

Assignment 2

Handout: **Wednesday, 8 November 2023**
Due: **23:59, Saturday, 18 November 2023**

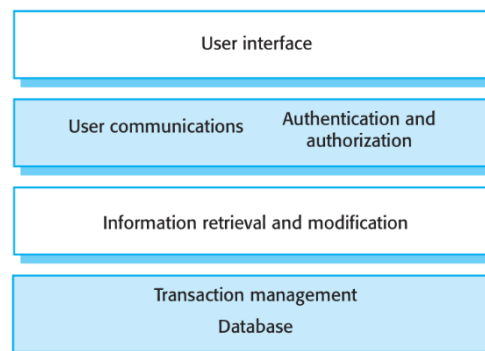
1. Class Diagram (13 marks)

Please read the following description, design the classes and their relations for the system, and draw the corresponding UML class diagram. Note that you need to choose the appropriate types of associations, but you do NOT need to specify the multiplicities for the associations. Your design should support all the requirements mentioned in the text.

An order management system keeps track of various products in stock and orders. For each product, its name, price, and quantity in stock are recorded. Both the price and quantity of a product may be updated. The system has two types of customers, namely companies and persons, and for each customer, the system records its phone number and address. Besides, the first and last names of persons and the government registration numbers of companies are recorded. All customers can place orders. The information associated with an order includes the customer, date, and order items. An order item is for a specific product, and it also specifies the quantity of the product as well as the price of the product at the time the order item was created. When creating an order item, an order to which the new order item belongs must be specified, and the newly created order item will be added to the order automatically. Order items may not be removed from an order.

2. Layered Architecture (15 marks)

A utility company wants to develop an information system to maintain information about assets like buildings, vehicles, and equipment owned by the company. It is intended that the information will be updatable by staff working in the field using mobile devices as new asset information becomes available. The company has several existing asset databases that should be integrated through this system.



Given the system's layered architecture as shown above and the following system modules (a) through (j), please decide which modules should be placed in which layers.

- | | | |
|------------------------------|----------------------|-------------------------------|
| (a) mobile device management | (b) database search | (c) buildings database |
| (d) equipment database | (e) database browser | (f) database query management |
| (g) forms management | (h) browser UI | (i) mobile UI |
| (j) vehicle database | | |

3. Software Testing (12 marks)

Suppose a developer has written a method named `canFormTriangle`, as shown on the next page, that a) takes three `double` parameters named `x`, `y`, and `z` as the input, b) treats the input values as denoting the lengths of three line segments, and 3) returns a `boolean` value indicating whether the three line segments can legitimately form a triangle. Note that three line segments of lengths `x`, `y`, and `z` can legitimately form a triangle if and only if they satisfy the following two conditions: a) `x`, `y`, and `z` are all greater than 0; b) the sum of any two of them is greater than the third.

```
1 boolean canFormTriangle(double x, double y, double z){
2     if(x <= 0 || y <= 0 || z <= 0)
3         return false;
4
5     if(x + y <= z)
6         return false;
7     else if(x + z <= y)
8         return false;
9     else if(y + z <= x)
10        return false;
11    else
12        return true;
13 }
```

Please apply partition testing to devise a set T of unit tests for method `canFormTriangle` such that, for every possible unit test t of the method, there is a unit test t' in T satisfying that t and t' have the same branch coverage on the method. Here, two tests t_1 and t_2 are said to have the same branch coverage on a method if and only if both conditions 1) and 2) are satisfied: 1) t_1 and t_2 exercise the same set of `if` statements in the method; 2) for each `if` statement exercised by both t_1 and t_2 , the corresponding `if` condition expression evaluates to the same value during t_1 and t_2 's executions. For example, if both tests t_1 and t_2 exercise only the first `if` statement in method `canFormTriangle` and the condition expression on line 2 evaluates to `true` in both tests, t_1 and t_2 are said to have the same branch coverage on method `canFormTriangle`. Write down all your tests, with one test on each line.

For each test, you need to specify the values of parameters `x`, `y`, and `z`, the expected return result from the method, and the branches covered by the test. In the example below, `c2==true` indicates that only the `if` condition on line 2 is evaluated during the test execution and the condition evaluates to `true`. You may separate the branches using commas.

x	y	z	ExpectedResult	BranchesCovered
1	1	-1	false	c2==true

4. Software Maintenance (10 marks)

As introduced in lecture one, there are four main types of maintenance in the modern view of software maintenance. Briefly describe the four main types and explain why it is sometimes difficult to distinguish between them. (≤ 200 words)

How to hand in:

Submit your typed, instead of handwritten, answers in a PDF file on Blackboard.