Course description

Course name	Course Code
Software Design and Architecture	

Teaching Staff	Department
Responsible lecturer: Vasilij Savin	Software Engineering Department Faculty of Mathematics and Computer Science Vilnius University

Level	Course Type
First (Bachelor)	Mandatory

Teaching form	Period	Language
On-site	5th semester	English/Lithuanian

Prerequisites
Object oriented programming, Software Engineering I/II

ECTS credits	Total workload (hrs)	Contact hours	Self-study
5	136	64	72

Course goals student learning outcomes
Course educational goal – give students proficiency in creating, analyzing, and evaluating large system technical designs and apply software design principles during system implementation and design. General competences: • Teamwork and collaboration Professional competences: • Create a sound technical design based on provided requirements. • Evaluate technical designs and provide constructive feedback. • Apply SOLID software design principles.

Learning outcomes	Teaching methods	Evaluation methods

Can apply SOLID software design principles	Group coding project	Project evaluation
Can collaboratively create large system technical design	Group projects	
Can provide constructive technical evaluation and feedback		
Can identify relevant system scalability requirements and adjust technical design to address them		
Can effectively collaborate, notice group dynamics changes and find constructive ways to solve intragroup conflicts	Personal reflection diary	Diary evaluation

Topics	Contact hours							Self-study	
	Paskaitos	Konsultacijos	Seminarai	Pratybos	Laboratoriniai darbai (LD)		**Visas kontaktinis darbas**	**Savarankiškas darbas**	**Assigments**
SOLID design principles	4						**4**	**8**	Group projects, personal diary, independent literature study
Scalability principles AKF Cube, microservices and monoliths	4				10		**14**	**6**	
Software Design Patterns	2						**2**	**2**	
Team dynamics/Conway's law	2						**2**	**2**	
User stories	4				6		**10**	**6**	
Coding smells	4				6		**10**	**6**	
Technical Debt	4				10		**14**	**6**	
Clean Architecture, Backend for Frontend architecture styles	4						**4**	**8**	
Common technical design pitfalls and challenges	4						**4**	**8**	
Total	**32**				**32**		**64**	**72**	

Assessment	**Grade share**	**Submission deadline**	**Assessment criteria**
Lab 1 – Requirement specification	20%	Oct 1st	Comprehensiveness of user stories and non-technical requirements, internal logical consistency. Quality of details and breakdown of the data model. Also, quality of feedback provided to a partner team will be assessed
Lab 2 – Technical design	40%	Nov 7th	Quality of diagrams, logical consistency within the final document. Also, quality of feedback provided to a partner team will be assessed.
Lab 3 – System implementation	30%	Dec 24th	Code cleanliness, test coverage, scope of implementation, ease of deployment. It is expected that implementation includes some 3rd party integrations.
Personal diary	20%	Dec 24th	Consistency – diary should be filled every week, entries should demonstrate ability to reflect on own thinking and team dynamics

Author	**Year**	**Book name**		**Publisher**
Mandatory Reading				
R.C. Martin	2008	Clean Code		Addison-Wesley
Matthias Noback	2019	Object Design Style Guide		Manning
R.C. Martin	2017	Clean Architecture: A Craftsman's Guide to Software Structure and Design		Addison-Wesley
Supplemental Reading				
John Oosterhout	2018	Philosophy of Software Design		Yaknyam Press
K. Beck	2007	Implementation Patterns		Addison-Wesley