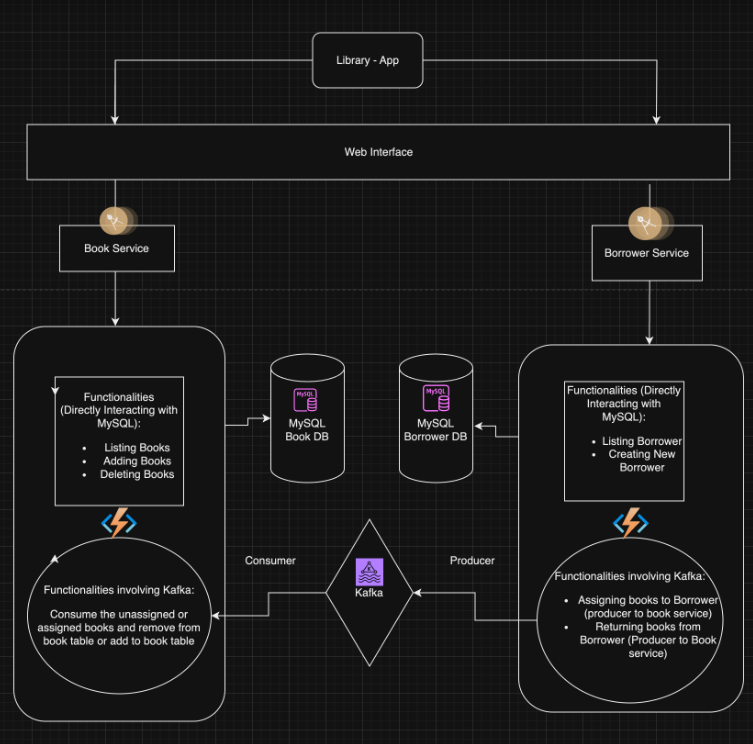
# LIBRARY MANAGEMENT

# SOFTWARE DESIGN DOCUMENT

1. PURPOSE
   * This is a library book management application intended for internal use of library resources (i.e. books). The tool can be used to keep track of books in the library along with keeping track of whether books are still assigned to the user if it is returned by user.
2. TECHNOLOGICAL STACK

* We have used the following technologies while developing the services:
* PLAY Framework
* Scala programming
* Kafka
* MySQL Database
* Google Cloud Platform
* Docker
* Akka Framework

1. SYSTEM OVERVIEW:
   * There are two services involved in my design namely:
     + Book Service
     + Borrower service
   * Book service functionalities:
     + This service is responsible for adding books, listing books and removing books from the library books pool.
     + This service consists of following components Book Controller, Book DAO, Book Model, Book Service and book view.
     + In the book service we have a Kafka consumer defined which is consuming messages from Borrower service on the assignment/return of book.
   * Borrower service functionalities:
     + This service is responsible for listing borrowers, adding borrowers and assigning/unassigning books to borrowers from library books pool.
     + This service consists of following components Borrow Controller, Borrower DAO, Borrower Model, Borrower Service and borrower view.
     + In the book service we have a Kafka producer defined which is producing messages from Borrower service with a flag to Book service whenever there is assignment/return of book.
     + Borrower service
   * Both the services use MySQL database to store the data i.e. for Book Service the books data is stored in book table maintained in MySQL and for Borrower service Borrower table is maintained in my SQL.
2. ARCHITECTURE DIAGRAM:
   * The application follows the below high-level architecture:



* Assumptions and Dependencies:
  + MySQL database is used for persistent storage.
  + Kafka is used for messaging between services.
  + Docker is used for containerization and deployment on GCP.

1. DATA DESIGN:
   1. Data Architecture
      1. MySQL Database: Stores borrower and book data.
   2. Database Design

* Borrower Table:
* id (Primary Key)
* name
* assigned books (Comma-separated list of book IDs)
* Book Table:

id (Primary Key)

name

assigned status: Boolean.

1. DEPLOYMENT ARCHITECHTURE

* GCP VM 1: Dockerized Borrower Service and Book Service.
* GCP VM 2: Kafka instance for message brokering.
* MySQL instance on VM in GCP

1. HUMAN INTERFACE DESIGN

* User Interfaces
  + Web interfaces for borrowers and book management.
* User Workflows
  + Borrower
    - Add a new borrower.
    - View and assign books to borrowers.
  + Book
    - Add a new book.
    - View and delete the book.

---------------------------------------------------------------------------------