



---

**TECHNICAL UNIVERSITY**

OF CLUJ-NAPOCA, ROMANIA

**FACULTY OF AUTOMATION AND COMPUTER SCIENCE  
COMPUTER SCIENCE DEPARTMENT**

# DISTRIBUTED SYSTEMS

## Assignment 1

### Request-Reply Communication Paradigm

### Online Energy Utility Platform

Marcel Antal  
Alexandru Rancea

Tudor Cioara  
Claudia Daniela Antal

Cristina Pop  
Ionut Anghel

2022

## Contents

1. Requirements .....	2
1.1. Functional requirements: .....	2
1.2. Implementation technologies: .....	2
1.3. Non-functional requirements: .....	3
2. Deliverables .....	3
3. Evaluation .....	3
3.1. Assignment Related Basic Questions: .....	3
3.2. Grading .....	3
4. Bibliography .....	4

## 1. Requirements

An online platform should be designed and implemented to manage users, their associated smart energy metering devices, and the monitored data from each device. The system can be accessed by two types of users after a login process: administrator (manager), and clients. The administrator can perform CRUD (Create-Read-Update-Delete) operations on **user accounts (defined by ID, name, role: admin/client)**, registered smart energy metering **devices (defined by ID, description, address, maximum hourly energy consumption)**, and on **the mapping of users to devices (each user can own one or more smart devices in different locations)**. After the mapping is done, for each device the energy consumption is stored on hourly basis as tuples of the form **<timestamp, energy consumption>** in the database.

### 1.1. Functional requirements:

- Users log in. Users are redirected to the page corresponding to their role.
- Administrator/Manager Role:
  - CRUD operations on users and devices.
  - Create user-device mappings.
- User/Client Role
  - Can view on his/her page all the associated devices.
  - Can view the daily energy consumption for each of his/her associated devices as line charts or bar charts per day (OX- hours; OY- energy value [kWh] for that hour). The day should be selected from a calendar.
- The users corresponding to one role will not be able to enter the pages corresponding to the other role (e.g., by log-in and then copy-paste the admin URL to the browser).

### 1.2. Implementation technologies:

- Use the following technologies: REST services for backend application (Java Spring REST or .NET Web API) and JavaScript-based frameworks for client application (Angular or ReactJS).

## 1.3. Non-functional requirements:

- Security: use authentication to restrict users to access the administrator pages (cookies, session, etc.)

## 2. Deliverables

- A solution description document (about 4 pages, Times New Roman, 10pt, Single Spacing) containing:
  - a) Conceptual architecture of the online platform.
  - b) DB design.
  - c) UML Deployment diagram.
  - d) Readme file containing build and execution considerations.
- Source files. The source files and the database dump will be uploaded on the personal GitHub account created at the Lab resources laboratory work, following the steps:
  - Create a private repository on *GitHub* with the exact name:  
*DS2022\_Group\_LastName\_FirstName\_Assignment\_Number*
  - 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 *utcndsrl*

## 3. Evaluation

## 3.1. Assignment Related Basic Questions:

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

- URI and URL
- Web Clients and Web Servers
- HTTP protocol and messages
- HTTP methods
- HTML web forms
- Query strings
- HTML Web Storage API / Cookies
- Session
- JavaScript – events, synchronous/asynchronous requests, form validation.
- Object-Relational Mapping (ORM)
- REST Services

## 3.2. Grading

The assignment will be graded as follows:

Points	Requirements
4 p	<b>Minimum to pass</b> <ul style="list-style-type: none"> <li>• REST Services and Frontend for CRUD operations for Administrator Role</li> <li>• Database</li> <li>• Documentation</li> </ul>

	• Correct answers to 3.1 questions
1 p	Log-in with redirect (administrator/client)
1 p	Administrator associates devices to users
2 p	Full implementation of client role: <ul style="list-style-type: none"> <li>- can view his/her devices,</li> <li>- can view the chart showing their daily energy consumption for their devices</li> </ul>
1p	Minimum Security: The users will not be able to enter other users' pages (filters according to role and sessions)
9p + 1p = 10p	Total

#### 4. Bibliography

1. <https://dsrl.eu/courses/sd/>
2. Lab Book: M. Antal, C. Pop, D. Moldovan, T. Petrican, C. Stan, I. Salomie, T. Cioara, I. Anghel, Distributed Systems – Laboratory Guide, Editura UTPRESS Cluj-Napoca, 2018 ISBN 978-606-737-329-5, 2018,  
<https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/329-5.pdf>
3. 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
4. Hibernate:
  - a. <http://www.tutorialspoint.com/hibernate/>
  - b. <http://www.javatpoint.com/hibernate-tutorial>
5. Maven: <https://maven.apache.org/>