Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Emblem of the department (in the presence)

Department of computer technology

Distributed information systems

curriculum (Syllabus)

Details of the Course

Level of higher education	*First (undergraduate)*
Branch of knowledge	*12 Information technologies*
Specialty	*123 Computer engineering*
Educational program	*Computer Engineering*
Discipline status	*Selective*
Form of education	*Full-time*
Year of training, semester	*2nd year, spring semester*
ECTS workload	*36 lectures, 18 practice, 66 self-study*
Semester control/ control measures	*Final Test*
Timetable	
Language of teaching	*English*
Information about the head of the course / teachers	*PhD, Assoc. prof. Andriy BOLDAK*
Access to the course	Presentations https://edu-dis-presents.github.io/p1/1 https://edu-dis-presents.github.io/p2/1 https://edu-dis-presents.github.io/p3/1 https://edu-dis-presents.github.io/p4/1 https://edu-dis-presents.github.io/p5/1 https://edu-dis-presents.github.io/p6/1 Repository template for performing laboratory work https://github.com/edu-dis-presents/edu-dis-labs-example Course of lectures https://boldak.github.io/dis-edu/

Program of Course

1. Description of the educational discipline, its purpose, subject of study and learning outcomes

The goal of the educational discipline is the formation of students' competencies and the training of a specialist capable of solving complex tasks and practical problems of distributed information processing systems and carrying out professional activities in the design, implementation, and selection of technologies and administration of distributed information processing systems.

As a result of the study of the academic discipline, students should have the following list of special (professional, subject) competencies:

- ability to think abstractly, analysis and synthesis;
- ability to apply knowledge in practical situations;
- ability to understand the subject area and professional activity;

- the ability to choose, design, deploy, integrate, manage, administer and support information systems, technologies and information communications, services and infrastructure of the organization;
- the ability to use technologies of distributed computing, virtualization of server systems, design corporate computing systems, use cluster and heterogeneous distributed computing systems to solve applied problems and conduct scientific research, solve problems of scalability, design and operation of distributed information systems.

The learning outcome is the ability to apply distributed computing technologies in the development of information management systems and decision support technologies.

2. Pre-requisites and post-requisites of the discipline (place in the structural and logical scheme of training according to the relevant educational program)

The discipline is a component of professional and technical training and is directly related to other regulatory disciplines of natural science, basic and professional training
"Discrete mathematics", "Object-oriented programming", "Organization of computer networks", "Software development technologies".
Basic knowledge for studying the discipline is the basics of object-oriented programming, design patterns, software development technologies.

3. Content of the academic discipline

Chapter 1. Concept of distributed information system.
Topic 1.1. Basic concepts of distributed information systems, classification of distributed information systems.
Topic 1.2. Communication of resources in distributed information systems.
Topic 1.3. Examples of distributed information systems.

Chapter 2. Architecture of distributed information systems.
Topic 2.1. Architectural principles of designing distributed information systems.
Topic 2.2. Client-server architecture.
Topic 2.3. Service-oriented architecture.
Topic 2.4. Event-oriented architecture.
Topic 2.5. Microservice architecture.

Chapter 3. Organization of the interaction of the components of the distributed information system.
Topic 3.1. Web server development using express.js.
Topic 3.2. RESTfull services.
Topic 3.3. GraphQL .
Topic 3.4. Push notification using Server-Sent Events .
Topic 3.5. Scripted Query services.

Chapter 4. Application software interface of services
Topic 4.1. OpenAPI Specification , Swagger .
Topic 4.2. Web service development using SwaggerUI and express.js.
Topic 4.3. MSAPI EDA specification.

Chapter 5. Implementation of interaction of components in EDA
Topic 5.1. Implementation of messaging using AMQP.
Topic 5.2. Implementation of simple interaction using message queues.
Topic 5.2. Implementation of a task queue.
Topic 5.3. Implementation of the PUB/SUB interaction pattern.
Topic 5.4. An implementation of the RPC pattern.

Chapter 6. Implementation of service interaction in MSA.

Topic 6.1. Implementation of gradual service degradation.
Topic 6.2. Implementation of saga orchestration.
Topic 6.3. Realization of the choreography of sagas.
Topic 6.4. Implementation of the service switch.

4. Educational materials and resources

1. Antonenko V. M. Modern information systems and technologies: knowledge management: education . manual / V. M. Antonenko, S. D. Mamchenko , Yu. V. Rogushina . - Irpin: National . DPS University of Ukraine, 2016. – 212 p.

2. Bidyuk P.I., Korshevnyuk L.O. Designing computer information systems for decision support: Tutorial. - Kyiv: NNK "IPSA" NTUU "KPI", 2010. - 340 p.

3. Shakhovska N.B. Design of information systems: teaching . help _ / N. B. Shakhovska, V. V. Lytvyn; under the editorship V. V. Pasichnyk; Ministry of Education and Science of Ukraine. - Lviv: Magnolia 2006, 2011. – 380 p.

4. Design of information systems: manual / V. S. Ponomarenko, editor. - K.: Academy, 2012. – 488 p.

5. Voronin A. M. Information systems of decision-making: training manual. / Voronin A. M., Ziatdinov Yu. K., Klimova A. S. - K.: NAU-druk, 2009. - 136 p.

6. Morse N.V. Information systems. Study _ manual _ /for sciences ed. N. V. Morse; Morse N.V., Pikh O.Z. - Ivano-Frankivsk, " LileyaNV ", - 2015. - 384 p.

7. Pavlysh V. A., Glinenko L. K. Fundamentals of information technologies and systems: Training manual. / Pavlysh V. A., Glinenko L. K. - Lviv: Publishing House of Lviv Polytechnic, 2013. - 500 p.

8. Guzhva V.M. Information systems and technologies at enterprises. K.: KNEU, 2001. – 158 p.

Additional

9. "SOA Source Book - What Is SOA?". collaboration.opengroup.org. Retrieved March 30, 2021. https://collaboration.opengroup.org/projects/soa-book/pages.php?action=show&ggid=1314
10. What is an Event-Driven Architecture ? Decoupled systems that run in response that events . https://aws.amazon.com/event-driven-architecture/
11. The Six Advantages of Microservices White Paper . A whitepaper for software architects planning a microservices deployment . https://hazelcast.com/lp/six-advantages-microservices/?utm_campaign=Microservices&utm_source=google&utm_medium=cpc&utm_term=microservices%20solution%20architecture&utm_content=adgroupid:30167106143%20creative:326783455375%20matchtype:p%20network:g%20device:c%20position:%20placement:&adgroupid=30167106143&creativeid=326783455375&campaignid=633821002&gclid=CjwKCAiA8OmdBhAgEiwAShr4007m9NYuIVu4xBAiwx4AStnODKLEx5jhw7ePuoaXPYHt3zDZP-Np5BoCoXAQAvD_BwE
12. OpenAPI : Beginner that Guru . https://www.udemy.com/course/openapi-beginner-to-guru/?utm_source=adwords&utm_medium=udemyads&utm_campaign=LongTail_la.EN_cc.ROW&utm_content=deal4584&utm_term=_._ag_77879424134_._ad_535397245863_._kw__._de_c_._dm__._pl__._ti_dsa-1007766171312_._li_21580_._pd__._&matchtype=&gclid=CjwKCAiA8OmdBhAgEiwAShr40-bv4kOeuh1s5AwJyIlOru5_4l_at0BcQjNOGLo-ZkbjDmvF0IdJ5xoCHlUQAvD_BwE
13. OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0 . http://docs.oasis-open.org/amqp/core/v1.0/amqp-core-transport-v1.0.html#doc-flow-control

14. MOLFAR. A software module for processing and validating microservices API specifications and workflows - @ molfar / msapi-schemas . https://github.com/wdc-molfar/msapi-schemas
15. MOLFAR. A software module for implementing the interaction of microservices using message transfer - @ molfar / amqp-client . https://github.com/wdc-molfar/amqp-client

Educational content

5. Methods of mastering an educational discipline (educational component)

Names of sections, topics	Distribution of study time				
	In total	including			
		Lecture s	Practical (seminar) classes	Laboratory work (computer workshop)	Self-stufy
1	2	3	4	5	6
Chapter 1. Concept of distributed information system.					
Topic 1.1. Basic concepts of distributed information systems, classification of distributed information systems.	3	1			2
Topic 1.2. Communication of resources in distributed information systems.	5	2		1	2
Topic 1.3. Examples of distributed information systems.	3	1			2
Total according to section 1	**11**	**4**	**0**	**1**	**6**
Chapter 2. Architecture of distributed information systems.					
Topic 2.1. Architectural principles of designing distributed information systems.	4	2			2
Topic 2.2. Client-server architecture.	3	1			2
Topic 2.3. Service-oriented architecture.	6	1		1	4
Topic 2.4. Event-oriented architecture.	8	2		2	4
Topic 2.5. Microservice architecture.	8	2		2	4
Total according to section 2	**29**	**8**	**0**	**5**	**16**
Chapter 3. Organization of the interaction of the components of the distributed information system.					
Topic 3.1. Web server development using express.js.	7	2		1	4
Topic 3.2. RESTfull services.	7	2		1	4
Topic 3.3. GraphQL .	5	1			4

Topic 3.4. Push notification using Server- Sent Events .	5	2		1	2
Topic 3.5. Scripted Query services.	5	2		1	2
Total according to section 3	**29**	**9**	**0**	**4**	**16**
Chapter 4. Application software interface of services					
Topic 4.1. OpenAPI Specification , Swagger .	4	1			3
Topic 4.2. Web service development using SwaggerUI and express.js.	5	2		1	2
Topic 4.3. MSAPI EDA specification.	4	2			2
Total according to chapter 4	**thirteen**	**5**	**0**	**1**	**7**
Chapter 5. Implementation of interaction of components in EDA					
Topic 5.1. Implementation of messaging using AMQP.	4	2			2
Topic 5.2. Implementation of simple interaction using message queues.	4	1		1	2
Topic 5.2. Implementation of a task queue.	4	1		1	2
Topic 5.3. Implementation of the PUB/SUB interaction pattern.	4	1		1	2
Topic 5.4. An implementation of the RPC pattern.	4	1		1	2
Total according to chapter 5	**20**	**6**	**0**	**4**	**10**
Chapter 6. Implementation of service interaction in MSA.					
Topic 6.1. Implementation of gradual service degradation.	3	1			2
Topic 6.2. Implementation of saga orchestration.	4	1		1	2
Topic 6.3. Realization of the choreography of sagas.	4	1		1	2
Topic 6.4. Implementation of the service switch.	4	1		1	2
Total according to chapter 6	**15**	**4**	**0**	**3**	**8**
Preparation for the test	2				2
Test	1				1
Total per semester	**120**	**36**	**0**	**18**	**66**

Lecture classes

lecture no	The name of the topic of the lecture and a list of the main questions (a list of didactic tools, references to the literature and tasks on the SRS)
1	The main ones concept distributed information systems, classification distributed information systems. Concept systems, integrability and system-forming elements factors, external environment and system, information system, distributed information system, integrity information systems, centralized or decentralized approach to organization functioning information systems, principles communication resources in distributed information systems.
2	Examples distributed information systems . Protocols exchange information , applied program interface , decomposition information systems , information infrastructure of the world center data " Geoinformatics and constant development " NTUU "KPI", web application PRO ET CONTRA, web application DATA FRESH on COVID-19, web application Crimea Water Resources, system collection and analysis data from the Internet media MOLFAR sources .
3	Architectural principles designing distributed information systems. Concept of architecture, templates of interaction, transparent access to resources, transparent resource allocation, resource replication, parallel access to resources, refusal of transparency.
4	Client-server and service-oriented architecture. Client-server architecture, SOA, service components, SaaS, PaaS, IaaS, interface, transport, registries, principles of reuse, provision of a formal usage contract, loose coupling, internal logic abstraction, compatibility, autonomy, detection support.
5	Event-oriented architecture. EDA, events, messages, emitters, sinks, event generators, event channels, event handlers, post-events, simple and complex events processing, AMQP, exchangers and queues, suppliers and users, work exchanger modes.
6	Microservice architecture . Architectural MSA style, microservices, repeated use, interaction services, data duplication, decomposition of systems, Sagas, choreography and orchestration of Sagas.
7	Development of a web server using express.js. Project preparation, file structure, dependency installation, server configuration, middleware configuration, access to static files, request processing, parameters request processing, Tabbed Postman - REST Client.
8	RESTful services. REST protocol, project preparation, controller development, router development, capacity service testing, automation service testing, GraphQL request language.
9	Push-notifications using Server-Sent Events. SSE, requirements for the push-server, EventSource, push-service development, service improvement.

10	Scripted Query services. Rapid development of programs. Low-Code Development Platform, the principle of data processing script delegation, service development with SQ support, SQ request handler with support for asynchronous operations, SQ request handler with SSE support.
11	Specification OpenAPI, Swagger. Specification OpenAPI, Swagger Editor tool, high level structure description, final points description, usage of multiple descriptions, query and body parameters description, operations grouping, SwaggerUI, swagger-stats.
12	MSAPI EDA specification. Base example, message processing chain, standard message handlers, message validation, usage metrics, listening to a single message queue, task queue organization, message flow merge, consequent work process of message handling, partial message handling and composition of results .
13	Message exchange realization using AMQP. RabbitMQ, simple interaction implementation with use of message queues.
14	Queue task realization using AMQP. Interaction service schemes, schedulers, management of task distribution.
15	Implementation of the interaction pattern PUB/SUB using AMQP . Modes: fanuot, direct, topic, RPC template implementation .
16	Realization orchestration of sagas in MSA. Orchestrator, facade, microservices, saga orchestration tests.
17	Choreography of saga implementations in MSA . Resource self-registration template, resource status monitoring, consumers and publishers of messages in the structure of microservices, saga choreography testing, microservices switch implementation.
18	Realization of gradual degradation of the services in MSA. The principle of failure isolation, project settings, facade configuration, API gateway, delegation of requests, gradual degradation service testing.

Laboratory classes (computer practice)

The main tasks of the cycle of laboratory classes (computer practice) are students' acquisition of the necessary practical skills related to the design, implementation and use of distributed information systems with microservice architecture. The topics of laboratory work cover separate stages of designing and implementing such a system within the limits of individual tasks for the development of applied information systems for various branches of industrial activity.

No. z/p	Name of laboratory work (computer practice)	Number of aud. hours
1	Repository preparation, setting up the means of automation documentation and testing.	2
2	Designing the structure of the information system using the MSA architectural style	2
3	Development of informational facade system and API gateway	2
4	Development of Open API specifications for information systems facade	2
5	Integration of facade information system with SwaggerUI, RESTful service testing	2

6	Development of the MSAPI specifications for microservices	2
7	Development of means of microservices interaction using AMQP	2
8	Development and unit-testing microservice controllers	2
9	Integration of distributed information systems	2
	Together :	18

6. Independent work of a student/graduate student

In the process of completing individual tasks, students must consolidate knowledge acquired during lectures and independent work, independently study specific topics, deepen their knowledge for further study. Students' independent work consists of the following:

- In preparation for lecture classes on the study of previous lecture material;
- In the performance of lecture assignments during self-study;
- In preparation for laboratory works with the study of the theory of a laboratory session with an oral answer to the given questions of the section;
- In execution with the preparation of the protocol for each laboratory session on the previous topic.

.

Policy and control

7. Policy of academic discipline (educational component)

In the process of completing individual tasks, students must consolidate knowledge acquired during lectures and independent work, independently study specific topics, deepen their knowledge for further study. Students' independent work consists of the following:

- preparation for lecture classes on the study of previous lecture material;
- performance of lecture self-study tasks;
- preparation for laboratory work with the study of the theory of laboratory work with an oral answer to the given questions of the section;
- implementation and preparation of the protocol for each laboratory session on the previous topic.

8. Types of control and rating system for evaluating learning outcomes

The student's rating consists of the points he receives for the performance and defense of laboratory work

System of rating points

1. Performance and protection of laboratory work.

During the semester, students perform 10 laboratory works. Weight score -9.
The maximum number of points for each work:

- For performing and defending laboratory work on time - 9 points.
- 10 points for early completion and defense of all laboratory work

Penalty points:

- absence from laboratory work without a valid reason - 0.25 points;
- being late for laboratory work by more than 5 minutes - 0.25 points;
- performance or defense of laboratory work not on time - 3 points.

The maximum number of points for laboratory works R_1 = 90 points: (9*10 laboratory works + 10 for early defense of all works).

The minimum number of points for one laboratory work is 6 points. That is, a student who completed all laboratory work can receive 60 points.

Calculation of the size (R) of the student rating:

The sum of the weighted points of control measures during the semester is:

$R_c = R_1$, where

- R_1 - the sum of weighted points of control measures during the semester for laboratory works,

The minimum value of R_c under the condition of completion of all laboratory works is $R_c = 60$.

The size of the credit module rating scale is 100 points.

A necessary condition for a student's admission to credit is the absence of laboratory work arrears.

To receive credit from the credit module "automatically" you need to have a rating of at least 60 points.

Students who have a rating of less than 60 points, as well as those who want to increase their rating in the ECTS system, undergo an interview, based on the results of which they can receive additional points . . (The maximum value of r_d = 20). A student's additional r_d points are added to his semester rating R.

$RD = R_c + r_d$.

The grade (ESTS and traditional) is assigned according to the RD points scored. The point scored by the student (total rating of the student) is RD according to the table.

Rating value from the RD credit module	ECTS assessment	Traditional credit assessment
95-100	A	Enrolled
85-94	B	
75-85	C	
65-75	D	
60-65	E	
<60	Fx	Not counted
There are arrears for laboratory work	F	Not allowed

The working program of the academic discipline (syllabus):

Compiled by: Associate Professor of the Computer Engineering Department, Candidate of Technology . Sciences, associate professor, Andrii BOLDAK

Approved by the Department of Computing (Protocol No. 10 dated 05/25/2022)

Agreed by the Methodical Commission of the faculty [1](protocol No. 10 dated 06/09/2022)