Software Engineering Institute			Semester 4. of the curriculum 2024-25-1			
Name of the subject:	Code of the subject:	Credits:	Weekly hours:			
				lec	sem	lab
Software Technology and Graphical User Interface Design (VK)	NSXSG1EBNE	5	full-time	2	0	3
Responsible person for the subject: Sipos Miklos			Classification: Assistant lecturer			
Subject lecturer(s): Sipos Miklos, Krutilla Zsolt						
Prerequisites:	NSXHF1EBNE	Advanced Development Techniques				
Way of the assessment:	Exam					

Course description	
Goal:	During the labs, the students familiarize themselves with the MVVM and MVC patterns using the C# language. During the lectures, the students familiarize themselves with the GoF basic design patterns.
Course description:	Uninterpretable during VK (exam only).

Lecture schedule	
Education week	Topic
1.	Uninterpretable during VK (exam only).
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	
13.	
14.	

Mid-term requirements	
Conditions for obtaining a mid-term grade/signature	**Uninterpretable during VK (exam only).**

Assessment schedule	
Education week	Topic
	Uninterpretable during VK (exam only).

Method used to calculate the *mid-term grade* (to be filled out only for subjects with mid-term grades)

Type of the replacement	
Type of the replacement of written test/mid-term grade/signature	**Uninterpretable during VK (exam only).**

Type of the exam (to be filled out only for subjects with exams)
Students can take the final exam in the exam season, if they have signature from one of the previous semesters. The final exam will be from the lecture materials in written form, where students must answer test-like

questions and essay-like questions as well. Test questions mean that the student must select the appropiate answer from a set. Essay questions mean that students must answer in sentences in their own words, exaplaining their way of thinking.
Calculation of the exam mark (to be filled only for subjects with exams)
The final grade will be the exam's grade. Students must reach at least 50% of the achievable points.
Final grade calculation methods:
One possible scale regarding the grading: **Failed (1): 0-49%** **Passed (2): 50-61%** **Satisfactory (3): 62-73%** **Good (4): 74-85%** **Excellent (5): 86-100%**

References	
Obligatory:	The lecture and lab materials (codes, slides etc.) provided by the teacher to the students during the semester. https://nik.siposm.hu/sgui
Recommended:	Martin Fowler: Refactoring (a.k.a. The Refactoring Bible) Design Patterns: Elements of Reusable Object-Oriented Software, 1994, ISBN-13: 978-0201633610 Microsoft official C# documentation: https://docs.microsoft.com/en-us/dotnet/csharp/ Design Patterns: http://dofactory.net/net/design-patterns
Other references:	Horváth Rudolf: Common Design Patterns Aniruddha Chakrabarti: Design Patterns (GoF) in .NET Michael Feathers: Working Effectively with Legacy Code Joshua Kerievsky: Refactoring to Patterns