

CSCI3170 (2014-2015 2nd term)

Introduction to Database Systems

Project– Library Inquiry System

Group Registration Deadline: 23:59 6th February 2015

Phase 1 Deadline: 23:59 13th March 2015

Phase 2 Deadline: 23:59 10th April 2015

1. Introduction

You are required to implement a Library Inquiry System for a library so that all information about books in the library, library users, and loans are stored. The system has to support interactive inquiries from users. You are required to use Java JDBC API to access the database. Our tutors will give a tutorial on how to use the JDBC API. You are required to implement a Java application to realize all system functions stated in this specification.

There are two phases. In phase 1, you are required to design the database for the system (including an ER-diagram and a relational schema). After the deadline of Phase 1, a suggested solution will be provided. You are required to use the suggested solution to complete Phase 2. In Phase 2, you are required to implement the Library Inquiry System as a Java command-line program. Our tutors will give tutorials on how to connect to a database system and deploy your work on the required platform.

This is a group project and each group consists of three members. ONLY one copy of solution is required for each group. Please sign the group registration on our course homepage before the group registration deadline.

2. Milestones

2.1. Preparation

- Read the document thoroughly and make sure you understand all the assumptions and regulations stated in Section 4.

2.2. Phase 1

- According to the data specifications in Section 3, design an ER-diagram and transform it into a relational schema which doesn't contain redundant fields and tables.

2.3. Phase 2

- According to the suggested solution of Phase 1, implement a Java application that fulfills all requirements stated in Section 5.
- Debug your system with different datasets and user inputs.
- Write a readme file to describe the compilation and deployment of your system.

3. Data Specification

All data files of the system are in UNIX text file format (i.e. Newline character is `\n`) encoded in ASCII. Your Java application is required to read records stored in the files and insert them into appropriate tables of the provided Oracle DBMS via JDBC API. There are four input files, a list of categories, a list of library users, a list of books and a list of check-out records.

3.1. Categories

Library users are divided into different categories. The loan period and the maximum number of books that can be borrowed by a library user are determined by his or her category. Each category has a unique category ID.

Item Name	Format	Description
Category ID	Non-empty positive integer with 1 digit	A unique identifier for a category.
Max books	Non-empty positive integer with at most 2 digits	The maximum number of books that can be borrowed by the library user in the corresponding category.
Loan period	Non-empty positive integer with at most 2 digits	The number of days for the library user in the corresponding category to borrow a book.

3.2. Library Users

Each library user has a unique user ID and belongs to exactly one category.

Item Name	Format	Description
User ID	Non-empty string with 10 characters	A unique identifier of the library user.
Name	Non-empty string with at most 25 characters	The name of the library user.
Address	Non-empty string with at most 100 characters	The address of the library user.
Category ID	Non-empty positive integer with 1 digit	It indicates which category the library user belongs to.

3.3. Books

Each book is a literary composition written by one or more authors and a library stores a number of physical copies for each book. Each book has a unique call number

Item Name	Format	Description
Call number	Non-empty string with 8 characters	It is used for the library users to search for the book.
Number of copies	Non-empty positive integer with 1 digit	The number of identical copies of the book.
Title	Non-empty string with at most 30 characters	The title of the book.
Author(s)	Non-empty string with at most 25 characters for each author	Author name(s) of the book concatenated as a string with comma character as the delimiter.
Date of publication	Non-empty date in the format of DD/MM/YYYY	The date that the book is published

3.4. Checked-Out Records

Each check-out record shows a borrow history of a library user.

Item Name	Format	Description
Call number	Non-empty 8 characters	The call number of the checked-out book copy.
Copy number	Non-empty 1 digit Positive integer	The copy number of the checked-out book copy.
User ID	Non-empty 10 characters	The user ID of the borrower.
Check-out date	Non-empty date in the format of DD/MM/YYYY	The date that the book is checked-out.
Return date	Date in the format of DD/MM/YYYY	The date that the book is returned.

4. Assumption and Regulations

4.1. System

- All numerical values will not be larger than the maximum integer value that can be handled by Java.
- The system is case sensitive.
- All the dates in the whole system has the following format: [DD]/[MM]/[YYYY] and have the same time zone as Hong Kong. (Note: Y=year, M=month, D=day)
- There is no duplicate row in any input and output.
- The current date is the current system date in the deployment machine.
- Your Java program may assume that any value entered into any input field is correct **in format only**.
- Your Java program may assume that any data file inputted into it is **correct in format and content**.

4.2. Categories

- Each category has a unique *category id* and it can be used to identify a category.
- Some library users may have the same maximum loan period or number of book that can be borrowed.

4.3. Library Users

- Each library user has a unique *user id* and it can be used to identify a library user.
- Some library users may have the same name or the same address.
- A user can only belong to one user category.

4.4. Books

- Each book has a unique *call number* and it can be used to identify a book.
- Some books may have the same title.
- A book copy refers to a physically existing book in the library.
- Each book copy has a unique pair of *call number* and *copy number* and they can be used jointly to identify a book copy.
- The book copy of each book has a copy number begins with one. If a book has four copies, these four book copies have the same call number, and their copy numbers are 1, 2, 3, 4.
- Each book must have at least one author.
- An author can be uniquely identified by his/her name.
- An author may write more than one book.

4.5. Checked-Out Records

- Each check-out record has a unique set of {*Call number*, *Copy number*, *User ID* and *Check-out date*} and this set of attributes can be used to identify a check-out record.
- Some library users may have never checked out any book copies.
- Some book copies may have never been checked out.
- A user may borrow the same book copy for more than one time.
- The return date of a book copy is NULL if the book copy is not returned.

4.6. Dataset

- The sample data set will be provided to you after the deadline of Phase 1. The details will be posted in the course homepage at that time.

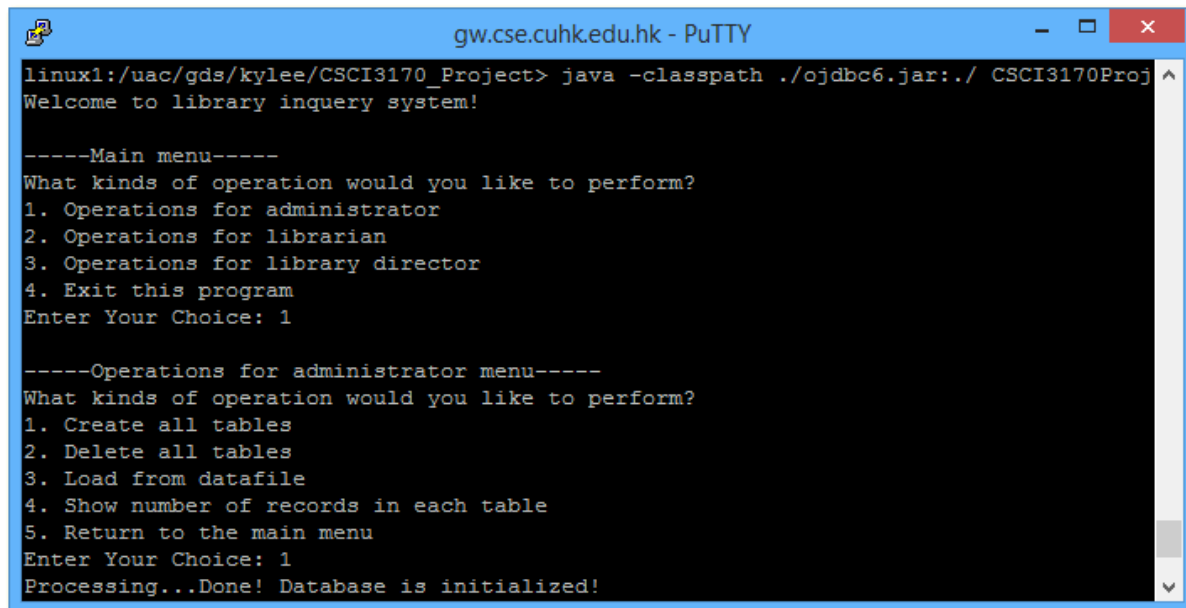
5. System Function Requirements

You are required to write a simple command line application in Java. After performing a function specified in any of the following sub-sections, the program should go back to the topmost level of menu. Any error or informative message of the Java program should be displayed in a new line. The Java program consists of the following functions:

5.1. Administrator

The functions that can be used by an administrator are:

- **Create table schemas in the database:** This function creates all the tables for this system based on the relational schema given.



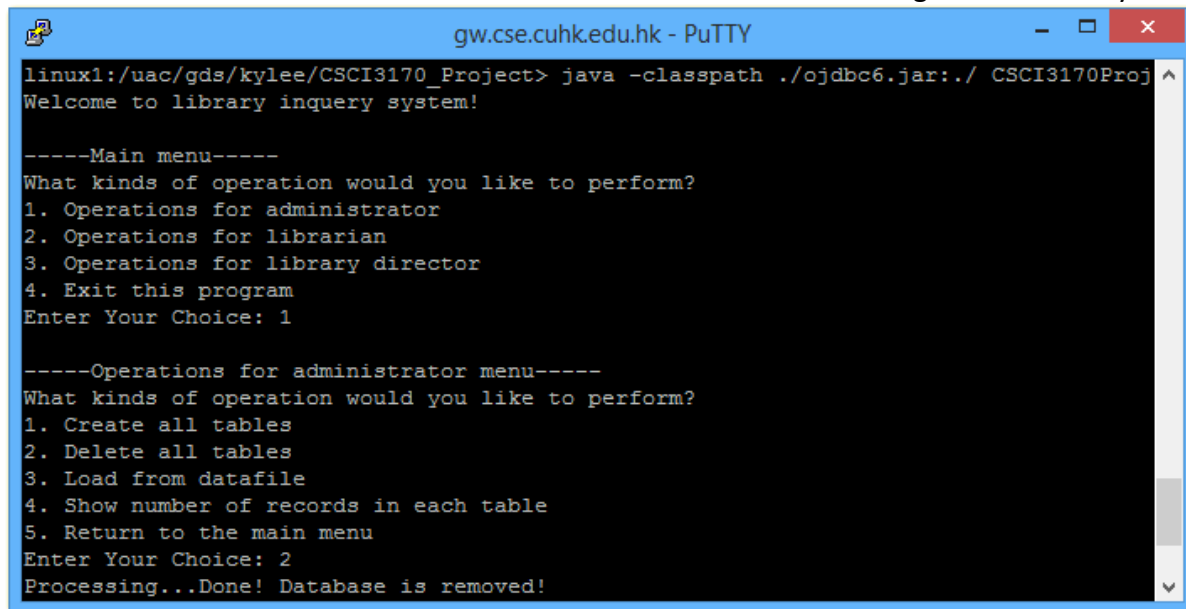
```
linux1:/uac/gds/kylee/CSCI3170_Project> java -classpath ./ojdbc6.jar:./ CSCI3170Proj ^
Welcome to library inquiry system!

-----Main menu-----
What kinds of operation would you like to perform?
1. Operations for administrator
2. Operations for librarian
3. Operations for library director
4. Exit this program
Enter Your Choice: 1

-----Operations for administrator menu-----
What kinds of operation would you like to perform?
1. Create all tables
2. Delete all tables
3. Load from datafile
4. Show number of records in each table
5. Return to the main menu
Enter Your Choice: 1
Processing...Done! Database is initialized!
```

Figure 1: Expected interactive input and output while creating table schemas

- **Delete table schemas in the database:** This function deletes all existing tables in the system.



```

gw.cse.cuhk.edu.hk - PuTTY
linux1:/uac/gds/kylee/CSCI3170_Project> java -classpath ./ojdbc6.jar:./ CSCI3170Proj ^
Welcome to library inquiry system!

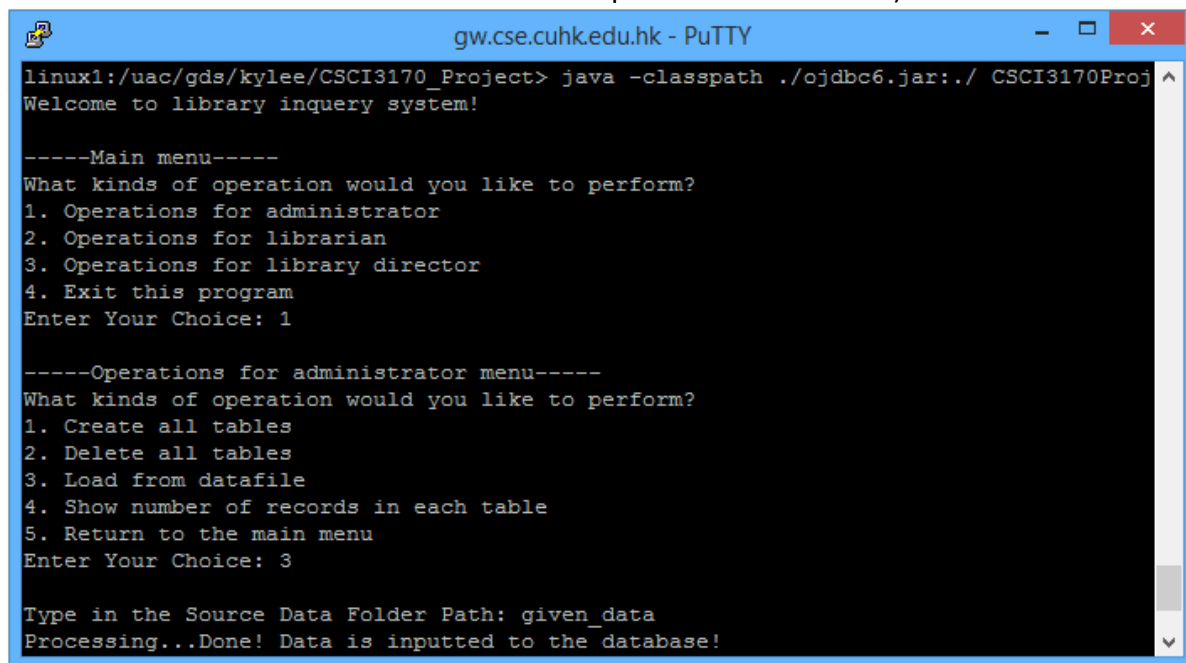
-----Main menu-----
What kinds of operation would you like to perform?
1. Operations for administrator
2. Operations for librarian
3. Operations for library director
4. Exit this program
Enter Your Choice: 1

-----Operations for administrator menu-----
What kinds of operation would you like to perform?
1. Create all tables
2. Delete all tables
3. Load from datafile
4. Show number of records in each table
5. Return to the main menu
Enter Your Choice: 2
Processing...Done! Database is removed!

```

Figure 2: Expected interactive input and output while deleting table schemas from the database

- **Load data from a dataset:** This function reads all data files from a user-specified folder and inserts the records into the appropriate table in the database. (Your program can assume that the user-specified folder must contain all 4 input files. These 4 input files are named category.txt, user.txt, book.txt and record.txt. Each data file stores the data corresponds to its filename.)



```

gw.cse.cuhk.edu.hk - PuTTY
linux1:/uac/gds/kylee/CSCI3170_Project> java -classpath ./ojdbc6.jar:./ CSCI3170Proj ^
Welcome to library inquiry system!

-----Main menu-----
What kinds of operation would you like to perform?
1. Operations for administrator
2. Operations for librarian
3. Operations for library director
4. Exit this program
Enter Your Choice: 1

-----Operations for administrator menu-----
What kinds of operation would you like to perform?
1. Create all tables
2. Delete all tables
3. Load from datafile
4. Show number of records in each table
5. Return to the main menu
Enter Your Choice: 3

Type in the Source Data Folder Path: given_data
Processing...Done! Data is inputted to the database!

```

Figure 3: Expected interactive input and output while loading table schemas from the database

- **Show the number of records in each table:** This function shows the total number of records in each existing table.

```

gw.cse.cuhk.edu.hk - PuTTY
linux1:/uac/gds/kylee/CSCI3170_Project> java -classpath ./ojdbc6.jar:./ CSCI3170Proj
Welcome to library inquiry system!

-----Main menu-----
What kinds of operation would you like to perform?
1. Operations for administrator
2. Operations for librarian
3. Operations for library director
4. Exit this program
Enter Your Choice: 1

-----Operations for administrator menu-----
What kinds of operation would you like to perform?
1. Create all tables
2. Delete all tables
3. Load from datafile
4. Show number of records in each table
5. Return to the main menu
Enter Your Choice: 4
Number of records in each table:
Table 1: xxx
Table 2: xxx
Table 3: xxx
Table 4: xxx

```

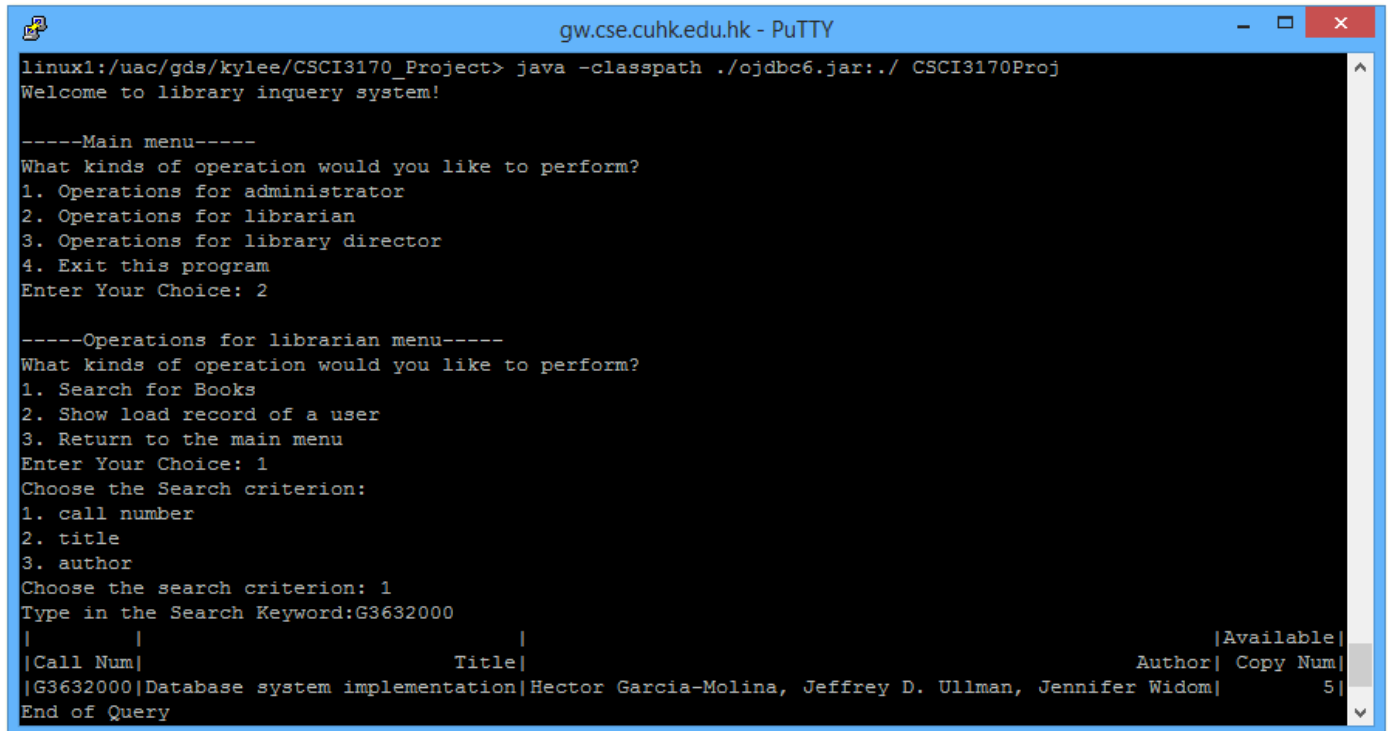
Figure 4: Expected interactive input and output while showing number of records in each table

Note: Please replace the words in *Italic* in the figure 4 with the tables in relational schema given in the suggested solution of phase 1. The number of tables may not be four as shown in figure 4.

5.2. Librarian

- **Search for books:** The system has to provide an interface to allow a librarian to search for the books in the library in three different ways
 - By call number (exact matching)
 - By title (partial matching)
 - By author (partial matching)

You can assume that only one searching method can be selected by the librarian for each query. After the librarian entered the search keyword, the program should perform the query and return all the matching books in terms of their *call number*, *title*, *authors* and *number of available copies*. The results of the query should be sorted in ascending order of *call number* and outputted as a table as follows:



```

linux1:/uac/gds/kylee/CSCI3170_Project> java -classpath ./ojdbc6.jar:./ CSCI3170Proj
Welcome to library inquiry system!

-----Main menu-----
What kinds of operation would you like to perform?
1. Operations for administrator
2. Operations for librarian
3. Operations for library director
4. Exit this program
Enter Your Choice: 2

-----Operations for librarian menu-----
What kinds of operation would you like to perform?
1. Search for Books
2. Show load record of a user
3. Return to the main menu
Enter Your Choice: 1
Choose the Search criterion:
1. call number
2. title
3. author
Choose the search criterion: 1
Type in the Search Keyword:G3632000

```

Call Num	Title	Author	Copy Num	Available
G3632000	Database system implementation	Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom	5	

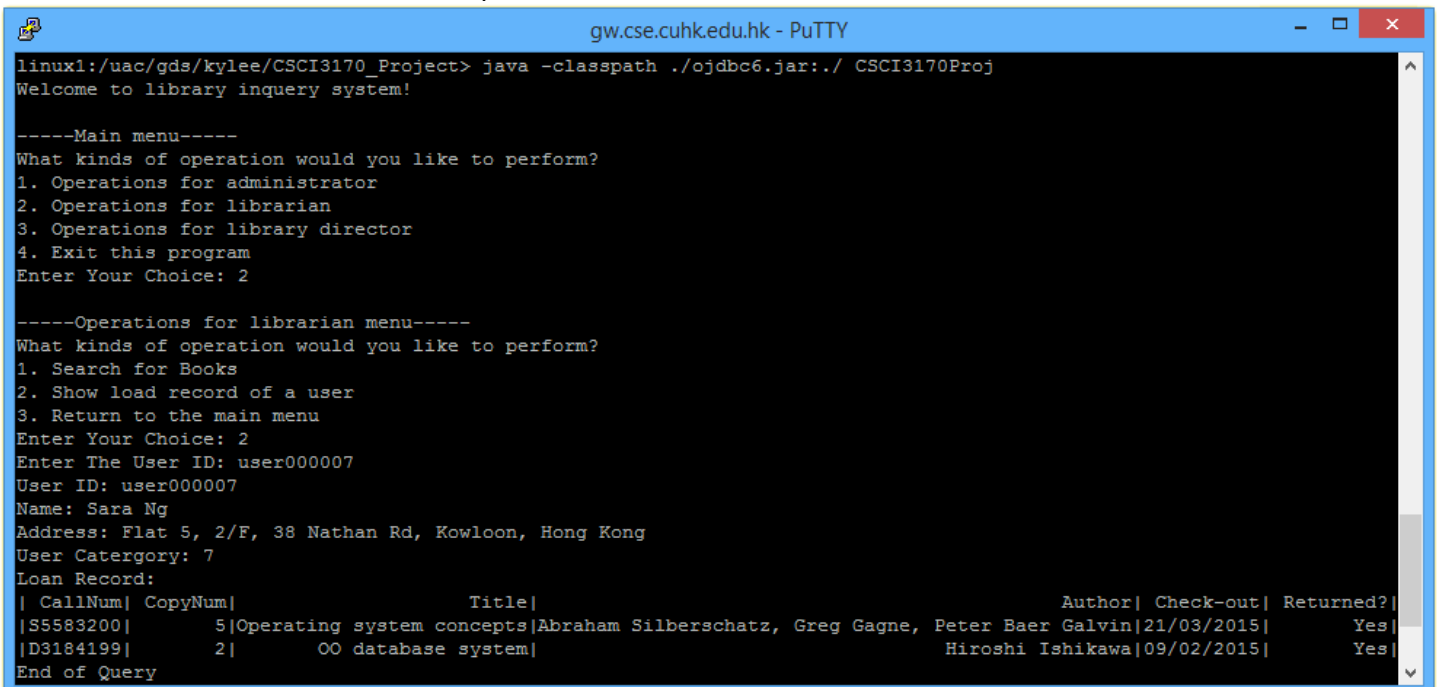
```

End of Query

```

Figure 5: Expected input and output while searching for books

- **Show all check-out records of a library user:** The system has to provide an interface to allow a librarian to show all check-out records of a library user with a given user ID. After the librarian enters the given user ID, the program will perform the query and return all the matching check-out records in term of *call number, copy number, title, authors, check out date and whether the book copy of the corresponding check-out record is returned*. The check-out records should be sorted in descending order of *check-out date* and outputted as a table as follows:



```

linux1:/uac/gds/kylee/CSCI3170_Project> java -classpath ./ojdbc6.jar:./ CSCI3170Proj
Welcome to library inquiry system!

-----Main menu-----
What kinds of operation would you like to perform?
1. Operations for administrator
2. Operations for librarian
3. Operations for library director
4. Exit this program
Enter Your Choice: 2

-----Operations for librarian menu-----
What kinds of operation would you like to perform?
1. Search for Books
2. Show load record of a user
3. Return to the main menu
Enter Your Choice: 2
Enter The User ID: user000007
User ID: user000007
Name: Sara Ng
Address: Flat 5, 2/F, 38 Nathan Rd, Kowloon, Hong Kong
User Category: 7
Loan Record:

```

CallNum	CopyNum	Title	Author	Check-out	Returned?
S5583200	5	Operating system concepts	Abraham Silberschatz, Greg Gagne, Peter Baer Galvin	21/03/2015	Yes
D3184199	2	OO database system	Hiroshi Ishikawa	09/02/2015	Yes

```

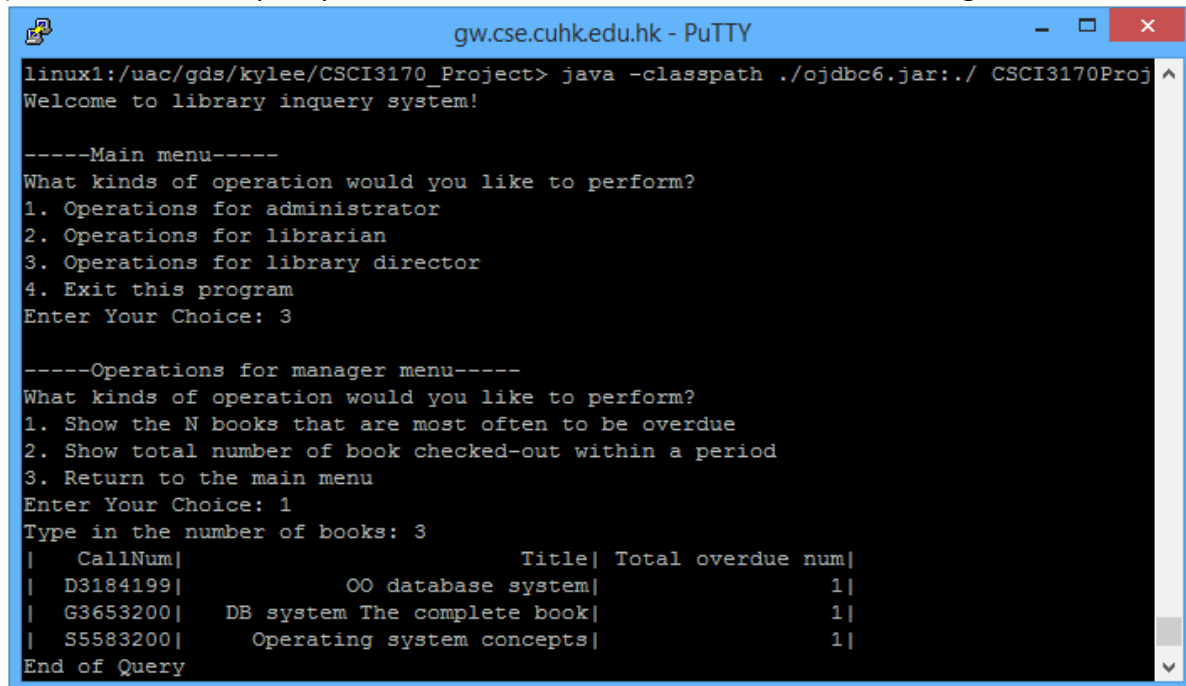
End of Query

```

Figure 6: Expected input and output while showing all check-out records of a library user

5.3. Library director

- **Show the N books that are most often to be overdue:** The system has to provide an interface to allow a library director to show the N books that are most often to be over-due. After the library director enters the number of books (N) that he/she wants to list, the program will perform the query and return the N books that are most often to be over-due in terms of *call number*, *title* and *total number of over-due*. The books should be sorted in descending order of number of over-due as the table below. (For the sake of simplicity, the check-out record with a null return date is ignored and $N > 0$)



```

linux1:/uac/gds/kylee/CSCI3170_Project> java -classpath ./ojdbc6.jar:./ CSCI3170Proj
Welcome to library inquiry system!

-----Main menu-----
What kinds of operation would you like to perform?
1. Operations for administrator
2. Operations for librarian
3. Operations for library director
4. Exit this program
Enter Your Choice: 3

-----Operations for manager menu-----
What kinds of operation would you like to perform?
1. Show the N books that are most often to be overdue
2. Show total number of book checked-out within a period
3. Return to the main menu
Enter Your Choice: 1
Type in the number of books: 3

```

CallNum	Title	Total overdue num
D3184199	OO database system	1
G3653200	DB system The complete book	1
S5583200	Operating system concepts	1

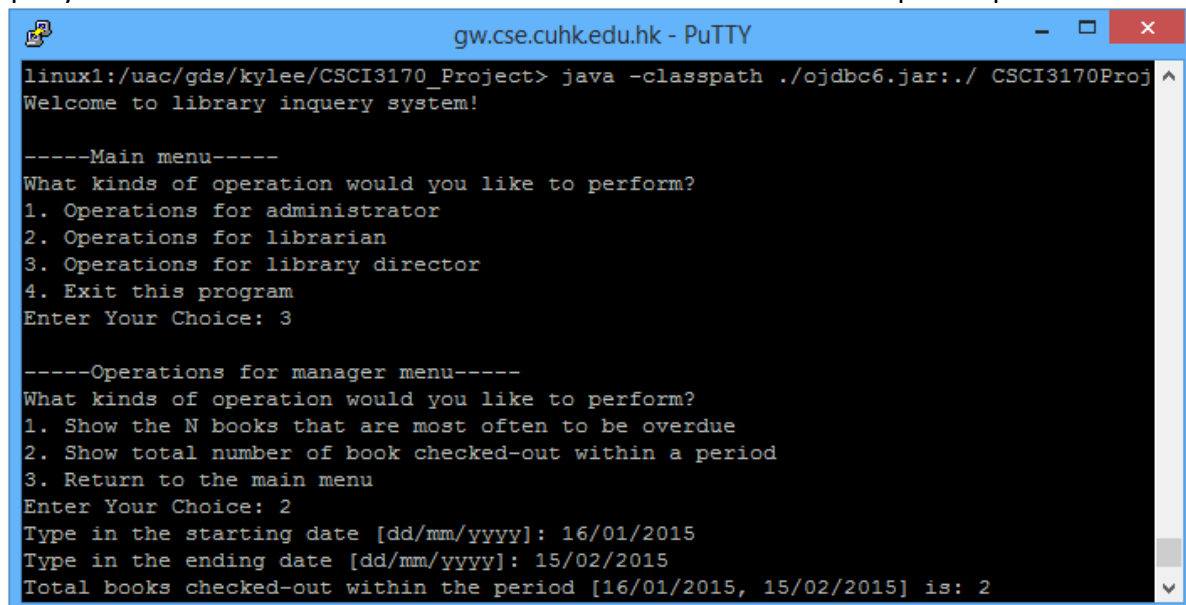
```

End of Query

```

Figure 7: Expected input and output while showing the N books that are most often to be overdue

- **Show total number of book checked-out within a period:** The system has to provide an interface to allow a library director to show the total number of book checked-out within a given period (e.g. from 16/01/2015 to 15/02/2015). After the library director enters the period, the program will perform the query and return the total number of book checked-out within the inputted period inclusively.



```

linux1:/uac/gds/kylee/CSCI3170_Project> java -classpath ./ojdbc6.jar:./ CSCI3170Proj
Welcome to library inquiry system!

-----Main menu-----
What kinds of operation would you like to perform?
1. Operations for administrator
2. Operations for librarian
3. Operations for library director
4. Exit this program
Enter Your Choice: 3

-----Operations for manager menu-----
What kinds of operation would you like to perform?
1. Show the N books that are most often to be overdue
2. Show total number of book checked-out within a period
3. Return to the main menu
Enter Your Choice: 2
Type in the starting date [dd/mm/yyyy]: 16/01/2015
Type in the ending date [dd/mm/yyyy]: 15/02/2015
Total books checked-out within the period [16/01/2015, 15/02/2015] is: 2

```

Figure 8: Expected input and output while showing total number of book checked-out within a period

6. Grading Policy

The marks are distributed as follows:

Phase	Content	Mark Distribution
1	ER-diagram	10%
	Relational schema (based on your ER-diagram)	10%
2	Java application	80%

- There will be a mark deduction if your application is terminated unexpectedly during the demonstration.
- You are not allowed to modify any source code during the demonstration.
- All members in the same group will receive the same marks for the project. In order to encourage every student to participate in the project, a question about this project may be asked in the final examination.

7. Demonstration

- All groups need to sign up for a demonstration on their works for phase 2, the registration page would be posted on the course website later.
- All group members should attend the demonstration.
- The duration for the demonstration for each group is about 20 minutes.
- The Java application will be tested in a Linux 64bit machine in the CSE department.
- The dataset used in the demonstration may be different from the dataset provided.

8. Submission Methods

8.1. Phase 1

- Submit a PDF file (one copy for each group) to the collection box at eLearning platform.
- The PDF file should consist of your groups ER diagram, relational schema, the group number, the names and the student IDs of all group members of your group.

8.2. Phase 2

- Submit a ZIP file (one copy for each group) to the collection box at eLearning platform. The ZIP file should consist of all your source codes and a README file (README.txt), which contains:
 - The group number of your group
 - The name and the student ID of each group members of your group
 - List of files with description
 - Methods of compilation and execution