**Problem Statement**

Libraries play an important role in society by providing access to knowledge and information. However, managing the operations of a library can be a complex and time-consuming task, especially in larger libraries with a wide variety of resources and patrons. The traditional manual method of managing a library is often inefficient and prone to errors. Therefore, there is a need for a reliable and efficient library management system that can automate the library's operations and make it easier for staff to manage the library's resources. Apart from books we also have other issues in library that needs to be managed such as managing the staff working in library and security system needs to be managed properly.

# **Software Requirement Specification (SRS)**

# **Introduction**:

## **Purpose of this Document:** The purpose of a library management system (LMS) is to provide a comprehensive and efficient solution for managing various tasks in a library. This includes tasks such as managing library resources, tracking borrowing and returning of books, maintaining borrower information, generating reports, and more. A library management system aims to automate these tasks, streamline processes, and improve the overall efficiency of a library.

## **Scope of this document** – The scope of a library management system includes various functions and features that help in managing the day-to-day operations of a library. This includes managing the catalogue of library resources, tracking the borrowing and returning of books, maintaining borrower information and history, generating reports, and more. The system may also provide features such as online reservations, online renewal of books, integration with external systems, and more. The scope of the system may vary depending on the needs of the library.

## **Overview** –A library management system is a software solution that provides a centralized platform for managing various tasks in a library. The system typically includes a catalogue of library resources, which can be searched and accessed by users. Borrowers can use the system to search for books, reserve books, and renew books online. The system also tracks the borrowing and returning of books, maintains borrower information, generates reports, and more. Overall, a library management system aims to automate and streamline library operations, improve efficiency, and provide a better user experience for library patrons.

# **General description:** The library management system is a software application designed to manage the various tasks involved in running a library. The system provides a centralized platform for managing library resources, tracking borrowing and returning of books, maintaining borrower information, generating reports, and more. The system is designed to automate and streamline library operations, improving efficiency and providing a better user experience for library patrons.

# **Functional Requirements:** The library management system must include the following functional requirements:

# Catalogue Management: The system must have the ability to manage the catalogue of library resources, including adding, editing, and deleting books.

# Borrower Management: The system must maintain borrower information and history, including adding and deleting borrower records, tracking borrowing and returning of books, and generating overdue notices.

# Reservation Management: The system must allow borrowers to reserve books online and track reservations.

# Renewal Management: The system must allow borrowers to renew books online and track renewals.

# Reporting: The system must generate reports, including overdue books, borrowed books, and reserved books.

# **Interface Requirements:** The library management system must meet the following interface requirements:

# User-Friendly Interface: The system must have a user-friendly interface that is easy to navigate and use.

# Mobile Compatibility: The system must be compatible with mobile devices, allowing borrowers to access the system from their mobile devices.

# Integration with External Systems: The system must integrate with external systems such as payment gateways and book suppliers.

# **Performance Requirements:** The library management system must meet the following performance requirements:

# Response Time: The system must respond quickly to user requests, providing a fast and efficient experience for library patrons.

# Scalability: The system must be scalable, able to handle a large number of users and books as the library grows.

# Security: The system must ensure the security of borrower information and prevent unauthorized access to the system.

# **Design Constraints:**

# The library management system must be user-friendly and accessible to individuals with limited technical knowledge.

# The system must be able to handle a large number of users simultaneously.

# The system must be secure and protect user data.

# The system must be able to support multiple languages.

# The system must be scalable and able to accommodate future growth.

# The system must be able to integrate with existing library systems, such as barcode scanners and bookshelves.

# The system must be able to handle different types of materials, such as books, magazines, and multimedia.

# **Non-Functional Attributes:**

# Performance: The system must be able to handle a high volume of transactions with minimal latency.

# Availability: The system must be available 24/7 to users.

# Reliability: The system must be reliable and minimize the risk of data loss.

# Security: The system must protect user data and ensure confidentiality.

# Usability: The system must be intuitive and easy to use for library staff and patrons.

# Maintainability: The system must be easy to maintain and upgrade over time.

# **Preliminary Schedule and Budget:**

# Phase 1: Requirements Gathering (1 month)Define project scope and requirementsDevelop use cases and user storiesDefine system architecture

# Phase 2: Design (2 months)Develop system design and architectureDevelop data model and schemaDefine system interfaces and APIs

# Phase 3: Development (4 months)Develop system components and modulesDevelop user interfacePerform unit testing and system integration testing

# Phase 4: Testing (1 month)Conduct functional testingConduct non-functional testing (performance, security, usability, etc.)Conduct user acceptance testing

# Phase 5: Deployment (1 monthDeploy system to production environmenConfigure and optimize system performanceTrain library staff on system useTotal time: 9 months