# Systems Design and Databases (CIS1018-N) UML Tutorial 2: Use Case Diagrams

## Before You Start

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

Attempt to complete this set of tasks before your next tutorial.

## Tutorial Overview

This lab session focuses on the use case diagrams and wireframe with a series of exercises.

* Exercise 1: Learn by Example
* Exercise 2: Advanced Use Case Diagrams
* Exercise 3: Reproduce the Wireframe

## Introduction

In this session, you will first learn the web-based diagram tool (<diagrams.net>). This tool is free to use and there is no compulsory registration needed. This is the tool we recommend you use for UML-related tasks in this module.

This session also aims to familiarise you with the use case diagrams with a series of exercises. Everything you need to know about use case diagrams is covered in the lecture. Feel free to go back to the slides if you would like to refresh your memory, or you may seek help from your lab tutor if you encounter any problems.

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

## Preparation

**Get Start with Diagrams.net:**

* Use the link to access <https://app.diagrams.net/>
* You are likely to see the following prompt window asking you where you would like to save your work:

Graphical user interface, application

Description automatically generated

* Feel free to choose any of your favourite cloud storage, or you can save your work locally on your hard drive by using the “Device” option.
* Alternatively, you can decide later when you finish your work: when you’re done, go to “File -> Save”.
* When you are working on your UML diagrams, it is recommended to use the templates provided (see the screenshot attached below), which will give your diagrams a more professional looking:

Graphical user interface, application

Description automatically generated

* Now you’re all set. Feel free to explore the functions provided from diagrams.net before you move onto the next lab task. Most importantly, test the save and open functions to make sure you can save your work and continue working on it later.

## Exercise 1: Learn by Example

The following example is about an online teamwork support system that helps the students (including team leaders and team members) and their instructors to manage the teamwork progress. The requirement list (formalised using the MoSCoW method) is given in Table 1 and the corresponding use case diagram is given in Figure 1.

\*Note that for simplicity, we assume the system does not require login before using.

|  |  |  |
| --- | --- | --- |
| **ID** | **Details** | **Priority** |
| R1 | Team leader shall be able to create a team. | MustHave |
| R2 | Team leader shall distribute workload. | MustHave |
| R3 | Everyone shall be able to participate discussion of the teamwork. | MustHave |
| R4 | Everyone shall be able to access the work file, aka the Assignment. | MustHave |
| R5 | Instructor shall be able to check students’ contribution scores. | MustHave |

Table 1 Teamwork Support System Requirement List

Diagram

Description automatically generated

Figure 1 Teamwork Support System Use Case Diagram

Finish the following tasks:

1. Understand the use case diagram given in Figure 1 by asking yourself the following questions and discussing with your groupmates or your lab tutors:
   1. How many actors are there? Who are they?
   2. How many use cases are there? What are they?
   3. What are the lines between the actors and the use cases? What do they mean?
2. Reproduce the use case diagram in <diagrams.net>

## Exercise 2: Advanced Use Case Diagrams

Based on Exercise 1, improve your use case diagram by introducing the following advanced features.

1. Generalisation

Note that both “Team Leader” and “Team Members” are students. Try to use actor generalisation to illustrate the idea that: “all students shall be able to participate discussion and access assignment, but only the team leaders can create a team and distribute the workload”.

1. Using <<include>>

To make the system more secure, now we require the instructor to authenticate his/her identity using an admin password before he/she can check the students’ contribution scores. This idea is presented in the updated requirement list and the following use case specifications. Add this feature to your use case diagram.

|  |  |  |
| --- | --- | --- |
| **ID** | **Details** | **Priority** |
| … | … | … |
| R6 | Authentication shall take place before privileged content can be accessed | MustHave |

Graphical user interface, application

Description automatically generated

1. Using <<extend>>

Some instructors suggest having another feature to help them pin down the students in the need of help. That is, if there are some students’ contribution score lower than a threshold, an alter message containing those students’ names should pop up when they check the contribution scores. This new requirement is added to the requirement list and the use case specifications are updated accordingly. Add this feature to your use case diagram.

|  |  |  |
| --- | --- | --- |
| **ID** | **Details** | **Priority** |
| … | … | … |
| R7 | An alter message shall be displayed when some student’s contribution score is too low | ShouldHave |

Graphical user interface, application

Description automatically generated

## Exercise 3: Reproduce the Wireframe

In this exercise you have to reproduce the following NHS Family Voice by using the [Diagrams.net](http://diagrams.net/)

## Extra Activity: ICA

From your last week’s lab, you have had the requirement list for your ICA project. Generate use case diagrams based on the requirements using the knowledge and skills you gain this week.