

Homework #8
Service-Oriented Architectures

Create an application to support the business processes in an e-commerce company (Reuse the domain model of the Homework #4).

Users (main stakeholders): Customer (with a specialization: Special Customer) as business actor and Operator as agent. Another business actor is Supplier that is specialized to provide the e-shop with a certain product.

The business use cases are the followings:

- User Management: create account, delete account and edit account.
- Product Management (Operator is the primary software actor): add new product, delete product, edit product description, order a product to some supplier, and check the product Stock in warehouse.
- Cart Management (Customer is the primary software actor): new cart instance (login), add product in the cart, total computation, place an order.
- Order Management (Operator is the primary software actor): list orders sorted according various criteria, verify inventory.

Constraints

1. There are two server machines in the company. The web server and the database for order management are running in the first machine. The second machine hosts the inventory application with the stock database of the company. The company intends to change the network configuration in the future either by substituting the two machine with a unique, more powerful one, or to add a new machine for the inventory application. This application should continue to run in both configurations without changes.
2. the application has to use J2EE technology

Requests

- a. Design a service-oriented architecture for this application**
- b. Identify the needed services and their types (basic, intermediary, process, public)**
- c. Implement the services as web services using the J2EE technology**
- d. Write a client application with a graphical user interface for two user types: customer and operator.**
- e. Implement and run the following scenario:**

1. The operator adds 6 new products in the inventory, each with a non-zero quantity
2. The first customer creates an order containing 3 products and places it to be delivered.
3. The second special customer creates an order containing 2 products and places it to be delivered.
4. The operator asks a supplier to provide the shop with a given quantity of items of a given product.
4. The operator asks to view the order to be delivered.

Hints

- Feel free to add attributes to the classes if needed.
- Feel free to add methods in the component interfaces if needed.
- Feel free to post comments to ask for additional data.