Lab assignment 2: Application architecture

Exercise 1

Design de architecture with all the necessary classes for a webshop with the following requirements:

The webshop contains products that we can add or remove from a shoppingcart
We can also add multiple copies of the same product in the shoppingcart
We can place an order based on our shoppingcart.
Orders can only be paid by creditcard
For every order we need to know
- The shipping address and the billing address
- The customer
- The creditcard information
- The shipping method

The webshop administrator should be able to add, remove and update products.
For every product we need to know the following information:
- ProductNumber, name , price, description.
- Reviews from customers that have purchased this product.
- Stock information: number in stock, and location code in the warehouse
- The supplier for this product

Customers can create an account, so they can login when they want to place on order
- For every customer we want to know the following information:
- CustomerNumber, first name, last name, email, phone and address information
- Creditcards from the customer
- Orders that the customer has placed

When an order is placed, the webshop should send an email to the customer.
The webshop administrator should be able to add, remove and update suppliers.

Draw a class diagram showing the different layers and their corresponding classes and relationships.

Draw a sequence diagram showing diagram showing how we place an order.

Exercise 2

Describe the different places where we can store state. Give the advantages and disadvantages of these different options.

Exercise 3

For each of the following integration possibilities, describe its advantages, its disadvantages and when you would apply it:

1. RMI
2. Messaging (JMS)
3. SOAP
4. REST
5. Serialized objects over HTTP
6. Database integration
7. File based integration

Exercise 4

Read the following article:

http://www.drdobbs.com/errant-architectures/184414966

Explain the advantages and disadvantages of distributed systems.