

Cheeks Family Website

Mobile Phone Support

User Interface (UI), “Look and Feel”

Typically, businesses will have two separate web application code lines or two separate applications, one for the standard screens (*tablet, laptop and desktop*) and another for mobile phones (*small screens*). That approach results in easier design, development, testing and maintenance for each code line/application because of limiting the configurations each will execute in.

The result is two different user interfaces (“look and feel”), one for mobile phone screens the other for standard screens. So, a user is required to be familiar with two distinctly different user interfaces.

The new website software engine implements and controls a common User Interface (“look-and-feel”). The mobile phone User Interface, it’s “look and feel”, is essentially identical to the tablet, laptop and desktop User Interface. No “hamburger” (*three horizontal line in a corner*) main menu. The same six menu tabs appear at the top of each primary webpage, no matter the screen size.

Dropdown submenus are identical and both menu and submenu tabs behave identical to their counterparts that appear on full-sized displays. Same colorway, same highlighting when selected, etc.

This means that you only need to become familiar with a single User Interface, no matter what size screen of your device in use, full-sized screen or small mobile phone screen.

Testing Results

The primary challenge to successful web applications supporting mobile phone is the almost infinite number of configurations that the application will be executed on. There are multiple phone models, multiple versions of the phone models, multiple phone operating systems, multiple versions of each operating system, multiple browsers (*when used*) and multiple versions of each browser.

Typically, a web application supporting mobile phone use is targeted at two product families identified by their operating system: Android and iOS (Apple). And targeted at three “mainstream” browsers: Chrome, Firefox and Edge. That still results in a huge number of combinations that will be executing the application. So, it is impossible to ensure “clean” application visual behavior in each combination of phone modes, phone operating systems and browser (*if used*).

The new engine has been successfully tested on multiple, current Samsung Galaxy phones, each with the latest release of the Android operating system and the latest release of Chrome, Firefox and Edge browsers. The tested visual display with Chrome and Firefox was excellent in both portrait mode (*tall, vertical*) and landscape mode (*wide, horizontal*) with almost identical displays with each browser.

Testing Results *(continued)*

Edge (*Microsoft browser*) was successful in both modes but the visual display was reduced in size, occasionally requiring the use of “Pinch-to-Zoom” functionality to expand the visual display.

Testing also completed successful using an iPhone16 configured with the latest iOS version.

Caveats and Support

In spite of any amount of successful testing there will be some number of mobile phone configurations where the visual display will range from mildly “off” to completely distorted. That can be due to the “age” of the mobile phone, older/no longer supported phone operating system, older version of a web browser that is no longer supported, etc.

So, some number of you **will**, not might, experience this situation at some level. If that occurs and results in a difficult to use situation, or worse, unusable due to visual display issues, contact me via the [Contact Support](#) webpage.

Include in the comments, your specific phone model and operating system version (*information available in the Settings option on your phone*). And the browser used and the version, which is available in the Settings option of the browser’s menu. If you have difficulty obtaining that information, include that in your comment and we will make arrangements for assistance.

If there are enough users, with the same or similar configurations, experiencing a significant level of unexpected visual behavior, I will investigate the amount of effort to add a code branch specifically for those configurations.

The engine can determine many characteristics of the device you are using, including screen size (*width and height*), orientation (*portrait or landscape*), aspect ratio (*width-to-height*), and with some JavaScript code, determine what browser and version is in use. All of that occurs as soon as you start the application (*browsing*) but prior to seeing anything from the application.

That functionality allows the engine to make configuration-specific decisions about what code path to execute. That is how the engine “knows” to adjust the visual display for a mobile phone (*small screen*) or a ‘standard’ screen.

A new code path for specific phone configuration(s), with visual display issues, would be executed by the engine when it detects initiation by a phone with the target configuration.