



جامعة أبوظبي  
Abu Dhabi University

Department of Computer Science  
and Information Technology

## COLLEGE OF ENGINEERING

FALL 2025

### SWE 401: SOFTWARE ENGINEERING

#### COURSE PROJECT

*Faculty: Dr. Murad Al-Rajab*

- **Due Date:**
  - ✓ Phase One Submission Due on **September 21, 2025**. Only submit through Blackboard.
  - ✓ Phase Two Submission Due on **October 19, 2025**. Only submit through Blackboard.
  - ✓ Phase Three Submission Due on **November 2, 2025**. Only submit through Blackboard.
  - ✓ Phase Four Submission Due on **November 9, 2025**. Only submit through Blackboard.
  - ✓ Phase Five Submission Due on **November 16, 2025**. Only submit through Blackboard.
  - ✓ Live Demo, **November 17 – 21, 2025**. Face to Face Presentation
- Percentage of Final Grade: 20%

#### **Sources of Information and Advice**

The output of this project will be the work of a group of **3 students** for the course project. Your report must be the work of each group only. All sources of information must be properly referenced.

To use the work of others without identifying it is considered as a plagiarism. This is a form of academic dishonesty and will not be accepted, and may lead to dismissal from the Course or a grade of **0** in the Project.

By submitting this project, you are declaring that the following is true: **We have read and understood the note on sources of information and advice. We have produced this piece of work without help from others apart from those noted. This is our own work.**

## **PROJECT TOPICS AND GUIDELINE**

### **★ Learning outcomes covered on this assessment:**

No.	Learning Outcome
1	Apply UML to create analysis and design models.
2	State measurable requirements for a software system.
3	Break down a large system into subsystems and do object design.
4	Translate system models into implementation code.
5	Compare a system against specific requirements.
6	Show effective teamwork skills.

### **★ Topics**

The centerpiece of this course is a semester-long project in which you will manage, analyze, design, and implement a software system by following the main activities, techniques, and processes of the Software Development Life Cycle (SDLC). This document provides guidelines for your final project in the SWE 401 course.

Your project should be an unconventional and innovative smart business software system or an enhancement of an existing system (an “as-is” software system). By “*unconventional and innovative*,” I mean that your project should introduce a new system with an original, intelligent, and impactful idea or functionality. Furthermore, there should be evidence that the software would be useful to its intended users.

You are encouraged to consult with me on any project-related questions. However, I also expect you to conduct the necessary background research to independently justify your project choice. You must write a proposal (project description) and post it to discourse by the due date of Phase #1.

### **★ Project Guidelines**

1. The project should address a real-life situation that can be implemented (preferably incorporating smart features and the use of artificial intelligence (AI)). For example, the following are good sources but you must NOT use them and have something different and new:
  - a) Airline system (or train, cruise, or other) ticket purchase
  - b) Shopping sites on the internet (E-commerce)
  - c) Online banks and financial institutions
  - d) Online insurance system (life, medical, homeowner, and others)
  - e) Blockchain Voting Application
  - f) Rental's system (car, video, audio, and others)
  - g) Student/ University Portal System
  - h) Robotic Delivery System
  - i) Clinic Online System (appointments, consulting)
  - j) Personal Health Monitoring
  - k) Pharmacy System
  - l) Home Automation System
  - m) Library System
  - n) Restaurant System

- o) Parking Garage Automation
- p) Hotel Management System
- q) Museum System

You can also use traditional firms if you can get the requisite permission from the appropriate people. Also, they should be able to provide you with the data to complete your project.

2. For the firm you have chosen, identify the major functions it can perform by applying the appropriate requirements elicitation techniques that you will study in this course.
3. Document each function. Figure out the process used in each function, the inputs used, and the outputs generated. However, web searching alone will be inadequate. You have to use your knowledge about the industry/firm/function to identify other data used by the system.
4. Identify at least **THREE** problems or limitations related to the functionality. The limitations could be new functions that could be useful to a potential customer or improvements over existing functions.
5. Analyze the UML models and propose a new system.
6. You should use and apply the concepts using the CASE tools.

***Note: The project should be sufficiently multifaceted. Please consult with me at the initial stage to determine whether your project meets the requirements of the course.***

## ★ Project Management

The “managerial” role between team members must be rotated; the student in charge of the project manager role in this case is responsible to make sure that things inside the project are on track, following the correct sequence; the work is done as planned, etc. Students must bear in mind that the task requires little management skills, instead of all team members as a whole shares responsibility for the work to be done, the management role then is to be shared to assure a high-quality work. Shared work can be a serious problem, when a team member will be unable, not serious or unwilling to take responsibilities, for that the role of a project manager comes up. The project grade will assess not only the project work but also the individual contribution of each team member.

## ★ Project Deliverables

### **Phase 1:**

Each group of students must identify a business need and define the requirements for either a new smart software system or an enhancement of an existing system. The proposed project should include at least **three new functions** or **three enhanced functions**. As part of this process, you must develop a written **Problem Statement** based on the issues you encounter during your research, along with a **functional model** of the proposed system. Be sure to note any potential improvements, ensuring that both the problems and improvements are directly related to the functionality provided by the system.

#### 1. **Cover Page:**

This will include University Name, Department Name, Course No, Course Name, Title of the project, Student IDs & Names of the team members, Instructor Name, and Date of submission.

#### 2. **Project Description (The Software System Proposal) (1-2 pages or 500 - 700 words):**

This will include:

- Introduction and definition of the **problem** – Clearly state what the project is about and the specific problem it aims to solve. It is recommended to include supporting statistical evidence (with references where appropriate) to justify the significance of the problem.
- General overview and main functionality of the system.
- Stakeholders of the system and the intended users – how the user will benefit from using your system.
- Description of the current system (if any) – “as-is system”.
- Key problems and challenges that you may face during the project’s development.
- Technologies that you will use (programming languages, frameworks, platforms, libraries, databases, etc).
- Team members brief CV (brief back ground information, technical skills, and the involvement in the project).

### **3. The Market Potential (between 200 and 400 words).**

- User interest – why the user will be interested using your software system.
- Any social and environmental impact (if any).
- Similar or related software systems – Identify and describe existing software systems or applications that are similar or related to your proposed system (provide references). You must also include a **benchmark comparison table** that highlights the differences between the current systems and your proposed system. The table should clearly show the strengths and limitations of the existing solutions, and how your proposed system provides improvements or new functionalities.
- Novelty of your proposed idea – what makes your software system innovative and different than others existing (you can use references at this stage).

### **4. The Project Management Plan (between 2-3 pages).**

- Define Activities and Sequence Activities (*Workplan*)
- Software Project Methodology Selection and justification of your selection.
- Develop the schedule (*Gantt Chart*)
- Determine Budget (*if any*)

### **Phase 2:**

1. Provide a **system context diagram** showing the system, external actors/systems, and data flows.
2. **Requirements Engineering (1-3 pages)**  
Begin with the requirements elicitation technique(s) you plan to use in order to collect the project requirements. Clearly explain why these techniques are appropriate, and then analyze the outcomes of your data collection.
3. **System users (1/2 page)**  
Identify and describe all types of users who may interact with or benefit from the software system. Explain their roles and how they will use the system.
4. **Functional Requirements: (1-2 pages)**  
Provide a detailed list of all functional requirements of the system. Each requirement should be described in moderate detail to clearly convey the system’s intended functionality.
5. **Non-Functional Requirements: (1-2 pages)**  
Provide a detailed list of all non-functional requirements and quality constraints of the system (e.g., performance, security, usability, reliability, scalability). Each requirement should be explained in moderate detail.

6. **Use Case Modeling:**
  - UML use case diagram covering the major use cases of the system.  
Provide detailed text descriptions of at least **three (3)** use cases.
7. **Requirements Specifications**  
Write at least **three (3)** Natural Language or Structured Natural Language specifications for the system requirements.
8. **Traceability Matrix (without weight)**

### **Phase 3:**

In this phase, you will make the **analysis object model** and **dynamic model** of the system you proposed in Phase 1. Make the following diagrams:

- 1) **Activity Diagram:**
  - Using the UML use cases modeled in Phase 2, draw activity diagrams that capture key workflows of your system.
  - Provide **at least three (3)** UML Activity Diagrams.
- 2) **Class Diagram:**
  - Based on your Phase 2 UML use cases and the Problem Statement, create a class diagram that identifies **entity, boundary, and control** classes.
  - Include relevant **attributes, operations (methods), and relationships** (association, aggregation/composition, generalization).
  - Specify **multiplicities** for all associations.
- 3) **Sequence Diagram and Communication Diagram**
  - Using the Phase 2 use cases and your class diagram, develop UML **Sequence Diagrams** to show object interactions over time.
  - Provide **at least three (3)** Sequence Diagrams.

### **Phase 4:**

In this phase, you will make the **system design model** of the proposed system you used in *Phase 2* and *Phase 3*.

1. **Initial System Design (Software Architecture)**
  - Provide at least **one system-level architectural diagram** (e.g., layered, microservices, client–server – 2 Teir, 3 Teir, etc.).
  - Derive the design from the **functional requirements**.
  - Identify **subsystems/modules**, their **responsibilities**, and **inter-dependencies** (diagram + short description for each subsystem).
2. **Data Design (if a database is used)**
  - List each **table** with its **attributes (columns)** and **primary key(s)**.
  - Indicate **relationships** (1–1, 1–many, many–many) and important **constraints** (NOT NULL, UNIQUE).
  - Example table (primary key in **bold**):
    - **Employees**(SSN, FirstName, LastName, DepartmentId, HireDate)

SSN	Last Name	First Name	Position	Department	Office	Telephone	Email
-----	-----------	------------	----------	------------	--------	-----------	-------

3. **Initial Hardware Design (optional; required if hardware is part of the project)**
  - Show the **hardware component organization** (e.g., sensors, gateways, servers) and how it integrates with the software components.

#### 4. User Interface Design Patterns (Mockups)

- Provide **6–8 UI snapshots** (wireframes or mockups) with **brief descriptions**.
- Each snapshot must label key UI components (e.g., panels, toolbars, menus, menu items, buttons, text boxes).
- Show representative **output formats**: reports, statistics, and/or charts, with example fields and layout.

### **Phase 5 – Implementation and Testing Phases:**

You are encouraged to implement your software system as specified in the previous phases, using any appropriate programming languages, frameworks, platforms, tools, and databases of your choice.

#### **What to include in this phase:**

- **Implementation Overview:** Briefly describe the tech stack and key design decisions carried over from earlier phases.
- **Source Control & Build:** Provide a repository link and basic build/run instructions.
- **Testing Plan:** Define and execute tests (unit, integration, system/end-to-end, and—if applicable—usability or performance tests).
- **Test Evidence:** Include test cases, expected vs. actual results, and screenshots/logs of successful runs.

## ★ Schedule of Project Deliverables

#	Deliverable	Due Date	What to Submit
1	Phase 1	Sep 21, 2025	Proposal submission
2	Phase 2	Oct 19, 2025	Requirements Engineering
3	Phase 3	Nov 2, 2025	System Analysis Diagrams
4	Phase 4	Nov 9, 2025	System Design
5	Phase 5	Nov 16, 2025	System Implementation
6	Project Report	Nov 16, 2025	Full Compiled Project Report
7	Project's presentations	Nov 17- Nov 21, 2025	Live Demo

## ★ Submission

- **Final Report Submission:** A complete final version of the report must be submitted in **soft copy** through the Blackboard website of the course by **Sunday, November 16, 2025**.
- **Late Policy:** *No late submissions will be accepted.* Any report submitted after the deadline will receive a grade of **0**.
- **Plagiarism Check:** All reports will be checked using a plagiarism detection system, *including checks for AI-generated writing*. Any report found to contain plagiarized or AI-generated content will receive a grade of **0**. Students must ensure that all work is **original**, properly **referenced**, and reflects their own effort.
- **Formatting Guidelines:** Reports must be formatted with **1.5 line spacing, font size 11–12 points**, and a professional font such as **Arial, Times New Roman, Calibri, or a similar style**.
- **Group Work:** Each project must be completed in **groups of three (3) students**. Collaboration is required, and all group members share responsibility for the final submission.

## ★ Grading

Student's grade for this project will be divided into two components. The first 75% (15 marks) of the grade is the team documentation and implementation work. The other 5% (5 marks) of the grade will be based on the project presentation in front of class colleagues. The presentation will be a 10 minutes presentation for each team including Qs & As. Individual team member work will be assessed as well. So, everyone needs to participate fully in their team's efforts.

The report will be evaluated for clarity, preciseness, and effort.

## ★ Communication

Use the following notation whenever you email the Instructor regarding your Project:

**SWE401\_FirstName\_LastName\_Project\_Subject**

😊 **BEST WISHES** 😊

**Always remember:** All submitted work must be original and reflect the group's own effort. Copying and pasting complete pages, articles, or programs from the Internet, AI tools, or any other sources is strictly prohibited. If information, ideas, figures, or data are taken from external resources, proper citation and referencing are mandatory. Any form of plagiarism or unacknowledged use of AI (including AI-generated content) will result in a grade of 0, and all group members share equal responsibility for maintaining academic integrity.

*Dr. Murad Al-Rajab*