MD-2068, CHISINAU, 9/7 STUDENTILOR STR, PHONE: 022 50-99-63, www.utm.md

APPLICATION-ORIENTED SOFTWARE ENVIRONMENTS

1. Course unit/module information

Faculty	Computers, Informatics, and Microelectronics				
Chair/department	Informatics and Systems Engineering				
Cycle of studies	Master's degree studies, cycle II				
Study program	**Data Science**				
Year of study	**Semester**	**Type of evaluation**	**Training category**	**Optionality category**	**ECTS credits**
II (full-time education);	3	E	F – Fundamental course unit	O – Obligatory course unit	5

2. Total estimated time

Total hours in the curriculum plan	Including				
	Auditory hours		Individual work		
	Course	Laboratory/Seminar	Term paper	Study of theoretical material	Application preparation
150	20	20		60	50

3. Preconditions for access to the course unit/module

According to the curriculum plan	Object-Oriented Programming, Operations Research, Advanced Programming Techniques, Systems Analysis and Modeling, Information Security Technologies, Software Testing, Interactive Programming.
According to competencies	Advanced knowledge of programming and the use of software frameworks. Familiarity with database management and API integration.

4. Conditions for conducting the educational process

Lecture	The theoretical material will be presented in the classroom using a projector and a computer. Educational materials will be made available to students on the course page of the department's pedagogical server.
Laboratory/Seminar	Students will complete reports according to the conditions outlined in the methodological guidelines. The deadline for submitting the laboratory work is two weeks after its completion. Late submissions will be penalized by a deduction of 0.25 points per day of delay.

5. Specific competencies acquired

Professional competencies	**CPM 1. Development and Design of Architecture** K1 Architecture Models, Methodologies, and System Design Tools K2 Requirements for System Architecture: Performance, Maintainability, Extensibility, Scalability, Availability, Security, and Accessibility K3 Costs, Benefits, and Risks of a System Architecture K4 Enterprise Architecture and Company Internal Standards K5 Emerging Technologies (e.g., Distributed Systems, Virtualization Models, Data Sets,

	Mobile Systems) **CPM 3. Application Development. Component Integration. Systems Engineering** K1 Appropriate Programs/Modules, DBMS, and Programming Languages. Cutting-Edge Technologies K3 The Impact of System Integration on the Organization or Existing System K4 Interface Techniques Between Modules, Systems, and Components K5 Integration Testing Techniques K6 Best Design Practices K7 Hardware Components, Tools, and Hardware Architectures K8 Functional and Technical Design K9 Fundamentals of Information Security K10 Prototyping
Transversal Competencies	**CT1**. Honorable, Responsible, and Ethical Behavior in the Spirit of the Law to Ensure the Fulfillment of Professional Tasks. **CT2**. Demonstrating the Ability to Work in a Team, Identifying Individual and Shared Roles and Responsibilities, Making Decisions, and Assigning Tasks, with the Application of Relationship Techniques and Effective Work Within the Team. **CT3**. Demonstrating Initiative and Action for Professional and Personal Development Through Continuous Training Using Documentation Sources in Romanian and International Languages.

6. Course/Module objectives

General objective	The development of the theoretical and practical competencies necessary for the design, development, and implementation of software applications using modern software environments and frameworks.
Specific objectives	As a result of studying the course, the student will know: • Understanding of Service-Oriented Architecture (SOA) applications. • Development of skills in using modern software environments and frameworks (e.g., Spring, Django). • Integration of APIs and utilization of web services. • Optimization of applications for performance and scalability. • Understanding of software testing practices and continuous delivery (CI/CD).

7. Course/Module content

Syllabus of teaching activities	Number of Hours, Full-Time Education
Course topics	
T1. Introduction to Application-Oriented Software Environments: Principles and Architectures.	4
T2. Software Development Frameworks: Overview of Technologies (Spring, Flask, Django).	4
T3. Web Services and API Integration: REST and GraphQL.	4
T4. Application Performance Optimization: Caching, Load Balancing.	4
T5. CI/CD and DevOps in Software Development.	4
Total lectures:	**20**

Syllabus of teaching activities	Number of hours, full-time education
Laboratory/seminar works topics	
LL1. Creating a simple web application using Django.	6
LL2. Implementing a REST API using Flask.	6
LL3. Integrating an external service into a software application.	4
LL4. Automating delivery with GitHub Actions and Docker.	4
Total laboratory/seminar works:	**20**

8. Using generative AI

Permission to use	The use of generative AI in assignments and projects is permitted, provided that students adhere to the following rules: • Generative AI may be used to generate ideas, text structures, or code, but all generated materials must be reviewed and adjusted by the student to ensure that they meet academic requirements. • Any use of generative AI must be declared in the appendix section of each paper, using the phrase: "During the preparation of this paper, the author used [NAME OF TOOL / SERVICE] for the purpose of [REASON]. After using this tool / service, the author reviewed and edited the content as necessary and assumes full responsibility for the content of the paper."
Restrictions to use	Students ***MUSTN'T consider generative AI as a reliable source of information***, as it does not provide clear references or documented sources. • ***Direct citation of AI-generated content*** in academic papers as if it were a primary source ***isn't permitted.*** • Activities in which the use of **generative AI is prohibited** are specified by the teacher and are usually ***intermediate and final assessments*** or that don't involve professional competence development activities.

9. Bibliographic references

Main	1. Freeman, E., & Robson, E. "Head First Design Patterns." O'Reilly Media, 2020. 2. Richardson, L. & Amundsen, M. "RESTful Web APIs." O'Reilly Media, 2022. 3. Chacon, S., & Straub, B. "Pro Git." Apress, 2021.
Supplementary e	1. Martin, R. "Clean Architecture: A Craftsman's Guide to Software Structure and Design." Prentice Hall, 2017. 2. Docker Documentation: https://docs.docker.com/ 3. Spring Framework Documentation: https://spring.io/projects/spring-framework.

10. Evaluation

Form of study	Periodic		Current	Individual work	Final exam
	Mid-term 1	Mid-term 2			
Full-time	15%	15%	15%	15%	40%
Minimum performance standards					

Attendance and participation in lectures and laboratory work Obtaining a minimum grade of "5" for each evaluation and laboratory work

11. Evaluation criteria

Activity	Evaluation components	Evaluation method, evaluation criteria	Weight in final grade for the Activity	Weight in course evaluation
Full-time education				
Mid term I	Theoretical content, topics 1-3	Test	100%	**15%**
Mid term II	Theoretical content, topics 4-5	Activities during practical work/seminar	100%	**15%**
Current evaluation	Practical activity	Attendance and participation in classes	50%	**15%**
Individual study	Classification of research by activity type	Presentation/Discussion on the topic	100%	**15%**
Final examination	Theoretical and practical content	Oral exam. Grading according to grading scale	100%	**40%**