**OVERALL**

The GUI is implemented by JAVA. MySQL is used for building the database. Through the GUI, we call the SQL query functions to get data from database. Call the SQL update function to update the database.

**SCHEMA**

8 tables are needed for the database schema :

BOOK ( Isbn , Title )

BOOK\_AUTHOR (Isbn , Author\_id)

AUTHORS (Author\_id , Fullname ,Title, Fname, Mname, Lname, Suffix)

LIBRARY\_BRANCH (Branch\_id, Branch\_name, Address)

BOOK\_COPIES (Book\_id, Branch\_id, No\_of\_copies)

BORROWER ( Card\_no, Ssn, Fname, Lname, Address, Phone)

BOOK\_LOANS (Loan\_id, Isbn, Branch\_id, Card\_no, Date\_out, Due\_date, Date\_in) FINES (Loan\_id, fine\_amt, paid)

All the tuples in the tables except the FINES were inserted based on datas in the CSV files.During the whole program, BORROWER, BOOK\_LOANS and FINES will be updated.

**INTERFACE**

Use Jframe to hold all the buttons, textfields and tables. Buttons are used to call functions. Textfields are set for the users to input data for searching, checking out, checking in and creating new borrower and to display the error message or success message. Tables are used to display search result, copy, loans and fines. Tables are also used to get data by clicking any row of it.

**GLOBAL VARIABLES**

Connection variables are used to connect the SQL server. Statement variables are used to do SQL query or update. String variables are used to transfer data between functions for SQL query or update.

**FUNCTIONS**

**Search**

String

searchBook(String isbn,String title,String au1,String au2,String au3)

Click the searchButton to call the searchBook function.

Check whether all the blanks for searching are empty. If so, return an error message.

If not, use Table BOOK, BOOK\_AUTHORS, AUTHORS to retrieve ISBN, Title and All the authors of the book.

Display the results in MyTable.

Condition for searching can be one of ISBN, Title and Authors or combine any of them.

Max number of authors is 3. All the books of every author will be displayed.

Results are displayed on a JTable.

**GetCopy**

void getCopy(String isbn)

By clicking any row of the table to get the isbn and set the value to variable bookIdStr.

Then call the getCopy function to get number of copies in each branch using table BOOK\_COPIES and LIBRARY\_BRANCH.

Use the BOOK\_LOANS to get number of books which are loaned out in each branch.

Then use number of copies minus number of loaned to get number of available books.

Display the result in CopyTable.

**Checkout**

String checkout(String isbn,String branch\_id,String card\_no)

By clicking the checkout button to call checkout function.

Card\_no can’t be empty. If so, return error\_message.

Check whether the card\_no is in BORROWER table, if not, return error message. If so, continue the program.

To query the loan number of certain card\_no in table BOOK\_LOANS. If the number equals 3, return error message. If the number is smaller than 3, continue the program.

Use the same method in getCopy() table to see if there is available book in certain branch. If so continue the program, if not, return error message.

When all the conditions are meet, create a new turple to the book\_loans table.

Arrange a new loan\_id by adding 1 to the current count of tuples.

When book is checked out, available number of this branch\_id minus one.

**GetLoans**

String getLoan(String book\_id,String card\_no,String name)

By clicking getLoanButton to call getLoan() function.

Get all the loans which are not returned using BORROWER and BOOK\_LOANS table.

Display the result in the loanTable.

Conditions can be card\_no, isbn and borrower\_name. The three blanks can’t be all empty, if so, return error message.

**Check\_In**

String checkIn(String loanId)

By clicking any row of the loanTable to get the loan\_id for check\_in. Set the value to variable loan\_id\_Str.

By clicking the check\_in button to call the checkIn() function.

After getting loans, books which are not returned can be checked in. When books are checked in , date\_in of table BOOK\_LOANS should be set to current\_date();

Check if the current\_date() is over the due\_date(), if so call updateFines() and return message to ask the user to pay fines.

**UpdateFines**

String UpdateFines()

By clicking the updateFines button to call updateFines()

Check all the books which are not returned in table book\_loans.

Choose those loans whose due\_date is smaller than today.

If the loan\_id exists in the FINES table, update the Fine\_amt of the turple.

If not, insert a new turple into the FINES table.

Display sum of Fine\_amt of each card\_no which are not paid.

**SetPaid**

String SetPaid(String cardno)

By clicking any row of fineTable to get the cardno for payment.

By clicking the setPaid button to call the SetPaid() function.

Check whether the loans of the cardno are all returned. If not , return failure message. If so, set the paid of all the tuples in table FINES of the cardno to 1.

**CreateBorrower**

String CreateNew(String tempssn,String fname,String lname,String addr,String tempphone)

By clicking the createNew Button to call CreateNew() function.

First check if the ssn is in BORROWER table. If so, return error message.

Check whether the format of ssn and phone is right. If not, return error message.

Check whether address exists. If not, return error message.

If all the requirements are meet, insert a new tuple in the BORROWER table.