SYLLABUS

1. Data about the program of study

1.1 Institution	The Technical University of Cluj-Napoca
1.2 Faculty	Faculty of Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Bachelor of Science
1.6 Program of study / Qualification	Computer science / Engineer
1.7 Form of education	Full time

2. Data about the subject

2.1 Subject name	***Distributed systems***			Subject code	**45**
2.2 Course responsible / lecturer	Assoc. prof. dr. eng. Pop Cristina-Bianca - Cristina.POP@cs.utcluj.ro				
2.3 Teachers in charge of seminars / Laboratory / project	Asist. drd. eng. Antonesi Gabriel - Gabriel.ANTONESI@cs.utcluj.ro				
2.4 Year of study	IV	2.5 Semester	7	2.6 Type of assessment (E - exam, C - colloquium, V – verification)	E
2.7 Subject category	Formative category: DF – fundamental, DS – speciality, DC – complementary				DS
	Optionality: DOB – mandatory, DOP – optional (alternative), DFA – optional (free choice)				DOB

3. Estimated total time

3.1 Number of hours per week	5	of which:	Course	2	Seminars	-	Laboratory	2	Project	1
3.2 Number of hours per semester	70	of which:	Course	28	Seminars	-	Laboratory	28	Project	14
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										30
(b) Supplementary study in the library, online and in the field										10
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										28
(d) Tutoring										-
(e) Exams and tests										12
(f) Other activities:										-

3.4 Total hours of individual study (suma (3.3(a)…3.3(f)))	80
3.5 Total hours per semester (3.2+3.4)	150
3.6 Number of credit points	6

4. Pre-requisites (where appropriate)

4.1 Curriculum	Computer networks, Software Design, Programming Techniques, Databases
4.2 Competence	Ability to analyze and design a local network using available simulators. Ability to design an application using layered architecture. Ability to code using OOP languages. Ability to design and implement a relational database, as well as to write queries both in SQL and in an ORM framework.

5. Requirements (where appropriate)

5.1. For the course	Whiteboard, projector, computer, Internet connection
5.2. For the applications	Computers, software specific tools.

6. Specific competence

6.1 Professional competences	**C4 -** Improving the performance of the hardware, software and communication systems (2 credits) • **C4.1** - Identifying and describing the defining elements of the performances of the hardware, software and communication systems • **C4.2** - Explaining the interaction of the factors that determine the performances of the hardware, software and communication systems • **C4.3 -** Applying the fundamental methods and principles for increasing the performance of the hardware, software and communication systems • **C4.4** - Choosing the criteria and evaluation methods of the performances of the hardware, software and communication systems • **C4.5** - Developing professional solutions for hardware, software and communication systems based on performance optimization **C5 -** Designing, managing the lifetime cycle, integrating and ensuring the integrity of hardware, software and communication systems (2 credits) • **C5.1** - Specifying the relevant criteria regarding the lifetime cycle, quality, security and the computing system's interaction with the environment and the human operator • **C5.2** - Using interdisciplinary knowledge for adapting the computing system to the specific requirements of the application field • **C5.3** - Using fundamental principles and methods for ensuring the security, the safety and ease of exploitation of the computing systems • **C5.4** - Proper utilization of the quality, safety and security standards in the field of information processing • **C5.5** - Creating a project including the problem's identification and analysis, its design and development, and also proving an understanding of the basic quality requirements **C6 -** Designing intelligent systems (1 credit) • **C6.1** - Describing the components of intelligent systems • **C6.2** - Using domain-specific tools for explaining and understanding the functioning of intelligent systems • **C6.3** - Applying the fundamental methods and principles • for specifying solutions for typical problems using intelligent systems • **C6.4** - Choosing the criteria and evaluation methods for the quality, performance and limitations of intelligent systems • **C6.5** - Developing and implementing professional projects for intelligent systems
6.2 Cross competences	N/A

7. Expected Learning Outcomes

Knowledge	Describe, identify, and summarize advanced concepts, models, and methods related to complex information systems, distributed systems, as well as the principles of management and methodology of IT projects.
Skills	Apply and design integrated IT solutions, including databases, networks, software systems, distributed systems, and security mechanisms, using methods of analysis, design, implementation, and documentation.
Responsibilities and autonomy	Takes responsibility for the quality, security, and correctness of the developed solutions and demonstrates autonomy in organizing design and learning activities, collaborating effectively in teams and adapting to new requirements and technologies.

8. Discipline objective (as results from the *key competences gained*)

8.1 General objective	Capacity to develop and implement distributed software systems
8.2 Specific objectives	- Capacity of designing distributed systems at both architectural and components' level by using the main concepts and paradigms of distributed systems as well as their relationship with other computer science disciplines. - Capacity of identifying the main models, techniques and technologies that can be used in the design of distributed systems by considering a set of functional and non-functional specifications and constraints. - Capacity of using Java and .NET technologies for implementing distributed systems. - Capacity of using distributed communication models and paradigms. - Capacity of using techniques for data distribution and for the management of distributed transactions. - Capacity of building Web applications using technologies such as Spring and React. - Capacity to develop client applications for distributed systems using JavaScript based technologies. - Capacity to design, develop, integrate and organize the deployment of a distributed application, considering the involved servers and network settings.

9. Contents

9.1 Lectures	Hours	Teaching methods	Notes
Introduction in Distributed Systems, Characterization of Distributed Systems	2	-Using modern multimedia teaching methods and direct access to internet. -Challenging questions during lecturers. -Students are invited to collaborate in research projects. -Personal assistance hours during the semester and before the exam.	
Introduction – use case of Google data center	2		
Non-Functional Requirements, QoS, Metrics	2		
Inter-process Communication paradigms	2		
Communicating Entities in Distributed Systems: client-server, Peer-to-peer	2		
Distributed Computation Model and Organization	2		
Time and Causality, Logic Clocks	2		
Global States, Snapshots, Distributed Algorithms	2		
Distributed Data Processing – Concepts, Reference Architectures	2		
Data Distribution Techniques	2		
Distributed Transactions Management	2		
Concurrency Control	2		
Cloud Computing basic Concepts	2		
Design elements of cloud-based systems	2		

Bibliography:

1. G. Coulouris, J.Dollimore, T.Kindberg – Distributed Systems. Concepts and Design (5th edition), Addison Wesley, 2014
2. A. Tanenbaum, M. van Steen – Distributed Systems, Createspace Independent Publishing Platform, 2017
3. Tudor Cioara, Marcel Antal, Cristina Pop - Lecture Notes, Lab Notes Project Notes and Assignments https://dsrl.eu/courses/sd/

9.2 Applications - Seminars / Laboratory / Project	Hours	Teaching methods	Notes
Introduction to the resources and requirements of the laboratory and the project.	2	- Predefined assignments and examples. -Short presentation of lab assignments, discussions about assignments.	
Laboratory – Assignment 1 – Development of distributed applications based on microservices using the Spring and React frameworks.	8		
Laboratory – Assignment 2 – Development of distributed applications using the indirect communication paradigm based on	6		

message queues (RabbitMQ).		- Design and implementation of Lab Assignments and Project -Tool for Continuous Integration, Deployment and Testing of the projects	
Laboratory – Assignment 3 – Development of distributed applications using WebSockets.	4		
Evaluation of laboratory assignments	8		
Project – Deployment of distributed applications using Docker	4		
Project – Techniques for implementing security in distributed systems	2		
Project – Use of the CI/CD pipeline in the development of distributed systems	2		
Project – Application integration	2		
Project evaluation	4		

Bibliography:

1. Ioan Salomie, Tudor Cioara, Ionut Anghel, Tudor Salomie – Distributed Computing and Systems – A practical Approach, Albastra Publ. House, 2008
2. 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. Tudor Cioara, Marcel Antal, Cristina Pop - Lecture Notes, Lab Notes Project Notes and Assignments https://dsrl.eu/courses/sd/

**Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.*

10. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

Distributed Systems is a subject of the domain "Computers and Information Technology". It teaches students about the development and implementation of distributed software systems. The content was developed based on the analysis of similar disciplines from other universities as well as based on the requirements of the IT employees. The content was also evaluated by Romanian governmental agencies CNEAA and ARACIS.

11. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	The level of assimilation of the knowledge about distributed systems, presented during the course	Written Exam	50%
Seminar	-	-	-
Laboratory Project	- Capacity of designing distributed systems at both architectural and components' level by using the main concepts and paradigms of distributed systems as well as the relationships with other computer science domains. - Capacity of identifying the main models, techniques and technologies that could be used in the design of distributed systems by considering a set of functional and non-functional specifications and constraints. - Individual activity during course, lab and project - Attendance	- Assignments evaluation, Project evaluation. -Tool for Continuous Integration, Deployment and Testing of Distributed Applications.	35% 15%

Minimum standard of performance:
- To be able to design and implement distributed software systems.

Grade calculus: 35% laboratory + 15% project + 50% final exam

Conditions for participating in the final exam: Laboratory ≥ 5, Project ≥5

Handing over all laboratory assignments and obtaining a minimum grade of 5 on each assignment; attendance to at least 11 laboratory sessions.

Conditions for promotion: final exam ≥ 5

Date of filling in: 28.04.2026	Responsible	Title First name Last name	Signature
	Course	Assoc.prof.dr.eng. Cristina POP	
	Applications	Asist. drd. eng. Gabriel ANTONESI	

Date of approval in the department	Head of department, Prof.dr.eng. Rodica Potolea
Date of approval in the Faculty Council	Dean, Prof.dr.eng. Vlad Mureșan