

# Work groups meeting - 4

---

## INF5040 (Open Distributed Systems)

**Sabita Maharjan**

[sabita@simula.no](mailto:sabita@simula.no)

Department of Informatics

University of Oslo

September 07, 2009



# Agenda

- The programming assignment
- Assignment demo



# Programming assignment #1

- Comments and corrections are welcome (assignment, tools, ...)
- Group teacher (Amir Taherkodi) will be available on 14<sup>th</sup> September, 10:15 –12:00 – IFI, Room 3B
- No work group meetings on the 21<sup>st</sup> and 28<sup>th</sup> of September, but you can email the group teacher (Amir Taherkodi) if there are any questions/queries related to the assignment.
- Deadline: 23:00 on Monday the 5<sup>th</sup> of October 2009 by email to [sabita@ifi.uio.no](mailto:sabita@ifi.uio.no), [amirhost@ifi.uio.no](mailto:amirhost@ifi.uio.no)
- More details are given in the assignment specification
- Assignment is mandatory and must be approved to be allowed to take the INF5040 exam



# The application

- A distributed application
- Using J2EE technologies and tools for development
- An e-banking application with the minimum set of functionalities
- Application from two different views
  - Application Logic
  - Technical



# E-banking Application

- Application logic
  - Creating a bank account
  - Transferring money from one account to another account
- Technical features
  - How to create a bank account
    - Use Java Persistence (hint: Entity Manager and entity beans)
  - How to transfer money

# E-banking Application

- Money transfer
  - This action should be run in the context of a **Transaction**
    - to preserve the consistency of data (amount of money)
  - *To learn the importance of using transaction: the following methods should be developed*

# Money Transfer

1. Successful transfer within the transaction (without any exception)
  2. Unsuccessful transfer within the transaction (with exception)
  3. Unsuccessful transfer without transaction
- Sample pseudo codes are available in the assignment specification



# Your tasks

- Client (JSP) for creating an account

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



# Prototype

- Client after creating an account

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



# Prototype

- A client for loading an account

<p><a href="#">Create Account</a> <a href="#">Load Account</a> <a href="#">Money Transfer</a></p>	<p><b>Load Account</b></p> <p>Please enter account no. :</p> <input data-bbox="736 715 1066 778" type="text"/> <input data-bbox="884 786 1070 853" type="button" value="load"/>
---	--

# Prototype

- Loaded account

<a href="#">Create Account</a> <a href="#">Load Account</a> <a href="#">Money Transfer</a>	<b>Account Information</b> Account No.: <b>2312345</b> First Name: <b>fname</b> Last Name: <b>lname</b> Address: <b>Dept. of Informatics, PO box 1080</b> Phone: <b>22852410</b> Amount: <b>94500</b>
--	---



# Prototype

- Money Transfer

<p><a href="#">Create Account</a> <a href="#">Load Account</a> <a href="#">Money Transfer</a></p>	<h3>Money Transfer</h3> <p>From: <input type="text"/></p> <p>To: <input type="text"/></p> <p>Amount: <input type="text"/></p> <p><input type="button" value="Successful Transfer"/></p> <p><input type="button" value="Unsuccessful Transfer(WT)"/> (Within Transaction)</p> <p><input type="button" value="Unsuccessful Transfer(WoT)"/> (Without Transaction)</p>
---	---



# Server-side tasks

- A Servlet for receiving client side actions and forwarding them to the the SessionBean.
  1. create account
  2. load account
  3. successful transfer
  4. unsuccessful transfer(WT)
  5. unsuccessful transfer(WoT))
- One Session Beans to handle the data: Use Java Persistence: Entity Manager and entity beans



# Development tools

- Application Server: JBoss
- IDE (Integrated Development Environment)  
: NetBeans, eclipse
- Database: Hypersonic SQL embedded in JBoss
- Links for download and further details are available at the group homepage