**Library Managment System**

**Intoduction :-**

The main objective of the Library Management System is a project that manages and stores books information electronically according to students needs. The system helps both students and library manager to keep a constant track of all the books available in the library. It allows both the admin and the student to search for the desired book. It becomes necessary for colleges to keep a continuous check on the books issued and returned and even calculate fine. This task if carried out manually will be tedious and includes chances of mistakes. These errors are avoided by allowing the system to keep track of information such as issue date, last date to return the book and even fine information and thus there is no need to keep manual track of this information which thereby avoids chances of mistakes.  
Thus this system reduces manual work to a great extent allows smooth flow of library activities by removing chances of errors in the details.

**Software Requirement :-**

#### Java : Jdk-18, Spring Tool Suite

* + - Database : MongoDb
    - Server : Web Server, Postman
    - Testing : Junit5

### 

### Hardware Requirement :-

* + - Processor : Core i3 and Above
    - RAM : 4GB
    - Hard disk capacity : 500 GB

**Service :-**

1)Registration/Login

2)Admin

3)User

4)Eureka

5) Api Gateway

6)Book Borrow

7)Fine Calculate

**Diagram :-**

1. **UML Diagram**

**Diagram

Description automatically generated**

**2)Control Flow Diagram**

**Diagram

Description automatically generated**

**3)User Flow**

**Diagram

Description automatically generated**

**Architecture diagram of library management system:**

**Diagram

Description automatically generated**

**User Story :-**

**Modules:**

* **Registration / Login:**
* **Admin login:** Admin is the one who administers the system by adding or removing e-books into and from the system respectively.
* **User login:** Students have to register themselves into the system to create an account. After registering successfully, they can then login into the system by entering 10 digit mobile number and their email id.
* **User** can Search book into a library by using curd operation.
* **Add and Update Books:** The admin can add books to the system by entering the details of the books and can even update the details.
* **Search option:** Admin and user can even search for books by entering the name of the book.
* **Eureka Services:**
* **Api Gateway:**

**UI Screen Shots(Wire Frame) :-**

**Graphical user interface, text, application, email

Description automatically generated**

**Code Coverage:-**

**1)Api\_GateWay\_Service**

**Graphical user interface, text, application

Description automatically generated**

**2)Eureka\_Service**

**Graphical user interface, text, application

Description automatically generated**

**3)Admin\_Service**

**Graphical user interface, text, application

Description automatically generated**

**Futher Enhancements :-**

This project is a library management system with all the basic as well as some innovative features for managing a library(ebook mailing and expiry notification via sms).

It consists of a large database of various books available in the library. It also lists various books issued to respective readers.

The system keeps track of all the books readily available and also the books that have been issued to various readers the time period for which the books have been issued. The system also handles ebooks database.

If the reader needs an ebook, the ebook can be directly mailed to the client email id through the system through a single click.

Readers usually tend to forget the date to resubmit their library books so these books that have been issued to the reader and have reached their expiry date, an sms notification is sent to the reader automatically by the system to remind him about the book return date.

Thus, this innovative library management system provides an enhanced library functionality for this modern day world.

* **View Order-**The admin can view order for the books.
* **Place order-** The students can place order for the books and simultaneously the quantity of the book ordered will be decremented.
* **Calculate Fine-** The student can view the issue and expiry date for the book issued and can even calculate fine.

**Conclusion :-**

The evidence gathered in this report reflects ongoing progress toward more openness in library automation systems, but also that much work remains. We see a variety of options and opportunities. Libraries that expect to work with their automation system as delivered and not become involved in local extensions or programming will find that the majority of systems were built for that kind of use. For libraries that want to do more with their automation systems, however, we see a great deal of functionality possible today through open interfaces with momentum toward creating much more.

The hype persists. Library automation systems, property and open source alike, compete more and more on the basis of enabling libraries to do more with their systems. That competition for openness drives the development of the technologies that enable that capability. The reality is still a bit messy. While we've seen a great deal of functionality exposed through Web services and other APIs, it still takes a lot of hard work to use the APIs in ways that benefit the library. The APIs available to library programmers continue to be quirky and less than comprehensive, even from the vendors with the strongest offerings in this area. We can also tell by the information received that vendors and libraries alike see the need to make systems more open. Hopefully, a better reality will evolve over time.