

LIBRARY MANAGEMENT SYSTEM

The main objective of the project is to maintain the books in a proper manner and distribute in a simple way. This project is useful for institutions for distributing the books in simple manner. By using this application, we can reduce the human efforts. Library is a place where all kinds of books are available which aims in developing a computerized system to maintain all the daily work of library. This project has many features like facility of Librarian login which is used to perform the activities like add book, delete book, issue book, search book, return book according to the requirement of students and admin login which is used to perform the activities like add librarian, view librarian and delete librarian. Overall, this project of ours is developed to help the librarians of library to maintain books in the best way possible and to reduce the human efforts and help the students. In this Abstract, an automated platform for managing result of all categories of books in a seamless and interactive manner is presented. The system was developed using Java, Spring Boot, HTML, CSS, MYSQL and hosted locally using our favorite browsers like Chrome, Internet Explorer and we presented the content in a beautiful and effective way. Finally, the conclusion is that the project clearly depicts the Library Management System efficiently. It also shows how the concept of Database Management System is used like MYSQL etc.

DESIGNING APPLICATIONS

a) UML Diagrams

- Use Case
- Class Diagram
- ER Diagram
- Sequence Diagram

b) Three – Tier Architecture

- Front end
- Middle ware
- Back end

Front end

- HTML
- CSS
- JavaScript (Js)

Middle ware

- Spring Boot: Rest Controller, JPA and Security, Junit.

Back end

- Database: MySQL

c) Testing

- J unit

d) Software

- STS

MODULES

- ADMIN
- LIBRARIAN

ADMIN MODULE

- **CLASSES:**

- Admin
- Librarian

- **CONTROLLERS:**

- AdminLoginController
- AddLibrarianController
- ViewLibrarianController
- DeleteLibrarianController
- AdminLogoutController

- **VIEWS (JSP):**

- a. index
- b. adminLogin
- c. adminSection
- d. addlibrarian
- e. viewLibrarian
- f. deleteLibrarian

- **TABLES:**

- a. Admin
- b. Librarian

- **MAPPING:**

- a. AdminLoginController -> adminlogin.jsp -> adminsection.jsp
- b. AddLibrarianController -> addlibrarian.jsp -> adminsection.jsp
- c. ViewLibrarianController -> viewlibrarian.jsp -> adminsection.jsp
- d. DeleteLibrarianController -> deletelibrarian.jsp -> adminsection.jsp
- e. AdminLogoutController -> index.jsp

LIBRARIAN MODULE

- **CLASSES:**

- a. Book
- b. Issue Book
- c. Student

- **CONTROLLERS:**

- a. LibrarianLoginController
- b. AddBookController
- c. DeleteBookController
- d. ViewBookController
- e. IssueBookController
- f. ViewIssuedBookController

- g. ReturnBookController
- h. LibrarianLogoutController

- **VIEWS (JSP):**

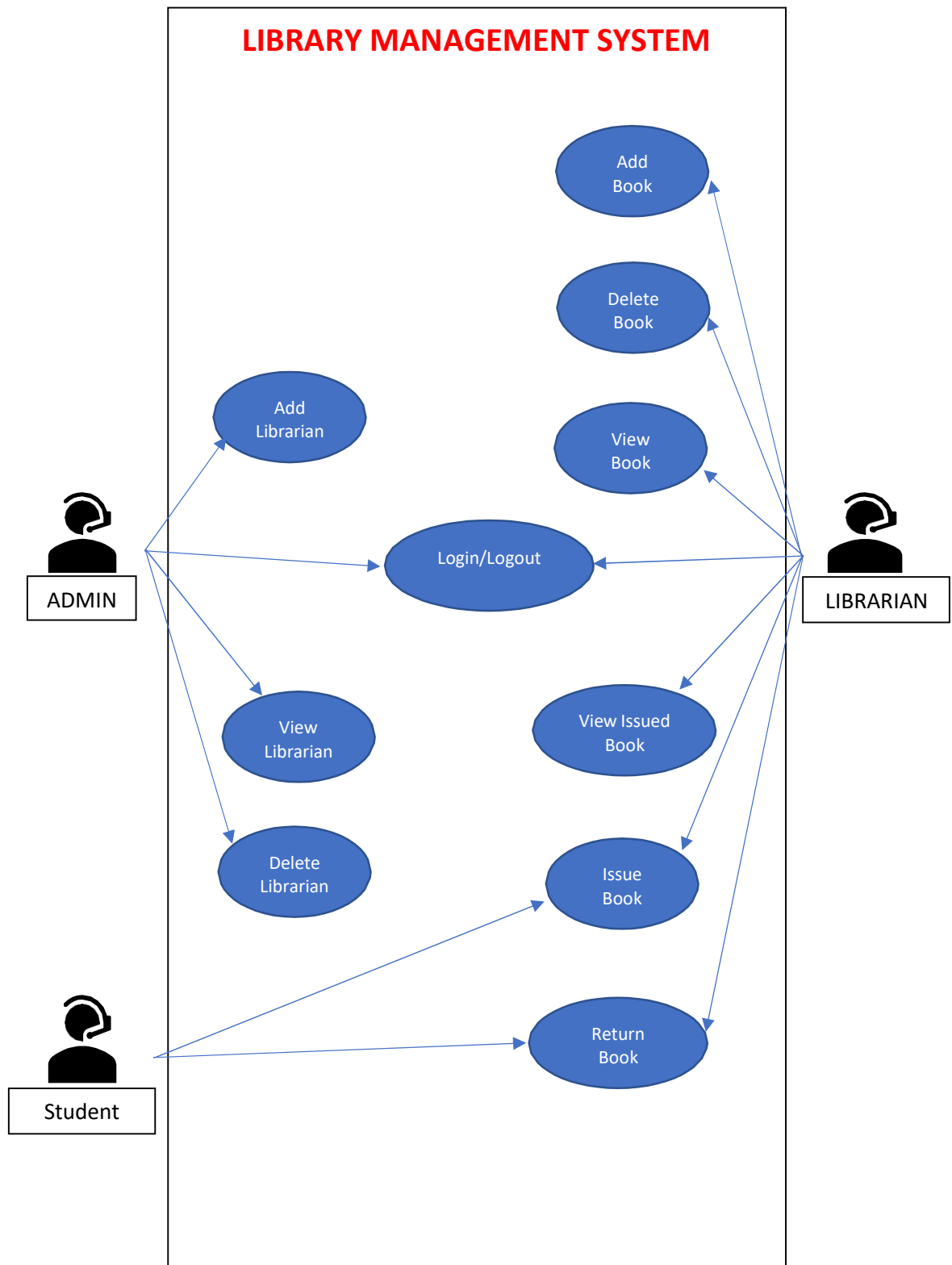
- a. index
- b. librarianLogin
- c. librarianSection
- d. addBook
- e. viewbook
- f. issueBook
- g. viewIssuedBook
- h. returnBook
- i. deleteBook

- **TABLES:**

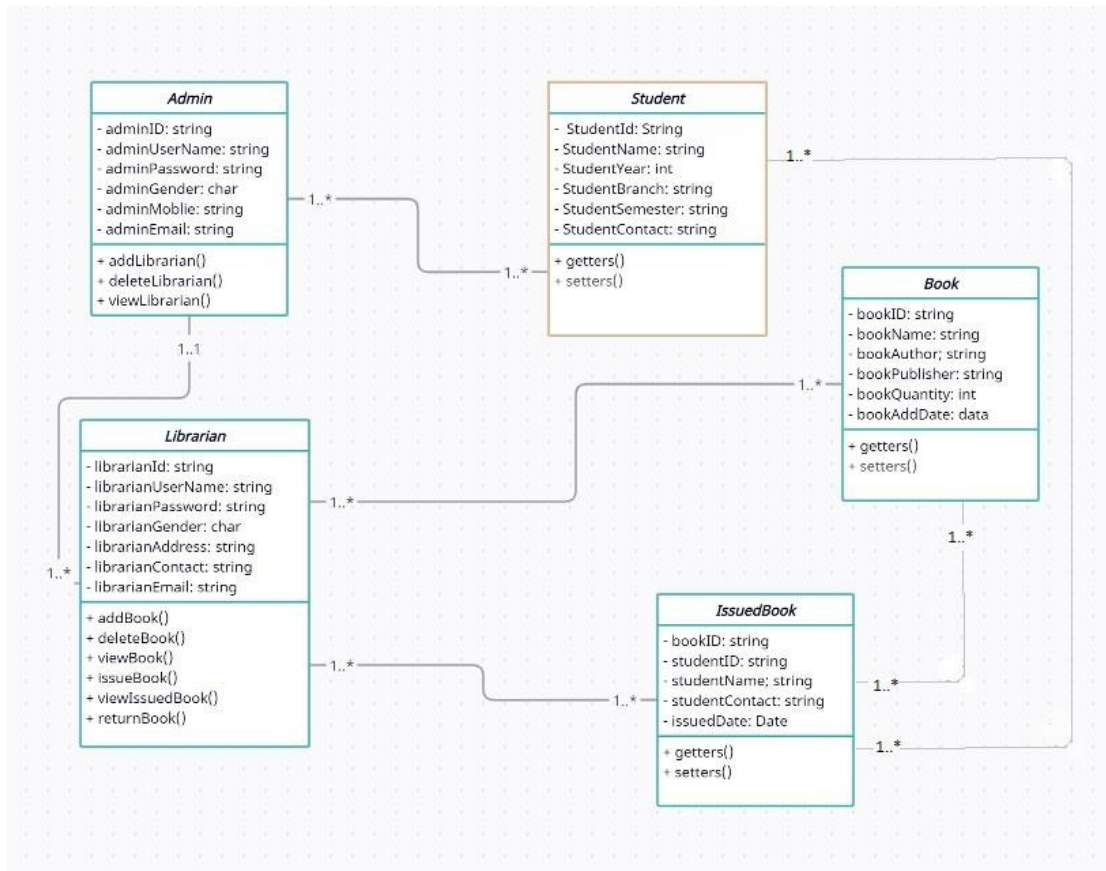
- a. Book
- b. IssueBook
- c. Student

- **MAPPING:**

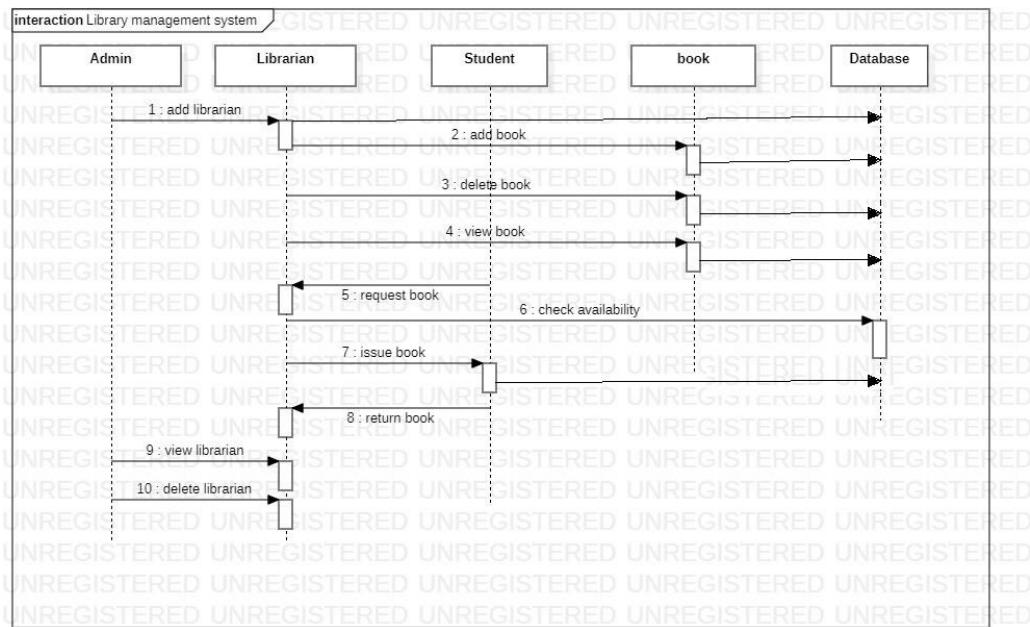
- a. LibrarianLoginController -> librarianlogin.jsp -> librariansection.jsp
- b. AddBookController -> addbook.jsp -> librariansection.jsp
- c. DeleteBookController -> deletebook.jsp -> librariansection.jsp
- d. ViewBookController -> viewbook.jsp -> librariansection.jsp
- e. IssueBookController -> issuebook.jsp -> librariansection.jsp
- f. ViewIssuedBookController -> viewissuebook.jsp -> librariansection.jsp
- g. ReturnBookController -> returnbook.jsp -> librariansection.jsp
- h. LibrarianLogoutController -> index.jsp



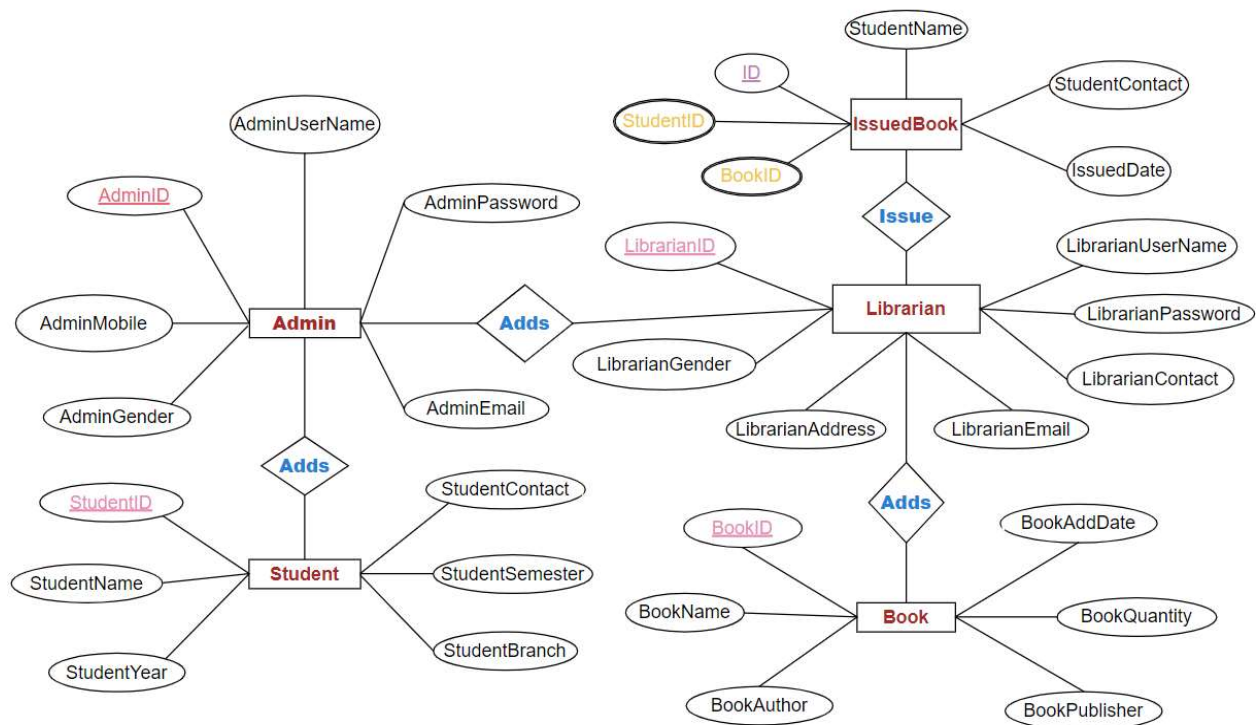
USE CASE DIAGRAM



CLASS DIAGRAM



SEQUENCE DIAGRAM



ENTITY-RELATIONSHIP(ER) DIAGRAM