Design Document

UI Design:

* The UI is done using Java Swings.
* The pages are designed as JFrames.
* Each frame is designed with necessary TextFields to get inputs from the user and with buttons to perform the desired function.
* Book ISBN, title and author are designed as TextFields Search books frame. On entering any of the values, the result set is displayed in the table. Any combination of the three values are accepted. Substring match is provided which allows partial values of title and author name as inputs from the user.
* In the Check-Out frame, the borrower is checked against outstanding fines. If any outstanding fines exist, the borrower is forbidden from checking out any books until all the borrowed books are returned and the fines are paid.
* In the Check-In frame, the loan is checked for fines by comparing the due date and the current date and the borrower is permitted to return the book only when the fine amount, if any, is paid.
* In the Borrowers frame, the phone number field is split into three sub-fields representing the area code, telephone exchange number and subscriber number. The number of digits in each field is limited to 3,3 and 4 respectively. Any wrong number of digits in a particular field is reported as error. This reduces the overhead of eliminating different format of phone numbers and hence processing each of them differently to store in a unique format (XXX) XXX-XXXX.
* In the Borrowers frame, the SSN field is also divided into three fields – area code, group number and serial number with the restriction of accepting 3,2 and 4 digits respectively.
* All forms are validated as and when required.
* In the fines form, two tables are shown. Upon clicking the view Outstanding fines button, one table is populated with outstanding fines corresponding to each loan while the other table is populated with outstanding fines corresponding to each borrower.
* In the fines form, two tables are shown. Upon clicking the view Paid fines button, one table is populated with fines paid corresponding to each loan while the other table is populated with fines paid so far by each borrower.

Additional features:

* The search books frame has a check-out button. The table is populated with search results. Selecting a particular record from the table and clicking on Check Out button prompts the user to enter the card number of the borrower. Entering the card number, the borrower is checked against any outstanding fines. If no outstanding fines exist, the borrower is granted the particular loan if he hasn’t borrowed more than two books at that time.
* The fines window is designed with a check-in frame. Selecting a particular loan from the loan fines table and clicking on check-in prompts the librarian to receive fines and check-in the particular book.

Back-end:

The back-end is created using MySQL.

The loan\_id in the book\_loans table is designed to increment automatically when new records are inserted thereby reducing the overhead of coming up with new id every time upon new entry.

The borrower’s card no is calculated by adding 1 to the number of entries in the table and concatenating it with ID for each new borrower added.

The data is filled into the corresponding tables by using ‘LOAD DATA INFILE’ command.

To populate the Book\_authors and authors table , the books.csv file was tokenized using java to ‘authors.txt’ to include as each record, a unique author\_id, fullname, First name ,middle name, last name and isbn associated with the author.

Connectivity:

The connectivity is established with the help of JDBC.