

Module Name: **Software Development and Application Modelling**

Module Code: **COMP40003**

Title of Assignment: **Assignment 2**

Module Learning Outcomes for This Assignment

3. Demonstrate, apply and document to the appropriate standards the key techniques of business analysis and application modelling.	Analysis Enquiry
4. Implement Object-Oriented application models in a suitable programming language.	Application

Submission deadline

Submit by: 22 September 2023 by 05.00 pm (via LMS)

Demonstration dates

Will be informed via LMS

Part 1 – Group work

This part of the assignment must be done in groups of 3-4 students.

Scenario: Training courses system

A company wants a computer system for tracking employee attendance at in-house training courses. Administrators can add employees, add training courses, register an employee on a training course, record employee's completion of a training course, view all training courses, and view an employee's training courses.

The computer system is to provide the following use cases:

Use case	Notes
Add employee	If the employee record does not already exist, create the employee record. The employee details are employee number, name, address, date of birth.
Add training course	Create a training course record. The training course details are course id, title, date of course, maximum number of places available.
Register employee for a course	If the employee and training course exist, and if there are places still available on the course, add the employee to the course.
Record employee's completion of a course	If the employee and training course exist, and if the employee is registered on the course, record employee's completion of the course.
View all training courses	Display the details of all training courses, including all employees registered for each course (employee number, name and whether completed or not).
View employee's training courses	Display the employee's details, including a list of all courses on which the employee is registered. For each course, show whether the employee completed it or not.

Your task

NB: This part of the assignment must be modelled as an Object-Oriented system.

As a group, follow the Unified Software Development Process (USDP) to develop an object-oriented model (using UML) for the computerised system described above. You must use Microsoft Visio, Visual

COMP40003: Software Development and Application Modelling

Paradigm or any other suitable tool for your models.

Each person in the group designs one use case using all the following diagrams. The diagrams should be developed in the order listed:

- Use case diagram (one for the entire application)
- Activity diagrams (one per use case)
- A class diagram with relationships between the classes (one for the entire application)
- Sequence diagrams (one per use case)

Part 2 – Individual work

This part of the assignment must be done on your own. You may only ask your SDAM tutor for help.

Scenario: Stock management system

A company wants a computer system for recording stock in its warehouse. Administrators record the addition of new stock, update the quantity of existing stock items when a delivery is received, update the quantity of stock items when someone takes one or more from the warehouse, delete an item of stock, view stock transactions, and view stock levels reports.

The computer system is to provide the following use cases:

Operation	Information
Add a new item of stock	If the stock item does not already exist, create the stock item record and make an entry in the transaction log. The stock item details to be recorded are stock code, name, quantity in stock. The transaction log entry details are date/time of the transaction, "Item added", stock code, stock item name, quantity added.
Add quantity to a stock item	If the stock item exists, add the new quantity to the stock item's quantity and make an entry in the transaction log. The transaction log entry details are date/time of the transaction, "Quantity added", stock code, stock item name, quantity added, new quantity in stock.
Remove quantity from a stock item	If the stock item exists and there is enough quantity in stock, subtract the required quantity from the stock item's quantity and make an entry in the transaction log. The transaction log entry details are date/time of the transaction, "Quantity removed", stock code, stock item name, quantity removed, new quantity in stock.
Delete a stock item	If the stock item exists and the quantity in stock is zero, delete the stock item record and make an entry in the transaction log. The transaction log entry details are date/time of the transaction, "Item deleted", stock code, stock item name.
View transaction log	Display all transaction log entries.
View stock levels	Display all existing stock item records.

Dates and times are to be displayed as dd/MM/yyyy HH:mm using 24-hour clock.

Your task

Use Case

Draw the UML use case diagram for the above use cases.

Implementation

This part of the assignment must be written as an Object-Oriented application in C#. You may use any appropriate software to design the diagrams.

Select only 'TWO' use cases,

1. Design the activity diagram
2. Design the class diagram
3. Design the sequence diagram
4. Implement the C# application

Test

Write a test plan (sample provided on LMS) and use it to test your application recording the actual results.

Submission

Submit your assignment using the following links on Blackboard:

- **"Assignment 2, Part 1 group submission"** for:
 - o Your group's UML model as one PDF document
- **"Assignment 2, Part 2 individual submission"** for:
 - o A ZIP file containing your visual studio project folder (clean the solution before compressing it)
 - o The test plan with actual results as a PDF document

Mark Scheme

You should refer frequently to the marking scheme during your work on this assignment.

You are reminded of the university's policy about plagiarism, as described at:

<https://www.staffs.ac.uk/legal/policies/academic-conduct-procedure.jsp>

Marks for this assignment will be allocated as follows:

Part 1 (30%)

Your model will be assessed in your group's presentation using Rubric 1 (see Appendix A), which will give you quantitative and qualitative feedback on your work. You should ensure that your work presents evidence to support the awarding of marks in each criterion of the rubric.

Part 2 (70%)

Your software and test plan will be assessed in your demonstration using Rubric 2 (see Appendix A), which will give you quantitative and qualitative feedback on your work. You should ensure that your work presents evidence to support the awarding of marks in each criterion of the rubric.

After submitting your work, you will be required to demonstrate your software to your module lecturer. You will run your software to prove that it works, and you will show how closely your implementation adheres to your model (so take a copy of your model to the demonstration). You will need to answer questions about your code correctly and confidently. Incorrect or unconfident answers might result in your marks being reduced. A code check activity may also be used to assess your knowledge in object oriented programming.

If you do not attend your scheduled demonstration slot, you will score zero marks for Part 2 of the assignment.

Even if your code is not functioning, you should still book and attend a demonstration so that you can discuss your attempt with the lecturer and receive feedback that may be helpful should you be referred.

Instructions for the demonstration

The demonstration timeslots will be published on LMS after the submission deadline.

To prevent delays in the demonstration, it is essential that you be present ten minutes before the allocated time for your appointment and do the following before your demonstration start-time:

1. Open the file containing your model
2. Open Visual Studio, and prepare your application

COMP40003: Software Development and Application Modelling

3. Be ready to show the tutor that the dates on your files are the same as those submitted via LMS.
4. Make sure that your system is in its initial state (i.e. ready to run without delay)
5. Await instructions from the lecturer who will assess your demonstration

Appendix A: Rubrics

Part 1 – Group work

	Marks										
Criterion (weighting)	0	15	25	35	45	55	65	75	85	95	100
Use case diagram (20%)	No diagrams	Another developer would not know which activity diagrams to prepare			Good use of notation; clear layout; another developer would know which activity diagrams to prepare with several supplementary questions			Presented to a professional standard; very minor mistakes in the use of the notation; another developer would know which activity diagrams to prepare with few supplementary questions			No mistakes in the use of the notation; another developer would know which activity diagrams to prepare without any supplementary questions
Activity diagrams (20%)	No activity diagrams	Another developer could not use the diagrams to derive the analysis diagrams			Good use of notation; clear layout; another developer could use the diagrams to derive the analysis diagrams with several supplementary questions			Presented to a professional standard; very minor mistakes in the use of the notation; another developer could use the diagrams to derive the analysis diagrams with few supplementary questions			No mistakes in the use of the notation; another developer could use the diagrams to derive the analysis diagrams without any supplementary questions
Class diagram (30%)	No class diagram	Many significant errors; Entirely inadequate for writing the program code			Good use of notation; another developer could use the diagram to write the program code with several supplementary questions; does not use inheritance for the transactions			Presented to a professional standard; no mistakes in the use of the notation; another developer could use the diagram to write the program code with few supplementary questions; uses inheritance for the transactions			No mistakes in the use of the notation; another developer could use the diagram to write the program code without any supplementary questions
Sequence diagrams (30%)	No sequence diagrams	Many significant errors; Entirely inadequate for writing the program code			Good use of notation; another developer could use the diagrams to write the program code with several supplementary questions			Presented to a professional standard; no mistakes in the use of the notation; another developer could use the diagram to write the program code with few supplementary questions			No mistakes in the use of the notation; another developer could use the diagrams to write the program code without any supplementary questions

COMP40003: Software Development and Application Modelling

Part 2 – Individual work

	Marks										
Criterion (weighting)	0	15	25	35	45	55	65	75	85	95	100
Activity diagrams (10%)	No activity diagrams	Another developer could not use the diagrams to derive the analysis diagrams			Good use of notation; clear layout; another developer could use the diagrams to derive the analysis diagrams with several supplementary questions			Presented to a professional standard; very minor mistakes in the use of the notation; another developer could use the diagrams to derive the analysis diagrams with few supplementary questions			
Class diagram (10%)	No class diagram	Many significant errors; Entirely inadequate for writing the program code			Good use of notation; another developer could use the diagram to write the program code with several supplementary questions; does not use inheritance for the transactions			Presented to a professional standard; no mistakes in the use of the notation; another developer could use the diagram to write the program code with few supplementary questions; uses inheritance for the transactions			
Sequence diagrams (10%)	No sequence diagrams	Many significant errors; Entirely inadequate for writing the program code			Good use of notation; another developer could use the diagrams to write the program code with several supplementary questions			Presented to a professional standard; no mistakes in the use of the notation; another developer could use the diagram to write the program code with few supplementary questions			
Test cases and unit testing (10%)	No test plan	Little evidence that the test plan has been followed			Good evidence that the test plan has mostly been followed			The test plan was followed completely and professionally			
User interfaces (for the two use cases selected) (10%)	Application not implemented or does not run.	Attempted, but has major problems			All use cases can be executed, but feedback to the user and presentation of information is poor			All use cases can be executed. Feedback to the user is very good and the presentation of information is clearly laid out and easily readable			
Implementation (for the two use cases selected) (30%)	Application not implemented or does not run.	Attempted, but has major problems			All use cases can be executed, the output has errors or does not display the correct results.			All use cases can be executed. No errors, all information presented clearly and the application is user friendly.			

COMP40003: Software Development and Application Modelling

Demonstration and viva (20%)	Not present	Many significant errors; Does not understand the concepts or the code implemented	Few errors; Can understand the concepts and explain the work submitted to some extent.	Presented to a professional standard; no mistakes in the explanations given. Clear justifications provided with confidence.
------------------------------	-------------	--	---	---