The International Journal Of Engineering And Science (IJES)
|| Volume || 4 || Issue || 5 || Pages || PP.78-91|| 2015 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Liquidizer.js: A Responsive Web Design Algorithm

Abdulrehman Mohamed, Dr. Cheruiyot W.K, PhD, & Dr. Richard Rimiru, PhD
Jomo Kenyatta University of Agriculture and Technology, Mombasa Campus, Institute of Computer Science and Information Technology, P. O. Box94090–80107, Mombasa, Kenya,

ABSTRACT
Internet technology is dynamically changing at lightning speeds that the academic brains cannot absorb. Emerging technologies such as Internet of Things (IoT), fog computing, cloud computing and just to mention a few have recently emerged as novel technologies. These technologies have not yet sunk in to the minds of academic scholars, while superior techniques are currently emerging. As a result of these fluid changes, the study is intrigued by the Responsive Web Design (RWD) technology. RWD is a novel paradigm to develop one single website for different screen sizes of smart phones, tablets, laptops, and desktops among others. The websites become responsive by being accessible anytime, anywhere, and on any such devices. Although lots of ink has been spilled on responsive algorithm framework development, the study developed an enhanced algorithm with dynamic attributes such as text color, background color, font family, and font size manipulation. These attributes can be changed on the fly and be accessed by a single line of code by web designers. The methodology employed to develop the algorithm was jQuery library framework. The outcome of the study was threefold; first, to develop an enhanced algorithm coined Liquidizer.js, second, to distribute the source code of Liquidizer.js under the GNU General Public License, and third, to extend the jQuery library platform.

KEYWORDS: *Responsive Web Design (RWD), Algorithm Liquidizer.js, Fluid Grid Concept and jQuery library platform.*

Date of Submission: 18-May-2015 Date of Accepted: 05-June-2015

I. INTRODUCTION

In the more recent times of digital age, the internet users have evolved their surfing habit experience in the web to ubiquitous computing. In contrast to desktop computing, ubiquitous computing can occur using any device, in any location, and in any format. As a result of which, web standard designers are building websites which are getting closer and closer to the ideal of one web accessible to everyone and everywhere. Therefore, web designers are implementing Responsive Web Design (RWD) technologies to develop one website for various screen sizes from smart phone to main frame computer which is accessible anytime, anywhere and by any device type. It is asserted by [10] that, Responsive design is not a single technology but a set of techniques that allow web pages to serve the needs of both mobile and desktop users. The core components are: CSS @media queries, Fluid images and video, JavaScript, often triggered by window match Media, Server-side solutions, and Scalable Vector Graphic (SVG) to create resolution-free images. A responsive site may utilize one, some, or all of these technologies, depending on the intentions of its designers. Web page text is fluid by default: as the browser window narrows, text reflows to occupy the remaining space. Images are not naturally fluid: they remain the same size and orientation at all configurations of the viewport, and will be cropped if they become too large for their container. This creates a conundrum when displaying images in a mobile browser: because they remain at their native size, images may be cut off or displayed out-of-scale compared to the surrounding text content as the browser narrows. But responsive web design is not only about adjustable screen resolutions and automatically resizable images, but rather about a whole new way of thinking about design. A website designed with RWD adapts the layout to the viewing environment by using fluid, proportion-based grids, flexible images, and CSS3 media queries, an extension of the @media rule. Several alternative technologies to RWD had been in existence such as Adaptive Web Delivery (AWD) and Tableless Web Design (TWD) but had their challenges and limitation. It is urged by [14] that, when it comes to responsive typography on the web, there’s more to do than just resizing the text’s container and having the text reflow inside of it. But choosing a font type and color, to achieving legible font sizes, line heights, and line lengths on different screen sizes, there are several ways to go about achieving fluid and truly responsive text on the web. This view is also supported by [13], who states that, responsive typography is the use of fonts which adapts to different resolutions so they are still viewable, with the overall layout still intact. Unlike using simple fonts for a separate mobile site, one is using fonts which are complex as one likes, hence stretch or shrink according to the screen’s need.

Moreover, it is urged by [16], that RESS (Responsive Web Design + Server Side Components) is a concept that, combines adaptive layouts with server side component (not full page) optimization, whereby a single set of page templates define an entire Web site for all devices, but the key components within that site have device-class specific implementations that are rendered at server side. Moreover, according to [12], RESS is a concept that a browser-detection can be used to help inform an overall responsive design as opposed to being the be-all-end-all for templating. This means that partial pieces of content can be inserted intelligently and where appropriate (thinking images) into a larger layout that is given to all browsers and is governed by responsive design principles. Furthermore, it is explained by [17] that, RESS in conjunction with client-side ones such as media queries can produce faster-loading sites for access over cellular networks and also deliver richer functionality or usability avoiding some of the pitfalls of device-side-only solutions.

Furthermore, it is explained by [1] that, fluid grid concept as involving the margins, page widths and indentation are all aspects of page design which can aid readability. The web presents difficulties for the designer with each of these. Browser windows can be resized, thereby changing the page size. Different web devices (such as web TV, high resolution monitors, PDAs) have different minimum and maximum window sizes. As with fixed font sizes, fixed page layout can lead to accessibility problems on the web. This view is asserted by [15] that, liquid layouts are closely linked to media queries and special styles for optimization. Percentage-based widths alone will likely not be enough to accommodate ones design for a large variety of display sizes. A flexible grid-based layout is one of the cornerstones of responsive design.

Finally, it is explained by [5] that, media queries allows the pages to use different CSS style rules based on characteristics of the device the site is being displayed on, most commonly the width of the browser. With media queries, designers can build multiple layouts using single HTML documents and selectively provide style-sheets based on different features such as browser size, orientation, resolution or color. It is further urged by [4] that, viewport width isn’t the only thing media queries can detect. There are a ton of media features one can detect, including color, color index, aspect ratio, device aspect ratio, width, device width, height, device height, orientation, monochrome, resolution, scan, pixel-density and many more.

II. ALGORITHM DEVELOPMENT

In recent times, more and more people surf through the internet using mobile devices compared to a desktop computer. Consequently, mobile devices and computer screens designers have been trying to provide users with quality web-browsing but this hasn’t been able to afford adequately users’ needs that are exposed to traditional website layouts. Therefore, there is a need to switch to Responsive Web design which is capable to reshaping itself depending on various screen sizes and resolutions from largest screen sizes to smallest on mobile devices.

Therefore, in order for a computer to carry out some task, it has to be supplied with a program, which is an implementation of an algorithm. This is expressed in a computer programming language. The study implemented the algorithm design methodology by pseudo-code and stepwise refinement models.

2.1 Pseudo-code

Pseudo-code is a form of stylized (or structured) natural language. One of the problems encountered when writing programs is that of preciseness. A common fault among algorithms is that the process described is almost the intended one, but not quite. Therefore, a good algorithm must have the following properties; analogy: giving directions, following a recipe. These are rarely completely precise, but instead rely on the common sense of the person receiving the instructions. However computers are not equipped with common sense. Another common failing is that execution usually results in the intended process being carried out, but in certain circumstances (unforeseen or overlooked by the designer) it does not.

Furthermore, another required property of an algorithm is that each step can actually be carried out – in other words, the algorithm is executable. The point here is to make sure there are no “impossible” or unknown steps in your algorithm (e.g. algorithm relies on solving a sub-problem which is known to have no solution; algorithm asserts that a sub- problem will be solved without specifying how; etc.).

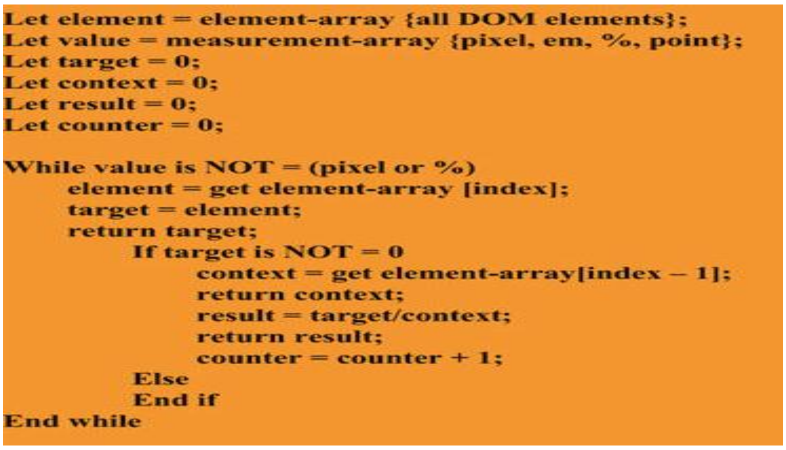
Finally, most processes are supposed to terminate. There are examples of some which don’t need to, but one assumes that all programs one is interested in should. Thus the designer of an algorithm must ensure: Preciseness of the algorithm (no ambiguities), All possible circumstances are handled, The algorithm is executable, Termination of the algorithm, Also have to worry about Efficiency - an algorithm may work correctly but be inefficient – by taking more time and using more resources than required to solve the problem - becomes more important for larger programs. For the pseudo-code to be more efficient then introduction of a technique of stepwise refinement is essential, and that is where our discussion is heading in the next sub-section below [6].

2.2 Stepwise Refinement

Stepwise refinement is a process that breaks a complex problem down into a number of simpler steps, each of which can be solved by an algorithm which is smaller and simpler than the one required solving the overall problem. Smaller and simpler, therefore easier to construct and sketch in detail. Sub-algorithms can themselves be broken into smaller portions. Refinement of the algorithm continues in this manner until each step is sufficiently detailed.

Refinement means replacing existing steps/instructions with a new version that fills in more details. When using stepwise refinement the designer must know when to stop refining. They must know when a particular step of the algorithm is sufficiently described to need no further refinement [7].

It seems clear from the above discussion that pseudo-code and stepwise refinement have alleviated the study's concern of alteration and modification despite their limitations. Hence, the techniques are the best choice for the study to adopt. Therefore, the Equation 1 below demonstrates one of the versions of a refinement pseudo-code of proposed algorithm Liquidizer.js



```
Let element = element-array {all DOM elements};
Let value = measurement-array {pixel, em, %, point};
Let target = 0;
Let context = 0;
Let result = 0;
Let counter = 0;

While value is NOT = (pixel or %)
      element = get element-array [index];
      target = element;
      return target;
            If target is NOT = 0
                  context = get element-array[index – 1];
                  return context;
                  result = target/context;
                  return result;
                  counter = counter + 1;
            Else
            End if
End while
```

Equation 1: Pseudo-code version of Liquidizer Algorithm

2.3 Programming Language

Although there are several web programming languages for algorithm implementation, such as Ajax (Asynchronous JavaScript and XML), ASP (Active Server Pages), XML (Extensible Markup Language), PHP (Hypertext Preprocessor) etc. but the most commonly use language is jQuey. It is a lightweight cross-browser JavaScript library, which emphasizes interaction between HTML DOM element and JavaScript. It is the most popular language among web developers; hence there is a lot of support forum for jQuery in the internet. This makes jQuery the best choice of the study.

Moreover, it is suggested by [2], an experience web developer and designer the best practice for plugin development for jQuery. In his article title "A Plugin Development Pattern", he proposed six best practices requirement to develop a plugin, which are; to claim only a single name in the jQuery namespace, to accept an options argument to control plugin behavior, to provide public access to default plugin settings, to provide public access to secondary functions (as applicable), to keep private functions private, and finally to support the Metadata Plugin. The study adopts the above best practices for algorithm implementation by jQuery language.

2.4 Liquidizer.js Development

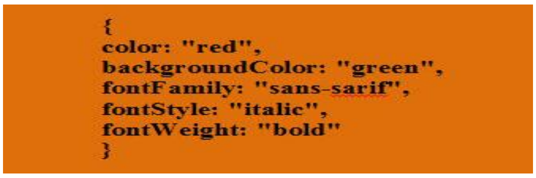
The study implemented the proposed design by coding in jQuery platform to build the algorithm Liquidizer.js. The source code can be accessed at https://github.com/almutwafy/Liquidizer, which is licensed under the GNU General Public License version 1.

2.5 Liquidier.js Description

Liquidizer.js is a responsive layout jQuery plug-in with dynamic attributes. The layout can be view in both small and large devices such as smart phones, tablets, laptops and desktops. It enables users to change the attributes on-fly, such as: color, background color, font family, font style, and font weight. The user, by simply specifying the values of the dynamic attributes and Liquidizer.js will do the rest. Moreover, the user can combine the 'selectors' and simultaneously apply the attributes at once!

2.6 How Liquidier.js Works

Liquidizer.js will dynamically change the selected elements using CSS name attributes property, when provided values by users. It has the capability to implement a responsive layout and to dynamically change the attributes for an element based on various values. The Equation 2 shows the Liquidizers attributes to be manipulated by the web designer.



```
{
color: "red",
backgroundColor: "green",
fontFamily: "sans-sarif",
fontStyle: "italic",
fontWeight: "bold"
}
```

Equation 2: Attributes of Liquidizer Algorithm

2.7 How to Use Liquidier.js

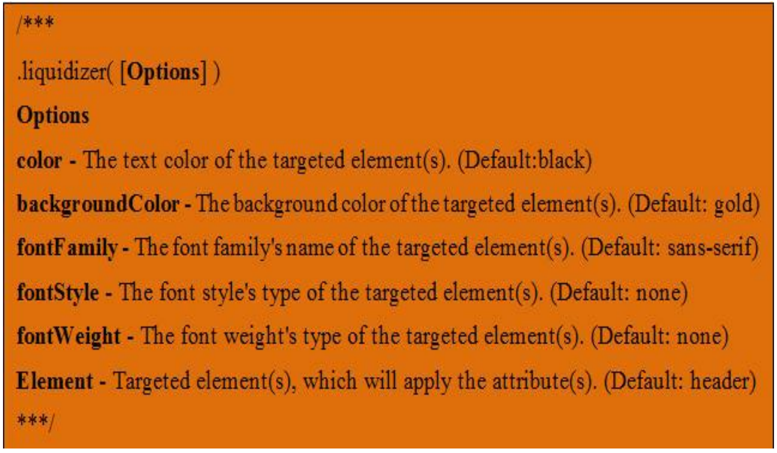
To implement liquidizer the following two simple steps are required.

1.) First, include jQuery and Liquidizer.js script files inside <head> tag as shown.
<script type="text/javascript" src="jquery.min.js"></script>
<script type="text/javascript" src="liquidizer.js"></script>

2.) Next, call the Liquidizer () function on jQuery object. It supports few settings, under the Configuration section.
$(document).ready(function() {$('selector').liquidizer();});

2.8 Liquidier.js Configuration

Moreover, the user has various options to dynamically apply attributes as shown below in Equation 3 below.



/***

.liquidizer(**[Options]**)

Options

color - The text color of the targeted element(s). (Default:black)

backgroundColor - The background color of the targeted element(s). (Default: gold)

fontFamily - The font family's name of the targeted element(s). (Default: sans-serif)

fontStyle - The font style's type of the targeted element(s). (Default: none)

fontWeight - The font weight's type of the targeted element(s). (Default: none)

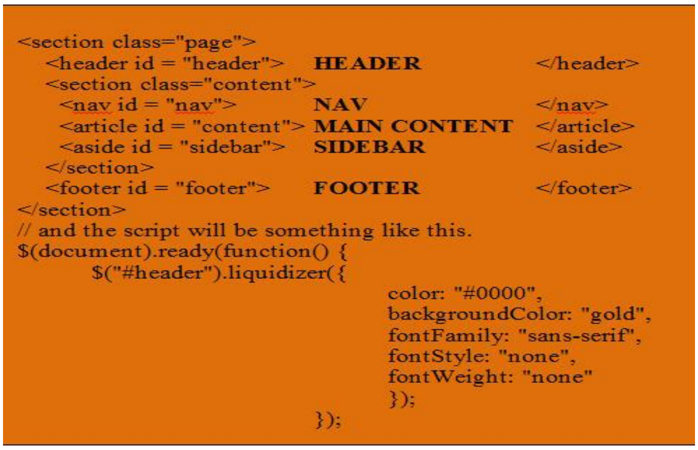
Element - Targeted element(s), which will apply the attribute(s). (Default: header)

***/

Equation 3: Configuration of Liquidizer Algorithm

2.9 Liquidier.js Configuration Example

The HTML mark-up for the responsive layout grid should be implemented as shown in Equation 4 below.



```
<section class="page">
  <header id = "header">    HEADER              </header>
  <section class="content">
    <nav id = "nav">         NAV                 </nav>
    <article id = "content"> MAIN CONTENT        </article>
    <aside id = "sidebar">   SIDEBAR             </aside>
  </section>
  <footer id = "footer">     FOOTER              </footer>
</section>
// and the script will be something like this.
$(document).ready(function() {
        $("#header").liquidizer({
                                        color: "#0000",
                                        backgroundColor: "gold",
                                        fontFamily: "sans-serif",
                                        fontStyle: "none",
                                        fontWeight: "none"
                                        });
                              });
```

Equation 4: HTML Implementation of Liquidizer Algorithm

The HTML implementation of Liquidizer Algorithm generates the following output as shown in Figure 1 below

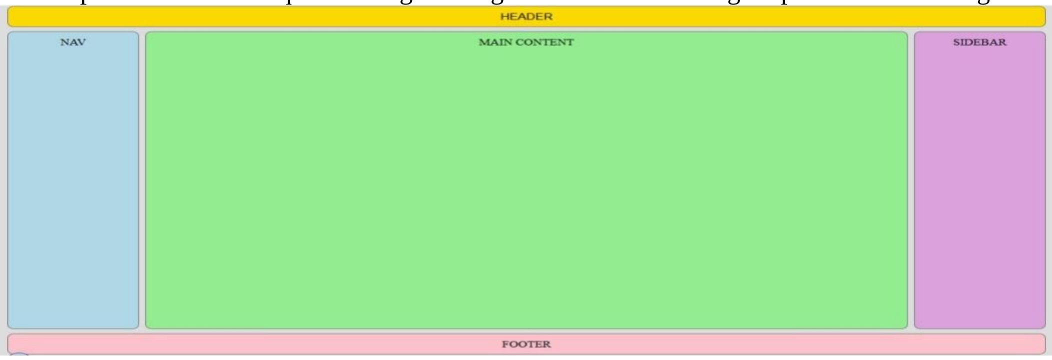


Figure 1: The Liquizer.js layout output

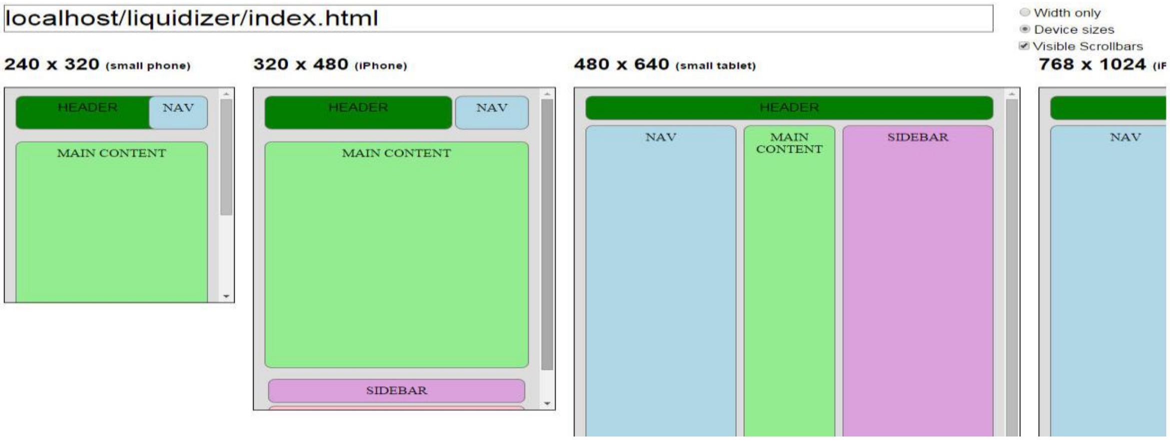
III. DATA COLLECTION

As software becomes more and more pervasive, there has been a growing concern in the academic community and Business world about software quality. This concern arises from the acknowledgment that the main objective of software in industries in is to balance price and quality to stay ahead of competitors. Some standard organizations, such as ISO and IEEE, tried the standardization of software quality by defining frameworks combining and relating software quality characteristics and sub-characteristics.

Meanwhile, researchers proposed software metrics as tools to measure programs source code, architecture, and performances. However, there is no clear and consensual relation among software quality frameworks and metrics. Moreover, the process of software quality assessment remains an open issue with many frameworks. Therefore, the study presents the most commonly used methods for testing , evaluation, and validating usability of a software; Matt Kersley RWD Tool, for testing Liquidizer.js, Bersoft Image Measurement (BIM) tool for validating Liquidizer.js, and measuring usability with USE questionnaire by [11], for evaluation of the Liquidizer.js algorithm.

3.1 Testing Liquidizer.js

The study used the Matt Kersley RWD Testing Tool. It is described by [8], as a testing tool that allows viewing responsive website in various screen sizes simultaneously in a single screen, while building or designing websites. The study prefers this tool mainly because it shows all the screen resolutions side-by-side which makes it easier for debugging. It is an online tool, which can be accessed at http://mattkersley.com/responsive/. The Figure 2 below demonstrates how the interface of the Testing Tool works.

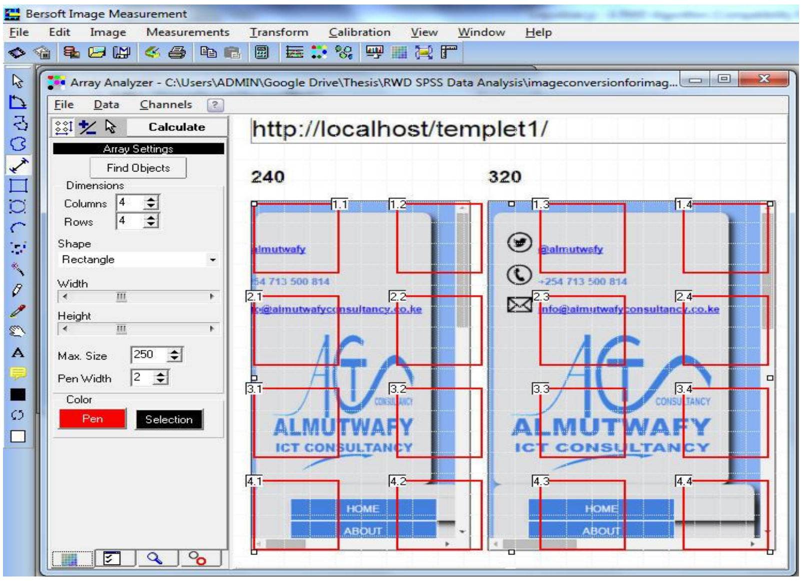

localhost/liquidizer/index.html

○ Width only
● Device sizes
☑ Visible Scrollbars

Figure 2: The Interface of Matt Kersley RWD Testing Tool

3.2 Validating Liquidizer.js

The study used the Bersoft Image Measurement (BIM) tool for algorithm Liquidizer.js validation. BIM is a powerful image asset management tool that makes it easy to measure, acquire, store, compare and analyze digital images. BIM performs image analysis functions that include gray scale and 24 bits color measurements: angle, distance, perimeter, area, point, line, pixel profile, object counting, histogram and statistics. BIM supports DICOM, JPEG, TIFF, and most popular image formats. The Figure 17 below shows the snap shot of BIM interface.[3]


Bersoft Image Measurement
File Edit Image Measurements Transform Calibration View Window Help
Array Analyzer - C:\Users\ADMIN\Google Drive\Thesis\RWD SPSS Data Analysis\imageconversionforimag...
File Data Channels
Calculate
Array Settings
Find Objects
Pen
Selection
http://localhost/templet1/
240
320
1.1
1.2
1.3
1.4
2.1
2.2
2.3
2.4
3.1
3.2
3.3
3.4
4.1
4.2
4.3
4.4
ALMUTWAFY
ICT CONSULTANCY
HOME
ABOUT

Dimensions
Columns 4
Rows 4
Shape Rectangle
Width
Height
Max. Size 250
Pen Width 2
Color

Figure 2: The Interface of BIM Validation Tool

3.3 Evaluating Liquidizer.js

The USE Questionnaire for Measuring Usability was used to collect data for evaluation of the Liquidizer.js. The questionnaire was developed over time and it started out with a large pool of items. It reported by [11] that, "The questionnaires were constructed as seven-point Likert rating scales. Users were asked to rate agreement with the statements, ranging from strongly disagree to strongly agree. Various forms of the questionnaires were used to evaluate user attitudes towards a variety of consumer products. Factor analyses following each study suggested that users were evaluating the products primarily using four dimensions; Usefulness, Ease of Use, Ease of Learning and Satisfaction." The Table 1 below summaries the USE Questionnaire.

Table 1: The USE Questionnaire Summary

A. Usefulness	B. Ease of Use	C. Ease of Learning	D. Satisfaction
It helps me be more effective	It is easy to use	I learned to use it quickly	I am satisfied with it
It helps me be more productive	It is simple to use	I easily remember how to use it	I would recommend it to a friend
It is useful	It is user friendly	It is easy to learn to use it	It is fun to use
It gives me more control over the activities in my life	It requires the fewest steps possible what I want to do with it	I quickly became skillful with it	It works the way I want it to work
It makes things I want to accomplish easier to get done	It is flexible		It is wonderful
It saves me time when I use it	Using it is effortless		I feel I need to have it
It meets my needs	I can us it without written instructions		It is pleasant to use
It does everything I would expect it to do	I don't notice any inconsistencies as I use it.		
	Both occasional and regular users would like it		
	I can recover from mistakes quickly and easily		
	I can use it successfully every time		

The questionnaires was constructed as seven-point Likert rating scales; from -3 (totally disagree) to +3 (totally agree), [11]. Therefore, the study adopts the USE Questionnaire for Measuring Usability to conduct a survey to evaluate the usability of the algorithm Liquidizer.js. The study built a questionnaire based on Lund's USE questionnaire of seven (7) point likert scale on level of agreement. The likert scale was coded as; 1 = "Strongly Disagree", 2 = "Disagree", 3 = "Slightly Disagree", 4 = "Neutral", 5 = "Slightly Agree", 6 = "Agree" and 7 = "Strongly Agree". The data collected was evaluated using Statistical Package for the Social Sciences (SPSS).

3.4 Evaluation Method

In order to evaluate the algorithm Liquidizer.js framework, a group of 30 web designers from three web design households were presented with the tool and shown how to use it. They were given simple instructions to include a line of code in a script tag to reference both the Liquidizer.js and jQuery.mini.js as shown below:

```
<script type = “text/javascript” src = “jquery.min.js”></script>
<script type = “text/javascript” src = “Liquidizer.js”></script>.
```

Thereafter, the questionnaires were given to the web designers to evaluate their perception of the algorithm Liquidizer.js framework as per the questions.

IV. RESEARCH FINDINGS AND IMPACT

The research findings and impact was presented into threefold; first, the testing of the Liquidizer.js by the Matt Kersley RWD Testing Tool, second, the validation of the Liquidizer.js by Bersoft Image Measurement (BIM) tool and third, the evaluation of Liqudizer.js by Statistical Package for the Social Sciences (SPSS). The SPSS tool conducted various tests to find relationship between variables, level of significance, reliability, and demographic data analysis. Specifically, the study performed Cronbach's alpha test, descriptive statistics tests, chi-square, Pearson correlation and Regression analysis.

4.1 Results for Liquidizer.js Testing

The Matt Kersley RWD testing tool was used to simulate varying screen size from mobile phone to a main frame computer. The Figure 3 [a] below presented the result before implementing the algorithm Liquidizer.js, while the Figure 3 [b] showed the findings after implementing the algorithm Liquidizer.js. The results are interpreted according to visual observation, whereby the page before integrating the Liquidizer was not responsive since the graphics and the menus are distorted and not fitting a single page. While Figure 3 b) below was responsive since the page can be viewed in single view without the graphics or menu being distorted.

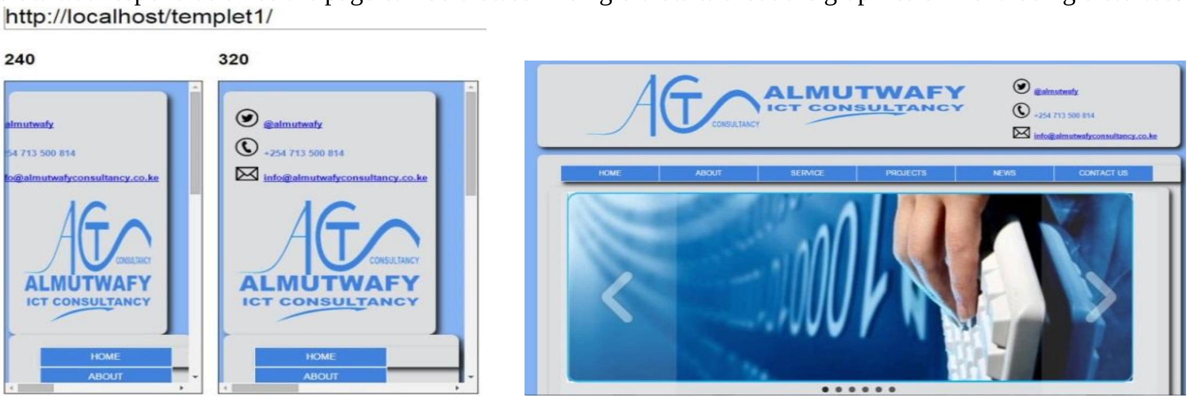


Figure 3: [a] Before implementing Algorithm [b] After implement Algorithm

Moreover, further responsive tests were conducted on the framework to simulate various screen sizes of; 240 x 320 pixels smart phone, 320 x 480 pixel iPhone, 480 x 640 pixels tablet, 768 x 1024 pixels iPad – Portrait and 1024 x 768 pixels iPad – Landscape. The Figures 4 [a], 4 [b], 4 [c], 4 [d] and 4 [e] showed that, the liquidizer.js framework is responsive at various screen sizes. The result can be explained by visual observation of the framework fitting in all screen size without being distorted

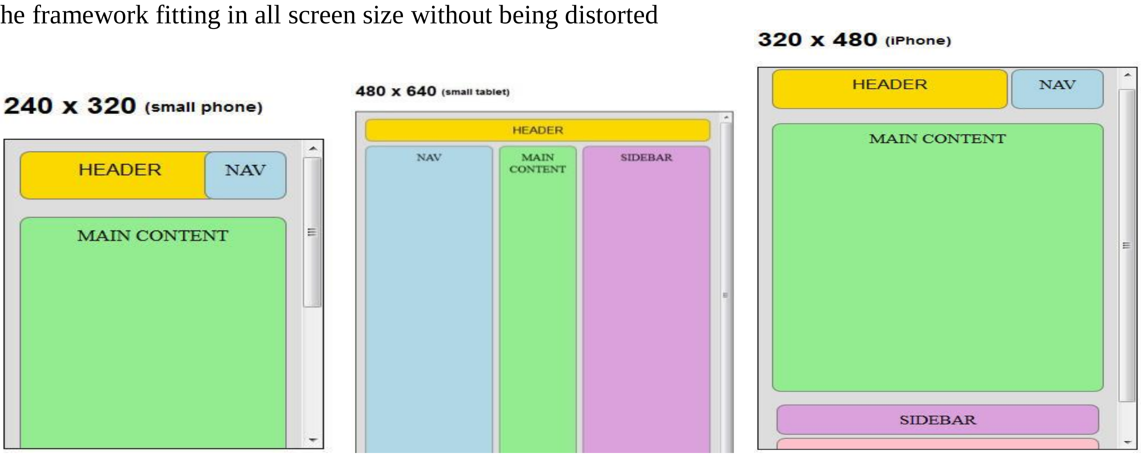
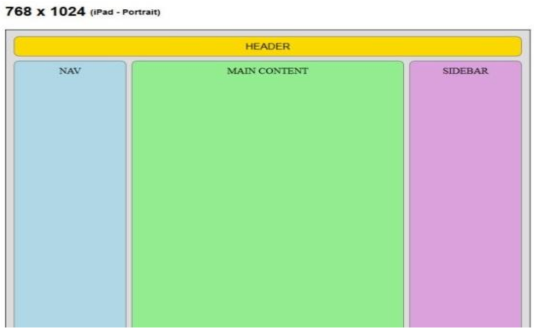
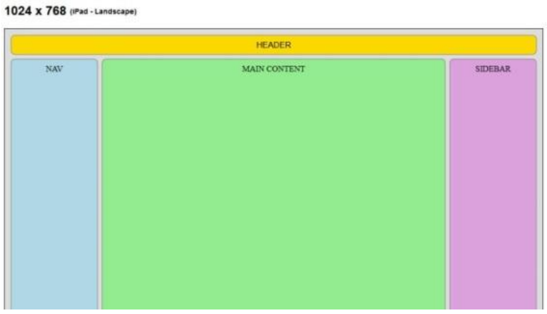


Figure 4: [a] Smart Phone [b] Small Tablet [c] iPad



[d] iPad-Portrait



[e] iPad-Landscape

4.2 Results for Liquidizer.js Validation

The study used Bersoft Image Measurement (BIM) tool to performed three validation analytical tests on the digital images generated by Liquidizer.js. The results are presented in threefold; the lane profile test, the pixel profile test and the descriptive statistics test.

4.2.1 The Lane Profile Results

The lane profile test analyses the pixel intensity against the pixel position. The lane profile was conducted on the framework generated by Liquidizer.js. The results are shown in Figure 5 below. It was observed from the results of four rows and columns pixel section of the digital image that the pixel position of each row and column divided in to 100px, corresponded to high pixel intensity of (R,G,B) (160, 160, 160) (light gray) which was fairly constant distributed at these regions. The result infers that, most of the pixels are highly dispersed at these ranges. The three colors; red, green and blue are evenly distributed hence displaying brighter quality picture since the range values of pixel intensity are from (0,0,0) (black) to (128,128,128) (gray) to (255, 255,255) (white). Therefore, the results can be explained by the visual observation of the non-distorted picture framework, which is responsive.

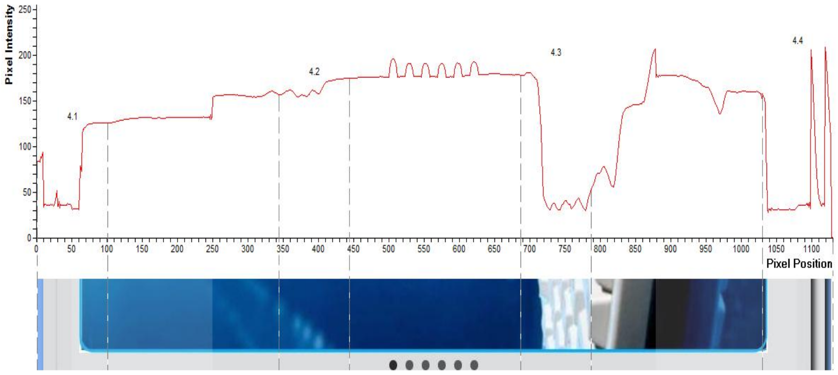


Figure 5: Liquidizer.js Lane Profile

4.2.2 The Pixel Profile Results

The pixel profile test evaluates the pixel values against the pixel distance of a digital image. The Liquidizer.js was applied to framework to generated digital images which were evaluated for pixel profile. The pixel profile was conducted on the framework generated by Liquidizer.js. The results are shown in Figure 6. It was observed from the results of the slice pixel section of the digital image that, the pixel distance of between 150px – 600px, corresponded to the pixel values of between 0 – 220, which are evenly distributed throughout the image. The result infers that, most of the pixels are fairly disperse over the region. The three colors; red, green and blue are highly distributed hence displaying good quality picture. Therefore, results can be explained by visual observation of the non-distorted picture which is responsive.

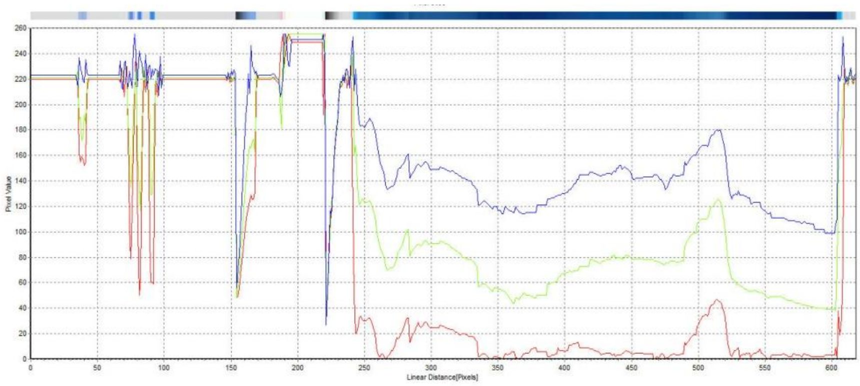
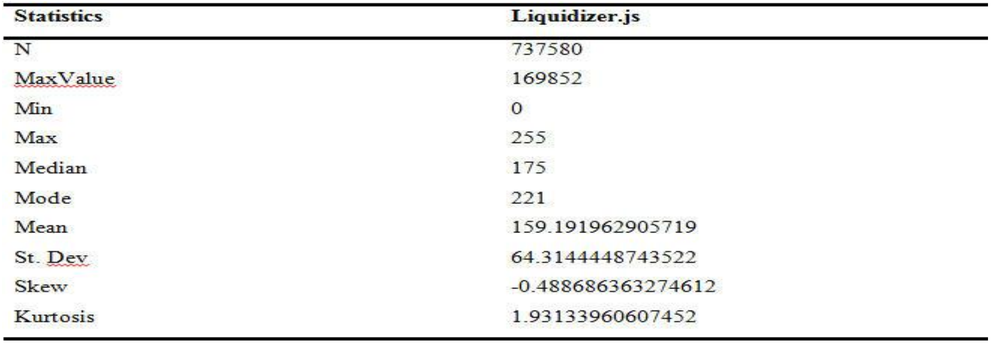


Figure 6: Liquidizer.js Pixel Profile

4.2.3 The Statistical Results

The descriptive statistics tests were conducted on the algorithm Liquidizer.js. The study evaluated and compared the skewness of the algorithm to ascertain the degree of distortion. The Skewness measure indicates the level of non-symmetry. If the distribution of the data is symmetric then Skewness will be close to 0 (zero). The further from 0, the more skewed the data. A negative value indicates a skew to the left. The Table 2 below shows various descriptive statistics test, the study had a particular interest on values of skewness. The results indicate that Liquidizer.js skewness value of -0.488686363274612 which is a score near to zero inferring to near perfectly normal distribution. Therefore, results can be explained by visual observation of the high degree of distorted picture quality of BlocksIt.js digital image as compared to Liquidizer.js digital image.

Table 2: Liquidizer.js Descriptive Statistics



Statistics	Liquidizer.js
N	737580
MaxValue	169852
Min	0
Max	255
Median	175
Mode	221
Mean	159.191962905719
St. Dev	64.3144448743522
Skew	-0.488686363274612
Kurtosis	1.93133960607452

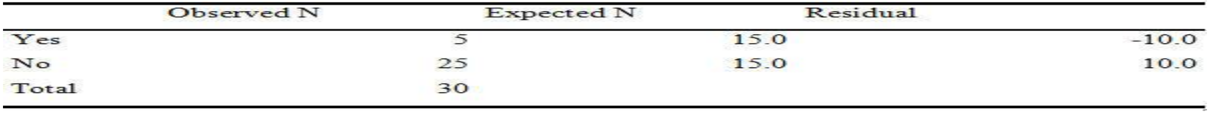
4.3 Results for Liquidizer.js Evaluation

The questionnaires were coded in SPSS to create a statistical model. The model was use to conduct various analytical and statistical tests on the information collected from web designer's perception of the Liquidizer.js algorithm.

4.3.1 Chi-Square Test

The survey sampled 30 respondents, and evaluated whether the number of respondents who use framework for design and development (f=5) was equal to the number of respondents do not use framework (f=25). The data was analyzed using a chi square goodness of fit test. The results are shown in the Table 3 Do you use framework for design and development.

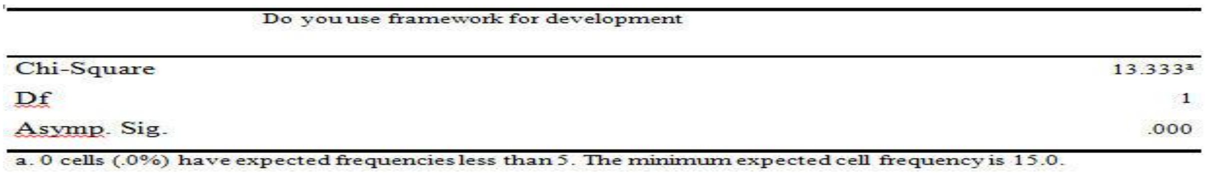
Table 3: Do you use framework for development



	Observed N	Expected N	Residual
Yes	5	15.0	-10.0
No	25	15.0	10.0
Total	30		

The null hypothesis (H_0) was crafted as; the number of respondents who use framework for development was equal to the number of respondents who do not use the framework. While the alternative hypothesis (H1) was crafted as; the number of respondents who use framework for development was not equal to the number of respondents who do not use the framework. The null hypothesis was rejected, $X^2(1) = 13.333$, $P < .05$. Seventy five percent (75%) of the respondent asserted that had not used the framework for development. The Table 4 demonstrates the chi-square test statistics of the construct

Table 4: Test Statistics



	Do you use framework for development
Chi-Square	13.333a
Df	1
Asymp. Sig.	.000

a. 0 cells (.0%) have expected frequencies less than 5. The minimum expected cell frequency is 15.0.

The results inferred that three quarters of the respondent do not use framework for development and design. The result concur with the report by [9], who asserted that web designers may not use fluid page designs for various reasons and further elaborated that, one of the reasons as being; images, video, and other types of content with set widths, need to be set at multiple widths to accommodate different screen resolutions.

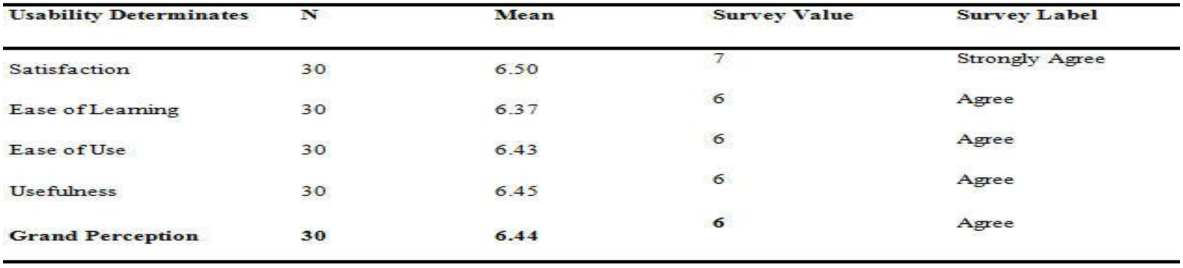
4.3.2 Survey Respondent Perception

Mean in statistical analysis and probability, are used to refer to measures of the central tendency either of a probability distribution or of the random variable characterized by that distribution. Therefore, the study sampled 30 respondents, and evaluated the distribution of the web designer's perception of the three dimensions of Usability; Usefulness, Satisfaction, Ease of Leaning (independent variables) and Ease of Use (dependent variable) of the Liquidizer.js. The data was analyzed using descriptive statistics by computing means of each Usability determinates. The means were then evaluated against the questionnaires approximate survey value coded to the survey labels, as shown in Table 5 Respondent's Perception of the survey. The responses were mapped to questionnaires as; 1 = "Strongly Disagree", 2 = "Disagree", 3 = "Slightly Disagree", 4 = "Neutral", 5 = "Slightly Agree", 6 = "Agree" and 7 = "Strongly Agree".

The findings showed that, the respondents asserted an average (mean = 6.50) for Satisfaction inferring to Survey Value (7), coded as "Strongly Agree" on the likert scale response, average (mean = 6.37) for Ease of Learning, inferring to Survey Value (6), coded as "Agree" response, average (mean = 6.43) for Ease of Use, inferring to Survey Value (6), coded as "Agree" response, and average (mean = 6.45) for Usefulness, inferring to Survey Value (6), coded as "Agree" response.

The inferential statistics indicate that, average (mean = 6.44) for Grand Perception of all Usability determinates, inferring to Survey Value (6), coded as "Agree" response. Therefore, it can be inferred that, respondents asserted a general agreement to the usability determinates of; Usefulness, Satisfaction, Ease of leaning (independent) which positively influence Ease of Use (dependent) of the Liquidizer.js, hence leading to a linear relationship.

Table 5: Respondent's Perception of the survey



Usability Determinates	N	Mean	Survey Value	Survey Label
Satisfaction	30	6.50	7	Strongly Agree
Ease of Learning	30	6.37	6	Agree
Ease of Use	30	6.43	6	Agree
Usefulness	30	6.45	6	Agree
Grand Perception	**30**	**6.44**	**6**	Agree

There are various contributing factors that may lead to linear relationship between variables, but more conclusive and affirmative tests such as correlation analysis should be consider for reliable output results

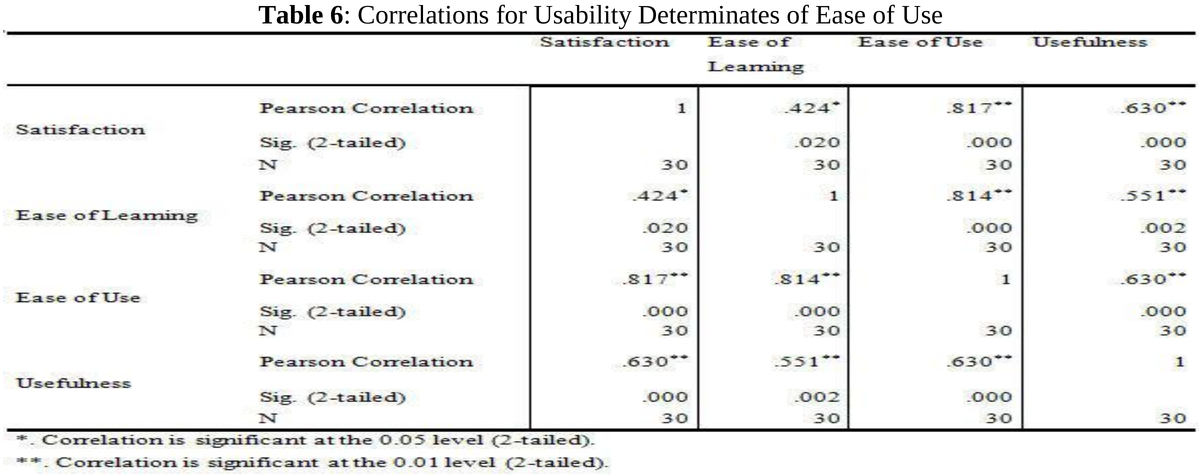
4.3.3 Correlation Analysis

Correlation refers to a technique used to measure the relationship between two or more variables. When two variables are correlated, it means that they vary together. Positive correlation means that high values on one are associated with high values on the other, and that low values on one are associated with values scores on the other. Correlation coefficient value is a numerical number between 0.0 and 1.0. The closer the correlation is to 1.0, the stronger the relationship between the two variables.

Therefore, the study conducted correlation analysis and computed four usability determinates of; Usefulness, Satisfaction, Ease of leaning (independent variables) which positively influence Ease of Use (dependent variable) of the Liquidizer.js for 30 respondents. The results are shown in Table 6 and suggested that, there was strong relationship of .817 between dependent variable Ease of Use and independent variable Satisfaction, which is statistically significant at r (30) = 0.817, p < .01, two-tailed. This infers that, the strong relationship between the variables was statistically significant at 99%. Moreover, the result further suggested that, there was a strong relationship of .814 between dependent variable Ease of Use and independent variable Ease of Learning at r (30) = 0.814, p < .01, two-tailed. This also infers that, the strong relationship between variables was statistically significant at 99%.

Finally, the result suggested an existence of a strong relationship of .630 between dependent variable Ease of Use and independent variable Usefulness at r (30) = 0.630, p < .01, two-tailed. This also infers that, the strong relationship between variables was statistically significant at 99%. In summary the result infers that, there was a strong positive linear relationship between the usability determinates of; Usefulness, Satisfaction, Ease of leaning (independent variables) and Ease of Use (dependent variable). The increase of Usability determinates, increases the Ease of Use. The Table 6 summarizes the correlation output computed by SPSS Version 20 for Usability Determinates of Ease of Use.

Table 6: Correlations for Usability Determinates of Ease of Use



		Satisfaction	Ease of Learning	Ease of Use	Usefulness
Satisfaction	Pearson Correlation	1	.424*	.817**	.630**
	Sig. (2-tailed)		.020	.000	.000
	N	30	30	30	30
Ease of Learning	Pearson Correlation	.424*	1	.814**	.551**
	Sig. (2-tailed)	.020		.000	.002
	N	30	30	30	30
Ease of Use	Pearson Correlation	.817**	.814**	1	.630**
	Sig. (2-tailed)	.000	.000		.000
	N	30	30	30	30
Usefulness	Pearson Correlation	.630**	.551**	.630**	1
	Sig. (2-tailed)	.000	.002	.000	
	N	30	30	30	30

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Therefore, it was clear from the discussion that correlation can only indicate the presence or absence of a relationship, not the nature of the relationship. Correlation is not causation. There is always the possibility that a third variable influenced the results. Therefore, other confirmatory tests such as regression should be conducted for reliable and conclusive results.

4.3.4 Regression Analysis

In statistics, regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. The regression analysis was modeled to measures how well one's overall model fits, and how well predictors; usability determinates of; Usefulness, Satisfaction, and Ease of Leaning are able to predict Ease of Use of Liquidizer.js. The linear regression analysis modeled the relationship between the dependent variable (Ease of Use) and independent variables (Usefulness, Satisfaction and Ease of Learning). Therefore, before conducting a reliable multiple regression analysis, it is a prerequisite to satisfy seven assumptions tests. These assumptions consist of tests for; outliers, collinearity of data, independent errors, random normal distribution of errors, homoscedasticity & linearity of data, and non-zero variances. If the data fails any of these assumptions test then one will need to investigate why and whether a multiple regression is really the best way to analyze the data. The study conducted the seven prerequisite assumption tests and the results satisfied all the conditions to conduct the regression analysis.

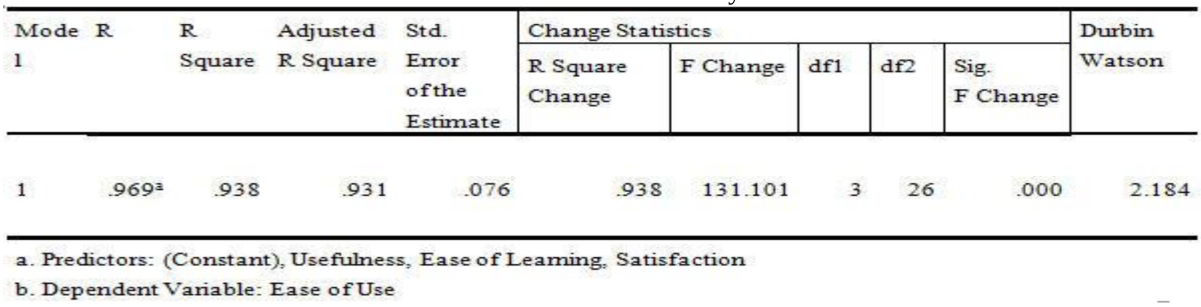
4.3.4.1 Reporting Multiple Regression

A multiple regression was conducted to see if the usability determinates of; Usefulness, Satisfaction, and Ease of Leaning are able to predict Ease of Use of Liquidizer.js. The study discuss the three main outputs of multiple regression analysis; Model Summary, ANOVA (Analysis Of Variance), and Coefficient of Multiple Regression

4.3.4.2 Model Summary

The regression model summary gives the measures of how well one's overall model fits, and how well predictors; Usefulness, Satisfaction, and Ease of Leaning, are able to predict Ease of Use of Liquidizer.js. The first measure in the Table 7 Model Summary is called R. This is a measure of how well predictors predict the outcome, but the study needed to take the square of R (R^2) to get a more accurate measure. This gives the study the amount of variance in Ease of Use of Liquidizer.js explained by the independent variables or predictors. The R^2 varies between 0 and 1. In the study R = .969, representing a 96.9% of the variance in Ease of Use of Liquidizer.js can be explained by the predictor variables above, although this does not imply causality. The final column gives us the standard error of the estimate. This is a measure of how much R is predicted to vary from one sample to the next. The R^2 is a statistic used in the context of statistical models whose main purpose is either the prediction of future outcomes or the testing of hypotheses, on the basis of other related information. An $R^2 = 1$ indicates that the regression line perfectly fits the data. In the Table 7 Model Summary, an $R^2 = .938$ indicates that the regression line moderately fits the data. Using the enter method it was found that the Usability determinates explain a significant amount of the variance in the value of Ease of Use of Liquidizer.js F (3, 26) = 131.101, p < .05, $R^2 = .938$, R^2 Adjusted = .931).

Table 7: Model Summary



Mode l	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics					Durbin Watson
					R Square Change	F Change	df1	df2	Sig. F Change	
1	.969[a]	.938	.931	.076	.938	131.101	3	26	.000	2.184

a. Predictors: (Constant), Usefulness, Ease of Learning, Satisfaction

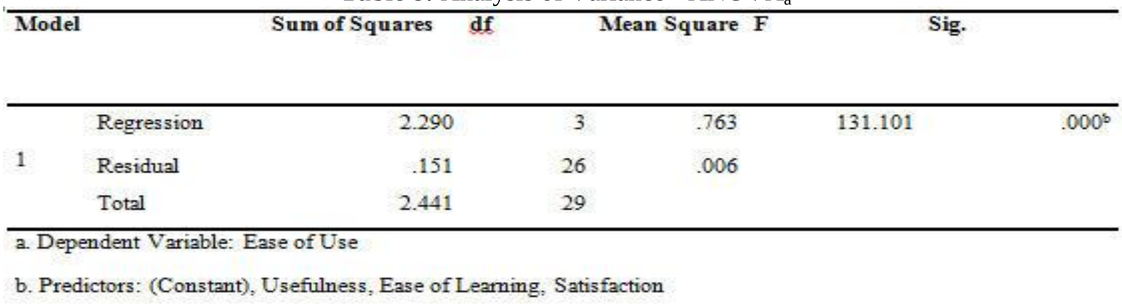
b. Dependent Variable: Ease of Use

4.3.4.3 ANOVA

Analysis of variance (ANOVA) is a collection of statistical models used to analyze the differences between group means and their associated procedures among and between groups. ANOVA provides a statistical test of whether or not the means of several groups are equal, and therefore generalizes the t-test to more than two groups. The F test (Fisher F distribution) is the ratio of two variances, which are used to determine if two variances are equal.

In the Table 8 ANOVA, the numerator df (3) tells how many predictors the study had (i.e. Usefulness, Ease of Learning and Satisfaction) and the denominator degrees of freedom (29 – 3 = 26) for bivariate regression use. The value of the F test in Table 8 ANOVA is F(3,26) = 131.101, (p < .05). This means the value of F is statistically significant at a level of 0.01, which suggests a linear relationship among the variables. The statistical significance at a 0.01 level means there is a 99 percent chance that the relationship among the variables is not due to chance.

Table 8: Analysis of Variance - ANOVA_a



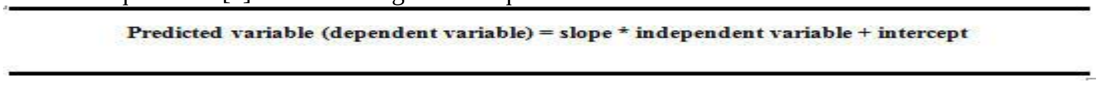
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	2.290	3	.763	131.101	.000[b]
	Residual	.151	26	.006		
	Total	2.441	29			

a. Dependent Variable: Ease of Use

b. Predictors: (Constant), Usefulness, Ease of Learning, Satisfaction

4.3.4.4 Coefficient

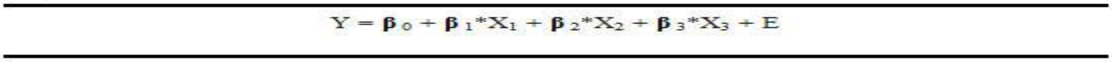
Regression coefficients represent the mean change in the dependent variable for one unit of change in the predictor variable while holding other predictors in the model constant. This statistical control that regression provides is important because it isolates the role of one variable from all of the others in the model. The regression coefficient is the slope of the regression line. It gives the information for writing the regression equation. The Equation 5 [a] shows the regression equation below.



Predicted variable (dependent variable) = slope * independent variable + intercept

Equation 5: [a] Regression Equation

The slope is how steep the line regression line is. A slope of 0 is a horizontal line, a slope of 1 is a diagonal line from the lower left to the upper right, and a vertical line has an infinite slope. The intercept is where the regression line strikes the Y axis when the independent variable has a value of 0. The Table 9 Coefficients presented the three predictor variables (Usefulness, Ease of Learning and Satisfaction). Therefore, a linear regression model with three predictor variables can be expressed as shown in Equation 5 [b].



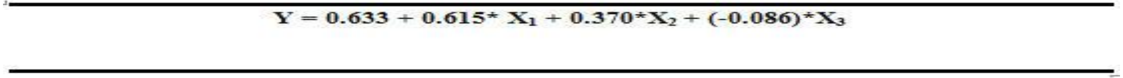
$$Y = \beta_0 + \beta_1 {*} X_1 + \beta_2 {*} X_2 + \beta_3 {*} X_3 + E$$

Equation 5: [b] Regression Equation

The variables in the model are Y, the dependent variable (Ease of Use); X_1, the first predictor independent variable (Usefulness); X_2, the second predictor variable (Ease of Learning); X_3, the third predictor variable (Satisfaction); and E, the residual error, which is an unmeasured variable.

The parameters in the model are $\boldsymbol{\beta}_0$, the Y-intercept (Constant = .633); $\boldsymbol{\beta}_1$, the first regression coefficient (Satisfaction = .615); $\boldsymbol{\beta}_2$, the second regression coefficient (Ease of Learning = -.370); and **β3**, the third regression coefficient (Usefulness = -.086).

The study modeled of the Ease of Use (Y) based on the Usability determinates in the Usefulness (X_1), Ease of Learning (X_2), and Satisfaction (X_3). Therefore, the regression equation was formulated as follows:

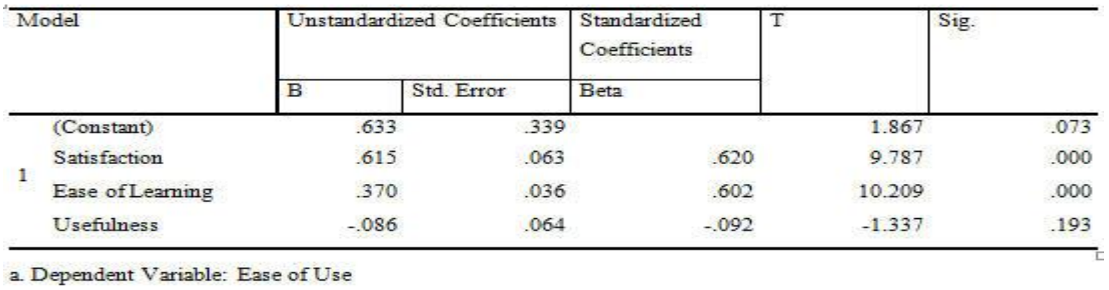


$$Y = 0.633 + 0.615^{*}\ X_1 + 0.370^{*}X_2 + (-0.086)^{*}X_3$$

Equation 5: [c] Regression Equation

The Y-intercept ($\boldsymbol{\beta}_0$), can be interpreted as the value one would predict for Y if $X_1 = 0$, $X_2 = 0$, $X_3 = 0$. The study would expect the Ease of Use of Liquidizer.js to be 0.633. However, this is only a meaningful interpretation if it is reasonable that if $X_1 = 0$, $X_2 = 0$, $X_3 = 0$, and if the dataset actually included values for if X_1, X_2, and X_3, that were near 0. If neither of these conditions is true, then $\boldsymbol{\beta}_0$ really has no meaningful interpretation. It just anchors the regression line in the right place.

Table 9: Coefficients



Model		Unstandardized Coefficients		Standardized Coefficients	T	Sig.
		B	Std. Error	Beta		
1	(Constant)	.633	.339		1.867	.073
	Satisfaction	.615	.063	.620	9.787	.000
	Ease of Learning	.370	.036	.602	10.209	.000
	Usefulness	-.086	.064	-.092	-1.337	.193

a. Dependent Variable: Ease of Use

Similarly, B_1 is interpreted as the difference in the predicted value in Y for each one-unit difference in X_1, if X_2 and X_3 remains constant. However, since X_1 is a categorical variable coded as; 1 = "Strongly Disagree", 2 = "Disagree", 3 = "Slightly Disagree", 4 = "Neutral", 5 = "Slightly Agree", 6 = "Agree" and 7 = "Strongly Agree" therefore, a one unit difference represents switching from one category to the other. B_1 is then the average difference in Y between the category for which $X_1 = 1$ (the reference group) and the category for which $X_1 = 2$, $X_1 = 3$, $X_1 = 4$, $X_1 = 5$, $X_1 = 6$ or $X_1 = 7$ (the comparison group).

Therefore, the study compared a one unit increase to Ease of Use of Liquidizer.js by computing different combination possibilities of the three predictors; Satisfaction, Ease of Learning and Usefulness. The inferential statistics of the variable Satisfaction was a game changer, since it is a single variable with a score value of 4.938. This value is approximately 5 presenting Slightly Agree in the likert scale. This implies that, for the Liquidizer.js was readily accepted by the web designers as a result of which, more efforts should be put on the variable Satisfaction to maximize the usability of the Liquidizer.js.

V. RECOMMENDATION

Even though the findings of the study showed that, the algorithm Liquidizer.js implementation was effectively responsive, an enhanced version is recommended whereby more dynamic attributes to be incorporated to increase acceptability by the web designers. Moreover, the study revealed that, the future development of the algorithm should emphasis on the usefulness of the web designers, since the variable usefulness contributes negatively for easy of usability and adaption. Finally, the evaluation sample used in the study consisted of a limited numbers of web designers due to budgetary and time constraints. These constraints may have introduced some biasness in the study's findings. Therefore, the study recommended for a more thorough global research in order to explore further the usability and adaptability of the algorithm Liquidizer.js by web designers.

REFERENCE

[1]. Allsopp, J. (2000, April 7). A Dao of Web Design. *A LIST APART*. Retrieved from http://alistapart.com/article/dao

[2]. Alsup, M. (2007, October 30). A Plugin Development Pattern » Learning jQuery - Tips, Techniques, Tutorials. Retrieved May 6, 2014, from http://www.learningjquery.com/2007/10/a-plugin-development-pattern/

[3]. BIM. (1995). Bersoft Image Measurement. Retrieved May 13, 2015, from http://image-measurement.com/

[4]. Frost, B. (2013). 7 Habits of Highly Effective Media Queries | Brad Frost Web. Retrieved September 18, 2014, from http://bradfrostweb.com/blog/post/7-habits-of-highly-effective-media-queries/

[5]. Hurb, E., Kapellari, P., Luong, S., & Spot, N. (2011, December). Survey Responsive Web Design.pdf. Retrieved from courses.iicm.tugraz.at/iaweb/.../g3-survey... Graz University of Technology

[6]. J. Eck, D. (2011a). *Introduction to Programming Using Java* (Sixth Edition, Vol. Version 6.0.3 January 2014). Retrieved from http://math.hws.edu/javanotes/c3/s2.html

[7]. J. Eck, D. (2011b). *Javanotes 6.0, Section 3.2 -- Algorithm Development* (Sixth Edition, Vol. Version 6.0.3 January 2014). Retrieved from http://math.hws.edu/javanotes/c3/s2.html

[8]. Kersley, M. (n.d.). Matt Kersley | Designer & Developer. Retrieved May 12, 2014, from http://mattkersley.com/

[9]. Knight, K. (2009, June 2). Fixed vs. Fluid vs. Elastic Layout: What's The Right One For You? | Smashing Magazine. Retrieved from http://www.smashingmagazine.com/2009/06/02/fixed-vs-fluid-vs-elastic-layout-whats-the-right-one-for-you/

[10]. Knight, K. (2011). Responsive Web Design: What It Is and How To Use It. Retrieved June 29, 2014, from http://www.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/

[11]. Lund, A. M. (2001). Measuring Usability with USE Questionnaire. *STC Usability SIG Newsletter*, *8*(2).

[12]. Olsen, D. (2013). RESS, Server-Side Feature-Detection and the Evolution of Responsive Web Design - dmolsen.com. Retrieved September 18, 2014, from http://dmolsen.com/2012/02/21/ress-and-the-evolution-of-responsive-web-design/

[13]. Smarty, A. (2013, July 15). A Look Into Responsive Typography. Retrieved from http://www.internetmarketingninjas.com/blog/design/a-look-into-responsive-typography/

[14]. Soueidan, S. (2013). Techniques for Responsive Typography. Retrieved September 18, 2014, from http://tympanus.net/codrops/2013/11/19/techniques-for-responsive-typography/

[15]. Tranfici, A. (2013, January 29). Responsive Web Design: Fluid Layouts. Retrieved from http://www.sitepoint.com/responsive-web-design-fluid-layouts/

[16]. Wroblewski, L. (2011). LukeW | RESS: Responsive Design + Server Side Components. Retrieved September 18, 2014, from http://www.lukew.com/ff/entry.asp?1392

[17]. Wroblewski, L. (2012, February 29). LukeW | Which One: Responsive Design, Device Experiences, or RESS? Retrieved May 7, 2014, from http://www.lukew.com/ff/entry.asp?1509