ASU Ira A. Fulton Schools of Engineering
Arizona State University

CSE 598: Advanced Software Analysis and Design

BMI Calculator Project Part 1

Purpose

This project, with opportunities for both creativity and practical application, is designed to provide learners with a suitable environment for applying practical concepts covered in this course. Learners will draw inspiration from real-world scenarios and identify a service-oriented application of their choosing. Using Visual Studio, an industry-standard IDE, they will develop SOAP and REST web services. By developing these web services, learners will get hands-on practice with web service development and service invocation. Looking ahead to BMI Calculator Project Part 2, learners will also be able to practice building an application.

Objectives

Learners will be able to:

- Develop C# WCF REST web services in Visual Studio.
- Develop C# WCF SOAP web services in Visual Studio.
- Develop ASP.NET website application that uses REST and SOAP web service.

Technology Requirements

- Visual Studio 2019
- .Net Framework 4.7.2
- A way to record your screen

Project Overview

Develop REST and SOAP web services.

Project Description

The BMI Calculator Project has been split into two (2) parts. In this first part, you will develop REST and SOAP web services.

In the course, there are Demonstration videos of REST API using ASP.NET and SOAP API using ASP.NET. There is also an example project of both SOAP and REST web services. Please review these resources in detail, as they will provide valuable information as you complete your project.

Directions

Technology Set Up

You have **two (2) options** to complete the project code: A) You can use a **local setup**, or B) you can use **Apporto**. Your choice will likely depend on what technology you have access to. Please review the set up directions for each option in the *Welcome and Start Here* module of your course, "Technology Set Up: BMI Calculator Project Part 1", and select the one most appropriate for you and your system.

Project Directions

For this part of the BMI Calculator Project, use C# and Visual Studio 2019. Develop both web services and the ASP.NET client application in the same Visual Studio solution. Include screenshots of the outputs of your SOAP and REST web services deployed to the web server with your submission. All of this should be submitted as a single ZIP file.

Once you have developed the web services and the ASP.NET client application, create two (2) video demonstrations of your applications' outputs (one for SOAP and one for RES). Each video must start from you opening your ZIP file, followed by compilation and run. You must demonstrate every possible test case scenario that you tested as well. Include both videos in a single ZIP file submission as either YouTube links or MP4s.

SOAP Web Service

Your WCF SOAP web service must take height in inches (in) and weight in pounds (lb).

The service must contain two (2) operations:

`double myBMI(int height, int weight);` // calculates the BMI.

Use this equation to calculate the BMI: **bmi = [weight (lb) / height (in) / height (in)] x 703**.

`bmi` `myHealth(int height, int weight);` // this API returns the **bmi** structure, which consists of following data members:

1. `bmi`: double value that holds the BMI of the person
2. `risk`: depending on the BMI value, return one (1) of the following messages:
   - *You are underweight if BMI is < 18: Blue Color*
   - *You are normal if BMI is ≥ 18 and < 25: Green Color*
   - *You are pre-obese if BMI is between 25 and 30: Purple Color*
   - *You are obese if BMI is greater than 30: Red Color*
3. `more`: array of strings that has the following three (3) external links:
   - `"https://www.cdc.gov/healthyweight/assessing/bmi/index.html"`
   - `"https://www.nhlbi.nih.gov/health/educational/lose_wt/index.htm"`
   - `"https://www.ucsfhealth.org/education/body_mass_index_tool/"`

REST Web Service

Your WCF RESTful web service must take height in inches (in) and weight in pounds (lb).

The service must contain two (2) operations:

`double myBMI(int height, int weight);` // calculates the BMI.

Use this equation to calculate the BMI: **bmi = [weight (lb) / height (in) / height (in)] x 703**.

`bmi` `myHealth(int height, int weight);` // this API returns the **bmi** structure, which consists of following data members:

1. `bmi`: double value that holds the BMI of the person
2. `risk`: depending on the BMI value, return one (1) of the following messages:
   - *You are **underweight** if BMI is < 18: Blue Color*
   - *You are **normal** if BMI is ≥ 18 and < 25: Green Color*
   - *You are **pre-obese** if BMI is between 25 and 30: Purple Color*
   - *You are **obese** if BMI is greater than 30: Red Color*
3. `more`: array of strings that has the following three (3) external links:
   - `"https://www.cdc.gov/healthyweight/assessing/bmi/index.html"`
   - `"https://www.nhlbi.nih.gov/health/educational/lose_wt/index.htm"`
   - `"https://www.ucsfhealth.org/education/body_mass_index_tool/"`

Submission Directions for Project Deliverables

You are given a limited number of attempts to submit your best work. The number of attempts is given to anticipate any submission errors you may have in regards to properly submitting your best work within the deadline (e.g., accidentally submitting the wrong paper). It is **not** meant for you to receive multiple rounds of feedback and then one (1) final submission. Only your most recent submission will be assessed. You must submit your BMI Calculator Project Part 1 as two (2) separate ZIP files in the designated submission space in the course.

Learners may not email or use other means to submit any project for review, including feedback, and grading. Your BMI Calculator Project Part 1 includes two (2) deliverables:

1. **Web Services ZIP file:** The web services and output screenshots must be included in a **single ZIP file** with the correct naming convention: *Your Name_CSE 598 ASAD_BMI Calculator Project Part 1*.

2. **Demo Videos ZIP file:** The video demonstrations must be included as YouTube links or MP4s in a **single ZIP file** with the correct naming convention: *Your Name_CSE 598 ASAD_BMI Calculator Project Part 1_Demos.*

Making File Submissions in Canvas

Before submitting, confirm that your deliverables follow the requirements for the project, and then submit your work in the designated submission space in the course. File submissions are manually graded by the course team.

1. In your course, go to **Submission: BMI Calculator Project Part 1**.

2. Click **Start Assignment**.

3. Click **Choose File**.

4. Locate and select **one (1)** deliverable file from your device.

5. If needed, click **+Add Another File** and repeat Steps 3 and 4 until all deliverables are added.

6. Select the **agreement** and then click **Submit Assignment**.

7. (If needed and allowed) To resubmit files:
    a. Return to the Canvas submission space, click **New Attempt**, and repeat the process from Step 3.

Evaluation

Please review the rubric for how your BMI Calculator Project Part 1 will be graded. Projects will be evaluated based on each criterion and will receive a total score. Projects missing any part of the project will be graded based on what was submitted against the rubric criteria. Missing parts submitted after the deadline will not be graded.

Review the course syllabus for details regarding late penalties.

Rubric

Rubrics communicate specific criteria for evaluation. Prior to starting any graded coursework, learners are expected to read through the rubric, so they know how they will be assessed. You are encouraged to self-assess your responses and make informed revisions before submitting your final report. Engaging in this learning practice will support you in developing your best work. Points may be deducted at the discretion of the faculty for disorganized submissions that convolute the grading process.

Component	No Attempt	Undeveloped	Developing	Approaching	Meets
SOAP Web Service Implementation	Provided no response.	Implemented service is not a SOAP service or is missing all of the functionality required according to the directions.	Implemented service is a SOAP service and has some of the functionality required according to the directions, but is missing the majority of what was specified.	Implemented service is a SOAP service and has most of the functionality required according to the directions, but not all of what was specified.	Implemented service is a SOAP service and has all the functionality required according to the directions, which involves the two (2) operations, including the correct equation for BMI calculation and three (3) data members for the bmi structure.
REST Web Service Implementation	Provided no response.	Implemented service is not a REST service or is missing all of the functionality required according to the directions.	Implemented service is a REST service and has some of the functionality required according to the directions, but is missing the majority of what was specified.	Implemented service is a REST service and has most of the functionality required according to the directions, but not all of what was specified.	Implemented service is a REST service and has all the functionality required according to the directions: the two (2) operations, including the correct equation for BMI calculation and the three (3) data members for the bmi structure.
SOAP Web Service Demo Video	Provided no response.	Video does not demonstrate the learner's SOAP web service.	Video demonstrates the learner's SOAP web service, but may not begin from opening the ZIP file or show the compilation and run. Video does not include a demonstration of test case scenarios.	Video demonstrates the learner's SOAP web service, beginning with opening the ZIP file and showing the compilation and run. Video includes a demonstration of very few test case scenarios.	Video demonstrates the learner's SOAP web service, beginning with opening the ZIP file and showing the compilation and run. Video includes a demonstration of every possible test case scenario that the learner tested.

Component	No Attempt	Undeveloped	Developing	Approaching	Meets
REST Web Service Demo Video	Provided no response.	Video does not demonstrate the learner's REST web service.	Video demonstrates the learner's REST web service, but may not begin from opening the ZIP file or may not show the complete compilation and run. Video does not include a demonstration of test case scenarios.	Video demonstrates the learner's REST web service, beginning with opening the ZIP file and showing the compilation and run. Video includes a demonstration of very few test case scenarios.	Video demonstrates the learner's REST web service, beginning with opening the ZIP file and showing the compilation and run. Video includes a demonstration of every possible test case scenario that the learner tested.