

DENUMIRE DISCIPLINĂ: SISTEME SOFTWARE COMPLEXE. Proiect semestrial.
1. Date despre disciplină/modul

Facultatea	Calculatoare, Informatică și Microelectronică				
Departamentul	Informatică și Ingineria Sistemelor				
Ciclul de studii	Ciclul II, Studii superioare de master				
Programul de studii	Managementul aplicațiilor informaționale				
Anul de studii	Semestrul	Tip de evaluare	Categoria formativă	Categoria de opționalitate	Credite ECTS
Anul II (<i>învățământ cu frecvență</i>)	III	PA	S – unite de curs de specialitate	O - unitate de curs obligatorie	10

2. Timpul total estimat

Total ore în planul de învățământ	Din care				
	Ore auditoriale		Lucrul individual		
	Curs	Practice	Proiect de an	Studiul materialului teoretic	Pregătire aplicații
Învățământ cu frecvență	-	-	80	80	220

3. Precondiții de acces la disciplină/modul

Conform planului de învățământ	Limba de programare orientate pe obiecte, Analiza și modelarea sistemelor, Baze de date relaționale (SQL) și a celor NoSQL, Sisteme distribuite, Protocoale de comunicare, Testare software, Aplicații și servicii Cloud.
Conform competențelor	Pentru cursul Sisteme Software Complexe , studenții trebuie să aibă competențe avansate în programare, inclusiv OOP, structuri de date și optimizarea codului. Ei trebuie să înțeleagă modele de proiectare software, să interpreteze și să creeze diagrame UML, să gestioneze baze de date relaționale și NoSQL și să utilizeze API-uri RESTful. Experiența cu sisteme distribuite, controlul versiunilor (Git) și CI/CD este esențială. Studenții trebuie să fie familiarizați cu testarea software, măsurile de securitate și utilizarea containerelor și serviciilor cloud. De asemenea, sunt necesare abilități de colaborare, management al proiectelor.

4. Condiții de desfășurare a procesului educațional pentru

Curs	Pentru prezentarea materialului teoretic în sala de curs este nevoie de tablă, cretă, proiector și calculator. Nu vor fi tolerate întârzierile studenților, folosirea laptopurilor, precum și convorbirile telefonice în timpul cursului.
Practice	Pentru petrecerea lucrărilor practice/seminare în sală este nevoie de tablă, cretă, calculatoare. Nu vor fi tolerate întârzierile studenților, precum și convorbirile telefonice în timpul cursului.

5. Competențe specifice acumulate

Competențe profesionale	CPM1 Sistemul informatic (SI) și alinierea strategiilor de afaceri CPM5 Proiectarea și dezvoltarea aplicațiilor CPM9 Îmbunătățirea proceselor
Competențe transversale	CT 1. Autonomie și responsabilitate. CT 2. Interacțiune socială. CT 3. Dezvoltare personală și profesională.

6. Obiectivele disciplinei/modulului

Obiectivul general	Obiectivul general al cursului Sisteme software complexe este de a pregăti studenții să proiecteze, dezvolte și gestioneze sisteme software scalabile, robuste și distribuite, utilizând arhitecturi avansate, tehnologii moderne (microservicii, DevOps, cloud) și principii de securitate, optimizare și integrare, pentru a rezolva provocările aplicative din medii complexe
Obiectivele specifice	- Dezvoltarea abilității de a alege și implementa arhitecturi adecvate pentru sisteme scalabile și distribuite. - Familiarizarea cu principiile comunicării între componente, sincronizarea proceselor și gestionarea datelor distribuite. - Aplicarea microserviciilor, DevOps, CI/CD, bazelor de date NoSQL și containerelor pentru a crea soluții eficiente și scalabile. - Îmbunătățirea performanței aplicațiilor și implementarea măsurilor de protecție împotriva vulnerabilităților. - Integrarea teoriei și practicii pentru a proiecta, dezvolta, testa și implementa soluții complete, în conformitate cu cerințele specifice.

7. Conținutul disciplinei/modulului

Tematica activităților didactice	Numărul de ore
	învățământ cu frecvență
Tematica activităților la proiectul semestrial	
Tema 1. Analiza domeniului și identificarea cerințelor proiectului • Definirea scopului proiectului. • Identificarea cerințelor funcționale și non-funcționale. • Realizarea unui document de cerințe (SRS – Software Requirements Specification). Rezultat: Documentul cerințelor proiectului.	8
Tema 2. Arhitecturi software avansate • Alegerea unei arhitecturi potrivite pentru proiect (monolitic, microservicii, event-driven, etc.). • Crearea unui design de nivel înalt, incluzând diagrame UML (diagrama de componente, de clase, etc.). Rezultat: Specificația arhitecturii proiectului.	8
Tema 3. Sisteme distribuite și concurente • Detalierea componentelor și modul în care acestea vor comunica între ele. • Proiectarea mecanismelor pentru gestiunea concurenței și a comunicării în sistem. Rezultat: Diagrame detaliate ale componentelor și specificații de flux.	8
Tema 4. Managementul dependențelor și integrarea continuă • Configurarea instrumentelor pentru CI/CD (Continuous Integration/Continuous Deployment). • Instalarea și configurarea mediului de lucru (framework-uri, librării). • Integrarea dependențelor esențiale. Rezultat: Mediu de dezvoltare configurat, pipeline de CI/CD activ.	8
Tema 5. Introducere în sisteme software complexe. Dezvoltarea modului de bază al aplicației • Implementarea funcționalităților de bază ale aplicației (MVP – Minimum Viable Product). • Testarea inițială a interacțiunilor între componente. Rezultat: MVP funcțional al proiectului.	8
Tema 6. Aspecte de performanță și optimizare. Implementarea funcționalităților avansate	8

Tematica activităților didactice	Numărul de ore
	învățământ cu frecvență
- Adăugarea de funcționalități suplimentare. - Optimizarea fluxurilor și îmbunătățirea performanței. Rezultat: Aplicație complet funcțională cu funcționalități avansate.	
Tema 7. Managementul datelor în disteme doftware complexe - Implementarea soluției de stocare a datelor. - Gestionarea datelor distribuite, replicarea și asigurarea consistenței. Rezultat: Sistem de stocare a datelor integrat în aplicație.	8
Tema 8. Testarea și verificarea sistemelor software complexe - Crearea testelor unitare, de integrare și funcționale. - Testarea performanței și identificarea eventualelor blocaje. Rezultat: Document cu rezultatele testării și soluții pentru remedierea erorilor.	8
Tema 9. Securitatea în issteme software complexe - Auditarea securității proiectului. - Implementarea măsurilor pentru prevenirea vulnerabilităților. - Testarea aplicației împotriva atacurilor cibernetice comune. Rezultat: Aplicație securizată, cu documentație privind măsurile de securitate.	8
Tema 10. Sisteme software pentru IoT și Cloud - Integrarea aplicației cu servicii Cloud sau dispozitive IoT, dacă este cazul. - Pregătirea unui demo pentru prezentarea finală. Rezultat: Proiectul finalizat și funcțional, gata pentru prezentare.	8
Total:	80

8. Referințe bibliografice

Principale	- Dorin Bocu, R. Bocu, <i>Întroducere în universul ingineriei softului : Despre abordarea metodologică în industria softului: Întroducere în modelare. Perspectiva UML</i>, București MATRIX ROM 2023 ISBN: 978-606-25-0816-6 - Дж. Рамбо, М. Блаха, <i>UML 2.0 : Объектно-ориентированное моделирование и разработка</i>, Москва Санкт-Петербург Нижний Новгород Питер 2007, ISBN: 5-469-00814-2 - Software Engineering, tutorial online, https://www.infobooks.org/pdfview/14007-software-engineering-tutorialspoint/ H. Sayama, <i>Introduction to the Modeling and Analysis of Complex Systems</i>. Available: https://archive.org/details/introduction-to-the-modeling-and-analysis-of-complex-systems. S. Matthews, T. Newhall, and K. C. Webb, <i>Dive into Systems</i>. Available: https://diveintosystems.org. - Defense Acquisition University, <i>Systems Engineering Fundamentals</i>. Available: https://www.infobooks.org/free-pdf-books/engineering/systems-engineering/. - Infobooks.org, <i>Software Engineering</i>. Available: https://www.infobooks.org/free-pdf-books/computers/software-engineering/. - Universitatea Politehnica Timișoara, "Proiectarea Sistemelor Software Complexe – Curs 1: Introducere". Available: https://www.aut.upt.ro/staff/diercan/data/PSSC/curs-01.pdf.
Suplimentare	M. Kleppmann, <i>Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems</i>. Sebastopol, CA, USA: O'Reilly Media, 2017. L. Bass, P. Clements, and R. Kazman, <i>Software Architecture in Practice</i>. 3rd ed., Boston, MA, USA: Addison-Wesley, 2012.

	3. S. Newman, <i>Building Microservices: Designing Fine-Grained Systems</i>. 2nd ed., Sebastopol, CA, USA: O'Reilly Media, 2021. 4. R. C. Martin, <i>Clean Architecture: A Craftsman's Guide to Software Structure and Design</i>. Boston, MA, USA: Prentice Hall, 2017. 5. A. S. Tanenbaum and M. Van Steen, <i>Distributed Systems: Principles and Paradigms</i>. 2nd ed., Upper Saddle River, NJ, USA: Pearson, 2007.
--	---

9. Evaluare

Periodică		Curentă	Studiu individual	Proiect/teză	Examen
EP 1	EP 2				
Învățământ cu frecvență					
10%	10%	10%	-	30%	40%