

Database Systems

Term Project

Spring 2013

As is indicated on the syllabus for this course, each student is required to contribute to a database design project this semester. This document outlines the scope of the project and the basic operating rules and guidelines.

Scope of the project

You are to analyze the requirements for, design, implement, demonstrate, and document an application that supports one of the two applications that are described in this document. Clearly, the application will involve development of a Web interface that accesses a database hosted on a server somewhere. The problem descriptions in this document are intentionally incomplete. Part of the project is filling out the requirements by thinking about the application domain, trying to understand what the needs really are, and exercising some creativity and thoughtful analysis.

Problem Description:

Each team will choose to engage in either Choice 1 or Choice 2 as described in this document. The descriptions in this document give the broad parameters of each choice, but you are (of course) free to add features as you find it appropriate to do so.

General assumptions

- You are not allowed to assume away a system requirement.
- You are free to add to the set of relevant data items and you are free to add to the set of tasks. You are not free to simplify the problem overly by not representing at least all of the relevant data or deleting tasks from the set of tasks in the description of the relevant choice. If there should be some reason why you would not want to implement one of the tasks described, explain why to the instructor, and we'll see.
- It is intended that this would be a web-based application. That means that some level of security should be provided. You should consider security issues as part of the design.

Basic Operating Rules

1. The project is to be done in groups of two or three individuals. No one is allowed to work alone. A roster for each group has been established. It is each group's responsibility to determine its own division of labor. You must indicate an individual who takes responsibility for each aspect of development and you must designate a backup person (in case someone falls down in front of a truck one morning). *If some truly irreconcilable issue arises, it is the group's responsibility to contact your*

instructor as soon as possible. If the instructor does not hear about these matters until after the project is due, she won't have a lot of sympathy.

2. In cases where the description of the application is incomplete (and there are some of these), it is acceptable to make assumptions about the application providing that: 1) they are **explicitly** stated in the report, and 2) they **do not conflict** with any of the requirements specified above, and 3) they are "reasonable". If you have questions about the acceptability of any of your assumptions, check with the professor.
3. A report must be handed in for grading at the end of each phase. The report must be formatted in a reasonable manner (i.e. using a text processor and a decent printer). Reports are due via Moodle on the date specified in the "Due Dates" section of this assignment.
4. The third phase of the project requires a working implementation of the system to be built, tested, and demonstrated. A significant part of the project grade depends upon the quality of this implementation. The implementation will be done using My SQL with PHP (unless your group has the instructor's permission). As part of a request for permission to use another framework, you must submit a written argument as to why some other framework would be better. A portion of the project grade will be based on the user interface. A portion of the grade will also depend on the quality of error detection and management, i.e. your system should be robust to data entry errors, etc.

Project Phases

- | | |
|------------|--|
| Phase I: | Requirement Analysis and System Analysis |
| Phase II: | Schema and Application Program Design |
| Phase III: | Implementation followed by a demo |
| Phase IV: | Testing another group's product and assessing the quality of the user documentation. |

Project Reports

The phase I report must contain:

1. [5 points] a short description of the purpose of the project and the purpose of this phase of the project
2. [15 points] A list of the assumptions that you have made about the enterprise
3. all the documentation produced in this phase, i.e.
 - [15 points] A complete description of the format of each input document that provides information to be stored in the database (make reasonable assumptions as you find it necessary).
 - [15 points] A complete description of the content and format of each routine report to be produced using the database
 - [10 points] Sketch the input and output screens for each routine transaction to be performed against the database

- [15 points] Generate a data dictionary for the project that includes an alphabetical list of every data item referenced in any report or routine transaction and an informal description of each term
- [15 points] Construct a cross-reference table by writing the names of all forms, reports, and transactions across the top and all items in the data dictionary along the side. If a data item on a given row appears on a particular form, report, or transaction, put a mark in the corresponding table entry.
- [10 points] Make a project timeline listing major tasks and subtasks, indicating assignment to team members, and providing dates for completion.

The phase II report must contain:

1. [5 points] a list of any revisions that were made to the specification described in the phase I report.
2. the documentation produced in this phase, i.e.,
 - [20 points] the graphical schema using the E-R model,
 - [5 points] a list of the attributes for each entity and relationship,
 - [5 points] explanations of the non-obvious entities and relationships,
 - [20 points] the schema in some appropriate Normal Form (BCNF or 3NF at worst!)
 - [5 points] explanations (e.g., primary keys, additional functional dependencies, why a table is not in BCNF, etc.),
 - [10 points] the DDL statements to create the above relational schema
 - [30 points] the pseudocode for each task and any embedded DML code

The phase III report must contain:

1. [20 points] the relational schema definition from Phase II and any revisions made to the specification that you included in your Phase II report (along with an explanation of why you made these revisions).
2. the documentation produced in this phase, i.e.,
 - [20 points] a source program listing,
 - [10 points] a users manual for the system
 - [20 points] your testing efforts: erroneous cases that your system can detect and handle reasonably
 - [10 points] a description of the systems limitations and the possibilities for improvements.
3. [20 points] In addition, a demo of the database will be required. All members of the group should attend this demo, to explain the aspects of the project for which they were responsible. The quality of the user interface will be a factor in the grading of this demo.

Due Dates

Project Starts: 3/11

Group rosters due: 3/14

Phase I due: 3/29

<< Spring Break is March 15 – 24 >>

Phase II due: 4/12

Phase III due: 4/26

Phase IV due: 4/29

Finals Week is April 29-May 3 and the final exam in this course is scheduled for Wednesday, May 1 at 11:45 AM in Rowley 244.

Choice 1

This application is a system that aids the user in meal planning. The idea is that the user has a set of available ingredients in his or her refrigerator and cupboards. The amount of each ingredient that is on hand is known. The user also has a set of recipes in his or her stored “cookbook”. These recipes may have attributes such as ease of making, healthiness, cost to make, type of meal, time of preparation, etc.

The system has at least the following capabilities:

- It should tell the user what (in the “cookbook”) can be made with the ingredients that are on hand.
- It should consult the stock of ingredients on hand and produce a “grocery list” of ingredients that are needed to make a particular recipe.
- It should allow the user to provide a set of recipes for the week and produce a grocery list of items required to make these recipes.
- It should allow the user to search for recipes that fall into particular categories (such as easy dinner recipes that take less than an hour to fix or healthy breakfasts, etc.).
- It should allow the user to search the recipes for those that contain particular ingredients. (So, for example, it should be able to tell the user what recipes are salmon recipes or what recipes are pasta recipes.)
- It should be able to display a recipe for the user.
- Other features may be added as desired.

Choice 2

This choice involves an application that would aid a student in planning his or her program of study. It would enable the user to track progress towards satisfaction of major degree requirements and help them to plan a semester by semester set of courses to take.

It must store (at least) the following information:

- What courses the user has already taken (and the grades the student has attained in them).
- What courses the user is currently enrolled in
- What courses are required for graduation in the user's major
- What elective course slots are left in the program of study and what restrictions are applicable to each of these slots.
- What the prerequisites are for the courses a student might possibly choose to take
- Which semester(s) each of the potential courses is offered.
- Other data may be needed.

The application must be able to:

- Report the courses the user has taken
- Ask the user what his or her major is and:
 - Report the courses the user still needs to take
 - For required courses, indicate the semester(s) they are offered.
 - For elective choices, indicate what "kind" of elective they are.
- Allow the user to engage in "what if" style planning to lay out a program of study.
- Calculate the student's overall GPA and also the GPA in major (because both are important in determining whether or not the person will graduate).
- Other functionality could be included to enhance the application.