Tentative Syllabus
CS 308: Software Engineering
Fall 2023

Lecture Hours:	Mondays	16.40-19.30	UC G030
Lab Hours:	Tuesday	11.40-12.30	FASS G062
	Thursday	14.40-16.30	Online

Instructor: Assoc.Prof.Dr.Tolga Ovatman (tolga.ovatman@sabanciuniv.edu)
Office hours (on zoom via appointment): Tuesdays 09.00-10.00

TAs: Elif Ecem Şamlıoğlu (esamlioglu@sabanciuniv.edu)
Mekan Myradov (mekan.myradov@sabanciuniv.edu)
Meryem Berfin Özkök (bozkok@sabanciuniv.edu)
Oğulcan Günaydin (ogulcan.gunaydin@sabanciuniv.edu)
Zeynep Işık (zeynep.isik1@sabanciuniv.edu)

DESCRIPTION

This course is an introductory level course to the fundamentals of software engineering. One focus of this course is to provide software engineering knowledge and skills that students can put into immediate practical use. Topics covered include requirements engineering, architecting and designing software systems, quality assurance, managing software process, and getting familiar with the state-of-the-art software development tools.

TENTATIVE PROGRAM

Week 1 Introduction - Software Projects and Software Development Lifecycle
Week 2 Agile Processes, Planning and Estimation
Week 3 Software Projects: A Quick and Complete Iteration
Week 4 Domain and Requirements Engineering
Week 5 Requirements Analysis and Modeling
Week 6 UML class/sequence diagrams - Structural/Behavioral Modeling
Week 7 Software Design Principles
Week 8 Software Design Patterns
Week 9 Software Architecture: Architectural Patterns
Week 10 Software Architecture: Backend and Frontend Software Development
Week 11 Code Management, Versioning and Code Review
Week 12 Software Testing 1: Unit Testing and Related Technologies
Week 13 Software Testing 2: Beyond Unit Testing: Integration, Performance, etc.
Week 14 Software as a Product and Software Quality

GRADING POLICY

	contribution (%)
Midterm	25
Final	35
Project	40

Be aware that, since the term project is an integral part of the course, getting good grades in the exams and quizzes is not sufficient to pass the course! To be assessed as successful,

students must significantly contribute to their project group's success.

COURSE PROJECT

Short topic: This term's topic is to prepare a system that can collect information on musical data and personalize this to offer further services such as recommendation, list creation, etc. A list of MVP features will be separately shared with the students during the first couple of weeks.

Students will be working in teams and will be free to choose any technology stack they want except the following restrictions:

- Teams should use Jira for work tracking and Github for code repository
- Teams should follow the organizational structure explained below
- Project should contain at least the following components:
    - Backend and API component,
    - Web-application component displayed via a browser
    - A **native** mobile application wither in Android or in iOS (not both)

Teams: Each team is expected to consist of 6-9 students. Even though the teams would be acting in a cross-functional fashion, there would be fixed responsibilities that would focus more on the designated area, as follows:

- Team leader (1 student): Team leader is responsible from project governance and work tracking. Tuesday lab hour will be dedicated to team leader meetings that would report and discuss the overall project development. Team leader is expected to contribute in the development of the project as well.
- Backend development group (2-3 students): Backend developers will provide data governance, data access as well as an API for the project.
- Web development group (2-3 students): Web development group is going to prepare a web application that will consume the API prepared by the backend team targeted against a single browser. Development should include an elegant, responsive UI development as well.
- Mobile development group (2-3 students): Mobile developers should either use XCode-Swift or Android Studio to develop a mobile application for the project.

Teams can decide on the rest of the technology stack as desired and also decide to assign more-less resources to specific components according to the team structure.

TURN-IN and LATENESS POLICY

Project progress demos and assignments (if any) can **either** be turned on time or two days late (not both). If turned in late, evaluation will be done over 60/100 points. No assignments or progress demos will be accepted more than 48 hours late for any reason!

COLLABORATION POLICY

Project groups may discuss ideas about their projects with other groups, but they should not share any project artifacts with others (e.g., requirement documents, design documents, source code, etc.) Each group is responsible in making sure that their artifacts are well protected from others.

MAKE-UP POLICY

If you do miss an exam, no makeup exams will be granted unless you have a documented emergency situation and notify the instructor within 48 hours after the exam date.

TEXTBOOK

There is no textbook for this course. The following, however, is a list of suggested books:

- Object Oriented Software Engineering: Practical Software Development using UML and Java, Timothy C. Lethbridge and Robert Laganiere, McGraw Hill, ISBN 0-07-710908-2
- Software Engineering, Ian Sommerville, Pearson, ISBN 0-13-394303-8

Additionally, at the end of each week's presentation you can find related resources.

OTHER POLICIES

- Tuesday Labs will be face-to-face with the sole participation of team leaders. There will be no demonstration, only Jira boards will be discussed.
- Thursday labs will be online, each team is going to make a 15 minute demonstration of their achievements.
- Attendance to project progress meetings will be graded. At most two of the planned meeting absence will be tolerated, rest will be subject to penalty. For team leaders, any absence for project meeting should be replaced by another member (same rules for absence apply to project meetings as well).
- Project groups that achieve announced MVP definitions will be guaranteed to get 60/100 (in the absence of any other penalties). The rest of the points will be awarded regarding the relative comparison of the projects.
- For the instructor's office hours, send an email to the instructor, indicating a couple available time slots within in the office hours, so that the meeting is scheduled beforehand.
- The syllabus may also be revised according to the reassessment to be made YÖK. The content to be delivered is certain, but the method of course delivery as well as all other details governing the course are subject to change.