**Library Management System – Design Document**

1. **Introduction**

The purpose of this document is to describe the implementation and design of the Library Management System. Library Management System is a database host application created for librarians (users of the system) who can search a book, check in and check out books for book borrowers, enter and update fines for the borrowers and also add new borrowers to the system.

1. **Technologies Used:**

The programming languages used for this project are:

Backend: PHP

Frontend: HTML, CSS, JavaScript

Database: MySQL (with MySQL Workbench)

1. **Design Overview:**

* *Database Architecture*

The library database has 6 tables as follows:

* authors (This contains the names of authors and their author ids)
* book (This contains the ISBN and the title of the book)
* book\_author (This contains the author ids and the corresponding ISBN of the book they have written)
* book\_loans (This contains the loan id, ISBN, card/borrower id, date\_out, due date and date\_in for a book that is issued)
* borrower (This contains the card/borrower id, ssn, borrowername, address and phone number)
* fines (This contains the loan id, fine amount and a binary value for paid/unpaid)
* *Software Functionalities:*

Library Management System provides five basic functionalities for the user as follows:

* *Search a book by ISBN, title, and author:* When a keyword is entered in the search field, the system queries the natural join of the books table, book authors and the authors table to look for the keyword. The search is based on substring matching.
* *Checking out:* Books can be checked out by the system user. On checking out a book, the card/borrower id is entered. To check if the borrower has not issued more than 3 books, the book\_loans table is queried using the card/borrower id. If the condition is satisfied, a new tuple is added to the book\_loans table along with the date\_in, date\_out and due\_date (which is 14 days from the date of issue). Else, an error message is displayed for exceeding the limit of borrowing 3 books.
* *Checking in:* Books can be returned and added back to the library. On checking in a book, the card/borrower id is entered and using a natural join for the book\_loans and borrower table, all the books borrowed are displayed.

The book that has to be returned is selected and the book\_loans table is updated for the corresponding isbn and card/borrower id.

* *Create new borrowers:* The user can add new borrowers to the system. The name, ssn, phone number and and address is entered. These details are inserted to the borrower table and an auto incremented card/borrower id is generated. (This is done by selecting the maximum value of card id in the borrower table and adding 1 to it), else an error message is displayed that says that the borrower already exists.
* *Fines:* This functionality includes two main parts – update fines, get fines and pay fines.  
  + - *Update fines :-* When a borrower exceeds the day-limit of 14 days to return a book, every extra day account for a fine of $0.25. If the date entered to return the book is post the return date, then the fines are calculated ((difference between the due date and the date in) \* $0.25)) and updated in the fines table for that card id.
    - *Get fines :-* To know if the borrower has any fines to be paid, the card id is entered and the fines for that borrower are displayed. For this, firstly, the paid attribute value from the fines table is obtained for that particular card id. If there are fines to be paid, then the sum of all the fines is displayed. (A natural join between the book\_loans table and the fines table gives all the books that the borrower has issued and the corresponding fines.)
    - *Pay fines :-* When a borrower wishes to pay the fine, the Boolean value for the paid attribute is set to True and the fine amount is updated simultaneously.