Team 404

Will Pruyn, Eric Lied, Claire Melzer, and Chenxing Liu

9/24/2015

Lab 1 Software Design Document

Introduction

For this lab we utilized software to create a view for the user so that they could see their temperature data in real time and additionally gave the user some controls for the third box. The software side was also used to send alerts via text to the user and store the temperature values data in a database.

Website Link

The website is currently being hosted here: <http://princofeedesignlab1-tempsensor.rhcloud.com/>

Technologies Used

* HTML5
* CSS3
* Javascript/JQuery/Ajax
* PHP
* Twitter Bootstrap 3 (http://getbootstrap.com/)
* Flot Charts (http://www.flotcharts.org/)
* Twilio (https://www.twilio.com/)
* Parse (<https://parse.com/>)
* OpenShift (https://www.openshift.com/)

Features

* Provides a scalable graph of the last 300 seconds of temperature data in real time to the user. This graph updates every second with new temperature data.
* Provides the current value of the temperature sensor in large letters in either degrees C or degrees F. User is able to specify which temperature scale is displayed.
* Provides a way for the user to virtually “press” the third button. When pressed on the website, the lights on the third box light up as though the physical button was pressed within 1 second of being pressed virtually.
* Sends a specified phone number a text message when the temperature data exceeds a set threshold value. The user is able to specify the upper and lower bounds for this value as well as the message that is sent for both. Further, the user may set or modify the phone number that the message is sent to.

Implementation

Below is how we utilized the technologies to provide the features required for the temperature sensor.

* **HTML5** – Used to create a skeleton for the website
* **CSS 3**- Used to style the website
* **Javascript/JQuery/Ajax** – Used to give the website functionality. Allowed us to implement a handful of open source APIs very easily in addition to providing front end logic for the system.
* **PHP** – our PHP pages were created so that the arduino could talk to our database more efficiently. We had originally tried to make the arduino send data to our database directly, however this took up to 14 seconds for each reading which is much too long. Instead, we wrote a couple PHP scripts and accessed them using HTTP GET requests. This allowed the temperature data to be send in less than a second. PHP was also used to send the text messages via a GET request made using Ajax on the front end.
* **Twitter Bootstrap 3** – Used to make the website look pretty.
* **Flot Charts** – Used to make the creation of the graph very simple as well as very clean.
* **Twilio –** Used to send text messages to the user’s phone via PHP
* **Parse –** Backend as a Service used for our database. This allowed us to make simple javascript calls rather than complex queries to access our data. Also allowed us to easily create and monitor the data in our database.
* **OpenShift –** Cloud deployment/hosting platform that allowed us to run our website on something other than localhost.