MINISTRY OF EDUCATION AND SCIENTIFIC RESEARCH

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA

FACULTY OF AUTOMATION AND COMPUTER SCIENCE
COMPUTER SCIENCE DEPARTMENT

DISTRIBUTED SYSTEMS

Assignment 2

Remote Procedure Call (RPC)

A2.2: RPC application using distributed objects (Java RMI or .NET)

Ioan Salomie
Marcel Antal
Teodor Petrican

Tudor Cioara
Claudia Daniela Pop

Ionut Anghel
Dorin Moldovan
Ciprian Stan

2018

Contents

1. Requirements

Design, implement and test a client-server distributed system that uses RPC to compute taxes and selling prices for cars.

1.1. Functional requirements:

- Users introduce the information of their cars using a simple form (Web or Desktop):
  - int year – fabrication year
  - int engineSize – engine size
  - double price- purchasing price
- The application uses RPC to send the car information to the distributed object from the server that computes the following information depending on the client request:
  - Tax for a car

$$tax = \left(\frac{engineSize}{200}\right) * sum$$

Where sum depends on the engine size:

Engine Size	Sum (in RON)
<1600	8
1601-2000	18
2001-2600	72
2601-3000	144
>3001	290

  - Selling price for a car

$$price_{selling} = \begin{cases} price_{purchasing} - \frac{price_{purchasing}}{7} * (2018 - year) & if\ 2018 - year < 7 \\ 0 & otherwise \end{cases}$$

- The result of the invoked operation, tax, respectively selling price, is displayed on the client GUI.

1.2. Implementation technologies:

- Use the following technologies: JAVA RMI or .NET Remoting.

2. Deliverables

- A solution description document (about 4 pages, Times New Roman, 10pt, Single Spacing) containing:
  a) Conceptual architecture of the distributed system.
  b) UML Deployment diagram.
  c) Readme file containing build and execution considerations.
- Source files. The source files will be uploaded on the personal ***bitbucket*** account created at the ***Lab resources*** laboratory work, following the steps:
  - Create a repository on *bitbucket* with the exact name: *DS2018_Group_FirstName_LastName_Assignment_2*

- Push the source code and the documentation (push the code not an archive with the code or war files)
- Share the repository with the user *utcn_dsrl*

➢ The source files will be uploaded on the personal ***bitbucket*** account created at the ***Lab resources*** laboratory work)

3. Evaluation

3.1. Assignment Related Basic Questions:

During project evaluation and grading you will be asked details about the following topics:

- Distributed objects middleware components: Stub, Skeleton, Dispatcher, etc
- JAVA RMI architecture
- .NET Remoting architecture
- Distributed objects vs Local objects
- Distributed objects problems: security, latency, life-cycle, etc

3.2. Grading

The assignment will be graded as follows:

Points	Requirements
5 p	**Minimum to pass** • Client – Server application using Java RMI or .NET Remoting with one distributed object and at least one method implemented (tax or price) • Documentation
2 p	Simple GUI (Desktop form or Web)
1 p	Both methods implemented in a distributed object
2p	Answers of Reinforcement Learning Questions of A1.1

4. Bibliography

1. http://www.coned.utcluj.ro/~salomie/DS_Lic/
2. Lab Book: I. Salomie, T. Cioara, I. Anghel, T.Salomie, *Distributed Computing and Systems: A practical approach*, Albastra, Publish House, 2008, ISBN 978-973-650-234-7
3. Java RMI: https://docs.oracle.com/javase/tutorial/rmi/
4. .NET Remoting: http://www.codeproject.com/Articles/14791/NET-Remoting-with-an-easy-example