COURSE DESCRIPTION CARD - SYLLABUS

Course name
Distributed systems design

Course

Field of study
Computing

Year/Semester
1/2

Area of study (specialization)
Distributed systems

Profile of study
general academic

Level of study
Second-cycle studies

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
20

Laboratory classes

Other (e.g. online)

Tutorials

Projects/seminars
30

Number of credit points

4

Lecturers

Responsible for the course/lecturer:
dr inż. Cezary Sobaniec

Responsible for the course/lecturer:

Prerequisites

Student starting this module should have a basic knowledge in the field of: operating systems, computer networks, distributed computing, and database systems. Student should be able to: obtain necessary information from available publications with ability to interpret and evaluate information to form an opinion, plan and conduct research (including measurements and computer simulations), interpret the outcome and verify initial hypothesis regarding complex engineering problems and simple research problems, combine knowledge from several other computing science fields (and also other academic disciplines if necessary).

Course objective

The objective for this course is to give students knowledge on design and implementation of complex distributed systems in modern public computing clouds.

Course-related learning outcomes

Knowledge

1. Has advanced and in-depth knowledge of widely understood distributed cloud computing systems, theoretical foundations of their functioning as well as services, tools and programming environments

used in the implementation of distributed applications.
2. Has structured and theoretically founded general knowledge related to distributed systems and cloud computing.
3. Has advanced detailed knowledge of the design and implementation of distributed applications in cloud computing systems.
4. Has advanced and detailed knowledge of the processes taking place in the life cycle of distributed information systems.

Skills

1. Is able to plan and carry out experiments, including measurements and computer simulations, interpret the obtained results and draw conclusions as well as formulate and verify hypotheses related to complex engineering problems and simple research problems.
2. Is able to use analytical and experimental methods to formulate and solve engineering tasks and simple research problems.
3. Is able to assess the usefulness and the possibility of using new achievements (methods and tools) and new IT products.
4. Is able to assess the usefulness of methods and tools for solving an engineering task, consisting in the construction or evaluation of an IT system or its components, including the limitations of these methods and tools.
5. Is able – using among others conceptually new methods – to solve complex IT tasks, including atypical tasks and tasks containing a research component.
6. Is able – in accordance with a given specification, taking into account non-technical aspects – to design a complex device, IT system or process and implement this project – at least in part – using appropriate methods, techniques and tools, including adapting to this purpose existing tools or developing new ones.

Social competences

1. Understands that in the field of IT the knowledge and skills quickly become obsolete.
2. Understands the importance of using the latest knowledge in the field of computer science in solving research and practical problems.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

The subject is settled on the basis of the project carried out throughout the semester. The task includes designing an application operating in cloud computing along with its prototype implementation. The assessment is individual and includes verification of knowledge of the technologies used.

Programme content

The lecture program includes the following topics:
1. Introduction to cloud computing architecture.
2. Comparison of monolithic and microservice architectures.
3. Microservice architectures in cloud computing.
4. Providing high availability of applications based on microservice architecture.

5. Designing mechanisms of data recovery in a cloud computing system.
6. Designing geographically distributed applications in public clouds.
7. Ensuring observability in distributed applications.
8. Serverless architectures in the public cloud.
9. Mechanisms of continuous integration and deployment in cloud computing.

The project is aimed at developing an application in a public cloud on a selected platform. The application should:
1. Ensure repeatability of application implementation using immutable infrastructure.
2. Provide application playback mechanisms at the following levels: warm-standby, pilot light and DR.
3. Provide security and audit of changes in infrastructure.
4. Use fan-out and fan-in patterns to handle events in infrastructure and applications.
5. Take into account monitoring mechanisms.
6. Be compliant with Amazon's Well Architected Framework.
7. Ensure the security of deployment through the mechanisms of continuous integration.

Teaching methods

1. Lecture: multimedia presentation, illustrated with examples given on the board.
2. Project: problem discussions, consultations of subsequent stages of the project.

Bibliography

Basic

1. Site reliability Engineering, Edited by Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy, O'Reilly, 2016.
2. Site reliability workbook, Edited by Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara and Stephen Thorne. O'Reilly, 2018.
3. Building Microservices, Sam Newman, 2015, O'Reilly.
4. Serverless Architectures on AWS, Second Edition , Peter Sbarski, Ajay Nair, 2018.

Additional

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,0
Classes requiring direct contact with the teacher	50	2,0
Student's own work (literature studies, preparation for laboratory classes, preparation of the project) [1]	50	2,0

[1] delete or add other activities as appropriate