PROJECT OVERVIEW

This project aims to design and develop a set of modularised components, which will form the basis of a responsive HTML template structure for integrating with the existing Cambridge University websites.

This is in response to a need for:
• aligning multiple sites which fall under the banner of Cambridge University to a consistent look and feel
• updating the existing templates with forward-looking technologies that have the aim of future proofing Cambridge University websites for the next 5 years
• allowing users to browse the websites across multiple devices and platforms

TECHNICAL SPECIFICATION

FUNCTIONAL REQUIREMENTS

The templates are intended to serve as a basis for integration with the top layers of the Cambridge University Information Architecture (IA) and will include working examples of all the components and modules found within these pages. The templates should adequately demonstrate the HTML, CSS and JavaScript necessary to implement these pages within the current content management systems.

LIVE STYLE GUIDES

To deliver the components as modularised HTML, we are proposing a set of static HTML pages with supporting CSS and Javascript that will detail each component individually, including notes on implementation and custom permutations.

The live style-guide will form a library structure that can be used as a reference for other developers during integration. The style guide will also include template variants showing how certain components can be combined to create page types and best practice for structuring whole pages. The live style-guide should also give explicit instructions on how modules and components can be extended.

The list of components and template variants detailed in the style guide will be agreed within the Explore stage of Project Light.

For the purposes of browser testing we will be producing several example pages to mimic content and components from the live site, based on a sample set of pages chosen during the Explore stage.
RESPONSIVE TECHNIQUES

We will be implementing a responsive layout with a mixture of percentage based fluid containers and the use of Media queries. Functionality delivered solely by media queries will be added as a progressive enhancement and therefore will not be available to users with Internet Explorer (IE) devices. IE users will receive a partially fluid site. We will not be using JavaScript to polyfill this functionality.

To create fully responsive images there needs to be some consideration as to whether images should be reduced or removed, depending on the users platform and whether or not this will be handled as a server side activity or within the CSS or JavaScript. We will ensure the templates are supported with the correct technique where necessary and display guidance for developers implementing these techniques.

It’s important to ensure video, audio and images assets have been optimised and produced at the requisite size and bitrate to ensure mobile devices can browse easily. We will make recommendations where necessary to allow guidance for developers in implementing these elements.

Touch screen devices do not respond to hover states and users will expect swiping and gesture recognition when browsing. We will ensure any interactive elements such as carousels and tabbed navigation are designed and built with multiple interaction paradigms in mind.

CSS COMPONENT STRUCTURE

When implementing the Build stage we will be using an Object Oriented CSS strategy (http://oocss.org/). This design pattern was developed as a way of helping large organisations to reduce the amount of code they were writing, and to ensure greater standardisation across existing code. We think this will be particularly useful to you if you are planning to roll out the build over a long time period and will ease the change management activities for others taking on the new implementation.

The CSS will be produced in a modular style detailing reset styles, font handling and structural information with supplementary CSS created for look-and-feel, component level styles and device specific styles. All basic tags will be styled with a default style to offset the effect of the reset used. We will endeavour to ensure the CSS can be used straight out the box for all basic use cases including nested lists and standard HTML tags.

To ease implementation we will produce two variants of the finalized stylesheets. One to include all possible styles for users who do not wish to customize their site. This will include guidance on re-teming the colour scheme for very basic customization purposes. We will also produce a more customizable boilerplate stylesheet handling the main structural elements which can then be augmented with the addition of required components. In addition to this, we will also provide a print stylesheet listing all possible print styles. The print styles are unlikely to require customization.
We will implement a naming convention for all class selectors that is project specific to reduce conflict with existing styles, or areas of the site which are using commonly found global class names. The stylesheets will not include any tag specific selectors to ensure classnames can be associated with HTML5 elements for those wishing to extend the supplied markup during integration.

**USE OF HTML5 AND CSS3**

We will be using an HTML5 doctype to allow the use of HTML5 video and audio tags. To mitigate the lack of W3c validation for this doctype we will be creating markup using the XHMTL 1.0 specification and validating pages during development before swapping in the HTML5 doctype before release. This is also intended to mitigate any integration issues with combining the supplied templates with existing CMS builds that restrict HTML5 markup usage.

We will be using HTML5 tags sparingly to reduce the need for a JavaScript polyfill required to style entire pages in IE. Where required we will add the shiv for audio and video when included in the page. If necessary we may also introduce the use of HTML5 API’s with a fall back, such as Flash video for older browsers. Any additional use of HTML5 not specified here will be agreed and defined as part of the project Explore stage.

Where possible we will be optimising the rendering by leveraging the devices native capabilities with CSS3 and not relying solely on JavaScript. This gives improved performance across low bandwidth, low RAM devices. We will also be rendering any progressive enhancements to the look and feel with CSS3 and styling older browsers with a more simplified view. In an effort to reduce code bloat we will not be using any JavaScript to polyfill missing CSS3 functionality.

We will be avoiding loading custom fonts with font face technologies. Rendering can be problematic on mobile devices and it also introduces an unwanted download overhead.

**NON-FUNCTIONAL REQUIREMENTS**

The templates will be viewable across a range of devices and platforms as specified in the agreed support matrix (see below). The websites should be optimised for performance so as not to compromise load times across devices with lower bandwidth.

**ACCESSIBILITY**

Templates and components will be built to WCAG 2.0 Level AA guidelines, as specified in the brief (Section 3 - Specification of Tender documentation). We will also ensure all content falls within the correct colour contrast levels for Level AA compatibility and that the necessary Access keys and Skip to links are included in the templates and navigation structures.
The meta-data for pages will be set to ensure zooming is still possible for users with visual impairment.

Any JavaScript required will be developed using unobtrusive JavaScript techniques to ensure content is visible and usable to any client without JavaScript.

VIDEO STREAMING RECOMMENDATIONS

Although the video streaming capabilities of the Streaming Media Service are outside the remit of Project Light, the question of video implementation intersects Project Light in a couple of places: players and responsive design.

It’s our assumption that the SMS people will want to implement an Oembed-style solution for sites which are capable of dealing with that standard. It should be recognised that not all systems will be able to use Oembed.

Video codec and container standards for HTML 5 have been in a state of wild flux, but browser vendors are starting to converge on a standard solution: H264 in an MP4 container. With the Mozilla Foundation apparently about to concede defeat on free video formats (see http://blog.lizardwrangler.com/2012/03/18/video-user-experience-and-our-mission/1234/), it’s very likely that H264 will be the de facto standard for video on the web for the next several years. The vast majority of the web will be able see H264, either natively using the HTML5 <video> tag, or via the Flash Player, which also has H264 support.

It’s up to the SMS admins whether they want to proceed with WebM encoding. WebM, Google’s free video format, plays natively in Chrome, Firefox 4+, and Opera, but not in other browsers. With future, but as-yet-unspecified versions of Firefox supporting H264, it may or may not be judged worth it to encode WebM versions of content.

Regardless of HTML 5 formats chosen for encoding by the SMS team, we recommend the use of the player from http://videojs.com/ for both the Oembed provider and for any non-Oembed embed codes as well. This player provides HTML5 video capabilities, with native playback support for WebM, H264, and the older and less-used OGV format. There is a Flash player fallback for clients which can’t handle HTML5 video natively.

The second issue is: how will the video player interact with the responsively designed site, especially at lower widths? What will happen to the video?

We are, in general, taking a capabilities-based approach to the design. This means that we’re not trying to detect every individual browser client, by user-agent, and offering a tailored version of the page based on that client. Instead, we’re using screen width as a way of inferring what the browser is capable of, and delivering appropriately-styled content based on this.
In the case of the <video> tag, we can use media queries to specify different files depending on what the client’s screen width is. For example:

```html
<video controls>
    <source src="mySmallVideo.webm" type="video/webm" media="all and (max-width:600px)">
    <source src="myVideo.webm" type="video/webm">
</video>
```

In this case, clients with display resolutions of 600px and below will see a small version of the video, saving on bandwidth and speeding up load times. All other clients will get the full-sized video.
BROWSER SUPPORT

The templates will be developed using the following browsers, which have been selected and graded based upon existing usage stats for this website and general browser usage behaviors on the respective platforms:

<table>
<thead>
<tr>
<th>BROWSER</th>
<th>VERSION</th>
<th>WIN 7</th>
<th>MAC OSX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Explorer</td>
<td>7.0</td>
<td>B</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>8.0</td>
<td>A</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>9.0</td>
<td>A</td>
<td>X</td>
</tr>
<tr>
<td>Mozilla Firefox</td>
<td>3.6</td>
<td>X</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>Latest stable</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Google Chrome</td>
<td>Latest stable</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Safari</td>
<td>Latest stable</td>
<td>X</td>
<td>A</td>
</tr>
<tr>
<td>Opera</td>
<td>Latest stable</td>
<td>X</td>
<td>B</td>
</tr>
</tbody>
</table>

NB: Whilst IE 10 is not listed as a supported browser until it reaches full release we do feel it should be considered as a potential platform for future use. We recommend undertaking browser testing on this platform on the final delivered templates as a side project to ascertain any future risk on supporting this platform before release.
<table>
<thead>
<tr>
<th>PLATFORM</th>
<th>OS</th>
<th>GRADE</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOS IPHONE</td>
<td>4.0</td>
<td>B</td>
</tr>
<tr>
<td>Mobile Safari</td>
<td>5.0</td>
<td>A</td>
</tr>
<tr>
<td>IOS IPAD</td>
<td>5.0</td>
<td>A</td>
</tr>
<tr>
<td>Mobile Safari</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Android</td>
<td>4.0</td>
<td>A</td>
</tr>
<tr>
<td>Android Browser (webkit)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blackberry</td>
<td>7.0</td>
<td>B</td>
</tr>
<tr>
<td>Blackberry Browser (webkit)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**GRADES:**

- Popular modern browsers are A grade. Head QA will test all templates in A-grade browsers.
- B-grade browsers are less popular, older or secondary browsers. These browsers are used by a very small percentage of site visitors. Issues found in secondary browsers will be given less priority over A-grade issues.
- Browsers that have negligible usage statistics or will soon be redundant by forced updates are given X grade status. As documented in the Statement of Work, IE6 is excluded. Its usage is in decline, it is no longer an industry supported browser and would require additional time and budget to facilitate its nuances. X-grade browsers (and all earlier versions unless otherwise stated) are not included in the QA process.
CONTACT

ALEX SEGROVE
Project Manager

email: alex.segrove@headlondon.com
mobile: +44 787 866 0923
direct: +44 20 7079 0260

DAVE HRYCYSZYN
Technical Director

email: dave.hrycyszyn@headlondon.com
mobile: +44 77 9976 6382
direct: +44 20 7323 9787

Office: +44 20 7323 9787