**Application1: Shopping Cart Web Application**

**Description:**

* This is a simple shopping cart application where different users can add and remove items from the shop to their cart during a session.
* The motivation of this application is to test different session beans and entity beans along with different persistence methodology rather than implementing a heavy scale E-Commerce application.
* This shopping cart application is implemented using java server face (jsf) and enterprise java bean (ejb).
* In this application **Session Bean** (both **Stateless** and **Stateful** session) and **Entity Bean** (a derby database) is examined.
* **The structure of this web based application** corresponds to the following **tiers**:
  + *Web Tier***:**  Contains the logic of the application and runs on a Java EE server. Web Tier is represented by the web module and contains JSFs that access the business logic in the EJB module.
  + *Business Tier:* Contains the business logic of the application and runs on Java EE Server. Business Tier means the EJB module. The EJB module contains the code that handles requests from the Web Tier clients and manages transactions and how objects are persisted to the database.
  + *EIS Tier:* Contains the persistent storage layer of the application. EIS tier is represented by the database where the messages are stored.
  + *Client Tier:* The application is then typically accessed from the **Client Tier.** The Client Tier is the environment where the client is run and is often a web browser on a user's local system.
* I have tested this a single machine to host the server, database and view the web pages. But it is possible to deploy the Web Tier and Business Tier in different Java EE servers that are hosted on different machines.

**Software / Resources used:**

* Development Environment: Mac OS X
* JDK 1.6
* Netbeans IDE 6.9
* J2EE 6
* Glassfish Server version 3.1.2
* Derby Database

**Initial Configuration:**

* Download and install JDK
  + http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html
* Install NetBeans (6.9)
  + https://netbeans.org/downloads/
* Install Glassfish Server
  + Download Glassfish: https://glassfish.java.net/downloads/3.1.2-final.html
  + Unzip to any directory
  + Add both the Glassfish3/bin and Glassfish3/Glassfish bin to the PATH variable
  + Command to start and stop Glassfish (I have started from Netbeans IDE)
    - To Start: Terminal -> asadmin start-domain domain1
    - To Stop: Terminal -> asadmin stop-domain domain1
  + Confirm the application server is up by pointing your browser to [http://localhost:8080](http://localhost:8080/)
  + Reference: <https://weblogs.java.net/blog/2005/11/26/running-glassfish-mac-os-x>
* Install MySQL (Optional)
  + Uninstall all the previous broken versions:
    - https://coderwall.com/p/os6woq
  + <http://dev.mysql.com/doc/refman/5.0/en/macosx-installation.html>
  + If change of root password required:
    - http://coolestguidesontheplanet.com/how-to-change-the-mysql-root-password/
* Integrate MySQL with Glassfish (Optional)

http://dev.mysql.com/doc/connector-j/en/connector-j-usagenotes-glassfish-config.html

**How to Run the Application:**

* Add and Start Glassfish server from NetBeans
  + Click the Services tab -> Expand the Servers node.
  + Right Click Server -> Add Server -> Glassfish Server -> Browser the GlassfishX directory previously configured -> Finish
  + Right-click the GlassFish Server instance and select Start.
* Open the Projects
  + File -> Open Project -> Select the folder ShoppingCart
* Create a Derby Database
  + Click the Services tab -> Expand the Database node -> Java DB - > Create Database -> Database name: shoppingCart, username: root, password: password and press OK
* Open the Projects
  + File -> Open Project -> Select the folder ShoppingCart
* Right Click created Database -> Connect
* Right Click created Database -> Execute -> paste the sql commands from products.sql (under ‘Server Resources’ tab in the ShoppingCart project)
* Make sure Glassfish is running: Services Tab -> Right-click the GlassFish Server instance -> Start /Restart
* Run the Application: Right click the Application ShoppingCart -> Run
  + It should open the initial servlet in the browser
  + http://localhost:8080/ShoppingCart/index.xhtml
* Some of the sample pages of the applications can be found in ShoppingCart/webpages

**How to Run Test Cases Using JMeter:**

* JMeter is used to test this application.
* Install Load Generator and JMeter Plugin in Netbeans
  + Tools -> Plugins -> Available Plugins Tab -> Search for Load -> Select Load Generator and JMeter -> Install
* Right Click Load Generator Scripts/useCaseTest.jmx -> External Edit -> Run -> Start
  + All the examined test cases can be viewed from ‘Recording Controller’ section of ShoppingCartUsers tab
  + Used Proxy server for some test cases
  + Tested with different combination of concurrent users
  + Details of each test cases can be viewed from Summary report section
  + Reference Link: <https://netbeans.org/kb/docs/javaee/ecommerce/test-profile.html#test>
* Some of the Session Bean classes also tested with junit
* To install JUnit plugin: Toos -> Plugins manager -> Available plugins Tab -> Install the JUnit plugin.
* To check the junit test cases:
  + Right click ShoppingCart Project -> Test
* Reference Link: <https://netbeans.org/kb/docs/javaee/javaee-entapp-junit.html>

**Application2: NewsPortal Application with EJB 3.1**

**Description:**

* This is a simple multi-tiered message board application where different users can post and retrieve news from a database.
* The main motivation of developing this simple application is to explore the **message bean**.
* This application is heavily inspired from one of the NetBeans EJB sample example:
  + https://netbeans.org/kb/docs/javaee/javaee-entapp-ejb.html#intro
* This application contains a EJB Module and a Web Module
* EJB Module contains an **entity bean** class (persist message into a database), a **session facade bean** for the entity class and a **message-driven bean**
* Web Module contains two servlet for posting news and displaying posted news and a singleton session bean that count number of currently active users of the message board.
* **The structure of this sample application** corresponds to the following **tiers**:
  + *Web Tier***:**  Contains the logic of the application and runs on a Java EE server. Web Tier is represented by the web module and contains servlets that access the business logic in the EJB module.
  + *Business Tier:* Contains the business logic of the application and runs on Java EE Server. Business Tier means the EJB module. The EJB module contains the code that handles requests from the Web Tier clients and manages transactions and how objects are persisted to the database.
  + *EIS Tier:* Contains the persistent storage layer of the application. EIS tier is represented by the database where the messages are stored.
  + *Client Tier:* The application is then typically accessed from the **Client Tier.** The Client Tier is the environment where the client is run and is often a web browser on a user's local system.
* I have tested this a single machine to host the server, database and view the servlets. But it is possible to deploy the Web Tier and Business Tier in different Java EE servers that are hosted on different machines.

**Software / Resources used and Initial Configuration:**

* Same as Shopping Cart Application (Please see above)

**How to Run the Application:**

* Add and Start Glassfish server from NetBeans
  + Click the Services tab -> Expand the Servers node.
  + Right Click Server -> Add Server -> Glassfish Server -> Browser the GlassfishX directory previously configured -> Finish
  + Right-click the GlassFish Server instance and select Start.
* Create a Derby Database (should be there by default)
  + Click the Services tab -> Expand the Database node -> Java DB - > Create Database -> Database name: sample, leave username/password blank -> OK
  + Right Click the sample Database -> Connect
* Open the Projects
  + File -> Open Project -> Select the folder NewsPortal
* Register JMS resources in the Glassfish server.
  + Open the admin console of glassfish server by right-click the application server node and choose View Admin Console. Or use <http://localhost:4848/>
  + Expand the Resources node and JMS Resources node in the left frame.
  + JMS Resources -> Destination Resources -> New -> Enter JNDI name: *jms/NewMessage* and Physical Destination Name: *NewMessage* and select type: *javax.jms.Queue*
  + JMS Resources -> Connection Factories -> New -> Enter JNDI name: *jms/NewMessageFactory* and select type: *javax.jms.QueueConnectionFactory*
* Make sure Glassfish is running: Services Tab -> Right-click the GlassFish Server instance -> Start /Restart
* Run the Application: Right click the Application NewsPortal -> Run
  + It should open the initial servlet in the browser
  + OR copy and paste the following in any browser: http://localhost:8080/NewsPortal-war/News
* Some of the sample pages of the applications can be found in NewsPortal/webpages

**How to Run Test Cases Using JMeter:**

* JMeter is also used to test this application.
* Install Load Generator and JMeter Plugin in Netbeans
  + Tools -> Plugins -> Available Plugins -> Search for Load -> Select Load Generator and JMeter -> Install
* Right Click Load Generator Scripts/useCaseTest.jmx -> External Edit -> Run -> Start
  + Please check the previoAll the examined test cases can be viewed from ‘Recording Controller’ section of NewsPortalUsers tab
  + Used Proxy server for some test cases
  + Tested with different combination of concurrent users
  + Details of each test cases can be viewed from Summary report section
  + Reference Link: <https://netbeans.org/kb/docs/javaee/ecommerce/test-profile.html#test>

Md Abu Naser Bikas

UIN: 651644268

mbikas2@uic.edu