**ASSIGNMENT NO. 10**

**TITLE**

Web Application using EJB

**OBJECTIVES**

1. Understand about basic concepts of java beans.

2. Understand the basic functionalities of JSP, HTML.

3. Having the knowledge of JBOSS server to deploy web application.

**PROBLEM STATEMENTS**

Design, Develop & Deploy web application using EJB.

**OUTCOMES**

*Students will be able to,*

1. Develop a dynamic webpage using Java Beans, HTML and JSP.

2. To understand the concepts and method of web based applications development Process using EJB.

3. Create a simple EJB 3 stateless session bean and a local Java application client which will call/invoke the bean to develop for addition of two numbers.

**SOFTWARE NEEDED**

1. Ubuntu 64 bit / Windows 7.

2. JDK 7 (Java SE 7)

3. EJB 3.0 (stateless session bean)

4. Eclipse luna

5. JBoss Application Server (AS) 7.1.1

**THEORY – CONCEPT**

**Java Beans :**

J2EE application container contains the components that can be used by the clients for executing the business logic .These components are known as Enterprise Java Beans (EJB) .

J2EE platform has component based architecture to provide multi-tiered, distributed and highly transactional features to enterprise level applications.

EJB mainly contains the business logic & business data. EJB component is an EJB class. It is a java class written by EJB developer & this class implements business logic.

It is used for developing very much scalable and robust enterprise level applications to be deployed Application Server such as JBOSS, Web Logic etc.

EJB 3.0 is being a large shift from EJB 2.0 and makes development of EJB based applications relatively easy.

**Features of EJBs:**

Some of the features of an application server include the following:

• **Client Communication:** The client, which is often a user interface, must be able to call the methods of objects on the application server via agreed-upon protocols.

• **State Management:** You'll recall our discussions on this topic in the context of JSP

(JavaServer Pages) and servlet development back in Chapter 6.

• **Transaction Management:** Some operations, for example, when updating data, must occur as a unit of work. If one update fails, they all should fail.

• **Database Connection Management:** An application server must connect to a database, often using pools of database connections for optimizing resources.

• **User Authentication and Role-Based Authorization:** Users of an application must often log in for security purposes. The functionality of an application to which a user is allowed access is often based on the role associated with a user ID.

• **Asynchronous Messaging:** Applications often need to communicate with other systems in an asynchronous manner; that is, without waiting for the other system to respond. This requires an underlying messaging system that provides guaranteed delivery of these asynchronous messages.

• **Application Server Administration:** Application servers must be administered. For example, they need to be monitored and tuned.

**Types of Enterprise Java Beans (EJB):**

There are three types of Enterprise Java Beans namely:

1. Session Beans

2. Entity Beans

3. Message driven beans

**Session Beans**

• Session beans are intended to allow the application author to easily implement portions of application code in middleware and to simplify access to this code.

• Represents a single client inside the server

• The client calls the session bean to invoke methods of an application on the server

• Perform works for its client, hiding the complexity of interaction with other objects the server

• Is not shared

• Is not persistent

When the client stops the session, the bean can be assigned to another client from the server

Session beans are divided into two types:

**1. Stateless Session Bean:**

Stateless Session Bean is intended to be simple and “light weight” components. The client, thereby making the server highly scalable, if required, maintains any state. Since no state is maintained in this bean type, stateless session beans are not tied to any specific client, hence any available instance of a stateless session bean can be used to service a client.

• Values only for the duration of the single invocation

• Except during method invocation, all instances of stateless session bean are equivalent

**Stateless Session Bean’s Life Cycle:**

• The client invoke the create method

• The EJB container :Instantiates the bean

Invokes the setSessionContext Invokes ejbCreate

• The bean is ready

• While in the ready state client may invoke a business method: A client may invoke the remove method and the container calls the bean's ejbRemove method. It’s never passivate .



**Figure.: Stateless Session Bean’s Life Cycle**

**2. Stateful Session Bean:**

State ful Session Bean provides easy and transparent state management on the server side.

Because state is maintained in this bean type, the application server manages client/bean pairs.

Stateful session beans can access persistent resources on behalf of the client, but unlike entity beans, they do not actually represent the data.

**Stateful Session Beans Life Cycle:**

• The client invoke the create method

• The EJB container: Instantiates the bean

Invokes the setSessionContext Invokes ejbCreate

• The bean is ready

• While in the ready state

• c ontainer may *passivate* the bean moving it from memory to secondary storage

• A client may invoke a business method

EJB container may activate a bean, moving it back to the ready stage, and then calls the bean's

ejbActivate method.

A client may invoke the remove method and the container calls the bean's ejbRemove method

**Difference between Stateless and State Full EJB Are as Follows**

**Stateless:**

1. Normally data members are not put in stateless session bean

2. Stateless beans are pooled

3. No effort for keeping client specific data

4. No Activation/Passivation in stateless session bean

**Stateful:**

1. Data members that represent state are present in stateleful session bean

2. Stateful beans are cached

3. Setting the tag idle-timeout-seconds determines how long data is maintained in stateful session bean

4. Activation – Passivation used

**An Entity Bean**

• An entity bean is an object representation of persistent data maintained in a permanent data store such as a database. A primary key identifies each instance of an entity bean. Entity beans are transactional and are recoverable in the event of a system crash.

• Entity beans are representations of explicit data or collections of data, such as a row in a relational database. Entity bean methods provide procedures for acting on the data representation of the bean. An entity bean is persistent and survives if its data remains in the database.

• An entity bean can implement either bean-managed or container-managed persistence. In the case of bean-managed persistence, the implementer of an entity bean stores and retrieves the information managed by the bean through direct database calls. The bean may utilize either

Java Database Connectivity (JDBC) or SQL-Java (SQLJ) for this method.

• In the case of container-managed persistence, the container provider may implement access to the database using standard APIs. The container provider can offer tools to map instance variables of an entity bean to calls to an underlying database. The container saves the data.

There is no code in the bean for access the database. The container handles all database access required for the bean which create links between beans are created using a structure called abstract schema.

**DESIGN / EXECUTION STEPS**

Following steps are used to Create and Execute web applications,

1. Design EJB project.

2. Start JBOSS & deploy it on JBOSS server.

3. Design html and jsp files with an extension of.html and .jsp

4. Run the application in browser and get the result

**CONCLUSION / ANALYSIS**

Hence, we have created a simple EJB 3 stateless session bean and a local Java application client which will call/invoke the bean to develop for performing addition of two numbers.

**ORAL QUESTIONS**

1. What is EJB?

2. What is JSP?

3. What is the purpose of JBOSS?

4. What is the syntax of JSP?

5. How to deploy java beans to server?