Software Requirements Specification (SRS)

Library Management System

Version 1.0
Date: 28-02-2025

Prepared by:

V. Sai Tarun	[SE22UCSE236]	se22ucse236@mahindrauniversity.edu.in
P. Jagadeep	[SE22UCSE204]	se22ucse204@mahindrauniversity.edu.in
S. Rohit Reddy	[SE22UCSE241]	se22ucse241@mahindrauniversity.edu.in
Tejas Varma	[SE22UCSE205]	se22ucse205@mahindrauniversity.edu.in
Rishi V	[SE22UCSE221]	se22ucse221@mahindrauniversity.edu.in
Anish P	[SE22UCSE211]	se22ucse211@mahindrauniversity.edu.in
E. Samprith	[SE22UCSE239]	se22ucse239@mahindrauniversity.edu.in

Instructor: Vijay Rao Duddu
Course: Software Engineering
Lab Section:
Teaching Assistant: Sravanthi
Date: 28-02-2025

---

1. Introduction

1.1 Document Purpose

This document specifies the software requirements for the **Library Management System – Book Issue and Deposit**. The system aims to automate library operations such as book borrowing, returning, and fine calculation. The document serves as a reference for developers, testers, and stakeholders to understand system functionality and constraints.

1.2 Product Scope

The **mobile application** offers an efficient solution for managing library books seamlessly. It enables students and faculty to borrow and return books digitally while providing real-time tracking of book availability. By reducing manual work, improving accuracy, and enhancing the user experience, the app ensures a streamlined and user-friendly library management system for both members and administrators.

1.3 Intended Audience and Document Overview

This document is intended for:

- **Developers** – To understand system requirements and implementation constraints.
- **Testers** – To verify system functionality.
- **Project Managers** – To track progress.
- **Clients (University Library)** – To review system features and expectations.

1.4 Definitions, Acronyms, and Abbreviations

- **LMS** – Flutter based Library Management System
- **UI** – User Interface
- **Firebase:** Used for Backend and data management
- **Flutter**: Develops a **cross-**platform mobile application

1.5 Document Conventions

This document follows the IEEE SRS format. Standard fonts include Arial size 11 or 12. Headings are bolded, and section numbering follows IEEE recommendations.

1.6 References and Acknowledgments

- **IEEE Software Engineering Standards**

---

2. Overall Description

2.1 Product Overview

The Library Management System is a **self-contained mobile application** designed to automate the book lending process. It enables users to log in using university credentials, borrow/return books, and view their transaction history. Librarians can manage books, check availability, and track due dates. The backend is built with **Firebase**, and the frontend uses **Flutter**.

2.2 Product Functionality

The system provides the following features:

- **User Authentication** – Secure login for students, faculty, and librarians.
- **Book Issue and Return** – Digital borrowing and deposit tracking.
- **Fine Calculation** – Automatic calculation of overdue fines.
- **Book Search and Filter** – Find books by title, author, or genre.
- **Admin Dashboard** – Librarians can add, update, or remove books.
- **Real-time Book Availability Updates**.

2.3 Design and Implementation Constraints

- The backend must use **Firebase (DART)**.
- The frontend must be developed in **Flutter**.
- **Firebase Realtime Database** will handle data efficiently with cloud storage.
- The system must support up to **1000 concurrent users**.

2.4 Assumptions and Dependencies

- Users must have university credentials for access.
- Book records will be manually updated by librarians.
- The system will primarily be accessed via mobiles.

---

3. Specific Requirements

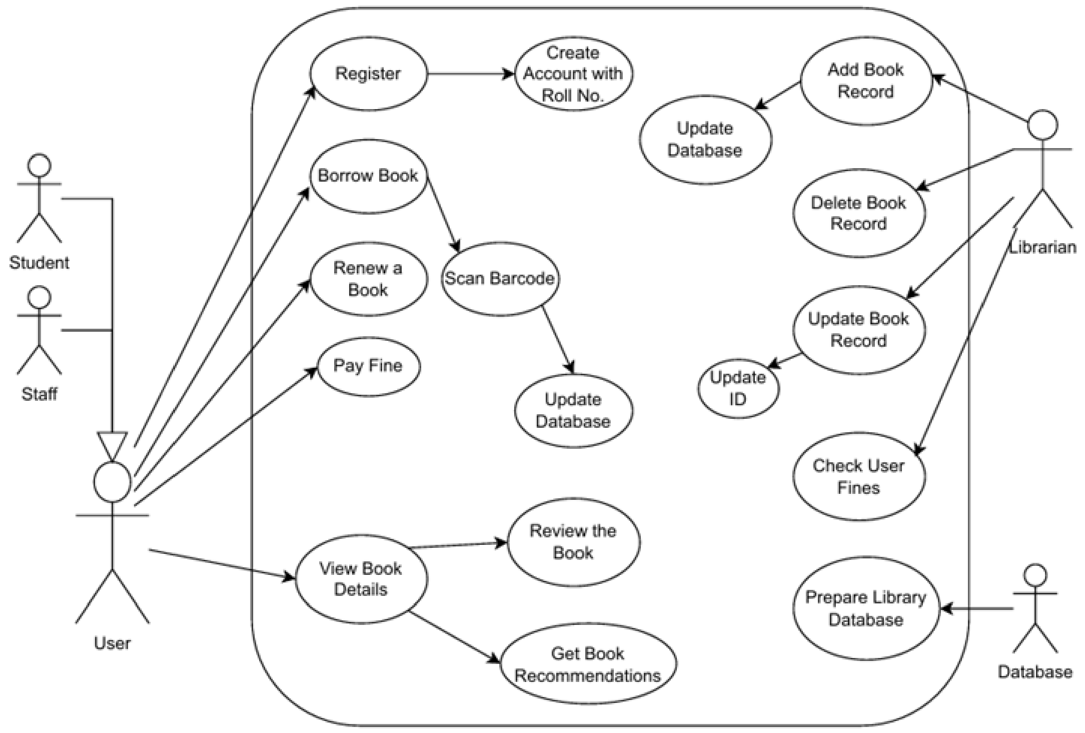
3.1 External Interface Requirements

- **User Interface:** Mobile App, Authentication, Dashboard, Book Search.
- **Mobile Devices –** Android & iOS smartphones with internet access**.**
- **Software Interface**: Frontend: Flutter (Dart), Backend: Firebase, APIs.

3.2 Functional Requirements

1. **User Authentication:** The system shall allow users to log in securely.
2. **Book Borrowing:** Users can borrow books and get a due date.
3. **Book Returning:** Users can return books and clear transactions.
4. **Fine Calculation:** The system shall calculate overdue penalties automatically.
5. **Book Search:** Users can search by title, author, or genre.
6. **Admin Management:** Librarians can manage book inventory.

3.3 Use Case Model



---

4. Other Non-functional Requirements

4.1 Performance Requirements

- The system shall process book transactions within **3 seconds**.
- It should support up to **1000 concurrent users**.

4.2 Safety and Security Requirements

- **User authentication** using university credentials.
- **Data encryption** for sensitive information.
- **Role-based access control** (students cannot modify book records).

4.3 Software Quality Attributes

1. **Reliability:** System uptime of at least **99.5%**.
2. **Usability:** Intuitive UI for ease of navigation.
3. **Scalability:** Supports up to **1000 users**, expandable if needed.

---

5. Other Requirements

- **Data Backup:** Database must be backed up daily.
- **Legal Compliance:** System follows university IT policies.

Appendices

Appendix A – Data Dictionary

S.no	**Field**	**Type**	**Description**
1	User_id	Integer	Unique Id for users
2	Book_id	Integer	Unique book Id
3	Issue_date	date	Book issue date
4	Return_date	date	Book return date
5	Fine	Float	Fine Calculation

Appendix B – Group Log

S.no	Date	Task	Member(s) Responsible
1	21/02/2025	UI/UX design	Rishi
2	14/03/2025	Feature Development	Tarun, Tejas, Rohit
3	07/04/2025	Database System	Anish, Samprith
4	20/04/2025	Testing and debugging	Jagadeep
5	07/05/2025	Mobile Application	All
