COURSE DESCRIPTION CARD - SYLLABUS

Course name
Internet Technologies for Distributed Processing [S2Inf1-SRC>TI]

Course

Field of study
Computing

Year/Semester
1/1

Area of study (specialization)
Distributed and cloud systems

Profile of study
general academic

Level of study
second-cycle

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
15

Laboratory classes
30

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

3,00

Coordinators

dr inż. Cezary Sobaniec
cezary.sobaniec@put.poznan.pl

Lecturers

Prerequisites

Student starting this module should have a basic knowledge in the field of: operating systems, computer networks, distributed computing, security of information systems, and database systems. The student should be able to obtain information from indicated sources, and be able to cooperate in groups.

Course objective

1. Transfer of knowledge in the field of construction of modern and scalable distributed systems modeled as service-oriented architecture (SOA) with the use of network services. 2. Design and construction of network services in the REST model. 3. Presentation of selected current web technologies useful for building distributed applications. 4. Presentation of the asynchronous model of web services implementation.

Course-related learning outcomes

Knowledge:
1. has general knowledge related to the design of distributed systems following priciples of service oriented architecture (soa).
2. has detailed knowledge of design and implementation of network services using the rest model.

3. has knowledge of currently available web technologies that can be used in the construction of distributed systems.
4. has knowledge of the asynchronous model of network services implementation.

Skills:
1. is able to design and implement network services following the rest model.
2. is able to properly select and use available web technologies to build distributed service systems.
3. is able to construct service servers according to the asynchronous processing model.

Social competences:
1. understands that in the field of it the knowledge and skills quickly become obsolete.
2. is able to cooperate in a group.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:
The lecture content is assessed through a written test consisting of 4 problem-related questions. For each question one can get 12 points, a positive assessment requires at least 24 points.
The laboratory is assessed on the basis of 2 projects implemented during the semester.

Programme content

1. HTML5 technologies.
2. New web programming interfaces.
3. Service oriented architecture (SOA).
4. (Big) Web Services (WS-*).
5. REST web services model.
6. Resource oriented architecture (ROA).
7. Asynchronous programming model for web services.

Course topics

The lecture program includes the following topics:

1. HTML5 technologies: evolution of markup languages, XML standard, XML Information Set, presentation of XML documents, XSL standard, XHTML, the genesis of HTML5.
2. New web technologies: Web Worker, Web Storage, Service Worker, WebSocket, HTTP/2.
3. SOA architecture: service definition, SOA motivations, architecture definition, SOA assumptions, ESB bus, service description language.
4. Web Services: Motivation, WS- * standards review, SOAP protocol, SOAP message format, WSDL service description standard, WS-I profiles.
5. REST: Web Services vs Web architecture, the problem of addressing network services, HTTP protocol, REST architectural style, REST goals, resource definition, resource representations, HTTP protocol methods and error codes, REST service example, HTTP protocol limitations, REST compliance tests, REST vs AJAX, security of REST, WebDAV services, implementation of REST services.
6. Resource-oriented architecture (ROA): network service models, the importance of URI addresses, hypermedia in REST, REST modeling, the problem of choosing a resource representation, the state of interaction in REST services, resource granularity, special resources, resource collections, connections between resources, microformats , caching servers, update validators, the problem of operation idempotency.
7. Asynchronous programming model for web services: generators and coroutines in Python.

The program of laboratory classes includes the following topics:

1. HTML5 technologies: detection of HTML5, WebWorker and SharedWorker mechanisms, Web Storage, Canvas object, Service Worker, other programming interfaces, CSS3.
2. Extensible Stylesheet Language: XML document transformations, presentation formatting using Formatting Objects.
3. Asynchronous communication using WebSocket (the protocol, the client, the server).

4. REST web services: REST service modeling, resource hierarchy, resource representation, HTTP protocol methods and their semantics, the problem of reliable processing, links between resources, frameworks supporting REST services implementation.
5. Implementation of network services using the asynchronous model, Tornado framework.
6. Implementation of 2 projects: asynchronous communication with the server using Web Socket, a network service using the REST model.

Teaching methods

1. Lecture: multimedia presentation, discussion.
2. Laboratory: practical exercises consisting in creating and developing simple examples testing various Internet technologies; consultations of final projects.

Bibliography

Basic
1. Leonard Richardson, Sam Ruby, RESTful Web Services, O"Reilly Media, 2008.

Additional
1. Web standards of Internet technologies. World Wide Web Consortium, https://www.w3.org.

Breakdown of average student's workload

	Hours	ECTS
Total workload	75	3,00
Classes requiring direct contact with the teacher	45	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	30	1,00