Michael Weaver

CST – 235

Brandon Bass

11/19/17

**CLC-Mini-Project 2**

GitHub Link: https://github.com/battousairurik/CST-235

**Group Participation**

Michael Weaver – Only Group member

**Tools/Concepts/Techniques**

For the first part of the assignment the use of an interface with the EJB is required. The EJB is then instantiated in the JSP page and meant to be displayed to the web server. The WildFly server installed to my local device runs the server application.

**Notes for this assignment**

Finding out how to install the WildFly and JBoss server is rather difficult. Even after unzipping all the files from the website the servers do not run. It took me a few hours to figure out how to get the servers to run by installing them directly on Eclipse.

Even more frustrating is determining how to enable Javafx on Eclipse EE. This was infuriating and there was no tutorials or answers anywhere to solve this problem. Eventually I found the solution by searching e(fx)clipse tutorials for a good 2 hours.

**Assignment Approach**

1: Using EJB with JSP seems to be very tricky. Even when you follow the guidelines step by step you can still wind up with errors that prevent you from finishing a project, as is the case with my own work. From what I can tell, you can create a JSP and then call the EJB as needed for whatever effect you wish to have. In the case of my example I attempt to call an addition EJB as listed in the online example that I had found. Sadly, the JSP has an error that I have no idea how to fix and thusly the entire example cannot be completed.

2: Because EJB are portable it allows a single bean to be used for different applications, this can also apply when using a single bean across many different layers of a system. This ties into the Scalability of EJB, given that they have the capacity to be easily copied from machine to machine, allowing them to be utilized efficiently.

For this portion of the assignment I created a bean then instantiated that bean in two different client classes, then called each client to the console. Each client prints out that they are using the bean in their program. This is a very primitive example of how beans can be used in multiple locations.

**List of Classes, Methods, Variables, etc.**

Addition.java Class

* Default constructor
* Add method implemented from interface

AdditionRemote.java interface

* Add abstract method

Index.jsp server page

* Text in the body

Enterprise Bean

* Constructor
* Private string
* Public set and get methods for the bean

Client One

* Private enterprise bean
* Public constructor
* Public bean get method

Client Two

* Private enterprise bean
* Public constructor
* Public bean get method

Main

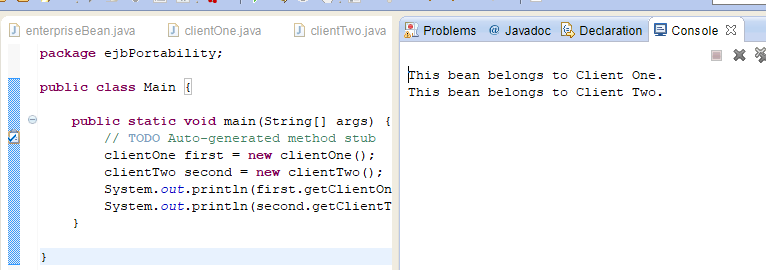
* Main method call to post example to console

**Screenshots**

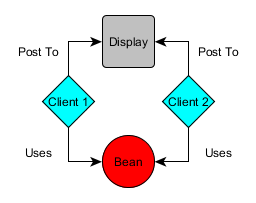
JSP and EJB Example



**Portability**



**Flowchart**



\*Two Clients utilizing the same bean to emulate Portability