GT COVID-19 Testing System

CS 4400: Introduction to Database Systems

Fall 2020: Semester Project

Project Purpose

In this project you will analyze, specify, design, implement, document, and demonstrate an online system. You are required to use the classical methodology for relational database development. The system will be implemented using a relational DBMS that supports standard SQL queries. You will use your localhost MySQL Server (Version 5.1 or above) to implement your database and the application. You also cannot use any other software like Access or SQLite. Ask the professors or TAs if you have questions.

Project Phases

Inputs (we give you)

• Text and UI description	• Advanced ERD • Raw initial data	• Database schema • Initialized database • Procedure shell	
Phase I	**Phase II**	**Phase III**	**Phase IV**
• Entity Relationship Diagram • Logical constraints	• Relational schema • Database schema • Initialized database	• Implemented procedures	• Fully functional application

Outputs (you turn in)

Important Dates

Mon, Nov 23 - Tue, Nov 24	Early Phase 4 Demos (Optional)*
Tuesday, Nov 24, 11:59PM EST	Phase 4 Assignment Due
Sunday, Nov 29, 11:59PM EST	NO PENALTY Grace Period for Phase 4

-- No code changes allowed beyond this point. --

Mon, Nov 30 – Fri, Dec 4	Phase 4 Demos

**See Timeline section for clarification*

Directions for Phase 4

In this phase, your team will implement a full-fledged, stand-alone application for the GT COVID-19 Testing System as described in the project description.

Phase 4 vs Final Exam

Phase 4 is completely optional: you can either take the Final Exam, or do Phase 4, but not both. If you do Phase 4, then all members of your team must be available for the live demo. If you decide to go with the Phase 4 option, you aren't "locked in" to that choice until you submit the assignment. In other words, if you decide five minutes before Phase 4 is due that you all want to take the Final Exam instead, then you're welcome to make that "change" with absolutely no penalties.

The biggest "challenge" that we've seen over the years is when one or more members of a team decides, very close to the Due Date (e.g., less than 24 hours in some cases), that they no longer want to do Phase 4, and won't complete their portions of the project, which leaves the remainder of the team in a very bad predicament. We encourage everyone to be very honest about how much you're committed to completing Phase 4. And thinking proactively, we encourage you to setup a realistic development schedule with checkpoints established well before the Phase 4 Due Date deadline to ensure that all team members are on track with their planned implementation efforts and results.

Timeline

We expect most students would rather have the extra time over Thanksgiving break to work on this then trying to manage it with other tests and projects on the final instructional days, which is why we have extended the timeline from the original Syllabus.

If you prefer to be done with the class before Thanksgiving break, we will have options for demos available on the Mon/Tue before break. The tradeoff is that you must be code-complete before your demo and you must make your Phase 4-vs-final decision before the demo (whereas otherwise, you have until Sunday, Nov 29th for both these things).

Demo Instructions

- The team will join the BlueJeans link **5 minutes prior** to the start of the demo with **the application running, the database pulled up, and the database seeded with the official initial data.** The team will provide this BlueJeans link when signing up for a demo appointment.
- The team should record the entire BlueJeans session and should submit a link to the recording in the “Phase 4 Recording” assignment after BlueJeans has finished processing the recording.
- Have all team members in attendance on time. **No credit will be given to absent members, and 15 points will be deducted from tardy (up to 10 minutes) members**.
- The TA will go through a script of user stories and ask you to demonstrate a comprehensive set of application functionalities

- The TA may ask questions to assess your understanding of the application as well as your participation within the team
- The TA may ask to see your database to ensure changes are persisted there
- The TA won't run your application on their personal computer. A team member (or multiple) will run the application on their computer and screenshare.
- The TA won't try to break your application via SQL injections or some nefarious edge case. However, anything that's listed or depicted in the description is fair game.
- Remember to be respectful of the TA. They are trying to assess your application in a fair and consistent way. They are also in the middle of their own final exams and projects. Be kind to them, and they'll be kind to you.
- You will have **exactly 45 minutes** to complete your demo. We cannot give you more time, so you must come prepared.
- You will not receive your grade directly after the demo. Don't ask for it, as the TA is not allowed to tell you.

Restrictions

- You must use a database. It does not have to be MySQL, but you must use some database to persist data that is not just an in-memory data structure.
- Your code should not be public and should only be shared with your team.
- Your screens should generally follow what we've shown in the description, but you are free to present the UI however you see fit as long as the functionality is met

During Demo Repairs

As mentioned above, you will have a **up to 45 minutes** to complete all of the steps. If you encounter any problems during the demonstration process where your queries (or application capabilities) are not working correctly, then we will offer you the opportunity to perform minor "on-the-spot" repairs.

You should weigh this offer very carefully:

- If you choose to "make some repairs", then the clock will continue to tick during your efforts, and you are still responsible for completing as much of the testing script as possible. Steps from the testing script that are left uncompleted will count against your final score.
- If you choose to accept/ignore the errors and continue with the script, then you will likely lose some points because of the errors. On the other hand, this might still result in a better overall score than stopping to make repairs.

Ultimately, this choice is your call to make as a team. The TAs are allowed to let you know where you are in the testing script (e.g. "You've completed 9 of the 15 steps so far..."), and can give you some very general sense of how severe the error is compared to the expected result, but they will not troubleshoot the error for you, nor will they determine the likely impact of the error on the remaining steps of the testing script. We recommend that you discuss this as a

team before the demonstration, so that you have a general strategy planned in advance - time is precious during the demo.

Note that we do expect the demo script to take a majority of the demo time, so do not submit code on Sunday night intending to make fixes during the demo. These "on-the-spot" repairs are mainly for minor fixes you don't find out about until the demo.

Demo Script Sample

The below sample is provided to give you a sense of what sorts of tasks the TA will ask you to perform as well as how you will be earning points (point values are hidden below). Note the full demo script is a comprehensive walkthrough of your entire application; below is just a sample.

Go to the Create Appointment screen. Try making a new appointment with these parameters: [Failed Appointment].

- **+X** for failing to make appointment with invalid params.

Make an appointment with these parameters: [Actual Appointment].

- **+X** for successfully creating the appointment.

Make another appointment with those same parameters. [Actual Appointment]

- **+X** for failing to make duplicate appointment

Go to the View Appointments screen and check to see if your new appointment is there.

- **+X** for updated View Appointments screen

Submission Instructions

1. You should submit a zip file including the following:
    a. All code required to setup and run your application
    b. A readme including:
        i. Instructions to setup your app
        ii. Instructions to run your app
        iii. **Brief** explanation of what technologies you used and how you accomplished your application (don't spend too much time on this)
        iv. Explanation of how work was distributed among the team members
2. To be clear, **your grade is almost entirely based on your demo**. The submission serves to ensure you are code complete by the deadline and also serves as a deliverable for your efforts.
3. If you have a Phase 4 submission and then change your mind to take the final before the deadline, you should overwrite your submission with an empty submission.