**CLC-Mini-Project 2**

Mat Paschall

Github: https://github.com/Mpaschall/CST\_235\_MP\_2

Question 1

EJBs can be used with JSP either through direct calls from the JSP file or using servlets. EJBs can be called inside the JSP file by surrounding your Java code for the call with <% %> symbols. This allows the EJB methods and properties to be referenced. You can also call them using a servlet, which will then redirect to the JSP file. As I understand, using servlets to do the “heavy lifting” such as calling methods and performing logic is a good practice. You can then reference the bean properties and methods using <jsp: getProperty/setProperty /> tags. In my example, I use a session bean created in my modified assignment 1 project to multiply the amount of video space calculated in the servlet and using storage\_bean by the amount of videos stored. I call the method allStorage in the session bean from the servlet, and then set the property for multStorage. Then I used JSP tags in the JSP page to access this information and display the data. I chose a session bean for my EJB because these types of EJBs are used to perform logical functions. Entity beans are mostly used to communicate with databases, which we haven’t covered yet, and message-driven beans are used to receive messages. Thus, the most logical bean to use was the session bean. I have uploaded my code example to my github at the address above.

**classes**

Index.html

myServlet.java

displayStorage.jsp

**variables**

videoQuality

runTime

transRate

videoSpace

storage

multStorage

videos

**beans**

Storage\_Bean.java

SessionBean.java

**Methods**

allStorage()

Question 2

EJBs promote portability because existing beans can be used in new applications. This allows great flexibility when creating new applications, as new beans don’t have to be created with every new application. As long as the application uses the standard APIs, these applications can run on any compliant Java EE server. Existing EJBs can be used simply by placing the EJB file in the source package and then importing them at the start of the application using “import EJBName”. EJBs also support scalability because it makes it easy to distribute components. The components of the application may need to be distributed among multiple additional machines as the user base grows. EJBs can also run on a variety of different machines and the location of these EJBs remains transparent to the clients. This can help tremendously when expanding the usage of your application.

import EJBName;

public class UseOldBean{

public static void main(String args){

EJBName ejb = new EJBName();

ejb.doStuff();

}

}

References

https://stackoverflow.com/questions/2460048/difference-between-java-bean-and-enterprise-java-beans

https://docs.oracle.com/cd/E19798-01/821-1841/gijsz/index.html