CS 6360 Database Design

Library System Programming Project

Individual Project - Milestone 2

Professor Chris Davis

Date: October 24th, 2022

By:

Jevani Nikitha Chittaluri(jxc210050)

1. Overview

This project involves the creation of a database host application that interfaces with a backend SQL database implementing a Library Management System. Users of the system are understood to be librarians.

2. Programming Language

The programming language that we will use to implement this project is **Java**. SQL database we are using is MySQL.

Frontend: JFrame
Database: MySQL
Programming Language: Java.
MySQL Platform: MySQL Workbench
Java IDE: Visual Studio Code

Libraries Used: mysql-connector.java and rs2xml.jar

3. Architecture and Design

Planned Architecture for this project is as follows,

Our database here is stored in MySQL. Then from our java program, using JDBC connection we will connect to MySQL.

Using MySQL Workbench, we are creating 4 tables Book, Borrower, Book_loans and Fine. Then, from App.java file we are creating a JDBC Connection. Then adding data from Books.csv and Borrower.csv files into our Tables.

Then we are adding JForms for all the pages. On Home Page we are giving users 4 buttons. One for searching books and checking out the desired book. Next a Check-In button to check-in any checked out books. When a book is checked in, we are making that book again available for users to check out if needed.

For Fines to get updated from time to time, we are running fines updating logics every time the home page is loaded.

We are also providing a add borrower button, on clicking on which a librarian can add a Borrower.

We are connecting all the JForms from MainFrame.java.

3. Assumptions:

1. We are assuming that the quantity of all the books is just 1. So, when loading books.csv file into database we are making default of is_available field to be true.
2. We are also assuming that once the user check outs a book, he would like to get redirected to main page. So, after check-out success we are automatically redirecting user to Main Frame.
3. Fine Amount would get updated every time home page load and when Fine button is clicked.
4. A new Borrower cannot have same SSN as the ones already in borrower table.
5. On Clicking Fine Paid button for a particular Borrower, we will consider that fines of all the loans for returned books are paid off. But, the fines of books not returned as considered no paid.

4. Tables:

Tables created as part of this project are as below:

a) Books
b) Borrower
c) Book_Loans
d) Fines

Tables Schema:

```
CREATE TABLE BOOK(
ISBN VARCHAR(10) NOT NULL,
TITLE VARCHAR(512) NOT NULL,
AUTHRO VARCHAR(512) NOT NULL,
COVER VARCHAR(512) NOT NULL,
PUBLISHER VARCHAR(512) NOT NULL,
PAGES INT NOT NULL,
IS_AVAILABLE BOOLEAN NOT NULL DEFAULT TRUE,
CONSTRAINT PK_BOOK_ISBN
PRIMARY KEY (ISBN));

CREATE TABLE BORROWER(
Card_NO VARCHAR(20) NOT NULL,
SSN VARCHAR(11) NOT NULL,
FIRST_NAME VARCHAR(20) NOT NULL,
LAST_NAME VARCHAR(20) NOT NULL,
EMAIL VARCHAR(255),
ADDRESS VARCHAR(255) NOT NULL,
CITY VARCHAR(200) NOT NULL,
STATE VARCHAR(200) NOT NULL,
Phone CHAR(20),
PRIMARY KEY(Card_NO));
```

```sql
CREATE TABLE BOOK_LOANS(
LOAN_ID INT auto_increment,
CARD_NO VARCHAR(20),
ISBN VARCHAR(10),
DATE_OUT DATE,
DUE_DATE DATE,
DATE_IN DATE default null,
PRIMARY KEY(LOAN_ID),
FOREIGN KEY(CARD_NO) REFERENCES BORROWER(Card_NO),
FOREIGN KEY(ISBN) REFERENCES BOOK(ISBN)
);


CREATE TABLE FINE(
LOAN_ID INT,
FINE_AMOUNT FLOAT DEFAULT 0,
FINE_PAID BOOLEAN DEFAULT FALSE,
PRIMARY KEY(LOAN_ID),
FOREIGN KEY(LOAN_ID) REFERENCES BOOK_LOANS(LOAN_ID));
```