You are applying for a C# developer. If you are interested then please fulfill the following task:

Develop a reusable library which will allow reading, modification and conversion of different file formats. The library should allow adding support for new user supplied file formats. Note that the file format specifies the layout of the data only, however all file formats should contain the same data. Each file format should contain several records which have the following look:

|  |  |
| --- | --- |
| **Field** | **Field format** |
| Date | DD.MM.YYYY format |
| Brand name | Unicode string |
| Price | Integer |

The library should allow editing of the specified record, adding new records and deleting the specified records. The other feature is format conversion.

The library should support 2 file formats which have the following look:

1. Xml file format, which has the following structure:

<?xml version="1.0" encoding="utf-8"?>

<Document>

<Car>

<Date>10.10.2008</Date>

<BrandName>Alpha Romeo Brera</BrandName>

<Price>37000</Price>

</Car>

</Document>

**Restrictions:** There may be 0 or more <Car> tags, the xml structure should be preserved.

1. Binary file format, which has the following structure (ordered):

|  |  |  |
| --- | --- | --- |
| **Field** | **Field size** | **Value** |
| Header | 2 bytes | 0x2526 |
| Records count | 4 bytes | Integer specifying number of records |
| Date | 8 bytes | DDMMYYYY |
| Brand Name length | 2 bytes | Length of the following string |
| Brand Name | 0..<Brand Name length \* 2> bytes | Unicode String |
| Price | 4 bytes | Integer |

**Restrictions:** There may be 0 or more records. <Records count>, <Price> and <Brand Name length> should have positive values, DDMMYYYY should contain valid values for DD and MM.

If some of the specified restrictions are violated then the corresponding exception should be thrown.

Additional requirements:

1. Provide the source code written in managed C# and code which uses the library.
2. You should provide architecture description in English with description of weak and strong sides and possible future library development.
3. The library should contain NUnit unit tests to validate its activities.
4. NAnt build script for building the library.