Homework

Instructions

Attached, please find a console application which main purpose is to convert the formats.

Prepare:

1. Please find at least 5 potential code issues an be able to explain the reason behind it.
2. Refactor the app to allow:
    a. Work with documents of various storages eg. filesystem, cloud storage or HTTP (HTML read-only) etc. Implement just one of them but be sure that implementation is versatile for adding other storages.
    b. Be capable of reading/writing different formats. Implement XML and JSON format, but be sure that implementation is versatile for adding more formats (YAML, BSON, etc.).
    c. Build the app in the way to be able to test classes **in isolation**
    d. Be able to add new formats and storages in the future so it will have **none** or **minimal impact** on the existing code
    e. Be able to use any combination of input/output storages and formats (eg. read JSON from filesystem, convert to XML and upload to cloud storage)

We're going to appraise the design of given code that should match the **quality of production application.** Thus imagine this application as a system ready for feature development (adding new storages or formats).

Tests should be written as demonstration of your skills, there is no need to cover everything.

Please version progress as usual, upload your homework to GitHub or other preferred git storage and send us link. Alternatively, use git archive and send us a ZIP archive.

Source file

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Xml.Linq;
using Newtonsoft.Json;

namespace Moravia.Homework
{
    public class Document
    {
        public string Title { get; set; }
        public string Text { get; set; }
    }

    class Program
    {
        static void Main(string[] args)
        {
            var sourceFileName = Path.Combine(Environment.CurrentDirectory, "..\\..\\..\\Source
Files\\Document1.xml");
            var targetFileName = Path.Combine(Environment.CurrentDirectory, "..\\..\\..\\Target
Files\\Document1.json");

            try
            {
                FileStream sourceStream = File.Open(sourceFileName, FileMode.Open);
                var reader = new StreamReader(sourceStream);
                string input = reader.ReadToEnd();
            }
            catch (Exception ex)
            {
                throw new Exception(ex.Message);
            }

            var xdoc = XDocument.Parse(input);
            var doc = new Document
            {
                Title = xdoc.Root.Element("title").Value,
                Text = xdoc.Root.Element("text").Value
            };

            var serializedDoc = JsonConvert.SerializeObject(doc);

            var targetStream = File.Open(targetFileName, FileMode.Create, FileAccess.Write);
            var sw = new StreamWriter(targetStream);
            sw.Write(serializedDoc);


        }
    }
}
```