**System Architecture:**

This project is an implementation of a Library Management system. It involves the creation of a database host application that interfaces with a backend MYSQL database.

**Database name**: Library.

**Tables:**

* book(Isbn Varchar(50) primary key, Title Varchar(200), Availability Varchar(10)
* authors(author\_id int(100) primary key,name varchar(200))
* book\_authors(author\_id int(100),Isbn varchar((50), FOREIGN KEY (Isbn) REFERENCES book(Isbn),FOREIGN KEY (author\_id) REFERENCES authors(author\_id),PRIMARY KEY(author\_id,Isbn));
* borrower(card\_id varchar(10) primary key auto\_increment, ssn varchar(20) unique, first\_ name varchar(14), address varchar(50), last\_name varchar(16), state varchar(50),city varchar(50), phone varchar(50));
* book\_loans(loan\_id int(11) primary key auto\_increment, Isbn varchar(50), card\_id varchar(10),date\_out date, due\_date date, date\_in date, FOREIGN KEY (Isbn) REFERENCES book(Isbn), FOREIGN KEY (card\_id) REFERENCES borrower(card\_id));
* fines(loan\_id int(11) primary key, fine\_amt decimal(10,2), paid tinyint(1), FOREIGN KEY (loan\_id) REFERENCES book\_loans(loan\_id)).

**Table Data:**

* The tables - book, authors, book\_authors and borrower have data from CSV files. The data is normalized using a python script.
* Book\_loans and fines tables are populated from the front end.

**Database schema modifications:**

* Borrower name is stored as first name and last name.
* An additional column ‘available’ is added to book table. This is column will help to check if the book is available, so that user can checkout only those books that are available.
* ISBN10 is used for book ISBN. ISBN13 is ignored, as both are unique.
* The address of the borrower is stored in 3 different columns namely, address, city and state. The country is assumed to be United States.
* The SSN column in Borrower has a ‘unique’ constraint. If a borrower is already in the system, then he/she cannot be added again with a new card\_id.

**Servlets:**

**SearchForBook.java**

This servlet is used to search for books based on the search criteria. It queries details from book, book\_authors and authors tables.

**AddBorrrower.java**

This servlet is used to insert/add a new borrower to the database. The details entered by the user are inserted into the borrower table.

**CheckInBook.java**

This servlet is used to query the details of the books that are loaned, based on the search criteria, from the book\_loans and borrower tables.

**CheckInUpdate.java**

This servlet is used to update the availability value to ‘YES’ in the ‘book’ table and to update date\_in in ‘book\_loans’ table.

**CheckOutBook.java**

This servlet is used to insert the checked out book’s information in the book\_loans table. It also updates the availability value to ‘NO’ in the ‘book’ table.

**PayFine.java**

This servlet updates the paid value in fine table to ‘true/1’.

**RefreshFines.java**

This servlet updates or inserts values into the Fines table as required.

**Connection:**

**Connection.java**

The connection class is used to connect to the database.

The code for front end design is in home.jsp.

The mappings are in web.xml.

The code for validations are in library.js

The home.jsp is the main file, and code must be run on server from this file.