**Calculator Design Document**

**Introduction**

The Business Analyst has been working with the clients to determine their specific requirements for a calculator application.

From the requirements put forward by the clients, and after workshopping some design layouts, it has been determined that the design described below will meet their needs.

**Required Functionality**

The calculator needs to run on a web browser on Macs, PCs, iPhones and Android phones.

It must be able to:

* Perform addition, subtraction, multiplication, division and square root calculations.
* Display numbers up to 15 digits long, including the minus sign (if the number is negative), a decimal point (if a decimal fraction is being displayed) and the digits in the decimal fraction. Examples are -2345678.012345 and 123456789012345.
* Allow the user to clear and restart the calculation.
* Allow the user to change the displayed number from positive to negative and vice versa.
* Display timed advertising at the top of the screen.

The client would also like the calculator web application to be installable as a Progressive Web App (PWA) on mobile phones but this is dependent upon whether a web server with HTTPS access is available for this project, as HTTPS is mandatory for PWA.

**User Experience (UX)**

*Operation*

The calculator is to operate in the same manner as a typical calculator.

*User Interface (UI)*

The following screen layout has been requested.



**Project Schedule**

Stage:

1. Develop the calculator web application, without the advertising component, using HTML, CSS and JavaScript.
2. jQuery and CSS.
   * 1. Replace JavaScript with jQuery code in order to simplfy the web application.
     2. Take out as many inline styles as possible and put them in the stylesheet.
     3. Experiment with some other, popular jQuery APIs and some popular stylesheets such as bootstrap.css.
3. TypeScript.
   * 1. Download and install Visual Studio Code (VS Code) from <https://code.visualstudio.com/download>.
     2. Follow the tutorial at <https://code.visualstudio.com/docs/typescript/typescript-tutorial> in order to install the TypeScript compiler, develop a simple *Hello World* program and explore the functionality available in VS Code.
     3. Use VS Code to replace the JavaScript from Stage 1 with TypeScript.

At this point, discuss which language is the easiest to read, modify and maintain. (This may well be a personal preference rather than an objective one.)

1. Node.js.
   * 1. Install and configure a Node.js server on your Mac or PC.
     2. Set up the JavaScript/jQuery version of the calculator web application, ie the one developed in Stage 2, so it is accessible by your Node.js server.
     3. Test your program by running a web browser and accessing the calculator web application using a URL, eg <http://localhost:8080/myCalculator.html>.
     4. Ensure it runs on Chrome or Safari on the Mac, PC, iPhone and Android phone.
2. Using AJAX calls, display timed advertising at the top of the screen. (You will need to create a HTML page for each advertisement and then cycle through them, displaying each in turn at the top of the calculator.)
3. If a HTTPS web server is available, configure the calculator web application as a PWA and install it on a mobile phone.