COMP2207 Assignment Instructions

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Module:	*Distributed Systems and Networks*			**Lecturer:**	*stw1e13*
Assignment:	*A Smart Metering System using Java RMI*			**Weighting:**	*30%*
Deadline:	*8th January 2015*	**Feedback:**	*29th January 2015*	**Effort:**	*45 hours*

Instructions

Objective

The purpose of this assignment is to design and implement a Smart Metering system using Java RMI. **This is a test of your understanding of Distributed Objects, and of Java RMI in particular. Therefore, solutions that use other distributed technologies will not be acceptable.**

In a Smart Metering system household gas and electricity meters have network connections to the power supply companies, and also to brokering agents that try to find a customer the best deal for their gas and electricity.

A meter sends its readings at regular intervals to the power company so that the power company can bill the customer. A power company can send commands to a meter so that it may perform a particular action (for example switch off the power supply if the customer has not paid their bill), and a meter can send alerts to the power company (for example if the customer has tried to tamper with the meter).

A customer can instruct a meter to contact a broker, and the broker will then try to find the customer a better deal for their gas and electricity. The tariff charged by a power company may vary depending upon how much energy the customer consumes, and how their consumption varies throughout the day (for example the price may be cheaper for energy consumed during the night). A broker can therefore ask a meter for a history of its readings, and ask a power company for information about its tariffs. If the customer decides to change supplier, the broker can register the meter with the new power company.

Some messages passed between the entities in the system are PUSHED (the sender decides when to send the data, usually this is when the data becomes available). Other messages are PULLED (the recipient requests the data, usually when it needs the data to perform some operation).

Detailed Requirements

There are three kinds of agent in the system: Meters, Power Companies, and Brokers.

Meters can:

1. Register with a Power Company (PUSH).
2. Unregister with a Power Company (PUSH).
3. Send readings to the currently registered Power Company (PUSH).
4. Send alerts to the currently registered Power Company (PUSH).
5. Receive commands from the currently registered Power Company (PUSH).
6. Send a request to find a better deal to a Broker (PUSH).

7. Send a history of readings to a Broker (PULL).
8. Send a request to accept a new deal to a Broker (PUSH).

Power Companies can:

1. Receive requests for a Meter to register with them (PUSH).
2. Receive requests for a Meter to unregister with them (PUSH).
3. Receive readings from a Meter that is registered with them (PUSH).
4. Receive alerts from a Meter that is registered with them (PUSH).
5. Send commands to a Meter that is registered with them (PUSH).
6. Send tariff information to a Broker (PULL).

Brokers can:

1. Receive a request to find a better deal from a Meter (PUSH).
2. Receive a history of readings from a Meter looking for a better deal (PULL).
3. Receive tariff information from a Power Company (PULL).
4. Receive a request to accept a new deal from a Meter (PUSH).
5. Register a Meter with a Power Company (PUSH).
6. Unregister a Meter with a Power Company (PUSH).

A Meter should only send readings to the Power Company it is registered with. A Meter should only be registered with one Power Company, and when it registers with a new Power Company, it should first unregister with the Power Company it is currently registered with. A Meter should only send readings to a Broker that it has asked to look for a better deal. A Meter can register directly with a Power Company, or a Broker can perform the registration on behalf of the Meter. A Meter should only send alerts to the Power Company it is registered with. A Meter should only receive commands from the Power Company it is registered with.

A Power Company should only receive readings from Meters that are registered with it. A Power Company should only receive alerts from Meters that are registered with it. A Power Company should only send commands to Meters that are registered with it.

A Broker should only receive readings from a Meter that has asked it to find a better deal. A Broker should only receive an accept new deal request from a Meter that has asked it to find a better deal, the Broker should then unregister the Meter from its current Power Company and register it with the new Power Company. A Broker cannot send commands to a Meter.

Assignment Instructions

You should design and implement a Java RMI based solution to the above requirements. Your solution should have separate applications for Meters, Power Companies, and Brokers, and they should communicate using remote method invocation.

You should pay careful attention to which communications are PUSHED, and which communications are PULLED.

You must show:

1. Power Companies can talk to multiple Meters simultaneously.
2. Meters can only be registered with one Power Company at a time.
3. Brokers can talk to multiple Meters and Power Companies simultaneously.
4. Failed communication links recover when connectivity is restored.
5. Meters can change Power Companies via a Broker.

You should concentrate on the messages passed between agents. Explicit details of the format of the meter readings and tariffs, and how the optimal tariff is determined, need NOT be included in your solution.

Your Smart Metering solution should be implemented in such a way that the applications for the Meters, Power Companies, and Brokers can all run on the same machine simultaneously. In other words, the person marking your solution must be able to run your complete solution on a single machine, and test that your Meters, Power Companies, and Brokers interact correctly. However, you must also indicate the changes needed to your Java code in order to run the different applications on different machines.

Submission

You should write a report describing the design, implementation and testing of your Smart Metering solution, and addressing the points identified above. **The report should focus on your use of the Distributed Objects paradigm and of Java RMI, and should motivate the choices made in the design and implementation of your Smart Metering solution.** Your report should not exceed 10 pages of single spaced A4 using 12pt font size (excluding the cover page and the appendix) and should provide the following information:

1. An explanation of how the classes for the Smart Metering solution have been designed, implemented, and tested (40%).
2. An explanation of how your Smart Metering solution allows a Meter to change Power Company via a Broker, and how it has been demonstrated. (25%).
3. An explanation of how the requirements that a Power Company can be connected to multiple Meters, and how a Broker can be connected to multiple Meters and Power Companies, have been demonstrated (10%).
4. An explanation of how lost connections can be handled (10%).
5. An explanation of how the Smart Metering solution can be implemented on several machines (5%).
6. Conclusions, including a critical evaluation of the suitability of Distributed Objects for the implementation of a Smart Metering system (10%).

Screenshots showing your application running can be included in the Appendix.

Your submission should consist of a PDF file containing your report, and a ZIP file containing the Java sources of your entire implementation. Please ensure that your Java sources can be compiled and executed on a lab machine, and that the Java sources contain sufficient comments to explain your code.

Please submit using the C-Bass electronic hand-in system (http://handin.ecs.soton.ac.uk) by 4pm on the due date.

Relevant Learning Outcomes

1. *A6. Distributed objects, including use through Java RMI.*
2. *C2. Build a Distributed Objects solution in Java.*

Marking Scheme

Criterion	**Outcomes**	**Total**
Classes for the Smart Metering solution and explanation of how they have been designed, implemented, and tested.	*1,2*	*40*
Explanation of how the Smart Metering solution allows a Meter to change Power Company via a Broker, and how it has been demonstrated.	*1,2*	*25*
Explanation of how the requirement that a Power Company can be connected to multiple Meters, and how a Broker can be connected to multiple Meters and Power Companies have been demonstrated.	*1,2*	*10*
Explanation of how lost connections can be handled.	*2*	*10*
Explanation of how the Smart Metering solution can be implemented on several machines.	*2*	*5*
Conclusions, including a critical evaluation of the suitability of Distributed Objects for implementing a Smart Metering system.	*1*	*10*

Feedback

We will aim to give you feedback, in the form of written comments on your solution, by Thursday 29th January. The latest you will receive feedback is Thursday 5th February.

Late Penalties

Late submissions will be penalised at 10% per working day. Work submitted more than 5 days late will receive a mark of zero. Please note the University regulations regarding academic integrity.

COMP2207 Assignment Feedback

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Module:	*Distributed Systems and Networks*	**Marked by:**	
Assignment:	*A Smart Metering System using Java RMI*	**Weighting:**	*30%*
Student:		**Submitted:**	

Marking Scheme

Criterion	**Outcomes**	**Total**
Classes for the Smart Metering solution and explanation of how they have been designed, implemented, and tested.	*1,2*	*40*
Explanation of how the Smart Metering solution allows a Meter to change Power Company via a Broker, and how it has been demonstrated.	*1,2*	*25*
Explanation of how the requirement that a Power Company can be connected to multiple Meters, and how a Broker can be connected to multiple Meters and Power Companies have been demonstrated.	*1,2*	*10*
Explanation of how lost connections can be handled.	*2*	*10*
Explanation of how the Smart Metering solution can be implemented on several machines.	*2*	*5*
Conclusions, including a critical evaluation of the suitability of Distributed Objects for implementing a Smart Metering system.	*1*	*10*

Qualitative Feedback

Good features of your work

…

…

…

…

Advice and suggestions for improvement

…

…

…

…

Marks Breakdown

Criterion	**N/A**	**Inferior**	**Marginal**	**Good**	**Excellent**	**Superb**	**Total**
Classes for the Smart Metering solution and explanation of how they have been designed, implemented, and tested.							
Explanation of how the Smart Metering solution allows a Meter to change Power Company via a Broker, and how it has been demonstrated.							
Explanation of how the requirement that a Power Company can be connected to multiple Meters, and how a Broker can be connected to multiple Meters and Power Companies have been demonstrated.							
Explanation of how lost connections can be handled.							
Explanation of how the Smart Metering solution can be implemented on several machines.							
Conclusions, including a critical evaluation of the suitability of Distributed Objects for implementing a Smart Metering system.							
Overall:		**Days Late:**		**Penalty:**		**Final Mark:**	

Marks are provisional until the June exam board.

Retain all assignments and feedback sheets until then.