**DESIGN DOCUMENT**

The Library management system is created in PHP and is hosted on local host using Apache server. To dynamically interact with user and maintain the inventory of books I have use MySQL server for database creation and information retrieval.

**Table creation and Importing data in MYSQL**

* Datbase Name: Library\_BookHub,
* Schema as per diagram given .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Fields** | **Primary Key** | **Foreign Key** |  |  |
| Book | ISBN13(char),Title VARCHAR | ISBN13 |  |  |  |
| Book\_authors: | author\_id(int,),ISBN13 int | Author\_id | ISBN13(varchar10,primary key) refers to book(ISBN13), author\_id refers to Authors(Author\_id), |  |  |
| Authors | Author\_id(autoincement),author(varchar200). | Author\_id |  |  |  |
| Borrowers | Loan\_id INT,card\_id(int) , ISBN13(char) , (date):dateOut, dueDate, dateIn,  bname(varchar), address(varchar), email (unique), phone(varchar). | CARD\_ID | ISBN13(char) refers to book(ISBN13),  card\_id(int)refers to borrowers(card\_id) |  |  |
| BOOK\_LOANS |  |  |  |  |  |
| Fines | loan\_Id(int),fine\_amount(decimal(8,2)), paid(boolean). | Loan id | Loan ID refers to book\_loans(loan\_id) |  |  |

**Fines:** Fields- loan\_Id(int,primary key)refers to book\_loans(loan\_id) ,fine\_amount(decimal(8,2)), paid(boolean).

Normalization of data consisted of:

* + - One of the uniquely identifying ISBN (ISBN13)was chosen and used as the primary key for consequential tables instead of the both the ISBN10 and ISBN13.
    - Multiple attributes were combined into a single attribute through concatenation in the case of bname(first name, last name) and address(address, city, state).
    - The book table was populated with another attribute, book\_availability, a boolean value that states whether the book exists in the database to check\_out.
* Connection to the database:

In order to connect to a MYSQL database a library is necessary, which I downloaded from https://dev.mysql.com/downloads/connector/j/5.1.html

The Connect to the database using java’s Connection object.

LibClass.java consists of the GUI implementation as well as the queries that fetch and display the data from MYSQL server. The GUI consists of four TABS, each of them implementing a single functionality of the library system**: SEARCH, CHECK\_IN, CHECK\_OUT, BORROWERS AND FINES.**

* **SEARCH TAB:** Consists of an absolute layout with the following components:
  + - A **textField** to search the database through any combination of ISBN13, title or the author name was coupled to the **search button** with an action event listener. Upon called, the data fetched is displayed in the **table view** provided.
* **CHECK\_OUT TAB:** Consists of an absolute layout with the following components:
* Consists of two **textfields**, ISBN number and card number which together are used to check out the book when **check out** button is used.
* **CHECK\_IN TAB:** Consists of an absolute layout with the following components:
* This panel consists of three **textfields** in total along with two **buttons** and a **table** **view**.
* Value could be entered in either of the textfields **(ISBN/card number/name**) which is used to display the books that are borrowed upon clicking **check** **for** **loans** button.
* When book names which are borrowed are fetched and displayed in the table view, we select the row and it is checked in when we use **check** **in** button.
* **ADD\_BORROWER TAB:** Consists of an absolute layout with the following components:
* This panel consists of six **textfields** which are used to take values of **first\_name**, **last\_name**, **SSN**, **Email\_address**, **Phone\_number** and **Address** of the borrowers.
* The two **buttons**, **add** **borrower** which is used to add the details of the borrower in the database and the other button **reset** **details** which is used to clear the values entered in the textfields.
* Validations are provided for the phone\_number format as well as to rule out redundancy as duplicate email addresses are not allowed.
* **FINES TAB**: Consists of an absolute layout with the following components:
* There is a **textfield** for **card** **number** which is used to display the fines of books the borrower had took upon clicking **get** **data** **button**.
* The fines are displayed in the **table** **view** we used in the panel.
* The **radio** **button**, **show** **previously paid** is used to display the records of the borrower where he already had paid the fine.
* **Pay fine button** is used to pay fine of the books selected.
* **Total fine button** is used to display the total fine amount of the borrower.
* **Refresh entries button** is used to update the fines of the books that borrower had borrowed.
* Validations are provided as to reject the option of paying fine without checking in the book as well as with the option of paying fines separately for a given card\_id.