Technical Design Document

Library Management System

Sangeeta Kadambala
Net ID: sxk160731
Version 1.0

Solution Design

1. System Architecture
   The library management uses java swing as GUI and connects to the MySql database using the mysql-connector-java-5.1.39.jar driver.

   1.1. Java Components:

   a. Class **DatabaseOperations** connects to the MySQL database and executes all updates and query statements and returns the respective results to the caller. It has two methods defined for these purpose:
      i. execQuery: This takes in a query in form of a String as input, executes it and returns the ResultSet of the query.
      ii. execUpdate: This takes in a DML statement in form of a string and returns the number of rows affected by executing the statement.

   b. Class **SearchPage** is responsible for the GUI display. It displays the main library search page to start with and allows the user to switch between different tabs that allow different functionality. Functionalities provided are:
      i. Search Books, ISBN or authors:
         Query used is :
```
SELECT DISTINCT
    ba.isbn, title
FROM
    book bk
        JOIN
    book_authors ba ON ba.isbn = bk.ISBN
        JOIN
    authors a ON ba.author_id = a.Author_id
WHERE
   LOWER(title) LIKE LOWER(%SEARCH KEY%)
    OR LOWER(bk.ISBN) LIKE LOWER(%SEARCH KEY%)
    OR LOWER(name) LIKE LOWER (%SEARCH KEY%)
Where SEARCH KEY is the input provided by the user.
```
      ii. Check Out of books : Allows check out only if:
         - The book is available:
```
SELECT
    COUNT(1)
FROM
    book_loans
WHERE
    isbn = 'isbn' AND date_in IS NULL;
Where ‘isbn’ is the ISBN of the book to be checked out.
```
         And if the borrower does not have:
         - any unpaid fines
```
SELECT
    COUNT(1) AS unpaidFineCnt
FROM
    fines
WHERE
    paid = 'U'
        AND loan_id IN (SELECT
            loan_id
```

```
                FROM
                    book_loans
                WHERE
                    card_id = 'card ID');
```

- overdue books

  ```
  SELECT
      COUNT(1) AS overdueBookCnt
  FROM
      book_loans
  WHERE
      due_date < SYSDATE()
          AND card_id = 'card ID'
          AND date_in IS NULL;
  ```

- 3 books already checked out.

  ```
  SELECT
      COUNT(1) AS maxCountForChkoutCnt
  FROM
      book_loans
  WHERE
      card_id = 'card ID' AND date_in IS NULL;
  ```

  Where ‘card ID’ is the borrower ID and is provided by the user.

iii. Adding Borrowers: Adds new borrowers. Doesn’t allow to add borrower with same SSN.

iv. Checking in of borrowed books
Checking in of overdue books creates a fine record:

```
if (dueDate.before(sysDate)) {
    insert into fines values ("loan id"," + fine + ",'U')
}
```

v. Paying the fines

vi. Displaying overdue Books
Query used:

```
SELECT
    title, bl.due_date, CONCAT(b.fname, ' ', b.lname) AS name
FROM
    book_loans bl,
    borrower b,
    book bk
WHERE
    bl.card_id = b.card_id
        AND bl.isbn = bk.ISBN
        AND bl.due_date < SYSDATE()
        AND bl.date_in IS NULL
        AND b.card_id = 'card ID';
```

1.2. Database components

a. Data for tables book and authors are directly imported from the generated clean tsv file using the import function in the workbench.

b. Data for tables book_authors , borrowers, address are populated using data in the temp table which in turn are populated by direct import using the workbench utility. Below are the temp table structures and the insert statements:

```
CREATE TABLE BOOK_TEMP (
   author_id int primary key auto_increment,
    isbn VARCHAR(10),
    title VARCHAR(200),
    author varchar(100)
);

ALTER TABLE BOOK_TEMP AUTO_INCREMENT = 101001;
```

```
insert into book_authors (author_id, isbn) select a.author_id , bt.isbn from authors
a, book_temp bt where a.author_id = bt.author_id;

CREATE TABLE borrower_temp (
    card_id VARCHAR(7),
    SSn varchar(12) UNIQUE,
    fname VARCHAR(50),
    lname VARCHAR(50),
    address_id INTEGER PRIMARY KEY AUTO_INCREMENT,
    address VARCHAR(100),
    city VARCHAR(20),
    state VARCHAR(2),
    phone varchar(20)
);

ALTER TABLE borrower_temp AUTO_INCREMENT = 201;

insert into borrower (card_id,ssn,fname,lname,address, phone) select
card_id,ssn,fname,lname,address_id, phone from borrower_temp;

insert into address (address_id, address,city,state) select address_id , address,
city, state from borrower_temp;
```

2. Assumptions
   - i. For every book in the system, only one copy of each is available.
   - ii. Search key is substring matched. i.e. if a user provides a string with words(strings with spaces in between), then the spaces are substring matched. Eg: Query: **what if**. The database is searched for the substring matching **'%what%if%'**
   - iii. Fines on a specific book is not visible until the book is returned. i.e. If a book is overdue and there is supposed to be a fine on the book, the fine can be seen only after the book is returned.
   - iv. Check-out and check-in consider current date as the check-in or check-out time.
   - v. Fines are calculated on a basis of $0.25 per day.
   - vi. For borrower table, SSN is the primary key and the only unique key(phone number is not unique).
   - vii. Table data in the GUI is not refreshed after any change(insert, update or delete). User needs to re input the operation to view the refreshed data.
   - viii. Complete address is split into 3 parts: Address, city, state.
   - ix. Check in and payment of fines can be done one at a time and not in bulk.

3. Third party Softwares used
   - a) Open Refine(http://openrefine.org/)
     This online tool is used to clean the noise in the csv data files. All the data cleaning done are as follows:
     - a. Removal of the '&amp;' characters
     - b. Splitting of the Authors column (the multivalue column separated by ',' in the csv file) to assign author ID to each Author(normalize author data) and allow isbn to author mapping.
     - c. Clustering different formats of the same value and converting them to a uniform value.
   - b) Eclipse neon
     This tool is used to create the java code using Swing API.
   - c) MySQL Workbench
     The data import functionality of this tool is used to import data into the tables from the tsv data file derived from the Open Refine software.