



## COURSE DESCRIPTION CARD - SYLLABUS

Course name

Advanced Internet Applications

### Course

Field of study

Computing

Area of study (specialization)

Level of study

First-cycle studies

Form of study

part-time

Year/Semester

3/5

Profile of study

general academic

Course offered in

english

Requirements

elective

### Number of hours

Lecture

16

Laboratory classes

16

Other (e.g. online)

Tutorials

Projects/seminars

### Number of credit points

4

### Lecturers

Responsible for the course/lecturer:

dr. inż. Maciej Piernik

email: maciej.piernik@cs.put.poznan.pl

tel: (+48 61) 665-30-57

wydział: Wydział Informatyki i Telekomunikacji

adres: ul. Piotrowo 2, 60-965 Poznań

Responsible for the course/lecturer:

dr inż. Paweł Boiński

email: Pawel.Boinski@cs.put.poznan.pl

tel: 61 6652965

wydział: Wydział Informatyki i Telekomunikacji

adres: ul. Piotrowo 3 60-965 Poznań

### Prerequisites

Students taking this course should possess basic knowledge about network protocols, database systems and object oriented programming. They should also have basic application programming skills using integrated development environments. They should also understand the necessity to broaden their competences and be ready to cooperate with others as a part of a team.

### Course objective

1. Gaining knowledge about www document and application development necessary to distinguish between basic internet application architectures and methods for implementing their modules.
2. Enhancing knowledge about network architectures, protocols, and distributed systems security.
3. Gaining skills in web application development using advanced user interface development technologies, such as CSS, JavaScript, presentation logic development, such as Java servlets, Java Server



Pages, PHP, ASP.NET, Single Page Application Frameworks, business logic development, e.g., JavaBeans, JSF tag library.

4. Gaining social competences concerning working as a part of a team, including organising team work, in particular leadership and communication in the process of group problem solving.

### Course-related learning outcomes

#### Knowledge

1. has a structured knowledge about www application architectures - [K1st\_W4]
2. knows basic methods, techniques, and tools used in solving simple computer science tasks concerning designing, implementing, and deploying web applications - [K1st\_W7]
3. has a systematized knowledge about network protocols and distributed systems security - [K1st\_W4]

#### Skills

1. is capable of designing and developing an internet application using appropriate tools, methods and techniques - [K1st\_U10]
2. can design algorithms and implement them using at least one of popular tools available - [K1st\_U11]
3. is capable of designing web applications based on database systems with interactive user interfaces - [K1st\_U12]
4. can design an appropriate user interface for various classes of web systems - [K1st\_U14]
5. can choose a web technology appropriate for a given domain of application - [K1st\_U18]

#### Social competences

1. can work as a part of a team and plan the work for each team member - [K1st\_K1]
2. realizes the importance of engineering knowledge in solving problems and knows examples and causes of failed systems - [K1st\_K2]
3. has good entrepreneurial skills and thinks about results commercialization - [K1st\_K3]

### Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Continuous grading:

- a) lectures: - based on answers to questions posed during lectures,
- b) laboratories: - based on monitoring the progress of completing the exercises.

Final grading:

- grading and defence of a project created during the semester,
- assessment of knowledge and skills in a test.

### Programme content

1. Different approaches to the problem of presentation logic on various platforms, such as ASP.NET and Java EE.
2. How to design a reusable business logic capable of serving multiple types of applications and different interfaces on the most common platforms.



3. How to design a data driven application. How to organize a data access layer such that it is reusable, scalable, efficient, and secure. Examples on various platforms.
4. Web application infrastructure. The most popular web development architectures.
5. Http servers
6. Advanced user interface: CSS preprocessors, responsive design, front-end frameworks.
7. Single Page Application development using popular JavaScript frameworks, advanced JavaScript concepts, asynchronous processing, designing and developing RESTful Web Services.
8. Authentication and authorization in web applications. The most important attacks and how to defend against them.
9. Testing web application functionality and efficiency.

Part of the program is planned as students' own work.

### Teaching methods

1. Lecture: presentation, examples on a blackboard, live demonstrations, live exercises.
2. Laboratory: completing exercises, working in teams, presentations, live demonstrations.

### Bibliography

#### Basic

1. J. Duckett, Web Design with HTML, CSS, JavaScript and jQuery, Wiley, 2014.
2. B. Sholtz, A. Tijms, The Definitive Guide to JSF in Java EE 8: Building Web Applications with JavaServer Faces, Apress, 2018.
3. K. Hadlock, Ajax for Web Application Developers, Sams Publishing, 2006.
4. J. Liberty, D. Hurwitz, B. MacDonald, Learning ASP.NET 2.0 with AJAX: A Practical Hands-on Guide, O'Reilly, 2007.

#### Additional

1. E. Jendrock, I. Evans, D. Gollapudi, K. Haase, C. Srivathsa, The Java EE 6 Tutorial, Oracle, 2010.



### Breakdown of average student's workload

	Hours	ECTS
Total workload	98	4,0
Classes requiring direct contact with the teacher	34	1,5
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests, homework assignments, project preparation) <sup>1</sup>	64	2,5

<sup>1</sup> delete or add other activities as appropriate