CSCI 585 - Database Systems - Fall Semester 2013

Assignment 2 - Description

Due: 11/6/2013 4:59 PM

Front Notes and Tips

- MOSS WILL BE USED FOR CHECKING ANY POTENTIAL CODE COPYING!
- Make sure you follow the discussions on the discussion board for any future communications and clarifications about this assignment.
- All questions should be posted on the discussion board or during the TA office hours; please DO NOT email or call the TAs. TAs will answer all the questions on the discussion board on a daily basis roughly around 12pm and 6pm.

Description

With this assignment you will design and implement an end-to-end java application that queries a relational database. You will create a database based on the provided ER model, and write a Java program which uses "JDBC" to query your relational database.

Input:

We have provided the following items for you:

1. *ER model*: This is a simplified version of ER model you generated as part of Assignment 1. With this assignment you will normalize the model such that it becomes 3NF. Thereafter, you will implement the normalized model as an Oracle database.
2. *data.xlsx*: You need to populate the database you create as part of this assignment. This Microsoft Excel file provides the data you need to populate onto your database.
3. A Java GUI program, namely "Assignment2.java", which serves as a template for the program you need to develop; you will develop your program by completing this template. You can compile the provided code in any Java IDE. e.g. Eclipse.

Output/Deliverables:

Part 1:.sqlFiles

You are required to prepare and submit two .sql files:

1. *createdb.sql*: This file should create all required tables of the database. In addition, it should also populate the tables with provided data and create index on columns of the tables, wherever appropriate.
2. *dropdb.sql*: When executed, this file should drop all tables and other objects created by createddb.sql.

Part 2:DescriptionFile

You are required to prepare and submit a description file including the following information

in fourseparate parts:

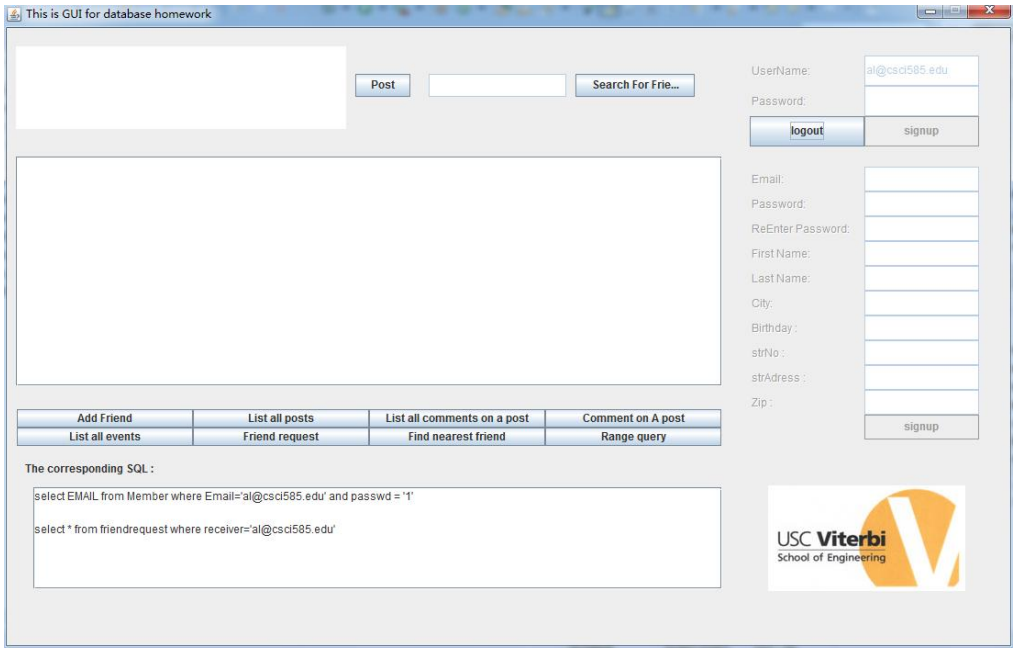
1. Your name, student id, and how we should compile and run your program.
2. The normalized version of the input ER schema/model.
3. An explanation of the reason you decide tocreate index (and perhaps a particular type of index) on each indexed column of the tables, wherever applicable.
4. All assumptions that you may make regarding your proposed normalization and indexing (optional).

Part 3: Java Program

You are required to complete the provided template Java GUI program by including the missing functions(please find guiding commentsin the template) and submit the completed program. Note that when the functions are executed through the GUI, in addition to executing the logic of the function (explained below), your code should display the corresponding SQL query (-ies) used for execution of each function in the "corresponding SQL query" box at the bottom of the GUI. The completed program queries the database to provide the following functionalities:

1. *Login and signup*

   With this function, the user can sign up as a new user or login to use his/her existing account. When signing up as a new user, the program should check the validity of theinput, i.e., an email address existing in the database can't be used as the Email address for the new user, "password" and "Reenter password" should match, Birthday should satisfy the input format you specify, StrNo. and Zip should be numbers. Also, no input field can remain unfilled.



2. *Search for friend*

   The user can use thesearch box to search for andobtain information about all users whose profile (Email, First Name or Last Name) matches the search query string; The information to be listed should include: Email, First Name, Last Name, Birthday, Street No., Street Addr., City, State, Zip, Country; empty string is not allowed in the search box.

3. *Add Friend*

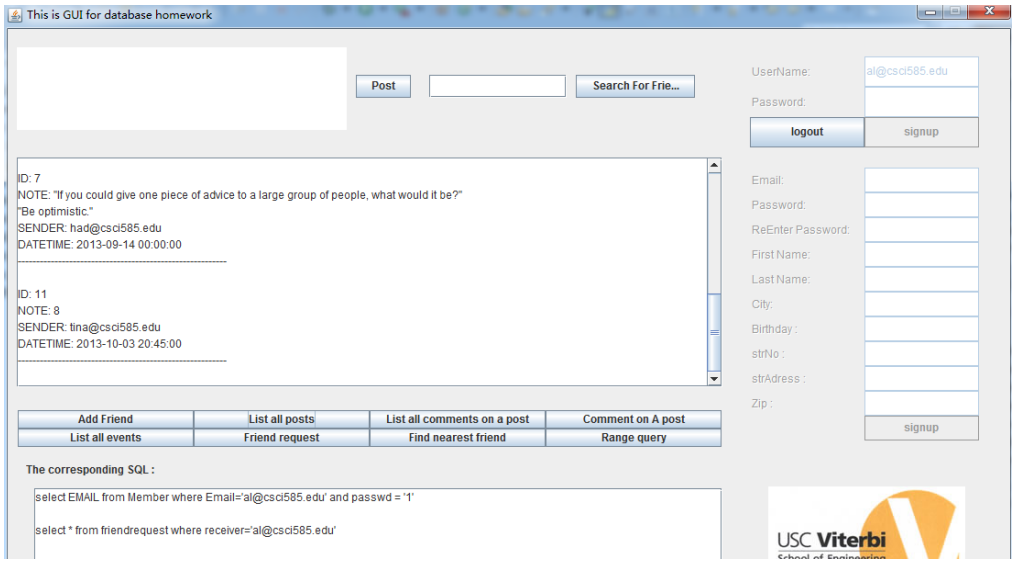
   Once the user obtains information about other users through search (see above), he/she can then add a friend using the friend ID. Next, when the added user logs into the system, there would be a notification icon on the application interface. Once the user clicks on the notification icon, he/she sees the list of all friend requests.

4. *Friend Request*

   Once a user receives friend requests (see above), he/she can either accept all, or decline somerequests before accepting the rest. He/she can decline by identifying a comma separate list of the requesters to be declined, or decline one requestat a time.

5. *List all posts that are visible to you*

   Once the user clicks on the “List all Posts” button, all posts are displayed on the result panel(In the middle of GUI).



6. *List all commentson a post*

   Once user clicks “List Comments" button, a new window will pop out. Once the user inputs post id and clicks OK, all comments on the identified post will be listed.

7. *Comment on a post*

   Once the user clicks on the “Comment” button, a new window will pop out. Once the user inputa post idas well as his/her commenton the post and clicks OK, the new comment will be inserted into database.

8. *Post*

   The user can use the top input text area to submit a post.

9. *List all events*

   Shows all currently available events.

10. *Find nearest friend*
    Each user address maps to a unique set of world coordinates, provided in the data.xlsx input data file (note: newly created user doesn't need to be considered for this part of the assignment). After logging in as a user(not the user you created), user can click on the “Find nearest” button to find his/her nearest friend in terms of distance.
11. *Find friends within a certain range*
    Ifuser clicks on the “Range Query "button, a new window will pop out. Once the user inputsleft top coordinate and bottom right coordinate of a rectangular spatial range and clicks search, allfriends living in this rectangle area should be listed.

Submission Guidelines

1. Compress all your deliverables into one zip file named “Assignment2.zip”; the zip file must only includethe following files: Assignment2.java, createdb.sql, dropdb.sql, and description.pdf. Make sure you Do NOT include the .class files in your zip file.

2. Submit your zip filethrough Blackboard.

Notes:

- You can use any Java Visual softwareyou wish to develop your program, but make sure your program is runnable.
- Start working on your assignment early.
- Submit early. Any submission related issues should be resolved before your submission. Do not try to submit the code right before the due. No late submissions will be accepted.

Tips

- Learn about JDBC here.
- Learn about Oracle Spatial Java APIhere.
- Learn about Oracle Online Documentationhere.
- Learn how to design a GUI in Javahere.
- There arealso lectures on JDBC and Spatial DB posted under "Useful Links" on Blackboard for your reference.