

# SRS for Library Management System

Group 14

## Index

- Index
- Introduction
  - Purpose
  - Scope
  - Product Perspective
- Overall Description
  - User Characteristics
    - \* Admin
    - \* Librarian
    - \* Faculty/Student
    - \* User
  - Product Functions
    - \* Use Cases related to user management
      - Login
      - Logout
      - View Profile
      - Edit Profile
    - \* Use Cases related to administration
      - Add new member
      - Remove member
      - Update member
      - Add a fine
      - Remove a fine
      - Update a fine
      - Pay a fine
      - View Fines
    - \* Use Cases related to books/journals/e-resources
      - Add new book

- Remove book
- Update book
- Add new book copy
- Remove book copy
- Add new E-Resource
- Remove E-Resource
- Update E-Resource
- View Book Catalog
- View E-Resource Catalog
- Search for Book/E-Resource/Journal
- View New Arrivals
- View Popular Books
- View Book/E-Resource/Journal Details
- Issue Book
- Update Book Issue Details
- Request Book Issue
- Reserve Book
- Return Book
- Request Book
- Request E-Resource/Journal
- Read E-Resource/Journal
- View All Book Issues/Returns/Reservations/Requests of a Member
- View All Requests
- Update Request
- View All Unresolved Issues/Returns/Reservations/Requests
- \* Miscellaneous Use Cases
  - View Information/FAQs (Frequently Asked Questions)
  - View Contact Us
  - View About Us
- Non-Functional Requirements/Constraints/Assumptions
  - Constraints
  - Assumptions
  - Performance Requirements
- Appendix
  - References

# Introduction

## Purpose

The purpose of this document is to provide a detailed description of the requirements for the Library Management System for a university. It will explain the purpose and features of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

## Scope

The system will be used by the following stakeholders:

- Librarian
- Administrator
- Faculty
- Students

The system will manage the following:

- Books and book copies
- E-Resources and Journals
- Members of the library
- Fines
- Book issues, returns, reservations and requests

## Product Perspective

The Library Management System will be a standalone system that will be used by the library of a university. It will be used to manage the books, journals, e-resources, members and fines of the library. It will also be used to manage the issues, returns, reservations and requests of the library.

# Overall Description

## User Characteristics

Note that **user** is a generic term that refers to any of the below actors.

Below is the list of users that we intend to support.

### Admin

Admin refers to the **administrator** of the library management system. The admin is responsible for the management of the library management system and users of the system.

### Librarian

The librarian is responsible for the following:

- Managing the books/book copies in the library.
- Managing the E-Resources / Journals in the library.
- Issuing books/E-Resources/Journals to the users.
- Managing and issuing fine to the users.

### Faculty/Student

The faculty/student can perform the following actions:

- Issue books from the library.
- Return books to the library.
- Read E-Resources / Journals.
- View the list of books issued or returned by them.
- Request for new books/E-Resources/Journals to be added to the library.
- Reserve books in advance when they are not available.

There would be little difference between the actions performed by a faculty and a student. Such differences would be values of certain fields in the database.

### User

Whether a user is logged in or not, they can perform the following actions:

- Search for books in the library.
- View the list of books in the library.
- View the list of Popular books/New Arrivals in the library.
- Read FAQs of the library.

In case the user is logged in, they can log out of the system. If the user is not logged in, they can log in to the system.

## Product Functions

### Use Cases related to user management

#### Login

- **Actor:** User
- **Description:** The user logs in to the system.
- **Preconditions:** The user is not logged in.
- **Postconditions:** The user is logged in.
- **Normal Flow:**
  1. The user enters their username and password.
  2. The user clicks on the login button.
  3. The system verifies the credentials.
  4. The system logs in the user.
  5. The system displays the home page of the user.
- **Alternative Flows:**
  - The user enters incorrect credentials.
    1. The system displays an error message.
    2. The system displays the login page again.
  - The user clicks on the forgot password link.
    1. The system displays a form to enter the email address.
    2. The user enters the email address.
    3. The system sends a password reset link to the email address.
    4. The user clicks on the password reset link.
    5. The system displays a form to enter the new password.
    6. The user enters the new password.
    7. The system resets the password.
    8. The system displays the login page.
  - The user clicks on the cancel button.
    1. The system displays the home page of a non-logged in user.

#### Logout

- **Actor:** User
- **Description:** The user logs out of the system.
- **Preconditions:** The user is logged in.
- **Postconditions:** The user gets logged out.

- **Normal Flow:**
  1. The user clicks on the logout button.
  2. The system logs out the user.
  3. The system displays the home page of a non-logged in user.

## View Profile

- **Actor:** User
- **Description:** The user views their profile.
- **Preconditions:** The user is logged in.
- **Postconditions:** The user views their profile and statistics related to their account.
- **Normal Flow:**
  1. The user clicks on the profile button.
  2. The system displays the profile page of the user.

## Edit Profile

- **Actor:** User
- **Description:** The user edits their profile.
- **Preconditions:** The user is logged in.
- **Postconditions:** The user changes details of their profile.
- **Normal Flow:**
  1. The user clicks on the profile button.
  2. The system displays the profile page of the user.
  3. The user clicks on the edit profile button.
  4. The system displays a form to edit the profile.
  5. The user edits the profile.
  6. The user clicks on the save button.
  7. The system saves the changes.
  8. The system displays the profile page of the user.
- **Alternative Flows:**
  - The user clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the profile page of the user.
  - The user enter invalid data.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the edit profile page again.

## Use Cases related to administration

### Add new member

- **Actor:** Admin
- **Description:** The admin adds a new member to the system.
- **Preconditions:** The admin is logged in.
- **Postconditions:** The new member is added to the system.
- **Normal Flow:**
  1. The admin clicks on the add member button.
  2. The system displays a form to add a new member.
  3. The admin enters the details of the new member.
  4. The admin clicks on the save button.
  5. The system saves the details of the new member.
  6. The system displays the list of members.
- **Alternative Flows:**
  - The admin clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the list of members.
  - The admin enter invalid data.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the add member page again.
  - The admin enters data that already exists.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the add member page again.

### Remove member

- **Actor:** Admin
- **Description:** The admin removes a member from the system.
- **Preconditions:** The admin is logged in.
- **Postconditions:** The member is removed from the system.
- **Normal Flow:**
  1. The admin clicks on the remove member button.
  2. The system displays a form to remove a member.
  3. The admin enters the details of the member to be removed.
  4. The admin clicks on the save button.
  5. The system removes the member.
  6. The system displays the list of members.

- **Alternative Flows:**
  - The admin clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the list of members.
  - The admin enter invalid data.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the remove member page again.

## Update member

- **Actor:** Admin
- **Description:** The admin updates the details of a member.
- **Preconditions:** The admin is logged in.
- **Postconditions:** The details of the member are updated.
- **Normal Flow:**
  1. The admin clicks on the update member button.
  2. The system displays a form to update the details of a member.
  3. The admin enters the details of the member to be updated.
  4. The admin clicks on the save button.
  5. The system updates the details of the member.
  6. The system displays the list of members.
- **Alternative Flows:**
  - The admin clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the list of members.
  - The admin enter invalid data.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the update member page again.

## Add a fine

- **Actor:** Librarian
- **Description:** The librarian adds a fine to a member.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The fine is added to the member.
- **Normal Flow:**
  1. The librarian clicks on the add fine button.
  2. The system displays a form to add a fine.



3. The librarian enters the details of the fine.
  4. The librarian clicks on the save button.
  5. The system adds the fine to the member.
  6. The system displays the list of fines.
- **Alternative Flows:**
    - The librarian clicks on the cancel button.
      1. The system does not save the changes.
      2. The system displays the list of fines.
    - The librarian enter invalid data.
      1. The system displays an error message.
      2. The system does not save the changes.
      3. The system displays the add fine page again.

## Remove a fine

- **Actor:** Librarian
- **Description:** The librarian removes a fine from a member.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The fine is removed from the member.
- **Normal Flow:**
  1. The librarian clicks on the remove fine button.
  2. The system displays a form to remove a fine.
  3. The librarian enters the details of the fine to be removed.
  4. The librarian clicks on the save button.
  5. The system removes the fine from the member.
  6. The system displays the list of fines.
- **Alternative Flows:**
  - The librarian clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the list of fines.
  - The librarian enter invalid data.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the remove fine page again.

## Update a fine

- **Actor:** Librarian
- **Description:** The librarian updates the details of a fine.
- **Preconditions:** The librarian is logged in.

- **Postconditions:** The details of the fine are updated.
- **Normal Flow:**
  1. The librarian clicks on the update fine button.
  2. The system displays a form to update the details of a fine.
  3. The librarian enters the details of the fine to be updated.
  4. The librarian clicks on the save button.
  5. The system updates the details of the fine.
  6. The system displays the list of fines.
- **Alternative Flows:**
  - The librarian clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the list of fines.
  - The librarian enter invalid data.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the update fine page again.

## Pay a fine

- **Actor:** User
- **Description:** The user pays a fine.
- **Preconditions:** The user is logged in.
- **Postconditions:** The fine is paid.
- **Normal Flow:**
  1. The user clicks on the pay fine button.
  2. The system displays a form to pay a fine.
  3. The user pays the fine using a payment method.
  4. The system verifies the payment.
  5. The system marks the fine as paid.
  6. The system displays the list of fines.
- **Alternative Flows:**
  - The user clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the list of fines.
  - The payment fails.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the pay fine page again.

## View Fines

- **Actor:** User
- **Description:** The user views the list of fines.
- **Preconditions:** The user is logged in.
- **Postconditions:** The list of fines is displayed.
- **Normal Flow:**
  1. The user clicks on the fines button.
  2. The system displays the list of fines.

## Use Cases related to books/journals/e-resources

### Add new book

- **Actor:** Librarian
- **Description:** The librarian adds a new book to the system.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The new book is added to the system.
- **Normal Flow:**
  1. The librarian clicks on the add book button.
  2. The system displays a form to add a new book.
  3. The librarian enters the details of the new book.
  4. The librarian clicks on the save button.
  5. The system saves the details of the new book.
  6. The system displays the list of books.
- **Alternative Flows:**
  - The librarian clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the list of books.
  - The librarian enter invalid data.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the add book page again.
  - The librarian enters data that already exists.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the add book page again.

### Remove book

- **Actor:** Librarian
- **Description:** The librarian removes a book from the system.
- **Preconditions:** The librarian is logged in.

- **Postconditions:** The book is removed from the system.
- **Normal Flow:**
  1. The librarian clicks on the remove book button.
  2. The system displays a form to remove a book.
  3. The librarian enters the details of the book to be removed.
  4. The librarian clicks on the save button.
  5. The system removes the book.
  6. The system displays the list of books.
- **Alternative Flows:**
  - The librarian clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the list of books.
  - The librarian enter invalid data.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the remove book page again.

## Update book

- **Actor:** Librarian
- **Description:** The librarian updates the details of a book.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The details of the book are updated.
- **Normal Flow:**
  1. The librarian clicks on the update book button.
  2. The system displays a form to update the details of a book.
  3. The librarian enters the details of the book to be updated.
  4. The librarian clicks on the save button.
  5. The system updates the details of the book.
  6. The system displays the list of books.
- **Alternative Flows:**
  - The librarian clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the list of books.
  - The librarian enter invalid data.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the update book page again.

## Add new book copy

- **Actor:** Librarian
- **Description:** The librarian adds a new book copy to the system.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The new book copy is added to the system.
- **Normal Flow:**
  1. The librarian clicks on the add book copy button.
  2. The system displays a form to add a new book copy.
  3. The librarian enters the details of book that the copy belongs to.
  4. The librarian clicks on the save button.
  5. The system saves the details of the new book copy.
  6. The system updates the number of copies of the book.
  7. The system displays the home page of the librarian.
- **Alternative Flows:**
  - The librarian clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the home page of the librarian.
  - The librarian enter invalid data.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the add book copy page again.

## Remove book copy

- **Actor:** Librarian
- **Description:** The librarian removes a book copy from the system.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The book copy is removed from the system.
- **Normal Flow:**
  1. The librarian clicks on the remove book copy button.
  2. The system displays a form to remove a book copy.
  3. The librarian enters the details of the book copy to be removed.
  4. The librarian clicks on the save button.
  5. The system removes the book copy.
  6. The system updates the number of copies of the book.
  7. The system displays the home page of the librarian.
- **Alternative Flows:**
  - The librarian clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the home page of the librarian.
  - The librarian enter invalid data.

1. The system displays an error message.
2. The system does not save the changes.
3. The system displays the remove book copy page again.

### Add new E-Resource

- **Actor:** Librarian
- **Description:** The librarian adds a new E-Resource to the system.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The new E-Resource is added to the system.
- **Normal Flow:**
  1. The librarian clicks on the add E-Resource button.
  2. The system displays a form to add a new E-Resource.
  3. The librarian enters the details of the new E-Resource.
  4. The librarian clicks on the save button.
  5. The system saves the details of the new E-Resource.
  6. The system displays the list of E-Resources.
- **Alternative Flows:**
  - The librarian clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the list of E-Resources.
  - The librarian enter invalid data.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the add E-Resource page again.
  - The librarian enters data that already exists.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the add E-Resource page again.

### Remove E-Resource

- **Actor:** Librarian
- **Description:** The librarian removes a E-Resource from the system.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The E-Resource is removed from the system.
- **Normal Flow:**
  1. The librarian clicks on the remove E-Resource button.
  2. The system displays a form to remove a E-Resource.
  3. The librarian enters the details of the E-Resource to be removed.

4. The librarian clicks on the save button.
  5. The system removes the E-Resource.
  6. The system displays the list of E-Resources.
- **Alternative Flows:**
    - The librarian clicks on the cancel button.
      1. The system does not save the changes.
      2. The system displays the list of E-Resources.
    - The librarian enter invalid data.
      1. The system displays an error message.
      2. The system does not save the changes.
      3. The system displays the remove E-Resource page again.

## Update E-Resource

- **Actor:** Librarian
- **Description:** The librarian updates the details of a E-Resource.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The details of the E-Resource are updated.
- **Normal Flow:**
  1. The librarian clicks on the update E-Resource button.
  2. The system displays a form to update the details of a E-Resource.
  3. The librarian enters the details of the E-Resource to be updated.
  4. The librarian clicks on the save button.
  5. The system updates the details of the E-Resource.
  6. The system displays the list of E-Resources.
- **Alternative Flows:**
  - The librarian clicks on the cancel button.
    1. The system does not save the changes.
    2. The system displays the list of E-Resources.
  - The librarian enter invalid data.
    1. The system displays an error message.
    2. The system does not save the changes.
    3. The system displays the update E-Resource page again.

## View Book Catalog

- **Actor:** User
- **Description:** The user views the book catalog of the library.
- **Preconditions:** None
- **Postconditions:** The book catalog is displayed.

- **Normal Flow:**
  1. The user clicks on the catalog button.
  2. The user clicks on the book catalog button.
  3. The system displays the book catalog.

### View E-Resource Catalog

- **Actor:** User
- **Description:** The user views the E-Resource catalog of the library.
- **Preconditions:** None
- **Postconditions:** The E-Resource catalog is displayed.
- **Normal Flow:**
  1. The user clicks on the catalog button.
  2. The user clicks on the E-Resource catalog button.
  3. The system displays the E-Resource catalog.

### Search for Book/E-Resource/Journal

- **Actor:** User
- **Description:** The user searches for a book in the library.
- **Preconditions:** None
- **Postconditions:** The list of books matching the search query is displayed.
- **Normal Flow:**
  1. The user clicks on the catalog button.
  2. The user enters the search query.
  3. The user clicks on the search button.
  4. The system displays the list of items matching the search query.

### View New Arrivals

- **Actor:** User
- **Description:** The user views the list of new arrivals in the library.
- **Preconditions:** None
- **Postconditions:** The list of new arrivals is displayed.
- **Normal Flow:**
  1. The user clicks on the catalog button.
  2. The user clicks on the new arrivals button.
  3. The system displays the list of new arrivals.

### View Popular Books

- **Actor:** User



- **Description:** The user views the list of popular books in the library.
- **Preconditions:** None
- **Postconditions:** The list of popular books is displayed.
- **Normal Flow:**
  1. The user clicks on the catalog button.
  2. The user clicks on the popular books button.
  3. The system displays the list of popular books.

## View Book/E-Resource/Journal Details

- **Actor:** User
- **Description:** The user views the details of a book in the library.
- **Preconditions:** None
- **Postconditions:** The details of the book are displayed.
- **Normal Flow:**
  1. The user clicks on the catalog button.
  2. The user selects a item.
  3. The user clicks on the item details button.
  4. The system displays the details of the item.

## Issue Book

- **Actor:** Librarian
- **Description:** The librarian issues a book to a member.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The book is issued to the member.
- **Normal Flow:**
  1. The librarian clicks on the issue book button.
  2. The system displays all the issue requests by the members.
  3. The librarian enters the details of the book to be issued and the member to whom the book is to be issued.
  4. The librarian clicks on the save button.
  5. The system issues the book to the member.
  6. The system displays the list of books issued by the member.
- **Alternative Flows:**
  - The librarian clicks on the cancel button.
    1. The system does not issue the book.
    2. The system displays the list of books issued by the member.
  - The librarian enter invalid data.
    1. The system displays an error message.

2. The system does not issue the book.
3. The system displays the issue book page again.

## Update Book Issue Details

- **Actor:** Librarian
- **Description:** The librarian updates the details of a book issue.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The details of the book issue are updated.
- **Normal Flow:**
  1. The librarian clicks on the update book issue button.
  2. The system displays a form to update the details of a book issue.
  3. The librarian enters the details of the book issue to be updated.
  4. The librarian clicks on the save button.
  5. The system updates the details of the book issue.
  6. The system displays the list of books issued by the member.
- **Alternative Flows:**
  - The librarian clicks on the cancel button.
    1. The system does not update the book issue.
    2. The system displays the list of books issued by the member.
  - The librarian enter invalid data.
    1. The system displays an error message.
    2. The system does not update the book issue.
    3. The system displays the update book issue page again.

## Request Book Issue

- **Actor:** User
- **Description:** The user request a book issue from the library.
- **Preconditions:** The user is logged in and the book is available.
- **Postconditions:** A request is sent to the librarian to issue the book.
- **Normal Flow:**
  1. The user clicks on the request book issue button.
  2. The system displays a form to request book issue.
  3. The user enters the details of the book to be requested.
  4. The user clicks on the save button.
  5. The system sends a request to the librarian to issue the book.
  6. The system displays the list of requested books issues by the user.
- **Alternative Flows:**
  - The user clicks on the cancel button.

1. The system does not send the request.
2. The system displays the list of requested books issues by the user.
- The user enter invalid data.
  1. The system displays an error message.
  2. The system does not send the request.
  3. The system displays the request book issue page again.
- The user has already been issued the book.
  1. The system displays an error message.
  2. The system does not send the request.
  3. The system displays the request book issue page again.
- The user has already been issued the maximum number of books.
  1. The system displays an error message.
  2. The system does not send the request.
  3. The system displays the request book issue page again.
- The user has pending fines.
  1. The system displays an error message.
  2. The system does not send the request.
  3. The system displays the pay fine page.

## Reserve Book

- **Actor:** User
- **Description:** The user reserves a book from the library.
- **Preconditions:** The user is logged in and the book is not available.
- **Postconditions:** A request is sent to the librarian to reserve the book.
- **Normal Flow:**
  1. The user clicks on the reserve book button.
  2. The system displays a form to reserve a book.
  3. The user enters the details of the book to be reserved.
  4. The user clicks on the save button.
  5. The system sends a request to the librarian to reserve the book.
  6. The system displays the list of books reserved by the user.
- **Alternative Flows:**
  - The user clicks on the cancel button.
    1. The system does not send the request.
    2. The system displays the list of books reserved by the user.
  - The user enter invalid data.
    1. The system displays an error message.
    2. The system does not send the request.
    3. The system displays the reserve book page again.

- The user has already reserved the book.
  1. The system displays an error message.
  2. The system does not send the request.
  3. The system displays the reserve book page again.
- The user has already reserved the maximum number of books.
  1. The system displays an error message.
  2. The system does not send the request.
  3. The system displays the reserve book page again.
- The user has already been issued the book.
  1. The system displays an error message.
  2. The system does not send the request.
  3. The system displays the reserve book page again.
- The user has already been issued the maximum number of books.
  1. The system displays an error message.
  2. The system does not send the request.
  3. The system displays the reserve book page again.
- The user has pending fines.
  1. The system displays an error message.
  2. The system does not send the request.
  3. The system displays the pay fine page.
- E-Resource of the book is available.
  1. The system displays that the E-Resource is available.
  2. The system asks the user if they want to read the E-Resource.
  3. Based on the user's response, the system either sends a request to the librarian to issue the book or displays the list of books reserved by the user.

## Return Book

- **Actor:** User
- **Description:** The user returns a book to the library.
- **Preconditions:** The user is logged in and the book is issued to the user.
- **Postconditions:** A request is sent to the librarian that the book is to be returned.
- **Normal Flow:**
  1. The user clicks on the return book button.
  2. The system displays a form to return a book.
  3. The user enters the details of the book to be returned.
  4. The user clicks on the save button.
  5. The system sends a request to the librarian that the book is to be returned.
  6. The system displays the list of books returned by the user.
- **Alternative Flows:**

- The user clicks on the cancel button.
  1. The system does not send the request.
  2. The system displays the list of books returned by the user.
- The user enter invalid data.
  1. The system displays an error message.
  2. The system does not send the request.
  3. The system displays the return book page again.
- The date of return is after the due date.
  1. The request is sent to the librarian.
  2. The system displays an error message that a fine is to be paid.
  3. The system asks the user if they want to pay the fine.
  4. If the user clicks on the yes button, the system displays the pay fine page.

## Request Book

- **Actor:** User
- **Description:** The user requests for a book to be added to the library.
- **Preconditions:** The user is logged in.
- **Postconditions:** A request is sent to the librarian to add the book.
- **Normal Flow:**
  1. The user clicks on the request book button.
  2. The system displays a form to request a book.
  3. The user enters the details of the book to be requested.
  4. The user clicks on the save button.
  5. The system sends a request to the librarian to add the book.
  6. The system displays the list of books requested by the user.
- **Alternative Flows:**
  - The user clicks on the cancel button.
    1. The system does not send the request.
    2. The system displays the list of books requested by the user.
  - The user enter invalid data.
    1. The system displays an error message.
    2. The system does not send the request.
    3. The system displays the request book page again.
  - The user has already requested the book.
    1. The system displays an error message.
    2. The system does not send the request.
    3. The system displays the request book page again.
  - The user has pending fines.
    1. The system displays an error message.

2. The system does not send the request.
3. The system displays the pay fine page.

### **Request E-Resource/Journal**

- **Actor:** User
- **Description:** The user requests for a E-Resource/Journal to be added to the library.
- **Preconditions:** The user is logged in.
- **Postconditions:** A request is sent to the librarian to add the E-Resource/Journal.
- **Normal Flow:**
  1. The user clicks on the request E-Resource/Journal button.
  2. The system displays a form to request a E-Resource/Journal.
  3. The user enters the details of the E-Resource/Journal to be requested.
  4. The user clicks on the save button.
  5. The system sends a request to the librarian to add the E-Resource/Journal.
  6. The system displays the list of E-Resources/Journals requested by the user.
- **Alternative Flows:**
  - The user clicks on the cancel button.
    1. The system does not send the request.
    2. The system displays the list of E-Resources/Journals requested by the user.
  - The user enter invalid data.
    1. The system displays an error message.
    2. The system does not send the request.
    3. The system displays the request E-Resource/Journal page again.
  - The user has already requested the E-Resource/Journal.
    1. The system displays an error message.
    2. The system does not send the request.
    3. The system displays the request E-Resource/Journal page again.
  - The user has pending fines.
    1. The system displays an error message.
    2. The system does not send the request.
    3. The system displays the pay fine page.

### **Read E-Resource/Journal**

- **Actor:** User
- **Description:** The user reads a E-Resource/Journal.
- **Preconditions:** The user is logged in.
- **Postconditions:** The E-Resource/Journal is read by the user.
- **Normal Flow:**

1. The user clicks on the read E-Resource/Journal button.
  2. The system displays a form to read a E-Resource/Journal.
  3. The user enters the details of the E-Resource/Journal to be read.
  4. The user clicks on the save button.
  5. The system displays the E-Resource/Journal.
- **Alternative Flows:**
    - The user clicks on the cancel button.
      1. The system does not display the E-Resource/Journal.
      2. The system displays the home page of the user.
    - The user enter invalid data.
      1. The system displays an error message.
      2. The system does not display the E-Resource/Journal.
      3. The system displays the read E-Resource/Journal page again.
    - The user does not have access to the E-Resource/Journal.
      1. The system displays an error message.
      2. The system does not display the E-Resource/Journal.
      3. The system displays the home page of the user.
    - The user has pending fines.
      1. The system displays an error message.
      2. The system does not send the request.
      3. The system displays the pay fine page.

### **View All Book Issues/Returns/Reservations/Requests of a Member**

- **Actor:** User
- **Description:** The user views the list of all book issues/returns/reservations/requests of the user.
- **Preconditions:** The user is logged in.
- **Postconditions:** The list of all book issues/returns/reservations/requests of the user are displayed.
- **Normal Flow:**
  1. The user clicks on the view all button.
  2. The system displays a form to view all book issues/returns/reservations/requests of the user.
  3. The user selects the type of book issues/returns/reservations/requests.
  4. The user clicks on the save button.
  5. The system displays the list of all book issues/returns/reservations/requests of the user.

### **View All Requests**

- **Actor:** Librarian
- **Description:** The librarian views the list of all requests.
- **Preconditions:** The librarian is logged in.

- **Postconditions:** The list of all requests are displayed.
- **Normal Flow:**
  1. The librarian clicks on the view all requests button.
  2. The system displays the list of all requests.

## Update Request

- **Actor:** Librarian
- **Description:** The librarian updates the details of a request.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The details of the request are updated.
- **Normal Flow:**
  1. The librarian clicks on the update request button.
  2. The system displays a form to update the details of a request.
  3. The librarian enters the details of the request to be updated.
  4. The librarian clicks on the save button.
  5. The system updates the details of the request.
  6. The system displays the list of all requests.
- **Alternative Flows:**
  - The librarian clicks on the cancel button.
    1. The system does not update the request.
    2. The system displays the list of all requests.
  - The librarian enter invalid data.
    1. The system displays an error message.
    2. The system does not update the request.
    3. The system displays the update request page again.

## View All Unresolved Issues/Returns/Reservations/Requests

- **Actor:** Librarian
- **Description:** The librarian views the list of all unresolved book issues/returns/reservations/requests.
- **Preconditions:** The librarian is logged in.
- **Postconditions:** The list of all unresolved book issues/returns/reservations/requests are displayed.
- **Normal Flow:**
  1. The librarian clicks on the view all unresolved button.
  2. The system displays a form to view all unresolved book issues/returns/reservations/requests.
  3. The librarian selects the type of book issues/returns/reservations/requests.
  4. The librarian clicks on the save button.
  5. The system displays the list of all unresolved book issues/returns/reservations/requests.



## Miscellaneous Use Cases

### View Information/FAQs (Frequently Asked Questions)

- **Actor:** User
- **Description:** The user views the information/FAQs of the library.
- **Preconditions:** None
- **Postconditions:** Information/FAQs of the library are displayed.
- **Normal Flow:**
  1. The user clicks on the information/FAQs button.
  2. The system displays the information/FAQs page.

### View Contact Us

- **Actor:** User
- **Description:** The user views the contact us page of the library.
- **Preconditions:** None
- **Postconditions:** The contact us page of the library is displayed.
- **Normal Flow:**
  1. The user clicks on the contact us button.
  2. The system displays the contact us page.

### View About Us

- **Actor:** User
- **Description:** The user views the about us page of the library.
- **Preconditions:** None
- **Postconditions:** The about us page of the library is displayed.
- **Normal Flow:**
  1. The user clicks on the about us button.
  2. The system displays the about us page.

# Non-Functional Requirements/Constraints/Assumptions

## Constraints

- The system should be secure and only authorized users should be able to access the main functions of library management system.
- The system should be fault tolerant and should be able to recover from failures.

## Assumptions

- The system will be run on a updated web browser.
- Admins and librarians will be trained to use the system.

## Performance Requirements

- The system should be able to handle a large number of users at the same time.
- The system should have a good response time.
- The system should be easy to use.

## Appendix

## References

Note that the following references are clickable links.

- Library Management System
- Existing SRS such as:
  - <https://www.studocu.com/in/document/national-institute-of-technology-rourkela/software-engineering/library-management-system-srs/7138104>
  - <https://www.slideshare.net/Chetan2608/software-requirements-specification-of-library-management-system>