Student Services Website Front-End Design Document

Global Requirements

•Contains a Header Area.

•Contains a Content Area.

•Contains a Footer Area.

•Content area should be easily replaced with other content.

•All field validation will be done on the back-end.

•All error message display handling will be done on the front-end.

•Any server calls should be done using JQuery and AJAX.

•The server will have no concept of "page".

•Moving from "Login" to "AccountInfo" will not change the page/URL.

•Nothing at all will change on the server to reflect workflow.

•The front-end will handle transitions between pages and workflow.

•There is no need for workflow beyond Login or Create Account.

•There is no need to handle back button functionality.

•All display changes due to form factor must be handled front-end.

•All responsive design should use width-based media queries.

•There is no front-end work required to implement Account management.

•Accounts are already stored as XML files on the server.

•Back-end has been coded so that case should not matter on the front-end.

Login Screen Only

•Login section allows a user who already has an account to log in.

•Successful login will take you to the Account Information screen.

•Unsuccessful login will display error messages on the screen.

•Create Account section will allow a user to create a new account.

•They must provide a unique username that is at least 6 characters.

•They must provide a password that is at least 6 characters.

•They must provide an e-mail address that contains an "@" character.

•They must verify their e-mail address by typing it in again.

•Successful login will take you to the Account Information screen.

•Unsuccessful login will display error messages on the screen.

Account Info Screen Only

•Account Info Section should list every value in the user's account.

•Each item should be updatable except for the Username.

•Disable/Replace the text-box and hide the update button for Username.

•No field validation required at this time for updating values.

•Do not use an HTML table for the grid; use styled divs instead.

•Add Element Section should allow the user to add a new element.

•User will provide a name and a value for the new account element.

•"Username" will not be allowed as the Element Name.

•This is currently the only input validation we have implemented.

Global Architecture

•Keep the HTML content layer free from obtrusive markup or scripts.

•Keep the HTML content layer organized and precisely scoped.

•Keep the CSS layer organized and tightly scoped.

•Use fully qualified class names wherever possible.

•Use direct-child selectors wherever possible.

•Apply Object-Oriented JavaScript principles to organize behavior.

•Fully qualify and scope all methods to specific objects.

•Encapsulate reusable code within object methods.

•Separate concerns; Keep objects and methods small and focused.