CMPSC 431W – Semester Project Specifications

In this project, you choose your own domain (the thing that the project is about); this should be something that you know something about. For example: race cars, poker tournaments, vegetarian cooking, etc. You need to design, implement, test, and demonstrate that your program (application) runs properly on the VM assigned to you this semester.

NOTE: You can also pick the domain from your midterm exam. If you do this, keep in mind that you still need to do all of the requirements below. The purpose of the extra points of credit for using the domain from your exam is to give you the chance to re-think your data model and hopefully realize how your exam answer could have been better. If you choose to use the domain from your exam, be sure to indicate this at the very top and prominently in your submission. (Extra points for your midterm are *up to* 25 points and are capped at you obtaining a 100% grade on the midterm.)

As part of the project, you will implement an HTTP interface to your database using PHP. (You may use Python if you wish, but no support for doing this will be available.) An example in PHP will be provided to you to get you started. In the end, you will be demonstrating the use of your product and I or a TA will be testing your product.

Here are the details:

- Your project needs to have **at least** five tables (this is the minimum; you may need more for your actual application). You will be supplying a data model and it must be in 3rd normal form.
- You will need to define keys and indexes as though your product was in use for tens of thousands of instances/records. You only need to demonstrate the product with a small subset of realistic data. You will need to obtain or create your own data, and there should be at least one table with at least 200 records in it.
- There has to be at least one query which joins at least four of the tables **in a useful way**. (A "non useful way" would be if you just added two otherwise useless tables to a query just to say it joined four tables.) This query might be a report from your product (e.g., a list of all the food items, amounts, and sources needed in a vegetarian recipe).
- Your application must include at least the following:
  - a way of inserting data
  - a way of deleting data
  - a way of updating data
  - a report (see below *) – this must include sorting
  - For example (restaurant app):
    - Create an employee login account, add (modify, delete) an employee to the schedule, print out a shift schedule for the week, add an item to the menu, change the price of an item on the menu, delete an item from the menu, print out a full menu, print out an allergen menu (for people with

allergies), enter the food orders from each table, calculate the bill for a table.

- Your application must include properly designed and implemented transactions including at least one *justifiable* rollback. (For example, the customer decided not to order a meal.)

Please note that this assignment is relatively free-form, leaving you the opportunity to think and implement creatively. Since this project is so open, *there should be no substantive duplication of ideas or code across projects*, *unless you use the domain from your exam*. Data models and code are like fingerprints – don't use anyone else's work. Any such duplication will be investigated for academic integrity violations. In addition, please keep in mind that "borrowing" your project from the internet is grounds for obtaining an "F" and being referred to the academic integrity committee. To be clear, you are certainly allowed to look up syntax, etc. but you are not allowed to use someone else's completed or semi-completed project. In this regard, note that I reserve the right to investigate and to search for such issues. ***In short, do the work yourself, and everything will be fine! Do not install any "frameworks" or other such code.***

*What I mean by "a report" – use the following images as examples.

What you need to hand in:

- Project design document (see outline below).
- Project code (all PHP/PY files). Do not ZIP these files. If Canvas will not let you upload the documents, then rename them such as update_screen.php**.txt**.
- A *link* to a 5 minute video demo (you can use your PSU file space to store this). Supply a voice-over as you demonstrate your project working. Demonstrate all of the required parts (insert, update, delete, report (with sorting), four+ table join(show the code), transaction(s), rollback). Limit the length to 5 minutes – anything after 5 minutes will not be used for grading. Start by displaying and saying your name, section, and user id. Next, display your crow's foot diagram for 5 seconds (this will allow us to pause the video and/or refer back to the diagram), and then start the demo – don't do anything "extra."

Please keep in mind that this is the "culminating experience" for the course, and it is expected that you will have to do outside research and some "figuring out" to complete this project. One of the ABET requirements (for accreditation) is that students have an experience where they are required to do independent problem solving.

CMPSC 431W – Project Design Document outline

NOTE: This is part of your writing ("W") grade.

Use the following outline to write your design document. Use the page lengths listed is a guide; these assume 12 point font, single spacing, normal margins.

1. Introduction (1/2 page)
2. Description of the problem to be solved (1 page)
    2.1. Screen shots and descriptions of all screens in your application (1 page per screen). Include a description of each field and each function.
3. Database Design
    3.1. Crow's foot diagram (1 page)
    3.2. Complete list of all tables including fields (if not in the crow's foot diagram)
    3.3. Complete list of all indexes and an explanation of why each is needed (each index explanation should be a full paragraph). *There should be at least four of these.*
4. List of all transactions (English description, not code)
5. Discussion of how error trapping is/will be handled (1 page, including code examples)
6. Summary/Conclusion (1 page)
    6.1. Discuss how this product will solve the problem as stated in item number 2 above.
    6.2. Discuss areas of technical difficulty and how they will be handled.