

Graduation Project Report

Cross-Platform Image Gallery

JavaScript Plugin



ILUMY® digital innovation

 **Fontys**
University of Applied Sciences



University of Applied Sciences

GRADUATION PROJECT REPORT
FONTYS UNIVERSITY OF APPLIED SCIENCES
HBO-ICT: English Stream

Data student:

Suprianto, Daniar Gita Nindiary

Student number: 2186211

Graduation project period: from 30 January 2012 till 15 June 2012

Data company:

Name company/institution: ILUMY B.V.

Department: Development

Location: Waterlooplein 125, 1011PG, Amsterdam

Hoevenaar, Stefan (Managing Director)

Data company tutor:

Hoevenaar, Stefan

Data final report:

Title of report: Cross-Platform Image Gallery JavaScript Plugin

Date final report has been issued: June 5th, 2012

Approved and signed by the company tutor:

Date:

Signature company tutor:

Name company tutor:

PREFACE

I joined EXISTENTE, a double degree program, which is a program for students who are willing to take their final year in Dutch University. I took 3 years studying in my home country before I take my final year in here.

As a final year student of Information & Communication Technology at Fontys University of Applied Sciences, I need to undergo an internship in a company. The internship itself must take place at least 90 working days. After looked for an internship among several companies, finally I chose ILUMY because the assignment matches with my interest, web development. I started being an intern at ILUMY on 31 January 2012 and will finish on 15 June 2012.

During my internship program I did several works, from training assignments until doing my own project. Meanwhile, I must make an internship report to be delivered in the end of my internship period of time

The purpose of this report is to explain what I did and learned during my internship with ILUMY as a web developer. This report is also a requirement for the partial fulfillment of graduation project thesis. It focuses primarily on the research and the assignment handled.

The research I did is about HTML 5 and its compatibility among several different browsers. Moreover, I have to figure out how HTML 5 can work as a web app in several mobile devices, additional to touch/gesture and sensors. It's about screen behavior like resolution and orientation as well. This research is served as the basis to do my assignment to make a jQuery plugin for an image gallery.

This opportunity to work in a real company is such an amazing experience for me. I learned a lot here, from how to initialize a project until deliver it to the client. I have a nice working environment and colleagues who are very helpful. We work in a small team yet we could have done many big works. That's such a big achievement.

I would like to thank my school tutor, Mr. Ad Maas and my company tutor, Mr. Stefan Hoevenaar, since they're very helpful during my internship. I would like to thank all my colleagues at ILUMY as well for the support.

I am not going to forget all the experiences I gained. It was terrific!

May 12th, 2012
Danar Gita Nindiyanto Suprianto

TABLE OF CONTENT

| | |
|--|------------|
| PREFACE | i |
| TABLE OF CONTENT | ii |
| TABLE OF FIGURES | iii |
| SUMMARY | 1 |
| GLOSSARY | 2 |
| 1. INTRODUCTION | 4 |
| 2. THE COMPANY..... | 6 |
| 2.1. About ILUMY | 6 |
| 2.3. The Structure of ILUMY..... | 7 |
| 2.4. ILUMY's Recent Projects | 7 |
| 3. INTERSHIP ASSIGNMENT | 10 |
| 3.1. Assignment overview | 10 |
| 3.2. Deliverables and Non-Deliverables..... | 11 |
| 4. Research..... | 12 |
| 4.1. HTML 5..... | 12 |
| 4.1.1. Overview | 12 |
| 4.1.2. Build a Cross-Browser HTML 5 Website ^[11] | 13 |
| 4.1.3. HTML 5 Compatibility | 14 |
| 4.2. Mobile Web Application | 17 |
| 4.2.1. Fragmentation..... | 18 |
| 4.2.2. Gesture | 18 |
| 4.2.4. How to Develop a HTML 5 Web Application ^[20] | 22 |
| 5. DEVELOPMENT METHODOLOGY..... | 26 |
| 6. IMPLEMENTATION..... | 29 |
| 6.1. The idea | 29 |
| 6.1.1. General process at glance..... | 29 |
| 6.1.2. Element wrapping..... | 29 |
| 6.1.3. Image size calculation..... | 31 |
| 6.2. Optional features adding..... | 32 |
| 6.2.1. Image navigation..... | 35 |
| 6.2.2. Image rotating and scaling | 36 |
| 6.2.4. Shake the device..... | 40 |
| 6.2.5. Image manager | 41 |
| 6.2.6. Server side image cropping and resizing | 43 |
| 6.2.7. Image preloading..... | 45 |
| CONCLUSION AND RECOMMENDATION | 46 |
| EXPERIENCE DURING GRADUATION PROJECT | 47 |
| BIBLIOGRAPHY | 48 |
| APPENDIX A: PROJECT PLAN | 50 |
| APPENDIX B: REQUIREMENT DOCUMENT | 61 |
| APPENDIX C: PROOF OF CONCEPT | 66 |
| APPENDIX D: PRODUCT DOCUMENTATION | 75 |
| APPENDIX E: TEST REPORT | 86 |

TABLE OF FIGURES

| | |
|---|----|
| Figure 2.1 Organizational Structure | 7 |
| Figure 2.2 Delta Lloyd Cross-Platform Ezine..... | 8 |
| Figure 2.3 Milk or Sugar | 8 |
| Figure 2.4 Safeberg | 9 |
| Figure 4.1 Opera Error Handling | 12 |
| Figure 4.2 Predicted Growth of Mobile Web Access | 17 |
| Figure 4.3 How to Make HTML 5 Mobile App Chart..... | 22 |
| Figure 4.4 Set Up The Server Chart..... | 22 |
| Figure 4.5 Make the HTML 5 App Chart | 23 |
| Figure 4.6 The JavaScript Chart..... | 24 |
| Figure 4.7 The CSS Chart | 25 |
| Figure 5.1 Iterative Application Development..... | 26 |
| Figure 6.1 How the idea works | 29 |
| Figure 6.2 Basic HTML Code..... | 30 |
| Figure 6.3 Element Wrapping Flowchart..... | 31 |
| Figure 6.4 Image adjusting (left : by its width; right : by its height)..... | 32 |
| Figure 6.5 Image Size Calculation Flowchart..... | 32 |
| Figure 6.6 Call the Plugin Constructor | 33 |
| Figure 6.7 Call the Plugin Constructor and Set the Features | 34 |
| Figure 6.8 Optional Features Adding Flowchart..... | 34 |
| Figure 6.9 Image Navigation Illustration | 36 |
| Figure 6.10 Touch Event Flowchart..... | 38 |
| Figure 6.11 Gesture checking | 39 |
| Figure 6.12 Motion Coordinate..... | 40 |
| Figure 6.13 Motion Coordinate with iPad..... | 41 |
| Figure 6.14 File browser | 41 |
| Figure 6.15 Images dropping area..... | 42 |
| Figure 6.16 Upload process..... | 42 |
| Figure 6.17 Image original size (1152x864px)..... | 44 |
| Figure 6.18 Image is stretched | 44 |
| Figure 6.19 Image cropping process | 45 |
| Table 4.1 Desktop Browser Compatibility ^[9] | 14 |
| Table 4.2 Mobile Browser Compatibility ^[8] | 16 |

SUMMARY

This report contains the progress and result of my internship at ILUMY as a web developer. This assignment is carried out by me, Daniar Gita Nindiary Suprianto, as a final year student of Information & Communication Technology at Fontys University of Applied Sciences.

The web development became an interest of mine when I started studying Informatics in my home university in Indonesia, but I didn't have many chances to go deeper about it, because informatics is more focused on application development, instead of web development. First I came as an intern here I only had a brief insight about web development, therefore I need to learn by myself and of course with the help of my colleagues a little bit harder in order to improve my web programming ability.

Every project we did here has 4 general steps before deliver it to the client, interaction design, visual design, development, and the last but not least is testing. Since my position here is as web developer, I do development task and fix some bugs found during the testing phase. Besides involved in some projects development in ILUMY, I have my own project to develop a jQuery plugin for image gallery.

The main requirement for this plugin is it must work in different platforms, such as iOS, Android, and Windows Phone. To implement the touch and gesture function to this plugin is the main challenge during the development process. Before starting the development process, I did a research corresponding to this project. From that research, I can finish the project right in time because I got a lot of knowledge from the result of the research, it does really help. Though there were several problems during the process, I could face it with the help of my colleagues who have more experience in developing website things.

GLOSSARY

| Terms | Explanation |
|--------------|---|
| Array buffer | A data type that is used to represent a generic, fixed-length binary data buffer |
| Attribute | Attributes provide additional information about an element |
| Base-64 | Base-64 encoding is used to convert binary data into a form that survives transmission across a network, so the data passed in its input argument is really binary data. It is carried in a string value because JavaScript doesn't support a special binary container. |
| Blob | A large file, typically an image or sound file, that must be handled (for example, uploaded, downloaded, or stored in a database) in a special way because of its size |
| CSS | Short for Cascading Style Sheets, a new feature being added to HTML that gives both Web site developers and users more control over how pages are displayed |
| Data type | A set of data with values having predefined characteristics. Examples of data types are: integer, floating point unit number, character, string, and pointer |
| Decoding | Is the process of converting code into plain text or any format that is useful for subsequent processes. Decoding is the reverse of encoding |
| Deliverable | Report or item that must be completed and delivered under the terms of an agreement or contract. |
| Element | An HTML element is everything from the start tag to the end tag |
| Emulator | A software application that can accurately imitate another computer, mobile phone, device or even arcade machine and run software from that computer |
| Encoding | The process of putting a sequence of characters (letters, numbers, punctuation, and certain symbols) into a specialized format for efficient transmission or storage |
| Fallback | A mechanism for carrying forth programmed instructions despite malfunction or failure of the primary device. |
| Flowchart | A flow chart is a graphical or symbolic representation of a process. Each step in the process is represented by a different symbol and contains a short description of the process step. The flow chart symbols are linked together with arrows showing the process flow direction. |
| HTML | The set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page |
| IETF | Internet Engineering Task Force. Internet standards, cooperating closely with the W3C and ISO/IEC standards bodies and dealing in particular with standards of the TCP/IP and Internet protocol suite. |

| | |
|----------------------|---|
| Interaction design | The practice of designing interactive digital products, environments, systems, and services. Interaction design is heavily focused on satisfying the needs and desires of the people who will use the product |
| JavaScript | A scripting language developed by Netscape to enable Web authors to design interactive sites |
| Markup | A set of labels that are embedded within text to distinguish individual elements or groups of elements for display or identification purposes. The labels are typically known as "tags." |
| Operating system | A computer program that manages the resources of a computer. It accepts keyboard or mouse inputs from users and displays the results of the actions and allows the user to run applications, or communicate with other computers via networked connections. |
| PHP | A general-purpose server-side scripting language originally designed for Web development to produce dynamic Web pages. It is one of the first developed server-side scripting languages to be embedded into an HTML source document, rather than calling an external file to process data |
| Platform | In the computer world, a "platform" typically refers to a computer's operating system. For example, a Dell computer running Windows XP would be considered to be running on a Windows platform |
| Plugin | A hardware or software module that adds a specific feature or service to a larger system |
| Programming language | An artificial language used to write instructions that can be translated into machine language and then executed by a computer. |
| Quirk mode | A technique used by some web browsers for the sake of maintaining backward compatibility with web pages designed for older browsers, instead of strictly complying with W3C and IETF standards in standards mode |
| Script | A program or sequence of instructions that is interpreted or carried out by another program rather than by the computer processor (as a compiled program is). |
| SDK | Short for Software Development Kit, a programming package that enables a programmer to develop applications for a specific platform |
| Syntax | The set of rules that define the combinations of symbols that are considered to be correctly structured programs in that language |
| Tag | A standard mark-up code in an HTML file that specifies how the Web browser should interpret the file contents. The tag starts with the less than character (<) and ends with the greater than character (>) |
| W3C | The main standards organization for the World Wide Web |

1. INTRODUCTION

Internet and websites are now like our daily need. Several years ago people only could access website through their desktop, but as we can see today they can surf the Internet using their phone. Smartphones, iPhone, and tablets are booming these days, not only in this country but also worldwide. Many users don't know that it becomes a big issue for web developer because each of those devices has different resolution, different browser, and of course different operating system.

It is a challenge for the developers and designers to make the website as attractive as possible. After asking some people (common users, not IT people) for their opinion about "What is the most important aspect in a website?", it turns out that mostly answered "user friendly", and the second aspect is "speed". For instance, I notice people habit when they use their touch screen device to open an image gallery. After they finish looking at one image, their fingers spontaneously swipe the screen to go to the next image. It will be so annoying if we have to push the next or previous button to see other images.

The aim of the project is to develop an interactive image gallery plugin by implementing touch/gestures interaction and sensor on different devices such as (iPad/iPhone/Android/Windows Phone) in order to make the website becomes more user friendly. It's also about screen behavior like resolution and orientation. For instance, when the device is being rotated, the image is also rotated or we can rotate image in the web app by moving our fingers. When the screen is pinched, the image will be zoomed. When we swap the image, the image will move and so on.

Somehow, not all these functions can be applied to all devices, therefore I have to find out what function can be applied, what cannot, and of course how to apply those functions. Touch or gesture and sensor depend on the device indeed, but the way to respond information given by the device sensor is different for every operating system.

This report contains all the activities and final result that delivered while doing internship at ILUMY. The structure of this report is as follows:

- a. This chapter gives an overview about the company, what they do and what project they have done.
- b. The next chapter comes out with more detailed assignment of the graduation project. Besides, the deliverable and non-deliverable products will be listed in this chapter.
- c. During the internship period I did many research in order to be able to do the assignment. The result of the research will be served in chapter 4.
- d. When developing a product usually a company or a developer uses a certain method. This time, I use IAD method. A clear definition of this method is on chapter 5.
- e. The last chapter explains about the implementation of project. How do I come up with the idea, the detailed of each developing process, etcetera are mentioned in this chapter.

2. THE COMPANY

2.1. About ILUMY

ILUMY is a digital design and innovation firm. They merge technological and social innovation to create new products and online experiences. They analyze, plan, design and deliver in a breeze. ILUMY designs inspiring and realistic views of the future, casting stories of the future in storyboards, movies or info-graphics. They analyze future trends for your customers, employees, technology, channels and processes. Venturing, for ILUMY, is getting ideas to reality. From idea to new business, website, application, product or service.

ILUMY gives input to, and help create customer's concept, propositions and business case. Bringing in our customer-centered mania, creativity and giving spicy feedback along the way. They create and improve user-centered experiences for web, intranets and devices. To position customer's brand, convey a message, increase conversion, improve productivity or simply to create a great experience.

ILUMY knows digital technology. They build in a breeze. They can help customer to choose the right platforms or develop new technology. They build websites, intranets, mobile applications, dashboards and team tools. Their core technologies include Symfony, Drupal, OpenAtrium, Wordpress, BuddyPress, Typo3, Magento, xCode/iOS, Eclipse/Android, Magento, HTML5, MySQL, Hadoop, C# and, if needed, .NET. And we know SaaS, including Ning, Wix, MailChimp, Sysomos Heartbeat, Radian6, Scoutlabs and HootSuite. They also provide transitional or long term high performance and secure hosting services.

2.3. The Structure of ILUMY

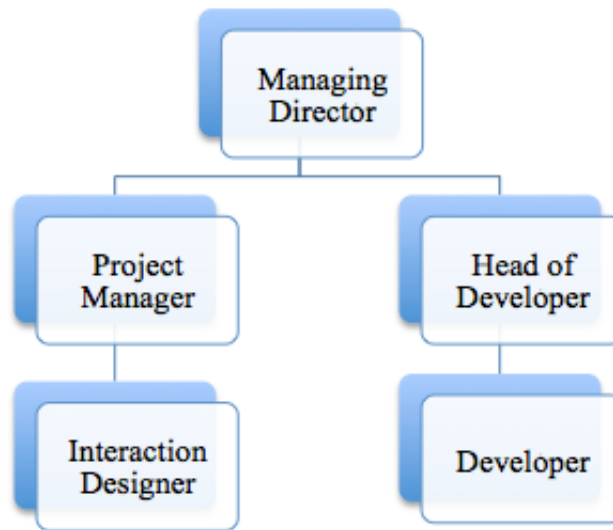


Figure 2.1 Organizational Structure

| | |
|----------------------------|---------------------------------------|
| Stefan Hoevenaar | Managing Director |
| Sjoerd Fit | Project Manager |
| David Clarisse | Head of Development |
| Finde Xumara | Developer |
| Luis Alberto Gomez Maestre | Developer |
| Murielle Smit | Project Management and Administration |

2.4. ILUMY's Recent Projects

a. Delta Lloyd Cross-Platform Ezine

An online magazine can be a very effective tool for internal communications, offering a compact, focused, informative, interactive, cover-to-cover experience. With ILUMY's partners Proof Reputation and Atelier van Gog, they developed a cross platform e-zine format for Delta Lloyd Group, based on their shared insights of what an e-zine should be and not be.

The new e-zine supports all types of copy and media, has room for user feedback and provides a smooth experience on any tablet/screen.



Figure 2.2 Delta Lloyd Cross-Platform Ezine

b. Milk or Sugar

Initially to serve ILUMY's research and curiosity, they built a database of websites where user can customize and order products online. They added reviews and the database became a site, offering the best places to go for users to order sleeping bag, samurai sword, bicycle, skate board, scrabble board, watch, machine parts, perfume, lingerie, wallpaper or lego model. And soon, user's built to order fire extinguisher, guitar, radio controlled car, protein shakes or kanu.

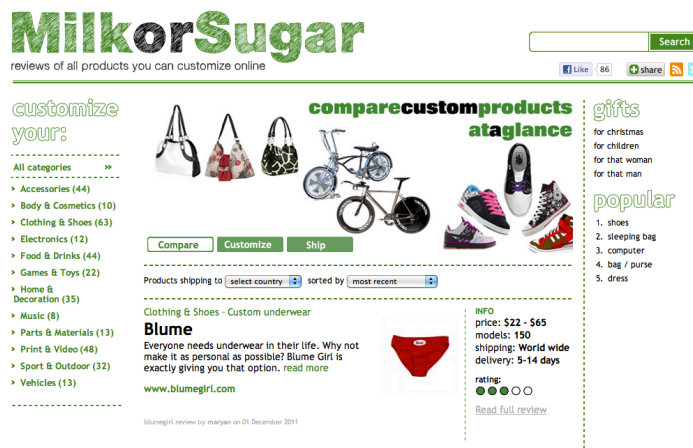


Figure 2.3 Milk or Sugar

c. Safeberg

ILUMY launched Safeberg: online backup with ultimate privacy and ultimate security. The unique paper key featured in many media. Now user can backup their (professional) life and relax.



Figure 2.4 Safeberg

The projects you see above are just small parts of their works. Of course they have done many more big projects, for big companies, such as Trespa, Tom Tom, etcetera.

3. INTERNSHIP ASSIGNMENT

3.1. Assignment overview

The project is to develop an interactive image gallery plugin by implementing touch/gestures interaction and sensor on different devices such as (iPad/iPhone/Android/Windows Phone) in order to make the website more user-friendly. It's also about screen behavior like resolution and orientation. For instance, when the device is being rotated, the image is also rotated or we can rotate image in the web app by moving our fingers. When the screen is pinched, the image will be zoomed. When we swap the image, the image will move and so on.

ILUMY has been doing many web apps, but they emphasize their web apps compatibility to iOS devices such as iPhone or iPad. While as we all know, the emerging devices are not merely monopolized by Apple only, Android and Windows Phone nowadays are taking part into this rivalry. Therefore ILUMY has to start adapting to this situation by expanding their web app products compatibility to be able to work on those devices. Based on this consideration, the first challenge on developing the assignment is to make it works on multi-platform devices, in this case, they are iOS devices (iPhone, iPad), Android devices (tablet), Windows Phone, and of course on the desktop.

Since we are trying to improve the compatibility to work on multiple platforms, we have to deal with the features that are provided by every device. In this case, I talk about finger interaction. Most of those devices (iPad/iPhone/tablet) are touch screen devices, therefore the only way to interact with it is by using our fingers touch. Most people who are using touch screen devices already know the common ways of interaction. I mean, it's such a habit, if they see a slideshow, they will swipe the image to move to another image. They will pinch the screen to zoom out the display, and so on. Those are basic touch features that are provided by the devices. But it turns out that if we make an application, we determine the way our application to response those functions. For instance, when user is using our application, we determine what will happen when

user taps a specific element, what will happen when user swipes a specific element, what will happen when user pinches the screen, and so on. It becomes the second challenge to develop this project, to make the gallery as interactive as possible according to user's common behaviors. That's a way to make a user-friendly image gallery.

This project will be developed in PHP Storm using JavaScript, PHP, and HTML5 programming languages. To create the script, I will use a JavaScript library called jQuery in order to simplify my script and make it more understandable. For the testing purpose, I will use iPhone, iPad and Android tablet. To test the gallery on Windows Phone, I will use an emulator provided by Microsoft because nobody has Windows Phone, so probably the accuracy is not that perfect.

3.2. Deliverables and Non-Deliverables

Project deliverables:

- jQuery plugin files and PHP files
- Project plan
- Requirement document
- Product documentation
- Test report

Non-deliverables:

- User manual

4. Research

4.1. HTML 5

4.1.1. Overview

HTML 5 is being developed as the next revision of HTML. It's an ongoing effort to introduce new elements and attributes in order to give easier and more structured way to develop a website or web app. There are several reasons why is HTML 5 better than the old HTML ^[19]:

- Backwards compatible

The websites or applications already written in HTML 4 can be smoothly changed to HTML 5 by simply adding an HTML 5 doctype and introducing HTML 5 tags and features as it makes sense to do so.

- Better error handling

The error handling and parsing rules of earlier HTML specifications are up to the browsers themselves. Users can see the effect of badly formed code because the error is exposed.

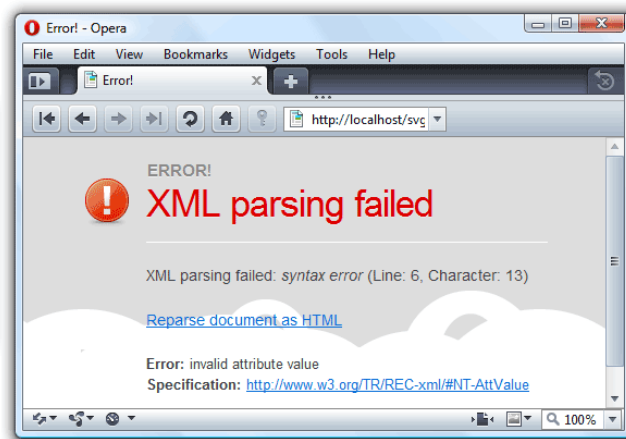


Figure 4.1 Opera Error Handling

HTML 5 doesn't expose errors to the user which make the website a better experience for users. HTML 5 just ignores unknown stuff or code and move on just like CSS does. HTML 5 specifications know how errors should be handled across different browser engines with

the intent that different browsers will produce the same result in the case of incorrect syntax. It means that the error handling is not up to the browsers themselves.

- Better rich media support, features that are being used today such as graphic, audio, video, etc.

These are some HTML 5 features that make significant enhancement in web development^[18]:

- Native audio and video support
Standardizing video and audio using HTML5 allows those media to work on iOS devices, which don't support flash
- Offline storage
Store content of the website on the computer or mobile device when not connected to the Internet to allow the application to behave more like a native application.
- Canvas
Draw anything from graphs or any other visual images right in the browser.
- Location services
Detect the location or share the location
- Inline document editing
Direct editing of a webpage right in the browser using HTML markup. This feature is useful for a website administrator if they want to manage the content.
- Enhanced form control
Placeholder text, autofocus, regular expressions, basic form validation and required field support. All those features can be used without the need of JavaScript.

4.1.2. Build a Cross-Browser HTML 5 Website^[11]

- a. Use `ul` and `li` elements to create a menu; don't use a table or a series of `p` elements.
- b. Place all visual formatting in CSS, HTML only for content.

- c. Validate the code before uploading the site using HTML validator and CSS validator. Usually we use “<http://validator.w3.org/>” as the official HTML validator and “<http://jigsaw.w3.org/css-validator/>” as the CSS validator.
- d. Avoid “quirks modes”. Generally a browser switches to quirks mode if the page doesn’t include a “DOCTYPE” (document type declaration). So to avoid quirks mode, make sure each page of the site includes a valid DOCTYPE.

Document type declaration is needed to know which type of document we’re creating. For example, this following document type declaration at the top of a web page indicates that the page should validate to HTML 4.01 Transitional ^[10]:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
http://www.w3.org/TR/html4/loose.dtd>
```

If we’re using HTML 5, the DOCTYPE is only:

```
<!DOCTYPE html>
```

- e. Use CSS reset rules because many browser engines have slightly different default values for things like line height and padding around elements. CSS reset rule makes it consistent across all browsers.
- f. Fix Internet Explorer issues by including IE-only markup so we can make additional HTML, CSS, or JavaScript to handle IE.
- g. Provide fallbacks for special features such as flash/image/video in case these features aren’t available on some browsers.

4.1.3. HTML 5 Compatibility

Table 4.1 Desktop Browser Compatibility ^[9]

| Feature | Mac | | | Windows | | | | | |
|---------------|--------|---------|--------|---------|-------------------|---|---------|-------|--------|
| | Safari | Firefox | Chrome | Safari | Internet Explorer | | Firefox | Opera | Chrome |
| | 5+ | 4+ | 15 | 5+ | 8 | 9 | 4+ | 11 | 15 |
| Canvas | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Audio | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Video | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Local storage | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

| | | | | | | | | | |
|---------------------|---|---|---|---|---|---|---|---|---|
| Session storage | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Post message | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Offline application | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ |
| Workers | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ |
| Query selector | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| WebSQL database | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ |
| IndexDB database | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ |
| Drag and drop | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ |
| Hash change (event) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| History management | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ |
| Web socket | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |
| Geo location | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Touch | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |

Table 4.2 Mobile Browser Compatibility ^[8]

| Feature | Safari on iOS | Android Browser | | BlackBerry Browser | | Nokia Browser | | Internet Explorer | Opera | | Firefox | webOS Browser |
|---|---------------|-----------------|---------|--------------------|---------|------------------|-----------|-------------------|-------------|------|---------|---------------|
| | iPhone, iPad | Phones | Tablets | Phones | Tablets | Meego - Nokia N9 | Symbian | Windows Phone | Mobile | Mini | Android | |
| | 3.2 | 1.5 | 3.0 | 5.0 | 1.0 | 1.2 | 3+ | 9 | 11 | 5 | 6 | |
| Application Cache | ✓ | ✓ (2.1+) | ✓ | ✓ (6.0+) | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ |
| Web storage | ✓ | ✓ (2.0+) | ✓ | ✓ (6.0+) | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ |
| Web SQL storage | ✓ | ✓ (2.0+) | ✓ | ✓ (6.0+) | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ |
| Geolocation | ✓ | ✓ (2.0+) | ✓ | ✓ (6.0+) | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ |
| Multimedia | ✓ | ✓ (2.3+) | ✓ | ✓ (7.0+) | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ |
| Server-sent events | ✓ (4.1+) | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ |
| Web sockets | ✓ (4.2+) | ✗ | ✗ | ✓ (6.1+) | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ |
| Web workers | ✓ (5.0+) | ✗ | ✗ | ✓ (6.0+) | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ |
| Canvas API | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ (Anna+) | ✓ | ✓ | ✓ | ✓ | ✓ |
| SVG | ✓ | ✓ (4.0+) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| Motion Sensors | ✓ (4.2) | ✓ (4.0+) | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| HTML 5 form virtual keyboard | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ (Anna+) | ✓ | ✓ | ✗ | ✓ | ✗ |
| HTML 5 form new controls | ✓ (5.0+) | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ |
| Touch (<i>only on touch screen devices</i>) | ✓ | ✓ (2.1+) | ✓ | ✓ (6.1+) | ✓ | ✓ | ✗ | ✗ | ✓ (android) | ✗ | ✓ | ✗ |

4.2. Mobile Web Application

A web application (web app) is actually a website in a mobile. What makes it different with other websites is it behaves as if a native application. Because it's generally the same as other websites, to build a web app we can use a browser-supported language such as JavaScript combined with HTML.

Web apps help users by making it easier to access the websites on their portable devices. Today, more people access the web via a mobile device than via a computer, and the discrepancy between them will grow more severe in the coming years as you can see in the chart below. Software or web developers take this into account in order to keep up this transformation ^[15].

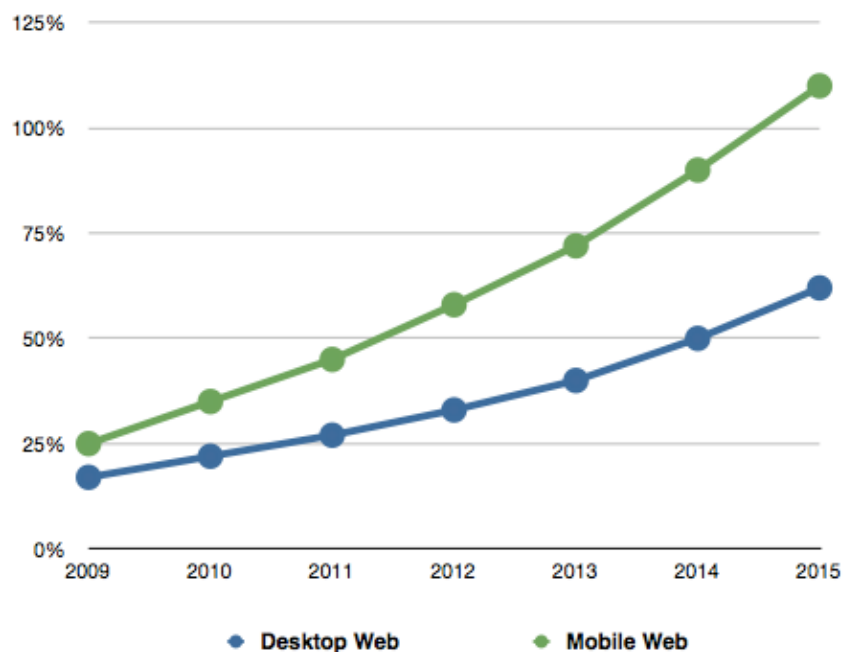


Figure 4.2 Predicted Growth of Mobile Web Access

A good starting point to make a mobile web app is to research the target audience. Sometimes it's fine to target one primary device, but in the other hand, we also need to think about multiple device support. One thing we need to keep in mind when choosing the target users, the more people we want to reach with our website or application, the more devices we will want to design for. The more

devices we design for, the more challenges we will create in design, development and in the device testing stages. Even if we plan to only support one platform, there is still a lot of uniqueness between devices, versions, carriers, etcetera ^[14].

After determine the target users of the website or web application, there are still several things that have to be considered when developing a mobile web app ^[13].

4.2.1. Fragmentation

Currently one of the biggest challenges in the mobile industry is device fragmentation. Device fragmentation refers to how the content will be displayed on different devices. There are 3 things that cause device fragmentation:

a. The hardware

This includes screen resolution, screen size, and many more technical features that vary from device to device.

b. The software

Each device will run different operating system as well as different browser version.

c. The rapid evolution of mobile

Mobiles that are up-to-date at the beginning of the year are quickly replaced by an upgraded version six months later. This causes software and hardware to fragment faster than anything we've seen in the desktop world.

4.2.2. Gesture

A gesture is defined as any movement that sends information or meaning by using our fingers. Usually we use a mouse on the desktop to interact and we now use our fingers for mobile. Most modern touch screen devices are now using gestures through various touch interfaces to allow users to access and navigate the content ^[12].

- Tap

Single tap an object, usually to open a file or navigate to a new view.

This gesture has the same function as mouse click in desktop.

- Press

Hold finger down on surface, this can be to view additional information or contextual menu relating to the object or screen

- Swipe

Swiping an image or object quickly, often to move to the next or previous object in a gallery or to scroll a long list of items

- Drag

To drag a finger across the screen without losing contact, usually to move an object from one location to another

- Pinch or spread

A pinching or spreading action of the fingers to change the zoom on an object. Pinch to zoom out and spread to zoom in.

To implement those gesture functions in a web app we can use a native class from the JavaScript called “TouchEvent Class Reference”. This class encapsulates information about a touch event. As fingers touch and move across the surface of the device, touched object will be sent.

Basically, there are 3 main function needed to manipulate the gestures event ^[16]:

- `touchstart`

Occurs when a finger touches the surface

- `touchmove`

Occurs when a finger moves across the surface

- `touchend`

Occurs when a finger lifts from the surface

`TouchEvent` object usually combined together with “`GestureEvent`” objects that are also sent during a multi-touch sequence. The `GestureEvent` class encapsulates information about a multi-touch gesture. It means that the `GestureEvent` objects are sent when there are two or more fingers interact with the surface of the device. This event contains scaling and rotation information (if it supported by the platform or device), if those functions are not supported by the platform or device, one gesture ends before another starts. The different types of `GestureEvent` objects that can occur are:

- `gesturestart`

Occurs when two or more fingers touch the surface

- `gesturechange`

Occurs when fingers are moved during a gesture

- `gestureend`

Occurs when the gesture ends (when there is only 1 or no finger touch the surface)

For example, for a two-finger gesture, the events occur as follow:

- a. `touchstart` for finger 1. Sent when the first finger touches the surface
- b. `gesturestart`. Sent when the second finger touches the surface
- c. `touchstart` for finger 2. Sent immediately after `gesturestart` when the second finger touches the surface.
- d. `gesturechange` for current gesture. Sent when both fingers move while still touching the surface.
- e. `gestureend`. Sent when the second finger lifts from the surface.
- f. `touchend` for finger 2. Sent immediately after `gestureend` when the second finger lifts from the surface.
- g. `touchend` for finger 1. Sent when the first finger lifts from the surface.

Objects returned from these events are called `TouchList` object. We know the number of `Touch` objects in this `TouchList` object. Every `Touch` object has these properties:

- `clientX`

The x-coordinate of the touch's location relative to the window's viewport.

- `clientY`

The y-coordinate of the touch's location relative to the window's viewport.

- `identifier`

The unique identifier for this touch object.

- `pageX`

The x-coordinate of the touch's location in page coordinates.

- `pageY`

The y-coordinate of the touch's location in page coordinates.

- `screenX`

The x-coordinate of the touch's location in screen coordinates.

- `screenY`

The y-coordinate of the touch's location in screen coordinates.

- `target`

The target of this touch.

It seems that every touch features is already provided by native JavaScript, but it turns out that we cannot fully use those features because some features are not compatible in all devices. For example, gesture events can be used in iOS only, that means that the event can be used for iPad/iPhone developer.

Every touched objects will be caught by our code in array form. iOS can catch up to 11 objects, whereas the earlier version of Android devices can detect only 1 object. This thing became a big issue in that period because the user couldn't apply some gesture interactions such as pinch/zoom or any other interaction that involves two or more fingers in Android devices. But the newer version of Android is now able to detect two or more fingers, but gesture event still doesn't work in Android ^[17].

4.2.4. How to Develop a HTML 5 Web Application [20]

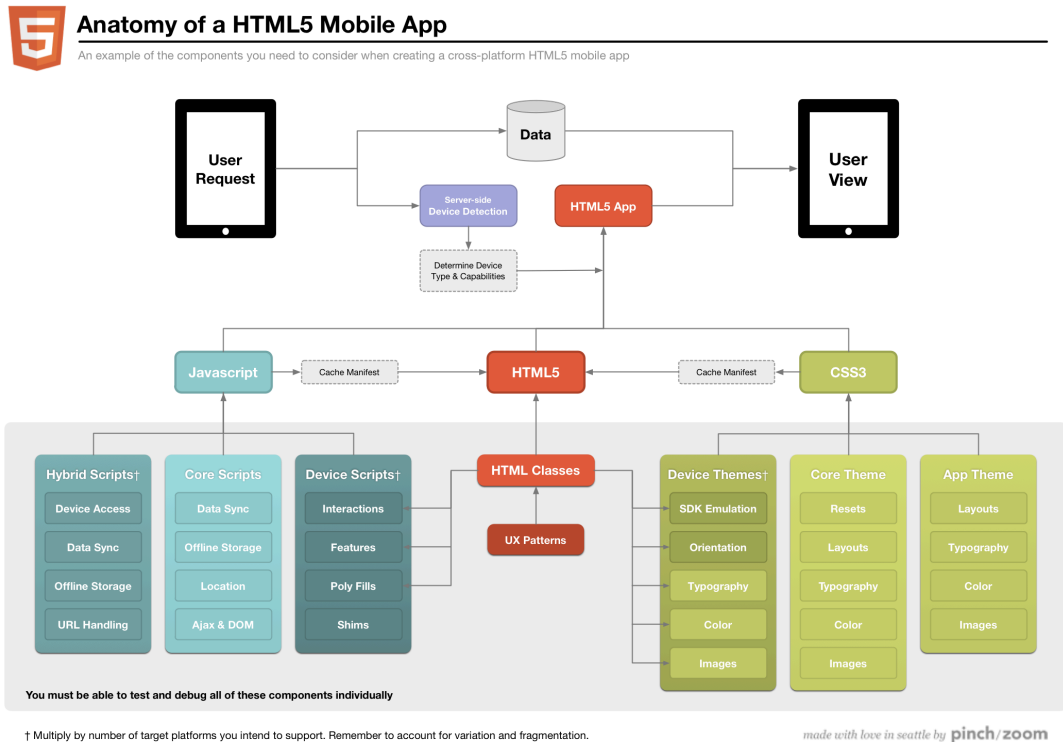


Figure 4.3 How to Make HTML 5 Mobile App Chart

The diagram above shows every part we need to create when developing HTML 5 mobile apps. There are so many parts to be created, this is why HTML 5 mobile apps are hard. Every part of that diagram will be explained below:

a. Setting up the server

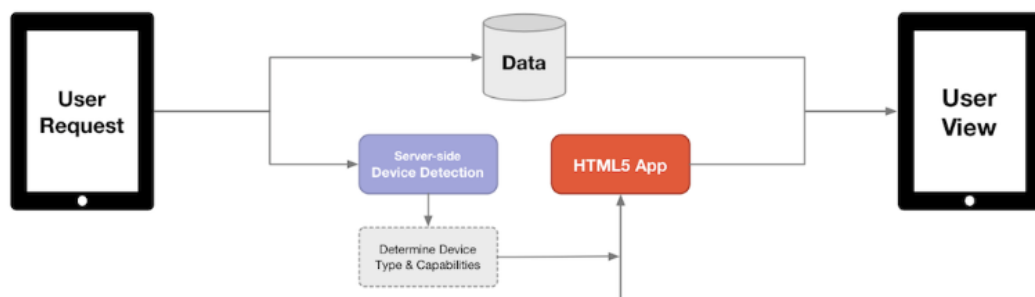


Figure 4.4 Set Up The Server Chart

This step starts with the user makes a request to view the content on a mobile device.

Since all devices are unique we need to have a server-side device detection to detect which device is requesting to view the content. To render the content into the app will need two things, the data and the HTML 5 app itself.

Most of all mobile apps launched when the device is offline, so we have to deal with the offline data experience as well. The issue is the data architecture usually designed to deliver pages over the Internet. Therefore we need to provide a fallback to anticipate the users data needs while the device is offline then synchronize everything together when the device is online.

b. Make the HTML 5 app

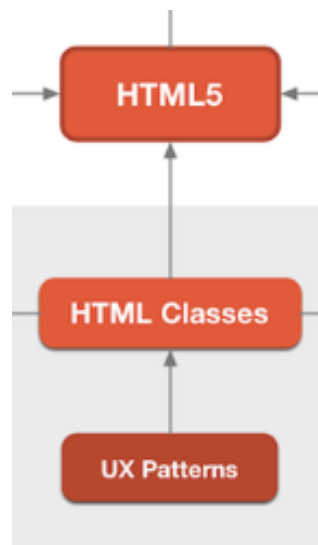


Figure 4.5 Make the HTML 5 App Chart

The next step is to create the HTML 5 app. Probably the best feature of HTML 5 that will be very useful in developing a mobile web app is the cache manifest (offline storage). This is a text file that lists out the files we wish to cache locally in the device memory such as the

common JavaScript, CSS and images files. Therefore if the user goes offline, the required interface elements are still available.

c. The JavaScript

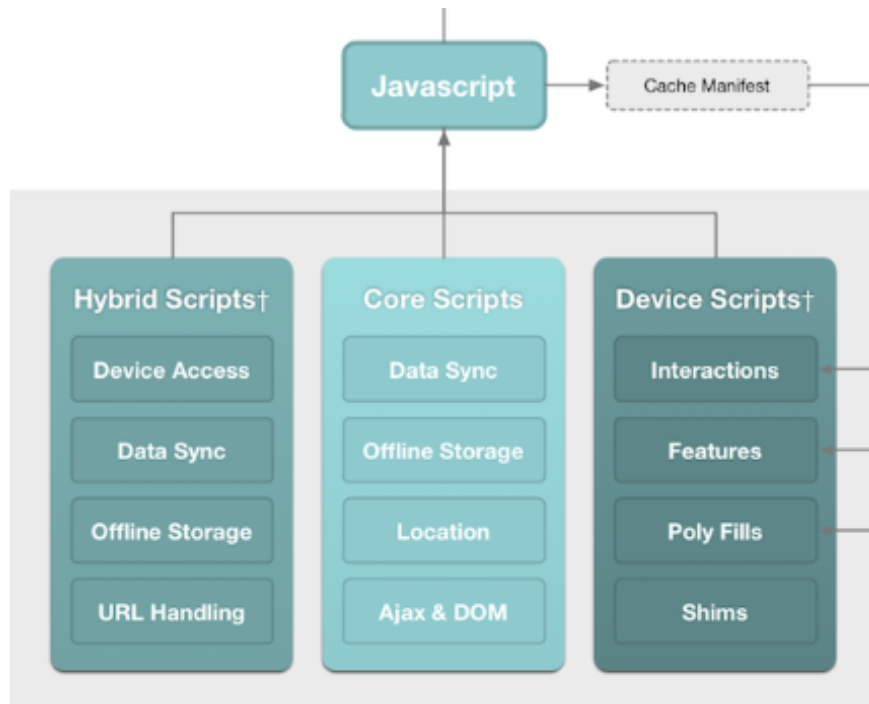


Figure 4.6 The JavaScript Chart

HTML 5 has many new features, but very few of them cannot be used if JavaScript is not applied. As we can see in the chart above, there are three parts of the JavaScript stack:

- Hybrid Scripts

These scripts are a bridge between the core scripts and the device SDK. We need a script to generalize script that works on multiple platforms.

- Core Scripts

These scripts are the common components of the app that will be available in every platform.

- Device Scripts

These scripts emulate native behaviors of the device.

d. The CSS

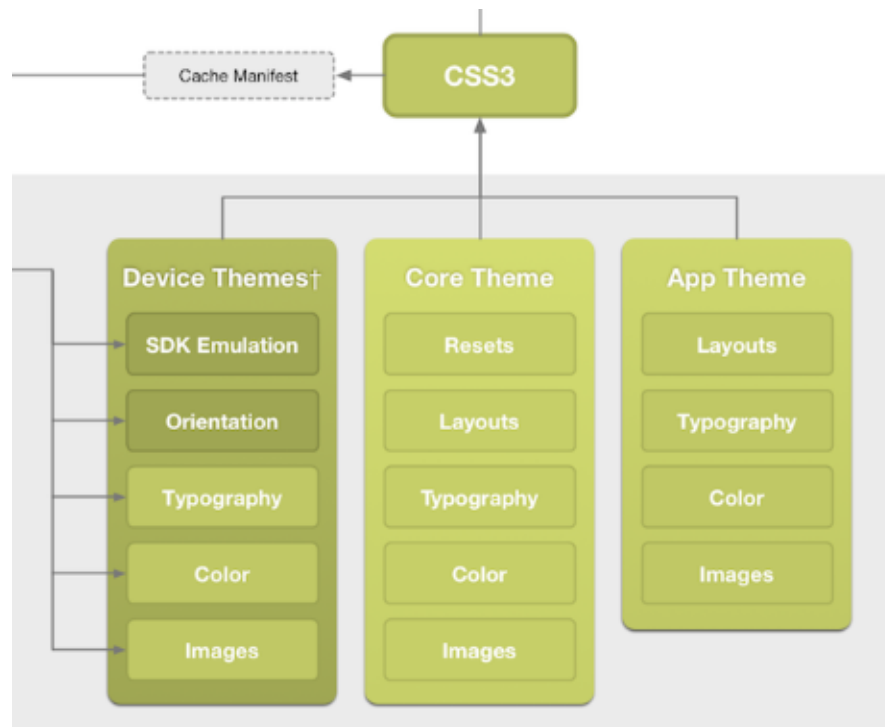


Figure 4.7 The CSS Chart

The CSS is the presentation layer of the app. When developing a mobile app, the CSS is split into three parts.

- Device Themes

These CSS is used to make the interface on the platform we've determined to develop.

- Core Themes

The core theme is the reusable theme of the application. These are the elements that should always be the same regardless of what platform we use. For example, the toolbar color that represents the brand color will be defined in the core theme.

- App Themes.

The app theme is the presentation elements that are specific to the app.

5. DEVELOPMENT METHODOLOGY

Development methodology is defined as the formal documentation for the phases of the system development life cycle. It defines the precise objectives for each phase and the results required from a phase before the next one can begin. It may include specialized forms for preparing the documentation describing each phase^[23].

For developing the project I will use the Iterative Application Development methodology (IAD). IAD is a method to develop system through repeated cycles. At each iteration, any modifications in the design or new functionalities will be added. The procedure consists of 3 steps, the initialization step, the iteration step, and the Project Control List^[21].

The initialization step creates a base version of the product. A project control list contains of all tasks that need to be performed. It is created to guide the iteration process. The control list is being revised constantly as a result of the analysis phase. The iteration itself involves the redesign and the analysis of the current version of the product.

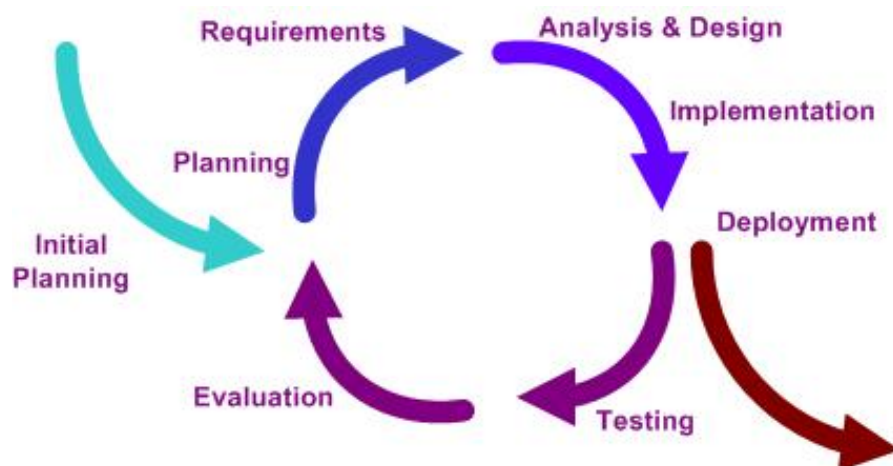


Figure 5.1 Iterative Application Development

Phase I: Introduction phase

- Activities
 - Identifies project scope
 - Do research regarding the project scope
 - Meeting with company tutor and project leader to gain more information about the project
 - Create a project plan
- Sub-deliverables
 - Project plan

Phase II: Definition phase

- Activities
 - Identifies the requirements (functional and non-functional requirements)
 - Create a risk management
- Sub-deliverables
 - User requirements document

Phase III: Design phase

- Activities
 - Create a draft design of the product
 - Preparing a test document (test plan and test report)
 - Make flowchart
- Sub-deliverables
 - Proof of concept
 - Test report

Phase IV: Realization phase

- Activities:
 - Implement the information gained from the research
 - Test the product of each iteration
 - Fix any bugs and errors

- Sub-deliverables:
 - Test report for each iteration

Phase V: Transfer phase

- Activities:
 - Deliver final product
 - Conduct final product testing
- Sub-deliverables:
 - Final test report
 - Final product mentioned in point 2.5

6. IMPLEMENTATION

6.1. The idea

6.1.1. General process at glance

In the initial situation, the plugin wraps each image with a `div` element with relative position to its parent, so that the image position can be centered. The next step is wrapping each `div` with `li` element and float it to the left. Then we come to the final step, in this step I show one `li` element in a display container. The illustration of those steps is shown below:

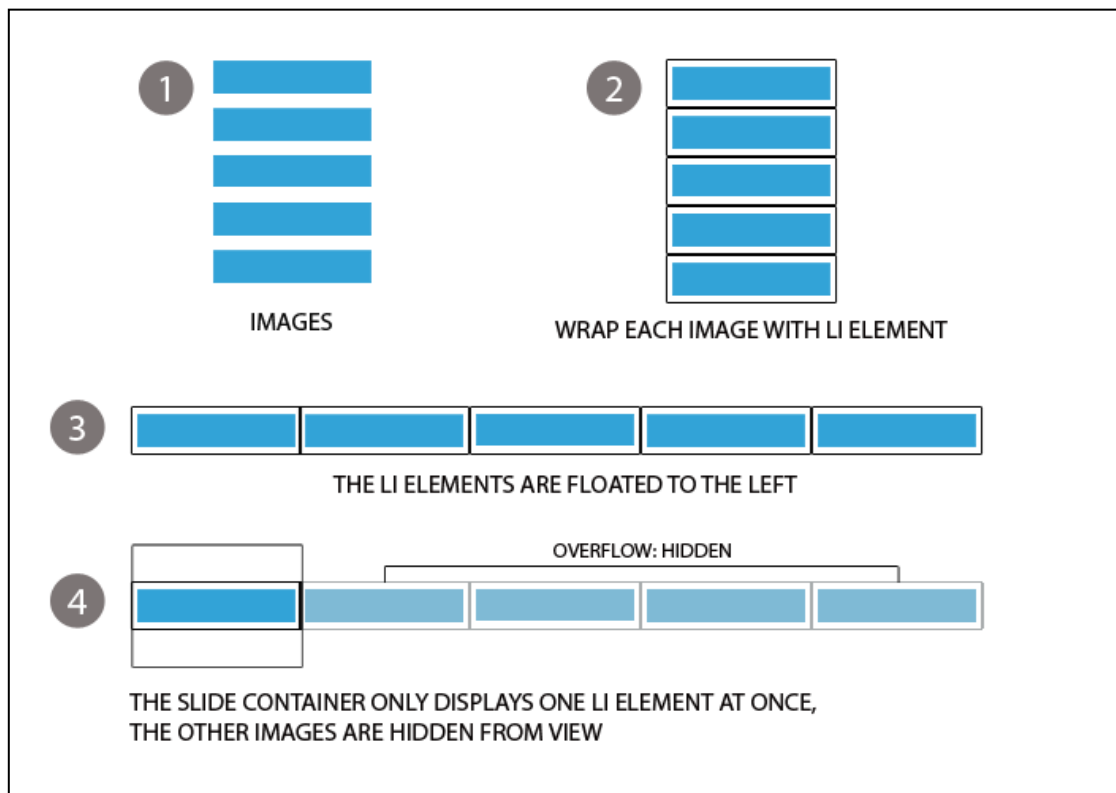


Figure 6.1 How the idea works

6.1.2. Element wrapping

The very first thing I do is wrap the element. I cannot expect every user builds a code as I want. The class name, the markup used, because every user has their own style to build the code. If I make the structure of the code is up to the

user, it will ruin the script for sure, therefore I have to wrap the user code with my own code, so the script goes well and the user doesn't need to change anything inside the script. In my script, I will interact with my own code, the classes I used are my own classes, which is given when the elements are wrapped. However, the user still has to make some basic code to be wrapped as shown below:

```

1      <div class="wrapper">
2          <ul>
3              <li> <!--WITH COMMENT-->
4                  
5                      <span>Lorem ipsum dolor sit amet, consectetur
6                          adipiscing elit.<br>Pellentesque molestie elit nec massa
7                          iaculis id tincidunt eros mattis.</span>
8                  </li>
9                      <!--WITHOUT COMMENT-->
10                 <li></li>
11                 <li>
12                     
13                         <span>Lorem ipsum dolor sit amet, consectetur
14                             adipiscing elit. Pellentesque molestie elit nec massa
15                             iaculis id tincidunt eros mattis.</span>
16                     </li>
17                 <li></li>
18                 <li>
19                     
20                         <span>Lorem ipsum dolor sit amet, consectetur
21                             adipiscing elit.<br>Pellentesque molestie elit nec massa
22                             iaculis id tincidunt eros mattis.</span>
23                     </li>
24                 </ul>
25             </div>

```

Figure 6.2 Basic HTML Code

When wrapping the element, the plugin first check is there any `` element inside the `` element, if the length of the image is greater than zero, which means there is at least one image inside, the wrapping process is started. While wrapping each image with div element, I also check if there is `` element inside the same `` with the image or not. If there is ``, it means that it is a comment for that image and the `` element will be wrapped using `<div>`. So with this plugin user can put comment for each image only by giving

 inside the . If there is no element, this process can be skipped.

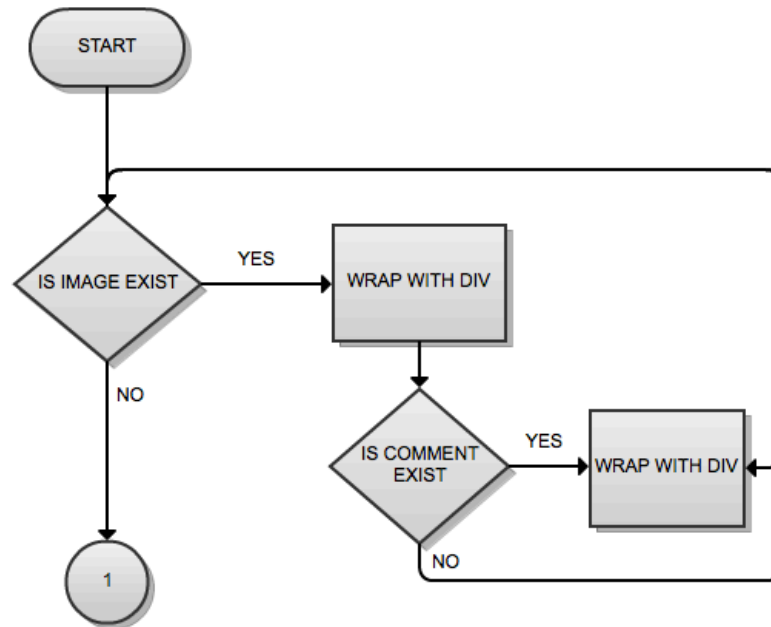


Figure 6.3 Element Wrapping Flowchart

6.1.3. Image size calculation

To set the image size I calculate the ratio of the image by dividing the width by the height. After I get the image ratio, the same calculation is applied to the container ratio. The next step is compare both values, if the image ratio is greater than container ratio, it means that the width of the image is closer to the width of the container (condition 1), and the other way around, if the image ratio is less than container ratio, the height of the image is closer to the height of the container (condition 2).

If condition 1 happens, the image width will be set equal to the width of the container. After we find the new width, we get back to the first ratio formula to recalculate the height. I already have the image ratio, so I only need to assign the new width to the formula to get the new height. I do the same in condition 2, only this time the image height is set the same as the container height and we recalculate the image width.



Figure 6.4 Image adjusting (**left**: by its width; **right**: by its height)

I do this way of calculation because I do not want any cropped image inside the container. Therefore, if the images are too high, I adjust its height equal with container height and the other way around as shown in the example above.

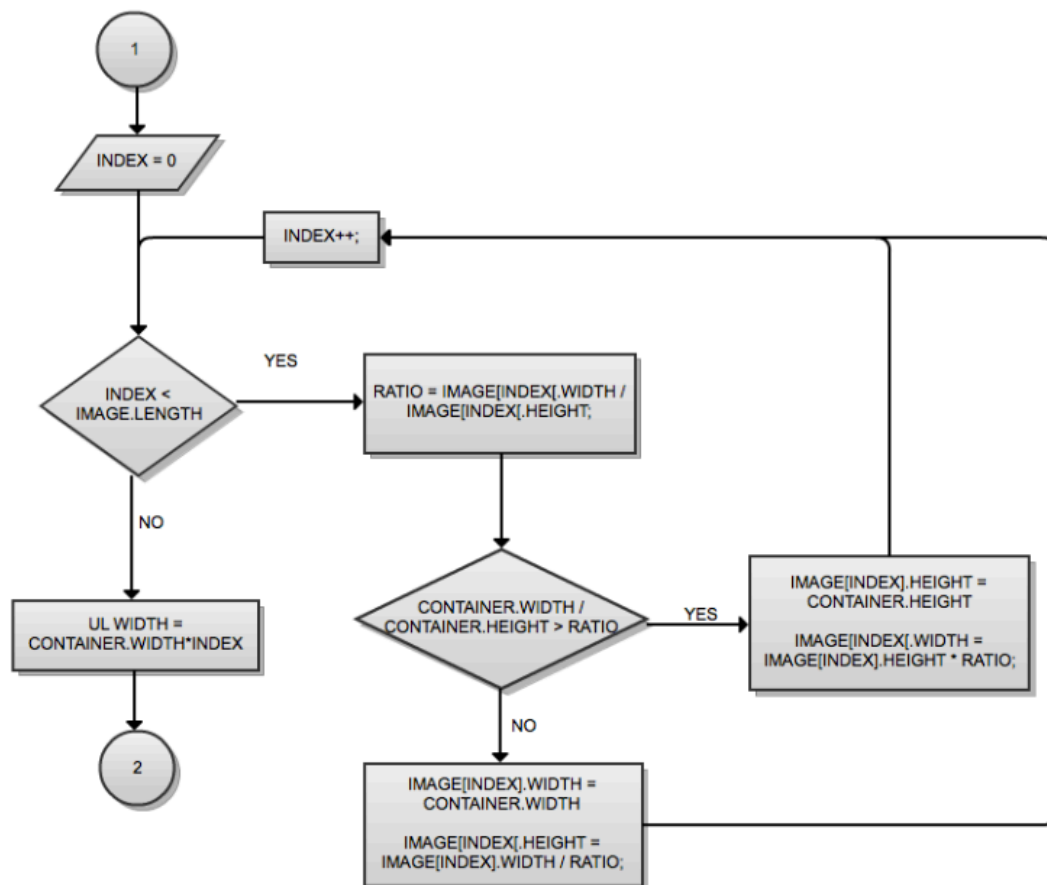


Figure 6.5 Image Size Calculation Flowchart

6.2. Optional features adding

This plugin offers several features, which is generally needed to have an interactive and user-friendly gallery. The developer may choose which features

they want to apply which features they do not. If the developer does not setup any option in their script, the value in every feature will be filled by the default value. To initialize the plugin user only needs to call the plugin constructor:

```

1 | $(document).ready(function() {
2 |     $('.wrapper').myGallery();
3 | });

```

Figure 6.6 Call the Plugin Constructor

By calling the constructor, the wrapping process is automatically run. The code shown above is the example for basic use and set all the features value by its default. As I said before, the value can be adjusted according to user need. User can do it as follows:

```

1 | $('.wrapper').myGallery({
2 |     navigation : true, // || false
3 |     pagination : true, // || false
4 |     slideshow : {
5 |         enable: true, // || false
6 |         firststate: 'start', // || stop
7 |         speed: 3000 // in milisecond
8 |     },
9 |     fullscreen : {
10 |         enable: true, // || false
11 |         imageSize: 'fill' // || stretch || fit
12 |     },
13 |     rotate : true, // || false
14 |     uploader : {
15 |         type: "none", // || file-browser || drag-and-drop ||
16 |         both
17 |         maxSize: 3, // maximum file size to be uploaded
18 |         targetPHP: "uploader.php"
19 |         // php file name, must be filled unless the type is set to
20 |         "none"
21 |     },
22 |     onWindowResize : {
23 |         enable: true, // || false
24 |         height: {
25 |             max: window.innerHeight,
26 |             // maximum height of the container when window is resized

```

```

23         min: 100
           // minimum height of the container when window is resized
24     },
25     width: {
           max: window.innerWidth,
26         // maximum width of the container when window is resized
           min: 100
27         // minimum width of the container when window is resized
28     }
29 },
30 preload : {
           enable: true, // || false
           targetPHP: "uploader.php"
           // php file name to resize image when loading, must be
           // filled unless the preload.enable is set to false
32     }
33 }
34 });

```

Figure 6.7 Call the Plugin Constructor and Set the Features

The value typed in bold is the default and the values next to it are the option user can take. Besides those values, the command will be ignored.

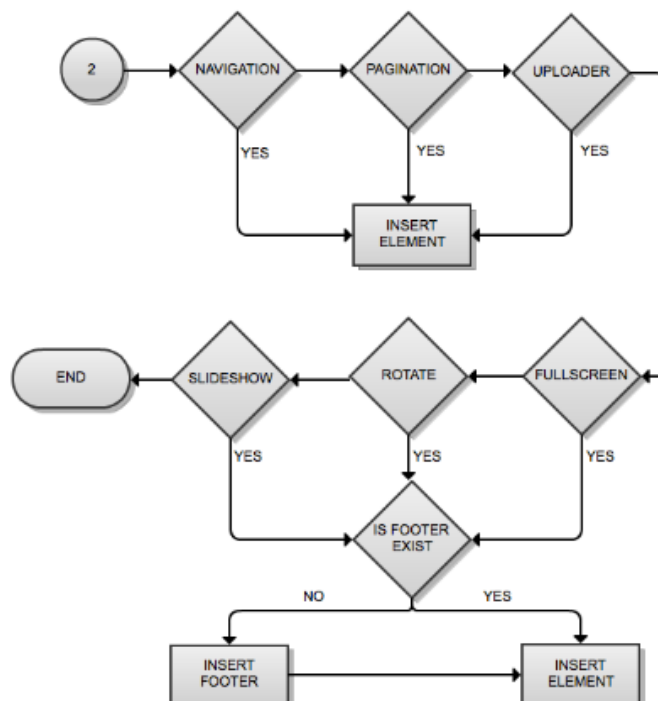


Figure 6.8 Optional Features Adding Flowchart

6.2.1. Image navigation

To navigate to another image, user can do it either by swiping (on mobile device) or by clicking the image and drag to the left or right (on the desktop). To implement these functions I use the same method for both desktop and mobile devices but with different event called. If the developer chooses to enable the navigation (next and previous button) or the pagination (button below the container) or even both, then those functions can also be used to navigate the images. But then again those are optional functions, the main function to navigate is by touching or clicking the image.

For swiping, I call `touchstart` event when user's finger touch the surface of the screen. I give a variable to save the boolean value to indicate that user is still touching the image. Then when user moves the finger over the image and the variable value is still true, event `touchmove` is called and ended when user lift the finger from the surface by binding `touchend` event. The value of the variable is set to false afterwards.

It is applied on the desktop as well. `mousedown` event is called when user is clicking the image, then check if user is still clicking the image or not. If the user is still clicking the image, the swiping function inside the `mousemove` event is applied, and will be ended when the mouse is released (`mouseup`) or stepped outside (`mouseout`) the container area.

Every touch event and mouse event will return its x and y values, so we know where the finger or the mouse current position is. In the `mousedown` or `touchstart` event, first we save the initial position of the finger or the mouse. After the first position is saved, when a finger or a mouse pointer is moving across the screen, we calculate the distance between the initial position and the current position. Now we get the distance, the next step is to move the image as far as the distance and make the image position follows our movement. See illustration below:

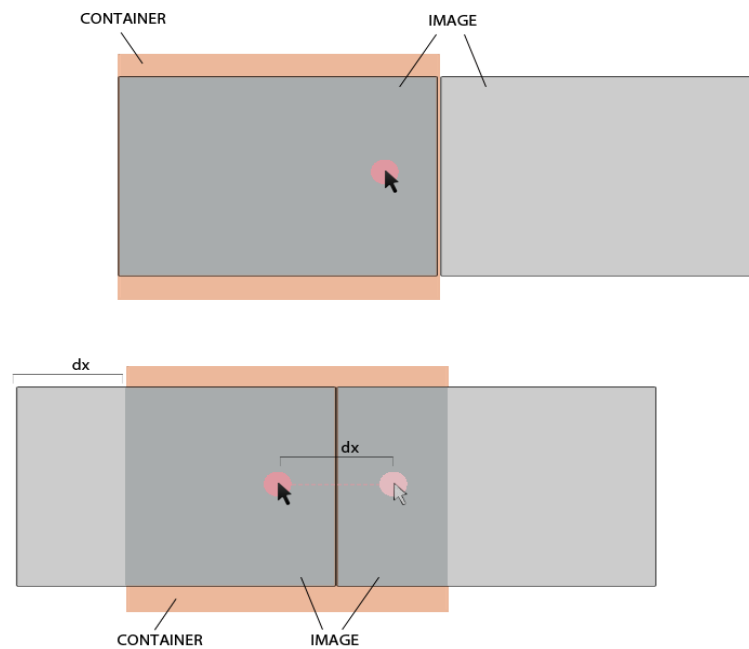


Figure 6.9 Image Navigation Illustration

This feature works well in the desktop and iOS devices, but when I tested in Android, the movement of the image and the response is so slow. The first time I encountered this issue, I thought that there are some mistakes in my plugin, but it turns out that Android devices cannot response the finger interaction quickly indeed. I tried to open another site that uses touches interaction as well as gestures interaction, the response was still so slow.

6.2.2. Image rotating and scaling

This is the other feature that supported by the plugin, image rotating. It is not merely rotate image by pressing a button (except if the developer chooses to add rotation button). If user is using mobile devices, they can rotate the image without pressing any button. The plugin reads user's gesture when two fingers start touching the screen. iOS devices have gesture event, which bind `gesturestart` when there are more than 1 finger on the surface of the screen, `gesturechange` when those fingers are moving across the surface, and `gestureend` when there is only 1 finger left on the screen. But since that event

only works in iOS devices, while this plugin is meant for other devices as well, instead of using that event I prefer to read the movement of user's finger manually.

From the research I did, Android can read up to 9 fingers, but the way it reads the object is different from iOS. In iOS, all fingers that touch the screen can be read at once, so when there are two fingers on the screen, the `touches.length` is two. `touches.length` returns the number of fingers touch the screen. Since it reads all the fingers at once, `touchstart` event is called once. While in Android, when there are two fingers touched the surface of the screen, it is considered as one finger touches the screen twice. Therefore the `touchstart` event is called twice, the number of its occurrence is the same as the number of the finger on the screen.

Every touched object has its ID, when the `touchstart` event occurs, I save every object ID and its position in an array, otherwise I cannot read more than one finger in Android. So when the array length is two, then it will be considered as gesture. When `touchend` event occurs, I destroy the array.

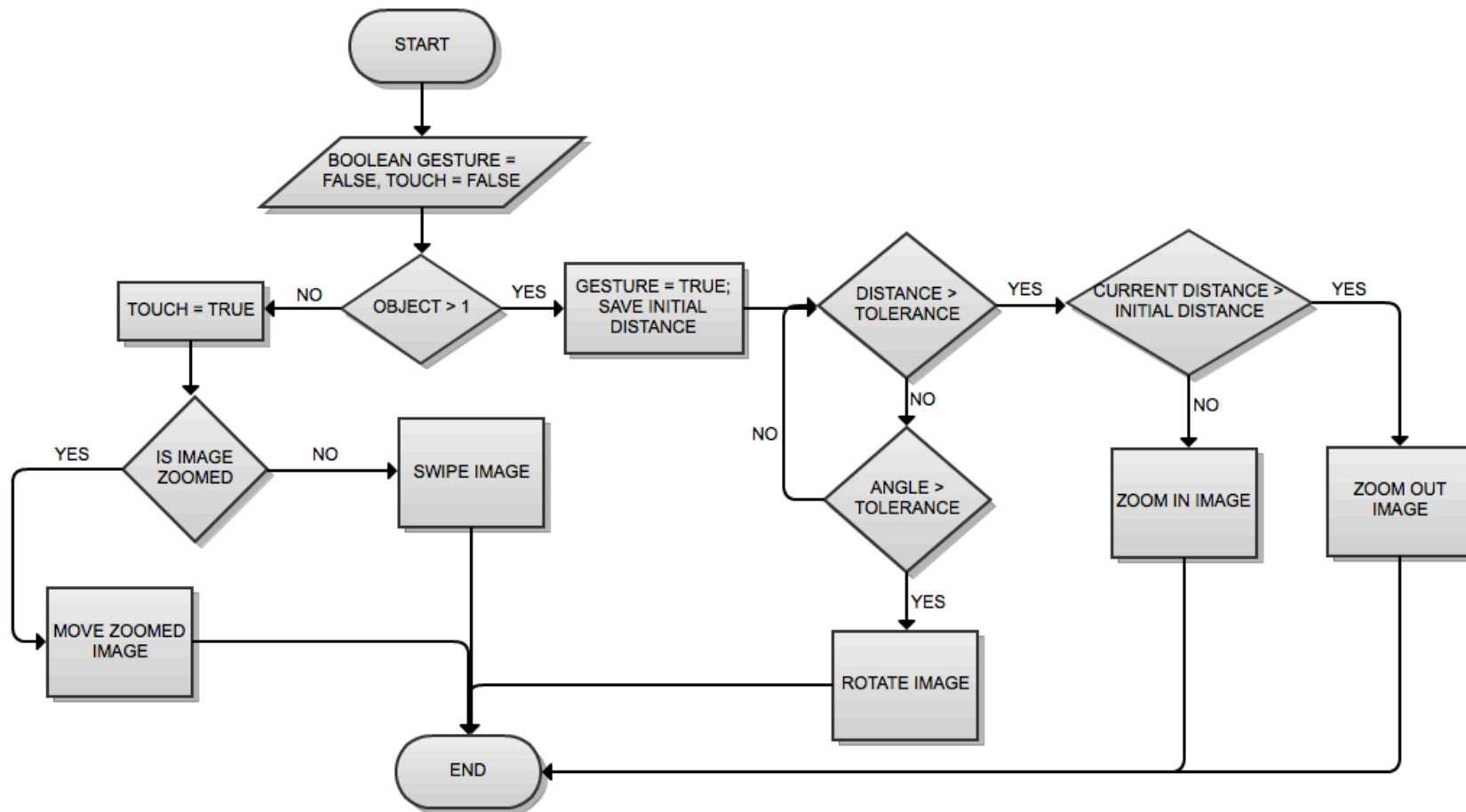


Figure 6.10 Touch Event Flowchart

The touch event can detect which object is touched and how many objects are being touched. So when `touchstart` is called, I get how many fingers are on the surface, if there is only one finger then I consider it as 'touch', if there are more I consider it as 'gesture'. Then inside the `touchmove` function it checks whether it is a 'touch' or a 'gesture'. If it is a 'touch' then the swiping process is started, if it is a 'gesture' then the rotating or scaling process is called.

The idea to detect whether user wants to rotate or scale the image is by calculating the angle between two fingers and the distance between them.

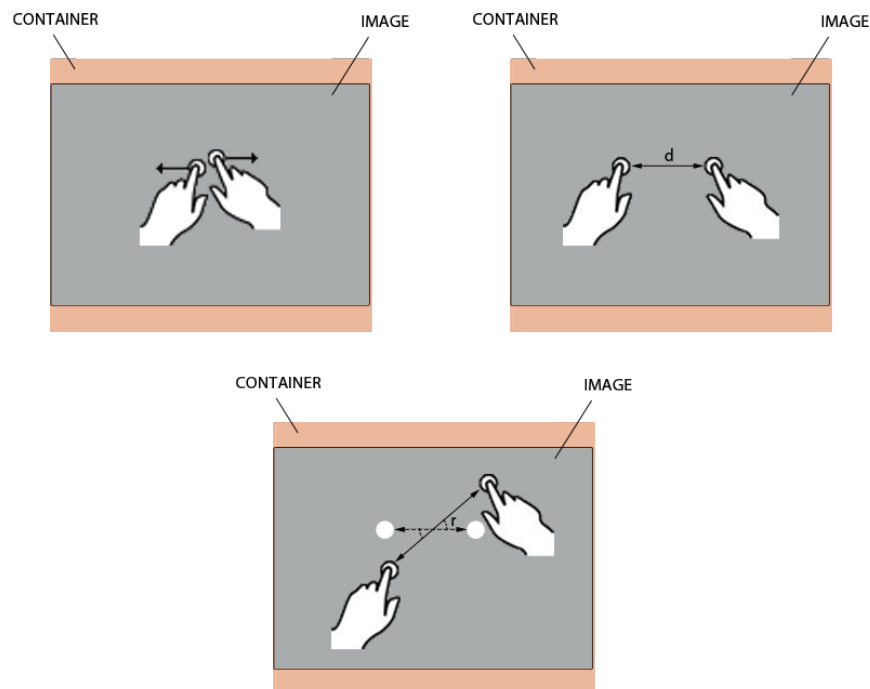


Figure 6.11 Gesture checking

First I give tolerance for each distance and angle. If the distance exceeds the tolerance value, then the gesture is considered as 'zoom' as well as angle, if the angle exceeds the tolerance, then the gesture is considered as 'rotate'.

6.2.4. Tilt the device

To enhance user-friendly experience, this plugin makes user easier to navigate the image. Explained above, user can either uses button or bullets or swipe/click the image to navigate. Besides, another way to navigate the image is to shake the device. Not really shaking the device, but just tilt the device to the right or left. If the device is tilted to the left the image will be navigate to the next image and the other way around.

I use `devicemotion` Event Listener to detect the movement of the device. That event gives the acceleration value of the movement. The values are divided into three parts: x, y, and z.

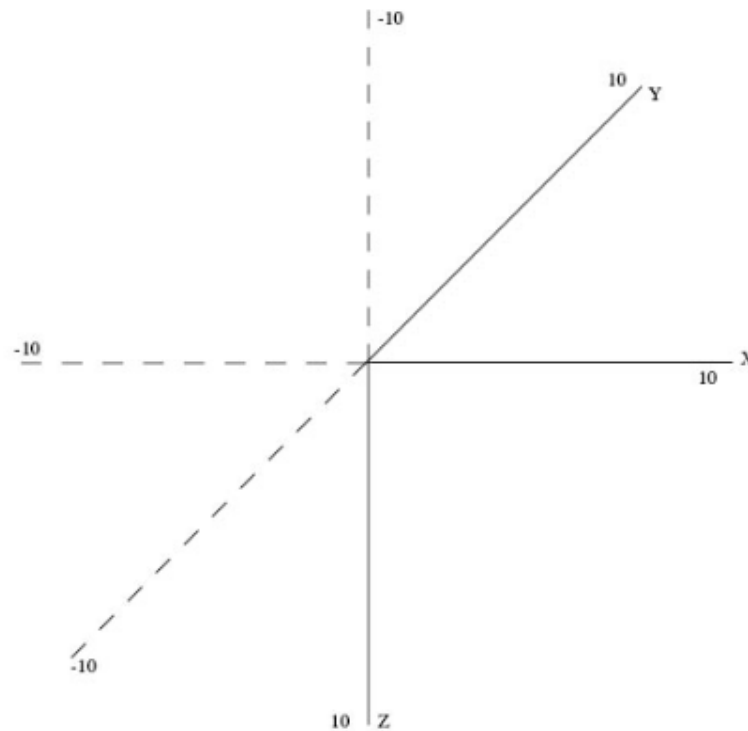


Figure 6.12 Motion Coordinate

As you can see in the chart below, the front screen is the main shaft. When the user holds it in vertical position, the values rendered are:

X: 0

Y: -10

Z: 0

If the screen is facing up the value of Z is -10, if the screen is facing down the value of Z is 10. That is how it works.

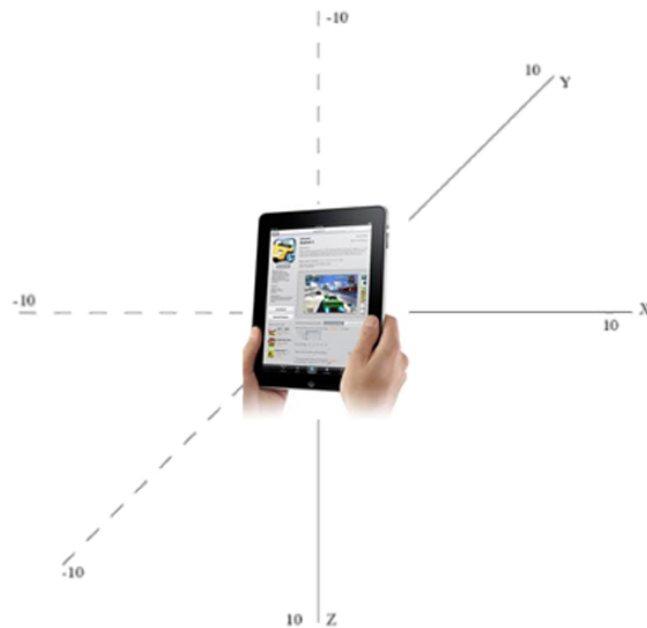


Figure 6.13 Motion Coordinate with iPad

6.2.5. Image manager

This feature is more useful and meant for the administrator of the website. What users can do with this feature is manage the images inside the slideshow. Users can either add image or remove image without modifying their code. The plugin provides two methods for uploading images, by browsing the file from the computer or simply by dragging the images and drop it into the dropping area.

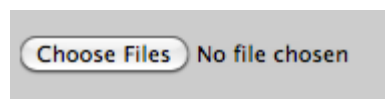


Figure 6.14 File browser

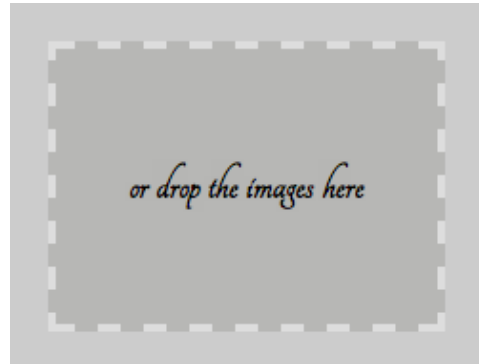


Figure 6.15 Images dropping area

The issue for this feature is several browsers and devices do not support File API. File API is API for representing file objects in web apps, as well as programmatically selecting them and accessing their data ^[22]. The fallback I give when the browser or device does not support File API is remove the uploader function when the web is accessed on that browser.

To generate a file browser button as shown above, HTML has its own standard `<input type="file" multiple>` element. The `multiple` attribute allows selecting several files or images at once. To handle the selected files, JavaScript returns those objects as a `FileList`. `FileList` objects can be obtained by calling `event.target.files` for file browser or `event.dataTransfer.files` for drag and drop method.

Once we get the file reference from `FileList` object, the next step is to capture the file information and to read its content into memory. This step can be done by initiating a `FileReader` object for every file.

```

1  var params = {};
2  params['file'] = name;
3  var queryString = obj2url(params,
4    setting.uploader.targetPHP);
5  xhr.open("POST", queryString, true);
6  xhr.setRequestHeader("X-Requested-With", "XMLHttpRequest");
7  xhr.setRequestHeader("X-File-Name",
8    encodeURIComponent(name));
9  xhr.setRequestHeader("Content-Type", "application/octet-
10 stream");
11 xhr.send(file);

```

Figure 6.16 Upload process

Each image that has been chosen is directly uploaded to the server and the slideshow is updated. Moreover, the images in the slideshow is shown as a thumbnail in the uploader box. If the users want to remove the image from the slideshow, the users can click on the selected images, then the thumbnail will be gone as well as the image. To upload the image I use `XMLHttpRequest`. The `XMLHttpRequest` object is used to exchange data within the server and client behind the scene. With this object, the web page can be updated without reloading the page, request data from server, receive data from server, and send data to server.

6.2.6. Server side image cropping and resizing

This feature is used in fullscreen mode. Its aim is to avoid stretched image in the slideshow. The users determine whether to use this feature or not, but I suggest to use it because it will be so ugly if we see 1024x768px image is stretched into 1440x900px. In normal mode I set the height 100% of the container, in fullscreen mode I do that on the other way around, I make the adjustment based on its width instead of its height. So the width is set to 100% of the viewport because I want the image fills the whole viewport. I keep the height complies its width.

For instance, the image original size is 1152x864px and the viewport size is 1440x701px. According to the viewport width, the new width of the image is set to 1440px. Then I calculate the ratio by dividing the new width with the original width. After get the ratio, I can get the height by multiplying the original height with the ratio. It works as follow:

| | |
|------------------------|------------------------------------|
| Original width: 1152px | Viewport width: 1440px = New width |
| Original height: 864px | Viewport height: 701px |

Ratio: new width / original width = $1440/1152 = 1.25$

The ratio means that the new width is 1.25 bigger than the original.

New height: original height * ratio = 864 * 1.25 = 1080px



Figure 6.17 Image original size (1152x864px)

Now the height has been calculated and we can see that the new height is bigger than the viewport height. I do not want to make the image height the same as the viewport height because if I do so, the image will be stretched.

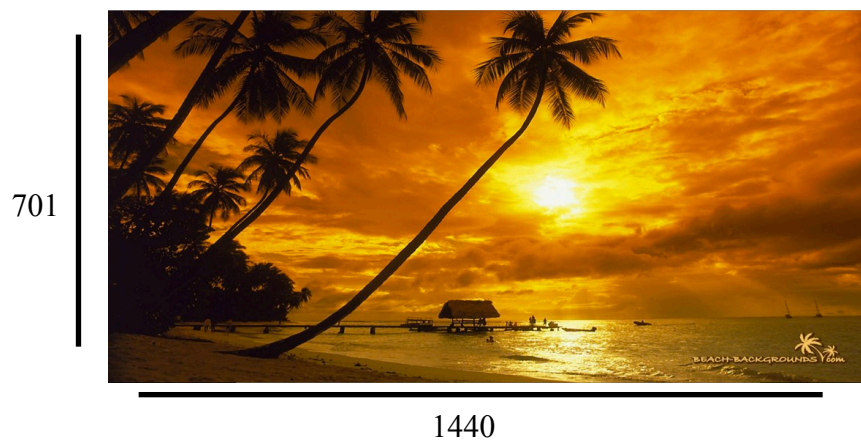


Figure 6.18 Image is stretched

To avoid that bad output, I give this feature to crop the image in the center. I remove several pixels on the top and bottom of the image to get its center.



Figure 6.19 Image cropping process

6.2.7. Image preloading

Another feature from this plugin is image preloading. If the value of this feature is set to true, then when the images is loading, user will not see a blank image container without any images or see a container with half image loaded. For preloading the image, I use image-resizing function. When the plugin is called, it hides all the original images then sends a request to PHP file to resize all the images into smaller size, after get the result, the original images are cloned but I change the source with the new resized images. By using this way, user will see blurry images in the container instead of blank container before the original images are loaded. If the images are already well loaded, the small sized images are removed and the original ones are shown.

CONCLUSION AND RECOMMENDATION

After finishing the project I can draw several conclusions about it. Even though the devices have its native touch feature, it does not mean that it can be implemented without doing or giving any script in the application, in this case, website or web applications. The way to respond the command given by the script is also different for every platform. Therefore when developing a web app, this is one of the most important thing to be considered. The more devices in the scope, the more effort has to be put in it because we have to provide any fallback for every possibility. Nevertheless, for every problem encountered during the development process always finds its solution eventually. For instance, when building the proof of concept, I found that the gesture function could not be implemented in Android devices. But through several researches and a little bit more coercion against the brain, the problem could be solved.

In my experience, I would like to recommend for a better time management and preparation for the project. Start earlier if it is possible and make sure the concept and topic of the project since the beginning of internship period, otherwise it will consume a lot of effort while doing the project in a short time. Before starting the internship, better do some research about the project first in the spare time, it will help a lot because when you start doing the project and you already have enough knowledge about it, it will be much faster and smoother though there must be several obstacles happen.

EXPERIENCE DURING GRADUATION PROJECT

I gained countless experiences during my internship period. I met the client, did a real project for them, those were really terrific. In the beginning, I was given a training program for one month and it taught me a lot. When I started my internship, there are two more new trainees who just started as well. We did the training program together, did presentation, brainstorming, etcetera. After finish the training program, we were given two different projects from ILUMY. I no longer worked with them since then and worked with different team instead. The good thing was one member of the team is Indonesian, it made me easier to ask and to discuss a lot of things, he helped me a lot.

At first I was given a topic for my graduation project, but it was really broad. The first topic is to make a web app using HTML5, which works in multiple platforms. In fact, there was no project corresponding to the topic at that time, therefore my tutor and my head of development decided to give me another assignment. After two months waiting for the assignment, finally in the beginning of April, I got my topic. I was quite anxious at that time because I had really short time to finish the project, on the other hand I have to do a project for the client as well and write a thesis report. I talked to Mr. Ad Maas about it, he supported me and it made me even more confident to do the project.

Actually I split the time for doing my project and for doing the client project, before lunch break and after lunch break. But it turned out the project for the client took a lot of time and my project either, therefore I have to spend more time after work hours to struggle with my project. I went home above 19.00 almost everyday, sometimes at 21.00 and continue doing it at home. Finally with hard work and helps from others, I could finish my project in time and it makes me quite proud of myself for the efforts I did.

Overall, though it took a lot of energy and time, I enjoy every single minute of the process I went through, because I got a project which is related to my hobby and interest, I was surrounded by kind people, I was supervised with two great tutors, I am really grateful of those things.

BIBLIOGRAPHY

- [1] **Element definition:** http://www.w3schools.com/html/html_elements.asp
- [2] **Attribute definition:**
http://www.w3schools.com/html/html_attributes.asp
- [3] **Tag definition:**
issemym.vector.com.mx/webspeed100B/doc/wshelp/glossary.html
- [4] **Syntax definition:**
[en.wikipedia.org/wiki/Syntax_\(programming_languages\)](http://en.wikipedia.org/wiki/Syntax_(programming_languages))
- [5] **Markup definition:** <http://www.answers.com/topic/markup-language>
- [6] **Quirks mode definition:** http://en.wikipedia.org/wiki/Quirks_mode
- [7] **Fallback definition:** <http://www.thefreedictionary.com/fallback>
- [8] **Mobile browser compatibility:** <http://mobilehtml5.org/>
- [9] **Desktop browser compatibility:**
<http://findmebyip.com/litmus/>
<http://html5readiness.com/>
- [10] **DOCTYPE declaration:**
<http://www.elated.com/articles/xhtml-document-types/>
- [11] **Build cross-browser HTML website:**
<http://www.elated.com/articles/cross-browser-website-tips/>
- [12] **Gesture:** <http://themobilebits.com/articles/5>
- [13] **Things need to know when developing a web app:**
<http://themobilebits.com/articles/4>
- [14] **How to start building a web app:**
<http://themobilebits.com/articles/2>
- [15] **Mobile apps:**
http://mobiledesign.org/size_and_scope_of_the_mobile_market
http://mobiledesign.org/why_mobile
- [16] **Touch and gesture events:**
<https://developer.apple.com/library/safari/#documentation/UserExperience/Reference/TouchEventClassReference/TouchEvent/TouchEvent.html>

- [17] **Android and iOS touch event compatibility:**
<http://backtothecode.blogspot.com/2009/10/javascript-touch-and-gesture-events.html>
- [18] **HTML 5 features enhancement:**
<http://paperkilledrock.com/2010/05/html5-localstorage-part-one/>
<http://themobilebits.com/articles/3>
- [19] **The reason HTML5 is better than the old HTML:**
<http://www.html5code.com/tutorials/introducing-html5/>
<http://www.webkit.org/blog/1273/the-html5-parsing-algorithm/>
- [20] **Step per step how to develop a web app:**
<http://pinchzoom.com/posts/anatomy-of-a-html5-mobile-app/anatomy-of-a-html5-mobile-app>
- [21] **Development methodology:**
http://en.wikipedia.org/wiki/Iterative_and_incremental_development
- [22] **File API:** <http://www.w3.org/TR/FileAPI/>
- [23] **Development methodology:**
<http://encyclopedia2.thefreedictionary.com/system+development+methodology>

APPENDIX A: PROJECT PLAN

Project Plan

Cross-Platform Image Gallery

JavaScript Plugin



Date : 3 April 2012

Author : Daniar Gita Nindiary Suprianto

Table of Contents

| | |
|---|----------|
| 1. Introduction..... | 1 |
| 2. Project Statement..... | 2 |
| 2.1. Formal Client..... | 2 |
| 2.2. Project Leader..... | 2 |
| 2.3. Current Situation | 2 |
| 2.4. Project Justification..... | 2 |
| 2.5. Project Product..... | 2 |
| 2.6. Project Deliverables and Non-Deliverables..... | 3 |
| 2.7. Project Constraints..... | 3 |
| 2.8. Project Risks | 3 |
| 3. Project Phasing | 5 |
| 4. Project Management Plan | 7 |
| 4.1. Money..... | 7 |
| 4.2. Skills | 7 |
| 4.3. Quality | 7 |
| 4.4. Information..... | 7 |
| 4.5. Time..... | 8 |
| 4.6. Organization | 8 |

1. Introduction

The aim of the project is to develop an interactive image gallery JavaScript plugin by implementing touch/gestures interaction and sensor on different devices such as (iPad/iPhone/Android/Windows Phone) in order to make the website becomes more user friendly. It's also about screen behavior like resolution and orientation. For instance, when the device is being rotated, the image is also rotated or we can rotate image in the web app by moving our fingers. When the screen is pinched, the image will be zoomed. When we swap the image, the image will move and so on.

This document is made to define the project and collect all information about it to manage it clearly. It will be used to share information and planning with the client.

2. Project Statement

2.1. Formal Client

ILUMY B.V.

Address : Waterlooplein 125, 1011PG
Amsterdam, Netherlands

Representative : Stefan Hoevenaar

Phone : +31 20 774033

Contact email : stefan.hoevenaar@ilumy.com

Website : <http://ilumy.com>

2.2. Project Leader

David Clarisse

Company : ILUMY B.V.

Phone : +31 20 774033

Contact email : david@ilumy.com

2.3. Current Situation

ILUMY has been doing many web apps, but they emphasize their web apps compatibility to iOS devices such as iPhone or iPad. Moreover, the touch/gestures functions that they implement are only some basic features such as tap and swipe.

2.4. Project Justification

Client wants some others functions that involve gesture interactions to be implemented on their web apps and the most important thing is to make those functions work on different devices (cross-platform).

2.5. Project Product

Interactive image slideshow gallery.

Features:

- Full screen

- Multi resolution
- Image size adjustment
- Image manager
- Enable gesture interactions (tap, swipe, rotate, pinch/zoom)

2.6. Project Deliverables and Non-Deliverables

Project deliverables:

- PHP, and JavaScript files
- Project plan
- Requirement document
- Product documentation
- Proof of concept
- Test report

Non-deliverables:

- User manual

2.7. Project Constraints

This project will be developed in PHP Storm using JavaScript, PHP, and HTML5 programming languages. To create a JavaScript, I will use a JavaScript library called jQuery in order to simplify my script and make it more understandable. For the testing purpose, I will use iPhone, iPad, Android tablet, and also Windows Phone emulator from Microsoft.

2.8. Project Risks

Table 2.1 Risk Management

| Risk | Indication | Probability (1-5) | Mitigation |
|-------------------------|------------|----------------------|---|
| Data loss | High | 2 | Use sub-version so the data is saved in the server after commit the code. |
| Testing devices are not | Medium | 4 | Use simulator, but it will not |

| | | | |
|---|--------|---|---|
| available | | | as precise as the real device. |
| Not enough knowledge to make the project | High | 2 | Do more research and ask the expert if there's already no solution. |
| Computer crash | Medium | 2 | Back up the files |
| Misunderstanding or changes in the requirements | Medium | 3 | Meet the client and discuss until everything is clear |
| Run out of time due to bad planning | High | 3 | Stick to the schedule |

3. Project Phasing

Phase I: Introduction phase

- Activities
 - Identifies project scope
 - Do research regarding the project scope
 - Meeting with company tutor and project leader to gain more information about the project
 - Create a project plan
- Sub-deliverables
 - Project plan

Phase II: Definition phase

- Activities
 - Identifies the requirements (functional and non-functional requirements)
 - Create a risk management
 - Create a draft design of the product
- Sub-deliverables
 - User requirements document

Phase III: Design phase

- Activities
 - Preparing a test document (test plan and test report)
- Sub-deliverables
 - Test report

Phase IV: Realization phase

- Activities:
 - Implement the information gained from the research
 - Test the product of each iteration
 - Fix any bugs and errors

- Sub-deliverables:
 - Test report for each iteration

Phase V: Transfer phase

- Activities:
 - Deliver final product
 - Conduct final product testing
- Sub-deliverables:
 - Final test report
 - Final product mentioned in point 2.5

4. Project Management Plan

4.1. Money

8 hours x 45 days = 360 hours of wage

4.2. Skills

To realize this project, I should have the planning skill in order to prevent any timing problems during the execution of the project. Since I will build a product, I should have the necessary technical skills such as PHP, HTML5, and JavaScript programming language and be able to make researches according the needs of the programming phase.

4.3. Quality

For quality assurance I will use the document control to keep track the progress and the changes of the product. By using that document, every changes and approval by the client will be recorded. At the end of development process, I will conduct a meeting with the client to review if the desired result is achieved.

4.4. Information

Table 4.1 Information

| | Project plan | User requirements document | Deliverable |
|----------------|--------------|----------------------------|-------------|
| Client | A, R | Di, A, R | R, Ar, A |
| Tutor | Di | Di | Di |
| Project leader | Di | Di | Di |
| Developer | Di, Dr, Ar | Dr, Di, Ar | Dr, Di |

Legenda:

| | | | |
|----|---------|----|---------|
| Dr | Draw up | S | Send |
| Di | Discuss | R | Receive |
| A | Approve | Ar | Archive |

4.5. Time

Table 4.2 Time

| Phase | Begin Date | Duration | Deliverables |
|----------------|---------------|----------|--|
| 1. Initiation | 3 April 2012 | 1 week | Project plan |
| 2. Definition | 10 April 2012 | 3 days | Requirement document |
| 3. Design | 14 April 2012 | 2 days | Test plan |
| 4. Realization | 17 April 2012 | 3 weeks | 1 st iteration of the product |
| 5. Testing | 9 May 2012 | 3 days | 1 st test report |
| 6. Realization | 16 May 2012 | 4 weeks | 2 nd iteration of the product |
| 7. Transfer | June 2012 | - | Final product, Test report, Project plan, Requirement document |

4.6. Organization

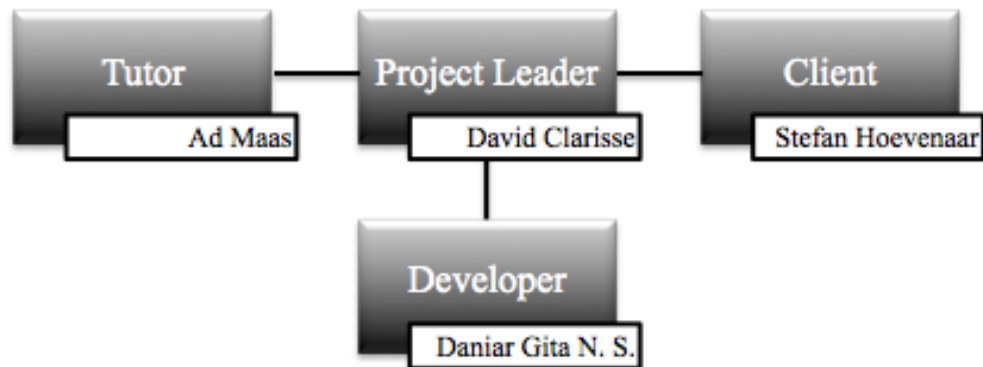


Figure 4.1 Organizational Chart

APPENDIX B: REQUIREMENT DOCUMENT

Requirement Document

Cross-Platform Image Gallery

JavaScript Plugin



Date : 3 April 2012

Author : Daniar Gita Nindiary Suprianto

1. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the image gallery. It will explain the purpose and features of the product. It also defines any non-functional requirements, constraints and what deliverables are to be supplied. Once approved, this document will provide a foundation for the development.

1.2. Scope

This image gallery will be used in mobile devices, particularly iOS devices such as iPhone and iPad, Android tablet, and Windows Phone. Any touch, gestures and sensor that will be applied in this product are tap, swipe, rotate, pinch/zoom, and device orientation.

2. Overall Description

2.1. Product Perspective

The project is an interactive image gallery JavaScript plugin with the goal of creating a better user experience in mobile app. The plugin itself will be developed using the help of JavaScript library named jQuery.

2.2. Product Functions

The image gallery plugin will be able to:

- Resize the image, fit to the size of the viewport
- Recognize finger gestures
- Recognize device orientation
- Work on different platforms
- Upload images

2.3. Design and Implementation Constraints

For the development of this product, programming language HTML5, PHP, and JavaScript (jQuery) will be used.

The image gallery should be able to run in iPad, iPhone, Android tablet, and Windows Phone.

3. Requirements

3.1. Functional Requirements

1. Full screen
2. Client side image resizing and cropping. If the image is too big, it won't be stretched, it will be cropped and the image size will be adjusted instead.
3. Multi platform and multi resolution
4. Image manager to upload a bunch of images
5. Preload image
6. Gestures and sensor recognition:
 - a. Tap
 - b. Swipe
 - c. Rotate
 - d. Pinch/zoom
 - e. Device orientation

3.2. Other Non-Functional Requirements

3.2.1. Performance Requirements

Performance of the product should be adequate to provide effective and responsive slideshow. Response time toward finger gestures should be under a second.

3.2.2. User Documentation

- Project plan
- Requirement document
- Documentation
- Proof of concept
- Test report

APPENDIX C: PROOF OF CONCEPT

Proof of Concept

Cross-Platform Image Gallery JavaScript Plugin



Date : 9 April 2012

Author : Daniar Gita Nindiary Suprianto

1. Introduction

In software development, a prototype is a working model of a product, which is built for demonstration purposes or as part of the development process ^[1]. This document is made as a base for developing the whole product. Before start developing, the developers have to make sure that their concept is possible to be built. By making the proof of concept, we will know which function can work well, which can't. If it doesn't work, we can choose several alternative ways. Without this document, if it turns out that the concept is not possible to do, it will trouble us eventually because we don't prepare any other alternative concepts.

2. Prototype

This is a mock-up product that I made to proof that my project is possible to do. I only made the prototype of 3 main functions of my projects, they are slide, rotate, and zoom. Besides, I also test the touch and gesture functions, which are the main issues here.

I tested this prototype on 4 different frameworks; they are iOS (iPhone and iPad), Android tablet, Windows Phone, and some different browsers on desktop. For Windows Phone I use emulator from Microsoft since I and my friends don't have any Windows Phone. So the accuracy for this platform is not 100% perfect.



Figure 2.1 Prototype in desktop (Chrome, Safari, Firefox, and Opera)

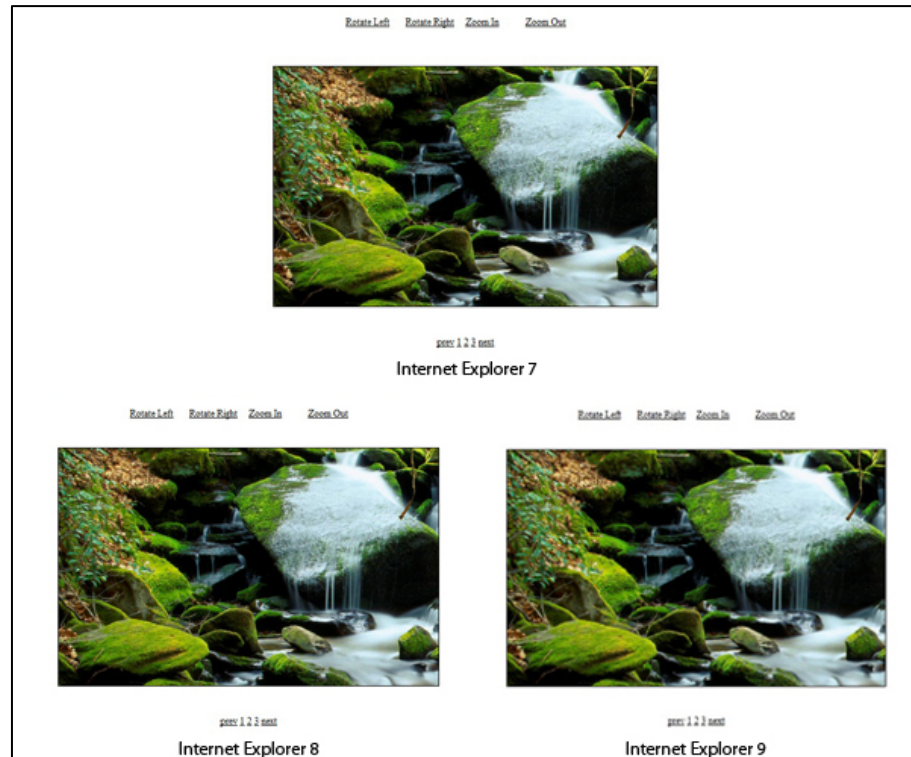


Figure 2.2 Prototype in desktop (Internet Explorer 7, 8, 9)

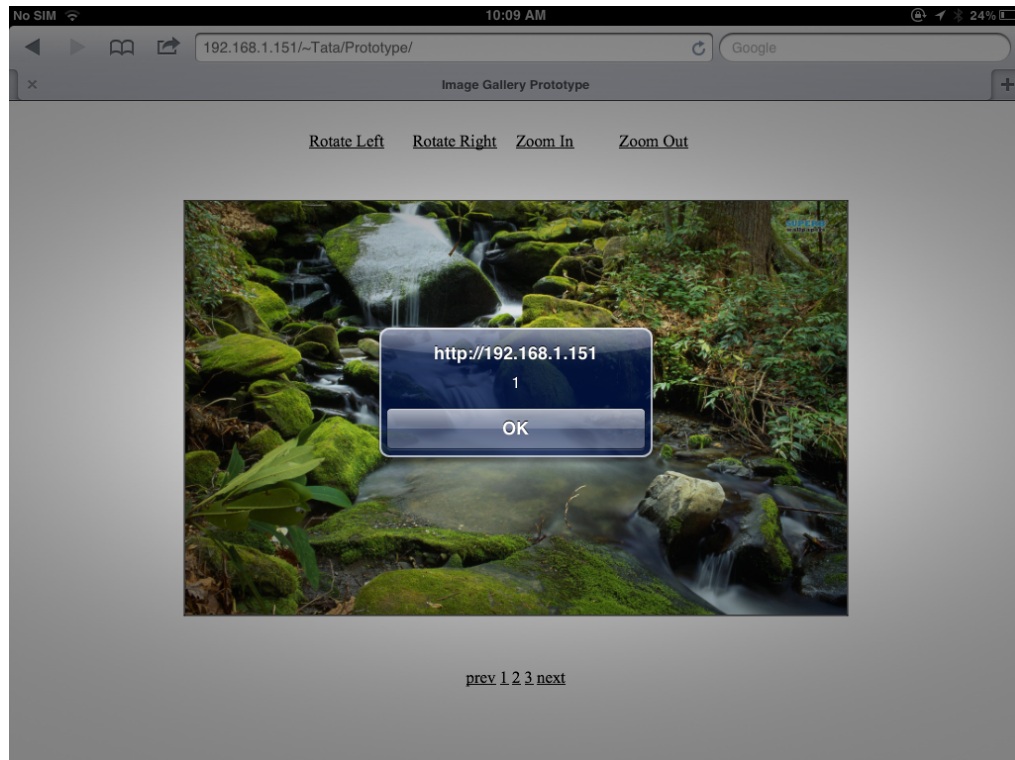


Figure 2.3 Prototype in iPad shows the amount of touched object

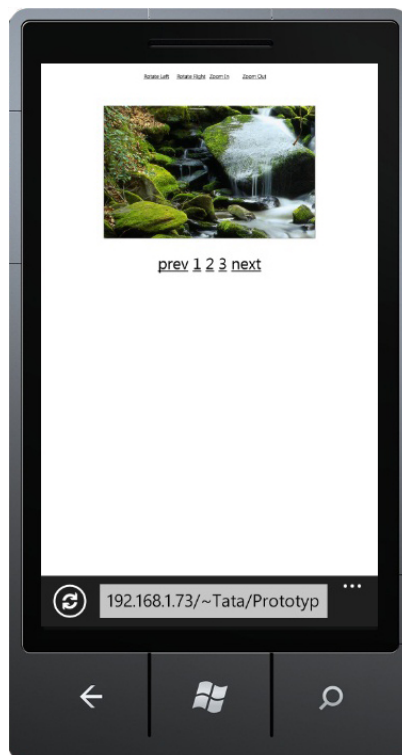


Figure 2.4 Prototype in Windows Phone emulator



Figure 2.5 Prototype in Android Tablet

3. Result

3.1. Image Transformation

Table 3.1 Image transformation testing result

| Devices | Browsers | Image Translation ¹ | Image Rotation | Image Scaling ² | Image Load |
|---------------|------------------------|--------------------------------|-------------------|----------------------------|------------|
| Android | Browser (WebKit based) | Not smooth (slow) | Not smooth (slow) | Not smooth | Slow |
| | Firefox | Not smooth (slow) | Not smooth (slow) | | Slow |
| | Opera Mobile | Not smooth (slow) | Doesn't work well | Not smooth (slow) | Fast |
| iPad | Safari | Smooth | Smooth | Smooth | Fast |
| iPhone | Safari | Smooth | Smooth | Smooth | Fast |
| Windows Phone | IE Mobile | Smooth | Not smooth (slow) | Smooth | Slow |
| Desktop | Chrome | Smooth | Smooth | Not smooth | Fast |
| | Safari | Smooth | Smooth | Smooth | Fast |
| | Firefox | Not smooth (slow) | Smooth | Not smooth | Fast |
| | IE 7 | Not smooth (slow) | Not supported | Not supported | Fast |
| | IE 8 | Not smooth (slow) | Not supported | Not supported | Fast |
| | IE 9 | Not smooth (slow) | Supported | Smooth | Fast |
| | Opera | Smooth | Not smooth (slow) | Smooth | Fast |

3.2. Global Functions

Table 3.2 Global functions testing result

| Devices | Browsers | Touch | Gesture | Others |
|---------------|------------------------|---------------------------------|--------------------------------|---|
| Android | Browser (WebKit based) | Supported | Not supported | |
| | Firefox | Supported | Not supported | Sometimes the images aren't loaded well |
| | Opera Mobile | Supported | Supported (up to 9 fingers) | Image size isn't appropriate |
| iPad | Safari | Supported | Supported | |
| iPhone | Safari | Supported | Supported | |
| Windows Phone | IE Mobile | Supported (partly) ¹ | Not supported | - Element positioning is messed up. |
| | | | | - Image size isn't appropriate |
| Desktop | Chrome | Not supported | | |
| | Safari | | | |
| | Firefox | | | |
| | IE 7 | | | - Element positioning is messed up. |
| | IE 8 | | | - Image size isn't appropriate |
| | IE 9 | | | Image size isn't appropriate |
| | Opera | | | Image size isn't appropriate |
| | | | | - Element positioning is messed up. |
| | | | - Image size isn't appropriate | |

¹ Not sure if it is partly supported indeed or because I used simulator therefore not all function can be applied.

APPENDIX D:

PRODUCT

DOCUMENTATION

Documentation

Cross-Platform Image Gallery

JavaScript Plugin



Date : 4 May 2012

Author : Daniar Gita Nindiary Suprianto

Introduction

To make a gallery usually we need a lot of work to do. But if there is a plugin, why should we bother about it anymore. This plugin deals with its pagination, its navigation, and etcetera. Moreover, it works everywhere, either desktop or mobile or tablet. It is effective and simple.



Figure1.1 The gallery

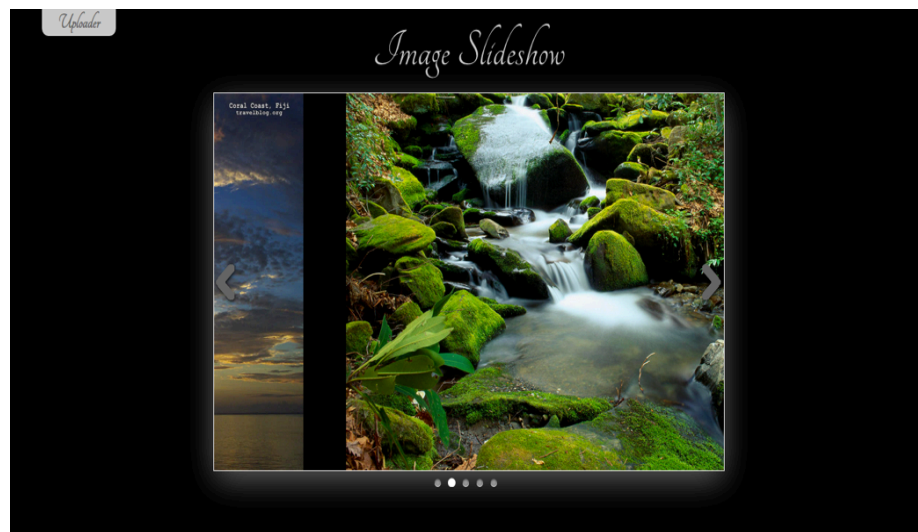


Figure1.2 Moving page

The idea

In the initial situation, the plugin wraps each image with div element with relative position to its parent, so that the image position can be centered. The next step is wrapping each div with li element and float it to the left. Then we come to the final step, in this step I show one li element in a display container.

The illustration of those steps is shown below:

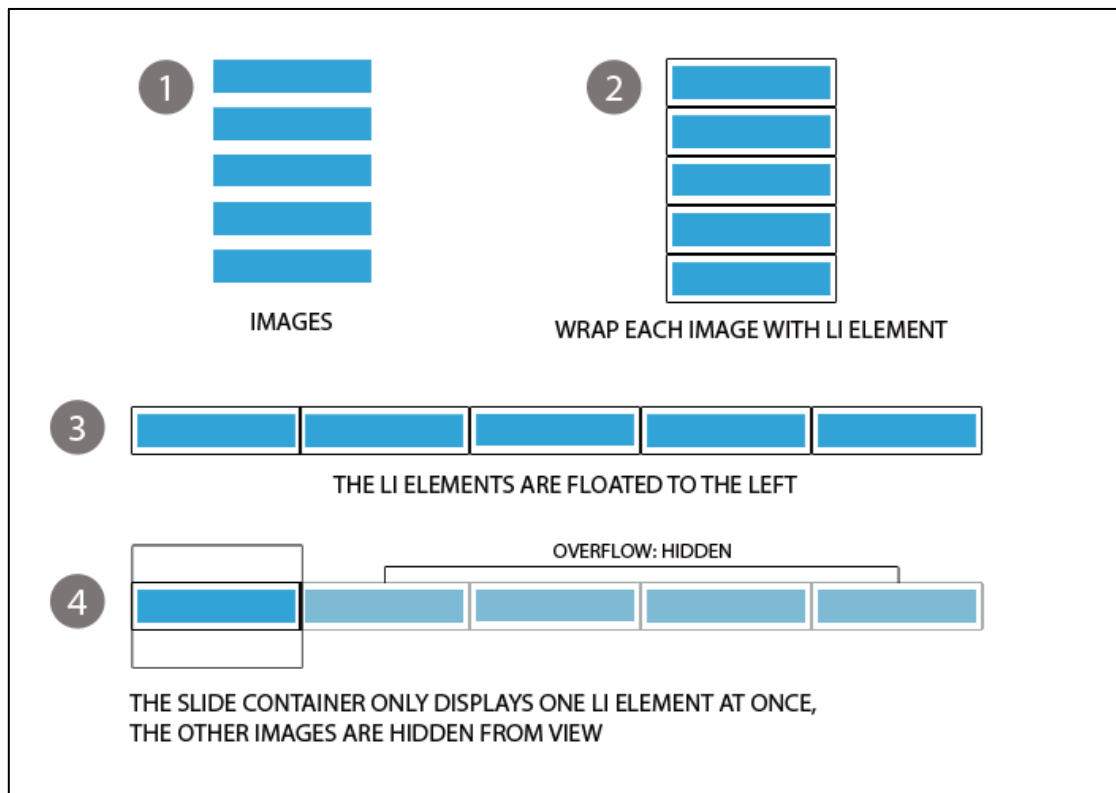


Figure1.1 How it works

Step 1 – Set up the HTML markup

The first step is to setup the HTML markup. The plugin only needs a bunch of images wrapped in a wrapper. User does not need to set anything apart from this step in their HTML file. There is an option to put a comment/note in every image. If you want to do so, all you have to do is adding `` markup inside the ``

index.html

```

1      <div class="wrapper">
2          <ul>
3              <li> <!--WITH COMMENT-->
4                  
5                      <span>Lorem ipsum dolor sit amet,
6                      consectetur adipiscing elit.<br>Pellentesque molestie
7                      elit nec massa iaculis id tincidunt eros mattis.</span>
8              </li>
9              <!--WITHOUT COMMENT-->
10             <li></li>
11             <li>
12                 
13                     <span>Lorem ipsum dolor sit amet,
14                     consectetur adipiscing elit. Pellentesque molestie elit
15                     nec massa iaculis id tincidunt eros mattis.</span>
16             </li>
17             <li></li>
18             <li>
19                 
20                     <span>Lorem ipsum dolor sit amet,
21                     consectetur adipiscing elit.<br>Pellentesque molestie
22                     elit nec massa iaculis id tincidunt eros mattis.</span>
23             </li>
24         </ul>
25     </div>

```

Step 2 – the CSS

In order to make the gallery becomes more attractive, we need to set the position, the color, etc. Well, in this part the plugin doesn't make it automatically generated because the need is different for each gallery. So I make it up to users themselves. But there are several general setting that have to be set as follow:

General setting

```
1  .wrapper {
2      position: relative;
3      margin: auto;
4      width: 1000px;
5      height: 550px;
6  }
7
8  .wrapper .slide-container {
9      display: block;
10     overflow: hidden;
11     margin: auto;
12     position: relative;
13 }
14
15 .wrapper .slide-container ul.slide-wrapper li {
16     overflow: hidden;
17     position: relative;
18 }
19
20 .wrapper .slide-container ul.slide-wrapper li .img-object {
21     position: relative;
22     margin: auto;
23 }
24
25 .wrapper .slide-container li .img-object img {
26     clear: both;
27     position: relative;
28     margin: auto;
29 }
```


Step 3 – the JavaScript

On the JavaScript file, users only need to call the plugin constructor and the plugin will do the rest process. Probably users want to remove or add several features to the gallery, the only thing they need to do is change the value for each feature then the setting will be adjusted.

Default constructor

```
1 $(document).ready(function() {
2     $('.wrapper').myGallery();
3 });
```

Set the features

```
1 $('.wrapper').myGallery({
2     navigation : true, // || false
3     pagination : true, // || false
4     slideshow : {
5         enable: true, // || false
6         firststate: 'start', // || stop
7         speed: 3000 // in milisecond
8     },
9     fullscreen : {
10        enable: true, // || false
11        imageSize: 'fill' // || stretch || fit
12    },
13    rotate : true, // || false
14    uploader : {
15        type: "none", // || file-browser || drag-and-drop ||
16        both
17        maxSize: 3, // maximum file size to be uploaded
18        targetPHP: "uploader.php"
19        // php file name, must be filled unless the type is set to
20        "none"
21    },
22    onWindowResize : {
23        enable: true, // || false
24        height: {
25            max: window.innerHeight,
26            // maximum height of the container when window is resized
27            min: 100
28            // minimum height of the container when window is resized
29        },
30    },
31    // ... other options ...
32});
```

```

25         width: {
26             max: window.innerWidth,
27             // maximum width of the container when window is resized
28             min: 100
29             // minimum width of the container when window is resized
30         },
31         preload : {
32             enable: true, // || false
33             targetPHP: "uploader.php"
34             // php file name to resize image when loading, must be
35             // filled unless the preload.enable is set to false
36         }
37     });

```

Note : values in bold are the default values.

Step 4 – the PHP

This plugin involves server-side process for image resizing, cropping, and uploading. If users want to use either image manager or fullscreen with fill mode or preload, users have to set the PHP file first. There are only two parts in the PHP file, the first part is to upload the image and the second part is for resizing and cropping the image which is needed for fullscreen fill mode and preload. However, these two parts can be merge in one PHP file only as I did.

uploader.php (image uploader)

```

1  class UploadedFileXhr {
2      function save($path) {
3          $input = fopen("php://input", "r");
4          $temp = tmpfile();
5          $realSize = stream_copy_to_stream($input, $temp);
6          fclose($input);
7          if ($realSize != $this->getSize()){ return false; }
8          $target = fopen($path, "w");
9          fseek($temp, 0, SEEK_SET);
10         stream_copy_to_stream($temp, $target);
11         chmod($path, 0777);
12         fclose($target);
13         return true;
14     }

```

```

15 function getName() { return replaceName($_GET['file']); }
16 function getSize() {
17     if (isset($_SERVER["CONTENT_LENGTH"])){
18         return (int)$_SERVER["CONTENT_LENGTH"]; }
19     else { throw new Exception('Getting content length is not
20         supported.');
```

```

21     } }
22
23 class UploadedFileForm {
24     function save($path) {
25         if(!move_uploaded_file($_FILES['file']['tmp_name'], $path)){
26             return false; }
27         chmod($path, 0777);
28         return true;
29     }
30
31     function getName() {
32         return replaceName($_FILES['file']['name']); }
33
34     function getSize() { return $_FILES['file']['size']; }
35
36 }
```

uploader.php (image resizing and cropping)

```

1 $upload_folder = 'uploads';
2
3 //works with both POST and GET
4 $method = $_SERVER['REQUEST_METHOD'];
5
6 if ($method == 'GET') {
7     if ($_GET['act'] == 'remove'){ unlink($_GET['url']);
8     } else{ imageResizer($_GET['url'], $_GET['fx'],
9     $_GET['fy'], $_GET['w'], $_GET['h'], $_GET['tx'],
10    $_GET['ty'], $_GET['ratio'], $_GET['type']); }
11 } elseif ($method == 'POST') {
12     if ($_POST['act'] == 'remove'){ unlink($_POST['url']);
13     } else{ imageResizer($_POST['url'], $_POST['fx'],
14    $_POST['fy'], $_POST['w'], $_POST['h'], $_POST['tx'],
15    $_POST['ty'], $_POST['ratio'], $_POST['type']); }
16 }
17
18 //FOR IMAGE PRELOAD AND FULLSCREEN TYPE "FILL"
19 function imageResizer($url, $fromX='', $fromY='', $width =
20 '', $height = '', $toX = '', $toY = '', $checkRatio =
21 false, $type = false) {
22     $properties = getimagesize($url);
23     if ($width == NULL || $width == '') $width =
24     $properties[0];
25     if ($height == NULL || $height == '') $height =
26     $properties[1];
```

```

19     if ($checkRatio || $checkRatio == 'true'){
20         $ratio_orig = $properties[0]/$properties[1];
21         if ($width/$height > $ratio_orig) { $width =
22             $height*$ratio_orig; }
23         else { $height = $width/$ratio_orig; }
24     }
25     $bigger = false;
26     $ratio = $width/$properties[0];
27     $heightRatio = $properties[1]*$ratio;
28
29     if ($fromX == NULL || $fromX == ''){
30         $fromX = ($properties[0]-$width)/2;
31         if ($fromX < 0){ $bigger = true; $fromX = 0; }
32     }
33     if ($fromY == NULL || $fromY == ''){
34         $fromY = ($heightRatio-$height)/2;
35         if ($fromY < 0) $fromY = 0;
36     }
37     if ($toX == NULL || $toX == ''){
38         if ($width > $properties[0]) $toX =
39             $properties[0];
40         else $toX = $width;
41     } elseif ($toX == 0)
42         $toX = $properties[0];
43
44     if ($toY == NULL || $toY == ''){
45         if ($height > $properties[1]) $toY =
46             $properties[1];
47         else{
48             if ($bigger) $toY = $height-$fromY;
49             else $toY = $height;
50         }
51     } elseif ($toY == 0) $toY = $properties[1];
52
53     $posName = strpos($url, '/');
54     $name = $url;
55     while ($posName!=false){
56         $name = substr($name, $posName+1, strlen($name));
57         $posName = strpos($name, '/');
58     }
59
60     if ($type == 'full') $path = 'uploads/full/'.$name;
61     elseif ($type == 'thumb') $path =
62         'uploads/thumb/'.$name;

```

```
59     else $path = 'uploads/.'.$name;
60
61     // This resamples the image
62     $image_p = imagecreatetruecolor($width, $height);
63     if ($properties['mime'] == 'image/jpeg'){ $image =
64     imagecreatefromjpeg($url);
65     } else if ($properties['mime'] == 'image/gif'){ $image
66     = imagecreatefromgif($url);
67     } else if ($properties['mime'] == 'image/png'){ $image
68     = imagecreatefrompng($url);
69     }
70     imagecopyresampled($image_p, $image, 0, 0, $fromX ,
71     $fromY, $width, $height, $toX, $toY);
72     imagejpeg($image_p, $path);
73     chmod($path, 0777);
74     echo $path;
75 }
```

APPENDIX E: TEST REPORT

Test Report

Cross-Platform Image Gallery

JavaScript Plugin



Date : 25 May 2012

Author : Daniar Gita Nindiary Suprianto

1. Objectives

This report intends to show the results of the tests carried out

2. Deliverables

Test Results

Table 2.1 Test Result

| Functional Requirements | Result |
|----------------------------|-------------------------------|
| 1 | Working |
| 2 | Working |
| 3 | Working (except Window Phone) |
| 4 | Working |
| 5 | Working |
| 6a&6b | Working |
| 6c | Working |
| 6d | Working |
| 6e | Working |
| Non-functional requirement | Working |