Prof. Tudor Cioara, Conf. Cristina Pop, S.l. Marcel Antal
2025-2026

Overall Schedule - Laboratory and Project

Week	Laboratory Presentation	Laboratory Evaluation	Project Presentation	Project Evaluation
1	• Lab Resources • Assignment 1 presentation • Laboratory work 1.1: 3 – tier REST services		• Project requirements	
2	• Laboratory work 1.2: Micro-services Development			
3	Q&A		Project P1: • Deployment using Docker • Reverse proxy	
4	Intermediary evaluation			
		Assignment 1 – User and Device backend		Project P1 – deployed User and Device Service and databases on Docker with reverse proxy
	• Laboratory work 1.3: React App Development			
5	Q&A		• Security	
6	• Assignment 2 presentation • Laboratory work 2.1: Indirect Communication Using Queues	Assignment 1 – full application		Project P1 – deployed on Docker with reverse proxy and security integrated
7	• Laboratory work 2.2: Indirect Communication Using Topics			
8	Q&A		Project P2: • Deployment using Docker	
9	• Assignment 3 presentation • Laboratory work: Web Sockets	Assignment 2 – full application		Project P2 – deployed on Docker with

				reverse proxy and security integrated
10			• CI/CD on Cloud (*optional for the extra point*)	
11	Q&A		Project P3: • Deployment using Docker	
12		Assignment 3 – full application		Project P3 - deployed on Docker with reverse proxy and security integrated
13				Late evaluations
14				Late evaluations

Detailed Description - Laboratory and Project

Lab Resources

Key Technical Skills	Setup Software Stack Version Control (Git and Gitlab) CI/CD Tutorial Deployment on cloud (any cloud provider account is accepted) for 1 bonus point at the exam

Assignment 1 – tier REST services and React App

Basic DS Concepts	Client - Server Architecture Request-Reply Communication Paradigm HTTP Protocol and Methods HTTP state management mechanisms URI-based resource access (REST Services)
Key Technical Skills	HTTP state management mechanisms on Client Side (Session storage, Cookies) Authorization based on roles and Authentication (basic auth and JWT) Process Custom Queries for fast DB access (eager vs lazy load) Deployment of web application in Docker
Conceptual Architecture	Client application - 3-tier REST Server-side services
Technologies	React (Angular) + Spring REST Hibernate + Mysql/PostgreSQL Traefik Docker (mandatory)
Useful Links	https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/329-5.pdf https://spring.io/projects/spring-security

Assignment 2 - Indirect Communication Using Queues

Basic DS Concepts	Message Oriented Middleware

	Event-driven and asynchronous communication Pub/Sub vs queue-based models
Key Technical Skills	Using Queues as Message Buffers for Client-Server communication Synchronization events across microservices using the queue Deployment of client, server and middleware applications
Conceptual Architecture	Queue-based Message Oriented Middleware
Technologies	RabbitMQ
Useful Links	https://www.rabbitmq.com/tutorials/tutorial-one-spring-amqp https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/329-5.pdf

Assignment 3 – Web sockets and load balancing

Basic DS Concepts	Using Web Sockets for Asynchronous Client-Server Communication Load balancing
Key Technical Skills	Web sockets-based communication between admin and clients Deployment and scaling of multiple replicas using load balancing Implementing a rule-based or AI-driven chatbot for client support
Conceptual Architecture	Chat microservice with WebSocket communication, load balancing microservice
Technologies	Web sockets
Useful Links	https://spring.io/guides/gs/messaging-stomp-websocket/

Final Project

Basic DS Concepts	Architecture of large heterogeneous distributed application Non-Functional requirements of Distributed Systems Virtualization CI/CD Pipeline Spring security Reverse Proxy Load Balancing
Key Technical Skills	Scalability: Deployment using Docker Security: HTTPS and JWT over REST SSL Setting Time zone and keeping time consistent within application Deployment in Docker, network configuration, port forwarding Load balancing and reverse proxy
Conceptual Architecture	Service-based architecture of large distributed system
Technologies	Spring REST + React (or Angular) Hibernate + MySql / PostgreSQL RabbitMQ JSR 356 or the Java API for WebSocket HTTPS and JWT over REST and SSL, WS Security Docker for resource virtualization JWT-based authentication Traefik for reverse proxy
Useful Links	https://www.guru99.com/security-web-services.html https://traefik.io/traefik https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/329-5.pdf