

CSC6710 Course Project

Fall 2015

Description

Consider the design of a social networking site, such as <http://www.patientslikeme.com/>, in which patients are registered with the website to consult each other in a “patients as doctors” fashion. Each patient is identified by a unique patient id, first name, last name, gender, age, email (which is also unique). On a particular day, each patient might have several symptoms (cough, depression, fatigue, etc), but the patient can insert new symptoms as they appear or delete old ones as they disappear. The database needs to keep track of when each symptom appear or disappear for each patient. In addition, each patient might be diagnosed with several conditions (stroke, breast cancer, diabetes, etc). In a similar way, these conditions will appear and disappear as time goes along and the database needs to keep track of when such conditions appear and disappear and for whom. Finally, each patient might receive several treatments (medicine, therapy, etc), the database needs to store the detailed information of the treatment. In addition, a patient can send a message to another patient for help and all the messages communicated among patients need to be stored. Each message is identified by a msgid, a subject, the sender, the receiver, the content of the message, a timestamp of the message.

Part 1

Design the database using the ER approach and then create the tables accordingly. Populate the tables so that each table contains at least 10 tuples. Then using Java and SQL, implement the following functionality:

1. Implement a button called “Initialize DB”. When a user clicks it, your database will be initialized with all tables and tuples; all students should use database name “sampledb” with password “pass1234”.
2. Input a new patient with no symptoms, no conditions, and no treatments. Patient ids should be generated automatically by the system and each patient id should have 3 digits exactly.
3. Search a patient by patient id, and search a patient by an arbitrary “AND” combination of other attributes (first name, last name, gender, age, email).

Some simple GUI interfaces are expected for each functionality.

Part 2

Based on part 1, implement the following functionality using Java and SQL with simple GUI interface. Populate each table with sufficient tuples so that each query below will return some results.

1. Insert/delete/update a patient (all attributes) without symptoms, conditions, and treatments.
2. Insert/delete/update a symptom for a given patient. A timestamp needs to be recorded for such an event.
3. Insert/delete/update a condition for a given patient. A timestamp needs to be recorded for such an event.
4. Insert/delete/update a treatment for a given condition. A timestamp needs to be recorded for such an event.
5. Search patients that have both symptoms “cough” and “fatigue” right now.
6. Search patients that have conditions “diabetes” sometime in the past but have no such conditions now.

7. Given a patient X (user specified), find the patients who sent the most number of message to patient X. If there is a tie among several such patients, then list them all; otherwise, just list one.
8. Find the patients who are under treatment “physical therapy” and who have never sent out any messages.
9. Find the patients who are the second oldest in the database. If there is a tie among several such patients, then list them all; otherwise, just list one.
10. Find a female patient who sent a message to a patient who has the condition “diabetes” now.

How to submit:

All files (source codes, class files, bat, and txt) should be zipped into one file called csc6710_xx_part1 or csc6710_xx_part2 for a student whose last name is xx and send the file (maybe via <https://www.hightail.com/> for big files) to Shiyong@wayne.edu and TA with the subject of “CSC6710 project submission part 1 or 2”. The following files must be contained in the submission:

1. Readme.txt, which lists your id, name, email, as well as the information of your partner. Detailed instructions regarding how to compile and run your program and user’s guide are part of this file.
2. Cc.bat, which is used to compile your program.
3. Run.bat, which is used to run your program. When a grader unzips your file under category, she/he just needs to type run.bat to run your program. Please make sure this requirement is satisfied by testing.

The students will also run the project in front of the TA or instructor for each problem listed above to demonstrate your work.

The project is to be done by a pair of students, but each student’s contribution needs to be clearly stated in readme.txt.

Start your project early, and ask questions if you have doubts. Do not wait until the last minute.

- [Previous example project sample video](#) (Courtesy of Rajiur Rahman)