

**AN ENHANCED APPROACH TO  
RESPONSIVE WEB DESIGN INFLUID GRID CONCEPT**

**ABDULREHMAN AHMED MOHAMED**

A thesis submitted in partial fulfillment for the degree of Master of Science  
in Computer Systems in the Jomo Kenyatta University of Agriculture and  
Technology

2015

## DECLARATION

This thesis is my original work and has not been presented for a degree in any other University.

Signature..... Date .....

**Abdulrehman Ahmed Mohamed**

This thesis has been submitted for examination with my approval as the University Supervisors.

Signature ..... Date .....

**Dr. Cheruiyot W.K, PhD**  
**JKUAT, Kenya**

Signature ..... Date .....

**Dr. Richard Rimiru, PhD**  
**JKUAT, Kenya**

## DEDICATION

This piece of work is dedicated to: my dear parents, my late father Mr. Ahmed Mohamed Almutwafy and my mother Mrs. Amana Mohamed Mzee, for all that they have sacrificed to raise me up and direct me in the right path of life. I really salute your dedication. My further dedication also goes to my family; my wife Mrs. Inat Ahmed Abdalla, our three sons; Ahmed, Abdilnoor and Bilkheir and three daughters; Khairat, Noorat and Rayaan for all the material and moral support, to whom I'll soon hand over the mentorship with this word of encouragement, "Whatever The Mind Can Conceive And Believe, It Can Achieve" *Napoleon Hill*.

## ACKNOWLEDGEMENT

I would like to acknowledge and appreciate the material and moral support that I have always received from my family, colleagues and friends. Special thanks to my supervisors; Dr. Cheruiyot W.K. for his kind words of encouragement and guidance and Dr. Richard Rimiru for his words of motivation “*Stop thinking do it*”. My sincere appreciation goes to my lecturers at the School of Computing and Information Technology (SCIT) at the Jomo Kenyatta University of Agriculture and Technology (JKUAT), Mombasa Central Business District (CBD) Campus for their support and fair criticism of this work so far. Special thanks to my Associate Chair of SCIT Mr. Collins Ondago for the kind words of encouragement and guidance, throughout this program, and Mr. Samuel Kyang’anda for his persistence critique and review of this document. I salute all of you for your time and efforts. My further sincere appreciation also goes to my family for their psychological and financial support. Special thanks to my wife; Inat Ahmed Abdalla, and our sons; Ahmed, Abdulnoor and Bilkheir, and our daughters; Khairat, Noorat and Rayaana for always believing in me, encouraging and pushing me to strive for even higher levels. To the only one and true God (Allah), for the gift of life and wisdom. To all the above mortal men, may God (Allah) richly bless your work

## TABLE OF CONTENTS

<b>DECLARATION</b> .....	<b>ii</b>
<b>DEDICATION</b> .....	<b>iii</b>
<b>ACKNOWLEDGEMENT</b> .....	<b>iv</b>
<b>LIST OF TABLES</b> .....	<b>x</b>
<b>LIST OF FIGURES</b> .....	<b>xii</b>
<b>LIST OF EQUATION</b> .....	<b>xiv</b>
<b>LIST OF APPENDICES</b> .....	<b>xv</b>
<b>LIST OF ABBREVIATIONS</b> .....	<b>xvi</b>
<b>ABSTRACT</b> .....	<b>xvii</b>
<b>CHAPTER ONE</b> .....	<b>1</b>
<b>1.0INTRODUCTION</b> .....	<b>1</b>
1.1    Background .....	1
1.1.1    The First Web Pages .....	1
1.1.2    Formation of World Wide Web Consortium .....	2
1.1.3    Table-based Designs .....	2
1.1.4    Design Over Structure.....	3
1.1.5    Flash-based Web Designs .....	3
1.1.6    Cascading Style Sheets (CSS)-based Web Designs.....	3
1.1.7    Responsive Web Design .....	4
1.1.8    Types of Web Layouts Approaches .....	5
1.1.9    Fixed-Width Layouts .....	6
1.1.10    Liquid Layouts .....	6
1.1.11    Elastic Layouts .....	6
1.1.12    Web Development Technologies .....	7

1.1.13	jQuery .....	7
1.1.14	Ajax (Asynchronous JavaScript and XML) .....	8
1.1.15	Cascading Style Sheets (CSS).....	8
1.1.16	HTML and Extensible Markup Language.....	8
1.1.17	Webkit .....	9
1.2	Statement of the Problem .....	9
1.3	Justification .....	10
1.4.0	Objectives.....	10
1.4.1	Broad objective .....	10
1.4.2	Specific objective .....	10
1.5	Research Questions .....	11
1.6	Conceptual Framework .....	11
<b>CHAPTER TWO .....</b>		<b>13</b>
<b>2.0</b>	<b>LITERATURE REVIEW.....</b>	<b>13</b>
2.1	Introduction .....	13
2.2	Theoretical Literature Review .....	14
2.2.1	Technologies of RWD.....	16
2.2.2	Types of Web Layouts Approaches .....	20
2.2.2.1	Fixed-Width Layouts.....	20
2.2.2.2	Liquid Layouts .....	21
2.2.2.3	Elastic Layouts .....	21
2.2.2.4	Fluid Grid Concept .....	21
2.3.0	Related work.....	23
2.3.1	Frame-based solution .....	23
2.3.2	Support-based solution .....	24
2.3.3	Algorithm-based solution .....	26
2.4	Usability Models .....	27
2.4.1	Usability Model Types .....	29

2.4.1.1 Usability in ISO 9241 .....	29
2.4.1.2 Usability in Factors Criteria Metrics (FCM) Model .....	30
2.4.1.3 Usability in ISO 9126 Model .....	31
2.5 Recommendation and Conclusion .....	32
<b>CHAPTER THREE .....</b>	<b>34</b>
<b>3.0 RESEARCH METHODOLOGY .....</b>	<b>34</b>
3.1 Introduction .....	34
3.2 Research Design.....	36
3.2.1 Experimental Research Design.....	36
3.3.1 Accessing BlocksIt.js .....	37
3.3.2 Accessing Liquidizer.js .....	38
3.4.0 Design Rationale .....	39
3.4.1 Algorithm Development Methodologies.....	40
3.4.2 Flowcharts .....	40
3.4.3 Pseudo-code .....	43
3.4.4 Stepwise Refinement.....	44
3.4.5 Programming Language .....	45
3.5 Algorithm Development.....	46
3.5.1 Liquidier.js Description.....	47
3.5.2 How Liquidier.js Works.....	48
3.5.3 How to Use Liquidier.js .....	48
3.5.4 Liquidier.js Configuration.....	49
3.5.5 Liquidier.js Configuration Example.....	49
3.6 Framework Validation and Verification .....	51
3.6.1 Evaluation Method.....	52
3.7.0 Analysis and Testing Tools.....	53
3.7.1 Responsivepx .....	53
3.7.2 Matt Kersley RWD Testing Tool.....	55

3.7.3	Algorithm Liquidizer.js Validation.....	56
3.7.4	Algorithm Registration for jQuery Library.....	57
3.8	Liquidizer.js Repository .....	58
<b>CHAPTER FOUR.....</b>		<b>59</b>
<b>4.0 RESULTS FINDINGS.....</b>		<b>59</b>
4.1	Introduction.....	59
4.2	TAM Satisfaction Questionnaire.....	59
4.3	Measuring Usability with the USE Questionnaire .....	60
4.4	Data Collection.....	62
4.5	Data Analysis .....	63
4.6	Findings .....	64
4.7	Results for Liquidizer.js Testing .....	64
4.8	Results for Liquidizer Validation .....	66
4.8.1	Lane Profile Test.....	66
4.8.1.1	Pixel Intensity.....	66
4.8.1.2	Pixel Position.....	67
4.8.1.3	Lane Profile Results .....	67
4.8.2	Pixel Profile Test.....	70
4.8.2.1	Pixel Values.....	70
4.8.2.2	Pixel distance.....	70
4.8.2.3	Pixel Profile Results .....	70
4.8.3	Results for Liquidizer.js Evaluation.....	72
4.8.3.1	Statistical Results .....	72
4.9	Evaluation Results by SPSS .....	74
4.9.1	Reliability Analysis.....	74
4.9.2	Demographic Data .....	75
4.9.2.1	Position held at the organization.....	75
4.9.2.2	Duration of stay in the organization .....	76



4.9.2.3 Number of Project Developed .....	76
4.9.2.4 Chi-Square Test .....	77
4.9.2.5 Reason not using a framework.....	78
4.9.2.6 Survey Respondent Perception .....	79
4.9.3 Correlation Analysis .....	80
4.9.4 Regression Analysis.....	82
4.9.4.1 Outliers Test.....	83
4.9.4.2 Collinearity Test .....	83
4.9.4.3 Independent Errors Test.....	84
4.9.4.4 Random Normally Distributed Errors Test.....	85
4.9.4.5 Homoscedasticity & Linearity Test .....	87
4.9.4.6 Non-Zero Variances Test.....	87
4.9.5 Reporting Multiple Regression .....	88
4.9.5.1 Model Summary .....	88
4.9.5.2 ANOVA.....	89
4.9.5.3 Coefficient .....	90
4.9.5.4 Interpreting the Intercept .....	91
4.9.5.5 Interpreting Coefficients of Categorical Predictor Variables .....	92
<b>CHAPTER FIVE.....</b>	<b>94</b>
<b>CONCLUSIONS AND RECOMMENDATIONS.....</b>	<b>94</b>
5.0 Introduction.....	94
5.1 Summary .....	94
5.2 What are existing technologies in RWD.....	95
5.2.1 What are the methodologies for algorithm design.....	95
5.2.2 What are the techniques for algorithm development.....	95
5.2.3 What are the tools for algorithm testing, validation and evaluation .....	96
5.2.5 What are the requirements for algorithm registration in jQuery .....	96
5.3 Recommendations.....	96

5.4 Further Research .....	97
REFERENCE.....	98
APPENDICES .....	105

## LIST OF TABLES

<b>TABLE 1: CRESWELL’S RECOMMENDATIONS FOR MATERIAL</b> .....	14
<b>TABLE 2: SOURCE: ISO 9126 CHARACTERISTIC AND SUB-CHARACTERISTICS</b> .....	32
<b>TABLE 3: BLOCKIT DEFAULT OPTIONS ADAPTED FROM (KENNY, 2012)</b> .....	37
<b>TABLE 4: THE WEB DESIGN FIRMS AND WEB PAGES TESTED</b> .....	52
<b>TABLE 5 : DESCRIPTIVE STATISTICS</b> .....	73
<b>TABLE 6: RELIABILITY STATISTICS</b> .....	75
<b>TABLE 7: POSITION HELD AT THE ORGANIZATION</b> .....	75
<b>TABLE 8: DO YOU USE FRAMEWORK FOR DEVELOPMENT</b> .....	77
<b>TABLE 9: TEST STATISTICS</b> .....	78
<b>TABLE 10: WHAT ARE REASONS OF NOT USING A FRAMEWORK?</b> .....	79
<b>TABLE 11: RESPONDENT’S PERCEPTION OF THE SURVEY</b> .....	80
<b>TABLE 12: CORRELATIONS FOR USABILITY DETERMINATES OF EASE OF USE</b> .....	82
<b>TABLE 13: RESIDUALS STATISTICS<sup>A</sup></b> .....	83
<b>TABLE 14: COEFFICIENTS<sup>A</sup></b> .....	84
<b>TABLE 15: MODEL SUMMARY</b> .....	85

**TABLE 16:** DESCRIPTIVE STATISTICS..... 88

**TABLE 17:** ANALYSIS OF VARIANCE - ANOVA<sub>A</sub> ..... 90

**TABLE 18:** COEFFICIENTS ..... 91

**TABLE 19:** USABILITY PREDICTION OF LIQUIDIZER.JS ..... 92

## LIST OF FIGURES

<b>FIGURE 1:</b> EVOLUTION OF WEB DESIGN ADAPTED FROM (BUKDAT, 2012).....	1
<b>FIGURE 2:</b> PROPORTION OF A 300PX ELEMENT, ADAPTED FROM (PETTIT, 2012) .....	23
<b>FIGURE 3:</b> PX TO EM CONVERSION MADE SIMPLE ADAPTED FROM (CRAY, 2012B) .....	25
<b>FIGURE 4:</b> DYNAMIC GRID LAYOUT BLOCKS. ADAPTED FROM (KENNY, 2012) .....	27
<b>FIGURE 5:</b> LIQUIDIZER.JS CONCEPTUAL FRAMEWORK.....	12
<b>FIGURE 6:</b> USE-CASE DIAGRAM INTERFACE FOR BLOCKIT.JS.....	38
<b>FIGURE 7:</b> FIGURE 3.3: USE-CASE DIAGRAM INTERFACE FOR LIQUIDIZER.JS .....	39
<b>FIGURE 8:</b> PROCESSING RESPONSIVE WEB USING BLOCKIT.JS FRAMEWORK .....	41
<b>FIGURE 9:</b> PROCESSING RESPONSIVE WEB USING LIQUIDIZER.JS FRAMEWORK.....	42
<b>FIGURE 10:</b> THE LIQUIZER.JS LAYOUT OUTPUT .....	51
<b>FIGURE 11:</b> USABILITY MODEL, SOURCE: (NIELSEN, 1994) .....	30
<b>FIGURE 12:</b> USABILITY FCM MODEL, SOURCE: (MCCALL, 1977).....	31
<b>FIGURE 13:</b> INTERFACE OF RESPONSIVEPX. ....	54
<b>FIGURE 14:</b> LOCAL HOST WEBSITE DEMONSTRATION .....	54
<b>FIGURE 15:</b> MATT KERSLEY DEFAULT INTERFACE .....	55
<b>FIGURE 16:</b> LOCAL HOST WEBSITE DEMONSTRATION .....	56
<b>FIGURE 17:</b> THE BIM INTERFACE ADAPTED FROM (BIM, 1995) .....	57

**FIGURE 18:** A) BEFORE IMPLEMENTING ALGORITHM B) AFTER ..... 65

**FIGURE 19:** A) SMART PHONE B) SMALL TABLET C) IPAD ..... 65

**FIGURE 20:** STANDARD COORDINATE FORMAT ..... 67

**FIGURE 21:** FIGURE 3: A) BLOCKSIT.JS LANE PROFILE..... 68

**FIGURE 22:** A) BLOCKSIT.JS PIXEL PROFILE ..... 71

**FIGURE 23:** DURATION OF STAY IN THE ORGANIZATION ..... 76

**FIGURE 24:** NUMBER OF PROJECT DEVELOPED..... 77

**FIGURE 25:** HISTOGRAM REGRESSION STANDARDIZED RESIDUAL..... 86

**FIGURE 26:** NORMAL P-P PLOT OF REGRESSION STANDARDIZED RESIDUAL ..... 86

**FIGURE 27:** SCATTER PLOT OF REGRESSION STANDARDIZED PREDICTED VALUE. 87

## LIST OF EQUATION

<b>EQUATION 1:</b> BLOCKIT.JS INTERFACE ADAPTED FROM (KENNY, 2012).....	26
<b>EQUATION 2:</b> PSEUDO-CODE VERSION OF LIQUIDIZER ALGORITHM.....	45
<b>EQUATION 3:</b> REGRESSION EQUATION.....	90

## **LIST OF APPENDICES**

<b>Appendix 1:</b> Cost and Material Estimate.....	106
<b>Appendix 2:</b> Gantt Chart .....	106
<b>Appendix 3:</b> Questionnaire .....	107
<b>Appendix 4:</b> Introductory Letter for Data Collection .....	111
<b>Appendix 5:</b> Publication .....	113

## **LIST OF ABBREVIATIONS**

<b>ABS</b>	Algorithm-based Solution
<b>CSS</b>	Cascading Styling Sheets
<b>DOM</b>	Document Object Model
<b>FBS</b>	Frame-based Solution
<b>HTML</b>	Hyper Text Markup Language
<b>jQuery</b>	Javascript Query
<b>RWD</b>	Responsive Web Design
<b>SBS</b>	Support-based Solution
<b>ISO</b>	International Standards Organization
<b>IEEE</b>	Institute of electrical and electronic Engineers



## ABSTRACT

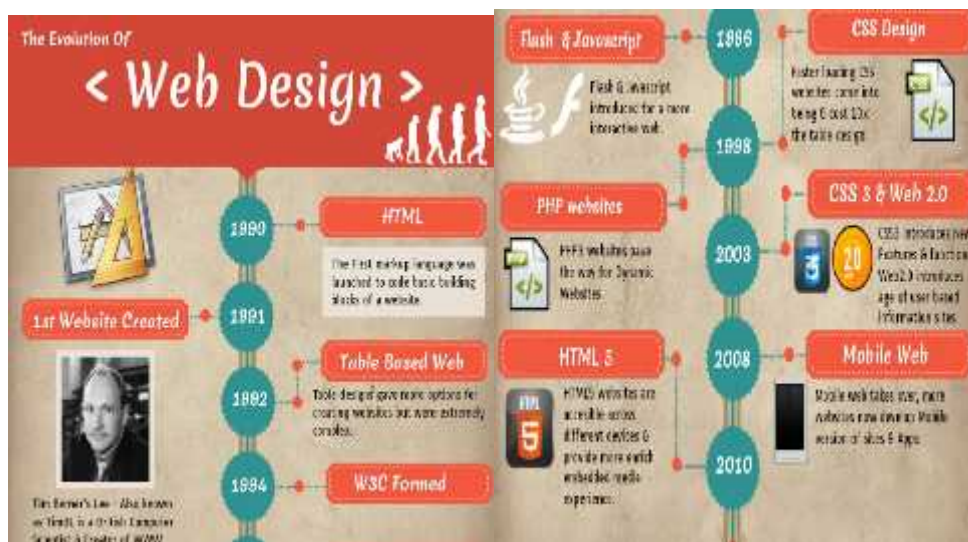
Web design is a dynamic field which evolves around user viewing experience trends. The recent trends have revolutionized the web design approach technologies. The result of which, Responsive Web Design (RWD) concept was coined by Web Designer Ethan Marcotte in 2010. RWD is an approach of new paradigms and techniques to develop one single website which looks different for different screen sizes so that it is usable on any access web device. The core concepts of RWD are; media query concept, fluid grid concept, and fluid image concept. The study was inspired by the fluid grid concept. Lots of ink has been spilled on the concept, especially in areas of conversion of website's fixed grid layouts to fluid grid layouts. However, these approaches involved heavy customization, resulting into slow adaption by web designers. Therefore, the study's objective was to address this gap by implementing an alternative approach that will be non-customizable and accessible by a single line of code. Hence, study presented an enhanced approach to RWD by implementing an algorithm coined Liquidizer.js in a jQuery Framework. The study used experimental research design and purposive sampling to sample the target population. Moreover, the study employed Matt Kersley RWD Tool: for testing Liquidizer.js, Bersoft Image Measurement (BIM) tool: for validating Liquidizer.js, and USE questionnaire for measuring usability: for evaluation of the Liquidizer.js algorithm in SPSS. The outcome of the study showed the pixel profile results of BlockIt.js indicated that the image is of low quality and distorted as compared to that of Liquidizer.js indicating the image of high quality and non-distorted. The results inferred that Liquidizer.js framework is more responsive as compared to BlockIt.js.

# CHAPTER ONE

## 1.0 INTRODUCTION

### 1.1 Background

Since the first websites in the early 1990s, designers have been experimenting with the way websites look. Early sites were entirely text-based, with minimal images and no real layout to speak of, other than headings and paragraphs. However, the industry progressed, eventually bringing table-based designs, Flash, then Cascading Style Sheets (CSS)-based designs and finally Responsive Web Design (RWD). The Figure 1 below summarizes the evolution of web designs.



**Figure 1:** Evolution of Web Design Adapted from (Bukdat, 2012)

#### 1.1.1 The First Web Pages

(Chapman, 2009), describes the history of the different eras of web design. She explains that, in August 1991, Tim Berners-Lee published the first website, a simple, text-based page with a few links. A copy from 1992 of the original page still exists online at <http://www.w3.org/History/19921103hypertext/hypertext/WWW/TheProject.html>. It had a dozen or so links, and simply served to tell people what the World Wide Web (WWW) was all about. Subsequent pages were similar, in that they were

entirely text-based and had a single-column design with inline links. Initial versions of Hyper Text Markup Language (HTML) only allowed for very basic content structure: headings, paragraphs, and links. Subsequent versions of HTML allowed the addition of images to pages, and eventually support for tables was added.

### **1.1.2 Formation of World Wide Web Consortium**

In 1994, the World Wide Web Consortium (W3C) was established, and they set HTML as the standard for marking up web pages. This discouraged any single company from building a proprietary browser and programming language, which could have had a detrimental effect on the web as a whole. The W3C continues to set standards for open web markup and programming languages (such as JavaScript).

### **1.1.3 Table-based Designs**

(Wavik, 2009), explains that, tables generally increase the complexity of documents and make them more difficult to maintain. Also, they reduce a website's flexibility in accommodating different media and design elements, and they limit a website's functionality. MAMA (Metadata Analysis and Mining Application) is a structural Web page search engine from Opera Software that crawls, Web pages and returns results detailing page structures. If one looks into MAMA's key findings, one sees that the average website has a table structure nested three levels deep. On the list of 10 most popular tags, table, td and tr are all there. The table element is found on over 80% of the pages whose URLs were crawled by MAMA. Semantically speaking, the table tag is meant for **listing tabular data**. It is not optimized to build structure.

It is explained by (Chapman, 2009) that, table-based layouts gave web designers more options for creating websites. The original table markup in HTML was meant for displaying tabular data, but designers quickly realized they could utilize it to give structure to their designs, and create more complicated, multi-column layouts than HTML was originally capable of. Table-based designs grew in complexity, incorporating sliced-up background images, often giving the illusion of a simpler structure than the actual table layout.

#### **1.1.4 Design Over Structure**

(Bukdat, 2012) elaborates that, the era of web layouts paid little attention to semantics and web accessibility, often opting for aesthetics over good markup structure. This was the same era where Spacer Graphics Interchange Formats (GIFs) were popularized to control whitespace of web layouts. The development of the first What You See Is What You Get (WYSIWYG) web design applications, all of which used table-based layouts, increased the use of tables. In addition, some of those programs created tables so complex that many designers would never have created them from scratch (such as tables with rows only 1-pixel high and hundreds of cells). Designers had to rely on tables if they wanted to create designs that were even mildly complex (such as multi-column designs).

#### **1.1.5 Flash-based Web Designs**

(Bukdat, 2012) explains that, Flash (originally known as Future Splash Animator, then Macromedia Flash, and currently as Adobe Flash) was developed in 1996. It started with very basic tools and a timeline, and progressed to have powerful tools to develop entire sites. Flash presented a ton of options beyond what was possible with HTML.

Around the same time as the introduction of Flash to the scene of web design (late 1990 s – early 2000 s), the popularization of Dynamic Hypertext Markup Language (DHTML) techniques, which consisted of several web technologies such as JavaScript and sometimes server-side scripting, for creating interactive/animated page elements were also the rage. During this time, with the inception of Flash and the popularity of DHTML, the concept of interactive web pages that allow users to not only read static content, but also to interact with web content, began.

#### **1.1.6 Cascading Style Sheets (CSS)-based Web Designs**

According (Beal, 2013), CSS is a feature added to HTML that gives both Web site developers and users more control over how pages are displayed. With CSS, designers and users can create style sheets that define how different elements, such

as headers and links, appear. These style sheets can then be applied to any Web page. The term *cascading* derives from the fact that multiple style sheets can be applied to the same Web page. CSS was developed by the World Wide Web Consortium (W3C). The Cascading Style Sheets (CSS)-based designs started gaining in popularity after the dotcom boom in the early 2000 s. While CSS had been available long before then, there was limited support for it in major browsers and many designers were unfamiliar with it (and even intimidated by it).

According to (Chapman, 2009), CSS-based designs have many advantages over table-based or Flash designs. The first is that it separates design elements from content, which ultimately meant that there would be greater distinction from the visual aspect of a web layout and its content. CSS is also a best practice for laying out a web page, where table-based layouts are not. It also reduced markup clutter and made for cleaner and semantic web layouts. CSS also makes it easier to maintain sites, as the content and design elements are separated. One can change the entire look of a CSS-based site without ever having to touch the content. The document sizes of CSS designs are generally smaller than table-based designs too, which translated to an improvement in page response times. Although there would be an initial bandwidth hit when first downloading the style-sheets of a website one never visited before, CSS is cached by the user's browser (by default) so that subsequent page views would be faster-loading.

### **1.1.7 Responsive Web Design**

It is asserted by (Marcotte, 2010) that, RWD is a web design approach aimed at crafting sites to provide an optimal viewing experience; easy reading and navigation with a minimum of resizing, panning, and scrolling; across a wide range of devices (from mobile phones to desktop computer monitors). A site designed with RWD adapts the layout to the viewing environment by using fluid, proportion-based grids, flexible images, and CSS media queries, an extension of the @media rule. Responsive Web Design (RWD) was founded by Ethan Marcotte who is a developer and a web designer. He had a particular interest in architecture and he wanted to

apply architectural principals in web design. Inspired by architecture way of thinking, he applied the architectural principal to web design; whereby a web site would adapt itself to the users various devices (smart phones, laptops, desktop e.t.c.). Hence, the idea of Responsive Web Design was coined

It is suggested by (Doyle, 2011) that, the following four core concepts to be apply for RWD implementation:

- a. **The fluid grid concept** – which calls for page element sizing to be in relative units like percentages, rather than absolute units like pixels or points.
- b. **Flexible images** – which calls also for sized in relative units, so as to prevent them from displaying outside their containing element.
- c. **Media queries** – which allows the page 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.
- d. **RESS(Responsive Web Design + Server Side Components)** – which 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/usability avoiding some of the pitfalls of device-side-only solutions.

It is described by (Doyle, 2011) that, another name used to describe this set of techniques is Adaptive Web Design (AWD). The name would match more since the website really adapts to the device, rather than responding continuously to changes in its environment.

### **1.1.8 Types of Web Layouts Approaches**

According to (Tranfici, 2013), classifies the web layouts as fixed-width layouts, liquid layouts, and elastic layouts.

### **1.1.9 Fixed-Width Layouts**

It is described by (Tranfici, 2013) that, in fixed-width layouts, the width of the site is bound to a certain number of pixels. Generally, the measure chosen is 960 pixels. This is because with the passing of time, developers have found 960 pixels to be the best size for grid layouts, because the number is easily divisible by 3, 4, 5, 6, 8, 10, 12, and 15. However, the fixed-width layouts have some disadvantages. The designers who want to create a fixed-width website have to keep in mind that every aspect of their work has to be usable and clearly visible to a large number of screens, browsers, and devices. The wide variety of devices on the market at this time, as well as the consequently great variability of screen sizes makes creating one-size-fits-all content quite a challenging task, and arguably a challenge that outweighs the precision and control of fixed-width design.

### **1.1.10 Liquid Layouts**

(Tranfici, 2013) explains that, the first basic difference between the fixed-width type of layout and liquid layouts is the measurements of their size. The fixed-width layouts are measured in pixels, but liquid or fluid layouts, dimensions are defined in percentages, and as expected, this affords greater malleability and fluidity. In other words, by setting a percentage, one won't have to think about device size or screen width, and consequently, one can find a reasonable solution for each case because the design's size will adapt to the size of the device used. Liquid layouts are closely linked to media queries and special styles for optimization. Percentage-based widths alone will likely not be enough to accommodate one's design for a large variety of display sizes.

### **1.1.11 Elastic Layouts**

It is elaborated by (Tranfici, 2013) that, elastic layouts are somewhat similar to liquid layouts. The main difference is once again the unit of measurement for size. The size indicator for elastic layouts is neither in pixels nor percentages; it's measured in *ems*. An *em* is the equivalent of the size (in pixels) defined in the *font-size* CSS rule. For example, if one styles text with a *font-size* of 20 pixels, 1 *em* would be equal to 20

pixels, 2 ems would correspond to 40, and so on. This type of layout gives the developer strong typographic control. Since the vast majority of layouts are predominantly populated with text, the precision of type treatments makes elastic layouts a strong contender for many projects. However, even with this type of solution, there is a risk of an unpleasant and unaesthetic horizontal scroll bar in some rare cases. The study is motivated to explore the fluid grid concept.

There exist various web development languages for implementation of responsive web design concepts. It is of paramount importance to discuss the strength and weakness of these languages, in order to present the most suitable one for Liquidizer.js implementation.

#### **1.1.12 Web Development Technologies**

According to (Y. Na, 2012), classifies the most common used technologies for web design are as follows:

#### **1.1.13 jQuery**

According to (John, 2006), jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

It is described by (Narayan, 2011) that, jQuery is very compact and well written JavaScript code that increases the productivity of the developer by enabling them to achieve critical UI functionality by writing very small amount of code. It is a lightweight cross-browser JavaScript library. jQuery emphasizes interaction between JavaScript and HTML. It is used by 27% of the 10,000 most visited websites, jQuery is the most popular JavaScript library currently in use.



#### **1.1.14 Ajax (Asynchronous JavaScript and XML)**

According to (Rouse, 2007), Ajax (Asynchronous JavaScript and XML) is a method of building interactive applications for the Web that process user requests immediately. Ajax combines several programming tools including JavaScript, dynamic HTML (DHTML), Extensible Markup Language (XML), cascading style sheets (CSS), the Document Object Model (DOM), and the Microsoft object, XMLHttpRequest. Ajax allows content on Web pages to update immediately when a user performs an action, unlike an HTTPrequest, during which users must wait for a whole new page to load.

It urged by (Fote, 2013) that, Ajax is not a programming language or a tool, but a concept. Ajax is a client-side script that communicates to and from a server/database without the need for a postback or a complete page refresh. Hence can be defined as; the method of exchanging data with a server, and updating parts of a web page - without reloading the entire page. Ajax itself is mostly a generic term for various JavaScript techniques used to connect to a web server dynamically without necessarily loading multiple pages. In a more narrowly-defined sense, it refers to the use of XMLHttpRequest objects to interact with a web server dynamically via JavaScript.

#### **1.1.15 Cascading Style Sheets (CSS)**

It described by (Keio, 2013) that, CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments.

#### **1.1.16 HTML and Extensible Markup (XML) Language**

According to (Keio, 2013), Hyper Text Markup Language (HTML) is the language for describing the structure of Web pages. HTML gives authors the means to: publish

online documents with headings, text, tables, lists, photos, etc.; retrieve online information via hypertext links, at the click of a button; design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.; and include spread-sheets, video clips, sound clips, and other applications directly in their documents. While Extensible HTML (XHTML) is a variant of HTML that uses the syntax of XML, the Extensible Markup Language XHTML has all the same elements (for paragraphs, etc.) as the HTML variant, but the syntax is slightly different. Because XHTML is an XML application, one can use other XML tools with it (such as XSLT, a language for transforming XML content).

#### **1.1.17 Webkit**

According to official website for the WebKit Open Source Project that, WebKit is an open source web browser engine, or the name of the OS X system framework version of the engine that's used by Safari, Dashboard, Mail, and many other OS X applications. **WebKit** is a layout engine software component for rendering web pages in web browsers. It powers Apple's Safari web browser and was previously used in Google's Chrome web browser. It provides a set of classes to display web content in windows, and implements browser features such as following links that the user clicks on, managing a back-forward list, and managing a history of recently visited pages

From above discussion the study adapts jQuery framework due to its popularity, flexibility, and reliability to query web elements through the Document Object Model (DOM).

## **1.2 Statement of the Problem**

According to (Knight, 2009), it is asserted that web designers may not use fluid page designs for various reasons and further elaborated that, one of the reasons as being that: images, video, and other types of content with set widths, need to be set at multiple widths to accommodate different screen resolutions. The

available framework solutions involve heavy customization, which discourage web designers to use RWD techniques. This has resulted into low adaption of the concept by web designers.

Therefore, the study is being inspired to address this gap by presenting an alternative enhanced approach to RWD by enhancing the existing algorithm BlocksIt.js to an automated algorithm coined Liquidizer.js that will be responsive and accessed by a single line of code to improve the adaptation of the technique by web designers.

### **1.3 Justification**

There is urgency for an enhanced approach for RWD by developing a tool that will free web designers from coding many lines of code to accomplish a simple design style. In contrast, coding less gives them some space to nurture their talents and meet user's requirement specifications. This need is real to all stakeholders: for web designers; to reduce development time, for internet users; to satisfy their viewing experiences, and for academic scholars; to acquire new knowledge in the field of RWD.

#### **1.4.0 Objectives**

The study is guided by the following broad objective and specific objective as describe below.

##### **1.4.1 Broad objective**

The broad objective of this study is to conduct research that, implements an alternative enhance approach to RWD in Fluid Grid Concept. The broad objective is guided by the following specific objectives.

##### **1.4.2 Specific objective**

The study is directed and guided by the following specific objectives during the period of this study;

- i. To evaluate the existing state-of-art RWD technologies.

- ii. To design an algorithm Liquidizer.js that achieves an enhanced RWD.
- iii. To develop and implement the algorithm Liquidizer.js that achieves an enhanced RWD.
- iv. To conduct experiments for testing, validating, and evaluating the Liquidizer.js algorithm.
- v. To register the liquidizer.js algorithm in the existing jQuery library platforms.

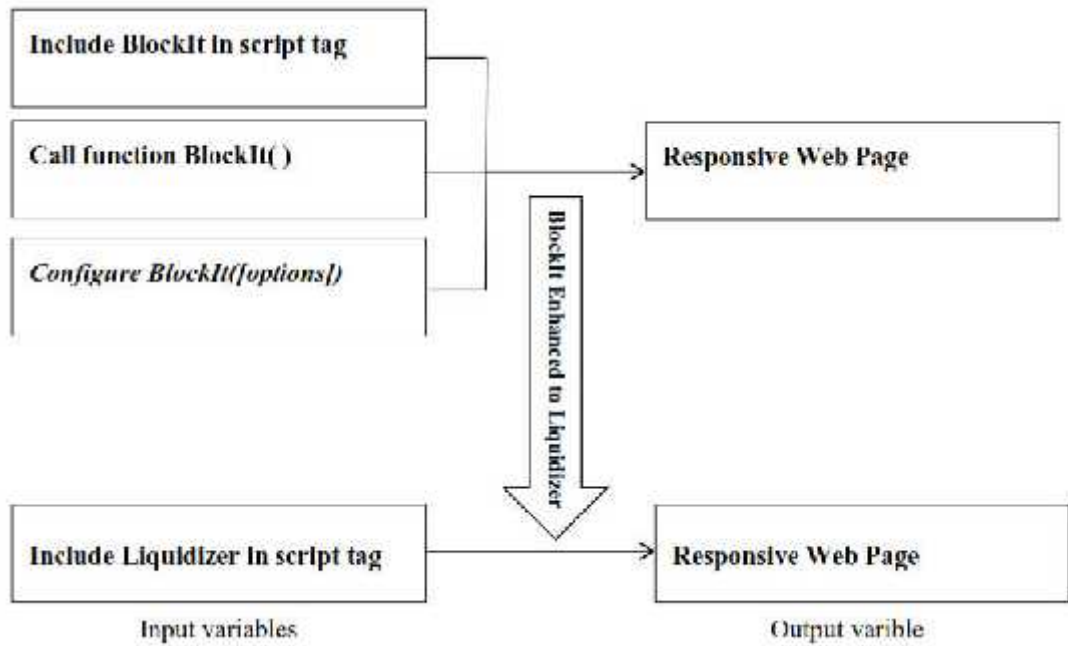
### **1.5 Research Questions**

The specific objectives instigate to answer and address the following research questions during the study;

- i. What are the existing technologies in RWD?
- ii. What are the methodologies for algorithm design?
- iii. What are the techniques for algorithm development?
- iv. What are the tools for algorithm testing, validation, and evaluation?
- v. What are the requirements for algorithm registration in jQuery library?

### **1.6 Conceptual Framework**

The conceptual framework initiates and is guided to answer the second research question; what are the various methodologies for algorithm design? The study presented a conceptual framework diagram of the algorithm BlockIt.js which is enhanced to Liquidizer.js algorithm in Figure 5 below.



**Figure 2:** Liquidizer.js Conceptual Framework

The study enhanced the RWD algorithm BlockIt.js by reducing the steps needed by the web designer in to one single step. The reduction of the steps usually induced the issues of quality of the algorithm. Therefore, study further enhanced RWD algorithm BlockIt to be more responsive. The enhanced algorithm was coined Liquizer.js by the study as shown in Figure 5.

## CHAPTER TWO

### 2.0 LITERATURE REVIEW

#### 2.1 Introduction

Extending beyond the boundaries of science, art, and culture, Responsive Web Design (RWD) provides new paradigms and techniques to develop one single website which looks different for different screen sizes from smart phones to main frame computers, so that it is usable on every devices. In the recent development, web standard designers are building websites which are getting closer and closer to the ideal of one web accessible to everyone and everywhere. According to (Solanki, 2012), RWD embraces four core concepts of; fluid grid, fluid image, media queries and responsive typography. Lots of ink has been spilled on literature reviews of RWD. However, in this literature survey, an effort has been made to show the chronological growth in RWD by presenting and in-depth review of fluid grid concept. The methodology adopted for the collection of literature review generally follows Creswell's recommendations for selecting material including articles published in respected journals (Creswell, 2009). The survey review reveals a shift away from traditional web design towards RWD. The study further, reveals a need for an alternative enhanced approach to RWD in fluid grid implementation, which has evolve to become an unavoidable good practice in web designing. Furthermore, after extensive exploration of the literature, the study recommends an outcome of classification of three meaningful categories of fluid grid concept solutions as; Frame-based Solution (FBS), Support-based Solution (SBS), and Algorithm-based Solutions (ABS) Approaches for Fluid Grid Concept implementation.

The study presents literature survey review in two sections, theoretical literature review and the related work published by other studies. The theoretical literature entails the state-of-art technologies of RWD, while the later entails the related work done by other studies.

It is suggested by (Creswell, 2009) that, the priority for the collection of literature review generally follows selecting material including articles published in respected journals. The priorities suggested and adopted by this study are reproduced in the Start with broad syntheses of literature

**Table 1:** Creswell’s Recommendations for Material, Adapted from (Creswell, 2009)

No	Method	Description
i.	Start with broad syntheses of literature.	Search overviews in encyclopedias, summaries of the literature in journal articles, etc.
ii.	Turn to respected journals	Search respected, national journals, especially ones that report research studies.
iii.	Turn to books.	Search books related to the topic.
iv.	Utilize conference papers.	Search for papers from major national conferences.
v.	Scan dissertation abstracts.	Quality varies, but may result in finding relevant dissertations on chosen subject.
vi.	Utilize the Web	This source is easy to access yet material must be rigorously screened for quality.

Therefore, this chapter of the study is triggered and guided to answer the first research question.

### **Research Question**

- i. What are the existing technologies in RWD?

## **2.2 Theoretical Literature Review**

In recent times, more and more people surf through the internet using mobile devices compared to desktop computers. As a result, mobile devices manufacturers and computer screen designers have been trying to provide users with quality web-browsing but they have not been able to adequately address users’ needs, which are exposed to traditional website layouts. Therefore, there is a need to switch to

Responsive Web design, which is capable of reshaping itself depending on various screen sizes and resolutions from largest screen sizes to smallest on mobile devices screen sizes. In the field of Web design and development, one can quickly get to the point of being unable to keep up with the endless new resolutions and devices. For many websites, creating a website version for each resolution and new device would be impossible, or at least impractical. Should one just suffer the consequences of losing visitors from one device, for the benefit of gaining visitors from another? Or is there another option?

It is asserted by (Marcotte, 2010) that, Responsive Web Design (RWD) stems from the notion of responsive architectural design, whereby a room or space automatically adjusts to the number and flow of people within it. Hence, transplanting this discipline onto Web design, one can have a similar yet whole new idea. Then, why should one create a custom Web design for each group of users? After all, architects don't design a building for each group size and type that passes through it. Like responsive architecture, Web design should automatically adjust. It shouldn't require countless custom-made solutions for each new category of users.

According to (Solanki, 2012), Responsive web design is the terminology given to the concept of designing and developing a website so that the layout changes depending on the device/viewport on which the website is being viewed. By device, this could be a mobile phone, tablet, laptop, desktop computer, or even a smart TV.

According to (Harb et al., 2011), Responsive Web design is an approach that suggests that, design and development should respond to the user's behavior and environment based on screen size, platform and orientation. The practice consists of a mix of flexible grids and layouts, flexible images and an intelligent use of CSS media queries. As the user switches from their laptop to iPad, the website should automatically switch to accommodate for resolution, image size and scripting abilities. In other words, the website should have the technology to



automatically respond to the user's preferences. This would eliminate the need for a different design and development phase for each new gadget on the market.

(Knight, 2011), states 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.

### **2.2.1 Technologies of RWD**

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. The survey is motivated to present the RWD technologies of Fluid Grid Concept, Fluid Images Concept, Media Query Techniques, Responsive Typography Technology, and RESS (Responsive Web Design + Server Side Components) Technology.

### **2.2.1.1 Fluid Grid Concept**

According to (Nimesh, 2012), a fluid is a substance that continually deforms (flows) under an applied shear stress. In adaptive grids, one defines pixel-based dimensions. Hence one will have to adjust the widths and heights manually in certain device viewports. Since fluid grids flow naturally within the dimensions of its parent container, limited adjustments will be needed for various screen sizes and devices. Mobile devices are getting smaller in size and people prefer using them in their personal work. On the other hand, desktop monitors are getting wider with higher resolutions. So, one cannot plan for smaller devices in responsive design. The advantage of fluid grid is that you can adjust the max-width and it will still work on larger screens due to the percentage based calculations.

It is elaborated by (Staff, 2013) that, grids provide structure to website, while relative units can provide fluidity to that structure. Fluid grids are defined using a maximum width for the design; whereas the grid contained therein, is defined using relative widths and/or heights, instead of pixels. This allows the widths and heights to adjust accordingly in relation to the parent container. In other words, as the size of the screen's website is being viewed, gets smaller or larger, the screen will adjust accordingly.

(Hurb et al., 2011b) asserts that, the fluid grid concept calls for page element sizing to be in relative units like percentages, rather than absolute units like pixels or points. So, the main idea of flexible grids is to create a layout where all elements are based on the calculated percentage width and so all elements in the layout are resizable in relation to one another.

### **2.2.2.2 Fluid Images Concept**

It is argue by (Boudreaux, 2012b) that, it has been a rule of thumb, one to set images to a specific height and width in pixels, but the downside, setting absolute image sizes is not flexible. Therefore, fixed image dimensions become difficult to view in

smaller screen sizes and resolutions. The fixed images may be perfect for desktop or laptop displays, but fail when rendered on smaller devices.

It is explained by (Graeve, 2012) that, the flexible images are sized, in relative units, so as to prevent them from displaying outside their containing element. It is a concept that allows developers to adapt images or other media to load differently depending on the device, either by scaling or by using the CSS overflow property.

It is demonstrated by (Storey, 2013) that, fluid images can be achieved by sizing images in relative units, rather than absolute pixel dimensions. The most common relative solution is to set the max-width of the image at 100%. While images with this CSS will display at their native dimension so long as there is enough room in the HTML container to do so; as the browser window narrows, the images will scale to fit.

### **2.2.2.3 Media Queries Technique**

It is elaborated by (Boudreaux, 2012a) that, media queries go beyond the conventional media types that have been used since CSS 2.1, which allowed one's websites to obtain some degree of media device responsiveness; such as `media="screen"` or `media="print"`. The W3C answer to improving device-specific website response is the media queries specification. Media queries open up the spectrum of possibilities with the "media" attribute that controls how the styles get applied. Media queries takes the previous scheme to the next level by allowing users to target styles based on a number of device properties, including screen width, orientation, resolution, and others. Media queries allow users to target not only certain devices and classes of devices, but it allows users to actually inspect the physical characteristics of the device.

According to (Hurb, 2011) explain 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 urged by (Frost, 2013) 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.

#### **2.2.2.4 Responsive Typography Technology**

It is urged by (Soueidan, 2013) 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.

It is asserted by (Smarty, 2013) 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.

It is suggested by (Pamental, 2014), the four core Responsive Web Type are: Performance; load what one needs, when one needs it, progression; ensure that all devices get a good design and enhance the experience for devices/browsers that can handle it, proportion; one scale doesn't fit, and polish, great design is details, and type is no different. He explains responsive typography is the notion that type must move and change and adapt just as the rest of the designs do, as users shift from device to desktop and beyond. It has a bigger impact on readability than any other aspect of design.

#### **2.2.2.5 RESS Technology**

It is urged by (Wroblewski, 2011), 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.

According to (Olsen, 2013), 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.

It is explained by (Wroblewski, 2012) 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.

The study is motivated to present an alternative approach to RWD by developing an automated conversion algorithm coined Liquidizer from Fixed Grid websites to Fluid Grid. Therefore, the Fluid Grid Concept is presented in-depth in the next sub-section.

### **2.2.2 Types of Web Layouts Approaches**

According to (Tranfici, 2013), classifies the web layouts as fixed-width layouts, liquid layouts, and elastic layouts.

#### **2.2.2.1 Fixed-Width Layouts**

It is described by (Tranfici, 2013) that, in fixed-width layouts, the width of the site is bound to a certain number of pixels. Generally, the measure chosen is 960 pixels. This is because with the passing of time, developers have found 960 pixels to be the best size for grid layouts, because the number is easily divisible by 3, 4, 5, 6, 8, 10, 12, and 15. However, the fixed-width layouts have some disadvantages. The designers who want to create a fixed-width website have to keep in mind that every aspect of their work has to be usable and clearly visible to a large number of screens, browsers, and devices. The wide variety of devices on the market at this time, as well as the consequently great variability of screen sizes makes creating one-size-fits-all content quite a challenging task, and arguably a challenge that outweighs the precision and control of fixed-width design.

### **2.2.2.2 Liquid Layouts**

(Tranfici, 2013) explains that, the first basic difference between the fixed-width type of layout and liquid layouts is the measurements of their size. The fixed-width layouts are measured in pixels, but liquid or fluid layouts, dimensions are defined in percentages, and as expected, this affords greater malleability and fluidity. In other words, by setting a percentage, one won't have to think about device size or screen width, and consequently, one can find a reasonable solution for each case because the design's size will adapt to the size of the device used. Liquid layouts are closely linked to media queries and special styles for optimization. Percentage-based widths alone will likely not be enough to accommodate one's design for a large variety of display sizes.

### **2.2.2.3 Elastic Layouts**

It is elaborated by (Tranfici, 2013) that, elastic layouts are somewhat similar to liquid layouts. The main difference is once again the unit of measurement for size. The size indicator for elastic layouts is neither in pixels nor percentages; it's measured in *ems*. An *em* is the equivalent of the size (in pixels) defined in the *font-size* CSS rule. For example, if one styles text with a *font-size* of 20 pixels, 1 *em* would be equal to 20 pixels, 2 *ems* would correspond to 40, and so on. This type of layout gives the developer strong typographic control. Since the vast majority of layouts are predominantly populated with text, the precision of type treatments makes elastic layouts a strong contender for many projects. However, even with this type of solution, there is a risk of an unpleasant and unaesthetic horizontal scroll bar in some rare cases. The study is motivated to explore the fluid grid concept.

### **2.2.2.4 Fluid Grid Concept**

It is explained by (Allsopp, 2000) that, 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. Therefore, to understand Fluid Grid Concept, then an in-depth critic of Fixed Grid layout and Fluid Grid layout are discuss.

It is asserted by (Tranfici, 2013) 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. The term “grid” is used rather freely and doesn’t imply a requirement to implement any of the available grid frameworks. What it means here is using CSS for positioning and for laying out margins and spacing, and for implementing various web layout types in a new way. Layouts and text sizes are typically expressed in pixels. But a pixel can be one dot on one device and eight dots on another. So how do one approach responsive web design if everything is pixel-based? The answer is: to stop using pixel-based layouts and start using percentages or the em for sizing.

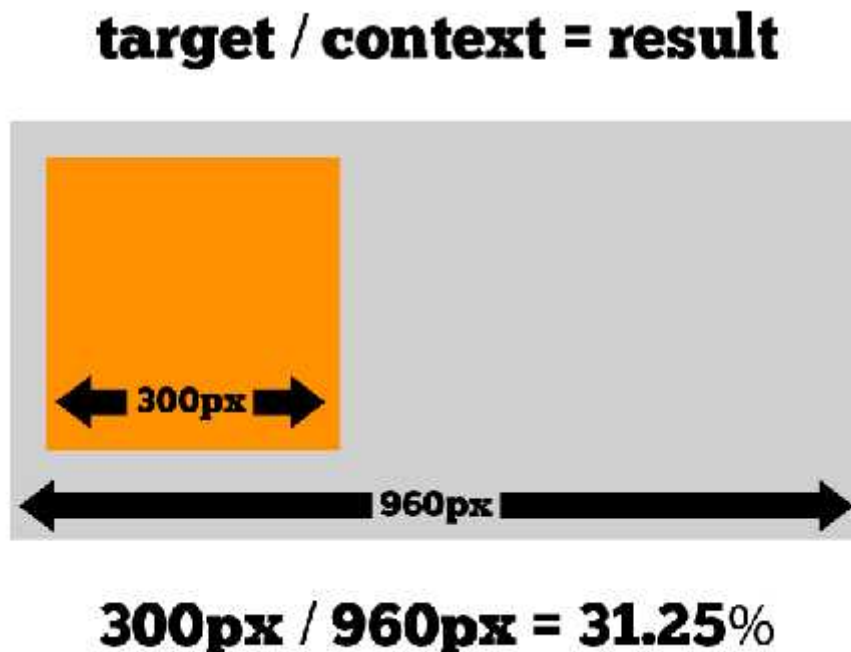
By basing text sizes, widths and margins on percentages or on the em, a unit of measurement based on a font’s point size, one can turn a fixed size into a relative size. This means one will need to do a little math to achieve a flexible grid and text size system.

It is asserted by (Marcotte, 2010), in Dan Cederholm’s book, title “Handcrafted CSS”, in a chapter covering fluid grids where, he provided a simple and consistent formula for converting fixed width pixels into proportional percentages, the following formula is applied.

$$\text{Target} \div \text{Context} = \text{Result}$$

It is demonstrated by (Pettit, 2012) that, in order to calculate the proportions for each page element using the formula  $\text{Target} \div \text{Context} = \text{Result}$ , one must divide the target element by its context. Currently, the best way to do this is to first create a high fidelity mockup in a pixel based imaged editor, like Photoshop. With one’s high fidelity mockup in hand, one can measure a page element and divide it by the full width of the page. For example, if one’s layout is a typical size like 960 pixels across, then this would be one’s “container” value. Then, let’s say that one’s target element is some arbitrary value, like 300 pixels wide. If one multiples the result by

100, one get the percentage value of 31.25% which one can apply to the target element. The Figure 2 below summarizes the calculations outcome.



**Figure 3:** Proportion of a 300px element, Adapted from (Pettit, 2012)

### 2.3.0 Related work

After extensive exploration of the related work, the study categorizes the existing solution for conversion from Fixed Grid Layout to Fluid Grid Layout in to three categories; Frame-based solutions, support-based solutions, and Algorithm-based solutions approaches.

#### 2.3.1 Frame-based solution

The frame-based solution (FBS); it can be defined as is framework approach that uses a predefined layout which needs installation, training, and heavy customization prior to its use.

According to (Johal, 2012), there are several solutions developed using this technique. Two of the most popular responsive frameworks are Twitter



Bootstrap and Foundation. Twitter has open-sourced Bootstrap, a framework they use on Twitter. While the latter, has a similar responsive framework created by ZURB. Both frameworks are fully responsive, allowing developers to use well-documented and tested components that will work on a large variety of screens and devices. Creating a website to have a responsive, fluid layout can be a lot of work. There are a few responsive CSS frameworks that do a lot of the heavy lifting. CSS frameworks are libraries that package CSS and JavaScript components for commonly used UI components and interactions such as grids, buttons, or carousels.

For example, if one is converting an existing website that has a design for horizontal navigation. Both Bootstrap and Foundation provide this component, however the default styles for the Foundation one is simple and plain, whereas the Bootstrap version is more polished. Foundation provides more control while Bootstrap does more work.

### **2.3.2 Support-based solution**

The support-based solution (SBS); it can be defined as an approach that uses standalone tools, which help a web designer to make some calculations to achieve responsive web design. The tools uses mathematical process associated with converting fixed-width design work to a fluid layout is converting absolute units of measurements (i.e. px and pt) into relative units of measurement such as ems and percent (%) for typography, spacing, container widths, etc. There are several tools available to date. The most commonly used is PXtoEM at <http://pxtoem.com/>

It is demonstrated by (Cray, 2012) that, PXtoEm is a tool that provides web designers with a simple conversion environment that help one with the entire math. The site also allows web designers the ability to quickly and easily change the base font size of their layout to something that leads to more manageable math. The Figure 3 below demonstrates how the PXtoEM interface works.

Pixels	EMs	Percent	Points	Pixels	EMs	Percent	Points
5px	0.3125em	19.38%	5pt	5px	0.3125em	19.38%	5pt
7px	0.4375em	27.34%	7pt	7px	0.4375em	27.34%	7pt
9px	0.5625em	35.30%	9pt	9px	0.5625em	35.30%	9pt
11px	0.6875em	43.26%	11pt	11px	0.6875em	43.26%	11pt
13px	0.8125em	51.22%	13pt	13px	0.8125em	51.22%	13pt
15px	0.9375em	59.18%	15pt	15px	0.9375em	59.18%	15pt
16px	1.0em	62.50%	16pt	16px	1.0em	62.50%	16pt
17px	1.0625em	65.94%	17pt	17px	1.0625em	65.94%	17pt
18px	1.125em	69.38%	18pt	18px	1.125em	69.38%	18pt
19px	1.1875em	72.81%	19pt	19px	1.1875em	72.81%	19pt
20px	1.25em	76.25%	20pt	20px	1.25em	76.25%	20pt
21px	1.3125em	79.69%	21pt	21px	1.3125em	79.69%	21pt
22px	1.375em	83.13%	22pt	22px	1.375em	83.13%	22pt
23px	1.4375em	86.56%	23pt	23px	1.4375em	86.56%	23pt
24px	1.5em	90.00%	24pt	24px	1.5em	90.00%	24pt

1. Enter a base pixel size

16 px

2. Convert

PX to EM      EM to PX

24 px      1.5em

3. Result

1.500em

**Figure 4: PX to EM Conversion made simple** Adapted from (Cray, 2012b)

One can simply use the formulas to calculate the desired conversion; example 16px is used as the body text size in all conversions because that is the browser default. One will change 16px to one's base text size. The following examples demonstrate how different conversions are calculated

**PX to EM** - Formula: size in pixels / parent size in pixels. Example: 12px / 16px = .75em

**PX to %** - Formula: size in pixels / parent size in pixels \* 100. Example: 12px / 16px \* 100 = 75%

**PX to PT** - Formula: size in pixels \* (points per inch / pixels per inch). Example: 16px \* (72pt / 96px) = 12pt

**EM to PX** - Formula: size in EMs \* parent size in pixels. Example: .75em \* 16px = 12px

**EM to %** - Formula: size in EMs \* 100. Example: .75em \* 100 = 75% (Cray, 2012)

### 2.3.3 Algorithm-based solution

The algorithm-based solution (ABS); it can be defined as an approach of developing algorithms which will automatically convert fixed grid layout to fluid grid layout. There exist several algorithms to implement fluid grid concept. The most commonly used is BlockIt.js.

According to (Kenny, 2012), BlockIt is a jQuery plugin for creating dynamic grid layout. It is used to convert HTML elements into '*blocks*' and position them in well-arranged grid layout. It allows joining of two or more blocks into a big block element. It is licensed under the GNU General Public License. The plugin can be access through the following function interface for customization and creation of dynamic grid layout shown in Equation 1 below.

```
$(document).ready(function() {
    $('#container').BlocksIt({
        numOfCol: 4,
        offsetX: 8,
        offsetY: 8,
        blockElement: '.block'
    });
});
```

**Equation 1:** BlockIt.js Interface Adapted from (Kenny, 2012)

**Where:**

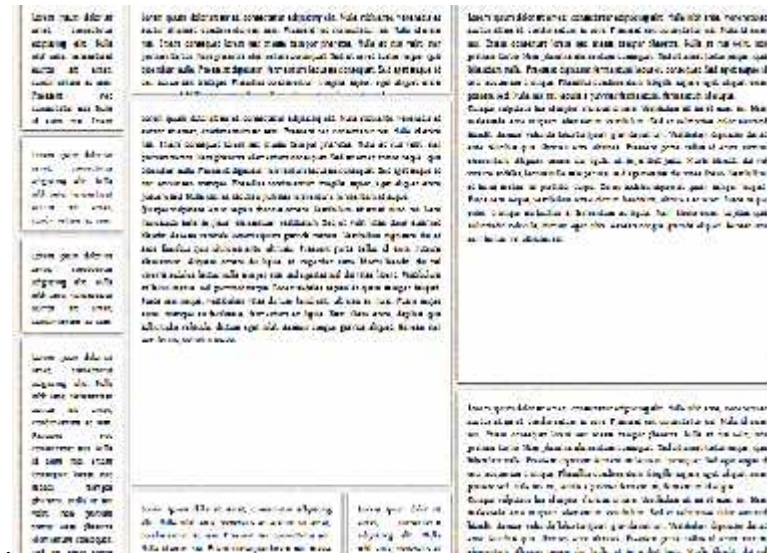
**numOfCol:** *Type: Int ( Default: 5 )* // The number of columns to be created.

**offsetX:** *Type: Int ( Default: 5 )* // Margin left and right for each block.

**offsetY:** *Type: Int ( Default: 5 )* //Margin top and bottom for each block.

**blockElement:** *Type: String ( Default: div )* //Targeted child element, which will converted into blocks.The Figure 4 below demonstrates how BlockIt.js can randomly

generate dynamic blocks. The source code algorithm for BlockIt.js is shown in Appendices A of the study.



**Figure 5:** Dynamic grid layout blocks. Adapted from (Kenny, 2012)

It seems clear from the above discussion that, ABS approach have address the implementation gap by developing an algorithm that generate dynamic grid layout but only after receiving an input through its interface. However, web designers have to spend more man-hours using the interface for data input hence prolonging development time. Therefore the study proposes an alternative enhanced approach to RWD by developing an algorithm Liquidizer.js. The Liquidizer.js is simply called by a single line of code between a script tags to perform the same task.

## 2.4 Usability Models

It is stated by (Abran, Khelifi, Suryan, & Seffah, 2003) that, within the last couple of years, a major shift has occurred in the development, design and deployment of software applications. With the considerable growth of distributed applications, it is to be noted that, especially for the Internet, the developers, technical staff and training instructors no longer have direct access to the end-user of their software systems. Software usability is no longer a luxury, but rather a basic determinant of productivity and of the acceptance of software applications.

It is argued by (Bevan, 1995) that, the term “usability” refers to a set of multiple concepts, such as execution time, performance, user satisfaction and ease of learning (“learnability”) taken together. But usability has not been defined homogeneously, either by the researchers or by the standardization bodies. According to (ISO/IEC 9126-1, 2000) defines usability as “The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions.” While, suggested by (ISO 9241-11, 1998)to be defined as, “The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.” It also asserted by (IEEE Std. 610.12, 1990)to be defined as “The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component.”

It is reported by (Bass, Bonnie, & Kates, 2001), that usability attributes characterizes the ease to instantiate or execute a software system. This explicitly does not include usability in terms of difficulty to maintain or reuse parts of the software system which are covered by other characteristics. The usability sub characteristics also include features like ease of operation of the user interface and its relevance, ability to train the users and ease with which they are able to learn the system (learn ability), ease of operating system that is operability of software system developed by system engineers.

The diversity of viewpoints and their related usability requirements have led to different perspectives on usability in the various ISO models that have been developed over years by distinct groups of usability experts. It is asserted by (Xenos, 2001) that, regardless of the way one views software quality, usability plays an important role in all quality perspectives: in recognizing the quality of a software product (transcendental view), in determining if it is fit for purpose (user view), in designing from early stages how the user will use it (manufacturing view), in determining its quality characteristics based among others on the usability factor

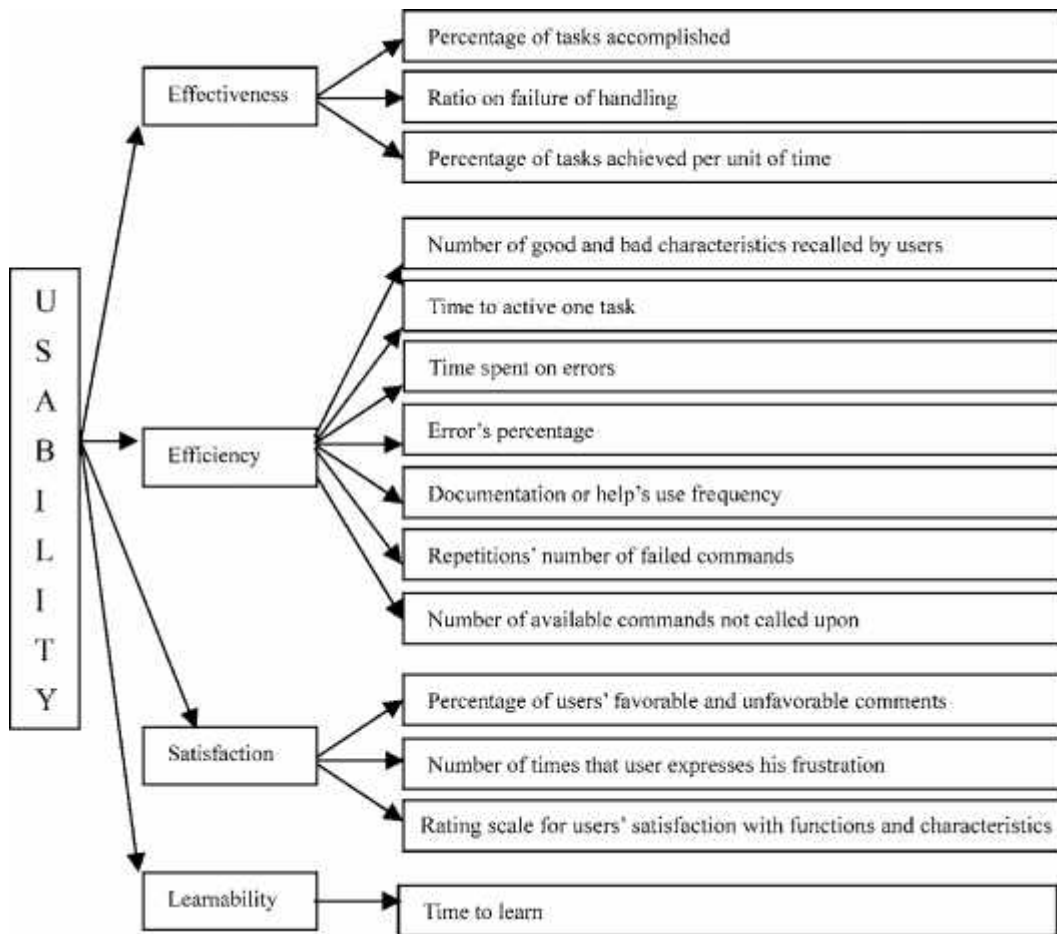
(product view), and in affecting the value that the user will be willing to pay for a usable or less usable product (value-based view). This study is guided to adapt the user view of the web designer.

#### **2.4.1 Usability Model Types**

This study presents usability perspective in most frequently used models of software quality characteristics, namely ISO 9241 Model, Factors Criteria Metrics (FCM) Model, and ISO 9126 Model. Furthermore, it discusses their limitations and strengths, in order to decide the best model to be adapted by the study

##### **2.4.1.1 Usability in ISO 9241**

According to (ISO 9241-11, 1998) defines usability, as useable software that allows the user to execute his task effectively, efficiently, and satisfactory in the specified context of use. According to the standard, measurement of system usability consists of three usability attributes; Effectiveness: How well do the users achieve their goals using the system, Efficiency: What resources are consumed in order to achieve their goals, and Satisfaction: How do the users feel about their use of the system? The standard presents usability guidelines and is used for evaluating usability according to the context of use of the software. Some researchers have proposed their own usability model, through additional definitions or attributes of the concept, often including the learnability characteristic for usability. The Figure 11 below presents the usability model of (Nielsen, 1994).

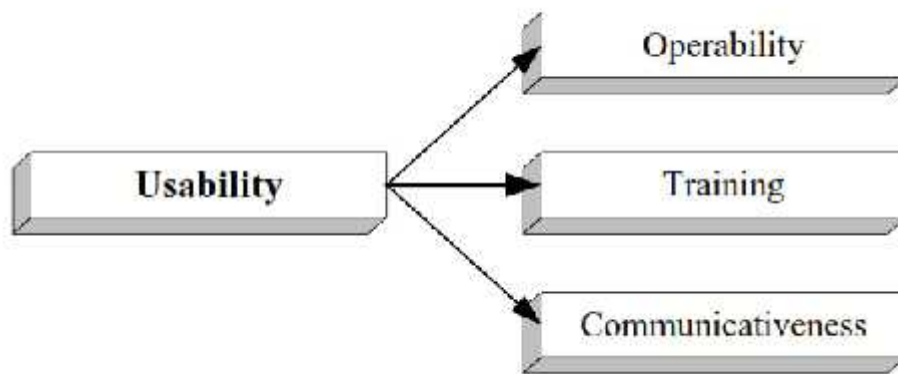


**Figure 6:** Usability Model, Source: (Nielsen, 1994)

#### 2.4.1.2 Usability in Factors Criteria Metrics (FCM) Model

Usability is an important factor of software quality. Usability has always been present, even in the very first models of software quality. According to (Xenos, 2001), the basic idea in all software quality models is to define software quality, which is a term that is too abstract to be studied directly, by dividing it into attributes (usually called quality factors). One of these factors is usability, a concept that is also abstract and therefore is usually divided into more specific attributes (usually called characteristics). In some models, these characteristics are further divided into sub-characteristics and so on, always with the same concept in mind: to divide an abstract term into better defined terms (terms that can be measured objectively or with a reduced level of subjectivity).

(McCall, 1977), reported the three criteria into which usability is divided according to the Factors Criteria Metrics (FCM) model are: Operability, Training and Communicativeness. These criteria are associated only with the usability factor. Operability is associated to the user's effort for operation and operation control (for example mouse support, macro-commands, etc.). Training is associated to the effort required to teach the use of software to the user, while communicativeness is associated to how well the software communicates to the user the purpose for which it has been developed and the method to use it. Based on the FCM measurement method, each criterion could be associated with metrics in the form of questions allowing subjective "yes" or "no" answers. The Figure 12 below presents the FCM Model



**Figure 7:** Usability FCM Model, Source: (McCall, 1977)

#### **2.4.1.3 Usability in ISO 9126 Model**

Based on FCM model, international efforts lead to the development of the international standard *ISO 9126* for software quality. *ISO 9126* comprises of a basic set of 6 independent quality characteristics: Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability. Unlike FCM, *ISO 9126* is completely hierarchical and relates each sub-characteristic to only one of the basic characteristics.



According to (ISO/IEC 9126-1, 2000) , usability can be divided into understandability (which is the user’s effort for recognizing the underlying concept of the software), learnability (which is the user’s effort for learning how to use the software) and operability (which is the user’s effort for operation and operation control like mouse support, macro-commands, etc.). ISO 9126 is currently used by many software developers to define quality goals and usability is always a prime concern in software quality. Table 4 summarizes the characteristics and sub-characteristics of ISO 9126.

**Table 2:**Source: ISO 9126 Characteristic and sub-characteristics (Abran et al., 2003)

Characteristic	Sub-characteristic	Explanation
Usability	Understandability	Does the user comprehend how to use the system easily?
	Learnability	Can the user learn to use the system easily?
	Operability	Can the user use the system without much effort?
	Attractiveness	Does the interface look good?

After an extensive analysis of the three models of FCM Model, ISO 9241 Model, and ISO 9126 Model, it is clear from the above discussion that, FCM Model and ISO 9241 are enhanced models to form the ISO 9126 Model. Therefore, the study adapts the ISO 9126 model by using the quality characteristics of usability and sub-characteristics of understandability, learnability, operability and attractiveness to evaluate Liquidizer.js framework to be measured by web designers. Based on the FCM measurement method, each sub-characteristic is associated with metrics in the form of questions allowing the web designers a “yes” or “no” answers. The Appendix D presents the sample questionnaire to be used by the respondents. While Appendix E presents an introductory letters to respondents.

## 2.5 Recommendation and Conclusion

It is a natural principle that, living and non-living things change with time. But the internet is one of the fastest changing things in the world. As a result of which RWD is dynamic. The pervasiveness of mobile devices today means audiences want to

consume content on whichever browser or device they prefer. As a result, web designers and publishers have rallied behind the idea of responsive design to reach the widest possible audience and promise the reader a great experience no matter what their screen size.

The review reveals a shift away from traditional web design towards RWD by the web designers, which has evolved to become an unavoidable good practice in web designing. Moreover, the review recommends a classification of three meaningful categories of fluid grid concept's solutions as: Frame-based Solution (FBS), Support-based Solution (SBS), and Algorithm-based Solutions (ABS). As a result of which, the study is guided to adopt an Algorithm-based Solution (ABS) approach for fluid grid implementation.

## **CHAPTER THREE**

### **3.0 RESEARCH METHODOLOGY**

#### **3.1 Introduction**

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.

This chapter introduces the research design and its rationale by reviewing various methodologies approaches and programming languages. It justifies the choices for the techniques and methods by addressing their potential limitations and domain applications. As a result of which, this chapter of the study is triggered and guided to answer the second research question.



### **3.1.1 Research Question**

- ii. What are the methodologies for algorithm design?

### **3.2 Research Design**

It is described by (Labaree, 2014) that, research design refers to the overall strategy that one chooses to integrate the different components of the study in a coherent and logical way, thereby, ensuring one will effectively address the research problem; it constitutes the blueprint for the collection, measurement, and analysis of data. Research design is the overall plan for connecting the conceptual research problems to the pertinent (and achievable) empirical research. In other words, the research design articulates what data is required, what methods are going to be used to collect and analyze this data, and how all of this is going to answer ones research question.

#### **3.2.1 Experimental Research Design**

According to .(Carroll, 2014) experimental research design is a type of research design most appropriate in controlled settings such as laboratories. The research design assumes random assignment of subjects and random assignment to groups. It attempts to explore cause and affect relationships where causes can be manipulated to produce different kinds of effects. Because of the requirement of random assignment, this design can be difficult to execute in the real world (non-laboratory) setting.

(Labaree, 2014) describes experimental research design as, a blueprint of the procedure that enables the researcher to maintain control over all factors that may affect the result of an experiment. In doing this, the researcher attempts to determine or predict what may occur. Experimental Research is often used where there is time priority in a causal relationship (cause precedes effect), there is consistency in a causal relationship (a cause will always lead to the same effect), and the magnitude of the correlation is great.

Hence, the study implements the Experimental Research Design by exploring the effect of applying algorithm Liquidizer.js to fixed layout page and non-application of the algorithm Liquidizer.js to a controlled fixed layout page. A group of thirty (30) web designers are used to test the implementation of the algorithm Liquidizer.js functionality, using testing tools.

### 3.3.1 Accessing BlocksIt.js

1. First, include jQuery and .BlocksIt.js script files inside <head> tag as shown below.

```
<script type="text/javascript"src="jquery.min.js">
</script><script type="text/javascript"src="blocksIt.js"></script>
```

2. Next, call the .BlocksIt() function on jQuery object using the following interface code to access the jQuery library

```
$(document).ready(function() {
$('#objectID').BlocksIt();
});
```

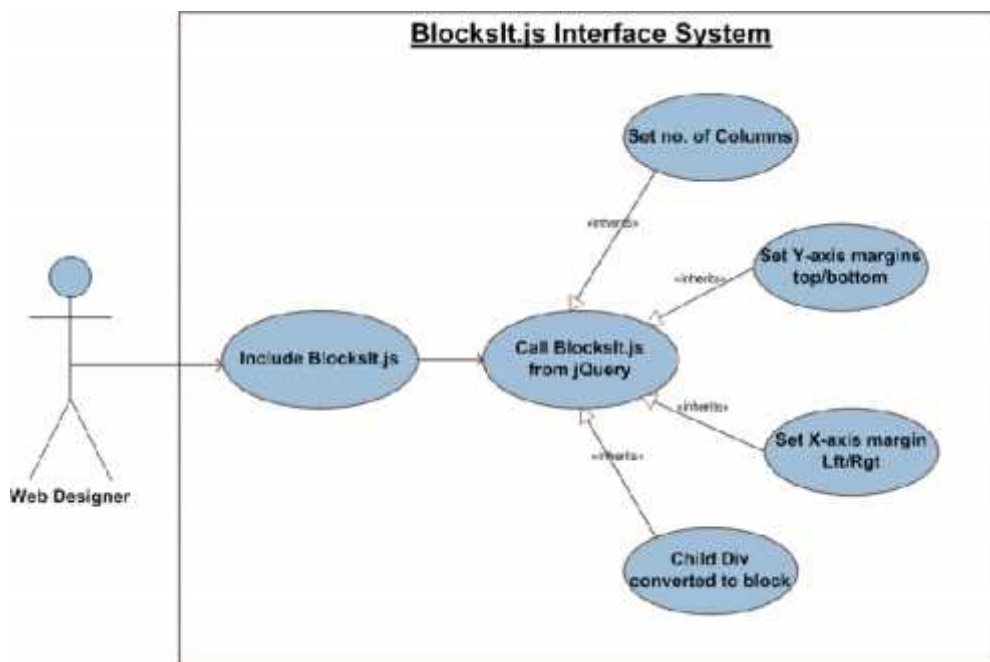
3. Then, the following option are available for .BlocksIt( [Options] ), where the table 2 presents an array to configure blocks for default values.

**Table 3:** BlockIt Default Options Adapted from (Kenny, 2012)

Name	Type	Default	Description
numOfCol	Int	5	The number of columns to be created.
Offset	Int	5	Margin left and right for each block.

Name	Type	Default	Description
Offset	Int	5	Margin top and bottom for each block.
blockElement	String	"div"	Targeted child element, which will converted into blocks.

The use case diagram in Figure 6 below elaborates the BlockIt.js interface system, where an actor, web designers invokes use-case, include BlockIt.js by including it in the script tag which calls the script from the jQuery library. The web designer accesses four options to create a framework in order to work on. This procedure discourages most of the web designers to use the framework as one must be conversant with jQuery language. The blocksIt.js source code is shown in Appendix A - 1: blocksIt.js algorithm



**Figure 8:** Use-Case Diagram Interface for BlockIt.js

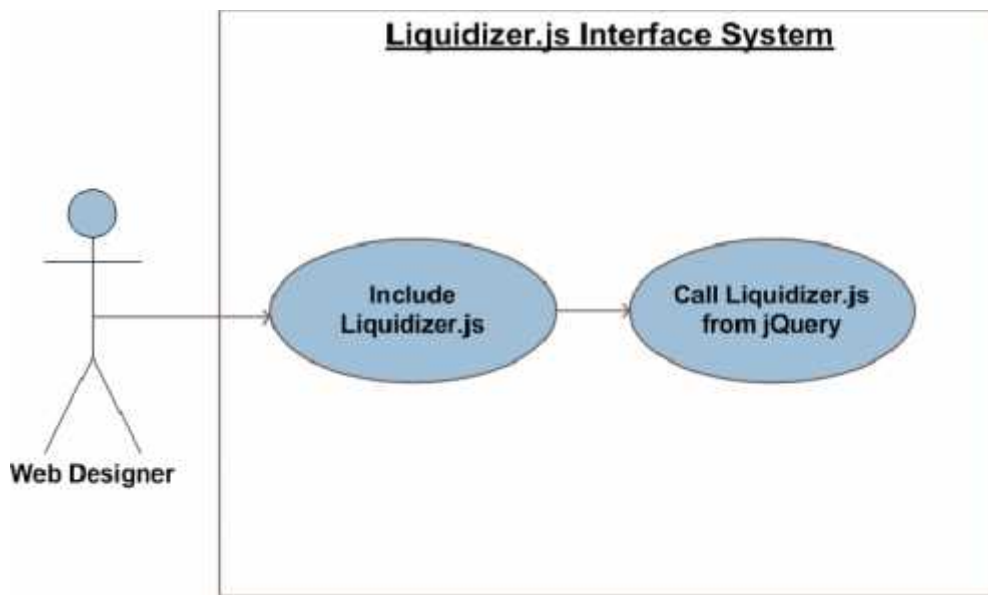
### 3.3.2 Accessing Liquidizer.js

Just, include jQuery and .Liquidizer.js script files inside <head> tag as shown below.

```
<script type="text/javascript"src="jquery.min.js"></script>
```

```
<script type="text/javascript"src="Liquidizer.js"></script>
```

The use-case diagram Figure 7 below elaborates the Liquidizer.js Interface System where an actor, web designers just invokes use-case, include Liquidizer.js by including it in the script tag which calls the script from the jQuery library. The web designer need not know jQuery language to use this script.



**Figure 9:** Use-Case Diagram Interface for Liquidizer.js

### 3.4.0 Design Rationale

(Burge & Brown, 2002) states that the design process is the set of steps, or activities, that take place in achieving the design goals, or objectives. Models of the design process are used in order to either describe the activities of the design process or prescribe how the designing should be done. Many decisions need to be made while designing. A process model can assist in guiding what decisions should be made when, and if, the model describes the design of a specific artifact, can even provide the knowledge to be used to make the decisions.



It is described by (Lee, 1997) that, design rationales include not only the reasons behind a design decision but also the justification for it, the other alternatives considered, the tradeoffs evaluated, and the argumentation that led to the decision.

### **3.4.1 Algorithm Development Methodologies**

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. Developing a correct algorithm can be a significant intellectual challenge – by contrast, coding it should be straightforward (although coding it well may not be!). This section triggers and is guided to answered the second research question of the study.

#### **Research Question**

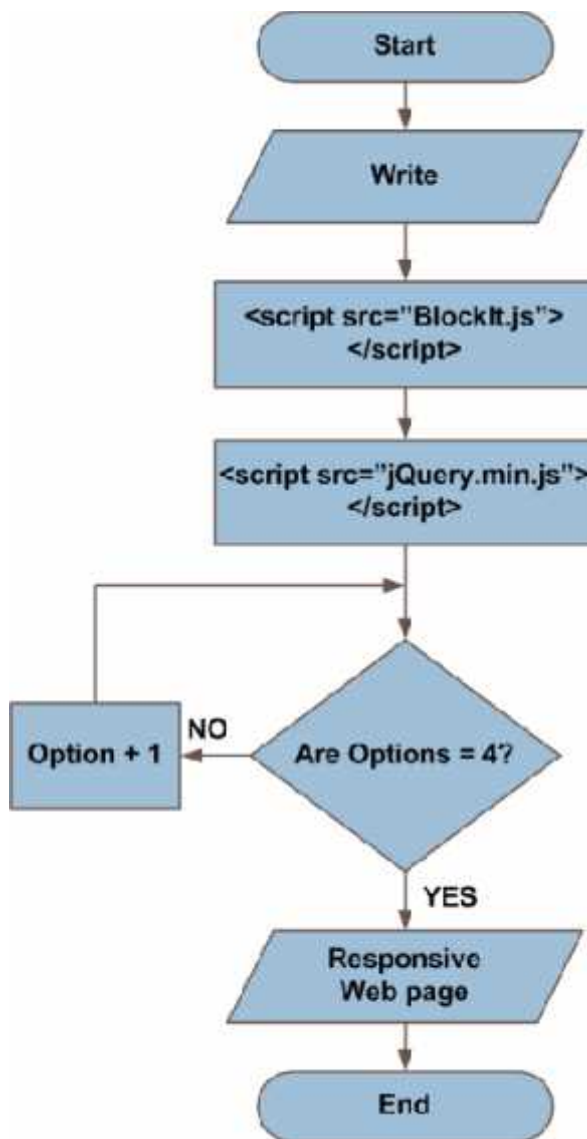
- ii. What are the methodologies for algorithm design?

Therefore, the most widely used notations for developing algorithms are flowcharts and pseudo-code. These are independent of the programming language to be used to implement the algorithm.

### **3.4.2 Flowcharts**

A flowchart is a diagram containing lines representing all the possible paths through the program. Flowcharts are used in designing and documenting complex processes or programs. Like other types of diagrams, they help visualize what is going on and thereby help the designers to understand a process, and perhaps also find flaws, bottlenecks, and other less-obvious features within it.

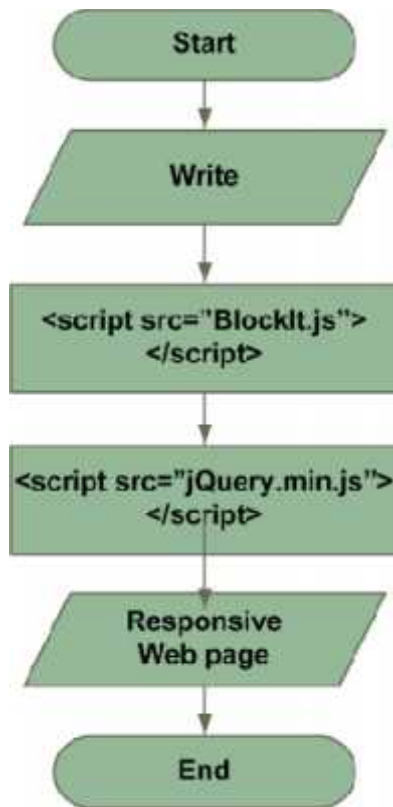
There are many different types of flowcharts, and each type has its own repertoire of boxes and notational conventions. The two most common types of boxes in a flowchart are; a processing step, usually called *activity*, and denoted as a rectangular box, and a decision, usually denoted as a diamond (“Flowchart,” 2014). The Figure 8 below demonstrates the processes a web designer undergoes to achieve a responsive web page using a BlockIt.js framework.



**Figure 10:** Processing Responsive Web using BlockIt.js Framework

This makes flowcharts as a better way of communicating the logic of a system to all concerned. With the help of flowchart, a problem can be analyzed in a more effective way. Furthermore, program flowcharts serve as a good program documentation, which is needed for various purposes. Hence, the flowcharts act as a guide or blueprint during the systems analysis and program development phase. Finally, the maintenance of operating program and debugging process becomes easy with the help of flowchart. This is demonstrated by the Figure 9 below of the Liquidizer.js

Framework simplicity to achieve responsive web page as opposed to BlockIt.js Framework.



**Figure 11:** Processing Responsive Web using Liquidizer.js Framework

However, flowcharts have their share of limitations. Sometimes, the program logic is quite complicated and in those cases, flowcharts become complex and clumsy. Moreover, if alterations and modifications are required the flowcharts may require re-drawing completely hence wasting valuable man-hours. Furthermore, as the flowchart symbols cannot be typed, reproduction of flowcharts becomes a big problem. Finally, the essentials of what are done can easily be lost in the technical details of how they are done.(Bhatnagar, 2009)

It seems clearly from the above discussion that even though, flowcharts are better way of communicating the logic of a system to all concerned but have limitations where, if alteration and modification are required, then one needs to re-draw. This limitation is a concern to this study hence making it to be an unsuitable choice for the study.

### 3.4.3 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 (Eck, 2011a).

#### **3.4.4 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 (Eck, 2011b)

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 2 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 2:**Pseudo-code version of Liquidizer Algorithm

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. Furthermore, the program is an implementation of a programming language.

### 3.4.5 Programming Language

There exists several web programming languages for algorithm implementation, as mention above in chapter one. The most commonly use language is jQuery. 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.

It is suggested by (Alsup, 2007), 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. Therefore, the above section was instigated and guided to answered research question four of the study.

### **3.5 Algorithm Development**

The study implemented the proposed design to code and builds the algorithm Liquidizer.js. The following is part of the enhanced code used for the development of the algorithm

```
(function ($) {
```

```
    $.fn.liquidizer = function(options)
```

```
    {
```

```
        // defining the defaults values
```

```
        var defaults =
```

```
        {
```

```
            color: "#000000",
```

```
            backgroundColor: "gold",
```

```
            fontFamily: "sans-serif",
```

```
            fontStyle: "none",
```

```
            fontWeight: "none"
```

```
        };
```

```
        // $.extend() - Merge the contents of two or more objects together into  
the first object
```

```

var settings = $.extend(
    {},
    defaults,
    options);

// making the plug-in chain-able
return this.css(
    {
        color: settings.color,
        backgroundColor: settings.backgroundColor,
        fontFamily: settings.fontFamily,
        fontStyle: settings.fontStyle,
        fontWeight: settings.fontWeight
    }
);
};
}(jQuery));

```

While the full integrated code is shown in Appendix A - 2: liquidizer.js algorithm.

### **3.5.1 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!



### 3.5.2 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, such as:

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

### 3.5.3 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();  
});
```

### 3.5.4 Liquidier.js Configuration

Moreover, the user has various options to dynamically apply attributes as shown 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)

```
***/
```

### 3.5.5 Liquidier.js Configuration Example

The HTML mark-up for the responsive layout grid should look like 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>
```

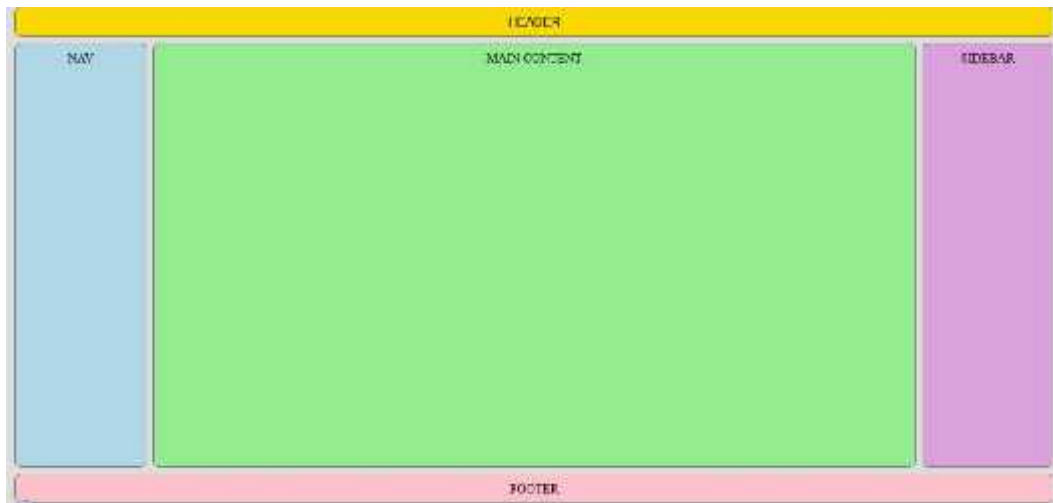
```
<footer id = "footer"> FOOTER </footer>
```

```
</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"  
    });  
});
```

The following output is generated as shown in Figure 10 below.



**Figure 12:** The Liquizer.js layout output

Therefore, above section answered the third research question; how to developed algorithm liquidizer.js?

### **3.6 Framework Validation and Verification**

The verification can be defined as the assessment of the accuracy of the solution to a computational model while validation as the assessment of the accuracy of computational simulation by comparison with experimental data. In verification, the relationship of the simulation to the real world is not an issue. In validation, the relationship between computation and the real world i.e. experimental data, is the issue(Oberkampf, Trucano, & Hirsch, 2002). This section initiates to answer the fourth research question of the study.

#### **Research Question**

- iv. What are the tools for algorithm testing, validation, and evaluation?

The algorithm Liquidizer.js framework developed is tested, validated and evaluated to ascertain its relevance in adaptability by web designers. A total of thirty (30) web designers are engaged in the review process of the framework. The review process is necessary to ensure that, an enhanced approached to responsive web design is achieved and the algorithm developed as the product of the study meets its intended

main objective of research of producing enhance approach to Responsive Web Design in Fluid Grid Concept.

### 3.6.1 Evaluation Method

In order to validate the framework tool, the following method is followed. The algorithm Liquidizer.js framework tool is given to web designers and shown how to use the framework. They are instructed to include a line of code in a script tag to reference both the Liquidizer.js and jQuery.mini.js. The instructions were executed by coding the following line; `<script type = "text/javascript" src = "jquery.min.js"></script>` and `<script type = "text/javascript" src = "Liquidizer.js"></script>`. To conduct the research table 3 shows the following web design households; Crablins Interactive, Jaffetek Computer Solutions and Mombasa Tech Community were engaged

**Table 4:** The Web Design Firms and Web Pages Tested

NO	Web Design Firms	Web Pages Tested
1	Crablins Interactive	Sea View Academy
2	Jafftek Computer Solutions	Almutwafy ICT Consultancy
3	Mombasa Tech Community	SportsVision Magazine

The Crablins Interactive; is a creative web design and optimization company based in Mombasa, Kenya. It designs and builds websites which depicts company's vision, brings out firms ideas and meets its business needs. It was chosen due to its rich portfolio in web design such as designing and developing in Lamu county website, Nawal center, shopping center in Mombasa with a customized e-commerce website, Shelter Tours, a Kenyan tour company, Sakafu Limited, a Kenyan floor safety company just to mention a few.

The Jafftek Computer Solutions; is a Kenyan based Software Development Company, focused on creating utmost qualitative, on time and cost effective software solution. It was chosen because of its development of successful online identities for various

organizations in Kenya. It has vast of experience of over ten (10) years in designing website solutions, which meet customer expectations. It has specialized in providing comprehensive website design services in e-commerce, magneto designs and in joomla framework platform.

The Mombasa-Tech Community; is a Community Based Organization (CBO) which focuses on inspiring and developing upcoming innovative technology ideas through networking, technical training and support, and professional mentoring and coaching. It was chosen because, it brings together techpreneurs, entrepreneurs, developers, designers, and investors to provide them with opportunities to learn, share knowledge, be mentored, and initiate novel ideas which will lead to the development of new technologies in Kenya and globally at large. The study targets its professional mentors and coaches web designers, to conduct the review process.

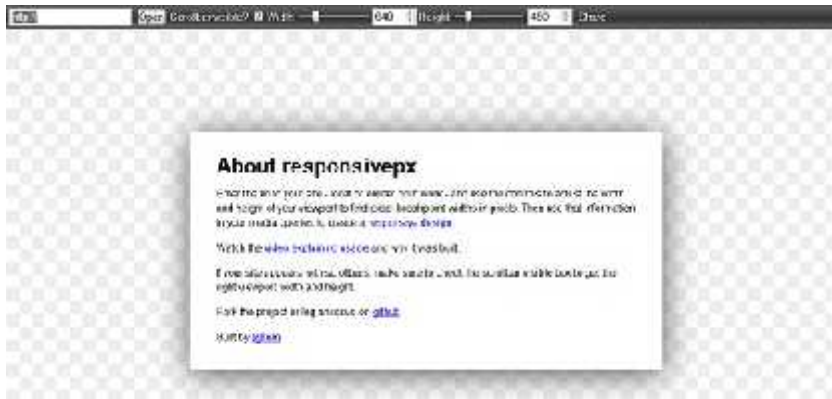
The web designers are then given a questionnaire to give their comments on observations they have made as regards the Liquidizer.js algorithm framework. The questions are based on software usability models.

### **3.7.0 Analysis and Testing Tools**

There exists various Analysis and testing tools which demonstrate how website responds to different screen and browser sizes implementing Responsive Web Designs. The two most commonly used are Responsivepx and Matt Kersley RWD Testing Tool.

#### **3.7.1 Responsivepx**

According to (responsivepx, n.d.), Responsivepx is a tool for testing responsive website design. The main feature that distinguishes it from others is its capability to resize the website pixel-by-pixel. This awesome feature will identify the breakpoints and also test how the CSS media queries are working in a website. It is an online tool, which can be accessed at <http://responsivepx.com/>. The Figure 13 below demonstrates how the interface of the Responsivepx works.



**Figure 13:** Interface of Responsivepx. Adapted from (responsivepx, n.d.)

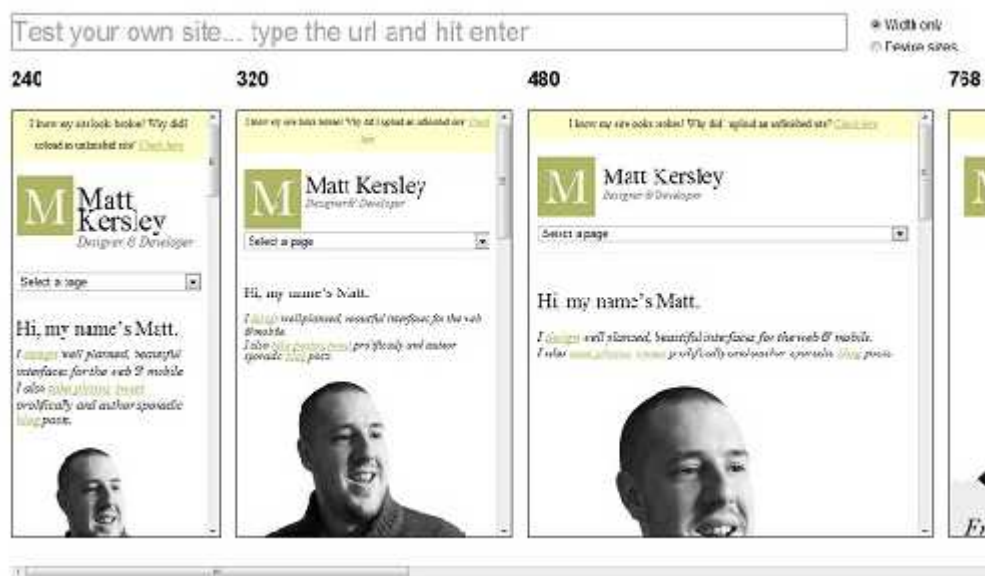
One simply, enters the URL website - local or online: both work - and use the controls to adjust the width and height of ones viewport to find exact breakpoint widths in pixels. Then one uses that information in one's media queries to create a responsive design. If the website appears with scrollbars, one should make sure to check the scrollbar visible box to get the right viewport width and height. The Figure 14 below demonstrates an example of a local host web site with a width of 700px and height of 567px.



**Figure 14:** Local host website demonstration

### 3.7.2 Matt Kersley RWD Testing Tool

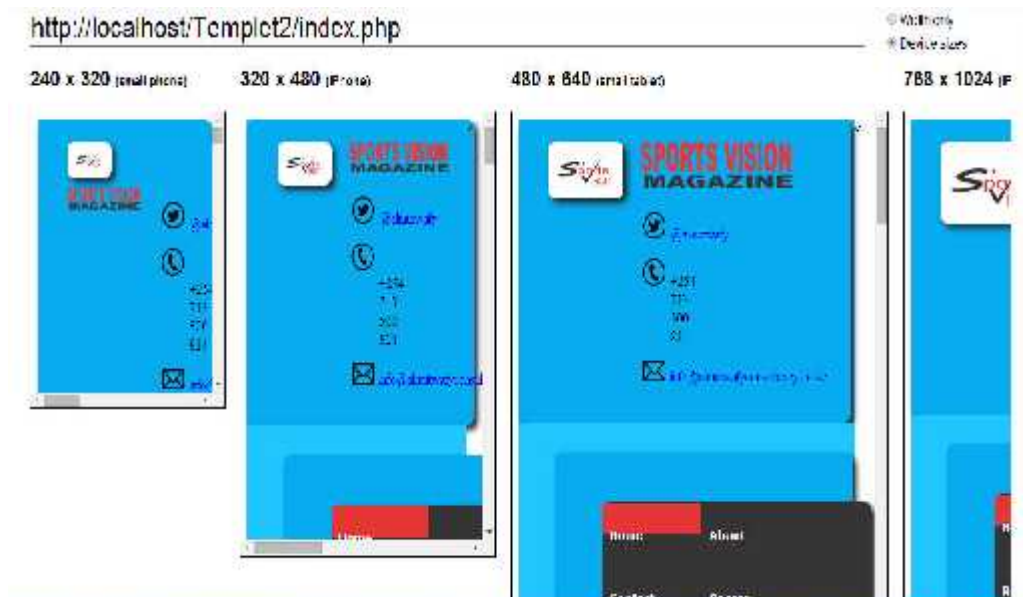
It is described by (Kersley, n.d.), as a testing tool that allows viewing responsive website in various screen sizes simultaneously in a single screen, while building or designing websites. The study prefers this tool mainly because it shows all the screen resolutions side-by-side which makes it easier for debugging. It is an online tool, which can be accessed at <http://mattkersley.com/responsive/>. The Figure 15 below demonstrates how the interface of the Testing Tool works.



**Figure 15:** Matt Kersley Default Interface Adapted from (Kersley, 2012)

The tool has been built to help with testing responsive websites while designing and build. One can enter website's URL into the address bar at the top of the page (not browser's address bar) to test a specific page. The Figure 16 below demonstrates how a local host website is display in various screen sizes simultaneously in a single screen.





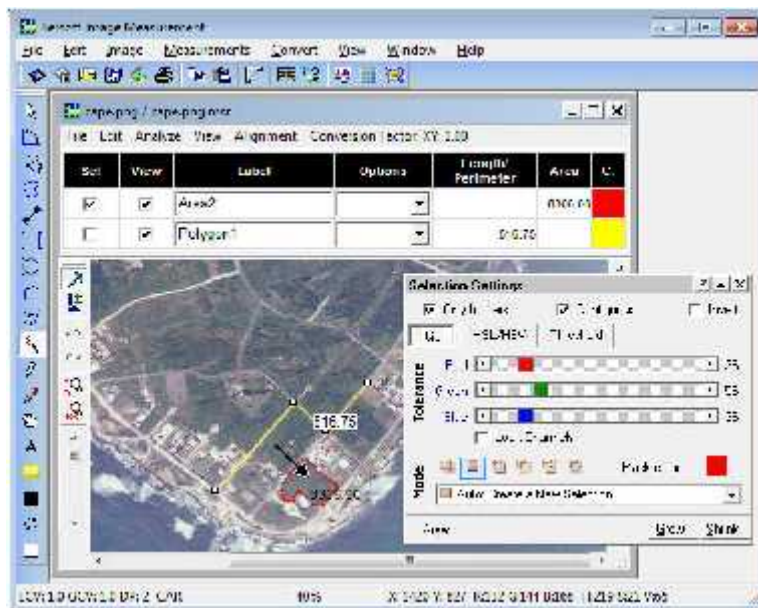
**Figure 16:** Local host website demonstration

Unfortunately, with the way browser security works, one is unable to navigate the website through the frames that website appears in. The only way this is possible is by hosting the testing tool on website's own host. Matt Kersley have provided at github repository for downloading and installation of the tool on any website.

It seems clear from the above discussion that Matt Kersley RWD Testing Tool has some limitation, however, its strength to display websites in various screen sizes simultaneously in a single screen make it suitable for this study.

### 3.7.3 Algorithm Liquidizer.js Validation

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. (BIM, 1995)



**Figure 17:** The BIM Interface adapted from (BIM, 1995)

### 3.7.4 Algorithm Registration for jQuery Library

This section is triggered to answer the fifth research question; what are the types of platform for algorithm publishing. It is reported by (jQuery, 2015) that, publishing of algorithm plugin on the their web site involves the following three step process:

1. **Adding a Service Hook-** Firstly, one's needed to enable the jQuery Plugins service hook on GitHub. On the settings page for one's repository, click the Webhooks and Services link, then click the Configure services button. Scroll down to find the jQuery Plugins service and enable it (there's no config, just check the Active checkbox and click the Update settings button).
2. **Adding a Manifest File to Repository** – Secondly, the jQuery Plugins Registry will look in the root level of ones repository for any files named \*.jquery.json. One will need to create \*ones-plugin\*.jquery.json according to the package manifest specification. One should use an online JSON verifier such as JSON-lint to make sure the file is valid.

- 3. Validate Manifest File**– Finally, one should upload the manifest file to check for common errors: Since this tool uses the new HTML5 File-Reader API to look at the file contents without actually uploading your file to the server.

The other alternative was to publish the results in various indexed journals such as the International Journals for Emerging Scientists (IJES), Computer Science Journals, and The International Journals for Engineering and Science etc. These journals have different requirement for document formatting, citation styles and general appearance of the manuscript to be published. The study adopted both the methods for publishing the algorithm plug in.

### **3.8 Liquidizer.js Repository**

The algorithm liquidizer.js is published at GitHub repository and can be access as an open source code at: <https://github.com/almutwafy/Liquidizer>. Liquidizer.js is licensed under the GNU General Public License version 1.0.0.

## CHAPTER FOUR

### 4.0 RESULTS FINDINGS

#### 4.1 Introduction

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 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, related works 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 questionnaire methods for testing and validating usability of a software; the Technology Acceptance Model (TAM) satisfaction questionnaire by (Davis, 1989) and Measuring Usability with USE questionnaire by (Lund, 2001) to test and validate the algorithm Liquidizer.js.

#### 4.2 TAM Satisfaction Questionnaire

The Technology Acceptance Model was created by Davis, 1989. The first six items measure perceived usefulness and the other five perceived ease of use. Both should explain use of a technology. Of this original simple version, several small variants in terms of wording exist. The items below were taken from (Davis, 1989)

1. Using [.....] in my job would enable me to accomplish tasks more quickly.
2. Using [.....] would improve my job performance.

3. Using [.....] would enhance my effectiveness on the job.
4. Using [.....] would make it easier to do my job.
5. I would find [.....] useful in my job.
6. Learning to operate [.....] would be easy for me.
7. I would find it easy to get [.....] to do what I want it to do.
8. My interaction with [.....] would be clear and understandable.
9. I would find [.....] to be flexible to interact with.
10. It would be easy for me to become skillful at using [.....].
11. I would find [.....] easy to use.

Response items use a 7-point likely - unlikely scale: extremely - quite - slightly - neither - slightly - quite – and extremely. More complex models also exist. These questions will measure the degree of perception of web developer's acceptance of the algorithm Liquidizer.js. Where the use of various framework have been readily accepted by web developers but their usability have discourage most to a large extend. Therefore, testing and validating usability is the scope of this study.

#### **4.3 Measuring Usability with the USE Questionnaire**

The questionnaire was developed over time and it started out with a large pool of items. It reported by (Lund, 2001) 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, Satisfaction, Ease of Learning and Ease of Use."

##### **A. Usefulness**

- i. It helps me be more effective.
- ii. It helps me be more productive.
- iii. It is useful.
- iv. It gives me more control over the activities in my life.
- v. It makes the things I want to accomplish easier to get done.

- vi. It saves me time when I use it.
- vii. It meets my needs.
- viii. It does everything I would expect it to do.

**B. Ease of Use**

- i. It is easy to use.
- ii. It is simple to use.
- iii. It is user friendly.
- iv. It requires the fewest steps possible to accomplish what I want to do with it.
- v. It is flexible.
- vi. Using it is effortless.
- vii. I can use it without written instructions.
- viii. I don't notice any inconsistencies as I use it.
- ix. Both occasional and regular users would like it.
- x. I can recover from mistakes quickly and easily.
- xi. I can use it successfully every time.

**C. Ease of Learning**

- i. I learned to use it quickly.
- ii. I easily remember how to use it.
- iii. It is easy to learn to use it.
- iv. I quickly became skillful with it.

**D. Satisfaction**

- i. I am satisfied with it.
- ii. I would recommend it to a friend.
- iii. It is fun to use.
- iv. It works the way I want it to work.
- v. It is wonderful.
- vi. I feel I need to have it.
- vii. It is pleasant to use.

The questionnaires were constructed as seven-point Likert rating scales, e.g. from -3 (totally disagree) to +3 (totally agree), (Lund, 2001). Therefore, the study adopts the USE Questionnaire for Measuring Usability to conduct a survey to evaluate the usability of the algorithm Liquidizer.js. The Appendix D presents a questionnaire based on Lund's USE questionnaire of seven (7) point likert scale on level of agreement. The likert scale is 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).

#### **4.4 Data Collection**

The study employed purposive sampling as its sampling technique to arrive at a sample of thirty (30) web designers. This concurs with a report by (Lund, 2012), who explains that, purposive sampling represents a group of different non-probability sampling techniques. Also known as judgmental, selective or subjective sampling, purposive sampling relies on the judgment of the researcher when it comes to selecting the units (e.g., people, cases/organizations, events, pieces of data) that are to be studied. Usually, the sample being investigated is quite small, especially when compared with probability sampling techniques

There are a wide range of purposive sampling techniques that one can use, such as Maximum variation sampling, Homogenous sampling, Typical case sampling, Extreme (or deviant) case sampling, Critical case sampling, Total population sampling and Expert sampling. The study adopted the homogenous sample as the most suitable for the study. It is further reported by (Lund, 2012) that, homogeneous sampling is a purposive sampling technique that aims to achieve a homogeneous sample; that is, a sample whose units (e.g., people, cases, etc.) share the same characteristics or traits (e.g., a group of people that are similar in terms of age, gender, background, occupation, etc.).

Therefore the study purposively selected a sample of ten (10) web designers from each of the three (3) software households of; Crablins Interactive, Jaffetek

Computer Solutions and Mombasa Tech Community to arrive at a sample of thirty (30) web designers. The web designers were instructed on how to use the Liquidizer.js. Moreover, the web designers were briefed on how to capture their perception on the Liquidizer.js using the questionnaires.

To validate the Liquidizer.js, the study conducted a survey to explore whether there is a relationship in the web designer's perception of the three dimensions of Usability; Usefulness, Satisfaction, and Ease of Learning (independent variables) and Ease of Use (dependent variable) of the Liquidizer.js. As a result of which the study presented the following hypothesis:

H<sub>0</sub> There is no relationship between the perception of dimensions of usability and ease of use of the Liquidizer.js

H<sub>1</sub> There is a relationship between the perception of dimensions of usability and ease of use of the Liquidizer.js

The questionnaires were immediately collected after the survey and analyzed using Statistical Package for Social Science (SPSS) version 20.

#### **4.5 Data Analysis**

To test the Liquidizer.js the study uses the Matt Kersley RWD Testing Tool. The tool presents visual output of the page in question either being responsive or not after or before implementation of the Liquidizer.js, by simultaneously simulating varying screen sizes of various devices.

To validate the Liquidizer.js the study uses Bersoft Image Measurement (BIM) tool. The tool is applied on both the generated digital images of frameworks of Liquidizer.js and BlocksIt.js algorithms. Three analytical tests of Lane Profile, Pixel Profile and descriptive statistics are conducted. The results are compared and interpreted.

To evaluate the Liquidizer.js descriptive statistics, correlation and regression analysis are conducted using Statistical Package for the Social Sciences (SPSS). The



questionnaires are inputted in SPSS to create a model. The model is use to output results of different analysis tests.

#### **4.6 Findings**

The research findings and results of the study were presented into threefold; first, the testing of the Liquidizer.js by the Matt Kersley RWD Testing Tool Results, second, the validation of the Liquidizer.js by Bersoft Image Measurement (BIM) tool Results and third, the evaluation of Liquidizer.js by Statistical Package for the Social Sciences (SPSS) Results. The SPSS tool conducted various tests to find relationship between variables, level of significance, reliability, and demographic data analysis. Specifically, the study used Cronbach's alpha test, descriptive statistics tests, chi-square, Pearson correlation and Regression analysis. As a result of this, the chapter is triggered and guided by the research question four of the study.

#### **Research Question**

- i. What are the tools for algorithm testing, validation, and evaluation?

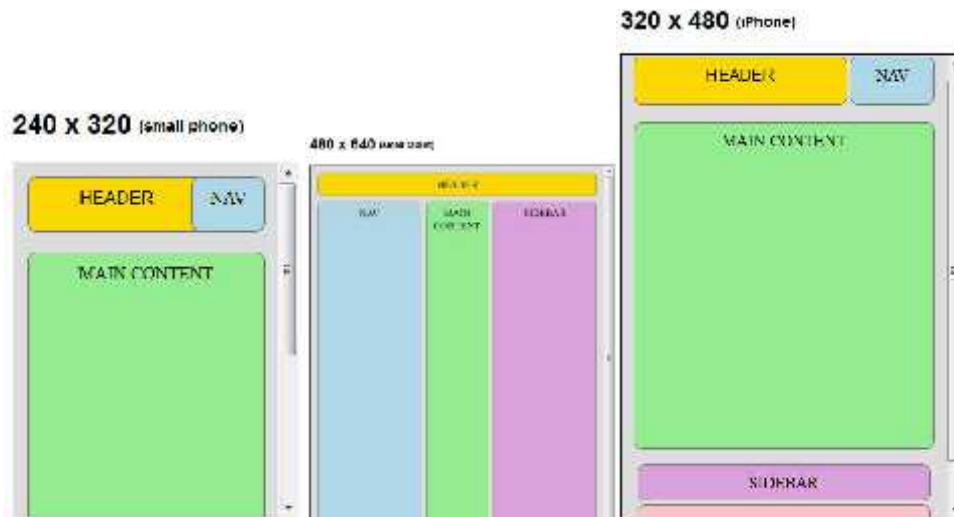
#### **4.7 Results for Liquidizer.js Testing**

The Matt Kersley RWD testing tool was used to simulate varying screen size from mobile phone to a main frame computer. The Figure 18 [a] presented the result before implementing the algorithm Liquidizer.js, while the Figure 18 [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 18 [b] is responsive since the page can be viewed in single view without the graphics or menu being distorted.

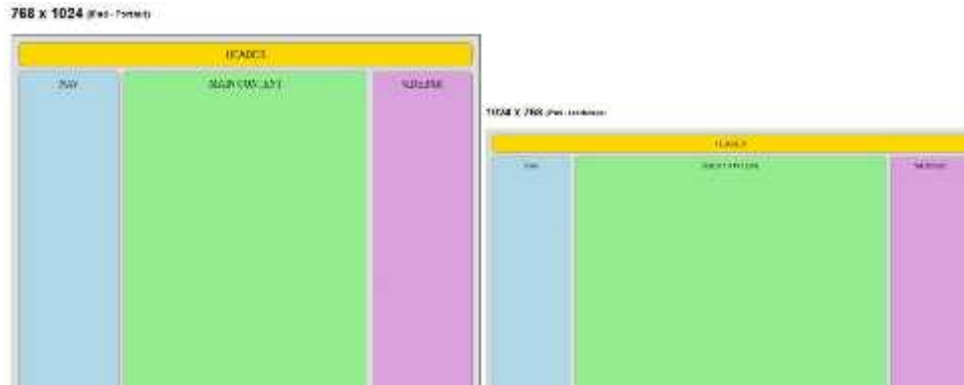


**Figure 18:** [a] Before implementing Algorithm [b] After implement Algorithm

Moreover, further responsive tests were conducted using Matt Kersley RWD testing and simulator tool on various screen sizes; 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 19 [a], 19 [b], 19 [c], 19 [d] and 19 [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 19:** [a] Smart Phone [b] Small Tablet [c] iPad



[d] iPad-Portrait      [e] iPad-Landscape

#### 4.8 Results for Liquidizer Validation

The study used Bersoft Image Measurement (BIM) tool to perform three validation analytical tests on the digital images generated by both Liquidizer.js and BlocksIt.js. The following are the tests which were conducted during validation.

- i. Lane Profile Test
- ii. Pixel Profile Test
- iii. Descriptive Statistics Test

##### 4.8.1 Lane Profile Test

The lane profile test analyses the pixel intensity against the pixel position. Both algorithms; the Liquidizer.js and BlocksIt.js's digital images were analyzed and their results compared against each other's. The purpose of the test is to compare the pixel intensity of the two algorithms. High pixel intensity indicates that the digital image is of high quality and non-distorted while; low pixel intensity indicates that the digital image is of low quality and distorted. The high quality and non-distorted is inferred to as responsive, while the low quality and distorted is inferred to as non-responsive.

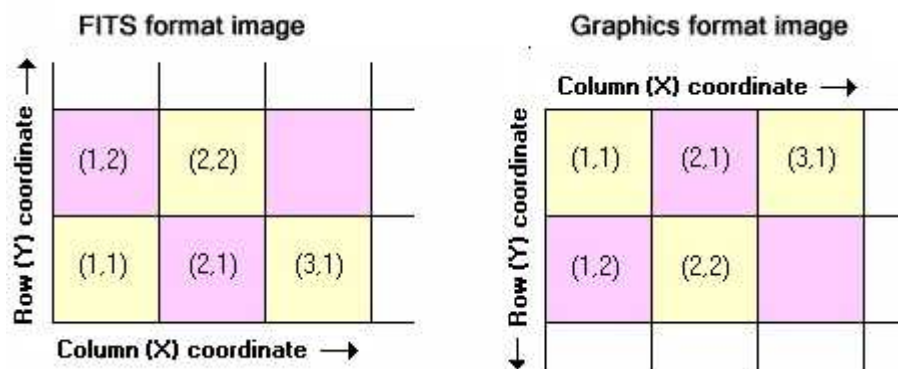
##### 4.8.1.1 Pixel Intensity

In an image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values. This histogram is a graph showing the number of pixels in an image at each different intensity value found in that image.

For an 8-bit gray-scale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those gray-scale values. Histograms can also be taken of color images either individual histogram of red, green and blue channels can be taken, or a 3-D histogram can be produced, with the three axes representing the red, blue and green channels, and brightness at each point representing the pixel count.

#### 4.8.1.2 Pixel Position

The pixel coordinate is a number that identifies the location of a pixel in the image array. There are various standard protocol for various format such as FITS and graphics format images. The Figure 20 below shows the standard coordinate protocols for FITS format image and Graphics format image.

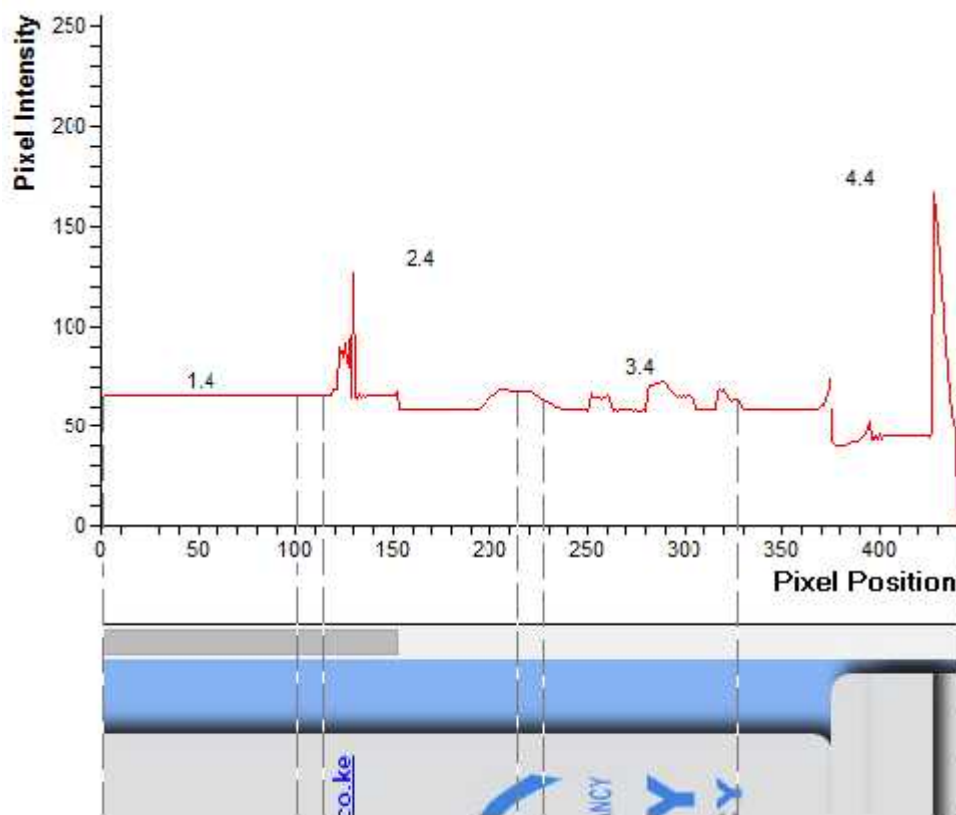


**Figure 20:** Standard Coordinate Format, adopted from (Fisher et al., 2003)

#### 4.8.1.3 Lane Profile Results

The lane profile was conducted using Bersoft Image Measurement (BIM) on the framework generated by BlocksIt.js. The results are shown in Figure 21 [a]. It is 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 low pixel intensity of (R,G,B) (55, 55, 55) (shade of black) which was fairly constant distributed at these regions. The result infers that, most of the pixels are cramped together at these ranges. The three colors; red, green and blue are not evenly

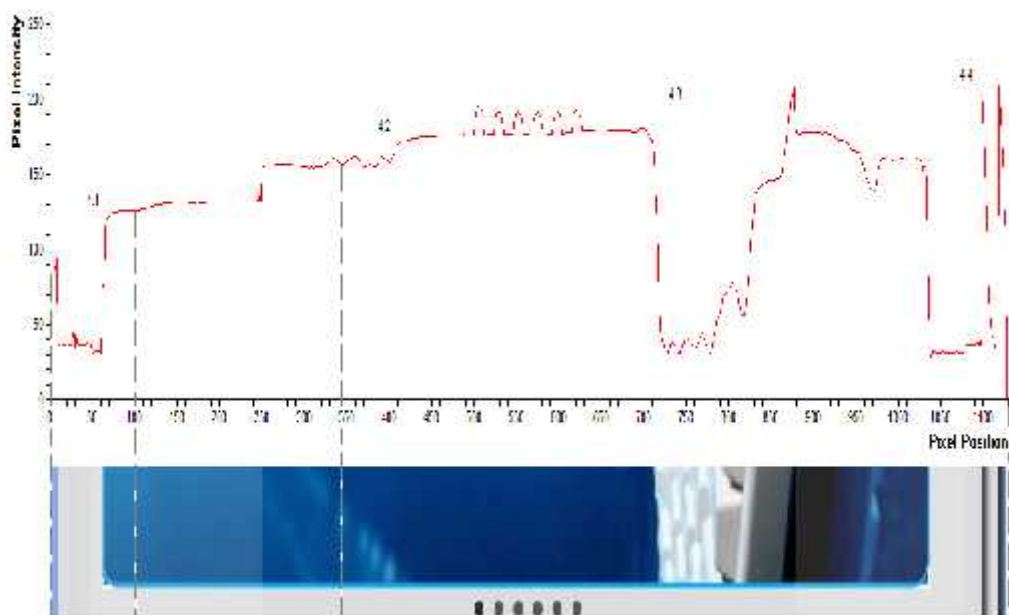
distributed hence displaying dark quality picture since the range values of pixel intensity are from (0,0,0) (black) to (128,128,128) (gray) to (255, 255,255) (white). Therefore, the results can be explained by the visual observation of the distorted picture framework, which is non-responsive. The non-responsive digital image will not fit in various screen sizes such as from mobile to framework computer without being distorted, while a responsive digital image will completely fit in various such screen sizes and be non-distorted.



**Figure 21:** [a] BlocksIt.js Lane Profile

The lane profile was also conducted on the framework generated by Liquidizer.js. The results are shown in Figure 21[b]. It is 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. A responsive digital image will completely fit in various such screen sizes and be non-distorted hence displaying a high quality image.



**Figure 21:** [b] Liquidizer.js Lane Profile

From the above two results of Figures 21 [a] and [b] observations, it can be clearly concluded that the image generated by Liquidizer.js is more responsive as compared to image generated by BlocksIt.js. The conclusion is arrived after analyzing the results of the two algorithms. The BlockIt.js results indicated that the image is of low quality and distorted as compared to that of Liquidizer.js indicating the image of high quality and non-distorted. Therefore this infers that the Liquidizer.js is more responsive as compared to BlocksIt.js algorithm.

## **4.8.2 Pixel Profile Test**

The pixel profile test evaluates the pixel values against the pixel distance of a digital image. Both the frameworks of Liquidizer.js and BlocksIt.js generated digital images which were evaluated for pixel profile.

### **4.8.2.1 Pixel Values**

In digital imaging, a pixel, or picture element is a physical point in a raster image, or the smallest addressable element in an all points addressable display device; so each of the pixels that represent an image stored inside a computer has a *pixel value* which describes how bright that pixel is, and/or what color it should be. To represent color images, separate red, green and blue components must be specified for each pixel (assuming an RGB color-space), and so the pixel 'value' is actually a vector of three numbers.(Fisher, Perkins, Walker, & Wolfart, 2003)

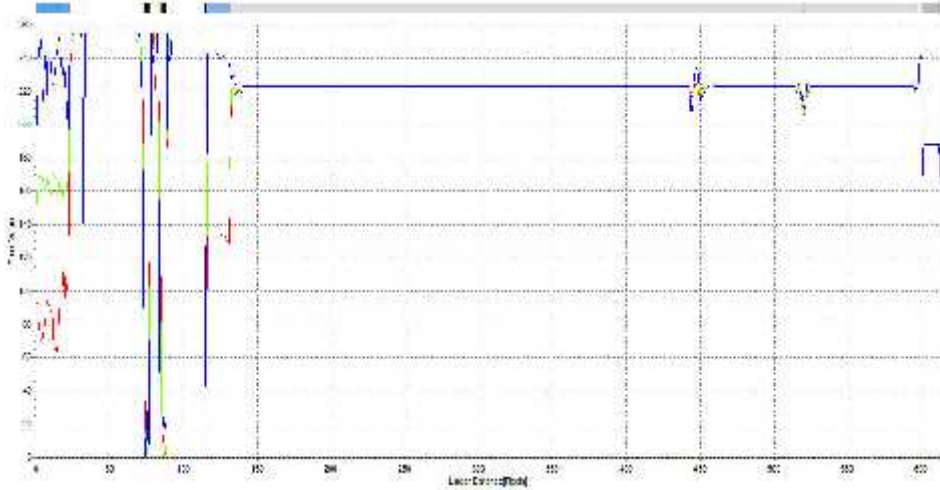
### **4.8.2.2 Pixel distance**

It is defined as the shortest m-path between the points. In this case, the distance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors (PU, 2013).

### **4.8.2.3 Pixel Profile Results**

The pixel profile was conducted using Bersoft Image Measurement (BIM) on the framework generated by BlocksIt.js. The results are shown in Figure 22 [a]. It is observed from the results of the slice pixel section of the digital image that the pixel distance between 150px – 600px, corresponded to the pixel value of 220 which was fairly constant at this region. The result infers that, most of the pixels are cramped together at this range. The three colors; red, green and blue are not evenly distributed hence displaying poor quality picture. Therefore, the results can be explained by the visual observation of the distorted picture framework, which is non-responsive. The non-responsive inferred that, web page framework will not automatically detect various devices' screen sizes when accessed by different users without page being

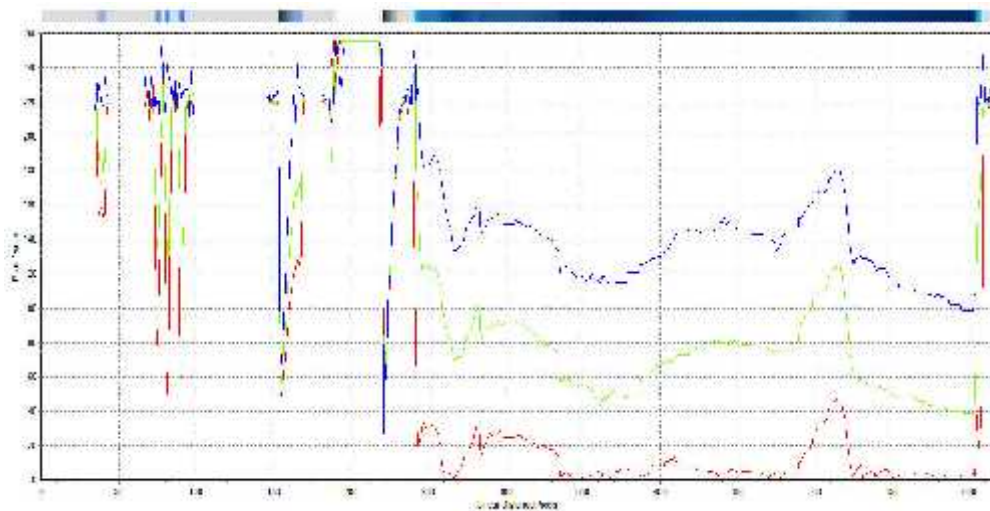
disfigured, while a responsive web page framework will automatically detect various devices' screen sizes and be non-disfigured.



**Figure 22:[a]** BlocksIt.js Pixel Profile

The pixel profile was also conducted on the framework generated by Liquidizer.js. The results are shown in Figure 22 [b]. 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**

**22:** [b] Liquidizer.js Pixel Profile

From the above two results of Figures 22 [a] and [b] observation it can be clearly concluded that the image generated by Liquidizer.js is more responsive as compared to image generated by BlocksIt.js

### **4.8.3 Results for Liquidizer.js Evaluation**

The descriptive statistics tests were conducted on both the algorithm Liquidizer.js and BlocksIt.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.

#### **4.8.3.1 Statistical Results**

The Table 5 shows various descriptive statistics test, the study had a particular interest on values of skewness. The results indicate that BlockIt.js had a skewness value of -2.07774911184475 which infers a large negative skew from the mean, media and mode as compared to Liquidizer.js skewness value of -0.488686363274612 which is a score near to zero inferring to near perfectly normal distribution. The normal distribution of the primary colors (RBG) indicated that the

pixel intensity is sparsely distributed on whole image. This inferred that a high quality image which is non-distorted. These results concurred with lane profile test above. 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 5 : Descriptive Statistics**

<b>Statistics</b>	<b>Liquidizer.js</b>	<b>BlocksIt.js</b>
N	737580	416021
MaxValue	169852	133517
Min	0	0
Max	255	255
Median	175	222
Mode	221	221
Mean	159.191962905719	213.507094593783
St. Dev	64.3144448743522	46.2189489446875
Skew	-0.488686363274612	-2.07774911184475
Kurtosis	1.93133960607452	8.51071458293398

#### **4.8.3.2 Conclusion**

The study conducted three tests of: lane profile, pixel profile and descriptive statistics to compare the framework digital images of Liquidizer.js against BlocksIt.js. The objective of these tests was to explore the quality of the framework digital images by analyzing three attributes of: pixel intensities, pixel values and pixel distributions. The higher values of these attributes indicate high quality image framework, which inferred to a responsive web page framework. A responsive web page will automatically adjust itself when viewed in any devices' screen size without being distorted. While lower values of these attributes indicate low quality image framework, which inferred to a non-responsive web page. In contrast a non-

responsive web page will not adjust itself when viewed in different devices' screen sizes hence the web page is viewed as distorted.

Therefore, from these two results observation, it can be clearly concluded that the image generated by Liquidizer.js is of better quality and more responsive as compared to image generated by BlocksIt.js

#### **4.9 Evaluation Results by SPSS**

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.

##### **4.9.1 Reliability Analysis**

The reliability of an instrument refers to its ability to produce consistent and stable measurements. The most common reliability coefficient is the Cronbach's alpha, which is a measure of internal consistency, that is, how closely related a set of items are as a group. A "high" value of alpha is often used (along with substantive arguments and possibly other statistical measures) as evidence that the items measure an underlying (or latent) construct. It takes values between 0 – 1, where 0 is the weakest and 1 the strongest.

Therefore, in this study to ensure the reliability of the instrument Cronbach's alpha is adopted as the reliability test of choice. All constructs depicted that the coefficient of Cronbach's Alpha above the suggested value of .70 (Note that a reliability coefficient of .70 or higher is considered "acceptable" in most social science research situations). The Table 6 shows the output of Cronbach's Alpha values of .831 for 35 construct items suggesting a strong value of 83.1% acceptable instrument. The result infers that the questionnaire is highly reliable to capture data that is consistent, stable and statistical reliable.

**Table 6:**Reliability Statistics

Cronbach's Alpha	N of Items
.831	35

**4.9.2 Demographic Data**

The study sought to establish the demographic data of the respondents. It begun by a general analysis on the demographic data from the respondents which included; the respondent’s position held at the organization, duration of stay in the organization, number of project developed, if they use framework in development and reason not using a framework. The study purposely targeted 30 respondents who participated to respond to the questionnaires.

**4.9.2.1 Position held at the organization**

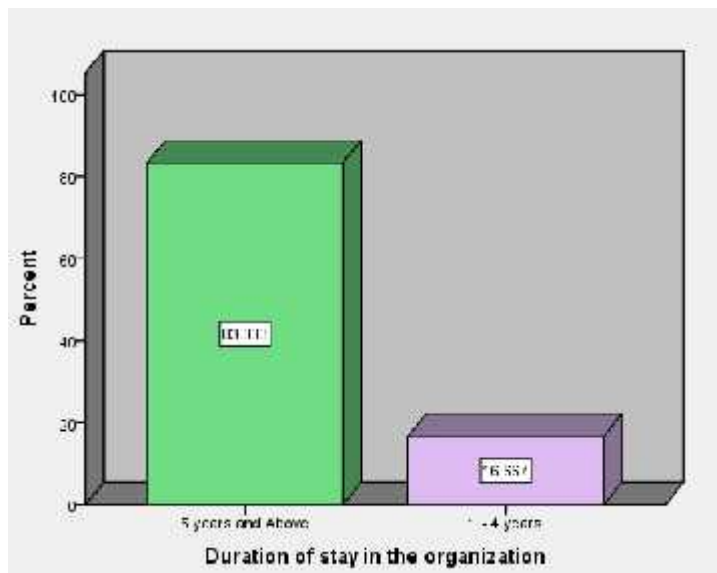
In the survey, the respondents were asked to state their position held at the organization. Out of the 30 respondents, 27 (90%) of the respondents were web designers, and only 3 (10%) of the respondents were software developers. The result illustrates that in every 10 positions held in most software house 9 were web designer, which concurs with importance of the design phase in SDLC. This higher level of web designer also concurs with the study’s purposive sampling type, where the study specially targeted the web designers. These results were generated using descriptive statistics in SPSS and Table 7 shows the output of the respondent’s positions held at their respective organization.

**Table 7:** Position held at the organization

	Frequency	Percent	Valid Percent	Cumulative Percent
Web Designer	27	90.0	90.0	90.0
Software Developer	3	10.0	10.0	100.0
Total	30	100.0	100.0	

#### 4.9.2.2 Duration of stay in the organization

The study investigated the duration of stay in the organization of the respondents. The descriptive statistics showed that majority 25 (83.3%) of the target respondents stay in the organization between 5 years and above, and only 5 (16.7%) between 1 – 4 years as shown in Figure 23 of the duration of stay in the organization. The result infers that more than  $\frac{3}{4}$  of the employees of the software house stay in the organization between 5 years and above. This could be attributed to the fact that software houses depend heavily on specialized technical personnel, who stay long duration in an organization in order to achieve a competitive advantage over their competitors.

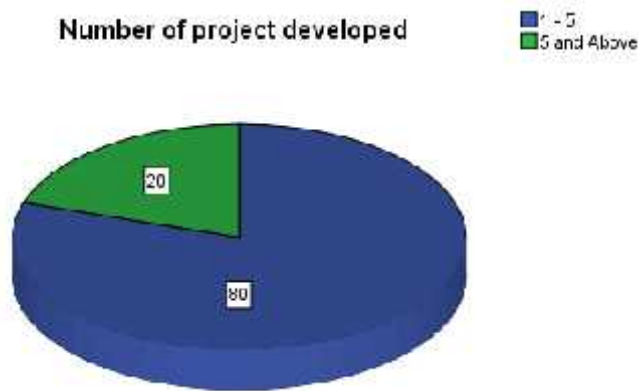


**Figure 23:** Duration of stay in the organization

#### 4.9.2.3 Number of Project Developed

The descriptive statistics generated by SPSS output showed that, twenty four (80%) of the respondents asserted that have developed between 1 – 5 projects and only six (20 %) between 5 years and above. The findings indicate that, more than  $\frac{3}{4}$  of respondents only developed between 1 – 5 projects during their stay in the organization of between 5 years and above, which is account for about one project a year. This is very low number of project development as compare to the developed world. These results infer that, majority of the software customers in this part of the

world, would not acquire customized developed software due to various reasons. The most prevalent reason is that, the customers usually acquire software over the shelf. The study findings are presented on Figure 24 below.



**Figure 24:** Number of Project Developed

#### 4.9.2.4 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 8 Do you use framework for design and development.

**Table 8:** 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 ( $H_1$ ) 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 9 demonstrates the chi-square test statistics of the construct

**Table 9:** Test Statistics

Do you use framework for development	
Chi-Square	13.333 <sup>a</sup>
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 infer that three quarters of the respondent do not use framework for development and design, which concur with the problem statement of this study. The result also concur with the report by(Knight, 2009), 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.

#### **4.9.2.5 Reason not using a framework**

The study sample 30 respondents, and evaluated reasons for not using a framework for web design and development. The findings showed that, 26 (86.7%) of the respondents asserted that it was hard to integrate the framework, 3 (10%) had no particular reason and 1 (3.3%) reported as time consuming. The data was analyzed using descriptive statistics by computing a frequency Table 10 what are reasons of not using a framework. The inferential statistics indicate that, majority of the respondent find it hard to integrate the framework during designing and development. Therefore, it can be deducted that, more than 80% of the respondents asserted that the major reason of not using the framework was hard to integrate the framework. These results concur with the problem statement of this study that, the web designers find it hard to customized frameworks.

**Table 10:** What are reasons of not using a framework?

	Frequency	Percent	Valid Percent	Cumulative Percent
None	3	10.0	10.0	10.0
Hard to integrate	26	86.7	86.7	96.7
Time Consuming	1	3.3	3.3	100.0
Total	30	100.0	100.0	

#### 4.9.2.6 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 Learning (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 11 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 11:** Respondent’s Perception of the survey

<b>Usability Determinates</b>	<b>N</b>	<b>Mean</b>	<b>Survey Value</b>	<b>Survey Label</b>
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
<b>Grand Perception</b>	<b>30</b>	<b>6.44</b>	<b>6</b>	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

#### **4.9.3 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.

The interpretation of correlation, needs four types of information; correlation coefficient value, correlation coefficient sign, correlation significance, and correlation effective size. 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. The sign of the correlation coefficient means either a positive or negative correlation coefficient. The positive correlation coefficient

means that the variables move in the same direction, while negative correlation means variables move in opposite directions. The correlation significance is indicated by a probability value of less than 0.05. This means that the probability of obtaining such a correlation coefficient by chance is less than five times out of 100, so the result indicates the presence of a relationship. Finally, the correlation effective size is the coefficient of determination and is defined as  $r^2$ . The coefficient of determination can vary from 0 to 1.00 and indicates that the proportion of variation in the values can be predicted from the relationship between the two variables.

Therefore, the study conducted correlation analysis and computed four usability determinates of; Usefulness, Satisfaction, Ease of learning (independent variables) which positively influence Ease of Use (dependent variable) of the Liquidizer.js for 30 respondents. The results are shown in Table 12 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 learning (independent variables) and Ease of Use (dependent variable). The increase of Usability determinates, increases the Ease of Use. The Table 12 summarizes the correlation output computed by SPSS Version 20 for Usability Determinates of Ease of Use.

**Table 12:** Correlations for Usability Determinates of Ease of Use

		Satisfactio n	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.

#### 4.9.4 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 Learning 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.

#### 4.9.4.1 Outliers Test

The study evaluated the Minimum and Maximum values of Std. Residual (Standardized Residual) subheading in Table 13 Residuals Statistics. If the minimum value is equal or below -3.29, or the maximum value is equal or above 3.29 then one have outliers. An analysis of standard residuals was carried out, which showed that the data contained no outliers (Std. Residual Min = -1.996, Std. Residual Max = 1.900).

**Table 13:** Residuals Statistics<sup>a</sup>

	Minimum	Maximum	Mean	S.Deviation	N
Predicted Value	5.71	6.75	6.43	.281	30
Residual	-.152	.145	.000	.072	30
Std. Predicted Value	-2.590	1.144	.000	1.000	30
Std. Residual	-1.996	1.900	.000	.947	30

a. Dependent Variable: Ease of Use

#### 4.9.4.2 Collinearity Test

In order the data to meet the assumption of collinearity, the study evaluated the Coefficients Table 14 in the regression analysis output results. The interpretations

of the heading Collinearity Statistics, under which are two subheadings, Tolerance and VIF (Variance Inflation Factor) are analyzed. If the VIF value is greater than 10, or the Tolerance is less than 0.10, then one has concerns over multi-collinearity. Otherwise, one's data has met the assumption of collinearity.

The study explored to see if the data met the assumption of collinearity, the results indicated that multi-collinearity was not a concern (Satisfaction, Tolerance = .595, VIF = 1.680; Ease of Learning, Tolerance = .687, VIF = 1.457; and Usefulness, Tolerance = .505, VIF = 1.979).

**Table 14: Coefficients<sup>a</sup>**

Model	Unstandardized Coefficients		Standardized Coefficients	Collinearity Statistics	
	B	Std. Error	Beta	Tolerance	VIF
(Constant)	.633	.339			
1 Satisfaction	.615	.063	.620	.595	1.680
Ease of Learning	.370	.036	.602	.687	1.457
Usefulness	-.086	.064	-.092	.505	1.979

a. Dependent Variable: Ease of Use

#### 4.9.4.3 Independent Errors Test

In order to check see if the data residual terms are uncorrelated then the study evaluated the Table 15 Model Summary by examining the Durbin-Watson value. Durbin-Watson values can be anywhere between 0 and 4, however what one looks for is a value as close to 2 as one can get in order to meet the assumption of independent errors. As a rule of thumb if the Durbin-Watson value is less than 1 or over 3 then it is counted as being significantly different from 2, and thus the assumption has not been met. The study tested to see if the data met the assumption of independent errors, the results indicated that the data met the assumption of independent errors (Durbin-Watson value = 2.184).

**Table 15:** Model Summary

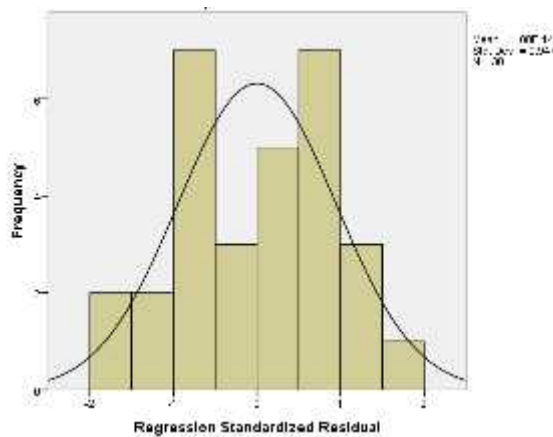
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics					Durbin Watson
					R Square Change	F Change	df1	df2	Sig. F Change	
1	.969 <sup>a</sup>	.938	.931	.076	.938	131.101	3	26	.000	2.184

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

b. Dependent Variable: Ease of Use

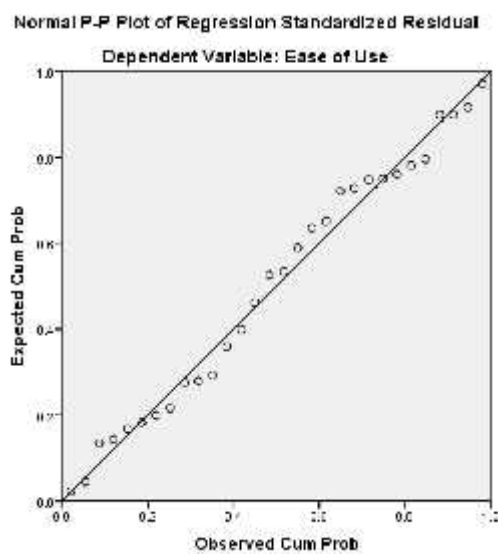
#### 4.9.4.4 Random Normally Distributed Errors Test

The random normally distributed errors analysis and evaluation are done visually, where the results became more of an art than a science as one need to look at some graphs and decide, if they meet the various assumptions. In the case of a Histogram one should have a nice looking normal distribution curve superimposed over a bar chart of the data. If one's data do, then this means that the data has met the assumption of normally distributed residuals, otherwise has not. The study examined Figure 25 and observed that the histogram of standardized residuals indicated that the data contained approximately normally distributed errors, hence met the assumption of random normally distributed errors.



**Figure 25:** Histogram Regression Standardized Residual

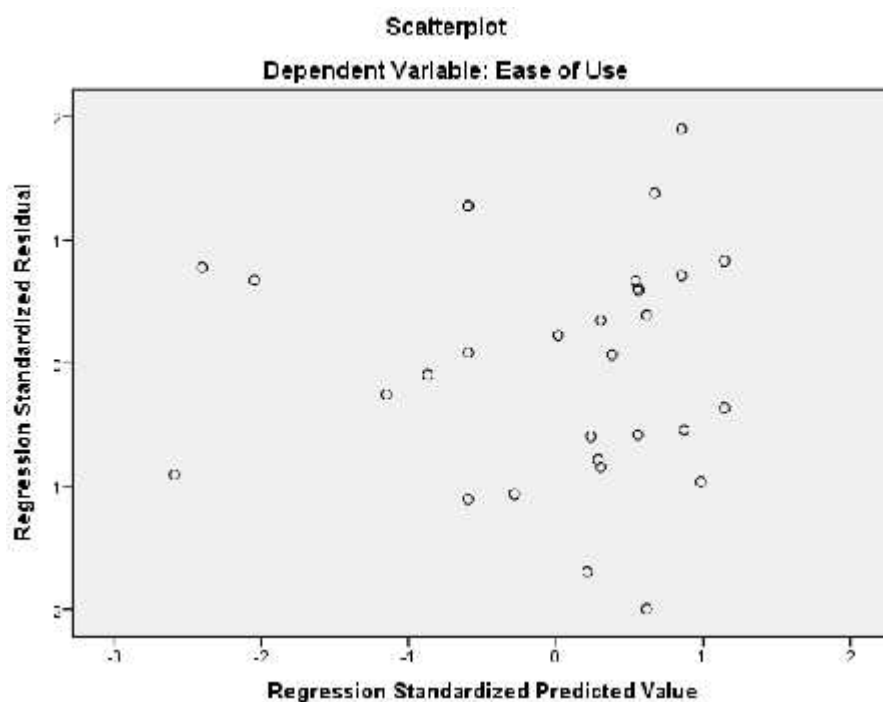
The study further, explored the Normal P-P Plot of Regression Standardized Residual of Figure 26, by analyzing if the dots are on, or close, to the line running diagonally across the graph. If ones data do, then this means that the data has met the assumption of normally distributed residuals, otherwise has not. The study explored Figure 26 and observed that the Normal P-P Plot of standardized residuals indicated that the data showed points that were not completely on the line, but close to it hence can be deduced to contain approximately normally distributed errors.



**Figure 26:** Normal P-P Plot of Regression Standardized Residual

#### 4.9.4.5 Homoscedasticity & Linearity Test

In order to meet the homoscedasticity and linearity, the study evaluated the scatterplot, which should present dots closely clustered together with very few outliers. Furthermore, if an imaginary linear line is drawn, it should be close to the cluster of dots. The study explored Figure 27 and observed the scatterplot of standardized predicted values showed that the data met the assumptions of homogeneity of variance and linearity, as the dots are closely clustered and a linear imaginary line could be easily drawn.



**Figure 27:** Scatter Plot of Regression Standardized Predicted Value

#### 4.9.4.6 Non-Zero Variances Test

To meet the non-zero variances, the study sampled 30 respondents by computing descriptive statistics, and evaluated the variance on usability determinates of; Usefulness, Satisfaction, and Ease of Learning of Ease of Use of the Liquidizer.js The data was analyzed to determine whether the values of the variances are over zero or not. If the values are non-zero then the assumption is met. The Table 16 showed descriptive statistics output where the data met the assumption of non-zero variances



(Satisfaction, Variance = .085; Ease of Learning, Variance = .223; Ease of Use, Variance = .084; and Usefulness, Variance = .097).

**Table 16:** Descriptive Statistics

	N	Mean	Std. Deviation	Variance
Satisfaction	30	6.50	.292	.085
Ease of Learning	30	6.37	.472	.223
Ease of Use	30	6.43	.290	.084
Usefulness	30	6.45	.311	.097
Valid N (listwise)	30			

#### 4.9.5 Reporting Multiple Regression

A multiple regression was conducted to see if the usability determinates of; Usefulness, Satisfaction, and Ease of Learning 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

##### 4.9.5.1 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 Learning, are able to predict Ease of Use of Liquidizer.js. The first measure in the Table 15 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 15 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  $F(3, 26) = 131.101, p < .05, R^2 = .938, R^2_{Adjusted} = .931$ .

#### **4.9.5.2 ANOVA**

Analysis of variance (ANOVA) is a collection of statistical models used to analyze the differences between group means and their associated procedures (such as "variation" among and between groups), developed by R.A. Fisher. In the ANOVA setting, the observed variance in a particular variable is partitioned into components attributable to different sources of variation. In its simplest form, 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. The F test has two numbers for its degrees of freedom. These are called the numerator and denominator degrees of freedom. In the Table 16 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 17 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 17:** Analysis of Variance - ANOVA<sub>a</sub>

<b>Model</b>	<b>Sum of Squares</b>	<b>Df</b>	<b>Mean Square</b>	<b>F</b>	<b>Sig.</b>
1 Regression	2.290	3	.763	131.101	.000 <sup>b</sup>
Residual	.151	26	.006		
Total	2.441	29			

a. Dependent Variable: Ease of Use

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

### 4.9.5.3 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 3 shows the regression equation below.

---

$$\text{Predicted variable (dependent variable)} = \text{slope} * \text{independent variable} + \text{intercept}$$

---

### Equation 3: 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 18 Coefficients presented the three predictor variables (Usefulness, Ease of Learning and Satisfaction). Therefore, a linear regression model with three predictor variables can be expressed with the following equation:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + E.$$

The variables in the model are Y, the dependent variable (Ease of Use); X<sub>1</sub>, the first predictor independent variable (Usefulness); X<sub>2</sub>, the second predictor variable (Ease of Learning); X<sub>3</sub>, the third predictor variable (Satisfaction); and E, the residual error, which is an unmeasured variable.

The parameters in the model are  $\beta_0$ , the Y-intercept (Constant = .633);  $\beta_1$ , the first regression coefficient (Satisfaction = .615);  $\beta_2$ , the second regression coefficient (Ease of Learning = -.370); and  $\beta_3$ , the third regression coefficient (Usefulness = -.086).

The study modeled of the Ease of Use(Y) based on the Usability determinates in the Usefulness (X<sub>1</sub>), Ease of Learning (X<sub>2</sub>), and Satisfaction (X<sub>3</sub>). Therefore, the regression equation was formulated as follows:

$$Y = 0.633 + 0.615 X_1 + 0.370 X_2 + (-0.086) X_3$$

**Table 18:** Coefficients

Model	Unstandardized Coefficients		Standardized Coefficients	T	Sig.
	B	Std. Error	Beta		
(Constant)	.633	.339		1.867	.073
1 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

#### 4.9.5.4 Interpreting the Intercept ( $\beta_0$ )

The Y-intercept (  $\beta_0$  ), can be interpreted as the value one would predict for Y if X<sub>1</sub> = 0, X<sub>2</sub> = 0, X<sub>3</sub> = 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<sub>1</sub>=

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  $\beta_0$  really has no meaningful interpretation. It just anchors the regression line in the right place.

#### 4.9.5.5 Interpreting Coefficients of Categorical Predictor Variables

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 result are presented in Table 19 Usability Prediction of Liquidizer.js.

**Table 19:** Usability Prediction of Liquidizer.js

Constant	Satisfaction	Ease of Learning	Usefulness	Prediction Value	Scale Value
*0.633	0.615*(7)	0.370*(0)	-0.86*(0)	4.938	5 = Slightly Agree
0.633	0.615*(0)	0.370*(7)	-0.86*(0)	3.223	3 = Slightly Disagree
0.633	0.615*(0)	0.370*(0)	-0.86*(7)	- 5.387	5 = Slightly Agree
**0.633	0.615*(7)	0.370*(7)	-0.86*(0)	7.528	7 = Strongly Agree
0.633	0.615*(7)	0.370*(0)	-0.86*(7)	- 1.082	1 = Strongly Disagree
0.633	0.615*(0)	0.370*(7)	-0.86*(7)	- 2.8	3 = Slightly Disagree
0.633	0.615*(0)	0.370*(0)	-0.86*(0)	0.663	1 = Strongly Disagree
0.633	0.615*(7)	0.370*(7)	-0.86*(7)	1.508	2 = Disagree

\*\*\*Strongly Agree

\*Slightly Agree

The prediction results showed that, when the variable Satisfaction is given the highest value of 7 representing Strongly Agree in the likert scale, while the rest of the variables are constant i.e. equated to  $x$  value = zero then, its prediction value is 4.938 which is approximately 5 in the likert scale, representing Slightly Agree for Liquidizer.js. It was also noted that, when the two variables of Satisfaction and Ease of Learning were given the highest values of 7 each representing Strongly Agree in the likert scale, while the variable Usefulness is constant i.e. equated to  $x$  value = zero then, their prediction is 7.528 which is approximately 7 in the likert scale, representing Strongly Agree for Liquidizer.js.

The inferential statistics shows that, although the combination of variables Satisfaction and Ease of Learning represent Strongly Agree in the likert scale but the value actually predicts a value approximately 8 which is not presented the likert scale. Moreover, the study seeks a possibility of a single variable which will be a game changer i.e. a single variable when increased will positively influence usability of Liquidizer.js to the maximum.

Therefore, the inferential statistics of the variable Satisfaction is 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 to be readily accepted by the web designers then, more efforts should be put on the variable Satisfaction to maximize the usability of the Liquidizer.js. Moreover, these results trigger the solution of the problem statement of this study that, the web designers will not use the frameworks due to difficulties of heavy customization of the frameworks. The Liquidizer.js provides this solution due to its simplicity of usability as shown by the survey results.

## CHAPTER FIVE

### CONCLUSIONS AND RECOMMENDATIONS

#### 5.0 Introduction

Internet technology is dynamically changing at the speeds of light. It has evolved human race from science fictions to science facts. 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 changed our livelihood activities, while superior techniques are currently emerging by the hour. As a result of these fluid changes, the study was intrigued by the Responsive Web Design (RWD) technology. As a result of this, the chapter discusses and elaborates the study's summary, recommendation and further research gaps which were not address by the study.

#### 5.1 Summary

Lots of brains have been poured on the concept of RWD especially in areas of conversion of website's fixed grid layouts to fluid grid layouts. However, these approaches involved heavy customization, resulting into low adaption by web designers. Therefore, for these reasons the study was intrigued and motivated to close this gap by presenting an alternative approach that will be flexible and accessible by a single line of code. As a result of which, study developed, tested and validated an algorithm coined Liquidizer.js in a jQuery Framework. The findings showed that, the variable Satisfaction as the game changer, whereby it's maximum impact will positively affect the usability of the algorithm Liquidizer.js to be readily adopted by the web designers. Finally the study's road map was guided by the following five research questions: what are the existing technologies in RWD, what are the methodologies for algorithm design, what are the techniques for algorithm development, what are the tools for algorithm testing, validation, and evaluation, and what are the requirements for algorithm registration in jQuery library?

## **5.2 What are existing technologies in RWD**

The first research question was answered by exploring the application of RWD technology, with a particular interest in fluid grid concept through in depth desk literature survey of start-of-art publication, books, magazines, blogs etc. The survey's outcome was a recommendation of a classification of three meaningful categories of fluid grid concept's solutions as; Frame-based Solution (FBS), Support-based Solution (SBS), and Algorithm-based Solutions (ABS). As a result of which, the study was guided to adopt an Algorithm-based Solution (ABS) approach for fluid grid implementation. Moreover, the literature survey was published by The International Journal of Engineering and Science (IJES), title "Responsive Web Design in Fluid Grid Concept: Literature Survey. Furthermore, the paper can be reference online at: [www.theijes.com/papers/v3-i7/Version-3/G0373049057.pdf](http://www.theijes.com/papers/v3-i7/Version-3/G0373049057.pdf).

### **5.2.1 What are the methodologies for algorithm design**

In order to achieve these second research question, the study conducted a literature survey in its depth, whereby various methodologies for algorithm design were explored. The investigation was conducted specifically on flowcharts, pseudo codes and stepwise refinement. After extensive exploration of the three methodologies, the study adopted the stepwise refinement despite its limitation of refinement alteration and modification as the most fitting methodology for the study.

### **5.2.2 What are the techniques for algorithm development**

In order to achieve the third research question, the algorithm physical and logical designs were used to build the algorithm by coding within jQuery platform. The algorithm was developed using stepwise refinement technique, then debugged, tested and implemented. The outcome of the study was a publication titled "Liquidier.js: A Responsive Web Design Algorithm" by The International Journal of Engineering and Science (IJES), which can be accessed online at: [www.theijes.com/papers/v3-i7/Version-3/G0373049057.pdf](http://www.theijes.com/papers/v3-i7/Version-3/G0373049057.pdf)



### **5.2.3 What are the tools for algorithm testing, validation and evaluation**

The study explored the related works on algorithm testing, evaluation and validation tools documented by various researchers. To answer the fourth research question, the study adopted the Matt Kersley RWDTool: for testing Liquidizer.js, Bersoft Image Measurement (BIM) tool: for validating Liquidizer.js, and the USE questionnaire for measuring usability: for evaluation of the Liquidizer.js algorithm. A truth ground group of web design's software houses of Crablinks Interactive, Jaffetek Computer Solutions and Mombasa Tech Community were engaged in the survey for evaluation and validation of the algorithm. Moreover, the study analyzed various questionnaire models and adapted the USE Questionnaire Model to collect the primary data from the web designers. The outcome of the survey revealed that, the variable Satisfaction was a game changer for maximum adoption of the algorithm by the web designers. The outcome of the thesis was a publication by The international Journal of Science and Engineering Research (IJSER), title "An Enhanced Approach to Responsive Web Design in Fluid Grid Concept, which can be access online at:

<http://www.ijser.org/onlineResearchPaperViewer.aspx?An-Enhanced-Approach-to-Responsive-Web-Design-in-Fluid-Grid-Concept.pdf>

### **5.2.5 What are the requirements for algorithm registration in jQuery**

The last and the fifth research question was achieved by exploring various methods for algorithm publishing such as the jQuery library platform and various academic indexed journals in computing and engineering. The jQuery library platform involved only three step process of; adding a service hook at GitHub repository, adding a manifest file to repository account and validating the Manifest File. While publishing in journals procedures had varying requirements and formats of manuscripts for each individual publisher. The study adopted both the methods in order to increase the scope of the ease of accessibility of the algorithm Liquidizer.js. The algorithm liquidizer was published at GitHub and can be accessed at URL; <https://github.com/almutwafy/Liquidizer>.

## **5.3 Recommendations**

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 incorporate 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 study recommended the adaption of the classification of three categories of fluid grid concept solution as; Frame-based Solution (FBS), Support-based Solution (SBS), and Algorithm-based Solutions (ABS).

#### **5.4 Further Research**

In the future studies the algorithm can be improved by considering development of algorithm using CSS only, since as a layout language, CSS is more creative and flexible in design as compared to JavaScript. Moreover, the study recommended the optimization of the algorithm to increase the page loading time. 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

- Abran, A., Khelifi, A., Suryan, W., & Seffah, A. (2003). Usability meanings and interpretations in ISO standards. *Software Quality Journal*, 11(4), 325–338.
- Allsopp, J. (2000). A Dao of Web Design. *A LIST APART*. Retrieved from <http://alistapart.com/article/dao>
- Alsup, M. (2007). A Plugin Development Pattern » Learning jQuery - Tips, Techniques, Tutorials. Retrieved from <http://www.learningjquery.com/2007/10/a-plugin-development-pattern/>
- Beal, V. (2013). What is Cascading Style Sheets (CSS)? Webopedia. Retrieved from <http://www.webopedia.com/TERM/C/CSS.html>
- Bevan, N. (1995). Measuring usability as quality of use. *Software Quality Journal*, 4(2), 115–130.
- Bhatnagar, P. (2009, March 20). Formal Designing Techniques-Pranav: Advantages and Disadvantages of Flowchart. Retrieved from <http://formaldesign-techniques.blogspot.com/2009/03/advantages-and-disadvantages-of.html>
- BIM. (1995). Bersoft Image Measurement. Retrieved from <http://image-measurement.com/>
- Boudreaux, R. (2012a). How to create media queries in Responsive Web Design. Retrieved from <http://www.techrepublic.com/blog/web-designer/how-to-create-media-queries-in-responsive-web-design/>
- Boudreaux, R. (2012). How to style flexible images for RWD. Retrieved from <http://www.techrepublic.com/blog/web-designer/how-to-style-flexible-images-for-rwd/>

- Burge, J., & Brown, D. . (2002). Integrating Design Rationale with a Process Mode. In *Workshop on Design Process Modelling, Artificial Intelligence in Design*. UK: Cambridge. Retrieved from <http://handouts.eas.miamioh.edu/burgeje/research/AIDPaper.pdf>
- Carroll. (2014). Research Designs. Retrieved May 18, 2014, from <http://www.dissertation-statistics.com/research-designs.html>
- Chapman, C. (2009). The Evolution of Web Design. Retrieved from [http://sixrevisions.com/web\\_design/the-evolution-of-web-design/](http://sixrevisions.com/web_design/the-evolution-of-web-design/)
- Cray, B. (2012). PXtoEM.com: PX to EM conversion made simple. Retrieved from <http://pxtoem.com/>
- Creswell, J. W. (2009). *Qualitative, Quantitative, and Mixed Methods Approaches*. Washington DC: SAGE publication, Inc.
- Creswell, J. W. (2009). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Washington DC: SAGE Publications.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13, 319 – 340.
- Doyle, M. (2011). Responsive Web Design Demystified. Retrieved from [http://www.elated.com/articles/responsive-web-design-demystified./](http://www.elated.com/articles/responsive-web-design-demystified/)
- Fisher, S., Perkins, A., Walker, A., & Wolfart, E. (2003). Pixel Values. Retrieved from <http://homepages.inf.ed.ac.uk/rbf/HIPR2/value.htm>
- Flowchart. (2014). In *Wikipedia, the free encyclopedia*. Retrieved from <http://en.wikipedia.org/w/index.php?title=Flowchart&oldid=607565258>

- Fote, W. (2013). What is Ajax and Where is it Used in Technology? Retrieved from <http://www.seguetech.com/blog/2013/03/12/what-is-ajax-and-where-is-it-used-in-technology>
- Frost, B. (2013). 7 Habits of Highly Effective Media Queries | Brad Frost Web. Retrieved from <http://bradfrostweb.com/blog/post/7-habits-of-highly-effective-media-queries/>
- Graeve, K. D. (2012). Responsive Web Design. Retrieved from <http://www.sitepoint.com/responsive-web-design/>
- Harb, E., Kapellari, P., Luong, S., & Spot, N. (2011). Litration Study: Responsive Web Design. Retrieved from <http://courses.iicm.tugraz.at/iaweb/surveys/ws2011/g3-survey-resp-web-design.pdf>
- Hurb, E., Kapellari, P., Luong, S., & Spot, N. (2011, December). Survey Responsive Web Design.pdf. Retrieved from [courses.iicm.tugraz.at/iaweb/.../g3-survey...](http://courses.iicm.tugraz.at/iaweb/.../g3-survey...)  
Graz University of Technology
- IEEE Std. 610.12. (1990). Product Oriented Standards. Nivada: IEEE
- ISO 9241-11. (1998). Product Oriented Standards. Nivada: IEEE
- ISO/IEC 9126-1. (2000). Product Oriented Standards. Nivada: IEEE
- 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>
- Eck, D.J. (2011). *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>

- Johal, P. (2012). Converting Your Website to Responsive Design - Part I. Retrieved from <http://www.openroad.ca/2012/08/20/converting-your-website-to-use-responsive-design-part-i/>
- John, R. (2006). jQuery. Retrieved from <http://jquery.com/>
- Query, J. (2015). Publishing Your Plugin | jQuery Plugin Registry. Retrieved from <https://plugins.jquery.com/docs/publish/>
- Keio, E. M. (2013). HTML and CSS - W3C. Retrieved from <http://www.w3.org/standards/webdesign/htmlcss>
- Kenny, W. (2012). BlocksIt.js - Dynamic Grid Layout jQuery Plugin. Retrieved from <http://www.inwebson.com/jquery/blocksit-js-dynamic-grid-layout-jquery-plugin/>
- Knight, K. (2009). 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/>
- Knight, K. (2011). Responsive Web Design: What It Is and How To Use It. Retrieved from <http://www.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/>
- Labaree, R. (2014). LibGuides. Organizing Your Social Sciences Research Paper. Types of Research Designs. Retrieved from <http://libguides.usc.edu/content.php?pid=83009&sid=818072>
- Lee, J. (1997). Design Rationale Systems: Understanding the Issues. *IEEE Expert*, 12(3), 79–85.
- Lund. (2012). Purposive sampling | Lærd Dissertation. Retrieved from <http://dissertation.laerd.com/purposive-sampling.php>

- Lund, A. M. (2001). Measuring Usability with USE Questionnaire. *STC Usability SIG Newsletter*, 8(2).
- Marcotte, E. (2010). Responsive Web Design. *A LIST APART*. Retrieved from <http://alistapart.com/article/responsive-web-design>
- Narayan, S. (2011). What is jQuery and How to Start using jQuery? - CodeProject. Retrieved from <http://www.codeproject.com/Articles/157446/What-is-jQuery-and-How-to-Start-using-jQuery>
- Nielsen, J. (1994). Usability Engineering. Holland: *Academic Press*.
- Nimesh, R. (2012). How Fluid Grids Work in Responsive Web Design. Retrieved, from <http://www.1stwebdesigner.com/tutorials/fluid-grids-in-responsive-design/>
- Oberkampff, W., Trucano, T., & Hirsch, C. (2002). Verification, Validation, and Prediction Capability. In *Verification and Validation for Modelling and Simulation in Computational Science and Engineering Application*. Maryland:Laurel,
- Olsen, D. (2013). RESS, Server-Side Feature-Detection and the Evolution of Responsive Web Design - dmolsen.com. Retrieved, from <http://dmolsen.com/2012/02/21/ress-and-the-evolution-of-responsive-web-design/>
- Pamental, J. (2014). Webcast: Responsive Typography: Design for Meaning, Not for Screen Size. Retrieved from <http://www.oreilly.com/pub/e/3144>
- Pettit, N. (2012). Beginner's Guide to Responsive Web Design | Treehouse BlogTreehouse Blog. Retrieved from <http://blog.teamtreehouse.com/beginners-guide-to-responsive-web-design>

- PU. (2013). List of Master Theses | Research Portal | Philadelphia University. Retrieved from <http://www.philadelphia.edu.jo/research-portal/theses/theses/>
- Rouse, M. (2007). What is Ajax (Asynchronous JavaScript and XML)? - Definition from WhatIs.com. Retrieved from <http://searchwindevelopment.techtarget.com/definition/Ajax>
- Smarty, A. (2013). A Look Into Responsive Typography. Retrieved from <http://www.internetmarketingninjas.com/blog/design/a-look-into-responsive-typography/>
- Solanki, K. (2012). Responsive Web Design - Ideas, Technology, and Examples. Retrieved from <http://www.onextrapixel.com/2012/05/17/responsive-web-design-ideas-technology-and-examples/>
- Soueidan, S. (2013). Techniques for Responsive Typography. Retrieved September 18, 2014, from <http://tympanus.net/codrops/2013/11/19/techniques-for-responsive-typography/>
- Staff, L. (2013). Basics of Responsive Design: Using Fluid Grids | Lunarpages Web Hosting. Retrieved September 18, 2014, from <http://www.lunarpages.com/uptime/basics-of-responsive-design-using-fluid-grids>
- Storey, D. (2013). CSS Fluid Image Techniques for Responsive Site Design. Retrieved September 18, 2014, from <http://demosthenes.info/blog/586/CSS-Fluid-Image-Techniques-for-Responsive-Site-Design>
- Tranfici, A. (2013). Responsive Web Design: Fluid Layouts. Retrieved from <http://www.sitepoint.com/responsive-web-design-fluid-layouts/>



- Wavik, G. (2009). Table Layouts vs. Div Layouts: From Hell to... Hell? Retrieved from <http://www.smashingmagazine.com/2009/04/08/from-table-hell-to-div-hell/>
- Wroblewski, L. (2011). LukeW | RESS: Responsive Design + Server Side Components. Retrieved from <http://www.lukew.com/ff/entry.asp?1392>
- Wroblewski, L. (2012). LukeW | Which One: Responsive Design, Device Experiences, or RESS? Retrieved from <http://www.lukew.com/ff/entry.asp?1509>
- Xenos, M. (2001). Usability Perspective in Software Quality. In *Usability Engineering Workshop*. Southern Cyprus: Printice Hall
- Y. Na, D. (2012). The What, Why, and How of Mobile Applications. Hydrabad: *Noblis Publication*.

## APPENDICES

### Appendix 1: Cost and Material Estimates

S/NO.	ITEM	SPECIFICATION	QNTY	EACH Kshs.	TOTAL Kshs.
1	Photo copies and spiral binding	For peer review, submission to BPS, and Conferences	18	300.00	9,000.00
2	Internet Bundles for c-resources	Working at home	8 months	1,000.00	8,000.00
3	Survey Paper Publishing	Journal Publishing fees	3 paper	8,500.00	25,500.00
4	Conference Expenses	Travel food and boarding	2 days	3,000.00	6,000.00
5	Final Thesis printing and Hard cover binding	Black Cover with Gold engraved wordings	6 copies	1000.00	6,000.00
<b>TOTAL</b>					<b>54,500.00</b>

### Appendix 2: Gantt Chart

ACTIVITY	YEAR 2014/2015											
	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Proposal writing and submission to EPS	■											
Algorithm development		■										
Algorithm Analysis and Testing			■									
Thesis Writing				■	■	■						
Conference presentation							■					
Survey paper preparation and publishing							■	■				
Submission of Final Thesis to EPS									■			
Thesis defense										■		
Corrections and final submission											■	

### Appendix 3: Questionnaire

The study, is conducting a survey to test and validate a responsive web design framework Liquidizer.js in the following software industries; Crablins Interactive, Jafftek Computer Solution and Mombasa Tech Community. The purpose of this survey is to assess the usability of the framework by web designers and developers. Please take a few minutes to complete this questionnaire after reviewing the given framework. The information generated using this questionnaire will be treated confidentially and only be used for the intended purpose of thesis writing and award of Master of Science in Computer Systems.

<b>A. GENERAL SECTION</b> (Tick Appropriately response)
1. What is your position in the organization software team? Web Designer <input type="checkbox"/> Software Developer <input type="checkbox"/> Others <input type="checkbox"/>
2. How many Years have you been in software Industry?
a) Below 1 year
b) 1- 4 years
c) 5 and Above
3. How many websites have you developed / designed or participated in developing?
a) None
b) 1-5
c) 5 and Above
4. Do you use frameworks for development / designed of website?
Yes <input type="checkbox"/> No <input type="checkbox"/>
5. If your answer in Q4 is Yes what is it possible reasons
a) None
b) Hard to Integrate
c) Time Consuming

## A. USABILITYSECTION

Use the scale below to answer the questions by ticking in the correct box appropriately.

**1 = “Strongly Disagree”, 2 = “Disagree”, 3 = “Slightly Disagree”, 4 = “Neutral”, 5 = “Slightly Agree”, 6 = “Agree” and 7 = “Strongly Agree”**

<b>1. Usefulness</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
1. The liquidizer js helps me more effectively							
2. The liquidizer js helps me more productively							
3. The liquidizer js is useful							
4. The liquidizer js gives me more control over the activities in my life							
5. The liquidizer js makes the things I want to accomplish easier to get done							
6. The liquidizer js save me time when I use it							
7. The liquidizer js meets my needs							
8. The liquidizer js does everything I would expect it to do							

2. Ease of Use							
	1	2	3	4	5	6	7
9. The liquidizer.js is easy to use							
10. The liquidizer.js is simple to use							
11. The liquidizer.js is user friendly							
12. The liquidizer.js requires the fewest steps possible to accomplish what I want to do with it							
13. The liquidizer.js is flexible							
14. The liquidizer.js is effortless							
15. I can use the liquidizer.js without written instructions							
16. I do not notice any inconsistencies as I use the liquidizer.js							
17. Both occasional and regular users would like the liquidizer.js							
18. I can recover from mistakes quickly and easily from liquidizer.js							
19. I can use the liquidizer.js successfully every time							

3. Ease of Learning							
	1	2	3	4	5	6	7
20. I learned to use the liquidizer.js quickly							
21. I easily remember how to use the liquidizer.js							
22. The liquidizer.js is easy to learn to use it							
23. I quickly become skillful with the liquidizer.js							

4. Satisfaction	1	2	3	4	5	6	7
24. I am satisfied with the liquidizer.js							
25. I would recommend the liquidizer.js to a friend							
26. The liquidizer.js is fun to use							
27. The liquidizer.js works the way I want it to work							
28. The liquidizer.js is wonderful							
29. I feel I need to have the liquidizer.js							
30. The liquidizer.js is pleasant to use							

**Thank you for taking the time to complete this questionnaire.**

**Appendix 4: Introductory Letter for Collect Data**

Research questionnaire



**JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY**  
**INSTITUTE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**  
**P.O BOX 62000 NAIROBI 00200**

---

Abdulrehman A. Mohamed

P.O Box 701 – 80100, Mombasa, Kenya

Dear Sir / Madam,

Re: Request To Participate in Master of Science Research Questionnaire

I am a Master of Science in Computer System student in the department of computing at the institute of computer science and information technology. I am currently writing a thesis entitled “An Enhanced Approach to Responsive Web Design in Fluid Grid Concept.”

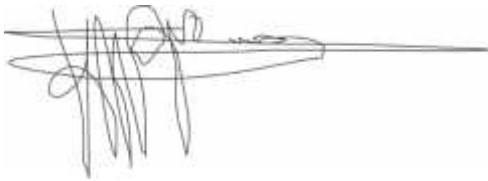
Extending beyond the boundaries of science, art, and culture, Responsive Web Design (RWD) provides new paradigms and techniques to develop one single website which looks different for different screen sizes from smart phones to main frame computers, so that it is usable on every devices. In the recent development, web standard designers are



building websites which are getting closer and closer to the ideal of one web accessible to everyone and everywhere.

I will highly appreciate your assistances by reviewing the responsive web design Liquidizer.js framework and asserting your response by filling in the questionnaire with regard to your experience in the software industry. The data generated by this questionnaire will be treated confidential and only be used for the intended purpose of thesis writing and the award of Master of Science in Computer Systems. Thank you in advance for your cooperation.

Yours Faithfully,

A handwritten signature in black ink, appearing to read 'Abdulrehman Ahmed Mohamed', written over a horizontal line.

**Abdulrehman Ahmed Mohamed**

**Reg.No.:** CS382-C005-3654/2013

**Mobile No.:** +254 713 500 814

**E-mail:** almutwafy@gmail.com

## Appendix 5:Publications

### 1. Literature Survey Responsive Web Design

Particulars	Description
Publisher	The International Journal of Engineering and Science (IJES)
Paper Id	37036
Paper Title	Responsive Web Design in Fluid Grid Concept Literature Survey
Author Name	Abdulrehman A. Mohamed    Dr. Cheruiyot W.K    Dr. Richard Rimiru    Collins Ondago
Page Number	49-57
Paper Index	09.1913/0373049057
Website	<a href="http://www.theijes.com/papers/v3-i7/Version-3/G0373049057.pdf">http://www.theijes.com/papers/v3-i7/Version-3/G0373049057.pdf</a>

### 2. Liquidizer.js A Responsive Web Design Algorithm

Particulars	Description
Publisher	The International Journal of Engineering and Science (IJES)
Paper Id	4559
Paper Title	Liquidizer.js: A Responsive Web Design Algorithm
Author Name	Abdulrehman Mohamed   Dr. Cheruiyot W.K   PhD   Dr. Richard Rimiru PhD
Page Number	78-91
Paper Index	09.1913/0453078091
Website	<a href="http://www.theijes.com/papers/v4-i5/Version-3/L0453078091.pdf">http://www.theijes.com/papers/v4-i5/Version-3/L0453078091.pdf</a>

### 3. Liquidizer.js Algorithm

Particulars	Description
Publisher	GitHub Repository
Algorithm Name	Liquidizer.js: Algorithm
Author Name	Abdulrehman A. Mohamed
Website	<a href="https://github.com/almutwafy/Liquidizer">https://github.com/almutwafy/Liquidizer</a>

### 4. An Enhanced Approach to Responsive Web Design in Fluid Grid Concept

Particulars	Description
Publisher	The International Journal of Science and Engineering Research (IJSER)
Paper Id	I066792
Paper Title	An Enhanced Approach to Responsive Web Design in Fluid Grid Concept
Author Name	Abdulrehman A. Mohamed    Dr. Cheruiyot W.K    Dr. Richard Rimiru
Page Number	1214-1224
Paper Index	Volume 6 Issue 6 June-2015
ISSN	2229-5518
Website	<a href="http://www.ijser.org/onlineResearchPaperViewer.aspx?An-Enhanced-Approach-to-Responsive-Web-Design-in-Fluid-Grid-Concept.pdf">http://www.ijser.org/onlineResearchPaperViewer.aspx?An-Enhanced-Approach-to-Responsive-Web-Design-in-Fluid-Grid-Concept.pdf</a>