CSE 445 Assignment 4 XML (50 Points)

Introduction

The aim of this project is to make sure that you understand and are familiar with the concepts covered in the lectures, including XML elements, attributes, statements, XML schema, XML validation, XML transformation (XSL), and the related library classes. By the end of the project, you should have applied these concepts and techniques in creating an XML file, its schema, its style sheet, and have written programs to process XML and Json files.

This is an **individual assignment**. Each student must complete and submit independent work. No cooperation is allowed, even among the team members for project 3. Do not use WebStrar (which is shared among your team members) to host your files, applications, or services in project 4. You can use GitHub site to host your files, such as XML, XSD, and XSL files (see Practice Exercises question 4) and use localhost to host the Web applications and services.

Practice Exercises (No submission required)

No submission is required for this part of exercises. However, doing these exercises can help you better understand the concepts and thus help you in quizzes or exams.

1. Reading: Textbook Chapter 4.

2. Answer the multiple choice questions 1.1 through 1.16 of the Text Section 4.8. Study the material covered in these questions can help you to prepare for the class exercises, quizzes, and the exam.

3. Study for questions 2 through 8 in Text Section 4.8. Make sure that you understand these questions and can briefly answer these questions. Studying the material covered in these questions can help you to prepare for the exam and understand the assignments.

4. In this assignment, we will use a public server for hosting the files. There are different servers are available, for example, Azure cloud, AWS cloud, and Google cloud, In this assignment, we will require you to use GitHub server: https://pages.github.com/ to host your files, such as XML file and XSD file. Please follow the instruction at https://pages.github.com/ to complete the following exercises:

    1) Create an account and a repository site;

    2) Deploy your files into the site;

    3) Use a web browser to access your files;

    4) Write a simple client, e.g., a Console Application to access your files;

5) Write a Web services, a RESTful service or a WSDL service, to access the files.

5 You can access the sample XML and XSD files at:

http://venus.sod.asu.edu/WSRepository/xml/Courses.xml
http://venus.sod.asu.edu/WSRepository/xml/Courses.xsd

Assignment Questions (Submission Required: 50 points)

1 In this question, you will create a directory of hotels with required information related to each hotel. The information will be represented as an XML tree. The diagram below shows the required structure of the directory of Hotels in XML tree notation that you will create in this assignment. All the "Hotel" elements have the same structure. Notice that different shapes and colors of boxes in the tree have different meanings, as shown in Figure 1. They represent elements, attributes, and text contents/values, respectively. The structure of elements and attributes given in the diagram in Figure 1 must be implemented as described. The given text contents/values in the diagram are a sample that you must include in your XML file. The solid arrows show parent-child element relations, and the dotted arrows show the element-content or attribute-value relations. The optional attribute means that the XML instance can have the attribute or not have the attribute, without causing a verification error against its schema. However, it is still required to define the optional attribute in the XML Scheme file. In your instances, you must provide this attribute for some hotels, but not for all hotels. For the required attribute, you must provide this attribute for all such elements.

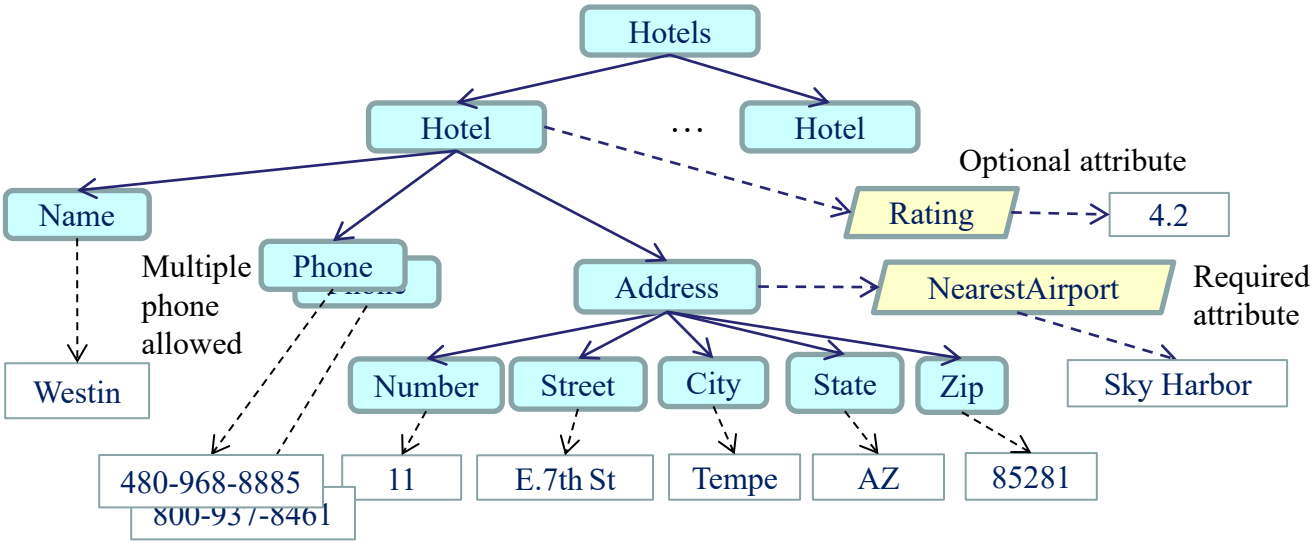

Figure 1. Required XML structure

1.1 Write the Hotels.xsd file that defines the XML schema allowing the structure shown in Figure 1. The elements at the same level should follow from left to right order. You can use any tool to create/edit the file. The file must be saved in a server for remote access through URL. GitHub server is recommended. [10 points]

1.2 Create an XML file Hotels.xml as an instance of your schema file. Enter the information of at least ten (10) real hotels into the Hotels.xml file, in addition to the hotel example given in Figure 1. You can use any tool to edit the file. You must include all the elements given in the XML tree in Figure 1. If an element has a Required attribute, you must provide the attribute for each of the elements. If an element has an Implied (Optional) attribute, you will provide this attribute for some elements, but not for all elements. For some instances, more than one phone numbers must be provided. The file must be saved in a server for remote access through URL. GitHub server is recommended. [5 points]

1.3 Create an XML file HotelsErrors.xml, which is a copy of Hotels.xml, with following errors (mutations) injected. These five errors are: [5 points]

(1) The root element name is Hotel, instead of Hotels.
(2) A required attribute is missing;
(3) A Phone element is missing on one of the hotels.
(4) An Address element has no closing tag.
(5) Two names are provided for one of the hotels.

The file must be saved in a server for remote access through URL. GitHub server is recommended.

2. Develop two methods: Verification(string xmlUrl, string xsdUrl) and Xml2Json(string xmlUrl)

2.1 public static string Verification(string xmlUrl, string xsdUrl) method. It takes the URL of an XML (.xml) file and the URL of the corresponding XMLS (.xsd) file as input and validates the XML file against the corresponding XMLS (XSD) file. The method returns "No errors are found" or an error message showing the available information at the error point. You must use the files that you created in this assignment, with and without fault injection, as the test cases. However, your service operation should work for other test cases (other XML file and its corresponding XSD file) too. [10 points]

2.2 public static string Xml2Json(string xmlUrl) method. The method will convert the XML without errors into a list of Json elements. The Json file should have the following structure: [10 points]

```
{
    "Hotels": {
        "Hotel": [
            {
                "Name": "Westin",
                "Phone": [
                    "480-968-8885",
                    "800-937-8461"
                ],
                "Address": {
                    "Number": "11",
                    "Street": "E 7th St",
                    "City": "Tempe",
                    "State": "AZ",
                    "Zip": "85281",
                    "NearestAirport": "Sky Habor"
                },
```

```
            "_Rating": "4.2"
        },
        {
            ...
            ...
            ...
        }
    ]
  }
}
```

Note: If Rating element is not given in the XML file, the Rating object in the Json file should not appear. **The returned Json text needs will be de-serializable by the autograder with Newtonsoft.Json package without any errors (JsonConvert.DeserializeXmlNode(jsonText)).**

3. Develop Main(string[] args) method containing the following operations. [10 points]

public static void Main(string[] args) method. The method will

(1) Call the Verification(string xmlUrl, string xsdUrl) method to verify that the XML file without errors confirms to its XSD definition. The input parameters are the files' remote URLs. The method should output the message: No errors are found.

(2) Call the Verification(string xmlUrl, string xsdUrl) method to verify that the XML file with errors does not confirm to its XSD definition. The input parameters are the files' remote URLs. The method should output the error message that lists all the errors.

(3) Call Xml2Json(string xmlUrl) method to convert the XML without errors into a list of Json elements.

The signatures of the classes, delegates, events, properties, and functions are given as follows. You must use all the classes, events, variables and functions listed. However, you can add more variables and functions. **Your code will be tested under .NET 4.7 with C# 7.0.**

```
using System;
using System.Xml.Schema;
using System.Xml;

namespace ConsoleApp1
{
  public class Program
  {
    // These URLs will be read by the autograder, please keep the variable name un-changed and link to the correct xml/xsd
       files.
    public static string xmlURL = "https://www.public.asu.edu/~YourASURITEID /Hotels.xml"; //Q1.2
    public static string xmlErrorURL = "https://www.public.asu.edu/~YourASURITEID /HotelsErrors.xml"; //Q1.3
```

```
    public static string xsdURL = "https://www.public.asu.edu/~YourASURITEID /Hotels.xsd"; //Q1.1

    public static void Main(string[] args)
    {
      // Q3: You can pick two of three
       string result = Verification(xmlURL, xsdURL);
       Console.WriteLine(result);

       result = Verification(xmlErrorURL, xsdURL);
       Console.WriteLine(result);

       result = Xml2Json("Hotels.xml");
       Console.WriteLine(result);

    }

    // Q2.1
    public static string Verification(string xmlUrl, string xsdUrl)
    {
       …
       …
       …
       //return "No errors are found" if XML is valid. Otherwise, return the desired exception message.
    }

    // Q2.2
    public static string Xml2Json(string xmlUrl)
    {
       …
       …
       …
       // The returned jsonText needs to be de-serializable by Newtonsoft.Json package.
       (JsonConvert.DeserializeXmlNode(jsonText))
        return jsonText;

    }
  }
}
```

Gradescope Testing and Submission

Testing: Based on your selection of the sub questions in question 3, provide your test function calls, inputs and output your test results in the Main method.

You must create and provide URLs to all three files: “Hotels.xml”, “Hotels.xsd” and “HotelsErrors.xml”. These files above are not programs and can be deployed to any web location, such as GitHub server. See Question 4 in Practice Exercises section. You may not use WebStrar server, as it is a shared server. This assignment is an individual assignment.

Notes:

1. You must follow what is defined in the assignment document. You have flexibility to choose your implementation details if they are not explicitly specified in the document. If you are not sure on any issue, ask the instructor or the TA by posting the question on the course discussion board.

2. The program and each component of the program must be well commented.

3. In this assignment, you may not need to have external input, but you must set your internal inputs in such a way that the program is properly tested, for example, each client must have completed at least one ordering process before the server terminates.

4. The submitted code must be named "submission.cs", which is the only one you need to submit.

5. All other files, such as “Hotels.xml”, “Hotels.xsd” and “HotelsErrors.xml” are supposed deployed to web location, such as GitHub server, and the URLs should be provided as variable in the "submission.cs" (Check code template provided).

6. The submitted code must belong to the namespace: "ConsoleApp1"

Submission Requirement and Grading Rubric

The assignment will be graded using Gradescope. Please check the grading rubric in a separate document and follow the Gradescope submission instruction.

Late submission deduction policy:

- Grace period (Sunday): No penalty for late submissions that are received within 24 hours of the given due date.
- 10% grade deduction for every day after the due date (from Monday through Tuesday!)
- Grade will be automatically deducted, and the grader does not need to calculate the penalty.
- No submission will be allowed after Tuesday midnight. The submission link will be disabled at 11:59pm on Tuesday. You must make sure that you complete the submission before 11:59pm. If your file is big, it may take more than an hour to complete the submission!