University of Westminster
Faculty of Science and Technology

Module code and title:
5COSC004W-Client Service Architecture

Module leader	**Gabor Terstyanszky**
Unit	**Coursework**
Coursework mode	**group coursework**
Weight	**50%**
Qualifying mark	**30 marks**
Description	**Designing and Implementing a Web Service Based e-Bank**
Learning Outcomes	**LO2 and LO3**
Handed Out:	**10/02/17**
Due Date	**06/04/17**
Expected deliverables	**UML diagrams and Java source code**
Method of Submission:	**electronically on BB**
Type of feedback and due date	**written assessment** **27/04/17** **All marks will remain provisional until formally agreed by an Assessment Board**

Coursework submission instructions

All coursework on this module is submitted via Blackboard only. It will automatically be scanned through a text matching system (designed to check for possible plagiarism). The system used will be either Turnitin or SafeAssign. The work you submit must be in Adobe Acrobat (.pdf) file format.

- You DO NOT need to attach a copy of the CA1 form;
- You DO need to include your name and student ID on the first page of your assignment.

To submit your assignment:

- Log on to Blackboard at http://learning.westminster.ac.uk;
- Go to the relevant module Blackboard site;
- Click on the Coursework submission link on the left-hand side as advised by the module leader;
- Click on the link to the relevant assignment;
- Follow the 'upload' and 'submit' instructions.

If you are unable to submit your work due to a finance hold you must email your work to **ecs-registry@westminster.ac.uk** by the same deadline, putting on the subject line the module code, assessment number, and your name. This shows that you have completed your work by the deadline. After the finance hold is lifted you must then submit the same work as normal on Blackboard, otherwise it will not be marked and you will get a fail for that assessment.

Assessment regulations

Refer to section 4 of the "How you study" guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:
http://www.westminster.ac.uk/study/current-students/resources/academic-regulations

Module code and title:
5COSC004W Client-Service Architecture

Coursework description

DESCRIPTION OF THE PROBLEM:

A high-street bank decided to install a client-server based IT system to support communication among its headquarter and local branches. This system will use web services to connect the headquarter and the local branches through two web services: employee service and customer service. The ***employee service*** manages bank employees' data and provides access to the customer service through the following operations:

- managing bank employees' data and,
- login/logout bank employees.

The ***customer service*** manages customers' accounts performing the following operations:

- displaying customers' list and
- managing customer's account.

Data of bank employees and customers is stored in storage, for example in a database or in file.

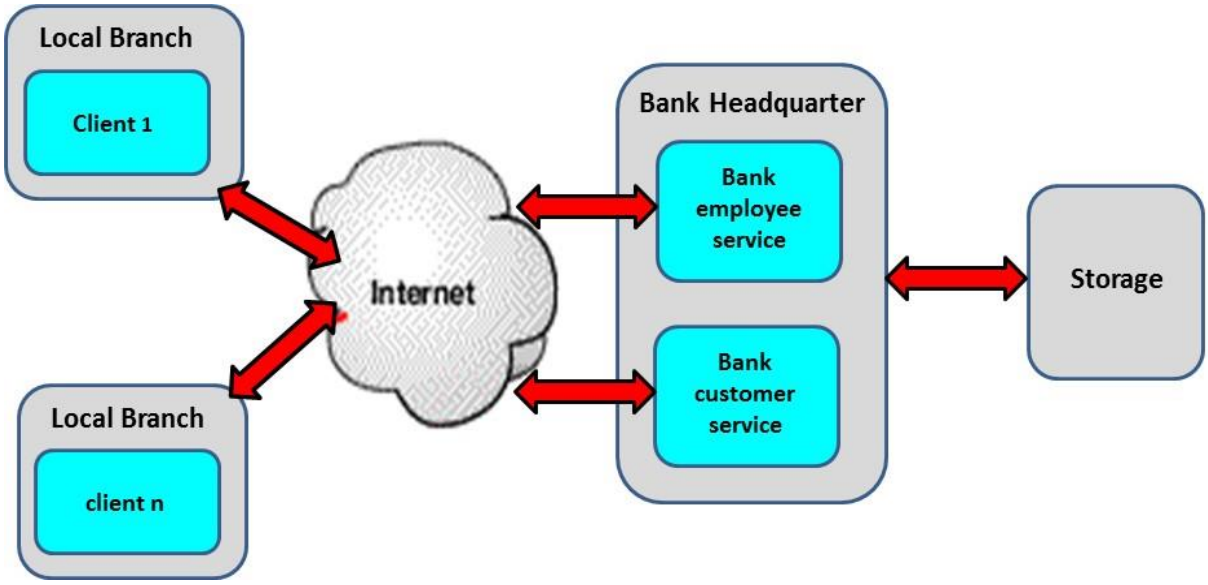

Figure 1: Web service-based bank

The bank employee invokes the employee and customer web services though a GUI. Starting the e-bank application the GUI displays the login page where the bank employee either can select the employee page or login. On the employee page the bank employee can manage employees' data. After login the customers' list page is presented. On this page the bank employee have two options: selecting an existing customer or creating a new customer. In the first case the GUI displays the selected customer's account page. In the second case an empty customer account page is displayed. Figure 2 contains the summary of GUI operations and their data.

TASKS to BE PERFORMED:

Task 1. **Design** and **implement a graphical user interface (GUI)** through that the bank employee can manage employees' data and customers' bank account data using NetBeans Java GUI Builder. Fig. 2 presents the GUI manage data and invoke operations.

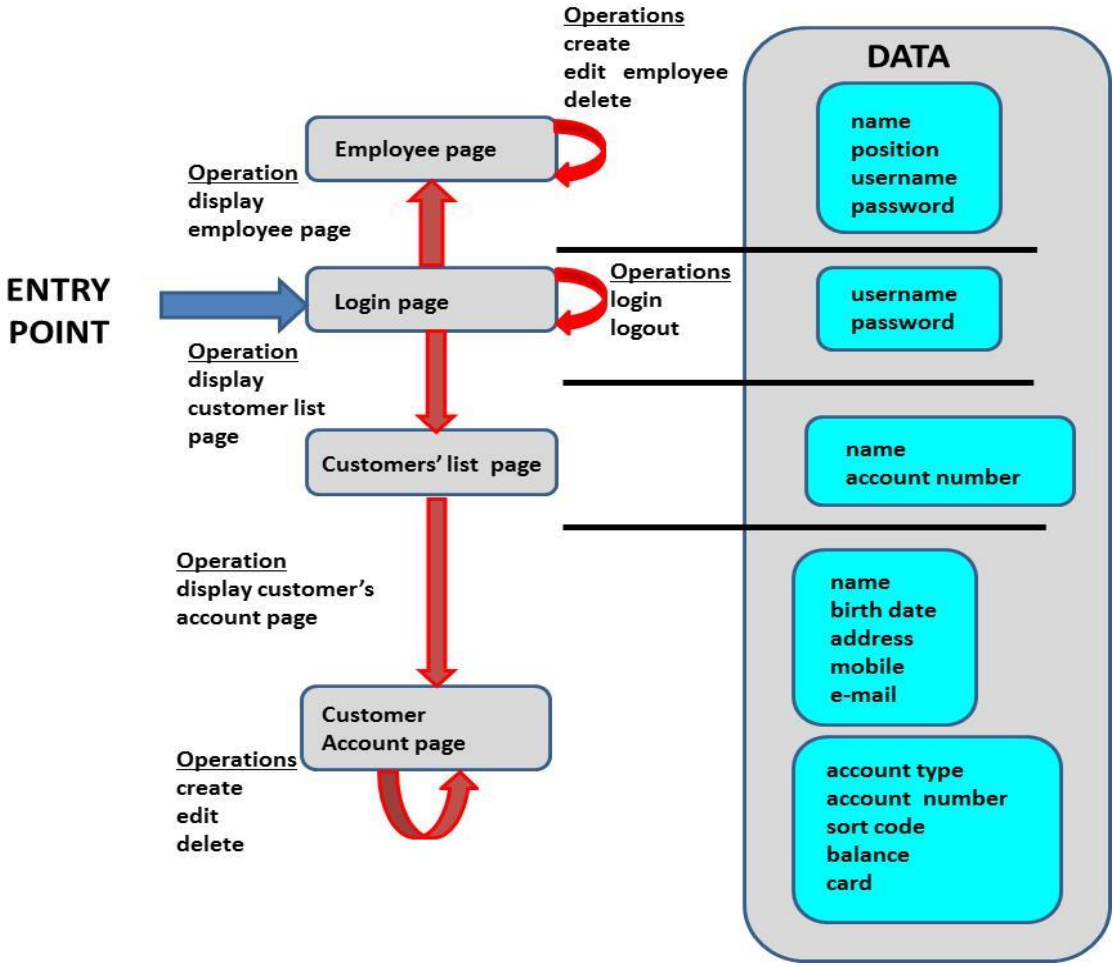

Fig. 2 e-Bank operations and the relavant data

Task 2. Develop the **UML use case diagram** of the e-bank application and the **sequence diagram** of the login, displaying customers' list and managing customer account operation to present communication between the GUI, web clients, the web services and the storage.

Task 3. Design and **implement** the **bank employee** and the **bank customer web service** to support the following operations:

bank employee service:

- managing bank employees' data — exception: none
- login/logout bank employees, — exception: wrong username/password

bank customer service:

- displaying customers' list, — exception: none
- managing customer's account — exception: invalid data
- creating new customer — exception: none

Task 4. Create a **web client** and **integrate** it with the bank GUI to invoke the above listed bank employee's and bank customer's operations:

Task 5. Install the **web services** and create at least two bank employees and at least two bank customers.

To submit

- GUI screenshots (Task 1)
- UML diagrams (Task 2)
- Web service Java code (Task3)
- Web client Java code (Task4)
- Employee and customer data (Task 5)

Coursework marking scheme

CW weighting% will be marked based on the following marking criteria:

Criteria	Mark per component	Mark provided	Comments
Task 1	20		
Task 2	20		
Task 3	25		
Task 4	25		
Task 5	10		
Total	**100**		