The Mandatory Programming Assignment INF5040 Autumn 2009

The mandatory assignment is to develop a distributed application that models a small e-banking application. The system consists of a client (JSP), a servlet for processing client requests, a server that consists of one or more EJBs, and a database.

The system has two main features:

- 1 Creating a bank account
- 2 Transferring money from one account to another account.

Technical features:

- 1. Bank account Creation: Use Java Persistence (hint: Entity Manager and entity beans)
- 2. Money transfer: Basically this action should be run in the context of a **Transaction** in order to preserve the consistency of data (amount of money transferred).

To learn the importance of using transaction the money transfer should be developed in three following ways:

- Successful transfer within the transaction (without any exception)

```
Start transaction
withdraw amount X from account A
deposit amount X to account B
End Transaction
```

- Unsuccessful transfer within the transaction (with exception)

```
Start transaction
withdraw amount X from account A
If (true) {
    throw Exception e
    }
    deposit amount X to account B
End Transaction
```

- Unsuccessful transfer without transaction

```
withdraw amount X from account A
If (true) {
  throw Exception e
}
deposit amount X to account B
```

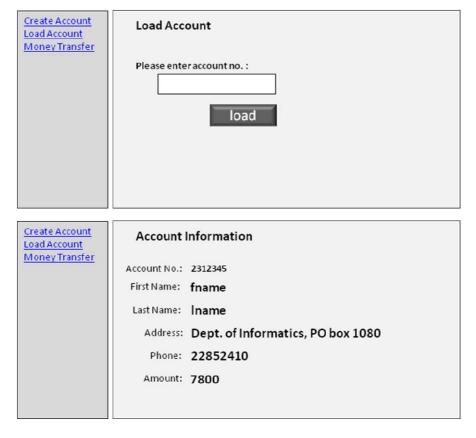
Your tasks:

1. Client (JSP)

A client for creating an account:

Create Account Load Account Money Transfer	Create Account First Name: Last Name: Address: Phone: Amount: Create
Create Account Load Account Money Transfer	Account created successfully
	Account No.: 2312345
	First Name: fname
	Last Name: Iname
	Address: Dept. of Informatics, PO box 1080
	Phone: 22852410
	Amount: 10000

A client for loading an account:



A Client for money transfer

Create Account Load Account Money Transfer	Money Transfer
	From:
	To:
	Amount:
	Successful Transfer
	Unsuccessful Transfer(WT) (Within Transaction)
	Unsuccessful Transfer(WoT) (Without Transaction)

- 2. A Servlet for receiving client side actions (1-create account, 2-load account, 3-successful transfer, 4-unsuccessful transfer(WT), and 5-unsuccessful Transfer(WoT)) and forwarding them to the SessionBean.
- 3. One Session Bean to handle the data:

Use Java Persistence: Entity Manager and entity beans

Development Tools:

 $Application\ Server:\ JBoss\ IDE\ (Integrated\ Development\ Environment):\ NetBeans,\ or$

other tools that you prefer

Database: Hypersonic SQL embedded in JBoss

Links for download and further details are available at the group homepage.

Deliverables:

Please send the following artifacts to sabita@ifi.uio.no

A description of your design and implementation (Word, plain text, or preferably PDF), which should include a user guide for compiling and running your distributed application.

Everything that is needed to compile and run your distributed application (in a zip archive), such as:

- source code (should be well commented) build.xml file (preferably only one file for the entire system, that can compile, deploy, execute the client(s), and clean up everything when finished)
- (*jar-file*(*s*))

•

Announcements requested by the Department of Informatics:

– This assignment is mandatory and must be approved to take the INF5040 exam. –

All students must read the departmental guidelines for written assignments:

Norwegian: http://www.ifi.uio.no/studinf/skjemaer/erklaring.pdf
English: http://www.ifi.uio.no/studinf/skjemaer/erklaring.pdf

Deadline: 23:00 on Monday the 5th October, 2009