Building a Database-driven Web-based Information System

There is a semester-long course project of building a database-driven, Web-based application for a real-world domain of your interest. The project will be done in several stages: First, you will identify an application of your interest that requires database systems as backends. Then, you will design your underlying database and key functionalities you may provide with the database. You will create an actual database using MySQL, for example, and populate the database. After that, you will write queries and modifications on the database to support your application logic. Finally, you will build a Web-based interface to support readily interaction with your real-world application. You will demo your system to TA and/or instructor.

This programming project takes up 40 points out of 100 for your total score. You have another option to implement a simplified project without Web interface and access, with 5 points deduction. Namely, your highest score will be 35 if you choose this simplified option without Web programming. The programming project is a multi-stage project and for each stage, you need to submit a reports from Canvas. You cannot proceed to the next step until you get approval (with comments and suggestions) from TA in Canvas.

- Stage 1: Group formation (0%)
- Stage 2: A detailed description of your real-world application and the Entity-Relationship diagram (15%)
- Stage 3: Development plan (15%)
- Stage 4: Demo and final report (70%)

Stage 1: Group formation

You must form a group of one to three students (more than three is not allowed). Create a name for your group and include student names , okey id, emails information. If a group contains more than one student, you are tied up throughout the semester. If one happens to leave/drop the class, the others should be on your own. So choose your groupmate carefully!

Note : If you have difficulties forming a team, 1) go to class and talk to people in the class; 2) ask *TA* for help.

Stage 2: Application description and ER diagram

Your group identifies an application you would like to manage with your database. Please pick an application of your interest, as you'll be working with it for the entire semester. Your application should be substantial, but not too complicated.

You need to submit a project summary including:

- Description of the application domain of your choice. State as clearly as possible what you want to do, and pay attention to the following factors: (1) Usefulness. State as clearly as possible why your chosen application is useful. Make sure to answer the following questions: Are there any similar or equivalent websites/applications out here? If so, what are they and how is yours different? (2) realness. Describe what your data is and where you will get them.
- Description of the functionality that you plan to offer. This is where you talk about how to meet the functionality requirements. There are two types of functions you need to offer: (1) Basic Functions:

see Stage 4 for what of basic functions you need to offer. (2) Advanced Functions: Remember to include at least one creative thing (advanced functions), i.e., something that don't exist in equivalent websites/applications. The advanced functions should go beyond the basic functions listed above; in addition, an advanced function should also be something that is technically challenging, meaning you would need to spend some significant time (at least a few days work) to implement it. Of course, such functions should be relevant and therefore useful for your application. Make clear what the features are, and explain why you think it is cool to have them.

You should submit the ER diagram of your application domain (the minimum number of entities is 3, so is the number of relationships), plus English descriptions on the assumptions you make.

Stage 3: Development Plan

Each group must submit a detailed and well thought-out development plan including the following:

- The relational schema of your database. Remember to include all keys and dependencies (e.g., functional dependencies) as appropriate.
- The final choice of databases and software platforms/languages that you will be using.
- Where and how you will get data for your application. Do you get it from the Web, some other application, or do you make it up.
- Describe the labor division among group members.
- A project timeline with milestones.

Note: If you received a "Not Approved" review for stage 2, you need to address the staff's concerns in your Development Plan.

Stage 4: Demo and Final Report

Each group must demo its application. The demonstration takes 15 minutes, and will contain all of the following:

- **Basic Functions:** (1) show how to insert records to the database; (2) show how to search the database and list or print returned results; (3) show a few different interesting queries over your database. One of the queries must involve join of multiple tables, and one must be an aggregate query; (4) show how to update records; (5) show how to delete records.
- **Advanced Functions:** the advanced functions should go beyond the basic functions listed above; in addition, an advanced function should also be something that is technically challenging, meaning you would need to spend some significant time (at least a few days work) to implement it. Of course, such functions should be relevant and therefore useful for your application.

Clarifications on advanced functions: An advanced function should be: (1) useful for your users; (2) technically more challenging than the basic functions; (3) creative or novel with respect to other similar applications in your domain. Examples: It's an advanced function because of novelty: as an example, there was a project that archived all the weather forecasts from several major weather sites. One of its advanced function predicted which weather site is the most accurate, given the current weather state. It's an advanced function because of implementation. As another example, there was a project that does recipe search. One of its advanced function is to recommend recipes based on the user's dietary constraints. Users enter their constraints, such as calorie intake, ingredient preferences, etc, and the function returns a list of recipes that meet the constraints. The novelty comes from their search implementation. The search is an exhaustive search over 1000+ recipes, involving multiple table joins and pair comparisons, and they optimized this search with indexes and views. Essentially, advanced functions are opportunities for your group to be creative and fun. At demo time, your group needs to make a case why your functions are advanced.

Report requirement: Please hand in your final project report to the TA during your demo. As a complete project report, you need to include all the detailed contents of your project and write it as a complete story. The report is expected to be single-column, 5-10 pages in length (the font size is 10-12pt). You need to address the following key components in your report:

- **Introduction**: What is your application? Why do you want to choose this application, instead of others, as the course project (motivation)? What are the key components in your project?
- **Database details**: How do you design the database? What is your E-R model and relational model? What are the tables you include in the database? How do you design your database to accommodate functional dependencies? Are your tables in BCNF or 3NF? How about other constraints?
- **Functionality details**: What are the basic functions and advanced functions in your database and how do you implement them?
- **Implementation details** : What languages and platform you've chosen to do the implementation? How do you implement the front-end Web interface and the application logic? How does the front-end Web interface interact with the backend database? Students are encouraged to upload their code to some online repositories, such as Dropbox, Github, Bitbucket, and provide the link in the report.
- **Experiences** : What have you learnt from this project? How have you solved hard problems in this project? How to extend your project to more advanced, mature systems in the future?
- **References** : Any resource you have turned to for help during the implementation of your project