Namal University, Mianwali

Department of Computer Science

CSC-225 – Software Engineering

Complex Computing Problem – Milestone 3

Submission Deadline: 18th January, 2026 This milestone is worth **<6%>** of total marks

Instructions:

You will continue working in your pre-defined project groups; however, marks will be awarded individually based on understanding and viva performance.

Use of AI tools is allowed only for understanding and idea generation. All diagrams, designs, and prototypes must be created by the students themselves and must align strictly with the requirements defined in your approved SRS.

Objective

The objective of Project Milestone 3 is to translate the approved Software Requirements Specification (SRS) into a complete system design. This milestone focuses on visualizing system behavior, structure, and interactions using standard software modeling techniques and validating the design through stakeholder feedback and prototyping.

This milestone ensures that your proposed system is well-thought-out, internally consistent, and traceable back to the requirements before implementation begins.

Task Overview

In this milestone, you will:

1. Design the system architecture strictly according to the functional and non-functional requirements defined in your SRS.
2. Develop a Use Case Diagram that accurately represents all system actors and major system functionalities.
3. Create Data Flow Diagrams (DFDs) starting from Level 0 and expanding up to Level 2, ensuring logical decomposition and consistency across levels.
4. Design Sequence Diagrams to model interactions for key system functionalities. For complex systems, functionality may be distributed across multiple sequence diagrams instead of a single diagram.
5. Design Activity Diagrams to represent important system workflows. Multiple activity diagrams may be used where required.
6. Develop a Class Diagram that reflects system structure, including classes, attributes, operations, and relationships derived from requirements and interaction diagrams.

7. Create a Component Diagram to illustrate the high-level system components and their dependencies.
8. Prepare a Requirements–Design Traceability Table that maps key functional requirements from the SRS to corresponding design artifacts (such as use cases, DFD processes, sequence diagrams, classes and prototype). This table should demonstrate how the proposed design directly originates from documented requirements.
9. Clearly document Design Assumptions and Constraints, including any assumptions made during system design (e.g., user roles, deployment environment, usage limitations) and constraints related to technology, platform, performance, or security.
10. Identify and explain Key Design Decisions, briefly justifying important choices made during design, such as:
    - Decomposition of processes in DFDs
    - Selection of class relationships (association, aggregation, composition)
    - Distribution of functionality across multiple diagrams
11. Conduct at least two meetings with your Requirement Provider (RP) during this milestone:
    - In the first meeting, present a paper-based prototype illustrating the main system interfaces and key features.
    - In the second meeting, present a complete interactive prototype developed in Figma, covering all major system functionalities.
    - Record and document both RP meetings, including meeting minutes and evidence of prototype discussion and feedback.
12. Maintain all diagrams, prototypes, and meeting-related artifacts in the group GitHub repository in a clear and organized structure.
13. Create a LinkedIn post describing your project, including:
    - A brief overview of the project
    - Describe your role in the project and key learnings gained from this project (be realistic)
    - A link to the project GitHub repository
    - A link of the Figma prototype
    - Screenshots of the system interface
    - Mention of the course instructor: Link to Profile
    - Mention of the Requirement Provider (if they are comfortable being tagged).

Submission Guidelines:

1. Students must prepare a Design Report that includes all deliverables of Project Milestone 3, including:
    - Introduction
    - Design Assumptions and Constraints
    - Key Design Decisions
    - All system design diagrams

   - Requirements–Design Traceability Table
   - GitHub and Figma Link

2. All project materials must be uploaded to the group GitHub repository.
3. The GitHub repository should be organized with clearly named folders such as:
   - Design Report
   - Design Diagrams
   - Traceability
   - Prototypes
   - Meeting Minutes
   - Meeting Videos
4. The Figma prototype link must be included in the repository README.
5. Meeting minutes must be maintained in the shared Google Sheet and linked in the README.
6. Submit the GitHub repository link on QOBE before the deadline.

Evaluation Rubric

	Excellent 1.0	Good 0.7-0.9	Satisfactory 0.5-0.6	Unsatisfactory 0.2-0.4	Poor 0.0-0.1
Viva	Answered all questions correctly	Answered most questions correctly	Answered some questions correctly	Answered very few questions correctly	Answered no questions correctly

		Weight	Excellent 90% - 100%	Good 70% - 89%	Satisfactory 50% - 69%	Unsatisfactory 20% - 49%	Poor < 20%
Behavioral Diagrams (CLO 3) 45 marks	Use Case Diagram	10	Diagram correctly identifies all actors and major use cases; relationships are accurate and fully consistent with SRS	Diagram mostly correct with minor missing use cases or small notation issues	Diagram included but lacks coverage of key functionality or has conceptual errors	Diagram poorly modeled or inconsistent with SRS	Missing or irrelevant diagram
	Data Flow Diagram	15	All DFD levels are logically decomposed, balanced, and correctly labeled	Minor balancing or naming issues across levels but overall logic is sound	DFD levels present but decomposition is weak or inconsistent	DFDs incomplete, incorrectly decomposed,	Missing DFDs or fundamentally incorrect
	Sequence Diagrams	10	Sequence diagrams clearly model interactions for key functionalities; lifelines, messages, and order are correct	Minor issues in message ordering or completeness	Diagrams included but lack clarity or cover trivial interactions	Diagrams poorly structured or inconsistent with system behavior	Missing or incorrect diagrams

	Activity Diagrams	10	Activity diagrams clearly represent workflows with correct control flows, decisions	Mostly correct workflows with minor notation or logic issues	Diagrams included but workflows are oversimplified or unclear	Diagrams confusing, incomplete, or incorrect	Missing activity diagrams
Structural Diagrams (CLO 3) 25 marks	Class Diagram	15	Class diagram accurately represents system structure with correct classes, relationships, attributes, and operations	Mostly correct structure with minor relationship or visibility issues	Diagram present but lacks proper abstraction or consistency	Diagram poorly modeled or resembles ERD instead of class design	Missing or invalid class diagram
	Component Diagram	10	Clearly shows major system components and their dependencies; aligns with system architecture	Components identified with minor dependency or clarity issues	Diagram included but components are vague or weakly connected	Diagram confusing or conceptually incorrect	Missing component diagram
Prototype (CLO 3) 40 marks	Paper-Based Prototype	10	Prototype clearly represents main system interfaces and core features; screens are logically organized and directly derived from requirements;	Main interfaces and features are represented with minor gaps or unclear transitions; overall intent is understandable	Basic screens included but lack clarity, completeness, or logical flow	Prototype is confusing, incomplete, or poorly explained	No prototype submitted or prototype irrelevant
	Figma-Based Interactive Prototype	35	Interactive prototype fully represents major	Prototype covers most major features	Prototype present but incomplete;	Prototype poorly structured,	Missing or non-

			system features and user flows; navigation is logical; screens align well with use cases and workflows;	with minor usability or flow issues	interactions are limited or unclear	confusing, or misaligned with system functionality	functional prototype
Project Maintenance (CLO 3) 15 marks	Design Report	10	Report clearly demonstrates strong traceability between requirements and design; design assumptions, constraints, key design are logically justified	Traceability is mostly correct with minor gaps; assumptions, constraints, and design decisions are present but lack depth or clarity	Basic traceability attempted but incomplete; assumptions or design decisions are weakly stated	Traceability is unclear or inconsistent; assumptions and design decisions are vague,	Missing report or report is largely irrelevant or unusable
	GitHub Repository Organization	5	Repository well-structured with all required folders and clear organization	Mostly organized with minor issues	Repository present but lacks clarity	Disorganized or incomplete	No repository

***All marks will be scaled based on Viva/ Presentation**