

## CIS 2308 (Kyvernitis) Insert Lab

### Overview:

In this lab, you'll create three new JSP pages: insertOther.jsp, insertUser.jsp, and insertAssoc.jsp. Each of these insert pages will perform field level validation and form level validation then communicate success or failure back to the user.

### Functional Requirements:

insertOther.jsp, insertUser.jsp, and insertAssoc.jsp shall:

1. Label and provide **one text box for each non-key field** in the database table.
2. Label and provide **one select tag for each foreign key field** in the database table.
  - insertUser.jsp shall have a pick list for role.
  - insertAssoc.jsp shall have a pick list for your "other" table.
  - Consider what makes the best sense for your web application. insertAssoc.jsp shall either:
    - Have a user pick list, OR
    - Use the id of the user who is logged on (or redirect if the user is not logged on).
3. When the user clicks submit, the form shall post to itself and all **user entered/selected values shall persist**.
4. Never show the word "**null**" on the page (not in first rendering, not in postback).
5. Display **field level error messages** next to any/all fields that do not pass field level validation (e.g., messages about required, wrong type, character string too long). All error messages shall be shown at once, not displayed one at a time.
6. Appropriately handle **nullable fields**. Double check that your database has the required nullable non-character fields and that your code handles that validation appropriately (or points will be deducted from this lab and future labs).
7. If there is a **problem getting a database connection**, the page shall provide a two part error message to the user.
  - The first part shall be "user friendly" (like "database unavailable, please try later") and the second part shall have the actual database exception message (for IT support).
  - Check that your display pages show this kind of database connection error message to the users. A good way to test for this is to run from home, not tunnelled in, and check all your pages.
8. Attempt to execute a **SQL INSERT** statement (using java.sql.PreparedStatement) if all field level validation is passed. Then,
  - If the insert is successful, a form level message shall be displayed to the user like "record inserted".
  - Otherwise, if a database exception was generated, a two part error message shall be displayed to the user.
    - The first part shall be a user friendly message first (like "that email address already exists in the database"), to explain the problem to the user.
    - The second part shall be the actual database exception message, to help IT support identify any unexpected errors.

### Navigational Requirements:

- **other.jsp** shall link to insertOther.jsp and insertOther.jsp shall link back to other.jsp
- **users.jsp** shall link to insertUser.jsp and insertUser.jsp shall link back to users.jsp
- **assoc.jsp** shall link to insertAssoc.jsp and insertAssoc.jsp shall link back to assoc.jsp
- **all these (6) links** shall be at the top of the pages in the content area.

## User Interface Requirements:

- The user interface should not “jump around” when errors are displayed. One way to avoid this is to not center a table that (sometimes) shows error messages.

## Design Specifications

1. **insertOther.jsp**, **insertUser.jsp**, and **insertAssoc.jsp** shall
  - a. Declare a database connection object, pass that to classes/methods as needed, and then close the database connection object when it is no longer needed (no database connection leaks).
  - b. Have as little code as possible (delegating functionality to reusable java classes), however, a JSP page is the only place where you can use JSP implicit objects such as input (request.getParameter), output (out.print), session (get/put logon information), and security (response.sendRedirect).
2. Your web application shall have a **model package** with 3 sub-packages, each sub-package named the same as one of your database tables (not assoc, not other). So, for example, if you had a database table named “student”, you shall have a student package under the model package.
3. Within each model sub-package shall be a **StringData** class, a **Validate** class, and a **DbMods** class.
4. The **StringData class** shall have one String property (can be public, getters/setters optional) for each field in the associated database table (even fields that are not of type VarChar in the database, including primary key and foreign key fields). In addition, the **StringData** class shall have an extra field for error message.
5. The **DbMods class** shall have an **insert method** that accepts a (user entered) **StringData** object, validates it and (if it passes validation) inserts it into the database. This method/class shall return a **StringData** object whose fields are either all filled with all empty string “” or it shall contain field and form level error messages.
6. The **validation code** shall be invoked by the insert method (as opposed to being “mixed” in with the insert code), so that the update method (in a future lab) will need to invoke the same (or similar) functionality.
7. Double check that your code that generated HTML tables for the list pages reside primarily in the **view** package,

## Labs Page (Blog)

- Your labs page shall always include a blog for each lab, explaining what you learned and linking to your work.

## Programming Style

- Adhere to what is listed in the section entitled “Requirements for All Labs and Project” (on the labs page).

## Homework submission

- After getting your code to work locally, publish it to cis-linux2 and test it.
- Submit a zip file of your whole project to blackboard.
- Make sure that you have a link from your labs page to a screen capture of your data model (that you created using mySqlWorkbench).

## Suggested Approach:

Complete the “Insert Lab Activity” [http://cis-linux2.temple.edu/~sallyk/cis3308/07\\_insert/insert\\_lab\\_activity.html](http://cis-linux2.temple.edu/~sallyk/cis3308/07_insert/insert_lab_activity.html) then repeat this approach for the other two insert pages.