Specification for the book of courses

Study program	Electrical Engineering and Computer Science		
Module	Computing and Informatics		
Type and level of studies	Undergraduate Academic Studies		
The name of the course	Distributed Systems		
Lecturer (for lectures)	Milovanović I. Emina, Stojanović M. Natalija		
Lecturer/associate (for exercises)	Stojanović M. Natalija, Veljanovski D. Aleksandar, Frtunić-Gligorijević B. Milena		
Lecturer/associate (for OFE)	Stojanović M. Natalija, Veljanovski D. Aleksandar, Frtunić-Gligorijević B. Milena		
Number of ECTS	6	**Course status (obligatory/elective)**	Obligatory
Prerequisites	Computer networks, Operating systems		
Course objectives	This is an introductory graduate level course in distributed systems. It will expose students to theoretical as well as practical aspects of designing such systems. The course covers fundamental models for distributed systems, inter process communication and how to handle synchronization, consistency, replication, fault tolerance and security in a distributed system		
Course outcomes	The students will be able to: explain important characteristics of distributed systems;describe architectural and fundamental models of distributed systems; explain and compare strategies for interprocess communication; explain and compare middleware models; explain the concept of logical time; use logical time to implement distributed algorithms.		
Course outline			
Theoretical teaching	Introduction. Examples of Distributed Systems–Trends in Distributed Systems – Focus on resource sharing – Challenges. Case study: World Wide Web. Communication in distributed systems. System Model. Inter process Communication – the API for internet protocols. External data representation. Remote Method Invocation and Objects. Remote procedure call. Case study: Java RMI – Group communication. Publish-subscribe systems – Message queues. Synchronization and replication. Clock synchronization. Event ordering: Logical clocks, vector clocks. Group communication: message ordering and message delivery, multicasting. Distributed mutual exclusion. Election algorithms. Fault Tolerance Byzantine Fault Tolerance- Detecting and Correcting Local Faults. Logging and Crash Recovery. Peer-to-peer Systems. Napster and its legacy. Routing overlays. Overlay case studies: Pastry, Tapestry. DHT p2p systems. Distributed File Systems. File service architecture . NFS. AFS. Google File System (GFS), Hadoop Distributed File System (HDFS)		
Practical teaching (exercises, OFE, study and research	Oral and laboratory exercises including programming tasks that exemplify problem statements examined in the course.		
Textbooks/references			
1	Maarten van Steen, Andrew S. Tanenbaum, Distributed Systems, third edition, 2017.		
2	A.S. Tanenbaum, M. van Steen, Distributed Systems: Principles and paradigms, Prentice Hall, 2007		
3	G. Coulouris, J. Dollimore, T. Kindberg , G. Blair, Distributed Systems: Concepts and Design, 5th Edition, Pearson, 2011.		
4	On line course material		
5			

Number of classes of active education per week during semester/trimester/year				
Lectures	**Exercises**	**OFE**	**Study and research work**	**Other classes**
2	2	1	0	0
Teaching methods	Lectures, oral and Lab exercises			

Grade (maximum number of points 100)			
Pre-exam duties	**Points**	**Final exam**	**Points**
Activity during lectures	5	**Written exam**	
Exercises	20	**Oral exam**	45
Colloquia	30		
Projects			