ENHANCING ONLINE LEARNING EXPERIENCE THROUGH A RIA BASED LEARNING APPLICATION

MEISAM MORADI

Edited By:

Sabzali Musa Kahn
University of Malaya

Jaffri Hanafi
University of Malaya

2012
CONTRIBUTORS

**********

**Meisam Moradi**
Postgraduate Student. Faculty of Creative Multimedia, Multimedia University

**Sabzali Musa Kahn**
Senior Lecturer. Visual Art Department, Cultural Centre. University of Malaya

**Jaffri Hanafi**
Lecturer. Visual Art Department, Cultural Centre. University of Malaya

**Zahra Kakaei**
Research Assistant, Faculty of Engineerings, University of Malaya
PREFACE

********

Faculty of Creative Multimedia, Multimedia University Book Series on Research in Creative Multimedia is a collection of peer reviewed papers by local and international scholars. This book aims to disseminate current research to both academicians and tertiary visual arts student. In this series, our contributors are Meisam Moradi, Sabzali Musa Kahn, Jaffri Hanafi, and Zahra Kakaei who are local and international researchers. This book draw upon insightful observation and analysis into Enhancing Online Learning Experience Through a RIA Based Learning Application Study.
TABLE OF CONTENTS

CONTRIBUTORS .................................................................................................................. II

PREFACE .................................................................................................................................. III

LIST OF TABLES ....................................................................................................................... VII

LIST OF FIGURES .................................................................................................................... VIII

CHAPTER 1: INTRODUCTION ON ENHANCING ONLINE LEARNING EXPERIENCE THROUGH A RIA BASED LEARNING APPLICATION.

1-1) Rich Internet Applications (RIA) ................................................................. 1
1-2) HTML5 as a Rich Internet Programming Language ..................... 8
1-3) Html5 New Features .................................................................................... 9
1-4) Benefits of HTML5 ........................................................................................ 10
1-5) HTML5 Elements ............................................................................................ 10
1-6) The Project’s Framework .......................................................................... 12

CHAPTER 2: HISTORY OF DISTANCE LEARNING AND REASONS FOR DEVELOPING RIA BASED ONLINE LEARNING APPLICATIONS

2-1) History of Distance Learning ................................................................. 14

2-2) Reasons for developing RIA Based Online Learning Applications .................................................................................................................. 16

2-3) Types of Online Learning Applications .............................................. 16

2-4) Problems and Issues ................................................................................. 17

CHAPTER 3: RIA CODES AND METHODOLOGY ON ENHANCING ONLINE LEARNING EXPERIENCE THROUGH A RIA BASED LEARNING APPLICATION

3-1) Prototype of the Application ................................................................. 19

3-2) MySQL Databases ..................................................................................... 32

3-3) Codes ........................................................................................................ 33
3-3-1) Index.html Code Explanation.................................34
3-3-2) Manager.html Code Explanation..............................37
3-3-3) Manager.js Code Explanation.................................39
3-3-4) Manager.server.php Code Explanation......................49
3-3-5) Manager.style.css Code Explanation.......................52
3-3-6) Config.php Code Explanation.................................53
3-3-7) Saveimage.php Code Explanation.............................53
3-3-8) Server.php Code Explanation.................................54
3-3-9) Sketch.js Code Explanation.................................58
3-3-10) Style.css Code Explanation.................................79

CHAPTER 4: ANALYSIS AND RESULTS ON ENHANCING ONLINE
LEARNING EXPERIENCE THROUGH A RIA BASED LEARNING
APPLICATION.

4-1) Codes Portability Tests .............................................81
4-2) Browsers Compatibility Tests .....................................81
4-3) Speed Tests..............................................................82

4-3-1) Statistics for Single Page Load.................................84
   4-3-1-1) Global.............................................................84
   4-3-1-2) Object Size in Total .........................................84
   4-3-1-3) External Objects..............................................85
   4-3-1-4) Download Times.............................................85
   4-3-1-5) Page Objects................................................86
   4-3-1-6) Request/Response Headers:..............................97

4-3-2) Double Load Test ...............................................91

4-4) Comparison .............................................................91
   4-4-1) Windows Tests ................................................91
   4-4-2) Mac Tests.......................................................92
CHAPTER 5: CONCLUSION ON ENHANCING ONLINE LEARNING EXPERIENCE THROUGH A RIA BASED LEARNING APPLICATION

5-1) Conclusion On Enhancing Online Learning Experience Through A Ria Based Learning Application .............. 96

References ............................................................................................................................. 99
LIST OF TABLES

TABLE 1: LIST OF AVAILABLE RIA FRAMEWORKS
TABLE 2: LIST OF RICH INTERNET APPLICATION FORMATS
TABLE 3: RIA GLOBAL USAGE COMPARISON AND AREA
TABLE 4: RIA GLOBAL USAGE COMPARISON, RESULTS
TABLE 5: HTML5 NEW TAGS LIST
TABLE 6: BOARDS TABLE STORES BOARD DATA
TABLE 7: SKETCHES TABLE STORES DRAWING DATA
TABLE 8: USERS TABLE STORES USER INFORMATION
TABLE 9: THE PROJECT RELEASES SPEED TESTS
TABLE 10: TOTAL HTTP REQUESTS SIZE
TABLE 11: HTTP RESPONSE TIME BASED ON OBJECT TYPES
TABLE 12: QUANTITY OF EXTERNAL OBJECTS
TABLE 13: DOWNLOAD TIME FOR DIFFERENT CONNECTIONS
TABLE 14: LIST OF THE OBJECTS WITH THEIR SIZE AND HEADER SIZE
TABLE 15: REQUEST/RESPONSE HEADERS FOR FIRST 25 REQUESTS
TABLE 16: THE RESULTS OF THE PROJECT DOUBLE VIEW LOAD TEST.
LIST OF FIGURES

FIGURE 1: THE CORE TECHNOLOGY ARCHITECTURE OF RIA
FIGURE 2: PROJECT FRAMEWORK
FIGURE 3: MAIN PAGE LAYOUT
FIGURE 4: BOARD MAIN PAGE IN MAIN THEME
FIGURE 5: BOARD MAIN PAGE IN BLACK THEME
FIGURE 6: MAIN PROJECT PAGE IN MODERN/BLUE THEME
FIGURE 7: BOARD MANAGEMENT PAGE
FIGURE 8: INDEX.HTML DIV TAGS TEMPLATE
FIGURE 9: OVERALL HTML5 SUPPORT DIAGRAM FOR SIX BROWSERS.
FIGURE 10: OBJECT LOAD TIME AND THE CPU AND B.W. DIAGRAM
FIGURE 11: OBJECT LOADING ORDERS AND CONNECTIONS
FIGURE 12: SIZE (IN BYTES) OF PROJECT OBJECTS
FIGURE 13: MAIN PAGE, MAIN THEME
FIGURE 14: BLACK THEME
FIGURE 15: MODERN THEME
FIGURE 16: SCREENSHOT 4: BOARD MANAGEMENT PAGE
FIGURE 17: COORDINATES BAR
FIGURE 18: CHATTING PANEL
FIGURE 19: THEME SELECT
FIGURE 20: BOARD SELECT
CHAPTER 1

INTRODUCTION ON ENHANCING ONLINE LEARNING EXPERIENCE THROUGH A RIA BASED LEARNING APPLICATION

Meisam Moradi
Sabzali Musa Kahn
Jaffri Hanafi
Zahra Kakaei

1-1) Rich Internet Applications (RIA)

Recently, the difficulty of tasks operated through Web applications are extending, especially when great levels of interaction, client-side processing, and multimedia capacities have to be provided. Therefore Rich Internet applications add new abilities to hypertext pages to improve client-side processing to achieve those more interaction and multimedia capacities. Because of limits in traditional HTTP-HTML web applications, the developers practice the rich internet applications to construct the future of the web. These rich internet applications (RIA) are applications with many additional features. (Sanchez F., Preciadol J. C., Lineal M. Comair S., 2010)

The concept of develop an Interaction inside browser environments has to be recognized as a key factor that takes place in web browser from learner. As Guy (2001) stated in his paper, the interaction may be a drag and drop, or moving mouse while it is clicked (painting in page) or other in/out-put events that possible in traditional personal computer software. In order to add these abilities
to a web based page, a company started first rich internet application in 1996 and name it Macromedia Flash. Many years later, in 2005, Flash technology started to distribute by Adobe Systems. This company is also developing the flash technology. (Guy, 2001)

Randy (2010) in his research article announced that the start point of adobe flash is a pioneer of rich internet applications. Adobe flash is first ever known RIA of its kind. Randy (2010) mentioned, Smart-Sketch was the first RIA which some 15 years ago was derived from a vector-oriented sketching program for a pen computer. Later Adobe Flash was born based on this vector based idea and become the roots of a platform for a new style of web applications known as RIAs. The next few years Flash evolve to embrace audio, video, and highly elaborate pixel and text handling capabilities under the aegis of Macromedia and, recently, Adobe. (Randy, 2010)

The below figure (figure 1) is a display diagram to show the core technology architecture of most RIA:
Figure 1: The core technology architecture of RIA
1-1-2) Types of RIA Frameworks

Below table (table 1) is a list of available rich Internet application frameworks and their built formats:

<table>
<thead>
<tr>
<th>Framework</th>
<th>Win</th>
<th>Linux</th>
<th>Mac OS X</th>
<th>Software license</th>
<th>Built formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appcelerator Titanium</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Apache 2</td>
<td>iPhone, iPad, Android</td>
</tr>
<tr>
<td>Curl</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Proprietary</td>
<td>Curl</td>
</tr>
<tr>
<td>Echo3</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>MPL/GPL/LGPL</td>
<td>JavaScript client, Java server</td>
</tr>
<tr>
<td>Cappuccino</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>LGPL</td>
<td>JavaScript, .sj</td>
</tr>
<tr>
<td>Adobe Flex</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>MPL</td>
<td>Adobe AIR, SWF</td>
</tr>
<tr>
<td>Google Web Toolkit</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Apache 2</td>
<td>JavaScript</td>
</tr>
<tr>
<td>iPFaces mobile</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Open Source</td>
<td>iPhone, BlackBerry, Java ME</td>
</tr>
<tr>
<td>JavaFX</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Open Source</td>
<td>jar, applet</td>
</tr>
<tr>
<td>JavascriptMV C</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Open Source</td>
<td>JavaScript, compressed and one single file</td>
</tr>
<tr>
<td>Tool</td>
<td>OS Supported</td>
<td>Platform Support</td>
<td>License</td>
<td>Technology Stack</td>
<td>Mobiles Supported</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>------------------------</td>
<td>---------</td>
<td>------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>JVx</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Apache 2</td>
<td>Java, Ext GWT, Qt</td>
</tr>
<tr>
<td>MotherApp</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Proprietary</td>
<td>iPhone, Android, BlackBerry, Windows Mobile, Symbian</td>
</tr>
<tr>
<td>Lively Kernel</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>MIT</td>
<td>JavaScript</td>
</tr>
<tr>
<td>PhoneGap</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Open Source</td>
<td>iPhone, Android, BlackBerry</td>
</tr>
<tr>
<td>OpenLaszlo</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Open Source</td>
<td>SWF, DHTML</td>
</tr>
<tr>
<td>qooxdoo</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>LGPL / EPL</td>
<td>JavaScript</td>
</tr>
<tr>
<td>.Net Framework</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Proprietary</td>
<td>Microsoft Silverlight</td>
</tr>
<tr>
<td>Panda3D</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Open Source</td>
<td>?</td>
</tr>
<tr>
<td>Qt Quick</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>MIT</td>
<td>JavaScript</td>
</tr>
<tr>
<td>Quick PHP</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>LGPL &amp; Commercial</td>
<td>JavaScript client, Java server</td>
</tr>
<tr>
<td>Vaadin</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Apache 2</td>
<td>JavaScript</td>
</tr>
<tr>
<td>Tersus</td>
<td>Yes</td>
<td>Yes</td>
<td>Open Source</td>
<td>Open Source</td>
<td>JavaScript, iPhone,</td>
</tr>
</tbody>
</table>
Table 1: List of available rich internet application frameworks

Below table (table 2) is generated based on operating system compatibility and development environment that is projecting a list of rich Internet applications with their software licenses.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AIR Adobe</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Adobe AIR EULA</td>
<td>Mostly any ActionScr ipt IDE</td>
</tr>
<tr>
<td>Curl</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Proprietary</td>
<td>Curl IDE</td>
<td>any text editor</td>
</tr>
<tr>
<td>Javascript</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Partial</td>
<td></td>
<td></td>
<td>any text editor</td>
</tr>
<tr>
<td>Mozilla XUL Runner</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Mozilla Tri-license</td>
<td>any text editor</td>
<td></td>
</tr>
</tbody>
</table>
1-1-2) Rich Internet Application Usage

The major RIA based systems are Adobe Flash, Java, Microsoft Silverlight, and HTML5. The below table (table 13) is comparing Adobe Flash, Microsoft Silverlight, and Java global usage in 2010 and 2011. The comparison was made before the time that HTML5 was announced.

Below table (table 3) is about to demonstrate RIA global usage area

<table>
<thead>
<tr>
<th>Year</th>
<th>Browser</th>
<th>Operating System</th>
<th>Connection Speed</th>
</tr>
</thead>
</table>
| 2010  | Internet Explorer 8  
Firefox 3.6  
Chrome 4 – 8  
Opera 10.50  
Safari 5  
Opera 11 | Any | Any |
Table 3: RIA global usage comparison and area

Below table (table 4) is the result of tested data which calculated by statowl.com.

<table>
<thead>
<tr>
<th>Application</th>
<th>Total</th>
<th>Dec '10</th>
<th>Jan '11</th>
<th>Feb '11</th>
<th>Mar '11</th>
<th>Apr '11</th>
<th>May '11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash Support</td>
<td>95.75%</td>
<td>96.30%</td>
<td>96.18%</td>
<td>95.86%</td>
<td>95.28%</td>
<td>95.57%</td>
<td>95.26%</td>
</tr>
<tr>
<td>Java Support</td>
<td>77.88%</td>
<td>78.52%</td>
<td>78.54%</td>
<td>78.26%</td>
<td>78.80%</td>
<td>77.51%</td>
<td>76.51%</td>
</tr>
<tr>
<td>SilverLight Support</td>
<td>62.31%</td>
<td>59.80%</td>
<td>61.19%</td>
<td>62.07%</td>
<td>62.82%</td>
<td>64.00%</td>
<td>64.16%</td>
</tr>
</tbody>
</table>

Table 4: RIA global usage comparison, results

1-2) HTML5 as a Rich Internet Programming Language

Between the different of RIA frameworks examined, the Web modeling language HTML and CSS recently expanded to achieve more RIAs design aims. In addition, the HTML5 planned to fit with further uncovered RIAs capabilities. Latest HTML release is HTML5, which has added a high level of interactivity and embedded media to World Wide Web. The high level of interactivity and enabled media is very appropriate to build an online learning system. These
features make it a technology to compete with Adobe Flash and other RIA frameworks in this field.

After many researches and comparisons between HTML5 and Flash, Lastly, HTML5 has been chosen as a client-side programing language and at the same time for server-side demands PHP and MySQL is selected to cover the server needs of this project.

1-3) Html5 New Features

By World Wide Web Consortium (W3C, 2011) explanation, HTML5 is a web application. The differences between HTML5 as a web application and HTML as a markup language are described below. W3C mentioned that HTML5 defines the fifth major revision of the core language of the World Wide Web, HTML. It is also called Web Applications 1.0 and after about 20 years, html added new elements to its new revision. This revision introduces some new elements to the web since 1990. (World Wide Web Consortium [W3C], 2011)

These new elements which introduced by World Wide Web Consortium (W3C, 2011) are categorized into four groups:

1) Structural (aside, figure, and section).
2) Inline (time, meter, and progress).
3) Embedding (video and audio).
4) Interactive (details, datagrid, and command).

Other new elements are (sorted alphabetically):
article, aside , audio , canvas, command, datalist, details, embed, fieldset, figcaption, figure, footer, header, hgroup, keygen, mark, meter, nav, output, progress, rt, ruby, section, source, summary, time, video, wbr

Hence, new features are provided with HTML5, there are a couple of elements that are deprecated. The deprecated elements are:
acronym, applet, basefont, big, center, dir, font, frame, frameset, noframes, strike, tt u.

1-4) Benefits of HTML5

Using HTML5 for online learning system has benefits instead of other languages. The HTML5 solely is nothing special, but alongside with CSS3 plus JavaScript and its libraries is a very good basement to create the powerful RIAs (W3C, 2011). In comparison with Adobe Flash, possibility of putting everything straight on the page and adding a script to them, make an interactions a very neat approach (Meyer, 2010). The one of the differences between flash and HTML5 is that Flash is having its own environment compared to HTML5-CSS-JavaScript which enables a way to put things exactly in page between normal texts. It is one of the advantages of HTML5 that HTML5 does not need installing additional plugins or add-ons on browsers before running the application.

HTML5 and Flash are totally different. They are both adding interactivity to web but in different way. (Meyer, 2010)

1-5) HTML5 Elements

Below table (table 5) is a full list of HTML5 new elements presented by World Wide Web Consortium (W3C, 2011). The list contains short description for each element. This project is solely based on using canvas element and its related methods. Later in chapter 3, in code descriptions, canvas element and its related methods are discussed in a more detailed manner.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;article&gt;</td>
<td>Defines an article</td>
</tr>
<tr>
<td>&lt;audio&gt;</td>
<td>Defines sound content</td>
</tr>
<tr>
<td>Tag</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>&lt;aside&gt;</code></td>
<td>Defines content aside from the page content</td>
</tr>
<tr>
<td><code>&lt;canvas&gt;</code></td>
<td><strong>Defines graphics</strong></td>
</tr>
<tr>
<td><code>&lt;command&gt;</code></td>
<td>Defines a command button</td>
</tr>
<tr>
<td><code>&lt;details&gt;</code></td>
<td>Defines details of an element</td>
</tr>
<tr>
<td><code>&lt;datalist&gt;</code></td>
<td>Defines a dropdown list</td>
</tr>
<tr>
<td><code>&lt;embed&gt;</code></td>
<td>Defines external interactive content or plugin</td>
</tr>
<tr>
<td><code>&lt;figcaption&gt;</code></td>
<td>Defines the caption of a figure element</td>
</tr>
<tr>
<td><code>&lt;fieldset&gt;</code></td>
<td>Defines a fieldset</td>
</tr>
<tr>
<td><code>&lt;footer&gt;</code></td>
<td>Defines a footer for a section or page</td>
</tr>
<tr>
<td><code>&lt;figure&gt;</code></td>
<td>Defines a group of media content, and their caption</td>
</tr>
<tr>
<td><code>&lt;hgroup&gt;</code></td>
<td>Defines information about a section in a document</td>
</tr>
<tr>
<td><code>&lt;header&gt;</code></td>
<td>Defines a header for a section or page</td>
</tr>
<tr>
<td><code>&lt;meter&gt;</code></td>
<td>Defines measurement within a predefined range</td>
</tr>
<tr>
<td><code>&lt;keygen&gt;</code></td>
<td>Defines a generated key in a form</td>
</tr>
<tr>
<td><code>&lt;mark&gt;</code></td>
<td>Defines marked text</td>
</tr>
<tr>
<td><code>&lt;progress&gt;</code></td>
<td>Defines progress of a task of any kind</td>
</tr>
<tr>
<td><code>&lt;nav&gt;</code></td>
<td>Defines navigation links</td>
</tr>
<tr>
<td><code>&lt;output&gt;</code></td>
<td>Defines some types of output</td>
</tr>
<tr>
<td><code>&lt;rt&gt;</code></td>
<td>Defines explanation to ruby annotations</td>
</tr>
<tr>
<td><code>&lt;section&gt;</code></td>
<td>Defines a section</td>
</tr>
<tr>
<td><code>&lt;ruby&gt;</code></td>
<td>Defines ruby annotations</td>
</tr>
<tr>
<td>&lt;summary&gt;</td>
<td>Defines the header of a detail element</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>&lt;source&gt;</td>
<td>Defines media resources</td>
</tr>
<tr>
<td>&lt;time&gt;</td>
<td>Defines a date/time</td>
</tr>
<tr>
<td>&lt;video&gt;</td>
<td>Defines a video</td>
</tr>
<tr>
<td>&lt;wbr&gt;</td>
<td>Defines a possible line-break</td>
</tr>
</tbody>
</table>

Table 5: HTML5 new tags list

1-6) The Project’s Framework

This project has two set of codes:
1) client-side
2) server-side

Client side files are image files and code files that are written using HTML5, CSS, and JavaScript. Same as any web application, HTML5 in conjunction with CSS provides the end-user front-end and JavaScript handles events and interactions in the page.

PHP codes and MySQL database make server-side connections. The PHP manages the JavaScript receive events and then answers them by using the retrieved data from the MySQL.

It is not necessary to put server files and client files in same location. For example after setup, the server location will be www.test.com/server.php and the client files could be placed anywhere on any domain name. The only configuration afterwards is to configure the client files to find www.test.com/server.php.

As a brief description to the project framework, Interface interactions are captured by window.addEventListener, then to contact with server, JavaScript uses XMLHttpRequest object. XMLHttpRequest calls a PHP server with some HTTP_GET values set, then PHP file starts to connect to database by
mysql_query and mysql_fetch_object, then returns the requested values to the JavaScript. This contact loop accrues when user interacting with board and repeatedly during the application run time.

The below figure (figure 2), displays the framework of this project. In this figure (figure 2) only five clients are demonstrated but it should be mentioned that in actual experience the number of clients is unlimited and it only depends on server capabilities.

Figure 2: Project Framework
2-1) History of Distance Learning

In this section, researcher is trying to propose a brief history of distance learning in detail by using the referenced mediums and resources. As this chapter is not history chapter, the researcher is not trying to dive too much into the historical reviews and stories. Likewise the history of distance learning is wide spread in three major development periods by the very first of its kind beginning by the mid-1800s. (Hinkle, 2009)

The second span of distance learning made by the advancement of technology in 1900 as Michel (2010) stated in his research, by the use and introduction of audiovisual devices such as television, technology-based distance education would be mentioned as of a pioneer. (Michel, 2010)

By introduction of television in schools as an instructional medium, learners and practitioners found an important point in distance learning for themselves.

Through the time distance education has been flamed up by the introduction of television as of an instructional medium that would apparently
makes theorists and practitioners highly incentive about the effects of a newly introduced medium against the correspondence study. (Jeffries, 2010)

As time passes, the importance of TV is distinctly making a highlight in distance learning as Hinkle (2009) mentioned the power of television began receiving focus and since then distance learning has entered the virtual age.

From 1960 and by more significantly by 1972 the number of educational television stations grew with a fascinating speed to 233 TV station channels. (Douglass, 2005)

From 1970’s to 1980’s advancement of technology made a newly born child of television named as teleconferencing which by the use of teleconferencing system in 1980s teachers were able to talk, hear and see their students actions and reactions in real time, as a result the ability of live bi-directional communication have gone far ahead of any regional or global boundaries. (Naseri, 2011)

The television fast-forward technological achievement and the audio-visual technology success that used mostly in learning process generated a renewed interest for it. The pioneer of the teleconferencing use was Open University of Britain as Zigerell (1984) Stated in his book.

From mid 80’s to 90’s the technological advancements brought new tools and devices like personal computers and computer networking. Distance education become capable of computer conferencing based on the internet where teachers and students are able to transmit and communicate necessary educational materials to each other. (Naseri, 2011)

From then, the distance education entered a unique area of fast forward moving. Naseri (2011) stated that by the expansion and advancements of computer networks in the 1980s and 1990s the number of teachers and students engaged in distance learning grew accordingly.
By having this little background of distance learning history, there is a pedagogical shift in learning environment. The form of learning was moved from the traditional classroom to the teleconferencing classes based on radio and television technology and then changed from radios and televisions to computers and networks. At the present time learning environment is going forward through mobility and mobile devices.

2-2) Reasons for Developing RIA Based Online Learning Applications

Simple internet applications are using techniques like normal HTML from page by page and displaying pages only limited to text, images, playing one-way audio and video streaming.

In contrast with simple internet applications in the RIAs (Rich Internet Application) the pages does not need to be refreshed to get the new content rather it is updating the content within the parts of the pages asynchronously by use of programming techniques.

This way by having all relating techniques of RIA, it is very time efficient and time saving for both the learners and the educators to use RIA as a learning environment. From the server side point of view the efficiency and performance are increased which means, each single server might handle load of more groups of learners and educators in larger class size at the same time.

2-3) Types of Online Learning Applications

According to description of Ellis (2009) research, there are three types of online learning systems:

A. Direct Model
   The package will purchased, installed, and directed in user own pc.

B. Third-party Model
   The administrator buys the software and installs it on its servers, but still the
upgrades, update and maintenance are administrated by the vendor or provider or another third party.

C. Software as a Service Model

A company buys software; it is both hosted and administrated remotely by admins, content managers, teachers or instructors, and users access the system over the internet.

All learning systems require additional consulting, technical supporter, administrator, and manager. It should be mentioned the above three types of learning applications are only based on researcher’s own thoughts and views, rather than major be a possibility of other types of online learning applications. The application that is designed in this study is an HTML5 base application which is operates well in all of these tree types of online learning systems.

2-4) Problems and Issues

To develop an online learning application there are a few problems and issues. Since this project is considering be a part of an online learning application the battle to overcome all the issues is the same. To name some, the proprietary standards and digital rights mismanagement, probability, compatibility and performance of application are among the issues.

Aside from issues and problems caused by human and hardware difficulties to adjust by themselves, there are a couple of minor mismatches from software side to get the optimum online learning application. HTML5 is the first battle in software aside. It is powerful to manage the interface when using CSS codes and together with JavaScript to handling events on the page but HTML5 and JavaScript are known to be a client side language that means it is not able to connect to the databases. The reason to have this difficulty is that the database is located on the server and HTML page and JavaScript within the HTML page are on the client-side system including PC, laptop, or mobile devices. The client side
languages are isolated by the both side, client browser and server firewall therefore it is impossible for the client-side programming languages to connect directly to a database.

To overcome the above-mentioned issue, the researcher has come up with a solution among many solutions that are in action. To name, some are like saving data to external CSV or XML file. This project is using the solution which is calling a web service that is in charge of putting the database connection on the same web service. This could be accomplished by writing a server side code using a server side language like PHP or ASP that is in charge of making connections from client side to server side. This web service will be explained at chapter 3.
CHAPTER 3

***********

RIA CODES AND METHODOLOGY ON ENHANCING ONLINE LEARNING EXPERIENCE THROUGH A RIA BASED LEARNING APPLICATION

Meisam Moradi
Sabzali Musa Kahn
Jaffri Hanafi
Zahra Kakaei

3-1) Prototype of the Application

The following chapter is a set of interface explanation and codes description from a live version of this project’s online application.

3-1-1) Interface

The main page interface template is demonstrated in below figure (figure 3).
Figure 3: Main page layout
Above figure (figure 3) is the main view of the board, which at the top there are a set of tools that could be used by user to interact with other users by using the drawing area below the tools. At the bottom of tools, at right side there are coordinates and the page refresh countdown number just above the chat area that enable users to talk to each other in real-time. Below the chat area there is a function that could be used by users to select their prepared theme. The latest in the right column is a board selecting function that switch user to selected board.

Figure 4: Board main page in main theme
Above figure (figure 4) is a view of the second theme named Black. This change is made by using the select theme option in the right column below the chat block.
Figure 6: Main project page in modern/blue theme

Above figure (figure 6) is just another view of main interface in Blue theme. This change also is made by using the select theme option.
Above figure (Figure 8) is a view of board management system page which enable users to create a new board, which is not present in the list, by using tools at the top of page or delete the board that is no longer needed by clicking on delete link in last column of each board row. The section at the
bottom of the page displays the detail of the latest data, which is send or received through database.

3-1-2) Actions

In this section all of the tools which are used in the application interface are described and the related code to the tool is mentioned.

3-1-2-1) Actions in index.html

Tool size small: Changes active tool size to thin.

HTML Code:

```
<li>
<a onClick="selectTools('pencil',1,null);" href="#" title="thin pencil">  
<img name="pencil" width="48" height="48" src="theme/default/pencil.png" />
<span class="btntext">pencil</span>
</a>
</li>
```

JavaScript Function: selectTools

Tool size medium: Changes active tool size to medium

HTML Code:

```
<li>
<a onClick="selectTools('pen',4,null);" href="#" title="medium pen">  
<img name="pen" width="48" height="48" src="theme/default/pen.png" />
<span class="btntext">pen</span>
</a>
</li>
```

JavaScript function: selectTools inside sketch.js
Tool size big: Changes active tool size to large.

HTML Code:

```html
<li>
<a onClick="selectTools('brush',20,null);" href="#" title="big brush">
<img name="brush" width="48" height="48" src="theme/default/brush.png" />
<span class="btntext">brush</span>
</a>
</li>
```

JavaScript function: selectTools inside sketch.js

Tool Color: Changes tool Color to selected color picker color.

HTML Code:

```html
<li>
<input id="colorPicker" value="FF6600" class="color{pickerMode:'HVS', pickerFace:10, pickerFaceColor:'#2C2C2C', pickerBorder:4, pickerBorderColor:'#1A1A1A #1A1A1A #1A1A1A #1A1A1A', pickerInset:2, pickerInsetColor:'#4D4B4B'}" onChange="selectTools(null,null,'#' + this.color);" autocomplete="off" />
</li>
```

JavaScript function inside colorPicker.js

Eraser Tool: Changes active tool to Eraser tool

HTML Code:

```html
<li>
<a onClick="selectTools('eraser',20,null);" href="#" title="Eraser">​
</a>
</li>
```
Clear Board: By using of this icon, the user is able to clear board from the client side and enables to send clear commands to the server, which clears all instances boards at the same time.

HTML Code:

```html
<li>
  <a id="clearScene" href="#" title="Clear the board">
    <img name="clear" width="48" height="48" src="theme/default/clear.png" />
    <span class="btntext">clear</span>
  </a>
</li>
```

JavaScript function: clearScene inside sketch.js

Save: this command grabs all the current sketches drawn on board, with transparent background then saves them as a PNG format.

HTML Code:

```html
<li>
  <a onClick="save2jpg();" href="#" title="Save the board as transparent PNG">
    <img name="save" width="48" height="48" src="theme/default/save.png" />
    <span class="btntext">save</span>
  </a>
</li>
```
JavaScript function: `save2jpg` inside `sketch.js`

**Print:** Prints the current board.

**HTML Code:**

```html
<li>
<a onClick="print();" href="#" title="Print the Board">
<img name="print" width="48" height="48" src="theme/default/print.png" />
<span class="btntext">print</span></a>
</li>
```

**Board Manager:** opens board management in new page.

**HTML Code:**

```html
<li>
<a href="manager.html" target="_blank" title="Open Board Manager">
<img name="newpage" width="48" height="48" src="theme/default/newpage.png" />
<span class="btntext">new page</span></a>
</li>
```

**Refresh:** Refreshes the Current page by sending refresh command to browser.

**HTML Code:**

```html
<li>
<a href="index.html" target="_self" title="Refresh this page">
<img name="refresh" width="48" height="48" src="theme/default/refresh.png"
```
3-1-2-2) Actions in manager.html

- **Add New Board:**
  Adds a new board-to-board list and sets its status to fresh board. The newly created board will be available to drawing immediately afterwards.

- **Clean all Boards:**
  Cleans all sketches from all boards. It also cleans the related data to the cleared board available in database.

- **Delete All Boards:**
  Removes all boards from the database.

- **DELETE:**
  The delete link, removes the current board from the database.

3-1-3) File and Folder List

This section is an introduction to the file structure of the application. The libs folder is libraries location that is included in this application. The files of this project are under GNU license and to check the license users are able to open the related files with any text editor and find the website link commented out in the most top of each file.

Themes folder contains image files related to each theme.

```
PROJECT
| config.php
| index.html
| manager.html
```
| manager.js
| manager.server.php
| manager.style.css
| output.png
| saveimage.php
| server.php
| sketch.js
| style.css
|
| ---libs
| | jscolor.js
| |
| \---img
| | arrow.gif
| | cross.gif
| | hs.png
| | hv.png
| | no.jpg
|
\---themes
+---black
| a.gif
| bg.jpg
| bg2.jpg
| brush.png
| circling-ball.gif
| clear.png
| eraser.png
+---default
|   bg.jpg
|   brush.png
|   circling-ball.gif
|   clear.png
|   eraser.png
|   newpage.png
|   pen.png
|   pencil.png
|   print.png
|   refresh.png
|   save.png
|   style.css
|   waitCircle.gif

\---modern
|   bg.jpg
|   bg2.jpg
|   brush.png
3-2) MySQL Databases

In addition to the files, the project also has a MySQL database that is named as *sketch*. The *sketch* database is used to store three tables which are named: *boards*, *sketches*, and *user*. Each table has its own row that is listed below.

Blow table (Table 6) is showing Board table information:

<table>
<thead>
<tr>
<th><strong>Row name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>bid</td>
<td>This row stores the board ids</td>
</tr>
<tr>
<td>bname</td>
<td>This row stores board names</td>
</tr>
<tr>
<td>bstatus</td>
<td>This board stores board condition which possible values are “cleared”, “drawing” and “idle”</td>
</tr>
<tr>
<td>buids</td>
<td>This row stores id of users which they are currently</td>
</tr>
</tbody>
</table>
interacting with the current board id

**Table 6: Boards table stores board data**

Blow table (Table 7) is showing Sketches table information:

<table>
<thead>
<tr>
<th>Row name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sid</td>
<td>This row stores sketches ids. Sketch id is a number to save the order of drawings on the board.</td>
</tr>
<tr>
<td>bid</td>
<td>This row stores board-ids for the current sketches and enables the application to understand which sketch is drawn in which board.</td>
</tr>
<tr>
<td>points</td>
<td>This row stores the coordinates of the current sketch id which usually are x, y, weight, color and type. This row also stores chat texts instead of points when received data type is chat.</td>
</tr>
</tbody>
</table>

**Table 7: Sketches table stores drawing data**

Blow table (Table 8) is showing Users table information:

<table>
<thead>
<tr>
<th>Row name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uid</td>
<td>This row stores user ids.</td>
</tr>
<tr>
<td>uname</td>
<td>This row stores user names.</td>
</tr>
<tr>
<td>uemail</td>
<td>This row stores user emails.</td>
</tr>
</tbody>
</table>

**Table 8: Users table stores user information**

**3-3) Codes**

The below mentioned files are this project main files. Missing any of these files will cause the project not to run properly or will result some errors.
3-3-1) Index.html Code Explanation

The below mentioned codes are taken form index file.
First, it contains a link to the CSS file and Javascript files at the head section of the page.

```html
<link id="stylesheet" href="style.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="libs/jscolor.js" ></script>
<script type="text/javascript" src="sketch.js" ></script>
```

After the head section, the body of the page is split to regions by the use of division elements (div).

```html
<body>
<div id="allcontainer">
  <div id="left">
    <div id="tools">
```
Result of the above mentioned code is displayed the below figure (figure 10).

![Diagram of div tags template](image.png)

**Figure 8: Index.html div tags template**

The tool section codes are described before so the only important thing in index.html file is *canvas* element. The *canvas* element is the area for sketching. Here is the code of *canvas* element.

```html
<div id="container">
  <canvas id="scene" width="612" height="510">
    Sorry, Your browser does not support HTML5.
  </canvas>
</div>
```
Above-mentioned tag creates canvas element in the page that has a 610 pixel width in 510 pixel height that generates the drawing area in the scene. The sentence inside \textit{canvas} tag is replaced with \textit{canvas} tag if the browser does not support HTML5 elements.

The canvas element is inside a div tag with id container. This container is a parent node for the main canvas. It is useful when the JavaScript wants to create temporary canvas. Temporary canvas is generated by the JavaScript during the \texttt{window.load} event that is in the beginning of \texttt{sketch.js} file and attached to container with \texttt{appendChild} method. More explanation about \texttt{sketch.js} file is covered in the following section (section 3-3-9).

3-3-2) \texttt{Manager.html} Code Explanation

The manager.html contains interface for \texttt{manager.php} and these two files along with the \texttt{manager.js} file are used to manage the boards in database, this way users are able to easily add, delete boards by using the manager tool.

```html
<html>
<head>
<meta charset="utf-8" />
<title>Colbrative Sketch Board</title>
<link href="manager.style.css" rel="stylesheet" type="text/css" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script type="text/javascript" src="manager.js"></script>
</head>
<body>
</body>
</html>
```
<div id="main">
  <div id="up">
    <p>
      <label>Board Name:</label>
      <input id="textField" class="text white" onFocus="this.value=null" value="Type Board Name Here..." type="text" />
      <input class="button orange" onClick="addNew();" value="Add New Board" type="button" />
      <input class="button orange" onClick="cleanAll();" value="Clean All Boards" type="button" />
      <input class="button orange" onClick="deleteAll();" value="Delete All Boards" type="button" />

      <span class="right">Thumbnails:
</span>
      <select id="thumbSize" onChange="changethumbSize(this.options[this.selectedIndex].value)">
        <option id="small" value="small">small</option>
        <option id="medium" value="medium">medium</option>
        <option id="large" value="large">large</option>
      </select>

      Refresh Delay:
      <select id="changetRefreshTimeout" onChange="changetRefreshTimeout(this.options[this.selectedIndex].value)">
        <option id="long" value="long">long</option>
        <option id="normal" value="normal">normal</option>
        <option id="short" value="short">short</option>
      </select>
    </p>
  </div>
</div>
3-3-3) Manager.js Code Explanation

First off there should be a variable declaration.

```javascript
var server_url = "manager.server.php";
var serverLoad = 2;
var sendQueue= new Array();
//var timer = setInterval(init, 3000);
var refreshTime = 7000;
var sc=10;
var firstTime = true;
```
After declaring the variables, `manager.js` will create functions for button actions:

```javascript
function addNew()
{
    serverConnect('addnew', document.getElementById('textField').value, null);
}

function deleteAll()
{
    serverConnect('deleteall', null, null);
}

function deleteBoard(bid)
{
    serverConnect("delete", null, bid);
}

function cleanAll()
{
    serverConnect('cleanall', null, null);
}

Then it will start `init` function that starts the whole script that is triggered in the window `onload` event:

```javascript
window.onload = function()
{
    serverConnect('getAll', null, null);
}

function init()
{
    serverConnect('getAll', null, null);
    serverConnect('getlast', null, null);
}```
The *init* function calling *serverConnect* function by setting the *getAll* and *getLast* values. This function connects to PHP server by using *XMLHttpRequest* object.

This object has an event named *onreadystatechange*, which triggers when PHP server changes the request status. For example return value could be 0,1,2,3,4. The value 4 means the connection has finished successfully and the received values are in *responseText* variable.

```javascript
function serverConnect(job, data, bid) {
    var xmlhttp = new XMLHttpRequest();
    url = server_url + '?job=' + job + '&data=' + data + '&bid=' + bid;
    xmlhttp.open("GET", url, true);
    xmlhttp.send(null);
    xmlhttp.onreadystatechange = function stateChanged() {
        if (xmlhttp.readyState == 4 &
            & xmlhttp.status == 200) {
            if (job == 'getAll') {
                // serverConnect('getAll',null,null);
                refreshOutput.board(xmlhttp.responseText);
                setTimeout(init, refreshTime);
            }
            if (job == 'getLast') {
                // serverConnect('getAll',null,null);
                refreshOutput.latest(xmlhttp.responseText);
            }
            if (job == 'getboardAll') {
                //console.log('getboardAll',xmlhttp.responseText);
                refreshOutput.thumbnail(xmlhttp.responseText);
                //refreshAddField();
            }
        }
    }
}
```
The `sendData` function utilizes an array named `itemsToSend` to make a queue for send all the data. If number of send data becomes large, `sendData` automatically queues it in the mentioned array to be sent later on.

```javascript
function sendData() {
    var itemsToSend = new Array();
    if(sendQueue.length > serverLoad){
        for(i=0; i< serverLoad;i++){
            itemsToSend.push(sendQueue[0]);
            sendQueue.shift();
        }
        serverConnect('set',itemsToSend,null);
    } else {
        for(i=0; i < sendQueue.length;i++){
            itemsToSend.push(sendQueue[0]);
            sendQueue.shift();
        }
        serverConnect('set',itemsToSend,null);
    }
    delete itemsToSend;
    refreshAddField();
}
```

This object has three sub functions:

i) refreshOutput.board

ii) refreshOutput.latest

iii) refreshOutput.thumbnail
As indicated by their names, `refreshOutput.board` refreshes the board, `refreshOutput.latest` refreshes the latest interaction in the below section of the page and `refreshOutput.thumbnail` refreshes the board thumbnails.

```javascript
var refreshOutput = {
    board: function(boardList) {
        var boardItems = boardList.split('/');
        var bid, bname, bstatus, busers;
        var subitem = [];
        var output = '';
        output += '<table class="boardtable">';
        output += '<tr class="headrow"><td>ID</td><td>Board Name</td><td>Board Status</td><td>Thumbnail</td><td>Action</td></tr>';
        //console.log('boardItems', boardItems);
        for (var i = 0; i < boardItems.length - 1; i++) {
            subitem = boardItems[i].split('&');
            bid = subitem[0];
            bname = subitem[1];
            bstatus = subitem[2];
            busers = subitem[3];

            //console.log('bid' + bid);
            serverConnect('getboardAll', null, bid);
            output += '<tr><td>' + bid + '</td><td>' + bname + '</td><td>' + bstatus + '</td><td><div id="thumb' + bid + '"></div></td><td>Action</td></tr>'';
        }
        //console.log('board item update: ' + i, bid + ' ' + bname + ' ' + bstatus + ' ' + busers);
        output += '</table>';"};
```
output += '<img width="(612 / sc)" height="(510 / sc)" src="libs/img/no.jpg">';
    firstTime= false;

    output += '</div></td><td><a href="#" onclick=deleteBoard('+ bid +');> DELETE </a></td></tr>';
}
output += '</table>);
document.getElementById('availibleBoards').innerHTML = output;
},
latest: function(data){
    var points = data.split('/');
    var output = ";
    for (var i = 0; i < points.length - 1; i++) {
        var point = points[i].split(',);
        var rx = point[0];
        var ry = point[1];
        if (rx == 's') {
            eraseMod = false;
            output += '<strong>Action:</strong> Draw<br>);
        } else if (rx == 'r') {
            output += '<strong>Action:</strong> Erased <br><strong>Points:</strong> (start) :
        } else if (rx == 'c') {
            output += '<strong>Action:</strong> Chat<br><strong>Texts:</strong> (start);
        } else if (rx == 'lw') {
            output += '<strong>Style:</strong> line Width' + ry;
            output += '<br><strong>Points:</strong> (start) :
        } else if (rx == 'c') {
            output += '<strong>Action:</strong> Chat<br><strong>Texts:</strong> (start)';
        } else if (rx == 'lw') {
            output += '<strong>Style:</strong> line Width' + ry;
} else if (rx == 'lc') {
    output += ' - <font color="#' + ry + '"> Color #' + ry + '<br><strong>Points: </strong>(start) ';
} else if (rx == 'e') {
    output += '(end)';
} else {
    output += '(' + rx + ',' + ry + ')';
} //if
} //for

document.getElementById('latest').innerHTML = output;
}
,
thumbnail: function(data){
    var canvasTemp = document.createElement('canvas');
    if (!canvasTemp) { alert('Error: cannot create canvas element!'); } 
    var contextTemp = canvasTemp.getContext('2d');
    canvasTemp.id = 'temp';
    canvasTemp.width = 612 /sc;
    canvasTemp.height = 510 /sc;
    var points = data.split('/');
    var eraseMod = false;
    var chatMode = false;
    var IWidth = 1;
    var IColor = '#F60';
    var IType = 'pencil';
    var lineCap = 'round';
    var lineJoin = 'round';
    var bid  
    var ew = 30;
contextTemp.lineCap = lineCap;
contextTemp.lineJoin = lineJoin;

for (var i = 0; i < points.length - 1; i++) {
    var point = points[i].split(',);
    var rx = point[0];
    var ry = point[1];

    if (rx == 's') {
        eraseMod = false;
        contextTemp.beginPath();
    } else if (rx == 'r') {
        eraseMod = true;
    } else if (rx == 'c') {
        chatMode = true;
    } else if (rx == 'd') {
        bid = ry;
    } else if (rx == 'lw') {
        contextTemp.lineWidth = ry/sc;
    } else if (rx == 'lc') {
        contextTemp.strokeStyle = '#' + ry;
    } else if (rx == 'e') {
        if (eraseMod) {
            eraseMod = false;
        } else if (chatMode) {
            chatMode = false;
        } else {
            contextTemp.stroke();
        }
    }
}
This little function allows the users to change the size of thumbnails quickly.

```javascript
function changethumbSize(size){
    switch (size){
    case 'small':
        sc=10 ;
        break;
    }
}
```
In addition, this function is giving an option to change the refresh time.

```javascript
function changeRefreshTimeout(time){
    switch (time){
        case 'long':
            refreshTime=7000 ;
            break;
        case 'normal':
            refreshTime=3000 ;
            break;
        case 'short':
            refreshTime=1000 ;
            break;
    }
    window.stop();
    init();
}
```
### 3-3-4) Manager.server.php Code Explanation

The first step is to include the configuration file:

```
include('config.php');
```

Then there should be a check if **HTTP GET** is set or not:

```
if(isset($_GET['job'])){ $job = $_GET['job']; }
if(isset($_GET['data'])){ $data = $_GET['data']; }
if(isset($_GET['bid'])){ $bid = $_GET['bid']; }
```

Then connection to the database:

```
if(!isset($MySQLConnection)){
    $MySQLConnection = mysql_connect( $dbHost, $dbUser, $dbPass );
    mysql_select_db( $dbName );
}
```

Then if the job variable was equal to *getAll*, then this part of code is executed, based on the condition, the require data will be retrieved from the database and sent to the JavaScript.

```
if($job=='getAll'){
    $Result = mysql_query( "SELECT * FROM boards" );
    if($Result){
        while( $Row = mysql_fetch_object( $Result ) ){
            echo $Row->bid.'&'. $Row->bname.'&'. $Row->bstatus.'&'. $Row->busers.'/';
        }
    } else {
        echo '0'; // RETURN ERROR
    }
}
```
If the job was \textit{getboardAll}, all boards and their information retrieved from database and send to JavaScript.

\begin{verbatim}
if($job=='getboardAll'){
    $Result = mysql_query( "SELECT * FROM sketches WHERE bid=".$bid ");
    if($Result){
        while( $Row = mysql_fetch_object( $Result ) ){  
            echo 'd,'.$bid.'/'.$Row->points;
        }
    } else {
        echo '0'; // RETURN ERROR
    }
}
\end{verbatim}

If the job was \textit{addnew}, the new board added to the database. Here there is no return value, the only return values are 1 for successful adding and 0 for error adding.

\begin{verbatim}
if($job=='addnew'){
    $Result = mysql_query( "INSERT INTO boards (bname,bstatus)
    VALUES (".$data ",".'fresh')" );
    if(mysql_query( $Result )){
        echo '1';
    } else {
        echo '0'; // RETURN ERROR
    }
}
\end{verbatim}
If the job was *cleanall*, all the information in the database about the board drawings will be deleted, but the boards itself will not be affected.

```php
if($job=='cleanall'){
    $Result = mysql_query( "TRUNCATE TABLE sketches" );
    if($Result ){
        $Result2 =mysql_query( "UPDATE boards SET bstatus = 'cleared'" );
        if( $Result2 ){
            echo '1'; //RETURN
        } else {
            echo '0'; // RETURN ERROR
        }
    } else {
        echo '0'; // RETURN ERROR
    }
}
```

For *getLast* job, the only latest boards interactions or activities are retrieved from the database and will be sent to JavaScript.

```php
if($job=='getlast'){
    $Result = mysql_query( "SELECT * FROM sketches ORDER BY sid DESC LIMIT 0 , 1" );
    if($Result){
        while( $Row = mysql_fetch_object( $Result ) ){
            echo $Row->points;
        }
    } else {
        echo '0'; // RETURN ERROR
    }
}
```
In delete command, only one board with specified bid value will be removed.

```php
if($job=='delete'){
    $Result = mysql_query( "DELETE FROM boards WHERE bid=".$bid );
    if($Result){
        while( $Row = mysql_fetch_object( $Result ) ){
            echo $Row->points;
        }
    } else {
        echo '0'; // RETURN ERROR
    }
}
```

### 3-3-5) Manager.style.css Code Explanation

All buttons in the `manager.html` are made of some pure CSS3 codes and there are no images used in their styles. The nice orange colored gradient and shadow are generated by the use of CSS3 too. The shadow making code is as below. There is the `-webkit-box-shadow` code for chrome browser and the `-moz-box-shadow` working for Firefox browsers.

```css
text-shadow: 0 1px 1px rgba(0,0,0,.3);
box-shadow: 0 1px 2px rgba(0,0,0,.2);
-webkit-box-shadow: 0 1px 2px rgba(0,0,0,.2);
-moz-box-shadow: 0 1px 2px rgba(0,0,0,.2);
```
The gradient color maker codes are as below. Gradient styles make the buttons looks nicer.

background: -webkit-gradient(linear, left top, left bottom, from(#09C), to(#03F));

background: -moz-linear-gradient(top, #09C, #03F);

3-3-6) Config.php Code Explanation

This file contains MySQL server information and database name.

```php
<?php
$dbHost = 'localhost';
$dbUser = 'root';
$dbPass = '';
$dbName = 'sketch';
?>
```

3-3-7) Saveimage.php Code Explanation

In the saveimage.php file, a method is used to make an image from the main canvas. This file is calling from save2jpg function from inside the sketch.js file.

```php
<?php
if (isset($GLOBALS['HTTP_RAW_POST_DATA']))
{
    $imageData=$GLOBALS['HTTP_RAW_POST_DATA'];
    $filteredData=substr($imageData, strpos($imageData, ',')+1);
    $unencodedData=base64_decode($filteredData);
```
$fp = fopen( 'output.png', 'wb' );
fwrite( $fp, $unencodedData);
fclose( $fp );
}
?>

3-3-8) Server.php Code Explanation

Server.php is very similar to manager.php but the only differences are in job and its orders. In all conditions, returning number zero means an error is occurred.

First includes config.php file and then connects to the database:

```php
include('config.php');

if(isset($_GET['points'])){ $points = $_GET['points']; }
if(isset($_GET['job'])){ $job = $_GET['job']; }
if(isset($_GET['bid'])){ $bid = $_GET['bid']; }
if(isset($_GET['bname'])){ $bname = $_GET['bname']; }
if(isset($_GET['bstatus'])){ $bstatus = $_GET['bstatus']; }
if(!isset($MySQLConnection)){
    $MySQLConnection = mysql_connect( $dbHost, $dbUser, $dbPass );
    mysql_select_db( $dbName );
}
```

This condition is just checking board status. This condition is calling repeatedly by JavaScript.

```php
if($job=='boardcheck'){
    $Result = mysql_query( "SELECT * FROM boards WHERE bid=".$bid );
}```
while( $Row = mysql_fetch_object( $Result ) ){
    //echo $Row->bid.'&'.$Row->bname.'&'.$Row->bstatus;
    echo $Row->bstatus;
}

This condition returns board list, board name and board status.

if($job=='boardlist'){
    $Result = mysql_query( "SELECT * FROM boards" );
    if($Result){
        while( $Row = mysql_fetch_object( $Result ) ){n
            echo $Row->bid.'&'. $Row->bname.'&'. $Row->bstatus.'&'. $Row->busers.'/';
        }
    } else {
        echo '0'; // RETURN ERROR
    };
}

This condition makes board status changed to drawing; therefore, all online
users understand that someone is drawing something at the time.

if($job=='drawing'){
    $Query = "UPDATE boards SET bstatus = 'drawing' WHERE bid=".$bid;
    //$Result = mysql_query( $Query );
    if(mysql_query( $Query )){
        echo 'drawing'; //RETURN
    } else {
        echo '0'; // RETURN ERROR
    };
}
This condition sets board status to idle. When no one is drawing in the board and the board is empty from users this job is called from JavaScript.

```php
if($job=='idle'){
    $Query = "UPDATE boards SET bstatus = 'idle' WHERE bid=".$bid;
    if(mysql_query( $Query )){
        echo 'drawing'; //RETURN
    } else {
        echo '0'; // RETURN ERROR
    }
}
```

In this condition all data for board number bid are retrieved and returned to JavaScript.

```php
if($job=='getAll' && $bid){
    $Query = "SELECT * FROM sketches WHERE bid=".$bid;
    $Result = mysql_query( $Query );
    if($Result){
        while( $Row = mysql_fetch_object( $Result ) ){
            echo $Row->points;
        }
    } else {
        echo '0'; // RETURN ERROR
    }
    //mysql_close($MySQLConnection);
```
This condition adds a new sketch, which is drawn by the user, to the database:

```php
if($job=='set'){
    mysql_query( "INSERT INTO sketches (bid, points) VALUES
    (".$bid.",".$points.")" );

    if(mysql_query( $Query )){
        echo 'Inserted to database';
    } else {
        echo '0'; // RETURN ERROR
    }
}
```

If a user clicks on clear board tool, this condition is called by the JavaScript.

```php
if($job=='clear'){
    $Query = "DELETE FROM sketches WHERE bid=" . $bid .";"
    if(mysql_query( $Query )){
        $Query2 = "UPDATE boards SET bstatus='cleared' WHERE bid=" . $bid .";"

        if(mysql_query( $Query2 )){
            echo 'cleared'; //RETURN
        } else {
            echo '1'; // RETURN ERROR
        }
    } else {
        echo '0'; // RETURN ERROR
    }
```
3-3-9) Sketch.js Code Explanation

This is the core JavaScript file of this project. At first, it defining global variables:

```javascript
var appName = 'cSketch';
var server_url = "server.php";
var serverInterval = 1000;
var icounterDefault = 100;
var firstTime = true;
var boardLock = false;
var icounter = icounterDefault;
var ew = 30; //eraser width
var cookieLife = 1; //Days
var currentBoardId = 1;
var uid = 1;
var theme = 'default';
var lWidth = 1;
var lColor = '#F60';
var lType = 'pencil';
var lineCap = 'round';
var lineJoin = 'round';
```

Add listener to the main window that listens to the load event:

```javascript
if (window.addEventListener) {
    window.addEventListener('load', bMain, false);
}
```
Most of things are managed in this main function. It first defines its local variables, and then creates `canvasTemp`, which is a temporary canvas for user drawing on it. This temporary canvas makes it easier to capture sketches from the user.

Then it starts to listen to the mouse events and button clicks.

```javascript
function bMain() {
    loadState();
    doBoard('boardlist');
    var canvasTemp, contextTemp; //Temporary Canvas

    var canvasMain = document.getElementById('scene');
    var contextMain = canvasMain.getContext('2d');
    var container = canvasMain.parentNode;
    var tool, tool_default = 'sketch'; //for later use to add more tool like line and text
    var cords = document.getElementById('cords');
    var points_to_send, points_recieved;

    canvasTemp = document.createElement('canvas');
    if (!canvasTemp) { alert('Error: cannot create canvas element!'); } 

    contextTemp = canvasTemp.getContext('2d');
    canvasTemp.id = 'temp';
    canvasTemp.width = canvasMain.width;
    canvasTemp.height = canvasMain.height;
    container.appendChild(canvasTemp);
```
function boardStart() {

    if (firstTime) {
        mBox(true, 'Connecting to server...'); ///mBox
        getData('firstTime');
    }

    tool = new tools[tool_default]();

    // event listeners.
    document.getElementById('clearScene').addEventListener('click', clearBoard, false);
    document.getElementById('clearScene').addEventListener('click', clearBoard, false);
    document.getElementById('chatSend').addEventListener('click', chatSend, false);

    canvasTemp.addEventListener('mousedown', event_canvas, false);
    canvasTemp.addEventListener('mousemove', event_canvas, false);
    canvasTemp.addEventListener('mouseup', event_canvas, false);

} //boardStart()

This function finds the mouse cursor position on the canvas and sends the information to tools function. If tools.started = true, it means user is drawing, otherwise this just passes x and y values to the coordinate displayer in the right side of the screen.

function event_canvas(current) {
    if (current.layerX || current.layerX == 0) { // firefox method
current._x = current.layerX;
current._y = current.layerY;
} else if (current.offsetX || current.offsetX == 0) { // opera method
    current._x = current.offsetX;
current._y = current.offsetY;
}
printCordinates(current._x,current._y);

// call tool's event handler
var doAction = tool[current.type];
if (doAction) {
    doAction(current);
}
} //event_canvas()

The sTimer is a timer object, which handles repetitive and loop events in this project.

// TIMER
var sTimer = setInterval( function() {

    icounter--;
    document.getElementById('points').innerHTML = ('Time to refresh page: ' + icounter);
    if (icounter <= 0) {
        icounter = icounterDefault;
        window.location.reload();
    }

    if (!boardLock) {
getData('timer');

    var img = '<img name="circlingball" style="vertical-align:middle;" src="theme/' + theme + '/circling-ball.gif">'
    logIt( img + img + img + 'Looking for changes' +img + img + img );

}, serverInterval);

Tool and its sketch method, with mousedown, mousemove, mouseup events, handle all are the procedure of drawing in canvas. There are also many key variables inside the tools, which manages attitudes of sketching and conditions of it.

var tools = {};

    // sketch
    tools.sketch = function() {
        var tool = this;
        this.started = false;

    //////////MOUSE DOWN
    this.mousedown = function(ev) {
        icounter = icounterDefault;
        boardLock = true;
        tool.started = true;
        doBoard('drawing');
        logIt('Drawing ....');
        console.log('Start Drawing...');

        if (lType == 'eraser') {

    }
points_to_send = 'r,r/';
logIt('Erasing .....'); //contextTemp.fillStyle = '#ddd'; // blue
} else {
points_to_send = 's,s/';
contextTemp.beginPath();
contextTemp.moveTo(ev._x, ev._y);
logIt('Drawing .....'); }\ncontextTemp.lineCap = lineCap;
contextTemp.lineJoin = lineJoin;
contextTemp.lineWidth = lWidth;
points_to_send += 'lw,' + lWidth + '/';
contextTemp.strokeStyle = lColor;
points_to_send += 'lc,' + lColor.slice(1) + '/';
points_to_send += (ev._x + ',' + ev._y + '/');
};

////////MOUSE MOVE
this.mousemove = function(ev) {
  if (tool.started) {

    if (lType == 'eraser') {
      contextMain.clearRect(ev._x - (ew / 2), ev._y - (ew / 2), ew, ew);
      contextTemp.clearRect(ev._x - (ew / 2), ev._y - (ew / 2), ew, ew);
      //just for safety
    } else {
contextTemp.lineTo(ev._x, ev._y);
contextTemp.stroke();
}
points_to_send += (ev._x + ',' + ev._y + '/');
logIt('X=' + ev._x + ' Y=' + ev._y); //just for log

} 

};


/////MOUSE UP

this.mouseup = function(ev) {
doBoard('idle');
if (tool.started) {
tool.started = false;
points_to_send += 'e,e/';
logIt('Sending to Server ..');
passData(points_to_send);
console.log('Start Syncing ...');
}
};

}; //tools.sketch


The *get*Data and The *pass*Data are two hearts and minds of this project. They handle the send/receive data procedure. This function sends queued data to the server.php to process and store.

//
//Data Operations
//

64
function passData(points) {
    var url = server_url;
    var xmlhttp = new XMLHttpRequest();

    url += "?points=" + points;
    url += "&job=set";
    url += "&bid=" + currentBoardId;

    xmlhttp.onreadystatechange = stateChanged;
    xmlhttp.open("GET", url, true);
    xmlhttp.send(null);

    function stateChanged() {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
            boardLock = false;
            logIt('Data is sync with server.);
            //alert(xmlhttp.responseText); // this will alert "true"
        }
    }
}

The getData process received data in its stateChanged function. Then do the appropriate action related to receive data.

function getData(action) {
    var url = server_url;
    var xmlhttp = new XMLHttpRequest();
    if (xmlhttp == null) { alert("Browser does not support HTTP Request"); }
if (action == 'firstTime') {
    url += '?job=getAll';
} else if (action == 'timer') {
    url += '?job=getAll';
} else {
    url += '?job=get';
}

url += '&bid=' + currentBoardId;

xmlhttp.onreadystatechange = stateChanged;
xmlhttp.open("GET", url, true);
//xmlhttp.open("POST", url, true); // for large data
//request.transport.abort();

xmlhttp.send(null);

} //getData()

The `stateChanged` function handles the response to the `XMLHttpRequest`. In this function, received data splitting into small items and then each item is sent to its own action. For example if the received item be a point with color red and width 10, then the function is sets

```
contextMain.strokeStyle=#red
contextMain.lineWidth=10
```

And then draws a line in the `rx,ry` position and then strokes the canvas.

```
function stateChanged() {
```
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
    if (!boardLock) {
        var data = xmlhttp.responseText;
        var points = data.split('/');
        var eraseMod = false;
        var chatMode = false;

        document.getElementById('chatTexts').value = "";

        contextMain.lineCap = lineCap;
        contextMain.lineJoin = lineJoin;

        for (var i = 0; i < points.length - 1; i++) {
            var point = points[i].split(' ,');
            var rx = point[0];
            var ry = point[1];

            if (rx == 's') {
                eraseMod = false;
                contextMain.beginPath();
                contextMain.moveTo(rx, ry);
            } else if (rx == 'r') {
                eraseMod = true;
            } else if (rx == 'c') {
                chatMode = true;
            } else if (rx == 'lw') {
                contextMain.lineWidth = ry;
            } else if (rx == 'lc') {

        } else if (rx == 'lw') {
            contextMain.lineWidth = ry;
        } else if (rx == 'lc') {

    } else if (rx == 'r') {
        eraseMod = true;
    } else if (rx == 'c') {
        chatMode = true;
    } else if (rx == 'lw') {
        contextMain.lineWidth = ry;
    } else if (rx == 'lc') {

} else if (rx == 'r') {
    eraseMod = true;
} else if (rx == 'c') {
    chatMode = true;
} else if (rx == 'lw') {
    contextMain.lineWidth = ry;
} else if (rx == 'lc') {

}
contextMain.strokeStyle = '#' + ry;

} else if (rx == 'e') {
    if (eraseMod) {
        eraseMod = false;
    } else if (chatMode) {
        chatMode = false;
    } else {
        if (tool.started == false) {
            contextMain.stroke();
            contextTemp.clearRect(0, 0, canvasTemp.width,
canvasTemp.height);
        }
    }
} else {
    if (eraseMod) {
        contextTemp.clearRect(rx - (ew / 2), ry - (ew / 2), ew, ew);
        contextMain.clearRect(rx - (ew / 2), ry - (ew / 2), ew, ew);
    } else if (chatMode) {
        document.getElementById('chatTexts').value += "user" + rx
+": " + ry + "\n";
    } else {
        contextMain.lineTo(rx, ry);
    }
} //if
} //for
};
if (firstTime) {

firstTime = false;

mBox(false, 'Board Ready.'); //mBox
}
} //stateChanged()

This function handles board operations. Such as changing status or drawing status, it also checks whether the board is cleared by other users or not.

```javascript
function doBoard(job) {
    var xmlhttp = new XMLHttpRequest();
    var url = server_url + '?bid=' + currentBoardId + '&job=' + job;

    xmlhttp.onreadystatechange = stateChanged;
    xmlhttp.open("GET", url, true);
    xmlhttp.send(null);

    function stateChanged() {

        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
            //mBox(true, xmlhttp.responseText);
            if (xmlhttp.responseText == 'cleared') {
                mBox(false, 'All cleared.);
                contextMain.clearRect(0, 0, canvasTemp.width, canvasTemp.height);
                contextTemp.clearRect(0, 0, canvasTemp.width, canvasTemp.height);
            } else if (xmlhttp.responseText == 'drawing') {
                mBox(false, 'Board idle');
            } else {

```
updateBoardList(xmlhttp.responseText);

This little function clears the board and tells the server that the board is clear thus clears all the client-side instance boards with this board id around the world.

function clearBoard() {
    mBox(true, 'Clearing all instance boards. Please wait...');
    logIt('Clearing...');
    doBoard('clear');
    icounter = icounterDefault;
}

This function gets the chat text from input text in the page and sends it to the server. This function is started when the send button in the page is clicked.

function chatSend() {
    var chattext = document.getElementById('chatInput').value;
    document.getElementById('chatInput').value = '';
    passData('c,c/' + uid + ',' + chattext + '/e,e/');
    console.log('passed:', chattext);
}

70
Finally start the listeners by calling `boardStart` function.

```
boardStart(); // RUN THE BOARD
```

The `selectTools` function, changes the sketcher attitudes according to the users clicks. It changes color and width of sketcher, also it changes current board tool to eraser tool.

```
function selectTools(name, width, color) {
  if (width) {
    lWidth = width;
    logIt('width:' + width);
  }
  if (color) {
    lColor = color;
    logIt('color:' + color);
  }
  if (name) {
    switch (name) {
      case 'brush':
        lType = 'brush';
        lWidth = 20;
        logIt('Brush tool selected');
        break;
      case 'pen':
        lType = 'pen';
        lWidth = 5;
        logIt('Pen tool selected');
        break;
      case 'eraser':
        lType = 'eraser';
        break;
    }
  }
}
```
The `mBox` function is a handler for dimmer dialogue box. It easily changes the visibility of dimmer dialogue by setting the `mBox` visibility to true or false.

```javascript
function mBox(visiblity, mboxText) {
    console.log(visiblity, mboxText);
    if (visiblity) {
        document.getElementById('dimmer').style.display = "block";
        document.getElementById('mbox').innerHTML = (mboxText + '<br><br><a onclick="mBox(false,null);" href="#">back</a>);
    } else {
        document.getElementById('dimmer').style.display = "none";
    }
}
```

The `logIt` is a function to change the log text. This text appears below the tools in the page.

```javascript
function logIt(logText) {
```

IWidth = ew;
logIt('Eraser tool selected');
break;

//case 'pencil':
default:
    IType = 'pencil';
    IWidth = 2;
    logIt('Pencil tool selected');
}
}
saveState();
```
The below `changeTheme` function changes the board theme by changing the page elements with files that are previously copied to theme folder as board themes.

```javascript
function changeTheme(themename) {

    console.log('Theme', themename);
    theme = themename;
    saveState();

    document.getElementById('stylesheet').href = ('theme/' + themename + '/style.css');
    document.body.style.background = ('#0000FF url(theme/' + themename + '/bg.jpg) no-repeat');
    document['pencil'].src = ('theme/' + themename + '/pencil.png');
    document['pen'].src = ('theme/' + themename + '/pen.png');
    document['brush'].src = ('theme/' + themename + '/brush.png');
    document['eraser'].src = ('theme/' + themename + '/eraser.png');
    document['clear'].src = ('theme/' + themename + '/clear.png');
    document['save'].src = ('theme/' + themename + '/save.png');
    document['print'].src = ('theme/' + themename + '/print.png');
    document['newpage'].src = ('theme/' + themename + '/newpage.png');
    document['refresh'].src = ('theme/' + themename + '/refresh.png');
    document['waitCircle'].src = ('theme/' + themename + '/waitCircle.gif');
```
Save2jpg is a function to save board as PNG image:

```javascript
function save2jpg() {
    mBox(true, 'Preparing content to save...'); ///mBox
    var canvas = document.getElementById('scene');
    var canvasData = canvas.toDataURL("image/png");
    ajax = new XMLHttpRequest();
    ajax.open("POST","saveimage.php",false);
    ajax.setRequestHeader('Content-Type', 'application/upload');
    ajax.onreadystatechange = ajaxStateChanged;
    ajax.send(canvasData);
    function ajaxStateChanged() {
        if (ajax.readyState == 4 && ajax.status == 200) {
            mBox(true, 'Image is ready to download. right-click on<br><br><font size=16><a target="_blank" href="output.png">download link</a></font><br><br>and choose save as.'); ///mBox
        }
    }
}
```

This function `printCoordinates`, Log coordinates to top of right bar

```javascript
function printCordinates(x,y) {
    //console.log('printCordinates', x,y);
    document.getElementById('cords').innerHTML = ('X=' + x + ' Y=' + y);
}
```

The `updateBoardList` is a function that updates available board lists in the system; therefore a user is able to click on any board name and change the board. In addition to board list, a link is generated below the list so a user is able
to access the board management page, which then able to add/remove boards by this link.

```javascript
function updateBoardList(boardList){
    var boardItems = boardList.split('/');
    var bid, bname , bstatus, busers;
    var subitem = [];
    var output;
    output = 'Click on board name to change the board. <br>Available Boards:<table class="boardtable">'
    //output += '<tr class="headrow"><td>id</td><td>name</td><td>status</td><td>user id</td></tr>'';
    output += '<tr class="headrow"><td id</td><td name</td><td status</td><td user id</td></tr>'';
    console.log('boardItems',boardItems);

    for (var i = 0; i < boardItems.length - 1; i++) {
        subitem = boardItems[i].split('&');
        bid = subitem[0];
        bname = subitem[1];
        bstatus = subitem[2];
        busers = subitem[3];
        //console.log('board item update: '+ i, bid + ' ' + bname + ' ' + bstatus + ' ' + busers);
        //output += '<tr><td>' + bid + '</td><td><a href="#" onclick=changeBoard('+ bid + ');>' + bname + '</a></td><td>' + bstatus + '</td><td>' + busers + '</td></tr>'';
    }
```

75
output += '<tr><td>' + bid + '</td><td><a href="" onclick=changeBoard(' + bid + ');>' + bname + '</a></td><td>' + bstatus + '</td></tr>;
}
output += '</table>};
output += '<a href="manager.html" target="_blank" >Manage Boards</a><br>';
document.getElementById('boardList').innerHTML = output;
}

Simply changes the board.

function changeBoard(bid){
    currentBoardId = bid;
    saveState();
    mBox(true, 'Changing board, Please Wait ...');
    location.reload(true);
}

The below group of functions manage cookies which covers write-cookies, read-cookies and change-cookies. The last function saveState, saves the state of the board. Therefore, by closing the browser or changing the situation, the pen color and width never changes.

function createCookie(name,value,days) {
    if (days) {
        var date = new Date();
        date.setTime(date.getTime()+(days*24*60*60*1000));
        var expires = ""; expires=""+date.toGMTString();
    }
```javascript
else var expires = "";

document.cookie = appName + ' ' + name + '=' + value + expires + '; path=/';
}

function readCookie(name) {
    var nameEQ = appName + ' ' + name + "=";
    var ca = document.cookie.split(';

    for(var i=0; i < ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0) == ' ') c = c.substring(1, c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length, c.length);
    }
    return null;
}

function eraseCookie(name) {
    createCookie(appName + ' ' + name, '', -1);
}

function saveState() {
    console.log('State Saved');
    if(isSet(currentBoardId)) createCookie('bid', currentBoardId, cookieLife);
    if(isSet(uid)) createCookie('uid', uid, cookieLife);
    if(isSet(lWidth)) createCookie('lWidth', lWidth, cookieLife);
    if(isSet(lColor)) createCookie('lColor', lColor, cookieLife);
    if(isSet(lType)) createCookie('lType', lType, cookieLife);
    if(isSet(theme)) createCookie('theme', theme, cookieLife);
```
function loadState() {

    if (readCookie('bid')) currentBoardId = readCookie('bid');
    if (readCookie('theme')) theme = readCookie('theme');
    if (readCookie('lWidth')) lWidth = readCookie('lWidth');
    if (readCookie('lColor')) lColor = readCookie('lColor');
    if (readCookie('lType')) lType = readCookie('lType');
    if (readCookie('uid')){
        uid = readCookie('uid');
    } else {
        uid = Math.floor(Math.random()*10000);
    }
    document.getElementById('colorPicker').value = lColor;
    document.getElementById('theme'+theme).selected = true;
    document.getElementById('currentTheme').innerHTML = theme;
    document.getElementById('currentBoard').innerHTML = currentBoardId;
    document.getElementById('currentUser').innerHTML = uid;
    console.log('State Loaded');
    changeTheme(theme);
}

function isSet( variable ){
    return( typeof( variable ) !== 'undefined' );
}
3-3-10) Style.css Code Explanation

This file contains style sheets. The important thing in this file is using CSS3 new features text-shadow and box-shadow. Text shadow creates blurred shadow under texts, and box shadow makes blurred shadows for rectangular tags such as div, table, etc.

Another interesting thing in style sheets is dimmer. When the page is loading or it is synchronizing which the server, a dialog box appears in front of the page and the page completely dim to black without it having to be refreshed. After loading or syncing, the page returns to the previous state. This dialog box and its dimmed background is just a CSS trick. In the index.html, there is a div tag with dimmer id that div tag covers the entire page in normal condition. The style codes of the div tag are listed below which the #dimmer is the background and the #dimmer div is the dialog box that is in front.

```css
#dimmer {
  position:absolute;
  background:#333;
  opacity:50;
  display:block;
  top:0;
  left:0;
  width:100%;
  height:100%;
  opacity:0.8;
}
#dimmer div {
  position:relative;
  background:#FFF;
}
```
This dimmer is hiding after the page load by below JavaScript code. The related function that controls visibility of dimmer is `mBox` function inside the `sketch.js` file.

```javascript
document.getElementById('dimmer').style.display = "none";
```
CHAPTER 4

**********

ANALYSIS AND RESULTS ON ENHANCING ONLINE LEARNING EXPERIENCE THROUGH A RIA BASED LEARNING APPLICATION

Meisam Moradi
Sabzali Musa Kahn
Jaffri Hanafi
Zahra Kakaei

4-1) Codes Portability Tests

To test portability of codes on different servers, the programing language that is used for the application is very important issue. As the project codes are in pure HTML, CSS, and JavaScript programming language, it does not need any extra add-on or plug-in installed on the server and if the client browser supports HTML5, the project can be executed on any server type such as Windows, Linux or is able to run on any client such as desktop and mobile devices.

4-2) Browsers Compatibility Tests

After many browser compatibility tests, the below results (figure 11) are achieved. The detailed test results and more information are available in appendix C. The below figure (figure 11) demonstrate the HTML5 compatibility comparison between six famous browsers (Chrome 12, IE 9 and 10, Firefox 5,
Opera 11, Safari 5.1). The below figure displays Chrome 12 and Firefox 5 are most compatible browsers with HTML5 therefore they are best browser to execute the project.

![Overall HTML5 support diagram for six browsers.](image)

This project’s most important HTML5 element is Canvas. By this figure and the test result of appendix C, all of browsers were fully supporting Canvas element. This means, it is possible to execute the collaborative sketch board project in all browsers and the best choice, as mentioned before is Chrome 12 and Firefox 5.

4-3) Speed Tests

The project application (collaborative sketch board) speed test made on iwebtool.com and webpagetest.org, which for all of the tests, the application address “http://sketch.zxq.net/R4” is the demo place of this project and it is just to test. This test host shows a bit of increase in the response time to external files such as image and CSS files. The below table (table 9) demonstrate the results:
<table>
<thead>
<tr>
<th>#</th>
<th>Release Name</th>
<th>Size KB</th>
<th>Load Time seconds</th>
<th>Average Speed Sec/KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The project release 1 (13-jul-2011)</td>
<td>2.51</td>
<td>0.66</td>
<td>0.26</td>
</tr>
<tr>
<td>2</td>
<td>The project release 2 (5-Jul-2011)</td>
<td>3.15</td>
<td>0.55</td>
<td>0.17</td>
</tr>
<tr>
<td>3</td>
<td>The project release 3 (30-Jun-2011)</td>
<td>3.64</td>
<td>0.48</td>
<td>0.13</td>
</tr>
<tr>
<td>4</td>
<td>The project release 4 (25-Jun-2011)</td>
<td>4.06</td>
<td>0.41</td>
<td>0.1</td>
</tr>
<tr>
<td>5</td>
<td>The project board manager for R4</td>
<td>1.92</td>
<td>0.98</td>
<td>0.51</td>
</tr>
</tbody>
</table>

Table 9: The project releases speed tests

The above-mentioned result shows improvement from changes made in the codes since the first release, and the improved application reduced page load from 0.26 to 0.10. It is acceptable and is a slightly increasing in performance.
4-3-1) Statistics for Single Page Load

4-3-1-1) Global

<table>
<thead>
<tr>
<th>Total HTTP Requests</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Size</td>
<td>98903 bytes</td>
</tr>
</tbody>
</table>

Table 10: Total http requests size

4-3-1-2) Object Size in Total

<table>
<thead>
<tr>
<th>Object type</th>
<th>Size (bytes)</th>
<th>Download @ 56K (seconds)</th>
<th>Download @ T1 (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>4160</td>
<td>1.03</td>
<td>0.22</td>
</tr>
<tr>
<td>HTML Images</td>
<td>54069</td>
<td>12.98</td>
<td>2.49</td>
</tr>
<tr>
<td>CSS Images</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total Images</td>
<td>54069</td>
<td>12.98</td>
<td>2.49</td>
</tr>
<tr>
<td>JavaScript</td>
<td>40007</td>
<td>8.37</td>
<td>0.61</td>
</tr>
<tr>
<td>CSS</td>
<td>667</td>
<td>0.33</td>
<td>0.20</td>
</tr>
<tr>
<td>Multimedia</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Other</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 11: Http response time in 56K and T1 internet connection based on object types
4-3-1-3) External Objects

<table>
<thead>
<tr>
<th>External Object</th>
<th>QTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total HTML</td>
<td>1</td>
</tr>
<tr>
<td>Total HTML Images</td>
<td>11</td>
</tr>
<tr>
<td>Total CSS Images</td>
<td>0</td>
</tr>
<tr>
<td>Total Images</td>
<td>11</td>
</tr>
<tr>
<td>Total Scripts</td>
<td>2</td>
</tr>
<tr>
<td>Total CSS imports</td>
<td>1</td>
</tr>
<tr>
<td>Total Frames</td>
<td>0</td>
</tr>
<tr>
<td>Total IFrames</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 12: Quantity of external objects

4-3-1-4) Download Times

<table>
<thead>
<tr>
<th>Connection Rate</th>
<th>Download Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.4K</td>
<td>79.65 seconds</td>
</tr>
<tr>
<td>28.8K</td>
<td>41.33 seconds</td>
</tr>
<tr>
<td>33.6K</td>
<td>35.85 seconds</td>
</tr>
<tr>
<td>56K</td>
<td>22.71 seconds</td>
</tr>
<tr>
<td>Connection</td>
<td>Time</td>
</tr>
<tr>
<td>----------------</td>
<td>---------</td>
</tr>
<tr>
<td>ISDN 128K</td>
<td>9.04 s</td>
</tr>
<tr>
<td>T1 1.44Mbps</td>
<td>3.52 s</td>
</tr>
</tbody>
</table>

Table 13: Download time for the project in different connections

Note that these download times are based on the full connection rate for ISDN and T1 connections. Modem connections (56Kbps or less) are corrected by a packet loss factor of 0.7. All download times include delays due to round-trip latency with an average of 0.2 seconds per object. With 15 total objects for this page, that computes to a total lag time due to latency of 3 seconds. Note also that this download time calculation does not take into account delays due to XHTML parsing and rendering.

### 4-3-1-5) Page Objects

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Size (bytes)</th>
<th>Type</th>
<th>File Name</th>
<th>Header size (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23772</td>
<td>SCRIPT</td>
<td>jscolor.js</td>
<td>241</td>
</tr>
<tr>
<td>1</td>
<td>21398</td>
<td>IMG</td>
<td>waitCircle.gif</td>
<td>226</td>
</tr>
<tr>
<td>1</td>
<td>16235</td>
<td>SCRIPT</td>
<td>sketch.js</td>
<td>241</td>
</tr>
<tr>
<td>1</td>
<td>5923</td>
<td>IMG</td>
<td>refresh.png</td>
<td>225</td>
</tr>
<tr>
<td>1</td>
<td>5351</td>
<td>IMG</td>
<td>print.png</td>
<td>225</td>
</tr>
<tr>
<td>1</td>
<td>4160</td>
<td>HTML</td>
<td>index.html</td>
<td>446</td>
</tr>
<tr>
<td>1</td>
<td>3586</td>
<td>IMG</td>
<td>clear.png</td>
<td>224</td>
</tr>
</tbody>
</table>
Table 14: A list of the project objects with their size and header size.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3521</td>
<td>IMG</td>
<td>eraser.png</td>
<td>224</td>
</tr>
<tr>
<td>1</td>
<td>2901</td>
<td>IMG</td>
<td>pen.png</td>
<td>224</td>
</tr>
<tr>
<td>1</td>
<td>2658</td>
<td>IMG</td>
<td>newpage.png</td>
<td>244</td>
</tr>
<tr>
<td>1</td>
<td>2482</td>
<td>IMG</td>
<td>pencil.png</td>
<td>244</td>
</tr>
<tr>
<td>1</td>
<td>2319</td>
<td>IMG</td>
<td>brush.png</td>
<td>244</td>
</tr>
<tr>
<td>1</td>
<td>1966</td>
<td>IMG</td>
<td>circling-ball.gif</td>
<td>580</td>
</tr>
<tr>
<td>1</td>
<td>1964</td>
<td>IMG</td>
<td>save.png</td>
<td>244</td>
</tr>
<tr>
<td>1</td>
<td>667</td>
<td>CSS</td>
<td>style.css</td>
<td>222</td>
</tr>
<tr>
<td>15</td>
<td>98903</td>
<td>Total (unique objects)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It should be mentioned that this application is not using HTTP compression, otherwise called content encoding using gzip and it is consider compressing textual content (XHTML, JavaScript, etc.) with mod_gzip or similar functions. Also CSS alternate style-sheets may be referenced in the HTML but are not actually downloaded until they are needed and are therefore not included in the total page size.

4-3-1-6) Request/Response Headers:

The next page table (table 15) is the information about request phase. The project makes a request per second. In table 15 the only the first 25 requests that are more important, are presented and the detailed information is presented in appendix E.
<table>
<thead>
<tr>
<th>Request/Response Headers for first 25 requests</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Path</th>
<th>Host</th>
<th>Version</th>
<th>Duration</th>
<th>Bytes</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/api</td>
<td>localhost</td>
<td>v1</td>
<td>1.6 sec</td>
<td>1.6 KB</td>
<td>200</td>
</tr>
<tr>
<td>POST</td>
<td>/api</td>
<td>localhost</td>
<td>v1</td>
<td>2.1 sec</td>
<td>2.1 KB</td>
<td>200</td>
</tr>
<tr>
<td>GET</td>
<td>/api</td>
<td>localhost</td>
<td>v1</td>
<td>3.2 sec</td>
<td>3.2 KB</td>
<td>200</td>
</tr>
<tr>
<td>POST</td>
<td>/api</td>
<td>localhost</td>
<td>v1</td>
<td>4.4 sec</td>
<td>4.4 KB</td>
<td>200</td>
</tr>
<tr>
<td>GET</td>
<td>/api</td>
<td>localhost</td>
<td>v1</td>
<td>5.5 sec</td>
<td>5.5 KB</td>
<td>200</td>
</tr>
<tr>
<td>POST</td>
<td>/api</td>
<td>localhost</td>
<td>v1</td>
<td>6.7 sec</td>
<td>6.7 KB</td>
<td>200</td>
</tr>
<tr>
<td>GET</td>
<td>/api</td>
<td>localhost</td>
<td>v1</td>
<td>7.8 sec</td>
<td>7.8 KB</td>
<td>200</td>
</tr>
<tr>
<td>POST</td>
<td>/api</td>
<td>localhost</td>
<td>v1</td>
<td>8.9 sec</td>
<td>8.9 KB</td>
<td>200</td>
</tr>
<tr>
<td>GET</td>
<td>/api</td>
<td>localhost</td>
<td>v1</td>
<td>10.1 sec</td>
<td>10.1 KB</td>
<td>200</td>
</tr>
<tr>
<td>POST</td>
<td>/api</td>
<td>localhost</td>
<td>v1</td>
<td>11.2 sec</td>
<td>11.2 KB</td>
<td>200</td>
</tr>
</tbody>
</table>

**Request Details**

- **Method**: GET/POST
- **Path**: /api
- **Host**: localhost
- **Version**: v1
- **Duration**: 1.6 sec to 11.2 sec
- **Bytes**: 1.6 KB to 11.2 KB
- **Status**: 200
Figure 10: Object load time and the CPU and Bandwidth usage diagram

The below figure (Figure 12) is a list of requests with their CPU and Bandwidth usage diagram which is generated based on objects loading time and the next figure (Figure 13) is demonstrating these objects loading orders and connections.
Figure 11: Object loading orders and connections
4-3-2) Double Load Test

<table>
<thead>
<tr>
<th></th>
<th>Load Time</th>
<th>First Byte</th>
<th>Start Render</th>
<th>Elements</th>
<th>DOM</th>
<th>Time</th>
<th>Requests</th>
<th>Bytes In</th>
<th>Time</th>
<th>Requests</th>
<th>Bytes In</th>
</tr>
</thead>
<tbody>
<tr>
<td>First View</td>
<td>5.082 sec</td>
<td>0.74 sec</td>
<td>3.21 sec</td>
<td>77</td>
<td>5.08 sec</td>
<td>19</td>
<td>19 sec</td>
<td>6.33 sec</td>
<td>24</td>
<td>201 KB</td>
<td></td>
</tr>
<tr>
<td>Repeat View</td>
<td>2.556 sec</td>
<td>0.63 sec</td>
<td>1.83 sec</td>
<td>77</td>
<td>2.55 sec</td>
<td>19</td>
<td>3 sec</td>
<td>2.90 sec</td>
<td>22</td>
<td>3 KB</td>
<td></td>
</tr>
</tbody>
</table>

Table 16: The results of the project double view load test.

4-4) Comparison

In this section researcher is trying to compare the portability, compatibility and speed of the similar Adobe Flash version of the application, which tests are between SWF file in a HTML4 and JavaScript file in HTML5 page. The tested browsers are Safari, Firefox, Chrome, and IE9. The detailed test results are listed in appendix F.

4-4-1) Windows Tests
Safari would not play HTML5 videos; this means there was no way to test HTML5 behavior on it.

On Firefox Adobe Flash (SWF file) is using CPU from 6% up to 22%. However, HTML5 file is only using 7.5% CPU.

Chrome was better on Windows compared to MAC. Playback with SWF file based was 58% more efficient than HTML5 based page.

With IE version 8, SWF is using 15% to 22% CPU.

4-4-2) Mac Tests

In Firefox, SWF was slightly less efficient compared to Safari, but was better than Chrome.

CPU in SWF version 10 is increasing up to 32%.

With Safari, HTML5 was the most efficient and consumed less CPU resources compared to SWF, which CPU utilization was at 37.41%.

In Chrome the SWF and HTML5 efficiency were both equally around 50%

4-5) Conclusion of Results

This is the recommendation for final release of the application based on analyzing the result of this chapter which are made for the fourth release of the project based on web site optimization system and validated in world wide web consortium (validator.w3.org)

4-5-1) Number of Files

The total number of HTML files on this page (including the main HTML file) is one which most browsers can multithread. Minimizing HTTP requests is a key to web site optimization.
4-5-2) Number of Objects

There are 15 total objects on this page. From 12 to 20 objects per page, the latency due to object overhead makes up from 75% to 80% of the delay of the average web page. Reducing, eliminating, and combining external objects (graphics, CSS, JavaScript, iFrames, and XHTML) which will reduce the total number of objects, and thus separate HTTP requests. In addition, by use of CSS sprites the decorative images are very handy to be used.

4-5-3) Number of Images

There are moderate amount of images on the project main page. Using fewer images on a site or try reusing any same images in multiple pages will take advantage of caching. Using CSS techniques such as colored backgrounds, borders, or spacing instead of graphic techniques can help reduce HTTP requests.

4-5-4) Number of CSS Files

The total number of external CSS files on this page is one. Because external CSS files must be in the HEAD of HTML documents, is must be loaded first before any BODY content is displayed although it is cached CSS files, which slows down the initial display of the page. To do the CSS files would be placed in the head section of the page and JavaScript files at the end of the BODY section enables progressive display.

4-5-5) Page Total Size

The total size of this page is 98903 bytes that should be loaded in 22.71 seconds on a 56Kbps modem. Ideally, a page loads in 3 to 4 seconds on a broadband
connection, and 8 to 12 seconds on a dialup connection. Obviously, there is always room for improvement.

4-5-6) Number of Script Files

The total number of external script files on this page is two. External scripts are less reliably cached compared to the CSS files that are combining scripts, which minimizes the HTTP requests, or even embedding those into high-traffic pages will do it as well. Placing external JavaScript files at the bottom of BODY, and CSS files in the HEAD enables progressive display in XHTML web pages.

4-5-7) HTML File Size

The total size of this HTML file is 4160 bytes, which less than 50K. By specifying the HEIGHT and WIDTH of images, this size allows HTML to display content in less than 10 seconds, which is the average time that users are willing to wait for a page to display without feedback.

4-5-8) Image File Size

The total size of all images is 54069 bytes, which exceeds 50K. It creatively cropped or combined where appropriated. By the way, it is better to replace graphic text and form controls with styled text to eliminate unnecessary HTTP requests. Ideally, each image should be less than 1160 bytes then easily will fit into one TCP-IP packet.

4-5-9) Script File Size
The total size of external scripts is 40007 bytes, which is over 20K. It is optimized JavaScript size, combined, and then used the HTTP compression for any scripts placed in the HEAD of the document.

4-5-10) CSS File Size

The total size of external CSS file is 667 bytes, which is less than 8K and it is excellent.

4-5-11) Multimedia File Size

The total size of multimedia files in the project is zero bytes that it means no multimedia is in page.
5-1) Conclusion On Enhancing Online Learning Experience Through A RIA Based Learning Application

In conclusion, producing a web application with HTML5 and JavaScript gives fast response to the user requests. This is proven in the experiment results shown in Chapter four. Although HTML5 has more requests and responses to the web page (calling images by CSS files, adding JavaScript to page, calling xml, etc.), in contrast to SWF file which have just one request through the Adobe Flash to the server and one response back to it, however, html and JavaScript requests and responses sizes are small compared to Adobe Flash. For example in chapter four experiments, the size of project was 152972 bytes, which was a sum of html, CSS, JavaScript, and all the images that shown in below figure (figure 12).
According to results in Chapter 4, the application with same features in an adobe flash SWF is weighting more than 400,000 bytes, therefore using html specially HTML5 makes pages very lighter compared to Adobe Flash files in term of the file size and bandwidth used. This HTML5 RIA takes shorter time to load therefore faster response in user interactions contributes to better user experience, hence better learning through this HTML5 application.

Another advantage of html is portability. It means HTML5 can be easily moved from one server to another server and editing the files are easily made possible remotely by any ftp client. Also this application can reach any user of different platforms without additional plugins or side installations because HTML5 is supported by modern internet browsers now.

In terms of application authoring and source file update, Adobe Flash technology does not allow the remote editing of its SWF files. For Adobe Flash, the projects are compiled as an SWF package, therefore for editing, there must be an access to the source files and then there is a need to recompile the edited version again to an SWF file and then upload it to the server. This procedure is needed for each new edition. Also SWF files want a pre installation of Adobe

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>4160</td>
</tr>
<tr>
<td>HTML Images</td>
<td>54069</td>
</tr>
<tr>
<td>CSS Images</td>
<td>0</td>
</tr>
<tr>
<td>Total Images</td>
<td>54069</td>
</tr>
<tr>
<td>JavaScript</td>
<td>40007</td>
</tr>
<tr>
<td>CSS</td>
<td>667</td>
</tr>
<tr>
<td>Total</td>
<td>152972</td>
</tr>
</tbody>
</table>

*Figure 12: Size (in bytes) of project objects*
Flash technology on the browser in form of side plugin or add-on. Thus HTML5 is more functional for learning system developers and learning content designers.

The last thing is elements priority in page loading order. Most web browsers load HTML main tags such as DIV, TABLE and TD at first and display page structure to the user. Then text, images, and media are loaded into page. Therefore, the user will see the frames and most of the time all the texts in the page instantly as soon as he goes to the page. These priorities speed up the page content access time. But Adobe Flash compiled SWF file needs to be loaded first. The Adobe Flash XML based environment also needs a loading time to load its environment first. All of us remember the boring progress bar at the beginning of flash websites.

Overall, this project takes HTML5 as its programming base and gives good results as discussed in previous chapters. In addition, nowadays there are a number of running sites using HTML5 with combination of JavaScript or its additional powerful libraries like JQuery. Therefore, with respect to Adobe Flash programmers and designer, it is expected that online learning experience is going to have a success future by utilize HTML5 and JavaScript capabilities.
References


