Submission for Milestone 2
Middleware Project
Winter Term 2014/15

TECHNISCHE UNIVERSITÄT DARMSTADT

Manisha Luthra (2687667)
Patrick Welzel (1478819)
Pratyush Agnihotri (2387187)

Problem 2.1 Goal overview

In this milestone, we achieved to create an Enterprise Application Client for the *Employee* to add and edit customer and product data. In addition, we implemented the feature "Order Process Support".

Problem 2.2 Architectural overview

In the first milestone we started with the order example of JavaEE 6 and implemented the registration forms for customer as well as products. Then, we started implementation of the enterprise application client for the employee. Now, for the application client to be able to access the session beans, we encountered to have an remote interface. Therefore, we created first an enterprise application that contained all our session beans, *CustomerBean*, *EmployeeBean*, *ProductBean* and *UserSessionBean* and a Java class library that contained the remote interface for the session beans. The business logic interacts with the database via remote interface, using Java persistence *Entity_Manager* imported from JavaEE 7 libraries.

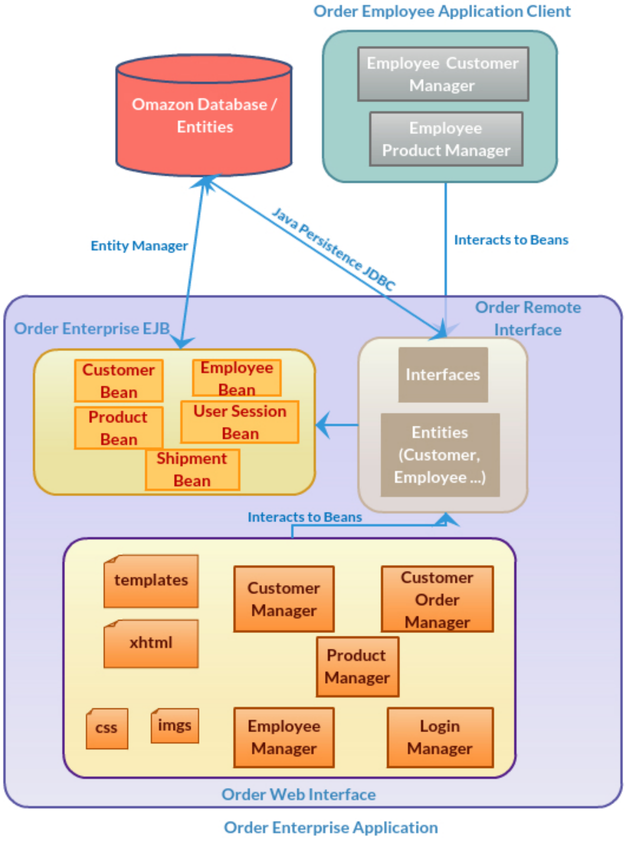


Figure 1: Omazon Architecture

The Employee application client and the user/employee web interface accesses the session beans, through the remote interface in the class library. The class library is accessed by the enterprise application (web interface) as well as the application client. The interaction between the various components is illustrated in the architecture diagram in Figure 1.

Problem 2.3 Encountered problems/Implementation Alternatives

Interaction between the models

1. We encountered that our previous architecture could not fulfill the requirement of an native client. To tackle this, we introduced the remote library for our business logic (beans), so that the interaction between the client and business logic could take place.
2. Now, interaction with the database via Java Persistence required importing existing entities from the database to the remote interface. Further communication of JavaEE persistence *Entity_Manager* required using JavaEE 7. So, we proceed using JavaEE 7 libraries for our project.
3. In addition to this, the interaction between the entities, primary key <-> foreign key, in the logic was a problem. Earlier we did not really took care of the relationships between models what we now had to add. The first point where we needed this was displaying the current customers orders on the web interface without writing a manual filter over all existing orders but using a *WHERE* clause and a parameterized query.

Code Cleanup and Refactoring

As already mentioned at several points, we started of with the basic example and often used the existing files as a reference. Now, as the project has grown, we realized that we have to refactor a lot to keep the overall code base and therefor the effort of debugging and also implementing new features low.

JSF/JSP different then "normal" Web Development

Compared to Web Development known to us from the PHP and Python world, JSP behaves much more like a native client framework and contradicts the way we used to think about web application and known practices in their development.

Problem 2.4 Implemented features

As illustrated in the architectural overview, the design contains an enterprise application with the necessary business logic of adding/editing customer/product data and the order process. The *Employee* application client can also add/edit the customer and product data by accessing the same business logic. As per the business logic, the user or the *Customer* can login and order a product/multiple products. The corresponding *Order* details of the customer is stored in the database, and the *shipment_ID* is automatically generated for the order. The *Status* is then marked by the *Employee*. *Customer* is then able to check the status via the web interface.

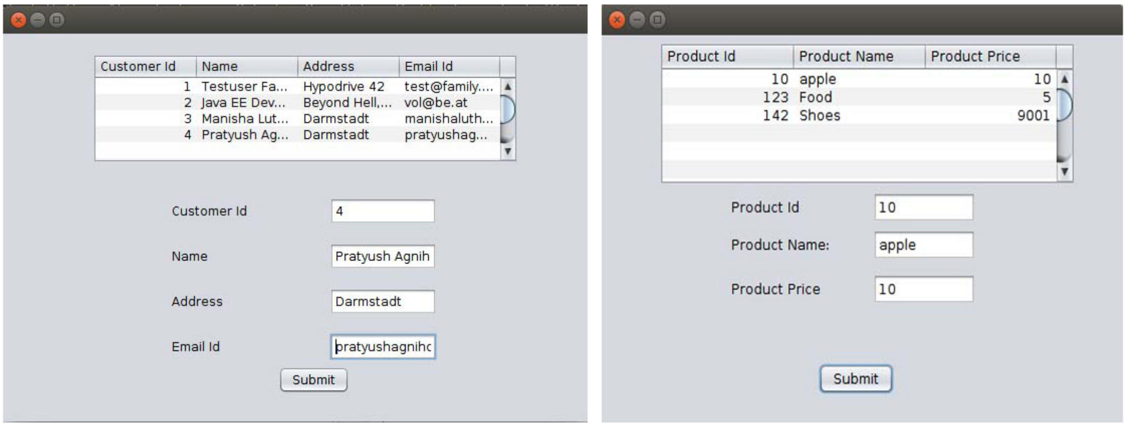
Problem 2.4.1 Java Native interface for Employee

Employee application client interacts to the session beans via Java class library. This displays the current list of products and customers to the employee. He can add or edit new data into the current list which is then synchronized with the Omazon Inc. database. The client functions can be seen in Figure 2.

Problem 2.4.2 Order Process Support

The functions of customer and employee user interface can be seen in Figure 3 and 4.

1. **Order Process:** Customer can add one or more products from the product list to his cart, which is only saved in a SessionScoped Manager. Once confirmed, the Order and an according Shipment is created in the database. The user has the possibility to see a summary of his order including the current shipment status.
2. **Shipment Status update:** Customer can check the status of his placed order via the web interface. The Employee can update the status of all orders.
3. **User Interface:** The *CustomerManager*, *CustomerOrderManager* and so on, the *ManagedBeans*, handles the communication and display the successful order table to the customer and the employee.



Customer Id	Name	Address	Email Id
1	Testuser Fa...	Hypodrive 42	test@family....
2	Java EE Dev...	Beyond Hell,...	vol@be.at
3	Manisha Lut...	Darmstadt	manishaluth...
4	Pratyush Ag...	Darmstadt	pratyushag...

Customer Id: 4

Name: Pratyush Agnih

Address: Darmstadt

Email Id: pratyushagnihc

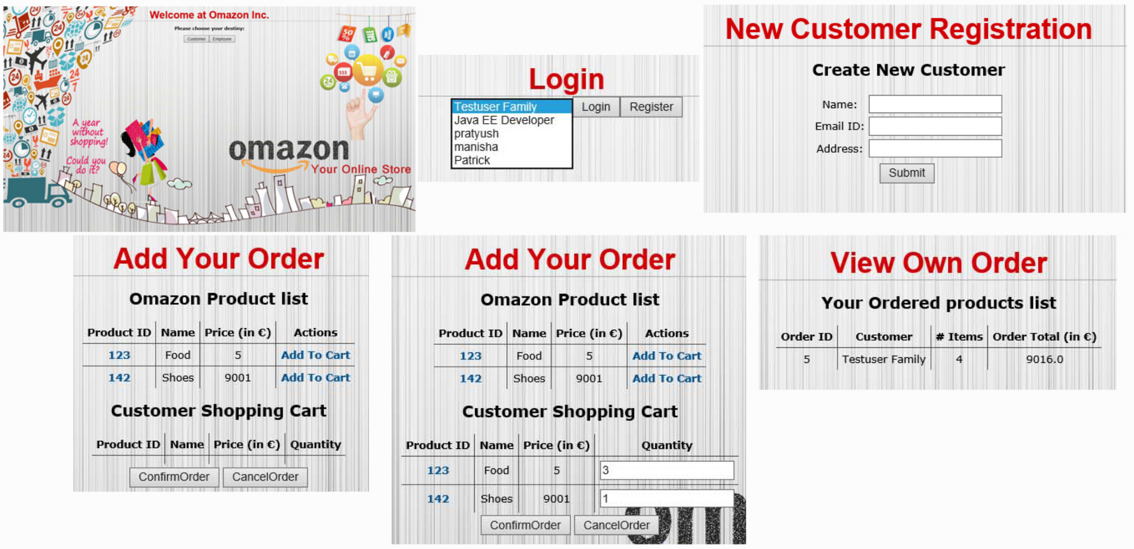
Product Id	Product Name	Product Price
10	apple	10
123	Food	5
142	Shoes	9001

Product Id: 10

Product Name: apple

Product Price: 10

Figure 2: Java native client for Employee



Login

Testuser Family
Java EE Developer
pratyush
manisha
Patrick

New Customer Registration

Create New Customer

Name:

Email ID:

Address:

Add Your Order

Omazon Product list

Product ID	Name	Price (in €)	Actions
123	Food	5	Add To Cart
142	Shoes	9001	Add To Cart

Customer Shopping Cart

Product ID	Name	Price (in €)	Quantity

Add Your Order

Omazon Product list

Product ID	Name	Price (in €)	Actions
123	Food	5	Add To Cart
142	Shoes	9001	Add To Cart

Customer Shopping Cart

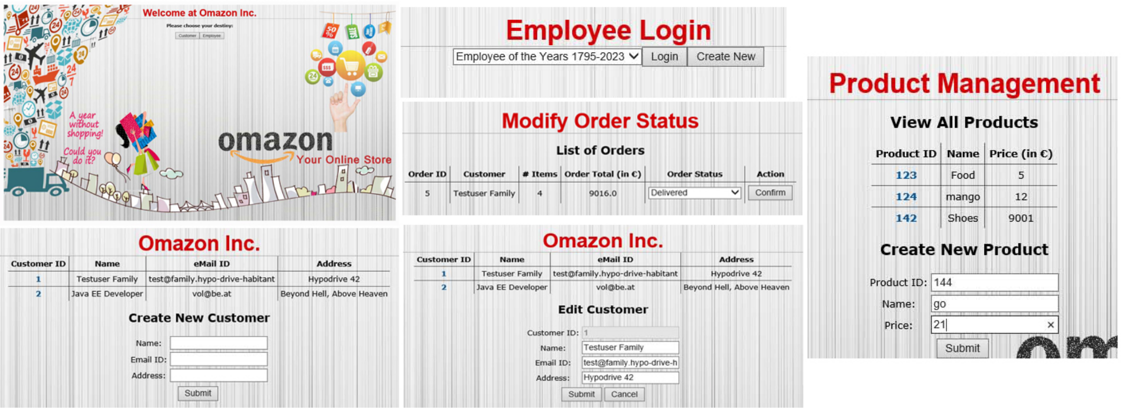
Product ID	Name	Price (in €)	Quantity
123	Food	5	3
142	Shoes	9001	1

View Own Order

Your Ordered products list

Order ID	Customer	# Items	Order Total (in €)
5	Testuser Family	4	9016.0

Figure 3: Customer Interface



Employee Login

Employee of the Years 1795-2023

Modify Order Status

List of Orders

Order ID	Customer	# Items	Order Total (in €)	Order Status	Action
5	Testuser Family	4	9016.0	Delivered	Confirm

Omazon Inc.

Customer ID	Name	eMail ID	Address
1	Testuser Family	test@family.hypo-drive-habitant	Hypodrive 42
2	Java EE Developer	vol@be.at	Beyond Hell, Above Heaven

Create New Customer

Name:

Email ID:

Address:

Omazon Inc.

Customer ID	Name	eMail ID	Address
1	Testuser Family	test@family.hypo-drive-habitant	Hypodrive 42
2	Java EE Developer	vol@be.at	Beyond Hell, Above Heaven

Edit Customer

Customer ID: 1

Name: Testuser Family

Email ID: test@family.hypo-drive-h

Address: Hypodrive 42

Product Management

View All Products

Product ID	Name	Price (in €)
123	Food	5
124	mango	12
142	Shoes	9001

Create New Product

Product ID: 144

Name: go

Price: 21

Figure 4: Employee Interface