CSE 445 Assignment 4 (80 Points)

Spring 2019

Due: Wednesday, April 17th 2019, by 11:59pm (Arizona Time)

Introduction

The aim of this assignment 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 and their classes in .Net FCL. By the end of the assignment, you should have applied these concepts and techniques in creating an XML file, its schema, its style sheet, and have written Web services and an SOA application to process these files.

This is an **individual assignment**. Each student must complete and submit independent work. No cooperation is allowed, even among the team members for assignment 3. You can use your ASU personal Web site space to host your XML file (see Part 0 question 4), and use localhost to host the services in Part 2.

Part 0 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.

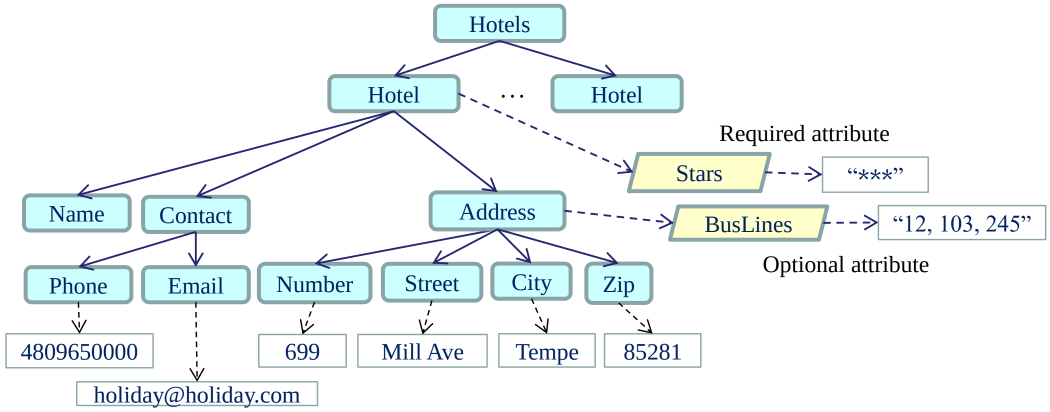
3. Study for the questions 2 through 8 in text Section 4.8. Make sure that you understand these questions and can briefly answer these questions. Study the material covered in these questions can help you to prepare for the exam and understand the homework assignment.

4. You can use WebStrar server account or ASU file service to upload XML and XSD file. If you have not activated your file service and personal Web hosting site at ASU, you can activate them at: https://selfsub.asu.edu/apps/WebObjects/ASURITEActivation. Then, you can search for Uploading Your Personal Web page within ASU search page to find the steps of uploading your file. Notice that ASU Personal Web site space hosts files only. You can use it host your .xml, .xsd, and .xsl files. It does not host programs such as Web services and Web applications. IIS are not installed.

Part 1 Creating XML, and XSD Files (40 points)

This part has 40 points. The grade will be based on your Part 1 submission. You will use three files created in Part 1 in your Part 2 questions, and you must include them in Part 2. You may modify these files when you resubmit them in Part 2. However, we will not grade these files in Part 2. We will grade questions 4 and 5 only in Part 2.

In this assignment, you will create a directory of hotels, which can be represented as an XML tree. The diagram below shows 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 of boxes have different meanings. They represent elements, attributes, and text contents/values, respectively. The structure of element and attributes given in the diagram below are required to implement, while the given text contents/values in the diagram are examples.



1. Create an XML file called Hotels.xml that comply with the XML tree given above. Enter the information of at least five (5) real hotels into the Hotels.xml file (you can find actual hotel names from the web). You can use any tool to edit the file. 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. [15 points]

2. Write the Hotels.xsd file that defines the XML schema allowing the structure shown in the diagram above. You can use any tool such as visual studio to create/edit the file. [10 points]

3. Create an ASP.NET Web site application (not a Web service) that takes the URL of an XML file as input (textbox) and display the element (tag) names, text contents, and attribute names, and attribute values in the GUI page. You can use any traversal algorithm discussed in the class to navigate the XML document. Make your Hotel.xml as the default input. **[Study the Example given with this homework before you attempt this question]** [15 points]

Submission list for questions 1, 2, 3: The complete solution folder with all project files question 3 and the application that can be tested by the TAs. You also need to submit the files used in the question 2, such as Hotels.xml, Hotels.xsd. Compress all these file into a zip file for submission. **You can name the project as hw4partI**

Part 2 Creating Web Services to Process XML, XSD, Files (40 points)

4. Develop a Web service (.svc) with **two** of the APIs listed below. The node mentioned in the sub questions below includes every component (element, content, and attribute) showing the XML tree in part 1.

4.1 API "verification" 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 API returns "No Error" or an error message showing the available information at the error point. You must use files that you create in the previous questions, with and without fault injection, as the test case. However, your service operation should work for other test cases too. [15 points]

4.2 Web operation "XPathSearch" takes the URL of an XML (.xml) file and a path expression as input. It returns the path expression value of the given path. It could be a list of nodes, the content value, etc., depending on the path give. [15 points]

5. Create a Web site application (ASP.NET), and add the project into the same "solution" that hosts your web service. The Web site application must provide a GUI (TryIt Page), which allows entering the required inputs, such as URLs and keyword, path, or contents, based on the questions that you select. The GUI must have the buttons required to invoke the service the service operations implemented above

- The button that invoke the validation of the XML file against the schema file;
- The button searches by keyword or by path in the XML file.

The Web application must use the Web service created in Question 5 to perform the required processing, and display the return message in the GUI. You can use a textbox, a list box, or a label to display your output. This assignment will be implemented on localhost or IIS Express. [20 points]

Submission list for question 4 and 5: The complete solution folder with all project files for the services and the application that can be tested by the TAs. You can create a single solution and have the web service and the ASP .NET application implemented. **You can name the solution as hw2partII**

Submission Requirement

All submissions must be electronically submitted to the assignment folder where you downloaded the assignment paper. All files must be zipped into a single zip file.

Grading

The TA will grade your program following these steps:

(1) The TA will read your program and give points based on the points allocated to each component, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementations of each function.

(2) Compile the code. If it does not compile, 40% of the points given in (1) will be deducted. For example, if you are given 20 points in step (1), your points will become 12 if the program fails to compile.

(3) If the code passes the compilation, the TA will execute and test the code. If, for any reason, the program gives an incorrect output or crashes for any input, 20% of the points given in (1) will be deducted.

Please notice that the TA will not debug your program to figure out how big or how small the error is. You may lose 40% or 20% of your points for a small error such missing a comma or a space!

Late submission deduction policy:

For each part of the submission

- No penalty for late submissions that are received within 24 hours of the given deadline;
- 1% grade deduction for every hour after the first 24 hours of the grace period!