

Database Design 6360.501

Library Management System

-Sxd173730

-Sashidhar Donthiri

System Architecture:

Backend Language – Python (Interpreter- 3.6.3)

Framework : Flask for Python(Microframework for Python based on Werkzeug, Jinja 2 and good intentions).

UI: bootstrap pre-defined classes.(Bootstrap cdn)

Database: MySQL database(Workbench).

Packages used: Flask, flaskext.mysql import MySQL, jsonify,request/response

The server system must have flask and python(3.6.3) installed in his PC.

The API should contain credentials for the MySQL Database for the database to be accessed.

Design Decisions and Assumptions:

- Borrower name (Bname) from Borrowers table in Schema has been split to Fname and Lname.
- ISBN 10 has been used as a unique key for the books instead of ISBN13.
- SSN for any new user has to be entered manually into a form in ManageBorrowers page.

- Address-NOT NULL
- City-NOT NULL
- State- NOT NULL

- Each cardID is unique and has been autoincremented in Schema starting from 1.
- Each loanID is unique and has been autoincremented in Schema starting from 1.
- Each AuthorID is unique and has been autoincremented in Schema starting from 1.

- In Checked-in page: There could be multiple results in results table with same ISBN (Case where an author name/title might have the same string for search).
 - In such case only 1 book can be checked-in.
- Display Fines- Displays the fines that aren't yet payed grouped by cardID.
- Display Paid Fines- Displays the fines that are payed and fines.paid='1'
- Pay Fines- A search bar is associated with the Pay fines button where a loanID can be searched and fine amount can be viewed which can be toggled and payed on pressing Pay button.
- Paid in Fines table is chosen as a Boolean value(Either 1 or 0)
 - If paid, p_value=1 else p_value=0;(A book paid bool value is changed only if the book is checked-in)