Course Syllabus

SE 6387: Advanced Software Engineering Project

T/R, 2:30 PM – 3:45 PM
CB 1.310

Spring 2023

Instructor

Rym Z. Wenkstern
Phone: (972) 883 2091
email: rymw@utdallas.edu
Office hours: in-person and by appointment, on Thursdays, 5:30 – 6:30 PM

Prerequisites

- *Required: (SE 6361 or SE 6362) and (SE 6329 or CS 6359). Corequisite: SE 6367*
  - *SE 6361: Advanced Requirements Engineering or SE 6362: Advanced Architecture and Design*
  - *SE 6329: Object-Oriented Software Engineering or CS 6359: Object-Oriented Analysis and Design*
  - SE *6367: Advanced Software Testing*
- Basic knowledge of app development

Course Modality and Expectations

Instructional Mode

The course will be taught face-to-face. The professor and students meet in class according to the schedule.

Course Platform

The course material, i.e., lecture notes, project deliverable templates, and project schedule, will be available in *elearning*.

Office Hours and Communication with Professor

Office Hours. The office hours will take place at the professor's office by appointment on Thursdays between 5:30-6:30 PM or at a date and time agreeable to the professor and the student. A sign-up sheet will be made available to students every week.

Communication with Professor. Given the large volume of emails received daily, students are asked to post questions related to the project on the course's elearning *discussion board*. For personal matters, please email the professor and **mention SE 6387 in your email's subject line**. Please note that your emails will be answered within two working days under normal circumstances.

Assignments and Project

Students will work in groups of four to complete the project. One designated group member will be charged with submitting hard copies and uploading the deliverable files on OneDrive ***before*** **the beginning of class** on the day assigned.

Submissions received after the beginning of classes will not be accepted.

Groups will be formed during the first week of classes. If a group member is not participating in the group meetings and not contributing during the semester, the professor should be notified as soon as possible. No complaints will be considered during the last three weeks of the semester.

Expectations

- Students are expected to be **punctual**. Therefore, students arriving after the lectures have begun may not be allowed in class.
- Class attendance is **mandatory** and will be recorded. *Three unexcused absences will result in a deduction of 30% from your final grade*. *More than three unexcused absences will result in a deduction of 50% of your grade.*
- There will be NO extra credit work for this course.
- Cell phones should be placed in your backpacks during sessions. In addition, computers should NOT be used in the classroom during sessions.

Class Recordings

Students are expressly prohibited from recording any part of this course. Failure to comply with this University requirement is a violation of the Student Code of Conduct.

Class Materials

The instructor may provide class materials that will be made available to all students registered for this class as they are intended to supplement the classroom experience. These materials may be downloaded during the course; however, these materials are for registered students' use only. ***Classroom materials may not be reproduced or shared with those not in class or uploaded to other online environments*** except to implement an approved Office of Student Accessibility accommodation. Failure to comply with these University requirements is a violation of the Student Code of Conduct.

Course Description (Graduate Catalog)

This course is intended to provide experience in a group project that requires advanced technical solutions, such as distributed multi-tier architectures, component-based technologies, automated software engineering, etc., for developing applications, such as web-based systems, real-time systems, biomedical systems, legal systems, mobile applications, etc. Four to five students form a group to develop and maintain requirements, architecture, detailed design, implementation, testing, and traceability relationships. Best practices in software engineering will be applied.

Course Description – Spring 2023

This course is intended to provide extensive hands-on experience in dealing with various software development issues. It involves a semester-long group project spanning problem definition, feasibility analysis, project planning and management, requirements specification and analysis, software design, implementation, and testing. UML is used in the context of an iterative, use case-driven, architecture-centric process. Deliverables are produced through a CASE tool that promotes an agile workflow by synchronizing changes in the various models and the code.

Course Learning Outcomes

- ❑ To provide experience in a group-oriented software development project
- ❑ To apply and practice advanced software project planning and management techniques and disciplines
- ❑ To apply and practice advanced software development processes and methodologies
- ❑ To apply and practice advanced requirements engineering techniques
- ❑ To apply and practice advanced software architecture and design techniques
- ❑ To apply and practice advanced software testing technique

Course Objectives

Planning and Management	Develop a project plan
	Execute the project plan and monitor the project activities
	Follow an iterative and agile process (UP)
Case Tools	*Microsoft Project*
	GitHub https://github.com/

Requirements	Gather and document requirements
	Create a domain model
	Analyze requirements with use cases
	Ensure traceability between use cases and requirements
	Validate requirements through model execution
	Define black-box test cases
Case Tools	*IBM Rational DOORS*

Design	Write high-frequency UML
	Design with architectural layers
	High-Level Design: represent system architecture design with Composite Structures, Component Diagrams, and Subsystem Diagrams
	Detailed Design: design object solutions: assign responsibilities to objects, design collaborations, design with patterns
	Verify design: relate analysis and design artifacts.
Case Tools	*IBM Rhapsody*

Implementation	Implement design solutions
Prog. Language	

Test & Quality Assurance	Define white-box test cases
	Test your solution
	Assess the quality of the implementation
Case Tools	*JUnit https://github.com/KentBeck/junit/wiki*
	Maverix http://www.maveryx.com/

	Top 50 Testing Tools *https://www.guru99.com/testing-tools.html*

Presentations and Demos	Presentations throughout the semester
	Final project demo and product video

Academic Calendar

First Day of Class	Tuesday, January 17
Last Day to Add/Swap	Tuesday, January 24
Census Day	Wednesday, February 1
Last Day to Drop Without a "W"	Wednesday, February 1
Last day of Withdrawal	Thursday, March 30
Spring Break	Monday, March 13 – Sunday, March 19
Last day of Class	Thursday, May 4

Final Project Presentation and Demo	**Thursday, May 4**
Final Project Deliverables	**Thursday, May 4**

Comet Creed

The UT Dallas student body voted on this creed in 2014. It is a standard that Comets choose to live by and encourage others to do the same:

"As a Comet, I pledge honesty, integrity, and service in all that I do."

The descriptions and timelines contained in this syllabus are subject to change at the discretion of the Professor.