CPRG352 - Web Application Programming

Fall 2021

Topic: JDBC data access using a Connection Pool

Problem : Manage a Shopping List using a Connection Pool

Create an application called **ShoppingList**. This application will use the **shoppinglist** database (creation SQL script provided with this lab) to manage shopping list items for a user.

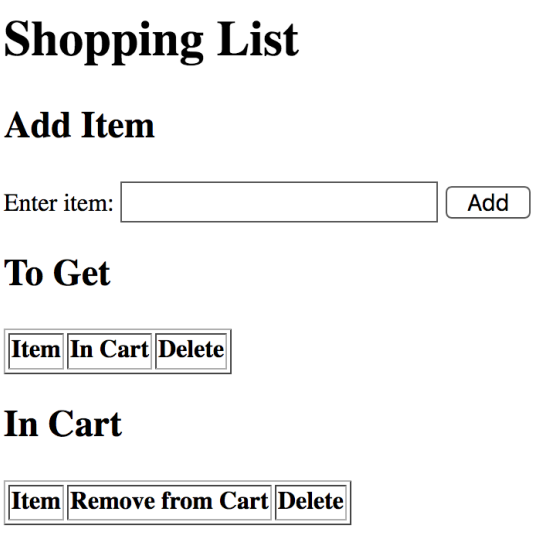
The functionality of the application should be as follows. Users can:

- Add new items to the shopping list. Initially they will not be in the cart when newly-added
- Move items into the cart (as they are gathered in the store)
- Move items from the cart back out again to the "not-in-cart" list
- Delete items from either the "in-cart" or "not-in-cart" lists

For each operation the database will be updated immediately. This application should **not** store state anywhere else, e.g. in session objects, in cookies, etc.

Sample Run

When first run, with an empty database, the application shows:



The "**Add Item**" form allows the user to enter a new shopping list item. Initially the item will be shown in the "**To Get**" list (for items not yet in the shoppers cart/basket).

If the user enters "***Apples***" and clicks on "**Add**" they should see:

Shopping List

Add Item

Enter item: Add

Item added

To Get

Item	In Cart	Delete
Apples	Add to Cart	Delete

In Cart

Item	Remove from Cart	Delete

The item is displayed in the "**To Get**" list, and the message "**Item added**" is shown just under the "**Add Item**" form.

If "***Bananas***" is then added the user sees:

Shopping List

Add Item

Enter item: Add
Item added

To Get

Item	In Cart	Delete
Apples	Add to Cart	Delete
Bananas	Add to Cart	Delete

In Cart

Item	Remove from Cart	Delete

Again, the "**Item added**" message is displayed under the form.

If the user then finds apples in the store and places some in his/her cart or basket s/he can tell the application this by clicking in the "**Add to Cart**" link beside "***Apples***" in the "**To Get**" list. This will move the apples entry into the "**In Cart**" list instead, e.g.:

Shopping List

Add Item

Enter item: Add
Item added to cart

To Get

Item	In Cart	Delete
Bananas	Add to Cart	Delete

In Cart

Item	Remove from Cart	Delete
Apples	Remove from Cart	Delete

In this case the message "**Item added to cart**" should be displayed under the form.

If the shopper then decides that s/he doesn't want these apples, but other ones instead then s/he can click on the "**Remove from Cart**" link beside the apples in the "**In Cart**" list to move that entry back into the "**To Get**" list (while also presumeably taking the unwanted apples out of the cart or basket also!), e.g.:

Shopping List

Add Item

Enter item: Add

Item removed from cart

To Get

Item	In Cart	Delete
Apples	Add to Cart	Delete
Bananas	Add to Cart	Delete

In Cart

Item	Remove from Cart	Delete

The message "**Item removed from cart**" should be displayed under the form also.

If the shopper wants to remove an item from either list s/he can click on the "**Delete**" link beside the item to remove, e.g. removing "**Bananas**" from the "**To Get**" list:

Shopping List

Add Item

Enter item: Add
Item deleted

To Get

Item	In Cart	Delete
Apples	Add to Cart	Delete

In Cart

Item	Remove from Cart	Delete

This time the message "**Item deleted**" should be shown under the form.

Application Requirements

The application should:

- Use a connection pool with the JNDI name **jdbc/shoppinglist** to get connections to the database as required
- Follow the "**front servlet**" pattern
- Perform all operations directly against the database as the user manipulates the data (use no other type of state management)
- Use a class like our typical **DBoperations** to do the actual data manipulation work (DB work should not be done directly in the controller)
- Use the singleton **ConnectionPool** class that we used in our lecture to provide database connections from the connection pool
- Use the **stored procedures** provided in the database to perform updates (*insert*, *update* and *delete* SQL statements) to manipulate the data in the database
- Use **prepared statements** to retrieve data from the database (***select*** SQL statements only)

Examine the database structure to see how you can use it to implement the functionality for this application.