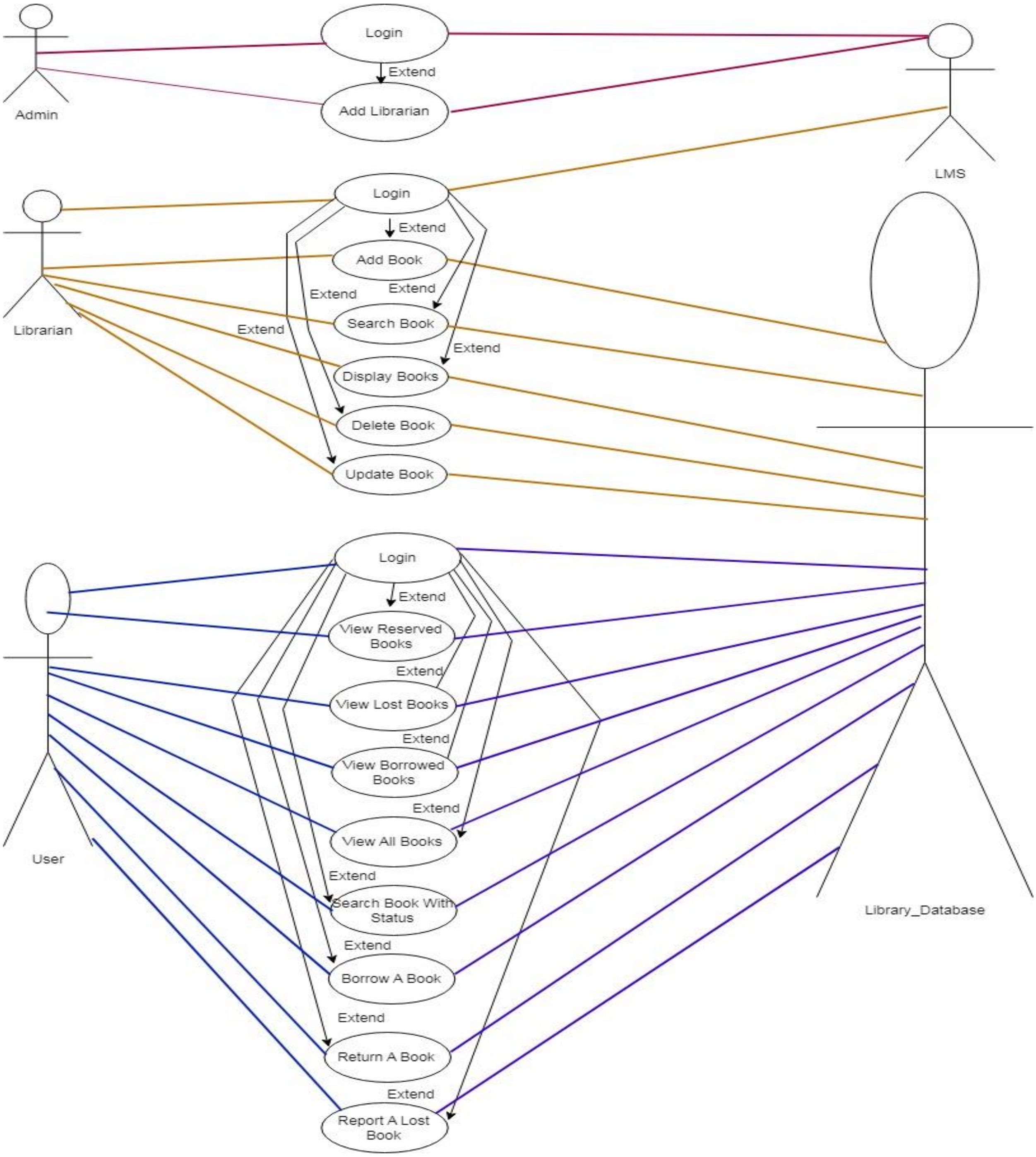
The diagram identifies three actors: Admin, Librarian, and User. Each actor is associated with various use cases, representing the actions they can perform within the system.

The admin actor has two primary use cases: Login and Add Librarian. The Login use case is extended, indicating additional functionalities related to login that an admin can perform. The Add Librarian use case allows the admin to add new librarians to the system.

The Librarian actor interacts with the system through several use cases, including Login, Add Book, Add Member, Search Book, Update Book Details, and Delete Book. Each of these use cases has an associated extension point labelled as Extend, suggesting additional functionalities branching from the main ones.

The User actor has the most complex set of interactions, with numerous interconnected use cases such as Login, Search Books, Borrow Books, View Borrowed Books, View All Books with Fine Details, Renewal, Extend A Book, Return A Book, and several others. These use cases represent the various actions a user can perform within the LMS.

All actors are ultimately connected to a central system entity represented by a rectangle labelled as Library Database. This indicates that all actions performed by any actor interact with or affect this central database.


Login
Extend
Add Librarian
Admin
LMS
Login
Extend
Add Book
Extend
Extend
Search Book
Extend
Extend
Display Books
Delete Book
Update Book
Librarian
Library_Database
Login
Extend
View Reserved Books
Extend
View Lost Books
Extend
View Borrowed Books
Extend
View All Books
Extend
Search Book With Status
Extend
Borrow A Book
Extend
Return A Book
Extend
Report A Lost Book
User

5. Non-Functional Requirements

5.1 Performance Requirements

The performance requirements for the Automated Book Reservation System encompass several key aspects:

- **Speed:** The system should respond promptly to user requests, with minimal latency in accessing book information, making reservations, and performing database transactions.
- **Precision**: The system must accurately retrieve and display book availability, reservation statuses, and user information to ensure precision in operations.
- **Concurrency**: The system should support multiple users simultaneously, allowing concurrent access to the system without performance degradation or data integrity issues.
- **Capacity**: The system should handle a large volume of book reservations and user interactions efficiently, without experiencing slowdowns or resource exhaustion.
- **Safety**: Data safety and security are paramount. The system must ensure the confidentiality, integrity, and availability of user data and system functionalities.
- **Reliability**: The system should operate reliably under various conditions, minimizing downtime and errors to provide a seamless user experience.

These performance requirements define the operational efficiency and effectiveness of the Automated Book Reservation System, meeting the needs of both users and library administrators.

5.2 Safety and Security Requirements

Safety Requirements: The system must ensure the protection of user data and prevent any potential loss or harm resulting from its use. Safeguards should be implemented to mitigate risks, such as data encryption and regular backups. Additionally, dangerous actions, such as unauthorized access or data manipulation, must be prevented to maintain system integrity and user trust.

Security Requirements: The system must prioritize security, integrity, and privacy to safeguard user data and system functionalities. Strong authentication mechanisms should be in place to verify user identity and ensure authorized access. Compliance with security policies and regulations, as well as adherence to privacy standards, is essential for maintaining user trust and meeting legal requirements.

6. Conclusion

In conclusion, this Software Requirements Specification (SRS) document outlines the comprehensive framework for the development of the Automated Book Reservation System. By addressing key components such as user interface, back-end logic, and database functionality, this document provides a clear roadmap for designing and implementing a modernized solution to streamline book reservation processes in libraries. The defined performance, safety, and security requirements underscore our commitment to delivering a system that not only enhances user experience but also prioritizes data protection and system reliability. With this SRS document as our guide, we embark on a journey to create a user-friendly, efficient, and secure platform that meets the evolving needs of patrons and library administrators alike.