

LIBRARY MANAGEMENT SYSTEM

BY

VAMSEEKRISHNA KATTIKA

VXK165930@UTDALLAS.EDU

DESIGN

This project (Library Management System) is a general-purpose project, which has all the basic needs of a typical library system. The project involves the creation of a database host application that interfaces with a SQL database that implements a Library Management System. Users of the system, are understood to be librarians.

The application GUI is programmed with HTML and PHP. MySQL Database is used. As per the functional requirements, there are five modules designed. They are,

- 1) GUI
- 2) Book Search and Availability
- 3) Book Loans
 - a) Checking out books
 - b) Checking in books
- 4) Borrower Management
- 5) Manage Fines

1) The GUI is designed using HTML and PHP. All interface with the Library database (queries, updates, deletes, etc.) are done from a graphical user interface of the original design. The GUI application is interfaced with the Library database via an appropriate SQL connector. Initial database creation and population are done from phpMyAdmin tool.

2) Using the GUI, able to search for a book, given any combination of ISBN, title, and/ or Author(s). The search interface is provided a single text search field (like Google) and is case insensitive. The query supports substring matching. It displays ISBN, Book title, Book author(s) (displayed as a comma separated list), Book availability (is the book currently checked out?).

3) Once found in the GUI search, able to check out a book after prompting for Borrower Card No. The due date is 14 days after the date out. If a borrower already has 3 books checked out, the system will show an error message and the checkout is aborted. If the requested book is already

checked out, then the checkout fails and returns a useful error message. If the borrower is not present, then check out fails and returns a message that the borrower doesn't exist.

Using the GUI, able to check in a book. Able to locate using book id, card no, and/or any substring of BORROWER name. Once located, a way of selecting one of potentially multiple results is provided by a radio button to check in (i.e. today as the date in in corresponding BOOK_LOANS tuple).

4) Using the GUI, able to create new borrowers in the system. All name, email, address, city, state and phone attributes are required to create a new account (i.e. value must be not null). The system automatically generates new card no primary keys for each new tuple that uses a compatible format with the existing borrower IDs. Borrowers are allowed to possess exactly one library card. If a new borrower attempts with the same email and phone, then the system rejects and returns a useful error message.

5) Fines are assessed at a rate of \$0.25 a day. A button is provided that updates/refreshes entries in the FINES table. For late books that have been returned, the fine will be the difference between due date and date out. For late books that are still out, the estimated fine will be the difference between due date and today. If a row already exists in FINES for a particular late BOOK_LOANS record, then, if paid =0, the system updates the fine amount if different than current value. If paid =1, nothing will be modified.

A mechanism for librarians to enter payment of fines is provided. The payment of a fine for books that are not yet returned is not allowed. The system only updates the fine amount as paid for those books which are returned. Display of Fines is grouped by card no. i.e. SUM of the fine amount for each Borrower. Display of Fines provides a mechanism to filter out previously paid fines by providing two buttons each for previous paid fines and to pay current fines.

The schema has not been changed except for borrower. For borrower, first name and last name columns are added in the place of name. SSN is removed from the table. Email, City and State are added to the Borrower table.

User Instructions for librarians:

1) A web interface using HTML has been provided for a user-friendly experience.

2) The user need to open any browser and type localhost/LibraryManagement.html.

Note: All the files are present in the default directory C:\MAMP\htdocs

3) Now the web interface is pretty much self-explanatory, and the user can perform the various desired actions.

4) The Web Interface consists of Search, Check-out books, Check-in books, Borrower Management, and Manage Fines. The user can click the appropriate links and perform the desired operation.

For more detailed information, please refer to the User Guide provided.

Design decisions:

The baseline data to initialize the database which is present in CSV files, is used and normalized onto the schema and the tables.

- 1) In the web interface, I have used HTML. The design consists of main page which contains 5 text hyperlinks.
- 2) The five hyperlinks contain the Search, Check-out books, Check-in books, Borrower Management, and Manage Fines.
- 3) If an error occurs because of the user's false input it is displayed on the page.
- 4) In the Database, the table book has a book_id which has been made the primary key.
- 5) In the book_authors table, the author_id and isbn form the primary key together.
- 6) In the authors table, author_id is the primary key.
- 7) The book_loans table has loan_id as its primary key. This is auto incremented so that each time a book is checked out from the library a new loan_id is generated.
- 8) The table borrower has card_no as its primary key. No borrower can have two cards. Also, the card_no is auto incremented so that when we add a new borrower we give him a unique card_no everytime. No two borrowers can have same email and phone number.
- 9) The fines table has loan_id as its primary key.
- 10) We can search for a result by giving substring as an input.
- 11) In the Web Page Update fines and Payments have been created. The Update fines updates the fines table.
- 12) The Payments page has option to show paid fines. It also has a facility to show current fines to be paid and option to type the card number and pay the fine.

Technical Dependencies:

- 1) Front end -HTML and PHP
- 2) Database - MySQL: 5.6.34
- 2) Other specifications:
 - Apache: 2.2.31
 - PHP: 7.1.5 & 7.0.19 (for PHP 7: Windows 7 minimum with SP1 and Windows Vista minimum with SP2)
 - phpMyAdmin: 4.4.15.5