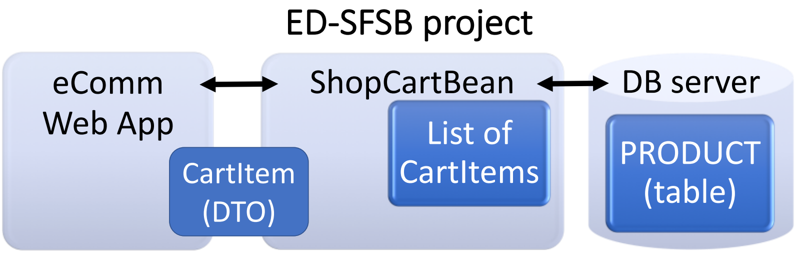
To finish the lab, you may need the following software:
1. NetBeans IDE version 12.2 or version 12.5
2. JDK version 1.8.0 (jdk1.8.0_202, or later)
3. GlassFish Server Open Source Edition version 5.1.0

Aim
Programming the EJB Stateful Session Bean “`ShopCartBean`” (a shopping cart holding a collection of “`CartItem`”s) and its client application.



Thin client (UI)
- Get input from users
- Display output to users

ShopCartBean (SFSB holding CartItems)
- Perform shopping cart function

CartItem (DTO)
- For remote communications and something else**

** E.G. Once order is confirmed, CartItem needs to consolidate with PRODUCT (left as further exercises)

DB server
- Actual DB functions

PRODUCT (database table)
- Store information

Figure 1 A rough design of an eComm Web App

In this Lab, we will focus only on how to implement the Stateful Session Bean.

Overview of Lab Tasks
In this Lab, you will learn to perform the following tasks in programming a stateful session bean:
- LT1. Set up a Class Library for the Remote Interface of the stateful session bean using NetBeans (Recap)
- LT2. Program the data transfer object class to be used in the Stateful session bean
- LT3. Set up Enterprise Application project using NetBeans
- LT4. Program an EJB Stateful Session Bean class
- LT5. Program the Remote Interface class of an EJB Stateful Session Bean
- LT6. Program a client application that calls an EJB Stateful Session Bean
- LT7. Deploy the EJB Stateful Session Bean using NetBeans
- LT8. Run the Enterprise Application Client using NetBeans (Recap)

Lab Tasks

This Lab should be run on MS Windows Platform

LT1. Create a new Java Class Library Project in NetBeans called "`ED-SFSB-RI`" to hold the remote interface of the stateful session bean
- LT1.1 Select "File" > "New Project"
- LT1.2 Choose " Java with Ant " > "Java Class Library", then click "Next"
- LT1.3 Enter "`ED-SFSB-RI`" in the "Project Name" field
- LT1.4 Click “Finish”
  Note: NetBeans has just created the project.

LT2. In this Lab task, you will program the DTO class “`CartItem`” in the “`ED-SFSB-RI`” project
- LT2.1 Select "ED-SFSB-RI"
- LT2.2 Right click and select "New" > "Java Class..."
- LT2.3 Enter the following information

a. Class Name: `CartItem`

b. Package: `dto`

LT2.4 Click "Finish"

LT2.5 Copy and paste the following code segment into CartItem.java

```
public class CartItem implements Serializable {

    private String itemId;
    private String description;
    private double unitPrice;
    private int quantity;

    public CartItem(String id, String desc, double price, int qty) {
        itemId = id;
        description = desc;
        unitPrice = price;
        quantity = qty;
    }
}
```

Remember to fix the "import statements" as well as the "format" of the code, and finally, save the file

LT2.6 Generate the Getter and Setter of "`CartItem.java`"

a. Right click while the mouse is in "CartItem.java"

b. Select "Insert Code..." > "Getter and Setter..."

c. Check all items in "Generate Getters and Setters" window

d. Click "Generate"

e. Save the file

Note: NetBeans now generates all getters and setters.

LT3. Create a new Enterprise Application Project in NetBeans called "`ED-SFSB`" (Recap)

LT3.1 Select "File" > "New Project"

LT3.2 Choose "Java with Ant" > "Java Enterprise" in the "Categories" selection area
Choose "**Enterprise Application**" in the "Projects" selection area
Click the "Next" button

LT3.3 Enter "`ED-SFSB`" in the Project Name field, then click "Next"

LT3.4 Use the default server "GlassFish 5.1.0" and Java EE version "Java EE 8" (i.e. Choose "Java EE 8" in the "Java EE Version" or Choose Jakarta EE 8 in the "Java EE Version")

LT3.5 Make sure the following check boxes are checked

a. "Create EJB Module"

b. "Create Web Application Module"

LT3.6 Click "Finish"

*Note: There are three new "projects" in the "Projects" Windows. They are

a. ED-SFSB (Enterprise Application Project)

b. ED-SFSB-ejb (EJB Module Project)

c. ED-SFSB-war (Web Application Project)

LT4. In this Lab task, you will learn how to program the Stateful Session Bean class, called "`ShopCartBean`"

LT4.1 Select the "ED-SFSB-ejb" project

LT4.2 Right click the mouse and select "New" > "Session Beans…"[1]

Note: If you cannot find "Session Beans…", select "Other…". Then, in the "New File" window, choose "Enterprise JavaBeans" in the "Categories:" selection box, then choose "Session Bean" in the "File Types:" selection box. Then, click "Next".

LT4.3 Enter "`ShopCartBean`" in the "EJB Name" field

LT4.4 Enter "`ed.sfsb`" in the "Package" field

LT4.5 Select "Stateful" for "Session Type"

[1] We do not use "Session Beans from Entity Classes…" option due to some unknown NetBeans issues.

LT4.6 Check "Remote" in “Create Interface” field

Note: When "Remote" is checked, NetBeans asks for a java class library project (in this case, select "ED-SFSB-RI ", the one we just created in LT1)

LT4.7 Click “Finish”

Note: NetBeans will create a package called “ed.sfsb” in the “Source Packages” under the “ED-SFSB-ejb” project. Furthermore, under this package, there is a file “`ShopCartBean.java`”. Finally, the bean class “`ShopCartBean.java`” is shown in the editor window.

Note: NetBeans also creates the remote interface class “`ShopCartBeanRemote.java`” under the package "`ed.sfsb` " in the "ED-SFSB-RI" project.

Now, we are going to add one instance variable to “`ShopCartBean`” class. The values of these variables depending on the individual clients and hence are considered as the “conversational states” of the stateful session bean ShopCartBean with respect to the clients. Basically, this is the ShopCartBean object for the client holding the items they put in the Shopping Cart, i.e. a list of CartItems. Hence, the ShopCartBean class (emulating the shopping cart) is a collection of CartItems, the items being “placed” in the shopping cart.

LT4.8 Add the Java EE 8 API Library in the “ED-SFSB-ejb” sub-project’s Libraries.

LT4.9 Copy and paste the following code segment to the stateful session bean “ShopCartBean”

```
private ArrayList<CartItem> cart;

public ShopCartBean() {
    cart = new ArrayList<CartItem>();
}
```

LT4.10 Check the EJB’s configuration file (This step is to fix some unknow bugs in NetBeans)

a. Select the ED-JEE-DTO-ejb project

b. Expand on "Configuration Files" tag

c. Open ejb-jar.xml file in an editor window

d. Check the content in ejb-jar.xml

It should look something like the following, especially the version number of EJB. If it is not, copy and paste the following code segment to the file.

ejb-jar.xml [EJB configuration file]

```
<?xml version="1.0" encoding="UTF-8"?>
<ejb-jar version="3.2"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
  http://xmlns.jcp.org/xml/ns/javaee/ejb-jar_3_2.xsd">
  <display-name> ED-SFSB-ejb</display-name>
  <enterprise-beans>
    <session>
      <ejb-name>ShopCartBean</ejb-name>
      <remote>ed.sfsb.ShopCartBeanRemote</remote>
      <ejb-class>ed.sfsb.ShopCartBean</ejb-class>
      <session-type>Stateful</session-type>
      <transaction-type>Container</transaction-type>
    </session>
  </enterprise-beans>
</ejb-jar>
```

LT4.11 Add the following business method to the stateful session bean “ShopCartBean”

```
public boolean add(CartItem cartIte)
```

which adds a CartItem object to the shopping cart, ShopCartBean. It also returns a boolean indicating whether the item can be added to the shopping cart.

Note: The steps are similar to those in the Stateless Session Bean Lab.

See LT6 in Lab_04 for creating a business method.
NetBeans creates the method in the “ShopCartBean” class and a declaration in the “ShopCartBeanRemote” interface class.

LT4.12 Copy and paste the following code segment in the "`add()`" method

```
@Override
public boolean add(CartItem cartItem) {
    boolean result = false;

    try {
        result = cart.add(cartItem);
    } catch (Exception ex) {
    }

    return result;
}
```

Note: This code segment unconditionally adds a cartItem to the shopping cart. This is just for demo purposes in the Lab. In real case, you need to program the logic to check at least whether (1) the item has been added to the cart previously and update the quantity accordingly and (2) you have enough stock for the item before actually putting it in the shopping cart. I will leave the real coding to you as extension in the portfolio tasks.

Note: Remember to fix the imports and format of the code.

LT4.13 Add the business method (see **LT6 in Lab_04 for creating a business method)**

```
public ArrayList<CartItem> getCart()
```

which returns the `cart` to the client. Tha is:

```
@Override
public ArrayList<CartItem> getCart() {
      return cart;
}
```

LT4.14 Copy and paste the following method in the stateful session bean

```
@Remove
public void remove() {
    cart = null;
}
```

Note: GlassFish server will call this method when it decides to “passivate” a stateful bean instance or destroy the stateful bean instance.

Note: Such method is considered as a destruction method.

LT5. In this Lab task, you will learn how to write the Remote interface class
If you used “Insert Code…” > “Add Business Method…” in LT4, all business methods’ signatures will be added to the corresponding Remote Interface class. Hence, no need to do anything.

LT6. In this Lab task, you will learn how to write an application client to access the Stateful Session Bean. Basically, the steps are the same as those for the application client to access a stateless session bean.

LT6.1 Create the " `ED-SFSB-app-client`" Enterprise Application Client project

a. Select “File” > “New Project”
b. Choose "Java with Ant" > "Java Enterprise" in the "Categories" selection area
c. Choose "**Enterprise Application Client**" in the "Projects" selection area
d. Click “Next”
e. In the "New Enterprise Application Client" window, do the following:
   i. Enter “`ED-SFSB-app-client`” in the "Project Name" text field

ii. Click the **Browse…** button

iii. In the "Select Project Location" window, do the following
- Browse to the folder that contains the ED-SFSB Enterprise Application project
- Click the **Open** button

iv. Check the "Project Location" and "Project Folder" text fields have been updated properly

v. Click the **Next >** button.

f. In the "New Enterprise Application Client" window, do the following

i. Select ED-SFSB in the "Add to Enterprise Application" combo box

ii. Choose the default server GlassFish 5.1.0 in the "Server" combo box

iii. Choose the Java EE 8 in the "Java EE Version" combo box

iv. Change the "Main class" text field to "`edsfsb.ShopCartAppClient`"

v. Click the **Finish** button

Note: NetBeans now creates the ED-SFSB-app-client (Enterprise Application Client Project) project.

LT6.2 Now, we need to add a reference to the Stateful session bean in the application client "`CartClient.java`" class

a. In the Projects window, select "`ED-SFSB-app-client`" project

b. Expand "ED-SFSB-app-client" project > "Source Packages" > "edsfsb" > "`ShopCartAppClient.java`"

c. Open "`ShopCartAppClient.java`" in the editor window

d. Locate the mouse within the "`ShopCartAppClient`" class

e. Right click the mouse and select "Insert Code..." > "Call Enterprise Bean…"

f. In the "Call Enterprise Bean" dialog window, expand on "ED-SFSB-ejb" node and select "ShopCartBean"

g. Select "Remote" in the "Referenced Interface"

h. Click "OK"

Note 1: NetBeans will create the following annotation for you "`@EJB private static ShopCartBeanRemote shopCartBean;`" (This actually creates a remote reference to the EJB Stateful Session Bean.)

Note 2: NetBeans also adds the "ED-SFSB-RI.jar" class library file as a project library.

i. Add the Java EE 8 API Library in the Application Client sub-project's Libraries

LT6.3 Rename "`shopCartBean`" to "`shopCart`" (just to save some typing. By the way, you are not having the actual "bean" in the client!)

LT6.4 Copy and paste the following segment of code to "`ShopCartAppClient.java`"

```
public ShopCartAppClient() {
}

public static void main(String[] args) {
    ShopCartAppClient appClient = new ShopCartAppClient();
    // show that the shopCart is empty
    appClient.displayCart();
    // assuming they are selected by the user
    CartItem item1 = new CartItem("000001", "Intel Core i7 CPU", 349.99, 2);
    CartItem item2 = new CartItem("000002", "Intel SSD 512GB", 299.99, 3);
    appClient.addCart(item1);
    appClient.displayCart();
    appClient.addCart(item2);
    appClient.displayCart();
}

public void addCart(CartItem item) {
    System.out.println("Adding item " + item.getDescription() + " to cart");
    if (shopCart.add(item)) {
        System.out.println("Your order of " + item.getQuantity()
            + " " + item.getDescription() + " has been added.");
    } else {
        System.out.println("Sorry, your order of " + item.getQuantity() + " "
            + item.getDescription() + " cannot be added due to low stock.");
    }
}

public void displayCart() {
    ArrayList<CartItem> ciList = shopCart.getCart();
    if (ciList.isEmpty()) {
        System.out.println("The shopping cart is empty!");
        return;
    }
    System.out.println("Your shopping cart has the following items:");
    double total = 0.0;
    for (CartItem ci : ciList) {
        double unitPrice = ci.getUnitPrice();
        double quantity = ci.getQuantity();
        double subTotal = unitPrice * quantity;
        System.out.println("Item: " + ci.getDescription()
            + "\tUnit Price: " + ci.getUnitPrice()
            + "\tQuantity: " + ci.getQuantity()
            + "\tSub-Total: " + subTotal);
        total = total + subTotal;
    }
    System.out.println("---");
    System.out.println("Total price: " + total);
    System.out.println("----End of Shopping Cart---");
}
```

Remember to fix the "import statements" as well as the "format" of the code, and finally, save the file

LT7. Deploy the enterprise application "`ED-SFSB`" project

LT7.1 Select "`ED-SFSB`" enterprise application project in the "Projects" window

LT7.2 Right click your mouse and select "Deploy" (You may want to select "Clean and Build" first to clean up any mess that has been done.)

LT8. Run the enterprise application client "`ED-SFSB-app-client`" project

LT8.1 Select "`ED-SFSB-app-client`" enterprise application client project in the "Projects" window

LT8.2 Right click the mouse and select "Run"
Now, observe the output of the client.

Note: Alternatively, you can expand and select "`ShopCartAppClient.java`" and select "Run File" after right clicking the mouse.

Finally, Take several deep breaths and relax.