Lab assignment 5: Domain Driven Design

Exercise 1: Bounded context/components

Create a new class diagram that shows the design of our webshop using bounded contexts/components based on our aggregates found in Lab4. Show all classes, including controllers, service classes, DAO's/repositories, DTO's and other plumbing classes.

Exercise 2: Implementation of the webshop with components

Given is the project WebShopComponents. It contains the implementation of the different components in the webshop.

Implement the todo's in in the file shop.WebShopApplication.java using the restTemplate:

```
	public void run(String... args) throws Exception {
		//create customer
		CustomerDTO customerDto = new
CustomerDTO("101","Frank","Brown","fBrown@Hotmail.com","123456");
		AddressDTO addressDTO = new AddressDTO("1000 N main Street",
"Fairfield","52557","USA");
		customerDto.setAddress(addressDTO);
		//todo: call the customer component to add the customer

		// get customer
		//todo: call the customer component to get the customer with
id 101 and print the result

		//create products

		//todo: call the product component to create the first
product
		//todo: call the product component to create the second
product

		//set stock
		//todo: call the product component to set the stock for the
first product

		//get a product
		//todo: call the product component to get the the first
product and print the result

		// add products to the shoppingcart
		//todo: call the shopping component to add the first product
to the cart
```

```
            //todo: call the shopping component to add the second
product to the cart


            //get the shoppingcart
            //todo: call the shopping component to get the cart and
print the result

            //checkout the cart
            //todo: call the shopping component to checkout the cart

            //get the order
            //todo: call the order component to get the order and print
the result

            //add customer to order
            //todo: call the order component to add a customer to the
order

            //confirm the order
            //todo: call the order component to confirm the order

            //get the order
            //todo: call the order component to get the order and print
the result

      }
```