



## COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Software Systems Architecture and Design	<b>PMAP7124</b>

Lecturer(s)	Department where the course unit is delivered
<b>Coordinator:</b> Vidas Vasiliauskas  <b>Other lecturers:</b> –	Department of Software Engineering Faculty of Mathematics and Informatics Vilnius University

Cycle	Level of course unit	Type of the course unit
Second	–	Compulsory

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	Autumn semester, first year of study	Lithuanian, English

Prerequisites and corequisites	
<b>Prerequisites:</b> –	<b>Corequisites (if any):</b> –

Number of credits allocated	Student's workload	Contact hours	Self-study hours
6	160	84	76

Purpose of the course unit: programme competences to be developed		
<p>The aim of the course unit is to develop key software engineering skills: designing logical and technical architecture of software systems, modelling architectural constructs such as subsystems, components and their relationships, applying architectural styles, implementing non-functional requirements including performance, security, availability, and modifiability, communicating with representatives of other professional fields, while solving problems of other fields or interdisciplinary issues.</p>		
Learning outcomes of the course unit: students will be able to	Teaching and learning methods	Assessment methods
Design high-level technical architecture of a software system	Problem-oriented teaching, case analysis, group discussion.	Laboratory assignments, examination in written form.
Design system's logical architecture based on selected viewpoints and quality characteristics.		
Apply archetypes and architectural styles.		
Design implementation of non-functional requirements such as performance, security, availability and modifiability.		
Integrate architecture design activities into software development process.		
Document software system's architecture.		
Choose development methodology		

Course content: breakdown of the topics	Contact hours						Self-study work: time and assignments		
	Lectures	Tutorials	Seminars	Practice	Laboratory work	Practical training	Contact hours	Self-study hours	Assignments
Introduction to software systems architecture and architect role	1				0		1	0	Individual reading, laboratory assignment No. 1
Web frameworks, MVC design pattern	2				2		4	4	
ORM technologies and their application	3				2		5	4	
REST API, Web server definition, API documentation	6				4		10	8	
High level technical architecture	3				2		5	4	
Development process methodologies (agile, waterfall) and architectural decisions that enable them	3				2		5	4	
Basic concepts of software systems architecture: architectural structure, viewpoint and views, quality characteristic, architect's role in a project.	3				2		5	4	Individual reading, laboratory assignment No. 2
Architecture definition process; architectural scope, concerns, principles and constraints; identifying and engaging stakeholders, communicating with stakeholders; identifying and using scenarios.	6				4		10	8	
Architectural styles; archetypes; reference models and reference architectures; architectural models, documenting software architectures.	3				2		5	4	
Viewpoint catalogue: functional, information, concurrency, development, deployment and operational viewpoints.	6				4		10	8	
Quality characteristic catalogue: performance and scalability, availability and resilience, security, modifiability and evolution quality characteristics.	6				4		10	8	
Fundamentals of Service Oriented Architecture (SOA) and Microservices	3				2		5	4	Individual reading
Basic concepts of Behaviour Driven Development (BDD) and Test Driven Development (TDD)	3				2		5	4	
Preparation for the exam (exam is taken in written form).		2					4	12	2 hours for tutorial before exam, 12 hours to prepare for the exam, 2 hours for exam.
<b>Total</b>	<b>48</b>	<b>2</b>			<b>32</b>		<b>84</b>	<b>76</b>	

Assessment strategy	Weight %	Deadline	Assessment criteria
Laboratory assignment No. 1	15%	Week 8	Students have to design and implement several use cases using learnt technologies from the curriculum. In total 1.5 point can be collected: - ~15% are for MVC implementation, - ~35% are for ORM implementation, - ~35% are for REST API implementation. - ~15% are for integrating MVC, ORM and REST API into single functional system. The penalty for exceeding the deadline is 0.25 points for each week exceeding the deadline.
Laboratory assignment No. 2	25%	End of the semester	Students have to design and document the architecture of a freely chosen sufficiently complex software system. Documentation form must satisfy requirements stated during lectures. In total 2.5 points can be collected: - ~20% are for scope, business process, use cases and scenarios documentation, - ~60% are for viewpoint documentation, - ~20% are for quality characteristic documentation.
Exam in written form	60%	Exam session	For the right to take the exam student must collect at least 0.75 points from the first laboratory assignment and at least 1 point from the second laboratory assignment. Exam consists of 12 open and semi-open questions, each question is evaluated from 0 to 0.5 points; maximum 6 points can be collected. 50%-60% of questions are related to software systems architecture, remaining questions are related to a technological platform that was taught during lectures.

Author	Publishing year	Title	Number or volume	Publisher or URL
<b>Required reading</b>				
Nick Rozanski, Eoin Woods	2012	Software Systems Architecture. Working with Stakeholders Using Viewpoints and Perspectives, 2 <sup>nd</sup> edition		Addison-Wesley Professional
Len Bass, Paul Clements, Rick Kazman	2012	Software Architecture in Practice, Third Edition		Addison-Wesley Professional
<b>Papildomi studijų šaltiniai</b>				
Joseph Albahari, Ben Albahari	2017	C# 7.0 in a nutshell		O'Reilly Media
Adam Freeman	2013	Pro ASP.NET MVC 5 (Expert's Voice in ASP.Net) 5th ed. Edition		Apress
Robert C. Martin	2008	Clean Code: A Handbook of Agile Software Craftsmanship 1st Edition		Prentice Hall
Jamie Kurtz, Brian Wortman	2014	ASP.NET Web API 2: Building a REST Service from Start to Finish 2nd Edition		Apress
Martin Fowler	2012	Patterns of Enterprise Application Architecture (2012 edition)		Addison-Wesley Professional