

# 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:

<a href="#">Create Account</a> <a href="#">Load Account</a> <a href="#">Money Transfer</a>	<h3>Create Account</h3> <p>First Name: <input type="text"/></p> <p>Last Name: <input type="text"/></p> <p>Address: <input type="text"/></p> <p>Phone: <input type="text"/></p> <p>Amount: <input type="text"/></p> <p><input type="button" value="create"/></p>
--	---

<a href="#">Create Account</a> <a href="#">Load Account</a> <a href="#">Money Transfer</a>	<h3>Account created successfully</h3> <p>Account No.: 2312345</p> <p>First Name: <b>fname</b></p> <p>Last Name: <b>lname</b></p> <p>Address: <b>Dept. of Informatics, PO box 1080</b></p> <p>Phone: <b>22852410</b></p> <p>Amount: <b>10000</b></p>
--	---

A client for loading an account:

<a href="#">Create Account</a> <a href="#">Load Account</a> <a href="#">Money Transfer</a>	<h3>Load Account</h3> <p>Please enter account no. :</p> <p><input type="text"/></p> <p><input type="button" value="load"/></p>
--	--

<a href="#">Create Account</a> <a href="#">Load Account</a> <a href="#">Money Transfer</a>	<h3>Account Information</h3> <p>Account No.: 2312345</p> <p>First Name: <b>fname</b></p> <p>Last Name: <b>lname</b></p> <p>Address: <b>Dept. of Informatics, PO box 1080</b></p> <p>Phone: <b>22852410</b></p> <p>Amount: <b>7800</b></p>
--	---

## A Client for money transfer

The screenshot shows a web application interface for money transfer. On the left side, there is a sidebar with three links: "Create Account", "Load Account", and "Money Transfer". The main content area is titled "Money Transfer" and contains three input fields: "From:", "To:", and "Amount:". Below these fields are three buttons: "Successful Transfer", "Unsuccessful Transfer(WT) (Within Transaction)", and "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](mailto: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/declaration.doc>

**Deadline: 23:00 on Monday the 5<sup>th</sup> October, 2009**