CmpE 321 Project 3 – Clinic Information System

Due: 15.05.2013 Wednesday, 17:00

In this project you are asked to develop a web based information system which for a private clinic. The minimum requirements of this system is

1. The system will be able to authenticate users with four different profiles namely administrator, operator, medical doctor and patient.
2. Administrator defines basic lookups and resolves problems. Its main role is to oversee smooth operation of the system. Therefore the user of this sort should have access to all web interfaces.
   a. A lookup relation contains lists which do not change frequently. They provide a consistency, efficiency basis for transactional relations. These relations emerge naturally from the normalization process. In the case of a clinic information system, definition of polyclinics (e.g. dermatology, ophthalmology etc.) and Medical Doctors associated with each polyclinic (assume one doctor has a single specialty as in real life so works only for exactly one polyclinic) can be counted as lookups.
   b. All lookups should be entered via the web interface by the system administrator (admin). You can assume that there is a single admin.
   c. Operators are responsible for registering patients to system and/or approving self-registering patients (this part is optional).
3. Patients are registered members of the system. The registration requires Social Security number (in Turkey its Turkish identity number) info, home address, email and phone.
4. A registered patient will be able to arrange an appointment for future days from among the available slots of corresponding MD. It will be wise to design the GUI for the patient to select the polyclinic then the MD of interest to see his/her schedule for a given day. Then select a slot (a slot is a 10-20 min time slice during which the patient will see the MD) and confirms the appointment.
   a. When the appointment is confirmed the system sends a notification mail to the email address provided by the patient. This mail will include a link to update/cancel the appointment.
   b. Patients can not make more than one appointments for a specific polyclinic in a day. That is to say a patient can neither steal more than one slot to see an MD or see other MDs of the same polyclinic in a day.
5. The lists of patients should be available to operators for all departments but an MD can only see his/her list with special options. During the day the doctor click a button named “Get next Patient” and the next person is seen on his screen. If the patient appears the patient will have another button to open a page with
   a. The history of this patient including links to all previous visits (for all polyclinics -this part is read only)
   b. An editable part to write the history, diagnosis and prescription. Kindly note that all data items should be handled separately. The MD will then be able to print i) the prescription and if needed a ii) medical report
6. For performance evaluation admins will be able to see the daily number of patients diagnosed by each MD for a given time span.
7. The passwords are expected to be encrypted in the database. You are encouraged to use stored procedures for all operations.
Tools and Techniques

The project consists of two phases. In the first phase you will design and implement the database for this application. You can use a DBMS system of your choice such as **MySql (encouraged)** or **MS SQL Server**, to implement the database. In the second phase you will design and implement the Web application. For the Web application, you can choose one of the technologies such as **PHP**, **ASP**, ASP. Net, or JSP. The specifications of the two phases are as follows.

**DB design and implementation phase:**
- Find out the information requirements of the Clinic Information System.
- Determine the constraints and domains. Determine the entities and relations.
- Identify the properties of the entities and their domains.
- Determine the identifier of each entity (i.e., find the primary key).
- Draw the Entity-Relationship (E-R) diagram
- Decide what the base relations are.
- Draw the Functional Dependency diagrams.
- Make sure your relations are in BCNF, if possible further normalize them to 4NF.
- Decide what the referential integrity constraints are (identify foreign keys).
- Decide which deletion integrity rules to use (restrict, set to NULL, or cascade).
- Identify user views.
- Considering frequent access, come up with a physical database schema.
- Create the database on a DBMS.
- Specify DDL statements in SQL (internal, external, conceptual level).
- Specify the queries (needed for the transactions you determined in step 1) using SQL.
- Write SQL insertion, deletion, modification and select statements (DML statements).
- Present your work in the DB design report. Submit the hard copy of your report.
- Submit the script (a.k.a. dump or back-up) of the database you have implemented in a CD. CD must also include the web application source codes.

**Web application design and implementation phase:**
- A simple web interface is adequate.
- You are recommended to use the following technologies:
  - PHP
  - Java (JSP, and JDBC) + Apache Tomcat Server (for JSP)
  - Visual Studio .NET ( ASP .NET )

**Some Useful Resources and Links:**
- A **manual** for web database application. We suppose this manual will be very useful for you. You do not need to consider the sections about XML.
- PHP Resources:
  - [http://www.php.net](http://www.php.net)
  - [http://www.w3schools.com/php/default.asp](http://www.w3schools.com/php/default.asp)
  - [www.phpbuilder.com](http://www.phpbuilder.com)
- Java resources:
  - Sun Microsystems: [http://java.sun.com](http://java.sun.com)
- JDBC Driver for Microsoft SQL Server Installation How-to: [http://www.akadia.com/services/sqlsrv jdbc.html](http://www.akadia.com/services/sqlsrv jdbc.html)
- A Complete JDBC Example: [http://www.eas.asu.edu/~cse494db/IonJDBC/JDBCExample.html](http://www.eas.asu.edu/~cse494db/IonJDBC/JDBCExample.html)
- JSP & Servlet Tutorials: [http://www.coreservlets.com](http://www.coreservlets.com)

- SQL resources:
  - [http://www.sqlite.org/sqlite.html](http://www.sqlite.org/sqlite.html)
  - MS SQL Constraints: [http://www.mssqlcity.com/Articles/General/using_constraints.htm](http://www.mssqlcity.com/Articles/General/using_constraints.htm)

- .NET resources:
  - [http://msdn.microsoft.com/tr-tr/netframework](http://msdn.microsoft.com/tr-tr/netframework)