XIAMEN UNIVERSITY MALAYSIA
厦門大學 馬來西亞分校

DESCRIPTION OF COURSEWORK

Course Code	CST 307
Course Name	Software Architecture and Development Environment
Lecturer	Chua Chong Chai
Academic Session	2025/09
Assessment Title	Assignment

A. Introduction/ Situation/ Background Information

This assignment is to evaluate the ability of the students to perform analysis and design based on object-oriented modeling approaches. Throughout the process of working on different types of UML Diagrams, the students can experience through software development process (focusing on requirement, analysis and design) and at the same time practicing software modeling techniques based on real case study.

B. Course Learning Outcomes (CLO) covered

At the end of this assessment, students are able to:

CLO 1   Demonstrate the knowledge and understanding of software engineering processes, including tools and best practices, as well as key practices specified by CMM.

CLO 2   Apply software engineering practice guided by practical skills based on UML diagrams.

CLO 3   Perform object-oriented software design and architecturing by utilising UML based software design and modeling tools.

CLO 4   Perform component-based software development based on various frameworks.

CLO 5   Illustrate various aspects of object-oriented analysis and design processes in software system development.

C. University Policy on Academic Misconduct

1. Academic misconduct is a serious offense in Xiamen University Malaysia. It can be defined as any of the following:

i. **Plagiarism** is submitting or presenting someone else's work, words, ideas, data or information as your own intentionally or unintentionally. This includes incorporating published and unpublished material, whether in manuscript, printed or electronic form into your work without acknowledging the source (the person and the work).

ii. **Collusion** is two or more people collaborating on a piece of work (in part or whole) which is intended to be wholly individual and passed it off as own individual work.

iii. **Cheating** is an act of dishonesty or fraud in order to gain an unfair advantage in an assessment. This includes using or attempting to use, or assisting another to use materials that are prohibited or inappropriate, commissioning work from a third party, falsifying data, or breaching any examination rules.

2. All assessments submitted must be the student's own work, without any materials generated by AI tools, including direct copying and pasting of text or paraphrasing. Any form of academic misconduct, including using prohibited materials or inappropriate assistance, is a serious offense and will result in a zero mark for the entire assessment or part of it. If there is more than one guilty party, such as in case of collusion, all parties involved will receive the same penalty.

D. Instruction to Students

You are required to design and draw different types of UML diagrams based on the topic assigned to you. The deadline of this assignment submission is on **19 December 2025 (Friday), by 5 pm**. You need to submit this assignment online through Moodle. Late submission will result in penalty.

E. Evaluation Breakdown

No.	Component Title	Percentage (%)
1.	Problems Analysis	10
2.	UML Diagrams	20
	TOTAL	**30**

F. Task(s)

1. Perform problems analysis based on the previous proposed topic.

2. Describe the analysis of the problems in proper structure.

3. Plan, prepare and draw different types of UML Diagrams to describe, present and discuss various aspects of the system analysis and design.

4. Organize 2 and 3 into a complete assignment report:

   a. Problems Analysis (from part 2), discuss the following aspects:

      - Description of the problems,

      - The users/clients/stakeholders,

      - The functionalities or services provided by the system,

      - The non-functional requirements for your system.

   b. Use Case Diagram

      - Each member in the group will need to create at minimum ONE (1) Use Case Diagram. (Specify the name of the member who contributes to each of the diagrams.)

   c. Activity Diagram

      - Expand Use Cases in 4.b into Activity Diagram.

      - Describe the business processes involve in the system using Activity Diagrams.

      - Each member in the group will need to create at minimum TWO (2) Activity Diagrams. (Specify the name of the member who contributes to each of the diagrams.)

   d. Class Diagram + Package Diagram

      - Level 1

         o Identify the objects (physical/tangible or concept/intangible) in this system and create Domain Model (with one or more class diagrams).

         o Expand the Domain Model with functionalities (in Sequence Diagram) and update into Design Model.

         o Present the final Design Model (complete Class Diagrams) in the report.

- Level 2 – Organise classes into grouping of packages and create Class Diagram with packages (Chapter 6.1).
- Level 3 – Use Architecture Pattern such as MVC or Layered Pattern to organise packages from Level 2 Class Diagram into a Package Diagram (Chapter 6.1).
- Each group will need to create at minimum ONE (1) Level 1, ONE (1) Level 2 and ONE (1) Level 3 Class Diagrams.

e. Sequence Diagram (Chapter 4.4)

- Based on the information 4.d, create Sequence Diagrams to describe the interaction between some of the objects identified.
- Each member in the group will need to create at minimum TWO (2) Sequence Diagrams. (Specify the name of the member who contributes to each of the diagrams.)

f. Collaboration Diagram (Chapter 4.4)

- Create Collaboration Diagrams based on the Sequence Diagrams in 4.e.
- Each member in the group will need to create at minimum TWO (2) Collaboration Diagrams. (Specify the name of the member who contributes to each of the diagrams.)

g. State Machine Diagram (Chapter 4.5)

- Create State Machine Diagrams to describe the states of the classes in your Class Diagram.
- Each member in the group will need to create at minimum ONE (1) State Machine Diagram. (Specify the name of the member who contributes to each of the diagrams.)

h. Component + Deployment Diagram (Chapter 7)

- Propose a deployment strategy for your software system, from here create a Deployment Diagram.
- Each group will need to create at minimum ONE (1) Deployment Diagram.

MARKING RUBRICS

Component Title						Percentage (%)	
Criteria	**Score and Descriptors**					**Weight (%)**	**Marks**
	Excellent (9-10)	**Good (7-8)**	**Average (5-6)**	**Need Improvement (3-4)**	**Poor (0-2)**		
Problems Analysis (CLO1, 5)	Provide accurate and great details of the problems, with high clarity and readability. With creative ideas. Ideas are properly structured and well presented. All requirements specified in the tasks 4.a. are fulfilled.	Provide accurate and sufficient details of the problems. Ideas are properly structured. With minimal issues with the overall description. Fulfilling 80~90% of the requirements in tasks 4.a.	Provide accurate materials, however with insufficient details. Fulfilling 60~79% of the requirements in tasks 4.a.	Contain some correct and relevant material, some issues are neglected or are covered incorrectly. Fulfilling 41~59% of the requirements in tasks 4.a.	Very little or nothing that is correct and relevant. Fulfilling less than 40% of the requirements in tasks 4.a.	10	

Criteria	**Score and Descriptors**					**Weight (%)**	**Marks**
	Excellent (14-15)	**Good (11-13)**	**Average (7-10)**	**Need Improvement (4-6)**	**Poor (0-3)**		
UML Diagrams (Content) (CLO2,3,4)	Diagrams are accurate and with great details. Demonstrate creativity in the analysis and design. All requirements specified in tasks 4.b. to 4.i. are fulfilled.	Diagrams are accurate and with sufficient details. Fulfilling 80~90% of the requirements in tasks 4.b. to 4.i.	Diagrams are accurate, however the content in the diagrams is insufficient or incomplete. Fulfilling 60~79% of the requirements in tasks 4.b. to 4.i.	Contain some correct and relevant material, some issues are neglected or are covered incorrectly. Fulfilling 41~59% of the requirements in tasks 4.b. to 4.i.	Very little or nothing that is correct and relevant. Fulfilling less than 40% of the requirements in tasks 4.b. to 4.i.	15	

Criteria	**Score and Descriptors**				**Weight (%)**	**Marks**
	Excellent (4-5)	**Good (3)**	**Average (2)**	**Poor (0-1)**		
UML Diagrams (Quality) (CLO2,3,4)	Diagrams are consistent, well organized, readable, no obvious errors in the UML diagrams.	Diagrams are properly organized, minimal errors in the UML diagrams.	Diagrams are properly organized, but with obvious and repeating errors in the diagrams.	UML diagrams are inconsistent, contradicting, with major errors in the diagrams.	5	
TOTAL					30	

Note to students: Please include the marking rubric when submitting your coursework.