

HW4: Teams and Databases

For this final homework before the project, you practice two key skills:

- collaboratively developing software with Subversion, and
- designing and implementing systems with a database backend.

As this is a team homework, you may discuss/share code as much as you want with your teammates. In fact, I encourage you to *work together—but only with your teammates*.

Although you will be working as a team, team members will have individual responsibilities (described below) on which they, and they alone, will be graded.

Getting going

Your team will be working on a project that I started. To facilitate collaboration, I have created an SVN project to be shared among your teammates and added an initial project.

To check out the project, use the SVN Repository Exploring perspective to add the repository:

https://utopia.cs.memphis.edu/course/comp4081-2012fall/teams/YOUR_TEAM/

For example, Team Jupiter's URL would be <https://utopia.cs.memphis.edu/course/comp4081-2012fall/teams/jupiter/>. Use your SVN username and password from before to authenticate. Then, check out the project's **trunk** using the usual method (i.e., **Find/Checkout As...**). In the New Project Wizard, don't forget to uncheck the checkbox that generates a DD (i.e., web.xml).

You will notice that I've also allocated your team a database on my server that the project accesses. (See the DD for the DB URL and authentication info.) You can use the MySQL Workbench (<http://www.mysql.com/products/workbench/>) to explore and manually manipulate the DB.

Warning: The DB limits the number of open connections that your team can have. In your code, make absolutely sure that you are closing connections when you are done with them (i.e., use a `finally` block). If you leak too many connections, the DB will refuse your new connections.

Tasks

Each team member must select one of the following feature-enhancement tasks to complete. The task you choose is the one you will be graded on. You may help your teammates out with their tasks, but just make sure that doing so doesn't come at the expense of you completing your own task. If your team has only 4 members, then ignore Task #5 below.

IMPORTANT! You must tell me which task you will be doing on the day that HW4 is assigned.

The app you will be working on is for a book distributor. It enables the distributor to manage its catalog of books. The system may be used by employees of the distributor, authors of the books in the catalog, and customers interested in seeing what books the distributor handles.

Task #1: View Catalog

This feature provides page that lists book information in a table. Users must be able to filter the books listed in the following ways:

- Show all books.
- Show all books by a particular author. Users should be able to select the author from a drop-down list (http://en.wikipedia.org/wiki/Drop-down_list).
- Show the book with a particular ISBN. Users should enter the ISBN in a text field.

Each row in the table must be a different book (i.e., no duplicate books in the table). The table must include the following columns:

- ISBN.
- Title.
- List of authors' last names. Here are a few examples:
 - Rand
 - Payne and Suphring
 - Dewey, Cheatem, and Howe
- Year of publication.
- Publisher name.

Task #2: Edit Author Info

This feature enables authors to edit information that is relevant to them.

First, the author must be able to select his name from a drop-down list, which brings up a form with all his/her personal information and books.

The author must be able to update his/her first and last names and year of birth.

The author must also be able to update the following information in each of his/her books:

- Title.
- Description.
- Year of publication.

Task #3: Update Book

This feature enables an employee of the distributor to update the information about books in the catalog.

The feature must first show a table listing all books in the catalog. Each row should have a different book and the columns should be the same as in Task #1 (View Catalog).

The user must be able to select a book in the table to update. This action should bring up a form with the following editable fields:

- ISBN (as a text field).
- Title (as a text field).
- Year of publication (as a drop-down list).

- Author #1–#10 (each as a drop-down list).
- Publisher (as a drop-down list).

Each field must be initially populated with the book's current information.

The user must be able to submit the changes or cancel. Submitting or cancelling must take the user back to the original table.

Task #4: Add Author, Add Publisher, Add Book

These three features each provide a form that the publisher can use to add new information to the catalog. Each form must be as follows.

The add author form must include the following:

- Author ID (as a text field).
- First name (as a text field).
- Last name (as a text field).
- Year of birth (as a drop-down list).

The add publisher form must include the following:

- Publisher ID (as a text field).
- Publisher name (as a text field).

The add book form must include the following:

- ISBN (as a text field).
- Title (as a text field).
- Edition Number (as a text field).
- Year published (as drop-down list).
- Description (as a text area).
- Authors #1–#10 (each as a drop-down list). Note that if nothing is selected for one of the drop-down lists, your app should not add anything to the DB for that drop-down list.

Each form must have submit and cancel buttons. Pressing either button must take the user to a page that reports the results of the operation (e.g., in the case of submit, was it successful?). The results page should also provide links to each of the add forms.

Task #5: Remove Authors, Remove Publishers, Remove Books

This feature enables distributor employees to remove authors, publishers, or books from the catalog.

The feature must first present a page with three forms in it: an author form, a publisher form, and a book form. Each form must contain a table as follows.

The author-form table must contain a different row for each author in the catalog, and it must have the following columns:

- Author ID.

- Last name.
- First name.
- Year of birth.

The publisher-form table must contain a different row for each publisher in the catalog, and it must have the following columns:

- Publisher ID.
- Publisher name.

The book-form table must contain a different row for each book in the catalog, and it must have the following columns:

- ISBN.
- List of authors' last names. Here are a few examples:
 - Rand
 - Payne and Suphring
 - Dewey, Cheatem, and Howe
- Title.
- Edition number.
- Publisher name.
- Year of publication.

Each row in each of the above tables must also have a checkbox, and below each table must be a remove button. The user must be able to select rows in each table to delete by checking the appropriate box(es) and then clicking the table's remove button. After a remove button is pressed, the remove page must update to display the revised tables.

Note that you must handle dependencies between tables in the DB in an intelligent way. For example, removing a publisher must also remove all books by that publisher. The DB may be set up to handle this automatically. Do some experiments to find out.

Additional Constraints

- Your team's project must use the MVC architecture.
- Do not implement any user authentication/authorization for this homework.
- Your team must make an **index.html** page that has links to each of your teams features. Designate this page as the welcome-page in the DD.
- Each of the above features must handle bad input and errors in a reasonable way. A feature must not fail silently. Displaying an error page with a link back to the **index.html** page will be acceptable in many cases.

Submission Instructions

After all team members have completed their work and merged it into the trunk, one member must tag the team's project as **HW4**. Use the "History" method of tagging described here:

http://www.cs.memphis.edu/~sdf/comp4081/homeworks/Revised_Tagging_Instructions.pdf