Distributed Applications

1. **Course number and name:** 020APDES4/020DAPES4 Distributed Applications

2. **Credits and contact hours:** 4 ECTS credits, 2x1:15 contact hours

3. **Name of course coordinator:** Rima Kilany

4. **Instructional materials:** course slides (Moodle); lab sessions (Moodle)

5. **Specific course information**
   a. **Catalog description:**
      This course raises students' awareness about the different software architecture patterns and enterprise applications patterns. This course also explains the need for using middleware in the context of object-oriented distributed applications (Java RMI), as well as distribution on the web. It covers distributed Jakarta EE components (Stateless and Stateful Session beans), as well as Message Driven Beans for asynchronous communication. It details Object Relational Mapping ORM and its implementation with JPA (Java persistence API) to manage persistence and access to Relational and non-relational databases. As for distributed web applications, this course covers Servlets, as well as the implementation, testing and deployment of REST web services respecting level 3 of the Richardson maturity model, and respecting the HATEOAS principle, enabling students to compare them to SOAP web services. The course covers the documentation of REST Web APIs using the Open API Specification (Swagger). It introduces containers and explains their importance when deploying applications on-premise or on the cloud.

   b. **Prerequisites:** None

   c. **Required** for CCE Software Engineering Option students; **Selected Elective** for students in the CCE Artificial Intelligence and Telecommunication Networks Options.

6. **Educational objectives for the course**
   a. **Specific outcomes of instruction:**
      - Evaluate the challenges of the distribution of applications, as well as the advantages / disadvantages of the various types of communication middleware.
      - Implement a distributed application using distributed objects (Java RMI)
      - Implement a distributed application using JEE (Stateless and Stateful session beans, asynchronous Message Driven Beans) and ORM / JPA (Java persistence API) to manage persistence and access to SQL and NoSQL databases.
      - Understand SOAP Web Services and compare them to REST Web services

- Implement and deploy a REST Web API respecting level 3 of the Richardson maturity model, and respecting the HATEOAS principle.
- Test the REST Web API using Postman.
- Document the Web API using the Open API Specification (Swagger).

b. PI addressed by the course:

PI	1.2	1.3	2.3
Covered	x	x	x
Assessed	x	x	

7. Brief list of topics to be covered

- Distributed Applications: Evolution and needs (1 lecture)
- Middleware – Evolution - Object Request Brokers Properties (1 lecture)
- Distributing objects with Java RMI (1 lecture)
- Implementing a distributed Java RMI application (1 lecture)
- Personalize RMI transport layer and Distributed Garbage Collecting (2 lectures)
- Introduction to JEE (Java Enterprise Edition) framework, APIs such as JNDI, and EJB components (Differences between Session Beans, Message Driven Beans, and JPA entities) (1 lecture)
- Session beans in detail: Stateless, Stateful, Pooling, Passivation, application server configuration, callback methods (2 lectures)
- Development of a Session bean and deployment on an Application server (1 lecture)
- JPA entities in detail, persistence frameworks, configuration, and database mapping (2 lectures)
- Servlets 31 (1 lecture)
- Implement a solution showing session beans and JPA entities interaction (1 lecture)
- Advanced notions on persistence: Inheritance, polymorphism (1 lecture)
- Asynchronous Middleware - JMS - Message Driven Beans (1 lecture)
- Configuration of destinations (Topic, Queue) on the Glassfish application server, and implementation of a Message Driven Bean (1 lecture)
- Transactions in an EJB environment (1 lecture)
- Security in an EJB environment (1 lecture)
- Intro to XML and to Web Services (1 lecture)
- SOAP: protocol stack, context of use Comparison with all distribution solutions (1 lecture)
- REST Web Services - Comparison with SOAP (2 lectures)
- Development of a REST Web Service in Java (1 lecture)
- Documentation and test of a REST web service with POSTMAN (2 lectures)
- Introduction to containerization. (1 lecture)