UNIVERSITY OF GLOBAL VILLAGE

Software Engineering & Project Management

COURSE PLAN

5th Semester

Department of CSE

Instructor

Galib Jaman

Lab Instructor

Department of CSE

Software Engineering and Project Management

Course Learning Outcomes (CLO)

1. **Analyze** a software problem to elicit and **formulate** comprehensive software requirements, documenting them in a structured Software Requirement and Design Specification (SRDS) document.
2. **Design** a high-level software architecture and database schema using standard modeling techniques like UML diagrams such as Use Case, Activity, Class and Entity-Relationship Diagrams (ERDs).
3. **Apply** modern software development tools and practices, including version control systems (Git/GitHub), for collaborative project implementation and management.
4. **Demonstrate** effective project management and teamwork skills by planning, executing, and delivering a software project within a specified timeline.
5. **Evaluate** a completed software project against its initial requirements and communicate the project's design, implementation, and outcomes effectively through written documentation and an oral demonstration.

CLO MAPPING OF

17-Week Course Plan

WEEK	TOPIC	ASSESSMENT	CLO
01	**Course Introduction & Project Ideation** Introduction to the course, CLOs, SDLC, and the importance of documentation. Overview of the SRDS template.	Class Wo... ▾	01 & 04
02	**Introduction & Scope Definition** Writing effective problem statements, defining system purpose, and articulating project importance.	Class Wo... ▾	01 & 04
03	**System Overview** Identifying key system features and defining primary/secondary users and stakeholders.	Class Wo... ▾	01 & 04
04	**Functional Requirements** Principles of requirements engineering. How to write clear, concise, and testable functional requirements.	Class Wo... ▾	01 & 04
05	**Non-Functional Requirements & Environment** Understanding NFR categories and Defining the system environment both for hardware and software.	Class Wo... ▾	01 & 04
06	**System Design & Components** Introduction to software architecture. Decomposing a system into logical modules	Class Wo... ▾	02 & 04
07	**Use Case Diagrams** Introduction to UML. In-depth tutorial on creating Use Case diagrams to model user-system interactions.	Class Wo... ▾	02 & 04

08	**Activity/Flow Diagrams** Modeling workflows and processes using Activity diagrams or traditional flowcharts.	Class Wo… ▾	02
09	**Class & Database Diagrams** Modeling the static structure with Class Diagrams. Introduction to database design and ERDs.	Class Wo… ▾	02
10	**Algorithm Descriptions & Final Review** Writing pseudocode or structured English for critical algorithms.	Class Wo… ▾	02
11	**Finalizing and Submitting the SRDS** Document formatting, referencing, proofreading, and avoiding plagiarism.	Class Wo… ▾	07 & 04
12	**Introduction to Git and GitHub** Version Control concepts. Core Git commands. Branching and merging strategies.	Practice ▾	05 & 04
13	**Implementing Core Functionality** Agile development mindset. Setting up the development environment.	Group W… ▾	05 & 04
14	**Feature Implementation** Connecting frontend to backend, API usage (if any).	Group W… ▾	05 & 04
15	**Integration, Testing, and Bug Fixing** Basic testing concepts such as unit and integration testing. Code refactoring and stabilization.	Group W… ▾	05 & 04
16	**Final Project Demonstrations** Professional presentation skills.	Group W… ▾	05 & 04
17	**Final Submission and Course Wrap-up** Code documentation and project hand-off.	Assignm… ▾	05 & 04