STUDY MODULE DESCRIPTION FORM

Name of the module/subject **Selected internet technologies**	Code **1010331561010337132**

Field of study **Information Engineering**	Profile of study (general academic, practical) **(brak)**	Year /Semester **3 / 6**
Elective path/specialty **Information Technologies**	Subject offered in: **Polish**	Course (compulsory, elective) **obligatory**

Cycle of study: **First-cycle studies**	Form of study (full-time,part-time) **full-time**

No. of hours Lecture: **30** Classes: **-** Laboratory: **30** Project/seminars: **-**	No. of credits **5**

Status of the course in the study program (Basic, major, other) **(brak)**	(university-wide, from another field) **(brak)**

Education areas and fields of science and art	ECTS distribution (number and %**)**

Responsible for subject / lecturer:

dr inż. Andrzej Szwabe
email: Andrzej.Szwabe@put.poznan.pl
tel. 61 665 3958
Faculty of Electrical Engineering
ul. Piotrowo 3A 60-965 Poznań

Prerequisites in terms of knowledge, skills and social competencies:

1	**Knowledge**	knows and understands the methodology of software development and basic design patterns, as well as the basic processes of the software development life cycle knows and understands the structure, operation and basic processes occurring in the life cycle of computer networks knows and understands the advanced knowledge of databases and data warehouses and basic processes taking place in the life cycle of databases and data warehouses [K1_W12 (P6S_WG), K1_W07 (P6S_WG), K1_W08 (P6S_WG)]
2	**Skills**	can use programming environments and platforms to write, execute and test simple coded programs in the imperative, object and declarative programming languages, use analytical, simulation and experimental methods for this purpose [K1_U10 (P6S_UW)] can design - according to the given specification - and create a simple database or data warehouse, use simple queries by means of appropriately selected methods, techniques and tools [K1_U12(P6S_UW)]
3	**Social competencies**	is ready to responsibly carry out professional roles, including adherence to professional ethics and requiring it from others [K1_K04 (P6S_KR)]

Assumptions and objectives of the course:

The main objective of the course is to provide the student with knowledge and skills in the area of technologies that enable the construction of advanced systems for the provision of services on the Internet - in particular systems designed in accordance with the microservices paradigm and the containerization (operating system level virtualization) paradigm. The use of machine learning functions in the Internet service system is regarded as the most important application scenario considered within the scope of the course. Therefore the scope of the course complements the scopes of courses devoted to network technologies, artificial intelligence systems and recommendation systems.

Study outcomes and reference to the educational results for a field of study

Knowledge:

1. knows and understands the advanced knowledge in the field of Internet technologies - [K1_W11 (P6S_WG)]

2. knows and understands the basic processes occurring in the life cycle of devices, facilities and information systems - [K1_W19 (P6S_WG)]

Skills:

1. can design, in accordance with the given specification, and perform basic tasks related to websites and internet services, using appropriately selected methods, techniques and tools - [K1_U15 (P6S_UW)]

2. can estimate the time needed for the implementation of the task ordered; develop and implement a work schedule that ensures deadlines; plan and organize individual and team work - [K1_U02 (P6S_UO)]

Social competencies:

1. is ready to responsibly carry out professional roles, including adherence to professional ethics and requiring it from others - [K1_K04 (P6S_KR)]

Assessment methods of study outcomes

Lecture: writing exam; minimal score 50,1%.

Laboratory: each laboratory exercise scored; minimal score 50,1%.

Course description

Lectures:

An important goal of the lecture is to provide the student with knowledge on microservices development technologies ? in particular technologies for designing, implementing (i.e., preparing the code, including the use of the programming technique of the so-called decorators) in Python (the choice of language is dictated by its dominant position among programming languages that enable commercial implementation of Web services, prediction services based on Machine Learning technologies) and testing microservices equipped with Web APIs or REST-type Web interfaces. A method of effective testing of an initial implementation ? working outside the container environment and without the use of an HTTP server providing multithreading ? followed by uncomplicated transformation of the implementation into a containerized form using Docker technology that ensures multi-threaded query handling (implemented using the so-called production-ready HTTP server) is presented. While presenting the area of containerization techniques, a special attention is paid to the functions of sharing disk space and inter-container network communication. The student is also provided with knowledge on database servers, indexing servers and their client applications - with particular emphasis on NoSQL technology and their applications to cache storage (including Redis and Elasticsearch) and knowledge on technologies for predictive models building, prediction generation services (based on the use of predefined prediction models, referred to as model serving) and data preparation services.

Laboratory:

The topics of laboratory exercises include: development of the code (in Python) of a Web service (in the form of a Web API and / or a REST service) using the programming technique of the so-called decorators, two-stage testing of micro-service code (providing Web API or REST Web service endpoint) in a way that allows testing of initial implementation outside container environment and without the use of a multithreaded HTTP server followed by a simple conversion of the implementation to a containerized form (using Docker technology) that ensures multi-threaded query handling (implemented with the use of a production-ready HTTP server), applications of Flask, Requests and CherryPy modules, containerization techniques (in particular the functions of disk space sharing and inter-container network communication), application of database NoSQL technologies (including Redis) and indexing servers (including Elasticsearch) for the implementation of micro-service architectures - with particular emphasis on the mechanism of the cache memory), microservices for the construction of prediction models, prediction generation using predefined prediction models (referred to as model serving), microservices for data preparation, TensorFlow technology applications (in particular, TensorFlow Estimator API and TensorFlow ModelServer).

Teaching methods - lectures:

- Presentation of the theory with frequent references to relevant practical examples of software implementations,
- Lecture with multimedia presentation and presentations of Python programming language source code examples with their execution and rapid development/modification,
- Students being asked questions during the lectures in order to provoke discussions.

Teaching methods - laboratory:

- Individual work on the system (based on a script of the laboratory exercise),
- Work on configuration and experimentation in a environment composed of multiple containers,
- Work on open source tools and software components (including those developed in Poznan University of Technology research projects) made available to students to support their work,
- At the end of each class an evaluation of the results made by the lecturer,
- At the end of the semester preparation of the report on the implementation of all the laboratory tasks.

Basic bibliography:

1. Sam Newman? Building Microservices. Designing Fine-Grained Systems, O'Reilly Media, 2015, https://books.google.pl/books/about/Building_Microservices.html

2. Mark Richards, Microservices vs. Service-Oriented Architecture, O'Reilly Media, 2016

3. Leonard Richardson and Sam Ruby. 2007. Restful Web Services (First ed.). O'Reilly, http://restfulwebapis.org/RESTful_Web_Services.pdf

4. Heng-Tze Cheng et al., Wide & Deep Learning for Recommender Systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016). ACM, New York, NY, USA, 2016, 7-10. DOI: http://dx.doi.org/10.1145/2988450.2988454, https://arxiv.org/pdf/1606.07792.pdf

5. D. Baylor et al, TFX: A TensorFlow-Based Production-Scale Machine Learning Platform, In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17), ACM, New York, NY, USA, 1387-1395, 2017, DOI: https://doi.org/10.1145/3097983.3098021, http://stevenwhang.com/tfx_paper.pdf

Additional bibliography:

1. M. Abadi et al., TensorFlow: a system for large-scale machine learning. In Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation (OSDI'16). USENIX Association, Berkeley, CA, USA, 265-283, https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

2. G. Zaccone, Getting Started with TensorFlow, Packt Publishing, July 2016, ISBN: 9781786468574, https://www.packtpub.com/free-ebook/getting-started-tensorflow

3. Sieci komputerowe i intersieci, D.E. Comer, Helion, Warszawa, 2012.

4. Sieci komputerowe, A. Tanenbaum, Helion, Gliwice, 2012.

Result of average student's workload

Activity	Time (working hours)
1. lecture	30
2. laboratory	30
3. exam and consulting hours with the teacher	10
4. preparation for exam	10
5. preparation for laboratory	45

Student's workload

Source of workload	hours	ECTS
Total workload	125	5
Contact hours	50	2
Practical activities	75	3