Technical Writing Assignment

Aaron Schwartz

Questions 1-3

When I click on a hyperlink in my browser, the browser will look at the code associated with that hyperlink on the page currently being rendered. In that code, the browser will find the information hidden in the code about the link, namely, its Uniform Resource Locator, or URL for short. Inside of that URL is the name of the website the hyperlink is supposed to take me too. The browser will find this name, and then use DNS to get the IP address of the server the website is housed in. The next step is for the browser to open a TCP connection to the server using the IP address. Once it has established that connection, it can send an HTTP/HTTPS request to the server, asking for a copy of the code for the website. Once the server has approved the request, it sends my browser a confirmation message over a TCP connection, and then begins to use the connection to send the source code for the website over to my browser piecemeal, in data packets, which my browser can then reassemble into html/CSS/JavaScript/whatever else code that makes up the website I want to go to. Then the browser renders said code and the process is finished.

Questions 4-5

The distinction between server code and client code is, in simple terms, client-side code is everything that can be seen or done by the user of the browser, while server-side code is all the code that cannot. For example, client-side code is everything that I would receive from the server via the method I outlined above. The HTML Source code, the CSS for that HTML, things like that. Client-side code also includes things like GUIs and forms and buttons, which function as the intermediary between the user, and the server-side code. When I fill out a form to go into a database, that's client-side code. When the client then sends the contents of that form to the server, it is the Server-side code's job to take the input sent by the client and put it into the database, so that it could then be seen by others. Probably the most straightforward and uncomplicated example of the client/server code dynamic is email. The user interface of my email client is client-side code. I use it to view my email accounts, view my mail, and compose new mail. Once I have composed an Email, it gets sent to SMTP server, and the rest of the process of that email reaching the person I sent it to falls under server-side code.

Questions 6-7

In the case of the link provided for the first question, there is only 1 client asset. That asset being the HTML document that makes up the webpage. In general, though, you can determine the number of client-side assets a webpage or website has access to by using the dev tools built into most browsers to inspect the webpage's sources. Server-side code much trickier. As I mentioned in the last question, server-side code is all the code that happens in the server, outside of the direct observation of the user, who can only interact with the server via the client. I'm not entirely certain that it is possible to determine how many possible instances of server-side code can be available at any given time without intimate knowledge of the server itself, since that number would vary based on the server (or severs plural) 's architecture.

Question 8

Runtime is much simpler to understand. Runtime refers to the phase of the programming cycle when the code is actually executed. The runtime environment is composed of the code itself, and the scope and state of all the variables in the code at execution time. Runtime, as its name suggests, is the phase of the coding process at which runtime errors can occur. Runtime errors are the result of unforeseen situations messing up code that should work in theory. Such as how valid code can produce errors like have data of the wrong datatype fed to a function, without being caught by Exception handling.

Question 9

This question is quite similar to question 7. Without backend access and knowledge of the server's architecture, a front-end user would have no way of knowing how many instances of a database could be connected to a server application. The number of possible instances would also depend on the specific server application.