A logo with a symbol and text

Description automatically generated

***Helwan University - Faculty of engineering***

***Computer Engineering and Systems Department***

Software Requirements Specification

**Library Management System** (LMS)

Prepared By

Hamsa Ragheb Abdullah Ragheb

Maha Mohamed Hamdy Ibrahim

Mahmoud Hamdy Salem Mohamed

Table of Contents

[**1.** **Preface** 4](#_Toc154564426)

[1.1 Document Purpose 4](#_Toc154564427)

[1.2 Target users 6](#_Toc154564428)

[1.3 Revision History 10](#_Toc154564429)

[**2.** **Introduction** 10](#_Toc154564430)

[2.1 Purpose 10](#_Toc154564431)

[2.2 Scope 11](#_Toc154564432)

[2.3 Overview 11](#_Toc154564433)

[**3.** **Glossary** 12](#_Toc154564434)

[3.1 Acronyms and Abbreviations 12](#_Toc154564435)

[3.2 Definitions 13](#_Toc154564436)

[**4.** **System Users** 15](#_Toc154564437)

[4.1 System Stakeholders 15](#_Toc154564438)

[4.2 Users' Objectives 16](#_Toc154564439)

[**5.** **User Requirements Definition** 17](#_Toc154564440)

[5.1 System Functions 17](#_Toc154564441)

[12.1 Constrains 21](#_Toc154564442)

[**13.** **System architecture** 22](#_Toc154564443)

[**14.** **System functional requirements** 23](#_Toc154564444)

[**15.** **Interface requirements** 30](#_Toc154564445)

[15.1 User Interfaces 30](#_Toc154564446)

[15.2 Hardware Interfaces 30](#_Toc154564447)

[15.3 Software Interfaces 30](#_Toc154564448)

[15.4 Communication Interfaces 30](#_Toc154564449)

[**16.** **Non Functional Requirements** 30](#_Toc154564450)

[16.1 Performance 30](#_Toc154564451)

[16.2 Reliability 31](#_Toc154564452)

[16.3 Scalability 31](#_Toc154564453)

[16.4 Security 31](#_Toc154564454)

[16.5 Compatibility 31](#_Toc154564455)

[16.6 Maintainability 31](#_Toc154564456)

[16.7 Portability 31](#_Toc154564457)

[**17.** **System Models and Diagrams** 32](#_Toc154564458)

[17.1 Use Case Diagram 32](#_Toc154564459)

[17.2 Class Diagram 33](#_Toc154564460)

[**18.** **System Evolution** 34](#_Toc154564461)

[**19.** **Time plan** 34](#_Toc154564462)

[**20.** **Appendices** 34](#_Toc154564463)

[**21.** **References** 34](#_Toc154564464)

Table of Figures

[***Figure 1:*** *Simple System Hierarchy………………………………… 24*](#_Toc153223912)

[*Figure 2: Use Case Diagram – HAS……………………………….. 30*](#_Toc153223913)

[*Figure 3: Class Diagram – HAS…………………………………….. 31*](#_Toc153223914)

# **Preface**

## Document Purpose

The primary objective of this Software Requirements Specification (SRS) document is to delineate, with precision and clarity, the comprehensive requirements, functionalities, and intricacies governing the development of Library Management System (LMS)

The SRS serves as a foundational cornerstone for all stakeholders involved in the project. It encapsulates a vivid depiction of the system's functional and technical specifications, thereby fostering a shared understanding among development teams, project management, quality assurance experts, and clients. By acting as a reference point, it ensures that the development effort remains aligned with the strategic vision while setting a standard of excellence for the project's execution.

In essence, the document purpose may be distilled into the following key objectives:

* **Specification**: To provide an exhaustive and detailed specification of the Smart Home Control System, elucidating its functionalities, behaviors, and performance expectations.
* **Clarity**: To offer an unambiguous, meticulously structured document that leaves no room for misinterpretation, thereby facilitating a seamless comprehension of system requisites.
* **Alignment**: To serve as a reference point and guidepost, ensuring that all stakeholders—developers, project managers, quality assurance teams, and clients—remain attuned to a common vision, thereby fostering a harmonious and unified project execution.
* **Transparency**: To imbue the development process with transparency, by rendering the requisites, constraints, and features in a format that enables all parties involved to make informed decisions and provide informed contributions.
* **Validation**: To provide the quality assurance and testing teams with a definitive source for the validation of system functionality, thereby enabling a rigorous and comprehensive testing process.

In sum, this SRS document stands as a pivotal instrument for the project's success, aiming to bring coherence, transparency, and precision to the development process of the Home Automation System.

## Target users

The comprehensive delineation of the Home Automation System System's target users is a critical component within this Software Requirements Specification (SRS) document. It stands as a fundamental element in ensuring that the document's content aligns with the multifaceted information needs of the diverse array of stakeholders. An exhaustive understanding of these users is pivotal to ensure the SRS effectively addresses their individual and collective requirements. The primary audience segments include :

1. Development and Engineering Personnel:

* **System Architects**:
* Role: System architects are the system's visionaries, responsible for orchestrating its architectural design. They rely on the SRS as the foundational source for validating their designs against documented requirements
* Information Needs: A comprehensive understanding of system functionalities, interfaces, and the integration of hardware and software components
* **Software Engineers**:
* Role: The software development team is tasked with translating design and functional specifications into code. The SRS serves as a cornerstone reference for this transformation
* Information Needs: Detailed insights into system functionalities, technical intricacies, data structures, and algorithms
* **Hardware Engineers**:
* Role: Hardware engineers undertake the design and implementation of the system's physical components, necessitating alignment with software features
* Information Needs: Detailed specifications regarding hardware-software interfaces, component interdependencies, and compliance with overall

system design

* **System Designers**:
* Role: System designers are pivotal in shaping the system's architecture and essential components. They ensure that the system's design aligns with the documented project requirements
* Information Needs: Precise system functionality descriptions and interface specifications
* **System Analysts**:
* Role: System analysts focus on the in-depth analysis of system requirements, translating them into comprehensive system specifications and designs
* Information Needs: Clear and unambiguous system requirements, dependencies, and constraints

1. Project Management

* **Project Managers:**
* Role: Project managers are entrusted with overseeing the project's strategic direction and ensuring the timely delivery of the Home Automation System
* Information Needs: Milestones, dependencies, project schedules, resource allocation, and alignment of project goals with system requirements

1. Quality Assurance and Testing Teams

* **Quality Assurance Specialists:**
* Role: Quality assurance experts formulate test cases, strategies, and validation criteria to ensure the system adheres to stipulated requirements and quality standards
* Information Needs: Detailed system functionalities, quality standards, and performance benchmarks
* **Testers**:
* Role: Testers execute test scenarios, validate system functionality, and report defects, contributing to the system's quality and reliability
* Information Needs: Clarity on system features, use cases, expected behaviors, and potential pitfalls

1. Stakeholders

* **Clients**:
* Role: Clients, often financially invested in the project, rely on the SRS to validate that the system aligns with their business or personal objectives.
* Information Needs: High-level system objectives, business impact, and how the system aligns with their specific goals.

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description of Changes |
| 1.0 | 22-10-2023 | Team Members | Initial draft of the SRS |
| 2.0 | 7-12-2023 | Team Members | Add Use Case Diagram and Class Diagram |

# **Introduction**

## Purpose

The purpose of the Library Management System (LMS) is to provide an efficient and systematic approach to managing the operations of a library. This software solution aims to streamline the process of book issuance, return, and overall library administration. By incorporating user-friendly graphical interfaces and a centralized database, the system aims to enhance the efficiency and effectiveness of library management tasks.

## Scope

The scope of the Library Management System encompasses the entire lifecycle of library operations, from book acquisition and cataloging to student interactions and book return. The system caters to the needs of administrators, librarians, and students, offering distinct functionalities for each user role. Key features include user authentication, book management, student information handling, and tracking the borrowing and returning of books. Additionally, the system extends its scope to facilitate the reservation of library sessions, ensuring optimal service for users

## Overview

The Library Management System is designed to operate within a C# environment, providing a user-friendly Graphical User Interface (GUI) for seamless interaction. The system is centered around a robust database that captures detailed information about books, students, and the transactions between them. Administrators, with the appropriate credentials, can access the system to perform essential tasks such as adding new books, managing student details, issuing and returning books, and searching for specific information.

The system accommodates various user roles, including administrators responsible for overall management and librarians who handle day-to-day library operations. Students benefit from features like reserving library sessions and tracking their borrowed books. The categorization of books and the ability to search for specific items contribute to a more organized and user-friendly experience.

In subsequent sections of this document, each feature of the Library Management System will be elaborated upon, providing detailed specifications and requirements to guide the development process..

# **Glossary**

## Acronyms and Abbreviations

* **SRS**: Software Requirements Specification
* **LMS**: Library Management System
* **DB**: Database
* **Admin**: Administrator
* **ISBN**: International Standard Book Number
* **CRUD**: Create, Read, Update, Delete

## Definitions

* **Library Management System (LMS)**: A software application designed to automate and streamline the processes involved in managing a library, including book cataloging, student information, and transaction tracking.
* **Graphical User Interface (GUI)**: A visual interface that allows users to interact with the Library Management System through graphical elements such as buttons, forms, and windows.
* **Database (DB)**: A structured collection of data organized for efficient retrieval and management. In the context of the LMS, the database stores information about books, students, and transactions.
* **Administrator (Admin)**: A user role with elevated privileges responsible for managing the overall operation of the Library Management System, including adding books, updating student details, and overseeing transactions.
* **International Standard Book Number (ISBN)**: A unique identifier assigned to each edition and variation of a book, facilitating the accurate identification and management of library resources.
* **CRUD Operations**: An acronym for Create, Read, Update, and Delete, representing the basic functions that can be performed on database records. In the context of the LMS, CRUD operations are essential for managing books and student information
* **UI**: User Interface, the visual and interactive components of the software.
* **PWM**: Pulse Width Modulation, a method for controlling fan speed.
* **Desktop Application**: Software for use on personal computers.
* **Performance**: The ability of the system to perform tasks efficiently.
* **Usability**: The user-friendliness and ease of use of the system.
* **Compatibility**: The ability of system components to work together seamlessly.

# **System Users**

## System Stakeholders

In this section, we identify the various user classes, or stakeholders, who will engage with the Smart Home Control System. Each class is described along with their key characteristics:

* **End Users**: These are the individuals who will use the system to control their home environment. They may vary in technical expertise, but all seek convenience and improved home automation.
* **Clients**: Clients are individuals or entities who have a financial stake in the project. They may include those who commissioned the system's development and have specific business or personal objectives related to its performance.
* **Development and Engineering Personnel:** This group encompasses system architects, software engineers, and hardware engineers. They are technically proficient and hold key roles in designing, developing, and implementing the system.
* **Project Managers**: Project managers are responsible for the strategic oversight of the project. They will use the system to ensure project milestones are achieved and resources are allocated effectively.
* **Quality Assurance and Testing Teams**: These professionals are tasked with ensuring the system functions correctly and meets quality standards. They will use the system to validate and verify its performance

## Users' Objectives

For each of the identified stakeholders, the objectives from the Smart Home Control System are defined:

* **End Users**: To have a user-friendly and efficient means of controlling home automation, enhancing convenience and security.
* **Clients**: To confirm that the system aligns with their business or personal objectives and that their investment in the project is justified.
* **Development and Engineering Personnel**: To design, develop, and implement a robust and functional system that aligns with the documented requirements.
* **Project Managers**: To ensure project milestones are achieved, schedules are met, and resources are allocated effectively to deliver the Smart Home Control System successfully.
* **Quality Assurance and Testing Teams**: To validate and verify the system, ensuring it meets the stipulated requirements and quality standards, and to identify and report any defects.

# **User Requirements Definition**

## System Functions

1. Login

**Description**: Users, specifically administrators, must be able to log in to the Library Management System using valid credentials.

**Requirements**:

The system shall authenticate users based on a username and password.

User authentication shall be secure and comply with best practices.

1. Add Book Detail

Description: Administrators should have the capability to add details of new books to the library database.

Requirements:

The system shall provide a user-friendly form for entering book information, including title, author, ISBN, category, and quantity.

ISBN validation shall be performed to ensure uniqueness.

1. View Books Detail

Description: Users, especially administrators, should be able to view a list of all books available in the library.

Requirements:

The system shall display a comprehensive list of books, including details such as title, author, ISBN, category, and available quantity.

Users shall have the option to sort and filter the book list.

1. Update & Delete Book Details

Description: Administrators need the ability to update and delete book details in the library database.

Requirements:

The system shall allow administrators to modify existing book information, such as title, author, and quantity.

An option to delete a book record shall be available, with appropriate confirmation prompts.

1. Add Student Details

Description: Administrators should be able to add and manage student information in the system.

Requirements:

The system shall provide a form for entering student details, including name, student ID, contact information, and program.

Validation checks shall be implemented to ensure accuracy in student data.

1. View Student Detail

Description: Users, especially administrators, should be able to view a list of all registered students.

Requirements:

The system shall display a comprehensive list of students, including details such as name, student ID, contact information, and program.

Users shall have the option to sort and filter the student list.

1. Update & Delete Student Details

Description: Administrators need the ability to update and delete student details in the system.

Requirements:

The system shall allow administrators to modify existing student information, such as contact details and program.

An option to delete a student record shall be available, with appropriate confirmation prompts.

1. Issue Books to Student

Description: Administrators should be able to issue books to students, with a limit of 3 books per student.

Requirements:

The system shall validate the availability of the requested books before issuing them.

A transaction record shall be created for each issued book, including timestamps and student details.

1. Return Books from Student

Description: Administrators should be able to process the return of books from students.

Requirements:

The system shall update the availability of returned books in the database.

Transaction records shall be updated with return timestamps.

1. Search Books

Description: Users, including administrators, should be able to search for specific books based on various criteria.

Requirements:

The system shall provide a search functionality allowing users to find books by title, author, ISBN, or category.

Search results shall be displayed in a clear and organized manner.

1. Search Student

Description: Users, especially administrators, should be able to search for specific students based on various criteria.

Requirements:

The system shall provide a search functionality allowing users to find students by name, student ID, or program.

Search results shall be displayed in a clear and organized manner.

1. Track Borrowed Books per Student

Description: Users, especially administrators, should be able to view the list of students who currently have books checked out.

Requirements:

The system shall display a report showing the names of students, along with the titles of books currently in their possession.

This report shall be accessible to administrators for efficient book tracking.

## Constrains

1. Authentication Constraints

Description: The system shall enforce secure authentication measures to protect user accounts.

Constraints:

User passwords shall be securely hashed and stored.

Failed login attempts shall trigger account lockout mechanisms to prevent unauthorized access.

1. Book Quantity Constraint

Description: The system shall enforce constraints on the quantity of books that can be issued to a single student.

Constraints:

Each student is allowed to borrow a maximum of 3 books at a time.

The system shall prevent the issuance of additional books beyond this limit.

1. Data Validation Constraints

Description: The system shall implement data validation to ensure the accuracy and integrity of information.

Constraints:

ISBN numbers shall be unique for each book.

Student IDs shall be unique for each student.

Input forms shall include validation checks to prevent the entry of invalid or incomplete data.

1. Session Reservation Constraints

Description: The system shall enforce constraints related to the reservation of library sessions.

Constraints:

Students shall be able to reserve library sessions within a specified time frame.

The system shall prevent overlapping session reservations for the same student.

1. Database Administrator Access

Description: Access to the entire system, including the database, shall be restricted to authenticated administrators.

Constraints:

Only users with administrator credentials shall have the authority to add, modify, or delete records in the system.

Regular users shall have read-only access to view information without the ability to perform administrative tasks.

# **System architecture**

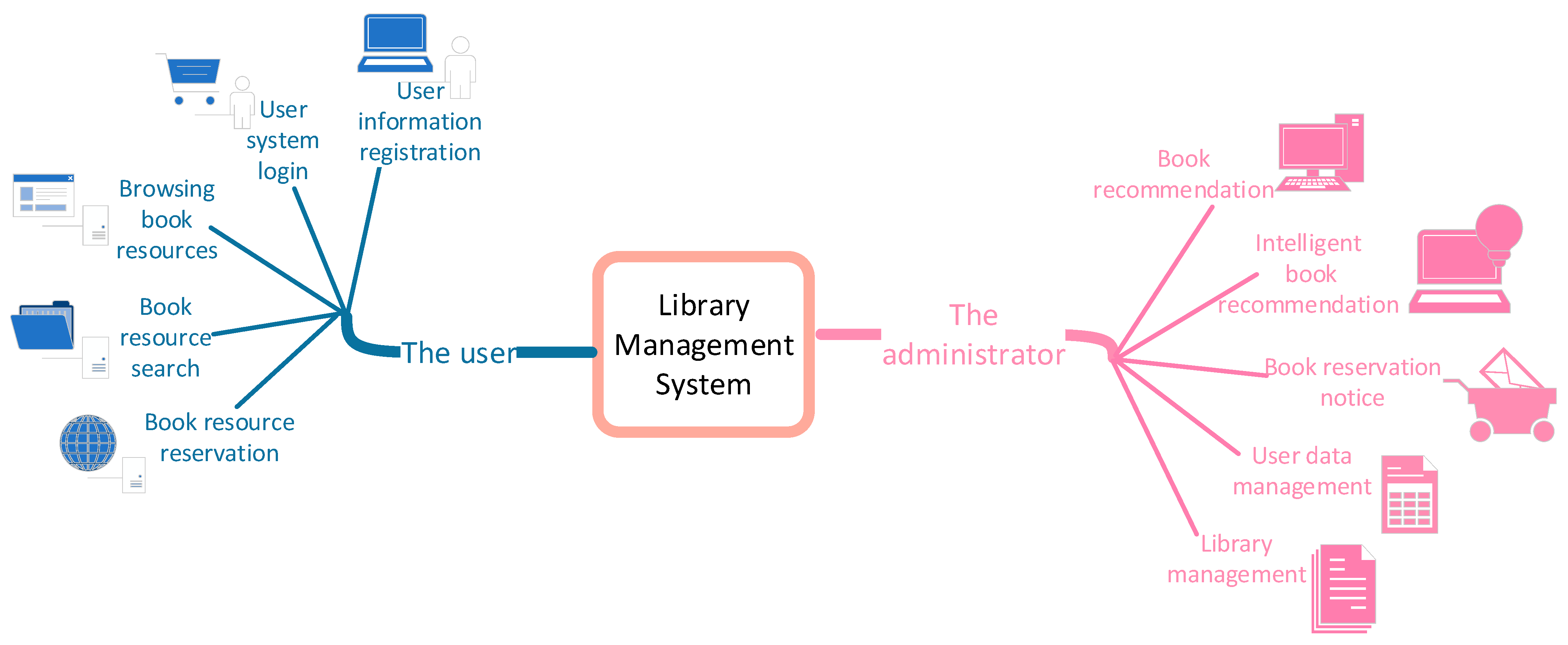


Figure 1 Simple System Hierarchy

# **System functional requirements**

The system functional requirements outline the specific functionalities that the Library Management System (LMS) must possess to meet the needs of its users. These requirements cover essential aspects such as book and student management, transaction processing, and system accessibility.

1. Login

Requirement

Description: The system shall authenticate users to ensure secure access to the LMS.

Functional Requirements:

Users must provide a valid username and password for authentication.

The system shall validate user credentials against stored records.

After successful authentication, users shall be granted access to the system.

1. Acceptance Criteria

Users can successfully log in with valid credentials.

Invalid login attempts trigger appropriate error messages.

Account lockout occurs after a specified number of consecutive failed login attempts.

1. Add Book Detail

14.2.1 Requirement

Description: Administrators shall have the ability to add new books to the library database.

Functional Requirements:

The system shall provide a user-friendly form for administrators to enter book details, including title, author, ISBN, category, and quantity.

ISBN validation shall be performed to ensure uniqueness.

The system shall generate a unique identifier for each book record.

1. Acceptance Criteria

Administrators can successfully add a new book with valid information.

ISBN validation prevents the addition of books with duplicate ISBN numbers.

1. View Books Detail

Requirement

Description: Users, particularly administrators, should be able to view a comprehensive list of all books available in the library.

Functional Requirements:

The system shall display a well-organized list of books, including details such as title, author, ISBN, category, and available quantity.

Users shall have the ability to sort and filter the book list based on various criteria.

1. Acceptance Criteria

The system presents a clear and organized list of all books in the library.

Users can sort the book list by title, author, or category.

Filters allow users to view books based on specific criteria, such as availability.

1. Update & Delete Book Details

Requirement

Description: Administrators need the capability to update and delete book details in the library database.

Functional Requirements:

The system shall provide an interface for administrators to modify existing book information, including title, author, and quantity.

An option to delete a book record shall be available, with appropriate confirmation prompts.

1. Acceptance Criteria

Administrators can successfully update book information.

Deleting a book requires confirmation and removes the book record from the database.

1. Add Student Details

Requirement

Description: Administrators should be able to add and manage student information in the system.

Functional Requirements:

The system shall provide a form for entering student details, including name, student ID, contact information, and program.

Validation checks shall be implemented to ensure accuracy in student data.

1. Acceptance Criteria

Administrators can successfully add a new student with valid information.

Validation prevents the addition of students with duplicate IDs.

1. View Student Detail

Requirement

Description: Users, especially administrators, should be able to view a list of all registered students.

Functional Requirements:

The system shall display a comprehensive list of students, including details such as name, student ID, contact information, and program.

Users shall have the ability to sort and filter the student list based on various criteria.

14.6.2 Acceptance Criteria

The system presents a clear and organized list of all registered students.

Users can sort the student list by name, student ID, or program.

Filters allow users to view students based on specific criteria.

1. Update & Delete Student Details

Requirement

Description: Administrators need the ability to update and delete student details in the system.

Functional Requirements:

The system shall allow administrators to modify existing student information, such as contact details and program.

An option to delete a student record shall be available, with appropriate confirmation prompts.

1Acceptance Criteria

Administrators can successfully update student information.

Deleting a student requires confirmation and removes the student record from the database.

1. Issue Books to Student

Requirement

Description: Administrators should be able to issue books to students, with a limit of 3 books per student.

Functional Requirements:

The system shall validate the availability of the requested books before issuing them.

A transaction record shall be created for each issued book, including timestamps and student details.

Acceptance Criteria

Administrators can successfully issue books to students without exceeding the maximum limit.

A transaction record is created for each issued book.

1. Return Books from Student

Requirement

Description: Administrators should be able to process the return of books from students.

Functional Requirements:

The system shall update the availability of returned books in the database.

Transaction records shall be updated with return timestamps.

Acceptance Criteria

Administrators can successfully process the return of books from students.

The availability of returned books is updated in the database.

1. Search Books

Requirement

Description: Users, including administrators, should be able to search for specific books based on various criteria.

Functional Requirements:

The system shall provide a search functionality allowing users to find books by title, author, ISBN, or category.

Search results shall be displayed in a clear and organized manner.

Acceptance Criteria

Users can successfully search for books based on title, author, ISBN, or category.

Search results are presented in a user-friendly and organized manner.

1. Search Student

Requirement

Description: Users, especially administrators, should be able to search for specific students based on various criteria.

Functional Requirements:

The system shall provide a search functionality allowing users to find students by name, student ID, or program.

Search results shall be displayed in a clear and organized manner.

Acceptance Criteria

Users can successfully search for students based on name, student ID, or program.

Search results are presented in a user-friendly and organized manner.

1. Track Borrowed Books per Student

Requirement

Description: Users, especially administrators, should be able to view the list of students who currently have books checked out.

Functional Requirements:

The system shall display a report showing the names of students, along with the titles of books currently in their possession.

This report shall be accessible to administrators for efficient book tracking

# **Interface requirements**

## User Interfaces

## Hardware Interfaces

## Software Interfaces

## Communication Interfaces

# **Non Functional Requirements**

## Performance

The system shall respond to user requests with an average latency of less than 1 second and a maximum latency of 3 seconds.

## Reliability

The system shall be available 99.9% of the time, excluding scheduled maintenance windows.

## Scalability

The system shall be able to handle a minimum of 10 concurrent users without degradation in performance

## Security

The system shall require a password of at least 8 characters, containing at least one uppercase letter, one lowercase letter, one number, and one special character.

## Compatibility

The system shall be compatible with the latest version of the OS , Android and IOS Platforms

## Maintainability

The system shall be designed and implemented in a modular fashion to facilitate future updates and enhancements

## Portability

The system shall be able to run on Windows, macOS, and Linux

# **System Models and Diagrams**

## Use Case Diagram

A diagram of a diagram

Description automatically generated with medium confidence

Figure 2: Use Case Diagram - HAS

## Class Diagram

A diagram of a computer

Description automatically generated

Figure 3: Class Diagram - HAS

* 1. Sequence diagram

A white rectangular object with black lines

Description automatically generated

ER Diagrams

A diagram of a flowchart

Description automatically generated

A computer screen shot of a computer

Description automatically generated

A screenshot of a computer diagram

Description automatically generated

GITHUB Project link

# **System Evolution**

# **Time plan**

# **Appendices**

# **References**