Library Management System

Software Requirements Specification

Date: 1/3/2025

Bara Al Omari - 60300383

Abdulrhman Ibrahim - 60101806

Contents

1. Introduction

1.1 Purpose

The purpose of this document is to define the functional and non-functional requirements for the **Library Management System**. The Library Management System is a desktop-based application for librarians to manage library operations efficiently. It includes book checkouts, returns, cataloging, member management, due date tracking, fines for late returns, and a search functionality for finding books, while ensuring security against potential threats.

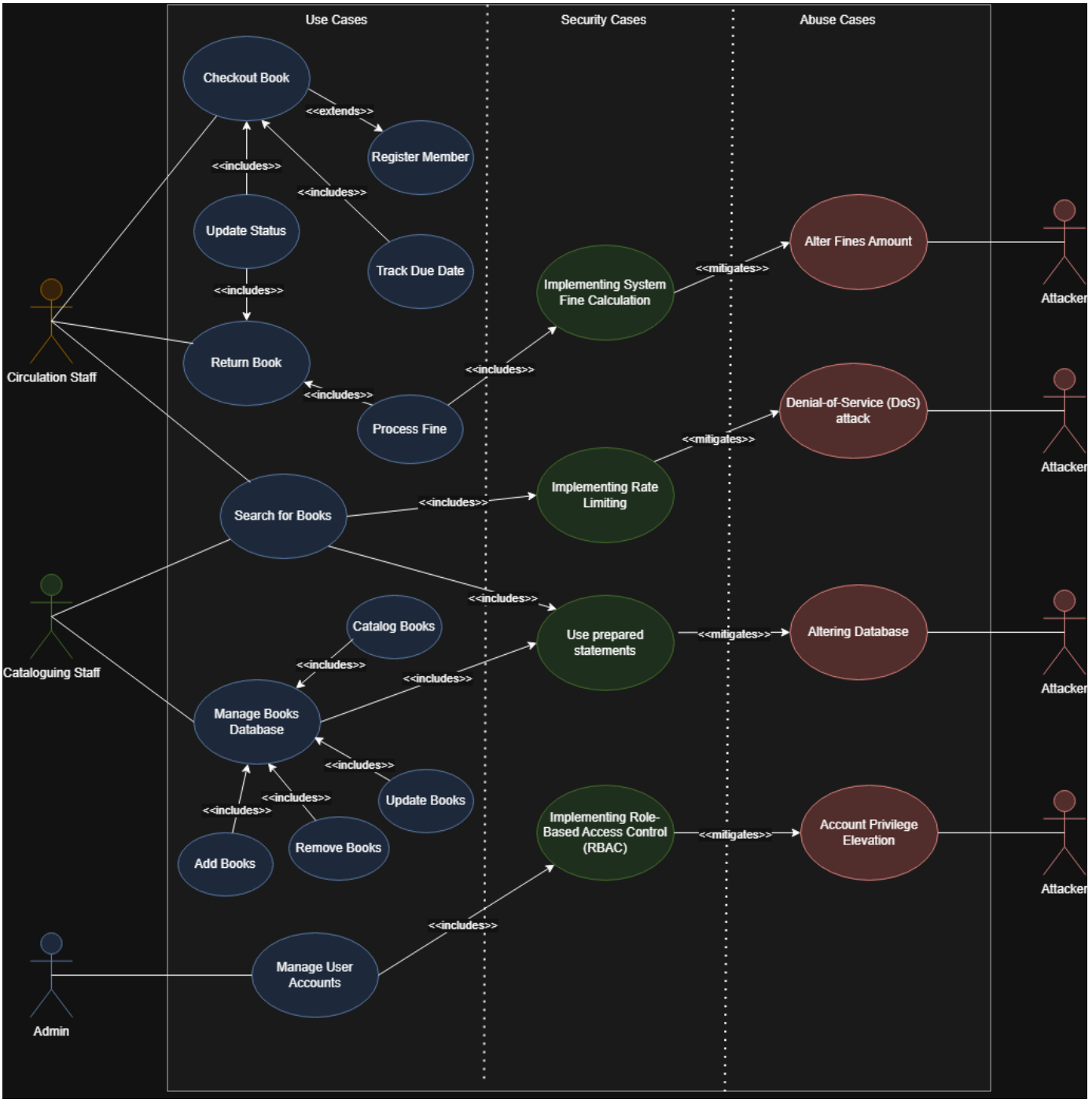
1.2 Scope

The Library Management System is intended for library staff and administrators to manage internal operations, including book lending, cataloging, and user account management. The system must also implement security measures to prevent unauthorized access and potential abuse.

2. Non-Functional Requirements

Category	**Requirement**
Security	The system should ensure security against potential threats.
Performance	The system should process transactions within 2 seconds.
Usability	The system should provide an intuitive UI for staff users.
Availability	The system must be available 99.9% of the time.

3. Use Case Diagram



4. Functional Use Cases

Use case Id:	F001
Name:	Checkout Book
Created By/Author:	Bara Al Omari
Date Created:	28/2/2025
Actor:	Circulation Staff
Description/Summary:	Staff check out books to library members.
Preconditions:	Book must be available, and member is registered.
Postconditions:	The book status is marked as checked out.
Normal course of events:	1. Staff enter ISBN. 2. Register Member if not registered 3. System records the checkout date and due date. 4. System updates the book's status.
Exceptions:	Book is already checked out.
Acceptance Criteria:	Book is successfully checked out to the member, book status is updated and due date tracking is initiated.

Use case Id:	F002
Name:	Return Book
Created By/Author:	Bara Al Omari
Date Created:	28/2/2025
Actor:	Circulation Staff
Description/Summary:	Staff marks a book as returned.
Preconditions:	The book must be checked out.
Postconditions:	The book is available for checkout again.
Normal course of events:	1. Staff enter ISBN. 2. System verifies the book's status. 3. System Calculates fine. 4. System updates records.
Exceptions:	
Acceptance Criteria:	The book's status changes to available.

Use case Id:	F003
Name:	Search for Books
Created By/Author:	Bara Al Omari
Date Created:	28/2/2025
Actor:	Circulation Staff & Cataloging Staff
Description/Summary:	Allows searching for books by title, author, or ISBN.
Preconditions:	System must have book records.
Postconditions:	Search results are displayed.
Normal course of events:	1. Staff enter search criteria. 2. System retrieves matching records.
Exceptions:	No matches found.
Acceptance Criteria:	Search results match the query.

Use case Id:	F004
Name:	Process Fine
Created By/Author:	Bara Al Omari
Date Created:	28/2/2025
Actor:	Circulation Staff
Description/Summary:	Calculates and processes overdue fines.
Preconditions:	Fine is processed and displayed in the system.
Postconditions:	Search results are displayed.
Normal course of events:	1. System checks due dates. 2. System calculates fines. 3. Staff processes payment.
Exceptions:	
Acceptance Criteria:	Fine is correctly recorded and processed.

Use case Id:	F005
Name:	Register Member
Created By/Author:	Bara Al Omari
Date Created:	28/2/2025
Actor:	Circulation Staff
Description/Summary:	Staff registers a new library member in the system.
Preconditions:	The person must provide valid identification. The person must not already be a registered member.
Postconditions:	The new member is successfully added to the system. The member receives a unique ID for future transactions.
Normal course of events:	1. Staff enters the person's details. 2. System assigns a unique member ID. 3. System stores the member's information in the database. 4. System confirms successful registration.
Exceptions:	The member is already registered.
Acceptance Criteria:	A unique member ID is generated. The new member appears in the system's database.

Use case Id:	F006
Name:	Manage Books Database
Created By/Author:	Bara Al Omari
Date Created:	28/2/2025
Actor:	Cataloguing Staff
Description/Summary:	The cataloguing staff manages the library's book database, including adding, updating, and removing book records.
Preconditions:	The staff must be logged into the system with the appropriate permissions.
Postconditions:	The book database is updated successfully.
Normal course of events:	1. The staff selects the "Manage Books Database" option. 2. The staff chooses an action: Add, Update or Remove a book. Add Book: ○ Staff enters book details (title, author, ISBN, category, etc.). Update Book: ○ Staff searches for a book in the database. ○ Staff modifies necessary fields. Remove Book: ○ Staff searches for a book in the database. ○ System asks for confirmation. 3. System confirms the completion of the operation.
Exceptions:	The book to be updated or removed is not found in the database. The staff does not have the required access rights.
Acceptance Criteria:	Books can be successfully added, updated, or removed. System confirms changes after each operation.

Use case Id:	F007
Name:	Manage User Accounts
Created By/Author:	Bara Al Omari
Date Created:	28/2/2025
Actor:	Admin
Description/Summary:	The Admin manages user accounts, including updating user information, removing accounts, modifying permissions, etc.
Preconditions:	The Admin must be logged into the system with appropriate privileges. And user must be registered.
Postconditions:	The user account database is updated successfully. Any modifications are logged in the system.
Normal course of events:	1. The Admin selects the "Manage User Accounts" option. 2. The Admin chooses an action: Modify Permissions, Update, or Remove a user. 3. Admin searches for an existing user. 4. Admin Modifies User. 5. System confirms the completion of the operation.
Exceptions:	
Acceptance Criteria:	Users can be successfully modified. System prevents unauthorized modifications. And confirms changes after each operation.

5.Abuse Cases

Abuse case Id:	A001
Name:	Alter Fines Amount
Created By/Author:	Bara Al Omari
Date:	1/3/2025
Priority:	High
Scope:	Library Management System – Fine Processing Module
Development Environment:	Desktop Application
Mis-actors:	Unauthorized users, Insider Threats
Access Right Levels:	Medium-Level System Users and High-Level System Users
Point of Entry:	Access through direct modification in the database.
Security Attributes Affected:	Integrity
Description:	An attacker or malicious staff member alters the fine amounts in the system to reduce or remove penalties for overdue books, causing financial losses for the library.
Sophistication:	Moderate to High
Preconditions:	The attacker has access to the system. The system does not enforce strict access controls. Fines are stored in the database and not presented automatically.
Assumptions:	Fine amounts are stored in a database and can be modified.
Postconditions:	Financial records are manipulated
Related Use Cases:	Process Fine
Related Threats:	Altering Database
Exceptions:	
Acceptance Criteria:	The system should prevent unauthorized modifications of fines.

Abuse case Id:	A002
Name:	Denial of Service (DoS) Attack
Created By/Author:	Bara Al Omari
Date:	1/3/2025
Priority:	High
Scope:	System Input fields
Development Environment:	Desktop Application
Mis-actors:	Malicious users
Access Right Levels:	Medium-Level System Users
Point of Entry:	Book search functionality
Security Attributes Affected:	Availability
Description:	Attackers flood the book search system with excessive requests, resulting in resource exhaustion and making it unavailable to legitimate users
Sophistication:	Medium
Preconditions:	No effective rate-limiting mechanism is in place
Assumptions:	No rate-limiting is being implemented, search functionality available in the system.
Postconditions:	System resource exhaustion. Performance degradation.
Related Use Cases:	Search for Books
Related Threats:	Automated Bots, Network Traffic Flooding
Exceptions:	
Acceptance Criteria:	The system prevents excessive search requests from a single source.

Abuse case Id:	A003
Name:	Altering Database
Created By/Author:	Bara Al Omari
Date:	1/3/2025
Priority:	High
Scope:	Library Management System Database
Development Environment:	Database Management System (MySQL/MongoDB)
Mis-actors:	Malicious Insiders
Access Right Levels:	Medium-Level System Users and High-Level System Users
Point of Entry:	Direct database access, SQL injection vulnerability
Security Attributes Affected:	Integrity, Confidentiality
Description:	A user or attacker modifies or deletes database records, potentially corrupting book catalog data, user accounts, etc
Sophistication:	High
Preconditions:	The database is accessible through system tools or SQL interfaces.
Assumptions:	The system has vulnerabilities (e.g., lack of prepared statements).
Postconditions:	System data integrity is compromised.
Related Use Cases:	Search for Books, Manage Books Database
Related Threats:	SQL Injection
Exceptions:	
Acceptance Criteria:	Unauthorized changes to the database are prevented

Abuse case Id:	A004
Name:	Account Privilege Elevation
Created By/Author:	Bara Al Omari
Date:	1/3/2025
Priority:	High
Scope:	Desktop Application
Development Environment:	Database Management System (MySQL/MongoDB)
Mis-actors:	Malicious Users, Insider Threats
Access Right Levels:	Medium-Level System Users
Point of Entry:	User account management system
Security Attributes Affected:	Integrity, Confidentiality
Description:	A malicious user attempts to elevate their privileges (e.g., from a normal user to an admin) to gain unauthorized access to system functionalities.
Sophistication:	High
Preconditions:	The system assigns roles to users.
Assumptions:	The system has vulnerabilities in authentication mechanisms.
Postconditions:	The attacker gains unauthorized admin privileges.
Related Use Cases:	Insider Threats
Related Threats:	SQL Injection
Exceptions:	
Acceptance Criteria:	Security measures are implemented (such as RBAC)

6. Security Cases

Use case Id:	S001
Name:	Implementing System Fine Calculation
Created By/Author:	Bara Al Omari
Date Created:	1/3/2025
Actor:	System
Description/Summary:	Ensures that the library system correctly calculates fines based on overdue book returns. It prevents unauthorized modifications to fine amounts and ensures consistency in fine enforcement.
Preconditions:	A book has been checked out by a library member and the system has predefined fine rules based on due dates.
Postconditions:	If a book is returned late, the system calculates and records the appropriate fine.
Normal course of events:	1.The system tracks due dates for all borrowed books. 2.When a book is returned, the system calculates any fines based on the return date.
Exceptions:	
Acceptance Criteria:	The system accurately calculates fines based on due dates.

Use case Id:	S002
Name:	Implementing Rate Limiting
Created By/Author:	Bara Al Omari
Date Created:	1/3/2025
Actor:	System
Description/Summary:	Ensures that users cannot overload the system with excessive requests, preventing denial-of-service (DoS) attacks and bot abuse while maintaining system performance.
Preconditions:	The system tracks requests frequency.
Postconditions:	If a user exceeds the allowed request limit within a short time, further requests are temporarily blocked.
Normal course of events:	1. The system tracks the number of requests per user within a given time frame. 2. If the user exceeds the search request threshold, further requests are blocked for a cooldown period.
Exceptions:	
Acceptance Criteria:	Users are restricted from excessive queries and system requests within a short period. System performance remains stable

Use case Id:	S003
Name:	Using Prepared Statements
Created By/Author:	Bara Al Omari
Date Created:	1/3/2025
Actor:	System
Description/Summary:	Prevents SQL injection attacks by enforcing the use of prepared statements for all database queries.
Preconditions:	The system interacts with a database.
Postconditions:	User inputs are sanitized before executing any database transactions.
Normal course of events:	1. A user submits data. 2. The system processes the input using prepared statements. 3. The database executes only predefined parameterized queries.
Exceptions:	
Acceptance Criteria:	All database queries use prepared statements.

Use case Id:	S004
Name:	Implementing Role-Based Access Control (RBAC)
Created By/Author:	Bara Al Omari
Date Created:	1/3/2025
Actor:	System
Description/Summary:	Ensure that users have access only to the features necessary for their role, preventing unauthorized actions.
Preconditions:	User roles are predefined (Admin, Circulation Staff, Cataloguing Staff, etc.).
Postconditions:	Users can only perform actions permitted by their role.
Normal course of events:	1. A user logs into the system. 2. The system checks the user's role and grants access only to permitted features. 3. If a user attempts an unauthorized action, access is denied.
Exceptions:	
Acceptance Criteria:	Role-based permissions are enforced correctly. Unauthorized actions are blocked and logged.