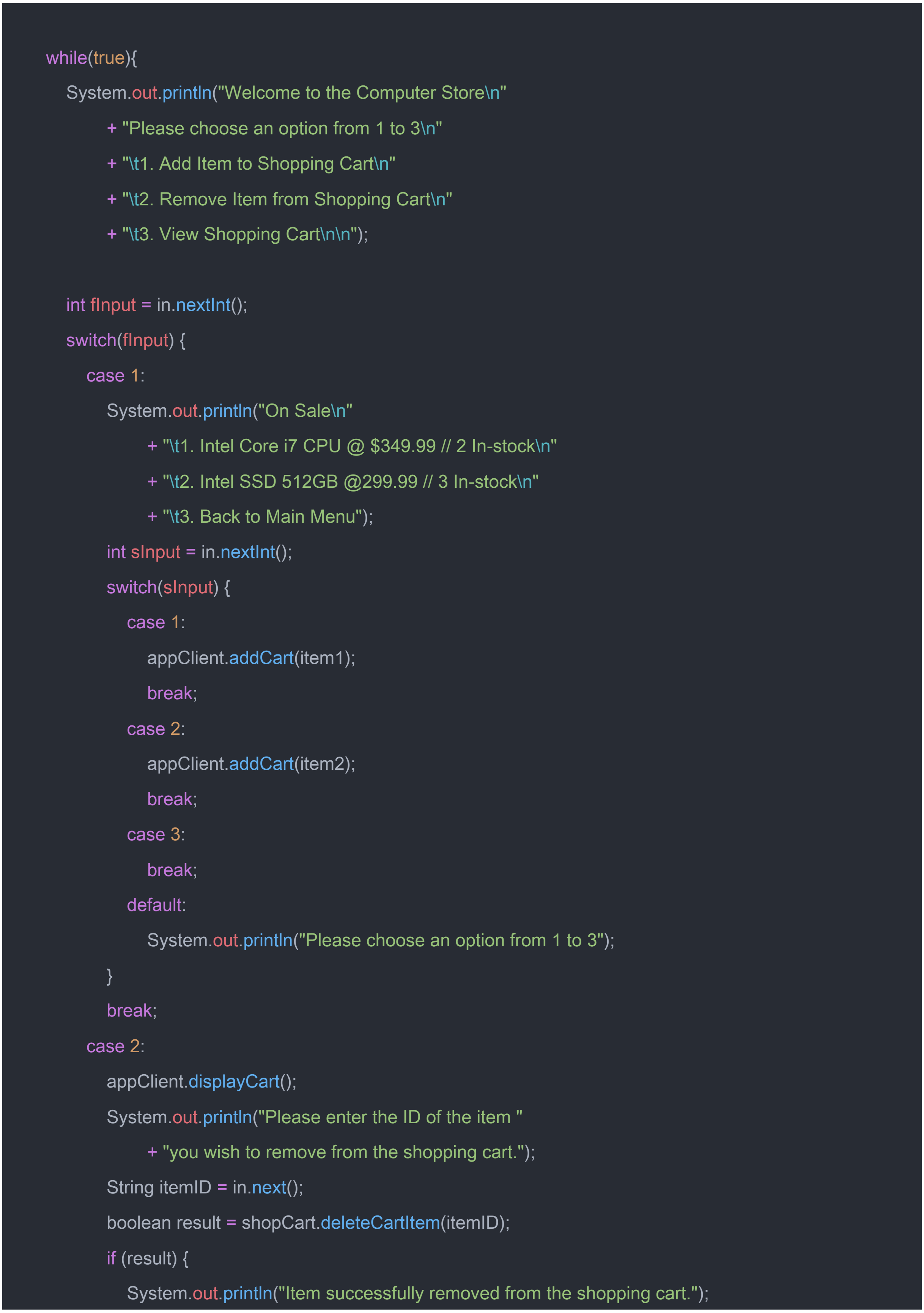
ShopCartAppClient



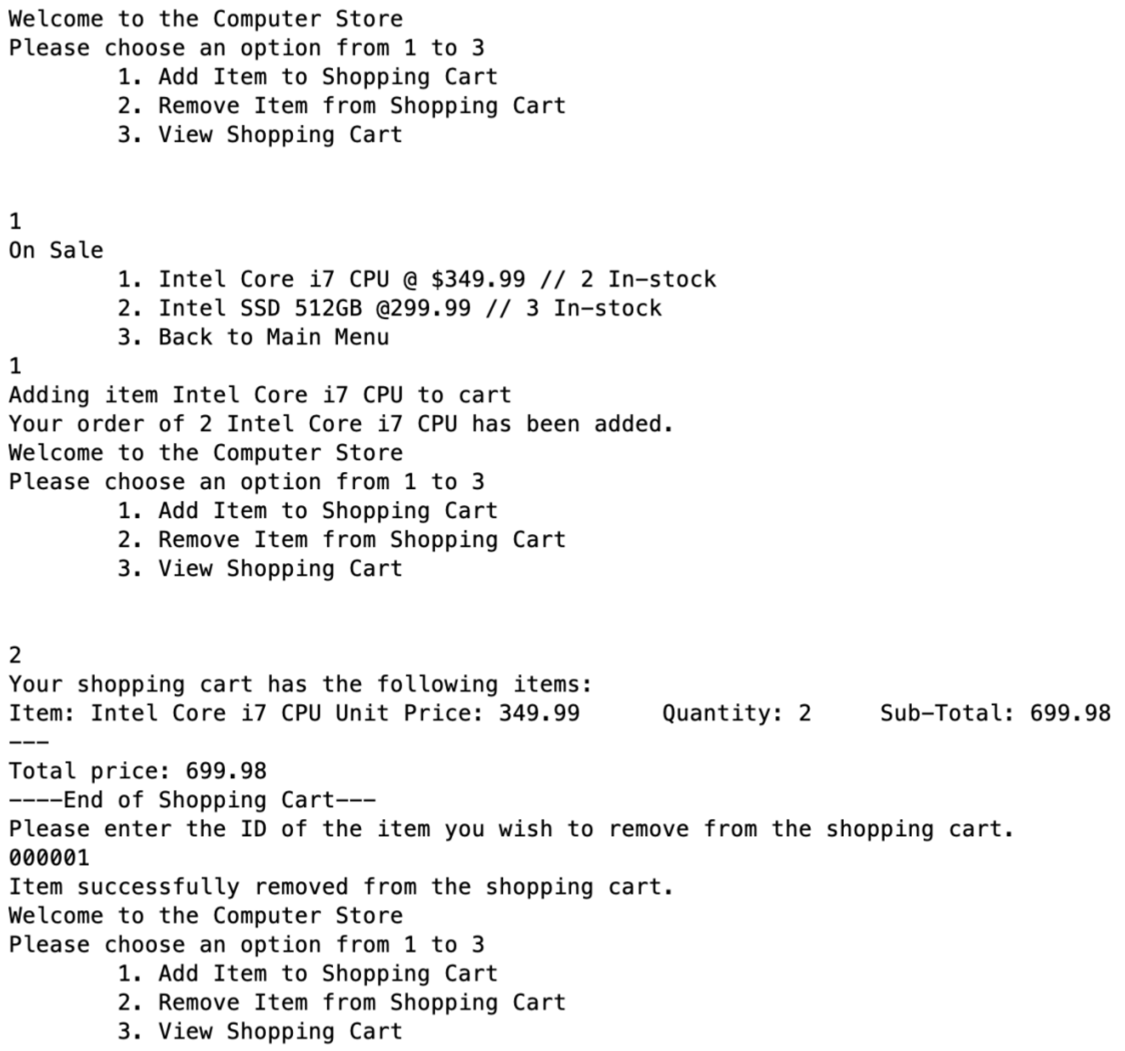
```
while(true){
    System.out.println("Welcome to the Computer Store\n"
            + "Please choose an option from 1 to 3\n"
            + "\t1. Add Item to Shopping Cart\n"
            + "\t2. Remove Item from Shopping Cart\n"
            + "\t3. View Shopping Cart\n\n");

    int fInput = in.nextInt();
    switch(fInput) {
        case 1:
            System.out.println("On Sale\n"
                    + "\t1. Intel Core i7 CPU @ $349.99 // 2 In-stock\n"
                    + "\t2. Intel SSD 512GB @299.99 // 3 In-stock\n"
                    + "\t3. Back to Main Menu");
            int sInput = in.nextInt();
            switch(sInput) {
                case 1:
                    appClient.addCart(item1);
                    break;
                case 2:
                    appClient.addCart(item2);
                    break;
                case 3:
                    break;
                default:
                    System.out.println("Please choose an option from 1 to 3");
            }
            break;
        case 2:
            appClient.displayCart();
            System.out.println("Please enter the ID of the item "
                    + "you wish to remove from the shopping cart.");
            String itemID = in.next();
            boolean result = shopCart.deleteCartItem(itemID);
            if (result) {
                System.out.println("Item successfully removed from the shopping cart.");
                } else {
                    System.out.println("Item failed to be removed from the shopping cart.\n"
                            + "Wrong ID entered. Try again.");
                }
                break;
            case 3:
                appClient.displayCart();
                break;
            default:
                System.out.println("Please choose an option from 1 to 3");
        }

    }
}

public void addCart(CartItem item) {
    System.out.println("Adding item " + item.getDescription() + " to cart");
    if (shopCart.addCartItem(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
```

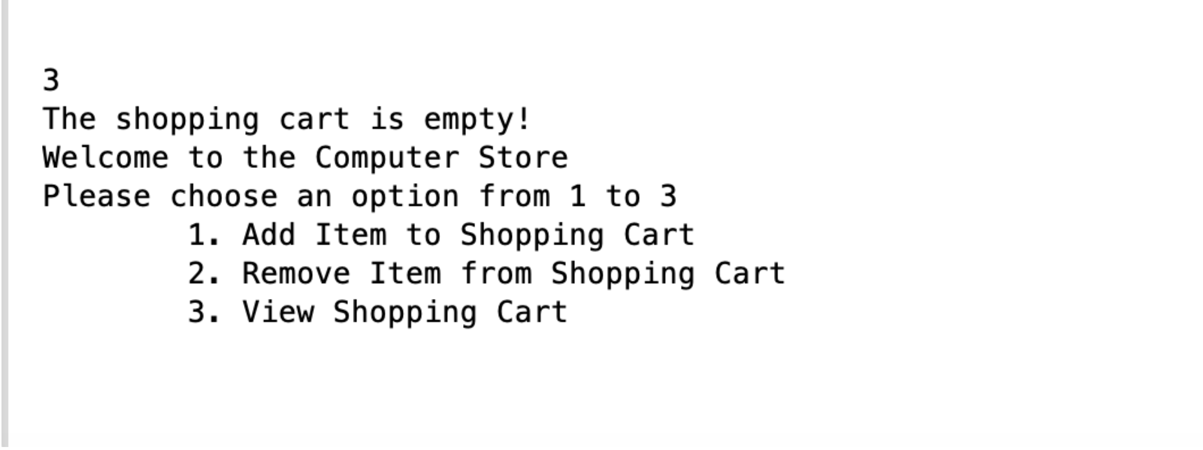
ShopCartBean

```
package ed.sfsb;

import dto.CartItem;
import java.util.ArrayList;
import javax.ejb.Remove;
import javax.ejb.Stateful;

@Stateful
public class ShopCartBean implements ShopCartBeanRemote {

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

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

    private boolean add(CartItem cartItem) {
        boolean result = false;
        try
        {
            result = cart.add(cartItem);
        }
        catch (Exception ex) {}

        return result;
    }

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

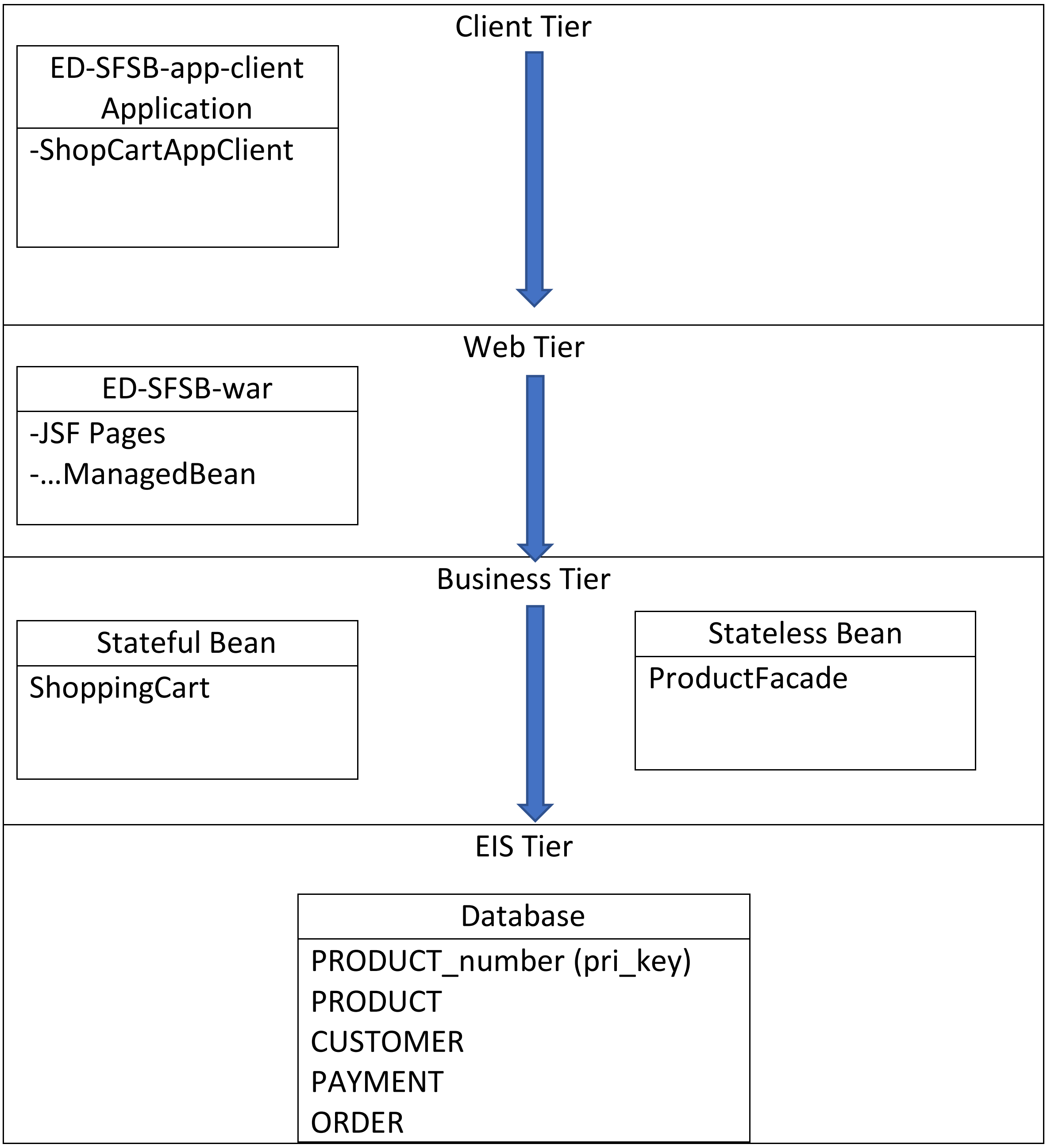
```
}

@Override
public boolean addCartItem(CartItem cartItem) {

    try
    {

        for (CartItem c : cart)
        {
            if (cartItem.getItemId().equals(c.getItemId())) { return updateCartItem(cartItem); }
        }



        if (cartItem == null) { return false; }
        return add(cartItem);
    }
    catch (Exception e)
    {
        System.out.println(e);
        return false;
    }
}

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

    try
    {
        for (CartItem c : cart)
        {
            if (cartItem.getItemId().equals(c.getItemId()))
            {
                cartItem.setQuantity(cartItem.getQuantity() + c.getQuantity());
                cart.remove(c);
                cart.add(cartItem);
                result = true;
            }
        }
        }
        catch (Exception e)
        {
            System.out.println(e);
        }

        return result;
    }

    @Override
    public boolean deleteCartItem(String itemId) {
        boolean result = false;

        try
        {
            for (CartItem c : cart)
            {
                if (itemId.equals(c.getItemId()))
                {
                    cart.remove(c);
                    result = true;
                }
            }
        }
        catch (Exception e)
        {
            System.out.println(e);
        }

        return result;
    }
}
```



```
Welcome to the Computer Store
Please choose an option from 1 to 3
        1. Add Item to Shopping Cart
        2. Remove Item from Shopping Cart
        3. View Shopping Cart


1
On Sale
        1. Intel Core i7 CPU @ $349.99 // 2 In-stock
        2. Intel SSD 512GB @299.99 // 3 In-stock
        3. Back to Main Menu
1
Adding item Intel Core i7 CPU to cart
Your order of 2 Intel Core i7 CPU has been added.
Welcome to the Computer Store
Please choose an option from 1 to 3
        1. Add Item to Shopping Cart
        2. Remove Item from Shopping Cart
        3. View Shopping Cart


2
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---
Please enter the ID of the item you wish to remove from the shopping cart.
000001
Item successfully removed from the shopping cart.
Welcome to the Computer Store
Please choose an option from 1 to 3
        1. Add Item to Shopping Cart
        2. Remove Item from Shopping Cart
        3. View Shopping Cart
```



```
3
The shopping cart is empty!
Welcome to the Computer Store
Please choose an option from 1 to 3
        1. Add Item to Shopping Cart
        2. Remove Item from Shopping Cart
        3. View Shopping Cart
```



Client tier component runs on the client machine. For this instance, the client performs their action on a text-based GUI. When enquired, the system will launch ShopCartAppClient.
Web tier component runs on the Jave EE server. This web tier consists of web pages created using JSF.
Business tier component runs on the Java EE component. Business code/ logic receives data from client, processes it and sends it to the enterprise information system tier for storage, vice versa. Stateful Bean preserves implementation of the shopping cart while stateless stores product information.
Enterise information system (EIS) runs on the EIS server tier handles EIS software and includes enterprise infrastructure systems. Database of the whole operation is stored here to access as well as retrieve.