Database Design CS 6360.501

Programming Project:
Library Management System

Instructor: Chris Irwin Davis

Student: Guang Yang (UTD ID：2021246531)

Date: 10/23/2015

Table of Content

1.Quick Start user guide for librarian system users

There are five main parts in GUI design of the librarian system: Register, Book search, Book check out, Late fees, Check in.

As you may see on the html page, the first part is for book searching, the user could search any book in our database based on book id (In this case is the 10 character ISBN numbers), title of the book, or the book's author. My system supports substring matching for all these parts, which means when the user can't remember exactly all 10 digit ISBN, he or she can use part of the ISBN to search and locate the book needed, or based on part of the book title or part of one or more author's name for searching. When the user press the submit button, the page direct the user to the book's information page, there are Book ID, Title, Author Name, the branch id of the book, number of copies in the branch, currently available copies and a "check out" button for the user to check out. When the available copy for the book is zero, the check out button will display "N/A" indicating it is not available, even when the user presses the button. The page will show message "Copy not available!". When the user found the book ned and it is available at a particular branch, then after clicking the check out button, the page will redirect to another page for the user to officially check out, in the page the user will need to type the library card number in order to check out, the all the book's and user information will be sent to the database. And there will be the "Return Homepage" for user to return to the main page.

Next part is for user to check in the book, this part is quite like the Book Search part, the system can display the information based on the user's input, it can be part of the book id ,card number or the user's name, after submitting the button, the

page will direct the user to the check in page, which display the book id and the book's check out date, due date and check in date, if the book has not been checked in , there will the check in form for the user to type. After submit the button, the page can have redirected to the homepage. Notice that there is a refresh button that updates/refreshes entries in the FINES table, which we will describe later on.

The third part for the fee payment, the system can display all the fees involved with the user's account when two case happens:

1. The book is returned after the due date, and the fees are calculated base on the days between the due date and today, multiply the rate.
2. The book has not even return after the due date, and since the fees will be increasing, the the user will get the estimate fee based on the current day's rate.

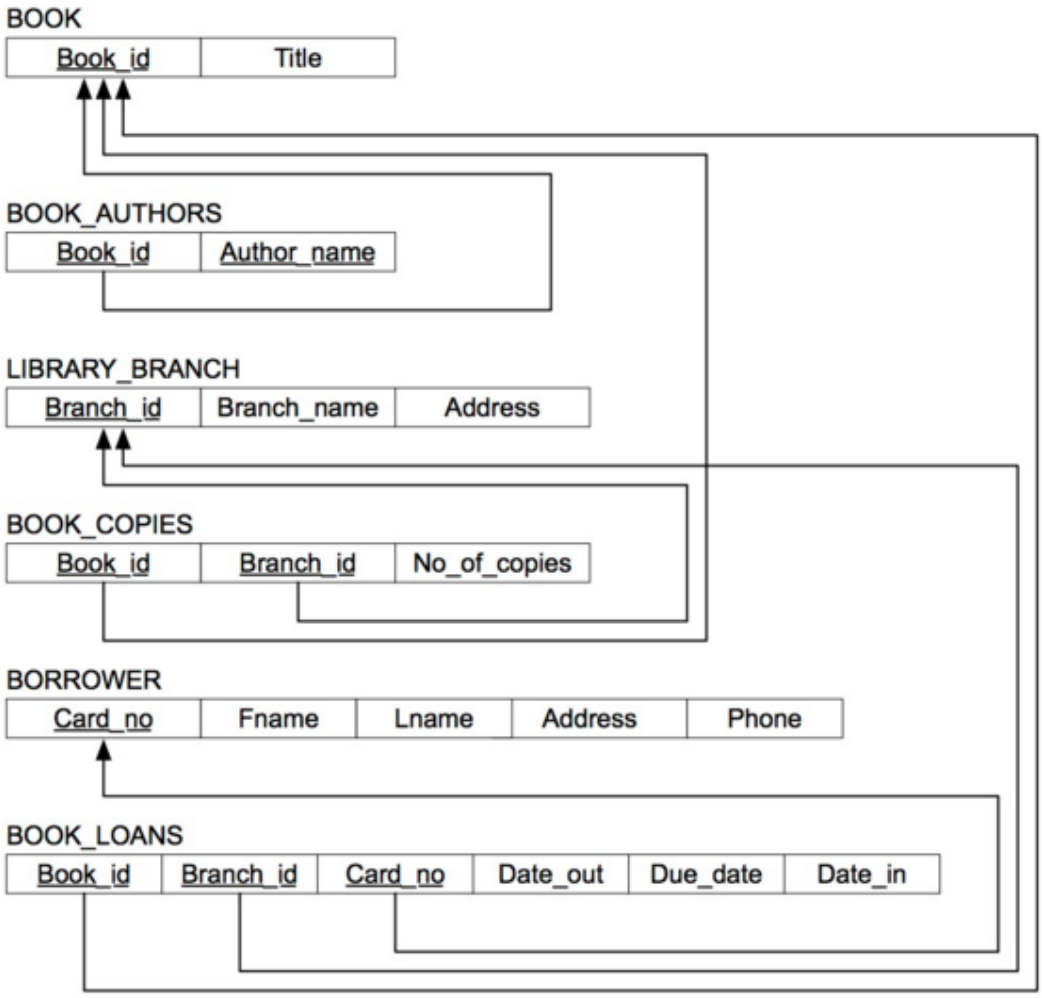
If the user has book after the due date but has not returned it, then the system will prevent the user from paying other books' fees if exists.

The last part, as can be seen on the side of the html page, it is a register form for the user to register their information if their name and address has not existed in the database, when both their full name and address are same, the system will reject and return an error message, otherwise the new browser's information will be stored in the database.

2. High-level architecture, design decisions, and justifications

Based on the project schema below, we first create the exact schema. The books.cvs file can be imported successfully into

the database at first, and then I noticed that there are a lot of noise data in the book_copies.cvs, which quite a lot of books are not existed in the Book table, since the Book_id in the BOOK_AUTHORS table is the foreign key that references the BOOK table, then I use the intermediate table used to screen out the Book_id that does not exist in the Book table. The table creation file specify the details of the creation process.



The last table is called FINES, which loan_id is the primary key as well as foreign key that references the Book_id in the BOOK_LOANS table, when creating all the tables , it is necessary consider all the attribute carefully, from the key choice , default value to all the constraint on all the tables.

After successfully created the schema and all the tables, the GUI design is created to implement all the function requirement for the project. First thing first, I design the web page using html file , which is the homepage of my library system , and is also the entry for the user to use this service. And I design the web page into several parts to fully implement all the necessary requirement as well as some extra function. I add css file for the clean and colorful html page, and using PHP to communicate from the user input to the database, and will talk about all the software in the next chapter. All the file can be reviewed below. The names are self explained . The library.php is like the base php file, it connects the library_checkin.php, library_loansRefresh.php, library_user.php, library_checkout.php, library_pay.php. In the book search section, the html page send all the info to library.php, and it process the data and choose the right query to connect to the database and execute the right statement, same as the user register, book check in, check out and fine payment section. The library_checkin.php send all the user's typing information to the library_finalCheckin.php and process the rest of the data. So does the ibrary_finalCheckout.php, it process all the data from the library_checkout.php, and the library_finalPay.php process the payment part and connect the payment to the database. I use css file to add "flavor" to both the html and all the php file.

File overview:

- library.css
- library.html
- library.php
- library_checkin.php
- library_checkout.php
- library_finalCheckin.php
- library_finalCheckout.php
- library_finalPay.php
- library_loansRefresh.php
- library_pay.php
- library_user.php

All the tables in the database schema are satisfy the NOT NULL, UNIQUE, PRIMARY KEY , FORNIGN KEY constraint to make the database system a fully and successfully design system to work for the user's daily use.

3. Technical dependencies (software libraries, software versions, etc.) and build instructions.

-- MAMP version 3.4
-- MySQL version 5.6.26
-- MySQL dump 10.13 Distrib 5.6.24, for osx10.8 (x86_64)
-- Host: localhost   Database: library
-- Server version   5.5.42
-- html, css, php editor : TextWrangle

All html, css and php file are located local in the htdocs folder in the MAMP application folder.I use the MySQL workbench for the query test and data importing and MAMP server for connecting all the file to the database, some build instruction can be reference from the second part.