Library

Actors:

1. Students
2. Librarians
3. System (Automated)

Domain Objects:

1. Physical Book
2. E-Book (pdf)
3. Package (Collection of Books)
4. Notification

Functional Requirements:

1. **Add/Update/Delete Book Metadata (Physical + E-Book):**
   Actor: Librarian

   The Librarian can add, update and delete book metadata.
   Metadata: (**name**, cover-page, **author names**, **publisher name**, **edition**, **copies available**, **fine/day**, additional key-value pair)
2. **Add/Update/Delete Book Package (Physical + E-Book):**
   Actor: Librarian

   A Package is composed of one or more than one book (a collection of books both physical and e-book). The Librarian can create and manipulate packages
3. **View Package Details (Physical + E-Book):**
   Actor: Librarian/Student

   A Librarian/Student can view the details of a package like a name, book name, etc.

4. **Search Book (Physical + E-Book):**
   Actor: Librarian + Student

   Students can search which books are available for check-out.
   Search Criteria include: book name, author name
5. **Request Book Checkout (Physical + E-Book):**
   Actor: Student

   Students will submit a request to check out a copy of the book in the system. Librarian will either accept the request or reject it. Also, relevant information needs to be tracked in case the Librarian decides to accept the request. (Info: Book id, student id, date of issue, expected date of return).
   NOTE: Only allowed if physical copies are available in the library

6. **Accept Book Checkout Request (Physical + E-Book):**
   Actor: Librarian

   Librarian will allocate a book to a student along with its expected return date.
7. **Reject Book Checkout Request (Physical + E-Book):**
   Actor: Librarian

   Librarian denies check-out request
8. **Accept Physical Book Return (Physical)**:
   Actor: Librarian

   After Students hand over the physical book to the Librarian, the Librarian will mark it as returned. If any fine needs to be paid, the system will show the fine amount,
   (amount depends on the book and extra days taken to return)

9. **Fine Calculation (Physical)**:
   Actor: System

   Every book will contain information about the **rental charges per day**, and that information will be used in the calculation of the fine.
   Fine = (actual return date - expected return date) * fine/day

10. **Display Recent Book Checkout Requests (Physical + E-Book):**
    Actor: Student, Librarian

    Students and Librarian can check their recently check-out requests.
    NOTE: the Librarian can see all Check-Out requests, but Students can only see requests which are made by themselves.

11. **Cancel Book Checkout Request (Physical + E-Book)**:
    Actor: Student

    Students can cancel a check-out request made, only if it is a PENDING state. A new check-out request starts with a PENDING state.

    States: PENDING, ACCEPTED, REJECTED

12. **Book Return Reminder Notification (Physical)**:
    Actor: System

    The system will automatically keep sending reminders to Students to return a checked-out book at regular intervals.
    Interval: (before expected return date): (7,3,1)

13. **Book Request Accept/Reject Notification (Physical + E-Book)**:
    Actor: System

    The System will push notifications to Students when Librarian accepts/rejects a book check-out request along with a custom message.

14. **Read Ebook Pages:**
    Actor: Student, Librarian

    Students and Librarians can view the ebook pages available Librarians can view any ebook, but the Student can only view the ebook which is allotted to them.

15. **Allocation of E-Book:**
    Actor: System

    The system should automatically allocate e-books to Student once the Librarian accepts a book checkout request.

16. **Deallocation of E-Book:**
    Actor: System

    The system should automatically deallocate e-books to Student once the expected return date is crossed.

    //Batch Insert Book Metadata (CSV)

Tech Stack:
BackEnd: Spring Boot
Database: PostgreSQL + MongoDB + ElasticSearch
Cache: Redis
Event Store: Kafka

Data Dictionary: (Underline means required)

Book:

1. id
2. name
3. authorName[] (empty not allowed)
4. publisherName
5. edition
6. coverPageImageId
7. tags[] (Key-Value Pair) (empty allowed)

PhysicalBook: Book:

1. copiesAvailable
2. finePerDay

EBook: Book:

1. readOnlineUrl

BookPackage:

1. Id
2. name
3. bookIds[] (empty allowed)

BookLeaseRequest:

1. bookId
2. userId
3. status (PENDING,ACCEPTED,REJECTED)
4. timestamp

BookLeaseRecord:

1. bookId
2. userId
3. issueDate
4. expectedReturnDate
5. isReturned

Notification

1. id
2. timestamp
3. payload

4. Acknowledged

Report:

- Synopsis
- Activity Plan
- Activity Completed

(PDF Report)

1. Project Synopsis on $projectName
2. Objective (ki korte chaicho)
3. Introduction
4. Plan of Work (ordered (number wise))
5. Proposed Model (flow digram + dfd + use case(optional))
6. Outcome (ki hote pare)
7. Individual responsibilities
8. Actitvity Completed
9. Conclusion
10. References

Last Date:
Document Verification: **29 Nov** (verify with Sayan sir)
Document Submission: **1 Dec**