Assignment 2

Release: **Tuesday, 5 November 2024**
Due: **14:00, Friday, 15 November 2024**

1. Class Diagram (15 marks)

Question 1. The Hong Kong Transport Department wants to build a software system for the management of bus lines and bus drivers. Please design the classes and their relations for the system and draw the corresponding UML class diagram in the white space below. A basic introduction to the system is given below, your design should support all the operations as mentioned in the introduction. Annotate the associations between classes appropriately.

> Bus lines are directly operated by bus companies, and they are never shared among companies. Each bus company has a name and an ID. Each bus line has an ID and covers a sequence of two or more bus stations. Each bus station has an ID. While a bus company may own different types of buses, only specific types of buses are allowed to serve each bus line. Each bus has a type and a capacity, i.e., the maximum number of passengers allowed, and it may be in state "working" or "under repair". Bus drivers must be employees of bus companies. Each driver has a name, an ID, and a driving license, which states the types of buses the driver is allowed to drive. Given a bus type, users of the system can find out whether the type of bus is allowed to serve on a particular bus line and whether a particular bus driver is allowed to drive that bus. A bus ride is a trip of a bus, driven by a driver, along a bus line. Each bus ride has a starting time and an ending time: the ending time should be ignored if the ride is not finished yet. Given a bus line, users of the system can find out all the bus rides on a particular date.

2. Architectural Design (12 marks)

A desktop application implements the layered architecture shown in Figure 1, where the presentation layer is concerned with presenting information to the user and managing all user interaction, the application processing layer is concerned with implementing the logic of the application and providing the required functionality to end-users, while the database layer stores the data and provides transaction management and query services.

Suppose a software company is tasked with transporting the application to a distributed environment, and the transported application should adopt the client-server architecture. The software architects in the company came up with two different designs shown in Figure 2 for the new application. In the first design, the presentation and application processing layers will run on the client side, while the database layer will run on the server side. In the second design, only the presentation layer will run on the client side, while the application processing and database layers will run on the server side. Give one advantage of each design (<200 words).

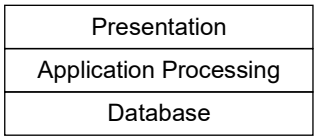

Figure 1. The layered architecture of a desktop application.

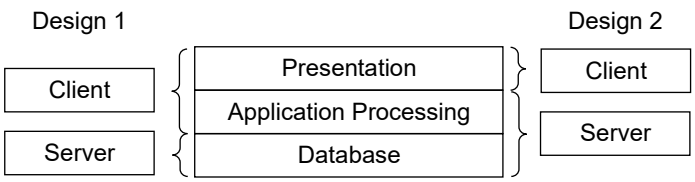

Figure 2. The client-server architecture of the distributed version of the application.

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 `t1` and `t2` are said to have the same branch coverage on a method if and only if both conditions 1) and 2) are satisfied: 1) `t1` and `t2` exercise the same set of `if` statements in the method; 2) for each `if` statement exercised by both `t1` and `t2`, the corresponding `if` condition expression evaluates to the same value during `t1` and `t2`'s executions. For example, if both tests `t1` and `t2` exercise only the first `if` statement in method `canFormTriangle` and the condition expression on line 2 evaluates to `true` in both tests, `t1` and `t2` 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 (11 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.