Course title	Analysis and Design of Information Systems				
Course code	DIS501				
Course type	Compulsory				
Level	Postgraduate				
Year / Semester	1st / 2nd				
ECTS	7.5	**Lectures / week**	1	**Laboratories / week**	
Course purpose and objectives	The success of an information system is fundamentally contingent upon a well-structured, design that ensures correctness, efficiency, scalability, and user acceptance. Whether employing traditional or contemporary software development methodologies, the analysis and design of information systems serve as a cornerstone of system development, influencing the system’s long-term viability and impact. In the absence of “just enough design”, organizations often face significant challenges, including low user satisfaction, poor usability, difficulties in workflow integration, cost overruns, and, ultimately, system abandonment. A rigorous and sound approach to information systems analysis and design provides a framework for the effective development of technological solutions. This approach is essential to prevent inefficiencies associated with fragmented, ad hoc, or incomplete processes, which often lead to delays, increased costs, and failure to meet stakeholder expectations. The success of an information system is not merely a function of technical correctness; rather, it is heavily dependent on the degree to which it aligns with user needs, business objectives, and organizational workflows. A critical aspect of effective information systems design is the active engagement of end-users throughout the entire development lifecycle. Users are not passive recipients of technology but active participants in its creation, implementation, and evolution. Their insights, experiences, and challenges provide valuable input that enables system designers to develop solutions that genuinely address real-world operational needs. A failure to account for user perspectives often leads to systems that, while technically sound, fail to gain adoption due to usability issues, misalignment with business processes, or lack of perceived value. A structured analysis of user workflows, pain points, and strategic objectives allows for the development of tailored systems that enhance productivity, decision-making, and overall user satisfaction. A key determinant of an information system’s effectiveness is the integration of User-Centered Design (UCD) principles. UCD ensures that technology solutions are not only functional but also intuitive, user-friendly, and				

	capable of delivering an optimal interaction experience. By prioritizing usability, accessibility, and user engagement, developers can create systems that facilitate seamless human-computer interaction, reduce cognitive load, and enhance overall system adoption. This course offers a comprehensive blend of theoretical knowledge and practical application, equipping students with the skills necessary for systematic information systems analysis and design, with a particular emphasis on human-centered design principles. Students will be introduced to a variety of methodologies, techniques, and tools that enable them to effectively analyze, model, and design information systems.Special attention is given to problem formulation and analysis through Soft Systems Methodology (SSM), the elicitation and interpretation of user requirements via user stories and personas, and the structured modeling of these requirements using both Object-Oriented and Structured systems analysis approaches. Additionally, the course covers the fundamentals of usability and interaction design, equipping students with the skills to develop system prototypes that adhere to human-centered design principles.
Learning outcomes	After the completion of the course, students will be able to: CLO[1] Develop a comprehensive and critical understanding of the key methods, techniques, and tools used in information systems analysis and design, integrating both Object-Oriented and Structured systems approaches, while applying User-Centered Design (UCD) principles to enhance usability, accessibility, and overall user experience. CLO[2] Critically evaluate, compare, and contrast various software development methodologies, assessing their strengths, limitations, and applicability across different business environments and system contexts. CLO[3] Analyse, deconstruct, and synthesise user requirements, business rules, and operational constraints, integrating them into a coherent and logically structured information system design that aligns with organizational goals and strategic imperatives. CLO[4] Construct rigorous, well-defined system specifications and produce high-quality, evidence-based documentation and modeling artifacts, utilising state-of-the-art tools for information system modeling and prototyping.

CLO[5] Collaborate effectively in a team setting to analyze and design a complex, real-world information system, gaining hands-on experience in requirements analysis, problem structuring, system modeling, and early-stage implementation, while demonstrating critical reflection on design decisions and iterative improvements.

A detailed breakdown of Course Objectives based on the Course Learning Outcomes can be found below:

1. Knowledge	By completing the course, you will be able to: C.O.[1] Demonstrate a comprehensive understanding of the key methods, techniques, and tools in information systems analysis and design, utilizing both Structured and Object-Oriented approaches. C.O.[2] Develop familiarity with major software development methodologies and conduct comparative evaluations of their suitability for different business environments. C.O.[3] Acquire knowledge of the fundamental principles of User-Centered Design (UCD) and its impact on usability and user acceptance of information systems.
2. Skills	C.O.[5] Apply appropriate software development methodologies to various types of information systems, considering both business and user needs. C.O.[6] Identify, prioritize, and evaluate system requirements, as well as develop both high-level and detailed models that define system specifications. C.O.[7] Design comprehensive system specifications, produce high-quality documentation, and utilize modern modeling techniques in system development.

<table>
<tr><td></td><td colspan="3"><table>
<tr><td></td><td>C.O.[8] Design and develop system specifications using UML tools and prototyping software.</td></tr>
<tr><td>3. Competencies (Responsibility and autonomy)</td><td>C.O.[9] Collaborate effectively in teams to manage real-world case studies and develop a functional prototype of an information system.
C.O.[10] Employ critical thinking and data analysis to enhance system design based on user needs and feedback.
C.O.[11] Implement Object-Oriented Design (OOD) principles to create detailed models that facilitate system implementation by developers.</td></tr>
</table></td></tr>
<tr><td><b>Prerequisites</b></td><td></td><td><b>Required</b></td><td></td></tr>
<tr><td><b>Course content</b></td><td colspan="3"><b>Week 1: Introduction to Systems Analysis and Design</b>
The first week introduces fundamental concepts related to information systems analysis and design. Students will explore the definition of a system, the key roles involved in system development, and the foundational elements that influence system design. The concept of the 5Ps (People, Process, Product, Project, Problem) will be presented, highlighting how these elements shape the system development process. Additionally, students will examine how information systems impact business and technological environments, emphasizing the importance of strategic thinking in system development. The session will also introduce the key differences between Structured and Object-Oriented approaches to system design, laying the foundation for subsequent weeks. By the end of this week, students will be able to identify the roles and stages involved in the lifecycle of an information system and connect key system development elements to real-world business challenges. <b>[CLO1, CLO2]</b>
<b>Week 2: Traditional and Modern Methodologies in System Development</b>
This week focuses on the methodologies and processes used in information system development, comparing traditional and modern approaches. Students will explore the differences between Structured Analysis and Object-Oriented Design, as well as various software development methodologies, including Waterfall, Spiral, V-Model, Agile methodologies (Scrum, Kanban), and DevOps. Special emphasis will be placed on how each methodology adapts to different project requirements and the criteria for selecting the most appropriate approach based on system constraints and challenges. Students will also engage with concepts such as Continuous Integration, Iterative Development, and the role of flexibility in modern business environments. By the end of this week, they will be able to critically evaluate</td></tr>
</table>

different development methodologies and apply them to real-world system design scenarios.**[CLO1, CLO2]**

Week 3: Understanding and Structuring Problems Using Soft Systems Methodology (SSM)

This week centers on managing complex and ill-defined problems using Soft Systems Methodology (SSM). Students will explore how SSM is applied in organizational contexts, contrasting hard and soft systems thinking. The key stages of SSM, including the creation of Rich Pictures, Root Definitions, and Conceptual Models, will be examined to demonstrate how problem analysis can lead to more effective system solutions. Additionally, the importance of participatory design and stakeholder involvement in system analysis will be discussed. **[CLO2, CLO3, CLO5]**

Week 4: Requirements Engineering – Information Gathering, Functional and Non-Functional Requirements

In the fourth week, students will focus on Requirements Engineering and the process of gathering and managing system requirements. Various techniques for requirements elicitation will be explored, including interviews, questionnaires, workshops, observations, and document analysis. The distinction between Functional and Non-Functional Requirements will be emphasized, along with the importance of clarity, completeness, and consistency in system requirements. The role of User Stories in requirement analysis and system design will also be discussed. By the end of the week, students will be able to identify, document, and articulate system requirements in a way that facilitates future development and documentation. **[CLO3, CLO4]**

Week 5: Introduction to Object-Oriented Analysis and Design (OOAD)

This week introduces students to the core principles of Object-Oriented Analysis and Design (OOAD), a fundamental approach for modern information system development. Students will gain an understanding of the Object-Oriented paradigm and its distinctions from Structured Analysis. Key concepts such as objects, classes, inheritance, polymorphism, and code reuse in system design will be explored. Emphasis will be placed on how Object-Oriented approaches facilitate the modeling of complex systems with improved clarity and scalability. By the end of this week, students will be able to assess the appropriateness of Object-Oriented Analysis and apply its principles to basic design problems. **[CLO1, CLO4]**

Week 6: UML Use Case Diagrams

This week focuses on the importance of Use Case Modeling in UML-based system design. Students will examine how Use Case Diagrams help document user interactions with the system and identify core functionalities required to meet business and technical needs. Key concepts such as actors, use cases, associations, include, and extend will be introduced through practical examples. The role of Use Case Diagrams in requirement gathering and system design will also be discussed. By the end of this week, students will be able to develop Use Case Diagrams for an information system and integrate them into the broader system design process. **[CLO1, CLO3, CLO4, CLO5]**

Week 7: UML Conceptual Class Diagram – Object Model

This week introduces students to Conceptual Class Diagrams in UML, which form the foundation of Object-Oriented Analysis and Design. Students will learn how to identify and define key system classes, specifying attributes and relationships between them. Topics such as multiplicity, aggregation, and composition will be covered, emphasizing clear system modeling. Through case studies, students will analyze existing systems using Class Diagrams, bridging the gap between requirements analysis and data structure design. **[CLO1, CLO3, CLO4, CLO5]**

Week 8: UML Dynamic Model – Object Sequence Diagrams

This week focuses on the dynamic behavior of systems through UML Object Sequence Diagrams. Students will explore how these diagrams depict communication flows between objects and how system operations are executed. Concepts such as messages, lifelines, activation bars, synchronous and asynchronous communication will be discussed using real-world system examples. The interconnection between Use Case Diagrams, Class Diagrams, and Sequence Diagrams will be emphasized. **[CLO1, CLO3, CLO4, CLO5]**

Week 9: UML Dynamic Model – State Transition Diagrams

This week introduces State Transition Diagrams as a tool for modeling system behavior. Students will examine how state transitions represent changes in an object's lifecycle, particularly in event-driven systems. Key topics include states, transitions, events, and actions, with practical examples from banking applications, e-commerce, and access control systems. **[CLO1, CLO3, CLO4, CLO5]**

Week 10: Structured System Analysis – Data Flow and Entity-Relationship Diagrams

Returning to Structured Systems Analysis, this week focuses on Data Flow Diagrams (DFD) and Entity-Relationship Diagrams (ERD). Students will explore how DFDs model data flows within a system, while ERDs define data structures and relationships. The importance of these models for structured documentation and their comparison to UML techniques will be discussed. **[CLO1, CLO3, CLO4, CLO5]**

Week 11: Human-Computer Interaction (HCI) and UX Design

This week focuses on Human-Computer Interaction (HCI) and the role of User Experience (UX) Design in information systems development. Key principles of User-Centered Design (UCD) will be examined, along with techniques for enhancing usability, accessibility, and user experience. Topics include prototyping, UX evaluation methods, and modern UI/UX design trends. By the end of this week, students will be able to integrate UX principles into system design to improve usability and user experience. **[CLO1, CLO4, CLO5]**

Week 12: From Design to Implementation and Quality Assurance

	This week explores the transition from system design to implementation, discussing how UML models translate into code while maintaining design consistency. Topics include Model-Driven Development (MDD), incremental implementation strategies, and Quality Assurance (QA). Students will also examine testing methodologies, including Unit Testing, Integration Testing, System Testing, and User Acceptance Testing (UAT), with an emphasis on automated testing and software reliability. **[CLO1, CLO4]** **Week 13: Review and Knowledge Synthesis** The final week is dedicated to reviewing key concepts covered throughout the course and preparing for final assessments. Students will revisit fundamental principles of system analysis and design, engage in Q&A sessions, and participate in feedback discussions on their group projects. Exam preparation strategies and example questions will be provided to support their study process. **[CLO1, CLO2, CLO3, CLO4, CLO5]**
Teaching methodology	Mix of lectures, active learning techniques, and activities. More precisely: ▪ Interactive online lectures ▪ Online Group activities/discussions ▪ Online activities and tutorials ▪ Formative and Summative Assignments ▪ Case study discussion and peer review exercises ▪ Web links and educational videos ▪ Online quizzes In addition to the final examination and the four interactive assignments, the course assessment includes a major group project, which constitutes a significant component of the overall evaluation. This project will require students to conduct a comprehensive analysis and modeling of an information system based on a real-world case study. Students will collaborate in teams of three to four members to complete this task. As part of the project, students are expected to provide a well-documented selection of a system development methodology, conduct a detailed requirements engineering process, and effectively model these requirements using appropriate techniques. Furthermore, they will be required to develop a prototype utilizing a prototyping tool such as Figma, or an alternative of their choice. The project will conclude in a 3-4 minute video presentation, where each group will present their work by effectively "selling" their proposed solution to the designated "user"—in this case, the course instructor. The presentation should emphasize the integrity of the system's design, its functionality, and its reusability, demonstrating how their implementation aligns with their initial design choices. Additionally, this assessment will evaluate students' ability to communicate the value and potential of their solution in a professional and structured manner.

Formative assessment is also a fundamental aspect of the course, playing a crucial role in monitoring and supporting students' learning progression. The primary objective of formative assessment is to enhance students' comprehension and knowledge by providing detailed feedback on submitted assignments. This feedback will serve as a mechanism for continuous improvement, allowing students to refine their understanding and enhance their work before the final submission of their major assessment.

This assessment approach ensures that students not only acquire theoretical knowledge but also develop essential skills in system analysis, design, prototyping, and professional communication, all of which are critical for their future careers.

Bibliography

Essential Bibliogtaphy

1. Kendall E.K and Kendall E.J (2021). *Systems Analysis and Design*. 10th edition, Pearson
2. Valacich, J. S., and George, J. F. *Modern Systems Analysis and Design*. 10th edition, Pearson, 2025.
3. Sommerville, I (2020), Τεχνολογία Προιόντων Λογισμικού, 1η έκδοση, Εκδόσεις Κλειδάριθμος
4. Kung, D.C (2024) Software Engineering, 2nd Edition, McGraw Hill
5. Pressman, R.S. (2020). Software Engineering: A practitioner's approach, 9th Edition, Mc-Graw Hill

Additional Bibliography

1. Sommerville, I. (2016). Software Engineering, Pearson
2. Additional educational material on the various modules of the course will be provided on a weekly basis through the course Moodle.

Assessment

	Percentage	CLO1	CLO2	CLO3	CLO4	CLO5
4 Interactive Activities	20%	√	√	√	√	√
Coursework	20%	√	√	√	√	√
Final Exam	60%	√	√		√	

Assignment Details:

Written Assignment	Quiz	Oral Presentation	Research Study Analysis	Software Development	Case Study

√	√	√		√	√

Formative Assessment (Non-Graded):

Peer Evaluation	Discussion Boards / Forums	Multimedia Activities	Wiki	Oral Presentation
	√	√	√	√

Language	English, Greek