CLOUD LIBRARY PLATFORM

(1)

❖ **Functional feasibility**

- We should identify the functionalities before developing the system and should figure out the possibilities of doing those processes.
    - A system administrator can change data present in the system.
    - A library admin can list books for sale.
    - A new user can register on the system themselves.
    - A user can reserve a book.
    - A user can lease a book.
    - A user can read preview of the books.
    - A user can make payment and order the books.

❖ **Technical feasibility**

- This section describes the technical specs that the system should have.

❖ **Financial feasibility**

- We calculate the below expenditures.
    - ➢ The cost of operations of the existing manual library system.
    - ➢ The cost of operations of the proposed internet-based system.
        - The cost of creating the suggested system.
        - Concerning the advantages of the suggested system.
- So, according to the above details, it reveals that the new system is more profitable than the existing manual system.

❖ **Operational feasibility**

- This describes how easy it is to use this system.
- This is an easy-to-use, simple application that can be used anywhere at any time.
- So, this is a user-friendly application.
- This system can be easily used by a person who does not have any computer literacy too.

❖ **Organizational feasibility**

- The system will be established with the support of the organizational staff.
- The system engineers will provide training to the library staff.
- Due to the easiness of this system, all users can use the system without any trouble.
- A user manual will be provided to the new users.

❖ **Human resources feasibility**

- In this section, we discussed the person who interacts with the admin side of the system.
- This describes the people who participate in the system's activities.
- The delivery people and other staff enrollment with the system by updating the books' info, users' info. and others.

---

(3)

Main scenarios :

- **User Registration and Login.**
  1. Fill in the registration form.
  2. Make a username and a password.
  3. Verify the user account by email.
  4. Make use of the provided username and password to log in to the system.

- **Add a book to the system.**
  1. The system administrator should fill out the detailed form for adding the book to the system.
  2. Make up a book ID.
  3. Update the database.

- **Issue a book**
  1. The system administrator has to sign in.
  2. The administrator (librarian) should see the requests made by readers (users).
  3. Then the administrator should release the books if it is possible.
  4. The admin should see the available book list to see the possibility of releasing the book.
  5. After releasing the book, the admin should update the database.

Alternative Scenarios :

- **Check the availability of a book**
  1. The user should login to the system.
  2. Search for the book by its name or according to another method.

3. See the book status.

- **Reset the user password**

1. Click on the “Forgot password” button.
2. Fill in the answers to the security questions asked.
3. Verify it using the email.
4. After that, type in a new password.

- **See the due date**

1. First, log into the library system.
2. Then, see your borrowed book list.
3. There is the date you borrowed and the date you should bring it back.

---

(4)

The waterfall model is used here. Because this is a linear process.

1. **Requirement gathering, analysis and definition.**

- Requirements for the library management system are collected by using the following methods.

  - Questionnaire
  - Interviews
  - Prototypes
  - Random people interviewed
  - Observation

- Then those requirements were categorized by requirements analysts according to functional and non-functional requirements basically. By studying requirements, the system’s basic functions and services were established. As well as the system’s constraints were established by studying and analysing requirements gathered.

2. **System and software design.**

- By establishing the overall system architecture, the technologies and developers allocate the requirements to either hardware or software systems.
- The architecture team creates the architectural diagrams and design documents.
- The developers use Java Script, MySQL and other languages in this stage.

3. **Implementation and unit testing**

- The issues in the system are solved here.
- Here, the system components are developed separately. After the development, they were checking for bugs.

- This process was done repeatedly until all the bugs disappeared.
- Each requirement is tested one by one.
- The validation process is responsible for fulfilling the customer's requirements.

4. Integration and system testing

- The separately developed components are integrated by the testing engineers and tested in an integrated hardware and software environment. Here, check whether the system is working according to the specific requirements. This is done by testing engineers and experts.

5. Operation and maintenance.

- While the system is released to users, the users' new requirements and the issues that occur at this stage are resolved.

SRS DOCUMENT

FOR

CLOUD LIBRARY PLATFORM

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this application is to Create a platform resembling an e-commerce website where users can browse, borrow, and manage books online. This document gives a detailed description of the library management system according to the client's mentioned way. This system manages the library participants. This system manages the books that the library has and the borrowed ones. Readers can see what the available books are and can also reserve books without physically visiting the library. All the activities inside the library can be maintained through this system. This application was developed after a number of research studies.

1.2 Document Conventions

IEEE standards are followed.

- Main heading: Calibri, Bold
- Sub-heading: Calibri, Bold
- Main points: Calibri, Regular, Using bullets or underlined
- Writing: Calibri, Regular

The following is the list of conventions used in this system.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document is the system designer, the administrative staff of the library, the system developer, tester, the system analysis, software architect, maintenance engineer, and users who are directly or indirectly involved with the project.

1.4 Product Scope

This system basically updates a on-premises library management system into an internet-based system by allowing any person who has a mobile device to be logged into the system. So, users can know the availability of the books that they require and can reserve them online through their mobile phones, without physically visiting the library.

1.5 References

Websites:

- www.wikipedia.org
- https://dipeshagrawal.files.wordpress.com/2018/07/srs-library-management-system.pdf
- https://studylib.net/doc/25528819/srs-document-format
- https://www.slideshare.net/ANASNAIN/17337071-srslibrarymanagementsystem
- https://www.geeksforgeeks.org/use-case-diagram-for-library-management-system/
- http://users.csc.calpoly.edu/~csturner/courses/308w09/Ch9_DocReqts.pdf
- https://sari-energy.org/wp-content/uploads/2016/07/SumitPresentation.pdf

Books:

- Software Engineering, Seventh Edition Ian Sommerville.
- IEEE Std. 830-1998: IEEE Recommended Practice for Software Requirements Specifications.
- SRS document of “LIBRARY MANAGEMENT SYSTEM FOR St. JOHN'S COLLEGE JAFFNA”.

2. Overall Description

2.1 Product Perspective

The main objective of this project is to create an innovative, user-friendly online library platform that mimics the operational model of cloud kitchens, but for books. This platform will offer users the convenience of browsing, borrowing, and returning books seamlessly, much like how e-commerce websites provide shopping experiences. By integrating a digital library with advanced search capabilities, personalized recommendations, and a simple borrowing system, the platform will enable users to easily access physical or digital books on demand. It will also streamline logistics, allowing for the delivery and return of books in a way that enhances user satisfaction and promotes wider access to books. The platform will aim to offer a modern, scalable solution for book lovers, libraries, and private collections, making reading and accessing books more flexible and accessible.

2.2 Product Functions

This system is used mainly by the librarian and the readers. The following features are available,

Anonymous User:

- Can view books according to each category.
- Can create an user account.
- Browse content.

Login Users:

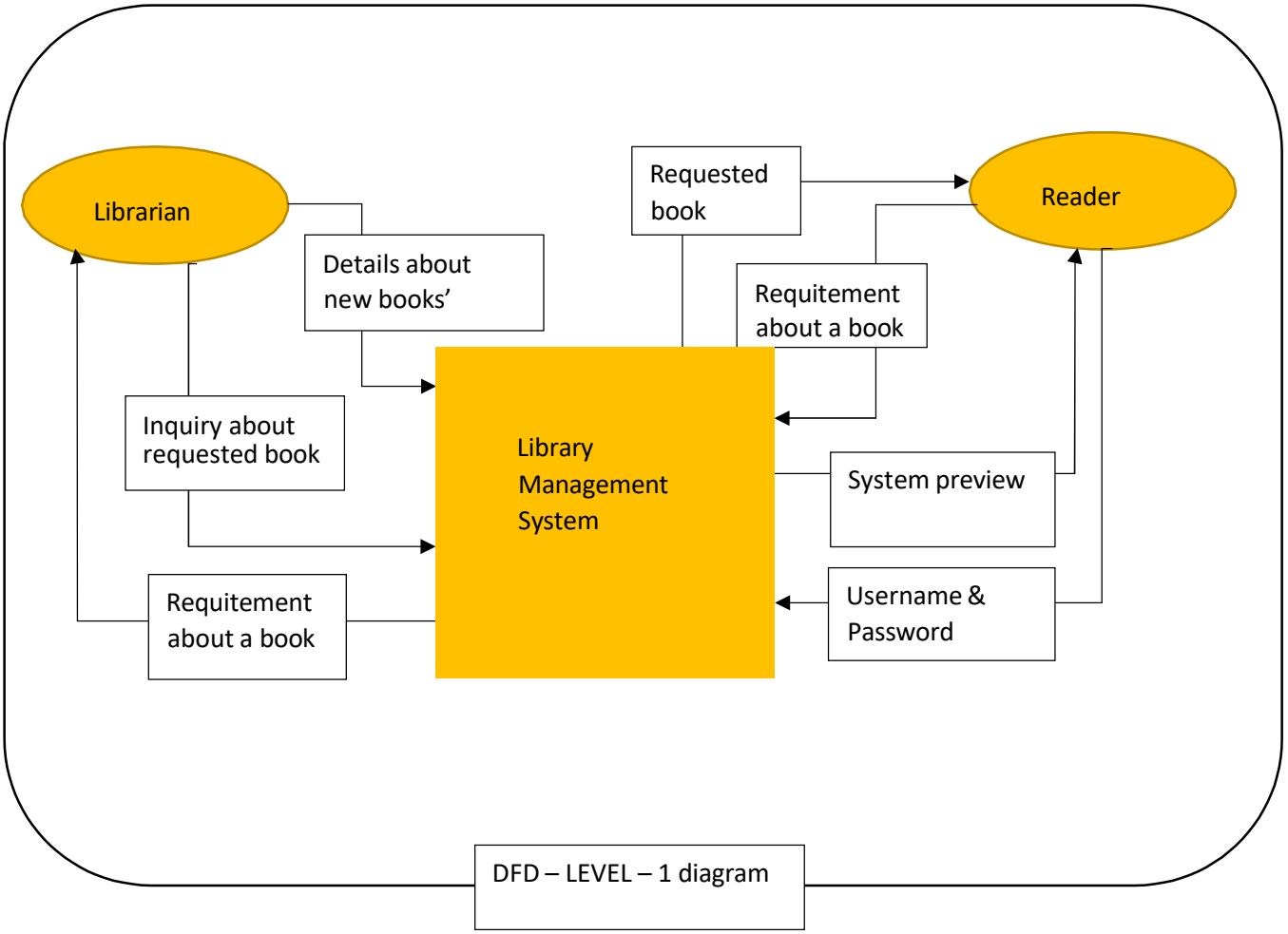
- Can view each category of books available in the library.
- Can view the available books category wise.
- Can buy / lease a book via digital transaction.
- Can put in a request for a new book.

System Admin:

- All privilages for configuration data access and edit based on requirement.

Library Admin:

- Login users' abilities are inbuilt.
- Can post books for sale / lease.
- Can create a librarian account.



DFD – LEVEL – 1 diagram

2.3 User Classes and Characteristics

There are two different users who will be using this product.

- Library Admin handles the book management part.
- The readers who are accessing the platform online.

Features available to the library admin are,

- A librarian can add their library book along with its necessory details.
- Can view the listed books in the platform in different categories.
- Can edit the books available in each category in stock.
- Can participate in financial transaction and set the price of a book.
- Can look at thelibrarian report panel for detailed view of all reports.
- Can interact with support team via in-built messages.

Features available to users (readers) are,

- Can view each category of books available in the library.
- Can create their own account in the library.
- Can view the books issued to him previously.
- Can view the history of the previously issued books.
- Can search for a particular book.
- Can view the details of the book and its preview.
- Can put in a request for a new book.
- Can pay for the book in sale / lease.

2.4 Operating Environment

The system should have a high-speed internet connection and the ability to run in any popular web browser (Microsoft Internet Explorer, Google Chrome, Mozilla Firefox).

Hardware requirements for the server PC

- RAM greater than 4GB
- HDDs larger than 500GB
- Basic input and output devices (monitor, mouse, keyboard, printer)

Software requirements for the server PC

Use Windows 7 or higher as an operating system, or Linux.

2.5 Design and Implementation Constraints

This product was developed using Tech-Stack mentioned in project proposal.

This system can also run-on Android devices.

The backend database for this is MySQL.

This product uses a login feature to make certain functionalities available to just certain readers.

2.6 User Documentation

There is a user manual that we have provided to you.

2.7 Assumptions and Dependencies

The assumptions

- The coding should be error free
- The system should be user-friendly so that it is easy to use for the users.
- The information about users and books must be stored in a way that can be easily accessed through the internet.
- The system should have enough storage capacity and provide fast access to the database.
- The system should provide a search facility and support quick transactions.
- The library system is open 24 hours a day.
- Users may access the site from any computer that has Internet browsing capabilities and an internet connection.
- Users must have their correct usernames and passwords to enter into their online accounts and take actions.
- The hardware and the network connection never fail.

The dependencies

- The specific hardware and software requirements for the product will be tested.
- On the basis of listing requirements and specifications, the project will be developed and tested.
- The end users (admins) should have a proper understanding of the product.
- The information of all the users must be stored in a database that can be easily accessed by the library system.
- Any update regarding the book from the library is to be recorded in the database and the data entered should be correct.

3. External Interface Requirements

3.1 User Interfaces

The system provides a complex platform for users to enter account information and login to their accounts. For users who enter the wrong account information, an error will pop up to show that. New users who want to register in the library and become members can simply sign up. If a user forgets their password, they can retrieve their password via a proper secure method.

This system produces a graphical user interface for both the user and the library admin.

- It makes it easier to view quick reports like books sold, issued and returned in a particular time period.
- It provides stock verification and a search facility for various criteria.
- The design should be straightforward.
- A standard template should be followed by all the different interfaces.
- The user management module should be able to interact through the user interface.

Login Interface: -

If the user has not yet registered, he can enter his information and register to create an account. Once his account is created, he can 'Login', which asks the user to type in his username and password. If the user enters either their username or password incorrectly, then an error message appears.

Search: -

The member or librarian can enter the type of book he is looking for and the title he is interested in, and then he can search for the required book by entering the book name, categories, and ISBN.

View: -

The Category view shows the categories of books available and provides the ability for the librarian to add, edit or delete a category from the list. librarians.

Control Panel: -

This control panel will allow librarians to add or remove users; add, edit, or remove a resource; and manage lending options.

3.2 Application Program Interface

The REST API is a critical component of the Online Library Platform, enabling seamless communication between the frontend, backend, and external systems.

3.2.1. API Overview

The REST API will provide endpoints for:

- User authentication and profile management.
- Book catalog operations (search, filter, view details).
- Borrowing and returning books.
- Admin inventory management.

- Payment integration.
- Logistics and tracking.

3.2.2. Design Principles

Stateless: All requests will be independent, with required data included in each call.

JSON Format: Input and output data will use JSON for simplicity and consistency.

Authentication: Use token-based authentication (e.g., JWT) for secure access.

3.3.3. Error Handling

Standardized HTTP Status Codes:

200: Success.

201: Resource created.

400: Bad request (e.g., invalid parameters).

401: Unauthorized (e.g., invalid token).

404: Not found (e.g., invalid endpoint or resource ID).

500: Internal server error.

3.3.4. Authentication

Method: JWT-based authentication.

Headers:

Authorization: Bearer <token> for protected endpoints.

3.3.5. Rate Limiting

Purpose: Prevent abuse and ensure fair usage.

Limit: 100 requests per minute per user.

3.3 Hardware Interfaces

Server side:

- Operating system: Windows 7 or higher or Linux
- RAM greater than 4GB
- HDDs larger than 500GB

Client side:

- Operating system: Windows 7 or higher, Linux, MAC, Android 5.1 or higher
- RAM greater than 4GB
- HDDs larger than 500GB

3.4 Software Interfaces

Software Interfaces for Online Library Platform
This section outlines the primary software interfaces that the platform will integrate with to achieve its functionality.

3.4.1. Frontend Interface

Type: Web and Mobile Applications
Purpose: Provide an intuitive user interface for accessing library services.
Technology:
Frameworks: Js frameworks.
Communication: REST API calls to the backend.
Integration: Frontend interacts with backend APIs for user authentication, book browsing, borrowing.

3.4.2. Backend Interface

Type: RESTful API Service
Purpose: Acts as the central logic layer, managing data flow and business rules.
Technology: Java (Spring Boot / Basic Servlets).
Integration: Connects to the database, third-party services, and frontend interfaces for processing requests and responses.

3.4.3. Database Interface

Type: Relational Database
Purpose: Store and manage data such as user profiles, book inventory, borrowing records, and payments.
Technology: MySQL.
Integration: The backend queries the database using ORM (e.g., Sequelize, SQLAlchemy) or direct SQL commands.

3.4.4. Payment Gateway Interface

Type: Third-Party Payment Integration
Purpose: Handle secure transactions for borrowing fees, deposits, or purchases.
Examples: Stripe, PayPal, Razorpay.
Integration: APIs are used for payment initiation, status verification, and refund processing.

3.4.5. Authentication Interface

Type: Security Layer
Purpose: Ensure secure access to the platform via user authentication.
Technology: JWT (JSON Web Tokens) or OAuth 2.0.
Integration: Frontend communicates with the authentication service through API endpoints to obtain tokens for accessing secure resources.

3.4.6. Search and Recommendation Interface

Type: Internal or Third-Party Service
Purpose: Provide book search functionality and personalized recommendations.
Technology: Elasticsearch for search; AI/ML models for recommendations.
Integration: Backend interacts with search indices or recommendation engines to fetch and display results.

3.4.7. Reporting and Analytics Interface

Type: Internal or External Analytics Tools
Purpose: Generate insights and reports on user behavior, borrowing trends, and inventory usage.
Examples: Google Analytics for frontend metrics, custom dashboards for backend data.
Integration: Backend collects and processes data for visualization tools or analytics services.

3.4.8. Cloud Storage Interface

Type: Cloud-based File Storage
Purpose: Store digital copies of books, user documents, or logs.
Technology: AWS S3, Google Cloud Storage, or Azure Blob Storage.
Integration: APIs are used to upload, retrieve, and manage files securely.

3.4.9. Email and Notification Interface

Type: Messaging Service
Purpose: Notify users about transactions, due dates, or promotions.
Examples: Twilio for SMS, SendGrid or Amazon SES for email notifications.
Integration: Backend triggers notifications through these services based on user actions or events.

4. System Features

This section shows the features, priorities, requirements, and a few other things.

4.1 System Login

4.1.1 Description and Priority

Here, the user's entered username and password have been authenticated.

Priority level: Very high

4.1.2 Stimulus/Response Sequences

5. The user runs the system.
6. The system displays the login page.
7. The user inputs their username and password and clicks on the “login” button.
8. The system authenticates the user’s validity.

4.1.3 Functional Requirements

- The system should only allow users with valid IDs and passwords to enter the system.
- The system should perform an authorization process that decides what level of access each user can access.
- The user should logout after they finish using the system.

4.2 Search book

4.2.1 Description and Priority

Users can search for and browse books using filters such as genre, author, or rating.

Priority level: Very high

4.2.2 Stimulus/Response Sequences

1. The user navigates to the search or browse page.
2. The system displays a search bar and filter options.
3. The user inputs keywords or selects filters.
4. The system retrieves and displays relevant results.

4.2.3 Functional Requirements

- The system should allow keyword-based search functionality.
- Filters should include genre, author, availability, language, and user ratings.
- Search results should display book details like title, author, genre, and availability status.

4.3 Register a new user

4.3.1 Description and Priority

Here, a new user can register on the system.

Priority level: Very high

4.3.2 Stimulus/Response Sequences

1. The new user should enter into the system.
2. The system displays the “Register” button.
3. The "Register" button should be selected by the user.
4. The system displays the relevant details for you to fill in.
5. The user should fill in the details asked and click on the "Register" button.
6. The system should save the user and should provide a user name.

4.3.3 Functional Requirements

- The system should provide a “Username” and “Password” to the user.
- The system should open a new database for the user and save the user’s data in it.

4.4 Issue a book

4.4.1 Description and Priority

Here, a user can borrow a book from the system.

Priority level: Very high

4.4.2 Stimulus/Response Sequences

1. The user should sign into the system.
2. The system shows its “home page.”
3. The user can select to “borrow books”.
4. The system displays a list of books.
5. The user should select a book.
6. The system should show relevant details and forms to fill in.
7. The user should fill out the application or the form given.

8. The system should issue the book after payment.

4.4.3 Functional Requirement

1. The system should verify the book's availability before processing a borrowing request.
2. Borrowing durations should be customizable based on user input.
3. The system should update inventory and track the borrowed book's status.

4.5 Book Return

4.5.1 Description and Priority

Users can initiate the return process for borrowed books.

Priority level: High

4.5.2 Stimulus/Response Sequences

1. The user navigates to the “My Borrowed Books” page.
2. The system displays a list of currently borrowed books.
3. The user clicks on the "Return" button for a specific book.
4. The system updates the book’s status and schedules a pickup if needed.

4.5.3 Functional Requirements

1. The system should track due dates and notify users about late returns.
2. Users should be able to schedule pickups for returns.
3. The system should update inventory upon return.

4.6 User Profile Management

4.6.1 Stimulus/Response Sequences

1. The user navigates to the "Profile" page.
2. The system displays the user’s profile details.
3. The user updates their information and saves changes.
4. The system validates and updates the data in the database.

4.6.2 Functional Requirement

1. The system should allow users to update personal details such as name, email, and address.
2. Users should be able to view their borrowing history and wishlist.
3. The system should validate data before saving changes.

4.7 Payment Processing

4.7.1 Description and Priority

The system processes payments for borrowing fees or deposits securely.
Priority level: Very High

4.7.2 Stimulus/Response Sequences

1. The user selects a borrowing option that requires payment.
2. The system displays payment options.
3. The user inputs payment details and confirms the transaction.
4. The system processes the payment and updates the transaction status.

4.7.3 Functional Requirements

1. The system should integrate with payment gateways (e.g., Stripe, PayPal).
2. Payment transactions should be secure and encrypted.
3. Users should receive a confirmation receipt after payment.

4.8 Notifications and Alerts

4.8.1 Description and Priority

The system sends notifications about due dates, promotions, and account activity.
Priority level: Medium

4.8.2 Stimulus/Response Sequences

1. A user action or system event triggers a notification.
2. The system composes and sends the notification.
3. The user receives the notification via email, SMS, or in-app alert.

4.8.3 Functional Requirements

1. The system should support customizable notification preferences.
2. Notifications should be sent for due dates, late fees, and promotional offers.
3. Users should be able to opt-in or opt-out of specific notification types.

4.9 Admin Inventory Management

4.9.1 Description and Priority

Admins can manage book inventory, including adding, updating, and removing books.
Priority level: High

4.9.2 Stimulus/Response Sequences

1. The admin logs in and navigates to the inventory page.
2. The system displays the current inventory.
3. The admin performs an action (e.g., add, update, or delete a book).
4. The system processes the action and updates the inventory.

4.9.3 Functional Requirements

1. The system should allow bulk uploads of book records.
2. Inventory updates should reflect immediately in the catalog.
3. The system should track the history of inventory changes for auditing.

4.10 Buying Books

4.10.1 Description and Priority

Users can purchase books directly from the platform.
Priority level: High

4.10.2 Stimulus/Response Sequences

1. The user selects a book and clicks the "Buy" button.
2. The system adds the book to the user's cart.
3. The user proceeds to checkout and enters payment and shipping details.
4. The system processes the payment and confirms the order.
5. The user receives order details, and the book is shipped.

4.10.3 Functional Requirements

1. The system should provide a shopping cart for book purchases.
2. Users should be able to select payment methods and enter shipping information.
3. The system should track order history and provide shipping updates.
4. Admins should manage available stock for books sold directly.

4.11 Dashboard Service for Library Admin

4.11.1 Description and Priority

Library admins can access a dashboard to monitor platform activity, manage users, and view analytics.
Priority level: Very High

4.11.2 Stimulus/Response Sequences

1. The admin logs into their account.
2. The system displays the admin dashboard with key metrics and actionable insights.
3. The admin navigates to specific sections like user management, book inventory, or reports.
4. The system processes the admin's actions and updates data accordingly.

4.11.3 Functional Requirements

1. The dashboard should display analytics such as:
2. Total books borrowed, purchased, and returned.
3. User activity trends (active users, overdue accounts).
4. Inventory stats (available, borrowed, purchased stock).
5. The system should allow admins to:
6. Add, edit, or delete inventory books.
7. Monitor overdue books and send reminders.
8. Generate custom reports (e.g., by date range, user activity).
9. The system should present data visually using graphs, charts, and tables.

5. Other Non-functional Requirements

5.1 Performance Requirements

- Users should be able to use the system 24 hours a day.
- The response time should be less than 2 seconds.

Server side

The proposed system that we are going to develop would meet functionally all the requirements that are specified.

- The performance of the system should be swift and precise.
- The library management system will handle expected and non-expected errors in ways that prevent loss of information and long downtime periods. Thus, it should have inbuilt error testing to identify invalid username and password combinations.
- Large data sets should be manageable for the system. Thus, it should accommodate a high number of books and users without any fault.

5.2 Safety Requirements

The database may crash at any certain time due to a virus or operating system failure. Therefore, it is required to take a database backup.

- The programme will make use of a safe database.
- Normal users can modify only their personal and some other information.
- The system will have different types of users and every user has access constraints.
- Proper user authentication should be provided.
- No one should be able to hack users' passwords.
- There should be separate accounts for admin and members such that no member can access the database and only the admin has the right to update the database.

5.3 Security Requirements

This is a secured database for the library. There are different user categories. It depends on the category of user how the access rights are decided. It indicates that if a user has administrative privileges, he or she may modify the data, delete, append, etc. All other users, other than library staff, only have the right to retrieve information about the database.

5.4 Software Quality Attributes

The quality of the database is maintained in such a way that it can be very user-friendly to all the users of the database.

Persistent data must be kept in a database by the system. The database ought to be capable of backups. The system's potential to develop will be limited by the lack of necessary software, including web servers, databases, and development tools.

Reliability: - The system has to be fully reliable due to the importance of data and the damage that can be done by incorrect or incomplete data.

Maintainability: - The system should provide automatic notification to patrons by e-mail about items that are overdue, reservation results, availability of reserved items and etc.

Performance: - Database should be updated for better performance. Searching should be fast. Login should be validated within 1 second.

Security: - Payment information should be protected and encrypted. Communication between a server and a user should be secure.

Accuracy: - Real-time information should be accurately provided by the system while accounting for various concurrency problems.

Interoperability: - The digital libraries can exchange and share documents, queries, and services.

Testability: - Testability measures the ability for software to demonstrate its faults. Since a significant amount of money is expended on testing, the system must ensure that the testability of the built system is robust.

5.5 Business Rules

- The readers should borrow books back to the library within X days. If not, the system automatically generates a fine payment. If the user (reader) could not borrow back the book, he should pay Rs. Y per day.

6. Other Requirements

Data and Category Requirements

There are various types of users, such as teaching staff, librarians, administrators, students, and so on. Access privileges are determined based on the type of user.

It indicates that if a user has administrative privileges, he or she may edit, delete, or add data. All other users, except the librarian, only have the right to retrieve information about the database. Similar to how there will be various book genres accessible.

The pertinent information for each category of book should be shown. The categories should be coded in the specific format, as should the data associated with each category.

Ethical Requirements

There is a huge user base for a library management system. The system has some of their details, such as emails. So, the system should work in an ethical manner and should operate in that manner.

Appendix A:

Glossary

The following is the list of conventions and acronyms used in this document and the project as well:

User: An all-purpose login ID given to most users.

Clients: The intended users of the software.

SQL: It stands for Structured Query Language, and it is used to retrieve data from a database.

Layer: Represents a section of the project.

Application Logic Layer: the part of the assignment that mentions the Web Server. This is where all computations are completed.

Data Storage Layer: The section of the assignment referring to where all the data is recorded.

Use Case: A broad-level diagram of the project showing a basic overview

A class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system's cases, their attributes, and the relationships between the classes.

Interface: Something used to communicate across different media.

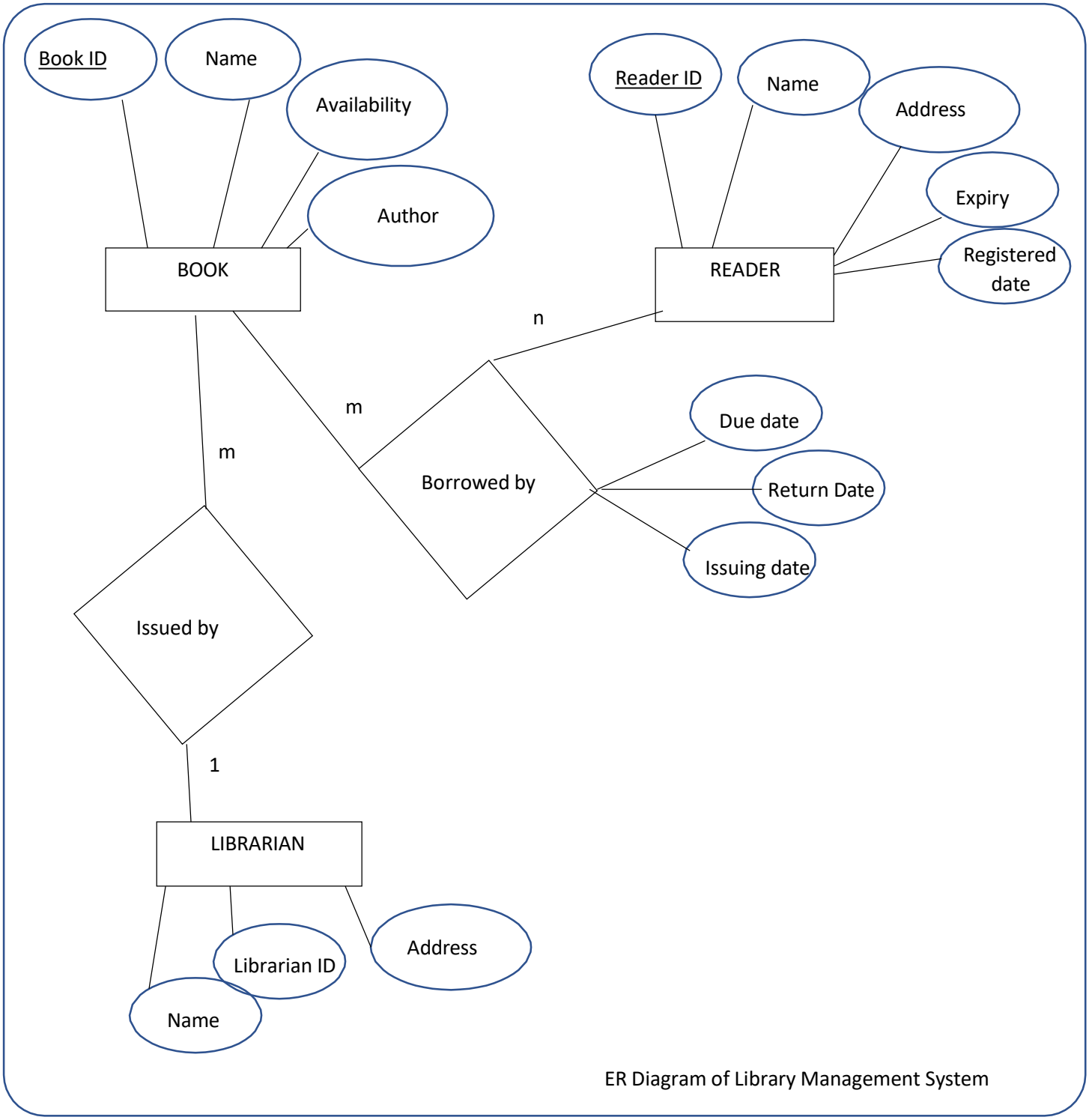
Unique Key: Used to differentiate entries in a database

OS: Operating System

DFD: Data Flow Diagram

ER-Entity Relationship

Appendix B:



ER Diagram of Library Management System

Thank You!!!