Browser based User Interface (UI) for the AguaClara Design Tool

Abstract—**The AguaClara design tool is our method of distributing the AguaClara technologies globally. We provide a free online design service that provides customized designs on demand. The design service must scale to meet global demand for municipal drinking water treatment facilities. The current version of our client software requires that the client download the LabVIEW runtimes engine in order to request a design. This is a significant deterrent to new users. We need a web-based "thin" client that allows clients to easily access our design services. The new UI also needs to be generalized to handle any design request based on the types of designs that our design tool supports. The distribution of the design files will need to be changed from email based to posting them on the web for download.**

students	3
skills	Computer Science

I. Create a browser based version of the client design tool software (2 CS M.Eng. students)

The web-based client must be compatible with our goal of distributing multiple design elements (entire plant, individual reactors, LFOM, chemical dosers).

The web based client must take the input values for the design element and transmit them to the AguaClara design server. The design server will place those requests in a FIFO buffer and process them. The design server will open a Mathcad file and a Microsoft Word Specifications document that corresponds to the design element requested.

The UI must be intuitive, easy to use, and easily configured for different types of design requests. The UI must have code that packages the request in a format that can be unpackaged by the design server. Data structures must be considered and handled carefully because the data type must be known in order to send the data to Mathcad correctly.

We need to decide if the UI needs to collect user information (name, company, email) or if we can leave the user anonymous.

The information from the UI must include the following:

- Country ID (based on a standard drop down menu of countries)
- Language (based on the options that we are willing to support - currently English and Spanish)
- Type of design (Complete plant, SRSF, flocculator, sedimentation tank, linear flow orifice meter, dose controller, entrance tank, calibration column, ram pump)
- List of variable names and their corresponding values. The values must be numbers stripped of their units and the units must have been meters/kilograms/seconds. The variable names must be the same as the variable names used in the Mathcad code.

II. Generalize the design server

The design server should be completely generalized so that it can process any design request. The browser based UI could send the design server a text string that is the name of the design element that is being requested. That text string would correspond to the name of the Mathcad file and specifications document file that LabVIEW would then open to process the design request. All of the client inputs for the design request need to be transmitted along with information about the type of variable and the variable name so that all of that information can be sent correctly to Mathcad. These inputs should be displayed in the design server so that the results of communication by clients can be verified. Determine what needs to change in the design server to receive and process requests from the browser based UI. Create a flexible data structure that will work for the design of any element (this design solution should not have anything that is specific for drinking water treatment). We are creating a design methodology that will be easily adapted to any automated design process. Upgrade the design server to be capable of designing anything.

III. Distribute the designs

The file size for an AutoCAD drawing of an AguaClara plant already exceeds 10 MB even without placing plate settlers in every sedimentation tank. As details of the structure are added the file size will exceed email capacity. Thus we need to switch to a method of posting the design requests on a web server and automatically generate a web page, the index, with links to the design files. We would then send an email to the client (if we collect the client's email address) with a link to the web page or alternately, the browser could tell the client where the files will be posted when the request is submitted. The web pages with links to the design files would then become a database of requested designs.