# Systems Design and Databases (CIS1018-N) UML Tutorial 3: Sequence and Class Diagrams

## Before You Start

**Hint:** It is important before starting the lab, you should walk-through the lecture(s) and demonstration exercises

Finish the Use Case Diagrams tutorial before attempting this one. Try to complete this set of tasks before your next tutorial.

## Introduction

In this week’s lecture, you have been introduced to sequence and class diagrams. This session aims to provide you with a deeper understanding of the topics using a series of exercises. The background story we are using in this lab is the same as the one we used in our last lab. If you are not familiar with or forget the scenario setup, please go back to last week’s lab material. You can also find useful information and guides from the lecture slides.

Note that, this lab is individual work; but you are encouraged to discuss with your groupmates and your lab tutors while you are doing some of the following exercises.

## Exercise 1: Sequence Diagrams

Recall the teamwork support system we have studied last week. In this exercise, we are going to devise sequence diagrams for the “Check Contribution” use case. To help you get started, we provided the following sequence diagram for a simplified version of the story, that is, once the Instructor clicks the “Check Contribution” button on the homepage, a query will be submitted to the database asking for the student's contribution scores; when the data is retrieved, it will be presented to the Instructor.

Diagram

Description automatically generated

Figure 1 A Simple Sequence Diagram

1. Understanding Sequence Diagrams

First try to answer the following questions:

* Who is the “Actor” in Figure 1? What are the other objects?

The instructor

* What are the vertical dashed lines?

The “lifelines” (see lecture slides for more infomation)

* Why there are two different types of the arrow?

The arrows with solid line indicate dispatching of messages while dashed arrows are returned messages

Then reproduce the diagram in diagrams.net.

1. Adding Authentication

Recall that for security reasons, we ask the instructors to use an admin password to authenticate his/her identity before they can use the “Check Contribution” function. Add this feature to your sequence diagram.

Hint: use the “loop” fragment. Within the loop, the Instructor can submit a password to the homepage; the homepage will check the password (by comparing it with the one stored in the database); if the password is correct, the loop ends and the Instructor can continue with other activities, otherwise, the Instructor is asked to retry (note that, this is an overly simplified process for practice purposes only, and in reality, you should never use a design like this).

Solution: see below

1. Adding Low Contribution Score Alter

Recall that we want the system to show an alter message containing the names of the students whose contribution scores are too low if there is any such student. Add this feature to your sequence diagram.

Hint: use the “alt” or the “opt” fragment. The idea is after the system retrieves the students’ contribution scores, and if it finds out there are some scores lower than the threshold, an alert message will be displayed.

Solution:

Diagram

Description automatically generated

## Exercise 2: Class Diagrams

1. Understanding Class Diagrams

Now we move on the class diagrams. Below, we show the Instructor class and the Team class.

Diagram

Description automatically generated

First try to answer the following questions:

* What are the attributes?

Instructor: name, email address; Team: ID, leader, supervisor, members

* What are the operations?

Check contribution, participate discussion, access assignment

* What does the line in the middle mean?

It’s an association, illustrating the relationship between the two classes, that is, an instructor supervises a team. It also indicates that a team must have one instructor while an instructor may supervise 0 to many teams.

Then reproduce the diagrams in diagrams.net.

1. Adding Other Classes

You may have noticed that the team leader and team member classes are still missing. Try to add them to your class diagram.

Also try to choose the appropriate associations to illustrate the relations between different classes.

Solution: see below

1. Using Inheritance

Note that both team leader and team member are students. Try to present this idea by introducing a new Student class, which is the superclass for both team leader and team member.

Try to add aggregation and/or composition to your class diagram.

Solution:

Diagram

Description automatically generated

## Extra Activity: ICA

From your previous labs, you should have the requirement list and use case diagrams worked out for your ICA project. Keep working on your ICA with the sequence and class diagrams related parts.