Software Requirements Specification

for

Library Management System

Prepared by
Priyansh Dimri (23/SE/119)
Prashant Pal (23/SE/116)

Delhi Technological University

February 28, 2024

Table of Contents

1. Introduction

1.1 Purpose

This document specifies the software requirements of web-based Library Management System designed to execute virtually all the operations in a Library in online mode. This document covers every aspect of the system and its features.

1.2 Document Conventions

This SRS follows the IEEE standard for software requirements specification documents. All high-level requirements are assumed to be inherited by detailed requirements. Priorities are assigned to each requirement statement.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, users, testers, and documentation writers. The SRS is organized into sections that describe the functional requirements, performance requirements, design constraints, and other specifications. Developers should start with the overview sections, while testers and users might find the functional requirements and interface requirements sections most useful.

1.4 Product Scope

The Library Management System is a software solution designed to streamline the process of utilizing library resources efficiently from anywhere in the world. The LMS provides an efficient and user-friendly interface for utilizing the library resources and services online. It is aimed towards improving the user experience and operational efficiency.

1.5 References

- IEEE Standard for Software Requirements Specifications, IEEE Std 830-1998
- User Interface Style Guide, Version 2.0, ABC Corp, 2023

2. Overall Description

2.1 Product Perspective

The Library Management System is a new, self-contained product designed to streamline the usage of library resources efficiently. It is not a replacement or a follow-on member of a product family but a standalone system.

2.2 Product Functions

- **Account Registration**: Creating a new account for the user based on specified roles.

- **User Authencation**: Logging in using username and password.
- **Manage Account Details**: Update user’s details.
- **Check Books in Library**: Get details about all the books in the library.
- **Reserve Books**: Reserve a limited amount of books based on library’s policies.
- **Renew Books**: Renewing a borrowed books a limited number of times based on library’s policies.
- **Issue Books**: Issue book to a system user who issued it physically.
- **Manage Book Instances**: Manage the status and details of all the book instances in the library.
- **Set Library Policies**: Manage all the operations and policies in the library by Library Administrators.
- **Manage Library Budget**: Manage all the library’s budget online by Library Administrators.

2.3 User Classes and Characteristics

- **System User**: Frequent users who need to reserve, renew or get details of books in the library online.
- **Librarian**: Use the system to manage the reservations, issue a book to a system user and manage all the book instances in the library.
- **Library Administrators**: Use the system to set, update or delete library policies and store the library budget and transactions online.
- **Super Admin**: Use the system to create, remove or update all the users in the Library Management System.

2.4 Operating Environment

The SBS will operate on standard commercial hardware platforms, running any modern operating system such as Windows, macOS, or Linux. All the services will be cloud-based including the database.

2.5 Design and Implementation Constraints

The Library Management System must adhere to all applicable regulatory policies and corporate guidelines.In addition, the following technical specifications will be used:

2.5.1 APIs

1. **User APIs**: For user registration, login, and managing user details.
2. **Book APIs**: For checking book availability, reserving books, renewing books, and managing book instances.
3. **Librarian APIs**: For issuing books to users and managing book reservations.
4. **Admin APIs**: For setting library policies and managing the library budget.

2.5.2 Security Measures

1. **Authentication**: Use JSON Web Tokens (JWT) for stateless server-side user authentication.

2. **Authorization**: Implement role-based access control (RBAC) to restrict access based on user roles.
3. **Data Encryption**: Use HTTPS for secure data transmission. Store passwords as hashed values (bcrypt can be used for hashing).
4. **Input Validation**: Use a library like express-validator to validate user inputs and protect against attacks such as SQL injection and XSS.
5. **Rate Limiting**: Use a package like express-rate-limit to prevent brute-force attacks.
6. **Security Headers**: Use a module like helmet.js to secure your Express apps by setting various HTTP headers.

2.5.3 Testing Tools

1. **Unit Testing**: Jest is a popular choice for unit testing in JavaScript. It's easy to set up and has a lot of features out of the box.
2. **Integration Testing**: Supertest is often used alongside Jest for integration testing of Node.js HTTP servers.
3. **End-to-End Testing**: Cypress is a powerful tool for end-to-end testing if you need to test interactions between multiple systems.
4. **Test Coverage**: Istanbul (nyc) is a good tool for checking how much of your code is covered by tests.

2.5.4 Storage Constraints

The free tier of Heroku provides 5GB of storage. This is the maximum storage capacity of the system. If the system needs to store more data, it will need to be upgraded to a paid tier or use an external storage service.

2.6 User Documentation

The Library Management System will come with comprehensive set of documentations for each user class along with a general README file. All documentations will be available in Markdown format.

2.7 Assumptions and Dependencies

It is assumed that the Library Management System will have access to reliable and fast internet connection for executing functions properly and securely. The system will be using the free tier of Heroku and MySQL, ExpressJS, ReactJS and NodeJS.

3. External Interface Requirements

3.1 User Interfaces

The Library Management System will provide a user-friendly interface that is easy to navigate and understand. The interface will be designed with a focus on usability and accessibility, ensuring that all users, regardless of their technical proficiency, can effectively interact with the system.

3.1.1 System User Interface

The interface for system users will include features for account registration, user authentication, account management, book reservation and book renewal. Users will be able to view and manage their account details, check the availability of books in the library, reserve books and renew borrowed books. Note that system users can not issue books online. That function can only be performed by librarian when the system user physically issues the book from the library.

3.1.2 Librarian Interface

The librarian interface will include features for managing book reservations, issuing books to system users, and managing book instances in the library. Librarians will be able to will be able to view and manage their account details, check the availability of books in the library, view and manage all book reservations, issue books to users, and manage the status and details of all book instances in the library.

3.1.3 Library Administrator Interface

The library administrator interface will include features to view and manage their account details, updating, or deleting library policies and managing the library's budget online. Administrators will be able to view and manage all operations and policies in the library, as well as manage the library's budget.

3.1.4 Super Admin Interface

The super admin interface will include features for creating, removing, or updating all users in the Library Management System. Super admins will have full control over user management within the system.

The user interfaces will be designed to be intuitive and easy to use, with clear instructions and prompts. They will be compatible with all modern operating systems such as Windows, macOS, and Linux, and will be accessible via a web browser for ease of access and use. The interfaces will adhere to all applicable regulatory policies and corporate guidelines.

3.2 Hardware Interfaces

The Library Management System will be designed to be compatible with a wide range of hardware interfaces. This includes but is not limited to personal computers, laptops, tablets, and smartphones. The system will support devices running on modern operating systems such as Windows, macOS, Linux, iOS, and Android.

3.1.1 Personal Computers and Laptops

The system will support personal computers and laptops with a minimum of 4GB RAM and 2GHz processor speed. The system will be accessible via a web browser, requiring an active internet connection.

3.2.2 Tablets and Smartphones

The system will support tablets and smartphones with a minimum of 2GB RAM and 1.5GHz processor speed. The system will be accessible via a web browser, requiring an active internet connection.

3.2.3 Communication Protocols

The system will use standard internet protocols (HTTP/HTTPS) for data transmission. It will also support WebSocket for real-time data updates.

3.2.4 Data Interactions

The system will interact with a server-side database to store and retrieve data. It will use the device’s network interface for communicating with the server.

3.3 Software Interfaces

The Library Management System will interact with several software components to provide a seamless and efficient service.

3.3.1 Operating Systems

The system will be compatible with all modern operating systems such as Windows (version 10 and above), macOS (version 10.15 and above), Linux (Ubuntu 18.04 and above), iOS (version 13 and above), and Android (version 9 and above).

3.3.2 Databases

The system will interact with MySQL for data storage. The database will store information about users, books, reservations, and transactions.

3.3.3 Libraries and Tools

The system will be built using MySQL, Express.js, React.js, and Node.js. It will also use HTML5/CSS3 and Bootstrap for front-end development and JavaScript for back-end development.

3.3.4 Data Sharing

Data will be shared across software components using JSON format. The system will use RESTful API for communication between the front-end and the back-end. The API will follow standard HTTP methods (GET, POST, PUT, DELETE) for CRUD operations.

3.3.5 Communication

The system will use HTTP/HTTPS protocols for communication between the client and the server.

3.4 Communications Interfaces

The Library Management System will use HTTP/HTTPS for communication between the client and the server. It will also use SQL for communication with the MySQL database server.

3.4.1 Web Browser

The system will be accessible via a web browser. It will support all modern web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

3.4.2 Network Server Communications Protocols

The system will use HTTP/HTTPS for client-server communication. It will also use WebSocket for real-time updates.

3.4.3 Electronic Forms

The system will use electronic forms for user registration, book reservation, and other data entry tasks. The forms will be designed with a focus on usability and accessibility.

3.4.4 Communication Standards

The system will adhere to standard internet protocols such as HTTP and HTTPS.

3.4.5 Communication Security and Encryption

All communication between the client and the server will be encrypted using SSL/TLS protocols to ensure data security. Passwords will be hashed and stored securely in the database.

3.4.6 Data Transfer Rates and Synchronization

The system will be designed to optimize data transfer rates and minimize latency.

4. System Features

4.1 Account Creation (Registration)

4.1.1 Description and Priority

This feature allows new users to create an account in the Library Management System. It is of High priority as it is the entry point for users to access the system's services.

4.1.2 Stimulus/Response Sequences

- User selects the option to create a new account.
- System prompts the user to enter required details.
- User enters details and submits the form.
- System validates the information and creates a new account.

4.1.3 Functional Requirements

REQ-1: System must prompt the user to enter a unique username, a valid email address, and a strong password.
REQ-2: System must validate the information provided by the user.

REQ-3: System must display an error message if the entered email is already associated with an existing account.
REQ-4: System must display a success message upon successful account creation.

4.1.4 Data Requirements

The system needs to collect and store the following data: ‘Username’, ‘Email’, 'Password', 'Role', 'Status'.

4.2 Checking Registration Status

4.2.1 Description and Priority

This feature allows users to check their registration status. It is of Medium priority.

4.2.2 Stimulus/Response Sequences

- User logs into their account.
- User navigates to the account settings page.
- System displays the user's registration status.

4.2.3 Functional Requirements

REQ-1: System must display the user's registration status on the account settings page.

4.2.4 Data Requirements

The system needs to retrieve and display the 'Status' of the user's account.

4.3 Login

4.3.1 Description and Priority

This feature allows users to log into their account. It is of High priority as it is necessary for users to access their accounts.

4.3.2 Stimulus/Response Sequences

- User enters their username and password.
- System validates the credentials.
- If the credentials are valid, the system logs the user in and redirects them to their dashboard.

4.3.3 Functional Requirements

REQ-1: System must prompt the user to enter their username and password.
REQ-2: System must validate the entered credentials.

REQ-3: System must display an error message if the entered credentials are incorrect.

4.3.4 Data Requirements

The system needs to collect and validate the following data: 'Username', 'Password'.

4.4 Change Credentials

4.4.1 Description and Priority

This feature allows users to change their login credentials. It is of Medium priority.

4.4.2 Stimulus/Response Sequences

- User navigates to the account settings page.
- User selects the option to change credentials.
- System prompts the user to enter their current password and their new credentials.
- User enters the required information and submits the form.
- System validates the information and updates the user's credentials.

4.4.3 Functional Requirements

REQ-1: System must prompt the user to enter their current password and their new credentials.
REQ-2: System must validate the entered information.
REQ-3: System must update the user's credentials upon successful validation.
REQ-4: System must display a success message upon successful update of credentials.

4.4.4 Data Requirements

The system needs to collect, validate, and update the following data: 'Username', 'Password'.

4.5 Check Books in Library

4.5.1 Description and Priority

This feature allows users to check the availability of books in the library. It is of High priority as it is necessary for users to access the library’s resources.

4.5.2 Stimulus/Response Sequences

- User navigates to the library catalog page.

- User enters the name or ISBN of the book they are looking for. Or they can choose to filter the books based on author and other criteria.

- System searches the library database and displays the availability status of the book.

4.5.3 Functional Requirements

REQ-1: System must prompt the user to enter the name or ISBN of the book.
REQ-2: System must search the library database for the entered book.
REQ-3: System must display the availability status of the book.
REQ-4: System must display an error message if the entered book is not found in the library database.

4.5.4 Data Requirements

The system needs to collect the 'Name' or 'ISBN' of the book and retrieve its 'Availability Status' from the database.

4.6 Renew Books

4.6.1 Description and Priority

This feature allows users to extend the due date of their borrowed books. Users can select the books they want to renew from their list of borrowed books. The system will then update the due date of the selected books according to the library's renewal policies. It is of High priority as it is necessary for users to extend their borrowing period.

4.6.2 Stimulus/Response Sequences

- User navigates to their account page.

- User selects the option to renew a borrowed book.

- System extends the due date of the borrowed book.

4.6.3 Functional Requirements

REQ-1: System must display the list of borrowed books in the user's account page.

REQ-2: System must allow the user to select a borrowed book to renew according to the Library's Renewal Policies.

REQ-3: System must extend the due date of the selected book upon renewal.

4.6.4 Data Requirements

The system needs to retrieve the 'Due Date' of the borrowed book and update it upon renewal.

4.7 Reserve Books

4.7.1 Description and Priority

This feature allows users to reserve available books. Users can search for books in the library's catalog and reserve them for a certain period of time. The system will then update the status of the reserved books and notify the user when the books are ready for pickup. It is of High priority as it is necessary for users to secure a book they wish to borrow.

4.7.2 Stimulus/Response Sequences

- User navigates to the library catalog page.

- User selects a book to reserve.

- System reserves the selected book for the user.

4.7.3 Functional Requirements

REQ-1: System must display the list of available books in the library catalog page.

REQ-2: System must allow the user to select a book to reserve.

REQ-3: System must reserve the selected book for the user according to the Library's Reservation Policies.

4.7.4 Data Requirements

The system needs to collect the 'ISBN' of the book to reserve and update its 'Availability Status'.

4.8 Reserve Books from Librarian Side

4.8.1 Description and Priority

This feature allows librarians to reserve books on behalf of users. Librarians can enter the user's account details and the details of the book they want to reserve into the system. The system will then reserve the book for the user and update the book's status accordingly. It is of Medium priority.

4.8.2 Stimulus/Response Sequences

- User provides their account details and the book they wish to reserve to the librarian.

- Librarian enters the user's account details and the book details into the system.

- System reserves the book for the user.

4.8.3 Functional Requirements

REQ-1: System must allow the librarian to enter the user's account details and the book details.

REQ-2: System must reserve the selected book for the user according to the Library's Reservation Policies.

4.8.4 Data Requirements

The system needs to collect the 'Username' and 'ISBN' of the book to reserve and update its 'Availability Status'.

4.9 Issue Book to System User by Librarian

4.9.1 Description and Priority

This feature allows librarians to issue books to users. Librarians can enter the user's account details and the details of the book they want to issue into the system. The system will then issue the book to the user and update the book's status accordingly. It is of High priority as it is necessary for users to borrow books.

4.9.2 Stimulus/Response Sequences

- User provides their account details and the book they wish to borrow to the librarian.

- Librarian enters the user's account details and the book details into the system.

- System issues the book to the user.

4.9.3 Functional Requirements

REQ-1: System must allow the librarian to enter the user's account details and the book details.

REQ-2: System must issue the selected book to the user according to the Library's Issuance Policies.

4.9.4 Data Requirements

The system needs to collect the 'Username' and 'ISBN' of the book to issue and update its 'Availability Status'.

4.10 Manage All Book Instances in the Library

4.10.1 Description and Priority

This feature allows librarians to manage all instances of books in the library. Librarians can add new books, delete existing books, or update the details of existing books. The system will then update the library's catalog accordingly. It is of High priority as it is necessary for maintaining the library's catalog.

4.10.2 Stimulus/Response Sequences

- Librarian navigates to the library management page.

- Librarian selects an action (add, delete, update) to perform on a book instance.

- System performs the selected action on the book instance.

4.10.3 Functional Requirements

REQ-1: System must display the list of all book instances in the library management page.

REQ-2: System must allow the librarian to select an action to perform on a book instance.

REQ-3: System must perform the selected action on the book instance.

4.10.4 Data Requirements

The system needs to retrieve and display all 'Book Instances' and allow the librarian to add, delete, or update a 'Book Instance'.

4.11 Setting Library Policies by Library Administrators

4.11.1 Description and Priority

This feature allows library administrators to set the library's policies. Administrators can enter the details of the new policies into the system. The system will then update the library's policies accordingly. It is of High priority as it is necessary for maintaining the library's operations.

4.11.2 Stimulus/Response Sequences

- Library administrator navigates to the library policies page.

- Library administrator enters the new policy details.

- System updates the library policies.

4.11.3 Functional Requirements

REQ-1: System must display the current library policies in the library policies page.

REQ-2: System must allow the library administrator to enter new policy details.

REQ-3: System must update the library policies with the new policy details.

4.11.4 Data Requirements

The system needs to collect and update the 'Library Policies'.

4.12 Managing Library Budget by Library Administrators

4.12.1 Description and Priority

This feature allows library administrators to manage the library's budget. Administrators can enter the details of the new budget into the system. The system will then update the library's budget accordingly. It is of High priority as it is necessary for maintaining the library's financial health.

4.12.2 Stimulus/Response Sequences

- Library administrator navigates to the library budget page.

- Library administrator enters the new budget details.

- System updates the library budget.

4.12.3 Functional Requirements

REQ-1: System must display the current library budget in the library budget page.

REQ-2: System must allow the library administrator to enter new budget details.

REQ-3: System must update the library budget with the new budget details.

4.12.4 Data Requirements

The system needs to collect and update the 'Library Budget'.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system should be able to handle multiple simultaneous user requests without significant degradation in response time. The system should also be able to process a book reservation or return within 2 seconds under normal load conditions.

5.2 Safety Requirements

The system should prevent unauthorized access to user data. It should also have safeguards in place to prevent data loss, such as regular backups and data recovery mechanisms.

5.3 Security Requirements

The system should use secure protocols for data transmission. User authentication should be enforced through secure methods like hashed passwords. The system should also comply with privacy regulations like GDPR.

5.4 Software Quality Attributes

The system should be:

- Adaptable to changes in the library’s policies or inventory.
- Available for use during the library’s operating hours.
- Correct in terms of processing transactions and managing data.
- The system should be designed in a modular way to allow for reuse of components where possible. This will make the system easier to maintain and extend in the future.
- Reliable with minimal downtime.
- The system should be easy to use, with a user-friendly interface and clear instructions. The system should also provide helpful error messages when something goes wrong.
- The system should be able to support up to 100 concurrent users. This is based on the expected usage of the system and the limitations of the free tier of Heroku.

- The system should be easy to maintain, with clear, well-commented code. It should also include comprehensive documentation for developers.

5.5 Business Rules

- Only authenticated users can reserve or return books.
- Librarians can add or remove books from the inventory.
- Only the super admin can create, remove, or update user accounts.

6. Other Requirements

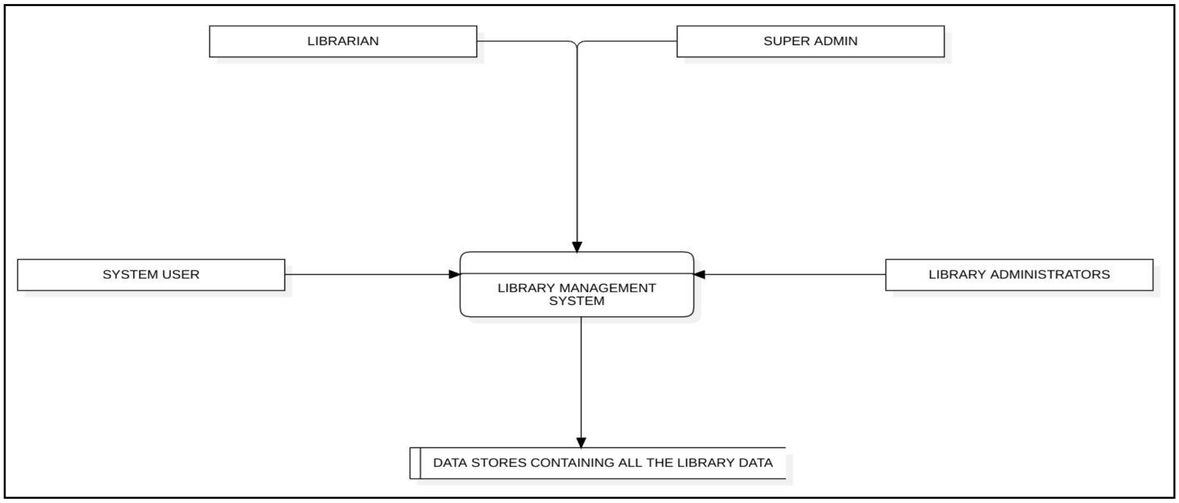
The system should be able to support multiple languages (internationalization requirements). It should also comply with all relevant legal requirements, such as data protection and privacy laws. The system should be designed in a modular way to allow for reuse of components where possible.

Appendix A: Glossary

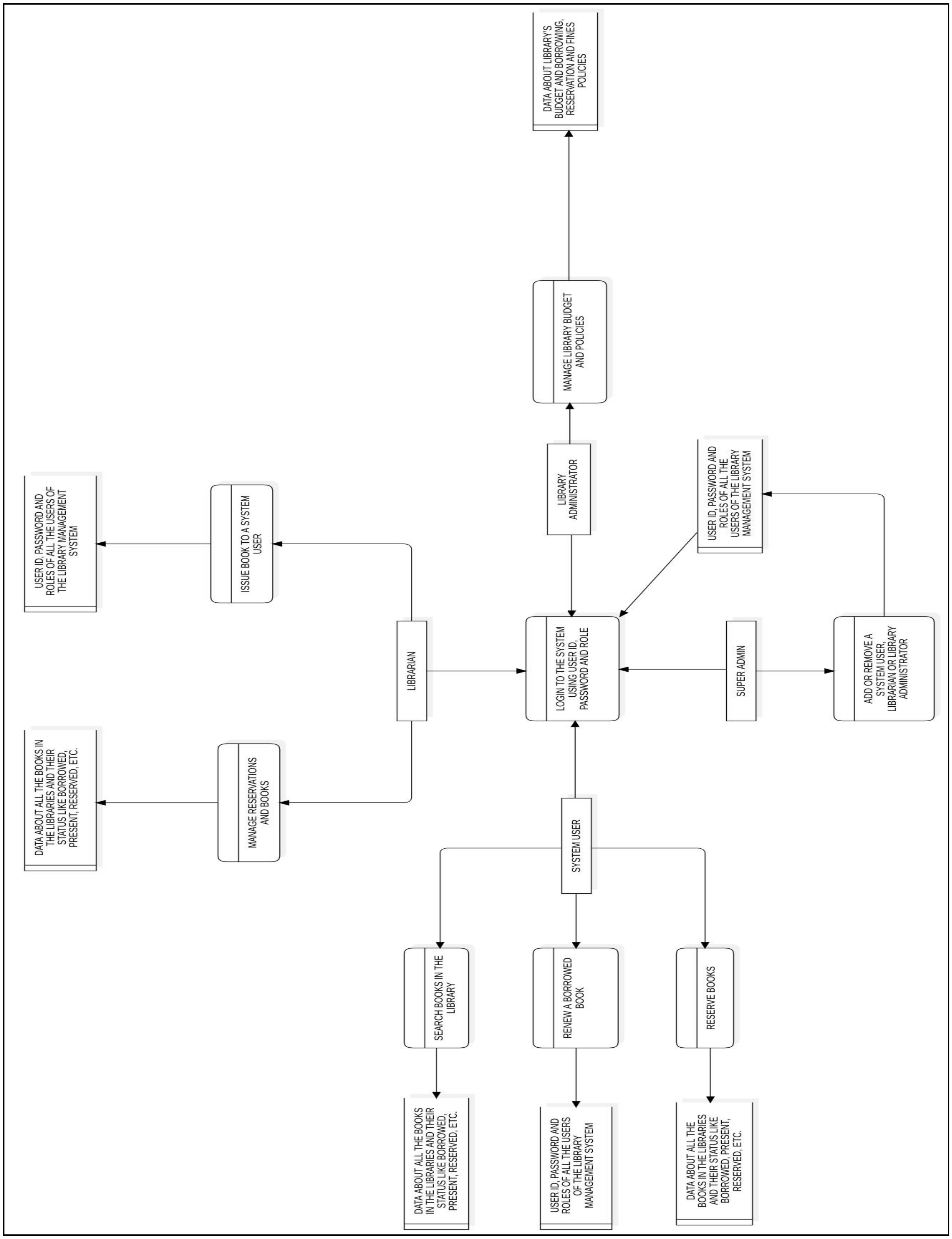
- **SRS**: Software Requirements Specification
- **MERN Stack**: MongoDB, Express.js, React.js, and Node.js
- **CRUD**: Create, Read, Update, Delete
- **HTTP/HTTPS**: Hypertext Transfer Protocol/Secure
- **RESTful API**: Representational State Transfer Application Programming Interface
- **JSON**: JavaScript Object Notation

Appendix B: Analysis Models

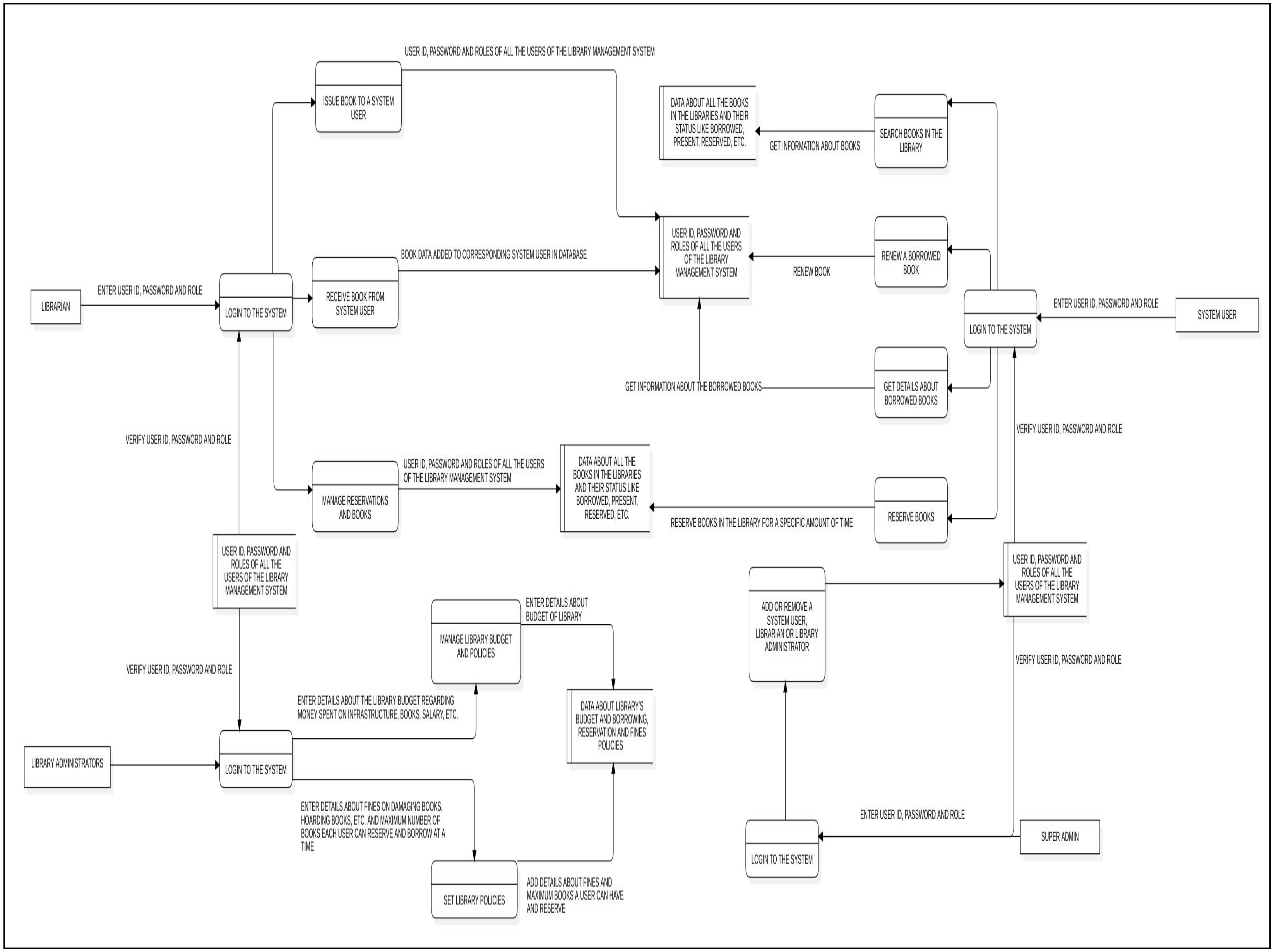
Context Level Diagram:


LIBRARIAN
SUPER ADMIN
SYSTEM USER
LIBRARY MANAGEMENT SYSTEM
LIBRARY ADMINISTRATORS
DATA STORES CONTAINING ALL THE LIBRARY DATA

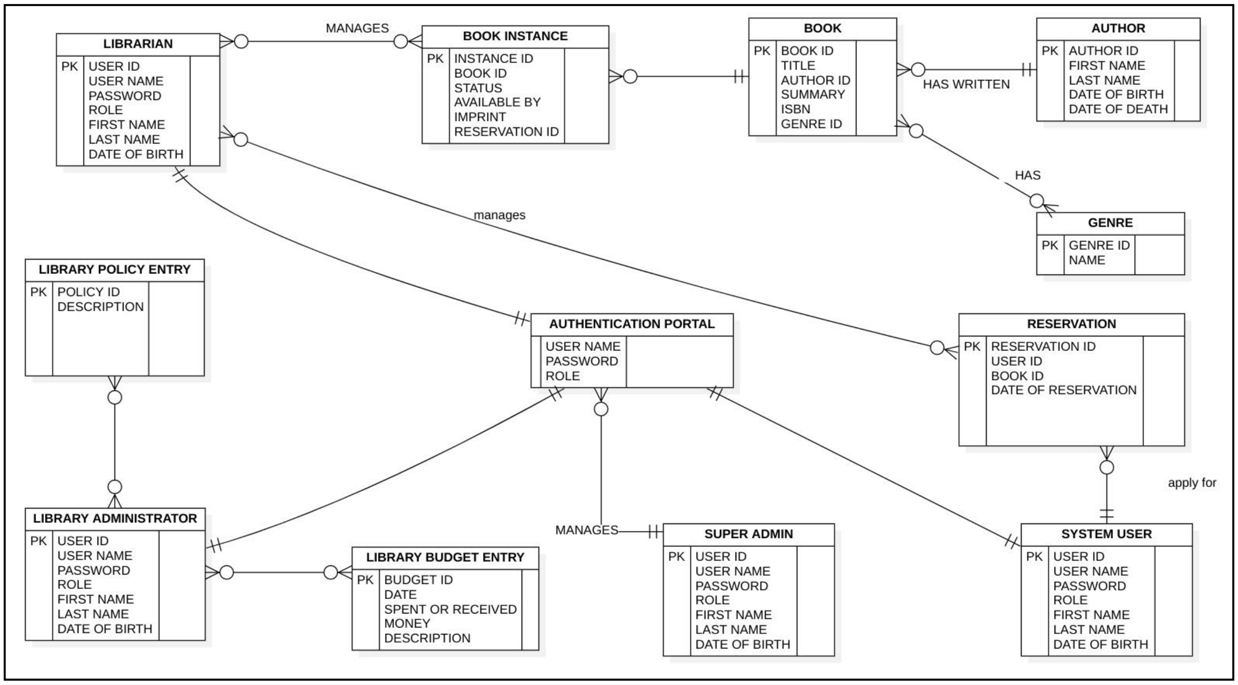
Level-1 DFD:


DATA ABOUT LIBRARY'S BUDGET AND BORROWING, RESERVATION AND FINES POLICIES
MANAGE LIBRARY BUDGET AND POLICIES
LIBRARY ADMINISTRATOR
USER ID, PASSWORD AND ROLES OF ALL THE USERS OF THE LIBRARY MANAGEMENT SYSTEM
USER ID, PASSWORD AND ROLES OF ALL THE USERS OF THE LIBRARY MANAGEMENT SYSTEM
ISSUE BOOK TO A SYSTEM USER
LIBRARIAN
LOGIN TO THE SYSTEM USING USER ID, PASSWORD AND ROLE
SUPER ADMIN
ADD OR REMOVE A SYSTEM USER, LIBRARIAN OR LIBRARY ADMINISTRATOR
DATA ABOUT ALL THE BOOKS IN THE LIBRARIES AND THEIR STATUS LIKE BORROWED, PRESENT, RESERVED, ETC.
MANAGE RESERVATIONS AND BOOKS
SYSTEM USER
SEARCH BOOKS IN THE LIBRARY
RENEW A BORROWED BOOK
RESERVE BOOKS
DATA ABOUT ALL THE BOOKS IN THE LIBRARIES AND THEIR STATUS LIKE BORROWED, PRESENT, RESERVED, ETC.
USER ID, PASSWORD AND ROLES OF ALL THE USERS OF THE LIBRARY MANAGEMENT SYSTEM
DATA ABOUT ALL THE BOOKS IN THE LIBRARIES AND THEIR STATUS LIKE BORROWED, PRESENT, RESERVED, ETC.

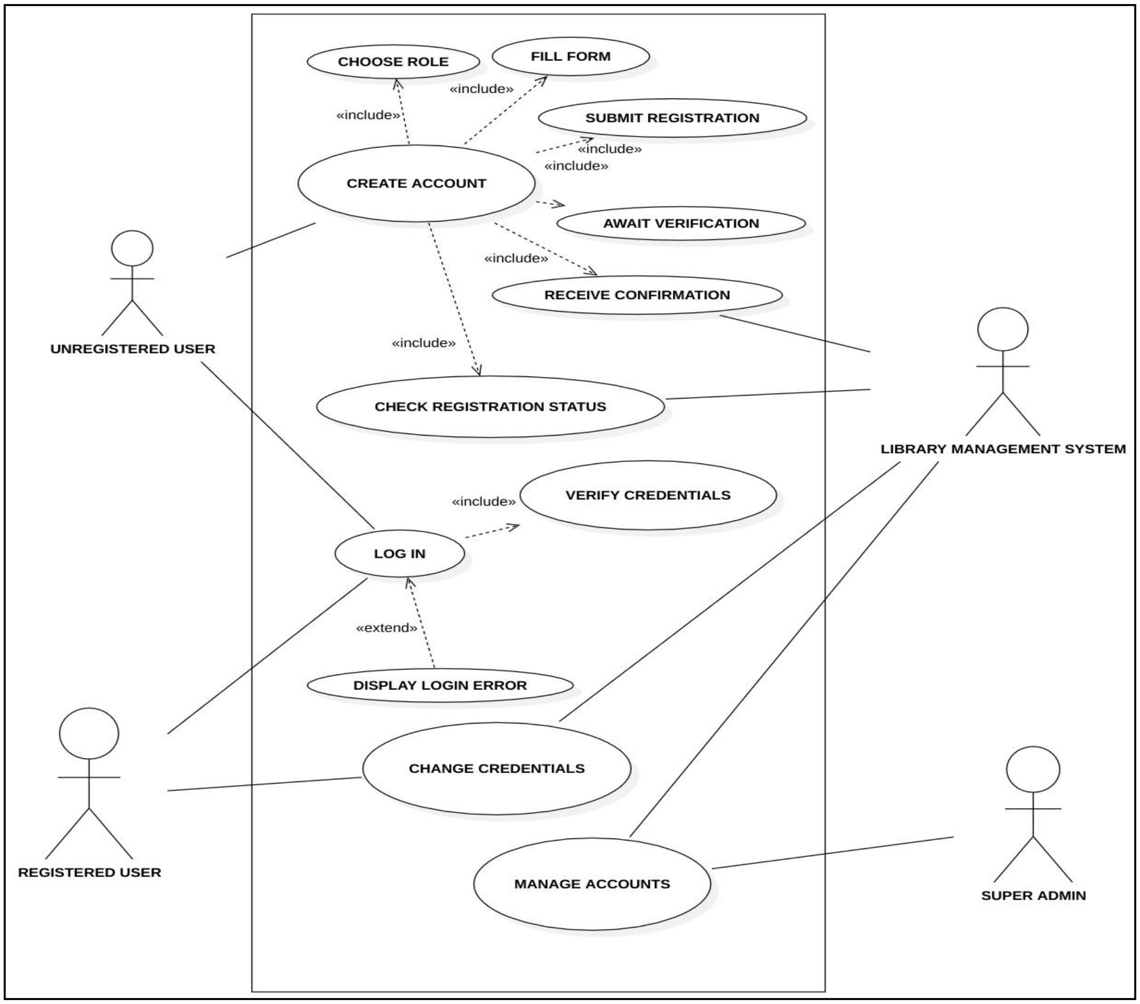
Level-2 DFD:


USER ID, PASSWORD AND ROLES OF ALL THE USERS OF THE LIBRARY MANAGEMENT SYSTEM
ISSUE BOOK TO A SYSTEM USER
DATA ABOUT ALL THE BOOKS IN THE LIBRARIES AND THEIR STATUS LIKE BORROWED, PRESENT, RESERVED, ETC.
GET INFORMATION ABOUT BOOKS
SEARCH BOOKS IN THE LIBRARY
BOOK DATA ADDED TO CORRESPONDING SYSTEM USER IN DATABASE
USER ID, PASSWORD AND ROLES OF ALL THE USERS OF THE LIBRARY MANAGEMENT SYSTEM
RENEW BOOK
RENEW A BORROWED BOOK
LIBRARIAN
ENTER USER ID, PASSWORD AND ROLE
LOGIN TO THE SYSTEM
RECEIVE BOOK FROM SYSTEM USER
LOGIN TO THE SYSTEM
ENTER USER ID, PASSWORD AND ROLE
SYSTEM USER
GET INFORMATION ABOUT THE BORROWED BOOKS
GET DETAILS ABOUT BORROWED BOOKS
VERIFY USER ID, PASSWORD AND ROLE
VERIFY USER ID, PASSWORD AND ROLE
USER ID, PASSWORD AND ROLES OF ALL THE USERS OF THE LIBRARY MANAGEMENT SYSTEM
MANAGE RESERVATIONS AND BOOKS
DATA ABOUT ALL THE BOOKS IN THE LIBRARIES AND THEIR STATUS LIKE BORROWED, PRESENT, RESERVED, ETC.
RESERVE BOOKS IN THE LIBRARY FOR A SPECIFIC AMOUNT OF TIME
RESERVE BOOKS
USER ID, PASSWORD AND ROLES OF ALL THE USERS OF THE LIBRARY MANAGEMENT SYSTEM
ADD OR REMOVE A SYSTEM USER, LIBRARIAN OR LIBRARY ADMINISTRATOR
USER ID, PASSWORD AND ROLES OF ALL THE USERS OF THE LIBRARY MANAGEMENT SYSTEM
ENTER DETAILS ABOUT BUDGET OF LIBRARY
MANAGE LIBRARY BUDGET AND POLICIES
VERIFY USER ID, PASSWORD AND ROLE
VERIFY USER ID, PASSWORD AND ROLE
ENTER DETAILS ABOUT THE LIBRARY BUDGET REGARDING MONEY SPENT ON INFRASTRUCTURE, BOOKS, SALARY, ETC.
DATA ABOUT LIBRARY'S BUDGET AND BORROWING, RESERVATION AND FINES POLICIES
LIBRARY ADMINISTRATORS
LOGIN TO THE SYSTEM
ENTER DETAILS ABOUT FINES ON DAMAGING BOOKS, HOARDING BOOKS, ETC. AND MAXIMUM NUMBER OF BOOKS EACH USER CAN RESERVE AND BORROW AT A TIME
ENTER USER ID, PASSWORD AND ROLE
SUPER ADMIN
LOGIN TO THE SYSTEM
SET LIBRARY POLICIES
ADD DETAILS ABOUT FINES AND MAXIMUM BOOKS A USER CAN HAVE AND RESERVE

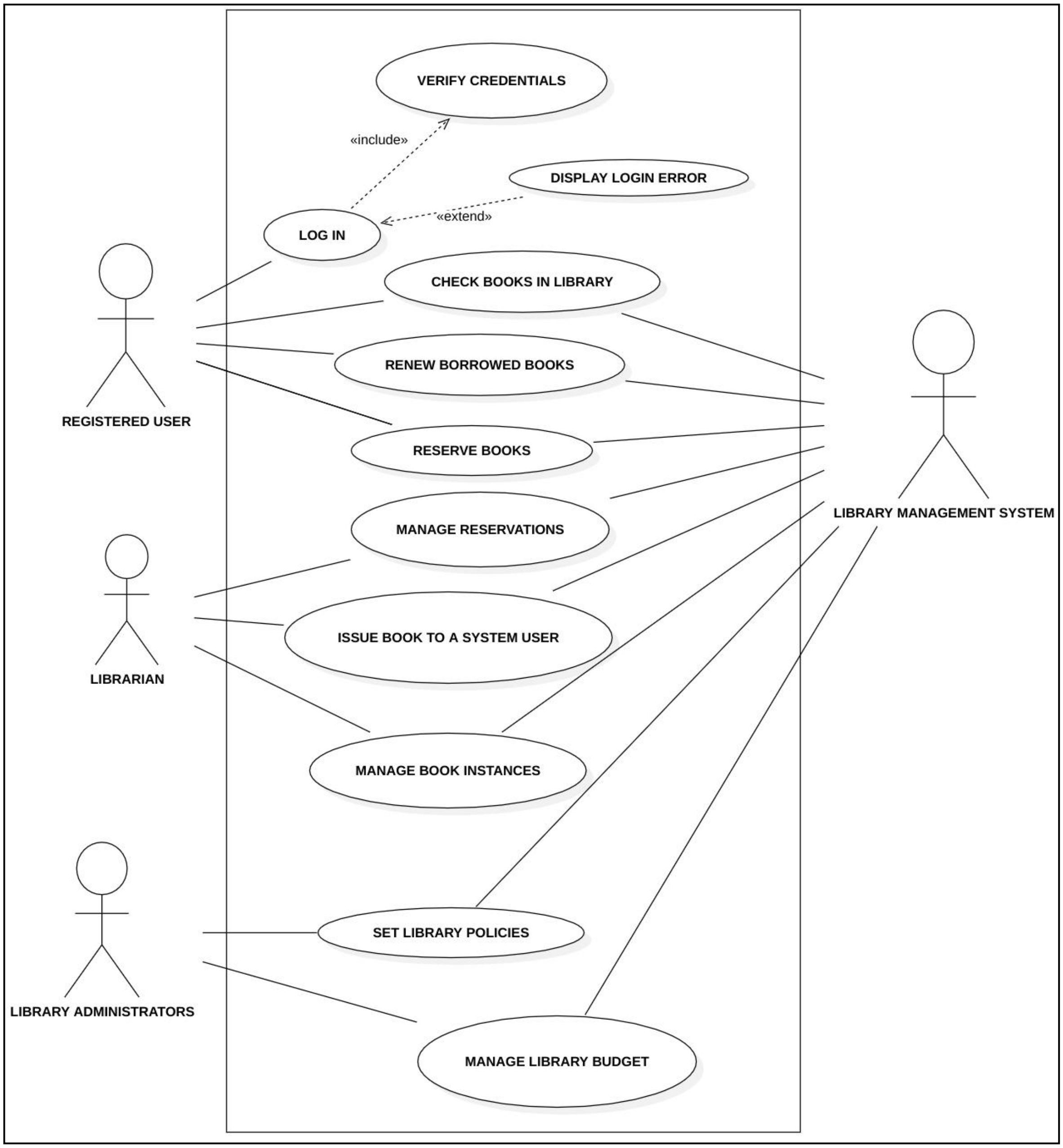
Entity Relationship Diagram

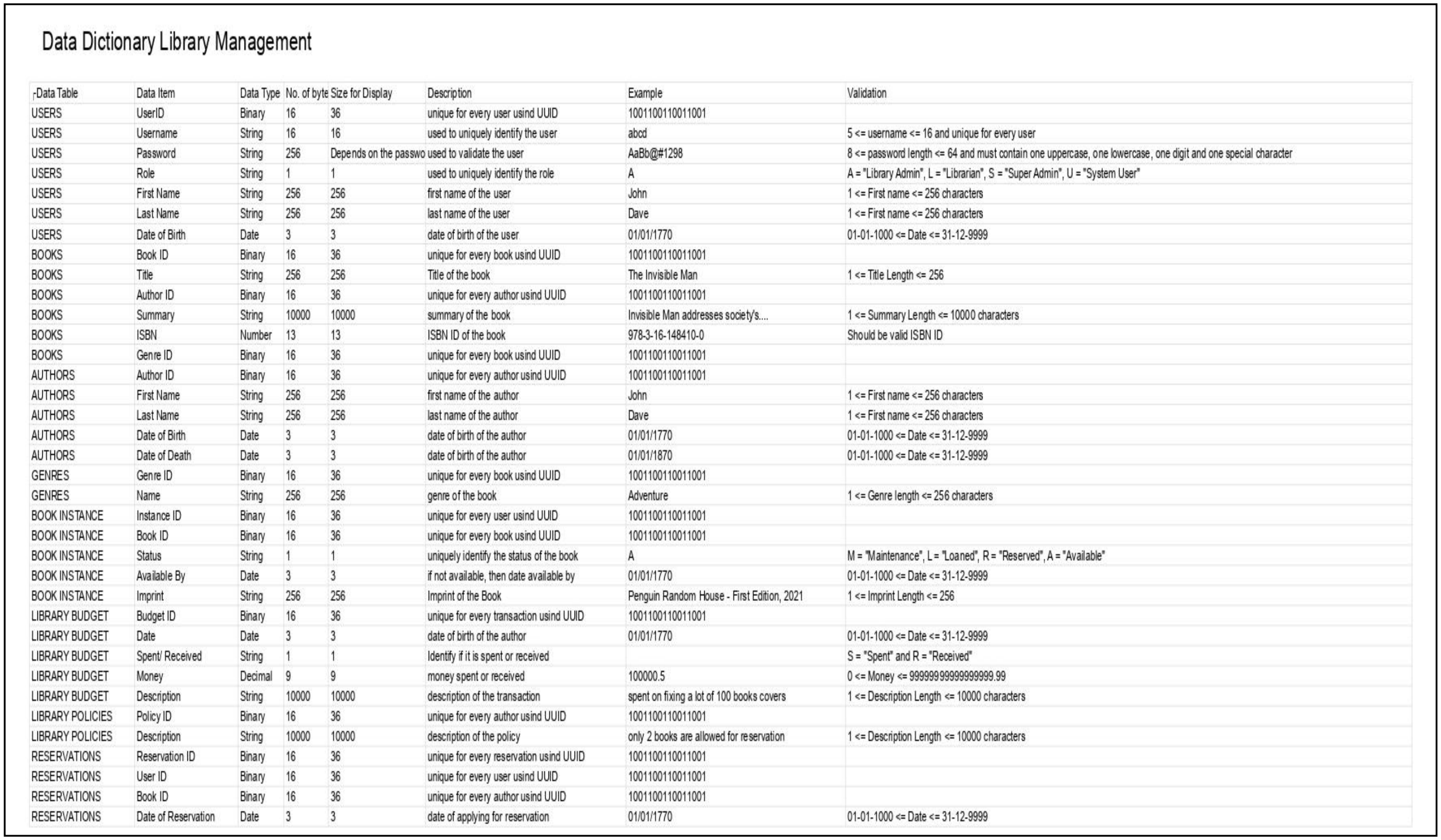


Use Case Diagram(User Management System)



Use Case Diagram(Other Functions)





Data Dictionary Library Management

┌Data Table	Data Item	Data Type	No. of byte	Size for Display	Description	Example	Validation
USERS	UserID	Binary	16	36	unique for every user usind UUID	1001100110011001	
USERS	Username	String	16	16	used to uniquely identify the user	abcd	5 <= username <= 16 and unique for every user
USERS	Password	String	256	Depends on the passwo	used to validate the user	AaBb@#1298	8 <= password length <= 64 and must contain one uppercase, one lowercase, one digit and one special character
USERS	Role	String	1	1	used to uniquely identify the role	A	A = "Library Admin", L = "Librarian", S = "Super Admin", U = "System User"
USERS	First Name	String	256	256	first name of the user	John	1 <= First name <= 256 characters
USERS	Last Name	String	256	256	last name of the user	Dave	1 <= First name <= 256 characters
USERS	Date of Birth	Date	3	3	date of birth of the user	01/01/1770	01-01-1000 <= Date <= 31-12-9999
BOOKS	Book ID	Binary	16	36	unique for every book usind UUID	1001100110011001	
BOOKS	Title	String	256	256	Title of the book	The Invisible Man	1 <= Title Length <= 256
BOOKS	Author ID	Binary	16	36	unique for every author usind UUID	1001100110011001	
BOOKS	Summary	String	10000	10000	summary of the book	Invisible Man addresses society's....	1 <= Summary Length <= 10000 characters
BOOKS	ISBN	Number	13	13	ISBN ID of the book	978-3-16-148410-0	Should be valid ISBN ID
BOOKS	Genre ID	Binary	16	36	unique for every book usind UUID	1001100110011001	
AUTHORS	Author ID	Binary	16	36	unique for every author usind UUID	1001100110011001	
AUTHORS	First Name	String	256	256	first name of the author	John	1 <= First name <= 256 characters
AUTHORS	Last Name	String	256	256	last name of the author	Dave	1 <= First name <= 256 characters
AUTHORS	Date of Birth	Date	3	3	date of birth of the author	01/01/1770	01-01-1000 <= Date <= 31-12-9999
AUTHORS	Date of Death	Date	3	3	date of birth of the author	01/01/1870	01-01-1000 <= Date <= 31-12-9999
GENRES	Genre ID	Binary	16	36	unique for every book usind UUID	1001100110011001	
GENRES	Name	String	256	256	genre of the book	Adventure	1 <= Genre length <= 256 characters
BOOK INSTANCE	Instance ID	Binary	16	36	unique for every user usind UUID	1001100110011001	
BOOK INSTANCE	Book ID	Binary	16	36	unique for every book usind UUID	1001100110011001	
BOOK INSTANCE	Status	String	1	1	uniquely identify the status of the book	A	M = "Maintenance", L = "Loaned", R = "Reserved", A = "Available"
BOOK INSTANCE	Available By	Date	3	3	if not available, then date available by	01/01/1770	01-01-1000 <= Date <= 31-12-9999
BOOK INSTANCE	Imprint	String	256	256	Imprint of the Book	Penguin Random House - First Edition, 2021	1 <= Imprint Length <= 256
LIBRARY BUDGET	Budget ID	Binary	16	36	unique for every transaction usind UUID	1001100110011001	
LIBRARY BUDGET	Date	Date	3	3	date of birth of the author	01/01/1770	01-01-1000 <= Date <= 31-12-9999
LIBRARY BUDGET	Spent/ Received	String	1	1	Identify if it is spent or received		S = "Spent" and R = "Received"
LIBRARY BUDGET	Money	Decimal	9	9	money spent or received	100000.5	0 <= Money <= 9999999999999999.99
LIBRARY BUDGET	Description	String	10000	10000	description of the transaction	spent on fixing a lot of 100 books covers	1 <= Description Length <= 10000 characters
LIBRARY POLICIES	Policy ID	Binary	16	36	unique for every author usind UUID	1001100110011001	
LIBRARY POLICIES	Description	String	10000	10000	description of the policy	only 2 books are allowed for reservation	1 <= Description Length <= 10000 characters
RESERVATIONS	Reservation ID	Binary	16	36	unique for every reservation usind UUID	1001100110011001	
RESERVATIONS	User ID	Binary	16	36	unique for every user usind UUID	1001100110011001	
RESERVATIONS	Book ID	Binary	16	36	unique for every author usind UUID	1001100110011001	
RESERVATIONS	Date of Reservation	Date	3	3	date of applying for reservation	01/01/1770	01-01-1000 <= Date <= 31-12-9999

Appendix C: To Be Determined List

1. Determine the specific performance metrics.
2. Decide on the specific security protocols to be used.
3. Finalize the design of the user interfaces.
4. Determine the backup and recovery strategy.