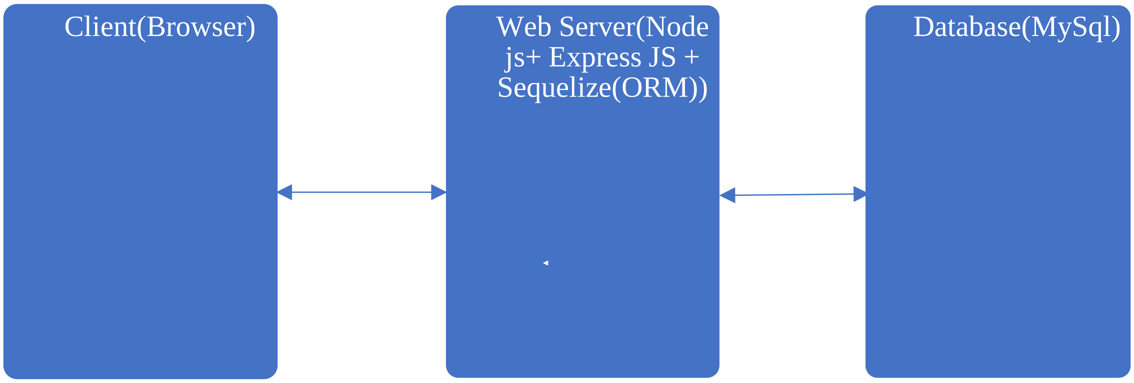
Introduction:

Library Management System is an end to end application which helps librarians find books, check-in, check-out, create new borrowers, update fines and help borrowers pay fines. The home screen houses a search bar and links to find books and check-in, create new borrowers, update fines, pending fines respectively.

Architecture:

Application uses a three-tier-architecture. A client (browser), a web server and the database.



Client is built using HTML, CSS, Embedded JS. Web-Server is built using Node JS, Express JS and utilizes Sequelize (ORM) for raw queries and the database used is MySQL.

Client uses HTTP protocol to make GET and POST requests. Sensitive client information like SSN, FullName, Address and Phone are sent to the server using a POST request.

The web pages are housed in the server and is used to render web pages to the client when necessary.

Client is used to request and insert data into in the database. The request is sent to the web-server to process the request and make the appropriate query to the database. The database sends the data as a response to the fetch query and success message for an update or insert request.

Design and Assumptions:

Initial Setup:

- When parsing book.csv, “/t” (Tab) as the delimiter.

- When parsing borrowers.csv, “,” (Comma) as the delimiter.
- All authors in book.csv file with the same name is considered the same author. For example, ‘Aldous Huxley’ appears as author for many books, but considered the same author and is assigned a unique author_id.
- Á and A are treated the same. For example, ‘Áldous Huxley’ and ‘Aldous Huxley’ are treated the same.
- Since, MySql is case insensitive by itself, the entered author name, title and ISBN is not converted to lower case while querying.
- ‘ISBN10’ attribute from book.csv is taken as the ISBN BOOK table in the DB.
- ‘ISBN13’, ‘Cover’, ‘Publisher’ and ‘Pages’ headers from book.csv are not mapped into the DB.
- ‘Email’ header from borrowers.csv is not mapped into the DB.

Check-Out:

- Books can be fetched by entering partial text of title and author name.
- Client is used to fetch the books based on Author Name, Title and ISBN and allows blank query (Click on Submit without entering anything in the textbox fetches all the books in the library).
- When there are two books with the same title but with different ISBN, they are considered different books.
- If a book is already checked-out, then the system will not allow to check the book out.
- If a borrower has already checked-out 3 books, the system will not allow to check out another book.
- Link Home is present on all the screens and used to navigate to the home page in case if the librarian wants to perform more than one task and incase of error.
- Current day is not counted in the calculation of the due date of the book. E.g.: If a book is checked out today i.e., 2nd March then the due date will be 16th March (14 days after date_out) to return the book.
- Enter the correct Card ID in the Card Number/ID field where ever the field is present.

Create New Borrower:

- SSN is stored as a string with the hyphens and while creating a new user, the librarian should enter the SSN as given in the quick starter guide.
- Phone number should be entered in the format given in the quick starter guide.
- Full Name (First Name<space>Last Name), SSN, Address (Address, City and State must be entered in the text field) fields are mandatory when creating new borrowers, while Phone is an optional field.

- Combined the ‘address’, ‘city’ and ‘state’ headers from the books.csv to store as a single ‘Address’ attribute in the DB.
- Combined the ‘First_Name’ and ‘Last_Name’ headers from borrowers.csv to store as a single attribute ‘Bname’ in the DB.

Check-In:

- To check-in the book(s) borrowed, enter Card ID/Name/ISBN and the name of the borrower and name can be partial text.
- Enter the correct Card ID in the Card Number/ID field where ever the field is present.
- The text field supports blank query, i.e., if submit is clicked without entering anything in the text field, then all records pertaining to the books that have been borrowed are displayed.

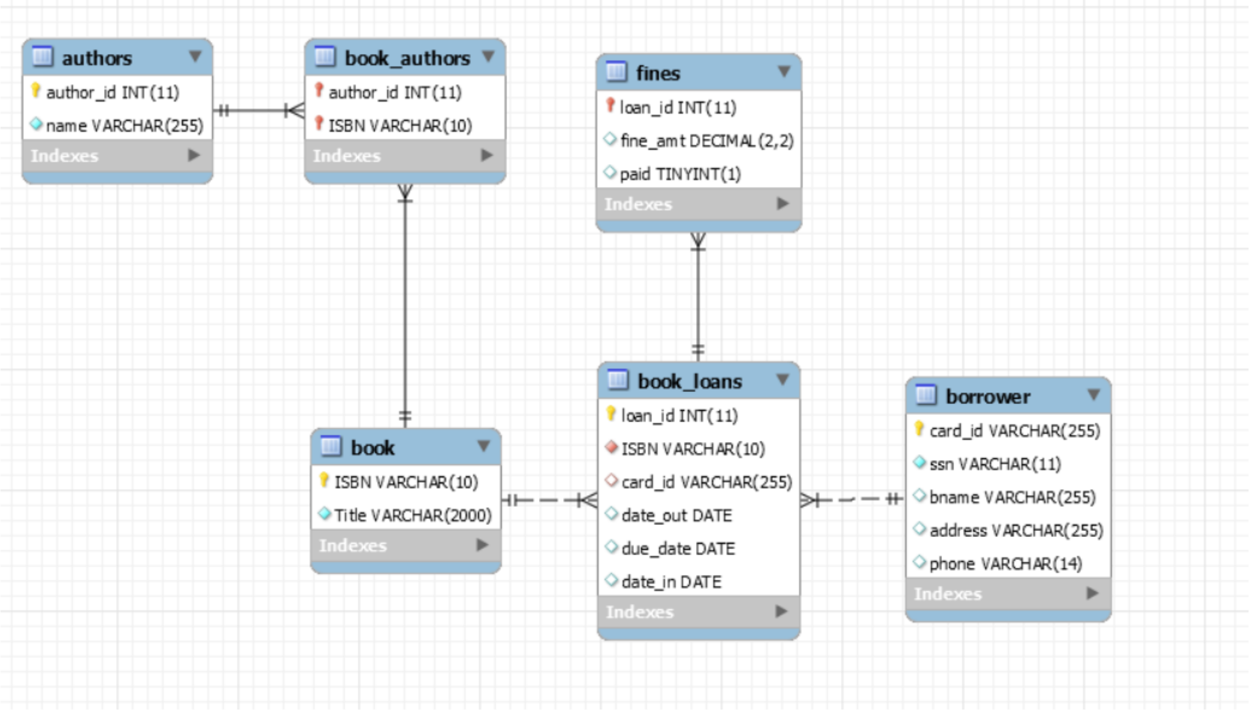
Pay Fines:

- To pay fines enter Card ID of the borrower and it should be correct to navigate to the display fines screen.
- The text field supports blank query, i.e., if submit is clicked without entering anything in the text field, then all records pertaining to the borrowers that must pay fines will be displayed along with the fine amount.
- Only the payable and liable fines are displayed, and the already payed history is filtered by default.

Update Fines:

- Librarian must click on the update fines everyday to update the fines.

Entity Diagram:


authors
author_id INT(11)
name VARCHAR(255)
Indexes
book_authors
author_id INT(11)
ISBN VARCHAR(10)
Indexes
fines
loan_id INT(11)
fine_amt DECIMAL(2,2)
paid TINYINT(1)
Indexes
book
ISBN VARCHAR(10)
Title VARCHAR(2000)
Indexes
book_loans
loan_id INT(11)
ISBN VARCHAR(10)
card_id VARCHAR(255)
date_out DATE
due_date DATE
date_in DATE
Indexes
borrower
card_id VARCHAR(255)
ssn VARCHAR(11)
bname VARCHAR(255)
address VARCHAR(255)
phone VARCHAR(14)
Indexes