1. **Course number and name:** CMSE 406 Graduation Project II
2. **Credits and contact hours, and categorization:** Credits: 3, Contact hours: 1, Engineering Sciences and Design
3. **Course Coordinator:** Prof. Dr. Duygu Çelik Ertuğrul
4. **Textbook:** None

**Other supplemental materials:** None**.**

1. **Specific course information**
2. **Catalog description:** This course is the sequel to CMSE 405. It consists in the implementation of a realistic, preferably interdisciplinary, engineering capstone project emphasizing engineering design principles on a software engineering topic. It is carried out by a team of students under the supervision of an instructor. The team must complete the detailed design and implementation of the preliminary design they started in the CMSE 405 course. It is an extended exercise in the professional application of the skills and experience gained in the undergraduate program. The team must perform a presentation and submit a detailed final report which documents the design, implementation and testing of a software project.
3. **Prerequisite:** CMSE 405
4. **Required/elective/selected elective:** Required
5. **Specific goals for the course**
6. **Course outcomes:** At the end of the course, each student should be able to
7. Identify, formulate, and solve a given open-ended, real-world, complex problems using appropriate mathematical and scientific knowledge, and approaches.
8. Learn new knowledge and skills required to realize the project in an independent way through the guidance of the supervisor.
9. Apply core computing knowledge such as programming, database, algorithm analysis, etc.
10. Work on a CS-related project of a reasonable complexity.
11. Produce a complete design of the system and implement it using modern tools, sound software engineering principles and industry standards.
12. Use multiple realistic constraints when designing the system.
13. Create new knowledge, engineering solutions, and functioning hardware.
14. Work in multi-disciplinary teams and provide leadership on tools-related problems that arise in multi-disciplinary work.
15. Communicate engineering designs to a professional standard.
16. Manage time and resources in a professional manner.
17. Apply testing concepts and techniques to the system.
18. Contribute new ideas, demonstrating knowledge of contemporary issues (as they relate to computer / software engineering)
19. Perform quality assessment on the developed system.
20. Be aware of ethical issues as they relate to software engineering.
21. Use project management tools and techniques through the project lifecycle.
22. Give an effective oral presentation.
23. **Student outcomes listed in Criterion 3**
24. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
25. an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
26. an ability to communicate effectively with a range of audiences
27. an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts
28. an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives
29. an ability to develop and conduct appropriate experimentation, analyze, and interpret data, and use engineering judgment to draw conclusions
30. an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.
31. **Topics covered**

There are no formal classes for this course. The topics to be covered during weekly meetings are tentatively as follows:

1. Review of the project concept, requirements and the design document format and contents.
2. Oral presentation of students on the related standards and design requirements
3. Determination of the detailed implementation methods and tools
4. Actual implementation of the project, together with test case scenarios
5. Testing
6. Documentation of the developed project
7. Review of coding and documentation standards, project binder requirements, final presentation, and demonstration.
8. Submission of the planning and final reports that summarizes project details
9. Oral presentation of outcomes.