CPSC 304

Project Part III

Partial Implementation

Due: Monday, March 31, 11:55 am. (Yes, it's 11:55 a.m.)

In this part of the project, you will implement a part of the Library database and define a user friendly graphical interface for the system using Java and JDBC. Again, we would like all the groups to implement the same design. Therefore, we ask you to use the tables which are listed in the next section.

To implement the system you need to

- define the tables,
- write code for the transactions, and
- design a reasonable user interface.

Usually, the tables are created by the database administrator using an interactive SQL command interpreter For Oracle, this is the tool SQL*Plus that you used in the lab. Usually, we put the table definitions in a text file (with a .sql extension) and executed it using the related Oracle command, as it is explained in the tutorial.

The code for the transactions is usually broken down into a set of classes and methods in the host language (in our case, in Java). The user interface is built on top of that. This approach offers great modularity and flexibility. For instance, you can easily add more functions to the system or completely change the interface in the future. The Java/JDBC Tutorials describe how to use Java and JDBC with the Oracle server. You should study the corresponding set of tutorials before you go to your labs and follow the examples we have developed there.

Tables can also be defined using the JDBC API. Because this part will be done only once, when the initial database is created, the code for the table definitions is usually placed in a separate program that is run once. In this part you can follow any approach. Just make sure to submit all the code you write for marking.

<u>Tables for Part III</u>

You should provide complete definitions for the following tables:

- *Borrower* (<u>bid</u>, password, name, address, phone, emailAddress, sinOrStNo, expiryDate, type)
- *BorrowerType* (<u>type</u>, bookTimeLimit)

- *Book* (callNumber, isbn, title, mainAuthor, publisher, year)
- *HasAuthor* (callNumber, name)
- *HasSubject* (callNumber, subject)
- *BookCopy* (callNumber, copyNo, status)
- *HoldRequest*(hid, bid, callNumber, issuedDate)
- *Borrowing*(borid, bid, callNumber, copyNo, outDate, inDate)
- *Fine* (fid, amount, issuedDate, paidDate, borid)

Note:

- We only have three borrower types: students, faculty and staff with the borrowing limits that were listed in part 1.
- We only deal with books, nothing else. All books are borrowable.
- There are no branches (or, the library has only one branch).
- There are no reservations.
- The status of a book copy can be "on-hold", "in", "out".

Operations for Part III

For each table, it may be convenient to define the following operations:

- *insert* a tuple into the table
- *delete* a tuple from the table
- *display* all the rows of the table (you need this for debugging and demonstration).

These operations are NOT required for the project, but they will be very useful for debugging your system and also for demonstrating your project. Insert operations require values for the components of a tuple. These operations check for duplicates and insert a new tuple in the table. Delete operations require the key for a tuple and delete that tuple from the table. These operations will be used by the more complex transactions you are going to define next. Some insert operations need to generate new values for a key (like the key of a hold request and a borrowing record). The tutorials should have some information on that.

After the tables have been created, you should write code for the following transactions.

Transactions performed by a clerk:

- Add a new borrower to the library. The user should provide all the required information
- Check-out items borrowed by a borrower. To borrow items, borrowers provide their card number and a list with the call numbers of the items they want to check out. The system determines if the borrower's account is valid and if the library items are available for borrowing. Then it creates one or more borrowing records and prints a note with the items and their due day (which is giver to the borrower).

- Processes a return. When an item is returned, the clerk records the return by providing the item's catalogue number. The system determines the borrower who had borrowed the item and records that the item is "in". If the item is overdue, a fine is assessed for the borrower. If there is a hold request for this item by another borrower, the item is registered as "on hold" and a message is send to the borrower who made the hold request.
- Checks overdue items. The system displays a list of the items that are overdue and the borrowers who have checked them out. The clerk may decide to send an email messages to any of them (or to all of them).

Transactions performed by a borrower:

- Search for books using keyword search on titles, authors and subjects. The result is a list of books that match the search together with the number of copies that are in and out.
- Check his/her account. The system will display the items the borrower has currently borrowed and not yet returned, any outstanding fines and the hold requests that have been placed by the borrower.
- Place a hold request for a book that is out. When the item is returned, the system sends an email to the borrower and informs the library clerk to keep the book out of the shelves.
- Pay a fine.

Transactions performed by a Librarian:

- Adds a new book or new copy of an existing book to the library. The librarian provides the information for the new book, and the system adds it to the library.
- Generate a report with all the books that have been checked out. For each book the report shows the date it was checked out and the due date. The system flags the items that are overdue. The items are ordered by the book call number. If a subject is provided the report lists only books related to that subject, otherwise all the books that are out are listed by the report.
- Generate a report with the most popular items in a given year. The librarian provides a year and a number n. The system lists out the top n books that where borrowed the most times during that year. The books are ordered by the number of times they were borrowed.

For each transaction, make sure to check for the basic errors and produce appropriate messages for the user. As you implement each of these transactions, if you find out that some information that is necessary for the transaction is missing, you can make reasonable assumptions about it. Your document with the short introduction for the system should contain a list with all the assumptions you made in your implementation.

User Interface

Create an easy to use GUI that allows the users to execute all the transactions provided by the system, and display the results appropriately. Use the menu facility provided by the graphics environment you use to define appropriate menus and submenus. Make sure to design your interface in such a way that all the error messages appear in separate pop-up dialogue windows, or in a designated area of the main window, so that they don't interfere with the other activities.

Do not worry about access control. Your interface should have at least three menus: one for the clerks one for the borrowers, and another for the librarians. Nevertheless, your program should not provide any access control. It should let any user access any one of the menus.

Submission and Project Demonstration

Test your system with enough data to make sure that works well. GUIs normally require a lot of time and multiple tries to get them right. We expect you to produce a simple but adequate interface. You should not spend a lot of time on the interface itself. You should concentrate on the coding of the functionality of your project rather than the GUI. The main goal of the project is to give you some experience with SQL and JDBC.

For Part III, you should hand in:

- A cover page filled with the information for each partner.
- A brief introduction to the system and a brief user manual.
- The file with the code for the creation of the tables.
- The files with the code for the operations, the transactions and your interface.

The person from each group who will hand in this part should do the following:

- create a directory **p3** inside your **cs304** directory
- place the project files (cover page, user manual, which should be in html or pdf form, and the code files) in **p3.**
- hand it in using:

```
> handin cs304 p3
```

To evaluate your project better, we'll ask each group to demonstrate their system. Demonstrations will be scheduled during the week of March 31. Each demo will be for 20-30 minutes. We'll pass a time sheet around in the lecture for the groups to book their presentation times. The mark you will receive for this part of the project will be based on the quality of your presentation, the functionality of your program, the overall design, and the design of your interface. Each member of the group should come to the presentation prepared with a demonstration plan.

Finally, this part of the project is worth 50% of the total project mark. Each of part I and II is worth 25% of the total project mark.