Lab 4: Software Design, Database Creation, Query Prototyping

Due date: Thursday, February 21, 4:00pm.

This is a **group lab**. Each group submits one set of deliverables. Each group member receives the same grade for the assignment.

Lab Overview

This is the last project-related lab for the class. During this lab each group will do the following:

- Finalize the database design.
- Create the database using data provided by the customer.
- Prepare the software design and the GUI prototypes for the key functionality of the software.
- Develop and debug prototypes for the SQL (or SQL/PL) queries to be used by the software you are developing.

The lab is organized into three stages:

1. Preparation. February 5. Each team studies the software spec and the specific assignment, prepares questions for the customer.
2. Software Requirements Elicitation. February 7. Each cohort consults with Stephanie Yarbro concerning their specific project.
3. Database creation, software design. February 12 – February 21. Teams prepare software design, finalize and build their database, put customer data in it, develop and debug queries.

Software Specification

The customer has submitted the project specification document. The document is available from the course web page and the project wiki. The direct URL for the document is:

> http://users.csc.calpoly.edu/∼dekhtyar/366-Winter2008/project/Software-spec.doc

The document outlines the overall software needs of the horse nutrition program. Due to the tight timeline of the course, each project team will implement a subset of the overall specifications, related to a specific collection of functionality. The following three software products are to be designed and implemented.

- **Day-to-day Management Tool.** Green cohort teams. The teams of the green cohort are tasked with the implementation of the *"base"* system for management of animal diets and inventory. The key components of the system are management of diets (creation, assignment to animals), management of animals (introduction of horses into the unit, assignment of locations) and management of inventories (tracking quantities of feed, purchases and withdrawls). This software is to be developed in Java.
- **Web-based tool.** Red cohort teams. The teams of the red cohort are tasked with the implementation of the web-based access to the animal nutrition database. The software will concentrate on retrieval of information from the database and its display for various categories of users, while the management component of the software will be limited to diets. This software can be developed using any web programming frameworks the teams are profecient with, with the caveat that the teams are responsible for proper database connectivity to the CSL Oracle server.
- **Analytical tool.** Blue cohort teams. The teams of the blue cohort are tasked with the implementation of the analytical tool, which allows the nutrition program and farm managers to track use of inventory and predict, based on various models, the necessary amount of feed to be ordered in the future. The analytical tool will have little management functionality and will cater to a limited number of user categories. However, it will involve more complex reports. The tool is to be developed in Java.

Specific instructions for each tool are given below. The instructions below assume familiarity with the customer specification document, and refer to various parts of it as needed.

Common Functionality

This section describes the software functionality that is common to all three software tools, and thus, must be present in every project implementation.

1. **Full database.** Each group shall create the full horse nutrition database. Notice, that it is possible that your group's database is designed in a way, that some components of the database are never accessed by your software. Even in this case the database must be created and stored in its entirety.

2. **Bulk-loader.** Each team's project must include a bulk-loading component, which takes the data presented by the customer and populates the team's database with data. The bulk-loader is expected to be separate from the main software tool built by the group, although green and blue cohort may integrate its functionality (in a properly protected manner) in the main tool for convenience. The bulk-loader is described in detail below.

3. **User/Login management.** Access to the functionality for each type of software needs to be controlled. While different tools have functionality targeted at different user categories, each tool is assumes at least two user categories with different permissions as far as the functionality of the tool is concerned.

Green Cohort: Day-to-day Management Tool

Each team in the Green cohort shall implement a version of the day-to-day horse nutrition program management tool. The exact functionality of the tool is described below.

User Categories. The tool shall be built for the following categories of users:

1. Program Administrator

2. Center Administrator

3. Manager/Student Manager

4. Student Employee

Any tasks in the customer specification limited to the user groups not listed above are explicitly excluded from the day-to-day management tool specification.

Functionality. The purpose of the tool is to provide employees of the nutrition program the ability to manage the nutrition program. The following functionality found in the customer specification document shall be implemented in the tool:

1. Taxon. The tool shall implement all tasks and all queries associated with the TAXON component of the database (animal taxonomy).
2. Specimens. The tool shall implement all tasks and all queries associated with the SPECIMENS component of the database (tracking individual animals housed in the facility).
3. Locations. The tool shall implement all tasks and all queries associated with the LOCATIONS component of the database (locations of animals).
4. Ingredients. The tool shall implement all tasks associated with the INGREDIENTS component of the database (feed ingredients). The tool shall implement the following queries associated with the INGREDIENTS component (query numbers refer to the customer specification):
   (a) Query 1.
   (b) Query 3.
   (c) Query 4.
   (d) Query 5.
   (e) Query 6.
   (f) Query 7.
5. Diets. The tool shall implement all tasks and all queries associated with the DIETS component of the database (meals/feedings and animal diets).
6. Users. The tool shall implement all tasks and queries associated with the USERS component of the database, w.r.t., the user categories listed above.
7. Global Tasks and Queries. The tool shall implement global task 1 (requests for diet reviews). The tool shall implement global queries 1 and 2 w.r.t. the ingredient quantities needed to be withdrawn for a single feeding. The tool shall implement global query 4.

Front End. The GUI of the tool shall be appropriate for the intended uses of the system. In particular:

1. No category of users shall have access to the tasks they are not authorized to perform. This means that all tool menus as well as any dialog and report windows shall be designed to be user-category-dependent. (for example, program administrator has the privilege to assign diets

to animals. This functionality may be made available from a dialog window displaying a list of animals. Farm managers are not allowed to assign diets, but are allowed to request diet reassessment. When farm managers browse the list of animals, diet reassessment request functionality shall be made available to them instead of diet reassignment.)

2. The tasks shall be broken into convenient subcategories. E.g., inventory tracking is a separate subcategory of tasks from animal diet assignment. Most of these may parallel the user categories.

3. Tasks and queries shall be represented in *logical* manner.

Red Cohort: Web-based Tool

Each team in the Red cohort shall implement a version of the web-based tool for access to the horse nutrition database. The exact functionality of the tool is described below.

User Categories. The tool shall be built for the following categories of users:

1. Program Administrator
2. Center Administrator
3. Manager/Student Manager
4. Student Employee
5. Animal Owner
6. Guest

Functionality. The tool's purpose is to provide nutritionists, farm managers and farm employees with off-site access to the information stored in the database as well as to give them some limited database management abilities. The tool also provides the opportunity for the horse owners to keep track of the information about their horses and communicate with the nutritionists. Finally, some general information from the database will be made available to general public. The following functionality found in the customer specification document shall be implemented in the tool.

Overall, the web tool shall provide similar access to the information as the day-to-day management tool. However, the management functionality of the tool is mostly restricted to horse nutrition (ingredients, diet creation, diet assignment) tasks, whereas tasks related to inventory are **left off** the functionality.

1. Taxon. The tool shall implement all tasks and all queries associated with the TAXON component of the database (animal taxonomy).

2. Specimens. The tool shall implement all tasks and all queries associated with the SPECIMENS component of the database (tracking individual animals housed in the facility).

3. Locations. The tool shall implement all queries associated with the LOCATIONS component of the database (locations of animals).

4. Ingredients. The tool shall implement the following tasks associated with the INGREDIENTS component of the database (task numbers refer to tasks from the customer specification):

   (a) **Task 1:** addition of a new ingredient to the list. This task **shall not** include addition of any information about the available quantities of the ingredient.

   (b) **Tasks 2 and 3:** as needed to successfully complete **Task 1**. That is, if a new unit or new units are needed to complete the description of a new ingredient (or a new weight conversion of an existing unit is required), it should be possible to do so. However, these tasks shall not be available outside of functionality related to completion of **Task 1.**

   The tool shall implement the following queries associated with the INGREDIENTS component.

   (a) Query 1.

   (b) Query 3.

   (c) Query 4.

   (d) Query 5.

   (e) Query 6.

   (f) Query 7.

5. Diets. The tool shall implement all tasks and all queries associated with the DIETS component of the database (meals/feedings and animal diets).

6. Users. The tool shall implement all tasks and queries associated with the USERS component of the database, w.r.t. all user categories.

7. Global Tasks and Queries. The tool shall implement global task 1 (requests for diet reviews) for the user category animal owner. The tool shall implement global queries 1 and 4.

Front End. The GUI of the tool shall be appropriate for the intended uses of the system. In particular:

1. No category of users shall have access to the tasks they are not authorized to perform. This means that all tool menus as well as any dialog and report interfaces shall be designed to be user-category-dependent.
2. The tasks shall be broken into convenient subcategories. E.g., inventory tracking is a separate subcategory of tasks from animal diet assignment. Most of these may parallel the user categories.
3. Tasks and queries shall be represented in *logical* manner.
4. The overall design shall be nice and clean. Do not go overboard with facny features.

Blue Cohort: Analytical Tool

Each team in the Blue cohort shall implement a version of the web-based tool for access to the horse nutrition database. The exact functionality of the tool is described below.

User Categories. The tool shall be built for the following categories of users:

1. Program Administrator
2. Center Administrator
3. Manager/Student Manager

Functionality. The tool's purpose is two-fold. The tool is to provide the nutrition program and the farm management with the ability to assess the operations of the facility. The tool is also to allow for analysis of the use ingredients for animal diets, and, based on the performed analysis, it to predict future ingredient needs. The tool shall have rather limited data management functionality, however, it shall provide access to all queries and reports required by the customer.

1. Taxon. The tool shall implement all queries associated with the TAXON component of the database (animal taxonomy).
2. Specimens. The tool shall implement all queries associated with the SPECIMENS component of the database (tracking individual animals housed in the facility).
3. Locations. The tool shall implement all queries associated with the LOCATIONS component of the database (locations of animals).

4. Ingredients. The tool shall implement all tasks and queries associated with the INGREDIENTS component of the database (task numbers refer to tasks from the customer specification):

5. Diets. The tool shall implement all tasks and all queries associated with the DIETS component of the database (meals/feedings and animal diets).

6. Global Tasks and Queries. The tool shall implement all global queries.

Front End. The GUI of the tool shall be appropriate for the intended uses of the system. In particular:

1. The GUI shall be centered on the analytical tasks: tracking current inventory, reconciling current inventory with the expected use of ingredients and projection of the future ingredient use.

2. Data management tasks included in the funtionality of the tool shall be offered as add-ons to the main functionality.

3. Tasks and queries shall be represented in *logical* manner.

Lab Assignment

Stage 1: Preparation

The first lab period of the lab, Tuesday, February 5, is reserved for preparation. Each team is expected to do the following:

- Discuss within each team, and with the instructor if necessary, the instructor's comments on the Lab 3 submission. Implement any followup design changes.

- Read and discuss within each team the customer specification (you should be familiar with the draft specification by then). Ensure that the team understand the scope of the assignment. Discuss any outstanding questions with the instructor.

- Prepare a list of questions for Stephanie Yarbro. The questions may be about any issues related to the database and the software.

Between the Tuesday, February 5 and Thursday, February 7 lab periods, the team liasons for each cohort are expected to coordinate the cohort's Q&A session with Stephanie (to be held on February 7).

Note: The more I know about the intended questions by the end of the February 5 lab period, the more prepared Stephanie will be to work with each cohort.

Stage 2: Requirements Elicitation.

On Thursday, February 7, Stephanie Yarbro will visit our lab period. She will spend around 25 minutes talking to students in each cohort, leaving the ballance of the time to devote to any outstanding questions.

The format of the discussion within each cohort is as follows:

- The discussion will be for students of the given cohort. Students from other cohorts may listen in.
- Only the students in the given cohort may ask questions.
- The liasons for the teams of the given cohort are to coordinate the question list in advance.
- Students in the given cohort are expected to take notes on Stephanie's answers.

After the discussion, each group shall document the answers to the questions posed by the group members on the project wiki. This shall be done before the next lab period.

Stage 3: Design.

Lab periods on February 12 - February 21 are devoted to software design and database testing and creation. The following tasks shall be performed.

1. Each group shall complete its database design, and make appropriate changes in the relational model and the SQL DDL implementing it.
2. By February 12, the customer will make a sample database available. Each group shall create bulk-loading software: programs which take the input data and insert it into the tables in the groups' relational databases. Any programming language can be used for this task (Red cohort - this is a server-side task, so, you can use **any** language as well). The customer dataset will be files in CSV (comma-separated values) format.

   Your bulk-loader shall generate a sequence of `INSERT INTO` statements, and shall have the ability to output these statements to `stdout` or to a disk file.
3. Each group shall develop and debug SQL templates for each type of query the group is assigned to implement. An SQL template is an SQL statement for a user input-dependent query. Values of specific attrbiutes, which are expected to be provided by the user can be replaced with templates. The SQL templates can be presented as-is or wrapped into Java methods, PL/SQL procedures, or other mechanisms that allow parameter passing. Working examples of the SQL templates shall also be prepared.

4. Each group shall commence working on the software design. By the end of the lab, each group shall have a collection of GUI prototypes representing the group's approach to the design of the tool's front end. The prototypes can be made in the native programming environment for the group's project (Java for Blue and Green cohorts, selected web programming environment for the Red cohort). The can also be made *"on paper"*, i.e., a graphical representation of the design (drawn, e.g., in MS Powerpoint, or in a CASE-based development tool) can be presented, without being implemented.

Submission Instructions

As part of the assignment, each group has to update the project wiki with the information obtained from the Q&A session(s).

At the end of the lab, each group needs to submit the following.

Submit as hardcopy:

1. Final database design.
2. Changelog for the final design.
3. Original design submissions from Labs 2 and 3.
4. Initial software design specification. It should contain group's notes on the software design, and include screenshots/GUI design for all major components of the system.
5. Query templates. Each template must be accomanied by brief explanations. If your electronic submission of queries/query templates (see below) contains sufficient comments, you can simply submit a printout of it.

Submit electronically in a single .zip or .tar.gz file via email to dekhtyar@csc.calpoly.edu:

1. The DB-setup.sql file with all `CREATE TABLE` statements for the group's database.
2. The DB-insert.sql file containing all necessary `INSERT INTO` statements to populate the group's database.
3. The DB-drop.sql file containing `DROP TABLE` statements for each table in your database.
4. All SQL code for the queries. This can be organized as each group decides. Submit a README file explaining the structure of submission, and provide sufficient comments throughout the submitted file or files.

Please, keep the soft copies of all submitted documents. You will be working with them in the labs that follow.