**Software Design of Library Management System**

By Hongyong Zhang

1. **Overview of system**
   1. Requirement
      1. Search book and check out

From the book information, librarian can search book typing the related string without specifying particular attribute. In this system, it allows to match the information of the attributes in ISBN, ISBN13, Author, Title, Publisher. On the other hand, if the search type is selected, the result set only contains the related information in the specified attribute.

Once the desired book is located, there is a mechanism to indicate if the book is available or not. Furthermore, if the number of the book kept in the library is more than one. Only if the number more than one can the book be checked out. In this case, the system implies that there is no differences among the same books in the library.

* + 1. Book’s loan and check in

In the book loan page, using the search function to filter all the book loans, i.e. look up for a student’s book loan records. If the book is still out, there should be an option to return the book.

Furthermore, during the check in of the book, system should automatically update its fine if needed.

* + 1. Borrower search and management

The same as the above page, there is a search function which can help the librarians out to locate the specific student for his/her information. i.e. the Card\_id.

On the other hand, there is another function to enter the information of a new borrower. After the insertion of the new borrower, the borrower page would display the new added borrower’s information the same as original ones.

* + 1. Fine and payment

The fine due to the late book is equal to the different days between the date out and the return day. However, even though the fine would be allowed to be estimated before the book is actually returned, the payment would be prohibited by the system until the book is returned.

There is a button which provides the librarians to update existing fines and possible fines.

* + 1. Data import

The CSV format file including the borrower&book information should be imported as the data to manipulate. It requires the system to separate the author and book information and their relation into three tables using the data normalization.

* 1. Software architecture

The system is implemented with servlet and JDBC which are used to provide the user interface and connection with MySQL.

The software of the application consists of three parts. The first part is the initialization of the connection between MySQL and application. The second part is the manipulation of the database combining the actions provided to the user. The third part is the xml file generator which produce the results using table, buttons to display the data.

Since the requirements of library management system have something in common. For an example, all the pages’ tables produced by the generator share the same basic function such as the manipulation of the pages. Each table can use several columns to sort the order of the rows.

Therefore, each requirement of the application only needs to initialize the parameters to specify and create its own xml page.

Due to the display of result set using the same table generator, the table has the same function to manipulate, i.e. clicking each column name row makes the result set sorted by that column. This feature helps unifying the same attributes of the GUI. The other distinctive features belonging to different pages could be customized by setting up the configuration and handlers to execute them.

The above diagram explains the transaction between user and software. The Web GUI is used to display all the data result from the XML generator, which is transferred by the server (such as Tomcat). The Servlet is the actually running object which is used to listen the request and parse the parameters to “XML generator” or “UploadHandler”. “JDBCUtil” is being called to initialize the database driver by the “Uploadhander” before the file is loaded into the database.

The main logic workflow about responding the request from users is implemented in the XML generator, which is responsible to retrieve the needed data and insert them into the xml stream. Then the servlet prints out the xml stream and transfer the result to the Web GUI (Browser) by server.

1. **Feature analyses**
   1. Search book & Check out

Under the book search page, if the user searches any content without specifying search type, then the software using SQL would retrieve the all the information containing the input characters.

After getting the information of the result set, the software decides whether the book could be checked out based on the available number of the book. If the book is still in the library, then software enables the “check out” button, otherwise, the button is disabled.

A student is being limited to check out three books.

* 1. Check in

Under the book loan page, the user could search books or students related loans in the table. And users can click the “check-in” button in the book’s row which is supposed to return to library. After checking in the book, system will modify the available number of the book in the database which brings the book back to the desk and allows another user to check out.

* 1. Add new borrower

Under the borrower management page, the software would list all the information of the borrowers in the library. User can enter a new borrower’s information to insert a new borrower, and the Card\_id and Ssn should be unique value other than the ones already existing in database. Otherwise, a failed prompt would show up.

* 1. Make payment

Before making the payment, the software would go through several procedures to make sure the payment could be done legally.

First, system tries to verify whether the book’s been checked in, if not, refuses to finish the payment. Secondly, system validates the loan in the database to make sure the loan exists. Finally, modifying the “paid” attribute to true indicates the payment has been done.

In order to simplify payment workflow, user can search the student who wants to pay for his/her fines. By selecting the “Amount Type” and “Paid Type”, user can pay for all the fines belongs to the student or pay for the particular fine by clicking the “To pay” button.

* 1. Update fines

The fine management page has a button to allow users to update the fines in the database. There are two possible scenarios to happen in the update function:

* + 1. The fines expired already exist in the database but not returned would be updated in the amount of payment.
    2. The potential fines whose loan has been expired, the book is returned but the fine is not existed in the database would be created in the database. This is a double check for fines which might be overlooked before.
  1. Import data & Initialize database

When the files are uploaded to the server, the server would parse the csv file into their own tables automatically. In that case, the files uploaded must be the same format as the files already given if users want to add more books and borrowers into database.

Since the value of “AvailNum” is set to 1 by default, so if user want to extend the application for multiple number of book, the software only needs to modify the parse function to adapt one more column which indicates the number of each book.

On the other hand, the parse function also tries to create tables automatically to make sure tables exist.