LIBRARY MANAGEMENT

DESIGN DOCUMENT

The library system management is designed so that the librarian can perform several activities such as check-in, check-out, add borrower, make fine payment, view payment history.

The architecture of the system follows MVC model, where the GUI is accessed by the user and the requests are processed and based on the request the data are fetched or updated in the database.

For storing data, MySQL is used as a database for storing information. Tables details:

1. Book table:

Book table stores information such as ISBN, cover, pages, publisher, and title. Isbn is the primary column. Extra column added to Book tables is available and is true by default. This column is set to false if a borrower checks out the book and it is updated to true once returned.

2. Authors table:

Authors table has author_id as the primary key and is auto increment column. Apart from this column, it has name column.

3. Book_Authors table:

This has a composite key, Author_id, and Isbn. Since a book can have multiple authors and one author can have multiple books so we have many to many relationships in this case.

4. Borrower table:

This table has card_id as the primary key and is auto increment column and ssn as a unique key. Since everyone has a unique ssn this column has a unique constraint. Additional columns in this table are address, bname, email, phone.

5. Book_loans table:

This table has loan_id as the primary key and is auto increment. Since many loan_id can be mapped to a single book so book_loans has many to one relationship with the book table. One user can have many book_loans so book_loans has many to one relationship with the borrower table. Date_out is updated with the current date at the time of check out and the due_date is updated with 14 days after date_out. The day the borrower returns the book the column is updated with that date.

6. Fines table:

This table has loan_id as the primary key and has one to one mapping with the book_loans table. The fine_amt stores the calculated fine amount. The paid column is by default is false if the fine is paid then this value is updated to true. PaidDate store the date in which the fine was paid.

JSF 2 with primefaces and hibernate is used in the project.

For the application to communicate with the database we are using hibernate.The entities are the java classes that have a mapping to the tables present in the database.

We are using managed bean for communicating with the GUI forms and get user inputs and manipulate data and display results to the users.

From the managed bean, the request to the database is accessed through the DAO classes.

Java mail jars are used to send an email when new borrower id is created.
It is assumed that the email id is already created for the library management system.

While adding books the user need to upload the .csv file only in the format provided.
If the pages size is zero, then that book is not updated in the database and the '&amp;' present in the csv is replaced with 'and'. The special characters are replaced by '';

It is assumed that while accessing the library management the tables are already created and the data from books.csv and borrower.csv files are imported into the database.

For initial setup, a new database is created from MySQL console.
Command: create database test4;

Now the tables are created using hibernate. For initial table creation,
<property name="hibernate.hbm2ddl.auto">create</property> is uncommented in the hibernate.cfg.xml. CreateTableTest.java is ran once which creates tables in the database.
After this, the property in the hibernate.cfg.xml is commented.
CSVReader.java is used to import the data into book, authors and book_authors table.
StudentCSVReader.java is used to import data into borrower table.

Technologies and server used:

1. Java 7
2. JSF 2
3. Primefaces 6
4. Hibernate 4
5. Apache Tomcat 7