**Database assignment – Library**

This assignment concerns defining your own database structure and accessing this database from a Java program.

Students and teachers can lend books from the faculty library. Different limits for students and teachers apply regarding the amount of books that can be borrowed simultaneously. Also, students and teachers can lend the books for a different number of days.

Students need to deposit money in order to borrow books. In order to borrow a book a student needs to have at least 10 euro in his/her account. If a book is returned late, students and teachers need to pay a fine. For each day the book is returned late, students and teachers need to pay 0.50 euro.

For testing purposes the current date (in the context of the application) needs to be stored in the database

The application needs to provide the following functionality:

1. Set the current date. This is needed for testing purposes. You may assume that only forward changes are allowed, i.e. the new date is later than the date currently stored
2. Enter and store information about a book. Here title, author and number of copies are mandatory fields that need to be entered by the user. The database should generate a unique id for each book, i.e. different copy of a book needs to have a different id
3. Enter and store info about students: name, student id (a unique 4 digit number) and a telephone number
4. Enter and store info about teachers: name, staff id ( a unique 3 character string) and department name
5. Set limit for number of books that can be borrowed by students
6. Set limit for number of books that can be borrowed by teachers
7. Set number of days a book may be borrowed by students
8. Set number of days a book may be borrowed by teachers
9. Deposit money for one student. This function is only applicable for existing students
10. Borrow a book by a student. This consists of the following steps:
    1. First the student needs to enter his name and student id. Here the student account is blocked, i.e. no books can be borrowed, after three consecutive wrong attempts. If name and student id yield a match in the database, the student is allowed to go to the next step.
    2. Enter the name and author of the book that student wants to borrow. If and only if (1) a copy of this book is available, (2) the student has not reached the limit for the amount of books and (3) the amount of money in the deposit of the concerned student is at least 10 euro, the student is lent a copy of the book.
    3. If student has still not reached the limit for the amount of books, the student is offered the choice to borrow another book.
11. Borrow a book by a teacher. This consists of the following steps:
    1. First the teacher needs to enter his name and staff id. Here the teacher account is blocked, i.e. no books can be borrowed, after two consecutive wrong attempts. If name and staff id yield a match in the database, the teacher is allowed to go to the next step.
    2. Enter the name and author of the book that teacher wants to borrow. If and only if (1) a copy of this book is available and (2) the teacher has not reached the limit for the amount of books, the teacher is lent a copy of the book.
    3. If teacher has still not reached the limit for the amount of books, the teacher is offered the choice to borrow another book.

1. Return book. This consists of the following steps:
   1. User enters the unique id of the book.
   2. If this book was returned within the allowed number of days (note: this number may be different for students and teachers) the returned book is accepted and stored in the library again. The steps below need to be skipped.
   3. If this book was returned after the allowed number of days, the program needs to calculate the fine that needs to be paid.
   4. If the book was lent by a teacher the fine needs to be added to the total amount that still needs to be paid by that teacher. The returned book is accepted and stored in the library again. The steps below need to be skipped.
   5. If the book was lent by a student and the calculated fine is smaller than the deposited amount of that student, the fine needs to be subtracted from this deposit. The returned book is accepted and stored in the library again. The steps below need to be skipped.
   6. If the book was lent by a student and the calculated fine is larger than the deposited amount of that student, the return of the book is not accepted and this function is ended, i.e. student first needs to deposit sufficient money and then start this “return book” function again.
2. Generate overview of all books that are currently lent to either students or teachers. In this overview the book id, name of the book, name of student or teacher and last allowed return date needs to be presented.
3. Generate overview of the amount of money (fines) that still needs to be paid by teachers. In this overview the name of the teacher, the department name and total fine to be paid needs to be presented, using the total fine to order this list.
4. Generate overview of the amount of fines that still need to be paid by students. In this overview the name of the student, student id and total fine to be paid needs to be presented, using the total fine to order this list.

You need to define a database capable of efficiently storing the required information (note: you need more than one database table). Furthermore you need to write a Java program that contains a menu allowing a user to perform any of the required actions on this database.

A user of your program should be able to perform an unlimited number of subsequent actions. For each step the user needs to be able to select any of the available actions or choose to exit the program.

Interaction with the user may be done via the Netbeans output window (no fancy user interface is required).

Any libraries you use in your program need to be included in your deliverable.

Information for accessing your database (e.g. hostname, port, username, password) needs to be defined in one location of your program and stored in final static variables.

Deliverables (uploaded to the indicated Blackboard assignment):   
- A small report describing the database tables in detail  
- A sql dump file of the complete database (including definition of tables)  
- The Java program  
- In addition you may be asked to demonstrate this program.