**CS 631**

**Data Management System Design**

**Project: Library Management System**

**Deliverable: 3**

Tahiatul Islam

Sohrab Namazi Nia

Mohammad Dindoost

**Introduction:** CS 631 has taught us various aspects of the database system. We also worked on a practical project using SQL to manage a library system. This report summarizes details of our projects in detail and gives a user manual who are interested in using the system.

**Background:** This project is a result of the continuous effort of designing a database starting from analyzing requirements to creating EER and then designing a schema. In this step we have developed an application using query language and used a host app to interact with it.

**Technical description**: The project is a web-based solution. We have used various technologies to develop the project. Each technology is used for different purposes. In the following, we are mentioning them along with their purposes.

1. PHP is used on the server side. Apache is used to host the application
2. HTML and CSS are used for front-end design along with JavaScript for responsive user interaction
3. MySql is used for database

**User Manual:** There are two options to access the system. One as an **admin** and another as a **reader**. In the following, the main features and key differences between them are presented.

*Admin Guide*:

On the login page of the system, admin login required an admin mobile no and password corresponding to that account. The login page view is attached in **Fig:1** in the later section of this report. Upon getting access to the admin dashboard, the following features are provided to the admin.

1. The admin can **add a new reader** to the library system. This option is provided in the left menu of the admin dashboard. The admin can initially provide a password for the reader which the reader can later change for convenience. The view of the application to add a reader is shown in Fig:2
2. The admin can also **add new branch information** to the library system. Information about the already existing branch can also be discovered from the admin panel.
3. Admin can **add a person** to the library system. The person can act as an author, editor, or guest editor for documents of the library.
4. Admin can **create documents** and add a copy of that document in the system. The document menu on the left of the admin dashboard displays the functionality/form to create that item. There are different types of documents namely Books, journals, and conference proceedings. There are different menus to add each of them separately.
5. Finally, **there is an analytics section** in the admin panel to get useful information from the library system. In the reports section of the dashboard, the admin can see analytics information about the frequent borrower, most borrowed book, book of the year, and average fine by a library branch.

*Reader guide:*

1. Go to CityLibrary/Library/CityLib/user-login.php URL. Log in to your account. The admin of the system has to create an account on your behalf of you. The user has no right to create an account on the system on their own.
2. In the dashboard, the user can see the list of documents with the title and publisher name.
3. There are two buttons against each document. One is “borrow” and another is “reserve”. You can select one of these buttons and add that particular document to your cart.
4. After adding documents to the cart, you can click on the “Cart” icon provided in the top right corner of the menu. Clicking on the cart icon would allow you to see the list of documents borrowed and reserved by you.
5. You can remove particular documents from your cart list by clicking the “remove” button.
6. To confirm your document order, you have to click the “checkout button” provided on the bottom right corner of the page.
7. In the dashboard, there are three options available for you: search by title, search by publisher, and search by document id. You can filter your documents by either one of these criteria or two of these criteria or even by all three of these criteria.
8. There is a submenu on the left side of the page. It is “My borrow & Reserve List”. You can see the list of documents borrowed or reserved by you separately.
9. You can click on the “Profile” submenu on the left side menu. You can see the total fine imposed on you.
10. You can also delete your account by clicking the “Delete” button in the left-side submenu

**SQL Commands:** SQL is used in the application to create tables for the database and also for the execution of different command. Since adding the all the SQL command will unnecessarily increase the report length, I am adding the SQL command to as a separate file with this document below. The file can be opened by clicking on them. DB\_TABLE\_CREATION.sql file include command to create the tables.



The sample sql queries for different purposes are added below. The parameters are passed from front end side of the application.

|  |  |
| --- | --- |
| **Objectives** | **SQL queries** |
| Add publisher | INSERT INTO PUBLISHER (PUBNAME, ADDRESS) VALUES('PubName1', 'PubAddress1’); |
| Insert Reader | INSERT INTO READER(RTYPE, RNAME, RADDRESS, PHONE\_NO) VALUES(1,'RNAME1', 'RADDRESS1', 'PHONE1'); |
| Insert Branch | INSERT INTO BRANCH(LNAME, LOCATION) VALUES('LNAME1', 'LOCATION1'); |
| Insert Book | INSERT INTO DOCUMENT(TITLE, PDATE, PUBLISHERID) VALUES('Book1','2021-04-01',1); INSERT INTO BOOK(DOCID, ISBN) VALUES(1, 'ISBN1'); |
| Insert Doc Copy | INSERT INTO COPY(DOCID, COPYNO, BID, POSITION) VALUES(1, 1, 1, 'ABCDEF'); |
| Insert Borrow | INSERT INTO BORROWING(BDTIME) VALUES(NOW()); INSERT INTO BORROWS(BOR\_NO, DOCID, COPYNO, BID, RID) VALUES(1, 1, 1, 1, 1); |
| Insert Reserve | INSERT INTO RESERVATION(DTIME) VALUES(NOW()); INSERT INTO RESERVES(RES\_NO, DOCID, COPYNO, BID, RID) VALUES(1, 1, 3, 1, 1); |
| Get Branch | SELECT \* FROM BRANCH; |
| Get Reader ID | SELECT RID FROM USER WHERE USER\_MOBILE\_NO='123456' AND PASSWORD='123456'; |
| Get available docs for reader | SELECT \* FROM (((DOCUMENT NATURAL JOIN PUBLISHER) NATURAL JOIN COPY) NATURAL JOIN BRANCH) WHERE (DOCID, COPYNO, BID) NOT IN (SELECT DOCID, COPYNO, BID FROM BORROWS NATURAL JOIN BORROWING WHERE RDTIME IS NULL UNION SELECT DOCID, COPYNO, BID FROM RESERVES); |
| Get Borrow List | SELECT DISTINCT BOR\_NO, BDTIME FROM BORROWING NATURAL JOIN BORROWS WHERE RDTIME IS NULL AND RID= 1 ORDER BY BDTIME DESC; |
| Get borrowed document list | SELECT TITLE, COPYNO, PUBNAME, LNAME FROM (((BORROWS NATURAL JOIN DOCUMENT) NATURAL JOIN PUBLISHER) NATURAL JOIN BRANCH WHERE BOR\_NO=1; |
| Return Docs | UPDATE BORROWING SET RDTIME=NOW() WHERE BOR\_NO=1; |
| Calculate Fine | SELECT SUM((DATEDIFF(NOW(), BDTIME) - 20) \* 20) AS FINE FROM BORROWS NATURAL JOIN BORROWING WHERE RID = 1 AND RDTIME IS NULL AND DATEDIFF(NOW(), BDTIME) > 20; |
| Clear reservation | DELETE FROM RESERVATION WHERE RES\_NO=1; |
| Most frequent borrowers | SELECT RID, RNAME, COUNT(\*) AS NUM\_DOCS FROM BORROWS NATURAL JOIN READER WHERE BID = 1 GROUP BY RID, RNAME ORDER BY NUM\_DOCS DESC LIMIT 10 |
| Most borrowed Documents | SELECT DOCID, TITLE, PDATE, PUBNAME, COUNT(\*) AS NUM\_DOCS FROM ((BORROWS NATURAL JOIN DOCUMENT) NATURAL JOIN PUBLISHER) WHERE BID = 1 GROUP BY DOCID ORDER BY NUM\_DOCS DESC LIMIT 10; |
| Most popular 10 documents in a year | SELECT DOCID, TITLE, PDATE, PUBNAME, COUNT(\*) AS NUM\_DOCS FROM ((BORROWS NATURAL JOIN BORROWING) NATURAL JOIN DOCUMENT) NATURAL JOIN PUBLISHER WHERE YEAR(BDTIME) = 2021 GROUP BY DOCID ORDER BY NUM\_DOCS DESC LIMIT 10 |
| Calculate Average Fine | SELECT BID, LNAME, SUM((DATEDIFF(RDTIME, BDTIME) - 20) \* 20) / COUNT(DISTINCT(RID)) AS AVG\_FINE FROM (BORROWS NATURAL JOIN BORROWING) NATURAL JOIN BRANCH WHERE RDTIME IS NOT NULL AND (YEAR(BDTIME) BETWEEN '2021' AND '2021') AND DATEDIFF(RDTIME, BDTIME) > 20 GROUP BY BID, LNAME |
| Delete a reader | DELETE FROM READER WHERE RID=1; |

**Challenged Faced:** One of the most challenging task was to efficiently connect the front end to the MySQL database side. We were hesitant about choosing the best option for our front end side and at the end decided to go with the PHP as it has easy built in functions to query on database.

Apart from that we also faced trouble about formatting the date in the SQL and UI side. How date is passed from front end side was different from MySQL default format. We needed to adjust that.

Populating the DB with data was also a cumbersome job. There are many level of granularity for documents. For example a journal has volume and multiple issues associated with each journal. Populating issues and then connecting them with volume was also challenging. Some placed we ignore the journal issue and only worked with volume as a stand-alone document.

**Conclusion:** Working on a project with practical requirement definitely strengthen our grasp on SQL. The project was interesting and challenging at the same time which in turn made us realize many concept of SQL query effectively.

**Different UI of Application:** In this section we will attach pictures of the different pages from our projects to get familiar with it.

Graphical user interface, application

Description automatically generated

Figure Login Page

Graphical user interface, text, application, email

Description automatically generated

Figure : Add reader

Graphical user interface, application

Description automatically generated

Figure : Admin dashboard

Graphical user interface, application

Description automatically generated

Figure : Reader dashboard

Graphical user interface, text, application, email

Description automatically generated

Figure : Sample Form to get data from User

Graphical user interface, text, application, email

Description automatically generated

Figure : Analytics section of admin panel