EECS 341 Database Projects

Jiong Yang; Fall 2007

The course project will give you a chance to (a) put into use the concepts you have learned in the database course so far, (b) learn a DBMS (Microsoft SQLServer or another one), and (c) become familiar with an application development framework; e.g., .Net framework and C#; or Java app development environment.

You are to write a project report with each of the steps below explicitly listed as a section, and each section elaborating on and responding to the tasks you have performed in that step. Underlined phrases of each section below are the section titles in your report.
Here are the steps of your project (and thus the major sections and their titles in your report. You are free to use as many subsections of your choice as you want).

1. *Introduction and Overview*. Give an overview of the environment about a web-based application that deploys a database needs to be developed. I will refer to the application as application X and the database as database D.

2. *Application Requirements Specifications.* List extensively the requirements specifications of the application X.
Note that the requirement specifications of X list in detail what X will do, what types of web interfaces it will have and their functionality (if possible, with mock-up web forms that you can put together), and the user interactions with the system. This section should be several pages, convincing me that you have spent enough time thinking over the details of your application, and you know what you are about to build as an application.

3. *Database Requirements Specifications.* List extensively the requirements specifications of the database D: (a) data (objects and relationships) to be maintained in the database, and their details, (b) queries and transactions that the implemented system will employ and the possible frequencies of these queries and transactions, (c) events, actions (triggers) of the database, and the (d) integrity constraints associated with the database. This section should also be several pages, illustrating your mastery of the knowledge needed for building a viable application.

Evaluation: Steps 2 and 3 evaluate your ability to analyze a problem, and identify and define the computing requirements appropriate to its solution. The evaluation score will reflect the following items: the project team
(1) Correctly observes requirements specifications,
(2) Understands and effectively uses the iterative nature of requirements specification refinement,
(3) Carefully documents requirements specifications, and
(4) Correctly identifies the tradeoffs involved in design choices.
(5) (Together with steps 5, 6, 7, 8), the accuracy and performance analysis of the specific algorithm chosen is complete.

4. *ER Data Model Design.* Design the ER data model of your database in detail. List the entities and their attributes. Specify the domain of each attribute. Specify the properties of each attribute (i.e., key, foreign key, composite/simple, single-valued/multi-valued, derived, incomplete with different nulls, roles, weak/strong entity type, etc.). Specify the relationships and their attributes. List the properties of each relationship (i.e., degree of the relationship, cardinality ratios (1-1, 1-N, N-M), participation constraints (total, partial), other application-specific constraints, etc.). Your ER model should have at least 5 entity types, 7 relationship types, and sufficiently many attributes and constraints of different types. Draw the E-R diagram, and incorporate everything discussed in this step into the E-R diagram.

Evaluation: Step 4, together with the other design and implementation steps, evaluate your ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
The evaluation score will reflect the following items: the project team
(1) Develops a design strategy, including a plan of attack, decomposition of work into subtasks, development of a timetable.

5. *Integrity Constraints for the ER Model.* List any integrity constraints in general on the entities and relationships as a whole. Explain how you intend to enforce them; i.e., are you going to build enforcement mechanisms by (a) specifying SQLServer constraints, triggers, (b) applying normalization (e.g., functional dependencies) (don't do normalization here; you don't have the relational model; just explain), or (c) applying external integrity enforcement (e.g., transactions/user-defined procedures, etc). If you need to enforce any constraints at database initialization/update time, list them, and specify how they are to be enforced.

6. *Transforming the ER Model to the Relational Model.* Translate your E-R model (by using explicit transformation) into the E-R model. Each entity should map to a (entity) relation, and each relationship should be accounted for (either represented in an entity relation or as a separate relation). Each entity/relationship attribute should be accounted for in the transformation. List for each relation the primary/candidate keys, foreign keys. Specify the entity and referential integrity constraints, and discuss why they are satisfied for each relation.

7. *Relational Database Design--Applying the Dependency Theory.* Perform normalization by applying all those algorithms you have mastered in the class. Are all your relations in BCNF or 3NF? If not, apply the algorithms you have seen in the class to decompose and make them BCNF/3NF. Please note that your decompositions should be lossless and dependency preserving.

Evaluation: Step (7) evaluates your ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. The evaluation score will reflect the following items: The project team
(1) Understands how sub-problems interrelate and demonstrates ability to integrate prior knowledge into new problems,
(2) Fully comprehends the time and space-complexity of the application being designed,
(3) Observes and completely defines the formal model of the system being designed,
(4) Correctly identifies the tradeoffs involved in design choices, and
(5) Uses algorithmic techniques effectively.

8. *SQL Queries and an Exercise in RA and RC.* List the queries of your project in English. Choose the toughest 5 of them, and specify them in SQL, RA and TRC.

9. *Revisiting the Relational Database Schema.* On the basis of the analysis you have done in sections 8-10, you may perhaps choose to revise your relational database schema (merging/splitting relations) so that (a) your most frequent and most costly queries run faster, and/or (b) your integrity constraints are enforced faster/easier. Explain your decisions.

Evaluation: Step 9, together with steps 4, 5, 6, 7, 10 and 11, evaluates your ability to apply design and development principles in the construction of software systems of varying complexity.
The evaluation score will reflect the following items: The project team
(1) Can relate design and development concepts to practical problem solving,
(2) Can defend design and development decisions effectively,
(3) Uses appropriate resources to locate design and development information needed to solve problems,
(4) Demonstrates understanding of how various pieces of the software system relate to each other and the whole,
(5) Formulates strategies for developing a complex software system, and
(6) The software system developed is correct and properly functions.

10. *DBMS (SQLServer?) Implementation.* (Start using the DBMS (SQLServer?). Create your schema, constraints, triggers and queries. Populate your database, and run/test your queries, triggers, constraints. Develop and test your stored procedures). Summarize the main components of your code here with proper explanations. Present (using screen dumps of the query analyzer output, if you are using SQLServer) the actual logical query trees for the five most difficult queries. Discuss any problems encountered and how you have solved them. If some of the problems require you to extensively redesign everything, you may choose to point them out, and not implement them. If some of your stored procedures are too elaborate, scale down your design, explain your decisions, and implement the scaled down version.

11. *Application Implementation* (.Net users: Start using .Net. Develop your asp.net and ado.net code for transactions, web forms, web services, etc. As you develop your code, add comments to your code—to the point that you can "extract" these comments for a "programmers manual" (.Net allows you to do this very easily)). List your code here, and add actual screen dumps etc. to illustrate the functionality of your code. Discuss any problems you have encountered, and how you have chosen to solve them.

Evaluation: Steps 10 and 11 evaluate your ability to use current techniques, skills, and tools necessary for computing practice. The evaluation score will reflect the following items: the project team

(1) Uses literature- and/or web-based resources effectively in assignments/projects,
(2) Seeks information on problems from multiple resources including appropriate on-line material,
(3) Is able to interpret and understand information from a variety of resources,
(4) Maintains current, state-of-the-art abilities in web and PC use,
(5) Is able to learn and effectively use stand-alone software or web-based resources,
(6) Understands and effectively uses digital libraries and other scholarly sources (web-based or not),
(7) Suggests new approaches and improves on what has been done before, and
(8) Uses programming techniques effectively.

12. ***Revisiting the Whole Project.*** Discuss any problems encountered overall during the whole project progress. What else would you have changed given more resources? What else would have been useful to add to the system? How could your project evolve into a commonly-used application?
Evaluation: Your responses to step 12 evaluate your ability to engage in continuing development. The evaluation score will reflect the following items: the project team
(1) Demonstrates ability to learn independently,
(2) Goes beyond what is required in completing the project, and brings information from outside sources into projects,
(3) Learns from mistakes and practices continuous improvement, and
(4) Demonstrates capability to think for one's self.

13. ***Team work.*** Specify in detail every team member's work.

14. ***Conclusions.*** Discuss here anything else you want to say, and conclude.

15. ***Appendix 1.*** Installation Manual. Write an Installation Manual that explains how one can compile and implement your system.
16. ***Appendix 2.*** Users manual. Write a Users Manual that explains to a naïve user how to start using your application.

17. ***Programmers Manual.*** Write a Programmers Manual that explains the classes, functions, etc and their functionality. Post your report, data, etc., on your web page. Supply any password info that I may need to run your application. Submit everything by the deadline posted in the class web pages.
Evaluation: This component is part of the evaluation listed in step 4. The evaluation score will reflect the following items: the project team
(1) Supports design procedure with documentation and references.

18. ***Code.*** At the time of your demo, prepare to walk through your code with me or the TA if needed, and to turn it in.

Grading:
1. Report. 35% (answers to the evaluation points. And: motivation, style, expressiveness, clarity, layout, grammar/spelling).
2. Design. 30% (Answers to evaluation points. And, originality, redundancy removal, realistic/simplistic, correctness, elegancy and simplicity and functionality of the design; in-depth discussions/coverage illustrating that you have done your homework in design. (This will be demonstrated by your presentation in your report)).
3. Implementation. 35% (Sophistication, code/effort size, how well it works, look-and-feel).