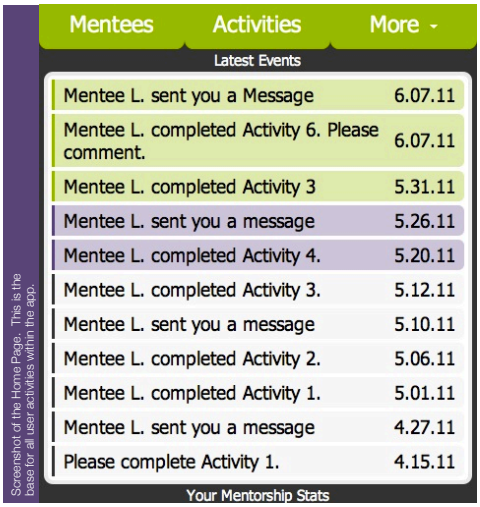
MOBILE WEBSITE

Obstacles to Overcome: what needs to be done

By Stephen McQuaid



What has been done:

During my internship in the Summer of 2011, I was able to design and code the layout and styling of the site. I optimized the UI for a mobile device by providing readable text, large buttons, exclusively vertical scrolling, priority functionality and layout that will be familiar to most mobile device users. A hardcoded version of the site is up, working, and provides enough functionality to get the idea of how the site should work. What is left to do is hookup the front end with the backend of the eMentoring Application. While fairly simple, there are a few large obstacles that must be over come.

A few notes on how I coded:

For formatting the pages, I semantically structured the page. The data is represented by the structure of tags, the layout is not, and should not be represented by the tags. After creating the html, I styled it using CSS. To make writing CSS easier I used a framework called LESS CSS (lessees.org) This framework required a compiler to render minified CSS output which I link on the page. The structure of my .less files is as follows: I created an object.less file to represent the style of each individual object on the page. I then created a layout.less file to link the semantic html tags with the objects I create. Since layout.less already includes objects.less, there is no need to import objects on the html page.

What needs to be done:

To make this site live there are a few tasks which need to be solved:

1. Add content dynamically.
2. Parse textarea input for illegal characters and mysql insertion attempts.
3. Create a mechanism to distinguish mobile users from desktop users and allow either user to access the opposite site.

1. To add content dynamically there are a few options.

a) Keep desktop HTML, override style with new CSS.

b) Create entire new view and dynamically deliver html for mobile device (saves bandwidth for user (is faster), as mobile determination is done mostly server-side.)

c) Keep desktop HTML but restructure DOM via javascript.

2. Currently the site uses YUI wysiwyg editor. Unfortunately it does not support mobile devices so the work around is <textarea> with lots of backend parsing. To my knowledge currently the parsing is done client side via javascript/JQuery for the YUI object. This shouldn't be too difficult to implement on a textarea, though there need to be more failsafes in case javascript is disabled and the user submits raw textarea data. Data also needs to be parsed (via regex) as different browsers have different escape characters and data is stored in the database is html with
 tags as new line markers.

3. Perhaps the simplest, yet also most complex facet of implementing this website is making the determination to switch to mobile or not. While a hardcoded user agent string parser could work, it is far from ideal. Not only do UAS's change weekly, but relying on the word mobile is not a good idea. Also forcing a user to use the mobile version is not ideal. Two rules: Keep mobile determination a mostly client-side process, that way browsers will tell if they are mobile or not (and may provide options to users on which mode to use). Make the process of switching between mobile and desktop as clear and clean as possible.

Solutions:

a) The subdomain approach. Have two separate sub domains for mobile vs desktop (Whether or not these urls actually deliver separate content is irrelevant to the user, the user knows where they are and what they are going to get at each location. Hyperlinks can guide this approach for accessibility reasons.

b) The cookie approach. For a first time user, no cookie will exist, and so the UAS, object detection/javascript, or the browser will determine a mobile browser or not and serve the appropriate cookie. Alternately, navigating to a specific URL will give the browser a cookie, which dictates that the server display either the desktop or mobile version. (IE m.sf.com contains 1 page which serves cookie and redirects to main site. Clicking a link to go to the mobile version or desktop version will change the value of this cookie.

Future work:

I will continue to develop this site to provide more functionality and improve user experience over time. The online repository can be found at:

https://github.com/stevemcquaid/Smart-Futures/tree/updates/MobileSiteV1

Please commit any changes to the core of this site here and provide written updates within the appropriate README file.