ASOS Software Engineering Code Test

This is a refactoring exercise based on a C# Visual Studio solution which currently has no unit tests and a number of maintainability problems.

# The problem

We would like you to begin refactoring the AddCustomer method in the CustomerService class in order to make the class easier to maintain.  You can change anything in the App project (method signatures, constructors, etc.), apart from making the CustomerDataAccess class and its methods non-static.

You should assume that this service forms part of a larger system. You must **not** change anything in the Harness project, which is an existing integration and backwards compatibility must be maintained. You do not need to add anything to or amend the Harness project in any way, but it must remain part of the solution and it must compile after you have completed the refactor.

During the refactoring process, you should consider the SOLID principles, the readability of the code and where tests might be appropriate. We are aware that the *logic* used to perform validation isn't perfect, but we do not expect you to fix that or make it more comprehensive.

By the end of the refactor, you **must** have at least one test that covers the successful addition of a customer, along with any other tests that you think are required to cover the logic.

You can use whichever test and/or mocking frameworks you wish. We’re not expecting the work to be ‘finished’, but you should have started the work and be able to discuss the changes you have made and what your next steps would be.  Make sure you attempt some refactoring and unit testing to demonstrate your skills in both.

# Your solution

You should aim to spend no more than two hours improving the solution to make it more maintainable, applying basic engineering principles such as SOLID, DRY, YAGNI and KISS. We want to see how you break down the problem and we’re looking for simple, clean, readable code to demonstrate this. Please try to avoid over-engineering or gold plating.

We don’t want lots of explanatory comments in the code, but if you run out of time and would still like to refactor something then feel free to add a comment in the relevant place. Likewise, you can comment with any other narrations that are relevant.

Feel free to use the internet to look up anything you need. Once you are done, please reply to the email you received containing this test, attaching a nice zipped solution that compiles and has passing unit tests.

All your unit tests **must** pass – a submission with failing tests will not pass the review stage.

# What will happen next?

A panel of ASOS engineers will review your submission and provide feedback to you. If we like your solution and would like to discuss it further, we’ll get in touch to make arrangements to meet face-to-face, where we will expect you to explain the changes you have made so far and what your next steps would be.