

# CS 417 Semester Project

Tom Kelliher

## 1 General Description and Goal

The goal of the semester project is to implement an application of a database system. This includes:

1. Finding an application for which a database system would be required.
2. Modeling the domain of the application, and defining the application functionalities.
3. Designing and implementing the schema.
4. Populating the database.
5. Building the application.

Projects will be done in groups of 2–3 students. Students are encouraged to pick an application domain that excites them, provided it has a nontrivial database component.

## 2 What Should be in the Project?

Every project should illustrate the following features:

1. Conceptual design of the domain.
2. Schema design for the application.
3. Sophisticated querying.
4. Database updates.
5. Use of multiple views.

A web front end to the application, using web.py, is expected.

## 3 Guidelines for the Semester Project

Here are some guidelines that should help you make project decisions. Of the following, the first three guidelines are the most important.

1. Effective use: Try to make effective use of as much of the course material as possible. As we progress through this course, try to incorporate what we learn into your projects. For example, use joins, aggregates, indexes, triggers, etc.

2. **Completeness:** The project must be complete and stable enough for a good demonstration at the end of the course. This requirement is very important. If you are unable to give a reasonable demonstration, your project grade will suffer greatly.
3. **Database size:** All project databases must be of nontrivial size. You may interpret nontrivial based on your application.
4. **Practicality:** The application you build should be of practical use to a sizable community (at least in the foreseeable future).
5. **Innovation:** The more innovative ideas you include in your project, the more credit you will receive.
6. **Miscellany:** In addition to the above, your project grade will take into account factors such as teamwork, overall effort, timeliness, answers to questions about the project, justification of design decisions, and intermediate and final project reports.

### 3.1 Groups Decided

One member from each group must send me an email with:

1. The names of the members of the group.
2. The group name. (The name must adhere to the traditional Linux username conventions, consisting of one to eight alphanumeric characters with the first character being non-numeric.)

The group name will be used to create each group's phoenix account and PostgreSQL database. Project deliverables will be published on the web under the group's phoenix account. Document deliverables should be published as MS Word documents. Image deliverables, if they can't be inserted into a document, the preferred way of publishing them, should be published as PNGs or JPGs.

Be sure to consider the following in selecting your partners:

1. Project interests.
2. Working style.
3. Goals.
4. Target grade.
5. Availability outside of class.
6. Strengths and weaknesses.

Do not be shy about interviewing potential partners — you are forming a short-term business venture.

### 3.2 Proposals

Students must put on their group web pages (this should only be a page or so long):

1. Project description: what is the domain, what aspects of the domain will be modeled by the database.
2. What are the application specifications? I.e., what functionality will the system provide?
3. What is the role of each project member in the project?
4. Schedule: what are the landmarks in your work.
5. Other, more specific comments if appropriate.

The proposals are meant to get you to start thinking about the project, get into groups and create a plan. All group proposals must be unique.

### 3.3 E/R Design and Relational Schema

Each group must post to their project web site:

1. E/R diagrams. Use `dia` with the ER sheet or some other software tool to produce a professional-looking E/R diagram.
2. Relational schemas. Again, use `dia` or another software tool. If using `dia`, use the Database sheet.
3. A written description detailing exactly what data their database will contain. This description should be fairly thorough.

### 3.4 Formal Specification

Each group must post to their web site:

1. What data they will have in their final application.  
What tables are you creating, containing what pieces of information each. (That means revisiting your design and figuring out for good what tables you need to create, taking into account the principles of good design and your desired functionality.)
2. What constraints exist and how they are modeled.
3. What functionality the final application will have; each project should have at least one cool feature. (The tasks each user of the system (including the system administrators) will be able to perform with your application.)
4. Updated division of labor.

### 3.5 Sample End-to-End Application

Each group must post to their project web site:

1. A couple of screen shots of their application running a simple query: getting the data from the database, and putting it into a simple version of the final application of the project.

### 3.6 Completed Project

Each group must have a demo of their project. By the project completion deadline, each group must have on their web site in its final form (no working after the deadline):

1. A short description of the finished project.
2. A description of how this differed from their formal specifications, if at all.
3. Documentation for their application. The documentation should address the person who will be installing and maintaining the application. (That is, the system administrator, not the end user.)