

CS174a COURSE PROJECT Part 3

System design and implementation

1 Overall System Architecture and Design

Recall the Health Information System you designed in part 2. Another system called the Health Messages Exchange interacts with the Health Information System. The Health Messages Exchange is a sql database which contains a universal table with all the data and last accessed datetime. In this project you will take a programmatic approach to access the Health Messages Exchange, read the data and store them in your Health Information System database. i.e. All of the data is present in HealthMessagesExchange; You will need to import them to your database. Java is recommended for this programming task.

The HealthInformationSystem's software system provides three interfaces "Patient", "Doctor", and "Administrator". The three interfaces are meant for the interactive users - one for patients, one for doctors and one for administrators who will perform interesting queries on the data. We now briefly discuss the components that you will design.

2 Health Messages Exchange

The system manages the inventory of messages of all the patients. It allows the IT engineer to download health messages based on the control id given by the patient. You will be provided with a sql database called HealthMessagesExchange. First you will need to import the database into your mysql environment by executing the healthmessagesexchange.sql file. Then using Java JDBC connectors, read the data from the healthmessagesexchange table and identify each field of the message. After identifying and separating you need to store the data into the schema you created. While accessing the HealthMessagesExchange table's records, update the last accessed time field of each record with the current system date and time. Simultaneously, the patient table's xmlCreationDatetime needs to be updated with the current date and time. This is a transaction which ensures that the data in your database is consistent with the Health Messages Exchange data. If these two date time fields differ, the data in HealthMessagesExchange will receive priority as the valid version.

3 Health Information System

3.1 Patient Interface

The patient interface allows patients to view or edit details about themselves. For the sake of simplicity, you need to be able to edit patient table and guardian table with the patient privilege.

3.2 Doctor/Author Interface

The doctor interface allows the doctor or author to view or edit details of any patient. For the sake of simplicity, you need to be able to edit the plan and allergies data.

3.3 Administrator Interface

The administrator interface allows the administrator to view interactively the relationship between various items in the database. From this interface you should be able to

- View number of patients for each type of allergy (substance).
- List the patients who have more than one allergy.
- List the patients who have a **plan** for surgery today.
- Identify authors with more than one patient.

4 Requirements

It is desirable but not compulsory for your system to have the user interface. i.e. Your interfaces need not be accessible from the web browser. You must store all data of your systems in the databases managed by the MySQL DBMS. This means that when your systems are not running, all data are in the databases and nothing would be stored anywhere else including files. (During the demo, your systems may be asked to shut down and restart.) All previous actions done on your system must be remembered. (Again, you should not use files.) Your system should be implemented in Java using JDBC to connect to your database managed by the MySQL DBMS. To sum up, you need to

1. Import the sql data in the universal patient table into your mysql environment.
2. Java provides the ability to connect to a mysql database and execute queries from within the java development environment. For example, if you want to retrieve the record of a patient you would do a 'select * from patients where patientid = x' in SQL. The same thing can be done in java. You need to find out how. The results will be returned to java. Now

create a class object in java for patient and store these results. Then perform a write to the database you have designed.

3. Re-curse through the SQL database and for each row of data returned, store in class object and writeback to the database you have designed. Proceed this way till you have ingested all the data in the universal data table and written to your database.
4. Now add read/write functionality to your database. There are three interfaces - patient interface can read all of his/her data, but edit only the 'Patient' and 'Guardian' tables. Doctor interface can read any patient's data but edit only the Plan and Allergy information.
5. For displaying these interfaces - doctor/patient/administrator, you need to either create a website or a window frame using one of the java frameworks like spring. At the minimum, you could just show these data in console. The console is less interactive compared to a window frame, so this means you need to make it interactive by questions to the user like 'Enter priviledge level - patient/doctor/administrator', 'Enter the table to be displayed', 'Enter the table to be edited' and so on.

4.1 Deadline

At the end of your project a demonstration of your systems operations has to be presented to the teaching staff of this course. The project deadline is at your scheduled demo. Regular demo will be arranged during the last week of instruction. The details about the demos will be announced at a later time.

4.2 Project Report

At the completion of your project each team should submit a project report. The report has to address the issues considered in making major decisions. The reports should be based on and a revision of your early reports. List the task division between team members. Show the final database schemas and all integrity constraints. Discuss each change you have made to the database schema after the early report. Describe brievely how you deal with a violation of each of the identified integrity constraints. Show a picture of the three interfaces and a brief description about them. List all JAVA classes, and for each class, list all methods from other classes that are invoked by this class, and all operations from Part 2 Data Requirements and Operations that use this class.

4.3 Demo and What to turn in

At the demo time you will turn in

- A final project report
- Schema definitions

- all SQL queries used in your program
- a listing of all Java source code
- In addition, you will give a half hour demo of your systems. We will provide a sample dataset. This data should be entered into your database before your demo.