

Course Title:	Design and development projects
Course Code:	CSE411
Program:	Master Degree In Computer Engineering
Department:	Computer Engineering
Course coordinator:	Dr. Thouraya GOUASMI
Institution:	Private Higher School of Engineers of Gafsa (ESIP)

A. Course Identification

1. Credit hours:	3 (1.5-0-1.5)
2. Course type	
a. College <input type="checkbox"/>	Department <input checked="" type="checkbox"/> Others <input type="checkbox"/>
b. Fundamental <input checked="" type="checkbox"/>	Transversal <input type="checkbox"/> Optional <input type="checkbox"/>
3. Level/year at which this course is offered:	2.2/3
4. Pre-requisites for this course (if any): supervised project I (CSE161), supervised project II (CSE261), (CSE131), (CSE221), (CSE323),(CSE332)	

1. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Self-study	Total workload
1	Traditional classroom	36	81
2	Blended		
3	E-learning		
4	Distance learning		
5	Other (Project)	45		

2. Contact Hours (based on academic semester)

No	Activity	Contact Hours
1	Lecture	-
2	Laboratory/Studio	-
3	Tutorial	-
4	Others(Project)	45
	Total	45

B. Course Objectives and Learning Outcomes

Course Description

This course is conducted through project-based learning, where students work in teams on projects. Each session includes progress tracking and guidance from the instructor. Students follow a structured development cycle model (e.g., Scrum, Agile), design UML diagrams, implement applications, create databases, and deploy their systems on a chosen platform. The course emphasizes collaborative development, software engineering methodologies, system design, implementation, testing, and project documentation.

Course Main Objectives

- ✓ Define a clear project scope, identifying requirements and objectives.
- ✓ Work collaboratively in teams, applying modern software engineering methodologies (Agile, Scrum, Kanban). Apply system analysis and design techniques, creating UML diagrams for their projects.
- ✓ Develop software solutions using appropriate programming languages, frameworks, and platforms.
- ✓ Implement, test, and deploy their applications effectively.
- ✓ Enhance presentation skills by demonstrating project outcomes and preparing a comprehensive project report.

1. Course Learning Outcomes

CLOs		Aligned PLOs
1	Knowledge and Understanding	
1.1	✓ Define a scope of work that clearly describes project objectives and requirements.	PLO.K1
1.2	✓ Work collaboratively in teams, using Agile methodologies to effectively manage tasks and workflow.	
2	Skills	
2.1	✓ Apply system analysis and design methodologies, including UML modeling.	PLO.S1
2.2	✓ Develop software systems using relevant programming languages, frameworks, and platforms, and conduct testing.	PLO.S5
2.3	✓ Present project outcomes effectively and produce a well-documented project report.	

C. Course Content

No	List of Topics	Contact Hours
1	Project Study & Feasibility Analysis <ul style="list-style-type: none"> Define project scope and objectives Conduct feasibility study and risk assessment Identify required technologies and methodologies 	6
2	Preparing a Specification Document <ul style="list-style-type: none"> Define functional and non-functional requirements System constraints, success criteria, and deliverables 	4
3	Task Distribution & Team Collaboration <ul style="list-style-type: none"> Assign responsibilities and team roles Use project management tools (Jira, Trello, GitHub) 	3
4	Project Design & System Architecture <ul style="list-style-type: none"> High-level and low-level system design UML Diagrams (Class, Sequence, Component, ERD) Database schema design and API integration 	6
4	Implementation (Coding & Development) <ul style="list-style-type: none"> Software/hardware development (Agile, Waterfall, CI/CD) Version control, security best practices Code optimization and documentation 	12
5	Testing and Maintenance <ul style="list-style-type: none"> Unit, integration, and system testing Debugging and fixing software issues 	5
6	Final Product Demonstration <ul style="list-style-type: none"> Project presentation and live demo Evaluation based on project objectives 	3
7	Writing the Final Report & Documentation <ul style="list-style-type: none"> Technical documentation and user manual Lessons learned and future improvements 	6
Total		45

D. Teaching and Assessment

1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
------	--------------------------	---------------------	--------------------

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.0	Knowledge and Understanding		
PLO.K1	<ul style="list-style-type: none"> ✓ Define a scope of work that clearly describes project objectives and requirements. ✓ Work collaboratively in teams, using Agile methodologies to effectively manage tasks and workflow. 	<ul style="list-style-type: none"> - Class discussions - Assignments - Projects 	Assignments, Quizzes, Report
2.0	Skills		
PLO.S1	<ul style="list-style-type: none"> ✓ Apply system analysis and design methodologies, including UML modeling. 		
PLO.S5	<ul style="list-style-type: none"> ✓ Develop software systems using relevant programming languages, frameworks, and platforms, and conduct testing. 	<ul style="list-style-type: none"> - Class discussions - Assignments - Projects 	Assignments, Quizzes, presentation
	<ul style="list-style-type: none"> ✓ Present project outcomes effectively and produce a well-documented project report. 		

2. Assessment Tasks for Students

#	Assessment task*	Week Due	Percentage of Total Assessment Score
1	Practical Work (written or oral)	Weekly	100%
2	Quizzes, Homework assignments	Random	00%
3	Exam	16th	100%

E. Student Academic Counseling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice:

- Office hours
- Blackboard interface
- Academic advisor
- Bibliotic

F. Learning Resources and Facilities

1. Learning Resources

Required Textbooks	<ol style="list-style-type: none"> 1. Pressman, Roger S., and Maxim, Bruce R. Software Engineering: A Practitioner's Approach. 9th ed., McGraw-Hill, 2020. 2. Sommerville, Ian. Software Engineering. 10th ed., Pearson, 2015
---------------------------	---

Essential References Materials	
Electronic Materials	<ol style="list-style-type: none"> 1. MIT OpenCourseWare: Software Project Management & Development 2. edX & Coursera: Agile Development, DevOps, Project Management 3. GitHub & GitLab: Open-source project repositories and best practices
Other Learning Materials	http://scrumalliance.org/members.cox.net/rising11/Articles/IEEEScrum.pdf https://www.scrum.org

2. Facilities Required

Item	Resources
Accommodation	Classroom board Computer lab with the necessary software Internet access
Technology Resources	Data projector

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Effectiveness of teaching and assessment.	Students, course coordinator, Alumni, Employers	Direct/Indirect
Extent of achievement of course learning outcomes.	Faculty, Program Leaders, quality department	Direct
Quality of Learning resources	Faculty, Program Leaders,	Direct, Indirect
Teaching and learning quality and effectiveness.	Students, Faculty Program Leaders,	Direct, Indirect

H. Specification Approval Data

Council / Committee	Computer Engineering Council
Date	07/02/2024