

Corey Welborn
Simple Documentation Template - Stubbing Project
CSC 414
11.2.2020

Section 1

1.0 - Scope

The objective of this project is to create a menu for a webpage. The software should be able to run on any server with Apache. The menu should update the body of the page using the client's browser through Javascript. The implementation of the menu will allow users to easily find and navigate to the appropriate page to view the content intended.

1.1 - Identification

This software is needed to complete a CSC 414 project. The final code will be uploaded to GitHub for review and future availability. This is Version 1.

1.2 - System Overview

Any server running Apache that needs a menu for an HTML website is appropriate. The page can be included in any public folder and will be functional and accessible to any client viewing the website.

1.3 - Document Overview

This document will aid in the organization and future implementations of the code. We will outline any documents that were referenced during development, as well as any requirements we discover along the way. We will draw any diagrams necessary to outline the flow in our design. Testing documentation will also be included to finalize our understanding of potential pitfalls in our solution.

Section 2

2.0 Referenced Documents

Javascript, HTML, and CSS were implemented in this project, throughout the design phase, a few websites were of service:

1. W3 for Javascript - w3schools.com/js/DEFAULT.asp
2. W3 for CSS - w3schools.com/css/
3. W3 for HTML - w3schools.com/html/

Section 3

3.0 - Requirements

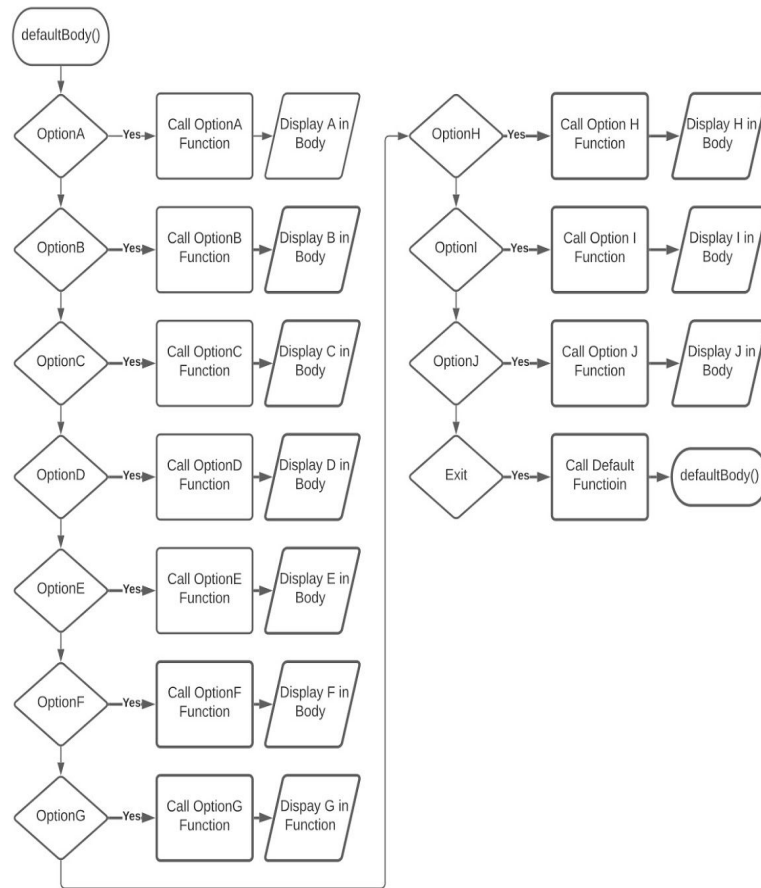
The ultimate goal of this project is to deliver content based on a user's input, the content should be displayed. Also, there should be a default state when nothing has been chosen or when the client is finished reviewed said content.

Source	Identifier	Statement	DDS Identifier	TDS Identifier	PDR	CDR	ProtoType	20 Build	Final
CSC 414	M1.00	Module shall update the body to corresponding text when input is given	CDR		Y	Y	P	P	P
CSC 414	M1.01	Module shall return default state when exit is desired	CDR		Y	Y	P	P	P

Section 4

4.0 - Design

Given the nature of the requirements, a website that will be hosted on a server seems to be the best course of action. HTML will be used to build a framework, CSS will be used to style each component, and finally Javascript will be the scripting language of choice to handle and input from the user. Below is a flow chart to help describe the architecture of our intended software:



To satisfy requirement identifier M1.00, we will create buttons in a div tag that will be aligned to the left, this button will have an onclick attribute that will call a function tied to the button clicked. This function will update the HTML in a div tag that is aligned in the center of the screen, this update will set the text in the centered div to the corresponding data of the clicked button. Also, to satisfy requirement identifier M1.01, we will implement an exit button in a similar fashion, but it will return the client to the default text that was loaded as soon as they visited the site. The desired deadline is November 17, 2020.

Section 5

5.0 - Testing

Upon testing the final product, we have seen repeated success on every test. When a user clicks a button, the desired content is displayed. When the client clicks the exit button, they are returned to the default state. Refreshing the screen also returns the user to the default state. This software has been thoroughly tested for bugs, and it appears that it is safe to deploy.

Stubbing Project

Option A

Option B

Option C

Option D

Option E

Option F

Option G

Option H

Option I

Option J

Exit

Default Text

Here we have a screenshot of the page loaded, the defaultFunction is ran and the centered div tags content is displayed. The user clicks Option A from the buttons in the left div tag. When this happens the optionA function is called and the center div tag's HTML is updated to display the content of Option A. Here we have the output:

Stubbing Project

Option A

Option B

Option C

Option D

Option E

Option F

Option G

Option H

Option I

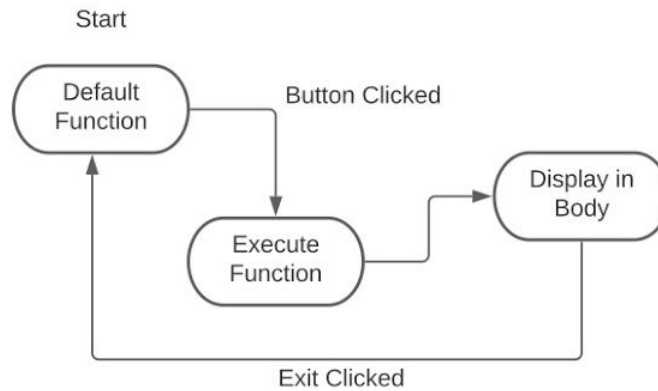
Option J

Exit

Option A not implemented.

Appendices

During the design phase, we implemented a state diagram to help us understand how the state of the content would flow:



Also, here we have a repository for the finished product hosted by GitHub:
github.com/coreywelborn211/StubbingProject.git