Temperature and Humidity Monitoring System

Marcel EnglmaierJustin Koehler, and Jason Pearson

December 10, 2014

sensors. To alleviate this problem we proposed to create a web site that would communicate with a network of reliable, home brewed sensor units. This website was originally created by another WMU Computer Science Senior Design team and it currently is used to communicate with the Temperature@lert sensors.

We are unaware at this time of some speci c information regarding the facili-

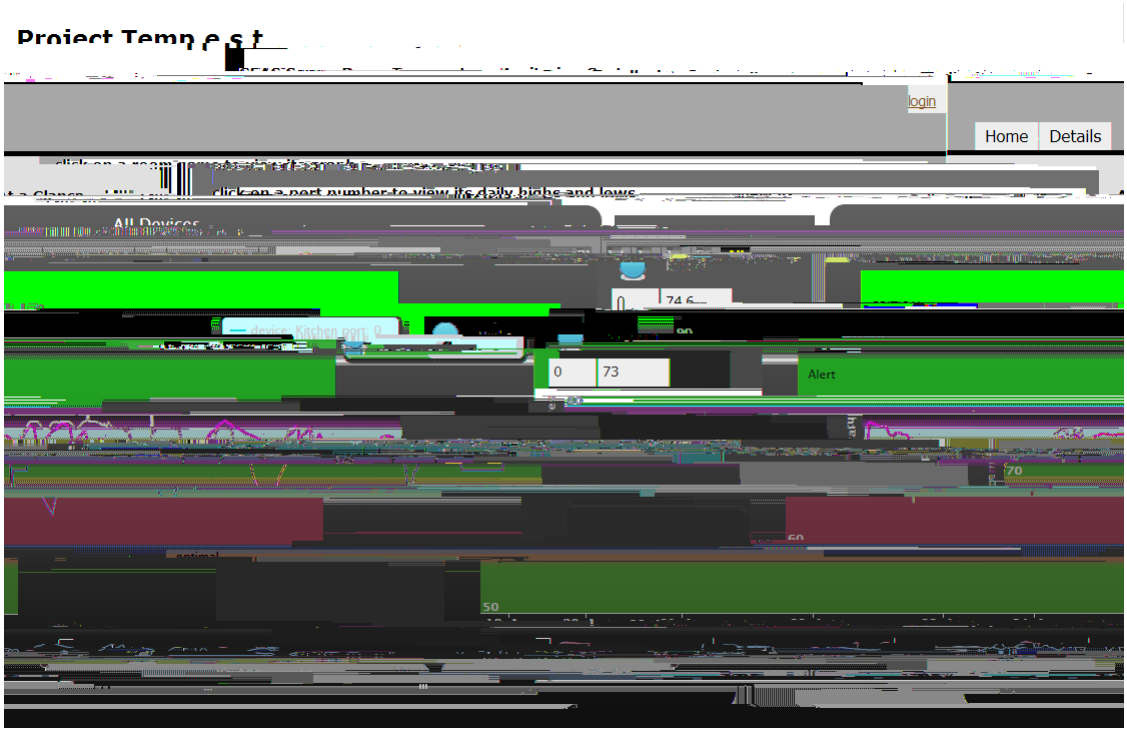


Figure 1: Initial View Of Site

The main page of the website that we inherited from the previous team is shown above. On this page there is a graph of all the temperatures of reported by all the sensor units that are currently 28(o)2eing tracked. It do8(o)2es not currently display any humidity readings. We will 28(o)2e evolving the site to include this information in future releases. The existing framework alloys easy data viewing with little programming e ort. We will use the existing framework and expand the

The login process is very basic at this time. The existing web site uses lack-

Figure 3: Add Page

Upon logging in as a system administrator this is what the admin sees. This is the general hub for editing anything on the site. From here the admin can see rooms, users, room assignments and device types easily. The page looks just the same for a room administrator when logging into the site. The only di erence is that the room user won't have the option to edit any49(ro)-28(om)-3s, devices, etc. If a non logged-in user tries to access this page it redirects them to the login page

Project Template

GIAS Server Room Temperature Monitoring Center

Add New Device

Figure 8: Page To Add Users

Above is the form for adding a new user. The required fields are name, email, and password. The password does have minimal requirements for good

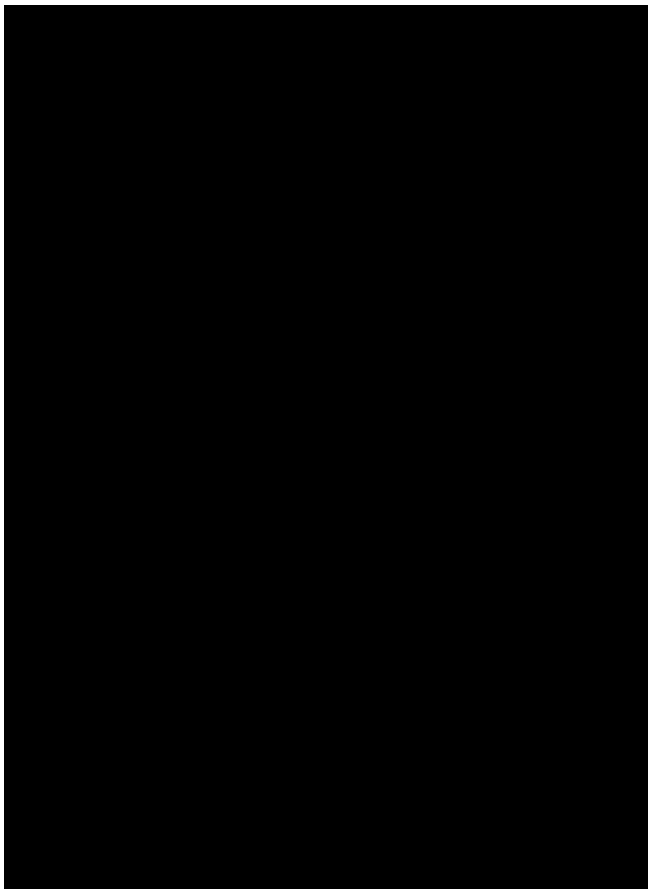


Figure 9: Entity Relationship Diagram

Above is the old database diagram of the web site. We decided to not make minor changes because the database was designed in a manner that didn't allow humidity recording w9thout adding another table

Design Decisions

Hardware, The Microprocessor

There were multiple options for microprocessors we could use. After some research, we narrowed down our list of possible units to these three: the RaspberryPi, the MSP430 Launchpad, and the Arduino Leonardo.

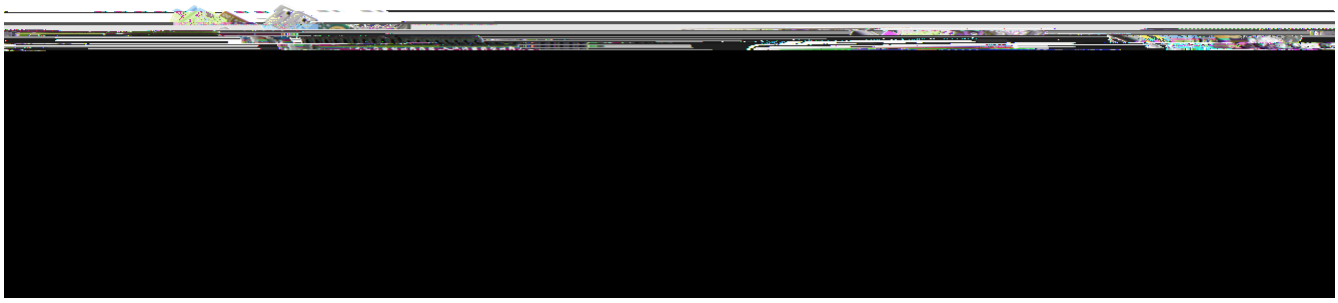


Figure 10: RaspberryPi, Arduino, MSP430 from Left to Right

Each of these devices had their own bene ts. The MSP430 Launchpad was the cheapest option while the RaspberryPi was the most expensive option. A big problem with the MSP430 Launchpad is that is o ers almost no features. It may be cheap, but the time, e ort, and money we would need to put in to this device to connect it to the internet made it unusable to us.

The Arduino Leonardo was our sedevith0to

Hardware, Enabling Internet Connectivity

We had two options for connecting our RaspberryPi sensors to the internet: a wireless or a wired connection. Both options we available to us by our Client.

Using the wired connection approach, we would be able to run the RaspberryPi using Power-over-Ethernet(PoE). All of WMU's ethernet cables in the

Software, Web Framework

Software, Sensor Unit Programming Language

Our RaspberryPi sensor unit requires some form of programming language.

Server

Web Site, Design and Styling

The existing web site that we took over from the previous team did not look ni1e at all. The buttons were ugly, the styling was poor, and the colors were mismatched. Furthermore, it was not mobile-compatibly at all. We had to nd a solution.

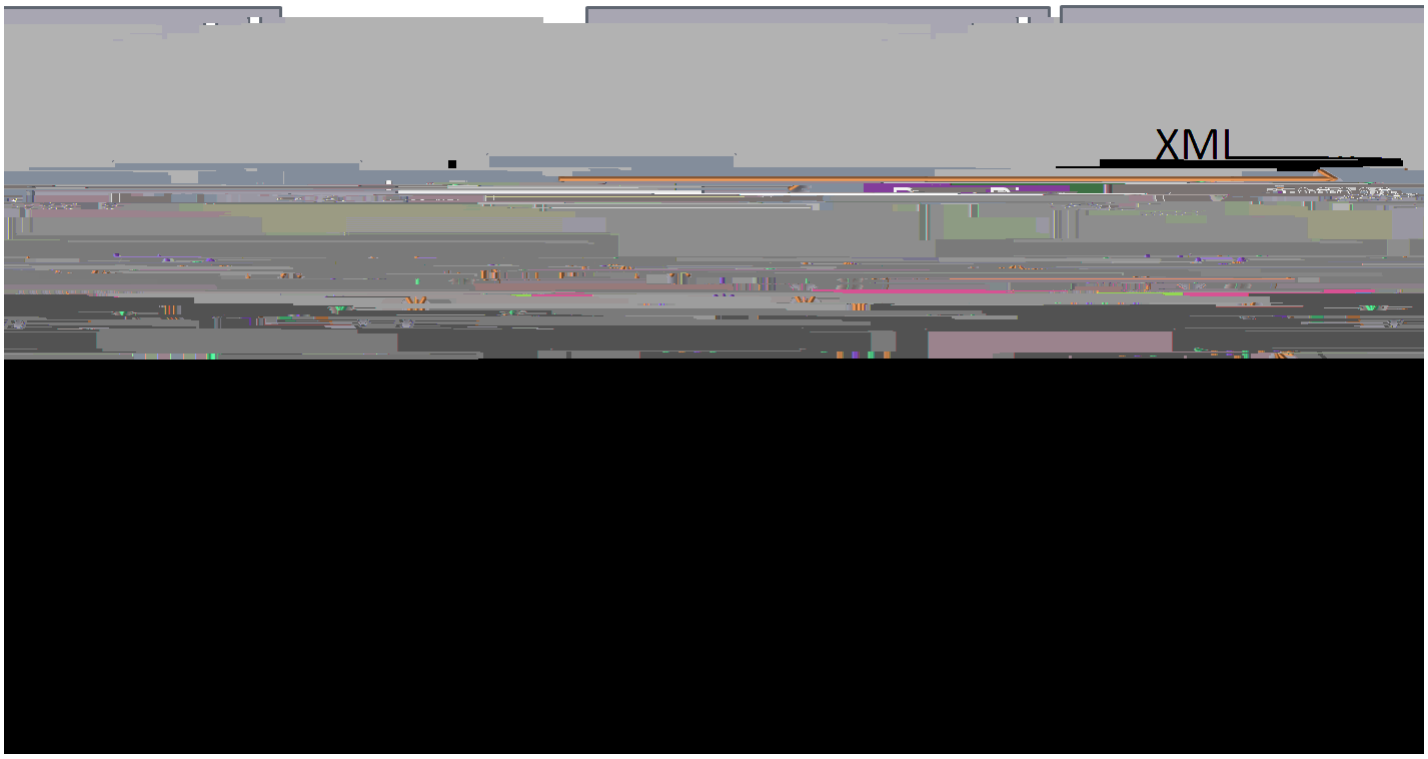
We decided to tackle the grap7s on the pages rst. The existing web site used High Charts, which is a rather useful grap7ing library. As it turns out, the version that was being used was quite outdated. We updated it, and most of the problems that existed were gone. The updated version provided mobile support, scaled well, and looked much cleaner.

There were alternatives for us to use. We could have s1rapped High Charts altogether and gone with commonly used libraries like Google Charts, or Chart.js. We also thought about using the YUI, the Yahoo User Interface Library. All of these are used often across many di erent implementations, but we decided to

Design

Next we shall look at how the pieces interact with each other to create the system. First we will look at an overview of how the server, the user and the devices interact and how they are setup.

On the right of the gure below, we have di erent rooms that are set up. Each room has sensors in it of any type. Each sensor hosts an xml document that shows the current information about the sensor. The central management


XML

Now when a user clicks on the tab it will expand to show what the current temperature and humidity are. Next to that is the status of the machine i.e. okay or critical. Upon mousing over the graph it will give the temperature or humidity at that time in history. See the next gure.

Figure 16: Open Tab and Data

The rooms tab has just those options and only has room name as its only parameter. See the next gure for the rooms user interface.

Figure 18: View of the Rooms Tab

The next tab devices tab. This tab shows the device id, name, location, ip address, type of device, number of ports on the device, warning, alert and critical temperatures, and lastly status. The administrator can add, remove and update devices.

Figure 19: Device Administration

The users tab allows new users and administrators to be created as well as updating and deleting users. This tab also allows veri cation of the users email and phon1 alerts. For a veri cation the user receives a text or email wio46l4 -4wio8we4wio8w6l4 -4wio8we

The room assignments tab is very important. This is where administrators

The last piece of the administration panel is the device tab. This device type tab. This is where the administrator can add di erent types of devices based on what they want to connect to their system. We have two device drivers that we have created. One is for the temperature @lert system which our raspberry pi also uses, and we have one for test sensors. If someone wanted to create their own driver they would make the connection here between the ui and the driver and then would need to create a driver to parse the XML.

One important feature that the client said was a must is that there must be support for mobile devices as we have stated before. The website on a mobile device looks almost identical to the desktop version of the site. Below are some screen shots from mobile devices looking at normal pages in the temperature window and the administration panel.

Now that the new site has been shown we will take a look at the nished product of the raspberry pi. First we have a picture of the device not assem-

Testing

There are two types of testing that we used in this project: Unit Testing

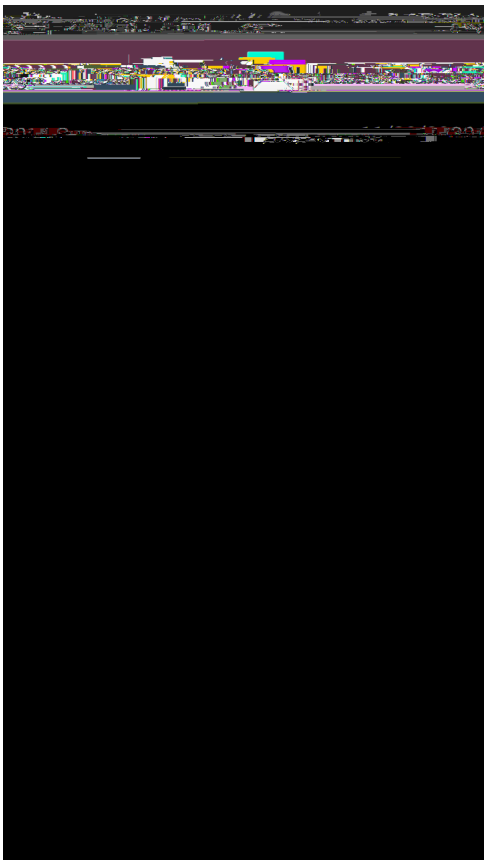


Figure 28: Text Message Alert and Verification Functional Testing

Figure 29: Email Alert Functional Testing

Figure 30: Multiple Sensor Testing

We also used the sensor units we build to do functional testing. We put the units near hot and cold things like an o/en or a refrigerator and measured the actual readings with a thermometer and humidity measuring device. We then checked the web site to make sure the sensor units were reporting accurate readings. Though it does not look pretty we were able to get it to show all sensors and create the graphs accordingly.

Security

Maintenance

After the completion of this project the current senior design team will o er no maintenance. All modi cations will be done by the sta at Western Michigan University and anyone else who uses the project. This being said there be added to increase the user experience for the project. These include

Adding a secure layer between the pi and the server,

Adding historical data logging to the Raspberry Pi,

Creating a way to retrieve historical data from the device if it was down for a period,

Upgrading the user control area to have group administration,

Making a 962gram that would 9oll servers for their data and add it to a graph.

Resources

Raspberry Pi

References

For everything raspberry pi we use these sites

Glossary

GUI
: Graphical User Interface. The windows a user interacts with.

MSP430 Launchpad
: A 16-bit microcontroller platform made by Texas Instruments.

RaspberryPi

Ownership

Licenses