

Design

System Architecture: The system consists of 3 servers: the database, the backend, and the frontend. The database server is using MySQL because all the team members were familiar. The backend is only used to pass SQL queries from the frontend to the database, and it is written in Node.js with Express for ease of use. The frontend is written using Node.js and React for good looks and easy http requests.

We also used the backend to parse the csv files and load them into the database, and a separate SQL script to create the tables.

GUI

The GUI was done using React. The main challenge with this task is taking the SQL queries from the other parts of the project and fitting them all together in one application.

To ease this, the site is split into three main components: books, borrowers, and fines. The books component contains all the functionality for book search and checking out books. The borrowers component contains functionality to create a new borrower and to search borrowers to check books in. The fines section contains all the functionality relating to fines.

Fines: fine_amt and paid attributes are as described in the project instructions. Fines are assessed at a rate of \$0.25/day (twenty-five cents per day). There are two scenarios for late books, one where late books that have been returned and one where late books are still out. There are mechanisms for all the desired mechanics with no extra components.

Checking Out Books and Checking In Books

The check out system was created according to the specifications in the instructions. Whenever a book is found through the GUI search the user can select to check out that book. When they do so they must enter their card id. The system then checks if the user already has 3 books checked out, as the maximum amount of allowed check outs is 3. The system will also check if the book is already checked out. If the book is already checked out or the user already has 3 books checked out, a relevant error message will be displayed. If this does not happen, then a tuple of BOOK_LOANS is created with a unique loan_id as a primary key. The date that the book is checked out is then set to the current date and the due date is set to be 14 days from then.

As for the check in system, this was also designed according to the specifications in the project instructions. When the user is able to locate their book_loan tuples by entering any combination of the ISBN, their card number, and any substring of their name. The GUI would then pull up a list of relevant information. They will then be able to select one of these items to check in.

Searching for Books

To search for a book, enter your search query into the search bar. The query can contain any combination of ISBN, title, and author. The results will return any matches to any of these fields. For instance, searching for 'Frank' will return results containing 'Frank' in the author and title fields.

This is achieved by searching each of the above fields of each record for any potential matches. For the sake of completeness, the entire search query is applied to each field for a substring match. This may seem somewhat redundant, but it ensures that all potential matches are returned in the results.

Borrower Management

The system should be able to add new borrowers when such an operation is initiated. A new borrower account requires inputs of a full name, SSN, and address. All values must not be null. The system will also generate a new card number automatically for each new added borrower. If a borrower is already in the system, the system will reject the add operation and prompt an error message.

Schema: We did not stray from the base schema, displayed below:

BOOK

<u>Isbn</u>	Title
-------------	-------

BOOK_AUTHORS

<u>Author_id</u>	<u>Isbn</u>
------------------	-------------

AUTHORS

<u>Author_id</u>	Name
------------------	------

BORROWER

<u>Card_id</u>	Ssn	Bname	Address	Phone
----------------	-----	-------	---------	-------

BOOK_LOANS

<u>Loan_id</u>	Isbn	Card_id	Date_out	Due_date	Date_in
----------------	------	---------	----------	----------	---------

FINES

<u>Loan_id</u>	Fine_amt	Paid
----------------	----------	------

