CS 166: Phase 3 of Project Description

In this phase, you are given scripts (/code/sql/) to create a database schema that implements a slightly simpler version of the ER diagram given to you in phase 2 of the project. Additionally, you are given a set of dummy data (/code/data/), scripts to load the data in DB (/code/sql/), and some Java code (/code/java/src/) that can be used to build your client application.

Your client application must implement the following functionality using the given schema:

1) Add Doctor: Ask the user for details of a Doctor and add it to the database.
2) Add Patient: Ask the user for details of a Patient and add it to the database.
3) Add Appointment: Ask the user for details of an Appointment and add it to the database.
4) Make an appointment: Given a patient, a doctor and an appointment of the doctor that s/he wants to take, determine the status of the appointment (Waitlisted/Available/ Active/Past) and add the appointment to the database with appropriate status.
5) List appointments of a given doctor: Given a doctor ID and a date range, find the list of active and available appointments of the doctor.
6) List all available appointments of a given department: Given a department name and a specific date, find the list of **available** appointments of the department.
7) List total number of different types of appointments per doctor in descending order: Return the list of doctors, the status and number of appointments in a decreasing order of number of different types of appointments that each doctor has.
8) Find total number of patients per doctor with a given status: Given an appointment status, return the number of patients per doctor with the given status.

NB: For each query, return all relevant attributes for the corresponding entity unless the attributes are specified.

Try to handle any exceptions that arise during the regular operation of your application. **A final report** (a .pdf file) about your system including the screenshots of the outputs of all queries, assumptions, comments and contributions, and **the entire source code** have to be submitted **on iLearn** within the due date. You must demonstrate your code to the TA according to the schedule (provided later).

Bonus credit: Groups that implement systems with user-friendly interfaces, add extra functionalities (e.g., the queries that are mentioned earlier, but omitted here) will receive extra credit. Any additional functionality must be explicitly described in a README file (inside the zip file) and pointed out to the TA during demo to receive extra credit. The relevant (GUIs, outputs) screenshots must be included in the final report.

Submission instructions:

- You should submit the final report and the final source code **in a zip file on iLearn**. The deadline is on Friday June 11th at 11:59 PM to upload your projects on iLearn. Make sure that the zip file can be uncompressed without any errors.
- Please note, you must demo your code to the TA either on June 12th or 13th. A spreadsheet will be shared closer to the deadline for the demo. Both of the team members should be present during the demo.
- **Rubric:**
    1. 50 points completeness and correctness (including SQL queries and indexes),
    2. 10 points for error handling (for example, handle incorrect data input), and
    3. 10 points for documentation (readme file, if any) and a final report (including screenshots, assumptions, comments, and contributions).