# Scope

Knowing the content and understanding how it fits into the website is important, if you have too much information people can get overwhelmed and not having enough the opposite. What is also true is if you have the wrong information. Developing a website around architecture but the content is based on coding will not give the required result.

### Content requirements (content inventory, sitemap)

We reviewed the content supplied by the faculty and noted there are a few groupings of information. We developed with concept mapping and after running through a few sitemap changes, but we feel the information when put into these groups allowed for the containment of information. This logical layout allowed for easy placement of data and content and also allowed for users to easily find information.

Note in the sitemap (Appendix 3) that we grouped into targeted info, event info, history and signup. Since developing the wireframe and with user testing we adjusted this, but still have basic groups of content.

### Technical Requirements

We have opted for a minimal approach to JavaScript and really tried to use it sparingly. However we’ve used JavaScript to try and enhance or make easier the use of the website. The three main areas we have used JavaScript in is:

* *Form validation*
* *Modal display for snippets of information*
* *Photo gallery*

The previous mentioned areas were designed first with HTML and CSS only and with mobile devices and other media in the front of our minds (this aside from form validation which without JavaScript will validate on server side). Once confirmation of only HTML and CSS was working we were then able to pursue JavaScript to help enrich the use of the website.

We started with the Form Validation; we did not want to recreate the wheel though considering it’s been done so many times before. We settled on jQuery Form Validator (http://www.formvalidator.net), mainly based on the usability and simple design and it’s integration with HTML5 validation. The form is more of an email off to staff and as such didn’t need very strict validation, however we wanted to make sure that people who have legitimate sign ups or requests gave the right information. The jQuery is run by setting events listeners to “data-validation” attribute in the “input” DOM element. This allows rapid deployment and can run over many forms in a single page. We only needed validation for having something in the name field, and properly formatted email and a phone number field with numbers in it.

The second area we focused on was modal display for the privacy policy. This is a very simple piece of JavaScript, however we believe rather important to be able to have more information in a website than you actually need to display in any one time, ie only showing when a user want’s to see it. We started with a normal link to a page that opens in a new tab to show the privacy policy just in case of JavaScript issues with client rendering. All we had to do then was create a DIV in HTML, set it to hidden in CSS and attach an event listener in JavaScript to set it back to display block.

The last area was the Photo Gallery, create this from scratch just would not have been feasible in the timeline we had, so again we went looking for a 3rd party alternative. We turned again to jQuery and found a powerful product called PhotoSwipe (http://photoswipe.com). This was exactly what we were looking for, having a gallery of thumbnail images already present and displaying then in a image gallery with failover to hard links to the full size images. PhotoSwipe did have a lot of setup and configuration though to get started. This didn’t just mean the import of the js files, but also had a set of css files and still needed to create the image gallery in the right format for the DOM to get parsed by the jQuery plugin. There are some powerful features in this plugin, but we went with a simple set of images with links already present and used the jQuery to find the images and create a list and then inject that in the plugin (the image gallery itself). The result is really something and looks great and also fails over with the end user not knowing any different.

With all the steps above noted we took a mobile first approach testing at each part on mobile and then working out kinks and “extra bits” on a computer. We also took this with naming and labeling and sematic layout of HTML to allow for non-web devices to be able to represent the data/information for those not even using a browser. This is tested in steps of disabling options in the browser, first JavaScript, second CSS and then using a text only browser. The site still gets across the required information albeit not as pretty but in a satisfactory fashion.