



Karta przedmiotu

Nazwa i kod przedmiotu	High Availability Distributed Systems, PG_00063936						
Kierunek studiów	Informatyka (studia w jęz. angielskim)						
Data rozpoczęcia studiów	październik 2024 r.		Rok akademicki realizacji przedmiotu		2024/2025		
Poziom kształcenia	II stopnia		Grupa zajęć		Grupa zajęć fakultatywnych Grupa zajęć specjalnościowych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne		Sposób realizacji		na uczelni		
Rok studiów	1		Język wykładowy		angielski		
Semestr studiów	2		Liczba punktów ECTS		3.0		
Profil kształcenia	ogólnoakademicki		Forma zaliczenia		zaliczenie		
Jednostka prowadząca	Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Architektury Systemów Komputerowych						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr inż. Mariusz Matuszek				
	Prowadzący zajęcia z przedmiotu		mgr inż. Hammed Mojeed dr inż. Mariusz Matuszek				
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	0.0	15.0	0.0	30
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	30		9.0		36.0	75
Cel przedmiotu	Celem przedmiotu jest wprowadzenie studentów w tematykę wytwarzania aplikacji rozproszonych, a także systemów gromadzenia i przetwarzania danych w sposób rozproszony. Ponadto w ramach przedmiotu studenci zostaną zaznajomieni z platformami wdrożeniowymi stosowanymi w przemyśle, których zadaniem jest zarządzanie zbiorami rozproszonych komponentów aplikacyjnych.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K7_W10] zna i rozumie w pogłębionym stopniu podstawowe procesy zachodzące w cyklu życia urządzeń, obiektów i systemów technicznych oraz metody wspomagania procesów i funkcji, specyficzne dla kierunku studiów	Student realizuje projekt, w którym poznaje pełen cykl wytwarzania i życia aplikacji rozproszonej.	[SW2] Ocena wiedzy zawartej w prezentacji
	[K7_W11] zna i rozumie w pogłębionym stopniu ogólne zasady tworzenia i rozwoju form indywidualnej przedsiębiorczości oraz ekonomiczne, prawne i inne uwarunkowania różnych rodzajów działań związanych z nadaną kwalifikacją, w tym zasady ochrony własności przemysłowej i prawa autorskiego	Tematyka realizowanego projektu zanurzona jest w środowisku mikroprzedsiębiorczości.	[SW3] Ocena wiedzy zawartej w opracowaniu tekstowym i projektowym
	[K7_U12] potrafi w pogłębionym stopniu analizować działanie elementów, układów i systemów związanych z kierunkiem studiów oraz mierzyć ich parametry i badać charakterystyki techniczne, a także planować i przeprowadzać eksperymenty związane z kierunkiem studiów, w tym symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski	Student projektuje mechanizmy komunikacyjne i równoważenia obciążenia w wytwarzanym projekcie aby dynamicznie reagować na zmiany w popycie i podaży usług.	[SU4] Ocena umiejętności korzystania z metod i narzędzi [SU3] Ocena umiejętności wykorzystania wiedzy uzyskanej w ramach przedmiotu
	[K7_W03] zna i rozumie w pogłębionym stopniu budowę i zasady działania komponentów i systemów związanych z kierunkiem studiów, w tym teorie, metody i złożone zależności między nimi oraz wybrane zagadnienia szczegółowe – właściwe dla programu kształcenia	Student zna i opisuje różne architektury wytwarzania aplikacji. Zna różnice, zalety i wady ze stosowania monolitycznych architektur warstwowych i docelowych architektur rozproszonych.	[SW1] Ocena wiedzy faktograficznej
Treści przedmiotu	Skalowalność aplikacji, deployment; Architektury aplikacji rozproszonych (monolit -> mikro usług (CQRS/ Event Sourcing/Saga); konteneryzacja usług -> docker, docker-compose, docker swarm, kubernetes, wdrażaniu i utrzymaniu aplikacji rozproszonej -> monitorowanie (klastry/chmury obliczeniowe OpenStack / AWS) Monitorowanie -> Sentry / Jaeger / Prometheus + Grafana / Load balancery / Systemy kolejkowe; Narzędzia do testowania obciążenia Locust.io / Jmeter Rozproszone systemy plikowe HDFS (Hive) / IPFS Rozproszone bazy danych (Hbase / Neo4j, ArangoDB) Blockchain -> Bitcoin / Ethereum / Stellar / GRP (grafowa) Rozproszone środowisko obliczeniowe (Apache Spark/ YARN -> JupyterLab -> PySparku -> .net context submit)		
Wymagania wstępne i dodatkowe	1. Student musi posiadać wiedzę i umiejętności programistyczne w technologiach .net lub java. 2. Student musi rozumieć metody komunikacji stosowane w Internecie 3. Student musi posiadać wiedzę i umiejętności w zakresie wdrażania aplikacji na serwer docelowy w wersji skonteneryzowanej.		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	wykład	50.0%	50.0%
	projekt	50.0%	50.0%

Zalecana lista lektur	Podstawowa lista lektur	1. Cloud Native DevOps with Kubernetes, John Arundel, Justin Domingus 2. Kubernetes Patterns: Reusable Elements for Designing Cloud-Native Applications, Bilgin Ibryam, Roland Huß 3. KUBERNETES: A Simple Guide to Master Kubernetes for Beginners and Advanced Users (2020 Edition), Brian Docker 4. Hands-On Docker for Microservices with Python: Design, deploy, and operate a complex system with multiple microservices using Docker and Kubernetes, Jaime Buelta 5. gRPC: Up and Running: Building Cloud Native Applications with Go and Java for Docker and Kubernetes, Kasun Indrasiri, Danesh Kuruppu 6. The Kubernetes Book, Nigel Poulton 7. Hands-On Microservices with C# 8 and .NET Core 3: Refactor your monolith architecture into microservices using Azure, 3rd Edition, Gaurav Arora 8. Ed Price 8. Pro ASP.NET Core 3: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages, Adam Freeman 9. Practical Microservices Architectural Patterns - Event-Based Java Microservices with Spring Boot and Spring Cloud, Binildas Christudas 10. Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith, Sam Newman 11. Practical Microservices: Build Event-Driven Architectures with Event Sourcing and CQRS, Ethan Garofolo 12. Architecting Modern Data Platforms, Jan Kunigk, Ian Buss, Paul Wilkinson & Lars George 13. Advanced Analytics with Spark, Sandy Ryza, Uri Laserson, Sean Owen & Josh Wills 14. Big Data Analytics with Hadoop 3, Sridhar Alla,
	Uzupełniająca lista lektur	1. Modern Big Data Processing with Hadoop, V. Naresh Kumar Prashant Shindgikar
	Adresy eZasobów	Adresy na platformie eNauczanie: High Availability Distributed Systems - Moodle ID: 45824 https://enauczanie.pg.edu.pl/moodle/course/view.php?id=45824
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	Skalowalność aplikacji, deployment; Architektury aplikacji rozproszonych (monolit -> mikro usług (CQRS/ Event Sourcing/Saga); konteneryzacja usług -> docker, docker-compose, docker swarm, kubernetes, wdrażaniu i utrzymaniu aplikacji rozproszonej -> monitorowanie (klastry/chmury obliczeniowe OpenStack / AWS) Monitorowanie -> Sentry / Jaeger / Prometheus + Grafana / Load balancery / Systemy kolejkowe; Narzędzia do testowania obciążenia Locust.io / Jmeter Rozproszone systemy plikowe HDFS (Hive) / IPFS Rozproszone bazy danych (Hbase / Neo4j, ArangoDB) Blockchain -> Bitcoin / Ethereum / Stellar / GRP (grafowa) Rozproszone środowisko obliczeniowe (Apache Spark/ YARN -> JupyterLab -> PySparku -> .net context submit)	
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy	

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.