

# Semester Project

Tom Kelliher, CS 318

Jan. 28, 2002

This project description is adapted from the following:

1. <http://www.stern.nyu.edu/~vassalos/class/db/>
2. <http://www.cs.umd.edu/class/fall2001/cmsc424-0101/>
3. <http://courses.cs.vt.edu/~cs4604/>

You will find it useful to visit these sites and poke around.

## 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 3–4 students. Students are encouraged to pick an application domain that excites them, provided it has a nontrivial database component. See the suggestions below.

## 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 PHP, 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. However, (1) there should be at least 10 nontrivial relations; and (2) the total number of attributes times the total number of tuples should be at least 5000. (These are guidelines only; if you feel your application warrants exceptions, come see me during office hours.)
4. **Practicality:** The application you build should be of practical use to a sizeable 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.

### 4 Schedule

There are several milestones we will follow to help you with assessing progress:

1. Groups formed: Feb. 1.
2. Proposals: Feb. 8.
3. E/R design: Feb. 22.
4. Formal specification: Mar. 8.
5. Sample end to end application: Apr. 5.
6. Project demonstration, outside of class by appointment: week of Apr. 29–May 2.
7. Final project report due: May 3.

Each group is permitted two “slide” days, although the groups formed, the project demonstration, and the final project report due date are hard deadlines — no slides permitted. Otherwise, 10% of the final project grade will be deducted for each day a milestone is late. Deliverables are due by 5:00 PM on the due date.

## 4.1 Groups Decided

One member from each group must send an e-mail to [kelliher@goucher.edu](mailto:kelliher@goucher.edu) with:

1. The members of the group.
2. The group name.
3. The URL of the group web page. Note: all documents posted to the group web page must be readable using Netscape running on any OS platform. This rules out posting MS Word documents, for example. You may post PDF documents.

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. For the purpose of creating the formal specification, I would suggest that each group contain at least one person from the software engineering class of fall 2000.

## 4.2 Proposals

Students must put on their group web pages (this should only be a page or less 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. Send an e-mail containing the URL of your proposal to the entire class ([GC\\_CS\\_318\\_001\\_SPRING\\_2002@gnav.goucher.edu](mailto:GC_CS_318_001_SPRING_2002@gnav.goucher.edu)) once your group has posted its completed proposal.

## 4.3 E/R Design

Each group must post to their project web site:

1. E/R diagrams.
2. Relational schemas.
3. A written description detailing exactly what data their database will contain.

#### **4.4 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.

#### **4.5 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.

#### **4.6 Completed Project**

Each group must have a demo of their project. By May 3, 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.)

Also by May 3, each student is required to email me with the following information:

1. How long did you work on the project?
2. What did you like the best about the project?
3. What did you like the least about the project?
4. What helped you learn the best in the project?
5. What distracted from your learning in the project?
6. If you could change the way that the project was organized, what would you change?

Answering these brief questions will help me make future projects better.

## 5 Project Ideas

The following ideas are merely suggestions — you have to flesh out your specific application. Also see:

- <http://courses.cs.vt.edu/~cs4604/assignments/proj1.html>
  - <http://www.cs.umd.edu/class/fall2001/cmsc424-0101/project-ideas.html>
1. Movies and reviews database.
  2. Online store (books, CDs, mp3s etc).
    - (a) Support browsing, searching, shopping cart, personalization.
    - (b) Model and include product information such as hierarchy of book topics, or music genres.
  3. Apartments.
    - (a) Model apartments and their properties, areas of town and their various properties (e.g., bus lines, crime rate distance from various landmarks).
    - (b) Offer apartments for rent, browse apartments, query using specific criteria.
  4. Library DB.
    - (a) Search collections, check status.
    - (b) Integrate with the Goucher Library DB (hard).
  5. Job classifieds database.
    - (a) Openings, contacts, references, reviews, requirements, bidding, ...
  6. Database of web sites.
    - (a) Model web sites: topic, URL, organization to which they belong, other sites they point to, reviews of the site, quality, etc.
    - (b) Update capability: New sites, new reviews of sites.
    - (c) Browse and search capabilities.
    - (d) Personalization.

See Yahoo.com for a partial inspiration.
  7. Room scheduler.
  8. Historical stock price data.
    - (a) retrieve, parse, and store stock price information from \*public domain\* sources over time.
    - (b) implement browsing and querying that can be used to discover trends, records, trivia.
  9. Goucher administrative database.

Simple idea, you have to come up with interesting functionality. GNAV, perhaps?

10. Personal Information Manager (PIM)

- (a) contacts, addresses, phones, ...
- (b) appointments.
- (c) diary
- (d) to-do list