Databases

Daniel R. Cender

Grand Canyon University: CST-235-O500

August 2, 2020

**All Source Code Retrievable At:** [**https://github.com/DanielCender/CST-235**](https://github.com/DanielCender/CST-235)

**Explanation**

The additions to code in this milestone add database access beans and methods.

Below is a simple breakdown of all the classes/views in the assignment. The only new methods were added to the FormController class.

**Classes**

assignment4

* + User.java (Managed Bean)
    - Methods: constructor, overloaded constructor, getters/setters
  + Order.java (POJO)
    - Methods: constructor, getters/setters
  + Orders.java (Managed Bean)
    - Methods: constructor, getters/setters
  + OrdersBusinessInterface.java (EJB interface)
    - Methods: test
  + OrdersBusinessService.java (EJB Implementation)
    - Methods: constructor, implements test
  + AnotherOrdersBusinessService.java (alternative EJB Implementation)
    - Methods: constructor, implements test
  + MyTimerService.java
    - Methods: constructor, setTimer, programmicTimer, scheduledTimeout
  + FormController.java (Managed Bean)
    - Methods: onSubmit, onFlash, getService, getAllOrders, insertOrder

**Pages**

assignment5

* + TestForm.xhtml
  + TestResponse.xhtml
  + TestResponse2.xhtml

**assignment5**

This assignment adds the use of database connections to store persistent data for the assignment JavaEE project.

The first step in setting up this milestone is to populate a PostgreSQL database with some starting records. The screenshot below shows the results of creating and pre-populating the PostgreSQL database with order records.

A screenshot of a social media post

Description automatically generated

The next step in this assignment required connecting successfully to the Postgres database with the appropriate JDBC driver and logging a success message.

A screenshot of a cell phone

Description automatically generated

With the database connection now intact, the next step involved querying and logging all orders in the testapp.Orders table. The resulting set items are iterated through and logged individually.

A screenshot of a cell phone

Description automatically generated

The last phase of this assignment required implementing a final data access method, insertOrder, which would add one item to the Orders table. The FormController now gets all orders before inserting one more. All orders are grabbed once more to verify a correct insertion. The screenshots below show the state of the Orders table before and after the insert statement executed.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated