Medical Facility Performance Dashboard Application

CSC 540 – Database Application Programming (Team Project 1)

Due Dates (see end of document)

Version1 : 09/29/2019 Version2: 10/15/2019

Teams of size 4 or 5 (if at least two distance students are included).

Background: We would like to develop an application that provides performance data about medical facilities with respect to the quality of patient visits and patient outcomes. The application will record different kinds of data about patient visits and support different queries and reporting functionalities.

Note 2: For the next two weeks, this document will be in edit mode as I revise to clarify based on your questions.

Description.

Medical facilities have a have a unique facility id (integer), name, classification (primary, secondary and tertiary represented by codes 01, 02, 03), address (number, street name, city, state and country), a number of service departments, capacity (i.e. number of beds), staff, a list of certifications. Each certification has a name, a unique alphanumeric acronym, and date certified and expiration date. Staff, as well as service departments, can be of two kinds: medical and non-medical. All staff have name, unique employeeid, designation (medical or non-medical), hiredate, a primary service department that they work for and possibly multiple secondary service departments. Examples of service departments are General/Internal Medicine, Cardiology, Ob-Gyn, Neurology, Oncology, Emergency Room, Labs, Pharmacy, and so on. Each service department has a name and a unique five character code (alphanumeric) and a director.

Service departments provide different services e.g. Cardiology provides echocardiogram services and Neurology performs MRIs. Each service has a name, a unique alphanumeric code and optionally some equipment (each identified by name-string) necessary for performing the service. Service departments also have specific parts of the body that are their areas of specialty e.g. Cardiology is the heart and Ob-Gyn has uterus, ovaries, fallopian tubes etc. It is possible for a part of the body to be associated to multiple service departments and vice versa. Body parts have name and code (alphanumeric). If service does not have specific body parts e.g. Pharmacy, Labs, Radiology, then the implication is that all body parts are associated.

Patients check-in at hospitals in a process that involves either registering into the system (new patients) or logging in (for existing patients) and then reporting on what symptoms that have prompted the visit. For new patients, the check-in process requires entry of facility id, patient information, first name, last name, date of birth (dob), address (number, street, city, state, country), mobile phone number. On entering storing this information, a patient id is assigned and stored as well, and then prompts to continue with the remainder of the check-in process (entering of symptoms) can proceed. Otherwise if a

patient exists (and is validated by login credentials – last name, dob, and city), then the patient is taken directly to the prompt for entering symptoms. If not validated, you should give a message “incorrect login credentials” and redirect back to the login screen.

Symptoms have a name (string) and unique alphanumeric code beginning with prefix “SYM '' e.g. sym01. Patients enter information about their symptoms by repeatedly selecting a symptom from a numbered list of symptoms, which then generates prompt to enter the metadata for that symptom. One of the menu options on this screen will be “Done” which indicates that a patient is done with symptom data entry. Symptom metadata include: an associated body part; duration i.e. how long this symptom has been going on, which is indicated in number of days (fractional days are allowed); if this is a first occurrence of this symptom or if symptom is recurring (recorded as Boolean), and if the symptoms began following a specific incident e.g. an accident, fall, contact with infected agent, etc., stated in text. Sometimes the body part is implied by symptom and will not be entered by patient e.g. blurriness of vision is associated with the eye and shortness of breath with lungs. On the other hand, for symptom pain, the body part will have to be indicated. Some example symptoms are Fever, Headache, Pain, Tenderness (to the touch), Tightness, Numbness, Lightheadedness, Shortness of breath, Blurred vision. A symptom may also be associated with a severity whose value is drawn from one of possible severity scales. For example, the pain symptom, the severity scale consists of numbers from 1 to 10. On the other hand, a symptom like bleeding has severity defined by a set of three values {Heavy, Normal, Light}. Your application should allow the possibility of entering a new severity scale as well as new symptoms and their associated severity scale.

Once patient login and symptom data entry is completed then patient check-in process is considered as officially started and the time is recorded. Sometime during the check-in phase, a staff member who must be a medical personnel staff takes and records the patient’s vital signals: temperature (integer) and blood pressure (two integers, systolic and diastolic). In other words, the option to enter vitals data for a patient should only be available for medical staff. Once vitals data entry is done, the check-in phase is officially ended and end time is recorded. At this point, the system automatically places a patient on one of three lists based on defined priority assessment rules. Assessment rules are essentially, combinations of symptoms and body parts that imply one of three priority statuses, High, Normal, Quarantine. For example, headache of scale > 7, plus blurred vision, plus lightheadedness is priority high. Similarly, Pain in chest > 7, Tightness and right arm is also high priority. Fever, diarrhea, vomiting and exposure risk (true or false) has an assessment of Quarantine. Your application should allow entering of new rules. Only relevant medical personnel can process a patient on a list. By relevant, we mean a medical staff that works for a service department where the associated body parts overlap with patients symptoms’ body parts. If an unauthorized person tries to access the list, a message should pop up saying something like “inadequate privileges” for this task. Once a patient is processed on a list (successful selection of patient from list), patient is now in a treatment phase so is removed from list and the time for this recorded also recorded.

The final task is to enter information about the patient experience and outcome. This must be done by someone that is a medical staff. There are multiple components to the outcome report:

1. Discharge Status: This allows one of three options to be entered (Treated Successfully, Deceased, Referred). This field is required to be nonempty.
2. Referral Status: This field is valid only if Discharge Status is Referred. The components of this field include the facility to which patient was referred (or a special value 0 indicating patient was not referred to a specific facility), Referrer (employee id), a list of Referral Reason descriptions. In other words, there might be multiple reasons associated with a referral (up to 4 will be allowed). Each reason description itself comprises three components: a Reason Code (1, 2 or 3); Name of Service (selected from list of services or Other if not in list); Description (additional textual description). The Reason codes: 1 - service unavailable at time of visit; 2 – service not present at facility; 3 - non payment.
3. Treatment given: free text description (required)
4. Negative Experience: A list of negative experiences each with one of two codes and a text description. The codes are 1.Misdiagnosis, 2. Patient acquired an infection during hospital stay. (optional)

Each component of the outcome report can be completed separately. The application should basically have a list of 4 menu options and e.g. Enter Discharge Status and then enter the relevant fields. There should be a Submit option. However, note that submission should be denied if required fields have not been completed. The submit process should present all the information for confirmation. If confirmed, the outcome report is then saved along with time. The final step is to present the outcome the outcome report to the patient for their acknowledgement (two options for “YES” or “No” should suffice. For the No option, an additional text field for text description should be present.

Patients cannot check in at the same facility before checking out, and their activities in different facilities do not interfere with each other.

Application Requirements

In general, your application should support only role-authorized actions. This will largely be accomplished through the combination of menus that present only appropriate actions for the given role and context, and advanced features like Views, Procedures or Triggers and Authorization. Additional specification about menu format is provided in the application flow document. This will allow a consistent menu format that will make grading the projects easier. It is expected that your application should handle errors elegantly and not reset on every simple error. For e.g. the application should prompt for another input in case the user input was invalid. Providing user-friendly messages to users when actions are invalid will also be expected.

We expect a command line interface for the application and do not expect a graphical user interface. Developing the latter is strictly optional and no extra credit will be provided for doing so.

You will observe that this application uses predefined values extensively. For the most part we will be using alphanumeric codes for these values (although we also store the text names). We will assume a simple scheme for coming up with codes e.g. first letter of the values prepended with first letters of attribute. More on this soon.

During data entry, generally attribute codes are used were applicable. However, in displaying back to users, application should always append the textual description of codes for

Application Flow

This part of description gives a general idea of what the application should be like, which menu items are available for a particular role, the kind of queries to run/actions to perform for a particular menu item, and the kind of output to expect. The application entry point should be an account creation screen (for patients only) or login, if an account already exists, for patients and employees (i.e. you do not need to provide option to enter employee information, it will be done manually not through interface). After log in, appropriate menu options should be presented depending on whether patient or staff member. A more complete application flow description will be made available as a separate document.

Sample Queries

Queries on your database will be helpful for assessing the quality of database design. However, it isn't possible to leave that to only demo day. Consequently, you will need to implement some queries as part of your project. The list of sample queries will be given shortly. However, here are some of the required queries that must be supported by your application.

Required Queries (tentative)

The following shows examples of query classes and queries that should be supported by your application:

Retrieval Queries

1. Find all patients that were discharged but had negative experiences at any facility, list their names, facility, check-in date, discharge date and negative experiences
2. Find facilities that did not have a negative experience for a specific period (to be given).
3. For each facility, find the facility that is sends the most referrals to.
4. Find facilities that had no negative experience for patients with cardiac symptoms
5. Find the facility with the most number of negative experiences (overall i.e. of either kind)
6. Find each facility, list the patient encounters with the top five longest check-in phases (i.e. time from begin check-in to when treatment phase begins (list the name of patient, date, facility, duration and list of symptoms).

Project Deliverables

Milestone 0 - Questions on the forum due October 13th

Post any questions for clarification of the project description on the forum.

Milestone 1 - Report, due October 20th

For this milestone, you should:

1. Fill in the form for deciding team members as soon as possible.
2. An ER-Diagram along with a list of Entity and Relationship Types that you identify in the project description. For each relationship type, you should state the arity of the relationship e.g. if it is binary, ternary, etc. Relationships should include any hierarchical relationships (subtypes) that you identify. There is no need for verbose text, just a categorized listing of these is fine.
3. Relational Model: A list of tables, 2 - 3 sentences description of each including what constraints (including referential constraints) it encodes, a listing of functional dependencies, a discussion of normal form choices faced and justification for the decision made.
4. For this report, you should list any application constraints (key, participation, other domain constraints) that you identify in the description. A statement acknowledging that you have asked all questions you need to clarify any ambiguities in the description.

Milestone 2 - Final Report (due Nov 10th)

Each team will have to book a timeslot to demo the application to the Professor or Teaching Assistants. The details about the demo will be conveyed later. The following need to be submitted.

1. Constraints: A description of constraints that were not implemented as part of table definitions and why and how they were implemented in the final design. In particular, a separate subsection here should highlight constraints that couldn't be implemented in the database at all and had to be implemented in application code. Note that a key part of assessing your design is how well you used the DBMS to implement constraints v/s implementing in application code.

Two SQL files: First one should contain SQL for triggers, tables, constraints, procedure. The second file should contain queries for populating the tables with the sample data. Sample data will be provided closer to demo date.

2. Executable file (e.g. - Executable JAR file) and source Java Code.
3. README.txt - This should contain the names of the members and any additional instructions that you think will be necessary to compile and execute your code.
4. Peer review. A link will be provided to submit the review when the milestone will be due.

Milestone 3 – Project Demo (sometime during Nov. 17th week)

Grading

Deliverable	Milestone	Percentage	Deadline
Report	Milestone 0	5	October 13th 2019
ER Diagram + Report with table definitions, constraints, etc	Milestone 1	20	October 8, 2019
Final Report	Milestone 2	45	November 10th, 2019
Peer Review	Milestone 2*	% of Average of peer review grades	November 10, 2019
Demo	Milestone 3	30	November 17th (week)

Getting Started

I would recommend using an IDE(like Eclipse or IntelliJ) for developing the project and some form of version control like GitHub for sharing project amongst team members. (You can create private repositories on NCSU Github). Using Oracle's SQL-Developer would also help you in writing long queries, triggers and procedures because of advanced features like debugging and static analysis of query.

The following link will help you to connect to the database using the JDBC Driver: Creating a connection using JDBC. Students are encouraged to read more about proper handling of connection and JDBC Best Practices.

Note that points will be deducted for **improper handling** of connection in the java application.