Web Service Need Assessment

Lab

Meissa Bayo

2019

**Web Service Lab**

In this lab, you are being asked to create a web service that allows users to enter three digits and then display the sum of these digits. The requirements for the system are as follows:

1. Construct your web service as a class and write it in C#
2. Use the HTTP Post protocol

Submit all the files you have created via BlackBoard.

Need assessment:

In this lab I was task to create a web service lab that allow the user to enter three numbers and then get a sum in return from the web service.

Figure1.1

A screenshot of a cell phone

Description automatically generated

This is the html page that I created for the web service.

Figure 1.2

A screenshot of a cell phone

Description automatically generated

This is seending a post request to the web service witch goes and get the headers from the web service

Figure1.3

A close up of a screen

Description automatically generated

This is the functionality behind the program this is what do the calculation.

Figure1.4

A black sign with white text

Description automatically generated

You set this up in the webconfig this is the get and post method

Figure1.5

A picture containing indoor

Description automatically generated

This is the header that we set up so the html file that I created comes and get this

Figure1.6

A screenshot of a cell phone

Description automatically generated

This is the script that I created to go the the web service and request the three numbers that are being post.

**Functional Requirements**

* The software should provide a user to insert 3 numbers and return the sum
* A POST call should be performed using the user data as parameters to the web service
* Headers need be established otherwise you run into Cross-Origin Response Issues
* A Web Reference must be designed so the application understands specifically where to get the web service
* The application should be able to reset on user error or mistype

**Non-functional Requirements**

* The UI should be both appealing also simple to operate
* The software should be effective and accessible to use to not discourage users
* Forms should be engaging and uniquely designed to present information in a satisfactory manner
* The software should always be quick and understanding despite improvements in workload
* Error checking must be performed to avoid accidental errors.