SYSTEM ARCHITECTURE

TECHNOLOGY STACK

For this project, following technologies have been used in the making:

Backend:

JAVA EE, Spring-core, Hibernate-core, MySQL database

Frontend:

AngularJS, HTML, CSS

BACKEND

The Library management system is implemented in Model-view-controller architecture design format. The backend has been carried out in 3 layers : –

- Data Access Layer
- Controller Layer
- Business Layer

DATA ACCESS LAYER

The Data access layer or DAO layer had two parts underneath:

Data Models:

Data models are the entity relationship mapping of database to the java objects. The ORM framework Hibernate was used to carry out the mapping. The business logic of the system is carried out by Java objects, while the database provides permanent storage for those objects. Java objects are stored in the database and retrieved later when they are needed.

Data Access Objects:

The Data access objects were used to establish connection between the database and the backend. The Data Access Object is basically an object or an interface that provides access to an underlying database or any other persistence storage. After establishing the connections, the data access layer helped in fetching the data for various entities using SQL queries and this fetched data was used in services for creating the business layer and sending it to the controller for API development.

BUSINESS LAYER

The Business layer was implemented keeping in mind considerations of the requirement of the system. It prescribes how business objects interact with one another. It also enforces the routes and the methods by which business objects are accessed and updated.

The business logic comprised of workflows that are the ordered tasks of passing documents or data from one participant (a person or a software system) to another.

This Layer had implementation in forms of services. There were services for Books, Authors, Borrowers, Fines and Loans. These services were written in Spring-core framework for coupling with the data access layer and the controller layer.

CONTROLLER LAYER

The controller layer comprised of the final point of interacting with the front end with REST APIs. The implementation of the code in form of Application interfaces was carried out in the controllers. The Controller's job is to translate incoming requests into outgoing responses. In order to do this, the controller takes request data and pass it into the Service layer. The service layer then returns data that the Controller injects into a View for rendering. This view is HTML for a standard web request; and JSON (JavaScript Object Notation) for a RESTful API request.

ASSUMPTIONS

- View layer of the MVC will have some display logic. All the business logic will be implemented in the business layer.
- Schema given the requirements is strictly followed and no real-world bias is involved.
- Default conventions are followed for folder/package and file naming.
- Assumptions on normalization of the data has been made.
- Entries with invalid and empty data are discarded and not considered in the database.