**Title:** Creating a Course Reserve textbook database management system for promoting SJSU Library’s Course Reserves program

**Team Members:** Dandan Zhao, CHING-MIN HU, Qiao Liu

**Project Description**

Internet technology has changed our daily lives dramatically, such as shopping, entertainment and business behavior. While the commercial world benefits tremendously from cutting-edge technology, some institutions have not yet adopted the newest tend. One example is public library systems.

As Master students at San Jose State University, our team members have all encountered inconvenience with purchasing course required textbooks. Textbooks can be very expensive, especially the newest editions. They can be heavy and thus hard to carry around all the time. They can depreciate quickly as newer editions come out. Purchasing a hundred-dollar textbook only for occasional use always makes students debate whether it is worth it or not.

Luckily, many universities in the United States provides course required textbooks rental at their school library. At the San Jose State University Library, a special collection of course required textbooks called Course Reserves are available for SJSU students. Instructors can make a request to the SJSU library to place a university-owned or personal item on the physical Course Reserves for student uses. Students can check out these materials with their student ID.

Library staffs are very passionate about this Course Reserves program because it provides convenience to the students, making their education more affordable. And many students benefit from this program.

But the Course Reserves program also has its limitations. With research and personal experiences, we identify the following issues with the Course Reserves program:

1. The program is highly under-promoted. Many students who needed the help do not know the existence of this program.

2. The program lacks a user-friendly interface to attract its target audience: college students, a large portion of which are undergraduates.

3. The number of textbooks for each course is not proportional with actual demand, causing waste of budget and longer wait time for popular textbooks.

4. The SJSU library website only supports textbook search based on course number and instructor’s name. A general search with other information is not supported by the current system.

The goal of this project is to help the Course Reserves program to build a more attractive user interface that allows more search options and design a backend database that would support these search options as well as store more information on renting behaviors to enable further analysis and optimization.

## Approach

Due to common practice of most library database systems, our team has decided to create a relational database. Our choice of RDBMS is MySQL due to our familiarity with the system and its compatibility with other programming languages.

Our project focuses on 3 target audiences: students who will be renting from the Course Reserves, librarians who performs daily managing of the textbooks and the database administrator. The project outcome should be as follows:

1. For the student side, our goal is to build a stable and user-friendly web interface, allowing them to search for course required textbooks by ISBN number, title, major, courses and instructors.
2. For the librarian side, we want to create a database with enough information for them to track usage of books and do analysis on the program. Ultimately, our goal is to optimize the Course Reserves program to reach maximum usage and minimum expense.
3. For the administrator, he or she would be given the authorization to both managing the librarians and making major changes to the database.

|  |  |
| --- | --- |
| Implementation steps | |
| Activity | Limitation and Restrictions |
| Identify entities, relationships, connectivity and cardinality |  |
| Establish the ER diagram |  |
| Collect and prepare data | Due to university policy, the course reserves list provided by the SJSU library is historical data from the past academic year. We also needed to create some fictional data due to unavailability of the actual data. |
| Create schema in MySQL |  |
| Import data |  |
| Analyze and check completeness of the database through running a few query statements |  |
| Create a web interface |  |

**Other Restrictions and Limitations**:

Because of data privacy, the Course Reserves list we used in this project is historical data from last year, courtesy of SJSU library. We will also need to produce artificial data of student records and renting history, for those data are not available.

Due to the choice of using a relational database, we will process check out and check in data through regular DML. This does not go against the convention of real-world practices, though new ways of processing it as streaming data may be more efficient. But the choice of a relational database can serve our purpose well enough.

Because we’ve decided to use ISBN as the primary key to our main book table, this database can only store textbooks with an ISBN number. Other formats such as manuscripts and videos are not supported by our database.

**Implementation Details**:

1. We first established our business rules as follows:

1) Course Reserves are not restricted by majors. The Course Reserves program opens to both undergraduate and graduate students. Some undergraduate students do not have a declared major.

2) A course can have 0 course reserves, but a course reserve textbook must correspond to at least 1 course.

3) Different classes of the same course may require different textbooks based on instructor’s preference.

4) Multiple classes can have the same textbook.

5) One class can have multiple textbooks.

6) One student can borrow multiple books.

7) The database supervisor will be supervising all librarian’s uses on the database. He or she has root access to the database.

8) Librarians may add items to the database but cannot delete anything.

9) Only SJSU student with a valid student ID can check out materials from the Course Reserves Stack.

2. We identified the entities needed in the database as the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| ENTITY | RELATIONSHIP | CONNECTIVITY | ENTITY |
| COURSE(0,1) | HAS | 1:M | RESERVE(0,M) |
| BOOK(0,1) | IS DISPLAYED ON | 1:M | RESERVE(0,M) |
| COURSE(1,1) | INCLUDES | 1:M | CLASS(0,M) |
| INSTRUCTOR(1,M) (needs authorization) | TEACHES | 1:M | CLASS(0,M) |
| CLASS (1,1) | IS FOUND IN | 1:M | ENROLL(0,M) |
| STUDENT(0,1) | IS WRITTEN IN | 1:M | ENROLL(0,M) |
| SUPERVISOR(1,1) | SUPERVISES | 1:M | LIBRARIAN(0,M) |
| LIBRARIAN(1,1)  (needs authorization) | RESERVES | 1:M | BOOK(0,M) |
| BOOK(1,1) | IS WRITTEN IN | 1:M | CHECKOUT(0,M) |
| STUDENT(1,1) | IS FOUND IN | 1:M | CHECKOUT(0,M) |
| NOTE1: RESERVE is the composite entity that implements the M:N relationship “COURSE reserves BOOK”  NOTE2: ENROLL is the composite entity that implements the M:N relationship “STUDENT enrolls in CLASS”  NOTE3: CHECKOUT  is the composite entity that implements the M:N relationship “ BOOK is checked by STUDENT” | | | |

3. Based on the business rules and entity table, we drew the following ER Diagram for our database. The database is in 3rd normal form.

A close up of a map

Description automatically generated

4. Data Preparation:

Raw data file of the Course Reserves list was provided by the SJSU IT department. After communicating through several emails, they compiled the historical data from past academic year into an excel sheet. The raw Course Reserves list contains the course name, book title and ISBN numbers of the textbooks. There were three main problems that we recognized about this raw data file:

1. A lot of ISBN numbers were missing.

2. The raw data file contained multiple ISBN numbers for the same book. There were different versions of ISBN numbers such as the 10-digit ISBN10 number and the 13-digit ISBN13 number.

3. The raw data file contained course materials of various formats such as manuscripts in PDF format and DVDs.

Our solutions for cleaning up the raw course list file was:

1. We manually filled in all the missing ISBN13 numbers.

2. We decided to use ISBN13 as our primary key to the BOOK table, so we manually cleaned out all superfluous ISBNs and other book identification numbers.

3. We decided to only limit our database to textbooks with an ISBN number, so we cleared out materials of other formats.

4. We converted it into csv format.

For the course list, we manually copied course information from the SJSU course catalog webpage and pasted it into an excel sheet, then converted it into csv format.

We also generated some fictional data due to the unavailability of the actual data. The data files we generated were: student information, librarian information, instructor information, check-in and check-out history, enrollment information.

We used some built-in functions in Excel to generate random numbers of a specific format. For student ids , we first defined the function randbetween(000000000, 999999999) in the first cell and then dragged down to generate random 9-digit number combinations. Similar procedure was used to generate how many copies of textbook there were. We defined the range as 1 to 3 copies for each book.

For student and instructor names, we found the top 500 popular names online and randomly assigned them to the student’s and instructor’s first and last names. After that, we used the function concatenate() to generate their email addresses like that of the SJSU official email address.

5. Creating the database

We used MySQL workbench to create our database schema and built-in functions to import the data from csv file.

* List of Tables:

A screenshot of a cell phone

Description automatically generated

* User&Authorizations:

We defined three types of accesses: root access for the database supervisor who has administrative privilege to the database, he or she is allowed to perform all actions to the database; librarian access for the librarians who can add books to and backup the database; student access who can only run queries but cannot modify the database.

Root access authorizations:

A screenshot of a cell phone

Description automatically generated

Librarian access authorizations:

A screenshot of a cell phone

Description automatically generated

Student access authorizations:

A screenshot of a social media post

Description automatically generated

* Table contents:

BOOK: information on all books in the Course Reserves Program.

A screenshot of a cell phone

Description automatically generated

CHECKOUT: records borrowing history such as check-in and check-out records and due dates.

A screenshot of a cell phone

Description automatically generated

CLASS: class sections of each course.

A screenshot of a social media post

Description automatically generated

COURSE: courses in a semester

A screenshot of a social media post

Description automatically generated

ENROLL: enroll table to store which student is enrolled in which class. Allows textbook search based on courses that the student takes.

A screenshot of a cell phone

Description automatically generated

INSTRUCTOR: instructor information. Allows textbook search based on instructor names.

A screenshot of a cell phone

Description automatically generated

LIBRARIAN: librarians who manages the Course Reserves stack.

A screenshot of a cell phone

Description automatically generated

RESERVE: table that stores which textbooks are reserved for which course.

A screenshot of a cell phone

Description automatically generated

STUDENT: records of all students on campus.

A screenshot of a cell phone

Description automatically generated

SUPERVISOR: the 1 supervisor who manages all librarians.

A screenshot of a social media post

Description automatically generated

6. Queries

Our database should allow students to search for a specific book with various information and allow librarians to perform daily operations and information checking. We’ve written down a few queries that support typical functions of this system. The query statements, their functions and screenshots of their results are listed below:

Functions for the student’s side:

1. Search for course reserve textbooks with course id containing “ANTH”, such as “ANTH 107”.

SELECT COURSE.COURSE\_ID, COURSE\_NAME, COURSE\_CREDIT, INSTRUCTOR\_FNAME, INSTRUCTOR\_LNAME, BOOK.BOOK\_ISBN, BOOK\_TITLE, BOOK\_QUANTITY

FROM COURSE

JOIN RESERVE ON COURSE.COURSE\_ID=RESERVE.COURSE\_ID

JOIN BOOK ON RESERVE.BOOK\_ISBN=BOOK.BOOK\_ISBN

JOIN CLASS ON CLASS.COURSE\_ID=COURSE.COURSE\_ID

JOIN INSTRUCTOR ON CLASS.INSTRUCTOR\_ID=INSTRUCTOR.INSTRUCTOR\_ID

WHERE COURSE.COURSE\_ID LIKE "%ANTH%";

A screenshot of a computer

Description automatically generated

2. Search for course reserve textbooks with course name containing “American”, such as “American History”.

SELECT COURSE.COURSE\_ID, COURSE\_NAME, COURSE\_CREDIT, INSTRUCTOR\_FNAME, INSTRUCTOR\_LNAME, BOOK.BOOK\_ISBN, BOOK\_TITLE, BOOK\_QUANTITY

FROM COURSE

JOIN RESERVE ON COURSE.COURSE\_ID=RESERVE.COURSE\_ID

JOIN BOOK ON RESERVE.BOOK\_ISBN=BOOK.BOOK\_ISBN

JOIN CLASS ON CLASS.COURSE\_ID=COURSE.COURSE\_ID

JOIN INSTRUCTOR ON CLASS.INSTRUCTOR\_ID=INSTRUCTOR.INSTRUCTOR\_ID

WHERE COURSE.COURSE\_NAME LIKE "%American%";

A screenshot of a cell phone

Description automatically generated

3. Search for course reserve textbooks with instructor’s last name containing “al”, such as “Albert”.

SELECT COURSE.COURSE\_ID, COURSE\_NAME, COURSE\_CREDIT, INSTRUCTOR\_FNAME, INSTRUCTOR\_LNAME, BOOK.BOOK\_ISBN, BOOK\_TITLE, BOOK\_QUANTITY

FROM COURSE

JOIN RESERVE ON COURSE.COURSE\_ID=RESERVE.COURSE\_ID

JOIN BOOK ON RESERVE.BOOK\_ISBN=BOOK.BOOK\_ISBN

JOIN CLASS ON CLASS.COURSE\_ID=COURSE.COURSE\_ID

JOIN INSTRUCTOR ON CLASS.INSTRUCTOR\_ID=INSTRUCTOR.INSTRUCTOR\_ID

WHERE INSTRUCTOR\_LNAME LIKE "%al%";

A screenshot of a computer

Description automatically generated

4. Search for course reserve textbooks with book title containing “al”, such as “cultural”.

SELECT COURSE.COURSE\_ID, COURSE\_NAME, COURSE\_CREDIT, INSTRUCTOR\_FNAME, INSTRUCTOR\_LNAME, BOOK.BOOK\_ISBN, BOOK\_TITLE, BOOK\_QUANTITY

FROM COURSE

JOIN RESERVE ON COURSE.COURSE\_ID=RESERVE.COURSE\_ID

JOIN BOOK ON RESERVE.BOOK\_ISBN=BOOK.BOOK\_ISBN

JOIN CLASS ON CLASS.COURSE\_ID=COURSE.COURSE\_ID

JOIN INSTRUCTOR ON CLASS.INSTRUCTOR\_ID=INSTRUCTOR.INSTRUCTOR\_ID

WHERE BOOK\_TITLE LIKE "%al%";

A screenshot of a cell phone

Description automatically generated

Functions for librarians:

5. Search for borrowing record by student id.

SELECT STUDENT.STUDENT\_ID, STUDENT\_FNAME, STUDENT\_LNAME, BOOK.BOOK\_ISBN, BOOK\_TITLE, CHECK\_OUT\_DATE, CHECK\_IN\_DATE, CHECK\_DUE\_DATE

FROM STUDENT

JOIN CHECKOUT ON STUDENT.STUDENT\_ID=CHECKOUT.STUDENT\_ID

JOIN BOOK ON BOOK.BOOK\_ISBN=CHECKOUT.BOOK\_ISBN

WHERE STUDENT.STUDENT\_ID="335493990";

A screenshot of a cell phone

Description automatically generated

6. Search for available books.

SELECT BOOK\_ISBN,BOOK\_TITLE,BOOK\_QUANTITY

FROM BOOK

WHERE BOOK\_QUANTITY>0

A screenshot of a cell phone

Description automatically generated

7. Search for overdue books.

SELECT STUDENT.STUDENT\_ID, STUDENT\_FNAME, STUDENT\_LNAME, BOOK.BOOK\_ISBN, BOOK\_TITLE, CHECK\_OUT\_DATE, CHECK\_IN\_DATE, CHECK\_DUE\_DATE

FROM STUDENT

JOIN CHECKOUT ON STUDENT.STUDENT\_ID=CHECKOUT.STUDENT\_ID

JOIN BOOK ON BOOK.BOOK\_ISBN=CHECKOUT.BOOK\_ISBN

WHERE CHECK\_IN\_DATE>CHECK\_DUE\_DATE OR (CHECK\_IN\_DATE IS NULL AND CHECK\_DUE\_DATE<CURRENT\_DATE());

A screenshot of a cell phone

Description automatically generated

8. Find most popular textbooks so they may purchase more copies in the future.

SELECT CHECKOUT.BOOK\_ISBN, BOOK.BOOK\_TITLE, COUNT(CHECKOUT.BOOK\_ISBN) AS "TIMES"

FROM BOOK

JOIN CHECKOUT ON CHECKOUT.BOOK\_ISBN=BOOK.BOOK\_ISBN

GROUP BY CHECKOUT.BOOK\_ISBN

ORDER BY TIMES DESC;

A screenshot of a cell phone

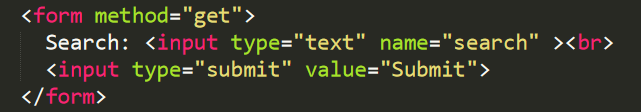
Description automatically generated

7. Web interface design

For the web interface, we designed a dashboard for Librarians to better monitor the borrowing behavior and perform searches on the database. We used Apache server from XAMPP, PHP and bootstrap template with HTML, CSS and JavaScript.

We tried 2 search options: 1. search by partial title name; 2. a drop-down list of other information.

For the search by partial title name function, we used an html form and HTTP get method to store user input in a global value. Code snippets are shown below:



A screenshot of a cell phone

Description automatically generated

But this search function was limited by the hardcoded table name, it can only search for results in the BOOK table. So, in further development we disabled this function and created a more dynamic search: we created a list of options for our users to select from. Each option corresponds to a specific query that will automatically create a view in our database.

The list currently contains four options: 1.Check for all the textbooks in the database; 2. Check for currently available books; 3. Check for all borrowing history(for analysis purposes); 4. Check all student information in the system(for analysis purposes). But it is easy for us to expand the list in the future.

A close up of a screen

Description automatically generated

After getting the selected option from user, our code automatically finds the corresponding query and create a view of the information selected. The views were created in our MySQL database.

A screenshot of a cell phone

Description automatically generated

Our code automatically creates the webpage table according to the view we’ve created and stored in the database. This search function was more dynamic than the previous one because it allows us to display different tables and results from joined tables.

A screen shot of a smart phone

Description automatically generated

A screen shot of a computer

Description automatically generated

And the display on the webpage looked like this, we’ve printed out the query statement for testing purposes:

1.Check for all the textbooks in the database:

A screenshot of a cell phone

Description automatically generated

2. Check for currently available books.

A screenshot of a social media post

Description automatically generated

3. Check for all borrowing history(for analysis purposes).

A screenshot of a social media post

Description automatically generated

4. Check all student information in the system(for analysis purposes).

A screenshot of a computer

Description automatically generated

**Conclusion**

In this project, we’ve theorized, designed, and implemented a MySQL database and a webpage interface to increase the usability and program efficiency of SJSU library’s Course Reserves program. By utilizing the skills learnt both in class and by ourselves, we’ve gained much hands-on experience and insight into web design, database design and management.

The goal of our project is to design a database that would facilitate and optimize the work of the SJSU Course Reserves program. This database should enable more search options and store more information on borrowing behaviors to enable further analysis and optimization.

The current Course Reserves search webpage only allows 2 search options: search by course name or instructor name. With our new design of user interface, it allows students to search for textbooks by partial course name, ISBN, partial title, course, class, instructor and availability. Our database also stores enough information for program managers to check textbook use frequency and borrowing behavior to optimize their operation. The information in our database will be enough for us to develop a machine learning model to optimize the Course Reserves program.

But our pursuit for better performance should not end here. In the future, we would like to expand our database to store different formats of materials using NoSQL database. We would also like to develop a recommendation system to provide research material recommendation based on borrowing history and an algorithm to produce most efficient purchase plan to both satisfy students’ needs and cut down cost for the Course Reserve Program.

Our project focuses on bringing a little convenience to our community. It can be of instant help to the SJSU library and Course Reserves program if implemented properly. It would provide a clear statistical analysis and visualization on the usage of Course Reserves, providing a better purchase plan for SJSU library as well as further promoting this program to SJSU students.