

# CS 480 Database Systems Project

## Spring 2021

Projects can be done in teams of **at most three students**.

### Project Description

Everything is affected by the pandemic, including our education. Vaccination is an important step towards overcoming this difficult situation and conducting it efficiently is crucial for getting back to normal routines of life, including in-person teaching. In this project, you will create a database system for UI Health (UIH) hospital to carry out the Covid-19 vaccination and scheduling it.

You should follow the two-tier structure for your design: you should design and develop a database system (back-end) and create a simple application (front-end) that interacts with the database, providing the functionalities.

- **Vaccine:** The core to the vaccination process is vaccine that has a name, name of the company, number of doses required for immunization (e.g., Pfizer requires two doses), and an optional textual description.
- UIH can receive vaccines of different types. For example, it may currently have 1100 Moderna and 2050 Pfizer vaccine in its repository.
- **Nurse:** The vaccination is carried out by nurses. Every nurse has a name (Fname, MI, Lname), EmployeeID, age, gender, phone#, and address.
- The vaccination is done during the 1-hr time slots (M, Feb. 15 2021, 10:00-11:00 am). UIH has a maximum capacity (100) on the number of patients it can accept for vaccination at a time slot.
- UIH needs to have at least one nurse per time slot. Every nurse can vaccinate at most 10 patients during each time slot. As a result, the maximum number of patients that can be vaccinated at time-slot  $j$  is:

$$\min(10 * N_j, 100)$$

where  $N_j$  is the number of nurses at time-slot  $j$ .

- **Patient:** Every patient has name (Fname, MI, Lname), SSN, age, gender, race, occupation class (e.g. educator, healthcare worker, etc.), medical history description (for simplification, suppose it is plain text), phone#, address.
- **Vaccine Delivery:** Upon delivery of a vaccine to UIH, their repository gets updated (e.g. if it currently has 1100 doses of Moderna and receives 1000, their repository gets updated to 2100 for Moderna).
- **Nurse Scheduling:** In order to schedule vaccination, UIH requires to schedule nurses for different timelines. For every time-slot UIH can schedule up to 12 nurses per time-slot.
- **Vaccination Scheduling:** Patients can schedule appointment, so long as there is an available time-slot. The Schedule includes the Vaccine that the patient will receive. An schedule puts one dose of a vaccine on-hold. A schedule can be set up if there is at least one dose of the vaccine in the repository that is not on-hold. (Hint: You can have a table VACCINE with one attribute

“availability” that shows the number of doses available per vaccine, and one attribute “on-hold” showing the number of doses on-hold)

- The patients who have received the first dose of vaccine, can only schedule the 2<sup>nd</sup> dose of the same vaccine. Therefore, they can schedule a time only if at least one dose of that vaccine is not on-hold – i.e., availability-onhold>0.
- **Vaccination Record:** Every vaccination for a patient is done by a nurse at a specific time. The vaccination record also includes the which vaccine has been used and the dose of vaccine (e.g. PatientX received dose 1 of Pfizer, by NurseY at time-slot j). Upon Vaccination, one vaccine that has been on-hold gets reduced from the repository.
- **(Bonus) Vaccination Eligibility:** Not everyone is eligible to schedule vaccination. Upon scheduling your system checks the eligibility of a patient and only allows scheduling if they are eligible.
- **(Bonus) Use your imagination:** Add features to your system, using your imagination to make it more realistic.

## Functionalities:

- Login: You have three types of users: one admin, nurses, patients. In the login page users should enter their username/password and select their type (admin, nurse, patient)
- Different users will have different functionalities:
  - o Admin:
    1. *Register a nurse:* nurses cannot self-register. The admin should register them. In addition to the information above, every nurse is assigned a username and a password.
    2. *Update nurse info:* any update in nurse info (other than phone# and address) should be carried out by the admin.
    3. *Delete a nurse:* remove a nurse from database.
    4. *Add Vaccine:* upon receiving new doses of a vaccine, the admin updates the repository.
    5. *Update Vaccine:* in any case some vaccine (not on-hold) are removed from the repository, admin updates the number of vaccines.
    6. *View Nurse info:* view the information of a nurse and the times they have scheduled for.
    7. *View Patient info:* view the information of a patient, the times they have scheduled for vaccination, and their vaccination history.
  - o Nurse:
    1. *Update information:* Nurses can update their address and phone#
    2. *Schedule time:* nurses can schedule for time slots that have less than 12 nurses scheduled for them.
    3. *Cancel a time:* nurses should be able to delete a time they have scheduled for.
    4. *View Information:* Nurses can view their information, including the times they have scheduled for.
    5. *Vaccination:* upon delivering a vaccine, nurses should record the vaccination
  - o Patient:
    1. *Register:* Patients can register their information. In addition to what described above, a patient needs to pick a username and a password.

2. *Update Info*: patients can update their information.
3. *Schedule a vaccination time*: Patients should see the available time slot and be able to select one as their schedule.
4. *Cancel Schedule*: Patients can delete their scheduled time (which will also release one on-hold vaccine).
5. *View information*: Patients can see their information, the times they have scheduled for vaccination, and their vaccination history.

## Implementation details:

- Your host application should have a GUI interface. You may implement either a native GUI application or a web interface (while a web interface is preferred, you may not lose points for implementing a native GUI).
- Your application GUI may be programmed with Java, Python, JavaScript, PHP, or C++.
- You may use a web programming framework like Django (Python), Laravel (PHP), Spring (Java), or Node.js (JavaScript), etc.
- You may use any front-end frameworks like React, Angular, Vue.js, jQuery, etc.
- You may use an ORM (object-relational mapping) framework, like Hibernate, provided that your SQL schema is compatible with the requirements.
- You may not use a platform-specific language, e.g., Objective-C, Swift (Apple), or C#, F# (Microsoft).
- Approved SQL databases are MySQL, PostgreSQL, SQLite or MS SQL Server.
- If you would like to use any other language or framework not explicitly listed above, you must obtain prior approval from the TA that he is able to effectively evaluate the language of your submission.

## Submission:

- Milestone0: Team formation. Deadline: Feb. 17 by midnight. Form a team, using the following google doc (you should login using your UIC email):  
<https://docs.google.com/spreadsheets/d/1n9f5ztBXqu6vAQzju4HFwHaaeQFnWSXwma2t15Jqy-s/edit#gid=0>
- Milestone1: Submit a documentation on BB, containing (a) the conceptual schema in form of an ERD and (b) relational schema. Deadline: March 3 2021 (by mid-night CST)
- Milestone2: For the final phase of the project, you may either be asked to demo your project on BB Collaborate or make a video presenting your work's functionalities. The deadline is the last week of classes. Further information will be provided in later announcements.