GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	High Performance Distributed Systems, PG_00054811		
Field of study	Informatics		
Date of commencement of studies	February 2023	Academic year of realisation of subject	2022/2023
Education level	second-cycle studies	Subject group	Optional subject group Subject group related to scientific research in the field of study
Mode of study	Full-time studies	Mode of delivery	at the university
Year of study	1	Language of instruction	English
Semester of study	1	ECTS credits	4.0
Learning profile	general academic profile	Assessment form	assessment
Conducting unit	Department of Computer Architecture -> Faculty of Electronics, Telecommunications and Informatics		
Name and surname of lecturer (lecturers)	Subject supervisor	dr inż. Andrzej Sobecki	
	Teachers	dr inż. Tomasz Boiński dr inż. Mariusz Matuszek	

Lesson types and methods of instruction	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	30.0	0.0	0.0	30.0	0.0	60
	E-learning hours included: 0.0						

Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan	Participation in consultation hours	Self-study	SUM
	Number of study hours	60	2.0	38.0	100

Subject objectives	The aim of the course is to introduce students to the subject of distributed applications development, as well as distributed data collection and processing systems. In addition, as part of the course, students will be familiarized with implementation platforms used in industry, whose task is to manage sets of distributed application components.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[K7_W41] Knows and understands, to an increased extent, the standards, production methods, life cycle and development trends of software as well as information systems and applications.	The student is able to recognize the need to use a distributed microservice architecture and knows the basic principles of service modeling. At the same time, the student acquires skills in the use of asynchronous methods of communication between services.	[SW1] Assessment of factual knowledge
[K7_W07] Knows and understands, to an increased extent, the general principles of creating and developing forms of individual entrepreneurship.	The student understands the division of responsibilities in development teams using microservice architecture and knows the rules of cooperation with other teams.	[SW1] Assessment of factual knowledge
[K7_W43] Knows and understands, to an increased extent, the nformal, technical and social aspects of the operation of complex information systems in the information society and in the global information n infrastructure.	The student is able to deliver his software in such a way that cooperation of a large group of programmers is possible. The containerization tools used make it possible to standardize the description of the environment and facilitate the transfer of the application code between programmers.	[SW1] Assessment of factual knowledge
[K7_W03] Knows and understands, to an increased extent, the construction and operating principles of components and systems related to the field of study, including theories, methods and complex relationships between them and selected specific issues - appropriate for the curriculum.	The student knows and describes various application development architectures. He knows the differences, advantages and disadvantages of using monolithic layered architectures and target distributed architectures.	[SW1] Assessment of factual knowledge
[K7_U06] can analyse the operation of components, circuits and systems related to the field of study; measure their parameters; examine technical specifications; interpret obtained results and draw conclusions	The student is able to use platforms for distributed data collection and processing in order to increase the efficiency of the created applications.	[SU1] Assessment of task fulfilment

Subject contents

Application scalability, deployment; Distributed application architectures (monolith -> micro services (CQRS / Event Sourcing / Saga); containerization of services -> docker, docker-compose, docker swarm, kubernetes, deployment and maintenance of a distributed application -> monitoring (clusters / clouds Computational OpenSTack / AWS) Monitoring -> Sentry / Jaeger / Promethes + Grafana / Load balancery / Queue systems; Locust.io / Jmeter load testing tools Distributed file systems HDFS (Hive) / IPFS Distributed databases (Hbase / Neo4j, ArangoDB) Blockchain - > Bitcoin / Ethereum / Stellar / GRP (graph) Distributed computing environment (Apache Spark / YARN -> JupyterLab -> PySpark -> .net context submit)

Prerequisites and co-requisites

1. The student must have knowledge and programming skills in .net or java technologies.2. The student must understand the methods of communication used on the Internet3. The student must have knowledge and skills in implementing the application on the target server in the containerized version.

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
lecture	50.0%	50.0%
project	50.0%	50.0%

Recommended reading	Basic literature	1. Cloud Native DevOps with Kubernetes, John Arundel, Justin Domingus2. Kubernetes Patterns: Reusable Elements for Designing Cloud-Native Applications, Bilgin Ibryam, Roland Huß3. KUBERNETES: A Simple Guide to Master Kubernetes for Beginners and Advanced Users (2020 Edition), Brian Docker4. Hands-On Docker for Microservices with Python: Design, deploy, and operate a complex system with multiple microservices using Docker and Kubernetes, Jaime Buelta5. gRPC: Up and Running: Building Cloud Native Applications with Go and Java for Docker and Kubernetes, Kasun Indrasiri, Danesh Kuruppu6. The Kubernetes Book, Nigel Poulton7. Hands-On Microservices with C# 8 and .NET Core 3: Refactor you monolith architecture into microservices using Azure, 3rd Edition, Gaurav Aroraa, Ed Price8. Pro ASP.NET Core 3: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages, Adam Freeman9. Practical Microservices Architectural Patterns - Event-Based Java Microservices with Spring Boot and Spring Cloud, Binildas Christudas10. Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith, Sam Newman11. Practical Microservices: Build Event-Driven Architectures with Event Sourcing and CQRS, Ethan Garofolo12. Architecting Modern Data Platforms, Jan Kunigk, Ian Buss, Paul Wilkinson & Lars George13. Advanced Analytics with Spark, Sandy Ryza, Uri Laserson, Sean Owen & Josh Wills14. Big Data Analytics with Hadoop 3, Sridhar Alla,15. Modern Big Data Processing with Hadoop, V. Naresh Kumar Prashant Shindgikar
	Supplementary literature	1. Modern Big Data Processing with Hadoop, V. Naresh Kumar Prashant Shindgikar
	eResources addresses	Adresy na platformie eNauczanie:
Example issues/ example questions/ tasks being completed	Application scalability, deployment; Distributed application architectures (monolith -> micro services (CQRS / Event Sourcing / Saga); containerization of services -> docker, docker-compose, docker swarm, kubernetes, deployment and maintenance of a distributed application -> monitoring (clusters / clouds Computational OpenSTack / AWS) Monitoring -> Sentry / Jaeger / Promethes + Grafana / Load balancery / Queue systems; Locust.io / Jmeter load testing tools Distributed file systems HDFS (Hive) / IPFS Distributed databases (Hbase / Neo4j, ArangoDB) Blockchain - > Bitcoin / Ethereum / Stellar / GRP (graph) Distributed computing environment (Apache Spark / YARN -> JupyterLab -> PySpark -> .net context submit)	
Work placement	Not applicable	