

SRS for Library Management System

Group 14

Index

- Index
- Introduction
 - Overview
 - Purpose
 - Scope
- Overall Description
 - Product Perspective
 - 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
- Assumptions and Dependencies
- Constraints
- Specific Requirements
- Acceptance Criteria
- Appendix
 - References

Introduction

Overview

The Library Management System is a software system that will automate the processes involved in the management of a library.

Section Purpose gives a brief description of the purpose of the system.

Section Scope gives a brief description of the scope of the system, including the stakeholders and the features of the system.

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, the interfaces 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

Overall Description

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.

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

Faculty

The faculty 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 to be added to the library.
- Reserve books in advance when they are not available.

Student

The 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 to be added to the library.
- Reserve books in advance when they are not available.

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.

Assumptions and Dependencies

Constraints

Specific Requirements

Acceptance Criteria

Appendix

References