**Title: High-Level Design Document - Library Management System**

1. Introduction The Library Management System is a web-based application designed to efficiently manage the books and student records in a library. The system follows a layered architecture approach using Spring MVC framework, consisting of the Controller layer, Service layer, and DAO (Data Access Object) layer. This document provides an overview of the high-level design of the system.
2. System Architecture The Library Management System follows a three-tier architecture, including the following layers:

* Presentation Layer: This layer handles user interactions, receives requests, and sends responses. It consists of the Spring MVC Controllers responsible for mapping incoming requests and invoking the appropriate service methods.
* Business Layer: Also known as the Service Layer, this layer contains the business logic of the application. It handles data validation, manipulation, and coordination between the Controller and DAO layers. It communicates with the DAO layer to perform CRUD (Create, Read, Update, Delete) operations on the data.
* Data Access Layer: The DAO layer interacts directly with the database and performs operations such as retrieving, storing, and updating data. It uses Spring Data JPA to provide an abstraction layer for database operations.

1. Component Overview 3.1. Controller Layer

* HomeController: Handles requests related to the home page and general navigation.
* StudentController: Handles requests related to student operations, such as registration, login, and fine payment.
* BookController: Handles requests related to book operations, such as searching, viewing, and CRUD operations.

3.2. Service Layer

* StudentService: Implements the business logic related to student operations, including registration, login, fine calculation, and payment.
* BookService: Implements the business logic related to book operations, including searching, viewing, and CRUD operations.

3.3. DAO Layer

* StudentDAO: Provides methods to perform CRUD operations on student records in the database.
* BookDAO: Provides methods to perform CRUD operations on book records in the database.

1. Data Model The Library Management System uses the following entities and their relationships:

* Student: Represents a student registered in the system. Contains attributes such as student ID, name, email, password, and fine amount.
* Book: Represents a book in the library. Contains attributes such as book ID, title, author, category, availability status, and issued date.

1. Technology Stack

* Spring MVC: Used for the presentation layer, including Controllers.
* Spring Boot: Provides a framework for building the application and managing dependencies.
* Spring Data JPA: Offers a convenient way to access and manipulate data from the database.
* MySQL: The database management system used to store student and book records.

1. Conclusion The High-Level Design of the Library Management System incorporates a layered architecture approach with separate components for the Controller, Service, and DAO layers. The Controller layer handles user interactions and routes requests to the appropriate Service methods. The Service layer encapsulates the business logic and interacts with the DAO layer to perform CRUD operations. The DAO layer communicates with the database using Spring Data JPA. The technology stack includes Spring MVC, Spring Boot, Spring Data JPA, and MySQL. This design allows for separation of concerns and promotes maintainability and scalability of the system.