PROJECT: A healthcare system (this is a group project).

1. PROJECT DESCRIPTION

1) Learning Objectives

The purpose of this project is to have you apply the knowledge you learn in this course to the design and implementation of a database-driven application. Through this project, you should gain familiarity with:

- the process of database design and implementation
- different techniques in interacting with DBMSs in programming languages
- specifying queries in SQL DDL and DML

2) Description

In this project, you will develop a health care system for a clinic. The clinic has administrators, doctors, nurses and patients, who have personal information, such as id (unique within their groups, for instance, a doctor has a unique doctor id, a patient has a unique patient id, if a doctor becomes a patient, a unique patient id is assigned to him/her), last name, first name, date of birth, mailing address (please break down into smaller components), and contact phone number, recorded in the database system. Each doctor may also several specialties stored in the DB.

Anyone who wants to see a doctor at the clinic must make an appointment first. If she is new, a nurse first creates a patient record. The nurse helps the patient to set up an appointment with a doctor on a specific date and time. The reasons for the appointment should also be specified. The system doesn't store which user helped with registering the patient. Note that patient's personal information should also editable by a nurse. A patient can also be flagged as active/inactive. Inactive patients cannot use the system until his/her status is turned active. **Double-booking should be disallowed**, i.e., the same doctor should not be booked for different patients at the same time on the same day.

When the patient comes in on the appointment date, a nurse (who may not be the person who registers the patient) does routine checks (blood pressure including systolic and diastolic reading, body temperature, pulse) and asks him about his symptoms and records the information in the system. The doctor then comes in and tries to make diagnoses. Sh/e may have to order some lab tests before diagnoses can be made. In that case, the initial diagnoses are recorded first. And the final diagnoses will be appended after the test results are out. When ordering tests, one can select which tests among a list of standard tests to order, the program should show all the tests that are ordered for the visit so that it's clear to the user. Before the order is submitted, one can remove from or add tests to the list.

The lab tests may or may not be performed on the same day of the visit. The test result typically will not be available until several days later. In any case, the nurse has to record the datetime when the tests are performed, the test results and the diagnoses made by the doctor based on the test results (in case diagnoses were not made right away at the time of visit) for that particular visit in the system.

In all the cases, the nurse does the bookkeeping work in the system (for instance, for a patient visit, recording which doctor performed the checkup and made what diagnoses, ordered what tests, the test result, etc.). To simplify the system, we omit the features that can be performed by doctors and patients.

The typical tests include white blood cell (WBC), Low Density Lipoproteins (LDL), hepatitis A test, hepatitis B test, and so on. Each test has a test code (unique), name(unique), low value (may be null, or a decimal value(9,2)), high value (may be null, or a decimal value(9,2)), and name of unit measurement (for instance, mIU/L is an abbreviation for milli-international units per litre, we will record mIU/L in the DB as the name of unit measurement). The test result can be anything (a numeric value, or a description). And the DB should also store if the test result is abnormal or not. If the low and high values are null, the normality will be entered manually. If either is not null, its value is based on the low and high value (the normal range is [low, high]).

The nurses must authenticate themselves before accessing the system. Once logged in, the program should show the username as well as first and last name of the person who is logged in. If it's a nurse, this nurse will be the person who performs the tasks, therefore, in the UI it should not let the user choose a nurse again.

Other than the tasks mentioned above that a nurse can perform, a nurse can also search for a patient's personal information and visit information (e.g. diagnoses, tests performed and results, etc.) by his date of birth, or name (last name and first name), or combination of both.

Once an appointment is found for a patient, the nurse can

- view the appointment information,
- edit the details of the appointment if it is not the appointment date yet.

Once the visit information is found for a patient, the nurse can

- view the visit information (e.g. routine check results and any tests ordered and the results if available)
- edit the visit by entering test results and final diagnosis. Once the final diagnosis is entered, visit information (e.g. test results, etc.) cannot be changed.

In addition, there should be an interface provided for administers, who should also authenticate themselves before accessing the system. The administrator can pose queries in SQL and view the result in tabular format. Furthermore, the administrator should be able to select any two dates and view a report containing the information about all the visits during the time specified. The visit information should include the visit date, patient id, name, the names of the doctor and nurse involved, the tests ordered, test perform date, tests result abnormality, and diagnosis. The result should be sorted by the visit date followed by the last name of the patient.

Please understand the initial specification described above may not be as complete as it should be, which reflects the nature of real application development. **One of your tasks here is to talk to your customer, me, to unveil the details that are yet to be discovered.** Please contact me if you have any questions about the requirements. The description may be elaborated based on our discussions as you, the developer and I, the customer, reach an agreement.

2. PROJECT REQUIREMENTS

You will design and implement a database application that shall accomplish the functionality described above. The database system you will be using is **MySQL DBMS** on a department server. Details on how to connect to the server will be provided later. You can choose **any programming language** that you are most comfortable with to implement the GUI. **Note that the application is not web-based.**

There will be several iterations during the development of the application. The requirements and due dates of them will be posted on Moodle as the project progresses. **Note that** the iterations may not be the same as what you have experienced in your Software Engineering class. They are only meant for keeping you on track. Finishing each one doesn't guarantee a finished system. You **need and have to** make sure that you and your partner(s) will work together to implement the system as required and **finish it on time**.

1) At the end of the semester, each group needs to submit a soft copy of all the code and make a demonstration of the project.

2) Each student will be individually evaluated on the project throughout the semester

   a. At the end of each iteration, **each student will evaluate herself/himself and the other group member(s) for contribution to that iteration.** Each member will get his/her deserved share of grade for that iteration

   b. After the final project demonstration, each student will perform self and group member evaluation. Each member will get his/her deserved share of grade for the project demo (e.g. the members of a A project, may get A, or B or C or worse on the project depending on each person's contribution).