Queen Mary University of London

ECS505U Software Engineering Coursework II

23/11/2025 v2.3

This coursework makes up 18% of the total marks for the module. Answer all the questions to achieve a maximum score of 100 (Q1 has 55 marks and Q2 has 45 marks).

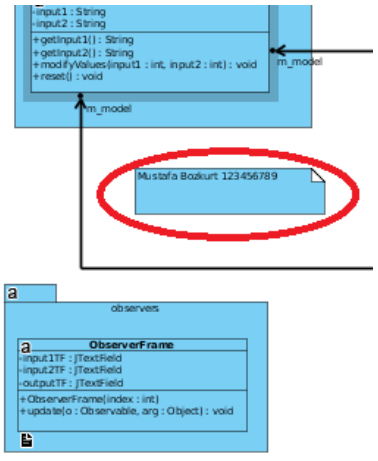
Submit your solution using the submission link on QMplus course webpage by the submission deadline. You should be able to see the details of the coursework submission on QMplus course page under assessed coursework section.

Please submit one PDF file for all questions. It is your responsibility to make sure the images are readable. We will not mark diagrams that are not readable. You will be penalized harshly if you do not submit your coursework with the required format: a two-page pdf file with each question on a separate page. If your Q2 answer is more than one page, then you can submit a three-page (max) file.

This coursework should take you about 3-4 hours to complete. **You are expected to complete it without collaborating with other students.**

It is recommended that you use Visual Paradigm UML tool to create diagrams as marks will be deducted if the tool you choose uses a non-standard UML notation.

Please add a note with your name and student ID on every diagram as depicted below to avoid any marking issues as we mark each diagram separately.



Overview

You are given the task of designing a software that handles day to day activities of a financial institution. Beware, the following requirements are a subset of an actual system. The requirements in this document are selected with the aim of enabling you to demonstrate your system design ability using UML. Thus, some of the requirements were omitted. However, you are not expected to improve the requirements of the software. Please, do not make assumptions and follow only the instructions.

If you have any questions regarding requirements, please ask it through the QMplus forum. I might not respond to questions via email.

Glossary:

The system: The software system designed to help staff with their day-to-day activities.
System User: A person who is using the system. Includes all types of users described below.
Staff: Employees who deals with day-to-day activities of the institution.
Customer: A person or a business who holds an account with the institution.
Personal customer (or person): A customer with a personal account.
Business customer (or business): A customer with a business account.

Question 1 (55 marks)

In this question, you are expected to draw a sequence diagram depicting the interactions of the following objects in removing a bank card from a business account. The following objects already available in the system for this scenario: user interface (GUI object), staff (Staff object), registry (Registry object), account (BusinessAccount object), card (BusinessCard object)

1. A staff makes a **search** for an **account** using GUI.
2. GUI calls the staff object for the search with the given number.
3. The Staff object calls registry to find if an account with the given number exists and assigns the return value to a local variable result and returns result to GUI.
4. If the result is null GUI **display**s an **error** message.
5. If the result is not null GUI **display**s **account** info to user.
6. If the account is found (result not null), the staff makes a request system to **get bank cards** for this account.
7. GUI gets the list of cards from the account.
8. To display cards GUI gets the name on card and number for all business cards from the list.
9. GUI **display cards** to the staff.
10. The staff **select**s **a card** to remove from the account using the GUI.
11. GUI calls the account to remove the selected card.
12. Account removes the card from the system.

HINTS:

- You need to download and use the coursework VPP file from QMplus. If you not going to use VP then download the class diagram image so you can use the correct method calls.
- Business accounts only have business cards.

Question 2 (45 marks)

Suggest a design pattern that will efficiently solve the given scenarios. Provide a UML solution depicting the application of the suggested design patter for each scenario and discuss why and how the suggested design pattern solves the given problem.

Each section is worth 15 marks: 8 marks for the correct solution and 7 marks for the design discussion.

a) According to the requirements (in CW1) the **registry** holds all the bank cards and accounts. The system must ensure that always one registry (object) exists in the system to manage all accounts and cards to avoid data inconsistencies.

b) The branch **staff** take on two **role**s: **cashier** and **customer representative**. The staff very frequently change roles as their role in the branch is based on the computer they log in from. When a staff logs in from computers at the counter desk, they are considered a cashier. However, when they log in from one of the office computers then they are considered a customer representative. The system should allow changing of **staff role**s efficiently and seamlessly.

c) Each bank card type (**businesscard** and **personalcard**) has common information that needs to be kept in the system. This information is **card type description** which is exactly the same for each type of card. Keeping this common information in the card classes causes the system to use more memory than needed. We want the system to run more efficiently, not use more memory than needed when dealing with bank cards.

HINTS:

- The entities you are expected to represent in your UML diagram are shown in **bold**.
- For the first design problem, you are expected to provide the necessary class members with their full signature.
- In the second design problem, although the explanation states that the staff change roles but in reality, what happens is that the staff (object) acts differently in different context (from the system's perspective). From design perspective these two are the same problem.
- You are expected to provide a UML solution with the necessary classes for each section. Please **do not** provide a full system design with your solution.
- The classes needed to apply the necessary design patterns might not be a part of CW1 requirements. You need to add these classes to your solution.
- For your solutions include a discussion explaining why and how your proposed solution solves the given design problem.

END OF CW INSTRUCTIONS