Cyrus Edgren 101606991
Tutor: Wei Lai

6.1P

Task 1.

The lab was completed and the following output was generated:

```
post-run-deploy:
run-deploy:
Copying 1 file to D:\Repos\Secure_Scalable_Software\6.1P\ED-SFSB-app-client\dist
Copying 2 files to D:\Repos\Secure_Scalable_Software\6.1P\ED-SFSB-app-client\dist\ED-SFSB-app-clientClient
Warning: D:\Repos\Secure_Scalable_Software\6.1P\ED-SFSB-app-client\dist\gfdeploy\ED-SFSB-app-client does not exist.
The shopping cart is empty!
Adding item Intel Core i7 CPU to cart
Your order of 2 Intel Core i7 CPU has been added.
Your shopping cart has the following items:
Item: Intel Core i7 CPU Unit Price: 349.99      Quantity: 2     Sub-Total: 699.98
---
Total price: 699.98
----End of Shopping Cart---
Adding item Intel SSD 512GB to cart
Your order of 3 Intel SSD 512GB has been added.
Your shopping cart has the following items:
Item: Intel Core i7 CPU Unit Price: 349.99      Quantity: 2     Sub-Total: 699.98
Item: Intel SSD 512GB   Unit Price: 299.99      Quantity: 3     Sub-Total: 899.97
---
Total price: 1599.95
----End of Shopping Cart---
run:
BUILD SUCCESSFUL (total time: 11 seconds)
```

Task 2.

The CRUD operations were added to ShopCartBean.java.

addCartItem()

```java
@Override
public boolean addCartData(CartItem cartItem) {
    boolean itemInCart = false;
    for (CartItem item : cart) {
        if (item.getItemId().equalsIgnoreCase(cartItem.getItemId())) {
            itemInCart = true;
            item.setQuantity(item.getQuantity() + cartItem.getQuantity());
            return true;
        }
    }

    if (!itemInCart && !cartItem.getItemId().equalsIgnoreCase("")) {
        try{
            cart.add(cartItem);
            return true;
        } catch (Exception ex) {
            return false;
        }
    }

    return false;
}
```

Cyrus Edgren 101606991
Tutor: Wei Lai

deleteCartData()

```
@Override
public boolean deleteCartItem(String itemId) {
    boolean itemInCart = false;
    CartItem itemToRemove = null;
    for (CartItem item : cart) {
        if (item.getItemId().equalsIgnoreCase(itemId)) {
            itemInCart = true;
            itemToRemove = item;
        }
    }

    if (itemInCart) {
        try {
            cart.remove(itemToRemove);
            return true;
        } catch (Exception ex) {
            return false;
        }

    }
    return false;
}
```

updateCartData()

```
@Override
public boolean updateCartItem(CartItem cartItem) {
    boolean itemInCart = false;
    CartItem itemToUpdate = null;
    for (CartItem item : cart) {
        if (item.getItemId().equalsIgnoreCase(cartItem.getItemId())) {
            itemInCart = true;
            itemToUpdate = item;
        }
    }

    if (itemInCart) {
        try {
            cart.remove(itemToUpdate);
            cart.add(cartItem);
            return true;
        } catch (Exception ex) {
            return false;
        }
    }

    return false;
}
```

Cyrus Edgren 101606991
Tutor: Wei Lai

Task 3/4.

A simple console menu was developed to test the functionality of the CRUD operations.

Adding an item:

```
Please choose what operation you would like to perform:
1: Add
2: Update
3: Delete
4: Display Cart
5: Exit

1

Please enter the id of the item:
001

Please enter the description of the item:
Nvidia RTX3070

Please enter the price of the item:
1300.00

Please enter the quantity of the item:
2
Adding item Nvidia RTX3070 to cart
Your order of 2 Nvidia RTX3070 has been added.
Your shopping cart has the following items:
Item: Nvidia RTX3070    Unit Price: 1300.0      Quantity: 2     Sub-Total: 2600.0
---
Total price: 2600.0
----End of Shopping Cart---
```

Adding the same item again:

```
Please choose what operation you would like to perform:
1: Add
2: Update
3: Delete
4: Display Cart
5: Exit

1

Please enter the id of the item:
001

Please enter the description of the item:
Nvidia RTX3070

Please enter the price of the item:
1300.00

Please enter the quantity of the item:
3
Adding item Nvidia RTX3070 to cart
Your order of 3 Nvidia RTX3070 has been added.
Your shopping cart has the following items:
Item: Nvidia RTX3070    Unit Price: 1300.0      Quantity: 5     Sub-Total: 6500.0
---
Total price: 6500.0
----End of Shopping Cart---
```

Cyrus Edgren 101606991
Tutor: Wei Lai

Adding a null item:

```
Please choose what operation you would like to perform:
1: Add
2: Update
3: Delete
4: Display Cart
5: Exit

1

Please enter the id of the item:


Please enter the description of the item:


Please enter the price of the item:


Please enter the quantity of the item:

Adding item  to cart
Sorry, your order of 5  cannot be added due to low stock.
Your shopping cart has the following items:
Item: Nvidia RTX 3070   Unit Price: 1300.0      Quantity: 5     Sub-Total: 6500.0
---
Total price: 6500.0
----End of Shopping Cart---
```

Updating an item:

```
Please choose what operation you would like to perform:
1: Add
2: Update
3: Delete
4: Display Cart
5: Exit

2

Please enter the id of the item:
001

Please enter the description of the item:
Nvidia RTX3060

Please enter the price of the item:
800.0

Please enter the quantity of the item:
3
Updating item Nvidia RTX3060 in cart
Your order of 3 Nvidia RTX3060 has been updated.
Your shopping cart has the following items:
Item: Nvidia RTX3060    Unit Price: 800.0       Quantity: 3     Sub-Total: 2400.0
---
Total price: 2400.0
----End of Shopping Cart---
```

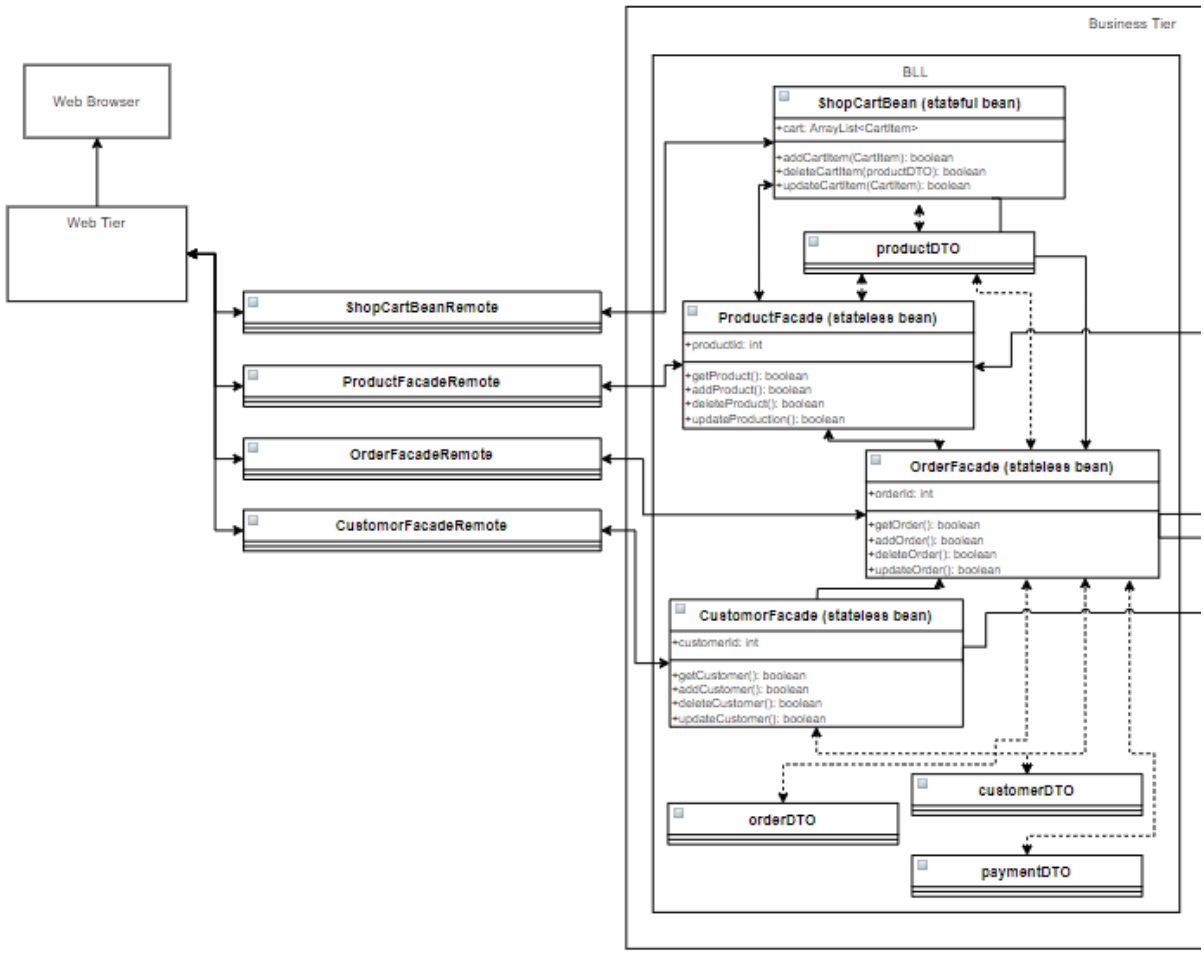
Deleting an item:

```
Please choose what operation you would like to perform:
1: Add
2: Update
3: Delete
4: Display Cart
5: Exit

3

Please enter the id of the item:
001
Deleting item with id 001 from cart
Item with id 001 was removed from cart
The shopping cart is empty!
```

Task 5.

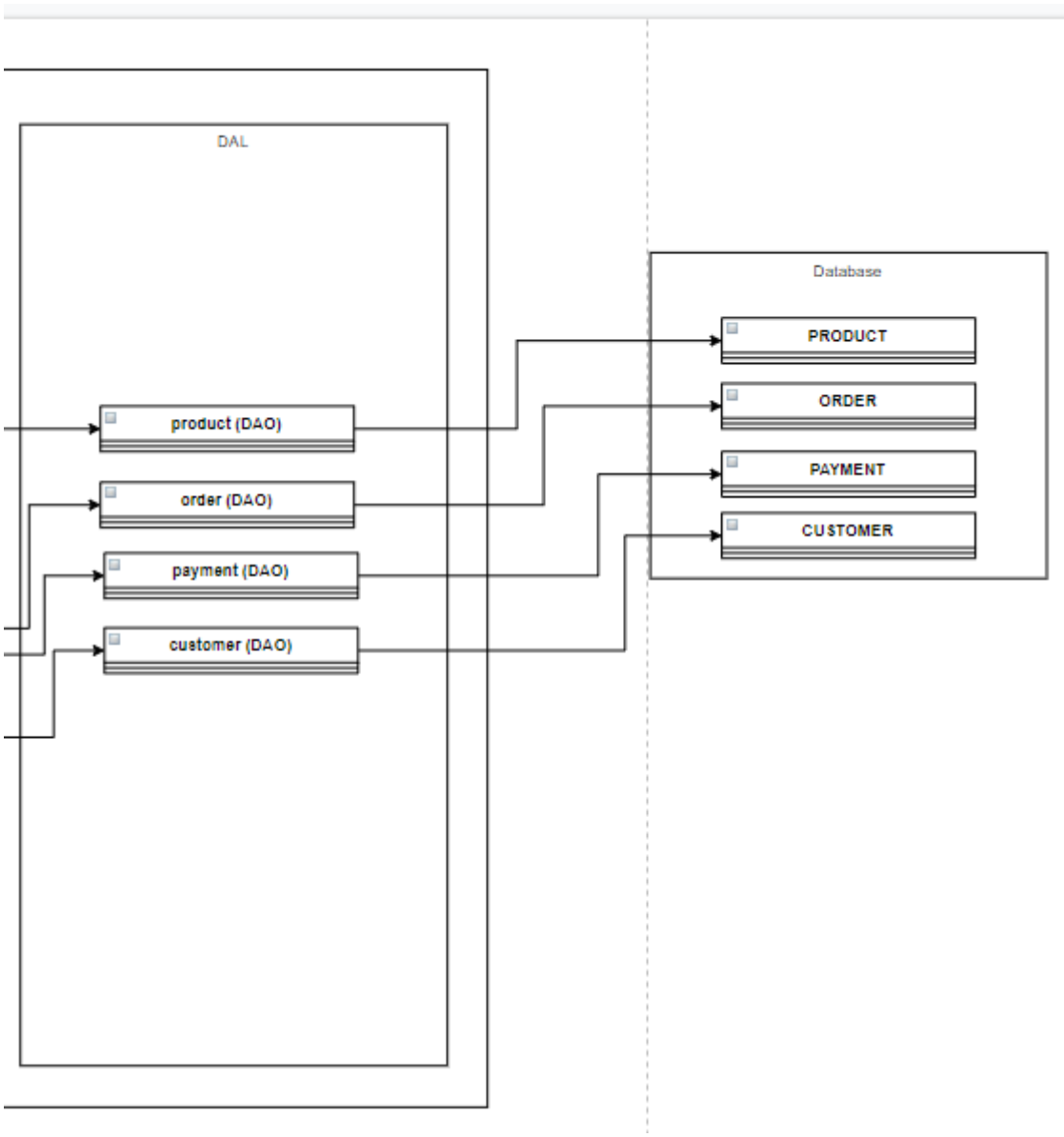
5.1

(First Half)



Cyrus Edgren 101606991
Tutor: Wei Lai

(Second Half)



5.2

ShopCartBean (Stateful Session Bean)

ShopCartBean is a stateful session bean which keeps track of products that a user may want to purchase in a cart. This needs to be able to interact with ProductFacade to pull product information as weel as OrderFacade to generate the order with the products that are stored in the cart. This bean is stateful unlike the other beans as is not used for a single transaction but rather needs a memory of what the user has selected and can be edited if needed.

ProductFacade (Stateless Session Bean)

ProductFacade is a stateless session bean that handles the business logic and CRUD operations for products. This bean will mostly interface with the ShopCartBean and adding products to the cart. OrderFacade also will need to access the ProductFacade to change product levels once an order has been made.

DAO's / Entity Class'

DAO's are the data access objects in charge of taking data and making queries to change data in the database as well as making queries to retrieve data.

OrderFacade (Stateless Session Bean)

OrderFacade is a stateless session bean that handles the business logic and CRUD operations for orders. This bean will need to collect the grouped product data from the ShopCartBean and customer data from CustomorFacade. The bean will also need to send updated stock levels to ProductFacade so they can be updated.

CustomerFacade (Stateless Session Bean)

CustomerFacade is a stateless session bean that handles the business logic and CRUD operations for customers. Orders will need to interface here to retrieve customer data to assign to an order.

DTO's

DTO's are the data transfer objects which are created whenever data needs to be stored in an object and used or manipulated. Does not directly manipulate the database, it just holds the data that is passed around by beans.